talend*
*open integration solutions

Talend Open Studio for Data Quality

User Guide

5.2.1

Adapted for v5.2.1.

Copyleft

Notices

Table of Contents

List of Tables

Preface

1. General information

1.1. Purpose

This User Guide explains how to manage *Talend Open Studio for Data Quality* functions in a normal operational context.

Information presented in this document applies to release **5.2.1** of *Talend Open Studio for Data Quality*.

1.2. Audience

This guide is for business users, database administrators and data analysts in charge of checking the quality of data and collecting statistics and information about that data.

The layout of GUI screens provided in this document may vary slightly from your actual GUI.

1.3. Typographical conventions

This guide uses the following typographical conventions:

- text in **bold:** window and wizard buttons and fields, keyboard keys, menus and menu options,
- text in **[bold]:** window, wizard and dialog box titles,
- text in `courier`: system parameters selected by the user,
- text in *italics*: file, schema, column, row and variable names,
- The  icon indicates an item that provides additional information about an important point. It is also used to add comments related to a table or a figure,
- The  icon indicates a message that gives information about the execution requirements or recommendation type. It is also used to refer to situations or information the end user needs to be aware of or pay special attention to.

2. Feedback and Support

Your feedback is valuable. Do not hesitate to give your input, make suggestions or requests regarding this documentation or product and find support from the **Talend** team, on **Talend**'s Forum Website at:

http://talendforge.org/forum

Chapter 1. Overview

This chapter introduces data profiling as the process of examining the data available in different data sources such as databases, files or Master Data Management (MDM) servers.

1.1. Why profiling data

Data profiling is the process of examining the data available in different data sources (for example, databases, files and MDM servers) and collecting statistics and information about this data. Data profiling helps to assess the quality level of the data according to defined set goals.

If data is of a poor quality, or managed in structures that cannot be integrated to meet the needs of the enterprise, business processes and decision-making suffer.

Compared to manual analysis techniques, data profiling technology improves the enterprise ability to meet the challenge of managing data quality and to address the data quality challenges faced during data migrations and data integrations.

1.2. About Talend data quality

The following sections introduce **Talend** data quality and list its key features.

1.2.1. What is Talend data quality

This data profiling tool allows you to identify potential problems before beginning data-intensive projects such as data integration.

The data profiler centralizes several elements including a:

- data profiler, for more information about the data profiler, see appendix *The studio management GUI*.
- data explorer, for more information about the data explorer, see appendix *Data Explorer management GUI*.
- pattern manager, for more information about the pattern manager, see section *Patterns and indicators*.
- metadata manager, for more information about the metadata manager, see section *Metadata repository*.

1.2.2. Core features

This section describes the basic features of this data profiling management solution.

1.2.2.1. Metadata repository

Using this solution, you can connect to databases, files and MDM servers to analyze their structure (catalogs, schemas and tables), and stores the description of their metadata in its metadata repository. You can then use this metadata to set up metrics and indicators.

For more information, see section *Connecting to a database* and chapter *Table analyses*.

1.2.2.2. Patterns and indicators

Patterns are sets of strings against which you can define the content, structure and quality of high complex data. The **Profiling** perspective of the studio lists two types of patterns: regular expressions, which are predefined regular patterns, and SQL patterns which are the patterns you add using `LIKE` clauses.

For more information about patterns, see section *Patterns*.

Indicators are the results achieved through the implementation of different patterns. They can represent the results of data matching and different other data-related operations. The **Profiling** perspective of the studio lists two types of indicators: system indicators, a list of predefined indicators, and user-defined indicators, a list of those defined by the user.

For more information about indicators, see section *Indicators*.

Chapter 2. Getting started with Talend data quality

This chapter introduces **Talend** data quality and guides you through the basics for launching the studio .

This chapter explains the typical sequence of profiling data using the studio and many other important miscellaneous subjects.

Before starting data profiling management procedures, you need to be familiar with the Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

2.1. Working principles of data quality

From the **Profiling** perspective of the studio, you can examine the data available in different data sources and collect statistics and information about this data.

A typical sequence of profiling data using the studio involves the following steps:

1. Connecting to a data source including databases, a Master Data Management (MDM) servers and delimited files or excel files in order to be able to access the tables and columns on which you want to define and execute analyses. For more information, see chapter *Before you begin profiling data*.

2. Defining any of the available data quality analyses including database content analysis, column analysis, table analysis, redundancy analysis, correlation analysis, etc. These analyses will carry out data profiling processes that will define the content, structure and quality of highly complex data structures. The analysis results will be displayed graphically next to each of the analysis editors, or in more detail in the **Analysis Results** view.

While you can use all analyses types to profile data in databases, you can only use **Column Analysis** and **Column Set Analysis** to profile data in a delimited or excel file and to profile master data on MDM servers.

2.2. Launching the studio

To open your studio for the first time, do the following:

1. Unzip the Talend studio zip file and, in the folder, double-click the executable file corresponding to your operating system.

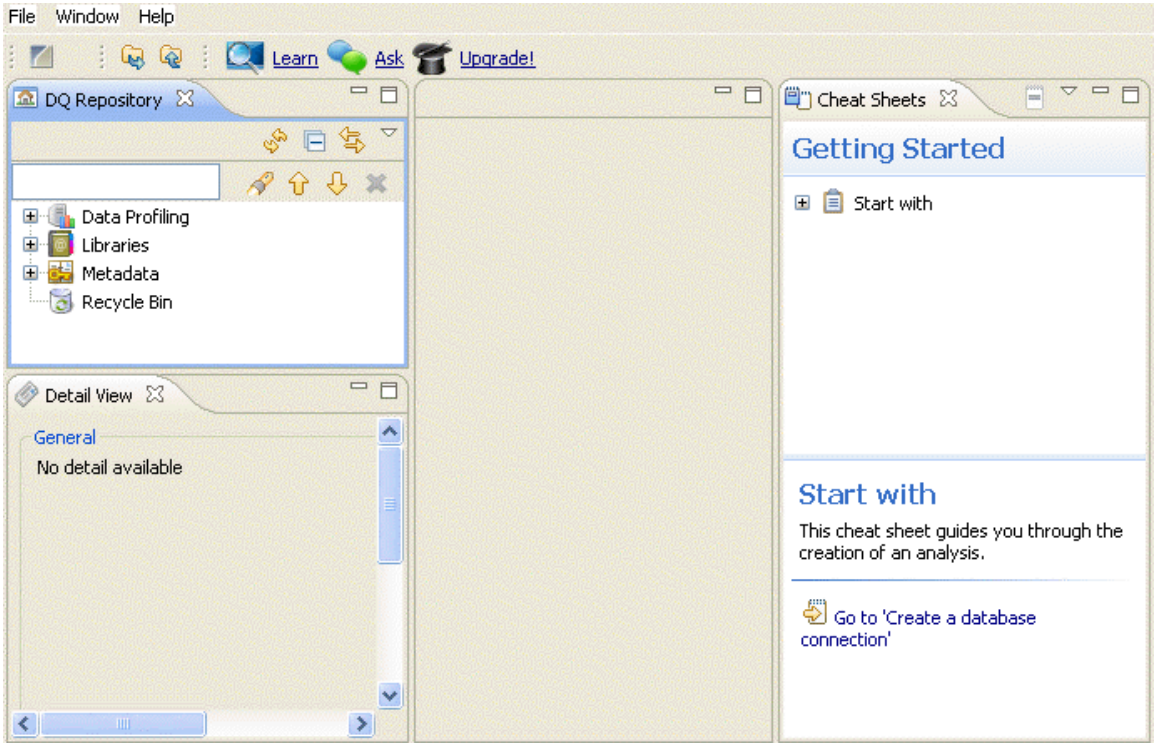
The studio zip archive contains binaries for several platforms including Mac OS X and Linux/Unix.

2. In the **[License]** window that is displayed, read and accept the terms of the license agreement to proceed to the next step.

   A registration window is displayed.

3. If required, follow the instructions provided to join **Talend** community or click **Register later** to open a welcome window.

4. In the welcome window, click **Start now** to open the studio.



The studio open on the **Profiling** perspective by default.

From this window, you can have access to the perspectives of other applications integrated within the studio. For more information, see section *Multi-perspective approach*.

You can now start to profile your data by creating your own analyses or importing already created ones.

For more information about creating new analyses, see section *Working principles of data quality*.

For more information about importing analyses and data quality items created in other studios, see section *Importing data profiling items or projects* and section *Upgrading projects items from older versions*.

2.3. Important features and configuration options

This section details some important information about analysis editors, the error log view and the help context embedded in your studio.

2.3.1. Defining the maximum memory size threshold

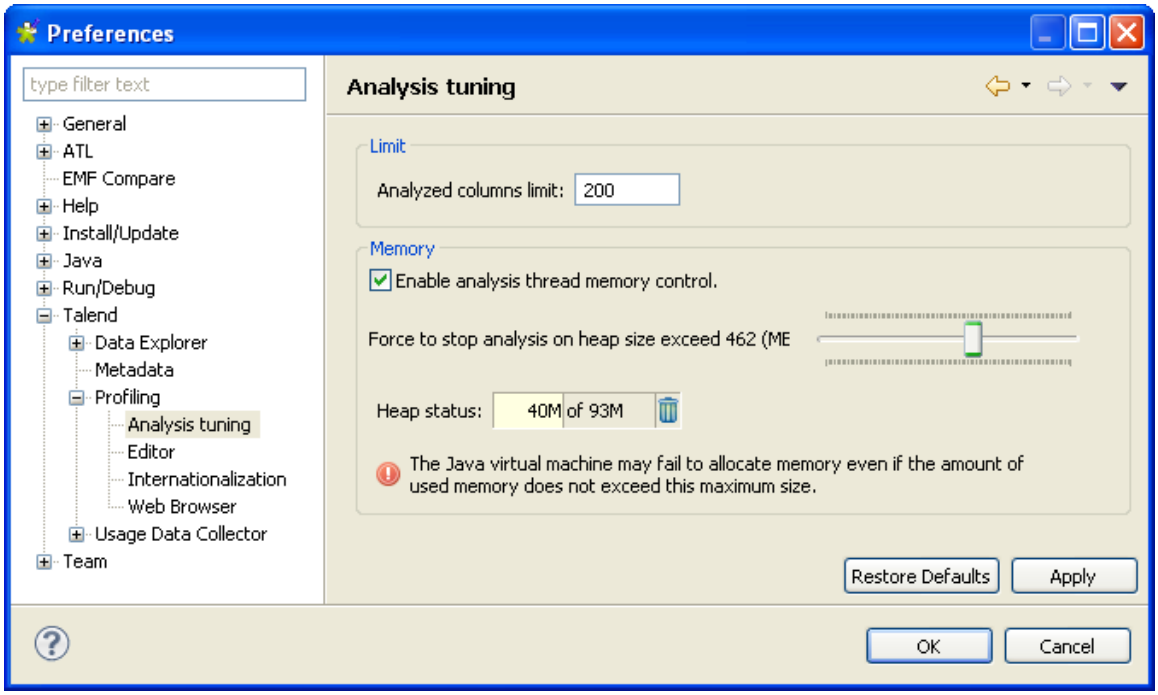
From the studio, you can control memory usage when using the Java engine to run two types of analyses: column analysis and the analysis of a set of columns.

Why would you like to set a memory limit when running such analyses? If you use column analysis or column set analysis to profile very big sets of data or data with many problems, you may run out of memory and end up with a Java heap error. By defining the maximum memory size threshold for these analyses, the Studio will stop

the analysis execution when the memory limit size is reached and provide you with the analysis results that were measured on the data before the analysis execution was terminated by the memory limit size.

To define the maximum memory size threshold, do the following:

1. On the menu bar, select **Window > Preferences** to display the **[Preferences]** window.
2. Expand **Talend > Profiling** and select **Analysis tuning**.



3. In the **Memory** area, select the **Enable analysis thread memory control** check box.
4. Move the slider to the right to define the memory limit at which the analysis execution will be stopped.

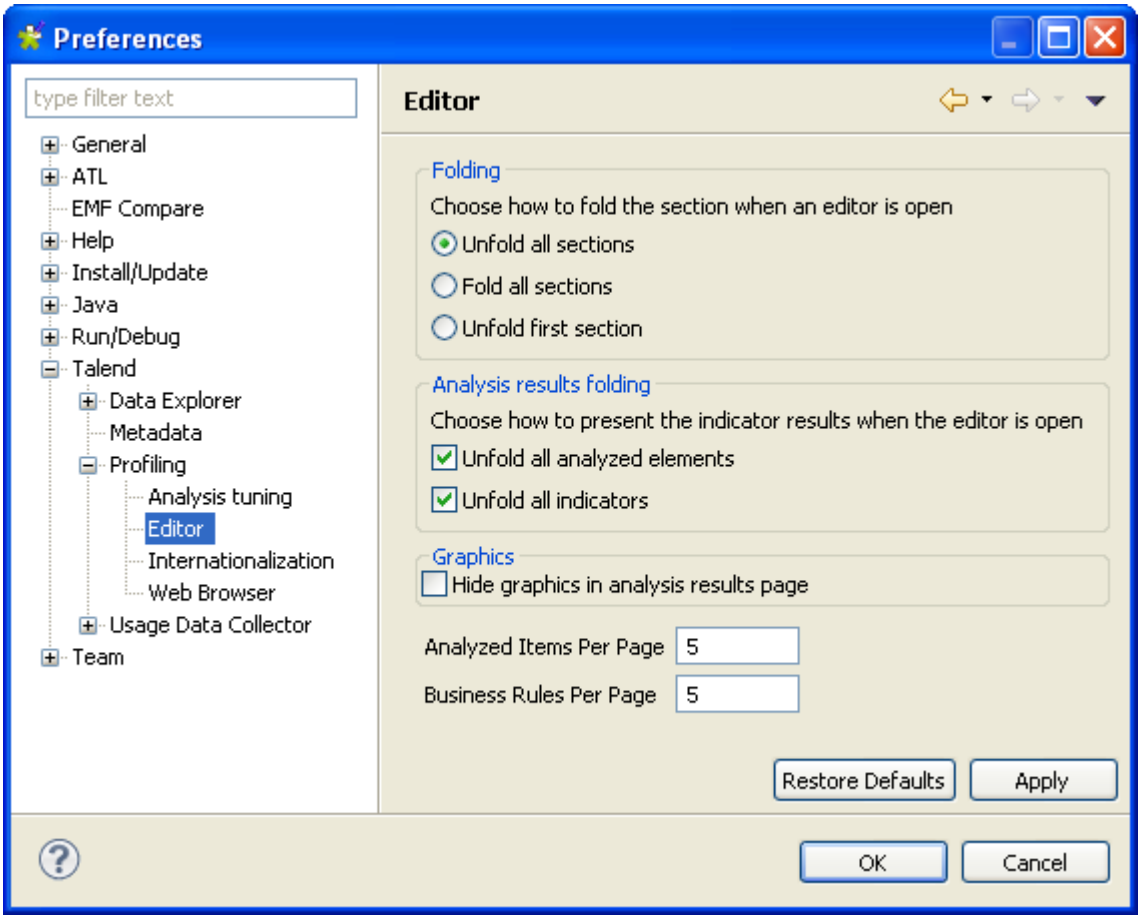
The execution of any column analysis or column set analysis will be stopped if it exceeds the allocated memory size. The analysis results given in the Studio will cover the data analyzed before the interruption of the analysis execution.

2.3.2. Setting preferences of analysis editors and analysis results

You can decide once for all what sections to fold by default when you open any of the connection or analysis editors. It also offers the possibility to set up the display of all analysis results and whether to show or hide the graphical results in the different analysis editors.

To set the display parameters for all editors, do the following:

1. On the menu bar, select **Window > Preferences** to display the **[Preferences]** window.
2. Expand **Talend > Profiling** and select **Editor**.



3. In the **Folding** area, select the check box(es) corresponding to the display mode you want to set for the different sections in all the editors.

4. In the **Analysis results folding** area, select the check boxes corresponding to the display mode you want to set for the statistic results in the **Analysis Results** view of the analysis editor.

5. In the **Graphics** area, select the **Hide graphics in analysis results page** option if you do not want to show the graphical results of the executed analyses in the analysis editor. This will optimize system performance when you have so many graphics to generate.

6. In the **Analyzed Items Per Page** field, set the number for the analyzed items you want to group on each page.

7. In the **Business Rules Per Page** field, set the number for the business rules you want to group in each page.

You can always click the **Restore Defaults** tab on the **[Preferences]** window to bring back the default values.

8. Click **Apply** and then **OK** to validate the changes and close the **[Preferences]** window.

While carrying on different analyses, all corresponding editors will open with the display mode you set in the **[Preferences]** window.

2.3.3. Displaying and hiding the help content

Your studio provides you with cheat sheets that you can use as a quick reference that guides you through all common tasks in data profiling.

You can also have access to a help panel that is attached to all wizards used in the studio to create the different types of analyses or to set thresholds on indicators.

2.3.3.1. Cheat sheets

When you open the **Profiling** perspective of the studio for the first time, the cheat sheet panel will open by default. However, this view is closed when you switch away from the **Profiling** perspective.

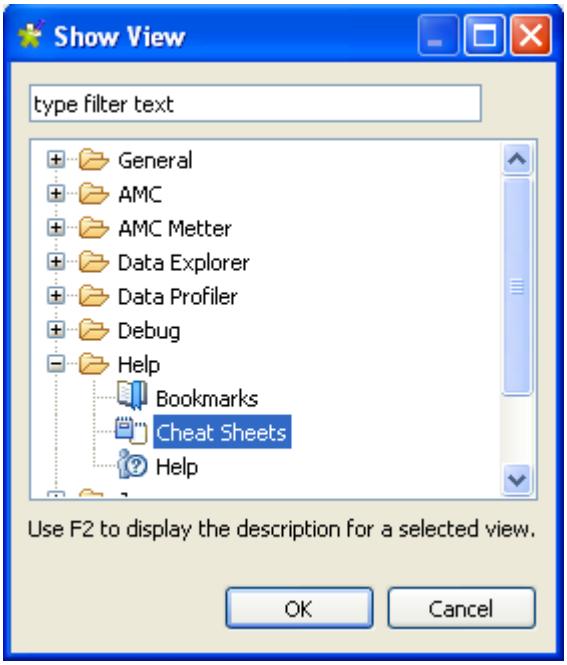
If you close the cheat sheet panel in the **Profiling** perspective of the studio, it will be always closed anytime you switch back to this perspective until you open it manually.

