Chapter 26.

February 2009

Welcome to the February 2009 edition of IBM InfoSphere Information Server Developer's Notebook. This month we answer the question;

> How do I perform application and/or server consolidation or migration using Information Server? Having one database application, I've acquired another which is similar, and wish to resolve dependencies and duplicate data-
>
> *Excellent question! This edition of (IBM) InfoSphere Information Server Developer's Notebook (IISDN) begins a series where we examine the Information Server components of Information Analyzer, Business Glossary, Business Glossary AnyWhere, and Metadata Workbench.*
>
> The Information Server components listed above offer numerous point and then high level areas of functionality. Perhaps the coolest area of functionality is end to end data lineage; How, When, Where and What was done to the piece of data I'm looking at-

Software versions

All of these solutions were *developed and tested* on (IBM) InfoSphere Information Server (IIS) version 8.1, using the Microsoft Windows XP/SP2 platform to support IIS client programs, and a RedHat Linux Advanced Server 4 (RHEL 4) FixPak U6 32 bit SMP server (Linux kernel version 2.6.9-67.EL-smp) to support the IIS server side components.

IBM InfoSphere Information Server allows for a single, consistent, and accurate view of data across the full width of the corporate enterprise, be it relational or non-relational, staged or live data. As a reminder, the IBM InfoSphere Information Server product contains the following major components;

> WebSphere Business Glossary Anywhere™, WebSphere Information Analyzer™, WebSphere Information Services Director™, WebSphere DataStage™, WebSphere QualityStage™, WebSphere Metadata Server and Metabridges™, WebSphere Metadata Workbench™, WebSphere Federation Server™, Classic Federation™, Event Publisher™, Replication Server™, InfoSphere Data Architect™, DataMirror Transformation Server™, and others.

Obviously, IBM InfoSphere Information Server is a large and capable product, addressing many strategic needs across the enterprise, and supporting different roles and responsibilities.

26.1 Terms and core concepts

As stated above, this edition of (IBM) InfoSphere Information Server Developer's Notebook (IISDN) is the first in a series that covers a coordinated set of topics. These topics include, but are not limited to;

- End to end data lineage
- Automatic data discovery
- Information maintenance, baselining and related

Which allow us to do server migrations and consolidations, application modernization, improve the accuracy of our reporting, and much more.

As a software product, Information Server is organized into a number of components, some optional, some always present. One of the always present components to Information Server is the (shared) Metadata Repository. In its physical sense, the Metadata Repository is just a SQL database. Each of the components of Information Server writes the knowledge and awareness of data they produce, to the Metadata Repository, where this metadata is available to every other component to Information Server.

By means of example;

An end user is looking at a business intelligence report of any source; Microsoft Excel, Cognos, Business Objects, any Web page, etcetera.

A column total in this report seems incorrect, so the end user highlights the report's column header title, and hot keys into a component of Information Server.

By navigating graphically, the end user can determine the data source or sources of the report, and whether any (Jobs/Programs) that feed these data sources have failed to run, whether Brett Favre will stay retired this time, and more.

The database and data used in this example

Information Server allows you to discover and maintain the *structure* of any data source, as well as its *content* data. For this edition of IISDN, we will use a modified version of the demonstration database from our favorite database server, IBM Informix Dynamic Server (IDS). If you don't have access to this sample database and data, the following is offered;

- The basic object hierarchy in any relational (SQL) database includes;

  Database Server Instance -> Database -> Schema -> Table -> Rows

  The given database vendors differ slightly as to the definition of a (database) server instance and/or database; generally, however, a

schema is a logical collection of tables. It matters little whether you have one schema with 100 tables, or 10 schemas with 10 tables each.

In our example we use 3 schemas, to allow us to better recognize and organize our tables.

– The IDS demonstration database is called "stores", and offers a basic representation of customer order processing; Customer, Order-Header, Order-Detail (the order line items), Stock, etcetera. Example as shown in Figure 26-1.

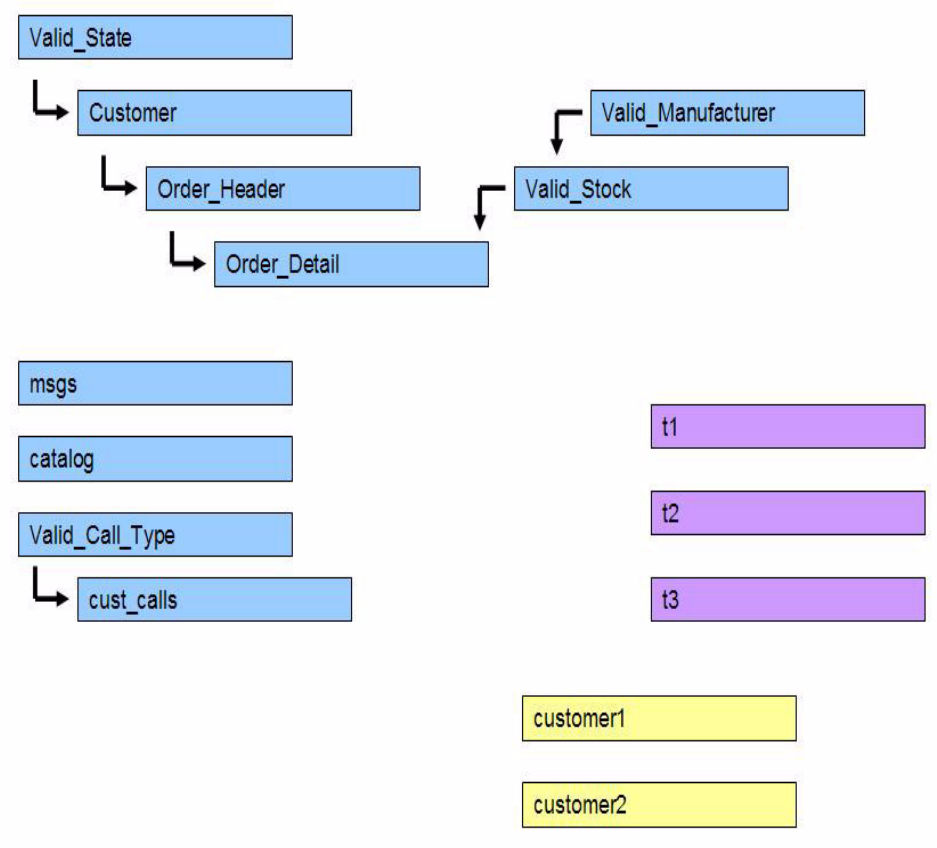

Figure 26-1 Sample database and data used in this example.

– In Figure 26-1. we hosted all of our tables and sample data in one database. We could just have easily used numerous databases or numerous database servers; Information Server cares little about these conditions.

We organized our tables into three schemas. The blue schema is entitled, Group_1, violet is entitled Group_2, and yellow is entitled Group_Delta.

– The Group_1 schema, contains the bulk of our tables by count. Any sample database you are using should contain the following conditions;

  • Single column primary key-

    The Valid_State table has a single primary key column (State_Abbr, of type CHAR(02)).

  • Single column foreign key-

    The Customer table, State_Abbr column, refers to Valid_State.State_Abbr primary key column, and thus forms a foreign key relationship.

  • Multi-column primary key-

    The sample database in Figure 26-1 has a Valid_Manufacturer table; the persons we buy our merchandise from. This database also has a Valid_Stock table. We may buy a volleyball, for example, from one or more Manufacturers.

    In our sample database, the Valid_Stock table has a multi-column primary key, on columns, Stock_Num and Manu_Code together.

  • Multi-column foreign key-

    The Order_Detail table contains the line items for a given Customer Order, and has a foreign key relationship to Valid_Stock. since Valid_Stock has a multi-column primary key, this foreign key relationship is also multi-column.

Note: We also created a table with no primary key, to see what that example looks like; see the table entitled, msgs (messages received from Customers).

  • Domains and ranges, and column formatting

    The Customer table, and more precisely the Phone (number) column, was specifically created with mixed format data. In the U.S., telephone numbers are 10 digits, with a format similar to; (303)555-1212.

    This column's data was specifically created with mixed and even short formatted data.

    In short, your sample database and data should work to contain mixed format data, NULL and/or incomplete to better demonstrate the examples that follow.

Note: The primary component to Information Server we are discussing in this edition of IISDN is Information Analyzer (IA). IA organizes its activities into logical steps;

- Column analysis
- Primary key analysis (single or multiple column)
- Foreign key analysis (single or multiple column)
- Cross domain analysis
- Base line analysis

The activities listed above form the basic/core functionality of IA. Of the activities listed above, only column analysis is a mandatory (and also first) step. Column analysis performs several examinations of the source columns being reviewed, including; column length, precision and scale, data type determination, formatting/masking of data, cardinality including blank and NULL values, and more.

In this edition of IISDN, we detail the configuration and core concepts of IA, through to the completion of foreign key analysis. The next edition of IISDN then completes the discussion of IA with cross domain analysis and base lining.

- Cross domain analysis

  The Group_2 schema contains three tables, generically entitled; t1, t2 and t3. Also the column names in each of these tables were changed to more obscure names; col01, col02, col03, and so on. And lastly, to further complicate matters, all of the column data types where changed to CHAR(64), regardless of whether the columns contained more precise INTEGER data, DECIMAL or DATETIME data, etcetera.

  These Group_2 schema tables, t1, t2 and t3, are exact copies of three tables from the Group_1 schema. Minimally, we will apply cross domain analysis from one of the obfuscated tables (table t2) into the Group_1 schema and look for a match. The simulated use case here would include having duplicate or nearly duplicate Customer files and the like.

  Cross domain analysis can be performed on key or non-key (non-index) columns. It may also be performed on multiple concurrent columns through the use of concatenated (virtual) columns.

- Base line analysis

The common use case for base line analysis is having an external data source that you rely upon. Because you don't own that data source, it may change in structure or content without advance notice. (You may gain or lose columns, change column format, etcetera, or the contents of a single or set of columns may change.)

The Group_3 schema contains two copies of the Customer table from Schema_1. We will baseline one of these tables, change it, and then automatically call to report on observed differences.

26.2 Configuration of Information Analyzer

There are a number of start up, one time only activities that must be performed before using the Information Analyzer component to Information Server. These activities include configuring the analysis database, analysis engine, and then defining the data sets that are available to given users.

Figure 26-2 displays the Home menu, inside the Information Analyzer graphical design program. There are 3 (count) parent menu items we work with, and in some cases, TABBED displays below these parent items. These items are labelled in the order in which we work with them.

