IBM

IBM WebSphere Information Analyzer and Data Quality Assessment

- IBM WebSphere Information Analyzer overview
- Financial services business scenario
- IBM Information Server overview

Nagraj Alur
Reginald Joseph
Harshita Mehta
Jørgen Tang Nielsen
Denis Vasconcelos

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM WebSphere Information Analyzer and Data Quality Assessment

December 2007

Note: Before using this information and the product it supports, read the information in "Notices" on page xxvii.

First Edition (December 2007)

This edition applies to Version 8, Release 0, Modification 1 Rollup Patch 6d of IBM WebSphere Information Analyzer (product number 5724-Q36).

Contents

Figures

Tables

Examples

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo) ®
z/OS®
AIX®
DB2 Universal Database™
DB2®
IBM®
IMS™
MVS™
Rational®
Redbooks®
System z™
TCS®
WebSphere®

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

DataStage, Ascential DataStage, Ascential AuditStage, Ascential, are trademarks or registered trademarks of Ascential Software Corporation in the United States, other countries, or both.

EJB, Java, JDBC, JSP, J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Excel, Expression, Microsoft, SQL Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication discusses how to implement IBM WebSphere® Information Analyzer and related technologies using a typical financial services business scenario.

The audience for this book includes IT architects, Information Management specialists, and Information Integration specialists who are responsible for developing IBM WebSphere Information Analyzer on a Red Hat Enterprise Linux® 4.0 platform.

The book offers a step-by-step approach to implementing IBM WebSphere Information Analyzer on Red Hat Enterprise Linux 4.0 platform accessing information stored on IBM z/OS® and IBM AIX® platforms.

This book is organized as follows:

- Chapter 1, “IBM WebSphere Information Analyzer overview” on page 1 provides a detailed description of IBM WebSphere Information Analyzer, its architecture, configuration flow, and runtime flow.
- Chapter 2, “Financial services business scenario” on page 417 describes a step-by-step approach to implementing IBM WebSphere Information Server on a Red Hat Enterprise Linux 4.0 platform using a typical merger and acquisition financial services business scenario involving migration and data integration.
- Appendix A, “IBM Information Server overview” on page 523 provides a detailed description of IBM Information Server, its architecture, configuration flow, and runtime flow.
- Appendix B, “IBM Information Integrator Classic Federation setup” on page 553 describes the setup of IBM Information Integrator Classic Federation for use by IBM WebSphere Information Analyzer.
- Appendix C, “Miscellaneous tips regarding IBM WebSphere Information Analyzer” on page 559 includes a random collection of tips about using IBM WebSphere Information Analyzer.
- Appendix D, “Code and scripts used in the business scenario” on page 569 documents some of the code and scripts used in the migration and data integration business scenarios.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), San Jose Center.

Nagraj Alur is a a Project Leader with the IBM ITSO, San Jose Center. He holds a master's degree in computer science from the Indian Institute of Technology (IIT), Mumbai, India. He has more than 33 years of experience in database management systems (DBMSs) and has been a programmer, systems analyst, project leader, independent consultant, and researcher. His areas of expertise include DBMSs, data warehousing, distributed systems management, database performance, information integration, and client/server and Internet computing. He has written extensively on these subjects and has taught classes and presented at conferences all around the world. Before joining the ITSO in November 2001, he was on a two-year assignment from the Software Group to the IBM Almaden Research Center, where he worked on Data Links solutions and an eSourcing prototype.

Reginald Joseph is a Senior Business Intelligence Consultant- Data Integration with IBM India Software Labs (ISL) in Bangalore, India. He is IBM Certified in Database Administrator DB2® UDB V8.1 for Linux, UNIX®, and Windows®, and DB2 UDB V8.1 Family Fundamentals. His areas of expertise include IBM Datastage, Quality Stage, Profile Stage, Datawarehouse Enterprise Edition (DWE) and DB2 Alphablox. He has delivered numerous presentations on these subjects, and participated in POC, POT, RFP and RFQ on these topics. His responsibilities include resolving Lab Services and System Integrators technical issues, providing Solution Architectures, design reviews, pre-sales support, and technical skills and services to accelerate the integration of IBM Information Management software with Business Partner applications. His team of Information Management experts provide technical consultancy services to System Integrators in India and abroad. Clients include WIPRO, Infosys, TCS®, Cognizant, Deloitte, CapGemini, and Accenture.

Harshita Mehta is a BI Consultant (ETL and Information Management) with IBM Lab Services and Solutions in India. She has 2.5 years of experience in Business Intelligence, Customer Relationship Management (CRM), and data warehousing. Her areas of expertise include ETL, data warehousing, and Siebel® CRM applications. She has worked extensively on competitive products such as Informatica Powercenter, IBM Information Server, Ascential™ DataStage™ 7.5, Siebel 7.7 OLAP-OLTP and DAC. Her areas of interest include data warehouse design/modelling, and ETL conceptualization. She is currently involved in IBM Information Server assignments in India and ASEAN region.

Jørgen Tang Nielsen is a Senior IT Specialist with IBM Denmark. He currently works with the Information Server suite in the Software Group, IBM Denmark. He has more than 10 years experience with application development, primarily on the z/OS platform, as well as a number of years of working with DB2 database administration in an application development environment. Jørgen is an IBM Certified Database Administrator with DB2 Universal Database™ V8.1 for z/OS. He holds a Master of Science in Engineering degree from the Technical University of Denmark.

Denis Vasconcelos is a Data Specialist with IBM Brazil. He had over five years experience with several non-IBM data management systems before joining IBM in 2006. His areas of expertise include database administration, data modeling, heterogeneous database migration, and project management. Denis has a Bachelor's degree in computer science and a post-graduate degree in project management.

Our very special thanks to Harald Smith, Christopher (Cass) Squire, and Daphne Png for enlightening us about the data quality assessment process.

Thanks to the following people for their contributions to this project:

Christopher (Cass) Squire
IBM Global Business Services

Joseph Bangs
Barry Scott Rosen
Shuyan He
Ravi Medikonduru
David Meeks
Harald Smith
IBM Westboro

Atul Chadha
Maribet Mason
Daphne Png
Asim Singh
IBM Silicon Valley Laboratory

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

IBM WebSphere Information Analyzer overview

In this chapter, we provide a detailed description of IBM WebSphere Information Analyzer, its architecture, and the configuration, execution, and monitoring of its main functions. We also include an overview of IBM WebSphere AuditStage and a data quality assessment methodology.

We cover the following topics in this chapter:

- Data quality assessment (DQA) methodology
- IBM WebSphere Information Analyzer architecture
- Main functions
- Main components
- Setting up your system
- Column analysis
- Primary key analysis
- Foreign key analysis
- Cross domain analysis
- Publish analysis results
- IBM WebSphere AuditStage business rule validation
- Baseline analysis
- Reports

1.1 Introduction

Organizations face an information challenge that begins with locating the information, getting the information when it is needed, and providing it in the form that it is needed. When the data is found, the next step is to discern further insights from it. Information validity and control are additional concerns. The challenges only mount if businesses cannot ensure access to authoritative, consistent, timely, and complete information. In fact, a wide range of corporate initiatives fail because of poor data quality.

Data quality can be broadly defined qualitatively as information that you can trust or ensuring that data, at a particular point in time, is suitable for its purpose. A more specific quantitative definition can include the level of compliance attained by an enterprise's data environment to independently defined rules that describe that data environment. The emphasis is on the business user's perception of data quality—what is delivered to the user and the semantics of the data presented. In this case, data quality might well depend upon the data itself (such as correct data types, consistent formatting, retrievability, and usability) and the processes or applications that deliver this data to the business user.

Important: To assess and improve data quality, you need a concerted, cooperative effort from a number of individuals, such as business users, subject matter experts (SME[a]), and IT data analysts (DA[b]), using state of the art tools and technologies. These tools and technologies need to assist individuals to detect poor quality, fix poor quality data, identify the causes and entry points of poor quality data, develop processes to trap poor quality data at the point of origin, and then monitor the success of the efforts to improve data quality.

[a] An SME has a thorough understanding of business requirements and is responsible for representing and interpreting the requirements for a data integration initiative. An SME acts as the liaison between the IT Project Team and business users. The SME also assists the IT Project Team prioritize requests for enhancements, define standard business metrics, terms and definitions, and define validation criteria.

[b] The DA is knowledgeable about the relevant data systems, including the technology to access this data. The DA understands how the data is processed and flows through the systems. The DA also has experience with technology to conduct data profiling and analysis. Finally, the DA ensures that decisions and requirements can be met from an IT perspective.

IBM Information Server addresses these requirements with an integrated software platform that provides the full spectrum of tools and technologies required to address data quality issues. This includes data profiling[1] (IBM WebSphere Information Analyzer and IBM WebSphere AuditStage), data cleansing (QualityStage), and data movement and transformation (DataStage). For an overview of IBM Information Server, refer to Appendix A, “IBM Information Server overview” on page 523.

IBM WebSphere Information Analyzer (the focus of this book) is a new module of IBM Information Server that represents the next generation data profiling and analysis tool. It is designed to help business and data analysts understand the content, quality, and structure of their data sources by automating the data discovery process. The IBM WebSphere Information Analyzer product module helps increase the productivity of data personnel as well as improve return on investment (ROI) and time to benefit of data-intensive projects.

IBM Information Server's data profiling, analysis, and monitoring capabilities lets you:

- Optimize your development resources by understanding your data before starting development because it identifies bad, missing, incorrect, and redundant data.
- Eliminate the risk and uncertainty of using bad data before you begin your data integration project. Avoid the 1/10/100 phenomenon[2] of downstream system costs due to unexpected data problems.
- Reduce data analysis time by as much as 50% using automated data profiling and analysis capabilities.
- Maintain the *quality* health of your data systems through monitoring so you can provide continuous feedback on data quality, detect unexpected changes, identify quality deterioration, and reinforce business users' confidence in the data.
- Integrate with other IBM Information Server modules by sharing metadata.

IBM WebSphere Information Analyzer profiles and analyzes data so that you can deliver trusted information to your users. It can scan your complete data automatically (or samples of it) to determine quality and structure. This analysis aids you in understanding the inputs to your integration process, ranging from individual fields to high-level data entities. Information analysis also enables you

[1] *Data profiling* is a data management practice that reveals the content, quality, and structure of the data. It is vital to the success of any business integration effort.

[2] This rule states that cost to fix a defect increases exponentially the later in the development life cycle that it is identified. A defect caught in requirements phase costs a factor of 1 (1x) to fix. A defect caught in construction costs 10 times as much as during the requirements phase. A defect caught in production costs up to 100 times as much as in the requirements phase.

to correct problems with structure or validity before they affect your project. While analysis of source data is a critical first step in any integration project, you must monitor the quality of the data continually. IBM WebSphere Information Analyzer enables you to treat profiling and analysis as an ongoing process and create business metrics that you can run and track over time.

In the following sections, we provide an overview of a typical data quality assessment effort, describe IBM WebSphere Information Analyzer's architecture, and using a sample set of tables and representative data, the configuration, execution and monitoring of its key functions of Column Analysis, Primary Key Analysis, Foreign Key Analysis, Cross Domain Analysis, Publish Analysis Results, and Baseline Analysis. Also covered is IBM WebSphere AuditStage (a bundled product component delivered with IBM WebSphere Information Analyzer) that provides business rule validation.

Note: This IBM Redbooks publication does not cover all the functions and features of IBM WebSphere Information Analyzer or those of IBM WebSphere AuditStage. You should refer to the resources described in "Related publications" on page 593 for complete details about IBM WebSphere Information Analyzer and IBM WebSphere AuditStage.

1.2 Data quality assessment (DQA) methodology

DQA is the process of exposing technical and business data issues in order to plan the data integration effort most likely to succeed within budget and time constraints.

- Technical quality issues based on target technical standards are generally easy to discover and correct, such as:
  - Different or inconsistent standards in structure, format, or values
  - Missing data, default values
  - Spelling errors, data in wrong fields
  - Buried information in free-form fields
- Alternatively, business quality issues are more subjective and are associated with business processes such as generating accurate reports, ensuring that data driven processes are working correctly, and shipments are going out on time.

  Because accuracy, timeliness, and correctness are subjective measures, it requires the involvement of the business community to assess the business quality of the data.

> **Note:** For enterprise level initiatives, such as ERP implementations or system consolidation, integration challenges at both the business and technical levels generally revolve around the semantic reconciliation of master data objects such as customer, product, and vendor.

Because the business is the ultimate recipient and user of the data that results from the integration effort, the success of a DQA is greatly dependent upon the ability and commitment of the business community to participate in the process, and more importantly, to resolve semantic and business rule differences at the functional level.

We describe the DQA methodology briefly in the following sections:

- Data assessment approach
- Data assessment tools
- Data assessment benefits

1.2.1 Data assessment approach

Data quality assessment is typically associated with a data integration effort, and is likely to occur at multiple stages of the data integration effort. The specific steps would depend upon the particular data integration effort.

The fundamental step in any level of analysis is to establish the effort involved in:

- Process of profiling
- Subsequent assessment/analysis activity

This data assessment approach involves the following key steps:

1. Identify profiling requirements

   This involves identifying the goals of profiling and analysis. The specific result to be achieved must be identified. You also need to identify the time frame in which this profiling and analysis should occur.

   Goals are usually geared around specific scenarios as follows:

   – For a Data Integration effort such as merging an acquired business into an existing system, the goals include validating core customer-related information (such as name, address, accounts, key flags, and codes), identifying incompatible or unassociated data, and documenting key decisions for developers. The scope of profiling are limited typically to a specific set of systems or tables that are relevant to integrate.

   – For an SAP® data quality effort, the goals are the extraction, cleansing, and reloading of core SAP tables. The scope of profiling focuses typically on master data tables, identifying candidate records or columns for further cleansing, and annotation of that data with key specifications on how to standardize or merge.

2. Identify relevant data sources and access considerations

   You need to identify the sources necessary to meet the goal and ensure that they are included in the project. Details of how and when to access these data sources should be investigated and documented. Finally, you need to establish appropriate security around these data sources.

3. Identify an execution plan

   You must designate the person or persons who will run the profiling and analyze the results. You also need to prioritize the tasks so that items most important to achieving the goals are addressed first. You need to perform these profiling and analysis tasks within the constraints or restrictions of access and availability.

4. Identify the need for further exploration

   As analysis proceeds, you need to note additional areas of concern or interest to the overall goal. You can adjust goals or time frames or expand analysis as required to address these concerns and interests.

5. Take advantage of the metadata repository

   As you gather and acquire information during profiling and analysis, record and annotate this information in the (shared) repository to ensure an expanding knowledge base to all members of the data integration initiative.

6. Iterate

   Based on goals and needs, you need to iterate the afore-mentioned process to assess changes over time.

While data profiling brings data issues to light, it requires data and business analysts to review the results, draw conclusions about the problem, come up with an action plan to address the problems, execute the action plan, and then validate the success of the executed plan.

The most common challenges faced by Data Analysts in analyzing data include:

- Understanding the scope and objectives

  If you do not know what you are trying to achieve, it will be difficult to identify an anomaly in the data. Identifying critical sources for analysis upfront is a significant challenge, as is the need to weed out and annotate extraneous sources.

- Articulating clearly the goals and deliverables

  If you do not make consistent and standard annotations on results, it can cause others to have to review and re-do profiling work.

- Realizing that direct analysis must follow profiling in the process

  Running a IBM WebSphere Information Analyzer job to profile the data delivers only a set of statistics and system inferences. The Data Analyst must then review that information to bring insight and clarity to those statistics and inferences. Table 1-8 on page 129 summarizes the series of key dimensions or questions on which a Data Analyst needs to focus when evaluating the results of a IBM WebSphere Information Analyzer Column Analysis profiling job.

- Keeping sight of the big picture when presented with overwhelming numbers of tables and attributes

  Focus on what is relevant rather than trying to do everything at one—also known as *boiling the ocean*. Establishing goals and time lines helps maintain the focus of the Data Analyst. You need to build out the knowledge base incrementally and take advantage of prior knowledge captured in the repository to avoid repeating previously performed profiling and analysis tasks.

- Remembering that profiling can be used throughout a project life cycle

  Profiling can be more than just an upfront source analysis activity. The data analyst or developer can continue to use data profiling to assess test data used in development or quality assurance (QA) as follows:

  – To evaluate and baseline the results and effectiveness of development during unit testing or system testing.

  – To validate data prepared for system load activities.

Figure 1-1 on page 8 provides a high-level overview of the main steps of DQA process. Those steps are as follows:

1. Prepare the data for assessment

   Select the data sources to be investigated and analyzed.

2. Conduct data discovery

   The DA and SME perform the investigation and analyses using tools such as IBM WebSphere Information Analyzer and IBM WebSphere AuditStage. This discovery involves checking metadata integrity, structural integrity, entity integrity, relational integrity, and domain integrity.

3. Document data quality issues and decisions

   After all information about data quality is known, you can make and implement the appropriate data alignment and cleansing decisions.

Note: Typical DQA durations are between four to eight weeks. In short, focused development efforts, these will be kept tight, although assessment can be ongoing and iterative. In longer, 6 month or longer development efforts, DQA durations will typically run 6+ weeks and are a key part of requirements definition.

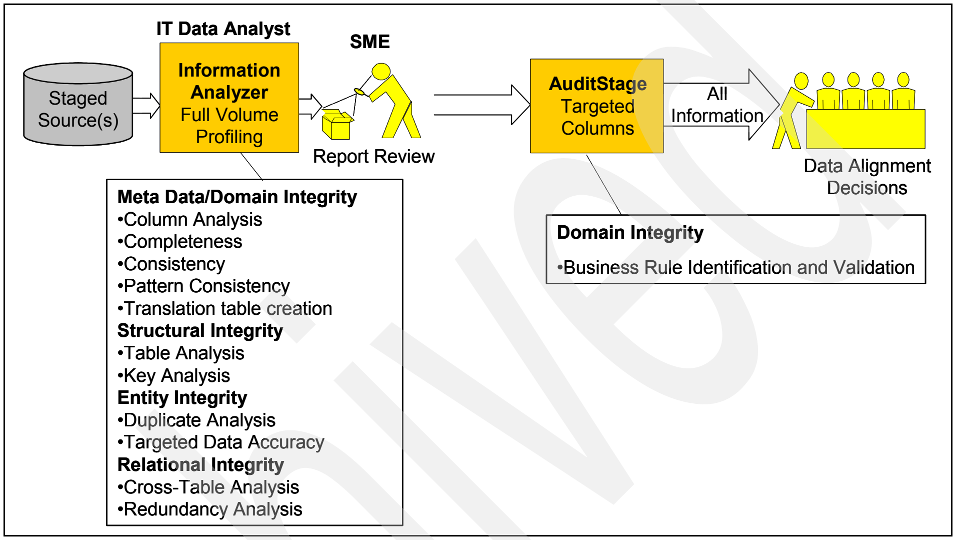


Figure 1-1 DQA approach - data assessment

In the sections that follow, we describe each of these steps in more detail.

Prepare the data for assessment

Data integration efforts typically involve very large volumes of data. Given budget and time constraints, it is necessary to scope down the data quality assessment effort for maximum gain.

The recommendation is to focus on the master data that constitutes the critical success factor (CSF) for the initiative and then assess that master data that needs to integrated for gaps in expected data quality. This approach is estimated to contribute 80% towards the success of the data integration effort.

As mentioned earlier, ensuring data quality is based on shared knowledge and shared responsibility of the DA and SME. They have to collaborate to investigate and determine how to resolve identified data quality issues. It also involves estimating the quality condition of the data. Data quality assessment is futile without such collaboration between the DA and SME.

This task is performed by the DA and SME, and the deliverable of this task is the extraction of the data staged for data assessment.

Conduct data discovery

This task involves verifying metadata integrity, domain integrity, structural integrity, and relational integrity aspects as follows:

- Metadata integrity

  Metadata integrity involves knowing what the field is.

  The DA makes basic interpretations about whether the data can be understood.

  If the data cannot be understood, the DA expands the understanding of the data with basic definitions of the data and the data values. Data definitions are reviewed on a file-by-file and field-by-field basis to assess the integrity of each individual domain including:

  – Business definition of field usage
  – Definition of field size and type
  – Definition of expected data patterns or formats
  – Definition of expected and acceptable data values
  – Definition of field relationships or hierarchies

  The DA annotates and reports what is ambiguous or anomalous.

  The basic steps in verifying metadata integrity are:

  a. First identify the fields that need further definition.

     Check whether definitions are available. They are typically found in external documents or data models. However, many sources have no definitions

  b. Determine whether the fields understandable.

     Many system definitions use cryptic codes or shorthand, which are not intuitively understandable because they have no clear meaning (for example, fields with names *Flag1*, *Flag2*, *Idx2*, and *MktIdx1*).

  c. Add aliases, definitions, and business terms to the tables and columns (in the Metadata Management functions of IBM WebSphere Information Analyzer or in conjunction with the IBM WebSphere Business Glossary) for a clearer understanding. For example, field ADDR_2 might be given an alias of *address line 2*, with a definition of “This is the second line of address, primarily utilized for long addresses or extraneous attention details,” and a business term of *Billing Address*, which represents the location where a bill is sent to a customer.

- Domain integrity

  Domain integrity involves knowing what the field contains and whether it is right.

  The DA reviews and makes decisions about the completeness, validity, and conformity of the data to known or expected data usage. This includes checking for:

  - Conformity to metadata (for example, the inferred metadata is consistent with the defined metadata such as field length).
  - Conformity to expected data patterns or formats.
  - Conformity to expected data values (for example, the data values are consistent with the understanding of the data usage or business rules).
  - Identification of multiple components (or mixed domains) within a single field that might make understanding or accuracy difficult to achieve (for example, an address field also has an individuals name).

  The basic steps in verifying domain integrity are:

  a. First identify what is relevant.

     Weed out and annotate the extraneous fields. If no data is available, it is either irrelevant or a gap. Look for a data classification of *Unknown* or a single constant value at the summary level in the IBM WebSphere Information Analyzer Column Analysis results. (We provide more information about this later in 1.7, “Column analysis” on page 109.)

  b. Next identify fields that needs further exploration.

     Take advantage of Data Classifications that are provided by IBM WebSphere Information Analyzer Column Analysis inferred data classifications to drive analytical review.[3] The DA can expand on this assessment through customized Data Sub-classifications or User Classifications for unique organizational needs. Table 1-8 on page 129 provides an overview of the various data classifications, their definitions, and their potential assessments.

  c. Annotate and Report what is anomalous (in IBM WebSphere Information Analyzer and potentially updating information in the IBM WebSphere Business Glossary).

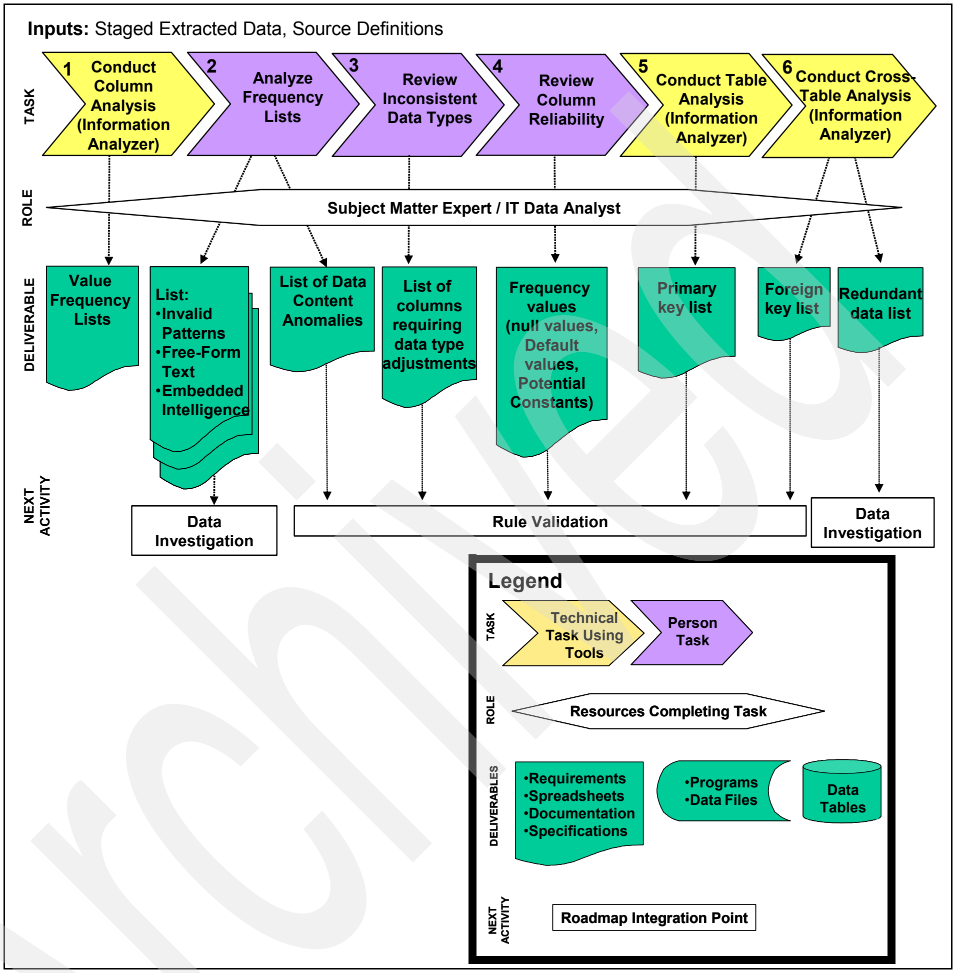
  d. Mark Reviewed what is done (in IBM WebSphere Information Analyzer).

[3] Identifiers, Indicators, Codes, Dates & Timestamps, Quantities, and Text all have unique focal points. What this means is that for each classification of data, the DA focuses on specific types of issues. With Identifiers the DA primarily assesses uniqueness, though formatting might also be important. With Codes the DA assesses clarity, conformance to known or established reference values, consistent usage, and so on.

- Structural integrity verification involves using a primary key or identifier by which to uniquely identify an instance of master data.
- Relational integrity verification involves the following:
  - From a technical perspective, that the primary key and foreign key relationships between tables are valid
  - From a business perspective, that the data complies with business rules and is consistent within and across data sources

These tasks are shared by the DA and SME.

Figure 1-2 shows the data profiling tasks in detail, the deliverables of each analyses, and the next activity to be performed.



Inputs: Staged Extracted Data, Source Definitions
TASK
1 Conduct Column Analysis (Information Analyzer)
2 Analyze Frequency Lists
3 Review Inconsistent Data Types
4 Review Column Reliability
5 Conduct Table Analysis (Information Analyzer)
6 Conduct Cross-Table Analysis (Information Analyzer)
ROLE
Subject Matter Expert / IT Data Analyst
DELIVERABLE
Value Frequency Lists
List: •Invalid Patterns •Free-Form Text •Embedded Intelligence
List of Data Content Anomalies
List of columns requiring data type adjustments
Frequency values (null values, Default values, Potential Constants)
Primary key list
Foreign key list
Redundant data list
NEXT ACTIVITY
Data Investigation
Rule Validation
Data Investigation
Legend
TASK
Technical Task Using Tools
Person Task
ROLE
Resources Completing Task
DELIVERABLES
•Requirements •Spreadsheets •Documentation •Specifications
•Programs •Data Files
Data Tables
NEXT ACTIVITY
Roadmap Integration Point

Figure 1-2 DQA approach - data profiling

Figure 1-3 shows the rule validation tasks, the deliverables, and the next activity to be performed.

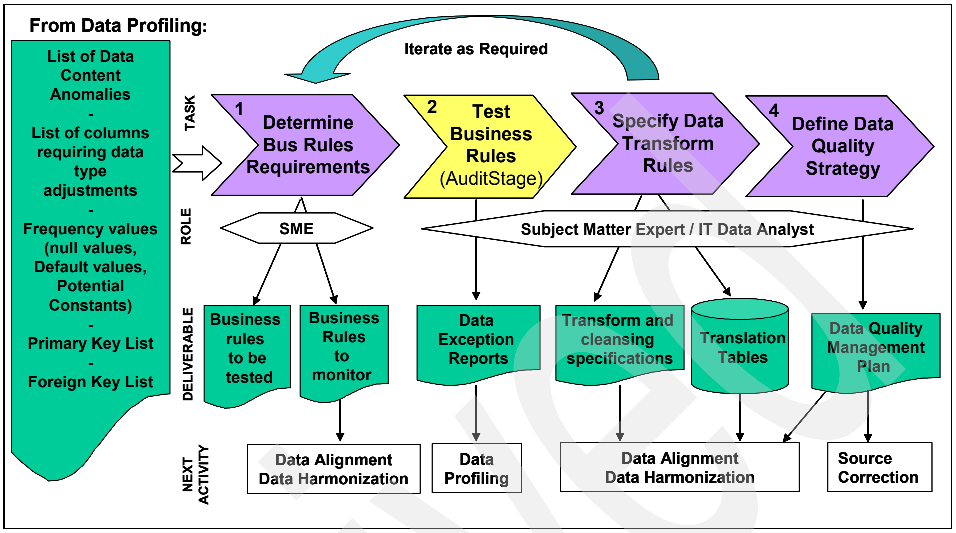

From Data Profiling:
List of Data Content Anomalies
-
List of columns requiring data type adjustments
-
Frequency values (null values, Default values, Potential Constants)
-
Primary Key List
-
Foreign Key List
Iterate as Required
TASK
1 Determine Bus Rules Requirements
2 Test Business Rules (AuditStage)
3 Specify Data Transform Rules
4 Define Data Quality Strategy
ROLE
SME
Subject Matter Expert / IT Data Analyst
DELIVERABLE
Business rules to be tested
Business Rules to monitor
Data Exception Reports
Transform and cleansing specifications
Translation Tables
Data Quality Management Plan
NEXT ACTIVITY
Data Alignment Data Harmonization
Data Profiling
Data Alignment Data Harmonization
Source Correction

Figure 1-3 DQA approach - rule validation

Figure 1-4 and Figure 1-5 show the data investigation of free-form text such as names, addresses, and descriptions. This investigation is performed by tools such as IBM WebSphere QualityStage, which is beyond the scope of this book but will be covered in an upcoming IBM Redbooks publication. Briefly, this process enables you to standardize free-form text in order to match potentially duplicate master data records, as well as enhance, cleanse, and determine relationships between master data records.

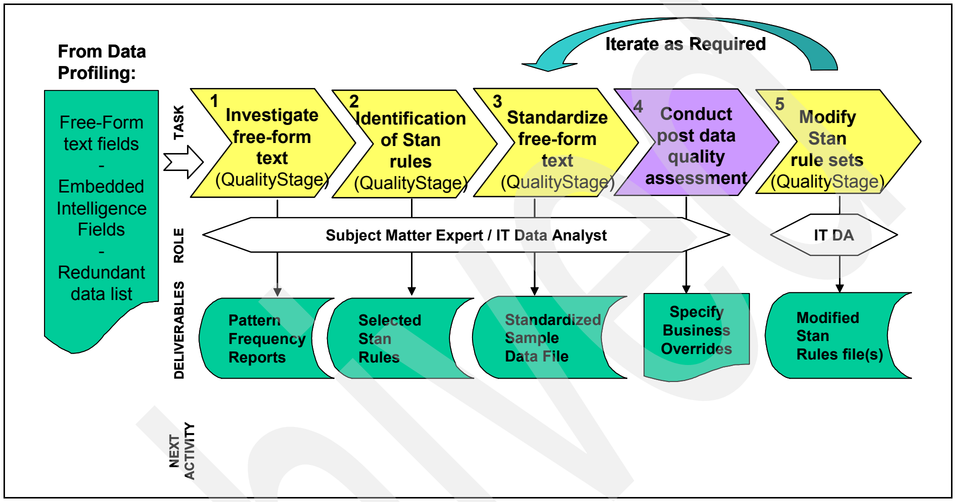

From Data Profiling:
Free-Form text fields - Embedded Intelligence Fields - Redundant data list
Iterate as Required
TASK
1 Investigate free-form text (QualityStage)
2 Identification of Stan rules (QualityStage)
3 Standardize free-form text (QualityStage)
4 Conduct post data quality assessment
5 Modify Stan rule sets (QualityStage)
ROLE
Subject Matter Expert / IT Data Analyst
IT DA
DELIVERABLES
Pattern Frequency Reports
Selected Stan Rules
Standardized Sample Data File
Specify Business Overrides
Modified Stan Rules file(s)
NEXT ACTIVITY

Figure 1-4 DQA approach - data investigation free-form text analysis

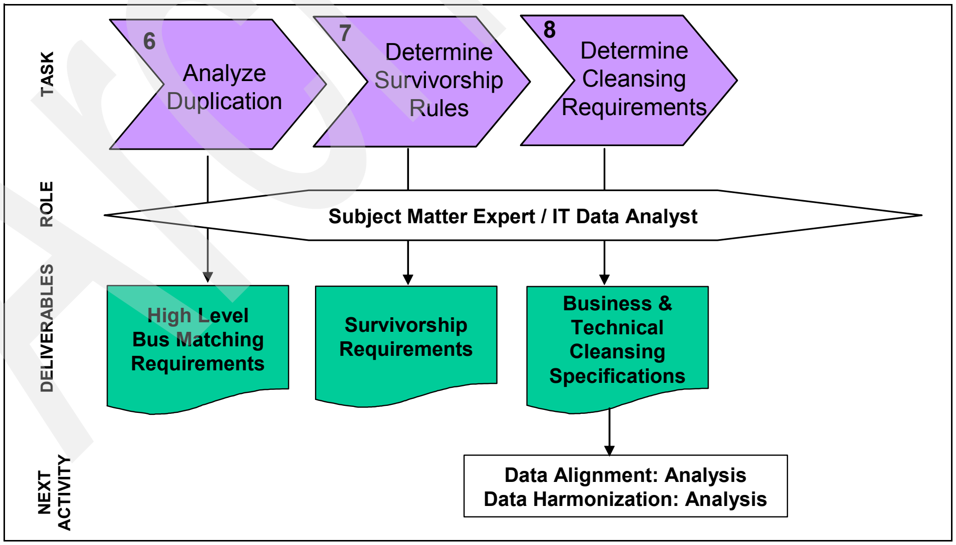

TASK
6 Analyze Duplication
7 Determine Survivorship Rules
8 Determine Cleansing Requirements
ROLE
Subject Matter Expert / IT Data Analyst
DELIVERABLES
High Level Bus Matching Requirements
Survivorship Requirements
Business & Technical Cleansing Specifications
NEXT ACTIVITY
Data Alignment: Analysis
Data Harmonization: Analysis

Figure 1-5 DQA approach - data investigation duplicate analysis and cleansing specification

Document data quality issues and decisions

Data harmonization is the effort to consolidate, integrate, and cleanse data from across all systems to support aligned business definitions, rules, and quality expectations of the data, for the integration initiative and eventually at the enterprise level.

Finally, document the data quality issues and decisions made to resolve the issues and to communicate these results to the team or teams in the integration effort for appropriate action.

Table 1-1 summarize the tasks, roles, and deliverables that are associated with the DQA process involving the use of IBM WebSphere Information Analyzer (IA) and IBM WebSphere AuditStage (AS).

Table 1-2 on page 17 summarizes the tasks, roles, and deliverables that are associated with more specific data cleansing requirements of free-form text, such as names and addresses, or unduplication of free-form text data using IBM WebSphere QualityStage (QS).

Table 1-1 Tasks, roles, and deliverables associated with DQA involving IA and AS

#	Task	Accountable roles	Deliverable
1	Determine the scope of data for assessment, and the expected quality condition of the data	SME, DA	Data extracted and staged for assessment
2	Run IBM WebSphere Information Analyzer Column Analysis on the data	DA	Column Analysis results
3	Analyze Column Analysis results against expected quality. ▶ Confirm column data values for completeness, validity, structure and format ▶ Specify mapping values, if any ▶ Document issues and resolution decisions	SME, DA	Columns reviewed for domain, structure and format integrity, and documented for the integration effort in IBM WebSphere Information Analyzer
4	Create Reference Tables (IA) for use with the integration team	DA	Lookup/Mapping tables available for export to extract, transform and load (ETL) projects
5	Capture analysis results as a baseline, if required (IA)	DA	Column statistics stored for future comparative analysis

#	Task	Accountable roles	Deliverable
6	Run Primary Key Analysis (IA) on tables, and Duplicate Analysis (IA) as needed	DA	Identified candidate primary keys and duplicate records for the table
7	Analyze primary key candidates and confirm selected key(s)	SME, DA	Column(s) selected as the primary key for the table/file
8	Run Foreign Key Analysis (IA) across tables	SME, DA	Identified PK/FK candidate relationships found in the data
9	Analyze PK/FK candidate relationships and confirm selected FK columns	SME, DA	Column(s) selected as foreign keys in the table
10	Run Cross-Table Analysis (IA) across tables	DA	Identified redundant column candidates in the data
11	Analyze redundant column candidates and confirm selected redundancy	SME, DA	Column(s) selected as redundant data
12	Determine specific business rules in order to test data to confirm/resolve data quality compliance	SME	List of business rules requiring data compliance
13	Conduct business rule tests against the data using IBM WebSphere AuditStage[a] (AS)	DA	Business rule compliance results
14	Iterate data analysis process using IA and AS, as needed, based on the data exception results	SME, DA	Documented specifications for the data integration or data quality management efforts

[a] In IBM WebSphere Information Analyzer Version 8.0.1, IBM WebSphere AuditStage is a complementary and bundled product that provides functionality to test data against business rules.

Table 1-2 Tasks, roles, and deliverables associated with free-form text data involving QS

#	Task	Accountable roles	Deliverable
1	Select data columns requiring further analysis for the purposes of standardization, and matching	SME, DA	Data extracted and staged for QS analysis
2	Conduct Word Investigate analysis (QS)	DA	Pattern frequency and token reports
3	Identify standardization rules for the data	SME, DA	Selected/Created standardization rules to test
4	Conduct Standardization (QS) tests on data and profile/analyze output (IA)	DA	Sample standardized data file and Column Analysis results for standardized output
5	Iterate analysis using QS as needed, based on rule overrides required. ► Determine high-level business rules for matching duplicate records, if needed ► Determine high-level survivorship rules for integrating duplicate records, if needed	SME, DA	Documented specifications for the data cleansing and integration efforts

1.2.2 Data assessment tools

The IBM Information Server product provides three tools for data assessment: IBM WebSphere Information Analyzer, IBM WebSphere QualityStage, and IBM WebSphere AuditStage:

- *IBM WebSphere Information Analyzer* enables you to quickly discover condition of data in large volumes of data in a fraction of the time that could be handled manually. Through its Column Analysis, Primary Key Analysis and Cross-Table Analysis functions, IBM WebSphere Information Analyzer enables systematic analysis and reporting of results, thereby allowing the data analyst and subject matter expert to focus on the real problem of data quality issues.
- *IBM WebSphere QualityStage* complements IBM WebSphere Information Analyzer by investigating free-form text fields such as names, addresses, and descriptions. IBM WebSphere QualityStage allows you to define rules for standardizing free-form text domains which is essential for effective probabilistic matching of potentially duplicate master data records. This level of sophisticated data assessment is critical to understanding the total

cleansing effort required for a data integration project. IBM WebSphere QualityStage will be covered in an upcoming IBM Redbooks publication.

- *IBM WebSphere AuditStage* enables you to apply professional quality control methods to manage the accuracy, consistency, completeness, and integrity of information stored in databases. By employing technology that integrates Total Quality Management (TQM) principles with data modeling and relational database concepts, IBM WebSphere AuditStage diagnoses data quality problems and facilitates data cleanup efforts. IBM WebSphere AuditStage is discussed briefly in 1.12, “IBM WebSphere AuditStage business rule validation” on page 336.

Figure 1-6 summarizes the functions provided by IBM WebSphere Information Analyzer, IBM WebSphere QualityStage, and IBM WebSphere AuditStage.

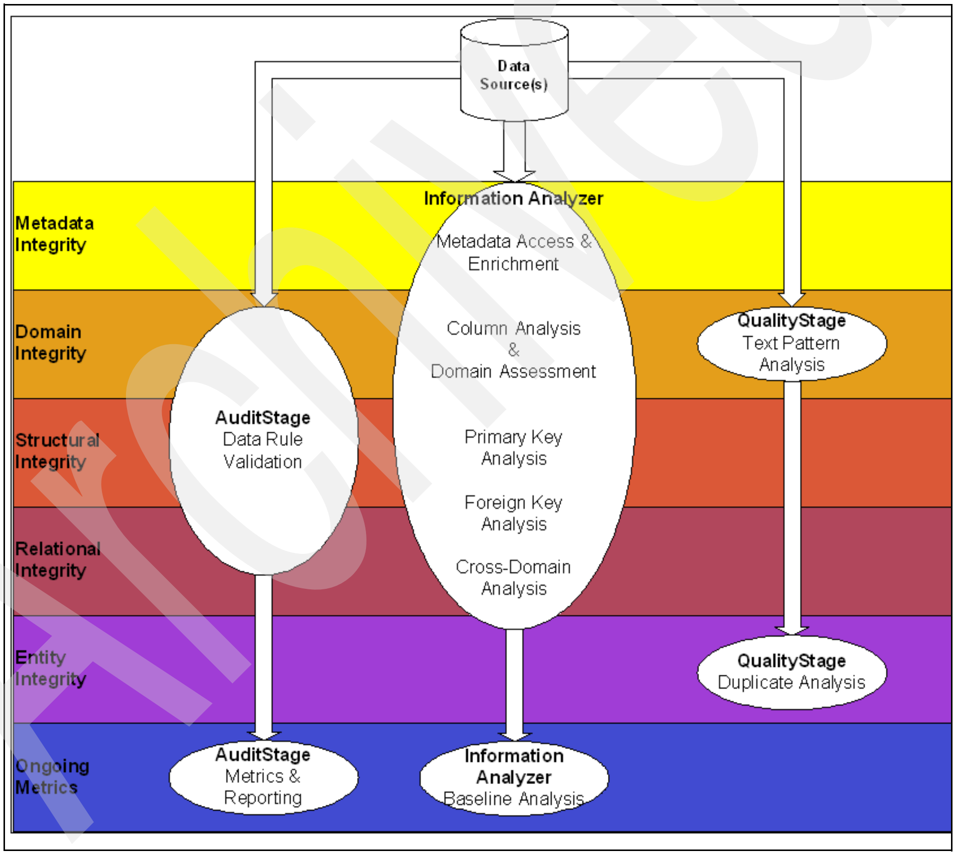

Data Source(s)
Metadata Integrity
Domain Integrity
Structural Integrity
Relational Integrity
Entity Integrity
Ongoing Metrics
Information Analyzer
Metadata Access & Enrichment
Column Analysis & Domain Assessment
Primary Key Analysis
Foreign Key Analysis
Cross-Domain Analysis
AuditStage Data Rule Validation
QualityStage Text Pattern Analysis
QualityStage Duplicate Analysis
AuditStage Metrics & Reporting
Information Analyzer Baseline Analysis

Figure 1-6 Data assessment tools functionality

1.2.3 Data assessment benefits

Industry analysts claim that data integration projects involving data transformation construction and testing that fail, do so around 80% of the time because of data quality issues.

A systematic approach to data quality assessment (backed by a strong business commitment), using products such as IBM WebSphere Information Analyzer and IBM WebSphere QualityStage, can help quantify the full scope of the data cleansing and transformation effort that is required to ensure the success of any data integration project early in the planning cycle. A relatively small fixed cost investment in data quality assessment can have a significant positive impact on the success of a data integration project.

Data quality assessment results in a number of benefits including:

- Business users take responsibility for data quality issues and help to resolve them.
- Project planning activities and time frames are more accurate because it takes into account the true status of data quality to be dealt with.
- Developers are able to address data quality issues proactively, thereby reducing any significant rework.
- Project team members have a significantly higher degree of confidence that the data integration effort would have a high probability of success.
- The enterprise shares a common understanding and management responsibility for its master data quality.

Early data quality assessments can reduce the risks of your data integration project significantly.

1.3 IBM WebSphere Information Analyzer architecture

IBM WebSphere Information Analyzer evaluates the content and structure of your data for consistency and quality.

IBM WebSphere Information Analyzer helps you to assess the quality of your data by identifying inconsistencies, redundancies, and anomalies in your data at the column, table, and cross-table levels. IBM WebSphere Information Analyzer also makes inferences about the best choices regarding data structure. Inferences help you to learn more about the optimal structure of your data and what you can do to improve the quality of your data. In addition, IBM WebSphere Information Analyzer provides a mechanism called *Baseline Analysis* to help you assess whether a data quality procedure that you implemented has resulted in an improvement in data quality by comparing a prior version of analysis results with the current analysis results for a given data source.

The main functions provided by IBM WebSphere Information Analyzer are:

- Column Analysis
- Primary Key Analysis
- Foreign Key Analysis
- Cross Domain Analysis
- Publish Analysis Results
- Baseline Analysis
- Reports

We describe these function in 1.4, “Main functions” on page 20.

IBM WebSphere Information Analyzer is supported by a broad range of shared suite components in IBM Information Server. Standard services are provided for data source connectivity, system access and security, logging, and job scheduling. IBM WebSphere Information Analyzer shares discrete services with other IBM Information Server components. We describe the main components of IBM WebSphere Information Analyzer and their interaction in 1.5, “Main components” on page 23.

Before you can begin performing the various analyses functions provided by IBM WebSphere Information Analyzer, you must set up your system as described in 1.6, “Setting up your system” on page 28. It involves creating and opening a project, establishing connectivity to a data source, configuring system resources, importing metadata, configuring the project, and setting up security.

1.4 Main functions

IBM WebSphere Information Analyzer provides the following main functions:

- *Column Analysis* creates a frequency distribution that summarizes the results for each column, such as statistics and inferences about the characteristics of your data. The frequency distribution is used to find anomalies in your data. The column analysis process incorporates four analyses as follows:
  - *Domain* analysis allows you to identify invalid and incomplete data values through techniques of direct identification, range assignment, or reference table comparison.
  - *Data classification* analysis infers a data class for each column in your data.

  - *Format* analysis creates a format expression for the values in your data. A format expression is a pattern that contains a character symbol for each distinct character in a column.
  - *Data properties* analysis compares the accuracy of defined properties about your data before analysis to the system-inferred properties that are made during analysis. Data properties define the characteristics of data such as field length or data type.

  We describe Column Analysis in greater detail in 1.7, “Column analysis” on page 109

- *Primary Key Analysis* involves evaluating the values and primary keys in the frequency distribution of a column for uniqueness. Defined primary keys are not in the functional distribution. To be considered a candidate for a primary key, a column must contain a higher percentage of unique values than the system-defined threshold. The system-defined threshold is a setting that you can modify to allow a specific amount of data variances. A primary key can be single column or multi-column.

  We describe Primary Key Analysis in greater detail in 1.8, “Primary key analysis” on page 224

- *Foreign Key Analysis* involves assessing your data for relationships between tables. The values in your data are evaluated for foreign key candidates and defined foreign keys. A column might be inferred as a candidate for a foreign key when the values in the column match the values of an associated primary key. After a foreign key analysis job completes, you can run a referential integrity analysis job on your data. *Referential integrity analysis* is an analysis that you use to fully identify violations between foreign key and primary key relationships. During a referential integrity analysis job, foreign key candidates are investigated at a concise level to ensure that they match the values of an associated primary key. A foreign key can be single column or multi-column.

  We describe Foreign Key Analysis in greater detail in 1.9, “Foreign key analysis” on page 264

- *Cross Domain Analysis* involves determining whether multiple columns share a common domain. A common domain exists when multiple columns contain overlapping data. Columns that share a common domain might signal the relationship between a foreign key and a primary key, which you can investigate further during a foreign key analysis job. However, most common domains represent either checks for consistency that must be validated or redundancies between columns. If there are redundancies in your data, you might want to use a data cleansing tool to remove them, or use normalization.

Important: Cross Domain Analysis is the primary tool for assessing data consistency.

We describe Cross Domain Analysis greater detail in 1.10, “Cross domain analysis” on page 301.

- *Publish Analysis Results* causes analysis results to be published to the metadata repository. You might want to publish statistics and annotations for a table or column to provide developers or data stewards in additional suite components, such as IBM WebSphere DataStage or IBM WebSphere Business Glossary, access to analytical results.

  We describe Publish Analysis Results in greater detail in 1.11, “Publish analysis results” on page 323.

- *Baseline Analysis* is used to compare a prior version of analysis results with the current analysis results for a given data source. If differences between both versions are found, you can assess the significance of the change, such as whether the quality has improved.

  We describe Baseline Analysis greater detail in 1.13, “Baseline analysis” on page 379.

- Reports can be generated that summarize analysis results and show details about your project. Reports are saved in the metadata repository and can be accessed by any user who is authorized to view them. Reports can show information in multiple ways. For example, analysis results can be displayed as the actual data that the results refer to, or, they can be shown in a graph or chart. Graphs and charts display general information about an object such as the percentage of columns that have been analyzed in a data source. Graphs and charts also highlight issues that might otherwise be difficult to locate in the text of a standard report. You can generate and view reports in the IBM Information Server Web console and the IBM Information Server console. Both environments contain predefined parameters and templates that you use to generate a report.

  We describe the various reports than can be generated in detail in 1.14, “Reports” on page 394.

These functions have an execution dependency among themselves as shown in Figure 1-7.

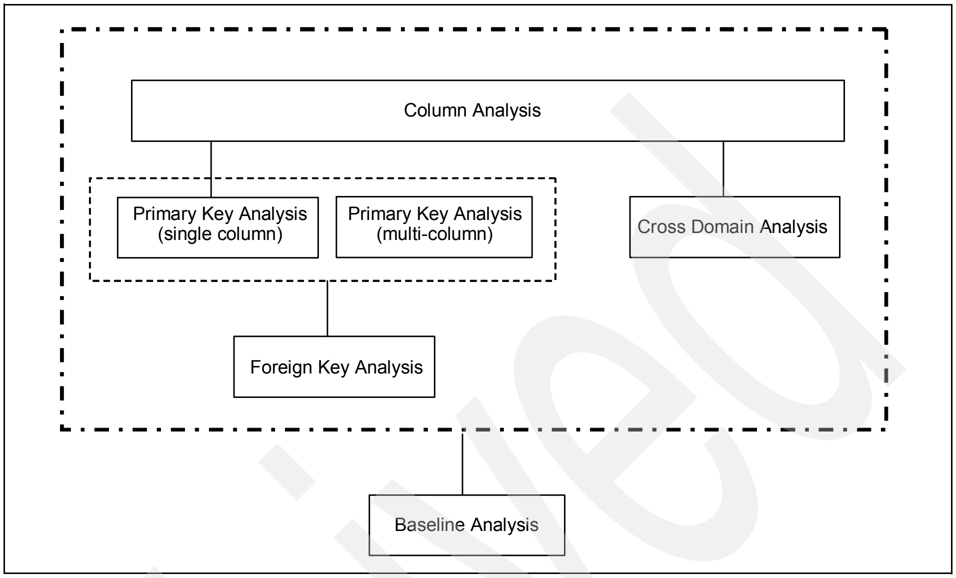


Figure 1-7 IBM WebSphere Information Analyzer function dependencies

Figure 1-7 shows the following:

- Baseline Analysis requires at least column analysis to be performed before it can be run.
- Column Analysis must be run before a single column Primary Key Analysis can be performed.
- A multi-column Primary Key Analysis can be performed independently of any of the other analyses. It invokes Column Analysis automatically under the covers if a Column Analysis has not been performed on the selected columns.
- A Foreign Key Analysis (single or multi-column) can only be performed after a Primary Key Analysis (single or multi-column) is performed.
- Cross Domain Analysis requires Column Analysis to have been run.

1.5 Main components

IBM WebSphere Information Analyzer is designed as an N-tier, service-oriented architecture (SOA), and model driven architecture (MDA) based architecture. It is built on top of the set of framework components that comprise the IBM

Information Server product suite and makes use of the variety of components within that framework.

The N-tier architecture of WebSphere Information Analyzer allows the separation of product tiers. Each of these tiers can exist on the same system or on entirely different systems. The various tiers are the User Interface (UI) tier, Server tier, Repository tier, and Engine tier. These are elaborated on briefly later in this section.

Figure 1-8 provides a high-level overview of the various layers and components within the Information Services Framework (ISF) framework as it specifically relates to the IBM WebSphere Information Analyzer product.

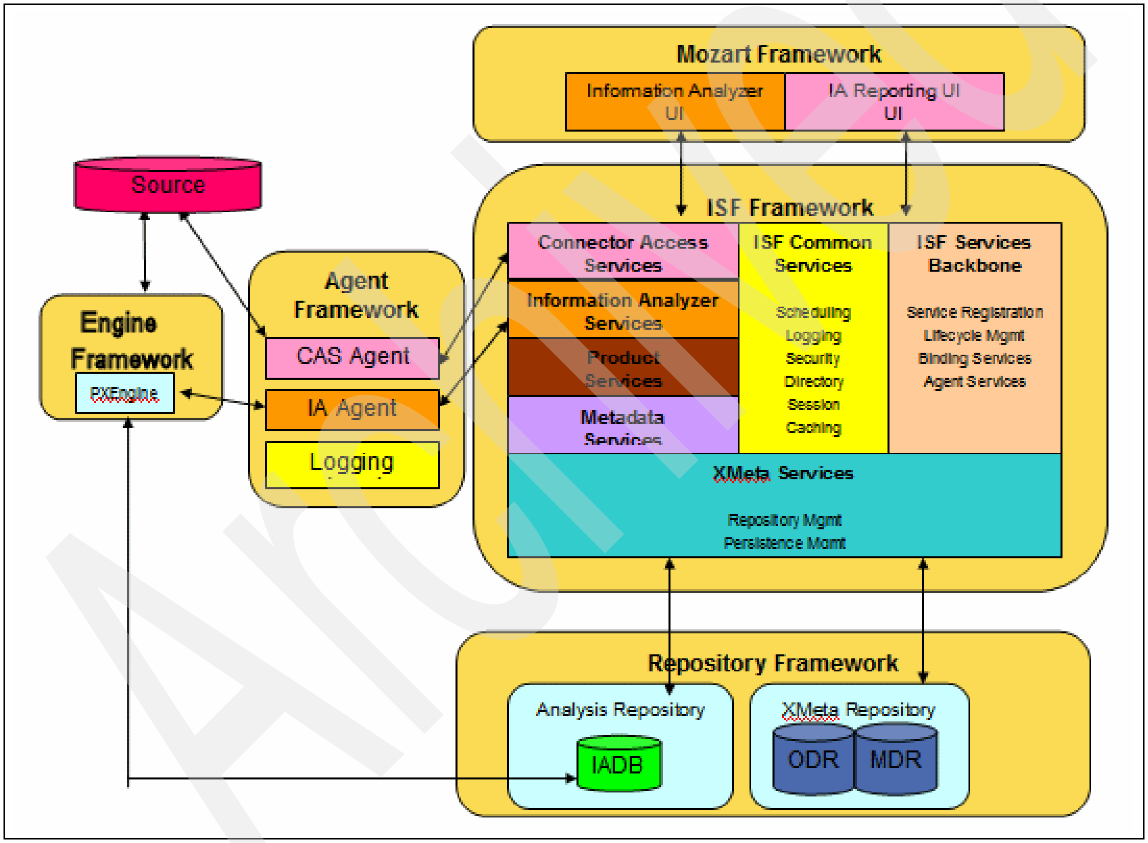


Figure 1-8 IBM WebSphere Information Analyzer main components

Here we describe the key components in Figure 1-8 (the Repository Framework, ISF Framework, Mozart Framework, and Agent Framework):

- **Repository Framework tier**

  This tier is used for persistent information storage. The WebSphere Information Analyzer architecture is MDA based around the usage of the Eclipse Modeling Framework (EMF) for persistence related activities. All persisted information is formally modeled using the XMeta framework for model definition. IBM WebSphere Information Analyzer has a complete and formal model defined that extends the ISF Common Model, providing approximately 120+ independent classes to store all information related to the IBM WebSphere Information Analyzer product set.

  This tier includes:

  - XMeta tier, where the XMeta metadata repository resides. This repository stores imported metadata, project configurations, reports, and results for all components of IBM Information Server.
  - IADB tier, where the IADB analysis repository resides. This repository contains all of the detailed frequency distributions, data samples, reference tables, and cross-domain data associations.

- **ISF Framework tier**

  This tier is used for all services used within ISF. This tier includes the SOA Application Tier where the IBM WebSphere Information Analyzer and ISF resides.

  The SOA-based architecture is based largely around the deployment of particular *services* that reside inside of the ISF application space and provide the set of cross-product and cross-purpose services used by the IBM WebSphere Information Analyzer product. The majority of the primary business logic resides within these services. Additionally, IBM WebSphere Information Analyzer makes heavy use of the range of ISF Common Services, such as Logging, Reporting, Authentication, Security, and Scheduling services.

  IBM WebSphere Information Analyzer specific services deployed within the ISF tier include:

  - Analysis Suite Service provides suite-level services, such as installation/configuration support, as well as some general validation capabilities.
  - Registration Service provides those services related to the registration of data sources within a IBM WebSphere Information Analyzer project.
  - Authoring Service provides functionality related to the creation of the analysis jobs, and communication with the Information Analyzer (IA) Handler.

  - Authentication Service provides IBM Information Server wide user validation.
  - Analysis Summary Service provides functionality related to all pre- and post-processing of analysis functionality.
  - Scheduler Service provides the command and control functionality for IBM WebSphere Information Analyzer scheduled activities.
  - Table Management Service provides capabilities around table management in IADB.
  - Reporting Service provides an implementation for all IBM WebSphere Information Analyzer Reports
  - Column Analysis Service provides functionality related to the Column-level analysis.

- **Mozart Framework tier**

  This tier is used for all client-level activities. This tier includes:

  - Rich Client Tier, where the Primary UI interface resides.
  - Reporting Tier, where the Primary Reporting UI resides.

- **Agent Framework tier**

  This tier is used for communication with the DataStage PX Engine and connector frameworks. This tier includes:

  - Analysis Engine tier, where the parallel execution (PX) Engine used for certain IBM WebSphere Information Analyzer analysis functionality resides.
  - SQL Engine tier, where the SQL Engine used for certain IBM WebSphere Information Analyzer analysis functionality resides.

Figure 1-9 describes the IBM WebSphere Information Analyzer execution flow.

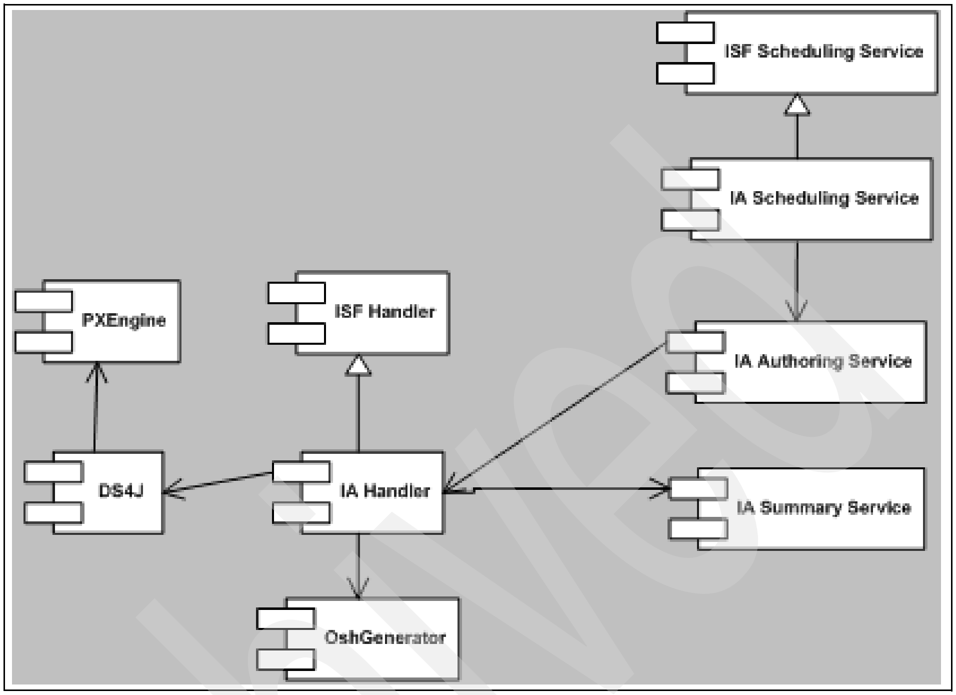


Figure 1-9 IBM WebSphere Information Analyzer execution flow

Figure 1-9 describes the IBM WebSphere Information Analyzer execution flow as follows:

- The IBM WebSphere Information Analyzer Scheduling Service is an implementation of the ISF Scheduling Service with the required interfaces. The IBM WebSphere Information Analyzer Scheduling Service invokes the Authoring Service to coordinate any of the various IBM WebSphere Information Analyzer analysis activities.
- The Authoring Service builds the IBM WebSphere Information Analyzer job to perform the requested analysis. It then communicates with the IBM WebSphere Information Analyzer Handler to provide to it the IBM WebSphere Information Analyzer job and all the parameters required by the execution flow template.[4]

[4] Parameters such as the isolation level when accessing relational data, whether auto commit should be turned on or not, whether the generated scripts should be retained or not, and whether the most recent column analysis results should completely replace a previous column analysis result, or just update relevant portions of the previous result.

- The IBM WebSphere Information Analyzer Handler is an implementation of the ISF Handler/Agent framework. It implements the interfaces required to participate in that pattern. Its primary responsibility it so receive messages from the Authoring Service, establish the connection to the to DataStage instance, invoke the OSHGenerator for the creation of the appropriate analysis flow, and provide command and control against that analysis flow.[5]

  > **Note:** The OSHGenerator is invoked for the column analysis, multi-column primary key analysis, and data sampling because these tasks require an IBM WebSphere DataStage job to be run. The other analyses, such as foreign key analysis, cross domain analysis, and baseline analysis, take advantage of the column analysis data that is stored in the IADB and do not require an IBM WebSphere DataStage job to be run.

- The IBM WebSphere Information Analyzer Handler uses the proprietary Java™ API (DS4J) into the DataStage Engine database (previously known as *Universe*). The DataStage job executes as a *Run Now* job initiated by the IBM WebSphere Information Analyzer Handler. In other words, the ISF Scheduling Service is invoked by IBM WebSphere Information Analyzer (either as *Run Now* or scheduled) for the analysis task, which is then executed as a DataStage *Run Now* job initiated by the IA Handler through the DS4J API. The progress of the execution of this job can be seen in the DataStage QualityStage Director, as well as the Activity Status pane as shown in Figure 1-94 on page 144.

  > **Note:** Events from the DataStage job are available through the Log View screens that present the events captured by the Logging Service. The Log View is the primary vehicle you should use to view job status rather than the DataStage QualityStage Director.

1.6 Setting up your system

After successfully installing IBM Information Server (which includes the IBM WebSphere Information Analyzer module), you need to set up the system before you can begin analyzing the data. Figure 1-10 shows the recommended steps for setting up the system.

[5] The Orchestrate Shell Generator (OSH) Generator is responsible for taking the definition of the analysis flow to be executed (with all of the parameters used to fill in the corresponding execution flow template) and create the appropriate OSH script, which is then laid down within the PXEngine execution environment, and executed as part of the IA Handler command and control structure. In other words, it puts a script wrapper around the IBM WebSphere Information Analyzer job created by the Authoring Service so that it can be delivered to DataStage as a DataStage parallel job.

You can also begin by creating the project and then configuring the data source connections and so forth, as documented in *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.

> **Attention:** In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

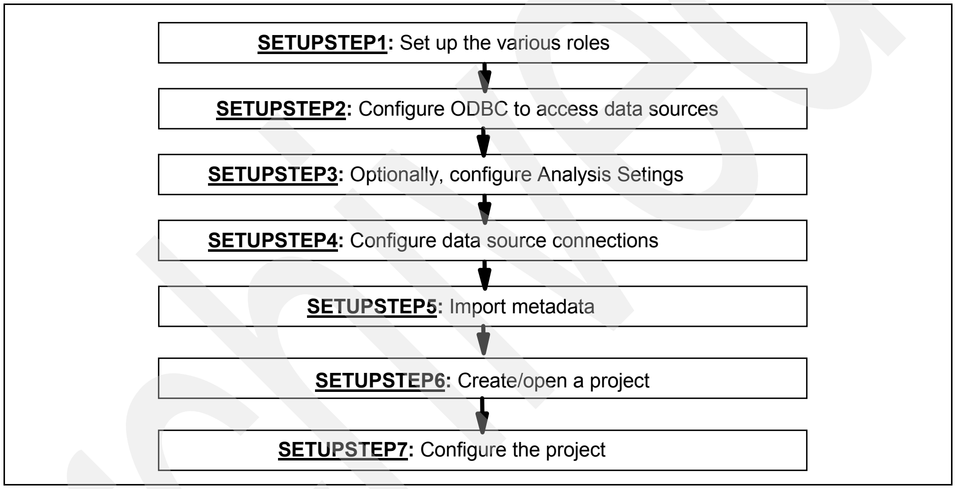

SETUPSTEP1: Set up the various roles
SETUPSTEP2: Configure ODBC to access data sources
SETUPSTEP3: Optionally, configure Analysis Setings
SETUPSTEP4: Configure data source connections
SETUPSTEP5: Import metadata
SETUPSTEP6: Create/open a project
SETUPSTEP7: Configure the project

Figure 1-10 Steps in setting up the IBM WebSphere Information Analyzer system

The steps can be summarized as follows:

- First, the various roles of your organization's staff are defined (SETUPSTEP1).
- Next, a database administrator, system administrator, or other data analyst configures ODBC to access the data sources to be analyzed (SETUPSTEP2).

- The system-wide Analysis settings can then be modified if required (SETUPSTEP3).
- The administrator[6] then configures data source connections (SETUPSTEP4), imports relevant metadata from the data stores into the metadata repository (SETUPSTEP5), creates a project (SETUPSTEP6), and then associates metadata with the project and configures security options for the project (SETUPSTEP7).

After the project is configured, you can run analysis tasks. When an analysis completes, you can review the results and inferences and then make decisions based on the evaluation. You can choose to have your review and inference decisions delivered to other authorized users within the user interface. For example, you can create a report that displays a summary of analysis results or you can share your results with other suite components. The results are saved in the metadata repository (IADB database) and are available to all authorized suite users.

[6] The *administrator* is the Information Analyzer Data Administrator or Information Analyzer Project Administrator. We describe the roles involved in the process later.

We describe these steps in further detail in this section. For our examples, we created a set of seven tables in a database called *IASAMPLE* on an AIX server (jamaica.itsosj.sanjose.ibm.com) that was modelled along the lines of the sample database that is provided with DB2 9 for LUW. The data model of these tables is shown in Figure 1-11. A schema name of *IA* was used. We designed the columns of these tables, the inter-relationships between these tables, and the content of these tables to showcase the main functions of IBM WebSphere Information Analyzer.

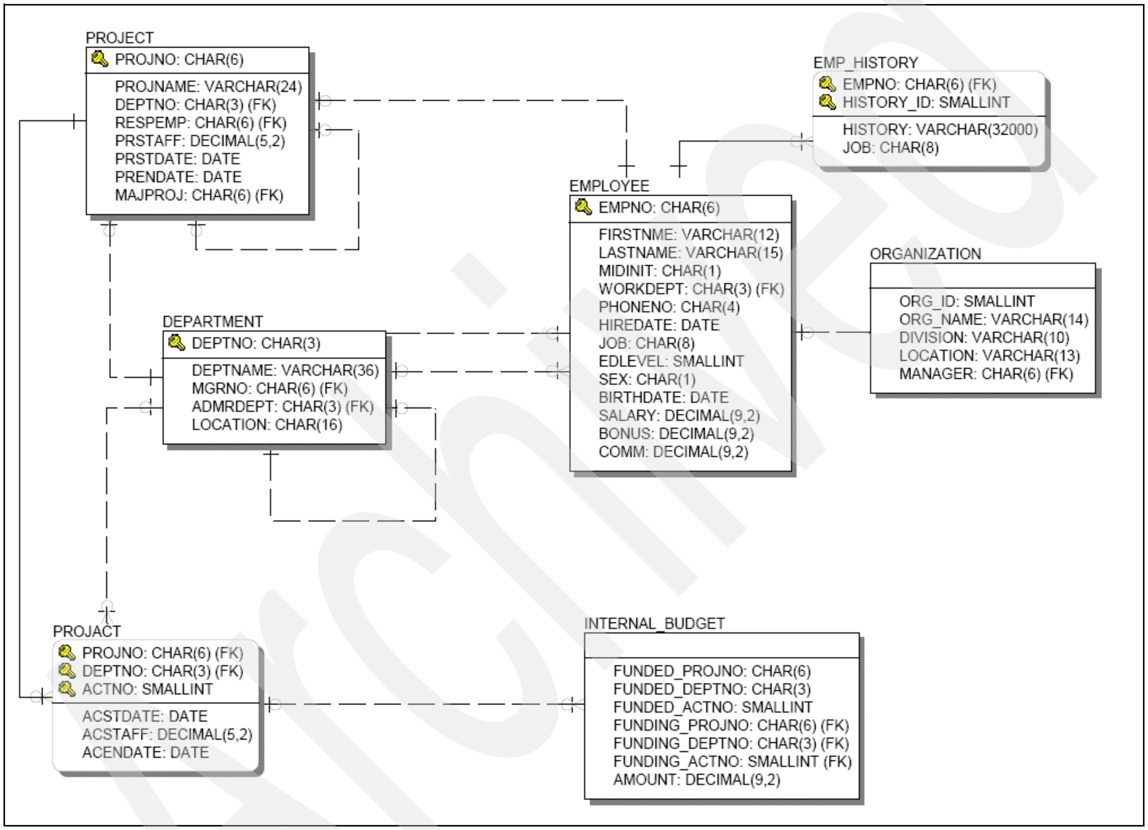


Figure 1-11 Data model of tables in IASAMPLE database with schema name of IA

1.6.1 SETUPSTEP1: Set up the various roles

To create a secure project environment, you can define a security policy that is based on user authentication and role identification. Users derive authority from the union of their individual and group roles. Roles are complimentary in that you can grant users greater responsibility by assigning multiple roles.

> **Attention:** In the examples used in this chapter, for the purposes of expediency, we used a single user that was assigned every individual suite role, suite component role, and project role. In a real-world environment, you would implement an appropriate role-based security infrastructure.

The security roles that IBM Information Server supports and the security roles that are required to perform information analysis tasks are as follows:

- Suite roles[7]

  These roles are:

  - Suite Administrator role provides maximum administration privileges throughout the suite.
  - Suite User role provides access to the suite and to suite components. If you assign a user the role of suite administrator and not that of suite user, the user cannot log on to any of the suite tools or component tools.

    This is the default role.

  Suite roles are created from the IBM Information Server Web Console under the Administration tab, and must be performed by a person who has Suite Administration privileges.

- Suite component roles

  The IBM WebSphere Information Analyzer suite component roles are:

  - Information Analyzer Data Administrator role can import metadata, modify analysis settings, and add or modify system sources.
  - Information Analyzer Project Administrator role can administer projects including creating, deleting, and modifying information analysis projects.
  - Information Analyzer User role can log on to IBM WebSphere Information Analyzer, view the dashboard, and open a project.

[7] The suite includes Information Analyzer, DataStage, QualityStage, FederationServer, and WebSphere Information Services Director.

Table 1-3 shows the suite component role that is required to perform information analysis tasks. The Information Analyzer User role is the basic suite component role that you assign to a user. You can also assign the Information Analyzer Project Administrator role or Information Analyzer Data Administrator role to grant the user additional rights.

Table 1-3 Suite component roles eligible to perform project and data tasks

Task	**Suite component roles**		
	Project Administrator	**Data Administrator**	**User**
My Home and Project Dashboard			
Configure My Home workspace	Yes	Yes	Yes
Configure reports to display on My Home workspace	Yes	Yes	Yes
View the Dashboard workspace	Yes	Yes	Yes
Modify the Project Dashboard	Yes	Yes	Yes
Projects			
Open a project	Yes	Yes	Yes
Create a project	Yes		
Delete a project	Yes		
Modify project properties	Yes		
Metadata			
Import metadata		Yes	
Apply metadata to schemas, tables, or columns	Yes	Yes	Yes
Add sources		Yes	

Task	Suite component roles		
Administer			
Modify analysis settings		Yes	
Export a table from table management		Yes	
Delete a table from table management		Yes	
Create or view notes	Yes	Yes	Yes

- Project roles

  The IBM WebSphere Information Analyzer project roles are:

  – Information Analyzer Business Analyst role reviews analysis results. This role can set baselines and checkpoints for baseline analysis, publish analysis results, delete analysis results, and view the results of analysis jobs.

  – Information Analyzer Data Operator role manages data analyses and logs. This role can run or schedule all analysis jobs.

  – Information Analyzer Data Steward role provides read-only views of analysis results. This role can also view the results of all analysis jobs.

  Table 1-4 shows the project roles (in addition to the suite component roles) that are required to perform various tasks. These roles specify the level of access that a user has within a specific project. For example, the Data Steward and Data Operator project roles are added to the suite component roles to either grant or restrict access rights. An Information Analyzer User might be granted Business Analyst and Data Operator project roles so that he could view analysis results and run analysis jobs.

Table 1-4 Project roles required to run and review analysis

Task	Project roles		
	Business Analyst	**Data Steward**	**Data Operator**
Column Analysis			
View Column Analysis workspace	Yes	Can only read results	Yes
Open column analysis results	Yes	Can only read results	
Run or schedule a column analysis job			Yes

Task	Project roles		
Primary key analysis			
View Primary Key Analysis workspace	Yes	Can only read results	Yes
Open primary key analysis results	Yes	Can only read results	
Run or schedule a primary key analysis job			Yes
Cross-domain analysis			
View Cross-Domain Analysis workspace	Yes	Can only read results	Yes
Open cross-domain analysis results	Yes	Can only read results	
Run or schedule a cross-domain analysis job			Yes
Foreign Key analysis			
View Foreign Key Analysis workspace	Yes	Can only read results	
Open foreign key analysis results	Yes	Can only read results	
Run or schedule a foreign key analysis job			Yes
Baseline analysis			
View Baseline Analysis workspace and open analysis results	Yes	Can only read results	
Set the checkpoint or baseline	Yes		
Analysis results publication			
View Publish Analysis Results workspace and open analysis results	Yes	Can only read results	
Publish analysis results	Yes		
Delete analysis results	Yes		

Task	**Project roles**		
Administer			
View Table Management workspace	Can only read results	Can only read results	
Set the checkpoint or baseline	Yes		

We describe the procedure for assigning user and groups to a project and assigning roles to these users and groups in “Assigning users/groups to a project and assigning roles” on page 86.

1.6.2 SETUPSTEP2: Configure ODBC to access data sources

IBM WebSphere Information Analyzer supports two approaches for connecting with the data sources to be analyzed using an ODBCConnector. In this section, we discuss only how to configure ODBC access to the data sources to be analyzed.

After the various roles have been designated and the suite roles and suite component roles have been assigned, you must create a data source name (DSN) for each source database or file on which you want to perform analysis. You must also create a DSN for the analysis database IADB. You must create the DSN for the source database on the machine that is running IBM WebSphere Metadata Server. In addition, you must create the DSN to the analysis database on the computer on which the parallel processing engine is installed.

To configure ODBC to access data sources that needs to be analyzed, you need to edit three files (dsenv, .odbc.ini, and uvodbc.config) that are located in the IBM WebSphere Information Analyzer installation directory, to set up the required ODBC connections to the data sources to be accessed by IBM WebSphere Information Analyzer.

Note: Non-wire drivers require different setup information than wire drivers. *Non-wire* drivers require information about the location of the database client software. *Wire* drivers require information about the database itself. For information about configuring the ODBC environment for your specific database, see the Data Direct Drivers Reference manual odbcref.pdf file located in the following directory:

/opt/IBM/InformationServer/Server/DSEngine[a]/Server/branded_odbc/books/odbc

You should also check the ODBCREAD.ME file in the branded_odbc directory.

[a] This is the default installation directory on Linux and AIX. The Windows default installation directory is c:\IBM\InformationServer\Server\DSEngine.

Attention: We configured ODBC on the Linux platform where we installed IBM Information Server. Configuration on Windows is quite different from Linux, and we do not describe that configuration here. For details about configuring ODBC on a Windows platform, refer to *IBM Information Server Version 8.0.1 IBM Information Server Planning, Installation, and Configuration Guide,* GC19-1048.

Here is the information that you need to edit in each of the three files:

- **dsenv:**

  This file stores environment variables. You might need to add new environment variables as you configure IBM WebSphere Information Analyzer to connect to different databases using plug-ins or ODBC drivers. You must add any environment variables that you need for interactive use of ODBC drivers to make a connection to an ODBC data source to the dsenv file. This lets the IBM Information Server Engine Framework (including the resident DataStage server) inherit the proper environment for ODBC connections.

  For a connection using a wire protocol driver (database), there are no changes required to dsenv. Because we configured ODBC access to z/OS data sources, we modified the dsenv file as described in B.2, “Configure ODBC data sources on the z/OS platform” on page 554 to add environment variables.

Example 1-1 shows the dsenv file, with the modifications shown in bold font.

Example 1-1 Modified dsenv file

```
# PLATFORM SPECIFIC SECTION

set +u

if [ -z "$DSHOME" ] && [ -f "/.dshome" ]
then
   DSHOME=`cat /.dshome`
   export DSHOME
fi

if [ -z "$DSHOME" ]
then
DSHOME=/opt/IBM/InformationServer/Server/DSEngine; export DSHOME
fi

if [ -z "$APT_ORCHHOME" ]
then
APT_ORCHHOME=/opt/IBM/InformationServer/Server/PXEngine; export
APT_ORCHHOME
fi

#if [ -z "$UDTHOME" ]
#then
UDTHOME=/opt/IBM/InformationServer/Server/DSEngine/ud41 ; export
UDTHOME
UDTBIN=/opt/IBM/InformationServer/Server/DSEngine/ud41/bin ; export
UDTBIN
#fi

#if [ -z "$ASBHOME" ] && [ -f "$DSHOME/.asbnode" ]
#then
   ASBHOME=`cat $DSHOME/.asbnode`
   export ASBHOME
#fi

#if [ -z "$ASBHOME" ]
#then
   #ASBHOME=`dirname \`dirname $DSHOME\``/ASBNode
   #export ASBHOME
#fi
```

```
if [ -n "$DSHOME" ] && [ -d "$DSHOME" ]
then
   ODBCINI=$DSHOME/.odbc.ini; export ODBCINI
   HOME=${HOME:-/}; export HOME

   #LANG="<langdef>";export LANG
   #LC_ALL="<langdef>";export LC_ALL
   #LC_CTYPE="<langdef>";export LC_CTYPE
   #LC_COLLATE="<langdef>";export LC_COLLATE
   #LC_MONETARY="<langdef>";export LC_MONETARY
   #LC_NUMERIC="<langdef>";export LC_NUMERIC
   #LC_TIME="<langdef>";export LC_TIME
   #LC_MESSAGES="<langdef>"; export LC_MESSAGES

#DB2 Version 9
DB2DIR=/opt/IBM/InformationServer/DB2;export DB2DIR
DB2INSTANCE=db2inst1; export DB2INSTANCE
INSTHOME=/export/home/db2inst1; export INSTHOME
PATH=$PATH:$INSTHOME/sqllib/bin:$INSTHOME/sqllib/adm:$INSTHOME/sqllib/m
isc
export PATH


   LD_LIBRARY_PATH=/opt/ibm/wsclassic91/cli/lib:`dirname
$DSHOME`/branded_odbc/lib:`dirname $DSHOME`/DSComponents/lib:`dirname
$DSHOME`/DSComponents/bin:$DSHOME/lib:$DSHOME/uvdlls:$ASBHOME/apps/jre/
bin:$ASBHOME/apps/jre/bin/classic:$ASBHOME/lib/cpp:$ASBHOME/apps/proxy/
cpp/linux-all-x86:$LD_LIBRARY_PATH

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTHOME/sqllib/lib;
        export LD_LIBRARY_PATH


CAC_CONFIG=/opt/ibm/wsclassic91/cli/lib/cac.ini
export CAC_CONFIG

THREADS_FLAG=native;export THREADS_FLAG


fi
```

For a connection using a non-wire protocol driver (database client), you generally need to specify the following information in the dsenv file:

- Environment variables that are required by the database client software
- Database home location
- Database library directory
- PATH environment variable

The library path environment variables are LIBPATH (AIX) and LD_LIBRARY_PATH (Linux).

- **.odbc.ini:**

  This file provides information about connecting to the database (wire protocol drivers) or the database client (non-wire protocol drivers). If your system uses a mix of drivers, your .odbc.ini file contains a mix of entry types.

  > **Note:** Configuration examples for the various platforms are provided in the $DSHome/branded_odbc/IBM_Tools directory in the odbc.ini file.

  Example 1-2 shows the contents of the .odbc.ini file that we used for this book. It includes the data source IASAMPLE that we used in the examples, and the data sources (redbank, DB8a, and CACSAMP) that we used in the business scenario we describe in Chapter 2, “Financial services business scenario” on page 417.

  In Example 1-2:

  - The [ODBC Data Sources] section lists the DSN names (IADB, CACSAMP, NORTHERN_CALIFORNIA_BANK, OVERVIEW, and NABANK) and associates them with the name of the driver.
  - Following this section is the Data Source Specification section (not explicitly marked), which must have one data source specification for each DSN that is specified.

    For example, DSN IADB has data source Database IADB, DSN CACSAMP has data source Database CACSAMP, while DSN OVERVIEW has data source Database IASAMPLE.
  - Finally, the section [ODBC] lists ODBC tracing options and specifies where the ODBC drivers are installed.

Example 1-2   Contents of the modified .odbc.ini file

```
[ODBC Data Sources]
IADB=DataDirect DB2 Wire Protocol Driver
CACSAMP=WebSphere Classic Federation Client
NORTHERN_CALIFORNIA_BANK=DataDirect DB2 Wire Protocol Driver
OVERVIEW=DataDirect DB2 Wire Protocol Driver
```

```
NABANK=DataDirect DB2 Wire Protocol Driver

[IADB]
Driver=/opt/IBM/InformationServer/Server/branded_odbc/lib/VMdb222.so
Description=DataDirect DB2 Wire Protocol Driver
AddStringToCreateTable=
AlternateID=
Connection=
Database=IADB
DynamicSections=100
GrantAuthid=PUBLIC
GrantExecute=1
IpAddress=9.43.86.77
IsolationLevel=CURSOR_STABILITY
Location=
LogonID=db2inst1
Password=itso13sj
Package=
PackageOwner=
TcpPort=50001
WithHold=1

[CACSAMP]
Driver=/opt/ibm/wsclassic91/cli/lib/libcacsqlcli.so
Database=CACSAMP

[NORTHERN_CALIFORNIA_BANK]
Driver=/opt/IBM/InformationServer/Server/branded_odbc/lib/VMdb222.so
Description=DataDirect DB2 Wire Protocol Driver
AddStringToCreateTable=
AlternateID=
Connection=
Database=redbank
DynamicSections=100
GrantAuthid=PUBLIC
GrantExecute=1
IpAddress=9.43.86.55
IsolationLevel=CURSOR_STABILITY
Location=
LogonID=db2inst1
Password=itso13sj
Package=
PackageOwner=
TcpPort=50000
WithHold=1
```

```
[OVERVIEW]
Driver=/opt/IBM/InformationServer/Server/branded_odbc/lib/VMdb222.so
Description=DataDirect DB2 Wire Protocol Driver
AddStringToCreateTable=
AlternateID=
Connection=
Database=IASAMPLE
DynamicSections=100
GrantAuthid=PUBLIC
GrantExecute=1
IpAddress=Jamaica.itsosj.sanjose.ibm.com
IsolationLevel=CURSOR_STABILITY
Location=
LogonID=db2inst1
Password=itso13sj
Package=
PackageOwner=
TcpPort=50000
WithHold=1

[NABANK]
Driver=/opt/IBM/InformationServer/Server/branded_odbc/lib/VMdb222.so
Description=DataDirect DB2 Wire Protocol Driver
AddStringToCreateTable=
AlternateID=
Connection=
Database=DB8A
DynamicSections=100
GrantAuthid=PUBLIC
GrantExecute=1
IpAddress=9.12.4.10
IsolationLevel=CURSOR_STABILITY
Location=
LogonID=nalur1
Password=itso13sj
Package=
PackageOwner=
TcpPort=38100
WithHold=1

[ODBC]
IANAAppCodePage=4
InstallDir=/opt/IBM/InformationServer/Server/branded_odbc
Trace=0
```

```
TraceDll=/opt/IBM/InformationServer/Server/branded_odbc/lib/odbctrac.so
TraceFile=odbctrace.out
UseCursorLib=0
```

- **uvodbc.config:**

  This file is used to specify the data source name (DSN) for each database that you are connecting to through ODBC.

  Example 1-3 shows the contents of the uvodbc.config file that we used in this book. It includes the DSN (OVERVIEW) that corresponds to the IASAMPLE database. It also includes the DSN (IADB) that corresponds to the IBM WebSphere Information Analyzer analysis database IADB, and other DSNs (CACSAMP, NORTHERN_CALIFORNIA_BANK and NABANK) that we used in the business scenario we describe in Chapter 2, “Financial services business scenario” on page 417.

  > **Note:** You can also place a subset copy of the uvodbc.config file in each project directory. For example, the default path for projects on Linux is /opt/IBM/WDIS/Server/Projects/. This copy is useful where you configure a data source that is known to some projects but not to others. This copy is achieved by placing a subset of the data sources that are defined in the home directory in the uvodbc.config file copy in the project directory. This action filters the data sources that are accessible by that project to those defined in the project directory configuration file. By default, the current project directory is searched for a uvodbc.config file and, if it finds one, uses this in preference to the file in $DSHOME. If you alter uvodbc.config in the home directory after creating projects, you should copy the relevant portions of the edited file to the appropriate project directories.

Example 1-3 Contents of the modified uvodbc.config file

```
[ODBC DATA SOURCES]

<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = 127.0.0.1
<IADB>
DBMSTYPE = ODBC
<CACSAMP>
DBMSTYPE = ODBC
<NORTHERN_CALIFORNIA_BANK>
DBMSTYPE = ODBC
<OVERVIEW>
DBMSTYPE = ODBC
```

```
<NABANK>
DBMSTYPE = ODBC
```

Note: Because we only used ODBC, we did not have to bind DB2 packages to a DSN as suggested in *IBM Information Server Version 8.0.1 IBM Information Server Planning, Installation, and Configuration Guide*, GC19-1048.

When you have configured ODBC to access the data sources of interest, you can proceed to configuring the Analysis Settings as described in 1.6.3, “SETUPSTEP3: Optionally, configure Analysis Settings” on page 44.

1.6.3 SETUPSTEP3: Optionally, configure Analysis Settings

Optionally, you can choose to modify the system-wide Analysis Settings if you feel that the defaults are not appropriate for your organization. Analysis threshold settings specify how the system performs analysis.

You can specify default system-wide settings to be inherited by the individual projects when a project is created. However, you can override these settings for an individual project as described in “Modify the project’s and data source’s analysis settings if required” on page 77.

Analysis settings apply to column analysis (Table 1-5 on page 45), table analysis (Table 1-6 on page 46), and cross-table analysis (Table 1-7 on page 46). Each threshold option is a percentage that controls the fraction of values that are required to meet this threshold before making the inference.

A *threshold setting* is a percentage that controls the total number of data values that are required to meet the threshold before an inference is made by the system. For example, if the threshold setting ConstantThreshold equals 99%, then almost all of the values in a column must be the same for that column to be inferred as a constant. If 100% of the values in a column are the same, then the column will be inferred as a constant. If 99.1% of the values in the column are the same, the column will also be inferred as a constant. However, in a table of 1 million rows, this indicates that there are 9000 non-constant values. Which means that there is a possibility that the inference might be wrong. When you review the results in the frequency distribution of a column, you can examine the values and determine whether 0.9% of the values are inconsistent or invalid. You can then change the inference in the frequency distribution if you choose.

Figure 1-12 on page 47 through Figure 1-15 on page 49 show the setting of system-wide Analysis Settings as follows:

1. On the Home navigator menu in the console, select **Configuration** → **Analysis Settings** as shown in Figure 1-12 on page 47**.**

2. In the Analysis Settings workspace in Figure 1-13 on page 47, the Analysis Database tab shows the configuration details such as the Database Name (IADB) and the Analysis ODBC DSN (IADB[8]). This information was provided during IBM WebSphere Information Analyzer installation.
3. In the Analysis Settings workspace in Figure 1-14 on page 48, the Analysis Engine tab shows the configuration details of the IBM WebSphere DataStage analysis engine. You need to provide this information prior to performing analysis in order for IBM WebSphere Information Analyzer jobs to run successfully.
4. In the Analysis Settings workspace in Figure 1-14 on page 48, the Analysis Settings tab shows the column analysis, table analysis, and cross-table settings that can be modified. You can choose to modify these settings and then click **Save All** to save these changes.

You can now proceed to configuring data source connections as described in 1.6.4, "SETUPSTEP4: Configure data source connections" on page 49.

Table 1-5 Column analysis threshold settings

Setting	Description
Nullability threshold	Infers whether a column allows null values. If a column has null values with a frequency percentage equal to or greater than the nullability threshold, the system determines that the column allows null values. If null values do not exist in the column or the frequency percent is less than threshold, the system determines that the column does not allow null values. The minimum and maximum percentage that can be specified is 0.01% and 10.0% respectively. The default is 1.0%.
Uniqueness threshold	Infers whether a column is considered unique. If a column has a percentage of unique values equal to or greater than the uniqueness threshold, the system determines that the column is unique. The minimum and maximum percentage that can be specified is 90.0% and 100.0% respectively. The default is 99.0%.
Constant threshold	Infers whether a column contains constant values. If a column has a single distinct value with a frequency percentage equal to or greater than the constant threshold, the system determines that the column is constant. The minimum and maximum percentage that can be specified is 90.0% and 100.0% respectively. The default is 99.0%.

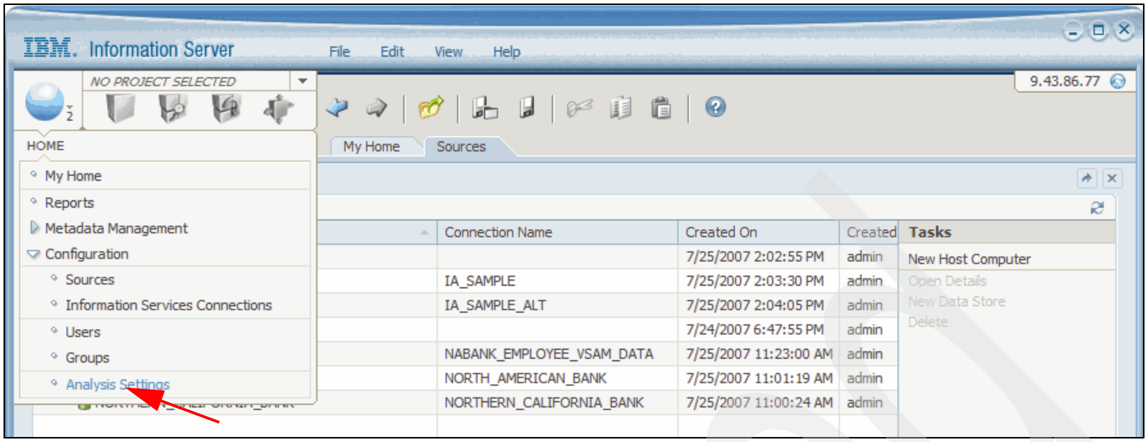
[8] This can be any string, but it must match the string in the .odbc.ini file as described in Example 1-2 on page 40.

Table 1-6 Table analysis settings

Setting	Description
Primary key threshold	Infers whether a column can be considered a primary key candidate. If a column has a cardinality percent equal or greater than the primary key threshold, the system determines that the column is a single column primary key candidate. If a multi-column concatenation has a cardinality percent equal or greater than the primary key threshold, the system determines that the columns are a multi-column primary key candidate. The minimum and maximum percentage that can be specified is 90.0% and 100.0% respectively. The default is 99.5%.
Data sample size	Controls the number of records that are included when a data sample of the table or file is created. The minimum and maximum number that can be specified is 1 and 999,999 respectively. The default is 2,000 records.
Data sample method	Determines which type of method is used to create a data sample: sequential, random, or every nth value. Default is random.
Data sample parameter	Specifies the n value for the n data sampling method. The minimum and maximum number that can be specified is 2 and 100 respectively. The default is 10.
Composite key maximum	Determines the maximum number of columns that can be combined when you search for primary key candidates. The minimum and maximum number that can be specified is 2 and 7 respectively. The default is 2.

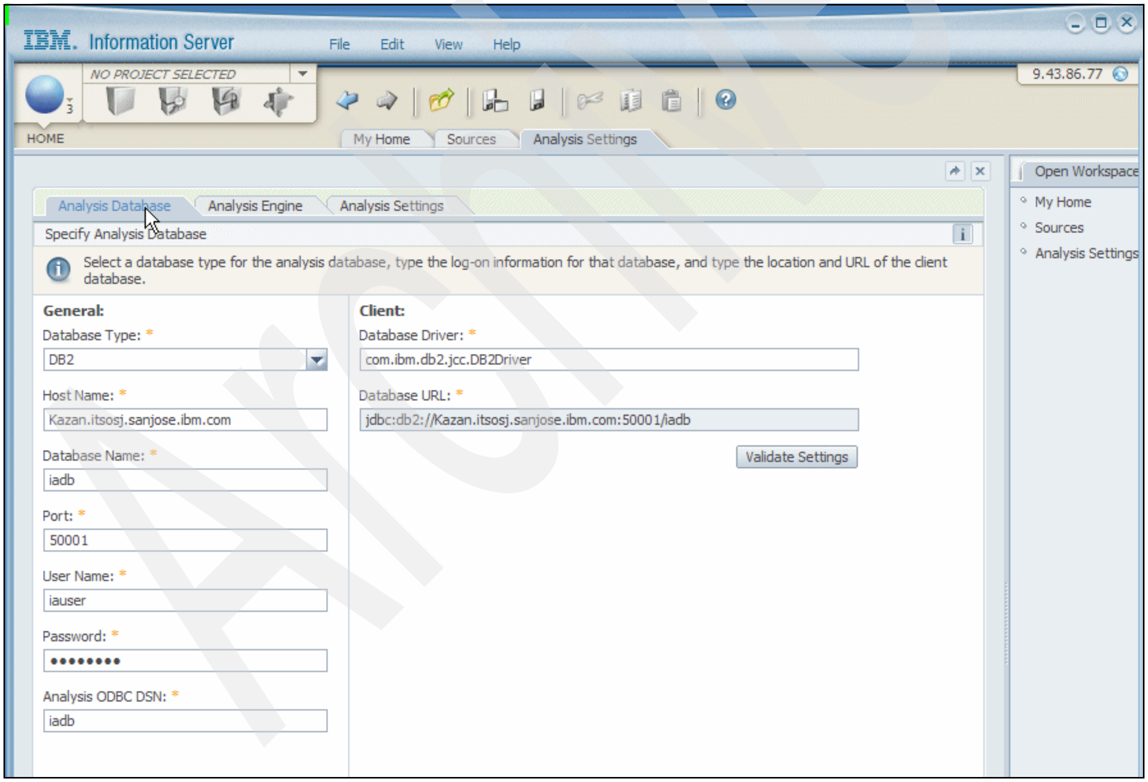
Table 1-7 Cross-table threshold settings

Setting	Description
Common domain threshold	Determines the percentage of distinct values that appear in the frequency distribution of one column that match distinct values in the frequency distribution of another column. If the percentage of matching distinct values is equal to or greater than the threshold, then the two columns are inferred to have a common domain. The minimum and maximum percentage that can be specified is 90.0% and 100.0% respectively. The default is 98%.



Connection Name	Created On	Created
	7/25/2007 2:02:55 PM	admin
IA_SAMPLE	7/25/2007 2:03:30 PM	admin
IA_SAMPLE_ALT	7/25/2007 2:04:05 PM	admin
	7/24/2007 6:47:55 PM	admin
NABANK_EMPLOYEE_VSAM_DATA	7/25/2007 11:23:00 AM	admin
NORTH_AMERICAN_BANK	7/25/2007 11:01:19 AM	admin
NORTHERN_CALIFORNIA_BANK	7/25/2007 11:00:24 AM	admin

Figure 1-12 Modify system-wide Analysis Settings 1/4



Specify Analysis Database

Select a database type for the analysis database, type the log-on information for that database, and type the location and URL of the client database.

General:

Database Type: * DB2

Host Name: * Kazan.itsosj.sanjose.ibm.com

Database Name: * iadb

Port: * 50001

User Name: * iauser

Password: * ••••••••

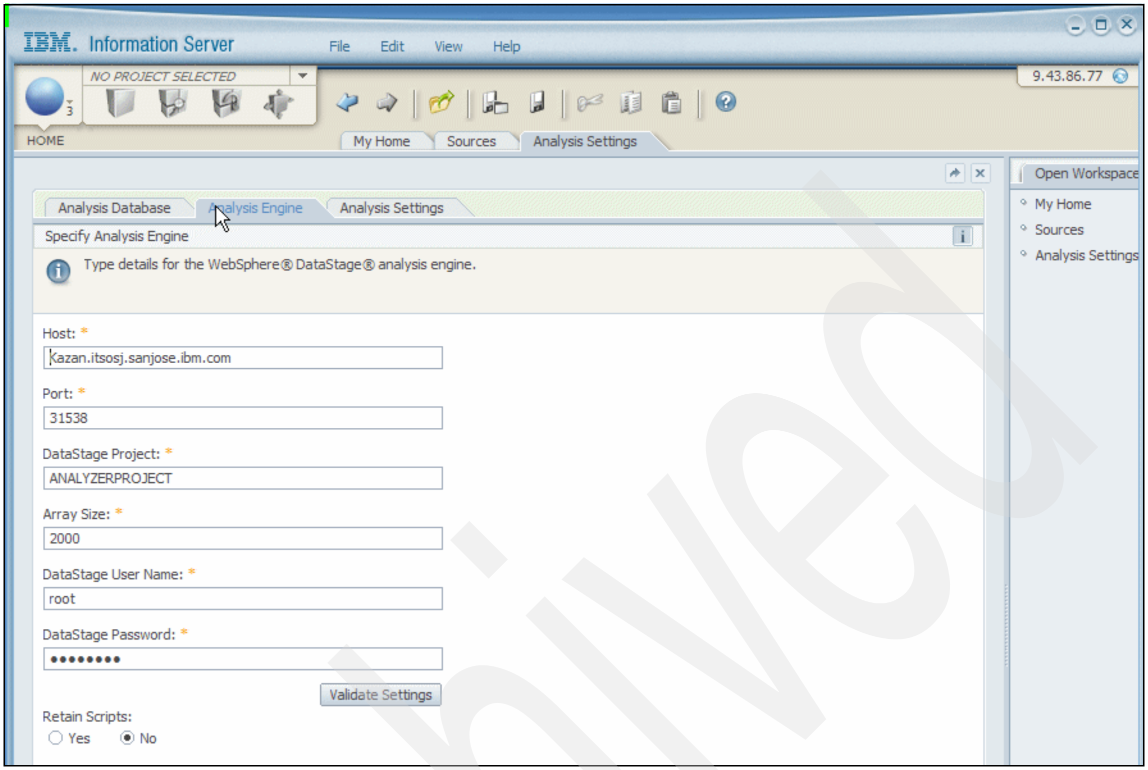
Analysis ODBC DSN: * iadb

Client:

Database Driver: * com.ibm.db2.jcc.DB2Driver

Database URL: * jdbc:db2://Kazan.itsosj.sanjose.ibm.com:50001/iadb

Figure 1-13 Modify system-wide Analysis Settings 2/4



Specify Analysis Engine

Type details for the WebSphere® DataStage® analysis engine.

Host: * kazan.itsosj.sanjose.ibm.com

Port: * 31538

DataStage Project: * ANALYZERPROJECT

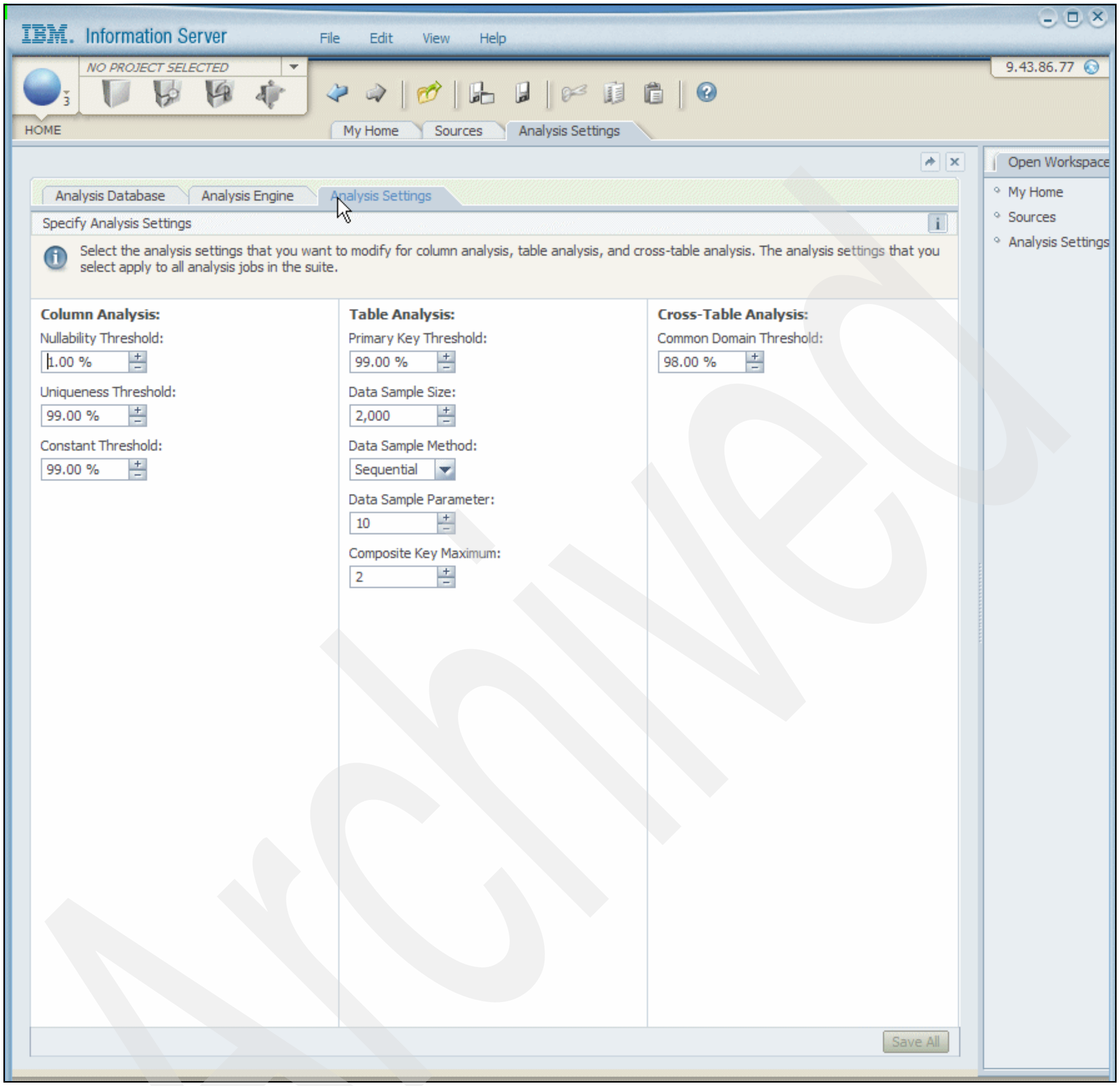
Array Size: * 2000

DataStage User Name: * root

DataStage Password: * ••••••••

Retain Scripts:
○ Yes ● No

Figure 1-14 Modify system-wide Analysis Settings 3/4



Specify Analysis Settings

Select the analysis settings that you want to modify for column analysis, table analysis, and cross-table analysis. The analysis settings that you select apply to all analysis jobs in the suite.

Column Analysis:

Nullability Threshold: 1.00 %

Uniqueness Threshold: 99.00 %

Constant Threshold: 99.00 %

Table Analysis:

Primary Key Threshold: 99.00 %

Data Sample Size: 2,000

Data Sample Method: Sequential

Data Sample Parameter: 10

Composite Key Maximum: 2

Cross-Table Analysis:

Common Domain Threshold: 98.00 %

Figure 1-15 Modify system-wide Analysis Settings 4/4

1.6.4 SETUPSTEP4: Configure data source connections

The metadata repository stores key metadata about objects such as data schemas, tables, and columns that you want to analyze.

To ensure that two distinct objects (such as two tables named *CUSTOMER* located in two distinct systems) are not inappropriately linked or overlaid, you must create the high-level and logical hierarchy of objects (called a *source object*) to provide an inventory of assets that you can use to access your data.[9] This includes identifying the data stores (which contain the data sources to analyze) by defining the location (host computer) of the data stores and the access methods to these data stores. For example, you define the host computer on which the data source is stored, identify and configure the data store that contains the data source, and set up a connection to that data store.

Note: A *data source* belongs to a single data store. Multiple data sources can be associated with a single data store. A host computer can have multiple data stores. Data stores can be a specific database type such as DB2 or Oracle®, or a logical system name such as SALESSYSTEM. A data source can be an EMPLOYEE table in DB2, and another data source can be a different EMPLOYEE table in Oracle.

[9] A *source object* is a hierarchical set of objects that defines the location of data and the method that is used to connect to that data. IBM WebSphere Information Analyzer uses the common metadata repository which is shared by all of the suite components in IBM Information Server. IBM WebSphere Information Analyzer organizes data from schemas, files, and other sources into a hierarchy of objects (a source object). These source objects are used to import metadata from the schemas, directories, tables, files, and columns into the metadata repository for use in analysis projects. After creation, these source objects are also available for use in other suite components. Results that are generated by IBM WebSphere Information Analyzer can be shared with other client programs such as the IBM WebSphere DataStage and IBM WebSphere QualityStage through their respective service layers. For example, you might have a host computer that you identify as ww-sales-data. On that host computer there is a data store for DB2. In that DB2 data store, there is a schema called United States sales. And within that schema there is a table called Address_ShipTo. You create an object that references this hierarchy and provides the information needed to connect to the table data. You can then import the table metadata, associate it with a project, and perform analysis on the data). Using the shared metadata repository and source objects within multiple suite components, you can share information between those components. For example, you might reference the table Address_ShipTo in a IBM WebSphere DataStage job and in an information analysis job. An Information Analyzer Data Analyst might review analysis results and create annotations and notes in the metadata of the table. Then, an Information Analyzer Project Administrator might publish the analysis results for a IBM WebSphere DataStage user to review. A IBM WebSphere DataStage developer can then go into a IBM WebSphere DataStage job, look at the Address_ShipTo table, and elect to look at the analysis results for that table. The developer can also view any notes or annotations created by the Information Analyzer Data Analyst. The developer can design or modify their job based on those results and notes. For example, the Information Analyzer Data Analyst might suggest that the data type of a particular column in the table should be changed, that a table should have a foreign key established, or that a table has invalid values that the job should eliminate.

The steps involved in configuring data source connections are as follows:

1. Define the host computer.
2. Identify and configure one or more data stores.

You must have Information Analyzer Data Administrator authority to perform these actions.[10]

Define the host computer

This step defines one or more host computers that store the data sources.

The host computer can be a physical server name, such as KAZAN.ITSOSJ.SANJOSE.IBM.COM, or a logical name such as DEMO_MACHINE.[11]

Note: You can only run analysis on metadata that is referenced and imported from a host computer.

Figure 1-16 on page 52 through Figure 1-18 on page 53 show the definition of a host computer using the logical name DEMO_MACHINE as follows:

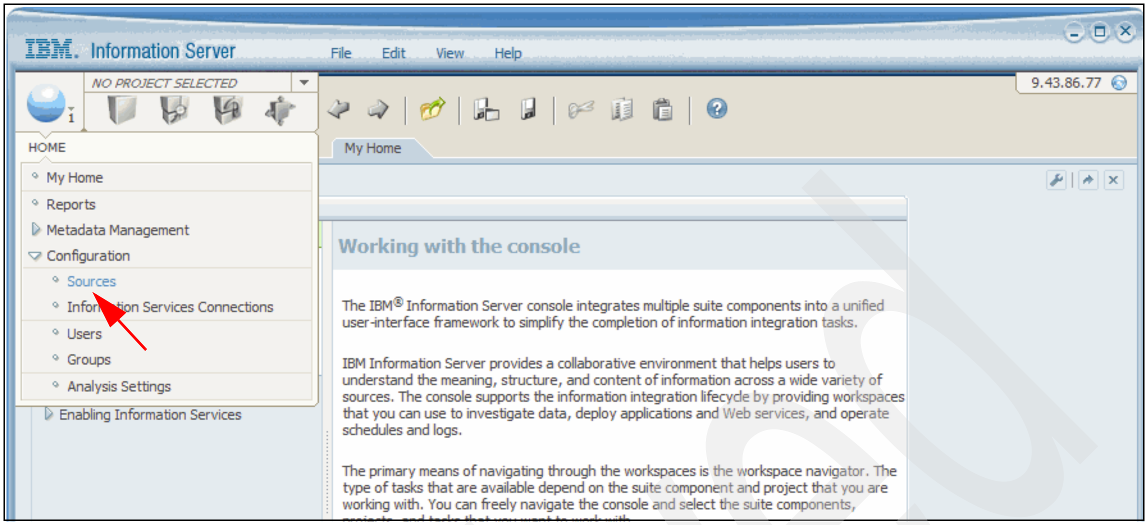
1. On the Home navigator menu in the console, select **Configuration** → **Sources** as shown in Figure 1-16 on page 52**.**
2. In the Sources workspace, click **New Host Computer** as shown in Figure 1-17 on page 52.
3. On the New Host pane, specify information about the host such as the Name (DEMO_MACHINE). Click **Save** as shown in Figure 1-18 on page 53.

After you have defined all the host computers of interest (only one in this case corresponding to the IASAMPLE database), you can proceed to add the data stores associated with each host computer as described in “Identify and configure one or more data stores” on page 54.

Note: It is not necessary for you to define all the host computers before creating Data Stores. You can create one Host computer and its associated Data Stores, followed by the addition of more host computers and their corresponding Data Stores, when appropriate.

[10] See 1.6.5, “SETUPSTEP5: Import metadata” on page 61 for details on this privilege.

[11] Many organizations move database systems from one physical server to another. To avoid being tied to a specific physical server reference that might change from time-to-time, a logical name might be the most appropriate choice for Host name.



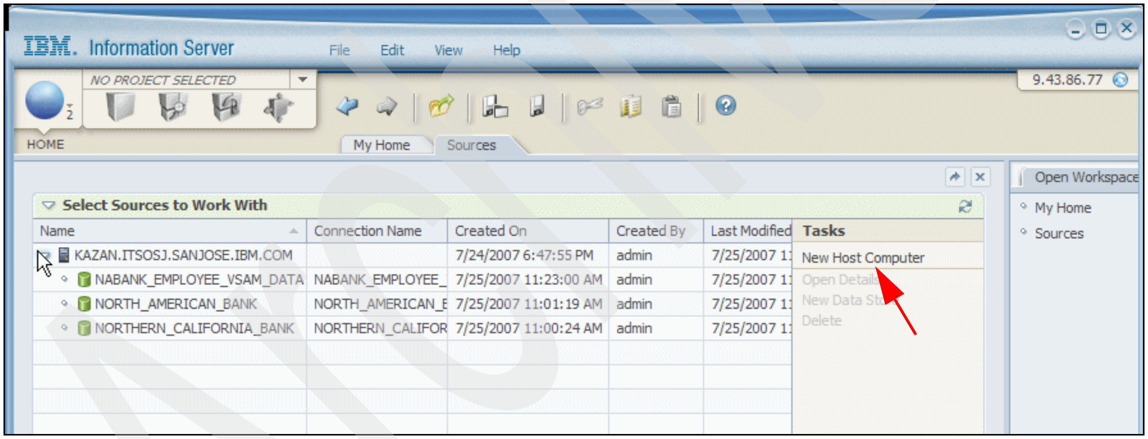
Working with the console

The IBM® Information Server console integrates multiple suite components into a unified user-interface framework to simplify the completion of information integration tasks.

IBM Information Server provides a collaborative environment that helps users to understand the meaning, structure, and content of information across a wide variety of sources. The console supports the information integration lifecycle by providing workspaces that you can use to investigate data, deploy applications and Web services, and operate schedules and logs.

The primary means of navigating through the workspaces is the workspace navigator. The type of tasks that are available depend on the suite component and project that you are working with. You can freely navigate the console and select the suite components,

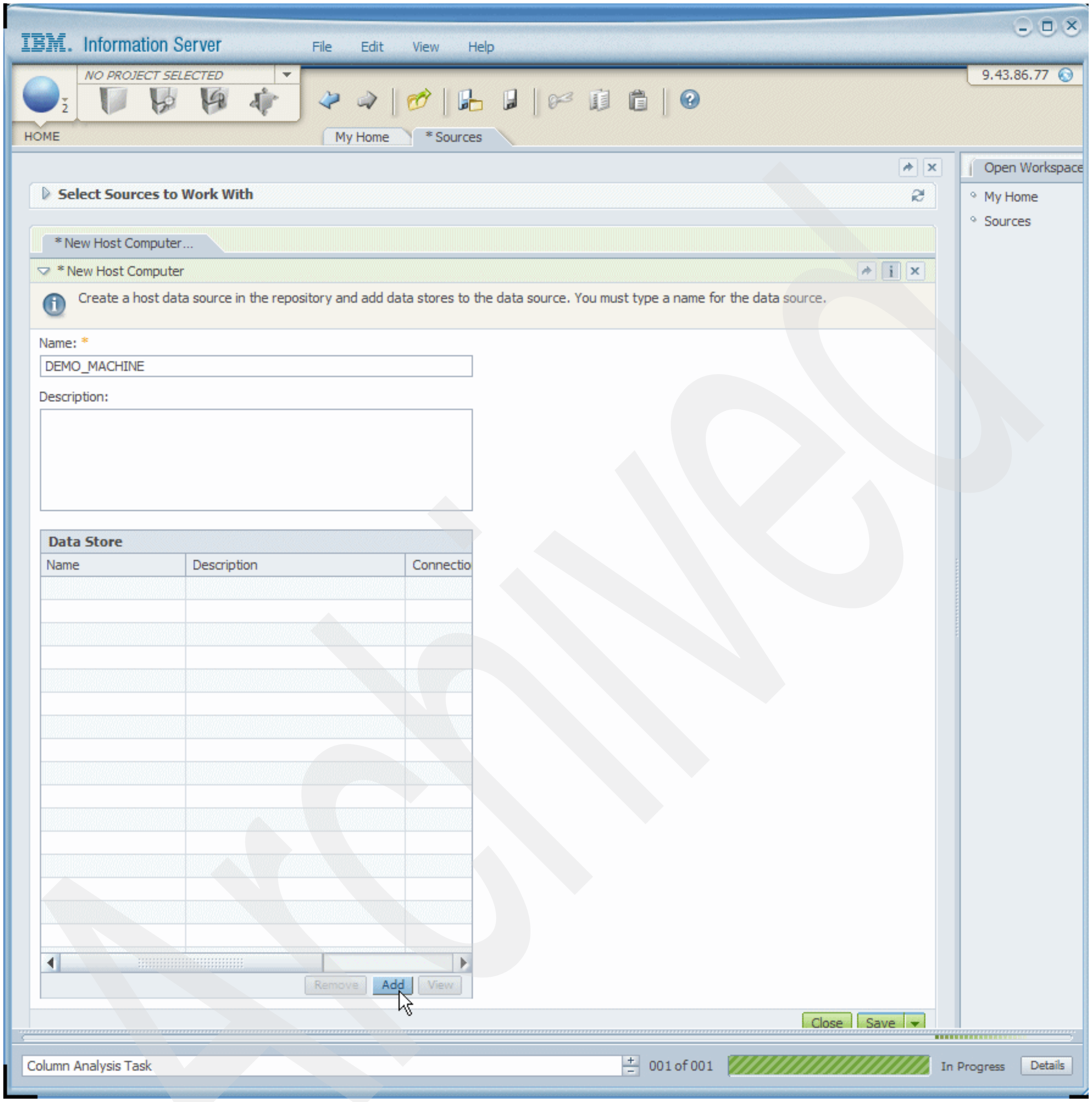
Figure 1-16 Define a host computer named DEMO_MACHINE 1/3



Select Sources to Work With

Name	Connection Name	Created On	Created By	Last Modified
KAZAN.ITSOSJ.SANJOSE.IBM.COM		7/24/2007 6:47:55 PM	admin	7/25/2007 1
NABANK_EMPLOYEE_VSAM_DATA	NABANK_EMPLOYEE_	7/25/2007 11:23:00 AM	admin	7/25/2007 1
NORTH_AMERICAN_BANK	NORTH_AMERICAN_E	7/25/2007 11:01:19 AM	admin	7/25/2007 1
NORTHERN_CALIFORNIA_BANK	NORTHERN_CALIFOR	7/25/2007 11:00:24 AM	admin	7/25/2007 1

Figure 1-17 Define a host computer named DEMO_MACHINE 2/3



Create a host data source in the repository and add data stores to the data source. You must type a name for the data source.

Name: * DEMO_MACHINE

Description:

Column Analysis Task 001 of 001 In Progress

Figure 1-18 Define a host computer named DEMO_MACHINE 3/3

Identify and configure one or more data stores

In this step, you identify and configure the data stores of interest on each host computer, define the connection to each data store by specifying a connector and a connection string, and test the connection to the data store for success.

> **Note:** The data connection links the logical metadata hierarchy to a real physical schema or directory.

Figure 1-19 on page 56 through Figure 1-24 on page 60 show the identification of two data stores and a successful connection to them as follows:

1. On the New Host Computer pane in the console, click **Add** as shown in Figure 1-18 on page 53. You can also perform this same function from the Sources workspace, by selecting the host computer to which you want to add a data store, and clicking **New Data Store**.
2. In the New Data Store pane, define the data store in the Name field as IA_SAMPLE as shown in Figure 1-19 on page 56.
3. Define the data connection in the same screen as follows:

   a. Enter a user-defined name for the connection in the Name field. We chose IA_SAMPLE, but this name need not be the same name as the data store.

   b. Optionally, type in a description for the connection.

   c. From the Connector menu, select the type of connector, which is the ODBC Connector in our case as shown in Figure 1-19 on page 56.

   d. In the Connection String field, enter a string or URL that is used to access the connector, or you select from the drop-down menu as shown in Figure 1-20 on page 57. This is the same string (OVERVIEW) as the ODBC DSN defined in the .odbc.ini file as shown in Example 1-2 on page 40.

   > **Note:** If no ODBC connection strings are listed, confirm that the .odbc.ini file was properly configured. For certain data sources such as text files, Microsoft® (MS) Access databases, MySQL databases, other databases that lack schemas, select DATA FILE as the store type to properly connect and ensure that the username/password is valid.

   e. Provide the User Name (db2inst1) and Password to access the connector. Click **Connect** to test the connection as shown in Figure 1-21 on page 58.

Attention: In Figure 1-21 on page 58, the Store Type field under the Data Store information has the value Database. For text-based data sources (whether fixed width or delimited, and including those from wire protocol drivers), you *must* choose Data File from the drop-down list. Data sources include MS Access and older databases such as Progress and Pervasive.

f. If the connection is successful, information about the data store displays in the Data Store Information fields as shown in Figure 1-22 on page 59. Ensure that the Store Type is correct, which is Database is our case. Click **Save and Close**.

Note: If the connection does not succeed, the Connection window lists the configuration details that caused the error. Modify the configuration details and then test the connection again.

g. Figure 1-23 on page 60 shows the creation of another data store named *IA_SAMPLE_ALT* on the DEMO_MACHINE with a data connection name of IA_SAMPLE_ALT. However, it has the same connection string as in the case of the IA_SAMPLE data connection. Click **Connect** to test this connection. The successful test of this connection is not shown here.

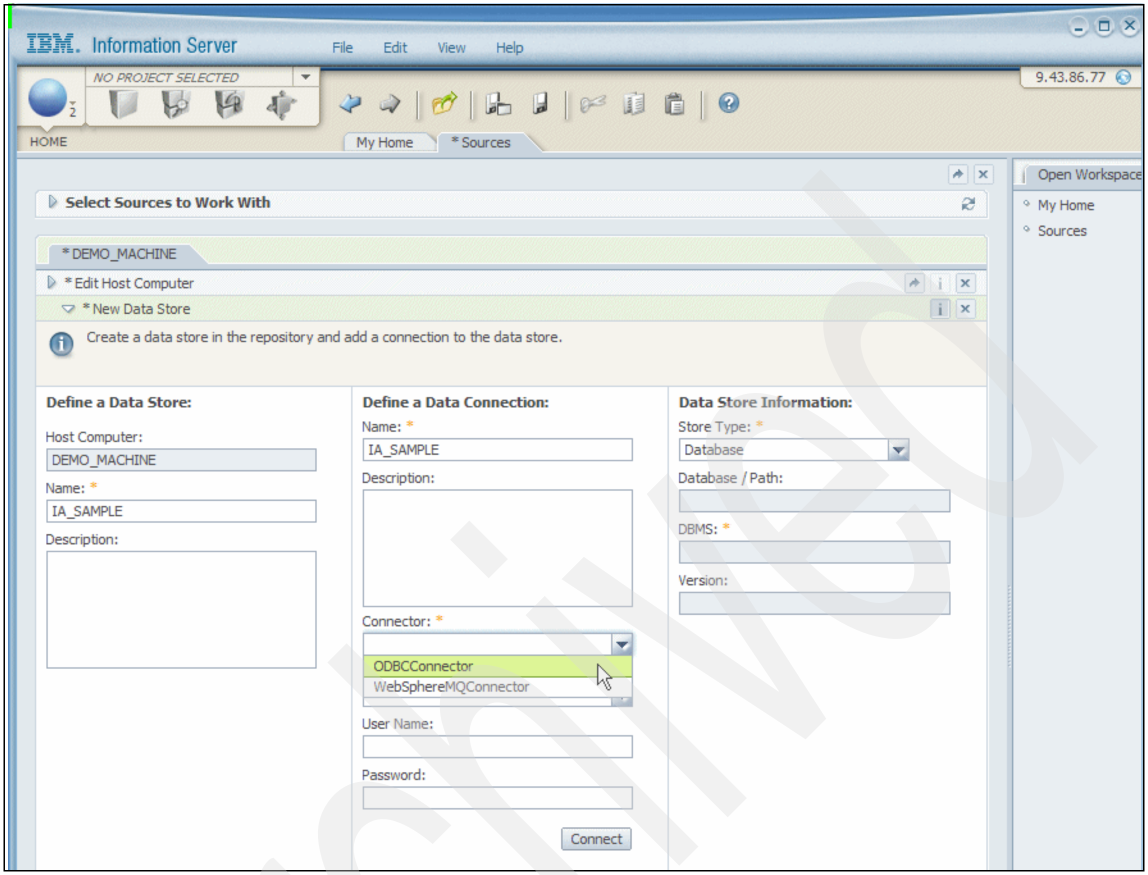
   This feature gives you the flexibility to view the same object as two different entries in the Import Metadata workspace as shown in Figure 1-24 on page 60. This allows you to have multiple sets of users work on the same object using different configuration options without interfering with each others work.

   One potential use of different configuration options is to set up different “Analysis Where” clauses for each connection to get different slices/segments of the data source.

Figure 1-24 on page 60 shows the two data stores that we just created. The intervening screens are not shown here.

After all the data stores have been successfully defined, you can proceed to importing the metadata into the metadata repository as described in 1.6.5, “SETUPSTEP5: Import metadata” on page 61.[12]

[12] You do not have to define all the data stores at the same time. You can define them incrementally over time.



Select Sources to Work With

* DEMO_MACHINE

* Edit Host Computer

* New Data Store

Create a data store in the repository and add a connection to the data store.

Define a Data Store:

Host Computer: DEMO_MACHINE

Name: * IA_SAMPLE

Description:

Define a Data Connection:

Name: * IA_SAMPLE

Description:

Connector: *

ODBCConnector

WebSphereMQConnector

User Name:

Password:

Data Store Information:

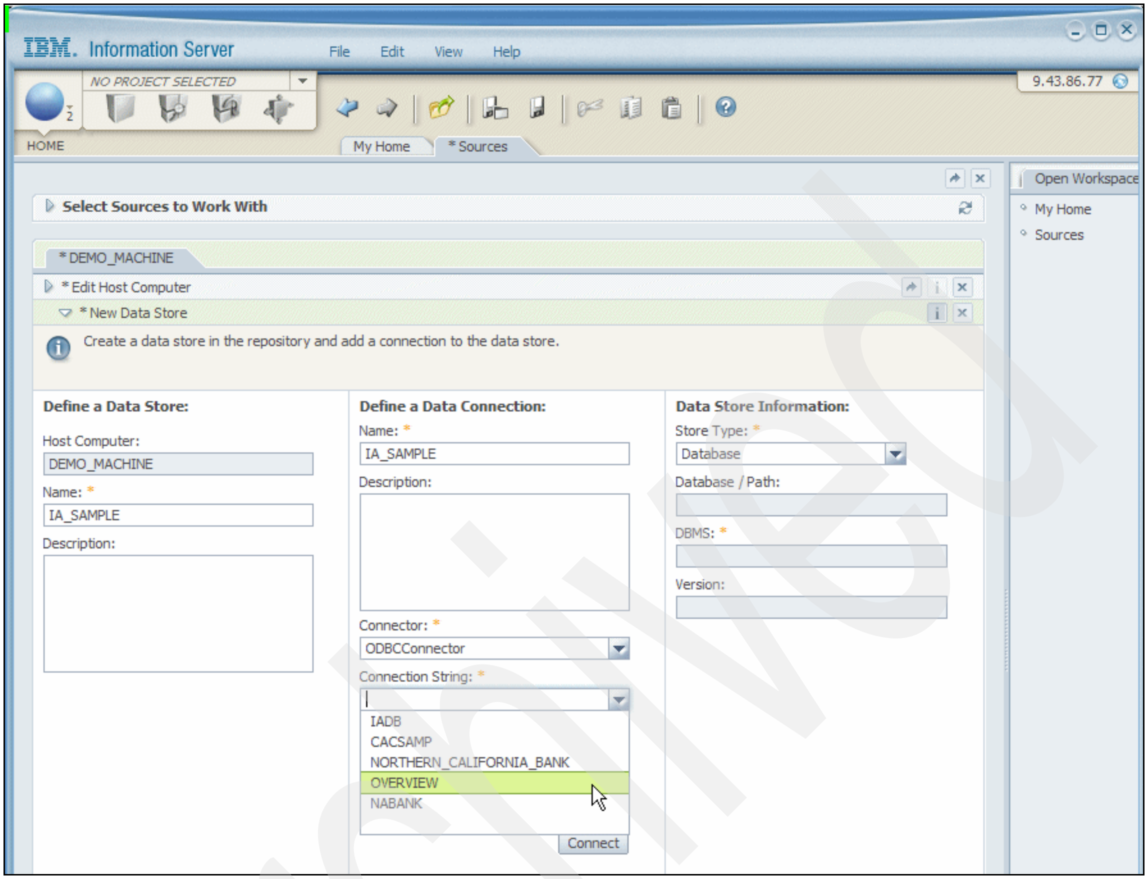
Store Type: * Database

Database / Path:

DBMS: *

Version:

Figure 1-19 Identify and configure a data store 1/6



Select Sources to Work With

* DEMO_MACHINE

* Edit Host Computer

* New Data Store

Create a data store in the repository and add a connection to the data store.

Define a Data Store:

Host Computer: DEMO_MACHINE

Name: * IA_SAMPLE

Description:

Define a Data Connection:

Name: * IA_SAMPLE

Description:

Connector: * ODBCConnector

Connection String: *

- IADB
- CACSAMP
- NORTHERN_CALIFORNIA_BANK
- OVERVIEW
- NABANK

Data Store Information:

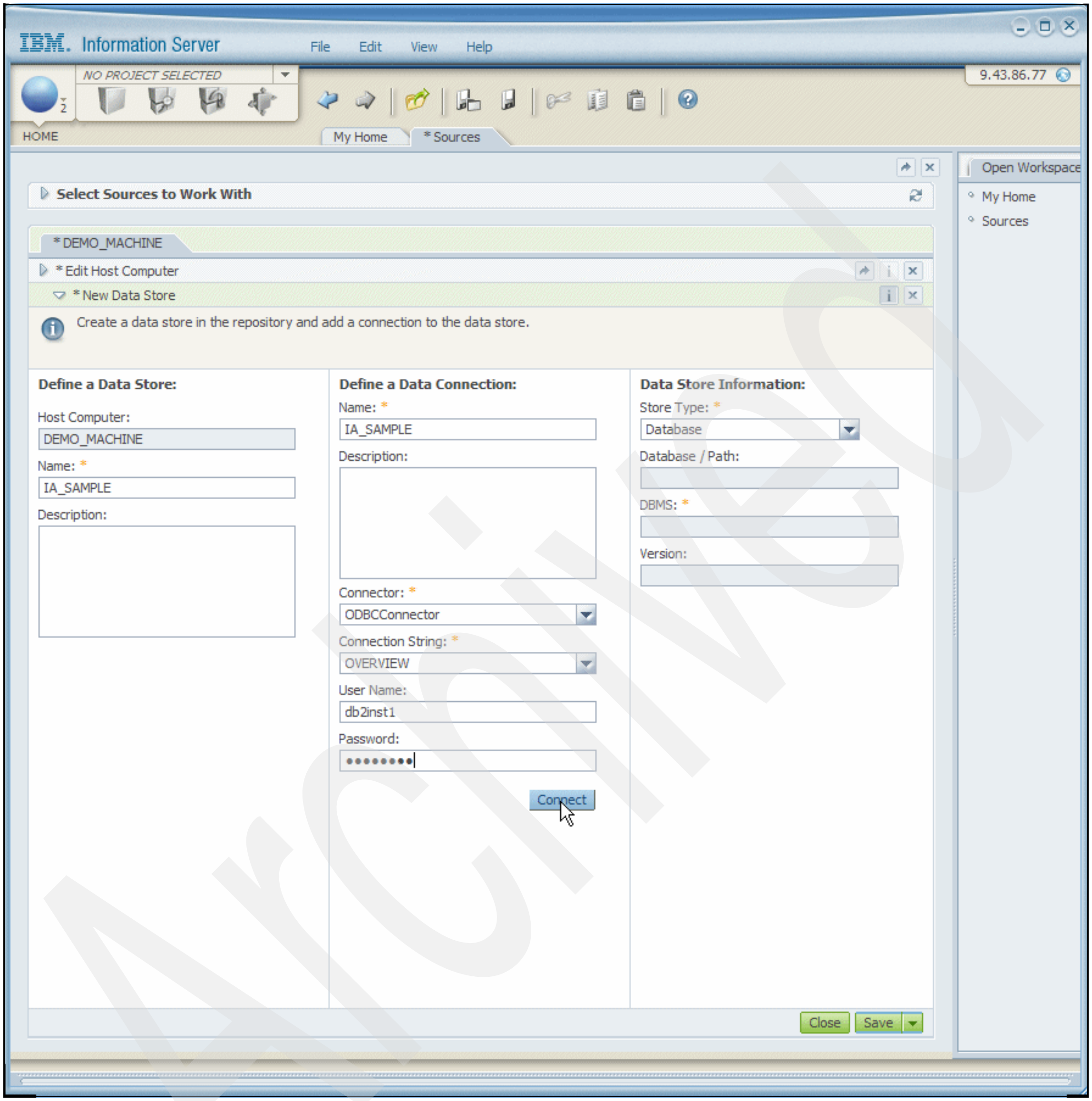
Store Type: * Database

Database / Path:

DBMS: *

Version:

Figure 1-20 Identify and configure a data store 2/6



Select Sources to Work With

* DEMO_MACHINE

* Edit Host Computer

* New Data Store

Create a data store in the repository and add a connection to the data store.

Define a Data Store:

Host Computer: DEMO_MACHINE

Name: * IA_SAMPLE

Description:

Define a Data Connection:

Name: * IA_SAMPLE

Description:

Connector: * ODBCConnector

Connection String: * OVERVIEW

User Name: db2inst1

Password: ••••••••

Data Store Information:

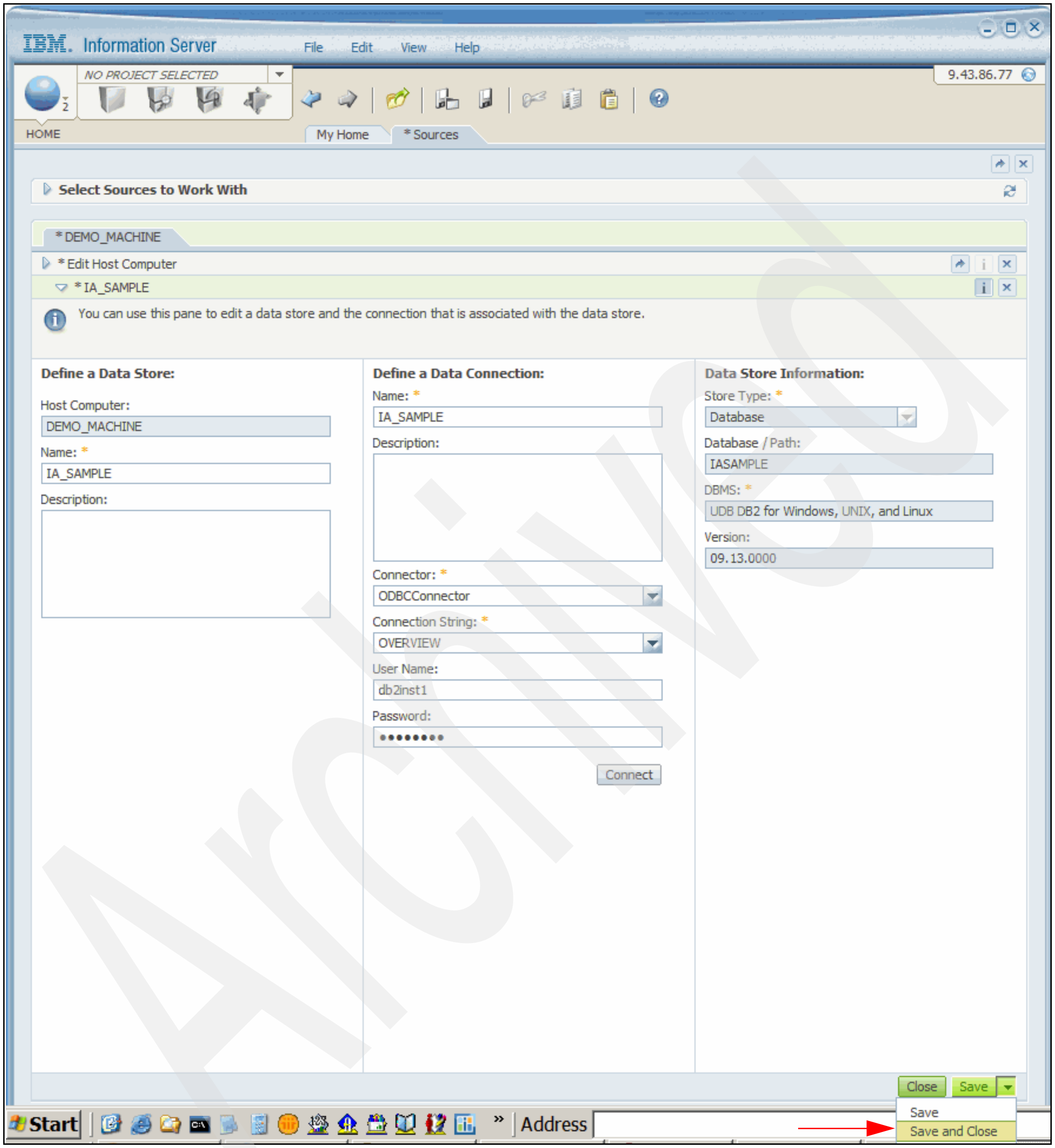
Store Type: * Database

Database / Path:

DBMS: *

Version:

Figure 1-21 Identify and configure a data store 3/6



Select Sources to Work With

* DEMO_MACHINE

* Edit Host Computer

* IA_SAMPLE

You can use this pane to edit a data store and the connection that is associated with the data store.

Define a Data Store:

Host Computer: DEMO_MACHINE

Name: * IA_SAMPLE

Description:

Define a Data Connection:

Name: * IA_SAMPLE

Description:

Connector: * ODBCConnector

Connection String: * OVERVIEW

User Name: db2inst1

Password: ••••••••

Data Store Information:

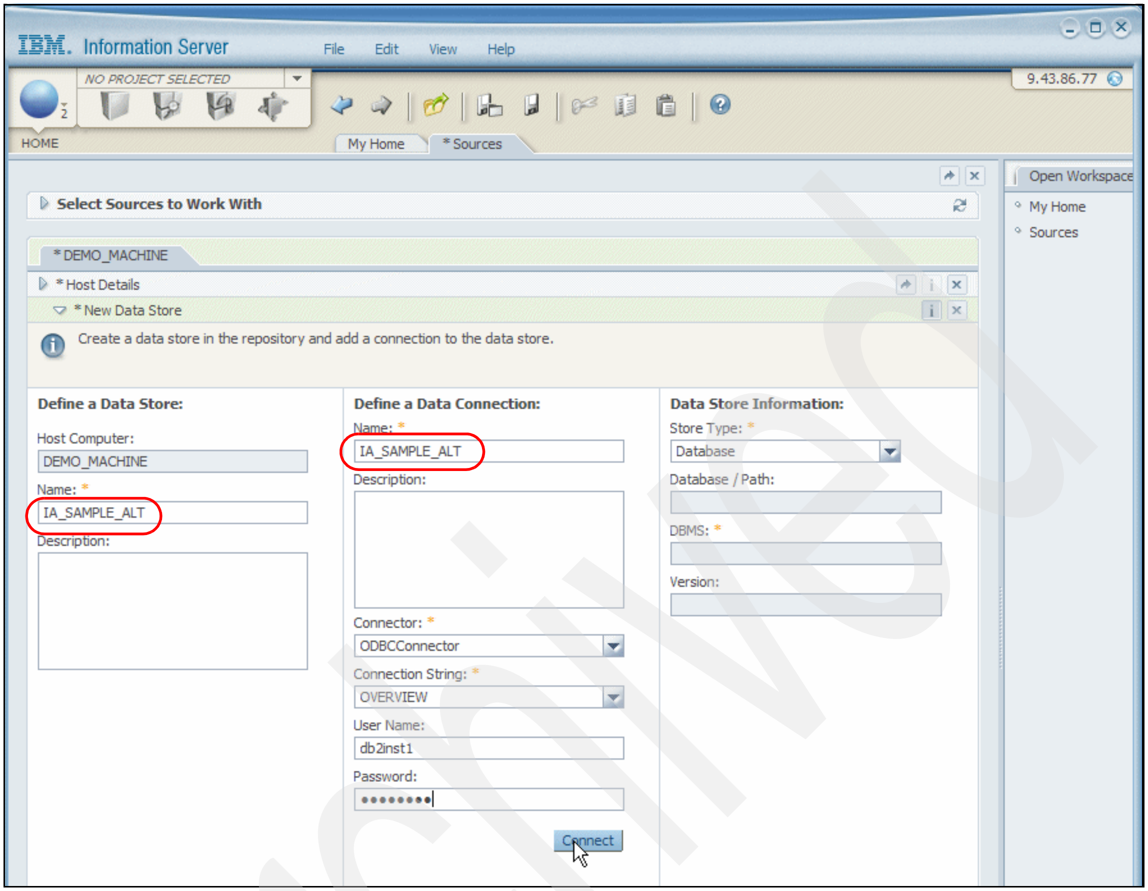
Store Type: * Database

Database / Path: IASAMPLE

DBMS: * UDB DB2 for Windows, UNIX, and Linux

Version: 09.13.0000

Figure 1-22 Identify and configure a data store 4/6



Select Sources to Work With

* DEMO_MACHINE

* Host Details

* New Data Store

Create a data store in the repository and add a connection to the data store.

Define a Data Store:

Host Computer: DEMO_MACHINE

Name: * IA_SAMPLE_ALT

Description:

Define a Data Connection:

Name: * IA_SAMPLE_ALT

Description:

Connector: * ODBCConnector

Connection String: * OVERVIEW

User Name: db2inst1

Password: ••••••••

Connect

Data Store Information:

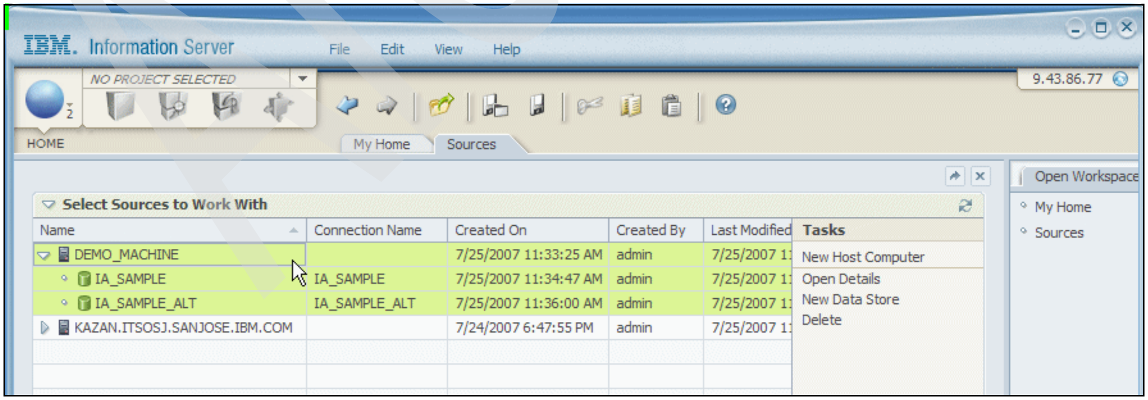
Store Type: * Database

Database / Path:

DBMS: *

Version:

Figure 1-23   Identify and configure a data store 5/6



Select Sources to Work With

Name	Connection Name	Created On	Created By	Last Modified
DEMO_MACHINE		7/25/2007 11:33:25 AM	admin	7/25/2007 11
IA_SAMPLE	IA_SAMPLE	7/25/2007 11:34:47 AM	admin	7/25/2007 11
IA_SAMPLE_ALT	IA_SAMPLE_ALT	7/25/2007 11:36:00 AM	admin	7/25/2007 11
KAZAN.ITSOSJ.SANJOSE.IBM.COM		7/24/2007 6:47:55 PM	admin	7/25/2007 11

Tasks

New Host Computer

Open Details

New Data Store

Delete

Figure 1-24   Identify and configure a data store 6/6

1.6.5 SETUPSTEP5: Import metadata

You import metadata into the metadata repository for use in all information analysis projects.[13] You must import metadata and associate it with a particular project before you can run information analysis jobs.

You can import all, or selected tables, files, or columns in a schema/directory.

The import process consists of multiple steps that you complete to import metadata into the metadata repository. Metadata is data that describes the content, structure, and other characteristics of an original data set. You use metadata in all of your analysis projects to describe the data that you want to analyze and to use as a foundation to access inferences. You can access information in the metadata repository from any component in the suite, which means that metadata or analysis results can be shared by multiple suite components.

Figure 1-25 describes the metadata import process.

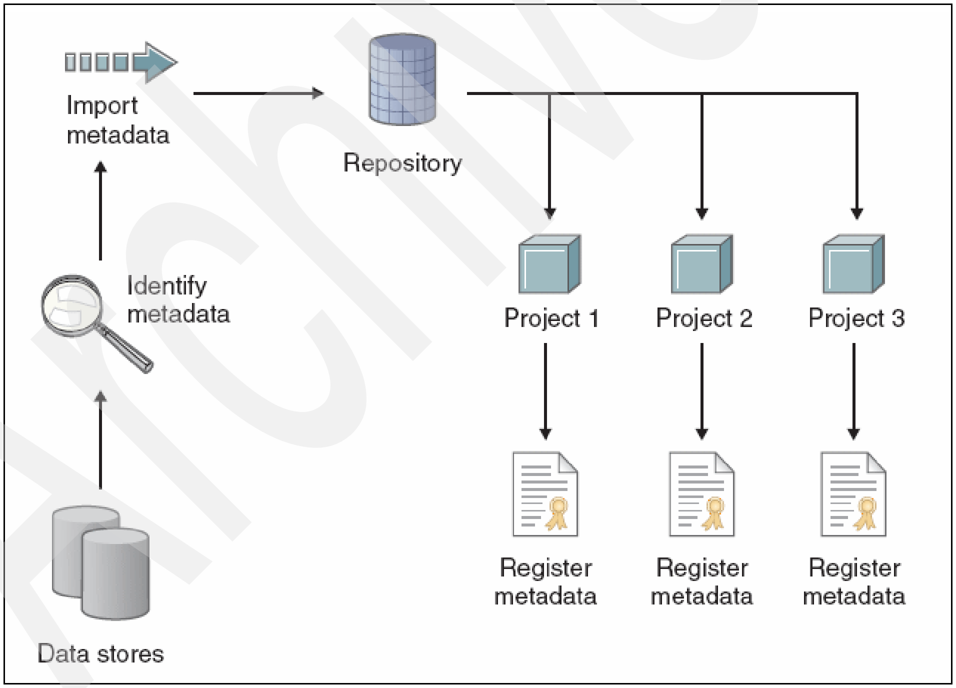


Figure 1-25 Metadata import process

[13] The metadata repository (XMETA database) stores imported metadata, project configurations, reports, and results for all components of IBM Information Server.

The steps are as follows:

1. Define the data stores as described in 1.6.4, “SETUPSTEP4: Configure data source connections” on page 49.
2. After selecting a data store, you need to discover the metadata that it contains. When you identify the data store, the system connects to the data store, and the Import Metadata workspace displays a hierarchy of the defined host computers and data stores. For each data store, if you have already discovered the metadata, the hierarchy contains the discovered schemas, directories, tables, files, and columns. This identification only persists across users sessions if you complete the import step. Otherwise, each time you open the Metadata Import workspace, you must again identify the hierarchy of schemas, directories, tables, or files on the host computer.

   A *hierarchy* is a list of objects that are displayed at specific levels depending on the size of the object. For example, you select a data store to view a schema within the data store. To view a table, you then select the schema to view the tables within the schema. You must identify down to the column level in order to import metadata.
3. After you identify the metadata in a data store, you select the objects that you want and then import these objects into the metadata repository. For example, you might want to import specific tables rather than all of the tables in the data store.
4. After you import metadata into the metadata repository, you must associate the metadata with a project or multiple projects that you want to use it in. You register metadata to your project at the data store, table, file, or column level.

> **Note:** After you import the metadata into the metadata repository, you cannot modify its names or attributes. However, you can add information to it, such as terms, policies, and contacts.

Figure 1-26 on page 64 through Figure 1-34 on page 69 describe the identification of the metadata and the import of the metadata from the IA_SAMPLE database into the metadata repository as follows:

1. On the Home navigator menu in the console, select **Metadata Management** → **Import Metadata** as shown in Figure 1-26 on page 64.
2. On the **Import Metadata** workspace, select the host computer (DEMO_MACHINE) that you want to identify tables, files, and columns in as shown in Figure 1-27 on page 64.
3. Identify the data stores of interest on the host computer such as IA_SAMPLE_ALT and IA_SAMPLE by highlighting them. Then click **Identify Next Level** in the Tasks pane as shown in Figure 1-27 on page 64. This option provides a direct discovery process that focuses only on the paths of

interest to you. To import metadata, you must continue to progressively identify data stores to the column level.

Note: When you click **Identify All Levels** on the Tasks pane, it discovers all of the items in the selected data store. For schemas or directories with thousands of tables and columns, this step will take a long time to complete and is not recommended.

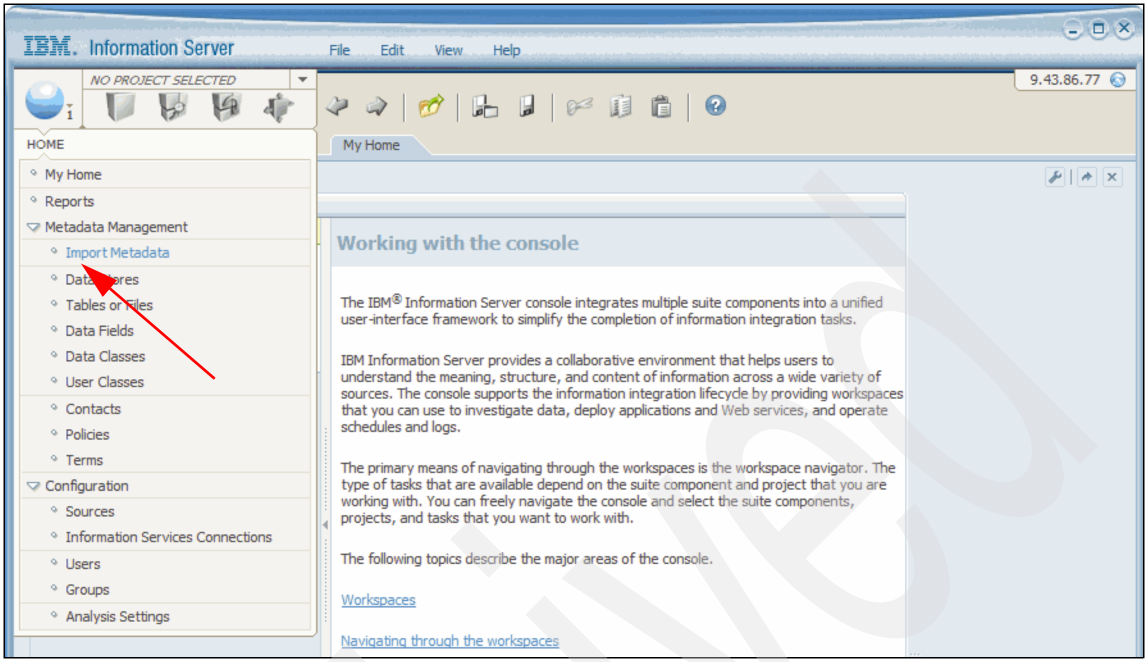
Figure 1-28 on page 65 and Figure 1-29 on page 65 show the successful identification of the data stores and the number of schemas and directories discovered.

4. Select the IA schema, and click **Identify Next Level** as shown in Figure 1-29 on page 65 to view all the tables associated with this schema as shown in Figure 1-30 on page 66 through Figure 1-31 on page 67.

   Repeat the process for the IA_SAMPLE data store (not shown here).

5. Select all the tables in the IA schema and click **Import** as shown in Figure 1-32 on page 68. The list of directories and tables that will be imported are identified as shown in Figure 1-33 on page 69.
6. Click **OK** as shown in Figure 1-33 on page 69 to initiate import of the selected data sources. Figure 1-34 on page 69 shows the successful import of the selected tables.
7. Repeat the process for all the tables in the IA schema in the IA_SAMPLE data store.

You can now proceed to create a new project or open an existing project as described in 1.6.6, “SETUPSTEP6: Create/open a project” on page 70.



Working with the console

The IBM® Information Server console integrates multiple suite components into a unified user-interface framework to simplify the completion of information integration tasks.

IBM Information Server provides a collaborative environment that helps users to understand the meaning, structure, and content of information across a wide variety of sources. The console supports the information integration lifecycle by providing workspaces that you can use to investigate data, deploy applications and Web services, and operate schedules and logs.

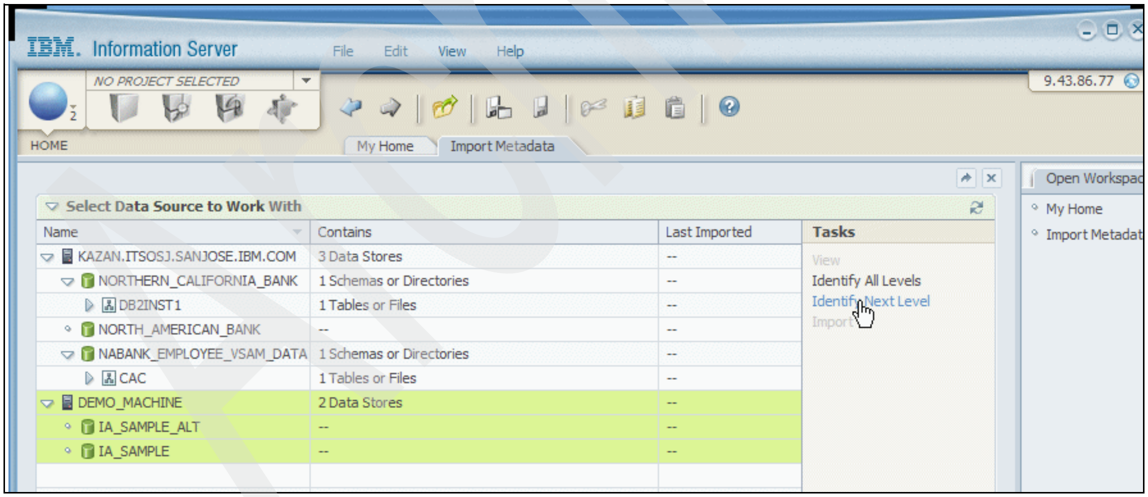
The primary means of navigating through the workspaces is the workspace navigator. The type of tasks that are available depend on the suite component and project that you are working with. You can freely navigate the console and select the suite components, projects, and tasks that you want to work with.

The following topics describe the major areas of the console.

Workspaces

Navigating through the workspaces

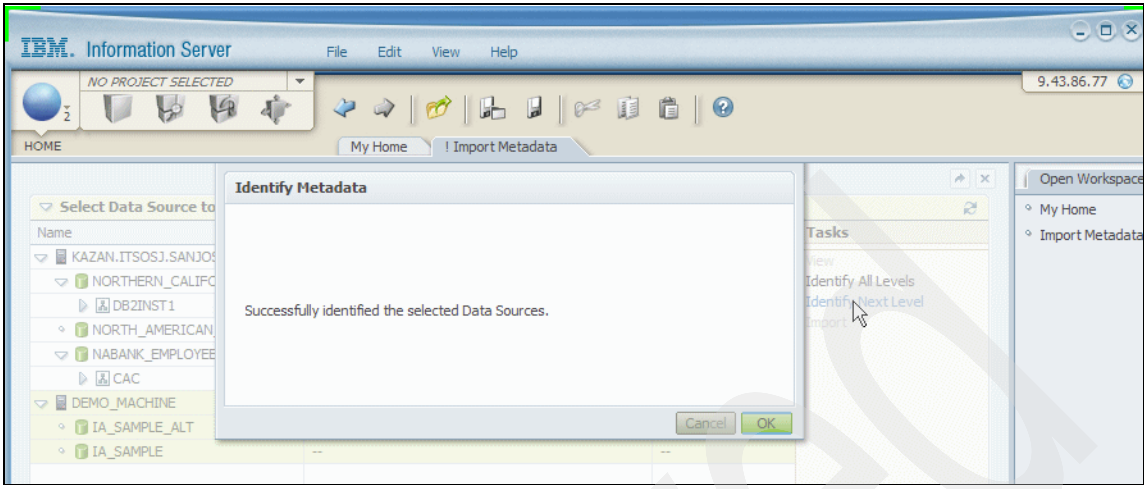
Figure 1-26 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 1/9



Select Data Source to Work With

Name	Contains	Last Imported
KAZAN.ITSOSJ.SANJOSE.IBM.COM	3 Data Stores	--
NORTHERN_CALIFORNIA_BANK	1 Schemas or Directories	--
DB2INST1	1 Tables or Files	--
NORTH_AMERICAN_BANK	--	--
NABANK_EMPLOYEE_VSAM_DATA	1 Schemas or Directories	--
CAC	1 Tables or Files	--
DEMO_MACHINE	2 Data Stores	--
IA_SAMPLE_ALT	--	--
IA_SAMPLE	--	--

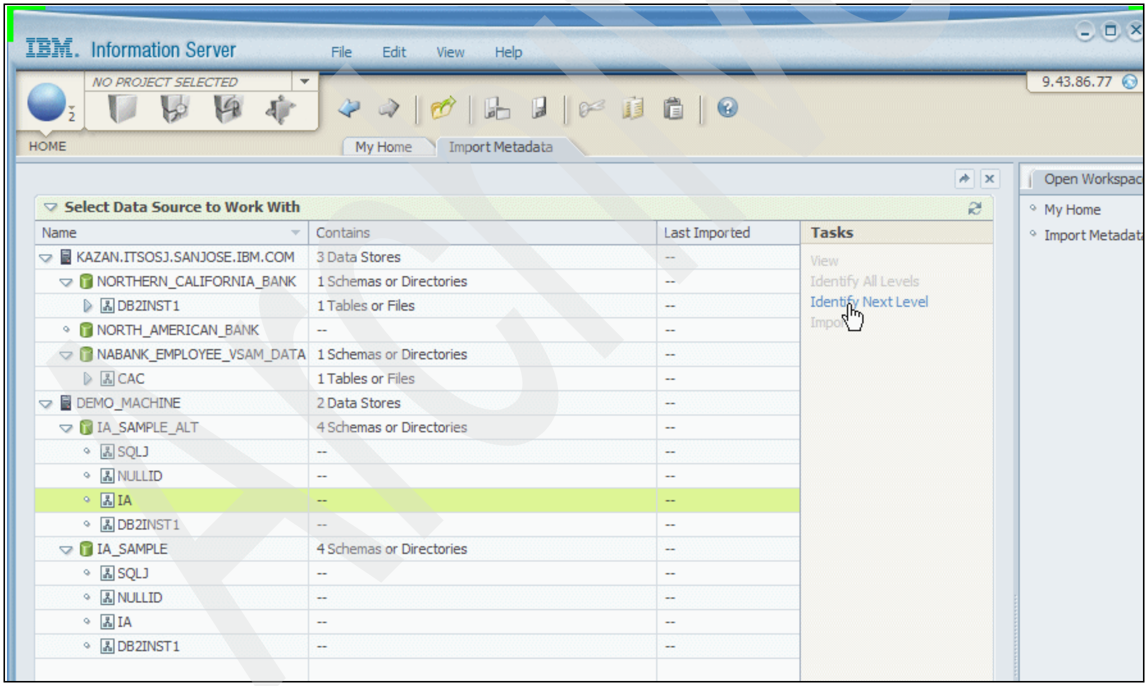
Figure 1-27 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 2/9



Identify Metadata

Successfully identified the selected Data Sources.

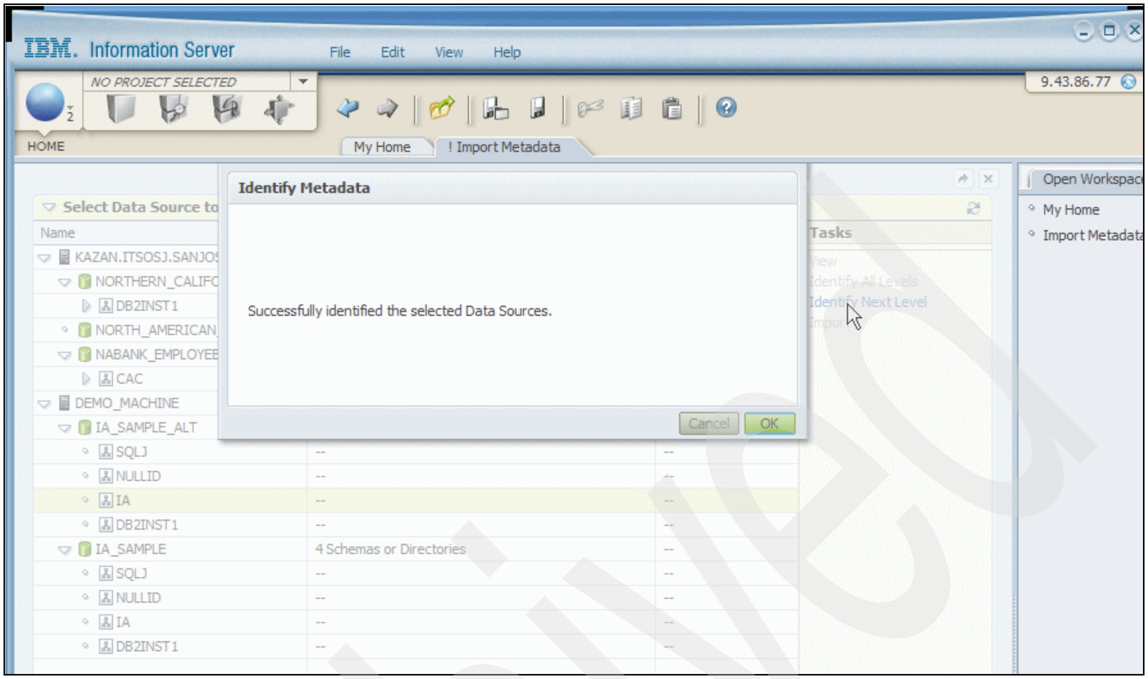
Figure 1-28 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 3/9



Select Data Source to Work With

Name	Contains	Last Imported
KAZAN.ITSOSJ.SANJOSE.IBM.COM	3 Data Stores	--
NORTHERN_CALIFORNIA_BANK	1 Schemas or Directories	--
DB2INST1	1 Tables or Files	--
NORTH_AMERICAN_BANK	--	--
NABANK_EMPLOYEE_VSAM_DATA	1 Schemas or Directories	--
CAC	1 Tables or Files	--
DEMO_MACHINE	2 Data Stores	--
IA_SAMPLE_ALT	4 Schemas or Directories	--
SQLJ	--	--
NULLID	--	--
IA	--	--
DB2INST1	--	--
IA_SAMPLE	4 Schemas or Directories	--
SQLJ	--	--
NULLID	--	--
IA	--	--
DB2INST1	--	--

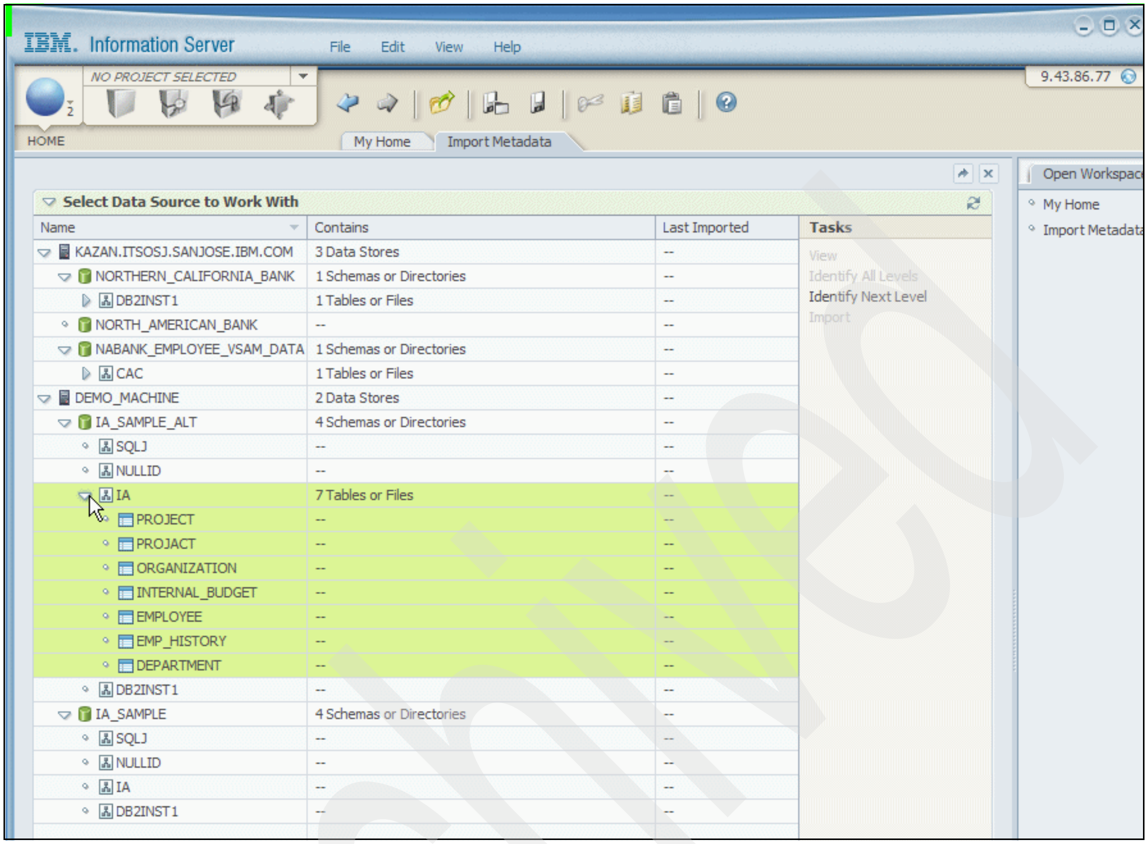
Figure 1-29 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 4/9



Identify Metadata

Successfully identified the selected Data Sources.

Figure 1-30 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 5/9



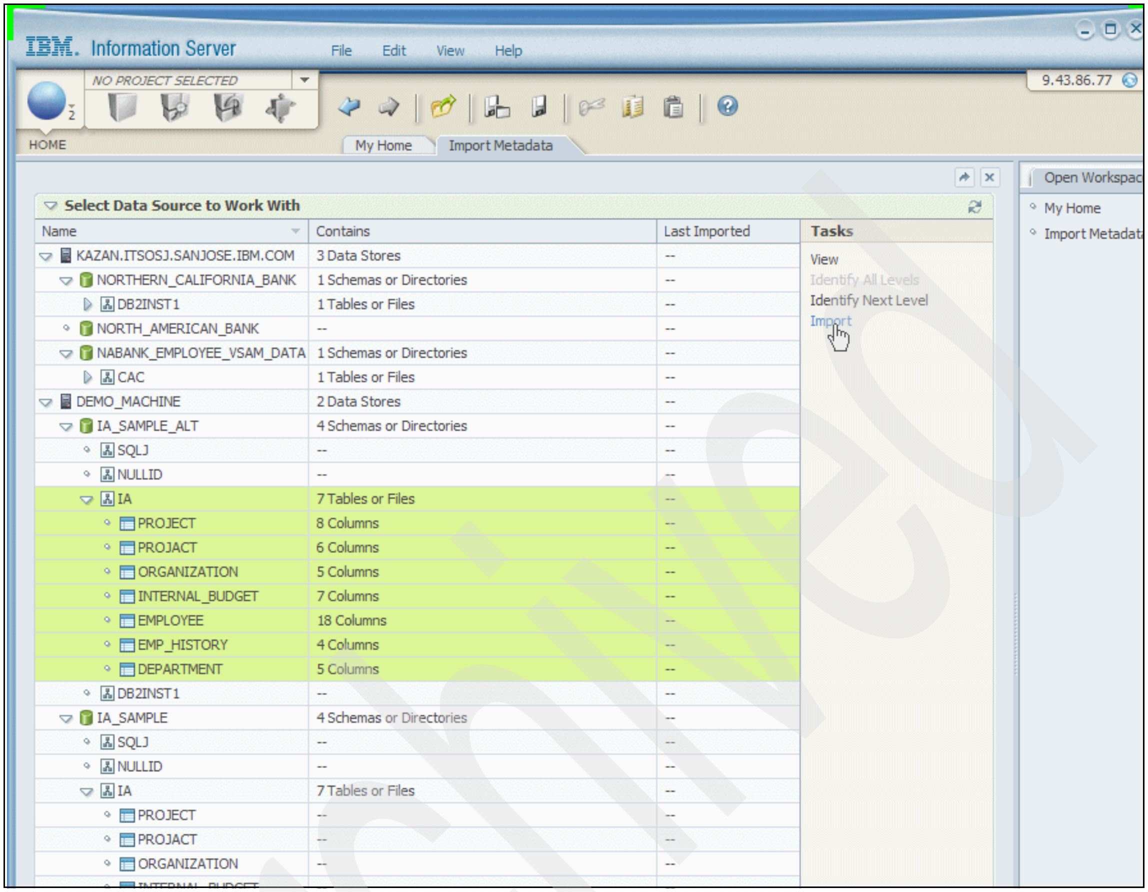
Select Data Source to Work With

Name	Contains	Last Imported
KAZAN.ITSOSJ.SANJOSE.IBM.COM	3 Data Stores	--
NORTHERN_CALIFORNIA_BANK	1 Schemas or Directories	--
DB2INST1	1 Tables or Files	--
NORTH_AMERICAN_BANK	--	--
NABANK_EMPLOYEE_VSAM_DATA	1 Schemas or Directories	--
CAC	1 Tables or Files	--
DEMO_MACHINE	2 Data Stores	--
IA_SAMPLE_ALT	4 Schemas or Directories	--
SQLJ	--	--
NULLID	--	--
IA	7 Tables or Files	--
PROJECT	--	--
PROJACT	--	--
ORGANIZATION	--	--
INTERNAL_BUDGET	--	--
EMPLOYEE	--	--
EMP_HISTORY	--	--
DEPARTMENT	--	--
DB2INST1	--	--
IA_SAMPLE	4 Schemas or Directories	--
SQLJ	--	--
NULLID	--	--
IA	--	--
DB2INST1	--	--

Tasks

View
Identify All Levels
Identify Next Level
Import

Figure 1-31 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 6/9



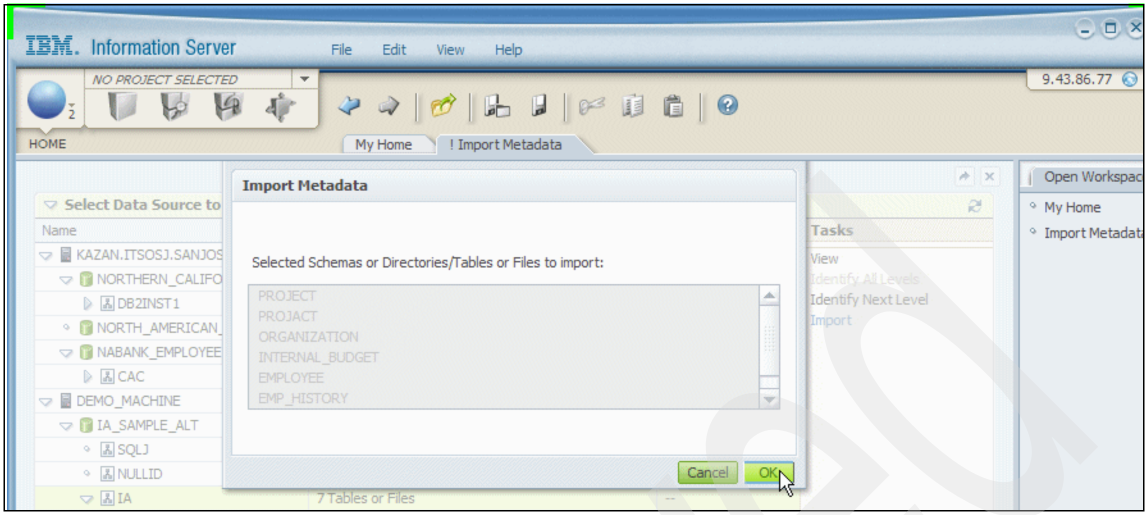
Select Data Source to Work With

Name	Contains	Last Imported
KAZAN.ITSOSJ.SANJOSE.IBM.COM	3 Data Stores	--
NORTHERN_CALIFORNIA_BANK	1 Schemas or Directories	--
DB2INST1	1 Tables or Files	--
NORTH_AMERICAN_BANK	--	--
NABANK_EMPLOYEE_VSAM_DATA	1 Schemas or Directories	--
CAC	1 Tables or Files	--
DEMO_MACHINE	2 Data Stores	--
IA_SAMPLE_ALT	4 Schemas or Directories	--
SQLJ	--	--
NULLID	--	--
IA	7 Tables or Files	--
PROJECT	8 Columns	--
PROJACT	6 Columns	--
ORGANIZATION	5 Columns	--
INTERNAL_BUDGET	7 Columns	--
EMPLOYEE	18 Columns	--
EMP_HISTORY	4 Columns	--
DEPARTMENT	5 Columns	--
DB2INST1	--	--
IA_SAMPLE	4 Schemas or Directories	--
SQLJ	--	--
NULLID	--	--
IA	7 Tables or Files	--
PROJECT	--	--
PROJACT	--	--
ORGANIZATION	--	--

Tasks

View
Identify All Levels
Identify Next Level
Import

Figure 1-32   Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 7/9



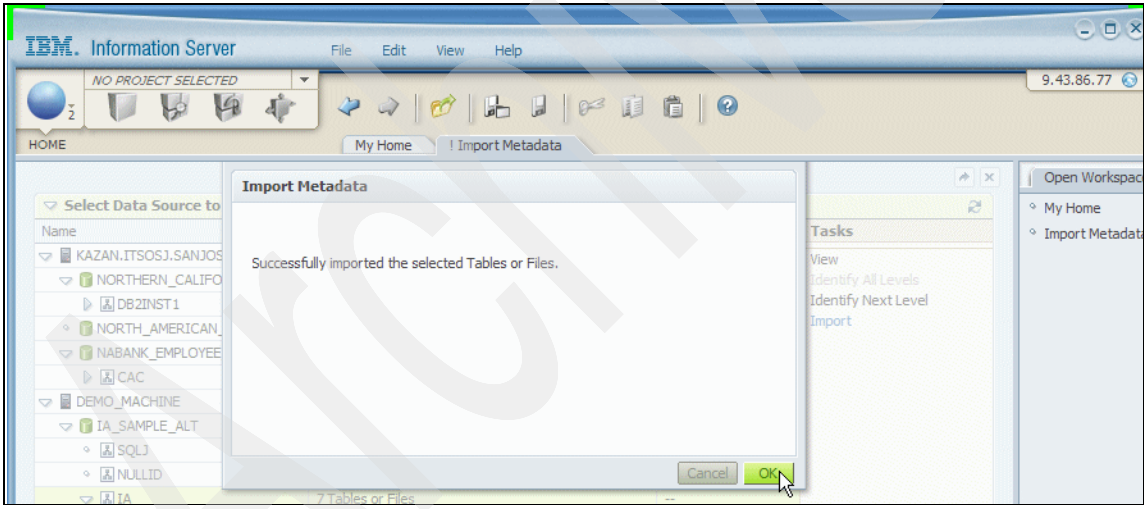
Import Metadata

Selected Schemas or Directories/Tables or Files to import:

PROJECT
PROJACT
ORGANIZATION
INTERNAL_BUDGET
EMPLOYEE
EMP_HISTORY

Cancel OK

Figure 1-33 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 8/9



Import Metadata

Successfully imported the selected Tables or Files.

Cancel OK

Figure 1-34 Import metadata from the IA_SAMPLE_ALT database in the IA_SAMPLE_ALT data store 9/9

1.6.6 SETUPSTEP6: Create/open a project

A *project* is an object that contains all the information that you need to build and run an analysis job and view analysis results. A project is a logical container that provides a secure framework that binds together a set of data sources for analysis with a set of users performing specific roles.

After creating a project, you:

1. Associate metadata with the project.
2. Assign users to the project.
3. Set the analysis options for all analysis that occurs in the project.

Each project is unique, depending on the analytical settings that were configured in the project when it was created. However, multiple projects can access the same metadata that is stored in the metadata repository, and share the same users and data sources across suite components.

Projects provide a collaborative environment that help users to understand the meaning, structure, and content of information across a wide variety of sources. Multiple users can contribute to a project and view the status of a project over time. In IBM WebSphere Information Analyzer, multiple users can access a project at one time. Users can view project data, review analysis results, enter annotations and notes, and run analysis jobs. When multiple analysis jobs are sent for processing, they are processed in a queue.[14] If you are adding annotations, modifying analysis results, or modifying project details, be aware that other users might also be modifying that information. Before you edit or add information, click the refresh icon to ensure that you are viewing the latest information.

Some of the best practices when creating a project are as follows:

- A project should be accessible to a specific role and multiple access to the same project should be avoided.
- A project should be assigned to only the required data sources in order to minimize overhead.
- A project should be assigned a name that identifies the implicit nature of the job and business domain it addresses.
- A project's properties (various threshold settings) should be defined as appropriate at the project level when the project is created.

[14] In Version 8.1, all submitted jobs will be processed in parallel.

> **Note:** Typically, the Information Analyzer Data Administrator imports metadata, and the Information Analyzer Project administrator registers it to the project. This segmentation and distinction is important because it allows an organization to distinguish these roles and provide separation of duties if required to protect sensitive information.

Figure 1-35 on page 71 through Figure 1-38 on page 72 describe the creation of a project as follows:

1. On the File menu in the console, select **New Project** as shown in Figure 1-35 on page 71.
2. In the New Project window, select the type of project that you want to create. In our case, this is Information Analyzer as shown in Figure 1-36 on page 71. The Type field appears only if more than one suite component is installed.
3. Provide the Name (IA_OVERVIEW_PROJECT) and click **OK** as shown in Figure 1-37 on page 72.
4. The Project Properties workspace opens as shown in Figure 1-38 on page 72.

After the project has been created (or opened in case it already exists), you can proceed to associate metadata with it as described in 1.6.7, “SETUPSTEP7: Configure the project” on page 73.

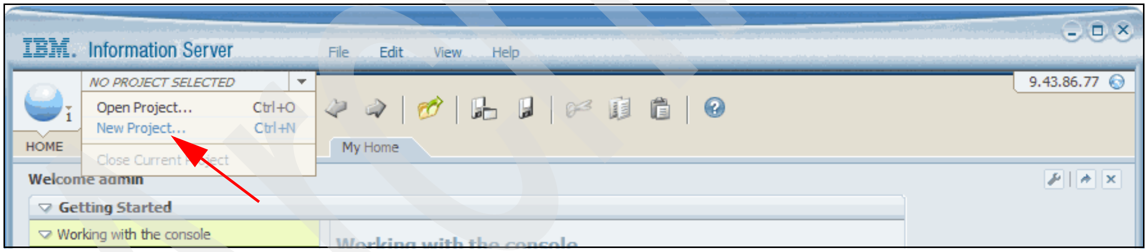


Figure 1-35 Create a project 1/4

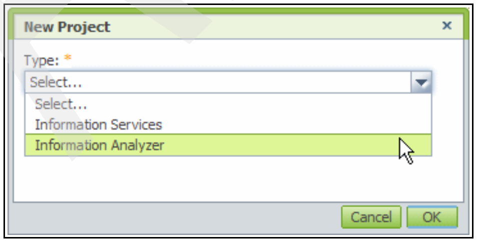
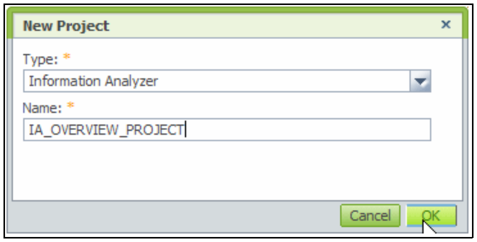
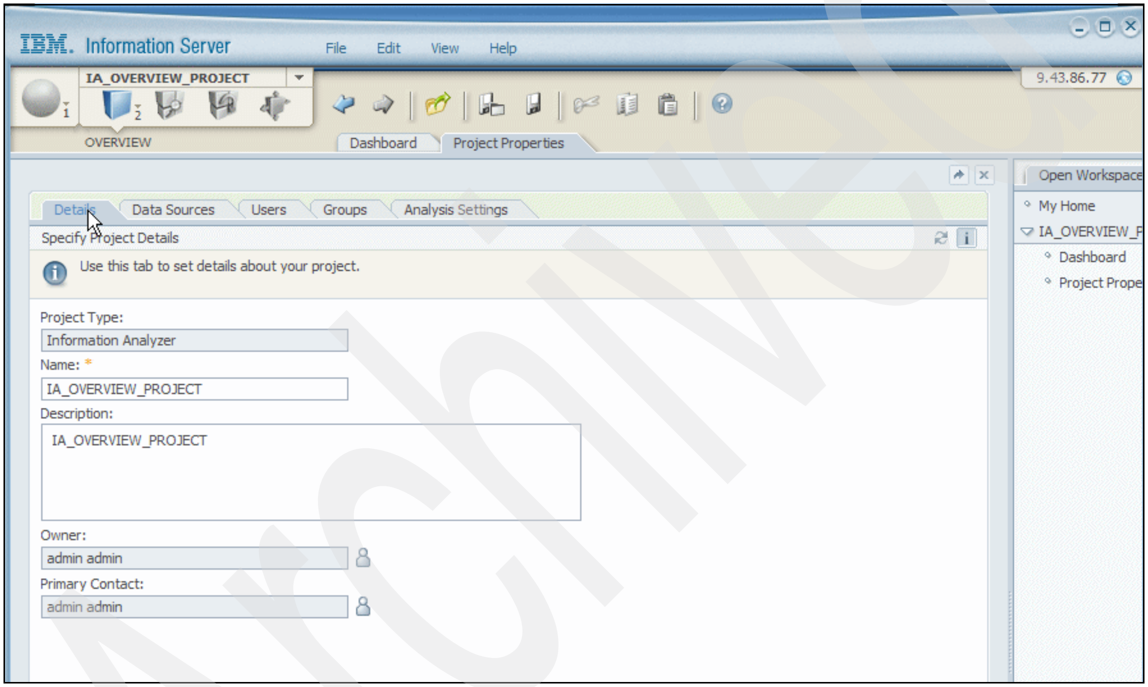


Figure 1-36 Create a project 2/4



Type: *
Information Analyzer
Name: *
IA_OVERVIEW_PROJECT

Figure 1-37   Create a project 3/4



Specify Project Details

Use this tab to set details about your project.

Project Type:
Information Analyzer
Name: *
IA_OVERVIEW_PROJECT
Description:
IA_OVERVIEW_PROJECT
Owner:
admin admin
Primary Contact:
admin admin

Figure 1-38   Create a project 4/4

1.6.7 SETUPSTEP7: Configure the project

In this step, you associate imported metadata with the created and opened project, modify a project's and data source's analysis settings if required, and set up security for the project.

Associate metadata

The associated metadata determines the data sources that can be analyzed in that project. This process of associating metadata with a project is also called registering the metadata to the project.

You must have Information Analyzer Project Administrator authority to perform this task.[15] You associate metadata with your project at the data store, schema, table, file, or column level. When you associate metadata with your project, you choose the information that you need to create analysis jobs, to run analysis jobs, and to subsequently review analytical results. You register metadata to your project at the data store, schema, table, or file level. This establishes the analytical structures in the repository used for associating and storing analytical results for those identified schemas, tables, and columns. It also allows you to reduce or control the amount of data presented to other project users. You can create multiple analysis projects for each object and register interest in data sources on different host computers, or you can create multiple data stores on the same host computer.

Figure 1-39 on page 74 through Figure 1-44 on page 77 show the association of the metadata of IA_SAMPLE tables with the IA_OVERVIEW_PROJECT as follows:

1. On the Overview navigator menu in the console for the selected project (IA_OVERVIEW_PROJECT), select **Project Properties** as shown in Figure 1-39 on page 74.
2. Select the Data Sources tab, and click **Please add data sources** as shown in Figure 1-40 on page 74 to identify the data sources that you want to associate with your project.
3. In the Select Data Sources window, expand the hierarchical tree to view schemas, directories, tables, files, and columns as shown in Figure 1-41 on page 75 and Figure 1-42 on page 75. Highlight both data stores (IA_SAMPLE_ALT and IA_SAMPLE) and click **OK** to perform the association.

> **Note:** You can also to choose to select one or more schemas, tables, files, or columns that you want to associate with this project. You can select multiple items by pressing Shift+Click or Ctrl+Click.

[15] See 1.6.5, "SETUPSTEP5: Import metadata" on page 61 for details about this privilege.

4. In the Data Sources tab, click **Save All** as shown in Figure 1-43 on page 76. The detailed list of all data sources associated with the IA_SAMPLE_OVERVIEW project is shown in Figure 1-44 on page 77.

After metadata has been associated with a project, you can proceed to modify the project’s analysis settings as described in “Modify the project’s and data source’s analysis settings if required” on page 77.

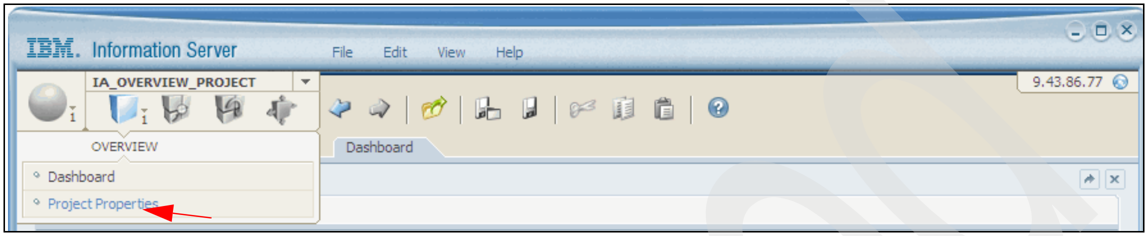
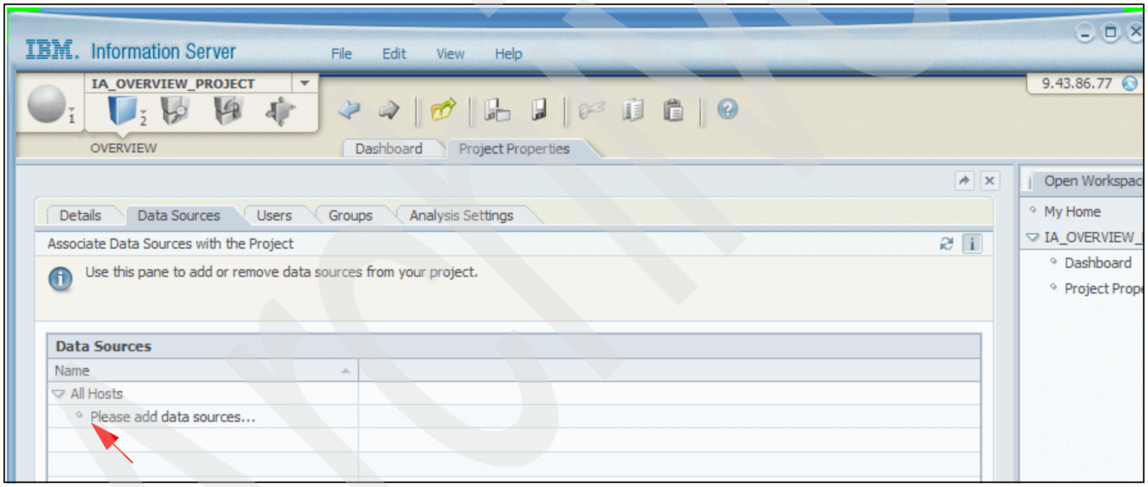


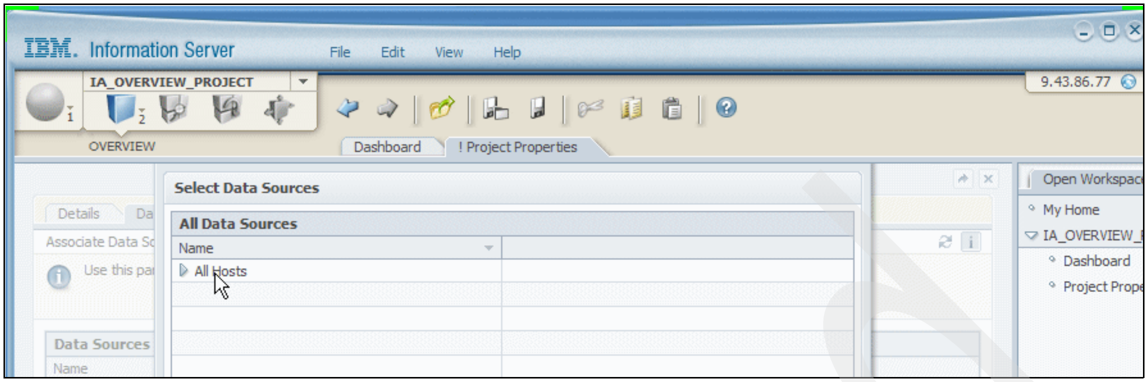
Figure 1-39 Associate a metadata with a project 1/6



Associate Data Sources with the Project

Use this pane to add or remove data sources from your project.

Figure 1-40 Associate a metadata with a project 2/6

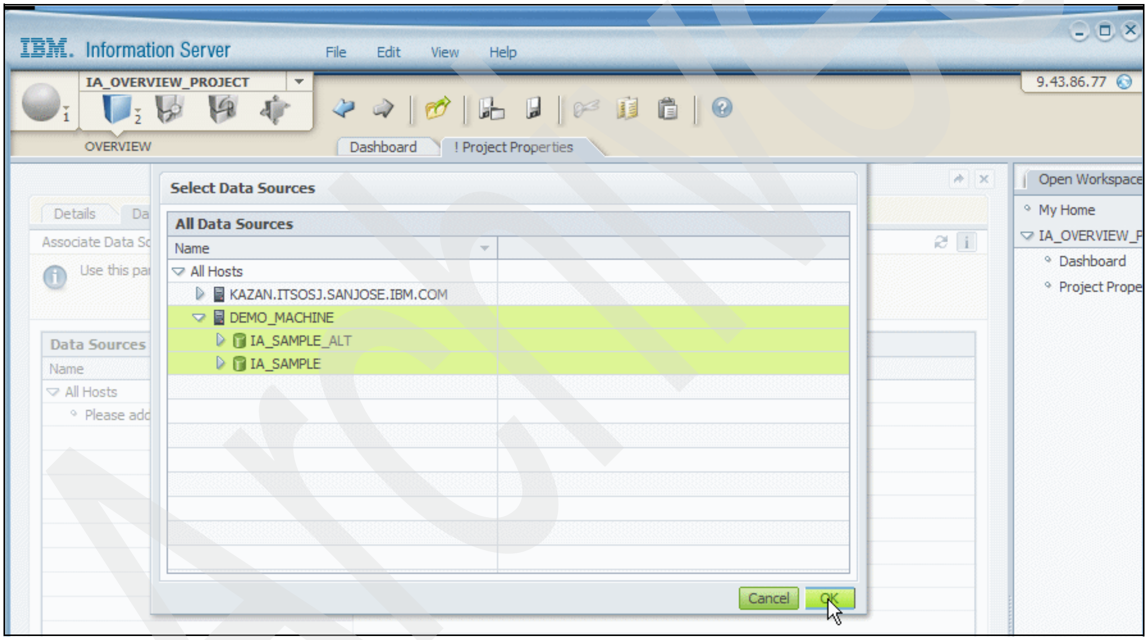


Select Data Sources

All Data Sources

Name
All Hosts

Figure 1-41   Associate a metadata with a project 3/6

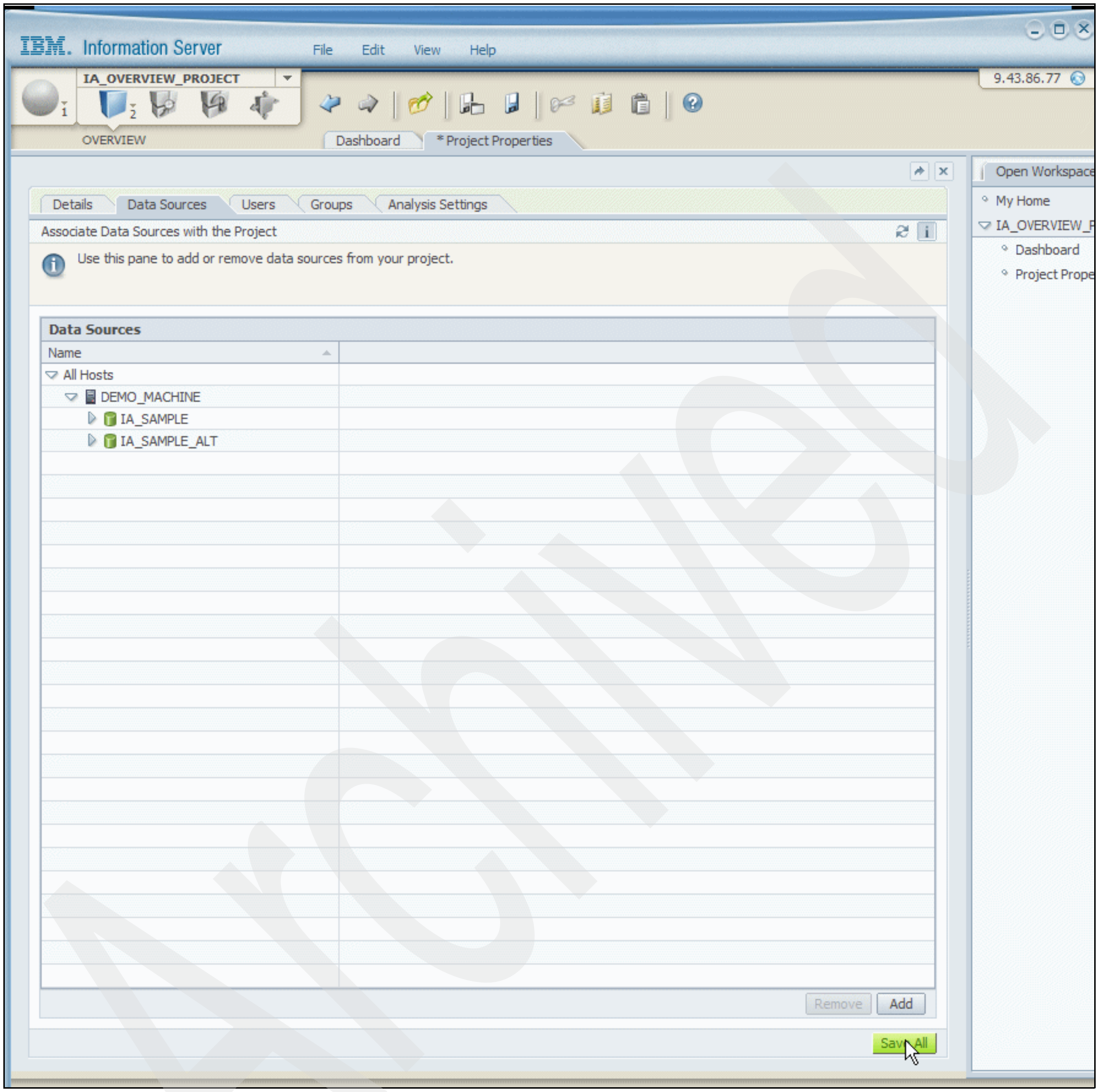


Select Data Sources

All Data Sources

Name
All Hosts
KAZAN.ITSOSJ.SANJOSE.IBM.COM
DEMO_MACHINE
IA_SAMPLE_ALT
IA_SAMPLE

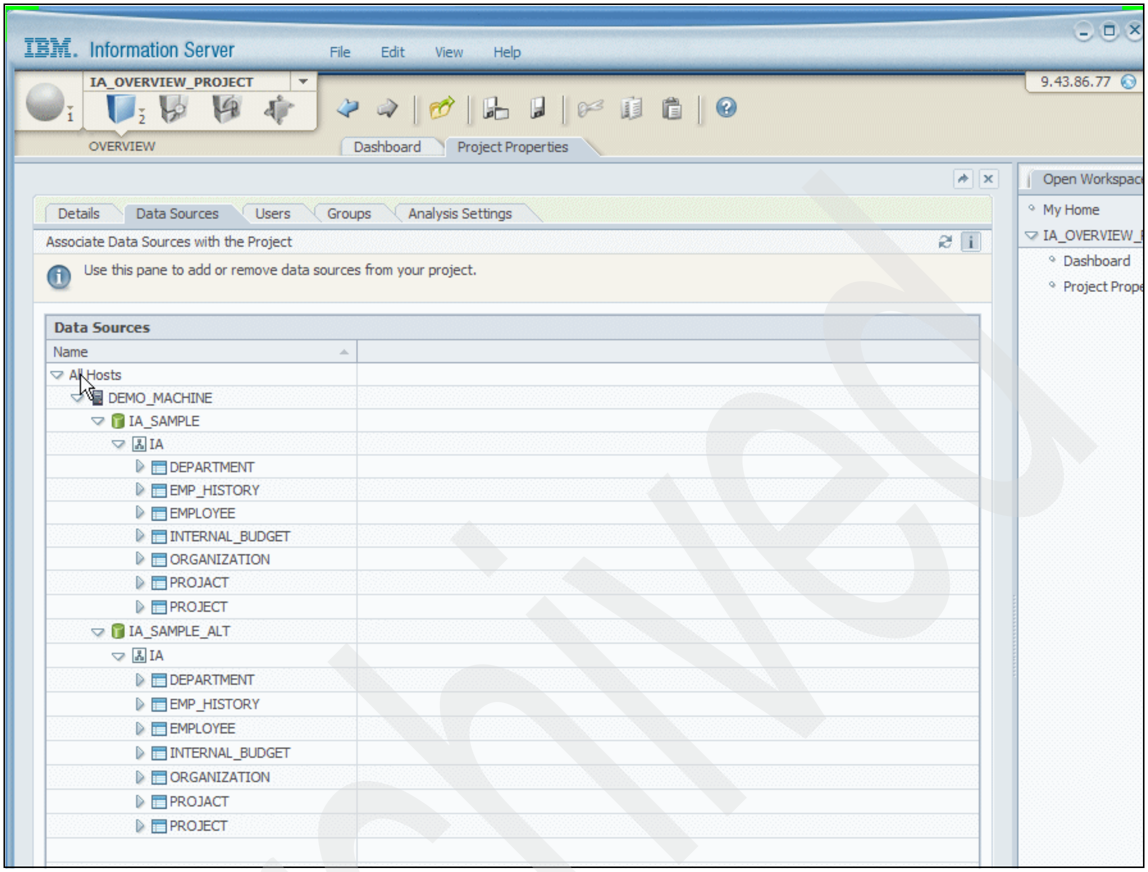
Figure 1-42   Associate a metadata with a project 4/6



Associate Data Sources with the Project

Use this pane to add or remove data sources from your project.

Figure 1-43 Associate a metadata with a project 5/6



Use this pane to add or remove data sources from your project.

Figure 1-44 Associate a metadata with a project 6/6

Modify the project’s and data source’s analysis settings if required

You can fine-tune the parameters, settings, and thresholds for the various analysis jobs. When you modify the project and data source analysis settings, you override the default system settings (in the Analysis Settings workspace) that we described in 1.6.3, “SETUPSTEP3: Optionally, configure Analysis Settings” on page 44. All new analysis jobs that are created in the project inherit the new settings. You can further modify the individual settings for each piece of analysis such as column, table, and cross-table that override the project settings. By modifying analysis options, you are typically tightening or loosening the capability of the system to make its analytical inferences.

You must have Information Analyzer Product Administrator authority to perform this task.

Figure 1-45 on page 79 describes the modification of a project’s analysis settings as follows:

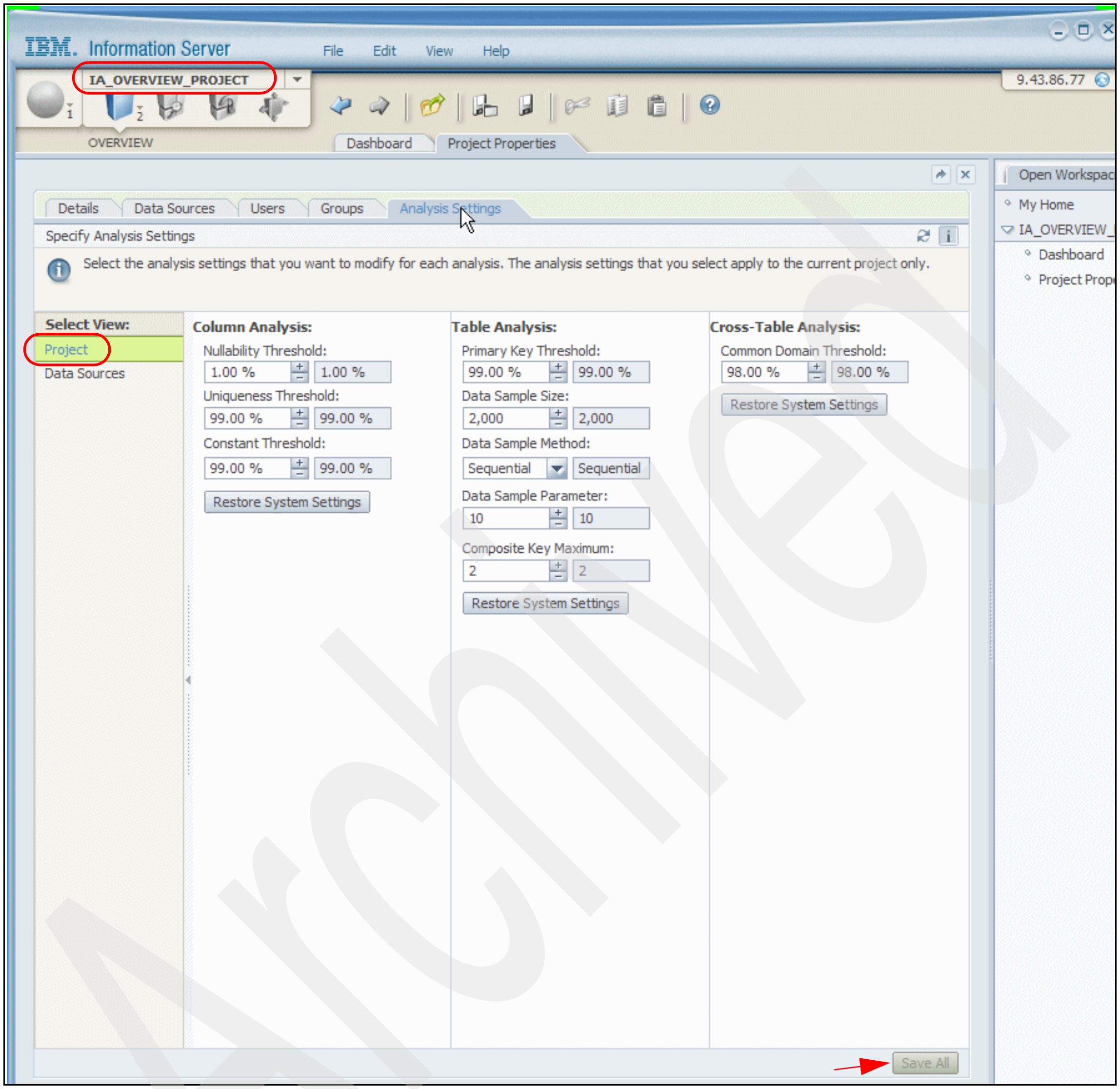
1. On the Overview navigator menu in the console for the selected project (IA_OVERVIEW_PROJECT), select **Project Properties** as shown in Figure 1-39 on page 74.
2. Select the Analysis Settings tab, and Project on the Select View pane as shown in Figure 1-45 on page 79. Modify the Column Analysis, Table Analysis, and Cross-Table Analysis settings as required, and click **Save All** to save the changes made as shown in Figure 1-45 on page 79.

Figure 1-46 on page 80 through Figure 1-53 on page 85 describe the modification of a data source’s analysis settings. When you modify the data source settings, you override the system settings.

1. On the Overview navigator menu in the console for the selected project (IA_OVERVIEW_PROJECT), select **Project Properties** as shown in Figure 1-39 on page 74.
2. Select the Analysis Settings tab, and Data Sources on the Select View pane. Figure 1-46 on page 80 through Figure 1-50 on page 82 show the various analysis settings (scrolling right—scroll bar is not shown here) that can be modified for a given data source.

   You can restore the project settings to their original values by clicking the **Restore Project Settings** as shown in Figure 1-50 on page 82.
3. Click the data source that you want to modify, and click **Modify** as shown in Figure 1-51 on page 83. In the Analysis Settings window, specify the new value (Composite Key Maximum was changed to 7 in our case), and click **OK** as shown in Figure 1-52 on page 84.
4. Click **Save All** in the Analysis Settings tab to confirm the changes as shown in Figure 1-53 on page 85.

After modifying the project’s and data source’s analysis settings, you can proceed to set up security for the project based on user authentication and role identification as described in “Assigning users/groups to a project and assigning roles” on page 86.



Specify Analysis Settings

Select the analysis settings that you want to modify for each analysis. The analysis settings that you select apply to the current project only.

Select View:
- Project
- Data Sources

Column Analysis:

Nullability Threshold: 1.00 % | 1.00 %

Uniqueness Threshold: 99.00 % | 99.00 %

Constant Threshold: 99.00 % | 99.00 %

Table Analysis:

Primary Key Threshold: 99.00 % | 99.00 %

Data Sample Size: 2,000 | 2,000

Data Sample Method: Sequential | Sequential

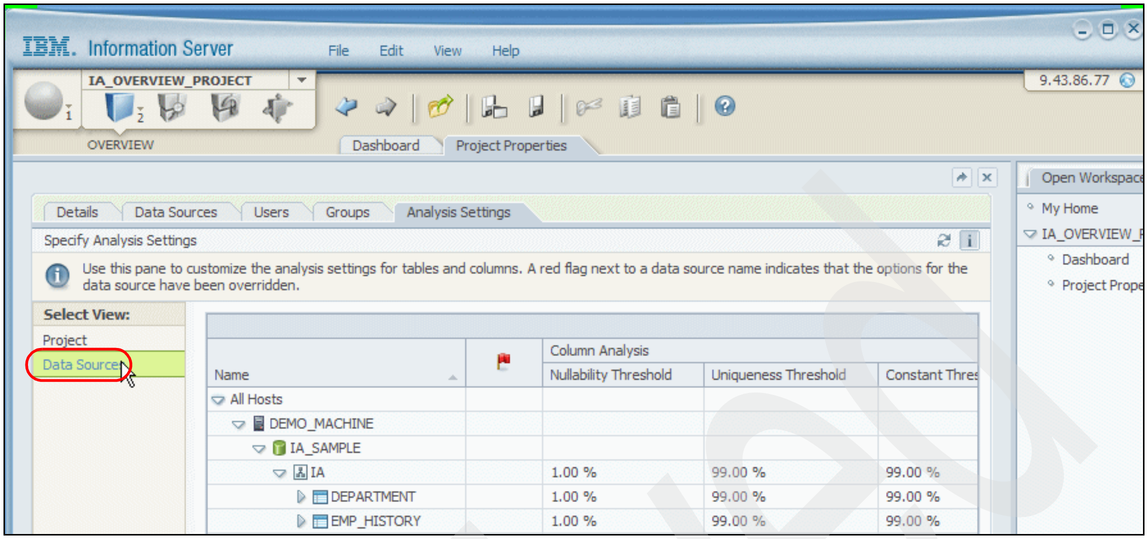
Data Sample Parameter: 10 | 10

Composite Key Maximum: 2 | 2

Cross-Table Analysis:

Common Domain Threshold: 98.00 % | 98.00 %

Figure 1-45 Modify a project's analysis settings

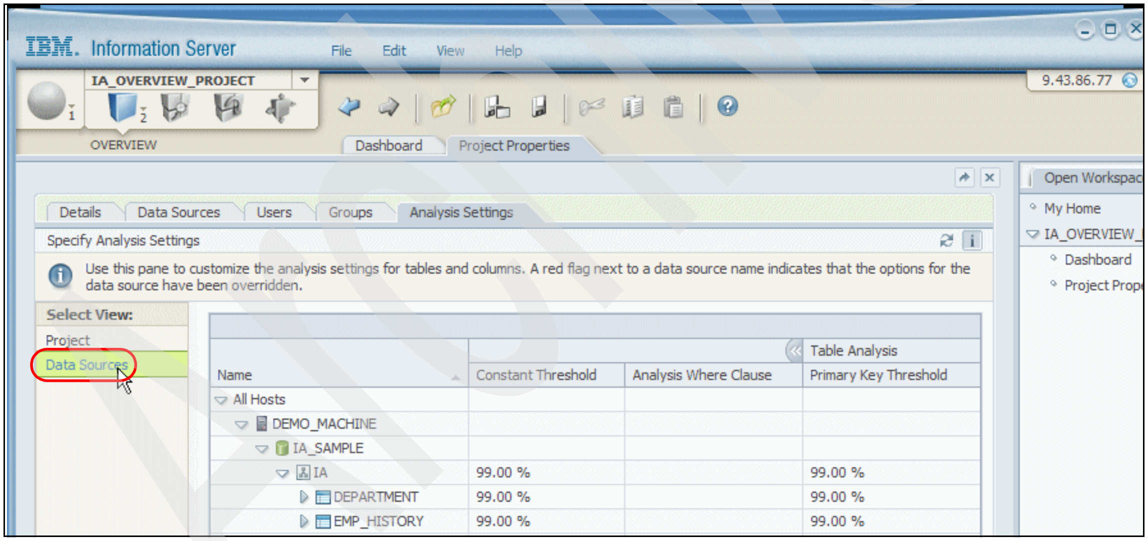


Specify Analysis Settings

Use this pane to customize the analysis settings for tables and columns. A red flag next to a data source name indicates that the options for the data source have been overridden.

		Column Analysis		
Name	🚩	Nullability Threshold	Uniqueness Threshold	Constant Thres
All Hosts				
DEMO_MACHINE				
IA_SAMPLE				
IA		1.00 %	99.00 %	99.00 %
DEPARTMENT		1.00 %	99.00 %	99.00 %
EMP_HISTORY		1.00 %	99.00 %	99.00 %

Figure 1-46 Modify a data source's analysis settings 1/8

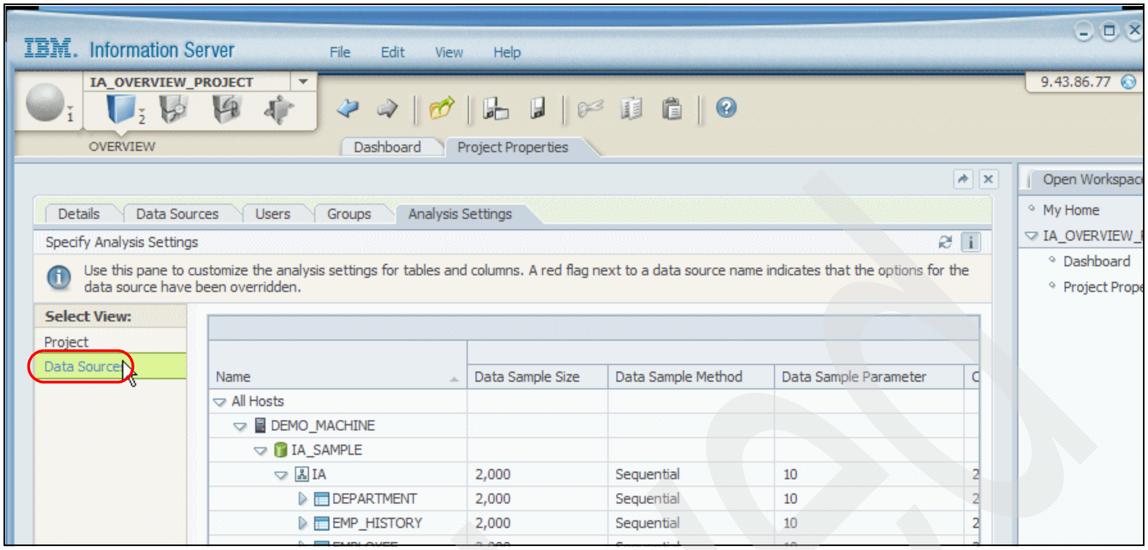


Specify Analysis Settings

Use this pane to customize the analysis settings for tables and columns. A red flag next to a data source name indicates that the options for the data source have been overridden.

			Table Analysis
Name	Constant Threshold	Analysis Where Clause	Primary Key Threshold
All Hosts			
DEMO_MACHINE			
IA_SAMPLE			
IA	99.00 %		99.00 %
DEPARTMENT	99.00 %		99.00 %
EMP_HISTORY	99.00 %		99.00 %

Figure 1-47 Modify a data source's analysis settings 2/8

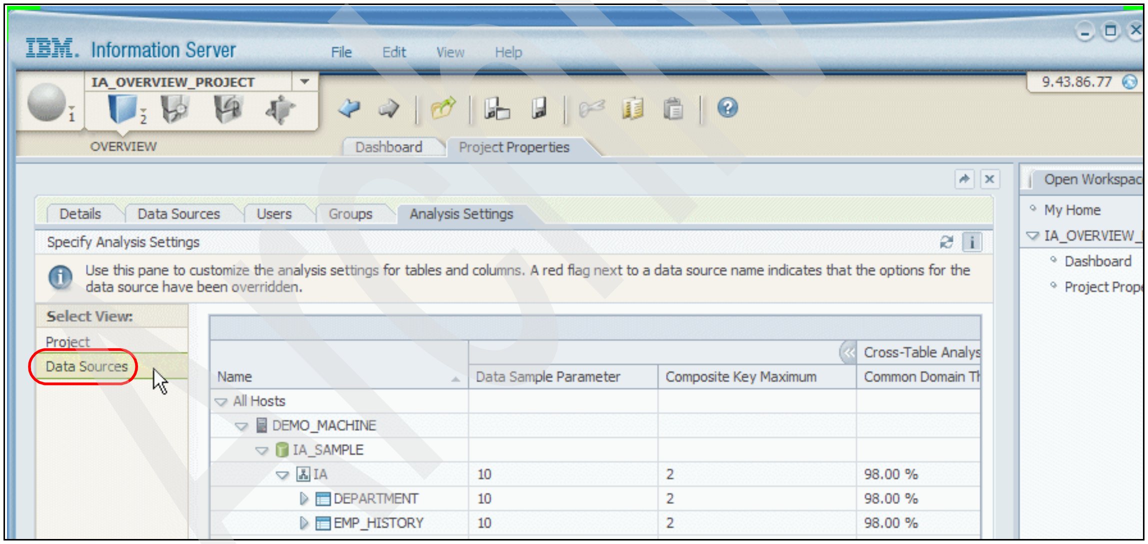


Specify Analysis Settings

Use this pane to customize the analysis settings for tables and columns. A red flag next to a data source name indicates that the options for the data source have been overridden.

Name	Data Sample Size	Data Sample Method	Data Sample Parameter	C
All Hosts				
DEMO_MACHINE				
IA_SAMPLE				
IA	2,000	Sequential	10	2
DEPARTMENT	2,000	Sequential	10	2
EMP_HISTORY	2,000	Sequential	10	2

Figure 1-48 Modify a data source's analysis settings 3/8

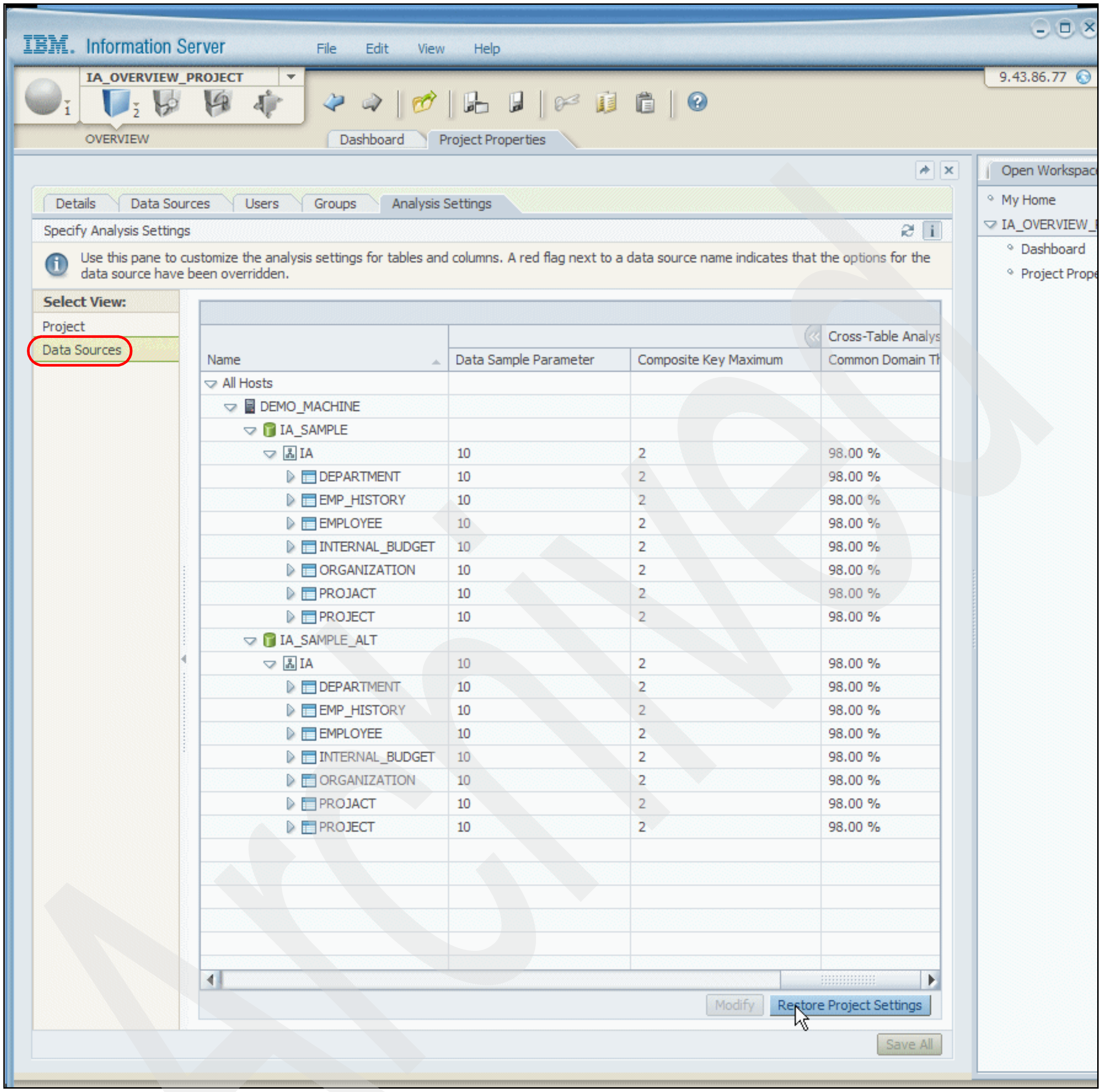


Specify Analysis Settings

Use this pane to customize the analysis settings for tables and columns. A red flag next to a data source name indicates that the options for the data source have been overridden.

			Cross-Table Analys
Name	Data Sample Parameter	Composite Key Maximum	Common Domain Th
All Hosts			
DEMO_MACHINE			
IA_SAMPLE			
IA	10	2	98.00 %
DEPARTMENT	10	2	98.00 %
EMP_HISTORY	10	2	98.00 %

Figure 1-49 Modify a data source's analysis settings 4/8

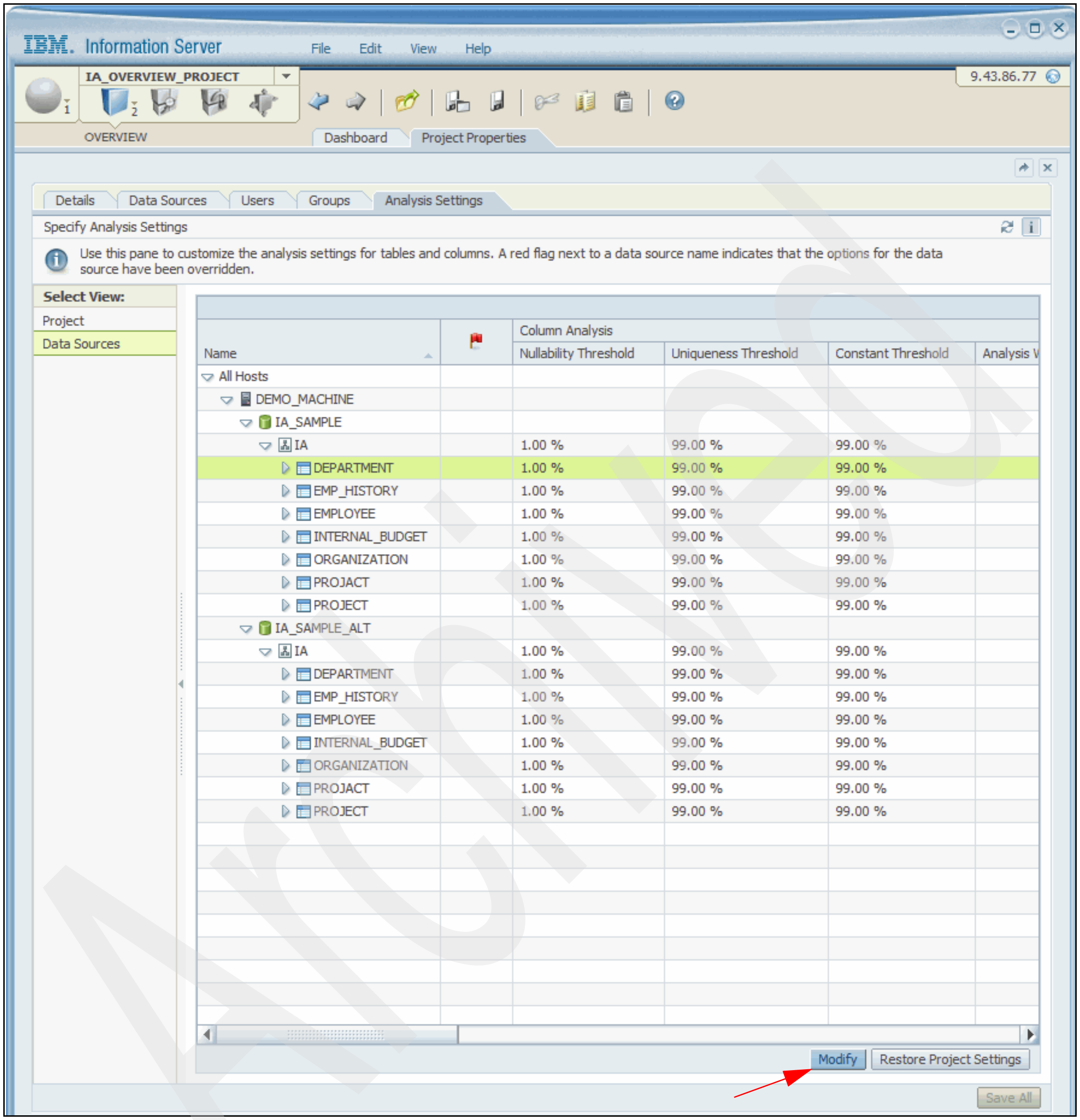


Specify Analysis Settings

Use this pane to customize the analysis settings for tables and columns. A red flag next to a data source name indicates that the options for the data source have been overridden.

			Cross-Table Analys
Name	Data Sample Parameter	Composite Key Maximum	Common Domain Th
All Hosts			
DEMO_MACHINE			
IA_SAMPLE			
IA	10	2	98.00 %
DEPARTMENT	10	2	98.00 %
EMP_HISTORY	10	2	98.00 %
EMPLOYEE	10	2	98.00 %
INTERNAL_BUDGET	10	2	98.00 %
ORGANIZATION	10	2	98.00 %
PROJACT	10	2	98.00 %
PROJECT	10	2	98.00 %
IA_SAMPLE_ALT			
IA	10	2	98.00 %
DEPARTMENT	10	2	98.00 %
EMP_HISTORY	10	2	98.00 %
EMPLOYEE	10	2	98.00 %
INTERNAL_BUDGET	10	2	98.00 %
ORGANIZATION	10	2	98.00 %
PROJACT	10	2	98.00 %
PROJECT	10	2	98.00 %

Figure 1-50 Modify a data source's analysis settings 5/8

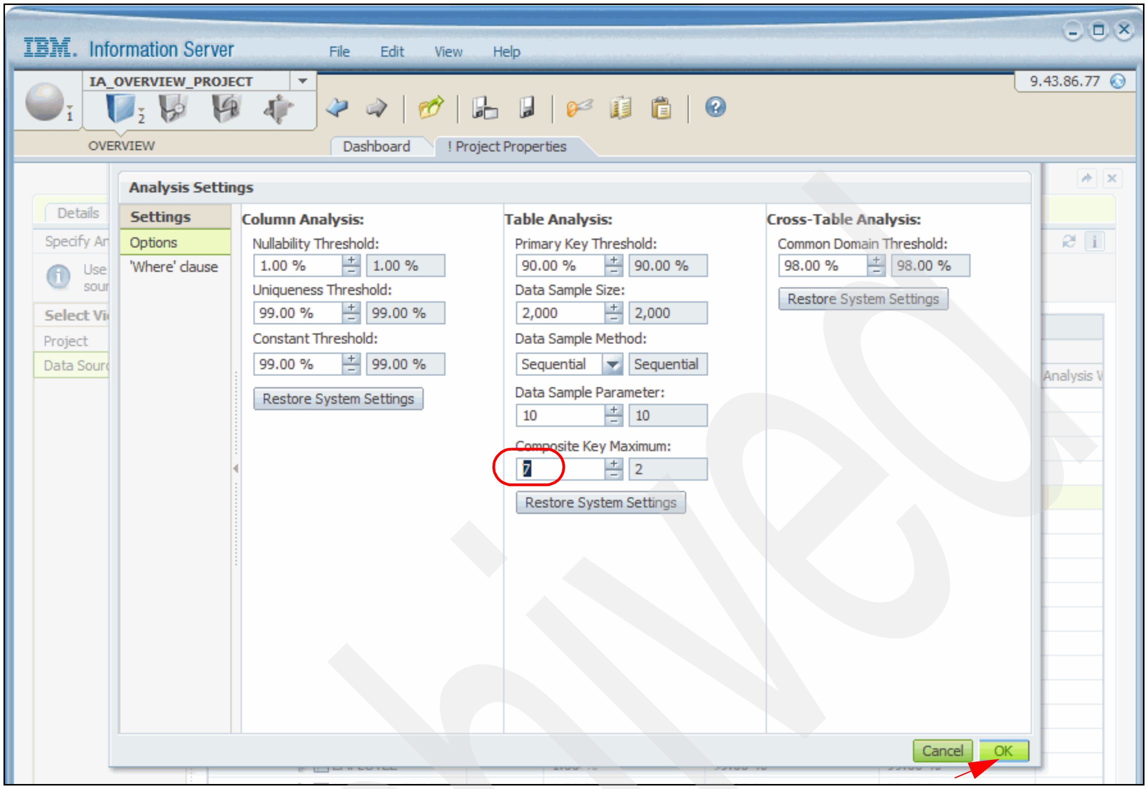


Specify Analysis Settings

Use this pane to customize the analysis settings for tables and columns. A red flag next to a data source name indicates that the options for the data source have been overridden.

Name		Column Analysis			
		Nullability Threshold	Uniqueness Threshold	Constant Threshold	Analysis V
All Hosts					
DEMO_MACHINE					
IA_SAMPLE					
IA		1.00 %	99.00 %	99.00 %	
DEPARTMENT		1.00 %	99.00 %	99.00 %	
EMP_HISTORY		1.00 %	99.00 %	99.00 %	
EMPLOYEE		1.00 %	99.00 %	99.00 %	
INTERNAL_BUDGET		1.00 %	99.00 %	99.00 %	
ORGANIZATION		1.00 %	99.00 %	99.00 %	
PROJACT		1.00 %	99.00 %	99.00 %	
PROJECT		1.00 %	99.00 %	99.00 %	
IA_SAMPLE_ALT					
IA		1.00 %	99.00 %	99.00 %	
DEPARTMENT		1.00 %	99.00 %	99.00 %	
EMP_HISTORY		1.00 %	99.00 %	99.00 %	
EMPLOYEE		1.00 %	99.00 %	99.00 %	
INTERNAL_BUDGET		1.00 %	99.00 %	99.00 %	
ORGANIZATION		1.00 %	99.00 %	99.00 %	
PROJACT		1.00 %	99.00 %	99.00 %	
PROJECT		1.00 %	99.00 %	99.00 %	

Figure 1-51 Modify a data source's analysis settings 6/8



Analysis Settings

Column Analysis:

Nullability Threshold: 1.00 % 1.00 %

Uniqueness Threshold: 99.00 % 99.00 %

Constant Threshold: 99.00 % 99.00 %

Table Analysis:

Primary Key Threshold: 90.00 % 90.00 %

Data Sample Size: 2,000 2,000

Data Sample Method: Sequential Sequential

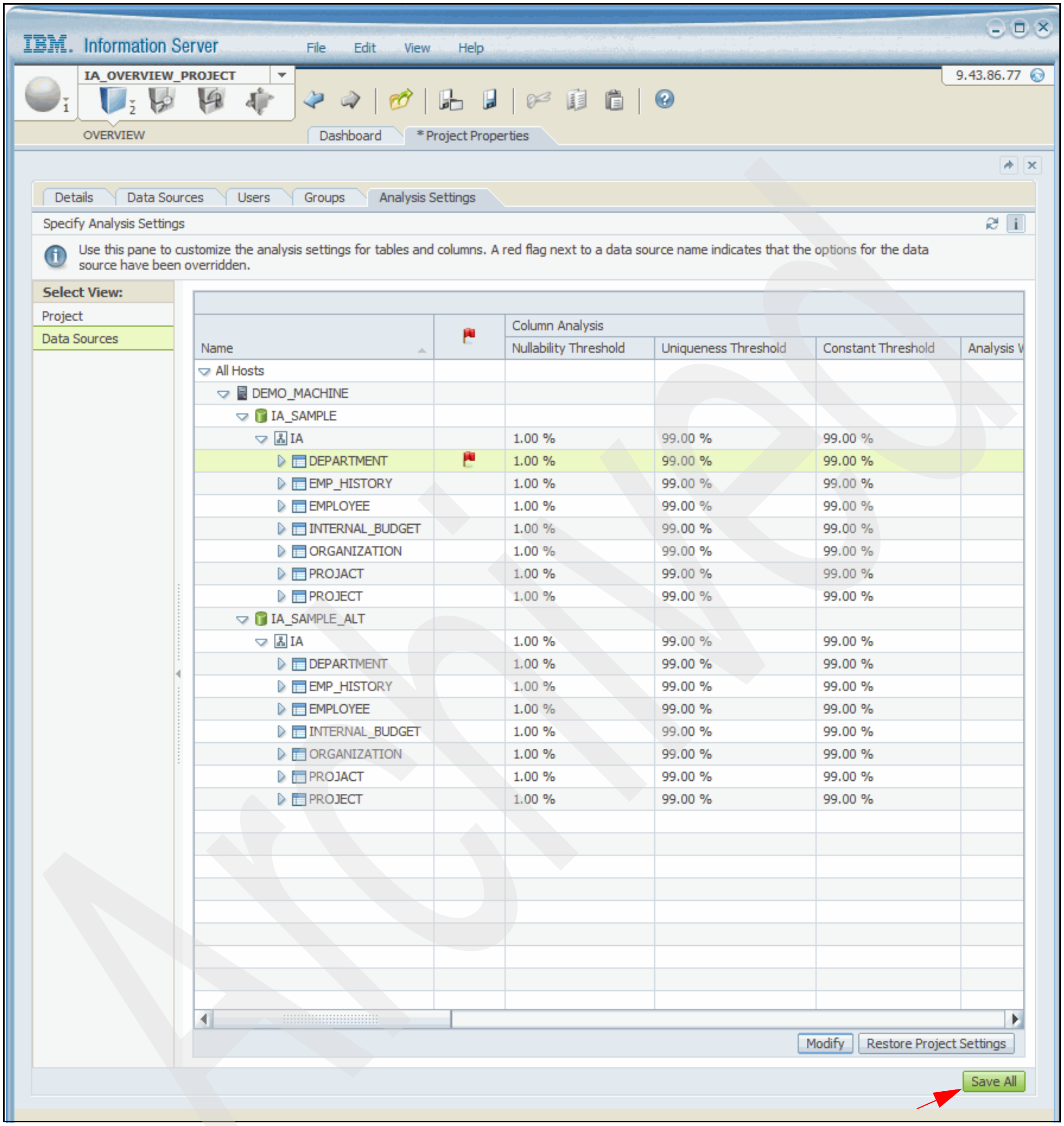
Data Sample Parameter: 10 10

Composite Key Maximum: 7 2

Cross-Table Analysis:

Common Domain Threshold: 98.00 % 98.00 %

Figure 1-52 Modify a data source's analysis settings 7/8



Specify Analysis Settings

Use this pane to customize the analysis settings for tables and columns. A red flag next to a data source name indicates that the options for the data source have been overridden.

Select View:
Project
Data Sources

		Column Analysis			
Name	🚩	Nullability Threshold	Uniqueness Threshold	Constant Threshold	Analysis V
All Hosts					
DEMO_MACHINE					
IA_SAMPLE					
IA		1.00 %	99.00 %	99.00 %	
DEPARTMENT	🚩	1.00 %	99.00 %	99.00 %	
EMP_HISTORY		1.00 %	99.00 %	99.00 %	
EMPLOYEE		1.00 %	99.00 %	99.00 %	
INTERNAL_BUDGET		1.00 %	99.00 %	99.00 %	
ORGANIZATION		1.00 %	99.00 %	99.00 %	
PROJACT		1.00 %	99.00 %	99.00 %	
PROJECT		1.00 %	99.00 %	99.00 %	
IA_SAMPLE_ALT					
IA		1.00 %	99.00 %	99.00 %	
DEPARTMENT		1.00 %	99.00 %	99.00 %	
EMP_HISTORY		1.00 %	99.00 %	99.00 %	
EMPLOYEE		1.00 %	99.00 %	99.00 %	
INTERNAL_BUDGET		1.00 %	99.00 %	99.00 %	
ORGANIZATION		1.00 %	99.00 %	99.00 %	
PROJACT		1.00 %	99.00 %	99.00 %	
PROJECT		1.00 %	99.00 %	99.00 %	

Figure 1-53 Modify a data source's analysis settings 8/8

Assigning users/groups to a project and assigning roles

As mentioned earlier, to create a secure project environment, you can define a security policy that is based on user authentication and role identification. The IBM Information Server administrator creates a security policy by performing the following tasks:

1. Creating a user ID for each person who needs to access the suite.
2. Assigning suite and suite component roles to each user.
3. Assigning each user to specific projects within the suite component.
4. Assigning each user roles in that project.

We describe the roles that are in 1.6.1, "SETUPSTEP1: Set up the various roles" on page 32.

After you create a project, you can specify which users and groups can access that project and the roles that these users and groups can assume.

Figure 1-54 on page 89 through Figure 1-61 on page 96 show the assignment of users to a project and the roles that they can assume as follows:

1. On the Overview navigator menu in the console for the selected project (IA_OVERVIEW_PROJECT), select **Project Properties** as shown in Figure 1-39 on page 74.
2. On the Project Properties workspace, select the Users tab, and check the Information Analyzer Data Operator and Information Analyzer Business Analyst in the Project Roles pane. Then click **Browse** in the Users pane as shown in Figure 1-54 on page 89 to add users to the project.
3. On the Add Users window, select the users that you want to add to the project. Select click **Add**, then click **OK** as shown in Figure 1-55 on page 90 and Figure 1-56 on page 91.