To display the cheat sheets, do one of the following:

1. Either:

   - press the **Alt+Shift+Q** and then **H** shortcut keys, or,
   - select **Window > Show View** from the menu bar.

   The **[Show View]** dialog box opens.



2. Expand the **Help** folder and then select **Cheat Sheets**.
3. Click **OK** to close the dialog box.

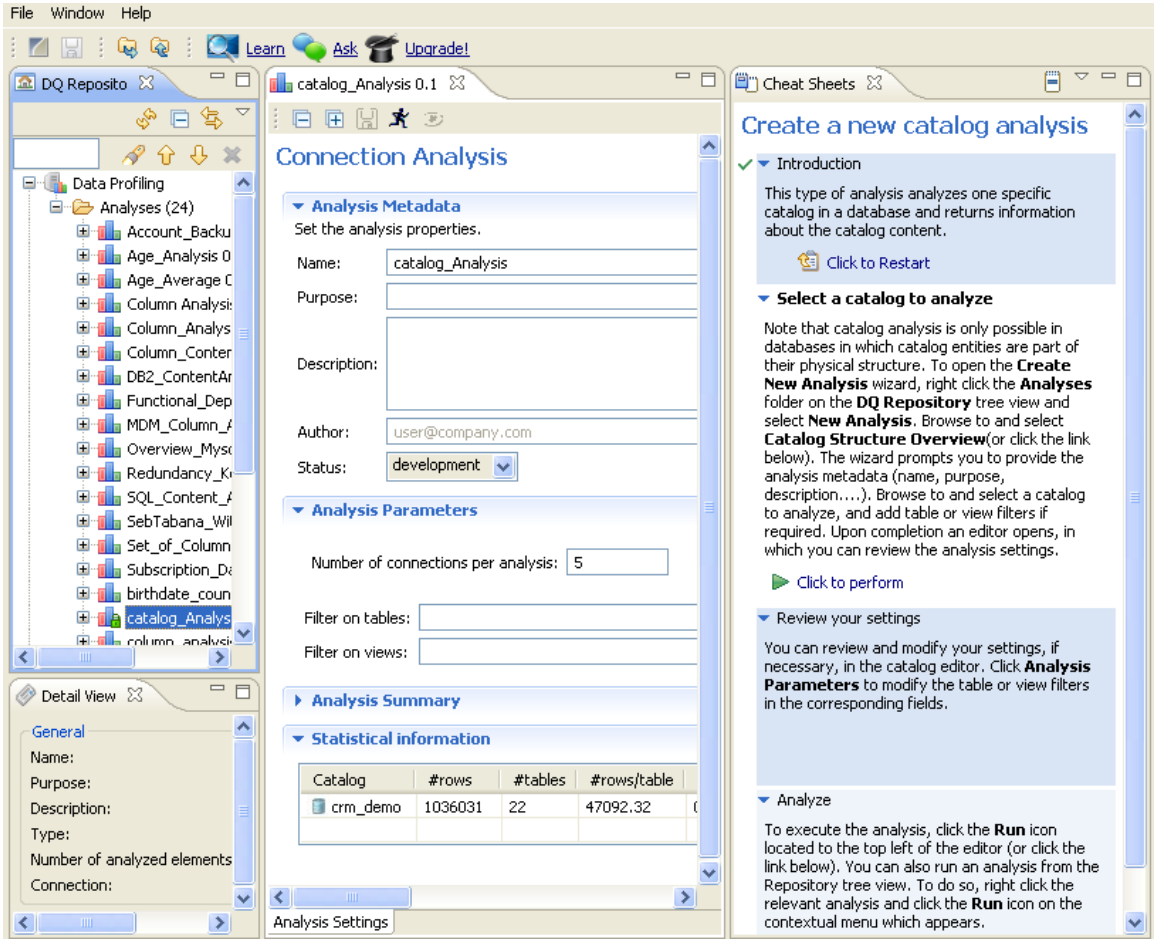
Or,

1. Select **Help > Cheat Sheets** from the menu bar. The **[Cheat Sheet Selection]** dialog box opens.

   You can also press the **Alt+H** shortcut keys to open the **Help** menu and then select **Cheat Sheets**.

2. Expand the **Help** folder and then select **Cheat Sheets**.
3. Select the cheat sheet you want to open in the studio and then click **OK** to close the dialog box.

   The selected cheat sheet opens in the studio main window. Use the local toolbar icons to manage the display of the cheat sheets.



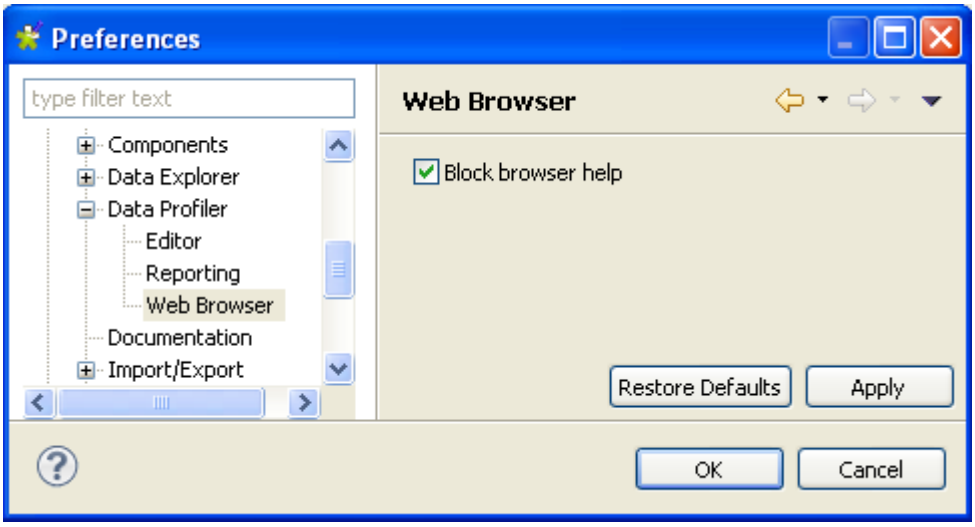
2.3.3.2. Help panel

The help panel attached to the analysis wizards is hidden by default.

To display the help panel in any of the wizards used in the Studio, do the following:

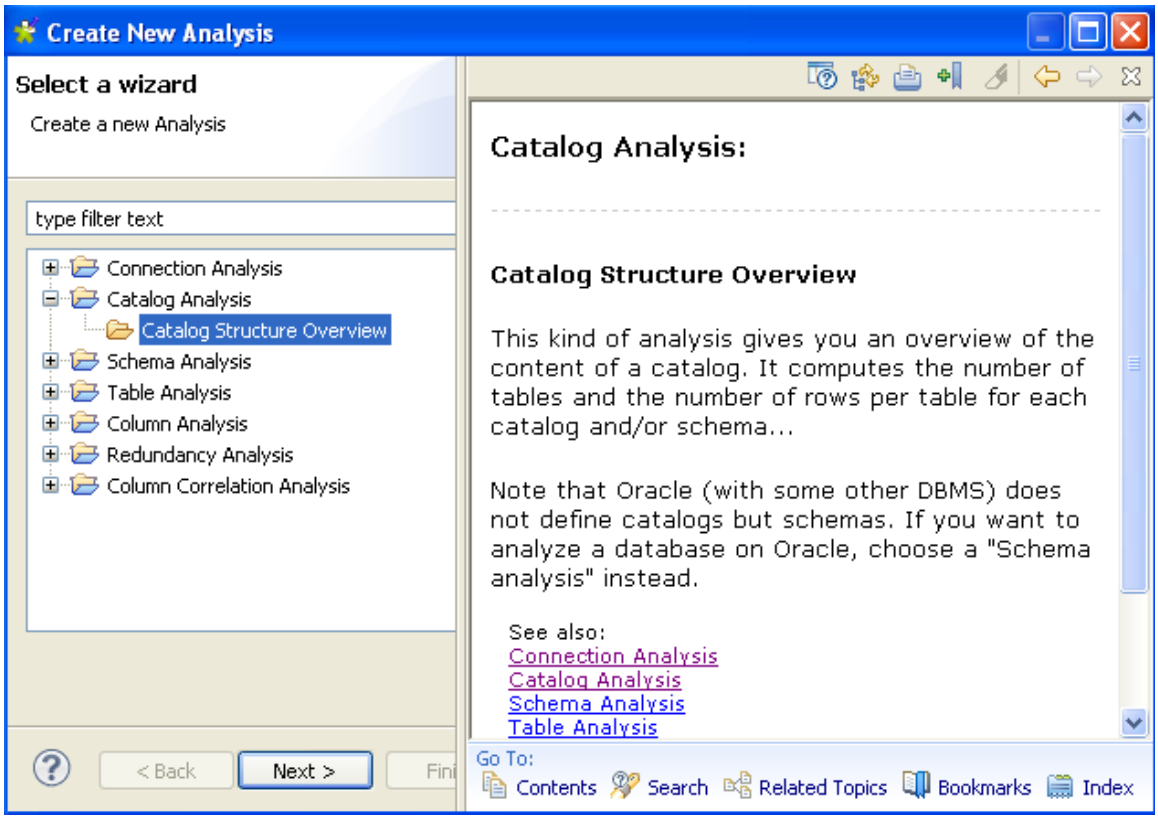
1. Select **Window > Preferences > Talend > Profiling > Web Browser**.

   The **[Web Browser]** view opens.



2. Clear the **Block browser help** check box and then click **OK** to close the dialog box.

   All the wizards in the studio will display with the help panel.



2.3.4. Displaying the error log view and managing log files

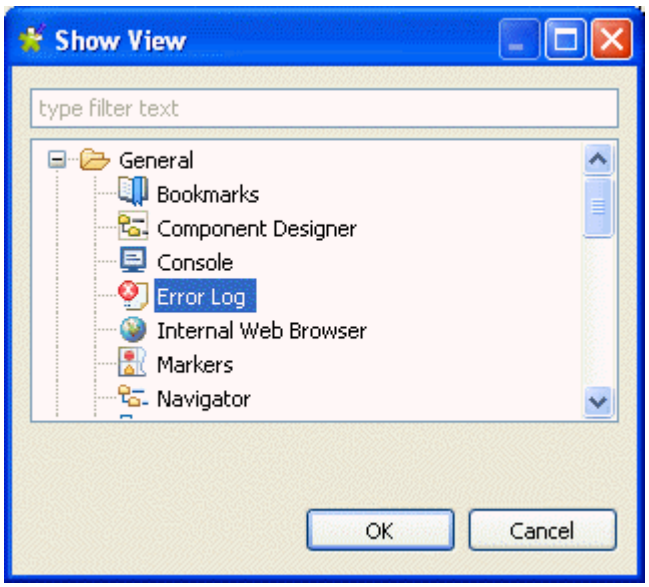
The studio provides you with very comprehensive log files that maintain diagnostic information and record any errors that are encountered in the data profiling process. The error log view is the first place to look when a problem occurs while profiling data, since it will often contain details of what went wrong and how to fix it.

To display the error log view in the Studio, do one of the following:

1. Either:

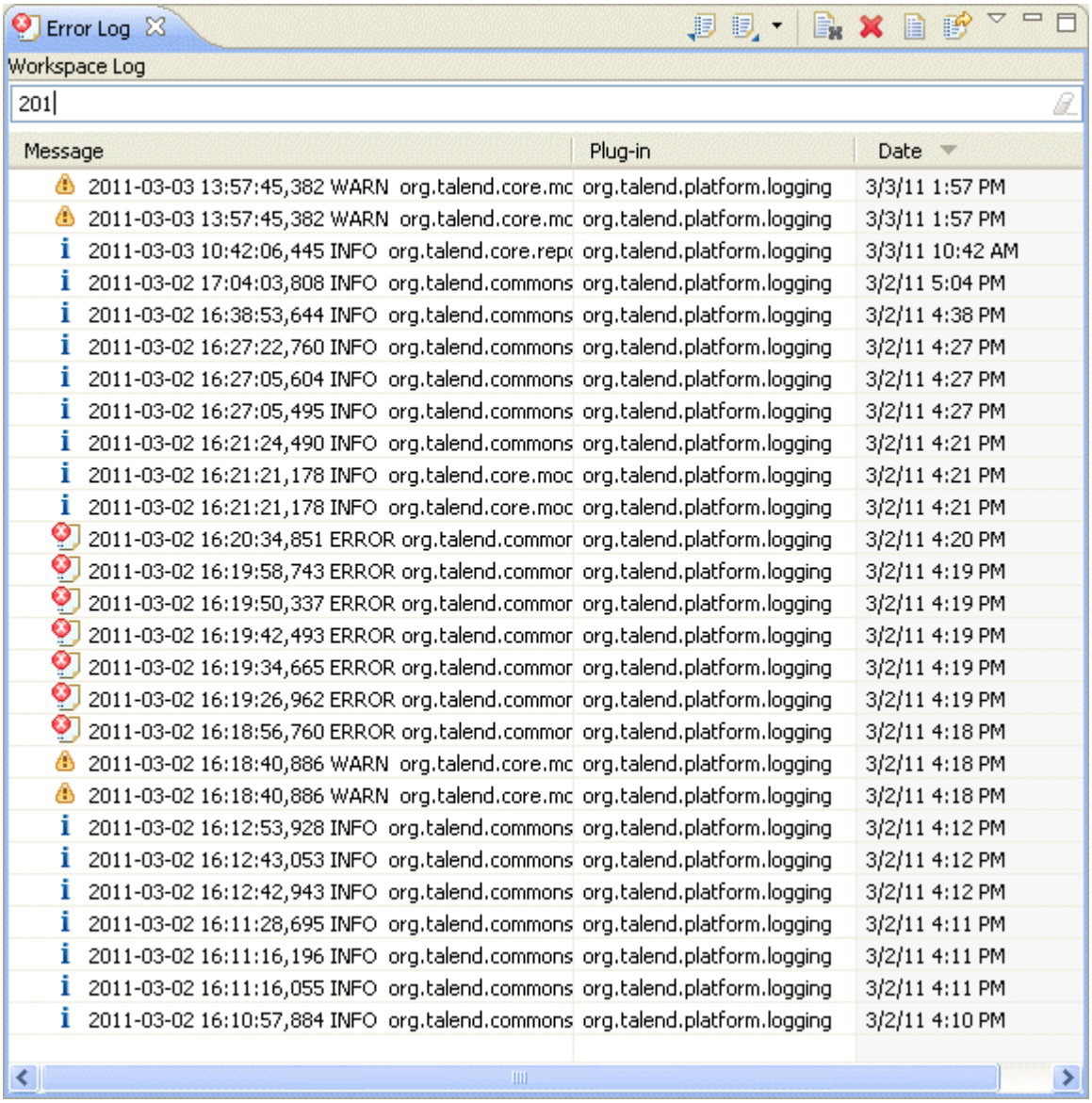
   - press the **Alt+Shift+Q** and then **L** shortcut keys, or,
   - select **Window > Show View** from the menu bar.

   The **[Show View]** dialog box opens.



2. Expand the **General** folder and select **Error Log**.
3. Click **OK** to close the dialog box.

   The **Error Log** view opens in the studio.

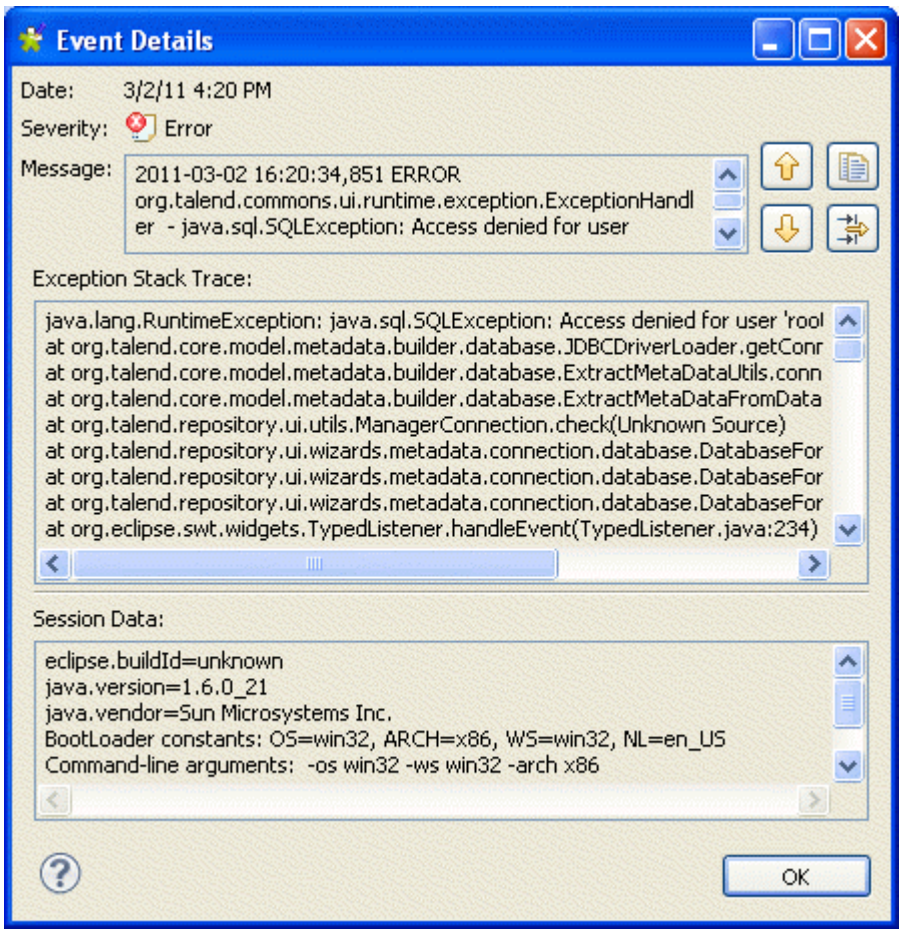


The filter field at the top of the view enables you to do dynamic filtering, i.e. as you type your text in the field, the list will show only the logs that match the filter.