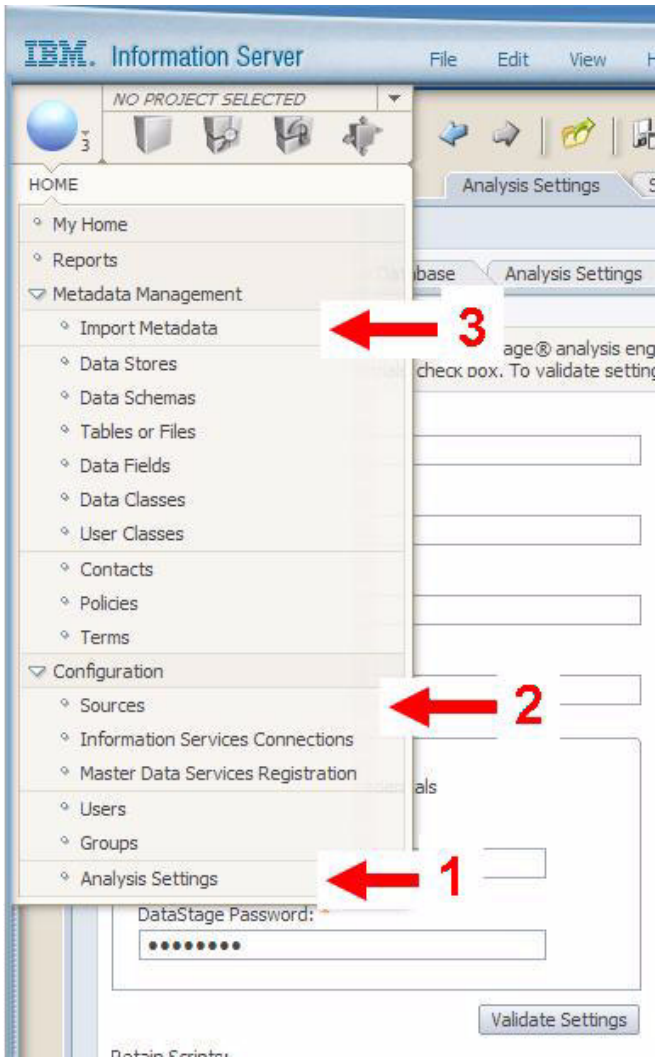

Figure 26-2 Main Menu of Information Analyzer.

Specify the Analysis Settings

Figure 26-3 displays the Main menu -> Analysis Settings -> Analysis Engine TABBED menu item. In this first of three TABS, we specify connectivity to the DSEngine proper, and then indirectly, the DataStage project hosting these component Jobs.

In Figure 26-3, be certain to make use of the, Validate Settings button.

Note: One of the other always present components to Information Server is the Parallel Framework. What the Parallel Framework allows is a single platform for all of the activities of Information Server to operate on top of, be it single CPU, single node (multi-CPU), or multi-node / multi-CPU.

Unbeknownst to casual users, many Information Analyzer activities generate DataStage parallel aware Jobs, and operate on top of the Parallel Framework. The net result is that Information Analyzer can handle even the largest sized data sets.

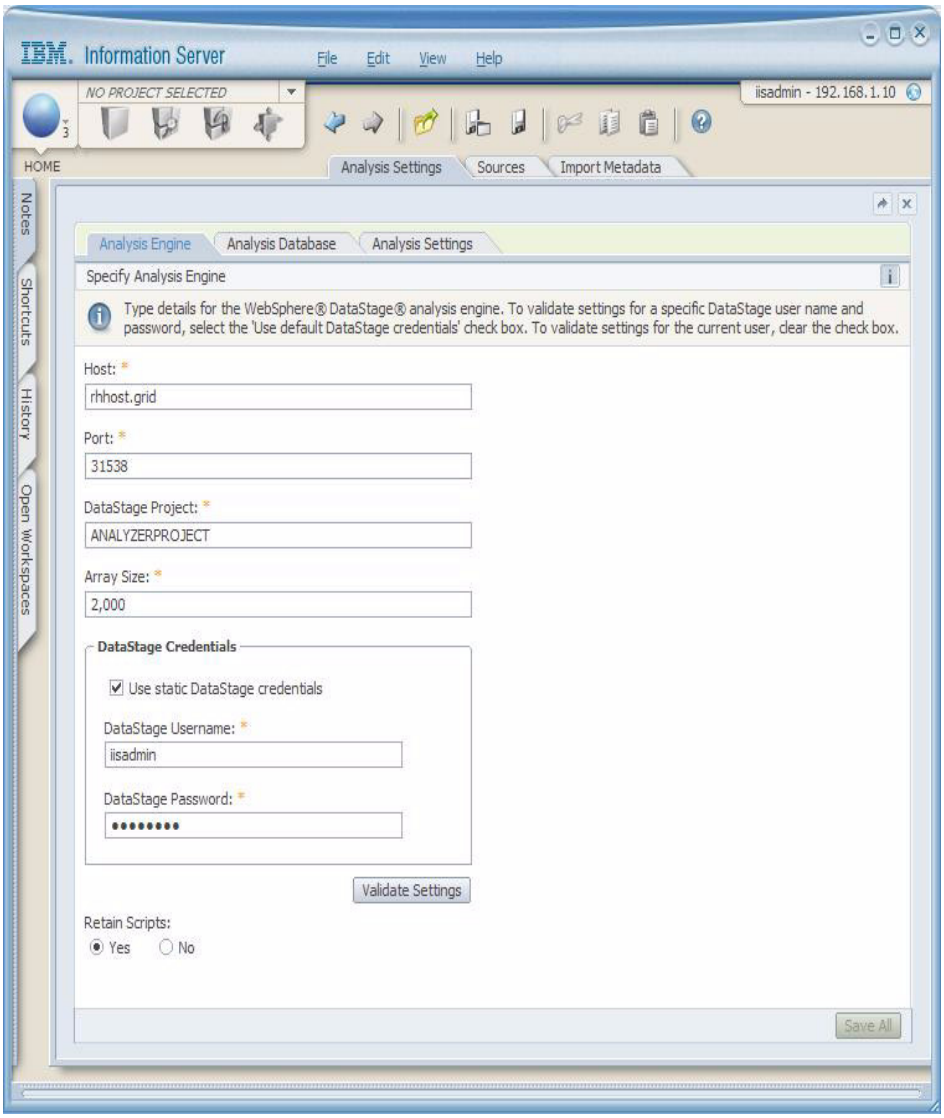

Figure 26-3 Analysis Settings -> Analysis Engine.

Figure 26-4 displays the Main menu -> Analysis Settings -> Analysis Database TABBED menu item. In this second of three TABS, we specify connectivity to the SQL database where Information Analysis tables and data will reside.

The text entry field entitled, Analysis ODBC DSN would be a new entry in the Information Server .odbc.ini configuration file, created in the manner described in the December/2007 edition of this document.

In Figure 26-4, be certain to make use of both Validation related buttons.

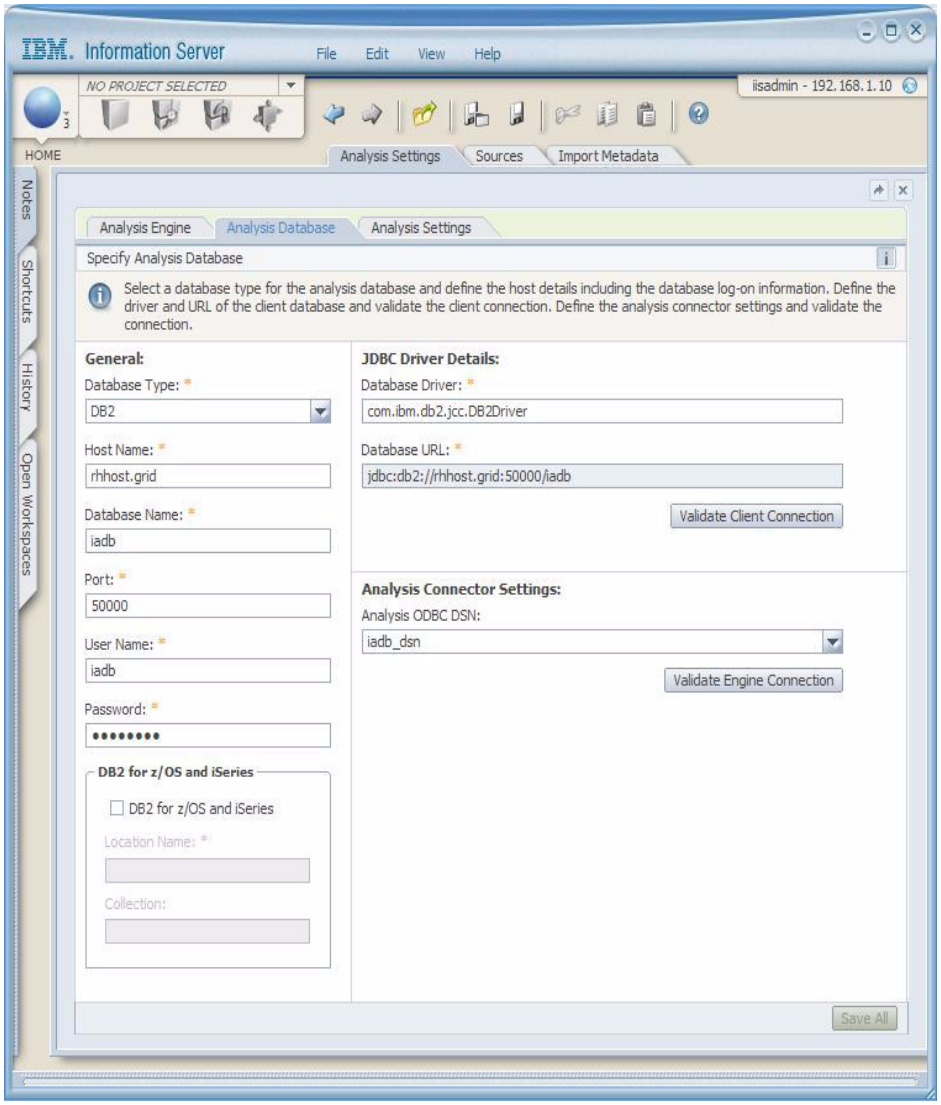

Figure 26-4 Analysis Settings -> Analysis Database.

Figure 26-5 displays the Main menu -> Analysis Settings -> Analysis Settings (Detail) TABBED menu item. In this third of three TABS, we specify operating limits.