Note: The list of users displayed in Figure 1-55 on page 90 comes from the list of users in the Linux operating system where IBM Information Server is installed. The option to select operating system (OS) users was chosen during IBM Information Server installation as shown in Figure 1-62 on page 97.

The setup using OS-level users can be an advantage in a IBM WebSphere DataStage environment where IBM Information Server users need operating system access, while the internal user registry of IBM Information Server would be easier to administer when no OS access is required.

Our fictitious organization had the following users assigned roles as follows:

- Tom Jon as the Suite Administrator and Information Analyzer Data Administrator. He defines data stores and imports metadata. He can also add data stores to projects, but cannot run analyses or see analyses results. However, he can grant himself authorization for these tasks. He was not assigned the Suite User role since we did not want Tom to be able to log on to any of the suite tools or component tools.
- Jorge Nunes as the Information Analyzer Project Administrator and Suite User. He creates projects, adds data stores to projects, and authorizes project users.
- Per Hansen as the Information Analyzer Data Administrator and Suite User.
- Shane Kelly as Suite User and Information Analyzer User.

The setup of Local OS User Registry users and Internal User Registry users and assignment of roles to them is established from the IBM Information Server Web Console as follows:

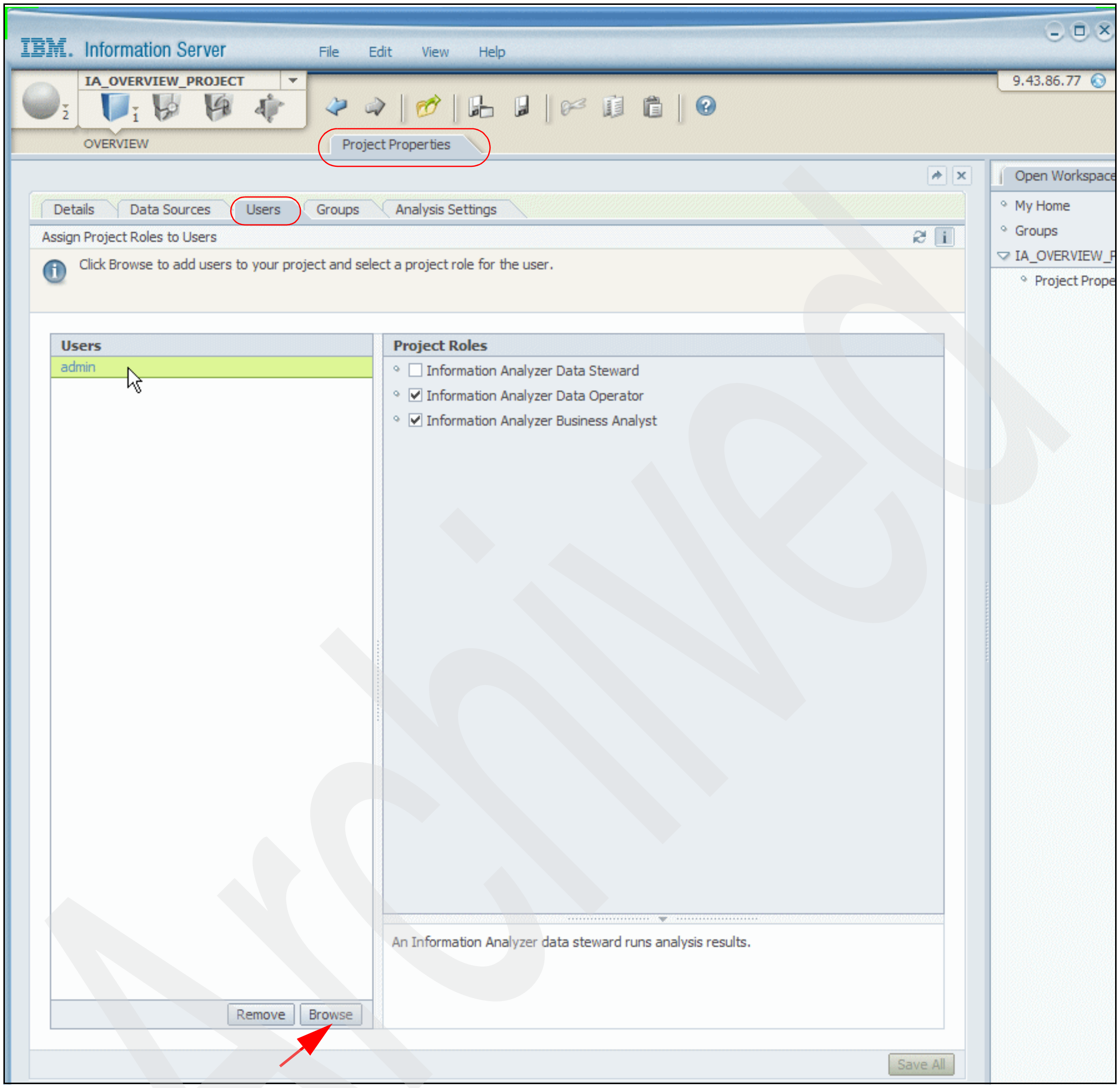
- Figure 1-63 on page 97 through Figure 1-68 on page 102 show the assignment of IBM Information Server and IBM WebSphere Information Analyzer roles to imported operating system level users. Because we did not want Tom to be able to log on to any of the suite tools or component tools, we click **No** in response to the prompt shown in Figure 1-68 on page 102.
- Figure 1-69 on page 102 through Figure 1-76 on page 109 show the creation of new users in the internal user registry and the assignment of IBM Information Server and IBM WebSphere Information Analyzer roles to them.

4. On the Project Roles pane, select a project role to assign to the selected user. A user can be assigned one or more roles in a project. Click **Save All** as shown in Figure 1-57 on page 92.

The assignment of groups to a project and roles to groups is similar to that of adding users as shown in Figure 1-58 on page 93 through Figure 1-61 on page 96 as follows:

1. On the Overview navigator menu in the console for the selected project (IA_OVERVIEW_PROJECT), select **Project Properties** as shown in Figure 1-39 on page 74.
2. On the Project Properties workspace, select the Groups tab, and click **Browse** in the Groups pane as shown in Figure 1-58 on page 93 to add groups to the project.
3. On the Add Groups window, select the groups (admin in our case) that you want to add to the project, click **Add**, and then click **OK** as shown in Figure 1-59 on page 94 and Figure 1-60 on page 95.
4. On the Project Roles pane, select one or more roles (Information Analyzer Data Operator and Information Analyzer Business Analyst) to assign to the selected group (admin). Click **Save All** to confirm all the changes as shown in Figure 1-61 on page 96.

After assigning users and groups to a project and assigning one or more project roles (such as Information Analyzer Business Analyst, Information Analyzer Business Analyst Data Operator, and Information Analyzer Data Steward as described in Table 1-4 on page 34) to these users and groups, you can now begin the process of information analysis such as column analysis, primary key analysis, foreign key analysis, cross-domain analysis, and baseline analysis.



Assign Project Roles to Users

Click Browse to add users to your project and select a project role for the user.

Users

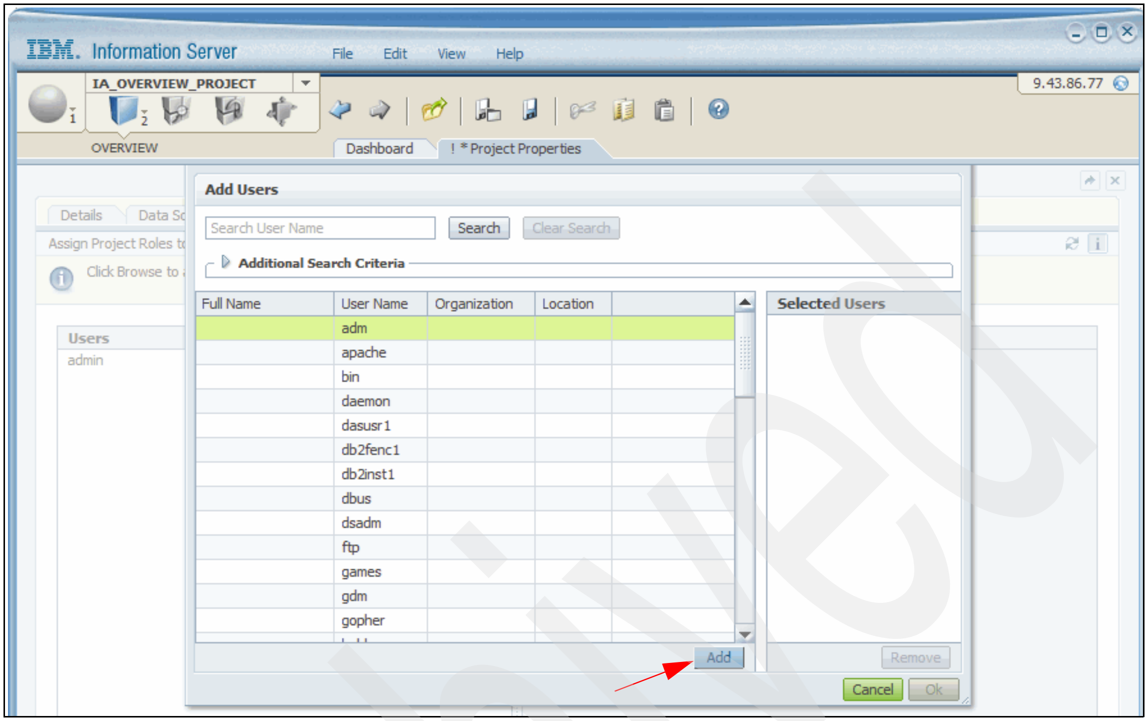
admin

Project Roles

- ☐ Information Analyzer Data Steward
- ☑ Information Analyzer Data Operator
- ☑ Information Analyzer Business Analyst

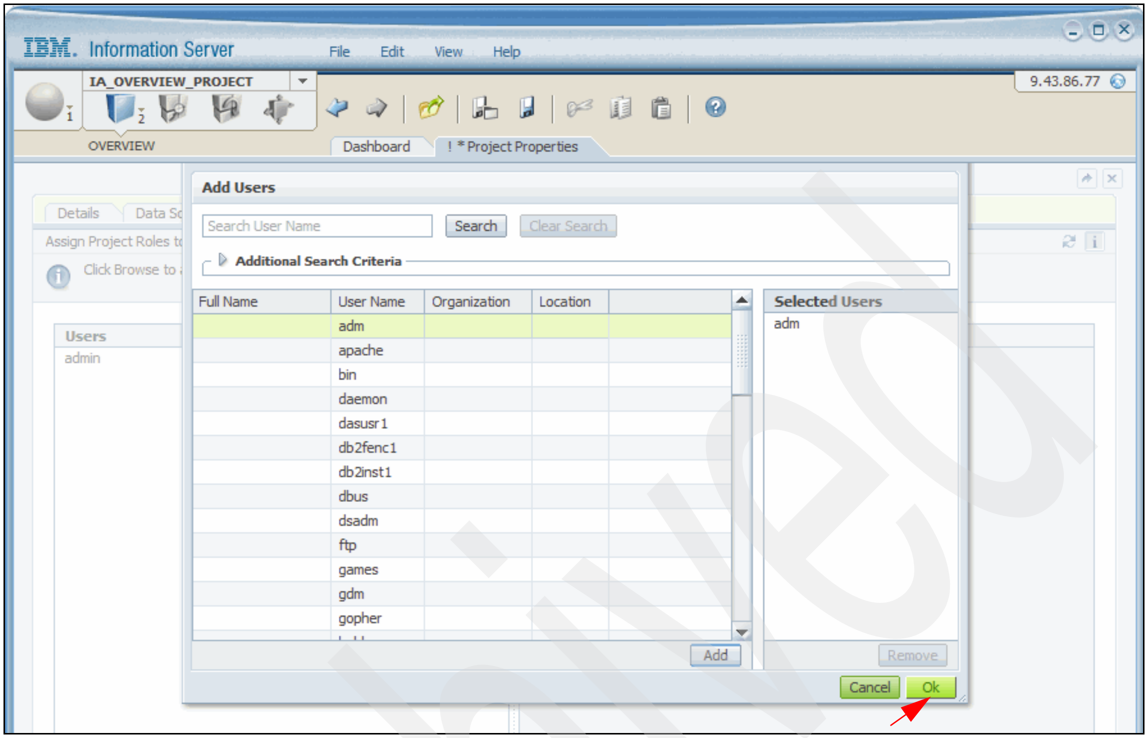
An Information Analyzer data steward runs analysis results.

Figure 1-54 Assign users/groups to a project and assign roles 1/8



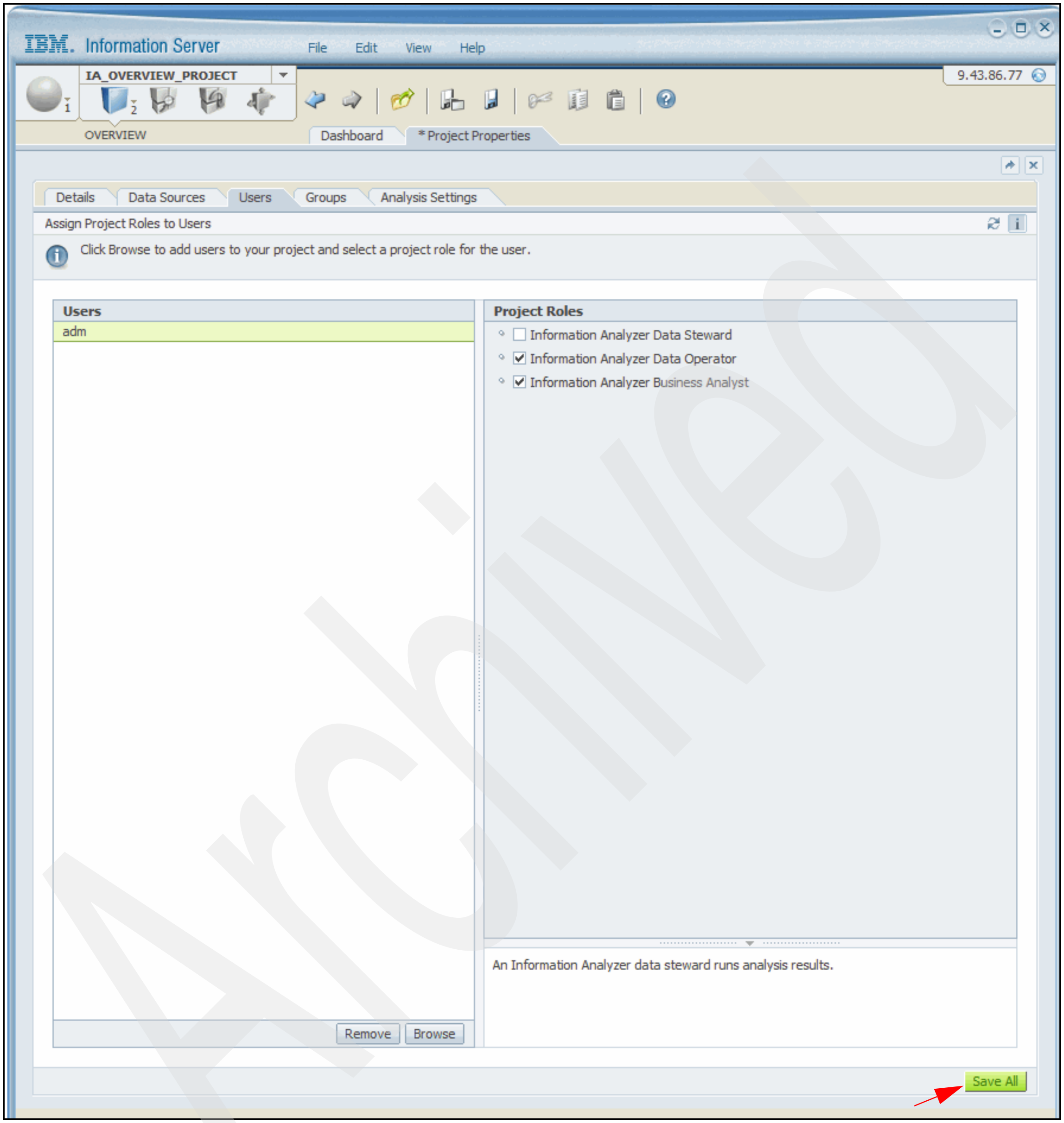
Full Name	User Name	Organization	Location
	adm		
	apache		
	bin		
	daemon		
	dasusr1		
	db2fenc1		
	db2inst1		
	dbus		
	dsadm		
	ftp		
	games		
	gdm		
	gopher		

Figure 1-55 Assign users/groups to a project and assign roles 2/8



Full Name	User Name	Organization	Location
	adm		
	apache		
	bin		
	daemon		
	dasusr1		
	db2fenc1		
	db2inst1		
	dbus		
	dsadm		
	ftp		
	games		
	gdm		
	gopher		

Figure 1-56 Assign users/groups to a project and assign roles 3/8



Assign Project Roles to Users

Click Browse to add users to your project and select a project role for the user.

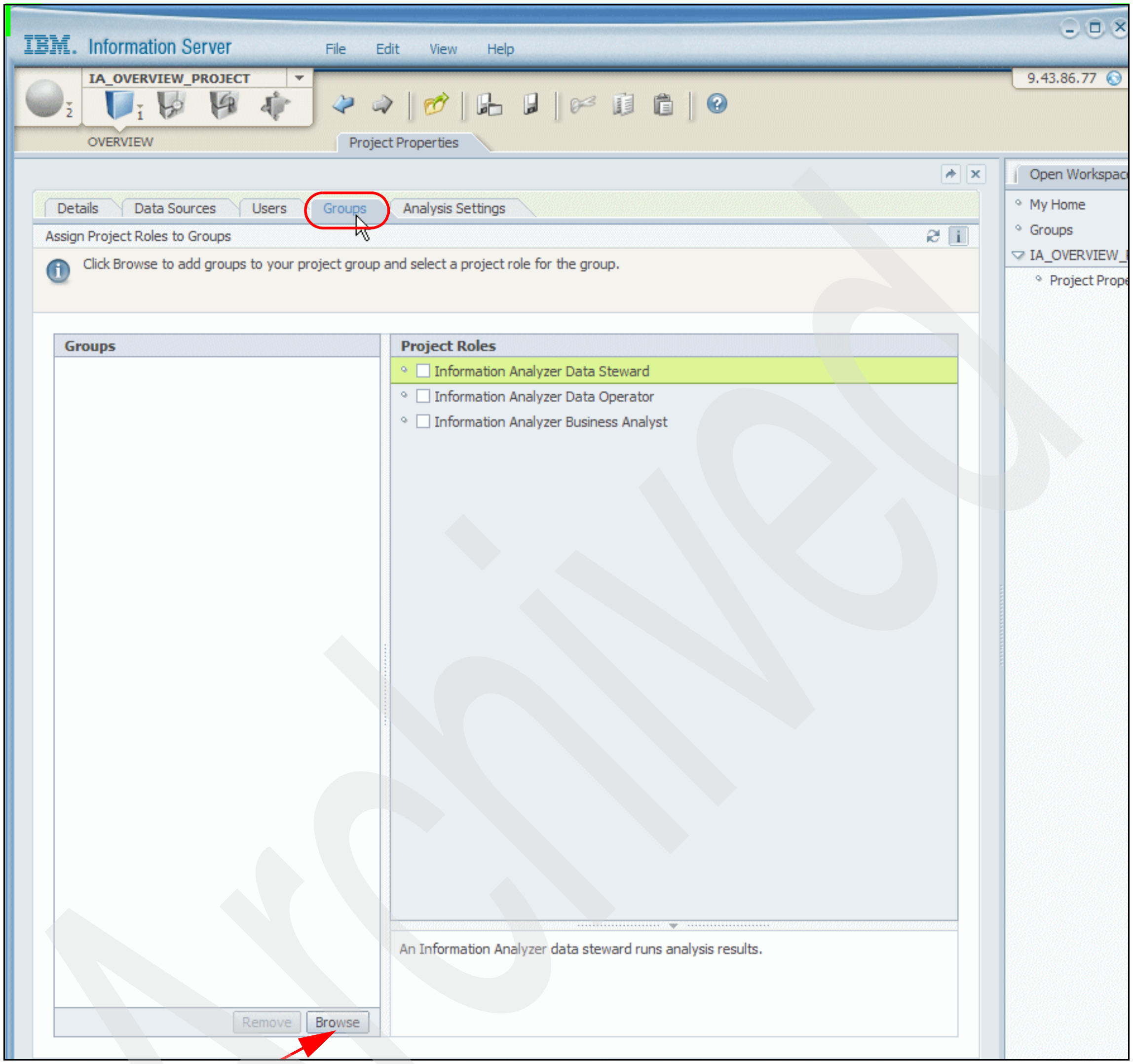
Users: adm

Project Roles:

- ☐ Information Analyzer Data Steward
- ☑ Information Analyzer Data Operator
- ☑ Information Analyzer Business Analyst

An Information Analyzer data steward runs analysis results.

Figure 1-57 Assign users/groups to a project and assign roles 4/8



Assign Project Roles to Groups

Click Browse to add groups to your project group and select a project role for the group.

Groups

Project Roles

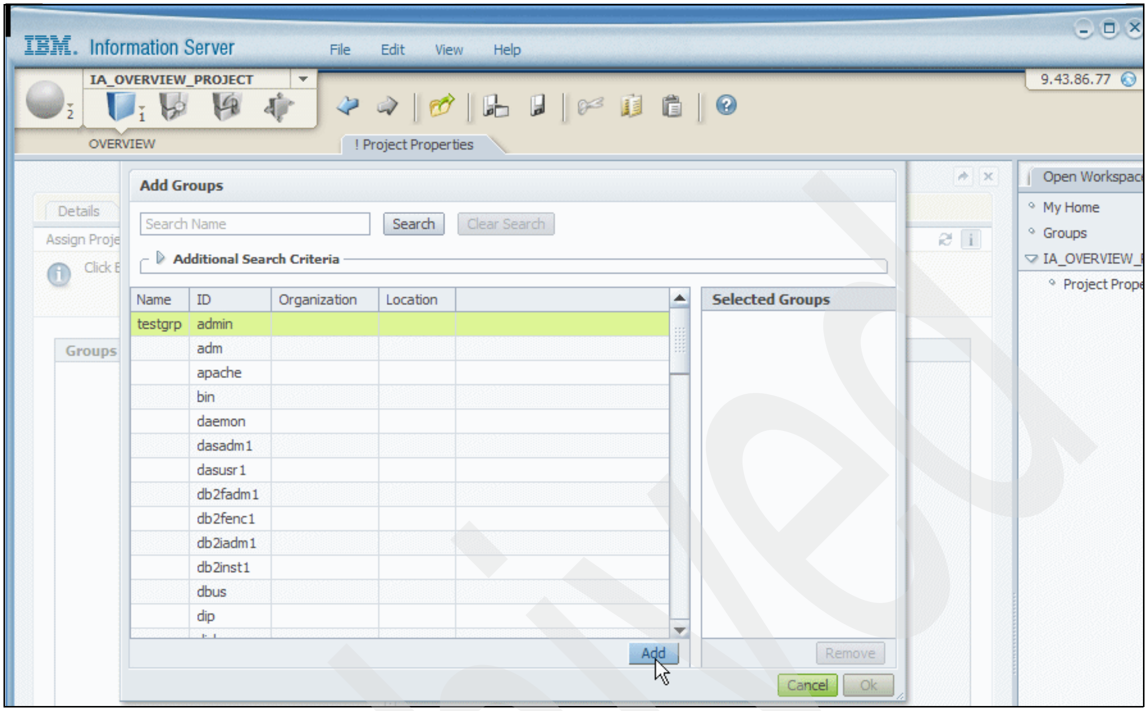
□ Information Analyzer Data Steward

□ Information Analyzer Data Operator

□ Information Analyzer Business Analyst

An Information Analyzer data steward runs analysis results.

Figure 1-58 Assign users/groups to a project and assign roles 5/8



Add Groups

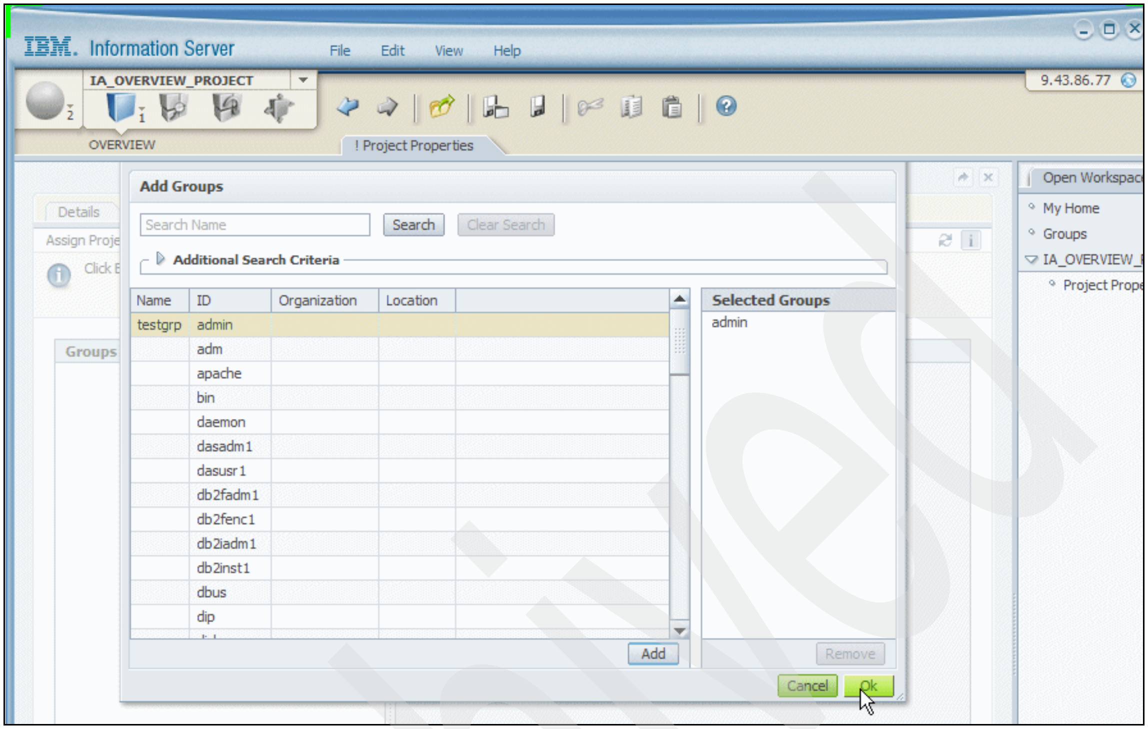
Search Name | Search | Clear Search

Additional Search Criteria

Name	ID	Organization	Location
testgrp	admin		
	adm		
	apache		
	bin		
	daemon		
	dasadm1		
	dasusr1		
	db2fadm1		
	db2fenc1		
	db2iadm1		
	db2inst1		
	dbus		
	dip		

Selected Groups

Figure 1-59 Assign users/groups to a project and roles 6/8

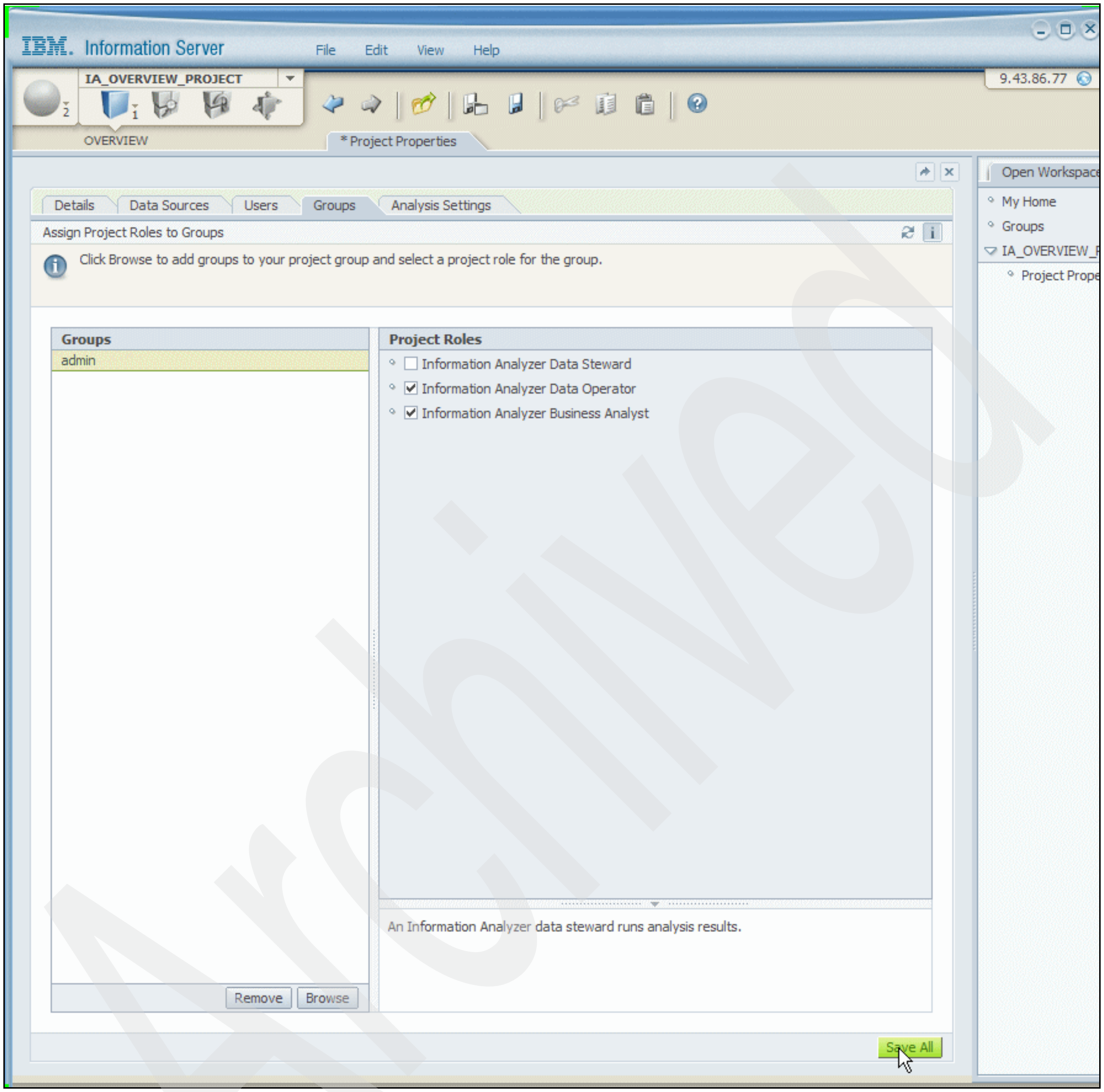


Add Groups

Name	ID	Organization	Location
testgrp	admin		
	adm		
	apache		
	bin		
	daemon		
	dasadm1		
	dasusr1		
	db2fadm1		
	db2fenc1		
	db2iadm1		
	db2inst1		
	dbus		
	dip		

Selected Groups
admin

Figure 1-60 Assign users/groups to a project and roles 7/8



Assign Project Roles to Groups

Click Browse to add groups to your project group and select a project role for the group.

Groups

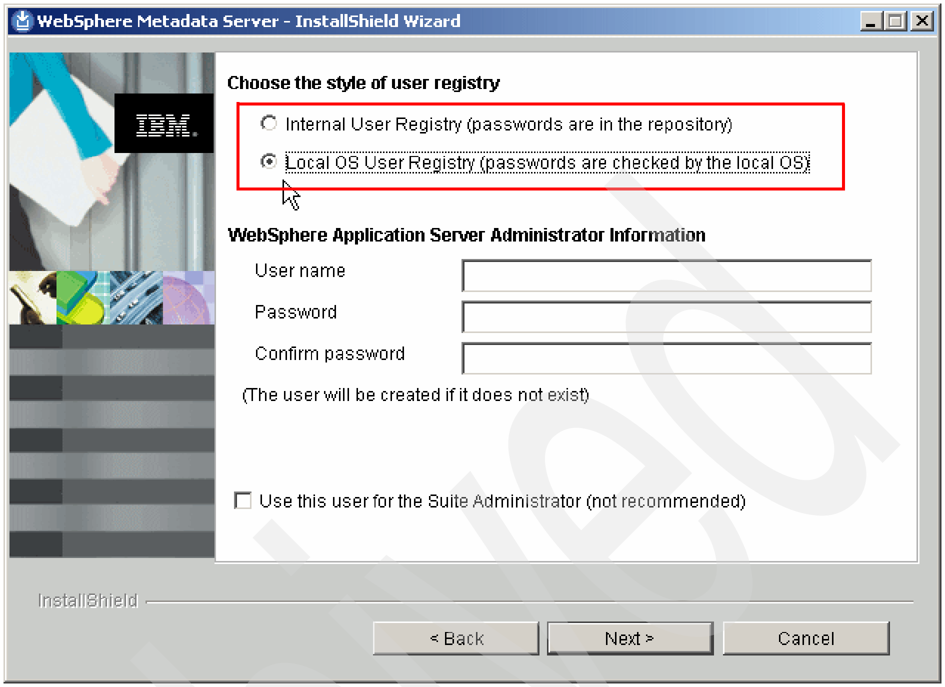
admin

Project Roles

- □ Information Analyzer Data Steward
- ☑ Information Analyzer Data Operator
- ☑ Information Analyzer Business Analyst

An Information Analyzer data steward runs analysis results.

Figure 1-61 Assign users/groups to a project and roles 8/8



WebSphere Metadata Server - InstallShield Wizard

Choose the style of user registry

○ Internal User Registry (passwords are in the repository)

● Local OS User Registry (passwords are checked by the local OS)

WebSphere Application Server Administrator Information

User name

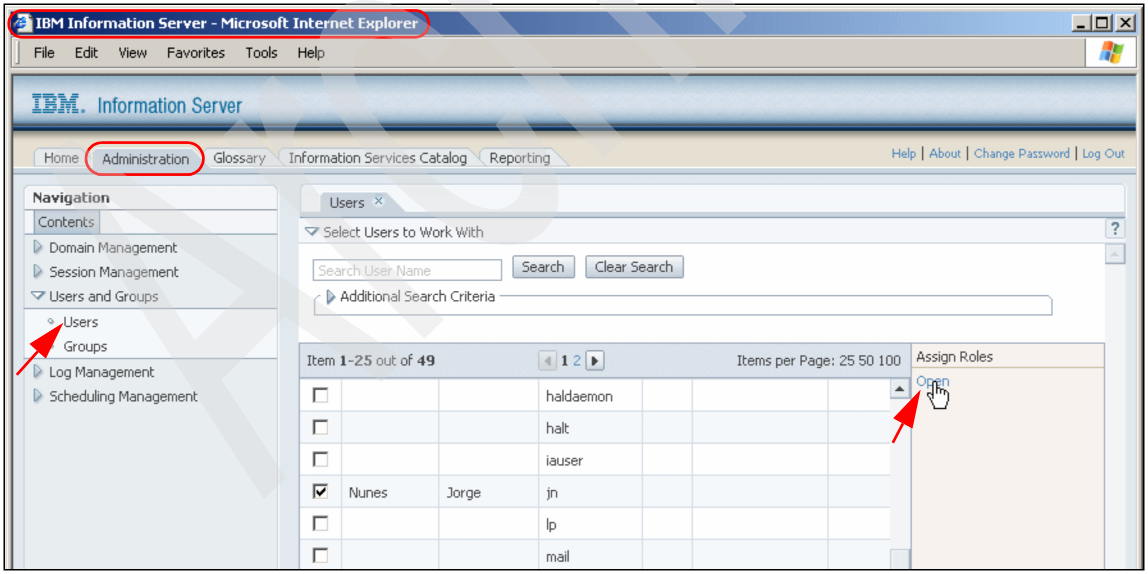
Password

Confirm password

(The user will be created if it does not exist)

□ Use this user for the Suite Administrator (not recommended)

Figure 1-62 Choosing the style of user registry - Internal or Local OS

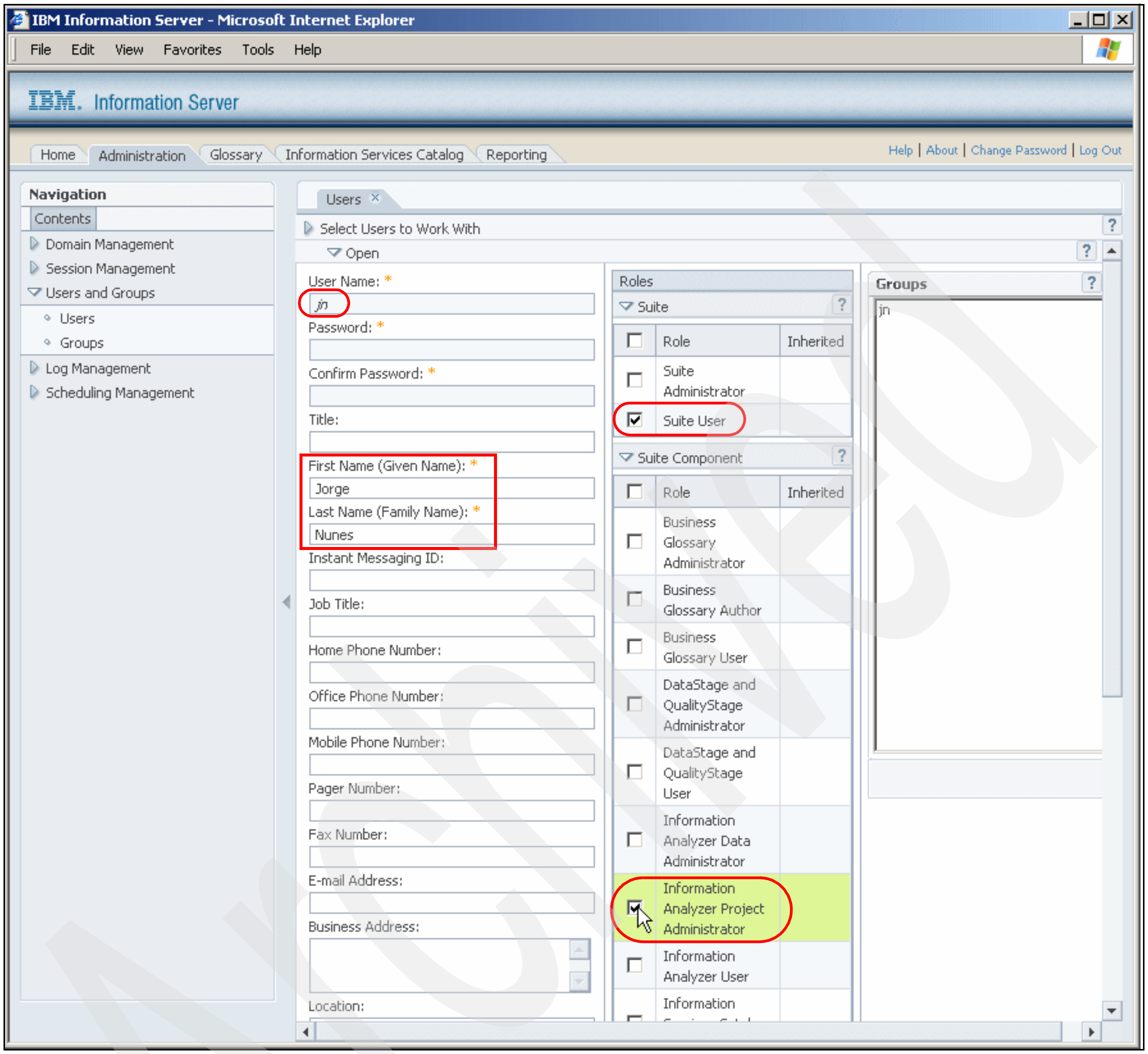


IBM Information Server - Microsoft Internet Explorer

Item 1-25 out of 49 | 1 2 | Items per Page: 25 50 100

□			haldaemon
□			halt
□			iauser
☑	Nunes	Jorge	jn
□			lp
□			mail

Figure 1-63 Set up operating system users 1/6



IBM Information Server - Microsoft Internet Explorer

File
Edit
View
Favorites
Tools
Help

IBM. Information Server

Home
Administration
Glossary
Information Services Catalog
Reporting
Help | About | Change Password | Log Out

Navigation

Contents

- Domain Management
- Session Management
- Users and Groups
  - Users
  - Groups
- Log Management
- Scheduling Management

Users ×

Select Users to Work With

Open

User Name: * jn

Password: *

Confirm Password: *

Title:

First Name (Given Name): * Jorge

Last Name (Family Name): * Nunes

Instant Messaging ID:

Job Title:

Home Phone Number:

Office Phone Number:

Mobile Phone Number:

Pager Number:

Fax Number:

E-mail Address:

Business Address:

Location:

Roles

Suite

□	Role	Inherited
□	Suite Administrator	
☑	Suite User	

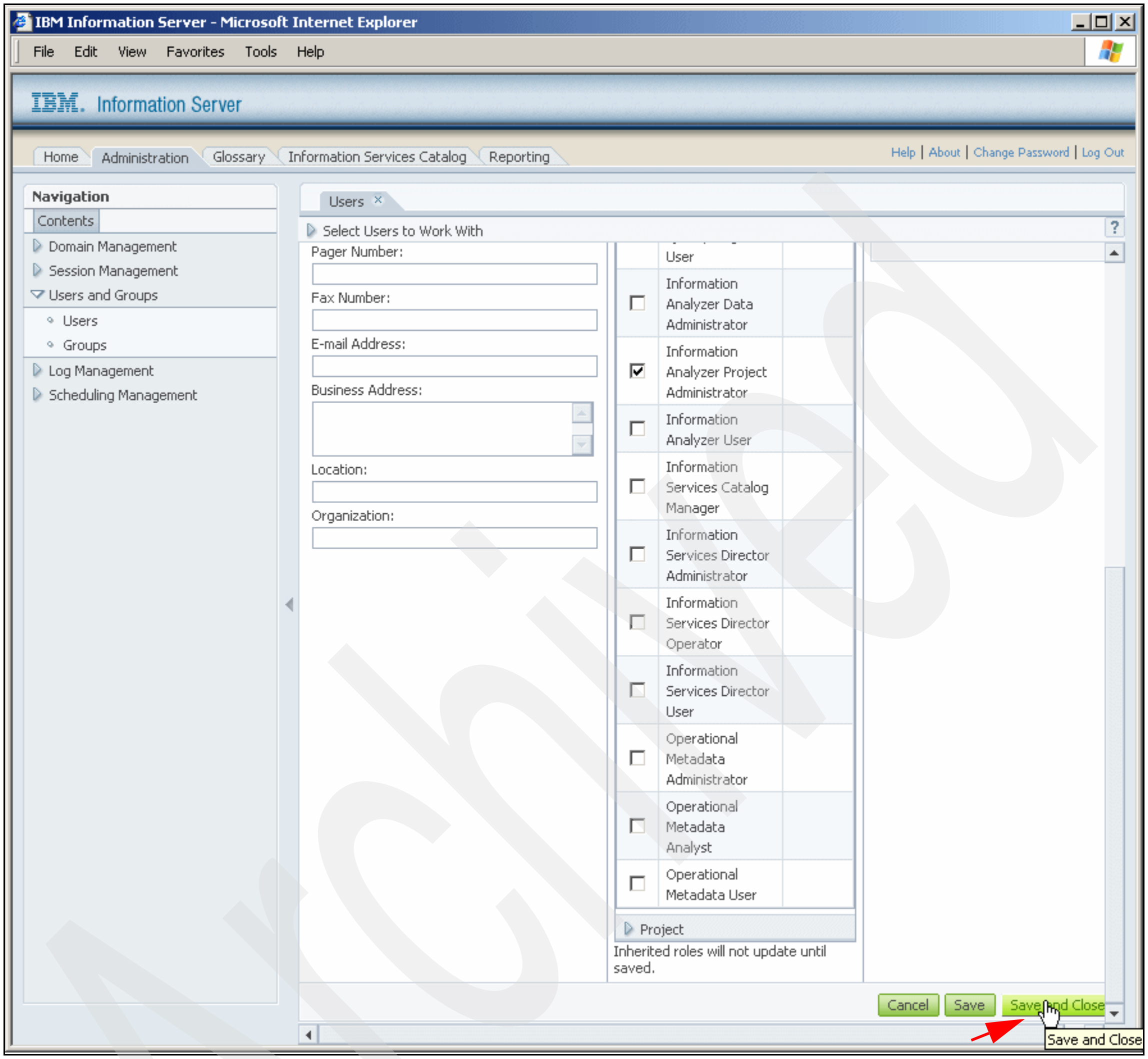
Suite Component

□	Role	Inherited
□	Business Glossary Administrator	
□	Business Glossary Author	
□	Business Glossary User	
□	DataStage and QualityStage Administrator	
□	DataStage and QualityStage User	
□	Information Analyzer Data Administrator	
☑	Information Analyzer Project Administrator	
□	Information Analyzer User	
□	Information	

Groups

jn

Figure 1-64 Set up operating system users 2/6



Pager Number:

Fax Number:

E-mail Address:

Business Address:

Location:

Organization:

User

☐ Information Analyzer Data Administrator

☑ Information Analyzer Project Administrator

☐ Information Analyzer User

☐ Information Services Catalog Manager

☐ Information Services Director Administrator

☐ Information Services Director Operator

☐ Information Services Director User

☐ Operational Metadata Administrator

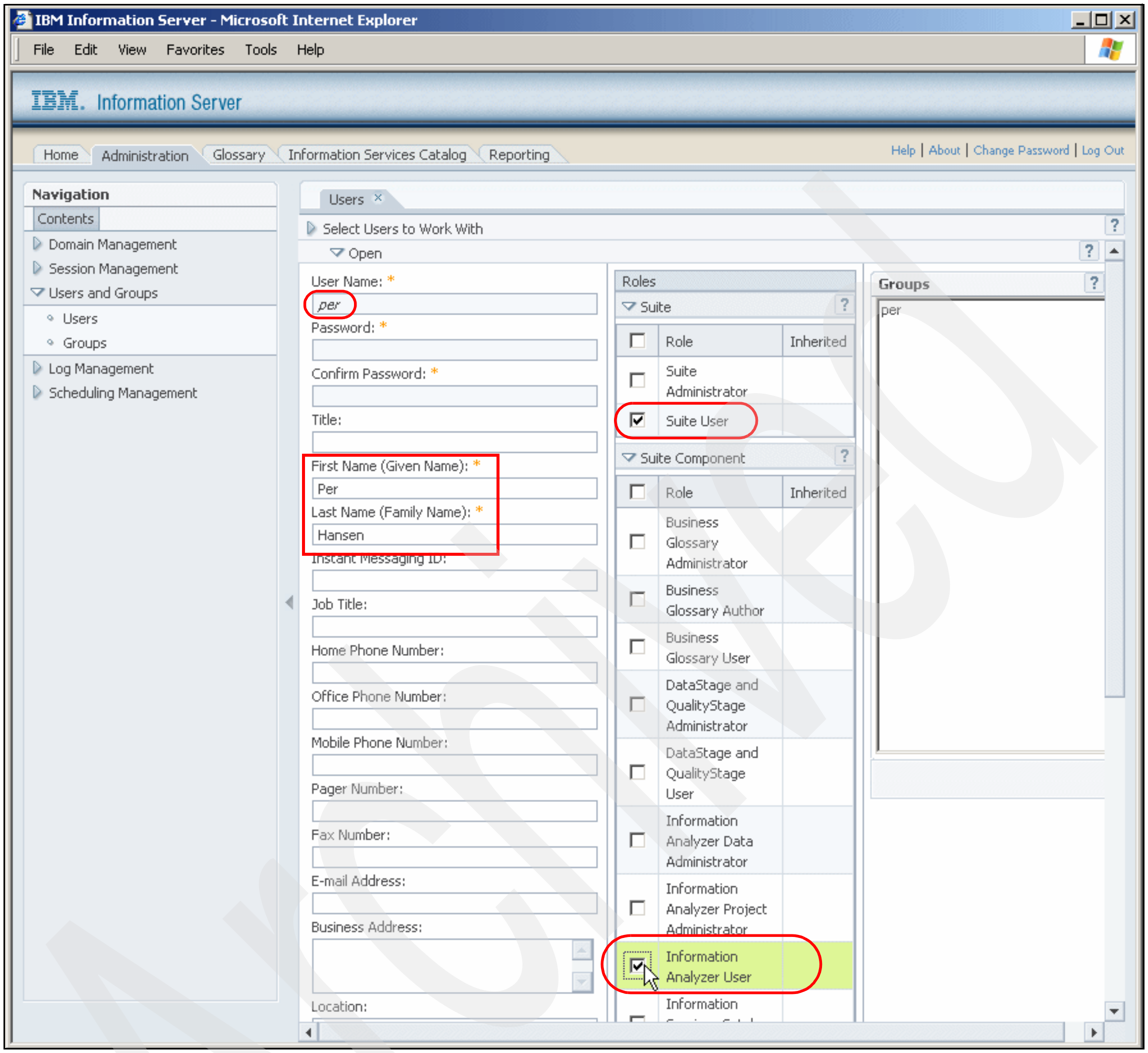
☐ Operational Metadata Analyst

☐ Operational Metadata User

Project

Inherited roles will not update until saved.

Figure 1-65 Set up operating system users 3/6


IBM Information Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
IBM. Information Server
Home | Administration | Glossary | Information Services Catalog | Reporting
Help | About | Change Password | Log Out
Navigation
Contents
Domain Management
Session Management
Users and Groups
Users
Groups
Log Management
Scheduling Management
Users ×
Select Users to Work With
Open

User Name: * per

Password: *

Confirm Password: *

Title:

First Name (Given Name): * Per

Last Name (Family Name): * Hansen

Instant Messaging ID:

Job Title:

Home Phone Number:

Office Phone Number:

Mobile Phone Number:

Pager Number:

Fax Number:

E-mail Address:

Business Address:

Location:

Roles

Suite

	Role	Inherited
□	Suite Administrator	
☑	Suite User	

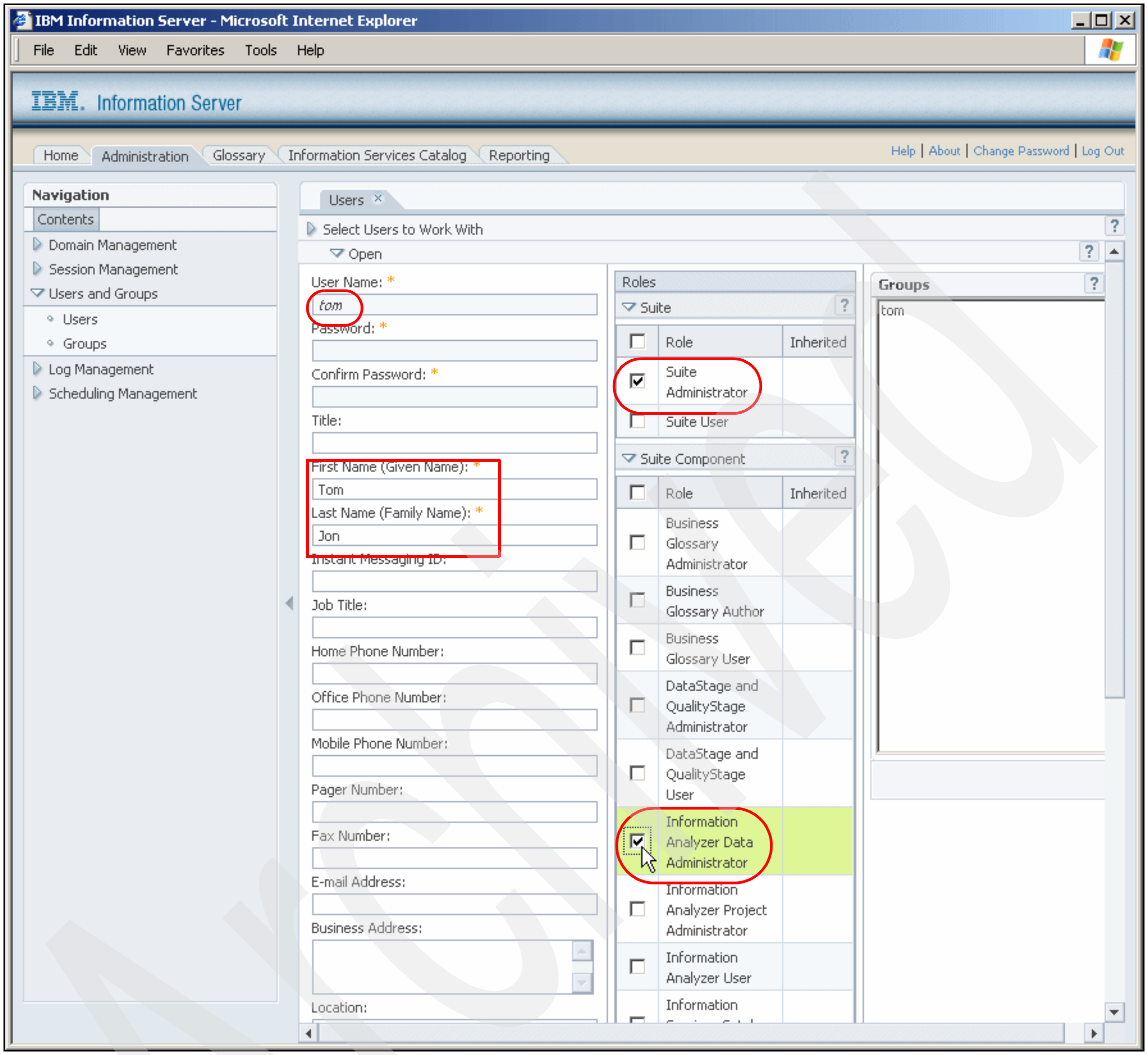
Suite Component

	Role	Inherited
□	Business Glossary Administrator	
□	Business Glossary Author	
□	Business Glossary User	
□	DataStage and QualityStage Administrator	
□	DataStage and QualityStage User	
□	Information Analyzer Data Administrator	
□	Information Analyzer Project Administrator	
☑	Information Analyzer User	
□	Information	

Groups

per

Figure 1-66 Set up operating system users 4/6


IBM Information Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
IBM. Information Server
Home Administration Glossary Information Services Catalog Reporting
Help | About | Change Password | Log Out
Navigation
Contents
Domain Management
Session Management
Users and Groups
Users
Groups
Log Management
Scheduling Management
Users
Select Users to Work With
Open
Roles
Suite
Suite Component
Role
Inherited
Groups

User Name: * tom
Password: *
Confirm Password: *
Title:
First Name (Given Name): * Tom
Last Name (Family Name): * Jon
Instant Messaging ID:
Job Title:
Home Phone Number:
Office Phone Number:
Mobile Phone Number:
Pager Number:
Fax Number:
E-mail Address:
Business Address:
Location:

Suite:

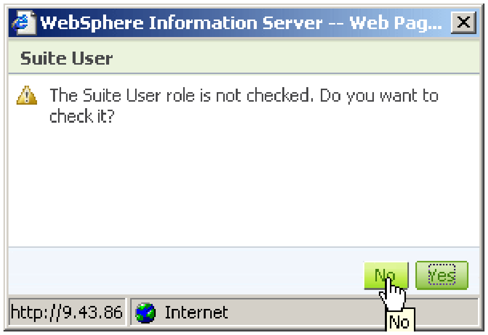
	Role	Inherited
☑	Suite Administrator	
□	Suite User	

Suite Component:

	Role	Inherited
□	Business Glossary Administrator	
□	Business Glossary Author	
□	Business Glossary User	
□	DataStage and QualityStage Administrator	
□	DataStage and QualityStage User	
☑	Information Analyzer Data Administrator	
□	Information Analyzer Project Administrator	
□	Information Analyzer User	
□	Information	

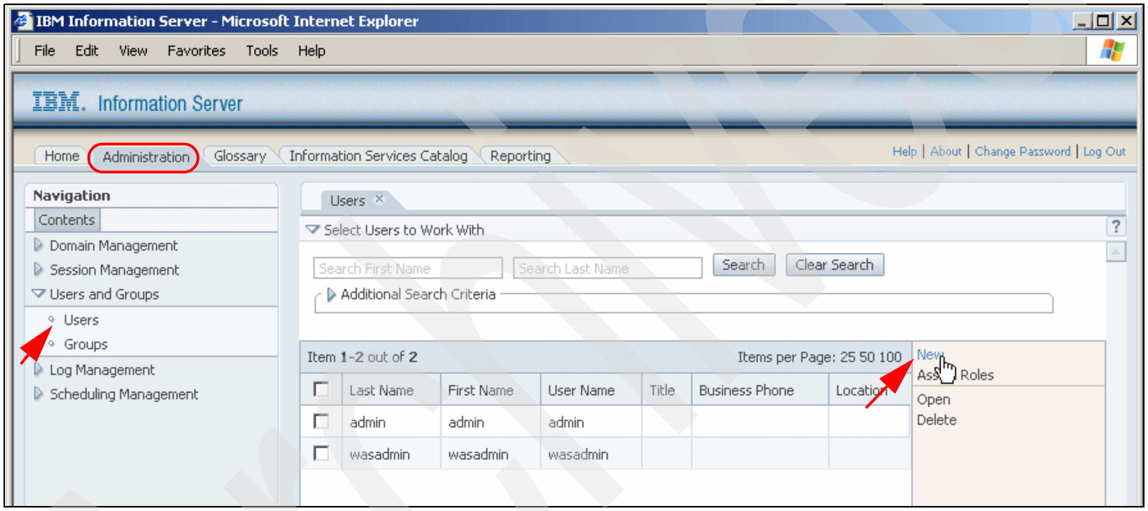
Groups: tom

Figure 1-67 Set up operating system users 5/6



The Suite User role is not checked. Do you want to check it?

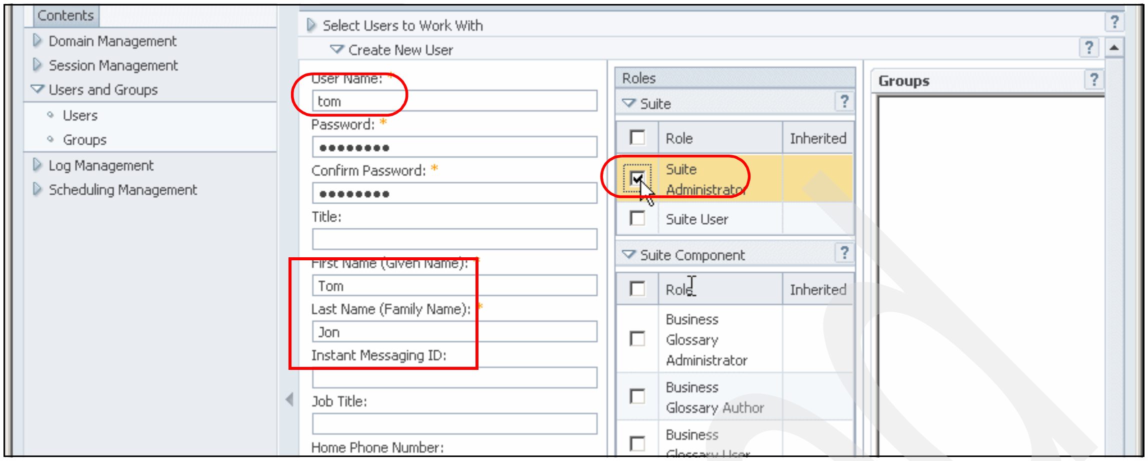
Figure 1-68 Set up operating system users 6/6



Item 1-2 out of 2 Items per Page: 25 50 100

□	Last Name	First Name	User Name	Title	Business Phone	Location
□	admin	admin	admin			
□	wasadmin	wasadmin	wasadmin			

Figure 1-69 Set up internal user registry users 1/8



User Name: tom

Password: * ●●●●●●●●

Confirm Password: * ●●●●●●●●

Title:

First Name (Given Name): * Tom

Last Name (Family Name): * Jon

Instant Messaging ID:

Job Title:

Home Phone Number:

Roles

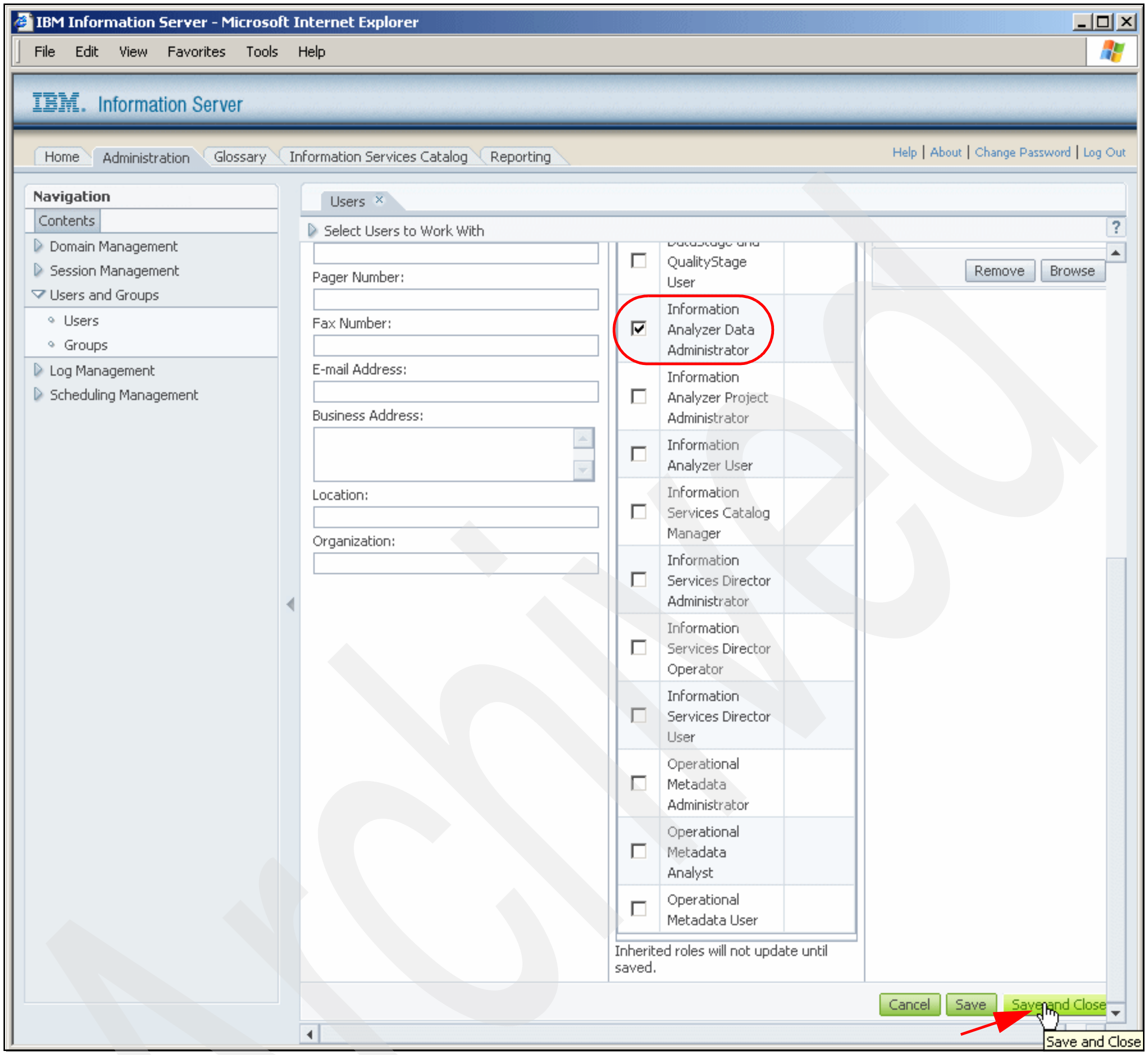
Suite

□	Role	Inherited
☑	Suite Administrator	
□	Suite User	

Suite Component

□	Role	Inherited
□	Business Glossary Administrator	
□	Business Glossary Author	
□	Business Glossary User	

Figure 1-70 Set up internal user registry users 2/8



IBM Information Server - Microsoft Internet Explorer

Pager Number:

Fax Number:

E-mail Address:

Business Address:

Location:

Organization:

□ DataStage and QualityStage User

☑ Information Analyzer Data Administrator

□ Information Analyzer Project Administrator

□ Information Analyzer User

□ Information Services Catalog Manager

□ Information Services Director Administrator

□ Information Services Director Operator

□ Information Services Director User

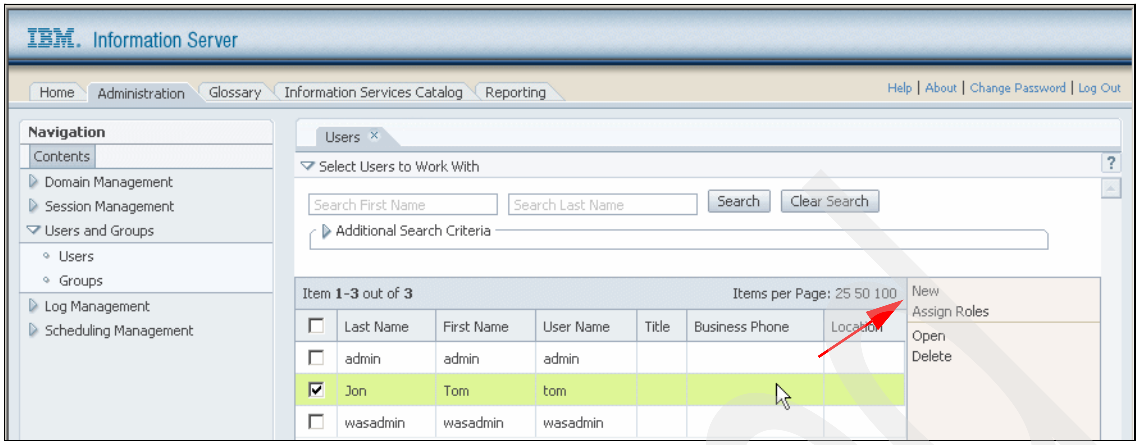
□ Operational Metadata Administrator

□ Operational Metadata Analyst

□ Operational Metadata User

Inherited roles will not update until saved.

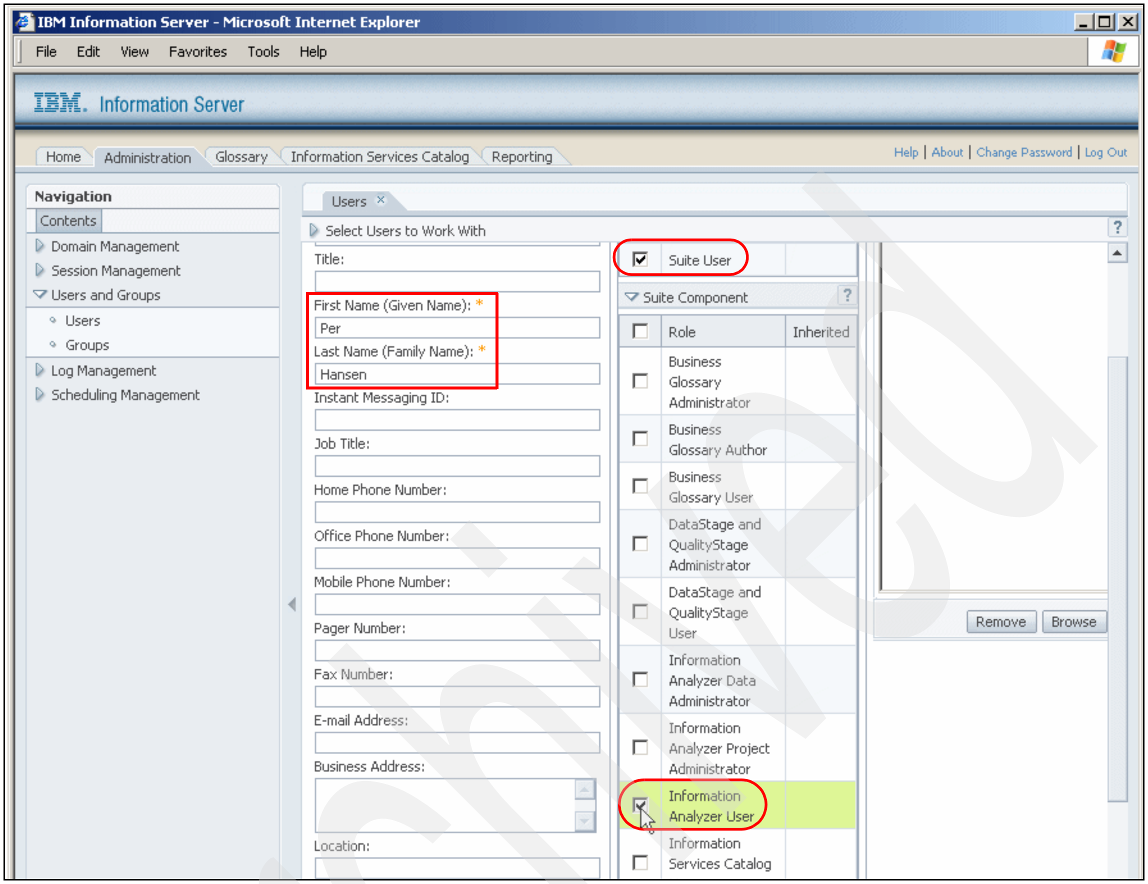
Figure 1-71 Set up internal user registry users 3/8



Item 1-3 out of 3 Items per Page: 25 50 100

□	Last Name	First Name	User Name	Title	Business Phone	Location
□	admin	admin	admin			
☑	Jon	Tom	tom			
□	wasadmin	wasadmin	wasadmin			

Figure 1-72 Set up internal user registry users 4/8


IBM Information Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
IBM. Information Server
Home Administration Glossary Information Services Catalog Reporting
Help | About | Change Password | Log Out
Navigation
Contents
Domain Management
Session Management
Users and Groups
Users
Groups
Log Management
Scheduling Management
Users
Select Users to Work With
Remove
Browse

Title:

First Name (Given Name): * Per

Last Name (Family Name): * Hansen

Instant Messaging ID:

Job Title:

Home Phone Number:

Office Phone Number:

Mobile Phone Number:

Pager Number:

Fax Number:

E-mail Address:

Business Address:

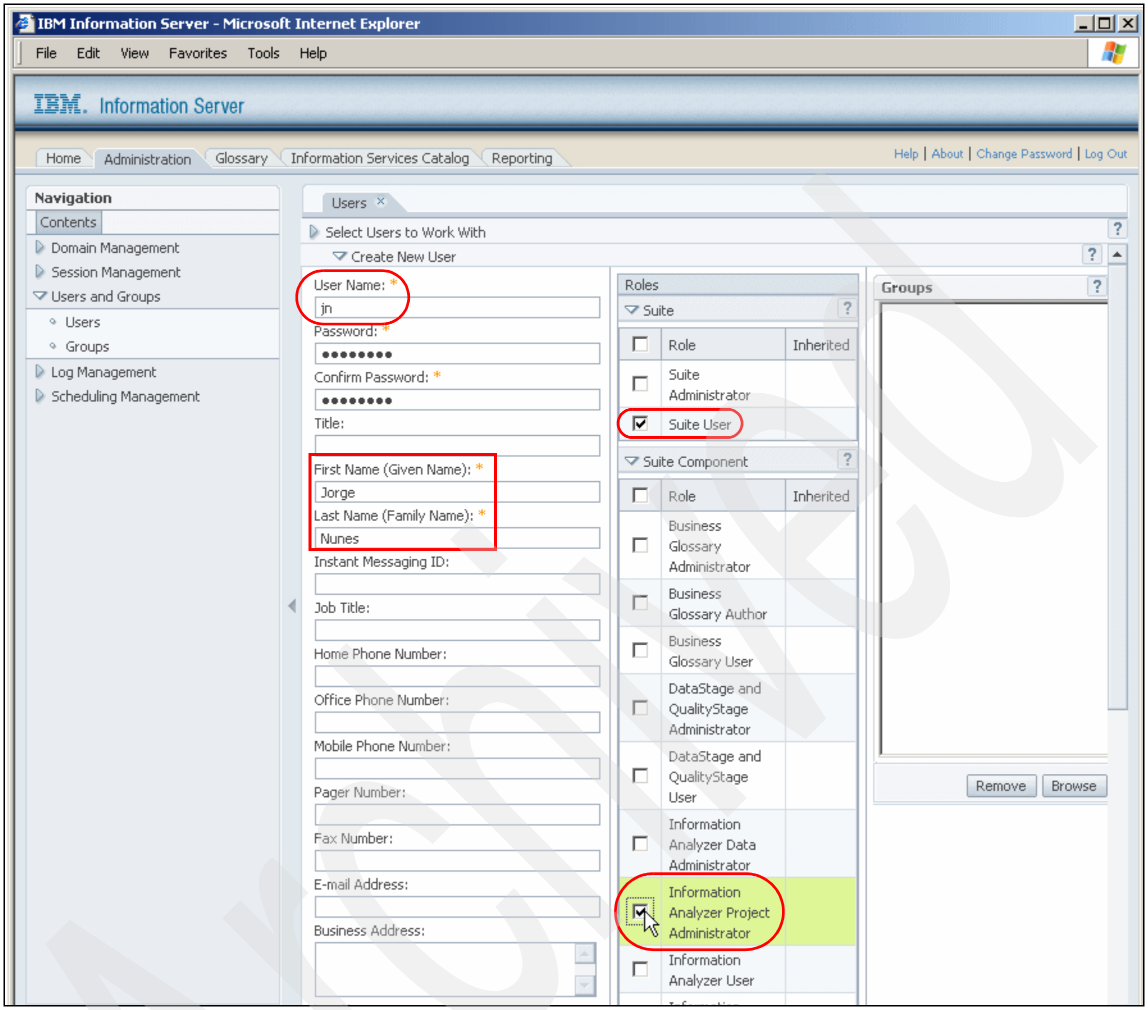
Location:

☑ Suite User

Suite Component

□	Role	Inherited
□	Business Glossary Administrator	
□	Business Glossary Author	
□	Business Glossary User	
□	DataStage and QualityStage Administrator	
□	DataStage and QualityStage User	
□	Information Analyzer Data Administrator	
□	Information Analyzer Project Administrator	
☑	Information Analyzer User	
□	Information Services Catalog	

Figure 1-73 Set up internal user registry users 5/8


IBM Information Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
IBM. Information Server
Home Administration Glossary Information Services Catalog Reporting
Help | About | Change Password | Log Out
Navigation
Contents
Domain Management
Session Management
Users and Groups
Users
Groups
Log Management
Scheduling Management
Users
Select Users to Work With
Create New User
Groups
Remove
Browse

User Name: * jn

Password: * ••••••••

Confirm Password: * ••••••••

Title:

First Name (Given Name): * Jorge

Last Name (Family Name): * Nunes

Instant Messaging ID:

Job Title:

Home Phone Number:

Office Phone Number:

Mobile Phone Number:

Pager Number:

Fax Number:

E-mail Address:

Business Address:

Roles

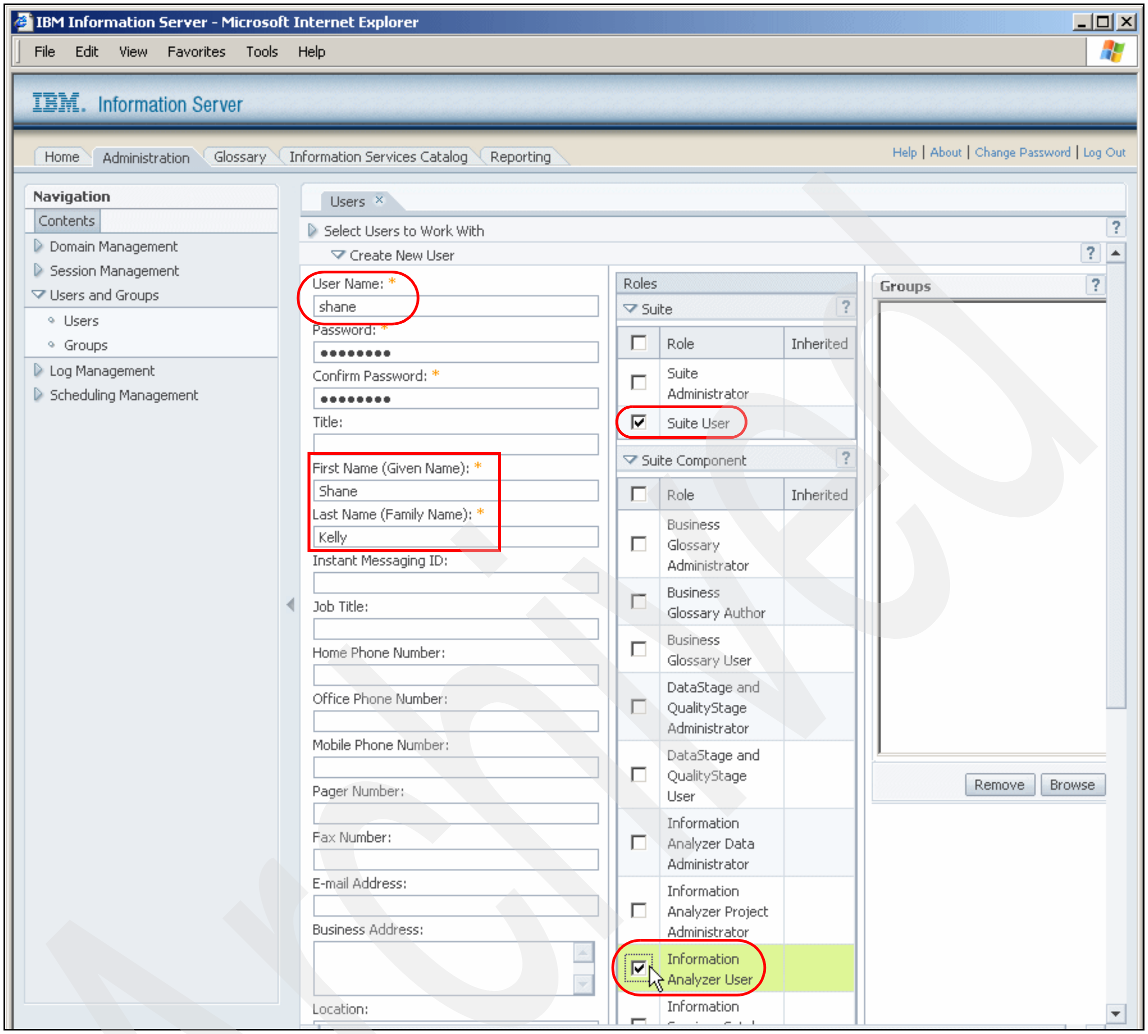
Suite

□	Role	Inherited
□	Suite Administrator	
☑	Suite User	

Suite Component

□	Role	Inherited
□	Business Glossary Administrator	
□	Business Glossary Author	
□	Business Glossary User	
□	DataStage and QualityStage Administrator	
□	DataStage and QualityStage User	
□	Information Analyzer Data Administrator	
☑	Information Analyzer Project Administrator	
□	Information Analyzer User	

Figure 1-74 Set up internal user registry users 6/8



IBM Information Server - Microsoft Internet Explorer

File Edit View Favorites Tools Help

IBM. Information Server

Home | Administration | Glossary | Information Services Catalog | Reporting

Help | About | Change Password | Log Out

Navigation

Contents

- Domain Management
- Session Management
- Users and Groups
  - Users
  - Groups
- Log Management
- Scheduling Management

Users

Select Users to Work With

Create New User

User Name: * shane

Password: * ••••••••

Confirm Password: * ••••••••

Title:

First Name (Given Name): * Shane

Last Name (Family Name): * Kelly

Instant Messaging ID:

Job Title:

Home Phone Number:

Office Phone Number:

Mobile Phone Number:

Pager Number:

Fax Number:

E-mail Address:

Business Address:

Location:

Roles

Suite

□	Role	Inherited
□	Suite Administrator	
☑	Suite User	

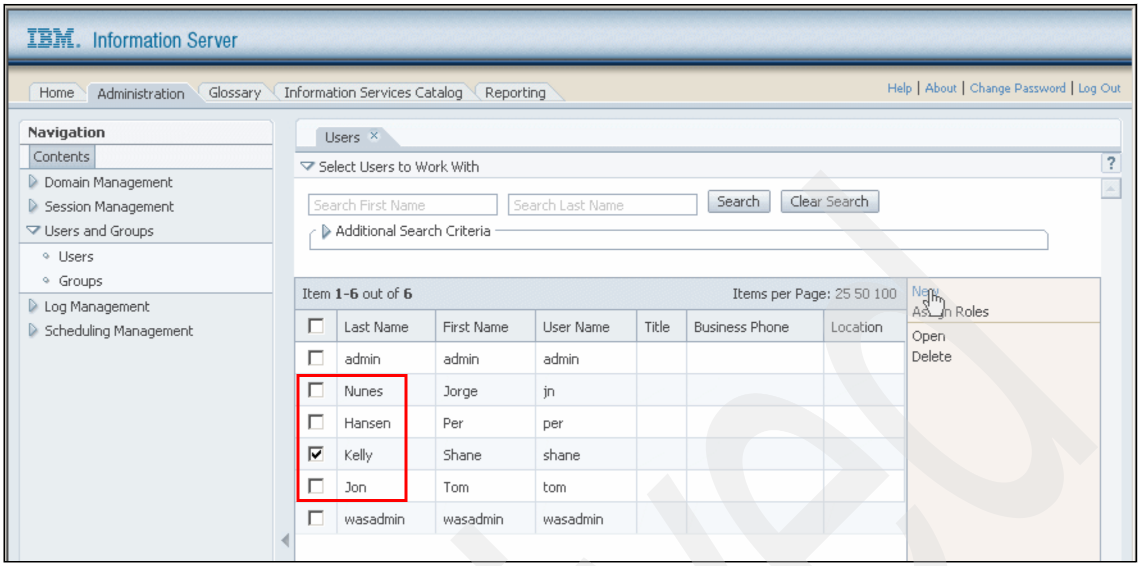
Suite Component

□	Role	Inherited
□	Business Glossary Administrator	
□	Business Glossary Author	
□	Business Glossary User	
□	DataStage and QualityStage Administrator	
□	DataStage and QualityStage User	
□	Information Analyzer Data Administrator	
□	Information Analyzer Project Administrator	
☑	Information Analyzer User	
□	Information	

Groups

Remove | Browse

Figure 1-75 Set up internal user registry users 7/8



Item 1-6 out of 6 Items per Page: 25 50 100

□	Last Name	First Name	User Name	Title	Business Phone	Location
□	admin	admin	admin			
□	Nunes	Jorge	jn			
□	Hansen	Per	per			
☑	Kelly	Shane	shane			
□	Jon	Tom	tom			
□	wasadmin	wasadmin	wasadmin			

Figure 1-76 Set up internal user registry users 8/8

1.7 Column analysis

To determine the structure and content of your data and to identify anomalies and inconsistencies in your data, you can analyze columns. Column analysis results are used as the foundation for most of the other data profiling tasks as shown in Figure 1-7 on page 23. After column analysis completes, a frequency distribution for each column is generated and then used as the input for the subsequent analyses. During profiling analysis, inferences are made about your data. The inferences often represent areas that might offer the opportunity to improve the quality of your data. When an analysis completes, you can review the inferences and accept or reject them.

In the following sections, we describe briefly the main functions of column analysis and the results that are produced by column analysis. Using a sample set of tables, we describe the process of running a column analysis and discuss the results of the column analysis.

1.7.1 Column analysis functions

Column analysis has the following characteristics:

- You must have Information Analyzer Data Operator privileges to run column analysis as shown in Table 1-4 on page 34. Also, you must have either Information Analyzer Data Steward or Information Analyzer Business Analyst privileges to open column analysis, and Information Analyzer Business Analyst privilege to edit and update column analysis information.
- Can be performed on all the columns of one or more tables, or selectively on certain columns.
- Allows you to run an analysis on the full volume of data, or on a subset using a sampling technique.
- Allows you to generate reference tables. You can use a reference table to share results from the frequency distribution with additional IBM Information Server capabilities or other systems. Valid, Range, Completeness, Invalid, and Mapping reference tables can be used in additional IBM Information Server capabilities or other systems to enforce domain requirements and completeness requirements or to control data conversion.
- Allows you to rebuild the inferences after you have marked data values as being valid, invalid, or incomplete.
- Allows you to provide reference tables or frequency distribution information about invalid values from a previous column analysis run, to be fed back in to a subsequent column analysis run to identify valid values.
- Provides options on specifying the isolation level when accessing relational data, whether auto commit should be turned on, whether the generated scripts should be retained, and whether the most recent column analysis results should completely replace a previous column analysis result or just update relevant portions of the previous result.
- Allows you to define *virtual columns* on which you can run analysis jobs. A virtual column is the concatenation of data from one or more columns in a single table into one column. For example, you can combine a First Name column and a Last Name column into one column called Name. You could then analyze this concatenated column, which contains the data in both the First Name column and the Last Name column. You might analyze virtual columns when you want to use the results in analyses that evaluate multiple columns, such as multiple column primary key analysis or foreign key analysis, or to validate a value pair combination from several fields.

  You can also create a virtual column on a single column to truncate or pad the data in that column.
- Can be scheduled to run at a set date and time.

- Reports can be produced of column analysis results such as column classification, column domain, column frequency, column inferred, column properties, and column summary. We describe these in 1.14, “Reports” on page 394.

Note: For complete details, refer to *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.

During a column analysis job, the characteristics of your data are evaluated by analyzing a frequency distribution of the data values in each column. After the frequency distribution is analyzed, inferences are made by the system. Inferences are the best initial choice that the system can make about a specific element in the data, while selection is the application of human knowledge to the system inference. For example, when your data is analyzed, data classes are inferred for each column.[16] A data field is classified into a data class based on the semantic business use and physical properties of the data field.

Important: You can accept the inferences or make another choice when you review the results of the column analysis in the frequency distribution. This corresponds to the Selected column in the panels. It is critical that you ensure that any inferred properties are correct, because the Selected property is used in subsequent analyses such as foreign key analysis.

[16] A *data class* categorizes a column according to how the data in the column is used. For example, if a column contains data such as 10/04/07, the Date class is assigned to the column because 10/04/07 is an expression for a date. By classifying each data column, you gain a better understanding of the type of data in that column and how it is used. To ensure that your data is of good quality, you must assign accurate classes to your data. To assign a data class, the frequency distribution of a column is evaluated for characteristics such as cardinality (the number of distinct values in a column) and data type. A *data type* describes the structural format of data in a column. For example, columns that contain numeric data are type N (numeric), and columns that contain alphabetic data are type A (alphabetic). The frequency distribution results are used by the system to infer a class for the column. After the analysis completes, you review, accept, or reject the inferences. One of eight data classes is inferred for each column. An *Identifier* column has a data value that is used to reference a unique entity. For example, a column with the class of Identifier might be a primary key or contain unique information such as a customer number. An *Indicator* column contains only two values. For example a column with the class of Indicator might contain data such as true and false or yes and no. A *Code* column contains code values that represent a specific meaning. For example, a column with the class of Code might contain data about the area code in a telephone number. A *Date* column includes chronological data. For example, a column with the class of Date might contain data such as 10/10/07. A *Quantity* column that contains data about the numerical value of something. For example, a column with the class of Quantity might contain data about the price of an object. A *Large Object* column has a BLOB data type. For example, a column with the class of Large Object might contain an array. A *Text* column contains free-form alphanumeric data. For example, a column with the class of Text might contain data about the name of a company or person. An *Unknown* column data class corresponds to columns that cannot be classified by the system are defined as Unknown. The Unknown class is applied temporarily during analysis.

You run a column analysis job to evaluate the following characteristics of columns in your data:

- Minimum, maximum, and average field length

  If a field is too short, you can lose data. If the field is too large, it uses memory that might be needed elsewhere.

- Precision and scale of numeric values

  Precision refers to the total length of a numeric field. The scale of a numeric value is the total length of the decimal component of a numeric field.

- Basic data types, including date and time formats

  Basic data types, such as numeric and alphabetic types, are evaluated for incorrect typing.

- Count of distinct values (cardinality)

  Cardinality refers to the number of distinct values in a column including blanks and nulls.

- Empty values and null values

  Empty values use space and can be removed.

During a column analysis job, the following inferences are made about the content and structure of columns in your data:

- Data type, length, precision, and scale

  Type, length, precision, and scale values are inferred.

- Data class

  A data class corresponding to one of eight system defined classes is inferred.

- Field value uniqueness and constancy

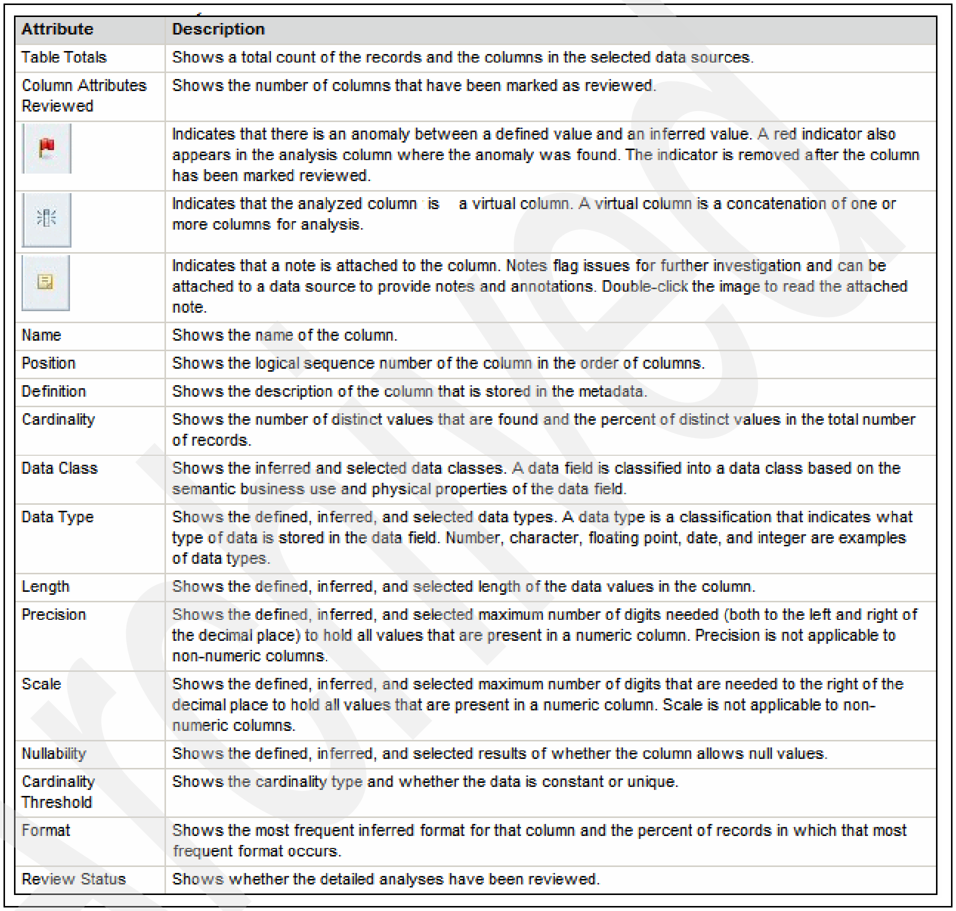
  Both unique values and constant values are inferred.

  - Unique values identify whether a column contains mostly unique data values.
  - Constant value identifies that a column contains one and only one value. The presence of a constant value can indicate an unused field (all nulls or spaces), or an extraneous or redundant field (field was planned for other use, but is not effectively used).

You can choose to build reference tables (by clicking **Reference Tables**) from the values in the column for consumption by a subsequent column analysis job, or other suite components, or external programs.

You can make changes to the column analysis results such as identifying valid and invalid values and formats, and then request a re-evaluation of inferences using all values or valid values by clicking **Rebuild Inferences**.

Figure 1-77 describes the properties, attributes, and values that are returned after a column analysis job runs.



Attribute	Description
Table Totals	Shows a total count of the records and the columns in the selected data sources.
Column Attributes Reviewed	Shows the number of columns that have been marked as reviewed.
🚩	Indicates that there is an anomaly between a defined value and an inferred value. A red indicator also appears in the analysis column where the anomaly was found. The indicator is removed after the column has been marked reviewed.
⇹	Indicates that the analyzed column is a virtual column. A virtual column is a concatenation of one or more columns for analysis.
📝	Indicates that a note is attached to the column. Notes flag issues for further investigation and can be attached to a data source to provide notes and annotations. Double-click the image to read the attached note.
Name	Shows the name of the column.
Position	Shows the logical sequence number of the column in the order of columns.
Definition	Shows the description of the column that is stored in the metadata.
Cardinality	Shows the number of distinct values that are found and the percent of distinct values in the total number of records.
Data Class	Shows the inferred and selected data classes. A data field is classified into a data class based on the semantic business use and physical properties of the data field.
Data Type	Shows the defined, inferred, and selected data types. A data type is a classification that indicates what type of data is stored in the data field. Number, character, floating point, date, and integer are examples of data types.
Length	Shows the defined, inferred, and selected length of the data values in the column.
Precision	Shows the defined, inferred, and selected maximum number of digits needed (both to the left and right of the decimal place) to hold all values that are present in a numeric column. Precision is not applicable to non-numeric columns.
Scale	Shows the defined, inferred, and selected maximum number of digits that are needed to the right of the decimal place to hold all values that are present in a numeric column. Scale is not applicable to non-numeric columns.
Nullability	Shows the defined, inferred, and selected results of whether the column allows null values.
Cardinality Threshold	Shows the cardinality type and whether the data is constant or unique.
Format	Shows the most frequent inferred format for that column and the percent of records in which that most frequent format occurs.
Review Status	Shows whether the detailed analyses have been reviewed.

Figure 1-77 Column Analysis result reference

1.7.2 Column analysis results

A column analysis produces the following categories of output:

- View Analysis Summary
- Overview
- Frequency Distribution
- Data Class
- Properties
- Domain & Completeness
- Format

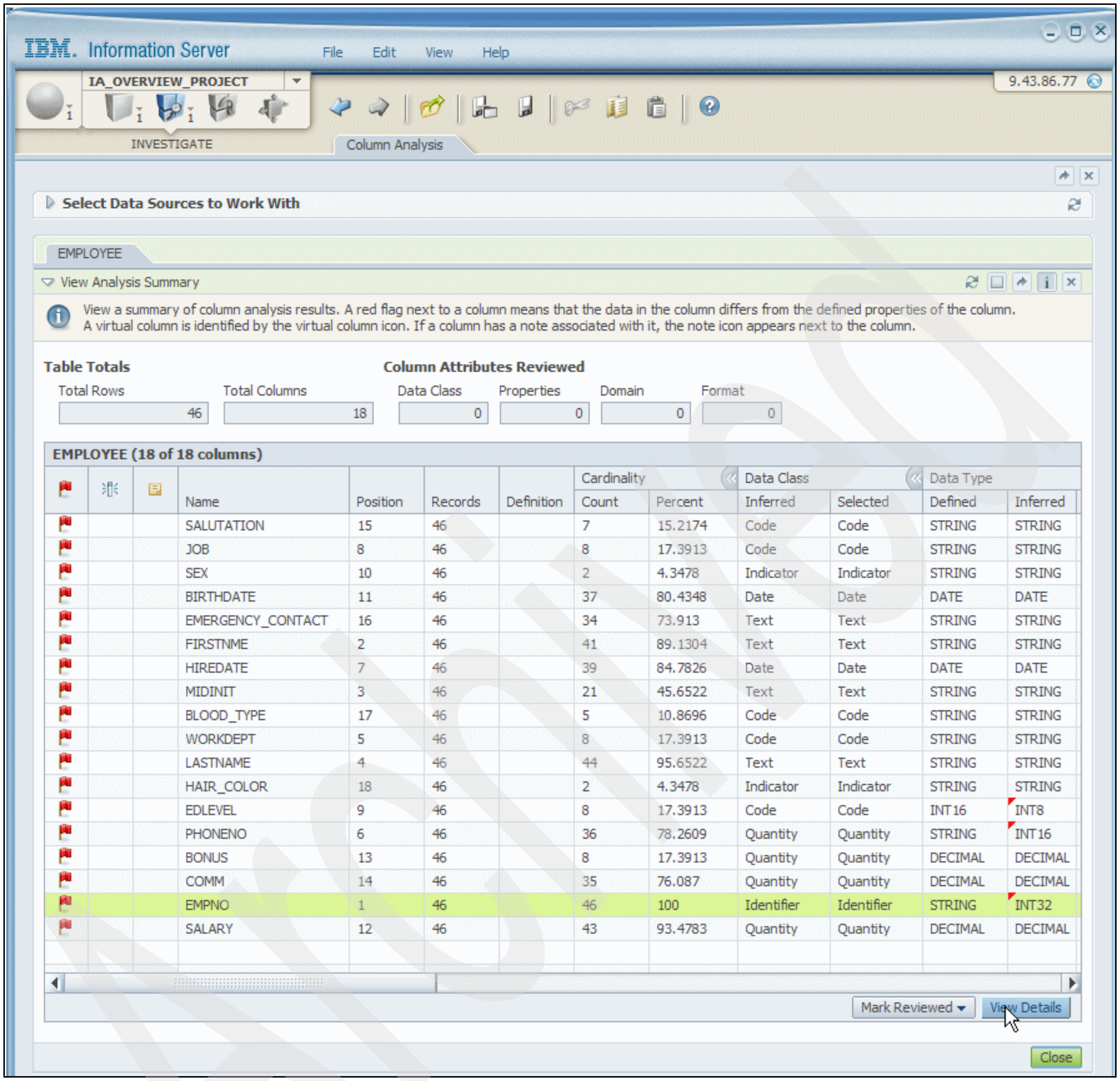
> **Important:** As described in 1.2.1, “Data assessment approach” on page 5, analysis is a critical follow on step to running a profiling job, which only delivers a set of statistics and system inferences. The Data Analyst must review that information to bring insight and clarity to those statistics and inferences. Table 1-8 on page 129 summarizes the series of key dimensions or questions on which a Data Analyst needs to focus when evaluating the results of an IBM WebSphere Information Analyzer Column Analysis profiling job.

We describe each of these categories briefly here.

- View Analysis Summary

  This is a summary of column analysis results as shown in Figure 1-78 on page 115. The summary provides details of the defined and inferred properties such as data type, length, precision, scale, nullability, cardinality type, and format. It also provides information about the cardinality and the inferred data class. It also identifies whether an individual column’s analysis was reviewed, whether the selected attribute. A red flag next to a column means that the data in the column differs from the defined properties of the column. A virtual column is identified by the virtual column icon . If a column has a note associated with it, the note icon appears next to the column. Click **View Details** in Figure 1-78 on page 115 to proceed to view details of the frequency distribution, data classes, properties, domain and completeness, and format as shown in Figure 1-79 on page 116 through in Figure 1-87 on page 128.

  After you have fully analyzed the results of column analysis, you can formally mark a review as complete to enable the system to keep track of which areas of analysis were reviewed in detail. We described this in 1.7.3, “Column analysis usage scenario” on page 134.



View a summary of column analysis results. A red flag next to a column means that the data in the column differs from the defined properties of the column. A virtual column is identified by the virtual column icon. If a column has a note associated with it, the note icon appears next to the column.

Table Totals

Total Rows	Total Columns
46	18

Column Attributes Reviewed

Data Class	Properties	Domain	Format
0	0	0	0

EMPLOYEE (18 of 18 columns)

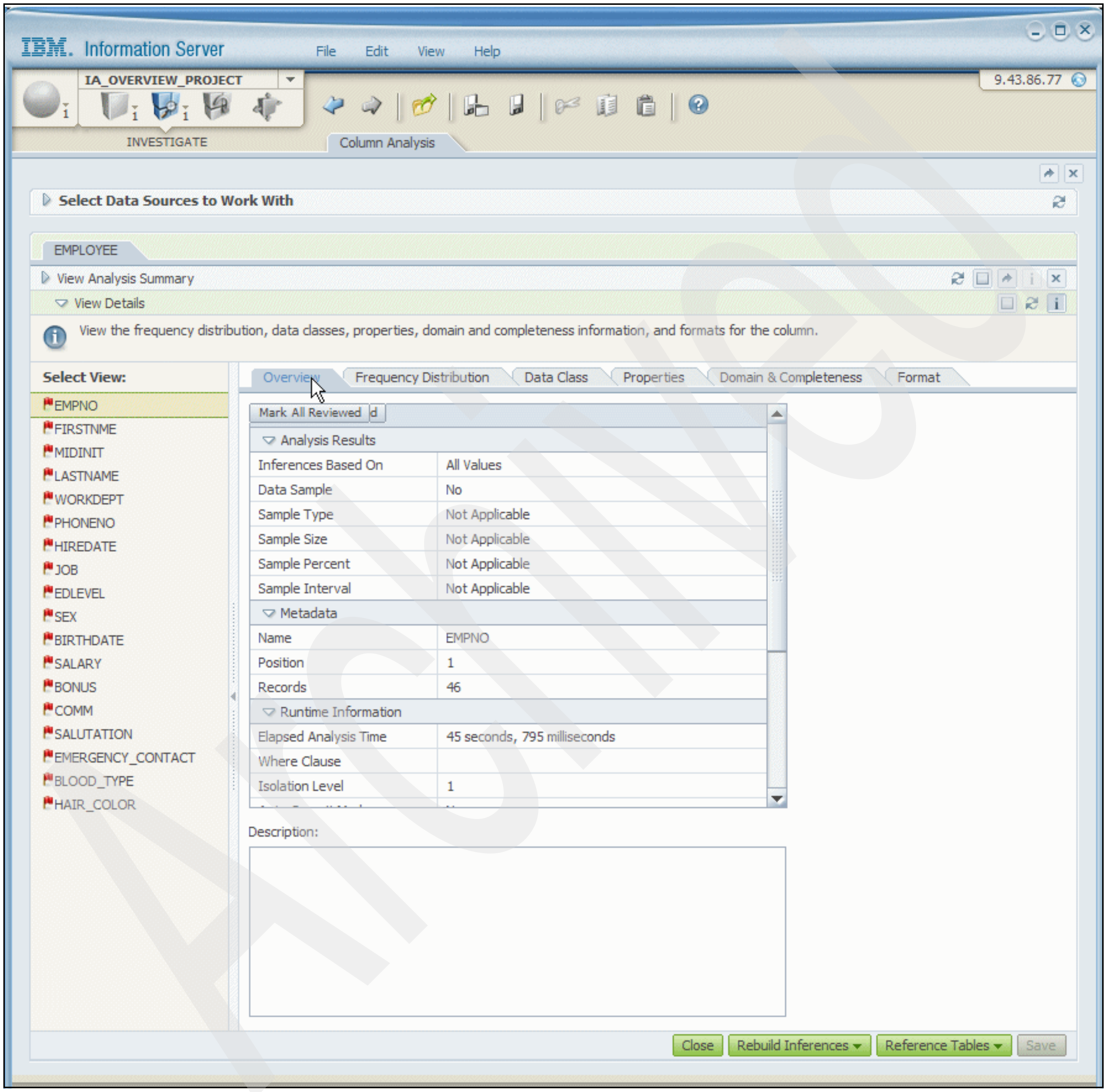
							Cardinality		Data Class		Data Type	
🚩			Name	Position	Records	Definition	Count	Percent	Inferred	Selected	Defined	Inferred
🚩			SALUTATION	15	46		7	15.2174	Code	Code	STRING	STRING
🚩			JOB	8	46		8	17.3913	Code	Code	STRING	STRING
🚩			SEX	10	46		2	4.3478	Indicator	Indicator	STRING	STRING
🚩			BIRTHDATE	11	46		37	80.4348	Date	Date	DATE	DATE
🚩			EMERGENCY_CONTACT	16	46		34	73.913	Text	Text	STRING	STRING
🚩			FIRSTNME	2	46		41	89.1304	Text	Text	STRING	STRING
🚩			HIREDATE	7	46		39	84.7826	Date	Date	DATE	DATE
🚩			MIDINIT	3	46		21	45.6522	Text	Text	STRING	STRING
🚩			BLOOD_TYPE	17	46		5	10.8696	Code	Code	STRING	STRING
🚩			WORKDEPT	5	46		8	17.3913	Code	Code	STRING	STRING
🚩			LASTNAME	4	46		44	95.6522	Text	Text	STRING	STRING
🚩			HAIR_COLOR	18	46		2	4.3478	Indicator	Indicator	STRING	STRING
🚩			EDLEVEL	9	46		8	17.3913	Code	Code	INT16	INT8
🚩			PHONENO	6	46		36	78.2609	Quantity	Quantity	STRING	INT16
🚩			BONUS	13	46		8	17.3913	Quantity	Quantity	DECIMAL	DECIMAL
🚩			COMM	14	46		35	76.087	Quantity	Quantity	DECIMAL	DECIMAL
🚩			EMPNO	1	46		46	100	Identifier	Identifier	STRING	INT32
🚩			SALARY	12	46		43	93.4783	Quantity	Quantity	DECIMAL	DECIMAL

Figure 1-78 View Analysis Summary for the EMPLOYEE table

- Overview

  This view provides an overview of the analysis results, metadata, runtime information, and analysis status as shown in Figure 1-79 on page 116. You can choose to formally mark the analysis status as being fully reviewed or reset a fully reviewed analysis status to that of not being reviewed.

The Runtime Information section not only provides information about how long it took to analyze the column, but whether it was run on full volume or a sample and whether a Where Clause was applied in the analysis. This is important context when reviewing analysis details.



Analysis Results	
Inferences Based On	All Values
Data Sample	No
Sample Type	Not Applicable
Sample Size	Not Applicable
Sample Percent	Not Applicable
Sample Interval	Not Applicable
Metadata	
Name	EMPNO
Position	1
Records	46
Runtime Information	
Elapsed Analysis Time	45 seconds, 795 milliseconds
Where Clause	
Isolation Level	1

Figure 1-79 Overview view for the column EMPNO in the EMPLOYEE table

- Frequency Distribution

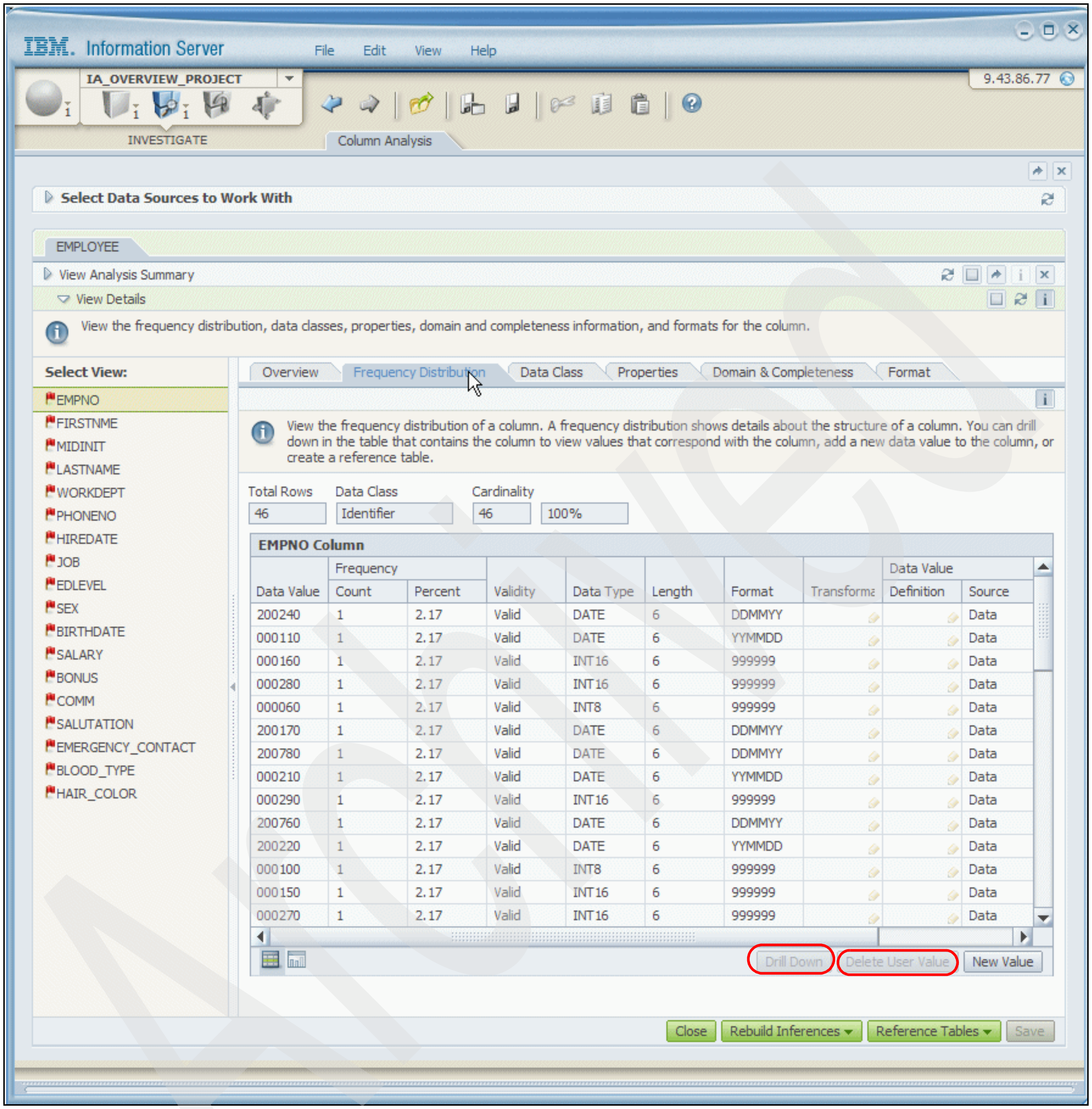
  This view shows the frequency distribution of a column as shown in Figure 1-80 on page 118. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view the values that correspond to the column, add a new data value to the column, or create a reference table. We describe this in 1.7.3, “Column analysis usage scenario” on page 134.

  If a data value appears in the frequency distribution but does not occur in the profiled column, you can delete that data value from your frequency distribution by selecting that value and clicking **Delete User Value**. The deleted value no longer appears in the frequency distribution that is saved in the metadata repository. Correspondingly, you can add a new value to be saved in the repository by clicking **New Value**.

  You can also select an individual value and click **Drill Down** to display the entire row corresponding to this value. You can also select multiple values.

  You can flag issues for further investigation and add notes to other analysts for investigation and review. Use notes to flag an issue, enter comments, or provide information to other reviewers.

  Note: For complete details, refer to *IBM WebSphere Information Analyzer Version 8.0.1 WebSphere Information Analyzer User Guide*, SC18-9902.



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the frequency distribution of a column. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view values that correspond with the column, add a new data value to the column, or create a reference table.

Total Rows: 46
Data Class: Identifier
Cardinality: 46 100%

EMPNO Column

	Frequency							Data Value	
Data Value	Count	Percent	Validity	Data Type	Length	Format	Transforma	Definition	Source
200240	1	2.17	Valid	DATE	6	DDMMYY			Data
000110	1	2.17	Valid	DATE	6	YYMMDD			Data
000160	1	2.17	Valid	INT16	6	999999			Data
000280	1	2.17	Valid	INT16	6	999999			Data
000060	1	2.17	Valid	INT8	6	999999			Data
200170	1	2.17	Valid	DATE	6	DDMMYY			Data
200780	1	2.17	Valid	DATE	6	DDMMYY			Data
000210	1	2.17	Valid	DATE	6	YYMMDD			Data
000290	1	2.17	Valid	INT16	6	999999			Data
200760	1	2.17	Valid	DATE	6	DDMMYY			Data
200220	1	2.17	Valid	DATE	6	YYMMDD			Data
000100	1	2.17	Valid	INT8	6	999999			Data
000150	1	2.17	Valid	INT16	6	999999			Data
000270	1	2.17	Valid	INT16	6	999999			Data

Figure 1-80 Frequency Distribution view for the EMPNO column in the EMPLOYEE table

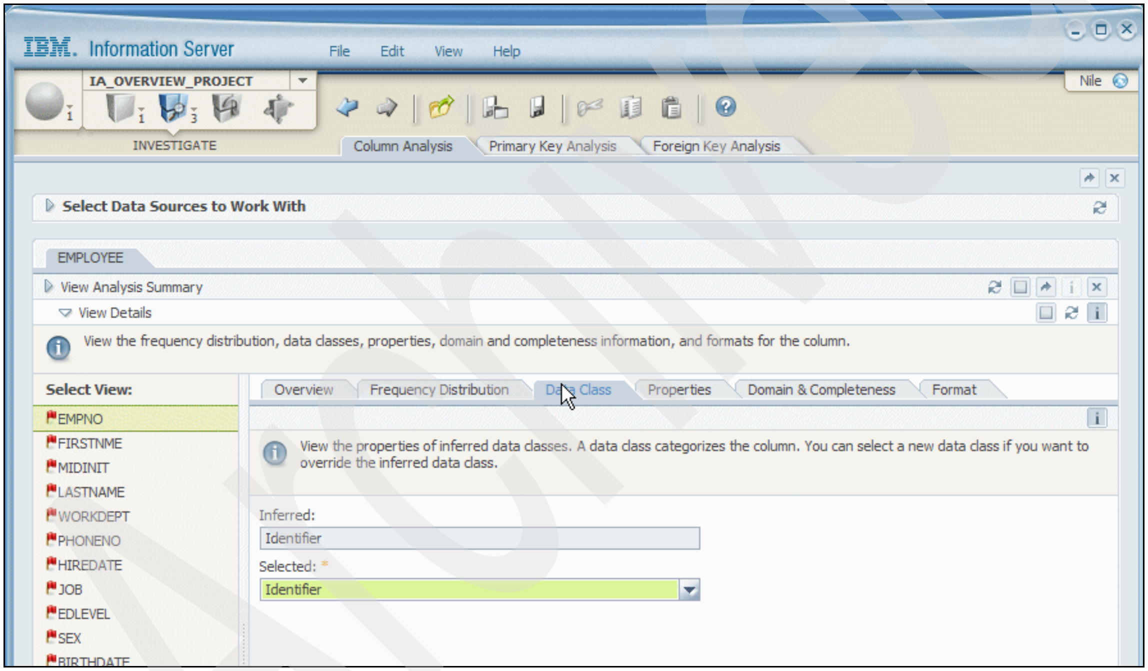
- Data Class

  This view provides details of the inferred data classes as shown in Figure 1-81 on page 119. A data class categorizes the column. You can select a new data class if you want to override the inferred data class.

As mentioned earlier, the system provides eight categories of data classes. You can choose to extend these with your own data classes, or you can add a subclass to an existing data class.

By classifying each data column, you gain a better understanding of the type of data in that column and how it is used as described in Table 1-8 on page 129. To ensure that your data is of good quality, you must assign accurate classes to your data.

> **Note:** For complete details, refer to *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the properties of inferred data classes. A data class categorizes the column. You can select a new data class if you want to override the inferred data class.

Inferred: Identifier

Selected: * Identifier

Figure 1-81 Data Class view for the EMPNO column in the EMPLOYEE table

- Properties

  This view shows the inferred and defined structural properties of a column. Properties describe the characteristics of your data. During a column analysis job, the column is evaluated and data properties are inferred. When the analysis completes, you review the inferred properties and accept the properties or choose new properties for the column.

Multiple types of properties are inferred during analysis:

– Data type

  An inferred Data type property describes the structural format of data in a column. For example, columns that contain numeric data are type N (numeric), and columns that contain alphabetic data are type A (alphabetic).

– Length

  The minimum, median, average, and maximum lengths of the values in a column.

– Precision

  The total number of digits in a column.

– Scale

  The total number of digits to the right of a decimal point in a column.

– Nulls

  If the percentage of null values in a column is equal to or greater than the system-defined threshold, the null property is inferred. The system-defined threshold is a setting that you can modify to allow a specific amount of data variances.

– Unique

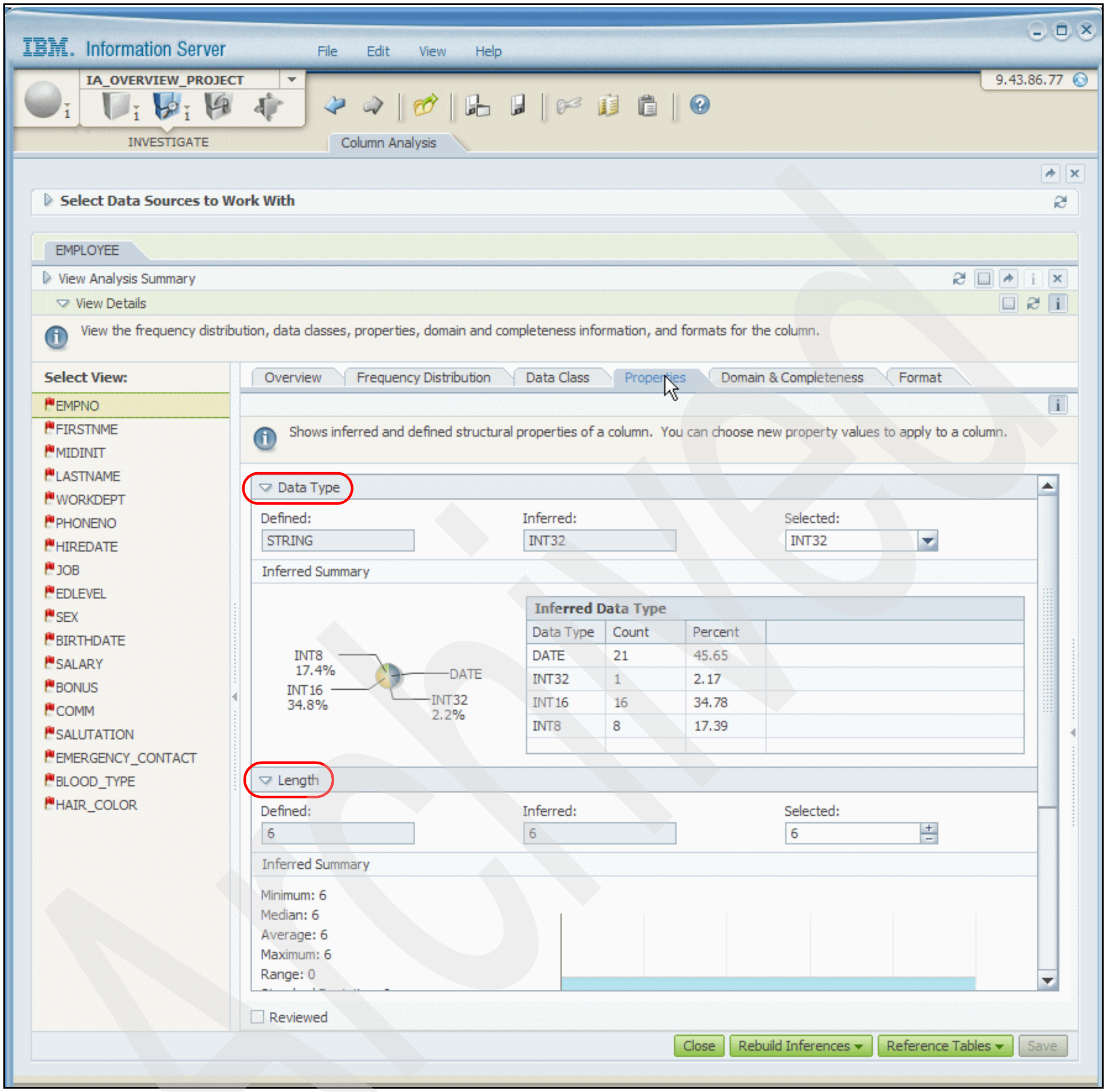
  If a column has a cardinality percentage that is equal to or greater than the system-defined threshold, the Unique property is inferred. Cardinality refers to the number of distinct values in a column including blanks and nulls.

– Constant

  If the highest frequency percentage for a single data value is equal to or greater than the system-defined threshold, the constant property is inferred.

Figure 1-82 on page 121 through Figure 1-85 on page 124 show the defined, inferred, and selected properties of the EMPNO column.

You can choose new property values to apply to a column.



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Data Type

Defined: STRING
Inferred: INT32
Selected: INT32

Inferred Data Type

Data Type	Count	Percent
DATE	21	45.65
INT32	1	2.17
INT16	16	34.78
INT8	8	17.39

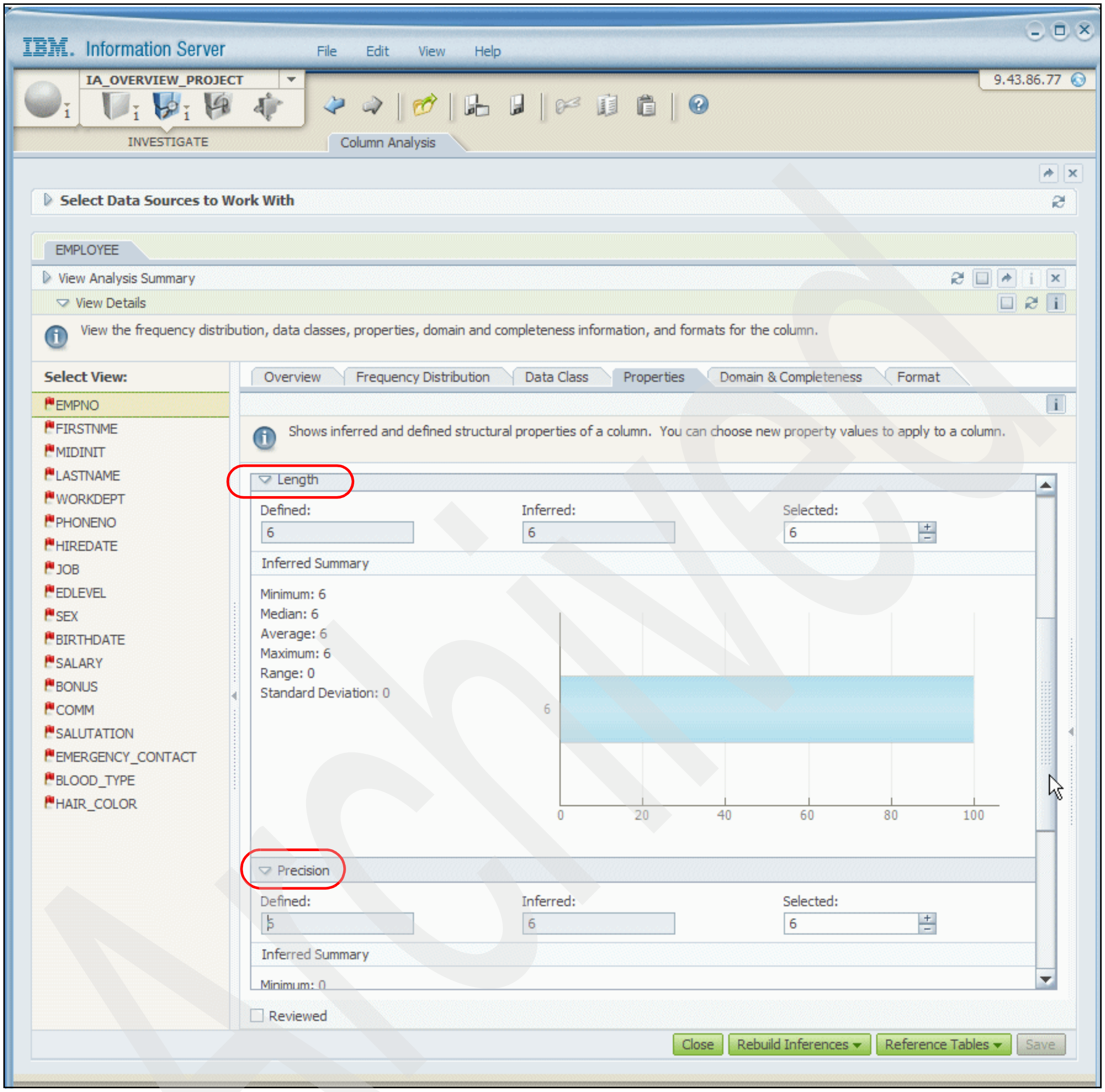
Length

Defined: 6
Inferred: 6
Selected: 6

Minimum: 6
Median: 6
Average: 6
Maximum: 6
Range: 0

☐ Reviewed

Figure 1-82 Properties view for the EMPNO column in the EMPLOYEE table 1/4



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Length

Defined: 6 Inferred: 6 Selected: 6

Minimum: 6
Median: 6
Average: 6
Maximum: 6
Range: 0
Standard Deviation: 0

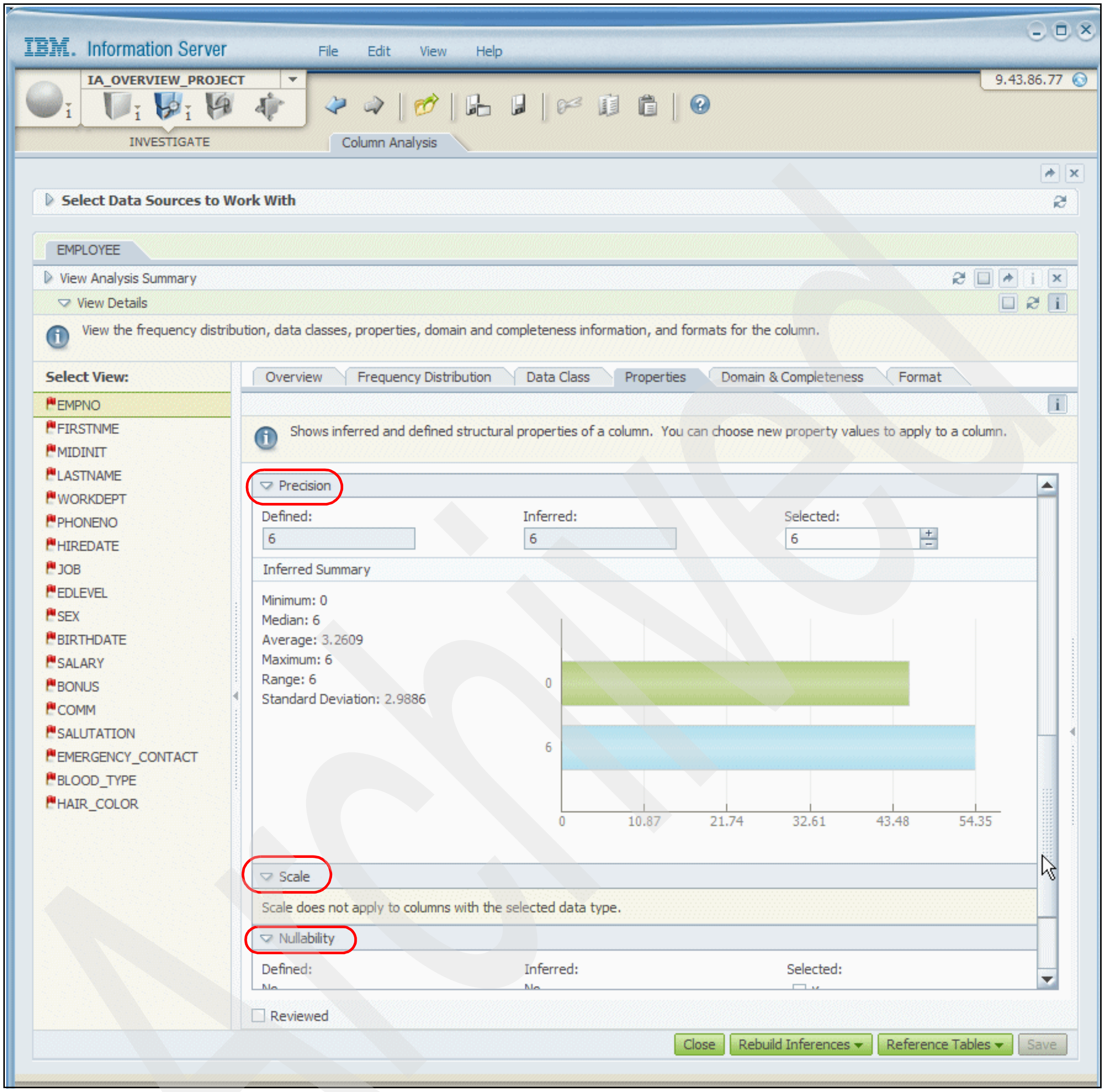
Precision

Defined: 5 Inferred: 6 Selected: 6

Minimum: 0

□ Reviewed

Figure 1-83 Properties view for the EMPNO column in the EMPLOYEE table 2/4



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Precision

Defined: 6 Inferred: 6 Selected: 6

Inferred Summary

Minimum: 0
Median: 6
Average: 3.2609
Maximum: 6
Range: 6
Standard Deviation: 2.9886

Scale

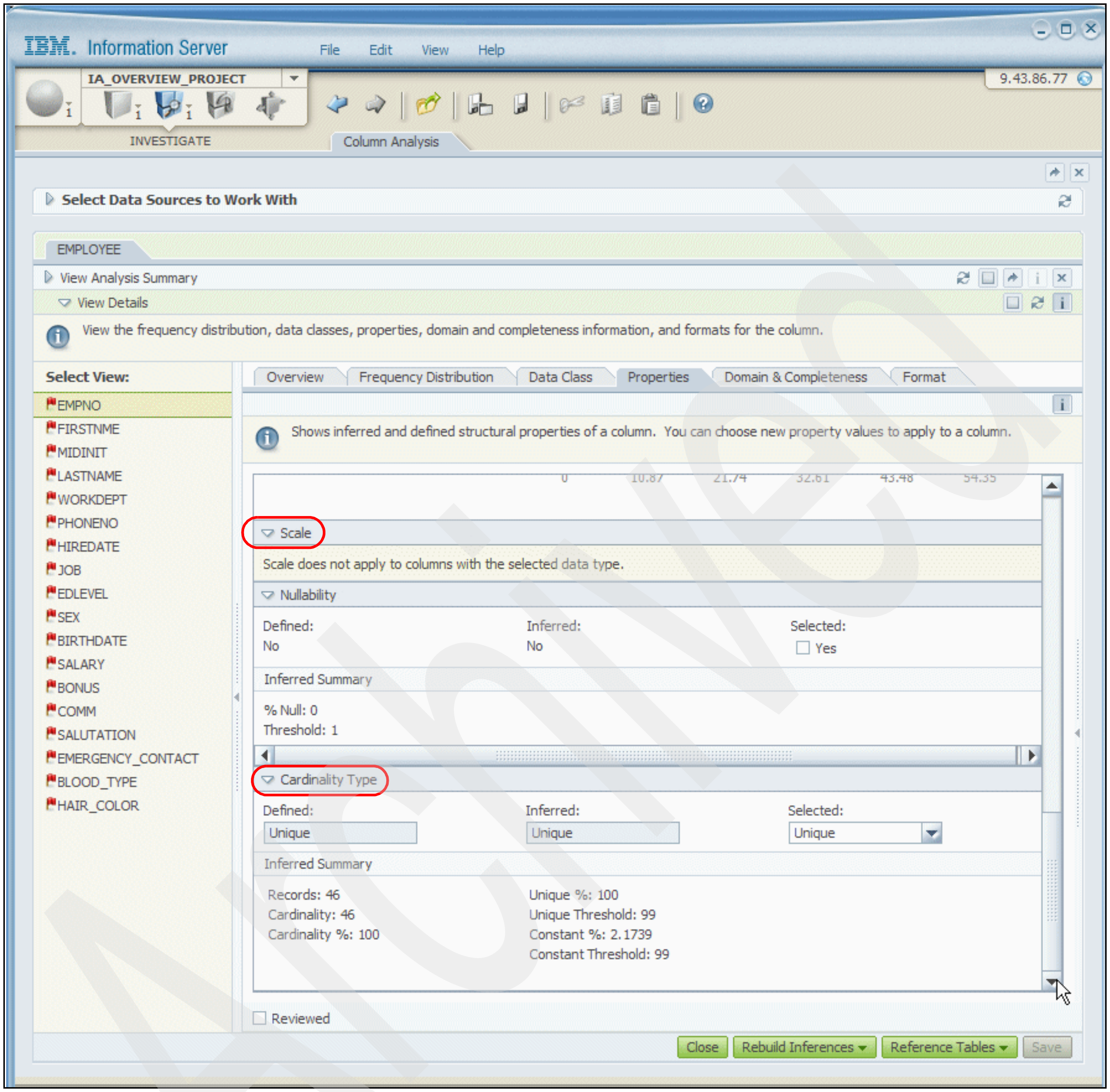
Scale does not apply to columns with the selected data type.

Nullability

Defined: Inferred: Selected:

□ Reviewed

Figure 1-84 Properties view for the EMPNO column in the EMPLOYEE table 3/4



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Scale

Scale does not apply to columns with the selected data type.

Nullability

Defined:	Inferred:	Selected:
No	No	□ Yes

Inferred Summary

% Null: 0
Threshold: 1

Cardinality Type

Defined:	Inferred:	Selected:
Unique	Unique	Unique

Inferred Summary

Records: 46	Unique %: 100
Cardinality: 46	Unique Threshold: 99
Cardinality %: 100	Constant %: 2.1739
	Constant Threshold: 99

□ Reviewed

Figure 1-85 Properties view for the EMPNO column in the EMPLOYEE table 4/4

- Domain & Completeness

  This view shows column frequency distribution values for identification of default or invalid values. You can use the results from a column analysis job to learn about the types of values in your data and to search for invalid, incomplete, or empty values in your data. To determine if your data is complete and valid, you can review the frequency distribution that is generated for each column in your data. The frequency distribution lists inferences about the valid and invalid values and maximum or minimum field lengths that were found during analysis. An invalid value might suggest that data in the field is missing or that an element of the column, such as the data type, is incorrect. If a field is incomplete or empty, the length of the field will be less than the minimum length that is allowed. The length will be zero for empty fields. You accept or reject the inferences in the frequency distribution.

  You can add a new value to the list in the frequency distribution if a value was not included in your imported metadata. All values are saved in the frequency distribution. You can then choose the method that you will use to examine the values.

  You choose one of three domain analysis methods to examine data values:

  - Value

    Select if you want to examine each value individually. Examine each data value for correct content and structure.

  - Range

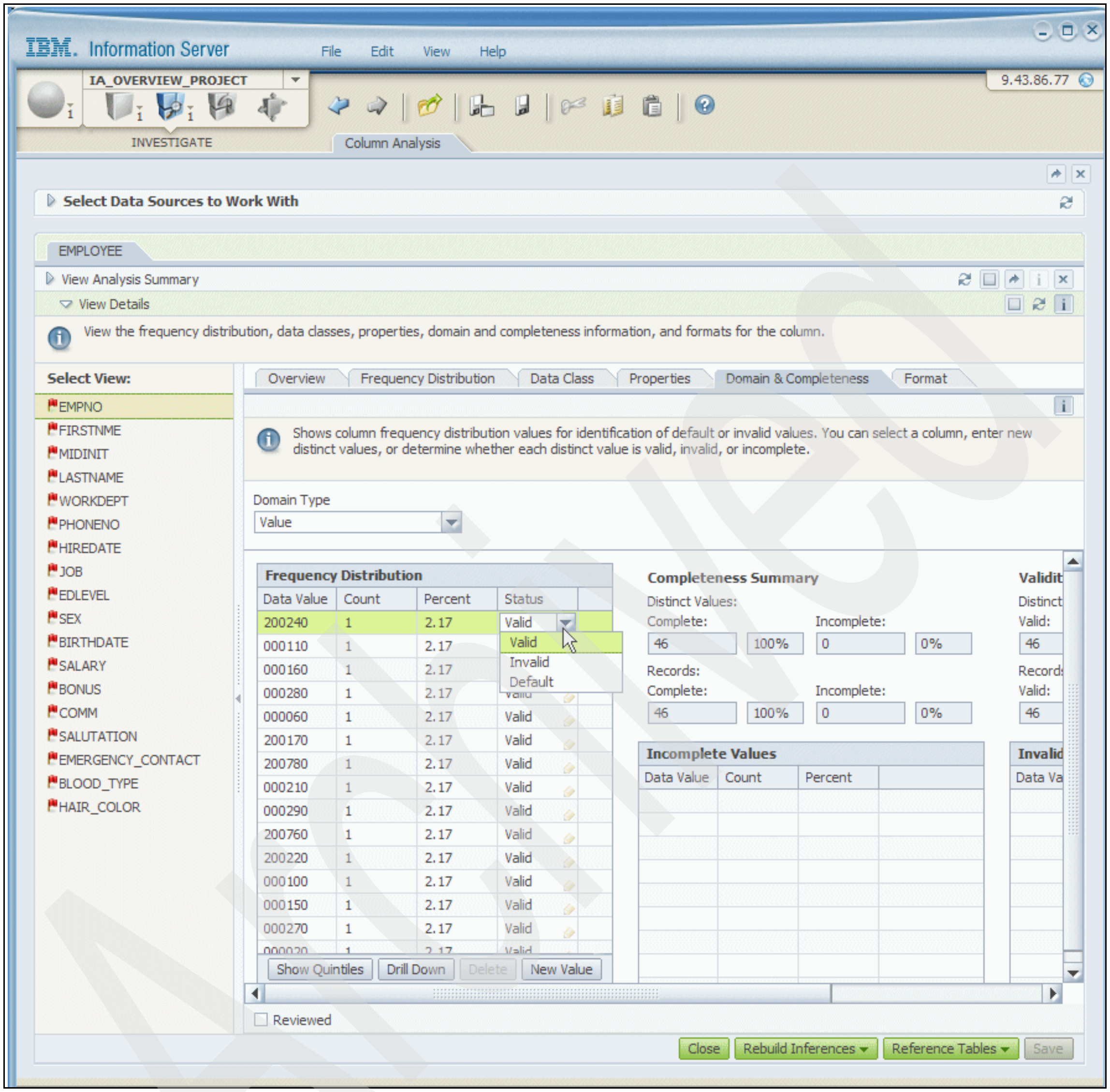
    Select if you want to specify a range to determine the validity of the values. For example, you can specify a minimum and maximum value for a date such as 10/10/07 and 10/10/08. Data values that are lower or higher than the minimum and maximum date values are invalid.

  - Reference Table

    Select whether you want to compare all of the values to the values in a reference table.

  After you evaluate your data, you do not have to make the same decisions again when you run subsequent analyses on the same data. The frequency distribution is saved in the metadata repository and is available if you want to make changes to it.

  Figure 1-86 on page 126 shows domain and completeness details for the EMPNO column.



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:
EMPNO
FIRSTNME
MIDINIT
LASTNAME
WORKDEPT
PHONENO
HIREDATE
JOB
EDLEVEL
SEX
BIRTHDATE
SALARY
BONUS
COMM
SALUTATION
EMERGENCY_CONTACT
BLOOD_TYPE
HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
200240	1	2.17	Valid
000110	1	2.17	Valid
000160	1	2.17	Invalid
000280	1	2.17	Default
000060	1	2.17	Valid
200170	1	2.17	Valid
200780	1	2.17	Valid
000210	1	2.17	Valid
000290	1	2.17	Valid
200760	1	2.17	Valid
200220	1	2.17	Valid
000100	1	2.17	Valid
000150	1	2.17	Valid
000270	1	2.17	Valid
000020	1	2.17	Valid

Completeness Summary

Distinct Values:
Complete: 46 100%
Incomplete: 0 0%

Records:
Complete: 46 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 46

Record
Valid: 46

Incomplete Values

Data Value	Count	Percent

Invalid

Data Va

Figure 1-86 Domain & Completeness view for the EMPNO column in the EMPLOYEE table

- Format

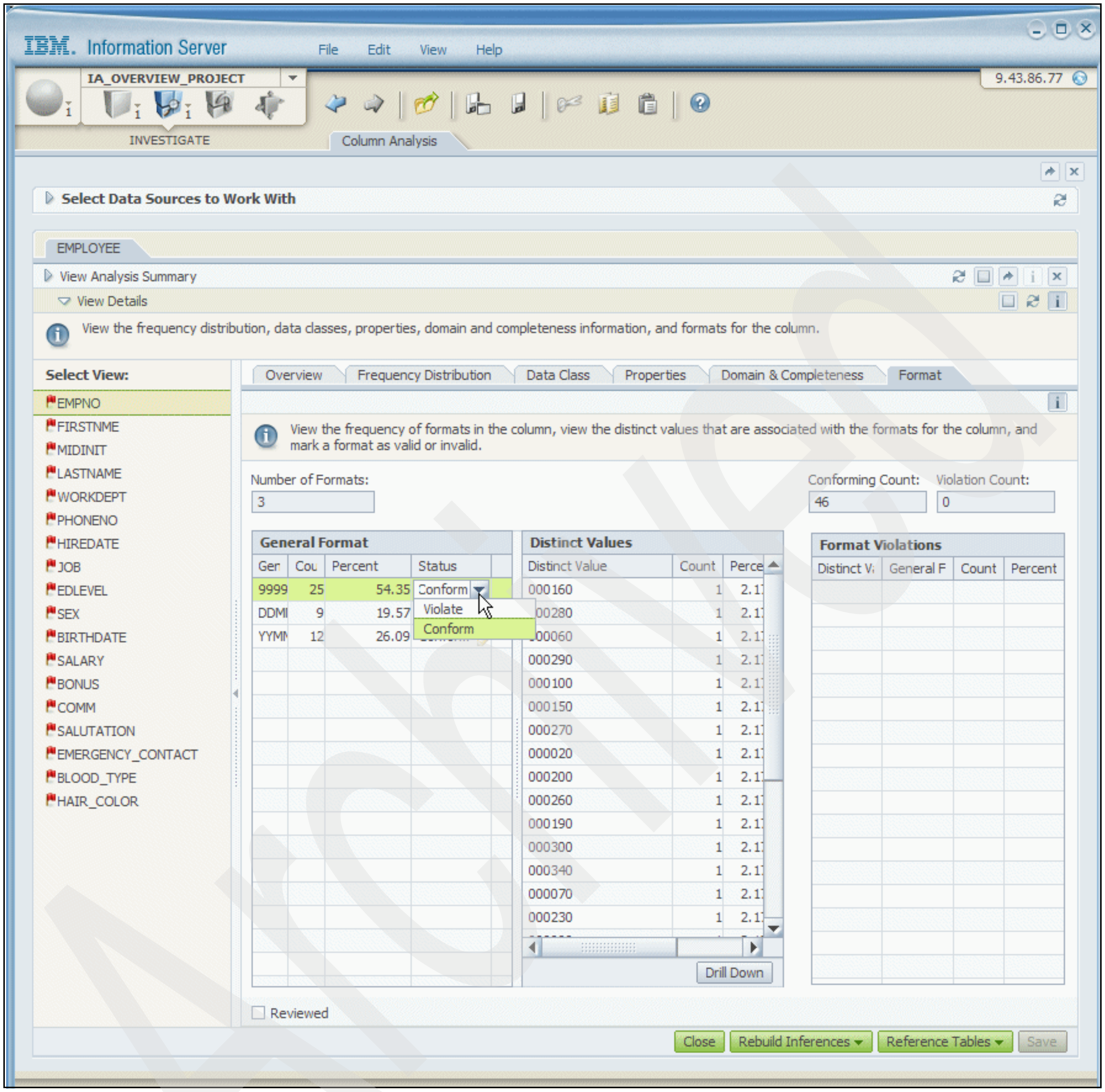
  This view shows the frequency of formats in the column and the distinct values that are associated with the formats for the column.

  To create a format expression for a column, the frequency distribution of the column is evaluated and each value in the column is translated into a format symbol. The format symbols create the format expression for the column. For example, alphabetic characters are represented by an A or a, depending on whether they are upper or lower case. Numerals are represented by the number *9* and special characters and spaces represent themselves.

  During a column analysis job, a format expression is created for each value in a column. A format expression is a pattern that describes the type of data in a column according to the type of the column. For example, if a column is of the class Date, a format expression such as YYMMDD might be assigned to the column (year, month, and day).

  This view permits you to mark a format as valid (Conform) or invalid (Violate).

  Figure 1-87 on page 128 shows the format analysis for the EMPNO column. Since EMPNO is a numeric field and of length six, it compares the pattern to a set of defined date formats to identify potential date inferences, which appears here. However the predominant inference is numeric (54.35%), not date.



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the frequency of formats in the column, view the distinct values that are associated with the formats for the column, and mark a format as valid or invalid.

Number of Formats: 3

Conforming Count: 46

Violation Count: 0

General Format

Ger	Cou	Percent	Status
9999	25	54.35	Conform
DDMI	9	19.57	
YYMI	12	26.09	

Distinct Values

Distinct Value	Count	Perce
000160	1	2.1
00280	1	2.1
00060	1	2.1
000290	1	2.1
000100	1	2.1
000150	1	2.1
000270	1	2.1
000020	1	2.1
000200	1	2.1
000260	1	2.1
000190	1	2.1
000300	1	2.1
000340	1	2.1
000070	1	2.1
000230	1	2.1

Format Violations

Distinct V	General F	Count	Percent

Figure 1-87 Format view for the EMPNO column in the EMPLOYEE table

Attention: When working with the detailed output from Column Analysis, you might find it useful to organize the data based on the inferred Data Classifications. Doing so allows you to concentrate on particular characteristics of the data rather than letting the often cryptic Column Name guide your work. The established project goals might dictate that certain types of data are more critical for analysis than others. Table 1-8 outlines the system inferred Data Classes, typical expectations of data within that class, and common conditions to analyze and assess.

Table 1-8 Data classifications, their definition, and assessment of data integrity

Inferred Data Classification	Definition of the Classification	Expectation of the Classification	Assessment of data integrity based on IBM WebSphere Information Analyzer Column Analysis profiling job results
Unknown	Field cannot be systematically identified.	Almost never populated and cannot be classified.	► From the metadata definitions, assess whether the field is understood correctly and whether the field is used. ► If nulls, blank values, and blank fields, consider annotating as an unused field. ► If single constant value, consider annotating as potentially redundant or irrelevant data.

Inferred Data Classification	Definition of the Classification	Expectation of the Classification	Assessment of data integrity based on IBM WebSphere Information Analyzer Column Analysis profiling job results
Identifier[a]	Uniquely identifying value for a record such as primary key, surrogate key, and natural identifiers such as name.	Unique values, constant format and always populated.	► **Data Type**: Should be Constant whether its Numeric or Character data type. ► **Length**: Select only the length that is needed since excess length negatively impacts storage and processing. ► **Cardinality Type**: Ensure uniqueness is set and understood. Use this to denote unique identifiers that are not Primary Keys. ► **Nulls and Duplicates**: Missing values prevent the identification of relationships between data in different tables and sources. Duplicate values create incorrect relationships of data. Review frequency distribution and domain analysis. ► **Invalid Format and Value Out-of-Range**: Both conditions can prevent correct handling of data. Review Frequency Distribution and Domain Analysis. Check Quintile Analysis for low-end/high-end values. Natural Identifiers such as social security number (SSN), tax identification number (TIN), and license numbers can have multiple usages, complex non-standardized formats (text-based identifiers), and unexpected duplicates. For example: – First three digits of a SSN identifies location which can be used, and formats vary from nine digits numeric to eleven character 999-99-9999. – Product/Part Codes often contain embedded business logic or information such as vendor or supplier code. Use Format Analysis to identify discrepancies. – Driver's license number varies by state, while some still use SSN.

Inferred Data Classification	Definition of the Classification	Expectation of the Classification	Assessment of data integrity based on IBM WebSphere Information Analyzer Column Analysis profiling job results
Indicator[b]	Flags or binary values such as M/F, True/False, Yes/No, 0/1. Often trigger actions elsewhere — sometimes set conditional situations such as only females have obstetric procedures.	Two concise single character binary values, preferably understandable	► **Data Type**: Should be Constant whether its Numeric or Character data type. ► **Length**: Minimize length without losing understandability such as Yes to Y, and No to N. ► **Cardinality Type**: Ensure that the constraint is understood. Should be neither Unique nor Constant. ► **Nulls**: If present or used, it is more likely to impact correct system behavior such as a failure to trigger specific business events or conditions. ► **Skewed values**: Understand when skewed values expected such as more active bank accounts than inactive accounts, while in distributed populations males and females should be more or less equally distributed. ► **Accuracy**: Identify master reference and follow up with Cross Domain Analysis (described in 1.10, “Cross domain analysis” on page 301) to confirm. ► **Understandability**: M/F more understandable than 0/1 for Gender. Consistency an issue with mergers and migrations. Use Cross Domain Analysis to confirm consistency. ► **Review Naming**: FLAG in metadata is a clue; reset Classification as needed.
Code[c]	Finite set of values, usually less than 100 values. Indicates states of action such as ordered, cancelled and shipped. Sometimes check digit for other fields, or shorthand for a reference such as zip code for a location.	Understandable (values N, P, L, and Y might not be understandable) and concise.	► **Data Type**: Should be Constant (usually String or Integer data type). ► **Length**: Minimize length without losing understandability. ► **Cardinality Type**: Ensure that the constraint is understood. Should be neither Unique nor Constant. ► **Nulls**: If present or used, it is more likely to impact correct system behavior such as a failure to trigger specific business events or conditions. Validity and accuracy by identifying the master reference and follow up with Cross Domain Analysis to confirm. Consistency an issue with mergers and migrations. Use Cross Domain Analysis to confirm consistency. ► **Review Naming**: CODE or CD in metadata is a clue; reset Classification as needed.

Inferred Data Classification	Definition of the Classification	Expectation of the Classification	Assessment of data integrity based on IBM WebSphere Information Analyzer Column Analysis profiling job results
Quantity[d]	Potentially infinite set of numeric values (integers, decimals, floating values); can be positive or negative. Includes quantities, prices, currency values.	A ranged set of data of consistent numeric representation. Can be externally entered or calculated. Defaults only for values that are triggered by subsequent system events such as Quantity Shipped starts at 0.	► **Data Type**: Should be Numeric (Integer, Decimal or Float). Review consistency of representation. Flat file sources can be seen as character or string instead of numeric data type. Watch for an unknown data type which typically signals the presence of nulls or spaces. ► **Precision** (total numeric length): Numeric data identified as String or Character data type shows Length, not Precision. Review the defined precision length versus utilized. ► **Scale** (decimal length): Numeric data identified as String or Character data type shows no Scale value. Review defined Scale versus utilized. ► **Nulls**: If present or used, impacts inferencing of data classification. Can also impact correct system behavior if it is reported incorrectly. ► **Zeroes**: If not valid, it is likely to impact calculations in other quantities. ► **Skewed values**: Understand when skewed values expected; common versus occasional conditions. Most individual orders are small, while institutional orders can be large but rare. Most salaries are within a typical range, but outliers are not unexpected. Skewed values are not expected with quantities that represent standard rates or fairly constant values such as shipping charges and tax rates. ► **Accuracy**: Determine what is reasonable by reviewing distribution, whether or not there should be negative values (asset values, prices, and salaries are not negative but sales can include returns which would be represented by negative values). Also determine what the permitted maximum values are. You need to identify relevant documentation for valid range of values. Consistency might be an issue with movements and calculations. On directly transferred data, Cross Domain Analysis can provide insight into consistency. ► **Review Naming**: VAL, QTY, PRC in metadata is a clue; reset Classification as needed.

Inferred Data Classification	Definition of the Classification	Expectation of the Classification	Assessment of data integrity based on IBM WebSphere Information Analyzer Column Analysis profiling job results
Date/Time[e]	Generally bounded set of calendar dates or timestamps	Externally entered or calculated. Can be defaults.	► **Data Type**: Should be Date. Review consistency of representation. Flat file sources can be seen as character or string instead of date data type. Watch for an unknown data type which typically signals the presence of nulls or spaces. ► **Length**: Standard length for date is 8 digits. Watch for inconsistent lengths. ► **Format**: Common problem with dates, multiple representations. Validate consistency of format usage. ► **Nulls and spaces**: If present or used, impacts inferencing of data classification. Can also impact correct system behavior if it is incorrectly reported. ► **Zeroes or defaults**: If not valid, it is likely to impact usage. Look for one or a couple of high frequency values (such as 19000101, or 19500101). The value is a "valid" date but really a default. ► **Skewed values**: Understand when skewed values expected. Cyclical occurrences such as billing cycle dates (once per month) and salary pay dates. Skewed values are not expected with dates that represent standard entry points such as entry/creation dates and birth dates in a general population. Watch for default entries. ► **Accuracy**: Determine what is reasonable by reviewing distribution, what the oldest date is, and the most recent date. You need to identify relevant documentation for valid range of dates. Consistency might be an issue with movements and calculations. On directly transferred data, Cross-Domain Analysis can provide insight into consistency.

Inferred Data Classification	Definition of the Classification	Expectation of the Classification	Assessment of data integrity based on IBM WebSphere Information Analyzer Column Analysis profiling job results
Text	Usually free-form string or alphanumeric data such as name and addresses.	Most are unique	► Are there any frequently occurring values including defaults. ► Focus on data formats. Are there common formats, special characters such as (,),/,#,* (might be executable code), statements such as DO NOT USE, and lack of standardization. ► Usage of Text fields. Is it well parsed or commingled data or domains? Can indicate a requirement for domain/field conditioning — for example, does AddressLine1 contain only street information or also city and state? ► Does it contain single or multiple entries/subjects. For example, does Name include a single individual name, or multiple name, or an organization or legal entity? ► Recommend additional analysis using IBM WebSphere QualityStage, which will be covered in an upcoming IBM Redbooks publication.
Large Object			Not assessed within IBM WebSphere Information Analyzer. Generally represents long descriptive entries, pictures/images, or other content specific data.

a. Not all identifiers are classified correctly. Set as Text if length >15 and cardinality < threshold (usually 98%)

b. Not all indicators are classified correctly. Inferred as Code if the number of distinct values > 2.

c. Not all codes are classified correctly. Inferred as Indicator if the number of distinct values = 2; there can be more values, just not currently in use.

d. Not all quantities are classified correctly. Inferred as Code if the number of distinct values is low; or inferred as Text if the source is a flat file and the number of distinct values is high; presence of nulls/spaces affects classification

e. Not all dates are classified correctly. Inferred as Quantity if nonstandard format; or inferred as Text if the source is a flat file and there are alpha characters or nulls/spaces which affect classification

1.7.3 Column analysis usage scenario

The modified sample database described in Figure 1-11 on page 31 is used in the column analysis examples that we describe here.

In this section, we describe how to run a column analysis job, followed by a review of the column analysis results. Also included is the setting of invalid values in the frequency distribution view, the generation of reference tables, validating data values using a reference table, and analyzing virtual columns.

Attention: In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

Run a column analysis on a single table

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), run a column analysis job on the EMPLOYEE table as follows:

1. On the Investigate navigator menu in the console, select **Column Analysis** as shown in Figure 1-88 on page 138.
2. On the Column Analysis workspace, select the EMPLOYEE table and click **Run Column Analysis** in the Tasks pane as shown in Figure 1-89 on page 139.
3. On the Run Column Analysis pane, type a new name for this analysis job in the Job Name field such as EMPLOYEE_CA. Optionally, in the Job Description field, type a description for this job such as Analysis Job as shown in Figure 1-89 on page 139.
4. On the Scheduler tab, you can choose to schedule the job at an appropriate time. While we actually clicked **Run Now**, we selected **Schedule** in order to view the available options for scheduling such as the start and end dates (with times) and the frequency of executing this job as shown in Figure 1-90 on page 140.
5. On the Sample tab, you can choose to perform the column analysis on a sample rather than the full volume of data by selecting **Use Sample**. You can then choose the total sample size and the type of sampling to use as shown in Figure 1-91 on page 141. We actually chose to run on the full volume of data.

   Note: Sampling is a useful profiling technique to gain insights into the data without requiring the analysis of full data volume and the associated costs of system processing and storage.

6. On the Options tab, you can specify miscellaneous runtime options, such as the array size, auto commit, and retaining scripts. Use this tab to specify whether the Frequency Distribution table should be updated with subsequent analysis jobs and whether you want to retain the scripts that are used to generate this analysis job. We chose the options as shown in Figure 1-92 on page 142.

A brief review of these options follows:

– Array Size

  This parameter controls the number of data rows to group together as a single operation, in order to provide more performance write operations. Increasing the array size will increase the number of elements included in each INSERT statement. It is important to note that increasing the array size will also increase the memory consumption, which if not used properly, could lead to performance degradation.

  The default setting for this parameter is 2000.

– Auto Commit

  Signifies whether or not all of the added rows are committed immediately upon insertion, or only after all of the rows are processed.

– Isolation Level

  Represents the standard isolation levels. These levels are 4 (Serializable), 3 (Repeatable read), 2 (Read committed), and 1 (Read uncommitted)

– Update Existing Tables

  If you are running a column analysis job on a column that was already analyzed, specify whether the frequency distribution table for that column should be updated during this column analysis job or whether a new frequency distribution table should be created.

  - If *Yes*, then if column analysis has previously been run, the frequency distribution table will be updated with the new frequencies, but the values and inferences will be retained.
  - If *No*, then any prior frequency distribution tables (and associated inferences) will be dropped and the frequency table will be created as new, with no prior inferences retained.

– Retain Scripts

  Specify whether you want to keep the script for this column analysis job. You might want to keep the script to debug a failure with the job.

Note: You will not generally need to check the Retain Scripts or Retain Datasets options as these consume system resources, unless assessment of the job process is needed after the job completes.

7. Click **Submit and Close** to run the analysis job as shown in Figure 1-93 on page 143.

8. Click **Details** to view the status of the job submission as shown in Figure 1-94 on page 144. While the job EMPLOYEE_CA is executing, the In Progress and Executing status is shown (see Figure 1-95 on page 145).

   You can view the progress of the job execution in the Log Views in the IBM Information Server console (preferred approach since it is available when you have only purchased IBM WebSphere Information Analyzer and not IBM WebSphere DataStage), or the DataStage and QualityStage Director which is provided with IBM WebSphere DataStage.

   – Log Views in the IBM Information Server console

     Log Views under the OPERATE menu should be considered the primary review and debugging tool.

     Figure 1-96 on page 146 through Figure 1-99 on page 148 describe some of the main steps in using Log Views as follows:

     - From the OPERATE menu, click **Log Views** as shown in Figure 1-96 on page 146.
     - In the Log View tab, click **New Log View** in the Tasks column as shown in Figure 1-97 on page 146.
     - Provide the parameters for the Log View for IBM WebSphere Information Analyzer as shown in Figure 1-98 on page 147. The Summary column shows summary of chosen options. Click **Save**.
     - Click **View Log** as shown in Figure 1-98 on page 147 to view the customized log view contents.

     Note: View the log for iasHandler information. If that log view does not contain relevant information, look at the log file IBM/Websphere/Application Server/profiles/default/iasHandler.log.

   – DataStage and QualityStage Director

     You can also view the progress of the job submission using the DataStage and QualityStage Director as shown in Figure 1-100 on page 149 through Figure 1-102 on page 150.

     If the Retain Scripts option (see Figure 1-92 on page 142) is not chosen, the job is a short living object in the DataStage and QualityStage Director and it will appear in the jobs logs while it is running. When the job completes, the system deletes the job, and you will not see it in the director.

     To continue to see the job in the job logs after job completion, you need to specify the Retain Scripts option.

> **Attention:** You can only view the progress of Column Analysis and Primary Key Analysis in DataStage and QualityStage Director.

Follow these steps:

i. After launching DataStage and QualityStage Director, view the jobs and their status as shown in Figure 1-100 on page 149 using the job name.

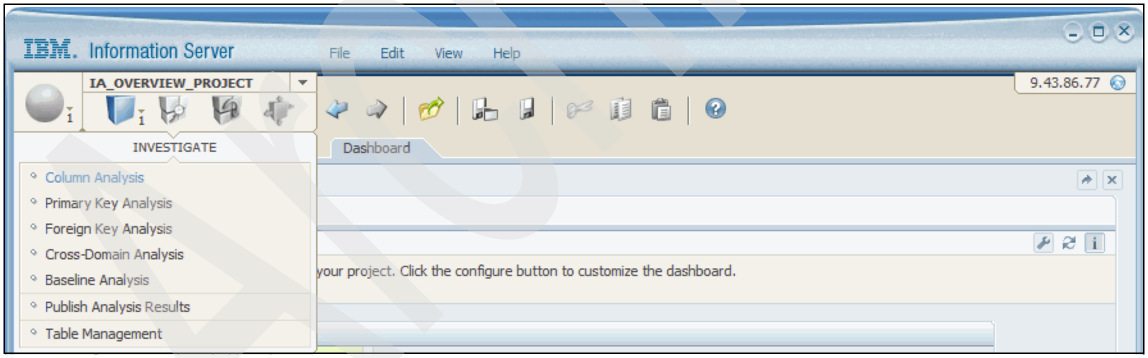
> **Note:** The job that we show here is different from the column analysis that is run on the EMPLOYEE table because it was run at a later date on another set of tables.

ii. To view the log of a particular job, right-click the job name and select **View Log** as shown in Figure 1-101 on page 149.

iii. The content of the log is shown in Figure 1-102 on page 150. To refresh the content of the log, click **Refresh** from the View menu.

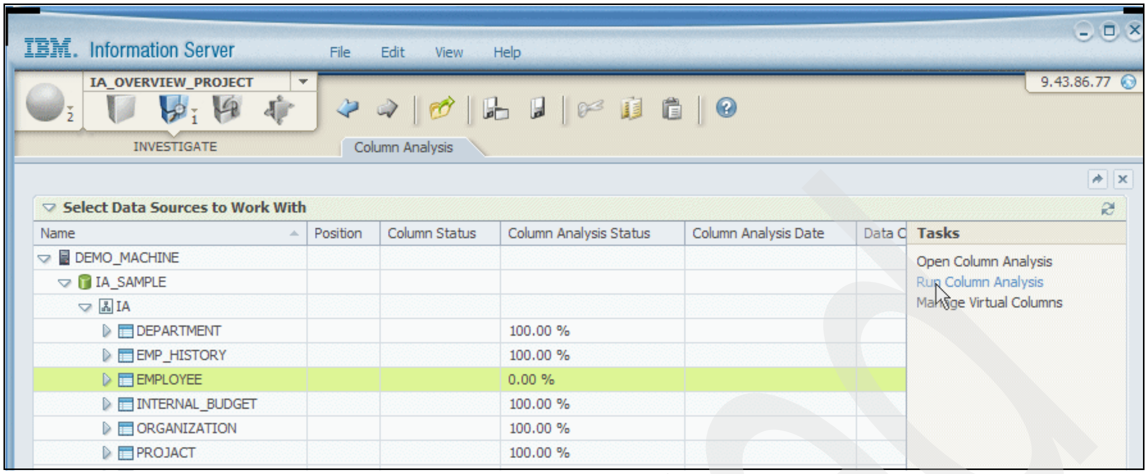
9. When the execution is complete, click **View Results** as shown in Figure 1-103 on page 154 to proceed to view the results of column analysis.

You can now proceed to view the results of column analysis as described in “Review column analysis results” on page 150.



your project. Click the configure button to customize the dashboard.

Figure 1-88 Run a column analysis job on a single table 1/8



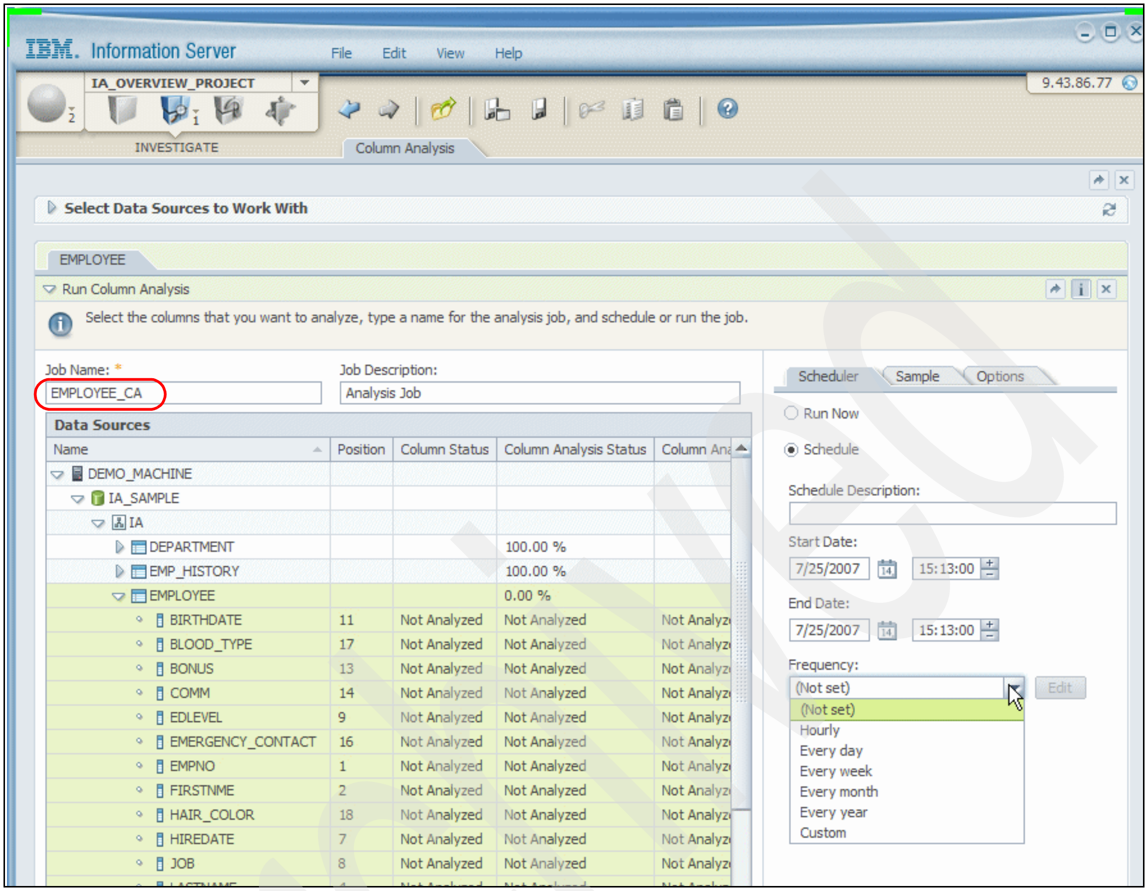
Select Data Sources to Work With

Name	Position	Column Status	Column Analysis Status	Column Analysis Date	Data C
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			0.00 %		
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		

Tasks

Open Column Analysis
Run Column Analysis
Manage Virtual Columns

Figure 1-89 Run a column analysis job on a single table 2/8



Select the columns that you want to analyze, type a name for the analysis job, and schedule or run the job.

Job Name: * EMPLOYEE_CA

Job Description: Analysis Job

Data Sources

Name	Position	Column Status	Column Analysis Status	Column Ana
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT			100.00 %	
EMP_HISTORY			100.00 %	
EMPLOYEE			0.00 %	
BIRTHDATE	11	Not Analyzed	Not Analyzed	Not Analyz
BLOOD_TYPE	17	Not Analyzed	Not Analyzed	Not Analyz
BONUS	13	Not Analyzed	Not Analyzed	Not Analyz
COMM	14	Not Analyzed	Not Analyzed	Not Analyz
EDLEVEL	9	Not Analyzed	Not Analyzed	Not Analyz
EMERGENCY_CONTACT	16	Not Analyzed	Not Analyzed	Not Analyz
EMPNO	1	Not Analyzed	Not Analyzed	Not Analyz
FIRSTNME	2	Not Analyzed	Not Analyzed	Not Analyz
HAIR_COLOR	18	Not Analyzed	Not Analyzed	Not Analyz
HIREDATE	7	Not Analyzed	Not Analyzed	Not Analyz
JOB	8	Not Analyzed	Not Analyzed	Not Analyz

○ Run Now

● Schedule

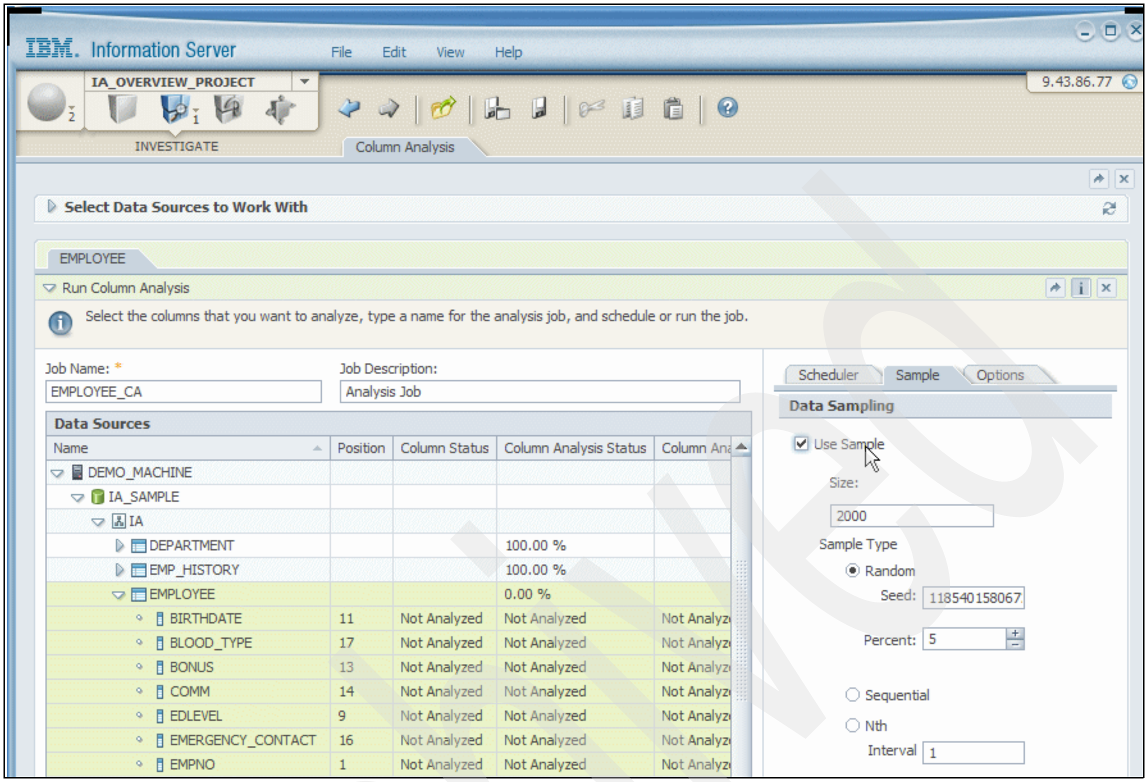
Schedule Description:

Start Date: 7/25/2007 15:13:00

End Date: 7/25/2007 15:13:00

Frequency: (Not set)

Figure 1-90 Run a column analysis job on a single table 3/8



Select Data Sources to Work With

EMPLOYEE

Run Column Analysis

Select the columns that you want to analyze, type a name for the analysis job, and schedule or run the job.

Job Name: * EMPLOYEE_CA

Job Description: Analysis Job

Data Sources

Name	Position	Column Status	Column Analysis Status	Column Ana
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT			100.00 %	
EMP_HISTORY			100.00 %	
EMPLOYEE			0.00 %	
BIRTHDATE	11	Not Analyzed	Not Analyzed	Not Analyz
BLOOD_TYPE	17	Not Analyzed	Not Analyzed	Not Analyz
BONUS	13	Not Analyzed	Not Analyzed	Not Analyz
COMM	14	Not Analyzed	Not Analyzed	Not Analyz
EDLEVEL	9	Not Analyzed	Not Analyzed	Not Analyz
EMERGENCY_CONTACT	16	Not Analyzed	Not Analyzed	Not Analyz
EMPNO	1	Not Analyzed	Not Analyzed	Not Analyz

Data Sampling

☑ Use Sample

Size: 2000

Sample Type

● Random

Seed: 118540158067

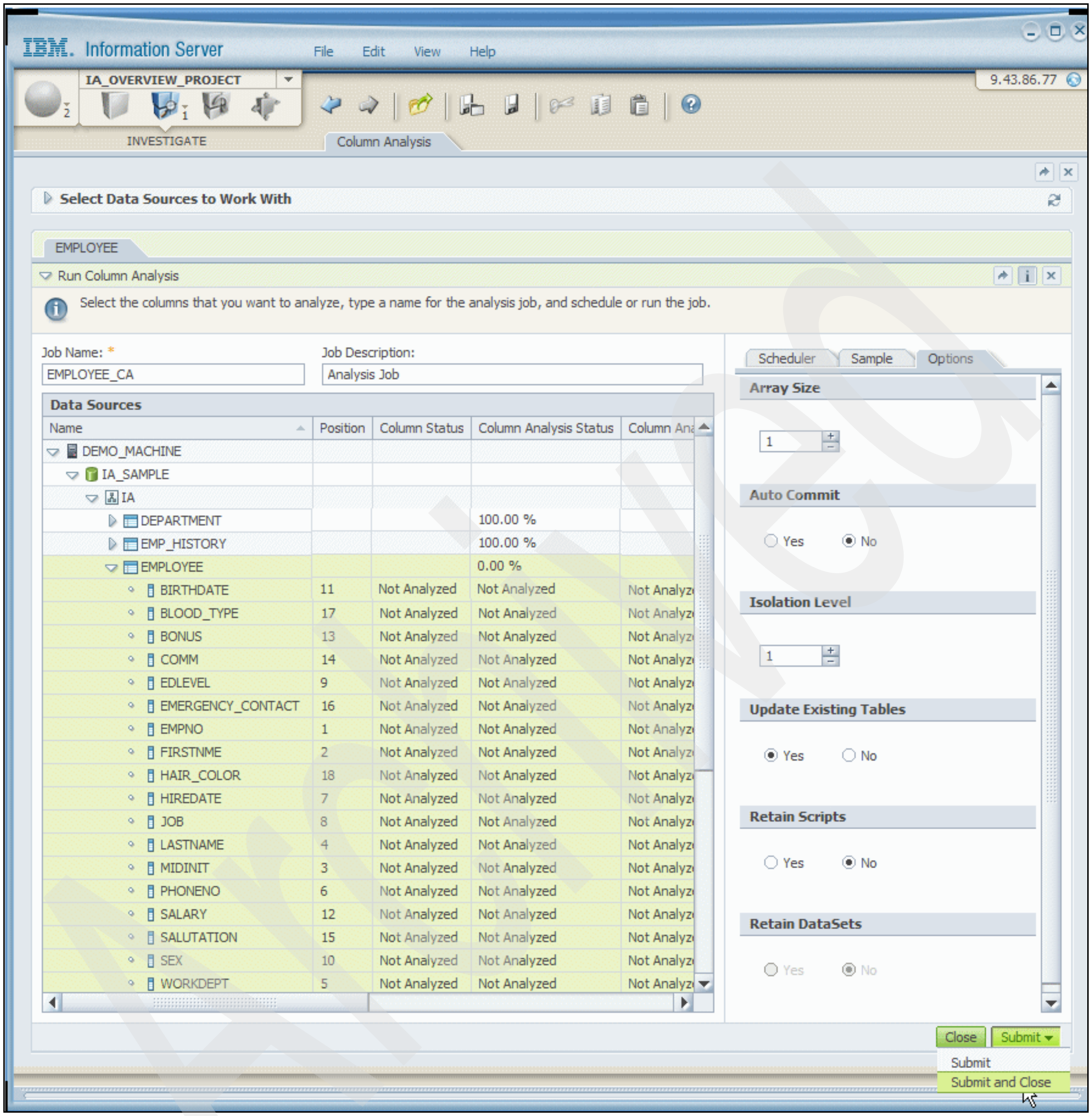
Percent: 5

○ Sequential

○ Nth

Interval 1

Figure 1-91 Run a column analysis job on a single table 4/8



Select Data Sources to Work With

EMPLOYEE

Run Column Analysis

Select the columns that you want to analyze, type a name for the analysis job, and schedule or run the job.

Job Name: * EMPLOYEE_CA

Job Description: Analysis Job

Data Sources

Name	Position	Column Status	Column Analysis Status	Column Ana
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT			100.00 %	
EMP_HISTORY			100.00 %	
EMPLOYEE			0.00 %	
BIRTHDATE	11	Not Analyzed	Not Analyzed	Not Analyz
BLOOD_TYPE	17	Not Analyzed	Not Analyzed	Not Analyz
BONUS	13	Not Analyzed	Not Analyzed	Not Analyz
COMM	14	Not Analyzed	Not Analyzed	Not Analyz
EDLEVEL	9	Not Analyzed	Not Analyzed	Not Analyz
EMERGENCY_CONTACT	16	Not Analyzed	Not Analyzed	Not Analyz
EMPNO	1	Not Analyzed	Not Analyzed	Not Analyz
FIRSTNME	2	Not Analyzed	Not Analyzed	Not Analyz
HAIR_COLOR	18	Not Analyzed	Not Analyzed	Not Analyz
HIREDATE	7	Not Analyzed	Not Analyzed	Not Analyz
JOB	8	Not Analyzed	Not Analyzed	Not Analyz
LASTNAME	4	Not Analyzed	Not Analyzed	Not Analyz
MIDINIT	3	Not Analyzed	Not Analyzed	Not Analyz
PHONENO	6	Not Analyzed	Not Analyzed	Not Analyz
SALARY	12	Not Analyzed	Not Analyzed	Not Analyz
SALUTATION	15	Not Analyzed	Not Analyzed	Not Analyz
SEX	10	Not Analyzed	Not Analyzed	Not Analyz
WORKDEPT	5	Not Analyzed	Not Analyzed	Not Analyz

Array Size: 1

Auto Commit: ○ Yes ● No

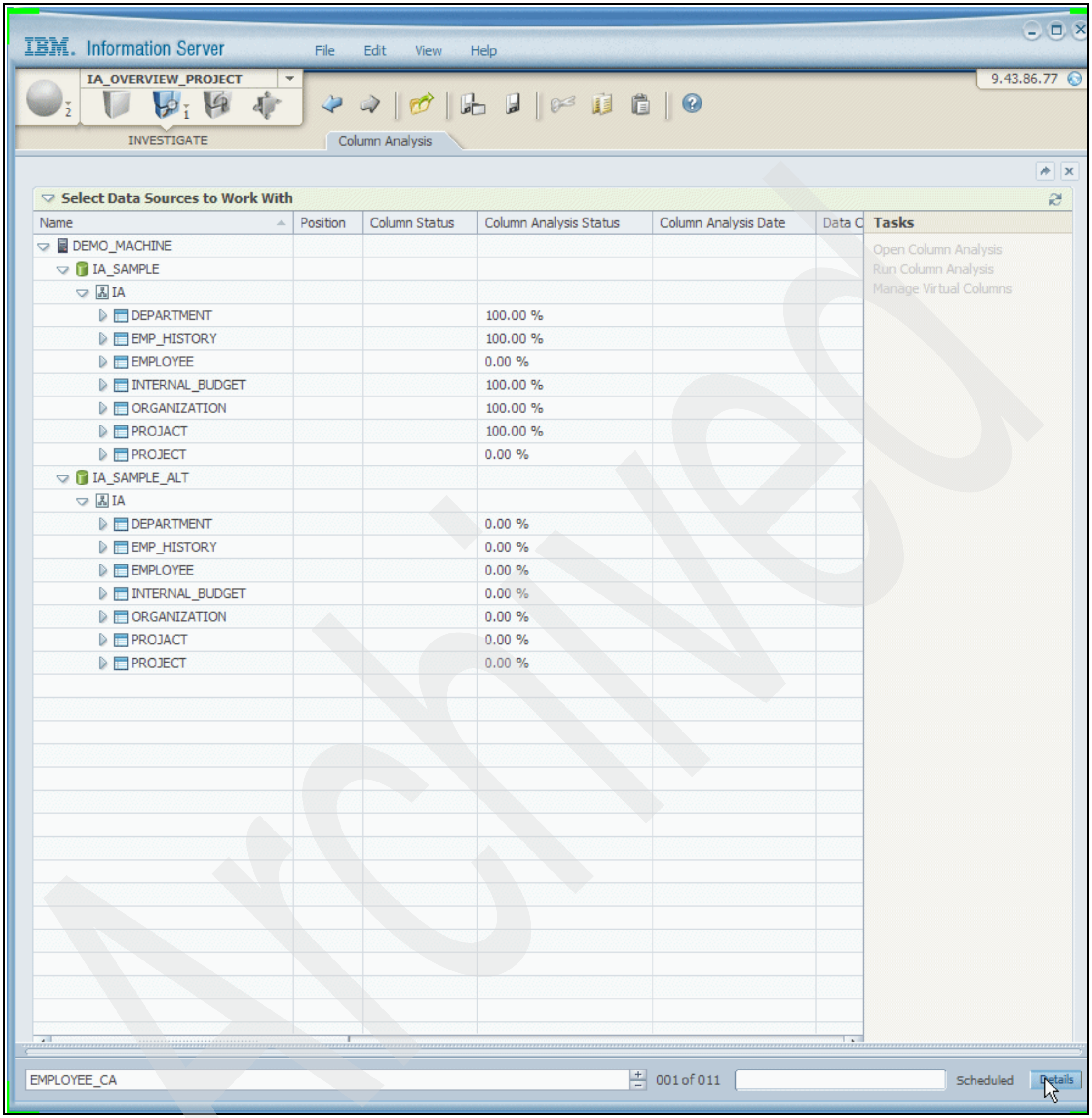
Isolation Level: 1

Update Existing Tables: ● Yes ○ No

Retain Scripts: ○ Yes ● No

Retain DataSets: ○ Yes ● No

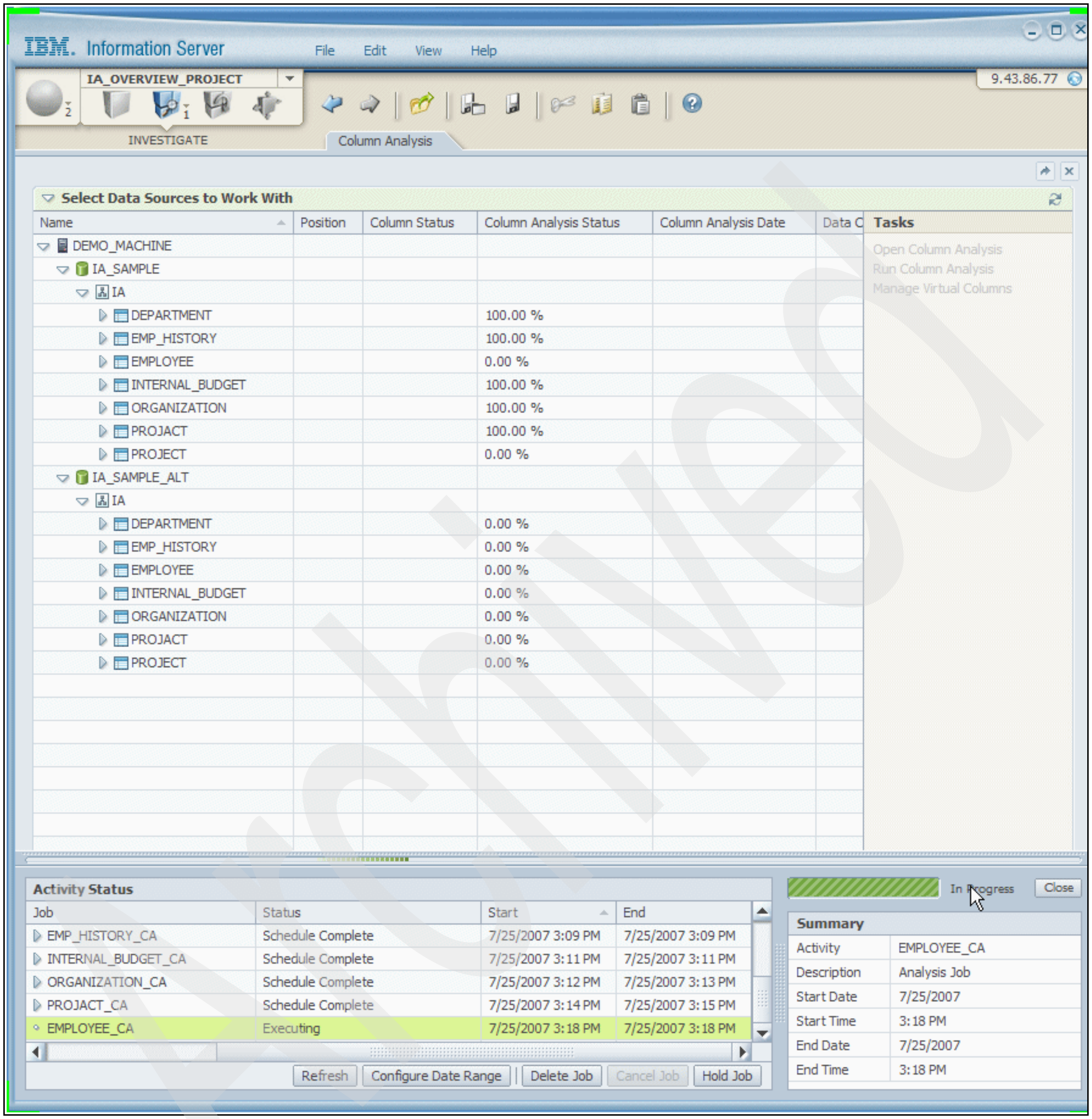
Figure 1-92 Run a column analysis job on a single table 5/8



Name	Position	Column Status	Column Analysis Status	Column Analysis Date	Data C
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			0.00 %		
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		
PROJECT			0.00 %		
IA_SAMPLE_ALT					
IA					
DEPARTMENT			0.00 %		
EMP_HISTORY			0.00 %		
EMPLOYEE			0.00 %		
INTERNAL_BUDGET			0.00 %		
ORGANIZATION			0.00 %		
PROJACT			0.00 %		
PROJECT			0.00 %		

EMPLOYEE_CA 001 of 011

Figure 1-93 Run a column analysis job on a single table 6/8



Select Data Sources to Work With

Name	Position	Column Status	Column Analysis Status	Column Analysis Date	Data C
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			0.00 %		
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		
PROJECT			0.00 %		
IA_SAMPLE_ALT					
IA					
DEPARTMENT			0.00 %		
EMP_HISTORY			0.00 %		
EMPLOYEE			0.00 %		
INTERNAL_BUDGET			0.00 %		
ORGANIZATION			0.00 %		
PROJACT			0.00 %		
PROJECT			0.00 %		

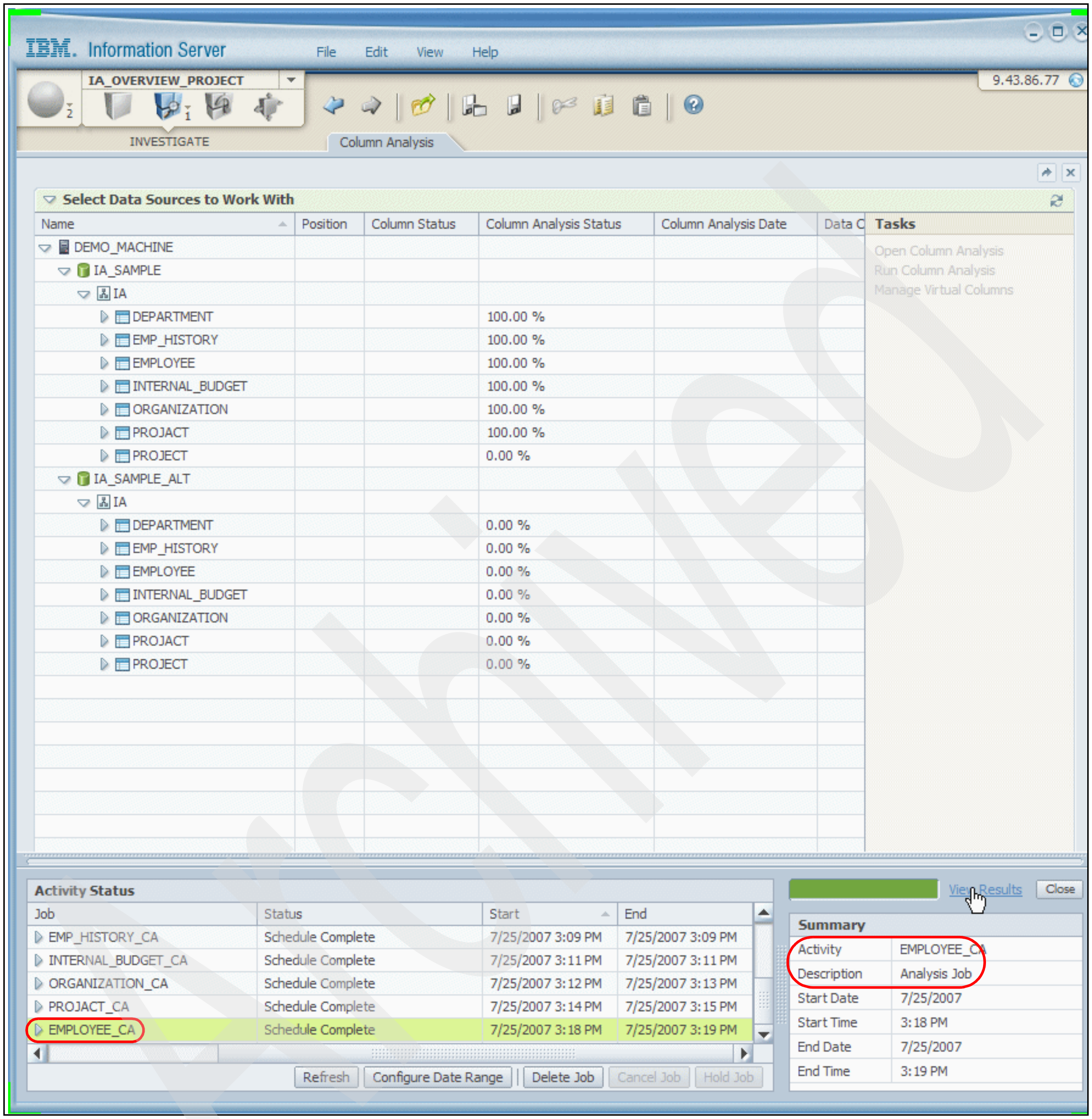
Activity Status

Job	Status	Start	End
EMP_HISTORY_CA	Schedule Complete	7/25/2007 3:09 PM	7/25/2007 3:09 PM
INTERNAL_BUDGET_CA	Schedule Complete	7/25/2007 3:11 PM	7/25/2007 3:11 PM
ORGANIZATION_CA	Schedule Complete	7/25/2007 3:12 PM	7/25/2007 3:13 PM
PROJACT_CA	Schedule Complete	7/25/2007 3:14 PM	7/25/2007 3:15 PM
EMPLOYEE_CA	Executing	7/25/2007 3:18 PM	7/25/2007 3:18 PM

Summary

Activity	EMPLOYEE_CA
Description	Analysis Job
Start Date	7/25/2007
Start Time	3:18 PM
End Date	7/25/2007
End Time	3:18 PM

Figure 1-94 Run a column analysis job on a single table 7/8



Select Data Sources to Work With

Name	Position	Column Status	Column Analysis Status	Column Analysis Date	Data C
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			100.00 %		
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		
PROJECT			0.00 %		
IA_SAMPLE_ALT					
IA					
DEPARTMENT			0.00 %		
EMP_HISTORY			0.00 %		
EMPLOYEE			0.00 %		
INTERNAL_BUDGET			0.00 %		
ORGANIZATION			0.00 %		
PROJACT			0.00 %		
PROJECT			0.00 %		

Activity Status

Job	Status	Start	End
EMP_HISTORY_CA	Schedule Complete	7/25/2007 3:09 PM	7/25/2007 3:09 PM
INTERNAL_BUDGET_CA	Schedule Complete	7/25/2007 3:11 PM	7/25/2007 3:11 PM
ORGANIZATION_CA	Schedule Complete	7/25/2007 3:12 PM	7/25/2007 3:13 PM
PROJACT_CA	Schedule Complete	7/25/2007 3:14 PM	7/25/2007 3:15 PM
EMPLOYEE_CA	Schedule Complete	7/25/2007 3:18 PM	7/25/2007 3:19 PM

Summary

Activity	EMPLOYEE_CA
Description	Analysis Job
Start Date	7/25/2007
Start Time	3:18 PM
End Date	7/25/2007
End Time	3:19 PM

Figure 1-95 Run a column analysis job on a single table 8/8

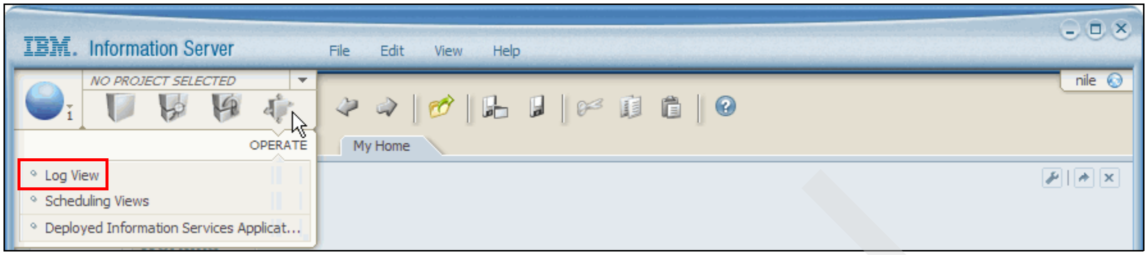
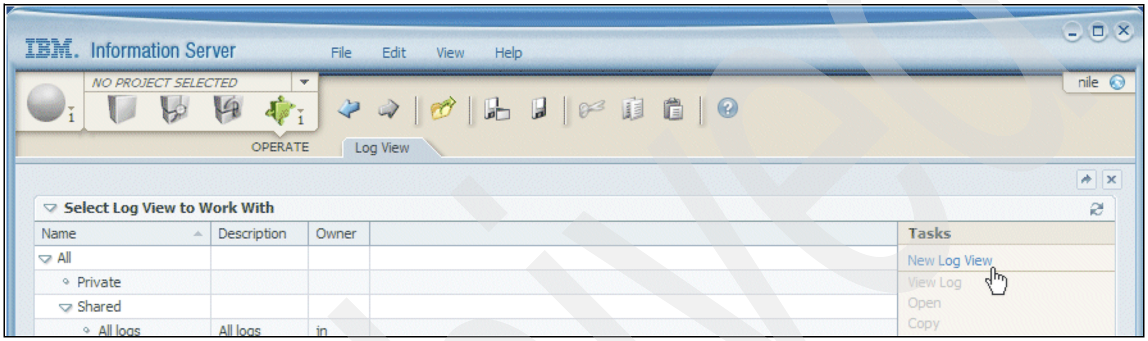
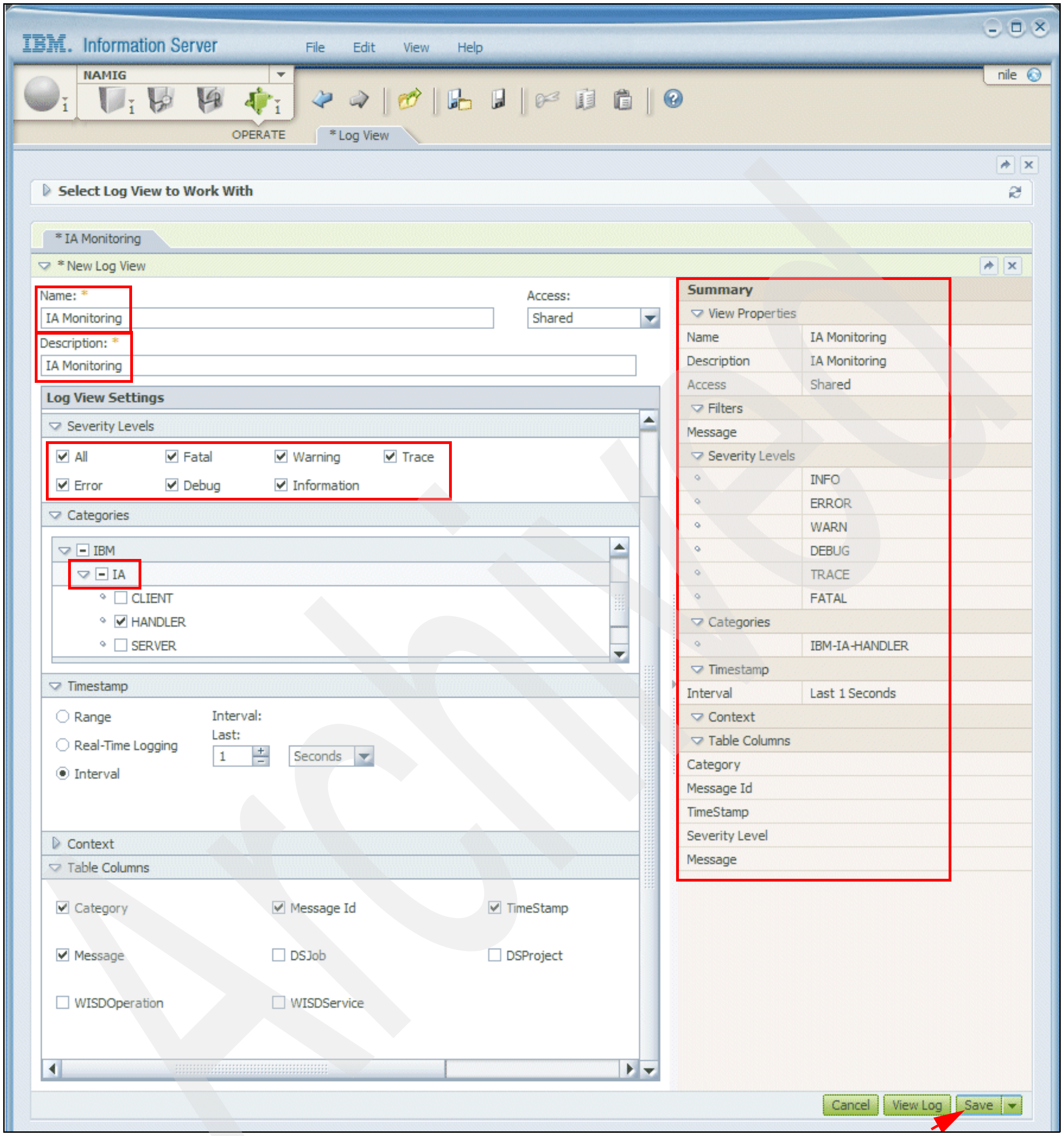


Figure 1-96 View job progress in Log Views 1/4



Name	Description	Owner
All		
Private		
Shared		
All logs	All logs	in

Figure 1-97 View job progress in Log Views 2/4



* New Log View

Name: * IA Monitoring

Description: * IA Monitoring

Access: Shared

Log View Settings

Severity Levels

☑ All ☑ Fatal ☑ Warning ☑ Trace

☑ Error ☑ Debug ☑ Information

Categories

- IBM
  - IA
    - □ CLIENT
    - ☑ HANDLER
    - □ SERVER

Timestamp

○ Range

○ Real-Time Logging

● Interval

Interval:
Last: 1 Seconds

Context

Table Columns

☑ Category ☑ Message Id ☑ TimeStamp

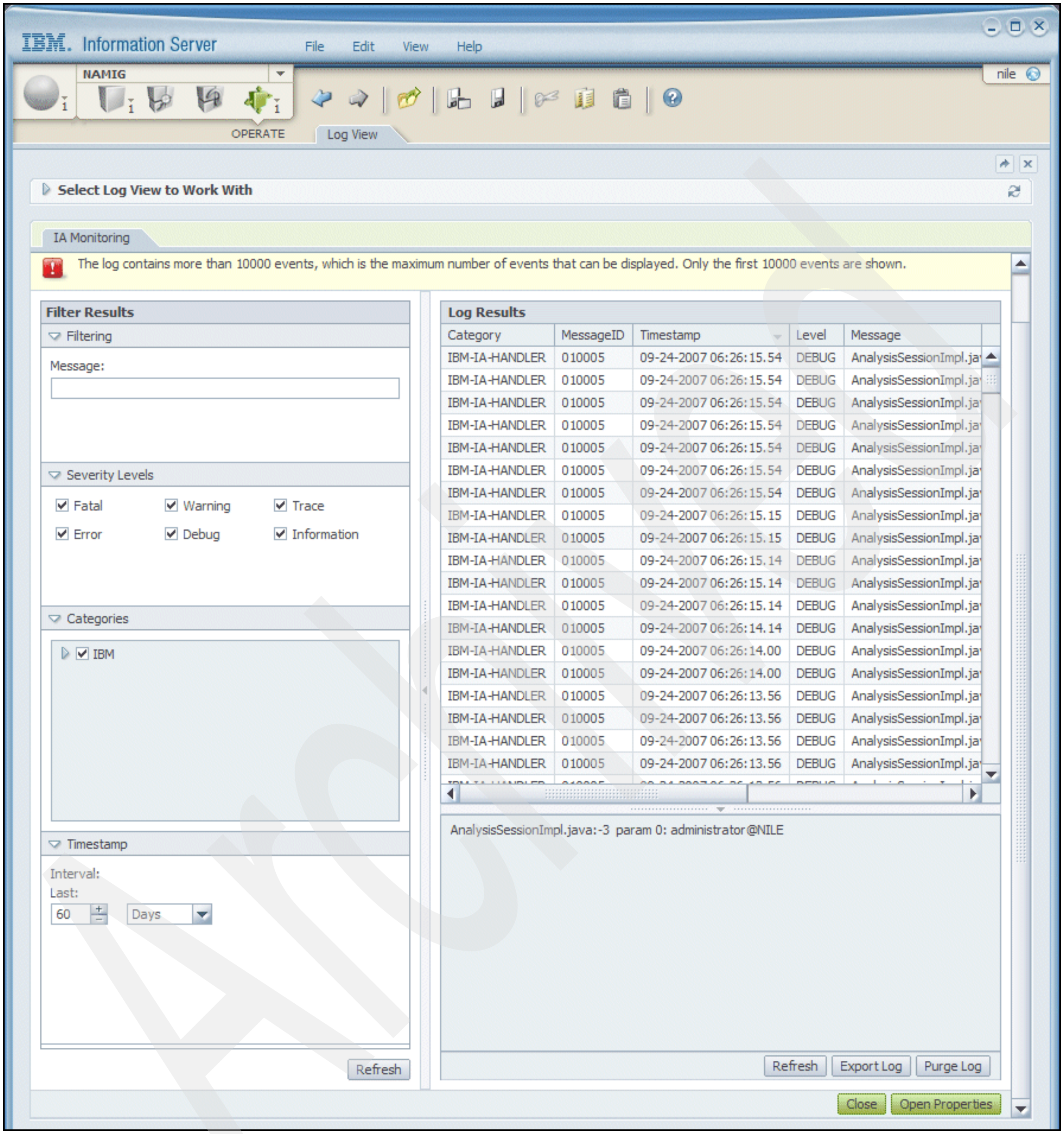
☑ Message □ DSJob □ DSProject

□ WISDOperation □ WISDService

Summary

View Properties	
Name	IA Monitoring
Description	IA Monitoring
Access	Shared
Filters	
Message	
Severity Levels	
	INFO
	ERROR
	WARN
	DEBUG
	TRACE
	FATAL
Categories	
	IBM-IA-HANDLER
Timestamp	
Interval	Last 1 Seconds
Context	
Table Columns	
Category	
Message Id	
TimeStamp	
Severity Level	
Message	

Figure 1-98 View job progress in Log Views 3/4



The log contains more than 10000 events, which is the maximum number of events that can be displayed. Only the first 10000 events are shown.

Filter Results

Filtering

Message:

Severity Levels

☑ Fatal ☑ Warning ☑ Trace

☑ Error ☑ Debug ☑ Information

Categories

☑ IBM

Timestamp

Interval:

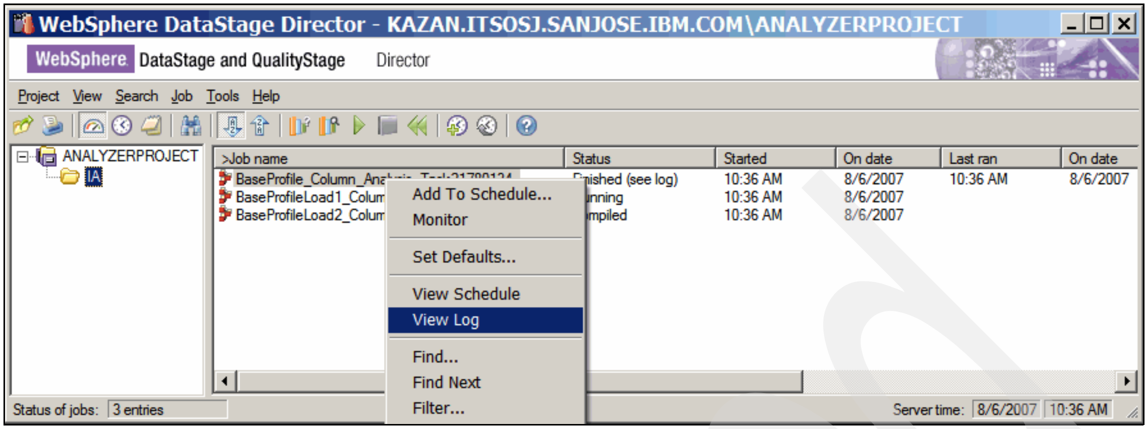
Last: 60 Days

Log Results

Category	MessageID	Timestamp	Level	Message
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.54	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.54	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.54	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.54	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.54	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.54	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.54	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.15	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.15	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.14	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.14	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:15.14	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:14.14	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:14.00	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:14.00	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:13.56	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:13.56	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:13.56	DEBUG	AnalysisSessionImpl.ja
IBM-IA-HANDLER	010005	09-24-2007 06:26:13.56	DEBUG	AnalysisSessionImpl.ja

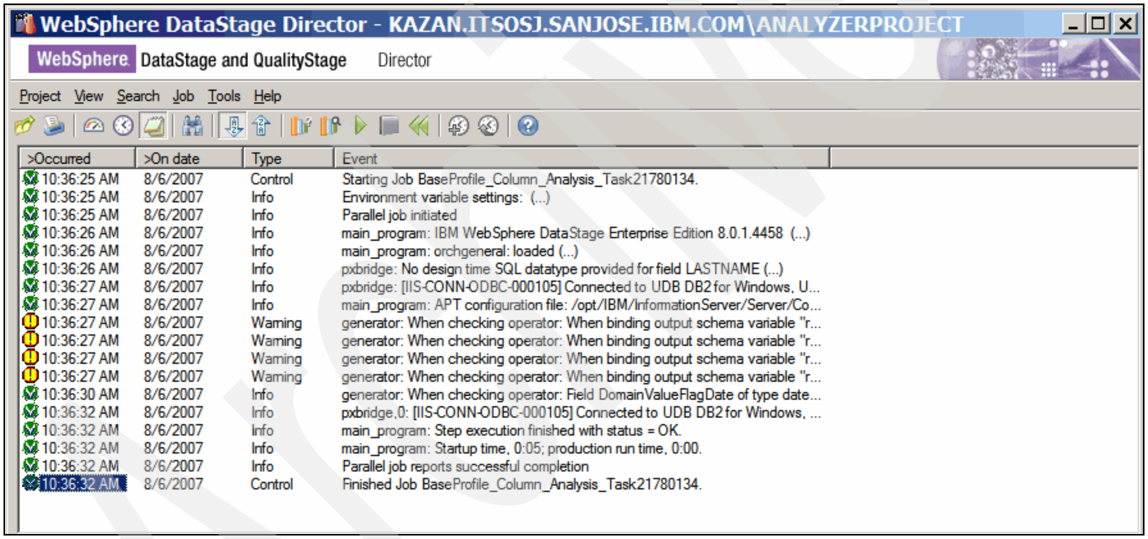
```
AnalysisSessionImpl.java:-3 param 0: administrator@NILE
```

Figure 1-99 View job progress in Log Views 4/4


WebSphere DataStage Director - KAZAN.ITSOSJ.SANJOSE.IBM.COM\ANALYZERPROJECT
WebSphere DataStage and QualityStage Director
Project View Search Job Tools Help
ANALYZERPROJECT
IA
Add To Schedule...
Monitor
Set Defaults...
View Schedule
View Log
Find...
Find Next
Filter...
Status of jobs: 3 entries
Server time: 8/6/2007 10:36 AM

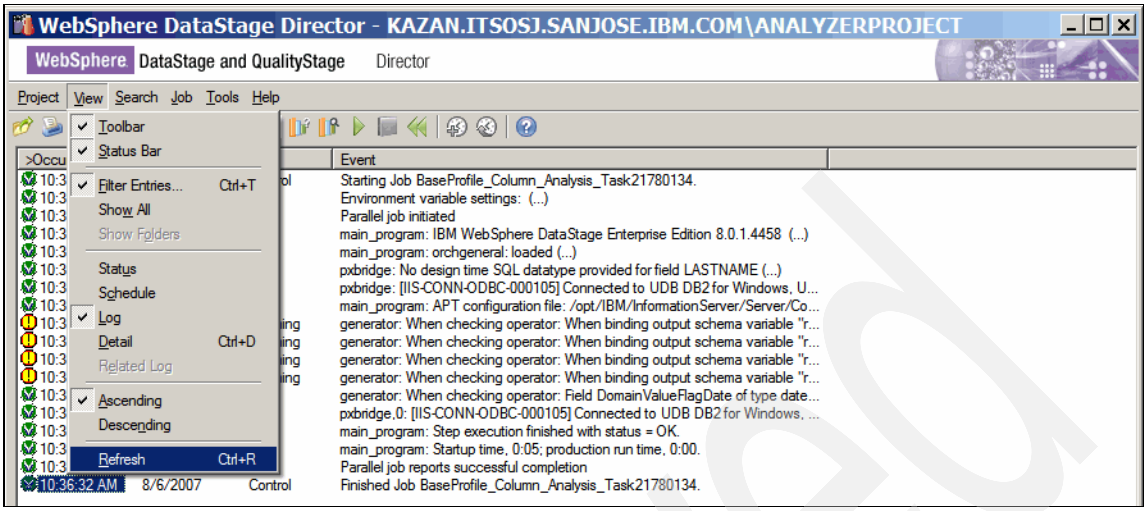
>Job name	Status	Started	On date	Last ran	On date
BaseProfile_Column_Ana	Finished (see log)	10:36 AM	8/6/2007	10:36 AM	8/6/2007
BaseProfileLoad1_Colum	nning	10:36 AM	8/6/2007		
BaseProfileLoad2_Colum	mpiled	10:36 AM	8/6/2007		

Figure 1-100 View job progress in DataStage and QualityStage Director 1/3


WebSphere DataStage Director - KAZAN.ITSOSJ.SANJOSE.IBM.COM\ANALYZERPROJECT
WebSphere DataStage and QualityStage Director
Project View Search Job Tools Help

>Occurred	>On date	Type	Event
10:36:25 AM	8/6/2007	Control	Starting Job BaseProfile_Column_Analysis_Task21780134.
10:36:25 AM	8/6/2007	Info	Environment variable settings: (...)
10:36:25 AM	8/6/2007	Info	Parallel job initiated
10:36:26 AM	8/6/2007	Info	main_program: IBM WebSphere DataStage Enterprise Edition 8.0.1.4458 (...)
10:36:26 AM	8/6/2007	Info	main_program: orchgeneral: loaded (...)
10:36:26 AM	8/6/2007	Info	pxbridge: No design time SQL datatype provided for field LASTNAME (...)
10:36:27 AM	8/6/2007	Info	pxbridge: [IIS-CONN-ODBC-000105] Connected to UDB DB2 for Windows, U...
10:36:27 AM	8/6/2007	Info	main_program: APT configuration file: /opt/IBM/InformationServer/Server/Co...
10:36:27 AM	8/6/2007	Warning	generator: When checking operator: When binding output schema variable "r...
10:36:27 AM	8/6/2007	Warning	generator: When checking operator: When binding output schema variable "r...
10:36:27 AM	8/6/2007	Warning	generator: When checking operator: When binding output schema variable "r...
10:36:27 AM	8/6/2007	Warning	generator: When checking operator: When binding output schema variable "r...
10:36:30 AM	8/6/2007	Info	generator: When checking operator: Field DomainValueFlagDate of type date...
10:36:32 AM	8/6/2007	Info	pxbridge,0: [IIS-CONN-ODBC-000105] Connected to UDB DB2 for Windows, ...
10:36:32 AM	8/6/2007	Info	main_program: Step execution finished with status = OK.
10:36:32 AM	8/6/2007	Info	main_program: Startup time, 0:05; production run time, 0:00.
10:36:32 AM	8/6/2007	Info	Parallel job reports successful completion
10:36:32 AM	8/6/2007	Control	Finished Job BaseProfile_Column_Analysis_Task21780134.

Figure 1-101 View job progress in DataStage and QualityStage Director 2/3



```
10:3      ol        Starting Job BaseProfile_Column_Analysis_Task21780134.
10:3                Environment variable settings:  (...)
10:3                Parallel job initiated
10:3                main_program: IBM WebSphere DataStage Enterprise Edition 8.0.1.4458  (...)
10:3                main_program: orchgeneral: loaded (...)
10:3                pxbridge: No design time SQL datatype provided for field LASTNAME (...)
10:3                pxbridge: [IIS-CONN-ODBC-000105] Connected to UDB DB2 for Windows, U...
10:3                main_program: APT configuration file: /opt/IBM/InformationServer/Server/Co...
10:3      ing       generator: When checking operator: When binding output schema variable "r...
10:3      ing       generator: When checking operator: When binding output schema variable "r...
10:3      ing       generator: When checking operator: When binding output schema variable "r...
10:3      ing       generator: When checking operator: When binding output schema variable "r...
10:3                generator: When checking operator: Field DomainValueFlagDate of type date...
10:3                pxbridge,0: [IIS-CONN-ODBC-000105] Connected to UDB DB2 for Windows, ...
10:3                main_program: Step execution finished with status = OK.
10:3                main_program: Startup time, 0:05; production run time, 0:00.
10:3                Parallel job reports successful completion
10:36:32 AM  8/6/2007  Control   Finished Job BaseProfile_Column_Analysis_Task21780134.
```

Figure 1-102   View job progress in DataStage and QualityStage Director 3/3

Review column analysis results

The View Analysis Summary is shown in Figure 1-103 on page 154. Scrolling to the right shows all the remaining columns (which we do not show here). This summary provides a bird's eye view of all the columns analyzed. Of specific interest are those columns that have a red flag next to it. In this case, all the columns have a red flag next to it, because we created test data in the tables that did not match the defined attributes. Follow these steps:

1. To view the column analysis details of a specific column, select it (EMPNO) and click **View Details** as shown in Figure 1-103 on page 154.

   The Mark Reviewed button allows you to mark the attribute (data class, properties, domain, and format) of a selected column as having been reviewed (not shown here).

2. As mentioned earlier, the detailed column level analysis includes an overview, frequency distribution, data class, properties, domain & completeness, and format information.

3. Figure 1-105 on page 156 through Figure 1-117 on page 168 show the details of column analysis for the EMPNO column as follows:

   – Figure 1-105 on page 156 summarizes the EMPNO column details under the Overview tab. Scrolling down (not shown here) provides information about the analysis status of the data class, properties, domain & completeness, and format.

- Frequency Distribution details are shown in Figure 1-105 on page 156 and Figure 1-106 on page 157. It shows the total number of rows (46), cardinality (number of distinct values in the column) as also being 46, (cardinality) percentage which is a calculation of the total number of distinct values in a column divided by the total number of values in the same column (100%), and the inferred data class as being an Identifier.

  Note the inferences of the data type for following values of EMPNO:

  - EMPNO 200240 is inferred to be a DATE data type with a format of DDMMYY
  - EMPNO 000110 is also inferred to be a DATE data type—this time with a format of YYMMDD
  - EMPNO 000160 is inferred to be an INT16 data type with a format of 999999
  - EMPNO 000060 on the other hand is inferred to be an INT8 data type with a format of 999999

- Data Class details are shown in Figure 1-107 on page 158. The inferred class is Identifier, which is correct. However, in case a wrong inference is made, you can correct it by selecting the appropriate data class from the drop-down list.

- Properties details are shown in Figure 1-108 on page 159 through Figure 1-111 on page 162.

  - In Figure 1-108 on page 159, the inferred data type is INT32 because integer is the predominant data type—(2.17% + 34.78% + 17.39%) for integer is greater than 45.65% for DATE. The Selected data type is the inferred data type, which you can choose to change if incorrect—in this case, it is correct. The pie-chart shows the data type distribution of the tabular information.
  - In Figure 1-109 on page 160, the defined, inferred, and selected values for length is 6, which is correct in this case. Additional information such as minimum, median, average, and maximum is provided along with a bar chart representation of the same information. You can choose to modify the Selected value if appropriate for consumption by other suite components or external programs.
  - In Figure 1-110 on page 161, the defined, inferred, and selected values for precision is 6, which is correct in this case. Additional information such as minimum, median, average, and maximum is provided along with a bar chart representation of the same information.
  - In Figure 1-111 on page 162, scale does not apply to this data type.

    The EMPNO column is defined as not nullable and inferred as such. You can choose to select it as nullable by checking Yes.

The cardinality type is defined as unique (because it is the primary key), inferred as unique as well based on the fact that all values are unique in this column. As before, you can change the selected value for cardinality type from the drop-down list.

– Domain & Completeness details are shown in Figure 1-112 on page 163 through Figure 1-116 on page 167.

  - In Figure 1-112 on page 163, when the Domain Type is Value (other choices in the drop-down list are Range and Reference Table, which is not shown here), the frequency distribution shows the individual values and the Status. By default, all values are initially identified as being valid. When you have changed some of these values to invalid or incomplete and rerun column analysis with the option of Updating Existing Tables (as described earlier), then the invalid/incomplete values are used to tag any new rows added with the correct invalid/incomplete status. It only tags new rows if the Range or Reference Table option is used. For the Value option new items are treated as Valid. However, it will maintain the identity of previously defined incomplete/invalid values and provide updated statistics based on the new run of the column analysis.

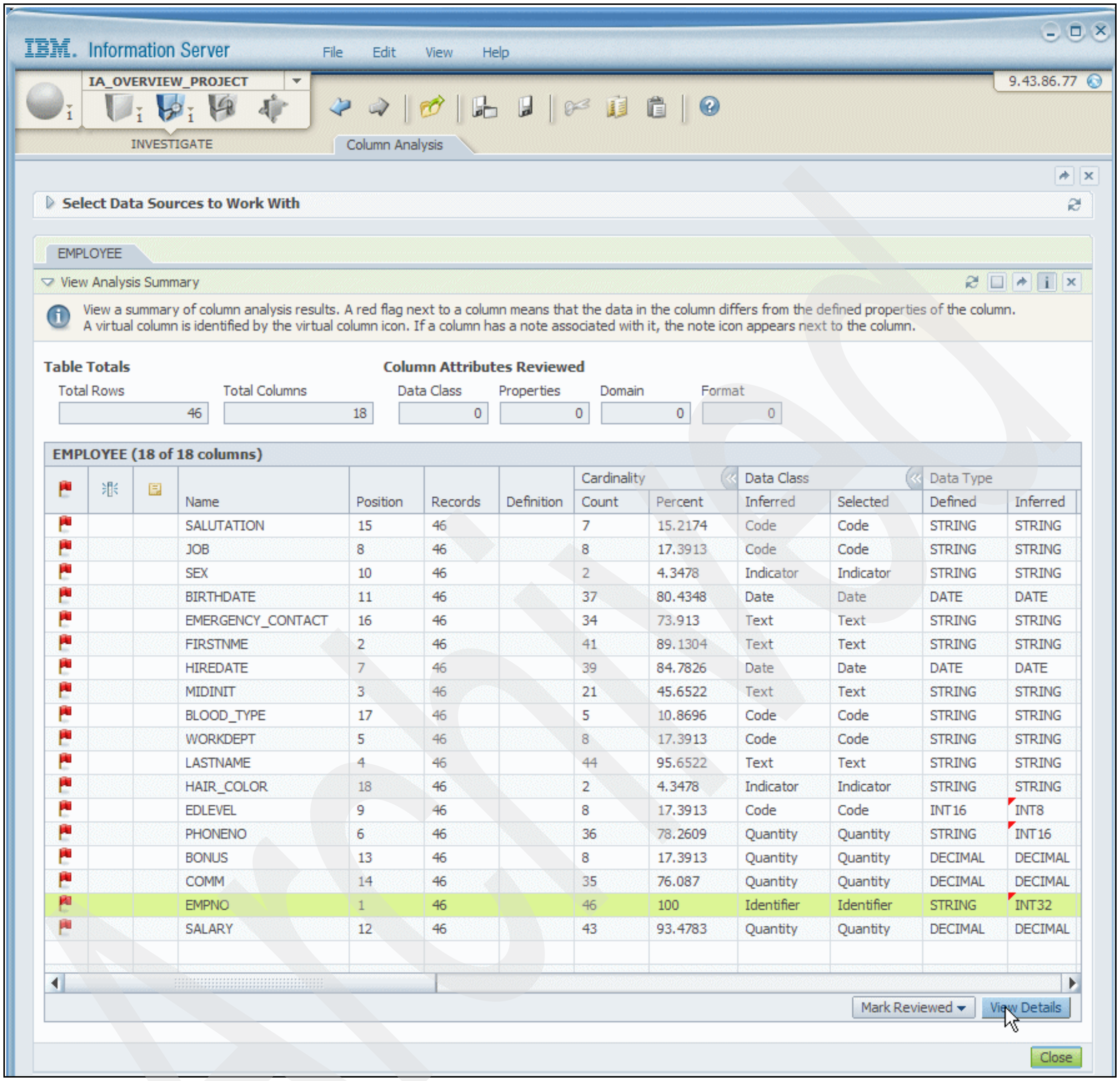
    Scrolling right, Figure 1-113 on page 164 shows the Completeness Summary, Validity Summary, Incomplete Values, and Invalid Values details. Because all the values in EMPNO are complete and valid, these fields reflect that.

- Scrolling left again, in Figure 1-114 on page 165 click **Show Quintiles** to view the frequency distribution as column charts.[17] Figure 1-115 on page 166 shows the quintile information. Additional information provided for each quintile includes the low and high values, number of distinct values, and number of records. This describes the statistical distribution of the values in the column.
- As mentioned earlier, you can mark values in the frequency distribution as being Invalid as shown in Figure 1-116 on page 167. When you make changes, you must click **Save** to confirm these changes. We did not modify the status of any of the values.

– Format details are shown in Figure 1-117 on page 168 through Figure 1-116 on page 167. The different formats (999999, DDMMYY, and YYMMDD) are shown here, as are the distinct values. The status of each format can be marked as being valid (Conform) or invalid (Violate). If they are marked as such and saved (by clicking **Save**), then these format will be displayed in the Format Violations section.

Note: As mentioned earlier, anytime you modify the status of values and save the changes, you can click **Rebuild Inferences** (using only Valid values or All values) to have IBM WebSphere Information Analyzer recompute and display new inferences.

[17] A *quintile* is one fifth or 20% of a given amount. In this case, the quintile is an equal division of the number of Distinct Values into five parts. The difference in the quintiles reflects the distribution of the number of records within those Distinct Values. This is a convenient way to identify skews in the data across the quintile divisions.



View a summary of column analysis results. A red flag next to a column means that the data in the column differs from the defined properties of the column. A virtual column is identified by the virtual column icon. If a column has a note associated with it, the note icon appears next to the column.

Table Totals

Total Rows: 46

Total Columns: 18

Column Attributes Reviewed

Data Class: 0

Properties: 0

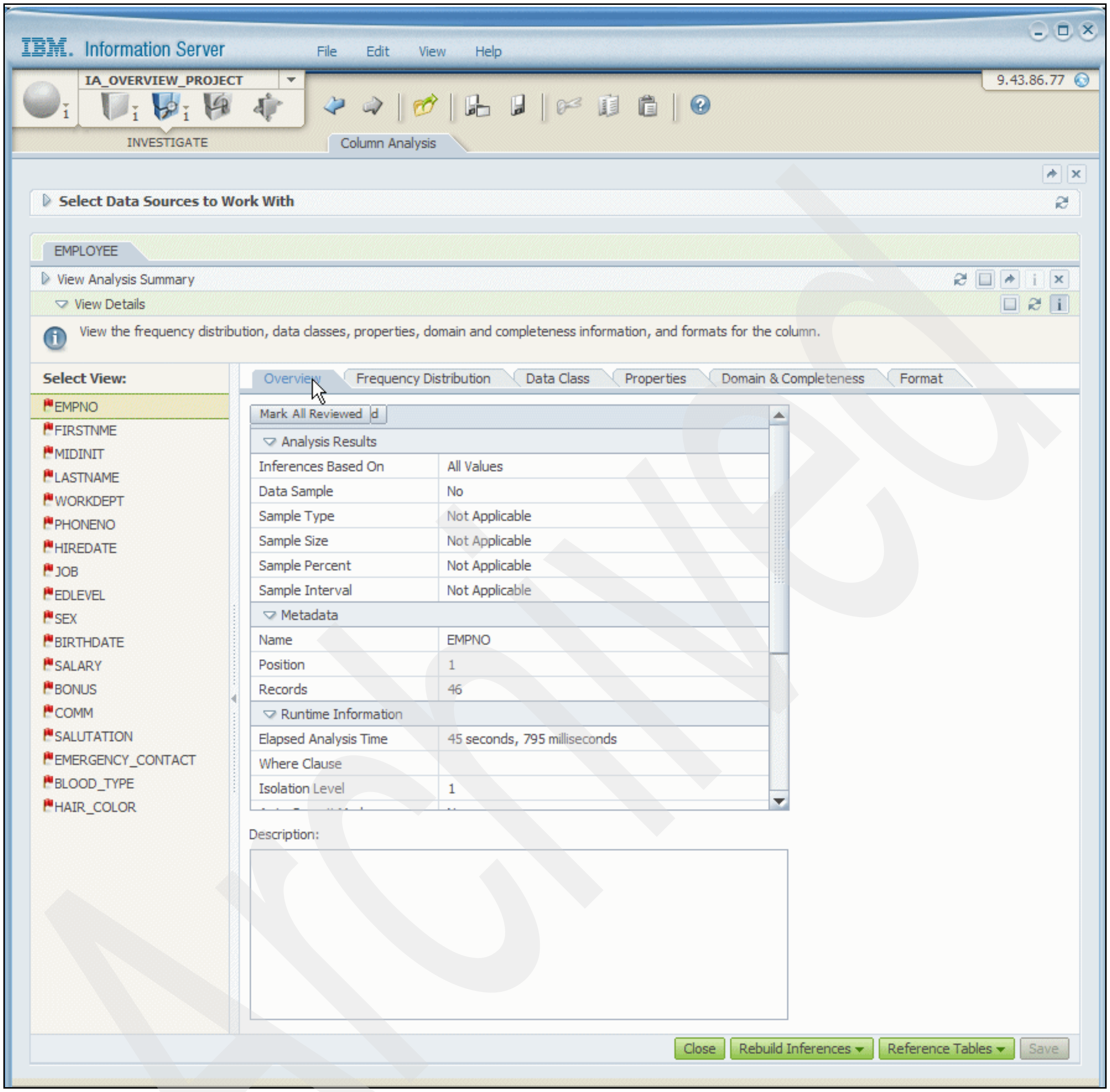
Domain: 0

Format: 0

EMPLOYEE (18 of 18 columns)

							Cardinality		Data Class		Data Type	
🚩			Name	Position	Records	Definition	Count	Percent	Inferred	Selected	Defined	Inferred
🚩			SALUTATION	15	46		7	15.2174	Code	Code	STRING	STRING
🚩			JOB	8	46		8	17.3913	Code	Code	STRING	STRING
🚩			SEX	10	46		2	4.3478	Indicator	Indicator	STRING	STRING
🚩			BIRTHDATE	11	46		37	80.4348	Date	Date	DATE	DATE
🚩			EMERGENCY_CONTACT	16	46		34	73.913	Text	Text	STRING	STRING
🚩			FIRSTNME	2	46		41	89.1304	Text	Text	STRING	STRING
🚩			HIREDATE	7	46		39	84.7826	Date	Date	DATE	DATE
🚩			MIDINIT	3	46		21	45.6522	Text	Text	STRING	STRING
🚩			BLOOD_TYPE	17	46		5	10.8696	Code	Code	STRING	STRING
🚩			WORKDEPT	5	46		8	17.3913	Code	Code	STRING	STRING
🚩			LASTNAME	4	46		44	95.6522	Text	Text	STRING	STRING
🚩			HAIR_COLOR	18	46		2	4.3478	Indicator	Indicator	STRING	STRING
🚩			EDLEVEL	9	46		8	17.3913	Code	Code	INT16	INT8
🚩			PHONENO	6	46		36	78.2609	Quantity	Quantity	STRING	INT16
🚩			BONUS	13	46		8	17.3913	Quantity	Quantity	DECIMAL	DECIMAL
🚩			COMM	14	46		35	76.087	Quantity	Quantity	DECIMAL	DECIMAL
🚩			EMPNO	1	46		46	100	Identifier	Identifier	STRING	INT32
🚩			SALARY	12	46		43	93.4783	Quantity	Quantity	DECIMAL	DECIMAL

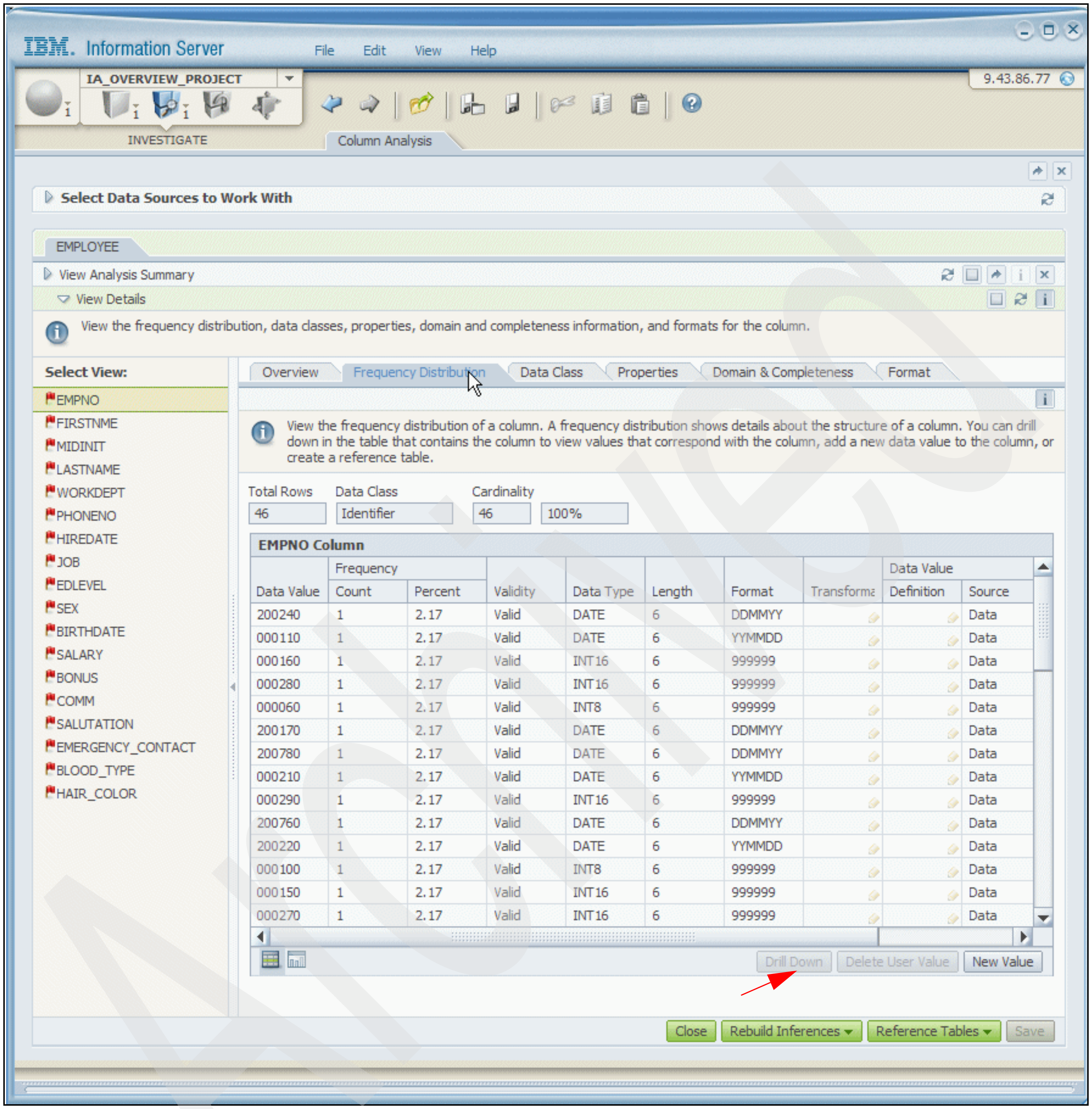
Figure 1-103 View EMPNO column details in the EMPLOYEE table 1/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Analysis Results	
Inferences Based On	All Values
Data Sample	No
Sample Type	Not Applicable
Sample Size	Not Applicable
Sample Percent	Not Applicable
Sample Interval	Not Applicable
Metadata	
Name	EMPNO
Position	1
Records	46
Runtime Information	
Elapsed Analysis Time	45 seconds, 795 milliseconds
Where Clause	
Isolation Level	1

Figure 1-104 View EMPNO column details in the EMPLOYEE table 2/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the frequency distribution of a column. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view values that correspond with the column, add a new data value to the column, or create a reference table.