You can use icons on the view toolbar to carry out different management options including exporting and importing the error log files.

Each error log in the list is preceded by an icon that indicates the severity of the log: for errors, for warnings and for information.

4. Double-click any of the error log files to open the **[Event Detail]** dialog box.



5. If required, click the icon in the **[Event Detail]** dialog box to copy the event detail to the Clipboard and then paste it anywhere you like.

2.3.5. Opening new editors

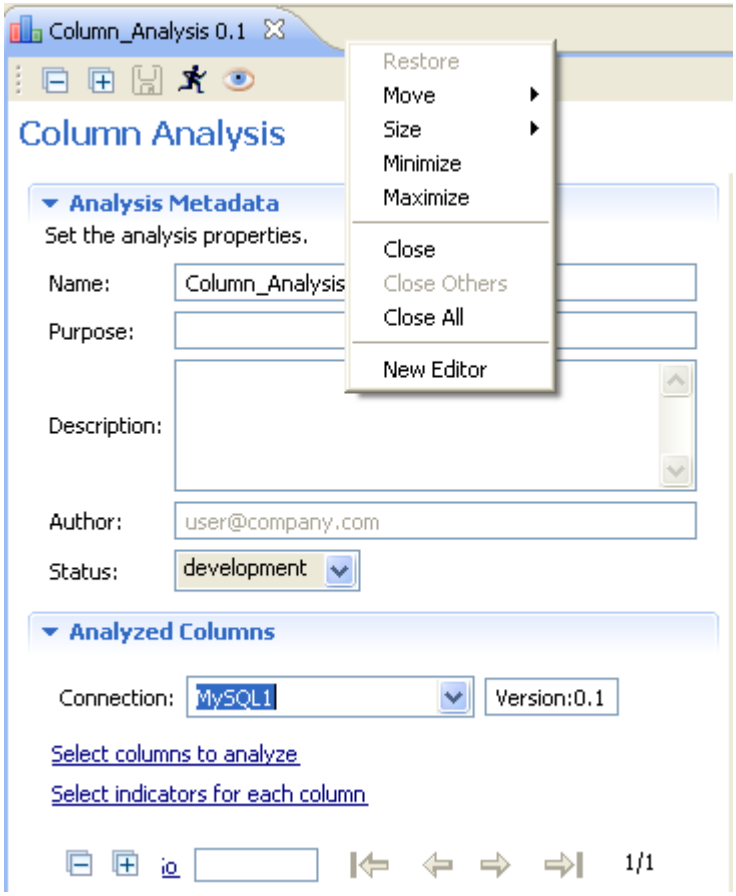
It is possible to open new analysis or SQL editors in the **Profiling** and **Data Explorer** perspectives respectively. You can either open a duplicate of the already open editor with the same analysis parameters or SQL query, or you can open a completely new empty editor.

Prerequisite(s): An analysis editor or an SQL query editor is open in the **Profiling** perspective of the studio.

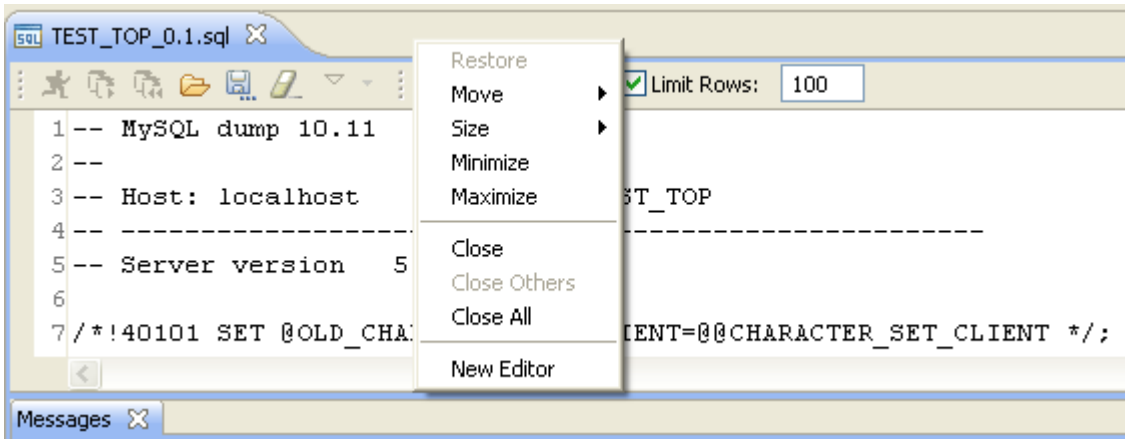
To open a duplicate of the already open editor, do the following:

1. In the open analysis or SQL editor, right-click the editor title tab.

   In the analysis editor:



In the SQL editor:



2. From the contextual menu, select **New Editor**.

   A new analysis or SQL editor opens on the same analysis metadata and parameters or on the same SQL query. The new editor will be an exact duplicate of the initial one.

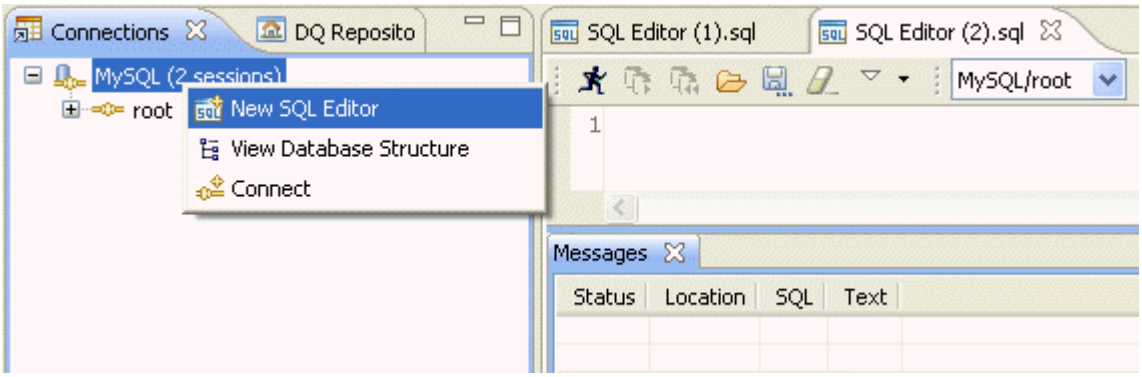
To open an empty new analysis editor, do the following:

1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.
2. Right-click the **Analysis** folder and select **New Analysis**.

To open an empty new SQL editor from the **Data Explorer** perspective, do the following:

1. In the **Connections** view of the **Data Explorer** perspective, right-click any connection in the list.

A contextual menu is displayed.



2. Select **New SQL Editor**.

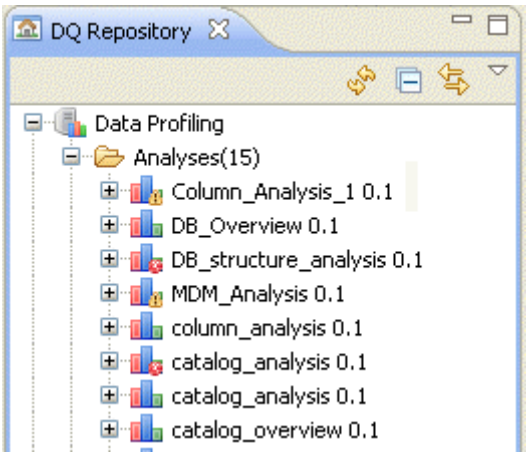
   A new SQL empty editor opens in the **Data Explorer** perspective.

To open an empty SQL editor from the **Profiling** perspective of the studio, see the procedure outlined in section *Creating and storing SQL queries*.

2.4. Icons appended on analyses names in the DQ Repository

When you create any analysis type from the studio, a corresponding analysis item is listed under the **Analyses** folder in the **DQ Repository** tree view.

The number of the analyses created in the studio will be indicated next to this **Analyses** folder in the **DQ Repository** tree view.



This analysis list will give you an idea about any problems in one or more of your analyses before even opening the analysis.

If an analysis fails to run, a small red-cross icon will be appended on it. If an analysis runs correctly but has violated thresholds, a warning icon is appended on such analysis.

2.5. Multi-perspective approach

Your **Talend** studio offers a comprehensive set of tools and functions for all its key capabilities. These tools are all accessible from different perspectives within the studio.

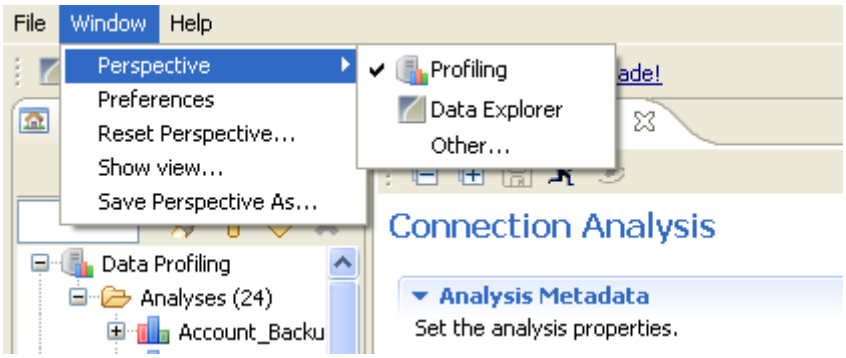
2.5.1. Switching between different perspectives

You can switch between the **Profiling** and **Data Explorer** perspectives by clicking the quick access icon in the top left corner of the studio.

Several other perspectives that extend the studio functionalities are also available within the studio.

To switch to perspectives using the menu bar, do the following:

1. On the menu bar, click **Window > Perspective**.



2. Select from the list:

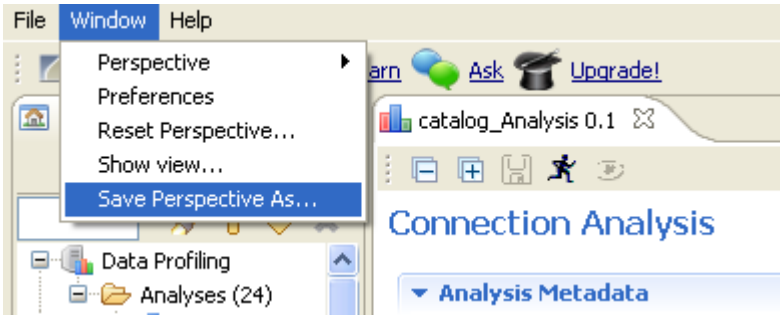
Item	to...
Profiling	open the data profiler perspective where you can examine data available in different data sources.
Data Explorer	open the data explorer perspective where you can browse and query analyzed data.
Other...	open a dialog box from which you can select to open different perspectives that extend the studio functionalities.

2.5.2. Saving the configuration of a perspective

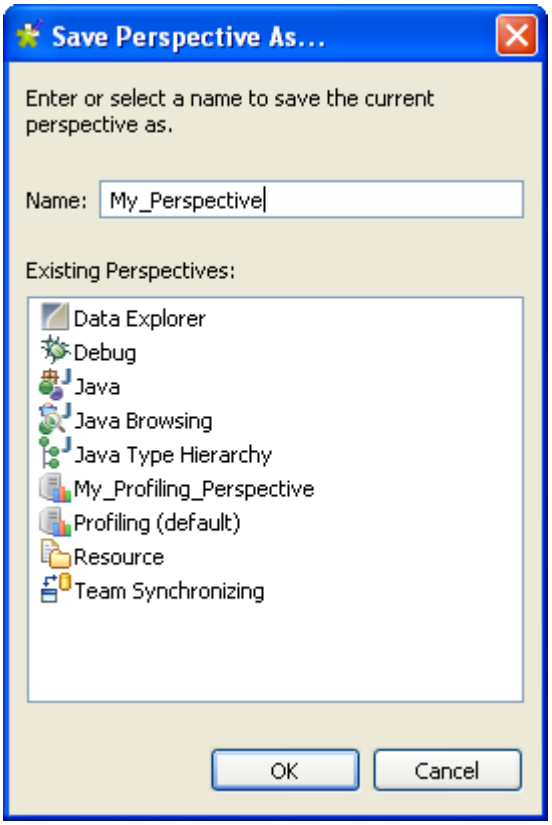
You can save the configuration of your current perspective in order to list it as a new perspective in the perspective dialog box.

To save the configuration of the current perspective, do the following:

1. On the menu bar, click **Window > Save Perspective As....**



2. In the **Name** field, enter a name.



3. Click **OK** .

   The current perspective is saved as a new perspective under the new name.

You can open this perspective any time by selecting it from the **[Open Perspective]** dialog box. For further information, see section *Switching between different perspectives*.

Chapter 3. Before you begin profiling data

The **Profiling** perspective of the studio enables you to profile data in databases, in files, or on Master Data Management (MDM) servers.

This chapter explains how to set up different connections to your data sources in order to be able to profile data in these sources. It describes as well how to manage such connections.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

3.1. Creating connections to different data sources

The **Profiling** perspective of the studio enables you to create connections to several databases, delimited or excel files or MDM servers in order to profile data in such different data sources.

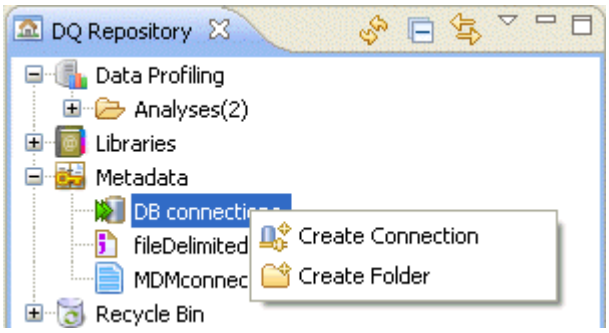
3.1.1. Connecting to a database

Before proceeding to analyze data in a specific database, you must first set up the connection to this database. From the **Profiling** perspective of the studio, you can create a connection on the DataBase Management System (DBMS) and show the content of the database in the **DQ Repository** tree view.

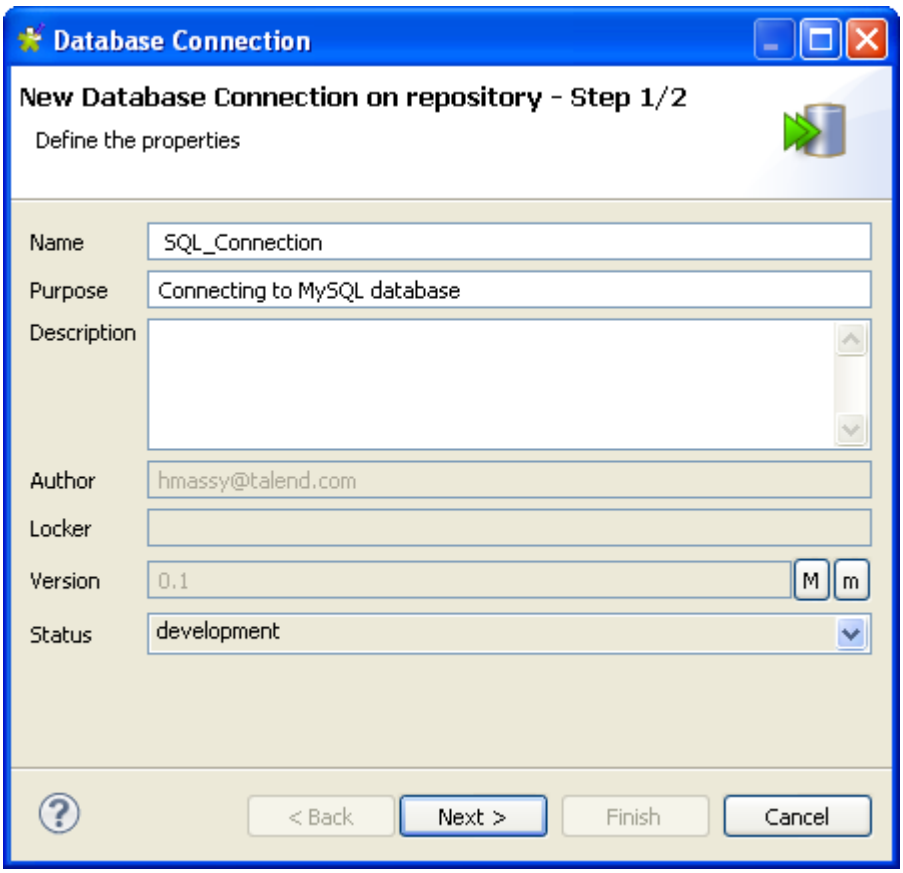
These connections to different databases are reflected by different tree levels and different icons in the **DQ Repository** tree view because the logical and physical structure of data differs from one relational database to another. The highest level structure “Catalog” followed by “Schema” and finally by “Table” is not applicable to all database types.

To create a database connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata**, right-click **DB Connections** and select **Create DB Connection**.