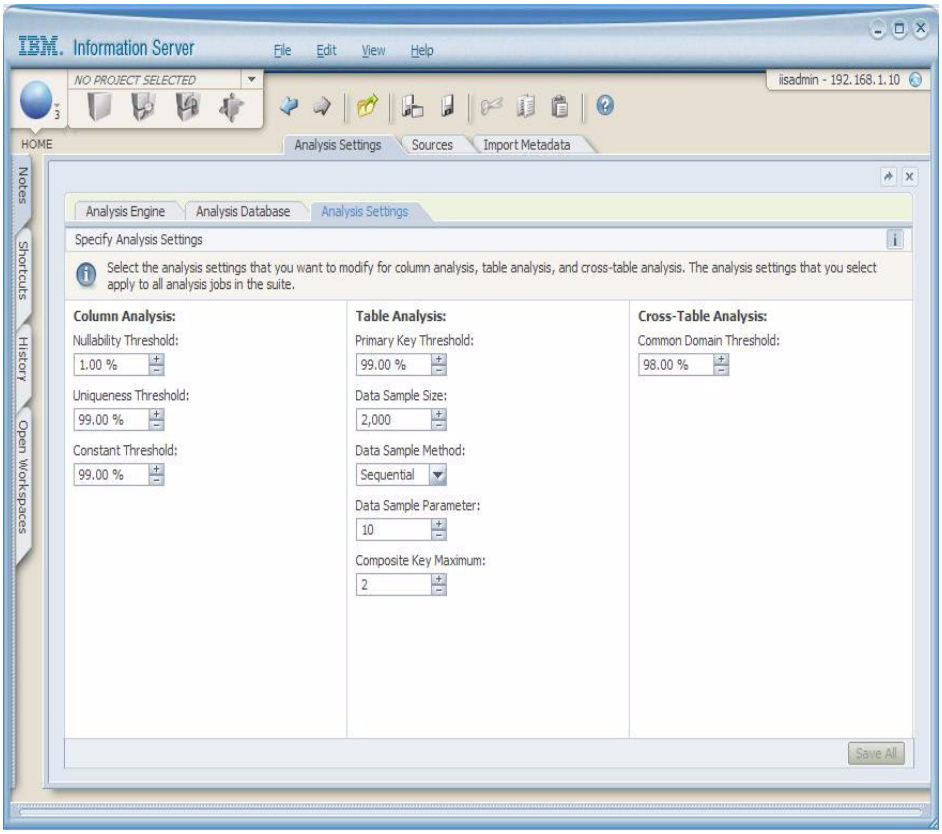

Figure 26-5 Analysis Settings -> Analysis Settings (detail).

Specify the Source (New Data Store)

Figure 26-6 displays the Main menu -> Source menu item. In effect we build a hierarchy of Hosts -> Data Stores, with Data Stores then offering physical (ODBC) connectivity parameters. The Data Stores defined here form the available list of Data Stores, which are then optionally added to Information Analyzer typed projects. (Think of the Source menu item, and its associated entries, as the global, system wide available list of data sources.)

Note: SQL schemas and tables are organized logically underneath Data Sources, but are not added underneath this menu item.

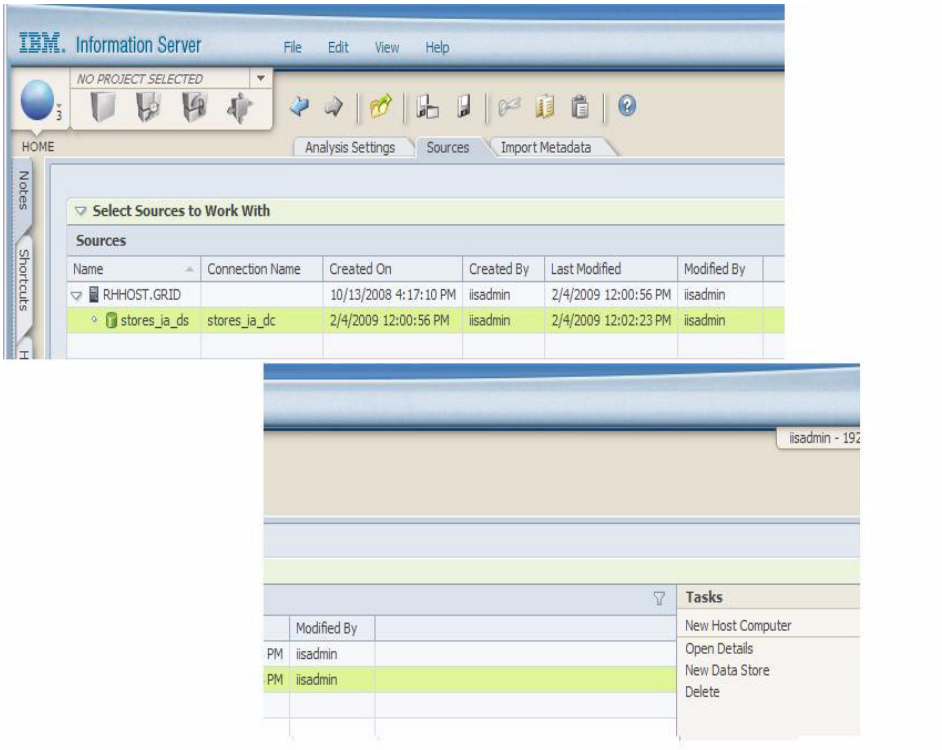

Figure 26-6 Sources -> (New Data Store).

Figure 26-7 displays the Main menu -> Source -> New Data Store menu item. The text entry field entitled, Connection String, is where we enter the ODBC DNS identifier; again, as detailed in the December/2007 edition of this document.

Additionally, not only must this identifier be defined in the Information Server .odbc.ini configuration file, this identifier must also be entered in the uvodbc.config file under the project directory for this Information Analyzer project. (Both topics are covered in the December/2007 document.)

Use the Connect button both for test, and to establish a live connection to this Data Store.

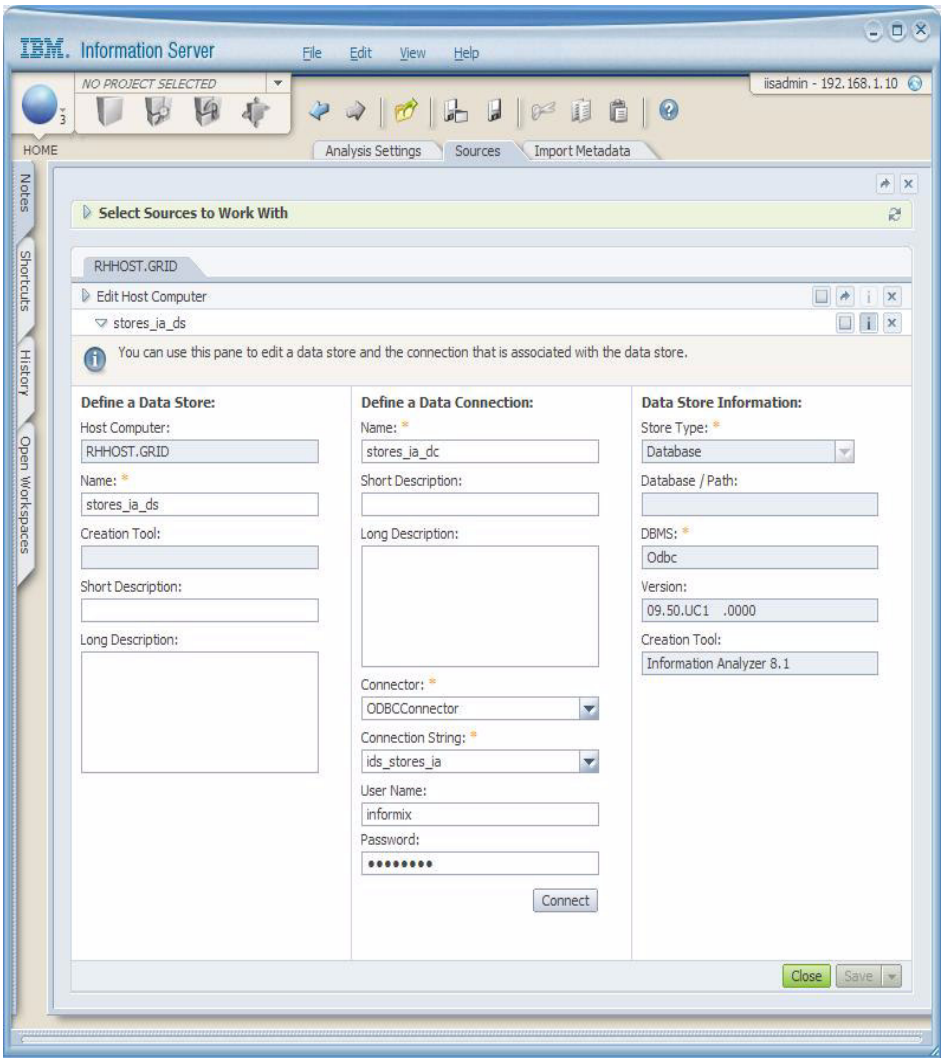

Figure 26-7 Sources -> New Data Store.

Importing MetaData (Schemas, Tables, Columns)

Figure 26-8 displays the Import MetaData menu item. From this location, we add schemas, tables and columns to the Hosts and Data Stores we added previously.

Procedurally and if we wish to be selective, we highlight a given Host or Data Store, click Import Next Level, and then Import; repeat as needed until all desired tables, etcetera are defined.

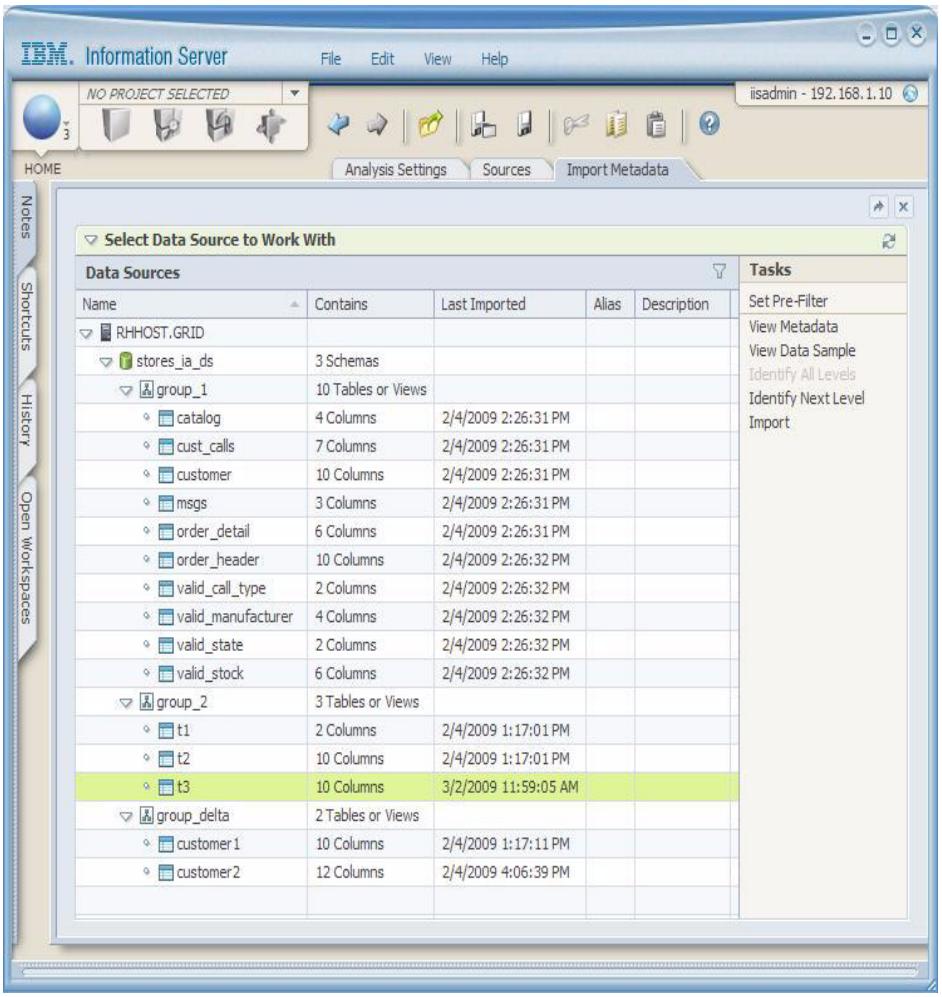

Figure 26-8 Import MetaData -> Identify Next Level -> Import.