Total Rows: 46

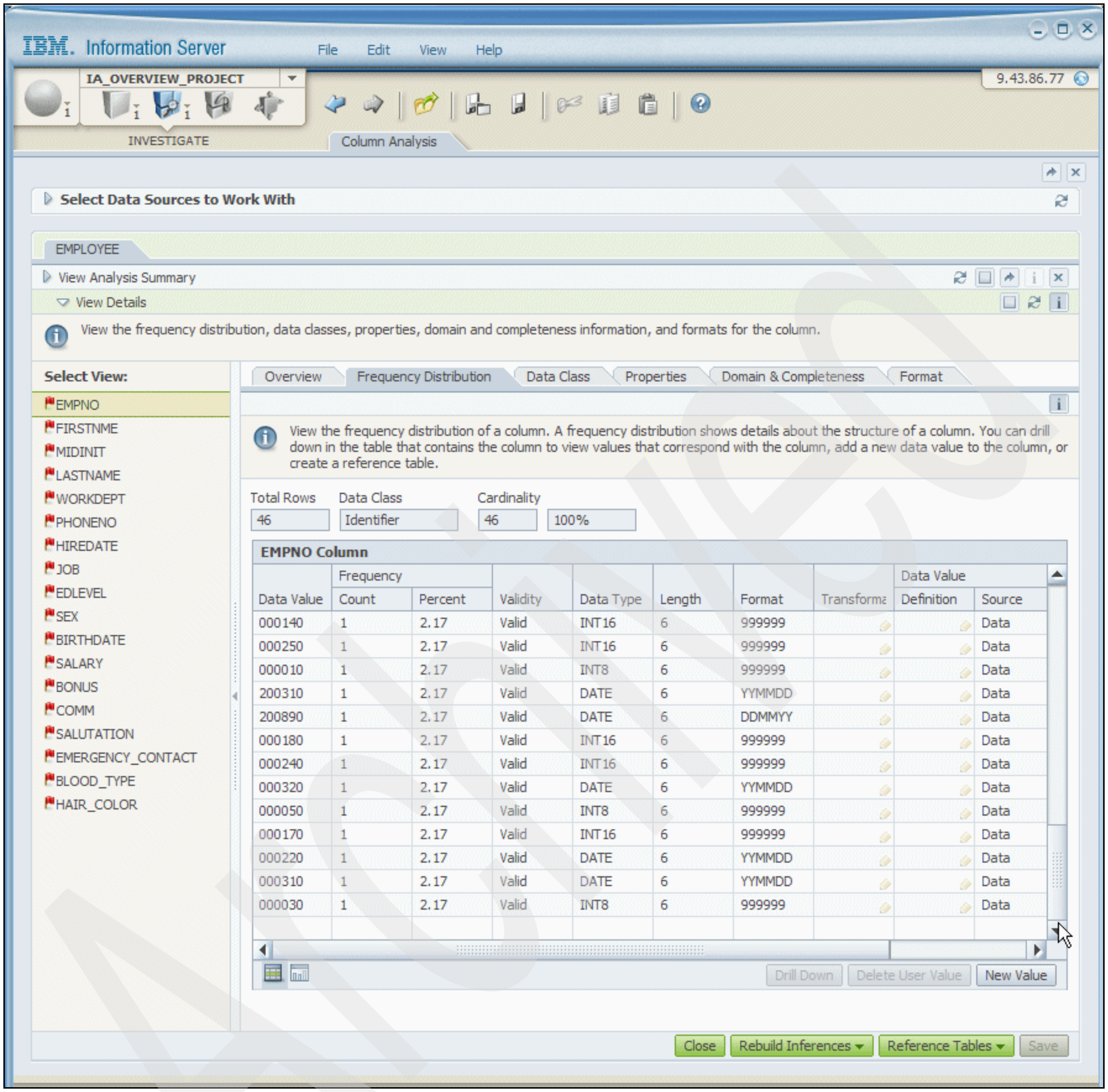
Data Class: Identifier

Cardinality: 46 100%

EMPNO Column

	Frequency							Data Value	
Data Value	Count	Percent	Validity	Data Type	Length	Format	Transforma	Definition	Source
200240	1	2.17	Valid	DATE	6	DDMMYY			Data
000110	1	2.17	Valid	DATE	6	YYMMDD			Data
000160	1	2.17	Valid	INT16	6	999999			Data
000280	1	2.17	Valid	INT16	6	999999			Data
000060	1	2.17	Valid	INT8	6	999999			Data
200170	1	2.17	Valid	DATE	6	DDMMYY			Data
200780	1	2.17	Valid	DATE	6	DDMMYY			Data
000210	1	2.17	Valid	DATE	6	YYMMDD			Data
000290	1	2.17	Valid	INT16	6	999999			Data
200760	1	2.17	Valid	DATE	6	DDMMYY			Data
200220	1	2.17	Valid	DATE	6	YYMMDD			Data
000100	1	2.17	Valid	INT8	6	999999			Data
000150	1	2.17	Valid	INT16	6	999999			Data
000270	1	2.17	Valid	INT16	6	999999			Data

Figure 1-105 View EMPNO column details in the EMPLOYEE table 3/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

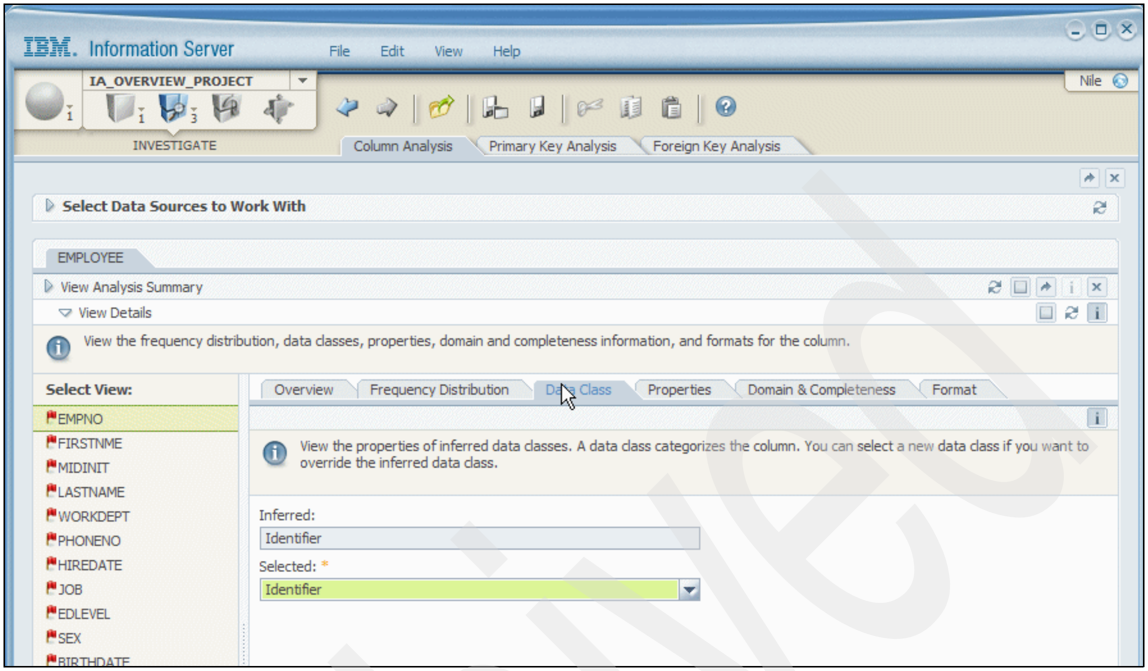
View the frequency distribution of a column. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view values that correspond with the column, add a new data value to the column, or create a reference table.

Total Rows: 46
Data Class: Identifier
Cardinality: 46 100%

EMPNO Column

	Frequency							Data Value	
Data Value	Count	Percent	Validity	Data Type	Length	Format	Transforma	Definition	Source
000140	1	2.17	Valid	INT16	6	999999			Data
000250	1	2.17	Valid	INT16	6	999999			Data
000010	1	2.17	Valid	INT8	6	999999			Data
200310	1	2.17	Valid	DATE	6	YYMMDD			Data
200890	1	2.17	Valid	DATE	6	DDMMYY			Data
000180	1	2.17	Valid	INT16	6	999999			Data
000240	1	2.17	Valid	INT16	6	999999			Data
000320	1	2.17	Valid	DATE	6	YYMMDD			Data
000050	1	2.17	Valid	INT8	6	999999			Data
000170	1	2.17	Valid	INT16	6	999999			Data
000220	1	2.17	Valid	DATE	6	YYMMDD			Data
000310	1	2.17	Valid	DATE	6	YYMMDD			Data
000030	1	2.17	Valid	INT8	6	999999			Data

Figure 1-106 View EMPNO column details in the EMPLOYEE table 4/15



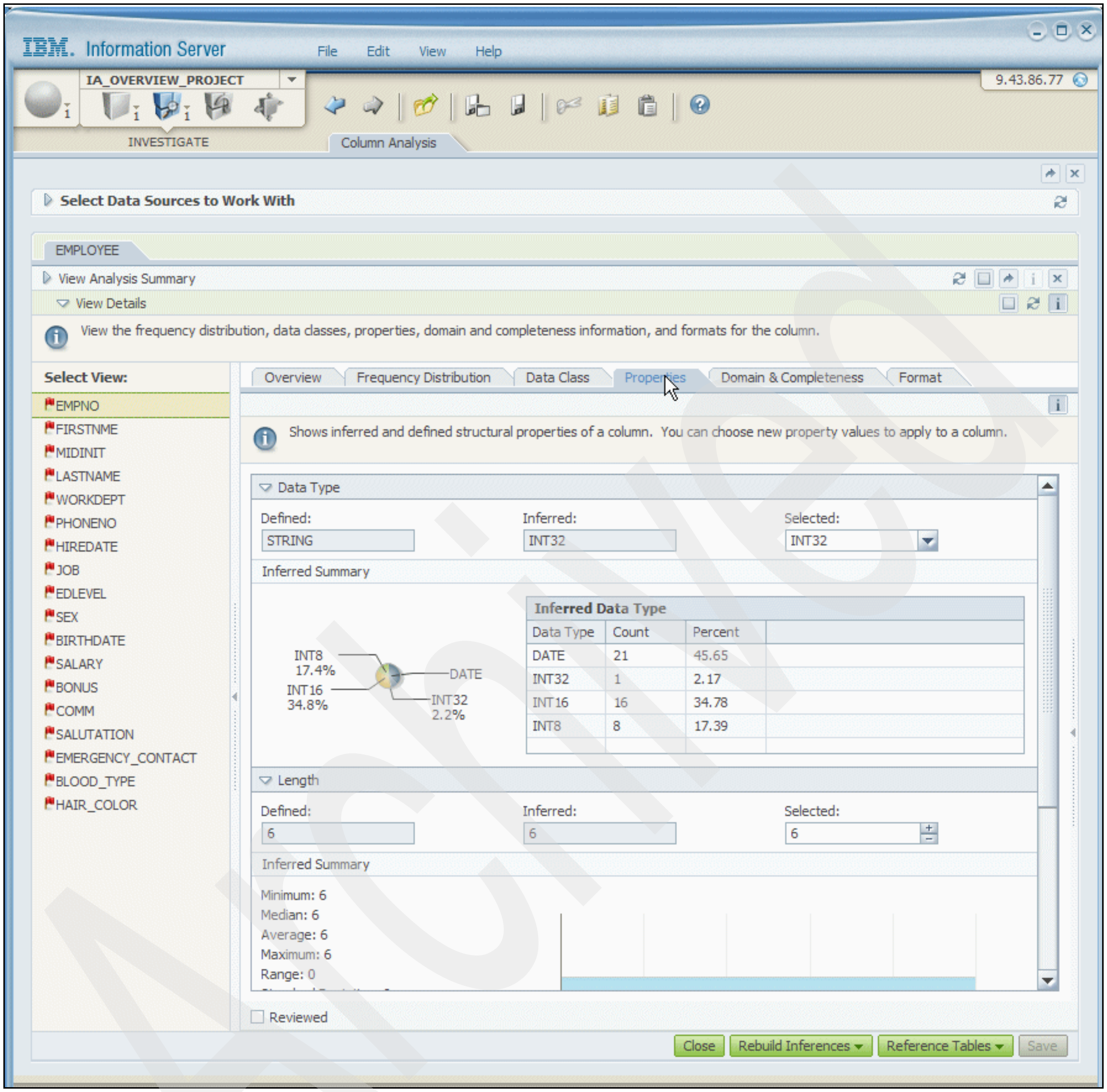
View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the properties of inferred data classes. A data class categorizes the column. You can select a new data class if you want to override the inferred data class.

Inferred: Identifier

Selected: * Identifier

Figure 1-107 View EMPNO column details in the EMPLOYEE table 5/15



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Data Type

Defined: STRING Inferred: INT32 Selected: INT32

Inferred Summary

Inferred Data Type

Data Type	Count	Percent
DATE	21	45.65
INT32	1	2.17
INT16	16	34.78
INT8	8	17.39

Length

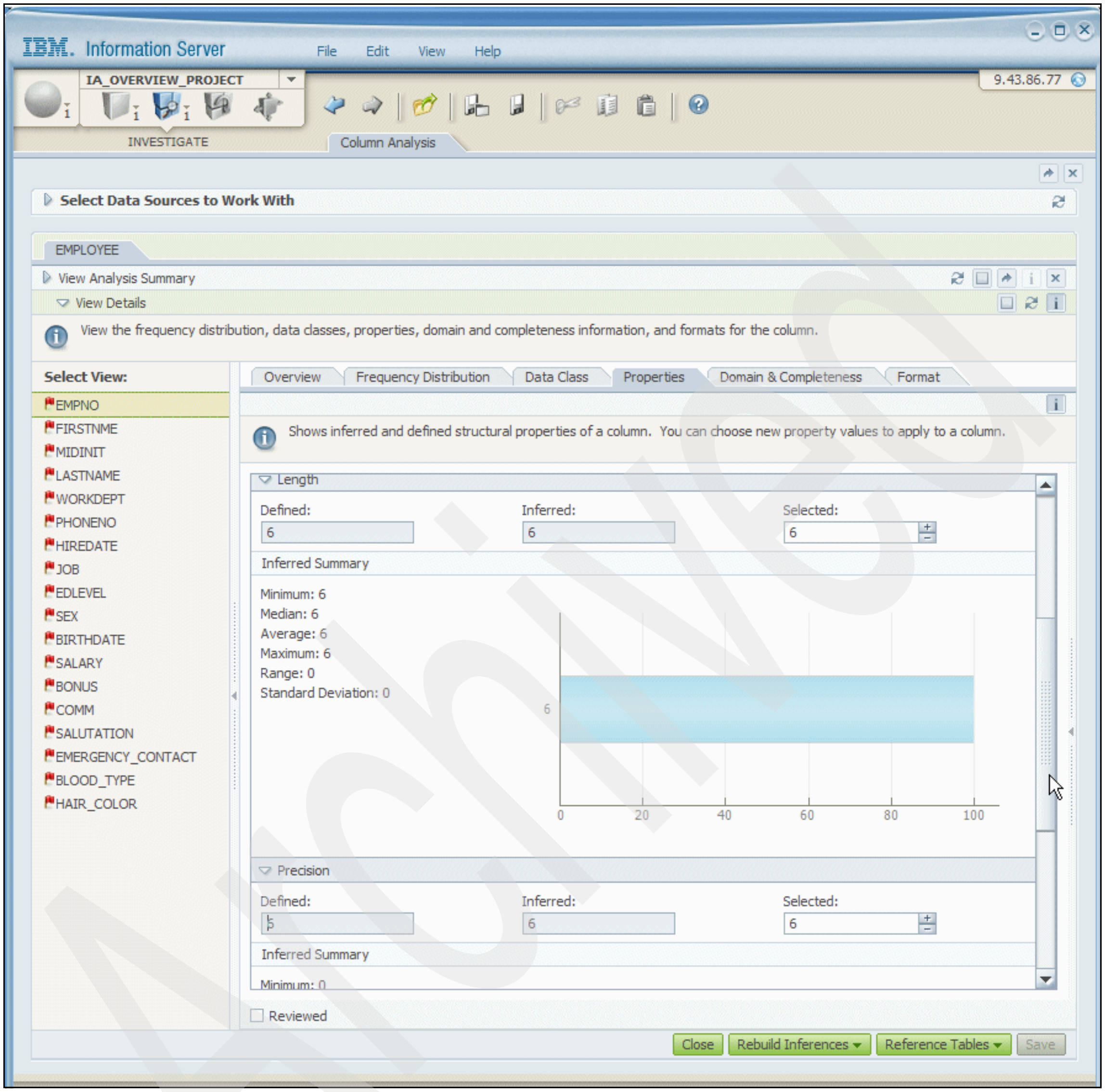
Defined: 6 Inferred: 6 Selected: 6

Inferred Summary

Minimum: 6
Median: 6
Average: 6
Maximum: 6
Range: 0

□ Reviewed

Figure 1-108 View EMPNO column details in the EMPLOYEE table 6/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Length

Defined: 6
Inferred: 6
Selected: 6

Inferred Summary

Minimum: 6
Median: 6
Average: 6
Maximum: 6
Range: 0
Standard Deviation: 0

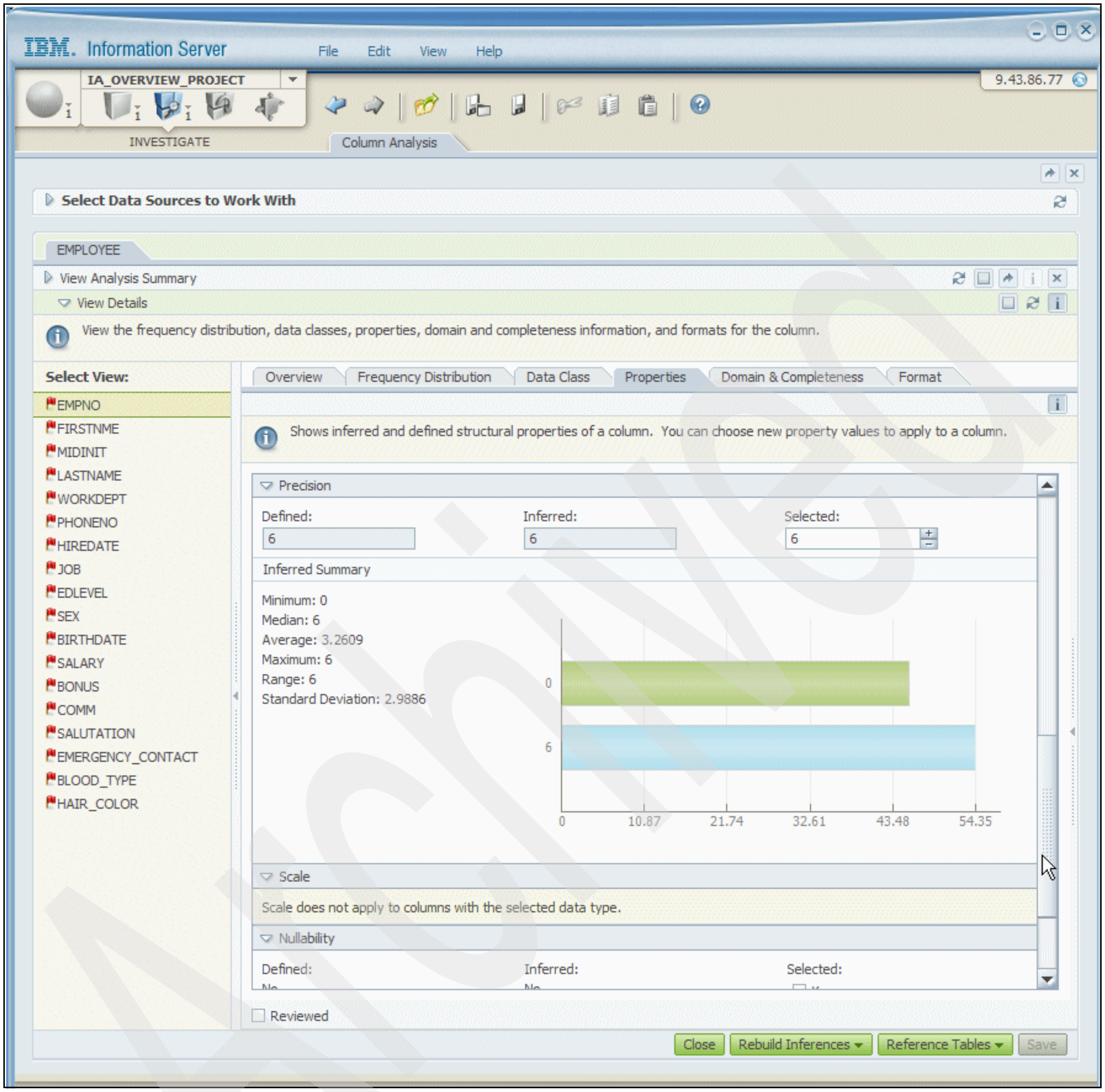
Precision

Defined: 5
Inferred: 6
Selected: 6

Inferred Summary

Minimum: 0

Figure 1-109 View EMPNO column details in the EMPLOYEE table 7/15



Select Data Sources to Work With

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Precision

Defined: 6
Inferred: 6
Selected: 6

Inferred Summary

Minimum: 0
Median: 6
Average: 3.2609
Maximum: 6
Range: 6
Standard Deviation: 2.9886

Scale

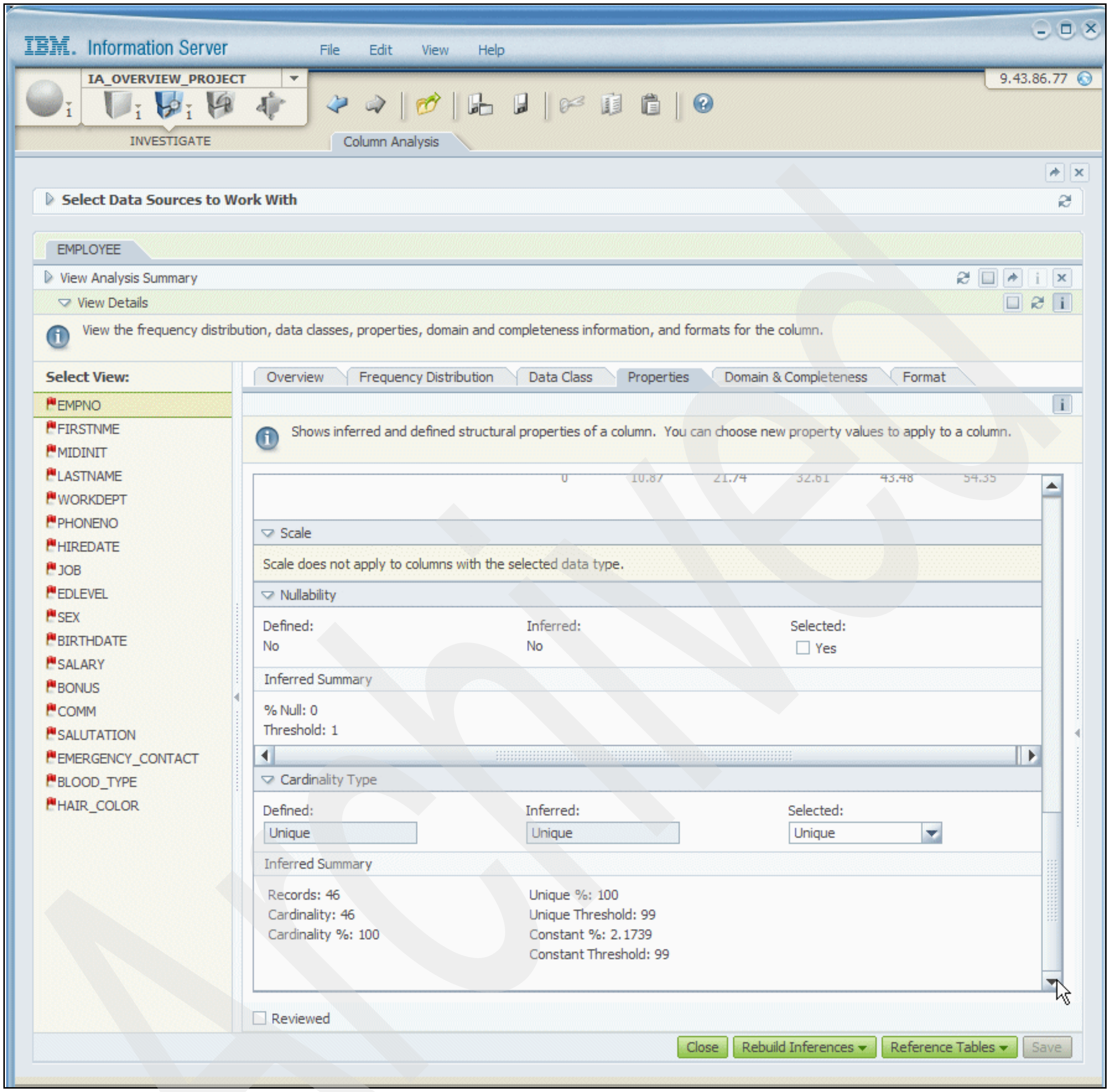
Scale does not apply to columns with the selected data type.

Nullability

Defined:
Inferred:
Selected:

□ Reviewed

Figure 1-110 View EMPNO column details in the EMPLOYEE table 8/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Scale does not apply to columns with the selected data type.

Defined: No

Inferred: No

Selected: □ Yes

Inferred Summary

% Null: 0
Threshold: 1

Defined: Unique

Inferred: Unique

Selected: Unique

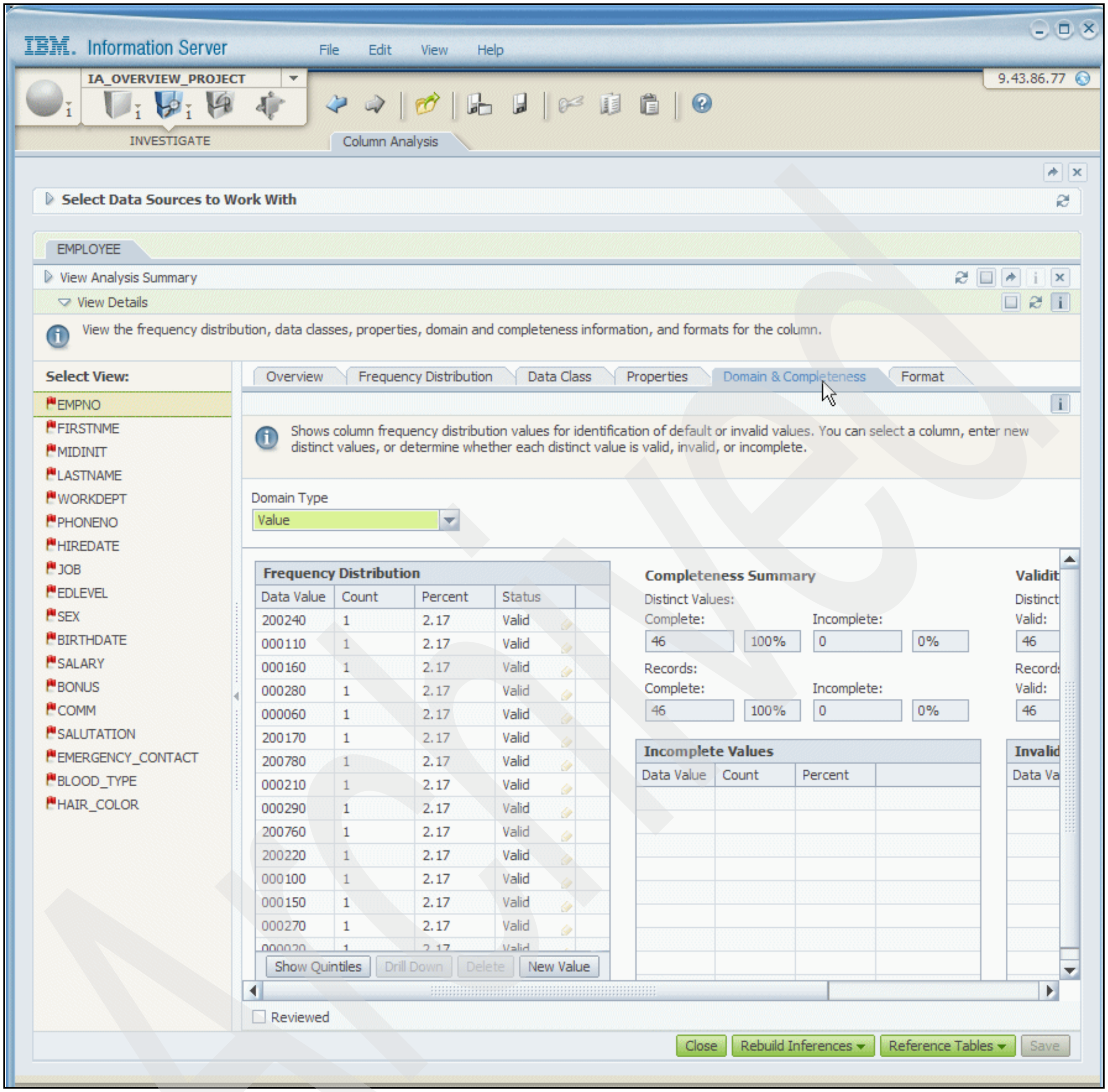
Inferred Summary

Records: 46
Cardinality: 46
Cardinality %: 100

Unique %: 100
Unique Threshold: 99
Constant %: 2.1739
Constant Threshold: 99

□ Reviewed

Figure 1-111 View EMPNO column details in the EMPLOYEE table 9/15



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
200240	1	2.17	Valid
000110	1	2.17	Valid
000160	1	2.17	Valid
000280	1	2.17	Valid
000060	1	2.17	Valid
200170	1	2.17	Valid
200780	1	2.17	Valid
000210	1	2.17	Valid
000290	1	2.17	Valid
200760	1	2.17	Valid
200220	1	2.17	Valid
000100	1	2.17	Valid
000150	1	2.17	Valid
000270	1	2.17	Valid
000020	1	2.17	Valid

Completeness Summary

Distinct Values:
Complete: 46 100%
Incomplete: 0 0%

Records:
Complete: 46 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 46

Record:
Valid: 46

Incomplete Values

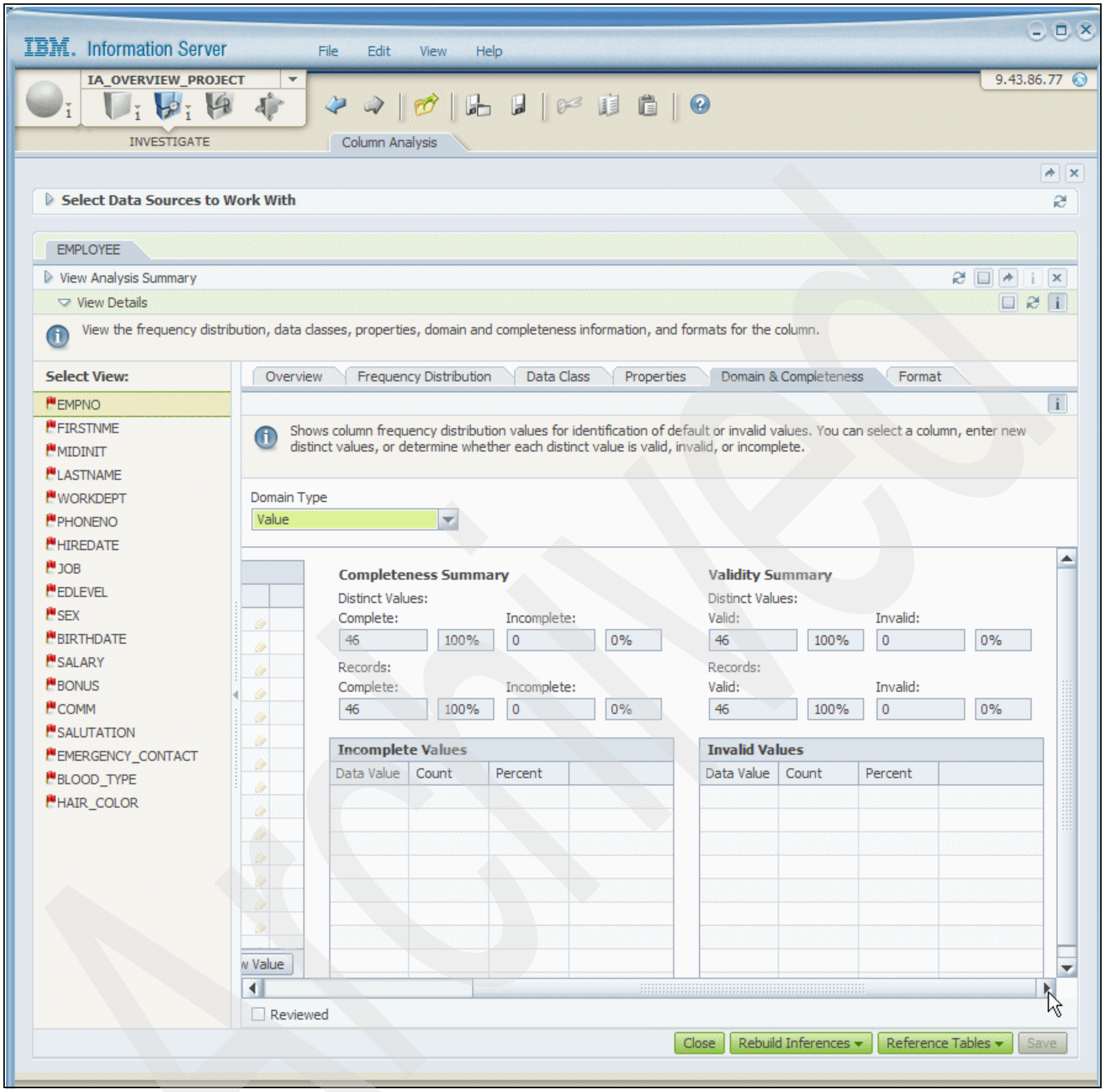
Data Value	Count	Percent

Invalid

Data Va

☐ Reviewed

Figure 1-112 View EMPNO column details in the EMPLOYEE table 10/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Completeness Summary

Distinct Values:
Complete: 46 100% Incomplete: 0 0%

Records:
Complete: 46 100% Incomplete: 0 0%

Validity Summary

Distinct Values:
Valid: 46 100% Invalid: 0 0%

Records:
Valid: 46 100% Invalid: 0 0%

Incomplete Values

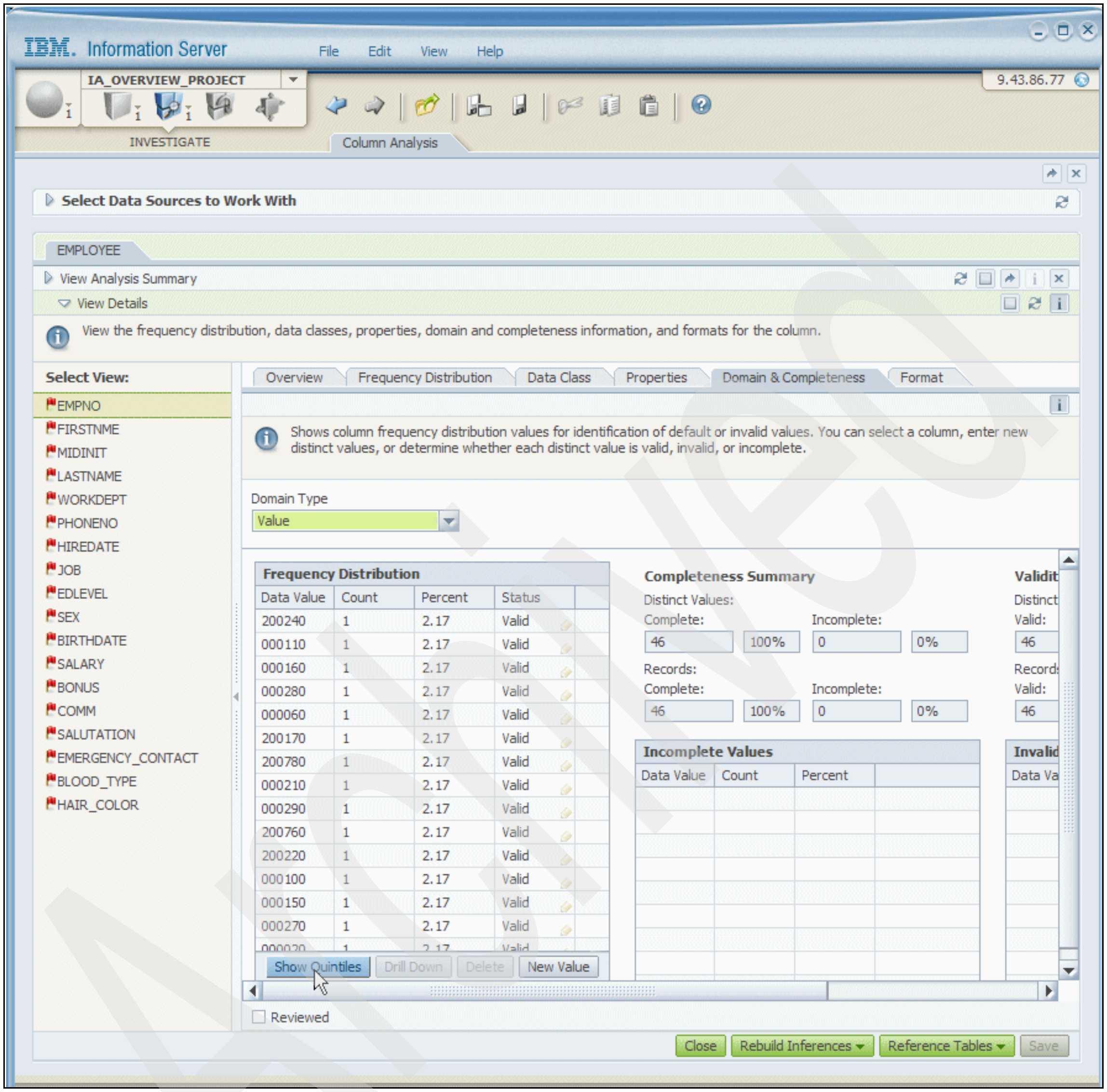
Data Value	Count	Percent

Invalid Values

Data Value	Count	Percent

□ Reviewed

Figure 1-113 View EMPNO column details in the EMPLOYEE table 11/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
200240	1	2.17	Valid
000110	1	2.17	Valid
000160	1	2.17	Valid
000280	1	2.17	Valid
000060	1	2.17	Valid
200170	1	2.17	Valid
200780	1	2.17	Valid
000210	1	2.17	Valid
000290	1	2.17	Valid
200760	1	2.17	Valid
200220	1	2.17	Valid
000100	1	2.17	Valid
000150	1	2.17	Valid
000270	1	2.17	Valid
000020	1	2.17	Valid

Completeness Summary

Distinct Values:
Complete: 46 100% Incomplete: 0 0%

Records:
Complete: 46 100% Incomplete: 0 0%

Validit

Distinct
Valid: 46

Record:
Valid: 46

Incomplete Values

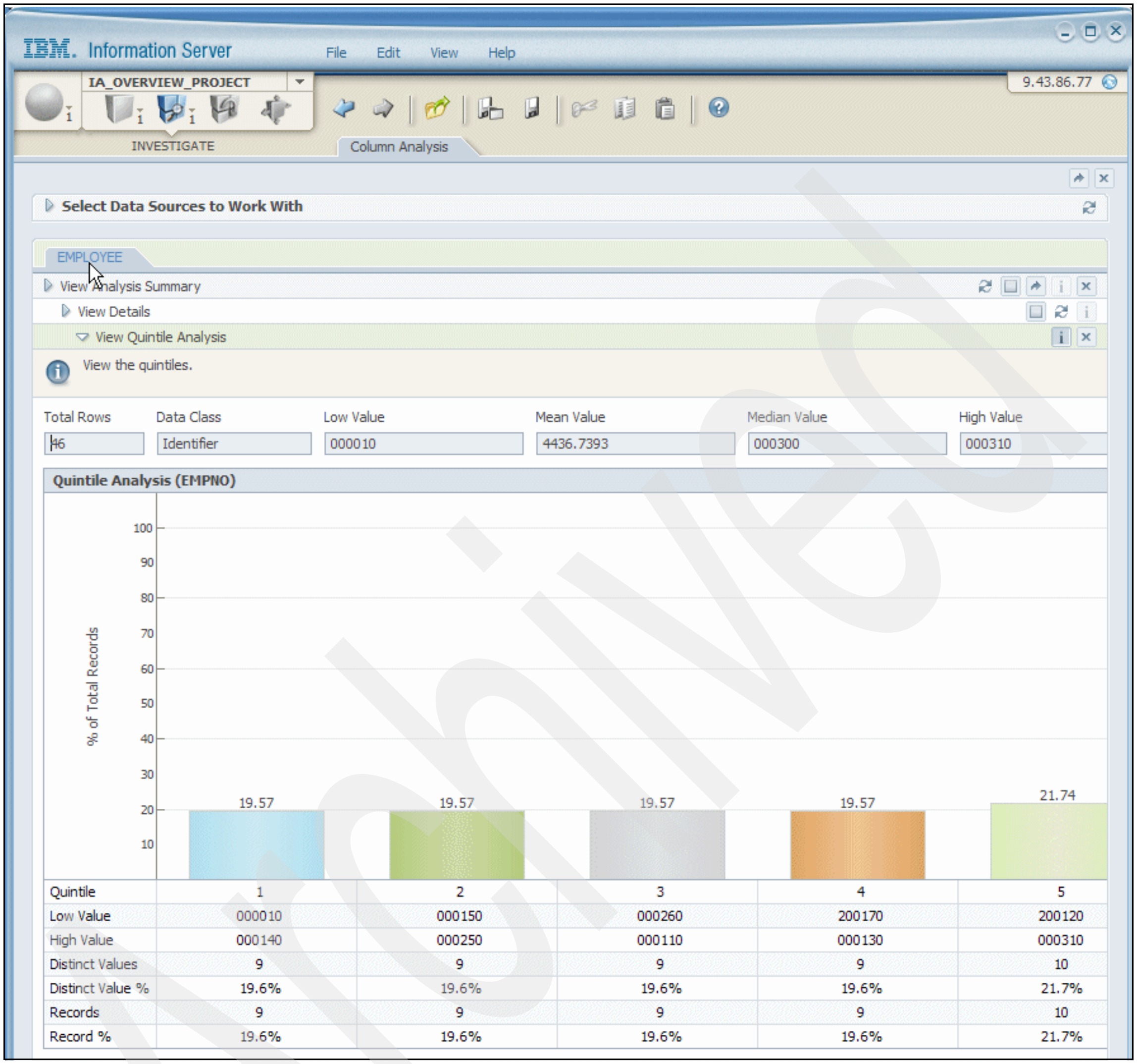
Data Value	Count	Percent

Invalid

Data Va

□ Reviewed

Figure 1-114 View EMPNO column details in the EMPLOYEE table 12/15



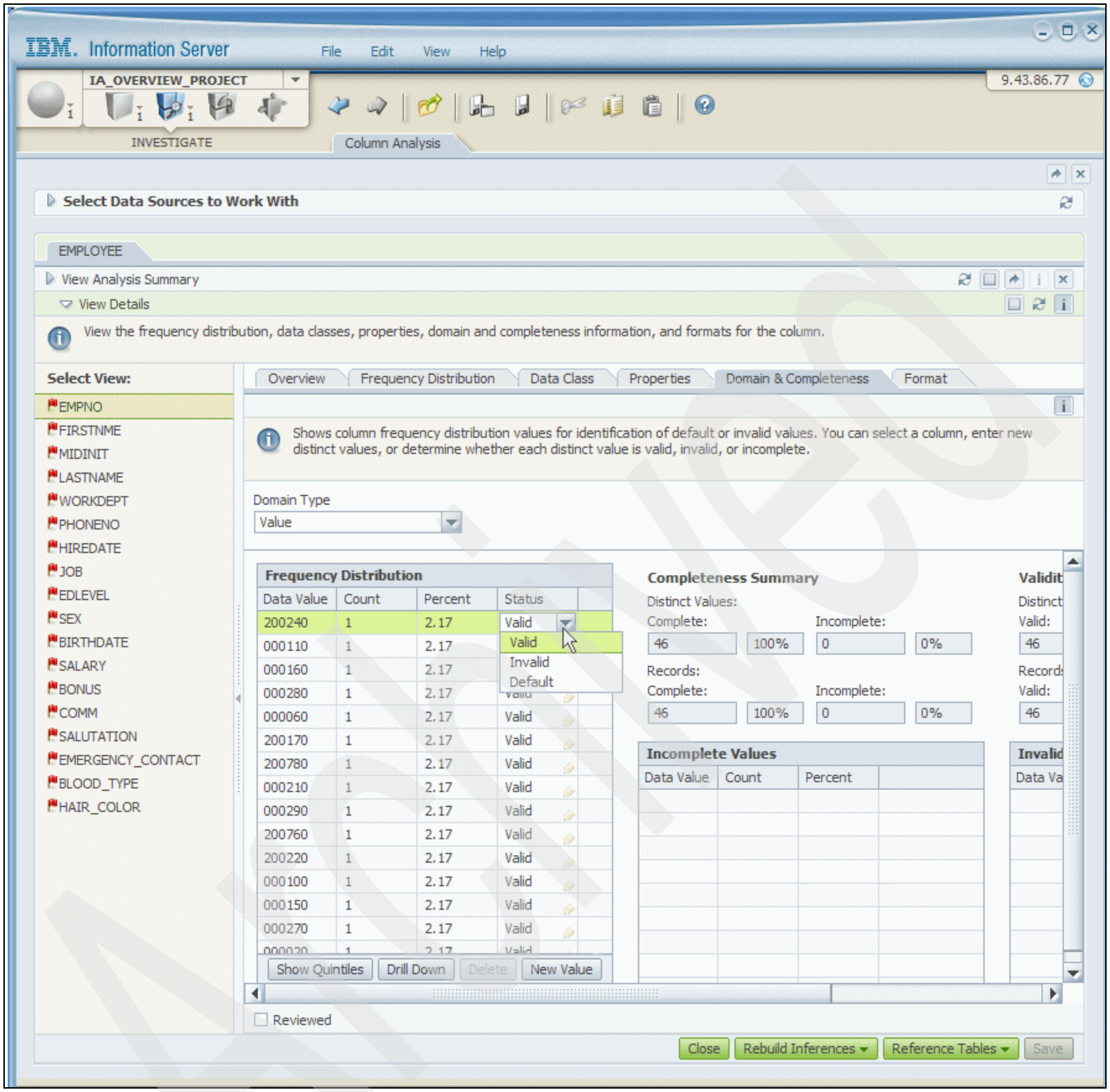
View the quintiles.

Total Rows	Data Class	Low Value	Mean Value	Median Value	High Value
46	Identifier	000010	4436.7393	000300	000310

Quintile Analysis (EMPNO)

Quintile	1	2	3	4	5
Low Value	000010	000150	000260	200170	200120
High Value	000140	000250	000110	000130	000310
Distinct Values	9	9	9	9	10
Distinct Value %	19.6%	19.6%	19.6%	19.6%	21.7%
Records	9	9	9	9	10
Record %	19.6%	19.6%	19.6%	19.6%	21.7%

Figure 1-115 View EMPNO column details in the EMPLOYEE table 13/15



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
200240	1	2.17	Valid
000110	1	2.17	Valid
000160	1	2.17	Invalid
000280	1	2.17	Default
000060	1	2.17	Valid
200170	1	2.17	Valid
200780	1	2.17	Valid
000210	1	2.17	Valid
000290	1	2.17	Valid
200760	1	2.17	Valid
200220	1	2.17	Valid
000100	1	2.17	Valid
000150	1	2.17	Valid
000270	1	2.17	Valid
000020	1	2.17	Valid

Completeness Summary

Distinct Values:
Complete: 46 100% Incomplete: 0 0%

Records:
Complete: 46 100% Incomplete: 0 0%

Validit
Distinct
Valid: 46
Record:
Valid: 46

Incomplete Values

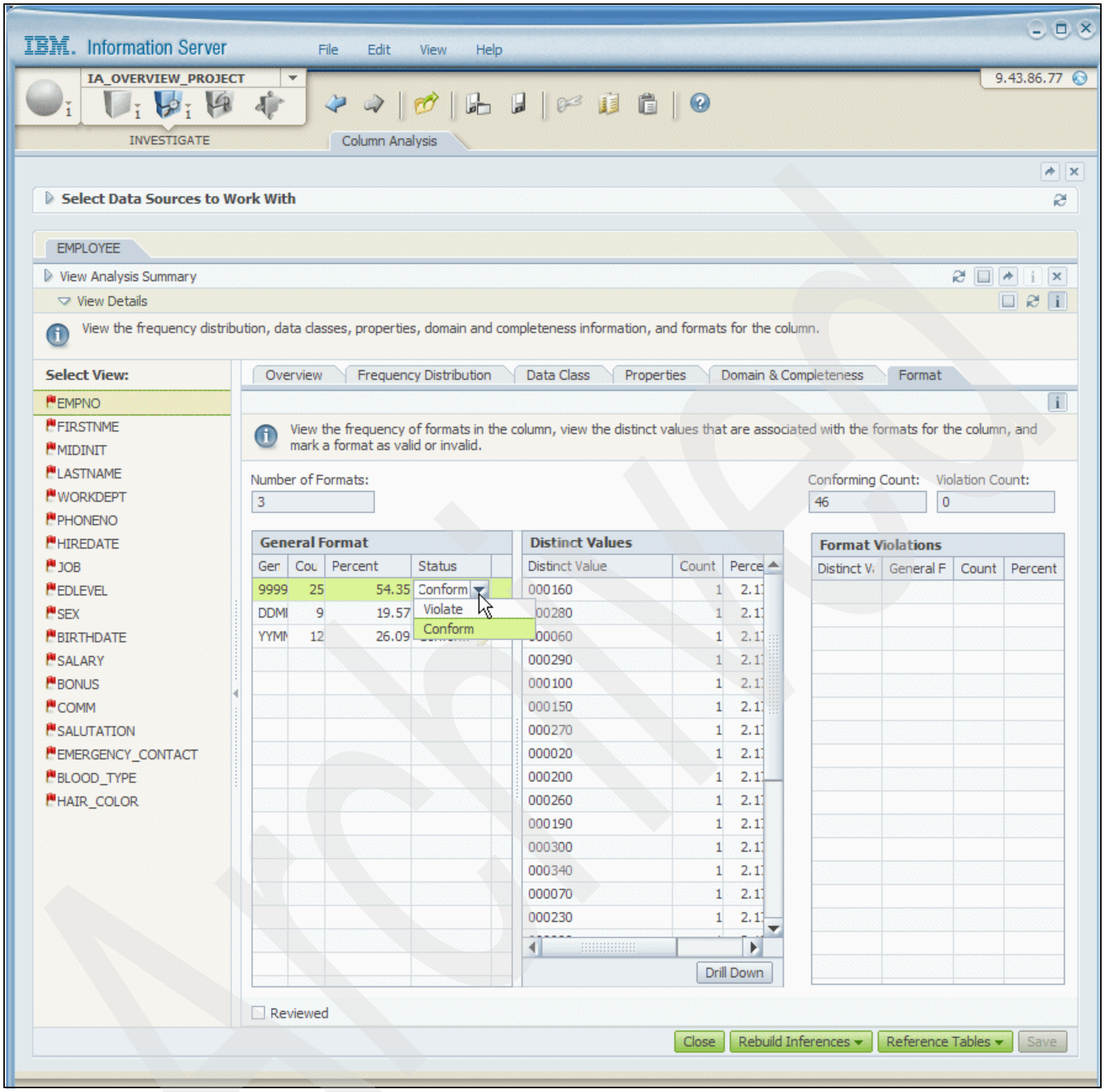
Data Value	Count	Percent

Invalid

Data Va

□ Reviewed

Figure 1-116 View EMPNO column details in the EMPLOYEE table 14/15



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the frequency of formats in the column, view the distinct values that are associated with the formats for the column, and mark a format as valid or invalid.

Number of Formats: 3

Conforming Count: 46

Violation Count: 0

General Format

Ger	Cou	Percent	Status
9999	25	54.35	Conform
DDMI	9	19.57	
YYMI	12	26.09	

Distinct Values

Distinct Value	Count	Perce
000160	1	2.1
00280	1	2.1
00060	1	2.1
000290	1	2.1
000100	1	2.1
000150	1	2.1
000270	1	2.1
000020	1	2.1
000200	1	2.1
000260	1	2.1
000190	1	2.1
000300	1	2.1
000340	1	2.1
000070	1	2.1
000230	1	2.1

Format Violations

Distinct V.	General F	Count	Percent

□ Reviewed

Figure 1-117 View EMPNO column details in the EMPLOYEE table 15/15

4. Figure 1-118 on page 171 through Figure 1-129 on page 182 show the details of column analysis for the SALARY column as follows:

   – Frequency Distribution details are shown in Figure 1-118 on page 171 and Figure 1-106 on page 157. It shows the total number of rows (46), cardinality (number of distinct values in the column) as being 43, (cardinality) percentage which is a calculation of the total number of distinct values in a column divided by the total number of values in the same column (93.48%), and the inferred data class as being a Quantity.

All the data values have the same inferred data type of DECIMAL, and format of 99999.99.

- Data Class details are shown in Figure 1-119 on page 172. The inferred class is Quantity, which is correct.
- Properties details are shown in Figure 1-120 on page 173 through Figure 1-124 on page 177.
  - In Figure 1-120 on page 173, the inferred data type is DECIMAL. Unlike the case of EMPNO, there was only one inferred data type. The Selected data type is the inferred data type, which you can choose to change if incorrect, which in this case is correct. The pie-chart shows the data type distribution of the tabular information, which shows DECIMAL as being inferred 100% of the time for each row.
  - In Figure 1-121 on page 174, the defined, inferred, and selected values for length is *9*, which is correct in this case.
  - In Figure 1-122 on page 175, the defined precision is *9*, but the inferred precision is 8 which is the default selected value. You can choose to go with the default selected value of *8* or correct it to *9* or some other value. As before, additional information such as minimum, median, average, and maximum is provided along with a bar chart representation of the same information.
  - In Figure 1-123 on page 176, the defined scale is *2*, which is the same as the inferred scale and the selected scale. As before, additional information such as minimum, median, average, and maximum is provided along with a bar chart representation of the same information.
  - In Figure 1-124 on page 177, the SALARY column is defined as nullable. However, because all rows had a non-null value in the SALARY column, the inferred value is not null. You can choose to select it as nullable by checking Yes.

    The cardinality type is defined as not constrained and inferred as not constrained. This is because the ratio of distinct values (43) to the total number of records (46) is 93.4783%, and this is below the Unique Threshold of 99%, the constant ratio (the highest frequency percentage for a single data value) is also below the Constant Threshold of 99%. Therefore, the default selection is not constrained, However, as before, you can change the selected value for cardinality type from the drop-down list (Unique and Unique and Constant).

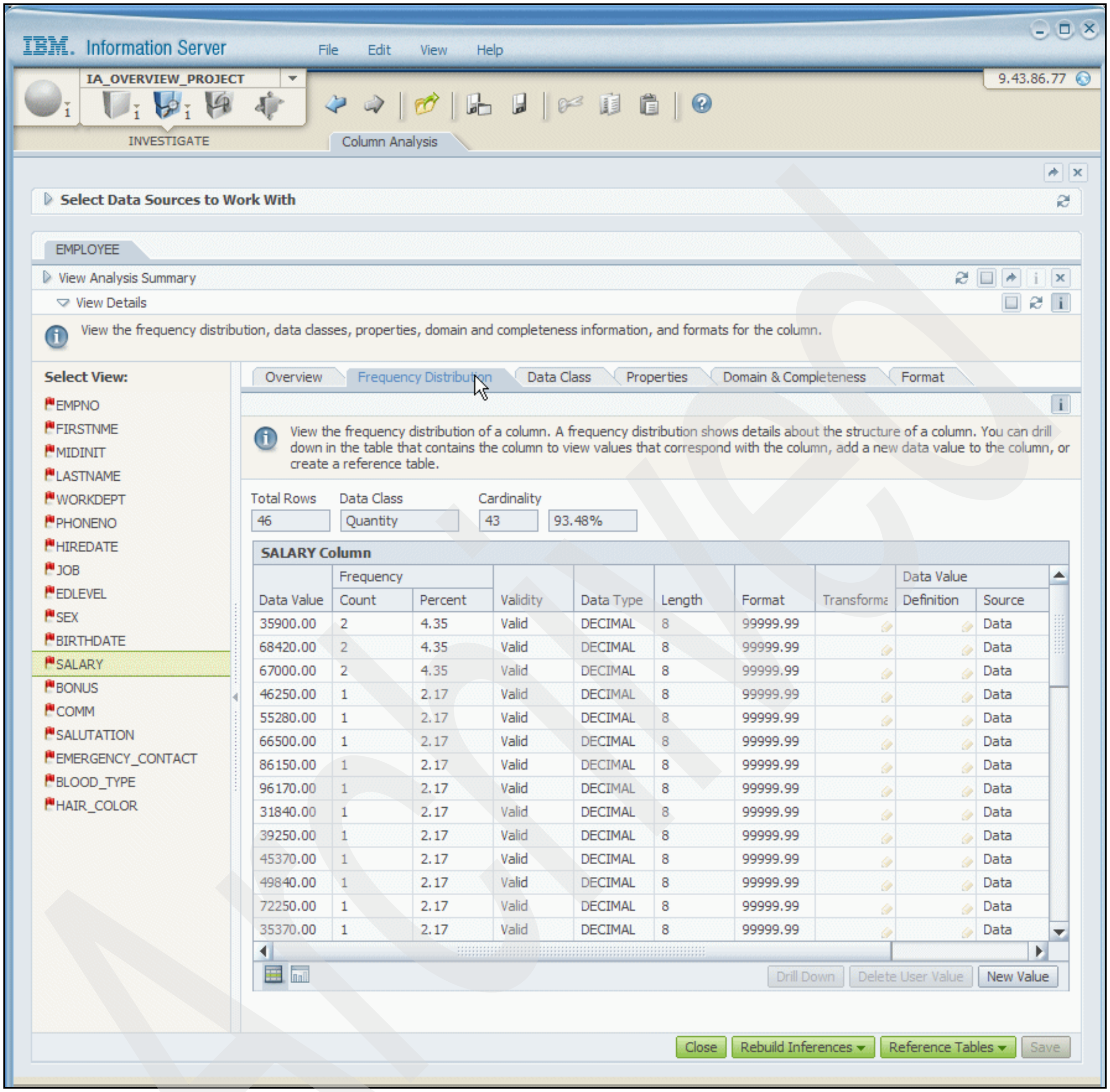
- Domain & Completeness details are shown in Figure 1-125 on page 178 through Figure 1-129 on page 182.

  - In Figure 1-126 on page 179, when the Domain Type is Range, the frequency distribution shows the individual values and the Status. By default, all values are initially identified as being valid. When you specify a Minimum (39000) and Maximum (160000), all values outside this range are marked as being invalid. The values need to be applied and saved to take effect.

    The value you specify in the Outliers field (default is 10) determines the 10 lowest and 10 highest values displayed regardless of whether the values are valid.

    The Completeness Summary shows 43 distinct values, zero incomplete (default) values out of a total of 46 records. No invalid values exist either as per the Validity Summary (shown partially in Figure 1-126 on page 179).

  - Scrolling right, you can change the Status as discussed earlier and the Min/Max column to Valid, Minimum, or Maximum as shown in Figure 1-129 on page 182. You can set individual values in the grid of the Frequency Distribution pane to valid. You can also select a particular value in the grid and mark it as Minimum or Maximum instead of specifying these values in the Minimum and Maximum fields mentioned earlier. Here again, you have to apply and save the changes to take effect.



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

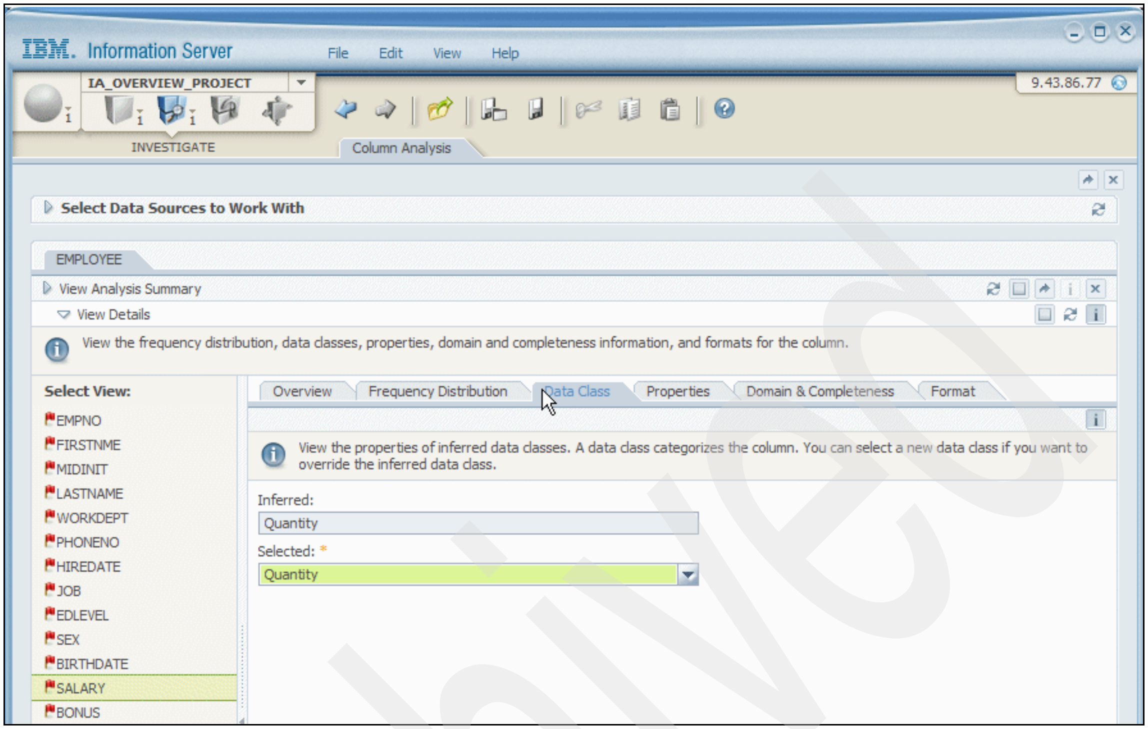
View the frequency distribution of a column. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view values that correspond with the column, add a new data value to the column, or create a reference table.

Total Rows: 46
Data Class: Quantity
Cardinality: 43 93.48%

SALARY Column

	Frequency							Data Value	
Data Value	Count	Percent	Validity	Data Type	Length	Format	Transforma	Definition	Source
35900.00	2	4.35	Valid	DECIMAL	8	99999.99			Data
68420.00	2	4.35	Valid	DECIMAL	8	99999.99			Data
67000.00	2	4.35	Valid	DECIMAL	8	99999.99			Data
46250.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
55280.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
66500.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
86150.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
96170.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
31840.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
39250.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
45370.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
49840.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
72250.00	1	2.17	Valid	DECIMAL	8	99999.99			Data
35370.00	1	2.17	Valid	DECIMAL	8	99999.99			Data

Figure 1-118 View SALARY column details in the EMPLOYEE table 1/12



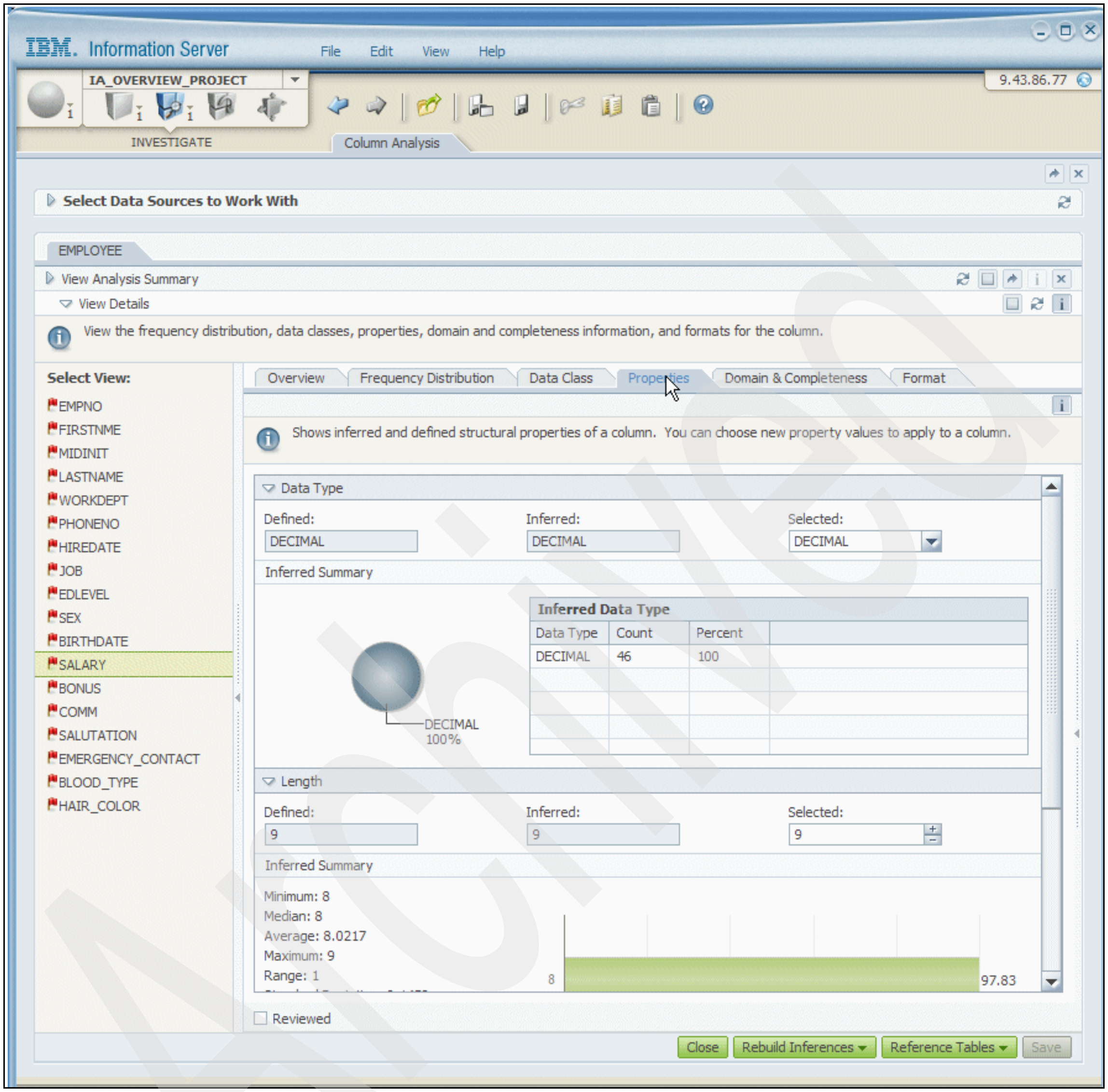
View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the properties of inferred data classes. A data class categorizes the column. You can select a new data class if you want to override the inferred data class.

Inferred: Quantity

Selected: * Quantity

Figure 1-119 View SALARY column details in the EMPLOYEE table 2/12



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Data Type

Defined: DECIMAL
Inferred: DECIMAL
Selected: DECIMAL

Inferred Data Type

Data Type	Count	Percent
DECIMAL	46	100

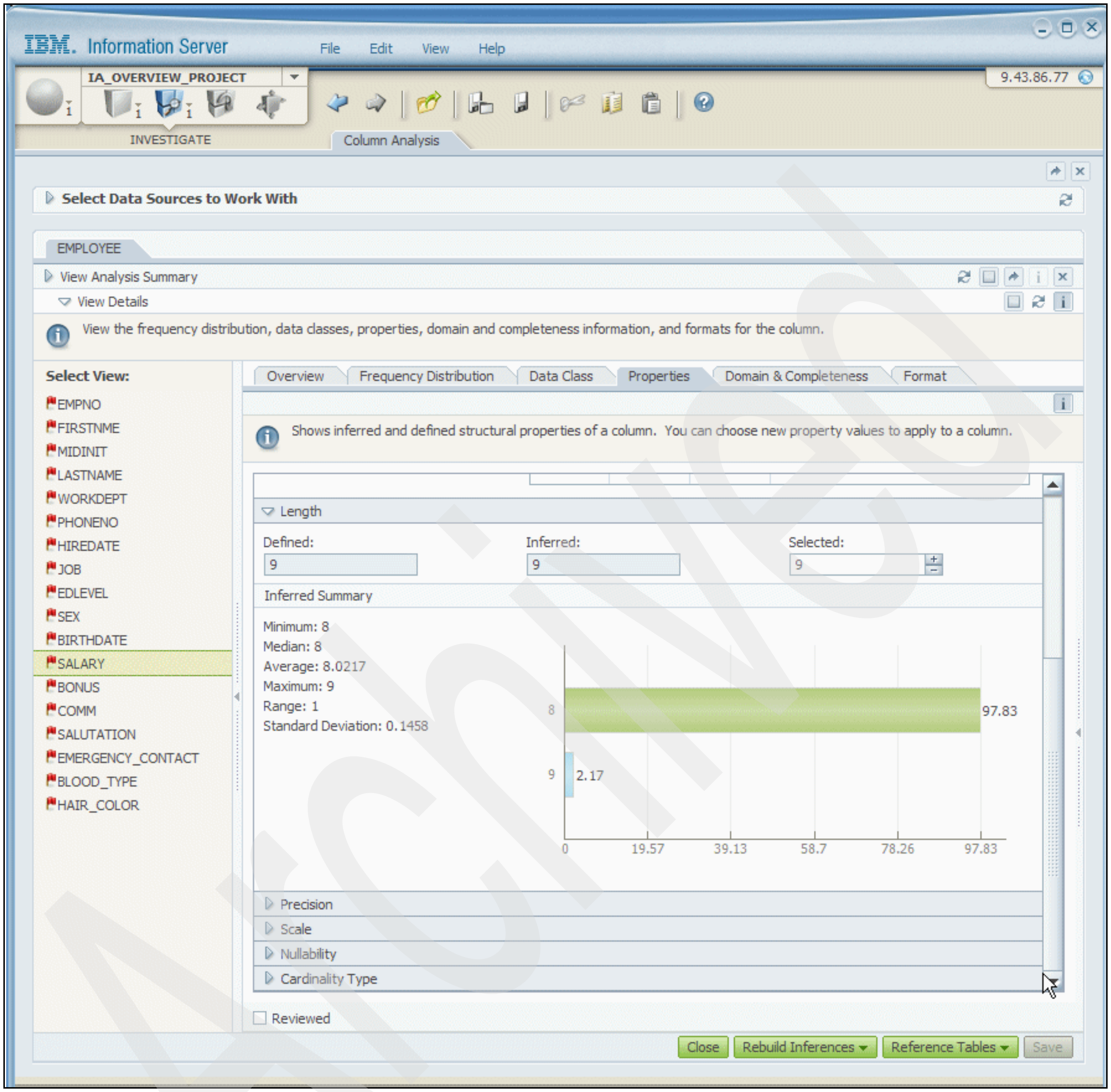
Length

Defined: 9
Inferred: 9
Selected: 9

Minimum: 8
Median: 8
Average: 8.0217
Maximum: 9
Range: 1

□ Reviewed

Figure 1-120 View SALARY column details in the EMPLOYEE table 3/12



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

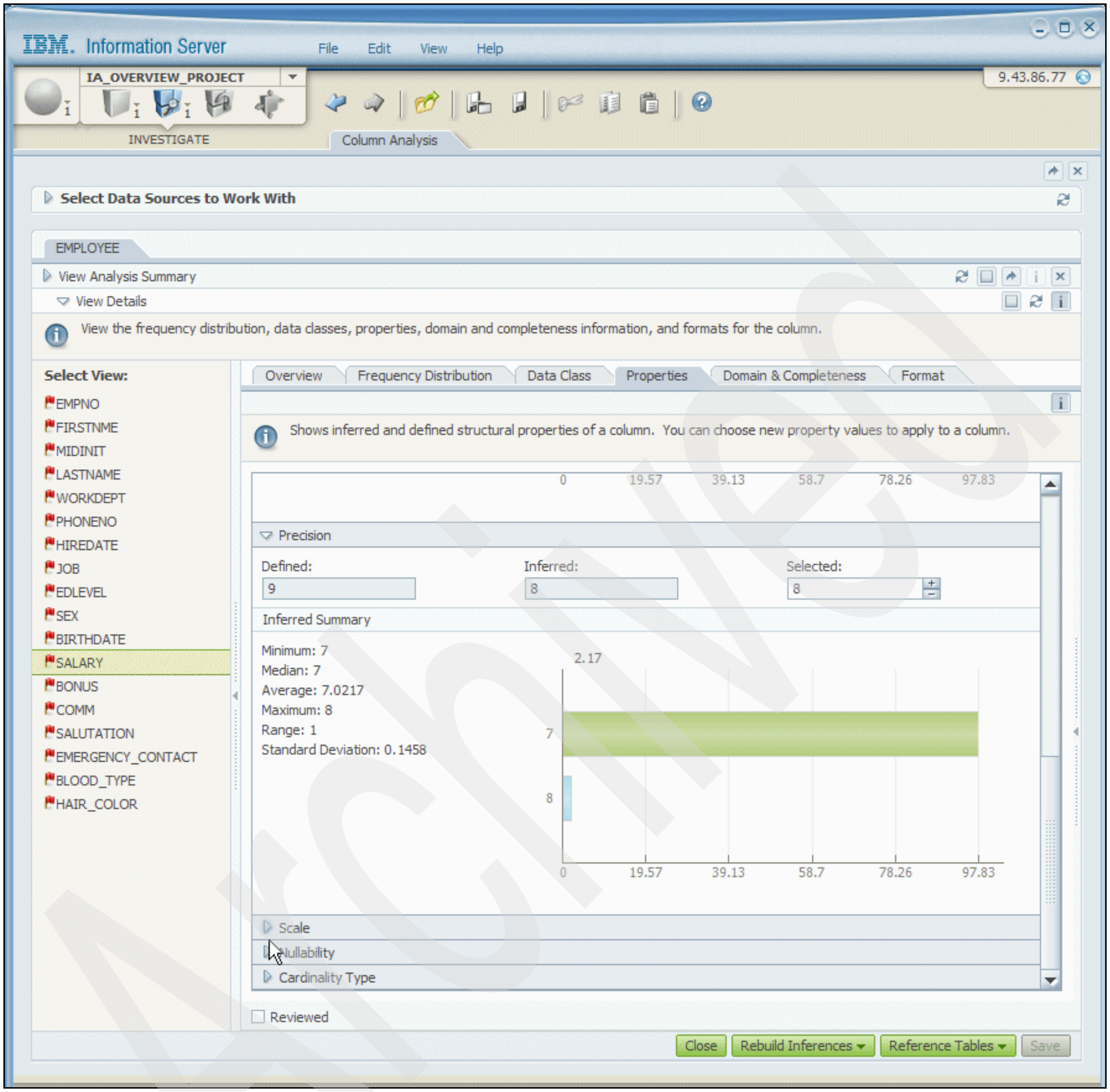
Defined: 9

Inferred: 9

Selected: 9

Minimum: 8
Median: 8
Average: 8.0217
Maximum: 9
Range: 1
Standard Deviation: 0.1458

Figure 1-121 View SALARY column details in the EMPLOYEE table 4/12



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Precision

Defined: 9

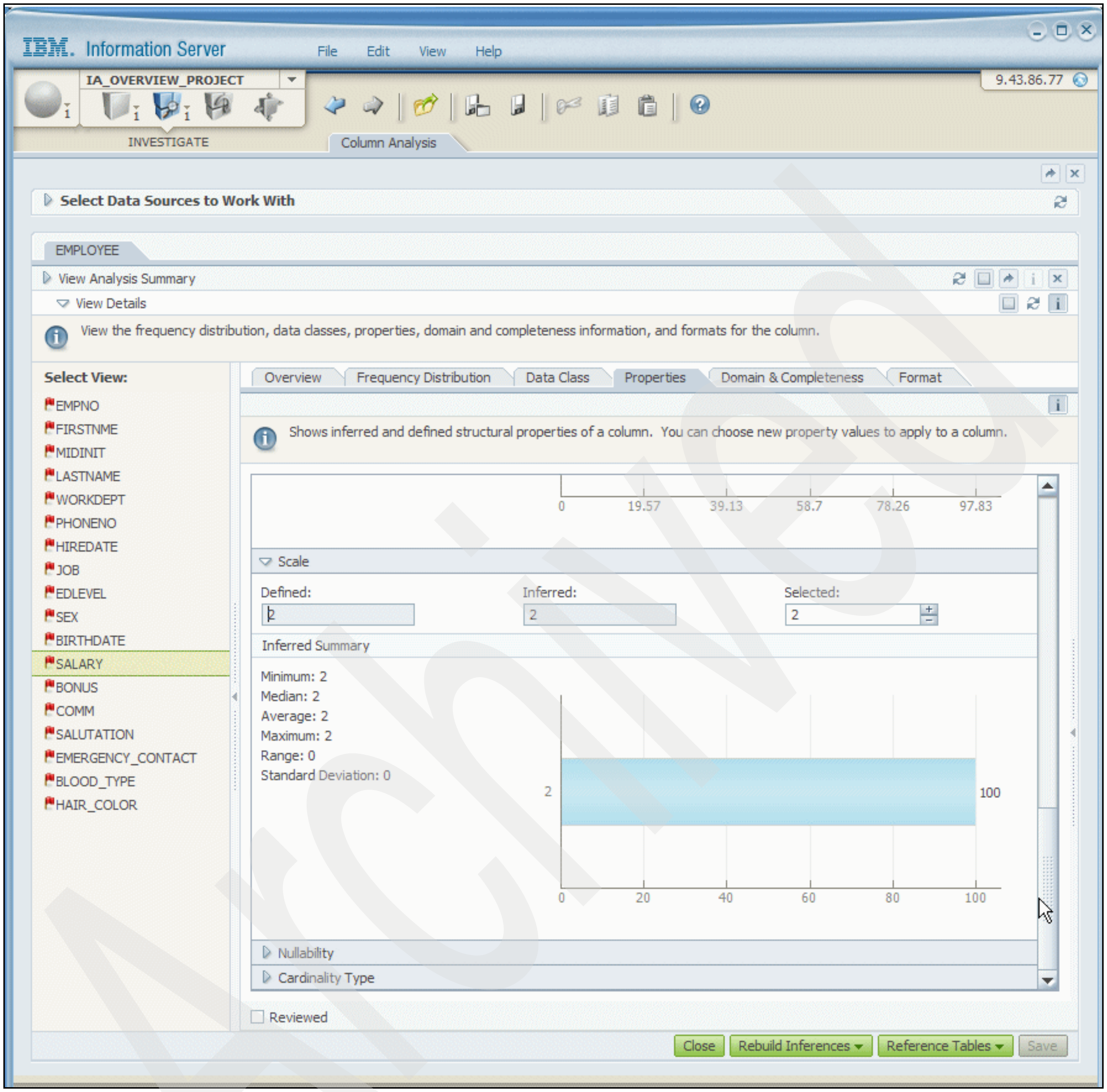
Inferred: 8

Selected: 8

Inferred Summary

Minimum: 7
Median: 7
Average: 7.0217
Maximum: 8
Range: 1
Standard Deviation: 0.1458

Figure 1-122 View SALARY column details in the EMPLOYEE table 5/12



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Defined: 2

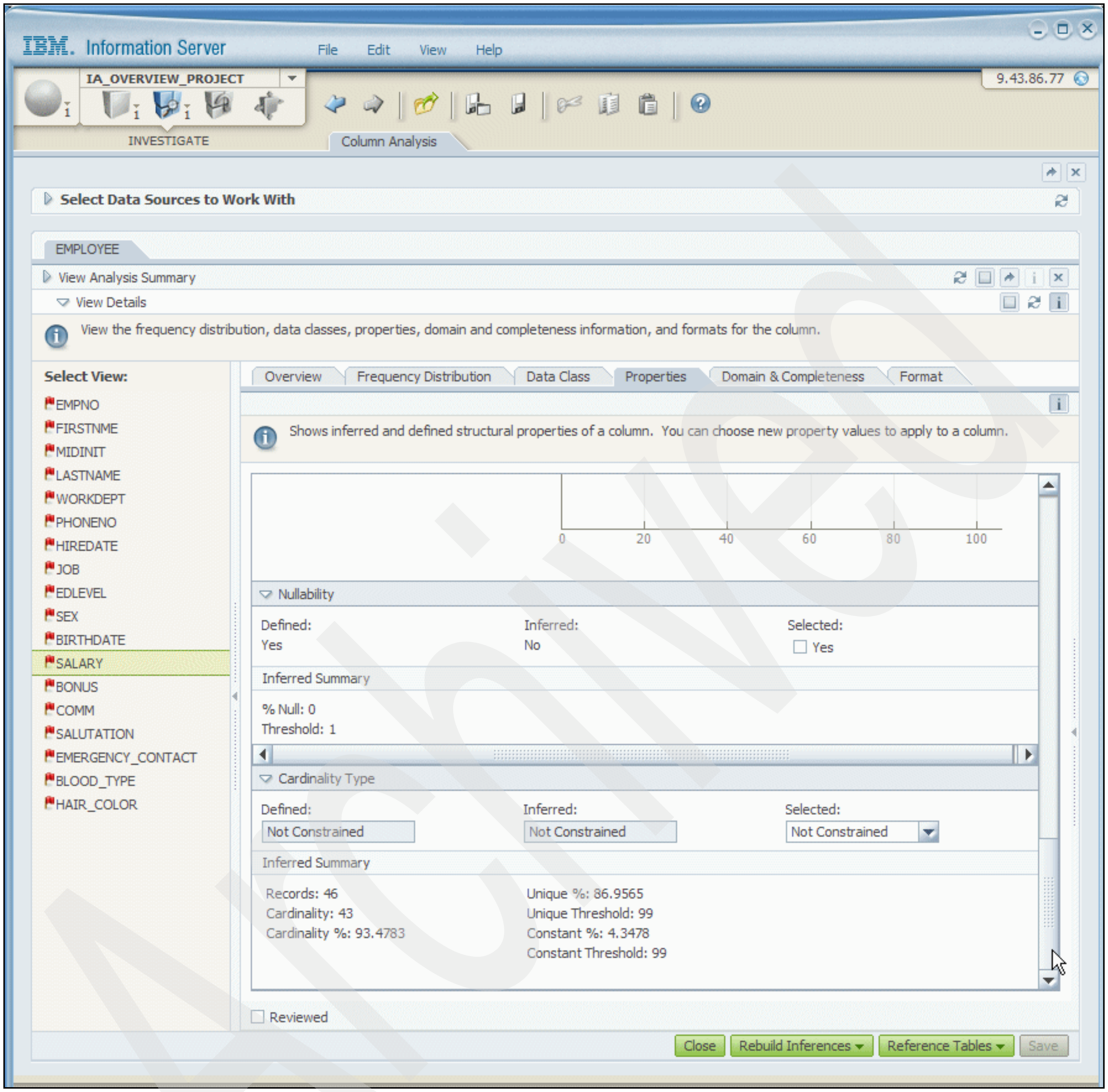
Inferred: 2

Selected: 2

Inferred Summary

Minimum: 2
Median: 2
Average: 2
Maximum: 2
Range: 0
Standard Deviation: 0

Figure 1-123 View SALARY column details in the EMPLOYEE table 6/12



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Nullability

Defined: Yes
Inferred: No
Selected: □ Yes

Inferred Summary

% Null: 0
Threshold: 1

Cardinality Type

Defined: Not Constrained
Inferred: Not Constrained
Selected: Not Constrained

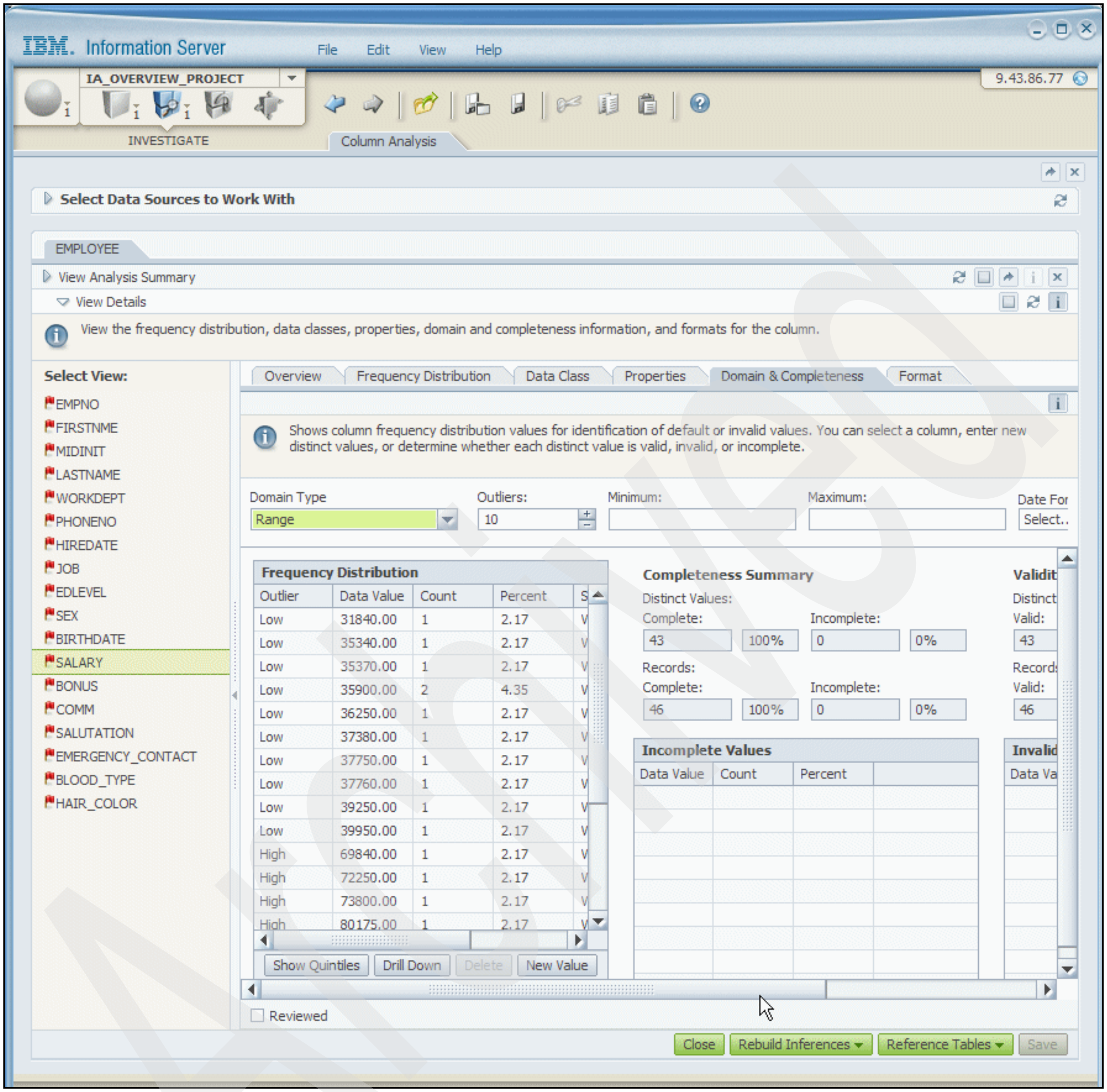
Inferred Summary

Records: 46
Cardinality: 43
Cardinality %: 93.4783

Unique %: 86.9565
Unique Threshold: 99
Constant %: 4.3478
Constant Threshold: 99

□ Reviewed

Figure 1-124 View SALARY column details in the EMPLOYEE table 7/12



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Range
Outliers: 10
Minimum:
Maximum:
Date For: Select..

Frequency Distribution

Outlier	Data Value	Count	Percent	S
Low	31840.00	1	2.17	V
Low	35340.00	1	2.17	V
Low	35370.00	1	2.17	V
Low	35900.00	2	4.35	V
Low	36250.00	1	2.17	V
Low	37380.00	1	2.17	V
Low	37750.00	1	2.17	V
Low	37760.00	1	2.17	V
Low	39250.00	1	2.17	V
Low	39950.00	1	2.17	V
High	69840.00	1	2.17	V
High	72250.00	1	2.17	V
High	73800.00	1	2.17	V
High	80175.00	1	2.17	V

Completeness Summary

Distinct Values:
Complete: 43 100%
Incomplete: 0 0%

Records:
Complete: 46 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 43

Record
Valid: 46

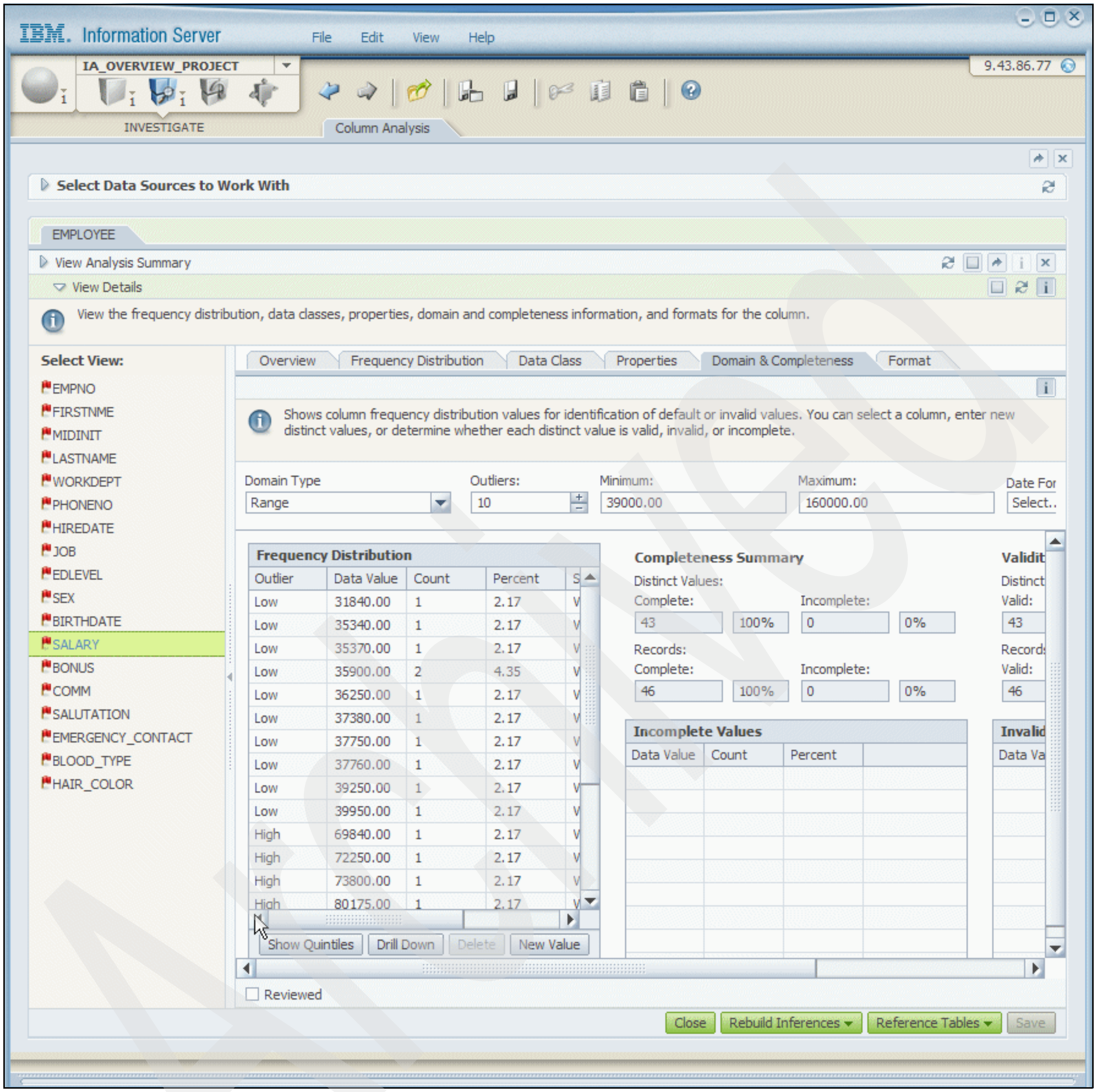
Incomplete Values

Data Value	Count	Percent

Invalid

Data Va

Figure 1-125 View SALARY column details in the EMPLOYEE table 8/12



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Range
Outliers: 10
Minimum: 39000.00
Maximum: 160000.00
Date For: Select..

Frequency Distribution

Outlier	Data Value	Count	Percent	S
Low	31840.00	1	2.17	V
Low	35340.00	1	2.17	V
Low	35370.00	1	2.17	V
Low	35900.00	2	4.35	V
Low	36250.00	1	2.17	V
Low	37380.00	1	2.17	V
Low	37750.00	1	2.17	V
Low	37760.00	1	2.17	V
Low	39250.00	1	2.17	V
Low	39950.00	1	2.17	V
High	69840.00	1	2.17	V
High	72250.00	1	2.17	V
High	73800.00	1	2.17	V
High	80175.00	1	2.17	V

Completeness Summary

Distinct Values:
Complete: 43 100%
Incomplete: 0 0%

Records:
Complete: 46 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 43

Record
Valid: 46

Incomplete Values

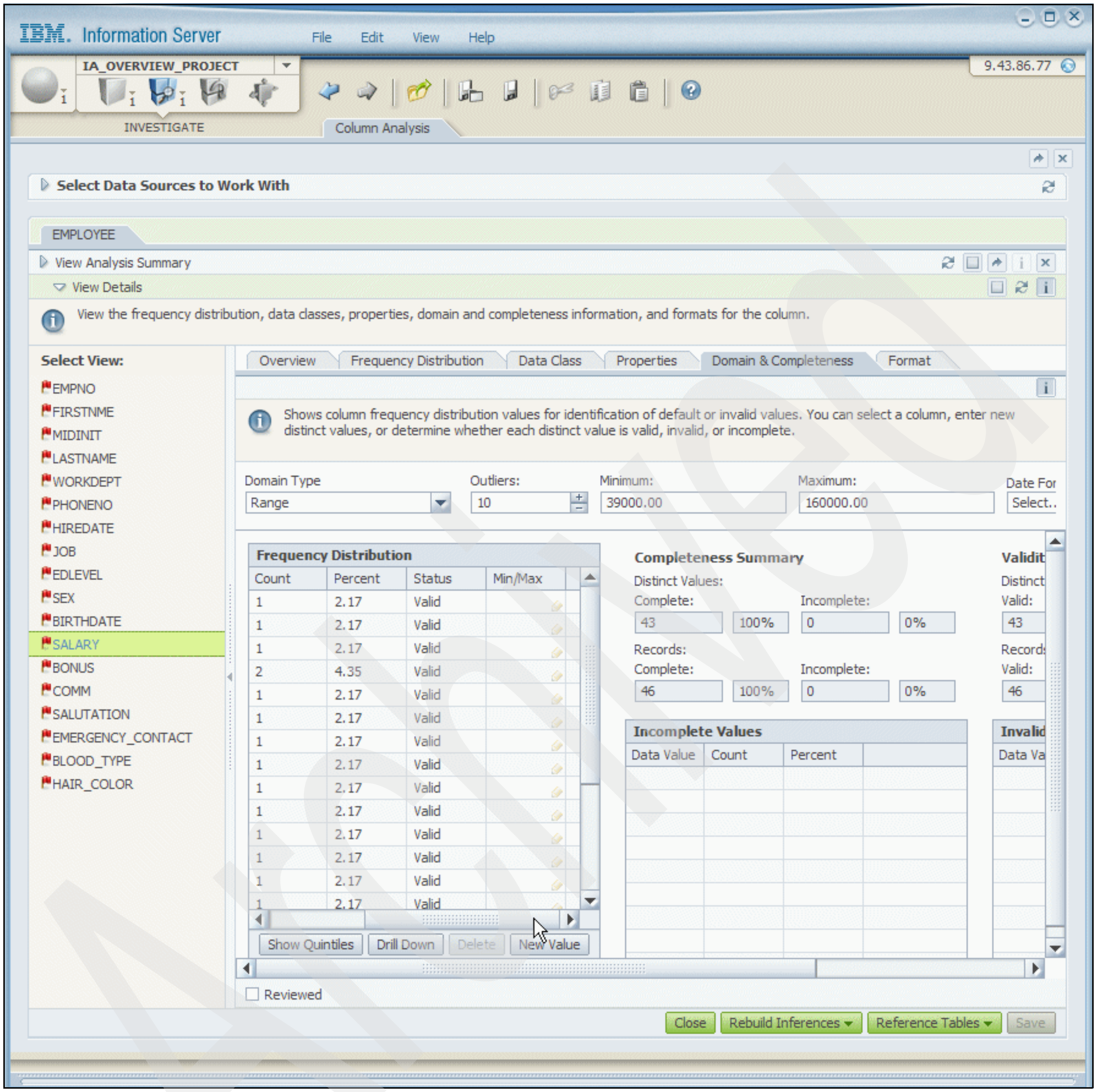
Data Value	Count	Percent

Invalid

Data Va

□ Reviewed

Figure 1-126 View SALARY column details in the EMPLOYEE table 9/12



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Range
Outliers: 10
Minimum: 39000.00
Maximum: 160000.00
Date For: Select..

Frequency Distribution

Count	Percent	Status	Min/Max
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
2	4.35	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	

Completeness Summary

Distinct Values:
Complete: 43 100%
Incomplete: 0 0%

Records:
Complete: 46 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 43

Record:
Valid: 46

Incomplete Values

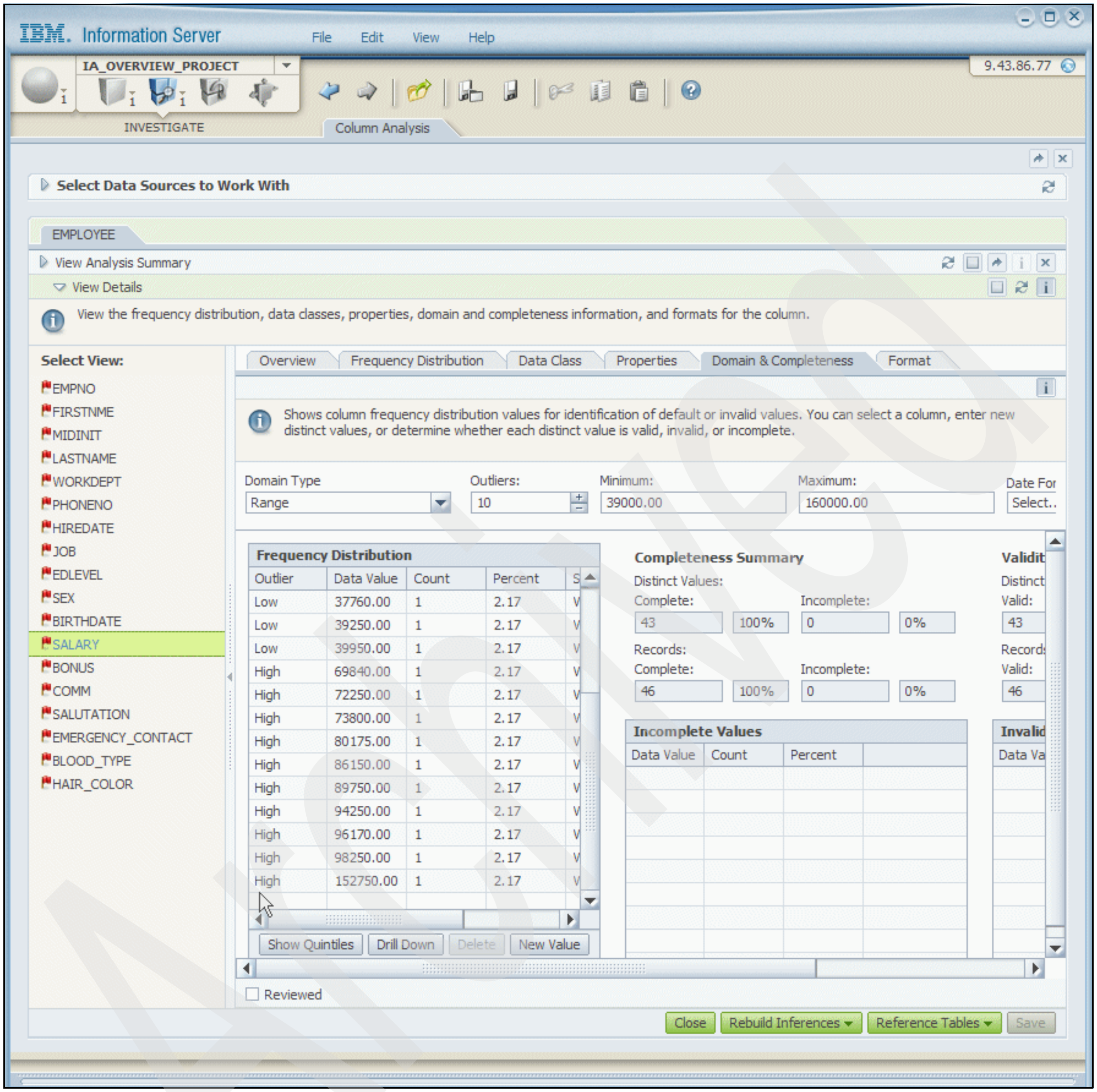
Data Value	Count	Percent

Invalid

Data Va

□ Reviewed

Figure 1-127 View SALARY column details in the EMPLOYEE table 10/12



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Range
Outliers: 10
Minimum: 39000.00
Maximum: 160000.00
Date For: Select..

Frequency Distribution

Outlier	Data Value	Count	Percent	S
Low	37760.00	1	2.17	V
Low	39250.00	1	2.17	V
Low	39950.00	1	2.17	V
High	69840.00	1	2.17	V
High	72250.00	1	2.17	V
High	73800.00	1	2.17	V
High	80175.00	1	2.17	V
High	86150.00	1	2.17	V
High	89750.00	1	2.17	V
High	94250.00	1	2.17	V
High	96170.00	1	2.17	V
High	98250.00	1	2.17	V
High	152750.00	1	2.17	V

Completeness Summary

Distinct Values:
Complete: 43 100%
Incomplete: 0 0%

Records:
Complete: 46 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 43

Record
Valid: 46

Incomplete Values

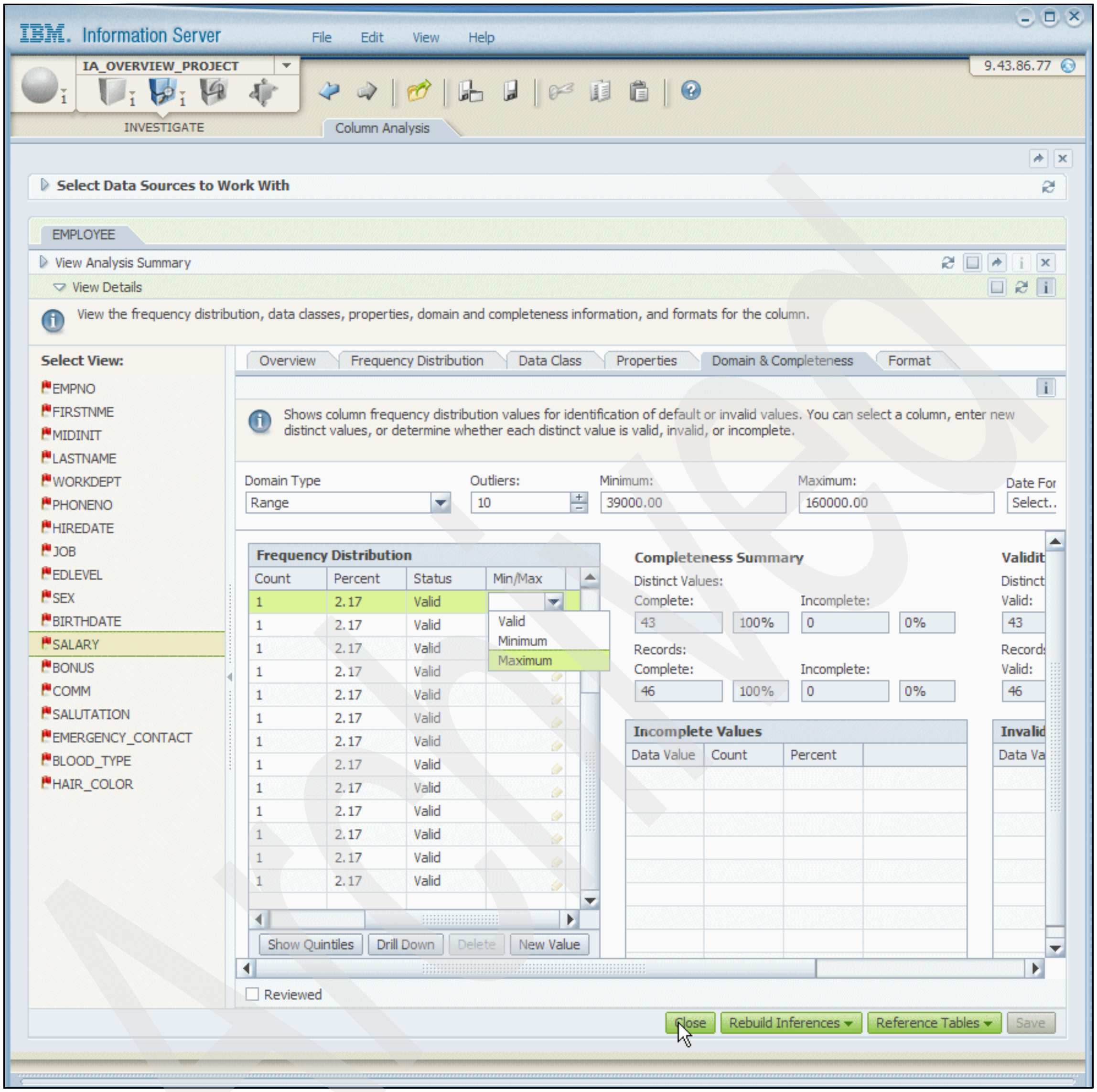
Data Value	Count	Percent

Invalid

Data Va

☐ Reviewed

Figure 1-128 View SALARY column details in the EMPLOYEE table 11/12



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Range
Outliers: 10
Minimum: 39000.00
Maximum: 160000.00
Date For: Select..

Frequency Distribution

Count	Percent	Status	Min/Max
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	
1	2.17	Valid	

Completeness Summary

Distinct Values:
Complete: 43 100%
Incomplete: 0 0%

Records:
Complete: 46 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 43

Record
Valid: 46

Incomplete Values

Data Value	Count	Percent

Invalid

Data Va

□ Reviewed

Figure 1-129 View SALARY column details in the EMPLOYEE table 12/12

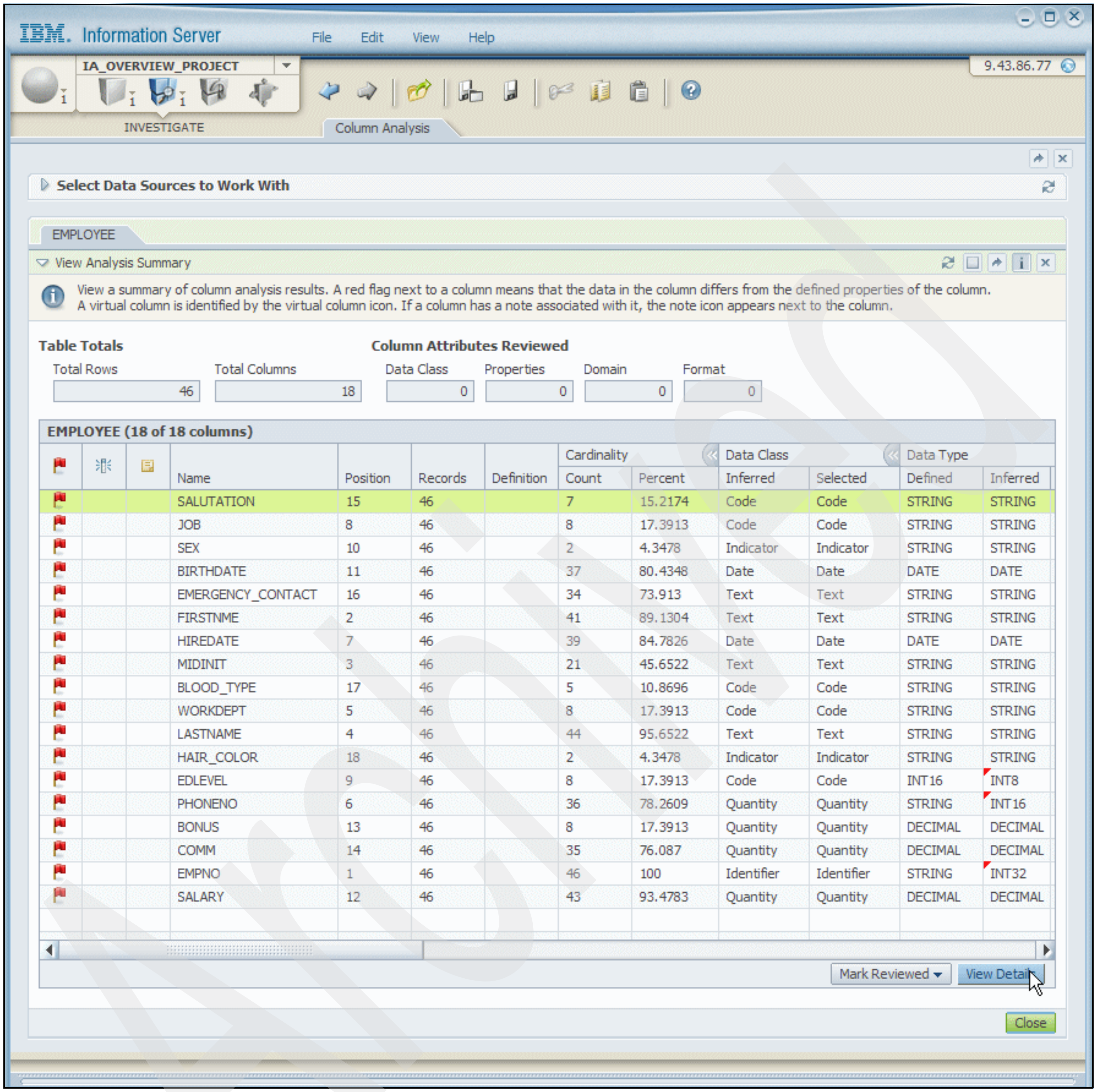
Set invalid values in frequency distribution

Figure 1-130 on page 184 through Figure 1-137 on page 191 describe the setting of the status of certain values in a column to invalid and show how that information is used in a subsequent column analysis to identify invalid values to rows that might have been added or updated.

1. Figure 1-130 on page 184 shows the View Analysis Summary of the EMPLOYEE table. Select the SALUTATION column and click **View Details**. The frequency distribution for this column is shown in Figure 1-131 on page 185.
2. The Domain & Completeness details for the SALUTATION column is shown in Figure 1-132 on page 186. Select the data value **Mr** and change its status to *Invalid* as shown in Figure 1-133 on page 187. The number of distinct values is *7* in the Completeness Summary. Click **Save**.
3. The Validity Summary now reflects the value *Mr* as being invalid (10 occurrences in the Count field) as shown in *Figure 1-134 on page 188*. The Validity Summary now shows the number of distinct values as being *6*, with one invalid value. The number of valid records shows *36*. This information is stored in the analysis database.
4. We then added a couple of rows to the EMPLOYEE table using SQL INSERT statements with a salutation of *Mr*. We do not show that here.
5. We then ran column analysis again on the EMPLOYEE table as shown in Figure 1-135 on page 189.
6. The Domain & Completeness details of the SALUTATION column are shown in Figure 1-136 on page 190 and Figure 1-137 on page 191. It shows the frequency distribution after this column analysis. The total rows is now *48*, and the count of invalid values of *Mr* is now *12* (two more than shown previously in Figure 1-134 on page 188).

> **Note:** The saved status of invalid values is used in subsequent column analysis as long as the option “Update Existing Tables” is chosen as shown in Figure 1-92 on page 142.

Note the of Code, and the different formats of the content of the SALUTATION column. The Validity Summary in Figure 1-137 on page 191 shows the count of 12 invalid values with the *Mr* data value.



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View a summary of column analysis results. A red flag next to a column means that the data in the column differs from the defined properties of the column. A virtual column is identified by the virtual column icon. If a column has a note associated with it, the note icon appears next to the column.

Table Totals

Total Rows: 46

Total Columns: 18

Column Attributes Reviewed

Data Class: 0

Properties: 0

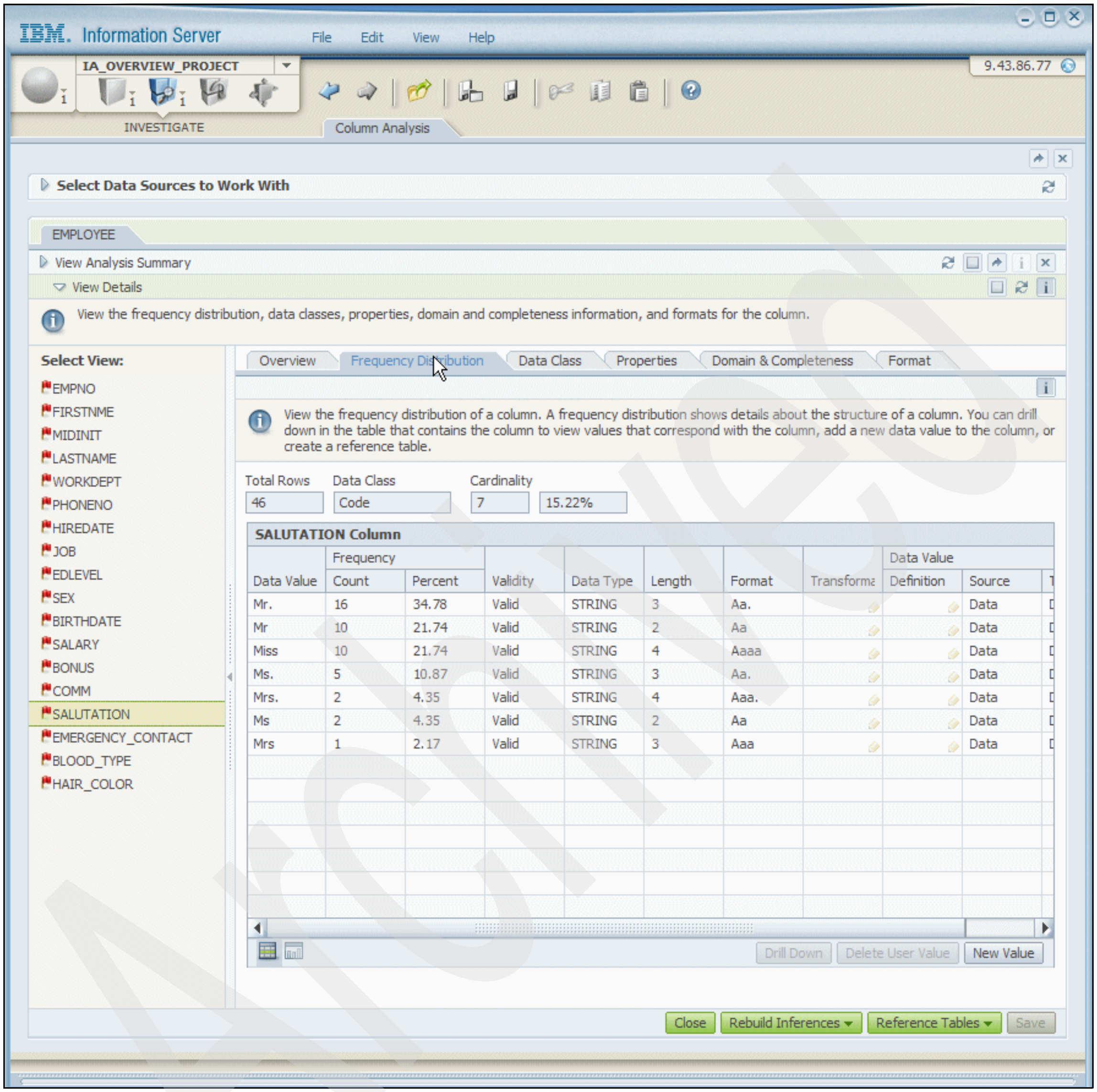
Domain: 0

Format: 0

EMPLOYEE (18 of 18 columns)

							Cardinality		Data Class		Data Type	
⚑			Name	Position	Records	Definition	Count	Percent	Inferred	Selected	Defined	Inferred
⚑			SALUTATION	15	46		7	15.2174	Code	Code	STRING	STRING
⚑			JOB	8	46		8	17.3913	Code	Code	STRING	STRING
⚑			SEX	10	46		2	4.3478	Indicator	Indicator	STRING	STRING
⚑			BIRTHDATE	11	46		37	80.4348	Date	Date	DATE	DATE
⚑			EMERGENCY_CONTACT	16	46		34	73.913	Text	Text	STRING	STRING
⚑			FIRSTNME	2	46		41	89.1304	Text	Text	STRING	STRING
⚑			HIREDATE	7	46		39	84.7826	Date	Date	DATE	DATE
⚑			MIDINIT	3	46		21	45.6522	Text	Text	STRING	STRING
⚑			BLOOD_TYPE	17	46		5	10.8696	Code	Code	STRING	STRING
⚑			WORKDEPT	5	46		8	17.3913	Code	Code	STRING	STRING
⚑			LASTNAME	4	46		44	95.6522	Text	Text	STRING	STRING
⚑			HAIR_COLOR	18	46		2	4.3478	Indicator	Indicator	STRING	STRING
⚑			EDLEVEL	9	46		8	17.3913	Code	Code	INT16	INT8
⚑			PHONENO	6	46		36	78.2609	Quantity	Quantity	STRING	INT16
⚑			BONUS	13	46		8	17.3913	Quantity	Quantity	DECIMAL	DECIMAL
⚑			COMM	14	46		35	76.087	Quantity	Quantity	DECIMAL	DECIMAL
⚑			EMPNO	1	46		46	100	Identifier	Identifier	STRING	INT32
⚑			SALARY	12	46		43	93.4783	Quantity	Quantity	DECIMAL	DECIMAL

Figure 1-130 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 1/8



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

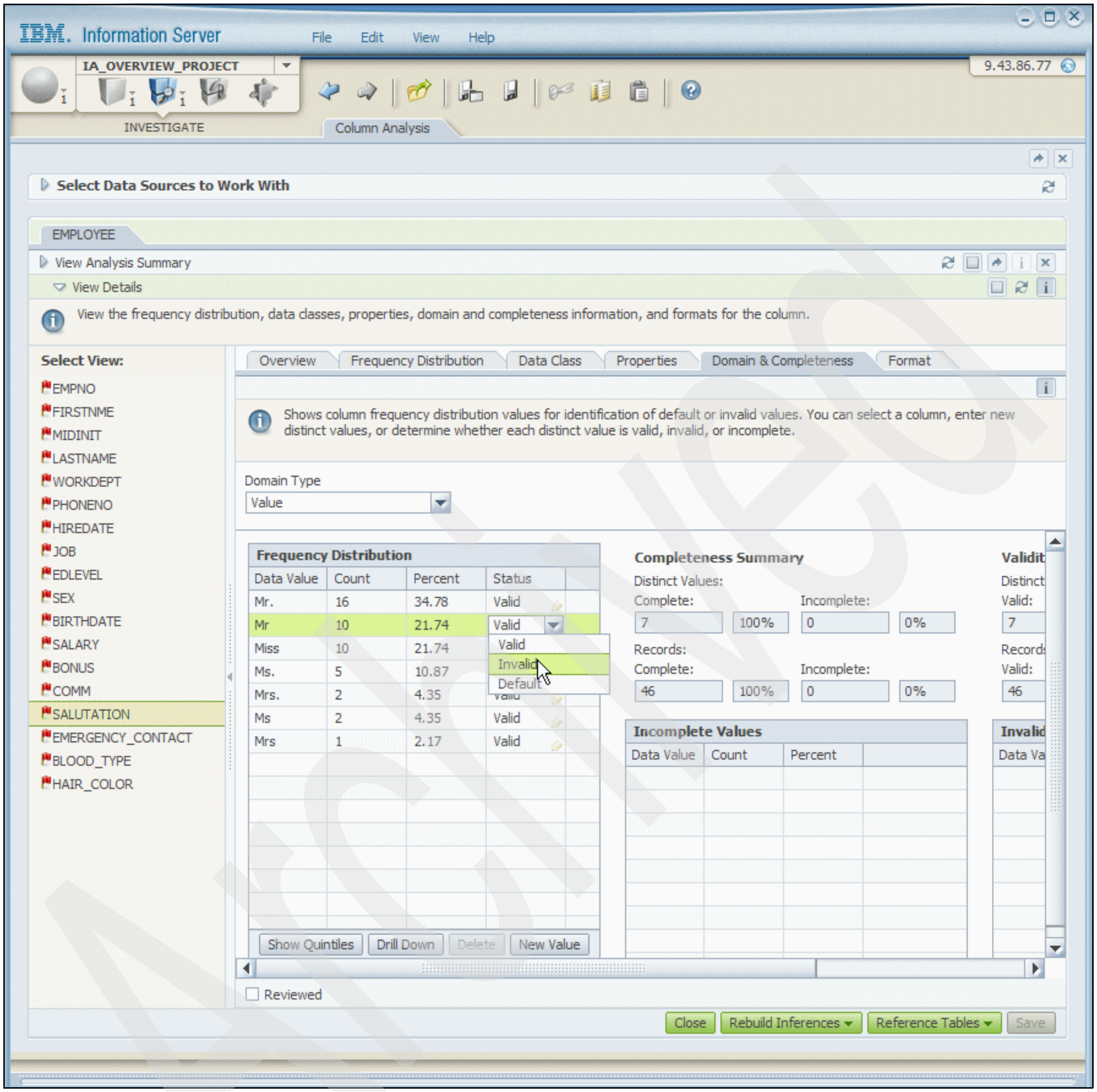
View the frequency distribution of a column. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view values that correspond with the column, add a new data value to the column, or create a reference table.

Total Rows: 46
Data Class: Code
Cardinality: 7 15.22%

SALUTATION Column

	Frequency							Data Value		
Data Value	Count	Percent	Validity	Data Type	Length	Format	Transforma	Definition	Source	T
Mr.	16	34.78	Valid	STRING	3	Aa.			Data	D
Mr	10	21.74	Valid	STRING	2	Aa			Data	D
Miss	10	21.74	Valid	STRING	4	Aaaa			Data	D
Ms.	5	10.87	Valid	STRING	3	Aa.			Data	D
Mrs.	2	4.35	Valid	STRING	4	Aaa.			Data	D
Ms	2	4.35	Valid	STRING	2	Aa			Data	D
Mrs	1	2.17	Valid	STRING	3	Aaa			Data	D

Figure 1-131 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 2/8



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
Mr.	16	34.78	Valid
Mr	10	21.74	Valid
Miss	10	21.74	
Ms.	5	10.87	
Mrs.	2	4.35	Valid
Ms	2	4.35	Valid
Mrs	1	2.17	Valid

Completeness Summary

Distinct Values:
Complete: 7 100% Incomplete: 0 0%

Records:
Complete: 46 100% Incomplete: 0 0%

Validit

Distinct
Valid: 7

Record
Valid: 46

Incomplete Values

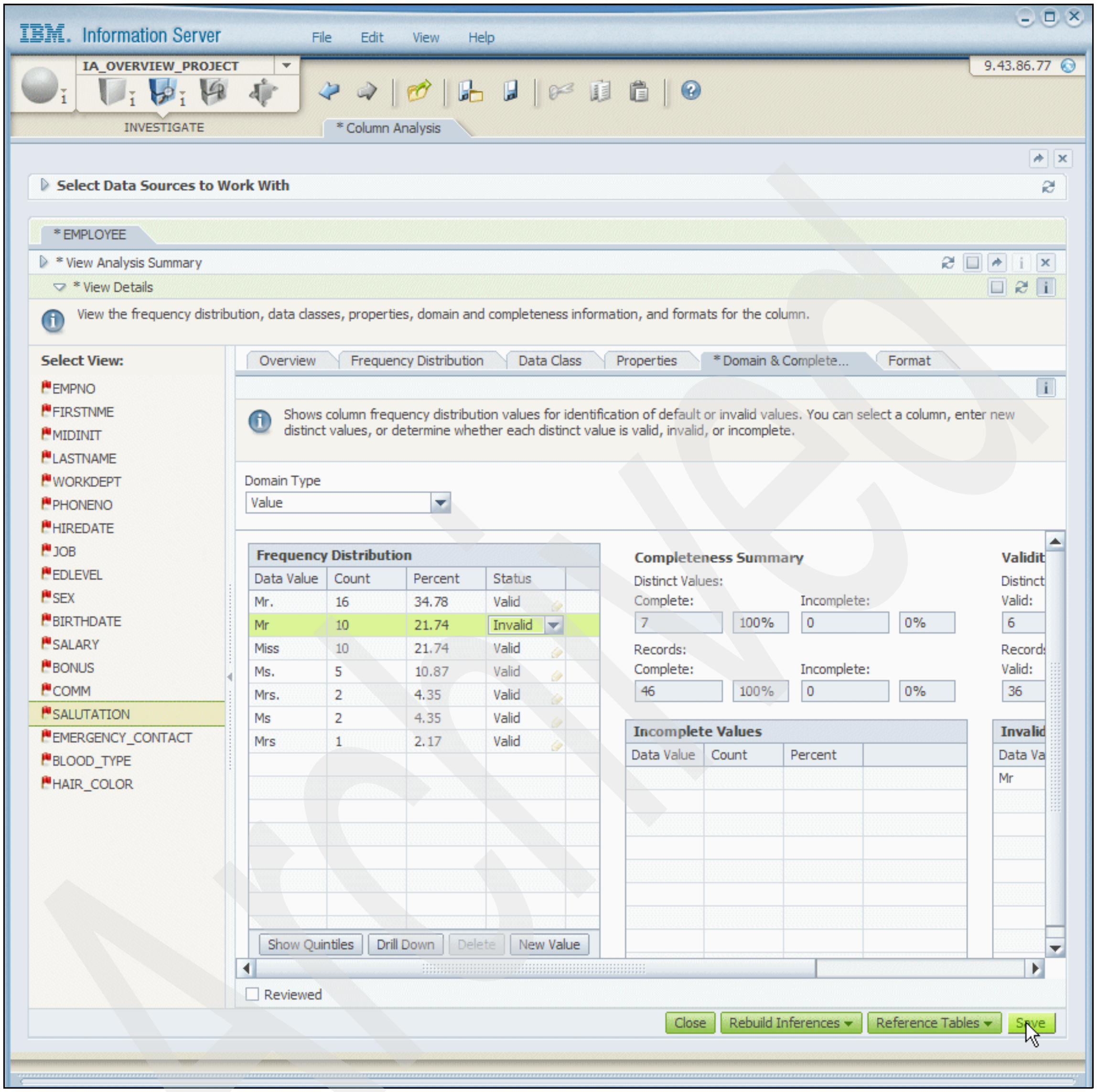
Data Value	Count	Percent

Invalid

Data Va

□ Reviewed

Figure 1-132 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 3/8



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
Mr.	16	34.78	Valid
Mr	10	21.74	Invalid
Miss	10	21.74	Valid
Ms.	5	10.87	Valid
Mrs.	2	4.35	Valid
Ms	2	4.35	Valid
Mrs	1	2.17	Valid

Completeness Summary

Distinct Values:

Complete:		Incomplete:	
7	100%	0	0%

Records:

Complete:		Incomplete:	
46	100%	0	0%

Validit

Distinct
Valid: 6

Record
Valid: 36

Incomplete Values

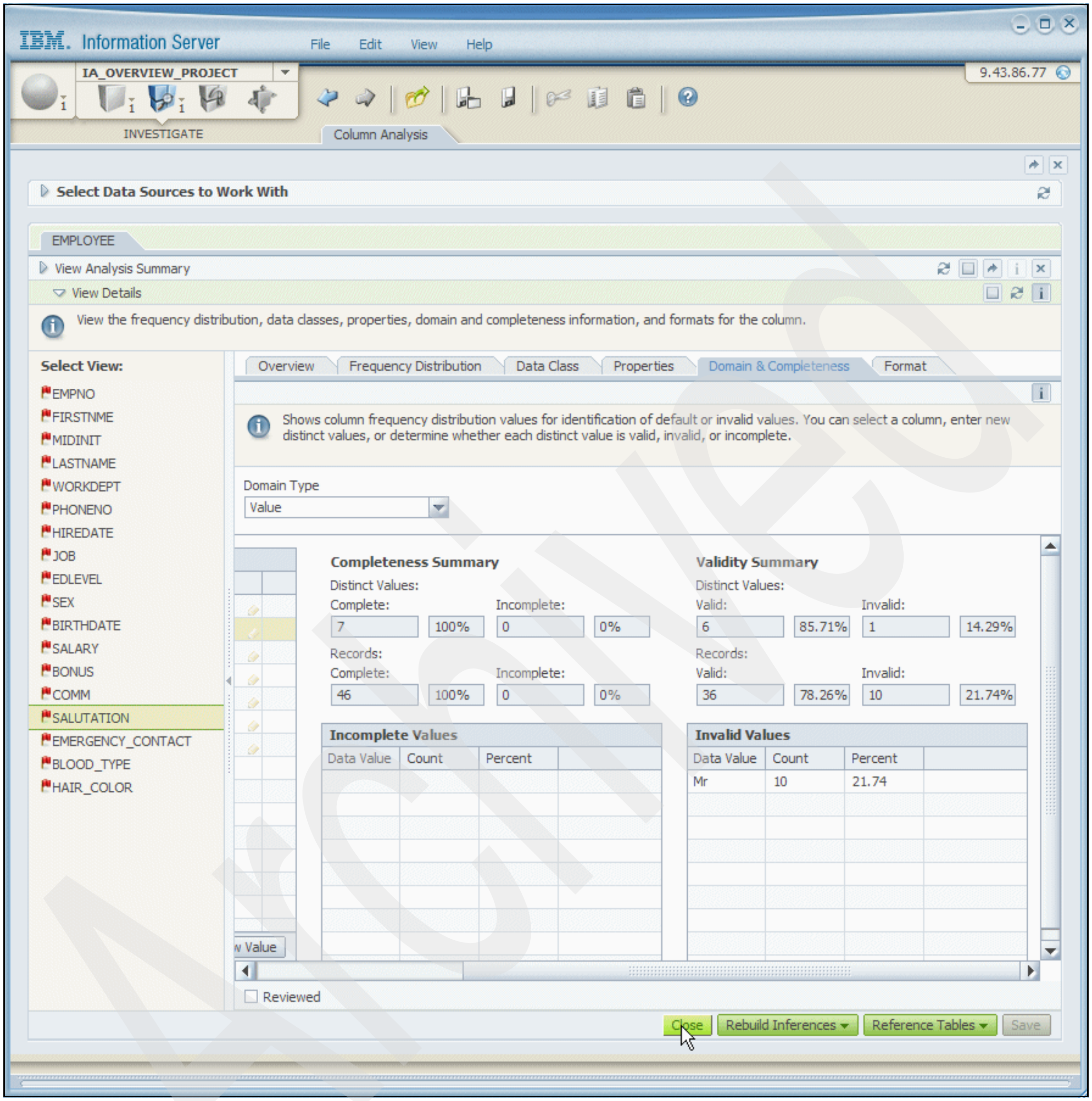
Data Value	Count	Percent

Invalid

Data Va
Mr

□ Reviewed

Figure 1-133 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 4/8



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Completeness Summary

Distinct Values:

Complete:		Incomplete:	
7	100%	0	0%

Records:

Complete:		Incomplete:	
46	100%	0	0%

Validity Summary

Distinct Values:

Valid:		Invalid:	
6	85.71%	1	14.29%

Records:

Valid:		Invalid:	
36	78.26%	10	21.74%

Incomplete Values

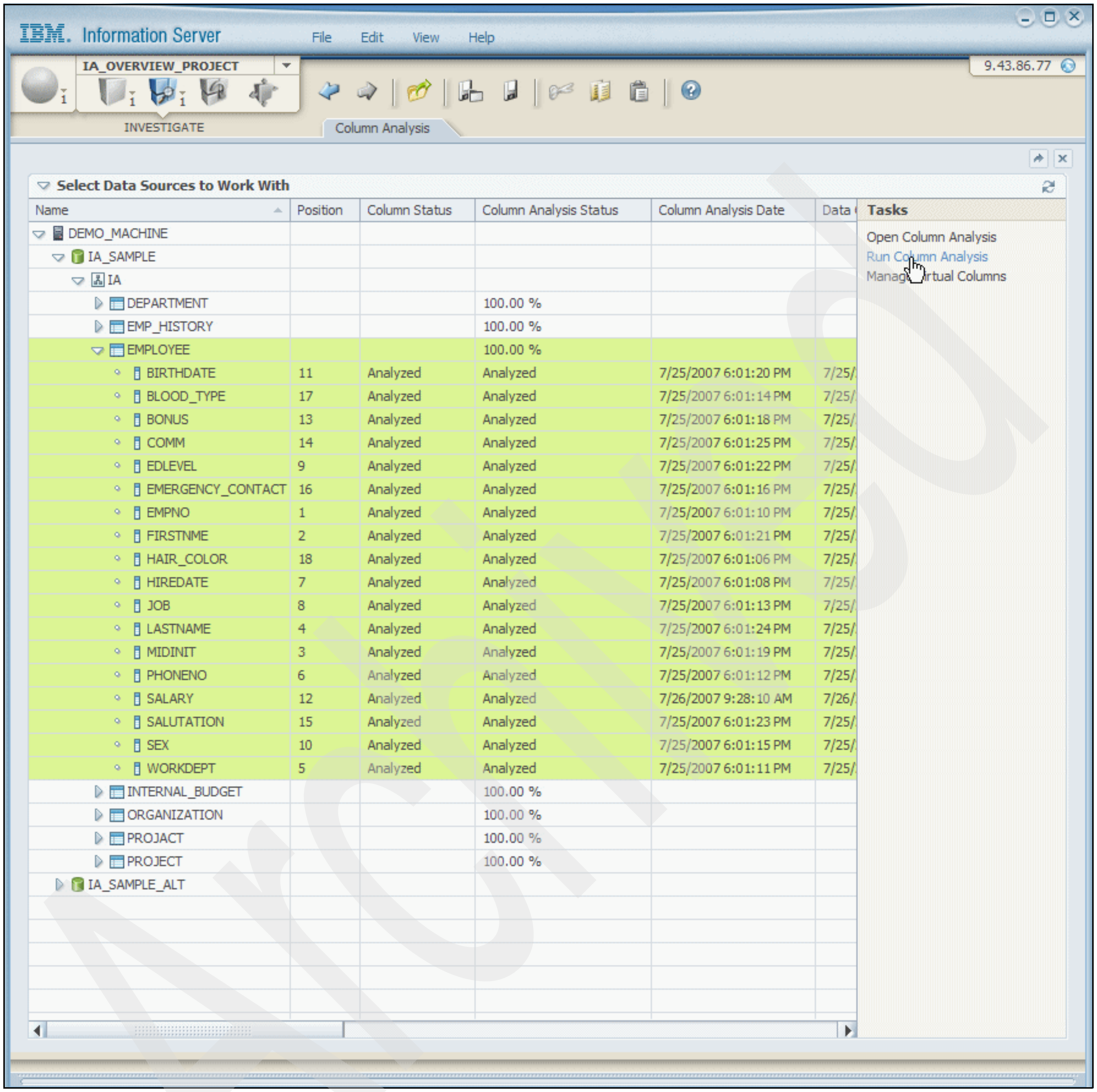
Data Value	Count	Percent

Invalid Values

Data Value	Count	Percent
Mr	10	21.74

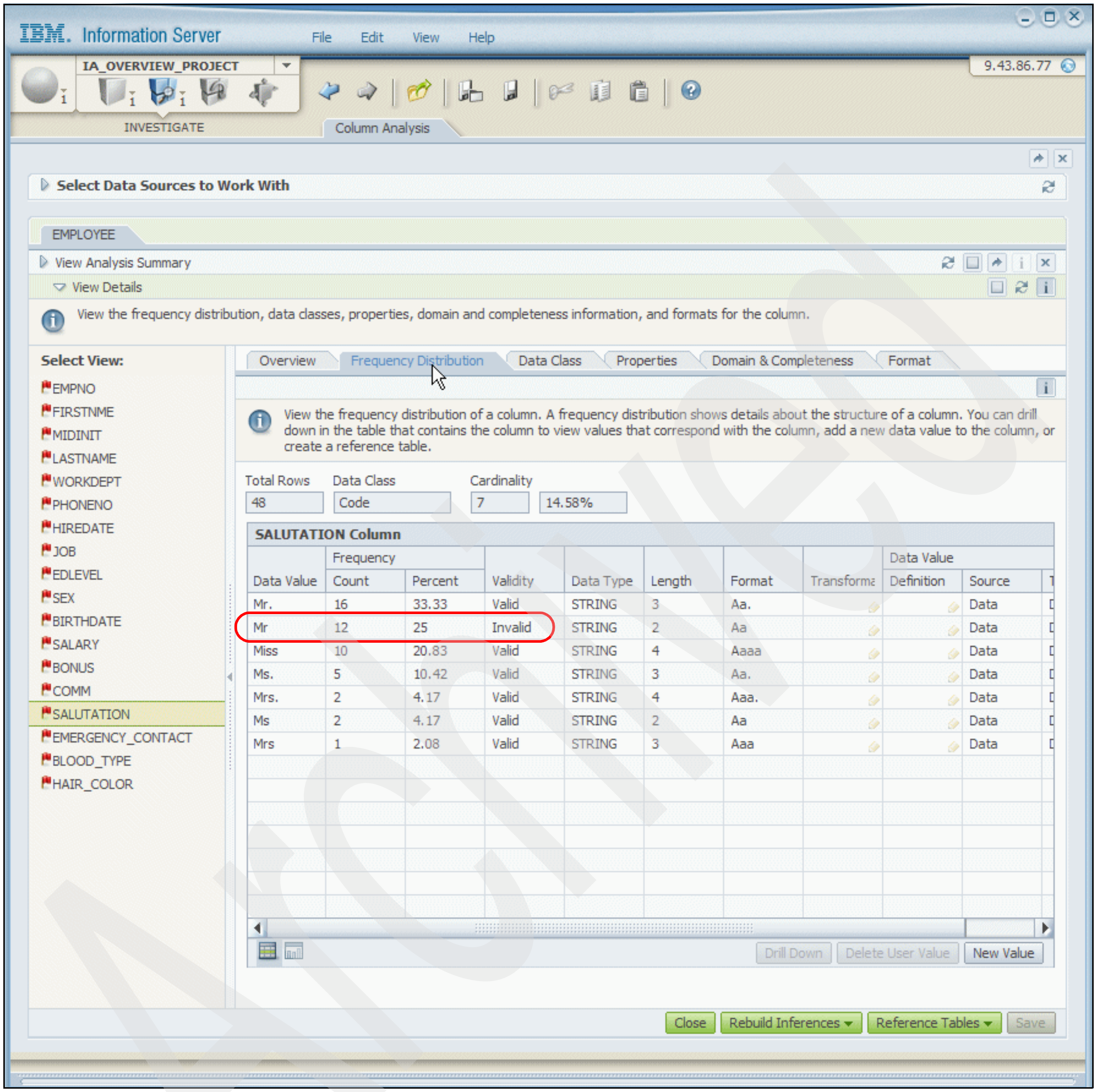
□ Reviewed

Figure 1-134 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 5/8



Name	Position	Column Status	Column Analysis Status	Column Analysis Date	Data (
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			100.00 %		
BIRTHDATE	11	Analyzed	Analyzed	7/25/2007 6:01:20 PM	7/25/
BLOOD_TYPE	17	Analyzed	Analyzed	7/25/2007 6:01:14 PM	7/25/
BONUS	13	Analyzed	Analyzed	7/25/2007 6:01:18 PM	7/25/
COMM	14	Analyzed	Analyzed	7/25/2007 6:01:25 PM	7/25/
EDLEVEL	9	Analyzed	Analyzed	7/25/2007 6:01:22 PM	7/25/
EMERGENCY_CONTACT	16	Analyzed	Analyzed	7/25/2007 6:01:16 PM	7/25/
EMPNO	1	Analyzed	Analyzed	7/25/2007 6:01:10 PM	7/25/
FIRSTNME	2	Analyzed	Analyzed	7/25/2007 6:01:21 PM	7/25/
HAIR_COLOR	18	Analyzed	Analyzed	7/25/2007 6:01:06 PM	7/25/
HIREDATE	7	Analyzed	Analyzed	7/25/2007 6:01:08 PM	7/25/
JOB	8	Analyzed	Analyzed	7/25/2007 6:01:13 PM	7/25/
LASTNAME	4	Analyzed	Analyzed	7/25/2007 6:01:24 PM	7/25/
MIDINIT	3	Analyzed	Analyzed	7/25/2007 6:01:19 PM	7/25/
PHONENO	6	Analyzed	Analyzed	7/25/2007 6:01:12 PM	7/25/
SALARY	12	Analyzed	Analyzed	7/26/2007 9:28:10 AM	7/26/
SALUTATION	15	Analyzed	Analyzed	7/25/2007 6:01:23 PM	7/25/
SEX	10	Analyzed	Analyzed	7/25/2007 6:01:15 PM	7/25/
WORKDEPT	5	Analyzed	Analyzed	7/25/2007 6:01:11 PM	7/25/
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		
PROJECT			100.00 %		
IA_SAMPLE_ALT					

Figure 1-135 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 6/8



▷ Select Data Sources to Work With

EMPLOYEE

▷ View Analysis Summary

▽ View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

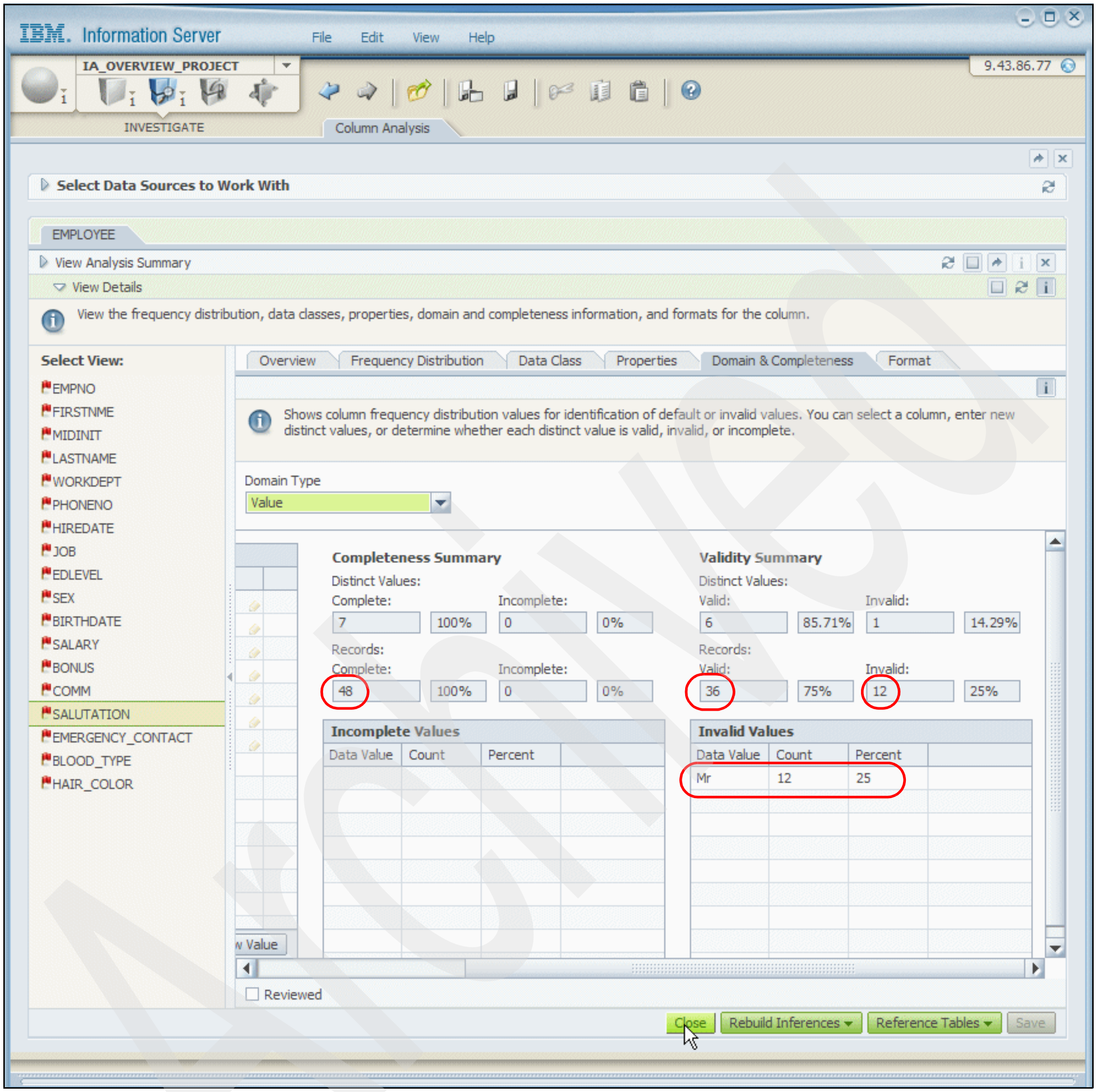
View the frequency distribution of a column. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view values that correspond with the column, add a new data value to the column, or create a reference table.

Total Rows: 48
Data Class: Code
Cardinality: 7 14.58%

SALUTATION Column

	Frequency							Data Value	
Data Value	Count	Percent	Validity	Data Type	Length	Format	Transforma	Definition	Source
Mr.	16	33.33	Valid	STRING	3	Aa.			Data
Mr	12	25	Invalid	STRING	2	Aa			Data
Miss	10	20.83	Valid	STRING	4	Aaaa			Data
Ms.	5	10.42	Valid	STRING	3	Aa.			Data
Mrs.	2	4.17	Valid	STRING	4	Aaa.			Data
Ms	2	4.17	Valid	STRING	2	Aa			Data
Mrs	1	2.08	Valid	STRING	3	Aaa			Data

Figure 1-136 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 7/8



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View:

- EMPNO
- FIRSTNME
- MIDINIT
- LASTNAME
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Completeness Summary

Distinct Values:

Complete:		Incomplete:	
7	100%	0	0%

Records:

Complete:		Incomplete:	
48	100%	0	0%

Validity Summary

Distinct Values:

Valid:		Invalid:	
6	85.71%	1	14.29%

Records:

Valid:		Invalid:	
36	75%	12	25%

Incomplete Values

Data Value	Count	Percent

Invalid Values

Data Value	Count	Percent
Mr	12	25

Figure 1-137 Set invalid values in frequency distribution SALUTATION column of EMPLOYEE table 8/8

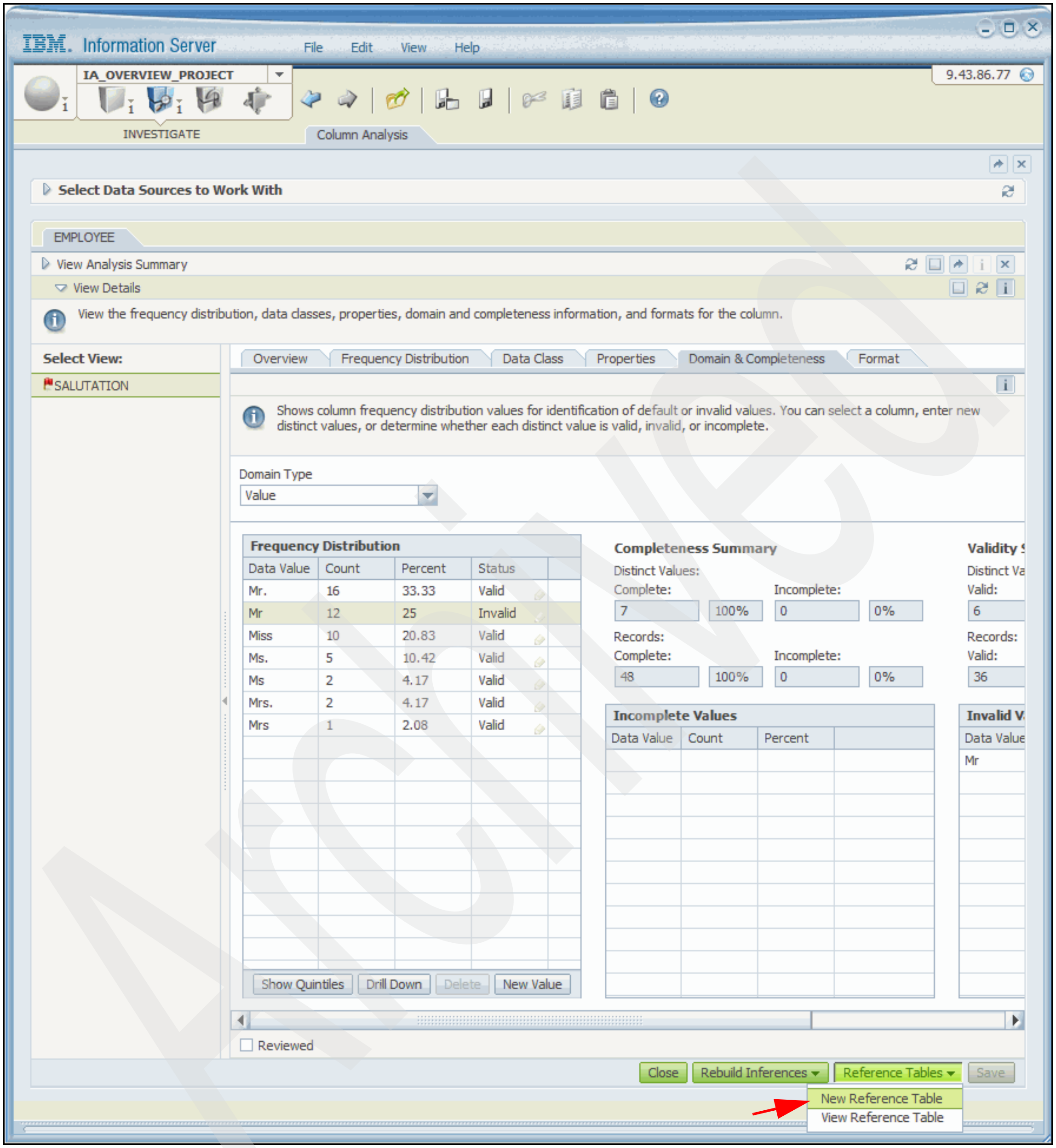
Generation of reference tables

You can create a reference table to use information in frequency distribution results outside of column analysis.

Valid, Range, Completeness, Invalid, and Mapping reference tables can be used in additional IBM Information Server capabilities or other systems to enforce domain requirements and completeness requirements or to control data conversion.

Figure 1-138 on page 193 through Figure 1-142 on page 197 describe the creation and viewing of a reference table consisting of invalid values in the SALUTATION column of the EMPLOYEE table as follows:

1. After you have run column analysis on the column and viewed and verified column analysis results, switch to the Domain & Completeness tab in the View Analysis Summary, select the *Mr* data value in the frequency distribution, and click **Reference Tables** → **New Reference Table** as shown in Figure 1-138 on page 193.
2. On the New Reference Table pane, type a Name (INVALIDSALUTATION) for the reference table, select the type of reference table that you want to create (Invalid in this case), and optionally enter a description for the reference table in the Definition field as shown in Figure 1-139 on page 194.
3. Click **Preview** as shown in Figure 1-139 on page 194 to display a preview of the reference table as shown in Figure 1-140 on page 195. Click **Save** to create the INVALIDSALUTATION reference table.
4. To view the contents of the newly created reference table, in the View Analysis Summary, select the *Mr* data value in the frequency distribution, and click **Reference Tables** → **View Reference Table** as shown in Figure 1-141 on page 196. The contents of the INVALIDSALUTATION reference table are shown in Figure 1-142 on page 197.



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Select View: SALUTATION

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
Mr.	16	33.33	Valid
Mr	12	25	Invalid
Miss	10	20.83	Valid
Ms.	5	10.42	Valid
Ms	2	4.17	Valid
Mrs.	2	4.17	Valid
Mrs	1	2.08	Valid

Completeness Summary

Distinct Values:
Complete: 7 100%
Incomplete: 0 0%

Records:
Complete: 48 100%
Incomplete: 0 0%

Validity S

Distinct Va
Valid: 6

Records:
Valid: 36

Incomplete Values

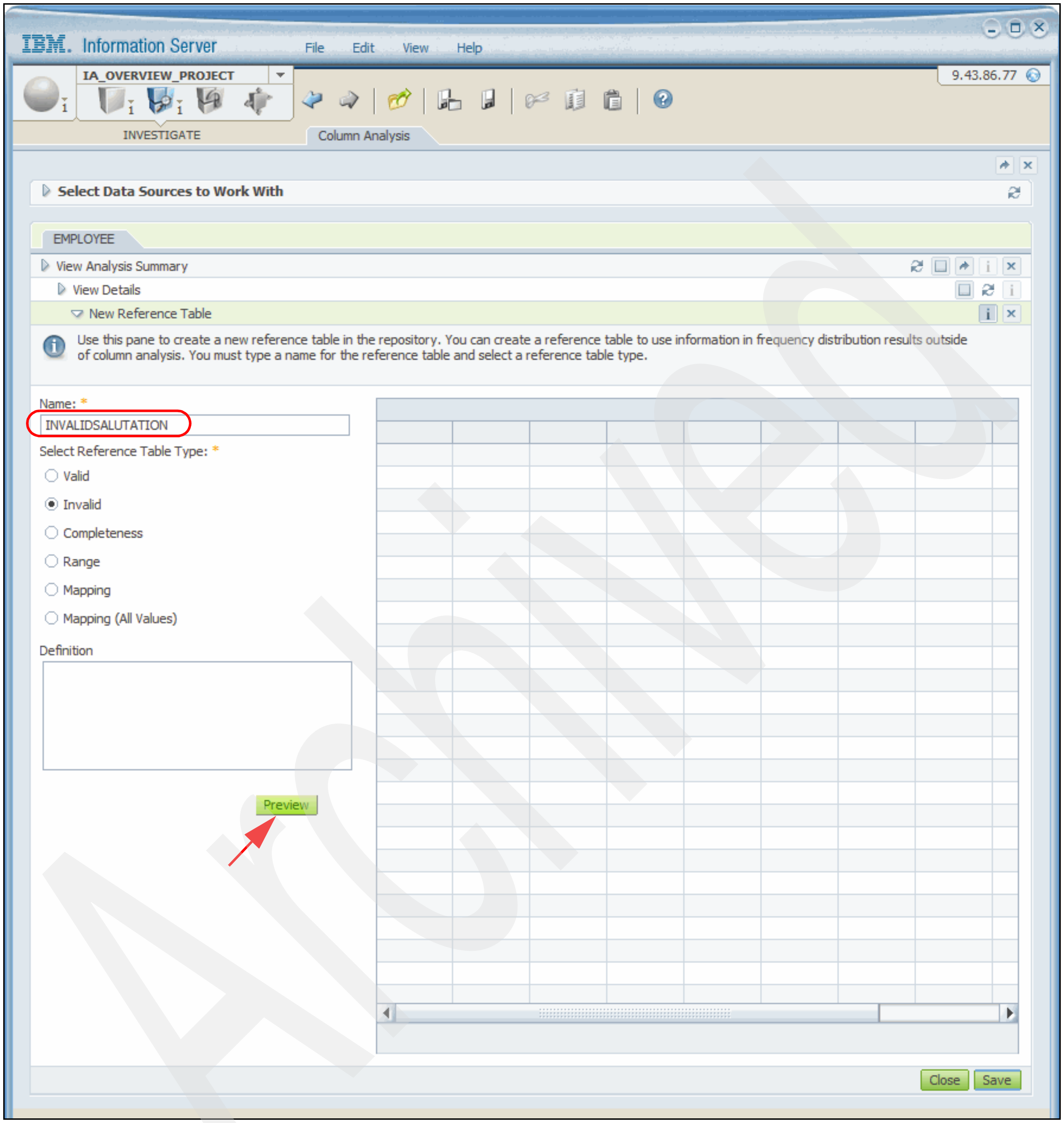
Data Value	Count	Percent

Invalid V

Data Value
Mr

□ Reviewed

Figure 1-138 Generate a reference table with invalid values 1/5



Use this pane to create a new reference table in the repository. You can create a reference table to use information in frequency distribution results outside of column analysis. You must type a name for the reference table and select a reference table type.

Name: *

INVALIDSALUTATION

Select Reference Table Type: *

○ Valid

● Invalid

○ Completeness

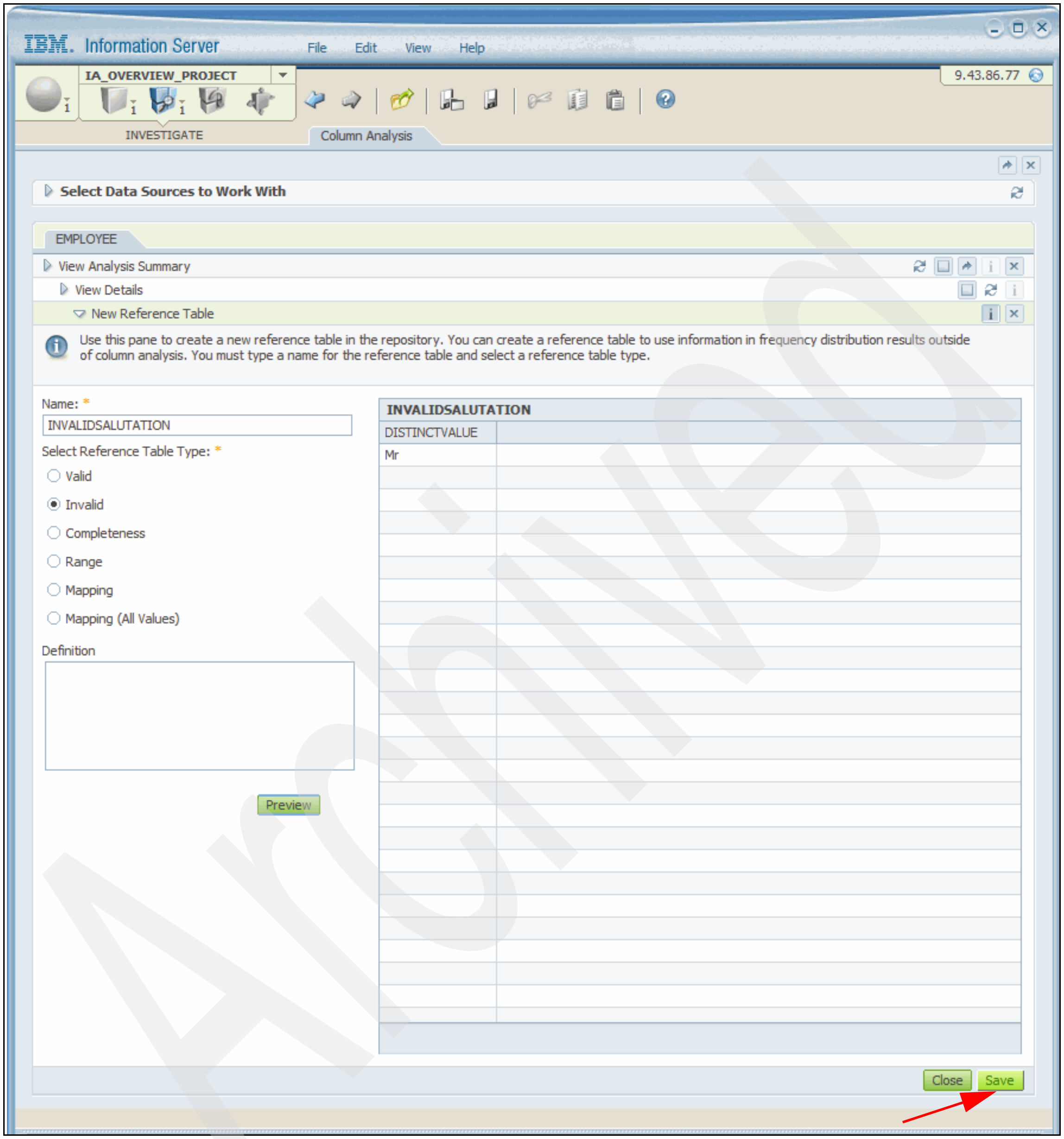
○ Range

○ Mapping

○ Mapping (All Values)

Definition

Figure 1-139   Generate a reference table with invalid values 2/5



Use this pane to create a new reference table in the repository. You can create a reference table to use information in frequency distribution results outside of column analysis. You must type a name for the reference table and select a reference table type.

Name: *
INVALIDSALUTATION

Select Reference Table Type: *

○ Valid
● Invalid
○ Completeness
○ Range
○ Mapping
○ Mapping (All Values)

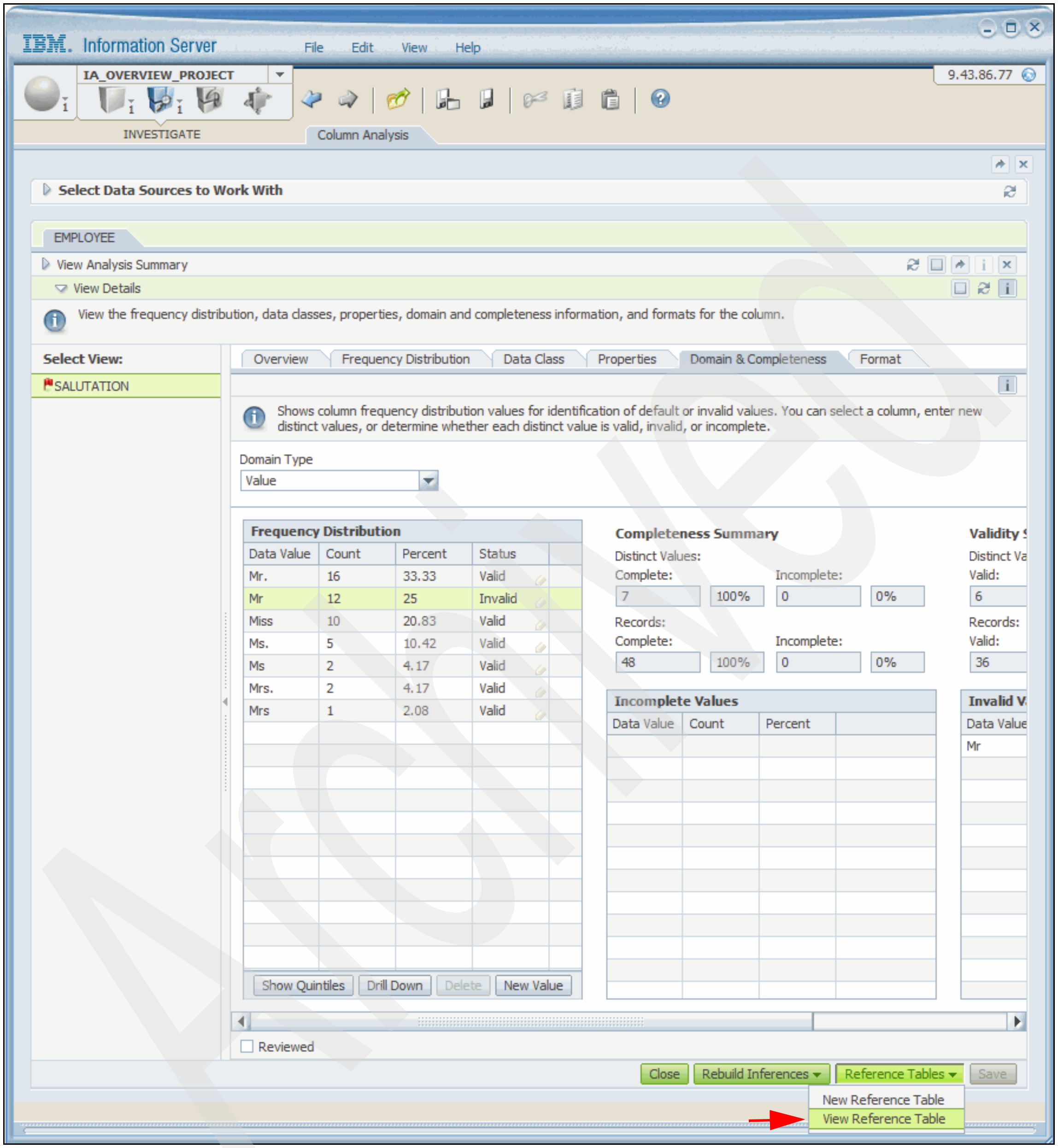
Definition

Preview

INVALIDSALUTATION
DISTINCTVALUE
Mr

Close Save

Figure 1-140 Generate a reference table with invalid values 3/5



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
Mr.	16	33.33	Valid
Mr	12	25	Invalid
Miss	10	20.83	Valid
Ms.	5	10.42	Valid
Ms	2	4.17	Valid
Mrs.	2	4.17	Valid
Mrs	1	2.08	Valid

Completeness Summary

Distinct Values:
Complete: 7 100% Incomplete: 0 0%

Records:
Complete: 48 100% Incomplete: 0 0%

Validity S

Distinct Va
Valid: 6

Records:
Valid: 36

Incomplete Values

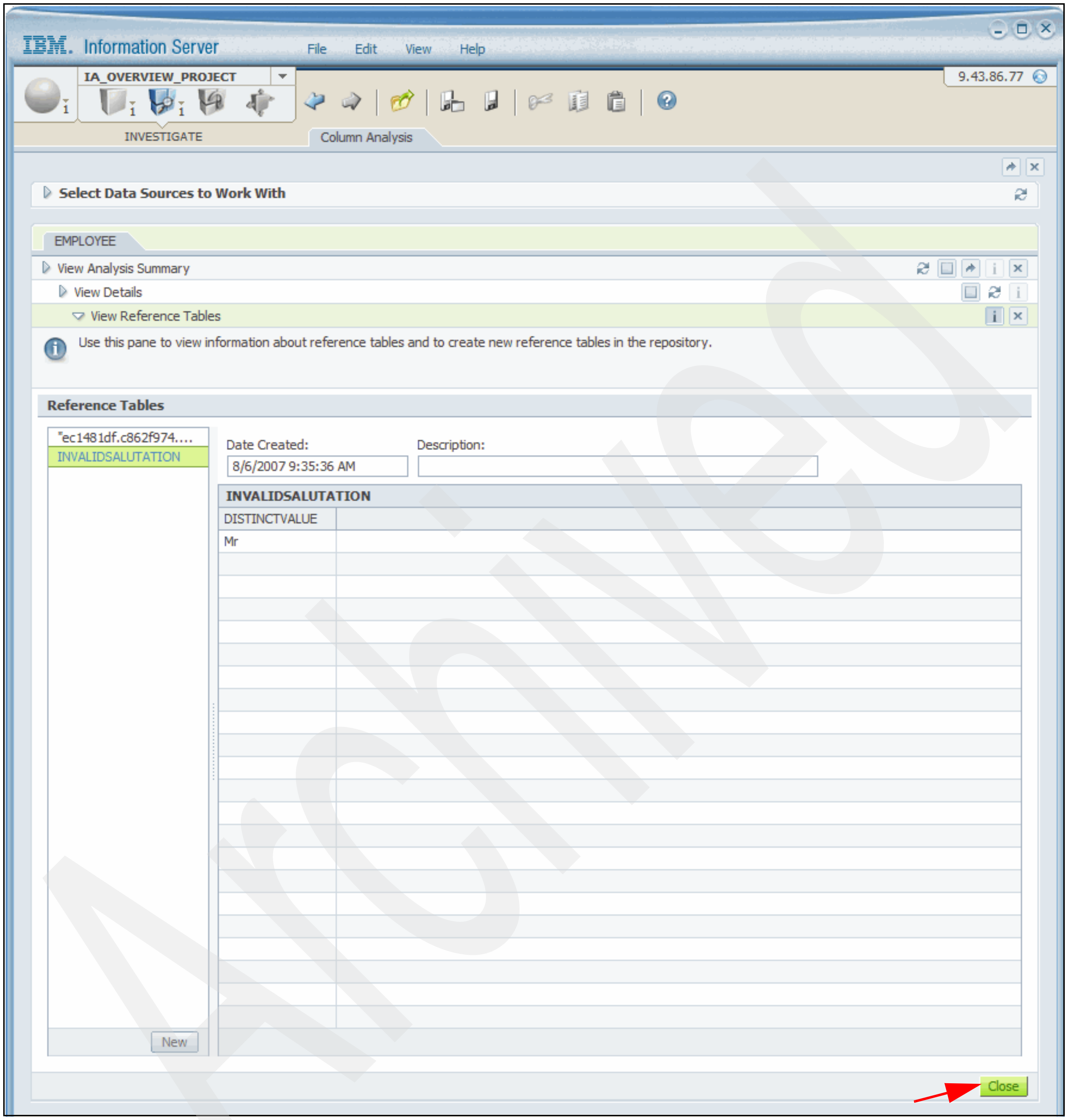
Data Value	Count	Percent

Invalid V

Data Value
Mr

☐ Reviewed

Figure 1-141 Generate a reference table with invalid values 4/5



Use this pane to view information about reference tables and to create new reference tables in the repository.

Date Created: 8/6/2007 9:35:36 AM

Description:

INVALIDSALUTATION
DISTINCTVALUE
Mr

Figure 1-142   Generate a reference table with invalid values 5/5

Validating data using reference tables

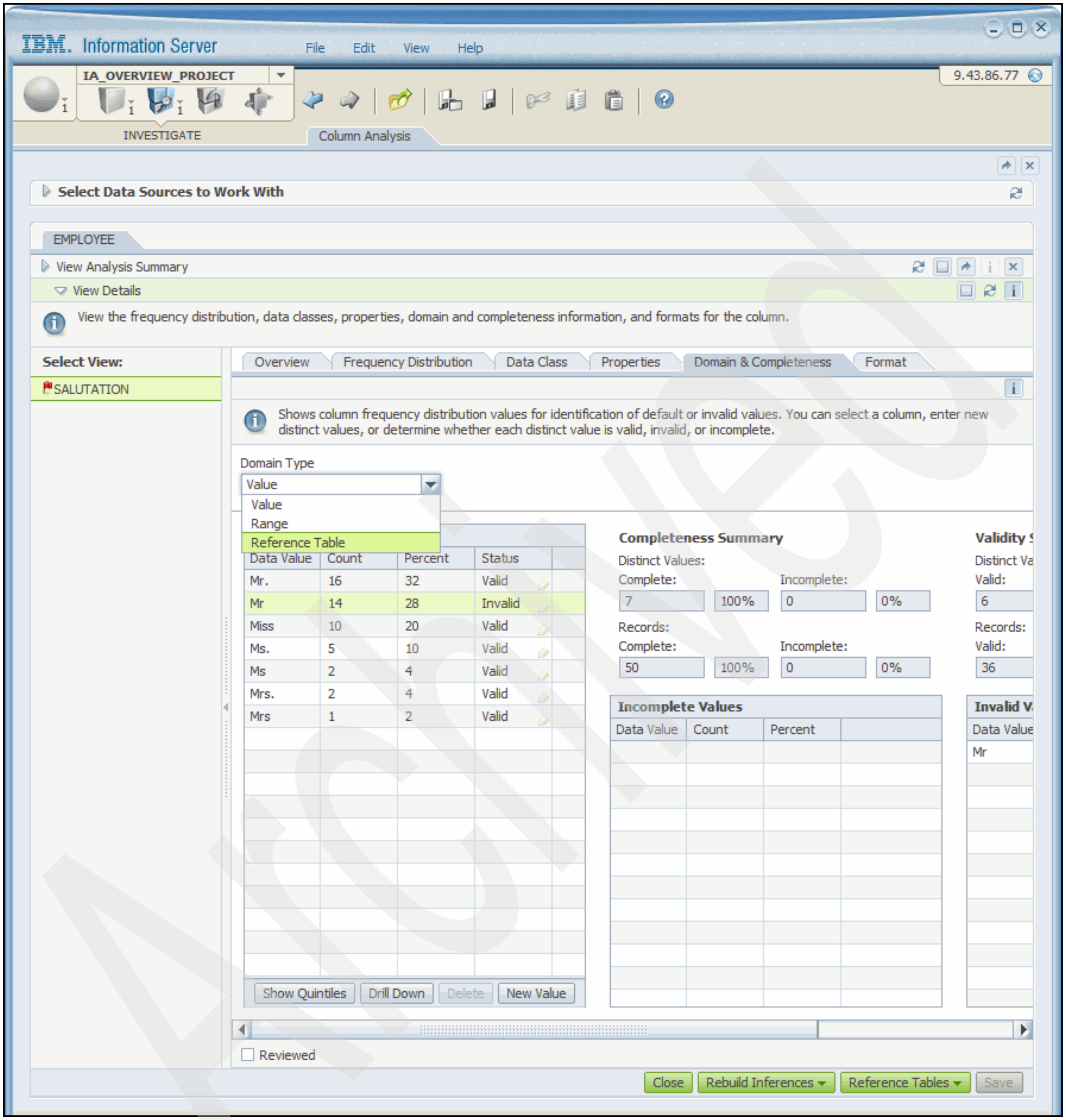
When you are reviewing the results of a column analysis job, you can compare the data values to a reference table that contains valid values. Any data values that are not found in the reference table are marked as invalid. Correspondingly, any data values found in an invalid values reference table are marked as invalid. After running a column analysis job, you might compare a column that contains city abbreviations for your orders to a reference table that contains a list of all known city abbreviations.

Figure 1-143 on page 199 through Figure 1-150 on page 204 show how to validate data values using a reference table as follows:

1. In the View Details pane, under the Domain & Completeness tab, select **Reference Table** in the Domain Type menu as shown in Figure 1-143 on page 199.
2. In the verification window, click **OK** to verify that you want to change the domain type as shown in Figure 1-144 on page 200.
3. In the Table Type field, select the type of reference table that you want to use, which is INVALID_TABLE in our case, as shown in Figure 1-145 on page 200.
4. In the Table Name field, select the reference table that you want to use from the drop-down list, which is INVALIDSALUTATION in our case, as shown in Figure 1-146 on page 201.
5. The data values are compared and invalid values are specified in the Frequency Distribution table as shown in Figure 1-147 on page 201.

> **Note:** There is no change in the results here because the invalid state for the data value *Mr* was already set during column analysis. Had we had another value in the INVALIDSALUTATION reference table, such as *Ms*, then we would have seen that data value marked as invalid in Figure 1-148 on page 202.

6. You can optionally click **Drill Down** on the Frequency Distribution table as shown in Figure 1-148 on page 202 to show all the instances of that data value in the analyzed column as shown in Figure 1-149 on page 203. You can also click **Show Quintiles** to view the results (Figure 1-150 on page 204).



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

- Value
- Range
- Reference Table

Data Value	Count	Percent	Status
Mr.	16	32	Valid
Mr	14	28	Invalid
Miss	10	20	Valid
Ms.	5	10	Valid
Ms	2	4	Valid
Mrs.	2	4	Valid
Mrs	1	2	Valid

Completeness Summary

Distinct Values:
Complete: 7 100%
Incomplete: 0 0%

Records:
Complete: 50 100%
Incomplete: 0 0%

Validity S

Distinct Va
Valid: 6

Records:
Valid: 36

Incomplete Values

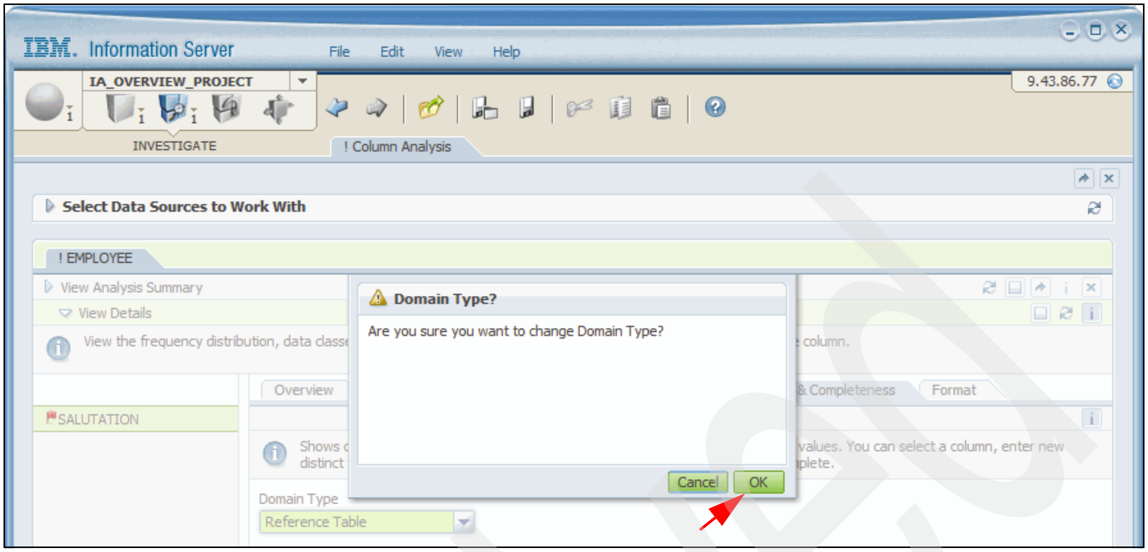
Data Value	Count	Percent

Invalid V

Data Value
Mr

□ Reviewed

Figure 1-143 Validate data using reference table 1/8

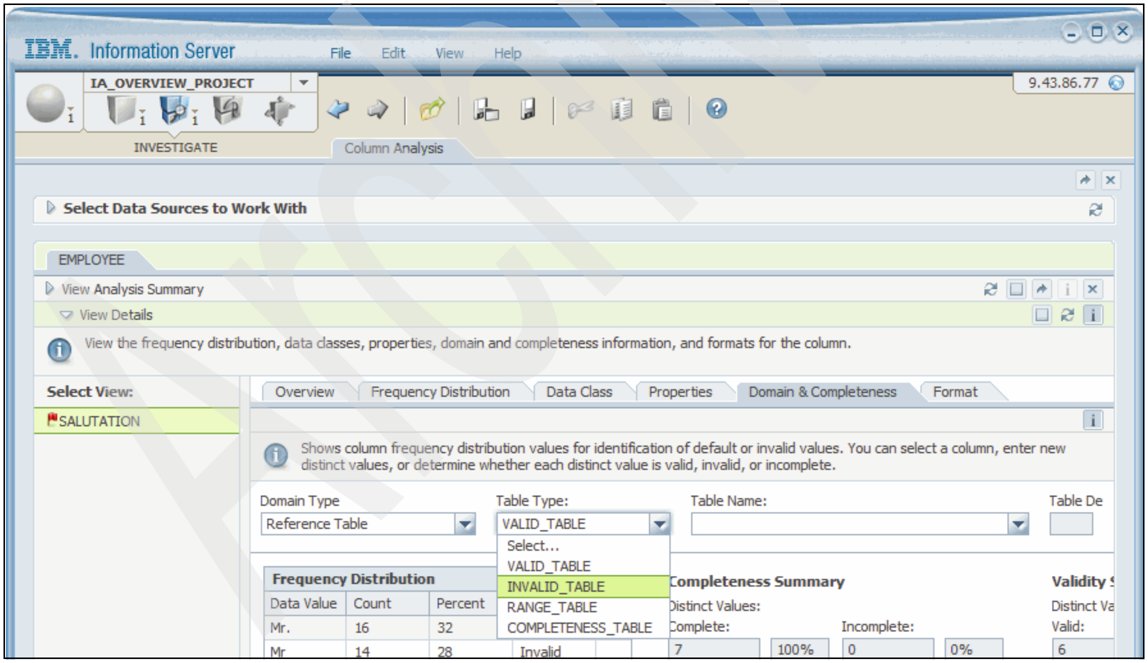


Domain Type?

Are you sure you want to change Domain Type?

Domain Type: Reference Table

Figure 1-144   Validate data using reference table 2/8



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Reference Table

Table Type: VALID_TABLE

Table Name:

Table De

Frequency Distribution

Data Value	Count	Percent	
Mr.	16	32	
Mr	14	28	Invalid

Completeness Summary

Distinct Values: 7

Complete: 100%

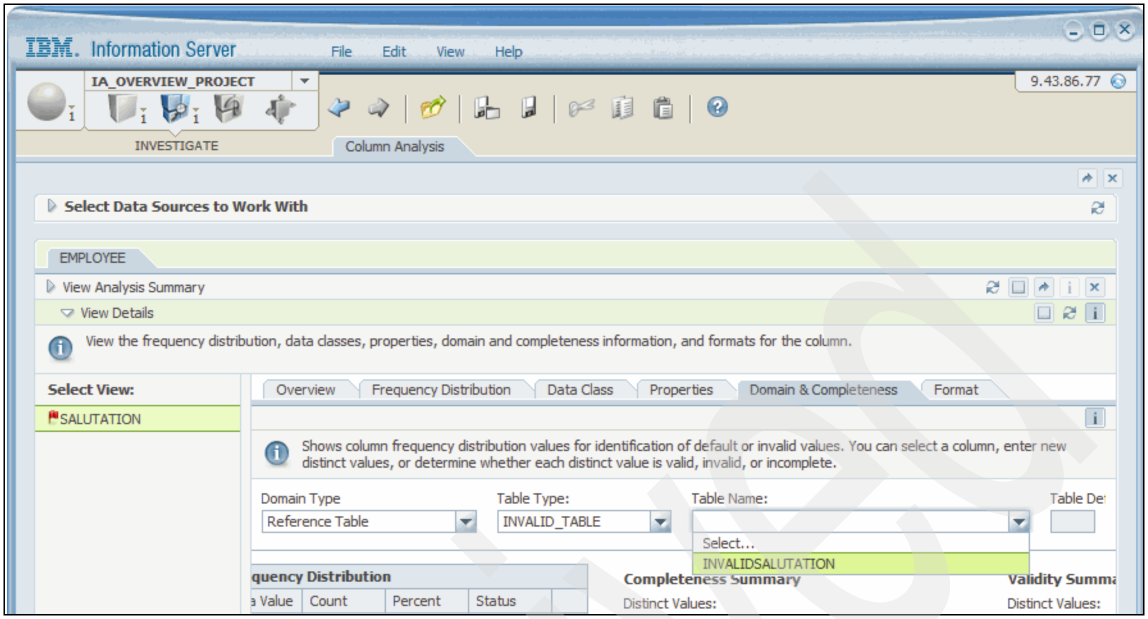
Incomplete: 0 0%

Validity S

Distinct Va

Valid: 6

Figure 1-145   Validate data using reference table 3/8



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

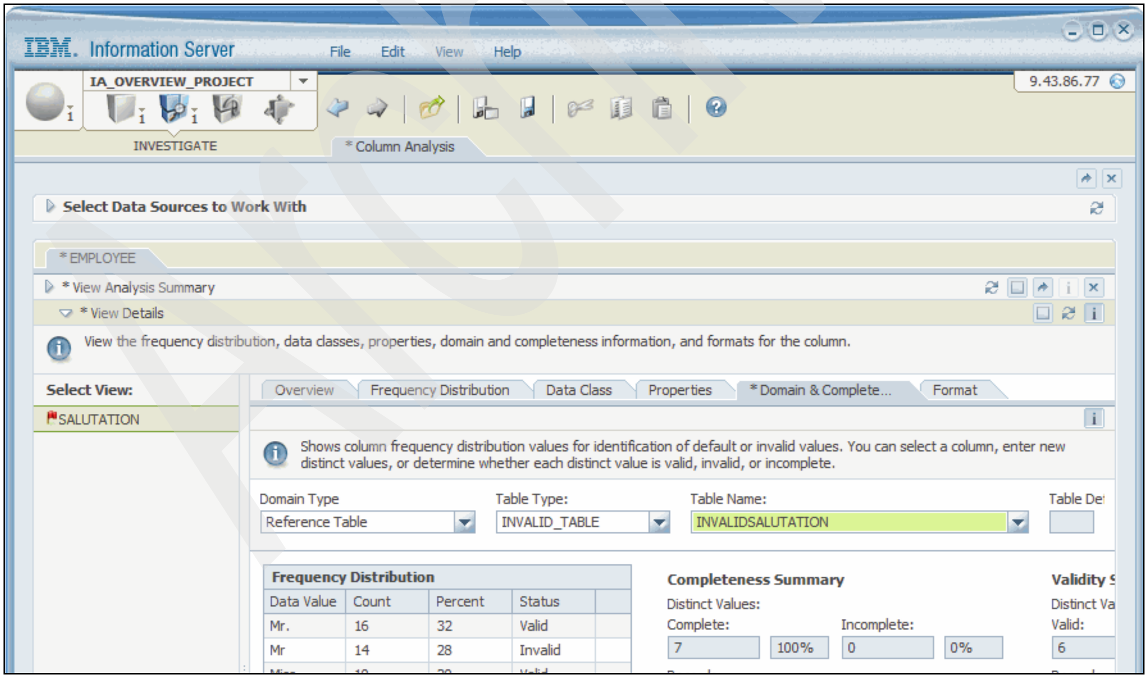
Domain Type: Reference Table

Table Type: INVALID_TABLE

Table Name: Select... / INVALIDSALUTATION

Table De

Figure 1-146   Validate data using reference table 4/8



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Reference Table

Table Type: INVALID_TABLE

Table Name: INVALIDSALUTATION

Table De

Frequency Distribution

Data Value	Count	Percent	Status
Mr.	16	32	Valid
Mr	14	28	Invalid

Completeness Summary

Distinct Values:

Complete: 7 100%

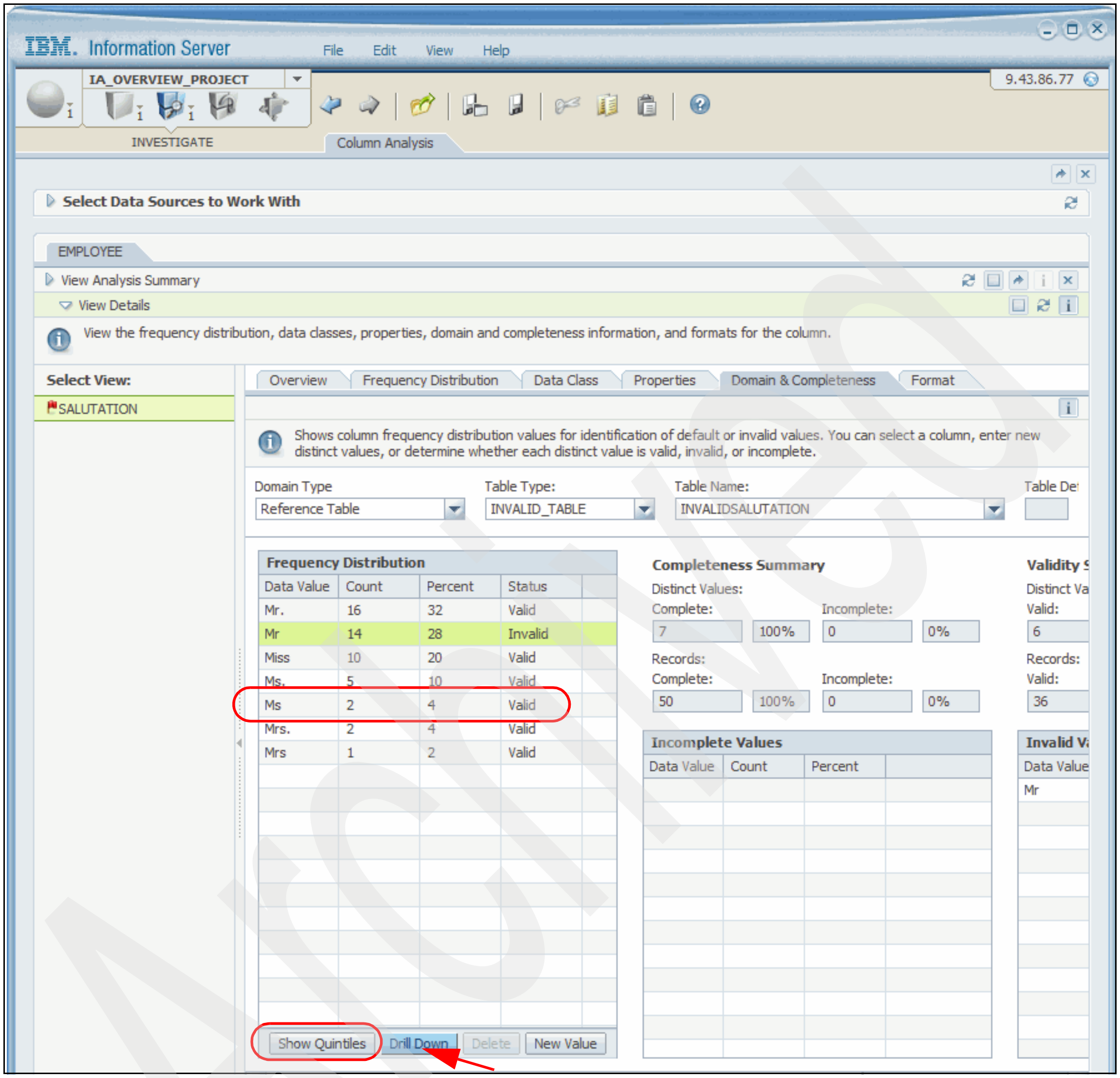
Incomplete: 0 0%

Validity S

Distinct Va

Valid: 6

Figure 1-147   Validate data using reference table 5/8



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Reference Table

Table Type: INVALID_TABLE

Table Name: INVALIDSALUTATION

Table De

Frequency Distribution

Data Value	Count	Percent	Status
Mr.	16	32	Valid
Mr	14	28	Invalid
Miss	10	20	Valid
Ms.	5	10	Valid
Ms	2	4	Valid
Mrs.	2	4	Valid
Mrs	1	2	Valid

Completeness Summary

Distinct Values:

Complete: 7 100%

Incomplete: 0 0%

Records:

Complete: 50 100%

Incomplete: 0 0%

Incomplete Values

Data Value	Count	Percent

Validity S

Distinct Va

Valid: 6

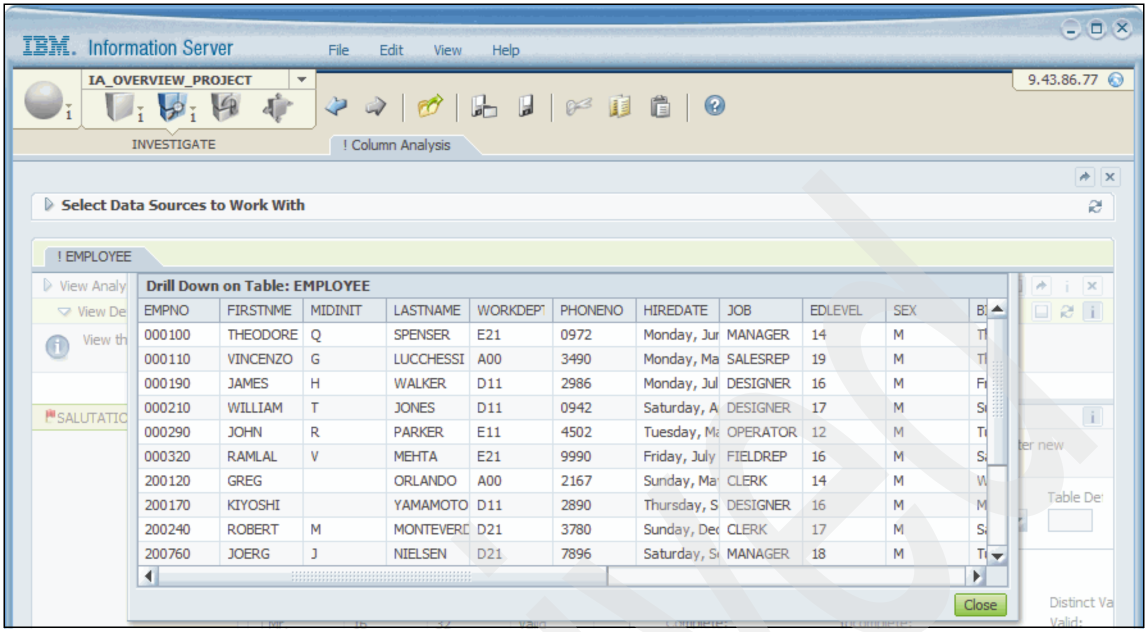
Records:

Valid: 36

Invalid V

Data Value
Mr

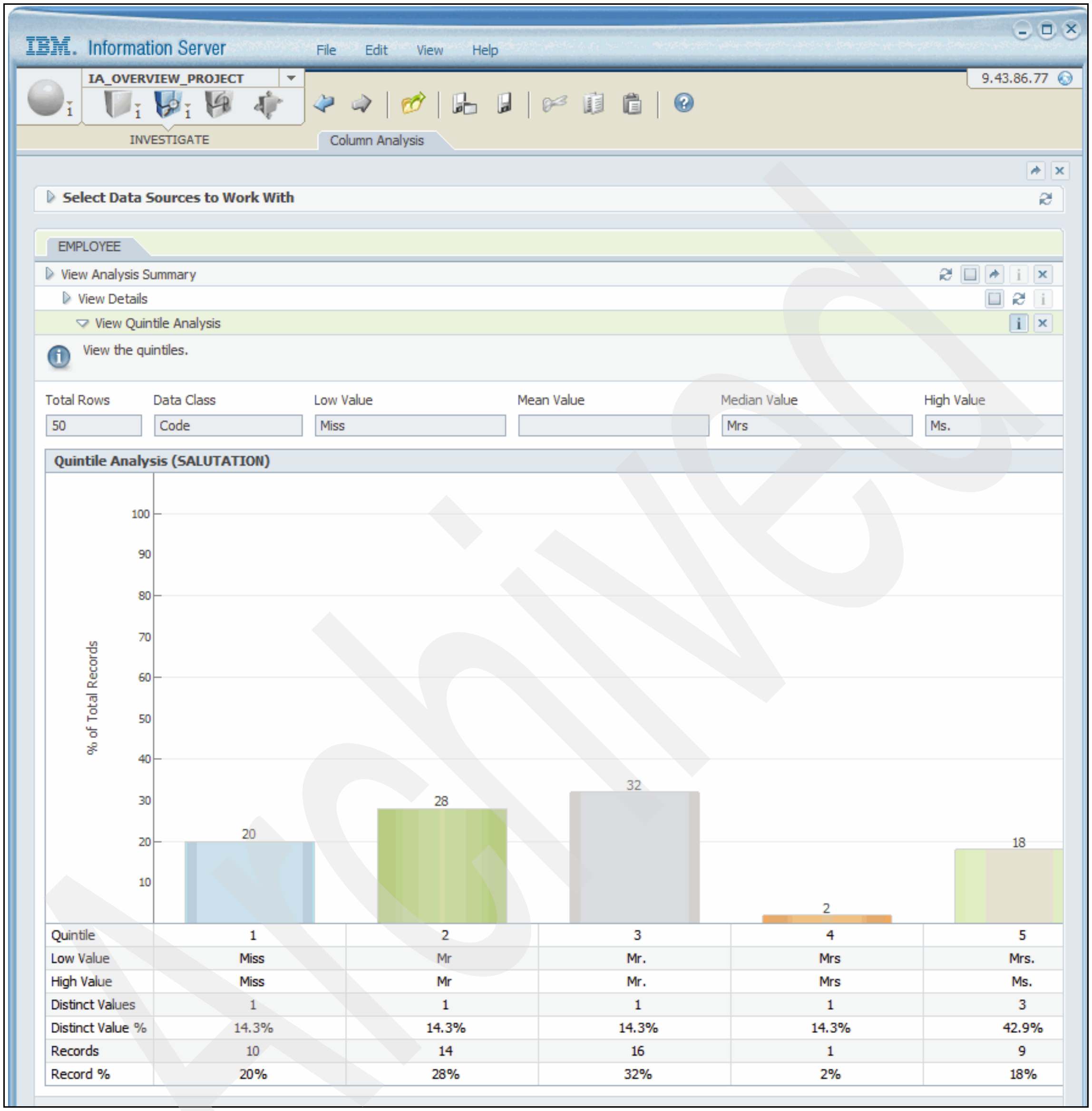
Figure 1-148 Validate data using reference table 6/8



Drill Down on Table: EMPLOYEE

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BI
000100	THEODORE	Q	SPENSER	E21	0972	Monday, Jur	MANAGER	14	M	Tl
000110	VINCENZO	G	LUCCHESSI	A00	3490	Monday, Ma	SALESREP	19	M	Tl
000190	JAMES	H	WALKER	D11	2986	Monday, Jul	DESIGNER	16	M	Fı
000210	WILLIAM	T	JONES	D11	0942	Saturday, A	DESIGNER	17	M	Sı
000290	JOHN	R	PARKER	E11	4502	Tuesday, Ma	OPERATOR	12	M	Tı
000320	RAMLAL	V	MEHTA	E21	9990	Friday, July	FIELDREP	16	M	Sa
200120	GREG		ORLANDO	A00	2167	Sunday, Ma	CLERK	14	M	W
200170	KIYOSHI		YAMAMOTO	D11	2890	Thursday, S	DESIGNER	16	M	M
200240	ROBERT	M	MONTEVERD	D21	3780	Sunday, Dec	CLERK	17	M	Sa
200760	JOERG	J	NIELSEN	D21	7896	Saturday, S	MANAGER	18	M	Tı

Figure 1-149   Validate data using reference table 7/8



Select Data Sources to Work With

View the quintiles.

Total Rows	Data Class	Low Value	Mean Value	Median Value	High Value
50	Code	Miss		Mrs	Ms.

Quintile Analysis (SALUTATION)

Quintile	1	2	3	4	5
Low Value	Miss	Mr	Mr.	Mrs	Mrs.
High Value	Miss	Mr	Mr.	Mrs	Ms.
Distinct Values	1	1	1	1	3
Distinct Value %	14.3%	14.3%	14.3%	14.3%	42.9%
Records	10	14	16	1	9
Record %	20%	28%	32%	2%	18%

Figure 1-150 Validate data using reference table 8/8

Manage virtual columns in the EMPLOYEE table

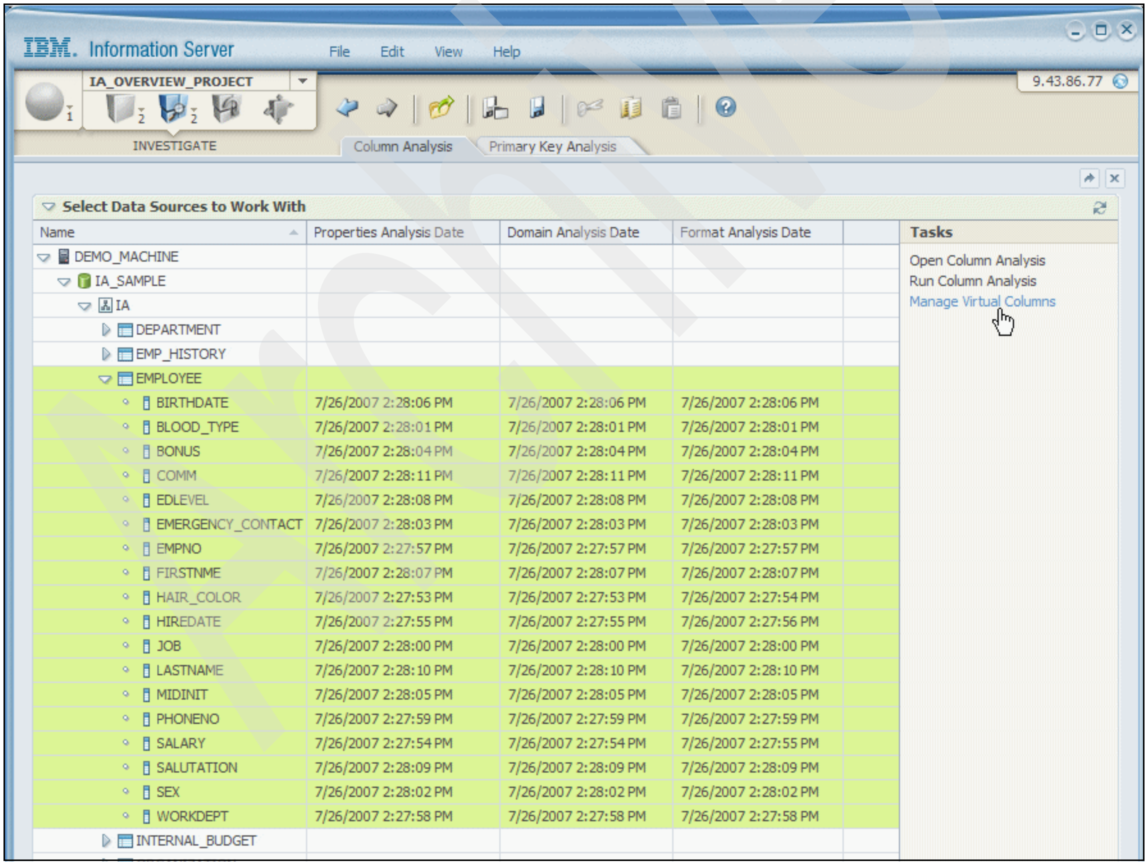
A virtual column is the concatenation of data from one or more columns into one column. A virtual column might also be used to assess potential multi-column primary keys or validate a value pair combination from several fields. In addition, you can create a virtual column on a single column to truncate or pad the data in that column.

You can then run subsequent analysis jobs on this concatenated column as shown in Figure 1-151 on page 206 through Figure 1-166 on page 221 as follows:

1. Select all the columns of the EMPLOYEE table in the Select Data Sources to Work With workspace and click **Manage Virtual Columns** as shown in Figure 1-151 on page 206.
2. Click **Create View** in Manage Virtual Columns to create a virtual column as shown in Figure 1-152 on page 207.
3. Figure 1-153 on page 208 shows the creation of a virtual column named FULLNAME, that is a concatenation of the FIRSTNME and LASTNAME columns. The concatenated length of the virtual column is 28. The intervening screens involving selecting columns from the available columns to create the virtual column are not shown here. Click **Save**. Then click **Close** as shown in Figure 1-154 on page 209.
4. Figure 1-155 on page 210 shows the newly created virtual column FULLNAME in the EMPLOYEE table. The icon that represents a virtual column is highlighted.
5. Select the virtual column FULLNAME, and submit the job as shown in Figure 1-156 on page 211. Then click **View Results** to view the View Analysis Summary as shown in Figure 1-157 on page 212.
6. Select **FULLNAME** in the View Analysis Summary and click **View Details** as shown in Figure 1-158 on page 213 to view details of the virtual column.
7. Figure 1-159 on page 214 shows the frequency distribution of the concatenated column, and the Data Class of Text. Cardinality is 48.
8. Figure 1-160 on page 215 shows the inferred Data Class of Text, and the corresponding default selection of Text under the Data Class tab.
9. Figure 1-161 on page 216 through Figure 1-164 on page 219 show the properties of the FULLNAME column under the Properties tab.
   - The defined and inferred data type is string, and the selected data type is string. All the column values have strings as indicate in the pie-chart.
   - There is no defined length, but the inferred length is 19. The minimum length is 8, the median is 13, and the maximum is 19. The bar chart in Figure 1-162 on page 217 shows the occurrence of each length.

- Precision and scale does not apply to this data type.
- Nullability is not defined for this virtual column nor is it inferred because all values are present as shown in Figure 1-163 on page 218.
- The cardinality type definition is not constrained. However, it is inferred to be unique because the values in the virtual column are all distinct and at 100%, which exceeds the Unique Threshold of 99%. The default selection is Unique.

10. The Domain & Completeness tab in Figure 1-164 on page 219 shows the frequency distribution, completeness, and validity details. All data values are considered valid.
11. Figure 1-165 on page 220 shows the 32 formats.
12. Figure 1-166 on page 221 highlights the successful completion of column analysis including property analysis, domain analysis, and format analysis for the virtual column FULLNAME.



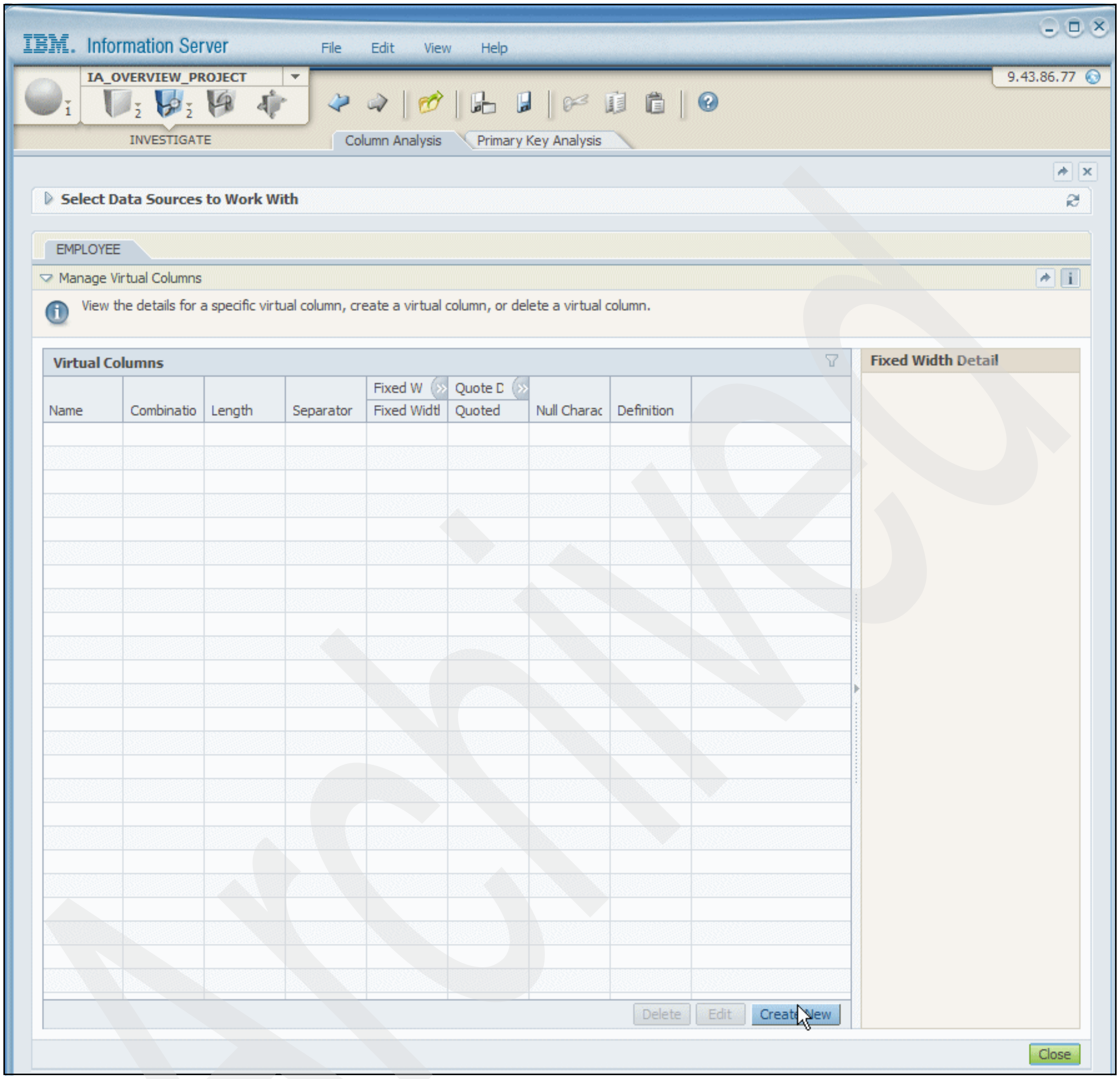
Select Data Sources to Work With

Name	Properties Analysis Date	Domain Analysis Date	Format Analysis Date
DEMO_MACHINE			
IA_SAMPLE			
IA			
DEPARTMENT			
EMP_HISTORY			
EMPLOYEE			
BIRTHDATE	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM
BLOOD_TYPE	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM
BONUS	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM
COMM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM
EDLEVEL	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM
EMERGENCY_CONTACT	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM
EMPNO	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM
FIRSTNME	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM
HAIR_COLOR	7/26/2007 2:27:53 PM	7/26/2007 2:27:53 PM	7/26/2007 2:27:54 PM
HIREDATE	7/26/2007 2:27:55 PM	7/26/2007 2:27:55 PM	7/26/2007 2:27:56 PM
JOB	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM
LASTNAME	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM
MIDINIT	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM
PHONENO	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM
SALARY	7/26/2007 2:27:54 PM	7/26/2007 2:27:54 PM	7/26/2007 2:27:55 PM
SALUTATION	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM
SEX	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM
WORKDEPT	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM
INTERNAL_BUDGET			

Tasks

Open Column Analysis
Run Column Analysis
Manage Virtual Columns

Figure 1-151 Manage virtual columns in the EMPLOYEE table 1/16

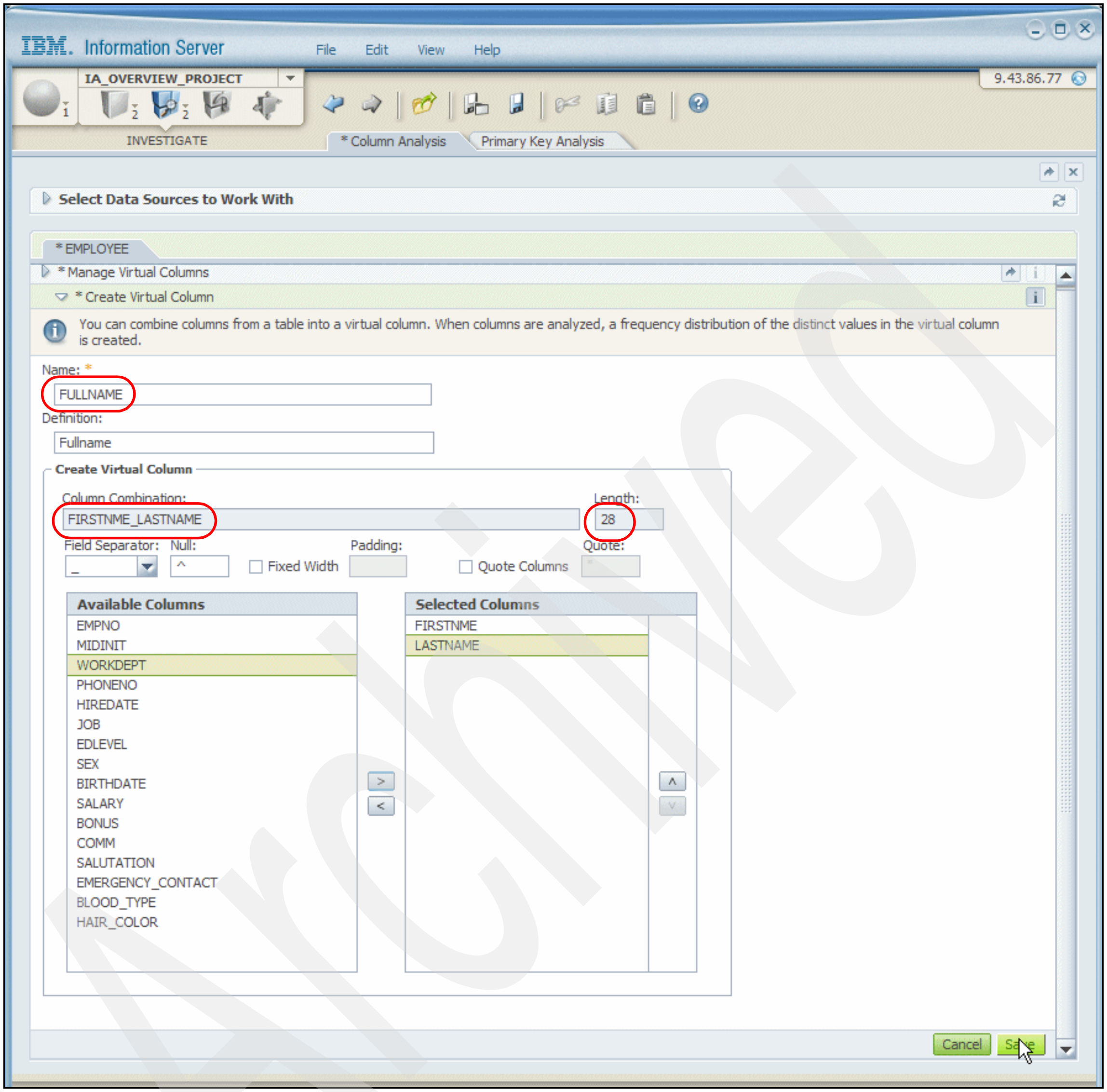


View the details for a specific virtual column, create a virtual column, or delete a virtual column.

Virtual Columns

				Fixed W	Quote D		
Name	Combinatio	Length	Separator	Fixed Widtl	Quoted	Null Charac	Definition

Figure 1-152 Manage virtual columns in the EMPLOYEE table 2/16



You can combine columns from a table into a virtual column. When columns are analyzed, a frequency distribution of the distinct values in the virtual column is created.

Name: * FULLNAME

Definition: Fullname

Create Virtual Column

Column Combination: FIRSTNME_LASTNAME

Length: 28

Field Separator: _

Null: ^

□ Fixed Width

Padding:

□ Quote Columns

Quote:

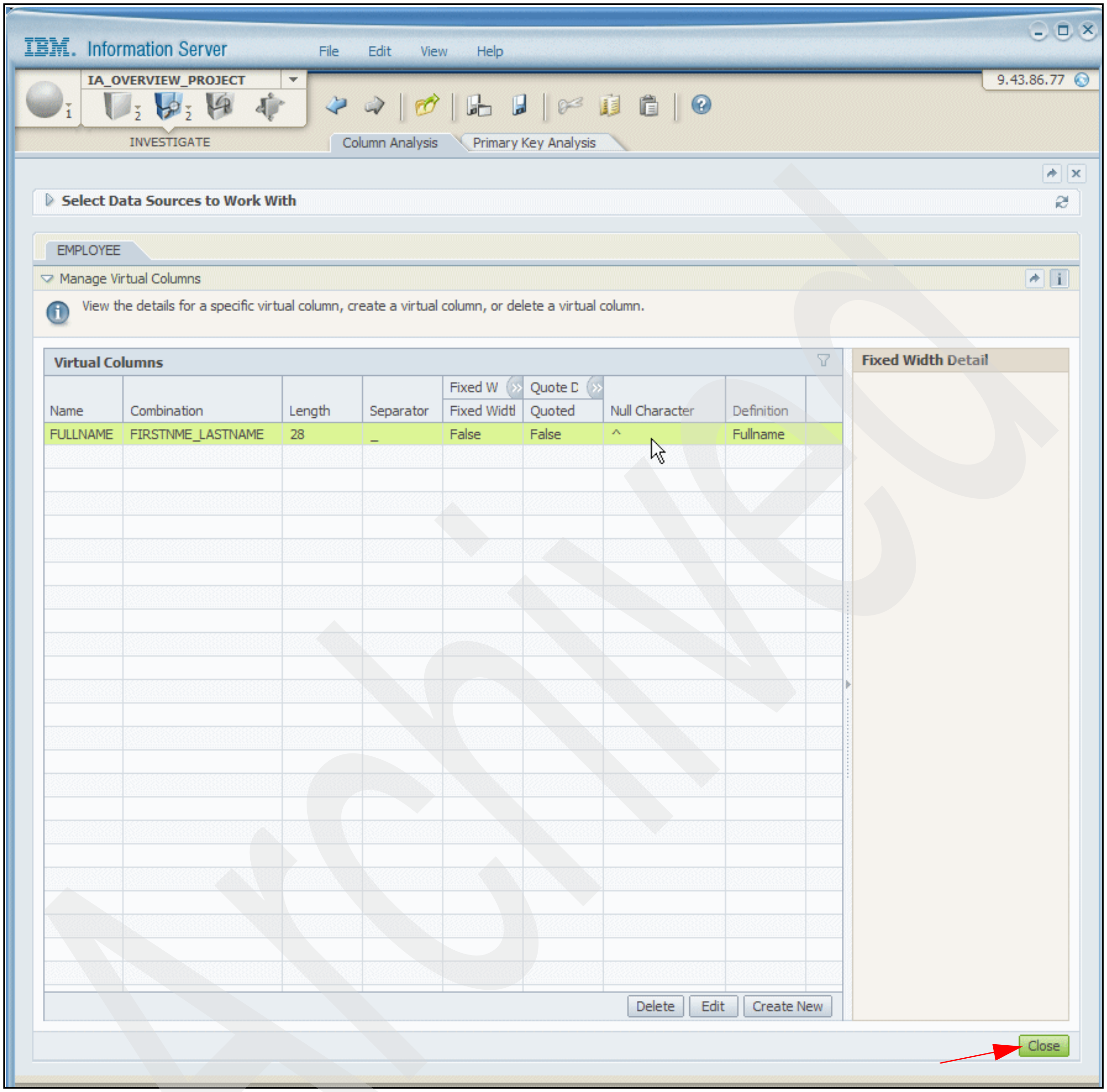
Available Columns

- EMPNO
- MIDINIT
- WORKDEPT
- PHONENO
- HIREDATE
- JOB
- EDLEVEL
- SEX
- BIRTHDATE
- SALARY
- BONUS
- COMM
- SALUTATION
- EMERGENCY_CONTACT
- BLOOD_TYPE
- HAIR_COLOR

Selected Columns

- FIRSTNME
- LASTNAME

Figure 1-153 Manage virtual columns in the EMPLOYEE table 3/16

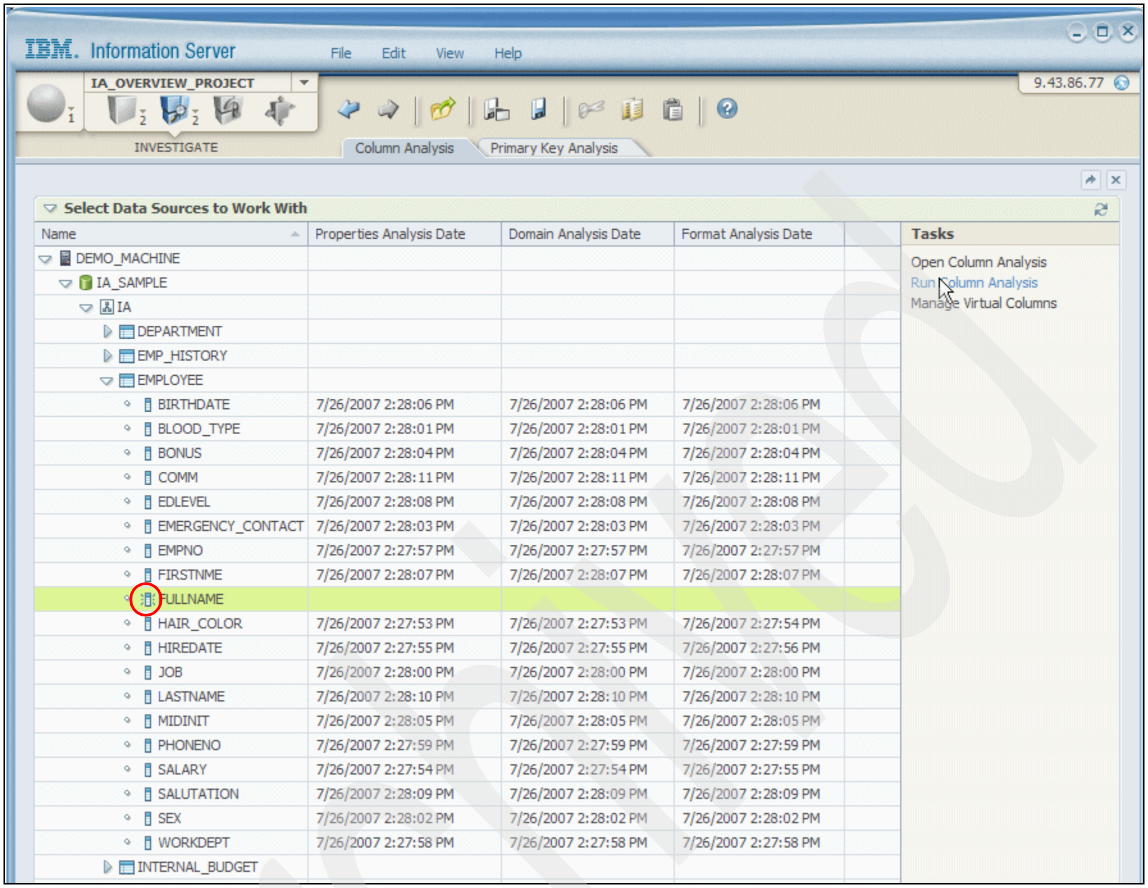


View the details for a specific virtual column, create a virtual column, or delete a virtual column.

Virtual Columns

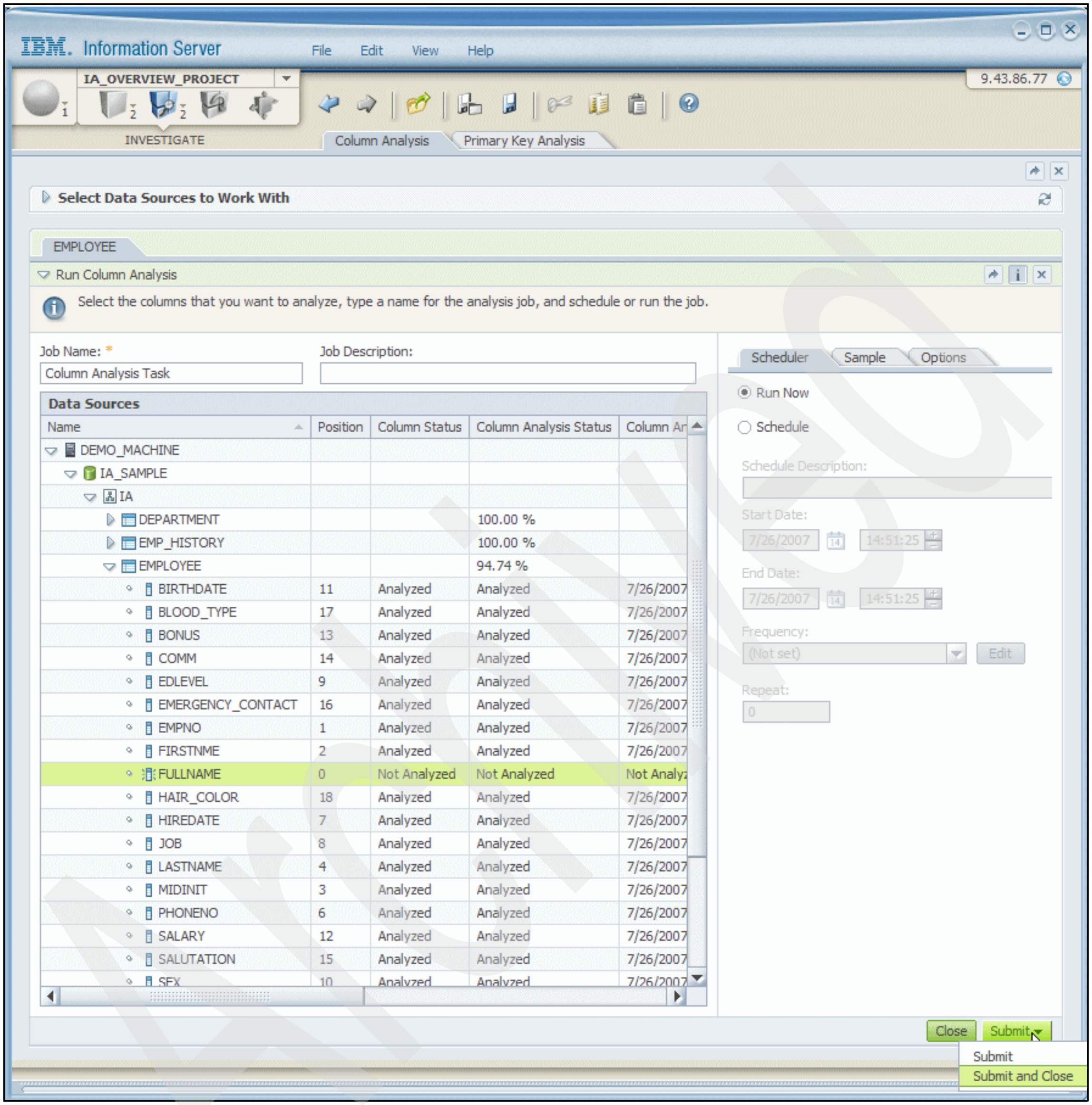
				Fixed W	Quote D		
Name	Combination	Length	Separator	Fixed Widtl	Quoted	Null Character	Definition
FULLNAME	FIRSTNME_LASTNAME	28	_	False	False	^	Fullname

Figure 1-154 Manage virtual columns in the EMPLOYEE table 4/16



Name	Properties Analysis Date	Domain Analysis Date	Format Analysis Date
DEMO_MACHINE			
IA_SAMPLE			
IA			
DEPARTMENT			
EMP_HISTORY			
EMPLOYEE			
BIRTHDATE	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM
BLOOD_TYPE	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM
BONUS	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM
COMM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM
EDLEVEL	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM
EMERGENCY_CONTACT	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM
EMPNO	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM
FIRSTNME	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM
FULLNAME			
HAIR_COLOR	7/26/2007 2:27:53 PM	7/26/2007 2:27:53 PM	7/26/2007 2:27:54 PM
HIREDATE	7/26/2007 2:27:55 PM	7/26/2007 2:27:55 PM	7/26/2007 2:27:56 PM
JOB	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM
LASTNAME	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM
MIDINIT	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM
PHONENO	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM
SALARY	7/26/2007 2:27:54 PM	7/26/2007 2:27:54 PM	7/26/2007 2:27:55 PM
SALUTATION	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM
SEX	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM
WORKDEPT	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM
INTERNAL_BUDGET			

Figure 1-155 Manage virtual columns in the EMPLOYEE table 5/16



Select Data Sources to Work With

EMPLOYEE

Run Column Analysis

Select the columns that you want to analyze, type a name for the analysis job, and schedule or run the job.

Job Name: * Column Analysis Task

Job Description:

Data Sources

Name	Position	Column Status	Column Analysis Status	Column An
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT			100.00 %	
EMP_HISTORY			100.00 %	
EMPLOYEE			94.74 %	
BIRTHDATE	11	Analyzed	Analyzed	7/26/2007
BLOOD_TYPE	17	Analyzed	Analyzed	7/26/2007
BONUS	13	Analyzed	Analyzed	7/26/2007
COMM	14	Analyzed	Analyzed	7/26/2007
EDLEVEL	9	Analyzed	Analyzed	7/26/2007
EMERGENCY_CONTACT	16	Analyzed	Analyzed	7/26/2007
EMPNO	1	Analyzed	Analyzed	7/26/2007
FIRSTNME	2	Analyzed	Analyzed	7/26/2007
FULLNAME	0	Not Analyzed	Not Analyzed	Not Analyz
HAIR_COLOR	18	Analyzed	Analyzed	7/26/2007
HIREDATE	7	Analyzed	Analyzed	7/26/2007
JOB	8	Analyzed	Analyzed	7/26/2007
LASTNAME	4	Analyzed	Analyzed	7/26/2007
MIDINIT	3	Analyzed	Analyzed	7/26/2007
PHONENO	6	Analyzed	Analyzed	7/26/2007
SALARY	12	Analyzed	Analyzed	7/26/2007
SALUTATION	15	Analyzed	Analyzed	7/26/2007
SEX	10	Analyzed	Analyzed	7/26/2007

● Run Now

○ Schedule

Schedule Description:

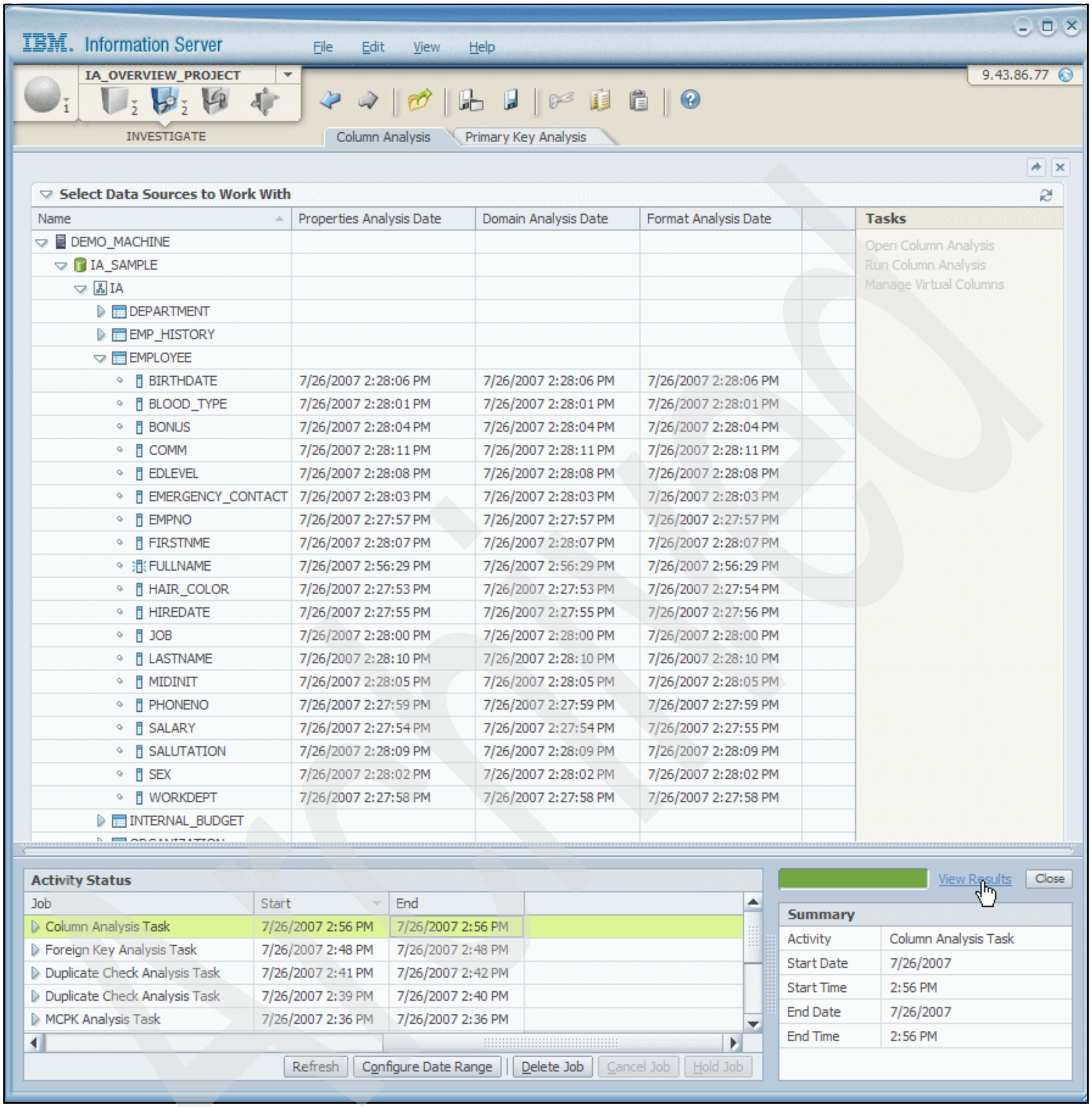
Start Date: 7/26/2007 14:51:25

End Date: 7/26/2007 14:51:25

Frequency: (Not set)

Repeat: 0

Figure 1-156 Manage virtual columns in the EMPLOYEE table 6/16



Select Data Sources to Work With

Name	Properties Analysis Date	Domain Analysis Date	Format Analysis Date
DEMO_MACHINE			
IA_SAMPLE			
IA			
DEPARTMENT			
EMP_HISTORY			
EMPLOYEE			
BIRTHDATE	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM
BLOOD_TYPE	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM
BONUS	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM
COMM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM
EDLEVEL	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM
EMERGENCY_CONTACT	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM
EMPNO	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM
FIRSTNME	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM
FULLNAME	7/26/2007 2:56:29 PM	7/26/2007 2:56:29 PM	7/26/2007 2:56:29 PM
HAIR_COLOR	7/26/2007 2:27:53 PM	7/26/2007 2:27:53 PM	7/26/2007 2:27:54 PM
HIREDATE	7/26/2007 2:27:55 PM	7/26/2007 2:27:55 PM	7/26/2007 2:27:56 PM
JOB	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM
LASTNAME	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM
MIDINIT	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM
PHONENO	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM
SALARY	7/26/2007 2:27:54 PM	7/26/2007 2:27:54 PM	7/26/2007 2:27:55 PM
SALUTATION	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM
SEX	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM
WORKDEPT	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM
INTERNAL_BUDGET			

Tasks

Open Column Analysis
Run Column Analysis
Manage Virtual Columns

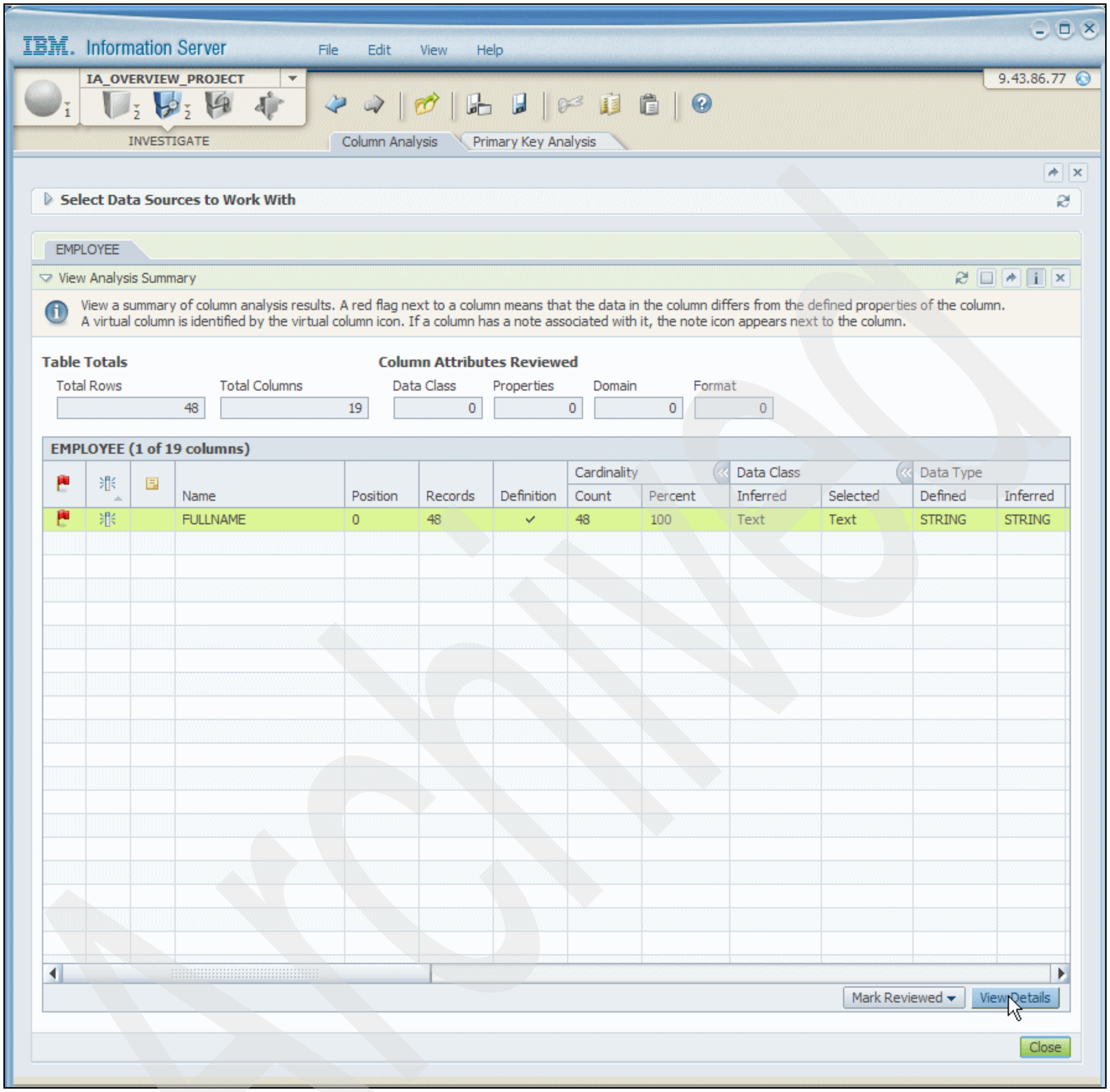
Activity Status

Job	Start	End
Column Analysis Task	7/26/2007 2:56 PM	7/26/2007 2:56 PM
Foreign Key Analysis Task	7/26/2007 2:48 PM	7/26/2007 2:48 PM
Duplicate Check Analysis Task	7/26/2007 2:41 PM	7/26/2007 2:42 PM
Duplicate Check Analysis Task	7/26/2007 2:39 PM	7/26/2007 2:40 PM
MCPK Analysis Task	7/26/2007 2:36 PM	7/26/2007 2:36 PM

Summary

Activity	Column Analysis Task
Start Date	7/26/2007
Start Time	2:56 PM
End Date	7/26/2007
End Time	2:56 PM

Figure 1-157 Manage virtual columns in the EMPLOYEE table 7/16



View a summary of column analysis results. A red flag next to a column means that the data in the column differs from the defined properties of the column. A virtual column is identified by the virtual column icon. If a column has a note associated with it, the note icon appears next to the column.