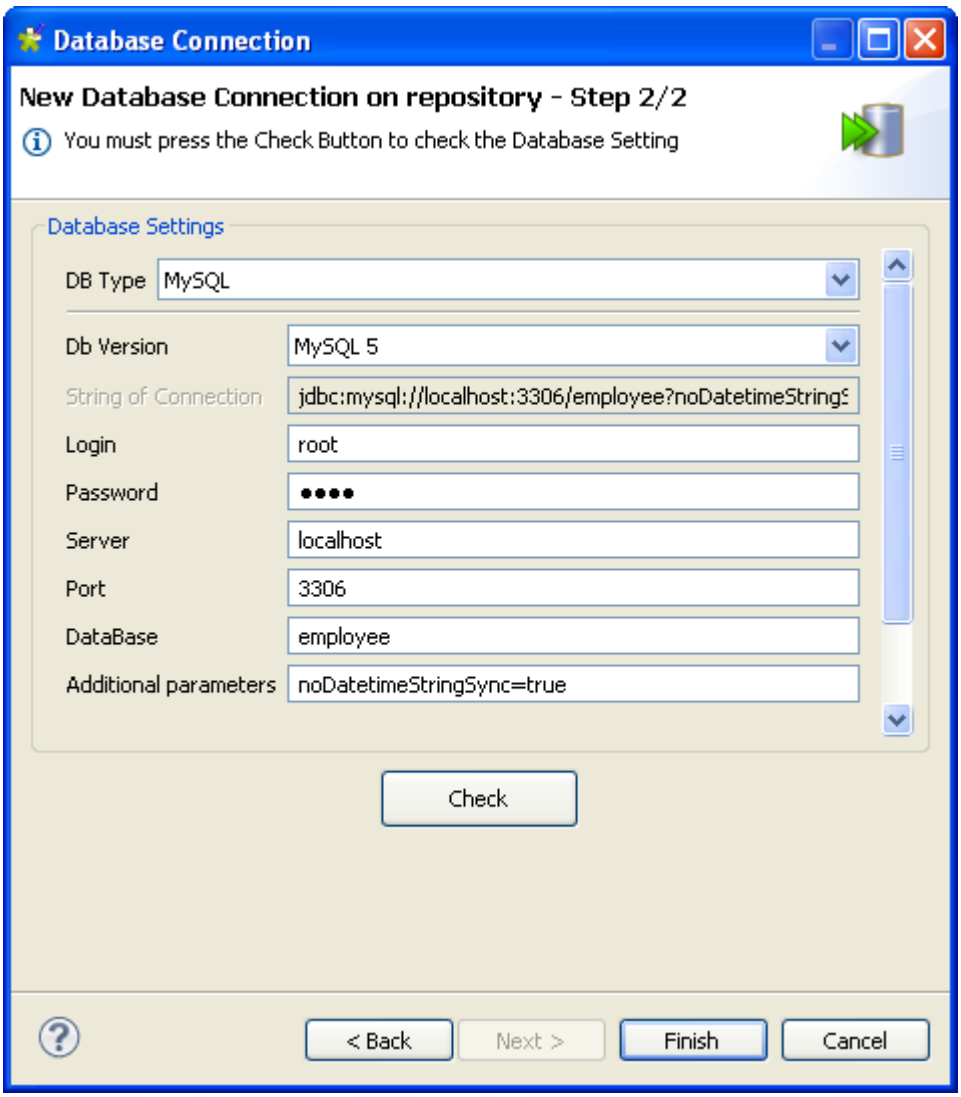


   The **[Database Connection]** wizard opens.



2. In the **Name** field, enter a name for this new database connection.

3. If required, set other connection metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



4. In the **DB Type** field and from the drop-down list, select the type of database to which you want to connect. For example, *MySQL*.

   If you select to connect to a database that is not supported in the studio (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on column analyses, see section *Defining the columns to be analyzed and setting indicators*, and for more information on the Java engine, see section *Using the Java or the SQL engine*.

5. In the **DB Version** field, select the version of the database to which you are creating the connection.

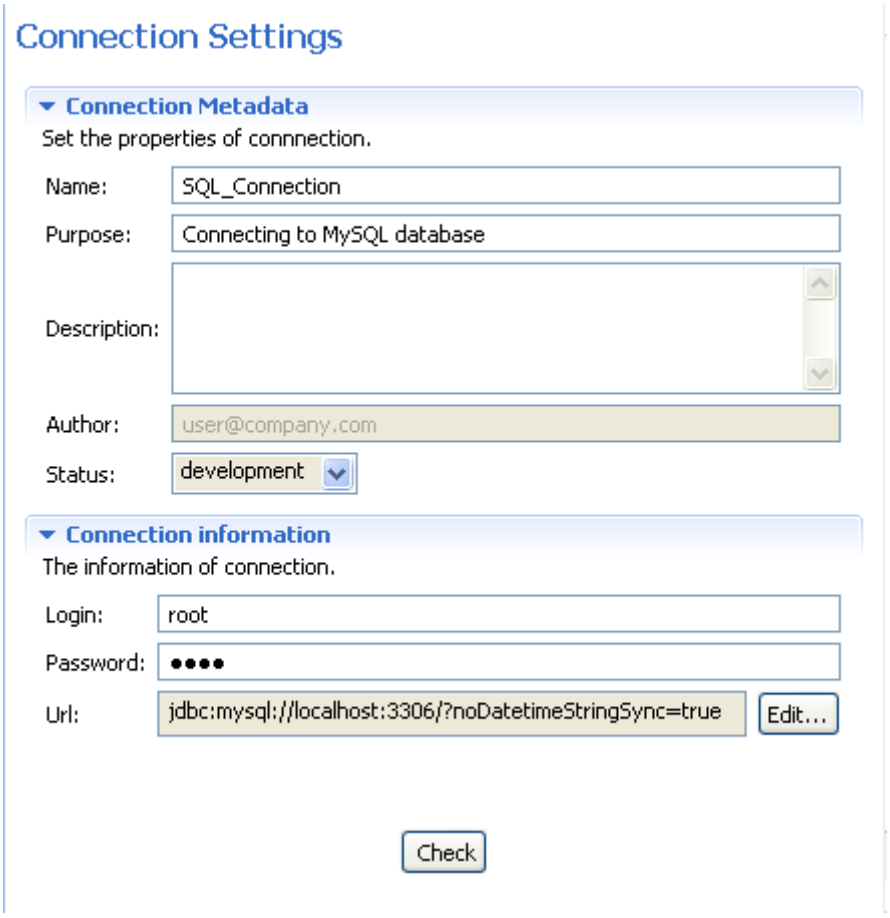
6. Enter your login, password, server and port information in their corresponding fields.

7. In the **Database** field, enter the database name you are connecting to. If you need to connect to all of the catalogs within one connection, if the database allows you to, leave this field empty.

8. Click the **Check** button to verify if your connection is successful.

9. Click **Finish** to close the **[Database Connection]** wizard.

A folder for the created database connection is displayed under **DB Connection** in the **DQ Repository** tree view. The connection editor opens with the defined metadata in the studio.



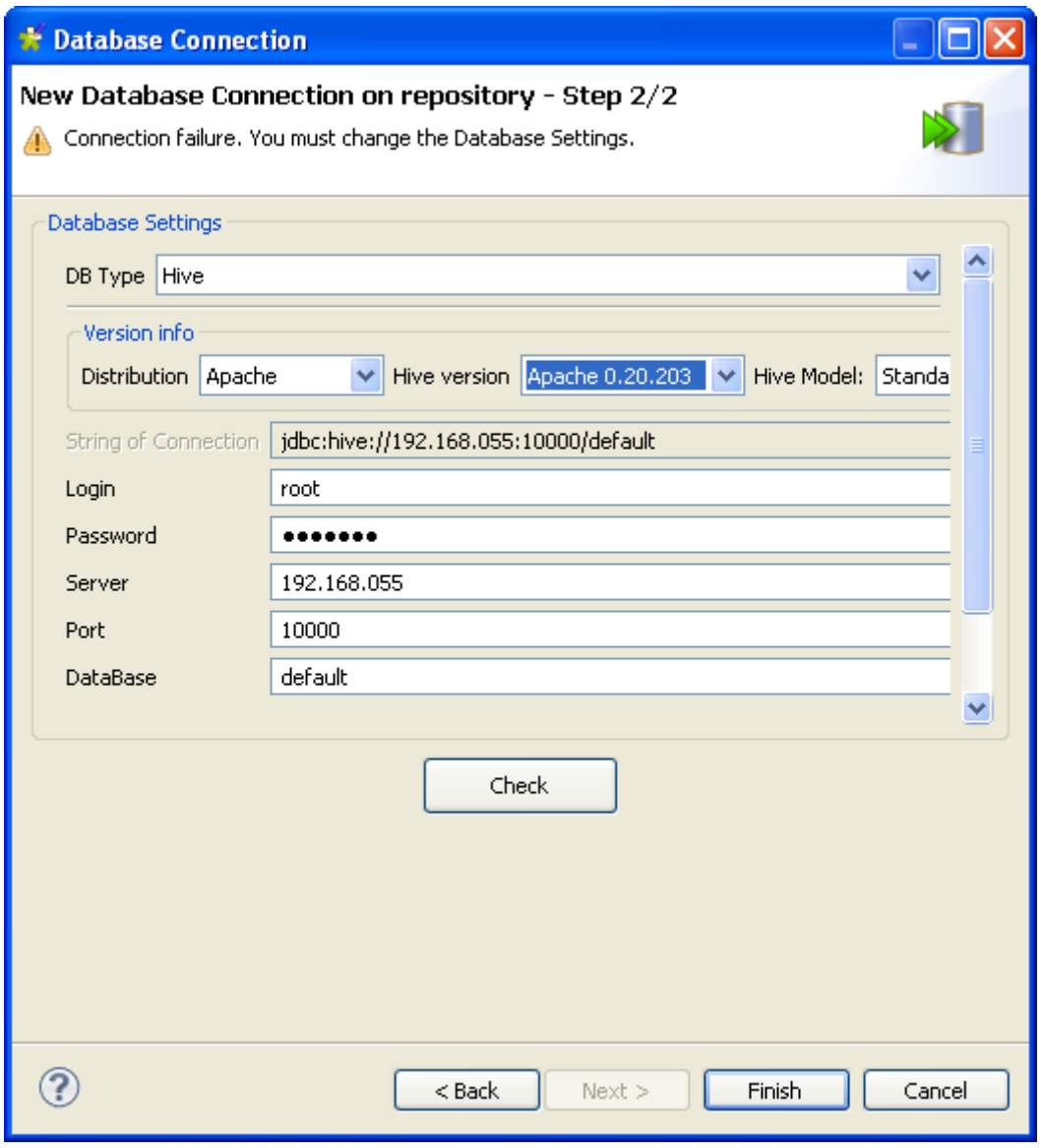
From the connection editor, you can:

- Click **Connection information** to show the connection parameters for the relevant database.
- Click the **Check** button to check the status of your current connection.
- Click the **Edit...** button to open the connection wizard and modify any of the connection information.

For information on how to set up a connection to a file, see section *Connecting to a file*. For information on how to set up a connection to an MDM server, see section *Connecting to an MDM server*.

3.1.1.1. What you need to know about some databases

If you select to connect to the Hive database, you will be able to create and execute different analyses as with the other database types.



In the connection wizard, you must select from the **Distribution** list the platform that hosts Hive. You must set as well Hive version and mode. For further information, check http://hadoop.apache.org/.

Please note that one analysis type and few indicators and functions are still not supported for Hive, see the table below for more detail:

Unsupported indicators	Unsupported functions	Unsupported analyses
with SQL engine: -Soundex Low Frequency Table. -Pattern(Low) Frequency Table. -Upper Quartile and Lower Quartile. -Median. - All Date Frequency indicators.	-the **View rows** contextual menu for column analyses with unique, duplicate and all textual indicators. For further information on the **View rows** menu, see section *Viewing and exporting analyzed data*. -the **View match rows** contextual menu for column analyses with unique, duplicate and all textual indicators. For further information on **View match rows**, see section *Comparing identical columns in different tables*. -all contextual menus on the analysis results of functional dependency analysis.	-the only analysis that is not supported for Hive is **Time Correlation Analysis** as the *Date* data type does not exist in Hive. For further information on this analysis type, see section *Time correlation analysis*.

Unsupported indicators	Unsupported functions	Unsupported analyses
	For further information on this analysis, see section *Detecting anomalies in the table columns: column functional dependency analysis*.	

3.1.2. Connecting to a file

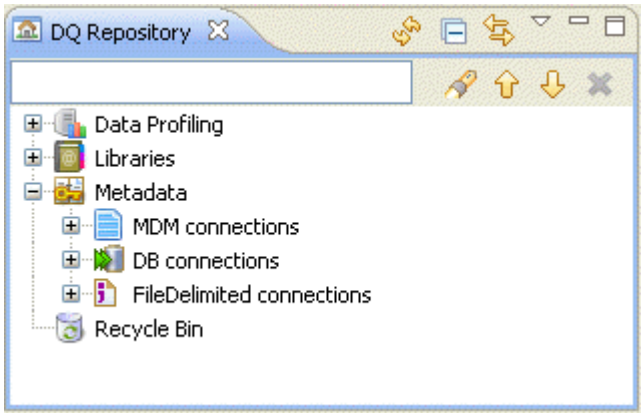
Before proceeding to analyze data in a delimited file or an excel file, you must first set up the connection to such a file.

3.1.2.1. How to connect to a delimited file

Before being able to profile data in a delimited file, you must first set up the connection to this file.

To create a connection to a delimited file, do the following:

1. Expand the **Metadata** folder.



2. Right-click **FileDelimited connections** and then select **Create File Delimited Connection** to open the **[New Delimited File]** wizard.

3. Follow the steps defined in the wizard to create a connection to a delimited file. For further information, see the *Talend Open Studio for Data Integration User Guide*.

   You can then create a column analysis and drop the columns to analyze from the delimited file metadata in the **DQ Repository** tree view to the open analysis editor. For further information, see section *Analyzing columns in a delimited file*.

   For information on how to set up a connection to a database, see section *Connecting to a database*. For further information on how to set up a connection to an MDM server, see section *Connecting to an MDM server*.

3.1.2.2. How to connect to an Excel file

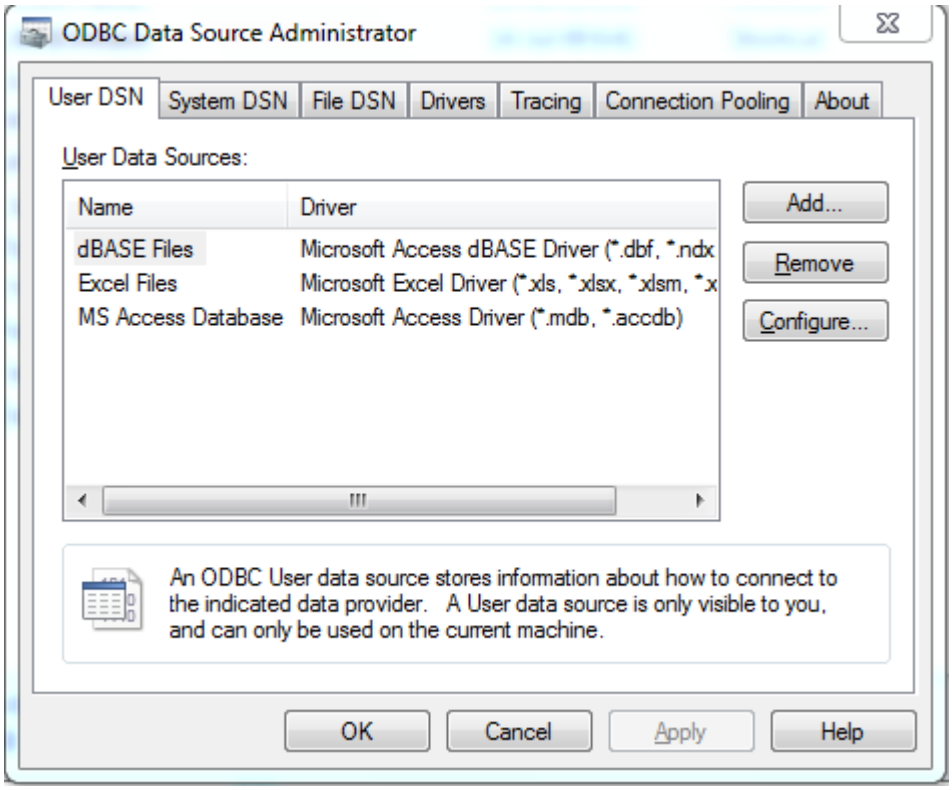
Before being able to profile data in an excel file, you must create your Data Source, and then set up the connection to this Data Source.

To create the Data Source, do the following:

1. On the task bar of your desktop, click the **Start** button and then select **Control Panel** to open the corresponding page.

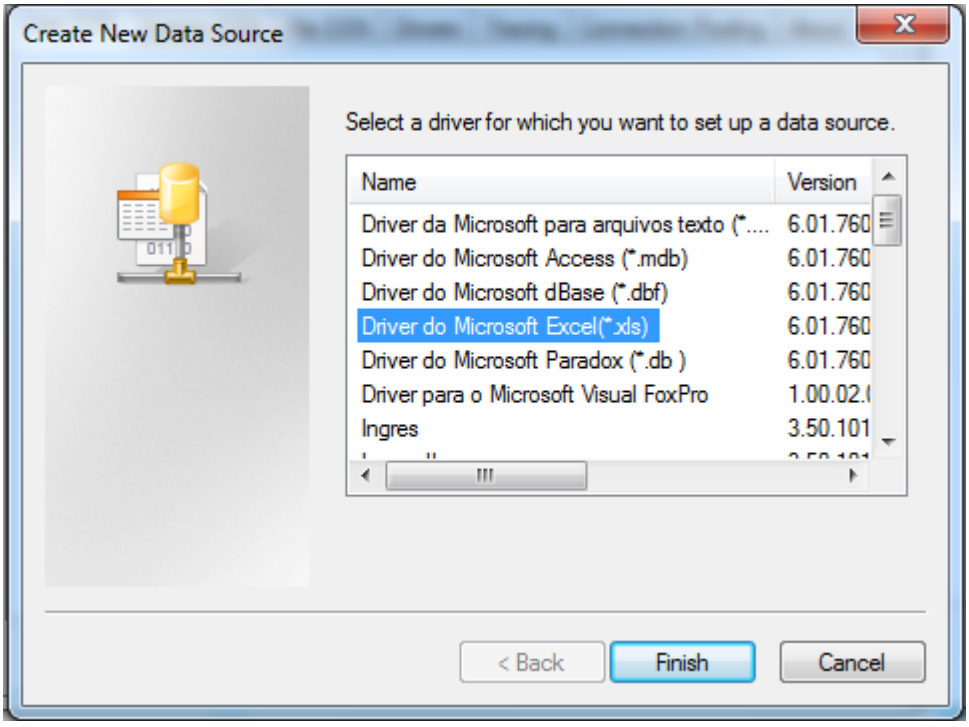
2. Double-click **Tools and Administrator** to open the corresponding page.