Figure 26-8 displays our one Host, and one Data Store, with three schemas and associated tables and columns imported (defined).

Create a new Information Analyzer Project

Figure 26-9 displays creation of a new Information Analyzer Project. From the menu bar, select, File -> New Project.

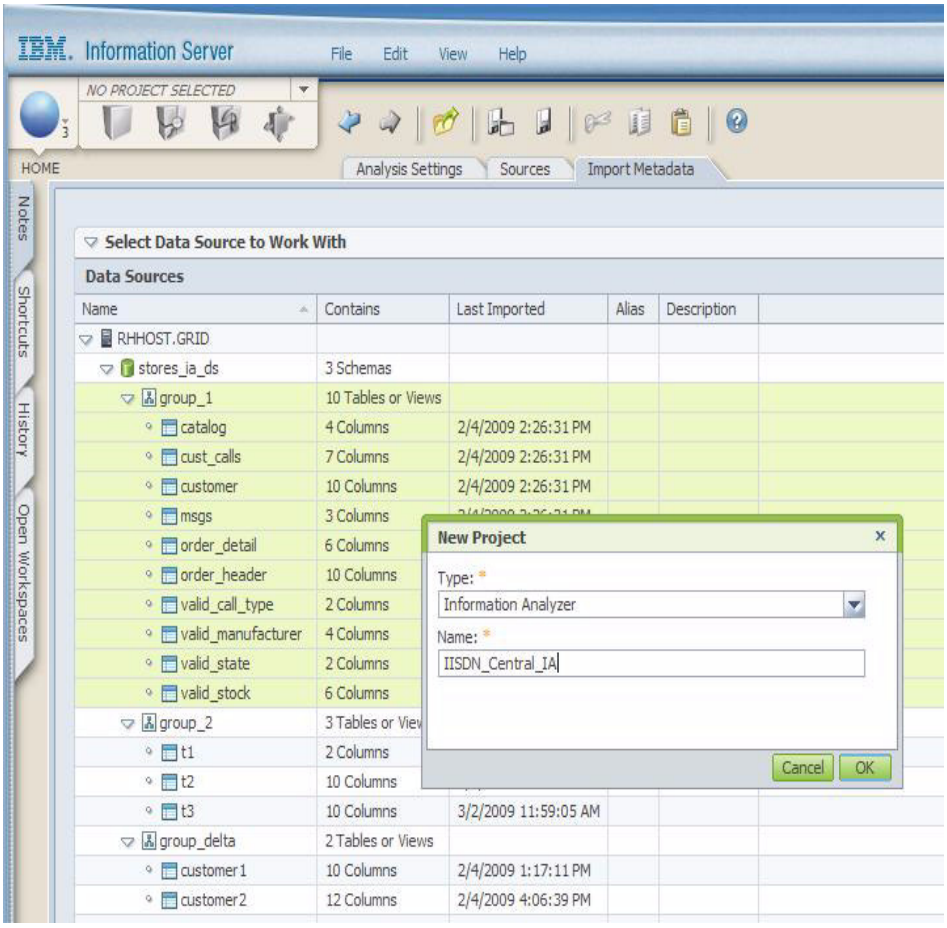

Figure 26-9 Creating a new Information Analyzer Project.

In Figure 26-10, the Overview -> Project Properties menu item offers a 5 TAB display screen; in Figure 26-10, only the Data Sources TAB is displayed. Under this TAB we pull in the Data Sources (all or a subset). First you Click the button entitled Add, and then navigate to the schemas and tables that were defined previously.

While Data Sources is the TAB the requires attention (mandatory changes), the other TABS for Users, Groups, and Details should be checked for accuracy of function.

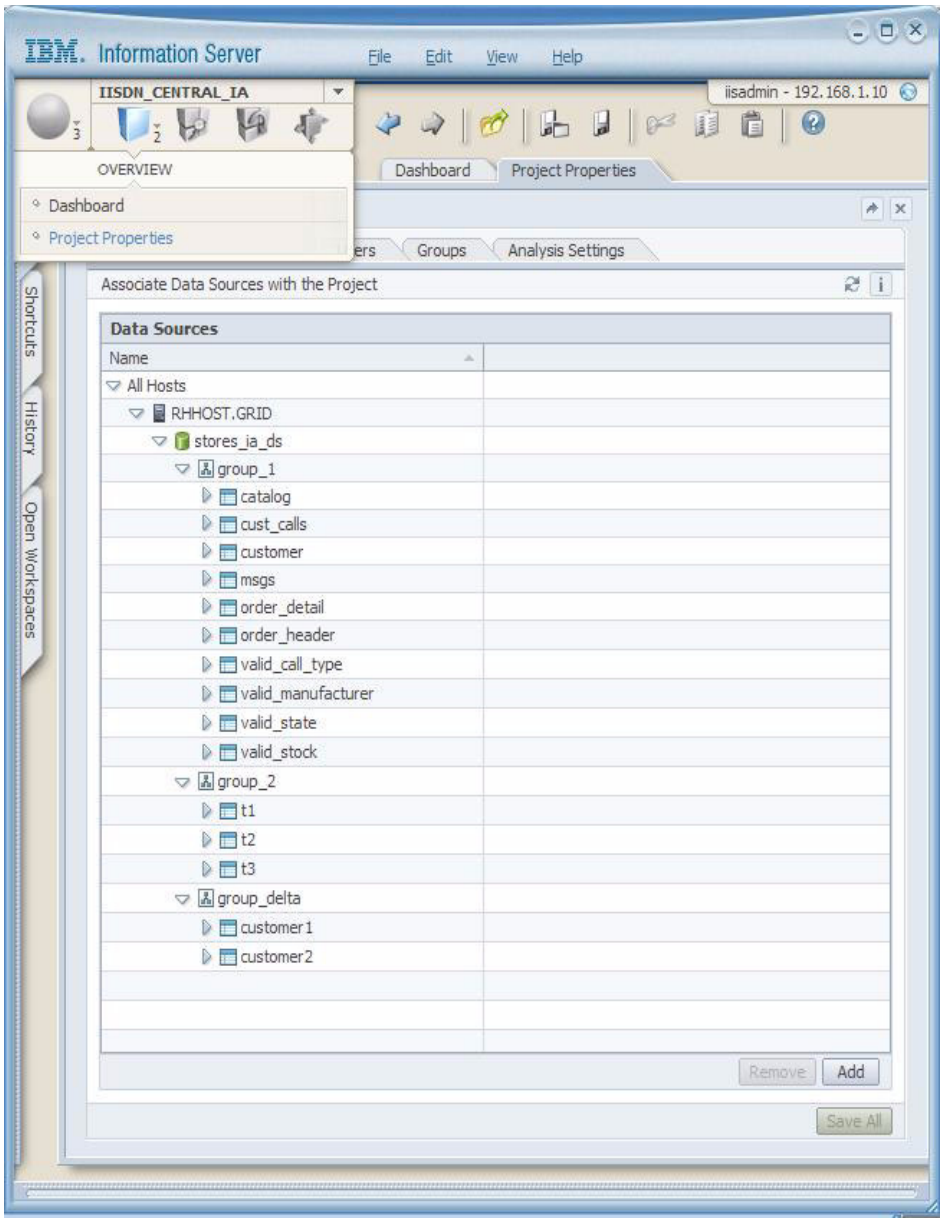

Figure 26-10 Overview -> Project Properties.

Thus completes our preparatory work; configuring Information Analyzer for use. By convention, notice we set the details of the Data Source at a server wide level, and then also within our Project; this allows scope to be set explicitly

system wide or then or a Project as this work is normally done by persons with differing levels of permission.

26.3 Create the examples in this document

In previous sections of this document, 26.1 and 26.2, we detailed the sample data source we are operating against in this example (26.1), and we detailed the post installation configuration of Information Analyzer (26.2). In this section of this document, we detail the Information Analyzer work proper. As this document offers part 1 in a multi-part series, we will complete the Information Analyzer through the following tasks;

- Column analysis, a required and mandatory first step
- Primary key analysis, single column
- Primary key analysis, multi-column
- Foreign key analysis, single column
- Foreign key analysis, multi-column

All of these tasks are accessed from the Investigate parent menu item, within the Information Server Console, as displayed in Figure 26-11.

Figure 26-11 Investigate menu.

The higher level, and more exciting tasks of (Cross) Domain Analysis and Base Line Analysis, follow in another edition of this document.

Performing column analysis within Information Analyzer

As mentioned previously, Information Analyzer has a number of primary activities it supports. Column analysis is the only mandatory of these steps, and hence, must happen first before all other activities. In each of the examples that follows, we will detail how to perform a given task one time, and then ask that it be repeated as needed.

To perform column analysis on the Valid_State table inside our target database, complete the following;

1. Logon to the Information Server Console, open the Information Server Project where your work is to be performed.
2. Perform column analysis-
   a. From the main menu, select Investigate -> Column Analysis. Navigate the Data Sources display to locate the Schema_1, and then the Valid_State table.

      Within the Valid_State table, highlight just one column; the State_Abbr column, the primary key.

Note: Many of the operations within Information Analyzer can be performed column by column, or table by table, schema by schema or even higher still, data source by data source.

In this example, and until you are certain that your runtime environment contains enough resource, perform activities column by column until you have successfully proven your environment can support the more resource intensive requests of table by table, and higher.

   b. With the State_Abbr column highlighted, the menu item on the right entitled, Run Column Analysis should become accessible. Click Run Column Analysis -> Run Now -> Submit -> Submit and Close.

      Example as displayed in Figure 26-12.