Table Totals

Total Rows: 48
Total Columns: 19

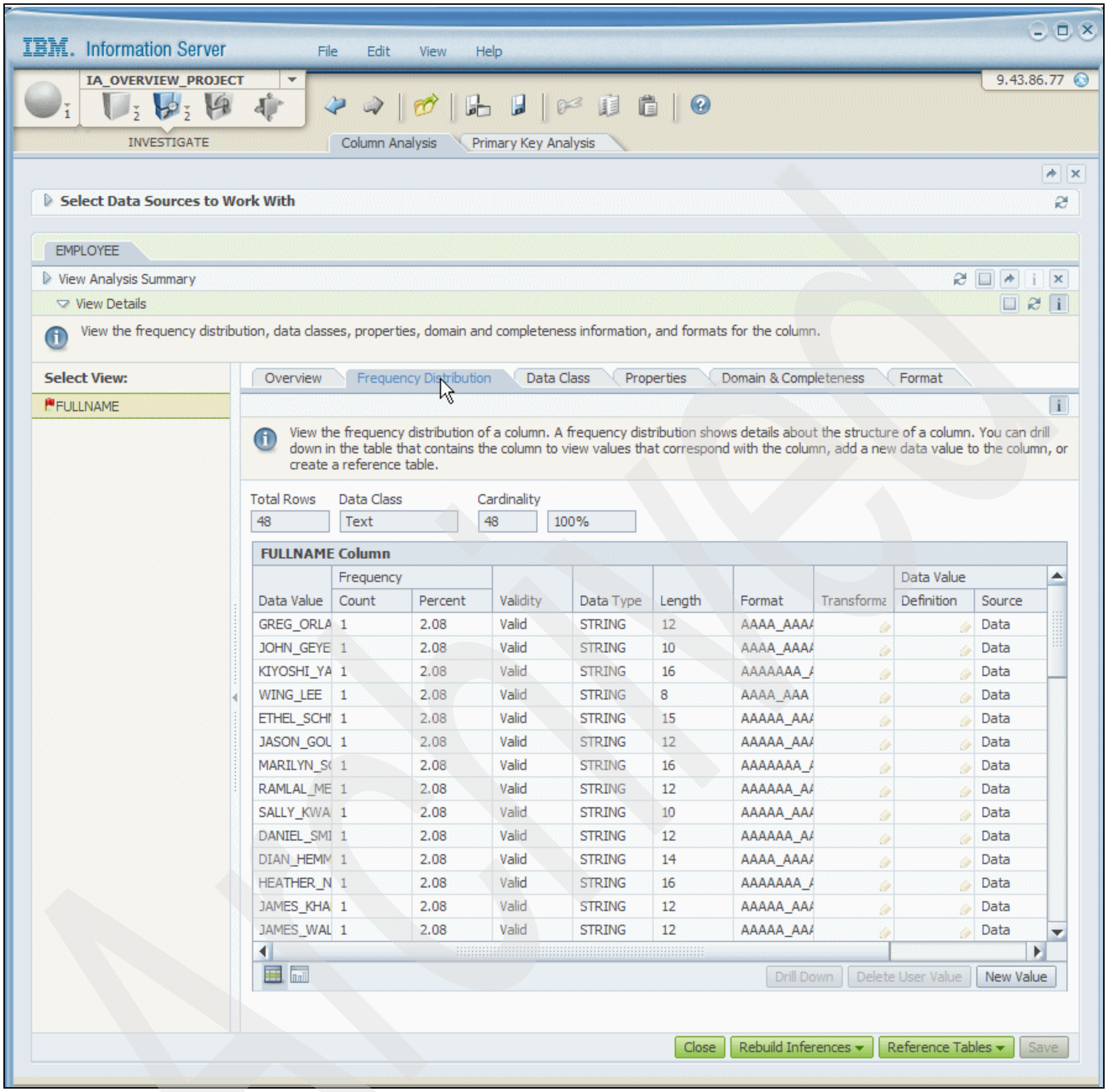
Column Attributes Reviewed

Data Class: 0
Properties: 0
Domain: 0
Format: 0

EMPLOYEE (1 of 19 columns)

							Cardinality		Data Class		Data Type	
🚩			Name	Position	Records	Definition	Count	Percent	Inferred	Selected	Defined	Inferred
🚩			FULLNAME	0	48	✓	48	100	Text	Text	STRING	STRING

Figure 1-158 Manage virtual columns in the EMPLOYEE table 8/16



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

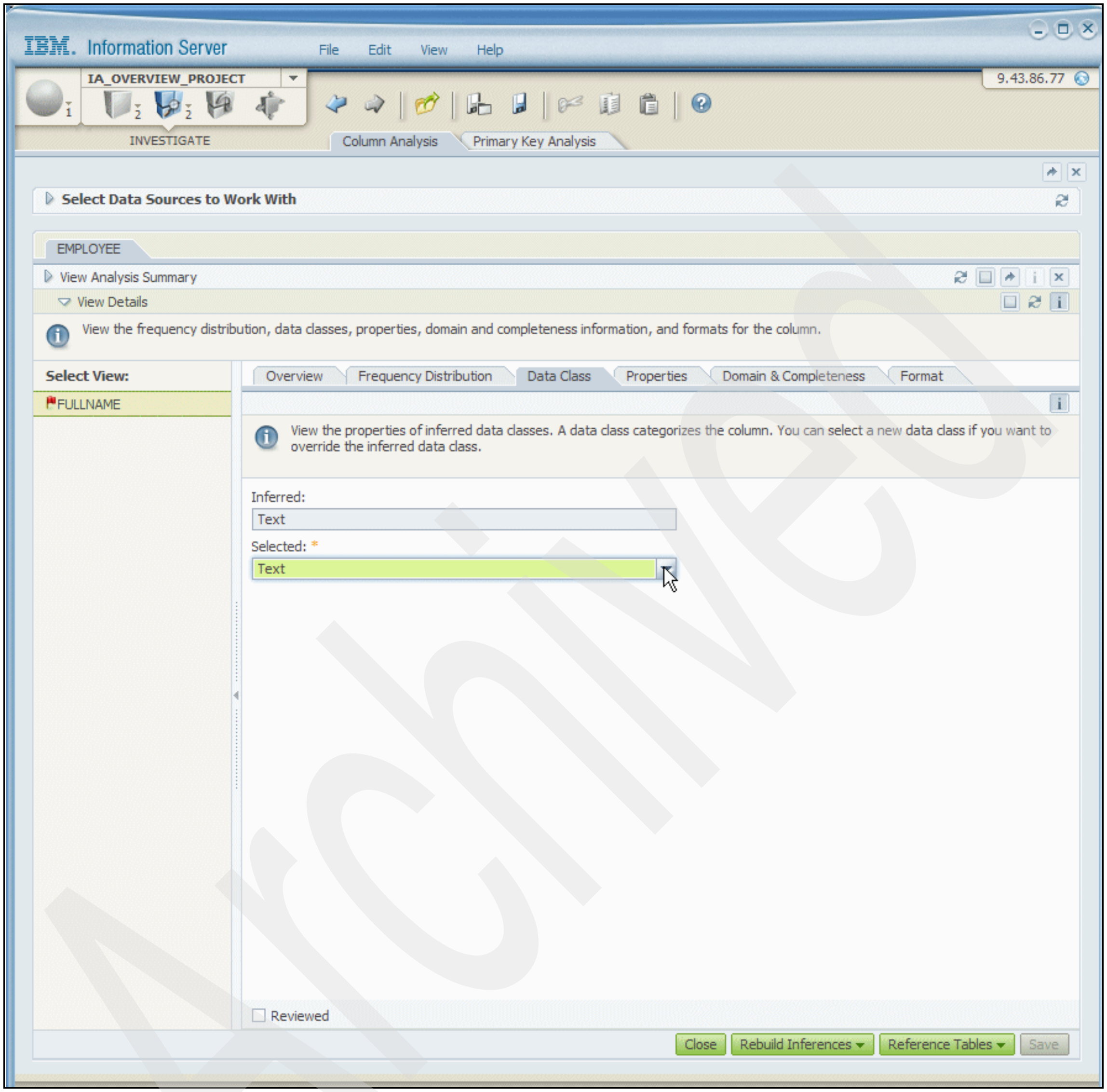
View the frequency distribution of a column. A frequency distribution shows details about the structure of a column. You can drill down in the table that contains the column to view values that correspond with the column, add a new data value to the column, or create a reference table.

Total Rows: 48
Data Class: Text
Cardinality: 48 100%

FULLNAME Column

	Frequency							Data Value	
Data Value	Count	Percent	Validity	Data Type	Length	Format	Transforma	Definition	Source
GREG_ORLA	1	2.08	Valid	STRING	12	AAAA_AAAA			Data
JOHN_GEYE	1	2.08	Valid	STRING	10	AAAA_AAAA			Data
KIYOSHI_YA	1	2.08	Valid	STRING	16	AAAAAAA_A			Data
WING_LEE	1	2.08	Valid	STRING	8	AAAA_AAA			Data
ETHEL_SCHI	1	2.08	Valid	STRING	15	AAAAA_AAA			Data
JASON_GOL	1	2.08	Valid	STRING	12	AAAAA_AAA			Data
MARILYN_S(	1	2.08	Valid	STRING	16	AAAAAAA_A			Data
RAMLAL_ME	1	2.08	Valid	STRING	12	AAAAAA_AA			Data
SALLY_KWA	1	2.08	Valid	STRING	10	AAAAA_AAA			Data
DANIEL_SMI	1	2.08	Valid	STRING	12	AAAAAA_AA			Data
DIAN_HEMM	1	2.08	Valid	STRING	14	AAAA_AAAA			Data
HEATHER_N	1	2.08	Valid	STRING	16	AAAAAAA_A			Data
JAMES_KHA	1	2.08	Valid	STRING	12	AAAAA_AAA			Data
JAMES_WAL	1	2.08	Valid	STRING	12	AAAAA_AAA			Data

Figure 1-159 Manage virtual columns in the EMPLOYEE table 9/16



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

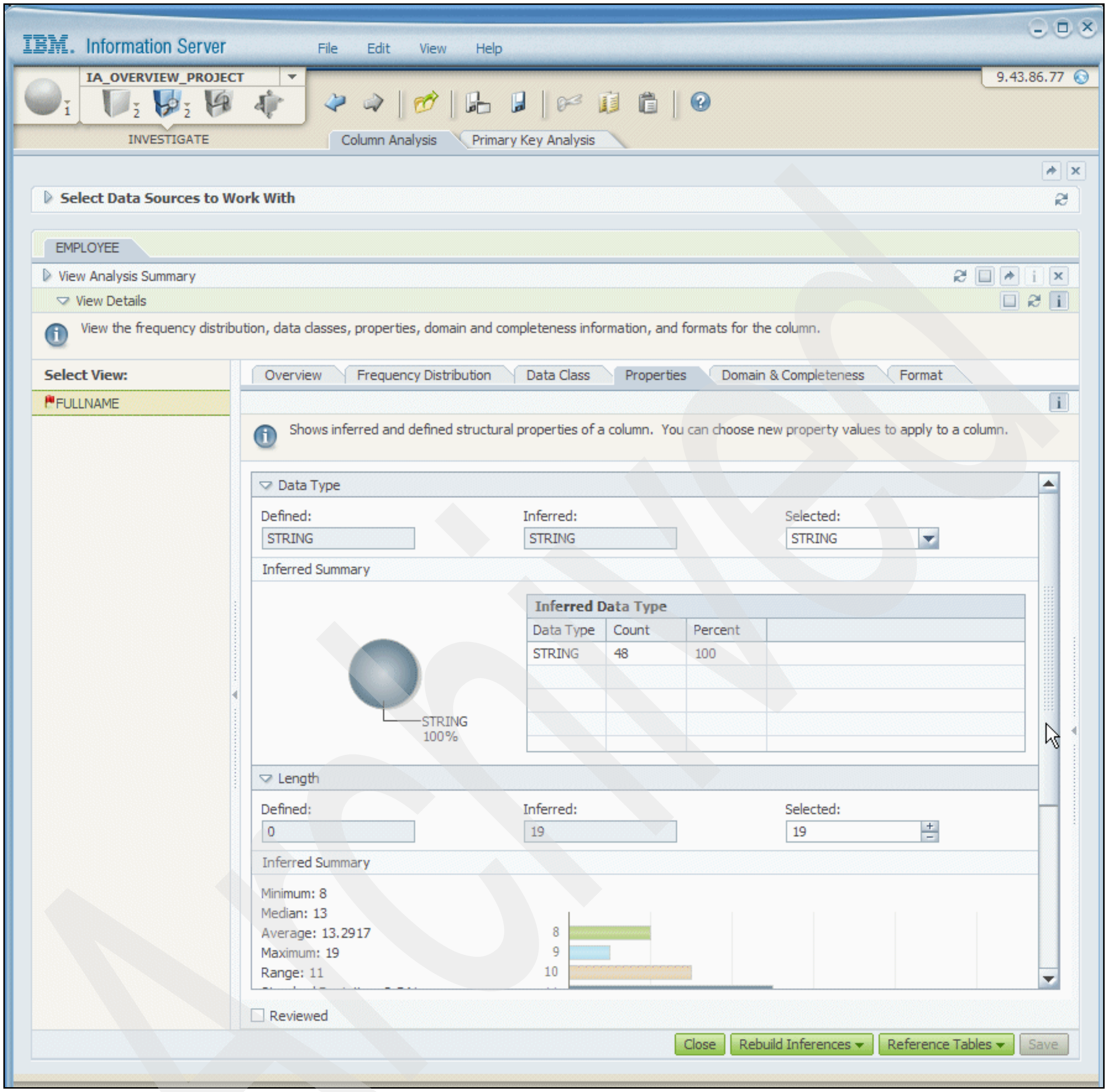
View the properties of inferred data classes. A data class categorizes the column. You can select a new data class if you want to override the inferred data class.

Inferred: Text

Selected: * Text

□ Reviewed

Figure 1-160 Manage virtual columns in the EMPLOYEE table 10/16



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Data Type

Defined: STRING
Inferred: STRING
Selected: STRING

Inferred Data Type

Data Type	Count	Percent
STRING	48	100

Length

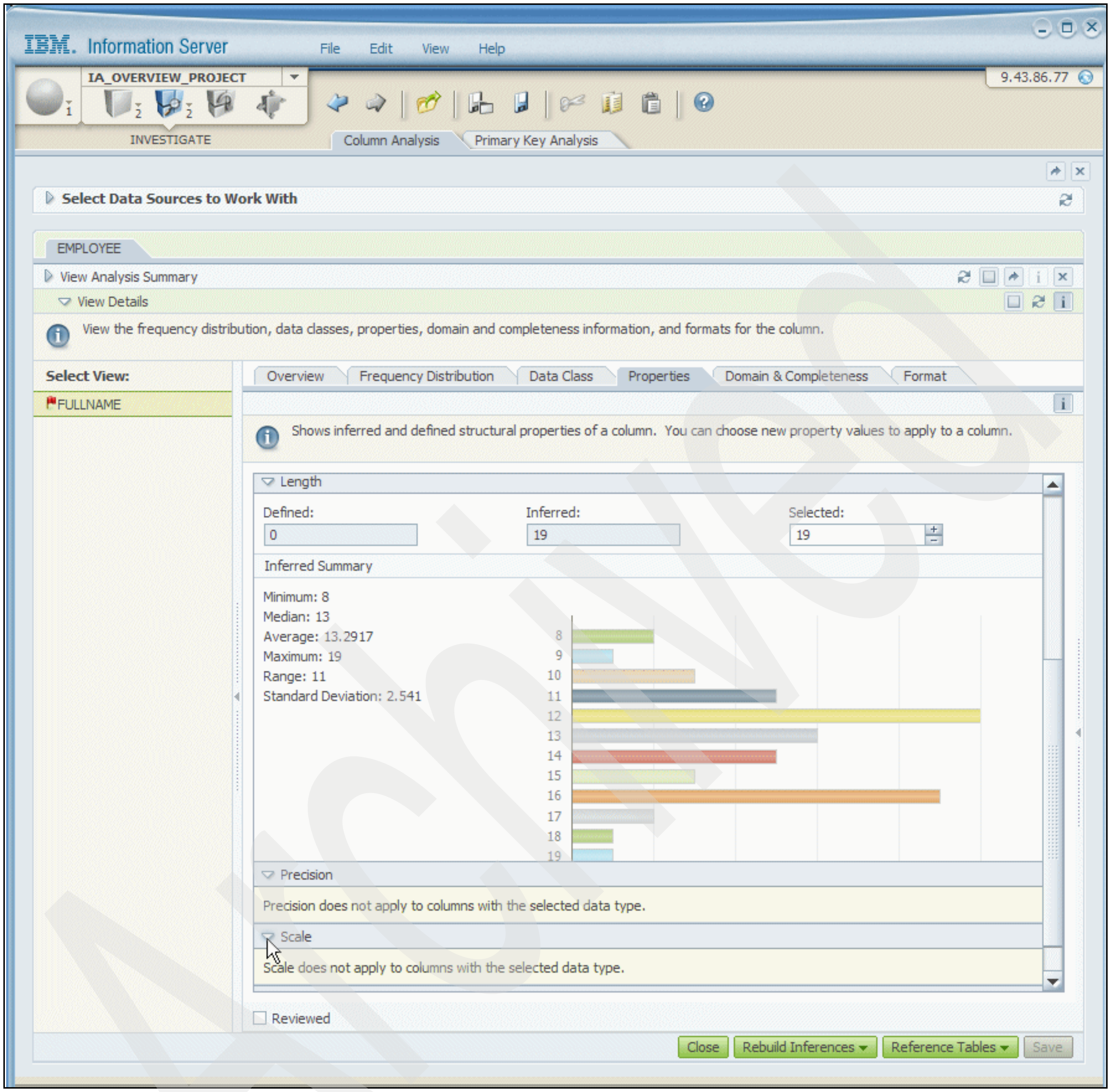
Defined: 0
Inferred: 19
Selected: 19

Inferred Summary

Minimum: 8
Median: 13
Average: 13.2917
Maximum: 19
Range: 11

□ Reviewed

Figure 1-161 Manage virtual columns in the EMPLOYEE table 11/16



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Defined: 0

Inferred: 19

Selected: 19

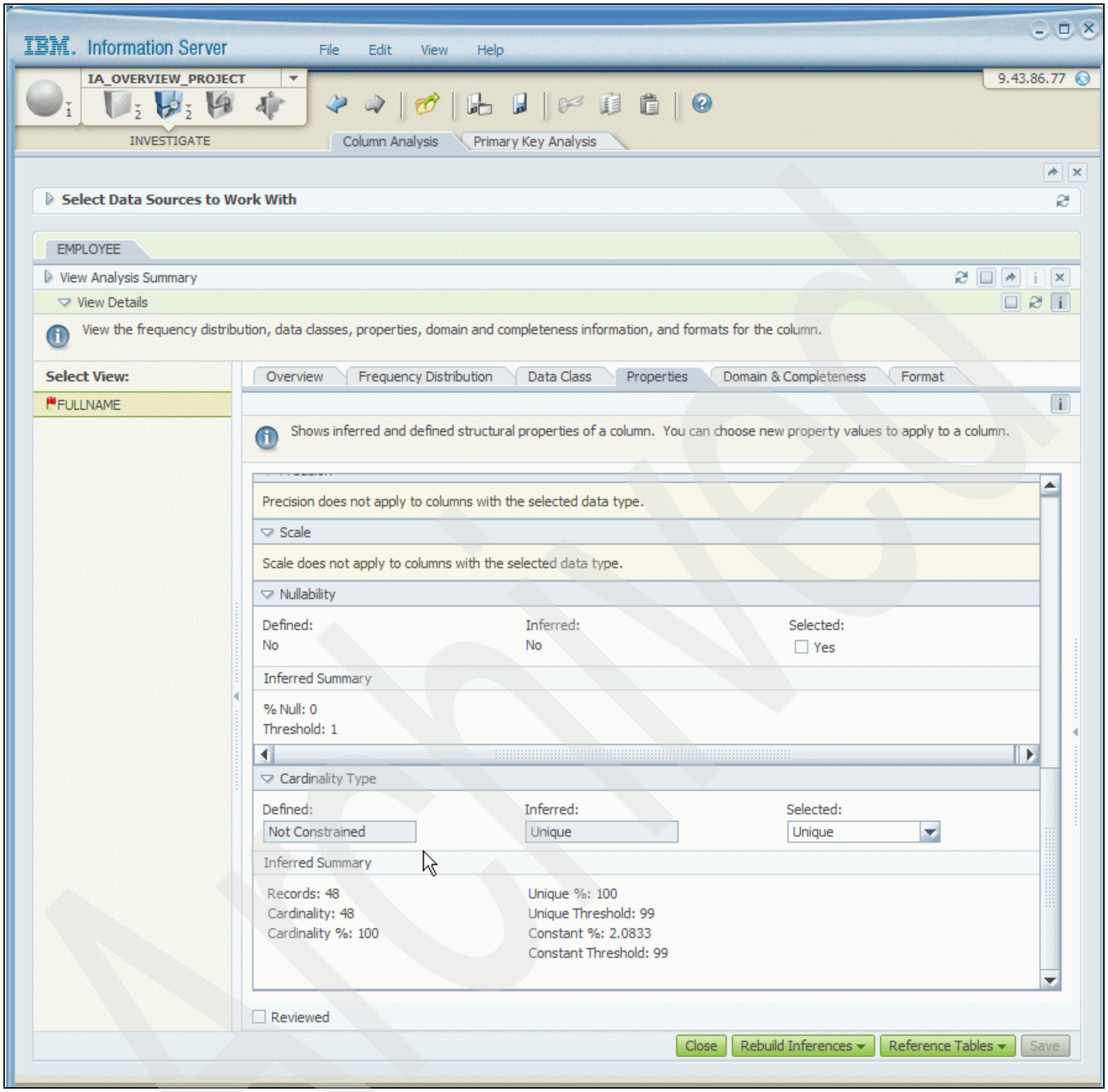
Minimum: 8
Median: 13
Average: 13.2917
Maximum: 19
Range: 11
Standard Deviation: 2.541

Precision does not apply to columns with the selected data type.

Scale does not apply to columns with the selected data type.

□ Reviewed

Figure 1-162 Manage virtual columns in the EMPLOYEE table 12/16



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows inferred and defined structural properties of a column. You can choose new property values to apply to a column.

Precision does not apply to columns with the selected data type.

Scale does not apply to columns with the selected data type.

Nullability

Defined: No
Inferred: No
Selected: □ Yes

Inferred Summary

% Null: 0
Threshold: 1

Cardinality Type

Defined: Not Constrained
Inferred: Unique
Selected: Unique

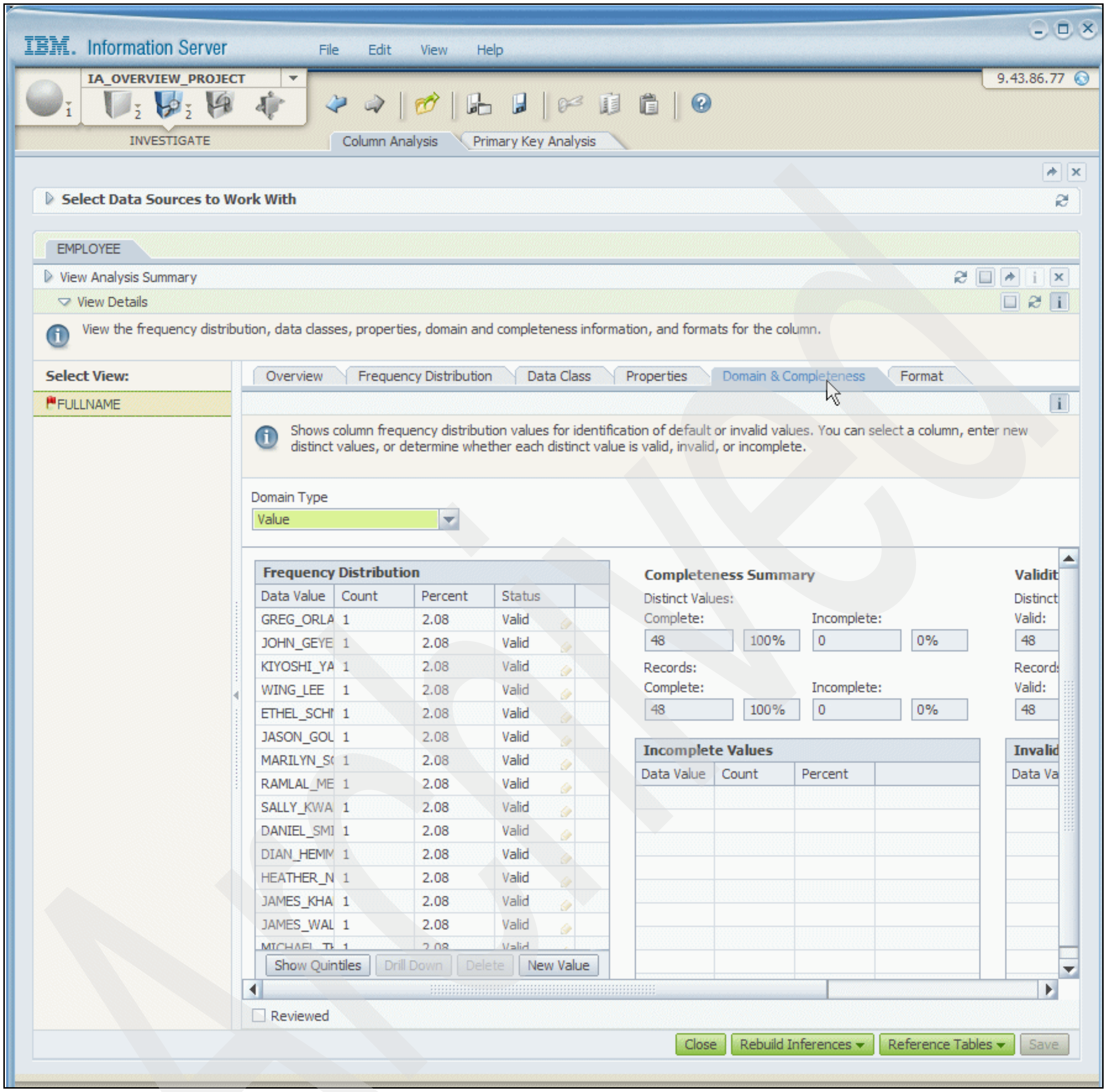
Inferred Summary

Records: 48
Cardinality: 48
Cardinality %: 100

Unique %: 100
Unique Threshold: 99
Constant %: 2.0833
Constant Threshold: 99

□ Reviewed

Figure 1-163 Manage virtual columns in the EMPLOYEE table 13/16



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View Details

View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

Shows column frequency distribution values for identification of default or invalid values. You can select a column, enter new distinct values, or determine whether each distinct value is valid, invalid, or incomplete.

Domain Type: Value

Frequency Distribution

Data Value	Count	Percent	Status
GREG_ORLA	1	2.08	Valid
JOHN_GEYE	1	2.08	Valid
KIYOSHI_YA	1	2.08	Valid
WING_LEE	1	2.08	Valid
ETHEL_SCHI	1	2.08	Valid
JASON_GOL	1	2.08	Valid
MARILYN_S(	1	2.08	Valid
RAMLAL_ME	1	2.08	Valid
SALLY_KWA	1	2.08	Valid
DANIEL_SMI	1	2.08	Valid
DIAN_HEMM	1	2.08	Valid
HEATHER_N	1	2.08	Valid
JAMES_KHA	1	2.08	Valid
JAMES_WAL	1	2.08	Valid
MICHAEL_TI	1	2.08	Valid

Completeness Summary

Distinct Values:
Complete: 48 100%
Incomplete: 0 0%

Records:
Complete: 48 100%
Incomplete: 0 0%

Validit

Distinct
Valid: 48

Record
Valid: 48

Incomplete Values

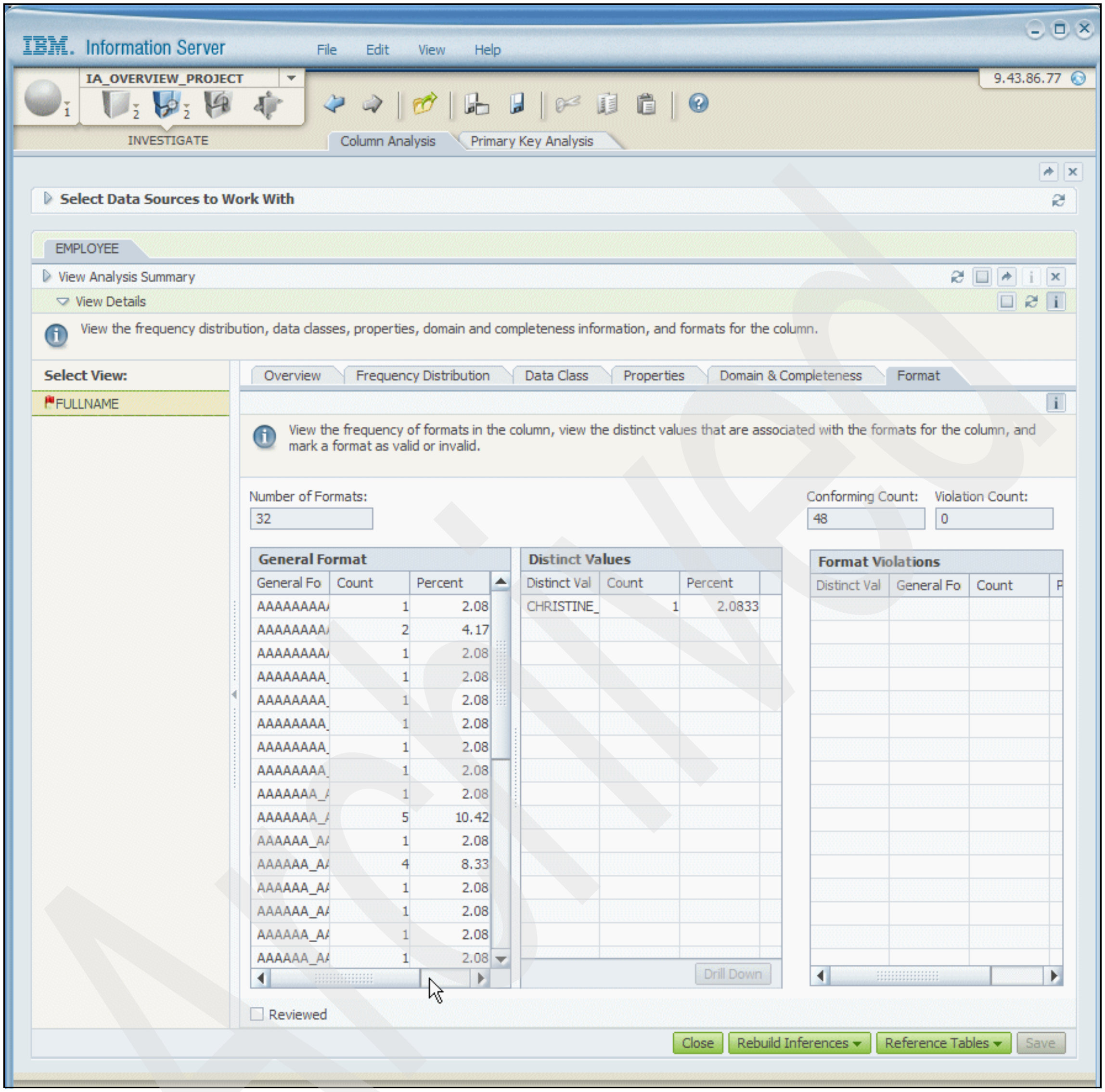
Data Value	Count	Percent

Invalid

Data Va

☐ Reviewed

Figure 1-164 Manage virtual columns in the EMPLOYEE table 14/16



View the frequency distribution, data classes, properties, domain and completeness information, and formats for the column.

View the frequency of formats in the column, view the distinct values that are associated with the formats for the column, and mark a format as valid or invalid.

Number of Formats: 32

Conforming Count: 48

Violation Count: 0

General Format

General Fo	Count	Percent
AAAAAAAA/	1	2.08
AAAAAAAA/	2	4.17
AAAAAAAA/	1	2.08
AAAAAAAA_	1	2.08
AAAAAAAA_	1	2.08
AAAAAAAA_	1	2.08
AAAAAAAA_	1	2.08
AAAAAAAA_	1	2.08
AAAAAAA_/	1	2.08
AAAAAAA_/	5	10.42
AAAAAA_A/	1	2.08
AAAAAA_A/	4	8.33
AAAAAA_A/	1	2.08
AAAAAA_A/	1	2.08
AAAAAA_A/	1	2.08
AAAAAA_A/	1	2.08

Distinct Values

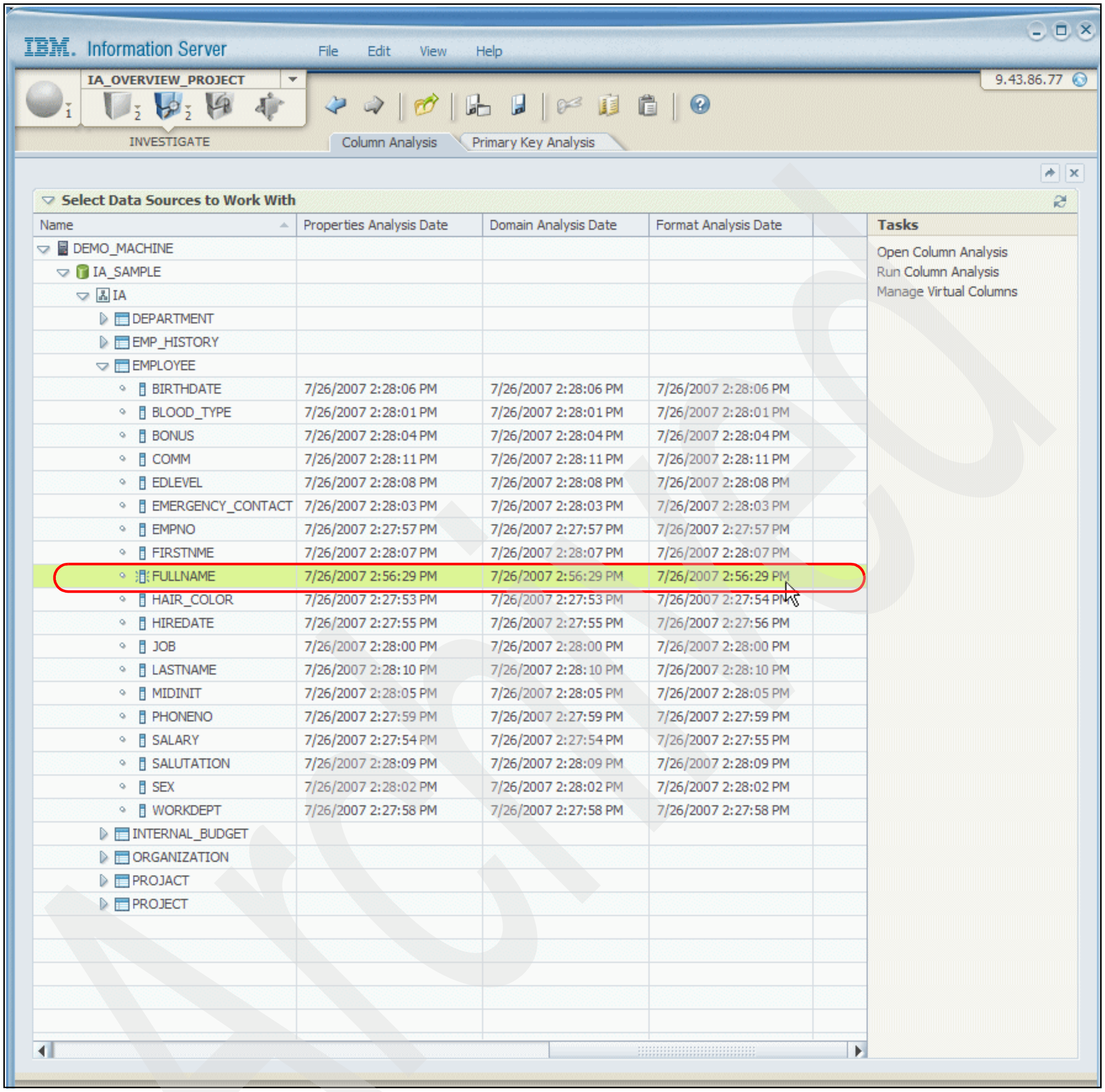
Distinct Val	Count	Percent
CHRISTINE_	1	2.0833

Format Violations

Distinct Val	General Fo	Count	F

☐ Reviewed

Figure 1-165 Manage virtual columns in the EMPLOYEE table 15/16



Select Data Sources to Work With

Name	Properties Analysis Date	Domain Analysis Date	Format Analysis Date
DEMO_MACHINE			
IA_SAMPLE			
IA			
DEPARTMENT			
EMP_HISTORY			
EMPLOYEE			
BIRTHDATE	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM	7/26/2007 2:28:06 PM
BLOOD_TYPE	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM	7/26/2007 2:28:01 PM
BONUS	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM	7/26/2007 2:28:04 PM
COMM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM	7/26/2007 2:28:11 PM
EDLEVEL	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM	7/26/2007 2:28:08 PM
EMERGENCY_CONTACT	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM	7/26/2007 2:28:03 PM
EMPNO	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM	7/26/2007 2:27:57 PM
FIRSTNME	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM	7/26/2007 2:28:07 PM
FULLNAME	7/26/2007 2:56:29 PM	7/26/2007 2:56:29 PM	7/26/2007 2:56:29 PM
HAIR_COLOR	7/26/2007 2:27:53 PM	7/26/2007 2:27:53 PM	7/26/2007 2:27:54 PM
HIREDATE	7/26/2007 2:27:55 PM	7/26/2007 2:27:55 PM	7/26/2007 2:27:56 PM
JOB	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM	7/26/2007 2:28:00 PM
LASTNAME	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM	7/26/2007 2:28:10 PM
MIDINIT	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM	7/26/2007 2:28:05 PM
PHONENO	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM	7/26/2007 2:27:59 PM
SALARY	7/26/2007 2:27:54 PM	7/26/2007 2:27:54 PM	7/26/2007 2:27:55 PM
SALUTATION	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM	7/26/2007 2:28:09 PM
SEX	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM	7/26/2007 2:28:02 PM
WORKDEPT	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM	7/26/2007 2:27:58 PM
INTERNAL_BUDGET			
ORGANIZATION			
PROJACT			
PROJECT			

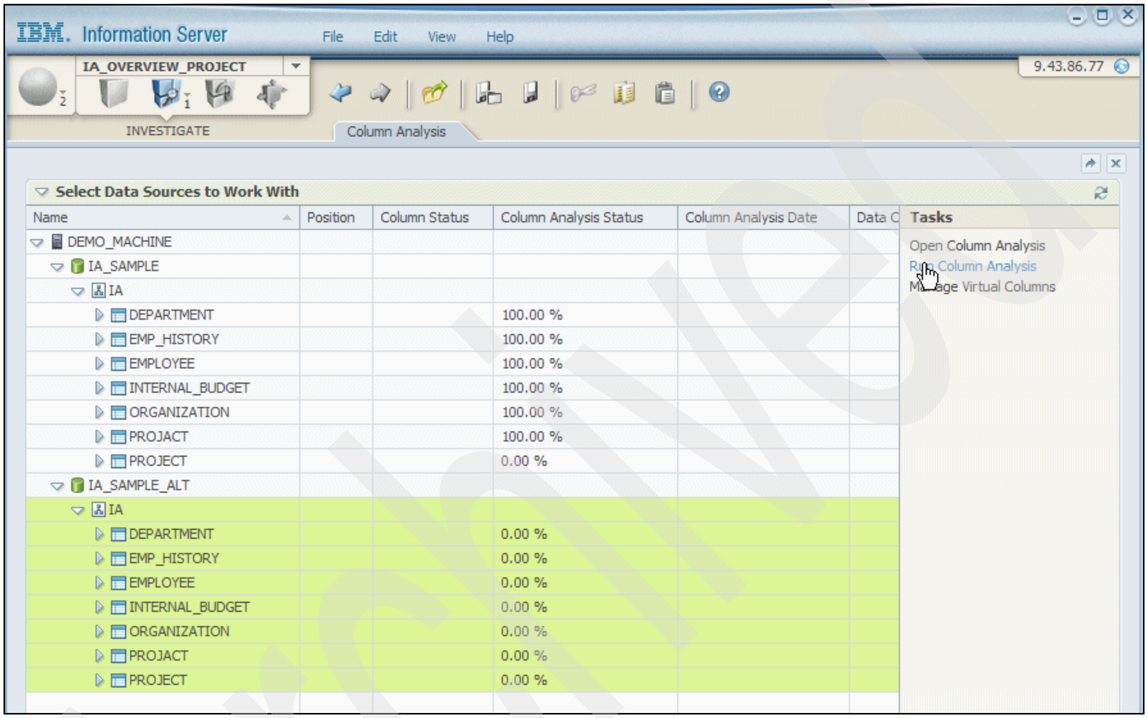
Tasks

- Open Column Analysis
- Run Column Analysis
- Manage Virtual Columns

Figure 1-166 Manage virtual columns in the EMPLOYEE table 16/16

Run column analysis on a set of tables

You can run column analysis at the granularity of a set of tables. In the Select Data Sources to Work With workspaces, select all the tables with the schema IA and click Run Column Analysis as shown in Figure 1-167 on page 222. At the end of the job run, the Column Analysis Status changes from 0% in Figure 1-167 to 100% in Figure 1-168 on page 223.

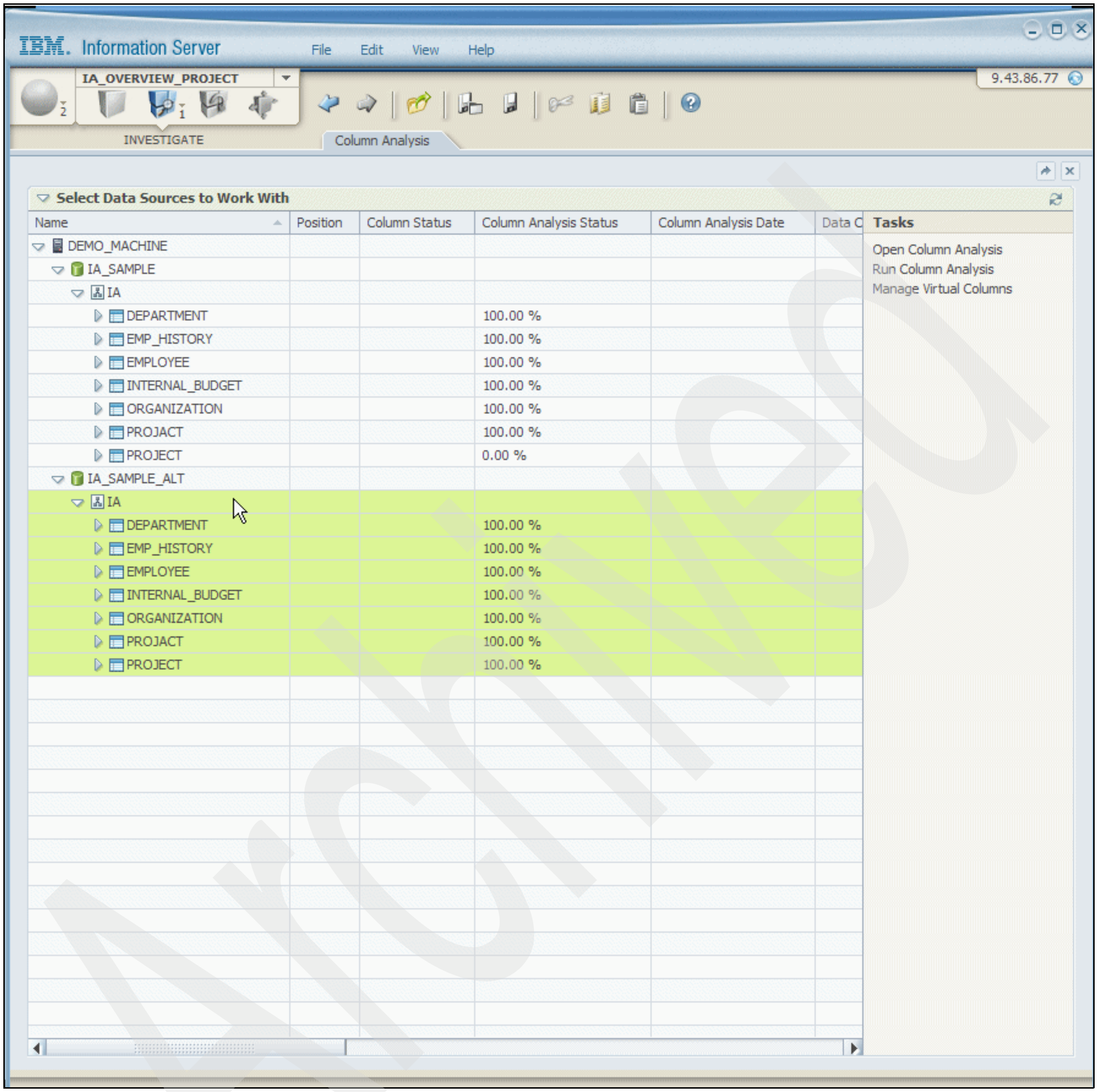


Name	Position	Column Status	Column Analysis Status	Column Analysis Date	Data C
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			100.00 %		
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		
PROJECT			0.00 %		
IA_SAMPLE_ALT					
IA					
DEPARTMENT			0.00 %		
EMP_HISTORY			0.00 %		
EMPLOYEE			0.00 %		
INTERNAL_BUDGET			0.00 %		
ORGANIZATION			0.00 %		
PROJACT			0.00 %		
PROJECT			0.00 %		

Tasks

Open Column Analysis
Run Column Analysis
Manage Virtual Columns

Figure 1-167   Run a column analysis job on a set of tables 1/2



Select Data Sources to Work With

Name	Position	Column Status	Column Analysis Status	Column Analysis Date	Data C
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			100.00 %		
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		
PROJECT			0.00 %		
IA_SAMPLE_ALT					
IA					
DEPARTMENT			100.00 %		
EMP_HISTORY			100.00 %		
EMPLOYEE			100.00 %		
INTERNAL_BUDGET			100.00 %		
ORGANIZATION			100.00 %		
PROJACT			100.00 %		
PROJECT			100.00 %		

Tasks

Open Column Analysis
Run Column Analysis
Manage Virtual Columns

Figure 1-168 Run a column analysis job on a set of tables 2/2

1.8 Primary key analysis

To understand the structure and integrity of your data, you can use primary key analysis to identify and validate primary key candidates. Primary keys are columns that uniquely identify all of the rows in a table:

- Single column primary key candidates are identified by analyzing the uniqueness of distinct values in each column based on its frequency distribution statistics.
- Multiple column primary key candidates are initially determined by analyzing the uniqueness of distinct values for each column combination based on a data sample of the table. You can then run a full analysis on the multiple column primary key candidates to confirm.

After a primary key analysis job completes, you can review and select a primary key from a list of inferred candidates. You can also identify duplicate primary key values for defined or selected primary keys.

You analyze primary keys that are already defined in your data and to identify columns that are candidates for primary keys. A primary key must be unique and cannot contain null values. For example, a column that contains a customer number might be inferred as a primary key candidate by the system because it is unique and does not have null values. During analysis, multiple unique columns might be inferred as candidates. When you review the results, you choose the columns that you want to define as primary keys from the list of inferred candidates.

In the following sections, we briefly describe the main functions of primary key analysis and the results that are produced by primary key analysis. Using a sample set of tables, we describe the process of running primary key analysis and discuss the results of primary key analysis.

1.8.1 Primary key analysis functions

Primary key analysis has the following characteristics:

- You must have the Information Analyzer Business Analyst role to open and edit Primary Key analysis. You must have the Information Analyzer Data Operator role to run multi-column primary key analysis or to create a data sample.

- Can be performed on a single column or on multiple columns.
  - Single column primary key analysis is derived from column analysis, and no separate job execution is involved.
  - Multi-column primary key analysis requires a separate job to be executed.
- Allows you to run multi-column primary key analysis on a subset of the data using a sampling technique.
- Perform duplicate checks on selected multi-column primary keys or candidate primary keys. These checks can be scheduled to run at a set date and time.
- Reports can be produced of primary key analysis such as candidate duplicate exceptions, defined and candidate summary, defined primary key duplicate exceptions, and defined primary key summary. These are described in 1.14, “Reports” on page 394.

> **Note:** For complete details, refer to *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.

After a column analysis job completes and a frequency distribution for the column is generated, all single columns and defined multi-column keys are available to review using primary key analysis. Using the frequency distributions that were generated during the column analysis job, the cardinality percentage is evaluated for each column in a table. A cardinality percentage is the total number of distinct values in the column divided by the total number of rows in the table that the column resides in. For example, 1000 distinct values divided by 1000 rows equals 1.0 or 100% uniqueness. Columns that have a cardinality percentage of 95% or greater than the system threshold are identified as primary key candidates. You configure the system threshold before you analyze primary keys if you want to allow more variations of unique values. For example, a threshold of 90% will allow more columns to be considered candidates than a threshold of 100%. A threshold of 100% does not allow any variations. During analysis, any cardinality percentage that is less than 100% indicates that there are duplicate values in a column.

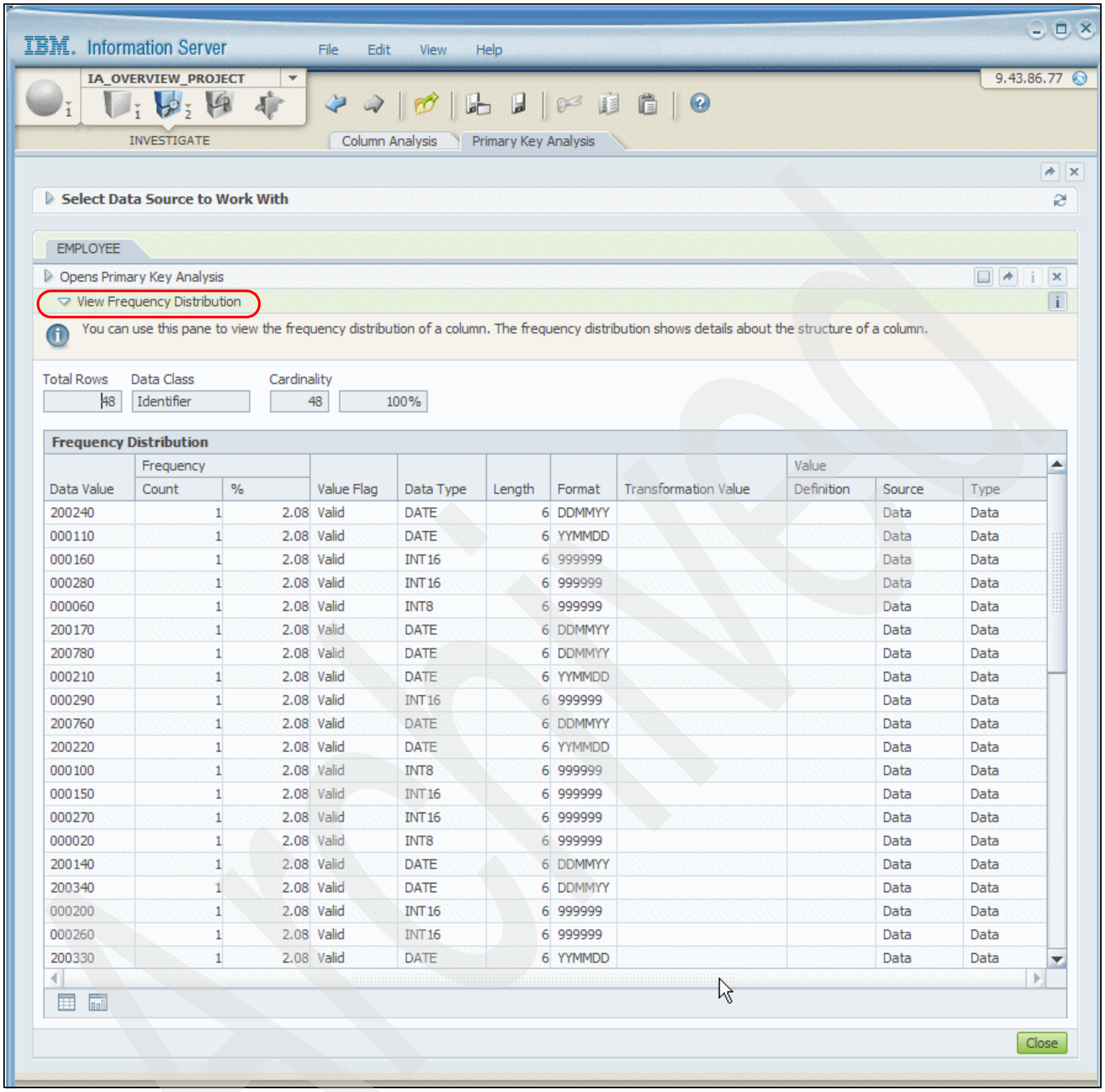
Duplicate columns might exist if your data was integrated with another data source or if the structure of your data changed. For example, by using a system threshold of 95%, column B in table 1 has a cardinality percentage of 98%. The cardinality percentage for column B is above the threshold, and column B is inferred as a primary key candidate. However, when you review the results, you find that 2% of the values in column B are duplicate values. The 2% of duplicate values renders the inference flawed because the column does not contain the uniqueness to be a primary key. You can either use a data cleansing tool to remove the duplicate values from the column, or choose another primary key candidate.

Attention: If your data already contains defined primary keys, the columns will be marked as primary keys during analysis. However, you can still use the results of a primary key analysis job to assess the validity of defined primary keys.

1.8.2 Primary key analysis results

A primary key analysis produces the following categories of output:

- Frequency distribution of the columns, and the identification of candidate keys based on the Primary Key Threshold. The Flag Percentage Minimum flags the candidate columns. You can accept a candidate as a primary key which then becomes *Selected*. If a Selected primary key already exists, it is replaced when you accept another column or set of columns. Figure 1-169 shows the frequency distribution of the EMPNO column in the EMPLOYEE table. This view is similar to the frequency distribution view in column analysis shown in Figure 1-80 on page 118.



Select Data Source to Work With

EMPLOYEE

Opens Primary Key Analysis

View Frequency Distribution

You can use this pane to view the frequency distribution of a column. The frequency distribution shows details about the structure of a column.

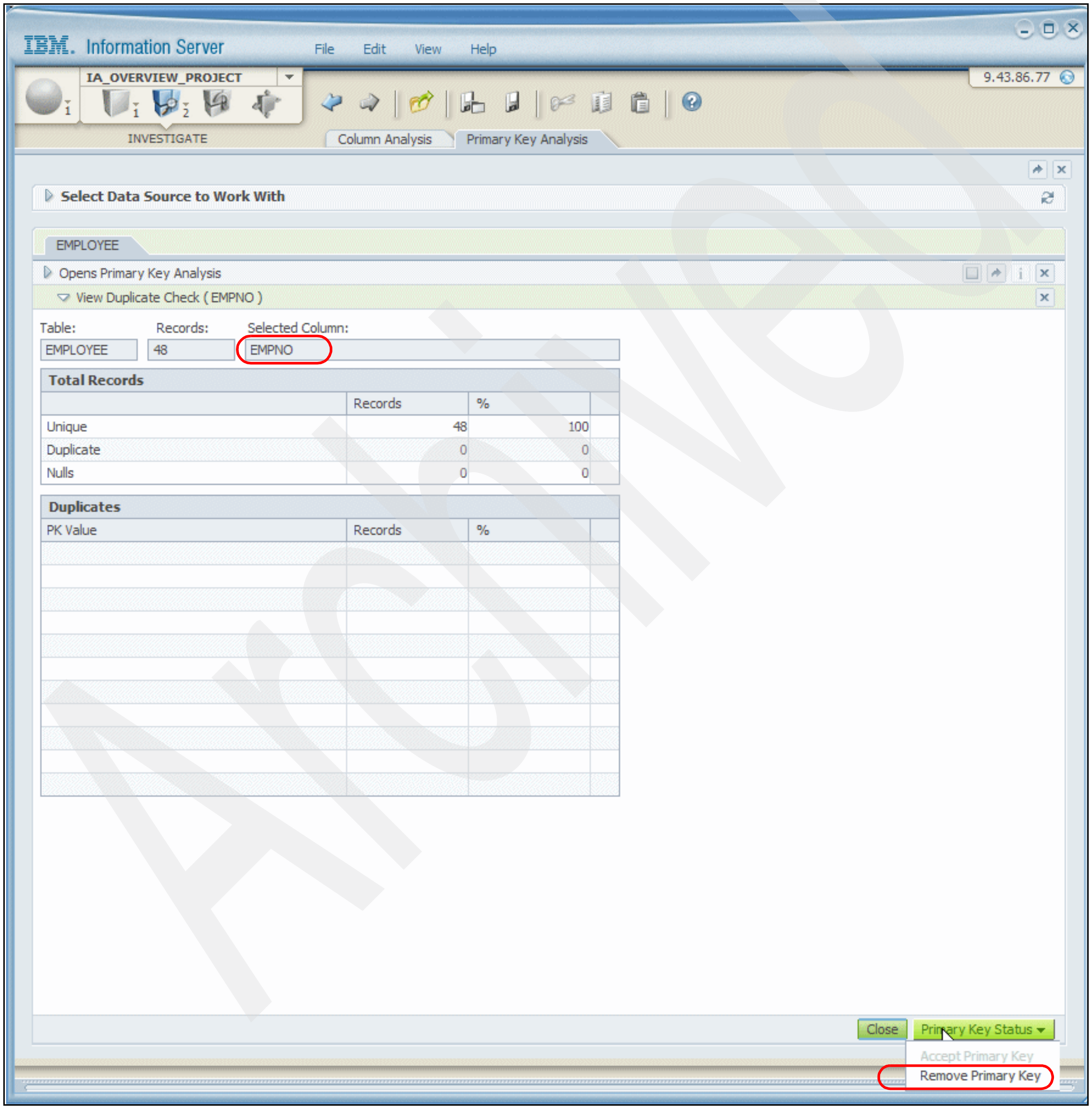
Total Rows: 48
Data Class: Identifier
Cardinality: 48 100%

Frequency Distribution

	Frequency							Value		
Data Value	Count	%	Value Flag	Data Type	Length	Format	Transformation Value	Definition	Source	Type
200240	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
000110	1	2.08	Valid	DATE	6	YYMMDD			Data	Data
000160	1	2.08	Valid	INT16	6	999999			Data	Data
000280	1	2.08	Valid	INT16	6	999999			Data	Data
000060	1	2.08	Valid	INT8	6	999999			Data	Data
200170	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
200780	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
000210	1	2.08	Valid	DATE	6	YYMMDD			Data	Data
000290	1	2.08	Valid	INT16	6	999999			Data	Data
200760	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
200220	1	2.08	Valid	DATE	6	YYMMDD			Data	Data
000100	1	2.08	Valid	INT8	6	999999			Data	Data
000150	1	2.08	Valid	INT16	6	999999			Data	Data
000270	1	2.08	Valid	INT16	6	999999			Data	Data
000020	1	2.08	Valid	INT8	6	999999			Data	Data
200140	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
200340	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
000200	1	2.08	Valid	INT16	6	999999			Data	Data
000260	1	2.08	Valid	INT16	6	999999			Data	Data
200330	1	2.08	Valid	DATE	6	YYMMDD			Data	Data

Figure 1-169 Frequency distribution of a single column primary key analysis

- You can view the duplicate values in a set of columns to determine whether those values exclude the columns from being considered as the primary key of the table by running a duplicate check. Figure 1-170 shows the duplicate check on the EMPNO column in the EMPLOYEE table. It shows all unique values, with no nulls. EMPNO is selected as the primary key, which allows you to remove its primary key status as shown.



Select Data Source to Work With

EMPLOYEE

Opens Primary Key Analysis

View Duplicate Check (EMPNO)

Table: EMPLOYEE

Records: 48

Selected Column: EMPNO

Total Records

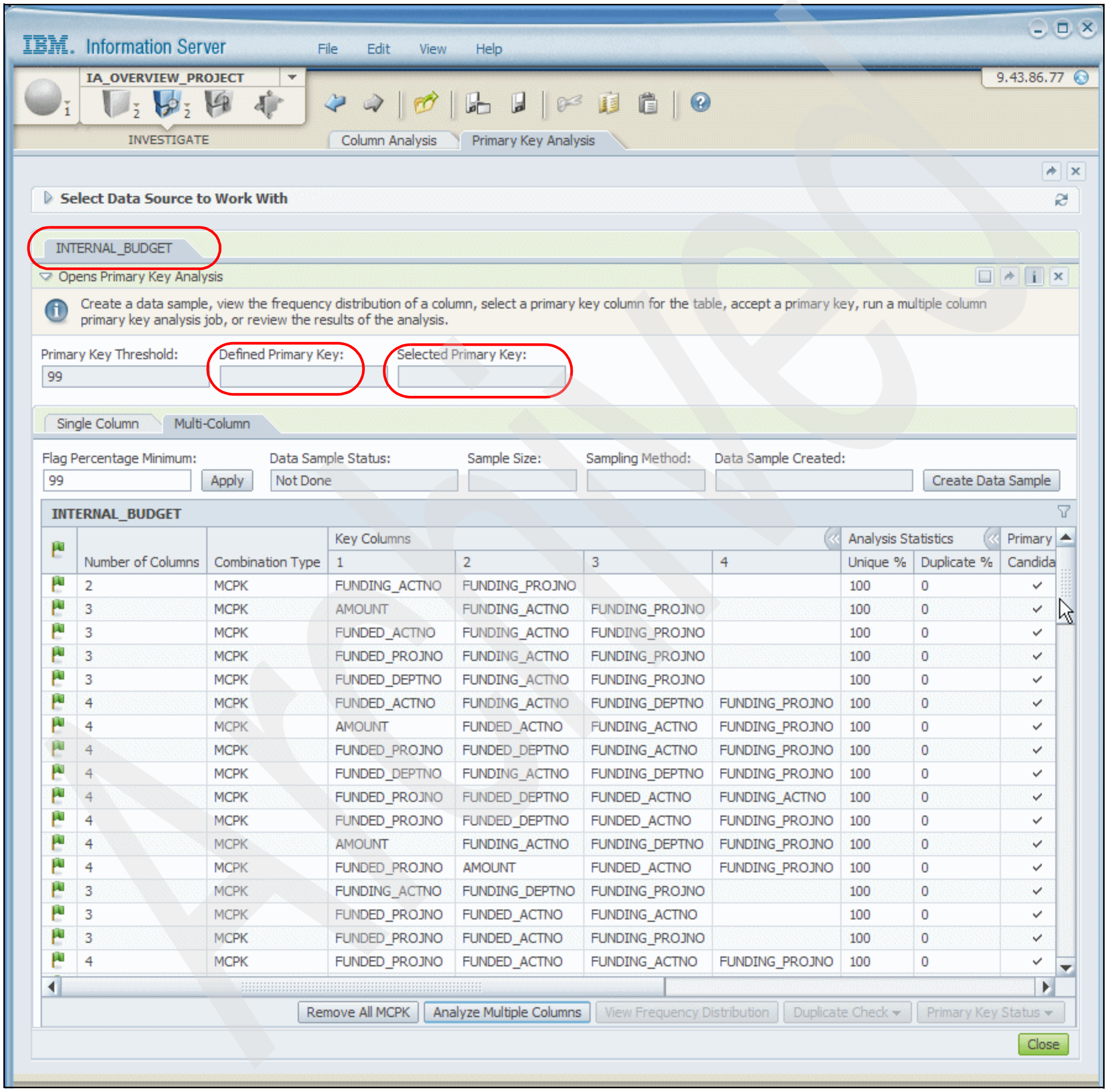
	Records	%
Unique	48	100
Duplicate	0	0
Nulls	0	0

Duplicates

PK Value	Records	%

Figure 1-170 Duplicate check on EMPNO column in the EMPLOYEE table

- When you run multi-column analysis, it identifies potential multi-column candidates for a primary key depending upon the Composite key maximum threshold specified and Max columns threshold. Figure 1-171 shows the potential multi-column primary key candidates for the INTERNAL_BUDGET table that has no primary key defined.



Create a data sample, view the frequency distribution of a column, select a primary key column for the table, accept a primary key, run a multiple column primary key analysis job, or review the results of the analysis.

Primary Key Threshold: 99

Defined Primary Key:

Selected Primary Key:

Flag Percentage Minimum: 99 Apply

Data Sample Status: Not Done

Sample Size:

Sampling Method:

Data Sample Created:

INTERNAL_BUDGET

⚑	Number of Columns	Combination Type	Key Columns				Analysis Statistics		Primary
			1	2	3	4	Unique %	Duplicate %	Candida
⚑	2	MCPK	FUNDING_ACTNO	FUNDING_PROJNO			100	0	✓
⚑	3	MCPK	AMOUNT	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
⚑	3	MCPK	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
⚑	3	MCPK	FUNDED_PROJNO	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
⚑	3	MCPK	FUNDED_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
⚑	4	MCPK	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	100	0	✓
⚑	4	MCPK	AMOUNT	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_PROJNO	100	0	✓
⚑	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO	100	0	✓
⚑	4	MCPK	FUNDED_DEPTNO	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	100	0	✓
⚑	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDED_ACTNO	FUNDING_ACTNO	100	0	✓
⚑	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDED_ACTNO	FUNDING_PROJNO	100	0	✓
⚑	4	MCPK	AMOUNT	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	100	0	✓
⚑	4	MCPK	FUNDED_PROJNO	AMOUNT	FUNDED_ACTNO	FUNDING_PROJNO	100	0	✓
⚑	3	MCPK	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO		100	0	✓
⚑	3	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_ACTNO		100	0	✓
⚑	3	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_PROJNO		100	0	✓
⚑	4	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_PROJNO	100	0	✓

Figure 1-171 Multi-column analysis on the INTERNAL_BUDGET table with no defined primary key

1.8.3 Primary key analysis usage scenario

The modified sample database described in Figure 1-11 on page 31 is used in the primary key analysis examples that we describe here.

In this section, we describe how to assess primary key analysis on a defined single column primary key table, defined multi-column primary key table, undefined single column primary key table, undefined multi-column primary key table, and a table with no primary key defined, followed by a review of the analysis results.

> **Attention:** In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

Defined single column primary key analysis

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), run a primary key analysis job on the EMPLOYEE table as follows:

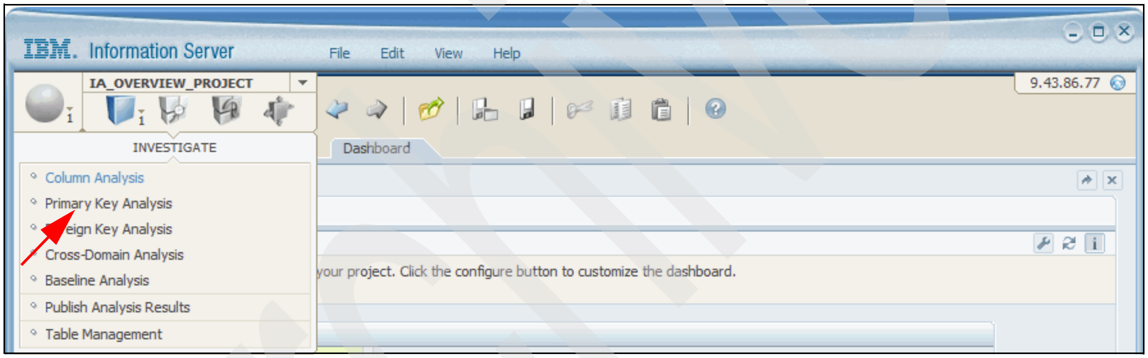
1. On the Investigate navigator menu in the console, select **Primary Key Analysis** as shown in Figure 1-172 on page 231.
2. On the Primary Key Analysis workspace, select the EMPLOYEE table and click **Open Primary Key Analysis** in the Tasks pane as shown in Figure 1-173 on page 232.
3. Select the Single Column tab to view the analysis results for all the columns in the EMPLOYEE table as shown in Figure 1-174 on page 233. The EMPNO column shows 100% unique. If the selected column is not 100% unique, click **View Duplicate Check** to view the duplicate data values within the selected column.
4. Select LASTNAME and click **View Duplicate Check** in Figure 1-174 on page 233 to view the details shown in Figure 1-175 on page 234. It shows the percentage of unique and duplicate values in the column and the duplicate values in the column. You can select it as a primary key by clicking **Accept Primary Key** (not shown here).

> **Attention:** This action does *not* change the primary key definitions in the table—it only saves the changes in the metadata repository. If you want to change the data itself, you must make the change in the external database and then rerun column analysis to see the data changes.

5. Click **Close** to return to the analysis results screen (Figure 1-175 on page 234).
6. Select EMPNO and click **Frequency Distribution** in the analysis results summary in Figure 1-176 on page 235 to view the frequency distribution of the EMPNO column. Figure 1-177 on page 236 shows the details about the structure of the EMPNO column.
7. Click **Close** as shown in Figure 1-177 on page 236 and the analysis results summary as shown in Figure 1-178 on page 237 to proceed to the Primary Key Analysis workspace shown in Figure 1-179 on page 238.

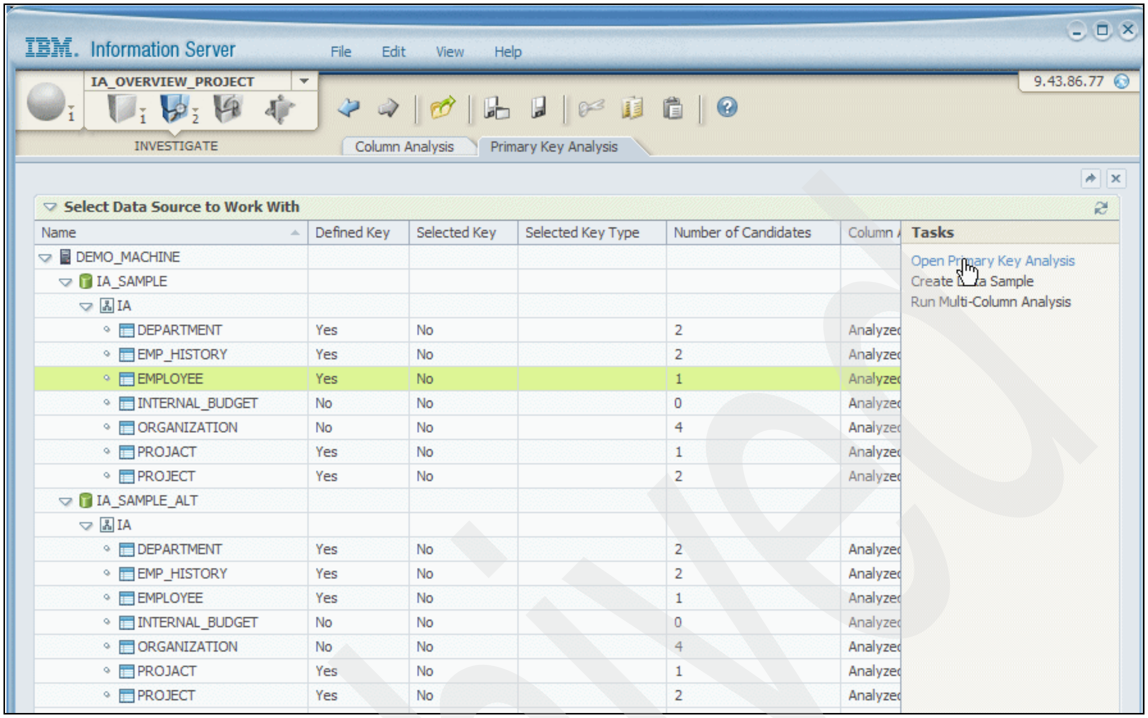
   Figure 1-179 on page 238 shows the EMPLOYEE table as having a single column Selected Key.

> **Note:** If you cannot identify an appropriate single column primary key, you can look for a multi-column primary key.



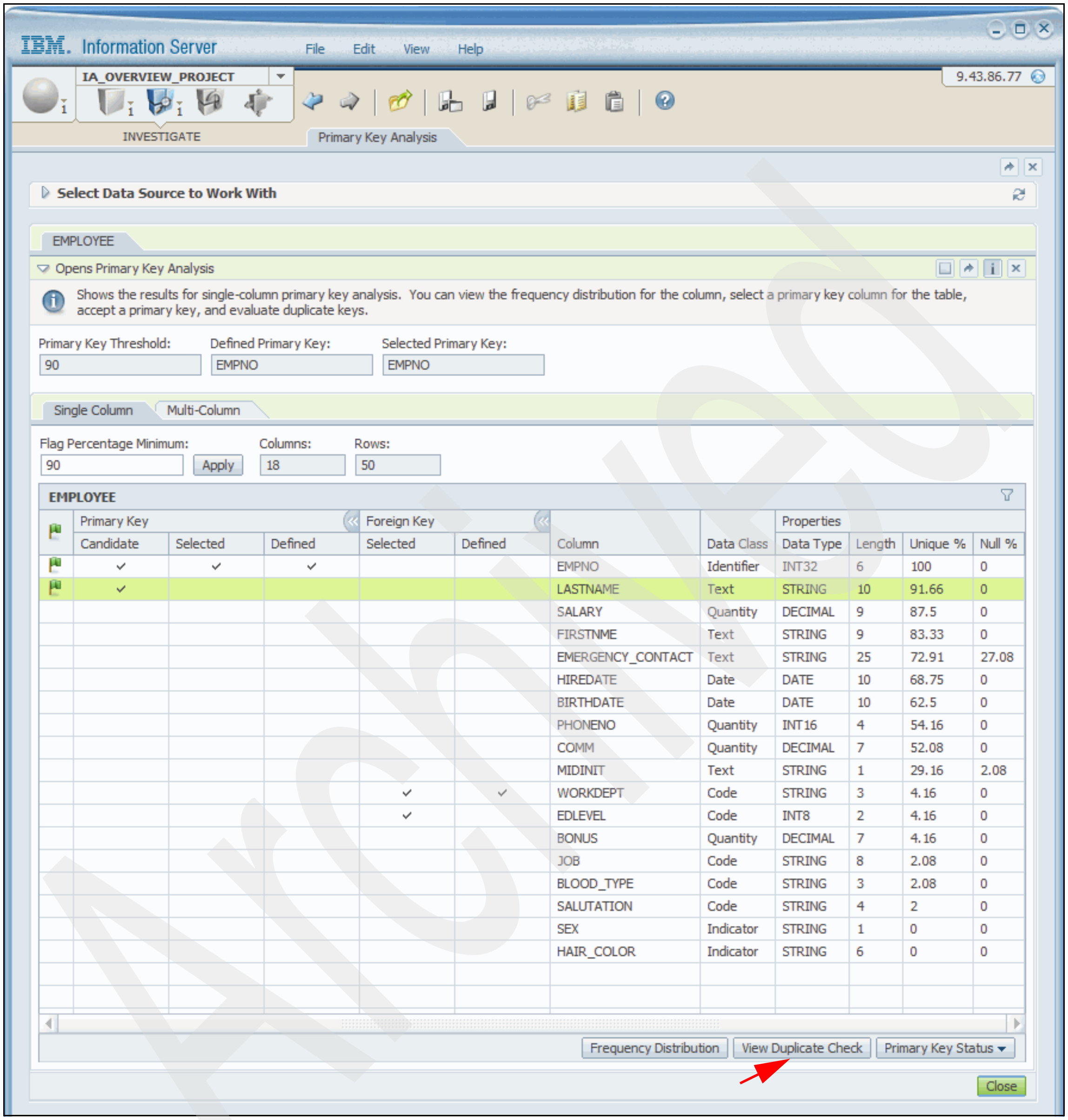
your project. Click the configure button to customize the dashboard.

Figure 1-172 Investigating Primary Key Analysis



Name	Defined Key	Selected Key	Selected Key Type	Number of Candidates	Column A
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT	Yes	No		2	Analyzed
EMP_HISTORY	Yes	No		2	Analyzed
EMPLOYEE	Yes	No		1	Analyzed
INTERNAL_BUDGET	No	No		0	Analyzed
ORGANIZATION	No	No		4	Analyzed
PROJACT	Yes	No		1	Analyzed
PROJECT	Yes	No		2	Analyzed
IA_SAMPLE_ALT					
IA					
DEPARTMENT	Yes	No		2	Analyzed
EMP_HISTORY	Yes	No		2	Analyzed
EMPLOYEE	Yes	No		1	Analyzed
INTERNAL_BUDGET	No	No		0	Analyzed
ORGANIZATION	No	No		4	Analyzed
PROJACT	Yes	No		1	Analyzed
PROJECT	Yes	No		2	Analyzed

Figure 1-173 Single column primary key analysis 1/7



Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 90

Defined Primary Key: EMPNO

Selected Primary Key: EMPNO

Flag Percentage Minimum: 90

Columns: 18

Rows: 50

EMPLOYEE

	Primary Key			Foreign Key				Properties			
	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null %
🟩	✓	✓	✓			EMPNO	Identifier	INT32	6	100	0
🟩	✓					LASTNAME	Text	STRING	10	91.66	0
						SALARY	Quantity	DECIMAL	9	87.5	0
						FIRSTNME	Text	STRING	9	83.33	0
						EMERGENCY_CONTACT	Text	STRING	25	72.91	27.08
						HIREDATE	Date	DATE	10	68.75	0
						BIRTHDATE	Date	DATE	10	62.5	0
						PHONENO	Quantity	INT16	4	54.16	0
						COMM	Quantity	DECIMAL	7	52.08	0
						MIDINIT	Text	STRING	1	29.16	2.08
				✓	✓	WORKDEPT	Code	STRING	3	4.16	0
				✓		EDLEVEL	Code	INT8	2	4.16	0
						BONUS	Quantity	DECIMAL	7	4.16	0
						JOB	Code	STRING	8	2.08	0
						BLOOD_TYPE	Code	STRING	3	2.08	0
						SALUTATION	Code	STRING	4	2	0
						SEX	Indicator	STRING	1	0	0
						HAIR_COLOR	Indicator	STRING	6	0	0

Figure 1-174 Single column primary key analysis 2/7

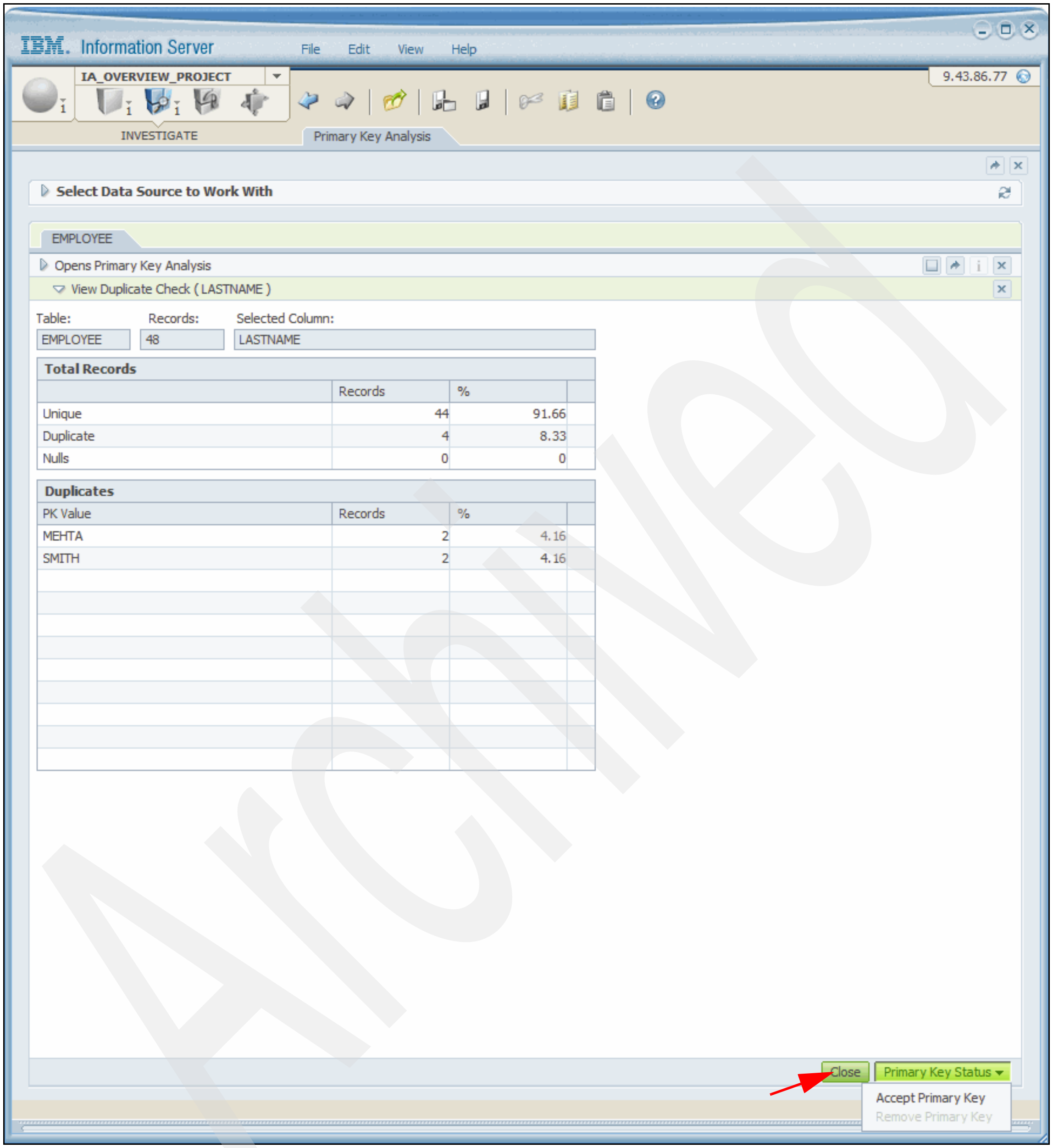


Table: EMPLOYEE
Records: 48
Selected Column: LASTNAME

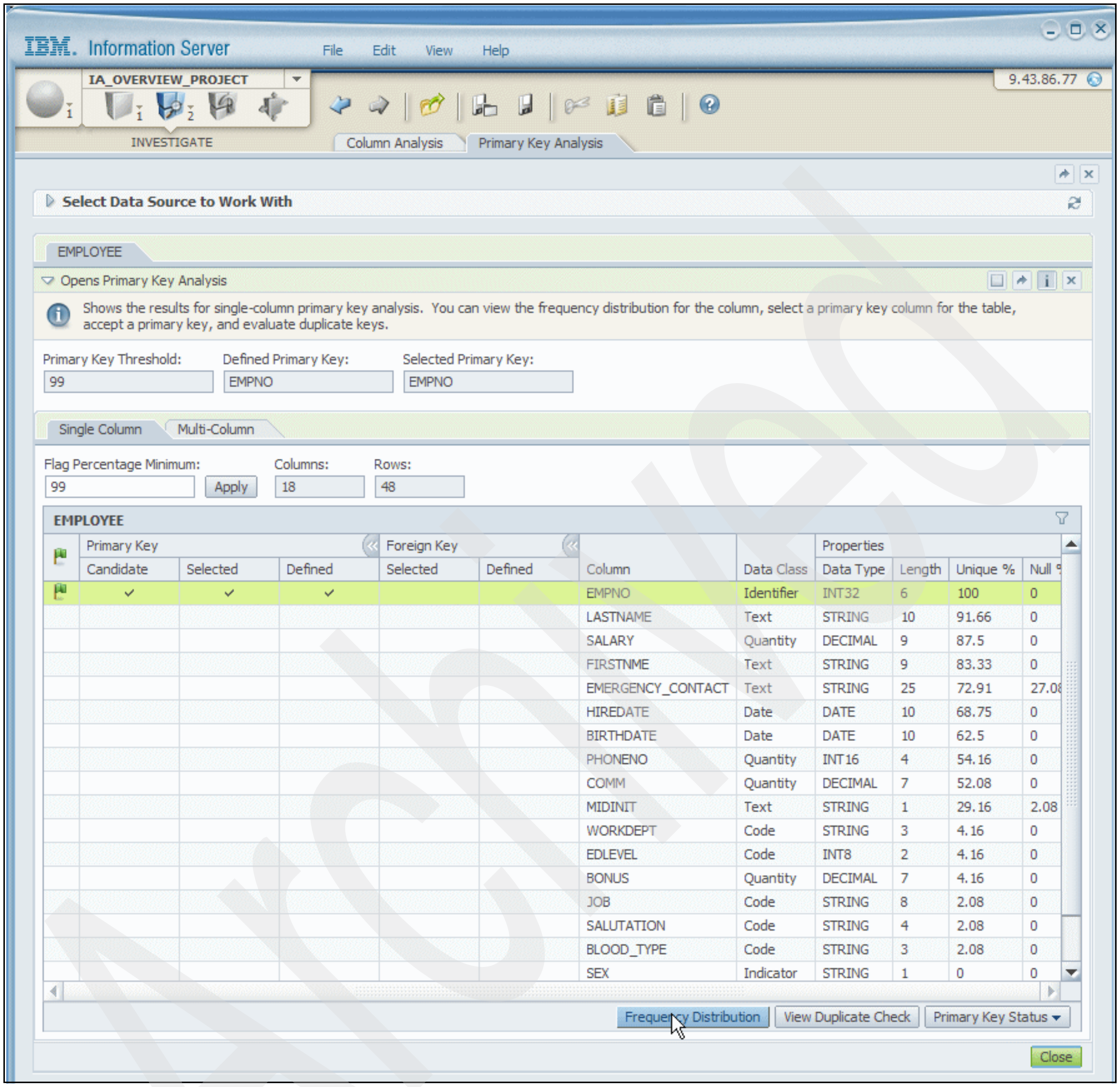
Total Records

	Records	%
Unique	44	91.66
Duplicate	4	8.33
Nulls	0	0

Duplicates

PK Value	Records	%
MEHTA	2	4.16
SMITH	2	4.16

Figure 1-175 Single column primary key analysis 3/7



Select Data Source to Work With

EMPLOYEE

Opens Primary Key Analysis

Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 99

Defined Primary Key: EMPNO

Selected Primary Key: EMPNO

Flag Percentage Minimum: 99

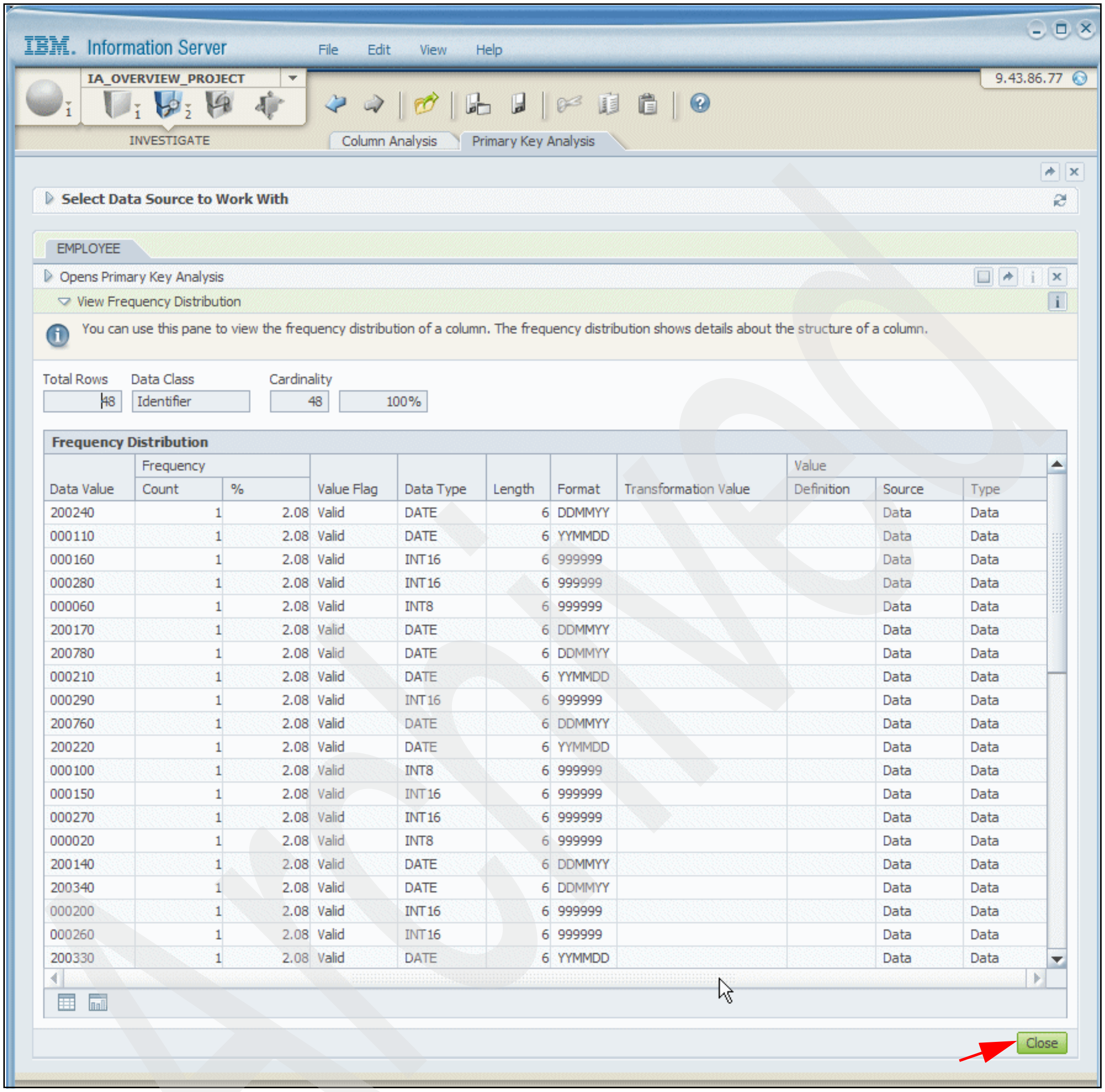
Columns: 18

Rows: 48

EMPLOYEE

	Primary Key			Foreign Key				Properties			
	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null %
⚑	✓	✓	✓			EMPNO	Identifier	INT32	6	100	0
						LASTNAME	Text	STRING	10	91.66	0
						SALARY	Quantity	DECIMAL	9	87.5	0
						FIRSTNME	Text	STRING	9	83.33	0
						EMERGENCY_CONTACT	Text	STRING	25	72.91	27.08
						HIREDATE	Date	DATE	10	68.75	0
						BIRTHDATE	Date	DATE	10	62.5	0
						PHONENO	Quantity	INT16	4	54.16	0
						COMM	Quantity	DECIMAL	7	52.08	0
						MIDINIT	Text	STRING	1	29.16	2.08
						WORKDEPT	Code	STRING	3	4.16	0
						EDLEVEL	Code	INT8	2	4.16	0
						BONUS	Quantity	DECIMAL	7	4.16	0
						JOB	Code	STRING	8	2.08	0
						SALUTATION	Code	STRING	4	2.08	0
						BLOOD_TYPE	Code	STRING	3	2.08	0
						SEX	Indicator	STRING	1	0	0

Figure 1-176 Single column primary key analysis 4/7



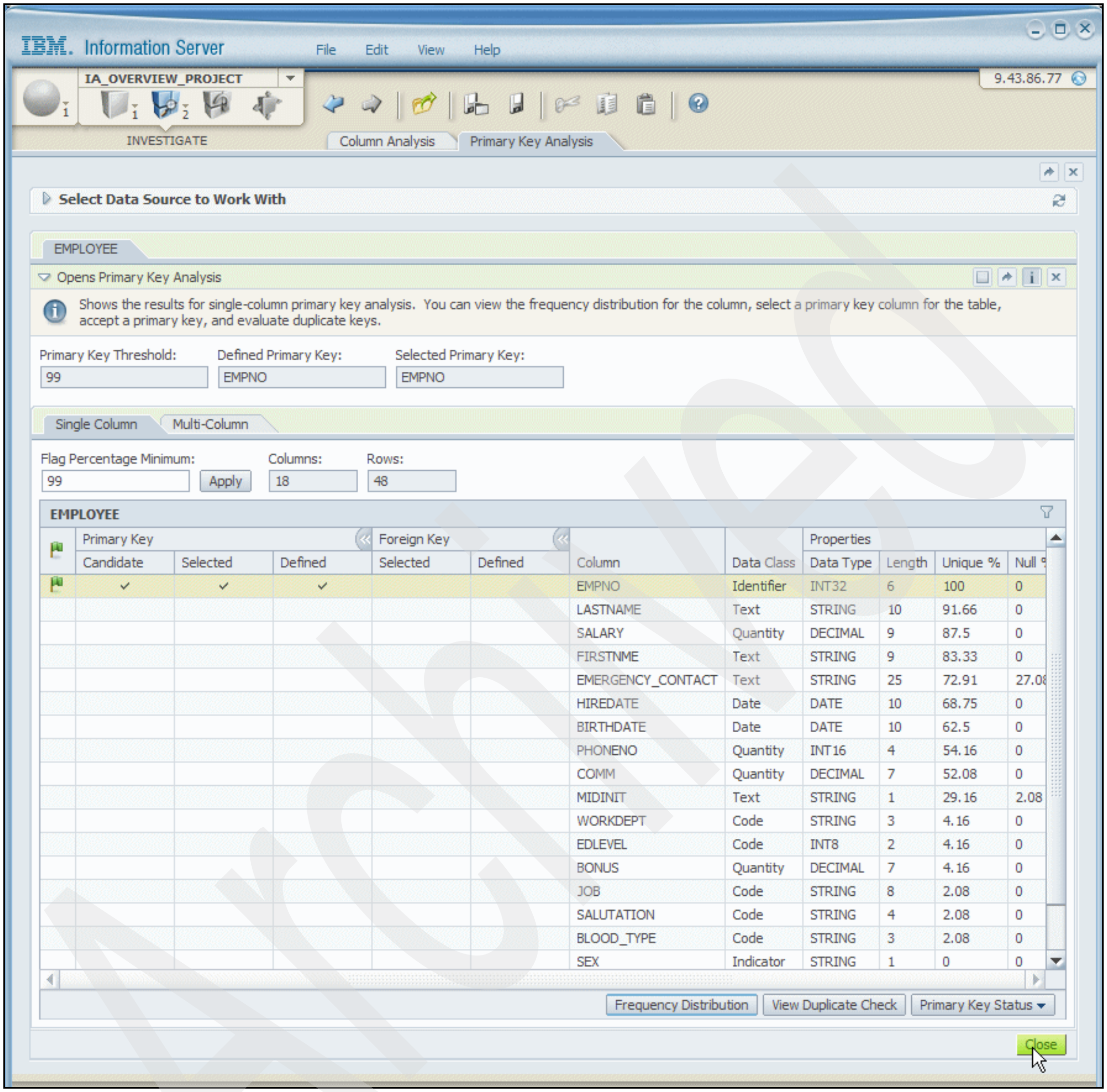
You can use this pane to view the frequency distribution of a column. The frequency distribution shows details about the structure of a column.

Total Rows: 48
Data Class: Identifier
Cardinality: 48 100%

Frequency Distribution

	Frequency							Value		
Data Value	Count	%	Value Flag	Data Type	Length	Format	Transformation Value	Definition	Source	Type
200240	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
000110	1	2.08	Valid	DATE	6	YYMMDD			Data	Data
000160	1	2.08	Valid	INT16	6	999999			Data	Data
000280	1	2.08	Valid	INT16	6	999999			Data	Data
000060	1	2.08	Valid	INT8	6	999999			Data	Data
200170	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
200780	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
000210	1	2.08	Valid	DATE	6	YYMMDD			Data	Data
000290	1	2.08	Valid	INT16	6	999999			Data	Data
200760	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
200220	1	2.08	Valid	DATE	6	YYMMDD			Data	Data
000100	1	2.08	Valid	INT8	6	999999			Data	Data
000150	1	2.08	Valid	INT16	6	999999			Data	Data
000270	1	2.08	Valid	INT16	6	999999			Data	Data
000020	1	2.08	Valid	INT8	6	999999			Data	Data
200140	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
200340	1	2.08	Valid	DATE	6	DDMMYY			Data	Data
000200	1	2.08	Valid	INT16	6	999999			Data	Data
000260	1	2.08	Valid	INT16	6	999999			Data	Data
200330	1	2.08	Valid	DATE	6	YYMMDD			Data	Data

Figure 1-177 Single column primary key analysis 5/7



Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 99

Defined Primary Key: EMPNO

Selected Primary Key: EMPNO

Flag Percentage Minimum: 99 Apply

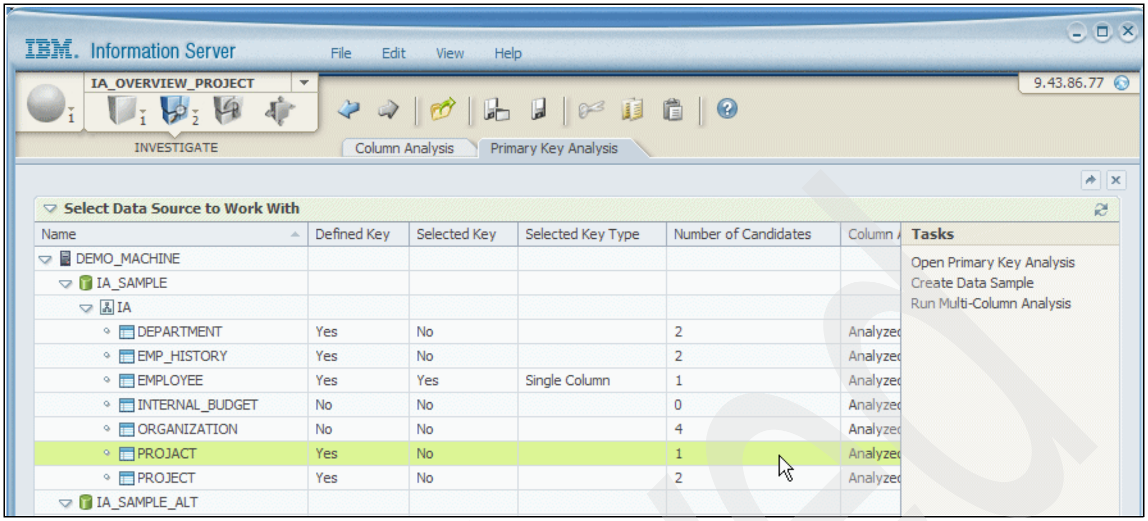
Columns: 18

Rows: 48

EMPLOYEE

	Primary Key			Foreign Key				Properties			
	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null
⚑	✓	✓	✓			EMPNO	Identifier	INT32	6	100	0
						LASTNAME	Text	STRING	10	91.66	0
						SALARY	Quantity	DECIMAL	9	87.5	0
						FIRSTNME	Text	STRING	9	83.33	0
						EMERGENCY_CONTACT	Text	STRING	25	72.91	27.08
						HIREDATE	Date	DATE	10	68.75	0
						BIRTHDATE	Date	DATE	10	62.5	0
						PHONENO	Quantity	INT16	4	54.16	0
						COMM	Quantity	DECIMAL	7	52.08	0
						MIDINIT	Text	STRING	1	29.16	2.08
						WORKDEPT	Code	STRING	3	4.16	0
						EDLEVEL	Code	INT8	2	4.16	0
						BONUS	Quantity	DECIMAL	7	4.16	0
						JOB	Code	STRING	8	2.08	0
						SALUTATION	Code	STRING	4	2.08	0
						BLOOD_TYPE	Code	STRING	3	2.08	0
						SEX	Indicator	STRING	1	0	0

Figure 1-178 Single column primary key analysis 6/7



Name	Defined Key	Selected Key	Selected Key Type	Number of Candidates	Column /
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT	Yes	No		2	Analyzec
EMP_HISTORY	Yes	No		2	Analyzec
EMPLOYEE	Yes	Yes	Single Column	1	Analyzec
INTERNAL_BUDGET	No	No		0	Analyzec
ORGANIZATION	No	No		4	Analyzec
PROJACT	Yes	No		1	Analyzec
PROJECT	Yes	No		2	Analyzec
IA_SAMPLE_ALT					

Tasks

Open Primary Key Analysis
Create Data Sample
Run Multi-Column Analysis

Figure 1-179 Single column primary key analysis 7/7

Multi-column defined primary key analysis

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), assess primary key analysis on the EMPLOYEE table as follows:

1. On the Investigate navigator menu in the console, select **Primary Key Analysis** as shown in Figure 1-172 on page 231.
2. On the Primary Key Analysis workspace, select the PROJACT table and click **Open Primary Key Analysis** in the Tasks pane as shown in Figure 1-179 on page 238.
3. Select the Single Column tab to view the analysis results for all the columns in the PROJACT table as shown in Figure 1-180 on page 240. The SQL070725170745310 column (which is an artificial representation of a defined multi-column primary key) shows 100% unique. It is a system created virtual column when the data source is registered to the project and the table has a defined multi-column primary key. It is also the defined, selected and candidate key.
4. Select the Multi-Column tab to discover potential multi-column primary key candidates. Figure 1-181 on page 241 shows the three column (PROJNO,ACTNO,DEPTNO) primary key that is defined, selected and a candidate. To check for duplicates, you need to click **Duplicate Check** → **Run Duplicate Check** which will access the underlying data source.

> **Note:** This is different from the View Duplicate Check which uses the information in the column analysis to display duplicate information. It does *not* change the primary key definitions in the table—it only saves the changes in the metadata repository. If you want to change the data itself, you must make the change in the external database and then rerun column analysis to see the data changes.

After selecting the columns, you can choose one of the following options:

- To view the frequency distribution (clicking **View Frequency Distribution**).
- To accept or remove the primary key (clicking **Primary Key Status** → **Accept Primary Key** or **Primary Key Status** → **Remove Primary Key**).
- To remove all multi-column primary key (clicking **Remove All MCPK**).
- To request analysis of multiple columns (clicking **Analyze Multiple Columns**).

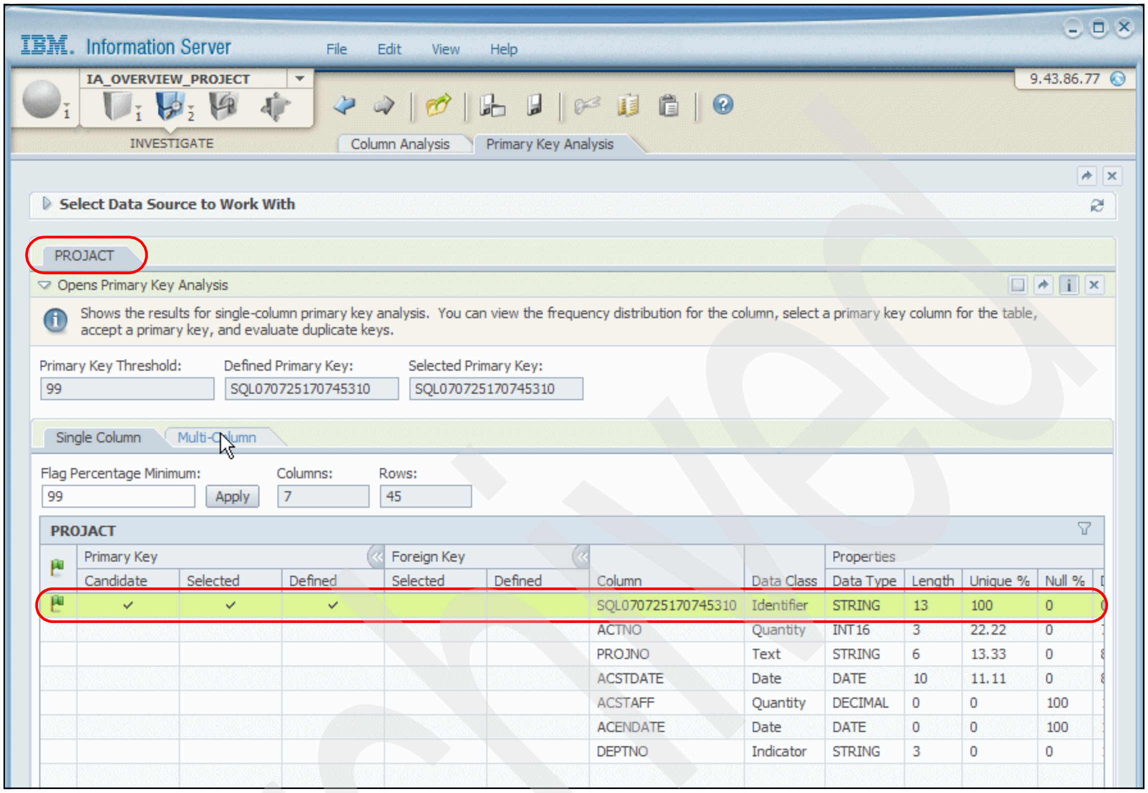
We discuss some of these options later.

5. Provide a Job Name (Duplicate Check Analysis Task), verify the columns (PROJNO,ACTNO,DEPTNO) in Column Sequence, and click **Submit** as shown in Figure 1-182 on page 242.
6. After the job has completed successfully, you can view the result by selecting the 3 column primary key and clicking **View Duplicate Check** shown in Figure 1-183 on page 243.
7. Figure 1-184 on page 244 confirms that this 3 column primary key only has unique values. You can choose to remove the primary key in the metadata repository by clicking **Primary Key Status** → **Remove Primary Key**, which we did not do. Click **Close**.

> **Attention:** Changing the primary key status does *not* change the primary key definitions in the table—it only saves the changes in the metadata repository. If you want to change the data itself, you must make the change in the external database, and then rerun column analysis to see the data changes.

8. Click **Frequency Distribution** in the analysis results summary in Figure 1-185 on page 245 to view the frequency distribution of the 3 column primary.
9. Figure 1-186 on page 246 shows the details about the structure of the (PROJNO,ACTNO,DEPTNO) column. Click **Close** in Figure 1-186 on page 246 to proceed to the Primary Key Analysis workspace shown in

Figure 1-187 on page 247, which shows the PROJACT table as having a multi-column Selected Key.



Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 99

Defined Primary Key: SQL070725170745310

Selected Primary Key: SQL070725170745310

Flag Percentage Minimum: 99

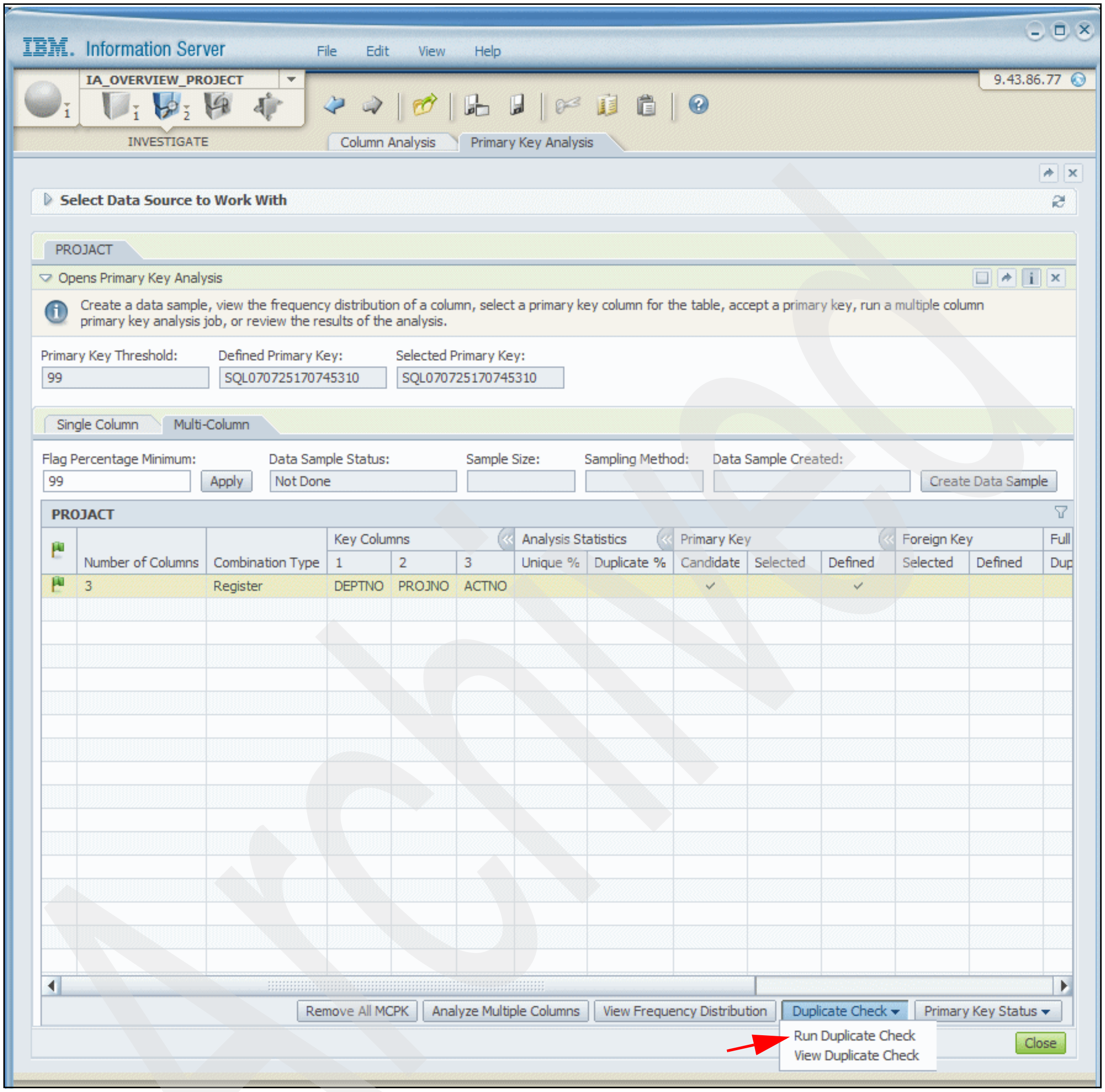
Columns: 7

Rows: 45

PROJACT

	Primary Key			Foreign Key				Properties			
	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null %
	✓	✓	✓			SQL070725170745310	Identifier	STRING	13	100	0
						ACTNO	Quantity	INT16	3	22.22	0
						PROJNO	Text	STRING	6	13.33	0
						ACSTDATE	Date	DATE	10	11.11	0
						ACSTAFF	Quantity	DECIMAL	0	0	100
						ACENDATE	Date	DATE	0	0	100
						DEPTNO	Indicator	STRING	3	0	0

Figure 1-180 Multi-column primary key analysis 1/8



Create a data sample, view the frequency distribution of a column, select a primary key column for the table, accept a primary key, run a multiple column primary key analysis job, or review the results of the analysis.

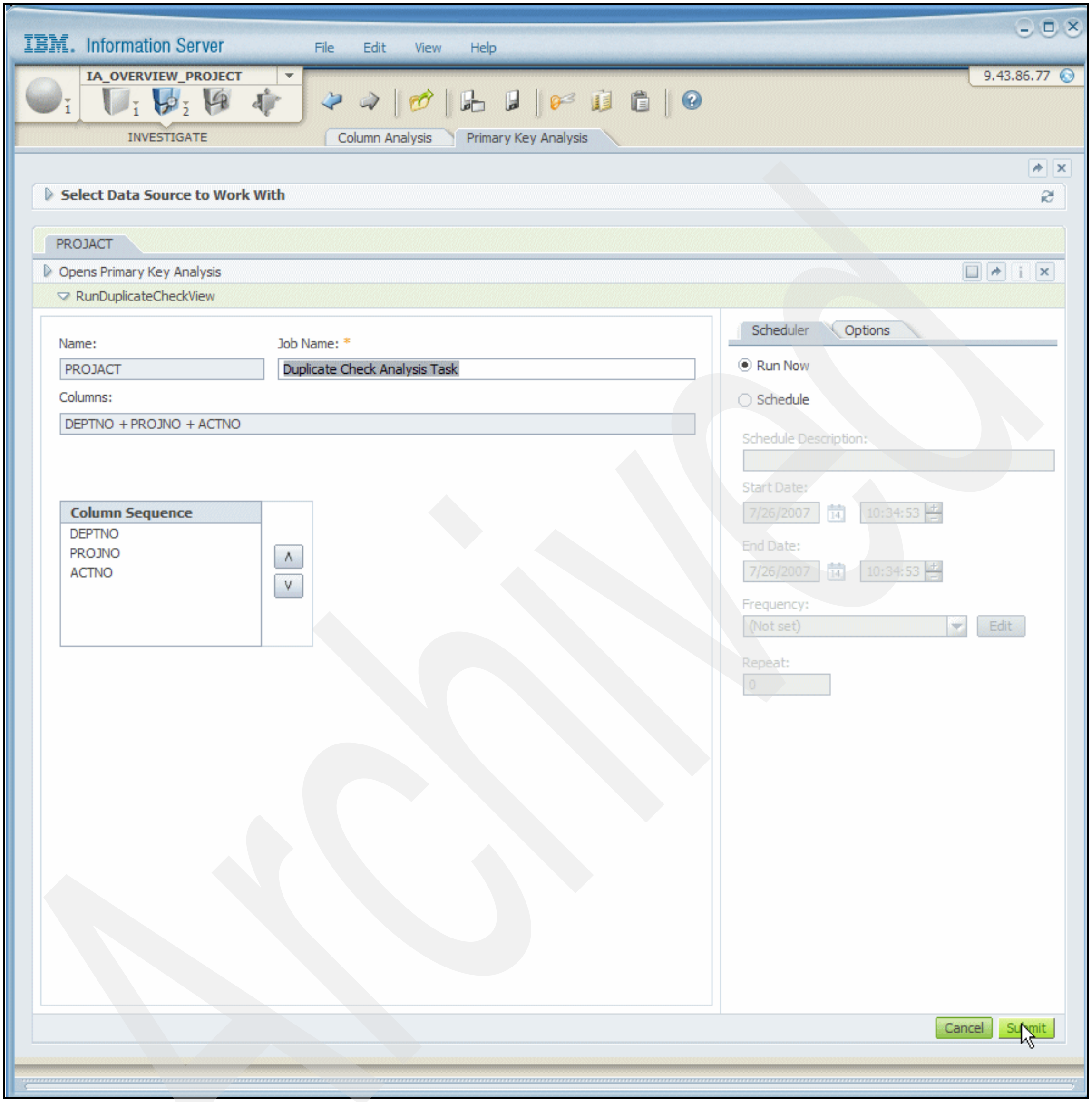
Primary Key Threshold: 99
Defined Primary Key: SQL070725170745310
Selected Primary Key: SQL070725170745310

Flag Percentage Minimum: 99
Data Sample Status: Not Done
Sample Size:
Sampling Method:
Data Sample Created:

PROJACT

			Key Columns			Analysis Statistics		Primary Key			Foreign Key		Full
	Number of Columns	Combination Type	1	2	3	Unique %	Duplicate %	Candidate	Selected	Defined	Selected	Defined	Dup
	3	Register	DEPTNO	PROJNO	ACTNO			✓		✓			

Figure 1-181 Multi-column primary key analysis 2/8



Name: PROJACT

Job Name: * Duplicate Check Analysis Task

Columns: DEPTNO + PROJNO + ACTNO

Column Sequence: DEPTNO, PROJNO, ACTNO

● Run Now

○ Schedule

Schedule Description:

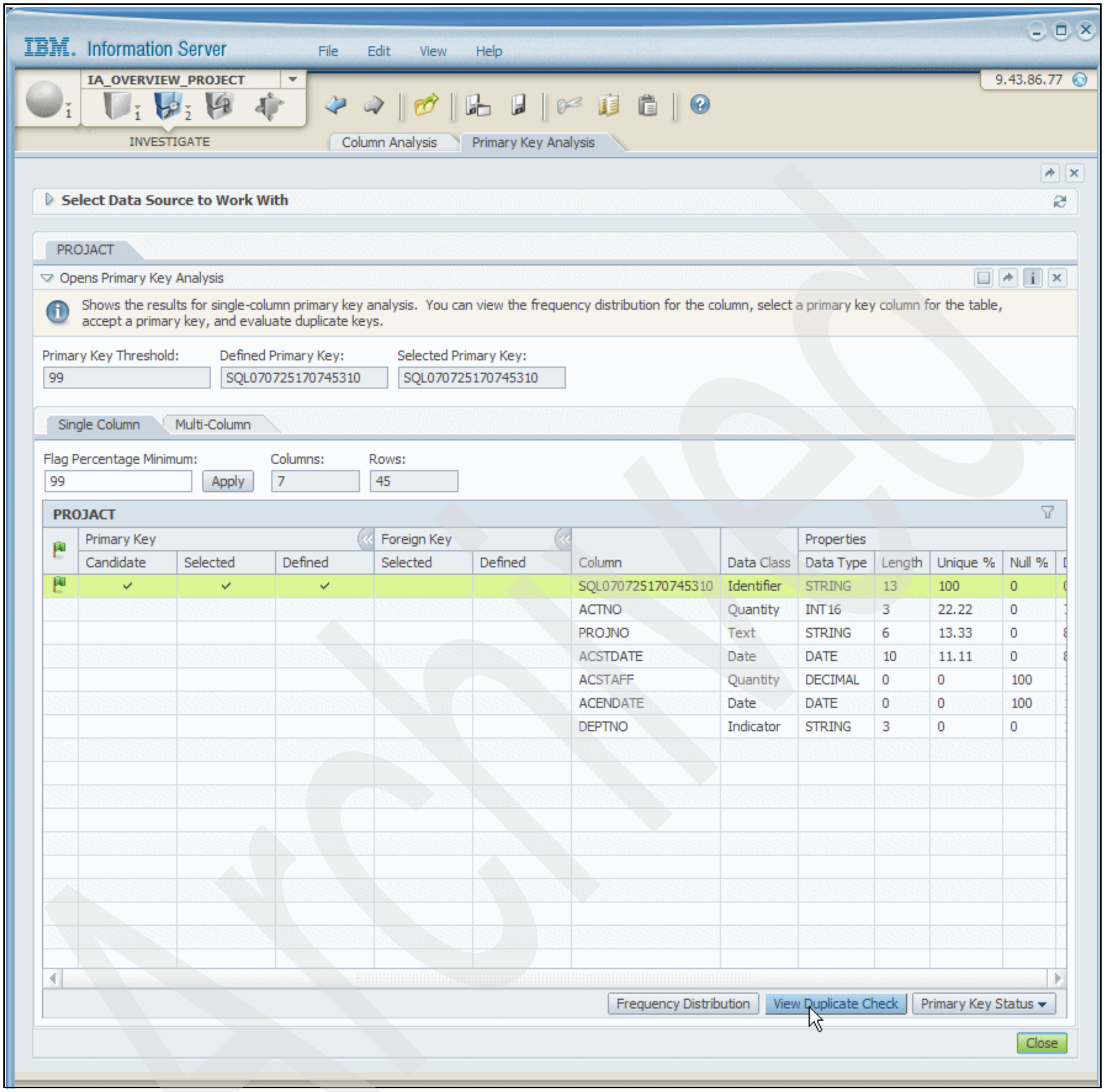
Start Date: 7/26/2007 10:34:53

End Date: 7/26/2007 10:34:53

Frequency: (Not set) Edit

Repeat: 0

Figure 1-182 Multi-column primary key analysis 3/8



Select Data Source to Work With

PROJACT

Opens Primary Key Analysis

Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 99

Defined Primary Key: SQL070725170745310

Selected Primary Key: SQL070725170745310

Flag Percentage Minimum: 99 Apply

Columns: 7

Rows: 45

PROJACT

	Primary Key			Foreign Key				Properties			
	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null %
⚑	✓	✓	✓			SQL070725170745310	Identifier	STRING	13	100	0
						ACTNO	Quantity	INT16	3	22.22	0
						PROJNO	Text	STRING	6	13.33	0
						ACSTDATE	Date	DATE	10	11.11	0
						ACSTAFF	Quantity	DECIMAL	0	0	100
						ACENDATE	Date	DATE	0	0	100
						DEPTNO	Indicator	STRING	3	0	0

Figure 1-183 Multi-column primary key analysis 4/8

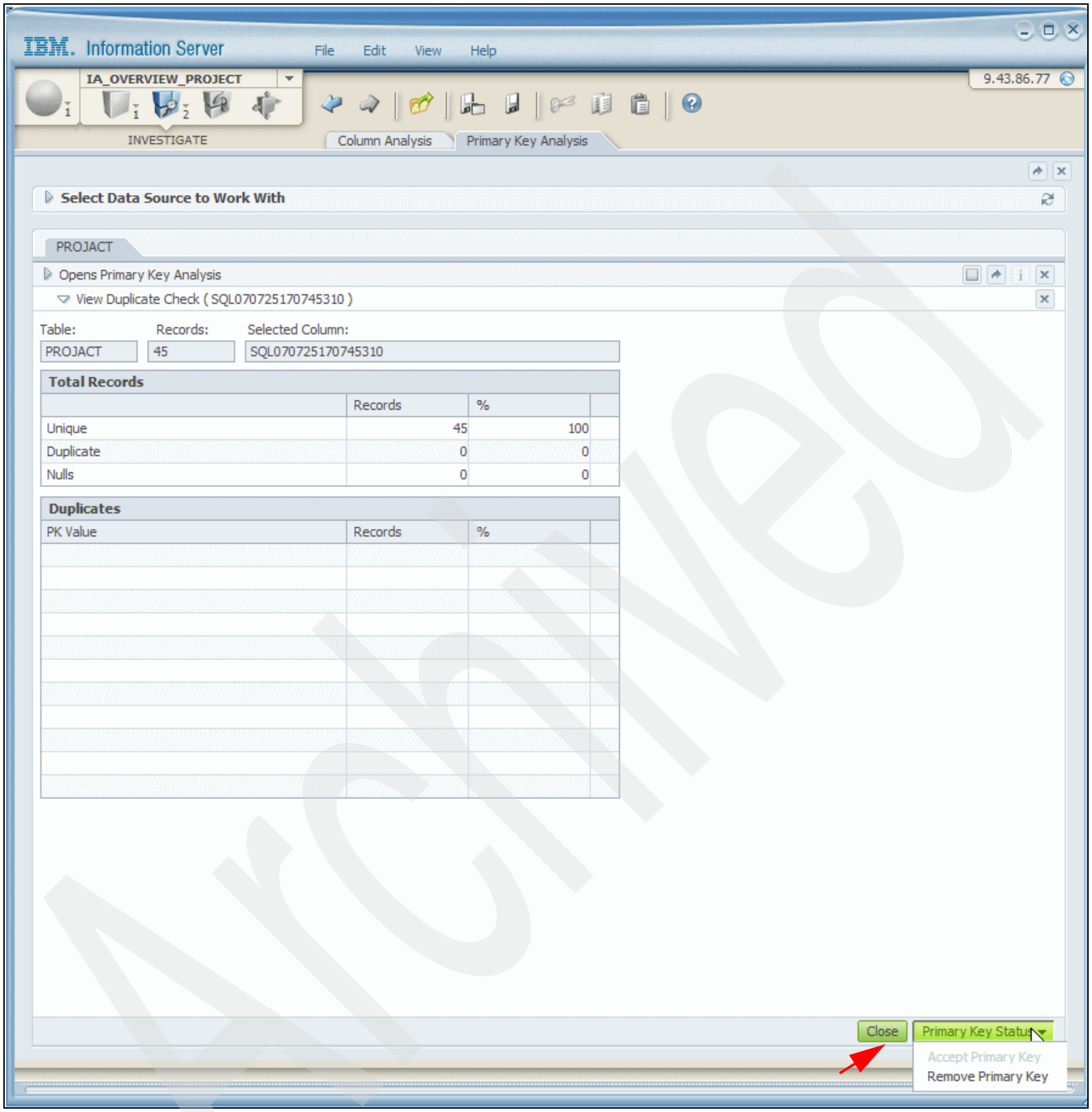


Table: PROJACT

Records: 45

Selected Column: SQL070725170745310

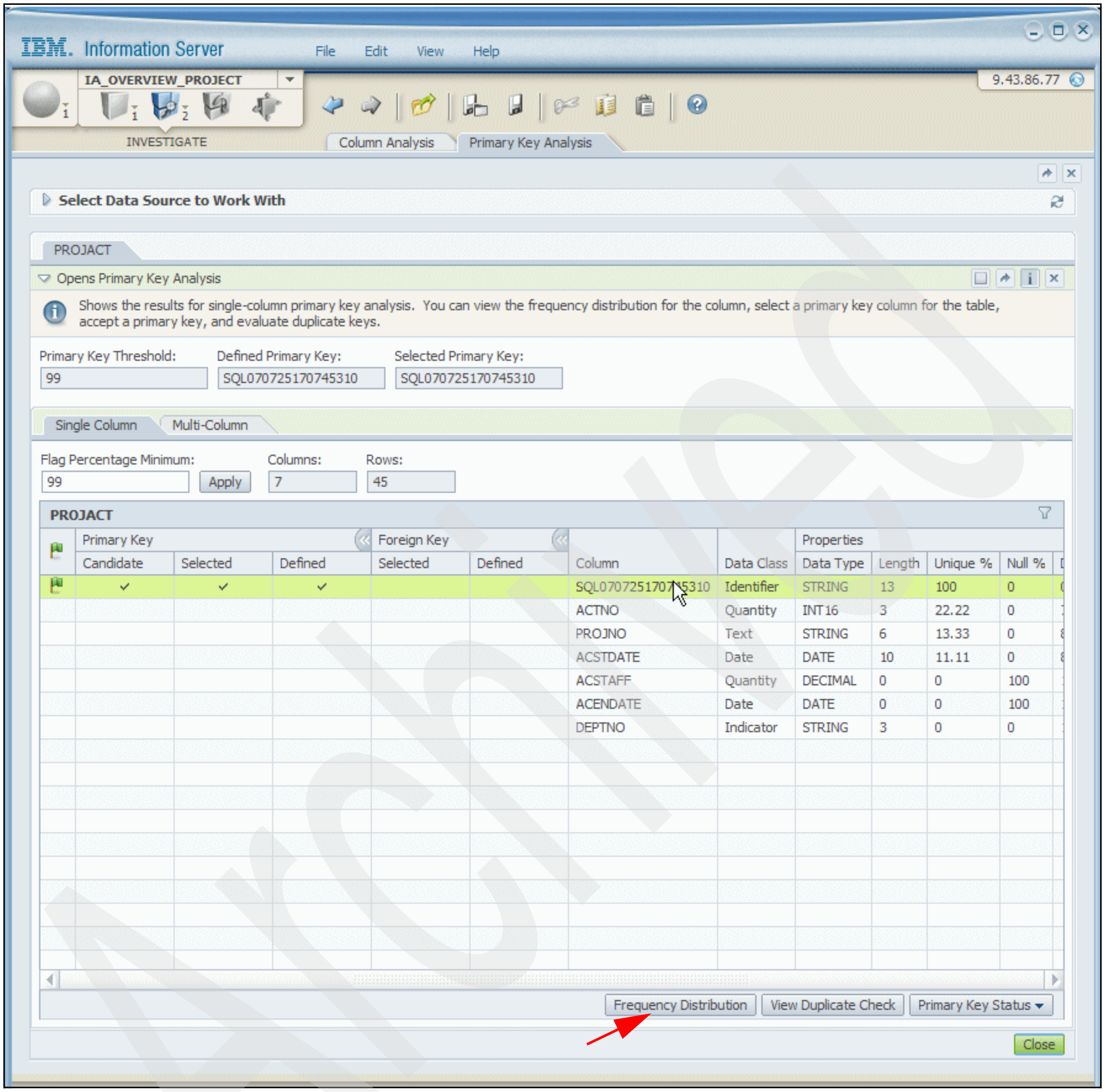
Total Records

	Records	%
Unique	45	100
Duplicate	0	0
Nulls	0	0

Duplicates

PK Value	Records	%

Figure 1-184 Multi-column primary key analysis 5/8



Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 99

Defined Primary Key: SQL070725170745310

Selected Primary Key: SQL070725170745310

Flag Percentage Minimum: 99

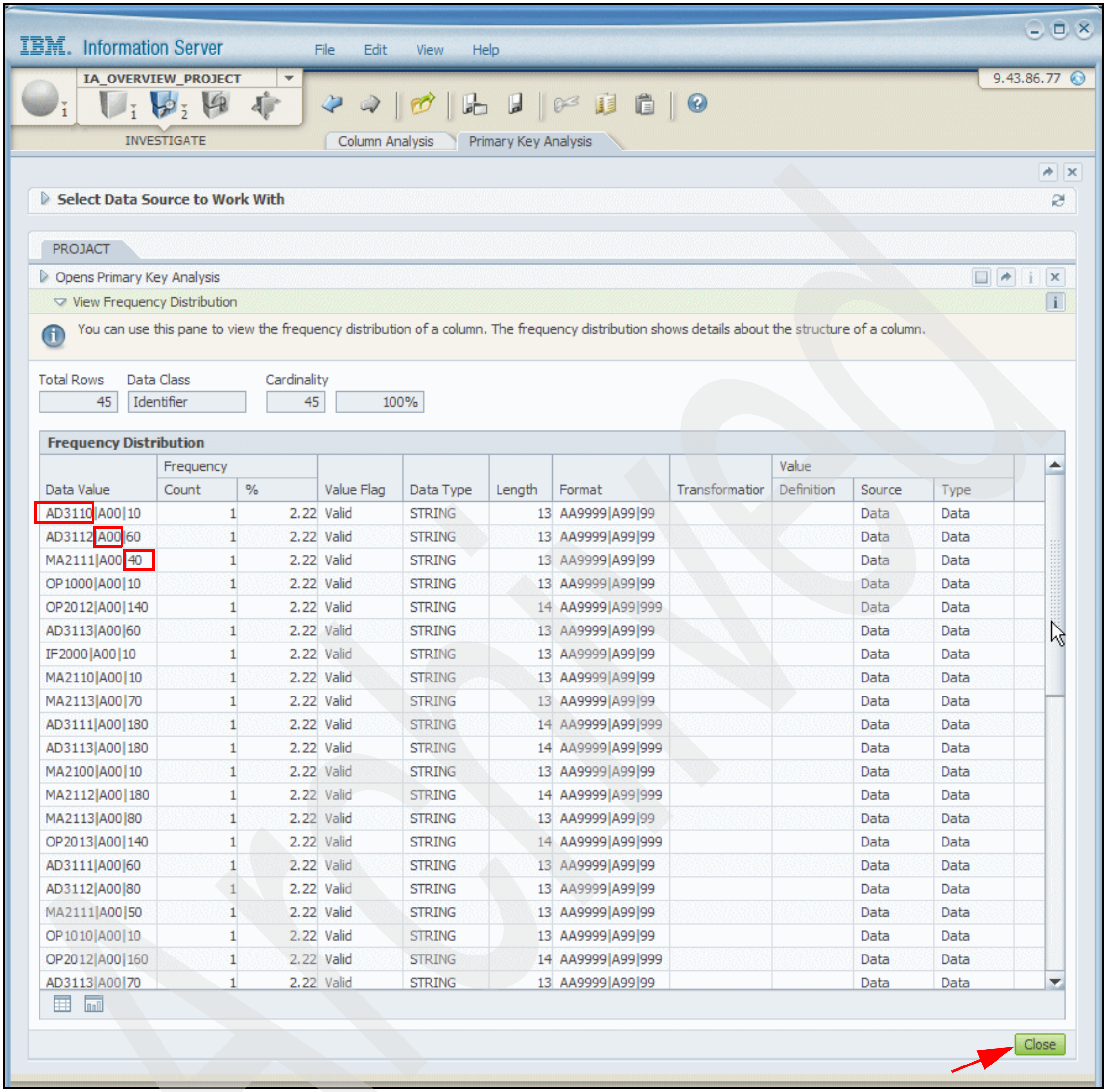
Columns: 7

Rows: 45

PROJACT

	Primary Key			Foreign Key				Properties			
	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null %
🟩	✓	✓	✓			SQL0707251707 5310	Identifier	STRING	13	100	0
						ACTNO	Quantity	INT16	3	22.22	0
						PROJNO	Text	STRING	6	13.33	0
						ACSTDATE	Date	DATE	10	11.11	0
						ACSTAFF	Quantity	DECIMAL	0	0	100
						ACENDATE	Date	DATE	0	0	100
						DEPTNO	Indicator	STRING	3	0	0

Figure 1-185 Multi-column primary key analysis 6/8



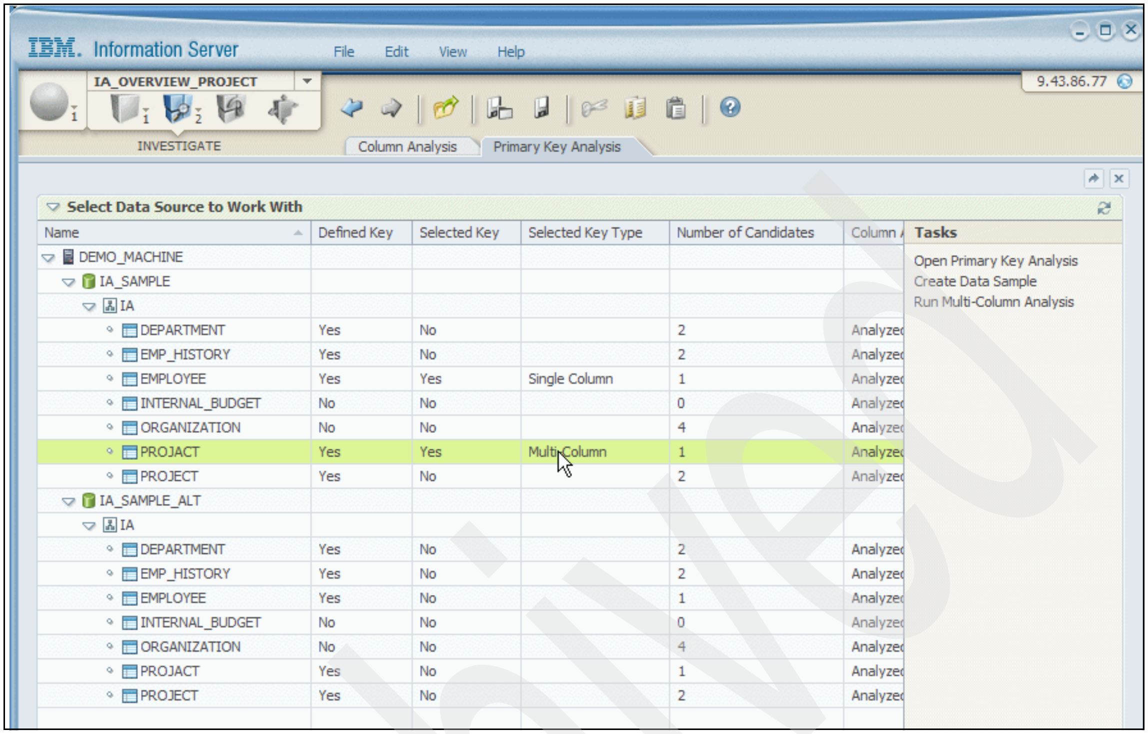
You can use this pane to view the frequency distribution of a column. The frequency distribution shows details about the structure of a column.

Total Rows: 45
Data Class: Identifier
Cardinality: 45 100%

Frequency Distribution

	Frequency							Value		
Data Value	Count	%	Value Flag	Data Type	Length	Format	Transformatior	Definition	Source	Type
AD3110|A00|10	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
AD3112|A00|60	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
MA2111|A00|40	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
OP1000|A00|10	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
OP2012|A00|140	1	2.22	Valid	STRING	14	AA9999|A99|999			Data	Data
AD3113|A00|60	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
IF2000|A00|10	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
MA2110|A00|10	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
MA2113|A00|70	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
AD3111|A00|180	1	2.22	Valid	STRING	14	AA9999|A99|999			Data	Data
AD3113|A00|180	1	2.22	Valid	STRING	14	AA9999|A99|999			Data	Data
MA2100|A00|10	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
MA2112|A00|180	1	2.22	Valid	STRING	14	AA9999|A99|999			Data	Data
MA2113|A00|80	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
OP2013|A00|140	1	2.22	Valid	STRING	14	AA9999|A99|999			Data	Data
AD3111|A00|60	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
AD3112|A00|80	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
MA2111|A00|50	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
OP1010|A00|10	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data
OP2012|A00|160	1	2.22	Valid	STRING	14	AA9999|A99|999			Data	Data
AD3113|A00|70	1	2.22	Valid	STRING	13	AA9999|A99|99			Data	Data

Figure 1-186 Multi-column primary key analysis 7/8



Name	Defined Key	Selected Key	Selected Key Type	Number of Candidates	Column A
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT	Yes	No		2	Analyzed
EMP_HISTORY	Yes	No		2	Analyzed
EMPLOYEE	Yes	Yes	Single Column	1	Analyzed
INTERNAL_BUDGET	No	No		0	Analyzed
ORGANIZATION	No	No		4	Analyzed
PROJACT	Yes	Yes	Multi-Column	1	Analyzed
PROJECT	Yes	No		2	Analyzed
IA_SAMPLE_ALT					
IA					
DEPARTMENT	Yes	No		2	Analyzed
EMP_HISTORY	Yes	No		2	Analyzed
EMPLOYEE	Yes	No		1	Analyzed
INTERNAL_BUDGET	No	No		0	Analyzed
ORGANIZATION	No	No		4	Analyzed
PROJACT	Yes	No		1	Analyzed
PROJECT	Yes	No		2	Analyzed

Tasks

Open Primary Key Analysis

Create Data Sample

Run Multi-Column Analysis

Figure 1-187 Multi-column primary key analysis 8/8

Undefined single column primary key analysis

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), run a primary key analysis job on the ORGANIZATION table (which has no primary key defined) as follows:

1. On the Investigate navigator menu in the console, select **Primary Key Analysis** as shown in Figure 1-172 on page 231.
2. On the Primary Key Analysis workspace, select the ORGANIZATION table and click **Open Primary Key Analysis** in the Tasks pane as shown in Figure 1-188 on page 248.
3. Select the Single Column tab to view the analysis results for all the columns in the ORGANIZATION table as shown in Figure 1-189 on page 249. The ORG_ID column shows 100% unique.

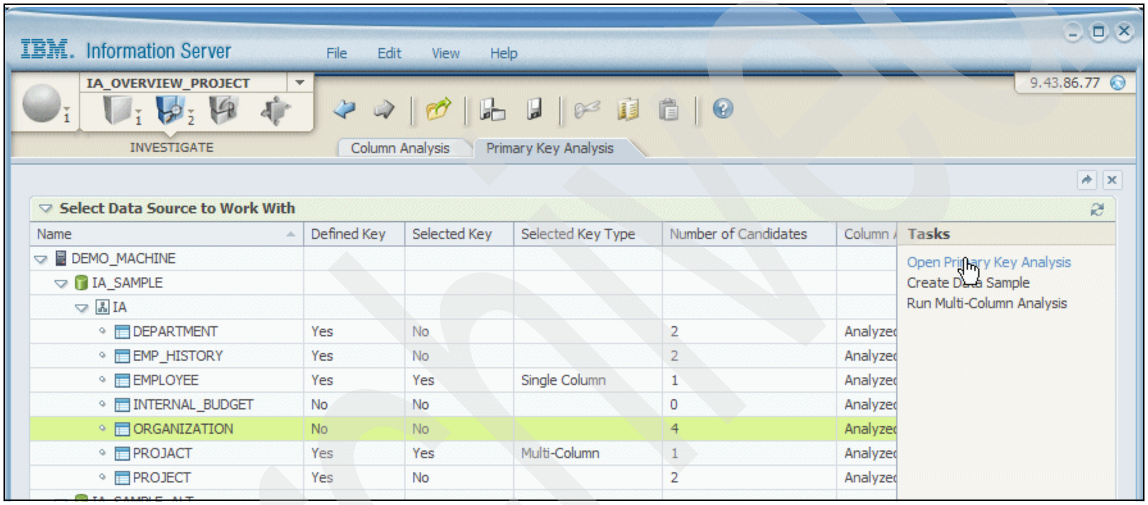
   Accept this column as a primary key by clicking **Primary Key Status** → **Accept Primary Key**. This is not shown here.

   If the selected column is not 100% unique, click **View Duplicate Check** to view the duplicate data values within the selected column as described earlier in Figure 1-183 on page 243. This is not shown here.

4. Select ORG_ID (which shows it as being *Selected* after accepting this column as a primary key) and click **Frequency Distribution** in the analysis results summary in Figure 1-189 on page 249 to view the frequency distribution of the selected ORG_ID column.

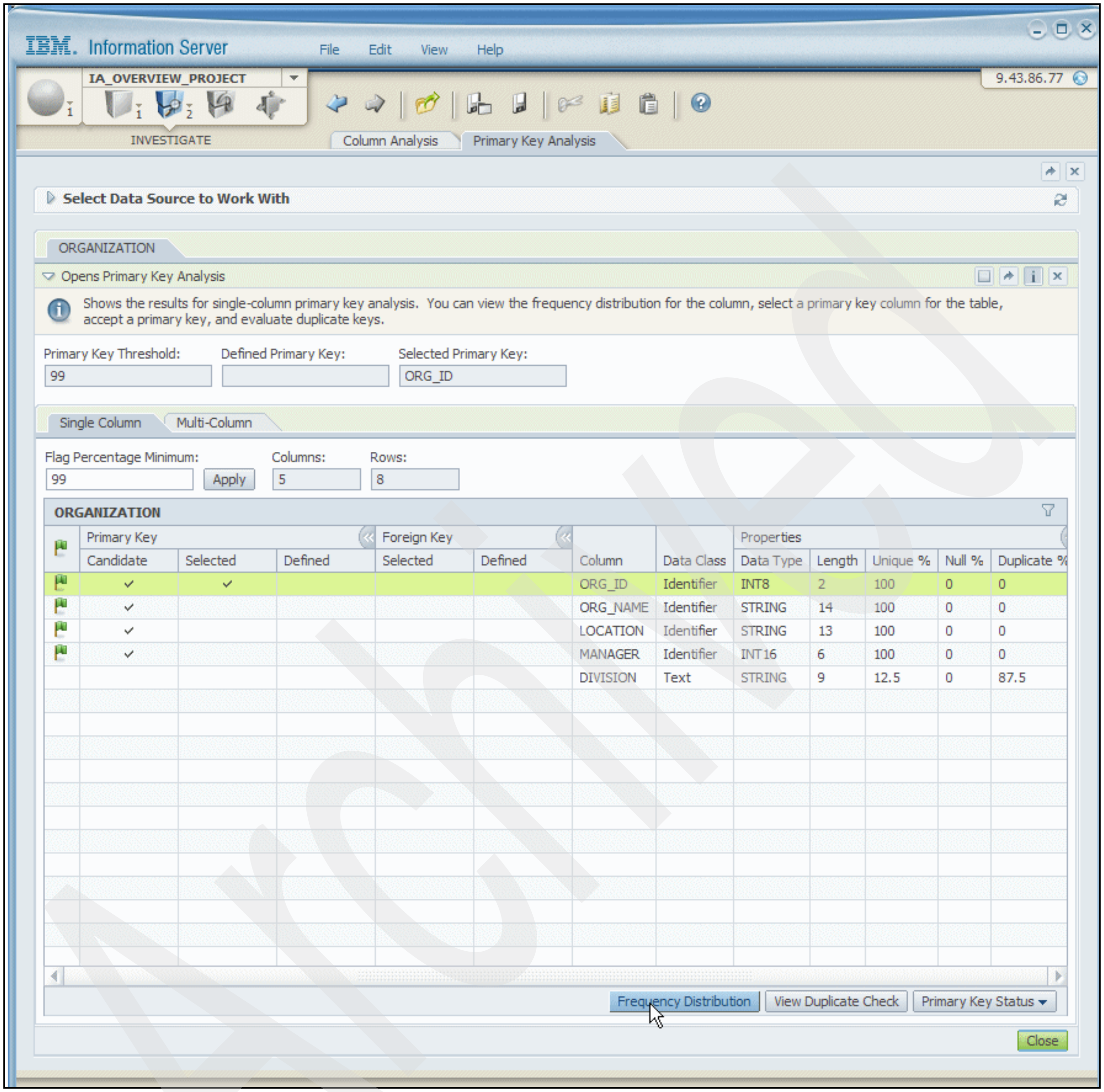
5. Figure 1-190 on page 250 shows the details about the structure of the ORG_ID column. Click the icon to view the frequency distribution in bar format as shown in Figure 1-191 on page 251.

6. Click **Close** in Figure 1-191 on page 251 and again in the analysis results summary (not shown here) to proceed to the Primary Key Analysis workspace shown in Figure 1-192 on page 254, which shows the ORGANIZATION table as having a single column Selected Key.



Name	Defined Key	Selected Key	Selected Key Type	Number of Candidates	Column A
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT	Yes	No		2	Analyzed
EMP_HISTORY	Yes	No		2	Analyzed
EMPLOYEE	Yes	Yes	Single Column	1	Analyzed
INTERNAL_BUDGET	No	No		0	Analyzed
ORGANIZATION	No	No		4	Analyzed
PROJACT	Yes	Yes	Multi-Column	1	Analyzed
PROJECT	Yes	No		2	Analyzed

Figure 1-188 Single column analysis on table with no defined primary key 1/4



Select Data Source to Work With

ORGANIZATION

Opens Primary Key Analysis

Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 99

Defined Primary Key:

Selected Primary Key: ORG_ID

Flag Percentage Minimum: 99 Apply

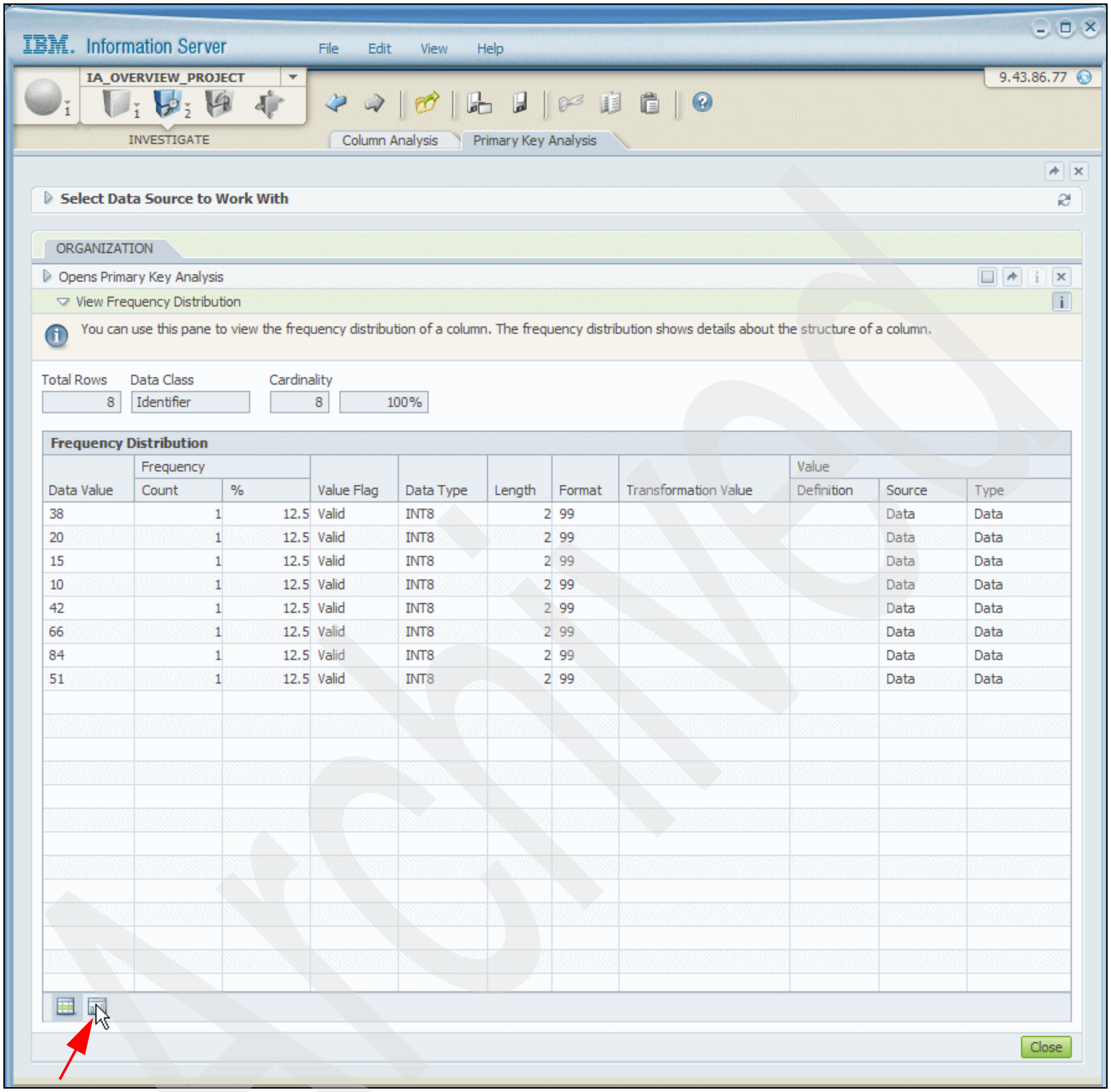
Columns: 5

Rows: 8

ORGANIZATION

	Primary Key			Foreign Key				Properties				
	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null %	Duplicate %
🟩	✓	✓				ORG_ID	Identifier	INT8	2	100	0	0
🟩	✓					ORG_NAME	Identifier	STRING	14	100	0	0
🟩	✓					LOCATION	Identifier	STRING	13	100	0	0
🟩	✓					MANAGER	Identifier	INT16	6	100	0	0
						DIVISION	Text	STRING	9	12.5	0	87.5

Figure 1-189 Single column analysis on table with no defined primary key 2/4



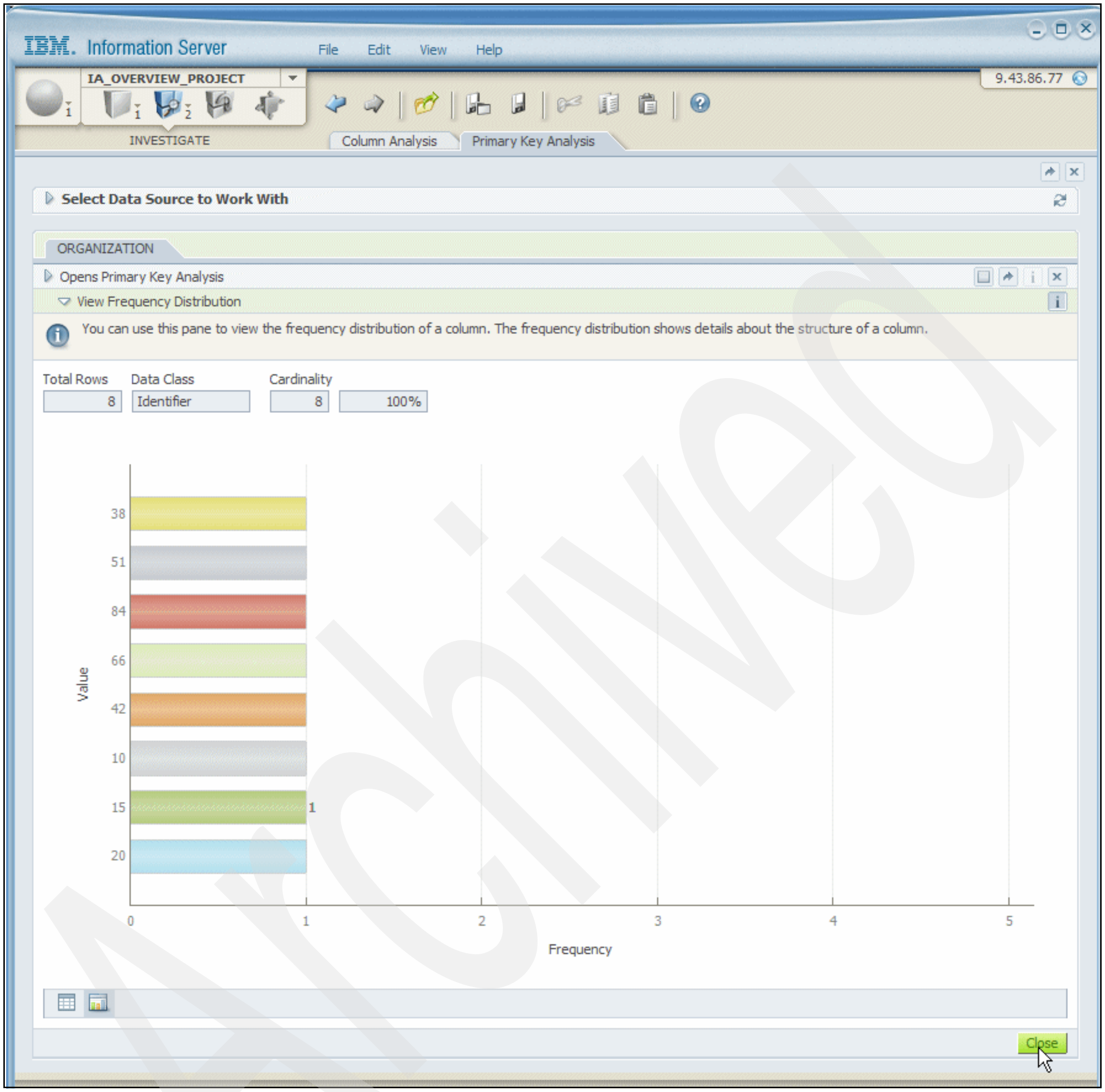
You can use this pane to view the frequency distribution of a column. The frequency distribution shows details about the structure of a column.

Total Rows: 8
Data Class: Identifier
Cardinality: 8 100%

Frequency Distribution

	Frequency							Value		
Data Value	Count	%	Value Flag	Data Type	Length	Format	Transformation Value	Definition	Source	Type
38	1	12.5	Valid	INT8	2	99			Data	Data
20	1	12.5	Valid	INT8	2	99			Data	Data
15	1	12.5	Valid	INT8	2	99			Data	Data
10	1	12.5	Valid	INT8	2	99			Data	Data
42	1	12.5	Valid	INT8	2	99			Data	Data
66	1	12.5	Valid	INT8	2	99			Data	Data
84	1	12.5	Valid	INT8	2	99			Data	Data
51	1	12.5	Valid	INT8	2	99			Data	Data

Figure 1-190 Single column analysis on table with no defined primary key 3/4



You can use this pane to view the frequency distribution of a column. The frequency distribution shows details about the structure of a column.

Total Rows	Data Class	Cardinality	
8	Identifier	8	100%

Figure 1-191 Single column analysis on table with no defined primary key 4/4

Undefined multi-column primary key analysis

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), run a primary key analysis job on the INTERNAL_BUDGET table as follows:

1. On the Investigate navigator menu in the console, select **Primary Key Analysis** as shown in Figure 1-172 on page 231.
2. On the Primary Key Analysis workspace, select the INTERNAL_BUDGET table and click **Open Primary Key Analysis** in the Tasks pane as shown in Figure 1-192 on page 254.
3. Select the Single Column tab to view the analysis results for all the columns in the INTERNAL_BUDGET table as shown in Figure 1-193 on page 255. There is no primary key candidate, defined or selected for this table, and none of the columns have a very high Unique percentage. This leads us to look at the possibility of selecting a multi-column primary key for this table.
4. Select the **Multi-Column** tab to discover potential multi-column primary key candidates, and click **Analyze Multiple Columns** in Figure 1-194 on page 256.
5. At the prompt to create a data sample, click **Analyze Full Table** as shown in Figure 1-195 on page 257. While the recommendation is to create a data sample, our data volumes were so low that it was appropriate to analyze all the rows in the table.

> **Attention:** When assessing tables with many columns, the potential number of column assessment that can be investigated grows on a factorial basis as the size of the composite maximum key combination is increased. If the likely multi-key combination is known or approximately known, select the probably Composite Key Threshold and *only* select the specific columns to assess. This can be run against either a data sample or full volume. If the multi-key combination is not known, then always start with a low Composite Key Threshold against a data sample. Raise the Composite Key Threshold as necessary against the sample to get more clarification. As the likely key combination emerges, reduce the columns selected in job to keep the number of data combinations minimized. Only after the likely key combination is identified should you run the Duplicate Check against the full volume of data.

6. In the RunMCAnalysisView in Figure 1-196 on page 258, modify the Composite Max field to a value of *4*, select all the columns in the table and click **Submit**. The Composite Max field corresponds to the Composite Key Threshold that has a default of *2* and a maximum value of *7*.

> **Note:** The combination of selecting large composite keys with many columns and large volumes of data will require extensive system resources.

7. Figure 1-197 on page 259 and Figure 1-198 on page 260 shows a partial list of the potential multi-column candidate keys that exceed the highlighted Primary Key Threshold of 99%.

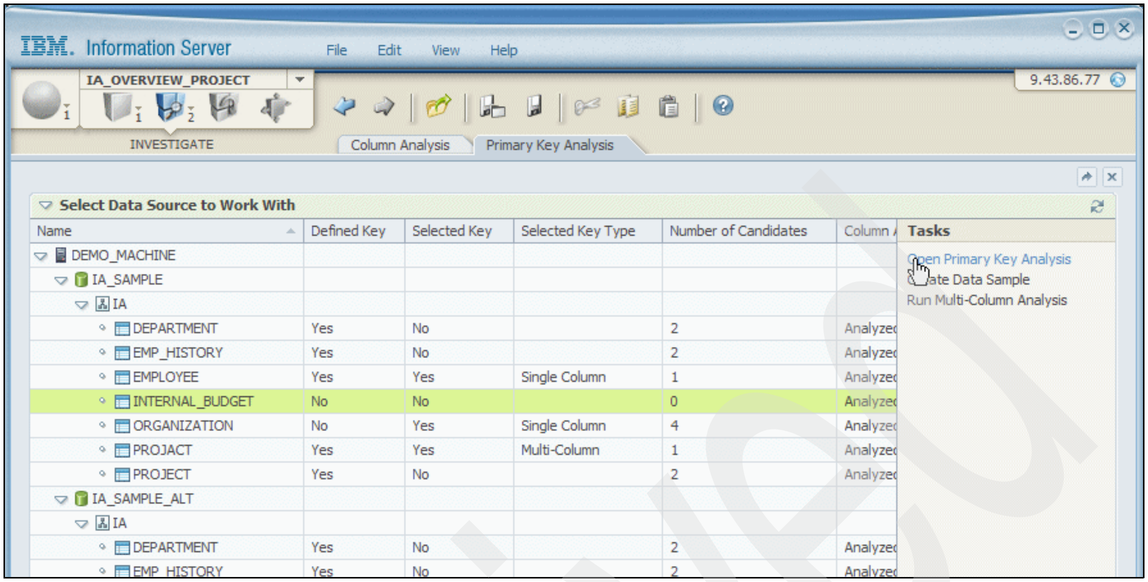
   The FUNDING_ACTNO, FUNDING_DEPTNO, FUNDING_PROJNO combination in Figure 1-197 on page 259 shows 100% unique values (as do some of the other combinations) and is considered the candidate to accept as the primary key. The acceptance of this set of columns as the primary key is not shown here.

8. Figure 1-199 on page 261 shows the 3 column (PROJNO,ACTNO,DEPTNO) primary key that is defined, selected and a candidate. Select the (PROJNO,ACTNO,DEPTNO) multi-column and click **View Frequency Distribution** to view the frequency distribution.

9. Figure 1-200 on page 262 and Figure 1-201 on page 263 show the tabular and bar chart of the partial list of values in the composite column.

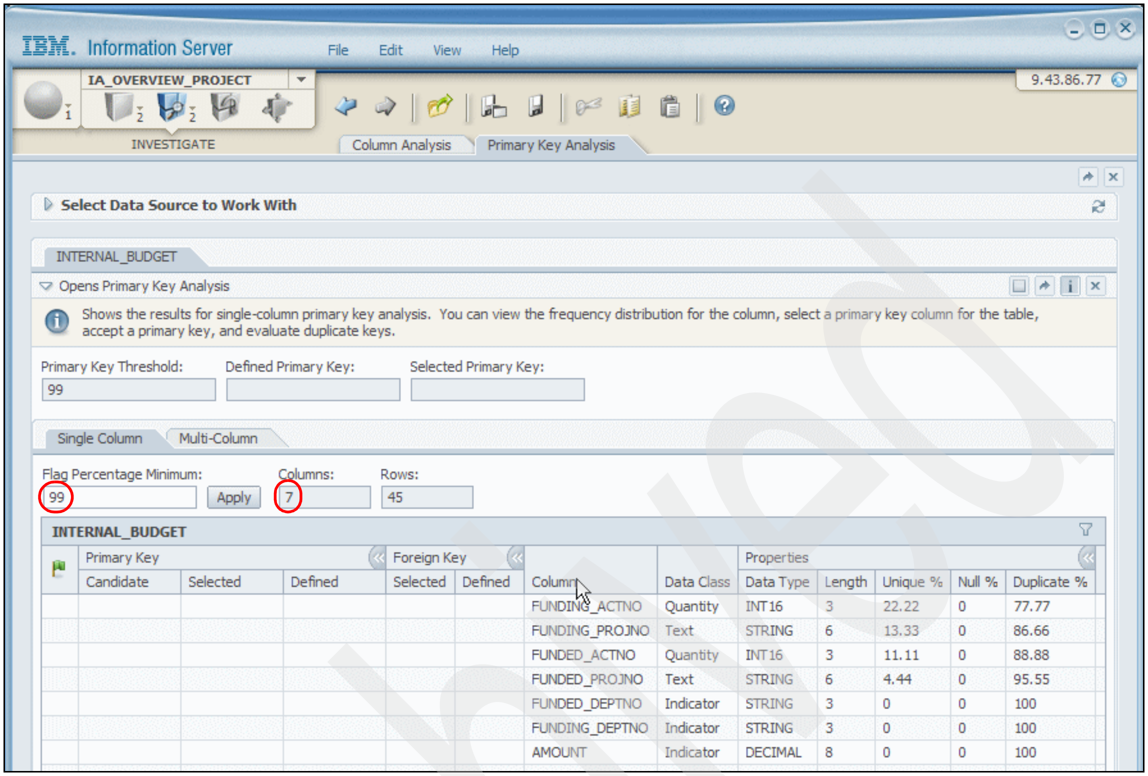
> **Note:** Figure 1-201 on page 263 shows a value of *Other*. In graphical mode, IBM WebSphere Information Analyzer displays only the top *N* values (currently this is hard coded with a value of 10), and the remaining values (total number of values minus *N*) are lumped together in the *Other* category. In the future, *N* will be configurable.

10. Click **Close** in Figure 1-201 on page 263 and in the analysis results screen (not shown here) to proceed to the Primary Key Analysis workspace shown in Figure 1-202 on page 264, which shows the INTERNAL_BUDGET table as having a multi-column Selected Key.



Name	Defined Key	Selected Key	Selected Key Type	Number of Candidates	Column A
DEMO_MACHINE					
IA_SAMPLE					
IA					
DEPARTMENT	Yes	No		2	Analyzed
EMP_HISTORY	Yes	No		2	Analyzed
EMPLOYEE	Yes	Yes	Single Column	1	Analyzed
INTERNAL_BUDGET	No	No		0	Analyzed
ORGANIZATION	No	Yes	Single Column	4	Analyzed
PROJACT	Yes	Yes	Multi-Column	1	Analyzed
PROJECT	Yes	No		2	Analyzed
IA_SAMPLE_ALT					
IA					
DEPARTMENT	Yes	No		2	Analyzed
EMP_HISTORY	Yes	No		2	Analyzed

Figure 1-192 Multi-column analysis on table with no defined primary key 1/11



Select Data Source to Work With

INTERNAL_BUDGET

Opens Primary Key Analysis

Shows the results for single-column primary key analysis. You can view the frequency distribution for the column, select a primary key column for the table, accept a primary key, and evaluate duplicate keys.

Primary Key Threshold: 99

Defined Primary Key:

Selected Primary Key:

Single Column | Multi-Column

Flag Percentage Minimum: 99 Apply

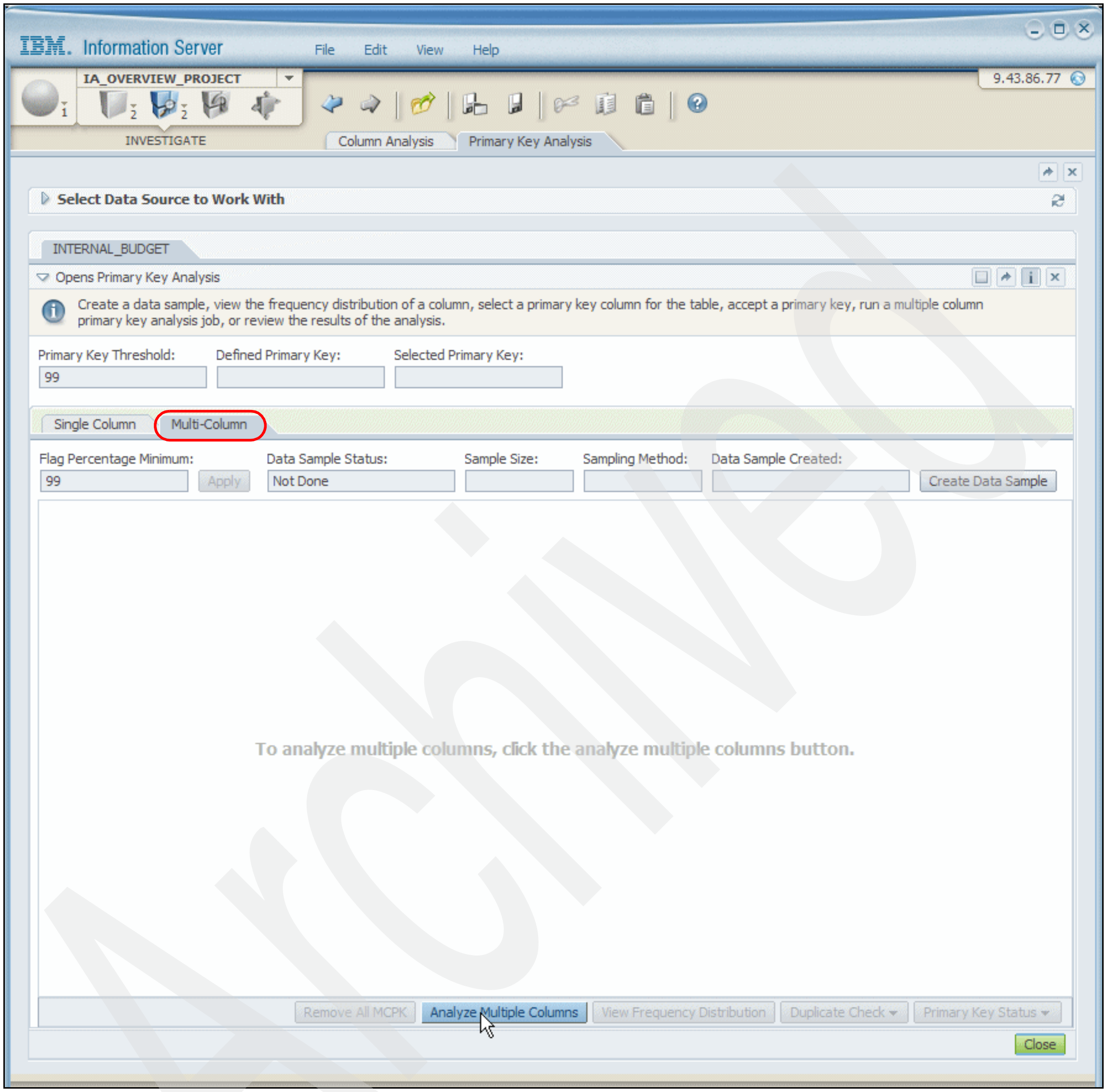
Columns: 7

Rows: 45

INTERNAL_BUDGET

	Primary Key			Foreign Key				Properties				
🏁	Candidate	Selected	Defined	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Null %	Duplicate %
						FUNDING_ACTNO	Quantity	INT16	3	22.22	0	77.77
						FUNDING_PROJNO	Text	STRING	6	13.33	0	86.66
						FUNDED_ACTNO	Quantity	INT16	3	11.11	0	88.88
						FUNDED_PROJNO	Text	STRING	6	4.44	0	95.55
						FUNDED_DEPTNO	Indicator	STRING	3	0	0	100
						FUNDING_DEPTNO	Indicator	STRING	3	0	0	100
						AMOUNT	Indicator	DECIMAL	8	0	0	100

Figure 1-193 Multi-column analysis on table with no defined primary key 2/11



Create a data sample, view the frequency distribution of a column, select a primary key column for the table, accept a primary key, run a multiple column primary key analysis job, or review the results of the analysis.

Primary Key Threshold: 99

Defined Primary Key:

Selected Primary Key:

Flag Percentage Minimum: 99

Data Sample Status: Not Done

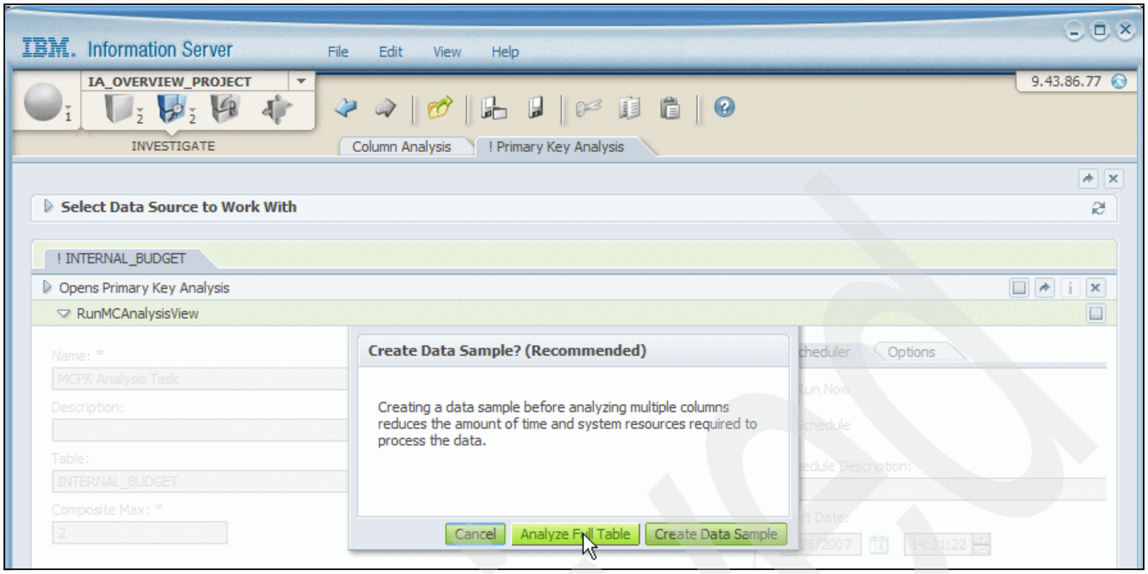
Sample Size:

Sampling Method:

Data Sample Created:

To analyze multiple columns, click the analyze multiple columns button.

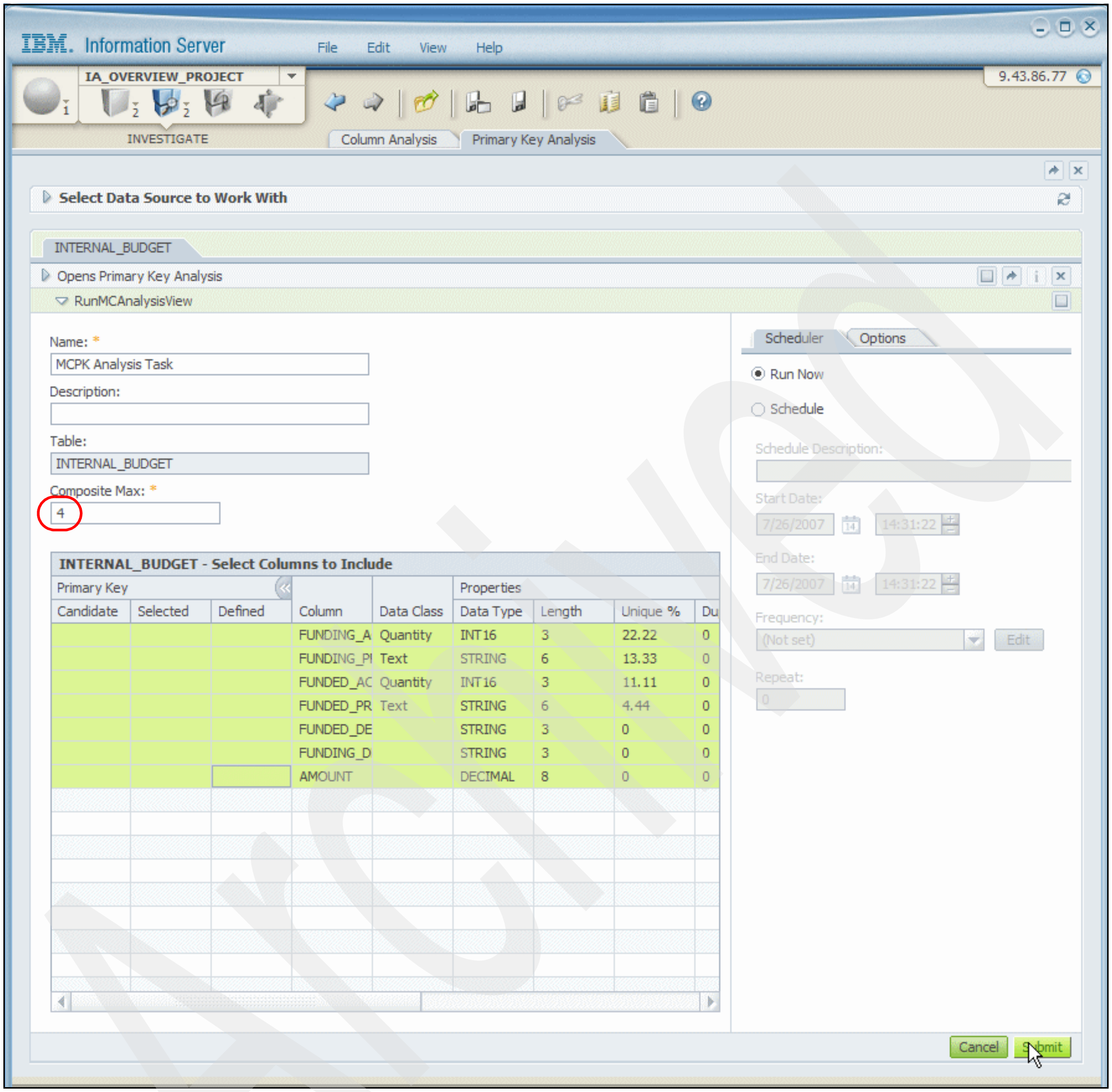
Figure 1-194 Multi-column analysis on table with no defined primary key 3/11



Create Data Sample? (Recommended)

Creating a data sample before analyzing multiple columns reduces the amount of time and system resources required to process the data.

Figure 1-195   Multi-column analysis on table with no defined primary key 4/11



Name: *
MCPK Analysis Task

Description:

Table:
INTERNAL_BUDGET

Composite Max: *
4

INTERNAL_BUDGET - Select Columns to Include

Primary Key					Properties			
Candidate	Selected	Defined	Column	Data Class	Data Type	Length	Unique %	Du
			FUNDING_A	Quantity	INT16	3	22.22	0
			FUNDING_PI	Text	STRING	6	13.33	0
			FUNDED_AC	Quantity	INT16	3	11.11	0
			FUNDED_PR	Text	STRING	6	4.44	0
			FUNDED_DE		STRING	3	0	0
			FUNDING_D		STRING	3	0	0
			AMOUNT		DECIMAL	8	0	0

● Run Now

○ Schedule

Schedule Description:

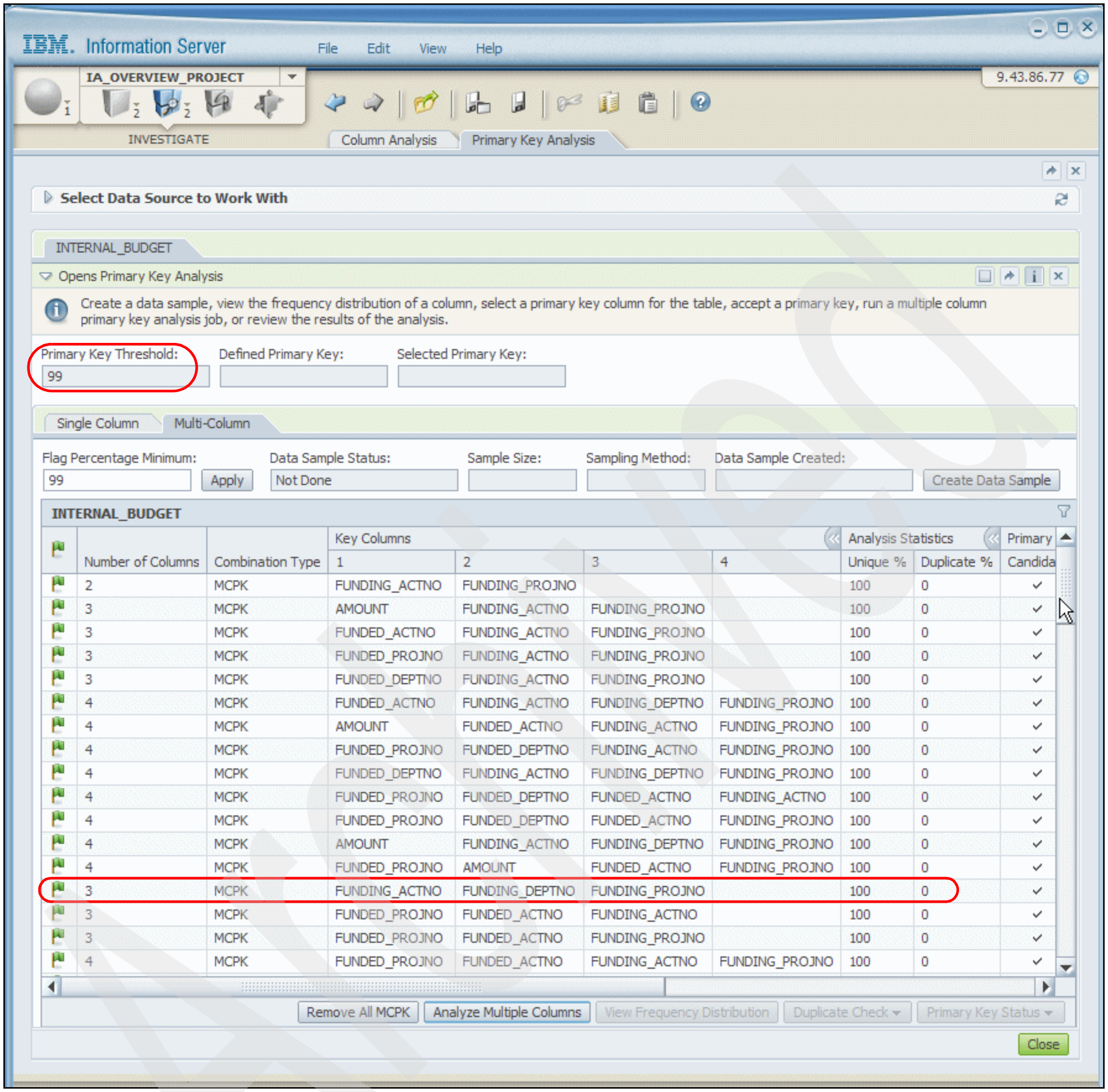
Start Date: 7/26/2007 14:31:22

End Date: 7/26/2007 14:31:22

Frequency: (Not set) Edit

Repeat: 0

Figure 1-196 Multi-column analysis on table with no defined primary key 5/11



Create a data sample, view the frequency distribution of a column, select a primary key column for the table, accept a primary key, run a multiple column primary key analysis job, or review the results of the analysis.

Primary Key Threshold: 99

Defined Primary Key:

Selected Primary Key:

Flag Percentage Minimum: 99 Apply

Data Sample Status: Not Done

Sample Size:

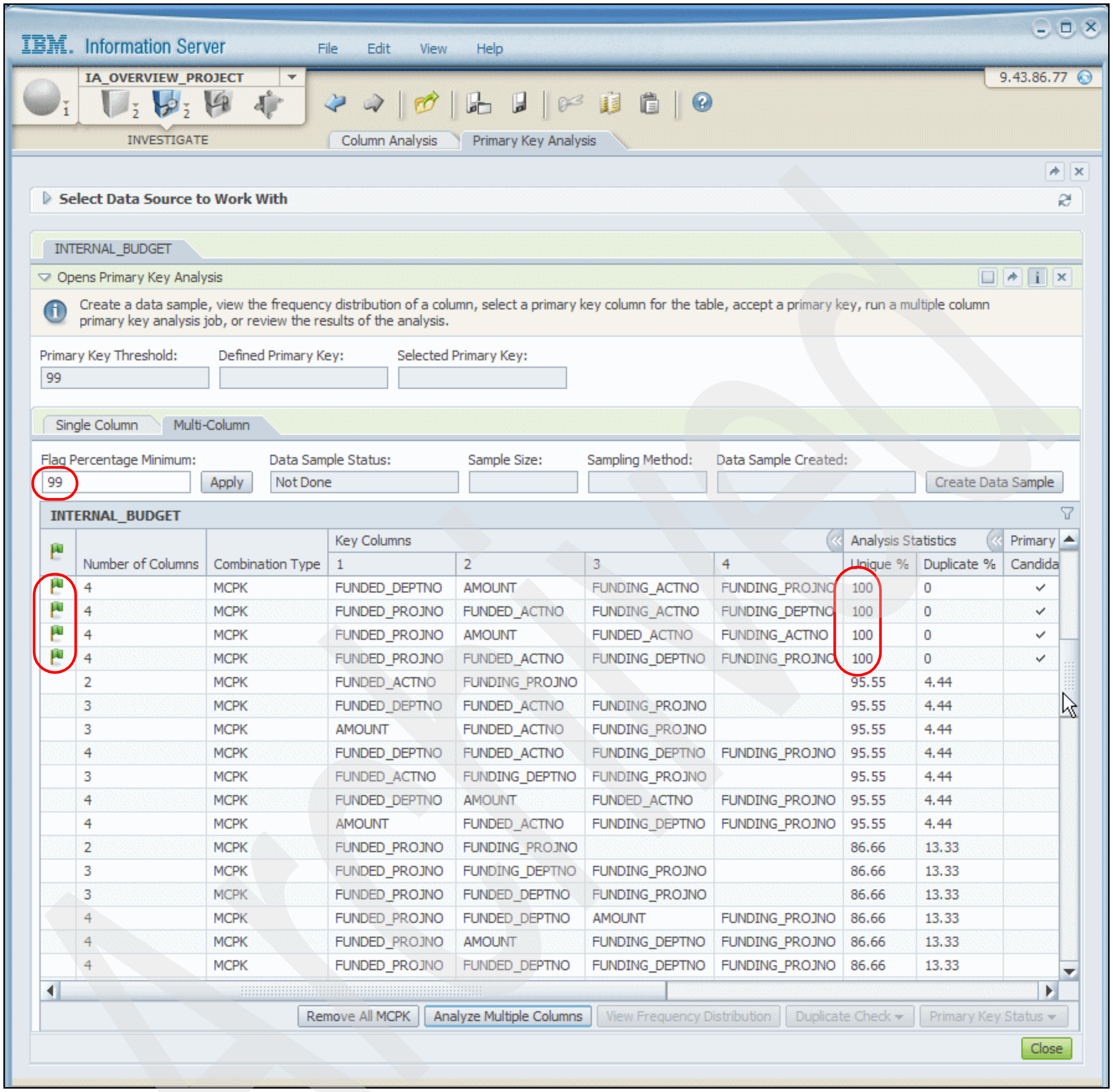
Sampling Method:

Data Sample Created:

INTERNAL_BUDGET

			Key Columns				Analysis Statistics		Primary
	Number of Columns	Combination Type	1	2	3	4	Unique %	Duplicate %	Candida
	2	MCPK	FUNDING_ACTNO	FUNDING_PROJNO			100	0	✓
	3	MCPK	AMOUNT	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
	3	MCPK	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
	3	MCPK	FUNDED_PROJNO	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
	3	MCPK	FUNDED_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
	4	MCPK	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	100	0	✓
	4	MCPK	AMOUNT	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_PROJNO	100	0	✓
	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO	100	0	✓
	4	MCPK	FUNDED_DEPTNO	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	100	0	✓
	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDED_ACTNO	FUNDING_ACTNO	100	0	✓
	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDED_ACTNO	FUNDING_PROJNO	100	0	✓
	4	MCPK	AMOUNT	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	100	0	✓
	4	MCPK	FUNDED_PROJNO	AMOUNT	FUNDED_ACTNO	FUNDING_PROJNO	100	0	✓
	3	MCPK	FUNDING_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO		100	0	✓
	3	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_ACTNO		100	0	✓
	3	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_PROJNO		100	0	✓
	4	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_PROJNO	100	0	✓

Figure 1-197 Multi-column analysis on table with no defined primary key 6/11



Create a data sample, view the frequency distribution of a column, select a primary key column for the table, accept a primary key, run a multiple column primary key analysis job, or review the results of the analysis.

Primary Key Threshold: 99

Defined Primary Key:

Selected Primary Key:

Flag Percentage Minimum: 99

Data Sample Status: Not Done

Sample Size:

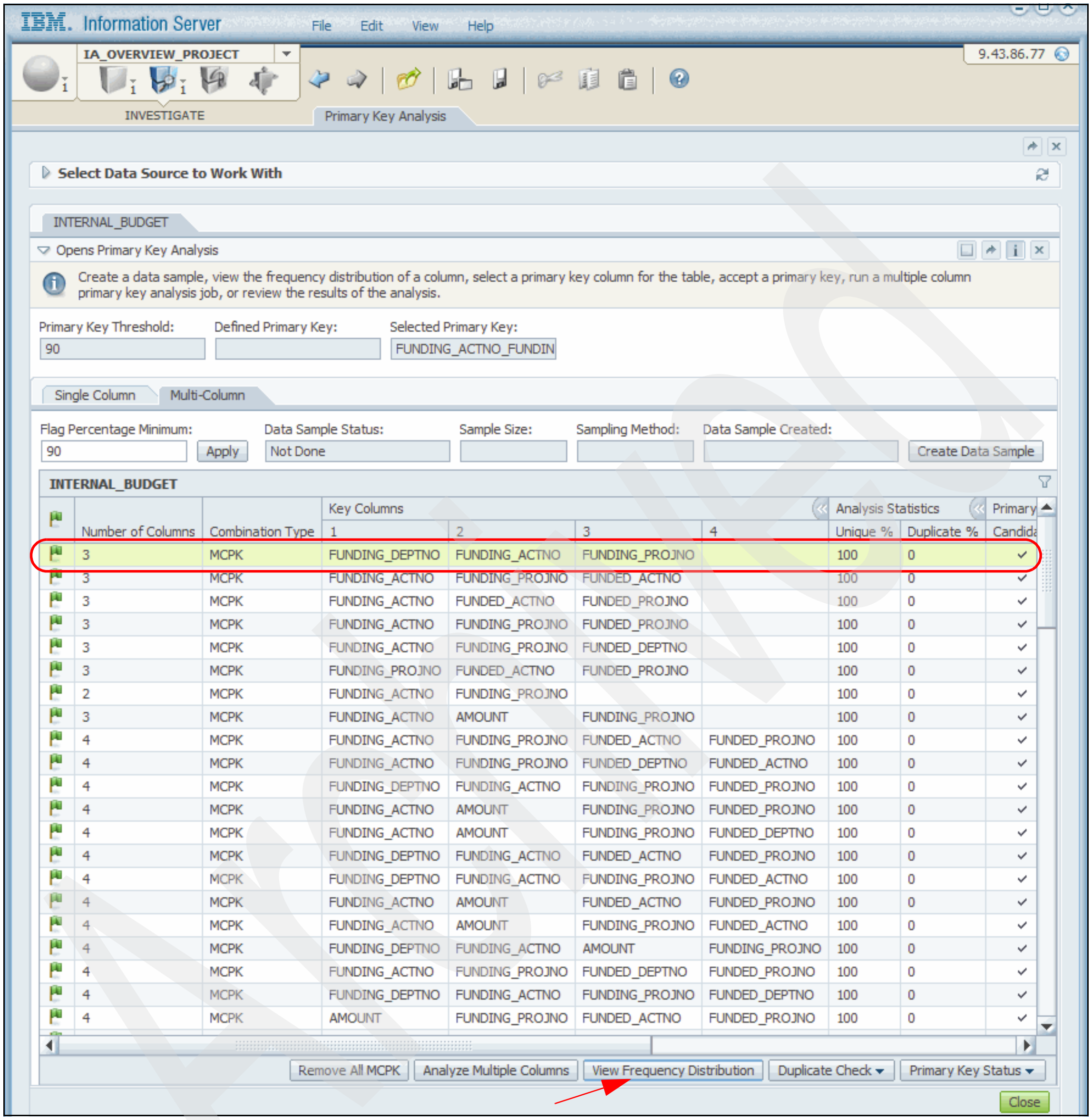
Sampling Method:

Data Sample Created:

INTERNAL_BUDGET

🚩	Number of Columns	Combination Type	Key Columns				Analysis Statistics		Primary
			1	2	3	4	Unique %	Duplicate %	Candida
🚩	4	MCPK	FUNDED_DEPTNO	AMOUNT	FUNDING_ACTNO	FUNDING_PROJNO	100	0	✓
🚩	4	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_ACTNO	FUNDING_DEPTNO	100	0	✓
🚩	4	MCPK	FUNDED_PROJNO	AMOUNT	FUNDED_ACTNO	FUNDING_ACTNO	100	0	✓
🚩	4	MCPK	FUNDED_PROJNO	FUNDED_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	100	0	✓
	2	MCPK	FUNDED_ACTNO	FUNDING_PROJNO			95.55	4.44	
	3	MCPK	FUNDED_DEPTNO	FUNDED_ACTNO	FUNDING_PROJNO		95.55	4.44	
	3	MCPK	AMOUNT	FUNDED_ACTNO	FUNDING_PROJNO		95.55	4.44	
	4	MCPK	FUNDED_DEPTNO	FUNDED_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	95.55	4.44	
	3	MCPK	FUNDED_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO		95.55	4.44	
	4	MCPK	FUNDED_DEPTNO	AMOUNT	FUNDED_ACTNO	FUNDING_PROJNO	95.55	4.44	
	4	MCPK	AMOUNT	FUNDED_ACTNO	FUNDING_DEPTNO	FUNDING_PROJNO	95.55	4.44	
	2	MCPK	FUNDED_PROJNO	FUNDING_PROJNO			86.66	13.33	
	3	MCPK	FUNDED_PROJNO	FUNDING_DEPTNO	FUNDING_PROJNO		86.66	13.33	
	3	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDING_PROJNO		86.66	13.33	
	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	AMOUNT	FUNDING_PROJNO	86.66	13.33	
	4	MCPK	FUNDED_PROJNO	AMOUNT	FUNDING_DEPTNO	FUNDING_PROJNO	86.66	13.33	
	4	MCPK	FUNDED_PROJNO	FUNDED_DEPTNO	FUNDING_DEPTNO	FUNDING_PROJNO	86.66	13.33	

Figure 1-198 Multi-column analysis on table with no defined primary key 7/11



Create a data sample, view the frequency distribution of a column, select a primary key column for the table, accept a primary key, run a multiple column primary key analysis job, or review the results of the analysis.

Primary Key Threshold: 90

Defined Primary Key:

Selected Primary Key: FUNDING_ACTNO_FUNDIN

Flag Percentage Minimum: 90

Data Sample Status: Not Done

Sample Size:

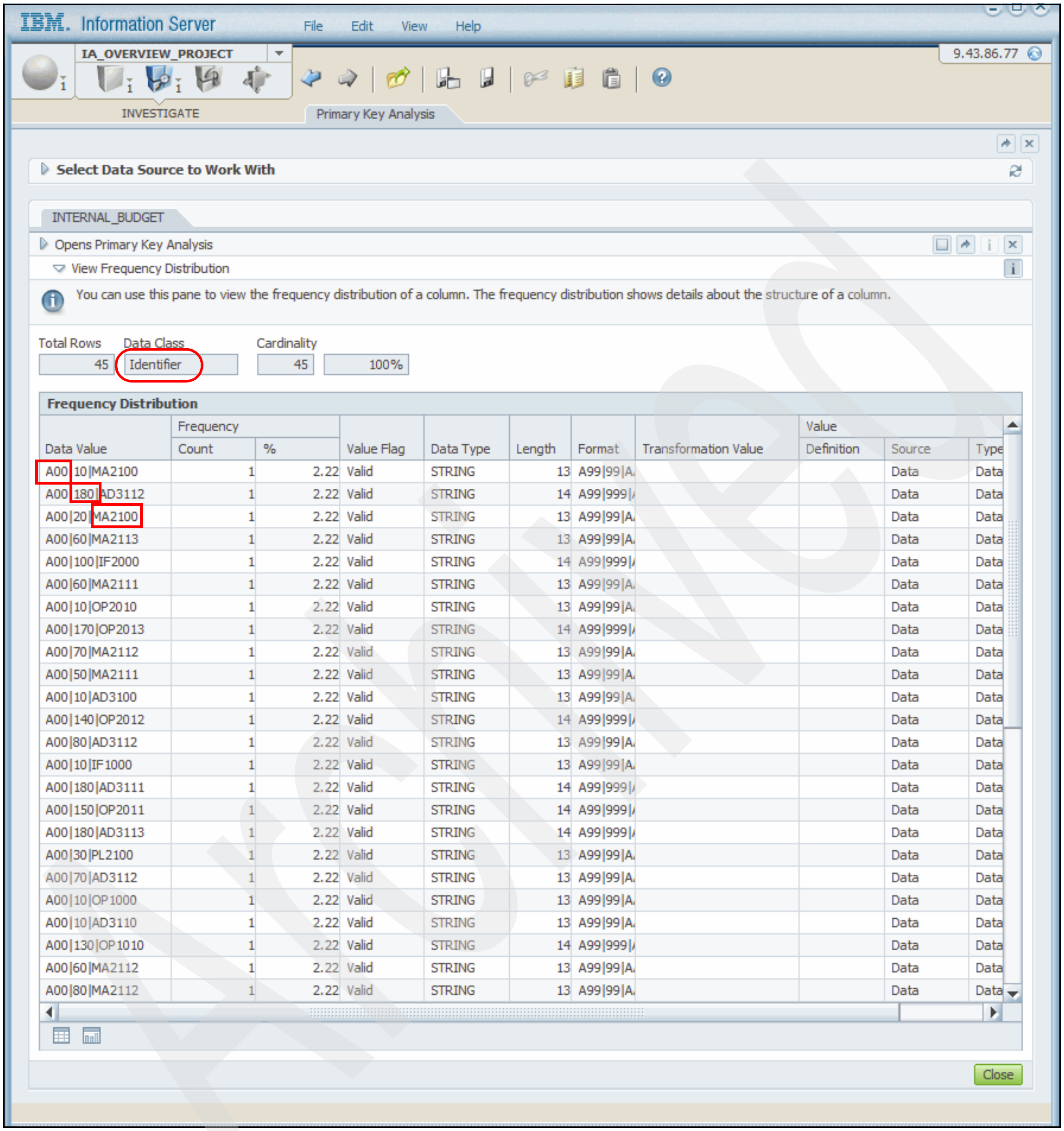
Sampling Method:

Data Sample Created:

INTERNAL_BUDGET

			Key Columns				Analysis Statistics		Primary
	Number of Columns	Combination Type	1	2	3	4	Unique %	Duplicate %	Candida
	3	MCPK	FUNDING_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO		100	0	✓
	3	MCPK	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_ACTNO		100	0	✓
	3	MCPK	FUNDING_ACTNO	FUNDED_ACTNO	FUNDED_PROJNO		100	0	✓
	3	MCPK	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_PROJNO		100	0	✓
	3	MCPK	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_DEPTNO		100	0	✓
	3	MCPK	FUNDING_PROJNO	FUNDED_ACTNO	FUNDED_PROJNO		100	0	✓
	2	MCPK	FUNDING_ACTNO	FUNDING_PROJNO			100	0	✓
	3	MCPK	FUNDING_ACTNO	AMOUNT	FUNDING_PROJNO		100	0	✓
	4	MCPK	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_ACTNO	FUNDED_PROJNO	100	0	✓
	4	MCPK	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_DEPTNO	FUNDED_ACTNO	100	0	✓
	4	MCPK	FUNDING_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_PROJNO	100	0	✓
	4	MCPK	FUNDING_ACTNO	AMOUNT	FUNDING_PROJNO	FUNDED_PROJNO	100	0	✓
	4	MCPK	FUNDING_ACTNO	AMOUNT	FUNDING_PROJNO	FUNDED_DEPTNO	100	0	✓
	4	MCPK	FUNDING_DEPTNO	FUNDING_ACTNO	FUNDED_ACTNO	FUNDED_PROJNO	100	0	✓
	4	MCPK	FUNDING_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_ACTNO	100	0	✓
	4	MCPK	FUNDING_ACTNO	AMOUNT	FUNDED_ACTNO	FUNDED_PROJNO	100	0	✓
	4	MCPK	FUNDING_ACTNO	AMOUNT	FUNDING_PROJNO	FUNDED_ACTNO	100	0	✓
	4	MCPK	FUNDING_DEPTNO	FUNDING_ACTNO	AMOUNT	FUNDING_PROJNO	100	0	✓
	4	MCPK	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_DEPTNO	FUNDED_PROJNO	100	0	✓
	4	MCPK	FUNDING_DEPTNO	FUNDING_ACTNO	FUNDING_PROJNO	FUNDED_DEPTNO	100	0	✓
	4	MCPK	AMOUNT	FUNDING_PROJNO	FUNDED_ACTNO	FUNDED_PROJNO	100	0	✓

Figure 1-199 Multi-column analysis on table with no defined primary key 8/11



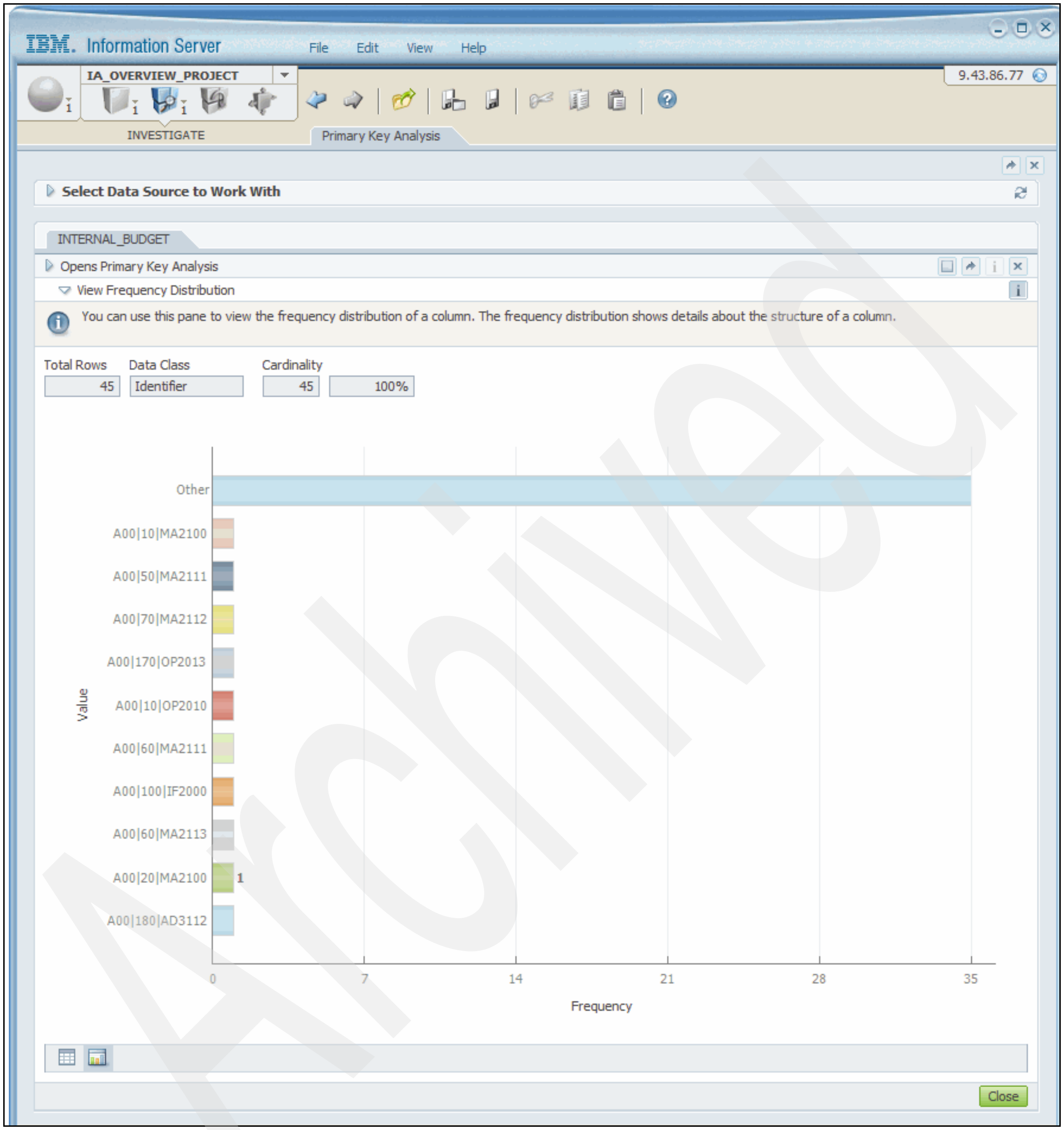
You can use this pane to view the frequency distribution of a column. The frequency distribution shows details about the structure of a column.

Total Rows: 45
Data Class: Identifier
Cardinality: 45 100%

Frequency Distribution

	Frequency							Value		
Data Value	Count	%	Value Flag	Data Type	Length	Format	Transformation Value	Definition	Source	Type
A00|10|MA2100	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|180|AD3112	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|20|MA2100	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|60|MA2113	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|100|IF2000	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|60|MA2111	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|10|OP2010	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|170|OP2013	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|70|MA2112	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|50|MA2111	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|10|AD3100	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|140|OP2012	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|80|AD3112	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|10|IF1000	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|180|AD3111	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|150|OP2011	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|180|AD3113	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|30|PL2100	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|70|AD3112	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|10|OP1000	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|10|AD3110	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|130|OP1010	1	2.22	Valid	STRING	14	A99|999|			Data	Data
A00|60|MA2112	1	2.22	Valid	STRING	13	A99|99|A			Data	Data
A00|80|MA2112	1	2.22	Valid	STRING	13	A99|99|A			Data	Data

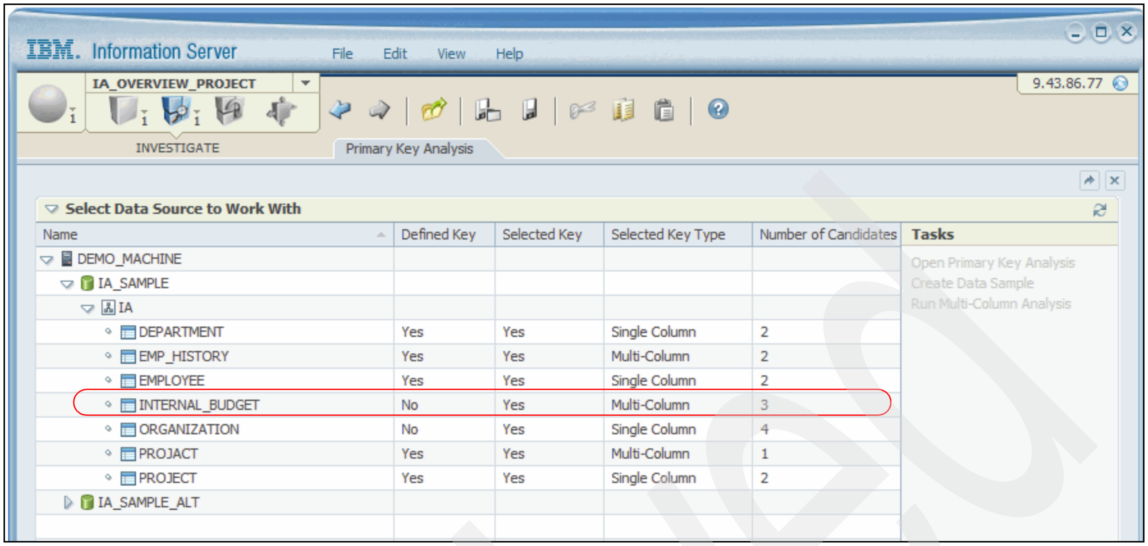
Figure 1-200 Multi-column analysis on table with no defined primary key 9/11



You can use this pane to view the frequency distribution of a column. The frequency distribution shows details about the structure of a column.

Total Rows: 45
Data Class: Identifier
Cardinality: 45 100%

Figure 1-201 Multi-column analysis on table with no defined primary key 10/11



Select Data Source to Work With

Name	Defined Key	Selected Key	Selected Key Type	Number of Candidates
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT	Yes	Yes	Single Column	2
EMP_HISTORY	Yes	Yes	Multi-Column	2
EMPLOYEE	Yes	Yes	Single Column	2
INTERNAL_BUDGET	No	Yes	Multi-Column	3
ORGANIZATION	No	Yes	Single Column	4
PROJACT	Yes	Yes	Multi-Column	1
PROJECT	Yes	Yes	Single Column	2
IA_SAMPLE_ALT				

Figure 1-202 Multi-column analysis on table with no defined primary key 11/11

1.9 Foreign key analysis

To define and validate the relationships between tables, you can run a foreign key analysis job on two or more tables to find candidates for foreign keys, select foreign keys, and then validate their referential integrity.

Prior to running a foreign key analysis, you must have performed the following tasks:

- Run column analysis on the columns in two or more tables
- Have a defined primary key or selected a primary key (if one is not defined explicitly) for each table

The foreign key analysis job builds a complete set of all the column pairs between the primary key columns and the remaining selected columns in the selected tables. The primary key column of one table is paired with all of the columns of the other tables. Next, the system performs a compatibility test on each column pair to determine whether those columns are compatible with each other. If the column pair is compatible, the columns are flagged and then evaluated further.

Any set of columns are considered to be a compatible pair when the conditions described in Table 1-9 apply.

Table 1-9 Column pair compatibility considerations

Criteria	Considerations
Format	Columns are paired that have at least one format match of the domain values. For example, assume column A has domain values 'A', 'B', and the number 1, and column B has domain values with numbers 6,7, and 8. Column A has two formats, A and 9, while column B has a single format of 9. Columns A and B will be considered compatible because both these columns share a common format of 9.
Length	The maximum length of the base column is greater than or equal to the minimum length of the candidate column, and the minimum length of the base column is less than or equal to maximum length of the candidate column. This will eliminate columns which are totally null or empty. This is applicable to all data types.
Scale	The maximum scale of the base column is greater than or equal to the minimum scale of the candidate column, and the minimum scale of the base column is less than or equal to the maximum scale of the candidate column.
Precision	The maximum precision of the base column is greater than or equal to the minimum precision of the candidate column, and the minimum precision of the base column is less than or equal to the maximum precision of the candidate column.

Restriction: Currently you cannot eliminate a compatible pair from the list compiled for foreign key analysis. This restriction is lifted in Version 8.1.

After reviewing the results of the job, you can test for referential integrity and determine if a foreign key candidate should be selected as a foreign key.

In the following sections, we describe briefly the main functions of foreign key analysis and the results produced by foreign key analysis. Using a sample set of tables, we describe the process of running foreign key analysis, performing referential integrity analysis, applying filters to foreign key analysis, and reviewing the results of the foreign key analysis.

1.9.1 Foreign key analysis functions

Foreign key analysis has the following characteristics:

- You must have Information Analyzer Data Operator privileges to perform foreign key analysis as shown in Table 1-4 on page 34.
- Can be performed on two or more tables.
- Allows you to filter out the tables and columns that should be excluded from pairing consideration.
- Allows you to check for orphans and parents that do not have children records by performing a referential integrity analysis.
- Can be scheduled to run at a set date and time.
- Reports can be produced of foreign key analysis results such as defined foreign key candidate and chosen summary, defined foreign key referential integrity exceptions, defined foreign key summary, primary keys without children detail, and referential integrity detail.

Note: For complete details, refer to *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.

As mentioned earlier, during a foreign key analysis job, a complete set of all the column pairs between the primary key columns and the remaining selected columns in the selected tables is built. The system then performs a compatibility test on each column pair to determine whether those columns are compatible with each other. If the column pair is compatible, the columns are flagged and then evaluated further.

You can then run a foreign key analysis job on those compatible column pairs. Column pairs that are not compatible are excluded from the foreign key analysis. When the compatibility test completes, you review the results. The results show the total number of pairs that are created, the total number of incompatible pairs, and the total number of compatible pairs. You review the compatibility test results before you continue with the analysis.

If you want to rerun the test to obtain different results, you can return to the table selection step, make modifications, and start again. You can also rerun a column analysis job to review or modify any inferences or selections before rerunning the foreign key analysis job.

Note: When you rerun a column analysis job and modify the results, a new frequency distribution is generated and used as input for the foreign key analysis job.

After you review the results from the compatibility test, the system compares the frequency distributions of each column in the pair to determine whether they have *common domains*. Columns share a common domain when they share the same values.

The values from the frequency distribution of the second column are compared against the values from the frequency distribution of the first column (the primary key column). After the comparison, the number of matching values in the columns are counted, and a *commonality percentage* is calculated.[18]

Next, the commonality percentage is compared with the *common domain threshold*. You set the common domain threshold to define the criteria for determining when two columns share a commonality. For example, if you set the common domain threshold to 95%, the columns must have a commonality percentage equal to 95% or higher for the system to infer that they share a common domain. You can tighten the criteria for commonality by raising the common domain threshold setting or loosen it by lowering the setting. In the example using column 1 and column 2, column 2 has a common domain with column 1 because its commonality percentage (98%) is higher than the common domain threshold (95%).

You use the results from the common domain test to determine whether a table contains a column or multiple columns that match the primary key column in another table. If there is a match, the columns in the first table are inferred as foreign key candidates. To qualify as a foreign key candidate, there must be at least one column in the first table that has a common domain with each of the primary key columns in the second table.

Note: If your data already contains defined foreign keys, the columns are marked as foreign keys during analysis. However, you can still use the results of a foreign key analysis job to assess the validity of defined foreign keys.

Referential integrity analysis shows the common domain percentages of the primary key-foreign key relationship including a graphical representation of the overlap. You run a referential integrity analysis job to evaluate whether all of the references between the foreign keys and the primary keys in your data are valid. You use the results to help you choose foreign keys or remove foreign key violations from your data.

During referential integrity analysis, the values in the foreign key candidate and the primary key are examined for referential integrity. Foreign keys and primary

[18] A *commonality percentage* describes the proportion of matching values between two columns. For example, you might have a column pair that contains column 1 (the primary key column) and column 2. If 49 out of 50 values in column 2 match column 1, the commonality percentage of column 2 to column 1 is 98%.

keys maintain a referential integrity when all of the values in the foreign key column match all of the values in the referenced primary key column. A referential integrity analysis job identifies the orphan values and calculates statistics about the relationship.

> **Attention:** During referential integrity analysis, if the foreign key column and primary key column are both single columns, the system uses the frequency distribution for each column to perform the test. However, if the foreign key candidate consists of multiple columns, the system creates a virtual column by concatenating the columns together and then generates a frequency distribution for the virtual column.

During a multiple column primary key analysis job, primary keys with multiple columns are already combined into a virtual column and a frequency distribution was generated for the virtual column. Next, the values in the frequency distribution of the foreign key column are matched with the values in the frequency distribution for the primary key column. If any of the values in the foreign key column are not found in the related primary key column, the foreign key data value is flagged as a referential integrity violation. An inverse analysis is also performed to determine whether the values in the primary key column have a matching value in the foreign key column. When the analysis completes, the system shows the results and lists any violations.

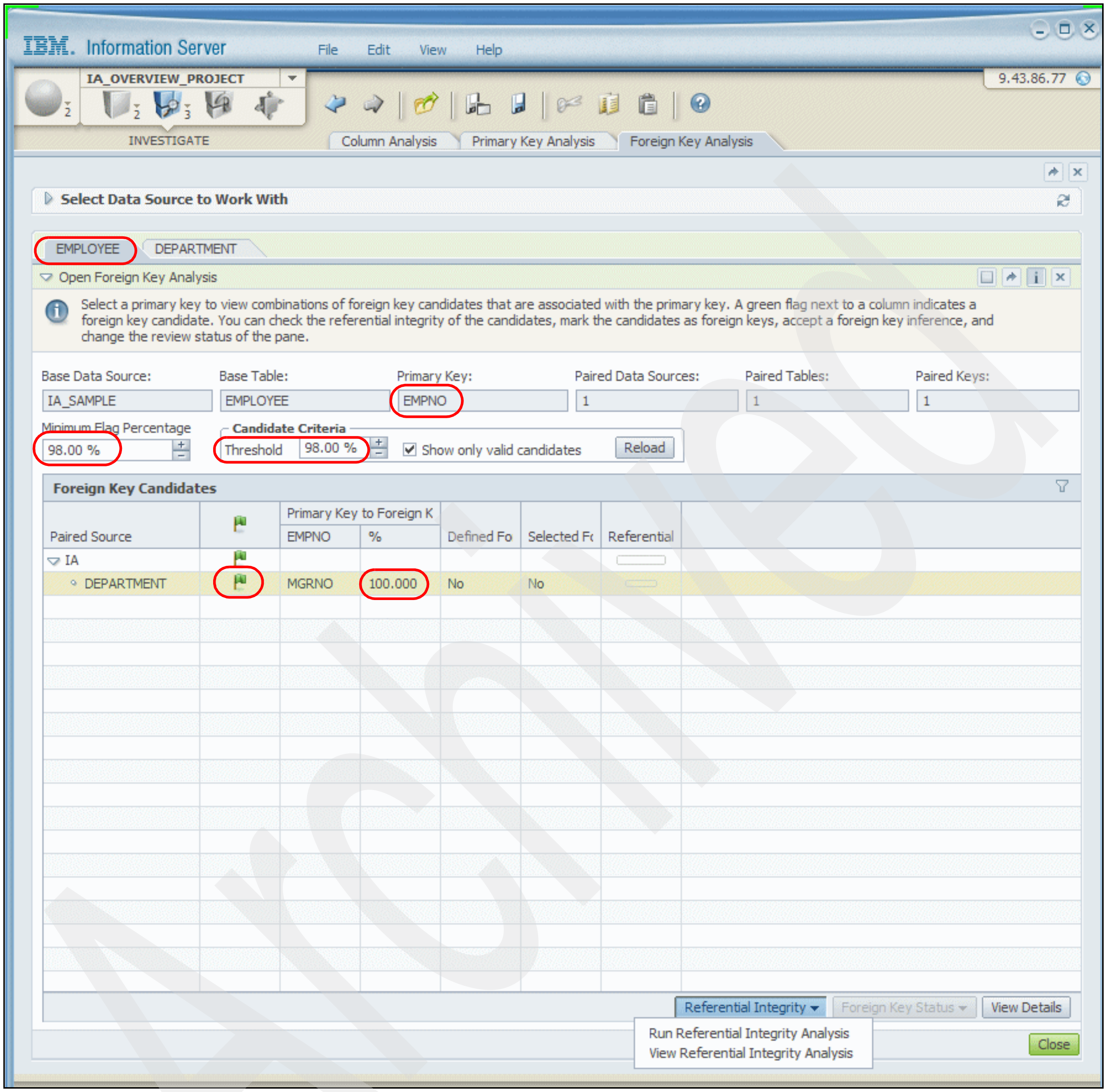
1.9.2 Foreign key analysis results

A foreign key analysis produces the following categories of output:

- Foreign key candidates

  This is a view of combinations of foreign key candidates that are associated with a selected primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of a pane.

  Figure 1-203 shows a view of foreign key candidates. The primary key for which candidates are discovered is EMPNO in the EMPLOYEE table. The Candidate Criteria is 98%, indicating that only those primary key to foreign key matches that exceed this percentage should be considered to be candidates. The Minimum Flag Percentage of 98% indicates that only candidates that exceed 98% should be flagged. You can select this candidate (MGRNO column in the DEPARTMENT table) and click **View Details** to obtain further information.



Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

Base Data Source:	Base Table:	Primary Key:	Paired Data Sources:	Paired Tables:	Paired Keys:
IA_SAMPLE	EMPLOYEE	EMPNO	1	1	1

Minimum Flag Percentage: 98.00 %

Candidate Criteria: Threshold 98.00 %

☑ Show only valid candidates

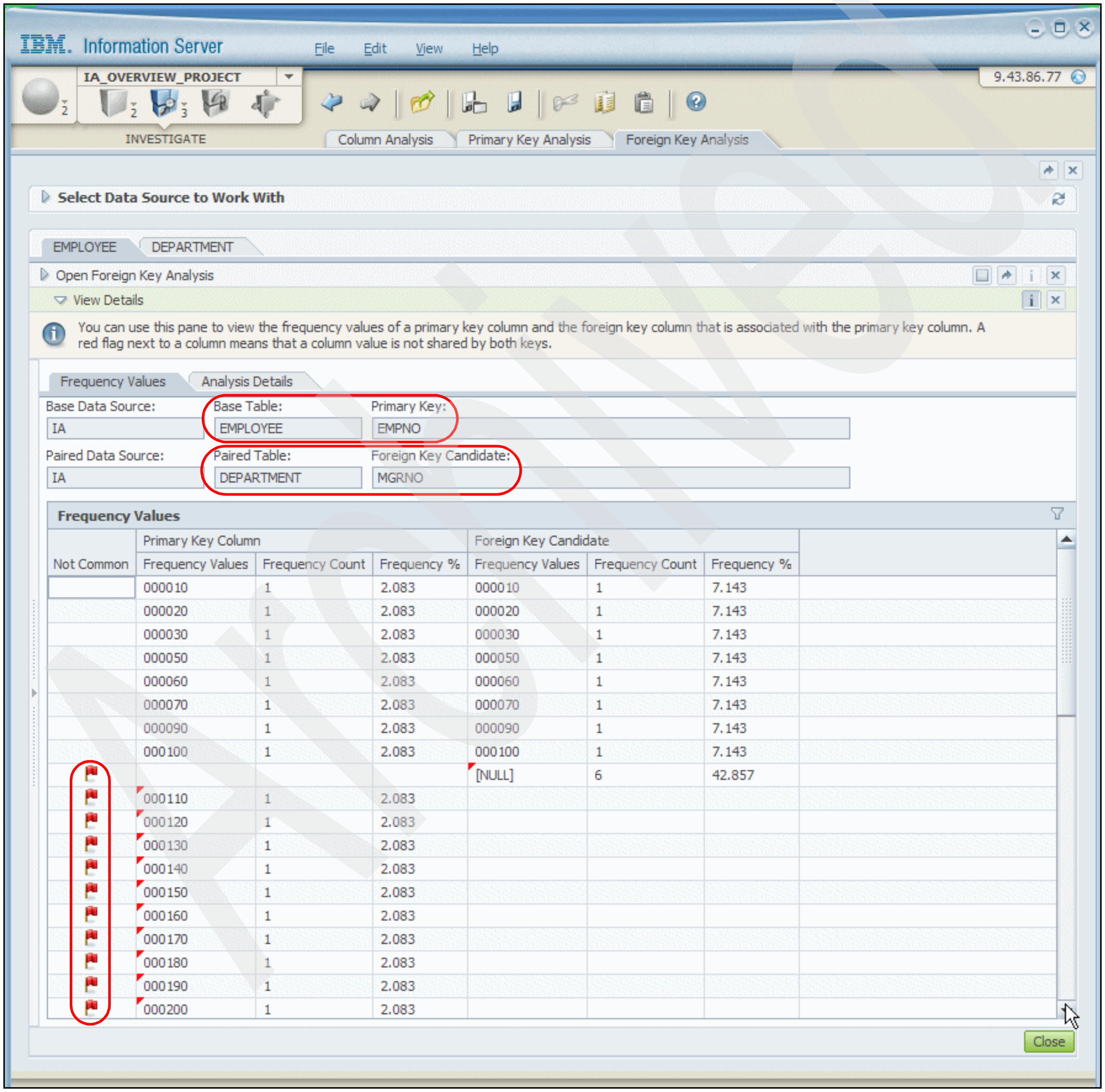
Foreign Key Candidates

		Primary Key to Foreign K				
Paired Source	⚑	EMPNO	%	Defined Fo	Selected Fc	Referential
IA	⚑					
DEPARTMENT	⚑	MGRNO	100.000	No	No	

Figure 1-203 View of foreign key candidates for a selected primary key

- View Details Frequency Values

  This view shows the frequency values of a primary key column and the foreign key column that is associated with the primary key column. A red flag next to a column means that a column values is not shared by both keys. Figure 1-204 shows this view for the EMPNO primary key column in the EMPLOYEE table and the MGRNO foreign key candidate column in the DEPARTMENT table.



You can use this pane to view the frequency values of a primary key column and the foreign key column that is associated with the primary key column. A red flag next to a column means that a column value is not shared by both keys.

Base Data Source: IA
Base Table: EMPLOYEE
Primary Key: EMPNO
Paired Data Source: IA
Paired Table: DEPARTMENT
Foreign Key Candidate: MGRNO

Frequency Values

	Primary Key Column			Foreign Key Candidate		
Not Common	Frequency Values	Frequency Count	Frequency %	Frequency Values	Frequency Count	Frequency %
	000010	1	2.083	000010	1	7.143
	000020	1	2.083	000020	1	7.143
	000030	1	2.083	000030	1	7.143
	000050	1	2.083	000050	1	7.143
	000060	1	2.083	000060	1	7.143
	000070	1	2.083	000070	1	7.143
	000090	1	2.083	000090	1	7.143
	000100	1	2.083	000100	1	7.143
🚩				[NULL]	6	42.857
🚩	000110	1	2.083			
🚩	000120	1	2.083			
🚩	000130	1	2.083			
🚩	000140	1	2.083			
🚩	000150	1	2.083			
🚩	000160	1	2.083			
🚩	000170	1	2.083			
🚩	000180	1	2.083			
🚩	000190	1	2.083			
🚩	000200	1	2.083			

Figure 1-204 View Details Frequency Values for the MGRNO foreign key column in DEPARTMENT

- View Details Analysis Details

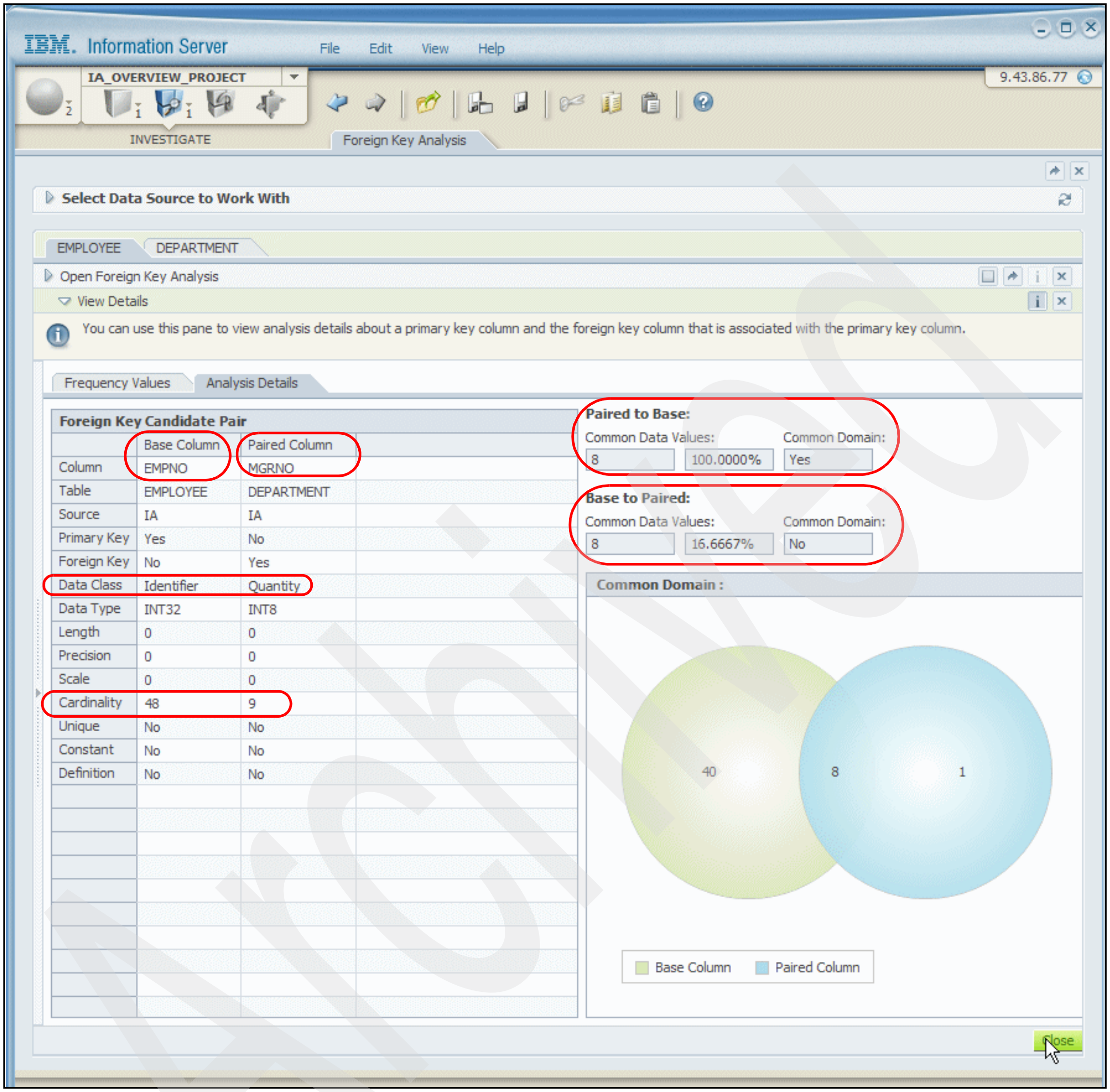
  This view shows the analysis details about a primary key column and the foreign key column that is associated with the primary key column.

  Figure 1-205 shows this view for the EMPNO primary key column in the EMPLOYEE table and the MGRNO foreign key candidate column in the DEPARTMENT table. It shows the base column as being the EMPNO column in the EMPLOYEE table, while the paired column is the MGRNO column in the DEPARTMENT table. The Venn diagram shows the common domains between these columns as follows:

  – There are eight common values between the paired to base column and it has a 100% commonality. All of the values in the MGRNO column are found in the EMPNO column. This is 100% of the values. Because it exceeds the common domain threshold (default is 98%), it is identified as sharing a common domain (Common Domain field has a Yes).

  – There are eight common values out of a total of 48 values between the base (EMPNO) column to paired (MGRNO) column. This corresponds to 16.667% commonality. This 16.667% does not exceed the common domain threshold (default is 98%) and is, therefore, not identified as sharing a common domain (Common Domain field has a No).

> **Note:** The Venn diagram shows one value in the MGRNO column (null) not finding a match in the EMPNO column. Nulls are not part of the percentage calculation.



You can use this pane to view analysis details about a primary key column and the foreign key column that is associated with the primary key column.

Foreign Key Candidate Pair

	Base Column	Paired Column
Column	EMPNO	MGRNO
Table	EMPLOYEE	DEPARTMENT
Source	IA	IA
Primary Key	Yes	No
Foreign Key	No	Yes
Data Class	Identifier	Quantity
Data Type	INT32	INT8
Length	0	0
Precision	0	0
Scale	0	0
Cardinality	48	9
Unique	No	No
Constant	No	No
Definition	No	No

Paired to Base:

Common Data Values: 8 100.0000%

Common Domain: Yes

Base to Paired:

Common Data Values: 8 16.6667%

Common Domain: No

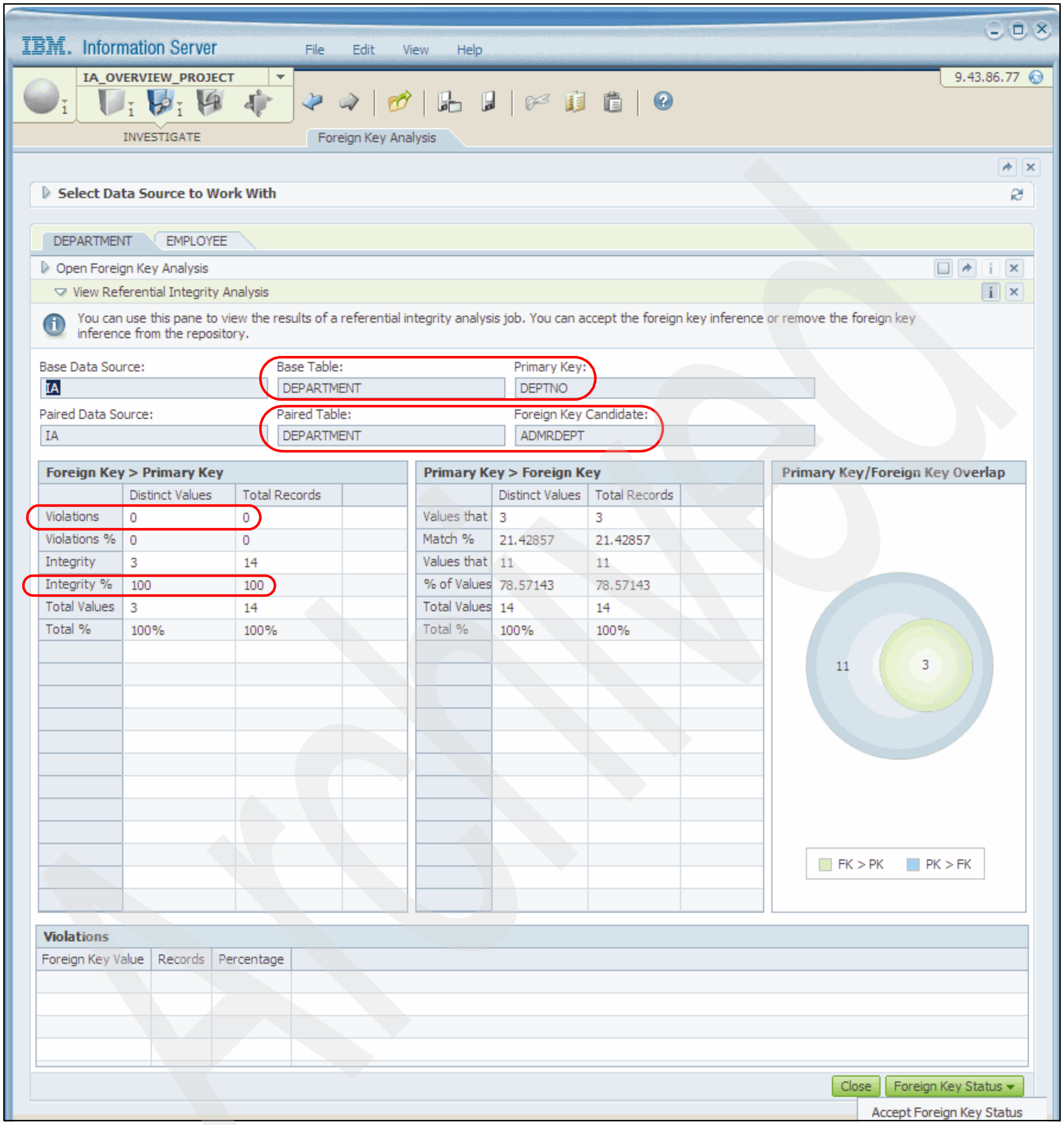
Figure 1-205 View Details Analysis Details for the MGRNO foreign key column in DEPARTMENT

- View Referential Integrity Analysis

  This view shows the results of a referential integrity analysis job. You can accept the foreign key inference, or remove the foreign key inference from the repository.

  Figure 1-206 shows this view for the DEPTNO primary key column in the DEPARTMENT table, and the candidate foreign key column ADMRDEPT in the DEPARTMENT table.

  - The foreign key to primary key portion of the analysis shows that there are no foreign key violations. All values in the ADMRDEPT column have a match in the DEPTNO column. It also shows that there are three distinct values in the ADMRDEPT column out of a total of 14 records.
  - The primary key to foreign key portion of the analysis shows that three values match a foreign key and 11 values in the primary key that do not match a foreign key.



You can use this pane to view the results of a referential integrity analysis job. You can accept the foreign key inference or remove the foreign key inference from the repository.

Base Data Source: IA
Base Table: DEPARTMENT
Primary Key: DEPTNO
Paired Data Source: IA
Paired Table: DEPARTMENT
Foreign Key Candidate: ADMRDEPT

Foreign Key > Primary Key

	Distinct Values	Total Records
Violations	0	0
Violations %	0	0
Integrity	3	14
Integrity %	100	100
Total Values	3	14
Total %	100%	100%

Primary Key > Foreign Key

	Distinct Values	Total Records
Values that	3	3
Match %	21.42857	21.42857
Values that	11	11
% of Values	78.57143	78.57143
Total Values	14	14
Total %	100%	100%

Primary Key/Foreign Key Overlap

Violations

Foreign Key Value	Records	Percentage

Figure 1-206 View Referential Integrity Analysis

1.9.3 Foreign key analysis usage scenario

The modified sample database described in Figure 1-11 on page 31 is used in the foreign key analysis examples described here.

In this section, we describe how to run a foreign key analysis job, perform referential integrity analysis, and a review of the foreign key analysis results.