3. Double-click **Data sources (ODBC)**.

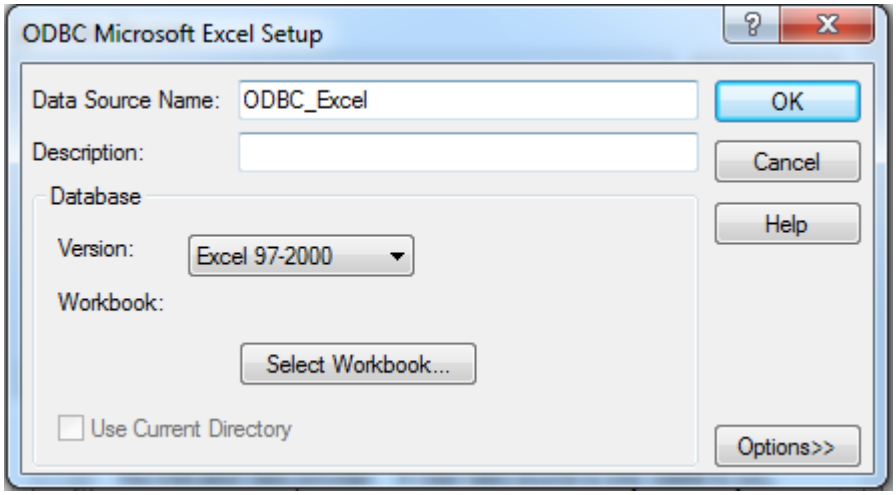


A dialog box opens.

4. In the **User DSN** view, click **Add...** to open a dialog box where you can select the ODBC driver, Microsoft Excel in this example, for the data source (database) to which you want to connect.



5. Click **Finish** to proceed to the step where you can define the Data Source.



6. In the **Data Source Name** field, enter a name for the Data Source, and then click the **Select Workbook...** tab to proceed to the step where you link this Data Source to the excel file you want to profile.

7. In the open dialog box, browse to the excel file to which you want to link your Data Source.

To be able to set an ODBC connection to the Data Source without problems, make sure that the excel files you want to profile are put in a folder, i.e. they are not located on the root directory of your system.

8. Select the excel file and then click **OK** to close the open dialog boxes. The Data Source you create is listed in the **User Data Sources** list.

9. Click **OK** to close the dialog box.

You can then create a column analysis and drop the columns to analyze from the excel file metadata in the **DQ Repository** tree view to the open analysis editor. For further information, see section *Analyzing columns in an excel file*.

For information on how to set up a connection to a database, see section *Connecting to a database*. For further information on how to set up a connection to an MDM server, see section *Connecting to an MDM server*.

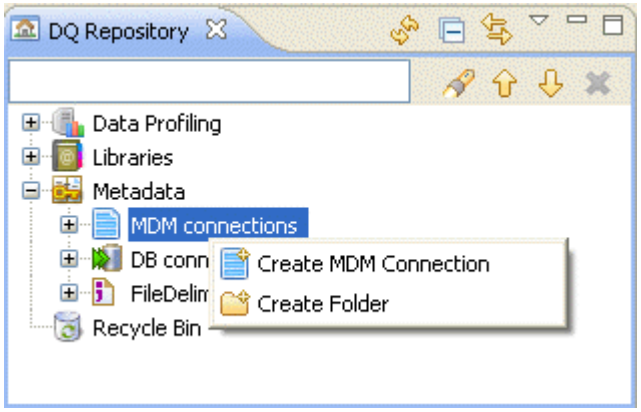
3.1.3. Connecting to an MDM server

Before proceeding to analyze master data on an MDM server, you must first set up the connection to such a server. *Talend Open Studio for Data Quality* enables you to create a connection to the MDM server. Once connected, the content of the server is displayed in the **DQ Repository** tree view.

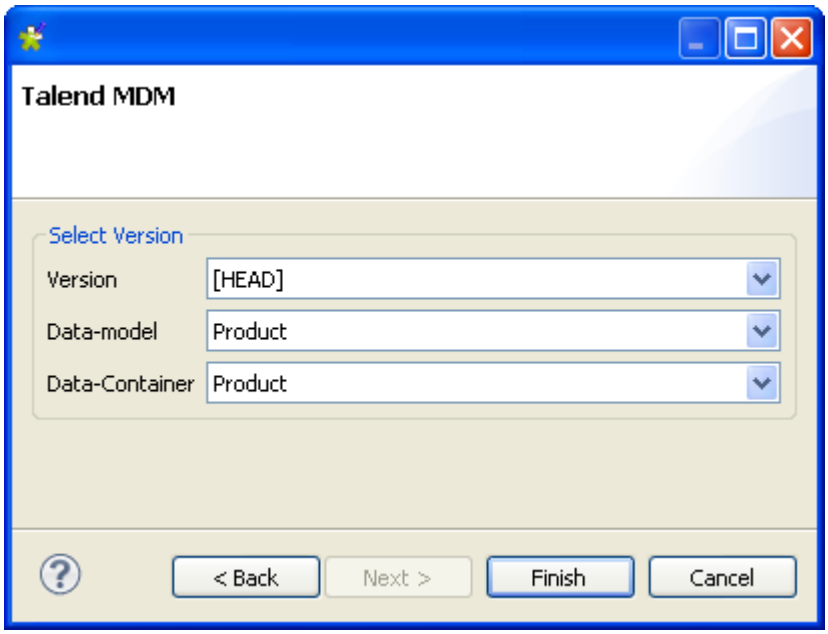
Prerequisite(s): The MDM server to which you want to connect is up and running.

To create an MDM connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata**, right-click **MDM Connections** and then select **Create MDM Connection**.



The **[MDM Connection]** wizard opens.

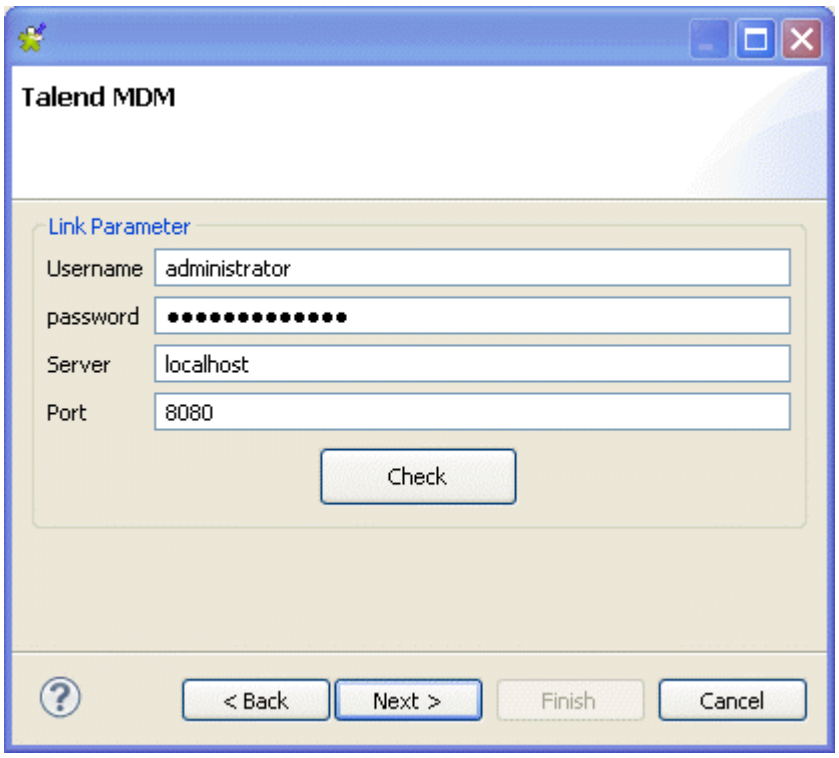


2. In the **Name** field, enter a name for this new MDM connection.

   Spaces between words are not allowed when typing in the connection name in this field.

3. If required, set a purpose and a description for the connection in the corresponding fields. The **Status** field is a customized field that can be defined. For more information, see the *Talend Open Studio for Data Integration User Guide*.

4. Click **Next** to proceed to the next step.



To set the connection parameters, do the following:

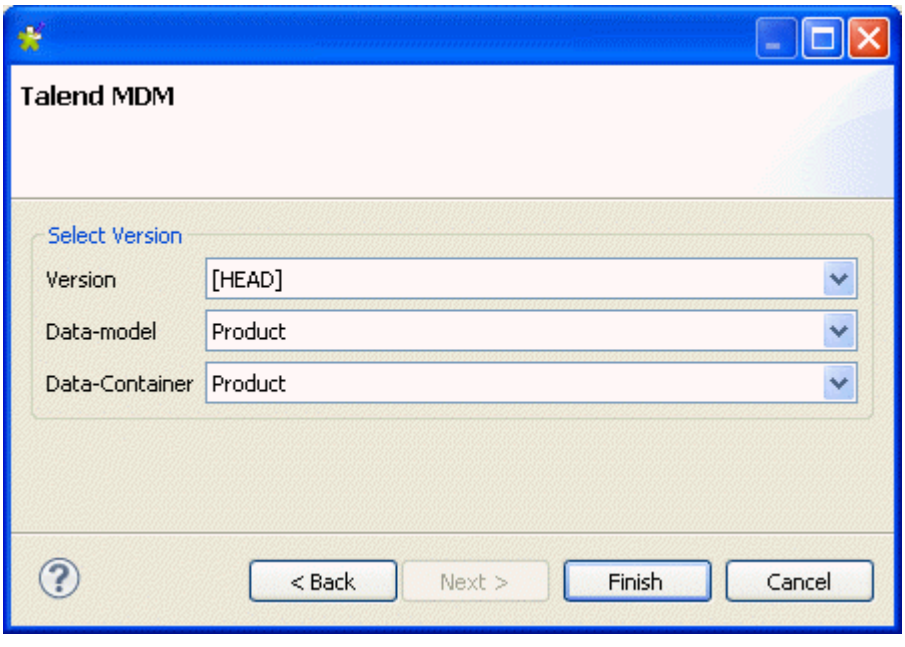
1. Enter your login and password to the MDM server in their corresponding fields.

   Make sure that the role that has been assigned to you in the MDM Studio gives you enough rights to access the MDM server via your studio. For further information, see the *Talend Open Studio for MDM Administrator Guide*.

2. Set the connection parameters to the MDM server in the **Server** and **Port** fields.

3. Click the **Check** button to verify if your connection is successful. A confirmation message is displayed.

4. Click **OK** to close the message and then **Next** to proceed to the next step.



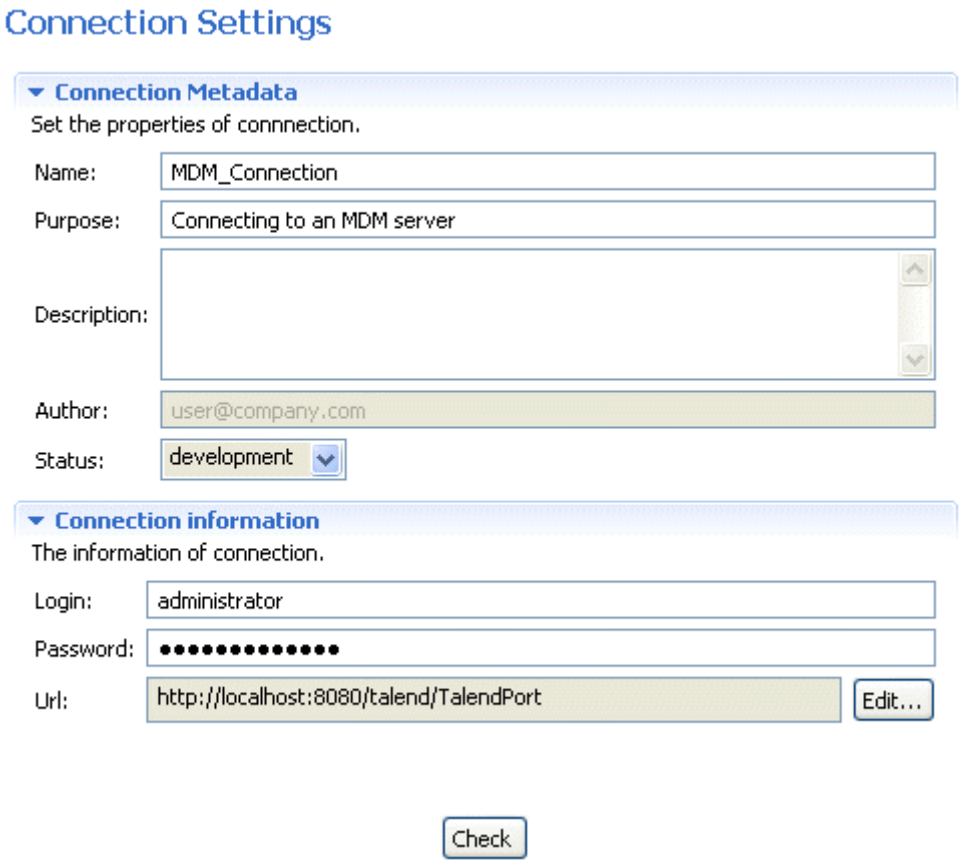
5. From the **Version** list, select the master data Version on the MDM server to which you want to connect.

6. From the **Data-Model** list, select the data model against which master data is validated.

7. From the **Data-Container** list, select the data container that holds the master data you want to access.

8. Click **Finish** to validate your changes and close the wizard.

A folder for the created MDM connection is displayed under the **MDM Connections** folder under the **Metadata** node in the **DQ Repository** tree view, and the analysis editor opens with the defined metadata.



The display of the connection editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

From the analysis editor, you can:

- Click **Connection information** to show the connection parameters for the relevant MDM server.
- Click the **Check** button to check the status of your current connection.
- Click the **Edit...** button to open the connection wizard where you can edit the connection parameters.

For information on how to set up a connection to a database, see section *Connecting to a database*. For further information on how to set up a connection to a file, see section *Connecting to a file*.

3.2. Managing connections to data sources

Several management options are available for each of the connections created in the studio.

3.2.1. Managing database connections

Many management options are available for database connections including editing and duplicating the connection or adding a task to it.

The sections below explain in detail these management options.

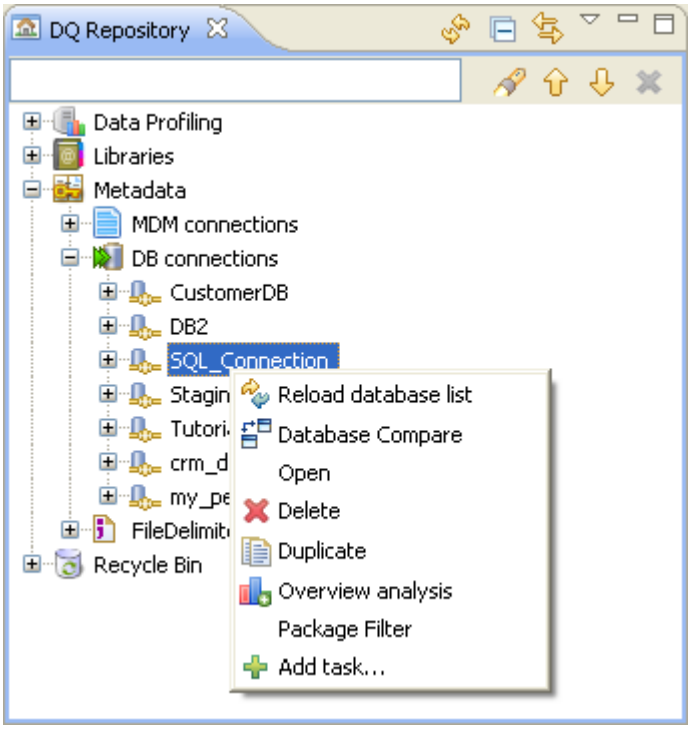
3.2.1.1. How to open or edit a database connection

You can edit the connection to a specific database and change the connection metadata and the connection information.