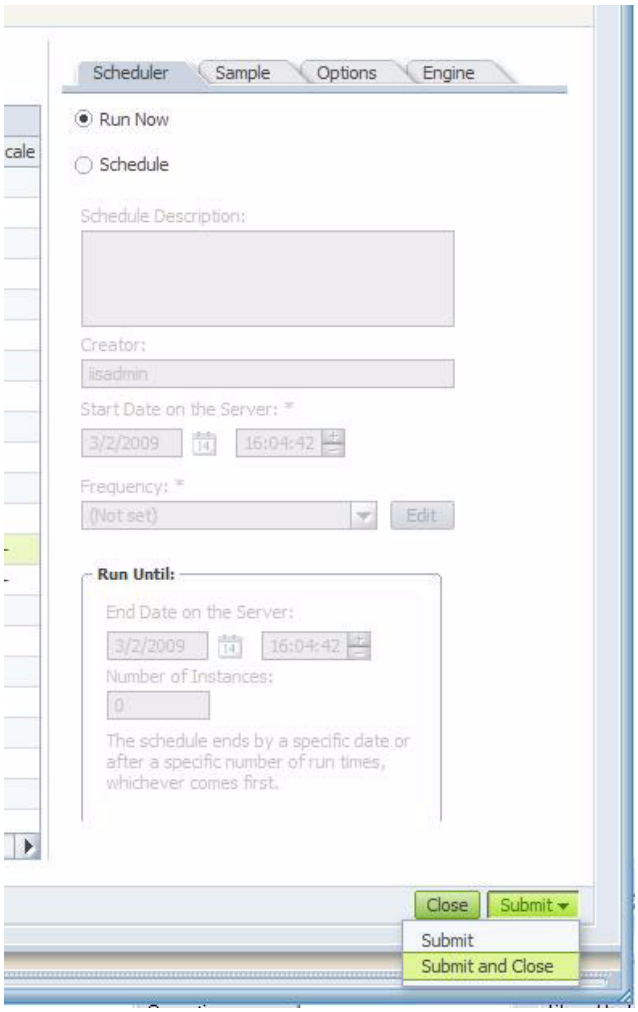

Figure 26-12 Run Column Analysis.

c. As displayed in Figure 26-13, a progress bar (see red arrow) will appear on the bottom of the window. By dragging open this (frame), you can increase size of this window to produce a Details button. Clicking this button gives you access to the display in the lower region of Figure 26-13.

   While these Jobs execute in the background, you may occasionally visit this screen to track progress.

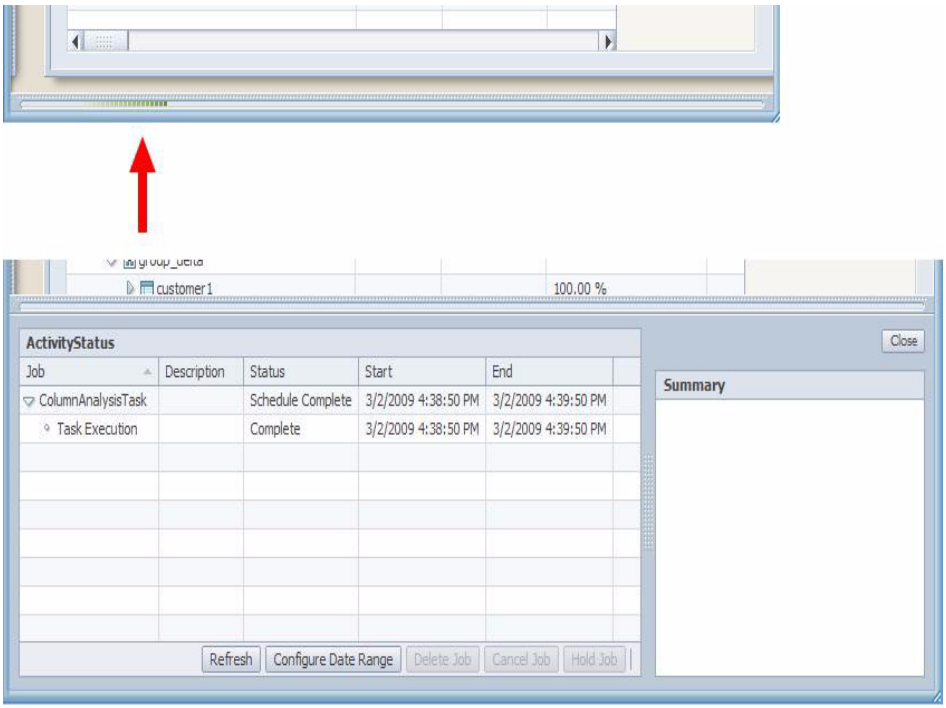

Figure 26-13 Activity Status window.

d. After column analysis is complete for the given column, highlight that column in the Data Sources to Work With area of the display, and then select Open Column Analysis from the menu on the right.

e. Select View Details from the screen that was produced.

   Figure 26-14 displays the Format TAB from the Customer table Phone column. Notice how you can highlight different values in the General format list to display live data.

   This example demonstrates the various distinct formats which our source data is comprised of, and, if we wished to produce standardized data, would be something we would need to address for this column.

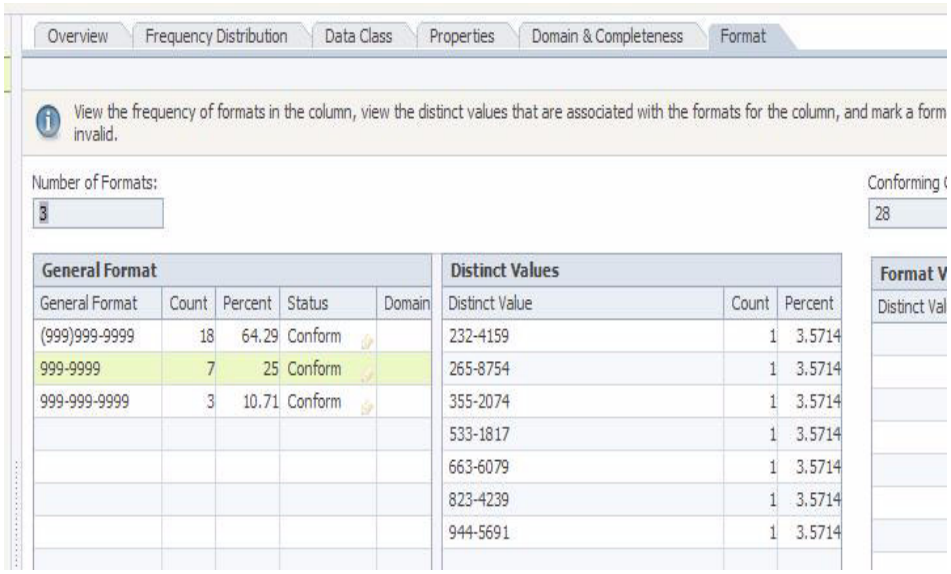

Figure 26-14 Open Column Analysis -> Format TAB.

Figure 26-15 displays the Properties TAB from Open Column Analysis.

If, for example, a column was encoded as type CHAR(64) in its source system, but really contained INTEGER data, you would want to record this condition. In a later document in this series, we are going to perform Cross Domain Analysis, and will want Information Analyzer to know our source column contains, in this case, INTEGER data.

Note: Why wouldn't Information Analyzer automatically record this condition?

The human operator may determine that the current source data contains only INTEGERS, but may at a future time for example, need to contain the less restrictive CHARACTER data.

Why does Information Analyzer need to know this at all?

After column analysis, Information Analyzer will use knowledge gained during this phase to perform heuristics; For example, it wouldn't compare a binary data type column to a date field for cross domain analysis. Recording the true and/or expected data type from column analysis supports your later stage processing.

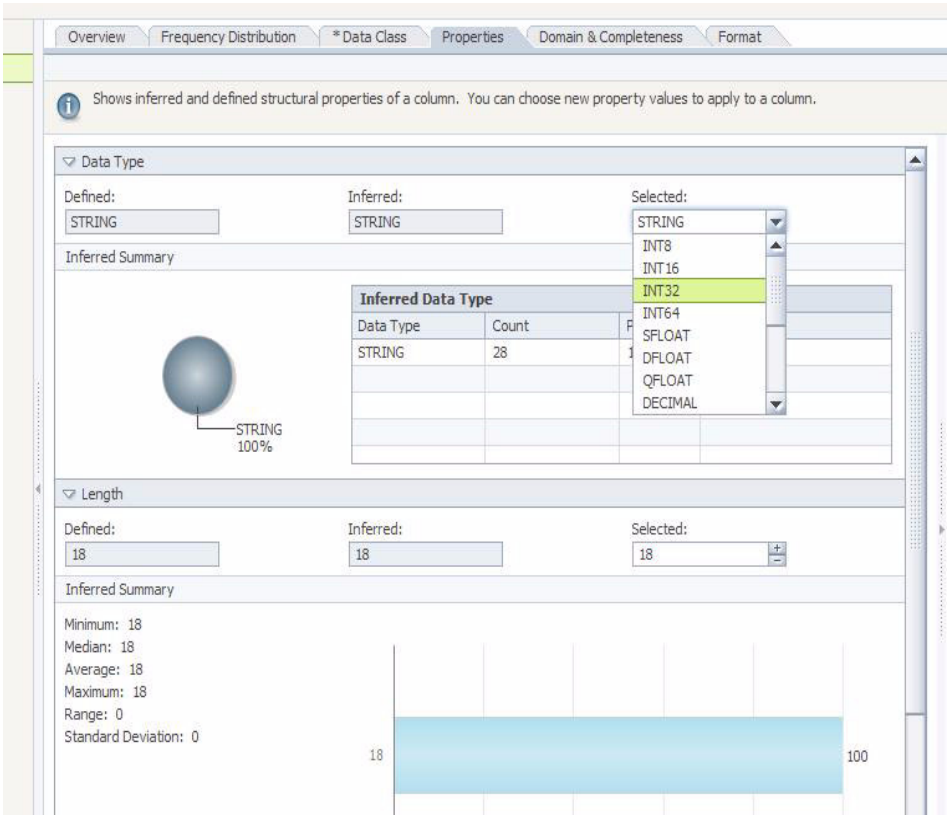

Figure 26-15 Open Column Analysis -> Properties TAB.

f. Complete column analysis for all remaining columns in the Data Source.

After completing all of the column analysis tasks, mark each column as being reviewed in each of the 4 member categories. Example as shown in Figure 26-16.

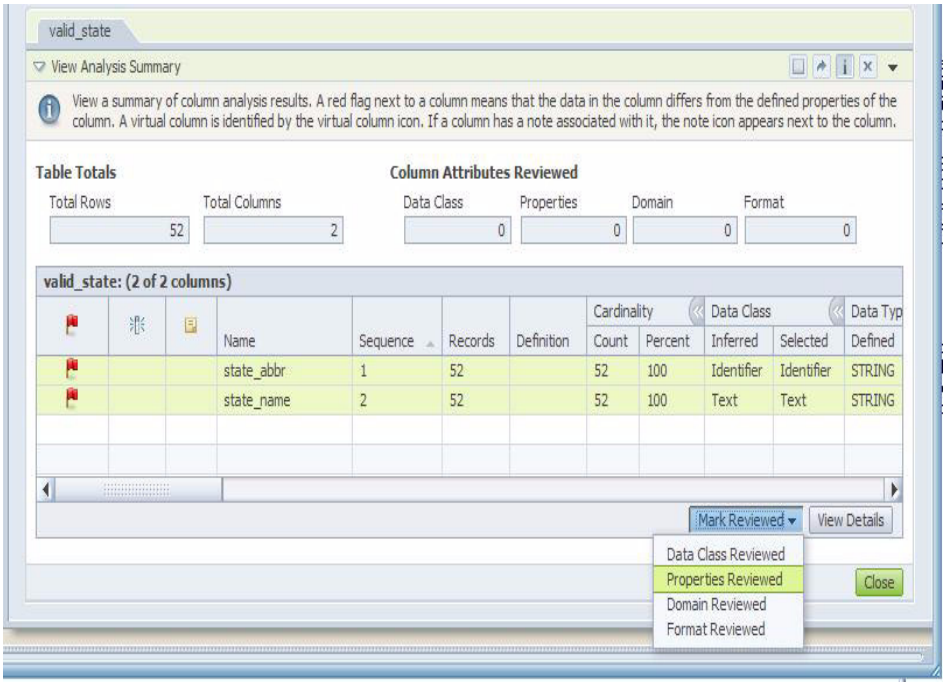

Figure 26-16 Column analysis, Marking columns as reviewed.