> **Attention:** In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

Run a foreign key analysis job

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), run a foreign key analysis job on the EMPLOYEE and DEPARTMENT tables as follows:

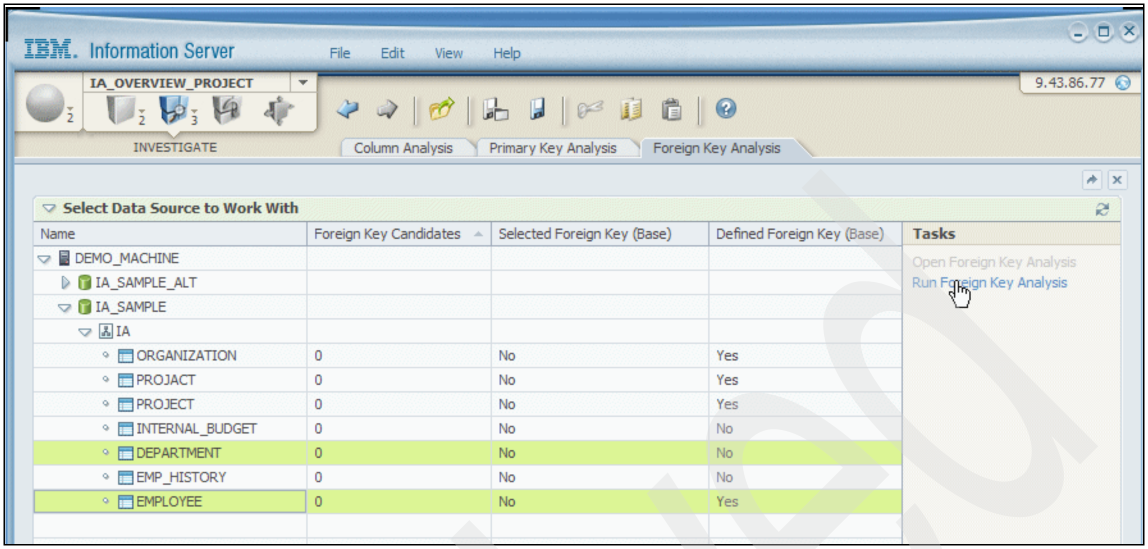
1. On the Investigate navigator menu in the console, select **Foreign Key Analysis** as shown in Figure 1-88 on page 138.
2. On the Foreign Key Analysis workspace, select the tables (DEPARTMENT and EMPLOYEE) that you want to identify the foreign key relationship in, and click **Run Foreign Key Analysis** in the Tasks pane as shown in Figure 1-207 on page 277, click **Run Foreign Key Analysis**.
3. In the Run Foreign Key Analysis pane, type a name (Foreign Key Analysis Task) and optional description for the analysis job as shown in Figure 1-208 on page 278. It shows three compatible pairs between the DEPARTMENT and EMPLOYEE tables. Select **Run Now** to run the job now (or select Schedule to specify a time and date to run the job). Click **Submit and Close**.

   > **Note:** The tab shows EMPLOYEE even though the foreign key analysis was run against the EMPLOYEE and DEPARTMENT tables. It always lists the first table (when multiple tables selected) in the **Foreign Key Analysis** workspace.

   You can filter out candidate foreign keys by applying filters as described in “Applying filters to foreign key analysis” on page 295.
4. At the completion of the foreign key analysis job, click **View Results** as shown in Figure 1-209 on page 279.

5. To view the results of the foreign key analysis, select the tables (EMPLOYEE and DEPARTMENT) in the Foreign Key Analysis workspace, and click **Open Foreign Key Analysis** in the Tasks pane as shown in Figure 1-210 on page 280.[19]
6. In the Open Foreign Key Analysis workspace, select the EMPLOYEE tab which shows the candidate foreign keys that are associated with the EMPNO primary key column. Select the MGRNO candidate foreign key and click **View Details** as shown in Figure 1-211 on page 281.
7. In the View Details pane under the EMPLOYEE tab, select the Frequency Values tab to view the frequency values of the EMPNO primary key and the associated foreign key MGRNO in the DEPARTMENT table as shown in Figure 1-212 on page 282. This is a partial list of all the values. A red flag next to a column indicates that a column value is not shared by both keys.
8. In Figure 1-213 on page 283, the Analysis Details tab shows a Venn diagram of the primary key and associated foreign key values. The Venn diagram shows the common domains between these columns as follows:
   - There are eight common values between the paired to base column and it has a 100% commonality. All of the values in the MGRNO column are found in the EMPNO column. This is 100% of the values, and because it exceeds the common domain threshold (default is 98%), it is identified as sharing a common domain (Common Domain field has a Yes).
   - There are eight common values out of a total of 48 values between the base (EMPNO) column to paired (MGRNO) column. This corresponds to 16.667% commonality. This 16.667% does not exceed the common domain threshold (default is 98%) and is, therefore, not identified as sharing a common domain (Common Domain field has a No).
9. Accept the MGRNO as a foreign key by selecting it in the Foreign Key Candidates, and clicking **Foreign Key → Status Accept Foreign Key Status** as shown in Figure 1-214 on page 284.
10. Figure 1-215 on page 285 shows the foreign key candidates associated with the primary key DEPTNO in the DEPARTMENT table. One of the candidates has a defined foreign key (WORKDEPT), while the other (ADMRDEPT) is not.

[19] Assumes a Run Foreign Key Analysis job has previously been run the set of tables.

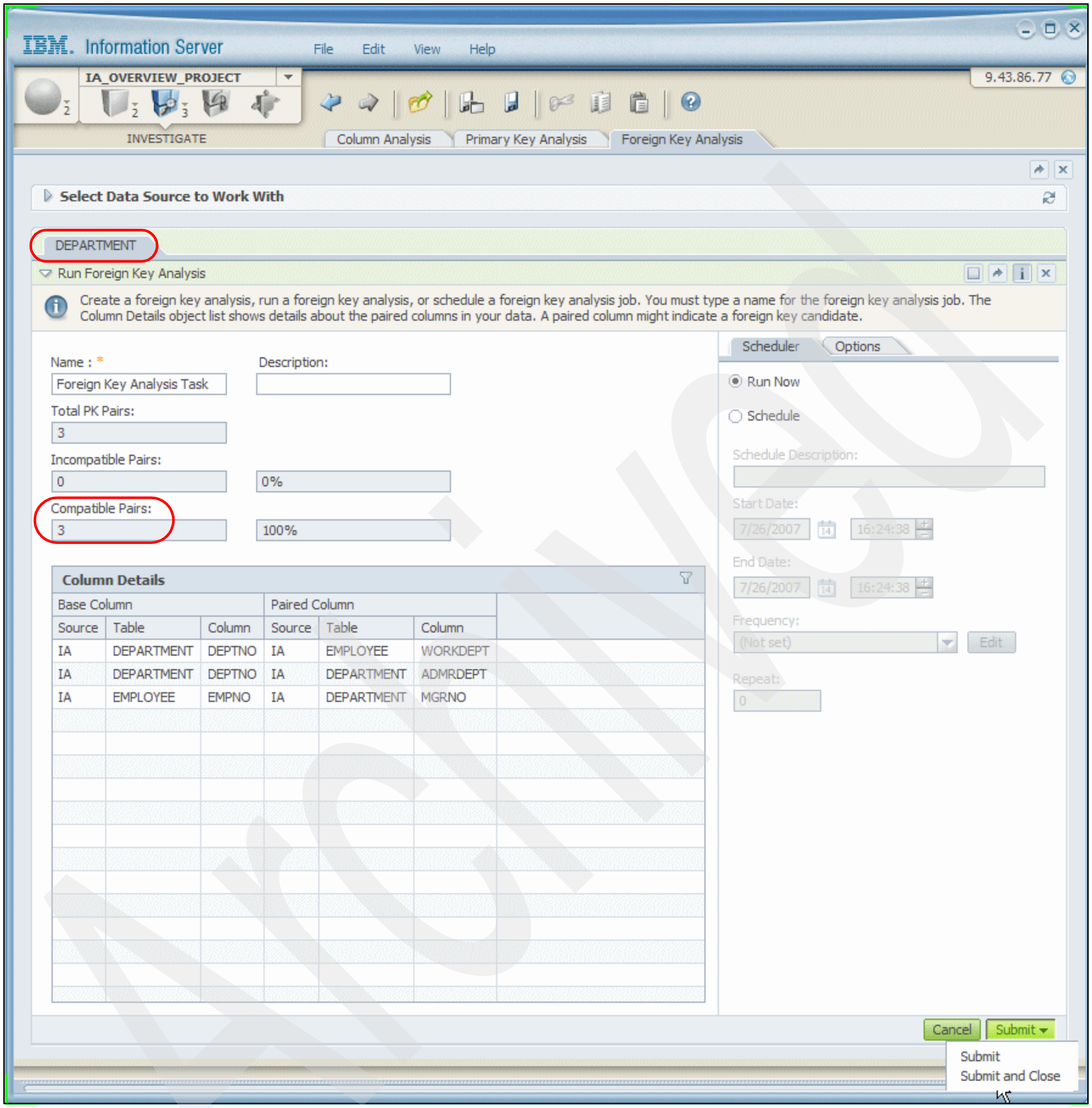


Name	Foreign Key Candidates	Selected Foreign Key (Base)	Defined Foreign Key (Base)
DEMO_MACHINE			
IA_SAMPLE_ALT			
IA_SAMPLE			
IA			
ORGANIZATION	0	No	Yes
PROJACT	0	No	Yes
PROJECT	0	No	Yes
INTERNAL_BUDGET	0	No	No
DEPARTMENT	0	No	No
EMP_HISTORY	0	No	No
EMPLOYEE	0	No	Yes

Tasks

- Open Foreign Key Analysis
- Run Foreign Key Analysis

Figure 1-207 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 1/9



Create a foreign key analysis, run a foreign key analysis, or schedule a foreign key analysis job. You must type a name for the foreign key analysis job. The Column Details object list shows details about the paired columns in your data. A paired column might indicate a foreign key candidate.

Name : * Foreign Key Analysis Task

Description:

Total PK Pairs: 3

Incompatible Pairs: 0 0%

Compatible Pairs: 3 100%

Column Details

Base Column			Paired Column		
Source	Table	Column	Source	Table	Column
IA	DEPARTMENT	DEPTNO	IA	EMPLOYEE	WORKDEPT
IA	DEPARTMENT	DEPTNO	IA	DEPARTMENT	ADMRDEPT
IA	EMPLOYEE	EMPNO	IA	DEPARTMENT	MGRNO

Scheduler Options

● Run Now

○ Schedule

Schedule Description:

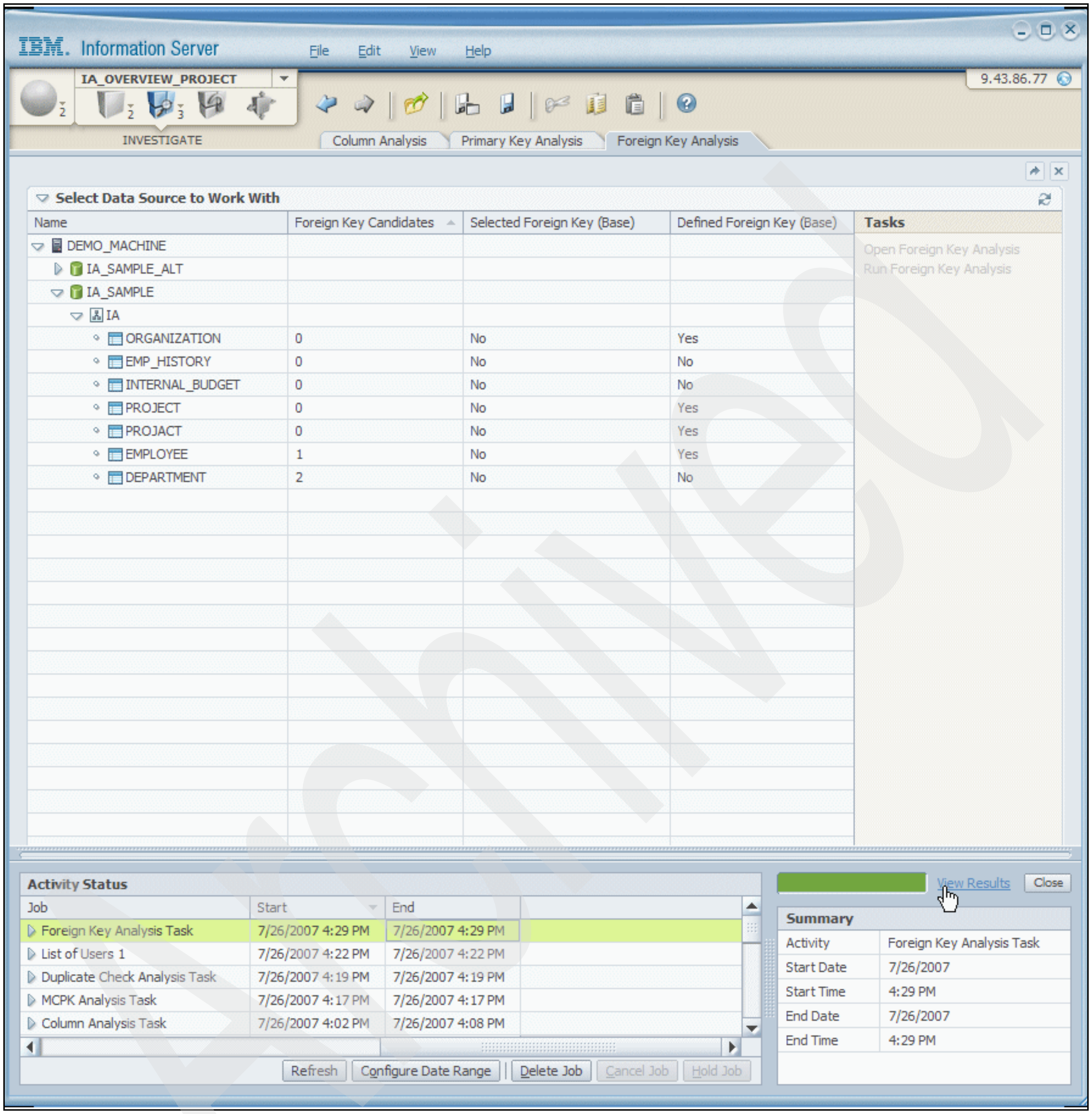
Start Date: 7/26/2007 16:24:38

End Date: 7/26/2007 16:24:38

Frequency: (Not set)

Repeat: 0

Figure 1-208 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 2/9



Select Data Source to Work With

Name	Foreign Key Candidates	Selected Foreign Key (Base)	Defined Foreign Key (Base)
DEMO_MACHINE			
IA_SAMPLE_ALT			
IA_SAMPLE			
IA			
ORGANIZATION	0	No	Yes
EMP_HISTORY	0	No	No
INTERNAL_BUDGET	0	No	No
PROJECT	0	No	Yes
PROJACT	0	No	Yes
EMPLOYEE	1	No	Yes
DEPARTMENT	2	No	No

Tasks

Open Foreign Key Analysis
Run Foreign Key Analysis

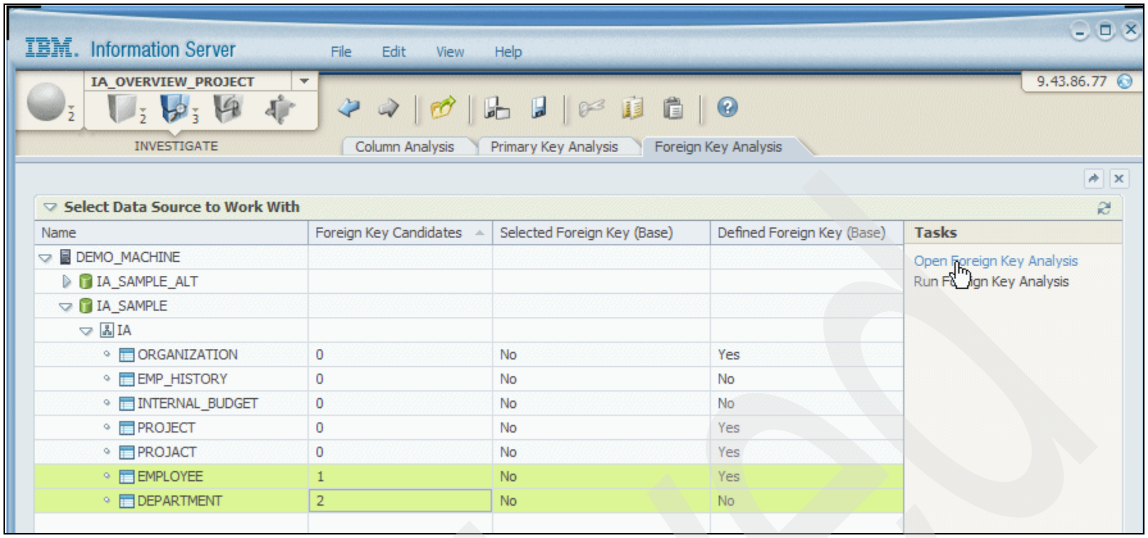
Activity Status

Job	Start	End
Foreign Key Analysis Task	7/26/2007 4:29 PM	7/26/2007 4:29 PM
List of Users 1	7/26/2007 4:22 PM	7/26/2007 4:22 PM
Duplicate Check Analysis Task	7/26/2007 4:19 PM	7/26/2007 4:19 PM
MCPK Analysis Task	7/26/2007 4:17 PM	7/26/2007 4:17 PM
Column Analysis Task	7/26/2007 4:02 PM	7/26/2007 4:08 PM

Summary

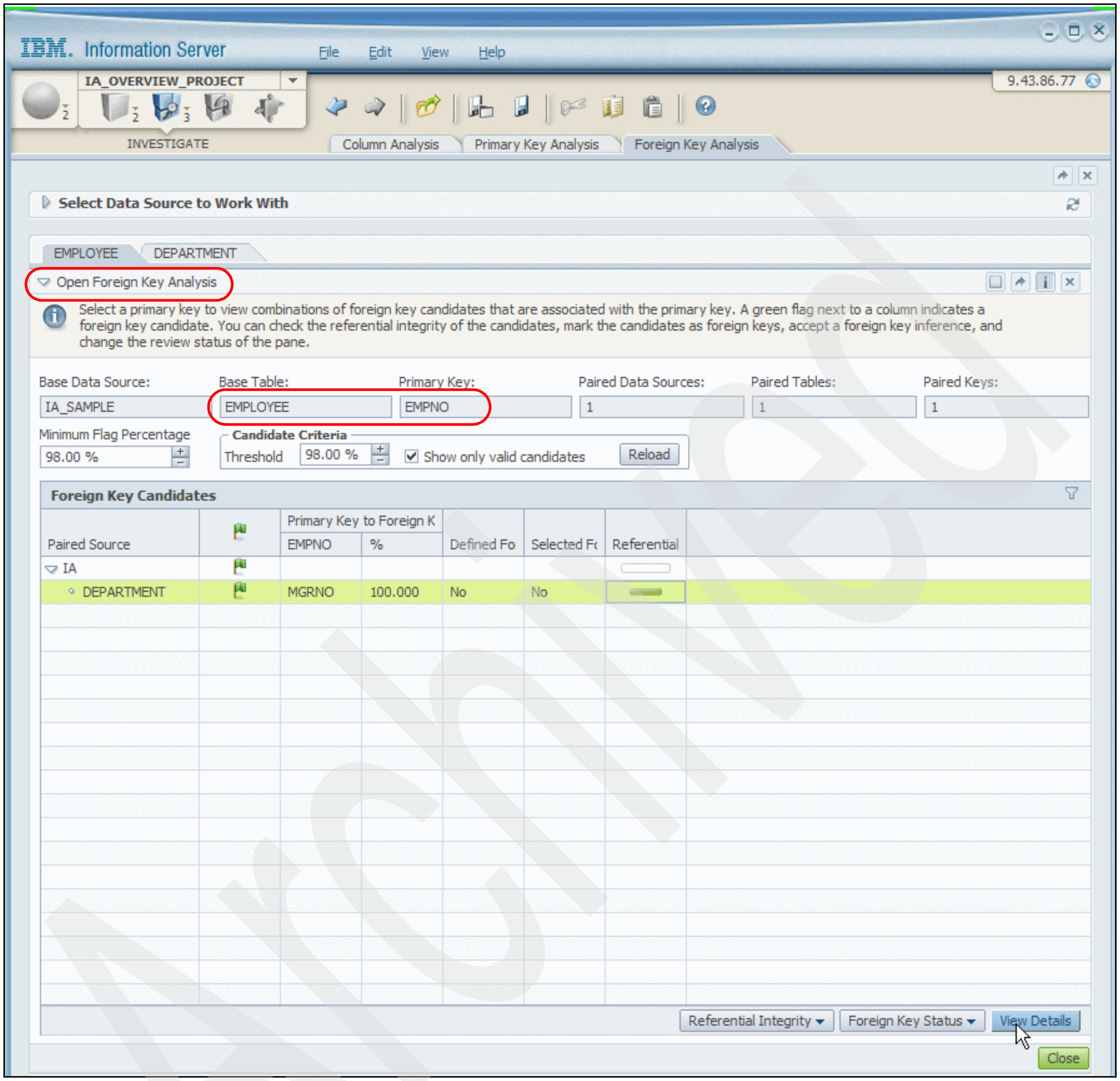
Activity	Foreign Key Analysis Task
Start Date	7/26/2007
Start Time	4:29 PM
End Date	7/26/2007
End Time	4:29 PM

Figure 1-209 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 3/9



Name	Foreign Key Candidates	Selected Foreign Key (Base)	Defined Foreign Key (Base)
DEMO_MACHINE			
IA_SAMPLE_ALT			
IA_SAMPLE			
IA			
ORGANIZATION	0	No	Yes
EMP_HISTORY	0	No	No
INTERNAL_BUDGET	0	No	No
PROJECT	0	No	Yes
PROJACT	0	No	Yes
EMPLOYEE	1	No	Yes
DEPARTMENT	2	No	No

Figure 1-210   Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 4/9



Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

Base Data Source: IA_SAMPLE

Base Table: EMPLOYEE

Primary Key: EMPNO

Paired Data Sources: 1

Paired Tables: 1

Paired Keys: 1

Minimum Flag Percentage: 98.00 %

Candidate Criteria

Threshold: 98.00 %

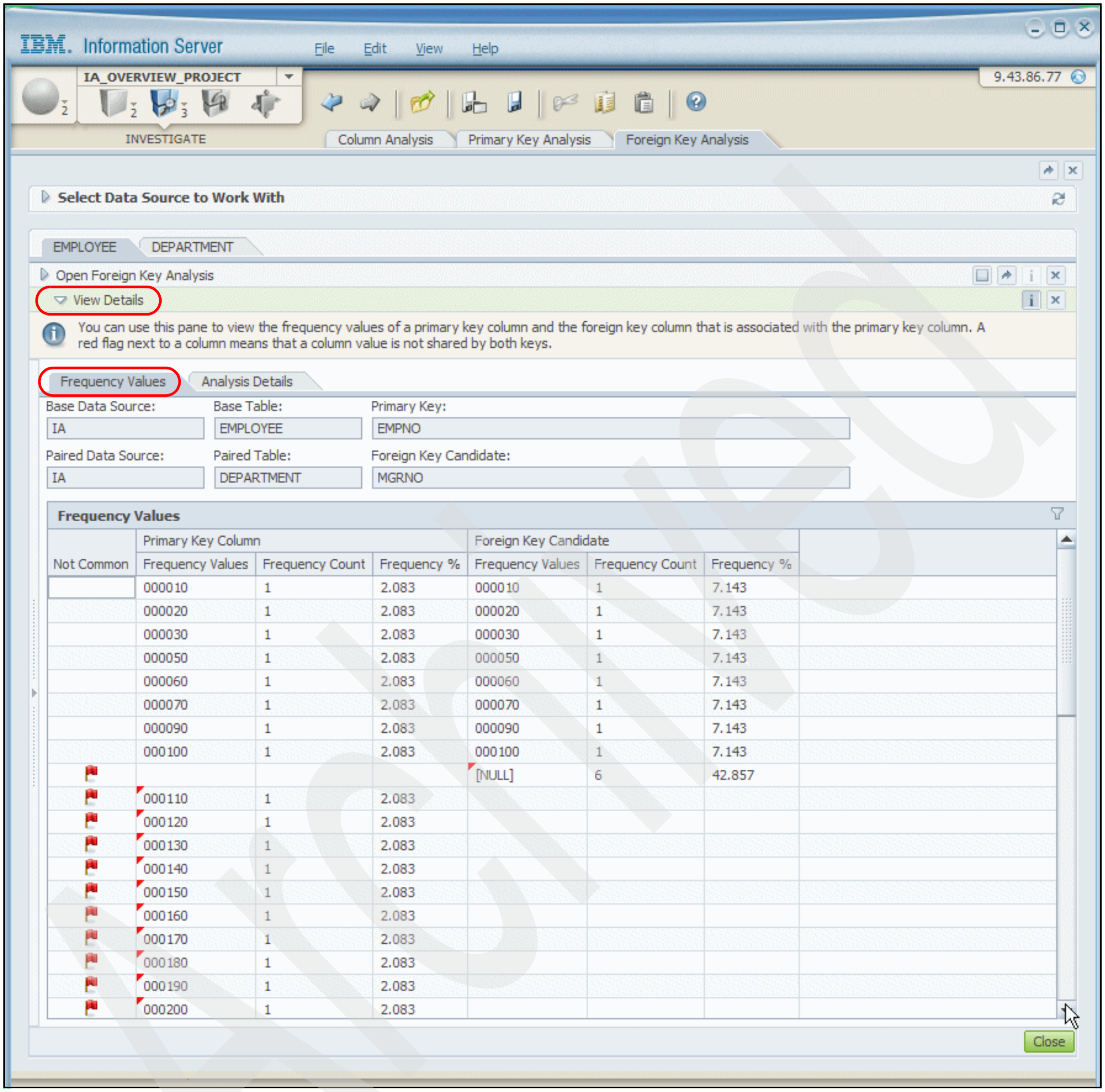
☑ Show only valid candidates

Reload

Foreign Key Candidates

		Primary Key to Foreign K				
Paired Source		EMPNO	%	Defined Fo	Selected Fo	Referential
IA	⚑					
DEPARTMENT	⚑	MGRNO	100.000	No	No	

Figure 1-211 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 5/9



You can use this pane to view the frequency values of a primary key column and the foreign key column that is associated with the primary key column. A red flag next to a column means that a column value is not shared by both keys.

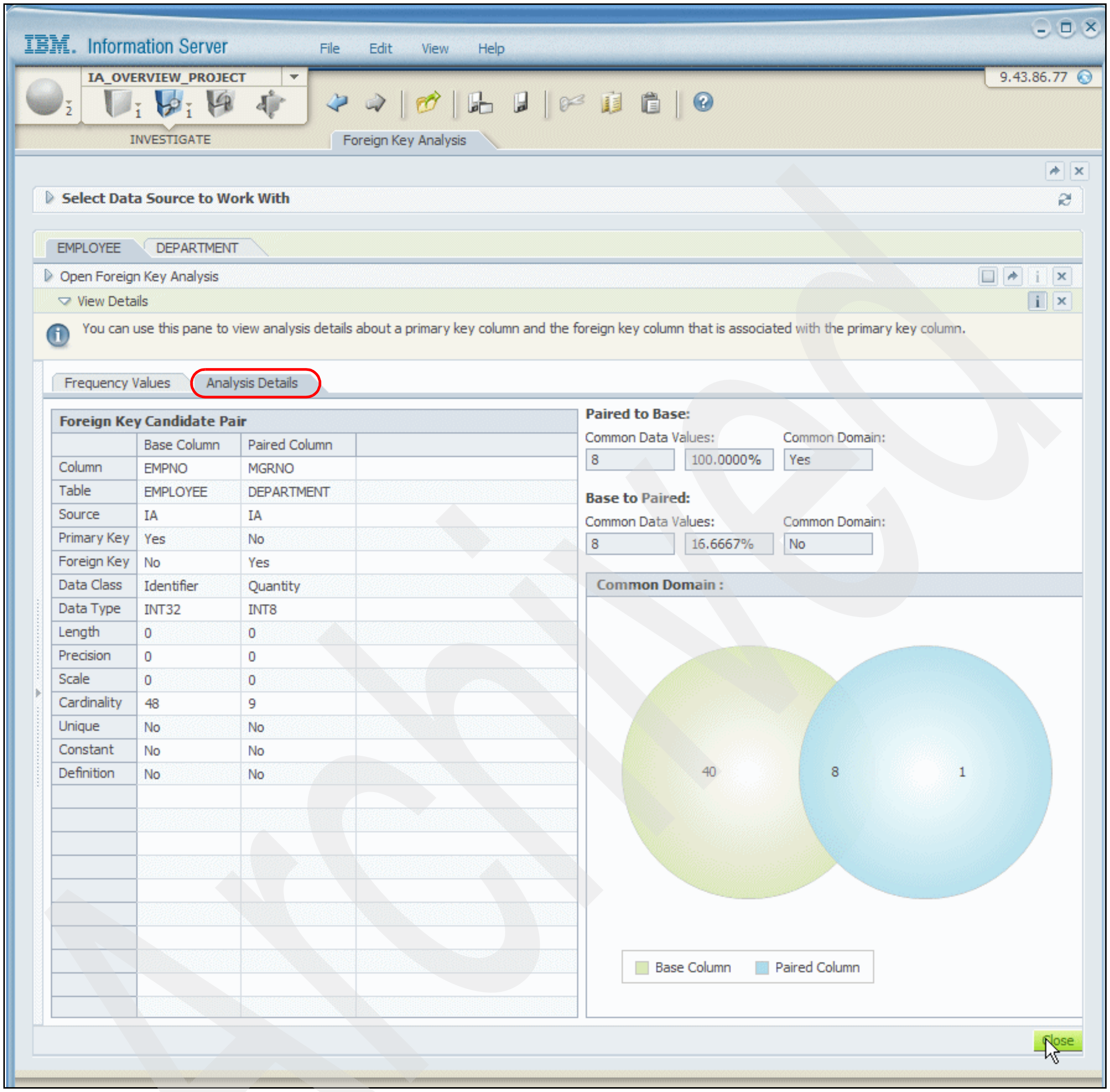
Base Data Source: IA
Base Table: EMPLOYEE
Primary Key: EMPNO

Paired Data Source: IA
Paired Table: DEPARTMENT
Foreign Key Candidate: MGRNO

Frequency Values

	Primary Key Column			Foreign Key Candidate		
Not Common	Frequency Values	Frequency Count	Frequency %	Frequency Values	Frequency Count	Frequency %
	000010	1	2.083	000010	1	7.143
	000020	1	2.083	000020	1	7.143
	000030	1	2.083	000030	1	7.143
	000050	1	2.083	000050	1	7.143
	000060	1	2.083	000060	1	7.143
	000070	1	2.083	000070	1	7.143
	000090	1	2.083	000090	1	7.143
	000100	1	2.083	000100	1	7.143
⚑				[NULL]	6	42.857
⚑	000110	1	2.083			
⚑	000120	1	2.083			
⚑	000130	1	2.083			
⚑	000140	1	2.083			
⚑	000150	1	2.083			
⚑	000160	1	2.083			
⚑	000170	1	2.083			
⚑	000180	1	2.083			
⚑	000190	1	2.083			
⚑	000200	1	2.083			

Figure 1-212 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 6/9



Select Data Source to Work With

EMPLOYEE | DEPARTMENT

Open Foreign Key Analysis

View Details

You can use this pane to view analysis details about a primary key column and the foreign key column that is associated with the primary key column.

Frequency Values | Analysis Details

Foreign Key Candidate Pair

	Base Column	Paired Column
Column	EMPNO	MGRNO
Table	EMPLOYEE	DEPARTMENT
Source	IA	IA
Primary Key	Yes	No
Foreign Key	No	Yes
Data Class	Identifier	Quantity
Data Type	INT32	INT8
Length	0	0
Precision	0	0
Scale	0	0
Cardinality	48	9
Unique	No	No
Constant	No	No
Definition	No	No

Paired to Base:

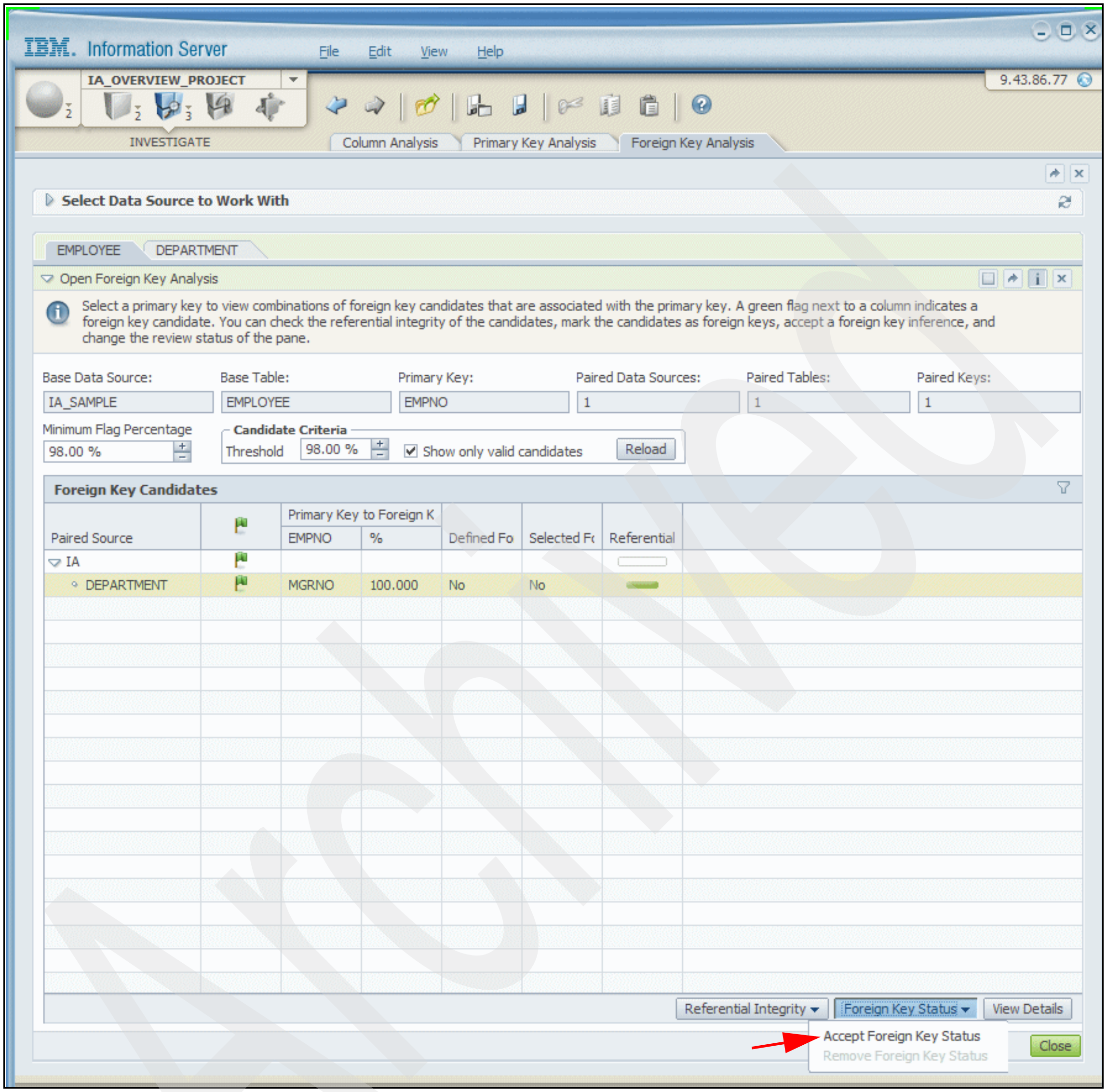
Common Data Values: 8 | 100.0000%
Common Domain: Yes

Base to Paired:

Common Data Values: 8 | 16.6667%
Common Domain: No

Common Domain :

Figure 1-213 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 7/9



Select Data Source to Work With

Open Foreign Key Analysis

Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

Base Data Source:	Base Table:	Primary Key:	Paired Data Sources:	Paired Tables:	Paired Keys:
IA_SAMPLE	EMPLOYEE	EMPNO	1	1	1

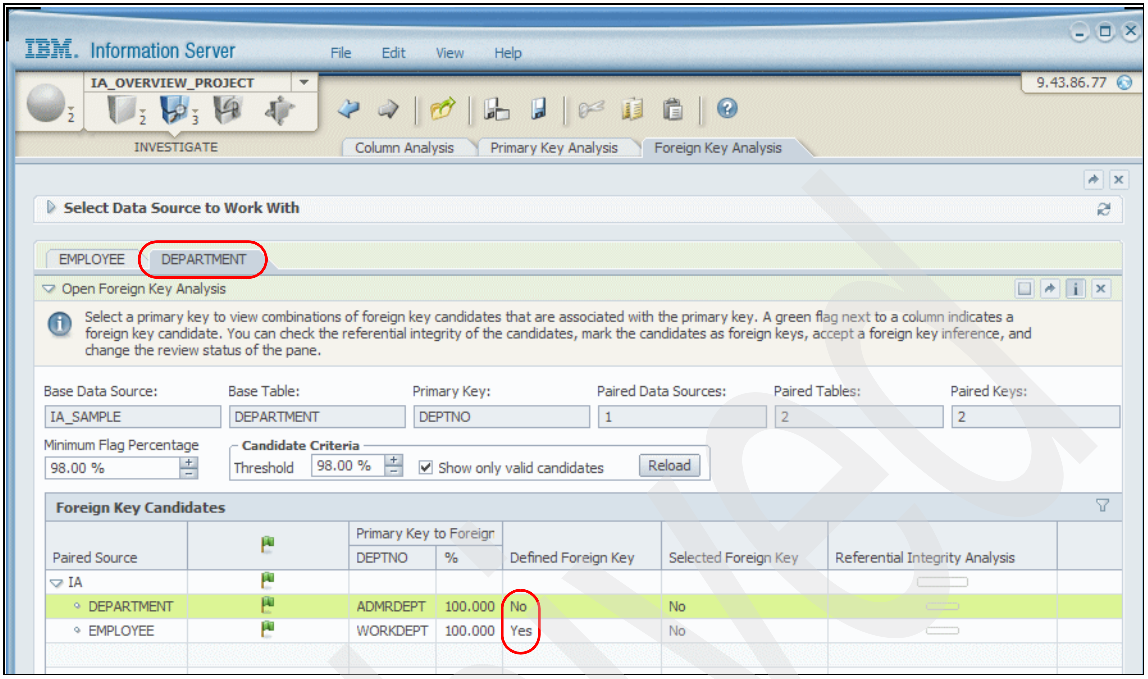
Minimum Flag Percentage: 98.00 %

Candidate Criteria: Threshold 98.00 % ☑ Show only valid candidates Reload

Foreign Key Candidates

Paired Source		Primary Key to Foreign K		Defined Fo	Selected Fo	Referential
	⚑	EMPNO	%			
IA	⚑					
DEPARTMENT	⚑	MGRNO	100.000	No	No	

Figure 1-214 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 8/9



Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

Base Data Source:	Base Table:	Primary Key:	Paired Data Sources:	Paired Tables:	Paired Keys:
IA_SAMPLE	DEPARTMENT	DEPTNO	1	2	2

Minimum Flag Percentage: 98.00 %

Candidate Criteria: Threshold 98.00 % ☑ Show only valid candidates

Foreign Key Candidates

Paired Source	⚑	Primary Key to Foreign DEPTNO	 %	Defined Foreign Key	Selected Foreign Key	Referential Integrity Analysis
IA	⚑					
DEPARTMENT	⚑	ADMRDEPT	100.000	No	No	
EMPLOYEE	⚑	WORKDEPT	100.000	Yes	No	

Figure 1-215 Run a foreign key analysis job on the DEPARTMENT and EMPLOYEE tables 9/9

Referential integrity analysis

You run a referential integrity analysis job to evaluate whether a relationship between the foreign keys and the primary keys is valid. Foreign key values that do not match a primary key value are identified as violations.

Referential integrity analysis calculates the total number of foreign key values and primary key values that are not common.

To determine the integrity of the foreign keys, open the project of interest (IA_OVERVIEW_PROJECT in our case) and run referential integrity analysis on a foreign key column as follows:

1. On the Investigate navigator menu in the console, select **Foreign Key Analysis** as shown in Figure 1-88 on page 138.
2. In the Foreign Key Analysis workspace, select the tables (EMPLOYEE and DEPARTMENT) that you want to run a referential integrity analysis job on. In the Tasks pane, click **Open Foreign Key Analysis** as shown in Figure 1-210 on page 280.[20]

[20] Assumes a Run Foreign Key Analysis job has previously been run the set of tables.

3. In the Open Foreign Key Analysis workspace, select the DEPARTMENT tab which shows the candidate foreign keys associated with the DEPTNO primary key column. There are two foreign key candidates (as shown in Figure 1-216 on page 288):

   – The ADMRDEPT column (not defined as a foreign key) in the DEPARTMENT table

   – The WORKDEPT column (defined as a foreign key) in the EMPLOYEE table

   Select the foreign key candidate on which you want to run analysis. This is the ADMRDEPT column in the DEPARTMENT table. Then select **Referential Integrity** → **Run Referential Integrity Analysis** to perform referential integrity analysis on the ADMRDEPT column in the DEPARTMENT table.

4. In the Run Referential Integrity Analysis pane in Figure 1-217 on page 289, type a name (Referential Integrity Analysis Task) for the analysis job, select **Run Now** to run the job now (or select Schedule to specify a time and date to run the job). Click **Submit and Close**.

5. After the job completes, in the Open Foreign Key Analysis pane under the DEPARTMENT tab, select the candidate foreign key ADMRDEPT and click **Referential Integrity** → **View Referential Integrity Analysis** as shown in Figure 1-218 on page 290.

   > **Attention:** You can try to get a larger list of candidate foreign keys by lowering the Candidate Criteria Threshold from 98% to a low value, such as 10%, if you do not see any candidate foreign keys in the results.

6. The View Referential Integrity Analysis pane under the DEPARTMENT tab in Figure 1-219 on page 291 shows the results of the referential integrity analysis job. It shows this view for the DEPTNO primary key column in the DEPARTMENT table and the candidate foreign key column ADMRDEPT in the DEPARTMENT table.

   – The foreign key to primary key portion of the analysis shows that there are no foreign key violations. All values in the ADMRDEPT column have a match in the DEPTNO column. It also shows that there are three distinct values in the ADMRDEPT column out of a total of 14 records.

   – The primary key to foreign key portion of the analysis shows that three values match a foreign key and 11 values in the primary key that do not match a foreign key.

   Because ADMRDEPT is not a defined foreign key column, you can choose to accept it as a foreign key by clicking **Foreign Key Status** → **Accept Foreign Key Status** (not shown here).

7. Figure 1-220 on page 292 shows the results of the referential integrity analysis job for the DEPTNO primary key column in the DEPARTMENT table, and the defined foreign key WORKDEPT column in the EMPLOYEE table.

   – The foreign key to primary key portion of the analysis shows that there are no foreign key violations. All values in the WORKDEPT column have a match in the DEPTNO column. It also shows that there are eight distinct values in the WORKDEPT column out of a total of 48 records.

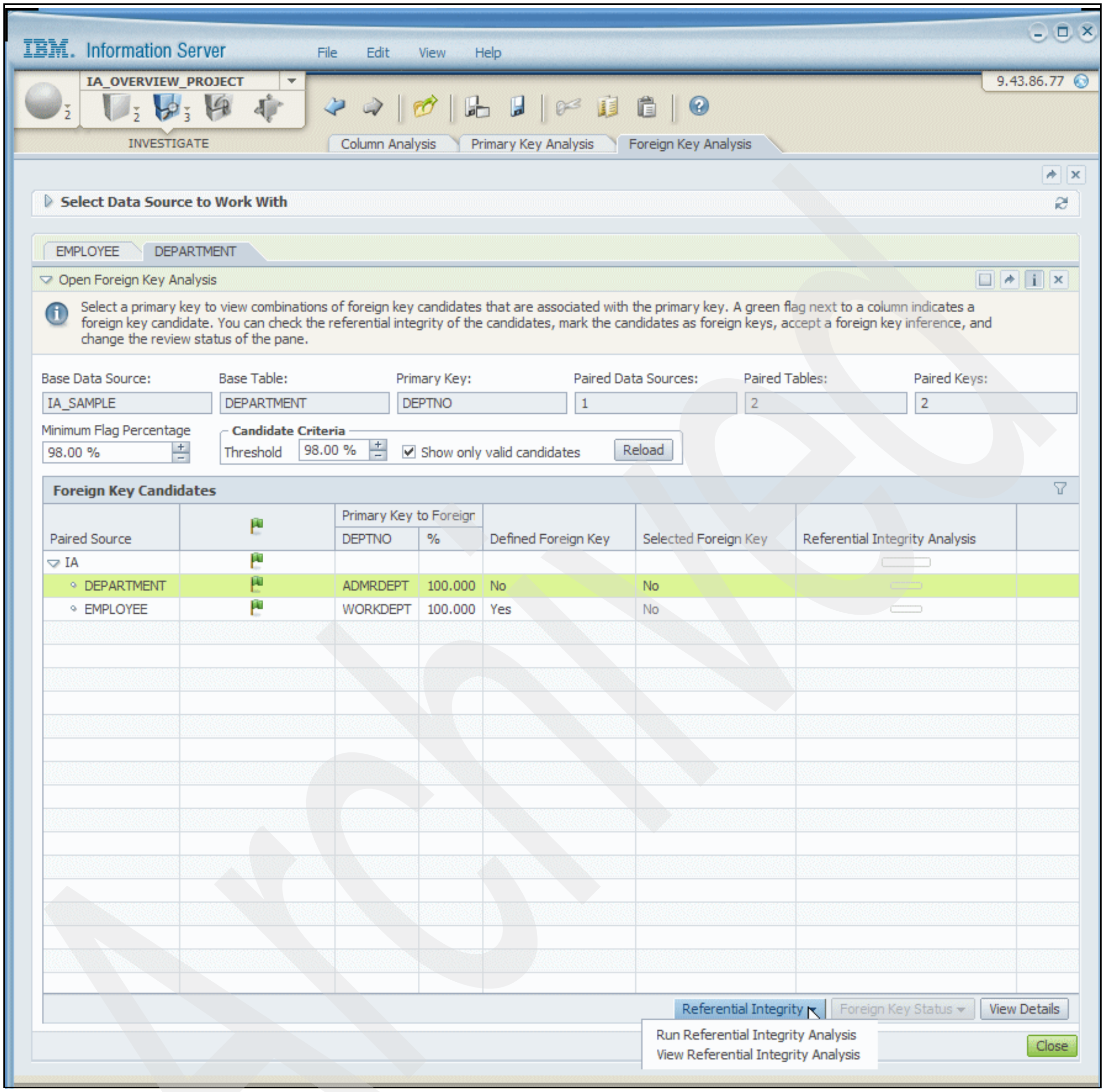
   – The primary key to foreign key portion of the analysis shows that eight values match a foreign key and six values in the primary key that do not match a foreign key.

   Because WORKDEPT is a defined foreign key column, you can choose to remove it as a foreign key by clicking **Foreign Key Status** → **Remove Foreign Key Status** (not shown here).

   Click **Close.**

8. Accept ADMRDEPT in the DEPARTMENT table as a foreign key by selecting it as shown in Figure 1-221 on page 293 and clicking **Foreign Key Status** → **Accept Foreign Key Status**.

9. Figure 1-222 on page 294 shows the status of the two candidate foreign keys as having been selected. They have a value of Yes.



Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

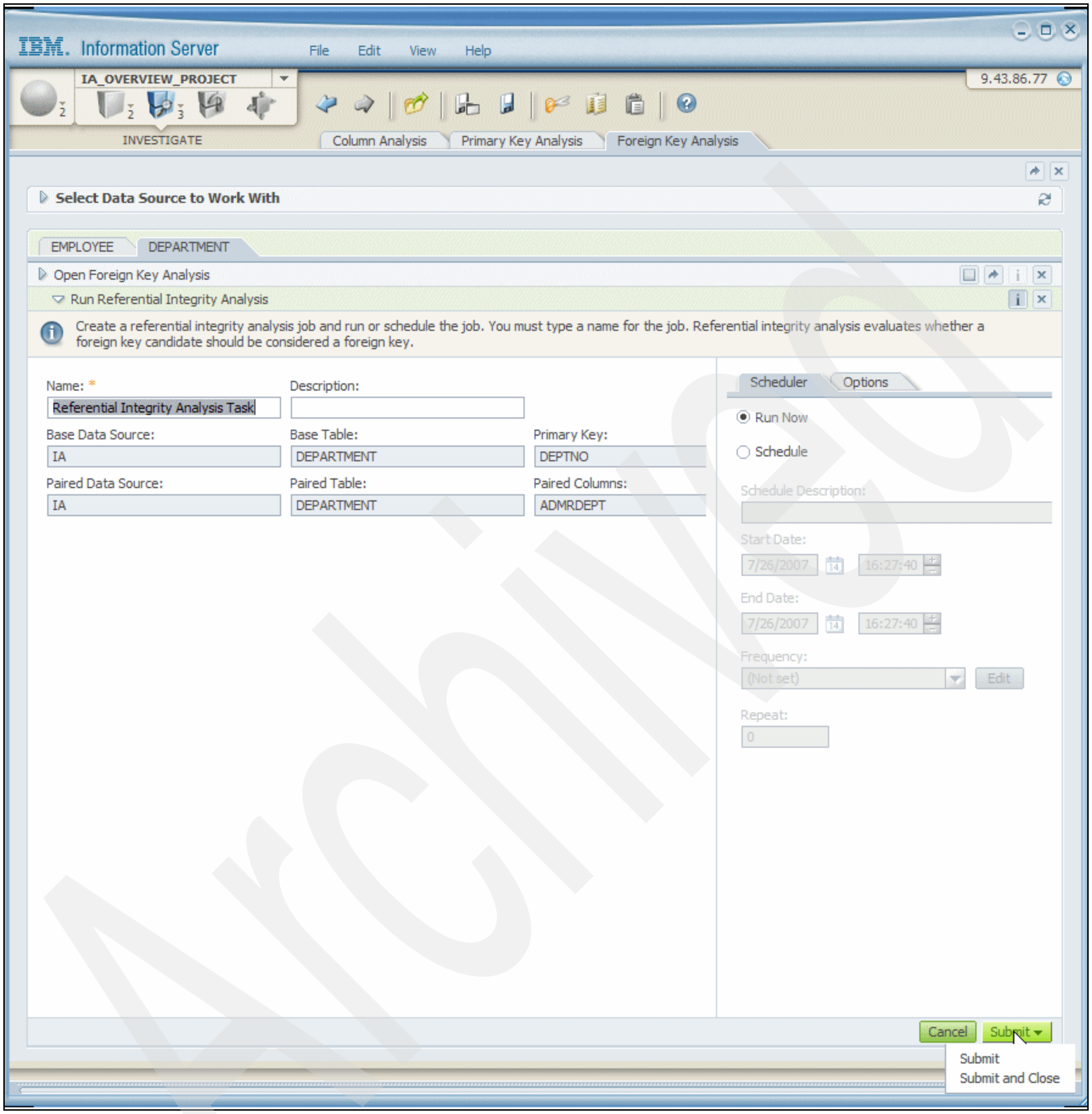
Base Data Source:	Base Table:	Primary Key:	Paired Data Sources:	Paired Tables:	Paired Keys:
IA_SAMPLE	DEPARTMENT	DEPTNO	1	2	2

Minimum Flag Percentage: 98.00 %

Candidate Criteria: Threshold 98.00 % ☑ Show only valid candidates

Paired Source	🏴	Primary Key to Foreign DEPTNO	%	Defined Foreign Key	Selected Foreign Key	Referential Integrity Analysis
IA	🏴					
DEPARTMENT	🏴	ADMRDEPT	100.000	No	No	
EMPLOYEE	🏴	WORKDEPT	100.000	Yes	No	

Figure 1-216 Run referential integrity analysis on the DEPARTMENT table foreign key candidates 1/7



Create a referential integrity analysis job and run or schedule the job. You must type a name for the job. Referential integrity analysis evaluates whether a foreign key candidate should be considered a foreign key.

Name: * Referential Integrity Analysis Task

Description:

Base Data Source: IA

Base Table: DEPARTMENT

Primary Key: DEPTNO

Paired Data Source: IA

Paired Table: DEPARTMENT

Paired Columns: ADMRDEPT

● Run Now

○ Schedule

Schedule Description:

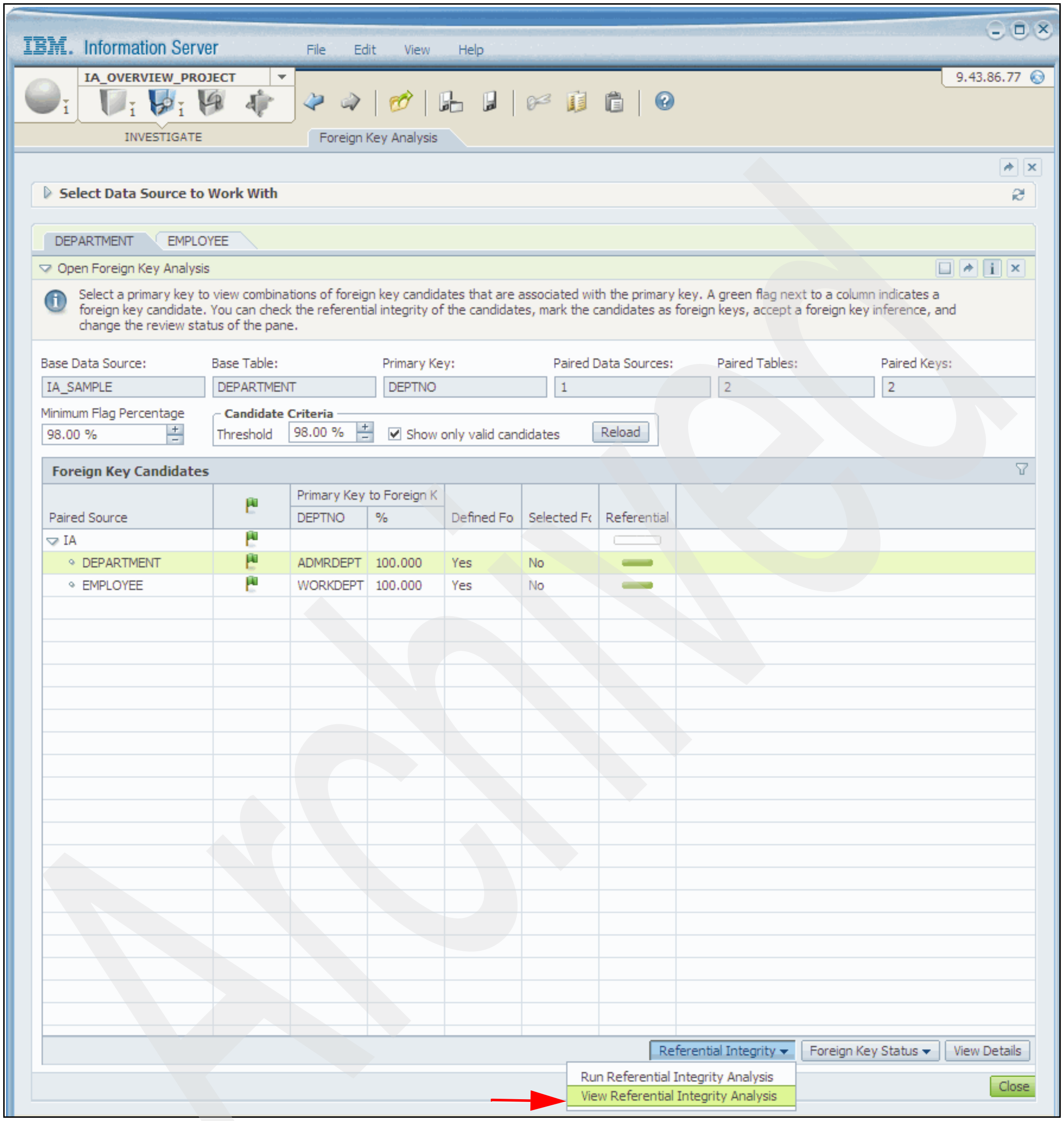
Start Date: 7/26/2007 16:27:40

End Date: 7/26/2007 16:27:40

Frequency: (Not set) Edit

Repeat: 0

Figure 1-217 Run referential integrity analysis on the DEPARTMENT table foreign key candidates 2/7



Select Data Source to Work With

Open Foreign Key Analysis

Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

Base Data Source:	Base Table:	Primary Key:	Paired Data Sources:	Paired Tables:	Paired Keys:
IA_SAMPLE	DEPARTMENT	DEPTNO	1	2	2

Minimum Flag Percentage: 98.00 %

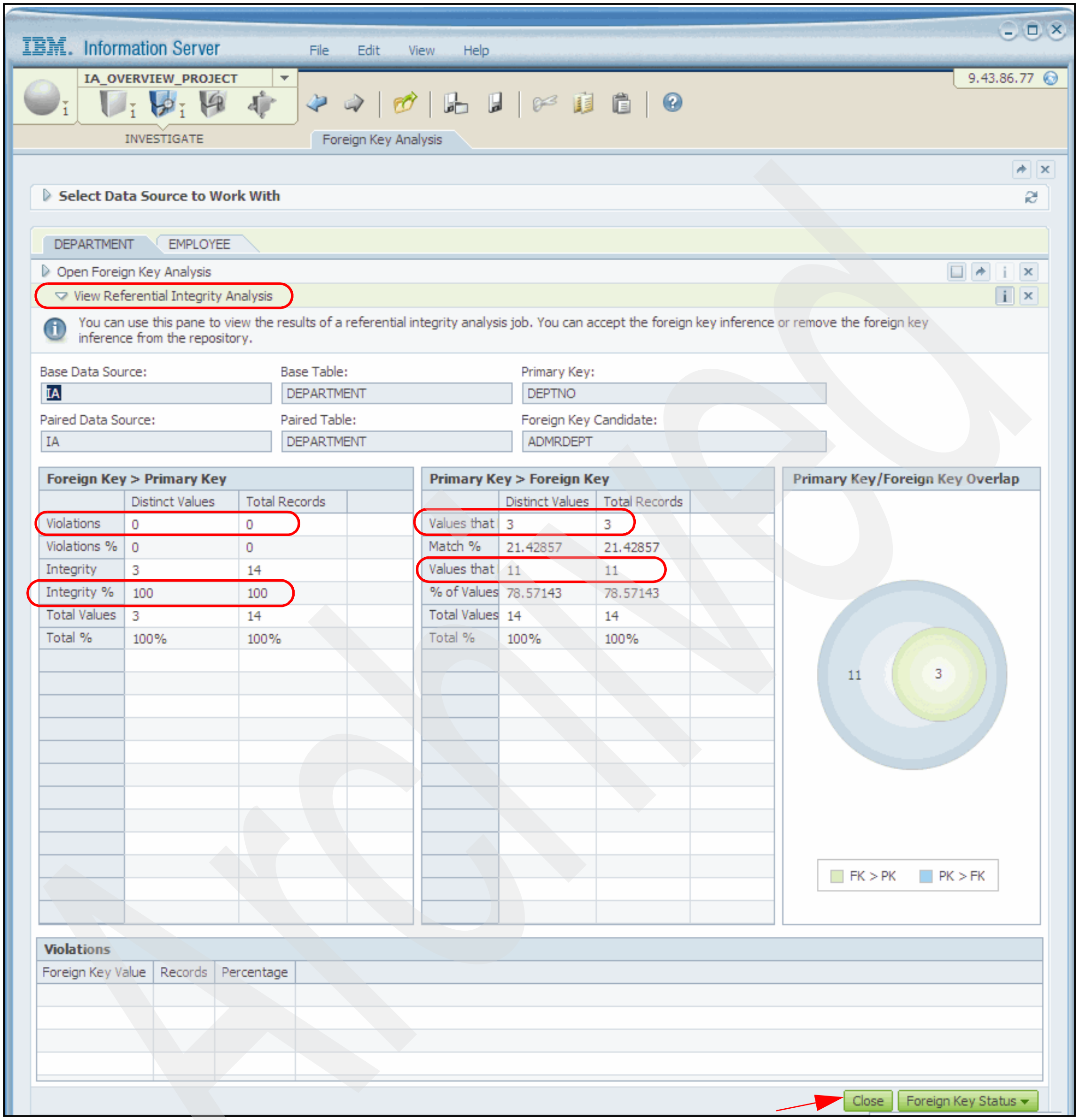
Candidate Criteria: Threshold 98.00 % ☑ Show only valid candidates Reload

Foreign Key Candidates

Paired Source	⚑	Primary Key to Foreign K		Defined Fo	Selected Fo	Referential
		DEPTNO	%			
IA	⚑					
DEPARTMENT	⚑	ADMRDEPT	100.000	Yes	No	
EMPLOYEE	⚑	WORKDEPT	100.000	Yes	No	

Referential Integrity
Foreign Key Status
View Details
Run Referential Integrity Analysis
View Referential Integrity Analysis
Close

Figure 1-218 Run referential integrity analysis on the DEPARTMENT table foreign key candidates 3/7



You can use this pane to view the results of a referential integrity analysis job. You can accept the foreign key inference or remove the foreign key inference from the repository.

Base Data Source:	Base Table:	Primary Key:
IA	DEPARTMENT	DEPTNO
Paired Data Source:	Paired Table:	Foreign Key Candidate:
IA	DEPARTMENT	ADMRDEPT

Foreign Key > Primary Key

	Distinct Values	Total Records
Violations	0	0
Violations %	0	0
Integrity	3	14
Integrity %	100	100
Total Values	3	14
Total %	100%	100%

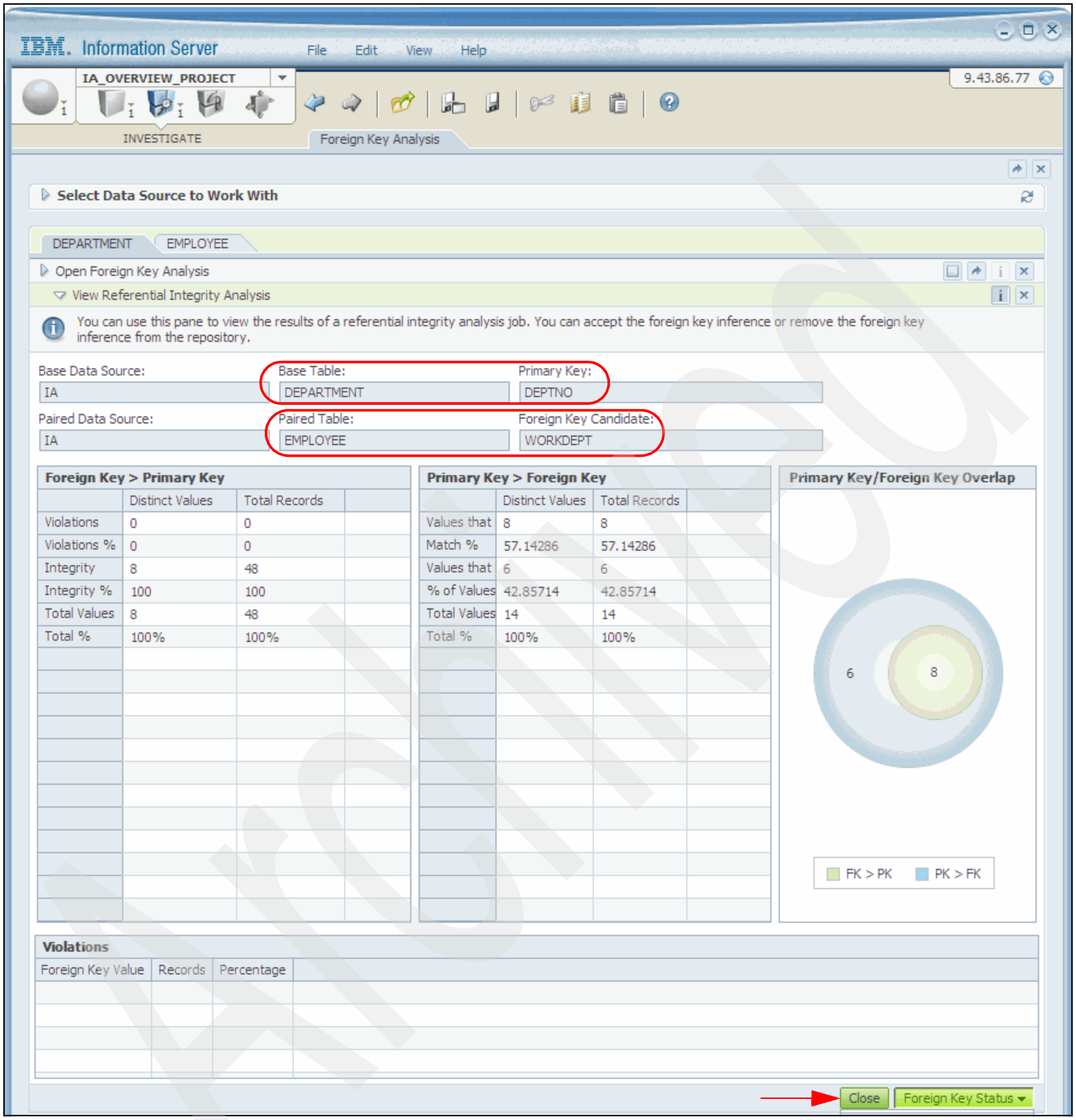
Primary Key > Foreign Key

	Distinct Values	Total Records
Values that	3	3
Match %	21.42857	21.42857
Values that	11	11
% of Values	78.57143	78.57143
Total Values	14	14
Total %	100%	100%

Violations

Foreign Key Value	Records	Percentage

Figure 1-219 Run referential integrity analysis on the DEPARTMENT table foreign key candidates 4/7



You can use this pane to view the results of a referential integrity analysis job. You can accept the foreign key inference or remove the foreign key inference from the repository.

Base Data Source: IA
Base Table: DEPARTMENT
Primary Key: DEPTNO
Paired Data Source: IA
Paired Table: EMPLOYEE
Foreign Key Candidate: WORKDEPT

Foreign Key > Primary Key

	Distinct Values	Total Records
Violations	0	0
Violations %	0	0
Integrity	8	48
Integrity %	100	100
Total Values	8	48
Total %	100%	100%

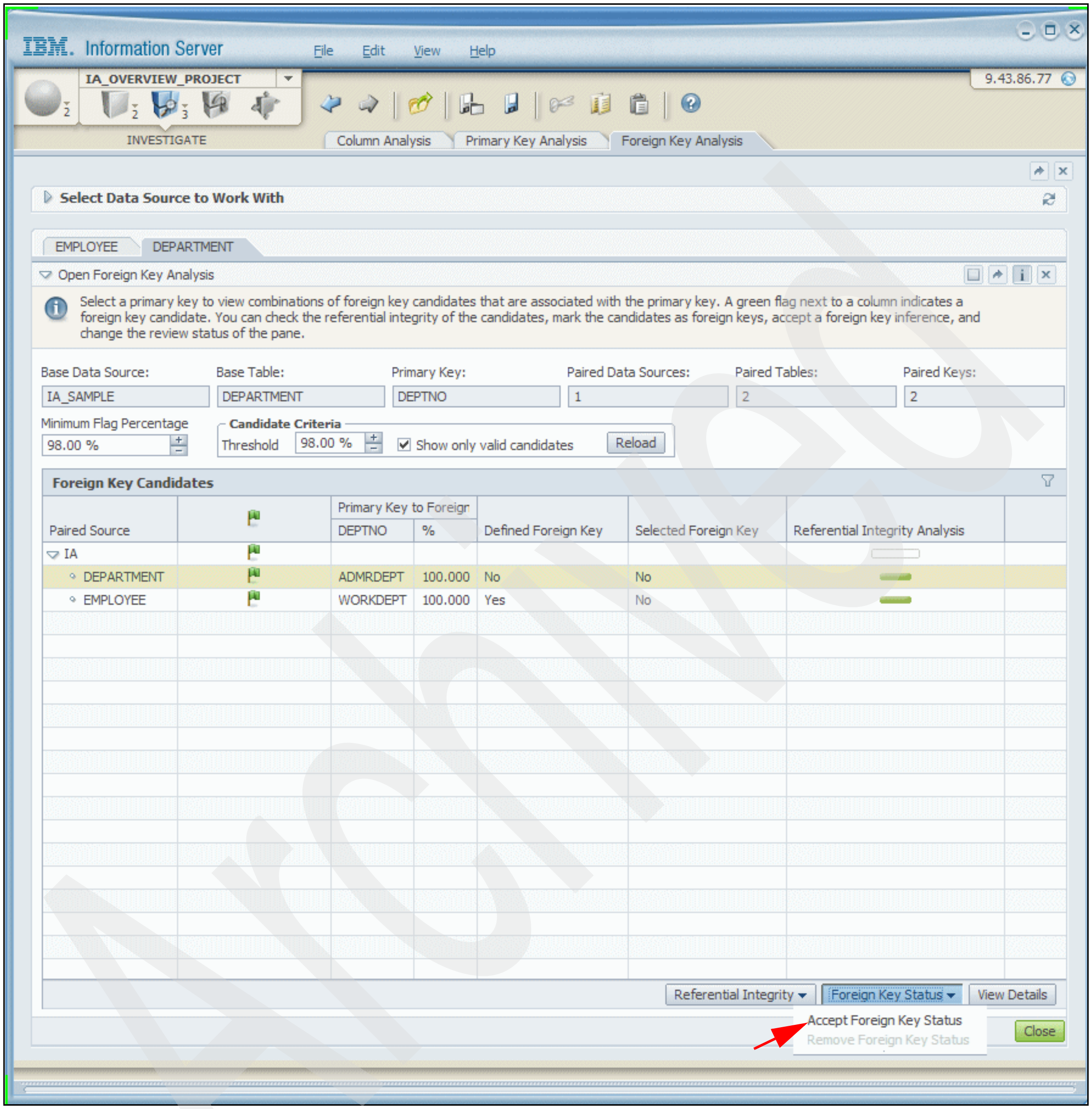
Primary Key > Foreign Key

	Distinct Values	Total Records
Values that	8	8
Match %	57.14286	57.14286
Values that	6	6
% of Values	42.85714	42.85714
Total Values	14	14
Total %	100%	100%

Violations

Foreign Key Value	Records	Percentage

Figure 1-220 Run referential integrity analysis on the DEPARTMENT table foreign key candidates 5/7



Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

Base Data Source:	Base Table:	Primary Key:	Paired Data Sources:	Paired Tables:	Paired Keys:
IA_SAMPLE	DEPARTMENT	DEPTNO	1	2	2

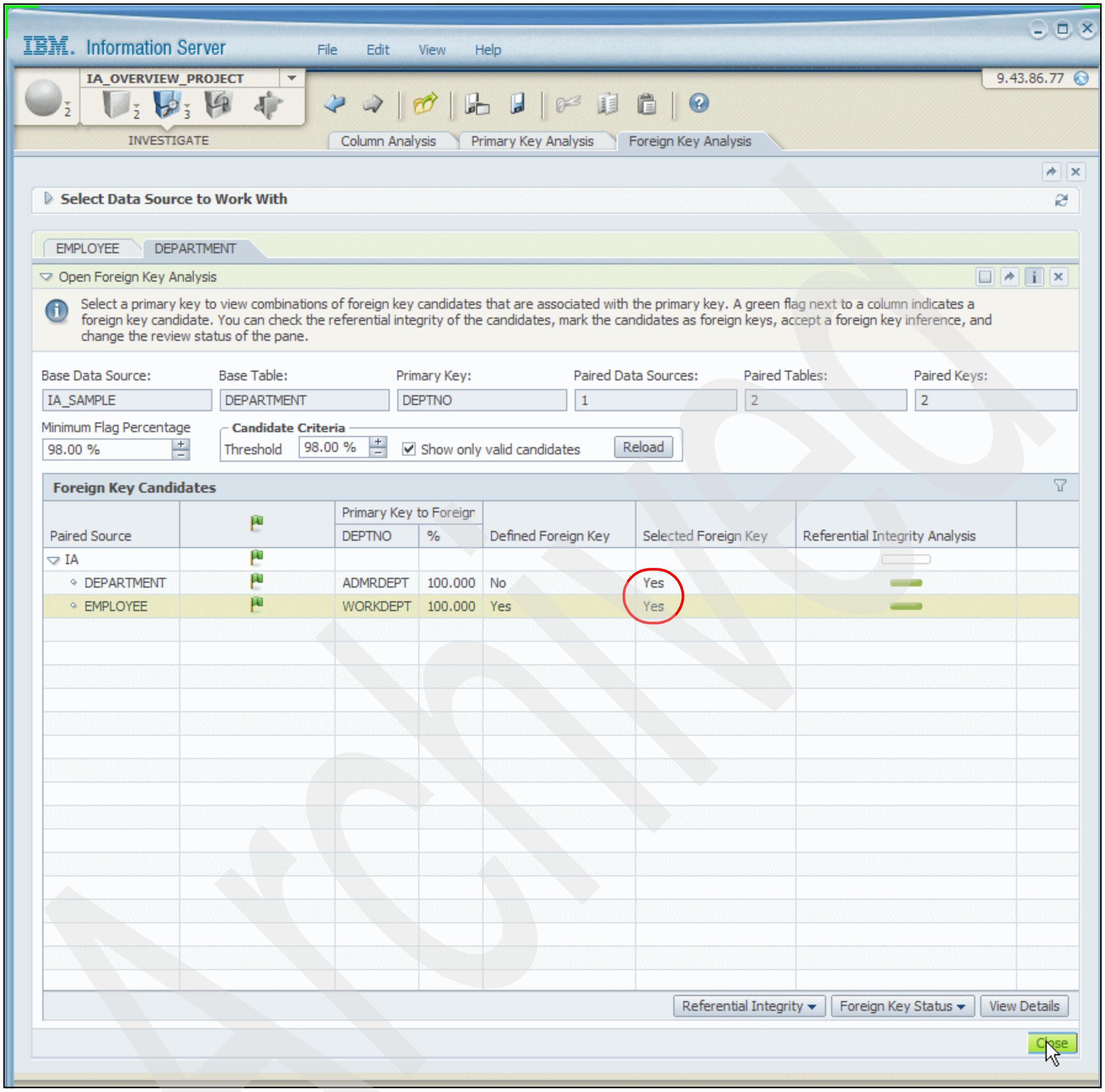
Minimum Flag Percentage: 98.00 %

Candidate Criteria: Threshold 98.00 % ☑ Show only valid candidates Reload

Foreign Key Candidates

Paired Source	🚩	Primary Key to Foreign		Defined Foreign Key	Selected Foreign Key	Referential Integrity Analysis
		DEPTNO	%			
IA	🚩					
DEPARTMENT	🚩	ADMRDEPT	100.000	No	No	
EMPLOYEE	🚩	WORKDEPT	100.000	Yes	No	

Figure 1-221 Run referential integrity analysis on the DEPARTMENT table foreign key candidates 6/7



Select a primary key to view combinations of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a foreign key candidate. You can check the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and change the review status of the pane.

Base Data Source: IA_SAMPLE

Base Table: DEPARTMENT

Primary Key: DEPTNO

Paired Data Sources: 1

Paired Tables: 2

Paired Keys: 2

Minimum Flag Percentage: 98.00 %

Candidate Criteria

Threshold: 98.00 %

☑ Show only valid candidates

Foreign Key Candidates

Paired Source		Primary Key to Foreign		Defined Foreign Key	Selected Foreign Key	Referential Integrity Analysis
		DEPTNO	%			
IA	🚩					
DEPARTMENT	🚩	ADMRDEPT	100.000	No	Yes	
EMPLOYEE	🚩	WORKDEPT	100.000	Yes	Yes	

Figure 1-222 Run referential integrity analysis on the DEPARTMENT table foreign key candidates 7/7

Applying filters to foreign key analysis

When you run a foreign key analysis against a set of tables (each of which can have a large number of columns defined) as shown in Figure 1-223 on page 296, you can be presented with a very large number of candidate foreign keys as shown in Figure 1-224 on page 297.

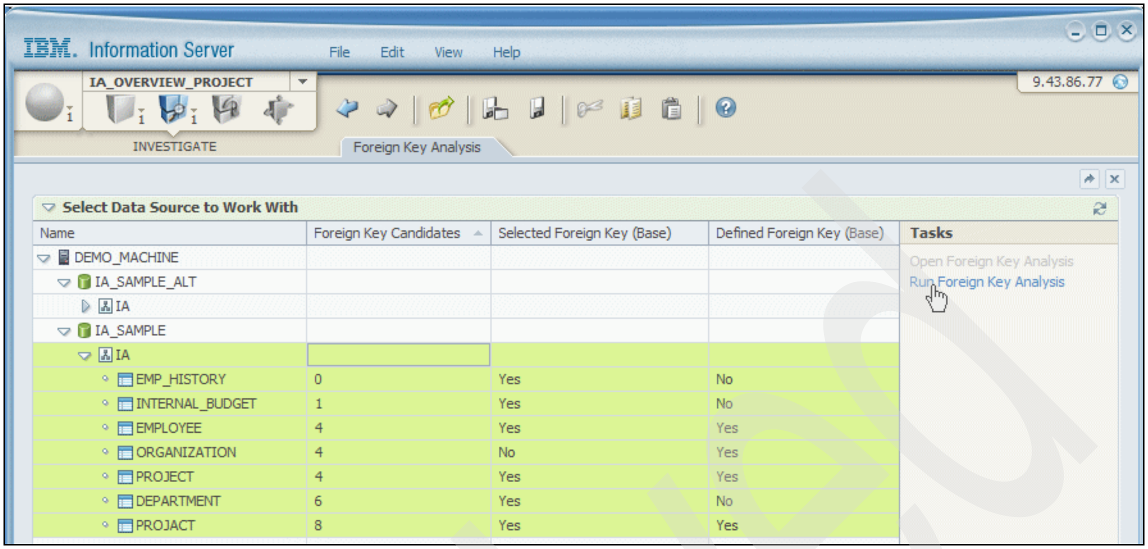
You might want to eliminate some of the columns from consideration as foreign key candidates because they obviously do not apply from a semantic viewpoint (for example, a COMMENT or REMARKS column in a table).

This is achieved as follows:

1. On the Run Foreign Key Analysis pane shown in Figure 1-224 on page 297, click the filter icon (highlighted by a red circle and arrow). This action expands the Column Details section with Apply Filters and Select check boxes as shown in Figure 1-225 on page 298.

   > **Note:** The Column Details pane is split into two parts, a Base Column portion and a Paired Column portion. The Base Column and Paired Column is each identified by three fields: Source, Table, and Column.

2. From the Select drop-down list, choose the particular Base Column field to be used for filtering such as “Base Column - Table” as shown in Figure 1-225 on page 298.
3. Choose the “Is” condition to apply to the “Base Column - Table” from the drop-down list as shown in Figure 1-226 on page 299.
4. Key in DEPARTMENT as the condition to check for, which results in a refresh of the foreign key candidates in the Column Details pane that only include the DEPARTMENT value in the “Base Column Table field as shown in Figure 1-227 on page 300.
5. You can now proceed to run the foreign key analysis on these filtered foreign key candidates by clicking **Submit**.



Select Data Source to Work With

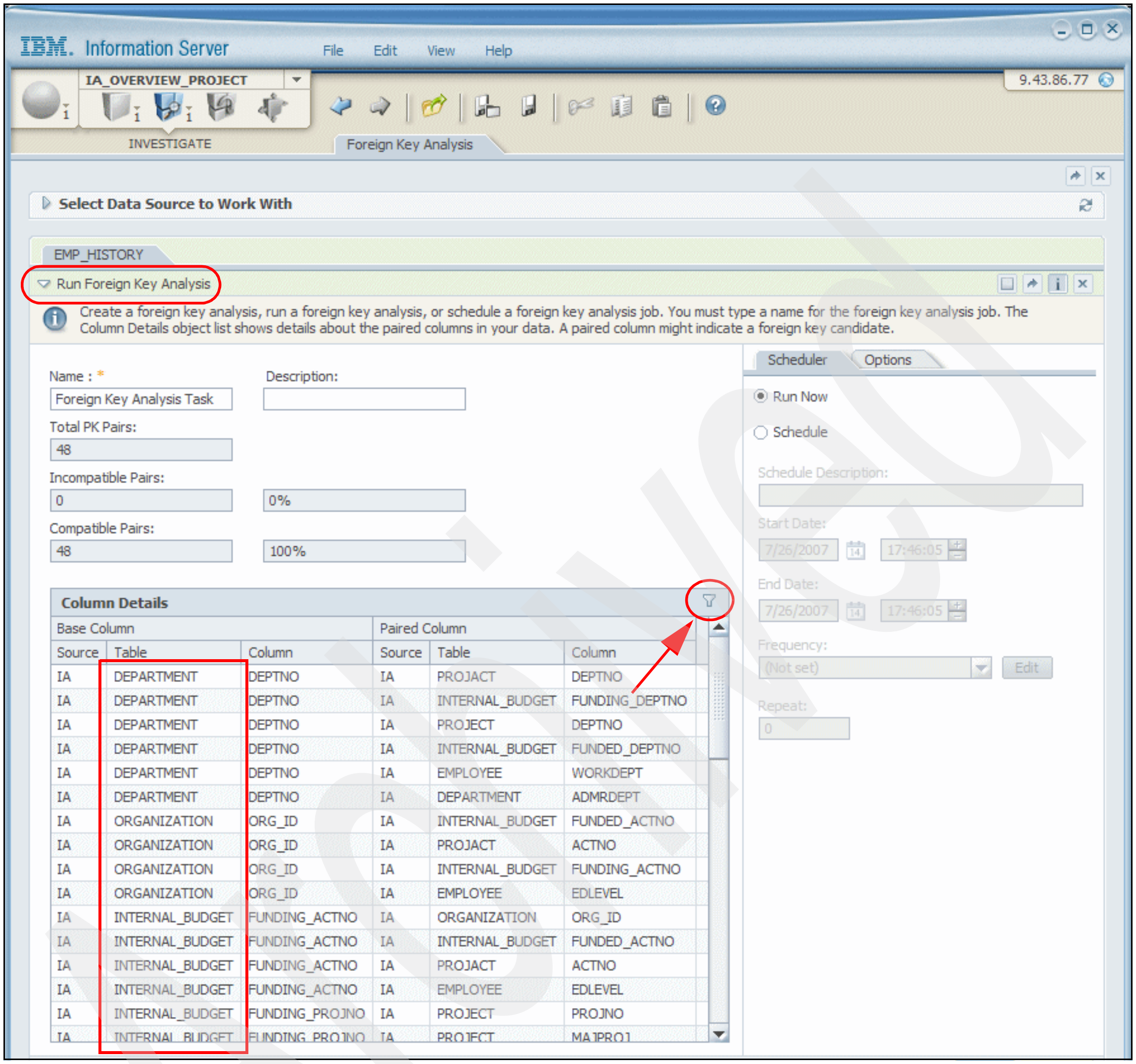
Name	Foreign Key Candidates	Selected Foreign Key (Base)	Defined Foreign Key (Base)
DEMO_MACHINE			
IA_SAMPLE_ALT			
IA			
IA_SAMPLE			
IA			
EMP_HISTORY	0	Yes	No
INTERNAL_BUDGET	1	Yes	No
EMPLOYEE	4	Yes	Yes
ORGANIZATION	4	No	Yes
PROJECT	4	Yes	Yes
DEPARTMENT	6	Yes	No
PROJACT	8	Yes	Yes

Tasks

Open Foreign Key Analysis

Run Foreign Key Analysis

Figure 1-223 Applying filters to foreign key analysis 1/5



Select Data Source to Work With

EMP_HISTORY

Run Foreign Key Analysis

Create a foreign key analysis, run a foreign key analysis, or schedule a foreign key analysis job. You must type a name for the foreign key analysis job. The Column Details object list shows details about the paired columns in your data. A paired column might indicate a foreign key candidate.

Name : * Foreign Key Analysis Task

Description:

Total PK Pairs: 48

Incompatible Pairs: 0 0%

Compatible Pairs: 48 100%

Scheduler | Options

● Run Now

○ Schedule

Schedule Description:

Start Date: 7/26/2007 17:46:05

End Date: 7/26/2007 17:46:05

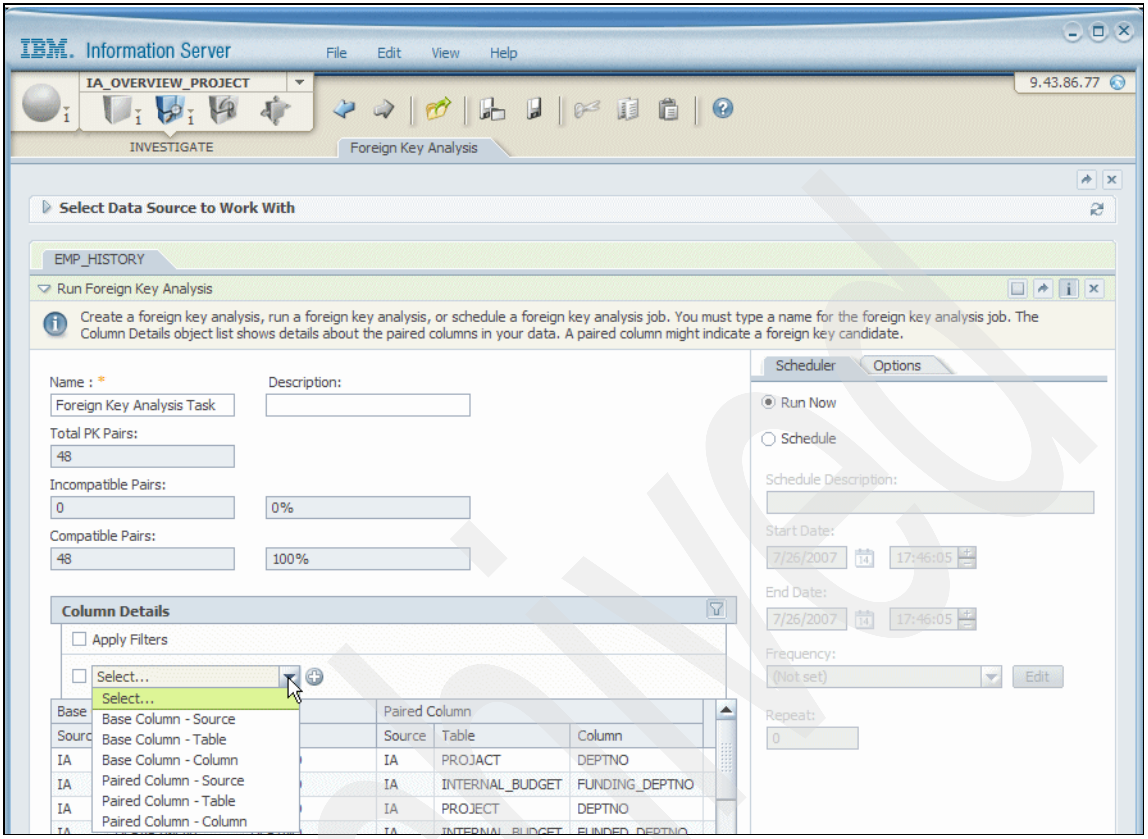
Frequency: (Not set) Edit

Repeat: 0

Column Details

Base Column			Paired Column		
Source	Table	Column	Source	Table	Column
IA	DEPARTMENT	DEPTNO	IA	PROJACT	DEPTNO
IA	DEPARTMENT	DEPTNO	IA	INTERNAL_BUDGET	FUNDING_DEPTNO
IA	DEPARTMENT	DEPTNO	IA	PROJECT	DEPTNO
IA	DEPARTMENT	DEPTNO	IA	INTERNAL_BUDGET	FUNDED_DEPTNO
IA	DEPARTMENT	DEPTNO	IA	EMPLOYEE	WORKDEPT
IA	DEPARTMENT	DEPTNO	IA	DEPARTMENT	ADMRDEPT
IA	ORGANIZATION	ORG_ID	IA	INTERNAL_BUDGET	FUNDED_ACTNO
IA	ORGANIZATION	ORG_ID	IA	PROJACT	ACTNO
IA	ORGANIZATION	ORG_ID	IA	INTERNAL_BUDGET	FUNDING_ACTNO
IA	ORGANIZATION	ORG_ID	IA	EMPLOYEE	EDLEVEL
IA	INTERNAL_BUDGET	FUNDING_ACTNO	IA	ORGANIZATION	ORG_ID
IA	INTERNAL_BUDGET	FUNDING_ACTNO	IA	INTERNAL_BUDGET	FUNDED_ACTNO
IA	INTERNAL_BUDGET	FUNDING_ACTNO	IA	PROJACT	ACTNO
IA	INTERNAL_BUDGET	FUNDING_ACTNO	IA	EMPLOYEE	EDLEVEL
IA	INTERNAL_BUDGET	FUNDING_PROJNO	IA	PROJECT	PROJNO
IA	INTERNAL_BUDGET	FUNDING_PROJNO	IA	PROJECT	MAJPROJ

Figure 1-224 Applying filters to foreign key analysis 2/5



Select Data Source to Work With

EMP_HISTORY

Run Foreign Key Analysis

Create a foreign key analysis, run a foreign key analysis, or schedule a foreign key analysis job. You must type a name for the foreign key analysis job. The Column Details object list shows details about the paired columns in your data. A paired column might indicate a foreign key candidate.

Name : * Foreign Key Analysis Task

Description:

Total PK Pairs: 48

Incompatible Pairs: 0 0%

Compatible Pairs: 48 100%

Scheduler Options

● Run Now

○ Schedule

Schedule Description:

Start Date: 7/26/2007 17:46:05

End Date: 7/26/2007 17:46:05

Frequency: (Not set) Edit

Repeat: 0

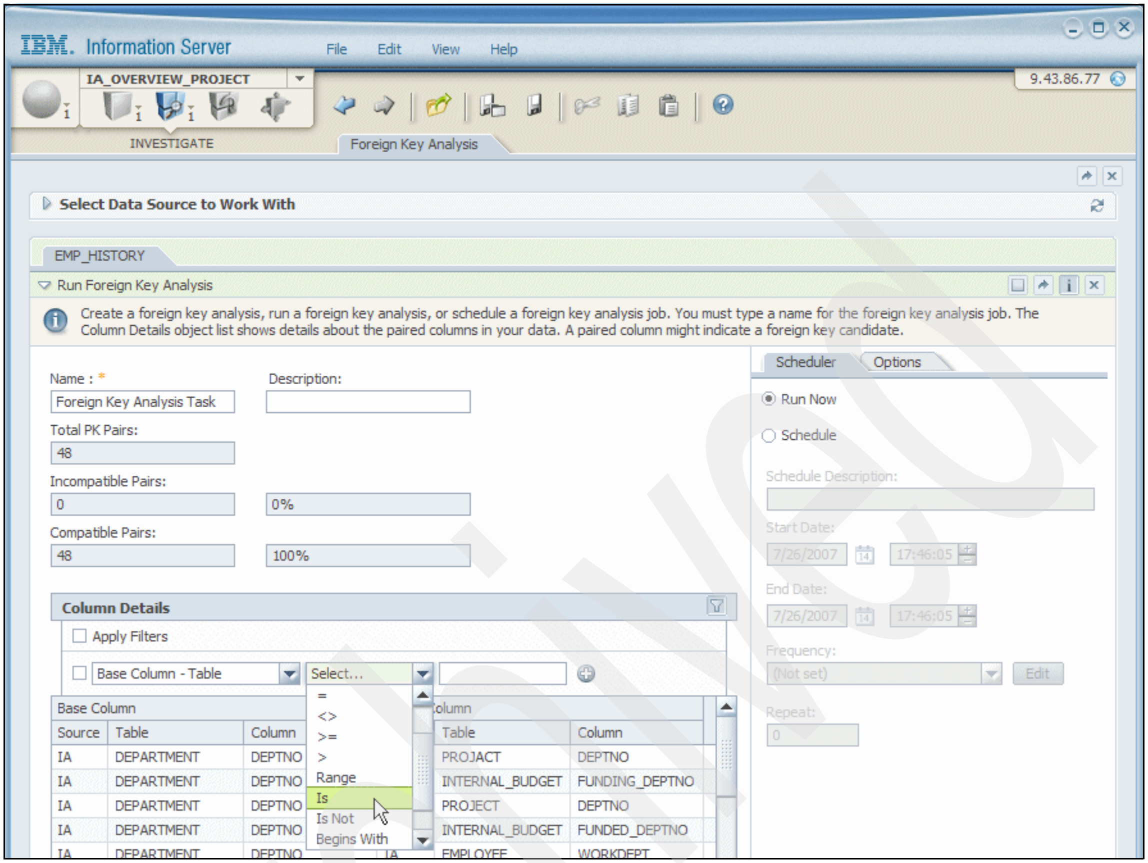
Column Details

□ Apply Filters

□ Select...

Base			Paired Column		
Sourc			Source	Table	Column
IA			IA	PROJACT	DEPTNO
IA			IA	INTERNAL_BUDGET	FUNDING_DEPTNO
IA			IA	PROJECT	DEPTNO
IA			IA	INTERNAL_BUDGET	FUNDED_DEPTNO

Figure 1-225 Applying filters to foreign key analysis 3/5



Create a foreign key analysis, run a foreign key analysis, or schedule a foreign key analysis job. You must type a name for the foreign key analysis job. The Column Details object list shows details about the paired columns in your data. A paired column might indicate a foreign key candidate.

Name : * Foreign Key Analysis Task

Description:

Total PK Pairs: 48

Incompatible Pairs: 0 0%

Compatible Pairs: 48 100%

● Run Now

○ Schedule

Schedule Description:

Start Date: 7/26/2007 17:46:05

End Date: 7/26/2007 17:46:05

Frequency: (Not set)

Repeat: 0

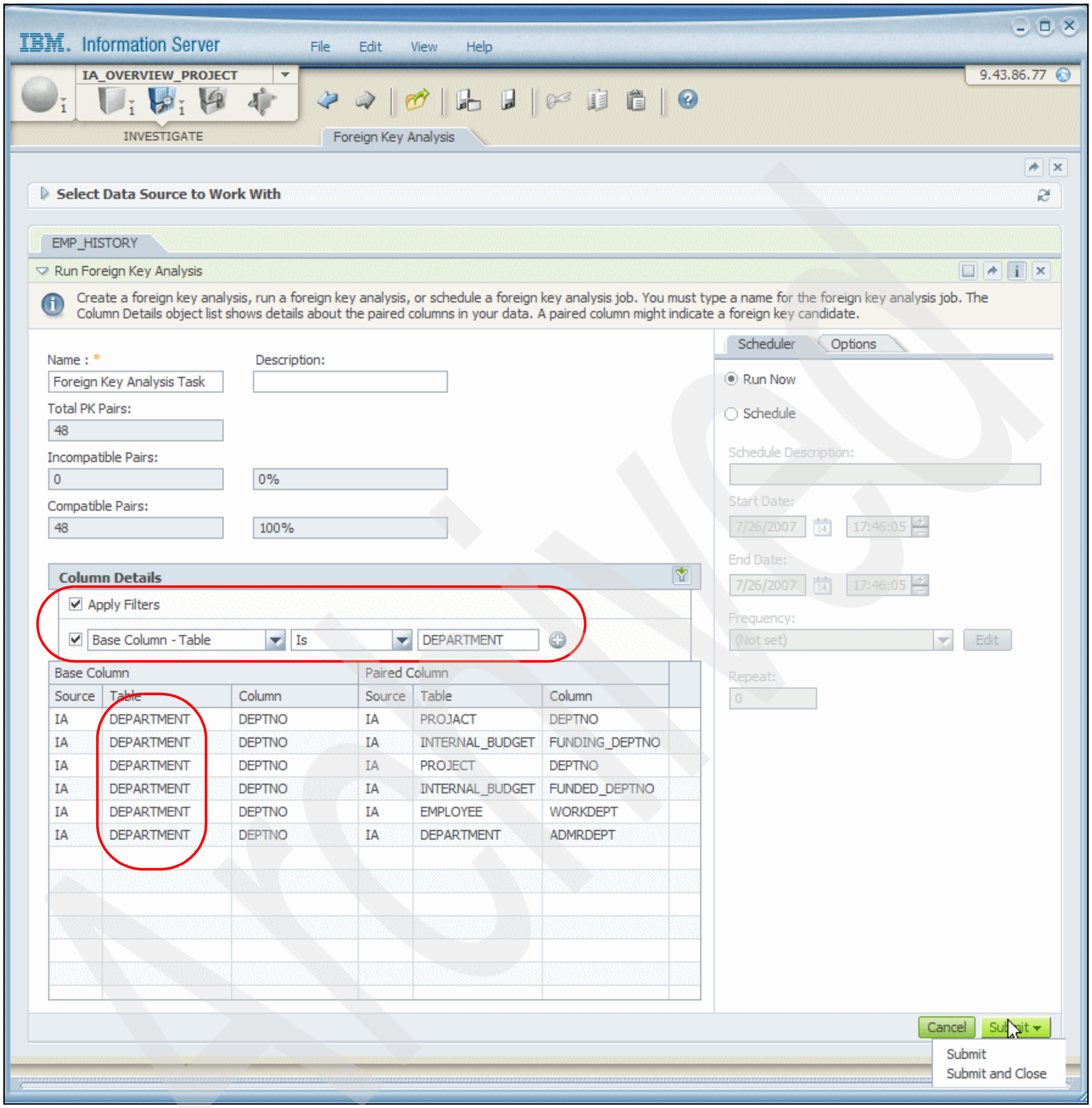
Column Details

□ Apply Filters

□ Base Column - Table Select...

Base Column				olumn	
Source	Table	Column		Table	Column
IA	DEPARTMENT	DEPTNO		PROJACT	DEPTNO
IA	DEPARTMENT	DEPTNO		INTERNAL_BUDGET	FUNDING_DEPTNO
IA	DEPARTMENT	DEPTNO		PROJECT	DEPTNO
IA	DEPARTMENT	DEPTNO		INTERNAL_BUDGET	FUNDED_DEPTNO
IA	DEPARTMENT	DEPTNO	IA	EMPLOYEE	WORKDEPT

Figure 1-226 Applying filters to foreign key analysis 4/5



Create a foreign key analysis, run a foreign key analysis, or schedule a foreign key analysis job. You must type a name for the foreign key analysis job. The Column Details object list shows details about the paired columns in your data. A paired column might indicate a foreign key candidate.

Name : * Foreign Key Analysis Task

Description:

Total PK Pairs: 48

Incompatible Pairs: 0 0%

Compatible Pairs: 48 100%

● Run Now

○ Schedule

Schedule Description:

Start Date: 7/26/2007 17:46:05

End Date: 7/26/2007 17:46:05

Frequency: (Not set) Edit

Repeat: 0

Column Details

☑ Apply Filters

☑ Base Column - Table Is DEPARTMENT

Base Column			Paired Column		
Source	Table	Column	Source	Table	Column
IA	DEPARTMENT	DEPTNO	IA	PROJACT	DEPTNO
IA	DEPARTMENT	DEPTNO	IA	INTERNAL_BUDGET	FUNDING_DEPTNO
IA	DEPARTMENT	DEPTNO	IA	PROJECT	DEPTNO
IA	DEPARTMENT	DEPTNO	IA	INTERNAL_BUDGET	FUNDED_DEPTNO
IA	DEPARTMENT	DEPTNO	IA	EMPLOYEE	WORKDEPT
IA	DEPARTMENT	DEPTNO	IA	DEPARTMENT	ADMRDEPT

Figure 1-227 Applying filters to foreign key analysis 5/5

1.10 Cross domain analysis

To determine whether columns contain overlapping or redundant data, you can run a cross-domain analysis job across columns in one or multiple tables or sources. Cross-domain analysis compares the data values between two columns to locate overlapping data.

Cross-domain analysis is a multiple step process as follows:

- After you select two or more columns and run a cross-domain analysis job, a list of all of the possible column pairs in your data is generated.
- The system then performs a compatibility test on each column pair to determine whether those columns are compatible with each as described in Table 1-9 on page 265. If the column pair is compatible, the columns are flagged and can be analyzed together. Column pairs that are not compatible are excluded from further analysis.

After the compatibility test, cross-domain analysis displays the results from the compatibility test for you to review and optionally mark a column redundant.

In the following sections, we describe briefly the main functions of cross domain analysis and the results that are produced by cross domain analysis. Using a sample set of tables, we describe the process of running cross domain analysis on a single table and a set of table and discuss the results of the cross domain analysis.

1.10.1 Cross domain analysis functions

Cross domain has the following characteristics:

- You must have Information Analyzer Data Operator privileges to perform cross domain analysis as shown in Table 1-4 on page 34.
- Can be performed on a single table or multiple tables.
- Allows you to filter out the tables and columns that should be excluded from pairing consideration.
- Can be scheduled to run at a set date and time.
- Reports can be produced of cross domain analysis results such as common domains, common domains - same name, domains compared - above threshold, and domains compared - redundant value detail.

> **Note:** For complete details, refer to *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.

As mentioned earlier, a cross domain analysis job identifies columns that have common domain values. In the first step, it generates a list of all possible column pairs and reports statistics that show the total number of pairs that were created, the total number of incompatible columns, and the total number of compatible columns. You review the compatibility test results before continuing with the analysis. If you want to modify the compatibility results, you can return to the column selection step and start again. You can also rerun a column analysis job on the data, review any inferences or selections that are made during analysis, and then adjust the inferences or selections before rerunning the cross-domain analysis job.

After you review the compatibility results and run the job, the system evaluates each of the column pairs to determine if the columns have common domains. The frequency distribution for each column in a pair is used to test the columns in two ways as follows:

- First, the values in the frequency distribution of one column are compared against the values in the frequency distribution of another column. The number of matching distinct values between the two columns are counted.
- After the values in a column pair are evaluated, a commonality percentage is calculated.

  A commonality percentage describes the proportion of matching values between two columns. For example, column 1 and column 2 are paired together.

  - If 7 out of 10 values in column 2 match column 1, the commonality percentage of column 2 to column 1 is 70%.
  - If 7 out of 7 values in column 1 match, the commonality percentage for column 1 to column 2 is 100%.

  Next, the commonality percentage is compared with the common domain threshold. You set the common domain threshold to define the criteria for determining when two columns share commonality. For example, If the common domain threshold is set to 95%, the columns must have a commonality percentage equal to 95% or higher for the system to infer that they share a common domain. You can tighten the criteria for commonality by raising the common domain threshold setting or loosen it by lowering the setting.

  In the commonality percentage example for column 1 and column 2, column 2 does not have a common domain with column 1 since 70% is less than 95%. However, column 1 does have a common domain with column 2 since 100% is greater than 95%.

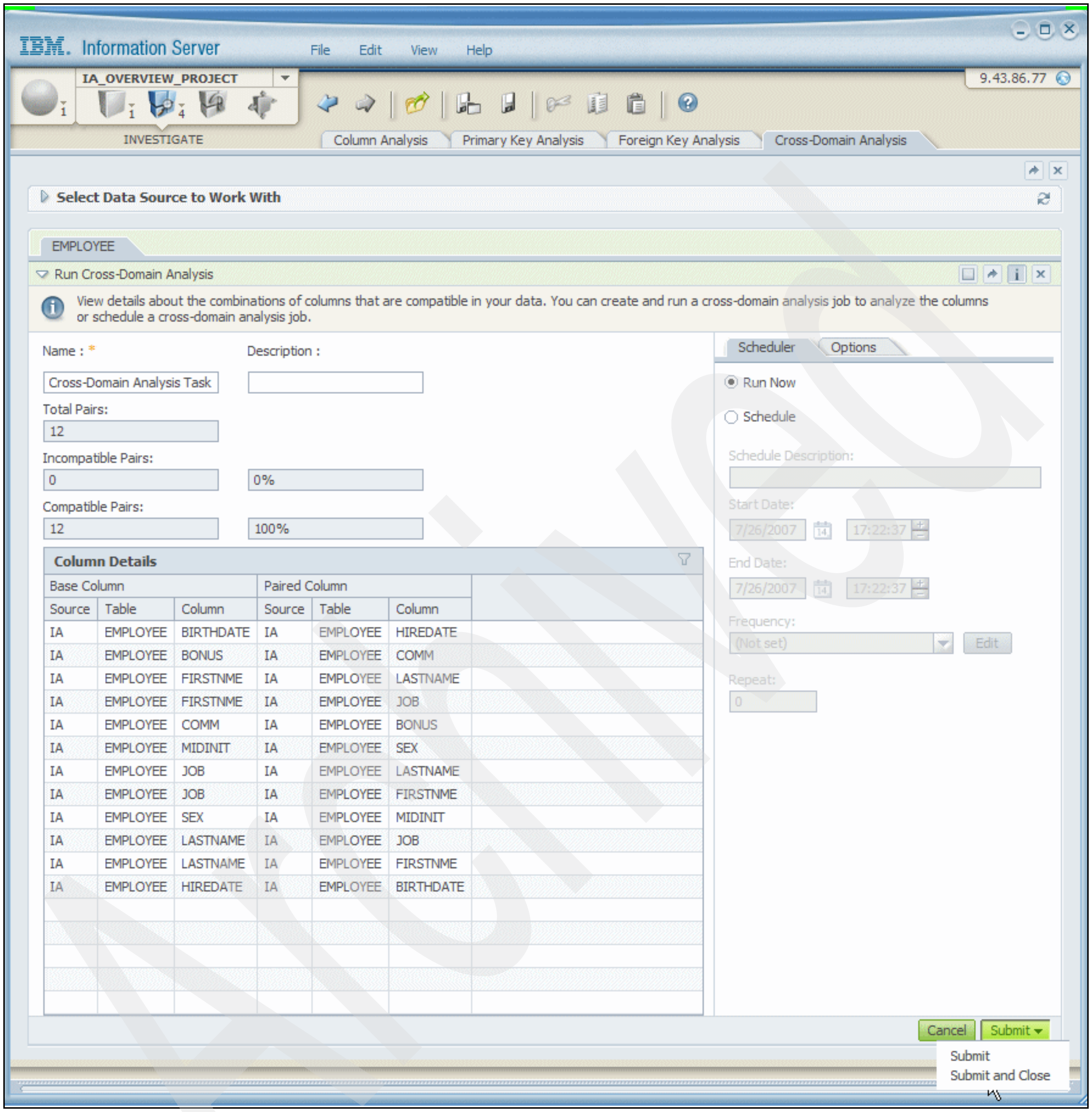
Column pairs that meet the common domain threshold criteria for having a common domain are flagged. Columns that share a common domain contain overlapping and potentially redundant data.

1.10.2 Cross domain analysis results

A cross domain analysis produces the following categories of output:

- Paired columns list

  This view provides details about the combinations of columns that are compatible in your data as shown in Figure 1-228 on page 304. You can create and run a cross domain analysis job to analyze the columns or schedule a cross domain analysis job.



View details about the combinations of columns that are compatible in your data. You can create and run a cross-domain analysis job to analyze the columns or schedule a cross-domain analysis job.

Name : * Cross-Domain Analysis Task

Description :

Total Pairs: 12

Incompatible Pairs: 0 0%

Compatible Pairs: 12 100%

● Run Now

○ Schedule

Schedule Description:

Start Date: 7/26/2007 17:22:37

End Date: 7/26/2007 17:22:37

Frequency: (Not set) Edit

Repeat: 0

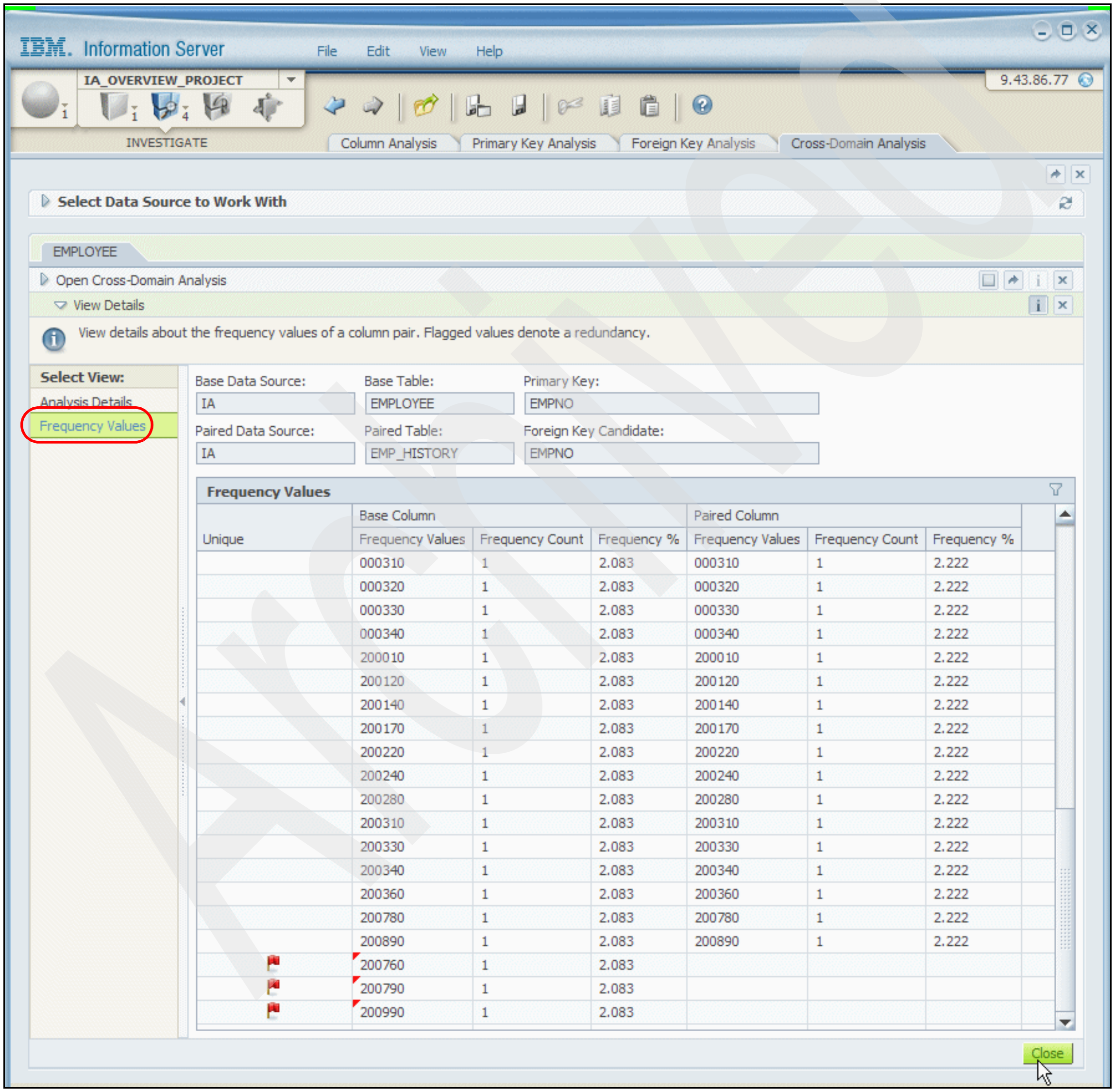
Column Details

Base Column			Paired Column		
Source	Table	Column	Source	Table	Column
IA	EMPLOYEE	BIRTHDATE	IA	EMPLOYEE	HIREDATE
IA	EMPLOYEE	BONUS	IA	EMPLOYEE	COMM
IA	EMPLOYEE	FIRSTNME	IA	EMPLOYEE	LASTNAME
IA	EMPLOYEE	FIRSTNME	IA	EMPLOYEE	JOB
IA	EMPLOYEE	COMM	IA	EMPLOYEE	BONUS
IA	EMPLOYEE	MIDINIT	IA	EMPLOYEE	SEX
IA	EMPLOYEE	JOB	IA	EMPLOYEE	LASTNAME
IA	EMPLOYEE	JOB	IA	EMPLOYEE	FIRSTNME
IA	EMPLOYEE	SEX	IA	EMPLOYEE	MIDINIT
IA	EMPLOYEE	LASTNAME	IA	EMPLOYEE	JOB
IA	EMPLOYEE	LASTNAME	IA	EMPLOYEE	FIRSTNME
IA	EMPLOYEE	HIREDATE	IA	EMPLOYEE	BIRTHDATE

Figure 1-228 Cross Domain Analysis paired columns on single table EMPLOYEE

- View Details - Frequency Values

  This view provides details of the frequency values of a column pair. In the View Details pane under the EMPLOYEE tab, select **Frequency Values** in the navigation pane to view the frequency values of the base column EMPNO in the EMPLOYEE and the corresponding paired column EMPNO column in the EMP_HISTORY table as shown in Figure 1-229 on page 305. This is a partial list of all the values. A red flag next to a value denotes a redundancy.



View details about the frequency values of a column pair. Flagged values denote a redundancy.

Base Data Source: IA
Base Table: EMPLOYEE
Primary Key: EMPNO
Paired Data Source: IA
Paired Table: EMP_HISTORY
Foreign Key Candidate: EMPNO

Frequency Values

	Base Column			Paired Column		
Unique	Frequency Values	Frequency Count	Frequency %	Frequency Values	Frequency Count	Frequency %
	000310	1	2.083	000310	1	2.222
	000320	1	2.083	000320	1	2.222
	000330	1	2.083	000330	1	2.222
	000340	1	2.083	000340	1	2.222
	200010	1	2.083	200010	1	2.222
	200120	1	2.083	200120	1	2.222
	200140	1	2.083	200140	1	2.222
	200170	1	2.083	200170	1	2.222
	200220	1	2.083	200220	1	2.222
	200240	1	2.083	200240	1	2.222
	200280	1	2.083	200280	1	2.222
	200310	1	2.083	200310	1	2.222
	200330	1	2.083	200330	1	2.222
	200340	1	2.083	200340	1	2.222
	200360	1	2.083	200360	1	2.222
	200780	1	2.083	200780	1	2.222
	200890	1	2.083	200890	1	2.222
🚩	200760	1	2.083			
🚩	200790	1	2.083			
🚩	200990	1	2.083			

Figure 1-229 Cross Domain View Details - Analysis Details on a single table EMPLOYEE

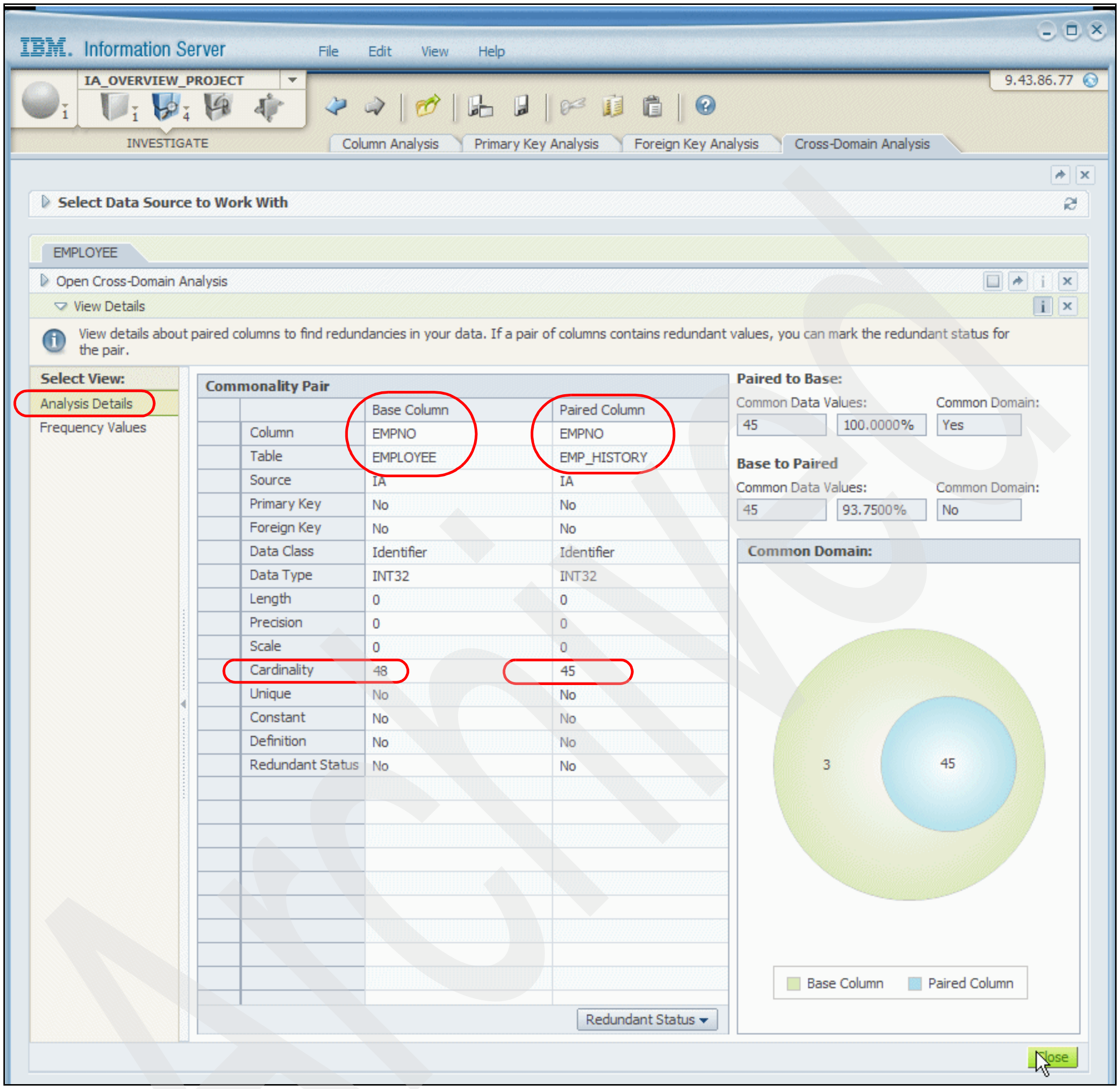
- View Details - Analysis Details

  This view provides details about paired columns to find redundancies in your data. If a pair of columns contains redundant columns, you can mark the redundant status for the pair.

  In the View Details pane under the EMPLOYEE tab, select **Analysis Details** in the navigation pane to view a Venn diagram of the base column EMPNO in the EMPLOYEE table and paired column EMPNO in the EMP_HISTORY table. The Venn diagram shows the common domains between these paired columns as follows:

  – There are 45 common data values between the paired to base column and it has a 100% commonality. All the 45 values in the paired EMPNO column in the EMP_HISTORY table are found in the base EMPNO column of the EMPLOYEE table. This is 100% of the values, and because it exceeds the common domain threshold (default is 98%), it is identified as sharing a common domain (Common Domain field has a Yes).

  – There are 45 common values out of a total of 48 values between the base (EMPNO in EMPLOYEE table) column to paired (EMPNO in EMP_HISTORY) column. This corresponds to 93.75% commonality. This 93.75% does not exceed the common domain threshold (default is 98%), and is therefore not identified as sharing a common domain (Common Domain field has a No).



View details about paired columns to find redundancies in your data. If a pair of columns contains redundant values, you can mark the redundant status for the pair.

Commonality Pair

	Base Column	Paired Column
Column	EMPNO	EMPNO
Table	EMPLOYEE	EMP_HISTORY
Source	IA	IA
Primary Key	No	No
Foreign Key	No	No
Data Class	Identifier	Identifier
Data Type	INT32	INT32
Length	0	0
Precision	0	0
Scale	0	0
Cardinality	48	45
Unique	No	No
Constant	No	No
Definition	No	No
Redundant Status	No	No

Paired to Base:

Common Data Values: 45 100.0000%

Common Domain: Yes

Base to Paired

Common Data Values: 45 93.7500%

Common Domain: No

Common Domain:

Figure 1-230 Cross Domain View Details - Analysis Details on a single table EMPLOYEE

1.10.3 Cross domain analysis usage scenario

The modified sample database described in Figure 1-11 on page 31 is used in the foreign key analysis examples described here.

In this section, we describe how to run a cross domain analysis job on a single table and on multiple tables using filters, and a review of the cross domain analysis results.

> **Attention:** In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

Run cross domain analysis on a single table

After opening a project of interest (IA_OVERVIEW_PROJECT in our case), run a cross domain analysis job on the EMPLOYEE table as follows to locate overlapping data across domains:

1. On the Investigate navigator menu in the console, select Cross-Domain Analysis as shown in Figure 1-88 on page 138.
2. In the Cross-Domain Analysis workspace, select all the columns of the EMPLOYEE table that you want to analyze, and click **Run Cross-Domain Analysis** in the Tasks pane as shown in Figure 1-231 on page 310.
3. In the Run Cross-Domain Analysis pane shown in Figure 1-232 on page 311, type a name (Cross-Domain Analysis Task) and optionally a description for the analysis job. A list of all possible column pairs generated is shown in the Column Details pane. The total number of pairs that were created (12), the total number of incompatible columns (0), and the total number of compatible columns (12). Select **Run Now** (or schedule the job to run at a later date) and click **Submit and Close**.
4. Figure 1-233 on page 312 shows the completed job. Click **View Results** to view the results of this job.
5. You can now view the results of the cross domain analysis for a column. Select EMPNO in the Cross Domain Analysis workspace, and click **Open Cross-Domain Analysis** as shown in Figure 1-234 on page 313.
6. Figure 1-235 on page 313 through Figure 1-237 on page 315 show summary information (scrolling right to view all column details) about the base and paired columns including the inferred data class, defined primary and foreign key, and the number of values and percentage of values shared from base to

paired column. For example, for the EMPNO base column in the EMPLOYEE table that is paired with column EMPNO in the EMP_HISTORY table, under the Base to Paired column, it shows 45 values (out of 48) being shared, which corresponds to 93.75%. To view further details of the pairing, select this pairing, and click **View Details** as shown in Figure 1-237 on page 315.

7. Figure 1-238 on page 316 provides details of the frequency values of the column pair in the View Details pane under the EMPLOYEE tab and Frequency Values in the navigation pane. This is a partial list of all the values in the base column and paired column. A red flag next to a value denotes a redundancy. Click **Close**.

8. Figure 1-239 on page 317 provides details about paired columns to find redundancies in your data in the View Details pane under the EMPLOYEE tab and Analysis Details in the navigation pane. The Venn diagram shows the common domains between these paired columns as follows:

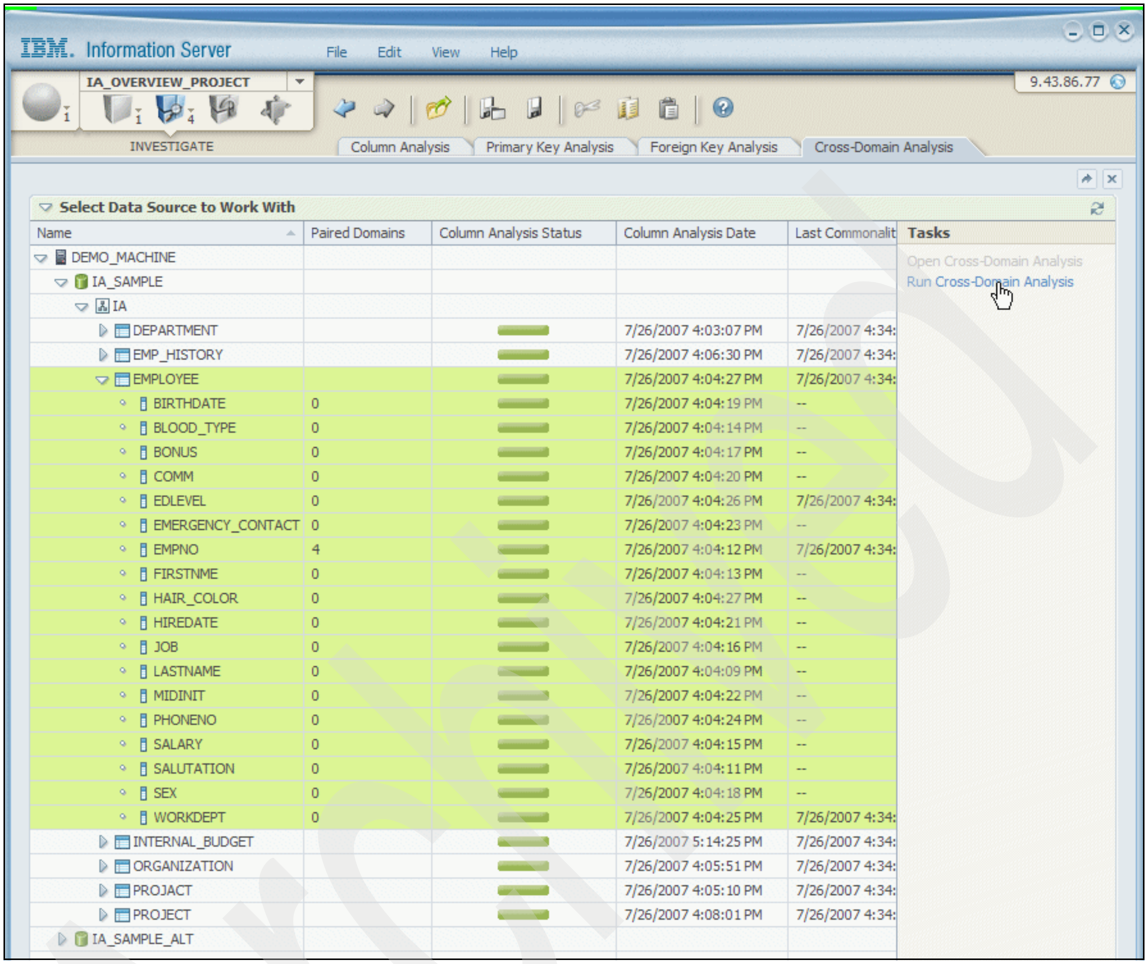
   – There are 45 common data values between the paired to base column and it has a 100% commonality. All the 45 values in the paired EMPNO column in the EMP_HISTORY table are found in the base EMPNO column of the EMPLOYEE table. This is 100% of the values, and because it exceeds the common domain threshold (default is 98%), it is identified as sharing a common domain (Common Domain field has a Yes).

   – There are 45 common values out of a total of 48 values between the base (EMPNO in EMPLOYEE table) column to paired (EMPNO in EMP_HISTORY) column. This corresponds to 93.75% commonality. This 93.75% does not exceed the common domain threshold (default is 98%), and is therefore not identified as sharing a common domain (Common Domain field has a No).

   Click **Close**.

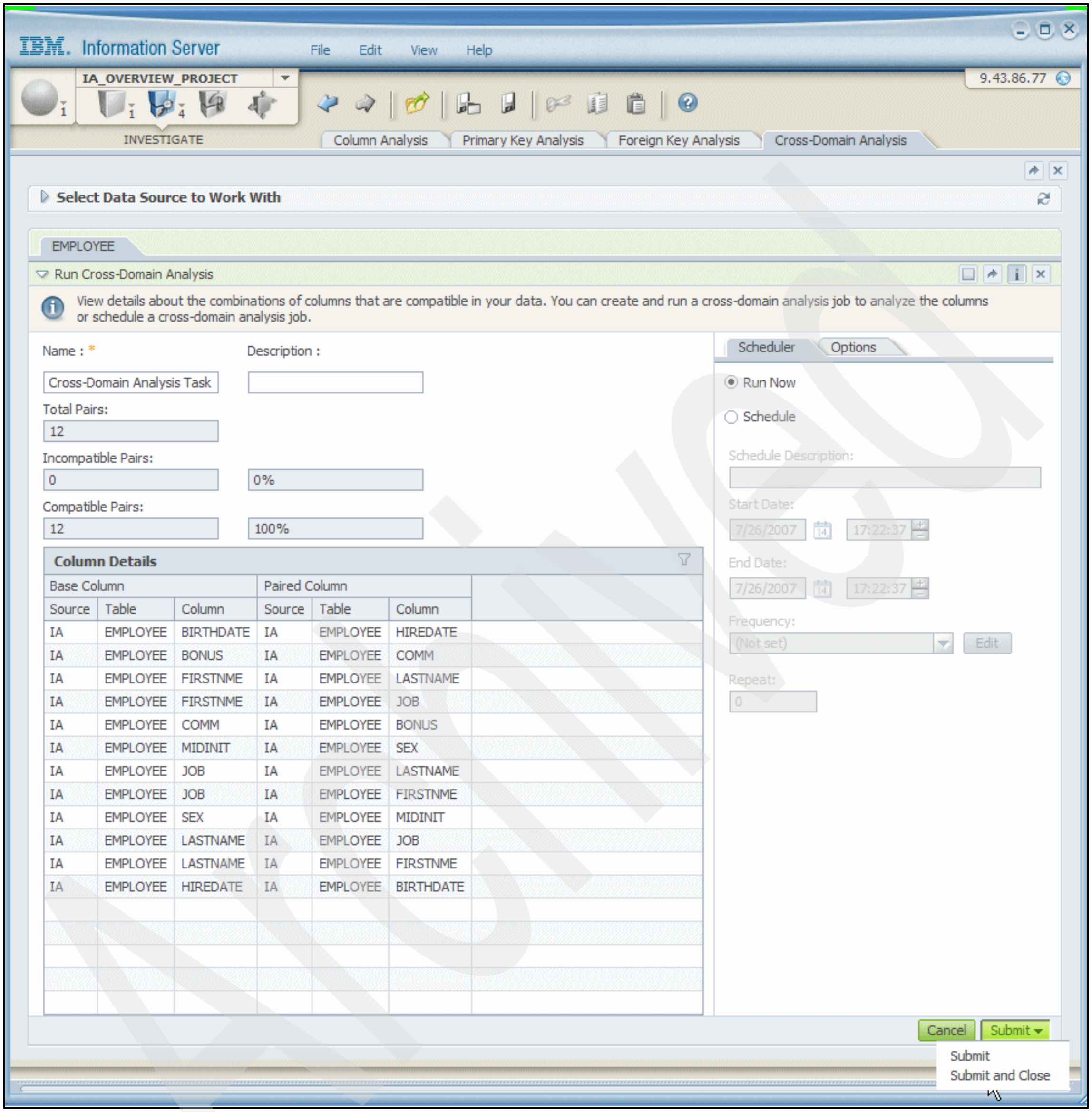
9. If you feel that this column pair contains redundant data, then you can mark it as such by clicking **Redundant Status Mark** → **Redundant Status** as shown in Figure 1-240 on page 318.

10. The highlighted icon under the paired column in Figure 1-241 on page 319 indicates that the pair qualifies as sharing a common domain. Click **Close**.



Name	Paired Domains	Column Analysis Status	Column Analysis Date	Last Commonalit
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT		▬	7/26/2007 4:03:07 PM	7/26/2007 4:34:
EMP_HISTORY		▬	7/26/2007 4:06:30 PM	7/26/2007 4:34:
EMPLOYEE		▬	7/26/2007 4:04:27 PM	7/26/2007 4:34:
BIRTHDATE	0	▬	7/26/2007 4:04:19 PM	--
BLOOD_TYPE	0	▬	7/26/2007 4:04:14 PM	--
BONUS	0	▬	7/26/2007 4:04:17 PM	--
COMM	0	▬	7/26/2007 4:04:20 PM	--
EDLEVEL	0	▬	7/26/2007 4:04:26 PM	7/26/2007 4:34:
EMERGENCY_CONTACT	0	▬	7/26/2007 4:04:23 PM	--
EMPNO	4	▬	7/26/2007 4:04:12 PM	7/26/2007 4:34:
FIRSTNME	0	▬	7/26/2007 4:04:13 PM	--
HAIR_COLOR	0	▬	7/26/2007 4:04:27 PM	--
HIREDATE	0	▬	7/26/2007 4:04:21 PM	--
JOB	0	▬	7/26/2007 4:04:16 PM	--
LASTNAME	0	▬	7/26/2007 4:04:09 PM	--
MIDINIT	0	▬	7/26/2007 4:04:22 PM	--
PHONENO	0	▬	7/26/2007 4:04:24 PM	--
SALARY	0	▬	7/26/2007 4:04:15 PM	--
SALUTATION	0	▬	7/26/2007 4:04:11 PM	--
SEX	0	▬	7/26/2007 4:04:18 PM	--
WORKDEPT	0	▬	7/26/2007 4:04:25 PM	7/26/2007 4:34:
INTERNAL_BUDGET		▬	7/26/2007 5:14:25 PM	7/26/2007 4:34:
ORGANIZATION		▬	7/26/2007 4:05:51 PM	7/26/2007 4:34:
PROJACT		▬	7/26/2007 4:05:10 PM	7/26/2007 4:34:
PROJECT		▬	7/26/2007 4:08:01 PM	7/26/2007 4:34:
IA_SAMPLE_ALT				

Figure 1-231 Cross Domain Analysis on a single table EMPLOYEE 1/11



View details about the combinations of columns that are compatible in your data. You can create and run a cross-domain analysis job to analyze the columns or schedule a cross-domain analysis job.

Name : * Cross-Domain Analysis Task

Description :

Total Pairs: 12

Incompatible Pairs: 0 0%

Compatible Pairs: 12 100%

Column Details

Base Column			Paired Column		
Source	Table	Column	Source	Table	Column
IA	EMPLOYEE	BIRTHDATE	IA	EMPLOYEE	HIREDATE
IA	EMPLOYEE	BONUS	IA	EMPLOYEE	COMM
IA	EMPLOYEE	FIRSTNME	IA	EMPLOYEE	LASTNAME
IA	EMPLOYEE	FIRSTNME	IA	EMPLOYEE	JOB
IA	EMPLOYEE	COMM	IA	EMPLOYEE	BONUS
IA	EMPLOYEE	MIDINIT	IA	EMPLOYEE	SEX
IA	EMPLOYEE	JOB	IA	EMPLOYEE	LASTNAME
IA	EMPLOYEE	JOB	IA	EMPLOYEE	FIRSTNME
IA	EMPLOYEE	SEX	IA	EMPLOYEE	MIDINIT
IA	EMPLOYEE	LASTNAME	IA	EMPLOYEE	JOB
IA	EMPLOYEE	LASTNAME	IA	EMPLOYEE	FIRSTNME
IA	EMPLOYEE	HIREDATE	IA	EMPLOYEE	BIRTHDATE

● Run Now

○ Schedule

Schedule Description:

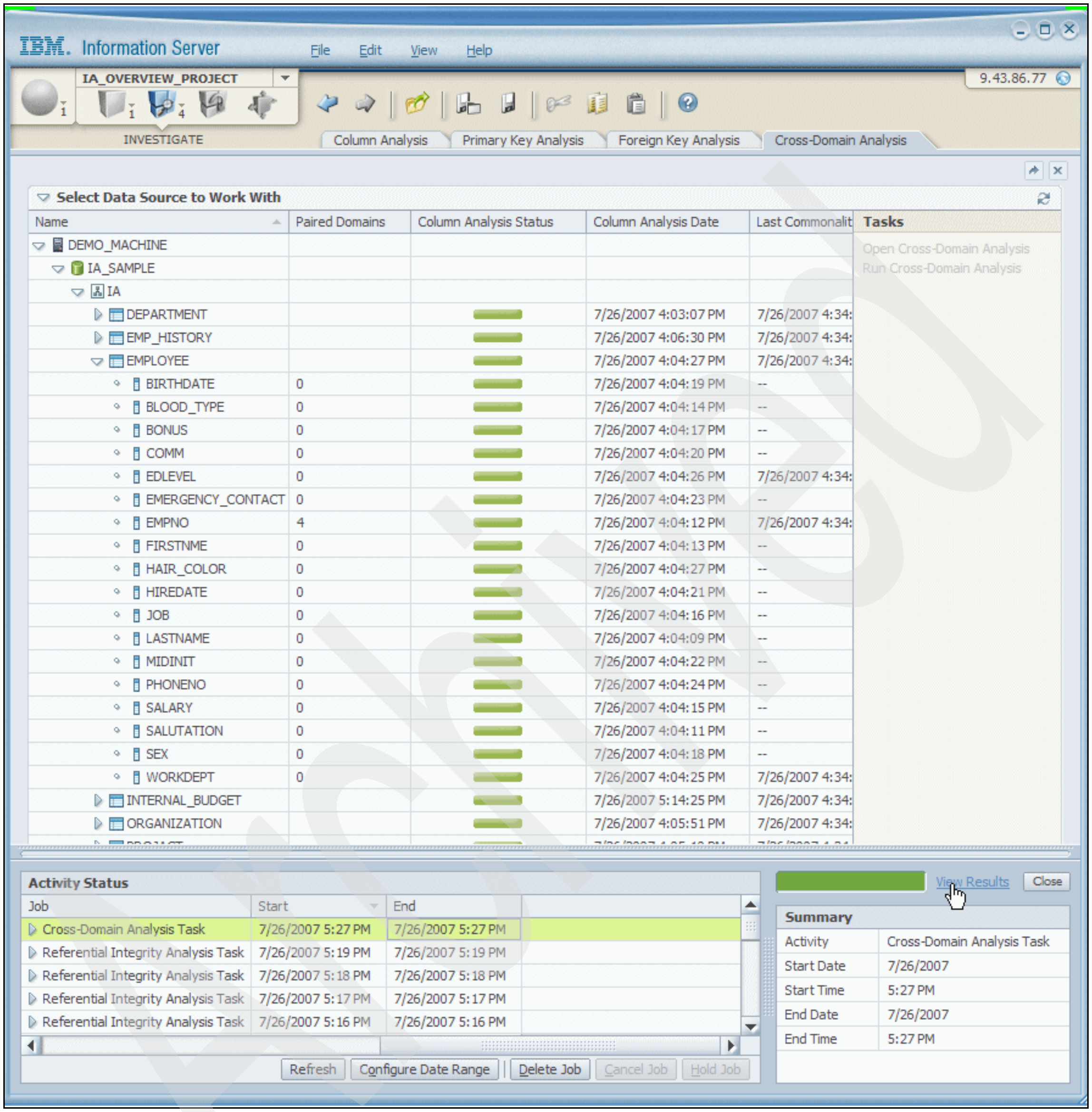
Start Date: 7/26/2007 17:22:37

End Date: 7/26/2007 17:22:37

Frequency: (Not set)

Repeat: 0

Figure 1-232 Cross Domain Analysis on a single table EMPLOYEE 2/11



Select Data Source to Work With

Name	Paired Domains	Column Analysis Status	Column Analysis Date	Last Commonalit
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT		🟩	7/26/2007 4:03:07 PM	7/26/2007 4:34:
EMP_HISTORY		🟩	7/26/2007 4:06:30 PM	7/26/2007 4:34:
EMPLOYEE		🟩	7/26/2007 4:04:27 PM	7/26/2007 4:34:
BIRTHDATE	0	🟩	7/26/2007 4:04:19 PM	--
BLOOD_TYPE	0	🟩	7/26/2007 4:04:14 PM	--
BONUS	0	🟩	7/26/2007 4:04:17 PM	--
COMM	0	🟩	7/26/2007 4:04:20 PM	--
EDLEVEL	0	🟩	7/26/2007 4:04:26 PM	7/26/2007 4:34:
EMERGENCY_CONTACT	0	🟩	7/26/2007 4:04:23 PM	--
EMPNO	4	🟩	7/26/2007 4:04:12 PM	7/26/2007 4:34:
FIRSTNME	0	🟩	7/26/2007 4:04:13 PM	--
HAIR_COLOR	0	🟩	7/26/2007 4:04:27 PM	--
HIREDATE	0	🟩	7/26/2007 4:04:21 PM	--
JOB	0	🟩	7/26/2007 4:04:16 PM	--
LASTNAME	0	🟩	7/26/2007 4:04:09 PM	--
MIDINIT	0	🟩	7/26/2007 4:04:22 PM	--
PHONENO	0	🟩	7/26/2007 4:04:24 PM	--
SALARY	0	🟩	7/26/2007 4:04:15 PM	--
SALUTATION	0	🟩	7/26/2007 4:04:11 PM	--
SEX	0	🟩	7/26/2007 4:04:18 PM	--
WORKDEPT	0	🟩	7/26/2007 4:04:25 PM	7/26/2007 4:34:
INTERNAL_BUDGET		🟩	7/26/2007 5:14:25 PM	7/26/2007 4:34:
ORGANIZATION		🟩	7/26/2007 4:05:51 PM	7/26/2007 4:34:

Tasks

Open Cross-Domain Analysis
Run Cross-Domain Analysis

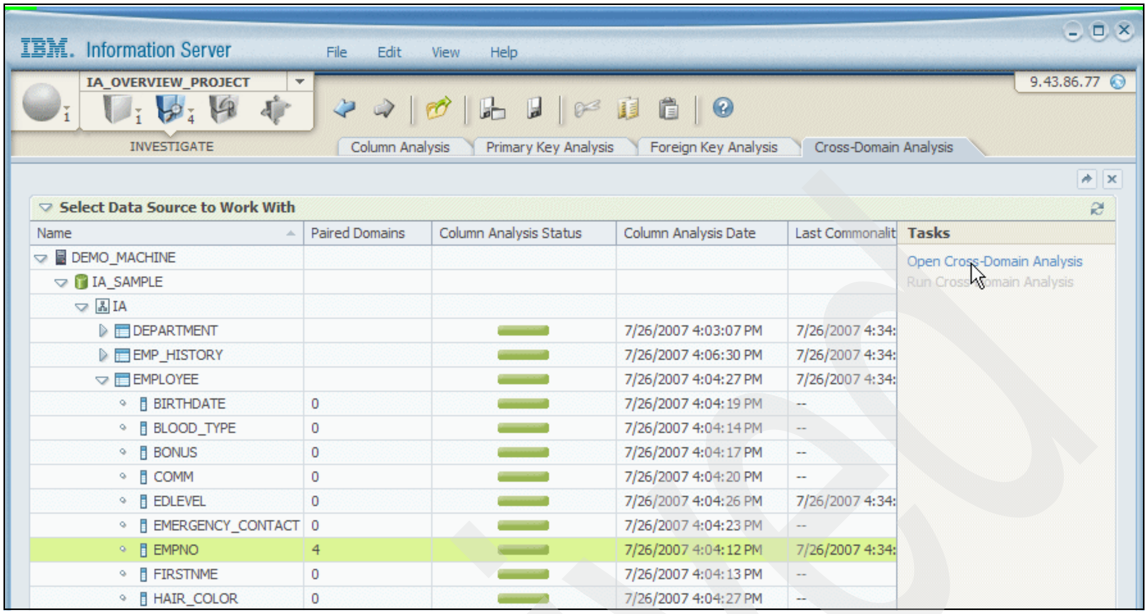
Activity Status

Job	Start	End
Cross-Domain Analysis Task	7/26/2007 5:27 PM	7/26/2007 5:27 PM
Referential Integrity Analysis Task	7/26/2007 5:19 PM	7/26/2007 5:19 PM
Referential Integrity Analysis Task	7/26/2007 5:18 PM	7/26/2007 5:18 PM
Referential Integrity Analysis Task	7/26/2007 5:17 PM	7/26/2007 5:17 PM
Referential Integrity Analysis Task	7/26/2007 5:16 PM	7/26/2007 5:16 PM

Summary

Activity	Cross-Domain Analysis Task
Start Date	7/26/2007
Start Time	5:27 PM
End Date	7/26/2007
End Time	5:27 PM

Figure 1-233 Cross Domain Analysis on a single table EMPLOYEE 3/11



Select Data Source to Work With

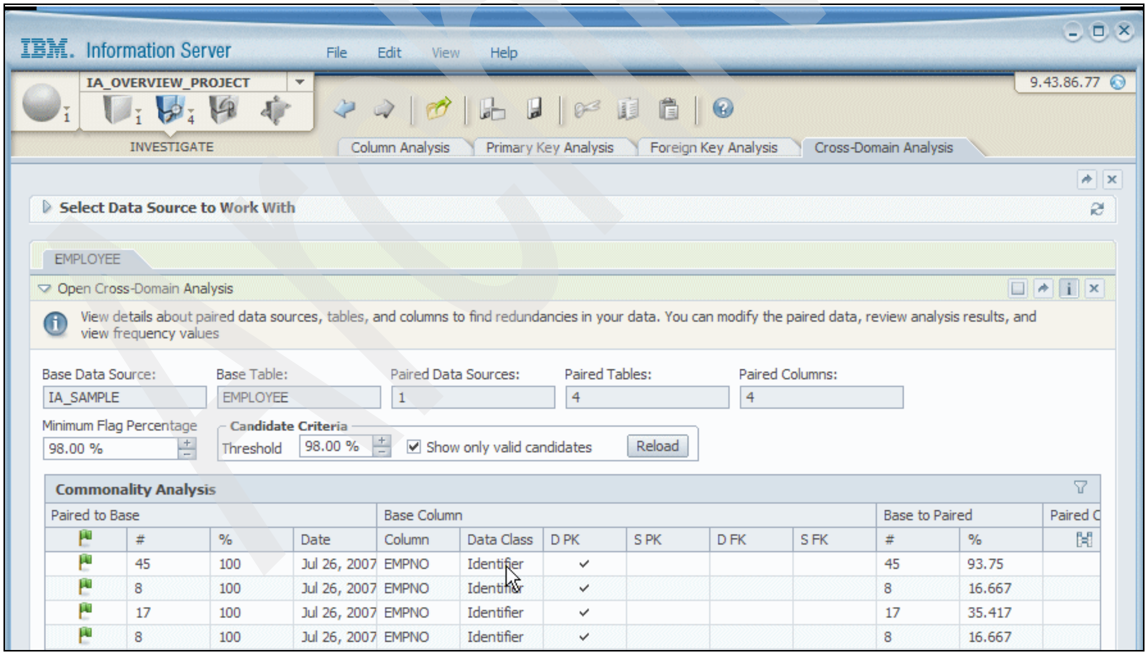
Name	Paired Domains	Column Analysis Status	Column Analysis Date	Last Commonalit
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT		🟩	7/26/2007 4:03:07 PM	7/26/2007 4:34:
EMP_HISTORY		🟩	7/26/2007 4:06:30 PM	7/26/2007 4:34:
EMPLOYEE		🟩	7/26/2007 4:04:27 PM	7/26/2007 4:34:
BIRTHDATE	0	🟩	7/26/2007 4:04:19 PM	--
BLOOD_TYPE	0	🟩	7/26/2007 4:04:14 PM	--
BONUS	0	🟩	7/26/2007 4:04:17 PM	--
COMM	0	🟩	7/26/2007 4:04:20 PM	--
EDLEVEL	0	🟩	7/26/2007 4:04:26 PM	7/26/2007 4:34:
EMERGENCY_CONTACT	0	🟩	7/26/2007 4:04:23 PM	--
EMPNO	4	🟩	7/26/2007 4:04:12 PM	7/26/2007 4:34:
FIRSTNME	0	🟩	7/26/2007 4:04:13 PM	--
HAIR_COLOR	0	🟩	7/26/2007 4:04:27 PM	--

Tasks

Open Cross-Domain Analysis

Run Cross-Domain Analysis

Figure 1-234 Cross Domain Analysis on a single table EMPLOYEE 4/11



Open Cross-Domain Analysis

View details about paired data sources, tables, and columns to find redundancies in your data. You can modify the paired data, review analysis results, and view frequency values

Base Data Source: IA_SAMPLE

Base Table: EMPLOYEE

Paired Data Sources: 1

Paired Tables: 4

Paired Columns: 4

Minimum Flag Percentage: 98.00 %

Candidate Criteria

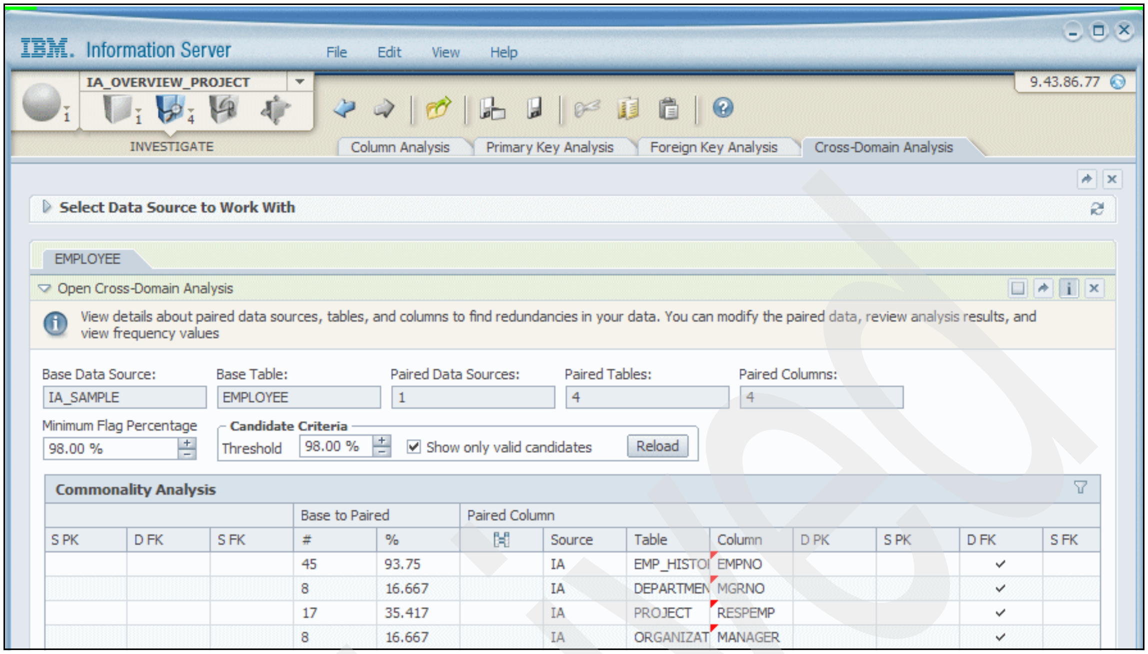
Threshold: 98.00 %

☑ Show only valid candidates

Commonality Analysis

Paired to Base			Base Column							Base to Paired		Paired C
	#	%	Date	Column	Data Class	D PK	S PK	D FK	S FK	#	%	
🚩	45	100	Jul 26, 2007	EMPNO	Identifier	✓				45	93.75	
🚩	8	100	Jul 26, 2007	EMPNO	Identifier	✓				8	16.667	
🚩	17	100	Jul 26, 2007	EMPNO	Identifier	✓				17	35.417	
🚩	8	100	Jul 26, 2007	EMPNO	Identifier	✓				8	16.667	

Figure 1-235 Cross Domain Analysis on a single table EMPLOYEE 5/11



View details about paired data sources, tables, and columns to find redundancies in your data. You can modify the paired data, review analysis results, and view frequency values

Base Data Source: IA_SAMPLE

Base Table: EMPLOYEE

Paired Data Sources: 1

Paired Tables: 4

Paired Columns: 4

Minimum Flag Percentage 98.00 %

Candidate Criteria

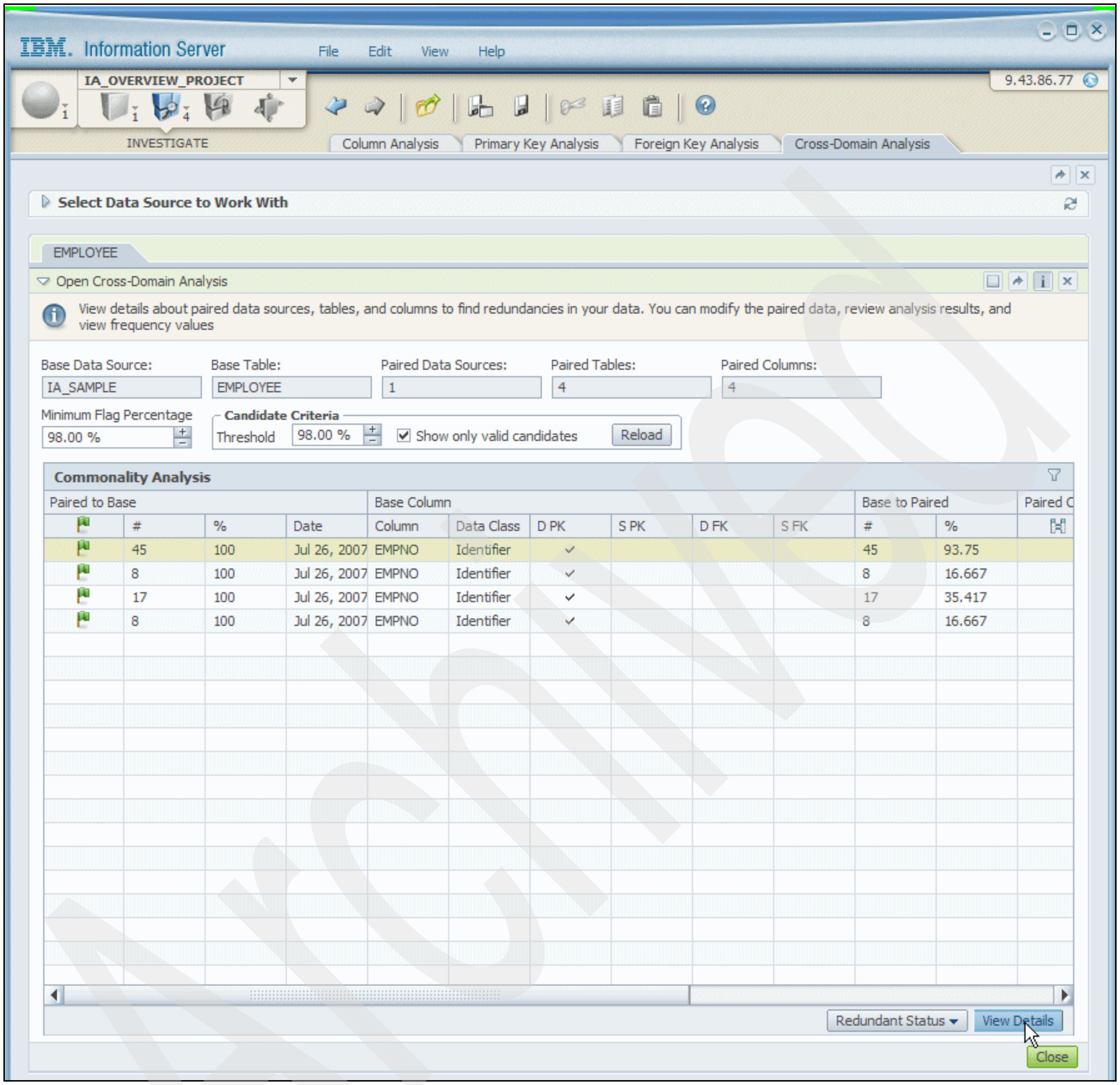
Threshold 98.00 %

☑ Show only valid candidates

Commonality Analysis

			Base to Paired		Paired Column							
S PK	D FK	S FK	#	%		Source	Table	Column	D PK	S PK	D FK	S FK
			45	93.75		IA	EMP_HISTOI	EMPNO			✓	
			8	16.667		IA	DEPARTMEN	MGRNO			✓	
			17	35.417		IA	PROJECT	RESPEMP			✓	
			8	16.667		IA	ORGANIZAT	MANAGER			✓	

Figure 1-236 Cross Domain Analysis on a single table EMPLOYEE 6/11



View details about paired data sources, tables, and columns to find redundancies in your data. You can modify the paired data, review analysis results, and view frequency values

Base Data Source: IA_SAMPLE
Base Table: EMPLOYEE
Paired Data Sources: 1
Paired Tables: 4
Paired Columns: 4

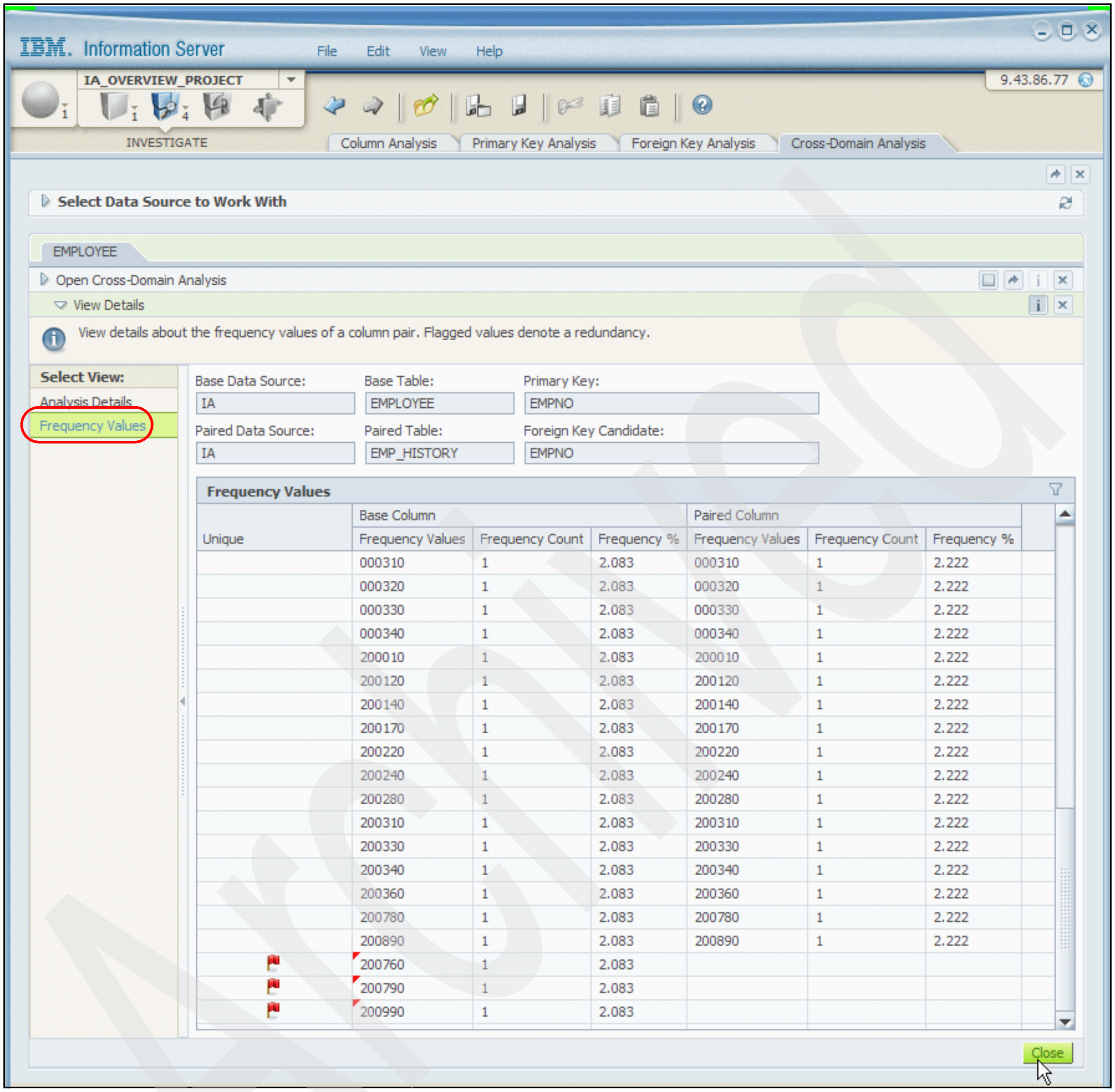
Minimum Flag Percentage: 98.00 %

Candidate Criteria
Threshold: 98.00 %
☑ Show only valid candidates

Commonality Analysis

Paired to Base				Base Column						Base to Paired		Paired C
	#	%	Date	Column	Data Class	D PK	S PK	D FK	S FK	#	%	
	45	100	Jul 26, 2007	EMPNO	Identifier	✓				45	93.75	
	8	100	Jul 26, 2007	EMPNO	Identifier	✓				8	16.667	
	17	100	Jul 26, 2007	EMPNO	Identifier	✓				17	35.417	
	8	100	Jul 26, 2007	EMPNO	Identifier	✓				8	16.667	

Figure 1-237 Cross Domain Analysis on a single table EMPLOYEE 7/11



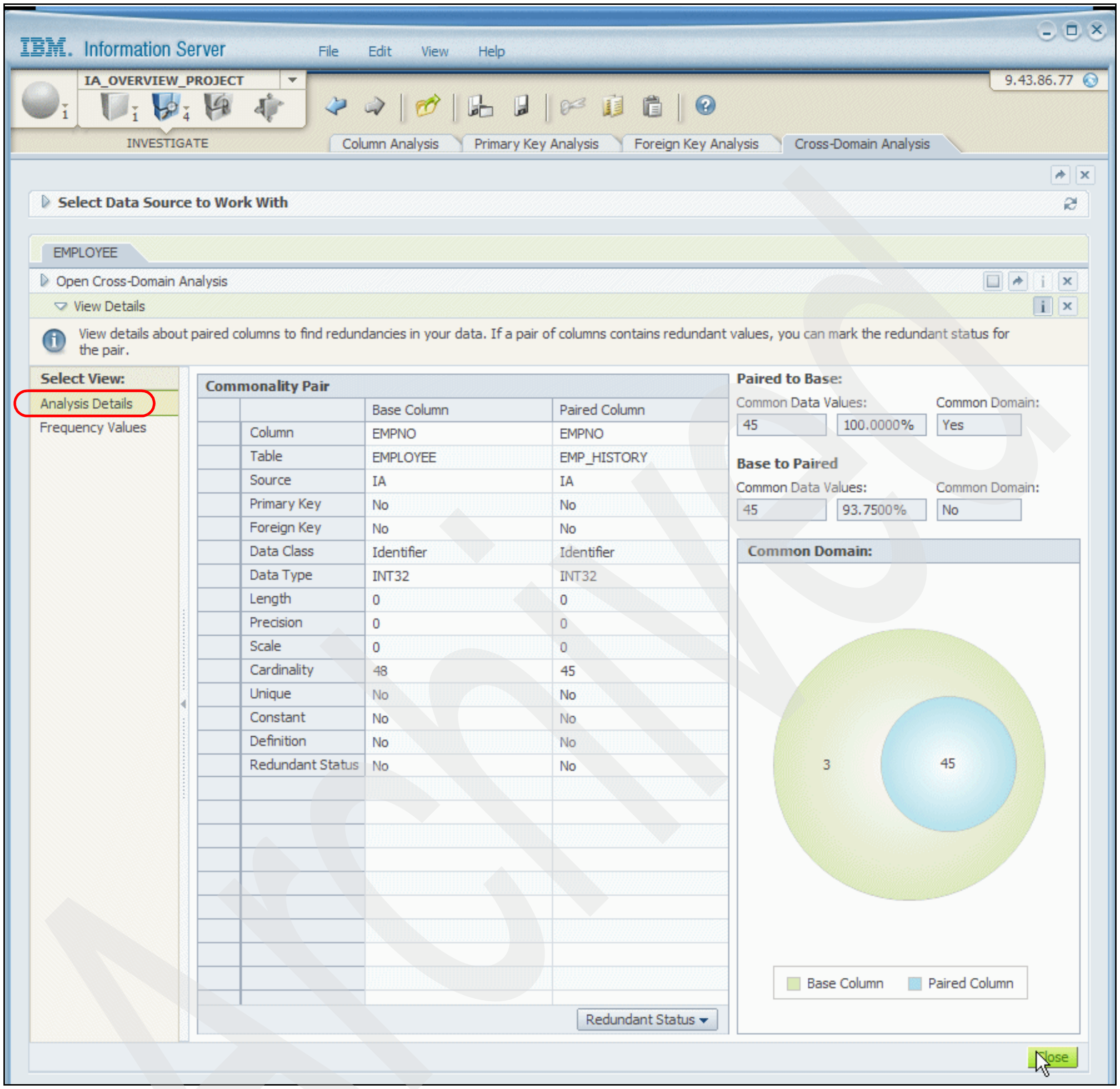
View details about the frequency values of a column pair. Flagged values denote a redundancy.

Base Data Source: IA
Base Table: EMPLOYEE
Primary Key: EMPNO
Paired Data Source: IA
Paired Table: EMP_HISTORY
Foreign Key Candidate: EMPNO

Frequency Values

	Base Column			Paired Column		
Unique	Frequency Values	Frequency Count	Frequency %	Frequency Values	Frequency Count	Frequency %
	000310	1	2.083	000310	1	2.222
	000320	1	2.083	000320	1	2.222
	000330	1	2.083	000330	1	2.222
	000340	1	2.083	000340	1	2.222
	200010	1	2.083	200010	1	2.222
	200120	1	2.083	200120	1	2.222
	200140	1	2.083	200140	1	2.222
	200170	1	2.083	200170	1	2.222
	200220	1	2.083	200220	1	2.222
	200240	1	2.083	200240	1	2.222
	200280	1	2.083	200280	1	2.222
	200310	1	2.083	200310	1	2.222
	200330	1	2.083	200330	1	2.222
	200340	1	2.083	200340	1	2.222
	200360	1	2.083	200360	1	2.222
	200780	1	2.083	200780	1	2.222
	200890	1	2.083	200890	1	2.222
🚩	200760	1	2.083			
🚩	200790	1	2.083			
🚩	200990	1	2.083			

Figure 1-238 Cross Domain Analysis on a single table EMPLOYEE 8/11



View details about paired columns to find redundancies in your data. If a pair of columns contains redundant values, you can mark the redundant status for the pair.

Commonality Pair

	Base Column	Paired Column
Column	EMPNO	EMPNO
Table	EMPLOYEE	EMP_HISTORY
Source	IA	IA
Primary Key	No	No
Foreign Key	No	No
Data Class	Identifier	Identifier
Data Type	INT32	INT32
Length	0	0
Precision	0	0
Scale	0	0
Cardinality	48	45
Unique	No	No
Constant	No	No
Definition	No	No
Redundant Status	No	No

Paired to Base:

Common Data Values: 45 100.0000%

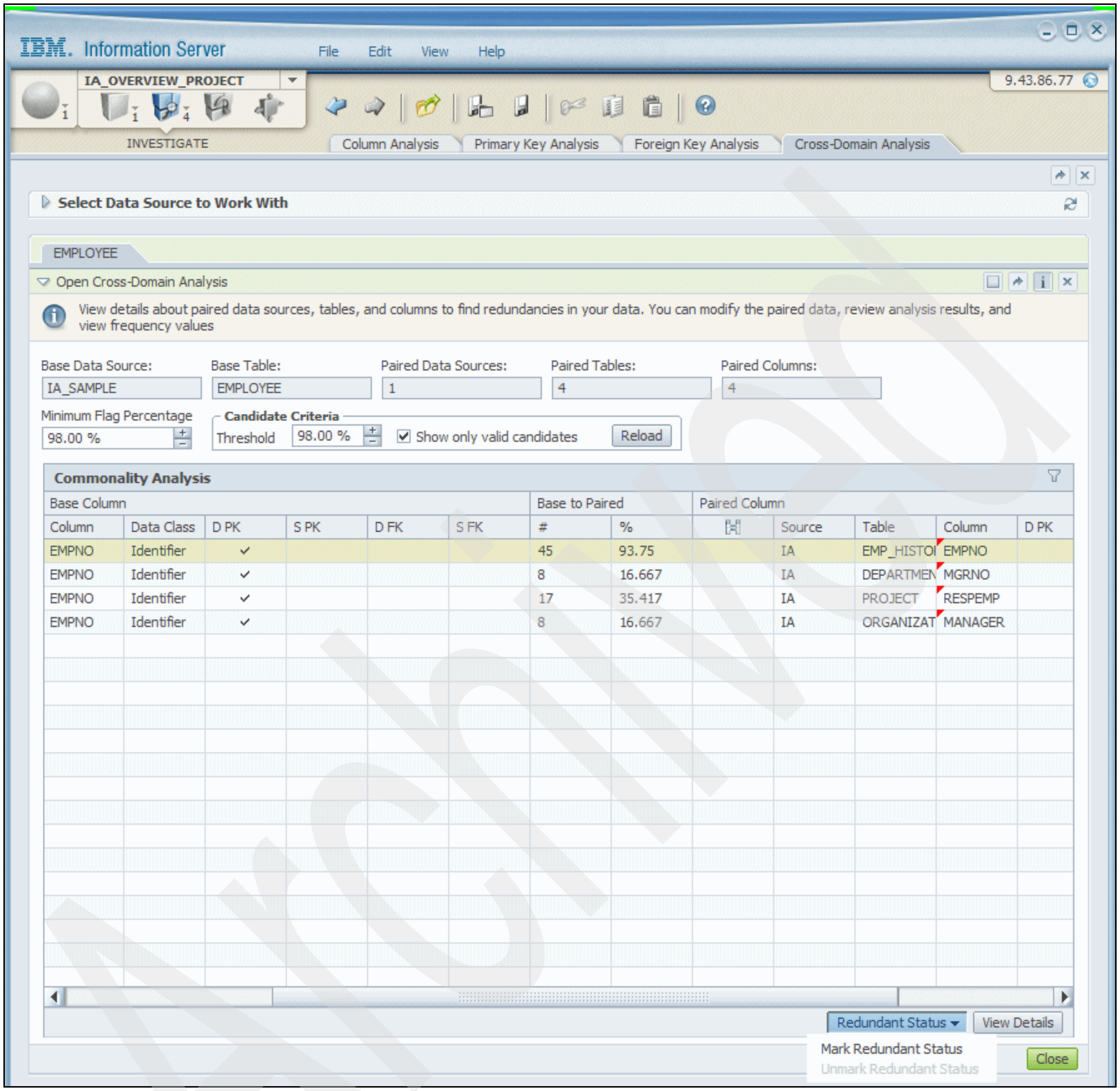
Common Domain: Yes

Base to Paired

Common Data Values: 45 93.7500%

Common Domain: No

Figure 1-239 Cross Domain Analysis on a single table EMPLOYEE 9/11



Select Data Source to Work With

EMPLOYEE

Open Cross-Domain Analysis

View details about paired data sources, tables, and columns to find redundancies in your data. You can modify the paired data, review analysis results, and view frequency values

Base Data Source:	Base Table:	Paired Data Sources:	Paired Tables:	Paired Columns:
IA_SAMPLE	EMPLOYEE	1	4	4

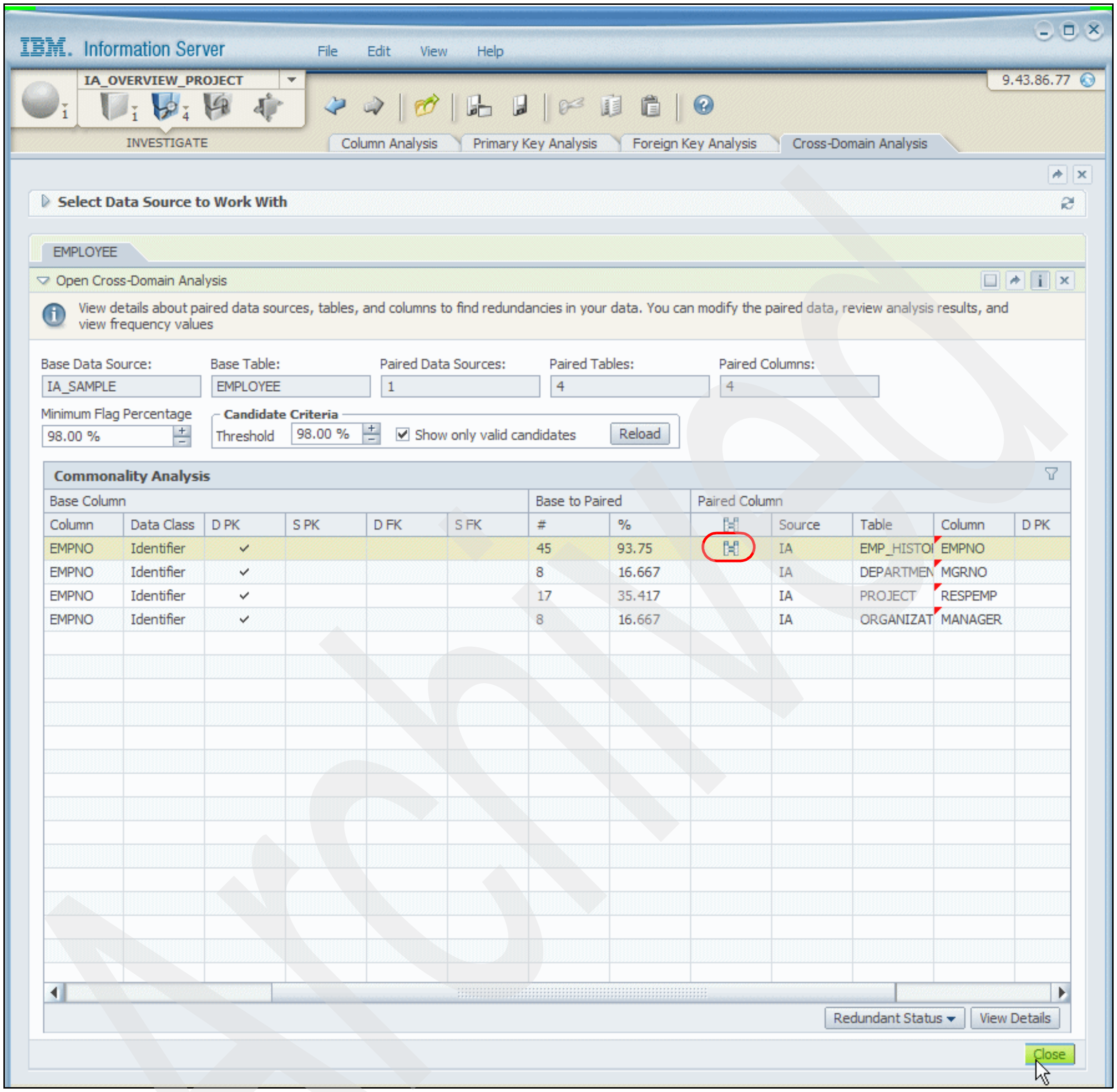
Minimum Flag Percentage: 98.00 %

Candidate Criteria
Threshold: 98.00 %
☑ Show only valid candidates

Commonality Analysis

Base Column						Base to Paired		Paired Column				
Column	Data Class	D PK	S PK	D FK	S FK	#	%		Source	Table	Column	D PK
EMPNO	Identifier	✓				45	93.75		IA	EMP_HISTOI	EMPNO	
EMPNO	Identifier	✓				8	16.667		IA	DEPARTMEN	MGRNO	
EMPNO	Identifier	✓				17	35.417		IA	PROJECT	RESPEMP	
EMPNO	Identifier	✓				8	16.667		IA	ORGANIZAT	MANAGER	

Figure 1-240 Cross Domain Analysis on a single table EMPLOYEE 10/11



View details about paired data sources, tables, and columns to find redundancies in your data. You can modify the paired data, review analysis results, and view frequency values

Base Data Source: IA_SAMPLE
Base Table: EMPLOYEE
Paired Data Sources: 1
Paired Tables: 4
Paired Columns: 4
Minimum Flag Percentage: 98.00 %
Candidate Criteria
Threshold: 98.00 %
☑ Show only valid candidates
Reload

Commonality Analysis

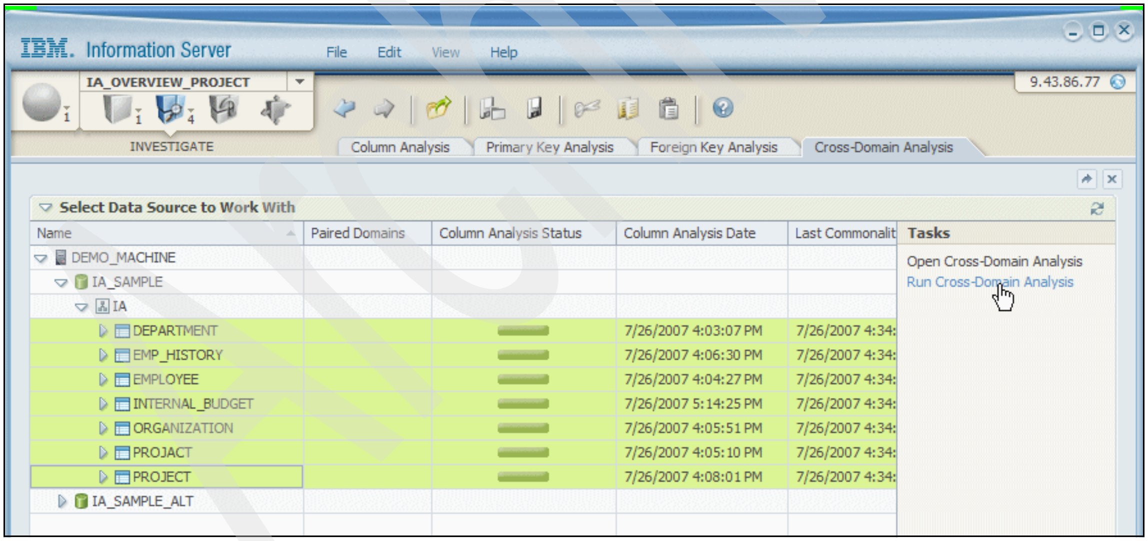
Base Column						Base to Paired		Paired Column				
Column	Data Class	D PK	S PK	D FK	S FK	#	%		Source	Table	Column	D PK
EMPNO	Identifier	✓				45	93.75		IA	EMP_HISTOI	EMPNO	
EMPNO	Identifier	✓				8	16.667		IA	DEPARTMEN	MGRNO	
EMPNO	Identifier	✓				17	35.417		IA	PROJECT	RESPEMP	
EMPNO	Identifier	✓				8	16.667		IA	ORGANIZAT	MANAGER	

Figure 1-241 Cross Domain Analysis on a single table EMPLOYEE 11/11

Run cross domain analysis on a set of tables

Figure 1-242 on page 320 through Figure 1-244 on page 322 show the selection of a set of tables for running cross domain analysis, and the filtering of columns to reduce the number of column pairs to analyze as follows:

1. In the Cross-Domain Analysis workspace, select all the tables in the IA data source and click **Run Cross-Domain Analysis** as shown in Figure 1-242 on page 320.
2. In the Run Cross-Domain Analysis pane in Figure 1-243 on page 321, type a name (Cross-Domain Analysis Task) and optionally a description for the analysis job. A list of all possible column pairs generated is shown in the Column Details pane—the total number of pairs that were created (170), the total number of incompatible columns (0), and the total number of compatible columns (170). This is a very large number of column pairs to analyze. You can choose to filter some of these columns out by clicking the filter icon as shown.
3. From the drop-down lists in Column Details in Figure 1-244 on page 322, choose the criteria for filtering. In this case, it is the selection of only tables in the base column that have the value DEPARTMENT. If the resulting column pairs are satisfactory, click **Submit and Close** to initiate cross domain analysis.



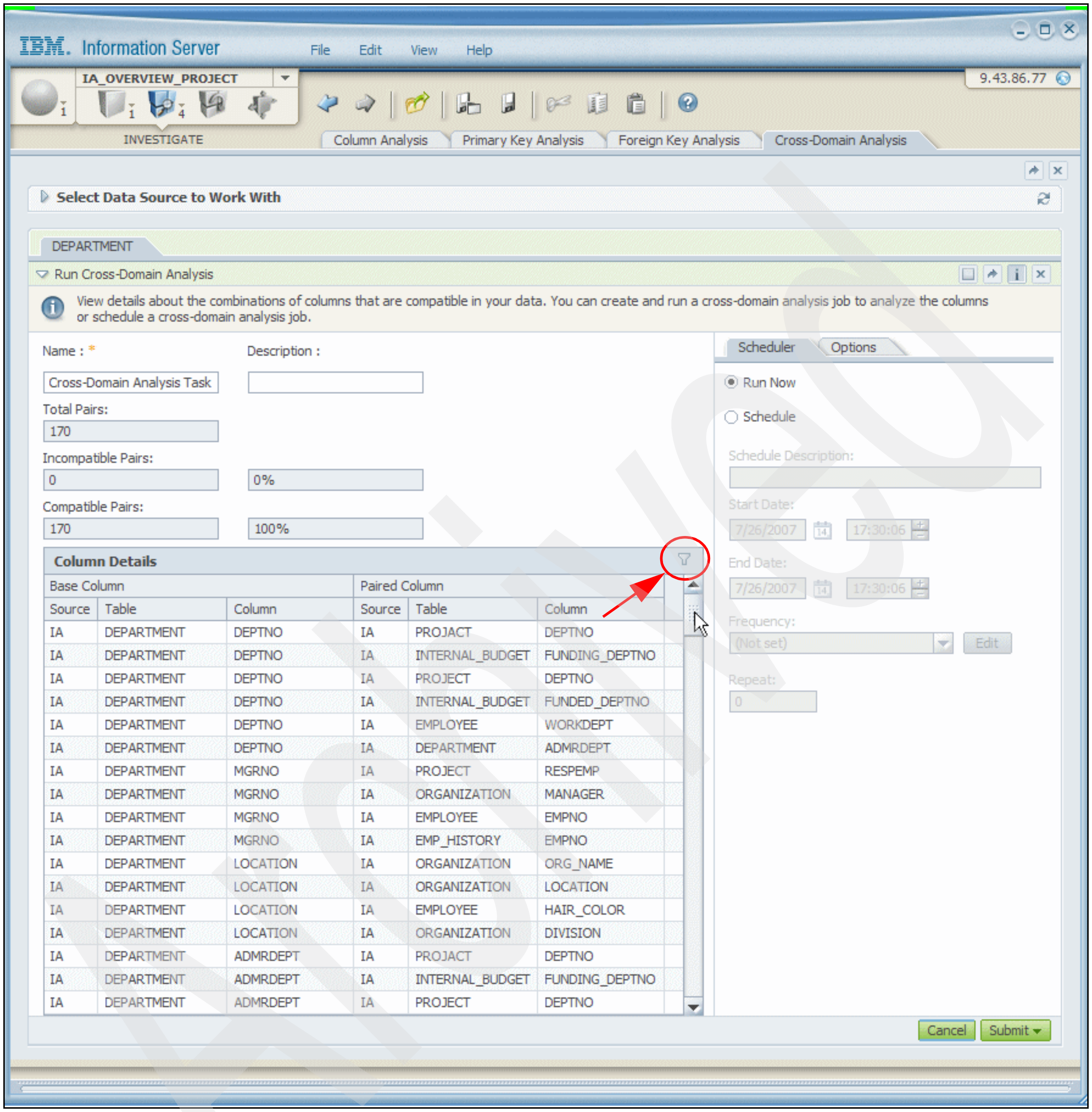
Name	Paired Domains	Column Analysis Status	Column Analysis Date	Last Commonalit
DEMO_MACHINE				
IA_SAMPLE				
IA				
DEPARTMENT		▬	7/26/2007 4:03:07 PM	7/26/2007 4:34:
EMP_HISTORY		▬	7/26/2007 4:06:30 PM	7/26/2007 4:34:
EMPLOYEE		▬	7/26/2007 4:04:27 PM	7/26/2007 4:34:
INTERNAL_BUDGET		▬	7/26/2007 5:14:25 PM	7/26/2007 4:34:
ORGANIZATION		▬	7/26/2007 4:05:51 PM	7/26/2007 4:34:
PROJACT		▬	7/26/2007 4:05:10 PM	7/26/2007 4:34:
PROJECT		▬	7/26/2007 4:08:01 PM	7/26/2007 4:34:
IA_SAMPLE_ALT				

Tasks

Open Cross-Domain Analysis

Run Cross-Domain Analysis

Figure 1-242 Cross Domain Analysis on a set of tables 1/3



View details about the combinations of columns that are compatible in your data. You can create and run a cross-domain analysis job to analyze the columns or schedule a cross-domain analysis job.

Name : * Cross-Domain Analysis Task

Description :

Total Pairs: 170

Incompatible Pairs: 0 0%

Compatible Pairs: 170 100%

● Run Now

○ Schedule

Schedule Description:

Start Date: 7/26/2007 17:30:06

End Date: 7/26/2007 17:30:06

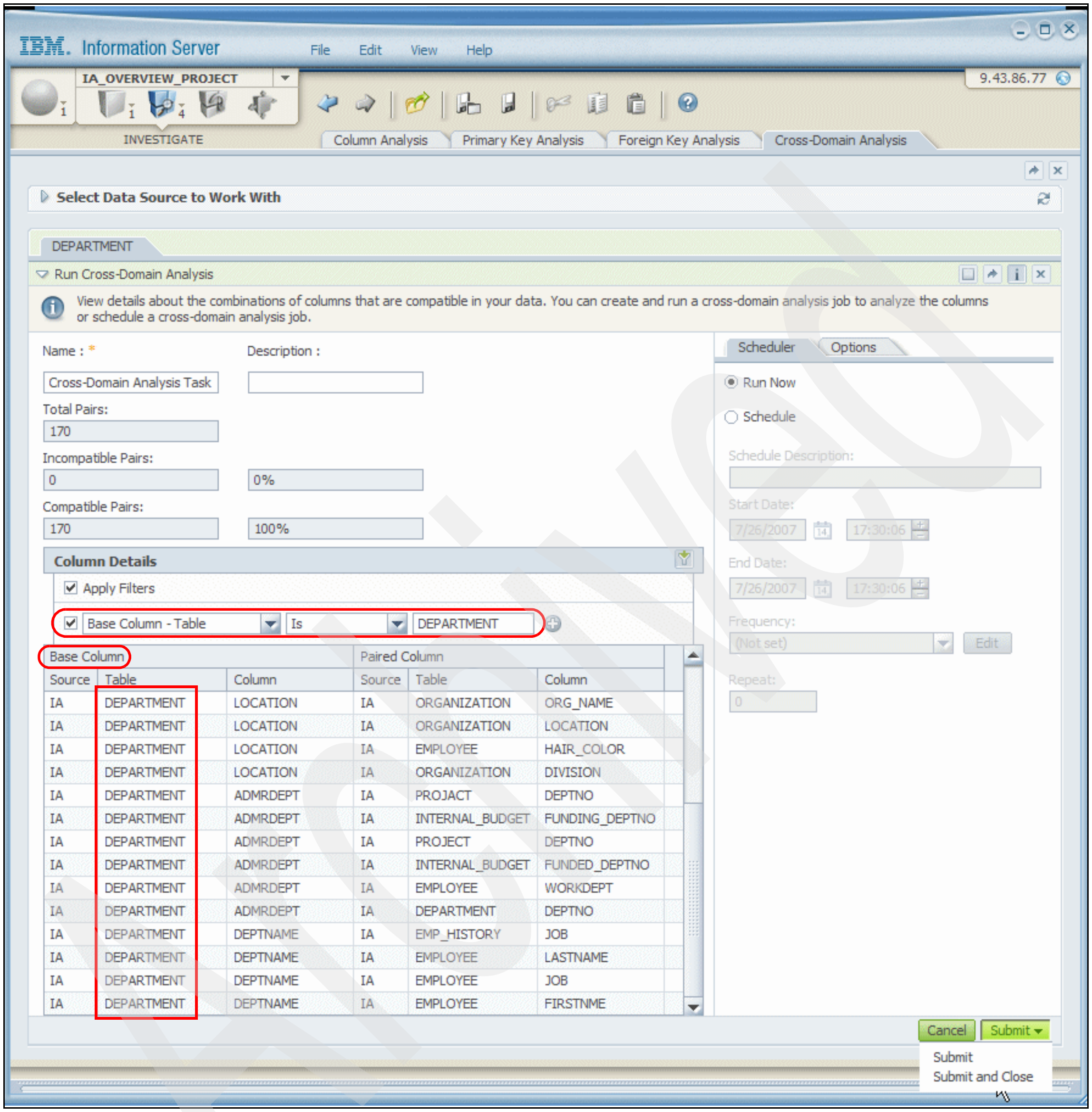
Frequency: (Not set) Edit

Repeat: 0

Column Details

Base Column			Paired Column		
Source	Table	Column	Source	Table	Column
IA	DEPARTMENT	DEPTNO	IA	PROJACT	DEPTNO
IA	DEPARTMENT	DEPTNO	IA	INTERNAL_BUDGET	FUNDING_DEPTNO
IA	DEPARTMENT	DEPTNO	IA	PROJECT	DEPTNO
IA	DEPARTMENT	DEPTNO	IA	INTERNAL_BUDGET	FUNDED_DEPTNO
IA	DEPARTMENT	DEPTNO	IA	EMPLOYEE	WORKDEPT
IA	DEPARTMENT	DEPTNO	IA	DEPARTMENT	ADMRDEPT
IA	DEPARTMENT	MGRNO	IA	PROJECT	RESPEMP
IA	DEPARTMENT	MGRNO	IA	ORGANIZATION	MANAGER
IA	DEPARTMENT	MGRNO	IA	EMPLOYEE	EMPNO
IA	DEPARTMENT	MGRNO	IA	EMP_HISTORY	EMPNO
IA	DEPARTMENT	LOCATION	IA	ORGANIZATION	ORG_NAME
IA	DEPARTMENT	LOCATION	IA	ORGANIZATION	LOCATION
IA	DEPARTMENT	LOCATION	IA	EMPLOYEE	HAIR_COLOR
IA	DEPARTMENT	LOCATION	IA	ORGANIZATION	DIVISION
IA	DEPARTMENT	ADMRDEPT	IA	PROJACT	DEPTNO
IA	DEPARTMENT	ADMRDEPT	IA	INTERNAL_BUDGET	FUNDING_DEPTNO
IA	DEPARTMENT	ADMRDEPT	IA	PROJECT	DEPTNO

Figure 1-243 Cross Domain Analysis on a set of tables 2/3



View details about the combinations of columns that are compatible in your data. You can create and run a cross-domain analysis job to analyze the columns or schedule a cross-domain analysis job.

Name : * Cross-Domain Analysis Task

Description :

Total Pairs: 170

Incompatible Pairs: 0 0%

Compatible Pairs: 170 100%

Column Details

☑ Apply Filters

☑ Base Column - Table | Is | DEPARTMENT

Base Column			Paired Column		
Source	Table	Column	Source	Table	Column
IA	DEPARTMENT	LOCATION	IA	ORGANIZATION	ORG_NAME
IA	DEPARTMENT	LOCATION	IA	ORGANIZATION	LOCATION
IA	DEPARTMENT	LOCATION	IA	EMPLOYEE	HAIR_COLOR
IA	DEPARTMENT	LOCATION	IA	ORGANIZATION	DIVISION
IA	DEPARTMENT	ADMRDEPT	IA	PROJACT	DEPTNO
IA	DEPARTMENT	ADMRDEPT	IA	INTERNAL_BUDGET	FUNDING_DEPTNO
IA	DEPARTMENT	ADMRDEPT	IA	PROJECT	DEPTNO
IA	DEPARTMENT	ADMRDEPT	IA	INTERNAL_BUDGET	FUNDED_DEPTNO
IA	DEPARTMENT	ADMRDEPT	IA	EMPLOYEE	WORKDEPT
IA	DEPARTMENT	ADMRDEPT	IA	DEPARTMENT	DEPTNO
IA	DEPARTMENT	DEPTNAME	IA	EMP_HISTORY	JOB
IA	DEPARTMENT	DEPTNAME	IA	EMPLOYEE	LASTNAME
IA	DEPARTMENT	DEPTNAME	IA	EMPLOYEE	JOB
IA	DEPARTMENT	DEPTNAME	IA	EMPLOYEE	FIRSTNME

● Run Now

○ Schedule

Schedule Description:

Start Date: 7/26/2007 17:30:06

End Date: 7/26/2007 17:30:06

Frequency: (Not set)

Repeat: 0

Figure 1-244 Cross Domain Analysis on a set of tables 3/3

1.11 Publish analysis results

You can view an analysis result and publish it to the metadata repository. You might want to publish statistics and annotations for a table or column to provide developers in additional suite components, such as IBM WebSphere DataStage or IBM WebSphere Business Glossary, access to analytical results.

In IBM WebSphere Information Analyzer, analysis occurs on a project basis. All the results of analysis are bound by the project. For example, you might have three projects that perform analysis on a table called “Address_ShipTo” and three sets of analysis results exist for that table. The results of those three analyses contain valid analysis; however, the administrator might want to make one of analysis results the official shared results.

Analysis results can be published to the source objects that they are associated with. Source objects are shared by all of the suite components. Therefore, if these source objects are referenced in IBM WebSphere DataStage or IBM WebSphere Business Glossary, those suite components also have access to the analysis results that are published.

For example, an Information Analyzer Project Administrator can create three projects: Address_primary, Address_compare, and Address_worldwide. Each of these projects analyze the table Address_ShipTo. While performing analysis, Information Analyzer Data Analysts can annotate the analysis results with notes that provide additional information to other reviewers. After the analysis review completes, the administrator might select the analysis results for the Address_ShipTo table from the Address_compare project as the enterprise standard results. The administrator can then publish those results to the metadata repository. The last analysis published overlays any prior publication.

A IBM WebSphere DataStage developer can then go into a IBM WebSphere DataStage job, look at the Address_ShipTo table and elect to look at the analysis results for that table. The developer can also view any notes or annotations created by the Information Analyzer Data Analyst. The developer can design or modify their job based on those results and notes. For example, the Information Analyzer Data Analyst might suggest that the data type of a particular column in the table should be changed, that a table should have a foreign key established, or that a table has invalid values that the job should eliminate.

When you no longer need published summaries of analysis results, you can delete them from the metadata repository.

In the following sections, we describe the process of publishing analysis results for a table and then reviewing it in another suite component IBM WebSphere DataStage.

Attention: In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

1.11.1 Publish an analysis result

We use the modified sample database described in Figure 1-11 on page 31 in the example that we describe here.

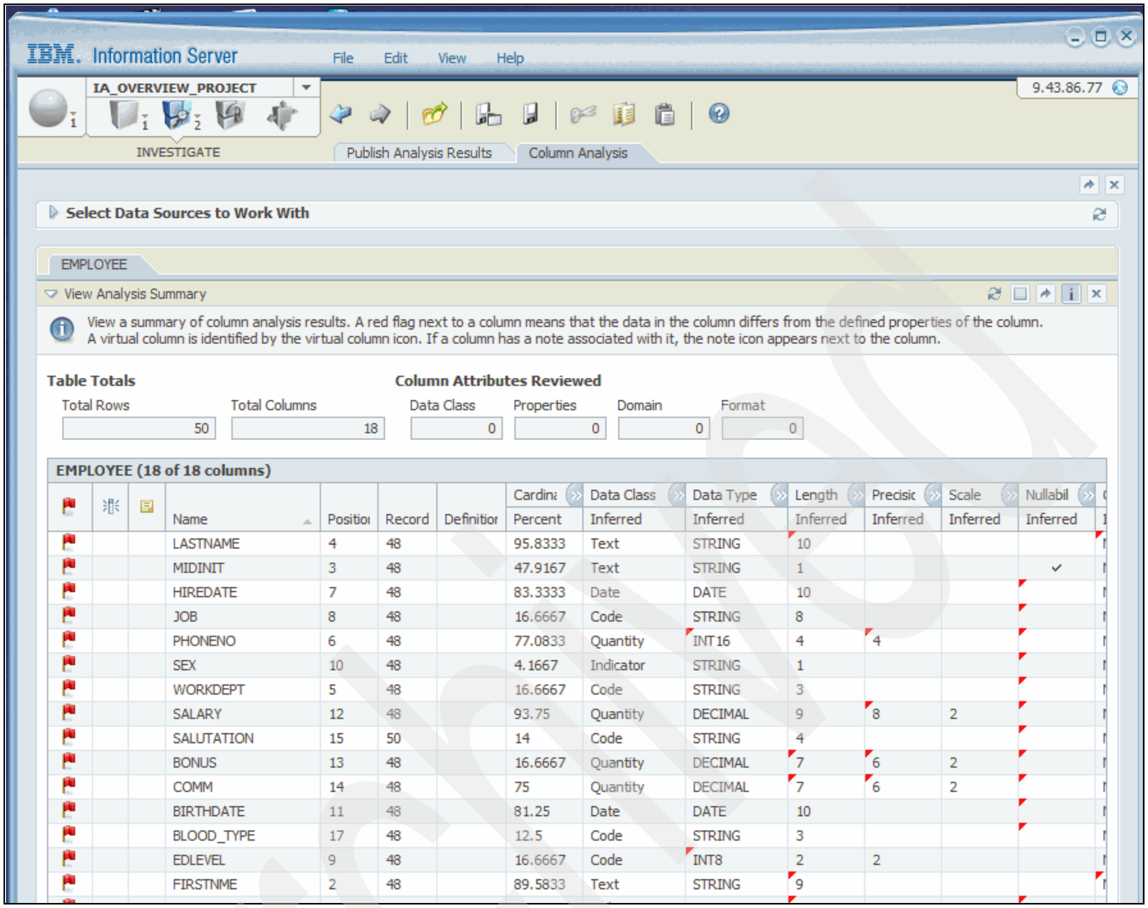
After opening a project of interest (IA_OVERVIEW_PROJECT in our case), run a column analysis on the EMPLOYEE table and view the results shown in Figure 1-245 on page 325.

You then choose to publish these results to the metadata repository as follows:

1. On the Investigate navigator menu in the console, select **Publish Analysis Results** as shown in Figure 1-246 on page 325.
2. In the Publish Analysis Results workspace, select the EMPLOYEE table.

   Note: Optionally, you can click **View Analysis Results** to review the analysis results in the Analysis Results Summary pane. We do not show that here.

   Click **Publish Analysis Results** in the Tasks pane as shown in Figure 1-247 on page 326 to prepare the analysis results for publishing.
3. In the Publish Analysis Results window in Figure 1-248 on page 326, select which analysis result you want to publish—the current analysis, the set checkpoint, or the baseline. In our case, we chose Current Analysis. You can optionally select whether you want to include notes (which we did not). Click **OK**.
4. Figure 1-249 on page 327 shows the status of the last publication in the Publish Analysis Results workspace. Your ID is attached as the publisher as shown.



Select Data Sources to Work With

EMPLOYEE

View Analysis Summary

View a summary of column analysis results. A red flag next to a column means that the data in the column differs from the defined properties of the column. A virtual column is identified by the virtual column icon. If a column has a note associated with it, the note icon appears next to the column.

Table Totals

Total Rows: 50

Total Columns: 18

Column Attributes Reviewed

Data Class: 0

Properties: 0

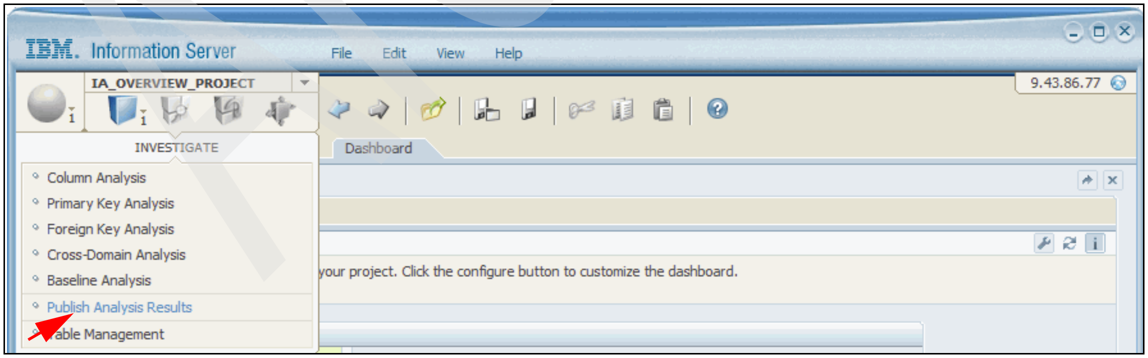
Domain: 0

Format: 0

EMPLOYEE (18 of 18 columns)

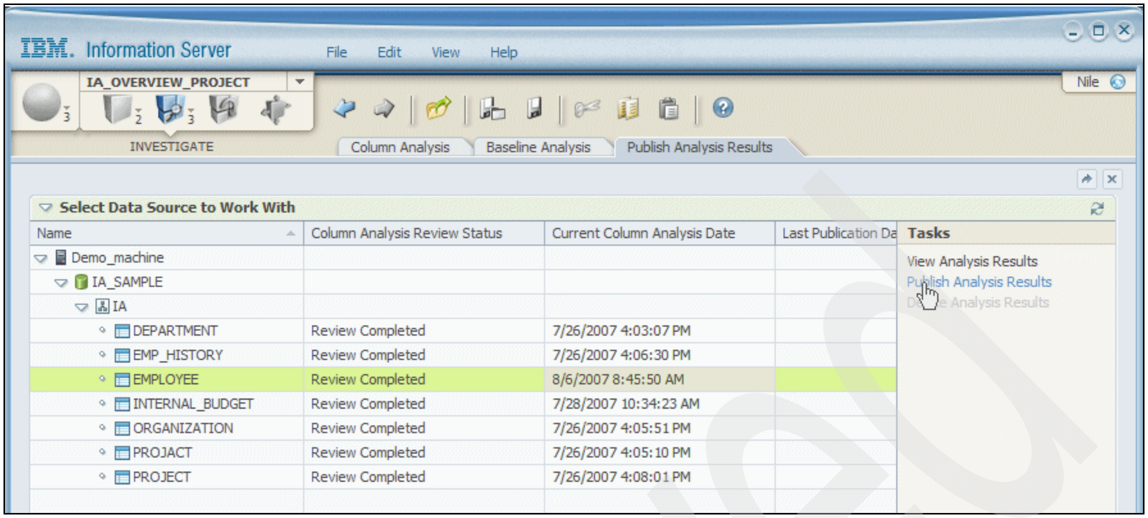
							Cardinz	Data Class	Data Type	Length	Precisic	Scale	Nullabil
🚩			Name	Positioı	Record	Definitior	Percent	Inferred	Inferred	Inferred	Inferred	Inferred	Inferred
🚩			LASTNAME	4	48		95.8333	Text	STRING	10			
🚩			MIDINIT	3	48		47.9167	Text	STRING	1			✓
🚩			HIREDATE	7	48		83.3333	Date	DATE	10			
🚩			JOB	8	48		16.6667	Code	STRING	8			
🚩			PHONENO	6	48		77.0833	Quantity	INT16	4	4		
🚩			SEX	10	48		4.1667	Indicator	STRING	1			
🚩			WORKDEPT	5	48		16.6667	Code	STRING	3			
🚩			SALARY	12	48		93.75	Quantity	DECIMAL	9	8	2	
🚩			SALUTATION	15	50		14	Code	STRING	4			
🚩			BONUS	13	48		16.6667	Quantity	DECIMAL	7	6	2	
🚩			COMM	14	48		75	Quantity	DECIMAL	7	6	2	
🚩			BIRTHDATE	11	48		81.25	Date	DATE	10			
🚩			BLOOD_TYPE	17	48		12.5	Code	STRING	3			
🚩			EDLEVEL	9	48		16.6667	Code	INT8	2	2		
🚩			FIRSTNME	2	48		89.5833	Text	STRING	9			

Figure 1-245 Publish Analysis Results 1/5



your project. Click the configure button to customize the dashboard.

Figure 1-246 Publish Analysis Results 2/5



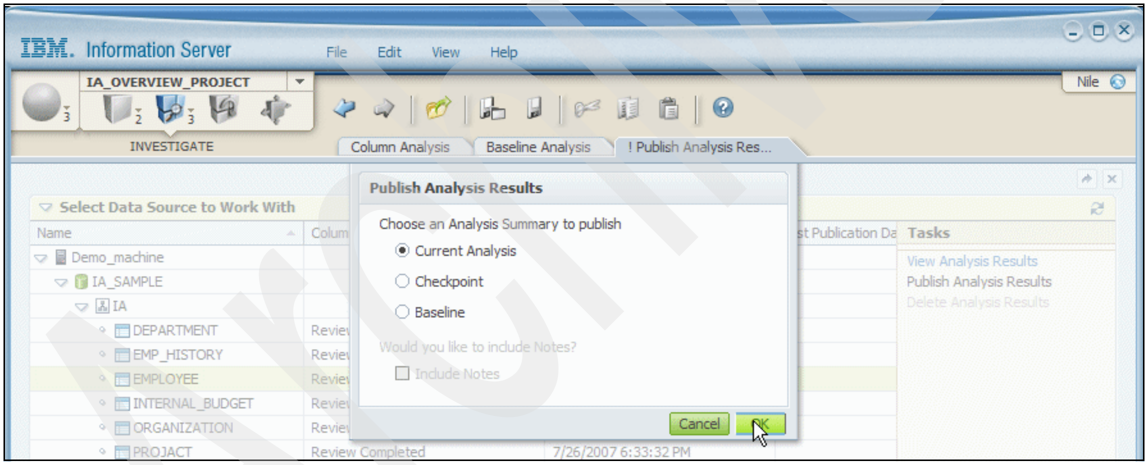
Select Data Source to Work With

Name	Column Analysis Review Status	Current Column Analysis Date	Last Publication Da
Demo_machine			
IA_SAMPLE			
IA			
DEPARTMENT	Review Completed	7/26/2007 4:03:07 PM	
EMP_HISTORY	Review Completed	7/26/2007 4:06:30 PM	
EMPLOYEE	Review Completed	8/6/2007 8:45:50 AM	
INTERNAL_BUDGET	Review Completed	7/28/2007 10:34:23 AM	
ORGANIZATION	Review Completed	7/26/2007 4:05:51 PM	
PROJACT	Review Completed	7/26/2007 4:05:10 PM	
PROJECT	Review Completed	7/26/2007 4:08:01 PM	

Tasks

- View Analysis Results
- Publish Analysis Results
- Delete Analysis Results

Figure 1-247 Publish Analysis Results 3/5



Publish Analysis Results

Choose an Analysis Summary to publish

● Current Analysis

○ Checkpoint

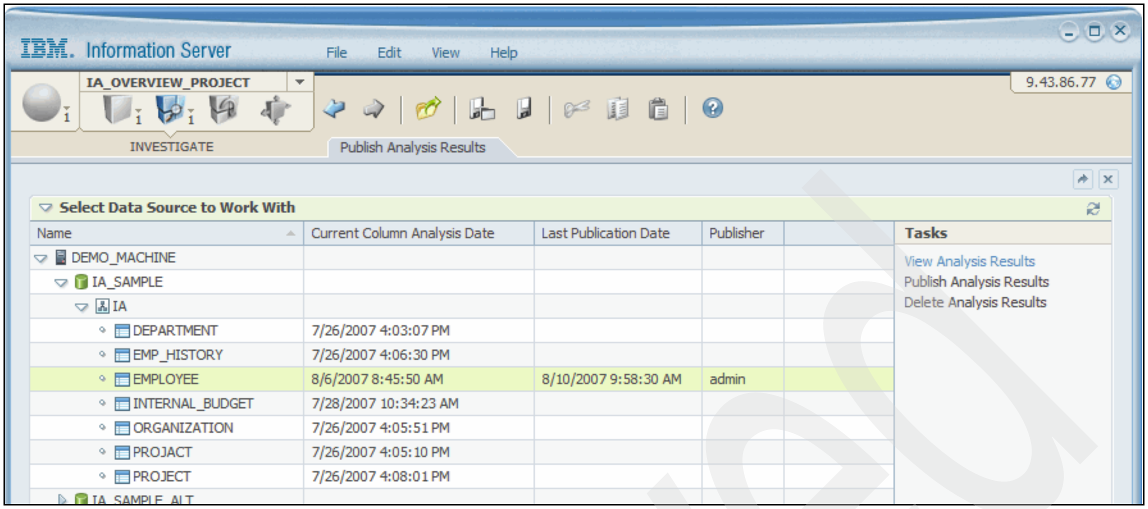
○ Baseline

Would you like to include Notes?

□ Include Notes

Cancel   OK

Figure 1-248 Publish Analysis Results 4/5



Select Data Source to Work With

Name	Current Column Analysis Date	Last Publication Date	Publisher
DEMO_MACHINE			
IA_SAMPLE			
IA			
DEPARTMENT	7/26/2007 4:03:07 PM		
EMP_HISTORY	7/26/2007 4:06:30 PM		
EMPLOYEE	8/6/2007 8:45:50 AM	8/10/2007 9:58:30 AM	admin
INTERNAL_BUDGET	7/28/2007 10:34:23 AM		
ORGANIZATION	7/26/2007 4:05:51 PM		
PROJACT	7/26/2007 4:05:10 PM		
PROJECT	7/26/2007 4:08:01 PM		
IA_SAMPLE_ALT			

Figure 1-249 Publish Analysis Results 5/5

1.11.2 Review a published analysis result in DataStage

To view the IBM WebSphere Information Analyzer published results in DataStage, you must first import the metadata of the EMPLOYEE table definition into DataStage after establishing connection details for the import as follows:

1. Launch the IBM WebSphere DataStage and QualityStage Designer by selecting **Start** → **All Programs** → **IBM Information Server** → **IBM WebSphere DataStage and QualityStage Designer from your Windows machine**.
2. Provide information about the project to which it should attach and click **OK** as shown in Figure 1-250 on page 328.
3. From the Menu bar, select **Repository** → **Metadata Sharing** → **Management** as shown in Figure 1-251 on page 329.
4. In the Import metadata using connectors window, select the **ODBC Connector** and click **Next** as shown in Figure 1-252 on page 329.
5. Select the IA_sample data source from the Data source drop-down list, supply the Username (db2inst1) and password, and click **Test connection** as shown in Figure 1-253 on page 330 and Figure 1-254 on page 330. A successful connection is shown in Figure 1-255 on page 331.
6. Click **Next** as shown in Figure 1-256 on page 331 to proceed to select the information to be imported.

7. Specify filtering criteria such as DB2INST1 from the Schema drop-down list and request views and tables by checking the appropriate boxes, before clicking **Next** as shown in Figure 1-257 on page 332.

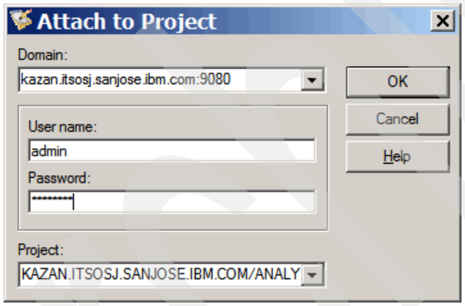
8. Select EMPLOYEE from the choices, check the boxes to include the primary key, foreign keys, and indexes, and click **Next** as shown in Figure 1-258 on page 332.

9. Confirm the EMPLOYEE table import selection by clicking **Import** as shown in Figure 1-259 on page 333.

10. In the Select Folder window, select Table Definitions for the Item name and click **OK** as shown in Figure 1-260 on page 333.

11. You can now view the IBM WebSphere Information Analyzer published results for the EMPLOYEE table (right click and select **Properties**—not shown here) in IBM WebSphere DataStage as shown in Figure 1-261 on page 334 (general information about the published result) and Figure 1-262 on page 335 (published analytical information).

    To view column level information, select the **Columns** tab, right-click the column of interest to view details (not shown here).



Domain: kazan.itsosj.sanjose.ibm.com:9080

User name: admin

Password: ********

Project: KAZAN.ITSOSJ.SANJOSE.IBM.COM/ANALY

Figure 1-250 Review a published analysis result in DataStage 1/13

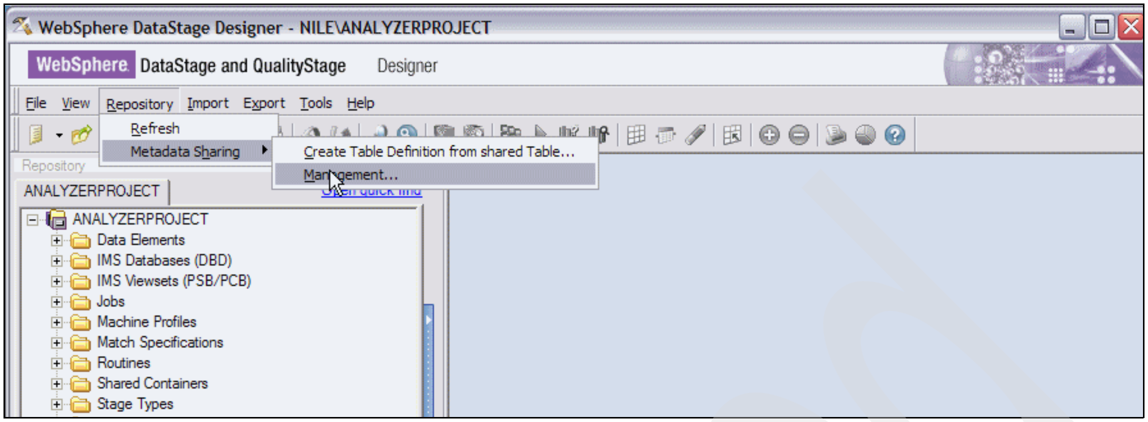
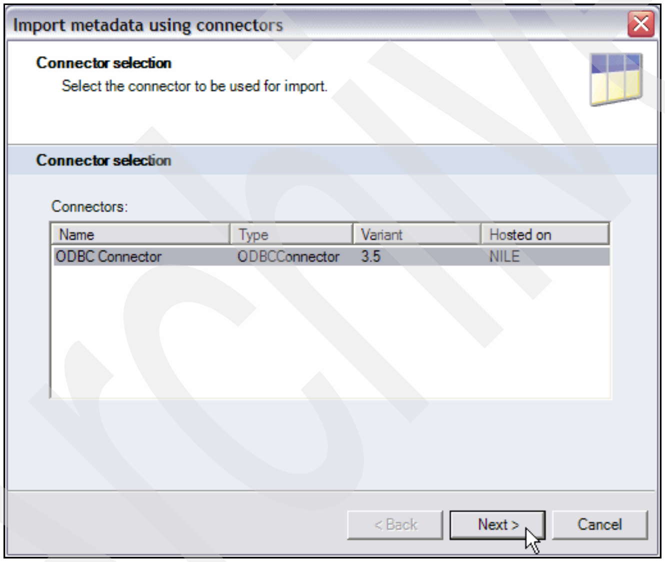


Figure 1-251 Review a published analysis result in DataStage 2/13



Connector selection

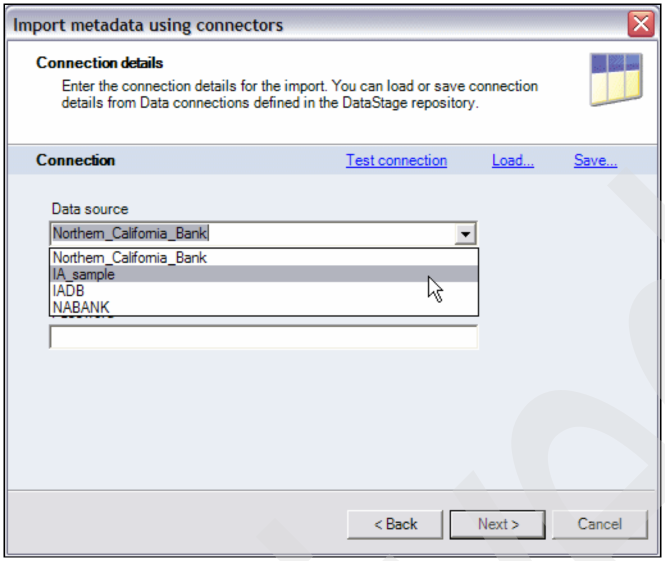
Select the connector to be used for import.

Connector selection

Connectors:

Name	Type	Variant	Hosted on
ODBC Connector	ODBCConnector	3.5	NILE

Figure 1-252 Review a published analysis result in DataStage 3/13


Import metadata using connectors

Connection details

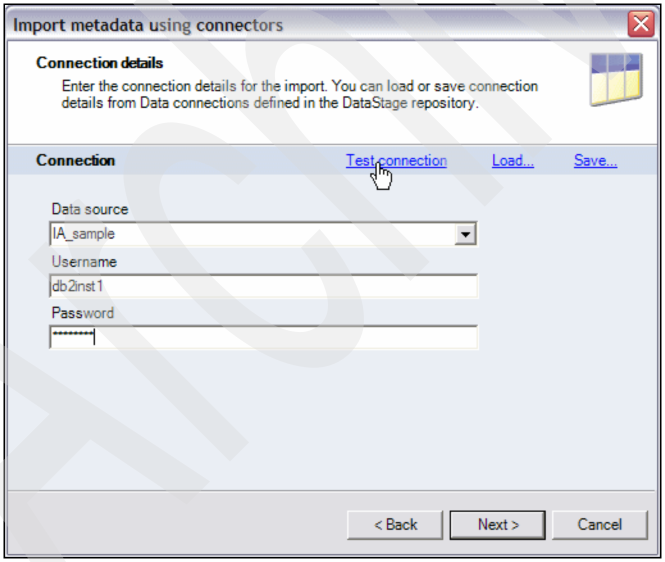
Enter the connection details for the import. You can load or save connection details from Data connections defined in the DataStage repository.

Connection

Data source: Northern_California_Bank

Northern_California_Bank
IA_sample
IADB
NABANK

Figure 1-253 Review a published analysis result in DataStage 4/13


Import metadata using connectors

Connection details

Enter the connection details for the import. You can load or save connection details from Data connections defined in the DataStage repository.

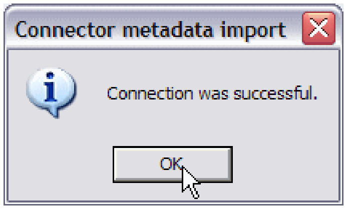
Connection

Data source: IA_sample

Username: db2inst1

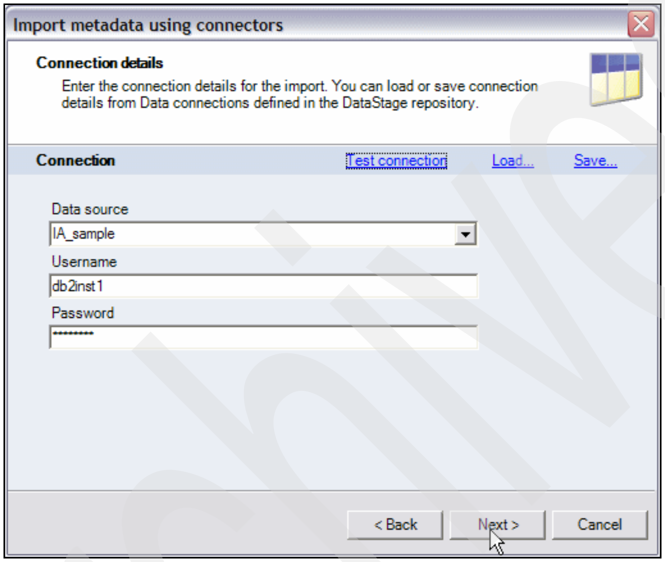
Password: *********

Figure 1-254 Review a published analysis result in DataStage 5/13


Connector metadata import

Connection was successful.

Figure 1-255 Review a published analysis result in DataStage 6/13


Import metadata using connectors

Connection details

Enter the connection details for the import. You can load or save connection details from Data connections defined in the DataStage repository.

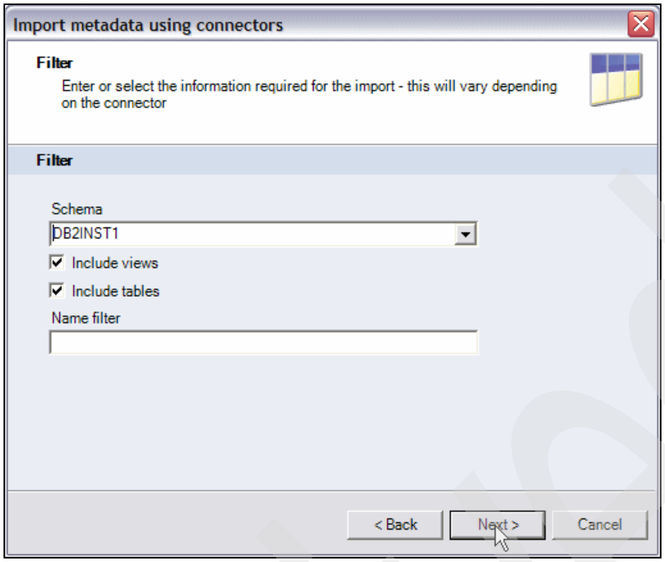
Connection

Data source: IA_sample

Username: db2inst1

Password: ********

Figure 1-256 Review a published analysis result in DataStage 7/13



Import metadata using connectors

Filter

Enter or select the information required for the import - this will vary depending on the connector

Filter

Schema

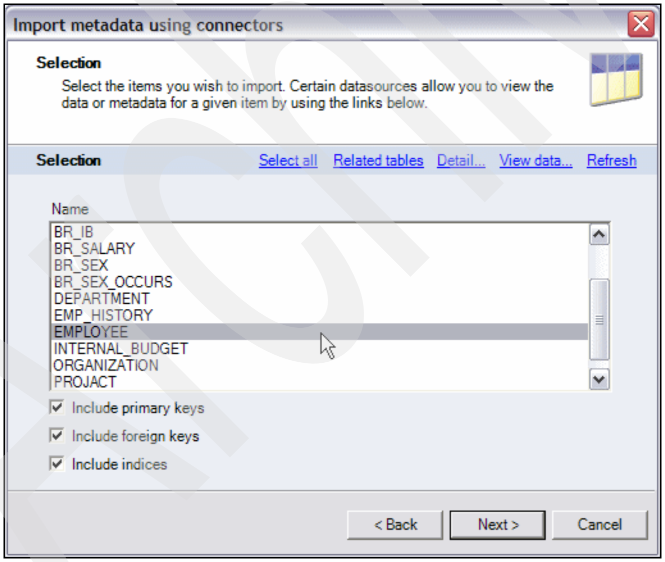
DB2INST1

☑ Include views

☑ Include tables

Name filter

Figure 1-257 Review a published analysis result in DataStage 8/13



Import metadata using connectors

Selection

Select the items you wish to import. Certain datasources allow you to view the data or metadata for a given item by using the links below.

Selection

Name

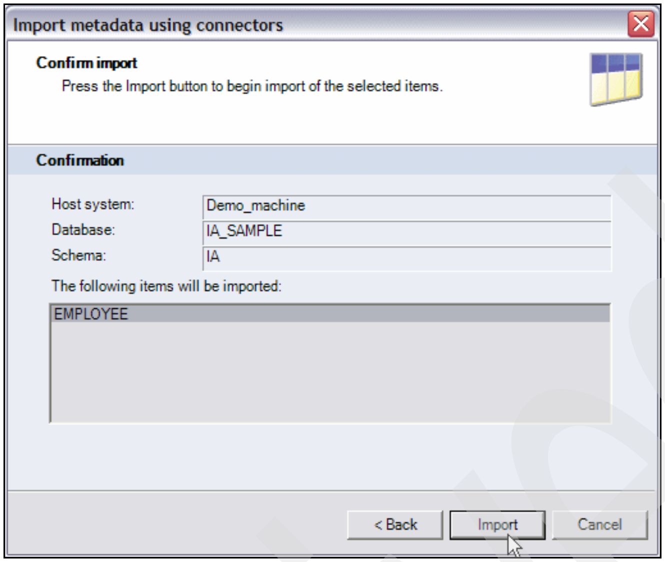
BR_IB
BR_SALARY
BR_SEX
BR_SEX_OCCURS
DEPARTMENT
EMP_HISTORY
EMPLOYEE
INTERNAL_BUDGET
ORGANIZATION
PROJACT

☑ Include primary keys

☑ Include foreign keys

☑ Include indices

Figure 1-258 Review a published analysis result in DataStage 9/13



Import metadata using connectors

Confirm import

Press the Import button to begin import of the selected items.

Confirmation

Host system: Demo_machine

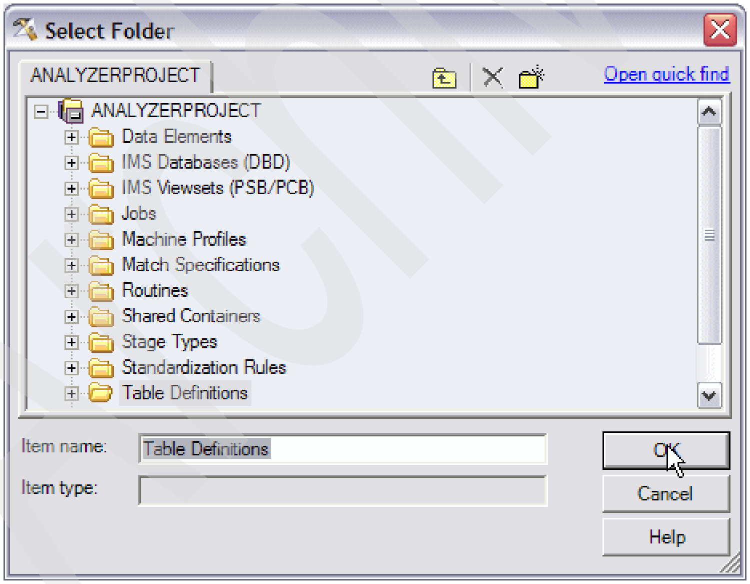
Database: IA_SAMPLE

Schema: IA

The following items will be imported:

EMPLOYEE

Figure 1-259 Review a published analysis result in DataStage 10/13



Select Folder

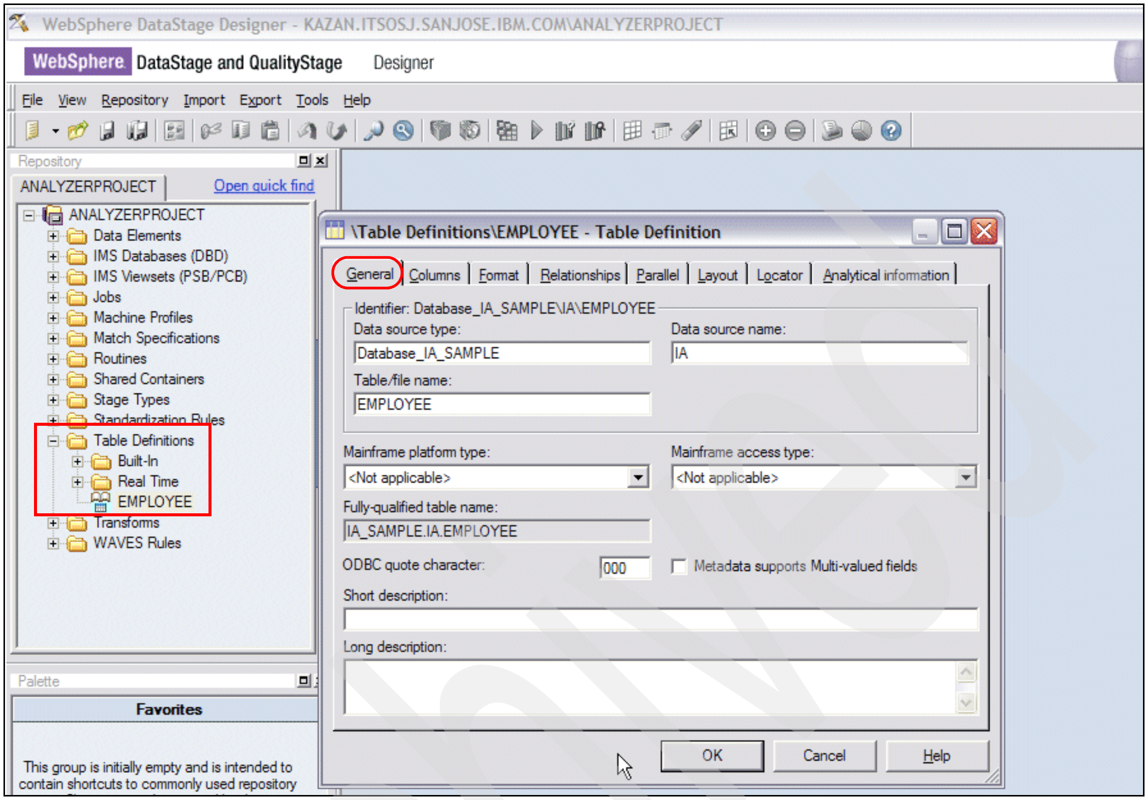
ANALYZERPROJECT

- ANALYZERPROJECT
  - Data Elements
  - IMS Databases (DBD)
  - IMS Viewsets (PSB/PCB)
  - Jobs
  - Machine Profiles
  - Match Specifications
  - Routines
  - Shared Containers
  - Stage Types
  - Standardization Rules
  - Table Definitions

Item name: Table Definitions

Item type:

Figure 1-260 Review a published analysis result in DataStage 11/13



Identifier: Database_IA_SAMPLE\IA\EMPLOYEE

Data source type: Database_IA_SAMPLE

Data source name: IA

Table/file name: EMPLOYEE

Mainframe platform type: <Not applicable>

Mainframe access type: <Not applicable>

Fully-qualified table name: IA_SAMPLE.IA.EMPLOYEE

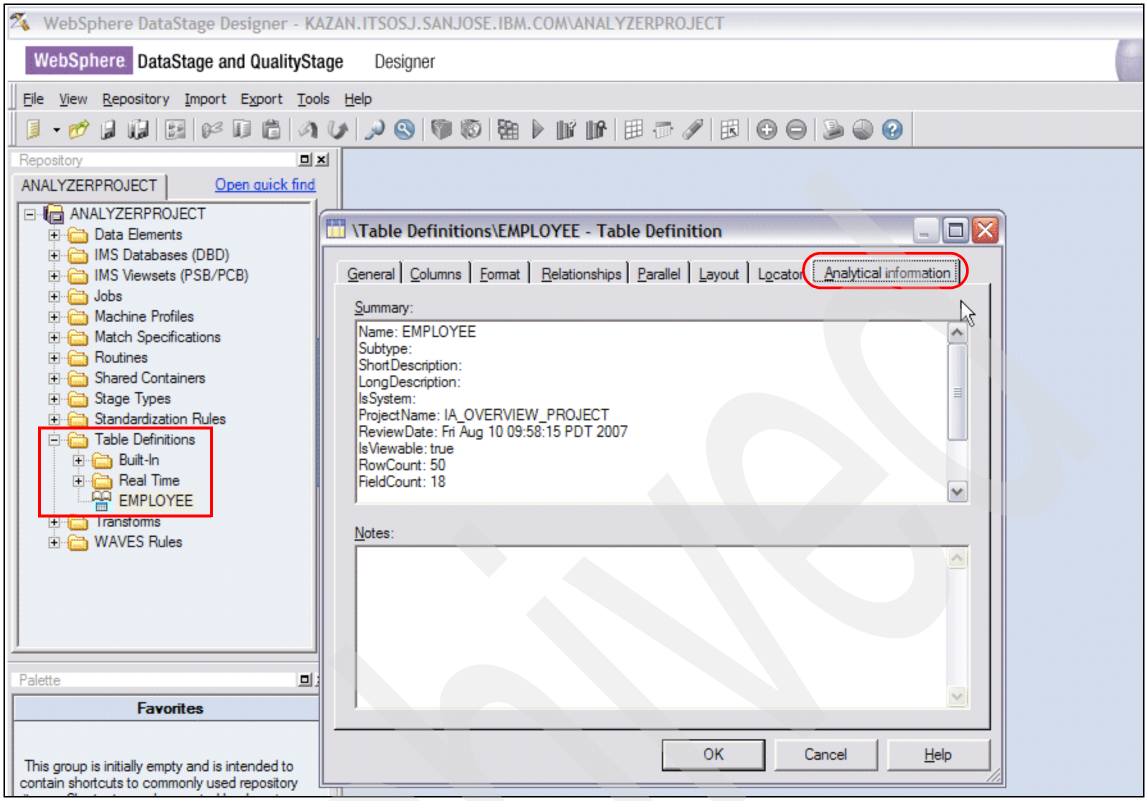
ODBC quote character: 000

☐ Metadata supports Multi-valued fields

Short description:

Long description:

Figure 1-261 Review a published analysis result in DataStage 12/13



Name: EMPLOYEE
Subtype:
ShortDescription:
LongDescription:
IsSystem:
ProjectName: IA_OVERVIEW_PROJECT
ReviewDate: Fri Aug 10 09:58:15 PDT 2007
IsViewable: true
RowCount: 50
FieldCount: 18

Figure 1-262 Review a published analysis result in DataStage 13/13

1.12 IBM WebSphere AuditStage business rule validation

IBM WebSphere AuditStage is a software tool that enables you to apply professional quality control methods to manage the accuracy, consistency, completeness, and integrity of information stored in databases. By employing technology that integrates Total Quality Management (TQM) principles with data modeling and relational database concepts, IBM WebSphere AuditStage diagnoses data quality problems and facilitates data cleanup efforts.

With IBM WebSphere AuditStage, you can quickly and easily develop:

- A Domain Analysis that determines existing values in critical data elements
- Metadata that records information about critical data elements
- An assessment of the completeness and validity of critical data elements—business rule compliance
- Data Filters that identify data defects according to the principles of data quality and your business rules and specifications
- Filter Sets that organize related Data Filters
- Metrics that quantify data quality by attaching business significance to Data Filters
- Exception Reports that provide detailed listings of defective data that allow you to pinpoint the sources of defects
- Metric Reports that present and format high-level information about data quality
- Trend Charts that monitor defect levels in databases in order to maintain statistical control of systems
- Analysis Charts that prioritize data quality issues to ensure time and resources are invested in the most critical areas
- Import Scripts that gather and consolidate data from any number of sources (available only when using the internal database)
- Create Table Scripts that build table structures
- SQL Scripts that perform ad hoc queries or updates
- Macros that run groups of Objects in batches at user-specified intervals.

In addition, you can generate Data Filters automatically using predefined templates or user-created templates, and you can run two predefined data quality analysis reports.

Note: In this section, we only focus on the business rule compliance capability of IBM WebSphere AuditStage that complements IBM WebSphere Information Analyzer's validity functionality.

Business Rule Compliance evaluates the quality of data in terms of specific business rules involving multiple data fields within or across records (or rows) that are logically related. In most cases, the type of business rules needed for business rule analysis will not be documented or even explicitly known before the evaluation begins. Therefore, business rules applicable to data will need to be developed, or at least refined, for this analysis. The most common sources for developing explicit business rules applicable to data quality work are knowledgeable people (subject matter experts), system documentation, and occasionally metadata repositories. In addition, results from Level 0 Domain Analysis and Level 1 Completeness and Validity Assessment can suggest additional business rules.

To illustrate an example of business rule analysis, consider a medical database:

- One data element is a code representing the various sites where medical treatment can be given. Valid sites include an ambulance, a physician's office, an emergency room, an operating room, and so forth.
- Another data element is a code representing the various procedures that are offered. Valid procedures include a routine physical exam, various types of laboratory work, open-heart surgery, delivery of a baby, and so forth.

The business rule states the legitimate relationships between site codes and procedure codes. A particular patient record includes the site code for ambulance and the procedure code for open-heart surgery. Each of these codes taken separately passes a validity test. When taken together, however, the resulting information is suspect in that open-heart surgery is not usually performed in an ambulance.

There are virtually endless possibilities for designing business rules applicable to data. Therefore, when doing this level of analysis, there are two recommendations to consider when developing and using business rules:

- Although many potential business rules can be developed and tried in IBM WebSphere AuditStage for their effectiveness in determining data quality, a limited, but prioritized, set of business rules should be selected for final use in analysis. This selection is based on their impact to the business or their ability to measure the integrity and reliability of key information taken from the data environment.

- A business rule will need to be specified, reviewed, and tested using IBM WebSphere AuditStage and then refined with the realities of the actual data before it can be considered accurate and precise enough for use in Business Rule Compliance results.

There are four common types of business rules:

- Valid-value combinations rules

  These rules entail two or more data elements that have a specific and valid relationship to one another. For example, the combination of city, state, and ZIP code must be valid as documented by the U.S. Postal Service. Assessing this type of rule in IBM WebSphere AuditStage lends itself well to the use of concatenation and a table of valid combinations.

- Computational rules

  These rules calculate either within records or across records:

  - Equation

    This type of computational rule uses two or more data elements in a calculation and compares them against another element. An example of such a rule is the rate of pay multiplied by hours worked equals the paid amount.

  - Set

    This type of computational rule uses two different tables. The sum of the detailed records in one table is compared with a summary record in a different table. For example, the sum of all order records for the year equals the amount in the order summary record.