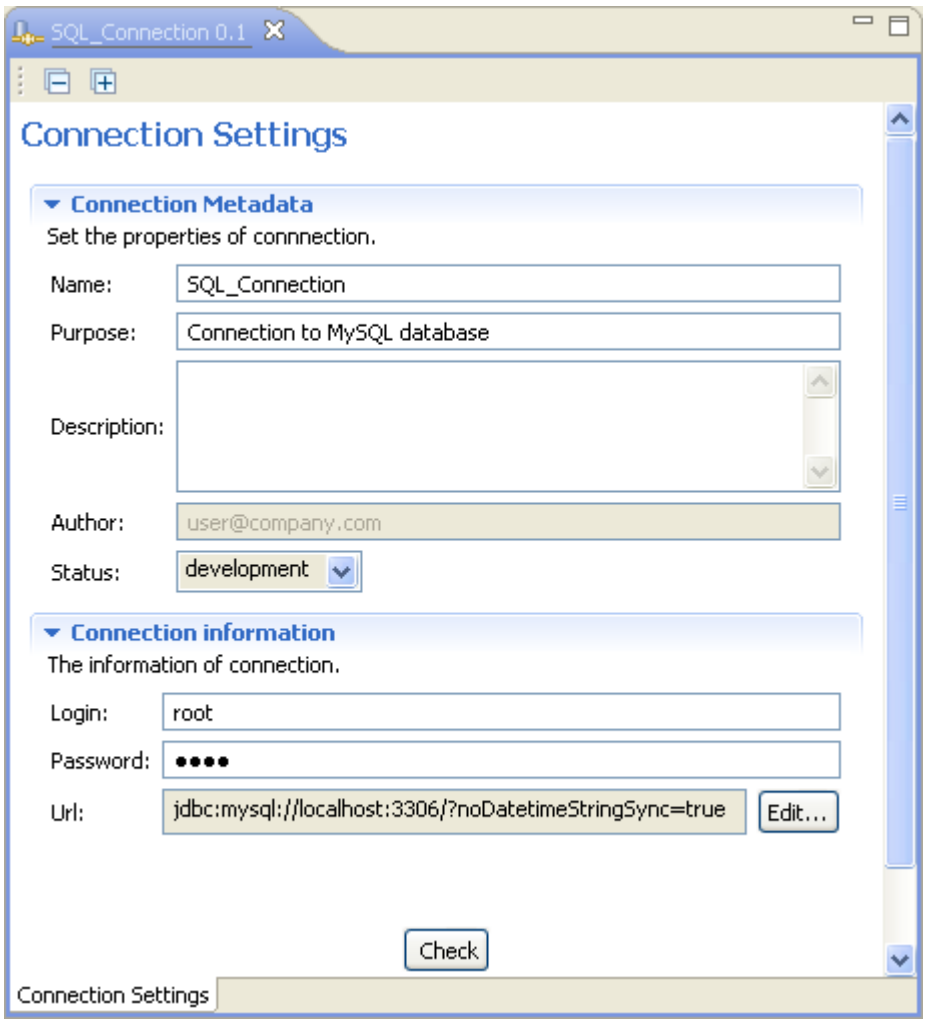
Prerequisite(s): A database connection is created in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

To edit an existing database connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connection**.
2. Either:
   - Double-click the database connection you want to open, or,
   - Right-click the database connection and select **Open** in the contextual menu.

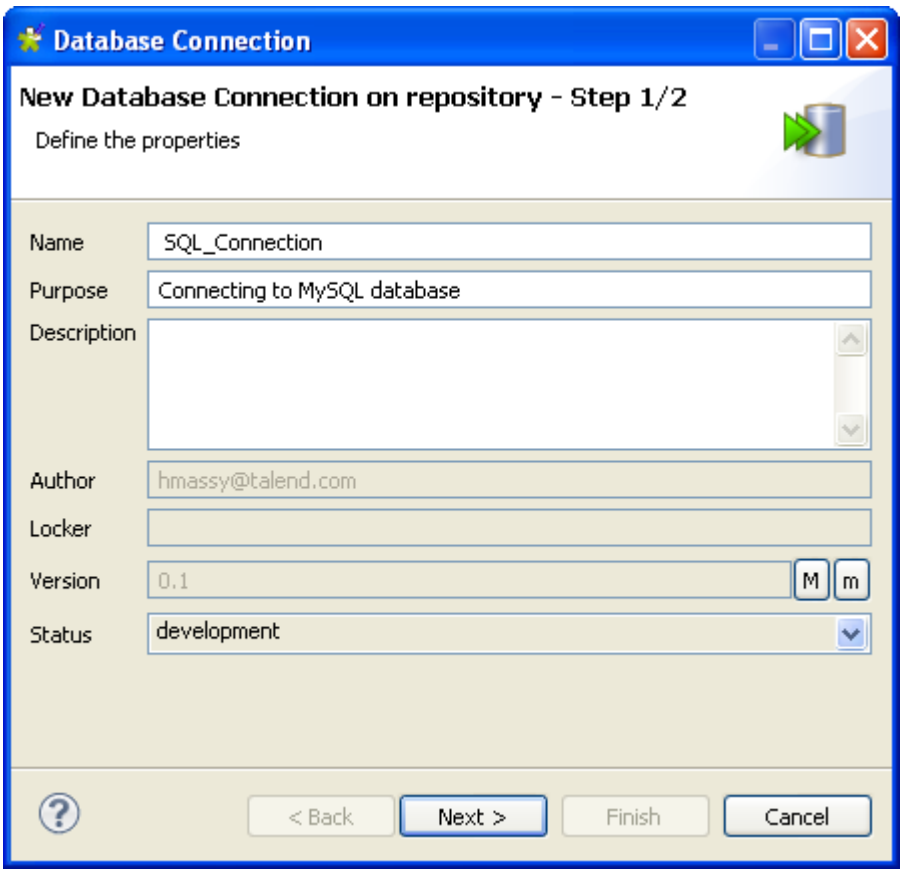


The connection editor for the selected database connection is displayed.



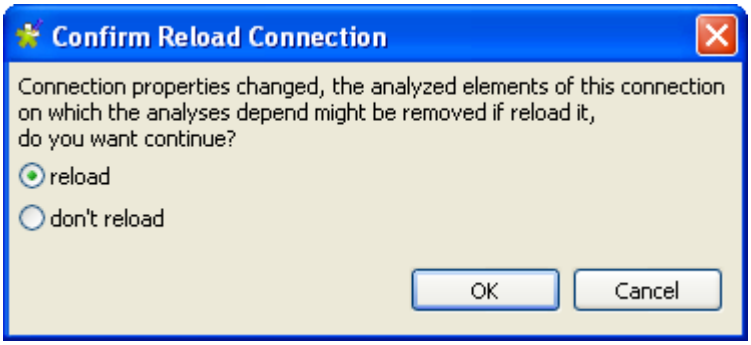
3. Modify the connection metadata in the **Connection Metadata** view, as required.

4. Click the **Edit...** button in the **Connection information** view to open the **[Database Connection]** wizard.



5. Go through the steps in the wizard and modify the database connection settings as required.
6. Click **Finish** to validate the modifications.

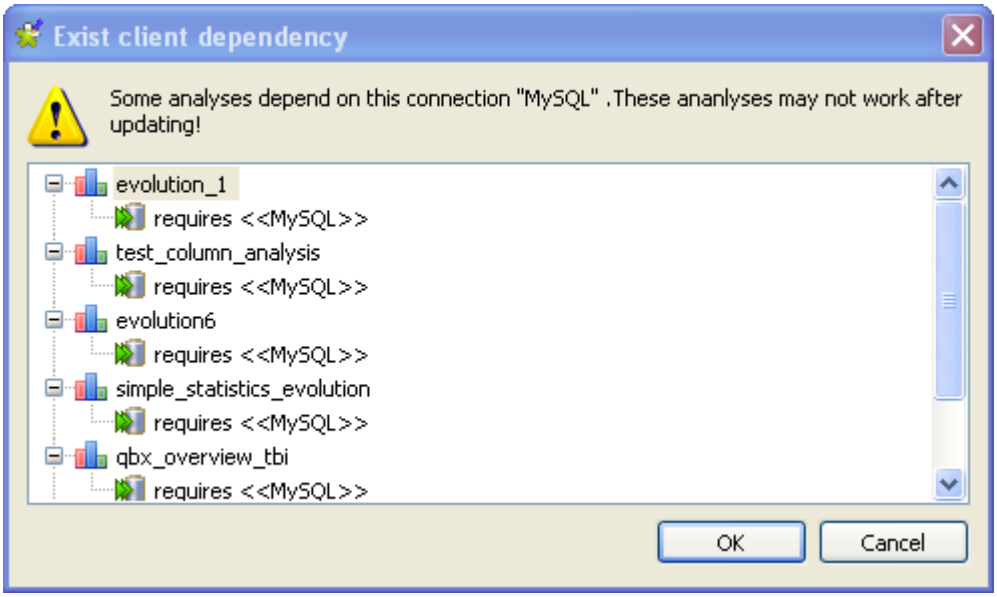
   A dialog box opens prompting you to reload the updated database connection.



7. Select the **reload** option if you want to reload the new database structure for the updated database connection.

   If you select the **don't reload** option, you will still be able to execute the analyses using the connection even after you update it.

   If the database connection is used by profiling analyses in the Studio, another dialog box is displayed to list all the analyses that use the database connection. It alerts you that if you reload the database new structure, all the analyses using the connection will become unusable although they will be always listed in the **DQ Repository** tree view.



8. Click **OK** to accept reloading the database structure or **Cancel** to cancel the operation and close the dialog box.

   A number of confirmation messages are displayed one after the other.

9. Click **OK** to close the messages and reload the structure of the new connection.

3.2.1.2. How to filter a database connection

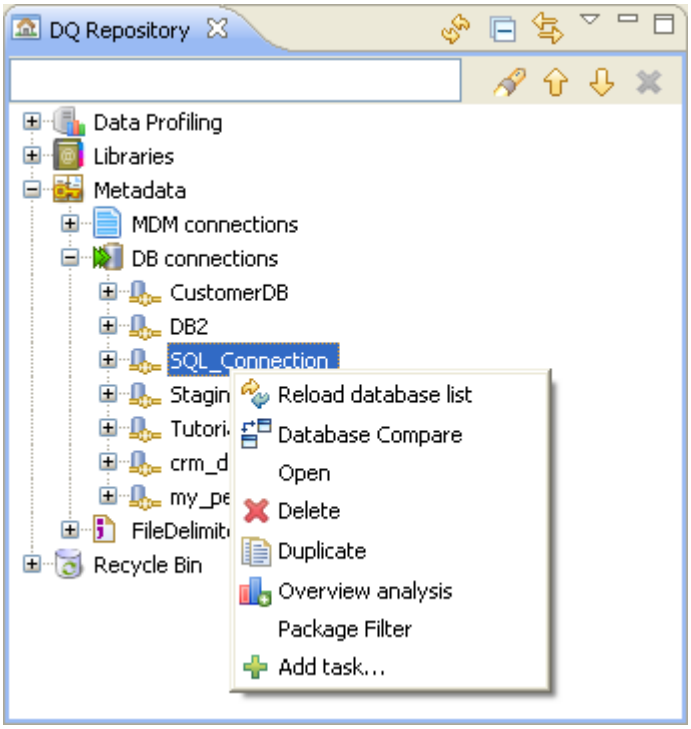
After setting a specific database connection in the studio, you may not want to view all databases in the **DQ Repository** tree view of your Studio.

You can filter your database connections to list the databases that match the filter you set. This option is very helpful when the number of databases in a specific connection is very big.

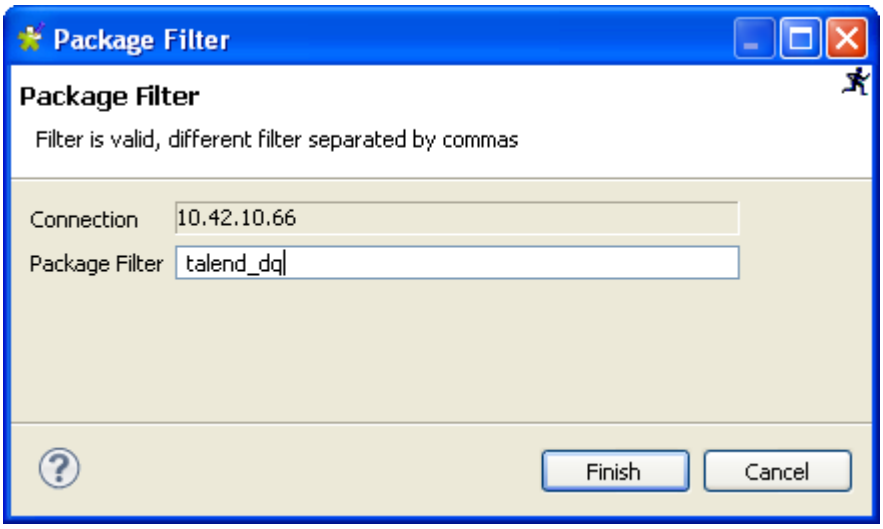
Prerequisite(s): A database connection is already created in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

To filter a database connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connection**.

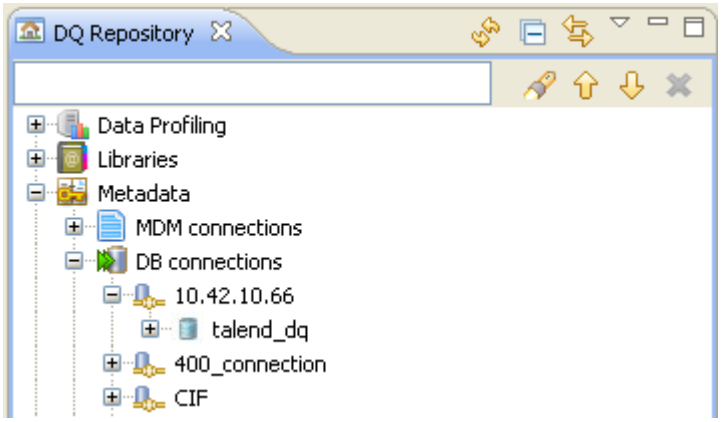


2. Right-click the database connection you want to filter and select **Package Filter** to open the corresponding dialog box.



3. In the **Package Filter** field, enter the complete name of the database you want to view and then click **Finish** to close the dialog box.

   Only the database that matches the filter you set is listed under the database connection in the **DQ Repository** tree view.



To cancel the filter, do the following:

1. In the **[Package Filter]** dialog box, delete the text from the **Package Filter** field.
2. Click **Finish** to close the dialog box.

   All databases are listed under the selected database connection in the **DQ Repository** tree view.

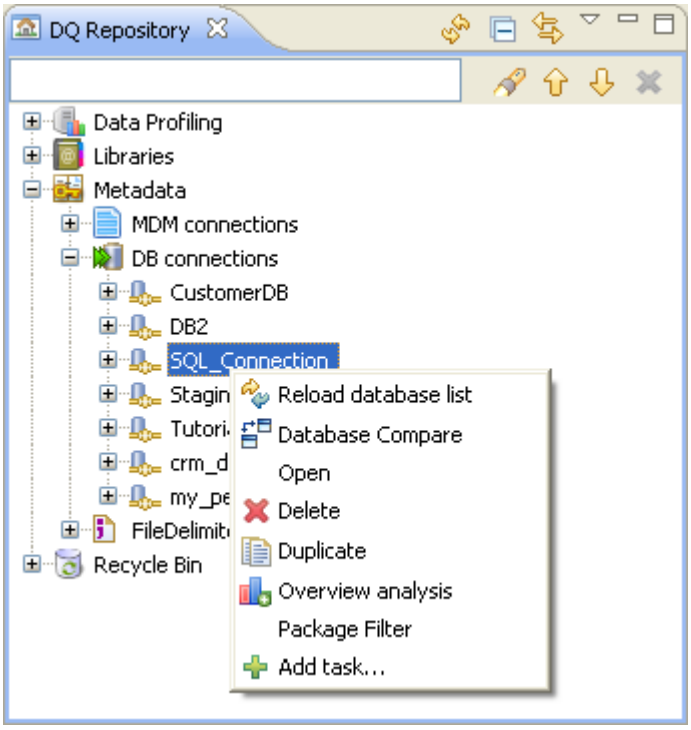
3.2.1.3. How to duplicate a database connection

To avoid creating a DB connection from scratch, you can duplicate an existing one in the **DB Connections** list and work around its metadata to have a new connection.

Prerequisite(s): A database connection is created in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

To duplicate a connection to a specific database, do the following:

1. In the **DQ Repository** tree view, expand **Metadata DB Connections**.
2. Right-click the connection you want to duplicate and select **Duplicate** from the contextual menu.



The duplicated database connection shows under the connection list in the **DQ Repository** tree view as a copy of the original connection. You can now open the duplicated connection and modify its metadata as needed.

3.2.1.4. How to add a task to a database connection or any of its elements

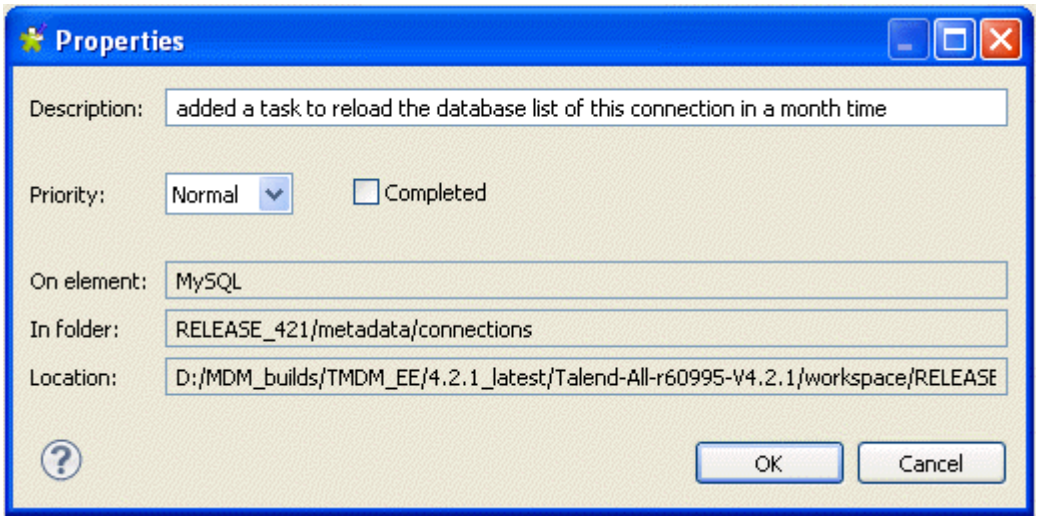
You can add a task to a database connection to use it as a reminder to modify the connection or to flag a problem that needs to be solved later, for example. You can also add a task to a catalog, a table or a column in the connection.