Performing primary key analysis, single and multi-column

As stated above, performing column analysis is a mandatory and first step to all further processing within Information Analyzer. After column analysis, single primary key column analysis is very easy; the column analysis phase already recorded the uniqueness of each (single) column.

3. Perform single column primary key analysis-
   a. From the main menu, select Investigate -> Primary Key Analysis
   b. Navigate the Data Sources display to locate the Schema_1, and then the Valid_State table.

      Primary key analysis is done table by table (or higher), not column by column (not lower than the table level). This is because having a Primary Key (or any key for that matter) is a property of a table, at the table level; State_Name it is not a property of single or set of columns.

   c. Highlight the Valid_State table and Click, Open Primary Key Analysis from the menu on the right.

      This action will produce the table, with all of its columns, and an indication as to the uniqueness of each individual column. For the Valid_State table,

both state abbreviation and state name columns contains exclusively unique values.

Highlight the State_Abbr column, and Click View Duplicate Check, example as shown in Figure 26-17.

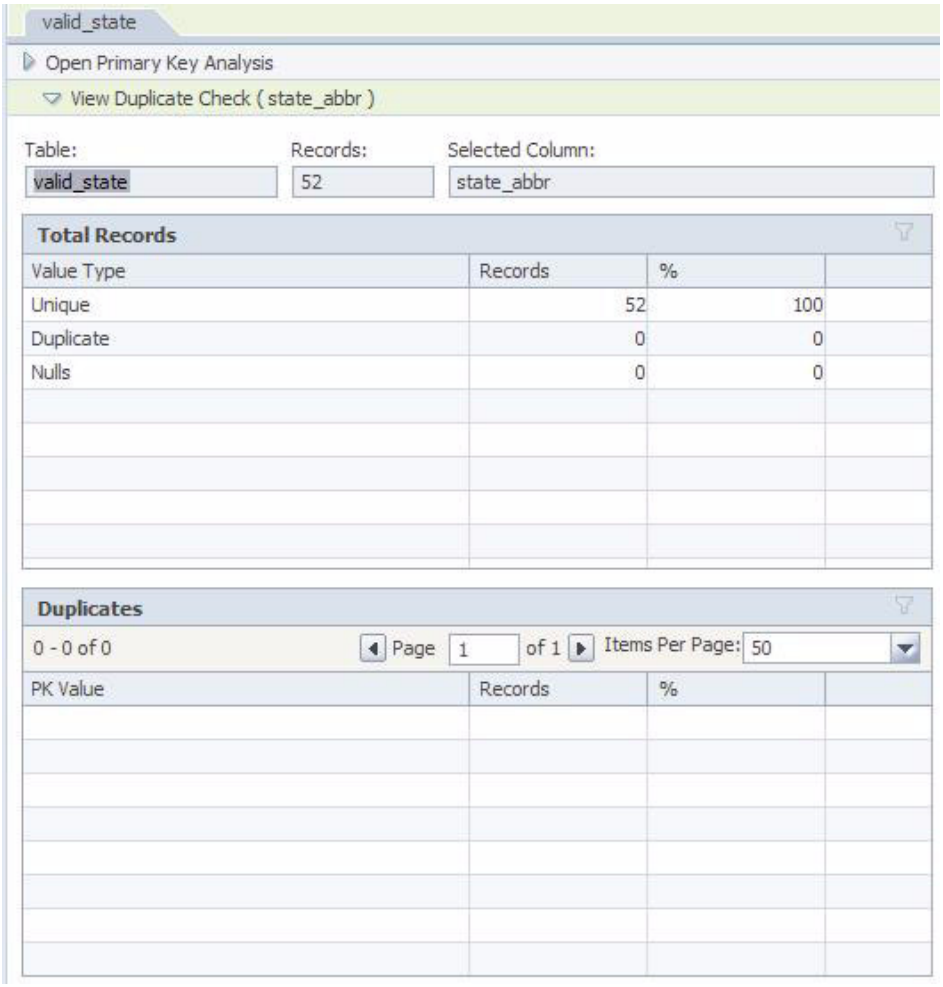

Figure 26-17 Open Primary Key Analysis -> View Duplicate Check.

Here the Valid_State table forms an odd example because every column is unique; symbol tables, as they are sometimes called, have two or more unique columns; the code column proper, and then any short name columns or related.

d. When finished touring, Click the Primary Key Status -> Accept Primary Key Status for the Valid_State table, State_Abbr column.

   If you choose another column for the primary key to this table, then you will be unable to form a foreign key relationship; foreign tables relate to State_Abbr, not the State_Name column.

4. Perform multi-column primary key analysis-

   Where the pre-requisite first step of column analysis has already been performed, allowing for all of the up front requirements supporting single column primary key analysis, multi-column primary key analysis requires additional processing. In effect, knowing the uniqueness of single columns in no way prepares you to know the uniqueness of combinations of columns; that work has yet to be performed.

   Column analysis precedes this work, because column data types and related tasks allow for further, and more intelligent processing towards identifying a multi-column primary key.

   a. Open the Investigate -> Primary Key Analysis menu item, and navigate the Data Sources display to Valid_Stock table.

Note: Legally, any table name would work in our example here. To allow for a more accurate modelling of activities, it might be best to choose a table with zero single column primary key candidates.

   b. From the menu on the right, Click, Open Primary Key Analysis.

      Again, there should be zero single column primary key candidates offered via the user interface.

   c. Click on the TAB entitled, Multicolumn.

      Upon entering this screen, you will see a background image with the label, "To analyze multiple columns, click the analyze multiple columns button".

   d. Click the Analyze Multiple Columns button.

      You will receive a further prompt; for small sized tables, we always Click, Analyze Full Table.

      This request produces a full column listing from the table. Generally, we analyze numeric columns, short length character columns, etcetera. Generally, binary or large text fields are not common primary key column candidates.

   e. Highlight two or more columns; this is done via Shift-clicking or Control-clicking. The multi-column primary key for the Valid_Stock table contains the columns Stock_num and Manu_code.

   f. After highlighting two or more columns, Click Submit.

g. Proceed through to the completion of accepting the primary key candidate, similar to the manner in which we accepted a single column primary key.

   A completed example appears in Figure 26-18.

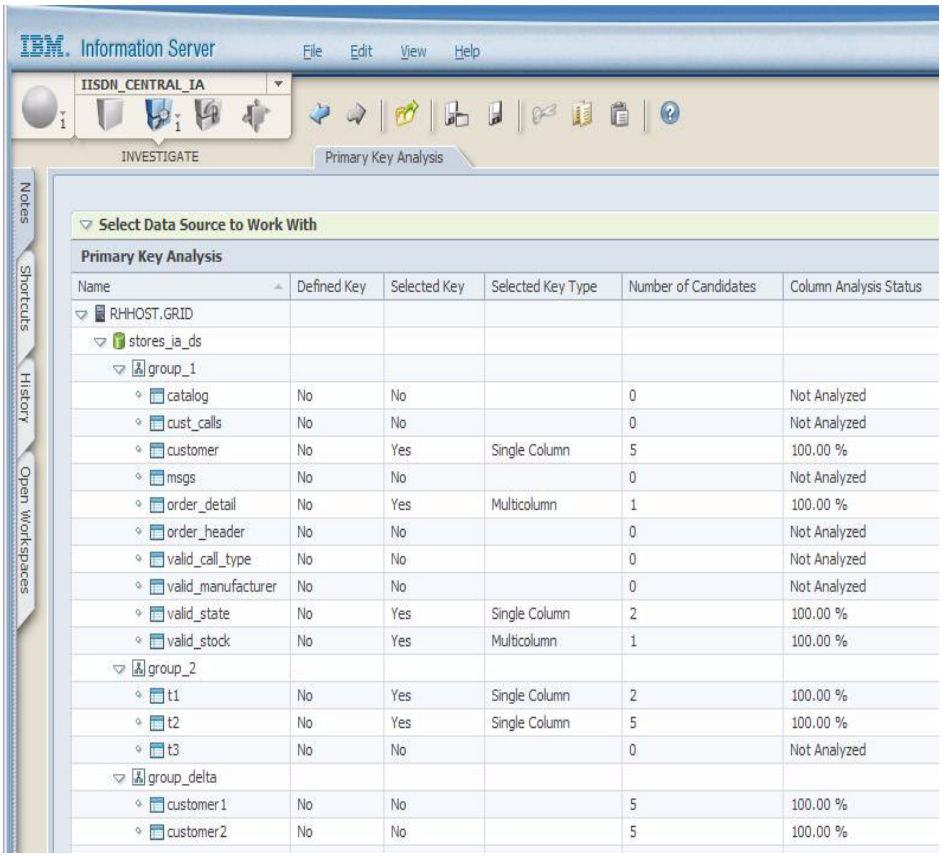

Figure 26-18 Primary key column status.

h. Complete primary key column analysis for all remaining tables in the Schema_1 schema.

Performing foreign key analysis, single and multi-column

Much like primary key analysis, we start our foreign key analysis example with the simpler single column task, and move forward to the multi-column task. In the sample database described in section 26.1 of this document, the Valid_State table has a primary key which is referenced by the Customer table.

5. Perform single column foreign key analysis-

   a. From the main menu, select Investigate -> Foreign Key Analysis

b. Navigate the Data Sources display to locate the Schema_1, and then the Valid_State and Customer tables.

   Foreign key analysis is done table by table (or higher), not column by column (not lower than the table level). Also, foreign key analysis specifies a relationship between two tables.

Note: As stated above, keys are a table level property, not a column level property. A foreign key, by definition, must relate to an existing primary key.

If foreign keys relate one table to another, why doesn't the Information Analyzer user interface help guide you to select two distinct tables when working with foreign keys?

First, while it is an uncommon use case, a single table may relate to itself, may be a foreign key table to itself. Hence, in this uncommon use case, you would select only one table. Generally this condition is referred to as a self-join.

Second, in these examples we are detailing how to go column by column or table by table, using the minimum amount of computer resource. If you are doing this work on a smallish or undersized laptop, you shouldn't be specifying to analyze hundreds of objects in one call.