- Time rules

  These rules compare one point in time against another either in the same record or in a different record:

  - Range

    This type of time rule requires that a point in time falls between a minimum or maximum time. For example, a project activity date must fall between the project start date and the project end date.

  - Sequence

    This type of time rule states that the time of a particular event must have a specific relationship to the time of another event. For example, the date of deposition must be greater than or equal to the date of filing and less than the date of trial.

- If…Then…Else rules

  These rules establish the relationship of two or more conditions. An example of such a rule is if the account status is closed, then there must not be any orders pending.

For complete details on IBM WebSphere AuditStage functionality, refer to *Ascential AuditStage Version 7.0.1 User's Guide*, Part No. 00D-001AS701.

In the following sections, we describe briefly the configuration of the data source name (DSN) followed by some business rule examples using the sample set of tables described in Figure 1-11 on page 31.

> **Attention:** In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

1.12.1 Configure the DSN

You need to configure the data source which will be accessed by IBM WebSphere AuditStage during business rule validation.

Figure 1-263 on page 341 through Figure 1-275 on page 347 show the configuration of a data source using an ODBC driver as follows:

1. Launch IBM WebSphere AuditStage by clicking **Start** → **All Programs** → **Ascential AuditStage™** → **AuditStage**. Then, log in by supplying the user ID and password. This action is not shown here.
2. From the File menu, select **New** to create a new project as shown in Figure 1-263 on page 341.
3. Provide the File Name (IA_SAMPLE) for the name of the project and click **Save** as shown in Figure 1-264 on page 341.
4. In the Select Data Source Type window, check External database (through ODBC) and click **OK** as shown in Figure 1-265 on page 342.
5. In the Select Data Source window, click **New** to add a new file data source as shown in Figure 1-266 on page 342.

> **Note:** The File Data Source tab lets you connect with a data source that has file data source names (DSNs). A file-based data source, not necessarily user-dedicated nor local to a computer, can be shared among all users who have the same drivers installed.
>
> The File Data Source tab displays all file DSNs and subdirectories of the directory indicated in the Look in box. Double-clicking a DSN connects to the data source.
>
> The Look in field indicates the directory for which the subdirectories and file DSNs are listed in the window below. Clicking the down arrow alongside this text box displays the entire directory structure.

6. We are creating a machine data source. Therefore, in the Select Data Source windows, click the **Machine Data Source** tab and click **New** to create a new data source as shown in Figure 1-267 on page 343.
7. In the Create New Data Source window, check System Data Source (Applies to this machine only) which creates a data source which is specific to this machine, and usable by any user who logs on to this machine as shown in Figure 1-268 on page 343. Click **Next.**

> **Note:** User Data Sources are specific to a user on this machine, while System Data Sources are usable by any user who logs on to this machine.

8. In the next window, select the IBM DB2 ODBC DRIVER as the driver for which you want to set up a data source as shown in Figure 1-269 on page 344, and click **Next**.
9. Click **Finish** in the new Create New Data Source window as shown in Figure 1-270 on page 344 to create the data source.
10. The ODBC IBM DB2 DRIVER prompts for information about the DB2 database to register. Provide the Data source name (IA_SAMPLE) and Database alias (IA) as shown in Figure 1-271 on page 345, and click **OK**.
11. In the Select Data Source window, select the Data Source Name (IA_SAMPLE) and click **OK** source name to connect to it as shown in Figure 1-272 on page 345.
12. In the Connect to DB2 Database window, provide the user ID and Password to connect to the database in Share mode as shown in Figure 1-273 on page 346, and click **OK**.
13. In the Select a Source Database window, select DB2 (UDB) as shown in Figure 1-274 on page 347.

14. At the prompt shown in Figure 1-275 on page 347, confirm it is a DB2 (UDB) database by clicking Yes.

15. Next, click **Continue** as shown in Figure 1-274 on page 347 to complete the configuration of the DSN.

You can now proceed to define business rules for this data source.

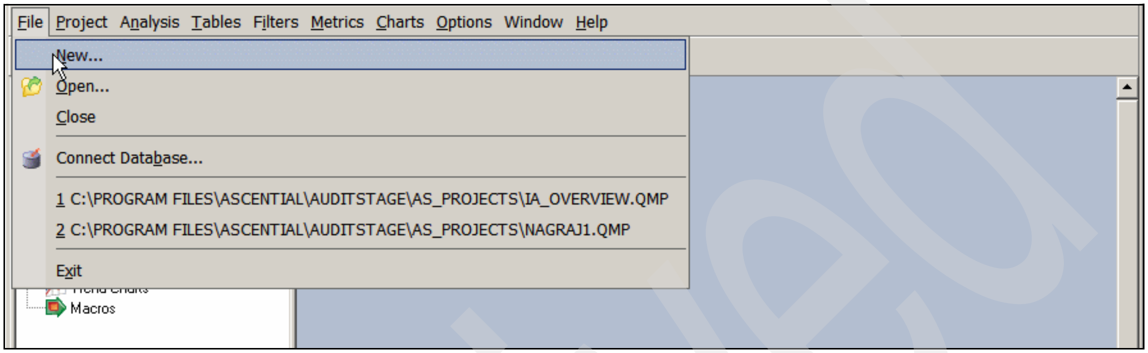
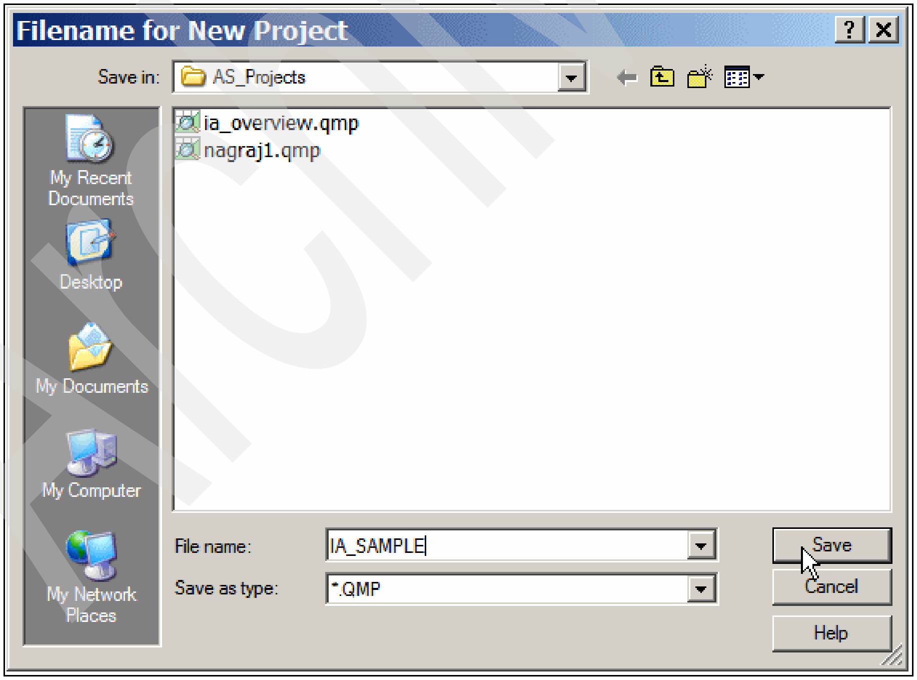

File
Project
Analysis
Tables
Filters
Metrics
Charts
Options
Window
Help
New...
Open...
Close
Connect Database...
1 C:\PROGRAM FILES\ASCENTIAL\AUDITSTAGE\AS_PROJECTS\IA_OVERVIEW.QMP
2 C:\PROGRAM FILES\ASCENTIAL\AUDITSTAGE\AS_PROJECTS\NAGRAJ1.QMP
Exit
Macros

Figure 1-263 Configure the DSN 1/13



Save in: AS_Projects

File name: IA_SAMPLE

Save as type: *.QMP

Figure 1-264 Configure the DSN 2/13

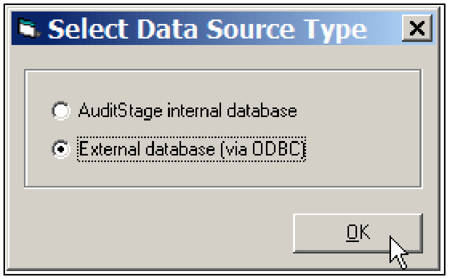
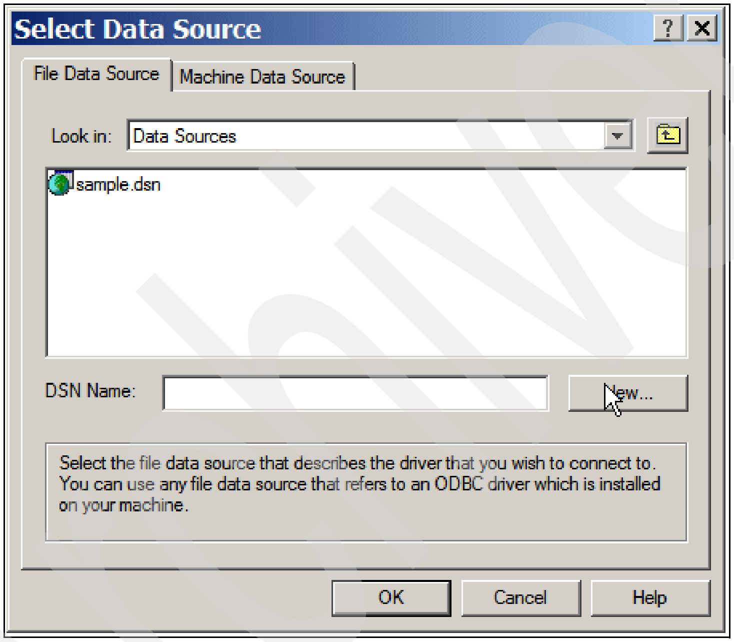
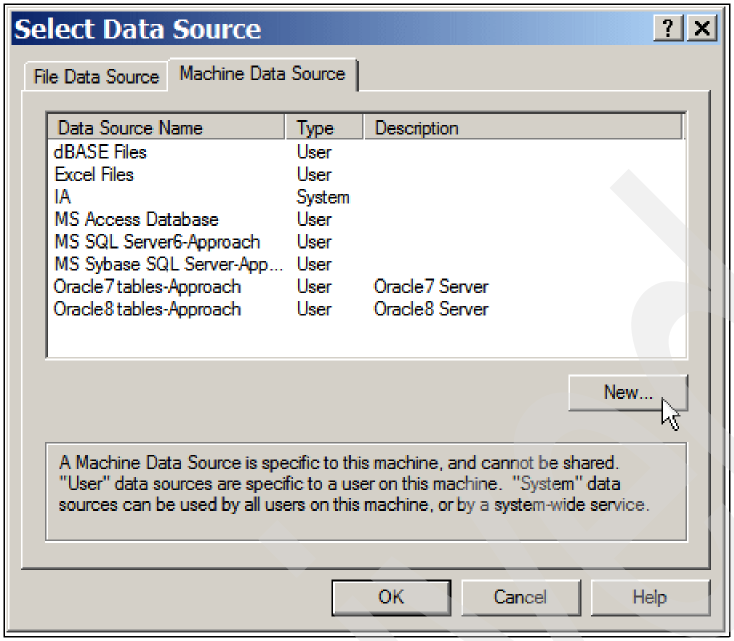

Select Data Source Type
○ AuditStage internal database
● External database (via ODBC)
OK

Figure 1-265 Configure the DSN 3/13



Select the file data source that describes the driver that you wish to connect to. You can use any file data source that refers to an ODBC driver which is installed on your machine.

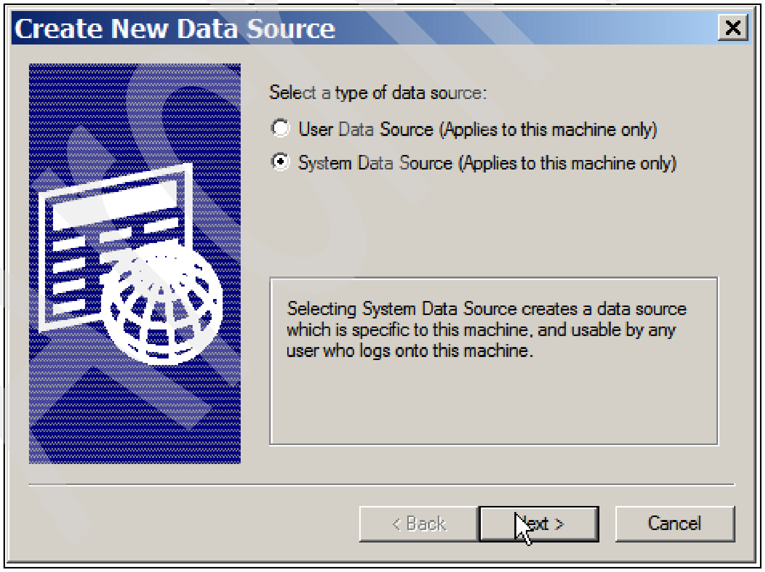
Figure 1-266 Configure the DSN 4/13



Data Source Name	Type	Description
dBASE Files	User	
Excel Files	User	
IA	System	
MS Access Database	User	
MS SQL Server6-Approach	User	
MS Sybase SQL Server-App...	User	
Oracle7 tables-Approach	User	Oracle7 Server
Oracle8 tables-Approach	User	Oracle8 Server

A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

Figure 1-267 Configure the DSN 5/13



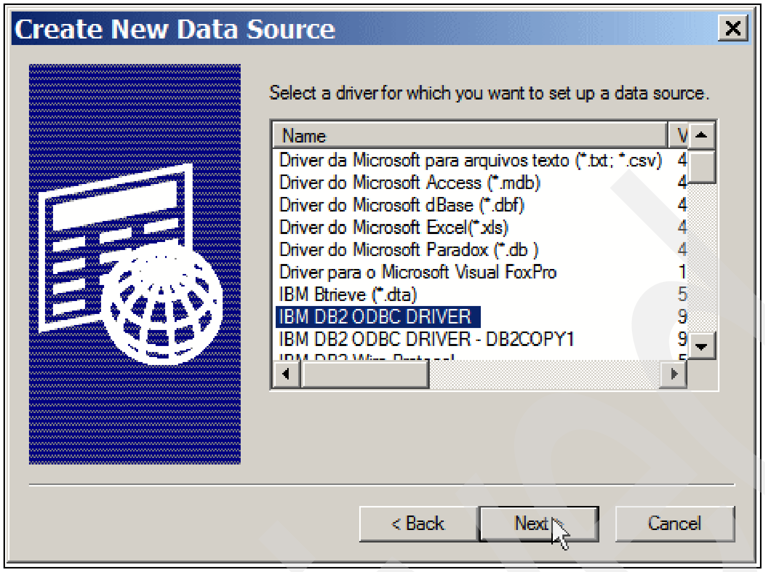
Select a type of data source:

○ User Data Source (Applies to this machine only)

● System Data Source (Applies to this machine only)

Selecting System Data Source creates a data source which is specific to this machine, and usable by any user who logs onto this machine.

Figure 1-268 Configure the DSN 6/13

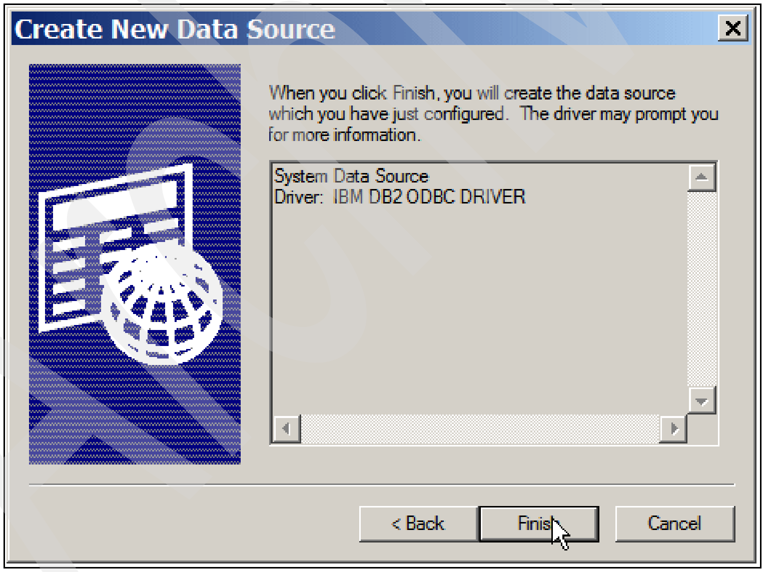


Create New Data Source

Select a driver for which you want to set up a data source.

Name	V
Driver da Microsoft para arquivos texto (*.txt; *.csv)	4
Driver do Microsoft Access (*.mdb)	4
Driver do Microsoft dBase (*.dbf)	4
Driver do Microsoft Excel(*.xls)	4
Driver do Microsoft Paradox (*.db)	4
Driver para o Microsoft Visual FoxPro	1
IBM Btrieve (*.dta)	5
IBM DB2 ODBC DRIVER	9
IBM DB2 ODBC DRIVER - DB2COPY1	9
IBM DB2 Wire Protocol	5

Figure 1-269 Configure the DSN 7/13

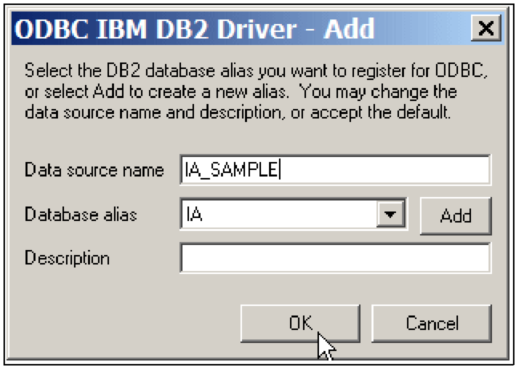


Create New Data Source

When you click Finish, you will create the data source which you have just configured. The driver may prompt you for more information.

System Data Source
Driver: IBM DB2 ODBC DRIVER

Figure 1-270 Configure the DSN 8/13



ODBC IBM DB2 Driver - Add

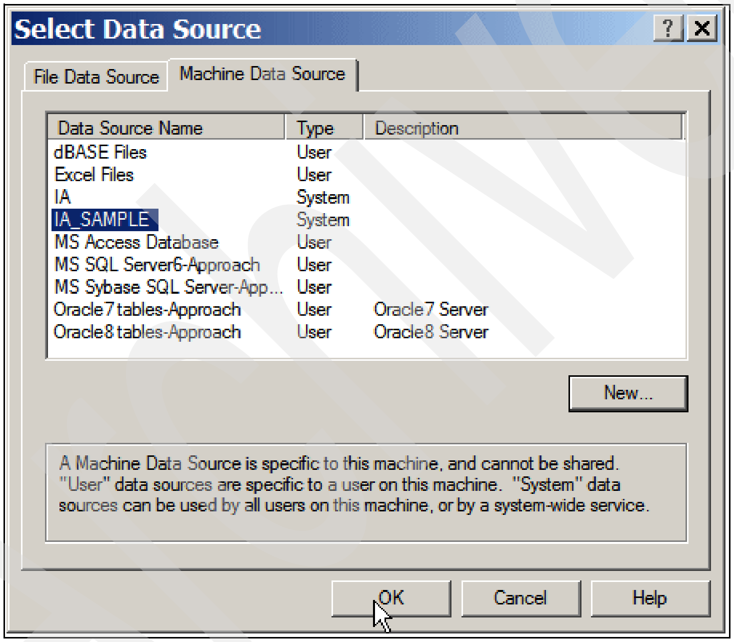
Select the DB2 database alias you want to register for ODBC, or select Add to create a new alias. You may change the data source name and description, or accept the default.

Data source name: IA_SAMPLE

Database alias: IA

Description:

Figure 1-271 Configure the DSN 9/13

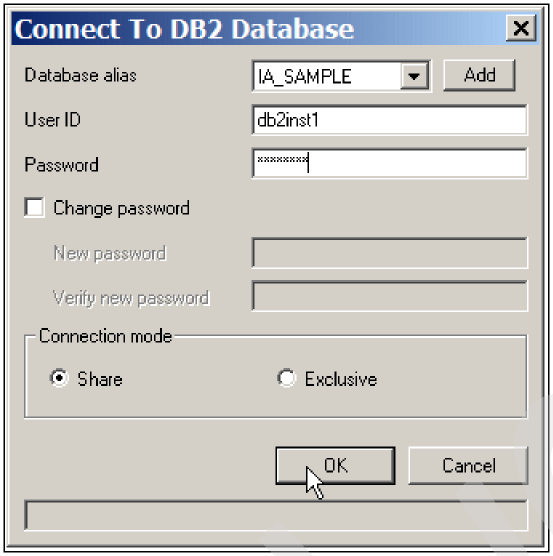


Select Data Source

Data Source Name	Type	Description
dBASE Files	User	
Excel Files	User	
IA	System	
IA_SAMPLE	System	
MS Access Database	User	
MS SQL Server6-Approach	User	
MS Sybase SQL Server-App...	User	
Oracle7 tables-Approach	User	Oracle7 Server
Oracle8 tables-Approach	User	Oracle8 Server

A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

Figure 1-272 Configure the DSN 10/13



Connect To DB2 Database

Database alias: IA_SAMPLE

User ID: db2inst1

Password: ********

☐ Change password

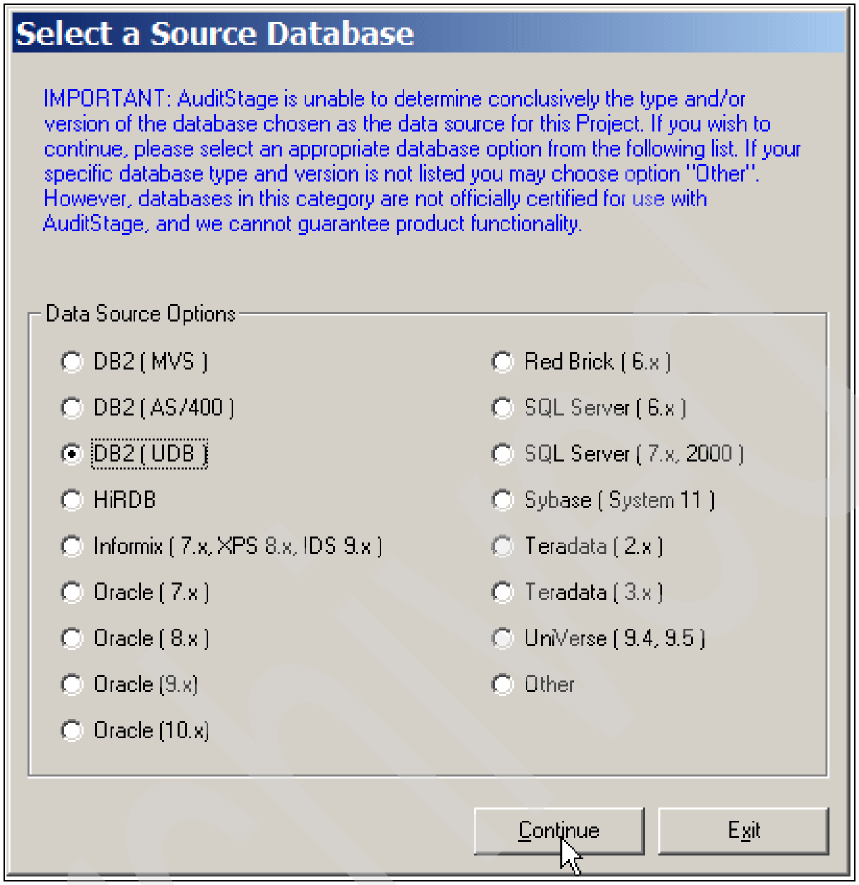
New password:

Verify new password:

Connection mode

● Share ○ Exclusive

Figure 1-273 Configure the DSN 11/13



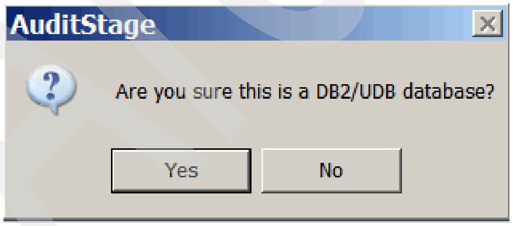
Select a Source Database

IMPORTANT: AuditStage is unable to determine conclusively the type and/or version of the database chosen as the data source for this Project. If you wish to continue, please select an appropriate database option from the following list. If your specific database type and version is not listed you may choose option "Other". However, databases in this category are not officially certified for use with AuditStage, and we cannot guarantee product functionality.

Data Source Options

- ○ DB2 (MVS)
- ○ DB2 (AS/400)
- ● DB2 (UDB)
- ○ HiRDB
- ○ Informix (7.x, XPS 8.x, IDS 9.x)
- ○ Oracle (7.x)
- ○ Oracle (8.x)
- ○ Oracle (9.x)
- ○ Oracle (10.x)
- ○ Red Brick (6.x)
- ○ SQL Server (6.x)
- ○ SQL Server (7.x, 2000)
- ○ Sybase (System 11)
- ○ Teradata (2.x)
- ○ Teradata (3.x)
- ○ UniVerse (9.4, 9.5)
- ○ Other

Figure 1-274 Configure the DSN 12/13



AuditStage

Are you sure this is a DB2/UDB database?

Figure 1-275 Configure the DSN 13/13

1.12.2 Business rule examples

Data Filters are IBM WebSphere AuditStage Objects that are used to evaluate the quality of specific data elements. Data Filters specify data characteristics and find instances where those characteristics are matched. These characteristics are violations of basic principles of data quality or user-defined business rules.

A Data Filter is built from one or more conditions. Each condition has three main elements: Source Data, Type of Check, and Reference Data (in some cases a reference data expression is not necessary). These conditions are linked together using the AND or OR operators. When the Data Filter definition is complete, the form of output for the Data Filter must be specified.

After you have developed and saved a Data Filter, you can rerun it on changing data. When a Data Filter runs, it finds exceptions, which are rows that match the specifications of the Data Filter.

> **Important:** In most cases, exceptions are defective data entities. However, Data Filters can also be used to pinpoint data which is *not* flawed but differs in some way from “normal” data. For example, you might use a Data Filter to measure the number of transactions that have been made with a certain company.

An important concept to understand is that Data Filters are based on the relational database model and SQL and are, therefore, declarative or non-procedural. This means that you do not need to tell IBM WebSphere AuditStage how to look for exceptions. You can tell it what constitutes an exception, and IBM WebSphere AuditStage takes care of the processing and optimization.

Data Filters have two levels of complexity:

- A simple Data Filter finds rows in one table for which one column has one characteristic. There are ten predefined characteristics, or Data Filter types such as (NOT) CONTAINS, (NOT) EQUALS, (NOT) EXISTS, (NOT) FORMAT, (NOT) OCCURS, (NOT) IN RANGE, (NOT) IN REF COL, (NOT) IN REF LIST, (NOT) TYPE, (NOT) UNIQUE, JOIN TO, (NOT) BASE, and SAMPLE.
- A complex Data Filter lets you combine these characteristics with logical operators. It finds rows from one or more tables in which multiple columns have multiple characteristics.

For complete details, refer to *Ascential AuditStage User’s Guide Version 7.0.1* Part No. 00D-001AS701.

Data Filters have three types of output:

- A simple measurement, or a count, of the number of exceptions in the data.
- An exception table to which exceptions are written.

  A Data Filter has the option of outputting distinct (grouped) exceptions, versus all exceptions. When Distinct is selected, the exception table produced when the Data Filter runs will not contain any duplicate occurrences. Therefore, it is dependent on the columns that have been mapped, either to the screen or to an exception table.
- A display of exceptions in a Report window, so that you can see them immediately. If this option is chosen, you can optionally map output columns or specify sorting.

If the output is sent to the screen, there are several ways to navigate through the screen output. The single arrows on the far right of the output allow you to scroll through the screen output one row at a time. The arrows with two lines allow you to scroll through the screen output by several rows. The number of rows is dependent on the size of the window you have open. The arrows with the single line take you to the top and bottom of the screen output.

In addition, the screen output has a search capability. To exercise the search, highlight the column that you want to search and enter one or more characters in the text box. Indicate whether the search is to be case-sensitive. The default is not case sensitive. Click **Search**. IBM WebSphere AuditStage takes you to the first occurrence that matches. If the search does not bring you to the desired row, click **Search** again. IBM WebSphere AuditStage brings you to the next match. If there is no match, IBM WebSphere AuditStage provides a message and leaves you at the bottom of the screen output. Entering a new set of search criteria starts the search from the top.

Note: Exception Table and Screen are not exclusive of Count. That is, they also count the number of exceptions for the Data Filter, but they perform an additional step.

When you run a Data Filter, no matter which of the three types of output it produces, IBM WebSphere AuditStage shows you the number of exceptions found, the total rows of data checked, and the percentage of overall rows that are exceptions. If the results are meaningful, you can choose to store them in the Project, by saving the results in Filter History.

To summarize, Data Filters can find simple data defects by operating on one column and performing one type of check. They can also perform compound checks on multiple columns and can join multiple tables. The rows detected by a

Data Filter can be counted, displayed in a Report window, or written to an exception table.

In the following sections, we describe the creation and execution of data filters that check for a variety of business rule compliances including Referential Integrity check, Uniqueness check, Existence check, Range check, and Complex data filters.

> **Note:** Some of the functions such as referential integrity, uniqueness, and range checks are also available in IBM WebSphere Information Analyzer. We included that information in this section for the sake of completeness. You would most probably use the equivalent facilities in IBM WebSphere Information Analyzer rather than the features in IBM WebSphere AuditStage.

> **Attention:** In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

Referential Integrity check

We check the referential integrity of the ADMRDEPT column in the DEPARMENT table against the DEPTNO column in the same table using the IN REF COL data filter, and write the exceptions to an exception table.

You can access the list of Data Filters for the current Project four ways as follows:

- Scroll through or search the Repository window
- Click the **Data Filters** toolbar button
- Press Ctrl+D
- Choose **Filters** → **Data Filters** from the AuditStage window

The last three actions bring you to the Data Filters directory in the Repository window and open a new Object window. If you are scrolling or searching the Repository window and want to create a new Data Filter, right-click **Data Filters** in the Repository window and choose **New**.

> **Note:** To open an existing Data Filter, double-click the name of the Data Filter in the list in the Repository window. To edit, delete, copy, rename, or run an existing one, highlight its name and right-click.

The following series of steps creates a referential integrity check data filter, executes it, and reviews the exceptions reported:

1. Choose **Filters** → **Data Filters** from the AuditStage window as shown in Figure 1-276 on page 353.
2. In the Untitled Data Filter (define) window, let the Category default to Other, specify the Filter Name (IA_BUSINESS_RULES_ADMRDEPT_RICHECK), select the Source Data column, and click **Expression®** as shown in Figure 1-277 on page 354 to use the Expression Builder for this column.[21]
3. In the Expression Builder window, identify the column in the source table (ADMRDEPT in the BR_DEPARTMENT table in our case) and click **OK** as shown in Figure 1-278 on page 354. The selected fields are shown in the bottom pane.[22]
4. In the Untitled Data Filter (define) window, select **Negative**, and select **NOT IN REF COL** from the Type of Check column. Let the Output field default to Count.[23]

   To view what happens when you enter an invalid value, we entered `ABC` (an invalid column name) in the Reference Data column. Click **Run** to execute the Data Filter created as shown in Figure 1-279 on page 355.
5. Figure 1-280 on page 355 shows the error message `Could not complete Filter` **“**`IA_BUSINESS_RULES_ADMRDEPT_RICHECK`**”.** `Do you want to see more detailed information?`. Click **Yes**.
6. Figure 1-281 on page 356 shows the detailed information about the error. It shows the generated SQL statement and indicates the error is associated with it.
7. Go back and use the Expression Builder for the Reference Data column in the Reference Column Check window as shown in Figure 1-282 on page 356. We build the column DEPTNO in table DEPARTMENT here. Click **OK**.

[21] This drop-down list is used to indicate the classification of a Data Filter. Valid categories are L1 CPL, L1 VAL, or Other. The default value for a new Data Filter is Other. Selecting the wrong category has implications for the L1 Completeness and Validity Report.

[22] An *expression* is a reference to a column in a table. It can be a simple reference to a single column in a table (such as table column), a complex reference to a single column using functions (such as LTRIM(table.column)), or a reference to two or more columns in the same table (such as table.column1 & table.column2). The maximum size of an expression is 255 characters. There are several places in IBM WebSphere AuditStage where you might need to build an expression, including Data Filter source data, Data Filter reference data, and Mapping output columns for Data Filters.

[23] When negative, the Data Filter finds rows for which the source data value does not exist in the specified reference column. When positive, it finds rows for which the source data value exists in the reference column.

8. Click **Run** as shown in Figure 1-283 on page 357 to execute the IA_BUSINESS_RULES_ADMRDEPT_RICHECK data filter just created, and create an output that shows the count of exceptions.

   When a Data Filter runs successfully, the Results dialog box opens, as shown in Figure 1-284 on page 357. This dialog box displays the number of exceptions found by the Data Filter, the total number of rows it operated on, and the exception percentage realized by these last two results. You are given the option to record these results in the Filter History. If you ran the Data Filter in an ad hoc manner and the results are not important to your Project, you should not save them.

9. Figure 1-284 on page 357 shows the results of the data filter execution. It identifies 1 exception out of a total of 14 rows indicating that 1 value in the ADMRDEPT column did not find a match in the DEPTNO column. Click **OK**.

10. Figure 1-285 on page 358 shows the transcript of the data filter job execution. The Transcript window shows the current date and time, the number of exceptions found, and continual updates on the progress of the execution.

    The Transcript window opens automatically when a Data Filter or Filter Set is executed, unless the option for writing to a file has been chosen. To open the Transcript window manually, choose **Window** → **Transcript** from the AuditStage window, or press Ctrl+S. If you have specified a file for transcript information, it is automatically opened and displayed when you select the Transcript window.

11. You can choose to write the exceptions to an exception table. If this option is chosen, you must map output columns. You must also type the name of a new exception table or choose one from the drop-down list.[24]

    Click **Columns** in Figure 1-285 on page 358 to map output columns. In the Map Output Columns window in Figure 1-286 on page 359 identify all the source columns to map in the output. Click **OK**.

12. In the Untitled Data Filter (define) window in Figure 1-287 on page 360, select Exception Table from the Output drop-down list and the name of the exception table (BR_ADMR_RICHECK), and click **Run**.

[24] You use a Mapping output columns dialog box to specify the output mapping for a Data Filter. That is, the columns listed here define rows that will be counted, written to an exception table, or displayed in a Report window when the Data Filter runs. You can use any valid data set expressions as columns, in any order, and assign whatever names you like to them. It is only necessary to map output columns when the form of output for a Data Filter is an exception table. This ensures that all of the output columns correspond to real columns in the exception table. If Count or Screen is chosen, you can skip the step of mapping columns. However, if you have chosen to make your output distinct, it is important to choose only the columns to which you want to apply the distinct (group) feature.

13. If the named exception table does not exist (as was the case here), a message is returned asking whether you want IBM WebSphere AuditStage to create it for you as shown in Figure 1-288 on page 360. Click **Yes**.

14. Figure 1-289 on page 361 shows you the list of columns for inclusion in the exception table. Click **Create**.

15. The results of the job execution are the same as shown earlier in Figure 1-284 on page 357, and repeated here in Figure 1-290 on page 361.

16. Click **Save** as shown in Figure 1-291 on page 362 save the data filter IA_BUSINESS_RULES_ADMRDEPT_RICHECK just created.

17. You can browse the contents of the exception table BR_ADMR_RICHECK as shown in Figure 1-292 on page 362 through Figure 1-294 on page 363.

    a. From the AuditStage window, from the menu bar, click **Tables** → **Browse** as shown in Figure 1-292 on page 362.

    b. Choose the exception table BR_ADMR_RICHECK in the Select a Table to Browse windows and click **OK** as shown in Figure 1-293 on page 363.

    c. Figure 1-294 on page 363 shows the row in the DEPARTMENT table whose foreign key ADMRDEPT violates referential integrity.

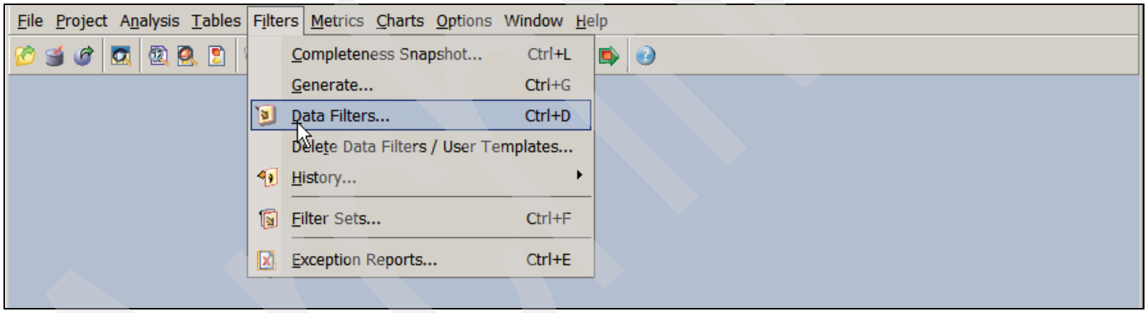
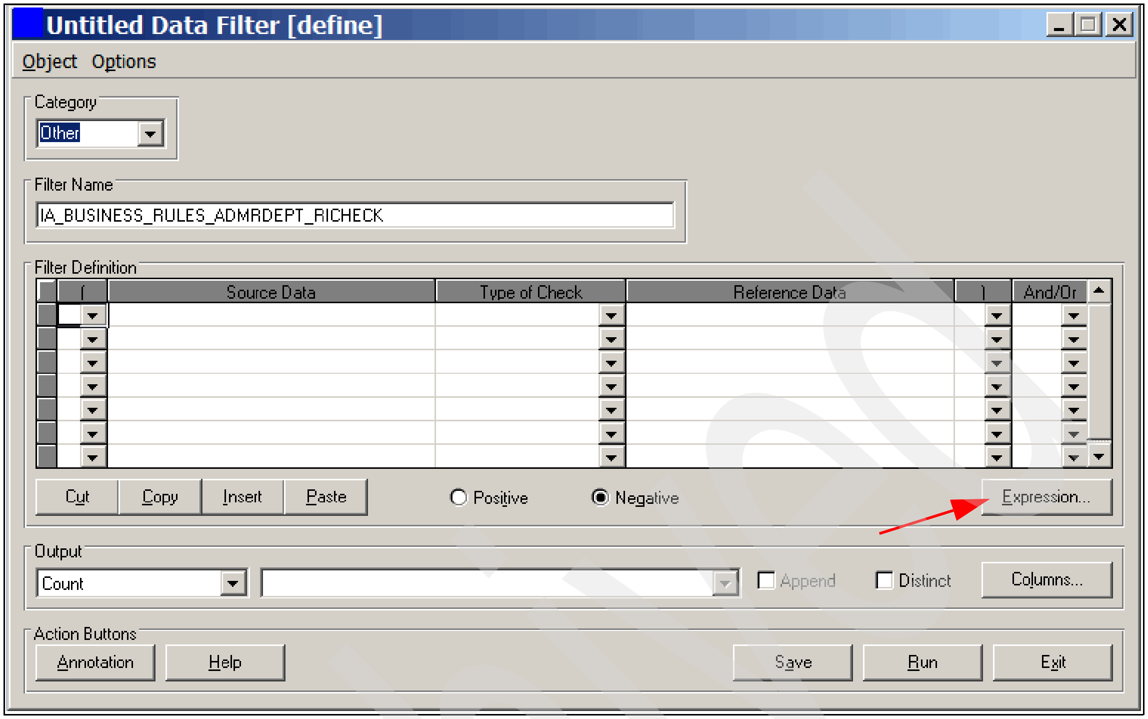


Figure 1-276 Referential integrity check 1/19



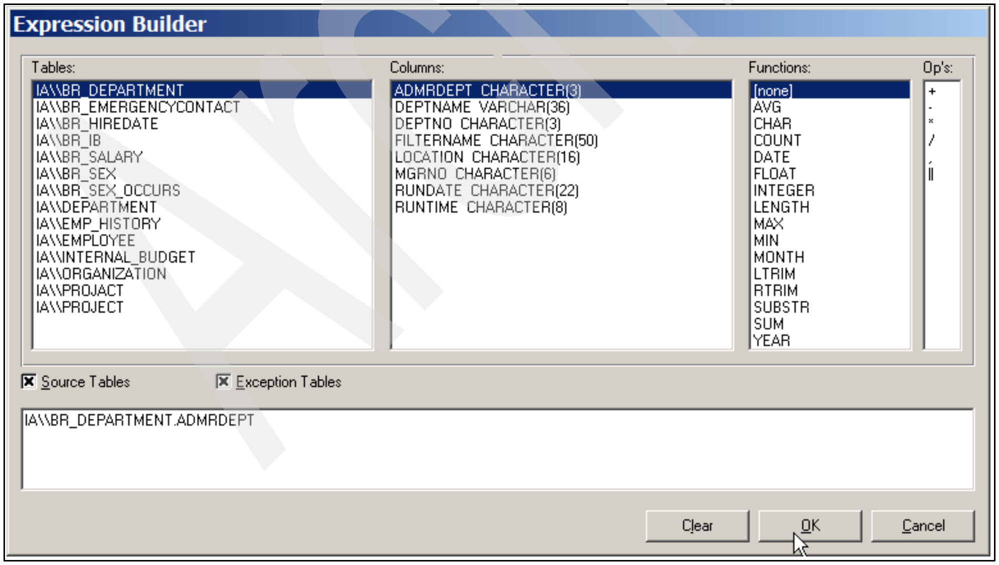
Category: Other

Filter Name: IA_BUSINESS_RULES_ADMRDEPT_RICHECK

○ Positive ● Negative

Output: Count □ Append □ Distinct

Figure 1-277 Referential Integrity check 2/19



Tables:
IA\\BR_DEPARTMENT
IA\\BR_EMERGENCYCONTACT
IA\\BR_HIREDATE
IA\\BR_IB
IA\\BR_SALARY
IA\\BR_SEX
IA\\BR_SEX_OCCURS
IA\\DEPARTMENT
IA\\EMP_HISTORY
IA\\EMPLOYEE
IA\\INTERNAL_BUDGET
IA\\ORGANIZATION
IA\\PROJACT
IA\\PROJECT

Columns:
ADMRDEPT CHARACTER(3)
DEPTNAME VARCHAR(36)
DEPTNO CHARACTER(3)
FILTERNAME CHARACTER(50)
LOCATION CHARACTER(16)
MGRNO CHARACTER(6)
RUNDATE CHARACTER(22)
RUNTIME CHARACTER(8)

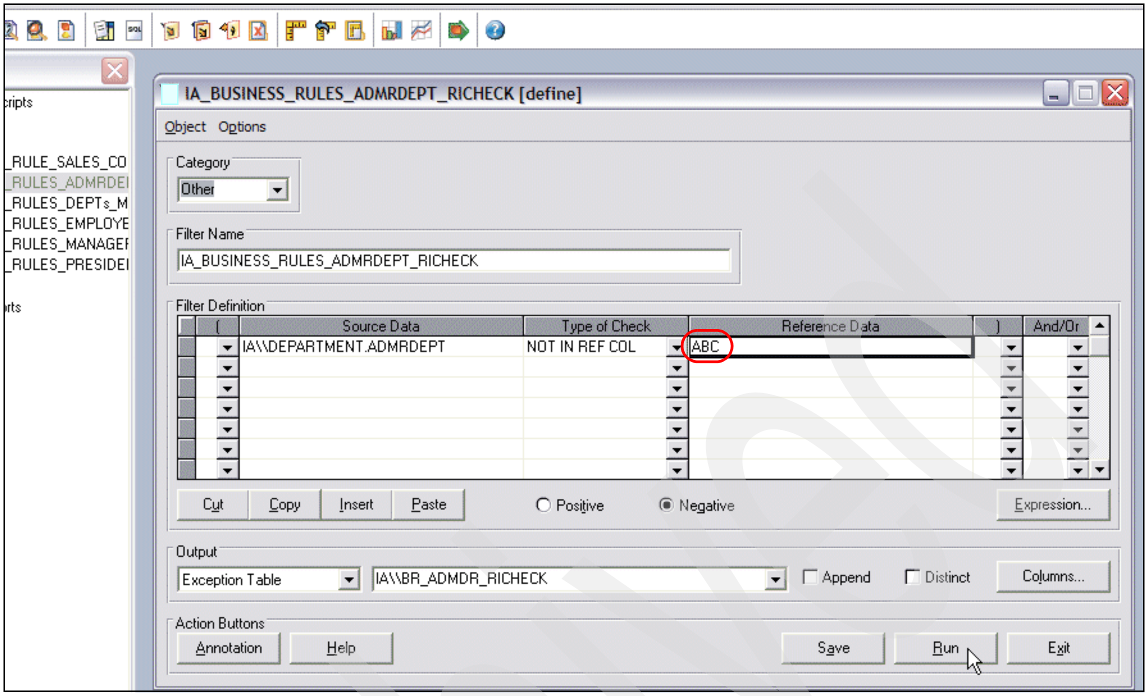
Functions:
[none]
AVG
CHAR
COUNT
DATE
FLOAT
INTEGER
LENGTH
MAX
MIN
MONTH
LTRIM
RTRIM
SUBSTR
SUM
YEAR

Op's:
+
-
*
/
,
||

☒ Source Tables ☒ Exception Tables

IA\\BR_DEPARTMENT.ADMRDEPT

Figure 1-278 Referential integrity check 3/19


IA_BUSINESS_RULES_ADMRDEPT_RICHECK [define]

Category: Other

Filter Name: IA_BUSINESS_RULES_ADMRDEPT_RICHECK

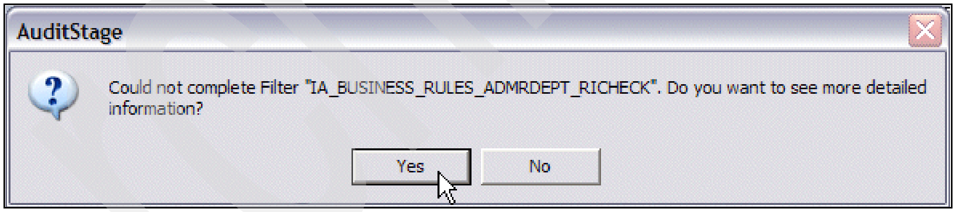
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\DEPARTMENT.ADMRDEPT	NOT IN REF COL	ABC		

○ Positive ● Negative

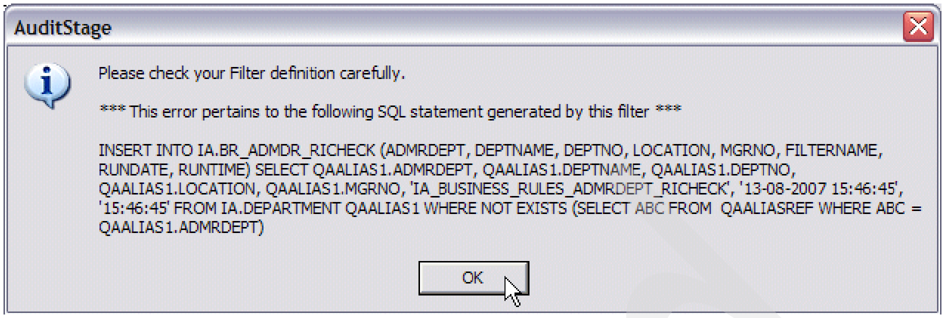
Output: Exception Table | IA\\BR_ADMDR_RICHECK | □ Append | □ Distinct

Figure 1-279 Referential integrity check 4/19


AuditStage

Could not complete Filter "IA_BUSINESS_RULES_ADMRDEPT_RICHECK". Do you want to see more detailed information?

Figure 1-280 Referential integrity check 5/19

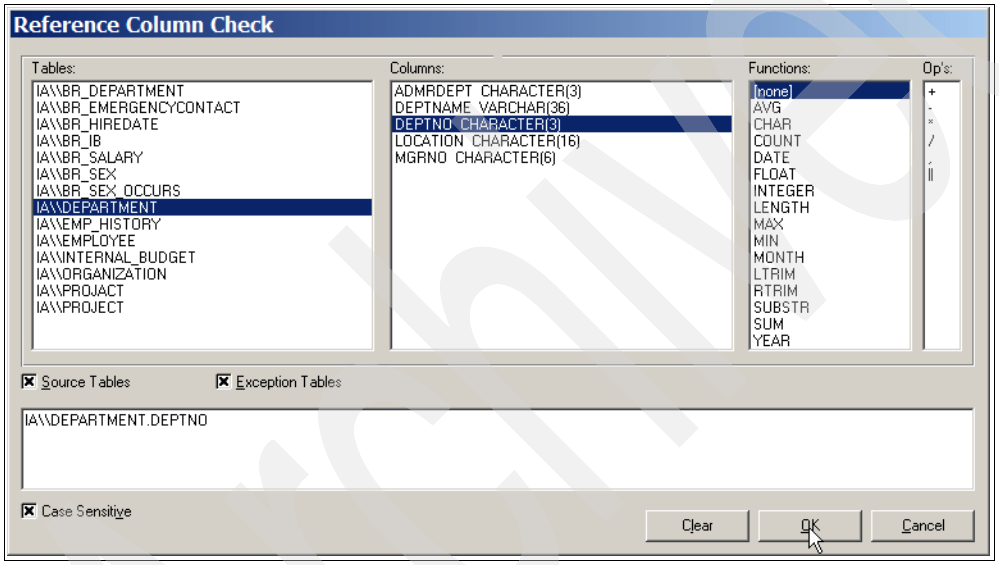


Please check your Filter definition carefully.

*** This error pertains to the following SQL statement generated by this filter ***

```
INSERT INTO IA.BR_ADMDR_RICHECK (ADMRDEPT, DEPTNAME, DEPTNO, LOCATION, MGRNO, FILTERNAME,
RUNDATE, RUNTIME) SELECT QAALIAS1.ADMRDEPT, QAALIAS1.DEPTNAME, QAALIAS1.DEPTNO,
QAALIAS1.LOCATION, QAALIAS1.MGRNO, 'IA_BUSINESS_RULES_ADMRDEPT_RICHECK', '13-08-2007 15:46:45',
'15:46:45' FROM IA.DEPARTMENT QAALIAS1 WHERE NOT EXISTS (SELECT ABC FROM  QAALIASREF WHERE ABC =
QAALIAS1.ADMRDEPT)
```

Figure 1-281   Referential integrity check 6/19



Tables:

IA\\BR_DEPARTMENT
IA\\BR_EMERGENCYCONTACT
IA\\BR_HIREDATE
IA\\BR_IB
IA\\BR_SALARY
IA\\BR_SEX
IA\\BR_SEX_OCCURS
IA\\DEPARTMENT
IA\\EMP_HISTORY
IA\\EMPLOYEE
IA\\INTERNAL_BUDGET
IA\\ORGANIZATION
IA\\PROJACT
IA\\PROJECT

Columns:

ADMRDEPT CHARACTER(3)
DEPTNAME VARCHAR(36)
DEPTNO CHARACTER(3)
LOCATION CHARACTER(16)
MGRNO CHARACTER(6)

Functions:

[none]
AVG
CHAR
COUNT
DATE
FLOAT
INTEGER
LENGTH
MAX
MIN
MONTH
LTRIM
RTRIM
SUBSTR
SUM
YEAR

Op's:

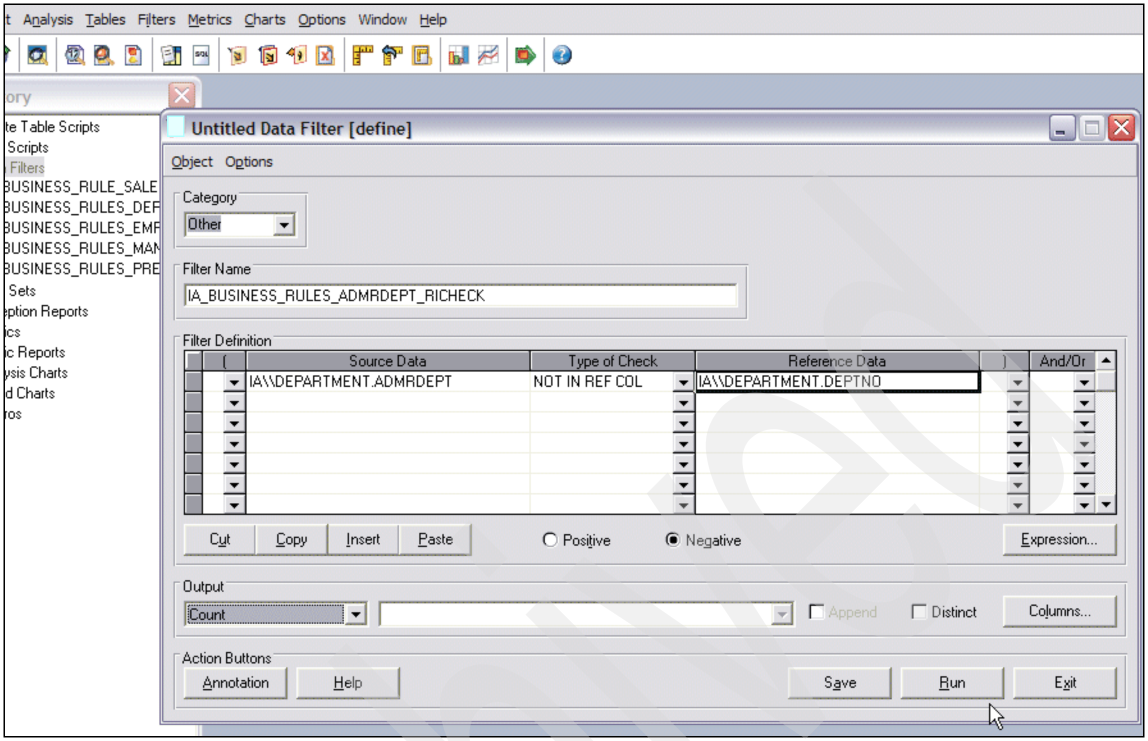
+
-
*
/
,
||

☒ Source Tables   ☒ Exception Tables

IA\\DEPARTMENT.DEPTNO

☒ Case Sensitive

Figure 1-282   Referential integrity check 7/19



Untitled Data Filter [define]

Category: Other

Filter Name: IA_BUSINESS_RULES_ADMRDEPT_RICHECK

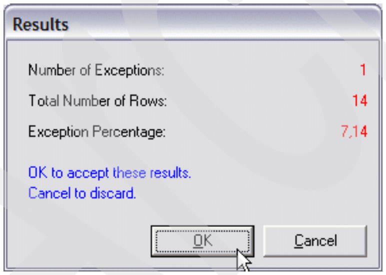
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\DEPARTMENT.ADMRDEPT	NOT IN REF COL	IA\\DEPARTMENT.DEPTNO		

○ Positive ● Negative

Output: Count □ Append □ Distinct

Figure 1-283 Referential integrity check 8/19

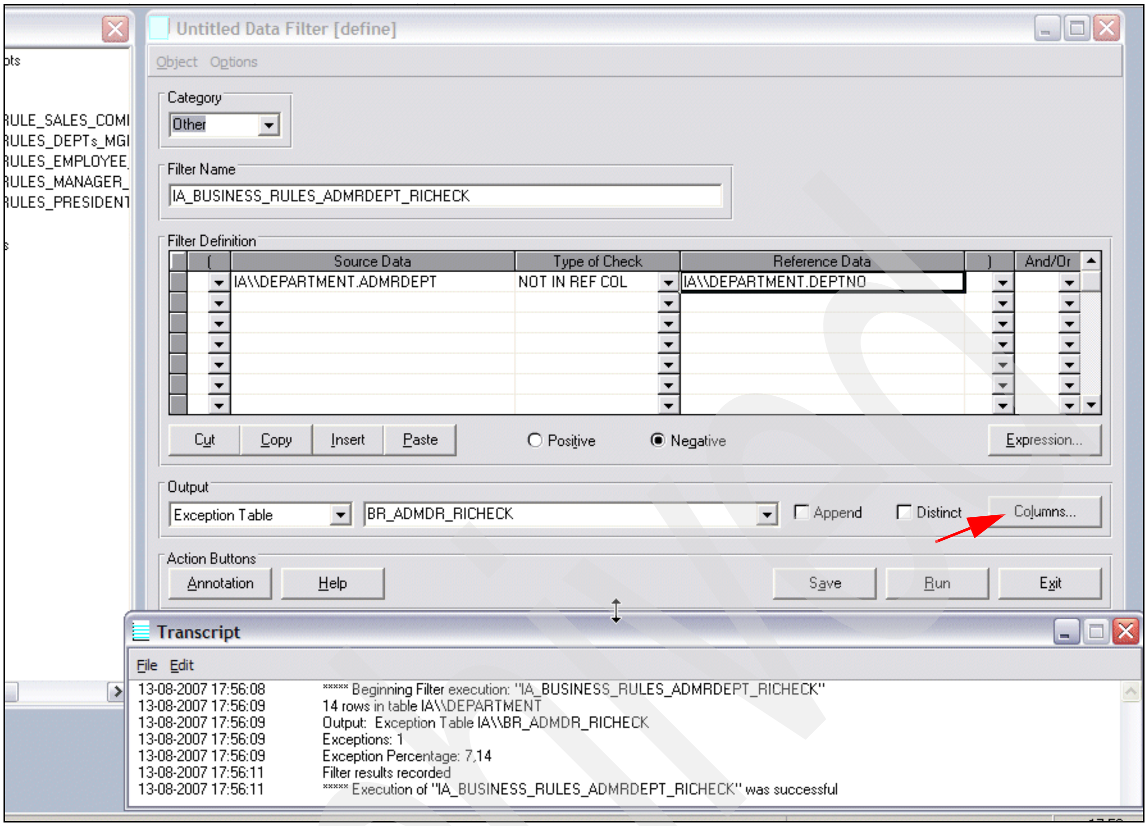


Results

Number of Exceptions:	1
Total Number of Rows:	14
Exception Percentage:	7,14

OK to accept these results.
Cancel to discard.

Figure 1-284 Referential integrity check 9/19



Category: Other

Filter Name: IA_BUSINESS_RULES_ADMRDEPT_RICHECK

Filter Definition

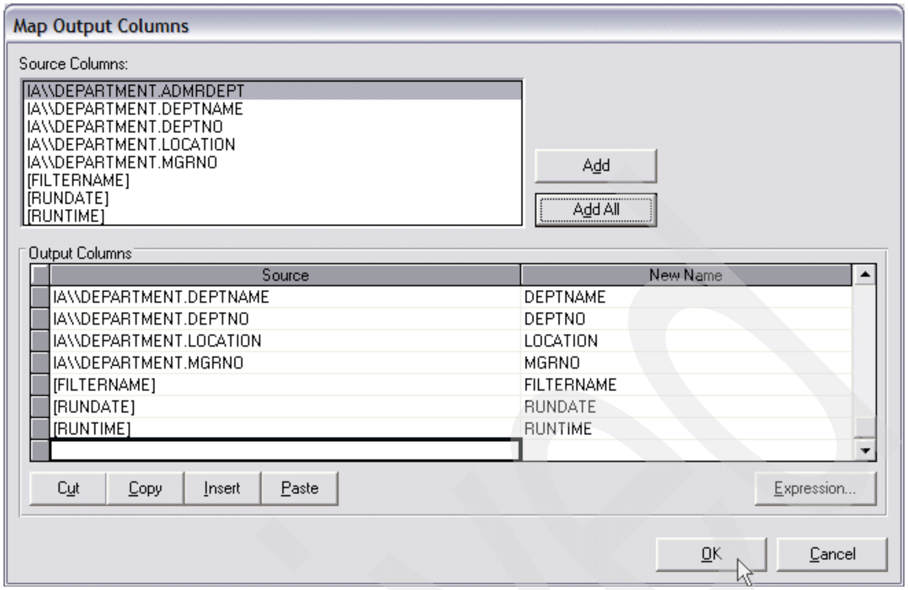
(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\DEPARTMENT.ADMRDEPT	NOT IN REF COL	IA\\DEPARTMENT.DEPTNO		

○ Positive ● Negative

Output: Exception Table BR_ADMDR_RICHECK □ Append □ Distinct

```
13-08-2007 17:56:08    ***** Beginning Filter execution: "IA_BUSINESS_RULES_ADMRDEPT_RICHECK"
13-08-2007 17:56:09    14 rows in table IA\\DEPARTMENT
13-08-2007 17:56:09    Output:  Exception Table IA\\BR_ADMDR_RICHECK
13-08-2007 17:56:09    Exceptions: 1
13-08-2007 17:56:09    Exception Percentage: 7,14
13-08-2007 17:56:11    Filter results recorded
13-08-2007 17:56:11    ***** Execution of "IA_BUSINESS_RULES_ADMRDEPT_RICHECK" was successful
```

Figure 1-285 Referential integrity check 10/19



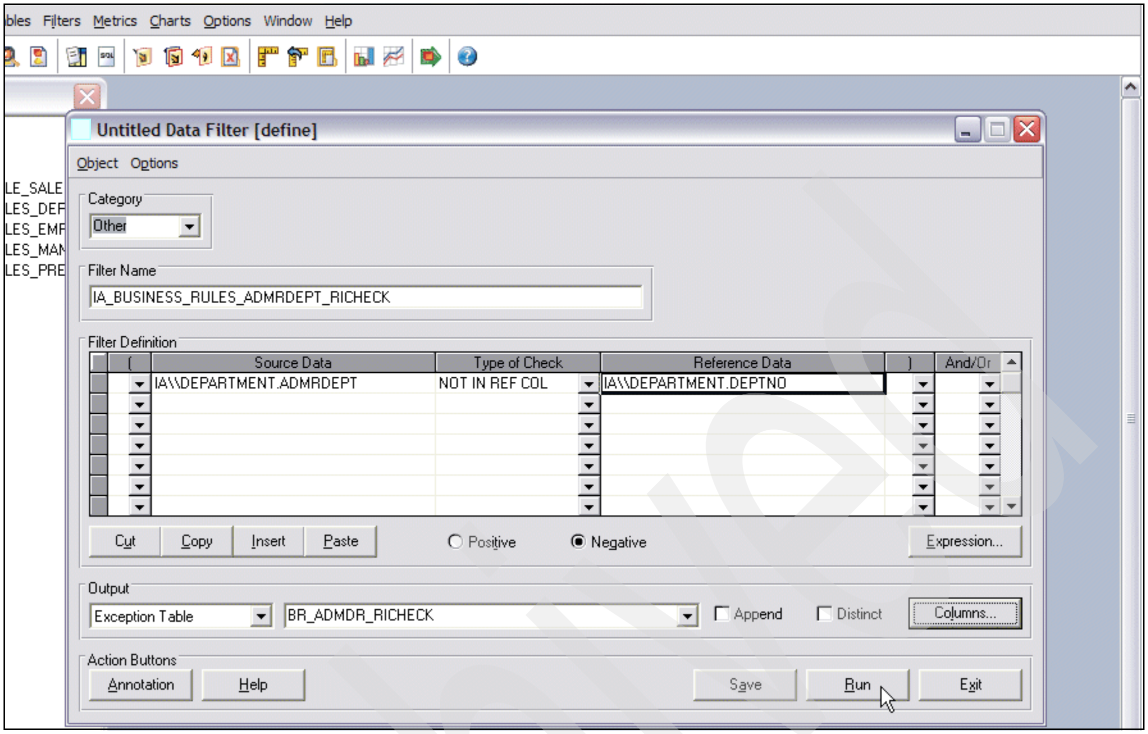
Source Columns:

- IA\\DEPARTMENT.ADMRDEPT
- IA\\DEPARTMENT.DEPTNAME
- IA\\DEPARTMENT.DEPTNO
- IA\\DEPARTMENT.LOCATION
- IA\\DEPARTMENT.MGRNO
- [FILTERNAME]
- [RUNDATE]
- [RUNTIME]

Output Columns

Source	New Name
IA\\DEPARTMENT.DEPTNAME	DEPTNAME
IA\\DEPARTMENT.DEPTNO	DEPTNO
IA\\DEPARTMENT.LOCATION	LOCATION
IA\\DEPARTMENT.MGRNO	MGRNO
[FILTERNAME]	FILTERNAME
[RUNDATE]	RUNDATE
[RUNTIME]	RUNTIME

Figure 1-286 Referential integrity check 11/19


bles Filters Metrics Charts Options Window Help
Untitled Data Filter [define]
Object Options

Category: Other

Filter Name: IA_BUSINESS_RULES_ADMRDEPT_RICHECK

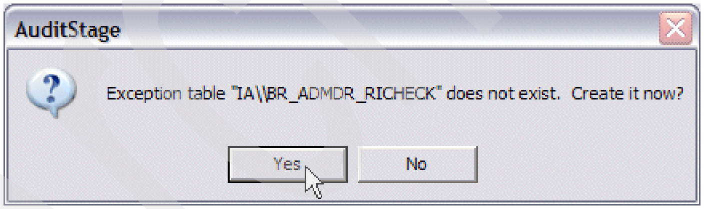
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\DEPARTMENT.ADMRDEPT	NOT IN REF COL	IA\\DEPARTMENT.DEPTNO		

○ Positive ● Negative

Output: Exception Table BR_ADMDR_RICHECK □ Append □ Distinct

Figure 1-287 Referential integrity check 12/19



Exception table "IA\\BR_ADMDR_RICHECK" does not exist. Create it now?

Figure 1-288 Referential integrity check 13/19

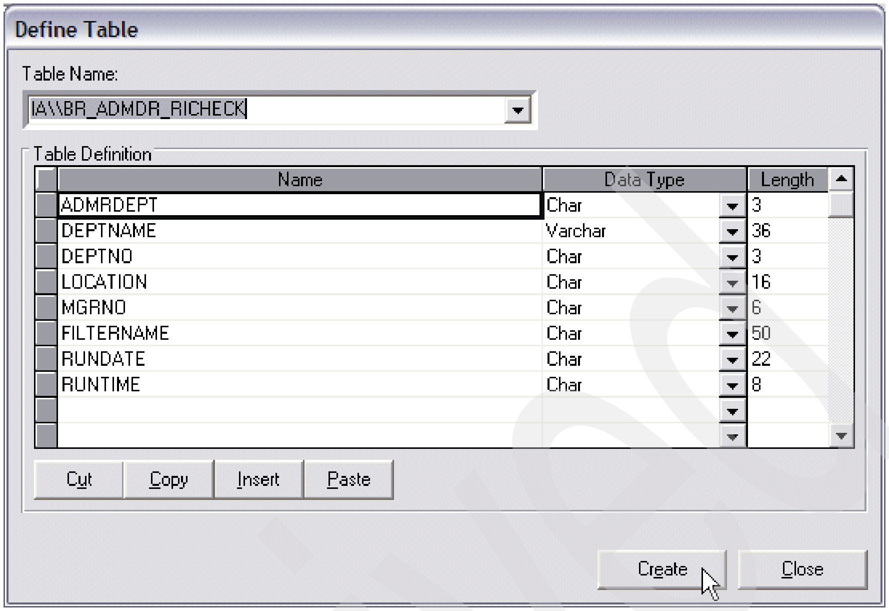
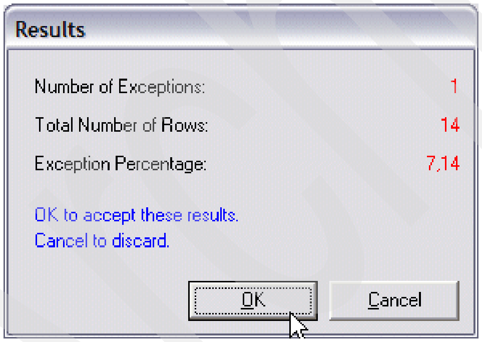


Table Name: IA\\BR_ADMDR_RICHECK

Table Definition

Name	Data Type	Length
ADMRDEPT	Char	3
DEPTNAME	Varchar	36
DEPTNO	Char	3
LOCATION	Char	16
MGRNO	Char	6
FILTERNAME	Char	50
RUNDATE	Char	22
RUNTIME	Char	8

Figure 1-289 Referential integrity check 14/19



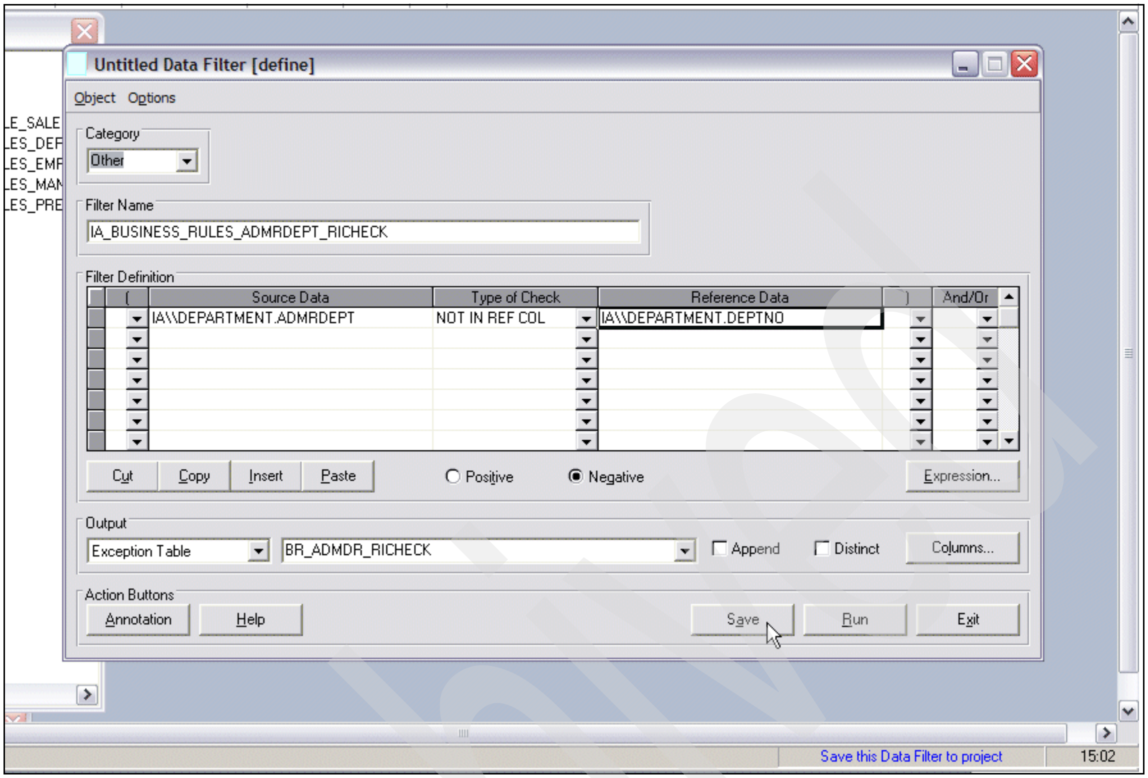
Number of Exceptions: 1

Total Number of Rows: 14

Exception Percentage: 7,14

OK to accept these results.
Cancel to discard.

Figure 1-290 Referential integrity check 15/19



Untitled Data Filter [define]

Category: Other

Filter Name: IA_BUSINESS_RULES_ADMRDEPT_RICHECK

Filter Definition:

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\DEPARTMENT.ADMRDEPT	NOT IN REF COL	IA\\DEPARTMENT.DEPTNO		

○ Positive ● Negative

Output: Exception Table BR_ADMDR_RICHECK □ Append □ Distinct

Figure 1-291 Referential integrity check 16/19

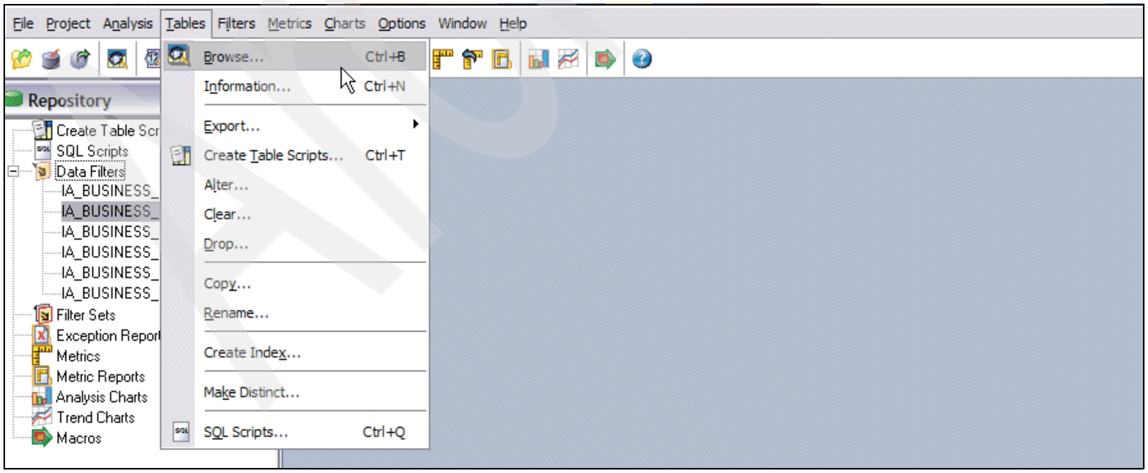
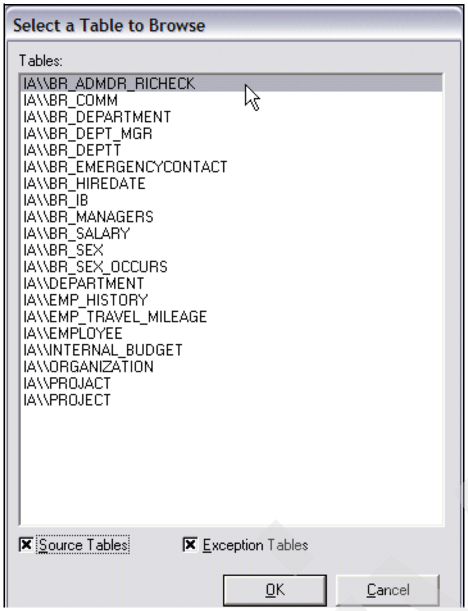


Figure 1-292 Referential integrity check 17/19

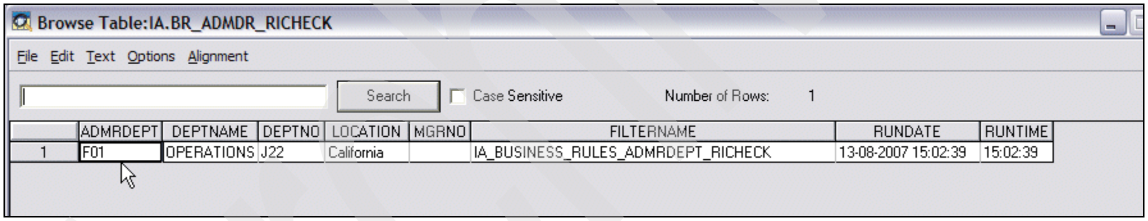


Tables:

IA\\BR_ADMDR_RICHECK
IA\\BR_COMM
IA\\BR_DEPARTMENT
IA\\BR_DEPT_MGR
IA\\BR_DEPTT
IA\\BR_EMERGENCYCONTACT
IA\\BR_HIREDATE
IA\\BR_IB
IA\\BR_MANAGERS
IA\\BR_SALARY
IA\\BR_SEX
IA\\BR_SEX_OCCURS
IA\\DEPARTMENT
IA\\EMP_HISTORY
IA\\EMP_TRAVEL_MILEAGE
IA\\EMPLOYEE
IA\\INTERNAL_BUDGET
IA\\ORGANIZATION
IA\\PROJACT
IA\\PROJECT

☒ Source Tables ☒ Exception Tables

Figure 1-293 Referential integrity check 18/19


Browse Table:IA.BR_ADMDR_RICHECK
File Edit Text Options Alignment
Search

☐ Case Sensitive

Number of Rows: 1

	ADMRDEPT	DEPTNAME	DEPTNO	LOCATION	MGRNO	FILTERNAME	RUNDATE	RUNTIME
1	F01	OPERATIONS	J22	California		IA_BUSINESS_RULES_ADMRDEPT_RICHECK	13-08-2007 15:02:39	15:02:39

Figure 1-294 Referential integrity check 19/19

Uniqueness check

We check the uniqueness of the DEPTNAME column in the DEPARTMENT table. This is not a primary key, nor does it have a unique index defined on it. We write any exceptions to an exception table.

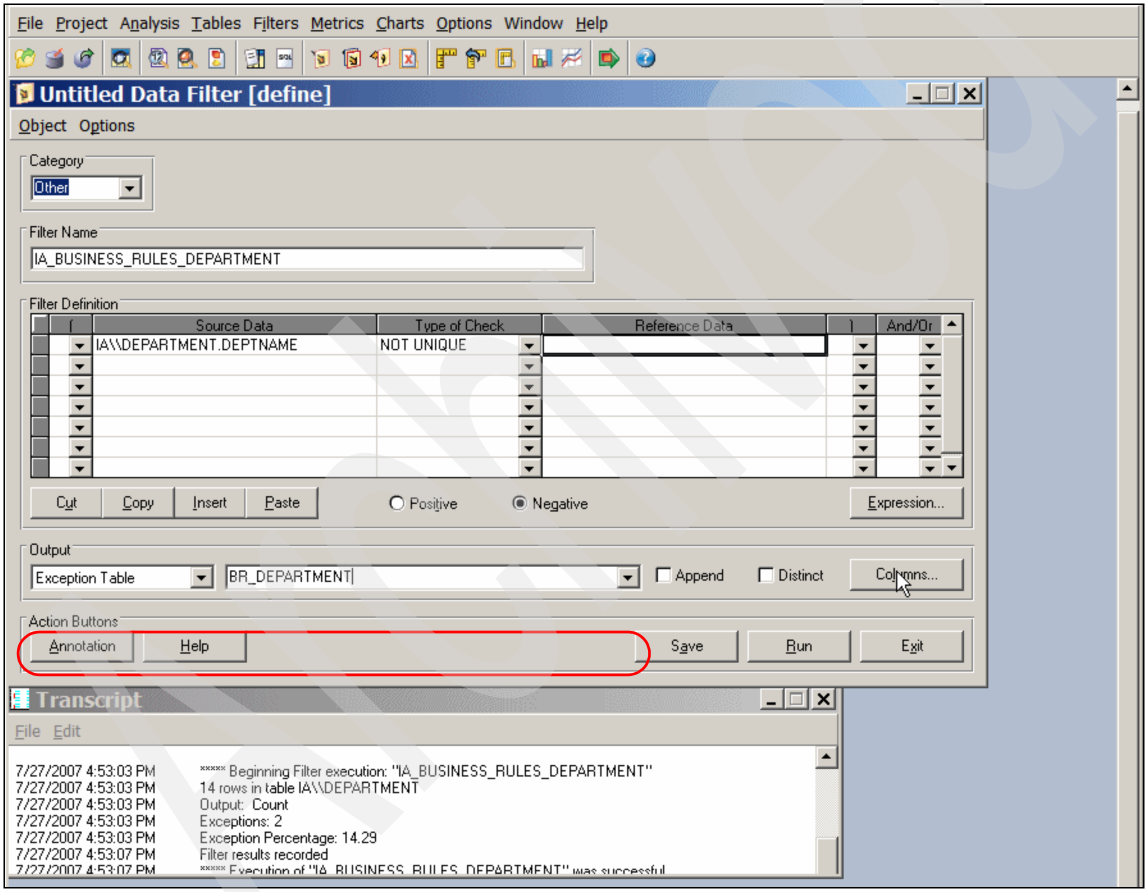
We show only the main panels of this business rule, because the rest are similar to those described in “Referential Integrity check” on page 350.

The following series of steps creates a uniqueness check data filter, executes it, and reviews the exceptions reported:

1. In the Untitled Data Filter (define) window, select **Negative**, and select **NOT UNIQUE** from the Type of Check column.[25] Select **Exception Table** from the Output drop-down list and the name of the exception table

(BR_DEPARTMENT), and click **Run**. The Transcript window shows the successful execution of this job (Figure 1-295 on page 364).

2. Figure 1-296 on page 365 shows the results of the data filter execution. It identifies two exceptions out of a total of 14 rows indicating that there are two duplicate values in the DEPTNAME column. These exceptions get written to the exception table BR_DEPARTMENT. Click **OK**.
3. Figure 1-298 on page 365 and Figure 1-299 on page 366 show the content of the exception table BR_DEPARTMENT that includes rows that have non-unique values in the DEPTNAME column.



Category: Other

Filter Name: IA_BUSINESS_RULES_DEPARTMENT

Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\DEPARTMENT.DEPTNAME	NOT UNIQUE			

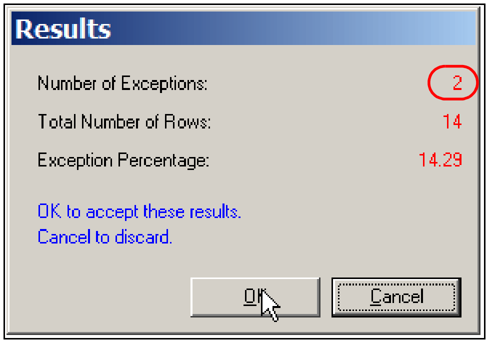
○ Positive ● Negative

Output: Exception Table | BR_DEPARTMENT | □ Append □ Distinct

```
7/27/2007 4:53:03 PM     ***** Beginning Filter execution: "IA_BUSINESS_RULES_DEPARTMENT"
7/27/2007 4:53:03 PM     14 rows in table IA\\DEPARTMENT
7/27/2007 4:53:03 PM     Output:  Count
7/27/2007 4:53:03 PM     Exceptions: 2
7/27/2007 4:53:03 PM     Exception Percentage: 14.29
7/27/2007 4:53:07 PM     Filter results recorded
7/27/2007 4:53:07 PM     ***** Execution of "IA_BUSINESS_RULES_DEPARTMENT" was successful
```

Figure 1-295 Uniqueness check 1/4

[25] When negative, the Data Filter finds rows for which the source data value does not exist in the specified reference column. When positive, it finds rows for which the source data value exists in the reference column.



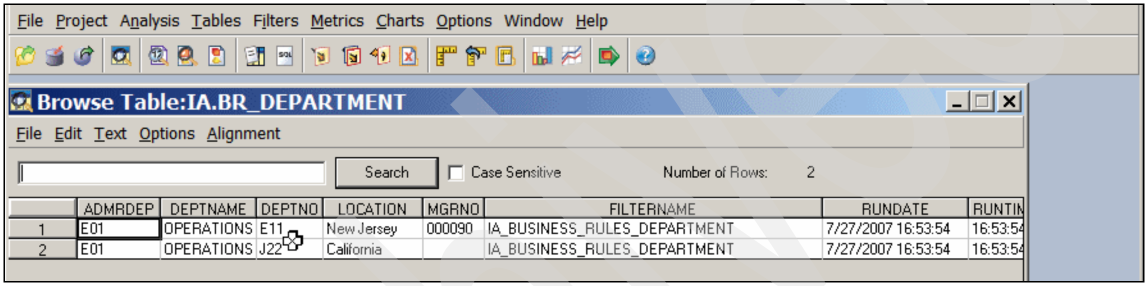
Number of Exceptions: 2

Total Number of Rows: 14

Exception Percentage: 14.29

OK to accept these results.
Cancel to discard.

Figure 1-296 Uniqueness check 2/4



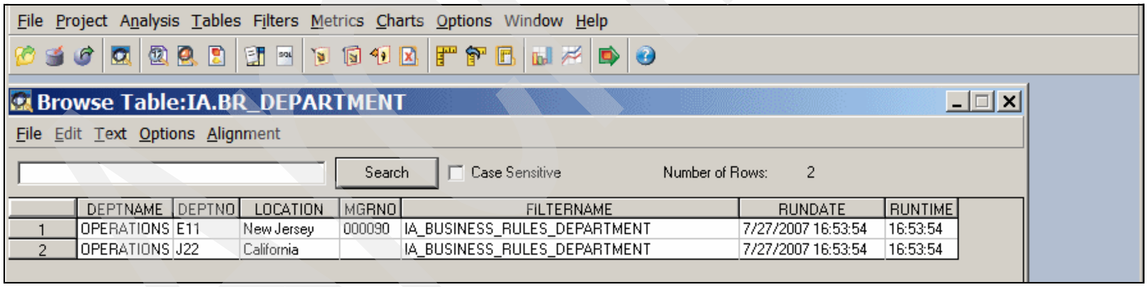
File Project Analysis Tables Filters Metrics Charts Options Window Help
Browse Table:IA.BR_DEPARTMENT
File Edit Text Options Alignment
Search
Case Sensitive

□ Case Sensitive

Number of Rows: 2

	ADMRDEP	DEPTNAME	DEPTNO	LOCATION	MGRNO	FILTERNAME	RUNDATE	RUNTIN
1	E01	OPERATIONS	E11	New Jersey	000090	IA_BUSINESS_RULES_DEPARTMENT	7/27/2007 16:53:54	16:53:54
2	E01	OPERATIONS	J22	California		IA_BUSINESS_RULES_DEPARTMENT	7/27/2007 16:53:54	16:53:54

Figure 1-297 Uniqueness check 3/4



File Project Analysis Tables Filters Metrics Charts Options Window Help
Browse Table:IA.BR_DEPARTMENT
File Edit Text Options Alignment
Search

□ Case Sensitive

Number of Rows: 2

	DEPTNAME	DEPTNO	LOCATION	MGRNO	FILTERNAME	RUNDATE	RUNTIME
1	OPERATIONS	E11	New Jersey	000090	IA_BUSINESS_RULES_DEPARTMENT	7/27/2007 16:53:54	16:53:54
2	OPERATIONS	J22	California		IA_BUSINESS_RULES_DEPARTMENT	7/27/2007 16:53:54	16:53:54

Figure 1-298 Uniqueness check 4/4

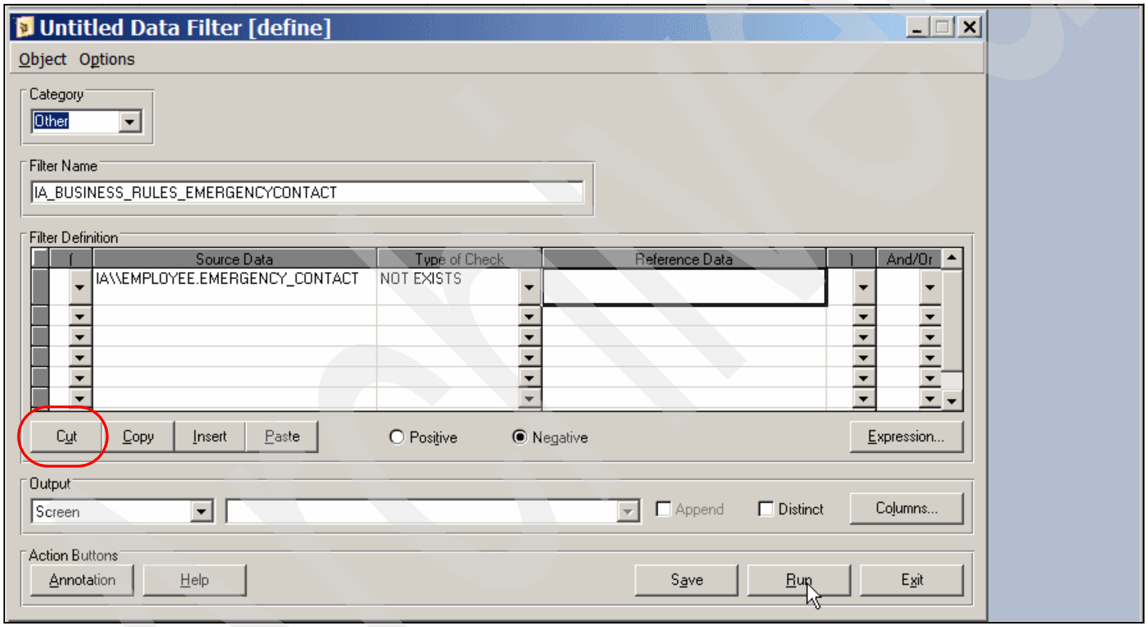
Existence check

We check for the existence of an emergency contact in the EMPLOYEE table. We return any exceptions to the screen.

We show only the main panels of this business rule, because the rest are similar to those described in “Referential Integrity check” on page 350.

The following series of steps creates an existence check data filter, executes it, and reviews the exceptions reported:

1. In the Untitled Data Filter (define) window, check the Negative[26] radio button, and select NOT EXISTS from the Type of Check column. Select Screen from the Output drop down list, and click **Run**. The Transcript window shows the successful execution of this job. This is shown in Figure 1-299.
2. Figure 1-300 on page 367 shows the results of the data filter execution. It identifies 13 exceptions out of a total of 48 rows indicating that there are 13 employees that have not provided emergency contact information. These exceptions are displayed on the screen. Click **OK**.
3. Figure 1-299 on page 366 shows on the screen a partial list of the rows in the EMPLOYEE table that do not have emergency contact information.



Category: Other

Filter Name: IA_BUSINESS_RULES_EMERGENCYCONTACT

Filter Definition

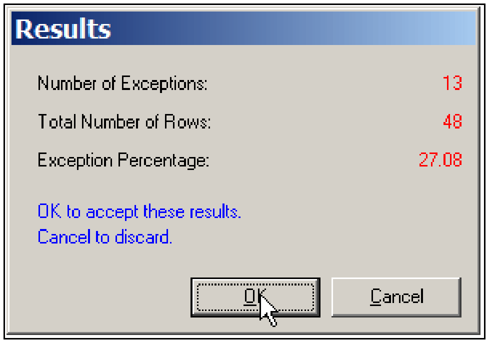
(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.EMERGENCY_CONTACT	NOT EXISTS			

○ Positive ● Negative

Output: Screen □ Append □ Distinct

Figure 1-299 Existence check 1/3

[26] When negative, the Data Filter finds rows for which the source data value does not exist in the specified reference column. When positive, it finds rows for which the source data value exists in the reference column.



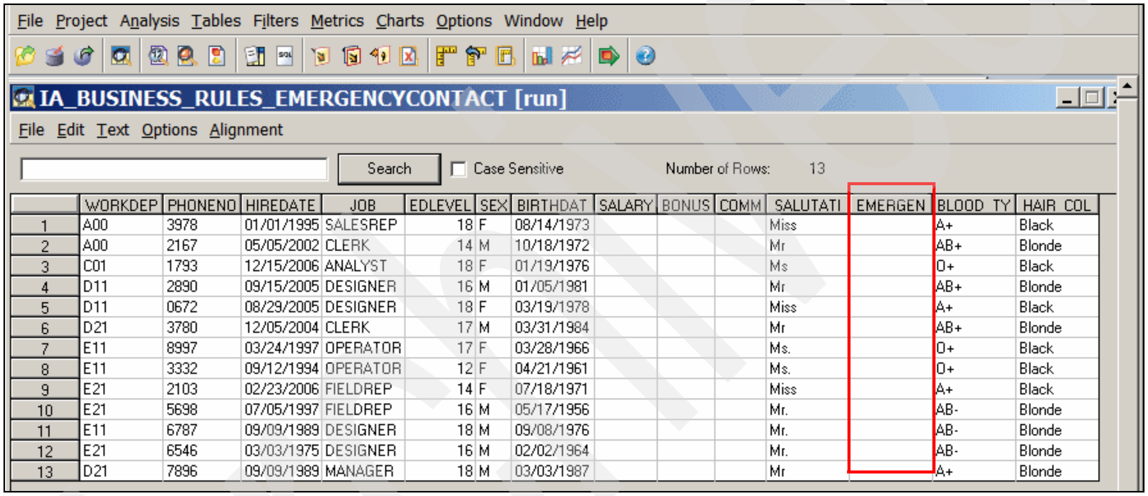
Number of Exceptions: 13

Total Number of Rows: 48

Exception Percentage: 27.08

OK to accept these results.
Cancel to discard.

Figure 1-300 Existence check 2/3



File Project Analysis Tables Filters Metrics Charts Options Window Help
IA_BUSINESS_RULES_EMERGENCYCONTACT [run]
File Edit Text Options Alignment
Search
□ Case Sensitive

Number of Rows: 13

	WORKDEP	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDAT	SALARY	BONUS	COMM	SALUTATI	EMERGEN	BLOOD TY	HAIR COL
1	A00	3978	01/01/1995	SALESREP	18	F	08/14/1973				Miss		A+	Black
2	A00	2167	05/05/2002	CLERK	14	M	10/18/1972				Mr		AB+	Blonde
3	C01	1793	12/15/2006	ANALYST	18	F	01/19/1976				Ms		O+	Black
4	D11	2890	09/15/2005	DESIGNER	16	M	01/05/1981				Mr		AB+	Blonde
5	D11	0672	08/29/2005	DESIGNER	18	F	03/19/1978				Miss		A+	Black
6	D21	3780	12/05/2004	CLERK	17	M	03/31/1984				Mr		AB+	Blonde
7	E11	8997	03/24/1997	OPERATOR	17	F	03/28/1966				Ms.		O+	Black
8	E11	3332	09/12/1994	OPERATOR	12	F	04/21/1961				Ms.		O+	Black
9	E21	2103	02/23/2006	FIELDREP	14	F	07/18/1971				Miss		A+	Black
10	E21	5698	07/05/1997	FIELDREP	16	M	05/17/1956				Mr.		AB-	Blonde
11	E11	6787	09/09/1989	DESIGNER	18	M	09/08/1976				Mr.		AB-	Blonde
12	E21	6546	03/03/1975	DESIGNER	16	M	02/02/1964				Mr.		AB-	Blonde
13	D21	7896	09/09/1989	MANAGER	18	M	03/03/1987				Mr		A+	Blonde

Figure 1-301 Existence check 3/3

Range check

We check that the salary of an employee is between 50,000 and 90,000 using the IN RANGE data filter check. We write any exceptions to an exception table.

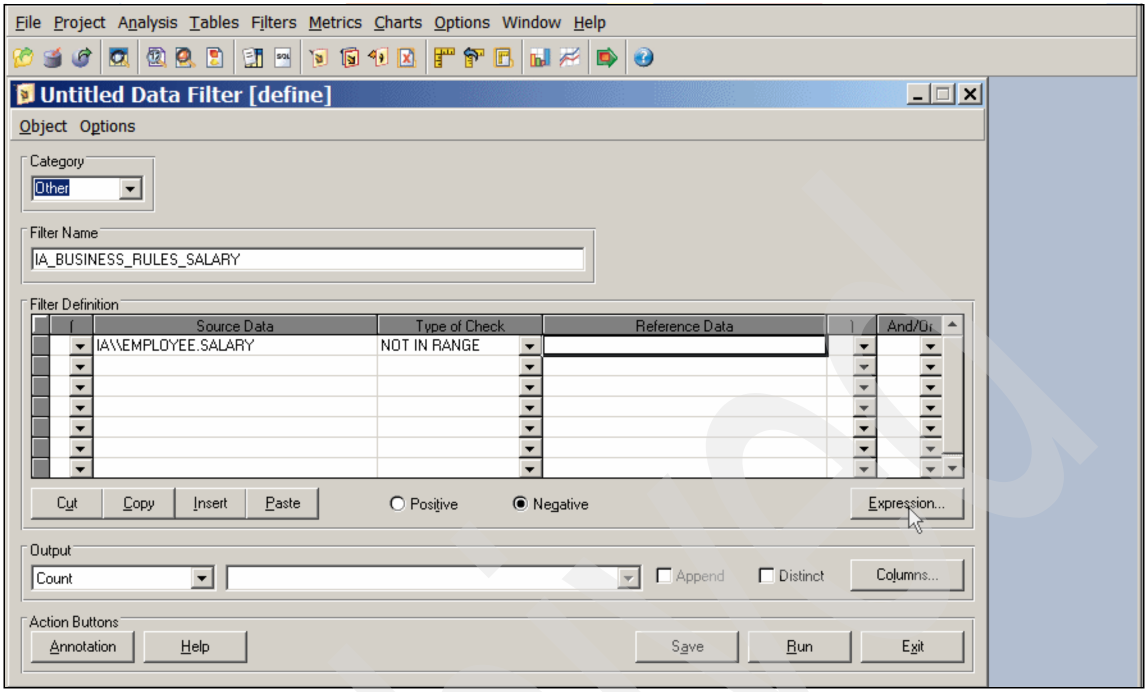
We show only the main panels of this business rule, because the rest are similar to those described in “Referential Integrity check” on page 350.

The following series of steps creates a uniqueness check data filter, executes it, and counts the exceptions reported:

1. In the Untitled Data Filter (define) window, select **Negative**, and select **NOT IN RANGE** from the Type of Check column.[27] Select the row under the column Reference Data and click **Expression** as shown in Figure 1-302 on page 369.
2. In the Range Check window, specify the Minimum (50000) and Maximum (90000) values and click OK as shown in Figure 1-303 on page 369.
3. In the Untitled Data Filter (define) window, select **Exception Table** from the Output drop-down list and the name of the exception table (BR_SALARY), and click **Run** as shown in Figure 1-304 on page 370.
4. Figure 1-305 on page 370 shows the results of the data filter execution. It identifies 26 exceptions out of a total of 48 rows indicating that there are 26 employees who have a salary outside the 50000 to 90000 range. These exceptions get written to the exception table BR_SALARY. Click **OK**.
5. Figure 1-306 on page 371 shows the Transcript window of the successful execution of this job.

   We do not show the contents of the exception table here.

[27] When negative, the Data Filter finds rows for which the source data value does not exist in the specified reference column. When positive, it finds rows for which the source data value exists in the reference column.


File Project Analysis Tables Filters Metrics Charts Options Window Help
Untitled Data Filter [define]
Object Options
Cut
Copy
Insert
Paste
Expression...
Columns...
Annotation
Help
Save
Run
Exit

Category: Other

Filter Name: IA_BUSINESS_RULES_SALARY

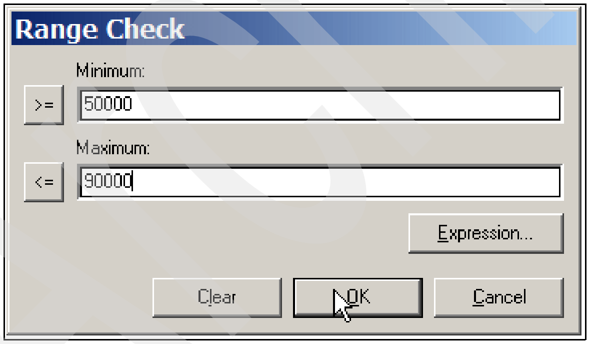
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.SALARY	NOT IN RANGE			

○ Positive ● Negative

Output: Count □ Append □ Distinct

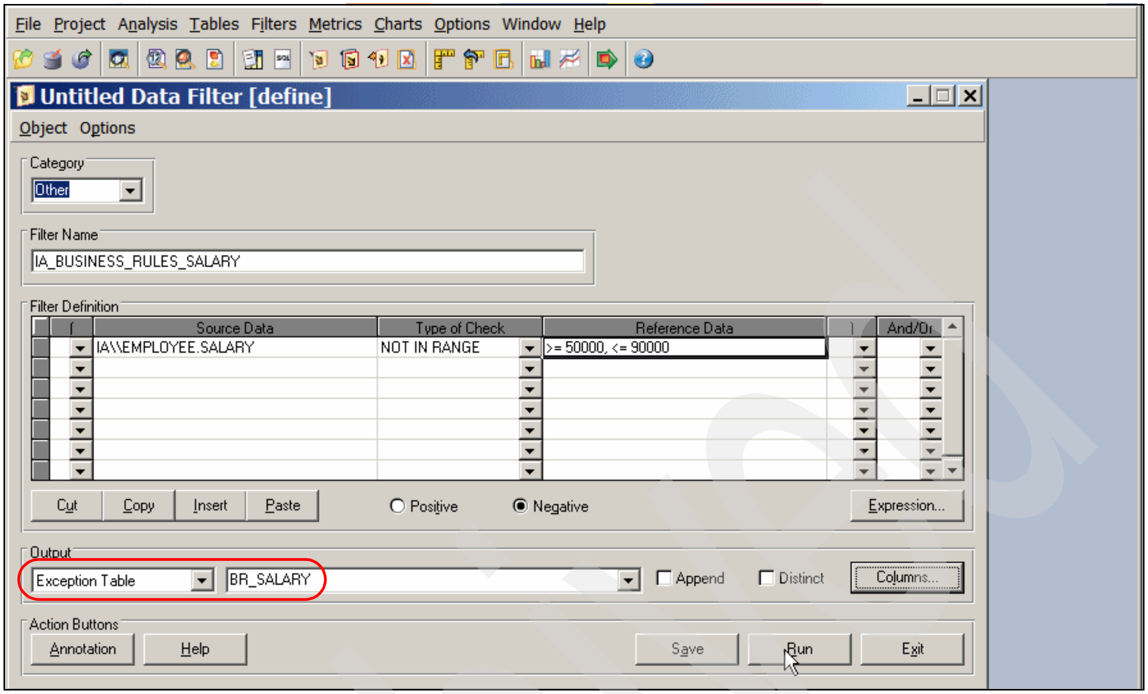
Figure 1-302 Range check 1/5



Minimum: 50000

Maximum: 90000

Figure 1-303 Range check 2/5



Untitled Data Filter [define]

Category: Other

Filter Name: IA_BUSINESS_RULES_SALARY

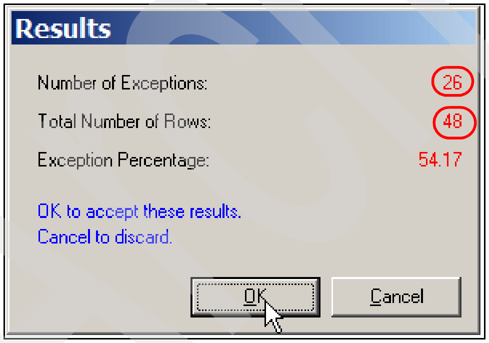
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.SALARY	NOT IN RANGE	>= 50000, <= 90000		

○ Positive ● Negative

Output: Exception Table BR_SALARY □ Append □ Distinct

Figure 1-304 Range check 3/5



Results

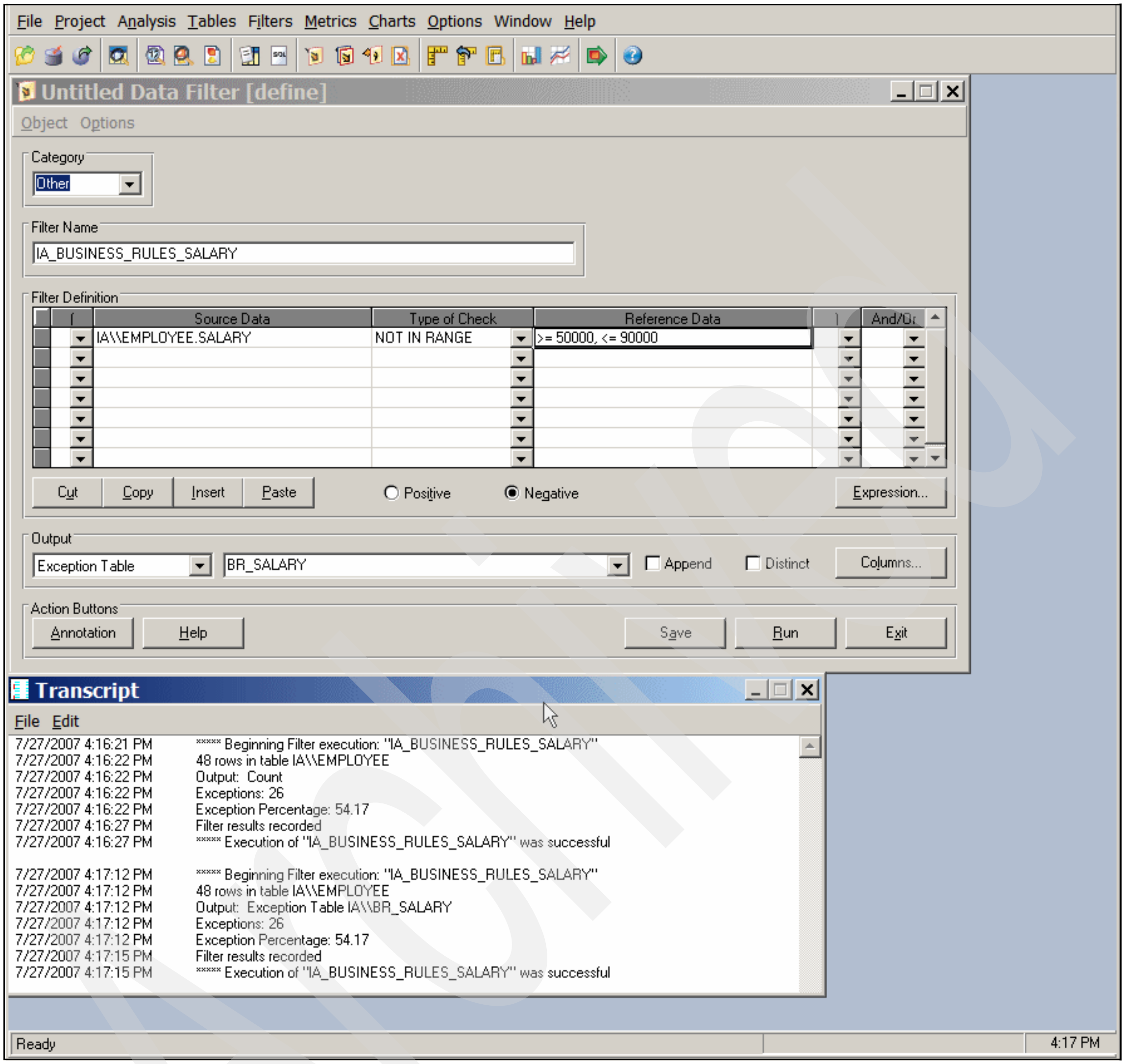
Number of Exceptions: 26

Total Number of Rows: 48

Exception Percentage: 54.17

OK to accept these results.
Cancel to discard.

Figure 1-305 Range check 4/5


File Project Analysis Tables Filters Metrics Charts Options Window Help
Untitled Data Filter [define]
Object Options
Cut Copy Insert Paste
Expression...
Columns...
Action Buttons
Annotation Help Save Run Exit
Transcript
File Edit

Category: Other

Filter Name: IA_BUSINESS_RULES_SALARY

Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.SALARY	NOT IN RANGE	>= 50000, <= 90000		

○ Positive ● Negative

Output: Exception Table BR_SALARY □ Append □ Distinct

```
7/27/2007 4:16:21 PM    ***** Beginning Filter execution: "IA_BUSINESS_RULES_SALARY"
7/27/2007 4:16:22 PM    48 rows in table IA\\EMPLOYEE
7/27/2007 4:16:22 PM    Output:  Count
7/27/2007 4:16:22 PM    Exceptions: 26
7/27/2007 4:16:22 PM    Exception Percentage: 54.17
7/27/2007 4:16:27 PM    Filter results recorded
7/27/2007 4:16:27 PM    ***** Execution of "IA_BUSINESS_RULES_SALARY" was successful

7/27/2007 4:17:12 PM    ***** Beginning Filter execution: "IA_BUSINESS_RULES_SALARY"
7/27/2007 4:17:12 PM    48 rows in table IA\\EMPLOYEE
7/27/2007 4:17:12 PM    Output:  Exception Table IA\\BR_SALARY
7/27/2007 4:17:12 PM    Exceptions: 26
7/27/2007 4:17:12 PM    Exception Percentage: 54.17
7/27/2007 4:17:15 PM    Filter results recorded
7/27/2007 4:17:15 PM    ***** Execution of "IA_BUSINESS_RULES_SALARY" was successful
```

Ready 4:17 PM

Figure 1-306 Range check 5/5

Complex Data Filter

In this section, we describe more complex filters that involve an AND operator involving multiple predicates. Two business rules are shown here as follows:

- Non-sales and non-clerical employees earning a commission
- One and only one person has the job of President

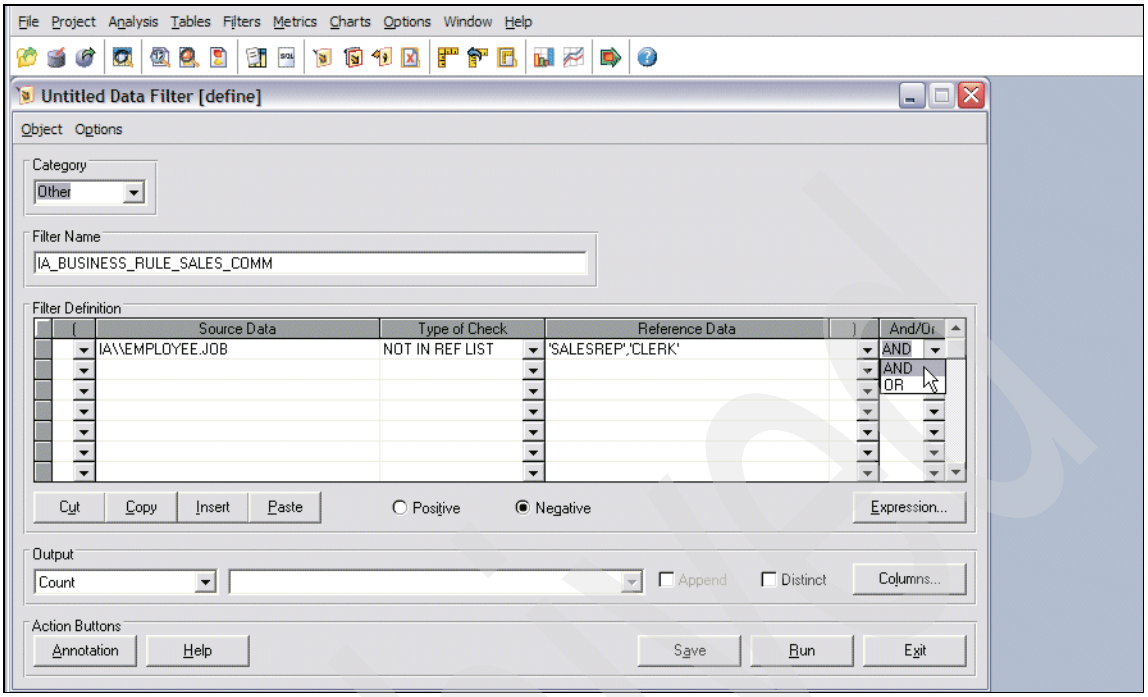
Non-sales and non-clerical employees earning a commission

We identify all employees that earn a commission who are not sales representatives or clerical staff using the using the NOT IN REF LIST and IN RANGE data filters. We write any exceptions to an exception table.

We show only the main panels of this business rule, because the rest are similar to those described in “Referential Integrity check” on page 350.

The following series of steps creates a complex data filter, executes it, and reviews the exceptions reported:

1. In the Untitled Data Filter (define) window, select **Negative**, and select **NOT IN REF LIST** from the Type of Check column. Key in comma separated ‘SALESREP’ and ‘CLERK’ values under the column Reference Data. Select AND from the And/Or column as shown in Figure 1-307 on page 373.
2. Add another predicate that checks the COMM column to be IN RANGE greater than zero as shown in Figure 1-308 on page 374. This time the **Positive** radio button is selected.
3. Select Exception Table from the Output drop down list and the name of the exception table (BR_COMM), and click **Run** as shown in Figure 1-309 on page 375.
4. Figure 1-310 on page 375 shows the results of the complex data filter execution. It identifies 39 exceptions out of a total of 49 rows indicating that there are 39 non-sales and non-clerical employees who earn a commission. These exceptions get written to the exception table BR_COMM. Click **OK**.
5. Figure 1-311 on page 376 shows the partial contents of the exception table BR_COMM that lists non-sales and non-clerical employees that earn a commission.



Category: Other

Filter Name: IA_BUSINESS_RULE_SALES_COMM

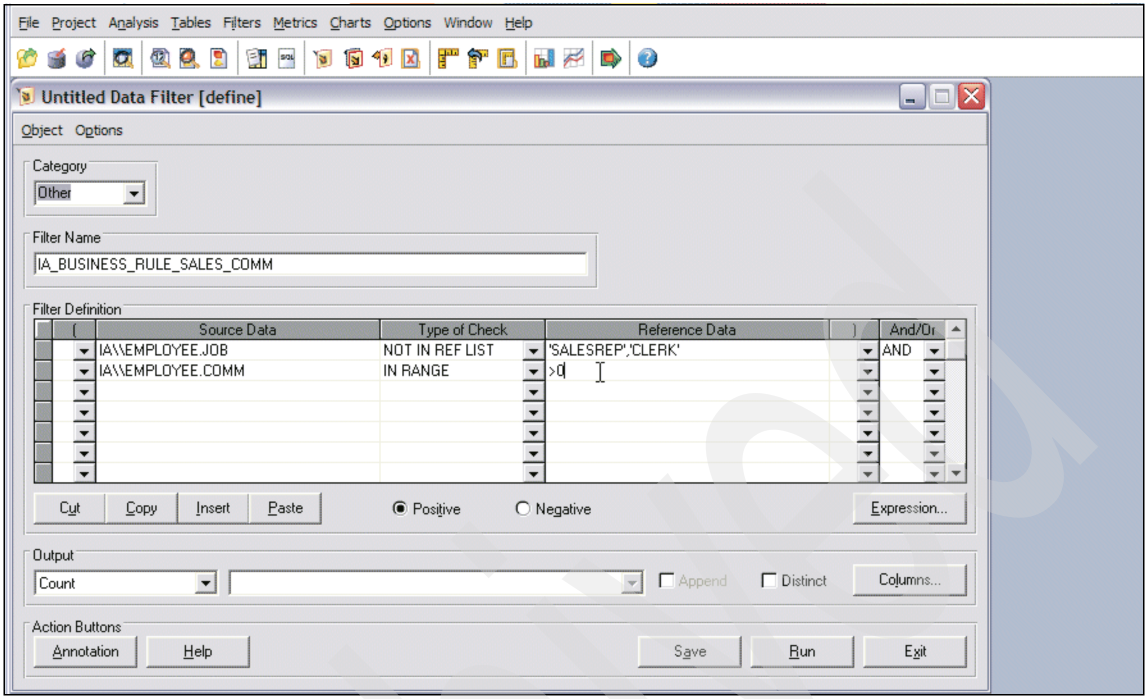
(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.JOB	NOT IN REF LIST	'SALESREP','CLERK'		AND

AND
OR

○ Positive ● Negative

Output: Count □ Append □ Distinct

Figure 1-307 Non-sales and non-clerical employees earning a commission 1/5


File Project Analysis Tables Filters Metrics Charts Options Window Help
Untitled Data Filter [define]
Object Options
Cut
Copy
Insert
Paste
Expression...
Columns...
Action Buttons
Annotation
Help
Save
Run
Exit

Category: Other

Filter Name: IA_BUSINESS_RULE_SALES_COMM

Filter Definition

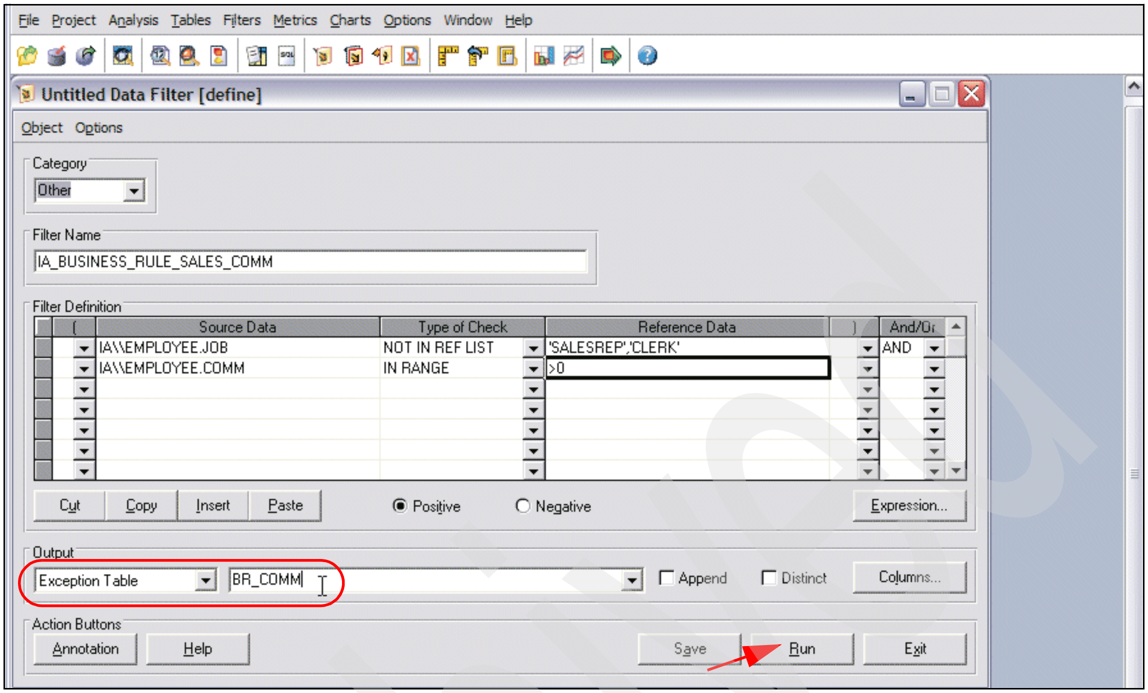
(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.JOB	NOT IN REF LIST	'SALESREP','CLERK'		AND
	IA\\EMPLOYEE.COMM	IN RANGE	>0		

● Positive ○ Negative

Output: Count

□ Append □ Distinct

Figure 1-308 Non-sales and non-clerical employees earning a commission 2/5



Untitled Data Filter [define]

Category: Other

Filter Name: IA_BUSINESS_RULE_SALES_COMM

Filter Definition

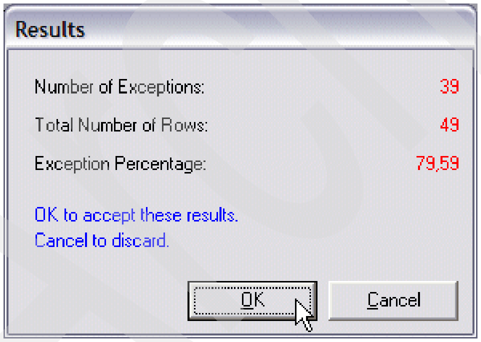
(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.JOB	NOT IN REF LIST	'SALESREP','CLERK'		AND
	IA\\EMPLOYEE.COMM	IN RANGE	>0		

● Positive ○ Negative

Output: Exception Table BR_COMM □ Append □ Distinct

Action Buttons

Figure 1-309 Non-sales and non-clerical employees earning a commission 3/5



Results

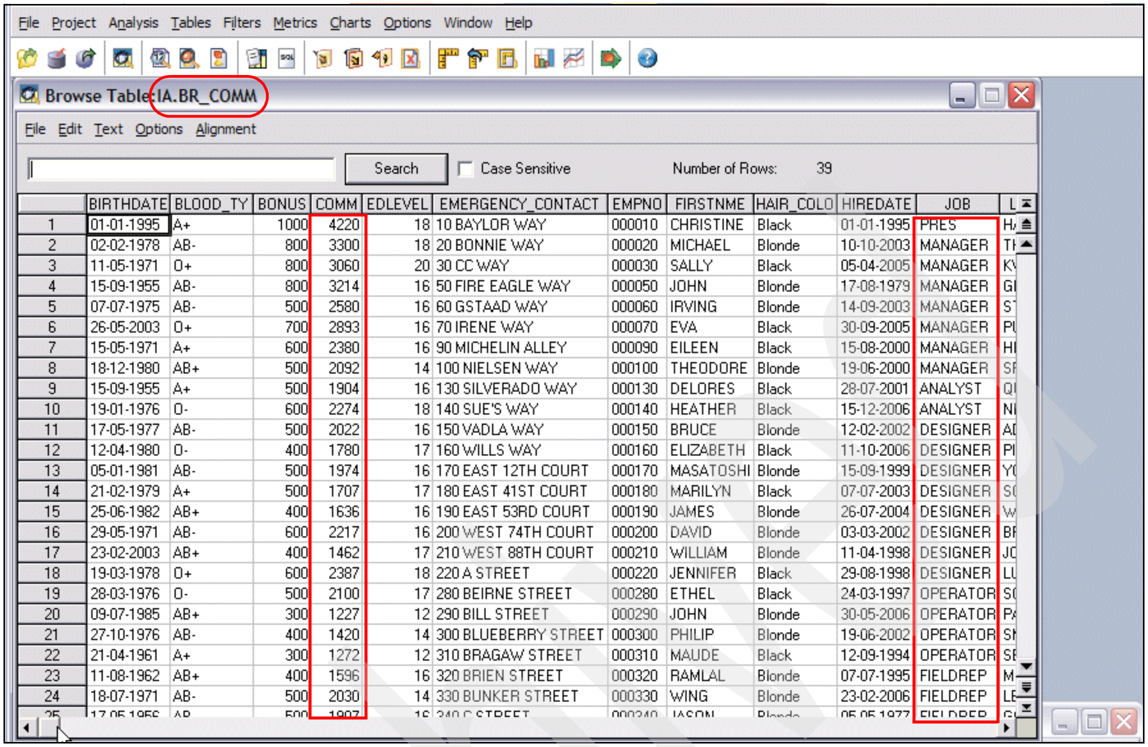
Number of Exceptions: 39

Total Number of Rows: 49

Exception Percentage: 79,59

OK to accept these results.
Cancel to discard.

Figure 1-310 Non-sales and non-clerical employees earning a commission 4/5


File Project Analysis Tables Filters Metrics Charts Options Window Help
Browse Table: IA.BR_COMM
File Edit Text Options Alignment
Search
□ Case Sensitive

Number of Rows: 39

	BIRTHDATE	BLOOD_TY	BONUS	COMM	EDLEVEL	EMERGENCY_CONTACT	EMPNO	FIRSTNME	HAIR_COLO	HIREDATE	JOB	L
1	01-01-1995	A+	1000	4220	18	10 BAYLOR WAY	000010	CHRISTINE	Black	01-01-1995	PRES	HA
2	02-02-1978	AB-	800	3300	18	20 BONNIE WAY	000020	MICHAEL	Blonde	10-10-2003	MANAGER	TH
3	11-05-1971	O+	800	3060	20	30 CC WAY	000030	SALLY	Black	05-04-2005	MANAGER	KV
4	15-09-1955	AB-	800	3214	16	50 FIRE EAGLE WAY	000050	JOHN	Blonde	17-08-1979	MANAGER	GI
5	07-07-1975	AB-	500	2580	16	60 GSTAAD WAY	000060	IRVING	Blonde	14-09-2003	MANAGER	ST
6	26-05-2003	O+	700	2893	16	70 IRENE WAY	000070	EVA	Black	30-09-2005	MANAGER	PU
7	15-05-1971	A+	600	2380	16	90 MICHELIN ALLEY	000090	EILEEN	Black	15-08-2000	MANAGER	HI
8	18-12-1980	AB+	500	2092	14	100 NIELSEN WAY	000100	THEODORE	Blonde	19-06-2000	MANAGER	SF
9	15-09-1955	A+	500	1904	16	130 SILVERADO WAY	000130	DELORES	Black	28-07-2001	ANALYST	QU
10	19-01-1976	O-	600	2274	18	140 SUE'S WAY	000140	HEATHER	Black	15-12-2006	ANALYST	NI
11	17-05-1977	AB-	500	2022	16	150 VADLA WAY	000150	BRUCE	Blonde	12-02-2002	DESIGNER	AD
12	12-04-1980	O-	400	1780	17	160 WILLS WAY	000160	ELIZABETH	Black	11-10-2006	DESIGNER	PI
13	05-01-1981	AB-	500	1974	16	170 EAST 12TH COURT	000170	MASATOSHI	Blonde	15-09-1999	DESIGNER	YO
14	21-02-1979	A+	500	1707	17	180 EAST 41ST COURT	000180	MARILYN	Black	07-07-2003	DESIGNER	SC
15	25-06-1982	AB+	400	1636	16	190 EAST 53RD COURT	000190	JAMES	Blonde	26-07-2004	DESIGNER	W
16	29-05-1971	AB-	600	2217	16	200 WEST 74TH COURT	000200	DAVID	Blonde	03-03-2002	DESIGNER	BR
17	23-02-2003	AB+	400	1462	17	210 WEST 88TH COURT	000210	WILLIAM	Blonde	11-04-1998	DESIGNER	JO
18	19-03-1978	O+	600	2387	18	220 A STREET	000220	JENNIFER	Black	29-08-1998	DESIGNER	LU
19	28-03-1976	O-	500	2100	17	280 BEIRNE STREET	000280	ETHEL	Black	24-03-1997	OPERATOR	SC
20	09-07-1985	AB+	300	1227	12	290 BILL STREET	000290	JOHN	Blonde	30-05-2006	OPERATOR	PA
21	27-10-1976	AB-	400	1420	14	300 BLUEBERRY STREET	000300	PHILIP	Blonde	19-06-2002	OPERATOR	SH
22	21-04-1961	A+	300	1272	12	310 BRAGAW STREET	000310	MAUDE	Black	12-09-1994	OPERATOR	SE
23	11-08-1962	AB+	400	1596	16	320 BRIEN STREET	000320	RAMLAL	Blonde	07-07-1995	FIELDREP	M-
24	18-07-1971	AB-	500	2030	14	330 BUNKER STREET	000330	WING	Blonde	23-02-2006	FIELDREP	LE
25	17-05-1956	AB	500	1907	16	340 C STREET	000340	JASON	Blonde	05-05-1977	FIELDREP	GI

Figure 1-311 Non-sales and non-clerical employees earning a commission 5/5

One and only one person has the job of president

We verify that there is one and only one employee having the job title of president using the OCCURS and EQUALS data filters. We just count the number of exceptions reported.

We show only the main panels of this business rule, because the rest are similar to those described in “Referential Integrity check” on page 350.

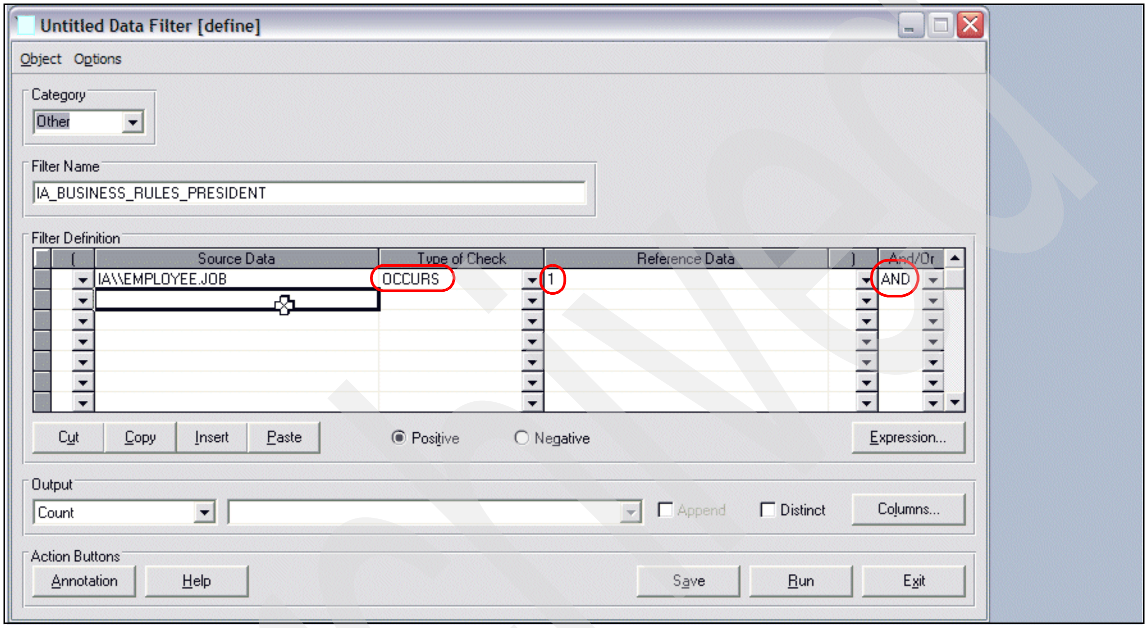
The following series of steps creates a complex data filter, executes it, and counts the exceptions reported:

1. In the Untitled Data Filter (define) window, select **Positive**, and select **OCCURS** from the Type of Check column. Key in a value 1 under the column Reference Data. Select AND from the And/Or column as shown in Figure 1-312 on page 377. This condition checks for only a single occurrence of a job as yet not identified.
2. Add another predicate that checks the JOB column as containing the string `PRES` corresponding to the president as shown in Figure 1-313 on page 378. The data filter check selected is EQUALS, and the string `PRES` is supplied

under the Reference Data column. This time, select **Positive**. Let the Output field default to Count. Click **Run**.

3. Figure 1-314 on page 378 shows the results of the complex data filter execution. It identifies one exception out of a total of 49 rows, indicating that there is only one occurrence of a row in the EMPLOYEE table with the job title of PRES.

   We do not show the screen output here.



Untitled Data Filter [define]

Category: Other

Filter Name: IA_BUSINESS_RULES_PRESIDENT

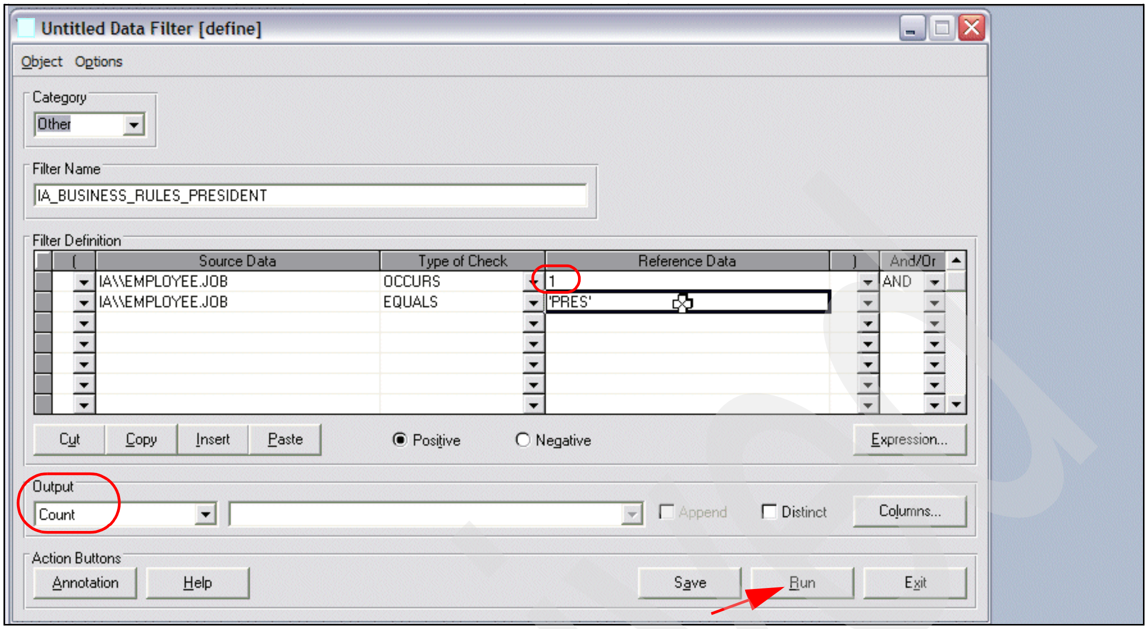
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.JOB	OCCURS	1		AND

● Positive ○ Negative

Output: Count □ Append □ Distinct

Figure 1-312 One and only one person with job of President 1/3



Untitled Data Filter [define]

Object Options

Category: Other

Filter Name: IA_BUSINESS_RULES_PRESIDENT

Filter Definition

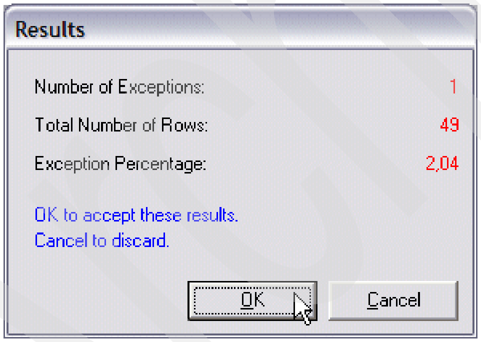
(	Source Data	Type of Check	Reference Data	)	And/Or
	IA\\EMPLOYEE.JOB	OCCURS	1		AND
	IA\\EMPLOYEE.JOB	EQUALS	'PRES'		

Cut Copy Insert Paste ● Positive ○ Negative Expression...

Output: Count □ Append □ Distinct Columns...

Action Buttons: Annotation Help Save Run Exit

Figure 1-313 One and only one person with job of President 2/3



Results

Number of Exceptions: 1

Total Number of Rows: 49

Exception Percentage: 2,04

OK to accept these results.
Cancel to discard.

OK Cancel

Figure 1-314 One and only one person with job of President 3/3

1.13 Baseline analysis

To determine whether the content and structure of your data has changed over time, you can use baseline analysis to compare a saved analysis summary of your table (baseline) to a current analysis result of the same table.

You can use baseline analysis to identify an analysis result that you want to set as the baseline for all comparisons. Over time, or as your data changes, you can import metadata for the table into the metadata repository again, run a column analysis job on that table, and then compare the analysis results from that job to the baseline analysis. You use the baseline version to compare all subsequent analysis results of the same data source.

> **Note:** You can continue to review and compare changes to the initial baseline as often as needed, or change the baseline if necessary. Only one baseline exists at any one time, creating a new baseline overwrites the previous one. If you know that your data has changed and that the changes are acceptable, you can create a new baseline at any time.

You can also save an analysis as a checkpoint. A maximum of one checkpoint can be defined. When you save an analysis as a checkpoint, it overwrites any previous checkpoint. You set a checkpoint to save the analysis results of the table for comparison. A checkpoint can also save results at a point in time for analysis publication.You can then choose to compare the baseline to the checkpoint or to the most recent analysis results.

In the following sections, we describe briefly the main functions of baseline analysis and the results produced by baseline analysis. Using a sample set of tables, we describe the process of setting a baseline, checkpoint, and performing comparison analysis. We discuss the results of the baseline analysis.

1.13.1 Baseline analysis functions

Baseline analysis has the following characteristics:

- You must have Information Analyzer Data Operator privileges to perform baseline analysis as shown in Table 1-4 on page 34. To view baseline analysis you only need Information Analyzer Business Analyst privileges.
- Can be performed on all the columns of one or more tables, or selectively on certain columns.
- Allows you to set a single baseline for a data source.
- Allows you to set a single checkpoint for a data source.

- Allows you to compare a current analysis with the baseline for the same data source.
- Allows you to compare a checkpoint analysis with the baseline for the same data source.
- Identifies changes to the structure and content of a data source over time.

Note: For complete details, refer to *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.

As mentioned earlier, when you want to know if your data has changed, you can use baseline analysis to compare the column analysis results for two versions of the same data source. The content and structure of data changes over time when it is accessed by multiple users. When the structure of data changes, the system processes that use the data are affected.

To compare your data, you choose the analysis results that you want to set as the baseline version. You use the baseline version to compare all subsequent analysis results of the same data source. For example, if you ran a column analysis job on data source A on Tuesday, you could then set the column analysis results of source A as the baseline and save the baseline in the repository. On Wednesday, when you run a column analysis job on data source A again, you can then compare the current analysis results of data source A with the baseline results of data source A.

Note: If you know that your data has changed and that the changes are acceptable, you can create a new baseline at any time. As mentioned earlier, there can only be one baseline analysis. When a new baseline is created, it overwrites the previous one.

To identify changes in your data, a baseline analysis job evaluates the content and structure of the data for differences between the baseline results and the current results. The content and structure of your data consists of elements such as data classes, data properties, primary keys, and data values. If the content of your data has changed, there will be differences between the elements of each version of the data.

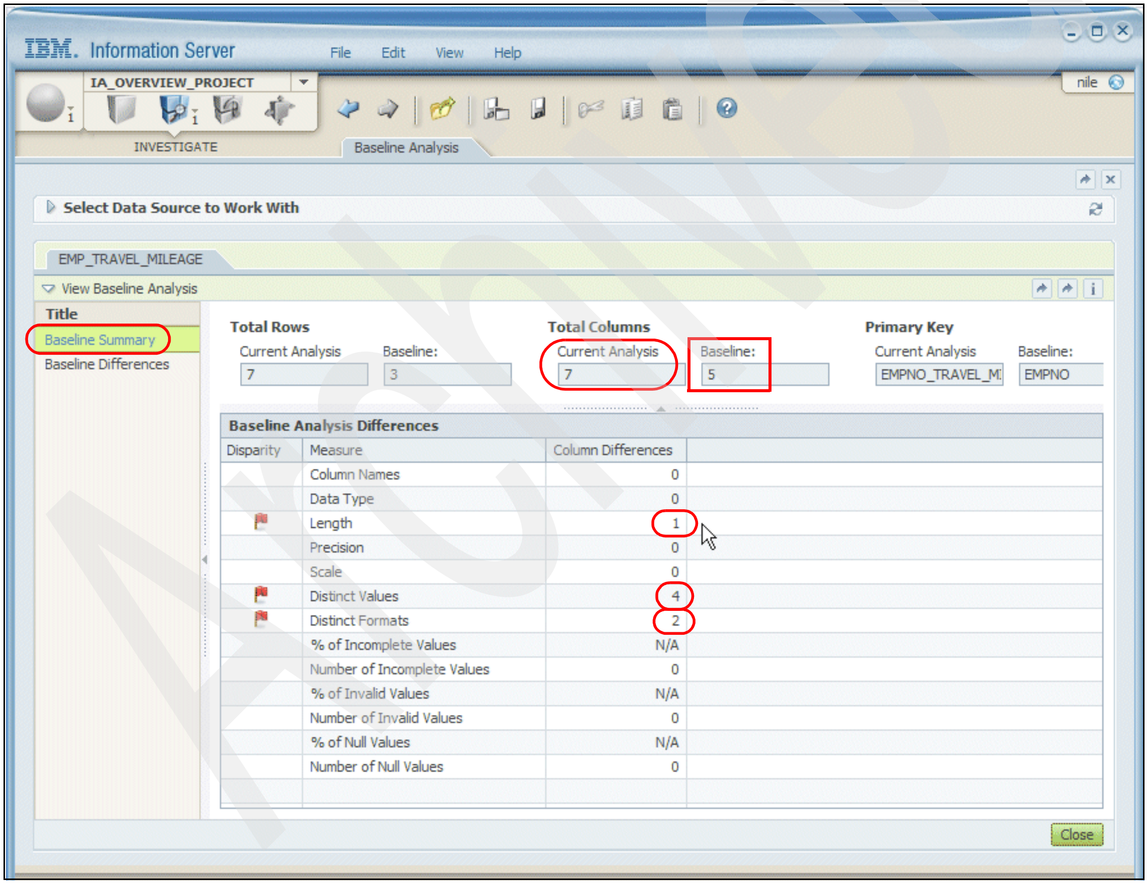
Note: If you are monitoring changes in the structure and content of your data on a regular basis, you might want to specify a checkpoint at regular intervals to compare to the baseline. You set a checkpoint to save the analysis results of the table for comparison. You can then choose to compare the baseline to the checkpoint or to the most recent analysis results.

1.13.2 Baseline analysis results

A baseline analysis produces the following output:

- Baseline Summary

  This view summarizes the differences between a baseline and a current analysis (or checkpoint) as shown in Figure 1-315 on page 381. It shows a comparison of the current analysis of the EMP_TRAVEL_MILEAGE table with the baseline. It shows a difference in the number of columns in this table: 7 columns in the current analysis as compared to 5 columns in the baseline analysis. The red flags indicate that there are one or more columns with differences between the current analysis and baseline in the Length, Distinct Values, and Distinct Formats measures.



Total Rows
Current Analysis: 7
Baseline: 3

Total Columns
Current Analysis: 7
Baseline: 5

Primary Key
Current Analysis: EMPNO_TRAVEL_M
Baseline: EMPNO

Baseline Analysis Differences

Disparity	Measure	Column Differences
	Column Names	0
	Data Type	0
🟥	Length	1
	Precision	0
	Scale	0
🟥	Distinct Values	4
🟥	Distinct Formats	2
	% of Incomplete Values	N/A
	Number of Incomplete Values	0
	% of Invalid Values	N/A
	Number of Invalid Values	0
	% of Null Values	N/A
	Number of Null Values	0

Figure 1-315 View Baseline Analysis - Baseline Summary

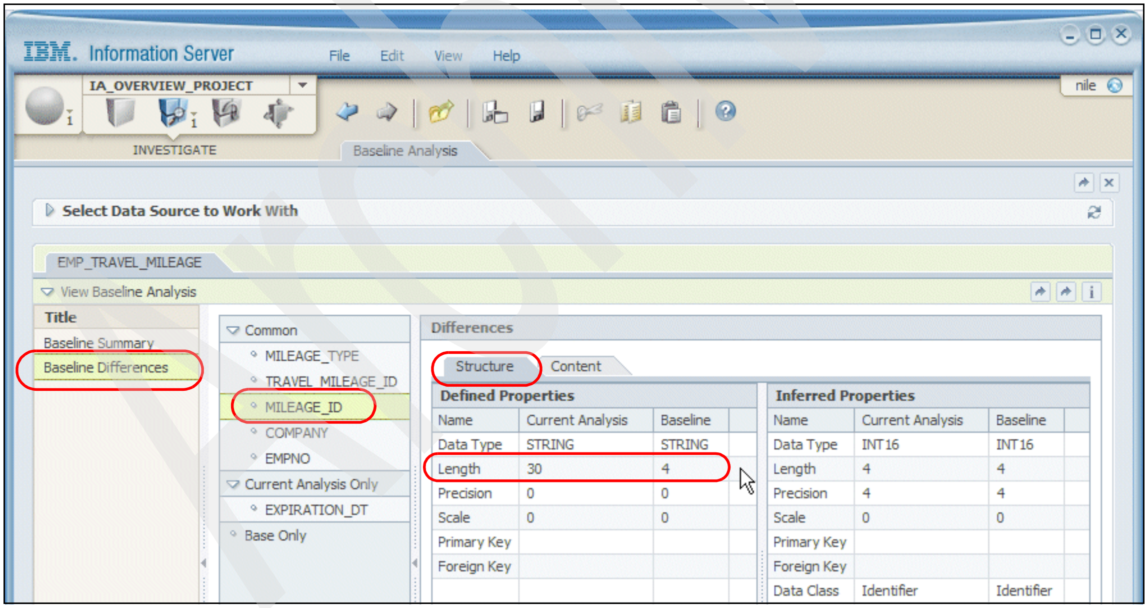
- Baseline Differences

  This view shows the difference between a certain version of the data and the baseline version of your data as shown in Figure 1-316 on page 382 and Figure 1-317 on page 383. It shows a comparison of the current analysis of the EMP_TRAVEL_MILEAGE table with the baseline.

  > **Attention:** To locate changes in data content, you must run column analysis again.
  >
  > To locate changes in data structure, you must first re-authenticate and re-import data sources. You must then add the re-imported data sources to your project again to update project metadata.

  – Figure 1-316 on page 382 shows the structure changes between the current analysis and the baseline. The MILEAGE_ID field has a length of 30 bytes in the current analysis but 4 bytes in the baseline.

  – Figure 1-317 on page 383 shows the content changes between the current analysis and the baseline. The MILEAGE_ID field has differences in the Cardinality, Distinct Values, and Distinct Formats.



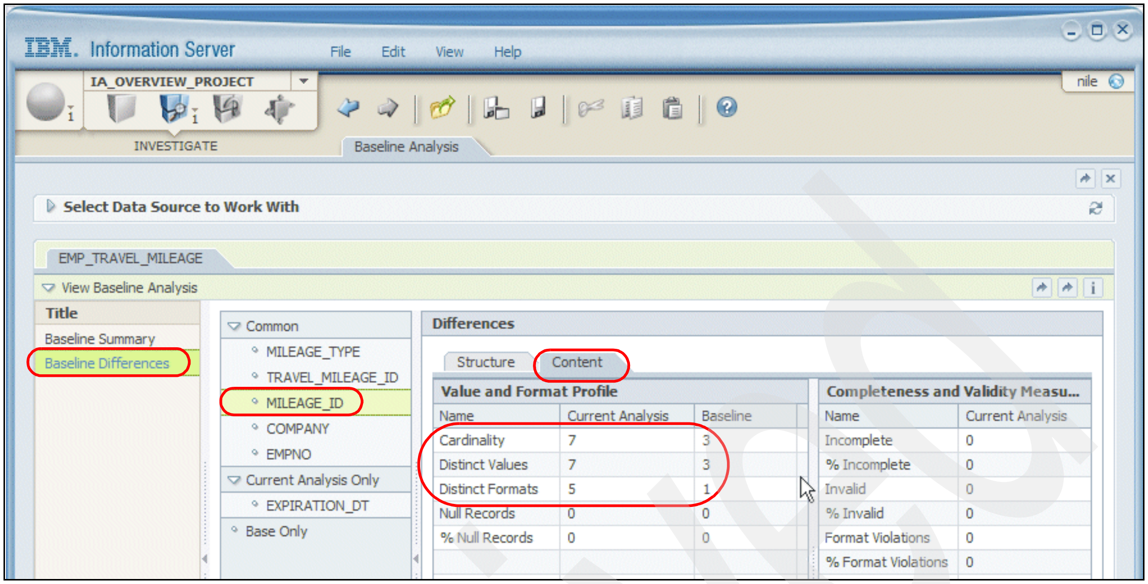
Defined Properties

Name	Current Analysis	Baseline
Data Type	STRING	STRING
Length	30	4
Precision	0	0
Scale	0	0
Primary Key		
Foreign Key		

Inferred Properties

Name	Current Analysis	Baseline
Data Type	INT16	INT16
Length	4	4
Precision	4	4
Scale	0	0
Primary Key		
Foreign Key		
Data Class	Identifier	Identifier

Figure 1-316 View Baseline Analysis - Baseline Differences Structure



Value and Format Profile

Name	Current Analysis	Baseline
Cardinality	7	3
Distinct Values	7	3
Distinct Formats	5	1
Null Records	0	0
% Null Records	0	0

Completeness and Validity Measu...

Name	Current Analysis
Incomplete	0
% Incomplete	0
Invalid	0
% Invalid	0
Format Violations	0
% Format Violations	0

Figure 1-317 View Baseline Analysis - Baseline Differences Content

1.13.3 Baseline analysis usage scenario

In this section, we describe how to set a baseline, set a checkpoint, and compare current analysis with a baseline.

> **Attention:** In the following sections, to avoid overburdening the text with excessive figures, we have *not* included all the panels that you would navigate through typically to perform the desired functions. Instead, we focus on including select panels (and in some cases, just portions of these panels) that highlight the key items of interest, thereby skipping the initial panels, as well as some intervening ones, in the process.

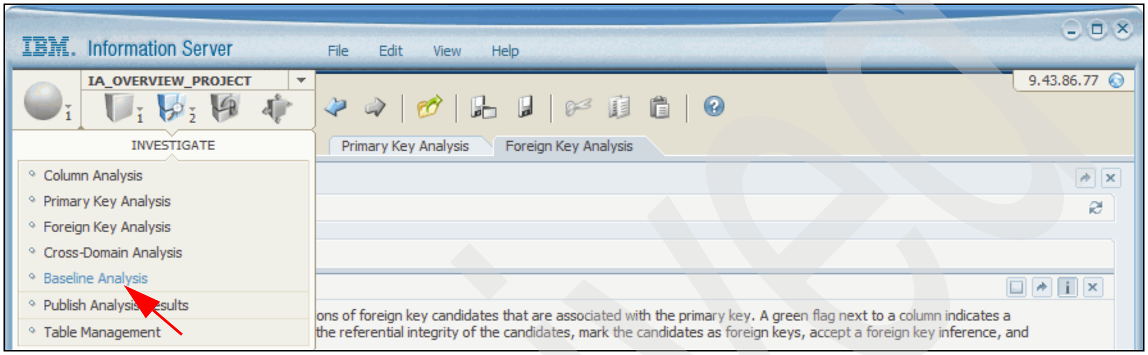
Set a baseline

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), set a baseline on multiple tables as follows:

1. On the Investigate navigator menu in the console, select **Baseline Analysis** as shown in Figure 1-318 on page 384.
2. In the Baseline Analysis workspace, select the EMP_TRAVEL_MILEAGE table with the IA schema and click **Set Baseline** on the Tasks pane as shown in Figure 1-319 on page 384.

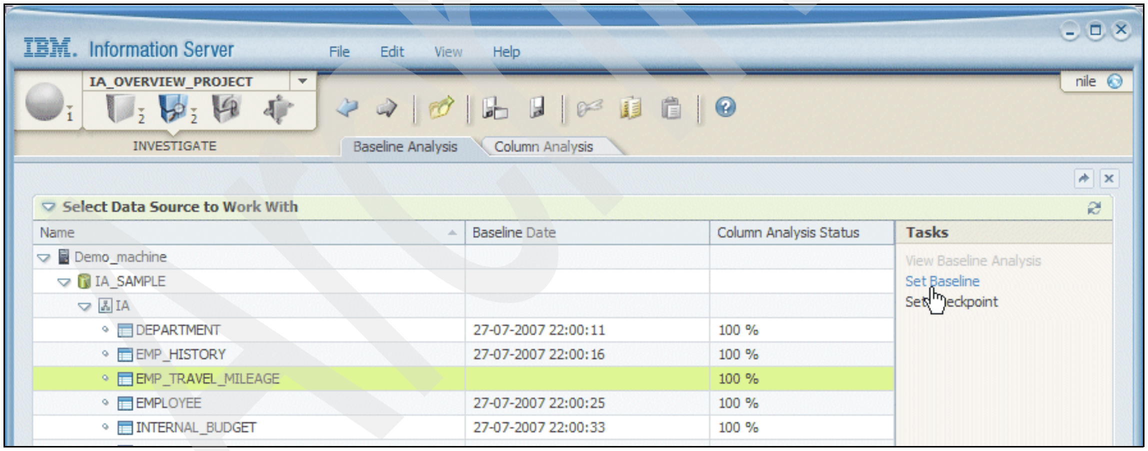
3. The Baseline Window in Figure 1-320 on page 385 shows the EMP_TRAVEL_MILEAGE table for which a baseline has been set. Click **Close**.
4. The Baseline Analysis workspace in Figure 1-321 on page 385 shows all the tables and the Baseline Date when it was set.

You can now compare the analysis baseline to a subsequent analysis result of the table.



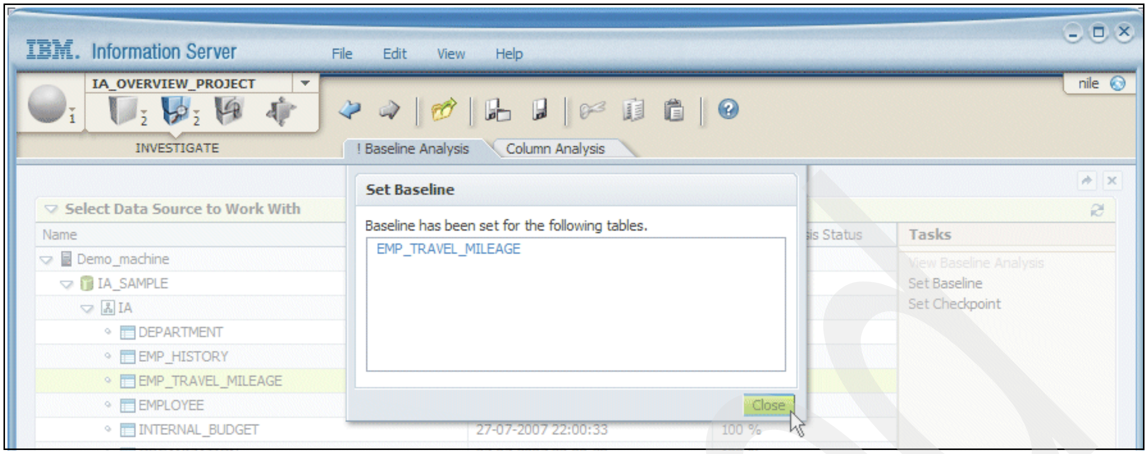
ons of foreign key candidates that are associated with the primary key. A green flag next to a column indicates a
the referential integrity of the candidates, mark the candidates as foreign keys, accept a foreign key inference, and

Figure 1-318 Set Baseline 1/4



Name	Baseline Date	Column Analysis Status
Demo_machine		
IA_SAMPLE		
IA		
DEPARTMENT	27-07-2007 22:00:11	100 %
EMP_HISTORY	27-07-2007 22:00:16	100 %
EMP_TRAVEL_MILEAGE		100 %
EMPLOYEE	27-07-2007 22:00:25	100 %
INTERNAL_BUDGET	27-07-2007 22:00:33	100 %

Figure 1-319 Set Baseline 2/4

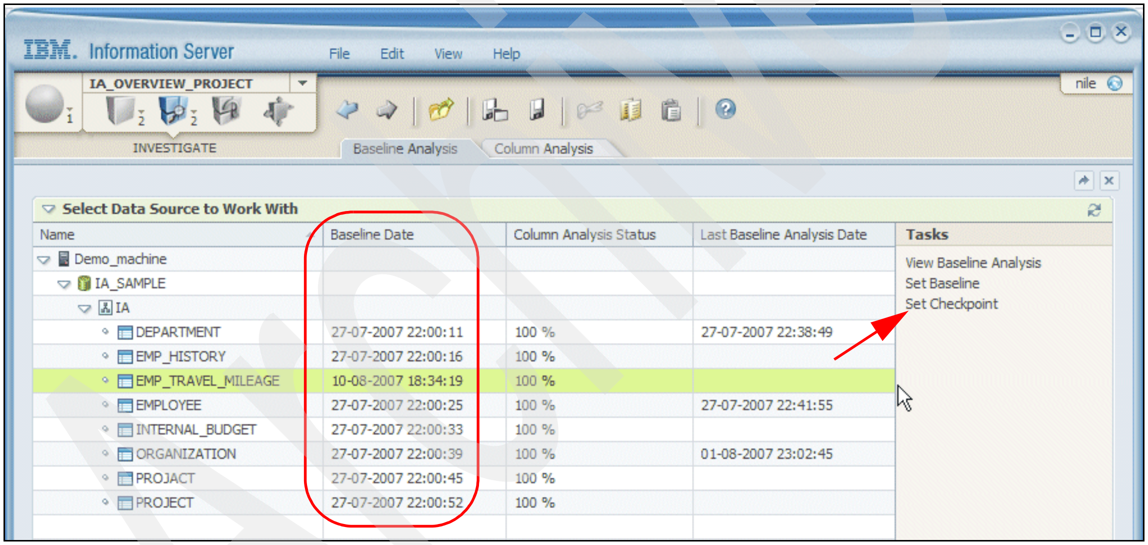


Set Baseline

Baseline has been set for the following tables.

EMP_TRAVEL_MILEAGE

Figure 1-320 Set Baseline 3/4



Select Data Source to Work With

Name	Baseline Date	Column Analysis Status	Last Baseline Analysis Date
Demo_machine			
IA_SAMPLE			
IA			
DEPARTMENT	27-07-2007 22:00:11	100 %	27-07-2007 22:38:49
EMP_HISTORY	27-07-2007 22:00:16	100 %	
EMP_TRAVEL_MILEAGE	10-08-2007 18:34:19	100 %	
EMPLOYEE	27-07-2007 22:00:25	100 %	27-07-2007 22:41:55
INTERNAL_BUDGET	27-07-2007 22:00:33	100 %	
ORGANIZATION	27-07-2007 22:00:39	100 %	01-08-2007 23:02:45
PROJACT	27-07-2007 22:00:45	100 %	
PROJECT	27-07-2007 22:00:52	100 %	

Tasks

- View Baseline Analysis
- Set Baseline
- Set Checkpoint

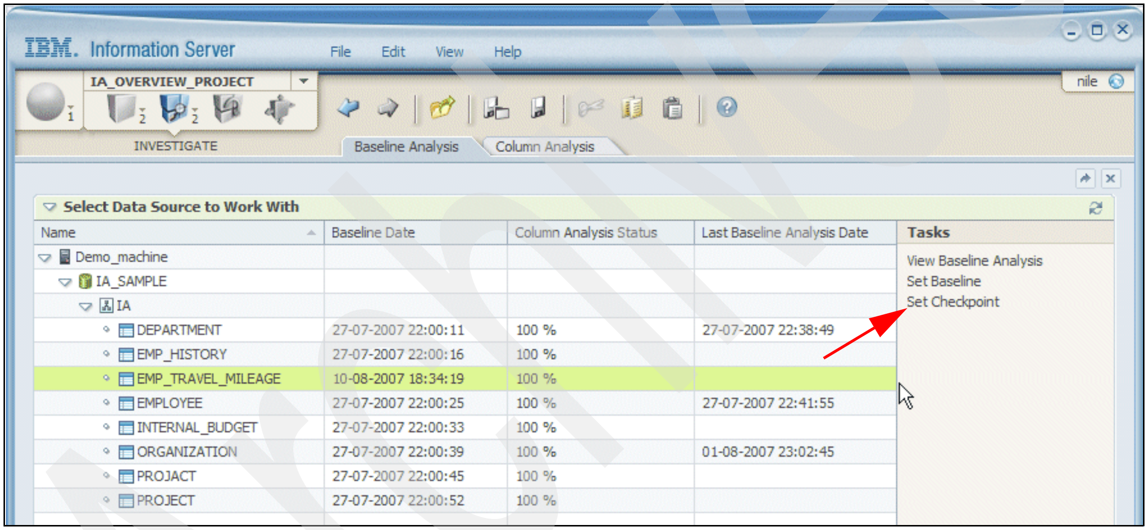
Figure 1-321 Set Baseline 4/4

Set a checkpoint

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), set a checkpoint on multiple tables as follows:

1. On the Investigate navigator menu in the console, select **Baseline Analysis** as shown in Figure 1-318 on page 384.
2. In the Baseline Analysis workspace, select the EMP_TRAVEL_MILEAGE table with the IA schema and click **Set Checkpoint** on the Tasks pane as shown in Figure 1-321 on page 385.
3. The Update Checkpoint window in Figure 1-322 on page 386 shows the EMP_TRAVEL_MILEAGE table for which checkpoint has been updated. Click **Close**.

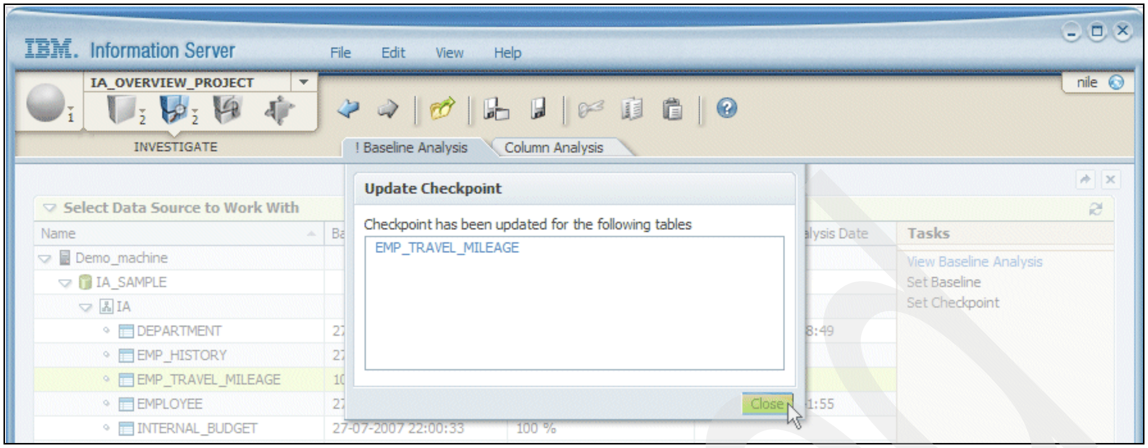
You can now compare the checkpoint to the baseline.



Select Data Source to Work With

Name	Baseline Date	Column Analysis Status	Last Baseline Analysis Date
Demo_machine			
IA_SAMPLE			
IA			
DEPARTMENT	27-07-2007 22:00:11	100 %	27-07-2007 22:38:49
EMP_HISTORY	27-07-2007 22:00:16	100 %	
EMP_TRAVEL_MILEAGE	10-08-2007 18:34:19	100 %	
EMPLOYEE	27-07-2007 22:00:25	100 %	27-07-2007 22:41:55
INTERNAL_BUDGET	27-07-2007 22:00:33	100 %	
ORGANIZATION	27-07-2007 22:00:39	100 %	01-08-2007 23:02:45
PROJACT	27-07-2007 22:00:45	100 %	
PROJECT	27-07-2007 22:00:52	100 %	

Figure 1-322 Set Checkpoint 1/2



Update Checkpoint

Checkpoint has been updated for the following tables

EMP_TRAVEL_MILEAGE

Figure 1-323 Set Checkpoint 2/2

Compare results

After opening the project of interest (IA_SAMPLE_OVERVIEW in our case), compare current analysis results with the baseline, and checkpoint with the baseline for the EMP_TRAVEL_MILEAGE table.

Current analysis versus baseline analysis

To compare results between current analysis and baseline analysis, proceed as follows:

1. On the Investigate navigator menu in the console, select **Baseline Analysis** as shown in Figure 1-318 on page 384.

1. In the Baseline Analysis workspace, select the EMP_TRAVEL_MILEAGE table for which we want to compare to the baseline analysis, and click **View Baseline Analysis** on the Tasks pane as shown in Figure 1-324 on page 388.

2. In the Pick an Analysis Summary window in Figure 1-325 on page 388, select which analysis result you want to compare to the baseline, and click **OK**.

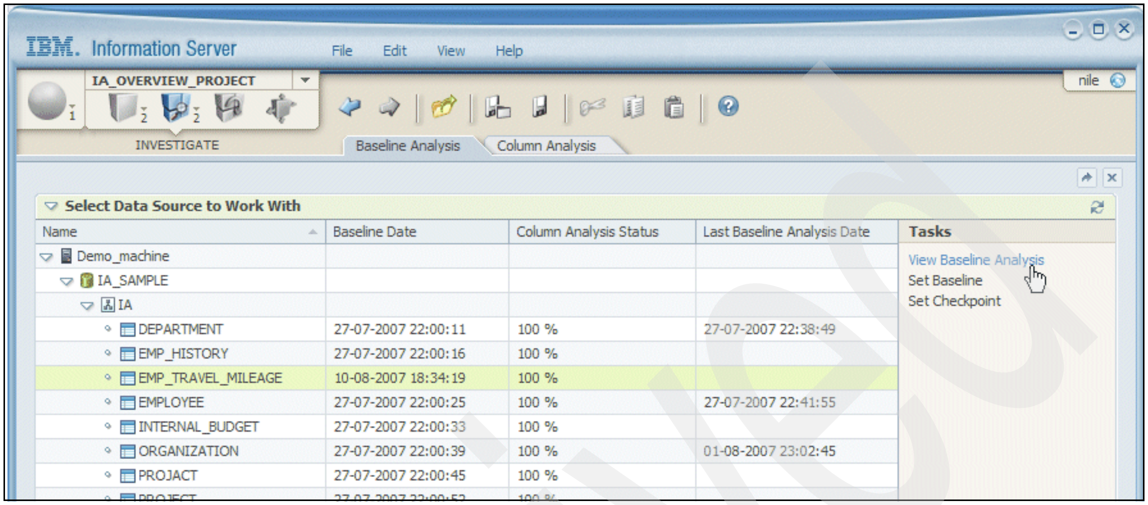
   – Select Checkpoint to compare the baseline to the latest checkpoint analysis. Remember that only one checkpoint analysis is maintained.

   – Select Current Analysis to compare the baseline to the last run analysis job. This is what we chose here.

3. The View Baseline Analysis pane shown in Figure 1-326 on page 389 details the changes in data.

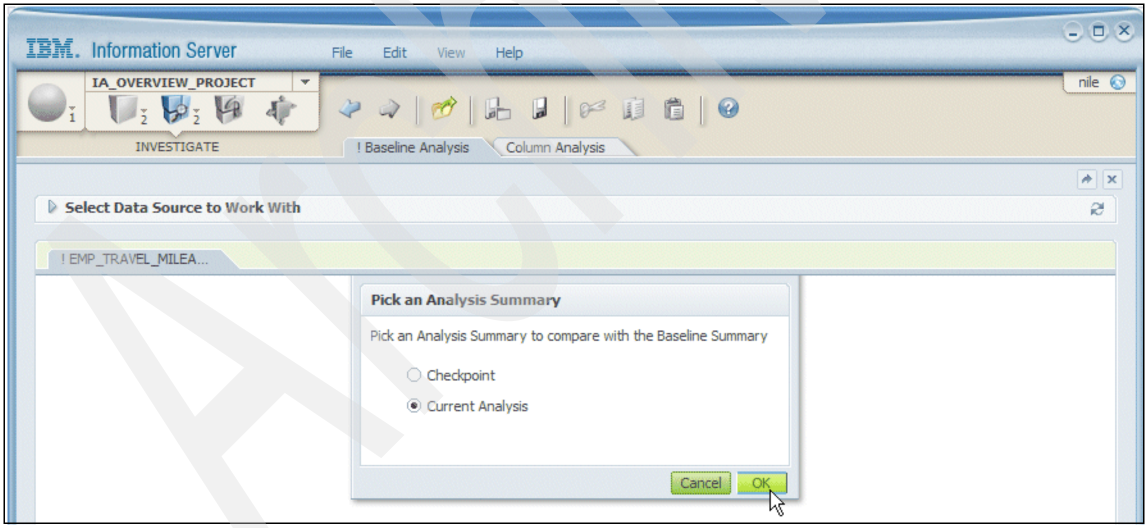
   The Baseline Summary summarizes the structure and content differences as highlighted by the red flag.

4. The Baseline Differences shown in Figure 1-327 on page 390 (structure) and Figure 1-328 on page 390 (content) highlight the differences between the current analysis and the baseline.



Name	Baseline Date	Column Analysis Status	Last Baseline Analysis Date
Demo_machine			
IA_SAMPLE			
IA			
DEPARTMENT	27-07-2007 22:00:11	100 %	27-07-2007 22:38:49
EMP_HISTORY	27-07-2007 22:00:16	100 %	
EMP_TRAVEL_MILEAGE	10-08-2007 18:34:19	100 %	
EMPLOYEE	27-07-2007 22:00:25	100 %	27-07-2007 22:41:55
INTERNAL_BUDGET	27-07-2007 22:00:33	100 %	
ORGANIZATION	27-07-2007 22:00:39	100 %	01-08-2007 23:02:45
PROJACT	27-07-2007 22:00:45	100 %	
PROJECT	27-07-2007 22:00:52	100 %	

Figure 1-324 View Baseline Analysis versus Current Analysis 1/5



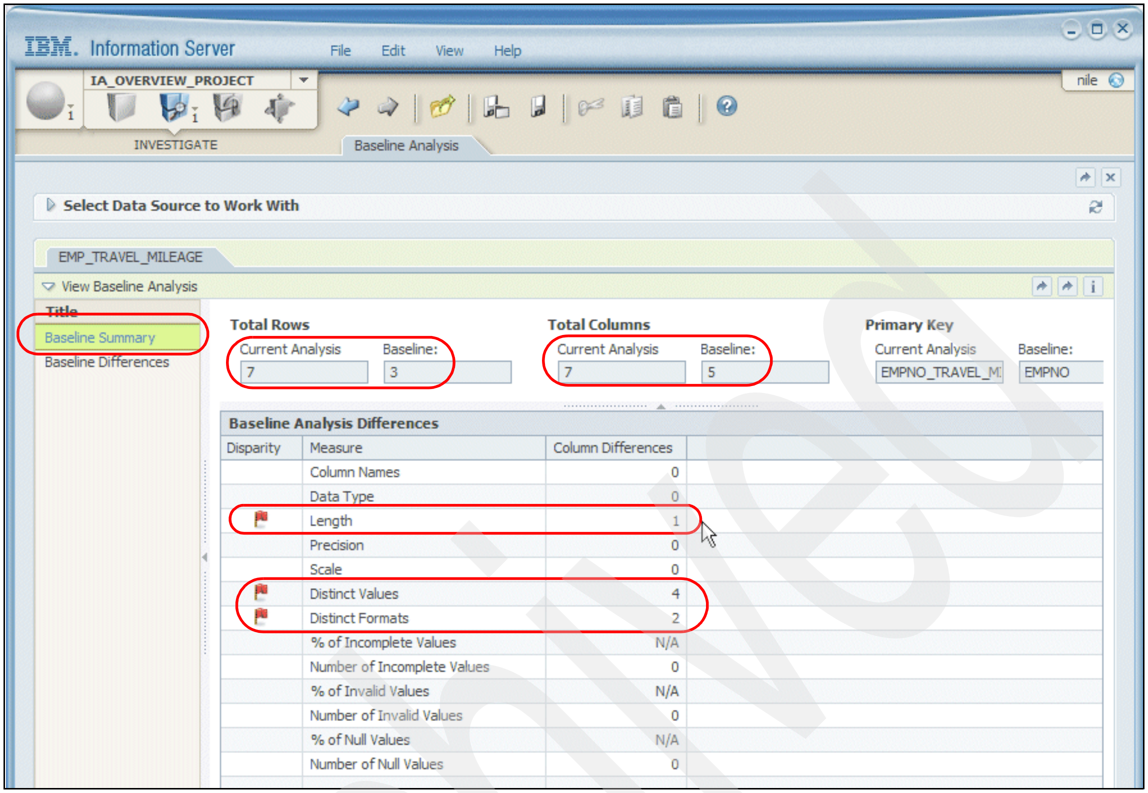
Pick an Analysis Summary

Pick an Analysis Summary to compare with the Baseline Summary

○ Checkpoint

● Current Analysis

Figure 1-325 View Baseline Analysis versus Current Analysis 2/5



Select Data Source to Work With

EMP_TRAVEL_MILEAGE

View Baseline Analysis

Title

Baseline Summary

Baseline Differences

Total Rows

Current Analysis: 7

Baseline: 3

Total Columns

Current Analysis: 7

Baseline: 5

Primary Key

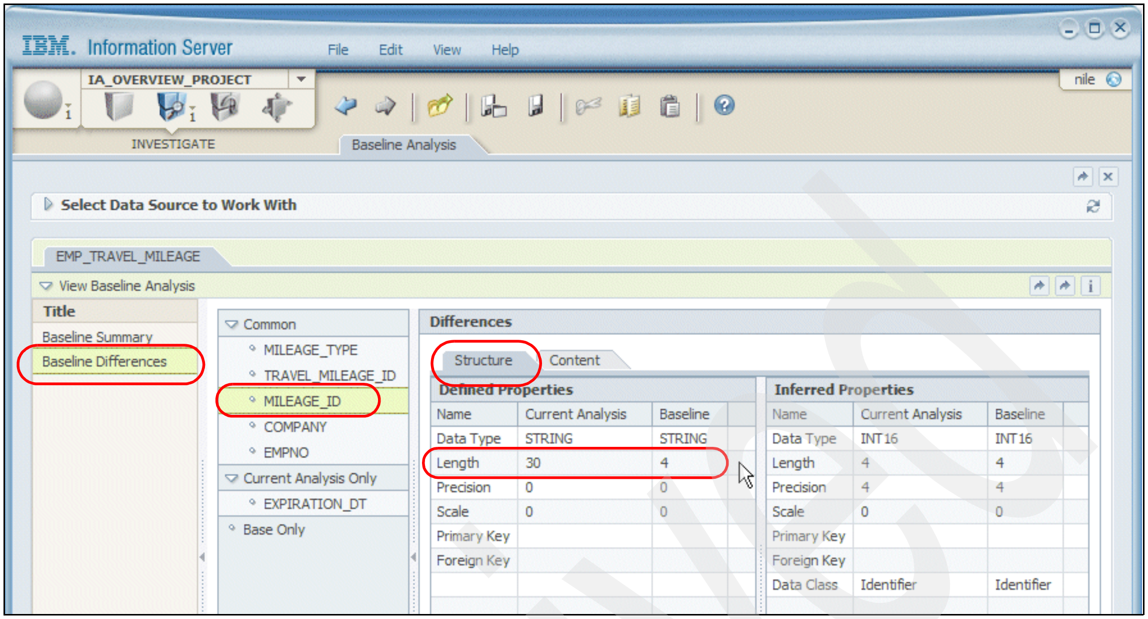
Current Analysis: EMPNO_TRAVEL_M

Baseline: EMPNO

Baseline Analysis Differences

Disparity	Measure	Column Differences
	Column Names	0
	Data Type	0
🚩	Length	1
	Precision	0
	Scale	0
🚩	Distinct Values	4
🚩	Distinct Formats	2
	% of Incomplete Values	N/A
	Number of Incomplete Values	0
	% of Invalid Values	N/A
	Number of Invalid Values	0
	% of Null Values	N/A
	Number of Null Values	0

Figure 1-326 View Baseline Analysis versus Current Analysis 3/5



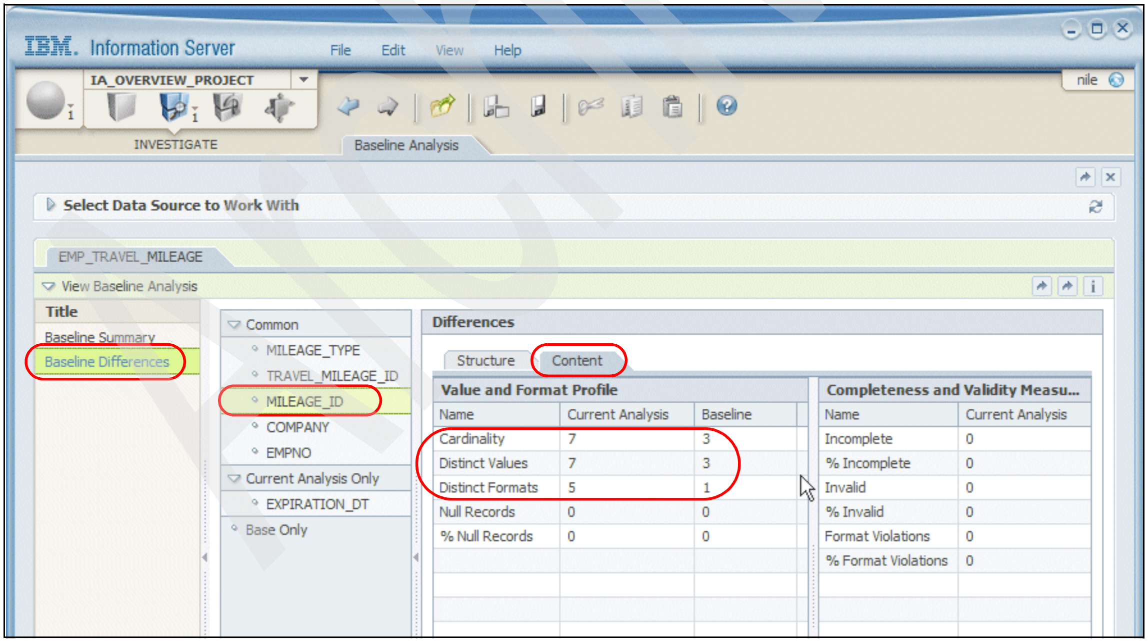
Defined Properties

Name	Current Analysis	Baseline
Data Type	STRING	STRING
Length	30	4
Precision	0	0
Scale	0	0
Primary Key		
Foreign Key		

Inferred Properties

Name	Current Analysis	Baseline
Data Type	INT16	INT16
Length	4	4
Precision	4	4
Scale	0	0
Primary Key		
Foreign Key		
Data Class	Identifier	Identifier

Figure 1-327 View Baseline Analysis versus Current Analysis 4/5



Value and Format Profile

Name	Current Analysis	Baseline
Cardinality	7	3
Distinct Values	7	3
Distinct Formats	5	1
Null Records	0	0
% Null Records	0	0

Completeness and Validity Measu...

Name	Current Analysis
Incomplete	0
% Incomplete	0
Invalid	0
% Invalid	0
Format Violations	0
% Format Violations	0

Figure 1-328 View Baseline Analysis versus Current Analysis 5/5

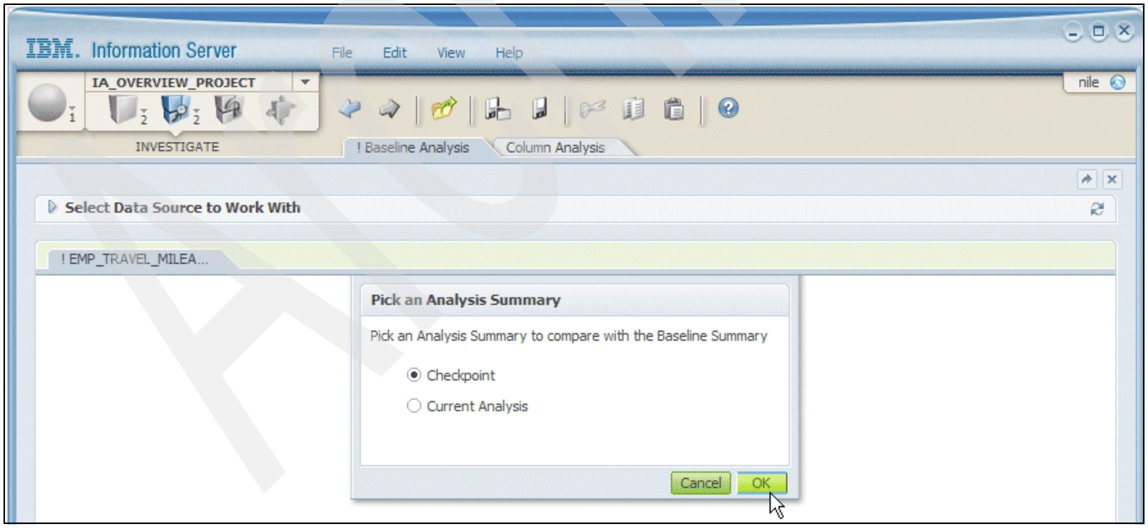
Checkpoint versus baseline analysis

To compare results between checkpoint and baseline analysis, proceed as follows:

1. On the Investigate navigator menu in the console, select **Baseline Analysis** as shown in Figure 1-318 on page 384.
2. In the Baseline Analysis workspace, select the EMP_TRAVEL_MILEAGE table for which you want to compare to the baseline analysis, and click **View Baseline Analysis** on the Tasks pane as shown in Figure 1-324 on page 388.
3. In the Pick an Analysis Summary window shown in Figure 1-329 on page 391, select the analysis result to which you want to compare the baseline, and click **OK**.

   Select **Checkpoint** to compare the baseline to the latest checkpoint analysis. Remember that only one checkpoint analysis is maintained.
4. The View Baseline Analysis pane shown in Figure 1-330 on page 392 details the changes in data.

   The Baseline Summary summarizes the structure and content differences as highlighted by the red flag.
5. The Baseline Differences in Figure 1-331 on page 393 (structure) and Figure 1-332 on page 393 (content) highlight the differences between the checkpoint and the baseline.



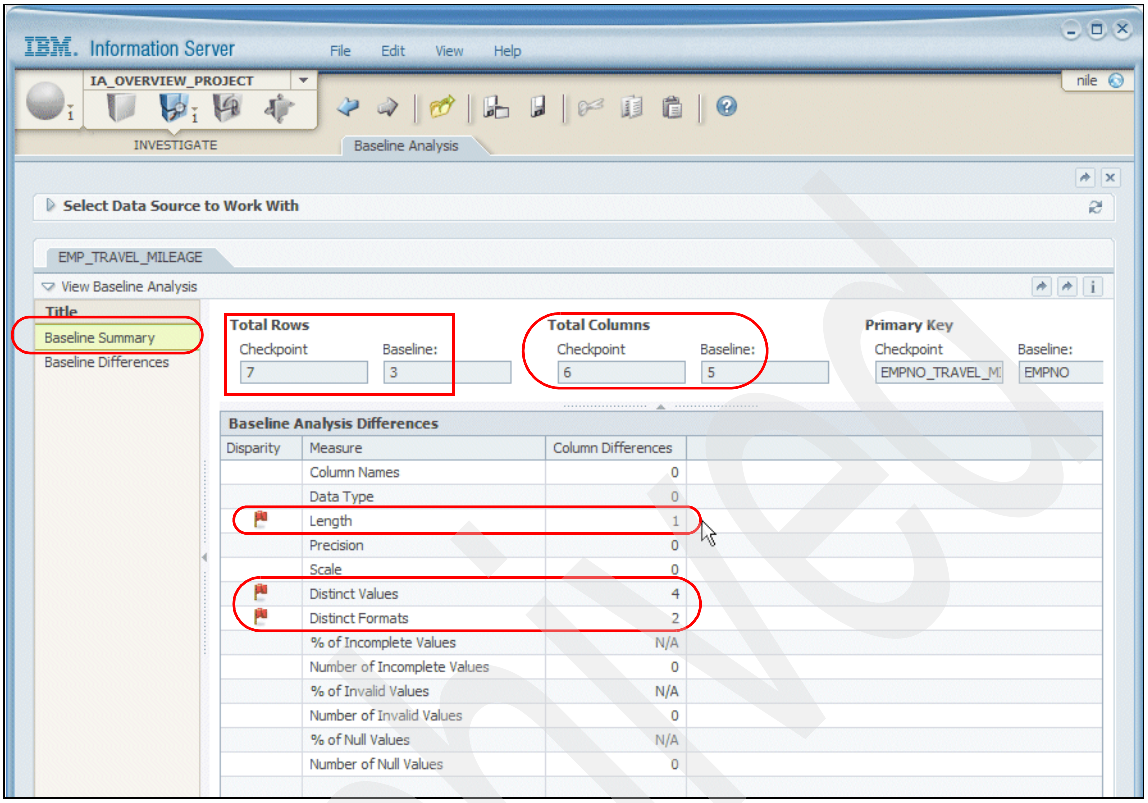
Pick an Analysis Summary

Pick an Analysis Summary to compare with the Baseline Summary

● Checkpoint

○ Current Analysis

Figure 1-329   View Baseline Analysis versus Checkpoint 1/4



Total Rows

Checkpoint: 7

Baseline: 3

Total Columns

Checkpoint: 6

Baseline: 5

Primary Key

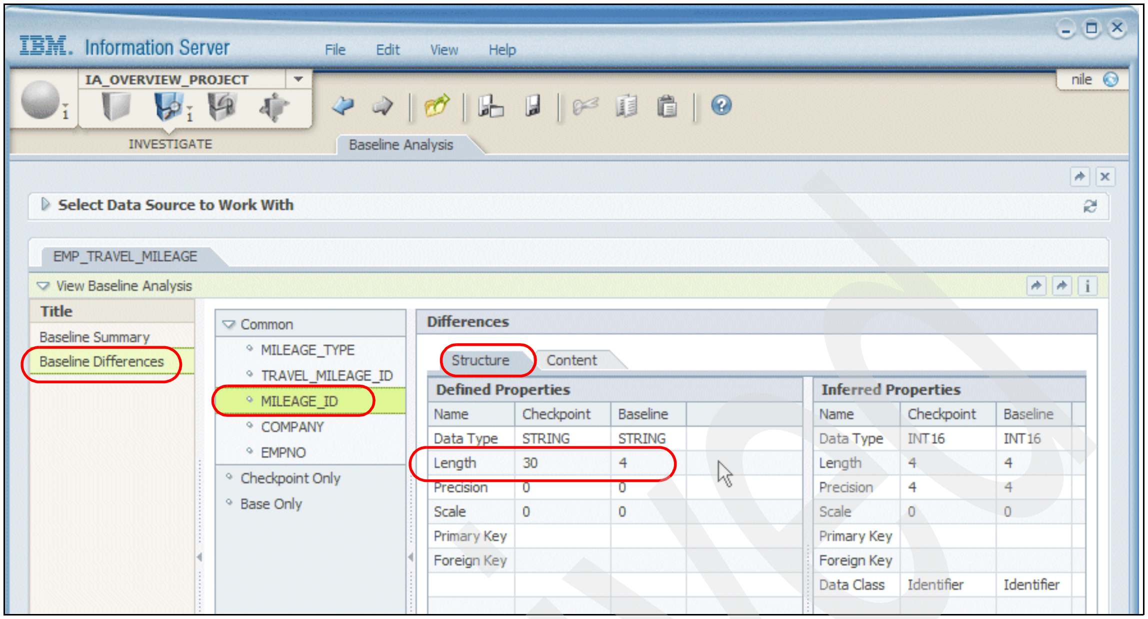
Checkpoint: EMPNO_TRAVEL_M

Baseline: EMPNO

Baseline Analysis Differences

Disparity	Measure	Column Differences
	Column Names	0
	Data Type	0
⚑	Length	1
	Precision	0
	Scale	0
⚑	Distinct Values	4
⚑	Distinct Formats	2
	% of Incomplete Values	N/A
	Number of Incomplete Values	0
	% of Invalid Values	N/A
	Number of Invalid Values	0
	% of Null Values	N/A
	Number of Null Values	0

Figure 1-330 View Baseline Analysis versus Checkpoint 2/4



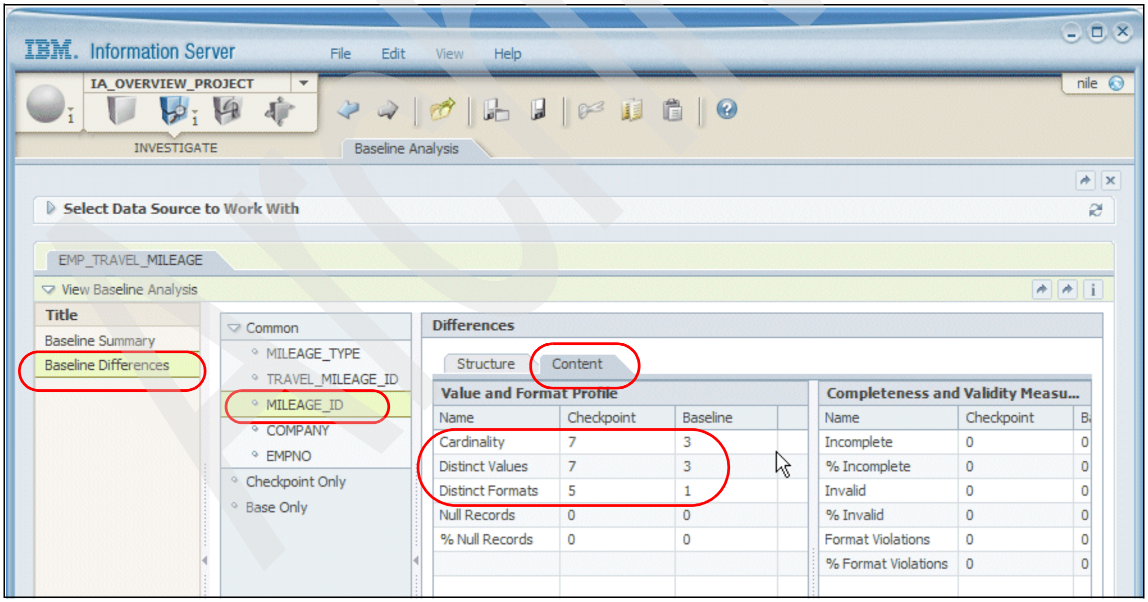
Defined Properties

Name	Checkpoint	Baseline
Data Type	STRING	STRING
Length	30	4
Precision	0	0
Scale	0	0
Primary Key		
Foreign Key		

Inferred Properties

Name	Checkpoint	Baseline
Data Type	INT16	INT16
Length	4	4
Precision	4	4
Scale	0	0
Primary Key		
Foreign Key		
Data Class	Identifier	Identifier

Figure 1-331 View Baseline Analysis versus Checkpoint 3/4



Value and Format Profile

Name	Checkpoint	Baseline
Cardinality	7	3
Distinct Values	7	3
Distinct Formats	5	1
Null Records	0	0
% Null Records	0	0

Completeness and Validity Measu...

Name	Checkpoint	B
Incomplete	0	0
% Incomplete	0	0
Invalid	0	0
% Invalid	0	0
Format Violations	0	0
% Format Violations	0	0

Figure 1-332 View Baseline Analysis versus Checkpoint 4/4

1.14 Reports

You can create reports that summarize analysis results and show details about your project. Reports are saved in the metadata repository and can be accessed by any user who is authorized to view them. Reports can be created without opening a project.

Reports can show information in multiple ways. For example, analysis results can be displayed as the actual data that the results refer to, or, they can be shown in a graph or chart. Graphs and charts display general information about an object such as the percentage of columns that have been analyzed in a data source. Graphs and charts also highlight issues that might otherwise be difficult to locate in the text of a standard report.

You can create and view reports in the IBM Information Server Web console and the IBM Information Server console. Both environments provide access to a number of predefined parameters and templates that you can use to generate reports.

- In the IBM Information Server Web console, you can create reports and associate reports with a project as follows:
  - Create, run, and view a report without having to save the report in the repository
  - Filter through projects to associate the report with a specific project
- In the Web console, you can create a report, configure multiple aspects of the report, and complete other reporting tasks as follows:
  - Schedule a report to run at a specific time
  - Configure a report to maintain a history of results
  - Schedule a report to be removed automatically at a specific time
  - Create reports in folders that you can name, modify, and delete
  - Create a business logo in the report
  - Configure security options for the report
  - Configure details of output types

The following categories of reports are available:

- Baseline Analysis
- Column Classification
- Column Domain
- Column Frequency
- Column Inferred
- Column Properties
- Column Summary
- Cross Table Domain Analysis
- Domain Quality Summary
- Foreign Key Analysis
- Table Primary Key Analysis

> **Note:** For complete details of all the available reports, refer to *IBM WebSphere Information Analyzer Version 8.0.1 IBM WebSphere Information Analyzer User Guide*, SC18-9902.

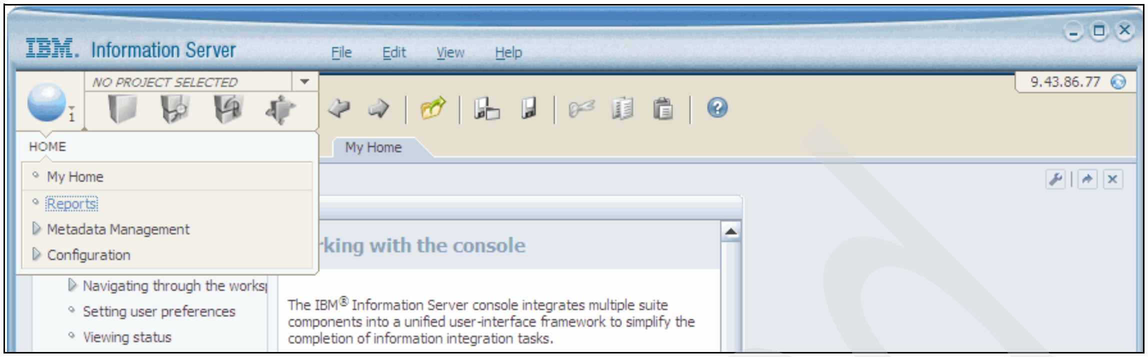
In the following sections, we describe briefly the process for generating a report, and show a portion of some of the reports using the sample database described in Figure 1-11 on page 31.

1.14.1 Generate a report

To generate a report from the IBM Information Server console, proceed as follows:

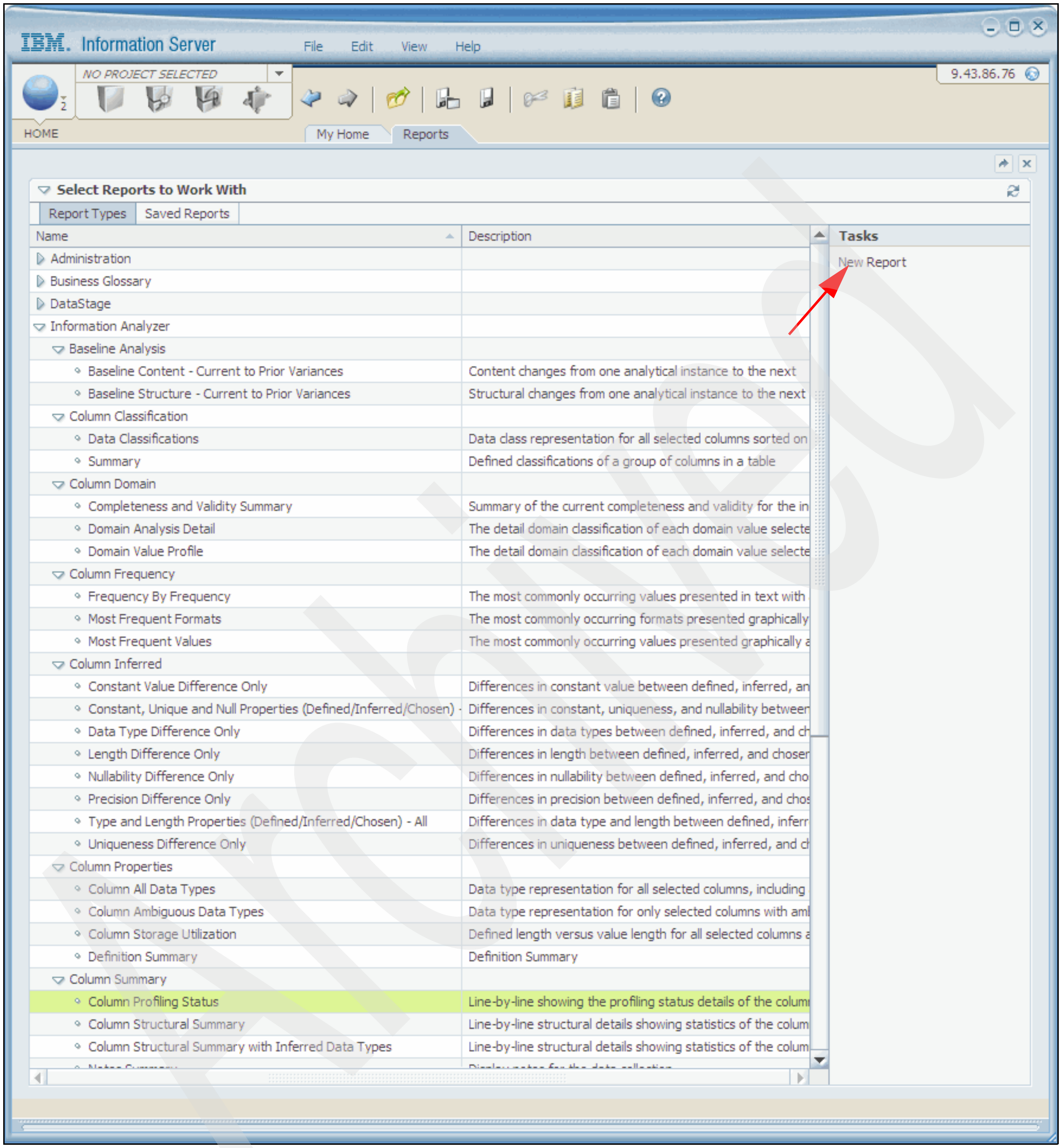
1. On the Home navigator menu in the console, select **Reports** as shown in Figure 1-333 on page 396.
1. Expand Information Analyzer in the Reports workspace under the Report Types tab to view the various reports available for generation as shown in Figure 1-334 on page 397 and Figure 1-335 on page 398.
2. Select the report Column Profiling Status in Column Summary in the Name pane and click **New Report** as shown in Figure 1-336 on page 399.
3. Select the sources of interest (EMP_TRAVEL_MILEAGE) as shown in Figure 1-336 on page 399. Click **Next**.
4. Specify the report parameters in Figure 1-337 on page 400. Supply the Description (Column Profiling Status) and click **Next**.
5. Specify the name and output, Output Format (PDF), as shown in Figure 1-338 on page 401 and click **Finish** to submit the job.
6. View the progress of the job (Column Profiling Status 5) and when it completes, click **View Results** as shown in Figure 1-339 on page 402.

Figure 1-340 on page 403 shows the results as PDF output.



The IBM® Information Server console integrates multiple suite components into a unified user-interface framework to simplify the completion of information integration tasks.

Figure 1-333 Generate a report 1/8



Select Reports to Work With

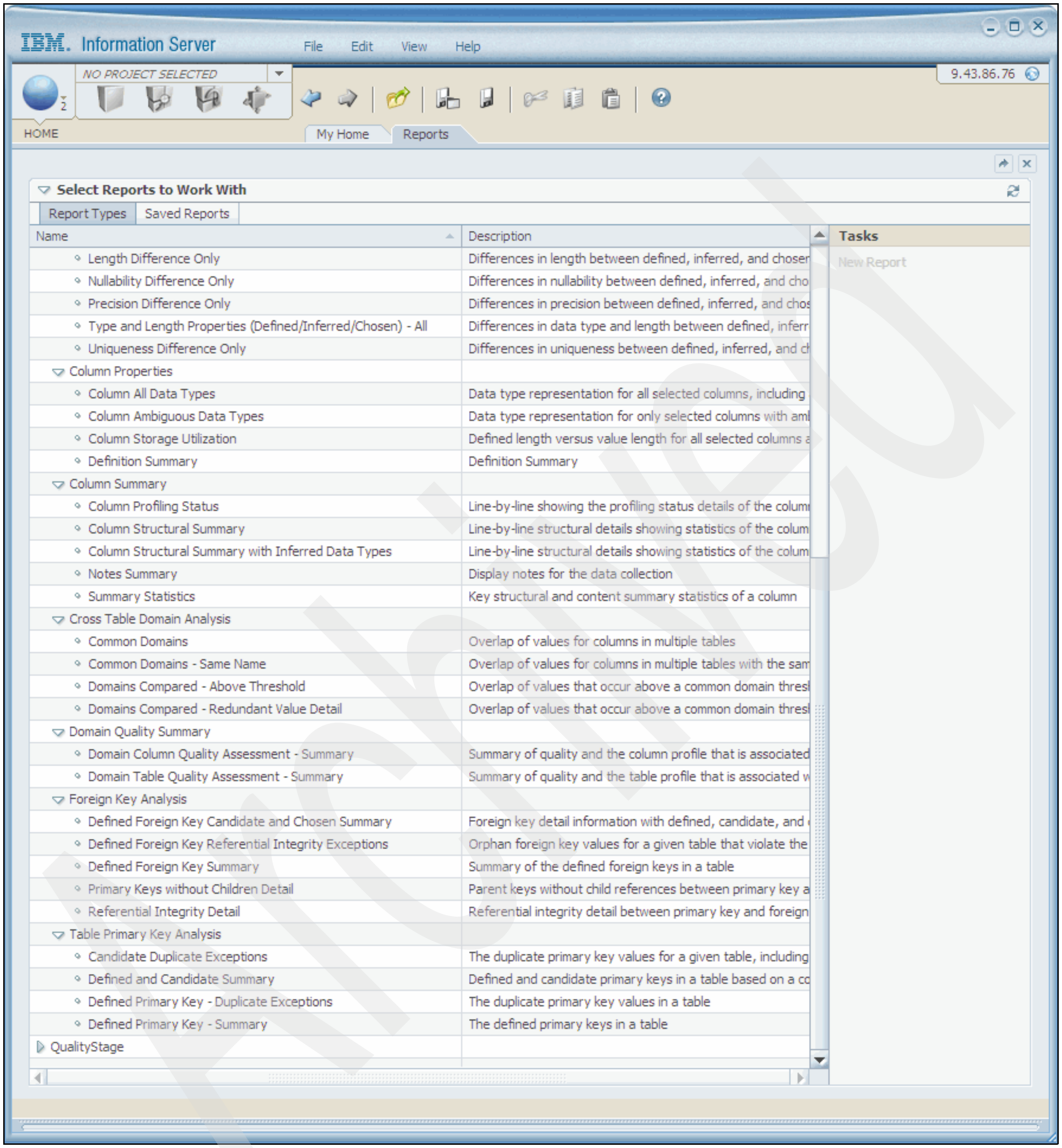
Report Types | Saved Reports

Name	Description
Administration	
Business Glossary	
DataStage	
Information Analyzer	
Baseline Analysis	
Baseline Content - Current to Prior Variances	Content changes from one analytical instance to the next
Baseline Structure - Current to Prior Variances	Structural changes from one analytical instance to the next
Column Classification	
Data Classifications	Data class representation for all selected columns sorted on
Summary	Defined classifications of a group of columns in a table
Column Domain	
Completeness and Validity Summary	Summary of the current completeness and validity for the in
Domain Analysis Detail	The detail domain classification of each domain value selecte
Domain Value Profile	The detail domain classification of each domain value selecte
Column Frequency	
Frequency By Frequency	The most commonly occurring values presented in text with
Most Frequent Formats	The most commonly occurring formats presented graphically
Most Frequent Values	The most commonly occurring values presented graphically a
Column Inferred	
Constant Value Difference Only	Differences in constant value between defined, inferred, an
Constant, Unique and Null Properties (Defined/Inferred/Chosen) -	Differences in constant, uniqueness, and nullability betweer
Data Type Difference Only	Differences in data types between defined, inferred, and ch
Length Difference Only	Differences in length between defined, inferred, and choser
Nullability Difference Only	Differences in nullability between defined, inferred, and cho
Precision Difference Only	Differences in precision between defined, inferred, and chos
Type and Length Properties (Defined/Inferred/Chosen) - All	Differences in data type and length between defined, inferr
Uniqueness Difference Only	Differences in uniqueness between defined, inferred, and cl
Column Properties	
Column All Data Types	Data type representation for all selected columns, including
Column Ambiguous Data Types	Data type representation for only selected columns with aml
Column Storage Utilization	Defined length versus value length for all selected columns a
Definition Summary	Definition Summary
Column Summary	
Column Profiling Status	Line-by-line showing the profiling status details of the colum
Column Structural Summary	Line-by-line structural details showing statistics of the colum
Column Structural Summary with Inferred Data Types	Line-by-line structural details showing statistics of the colum

Tasks

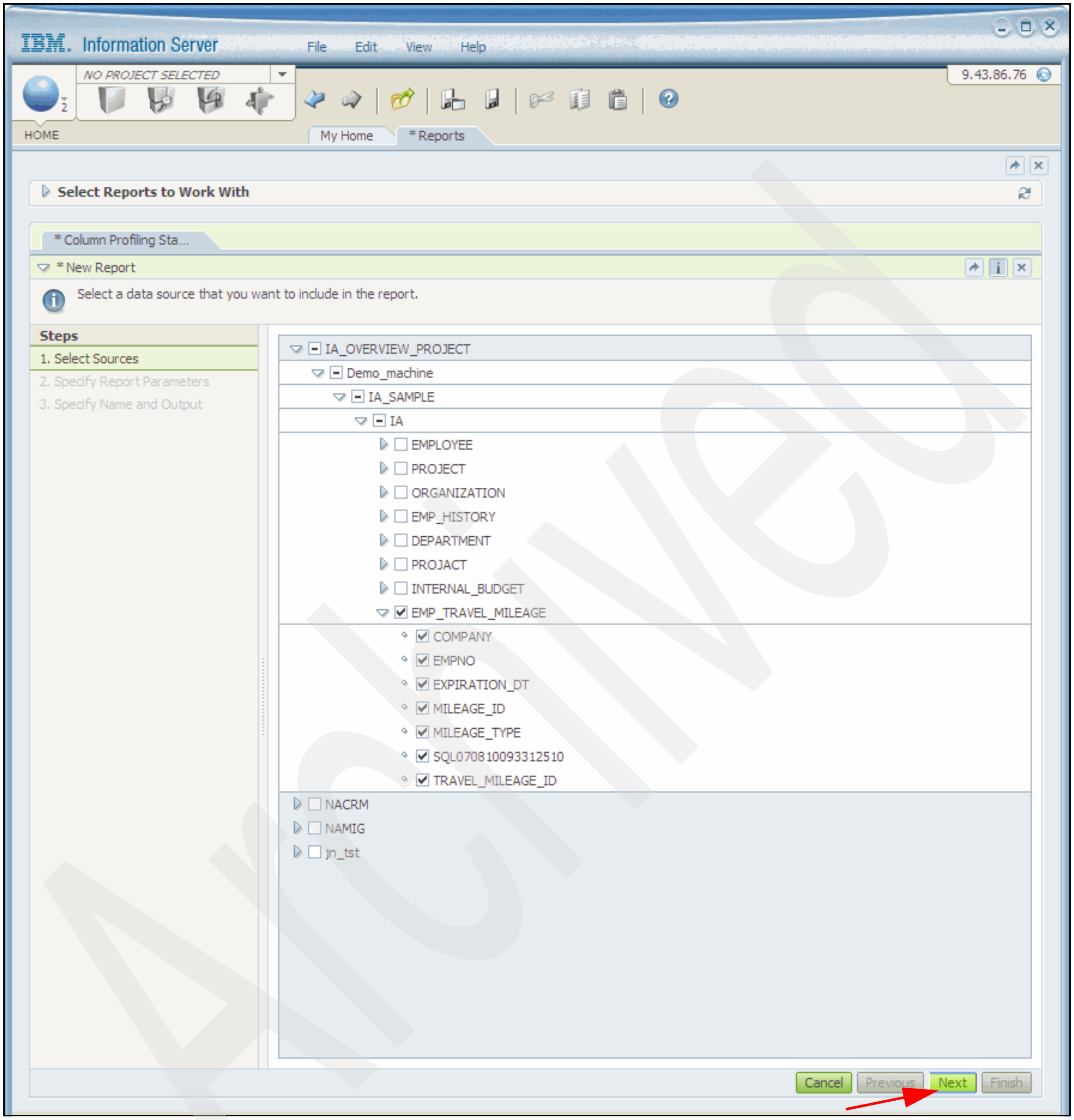
New Report

Figure 1-334 Generate a report 2/8



Name	Description
Length Difference Only	Differences in length between defined, inferred, and chosen
Nullability Difference Only	Differences in nullability between defined, inferred, and cho
Precision Difference Only	Differences in precision between defined, inferred, and chos
Type and Length Properties (Defined/Inferred/Chosen) - All	Differences in data type and length between defined, inferr
Uniqueness Difference Only	Differences in uniqueness between defined, inferred, and ch
Column Properties	
Column All Data Types	Data type representation for all selected columns, including
Column Ambiguous Data Types	Data type representation for only selected columns with aml
Column Storage Utilization	Defined length versus value length for all selected columns a
Definition Summary	Definition Summary
Column Summary	
Column Profiling Status	Line-by-line showing the profiling status details of the colum
Column Structural Summary	Line-by-line structural details showing statistics of the colum
Column Structural Summary with Inferred Data Types	Line-by-line structural details showing statistics of the colum
Notes Summary	Display notes for the data collection
Summary Statistics	Key structural and content summary statistics of a column
Cross Table Domain Analysis	
Common Domains	Overlap of values for columns in multiple tables
Common Domains - Same Name	Overlap of values for columns in multiple tables with the sam
Domains Compared - Above Threshold	Overlap of values that occur above a common domain thres
Domains Compared - Redundant Value Detail	Overlap of values that occur above a common domain thres
Domain Quality Summary	
Domain Column Quality Assessment - Summary	Summary of quality and the column profile that is associated
Domain Table Quality Assessment - Summary	Summary of quality and the table profile that is associated w
Foreign Key Analysis	
Defined Foreign Key Candidate and Chosen Summary	Foreign key detail information with defined, candidate, and
Defined Foreign Key Referential Integrity Exceptions	Orphan foreign key values for a given table that violate the
Defined Foreign Key Summary	Summary of the defined foreign keys in a table
Primary Keys without Children Detail	Parent keys without child references between primary key a
Referential Integrity Detail	Referential integrity detail between primary key and foreign
Table Primary Key Analysis	
Candidate Duplicate Exceptions	The duplicate primary key values for a given table, including
Defined and Candidate Summary	Defined and candidate primary keys in a table based on a co
Defined Primary Key - Duplicate Exceptions	The duplicate primary key values in a table
Defined Primary Key - Summary	The defined primary keys in a table
QualityStage	

Figure 1-335 Generate a report 3/8

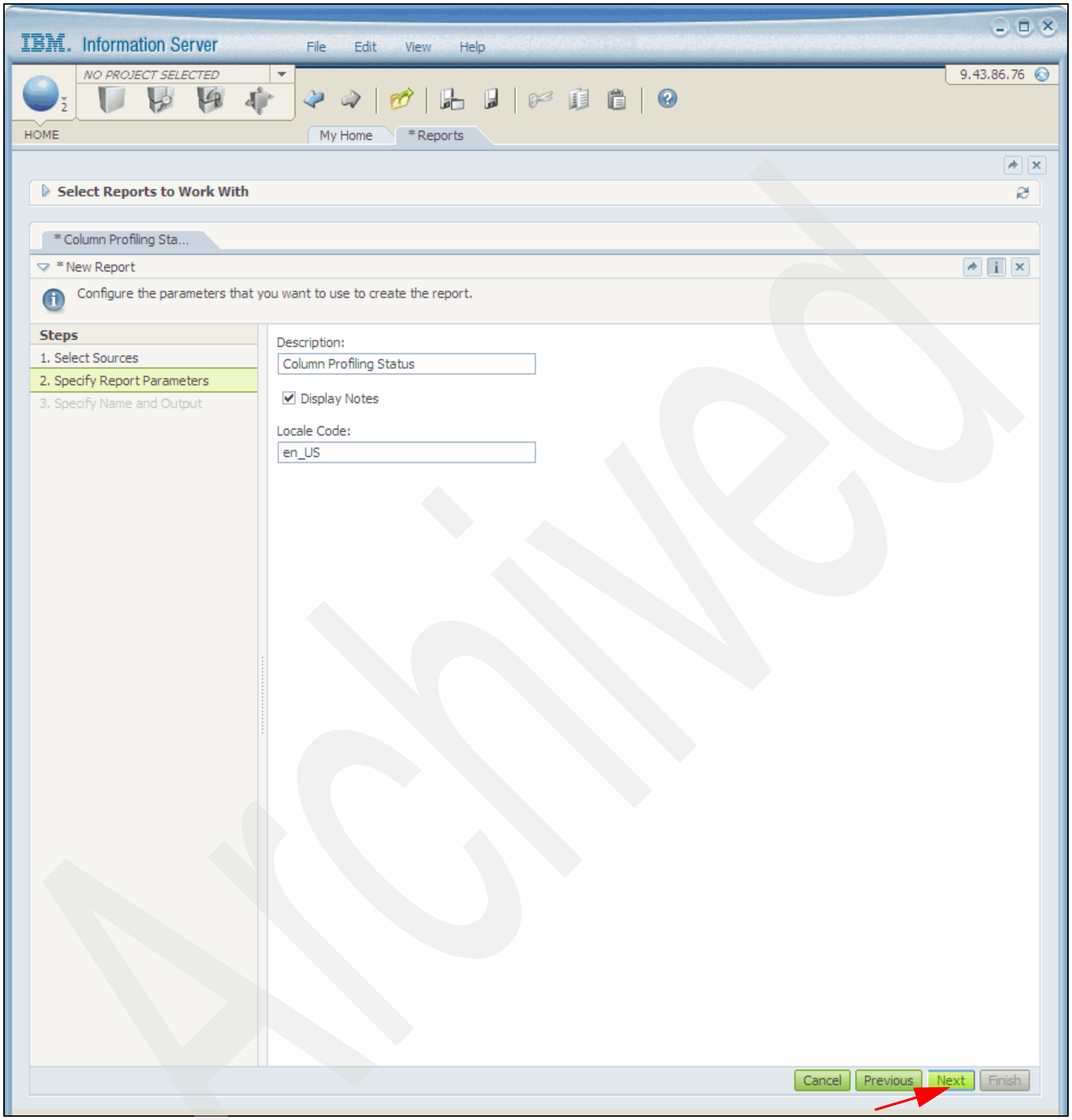


Select a data source that you want to include in the report.

1. Select Sources
2. Specify Report Parameters
3. Specify Name and Output

- IA_OVERVIEW_PROJECT
  - Demo_machine
    - IA_SAMPLE
      - IA
        - □ EMPLOYEE
        - □ PROJECT
        - □ ORGANIZATION
        - □ EMP_HISTORY
        - □ DEPARTMENT
        - □ PROJACT
        - □ INTERNAL_BUDGET
        - ☑ EMP_TRAVEL_MILEAGE
          - ☑ COMPANY
          - ☑ EMPNO
          - ☑ EXPIRATION_DT
          - ☑ MILEAGE_ID
          - ☑ MILEAGE_TYPE
          - ☑ SQL070810093312510
          - ☑ TRAVEL_MILEAGE_ID
- □ NACRM
- □ NAMIG
- □ jn_tst

Figure 1-336 Generate a report 4/8



Configure the parameters that you want to use to create the report.

Steps

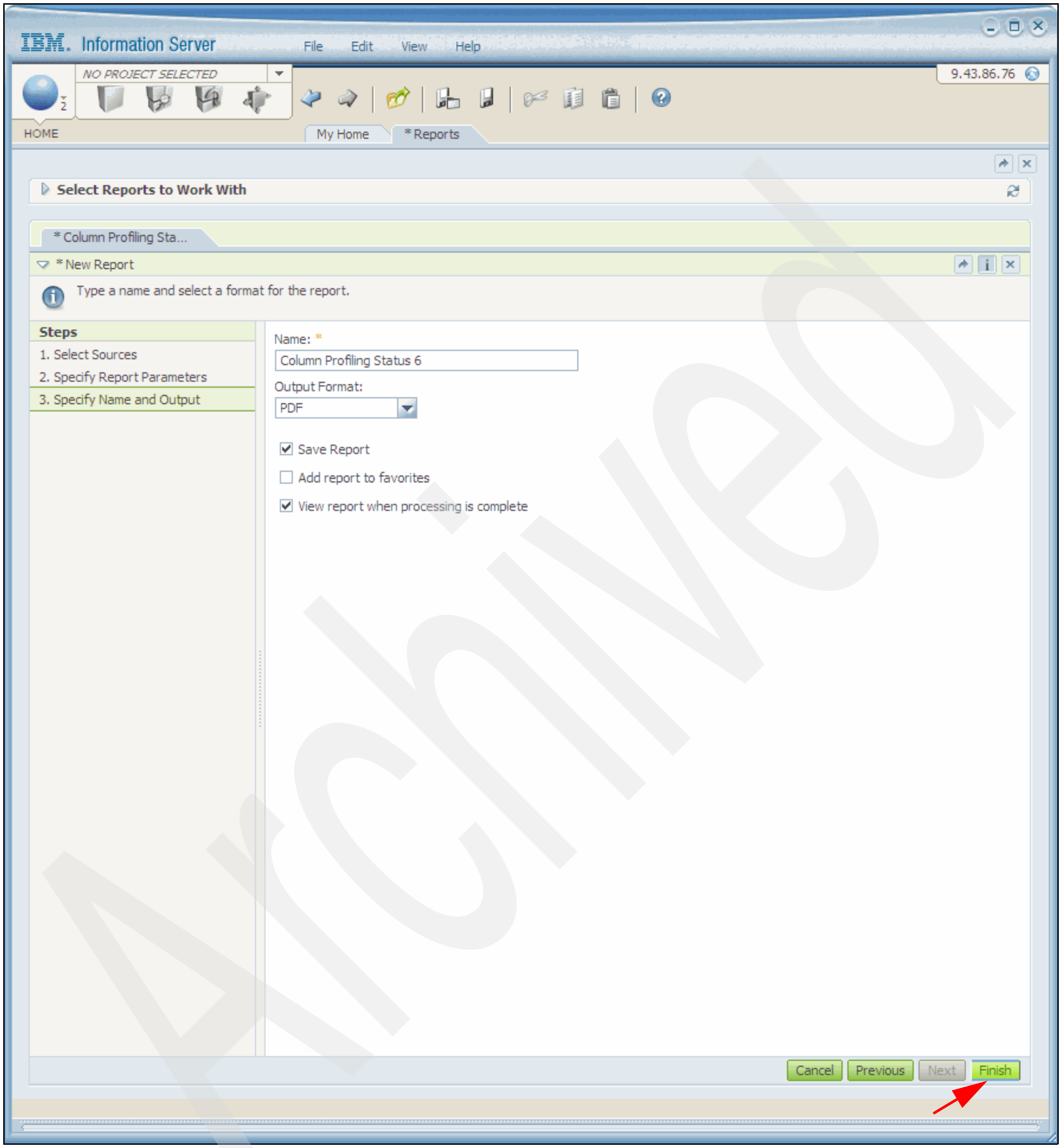
1. Select Sources
2. Specify Report Parameters
3. Specify Name and Output

Description: Column Profiling Status

☑ Display Notes

Locale Code: en_US

Figure 1-337 Generate a report 5/8



Type a name and select a format for the report.

Steps

1. Select Sources
2. Specify Report Parameters
3. Specify Name and Output

Name: * Column Profiling Status 6

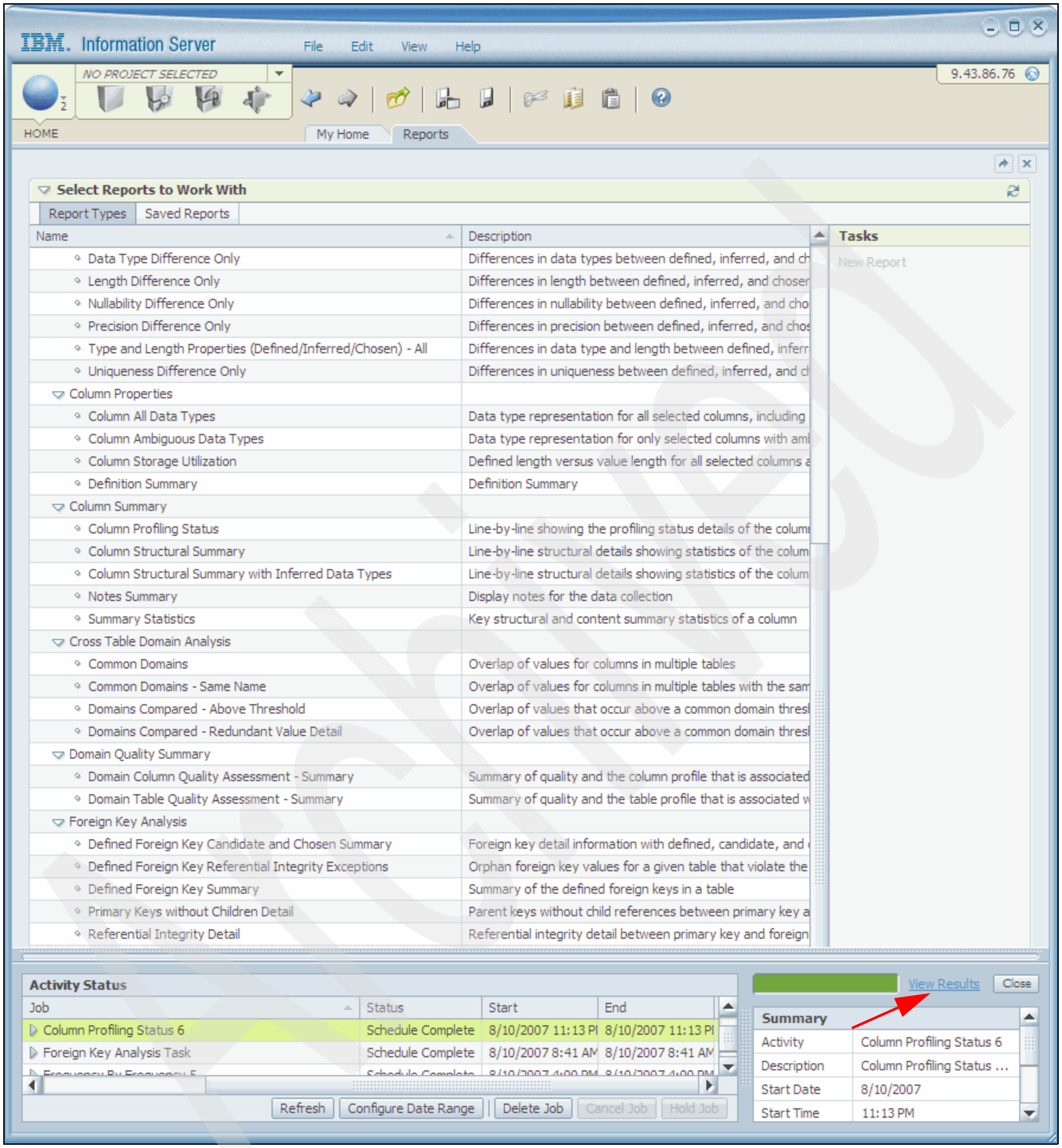
Output Format: PDF

☑ Save Report

□ Add report to favorites

☑ View report when processing is complete

Figure 1-338 Generate a report 6/8



Select Reports to Work With

Report Types | Saved Reports

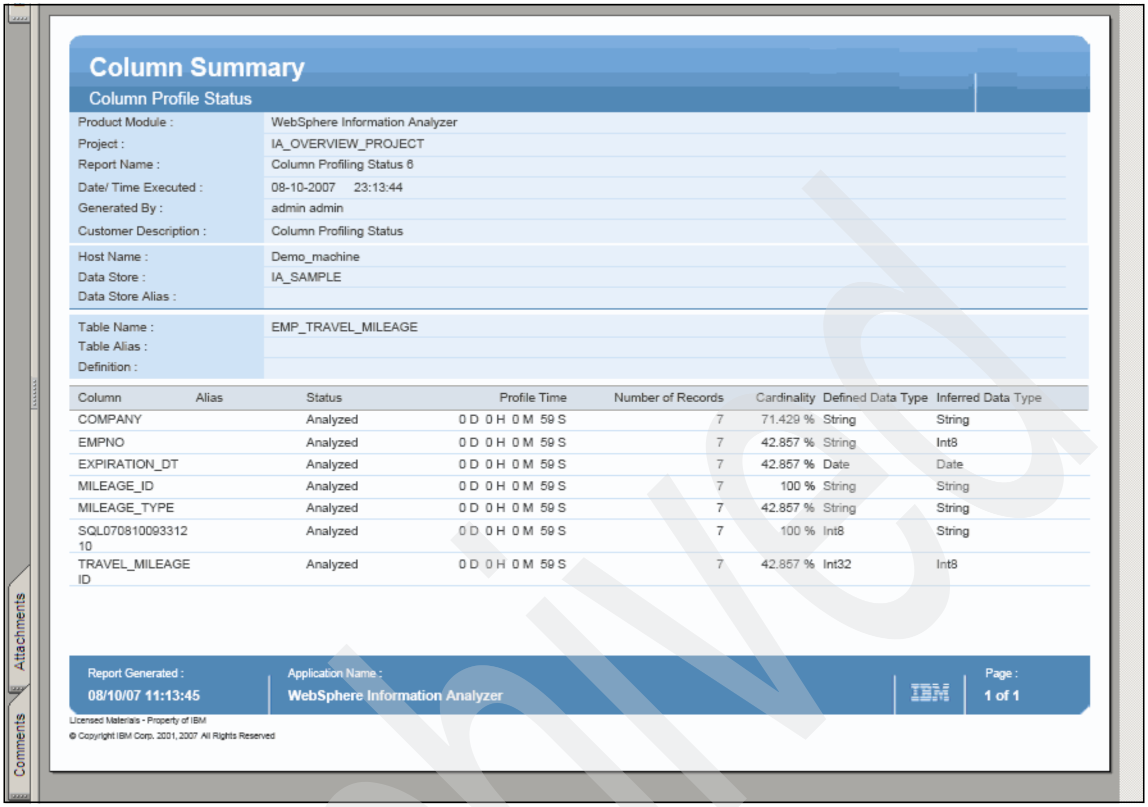
Name	Description
Data Type Difference Only	Differences in data types between defined, inferred, and ch
Length Difference Only	Differences in length between defined, inferred, and chosen
Nullability Difference Only	Differences in nullability between defined, inferred, and cho
Precision Difference Only	Differences in precision between defined, inferred, and chos
Type and Length Properties (Defined/Inferred/Chosen) - All	Differences in data type and length between defined, inferr
Uniqueness Difference Only	Differences in uniqueness between defined, inferred, and ch
Column Properties	
Column All Data Types	Data type representation for all selected columns, including
Column Ambiguous Data Types	Data type representation for only selected columns with aml
Column Storage Utilization	Defined length versus value length for all selected columns a
Definition Summary	Definition Summary
Column Summary	
Column Profiling Status	Line-by-line showing the profiling status details of the colum
Column Structural Summary	Line-by-line structural details showing statistics of the colum
Column Structural Summary with Inferred Data Types	Line-by-line structural details showing statistics of the colum
Notes Summary	Display notes for the data collection
Summary Statistics	Key structural and content summary statistics of a column
Cross Table Domain Analysis	
Common Domains	Overlap of values for columns in multiple tables
Common Domains - Same Name	Overlap of values for columns in multiple tables with the sam
Domains Compared - Above Threshold	Overlap of values that occur above a common domain thresl
Domains Compared - Redundant Value Detail	Overlap of values that occur above a common domain thresl
Domain Quality Summary	
Domain Column Quality Assessment - Summary	Summary of quality and the column profile that is associated
Domain Table Quality Assessment - Summary	Summary of quality and the table profile that is associated w
Foreign Key Analysis	
Defined Foreign Key Candidate and Chosen Summary	Foreign key detail information with defined, candidate, and
Defined Foreign Key Referential Integrity Exceptions	Orphan foreign key values for a given table that violate the
Defined Foreign Key Summary	Summary of the defined foreign keys in a table
Primary Keys without Children Detail	Parent keys without child references between primary key a
Referential Integrity Detail	Referential integrity detail between primary key and foreign

Activity Status

Job	Status	Start	End
Column Profiling Status 6	Schedule Complete	8/10/2007 11:13 PI	8/10/2007 11:13 PI
Foreign Key Analysis Task	Schedule Complete	8/10/2007 8:41 AM	8/10/2007 8:41 AM
Frequency By Frequency 5	Schedule Complete	8/10/2007 4:00 PM	8/10/2007 4:00 PM

Summary	
Activity	Column Profiling Status 6
Description	Column Profiling Status ...
Start Date	8/10/2007
Start Time	11:13 PM

Figure 1-339 Generate a report 7/8



Column Summary
Column Profile Status

Product Module :	WebSphere Information Analyzer
Project :	IA_OVERVIEW_PROJECT
Report Name :	Column Profiling Status 6
Date/ Time Executed :	08-10-2007 23:13:44
Generated By :	admin admin
Customer Description :	Column Profiling Status
Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Table Name :	EMP_TRAVEL_MILEAGE
Table Alias :	
Definition :	

Column	Alias	Status	Profile Time	Number of Records	Cardinality	Defined Data Type	Inferred Data Type
COMPANY		Analyzed	0 D 0 H 0 M 59 S	7	71.429 %	String	String
EMPNO		Analyzed	0 D 0 H 0 M 59 S	7	42.857 %	String	Int8
EXPIRATION_DT		Analyzed	0 D 0 H 0 M 59 S	7	42.857 %	Date	Date
MILEAGE_ID		Analyzed	0 D 0 H 0 M 59 S	7	100 %	String	String
MILEAGE_TYPE		Analyzed	0 D 0 H 0 M 59 S	7	42.857 %	String	String
SQL070810093312 10		Analyzed	0 D 0 H 0 M 59 S	7	100 %	Int8	String
TRAVEL_MILEAGE ID		Analyzed	0 D 0 H 0 M 59 S	7	42.857 %	Int32	Int8

Report Generated : 08/10/07 11:13:45
Application Name : WebSphere Information Analyzer
IBM
Page : 1 of 1

Licensed Materials - Property of IBM
© Copyright IBM Corp. 2001, 2007 All Rights Reserved

Figure 1-340 Generate a report 8/8

1.14.2 Sample reports

This section shows portions of the following reports that we generated for some of the tables in the IA_SAMPLE_PROJECT:

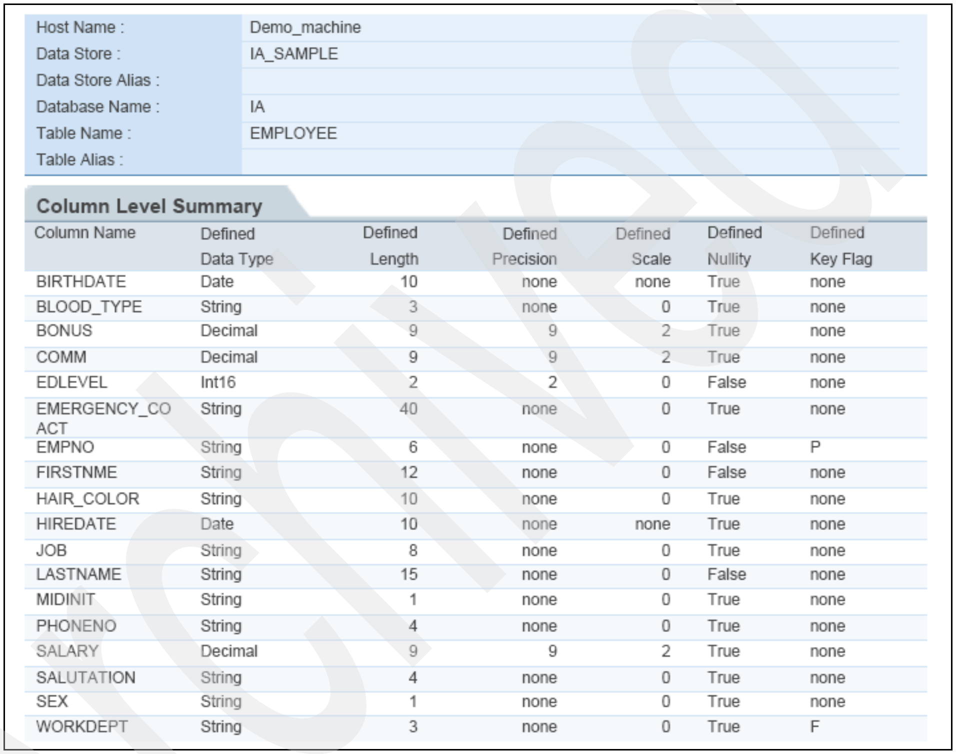
- Column level
- Primary key
- Domain quality
- Cross domain
- Foreign key

Column level

A number of column level reports are available as shown in Figure 1-334 on page 397 and Figure 1-335 on page 398. We show portions of these reports here.

- Column definitions

  Figure 1-341 shows a summary of the column definitions for the EMPLOYEE table.



Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA
Table Name :	EMPLOYEE
Table Alias :	

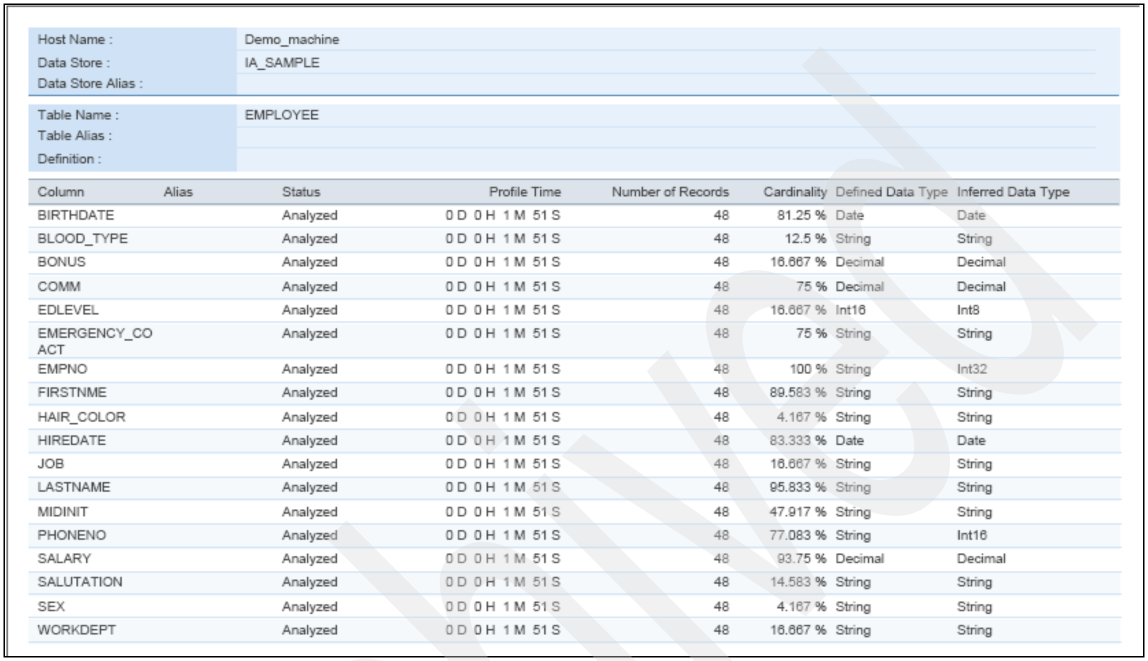
Column Level Summary

Column Name	Defined Data Type	Defined Length	Defined Precision	Defined Scale	Defined Nullity	Defined Key Flag
BIRTHDATE	Date	10	none	none	True	none
BLOOD_TYPE	String	3	none	0	True	none
BONUS	Decimal	9	9	2	True	none
COMM	Decimal	9	9	2	True	none
EDLEVEL	Int16	2	2	0	False	none
EMERGENCY_CO ACT	String	40	none	0	True	none
EMPNO	String	6	none	0	False	P
FIRSTNME	String	12	none	0	False	none
HAIR_COLOR	String	10	none	0	True	none
HIREDATE	Date	10	none	none	True	none
JOB	String	8	none	0	True	none
LASTNAME	String	15	none	0	False	none
MIDINIT	String	1	none	0	True	none
PHONENO	String	4	none	0	True	none
SALARY	Decimal	9	9	2	True	none
SALUTATION	String	4	none	0	True	none
SEX	String	1	none	0	True	none
WORKDEPT	String	3	none	0	True	F

Figure 1-341 Column definitions for EMPLOYEE

- Column profiling summary

  Figure 1-342 shows a column-wise summary of the profile analysis for the EMPLOYEE table.



Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	

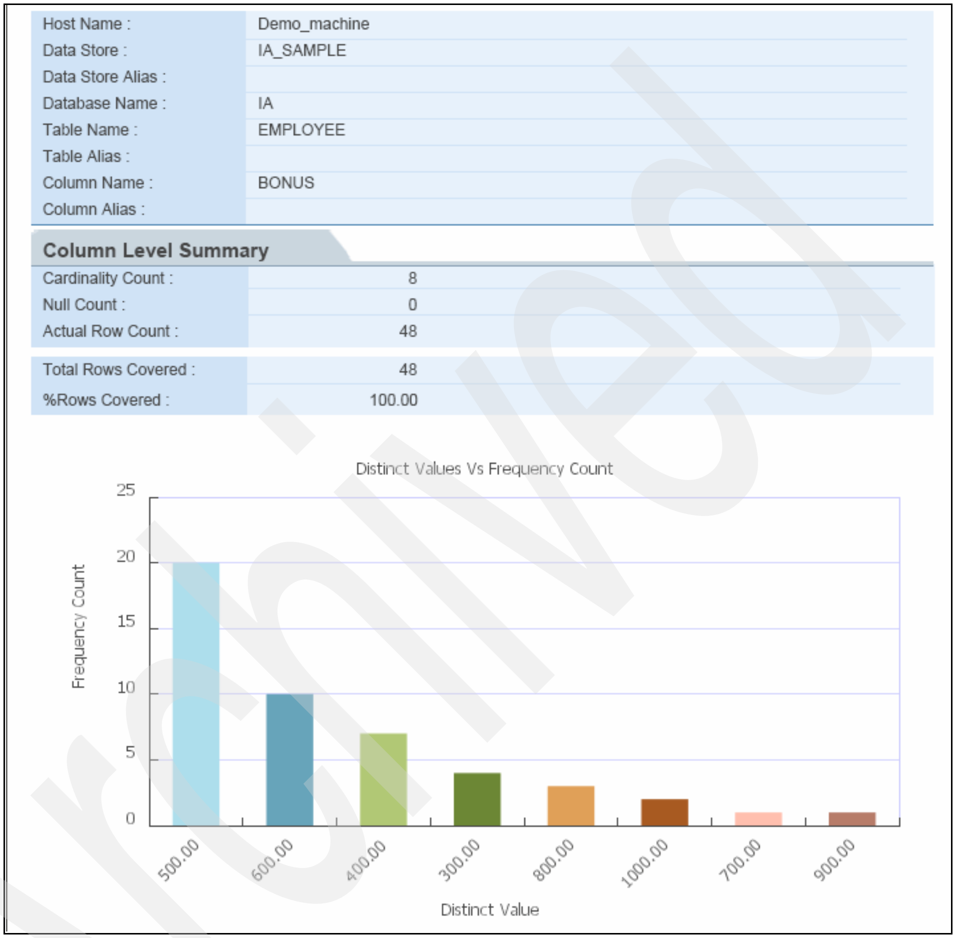
Table Name :	EMPLOYEE
Table Alias :	
Definition :	

Column	Alias	Status	Profile Time	Number of Records	Cardinality	Defined Data Type	Inferred Data Type
BIRTHDATE		Analyzed	0 D 0 H 1 M 51 S	48	81.25 %	Date	Date
BLOOD_TYPE		Analyzed	0 D 0 H 1 M 51 S	48	12.5 %	String	String
BONUS		Analyzed	0 D 0 H 1 M 51 S	48	16.667 %	Decimal	Decimal
COMM		Analyzed	0 D 0 H 1 M 51 S	48	75 %	Decimal	Decimal
EDLEVEL		Analyzed	0 D 0 H 1 M 51 S	48	16.667 %	Int16	Int8
EMERGENCY_COACT		Analyzed	0 D 0 H 1 M 51 S	48	75 %	String	String
EMPNO		Analyzed	0 D 0 H 1 M 51 S	48	100 %	String	Int32
FIRSTNME		Analyzed	0 D 0 H 1 M 51 S	48	89.583 %	String	String
HAIR_COLOR		Analyzed	0 D 0 H 1 M 51 S	48	4.167 %	String	String
HIREDATE		Analyzed	0 D 0 H 1 M 51 S	48	83.333 %	Date	Date
JOB		Analyzed	0 D 0 H 1 M 51 S	48	16.667 %	String	String
LASTNAME		Analyzed	0 D 0 H 1 M 51 S	48	95.833 %	String	String
MIDINIT		Analyzed	0 D 0 H 1 M 51 S	48	47.917 %	String	String
PHONENO		Analyzed	0 D 0 H 1 M 51 S	48	77.083 %	String	Int16
SALARY		Analyzed	0 D 0 H 1 M 51 S	48	93.75 %	Decimal	Decimal
SALUTATION		Analyzed	0 D 0 H 1 M 51 S	48	14.583 %	String	String
SEX		Analyzed	0 D 0 H 1 M 51 S	48	4.167 %	String	String
WORKDEPT		Analyzed	0 D 0 H 1 M 51 S	48	16.667 %	String	String

Figure 1-342 Column profiling summary for EMPLOYEE

- Column level summary

  Figure 1-343 shows a column level summary of the profile of the BONUS column of the EMPLOYEE table.



Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA
Table Name :	EMPLOYEE
Table Alias :	
Column Name :	BONUS
Column Alias :	

Column Level Summary

Cardinality Count :	8
Null Count :	0
Actual Row Count :	48

Total Rows Covered :	48
%Rows Covered :	100.00

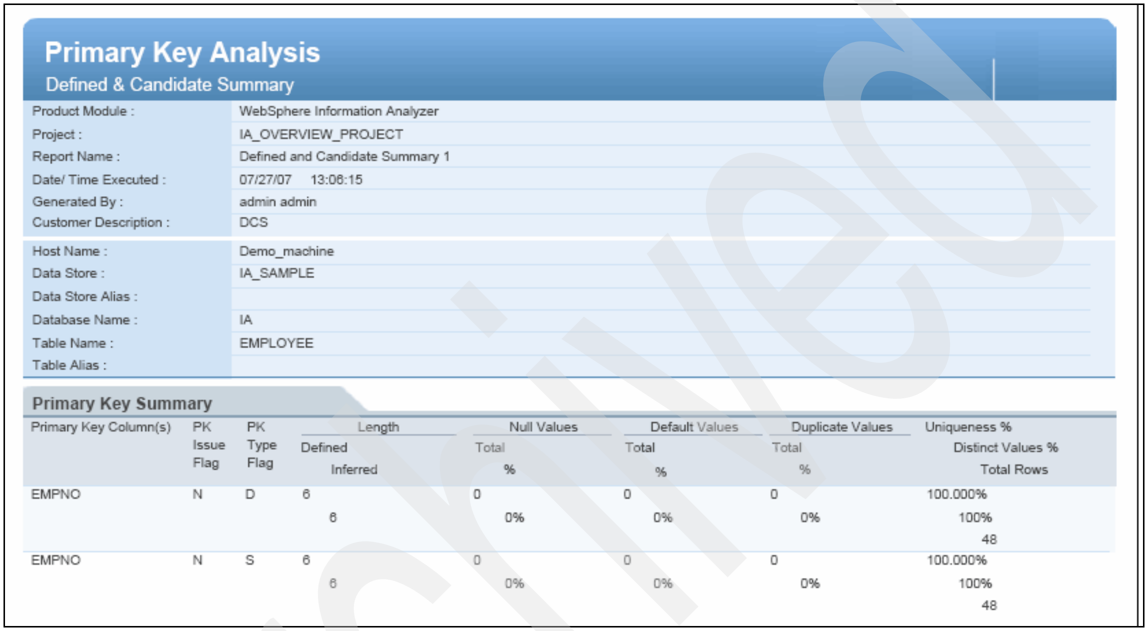
Figure 1-343 Column level summary for BONUS column in EMPLOYEE

Primary key

A number of primary key reports are available as shown in Figure 1-334 on page 397 and Figure 1-335 on page 398. We show portions of these here.

- Defined and candidate primary keys

  Figure 1-344 and Figure 1-345 show the defined and candidate primary keys for the EMPLOYEE table.



Primary Key Analysis

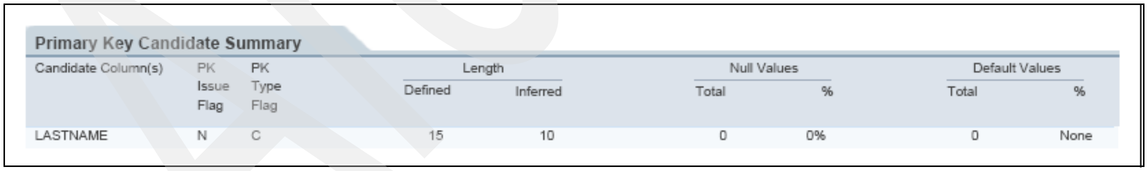
Defined & Candidate Summary

Product Module :	WebSphere Information Analyzer
Project :	IA_OVERVIEW_PROJECT
Report Name :	Defined and Candidate Summary 1
Date/ Time Executed :	07/27/07 13:06:15
Generated By :	admin admin
Customer Description :	DCS
Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA
Table Name :	EMPLOYEE
Table Alias :	

Primary Key Summary

Primary Key Column(s)	PK Issue Flag	PK Type Flag	Length Defined	Length Inferred	Null Values Total	Null Values %	Default Values Total	Default Values %	Duplicate Values Total	Duplicate Values %	Uniqueness %	Distinct Values %	Total Rows
EMPNO	N	D	6	6	0	0%	0	0%	0	0%	100.000%	100%	48
EMPNO	N	S	6	6	0	0%	0	0%	0	0%	100.000%	100%	48

Figure 1-344 Defined and candidate primary keys 1/2



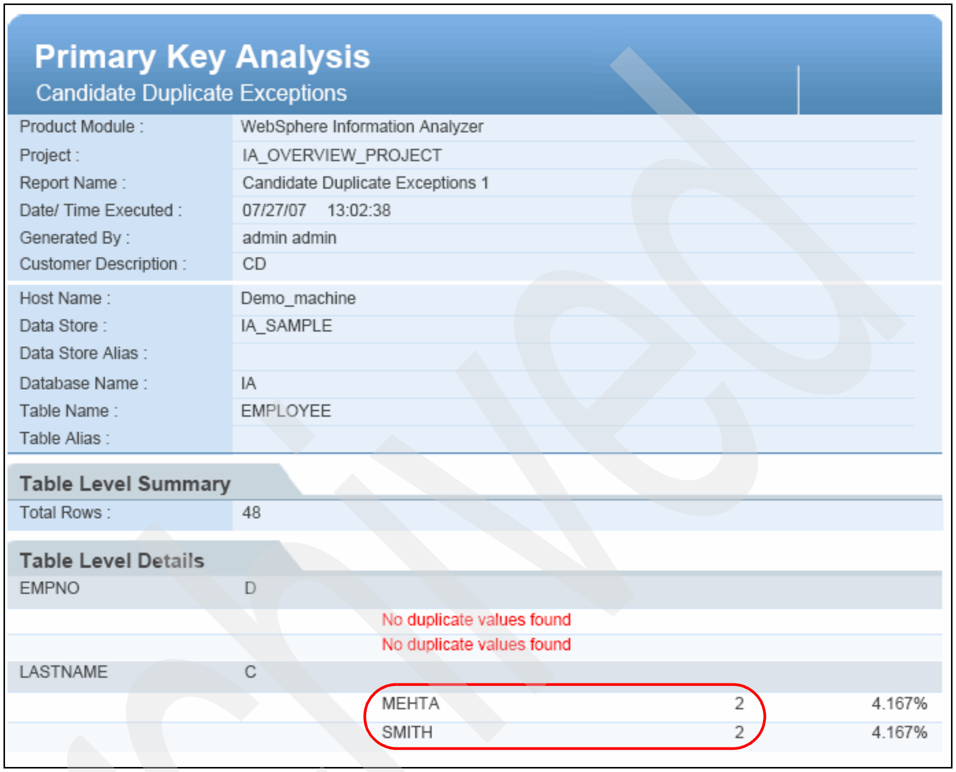
Primary Key Candidate Summary

Candidate Column(s)	PK Issue Flag	PK Type Flag	Length Defined	Length Inferred	Null Values Total	Null Values %	Default Values Total	Default Values %
LASTNAME	N	C	15	10	0	0%	0	None

Figure 1-345 Defined and candidate primary keys 122

- Candidate duplicate exceptions

  Figure 1-346 shows the duplicate exceptions in the defined and candidate primary keys of the EMPLOYEE table.



Primary Key Analysis

Candidate Duplicate Exceptions

Product Module :	WebSphere Information Analyzer
Project :	IA_OVERVIEW_PROJECT
Report Name :	Candidate Duplicate Exceptions 1
Date/ Time Executed :	07/27/07 13:02:38
Generated By :	admin admin
Customer Description :	CD
Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA
Table Name :	EMPLOYEE
Table Alias :	

Table Level Summary

Total Rows :	48

Table Level Details

EMPNO	D			
		No duplicate values found		
		No duplicate values found		
LASTNAME	C			
		MEHTA	2	4.167%
		SMITH	2	4.167%

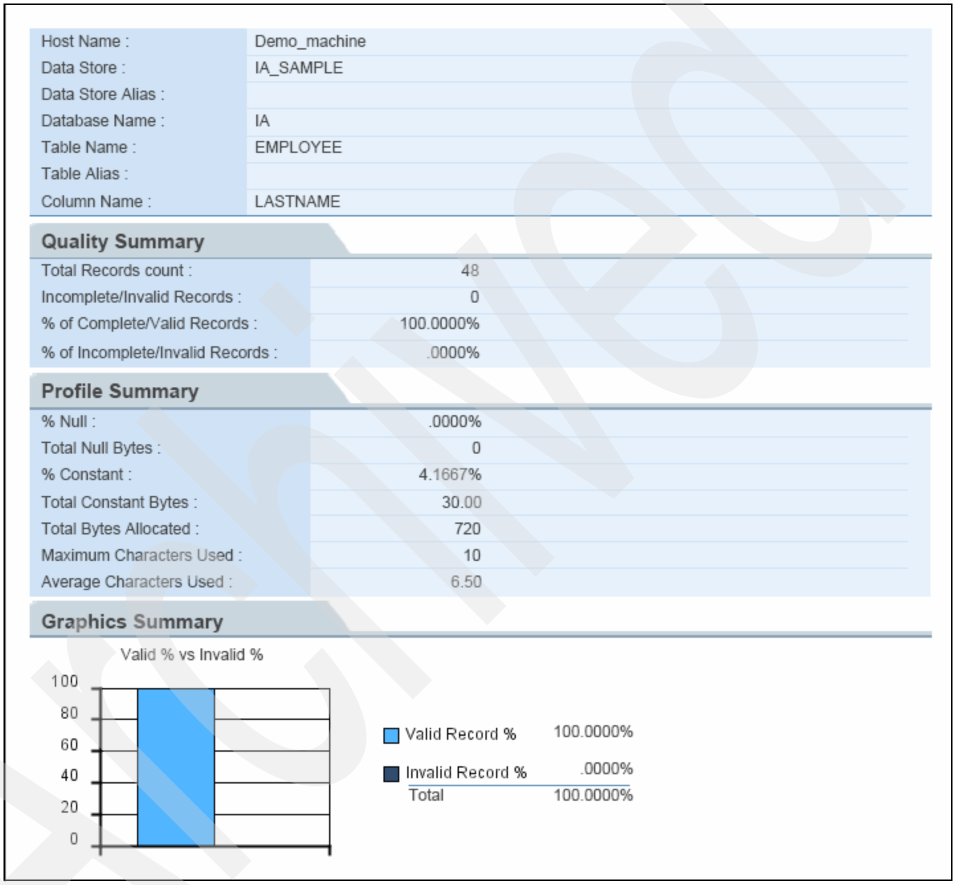
Figure 1-346 Candidate duplication exceptions

Domain quality

A number of domain quality reports are available as shown in Figure 1-334 on page 397 and Figure 1-335 on page 398. We show portions of these here.

- Domain column quality

  Figure 1-347 shows the domain column quality summary for the LASTNAME column of the EMPLOYEE table.



Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA
Table Name :	EMPLOYEE
Table Alias :	
Column Name :	LASTNAME

Quality Summary

Total Records count :	48
Incomplete/Invalid Records :	0
% of Complete/Valid Records :	100.0000%
% of Incomplete/Invalid Records :	.0000%

Profile Summary

% Null :	.0000%
Total Null Bytes :	0
% Constant :	4.1667%
Total Constant Bytes :	30.00
Total Bytes Allocated :	720
Maximum Characters Used :	10
Average Characters Used :	6.50

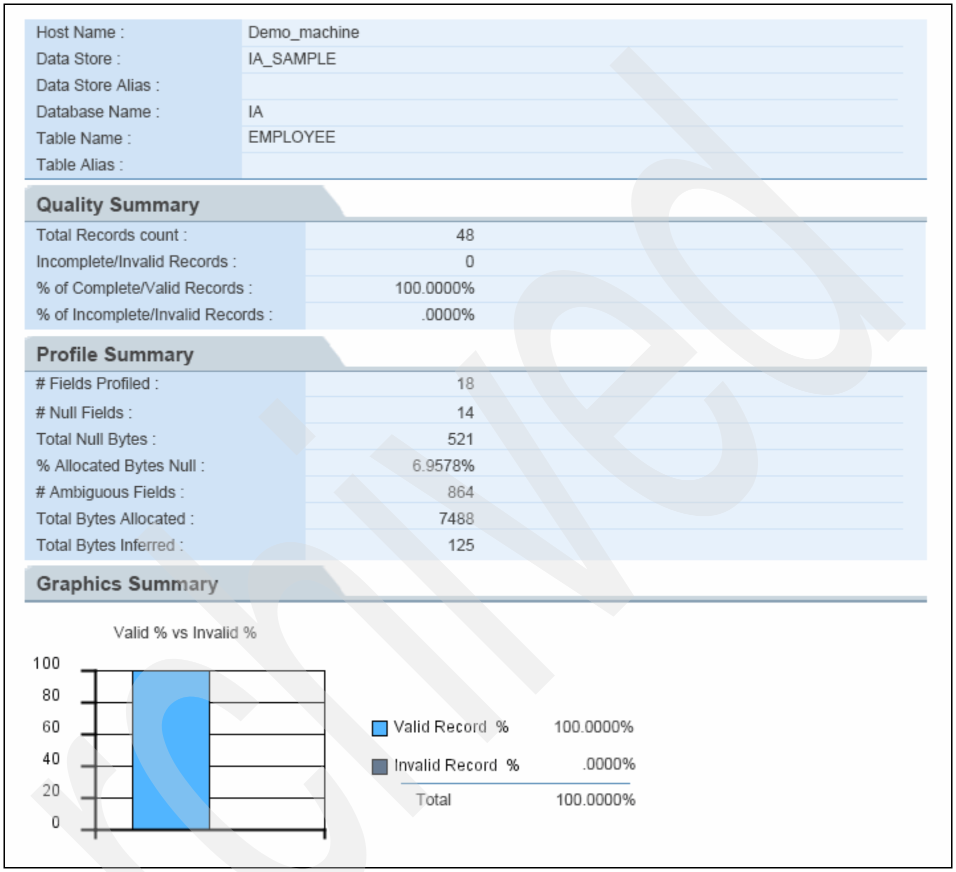
Graphics Summary

🟦	Valid Record %	100.0000%
■	Invalid Record %	.0000%
	Total	100.0000%

Figure 1-347 Domain column quality

- Domain table quality

  Figure 1-348 shows the domain table quality summary for the EMPLOYEE table.



Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA
Table Name :	EMPLOYEE
Table Alias :	

Quality Summary

Total Records count :	48
Incomplete/Invalid Records :	0
% of Complete/Valid Records :	100.0000%
% of Incomplete/Invalid Records :	.0000%

Profile Summary

# Fields Profiled :	18
# Null Fields :	14
Total Null Bytes :	521
% Allocated Bytes Null :	6.9578%
# Ambiguous Fields :	864
Total Bytes Allocated :	7488
Total Bytes Inferred :	125

Graphics Summary

🟦 Valid Record %	100.0000%
▪ Invalid Record %	.0000%
Total	100.0000%

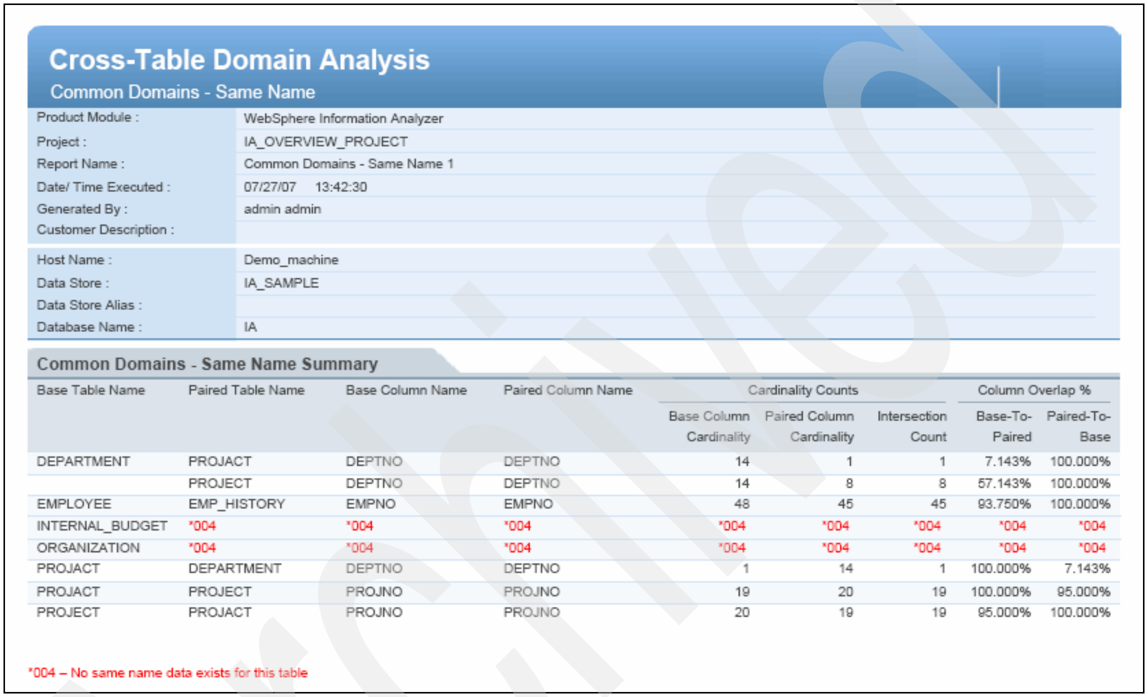
Figure 1-348 Domain table quality

Cross Domain

A number of domain quality reports are available as shown in Figure 1-334 on page 397 and Figure 1-335 on page 398. We show portions of these here.

- Common domains - same name

  Figure 1-349 shows the common domains in the IA_OVERVIEW_PROJECT that have the same name.



Cross-Table Domain Analysis

Common Domains - Same Name

Product Module :	WebSphere Information Analyzer
Project :	IA_OVERVIEW_PROJECT
Report Name :	Common Domains - Same Name 1
Date/ Time Executed :	07/27/07 13:42:30
Generated By :	admin admin
Customer Description :	
Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA

Common Domains - Same Name Summary

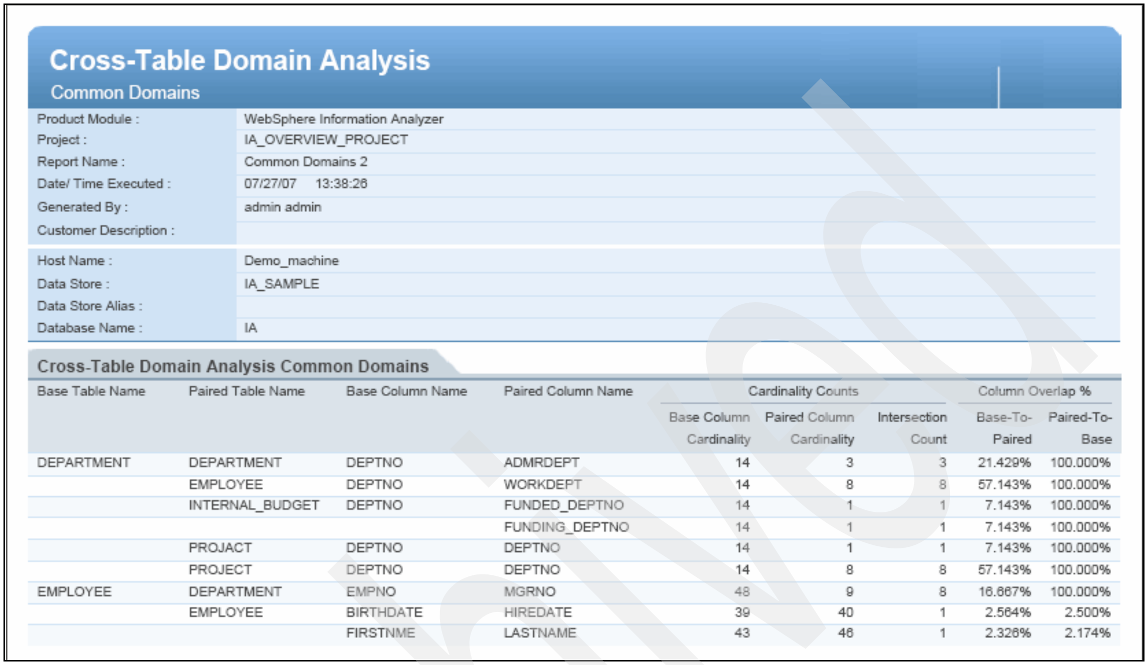
Base Table Name	Paired Table Name	Base Column Name	Paired Column Name	Cardinality Counts			Column Overlap %	
				Base Column Cardinality	Paired Column Cardinality	Intersection Count	Base-To-Paired	Paired-To-Base
DEPARTMENT	PROJACT	DEPTNO	DEPTNO	14	1	1	7.143%	100.000%
	PROJECT	DEPTNO	DEPTNO	14	8	8	57.143%	100.000%
EMPLOYEE	EMP_HISTORY	EMPNO	EMPNO	48	45	45	93.750%	100.000%
INTERNAL_BUDGET	*004	*004	*004	*004	*004	*004	*004	*004
ORGANIZATION	*004	*004	*004	*004	*004	*004	*004	*004
PROJACT	DEPARTMENT	DEPTNO	DEPTNO	1	14	1	100.000%	7.143%
PROJACT	PROJECT	PROJNO	PROJNO	19	20	19	100.000%	95.000%
PROJECT	PROJACT	PROJNO	PROJNO	20	19	19	95.000%	100.000%

*004 – No same name data exists for this table

Figure 1-349 Common domains - same name

- Common domains

  Figure 1-350 shows the common domains in the IA_OVERVIEW_PROJECT.



Cross-Table Domain Analysis

Common Domains

Product Module :	WebSphere Information Analyzer
Project :	IA_OVERVIEW_PROJECT
Report Name :	Common Domains 2
Date/ Time Executed :	07/27/07 13:38:28
Generated By :	admin admin
Customer Description :	

Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database Name :	IA

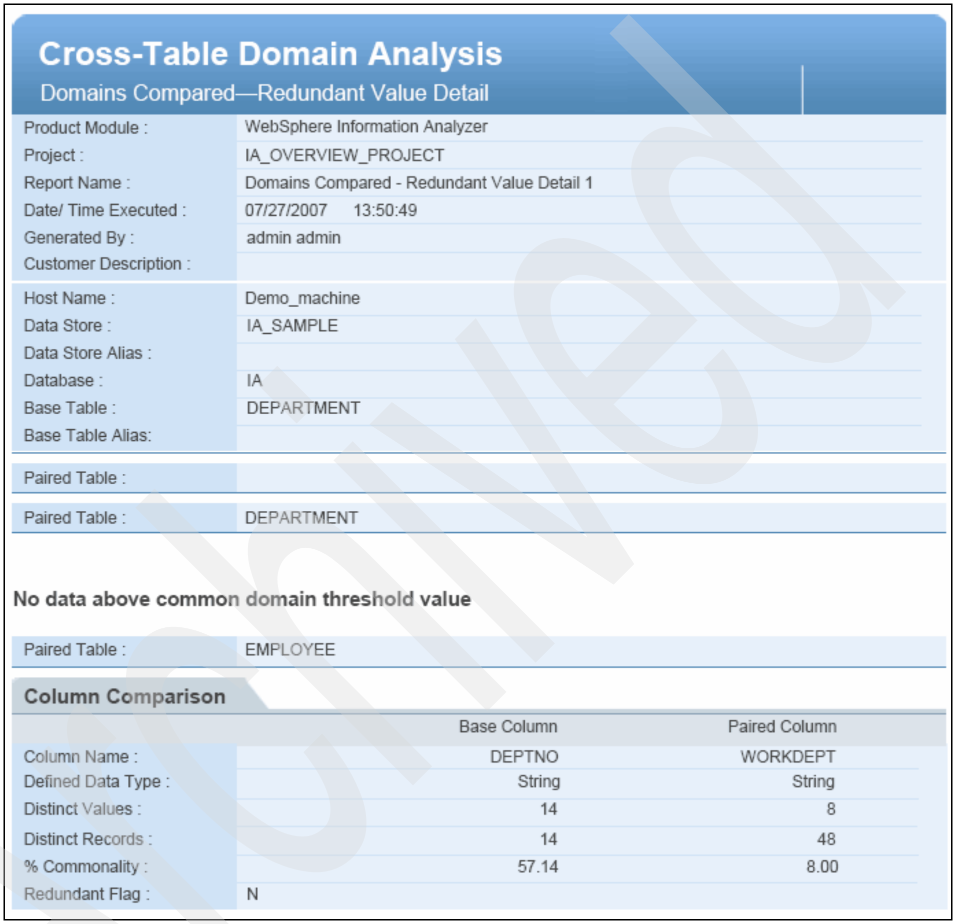
Cross-Table Domain Analysis Common Domains

Base Table Name	Paired Table Name	Base Column Name	Paired Column Name	Cardinality Counts			Column Overlap %	
				Base Column Cardinality	Paired Column Cardinality	Intersection Count	Base-To-Paired	Paired-To-Base
DEPARTMENT	DEPARTMENT	DEPTNO	ADMRDEPT	14	3	3	21.429%	100.000%
	EMPLOYEE	DEPTNO	WORKDEPT	14	8	8	57.143%	100.000%
	INTERNAL_BUDGET	DEPTNO	FUNDED_DEPTNO	14	1	1	7.143%	100.000%
			FUNDING_DEPTNO	14	1	1	7.143%	100.000%
	PROJACT	DEPTNO	DEPTNO	14	1	1	7.143%	100.000%
	PROJECT	DEPTNO	DEPTNO	14	8	8	57.143%	100.000%
EMPLOYEE	DEPARTMENT	EMPNO	MGRNO	48	9	8	16.667%	100.000%
	EMPLOYEE	BIRTHDATE	HIREDATE	39	40	1	2.564%	2.500%
		FIRSTNME	LASTNAME	43	46	1	2.326%	2.174%

Figure 1-350 Common domains

- Domains compared - redundant value detail

  Figure 1-351 shows the domain comparison of the DEPTNO column in the DEPARTMENT table and the WORKDEPT column in the EMPLOYEE table, with their commonality percentage.



Cross-Table Domain Analysis

Domains Compared—Redundant Value Detail

Product Module :	WebSphere Information Analyzer
Project :	IA_OVERVIEW_PROJECT
Report Name :	Domains Compared - Redundant Value Detail 1
Date/ Time Executed :	07/27/2007 13:50:49
Generated By :	admin admin
Customer Description :	

Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database :	IA
Base Table :	DEPARTMENT
Base Table Alias:	

Paired Table :	

Paired Table :	DEPARTMENT

No data above common domain threshold value

Paired Table :	EMPLOYEE

Column Comparison

	Base Column	Paired Column
Column Name :	DEPTNO	WORKDEPT
Defined Data Type :	String	String
Distinct Values :	14	8
Distinct Records :	14	48
% Commonality :	57.14	8.00
Redundant Flag :	N	

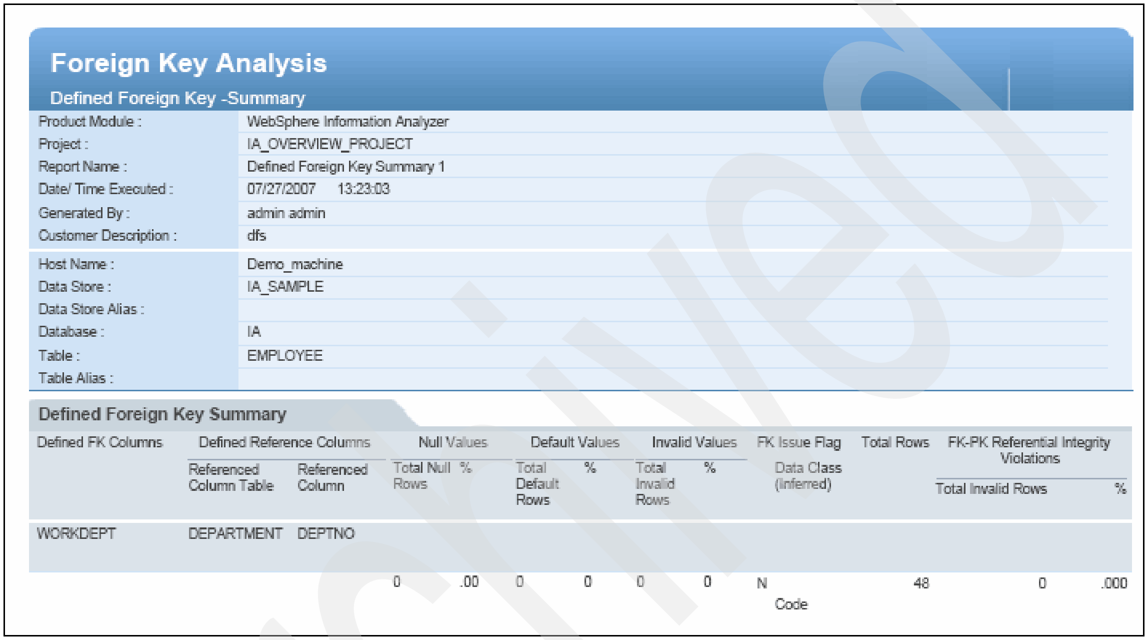
Figure 1-351 Domains compared - redundant value detail

Foreign keys

A number of foreign key reports are available as shown in Figure 1-334 on page 397 and Figure 1-335 on page 398. We show portions of these here.

- Defined foreign key summary

  Figure 1-352 shows a summary of all the defined foreign keys in the EMPLOYEE table.



Foreign Key Analysis

Defined Foreign Key -Summary

Product Module :	WebSphere Information Analyzer
Project :	IA_OVERVIEW_PROJECT
Report Name :	Defined Foreign Key Summary 1
Date/ Time Executed :	07/27/2007 13:23:03
Generated By :	admin admin
Customer Description :	dfs

Host Name :	Demo_machine
Data Store :	IA_SAMPLE
Data Store Alias :	
Database :	IA
Table :	EMPLOYEE
Table Alias :	

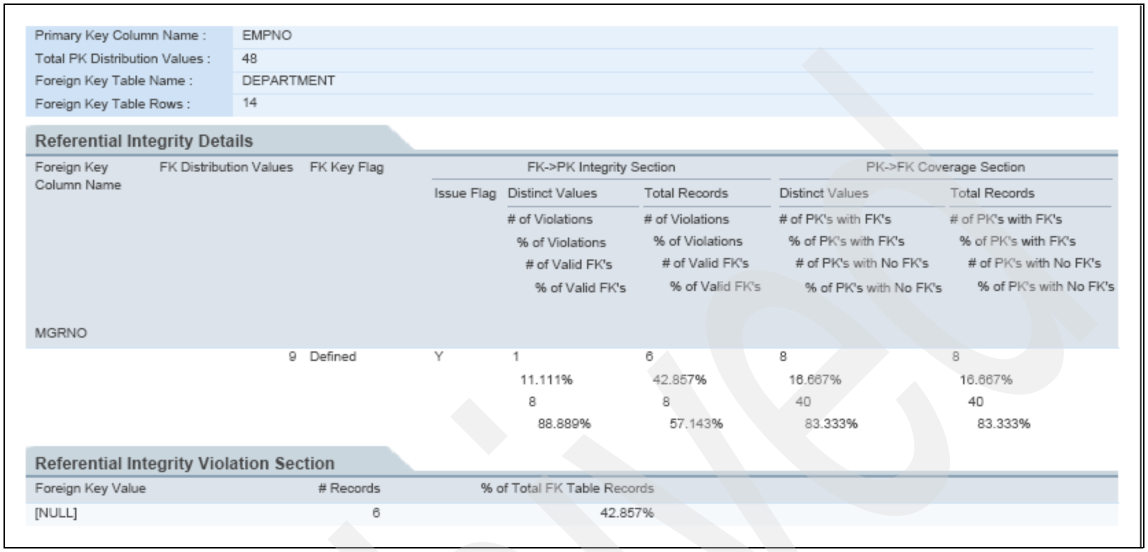
Defined Foreign Key Summary

Defined FK Columns	Defined Reference Columns		Null Values		Default Values		Invalid Values		FK Issue Flag		Total Rows	FK-PK Referential Integrity Violations	
	Referenced Column Table	Referenced Column	Total Null Rows	%	Total Default Rows	%	Total Invalid Rows	%		Data Class (Inferred)		Total Invalid Rows	%
WORKDEPT	DEPARTMENT	DEPTNO											
			0	.00	0	0	0	0	N	Code	48	0	.000

Figure 1-352 Defined foreign key - summary for EMPLOYEE

- Referential Integrity details for DEPARTMENT

  Figure 1-353 shows the referential integrity details of the foreign key MGRNO in the DEPARTMENT table including a listing of all violations.



Primary Key Column Name :	EMPNO
Total PK Distribution Values :	48
Foreign Key Table Name :	DEPARTMENT
Foreign Key Table Rows :	14

Referential Integrity Details

			FK->PK Integrity Section			PK->FK Coverage Section	
Foreign Key Column Name	FK Distribution Values	FK Key Flag	Issue Flag	Distinct Values	Total Records	Distinct Values	Total Records
				# of Violations % of Violations # of Valid FK's % of Valid FK's	# of Violations % of Violations # of Valid FK's % of Valid FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No FK's
MGRNO							
	9	Defined	Y	1 11.111% 8 88.889%	6 42.857% 8 57.143%	8 16.667% 40 83.333%	8 16.667% 40 83.333%

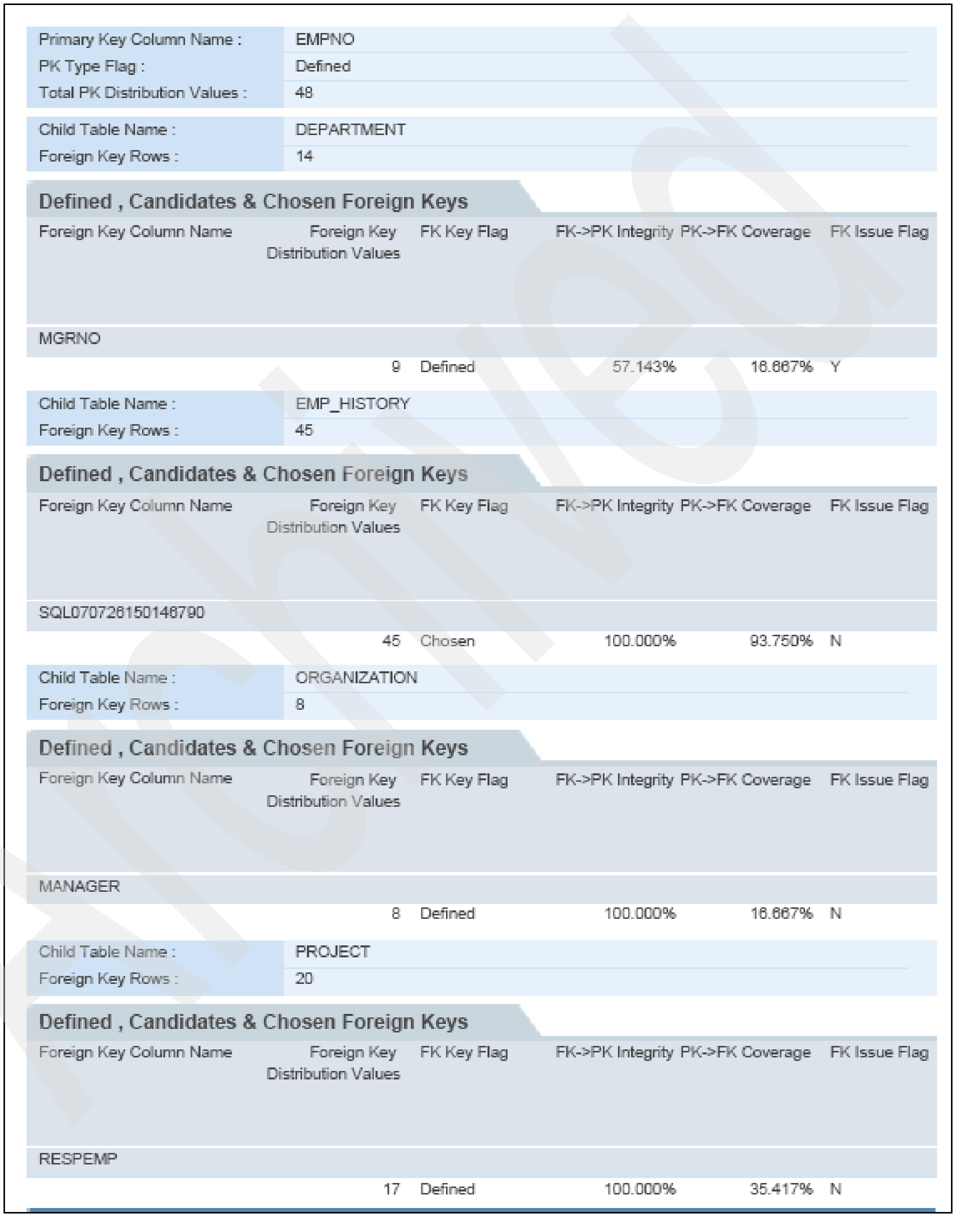
Referential Integrity Violation Section

Foreign Key Value	# Records	% of Total FK Table Records
[NULL]	6	42.857%

Figure 1-353 Referential Integrity details for DEPARTMENT

- Defined, candidate and chosen foreign keys

  Figure 1-354 shows a summary of all defined, candidate, and chosen (selected) foreign keys referencing the (primary key EMPNO of the) EMPLOYEE table.



Primary Key Column Name :	EMPNO
PK Type Flag :	Defined
Total PK Distribution Values :	48

Child Table Name :	DEPARTMENT
Foreign Key Rows :	14

Defined , Candidates & Chosen Foreign Keys

Foreign Key Column Name	Foreign Key Distribution Values	FK Key Flag	FK->PK Integrity	PK->FK Coverage	FK Issue Flag
MGRNO					
	9	Defined	57.143%	16.667%	Y

Child Table Name :	EMP_HISTORY
Foreign Key Rows :	45

Defined , Candidates & Chosen Foreign Keys

Foreign Key Column Name	Foreign Key Distribution Values	FK Key Flag	FK->PK Integrity	PK->FK Coverage	FK Issue Flag
SQL070726150146790					
	45	Chosen	100.000%	93.750%	N

Child Table Name :	ORGANIZATION
Foreign Key Rows :	8

Defined , Candidates & Chosen Foreign Keys

Foreign Key Column Name	Foreign Key Distribution Values	FK Key Flag	FK->PK Integrity	PK->FK Coverage	FK Issue Flag
MANAGER					
	8	Defined	100.000%	16.667%	N

Child Table Name :	PROJECT
Foreign Key Rows :	20

Defined , Candidates & Chosen Foreign Keys

Foreign Key Column Name	Foreign Key Distribution Values	FK Key Flag	FK->PK Integrity	PK->FK Coverage	FK Issue Flag
RESPEMP					
	17	Defined	100.000%	35.417%	N

Figure 1-354 Defined, candidate, and foreign keys

2

Financial services business scenario

Note: This scenario that we present in this chapter is based upon two fictitious financial institutions. In addition, the data that we present in our model is fictional also. Any resemblance to real institutions or data is totally coincidental.

In this chapter we describe a step-by-step approach to implementing IBM WebSphere Information Analyzer on a Red Hat Advanced Server Enterprise 4.0 platform using a typical financial services business scenario.

The topics covered include:

- Business requirements
- Environment configuration
- General approach
- Migration scenario
- Data integration scenario

2.1 Introduction

North American Bank provides core banking services such as savings, checking, and auto and home loans in North America and has significant market share in the eastern and midwest regions of the U.S. It additionally provides credit card and auto insurance services to its customer base. The primary IT platform of this bank is z/OS with DB2 and VSAM data sources.

Northern California Bank is a regional bank that provides core banking services such as savings, checking, and auto and home loans in the western region of the USA, and has significant market share in that region. It also provides brokerage services to its customer base. The primary IT platform of this bank is AIX with DB2 UDB as the data source.

Seeking a synergistic relationship, the two banks entered into a merger with the intent of becoming a national bank by growing their individual customer bases, and upselling and cross selling each others products' to their individual customers.

The two banks expect there to be some overlap of customers or groups of customers (such as members of a single family), who when identified could be granted special status. However, the two customer bases are mostly expected to be non-overlapped.

2.2 Business requirements

With the objective of streamlining the combined IT operations of the two banks, and taking advantage of the synergy of the products that are offered by the individual banks in the most expeditious manner, management made a business decision to implement the following strategy:

- Migrate the core services such as saving, checking, and auto and home loans, from the North American Bank system implementations to the those implemented by the Northern California Bank (generally considered to be superior in function and architecture).

Important: The general guideline when mismatches are found in the migration scenario is to have the target system have the over riding authority and not be modified except in absolutely unavoidable situations. This means ignoring data elements in the source that have no correspondence in the target, to truncate source data element content when the target data element's precision is lesser than that of the source, and accept coarser granularity data content values supported by the target where codes are involved. Judicious decisions involving minimal modifications to the target system should be the norm when mismatches are encountered.

- Build a customer relationship management (CRM) system to quickly take advantage of the individual brokerage, credit card, and auto insurance business' of each bank to upsell and cross-sell these products across the customer base.

  The CRM design is an "off-the-shelf" design that is customized to suit the particular requirements of the merged bank. It integrates information from the following sources with no changes in the interim to any of the existing systems:

  - Core business services from both the Northern California Bank and the North American Bank
  - Brokerage services from the Northern California Bank
  - Credit card and auto insurance services from the North American Bank

Important: The general guideline during data integration is that when there are mismatches between the sources and the new CRM system, that the CRM system design be modified to accommodate the existing system functions that are considered essential.

- A decision on less essential services such as Human Resources systems in each bank is deferred until the completion of the migration and CRM system implementation.
- The migration and CRM implementation efforts are to proceed in parallel, with no dependency of one over the other.

  Therefore, the CRM system is designed to be sourced from the core services of the individual banks even as migration of these systems is occurring from one bank to the other. At the completion of migration of the core business services, it would be easy to have the CRM sourced entirely from the migrated system than both the systems by discarding data from the migration source system.

The data models of the systems in our scenario are as follows:

- The data model of the Northern California Bank is shown in Figure 2-1. The DDL is shown in Example D-1 on page 570.
- The data model of the North American Bank is shown in Figure 2-2 on page 421. The DDL and field in the VSAM file are shown in Example D-2 on page 579 and Example D-3 on page 585.
- The “off-the-shelf” data model of the CRM is shown in Figure 2-3 on page 422. The DDL is shown in Example D-4 on page 586.

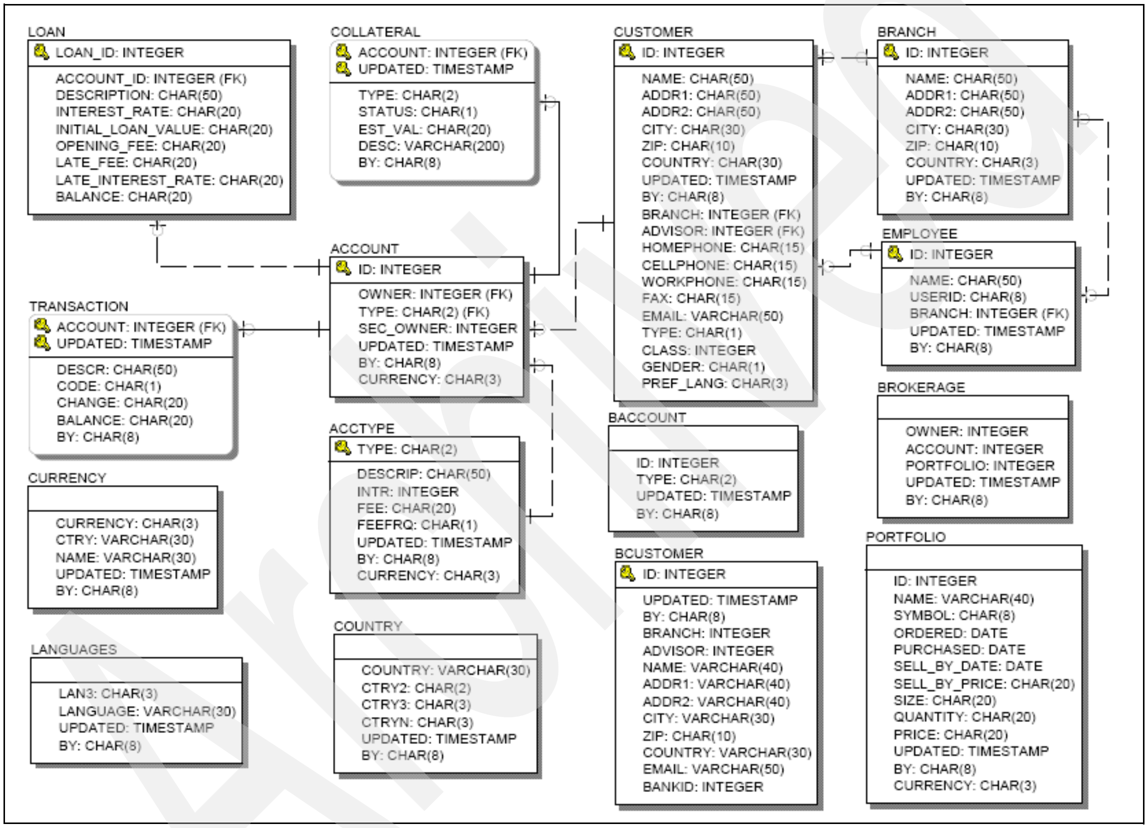


Figure 2-1 Data model of the Northern California Bank

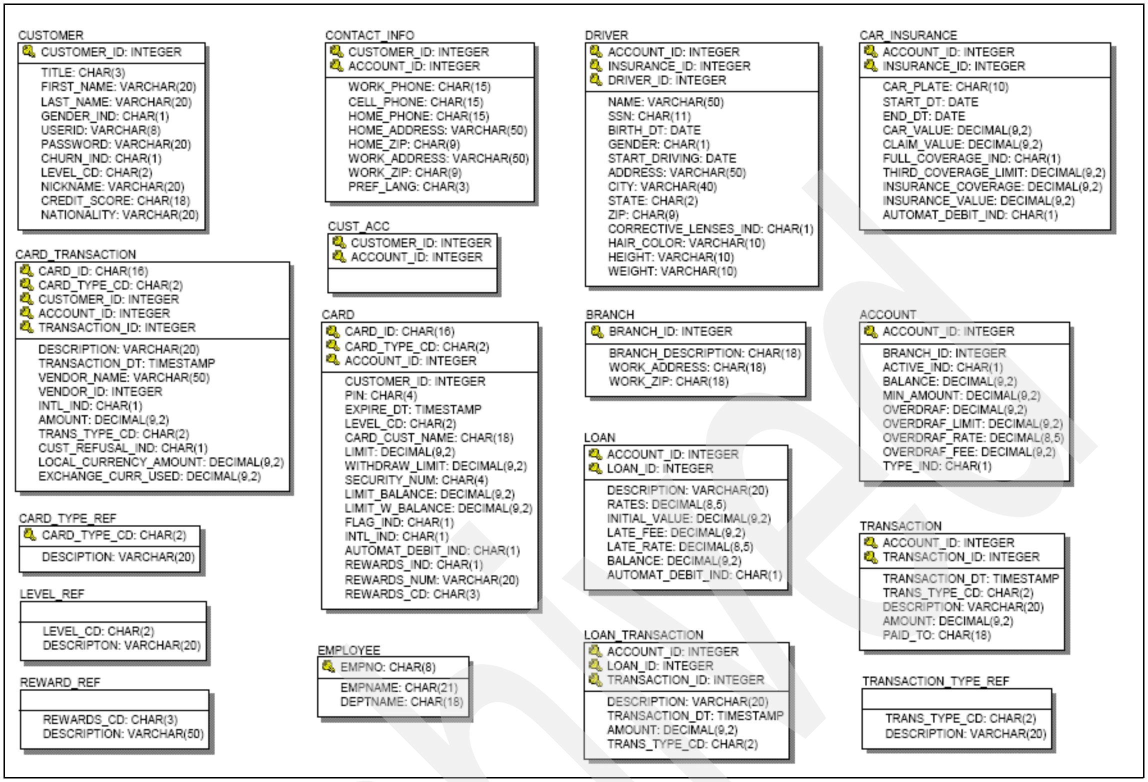


Figure 2-2 Data model of the North American Bank

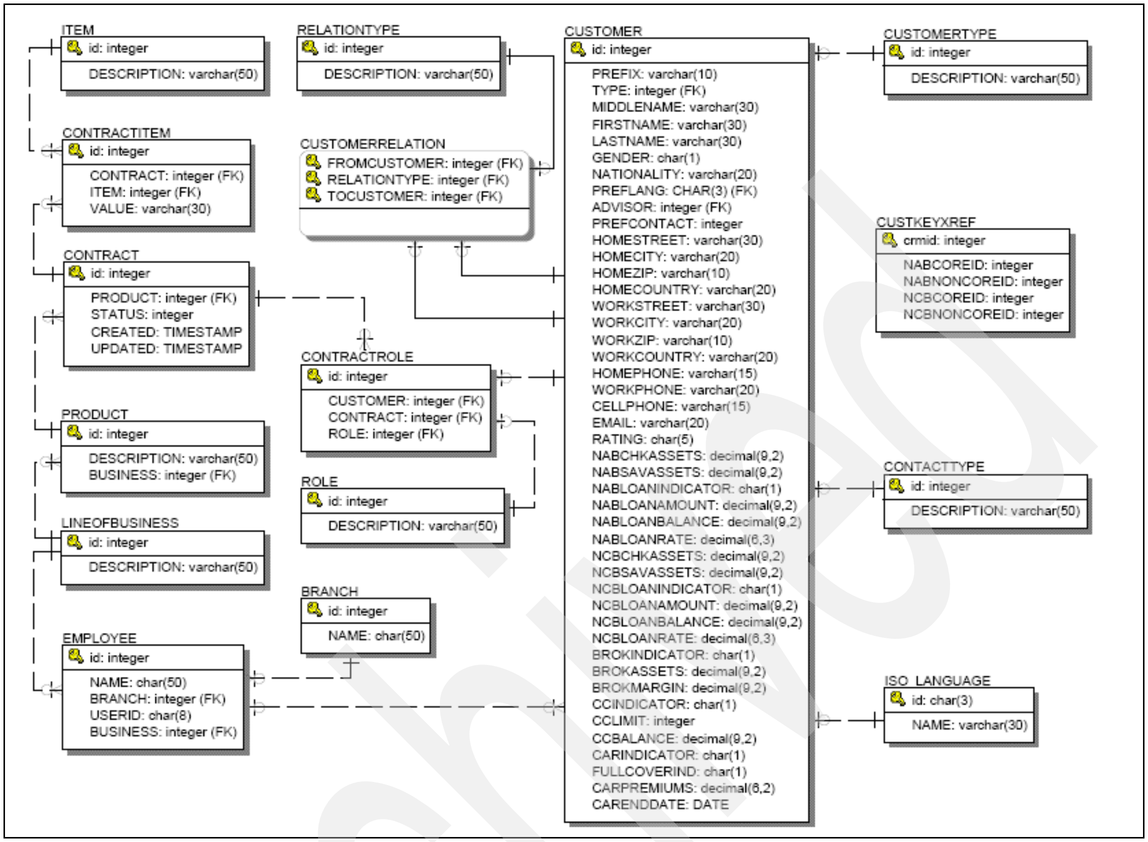


Figure 2-3 "Off-the-shelf" data model of the CRM

2.3 Environment configuration

Figure 2-4 shows the configuration of the merged environment with the new CRM system.

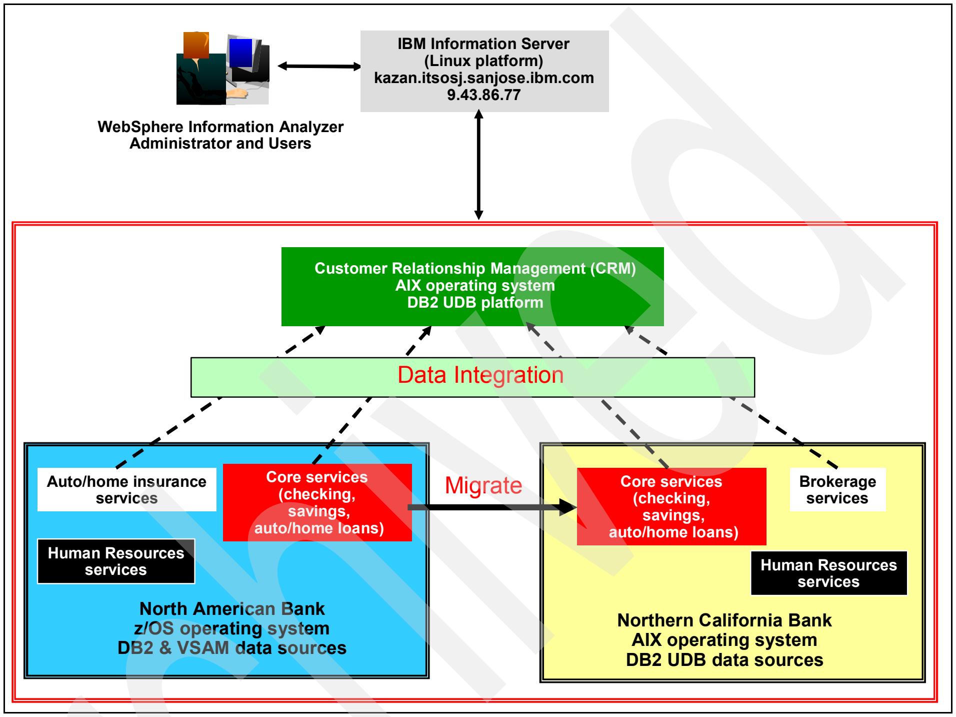


Figure 2-4 Merged banks' environment configuration

Figure 2-4 shows:

- A Red Hat Enterprise Linux 4 server (kazan.itsosj.sanjose.ibm.com, 9.43.86.77) that has IBM Information Server and IBM WebSphere Information Analyzer installed.
- A single IBM AIX 5.2 server (Jamaica.itsosj.sanjose.ibm.com, 9.43.86.55) runs the Northern California Bank's IT systems including the core services, brokerage services, and human resources services. DB2 V9.1 is used for the data sources.

> **Attention:** The CRM system is meant to be implemented on the IBM AIX 5.2 server. In this book, we do not actually build the CRM system. We merely take note of the CRM data model to ensure that it is capable of supporting the data content, data types, precision, and scale of the data to be integrated from the core services, credit card, auto insurance services, and brokerage services of the two banks.

- A single z/OS image (wtsc59.itso.ibm.com, 9.12.4.10) that runs the North American Bank's IT systems, including the core services, credit card, auto insurance services, and human resources services. DB2 for z/OS V8 and VSAM is used for the data sources. WebSphere Information Integrator Classic Federation (IICF) V9.1 provides the connectivity from IBM WebSphere Information Analyzer to the VSAM and DB2 for z/OS data sources.

> **Attention:** Our configuration in Figure 2-4 is meant to represent the eclectic mix of operating systems and platforms that are typical of mergers between multiple organizations and describes how IBM Information Server and IBM WebSphere Information Analyzer integrate into such an environment. We are *not* making recommendations about how to configure your environment in this manner or stating that it will deliver the scalability and performance requirements of your business solution.

2.4 General approach

Figure 2-5 shows the recommended sequence of steps in a migration or data integration scenario. IBM WebSphere Information Analyzer plays the key role in the second step that involves identifying the differences (both structure and content) between the sources and targets.

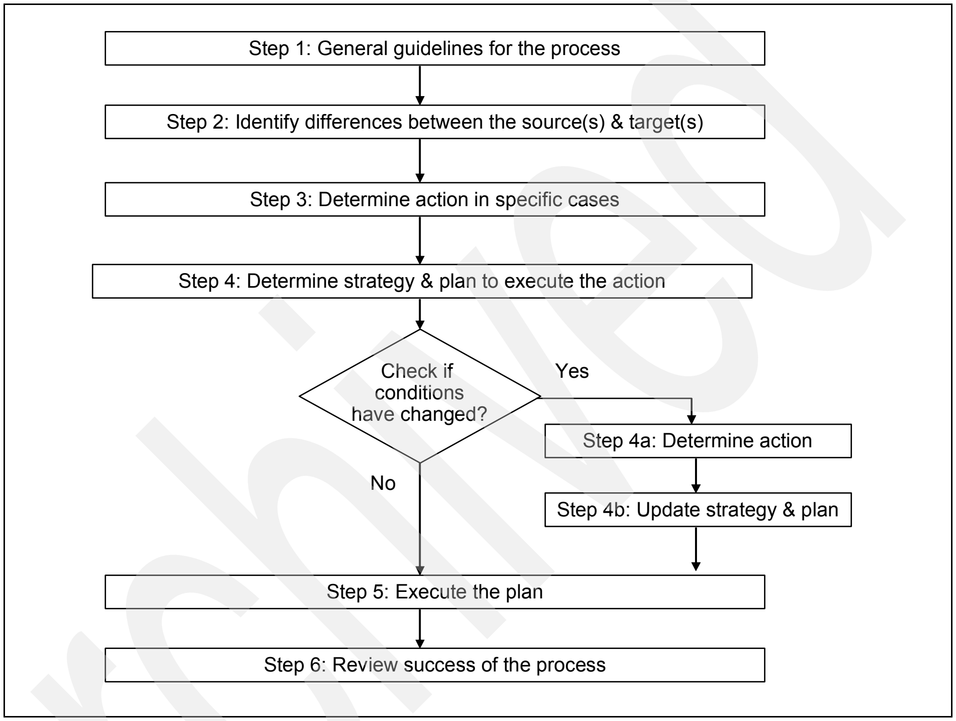

Step 1: General guidelines for the process
Step 2: Identify differences between the source(s) & target(s)
Step 3: Determine action in specific cases
Step 4: Determine strategy & plan to execute the action
Check if conditions have changed?
Yes
No
Step 4a: Determine action
Step 4b: Update strategy & plan
Step 5: Execute the plan
Step 6: Review success of the process

Figure 2-5 General approach

We describe each of the steps shown in Figure 2-5 briefly in this section.

2.4.1 Step 1: General guidelines for the process

In a migration or a data integration effort, structural and content differences between the sources and targets will need to be resolved. Table 2-1 shows many of the commonly encountered differences and potential actions. These actions are self-explanatory.

We recommend that you define broad action guidelines for mismatches in critical and non-critical elements between the data sources and targets for the commonly encountered differences. Consequently, if unexpected differences are discovered, you can focus your energies on addressing them in the inevitable time constrained circumstances of migration and data integration projects.

Table 2-1 Commonly encountered differences and potential actions

Commonly encountered differences	Potential actions
Data elements in the source not found in the target	► Add the data elements to the target; significant impact on target applications likely ► Ignore it; loss of function ► Combination of the above depending upon the data element
Data type mismatch between the source and target data elements ► Compatible ► Incompatible ► Coarse to fine ► Fine to coarse	► Compatible – Simple mapping ► Incompatible – Use a cross reference table to map from the source data type to the target data type ► Coarse to fine – No action other than possible transformation ► Fine to coarse – Modify target definition to match fine granularity of the source; significant impact on target applications likely – Transform with loss of granularity; loss of function
Precision, and scale mismatch between the source and the target data elements ► Coarse to fine ► Fine to coarse	► Coarse to fine – No action other than possible transformation ► Fine to coarse – Modify target definition to match fine granularity of the source; significant impact on target applications likely – Transform with loss of granularity; loss of function

Commonly encountered differences	Potential actions
Data elements in the target not found in the source, and target data element not nullable or has no default values	► Define default values in the target; some impact on target applications
Code mismatch between the source and the target data elements; for example, a salutation can be *Mr*, *Mrs*, *Dr*, *Miss*, and *Ms*, and source and target do not have corresponding codes ► Coarse to fine ► Fine to coarse	► Coarse to fine – Use transformation to perform the mapping ► Fine to coarse – Transform with loss of granularity; loss of function – Modify target definition to match fine granularity of the source; significant impact on target applications likely
Multiple data elements in the source maps to a single data element in the target ► For example, an address	► Use transformation to perform the mapping
Single data element in the source maps to multiple data elements in the target ► For example, an address	► Use transformation to perform the mapping – Might require standardization software
Different character maps such as Unicode and ASCII	Transformation to perform the mapping

2.4.2 Step 2: Identify differences between the sources and targets

Differences between the source and target can be determined as follows:

- From active relational database catalogs, dictionaries, repositories, and other documentation that provide details about the metadata of the various data sources and targets.

  More often than not, metadata information stored in sources (other than the active relational database catalogs) tends to be out of date because it is not maintained regularly as systems evolve.

- From an analysis of the data itself using tools such as IBM WebSphere Information Analyzer. Metadata is deduced from the data and presented to the data analyst for review and affirmation or denial of the deduction.

These are complementary approaches, essential to achieving a fuller understanding of how synchronized the definition of the metadata is with the data content. It also enables you to keep the metadata about data sources current, which is critical for building new systems that require data integration from existing systems.

Attention: We strongly recommend that tools such as IBM WebSphere Information Analyzer sharply focus analyses on data content in specific tables and columns based on metadata information obtained from active relational database catalogs, dictionaries, repositories, and other documentation. This will avoid unnecessary data analysis by data analysts of data that is not appropriate for such an investigation. An added benefit is limiting unnecessary and irrelevant processing that could consume valuable processing power that would be best consumed by other applications.

IBM WebSphere Information Analyzer identifies differences between the defined metadata for a data source and the inferences made from the actual data content in these data sources.

Comparing independently generated IBM WebSphere Information Analyzer analyses of different data sources is beyond the scope of IBM WebSphere Information Analyzer and is likely a manual process.

2.4.3 Step 3: Determine action in specific cases

After you compare the metadata and inferred metadata from the data content, you can choose to synchronize them with appropriate actions, such as modifying the metadata definitions or cleansing the data or ignoring them at your own peril.

In the migration and data integration scenarios discussed in this chapter, we compare the metadata and data content between multiple data sources manually. Based on the identified differences with specific data elements, we choose the most acceptable action from the available options, some of which are described in Table 2-1 on page 426.

2.4.4 Step 4: Determine strategy and plan to execute action

After you have chosen the action, you need to design the appropriate tools and procedures to effect the chosen action for each data element. This will most likely involve the use of data cleansing tools (such as IBM WebSphere QualityStage), ETL[1] tools (such as IBM WebSphere DataStage), data management and database tools (such as DB2 Data Warehouse Edition) where stored procedures or specific database functions are involved, and user-written code.

The execution of the plan would most likely take an extended period of many hours or days. To ensure that migration or data integration occurs without disrupting the availability of the source applications, you will most likely

[1] Extract, transform, and load

synchronize the source and target in multiple phases. A snapshot of the source is initially bulk loaded into the target, followed by an incremental update of the target with changes occurring in the source during the bulk load.

A well planned and rigorously tested set of procedures is essential for a smooth and successful migration or data integration project.

A discussion of this topic is beyond the scope of this book. However, it would be useful to note that IBM WebSphere Information Analyzer can play an active part in the rigorous testing of the process. This can occur at multiple points as follows:

- Assessment of extracted test data to ensure conformity to defined test and transformation objectives.
- Evaluation and comparison of outputs from the development process to ensure broad and quick review.
- Validation of test/QA output against test objectives, particularly for cross-domain comparison of source input to target output and proper data mapping.
- Review of target load files to ensure completeness, comprehensiveness, and consistency to expected results.

In all cases, IBM WebSphere Information Analyzer can facilitate this work by providing rapid insight into the data.

> **Important:** The analysis of data can be expected to take several weeks given the fact that multiple personnel IT Data Analysts (DA) and subject matter experts (SME) are involved in ensuring data quality. Therefore, it is conceivable that changes could occur to the metadata and data content of the data sources and targets during this interval. Therefore, prior to executing the plan, you should check if structure or content changes have occurred. If so, you should determine its impact on the existing plan, update the strategy and plan if required, and test the revised plan before execution.

2.4.5 Step 5: Execute the plan

This step involves executing the plan that was designed in 2.4.4, “Step 4: Determine strategy and plan to execute action” on page 428.

2.4.6 Step 6: Review success of the process

After the designed plan has been executed, you need to verify that the process was successful by comparing the metadata and content in the sources and targets.

The process for doing this would vary depending upon whether a migration[2] or data integration[3] was involved.

A discussion of this topic is beyond the scope of this book.

2.5 Migration from North American Bank systems to Northern California Bank systems

As mentioned earlier, a business decision was made to migrate the core services (checking, savings, and auto/home loans) of the North American Bank on the z/OS platform to those of the Northern California Bank on the AIX platform.

> **Important:** A number of assumptions are made about the migration in this scenario, some of which might not apply to your particular environment. What we want to achieve in this scenario is to highlight IBM WebSphere Information Analyzer functionality (through its reports) that can be used to identify defined and inferred metadata differences within the same data source, validate the integrity of data within a data source (within a table and across tables), and understand the frequency distribution of data values within specific data elements. IBM WebSphere AuditStage is used to ensure business rule compliance of data within a data source. Such functionality is not currently available in IBM WebSphere Information Analyzer but is expected to become available in it in future.

> **Attention:** Because the focus of this book is IBM WebSphere Information Analyzer, we do *not* describe the procedures for unloading, cleansing, transforming, and loading of the source data into the target environment. Those tasks are the domain of IBM WebSphere QualityStage and IBM WebSphere DataStage and will be covered in upcoming IBM Redbooks publications.

[2] One-time activity

[3] On-going activity, because data is continuously fed to the target, and changes to both the metadata and data content occur at the sources

In this section, we describe the following topics related to the migration:

- Assumptions about the migration
- IBM WebSphere Information Analyzer features used
- IBM WebSphere AuditStage features used
- North American Bank analysis
- Northern California Bank analysis
- Migration Analysis

2.5.1 Assumptions about the migration

The assumptions made about the migration are as follows:

- As far as possible, no changes should be made to the data model structures of either the North American Bank and Northern California Bank for the duration of the migration. It is more difficult to try and enforce the same on data content because there are many points of data entry.
- Data models of the North American Bank and Northern California Bank are known.

  The data model and fields in the individual tables/files of the core services (being migrated) for the North American Bank are shown in Figure 2-2 on page 421, Example D-2 on page 579, and Example D-3 on page 585), while those of Northern California Bank are shown in Figure 2-1 on page 420 and Example D-1 on page 570.
- The primary keys and foreign keys of most of the tables and files are known. Some of the referential integrity relationships are explicitly defined in the tables, while others are implicitly defined and enforced by user-applications. There are some tables with no explicitly defined primary keys or foreign keys.
- The data model of the Northern California Bank (target of migration) is considered acceptable, thereby eliminating the need to perform functional dependency analysis.

  Because the North American Bank core services system (source of migration) is to be eliminated, there is no need to perform functional dependency analysis on this system either.
- When data elements are to be migrated over from the source North American Bank core services systems to the target Northern California Bank core services systems and when data elements in multiple data sources appear to have the same information, you need to identify the one source (“system of record”) that is considered to be the definitive source to be used in migrating this data content.

  We assume that the “system of record” is known for all the data items in the source to be migrated to the target.

Note: If the “system of record” is not known for a particular data item, you might consider performing column analysis on the candidate columns and looking at the cardinality and frequency distribution to make the decision.

- Invalid data might exist in the data elements in the source to be migrated such as invalid values, primary key violations (non-unique or nulls), referential integrity violations (foreign key), and business rules across multiple data elements in one or more tables (for example, only employees who are sales persons and executives can have a commission).
- There are some data elements in the source system that do not exist in the target system.
- There are some data elements in the target system that do not exist in the source system. These data elements are not nullable and do not have default values.
- There are data elements that are common to both the source and target systems, but the precision and scale of the source system is greater than that of the target system.
- There are data elements that are common to both the source and target systems, but the precision and scale of the source system is less than that of the target system.
- There are data elements that are common to both the source and target systems but have a different data type, as in the case of amount columns where it is defined as DECIMAL(9,2) in the North American Bank systems, and CHAR(20) in the Northern California Bank systems.
- Exclude any detailed data profiling of free text fields, such as names and addresses, because they are not critical to a successful migration.

Note: It is critical to profile names and addresses when you try to determine relationships between the merged customer bases to detect potentially high value customers and to standardize address formats across the data in the two banks using tools such as IBM WebSphere QualityStage. More information about how to profile will be included in an upcoming IBM Redbooks publication involving IBM WebSphere QualityStage.

- The keys used in the source and target have different data types, precisions, and domains.
- There are many corresponding code and indicator fields in the source and target. They have different data types, precision, scale, and content as in the case of the LEVEL_CD[4] and CLASS[5] columns.

Important: The general guideline when mismatches are found in the migration scenario is to have the target system have the over riding authority and not be modified except in absolutely unavoidable situations. This means ignoring data elements in the source that have no correspondence in the target, to truncate source data element content when the target data element's precision is lesser than that of the source, and to accept coarser granularity data content values supported by the target where codes are involved. Judicious decisions involving minimal modifications to the target system should be the norm when mismatches are encountered.

We have deliberately introduced a number of differences between the data types and content of the core services systems of the two banks. Table 2-2 shows some of these differences.

These mismatches will be detected by IBM WebSphere Information Analyzer and will need further analysis by IT Data Analysts (DA) and SMEs in order to determine the proper course of action.

Table 2-2 Data type and content mismatch introduced between the source and target banks

Data elements	Northern American Bank (source) on z/OS platform	Northern California Bank (target) on AIX platform
Keys in the tables	Some with greater precision, and some with lesser precision than the target	Some with greater precision, and some with lesser precision than the source
Customer's gender	"M" for male, "F" for female	"0" for male, "1" for female
Customer's status	"SL" for silver, "GL" for gold, "PL" for platinum	1 through 9; 1 highest, 9 lowest
Customer's preferred language	not defined	ISO code such as "ENG" or "SPA"
Account type	"S" for savings and "C" checking. Loans in a separate table	"SS" for standard savings, "SC" for standard checking, and "LN" for loans
Currency	not defined	ISO code such as "EUR" or USD
Country	not defined	ISO code such as "US" or AU or CHN

[4] CHAR(2) data type with values "SL", "GL", and "PL" in the CUSTOMER table in the North American Bank system

[5] INTEGER data type with values 1 through 9 in the CUSTOMER table in the Northern California Bank system

Data elements	Northern American Bank (source) on z/OS platform	Northern California Bank (target) on AIX platform
Nationality	Free text field	not defined
Transaction type	“T” for transfer; the amount is negative for debit, and positive for credit	“C” for credit, “D” for debit
Phone numbers	nnnnnnnnnn (10 digits)	nnn-nnn-nnnn (hyphenated 10 digits)
Account fee frequency	not defined	“Y” for yearly, “M” for monthly, and “C” per check
Customer name	First name and Last name in separate fields	Free text field containing full name
Customer address	Free text field containing full address including city	Multiple fields for street names
City	not defined	Separate field
Customer type	not defined	“P” for person, “O” for organization
Various amounts	Decimal (9,2)	Character (20)

2.5.2 IBM WebSphere Information Analyzer features used

As mentioned earlier in 2.5.1, “Assumptions about the migration” on page 431, we use the metadata inferencing, frequency distribution, data integrity, validation, and baseline analysis features of IBM WebSphere Information Analyzer but do *not* use its discovery features, such as candidate primary key analysis, candidate foreign key analysis, and cross domain analysis for redundancy detection.

> **Note:** While we are not using cross-domain analysis in this scenario, it could be applied for assessing overlaps of source to target fields and validating that there are no unexpected conditions.

We will be using the following features of IBM WebSphere Information Analyzer:

- Column Analysis for metadata inferencing, frequency distribution, and data integrity and validation.
- Primary Key Analysis for data integrity and validation.
- Foreign Key Analysis for validating referential integrity.

- Baseline Analysis to determine whether any significant changes have occurred to the structure and content of the source from the time the data profiling effort commenced to just prior to the execution of the plan to migrate the data sources to the target.

2.5.3 IBM WebSphere AuditStage features used

We will be using IBM WebSphere AuditStage's Data Filters functionality to perform business rule compliance for data elements within the same table and across multiple tables in the source.

2.5.4 North American Bank analysis

As mentioned earlier, North American Bank's core services are to be migrated to those of the Northern California Bank.

Using metadata information available in dictionaries, documentation and the relational catalogs, the keys, code fields, indicator fields, and referential integrity relationships (both implicit and explicit) to be profiled were identified as shown in Table 2-3.

> **Note:** For convenience, we chose to include all the columns in all the tables in our data profiling effort. In the real world, however, to avoid report and information overload, you would apply the 80/20 rule and focus on only the critical master data as described in 1.2.1, "Data assessment approach" on page 5.

We include only selected portions of the generated reports (for some of the fields shown in Table 2-3) of the data profiling effort in this section. You can download the complete reports from the IBM Redbooks Web site:

ftp://www.redbooks.ibm.com/redbooks/SG247508/

> **Note:** We describe the process of generating reports in 1.14, "Reports" on page 394 and do not repeat that information here. We show only the relevant portions of reports here.

Business rules involving data elements in one or more tables are shown in Table 2-4. These are verified using IBM WebSphere AuditStage.

Table 2-3 Main keys, codes, indicators in North American Bank core services

Table	Key column(s)	Code column	Indicator column	RI relationships
CUSTOMER	CUSTOMER_ID	▸ TITLE ▸ LEVEL_CD	▸ GENDER_IND ▸ CHURN_IND	LEVEL_CD to LEVEL_REF table
CONTACT_INFO	CUSTOMER_ID, ACCOUNT_ID	▸ HOME_ZIP ▸ WORK_ZIP		CUSTOMER_ID to CUSTOMER table
ACCOUNT	ACCOUNT_ID		▸ ACTIVE_IND ▸ TYPE_IND	ACCOUNT_ID to ACCOUNT table
LOAN	ACCOUNT_ID, LOAN_ID		AUTOMAT_DEBIT_IND	ACCOUNT_ID to ACCOUNT table
LOAN_TRANSACTION	ACCOUNT_ID, LOAN_ID, TRANSACTION_ID	TRANS_TYPE_ID	AUTOMAT_DEBIT_IND	▸ ACCOUNT_ID to ACCOUNT table ▸ TRANS_TYPE_CD to TRANSACTION_TYPE_REF table
TRANSACTION	ACCOUNT_ID, TRANSACTION_ID	TRANS_TYPE_CD		▸ ACCOUNT_ID to ACCOUNT table ▸ TRANS_TYPE_CD to TRANSACTION_TYPE_REF table
BRANCH	BRANCH_ID	WORK_ZIP		
CUST_ACC	CUSTOMER_ID, ACCOUNT_ID			▸ CUSTOMER_ID to CUSTOMER table ▸ ACCOUNT_ID to ACCOUNT table
TRANSACTION_TYPE_REF	TRANS_TYPE_CD	TRANS_TYPE_CD		

Table 2-4 Business rules in North American Bank core services

Serial number	Description
1	An account (in the ACCOUNT table) can only have a transaction if the ACTIVE_IND column is set to "Y"
2	The OVERDRAFT column value cannot exceed the OVERDRAFT_LIMIT column value in the ACCOUNT table
3	When the OVERDRAFT column has a value, the OVERDRAFT_RATE and OVERDRAFT_FEE columns in the ACCOUNT table must be filled

Column Analysis reports

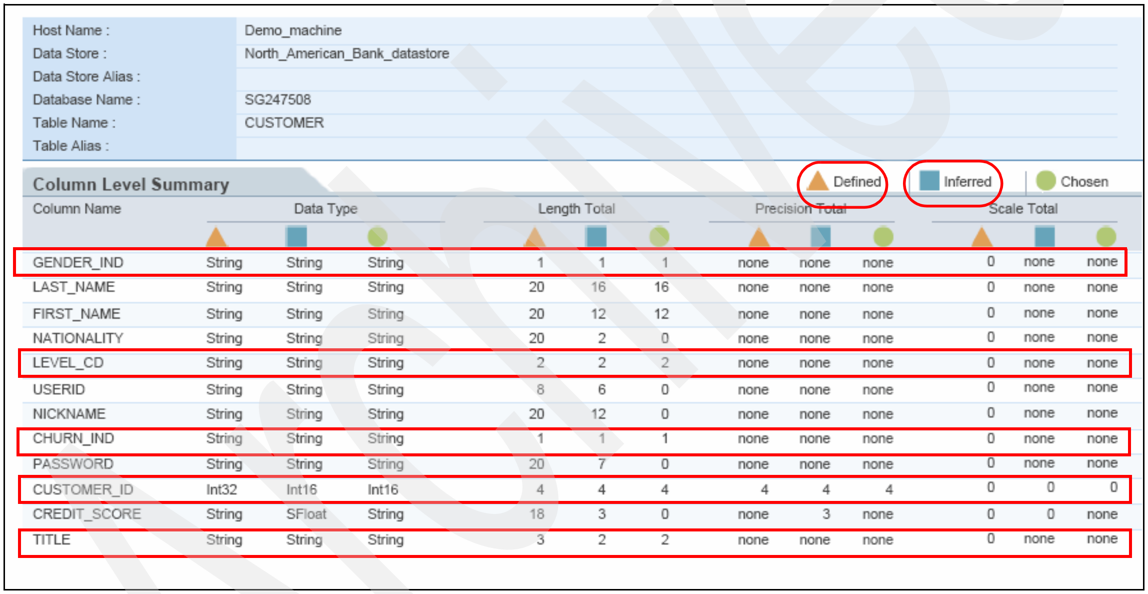
These reports provide information about column values for completeness, validity, structure, and format as described in 1.7, "Column analysis" on page 109. These reports are used to review the columns for domain, structure and format integrity.

Figure 2-6 on page 437 shows the Column Inferences "Type & Length Properties (Defined/Inferred/Chosen)" report for the CUSTOMER table. It identifies the data type, length, precision and scale of each column as defined in the metadata and inferred from the data content. Any selected status of each of these categories is also shown here. This information (along with those relating to other columns in other tables) is recorded in Table 2-13 on page 475. Our focus is on keys, codes, and indicators as highlighted.

Figure 2-7 on page 438 shows the Column Frequency “Frequency By Frequency” report for the GENDER_IND column in the CUSTOMER table. It shows some unexpected values such as spaces, in addition to the valid values M, F, U, and NULL.

Figure 2-8 on page 439 shows the Column Domain Values “Completeness and Validity Summary” report for the CUSTOMER table. It shows 41 occurrences of incomplete (NULL) values, and 41 occurrences of invalid values in the GENDER_IND column.

Document all the valid and invalid values for a column in a table as shown in Table 2-5 on page 439.



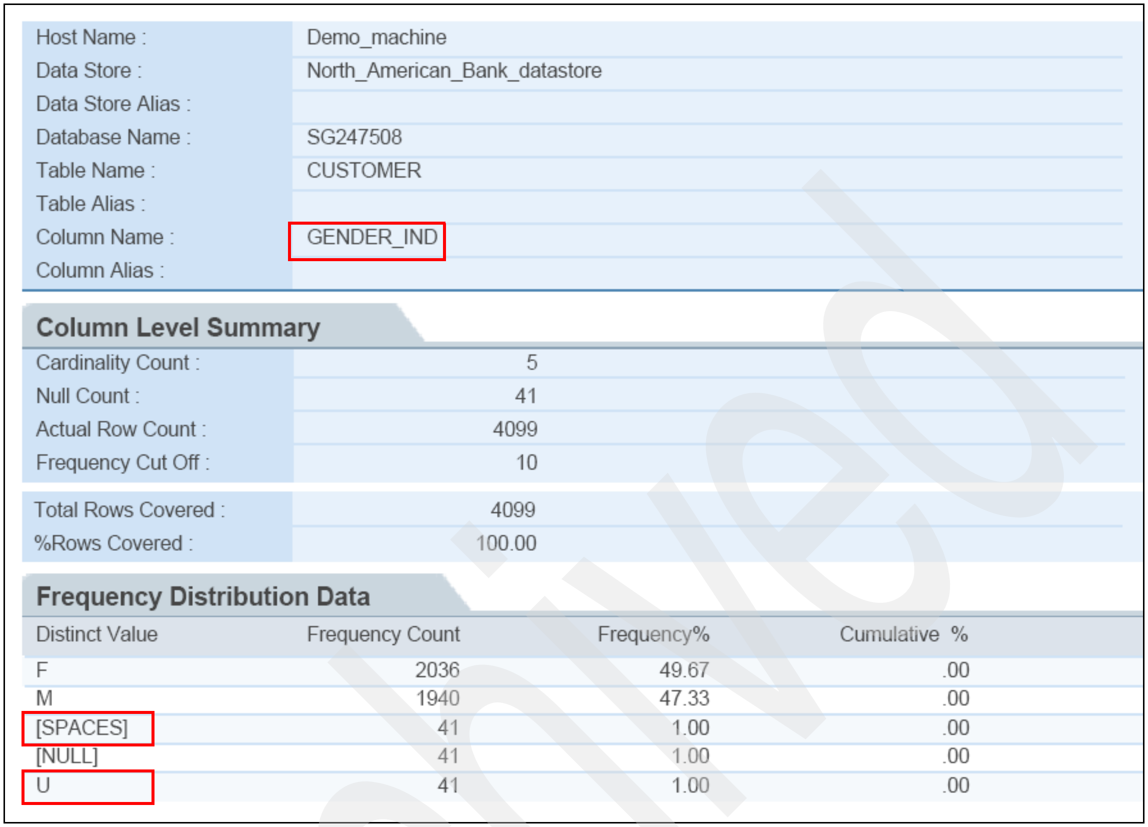
Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	CUSTOMER
Table Alias :	

Column Level Summary

▲ Defined ■ Inferred ● Chosen

Column Name	Data Type			Length Total			Precision Total			Scale Total		
	▲	■	●	▲	■	●	▲	■	●	▲	■	●
GENDER_IND	String	String	String	1	1	1	none	none	none	0	none	none
LAST_NAME	String	String	String	20	16	16	none	none	none	0	none	none
FIRST_NAME	String	String	String	20	12	12	none	none	none	0	none	none
NATIONALITY	String	String	String	20	2	0	none	none	none	0	none	none
LEVEL_CD	String	String	String	2	2	2	none	none	none	0	none	none
USERID	String	String	String	8	6	0	none	none	none	0	none	none
NICKNAME	String	String	String	20	12	0	none	none	none	0	none	none
CHURN_IND	String	String	String	1	1	1	none	none	none	0	none	none
PASSWORD	String	String	String	20	7	0	none	none	none	0	none	none
CUSTOMER_ID	Int32	Int16	Int16	4	4	4	4	4	4	0	0	0
CREDIT_SCORE	String	SFloat	String	18	3	0	none	3	none	0	0	none
TITLE	String	String	String	3	2	2	none	none	none	0	none	none

Figure 2-6 Column Inferences Type & Length Properties (Defined/Inferred/Chosen) report for the CUSTOMER table



Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	CUSTOMER
Table Alias :	
Column Name :	GENDER_IND
Column Alias :	

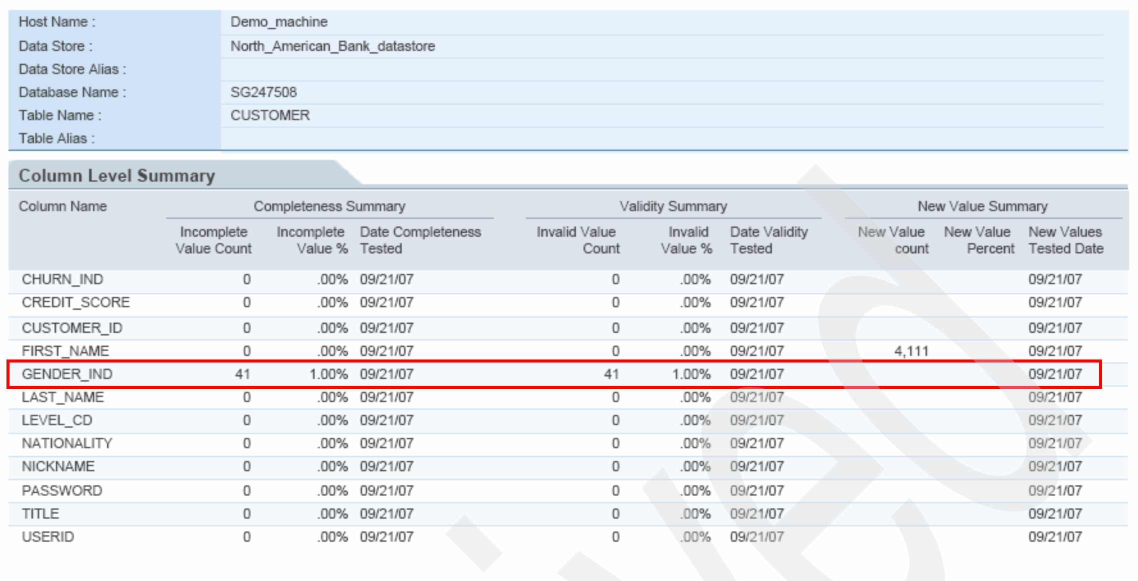
Column Level Summary

Cardinality Count :	5
Null Count :	41
Actual Row Count :	4099
Frequency Cut Off :	10
Total Rows Covered :	4099
%Rows Covered :	100.00

Frequency Distribution Data

Distinct Value	Frequency Count	Frequency%	Cumulative %
F	2036	49.67	.00
M	1940	47.33	.00
[SPACES]	41	1.00	.00
[NULL]	41	1.00	.00
U	41	1.00	.00

Figure 2-7 Column Frequency "Frequency By Frequency" report for the GENDER_IND column in the CUSTOMER table



Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	CUSTOMER
Table Alias :	

Column Level Summary

Column Name	Completeness Summary			Validity Summary			New Value Summary		
	Incomplete Value Count	Incomplete Value %	Date Completeness Tested	Invalid Value Count	Invalid Value %	Date Validity Tested	New Value count	New Value Percent	New Values Tested Date
CHURN_IND	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
CREDIT_SCORE	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
CUSTOMER_ID	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
FIRST_NAME	0	.00%	09/21/07	0	.00%	09/21/07	4,111		09/21/07
GENDER_IND	41	1.00%	09/21/07	41	1.00%	09/21/07			09/21/07
LAST_NAME	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
LEVEL_CD	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
NATIONALITY	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
NICKNAME	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
PASSWORD	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
TITLE	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
USERID	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07

Figure 2-8 Column Domain Values "Completeness & Validity Summary" report for the CUSTOMER table

Table 2-5 Valid and invalid column values

Column name	Valid values	Invalid values
GENDER_IND in the CUSTOMER table	M, F, NULL, U	" "

Primary Key Analysis reports

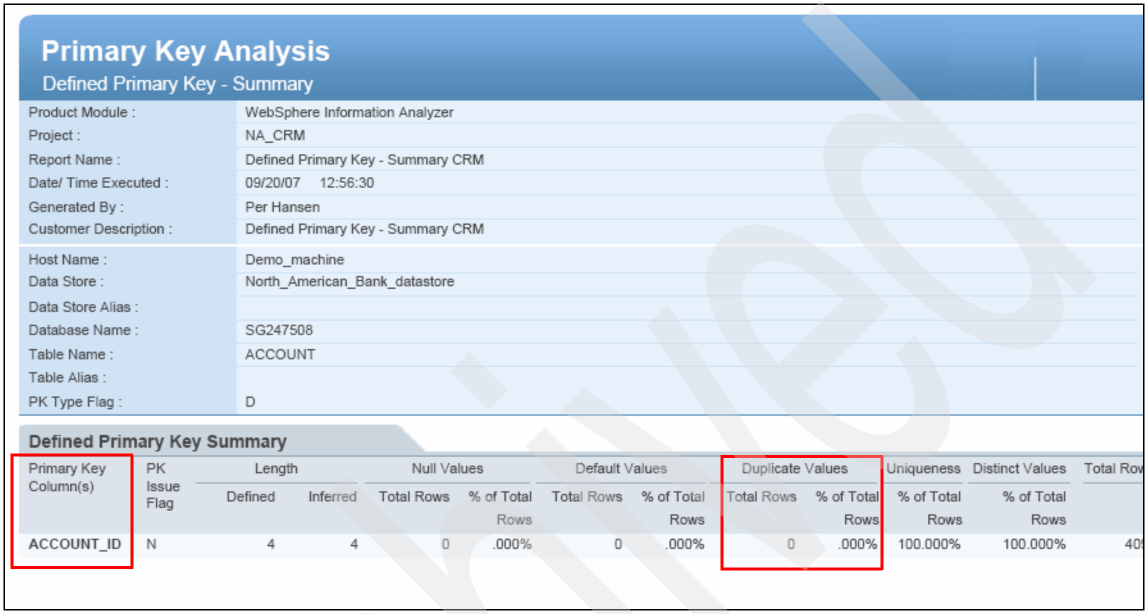
These reports provide information about candidate primary keys as well as duplicate records in the table.

Figure 2-9 on page 440 shows the Primary Key Analysis "Defined Primary Key - Summary" report for the ACCOUNT table. It identifies the length (defined and inferred), number of null values, default values, uniqueness values, distinct values, and number of rows in the table. This report shows no duplicate records or null values in the table.

Figure 2-10 on page 441 shows the Primary Key Analysis "Duplicate Exceptions" report for the CONTACT_INFO table. It identifies duplicate values in each column of the primary key. There are none in this case.

Figure 2-11 on page 442 shows the Primary Key Analysis "Defined & Candidate Summary" report for the BRANCH table. It identifies duplicate values in each column of the primary key. There are none in this case, but candidate keys are identified.

Any duplicate rows or tables with no primary key (but candidate primary keys) should be identified and documented as shown in Table 2-6 on page 442.



Primary Key Analysis

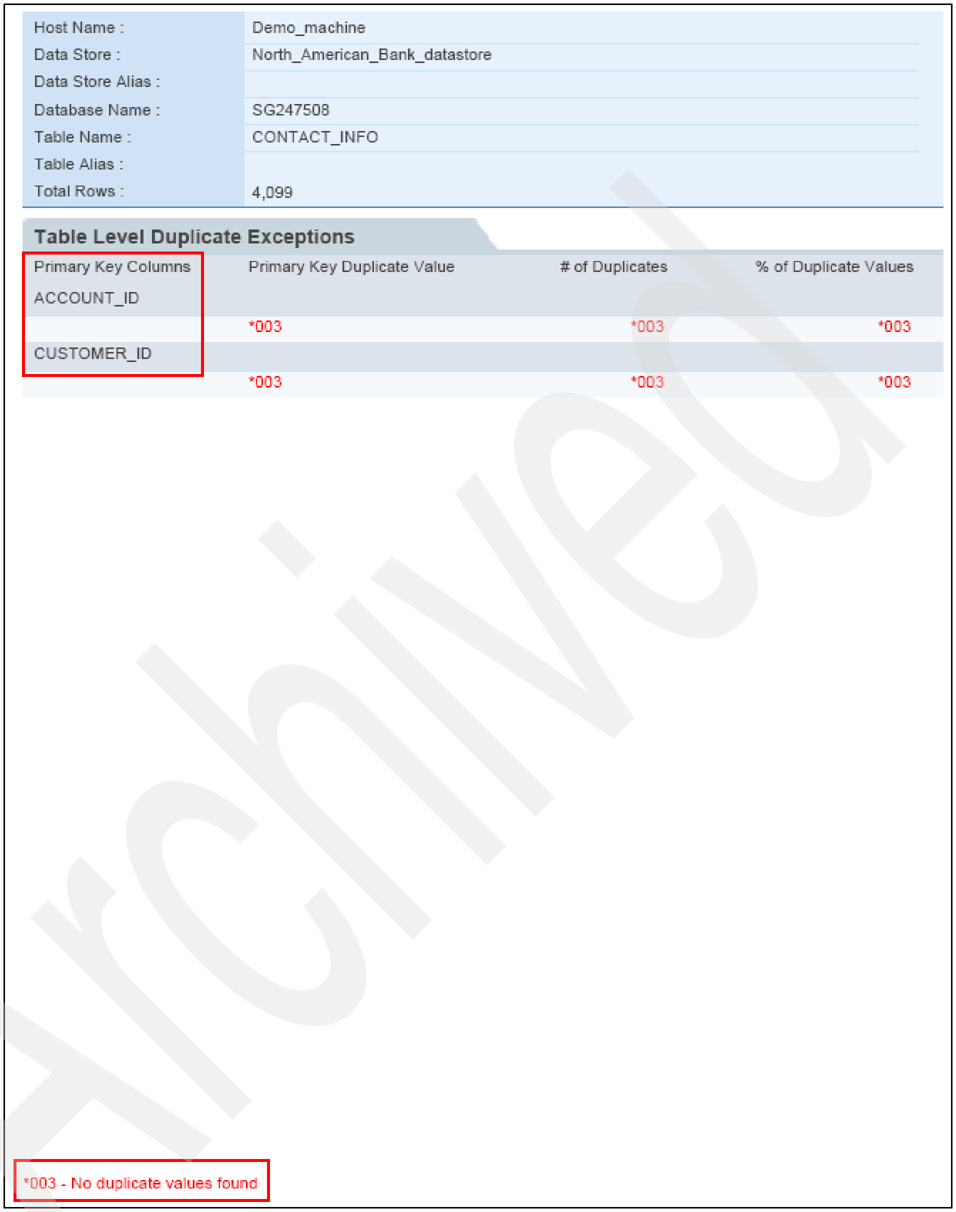
Defined Primary Key - Summary

Product Module :	WebSphere Information Analyzer
Project :	NA_CRM
Report Name :	Defined Primary Key - Summary CRM
Date/ Time Executed :	09/20/07 12:56:30
Generated By :	Per Hansen
Customer Description :	Defined Primary Key - Summary CRM
Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	ACCOUNT
Table Alias :	
PK Type Flag :	D

Defined Primary Key Summary

Primary Key Column(s)	PK Issue Flag	Length		Null Values		Default Values		Duplicate Values		Uniqueness	Distinct Values	Total Row
		Defined	Inferred	Total Rows	% of Total Rows	Total Rows	% of Total Rows	Total Rows	% of Total Rows	% of Total Rows	% of Total Rows	
ACCOUNT_ID	N	4	4	0	.000%	0	.000%	0	.000%	100.000%	100.000%	40

Figure 2-9 Primary Key Analysis "Defined Primary Key - Summary" report for the ACCOUNT table



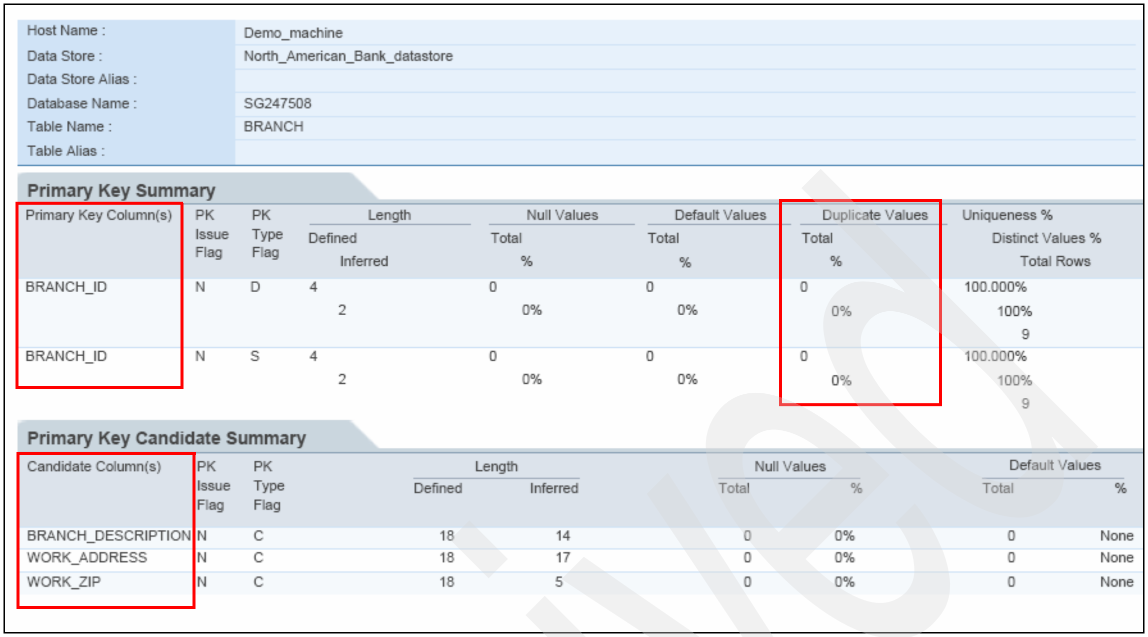
Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	CONTACT_INFO
Table Alias :	
Total Rows :	4,099

Table Level Duplicate Exceptions

Primary Key Columns	Primary Key Duplicate Value	# of Duplicates	% of Duplicate Values
ACCOUNT_ID			
	*003	*003	*003
CUSTOMER_ID			
	*003	*003	*003

*003 - No duplicate values found

Figure 2-10 Primary Key Analysis “Duplicate Exceptions” report for the CONTACT_INFO table



Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	BRANCH
Table Alias :	

Primary Key Summary

Primary Key Column(s)	PK Issue Flag	PK Type Flag	Length		Null Values		Default Values		Duplicate Values		Uniqueness %		
			Defined	Inferred	Total	%	Total	%	Total	%		Distinct Values %	Total Rows
BRANCH_ID	N	D	4	2	0	0%	0	0%	0	0%	100.000%	100%	9
BRANCH_ID	N	S	4	2	0	0%	0	0%	0	0%	100.000%	100%	9

Primary Key Candidate Summary

Candidate Column(s)	PK Issue Flag	PK Type Flag	Length		Null Values		Default Values	
			Defined	Inferred	Total	%	Total	%
BRANCH_DESCRIPTION	N	C	18	14	0	0%	0	None
WORK_ADDRESS	N	C	18	17	0	0%	0	None
WORK_ZIP	N	C	18	5	0	0%	0	None

Figure 2-11 Primary Key Analysis "Defined and Candidate Summary" report for the BRANCH table

Table 2-6 Primary Key violations

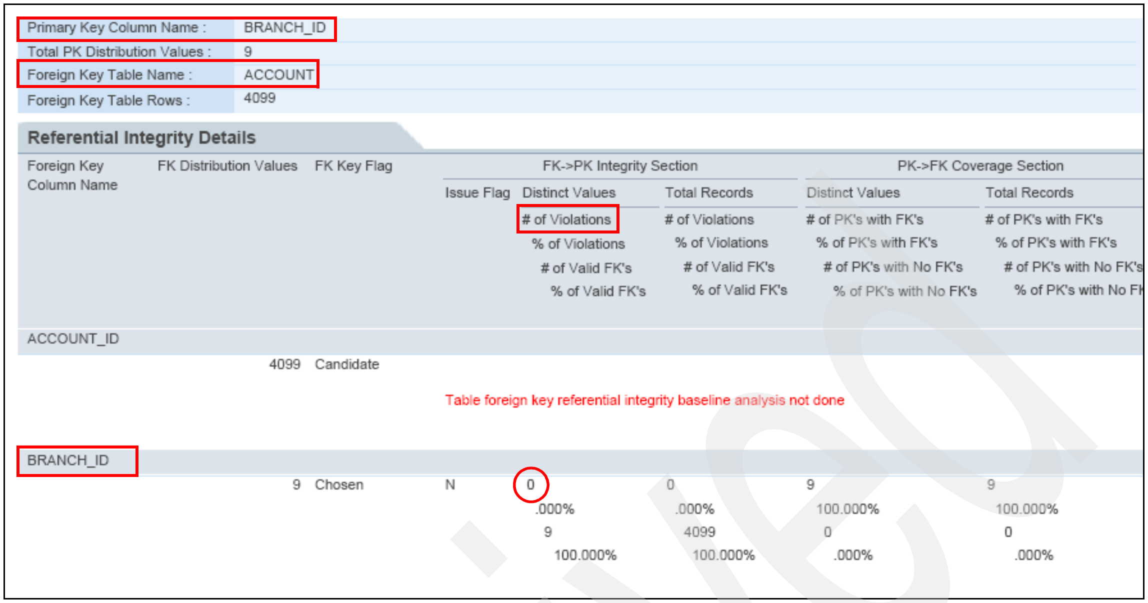
Primary Key Table Name	Primary Key Column Name	Invalid values
ACCOUNT	ACCOUNT_ID	

Foreign Key Analysis reports

These reports provide information about referential integrity violations for explicitly defined foreign keys, and can identify candidate foreign keys.

Figure 2-9 on page 440 shows the Foreign Key Analysis "Referential Integrity Detail" report for the ACCOUNT table. It identifies the primary key column (BRANCH_ID), foreign key column (BRANCH_ID) and referential integrity violations. This report shows no foreign key violations.

All violations should be documented for future correction in a table as shown in Table 2-7 on page 443.



Primary Key Column Name :	BRANCH_ID
Total PK Distribution Values :	9
Foreign Key Table Name :	ACCOUNT
Foreign Key Table Rows :	4099

Referential Integrity Details

Foreign Key Column Name	FK Distribution Values	FK Key Flag	FK->PK Integrity Section			PK->FK Coverage Section	
			Issue Flag	Distinct Values	Total Records	Distinct Values	Total Records
				# of Violations % of Violations # of Valid FK's % of Valid FK's	# of Violations % of Violations # of Valid FK's % of Valid FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No F
ACCOUNT_ID							
	4099	Candidate					
			Table foreign key referential integrity baseline analysis not done				
BRANCH_ID							
	9	Chosen	N	0 .000% 9 100.000%	0 .000% 4099 100.000%	9 100.000% 0 .000%	9 100.000% 0 .000%

Figure 2-12 Foreign Key Analysis “Referential Integrity Detail” report for the ACCOUNT table

Table 2-7 Referential Integrity violations

Foreign Key Table Name	Foreign Key Column Name	Primary Key Table Name	Primary Key Column Name	Invalid values
ACCOUNT	BRANCH_ID	BRANCH	BRANCH_ID	

Baseline Analysis reports

Baseline analysis is used to compare a prior version of analysis results with the current analysis results for a given data source. If differences between both versions are found, you can assess the significance of the change, such as whether you need to re-evaluate the migration strategy and execution plan.

Figure 2-13 on page 444 shows the partial report of the Baseline Structure “Current to Prior Structural Variances” report for the ACCOUNT table. It identifies the current to baseline variances in data types, length, precision, and scale for each of the columns in the table. This report shows no variances.

Figure 2-14 on page 445 and Figure 2-15 on page 446 show the partial report of the Baseline Structure “Current to Prior Variances” report for the ACCOUNT table. It identifies the variances in data content for each column in the table. This report shows no variations in the data content.

All the structural and data content variances should be documented for potential action.

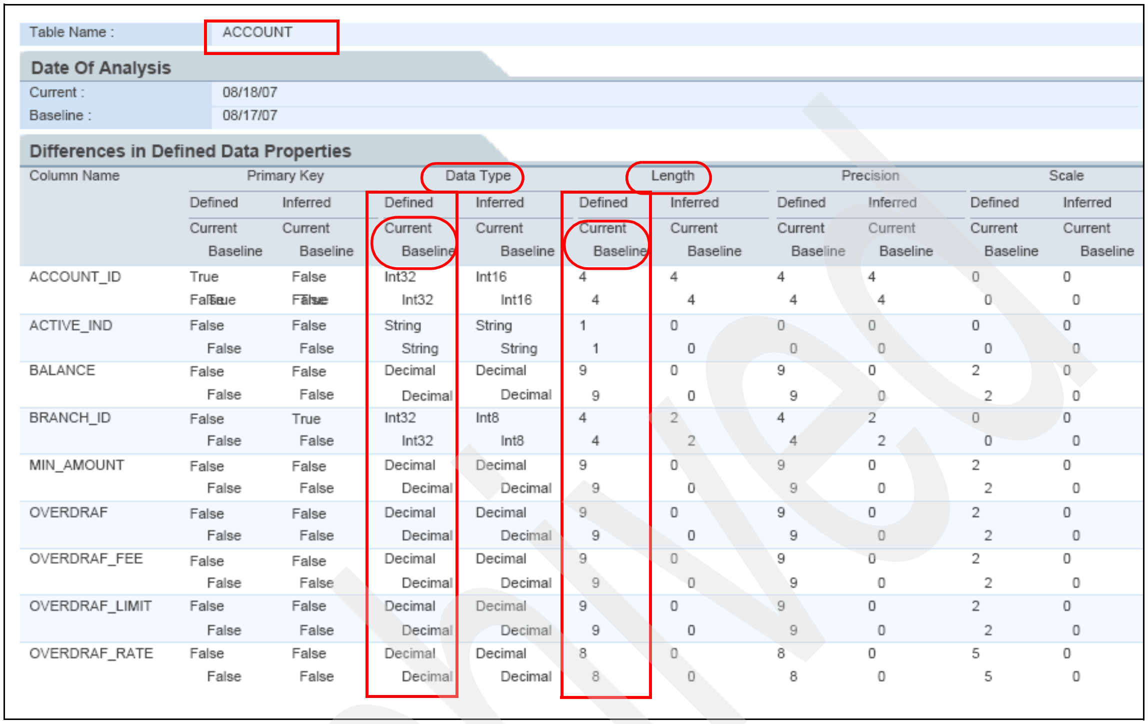


Table Name : ACCOUNT

Date Of Analysis

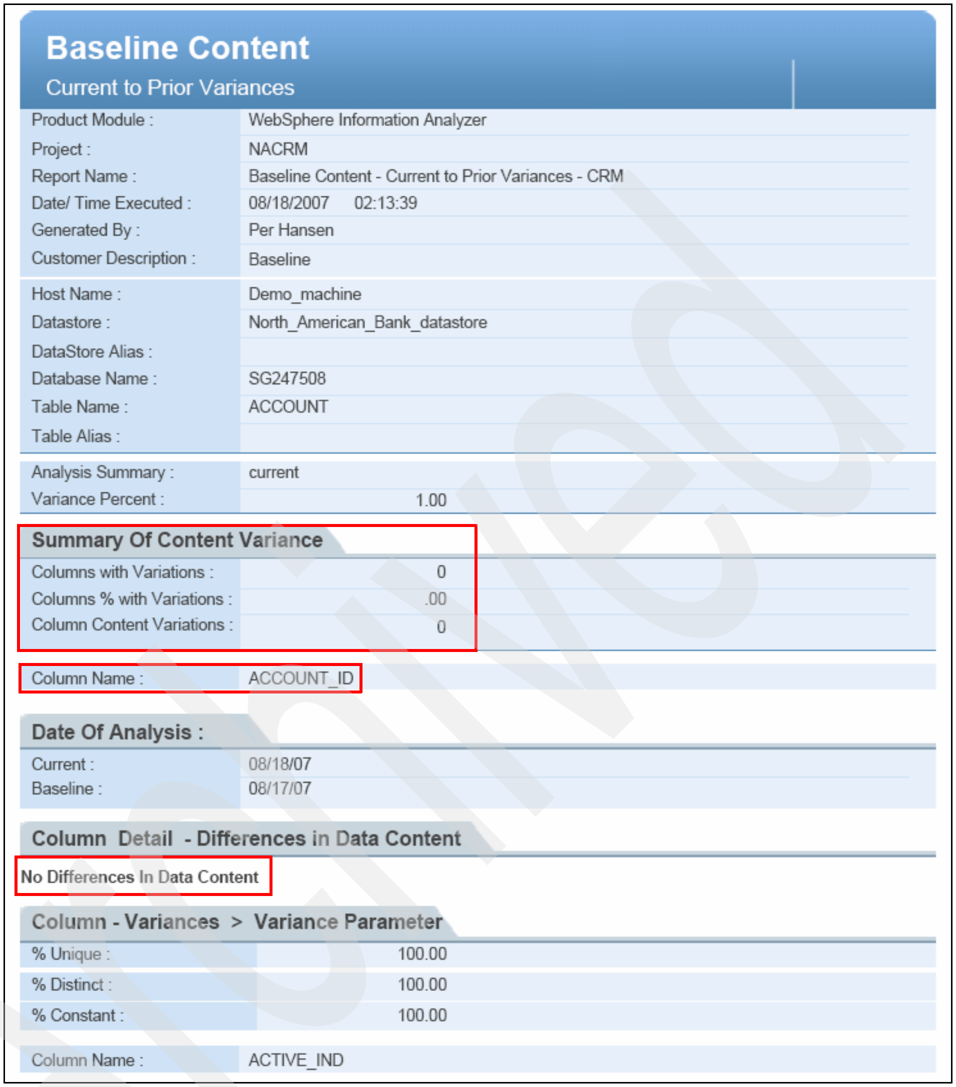
Current : 08/18/07

Baseline : 08/17/07

Differences in Defined Data Properties

Column Name	Primary Key		Data Type		Length		Precision		Scale	
	Defined	Inferred	Defined	Inferred	Defined	Inferred	Defined	Inferred	Defined	Inferred
	Current	Current	Current	Current	Current	Current	Current	Current	Current	Current
	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline
ACCOUNT_ID	True	False	Int32	Int16	4	4	4	4	0	0
	FalTrue	FTalse	Int32	Int16	4	4	4	4	0	0
ACTIVE_IND	False	False	String	String	1	0	0	0	0	0
	False	False	String	String	1	0	0	0	0	0
BALANCE	False	False	Decimal	Decimal	9	0	9	0	2	0
	False	False	Decimal	Decimal	9	0	9	0	2	0
BRANCH_ID	False	True	Int32	Int8	4	2	4	2	0	0
	False	False	Int32	Int8	4	2	4	2	0	0
MIN_AMOUNT	False	False	Decimal	Decimal	9	0	9	0	2	0
	False	False	Decimal	Decimal	9	0	9	0	2	0
OVERDRAF	False	False	Decimal	Decimal	9	0	9	0	2	0
	False	False	Decimal	Decimal	9	0	9	0	2	0
OVERDRAF_FEE	False	False	Decimal	Decimal	9	0	9	0	2	0
	False	False	Decimal	Decimal	9	0	9	0	2	0
OVERDRAF_LIMIT	False	False	Decimal	Decimal	9	0	9	0	2	0
	False	False	Decimal	Decimal	9	0	9	0	2	0
OVERDRAF_RATE	False	False	Decimal	Decimal	8	0	8	0	5	0
	False	False	Decimal	Decimal	8	0	8	0	5	0

Figure 2-13 Baseline Structure "Current to Prior Structural Variances" report for the ACCOUNT table



Baseline Content

Current to Prior Variances

Product Module :	WebSphere Information Analyzer
Project :	NACRM
Report Name :	Baseline Content - Current to Prior Variances - CRM
Date/ Time Executed :	08/18/2007 02:13:39
Generated By :	Per Hansen
Customer Description :	Baseline

Host Name :	Demo_machine
Datastore :	North_American_Bank_datastore
DataStore Alias :	
Database Name :	SG247508
Table Name :	ACCOUNT
Table Alias :	

Analysis Summary :	current
Variance Percent :	1.00

Summary Of Content Variance

Columns with Variations :	0
Columns % with Variations :	.00
Column Content Variations :	0

Column Name :	ACCOUNT_ID

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

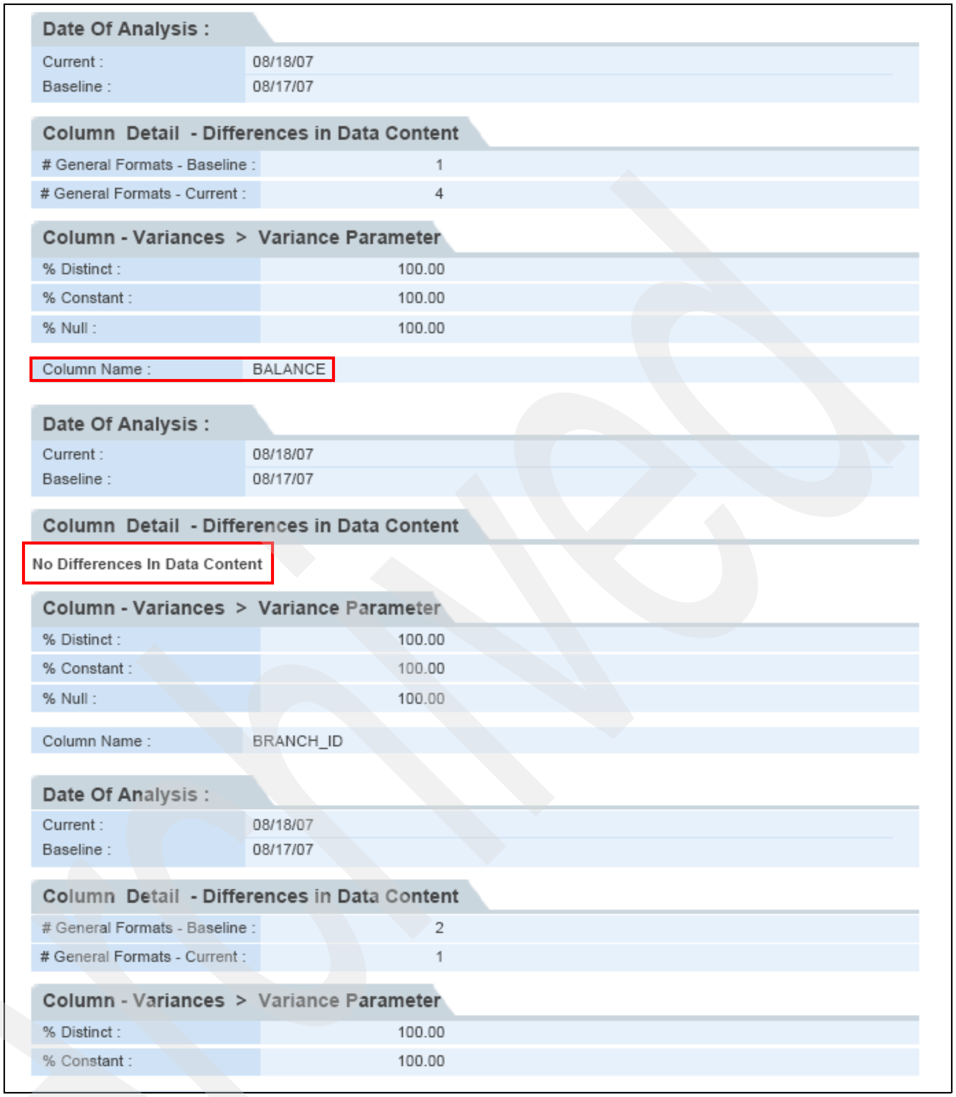
No Differences In Data Content

Column - Variances > Variance Parameter

% Unique :	100.00
% Distinct :	100.00
% Constant :	100.00

Column Name :	ACTIVE_IND

Figure 2-14 Baseline Structure “Current to Prior Variances” report for the ACCOUNT table 1/2



Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	1
# General Formats - Current :	4

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00
% Null :	100.00

Column Name :	BALANCE

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

No Differences In Data Content

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00
% Null :	100.00

Column Name :	BRANCH_ID

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	2
# General Formats - Current :	1

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00

Figure 2-15 Baseline Structure "Current to Prior Variances" report for the ACCOUNT table 2/2

IBM WebSphere AuditStage business rule compliance

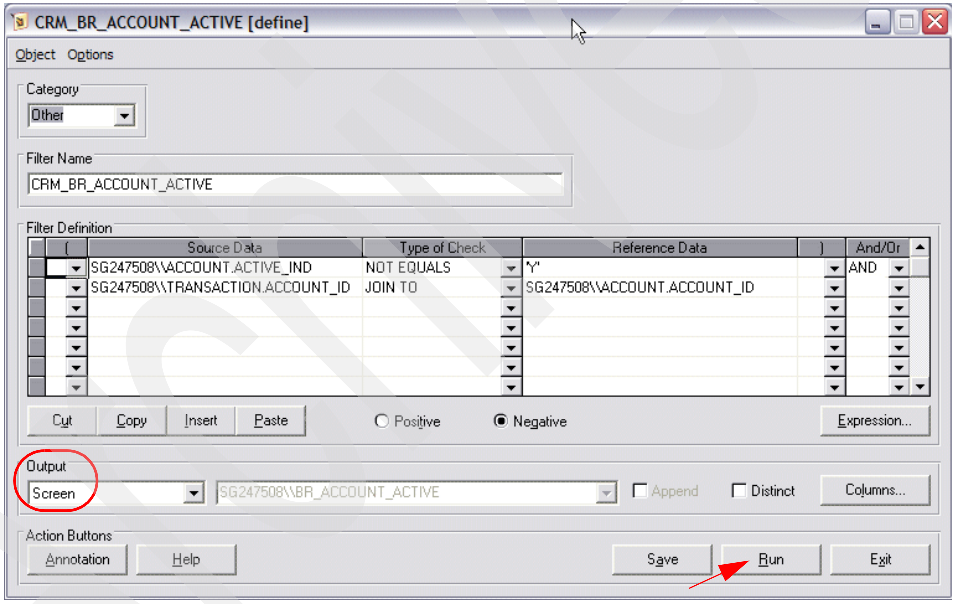
Business Rule Compliance evaluates the quality of data in terms of specific business rules involving multiple data fields within or across records (or rows) that are logically related. IBM WebSphere AuditStage is used to perform this check. In this section, we do not describe the process of building the data filter because that is described in 1.12, "IBM WebSphere AuditStage business rule validation" on page 336 but focus on the completed data filter specification and the results of executing the business rule.

The business rules described in Table 2-4 on page 436 are verified using the data filters developed using IBM WebSphere AuditStage.

1: Check ACTIVE_IND column and transactions

Figure 2-16 on page 447 shows the data filter for verifying the business rule that an account in the ACCOUNT table can only have a transaction if the ACTIVE_IND column is set to "Y". After building the data filter which checks whether there are transactions for an account that has the ACTIVE_IND not equal to "Y", click **Run** to execute the data filter.

Figure 2-17 on page 448 shows the number of exceptions. One row qualifies which is shown in Figure 2-18 on page 448.



Category: Other

Filter Name: CRM_BR_ACCOUNT_ACTIVE

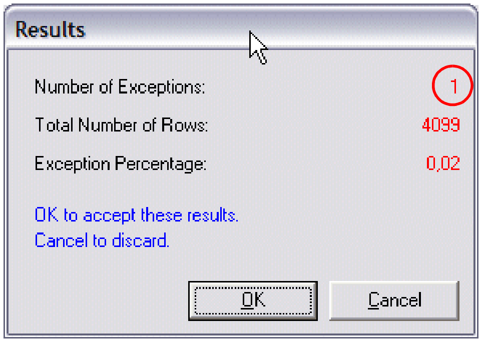
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	SG247508\\ACCOUNT.ACTIVE_IND	NOT EQUALS	'Y'		AND
	SG247508\\TRANSACTION.ACCOUNT_ID	JOIN TO	SG247508\\ACCOUNT.ACCOUNT_ID		

○ Positive ● Negative

Output: Screen SG247508\\BR_ACCOUNT_ACTIVE □ Append □ Distinct

Figure 2-16 Check ACTIVE_IND column and transactions 1/3



Results

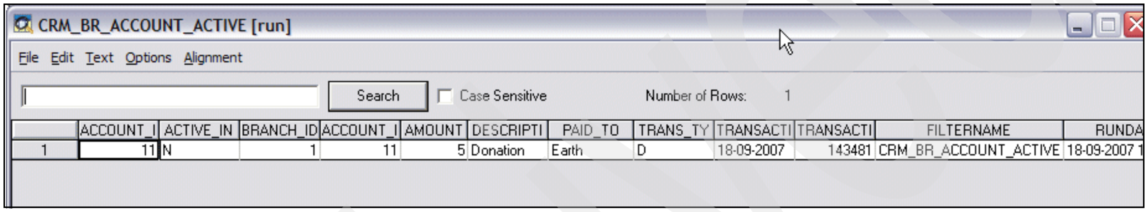
Number of Exceptions: 1

Total Number of Rows: 4099

Exception Percentage: 0,02

OK to accept these results.
Cancel to discard.

Figure 2-17 Check ACTIVE_IND column and transactions 3/3



CRM_BR_ACCOUNT_ACTIVE [run]

□ Case Sensitive

Number of Rows: 1

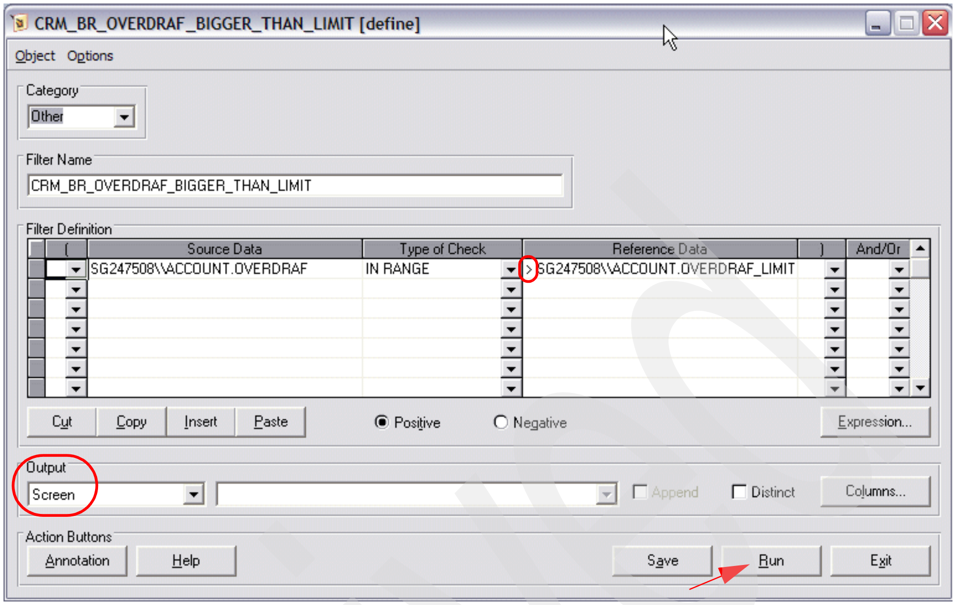
	ACCOUNT_I	ACTIVE_IN	BRANCH_ID	ACCOUNT_I	AMOUNT	DESCRIPTI	PAID_TO	TRANS_TY	TRANSACTI	TRANSACTI	FILTERNAME	RUNDA
1	11	N	1	11	5	Donation	Earth	D	18-09-2007	143481	CRM_BR_ACCOUNT_ACTIVE	18-09-2007 1

Figure 2-18 Check ACTIVE_IND column and transactions 3/3

2: Check OVERDRAFT column and OVERDRAFT_LIMIT

Figure 2-19 on page 449 shows the data filter for verifying the business rule that the OVERDRAFT column value cannot exceed the OVERDRAFT_LIMIT column value in the ACCOUNT table. After building the data filter which checks whether the OVERDRAFT column value is greater than the OVERDRAFT_LIMIT column value, click **Run** to execute the data filter.

Figure 2-20 on page 449 shows the number of exceptions. Two rows qualify as shown in Figure 2-21 on page 449.



CRM_BR_OVERDRAF_BIGGER_THAN_LIMIT [define]

Category: Other

Filter Name: CRM_BR_OVERDRAF_BIGGER_THAN_LIMIT

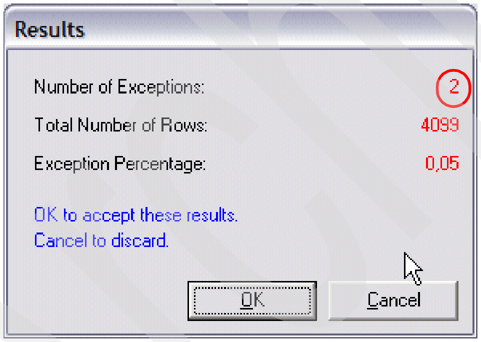
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	SG247508\\ACCOUNT.OVERDRAF	IN RANGE	> SG247508\\ACCOUNT.OVERDRAF_LIMIT		

● Positive ○ Negative

Output: Screen □ Append □ Distinct

Figure 2-19 Check OVERDRAFT column and OVERDRAFT_LIMIT 1/3

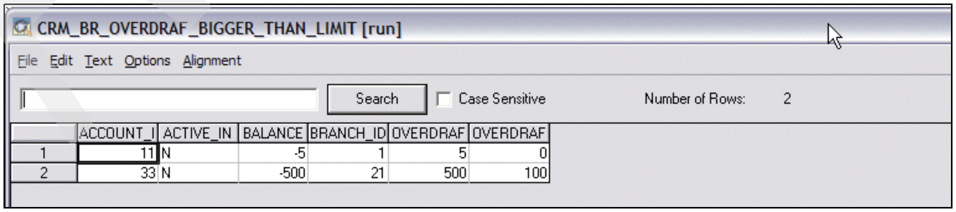


Results

Number of Exceptions:	2
Total Number of Rows:	4099
Exception Percentage:	0,05

OK to accept these results.
Cancel to discard.

Figure 2-20 Check OVERDRAFT column and OVERDRAFT_LIMIT 2/3



CRM_BR_OVERDRAF_BIGGER_THAN_LIMIT [run]

□ Case Sensitive Number of Rows: 2

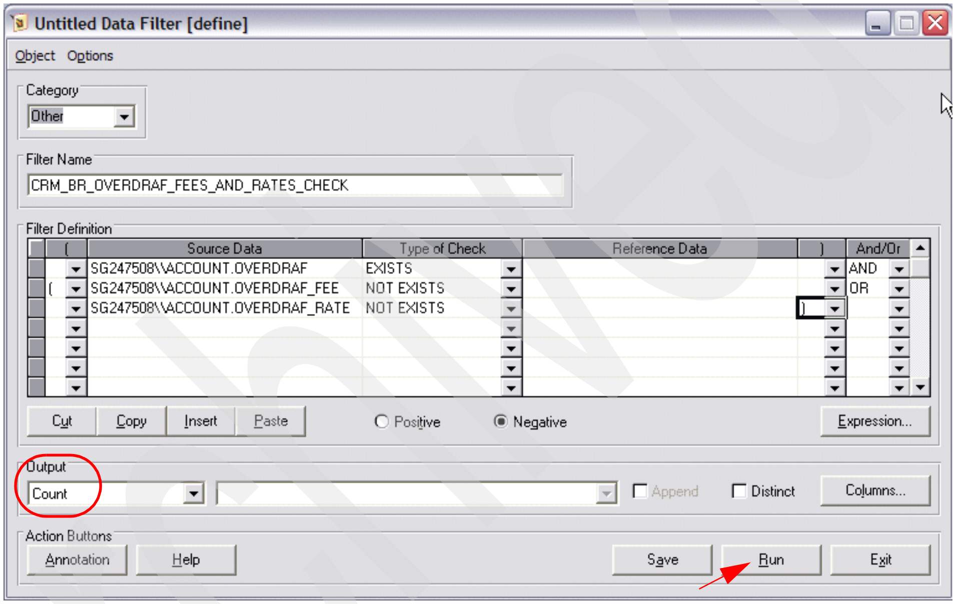
	ACCOUNT_I	ACTIVE_IN	BALANCE	BRANCH_ID	OVERDRAF	OVERDRAF
1	11	N	-5	1	5	0
2	33	N	-500	21	500	100

Figure 2-21 Check OVERDRAFT column and OVERDRAFT_LIMIT 3/3

3: Check OVERDRAFT column and associated fees and rates

Figure 2-22 shows the data filter for verifying the business rule that requires that when the OVERDRAFT column has a value, the OVERDRAFT_RATE and OVERDRAFT_FEE columns in the ACCOUNT table must also have a value. After building the data filter which checks columns that have an OVERDRAFT value cannot have a missing value in either the OVERDRAFT_RATE and OVERDRAFT_FEE columns, click **Run** to execute the data filter.

Figure 2-23 shows the number of exceptions. One row qualifies as shown in Figure 2-24 on page 451.



Category: Other

Filter Name: CRM_BR_OVERDRAF_FEES_AND_RATES_CHECK

Filter Definition

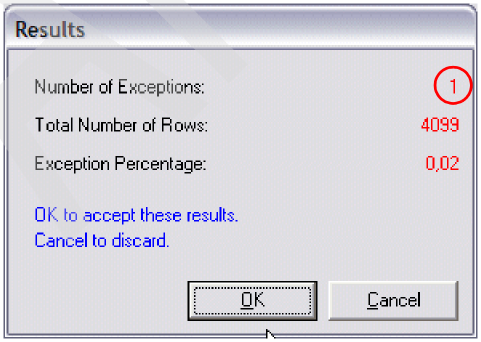
(	Source Data	Type of Check	Reference Data	)	And/Or
	SG247508\\ACCOUNT.OVERDRAF	EXISTS			AND
(	SG247508\\ACCOUNT.OVERDRAF_FEE	NOT EXISTS			OR
	SG247508\\ACCOUNT.OVERDRAF_RATE	NOT EXISTS		)	

○ Positive ● Negative

Output: Count

□ Append □ Distinct

Figure 2-22 Check OVERDRAFT column and associated fees and rates 1/3



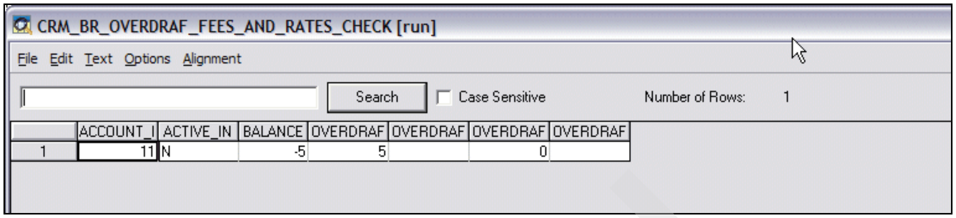
Number of Exceptions: 1

Total Number of Rows: 4099

Exception Percentage: 0,02

OK to accept these results.
Cancel to discard.

Figure 2-23 Check OVERDRAFT column and associated fees and rates 2/3



CRM_BR_OVERDRAF_FEES_AND_RATES_CHECK [run]

Number of Rows: 1

	ACCOUNT_I	ACTIVE_IN	BALANCE	OVERDRAF	OVERDRAF	OVERDRAF	OVERDRAF
1	11	N	-5	5		0	

Figure 2-24 Check OVERDRAFT column and associated fees and rates 3/3

2.5.5 Northern California Bank analysis

As mentioned earlier, Northern California Bank's core services are the target of the migration of core services from the North American Bank.

Using metadata information available in dictionaries, documentation and the relational catalogs, the keys, code fields, indicator fields, and referential integrity relationships (both implicit and explicit) to be profiled were identified as shown in Table 2-8 on page 452.

Note: For convenience, we chose to include all the columns in all the tables in our data profiling effort. In the real world, however, to avoid report and information overload, you would apply the 80/20 rule and focus on only the critical master data as described in 1.2.1, "Data assessment approach" on page 5.

We are only including selected portions of the generated reports (for some of the fields shown in Table 2-8 on page 452) of the data profiling effort in this section. You can download the complete reports from the IBM Redbooks Web site:

`ftp://www.redbooks.ibm.com/redbooks/SG247508/`

Note: We describe the process of generating reports in 1.14, "Reports" on page 394 and do not repeat that information here. We show only the relevant portions of the reports here.

Business rules involving data elements in one or more tables are shown in Table 2-9 on page 452. These are verified using IBM WebSphere AuditStage.

Table 2-8 Main keys, codes, and indicators in Northern California Bank's core services

Table	Key column(s)	Code column	Indicator column	RI relationships
CUSTOMER	ID	► ZIP ► PREF_LANG ► TYPE	GENDER	► BRANCH to the BRANCH table ► PREF_LANG to the LANGUAGES table ► ADVISOR to the EMPLOYEE table
ACCOUNT	ID	► TYPE ► CURRENCY		► CURRENCY to the CURRENCY table ► TYPE to the ACCTYPE table
LOAN	LOAN_ID			ACCOUNT_ID to the ACCOUNT table
COLLATERAL	ACCOUNT, UPDATED	► TYPE ► STATUS		TYPE to the ACCTYPE table
ACCTYPE	TYPE	► FEEFRQ ► CURRENCY		CURRENCY to the CURRENCY table
EMPLOYEE	ID			BRANCH to the BRANCH table
BRANCH	ID	► ZIP ► COUNTRY		COUNTRY to the COUNTRY table
COUNTRY	CTRY3 (not a defined primary key)			
TRANSACTION	ACCOUNT, UPDATED	CODE		ACCOUNT to the ACCOUNT table
CURRENCY	CURRENCY (not a defined primary key)			
LANGUAGES	LAN3 (not a defined primary key)			

Table 2-9 Business rules in Northern California Bank's core services

Serial number	Description
1	Customer data in both the CUSTOMER and BCUSTOMER tables can only be updated by the employee who is the advisor to that customer. The ADVISOR column in the CUSTOMER and BCUSTOMER tables identify their advisor employee in the EMPLOYEE table. The ADVISOR column might be null. The BY column in the CUSTOMER and BCUSTOMER tables should correspond to the relevant USERID column in the EMPLOYEE table.

<table>
<tr><th>Serial number</th><th>Description</th></tr>
<tr><td>2</td><td>The CLASS column in the CUSTOMER table is computed according to the following algorithm coded in SQL.
<pre>
update customer c set class = (select class from
(
   select id,10 - sum(class) as class from
   (
      select id
           ,name
           ,case
              when average >= 5000000 then 4
              when average < 5000000
               and average >= 2500000 then 3
              when average < 2500000
               and average >= 1000000 then 2
              when average < 1000000
               and average >   0 then 1
            end as class
        from status
        where account_type = 'SS'
      union
      select id
           ,name
           ,case
              when average >= 100000 then 3
              when average < 100000
               and average >= 50000 then 2
              when average <   50000
               and average >   0 then 1
            end as class
        from status
        where account_type = 'SC'
      union
      select id
           ,name
           ,3 as class
        from status
        where account_type = 'LN'
) x
   group by id,name
) y
 where c.id = y.id);
</pre></td></tr>
</table>

Column Analysis reports

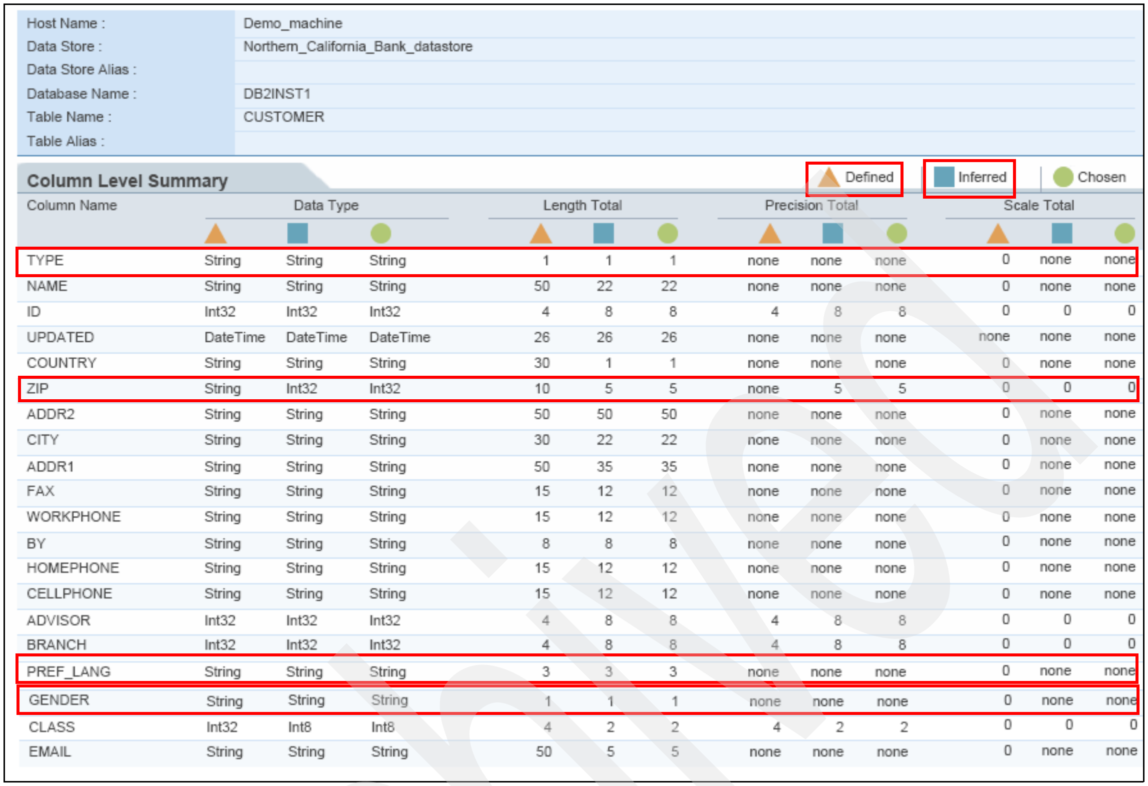
As mentioned earlier, these reports provide information about column values for completeness, validity, structure, and format as described in 1.7, "Column analysis" on page 109. These reports are used to review the columns for domain, structure and format integrity.

Figure 2-25 on page 455 shows the Column Inferences "Type & Length Properties (Defined/Inferred/Chosen)" report for the CUSTOMER table. It identifies the data type, length, precision and scale of each column as defined in the metadata and inferred from the data content. Any selected status of each of these categories is also shown here. This information (along with those relating to other columns in other tables) is recorded in Table 2-13 on page 475. Our focus is on keys, codes, and indicators as highlighted.

Figure 2-26 on page 456 shows the Column Frequency "Frequency By Frequency" report for the GENDER column in the CUSTOMER table. It shows valid values such as "0", "1", "M", and "F", and an invalid value of "X".

Figure 2-8 on page 439 shows the Column Domain Values "Completeness and Validity Summary" report for the CUSTOMER table. It shows zero occurrences of incomplete values, and 138 occurrences of invalid values in the GENDER column.

Document all the valid and invalid values for a column in a table as shown in Table 2-10 on page 457.



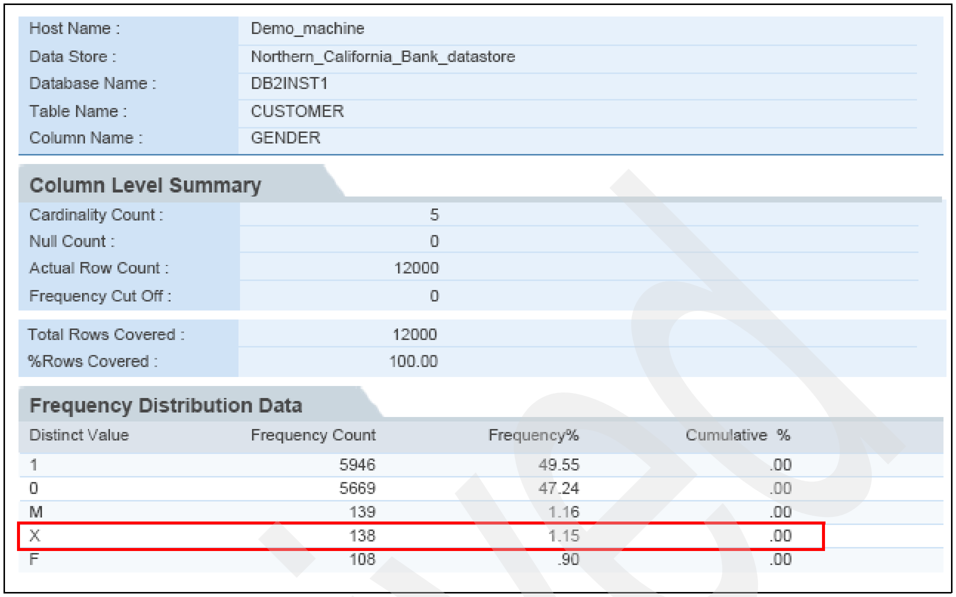
Host Name : Demo_machine
Data Store : Northern_California_Bank_datastore
Data Store Alias :
Database Name : DB2INST1
Table Name : CUSTOMER
Table Alias :

Column Level Summary

▲ Defined ■ Inferred ● Chosen

Column Name	Data Type			Length Total			Precision Total			Scale Total		
	▲	■	●	▲	■	●	▲	■	●	▲	■	●
TYPE	String	String	String	1	1	1	none	none	none	0	none	none
NAME	String	String	String	50	22	22	none	none	none	0	none	none
ID	Int32	Int32	Int32	4	8	8	4	8	8	0	0	0
UPDATED	DateTime	DateTime	DateTime	26	26	26	none	none	none	none	none	none
COUNTRY	String	String	String	30	1	1	none	none	none	0	none	none
ZIP	String	Int32	Int32	10	5	5	none	5	5	0	0	0
ADDR2	String	String	String	50	50	50	none	none	none	0	none	none
CITY	String	String	String	30	22	22	none	none	none	0	none	none
ADDR1	String	String	String	50	35	35	none	none	none	0	none	none
FAX	String	String	String	15	12	12	none	none	none	0	none	none
WORKPHONE	String	String	String	15	12	12	none	none	none	0	none	none
BY	String	String	String	8	8	8	none	none	none	0	none	none
HOMEPHONE	String	String	String	15	12	12	none	none	none	0	none	none
CELLPHONE	String	String	String	15	12	12	none	none	none	0	none	none
ADVISOR	Int32	Int32	Int32	4	8	8	4	8	8	0	0	0
BRANCH	Int32	Int32	Int32	4	8	8	4	8	8	0	0	0
PREF_LANG	String	String	String	3	3	3	none	none	none	0	none	none
GENDER	String	String	String	1	1	1	none	none	none	0	none	none
CLASS	Int32	Int8	Int8	4	2	2	4	2	2	0	0	0
EMAIL	String	String	String	50	5	5	none	none	none	0	none	none

Figure 2-25 Column Inferences “Type & Length Properties (Defined/Inferred/Chosen)” report for the CUSTOMER table



Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Database Name :	DB2INST1
Table Name :	CUSTOMER
Column Name :	GENDER

Column Level Summary

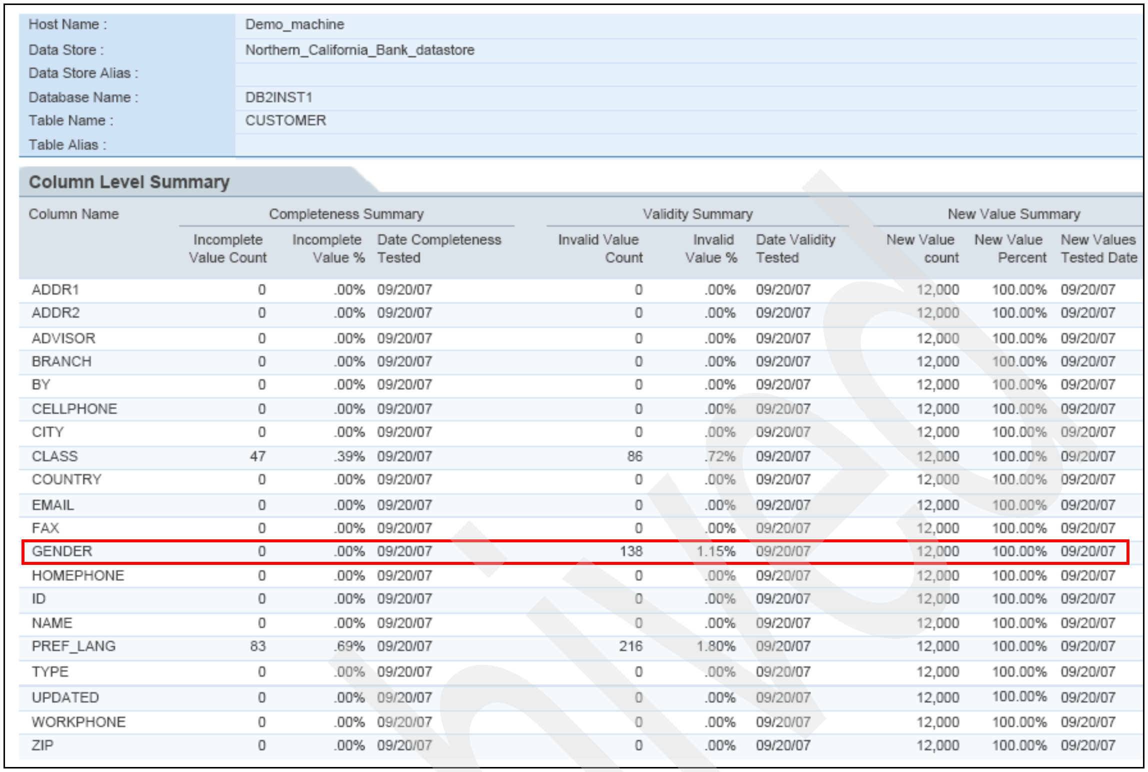
Cardinality Count :	5
Null Count :	0
Actual Row Count :	12000
Frequency Cut Off :	0

Total Rows Covered :	12000
%Rows Covered :	100.00

Frequency Distribution Data

Distinct Value	Frequency Count	Frequency%	Cumulative %
1	5946	49.55	.00
0	5669	47.24	.00
M	139	1.16	.00
X	138	1.15	.00
F	108	.90	.00

Figure 2-26 Column Frequency "Frequency By Frequency" report for the GENDER column in the CUSTOMER table



Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	CUSTOMER
Table Alias :	

Column Level Summary

Column Name	Completeness Summary			Validity Summary			New Value Summary		
	Incomplete Value Count	Incomplete Value %	Date Completeness Tested	Invalid Value Count	Invalid Value %	Date Validity Tested	New Value count	New Value Percent	New Values Tested Date
ADDR1	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
ADDR2	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
ADVISOR	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
BRANCH	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
BY	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
CELLPHONE	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
CITY	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
CLASS	47	.39%	09/20/07	86	.72%	09/20/07	12,000	100.00%	09/20/07
COUNTRY	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
EMAIL	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
FAX	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
GENDER	0	.00%	09/20/07	138	1.15%	09/20/07	12,000	100.00%	09/20/07
HOMEPHONE	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
ID	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
NAME	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
PREF_LANG	83	.69%	09/20/07	216	1.80%	09/20/07	12,000	100.00%	09/20/07
TYPE	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
UPDATED	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
WORKPHONE	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07
ZIP	0	.00%	09/20/07	0	.00%	09/20/07	12,000	100.00%	09/20/07

Figure 2-27 Column Domain Values “Completeness and Validity Summary” report for the CUSTOMER table

Table 2-10 Valid and invalid column values

Column name	Valid values	Invalid values
GENDER in the CUSTOMER table	“0”, “1”, “M”, “F”	“X”

Primary Key Analysis reports

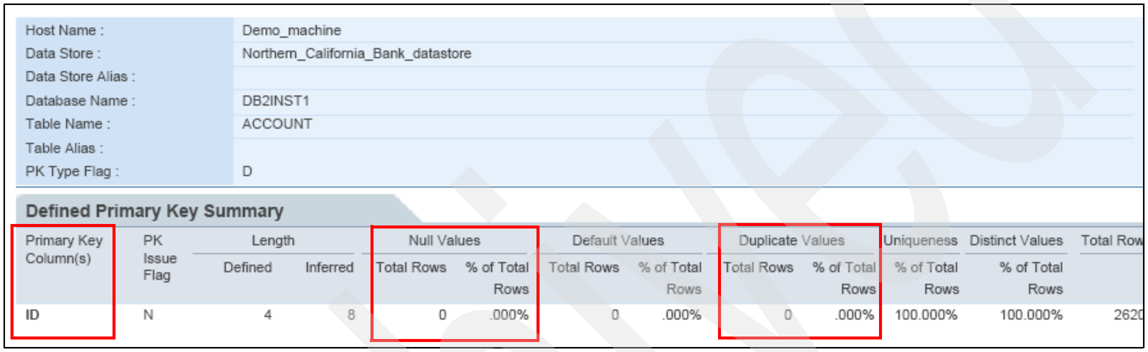
As mentioned earlier, these reports provide information about candidate primary keys as well as duplicate records in the table.

Figure 2-28 on page 458 shows the Primary Key Analysis “Defined Primary Key - Summary” report for the ACCOUNT table. It identifies the length (defined and inferred), number of null values, default values, uniqueness values, distinct values, and number of rows in the table. This report shows no duplicate records or null values in the table.

Figure 2-29 on page 459 shows the Primary Key Analysis “Duplicate Exceptions” report for the CUSTOMER table. It identifies duplicate values in each column of the primary key. There are none in this case.

Figure 2-30 on page 460 shows the Primary Key Analysis “Defined & Candidate Summary” report for the BRANCH table. It identifies duplicate values in each column of the primary key. There are none in this case, but candidate keys are identified.

Any duplicate rows or tables with no primary key (but candidate primary keys) should be identified and documented as shown in Table 2-11 on page 460.

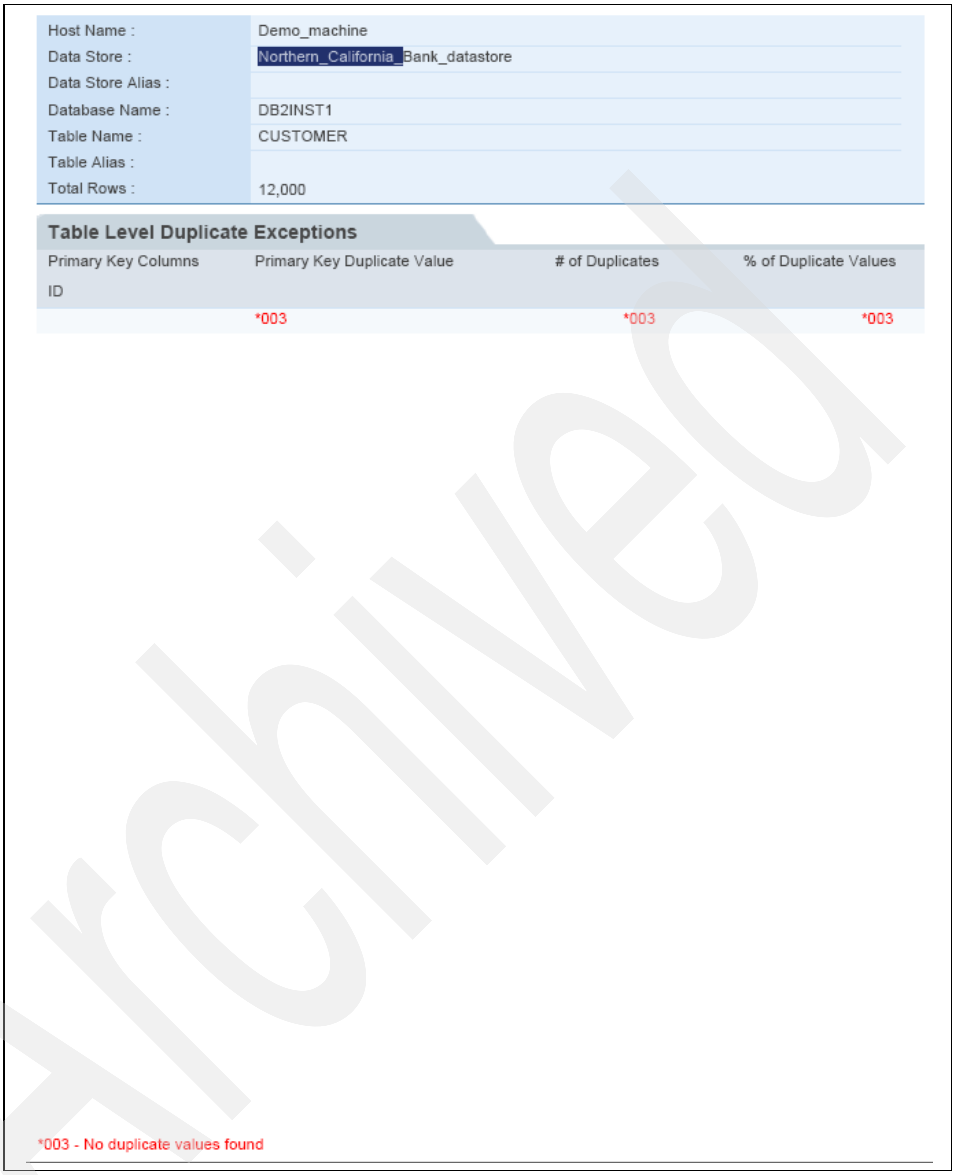


Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	ACCOUNT
Table Alias :	
PK Type Flag :	D

Defined Primary Key Summary

Primary Key Column(s)	PK Issue Flag	Length		Null Values		Default Values		Duplicate Values		Uniqueness	Distinct Values	Total Row
		Defined	Inferred	Total Rows	% of Total Rows	Total Rows	% of Total Rows	Total Rows	% of Total Rows	% of Total Rows	% of Total Rows	
ID	N	4	8	0	.000%	0	.000%	0	.000%	100.000%	100.000%	2620

Figure 2-28 Primary Key Analysis “Defined Primary Key - Summary” report for the ACCOUNT table



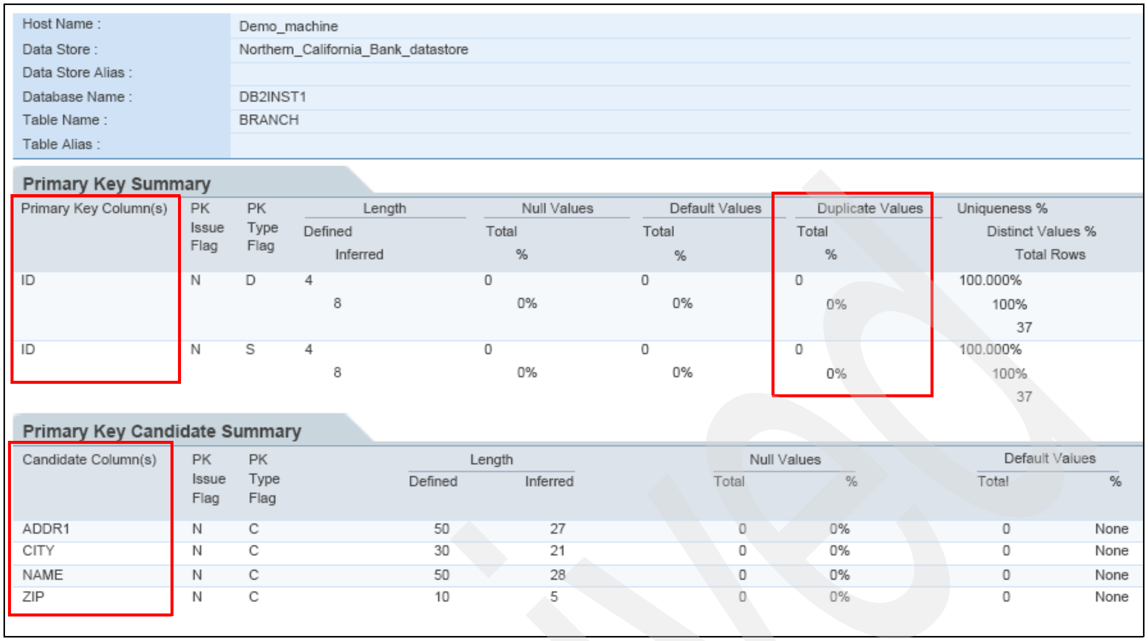
Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	CUSTOMER
Table Alias :	
Total Rows :	12,000

Table Level Duplicate Exceptions

Primary Key Columns	Primary Key Duplicate Value	# of Duplicates	% of Duplicate Values
ID			
	*003	*003	*003

*003 - No duplicate values found

Figure 2-29 Primary Key Analysis “Duplicate Exceptions” report for the CUSTOMER table



Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	BRANCH
Table Alias :	

Primary Key Summary

Primary Key Column(s)	PK Issue Flag	PK Type Flag	Length		Null Values		Default Values		Duplicate Values		Uniqueness %		
			Defined	Inferred	Total	%	Total	%	Total	%		Distinct Values %	Total Rows
ID	N	D	4	8	0	0%	0	0%	0	0%	100.000%	100%	37
ID	N	S	4	8	0	0%	0	0%	0	0%	100.000%	100%	37

Primary Key Candidate Summary

Candidate Column(s)	PK Issue Flag	PK Type Flag	Length		Null Values		Default Values	
			Defined	Inferred	Total	%	Total	%
ADDR1	N	C	50	27	0	0%	0	None
CITY	N	C	30	21	0	0%	0	None
NAME	N	C	50	28	0	0%	0	None
ZIP	N	C	10	5	0	0%	0	None

Figure 2-30 Primary Key Analysis "Defined & Candidate Summary" report for the BRANCH table

Table 2-11 Primary Key violations

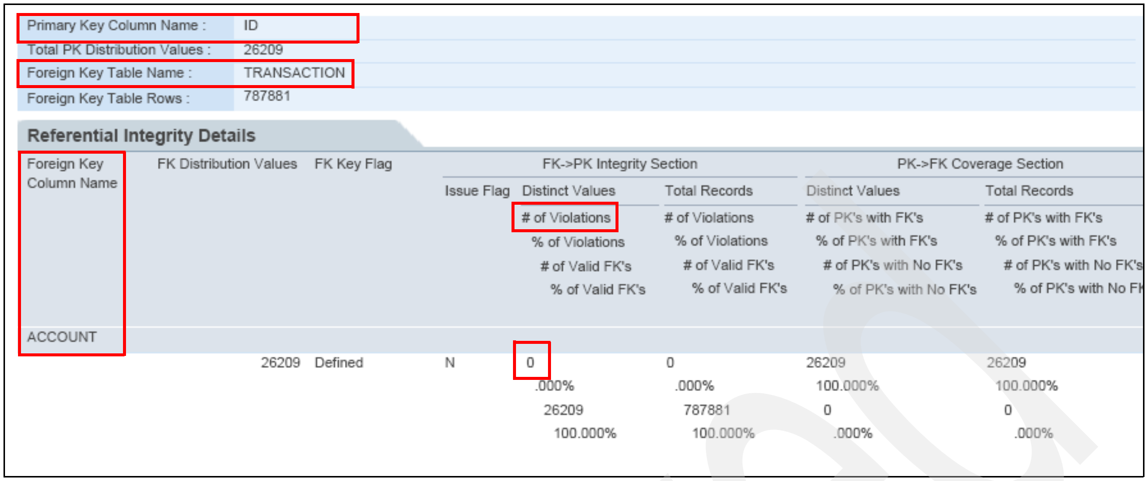
Primary Key Table Name	Primary Key Column Name	Invalid values
ACCOUNT	ID	

Foreign Key Analysis reports

These reports provide information about referential integrity violations for explicitly defined foreign keys, and can identify candidate foreign keys.

Figure 2-9 on page 440 shows the Foreign Key Analysis "Referential Integrity Detail" report for the TRANSACTION table. It identifies the primary key column (ID), foreign key column (ACCOUNT) and referential integrity violations. This report shows no foreign key violations.

All violations should be documented for future correction as shown in Table 2-12 on page 461.



Primary Key Column Name :	ID
Total PK Distribution Values :	26209
Foreign Key Table Name :	TRANSACTION
Foreign Key Table Rows :	787881

Referential Integrity Details

Foreign Key Column Name	FK Distribution Values	FK Key Flag	FK->PK Integrity Section			PK->FK Coverage Section	
			Issue Flag	Distinct Values	Total Records	Distinct Values	Total Records
				# of Violations % of Violations # of Valid FK's % of Valid FK's	# of Violations % of Violations # of Valid FK's % of Valid FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No F
ACCOUNT							
	26209	Defined	N	0 .000% 26209 100.000%	0 .000% 787881 100.000%	26209 100.000% 0 .000%	26209 100.000% 0 .000%

Figure 2-31 Foreign Key Analysis "Referential Integrity Detail" report for the TRANSACTION table

Table 2-12 Referential Integrity violations

Foreign Key Table Name	Foreign Key Column Name	Primary Key Table Name	Primary Key Column Name	Invalid values
TRANSACTION	ACCOUNT	ACCOUNT	ID	

Baseline Analysis reports

Baseline analysis is used to compare a prior version of analysis results with the current analysis results for a given data source. If differences between both versions are found, you can assess the significance of the change, such as whether you need to re-evaluate the migration strategy and execution plan. When assessing the progress of development and data quality assessment work, Baseline Analyses are particularly useful to show the changes between one run of a process and the next, and whether the changes made to correct issues have resulted in improved output.

Figure 2-32 on page 462 through Figure 2-34 on page 463 show the partial report of the Baseline Structure "Current to Prior Structural Variances" report for the CUSTOMER table. It identifies the current to baseline variances in data types, length, precision, and scale for each of the columns in the table. This report shows no variances.

Figure 2-35 on page 464 and Figure 2-36 on page 465 show the partial report of the Baseline Structure "Current to Prior Variances" report for the CUSTOMER table. It identifies the variances in data content for each column in the table. This report shows a number of variations in data content by column. You should then use the column frequency distribution to identify the specify content differences (this is not shown here).

All the structural and data content variances should be documented for potential action.

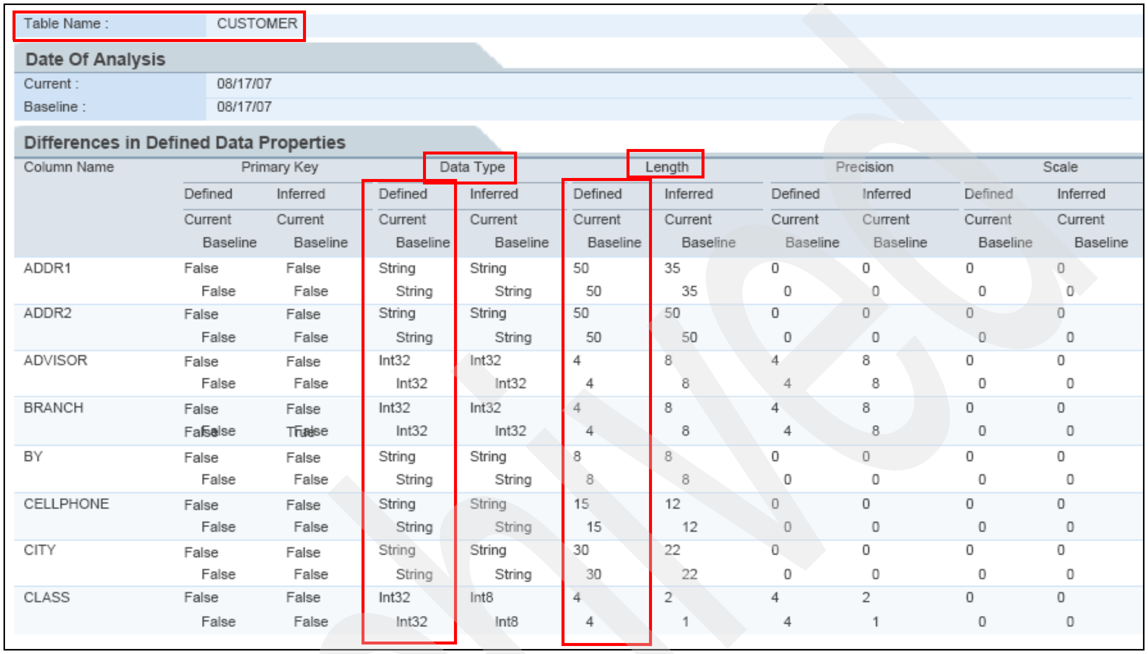


Table Name :	CUSTOMER

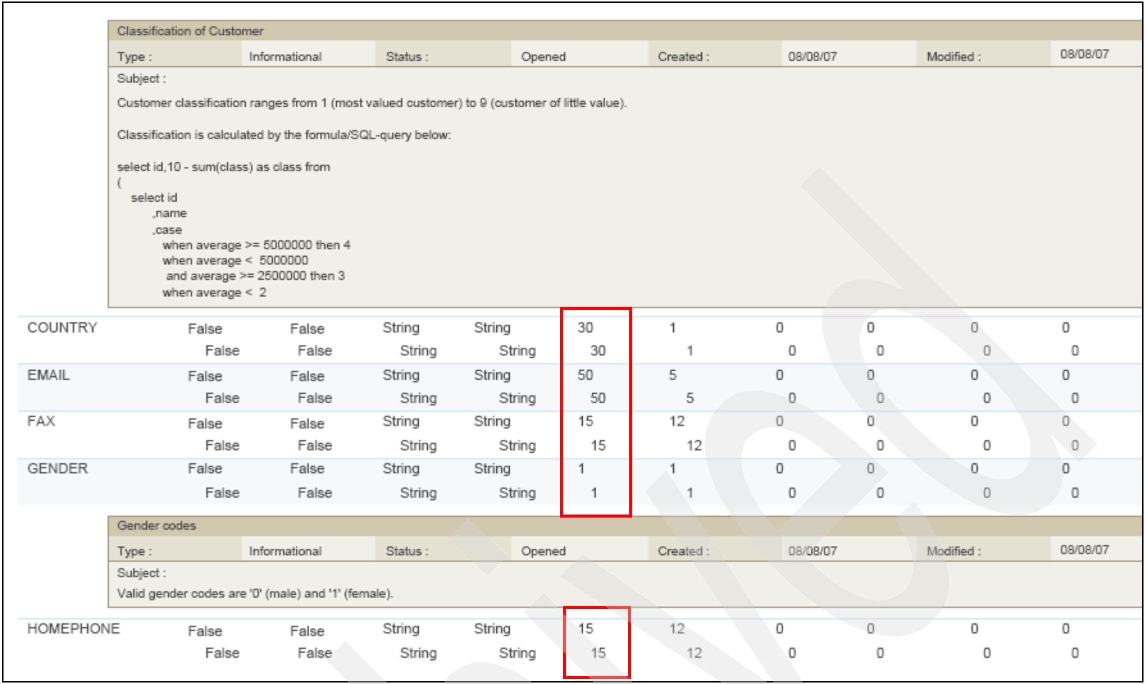
Date Of Analysis

Current :	08/17/07
Baseline :	08/17/07

Differences in Defined Data Properties

Column Name	Primary Key		Data Type		Length		Precision		Scale	
	Defined	Inferred	Defined	Inferred	Defined	Inferred	Defined	Inferred	Defined	Inferred
	Current	Current	Current	Current	Current	Current	Current	Current	Current	Current
	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline	Baseline
ADDR1	False	False	String	String	50	35	0	0	0	0
	False	False	String	String	50	35	0	0	0	0
ADDR2	False	False	String	String	50	50	0	0	0	0
	False	False	String	String	50	50	0	0	0	0
ADVISOR	False	False	Int32	Int32	4	8	4	8	0	0
	False	False	Int32	Int32	4	8	4	8	0	0
BRANCH	False	False	Int32	Int32	4	8	4	8	0	0
	FaFalse	TrFalse	Int32	Int32	4	8	4	8	0	0
BY	False	False	String	String	8	8	0	0	0	0
	False	False	String	String	8	8	0	0	0	0
CELLPHONE	False	False	String	String	15	12	0	0	0	0
	False	False	String	String	15	12	0	0	0	0
CITY	False	False	String	String	30	22	0	0	0	0
	False	False	String	String	30	22	0	0	0	0
CLASS	False	False	Int32	Int8	4	2	4	2	0	0
	False	False	Int32	Int8	4	1	4	1	0	0

Figure 2-32 Baseline Structure “Current to Prior Structural Variances” report for the CUSTOMER table 1/3



Classification of Customer

Type :	Informational	Status :	Opened	Created :	08/08/07	Modified :	08/08/07

Subject :

Customer classification ranges from 1 (most valued customer) to 9 (customer of little value).

Classification is calculated by the formula/SQL-query below:

```
select id,10 - sum(class) as class from
(
   select id
        ,name
        ,case
          when average >= 5000000 then 4
          when average <  5000000
           and average >= 2500000 then 3
          when average <  2
```

COUNTRY	False	False	String	String	30	1	0	0	0	0
	False	False	String	String	30	1	0	0	0	0
EMAIL	False	False	String	String	50	5	0	0	0	0
	False	False	String	String	50	5	0	0	0	0
FAX	False	False	String	String	15	12	0	0	0	0
	False	False	String	String	15	12	0	0	0	0
GENDER	False	False	String	String	1	1	0	0	0	0
	False	False	String	String	1	1	0	0	0	0

Gender codes

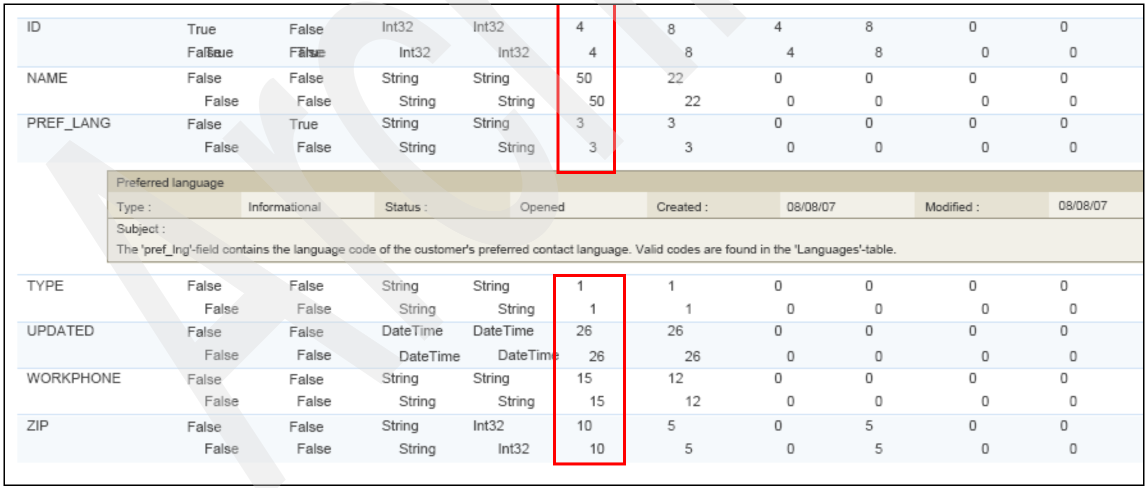
Type :	Informational	Status :	Opened	Created :	08/08/07	Modified :	08/08/07

Subject :

Valid gender codes are '0' (male) and '1' (female).

HOMEPHONE	False	False	String	String	15	12	0	0	0	0
	False	False	String	String	15	12	0	0	0	0

Figure 2-33 Baseline Structure "Current to Prior Structural Variances" report for the CUSTOMER table 2/3



ID	True	False	Int32	Int32	4	8	4	8	0	0
	False True	False True	Int32	Int32	4	8	4	8	0	0
NAME	False	False	String	String	50	22	0	0	0	0
	False	False	String	String	50	22	0	0	0	0
PREF_LANG	False	True	String	String	3	3	0	0	0	0
	False	False	String	String	3	3	0	0	0	0

Preferred language

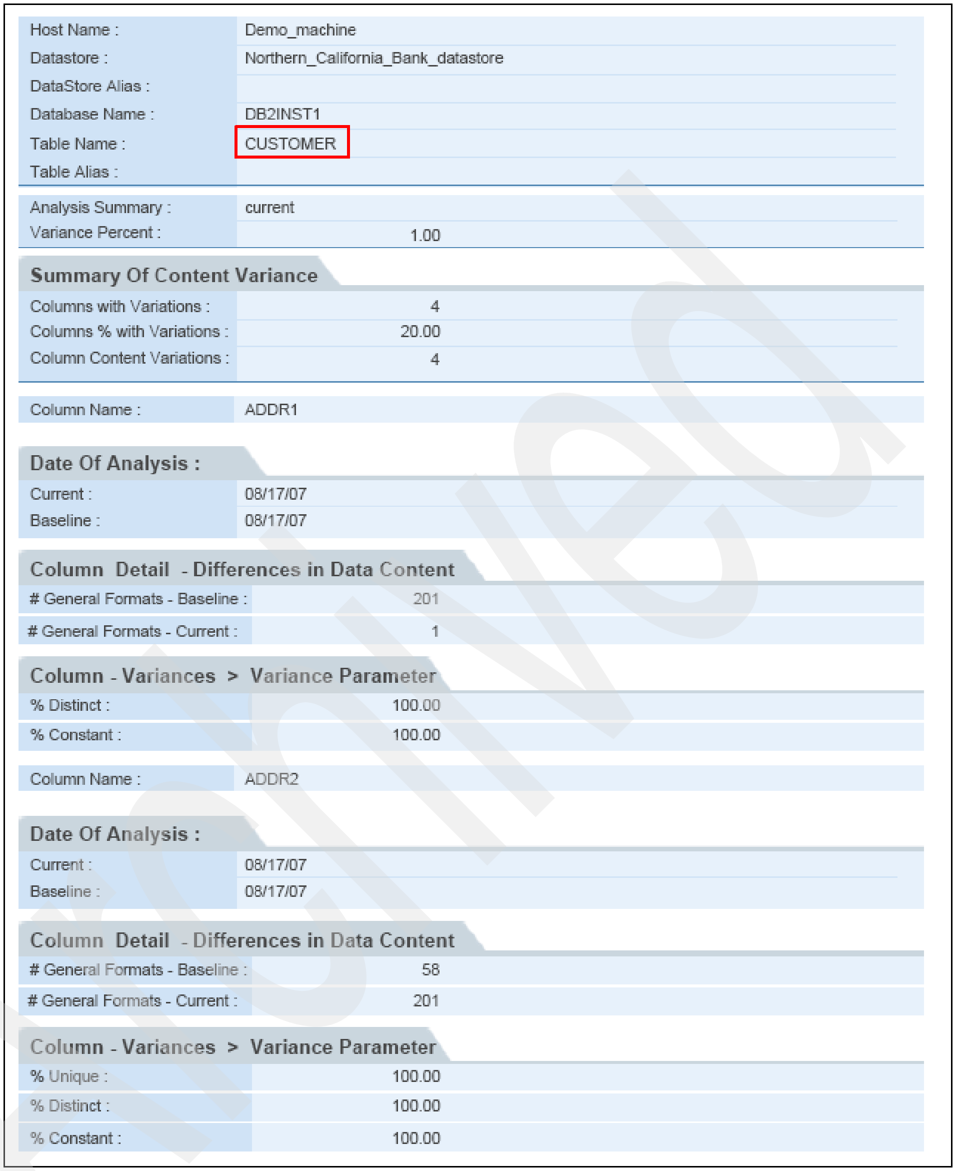
Type :	Informational	Status :	Opened	Created :	08/08/07	Modified :	08/08/07

Subject :

The 'pref_lng'-field contains the language code of the customer's preferred contact language. Valid codes are found in the 'Languages'-table.

TYPE	False	False	String	String	1	1	0	0	0	0
	False	False	String	String	1	1	0	0	0	0
UPDATED	False	False	DateTime	DateTime	26	26	0	0	0	0
	False	False	DateTime	DateTime	26	26	0	0	0	0
WORKPHONE	False	False	String	String	15	12	0	0	0	0
	False	False	String	String	15	12	0	0	0	0
ZIP	False	False	String	Int32	10	5	0	5	0	0
	False	False	String	Int32	10	5	0	5	0	0

Figure 2-34 Baseline Structure "Current to Prior Structural Variances" report for the CUSTOMER table 3/3



Host Name :	Demo_machine
Datastore :	Northern_California_Bank_datastore
DataStore Alias :	
Database Name :	DB2INST1
Table Name :	CUSTOMER
Table Alias :	

Analysis Summary :	current
Variance Percent :	1.00

Summary Of Content Variance

Columns with Variations :	4
Columns % with Variations :	20.00
Column Content Variations :	4

Column Name :	ADDR1

Date Of Analysis :

Current :	08/17/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	201
# General Formats - Current :	1

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00

Column Name :	ADDR2

Date Of Analysis :

Current :	08/17/07
Baseline :	08/17/07

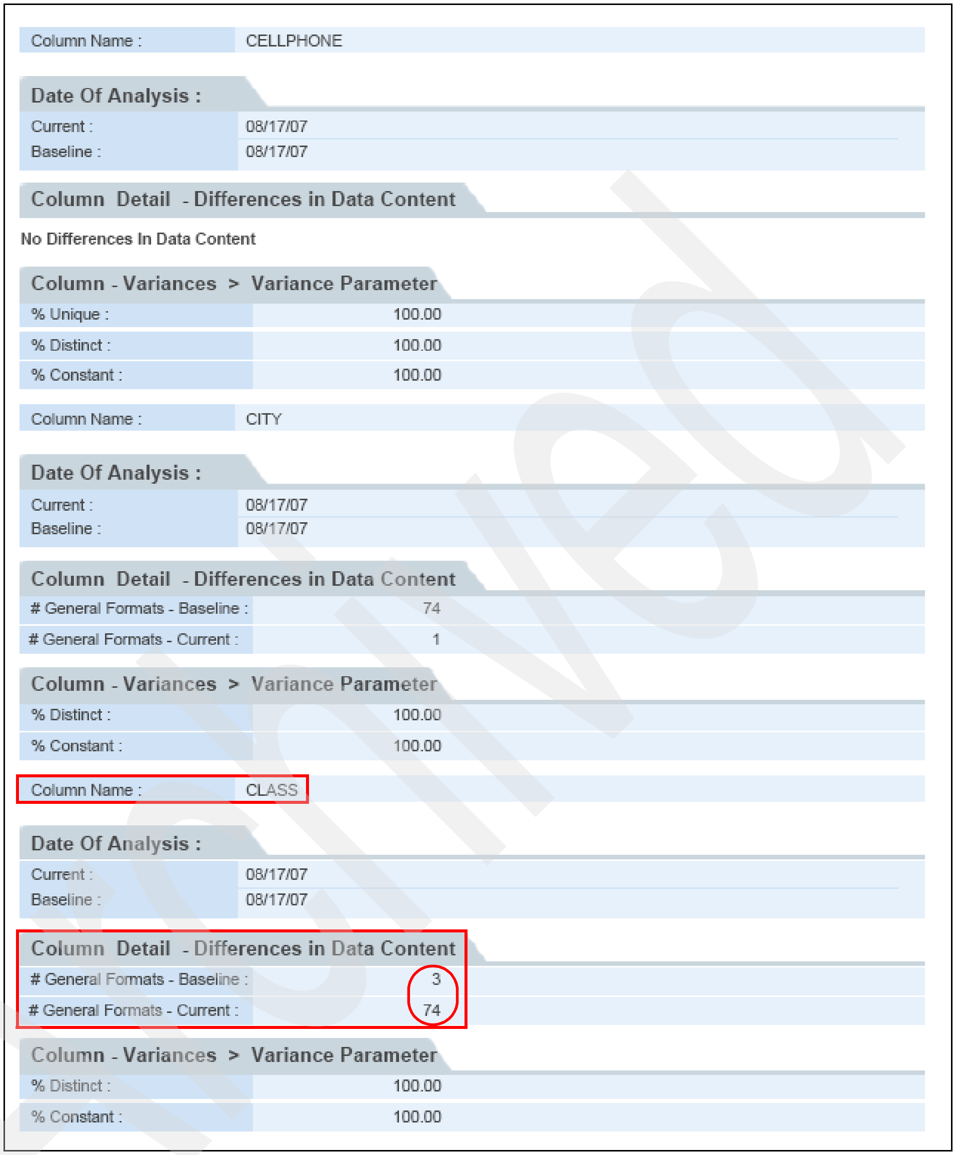
Column Detail - Differences in Data Content

# General Formats - Baseline :	58
# General Formats - Current :	201

Column - Variances > Variance Parameter

% Unique :	100.00
% Distinct :	100.00
% Constant :	100.00

Figure 2-35 Baseline Structure "Current to Prior Variances" report for the CUSTOMER table 1/2



Column Name :	CELLPHONE

Date Of Analysis :

Current :	08/17/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

No Differences In Data Content

Column - Variances > Variance Parameter

% Unique :	100.00
% Distinct :	100.00
% Constant :	100.00

Column Name :	CITY

Date Of Analysis :

Current :	08/17/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	74
# General Formats - Current :	1

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00

Column Name :	CLASS

Date Of Analysis :

Current :	08/17/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	3
# General Formats - Current :	74

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00

Figure 2-36 Baseline Structure "Current to Prior Variances" report for the CUSTOMER table 2/2

IBM WebSphere AuditStage business rule compliance

As mentioned earlier, Business Rule Compliance evaluates the quality of data in terms of specific business rules involving multiple data fields within or across records (or rows) that are logically related. IBM WebSphere AuditStage is used to perform this check. In this section, we do not describe the process of building the data filter because that is described in 1.12, “IBM WebSphere AuditStage business rule validation” on page 336 but focus on the completed data filter specification and the results of executing the business rule.

The business rules described in Table 2-9 on page 452 are verified using the data filters developed using IBM WebSphere AuditStage.

1: Check ADVISOR and USERID of employee matches up

This business rule specifies that the customer data in both the CUSTOMER and BCUSTOMER (part of the non-core services) tables can only be updated by the employee who is the advisor to that customer. The ADVISOR column in the CUSTOMER and BCUSTOMER tables identify their advisor employee in the EMPLOYEE table. The ADVISOR column might be null. The BY column in the CUSTOMER and BCUSTOMER tables should correspond to the relevant USERID column in the EMPLOYEE table. Figure 2-37 on page 468 through Figure 2-45 on page 472 show the business rule compliance execution.

Multiple data filters are defined with the exceptions generated from each data filter execution written to an exception table BR_CUSTOMER. Each data filter’s execution exceptions are appended to this table. The contents of this exception table are then listed as shown in Figure 2-45 on page 472.

1. Figure 2-37 on page 468 shows the data filter that checks whether the BY column (which contains the user ID performing the update) in the BCUSTOMER table has a value when the ADVISOR column in the same table is empty. It shows that the exceptions should be written to an exception table BR_CUSTOMER. Click **Run** to execute the data filter.

   Figure 2-38 on page 468 shows 5402 exceptions being generated that are written to the exception table BR_CUSTOMER.

2. Figure 2-39 on page 469 shows the data filter that checks whether the BY column (which contains the user ID performing the update) in the CUSTOMER table has a value when the ADVISOR column in the same table is empty. It shows that the exceptions should be appended to the exception table BR_CUSTOMER. Click **Run** to execute the data filter.

   Figure 2-40 on page 469 shows 4919 exceptions being generated that are appended to the exception table BR_CUSTOMER.

3. Figure 2-41 on page 470 shows whether the BY column (which contains the user ID performing the update) in the BCUSTOMER table has a value that does not match the USERID column in the EMPLOYEE table for a given customer in the BCUSTOMER table that has a non-null ADVISOR column. It shows that the exceptions should be appended to the exception table BR_CUSTOMER. Click **Run** to execute the data filter.

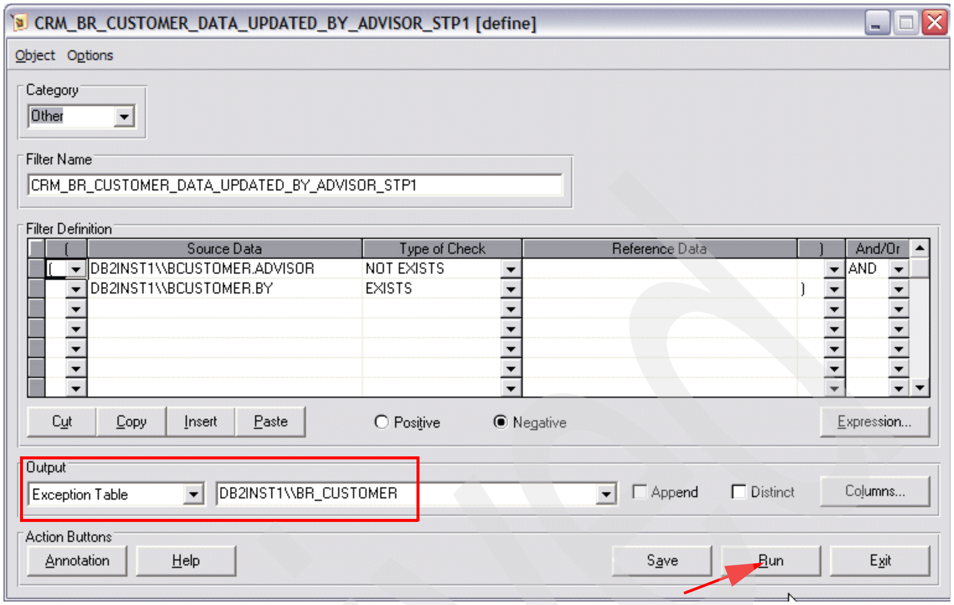
> **Note:** The EMPLOYEE source has the NOT BASE Type of Check. The BASE clause is not an actual Data Filter type but a dummy that tells IBM WebSphere AuditStage how to interpret a Data Filter. A BASE clause must contain a table name in the Source Data column. AND/OR operators have no effect on the execution, because a BASE clause is interpreted independently from the rest of the Data Filter.
>
> When positive, a BASE clause simply represents an entire table. It is used to force IBM WebSphere AuditStage to select data from the table when it cannot be inferred otherwise. It can also be used as a universal quantifier to select all data from a table with no conditions placed on it.
>
> The most common use for a BASE clause is in the negative syntax (NOT BASE as shown in Figure 2-41 on page 470), where it excludes the specified table from the calculation of statistical results shown in the Results dialog box. The BASE clause affects the statistics only and not the action of the associated Data Filter. As an example, consider a Data Filter that joins table A with table B. Normally, the total rows reported would be the number of rows in A times the number of rows in B. However, if you only want to report exceptions from table A against the total number of rows in table A, you can exclude table B from the base.

Figure 2-42 on page 470 shows zero exceptions being generated.

4. Figure 2-43 on page 471 shows whether the BY column (which contains the user ID performing the update) in the CUSTOMER table has a value that does not match the USERID column in the EMPLOYEE table for a given customer in the CUSTOMER table that has a non-null ADVISOR column. It shows that the exceptions should be appended to the exception table BR_CUSTOMER. Click **Run** to execute the data filter.

   Figure 2-44 on page 471 shows zero exceptions being generated.

5. Browse the contents of the BR_CUSTOMER exception table with 10321 rows as shown in Figure 2-45 on page 472 following the steps described in Figure 1-292 on page 362 and Figure 1-293 on page 363.


CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1 [define]

Category: Other

Filter Name: CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1

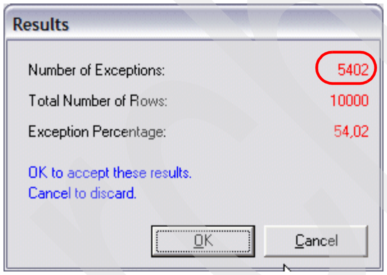
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
(	DB2INST1\\BCUSTOMER.ADVISOR	NOT EXISTS			AND
	DB2INST1\\BCUSTOMER.BY	EXISTS		)	

○ Positive ● Negative

Output: Exception Table DB2INST1\\BR_CUSTOMER □ Append □ Distinct

Figure 2-37 Check ADVISOR and USERID of employee matches up 1/9


Results

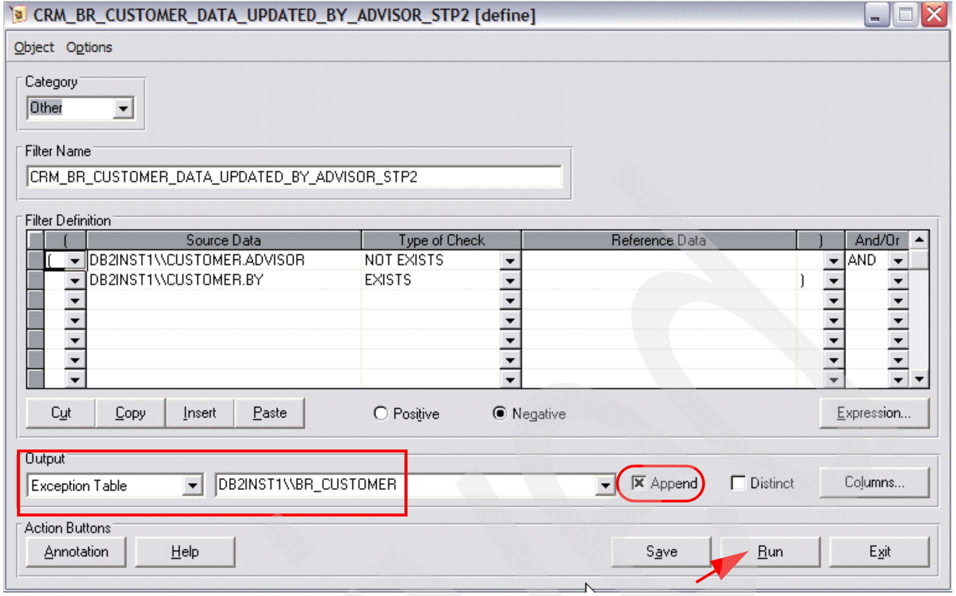
Number of Exceptions: 5402

Total Number of Rows: 10000

Exception Percentage: 54,02

OK to accept these results.
Cancel to discard.

Figure 2-38 Check ADVISOR and USERID of employee matches up 2/9



CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP2 [define]

Category: Other

Filter Name: CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP2

Filter Definition

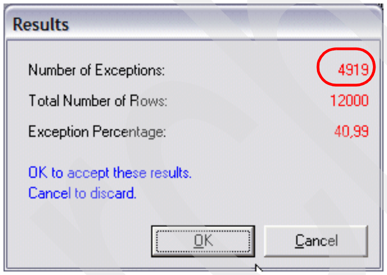
(	Source Data	Type of Check	Reference Data	)	And/Or
(	DB2INST1\\CUSTOMER.ADVISOR	NOT EXISTS			AND
	DB2INST1\\CUSTOMER.BY	EXISTS		)	

○ Positive ● Negative

Output: Exception Table DB2INST1\\BR_CUSTOMER ☒ Append □ Distinct

Action Buttons

Figure 2-39 Check ADVISOR and USERID of employee matches up 3/9



Results

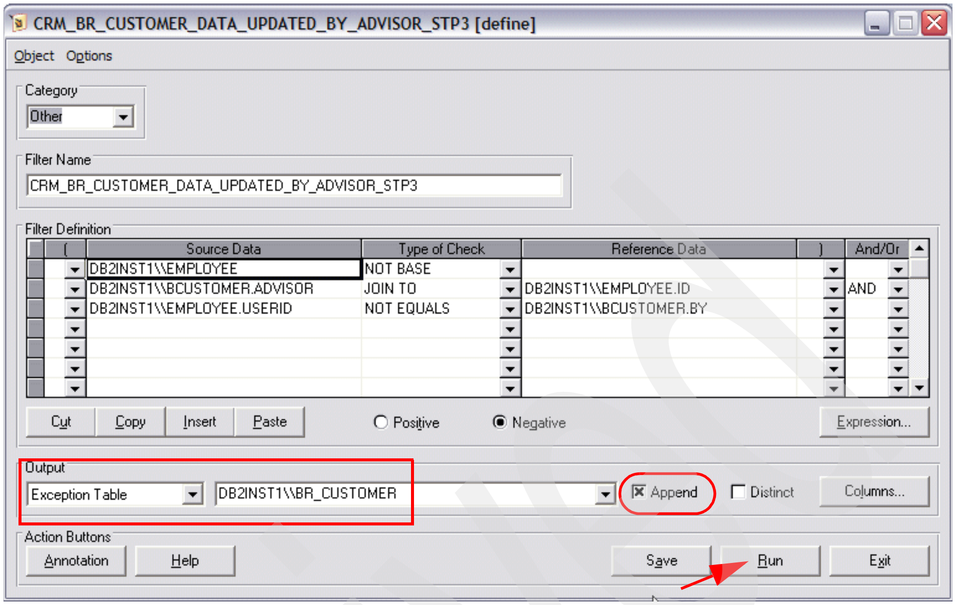
Number of Exceptions: 4919

Total Number of Rows: 12000

Exception Percentage: 40,99

OK to accept these results.
Cancel to discard.

Figure 2-40 Check ADVISOR and USERID of employee matches up 4/9



CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP3 [define]

Category: Other

Filter Name: CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP3

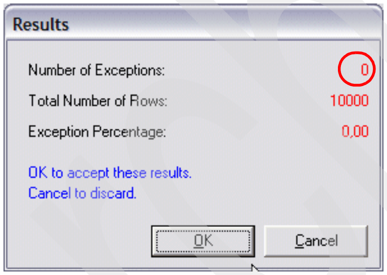
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	DB2INST1\\EMPLOYEE	NOT BASE			
	DB2INST1\\BCUSTOMER.ADVISOR	JOIN TO	DB2INST1\\EMPLOYEE.ID		AND
	DB2INST1\\EMPLOYEE.USERID	NOT EQUALS	DB2INST1\\BCUSTOMER.BY		

○ Positive ● Negative

Output: Exception Table DB2INST1\\BR_CUSTOMER ☒ Append □ Distinct

Figure 2-41 Check ADVISOR and USERID of employee matches up 5/9

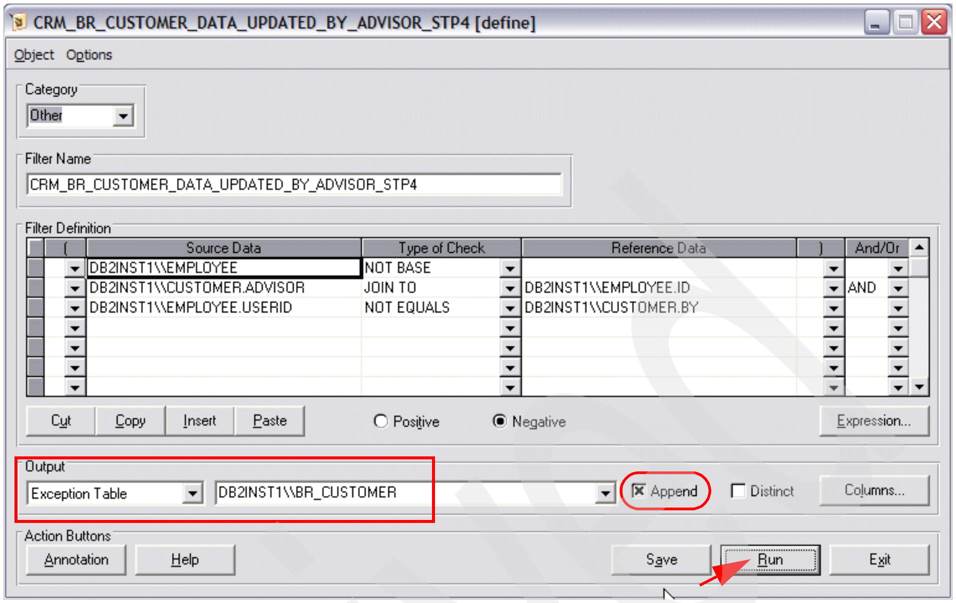


Results

Number of Exceptions:	0
Total Number of Rows:	10000
Exception Percentage:	0,00

OK to accept these results.
Cancel to discard.

Figure 2-42 Check ADVISOR and USERID of employee matches up 6/9



CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP4 [define]

Category: Other

Filter Name: CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP4

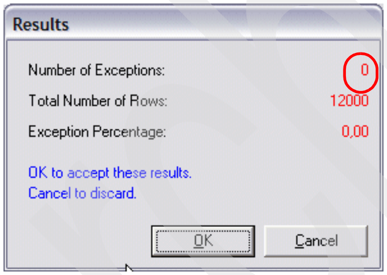
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	DB2INST1\\EMPLOYEE	NOT BASE			
	DB2INST1\\CUSTOMER.ADVISOR	JOIN TO	DB2INST1\\EMPLOYEE.ID		AND
	DB2INST1\\EMPLOYEE.USERID	NOT EQUALS	DB2INST1\\CUSTOMER.BY		

○ Positive ● Negative

Output: Exception Table DB2INST1\\BR_CUSTOMER ☒ Append ☐ Distinct

Figure 2-43 Check ADVISOR and USERID of employee matches up 7/9



Results

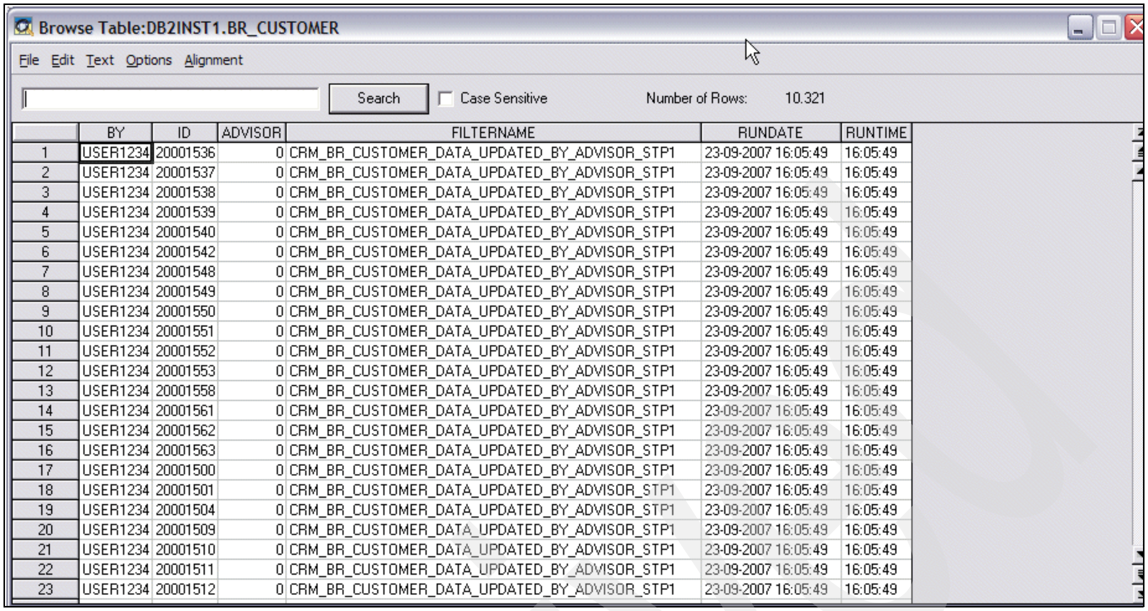
Number of Exceptions: 0

Total Number of Rows: 12000

Exception Percentage: 0,00

OK to accept these results.
Cancel to discard.

Figure 2-44 Check ADVISOR and USERID of employee matches up 8/9



Browse Table:DB2INST1.BR_CUSTOMER
File Edit Text Options Alignment
Search
□ Case Sensitive

Number of Rows: 10.321

	BY	ID	ADVISOR	FILTERNAME	RUNDATE	RUNTIME
1	USER1234	20001536	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
2	USER1234	20001537	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
3	USER1234	20001538	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
4	USER1234	20001539	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
5	USER1234	20001540	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
6	USER1234	20001542	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
7	USER1234	20001548	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
8	USER1234	20001549	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
9	USER1234	20001550	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
10	USER1234	20001551	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
11	USER1234	20001552	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
12	USER1234	20001553	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
13	USER1234	20001558	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
14	USER1234	20001561	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
15	USER1234	20001562	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
16	USER1234	20001563	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
17	USER1234	20001500	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
18	USER1234	20001501	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
19	USER1234	20001504	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
20	USER1234	20001509	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
21	USER1234	20001510	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
22	USER1234	20001511	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49
23	USER1234	20001512	0	CRM_BR_CUSTOMER_DATA_UPDATED_BY_ADVISOR_STP1	23-09-2007 16:05:49	16:05:49

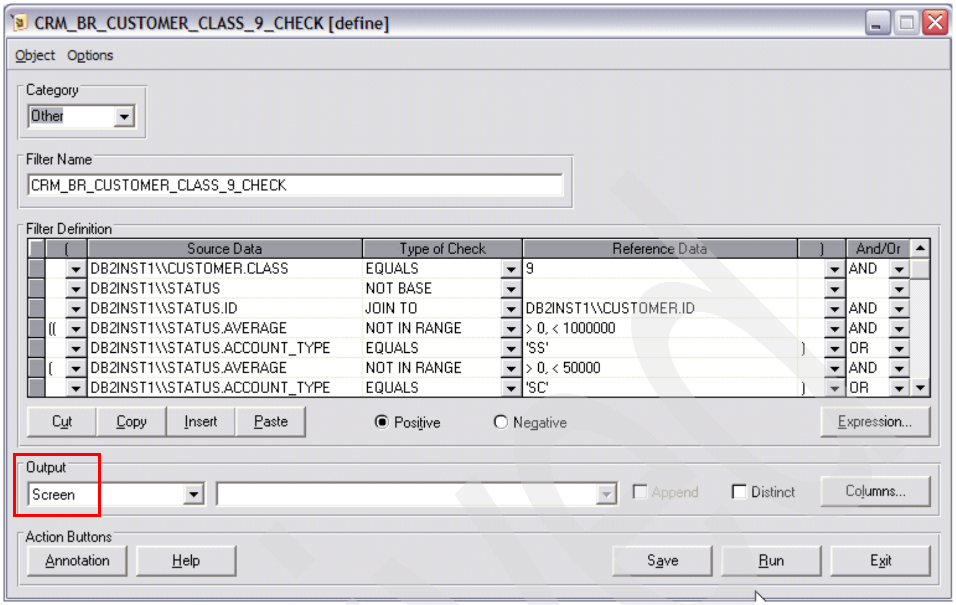
Figure 2-45 Check ADVISOR and USERID of employee matches up 9/9

2: Check CLASS column is computed correctly

Figure 2-46 on page 473 shows the data filter for verifying the business rule that the CLASS column in the CUSTOMER table is computed correctly for CLASS value of 9. After building the data filter with the output being sent back to the screen, click **Run** to execute the data filter.

> **Note:** You can verify other values for the CLASS column (1 through 8) in a similar fashion.

Figure 2-47 on page 473 shows the number of exceptions. 39 rows qualify as shown in Figure 2-48 on page 474.



CRM_BR_CUSTOMER_CLASS_9_CHECK [define]

Category: Other

Filter Name: CRM_BR_CUSTOMER_CLASS_9_CHECK

Filter Definition

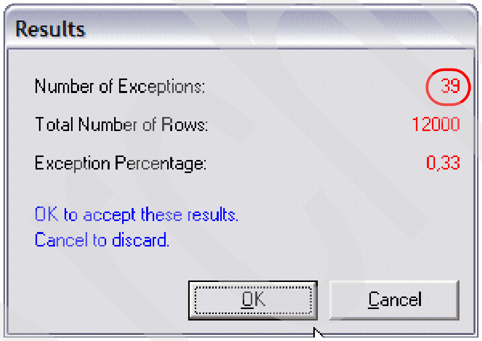
(	Source Data	Type of Check	Reference Data	)	And/Or
	DB2INST1\\CUSTOMER.CLASS	EQUALS	9		AND
	DB2INST1\\STATUS	NOT BASE			
	DB2INST1\\STATUS.ID	JOIN TO	DB2INST1\\CUSTOMER.ID		AND
((	DB2INST1\\STATUS.AVERAGE	NOT IN RANGE	> 0, < 1000000		AND
	DB2INST1\\STATUS.ACCOUNT_TYPE	EQUALS	'SS'	)	OR
(	DB2INST1\\STATUS.AVERAGE	NOT IN RANGE	> 0, < 50000		AND
	DB2INST1\\STATUS.ACCOUNT_TYPE	EQUALS	'SC'	)	OR

● Positive ○ Negative

Output: Screen

□ Append □ Distinct

Figure 2-46 Check CLASS column is computed correctly 1/3



Results

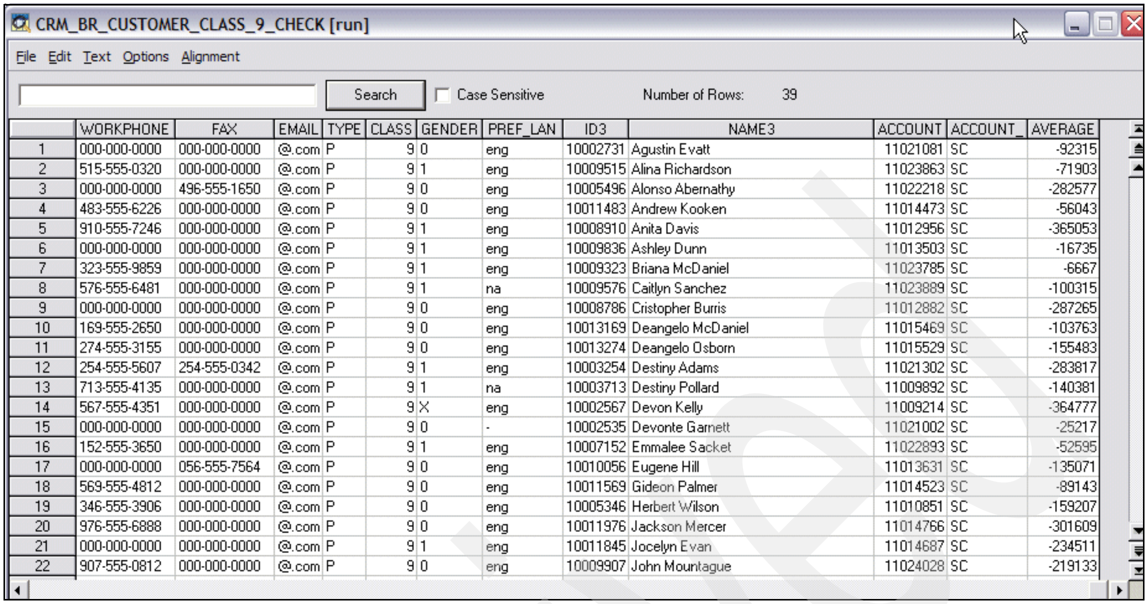
Number of Exceptions: 39

Total Number of Rows: 12000

Exception Percentage: 0,33

OK to accept these results.
Cancel to discard.

Figure 2-47 Check CLASS column is computed correctly 2/3



CRM_BR_CUSTOMER_CLASS_9_CHECK [run]

Number of Rows: 39

	WORKPHONE	FAX	EMAIL	TYPE	CLASS	GENDER	PREF_LAN	ID3	NAME3	ACCOUNT	ACCOUNT_	AVERAGE
1	000-000-0000	000-000-0000	@.com	P	9	0	eng	10002731	Agustin Evatt	11021081	SC	-92315
2	515-555-0320	000-000-0000	@.com	P	9	1	eng	10009515	Alina Richardson	11023863	SC	-71903
3	000-000-0000	496-555-1650	@.com	P	9	0	eng	10005496	Alonso Abernathy	11022218	SC	-282577
4	483-555-6226	000-000-0000	@.com	P	9	0	eng	10011483	Andrew Kooken	11014473	SC	-56043
5	910-555-7246	000-000-0000	@.com	P	9	1	eng	10008910	Anita Davis	11012956	SC	-365053
6	000-000-0000	000-000-0000	@.com	P	9	1	eng	10009836	Ashley Dunn	11013503	SC	-16735
7	323-555-9859	000-000-0000	@.com	P	9	1	eng	10009323	Briana McDaniel	11023785	SC	-6667
8	576-555-6481	000-000-0000	@.com	P	9	1	na	10009576	Caitlyn Sanchez	11023889	SC	-100315
9	000-000-0000	000-000-0000	@.com	P	9	0	eng	10008786	Cristopher Burris	11012882	SC	-287265
10	169-555-2650	000-000-0000	@.com	P	9	0	eng	10013169	Deangelo McDaniel	11015469	SC	-103763
11	274-555-3155	000-000-0000	@.com	P	9	0	eng	10013274	Deangelo Osborn	11015529	SC	-155483
12	254-555-5607	254-555-0342	@.com	P	9	1	eng	10003254	Destiny Adams	11021302	SC	-283817
13	713-555-4135	000-000-0000	@.com	P	9	1	na	10003713	Destiny Pollard	11009892	SC	-140381
14	567-555-4351	000-000-0000	@.com	P	9	X	eng	10002567	Devon Kelly	11009214	SC	-364777
15	000-000-0000	000-000-0000	@.com	P	9	0	-	10002535	Devonte Garnett	11021002	SC	-25217
16	152-555-3650	000-000-0000	@.com	P	9	1	eng	10007152	Emmalee Sacket	11022893	SC	-52595
17	000-000-0000	056-555-7564	@.com	P	9	0	eng	10010056	Eugene Hill	11013631	SC	-135071
18	569-555-4812	000-000-0000	@.com	P	9	0	eng	10011569	Gideon Palmer	11014523	SC	-89143
19	346-555-3906	000-000-0000	@.com	P	9	0	eng	10005346	Herbert Wilson	11010851	SC	-159207
20	976-555-6888	000-000-0000	@.com	P	9	0	eng	10011976	Jackson Mercer	11014766	SC	-301609
21	000-000-0000	000-000-0000	@.com	P	9	1	eng	10011845	Jocelyn Evan	11014687	SC	-234511
22	907-555-0812	000-000-0000	@.com	P	9	0	eng	10009907	John Mountague	11024028	SC	-219133

Figure 2-48 Check CLASS column is computed correctly 3/3

2.5.6 Migration Analysis

Figure 2-5 on page 425 describes the general approach for migration of the core services of North American Bank's systems.

Details of the steps as they relate to migration are described here.

- Step 1: General guidelines for the process are described in 2.5.1, "Assumptions about the migration" on page 431.
- Step 2: Identify differences between the source(s) & target(s) is described here.

  As indicated earlier, the data elements of interest are corresponding keys, codes, and indicators in the source and target systems.

  A manual comparison of the source (described in 2.5.4, "North American Bank analysis" on page 435) and target (described in 2.5.5, "Northern California Bank analysis" on page 451) analyses indicates the differences shown in Table 2-13 on page 475 and Table 2-14 on page 478. The action to take to manage these differences are also indicated in Table 2-13 on page 475 and Table 2-14 on page 478. This comparative process could be done in a spreadsheet such as Excel®, annotated or noted in IBM WebSphere Information Analyzer for Analysis Publication, or might use products that specify and map sources to targets. Cross domain analysis (which we did not use in this scenario) could be used to facilitate some of this comparison.

> **Note:** A business decision was made to not migrate the human resources system (EMPLOYEE table in our data model) and not profile in any detail the name and address fields in the North American Bank and Northern California Bank systems.

Table 2-13 Summary of differences between source and target core services and the action to be taken

North American Bank (source)				Northern California Bank (target)			Action to be taken
Column in the table	**Metadata**		**Data Content example**	**Column in the table**	**Metadata Defined**	**Data Content example**	
	Defined	**Inferred**					
CUSTOMER_ID in CUSTOMER	INT32	INT16	4079	ID in CUSTOMER	INT32	10003828	Generate new keys for the records in the source in the target. If required, create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
ACCOUNT_ID in ACCOUNT	INT32	INT16	216	ID in ACCOUNT	INT32	11001500	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
ACCOUNT_ID,LOAN_ID in LOAN	(INT32, INT32)	(INT16, INT16)	(3295,2197)	LOAN_ID in LOAN	INT32	11001500	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved

North American Bank (source)				Northern California Bank (target)			Action to be taken
ACCOUNT_ID,LOAN_ID ,TRANSACTION_ID in LOAN_TRANSACTION	(INT32,INT32, INT32)	(INT16, INT16, INT32)	(152,102,13767)	ACCOUNT, UPDATED in TRANSACTION	(INTEGER, TIMESTAMP)	(11001583, 2005-11-19 11:28:29.745877)	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
ACCOUNT_ID, TRANSACTION_ID in TRANSACTION	(INT32,INT32)	(INT16,INT32)	(100,27397)	ACCOUNT, UPDATED in TRANSACTION	(INT32, TIMESTAMP)	(11001583, 2005-11-19 11:28:29.745877)	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
BRANCH_ID in BRANCH	INT32	INT8	51	ID in BRANCH	INT32	12001536	Generate new keys for the records in the source in the target. If necessary, create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
GENDER_IND in CUSTOMER	CHAR(1)	CHAR(1)	F	GENDER in CUSTOMER	CHAR(1)	1	Transform the values from the source to the target.
LEVEL_CD in CUSTOMER	CHAR(2)	CHAR(2)	SL	CLASS in CUSTOMER	INT32	7	Map with data type transformation
TITLE in CUSTOMER	CHAR(3)	CHAR(2)	MS	NAME in CUSTOMER	CHAR(50)	Jazmine Fisher	Map the TITLE, FIRST_NAME and LAST_NAME columns in the source into the NAME column in the target
FIRST_NAME in CUSTOMER	VARCHAR(20)	CHAR(12)	SHAYLA	NAME in CUSTOMER	CHAR(50)	Jazmine Fisher	
LAST_NAME in CUSTOMER	VARCHAR(20)	CHAR(16)	VAN DER ZIJDEN	NAME in CUSTOMER	CHAR(50)	Jazmine Fisher	
HOME_ADDRESS in CONTACT_INFO	VARCHAR(50)	VARCHAR(36)	1301 Evans Avenue, San Francisco, CA	ADDR1 in CUSTOMER	VARCHAR(50)	397 CHISANA STREET	Direct mapping
HOME_ZIP in CONTACT_INFO	CHAR(9)	INT32 length 5	94124	ZIP in CUSTOMER	CHAR(10)	90028	Transform by including a hyphen
HOME_PHONE in CONTACT_INFO	CHAR(15)	INT64 length 10	8005553717	HOMEPHONE in CUSTOMER	CHAR(15)	517-555-1385	Map and include hyphen

North American Bank (source)				Northern California Bank (target)			Action to be taken
WORK_PHONE in CONTACT_INFO	CHAR(15)	CHAR(10)	8145553731,NA	WORKPHONE in CUSTOMER	CHAR(15)	000-000-0000	Map and include hyphen
CELL_PHONE in CONTACT_INFO	CHAR(15)	CHAR(10)	8175553734	CELLPHONE in CUSTOMER	CHAR(15)	517-555-4641	Map and include hyphen
TYPE_IND in ACCOUNT	CHAR(1)	CHAR(1)	C	TYPE in ACCOUNT	CHAR(2)	SS	Transform the values from the source to the target.
RATES in LOAN	DECIMAL(8,5)	DECIMAL (7,5)	2.15	INTEREST_RATE in LOAN	CHAR(20)	19.75	Map with data type transformation
INITIAL_VALUE in LOAN	DECIMAL(9,2)	DECIMAL(8,2)	1000	INITIAL_LOAN_VALUE in LOAN	CHAR(20)	100000	Map with data type transformation
LATE_FEE in LOAN	DECIMAL(9,2)	DECIMAL(6,2)	100	LATE_FEE in LOAN	CHAR(20)	50	Map with data type transformation
LATE_RATE in LOAN	DECIMAL(8,5)	DECIMAL(7,5)	5.123	LATE_INTEREST_RATE in LOAN	CHAR(20)	10	Map with data type transformation
BALANCE in LOAN	DECIMAL(9,2)	DECIMAL(8,2)	500	BALANCE in LOAN	CHAR(20)	75000	Map with data type transformation
TRANS_TYPE_CD in LOAN_TRANSACTION	CHAR(2)	CHAR(2)	D	CODE in TRANSACTION	CHAR(1)	C	Transform the values from the source to the target.
AMOUNT in LOAN_TRANSACTION	DECIMAL(9,2)	DECIMAL(8,2)	400.00	BALANCE in TRANSACTION	CHAR(20)	300	Map with data type transformation
TRANS_TYPE_CD in TRANSACTION	CHAR(2)	CHAR(2)	D	CODE in TRANSACTION	CHAR(1)	C	Transform the values from the source to the target.
AMOUNT in TRANSACTION	DECIMAL(9,2)	DECIMAL(7,2)	400.00	BALANCE in TRANSACTION	CHAR(20)	300	Map with data type transformation
TRANS_TYPE_CD in TRANSACTION_TYPE_REF	CHAR(2)	CHAR(2)	D	CODE in TRANSACTION	CHAR(1)	C	Transform the values from the source to the target.
LEVEL_CD in LEVEL_REF	CHAR(2)	CHAR(2)	PL				Ignore this column because it is part of a reference table

Table 2-14 Missing information in source or target bank relating to core services and action to be taken

Data element	North American Bank (source)	Northern California Bank (target)	Action to be taken
Customer's preferred language for interaction	not defined	ISO code (PREF_LANG column in the CUSTOMER table defined as NOT NULL WITH DEFAULT 'ENG')	No action because default will be applied
Currency of deposits	not defined	ISO code (CURRENCY column in the ACCOUNT and PORTFOLIO tables defined as nullable; CURRENCY column in the ACCTYPE table defined as NOT NULL WITH DEFAULT ' ')	No action because null or default will be applied
Country	not defined	ISO code (COUNTRY column in the BRANCH table defined as nullable; CTRY2, CTRY3, and CTRYN columns in the COUNTRY lookup table defined as NOT NULL)	No action because null or default will be applied for the COUNTRY column in the BRANCH table. CTRY2, CTRY3, and CTRYN columns are in the COUNTRY lookup table and need no action
Nationality	Free text field (NATIONALITY column in the CUSTOMER table)	not defined	Ignore this field during migration
Nickname	Free text field (NICKNAME column in the CUSTOMER table)	not defined	Ignore this field during migration
Credit score	Free text field (CREDIT_SCORE column in the CUSTOMER table)	not defined	Ignore this field during migration
Churn indicator	CHAR(1) (CHURN_IND column in the CUSTOMER table)	not defined	Ignore this field during migration

Data element	North American Bank (source)	Northern California Bank (target)	Action to be taken
Automatic debit indicator	CHAR(1) (AUTOMAT_DEBIT_IND column in the LOAN table)	not defined	Ignore this field during migration
Work address	Free text field (WORK_ADDRESS column in the CONTACT_INFO table)	not defined	Ignore this field during migration
Work ZIP code	Free text field (WORK_ZIP column in the CONTACT_INFO table)	not defined	Ignore this field during migration
Account fee frequency	not defined	“Y” for yearly, “M” for monthly, and “C” per check (FEEFRQ column in the ACCTYPE table defined as nullable)	No action
City	not defined	Separate field (CITY column in the CUSTOMER table defined as NOT NULL)	CITY column in the CUSTOMER table will have to be populated by extracting this information from the HOME_ADDRESS column in the CONTACT_INFO table in the North American Bank core services
Customer type	not defined	“P” for person, “O” for organization (TYPE column in the CUSTOMER table defined as NOT NULL WITH DEFAULT ‘-’)	No action because default will be applied
Employee information	Defined	Defined	Not going to migrate the HR system.

Data element	North American Bank (source)	Northern California Bank (target)	Action to be taken
Various amounts	► DECIMAL(9,2) (BALANCE in ACCOUNT) ► DECIMAL(9,2) (MIN_AMOUNT in ACCOUNT) ► DECIMAL(9,2) (OVERDRAFT in ACCOUNT) ► DECIMAL(9,2) (OVERDRAFT_LIMIT in ACCOUNT) ► DECIMAL(8,5) (OVERDRAFT_RATE in ACCOUNT) ► DECIMAL(9,2) (OVERDRAFT_FEE in ACCOUNT)		Modify target system with the correct data type, precision and scale. Significant impact.

- Step 3: Determine action in specific cases

  The general guideline of the target system being the over riding authority is considered inappropriate in the following cases because it would seriously jeopardize customer service satisfaction of North American Bank's patrons:

  – Add home and work numbers to the Northern California Bank system to continue to provide superior service for the North American Bank customer base.

  – Add credit score information to the Northern California Bank system for faster loan approvals to continue to provide superior customer service for the North American Bank customer base.

- Step 4: Determine strategy and plan to execute action

  As mentioned earlier, the strategy and plan should include the following:

  – Addressing the data integrity violations (primary key in Table 2-6 on page 442, foreign key in Table 2-7 on page 443, domains in Table 2-5 on page 439, and business rule compliance in Table 2-18 on page 448, Table 2-21 on page 449 and Table 2-24 on page 451) detected in North American Bank's core services systems' prior to initiating the migration, or alternatively during the migration.

  - Design cleansing, extract, transform, and load procedures to migrate the data. This largely involves the use of tools such as IBM WebSphere QualityStage and IBM WebSphere DataStage but can also involve user-written code.

  This plan needs to be rigorously tested using representative data used in the data profiling analyses.

  Note: IBM WebSphere Information Analyzer should be used to profile the representative data and to ensure its usefulness in handling data quality issues and for establishing a a source baseline to evaluate outputs. IBM WebSphere Information Analyzer should also be used to test the development outputs and the quality assurance results. This can facilitate a rapid review that otherwise would require hand-coding and might miss key issues in data cleansing and transformation. Use of Cross-domain Analysis can evaluate the overlap of the source and target data that is being transformed. Use of Baseline Analysis can establish output measures to evaluate success of changes and updates to the cleansing and transformation processes.

  Designing this strategy and plan is beyond the scope of this book.

- Step 5: Execute the plan

  Prior to executing the strategy and plan designed in the earlier step, you need to ensure that no structural and data content changes have occurred to the data sources and targets in the interval because data profiling analyses was initiated as described in “Baseline Analysis reports” on page 443. This is achieved by executing Baseline Analysis on the data sources and targets.

  If Baseline Analysis identifies changes that affect the migration strategy and plan, then the strategy and plan should be revised and re-tested.

  Execute the plan to migrate the core services of the North American Bank.

  This topic is beyond the scope of this book.

- Step 6: Review success of the process

  You should verify that the migration was successful by performing data profiling analyses on the Northern California Bank core services systems’ after migration.

  This topic is beyond the scope of this book.

2.6 Data integration of North American Bank and Northern California Bank systems

As mentioned earlier, a business decision was made to integrate the core (savings, checking, and loans) and non-core services (credit card and auto insurance) of the North American Bank on the z/OS platform with the core (savings, checking, and loans) and non-core services (brokerage) of the Northern California Bank on the AIX platform into the CRM system.

Important: A number of assumptions are made about data integration in this scenario, some of which might not apply to your particular environment. What we hope to achieve in this scenario is to highlight IBM WebSphere Information Analyzer functionality (through its reports) that can be used to identify defined and inferred metadata differences within the same data source, validate the integrity of data within a data source (within a table and across tables), and understand the frequency distribution of data within specific data elements. IBM WebSphere AuditStage is used to ensure business rule compliance of data within a data source. Such functionality is not currently available in IBM WebSphere Information Analyzer but is expected to become available in future.

Attention: Because the focus of this book is IBM WebSphere Information Analyzer, we do *not* describe the procedures for unloading, cleansing, transforming, and loading of the source data into the target environment. Those tasks are the domain of IBM WebSphere QualityStage and IBM WebSphere DataStage and will be covered in upcoming IBM Redbooks publications.

In this section, we describe the following topics related to data integration:

- Assumptions about data integration
- IBM WebSphere Information Analyzer features used
- IBM WebSphere AuditStage features used
- North American Bank non-core services analysis
- Northern California Bank non-core services analysis
- Data integration analysis

2.6.1 Assumptions about data integration

Most of the assumptions made about the data integration are the same as those described for migration in 2.5.1, “Assumptions about the migration” on page 431 plus the following:

- The CRM system provides an integrated view of all the customers in the merged bank including details of the core services (such as checking, savings, and loans) and non-core services products (such as credit card or brokerage services) consumed. The objective of the CRM system is to upsell and cross sell the merged banks’ products and services to the customer base. The management of the products and services sold continues to be in the original source system. In other words, the systems representing the core and non-core services remain in their existing environment and are merely referenced and invoked from the CRM system.
- The CRM system does not contain any transactions from the core and non-core services of the North American Bank and the Northern California Bank.
- New data elements exist in the CRM data model that do not exist in the existing systems of the North American Bank and Northern California Bank.
- Surrogate keys are used in the CRM. The keys in the existing systems are mapped to the surrogate keys.
- The CRM data model can be modified based on IBM WebSphere Information Analyzer analyses of the existing systems. This implies no loss of granularity.

> **Important:** Unlike the migration scenario, the general guideline when mismatches are found in the data integration scenario is to modify the data model of the target CRM system to accommodate the mismatches in the existing systems.

2.6.2 IBM WebSphere Information Analyzer features used

The role of Information Analyzer in data integration for the CRM is similar to that described for migration in 2.5.2, “IBM WebSphere Information Analyzer features used” on page 434. In addition, it will help determine the column data type and precision and scale of the corresponding columns in the CRM data model based on the defined and inferred data type, precision, scale, and nullability data profiling analyses of the existing systems.

2.6.3 IBM WebSphere AuditStage features used

The role of IBM WebSphere AuditStage in data integration for the CRM is similar to that described for migration in 2.5.3, “IBM WebSphere AuditStage features used” on page 435.

2.6.4 North American Bank non-core services analysis

As mentioned earlier, North American Bank’s non-core services are to be integrated into the CRM system.

Using metadata information available in dictionaries, documentation and the relational catalogs, the keys, code fields, indicator fields, and referential integrity relationships (both implicit and explicit) to be profiled were identified as shown in Table 2-15 on page 484.

> **Note:** For convenience, we chose to include all the columns in all the tables in our data profiling effort. In the real world, however, to avoid report and information overload, you would apply the 80/20 rule and focus on only the critical master data as described in 1.2.1, “Data assessment approach” on page 5.

We include only selected portions of the generated reports (for some of the fields shown in Table 2-15) of the data profiling effort in this section. You can download the complete reports from the IBM Redbooks Web site:

ftp://www.redbooks.ibm.com/redbooks/SG247508/

> **Note:** We describe the process of generating reports in 1.14, “Reports” on page 394 and do not repeat that information here. We show only the relevant portions of the reports here.

Business rules involving data elements in one or more tables are shown in Table 2-16 on page 485. These are verified using IBM WebSphere AuditStage.

Table 2-15 Main keys, codes, and indicators in North American Bank’s credit card and auto insurance

Table	Key column(s)	Code column	Indicator column	RI relationships
CARD	CARD_ID, CARD_TYPE_CD, CUSTOMER_ID, ACCOUNT_ID	▸ LEVEL_CD ▸ REWARDS_CD	▸ FLAG_IND ▸ INTL_IND ▸ REWARDS_IND ▸ AUTOMAT_DEBIT_IND	▸ LEVEL_CD to the LEVEL_REF table ▸ REWARDS_CD to the REWARD_REF table
CARD_TYPE_REF	CARD_TYPE_CD			
REWARD_REF	REWARDS_CD			

Table	Key column(s)	Code column	Indicator column	RI relationships
CARD_TRANSACTION	CARD_ID, CARD_TYPE_CD, CUSTOMER_ID, ACCOUNT_ID, TRANSACTION_ID	TRANS_TYPE_CD	► INTL_IND ► CUST_REFUSAL_IND	(CARD_ID, CARD_TYPE_CD, CUSTOMER_ID, ACCOUNT_ID) to the CARD table
TRANSACTION_TYPE_REF	TRANS_TYPE_CD	TRANS_TYPE_CD		
CAR_INSURANCE	ACCOUNT_ID, INSURANCE_ID		► FULL_COVERAGE_IND ► AUTOMAT_DEBIT_IND	ACCOUNT_ID to the ACCOUNT table
DRIVER	ACCOUNT_ID, INSURANCE_ID, DRIVER_ID	► CORRECTIVE_LENSES_IND ► STATE	GENDER	(ACCOUNT_ID, INSURANCE_ID) to the CAR_INSURANCE table

Table 2-16 Business rules in North American Bank credit card/auto insurance

Serial number	Description
1	International transactions in the CARD_TRANSACTION table can only occur if the CARD table has the INTL_IND indicator is set to “Y”
2	Age of the driver must be greater than 18; compute from BIRTH_DT in the DRIVER table
3	PIN column in the CARD table must be all numeric
4	REWARDS_NUM and REWARDS_CD can only have values when REWARDS_IND is “Y” in the CARD table

Column Analysis reports

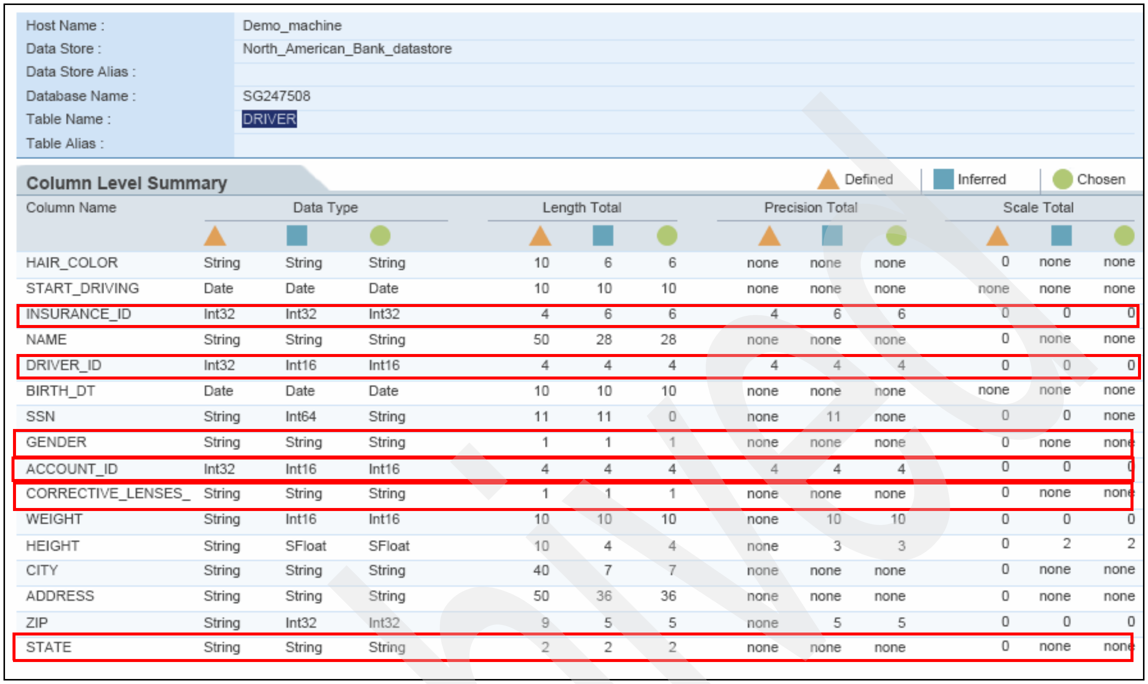
As mentioned earlier, these reports provide information about column values for completeness, validity, structure, and format as described in 1.7, “Column analysis” on page 109. These reports are used to review the columns for domain, structure and format integrity.

Figure 2-49 on page 486 shows the Column Inferences “Type & Length Properties (Defined/Inferred/Chosen)” report for the DRIVER table. It identifies the data type, length, precision and scale of each column as defined in the metadata and inferred from the data content. Any selected status of each of these categories is also shown here. This information (along with those relating to other columns in other tables) is recorded in Table 2-26 on page 514. Our focus is on keys, codes, and indicators as highlighted.

Figure 2-50 on page 487 shows the Column Frequency “Frequency By Frequency” report for the GENDER column in the DRIVER table. It shows valid values such as “M” and “F” with no invalid values.

Figure 2-51 on page 488 shows the Column Domain Values “Completeness and Validity Summary” report for the DRIVER table. It shows no occurrences of incomplete values or invalid values in the GENDER column.

Document all the valid values for a column in a table as shown in Table 2-17 on page 488.