Prerequisite(s): A database connection is created in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

To add a task to a database connection, do the following:

1. Expand **Metadata** and **DB connections**.
2. Right-click the connection to which you want to add a task, and then select **Add task...** from the contextual menu.

   The **[Properties]** dialog box opens showing the metadata of the selected connection.



3. In the **Description** field, enter a short description for the task you want to attach to the selected connection.
4. On the **Priority** list, select the priority level and then click **OK** to close the dialog box.

   The created task is added to the **Tasks** list.

You can follow the same steps in the above procedure to add a task to a catalog, a table or a column in the connection. For further information, see section *Adding a task to a column in a database connection*.

For more information on how to access the task list, see section *Displaying the task list*.

3.2.1.5. How to filter tables/views in a database connection

You can filter the tables/views to list under any database connection.

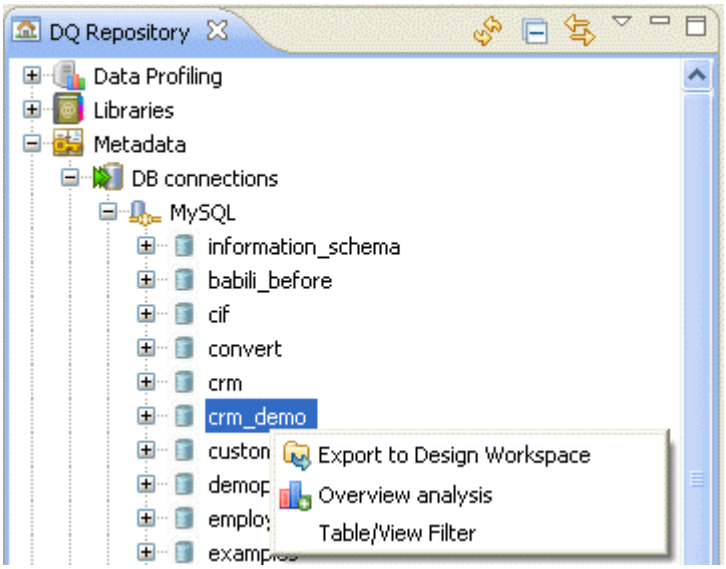
This option is very helpful when the number of tables in the database to which the studio is connecting is very big. If so, a message is displayed prompting you to set a table filter on the database connection in order to list only defined tables in the **DQ Repository** tree view.

Prerequisite(s): A database connection is already created in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

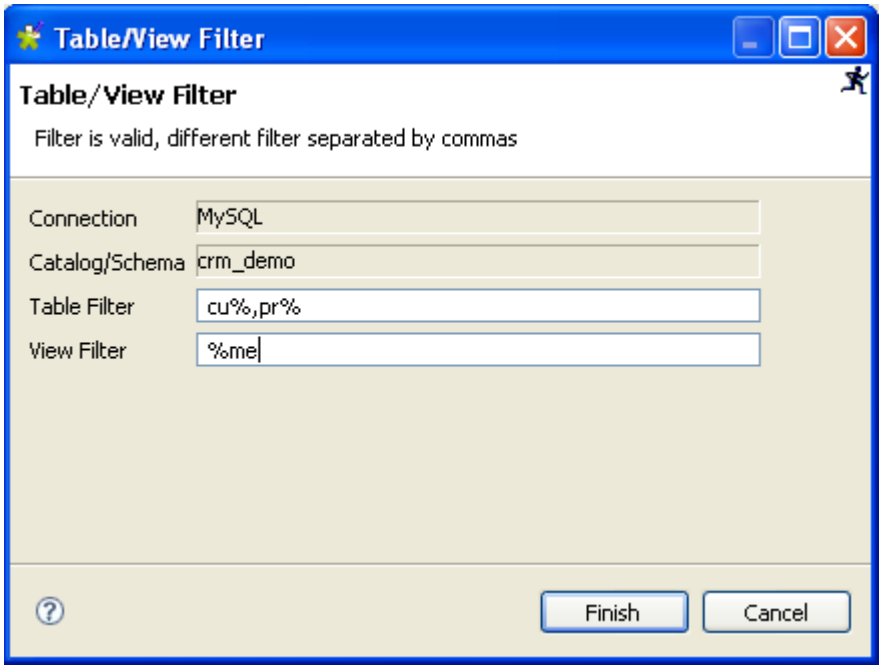
To filter table/views in a database connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.

2. Expand the database connection in which you want to filter tables/views and right-click the desired catalog/schema.



3. Select **Table/View Filter** from the list to display the corresponding dialog box.



4. Set a table and a view filter in the corresponding fields and click **Finish** to close the dialog box.

Only tables/views that match the filter you set are listed in the **DQ Repository** tree view.

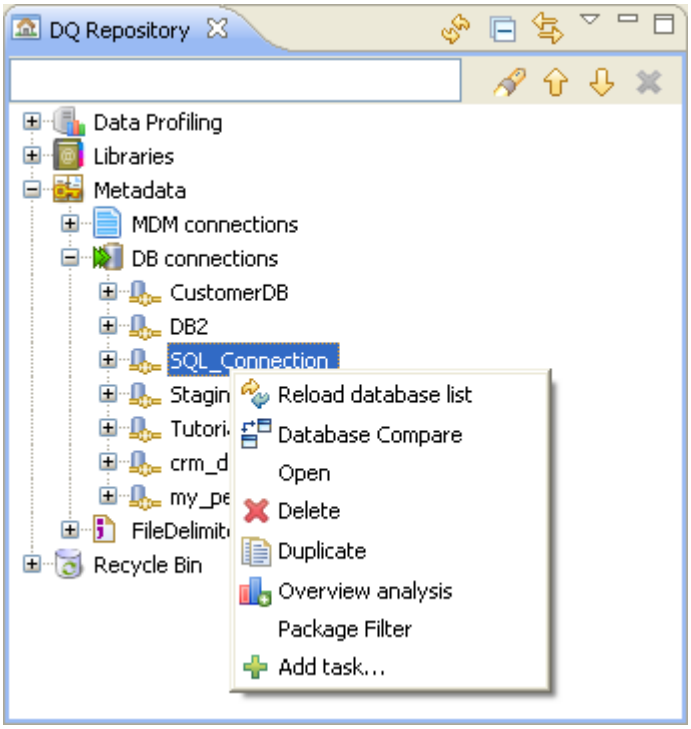
3.2.1.6. How to delete or restore a database connection

You can delete a database connection whether it is used by analyses or not. You can also restore a deleted database connection.

Prerequisite(s): A database connection is created in the studio. For further information, see section *Connecting to a database*.

To delete a database connection from the **Metadata** node, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Right-click the database connection you want to delete and select **Delete** in the contextual menu.



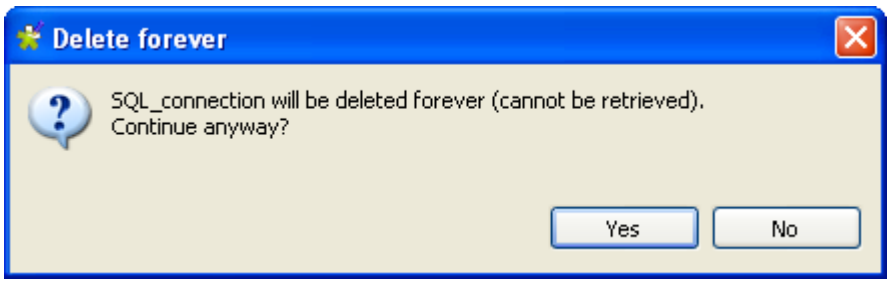
The database connection is moved to the **Recycle Bin**.

You will always be able to run any analysis that uses the connection moved to the recycle bin. However, an alert message will be displayed next to the connection name in the analysis editor.

To delete it from the **Recycle Bin**, do the following:

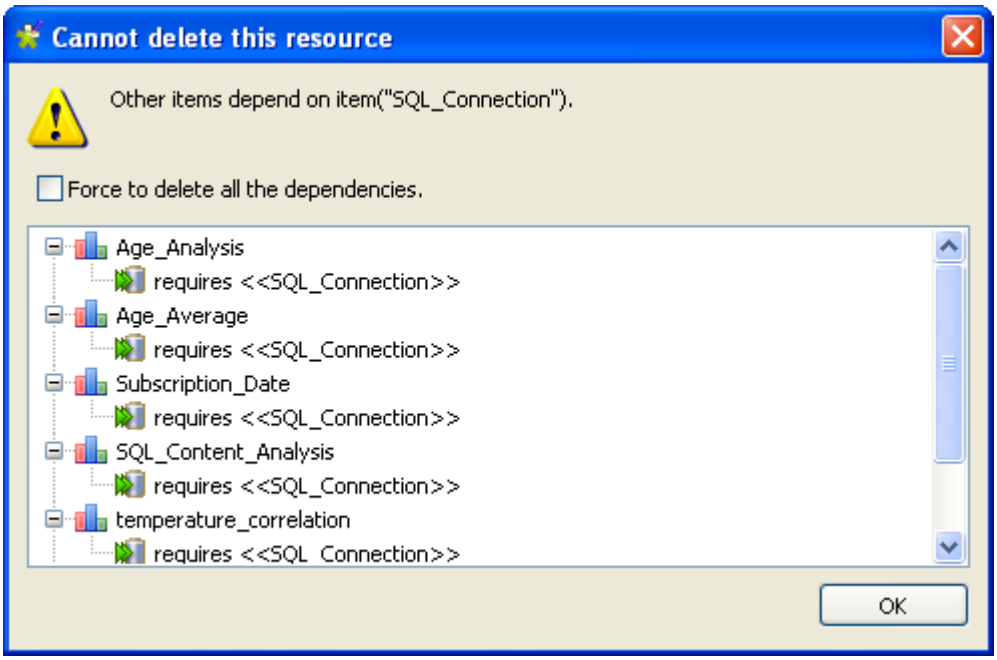
1. Right-click the database connection in the **Recycle Bin** and choose **Delete** from the contextual menu.

   If the connection is not used by any analysis in the current Studio, a **[Delete forever]** dialog box is displayed.



2. Click **Yes** to confirm the operation and close the dialog box.

   If the connection is used by one or more analyses in the current Studio, a dialog box is displayed to list all the analyses that use this database connection.



3. Either:

   - Click **Ok** to close the dialog box without deleting the database connection from the recycle bin.
   - Select the **Force to delete all the dependencies** check box and then click **OK** to delete the database connection from the **Recycle Bin** and to delete all the dependent analyses from the **Data Profiling** node.

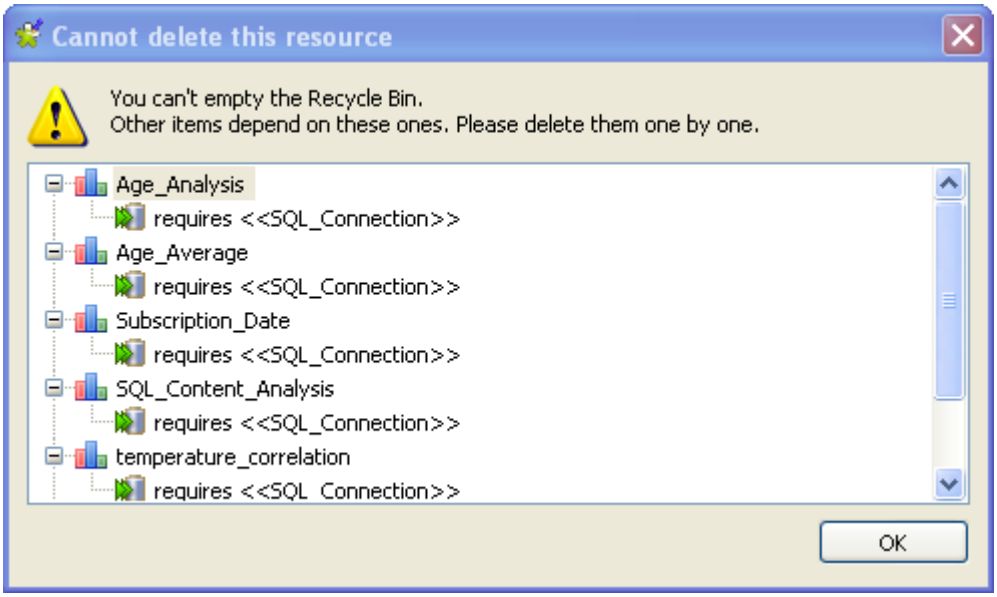
You can also delete permanently the database connection by emptying the recycle bin. To empty the **Recycle Bin**, do the following:

1. Right-click the **Recycle Bin** and select **Empty recycle bin**.

   If the connection is not used by any analysis in the current Studio, a confirmation dialog box is displayed.

2. Click **Yes** to empty the recycle bin.

   If the connection is used by one or more analyses in the current Studio, a dialog box is displayed to list all the analyses that use this database connection.



3. Click **OK** to close the dialog box without removing the connection from the recycle bin.

To restore a database connection from the **Recycle Bin**, do the following:

- In the **Recycle Bin**, right-click the connection and select **Restore**.

  The database connection is moved back to the **Metadata** node.

3.2.2. Managing MDM connections

Many management options are available for MDM connections including editing and duplicating the connection or adding a task to it.

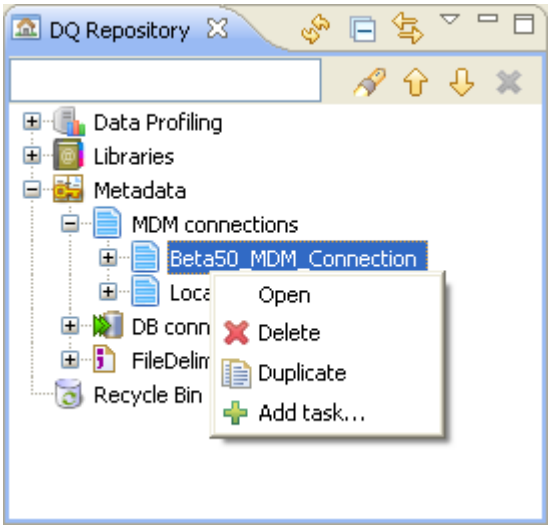
The sections below explain in detail these management options.

3.2.2.1. How to open/edit an MDM connection

Prerequisite(s): An MDM connection is already created in the **Profiling** perspective of the studio. For further information, see section *Connecting to an MDM server*.

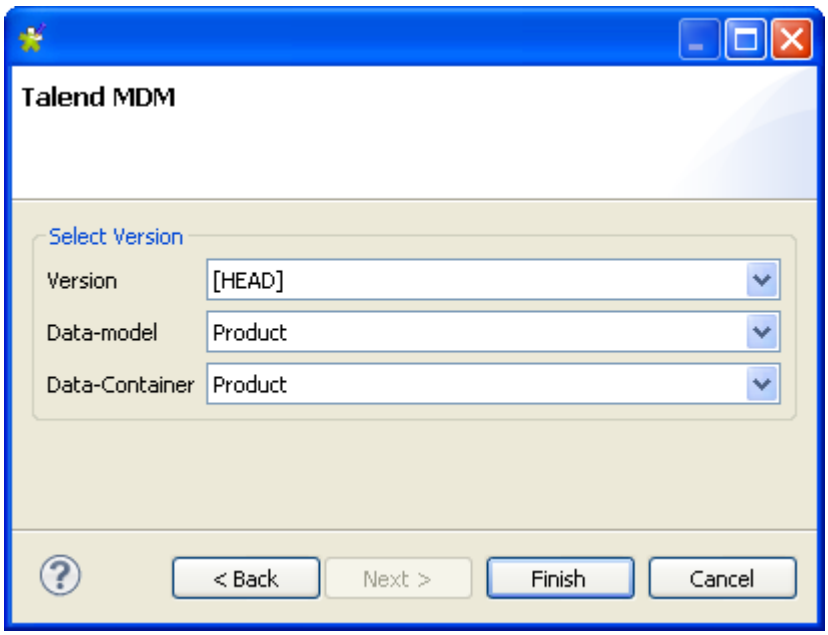
To open an existing MDM connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > MDM Connections**.
2. Either:
   - Double-click the MDM connection you want to open, or
   - Right-click the MDM connection and select **Open** from the contextual menu.



The analysis editor for the selected MDM connection is displayed.

3. Modify the connection metadata as required.
4. Click the **Edit...** button in the **Connection information** view to open the connection wizard again.



5. Go through the steps in the wizard and modify the MDM connection information as required, and then click **Finish** to validate the modifications and close the wizard.

   A number of confirmation messages are displayed one after the other

6. Click **OK** to close the messages and save the modifications.

   *If this MDM connection is used by profiling analyses in the Studio, all these analyses will become unusable although they will be always listed in the **DQ Repository** tree view.*

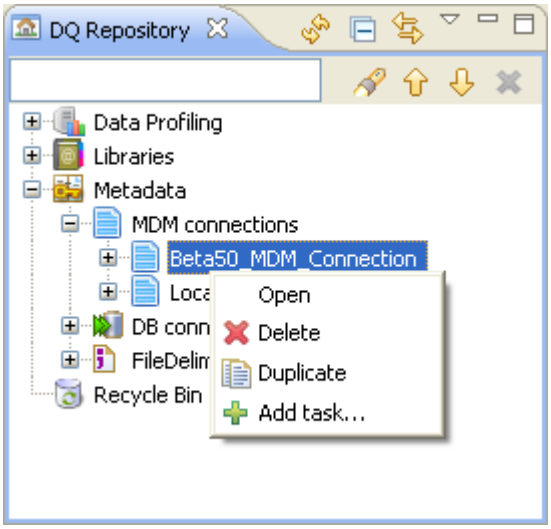
3.2.2.2. How to duplicate an MDM connection

To avoid creating an MDM connection from scratch, you can duplicate an existing one in the **MDM Connections** list and work around its metadata to have a new connection.