In the real world, and on an appropriately sized box for the work at hand, we would specify to analyze hundreds of columns and tables in one call.

c. Highlight the Valid_State and Customer tables (use Control-Click to highlight both tables concurrently) and Click, Run Foreign Key Analysis from the menu on the right.

   This action produces the displays as shown in Figure 26-19.

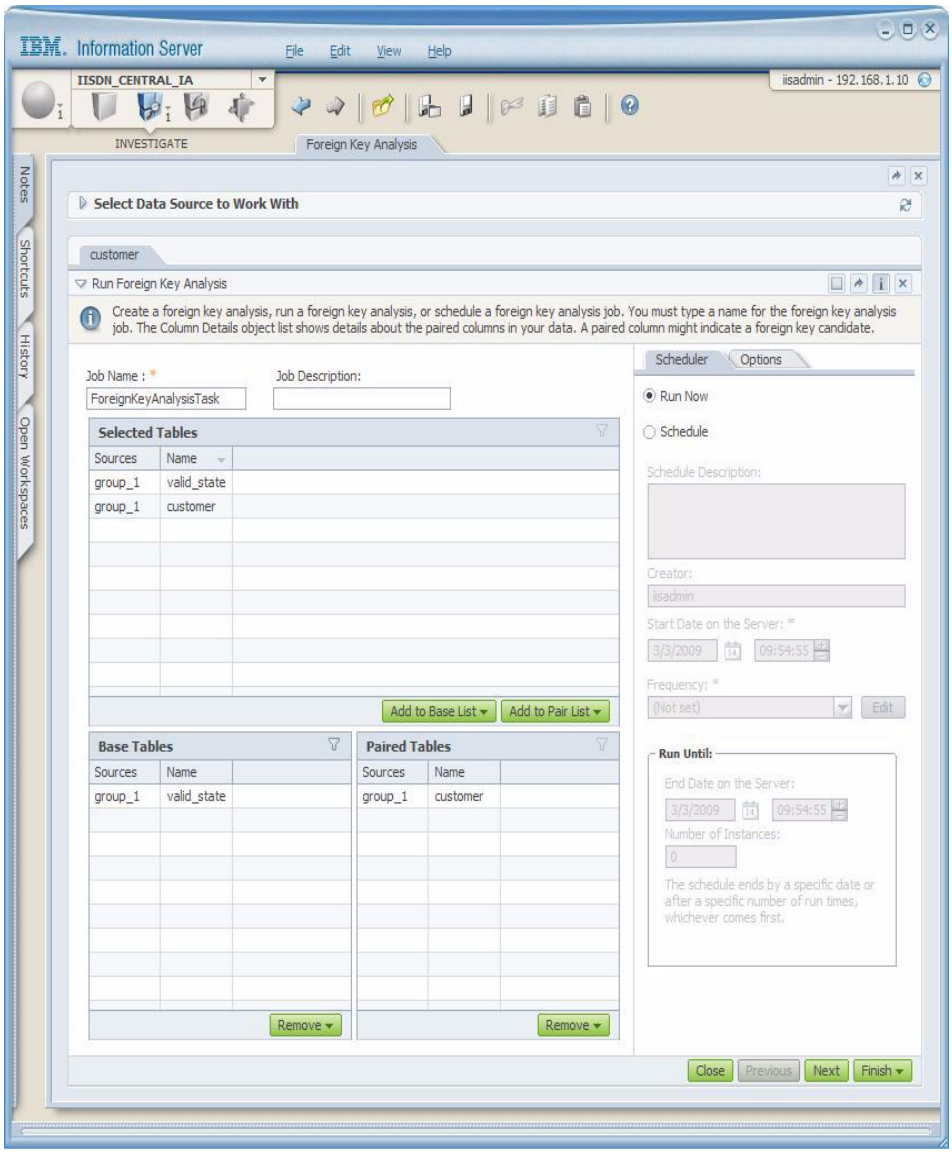

Figure 26-19 Investigate -> Foreign Key Analysis -> Run Foreign Key Analysis.

d. In Figure 26-19, manage the display to affect the following-

   i. In the area of the display entitled, Base Tables, manage the display so that Valid_State is the only table listed.

Base tables are those with a primary key, that other tables form a foreign key relationship to.

ii. In the area of the display entitled, Paired Tables, manage the display so that Customer is the only table listed.

Paired tables are those with foreign key constraints.

iii. Click Next.

This action produces the display as shown in Figure 26-20.

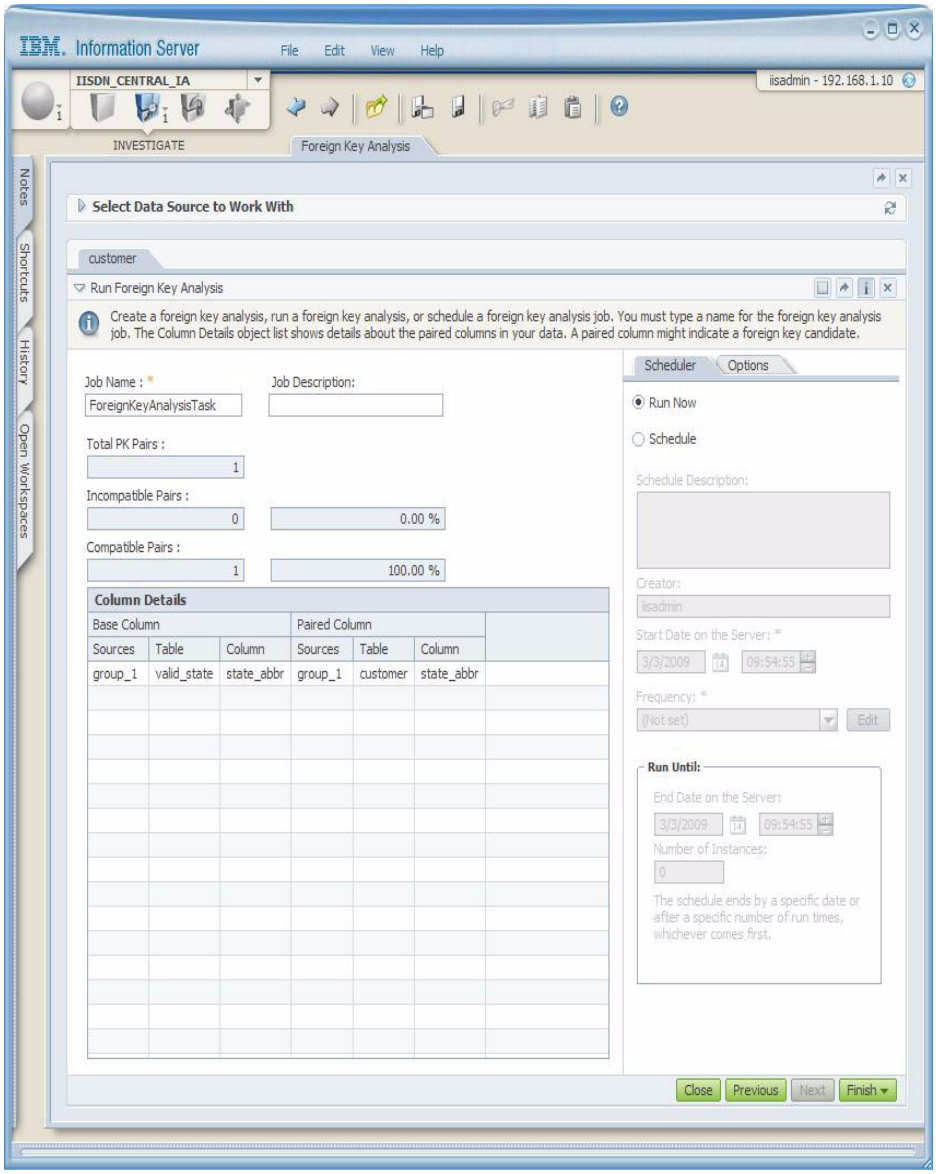

Figure 26-20 Foreign key analysis, column selection.

e. Figure 26-20 displays the column selection (pairings) screen.

   Here we see that the primary key column(s) from the base table primary key was matched with available column(s) of the paired table. This matching is done from knowledge gained during the column analysis

phase; meaning, Information Analyzer will not attempt to pair incompatible column data types.

Click, Finish -> Finish and Close.

This action will return you to the Investigate -> Foreign Key Analysis screen, with the associated analysis Job running in the background.

f. After the Job above has completed, the menu item entitled, Open Foreign Key Analysis will be available. Between the two tables, that with the primary key and heading the foreign key relationship (in this case, Valid_State), and the table making the foreign key reference (in this case, Customer), the table you Open Foreign Key Analysis for is Valid_State, the table to which other tables refer.

Highlight the Valid_State table, and Click Open Foreign Key Analysis from the menu on the right.

This action will produce the displays as shown in Figure 26-21.

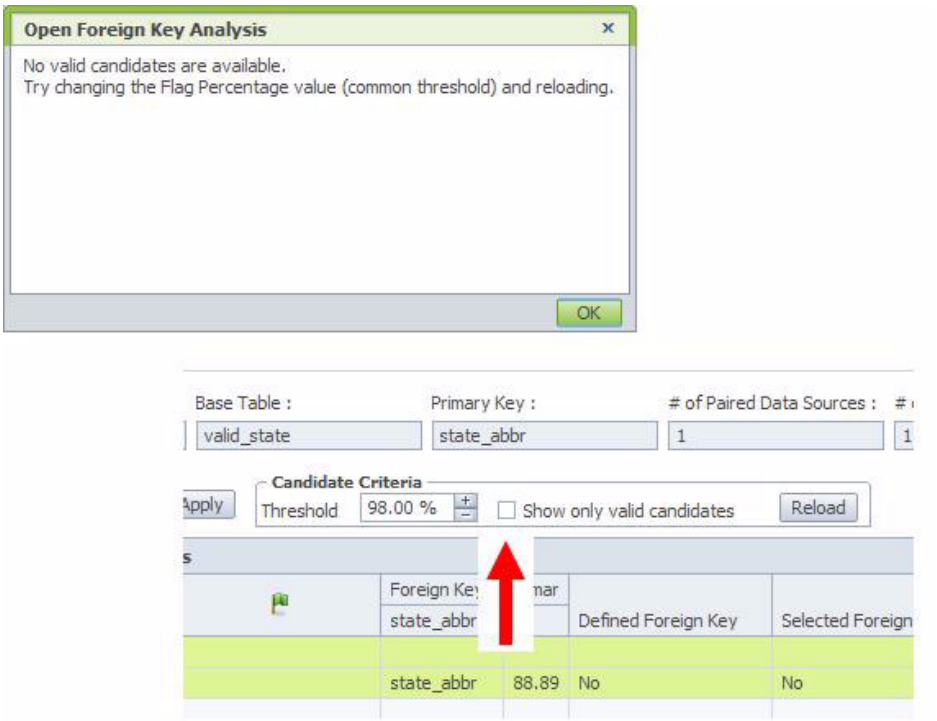

Figure 26-21 Open Foreign Key Analysis, setting threshold criteria.

g. In Figure 26-21, first we receive a warning dialog box. Our Valid_State table contains 50 or more states worth of data, but our detail table (in this

case, Customer), is only using a smallish subset of states; the parent/code table is heavy.