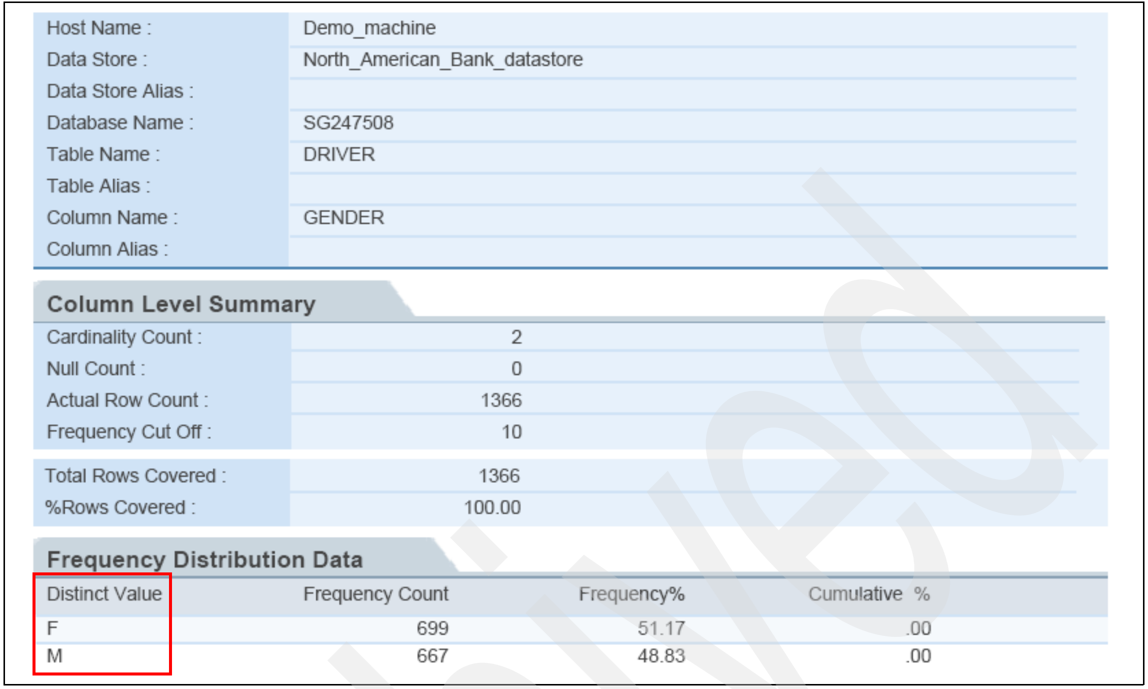
Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	DRIVER
Table Alias :	

Column Level Summary

▲ Defined ■ Inferred ● Chosen

Column Name	Data Type			Length Total			Precision Total			Scale Total		
	▲	■	●	▲	■	●	▲	■	●	▲	■	●
HAIR_COLOR	String	String	String	10	6	6	none	none	none	0	none	none
START_DRIVING	Date	Date	Date	10	10	10	none	none	none	none	none	none
INSURANCE_ID	Int32	Int32	Int32	4	6	6	4	6	6	0	0	0
NAME	String	String	String	50	28	28	none	none	none	0	none	none
DRIVER_ID	Int32	Int16	Int16	4	4	4	4	4	4	0	0	0
BIRTH_DT	Date	Date	Date	10	10	10	none	none	none	none	none	none
SSN	String	Int64	String	11	11	0	none	11	none	0	0	none
GENDER	String	String	String	1	1	1	none	none	none	0	none	none
ACCOUNT_ID	Int32	Int16	Int16	4	4	4	4	4	4	0	0	0
CORRECTIVE_LENSES_	String	String	String	1	1	1	none	none	none	0	none	none
WEIGHT	String	Int16	Int16	10	10	10	none	10	10	0	0	0
HEIGHT	String	SFloat	SFloat	10	4	4	none	3	3	0	2	2
CITY	String	String	String	40	7	7	none	none	none	0	none	none
ADDRESS	String	String	String	50	36	36	none	none	none	0	none	none
ZIP	String	Int32	Int32	9	5	5	none	5	5	0	0	0
STATE	String	String	String	2	2	2	none	none	none	0	none	none

Figure 2-49 Column Inferences “Type & Length Properties (Defined/Inferred/Chosen)” report for the DRIVER table



Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	DRIVER
Table Alias :	
Column Name :	GENDER
Column Alias :	

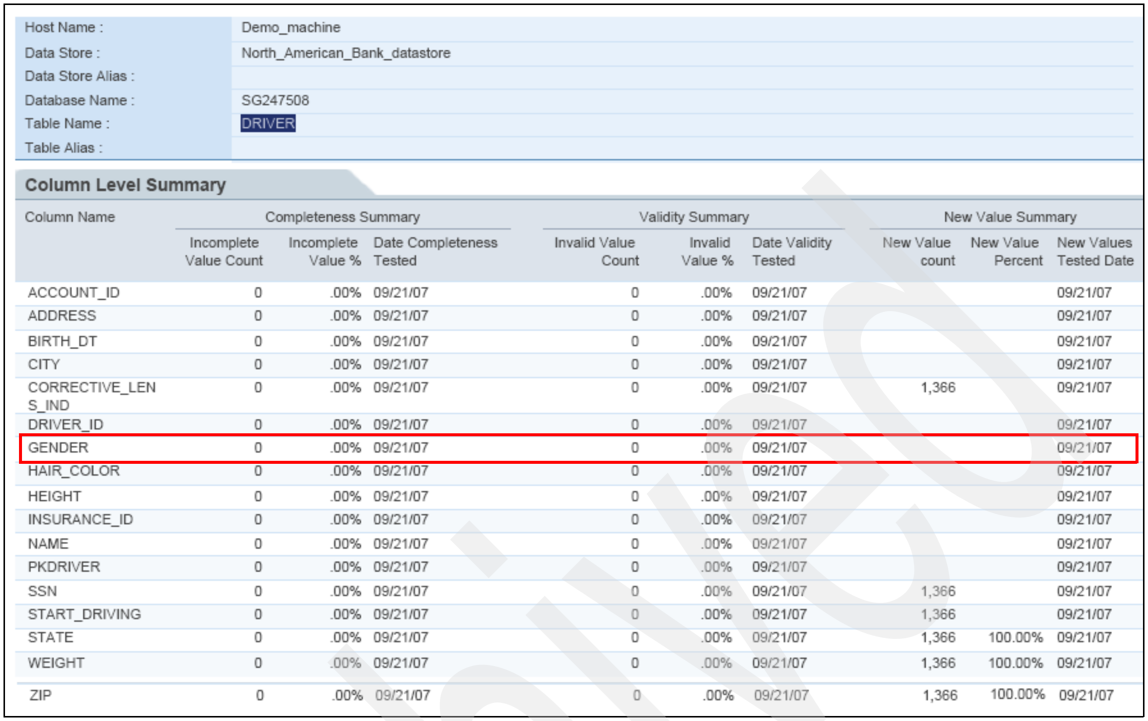
Column Level Summary

Cardinality Count :	2
Null Count :	0
Actual Row Count :	1366
Frequency Cut Off :	10
Total Rows Covered :	1366
%Rows Covered :	100.00

Frequency Distribution Data

Distinct Value	Frequency Count	Frequency%	Cumulative %
F	699	51.17	.00
M	667	48.83	.00

Figure 2-50 Column Frequency “Frequency By Frequency” report for the DRIVER table



Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	DRIVER
Table Alias :	

Column Level Summary

Column Name	Completeness Summary			Validity Summary			New Value Summary		
	Incomplete Value Count	Incomplete Value %	Date Completeness Tested	Invalid Value Count	Invalid Value %	Date Validity Tested	New Value count	New Value Percent	New Values Tested Date
ACCOUNT_ID	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
ADDRESS	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
BIRTH_DT	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
CITY	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
CORRECTIVE_LENS_IND	0	.00%	09/21/07	0	.00%	09/21/07	1,366		09/21/07
DRIVER_ID	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
GENDER	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
HAIR_COLOR	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
HEIGHT	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
INSURANCE_ID	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
NAME	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
PKDRIVER	0	.00%	09/21/07	0	.00%	09/21/07			09/21/07
SSN	0	.00%	09/21/07	0	.00%	09/21/07	1,366		09/21/07
START_DRIVING	0	.00%	09/21/07	0	.00%	09/21/07	1,366		09/21/07
STATE	0	.00%	09/21/07	0	.00%	09/21/07	1,366	100.00%	09/21/07
WEIGHT	0	.00%	09/21/07	0	.00%	09/21/07	1,366	100.00%	09/21/07
ZIP	0	.00%	09/21/07	0	.00%	09/21/07	1,366	100.00%	09/21/07

Figure 2-51 Column Domain Values "Completeness and Validity Summary" report for the DRIVER table

Table 2-17 Valid and invalid column values

Column name	Valid values	Invalid values
GENDER in the DRIVER table	"M", "F"	No invalid values

Primary Key Analysis reports

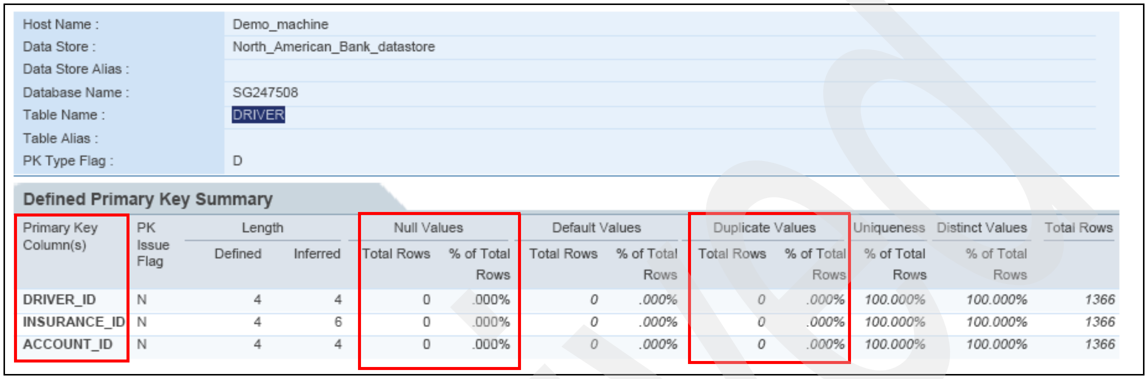
As mentioned earlier, these reports provide information about candidate primary keys as well as duplicate records in the table.

Figure 2-52 on page 489 shows the Primary Key Analysis "Defined Primary Key - Summary" report for the DRIVER table. It identifies the length (defined and inferred), number of null values, default values, uniqueness values, distinct values, and number of rows in the table. This report shows no duplicate records or null values in the table.

Figure 2-53 on page 490 shows the Primary Key Analysis "Duplicate Exceptions" report for the DRIVER table. It identifies duplicate values in each column of the primary key. There are none in this case.

Figure 2-54 on page 491 shows the Primary Key Analysis “Defined & Candidate Summary” report for the BRANCH table. It identifies duplicate values in each column of the primary key. There are none in this case, but candidate keys are identified.

Any duplicate rows or tables with no primary key (but candidate primary keys) should be identified and documented as shown in Table 2-18 on page 491.

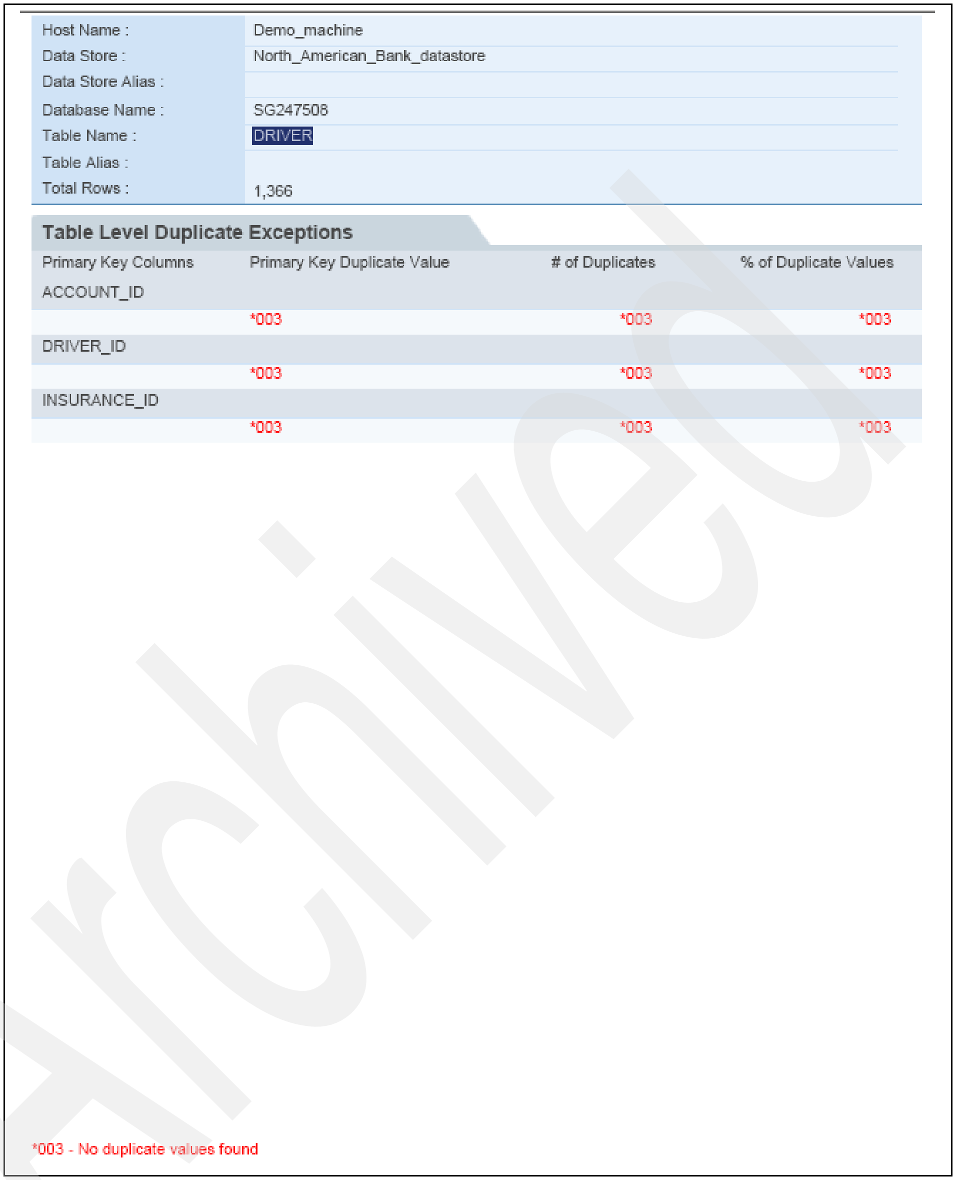


Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	DRIVER
Table Alias :	
PK Type Flag :	D

Defined Primary Key Summary

Primary Key Column(s)	PK Issue Flag	Length		Null Values		Default Values		Duplicate Values		Uniqueness	Distinct Values	Total Rows
		Defined	Inferred	Total Rows	% of Total Rows	Total Rows	% of Total Rows	Total Rows	% of Total Rows	% of Total Rows	% of Total Rows	
DRIVER_ID	N	4	4	0	.000%	0	.000%	0	.000%	100.000%	100.000%	1366
INSURANCE_ID	N	4	6	0	.000%	0	.000%	0	.000%	100.000%	100.000%	1366
ACCOUNT_ID	N	4	4	0	.000%	0	.000%	0	.000%	100.000%	100.000%	1366

Figure 2-52 Primary Key Analysis “Defined Primary Key - Summary” report for the DRIVER table



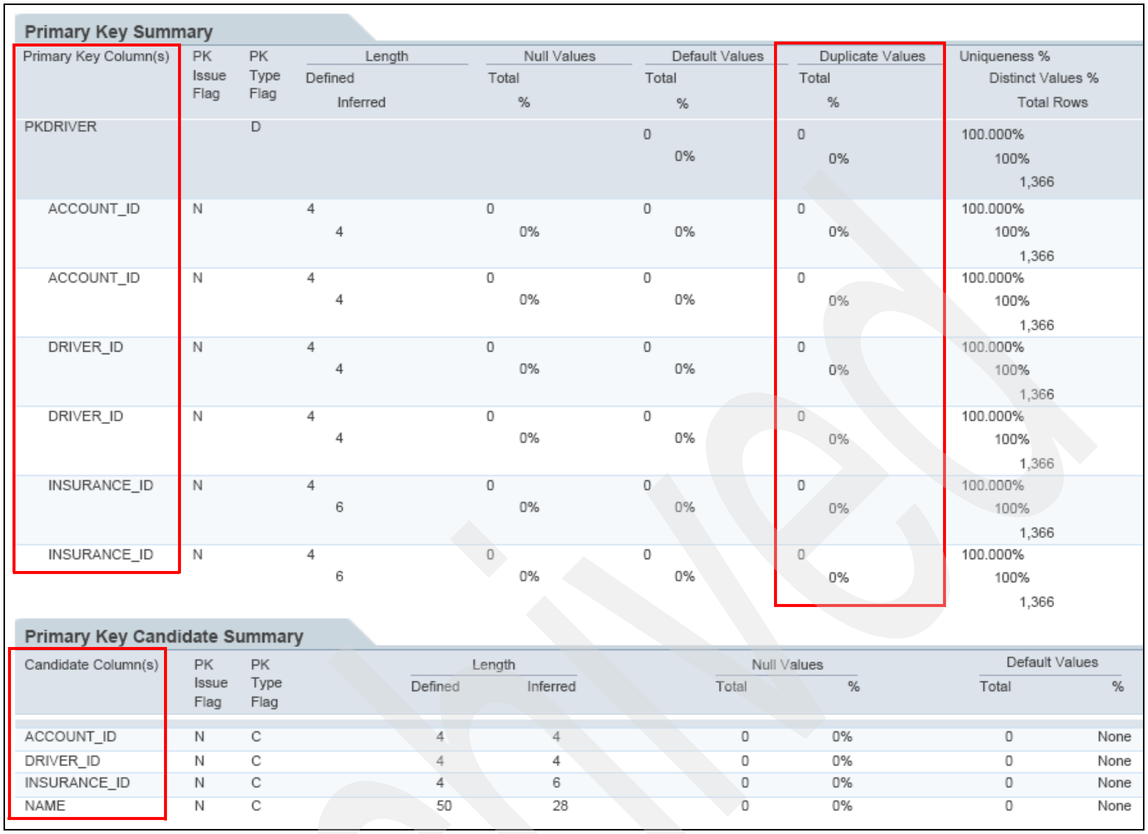
Host Name :	Demo_machine
Data Store :	North_American_Bank_datastore
Data Store Alias :	
Database Name :	SG247508
Table Name :	DRIVER
Table Alias :	
Total Rows :	1,366

Table Level Duplicate Exceptions

Primary Key Columns	Primary Key Duplicate Value	# of Duplicates	% of Duplicate Values
ACCOUNT_ID			
	*003	*003	*003
DRIVER_ID			
	*003	*003	*003
INSURANCE_ID			
	*003	*003	*003

*003 - No duplicate values found

Figure 2-53 Primary Key Analysis "Duplicate Exceptions" report for DRIVER table



Primary Key Summary

Primary Key Column(s)	PK Issue Flag	PK Type Flag	Length		Null Values		Default Values		Duplicate Values		Uniqueness %		
			Defined	Inferred	Total	%	Total	%	Total	%		Distinct Values %	Total Rows
PKDRIVER		D					0	0%	0	0%	100.000%	100%	1,366
ACCOUNT_ID	N		4	4	0	0%	0	0%	0	0%	100.000%	100%	1,366
ACCOUNT_ID	N		4	4	0	0%	0	0%	0	0%	100.000%	100%	1,366
DRIVER_ID	N		4	4	0	0%	0	0%	0	0%	100.000%	100%	1,366
DRIVER_ID	N		4	4	0	0%	0	0%	0	0%	100.000%	100%	1,366
INSURANCE_ID	N		4	6	0	0%	0	0%	0	0%	100.000%	100%	1,366
INSURANCE_ID	N		4	6	0	0%	0	0%	0	0%	100.000%	100%	1,366

Primary Key Candidate Summary

Candidate Column(s)	PK Issue Flag	PK Type Flag	Length		Null Values		Default Values	
			Defined	Inferred	Total	%	Total	%
ACCOUNT_ID	N	C	4	4	0	0%	0	None
DRIVER_ID	N	C	4	4	0	0%	0	None
INSURANCE_ID	N	C	4	6	0	0%	0	None
NAME	N	C	50	28	0	0%	0	None

Figure 2-54 Primary Key Analysis "Defined & Candidate Summary" report for the DRIVER table

Table 2-18 Primary Key violations

Primary Key Table Name	Primary Key Column Name	Invalid values
DRIVER	DRIVER_ID, INSURANCE_ID, ACCOUNT_ID	

Foreign Key Analysis reports

These reports provide information about referential integrity violations for explicitly defined foreign keys, and can identify candidate foreign keys.

Instead of using IA's foreign key analysis reports, as a matter of convenience and familiarity with existing procedures, we checked for referential integrity violations using the CHECK DATA utility of DB2 for z/OS on all the non-core services tables with declared foreign keys. This is not shown here. No referential integrity violations were found.

Any and all violations should be documented for future correction as shown in Table 2-12 on page 461.

Baseline Analysis reports

As mentioned earlier, baseline analysis is used to compare a prior version of analysis results with the current analysis results for a given data source. If differences between both versions are found, you can assess the significance of the change, such as whether you need to re-evaluate the data integration strategy and execution plan. When assessing the progress of development and data quality assessment work, Baseline Analyses are particularly useful to show the changes between one run of a process and the next, and whether the changes made to correct issues have resulted in improved output.

Figure 2-55 on page 493 shows the partial report of the Baseline Structure “Current to Prior Structural Variances” report for the DRIVER table. It identifies the current to baseline variances in data types, length, precision, and scale for each of the columns in the table. This report shows no variances.

Figure 2-56 on page 494 and Figure 2-57 on page 495 show the partial report of the Baseline Structure “Current to Prior Variances” report for the DRIVER table. It identifies the variances in data content for each column in the table. This report shows variations in data content by column such as GENDER. You should then use the column frequency distribution to identify the specify content differences (this is not shown here).

All the structural and data content variances should be documented for potential action.

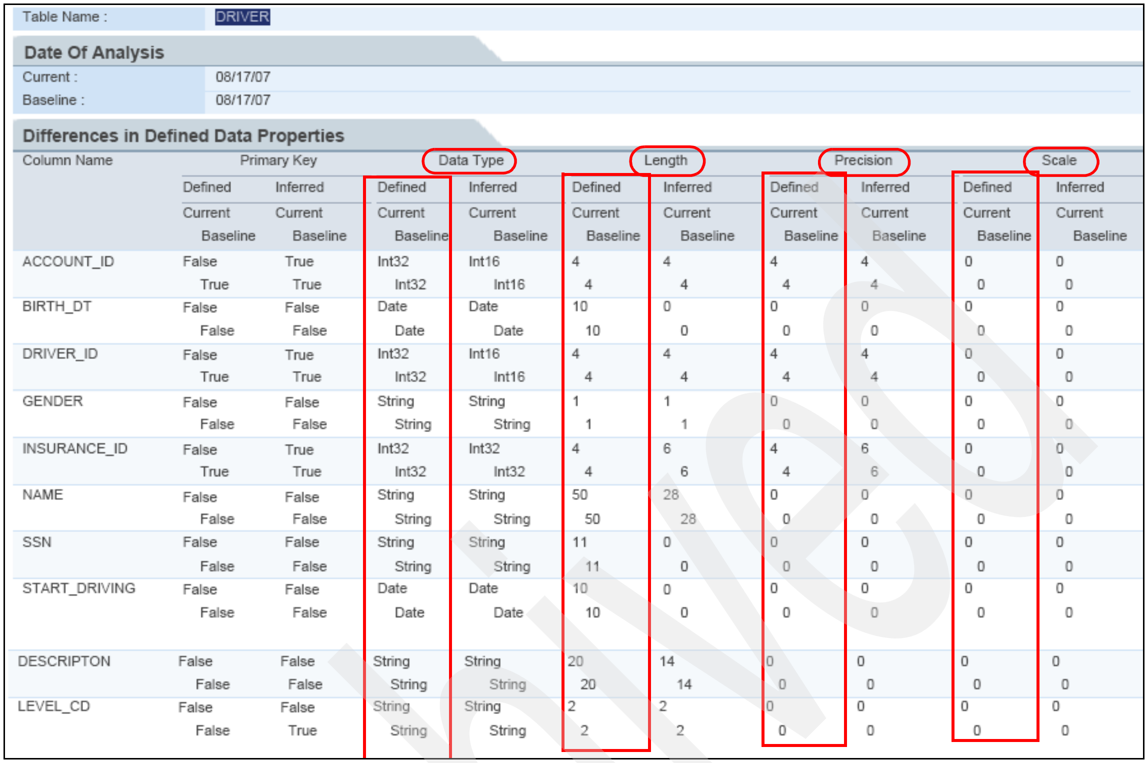


Table Name :	DRIVER

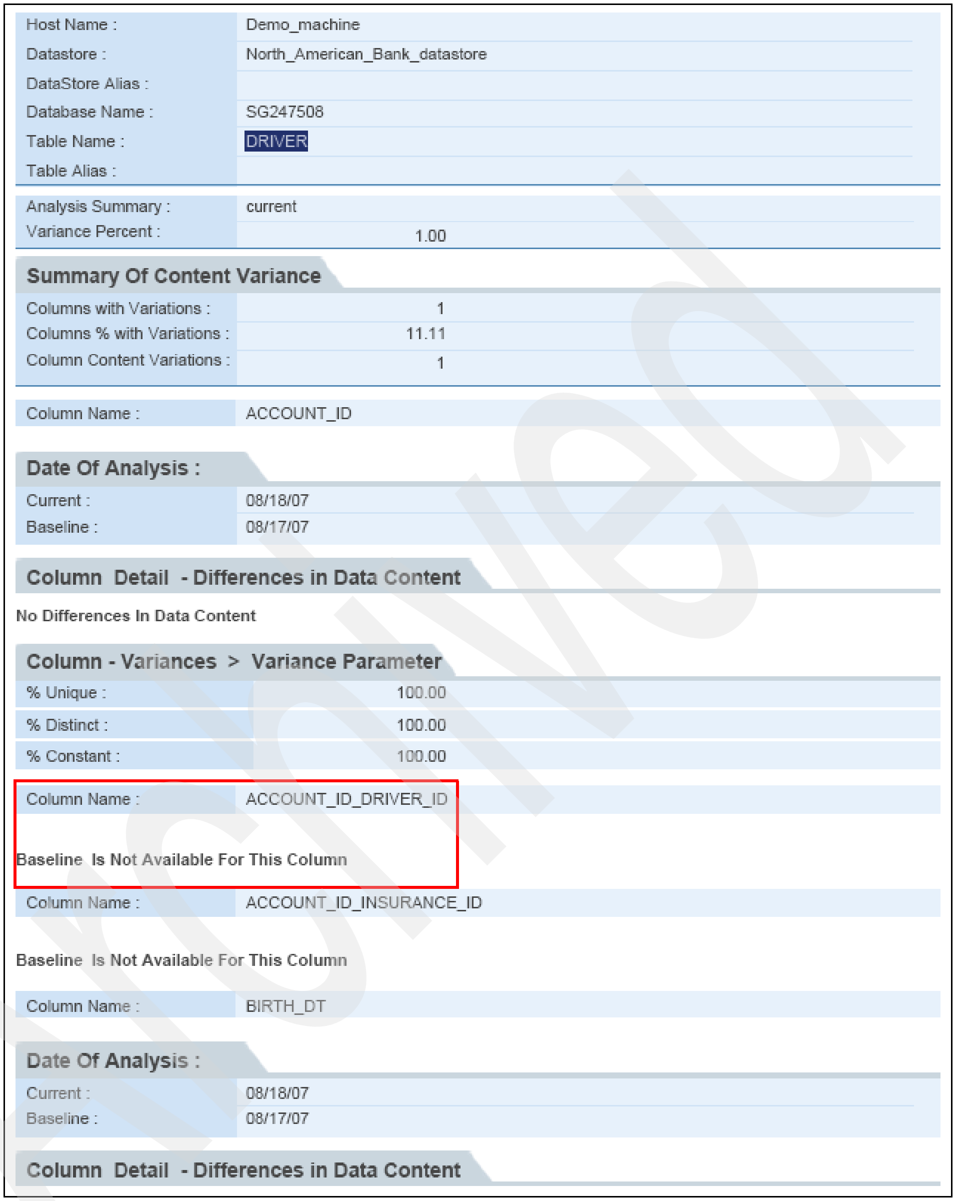
Date Of Analysis

Current :	08/17/07
Baseline :	08/17/07

Differences in Defined Data Properties

Column Name	Primary Key		Data Type		Length		Precision		Scale	
	Defined	Inferred	Defined	Inferred	Defined	Inferred	Defined	Inferred	Defined	Inferred
	Current Baseline	Current Baseline	Current Baseline	Current Baseline	Current Baseline	Current Baseline	Current Baseline	Current Baseline	Current Baseline	Current Baseline
ACCOUNT_ID	False True	True True	Int32 Int32	Int16 Int16	4 4	4 4	4 4	4 4	0 0	0 0
BIRTH_DT	False False	False False	Date Date	Date Date	10 10	0 0	0 0	0 0	0 0	0 0
DRIVER_ID	False True	True True	Int32 Int32	Int16 Int16	4 4	4 4	4 4	4 4	0 0	0 0
GENDER	False False	False False	String String	String String	1 1	1 1	0 0	0 0	0 0	0 0
INSURANCE_ID	False True	True True	Int32 Int32	Int32 Int32	4 4	6 6	4 4	6 6	0 0	0 0
NAME	False False	False False	String String	String String	50 50	28 28	0 0	0 0	0 0	0 0
SSN	False False	False False	String String	String String	11 11	0 0	0 0	0 0	0 0	0 0
START_DRIVING	False False	False False	Date Date	Date Date	10 10	0 0	0 0	0 0	0 0	0 0
DESCRIPTON	False False	False False	String String	String String	20 20	14 14	0 0	0 0	0 0	0 0
LEVEL_CD	False False	False True	String String	String String	2 2	2 2	0 0	0 0	0 0	0 0

Figure 2-55 Baseline Structure “Current to Prior Structural Variances” report for the DRIVER table



Host Name :	Demo_machine
Datastore :	North_American_Bank_datastore
DataStore Alias :	
Database Name :	SG247508
Table Name :	DRIVER
Table Alias :	

Analysis Summary :	current
Variance Percent :	1.00

Summary Of Content Variance

Columns with Variations :	1
Columns % with Variations :	11.11
Column Content Variations :	1

Column Name :	ACCOUNT_ID

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

No Differences In Data Content

Column - Variances > Variance Parameter

% Unique :	100.00
% Distinct :	100.00
% Constant :	100.00

Column Name :	ACCOUNT_ID_DRIVER_ID

Baseline Is Not Available For This Column

Column Name :	ACCOUNT_ID_INSURANCE_ID

Baseline Is Not Available For This Column

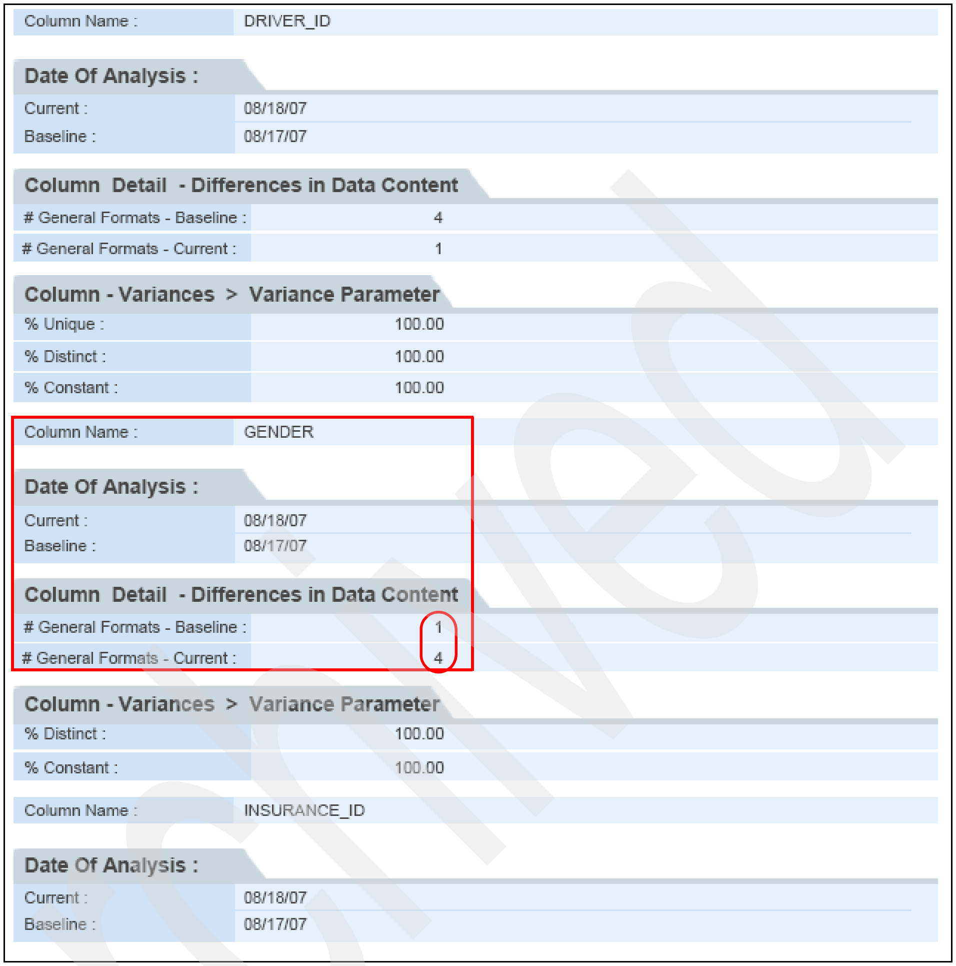
Column Name :	BIRTH_DT

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

Figure 2-56 Baseline Structure "Current to Prior Variances" report for the DRIVER table 1/2



Column Name :	DRIVER_ID

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	4
# General Formats - Current :	1

Column - Variances > Variance Parameter

% Unique :	100.00
% Distinct :	100.00
% Constant :	100.00

Column Name :	GENDER

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	1
# General Formats - Current :	4

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00

Column Name :	INSURANCE_ID

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Figure 2-57 Baseline Structure "Current to Prior Variances" report for the DRIVER table 2/2

IBM WebSphere AuditStage business rule compliance

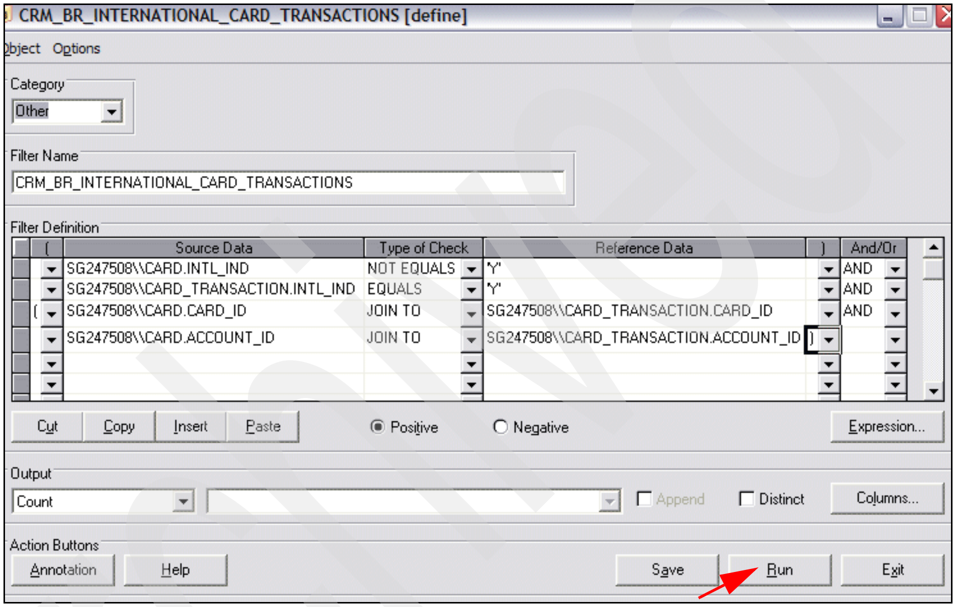
As mentioned earlier, Business Rule Compliance evaluates the quality of data in terms of specific business rules involving multiple data fields within or across records (or rows) that are logically related. IBM WebSphere AuditStage is used to perform this check. In this section, we do not describe the process of building the data filter because that is described in 1.12, "IBM WebSphere AuditStage business rule validation" on page 336 but focus on the completed data filter specification and the results of executing the business rule.

The business rules described in Table 2-16 on page 485 are verified using the data filters developed using IBM WebSphere AuditStage.

1: Check INTL_IND column and transactions

Figure 2-60 on page 497 shows the data filter for verifying the business rule that international transactions can only occur in the CARD_TRANSACTION table when the INTL_IND column has a value of “Y” in the CARD table. After building the data filter which checks for international transactions (INTL_IND = “Y”) in the CARD_TRANSACTION table when the INTL_IND in the CARD table is not “Y” for each card holder, click **Run** to execute the data filter.

Figure 2-59 on page 496 shows the number of exceptions as being zero.



Category: Other

Filter Name: CRM_BR_INTERNATIONAL_CARD_TRANSACTIONS

Filter Definition

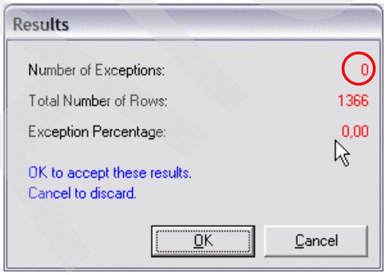
(	Source Data	Type of Check	Reference Data	)	And/Or
	SG247508\\CARD.INTL_IND	NOT EQUALS	'Y'		AND
	SG247508\\CARD_TRANSACTION.INTL_IND	EQUALS	'Y'		AND
(	SG247508\\CARD.CARD_ID	JOIN TO	SG247508\\CARD_TRANSACTION.CARD_ID		AND
	SG247508\\CARD.ACCOUNT_ID	JOIN TO	SG247508\\CARD_TRANSACTION.ACCOUNT_ID	)	

● Positive ○ Negative

Output: Count □ Append □ Distinct

Action Buttons

Figure 2-58 Check INTL_IND column and transactions 1/2



Number of Exceptions: 0

Total Number of Rows: 1366

Exception Percentage: 0,00

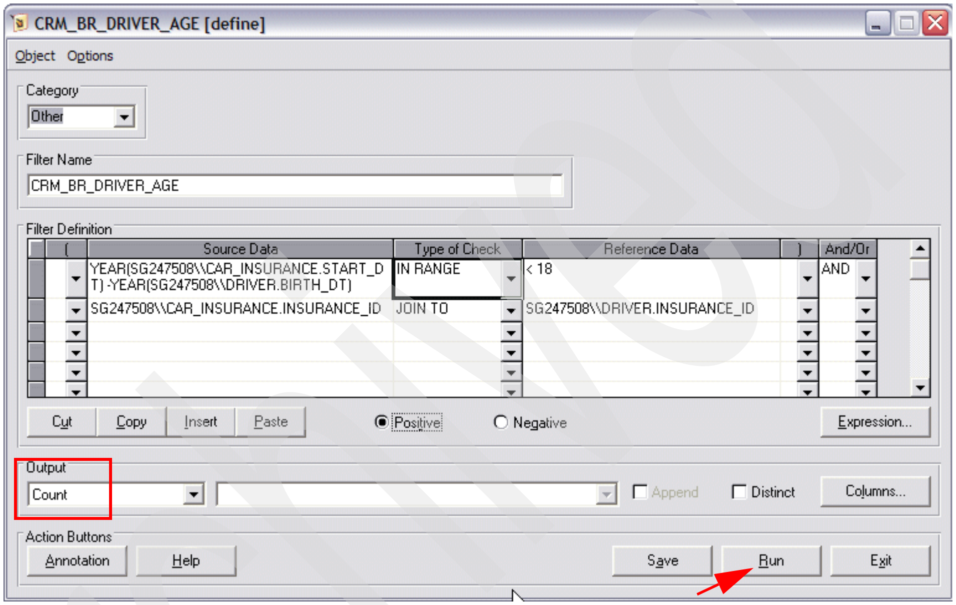
OK to accept these results.
Cancel to discard.

Figure 2-59 Check INTL_IND column and transactions 2/2

2: Check age of driver less than 18

Figure 2-60 on page 497 shows the data filter for verifying the business rule that the driver's age must be greater than 18 when insurance is granted. After building the data filter which checks for the difference in years between the START_DT column in the CAR_INSURANCE table and the BIRTH_DT column in the DRIVER table to be less than 18, click **Run** to execute the data filter.

Figure 2-59 on page 496 shows the number of exceptions as being zero.



Category: Other

Filter Name: CRM_BR_DRIVER_AGE

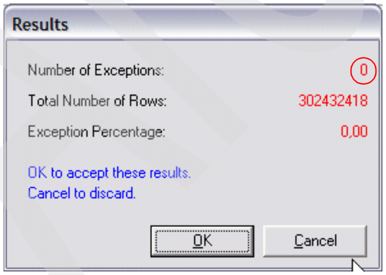
Filter Definition

(	Source Data	Type of Check	Reference Data	)	And/Or
	YEAR(SG247508\\CAR_INSURANCE.START_DT)-YEAR(SG247508\\DRIVER.BIRTH_DT)	IN RANGE	< 18		AND
	SG247508\\CAR_INSURANCE.INSURANCE_ID	JOIN TO	SG247508\\DRIVER.INSURANCE_ID		

● Positive ○ Negative

Output: Count □ Append □ Distinct

Figure 2-60 Check INTL_IND column and transactions 1/2



Number of Exceptions: 0

Total Number of Rows: 302432418

Exception Percentage: 0,00

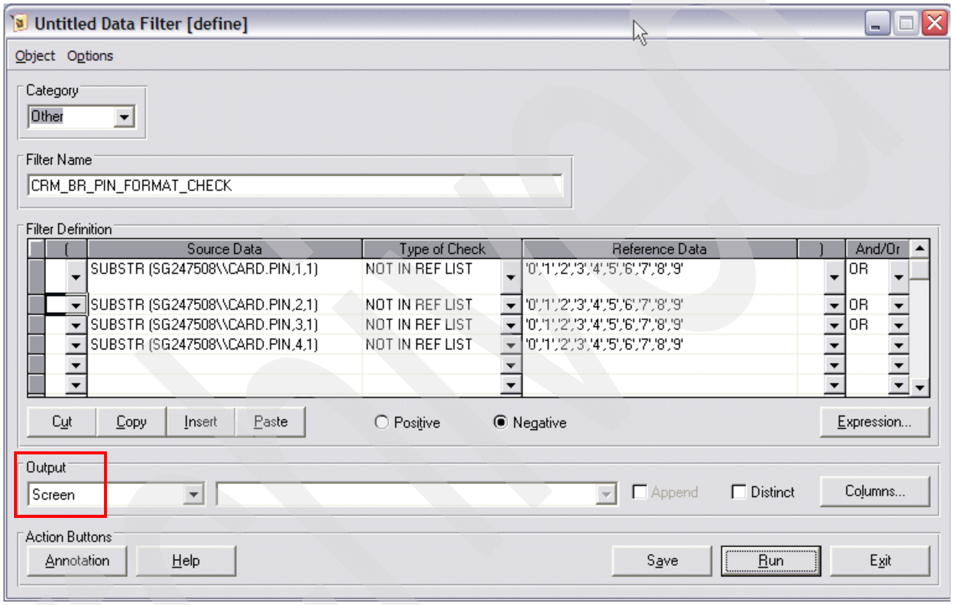
OK to accept these results.
Cancel to discard.

Figure 2-61 Check INTL_IND column and transactions 2/2

3: Check PIN column is completely numeric

Figure 2-62 shows the data filter for verifying the business rule that the PIN number in the PIN column of the DRIVER table must be all numeric. After building the data filter which checks whether any of the individual characters in the PIN column in the DRIVER table is not numeric with the output being sent to the screen, click **Run** to execute the data filter.

Figure 2-63 shows the number of exceptions. One row qualifies as shown in Figure 2-64 on page 499.



Category: Other

Filter Name: CRM_BR_PIN_FORMAT_CHECK

Filter Definition

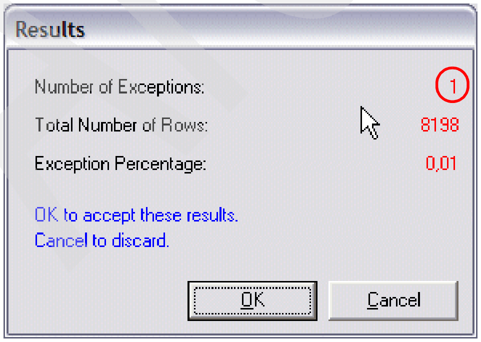
(	Source Data	Type of Check	Reference Data	)	And/Or
	SUBSTR (SG247508\\CARD.PIN,1,1)	NOT IN REF LIST	'0','1','2','3','4','5','6','7','8','9'		OR
	SUBSTR (SG247508\\CARD.PIN,2,1)	NOT IN REF LIST	'0','1','2','3','4','5','6','7','8','9'		OR
	SUBSTR (SG247508\\CARD.PIN,3,1)	NOT IN REF LIST	'0','1','2','3','4','5','6','7','8','9'		OR
	SUBSTR (SG247508\\CARD.PIN,4,1)	NOT IN REF LIST	'0','1','2','3','4','5','6','7','8','9'		

○ Positive ● Negative

Output: Screen

□ Append □ Distinct

Figure 2-62 Check PIN column is completely numeric 1/3



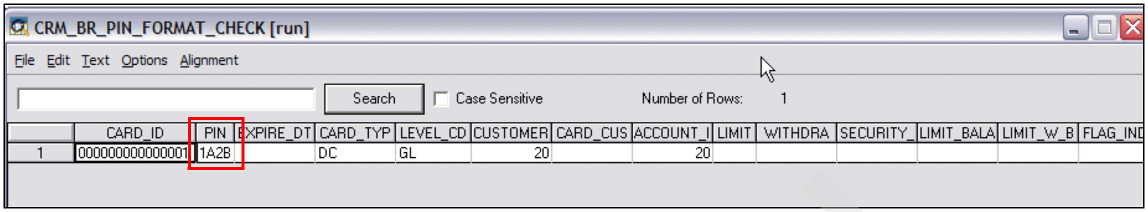
Number of Exceptions: 1

Total Number of Rows: 8198

Exception Percentage: 0,01

OK to accept these results.
Cancel to discard.

Figure 2-63 Check PIN column is completely numeric 2/3


CRM_BR_PIN_FORMAT_CHECK [run]
File Edit Text Options Alignment
Search
Case Sensitive

Number of Rows: 1

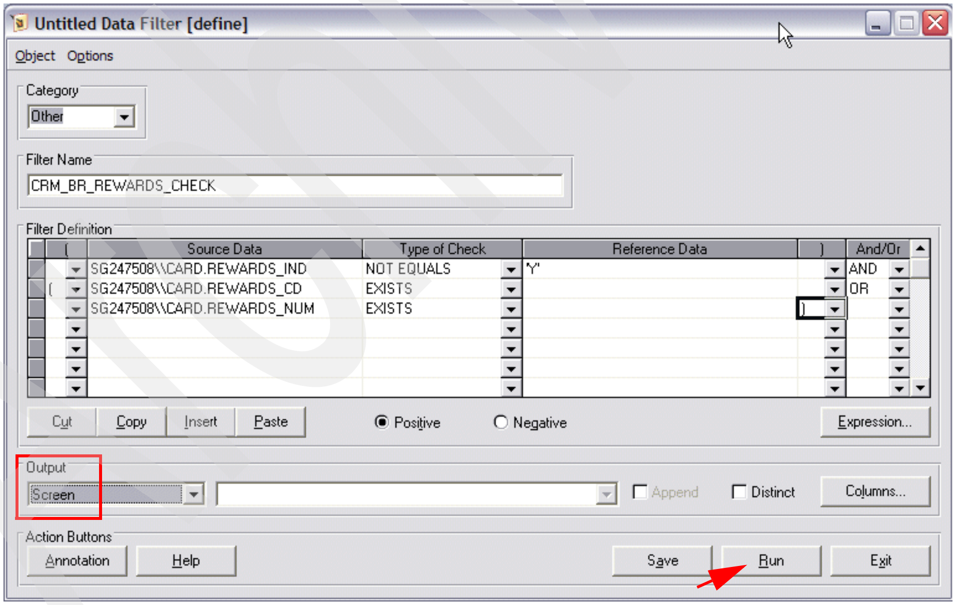
	CARD_ID	PIN	EXPIRE_DT	CARD_TYP	LEVEL_CD	CUSTOMER	CARD_CUS	ACCOUNT_I	LIMIT	WITHDRA	SECURITY_	LIMIT_BALA	LIMIT_W_B	FLAG_IND
1	000000000000001	1A2B		DC	GL	20		20						

Figure 2-64 Check PIN column is completely numeric 3/3

4: Check REWARDS_IND column and associated values

Figure 2-65 shows the data filter for verifying the business rule that the REWARDS_NUM column and REWARDS_CD column in the CARD table can only have values when the REWARDS_IND column in the same table has a value of “Y”. After building the data filter which checks for values in the REWARDS_NUM and REWARDS_CD columns when the REWARDS_IND column does not have a value of “Y” with the output being sent to the screen, click **Run** to execute the data filter.

Figure 2-66 on page 500 shows zero exceptions being generated.



Category: Other

Filter Name: CRM_BR_REWARDS_CHECK

Filter Definition

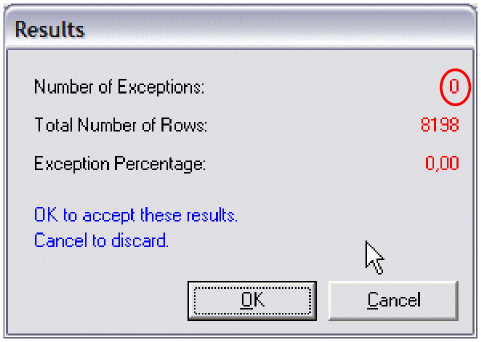
(	Source Data	Type of Check	Reference Data	)	And/Or
	SG247508\\CARD.REWARDS_IND	NOT EQUALS	'Y'		AND
(	SG247508\\CARD.REWARDS_CD	EXISTS			OR
	SG247508\\CARD.REWARDS_NUM	EXISTS		)	

● Positive ○ Negative

Output: Screen

□ Append □ Distinct

Figure 2-65 Check REWARDS_IND column and associated values 1/2



Results

Number of Exceptions: 0

Total Number of Rows: 8198

Exception Percentage: 0,00

OK to accept these results.
Cancel to discard.

Figure 2-66 Check REWARDS_IND column and associated values 2/2

2.6.5 Northern California Bank non-core services analysis

As mentioned earlier, Northern California Bank's non-core services are to be integrated into the CRM system.

Using metadata information available in dictionaries, documentation and the relational catalogs, the keys, code fields, indicator fields, and referential integrity relationships (both implicit and explicit) to be profiled were identified as shown in Table 2-19 on page 501.

Note: For convenience, we chose to include all the columns in all the tables in our data profiling effort. In the real world, however, to avoid report and information overload, you would apply the 80/20 rule and focus on only the critical master data as described in 1.2.1, "Data assessment approach" on page 5.

We are only including selected portions of the generated reports (for some of the fields shown in Table 2-19 on page 501) of the data profiling effort in this section. You can download the complete reports from the IBM Redbooks Web site:

`ftp://www.redbooks.ibm.com/redbooks/SG247508/`

Note: We describe the process of generating reports in 1.14, "Reports" on page 394 and do not repeat that information here. We show only the relevant portions of the reports here.

Business rules involving data elements in one or more tables are shown in Table 2-20 on page 501.These are verified using IBM WebSphere AuditStage.

Table 2-19 Main keys, codes, and indicators in Northern California Bank's brokerage services

Table	Key column(s)	Code column	Indicator column	RI relationships
BCUSTOMER	ID			► BRANCH to the BRANCH table ► ADVISOR to the EMPLOYEE table
BACCOUNT	ID	► TYPE		
BROKERAGE	OWNER, ACCOUNT, PORTFOLIO			► OWNER to the BCUSTOMER table ► ACCOUNT to the BACCOUNT table ► PORTFOLIO to the PORTFOLIO table
PORTFOLIO	ID	► SYMBOL ► CURRENCY		CURRENCY to the CURRENCY table
EMPLOYEE	ID			BRANCH to the BRANCH table
BRANCH	ID	► ZIP ► COUNTRY		COUNTRY to the COUNTRY table
CURRENCY	ID			

Table 2-20 Business rules in Northern California Bank's brokerage services

Serial number	Description
1	Customer data in the BCUSTOMER table can only be updated by the employee who is the advisor to that brokerage customer. The ADVISOR column in the BCUSTOMER table identifies its advisor employee in the EMPLOYEE table. The ADVISOR column might be null. The BY column in the BCUSTOMER table should correspond to the USERID column in the EMPLOYEE table.

Column Analysis reports

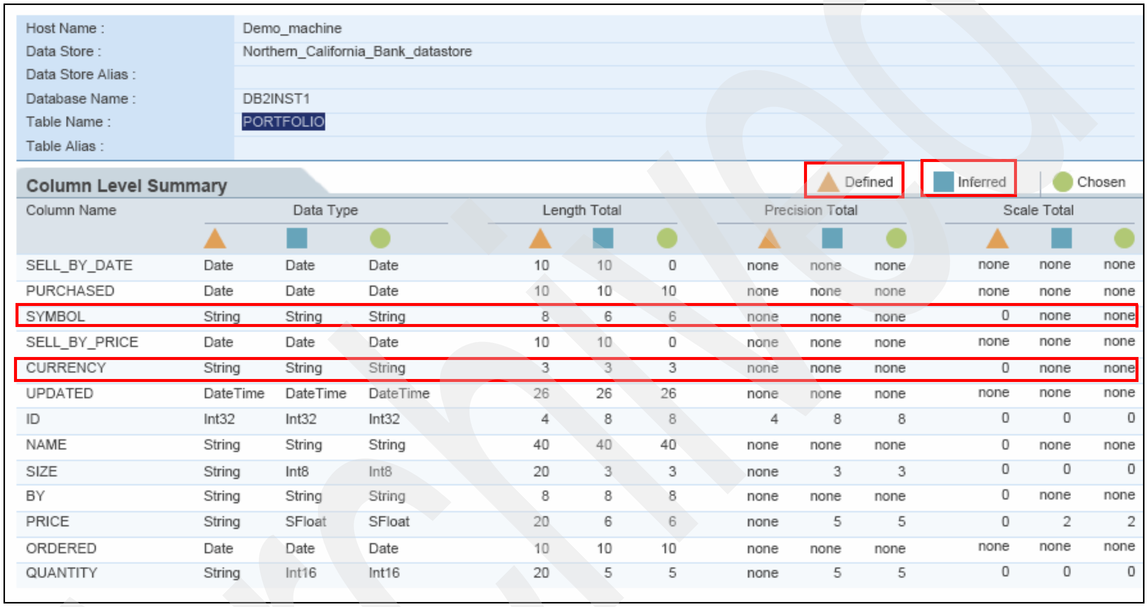
As mentioned earlier, these reports provide information about column values for completeness, validity, structure, and format as described in 1.7, "Column analysis" on page 109. These reports are used to review the columns for domain, structure and format integrity.

Figure 2-67 on page 502 shows the Column Inferences "Type & Length Properties (Defined/Inferred/Chosen)" report for the PORTFOLIO table. It identifies the data type, length, precision and scale of each column as defined in the metadata and inferred from the data content. Any selected status of each of these categories is also shown here. This information (along with those relating to other columns in other tables) is recorded in Table 2-26 on page 514. Our focus is on keys, codes, and indicators as highlighted.

Figure 2-68 on page 503 shows the Column Frequency “Frequency By Frequency” report for the CURRENCY column in the PORTFOLIO table. It shows a single valid value “USD” with no invalid values.

Figure 2-69 on page 504 shows the Column Domain Values “Completeness and Validity Summary” report for the PORTFOLIO table. It shows no occurrences of incomplete values or invalid values in the CURRENCY column.

Document all the valid values for a column in a table as shown in Table 2-21 on page 504.



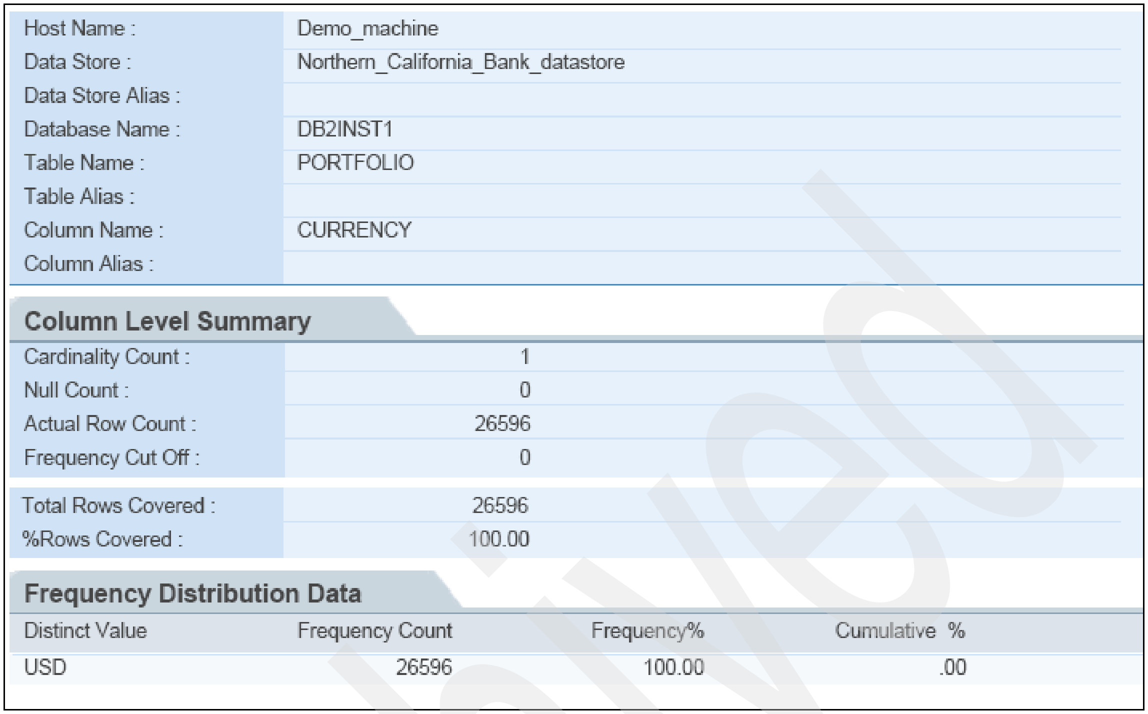
Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	PORTFOLIO
Table Alias :	

Column Level Summary

▲ Defined ■ Inferred ● Chosen

Column Name	Data Type			Length Total			Precision Total			Scale Total		
	▲	■	●	▲	■	●	▲	■	●	▲	■	●
SELL_BY_DATE	Date	Date	Date	10	10	0	none	none	none	none	none	none
PURCHASED	Date	Date	Date	10	10	10	none	none	none	none	none	none
SYMBOL	String	String	String	8	6	6	none	none	none	0	none	none
SELL_BY_PRICE	Date	Date	Date	10	10	0	none	none	none	none	none	none
CURRENCY	String	String	String	3	3	3	none	none	none	0	none	none
UPDATED	DateTime	DateTime	DateTime	26	26	26	none	none	none	none	none	none
ID	Int32	Int32	Int32	4	8	8	4	8	8	0	0	0
NAME	String	String	String	40	40	40	none	none	none	0	none	none
SIZE	String	Int8	Int8	20	3	3	none	3	3	0	0	0
BY	String	String	String	8	8	8	none	none	none	0	none	none
PRICE	String	SFloat	SFloat	20	6	6	none	5	5	0	2	2
ORDERED	Date	Date	Date	10	10	10	none	none	none	none	none	none
QUANTITY	String	Int16	Int16	20	5	5	none	5	5	0	0	0

Figure 2-67 Column Inferences “Type & Length Properties (Defined/Inferred/Chosen)” report for the PORTFOLIO table



Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	PORTFOLIO
Table Alias :	
Column Name :	CURRENCY
Column Alias :	

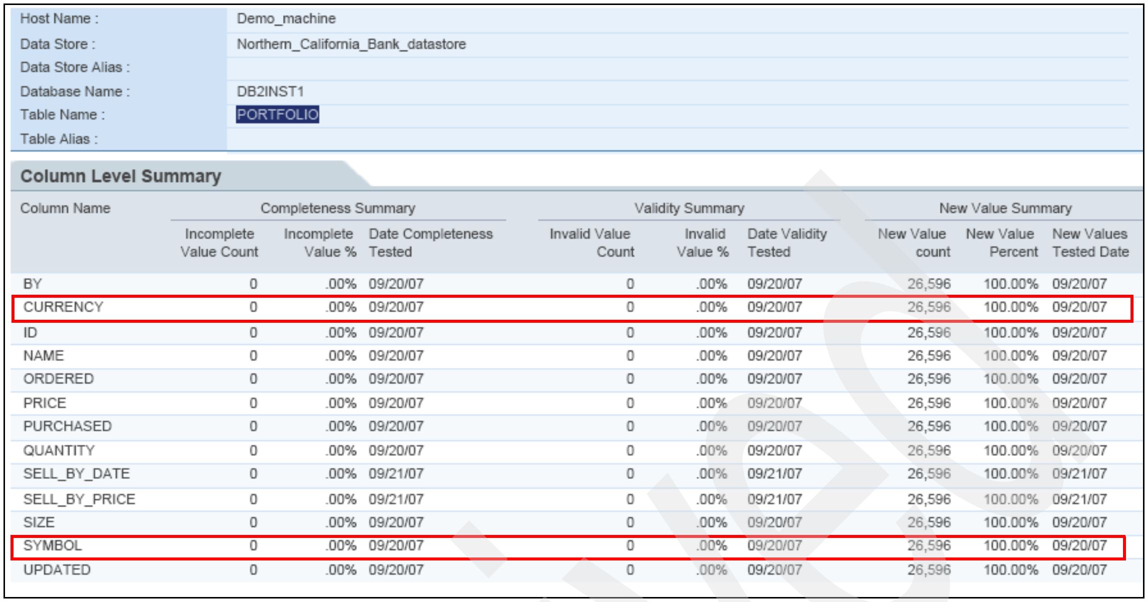
Column Level Summary

Cardinality Count :	1
Null Count :	0
Actual Row Count :	26596
Frequency Cut Off :	0
Total Rows Covered :	26596
%Rows Covered :	100.00

Frequency Distribution Data

Distinct Value	Frequency Count	Frequency%	Cumulative %
USD	26596	100.00	.00

Figure 2-68 Column Frequency "Frequency By Frequency" report for the PORTFOLIO table



Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	PORTFOLIO
Table Alias :	

Column Level Summary

Column Name	Completeness Summary			Validity Summary			New Value Summary		
	Incomplete Value Count	Incomplete Value %	Date Completeness Tested	Invalid Value Count	Invalid Value %	Date Validity Tested	New Value count	New Value Percent	New Values Tested Date
BY	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
CURRENCY	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
ID	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
NAME	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
ORDERED	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
PRICE	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
PURCHASED	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
QUANTITY	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
SELL_BY_DATE	0	.00%	09/21/07	0	.00%	09/21/07	26,596	100.00%	09/21/07
SELL_BY_PRICE	0	.00%	09/21/07	0	.00%	09/21/07	26,596	100.00%	09/21/07
SIZE	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
SYMBOL	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07
UPDATED	0	.00%	09/20/07	0	.00%	09/20/07	26,596	100.00%	09/20/07

Figure 2-69 Column Domain Values "Completeness and Validity Summary" report for the DRIVER table

Table 2-21 Valid and invalid column values

Column name	Valid values	Invalid values
CURRENCY in the PORTFOLIO table	"USD"	

Primary Key Analysis reports

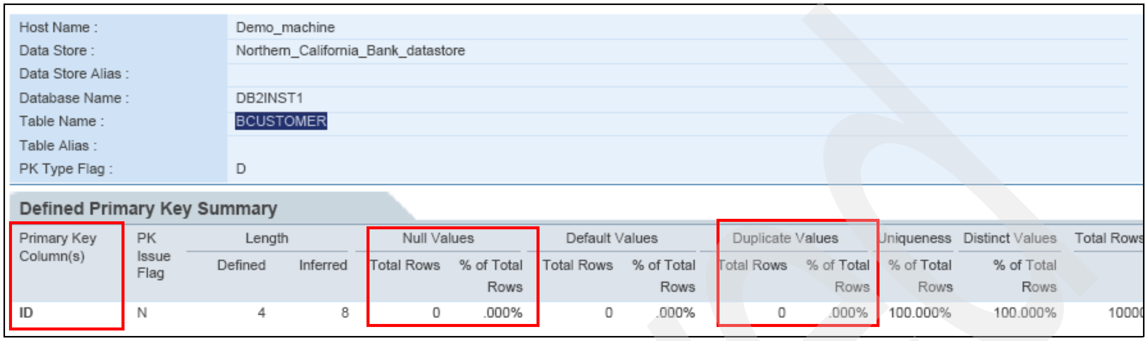
As mentioned earlier, these reports provide information about candidate primary keys as well as duplicate records in the table.

Figure 2-70 on page 505 shows the Primary Key Analysis "Defined Primary Key - Summary" report for the BCUSTOMER table. It identifies the length (defined and inferred), number of null values, default values, uniqueness values, distinct values, and number of rows in the table. This report shows no duplicate records or null values in the table.

Figure 2-71 on page 506 shows the Primary Key Analysis "Duplicate Exceptions" report for the BCUSTOMER table. It identifies duplicate values in each column of the primary key. There are none in this case.

Figure 2-72 on page 507 shows the Primary Key Analysis "Defined & Candidate Summary" report for the BCUSTOMER table. It identifies duplicate values in each column of the primary key. There are none in this case, but candidate keys are identified.

Any duplicate rows or tables with no primary key (but candidate primary keys) should be identified and documented as shown in Table 2-22 on page 507.

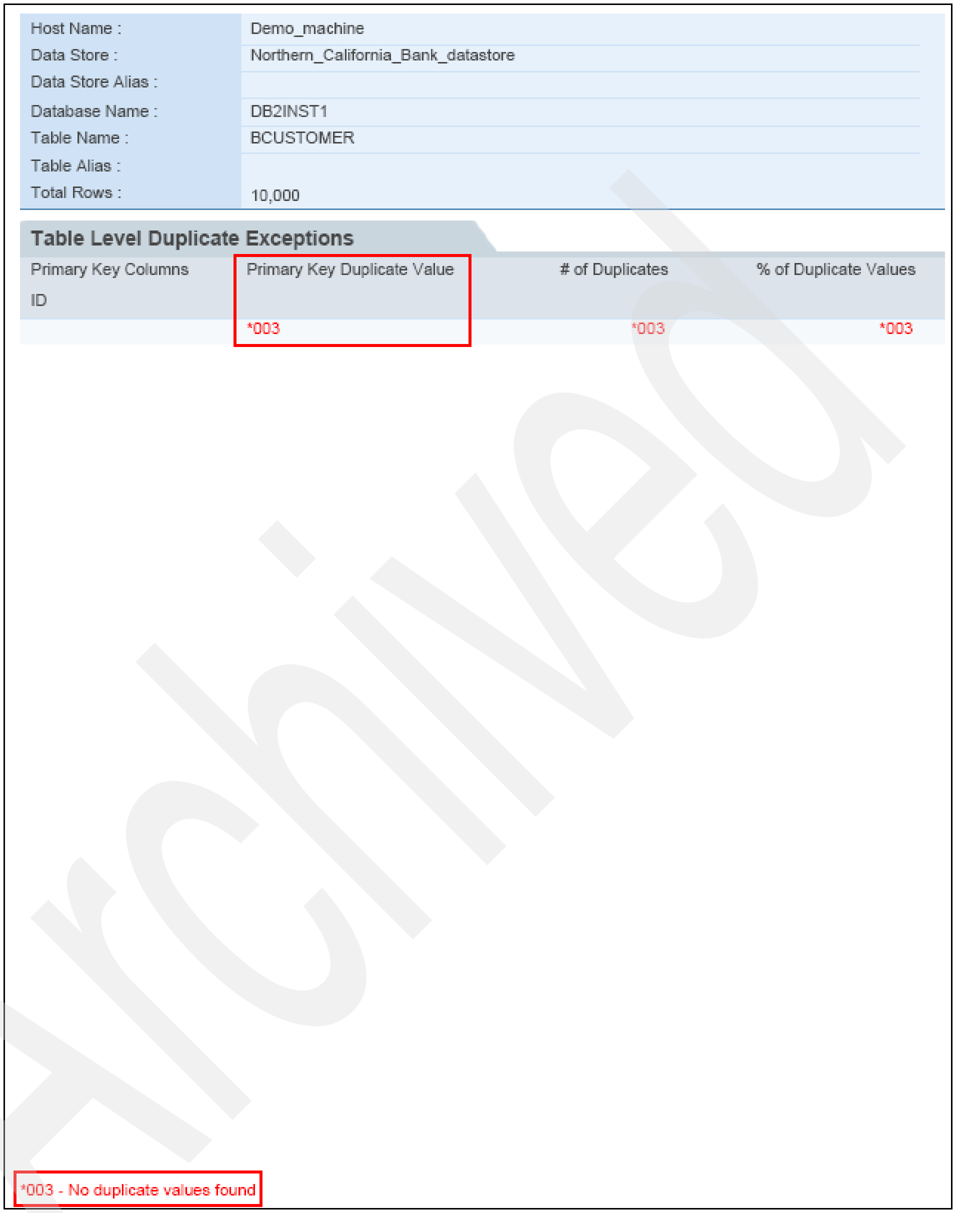


Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	BCUSTOMER
Table Alias :	
PK Type Flag :	D

Defined Primary Key Summary

Primary Key Column(s)	PK Issue Flag	Length		Null Values		Default Values		Duplicate Values		Uniqueness	Distinct Values	Total Rows
		Defined	Inferred	Total Rows	% of Total Rows	Total Rows	% of Total Rows	Total Rows	% of Total Rows	% of Total Rows	% of Total Rows	
ID	N	4	8	0	.000%	0	.000%	0	.000%	100.000%	100.000%	10000

Figure 2-70   Primary Key Analysis "Defined Primary Key - Summary" report for the DRIVER table



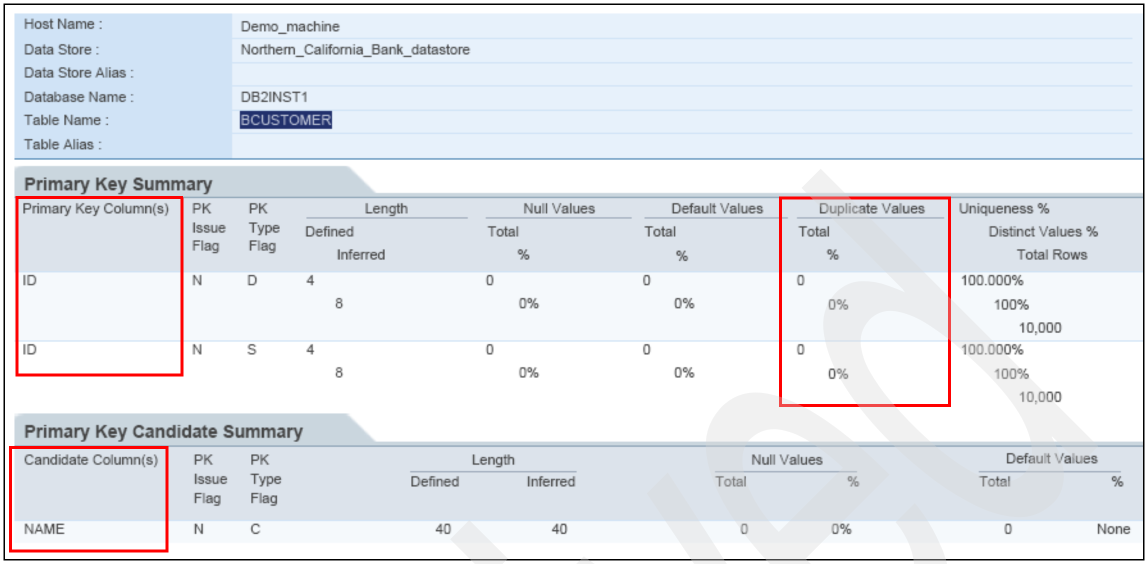
Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	BCUSTOMER
Table Alias :	
Total Rows :	10,000

Table Level Duplicate Exceptions

Primary Key Columns	Primary Key Duplicate Value	# of Duplicates	% of Duplicate Values
ID			
	*003	*003	*003

*003 - No duplicate values found

Figure 2-71 Primary Key Analysis “Duplicate Exceptions” report for DRIVER table



Host Name :	Demo_machine
Data Store :	Northern_California_Bank_datastore
Data Store Alias :	
Database Name :	DB2INST1
Table Name :	BCUSTOMER
Table Alias :	

Primary Key Summary

Primary Key Column(s)	PK Issue Flag	PK Type Flag	Length Defined Inferred	Null Values Total %	Default Values Total %	Duplicate Values Total %	Uniqueness % Distinct Values % Total Rows
ID	N	D	4 8	0 0%	0 0%	0 0%	100.000% 100% 10,000
ID	N	S	4 8	0 0%	0 0%	0 0%	100.000% 100% 10,000

Primary Key Candidate Summary

Candidate Column(s)	PK Issue Flag	PK Type Flag	Length Defined	 Inferred	Null Values Total	 %	Default Values Total	 %
NAME	N	C	40	40	0	0%	0	None

Figure 2-72 Primary Key Analysis "Defined & Candidate Summary" report for the DRIVER table

Table 2-22 Primary Key violations

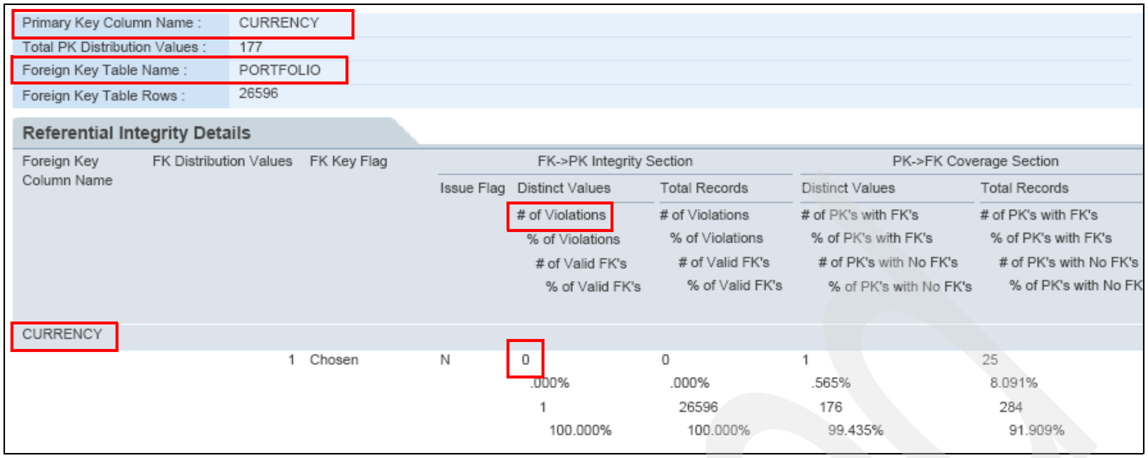
Primary Key Table Name	Primary Key Column Name	Invalid values
BCUSTOMER	ID	

Foreign Key Analysis reports

These reports provide information about referential integrity violations for explicitly defined foreign keys, and can identify candidate foreign keys.

Figure 2-73 on page 508 shows the Foreign Key Analysis "Referential Integrity Detail" report for the PORTFOLIO table. It identifies the primary key column (CURRENCY), foreign key column (CURRENCY) and referential integrity violations. This report shows no foreign key violations.

All violations should be documented for future correction as shown in Table 2-23 on page 508.



Primary Key Column Name :	CURRENCY
Total PK Distribution Values :	177
Foreign Key Table Name :	PORTFOLIO
Foreign Key Table Rows :	26596

Referential Integrity Details

Foreign Key Column Name	FK Distribution Values	FK Key Flag	FK->PK Integrity Section			PK->FK Coverage Section	
			Issue Flag	Distinct Values	Total Records	Distinct Values	Total Records
				# of Violations % of Violations # of Valid FK's % of Valid FK's	# of Violations % of Violations # of Valid FK's % of Valid FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No FK's	# of PK's with FK's % of PK's with FK's # of PK's with No FK's % of PK's with No FK
CURRENCY							
	1	Chosen	N	0 .000% 1 100.000%	0 .000% 26596 100.000%	1 .565% 176 99.435%	25 8.091% 284 91.909%

Figure 2-73 Foreign Key Analysis "Referential Integrity Detail" report for the PORTFOLIO table

Table 2-23 Referential Integrity violations

Foreign Key Table Name	Foreign Key Column Name	Primary Key Table Name	Primary Key Column Name	Invalid values
PORTFOLIO	CURRENCY	CURRENCY	CURRENCY	

Baseline Analysis reports

As mentioned earlier, baseline analysis is used to compare a prior version of analysis results with the current analysis results for a given data source. If differences between both versions are found, you can assess the significance of the change, such as whether you need to re-evaluate the data integration strategy and execution plan.

Figure 2-74 on page 509 shows the partial report of the Baseline Structure "Current to Prior Structural Variances" report for the BCUSTOMER table. It identifies the current to baseline variances in data types, length, precision, and scale for each of the columns in the table. This report shows no variances.

Figure 2-56 on page 494 and Figure 2-57 on page 495 show the partial report of the Baseline Structure "Current to Prior Variances" report for the BCUSTOMER table. It identifies the variances in data content for each column in the table. This report shows variations in data content by column such as CITY. You should then use the column frequency distribution to identify the specify content differences (this is not shown here).

All the structural and data content variances should be documented for potential action.

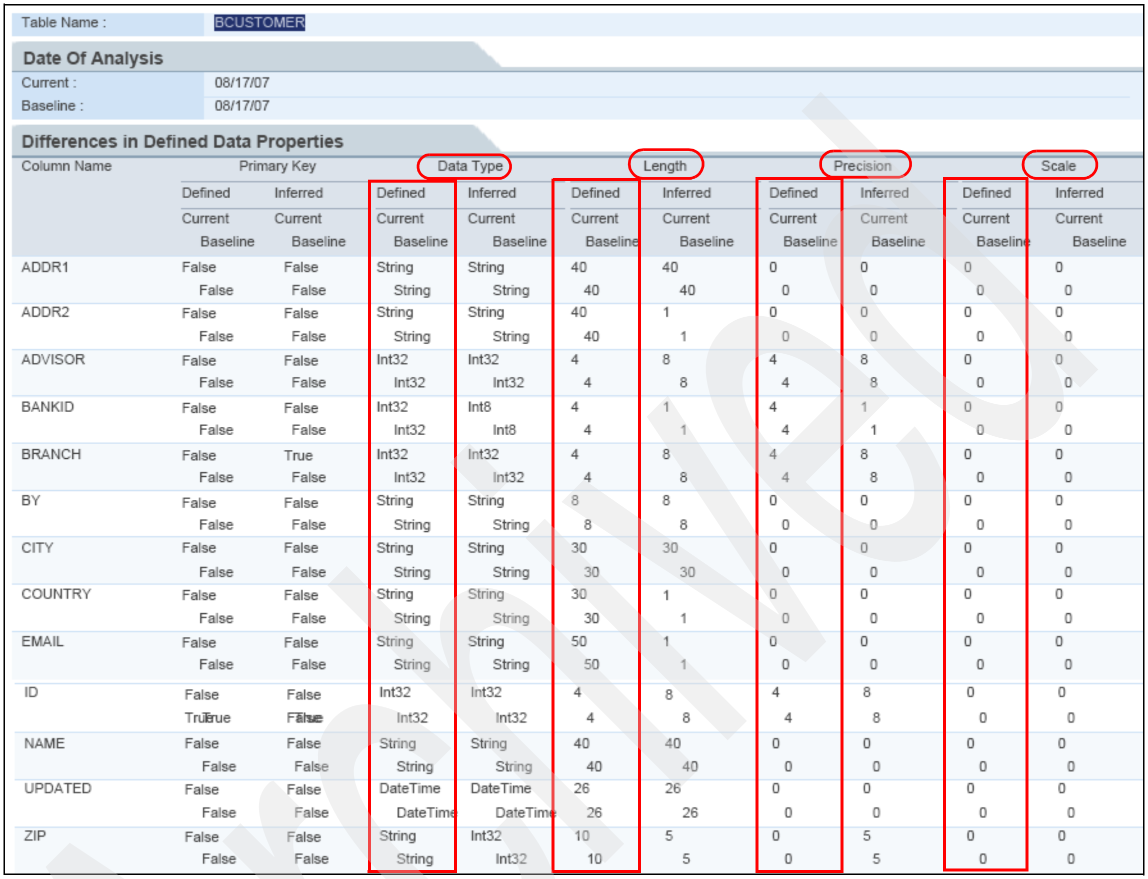


Table Name :	BCUSTOMER

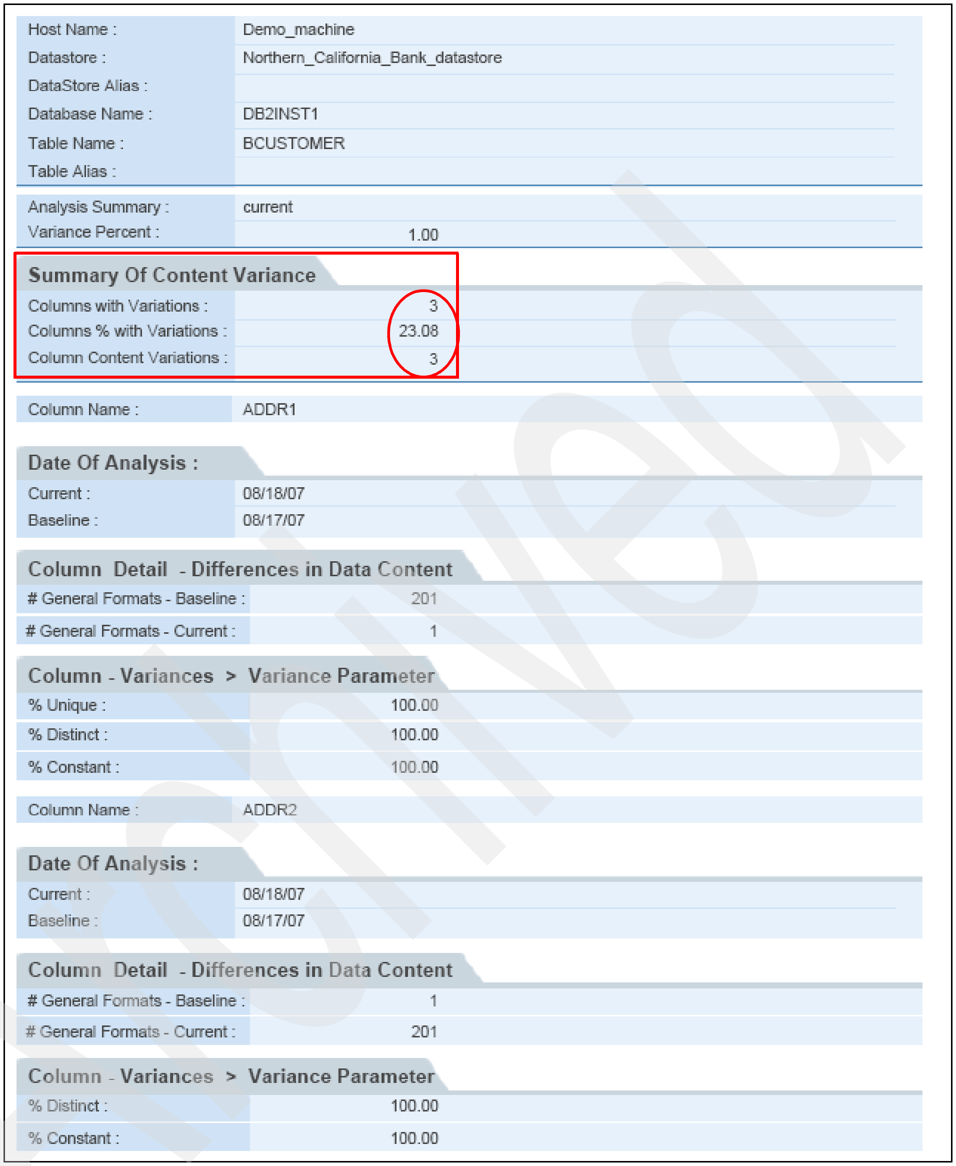
Date Of Analysis

Current :	08/17/07
Baseline :	08/17/07

Differences in Defined Data Properties

Column Name	Primary Key				Data Type				Length				Precision				Scale			
	Defined		Inferred		Defined		Inferred		Defined		Inferred		Defined		Inferred		Defined		Inferred	
	Current	Baseline	Current	Baseline	Current	Baseline	Current	Baseline	Current	Baseline	Current	Baseline	Current	Baseline	Current	Baseline	Current	Baseline	Current	Baseline
ADDR1	False	False	False	False	String	String	String	String	40	40	40	40	0	0	0	0	0	0	0	0
ADDR2	False	False	False	False	String	String	String	String	40	40	1	1	0	0	0	0	0	0	0	0
ADVISOR	False	False	False	False	Int32	Int32	Int32	Int32	4	4	8	8	4	4	8	8	0	0	0	0
BANKID	False	False	False	False	Int32	Int32	Int8	Int8	4	4	1	1	4	4	1	1	0	0	0	0
BRANCH	False	False	True	False	Int32	Int32	Int32	Int32	4	4	8	8	4	4	8	8	0	0	0	0
BY	False	False	False	False	String	String	String	String	8	8	8	8	0	0	0	0	0	0	0	0
CITY	False	False	False	False	String	String	String	String	30	30	30	30	0	0	0	0	0	0	0	0
COUNTRY	False	False	False	False	String	String	String	String	30	30	1	1	0	0	0	0	0	0	0	0
EMAIL	False	False	False	False	String	String	String	String	50	50	1	1	0	0	0	0	0	0	0	0
ID	False	TruTrue	False	FaTrue	Int32	Int32	Int32	Int32	4	4	8	8	4	4	8	8	0	0	0	0
NAME	False	False	False	False	String	String	String	String	40	40	40	40	0	0	0	0	0	0	0	0
UPDATED	False	False	False	False	DateTime	DateTime	DateTime	DateTime	26	26	26	26	0	0	0	0	0	0	0	0
ZIP	False	False	False	False	String	String	Int32	Int32	10	10	5	5	0	0	5	5	0	0	0	0

Figure 2-74 Baseline Structure "Current to Prior Structural Variances" report for the BCUSTOMER table



Host Name :	Demo_machine
Datastore :	Northern_California_Bank_datastore
DataStore Alias :	
Database Name :	DB2INST1
Table Name :	BCUSTOMER
Table Alias :	

Analysis Summary :	current
Variance Percent :	1.00

Summary Of Content Variance

Columns with Variations :	3
Columns % with Variations :	23.08
Column Content Variations :	3

Column Name :	ADDR1

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	201
# General Formats - Current :	1

Column - Variances > Variance Parameter

% Unique :	100.00
% Distinct :	100.00
% Constant :	100.00

Column Name :	ADDR2

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

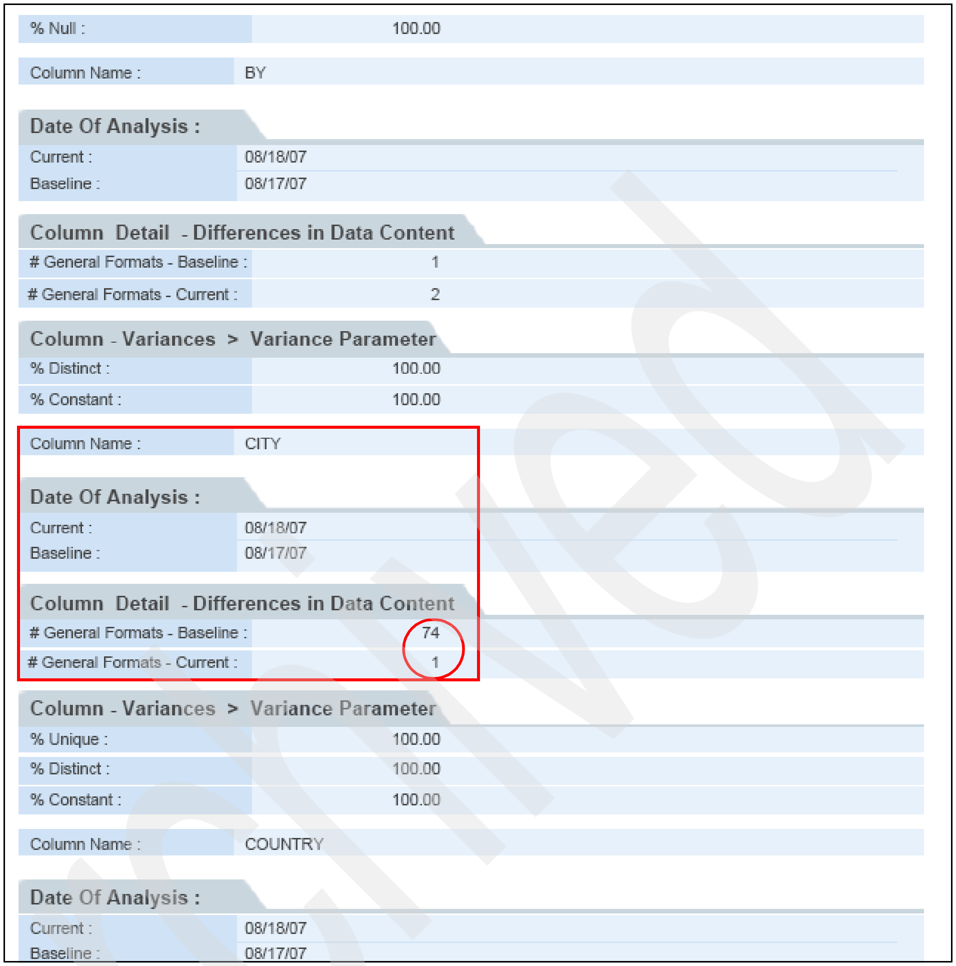
Column Detail - Differences in Data Content

# General Formats - Baseline :	1
# General Formats - Current :	201

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00

Figure 2-75 Baseline Structure "Current to Prior Variances" report for the BCUSTOMER table 1/2



% Null :	100.00

Column Name :	BY

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	1
# General Formats - Current :	2

Column - Variances > Variance Parameter

% Distinct :	100.00
% Constant :	100.00

Column Name :	CITY

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Column Detail - Differences in Data Content

# General Formats - Baseline :	74
# General Formats - Current :	1

Column - Variances > Variance Parameter

% Unique :	100.00
% Distinct :	100.00
% Constant :	100.00

Column Name :	COUNTRY

Date Of Analysis :

Current :	08/18/07
Baseline :	08/17/07

Figure 2-76 Baseline Structure "Current to Prior Variances" report for the BCUSTOMER table 2/2

IBM WebSphere AuditStage business rule compliance

As mentioned earlier, Business Rule Compliance evaluates the quality of data in terms of specific business rules involving multiple data fields within or across records (or rows) that are logically related. IBM WebSphere AuditStage is used to perform this check. In this section, we do not describe the process of building the data filter because that is described in 1.12, "IBM WebSphere AuditStage business rule validation" on page 336 but focus on the completed data filter specification and the results of executing the business rule.

The business rule described in Table 2-20 on page 501 is identical to the one described in “1: Check ADVISOR and USERID of employee matches up” on page 466 and is not repeated here.

2.6.6 Data integration analysis

As mentioned earlier, Northern California Bank’s core and non-core services are to be integrated into the CRM system, along with the core and non-core services of North American Bank.

Because the CRM system has no data content, we consider only the “off-the-shelf” data model definitions. The keys, code fields, indicator fields, and referential integrity relationships (both implicit and explicit) of interest for the data integration were identified as shown in Table 2-24.

Business rules involving data elements in one or more tables are shown in Table 2-25.

Table 2-24 Keys, codes, and indicators in the CRM system

Table	Key column(s)	Code column	Indicator column	RI relationships
CUSTOMERTYPE	ID			
CUSTOMER	ID	► PREFIX ► RATING	► GENDER ► NABLOANINDICATOR ► NCBLOANINDICATOR ► BROKINDICATOR ► CCINDICATOR ► CARINDICATOR ► FULLCOVERIND	► TYPE to the CUSTOMERTYPE table ► PREFCONTACT to the CONTACTTYPE table ► PREFLANG to the ISO_LANGUAGE table
RELATIONTYPE	ID			
CUSTOMERRELATION	FROMCUSTOMER, RELATIONTYPE, TOCUSTOMER			► FROMCUSTOMER to the CUSTOMER table ► TOCUSTOMER to the CUSTOMER table ► RELATIONTYPE to the RELATIONTYPE table
PRODUCT	ID			BUSINESS to the LINEOFBUSINESS table
LINEOFBUSINESS	ID			
ROLE	ID			
ITEM	ID			
CONTRACT	ID	STATUS		PRODUCT to the PRODUCT table
CONTRACTITEM	ID			► CONTRACT to the CONTRACT table ► ITEM to the ITEM table

Table	Key column(s)	Code column	Indicator column	RI relationships
CONTRACTROLE	ID			► CUSTOMER to the CUSTOMER table ► CONTRACT to the CONTRACT table ► ROLE to the ROLE table
BRANCH	ID			
EMPLOYEE	ID			► BRANCH to the BRANCH table ► BUSINESS to the LINEOFBUSINESS table
ISO_LANGUAGE	ID			
CUSTKEYXREF	CRMID			

Table 2-25 Business rules in the CRM system

Serial number	Description
1	RATING column in the CUSTOMER table. The business rule for determining the stars is to be determined after an analysis of the North American Bank and Northern California core and non-core systems.

Figure 2-5 on page 425 describes the general approach for migration or data integration of North American Bank's systems.

Details of the steps as they relate to data integration are described here:

- Step 1: General guidelines for the process are described in 2.6.1, "Assumptions about data integration" on page 483.
- Step 2: Identify differences between the source(s) & target(s) is described here.

  As indicated earlier, the data elements of interest are corresponding keys, codes, and indicators in the source and target systems.

  A manual comparison of the sources (described in 2.5.4, "North American Bank analysis" on page 435, "North American Bank non-core services analysis" on page 484, "Northern California Bank analysis" on page 451, and "Northern California Bank non-core services analysis" on page 500) and target (CRM described in Table 2-24 on page 512 and Table 2-25 on page 513) analyses indicates the differences shown in Table 2-26 on page 514, Table 2-27 on page 521, and Table 2-28 on page 521.

  The action to take to manage these differences are also indicated in Table 2-26 on page 514, Table 2-27 on page 521, and Table 2-28 on page 521. This comparative process could be done in a spreadsheet like Excel, could be annotated/noted in IBM WebSphere Information Analyzer for

Analysis Publication, or might leverage products that specify and map sources to targets. Cross domain analysis (which we did not use in this scenario) could have been utilized to facilitate some of this comparison.

Table 2-26 Summary of differences between source and target and the action to be taken

North American Bank (NAB) & Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
Column in the table	Metadata		Data Content example	Column in the table	Metadata Defined	Data Content example	
	Defined	Inferred					

Best of
— TITLE in CUSTOMER (NAB)
— NAME in DRIVER (NAB)
— NAME in CUSTOMER (NCB)
— NAME in BCUSTOMER (NCB)
	CHAR(3)
	VARCHAR(50)
	CHAR(50)
	VARCHAR(40)
		CHAR(3)
		VARCHAR(28)
		CHAR(22)
		VARCHAR(40)
			MR.
			MR. JOHN DOE
			JON DOW
			DR. JOHN DOE
				PREFIX in the CUSTOMER table	VARCHAR(10)	DR.	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— FIRST_NAME in CUSTOMER (NAB)
— NAME in DRIVER (NAB)
— NAME in CUSTOMER (NCB)
— NAME in BCUSTOMER (NCB)
	VARCHAR(20)
	VARCHAR(50)
	CHAR(50)
	VARCHAR(40)
		VARCHAR(12)
		VARCHAR(28)
		CHAR(22)
		VARCHAR(40)
			MICHAEL
			MR. MIKE HUIS
			MR. MICHAEL HUIS
				FIRSTNAME in the CUSTOMER table	VARCHAR(30)	MICHAEL	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— NAME in DRIVER (NAB)
— NAME in CUSTOMER (NCB)
— NAME in BCUSTOMER (NCB)
	VARCHAR(50)
	CHAR(50)
	VARCHAR(40)
		VARCHAR(28)
		CHAR(22)
		VARCHAR(40)
			MS. CHRISTINE JANE DAY
			CHRISTINE DAY
			MS. CHRISTINE J DAY
				MIDDLENAME in the CUSTOMER table	VARCHAR(30)	JANE	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— LAST_NAME in CUSTOMER (NAB)
— NAME in DRIVER (NAB)
— NAME in CUSTOMER (NCB)
— NAME in BCUSTOMER (NCB)
	VARCHAR(20)
	VARCHAR(50)
	CHAR(50)
	VARCHAR(40)
		VARCHAR(16)
		VARCHAR(28)
		CHAR(22)
		VARCHAR(40)
			PESCO
			JOSEPH PESCO
				LASTNAME in the CUSTOMER table	VARCHAR(30)	PESCO	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

North American Bank (NAB) & Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
Best of				GENDER in the CUSTOMER table	CHAR(1)	M	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— GENDER in CUSTOMER (NAB)	CHAR(1)	CHAR(1)	M				
— GENDER in DRIVER (NAB)	CHAR(1)	CHAR(1)					
— GENDER in CUSTOMER (NCB)	CHAR(1)	CHAR(1)	0				
NATIONALITY in CUSTOMER (NAB)	VARCHAR(20)	VARCHAR(2)	AU	NATIONALITY in the CUSTOMER table	VARCHAR(20)	AU	Direct mapping
TYPE in CUSTOMER (NCB)	CHAR(1)	CHAR(1)	P	TYPE in the CUSTOMER table	INTEGER	1	Map using CUSTOMERTYPE reference table
PREF_LANG in CUSTOMER (NCB)	CHAR(3)	CHAR(3)	ENG	PREFLANG in the CUSTOMER table	CHAR(3)	ENG	Map using ISO_LANGUAGE reference table
ADVISOR in CUSTOMER (NCB)	INT32	INT32	555110	ADVISOR in the CUSTOMER table	INTEGER	555110	Map using EMPLOYEE table
				PREFCONTACT in the CUSTOMER table	INTEGER		
Best of				HOMESTREET in the CUSTOMER table	VARCHAR(30)	63, KALINDA ROAD	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— HOME_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	..63 KALINDA RD.......				
— ADDRESS in DRIVER (NAB)	VARCHAR(50)	VARCHAR(36)	63, KALINDA				
— ADDR1 in CUSTOMER (NCB)	CHAR(50)	CHAR(35)					
— ADDR2 in CUSTOMER (NCB)	CHAR(50)	CHAR(50)					
— ADDR1 in BCUSTOMER (NCB)	VARCHAR(40)	CHAR(40)					
— ADDR2 in BCUSTOMER (NCB)	VARCHAR(40)	CHAR(1)					
Best of				HOMECITY in the CUSTOMER table	VARCHAR(20)	BRENTWOOD	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— HOME_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	..BRENTWOOD.				
— CITY in DRIVER (NAB)	VARCHAR(40)	VARCHAR(7)	BRENTWOOD				
— CITY in CUSTOMER (NCB)	CHAR(30)	CHAR(22)					
— CITY in BCUSTOMER (NCB)	VARCHAR(30)	VARCHAR(30)					

North American Bank (NAB) & Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken

Best of
— HOME_ADDRESS in CONTACT_INFO (NAB)
— STATE in DRIVER (NAB)
— ADDR1 in CUSTOMER (NCB)
— ADDR2 in CUSTOMER (NCB)
— ADDR1 in BCUSTOMER (NCB)
— ADDR2 in BCUSTOMER (NCB)
	VARCHAR(50)
	CHAR(2)
	CHAR(50)
	CHAR(50)
	VARCHAR(40)
	VARCHAR(40)
		VARCHAR(36)
		CHAR(2)
		CHAR(35)
		CHAR(50)
		VARCHAR(40)
		VARCHAR(1)
			..CALIFORNIA... .
			CA
				HOMESTATE in the CUSTOMER table	CHAR(2)	CA	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— HOME_ZIP in CONTACT_INFO (NAB)
— ZIP in DRIVER (NAB)
— ZIP in CUSTOMER (NCB)
— ZIP in BCUSTOMER (NCB)
	CHAR(9)
	CHAR(9)
	CHAR(10)
	CHAR(10)
		CHAR(5)
		INT32 length 5
		INT32 length 5
		INT32 length 5
			95123	HOMEZIP in the CUSTOMER table	VARCHAR(10)	95123-4865	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— COUNTRY in CUSTOMER (NCB)
— COUNTRY in BCUSTOMER (NCB)
	CHAR(30)
	VARCHAR(30)
		CHAR(1)
		VARCHAR(1)
				HOMECOUNTRY in the CUSTOMER table	VARCHAR(20)	U.S.A.	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

WORK_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	..555 BAILEY AVENUE...	WORKSTREET in the CUSTOMER table	VARCHAR(30)	555 BAILEY AVE	Use IBM WebSphere QualityStage to standardize and map/transform this information to the target
WORK_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	..SAN JOSE..	WORKCITY in the CUSTOMER table	VARCHAR(20)	SAN JOSE	Use IBM WebSphere QualityStage to standardize and map/transform this information to the target
WORK_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	..CALIFORNIA...	WORKSTATE in the CUSTOMER table	CHAR(2)	CA	Use IBM WebSphere QualityStage to standardize and map/transform this information to the target
WORK_ZIP in CONTACT_INFO (NAB)	CHAR(9)	INT32 length 5	95123	WORKZIP in the CUSTOMER table	VARCHAR(10)	95123-4865	Mapping with hyphen included
				WORKCOUNTRY in the CUSTOMER table	VARCHAR(20)		

North American Bank (NAB) & Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
Best of — HOME_PHONE in CONTACT_INFO (NAB) — HOMEPHONE in CUSTOMER (NCB)	CHAR(15) CHAR(15)	INT64 length 10) INT64 length 10	4085551234	HOMEPHONE in the CUSTOMER table	VARCHAR(15)	408-5551234	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— WORK_PHONE in CONTACT_INFO (NAB)
— WORKPHONE in CUSTOMER (NCB)
	CHAR(15)
	CHAR(15)
		CHAR(10)
		CHAR(12)
			6505555678	WORKPHONE in the CUSTOMER table	VARCHAR(20)	650-5555678	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— CELL_PHONE in CONTACT_INFO (NAB)
—CELLPHONE in CUSTOMER (NCB)
	CHAR(15)
	CHAR(15)
		CHAR(10)
		CHAR(12)
			4155553456	CELLPHONE in the CUSTOMER table	VARCHAR(15)	4155553456	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— EMAIL in CUSTOMER (NCB)
— EMAIL in BCUSTOMER (NCB)
	VARCHAR(50)
	VARCHAR(50)
		VARCHAR(5)
		VARCHAR(1)
			IADM	EMAIL in the CUSTOMER table	VARCHAR(20)	IADM	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

				RATING in the CUSTOMER table	CHAR(5)		
				NABCHKASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NABSAVASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NABLOANINDICATOR in the CUSTOMER table	CHAR(1)		
INITIAL_VALUE in LOAN (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	400000.00	NABLOANAMOUNT in the CUSTOMER table	DECIMAL(9,2)	400000.00	Direct mapping
BALANCE in LOAN (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	350000.00	NABLOANBALANCE in the CUSTOMER table	DECIMAL(9,2)	350000.00	Direct mapping
RATES in LOAN (NAB)	DECIMAL(8,5)	DECIMAL(7,5)	6.35	NABLOANRATE in the CUSTOMER table	DECIMAL(6,3)	6.35	Mapping with possible truncation. Should consider increasing precision and scale of target

North American Bank (NAB) & Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
				NCBCHKASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NCBSAVASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NCBLOANINDICATOR in the CUSTOMER table	CHAR(1)		
INITIAL_LOAN_VALUE in LOAN (NCB)	CHAR(20)	INT32 length 8	500000.00	NCBLOANAMOUNT in the CUSTOMER table	DECIMAL(9,2)	500000.00	Mapping with data type transformation
BALANCE in LOAN (NCB)	CHAR(20)	INT32 length 9	495000.00	NCBLOANBALANCE in the CUSTOMER table	DECIMAL(9,2)	495000.00	Mapping with data type transformation
INTEREST_RATE in LOAN (NCB)	CHAR(20)	SFLOAT length 4	5.35	NCBLOANRATE in the CUSTOMER table	DECIMAL(6,3)	5.35	Mapping with data type transformation
				BROKINDICATOR in the CUSTOMER table	CHAR(1)		
				BROKASSETS in the CUSTOMER table	DECIMAL(9,2)		
				BROKMARGIN in the CUSTOMER table	DECIMAL(9,2)		
				CCINDICATOR in the CUSTOMER table	CHAR(1)		
LIMIT in CARD (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	21000.00	CCLIMIT in the CUSTOMER table	INTEGER	21000	Mapping with truncation of scale. Consider adding scale to the target
LIMIT_BALANCE in CARD (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	4971.50	CCBALANCE in the CUSTOMER table	DECIMAL(9,2)	4971.50	Direct mapping
LIMIT_W_BALANCE in CARD (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	15029.50				Ignore this fields because it can be computed
				CARINDICATOR in the CUSTOMER table	CHAR(1)		
FULL_COVERAGE_IND in CAR_INSURANCE	CHAR(1)	CHAR(1)	N	FULLCOVERIND in the CUSTOMER table	CHAR(1)	N	Direct mapping
INSURANCE_VALUE in CAR_INSURANCE	DECIMAL(9,2)	DECIMAL(8,2)	1200.00	CARPREMIUMS in the CUSTOMER table	DECIMAL(6,2)	1200.00	Mapping with possible truncation. Consider increasing precision
END_DT in CAR_INSURANCE	DATE	DATE	12/31/2007	CARENDDATE in the CUSTOMER table	DATE	12/31/2007	Direct mapping
				CRMID in the CUSTKEYXREF table	INTEGER		
CUSTOMER_ID in CUSTOMER (NAB)	INT32	INT16 length 4	1344	NABCOREID in the CUSTKEYXREF table	INTEGER		

North American Bank (NAB) & Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
DRIVER_ID in DRIVER (NAB)	INT32	INT16 length 4	338	NABNONCOREID in the CUSTKEYXREF table	INTEGER		Direct mapping. Note that this information can be used to get access to credit cards held by this individual using the CUSTKEYXREF table
ID in CUSTOMER (NCB)	INT32	INT32 length 8	10001500	NCBCOREID in the CUSTKEYXREF table	INTEGER	10001500	Direct mapping
ID in BCUSTOMER (NCB)	INT32	INT32 length 8	20001500	NBCNONCOREID in the CUSTKEYXREF table	INTEGER	20001500	Direct mapping
				ID in the CUSTOMERTYPE table	INTEGER		To be generated
				DESCRIPTION in the CUSTOMERTYPE table	VARCHAR(50)	PERSON	To be generated
				ID in the RELATIONTYPE table	INTEGER		To be generated
				DESCRIPTION in the RELATIONTYPE table	VARCHAR(50)	IS SON OF	To be generated
				ID in the ISO_LANGUAGE table	CHAR(3)		ISO standard code
				DESCRIPTION in the ISO_LANGUAGE table	VARCHAR(50)	DANISH	ISO standard description
				ID in the CONTACTTYPE table	INTEGER		To be generated
				DESCRIPTION in the CONTACTTYPE table	VARCHAR(50)	MAILING ADDRESS	To be generated
				ID in the LINEOFBUSINESS table	INTEGER		To be generated
				DESCRIPTION in the LINEOFBUSINESS table	VARCHAR(50)	BROKERAGE	To be generated
				ID in the ROLE table	INTEGER		To be generated
				DESCRIPTION in the ROLE table	VARCHAR(50)	BENIFICIARY	To be generated
				ID in the ITEM table	INTEGER		To be generated
				DESCRIPTION in the ITEM table	VARCHAR(50)	INTEREST RATE	To be generated
ID in BRANCH (NCB)	INT32	INT8 length 2	12001500	ID in the BRANCH table	INTEGER	12001500	Direct mapping
NAME in BRANCH (NCB)	CHAR(50)	CHAR(28)	SANTA TERESA	NAME in the BRANCH table	VARCHAR(50)	SANTA TERESA	Direct mapping
				FROMCUSTOMER in the CUSTOMERRELATION table	INTEGER		To be generated

North American Bank (NAB) & Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
				TOCUSTOMER in the CUSTOMERRELATION table	INTEGER		To be generated
				RELATIONTYPE in the CUSTOMERRELATION table	INTEGER		To be generated
				ID in the CONTRACT table	INTEGER		To be generated
				PRODUCT in the CONTRACT table	INTEGER		To be generated
				STATUS in the CONTRACT table	INTEGER		To be generated
				CREATED in the CONTRACT table	TIMESTAMP		To be generated
				UPDATED in the CONTRACT table	TIMESTAMP		To be generated
				ID in the CONTRACTITEM table	INTEGER		To be generated
				CONTRACT in the CONTRACTITEM table	INTEGER		To be generated
				ITEM in the CONTRACTITEM table	INTEGER		To be generated
				VALUE in the CONTRACTITEM table	VARCHAR(30)		To be generated
				ID in the CONTRACTROLE table	INTEGER		To be generated
				CUSTOMER in the CONTRACTROLE table	INTEGER		To be generated
				CONTRACT in the CONTRACTROLE table	INTEGER		To be generated
				ROLE in the CONTRACTROLE table	INTEGER		To be generated
ID in EMPLOYEE (NCB)	INT32	INT32 length 8	13001500	ID in the EMPLOYEE table	INTEGER	13001500	Direct mapping
NAME in EMPLOYEE (NCB)	CHAR(50)	CHAR(20)	JEFFREY JONES	NAME in the EMPLOYEE table	CHAR(50)	JEFFREY JONES	Direct mapping
USERID in EMPLOYEE (NCB)	CHAR(8)	CHAR(8)	JJONES	USERID in the EMPLOYEE table	CHAR(8)	JJONES	Direct mapping
BRANCH in EMPLOYEE (NCB)	INT32	INT32 length 8	12001500	BRANCH in the EMPLOYEE table	INTEGER	12001500	Direct mapping
				BUSINESS in the EMPLOYEE table	INTEGER		To be generated

Table 2-27 North American Bank information missing in the CRM data model

Data element	Action to be taken
NICKNAME in CUSTOMER	Ignore because it is not considered relevant to customer relationship management
CHURN_IND in CUSTOMER	Include this information in the CRM data model
CREDIT_SCORE in CUSTOMER	Include this information in the CRM data model

Table 2-28 Northern California Bank information missing in the CRM data model

Data element	Action to be taken
FAX IN CUSTOMER	Include this information in the CRM data model

- Step 3: Determine action in specific cases

  The general guideline of the target system is to accommodate the requirements of the existing systems, unless deemed inappropriate for the CRM. For example, the NICKNAME column is not appropriate for customer relationship management.

- Step 4: Determine strategy and plan to execute action

  As mentioned earlier, the strategy and plan should include the following:

  – Addressing the data integrity violations detected in North American Bank and Northern California Bank's core and non-core services systems' prior to initiating the data integration

    - primary key in Table 2-6 on page 442, Table 2-11 on page 460, Table 2-18 on page 491, and Table 2-22 on page 507
    - foreign key in Table 2-7 on page 443, Table 2-12 on page 461, and Table 2-23 on page 508
    - domains in Table 2-5 on page 439, Table 2-10 on page 457, Table 2-17 on page 488, and Table 2-21 on page 504
    - business rule compliance in Figure 2-18 on page 448, Figure 2-21 on page 449, Figure 2-24 on page 451, Figure 2-45 on page 472, Figure 2-48 on page 474, and Figure 2-64 on page 499

  – Design cleansing, extract, transform, and load procedures to migrate the data. This process would largely involve the use of tools such as IBM WebSphere QualityStage and IBM WebSphere DataStage but can also involve user-written code.

  This plan needs to be tested rigorously using representative data used in the data profiling analyses.

As mentioned earlier, IBM WebSphere Information Analyzer should be used to profile the representative data and ensure its usefulness in handling data quality issues, and for establishing a a source baseline to evaluate outputs. IBM WebSphere Information Analyzer should also be used to test the development outputs and the quality assurance results. This can facilitate a rapid review that otherwise would require hand-coding and might miss key issues in data cleansing and transformation. Use of Cross-domain Analysis can evaluate the overlap of the source and target data that is being transformed. Use of Baseline Analysis can establish output measures to evaluate success of changes and updates to the cleansing and transformation processes.

Designing this strategy and plan is beyond the scope of this book.

- Step 5: Execute the plan

  Prior to executing the strategy and plan designed in the earlier step, you need to ensure that no structural and data content changes have occurred to the data sources in the interval since data profiling analyses was initiated. This is achieved by executing Baseline Analysis on the data sources as described in “Baseline Analysis reports” on page 443, “Baseline Analysis reports” on page 461, “Baseline Analysis reports” on page 492, and “Baseline Analysis reports” on page 508.

  If Baseline Analysis identifies changes that affect the data integration strategy and plan, then the strategy and plan should be revised and re-tested.

  Execute the plan to integrate the core and non-core services of the North American Bank and Northern California Bank.

  This topic is beyond the scope of this book.

- Step 6: Review success of the process

  You should verify that the data integration was successful by performing data profiling analyses on the CRM system after data integration.

  This topic is beyond the scope of this book.

A

IBM Information Server overview

In this chapter we provide an overview of IBM Information Server architecture and processing flow.

The topics that we cover are:

- IBM Information Server architecture
- Configuration flow
- Runtime flow

A.1 Introduction

Over the years, most organizations have made significant investments in enterprise resource planning, customer relationship management, and supply chain management packages in addition to their home grown applications. These packages have resulted in larger amounts of data being captured about their businesses. To turn all this data into consistent, timely, and accurate information for decision-making requires an effective means of integrating information. Statutory compliance requirements, such as Basel II and Sarbanes-Oxley, place additional demands for consistent, complete, and trustworthy information.

IBM Information Server addresses these critical information integration requirements of consistent, complete, and trustworthy information with a comprehensive, unified foundation for enterprise information architectures, capable of scaling to meet any information volume requirement so that companies can deliver business results faster and with higher quality results for all the following initiatives:

- Business intelligence

  IBM Information Server makes it easier to develop a unified view of the business for better decisions. It helps you understand existing data sources, cleanse, correct, and standardize information, and load analytical views that can be reused throughout the enterprise.

- Master data management

  IBM Information Server simplifies the development of authoritative master data by showing where and how information is stored across source systems. It also consolidates disparate data into a single, reliable record, cleanses and standardizes information, removes duplicates, and links records together across systems. This master record can be loaded into operational data stores, data warehouses, or master data applications such as WebSphere Customer Center and WebSphere Product Center. The record can also be assembled, completely or partially, on demand.

- Infrastructure rationalization

  IBM Information Server aids in reducing operating costs by showing relationships between systems and by defining migration rules to consolidate instances or move data from obsolete systems. Data cleansing and matching ensure high-quality data in the new system.

- Business transformation

  IBM Information Server can speed development and increase business agility by providing reusable information services that can be plugged into applications, business processes, and portals. These standards-based information services are maintained centrally by information specialists but are widely accessible throughout the enterprise.

- Risk and compliance

  IBM Information Server helps improve visibility and data governance by enabling complete, authoritative views of information with proof of lineage and quality. These views can be made widely available and reusable as shared services, while the rules inherent in them are maintained centrally.

IBM Information Server combines the technologies of key information integration functions within the IBM Information Integration Solutions portfolio into a single unified platform that enables companies to understand, cleanse, transform, move, and deliver trustworthy and context-rich information as shown in Figure A-1 on page 527.

IBM Information Server includes the following product modules:

- IBM WebSphere DataStage

  It enables organizations to design data flows that extract information from multiple source systems, transform it in ways that make it more valuable, and then deliver it to one or more target databases or applications.

- IBM WebSphere QualityStage

  Designed to help organizations understand and improve the overall quality of their data assets, WebSphere QualityStage provides advanced features to help investigate, repair, consolidate, and validate heterogeneous data within an integration workflow.

- IBM WebSphere Federation Server

  It enables applications to access and integrate diverse data and content sources as though they were a single resource—regardless of where the information resides—while retaining the autonomy and integrity of the heterogeneous data and content sources. This enabling technology is transparent, heterogeneous, and extensible, and provides high function and high performance.

- IBM WebSphere Information Services Director

  IBM Information Server provides a unified mechanism for publishing and managing shared service-oriented architecture (SOA) services across data quality, data transformation, and federation functions, allowing information specialists to easily deploy services for any information integration task and consistently manage them. This enables developers to take data integration logic built using IBM Information Server and publish it as an *always on* service—in minutes. The common services also include the metadata services, which provide standard service-oriented access and analysis of metadata across the platform.

- IBM WebSphere Information Analyzer

  IBM WebSphere Information Analyzer profiles and analyzes data so that you can deliver trusted information to your users. It can automatically scan samples of your data to determine their quality and structure. This analysis aids you in understanding the inputs to your integration process, ranging from individual fields to high-level data entities. Information analysis also enables you to correct problems with structure or validity before they affect your project. While analysis of source data is a critical first step in any integration project, you must continually monitor the quality of the data. IBM WebSphere Information Analyzer enables you to treat profiling and analysis as an ongoing process and create business metrics that you can run and track over time.

- IBM WebSphere Business Glossary

  IBM Information Server provides a Web-based tool that enables business analysts and subject-matter experts to create, manage, and share a common enterprise vocabulary and classification system. WebSphere Business Glossary functionality is powered by and actively connected to WebSphere Metadata Server. This enables users to link business terms to more technical artifacts managed by WebSphere Metadata Server. The Metadata Server also enables sharing of the business terms by IBM Rational® Data Architect and WebSphere Information Analyzer, creating a common set of semantic tags for reuse by data modelers, data analysts, business analysts, and users.

- IBM WebSphere Metadata Server

  IBM Information Server provides the next-generation metadata repository that is fully integrated and common across all product modules, including WebSphere Information Analyzer, WebSphere QualityStage, WebSphere DataStage, and WebSphere Business Glossary. The metadata services infrastructure of IBM Information Server is designed to allow metadata to be more easily managed, accessed by those who need it, and shared across heterogeneous technologies through an SOA.

- IBM WebSphere DataStage MVS™ Edition

  IBM Information Server brings data transformation capabilities to the mainframe with its WebSphere DataStage MVS Edition product module. WebSphere DataStage MVS Edition consolidates, collects, and centralizes information from various systems and mainframes using native execution, from a single design environment.

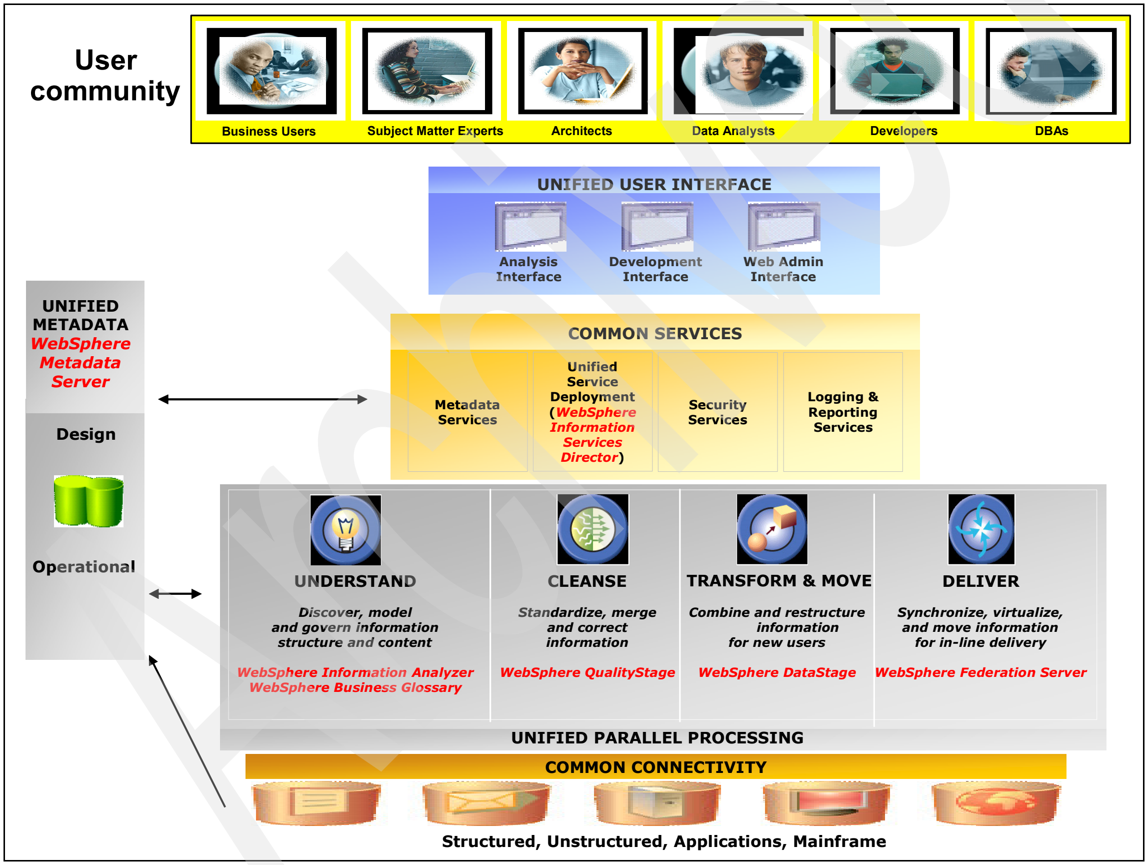
Note: For complete details on these product modules, refer to the documentation at the following Web site:

http://www.ibm.com/software/data/integration/info_server/

A number of companion products support IBM Information Server, such as Rational Data Architect and WebSphere Replication Server Event Publisher.

A.2 IBM Information Server architecture

IBM Information Server provides a unified architecture that works with all types of information integration as shown in Figure A-1. A unified user interface, common services, key integration functions (understand, cleanse, transform, and move, and deliver), unified parallel processing, and unified metadata are at the core of the architecture.



User community
Business Users
Subject Matter Experts
Architects
Data Analysts
Developers
DBAs
UNIFIED USER INTERFACE
Analysis Interface
Development Interface
Web Admin Interface
UNIFIED METADATA
WebSphere Metadata Server
Design
Operational
COMMON SERVICES
Metadata Services
Unified Service Deployment (WebSphere Information Services Director)
Security Services
Logging & Reporting Services
UNDERSTAND
Discover, model and govern information structure and content
WebSphere Information Analyzer
WebSphere Business Glossary
CLEANSE
Standardize, merge and correct information
WebSphere QualityStage
TRANSFORM & MOVE
Combine and restructure information for new users
WebSphere DataStage
DELIVER
Synchronize, virtualize, and move information for in-line delivery
WebSphere Federation Server
UNIFIED PARALLEL PROCESSING
COMMON CONNECTIVITY
Structured, Unstructured, Applications, Mainframe

Figure A-1 IBM Information Server architecture

The architecture is service oriented, enabling IBM Information Server to work within an organization's evolving enterprise service-oriented architectures. An SOA also connects the individual product modules of IBM Information Server. By eliminating duplication of functions, the architecture efficiently uses resources and reduces the amount of development and administrative effort that are required to deploy an integration solution.

We describe each of the following components of the architecture briefly in the following sections:

- Unified user interface
- Common services
- Key integration functions (understand, cleanse, transform and move, deliver)
- Unified parallel processing
- Unified metadata
- Common connectivity

A.2.1 Unified user interface

The unified user interface enables an organization's entire user community of business users, subject matter experts, architects, data analysts, developers, and database administrators (DBAs) to collaborate, administer and query information within the enterprise. A security infrastructure ensures that users are permitted to access information and perform tasks for which they are authorized.

The face of IBM Information Server is a common graphical interface and tool framework. Shared interfaces such as the IBM Information Server console and Web console provide a common look and feel, visual controls, and user experience across products. Common functions such as catalog browsing, metadata import, query, and data browsing all expose underlying common services in a uniform way. IBM Information Center provides rich client interfaces for highly detailed development work and thin clients that run in Web browsers for administration. Application programming interfaces (APIs) support a variety of interface styles that include standard request-reply, service-oriented, event-driven, and scheduled task invocation.

Figure A-1 on page 527 shows the three broad user interface categories are the analysis interface, development interface, and Web Admin interface.

A.2.2 Common services

IBM Information Server is built entirely on a set of shared services that centralize core tasks across the platform. These include administrative tasks such as unified service deployment, security, user administration, logging, and reporting. The common services provides flexible, configurable interconnections among the many parts of the architecture.

Shared services allow these tasks to be managed and controlled in one place, regardless of which product module is being used. The common services also include the metadata services, which provide standard service-oriented access and analysis of metadata across the platform. In addition, the common services layer manages how services are deployed from any of the product functions, allowing cleansing and transformation rules or federated queries to be published as shared services within an SOA, using a consistent and easy-to-use mechanism.

The common services layer is deployed on J2EE™-compliant application servers such as IBM WebSphere Application Server.

Attention: Today, common services are consumed exclusively by the various components of IBM Information Server. These common services are currently *not* exposed as public SOA services and, therefore, cannot be invoked by applications or tools.

IBM Information Server products can access four general categories of service, such as design, execution, metadata, and unified service deployment, which we describe in the following sections.

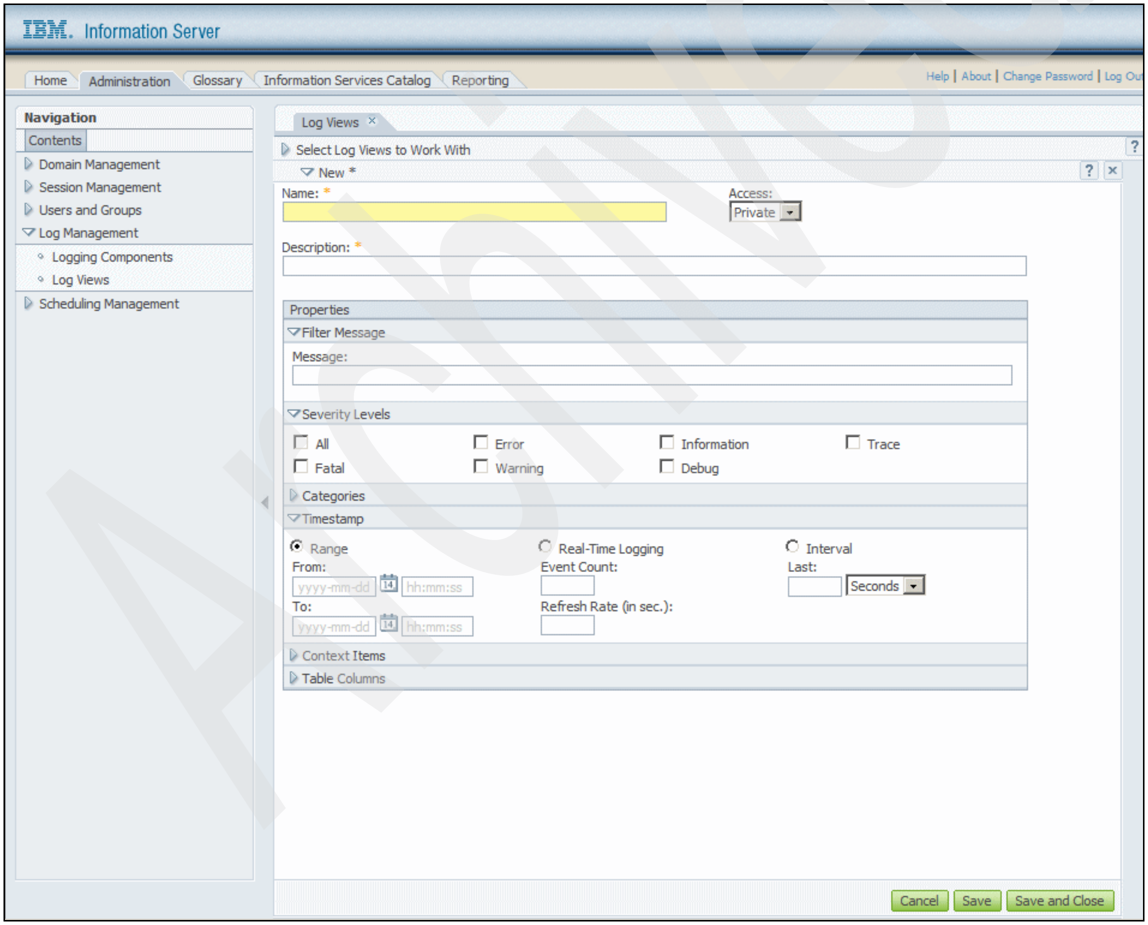
Design services

Design services help developers create function-specific services that can also be shared. For example, WebSphere Information Analyzer calls a column analyzer service that was created for enterprise data analysis but can be integrated with other parts of IBM Information Server because it exhibits common SOA characteristics.

Execution services

Execution services include logging, scheduling, monitoring, reporting, security, and Web framework. We define those further here:

- *Log services* help you manage logs across all of the IBM Information Server suite components. The Web console shown in Figure A-2 on page 530 provides a central place to view logs and resolve problems. Logs are stored in the common repository, and each IBM Information Server suite component defines relevant logging categories. You can configure which categories of logging messages are saved in the repository. Log views are saved queries that an administrator can create to help with common tasks. For example, you might want to display all of the errors in DataStage jobs that ran in the past 24 hours. Logging is organized by server components. The Web console displays default and active configurations for each component.



Name: *

Access: Private

Description: *

Message:

□ All □ Error □ Information □ Trace
□ Fatal □ Warning □ Debug

● Range ○ Real-Time Logging ○ Interval

From: yyyy-mm-dd hh:mm:ss

To: yyyy-mm-dd hh:mm:ss

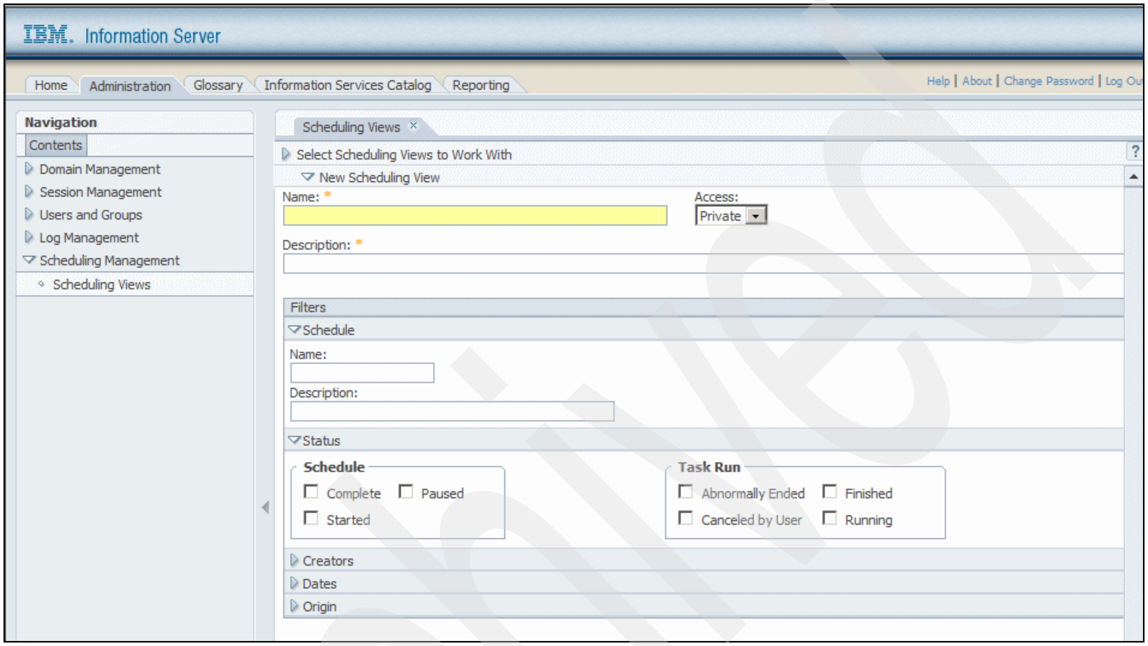
Event Count:

Refresh Rate (in sec.):

Last: Seconds

Figure A-2 Web console for setting up logs

- *Scheduling services* help plan and track activities such as logging and reporting, and suite component tasks such data monitoring and trending. Schedules are maintained using the IBM Information Server console shown in Figure A-3 on page 531, which helps you to define schedules; to view their status, history, and forecast; and to purge them from the system.



Name: *

Access: Private

Description: *

Schedule

Name:

Description:

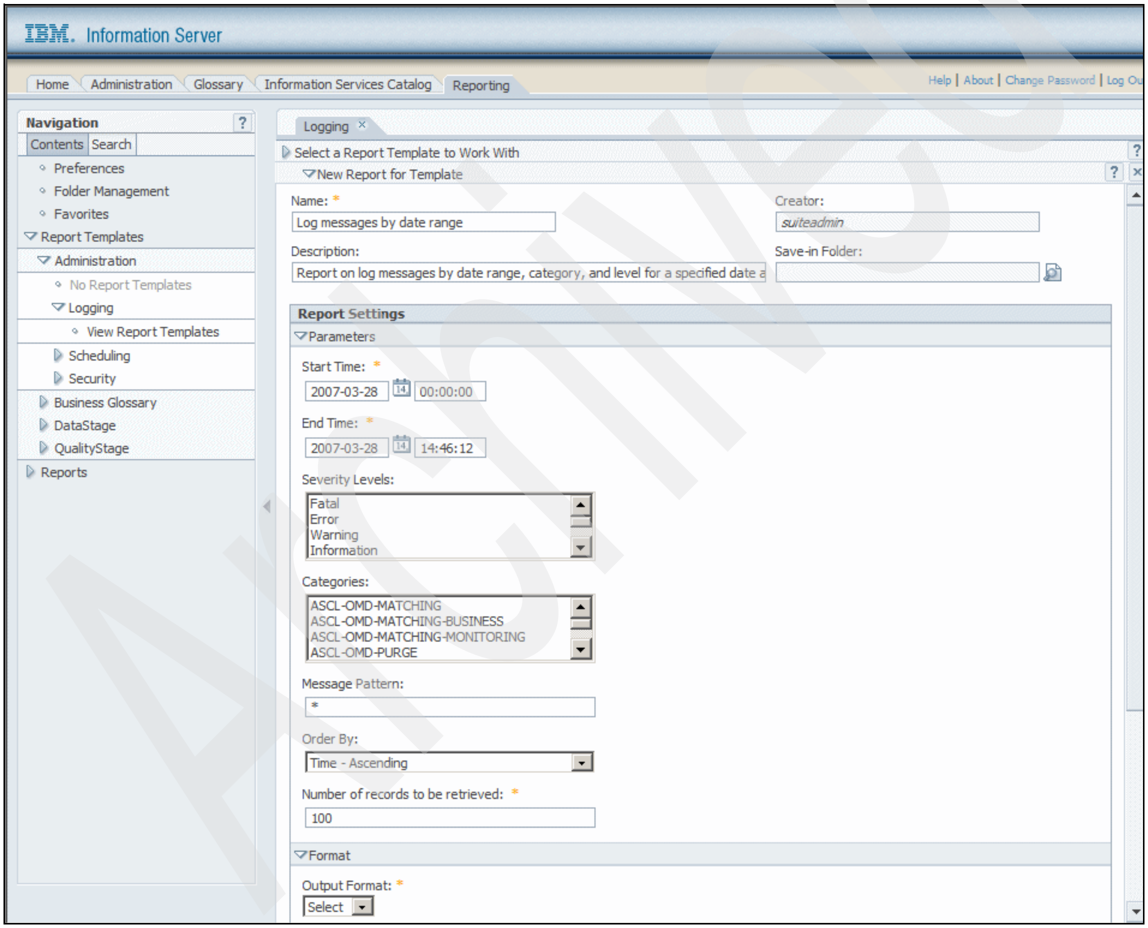
Status

Schedule: □ Complete □ Paused □ Started

Task Run: □ Abnormally Ended □ Finished □ Canceled by User □ Running

Figure A-3 Web console scheduling view creation

- *Reporting services* manage run time and administrative aspects of reporting for IBM Information Server. You can create product-specific reports for WebSphere DataStage, WebSphere QualityStage, and WebSphere Information Analyzer, and cross-product reports for logging, monitoring, scheduling, and security services. All reporting tasks are set up and run from a single interface—the IBM Information Server Web console. You can retrieve and view reports and schedule reports to run at a specific time and frequency. You define reports by choosing from a set of predefined parameters and templates as shown in Figure A-4. You can specify a history policy that determines how the report will be archived and when it expires. Reports can be formatted as HTML, PDF, or Microsoft Word documents.



Name: * Log messages by date range

Creator: suiteadmin

Description: Report on log messages by date range, category, and level for a specified date a

Save-in Folder:

Start Time: * 2007-03-28 00:00:00

End Time: * 2007-03-28 14:46:12

Severity Levels: Fatal, Error, Warning, Information

Categories: ASCL-OMD-MATCHING, ASCL-OMD-MATCHING-BUSINESS, ASCL-OMD-MATCHING-MONITORING, ASCL-OMD-PURGE

Message Pattern: *

Order By: Time - Ascending

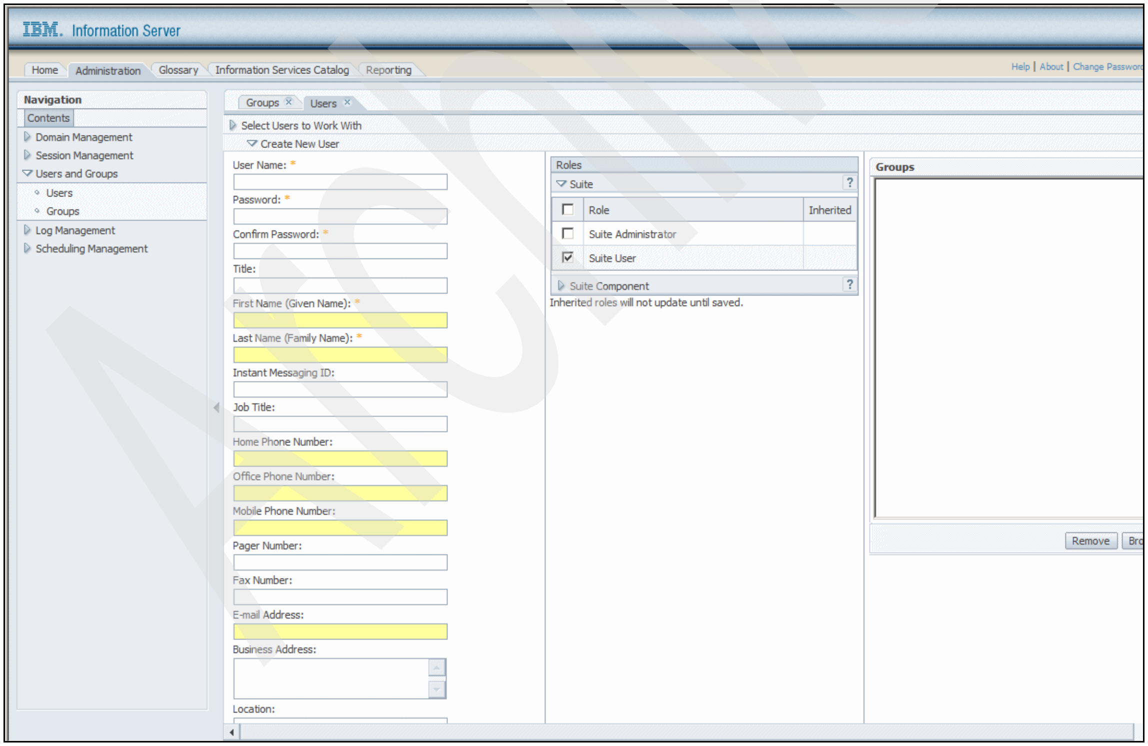
Number of records to be retrieved: * 100

Output Format: * Select

Figure A-4 Web console logging report creation

- *Security services* support role-based authentication of users, access-control services, and encryption that complies with many privacy and security regulations. The Web console shown in Figure A-5 helps administrators add users, groups, and roles and lets administrators browse, create, delete, and update operations within Information Server. Directory services act as a central authority that can authenticate resources and manage identities and relationships among identities. You can base directories on IBM Information Server's own internal directory or on external directories that are based on LDAP, Microsoft's Active Directory®, or UNIX. Users only use one credential to access all the components of Information Server. A set of credentials is stored for each user to provide single sign-on to the products registered with the domain.

> **Note:** A white paper is currently being developed to provide you with guidelines on implementing authentication, access control and encryption security within an IBM Information Server environment. A reference to this white paper will be made available when it becomes publicly available.



☐	Role	Inherited
☐	Suite Administrator	
☑	Suite User	

Inherited roles will not update until saved.

Figure A-5 Web console to administer users and groups

Metadata services

Metadata services enable metadata to be shared “live” across tools so that changes made in one IBM Information Server product are instantly visible across all of the product modules. Metadata services are tightly integrated with the common repository and are packaged in WebSphere Metadata Server. You can also exchange metadata with external tools by using metadata services.

The major metadata services components of IBM Information Server are WebSphere Business Glossary, WebSphere Metadata Server, and WebSphere MetaBrokers and bridges:

- *WebSphere Business Glossary* is a Web-based application that provides a business-oriented view into the data integration environment. Using WebSphere Business Glossary, you can view and update business descriptions and access technical metadata. Metadata is best managed by business analysts who understand the meaning and importance of the information assets to the business. Designed for collaborative authoring, WebSphere Business Glossary gives users the ability to share insights and experiences about data. It provides users with the following information about data resources:

  - Business meaning and descriptions of data
  - Stewardship of data and processes
  - Standard business hierarchies
  - Approved terms

  WebSphere Business Glossary is organized and searchable according to the semantics that are defined by a controlled vocabulary, which you can create by using the Web console.

- *WebSphere Metadata Server* provides a variety of services to other components of IBM Information Server:

  - Metadata access
  - Metadata integration
  - Metadata import and export
  - Impact analysis
  - Search and query

  WebSphere Metadata Server provides a common repository with facilities that are capable of sourcing, sharing, storing, and reconciling a comprehensive spectrum of metadata including business metadata and technical metadata as follows:

  - Business metadata provides business context for information technology assets and adds business meaning to the artifacts that are created and managed by other IT applications. Business metadata includes controlled vocabularies, taxonomies, stewardship, examples, and business definitions.

  - Technical metadata provides details about source and target systems, their table and field structures, attributes, derivations, and dependencies. Technical metadata also includes details about profiling, quality, and ETL processes, projects, and users.

- *WebSphere MetaBrokers* and *bridges* provide semantic model mapping technology that allows metadata to be shared among applications for all products that are used in the data integration life cycle:
  - Data modeling or case tools
  - Business intelligence applications
  - Data marts and data warehouses
  - Enterprise applications
  - Data integration tools

You can use these components to establish common data definitions across business and IT functions.

- Drive consistency throughout the data integration life cycle
- Deliver business-oriented and IT-oriented reporting
- Provide enterprise visibility for change management
- Easily extend to new, existing, and home grown metadata sources

Unified service deployment

IBM Information Server provides an SOA infrastructure that exposes data transformation processes[1], federated queries[2], and database stored procedures as a set of shared services and operations. This is performed by using a consistent and intuitive graphical interface, and managed after publication using the same user interface.

IBM Information Server provides standard service-oriented interfaces for enterprise data integration. The built-in integration logic of IBM Information Server can easily be encapsulated as service objects that are embedded in user applications. These service objects have the following characteristics:

- Always on

  By definition, the services are always running and waiting for requests. This ability removes the overhead of batch startup and shutdown and enables services to respond instantaneously to requests.

- Scalable

  The services distribute request processing and stop and start jobs across multiple WebSphere DataStage servers, enabling high performance with large, unpredictable volumes of requests.

[1] Created from new or existing WebSphere DataStage or WebSphere QualityStage jobs

[2] Created by WebSphere Federation Server

- Standards-based

  The services are based on open standards and can easily be invoked by standards-based technologies including Web Services Description Language (WSDL), enterprise application integration (EAI), and enterprise service bus (ESB) platforms, applications, and portals.

- Manageable

  Monitoring services coordinate timely reporting of system performance data.

- Flexible

  You can invoke the services by using multiple mechanisms (bindings) and choose from many options for using the services.

- Reliable and highly available

  If any WebSphere DataStage server becomes unavailable, it routes service requests to a different server in the pool.

- Reusable

  The services publish their own metadata, enabling them to be found and called across any network.

- High performance

  Load balancing and the underlying parallel processing capabilities of IBM Information Server create high performance for any type of data payload.

A data integration service is created by designing the data integration process logic in IBM Information Server and publishing it as a service. These services can then be accessed by external projects and technologies.

WebSphere Information Services Director provides a foundation for information services by allowing you to leverage the other components of IBM Information Server for understanding, cleansing, and transforming information and deploying those integration tasks as consistent and reusable information services.

WebSphere Information Services Director provides an integrated environment for designing services that enables you to rapidly deploy integration logic as services without assuming extensive development skills. With a simple, wizard-driven interface, in a few minutes you can attach a specific binding and deploy a reusable integration service. WebSphere Information Services Director also provides these features:

- Administrator services for cataloging and registering services.
- Shared reporting and security services.
- A metadata services layer that promotes reuse of the information services by actually defining what the service does and what information it delivers.

A.2.3 Key integration functions

We describe the key integration functions shown in Figure A-1 on page 527 briefly here:

- Understand your data

  IBM Information Server helps you to discover, define, and model information content and structure automatically and to understand and analyze the meaning, relationships, and lineage of information. By automating data profiling and data-quality auditing within systems, organizations can achieve the following goals:

  - Understand data sources and relationships
  - Eliminate the risk of using or proliferating bad data
  - Improve productivity through automation
  - Take advantage of existing IT investments

  IBM Information Server makes it easier for businesses to collaborate across roles. Data analysts can use analysis and reporting functionality, generating integration specifications and business rules that they can monitor over time. Subject matter experts can use Web-based tools to define, annotate, and report on fields of business data. A common metadata foundation makes it easier for different types of users to create and manage metadata by using tools that are optimized for their roles.

  The upcoming WebSphere Information Analyzer product module will provide this functionality.

- Cleanse your information

  IBM Information Server supports information quality and consistency by standardizing, validating, matching, and merging data. It can certify and enrich common data elements, use trusted data such as postal records for name and address information, and match records across or within data sources. IBM Information Server allows a single record to survive from the best information across sources for each unique entity, helping you to create a single, comprehensive, and accurate view of information across source systems.

  The WebSphere QualityStage product module currently provides this functionality.

- Transform your data into information and move

  IBM Information Server transforms and enriches information to ensure that it is in the proper context for new uses. Hundreds of prebuilt transformation functions combine, restructure, and aggregate information.

  Transformation functionality is broad and flexible to meet the requirements of varied integration scenarios. For example, IBM Information Server provides inline validation and transformation of complex data types such as the U.S. Health Insurance Portability and Accountability Act (HIPAA), along with high-speed joins and sorts of heterogeneous data. IBM Information Server also provides high-volume, complex data transformation and movement functionality that can be used for stand-alone extract/transform/load (ETL) scenarios or as a real-time data processing engine for applications or processes.

The WebSphere DataStage product modules currently provide this functionality.

- Deliver your information

  IBM Information Server provides the ability to virtualize, synchronize, or move information to the people, processes, or applications that need it. Information can be delivered through federation or time-based or event-based processing, moved in large bulk volumes from location to location, or accessed in place when it cannot be consolidated. IBM Information Server provides direct, native access to a wide variety of information sources, both mainframe and distributed. It provides access to databases, files, services, and packaged applications and to content repositories and collaboration systems. Companion products allow high-speed replication, synchronization, and distribution across databases, change data capture, and event-based publishing of information.

  The WebSphere Federation Server product module currently provides this functionality.

A.2.4 Unified parallel processing

Much of the work that IBM Information Server does takes place within the parallel processing engine. The engine handles data processing needs as diverse as performing analysis of large databases for WebSphere Information Analyzer, data cleansing for WebSphere QualityStage, and complex transformations for WebSphere DataStage. This parallel processing engine is designed to deliver:

- Parallelism and pipelining to complete increasing volumes of work in decreasing time windows.

  – Data partitioning is an approach to parallelism that involves breaking the record set into partitions, or subsets of records. Data partitioning generally provides linear increases in application performance.

    IBM Information Server partitions data automatically based on the type of partition that the stage requires. In a well-designed, scalable architecture, the developer does not need to be concerned about the number of partitions that will run, the ability to increase the number of partitions, or re-partitioning data.

  - Data pipelining is the process of pulling records from the source system and moving them through the sequence of processing functions that are defined in the data-flow (the job). Because records are flowing through the pipeline, they can be processed without writing the records to disk.
- Scalability by adding hardware (for example, processors or nodes in a grid) with no changes to the data integration design.
- Optimized database, file, and queue processing to handle large files that cannot fit in memory all at once or with large numbers of small files.

Note: The dynamic parallelization of all potential service implementations is an objective of this architecture.

A.2.5 Unified metadata

IBM Information Server is built on a unified metadata infrastructure that enables shared understanding between business and technical domains. This infrastructure reduces development time and provides a persistent record that can improve confidence in information.

All functions of IBM Information Server share the same metamodel, making it easier for different roles and functions to collaborate. A common metadata repository provides persistent storage for all IBM Information Server product modules. All of the products depend on the repository to navigate, query, and update metadata.

The repository contains two kinds of metadata:

- Dynamic metadata that includes design-time information.
- Operational metadata that includes performance monitoring, audit and log data, and data profiling sample data.

Because the repository is shared by all product modules, profiling information that is created by WebSphere Information Analyzer is available instantly to users of WebSphere DataStage and QualityStage, for example. The repository is a J2EE application that uses a standard relational database such as IBM DB2, Oracle, or SQL Server™ for persistence (DB2 is provided with IBM Information Server). These databases provide backup, administration, scalability, parallel access, transactions, and concurrent access.

A.2.6 Common connectivity

IBM Information Server connects to information sources whether they are structured, unstructured, on the mainframe, or applications.

Metadata-driven connectivity is shared across the product modules, and connection objects are reusable across functions. Connectors provide design-time importing of metadata, data browsing and sampling, runtime dynamic metadata access, error handling, and high functionality and high performance runtime data access.

Prebuilt interfaces for packaged applications, called *Packs*, provide adapters to SAP, Siebel, Oracle, and others, enabling integration with enterprise applications and associated reporting and analytical systems.

A.2.7 Client application access to services

After an information service is enabled by IBM Information Server, any enterprise application, .Net or Java developer, Microsoft Office, or integration software, can invoke the service by using a binding protocol such as SOAP over HTTP or EJB™.

Figure A-6 shows how IBM Information Server information services participate in the SOA Reference Architecture. Briefly:

- An information service (blue dots) can access content systems and data systems, while other (non IBM Information Server) services (pink dots) can access applications and registry services. Applications will most likely access data or content using *proprietary* APIs.
- Information services can be invoked by other (non IBM Information Server) services.
- Business processes can invoke information services and other (non IBM Information Server) services.
- Service consumers can invoke information services, business processes, or other (non IBM Information Server) services directly or indirectly.

The Enterprise Service Bus (ESB) layer enables the integration of services through the introduction of a reliable set of capabilities such as intelligent routing, protocol mediation, and other transformation mechanisms. An ESB provides a location independent mechanism for integration.

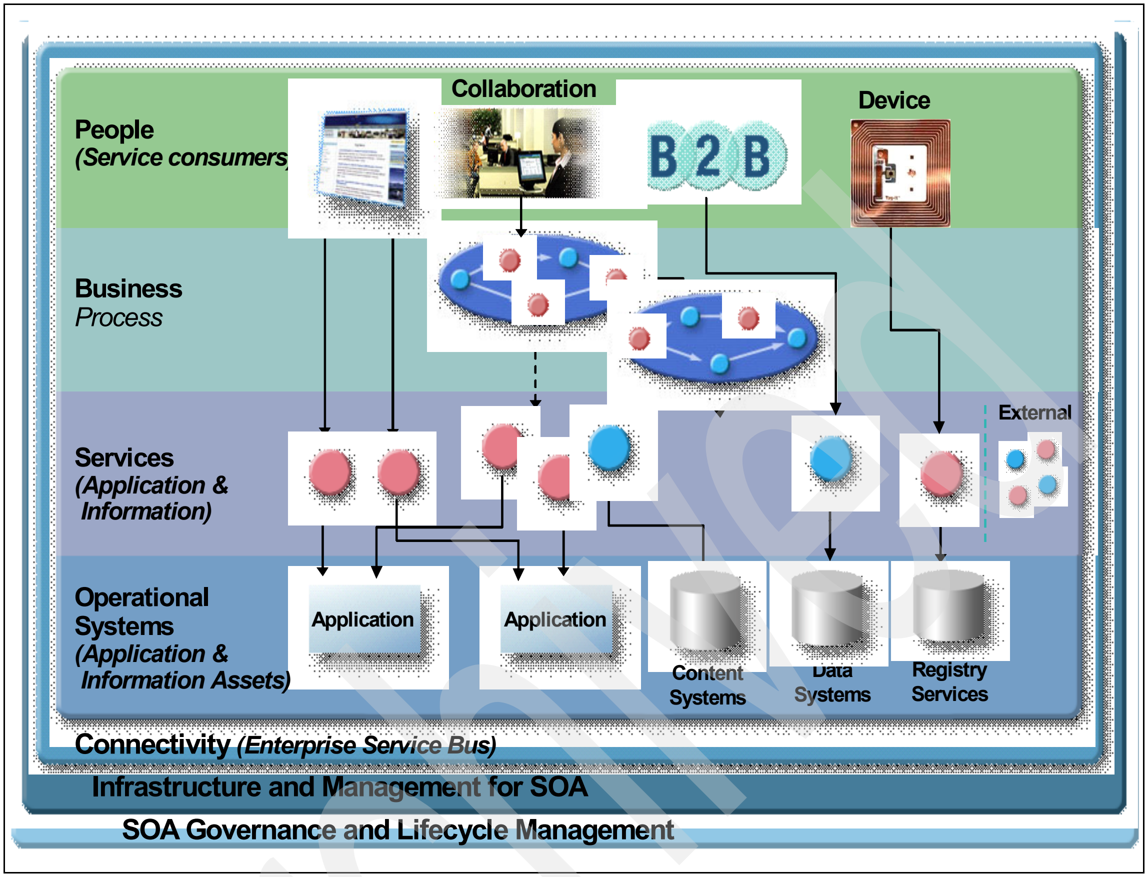


Figure A-6 Information Services in the SOA Reference Architecture

A service-ready data integration job accepts requests from client applications, mapping request data to:

- Input rows and pass them to the underlying jobs in the case of DataStage and QualityStage jobs.
- Input parameters that are executed against a federated database in the case of a federated query or stored procedure.

A job instance can include database access (federated queries), transformations (DataStage jobs), data standardization and matching (QualityStage jobs), and other data integration tasks (database stored procedures) that are supplied by IBM Information Server.

The design of a real-time job determines whether it is always running or runs once to completion. All jobs that are exposed as services process requests on a 24-hour basis.

The SOA infrastructure supports three job topologies for different load and work style requirements. This relates specifically to DataStage and QualityStage jobs, and not to federated queries or stored procedures:

- Batch jobs

  This topology uses new or existing batch jobs that are exposed as services. A batch job starts on demand. Each service request starts one instance of the job that runs to completion. This job typically initiates a batch process from a real-time process that does not need direct feedback on the results. This topology is tailored for processing bulk data sets and is capable of accepting job parameters as input arguments.

- Batch jobs with a Service Output stage

  This topology uses an existing batch job and adds an output stage. The Service Output stage is the exit point from the job, returning one or more rows to the client application as a service response. These jobs typically initiate a batch process from a real-time process that requires feedback or data from the results. This topology is designed to process large data sets and can accept job parameters as input arguments.

- Jobs with a Service Input stage and Service Output stage

  This topology uses both a Service Input stage and a Service Output stage. The Service Input stage is the entry point to a job, accepting one or more rows during a service request. These jobs are always running. This topology is typically used to process high volumes of smaller transactions where response time is important. It is tailored to process many small requests rather than a few large requests.

Current restriction: Client applications can only access IBM Information Server services in synchronous mode, where the model requires feedback to be received for any request made before the client application can proceed to its next course of action. IBM Information Server services that are long running tasks (batch jobs and batch jobs with service output stage topologies) or tasks where no feedback is returned to the requestor (batch jobs topology) need special handling if the client application is to avoid waiting. Such topology jobs must be redesigned to return feedback to the requestor as soon as the request has been processed, and the client application will have to be designed to check on the status of the job at some later point in time. An upcoming release will provide asynchronous support with JMS binding.

A.3 Configuration flow

This section describes the processing flow involved when configuring a service.

Multiple steps are involved in configuring a service before it can become the target of a client application invocation.

There is a hierarchy of containers when defining an information service: **Project** → **Application** → **Service** → **Operation**. This is reflected in the main steps in creating an SOA service using IBM Information Server shown in Figure A-7.

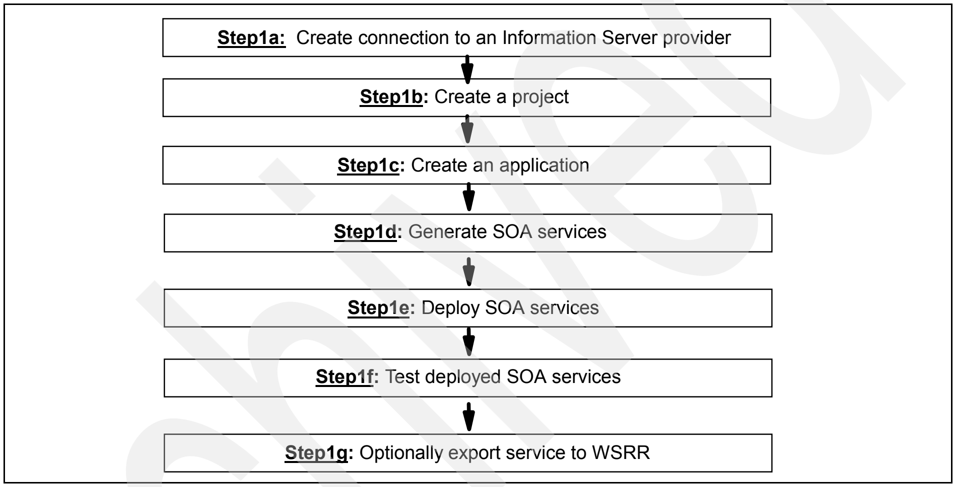

Step1a: Create connection to an Information Server provider
Step1b: Create a project
Step1c: Create an application
Step1d: Generate SOA services
Step1e: Deploy SOA services
Step1f: Test deployed SOA services
Step1g: Optionally export service to WSRR

Figure A-7 Steps in creating SOA services

We describe these steps in more detail here.

A.3.1 Step1a: Create connection to an Information Server provider

An *information provider* is both the server that contains functions that you can expose as services and the functions themselves, such as WebSphere DataStage and WebSphere QualityStage jobs, database stored procedures, or federated SQL queries.

Before an SOA service can be generated for a function, the information provider must be enabled using WebSphere Information Services Director.

There are two types of Information Server providers:

- A *DataStage and QualityStage* type for DataStage and QualityStage jobs
- A *DB2 or Federation Server* type for database stored procedures and federated queries

A.3.2 Step1b: Create a project

A *project* is a collaborative environment that you use to design applications, services, and operations. All project information that you create is saved in the common metadata repository so that it can easily be shared among other IBM Information Server components. You can export a project to back up your work or share work with other IBM Information Server users. The export file includes applications, services, operations, and binding information.

Therefore, you must create a project first.

A.3.3 Step1c: Create an application

An *application* is a container for a set of services and operations. An application contains one or more services that you want to deploy together as an Enterprise Archive (EAR) file on an application server.

All design-time activity occurs in the context of applications:

- Creating services and operations
- Describing how message payloads and transport protocols are used to expose a service
- Attaching a reference provider, such as a WebSphere DataStage job or an SQL query, to an operation

You can also export services from an application before it is deployed and import the services into another application.

Therefore, you must create an application in the project created.

A.3.4 Step1d: Generate SOA services

An *information service* exposes results from processing by information providers such as DataStage servers and federated servers. A deployed service runs on an application server and processes requests from service client applications.

An information service is a collection of operations that are selected from jobs, federated queries, or other information providers. You can group operations in the same information service or design them in separate services. You create an information service for a set of operations that you want to deploy together. You also specify the bindings (SOAP over HTTP or EJB) for the service.

As mentioned earlier, an information service is associated with a particular application.

A.3.5 Step1e: Deploy SOA services

After the information service is generated, it must be *deployed*. You deploy an application on WebSphere Application Server to enable the information services that are contained in the application to receive service requests. You can exclude one or more services, bindings, and operations from the deployment, change runtime properties such as minimum number of job instances, or, for WebSphere DataStage jobs, set constant values for job parameters. WebSphere Information Services Director deploys the Enterprise Archive (EAR) file on the application server.

A.3.6 Step1f: Test deployed SOA services

After deployment, we strongly recommend that you *test* the deployed information service before making it available to client applications.

A.3.7 Step1g: Optionally export service to WebSphere Service Registry and Repository

Note: WebSphere Service Registry and Repository is not a prerequisite for IBM Information Server nor is it mandatory for SOA. If your organization has implemented a WebSphere Server Registry and Repository, you can choose to export the IBM Information Server service you generated to it.

The WebSphere Service Registry and Repository is a separate entity from IBM Information Server that can serve as the master metadata repository (not related in any way to the IBM Information Server metadata repository mentioned earlier) for service descriptions. As the integration point for service metadata, WebSphere Service Registry and Repository establishes a central point for finding and managing service metadata that is acquired from a number of sources, including service application deployments and other service metadata and endpoint registries and repositories, such as UDDI. It is where service metadata that is scattered across an enterprise is brought together to provide a single, comprehensive description of a service. When that happens, visibility is controlled, versions are managed, proposed changes are analyzed and communicated, usage is monitored, and other parts of the SOA foundation can access service metadata with the confidence that they have found the copy of record.

In this context, WebSphere Service Registry and Repository handles the metadata management aspects of operational services and provides the system of record of these metadata artifacts—the place where anybody looking for a catalog of all services deployed in or used by the enterprise would go first. The WebSphere Service Registry and Repository provides registry functions supporting publication of metadata about services, their capabilities, requirements and semantics of services that enable service consumers to find services or to analyze their relationships.

A.4 Runtime flow

This section provides an overview of the runtime flow associated with processing an invocation of an IBM Information Server service by a client application.

In this section, we cover a brief overview of the service artifacts before describing the flow of a request through the system.

A.4.1 Service artifacts

Every IBM Information Server service generated by WebSphere Information Services Director is generated as an EJB session bean (Service Session Bean in Figure A-8 on page 547) regardless of whether the function is a DataStage job, QualityStage job, database stored procedure, or federated query. After the service session bean has been generated, additional artifacts get created depending upon whether SOAP over HTTP or EJB binding is requested for the service:

- With SOAP over HTTP binding, a router servlet and facade session bean is generated. The servlet invokes the facade session bean that in turn invokes the service session bean.
- With EJB binding, a facade session bean is generated that invokes the service session bean.

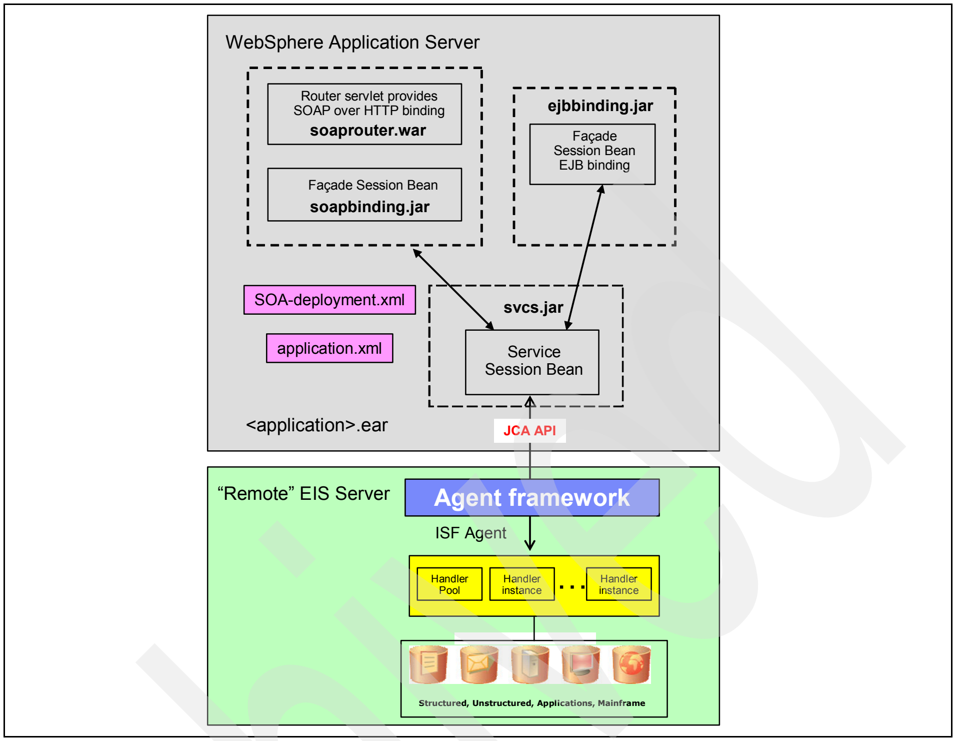


Figure A-8 Partial contents of IBM Information Server application EAR file

Note: The jar and war files shown in Example A-1 through Example A-4 relate to the creation of an information service named AccountService with SOAP over HTTP and EJB bindings in the BrokerageApp application within the A2ZProject.

The service session bean is packaged into an svcs.jar file along with other files as shown in Example A-1.

Example: A-1 The svcs.jar file contents

```
AccountService.class
AccountServiceBean.class
AccountServiceForWSDL.class
AccountServiceHome.class
AccountServiceRemote.class
ejb-jar.xml
ibm-ejb-jar-bnd.xmi
Manifest.mf
Response.class
RTIServiceEJBBase.class
```

The router servlet is packaged into a soaprouter.war file along with other files as shown in Example A-2.

Example: A-2 The soaprouter.war file contents

```
ibm-web-bnd.xml
ibm-web-ext.xml
Manifest.mf
web.xml
```

The facade session bean is packaged into a soapbinding.jar file along with other files as shown in Example A-3.

Example: A-3 The soapbinding.jar file contents

```
AccountService.wsdl
AccountService_mapping.xml
AccountServiceSOAPBindingBean.class
ejb-jar.xml
ibm-ejb-jar-bnd.xmi
ibm-webservices-bnd.xmi
ibm-webservices-ext.xmi
Manifest.mf
webservices.xml
```

The facade session bean is packaged into an ejbbinding.jar file along with other files as shown in Example A-4.

Example: A-4 The ejbbinding.jar file contents

```
AccountServiceBean.class
ejb-jar.xml
ibm-ejb-jar-bnd.xmi
Manifest.mf
```

All of these jar files, along with a soa-deployment.xml descriptor (shown in Example A-5 including both SOAP over HTTP and EJB bindings), and application.xml (Example A-7 on page 550) descriptor are packaged into A2ZBrokerageApp.ear file (shown in Example A-6 on page 550) that eventually gets deployed on the WebSphere Application Server (associated with IBM Information Server) by WebSphere Information Services Director.

> **Important:** We strongly recommend that you do *not* modify the various descriptors in the IBM Information Server <application>.ear file that is generated by WebSphere Information Services Director and attempt to deploy it in WebSphere Application Server using other tools. The results can be unpredictable.

Example: A-5 The soa-deployment.xml descriptor

```
<?xml version="1.0" ?>
- <soa-descriptor name="A2ZBrokerageApp">
- <service-descriptor name="AccountService" type="value">
- <description>
- <![CDATA[ Service for opening accounts
  ]]>
  </description>
- <entry-point>
  <home>com.ibm.isd.A2ZBrokerageApp.AccountService.server.AccountServiceHome</home>
<remote>com.ibm.isd.A2ZBrokerageApp.AccountService.server.AccountServiceRemote</remote>
<ejb-class>com.ibm.isd.A2ZBrokerageApp.AccountService.server.impl.AccountServiceBean</ej
b-class>
  <business>com.ibm.isd.A2ZBrokerageApp.AccountService.AccountService</business>
  </entry-point>
- <j2ee-descriptor reference="true">
  <jndi-name>ascential/rti/A2ZBrokerageApp/AccountService</jndi-name>
  <ejb-name>com.ibm.isd.A2ZBrokerageApp.AccountService.AccountService</ejb-name>
  <bean-type value="stateless" />
  </j2ee-descriptor>
  <category>/RTI</category>
  <initialization jndiName="" priority="1" />
  <allowable-binding>EJB</allowable-binding>
- <binding name="EJB">
  <property name="BeanDescription" value="" />
  <property name="JNDIName" value="ejb/A2ZBrokerageApp/AccountService" />
  <property name="Package" value="com.ibm.isd.A2ZBrokerageApp.AccountService.ejb" />
  </binding>
  <allowable-binding>SOAPHttp</allowable-binding>
- <binding name="SOAPHttp">
  <property name="Package" value="com.ibm.isd.A2ZBrokerageApp.AccountService" />
  <property name="TargetNameSpace" 
value="http://AccountService.A2ZBrokerageApp.isd.ibm.com/soapoverhttp/" />
  <property name="UriRoot" value="wisd" />
  <property name="WSDLClass" 
value="com.ibm.isd.A2ZBrokerageApp.AccountService.AccountServiceForWSDL" />
  <property name="SOAPStyle" value="DOCLIT" />
  <property name="SOAPAction" value="NONE" />
  </binding>
  </service-descriptor>
  </soa-descriptor>
```

Example: A-6 BrokerageApp.ear file

```
A2ZBrokerageApp_client.jar
application.xml
ejbbinding.jar
Manifest.mf
soa-deployment.xml
soapbinding.jar
soaprouter.war
svcs.jar
```

Example: A-7 Information Server application.xml

```
<?xml version="1.0" encoding="UTF-8" ?> 
- <application xmlns="http://java.sun.com/xml/ns/j2ee" 
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="Application_ID" 
version="1.4" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee 
http://java.sun.com/xml/ns/j2ee/application_1_4.xsd">
  <display-name>A2ZBrokerageApp</display-name> 
- <module>
  <ejb>svcs.jar</ejb> 
  </module>
- <module>
- <web>
  <web-uri>soaprouter.war</web-uri>
  <context-root>/wisd/A2ZBrokerageApp</context-root>
  </web>
  </module>
- <module>
  <ejb>soapbinding.jar</ejb>
  </module>
- <module>
  <ejb>ejbbinding.jar</ejb>
  </module>
  </application>
```

A.4.2 Flow of a request

A request for an IBM Information Server service can be invoked using SOAP over HTTP binding or EJB binding.

With SOAP over HTTP binding, an incoming request from a remote client is deserialized by the WebSphere Application Server SOAP stack (associated with IBM Information Server) and passed to the IBM Information Server and Information Services Framework as follows:

1. Invokes the router servlet with the interface parameters.
2. The router servlet then invokes the facade session bean.
3. Facade session bean invokes the service session bean using EJB binding.
4. The service session bean connects to the Information Services Framework (ISF) Agent[3] through the Agent framework using a J2EE Connector Architecture[4] (JCA) API to send a request to the backend and obtain a response.

> **Note:** As shown in Figure A-8 on page 547, the Agent Framework, ISF Agent, and backend data sources reside on a logically “remote” EIS server, which means that these components can be located physically on a separate server or co-located on the same server as the IBM Information Server.

[3] There is one ISF Agent associated with a DataStage server or Federation Server. If both DataStage server and Federation Server are installed on a server, only a single ISF Agent is installed on that server. An ISF Agent can be configured to access only one IBM Information Server. This architecture allows DataStage and Federation servers to be installed on servers distinct from where IBM Information Server is installed. A discussion of configuring IBM Information Server with multiple ISF Agents is beyond the scope of this publication.

The ISF Agent provides a framework for sending requests from the IBM Information Server to remote (ISF) clients (where the ISF Agent is located) without the need for a full J2EE Application Server at each client location. The ISF Agent utilizes a plugin architecture to allow different types of requests to be passed from the IBM Information Server. The ISF Agent framework also provides load balancing and pooling of resources.

The code that processes a request in the ISF Agent is called a Handler. Each ISF Agent can be configured to support multiple different Handlers at the same time. Currently, there are two handlers (a DataStage/QualityStage handler and a database stored procedure/federated query handler).

The ISF Agent framework takes care of routing requests and returning any responses. The clients (service session bean in our case) of the ISF Agent use the Java Connection Architecture (JCA) to send data. This consists of obtaining a Connection in much the same way that a JDBC Connection is obtained.

5. As each request arrives, the ISF Agent framework selects an instance of the requested Handler to process it. It does that by requesting a Handler instance from the handler pool. The handler pool in turn can decide to create a new instance or reuse an existing instance.
6. The Handler instance then processes the request and returns a response.

With EJB binding, an application such as a servlet or JSP™ invoke the facade session bean directly which invokes the service session bean. Thereafter, the processing is be identical to that of SOAP over HTTP.

[4] The J2EE Connector Architecture specifies a standard architecture for accessing resources in diverse Enterprise Information Systems (EIS) such as ERP systems, mainframe transaction processing systems, existing applications and non-relational database systems. The Connector Architecture defines a common interface (using the JCA API) between application servers and EIS systems, implemented in EIS specific resource adapters. A resource adapter is a system library specific to an EIS system such as SAP, and provides connectivity to that EIS through the JCA API. It is somewhat similar to a JDBC™ driver. The interface between the resource adapter and the EIS is typically specific to the underlying EIS. A Connector Architecture compliant resource adapter works with any J2EE server. A single resource adapter is provided with IBM Information Server that handles both the DataStage/QualityStage data source, and database stored procedure/federated query.

B

IBM Information Integrator Classic Federation setup

In this appendix we describe the setup of IBM Information Integrator Classic Federation for use by IBM WebSphere Information Analyzer.

B.1 Introduction

North American Bank's core and non-core services are provided on a z/OS platform. While most of the data sources are on DB2 for z/OS, one of the data sources is a VSAM file.

In order for IBM WebSphere Information Analyzer on our Linux platform to access North American Bank's data sources on the z/OS platform, you first need to install IBM WebSphere Classic Data Architect[1] and IBM WebSphere Classic Federation Server for z/OS[2] and then configure access to ODBC data sources on the z/OS platform.

1.6.2, "SETUPSTEP2: Configure ODBC to access data sources" on page 36 describes how to configure ODBC to access data sources in the distributed environment.

In this section, we describe how to configure data sources on the z/OS platform.

B.2 Configure ODBC data sources on the z/OS platform

The steps involved are as follows:

1. Install IBM WebSphere Classic Data Architect with the typical setup option on the Linux platform where IBM WebSphere Information Analyzer is installed—kazan.itsosj.sanjose.ibm.com in our case.

   For details on how to install IBM WebSphere Classic Data Architect, refer to:

   http://publib.boulder.ibm.com/infocenter/iisclzos/v9r1/index.jsp?topic=/com.ibm.websphere.ii.product.install.clas.doc/topics/iiypicac-instcda.html

2. Install IBM WebSphere Classic Federation Server for z/OS on the z/OS platform where North American Bank's DB2 for z/OS and VSAM data sources are located.

[1] IBM WebSphere Classic Data Architect is a new Eclipse-based GUI tool that assists you in configuring access to mainframe data sources and WebSphere Classic components.

[2] IBM WebSphere Classic Federation Server for z/OS, V09.01 (5655-R52) provides SQL access to mainframe databases and files with transactional speed and enterprise scale without mainframe programming. Using IBM WebSphere Classic Federation Server, applications and tools can issue SQL SELECT, INSERT, UPDATE, and DELETE commands using open database connectivity (ODBC), java database connectivity (JDBC), or a command-level Interface (CLI) to access System z™ data stored in VSAM, IAM and sequential files, as well as DB2 UDB for z/OS, IMS™, Software AG Adabas, and Computer Associates CA-Datacom and CA-IDMS databases, all without mainframe programming.

For details about how to install IBM WebSphere Classic Federation Server for z/OS, refer to *Program Directory for IBM WebSphere Classic Federation Server for z/OS V09.01.00, Program Number 5655-R52*, GI10-8750.

> **Attention:** It is essential that you install IBM WebSphere Classic Data Architect *before* you install IBM WebSphere Classic Federation Server for z/OS. Failure to do so will result in the ODBC drivers for z/OS not being installed.

3. You need to edit three files (dsenv, .odbc.ini, and uvodbc.config) that are located in the IBM WebSphere Information Analyzer installation directory to set up the required ODBC connections to the data sources to be accessed by IBM WebSphere Information Analyzer as follows:

   a. Modify the dsenv file (stores the environment variables) in the directory path /opt/IBM/InformationServer/Server/DSEngine on the Linux platform. Example B-1 on page 556 shows the partial contents of this file with the modifications. The highlighted entries show the modifications made to the dsenv file as follows:

      - LD_LIBRARY_PATH must specify the library path (/opt/ibm/wsclassic91/cli/lib) of the IBM WebSphere Classic Federation Server for z/OS Client.

        > **Attention:** This *must always* be the first entry.

      - Add the CAC_CONFIG parameter that includes the path (/opt/ibm/wsclassic91/cli/lib/cac.ini) of the cac.ini file.

        > **Note:** The cac.ini file must be edited for your specific z/OS configuration as shown in Example B-2 on page 556. The format for the DATASOURCE entry in this file when using TCP/IP is as follows:
        >
        > ```
        > DATASOURCE = sourcename tcp/hostname/portnumber
        > ```

Example: B-1 Partial contents of the modified dsenv file for z/OS ODBC access

```
...........
...........

   LD_LIBRARY_PATH=/opt/ibm/wsclassic91/cli/lib:`dirname
$DSHOME`/branded_odbc/lib:`dirname $DSHOME`/DSComponents/lib:`dirname
$DSHOME`/DSComponents/bin:$DSHOME/lib:$DSHOME/uvdlls:$ASBHOME/apps/jre/
bin:$ASBHOME/apps/jre/bin/classic:$ASBHOME/lib/cpp:$ASBHOME/apps/proxy/
cpp/linux-all-x86:$LD_LIBRARY_PATH

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTHOME/sqllib/lib;
        export LD_LIBRARY_PATH


CAC_CONFIG=/opt/ibm/wsclassic91/cli/lib/cac.ini
export CAC_CONFIG

THREADS_FLAG=native;export THREADS_FLAG
............
............
```

Example: B-2 cac.ini contents

```
* Path for the libraries
NL CAT = /opt/ibm/wsclassic91/cli/lib
NL = US English
* user id/pwd needed for catalog security (z/OS userid and password)
USERID = nalur1
USERPASSWORD = 12345678
* default datasource location
DEFLOC = CACSAMP
* if you have more than one server or one datasource you
* must provide multiple lines here
DATASOURCE = CACSAMP tcp/wtsc59.itso.ibm.com/5525
* performance and memory parameters
FETCH BUFFER SIZE = 32000
MESSAGE POOL SIZE = 4000000
* codepage parameters
SERVER CODEPAGE = IBM-037
CLIENT CODEPAGE = IBM-850
```

b. Edit the .odbc.ini file.

Example B-3 on page 557 shows the partial contents of the .odbc.ini file related to the classic federation setup. (You can find the complete contents of the .odbc.ini file that we used in our scenarios in Example 1-2 on page 40.)

Add the highlighted entries as shown in Example B-3.

```
[CACSAMP]
Driver=/opt/ibm/wsclassic91/cli/lib/libcacsqlcli.so
Database=CACSAMP
```

The highlighted names *must be* identical.

> **Note:** As mentioned in 1.6.2, “SETUPSTEP2: Configure ODBC to access data sources” on page 36, the [ODBC Data Sources] section lists the DSN names such as CACSAMP and associates them with the name of the driver, while the [ODBC] section lists ODBC tracing options and specifies where the ODBC drivers are installed.

Example: B-3 The .odbc.ini file contents

```
[ODBC Data Sources]
............
CACSAMP=WebSphere Classic Federation Client
...........
............
[CACSAMP]
Driver=/opt/ibm/wsclassic91/cli/lib/libcacsqlcli.so
Database=CACSAMP
...........
...........

[ODBC]
IANAAppCodePage=4
InstallDir=/opt/IBM/InformationServer/Server/branded_odbc
Trace=0
TraceDll=/opt/IBM/InformationServer/Server/branded_odbc/lib/odbctrac.so
TraceFile=odbctrace.out
UseCursorLib=0
```

c. Edit the uvodbc.config file.

Example B-4 shows the partial contents of the uvodbc.config file related to the classic federation setup. (You can find the complete contents of the uvodbc.config file that we used in our scenarios in Example 1-3 on page 43.)

Add the following entries as shown in Example B-4.

```
<CACSAMP>
DBMSTYPE = ODBC
```

Example: B-4 The uvodbc.config file contents

```
[ODBC DATA SOURCES]

<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = 127.0.0.1
........
<CACSAMP>
DBMSTYPE = ODBC
........
```

C

Miscellaneous tips regarding IBM WebSphere Information Analyzer

In this appendix we provide some general tips when using IBM WebSphere Information Analyzer.

C.1 General tips

This appendix includes a collection of tips about using IBM WebSphere Information Analyzer:

- Installation
  - Installation of IBM WebSphere Information Analyzer on Linux does not use the DB2 port number entered during installation configuration. It defaults to 50000.
  - DataStage setup in IBM WebSphere Information Analyzer

    In order to run IBM WebSphere Information Analyzer jobs, you must provide the DataStage User Name and DataStage Password under the Analysis Engine tab in the Analysis Settings workspace. You get to this workspace by clicking **Home** → **Configuration** → **Analysis Settings** and selecting the Analysis Engine tab.
- Metadata in a project is not refreshed automatically when the metadata is re-imported. Importing metadata only imports into the metadata repository. It does not update the metadata in projects.

  To update the metadata in a project, the data source must be added again to the project as follows:

  a. Assume that you changed the HOME_PHONE column in the CONTACT_INFO table from CHAR(10) to CHAR(15) in a data source.

  b. Re-import the metadata into IBM WebSphere Information Analyzer.

  c. Delete the HOME_PHONE column from the project under data sources, and save the settings (with the HOME_PHONE column deleted).

  d. Add the HOME_PHONE column to the project again, and save the settings with the HOME_PHONE column added. This now has the new metadata information.
- ODBC connection setup for IADB

  The setup of the ODBC-connection for the IADB database is not created during installation on the Windows platform. You must create this connection manually later.

  The connection to the IADB database works with both the IBM DB2 ODBC DRIVER - DB2 as well as the IBM DB2 Wire Protocol driver.

Figure C-1 on page 562 through Figure C-6 on page 565 shows the configuration of the ODBC connection to the IADB database using the IBM DB2 ODBC DRIVER - DB2.

a. To configure the ODBC connection, click **Start** → **Control Panel** → **Administrative Tools** and double-click **Data Sources (ODBC)** to display the ODBC Data Source Administrator screen as shown in Figure C-1 on page 562. Select the System DSN tab and click **Add** to add a System DSN.[1]

b. In the Create Data Source window, select the driver (IBM DB2 ODBC DRIVER - DB2) for which you want to set up a data source and click **Finish** as shown in Figure C-2 on page 562.

c. In the ODBC IBM DB2 Driver - Add window, select the DB2 database alias (IADB) that you want to register for ODBC, or select **Add** to create a new alias. Provide a Data source name (IADB) and optionally a Description (Information Analyzer database). Click **OK** as shown in Figure C-3 on page 563.

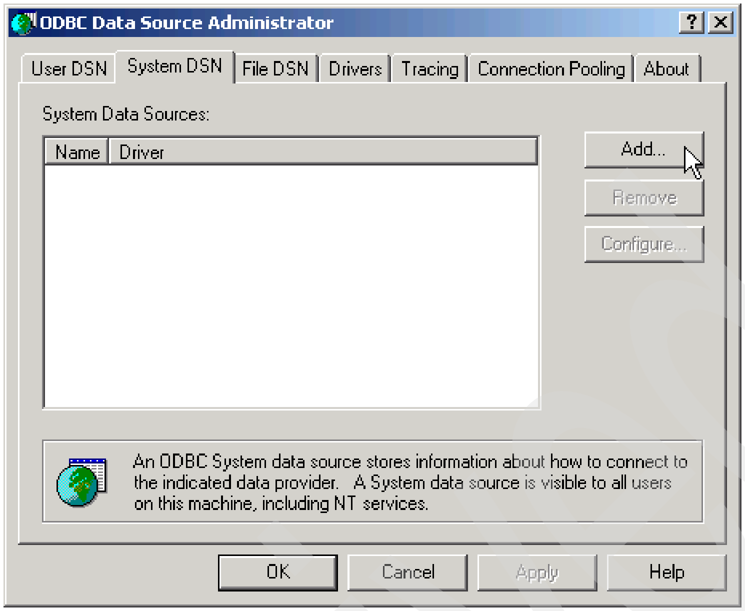
d. Select the TCP/IP tab in the CLI/ODBC Settings - IADB window in Figure C-4 on page 563, and fill in the details of the Database name (IADB), Host name (9.43.86.55), and Port number (50000) and click **OK** as shown in Figure C-5 on page 564. This completes the configuration of the ODBC connection to the IADB database using the IBM DB2 ODBC DRIVER - DB2 driver.

Important: You must set up the user ID and password for the database connection in IBM WebSphere Information Analyzer as shown in Figure 1-13 on page 47.

Figure C-6 on page 565 shows the configured System Data Source.

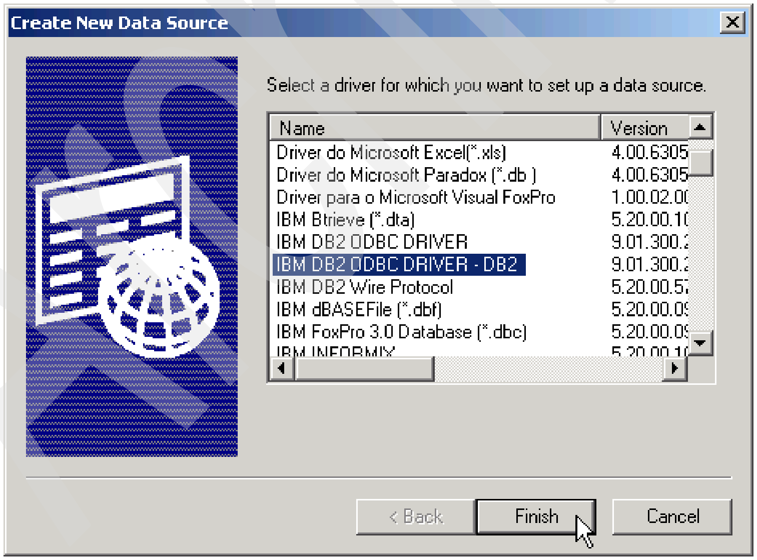
Note: Figure C-7 on page 565 through Figure C-9 on page 567 shows selected panels of the configuration of the ODBC connection to the redbank database using the IBM DB2 Wire Protocol driver. We do not describe this any further here.

[1] An ODBC System data source stores information about how to connect to the indicated data provider



An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

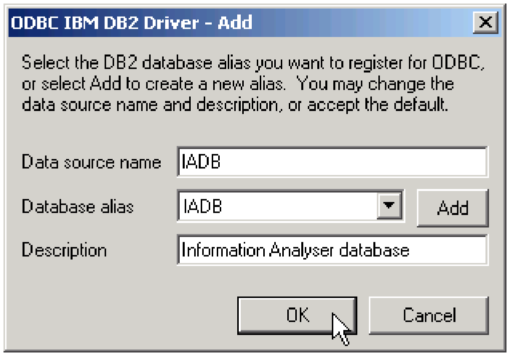
Figure C-1 Configure ODBC connection to IADB using IBM DB2 ODBC DRIVER - DB2 1/6



Select a driver for which you want to set up a data source.

Name	Version
Driver do Microsoft Excel(*.xls)	4.00.6305
Driver do Microsoft Paradox (*.db)	4.00.6305
Driver para o Microsoft Visual FoxPro	1.00.02.0(
IBM Btrieve (*.dta)	5.20.00.1(
IBM DB2 ODBC DRIVER	9.01.300.:
IBM DB2 ODBC DRIVER - DB2	9.01.300.:
IBM DB2 Wire Protocol	5.20.00.5;
IBM dBASEFile (*.dbf)	5.20.00.0!
IBM FoxPro 3.0 Database (*.dbc)	5.20.00.0!
IBM INFORMIX	5.20.00.1(

Figure C-2 Configure ODBC connection to IADB using IBM DB2 ODBC DRIVER - DB2 2/6



ODBC IBM DB2 Driver - Add

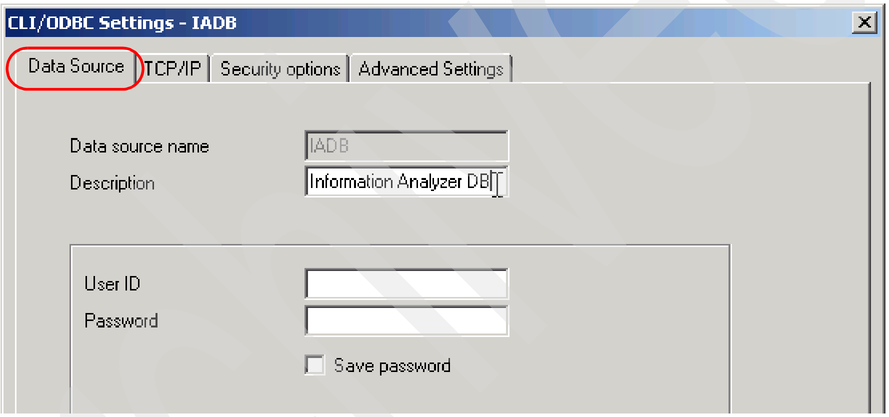
Select the DB2 database alias you want to register for ODBC, or select Add to create a new alias. You may change the data source name and description, or accept the default.

Data source name: IADB

Database alias: IADB

Description: Information Analyser database

Figure C-3 Configure ODBC connection to IADB using IBM DB2 ODBC DRIVER - DB2 3/6



CLI/ODBC Settings - IADB

Data source name: IADB

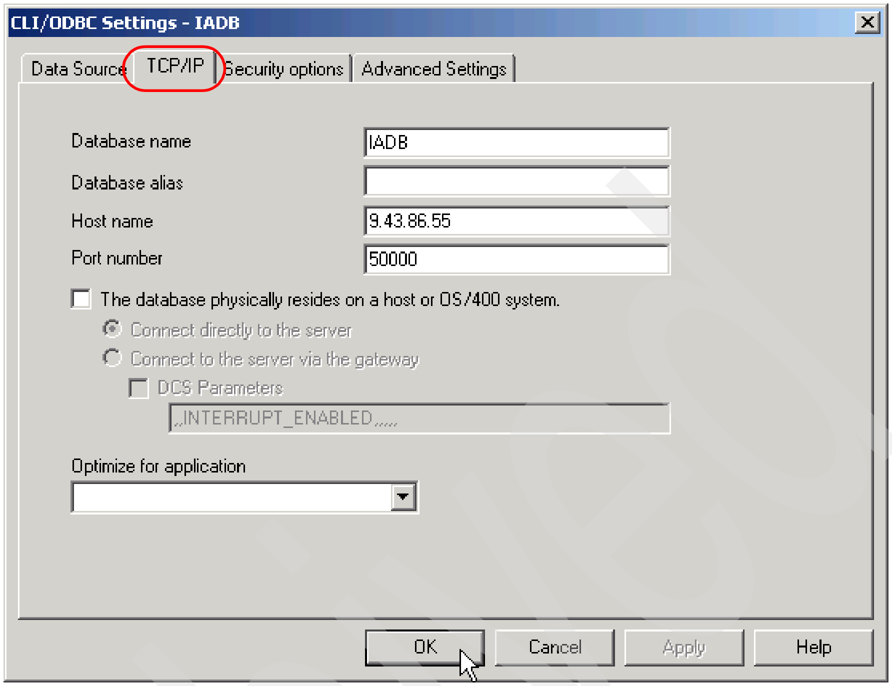
Description: Information Analyzer DB

User ID:

Password:

□ Save password

Figure C-4 Configure ODBC connection to IADB using IBM DB2 ODBC DRIVER - DB2 4/6



Database name: IADB

Database alias:

Host name: 9.43.86.55

Port number: 50000

□ The database physically resides on a host or OS/400 system.

● Connect directly to the server

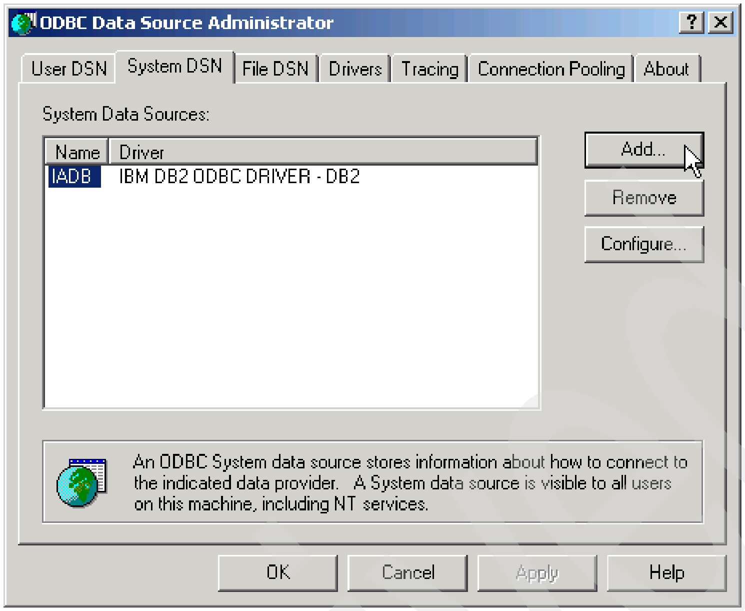
○ Connect to the server via the gateway

□ DCS Parameters

,,INTERRUPT_ENABLED,,,,,

Optimize for application:

Figure C-5 Configure ODBC connection to IADB using IBM DB2 ODBC DRIVER - DB2 5/6

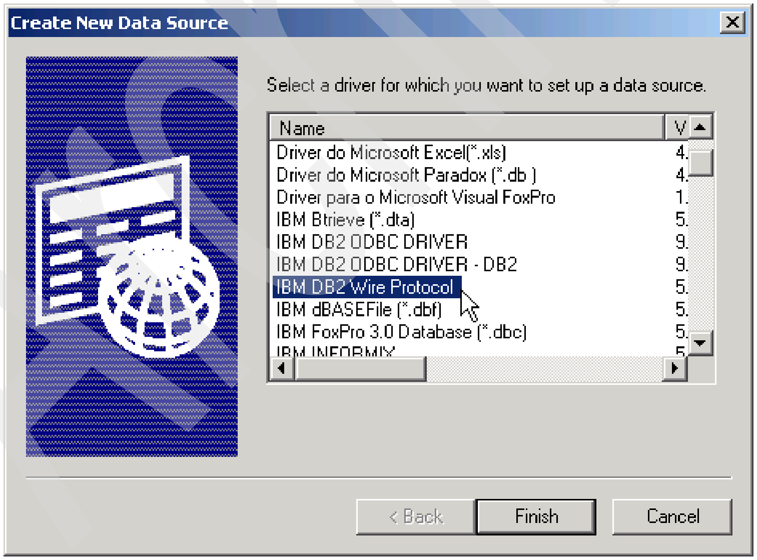


System Data Sources:

Name	Driver
IADB	IBM DB2 ODBC DRIVER - DB2

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

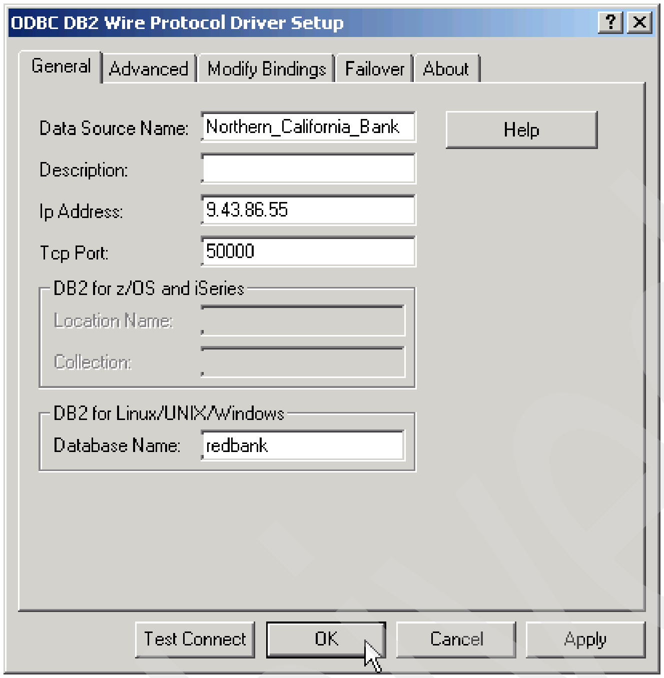
Figure C-6 Configure ODBC connection to IADB using IBM DB2 ODBC DRIVER - DB2 6/6



Select a driver for which you want to set up a data source.

Name	V
Driver do Microsoft Excel(*.xls)	4.
Driver do Microsoft Paradox (*.db)	4.
Driver para o Microsoft Visual FoxPro	1.
IBM Btrieve (*.dta)	5.
IBM DB2 ODBC DRIVER	9.
IBM DB2 ODBC DRIVER - DB2	9.
IBM DB2 Wire Protocol	5.
IBM dBASEFile (*.dbf)	5.
IBM FoxPro 3.0 Database (*.dbc)	5.
IBM INFORMIX	5.

Figure C-7 Configure ODBC connection to redbank using IBM DB2 Wire Protocol 1/3



Data Source Name: Northern_California_Bank

Description:

Ip Address: 9.43.86.55

Tcp Port: 50000

DB2 for z/OS and iSeries

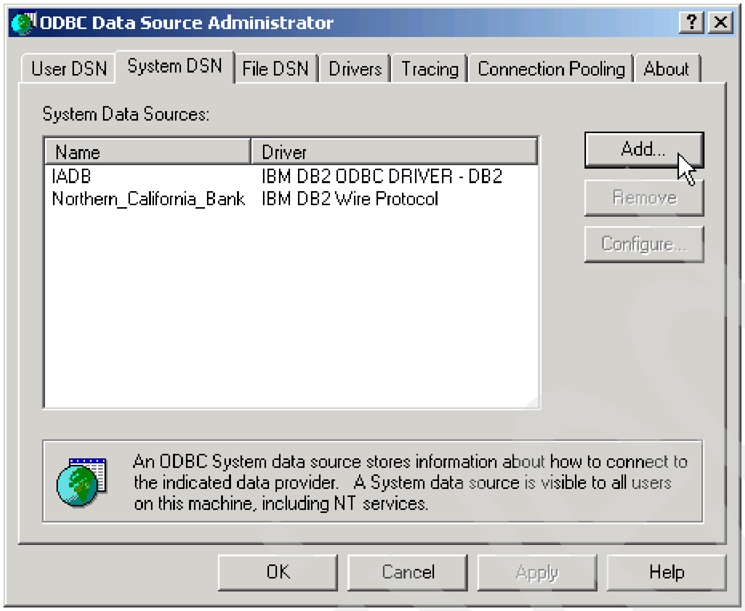
Location Name:

Collection:

DB2 for Linux/UNIX/Windows

Database Name: redbank

Figure C-8 Configure ODBC connection to redbank using IBM DB2 Wire Protocol 2/3



System Data Sources:

Name	Driver
IADB	IBM DB2 ODBC DRIVER - DB2
Northern_California_Bank	IBM DB2 Wire Protocol

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Figure C-9 Configure ODBC connection to redbank using IBM DB2 Wire Protocol 3/3

D

Code and scripts used in the business scenario

In this appendix we document some of the code and scripts that we used in the migration and data integration business scenarios included with this book.

D.1 Introduction

The code and scripts that we used in the migration and data integration business scenarios in this book include the following examples:

- Example D-1 on page 570 shows the DDL for creating the tables in the Northern California Bank data model.
- Example D-2 on page 579 and Example D-3 on page 585 show the DDL for creating the tables in the North American Bank data model and VSAM file definition respectively.
- Example D-4 on page 586 shows the DDL for creating the tables in the CRM data model.

Example: D-1 Fields in the tables in the Northern California Bank data model

```
-- This CLP file was created using DB2LOOK Version 9.1
-- Timestamp: Tue Sep 11 11:32:35 CDT 2007
-- Database Name: REDBANK
-- Database Manager Version: DB2/AIX64 Version 9.1.3
-- Database Codepage: 1252
-- Database Collating Sequence is: UNIQUE
CONNECT TO REDBANK;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."ACCTYPE"
------------------------------------------------
CREATE TABLE "DB2INST1"."ACCTYPE"  (
        "TYPE" CHAR(2) NOT NULL ,
        "DESCRIP" CHAR(50) NOT NULL ,
        "INTR" INTEGER ,
        "FEE" CHAR(20) ,
        "FEEFRQ" CHAR(1) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL ,
        "CURRENCY" CHAR(3) NOT NULL WITH DEFAULT '' )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."ACCTYPE"
ALTER TABLE "DB2INST1"."ACCTYPE"
   ADD CONSTRAINT "ACCTYPE_PK" PRIMARY KEY
      ("TYPE");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."CURRENCY"
------------------------------------------------

CREATE TABLE "DB2INST1"."CURRENCY"  (
        "CURRENCY" CHAR(3) NOT NULL ,
        "CTRY" VARCHAR(30) NOT NULL ,
        "NAME" VARCHAR(30) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL WITH DEFAULT 'USER5555' )
       IN "USERSPACE1" ;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."COUNTRY"
------------------------------------------------
CREATE TABLE "DB2INST1"."COUNTRY"  (
        "COUNTRY" VARCHAR(30) NOT NULL ,
        "CTRY2" CHAR(2) NOT NULL ,
        "CTRY3" CHAR(3) NOT NULL ,
        "CTRYN" CHAR(3) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL WITH DEFAULT 'USER5555' )
       IN "USERSPACE1" ;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."LANGUAGES"
------------------------------------------------
CREATE TABLE "DB2INST1"."LANGUAGES"  (
        "LAN3" CHAR(3) NOT NULL ,
        "LANGUAGE" VARCHAR(30) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL WITH DEFAULT 'USER5555' )
       IN "USERSPACE1" ;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BACCOUNT"
------------------------------------------------
CREATE TABLE "DB2INST1"."BACCOUNT"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +21001500
          INCREMENT BY +1
          MINVALUE +21001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "TYPE" CHAR(2) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
ALTER TABLE "DB2INST1"."BACCOUNT" ALTER COLUMN "ID" RESTART WITH
21001539;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BROKERAGE"
```

```
------------------------------------------------
CREATE TABLE "DB2INST1"."BROKERAGE"  (
        "OWNER" INTEGER NOT NULL ,
        "ACCOUNT" INTEGER NOT NULL ,
        "PORTFOLIO" INTEGER NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for indexes on Table "DB2INST1"."BROKERAGE"

CREATE INDEX "DB2INST1"."XBROKERAGE" ON "DB2INST1"."BROKERAGE"
      ("OWNER" ASC,
       "ACCOUNT" ASC,
       "PORTFOLIO" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BRANCH"
------------------------------------------------
CREATE TABLE "DB2INST1"."BRANCH"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +12001500
          INCREMENT BY +1
          MINVALUE +12001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "NAME" CHAR(50) NOT NULL ,
        "ADDR1" CHAR(50) NOT NULL ,
        "ADDR2" CHAR(50) ,
        "CITY" CHAR(30) NOT NULL ,
        "ZIP" CHAR(10) NOT NULL ,
        "COUNTRY" CHAR(3) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."BRANCH"
ALTER TABLE "DB2INST1"."BRANCH"
   ADD CONSTRAINT "BRANCH_PK" PRIMARY KEY
      ("ID");
ALTER TABLE "DB2INST1"."BRANCH" ALTER COLUMN "ID" RESTART WITH
12001559;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."CUSTOMER"
```

```
------------------------------------------------
CREATE TABLE "DB2INST1"."CUSTOMER"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +10001500
          INCREMENT BY +1
          MINVALUE +10001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "NAME" CHAR(50) NOT NULL ,
        "ADDR1" CHAR(50) NOT NULL ,
        "ADDR2" CHAR(50) ,
        "CITY" CHAR(30) NOT NULL ,
        "ZIP" CHAR(10) NOT NULL ,
        "COUNTRY" CHAR(30) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL ,
        "BRANCH" INTEGER ,
        "ADVISOR" INTEGER ,
        "HOMEPHONE" CHAR(15) ,
        "CELLPHONE" CHAR(15) ,
        "WORKPHONE" CHAR(15) ,
        "FAX" CHAR(15) ,
        "EMAIL" VARCHAR(50) ,
        "TYPE" CHAR(1) NOT NULL WITH DEFAULT '-' ,
        "CLASS" INTEGER NOT NULL WITH DEFAULT 0 ,
        "GENDER" CHAR(1) NOT NULL WITH DEFAULT '-' ,
        "PREF_LANG" CHAR(3) NOT NULL WITH DEFAULT 'ENG' )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."CUSTOMER"
ALTER TABLE "DB2INST1"."CUSTOMER"
   ADD CONSTRAINT "CUSTOMER_PK" PRIMARY KEY
      ("ID");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."EMPLOYEE"
------------------------------------------------
CREATE TABLE "DB2INST1"."EMPLOYEE"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +13001500
          INCREMENT BY +1
          MINVALUE +13001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
```

```
           ) ,
        "NAME" CHAR(50) NOT NULL ,
        "USERID" CHAR(8) ,
        "BRANCH" INTEGER ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for indexes on Table "DB2INST1"."EMPLOYEE"
CREATE INDEX "DB2INST1"."IEMPLOYEE1" ON "DB2INST1"."EMPLOYEE"
      ("ID" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
-- DDL Statements for primary key on Table "DB2INST1"."EMPLOYEE"
ALTER TABLE "DB2INST1"."EMPLOYEE"
   ADD CONSTRAINT "EMPLOYEE_PK" PRIMARY KEY
      ("ID");
ALTER TABLE "DB2INST1"."EMPLOYEE" ALTER COLUMN "ID" RESTART WITH
13001979;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."ACCOUNT"
------------------------------------------------
CREATE TABLE "DB2INST1"."ACCOUNT"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +11001500
          INCREMENT BY +1
          MINVALUE +11001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "OWNER" INTEGER NOT NULL ,
        "TYPE" CHAR(2) NOT NULL ,
        "SEC_OWNER" INTEGER ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL ,
        "CURRENCY" CHAR(3) )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."ACCOUNT"
ALTER TABLE "DB2INST1"."ACCOUNT"
   ADD CONSTRAINT "ACCOUNT_PK" PRIMARY KEY
      ("ID");
-- DDL Statements for indexes on Table "DB2INST1"."ACCOUNT"
CREATE INDEX "DB2INST1"."IACCOUNT2" ON "DB2INST1"."ACCOUNT"
      ("ID" ASC,
       "OWNER" ASC,
       "TYPE" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
ALTER TABLE "DB2INST1"."ACCOUNT" ALTER COLUMN "ID" RESTART WITH
11027739;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."COLLATERAL"
------------------------------------------------
CREATE TABLE "DB2INST1"."COLLATERAL"  (
        "ACCOUNT" INTEGER NOT NULL ,
        "TYPE" CHAR(2) NOT NULL ,
        "STATUS" CHAR(1) NOT NULL ,
        "EST_VAL" CHAR(20) NOT NULL ,
        "DESC" VARCHAR(200) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."COLLATERAL"
ALTER TABLE "DB2INST1"."COLLATERAL"
   ADD CONSTRAINT "COLLATERAL_PK" PRIMARY KEY
      ("ACCOUNT",
       "UPDATED");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."TRANSACTION"
------------------------------------------------
CREATE TABLE "DB2INST1"."TRANSACTION"  (
        "ACCOUNT" INTEGER NOT NULL ,
        "DESCR" CHAR(50) NOT NULL ,
        "CODE" CHAR(1) NOT NULL ,
        "CHANGE" CHAR(20) NOT NULL ,
        "BALANCE" CHAR(20) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."TRANSACTION"
ALTER TABLE "DB2INST1"."TRANSACTION"
   ADD CONSTRAINT "TRANSACTION_PK" PRIMARY KEY
      ("ACCOUNT",
       "UPDATED");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BCUSTOMER"
------------------------------------------------
CREATE TABLE "DB2INST1"."BCUSTOMER"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +20001500
          INCREMENT BY +1
          MINVALUE +20001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT CURRENT TIMESTAMP ,
        "BY" CHAR(8) NOT NULL ,
        "BRANCH" INTEGER ,
        "ADVISOR" INTEGER ,
        "NAME" VARCHAR(40) NOT NULL ,
        "ADDR1" VARCHAR(40) NOT NULL ,
        "ADDR2" VARCHAR(40) ,
        "CITY" VARCHAR(30) NOT NULL ,
        "ZIP" CHAR(10) NOT NULL ,
        "COUNTRY" VARCHAR(30) ,
        "EMAIL" VARCHAR(50) ,
        "BANKID" INTEGER )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."BCUSTOMER"
ALTER TABLE "DB2INST1"."BCUSTOMER"
   ADD CONSTRAINT "BCUSTOMER_PK" PRIMARY KEY
      ("ID");
-- DDL Statements for indexes on Table "DB2INST1"."BCUSTOMER"
CREATE INDEX "DB2INST1"."BCUSTOMER_PK" ON "DB2INST1"."BCUSTOMER"
      ("ID" ASC,
       "UPDATED" ASC)
      ALLOW REVERSE SCANS;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."PORTFOLIO"
------------------------------------------------
CREATE TABLE "DB2INST1"."PORTFOLIO"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +22001500
          INCREMENT BY +1
          MINVALUE +22001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "NAME" VARCHAR(40) NOT NULL ,
        "SYMBOL" CHAR(8) NOT NULL ,
        "ORDERED" DATE NOT NULL ,
        "PURCHASED" DATE ,
        "SELL_BY_DATE" DATE ,
        "SELL_BY_PRICE" CHAR(10) ,
        "SIZE" CHAR(20) NOT NULL ,
        "QUANTITY" CHAR(20) NOT NULL ,
        "PRICE" CHAR(20) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) ,
        "CURRENCY" CHAR(3) )
       IN "USERSPACE1" ;
-- DDL Statements for indexes on Table "DB2INST1"."PORTFOLIO"
CREATE INDEX "DB2INST1"."XPORTFOLIO" ON "DB2INST1"."PORTFOLIO"
      ("ID" ASC,
       "UPDATED" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
ALTER TABLE "DB2INST1"."PORTFOLIO" ALTER COLUMN "ID" RESTART WITH
22028339;

------------------------------------------------
-- DDL Statements for table "DB2INST1"."LOAN"
------------------------------------------------
CREATE TABLE "DB2INST1"."LOAN"  (
        "LOAN_ID" INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY (
          START WITH +1
          INCREMENT BY +1
          MINVALUE +1
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "ACCOUNT_ID" INTEGER NOT NULL ,
        "DESCRIPTION" CHAR(50) ,
        "INTEREST_RATE" CHAR(20) ,
        "INITIAL_LOAN_VALUE" CHAR(20) ,
        "OPENING_FEE" CHAR(20) ,
        "LATE_FEE" CHAR(20) ,
        "LATE_INTEREST_RATE" CHAR(20) ,
        "BALANCE" CHAR(20) )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."LOAN"
ALTER TABLE "DB2INST1"."LOAN"
   ADD PRIMARY KEY
      ("LOAN_ID");
-- DDL Statements for foreign keys on Table "DB2INST1"."CUSTOMER"
ALTER TABLE "DB2INST1"."CUSTOMER"
   ADD CONSTRAINT "ADVISOR_FK" FOREIGN KEY
```

```
      ("ADVISOR")
   REFERENCES "DB2INST1"."EMPLOYEE"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
ALTER TABLE "DB2INST1"."CUSTOMER"
   ADD CONSTRAINT "BRANCH_FK" FOREIGN KEY
      ("BRANCH")
   REFERENCES "DB2INST1"."BRANCH"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."EMPLOYEE"
ALTER TABLE "DB2INST1"."EMPLOYEE"
   ADD CONSTRAINT "CC1185467343863" FOREIGN KEY
      ("BRANCH")
   REFERENCES "DB2INST1"."BRANCH"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."ACCOUNT"
ALTER TABLE "DB2INST1"."ACCOUNT"
   ADD CONSTRAINT "ACCOUNTOWNER_FK" FOREIGN KEY
      ("OWNER")
   REFERENCES "DB2INST1"."CUSTOMER"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
ALTER TABLE "DB2INST1"."ACCOUNT"
   ADD CONSTRAINT "ACCOUNTTYPE_FK" FOREIGN KEY
      ("TYPE")
   REFERENCES "DB2INST1"."ACCTYPE"
      ("TYPE")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."COLLATERAL"
ALTER TABLE "DB2INST1"."COLLATERAL"
   ADD CONSTRAINT "COLLATERAL_FK" FOREIGN KEY
      ("ACCOUNT")
   REFERENCES "DB2INST1"."ACCOUNT"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."TRANSACTION"
ALTER TABLE "DB2INST1"."TRANSACTION"
```

```
   ADD CONSTRAINT "ACCOUNT_FK" FOREIGN KEY
      ("ACCOUNT")
   REFERENCES "DB2INST1"."ACCOUNT"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."LOAN"
ALTER TABLE "DB2INST1"."LOAN"
   ADD CONSTRAINT "SQL070911112510600" FOREIGN KEY
      ("ACCOUNT_ID")
   REFERENCES "DB2INST1"."ACCOUNT"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
```

Example: D-2   Fields in the tables in the North American Bank data model

```
-- This CLP file was created using DB2LOOK Version 9.1
-- Timestamp: 13-09-2007 14:27:14
-- Database Name: DB8A
-- Database Manager Version: DB2 Version 8.1.5
-- Database Codepage: 1208
------------------------------------------------
-- DDL Statements for table "SG247508"."ACCOUNT"
------------------------------------------------
CREATE TABLE "SG247508"."ACCOUNT"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "BRANCH_ID"  INTEGER NOT NULL ,
       "ACTIVE_IND"  CHAR(1) ,
       "BALANCE"  DECIMAL(9,2) ,
       "MIN_AMOUNT"  DECIMAL(9,2) ,
       "OVERDRAF"  DECIMAL(9,2) ,
       "OVERDRAF_LIMIT"  DECIMAL(9,2) ,
       "OVERDRAF_RATE"  DECIMAL(8,5) ,
       "OVERDRAF_FEE"  DECIMAL(9,2) ,
       "TYPE_IND"  CHAR(1)
      );
CREATE UNIQUE INDEX "NALUR1"."PKACCOUNT" ON "SG247508"."ACCOUNT"
      ( "ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."BRANCH"
------------------------------------------------
```

```
CREATE TABLE "SG247508"."BRANCH"
      (
       "BRANCH_ID"  INTEGER NOT NULL ,
       "BRANCH_DESCRIPTION"  CHAR(18) ,
       "WORK_ADDRESS"  CHAR(18) ,
       "WORK_ZIP"  CHAR(18)
      );
CREATE UNIQUE INDEX "NALUR1"."PKBRANCH" ON "SG247508"."BRANCH"
      ( "BRANCH_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CARD"
------------------------------------------------
CREATE TABLE "SG247508"."CARD"
      (
       "CARD_ID"  CHAR(16)  NOT NULL ,
       "PIN"  CHAR(4) ,
       "EXPIRE_DT"  TIMESTAMP ,
       "CARD_TYPE_CD"  CHAR(2)  NOT NULL ,
       "LEVEL_CD"  CHAR(2)  NOT NULL ,
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "CARD_CUST_NAME"  CHAR(18) ,
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "LIMIT"  DECIMAL(9,2) ,
       "WITHDRAW_LIMIT"  DECIMAL(9,2) ,
       "SECURITY_NUM"  CHAR(4) ,
       "LIMIT_BALANCE"  DECIMAL(9,2) ,
       "LIMIT_W_BALANCE"  DECIMAL(9,2) ,
       "FLAG_IND"  CHAR(1) ,
       "INTL_IND"  CHAR(1) ,
       "AUTOMAT_DEBIT_IND"  CHAR(1) ,
       "REWARDS_IND"  CHAR(1) ,
       "REWARDS_NUM"  VARCHAR(20) ,
       "REWARDS_CD"  CHAR(3)
      );
CREATE UNIQUE INDEX "NALUR1"."PKCARD" ON "SG247508"."CARD"
      ( "CARD_ID" ASC,    "CARD_TYPE_CD" ASC,    "CUSTOMER_ID" ASC,
"ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CARD_TRANSACTION"
------------------------------------------------
CREATE TABLE "SG247508"."CARD_TRANSACTION"
      (
       "CARD_ID"  CHAR(16)  NOT NULL ,
       "CARD_TYPE_CD"  CHAR(2)  NOT NULL ,
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "TRANSACTION_ID"  INTEGER NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "TRANSACTION_DT"  TIMESTAMP ,
       "VENDOR_NAME"  VARCHAR(50) ,
       "VENDOR_ID"  INTEGER,
       "INTL_IND"  CHAR(1) ,
       "AMOUNT"  DECIMAL(9,2) ,
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL ,
       "CUST_REFUSAL_IND"  CHAR(1) ,
       "LOCAL_CURRENCY_AMOUNT"  DECIMAL(9,2) ,
       "EXCHANGE_CURR_USED"  DECIMAL(9,2)
      );
CREATE UNIQUE INDEX "NALUR1"."PKCARDTRANS" ON
"SG247508"."CARD_TRANSACTION"
      ( "CARD_ID" ASC,    "CARD_TYPE_CD" ASC,    "CUSTOMER_ID" ASC,
"ACCOUNT_ID" ASC,    "TRANSACTION_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CARD_TYPE_REF"
------------------------------------------------
CREATE TABLE "SG247508"."CARD_TYPE_REF"
      (
       "CARD_TYPE_CD"  CHAR(2)  NOT NULL ,
       "DESCIPTION"  VARCHAR(20)
      );
CREATE UNIQUE INDEX "NALUR1"."PKCARDREF" ON "SG247508"."CARD_TYPE_REF"
      ( "CARD_TYPE_CD" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CAR_INSURANCE"
------------------------------------------------
CREATE TABLE "SG247508"."CAR_INSURANCE"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "INSURANCE_ID"  INTEGER NOT NULL ,
       "CAR_PLATE"  CHAR(10) ,
       "START_DT"  DATE,
       "END_DT"  DATE,
       "CAR_VALUE"  DECIMAL(9,2) ,
       "CLAIM_VALUE"  DECIMAL(9,2) ,
       "FULL_COVERAGE_IND"  CHAR(1) ,
       "THIRD_COVERAGE_LIMIT"  DECIMAL(9,2) ,
       "INSURANCE_COVERAGE"  DECIMAL(9,2) ,
       "INSURANCE_VALUE"  DECIMAL(9,2) ,
       "AUTOMAT_DEBIT_IND"  CHAR(1)
      );
```

```
CREATE UNIQUE INDEX "NALUR1"."PKCARINS" ON "SG247508"."CAR_INSURANCE"
      ( "ACCOUNT_ID" ASC,    "INSURANCE_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CONTACT_INFO"
------------------------------------------------
CREATE TABLE "SG247508"."CONTACT_INFO"
      (
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "WORK_PHONE"  CHAR(15) ,
       "CELL_PHONE"  CHAR(15) ,
       "HOME_PHONE"  CHAR(15) ,
       "HOME_ADDRESS"  VARCHAR(50) ,
       "HOME_ZIP"  CHAR(9) ,
       "WORK_ADDRESS"  VARCHAR(50) ,
       "WORK_ZIP"  CHAR(9) ,
       "PREF_LANG"  CHAR(3)  NOT NULL  WITH DEFAULT 'ENG'
      );
CREATE UNIQUE INDEX "NALUR1"."PKCUSTINFO" ON "SG247508"."CONTACT_INFO"
      ( "CUSTOMER_ID" ASC,    "ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CUSTOMER"
------------------------------------------------
CREATE TABLE "SG247508"."CUSTOMER"
      (
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "TITLE"  CHAR(3) ,
       "FIRST_NAME"  VARCHAR(20) ,
       "LAST_NAME"  VARCHAR(20) ,
       "GENDER_IND"  CHAR(1) ,
       "USERID"  VARCHAR(8) ,
       "PASSWORD"  VARCHAR(20) ,
       "CHURN_IND"  CHAR(1)  NOT NULL  WITH DEFAULT,
       "LEVEL_CD"  CHAR(2)  NOT NULL  WITH DEFAULT,
       "NICKNAME"  VARCHAR(20) ,
       "CREDIT_SCORE"  CHAR(18) ,
       "NATIONALITY"  VARCHAR(20)
      );

CREATE UNIQUE INDEX "NALUR1"."PKCUSTOMER" ON "SG247508"."CUSTOMER"
      ( "CUSTOMER_ID" ASC);

------------------------------------------------
-- DDL Statements for table "SG247508"."CUST_ACC"
------------------------------------------------
```

```
CREATE TABLE "SG247508"."CUST_ACC"
      (
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "ACCOUNT_ID"  INTEGER NOT NULL
      );
CREATE INDEX "SG247508"."IXCUSTAC2" ON "SG247508"."CUST_ACC"
      ( "ACCOUNT_ID" ASC);
CREATE INDEX "SG247508"."IXCUSTACC" ON "SG247508"."CUST_ACC"
      ( "CUSTOMER_ID" ASC);
CREATE UNIQUE INDEX "NALUR1"."PKCUST_ACC" ON "SG247508"."CUST_ACC"
      ( "CUSTOMER_ID" ASC,    "ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."DRIVER"
------------------------------------------------
CREATE TABLE "SG247508"."DRIVER"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "INSURANCE_ID"  INTEGER NOT NULL ,
       "DRIVER_ID"  INTEGER NOT NULL ,
       "NAME"  VARCHAR(50) ,
       "SSN"  CHAR(11) ,
       "BIRTH_DT"  DATE,
       "GENDER"  CHAR(1) ,
       "START_DRIVING"  DATE,
       "ADDRESS"  VARCHAR(50) ,
       "CITY"  VARCHAR(40) ,
       "STATE"  CHAR(2) ,
       "ZIP"  CHAR(9) ,
       "CORRECTIVE_LENSES_IND"  CHAR(1) ,
       "HAIR_COLOR"  VARCHAR(10) ,
       "HEIGHT"  VARCHAR(10) ,
       "WEIGHT"  VARCHAR(10)
      );
CREATE UNIQUE INDEX "NALUR1"."PKDRIVER" ON "SG247508"."DRIVER"
      ( "ACCOUNT_ID" ASC,    "INSURANCE_ID" ASC,    "DRIVER_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."LEVEL_REF"
------------------------------------------------
CREATE TABLE "SG247508"."LEVEL_REF"
      (
       "LEVEL_CD"  CHAR(2)  NOT NULL ,
       "DESCRIPTON"  VARCHAR(20)
      );
------------------------------------------------
-- DDL Statements for table "SG247508"."LOAN"
```

```
------------------------------------------------
CREATE TABLE "SG247508"."LOAN"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "LOAN_ID"  INTEGER NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "RATES"  DECIMAL(8,5) ,
       "INITIAL_VALUE"  DECIMAL(9,2) ,
       "LATE_FEE"  DECIMAL(9,2) ,
       "LATE_RATE"  DECIMAL(8,5) ,
       "BALANCE"  DECIMAL(9,2) ,
       "AUTOMAT_DEBIT_IND"  CHAR(1)
      );
CREATE UNIQUE INDEX "NALUR1"."PKLOAN" ON "SG247508"."LOAN"
      ( "ACCOUNT_ID" ASC,    "LOAN_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."LOAN_TRANSACTION"
------------------------------------------------
CREATE TABLE "SG247508"."LOAN_TRANSACTION"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "LOAN_ID"  INTEGER NOT NULL ,
       "TRANSACTION_ID"  INTEGER NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "TRANSACTION_DT"  TIMESTAMP ,
       "AMOUNT"  DECIMAL(9,2) ,
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL
      );

CREATE UNIQUE INDEX "NALUR1"."PKLOAN_TRANS" ON
"SG247508"."LOAN_TRANSACTION"
      ( "ACCOUNT_ID" ASC,    "LOAN_ID" ASC,    "TRANSACTION_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."REWARD_REF"
------------------------------------------------
CREATE TABLE "SG247508"."REWARD_REF"
      (
       "REWARDS_CD"  CHAR(3)  NOT NULL ,
       "DESCRIPTION"  VARCHAR(50)
      );
CREATE UNIQUE INDEX "NALUR1"."PKREWARD" ON "SG247508"."REWARD_REF"
      ( "REWARDS_CD" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."TRANSACTION"
------------------------------------------------
```

```
CREATE TABLE "SG247508"."TRANSACTION"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "TRANSACTION_ID"  INTEGER NOT NULL ,
       "TRANSACTION_DT"  TIMESTAMP ,
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "AMOUNT"  DECIMAL(9,2) ,
       "PAID_TO"  CHAR(18)
      );
CREATE UNIQUE INDEX "NALUR1"."PKTRANSA" ON "SG247508"."TRANSACTION"
      ( "ACCOUNT_ID" ASC,    "TRANSACTION_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."TRANSACTION_TYPE_REF"
------------------------------------------------
CREATE TABLE "SG247508"."TRANSACTION_TYPE_REF"
      (
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL ,
       "DESCRIPTION"  VARCHAR(20)
      );
```

Example: D-3   VSAM file containing EMPLOYEE records

```
CREATE TABLE "CAC"."EMPLOYEE" DBTYPE VSAM
   DS "CAC.VSAM.EMPLOYEE"
   (
   "EMPNAME" SOURCE DEFINITION
      DATAMAP OFFSET 0 LENGTH 21
      DATATYPE C
      USE AS CHAR(21),
   "DEPTNAME" SOURCE DEFINITION
      DATAMAP OFFSET 47 LENGTH 18
      DATATYPE C
      USE AS CHAR(18),
   "EMPNO" SOURCE DEFINITION
      DATAMAP OFFSET 72 LENGTH 8
      DATATYPE C
      USE AS CHAR(8));
```

Example: D-4 Fields in the tables in the CRM data model

```
--ISO code for languages
create table nabncb.iso_language (
       id            char(3) not null,
       name          varchar(30) not null,
       primary key(id)
       );
-- Home address, work address, home phone, call phone, e-mail
create table nabncb.contacttype (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Commercial banking, incurance, brokerage
create table nabncb.lineofbusiness (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Cross reference between CRM and other core and non-core systems
create table nabncb.custkeyxref (
       crmid            integer not null,
       nabcoreid        integer,
       nabnoncoreid     integer,
       ncbcoreid        integer,
       ncbnoncoreid     integer,
       primary key(crmid)
       );
-- Person, organization
create table nabncb.customertype (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Account owner, beneficiary, stakeholder, insurer
create table nabncb.role (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Currency, information on collateral for loans etc.
create table nabncb.item (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Types of relationships between customers; member of same household,
-- owner of business
create table nabncb.relationtype (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Branch information
create table nabncb.branch (
      id        integer not null,
      name      char(50) not null,
      primary key (id)
      );
-- Employee information
create table nabncb.employee (
      id        integer not null,
      name      char(50) not null,
      userid    char(8) not null,
      branch    integer not null,
      business  integer not null,
      primary key (id),
      foreign key (branch) references nabncb.branch (id),
      foreign key (business) references nabncb.lineofbusiness (id)
      );
-- Customer information. Rating from one to five stars, the more the
-- better
create table nabncb.customer (
       id               integer not null,
       prefix           varchar (10) not null,
       firstname        varchar(30) not null,
       middlename       varchar(30),
       lastname         varchar(30) not null,
       gender            char(1),
       nationality      varchar(20) not null,
       "TYPE"                   integer not null,
       preflang         CHAR(3) not null,
       advisor          integer,
       prefcontact      integer not null,
       homeStreet       varchar(30) not null,
       homeCity         varchar(20) not null,
       homeZip          varchar(10),
       homeCountry      varchar (20),
       workStreet       varchar(30) not null,
       workCity        varchar(20) not null,
       workZip         varchar(10),
       workCountry     varchar (20),
       homephone       varchar(15),
       workphone       varchar (20),
       cellphone       varchar(15),
       email           varchar (20),
       rating          char(5) not null,
       nabchkassets            decimal (9,2),
       nabsavassets            decimal (9,2),
       nabloanindicator        char(1) not null,
       nabloanamount           decimal (9,2),
       nabloanbalance          decimal (9,2),
       nabloanrate             decimal (6,3),
       ncbchkassets            decimal (9,2),
       ncbsavassets            decimal (9,2),
       ncbloanindicator        char(1) not null,
       ncbloanamount           decimal (9,2),
       ncbloanbalance          decimal (9,2),
       ncbloanrate             decimal (6,3),
       Brokindicator   char(1) not null,
       Brokassets      decimal (9,2),
       Brokmargin      decimal (9,2),
       CCindicator     char(1) not null,
       CClimit         integer,
       CCbalance                                       decimal (9,2),
       Carindicator    char(1) not null,
       Fullcoverind    char(1) not null,
       Carpremiums           decimal (6,2),
       Carenddate      date,
       primary key(id),
       foreign key (Advisor) references nabncb.employee (id),
       foreign key ("TYPE") references nabncb.contacttype (id),
       foreign key (preflang) references nabncb.iso_language (id),
       foreign key ("TYPE") references nabncb.customertype (id)
       );
-- Register relationship between customers; member of same household,
-- owner of business
create table nabncb.customerrelation (
       fromcustomer    integer not null,
       relationtype    integer not null,
       tocustomer      integer not null,
       primary key (fromcustomer,relationtype,tocustomer),
       foreign key (fromcustomer) references nabncb.customer (id),
       foreign key (relationtype) references nabncb.relationtype (id),
       foreign key (tocustomer) references nabncb.customer (id)
       );
-- Savings account, checkings account, car loan, home loan etc.
create table nabncb.product (
       id            integer not null,
       description   varchar(50) not null,
       business      integer not null,
       primary key(id),
       foreign key (business) references nabncb.lineofbusiness (id)
       );
-- Instance of a product related to one or more customers through
-- customerrole
create table nabncb.contract (
       id          integer not null,
       product     integer not null,
       status      integer not null,
       created     timestamp not null with default,
       updated     timestamp,
       primary key (id),
       foreign key (product) references nabncb. product (id)
       );
-- Additional information related to a specific contract e.g. currency
create table nabncb.contractitem (
       id          integer not null,
       contract    integer not null,
       item        integer not null,
       "value"       varchar(30) not null,
       primary key (id),
       foreign key (contract) references nabncb.contract (id),
       foreign key (item) references nabncb.item (id)
       );
-- Identifies the customer's role e.g. account owner, beneficiary,
-- stakeholder
create table nabncb.contractrole (
       id          integer not null,
       customer    integer not null,
       contract    integer not null,
       role        integer not null,
       primary key (id),
       foreign key (customer) references nabncb.customer (id),
       foreign key (contract) references nabncb.contract (id),
       foreign key (role) references nabncb.role (id)
       );
```

E

Additional material

This book refers to additional material that you can download from the Internet as described here.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247508`

Alternatively, you can go to the IBM Redbooks Web site at:

`ibm.com/redbooks`

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247508.

Using the Web material

The additional Web material that accompanies this Redbooks publication includes the following files (all code samples on a Windows operating system):

File name	*Hard disk space required*
SG247508.zip	Compressed Code Samples

How to use the Web material

Create a subdirectory (folder) on your workstation, and decompress the contents of the Web material zipped file into this folder.

Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 594. Note that some of the documents referenced here might be available in softcopy only.

- *SOA Solutions Using IBM Information Server*, SG24-7402
- *IBM WebSphere QualityStage Methodologies, Standandarization, and Matching,* SG24-7546 (Upcoming book; not yet available)

Other publications

These publications are also relevant as further information sources:

- *IBM Information Server - Delivering information you can trust,* IBM United States Announcement 206-308 dated 12 December 2006
- *IBM Information Server Version 8.0.1 Planning, Installation, and Configuration Guide,* GC19-1048
- *IBM Information Server Version 8.0.1 Information Server Introduction,* SC19-1049
- *IBM Information Server Version 8.0.1 IBM Information Server Administration Guide,* SC19-9929
- *IBM Information Server Version 8.0.1 Reporting Guide,* SC19-1162
- *IBM Information Server Quick Start Guide*
- *IBM Information Server — Delivers next generation data profiling analysis and monitoring through the new IBM WebSphere Information Analyzer module,* IBM United States Announcement 207-043 dated 13 March 2007
- *IBM Information Management Software Profiling: Take the first step toward assuring data quality*, December 2006, IMW11808-USEN-00

- *WebSphere Information Analyzer Version 8.0.1 WebSphere Information Analyzer User Guide,* SC18-9902
- *Ascential AuditStage Installation Guide Version 7.0.1*, 00D-002AS701
- *Ascential AuditStage Methodology and Application Guide Version 7.0.1*, 00D-003AS701
- *Ascential AuditStage User's Guide Version 7.0.1,* 00D-001AS701

Online resources

This Web site is also relevant as further information sources:

- IBM Information Server information center

  `http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp`

How to get IBM Redbooks publications

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

IBM WebSphere Information Analyzer and Data Quality Assessment

(1.0" spine)
0.875"<->1.498"
460 <-> 788 pages

IBM WebSphere Information Analyzer and Data Quality Assessment

Redbooks®

IBM WebSphere Information Analyzer overview

Financial services business scenario

IBM Information Server overview

IBM Information Server is a revolutionary new software platform that helps organizations derive more value from the complex heterogeneous information that is spread across their systems. It enables organizations to integrate disparate data and deliver trusted information wherever and whenever needed, in line and in context, to specific people, applications, and processes.

IBM WebSphere Information Analyzer is a data profiling and analysis tool that is a critical component of IBM Information Server. It is designed to help business and data analysts understand the content, quality, and structure of their data sources by automating the data discovery process. Bundled with IBM WebSphere Information Analyzer is AuditStage, a data rule monitoring tool that is designed to help business and data analysts validate data and assess ongoing data quality trends.

This book describes a usage scenario that covers all dimensions of profiling, rule building, deployment, and quality monitoring through a data integration life cycle.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7508-00 ISBN 0738485500