Click OK, to close the warning dialog box.

Then Un-check the box entitled, Show only valid (foreign key column) candidates. Example as shown in Figure 26-21.

This action should now display our matching foreign key column of Valid_State.State_Abbr -> Customer.State_Abbr.

Note: We have 50 state abbreviation in the parent code table (the primary key table), but reference many fewer states in the detail table (the foreign key table). This is uncommon, and hence, gives need to view all matching candidates, not just those above a given threshold.

h. Highlight the available foreign key candidate entry with a single Click, and select the menu item Referential Integrity -> Run Referential Integrity Analysis -> Submit -> Submit and Close, on the bottom of the screen.

i. If this Job proves out, you may then select the menu items for, Foreign Key Status -> Accept Foreign Key Status -> Close.

This action should record that this foreign key relationship has been identified within the Metadata Repository. Example as shown in Figure 26-22.

Select Data Source to Work With

Foreign Key

Name	Foreign Key Candidates	Column Analysis Status	Column Analysis Date	Defined Primary Key	Selected Primary Key	Defined Foreign Key
RHHOST.GRID						
stores_ia_ds						
group_1						
catalog	0		--	No	No	No
cust_calls	0		--	No	No	No
customer	0		2/4/2009 2:45:24 PM	No	Yes	No
msgs	0		--	No	No	No
order_detail	0		2/4/2009 2:38:43 PM	No	Yes	No
order_header	0		--	No	No	No
valid_call_type	0		--	No	No	No
valid_manufacturer	0		--	No	No	No
valid_state	1		3/2/2009 4:39:46 PM	No	Yes	No
valid_stock	2		2/4/2009 2:32:10 PM	No	Yes	No
group_2						

Figure 26-22 Investigate -> Foreign Key Analysis.

6. Perform multi-column foreign key analysis-

Much like a single column foreign key analysis, defining a multi-column foreign key has the dependency that a compatible primary key exist for it to refer to; in this case, a multi-column primary key must exist in your example.

In our sample database, a multi-column primary key exists in Valid_Stock. The table we will foreign key from is Order_Detail.

a. From the main menu, select Investigate -> Foreign Key Analysis

b. Navigate the Data Sources display to locate the Schema_1, and then the Valid_Stock and Order_Detail tables.

   Foreign key analysis is done table by table (or higher), not column by column (not lower than the table level). Also, foreign key analysis specifies a relationship between two tables.

c. Highlight the Valid_Stock and Order_Detail tables (use Control-Click to highlight both tables concurrently) and Click, Run Foreign Key Analysis from the menu on the right.

d. In the area of the display entitled, Base Tables, manage the display so that Valid_Stock is the only table listed.

   Base tables are those with a primary key, that other tables form a foreign key relationship to.

e. In the area of the display entitled, Paired Tables, manage the display so that Order_Detail is the only table listed.

   Paired tables are those with foreign key constraints.

f. Click Next.

g. This action produces the display as shown in Figure 26-23.

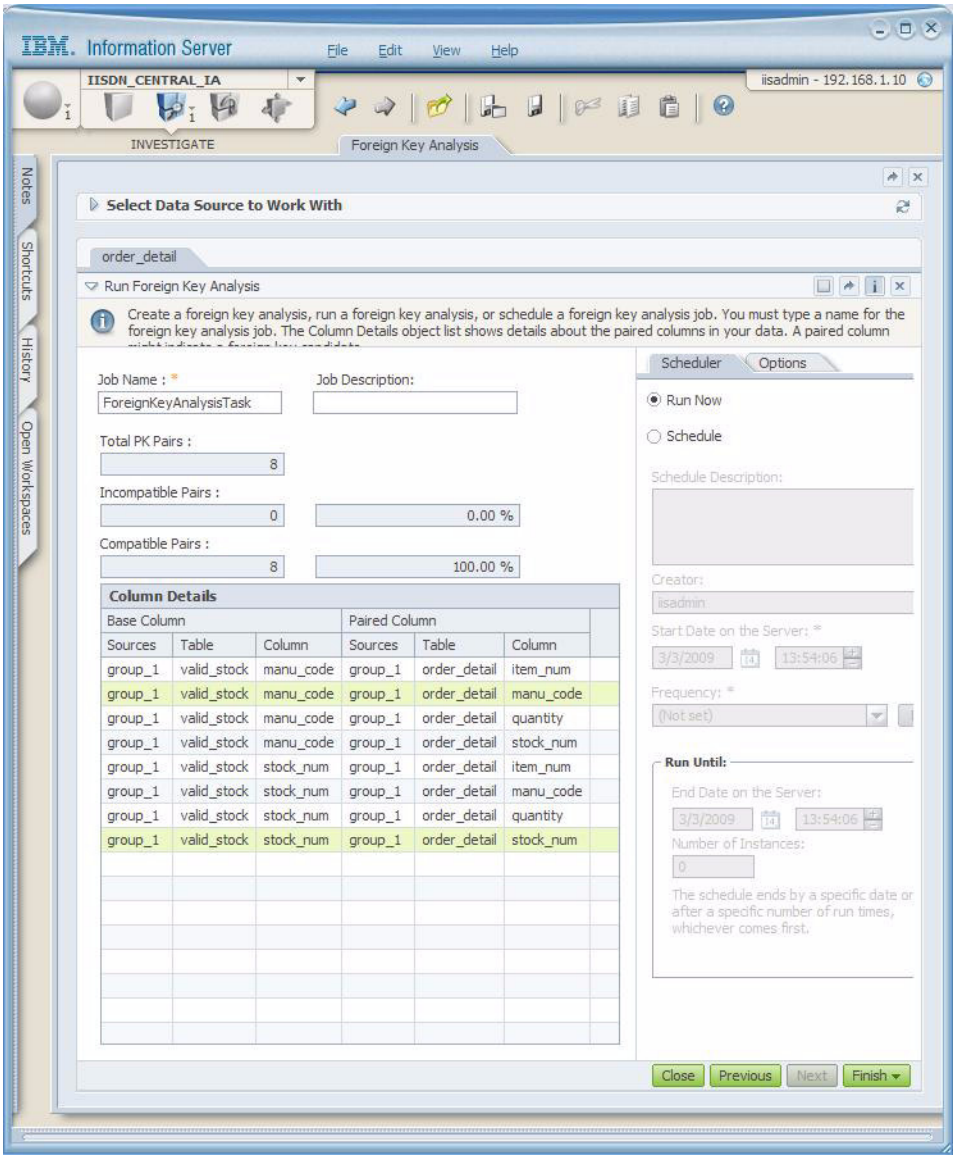

Figure 26-23 Column selection for multi-column foreign key analysis.

h. In Figure 26-23, we are presented with each pair of columns that may participate in the multi-column foreign key relationship. While we may select every pairing in this list, we will save execution time and resource if we are more selective.

In the list of available choices, highlight only the Valid_Stock.Manu_Code to Order_Detail.Manu_Code column, and the Valid_Stock.Stock_Num to Order_Detail.Stock_Num column. (Again, use Control-Click to do this.)

i. Select the menu item, Finish -> Finish and Close.

As before, this will submit your (multi-column foreign key) analysis Job to the background. Continue when ready.

j. When the submitted Job above is complete, highlight the Valid_Stock table, and click the menu item entitled, Open Foreign Key Analysis.

This screen is displaying the possible foreign key pairs, from the Valid_Stock table to a given target (paired) Table.

k. Select the entry for the Order_Detail table, with the Stock_Num and Manu_Code columns. Select the menu item entitled, Referential Integrity -> Run Referential Integrity Analysis -> Submit -> Submit and Close.

l. When the above is complete, select the menu item entitled, Foreign Key Status -> Accept Foreign Key Status.

A completed example appears in Figure 26-24.

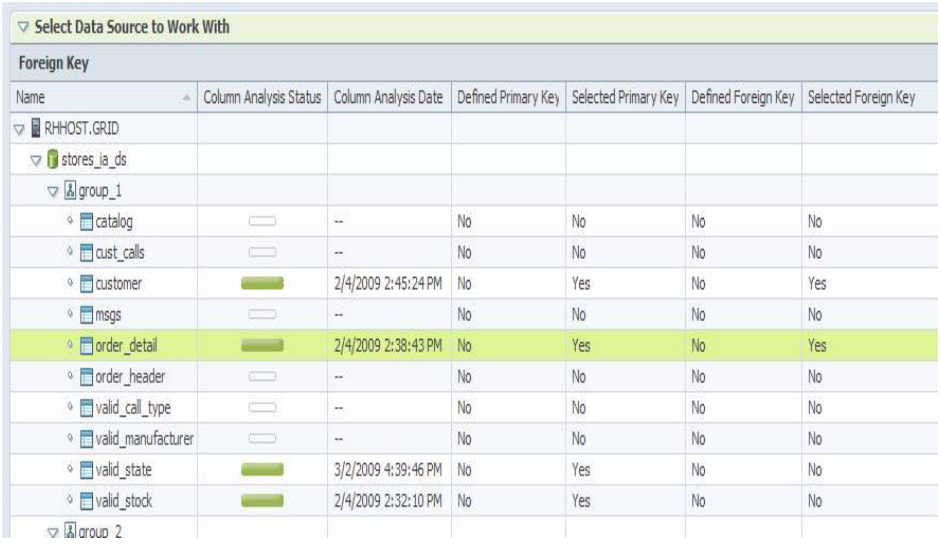

Figure 26-24 Foreign key column status.

m. Complete foreign key column analysis for all remaining tables in the Schema_1 schema.

26.4 In this document, we reviewed or created:

As part of a new multiple edition series, we examined the (IBM) InfoSphere Information Server (IIS), Information Analyzer component. Specifically we configured this component, and performed the Information Analyzer component activities of; column analysis, primary key analysis, and foreign key analysis. The Information analyzer component activities of (cross) domain analysis and base line analysis come in a document to follow.

Information Analyzer is a hugely productive component to Information Server. Besides the automatic team communication activities it supports, the activities of Information Analyzer are automated; normally you'd be writing tons of complicated SQL and spreadsheet analysis to do this work. And, all of the knowledge Information Analyzer gathers is automatically placed in a shared metadata repository, where it benefits the other components to Information Server, promoting higher productivity, data quality, and an audit trail of end to end data lineage.

Persons who help this month.

Walter Crockett Jr, Steve Fazio, Bill Shubin, James Wilson, and Carrie Klapheke.

Additional resources:

The IBM InfoSphere Information Server, Information Analyzer component, product tutorial.

Legal statements:

Special attributions:

The listed trademarks of the following companies require marking and attribution:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Microsoft trademark guidelines

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel trademark information

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

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

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

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