Prerequisite(s): At least one MDM connection is created in the **Profiling** perspective of the studio. For further information, see section *Connecting to an MDM server*.

To duplicate a connection to the MDM server, do the following:

1. In the **DQ Repository** tree view, expand **Metadata** > **MDM Connections**.
2. Right-click the connection you want to duplicate and select **Duplicate...** from the contextual menu.



The duplicated MDM connection shows under the connection list in the **DQ Repository** tree view as a copy of the original connection. You can now open the duplicated connection and modify its metadata as needed.

3.2.2.3. How to add a task to an MDM connection or any of its elements

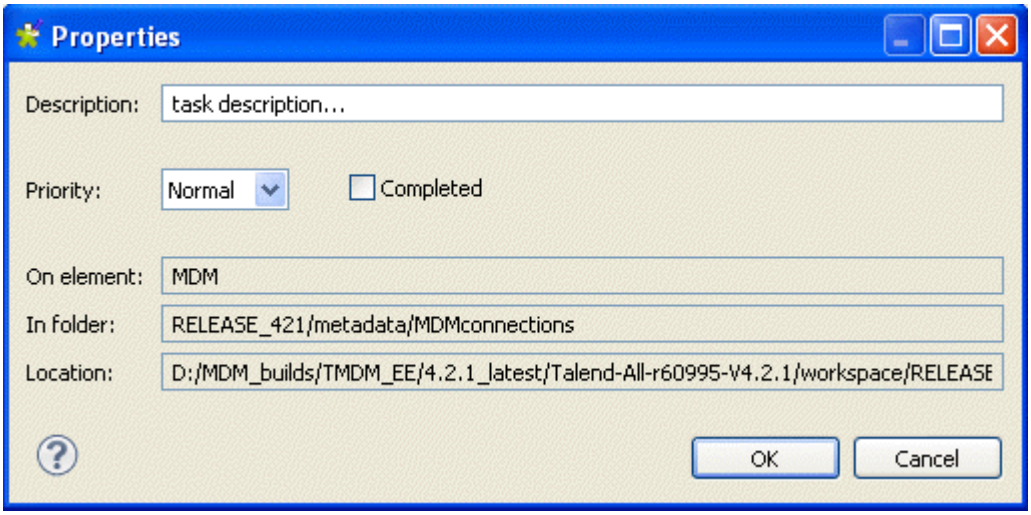
You can add a task to an MDM connection to use it as a reminder to modify the connection or to flag a problem that needs to be solved later, for example. You can also add a task to any entity or column in the connection.

Prerequisite(s): An MDM connection is created in the **Profiling** perspective of the studio. For further information, see section *Connecting to an MDM server*.

To add a task to an MDM connection, do the following:

1. Expand **Metadata** and **MDM connections**.
2. Right-click the connection to which you want to add a task, and then select **Add task...** from the contextual menu.

   The **[Properties]** dialog box opens showing the metadata of the selected connection.



3. In the **Description** field, enter a short description for the task you want to attach to the selected connection.
4. On the **Priority** list, select the priority level and then click **OK** to close the dialog box. The created task is added to the **Tasks** list.

You can follow the same steps in the above procedure to add a task to an entity or column in the connection.

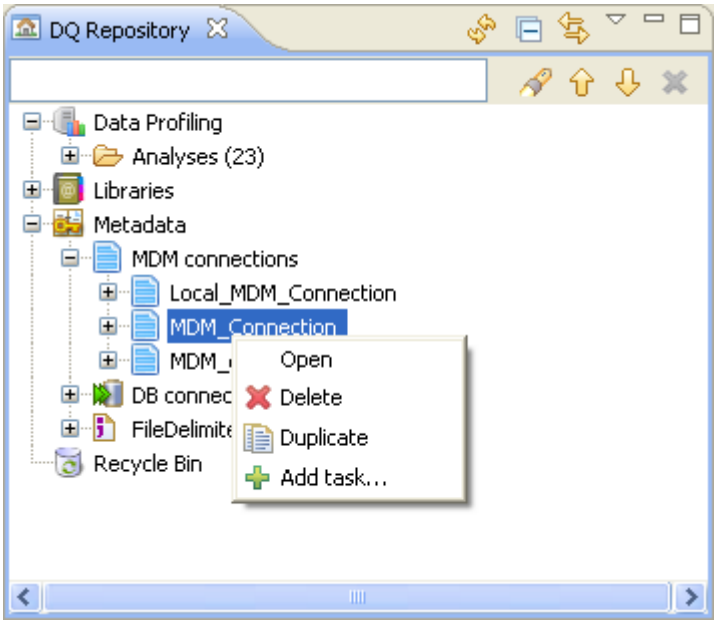
For more information on how to access the task list, see section *Displaying the task list*.

3.2.2.4. How to delete or restore an MDM connection

Prerequisite(s): An MDM connection is created in the **Profiling** perspective of the studio. For further information, see section *Connecting to an MDM server*.

To delete an MDM connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > MDM Connections**.
2. Right-click the MDM connection you want to delete and select **Delete** from the contextual menu.



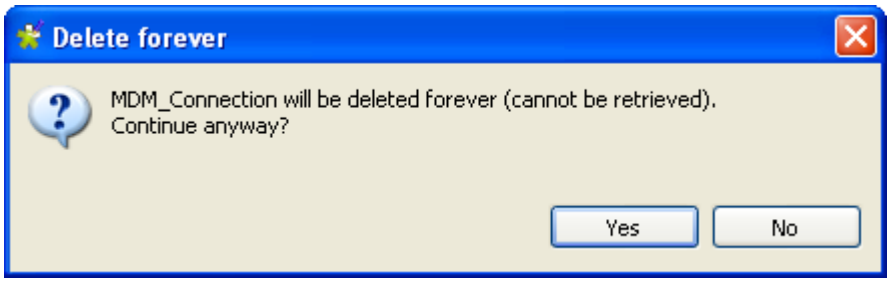
The MDM connection is moved to the **Recycle Bin**.

You will always be able to run any analysis that uses the connection moved to the recycle bin. However, an alert message will be displayed next to the connection name in the analysis editor.

To delete it from the **Recycle Bin**, do the following:

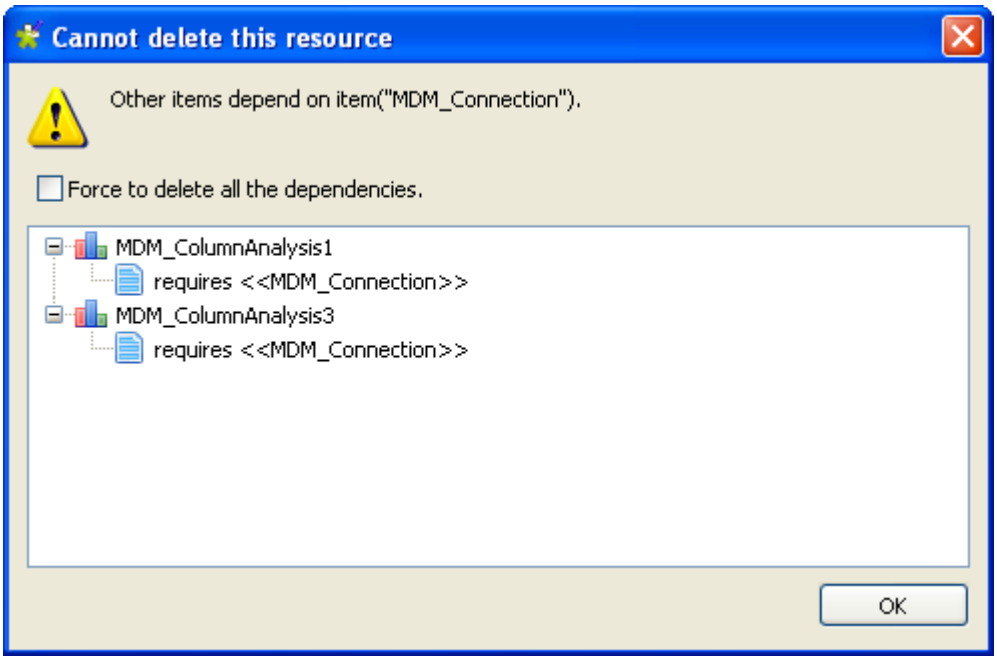
1. Right-click it in the **Recycle Bin** and choose **Delete** from the contextual menu.

   If the connection is not used by any analysis in the current Studio, a **[Delete forever]** dialog box is displayed.



2. Click **Yes** to confirm the operation.

   If the connection is used by one or more analyses in the current Studio, a dialog box is displayed to list all the analyses that use this MDM connection.



3. Either:

   - Click **OK** to close the dialog box without deleting the MDM connection from the recycle bin.
   - Select the **Force to delete all the dependencies** check box and then click **OK** to delete the connection from the **Recycle Bin** and to delete all the dependent analyses from the **Data Profiling** node.

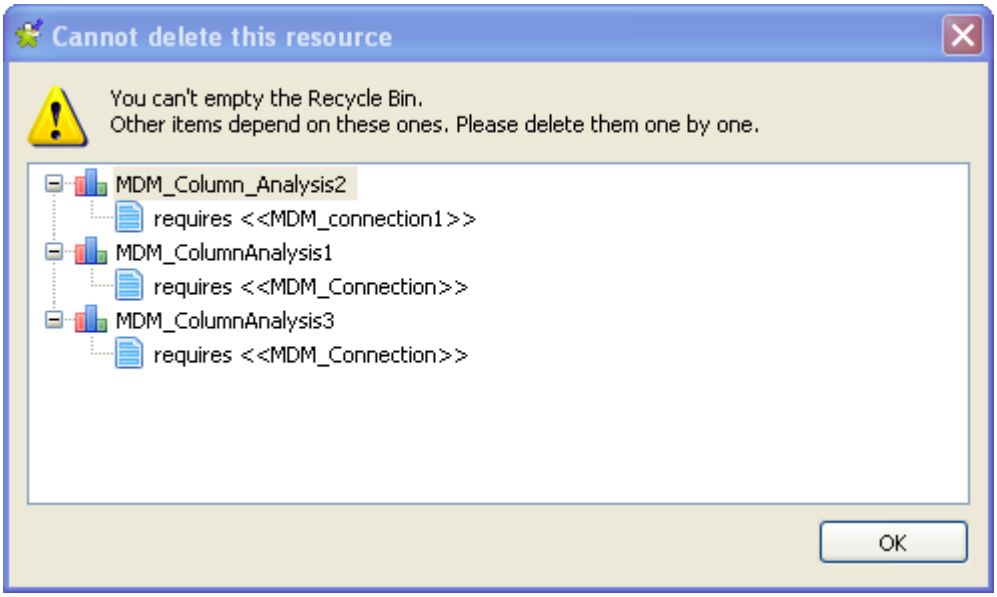
You can also delete permanently the MDM connection by emptying the recycle bin. To empty the **Recycle Bin**, do the following:

1. Right-click the **Recycle Bin** and select **Empty recycle bin**.

   If the connection is not used by any analysis in the current Studio, a confirmation dialog box is displayed.

2. Click **Yes** to empty the recycle bin.

   If the connection is used by one or more analyses in the current Studio, a dialog box is displayed to list all the analyses that use the MDM connection.



3. Click **OK** to close the dialog box without removing the connection from the recycle bin.

To restore an MDM connection from the **Recycle Bin**, do the following:

- In the **Recycle Bin**, right-click the connection and select **Restore**.

  The MDM connection is moved back to the **Metadata** node.

3.2.3. Managing file connections

Many management options are available for file connections including editing and duplicating the connection or adding a task to it.

The procedures to manage file connections are the same as those for managing MDM connections. For further information, see section *Managing MDM connections*.

3.3. Catalogs and schemas in database systems

The structure of a database defines how objects are organized in the database. Different data storage structures are used to store objects in databases. For example, the highest level structure “Catalog” followed by “Schema” and finally by “Table”. The tables are not applicable to all database types.

The table below describes the structure of some databases in terms of catalog and schemas:

DB name	Version	Catalog	Schema
Oracle		no	yes
MySQL		yes	no
SQLServer	2000/2005/2008	yes	yes
DB2		no	yes
DB2 ZOS		no	yes
Sybase		yes	yes
Informix		yes	yes
PointBase		no	yes
PostgreSQL		yes	yes
AS400	V5R4	yes	yes
Ingres		no	yes
Teradata		no	yes
Netezza		yes	yes
SQLite		no	no

Chapter 4. Profiling database content

This chapter provides the information you need to analyze database content to have an overview of the number of tables in the database, rows per table and indexes and primary keys.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

4.1. Managing database content analyses

You can analyze the content of a database to have an overview of the number of tables in the database, rows per table and indexes and primary keys.

You can also analyze one specific catalog or schema in a database, if this entity is used in the physical structure of the database.

4.1.1. Creating a database content analysis

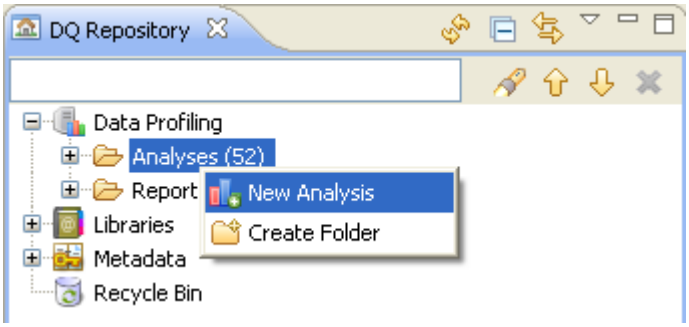
From the studio, you can create an analysis of the content of a given database.

Prerequisite(s): At least, one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

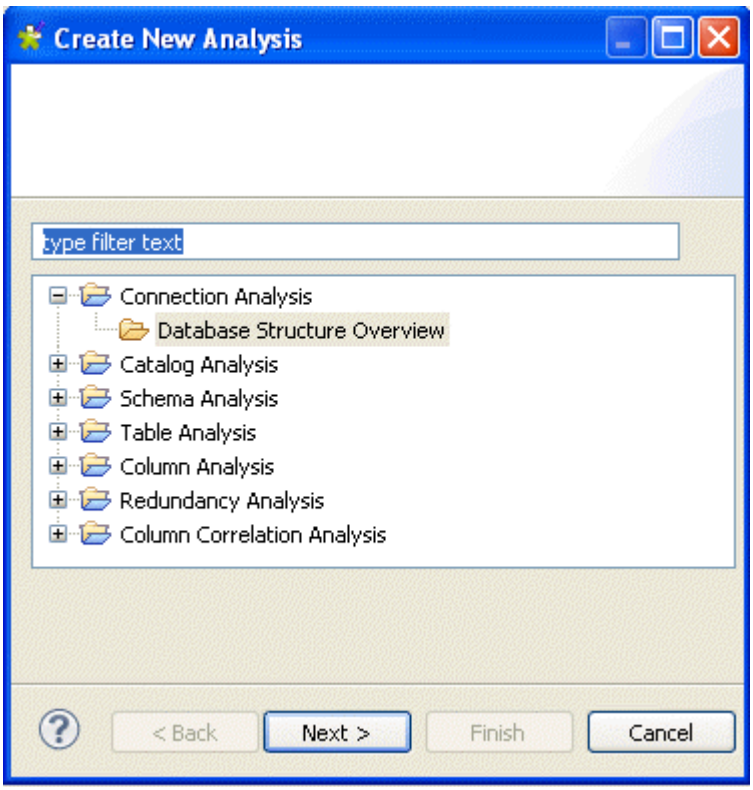
To create a database content analysis, you must first define the relevant analysis and then select the database connection you want to analyze.

Defining the analysis

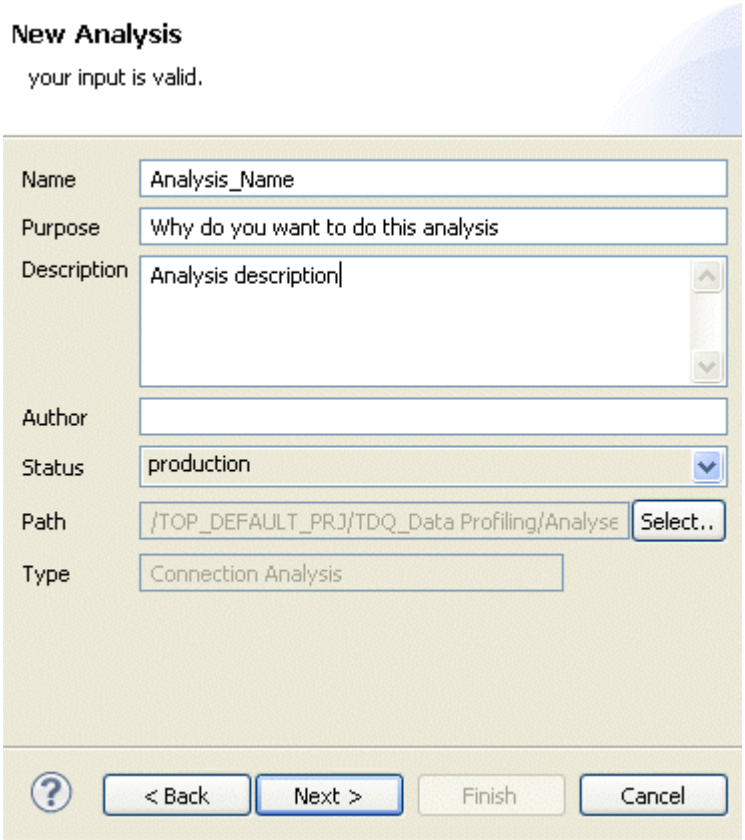
1. In the **DQ Repository** tree view, expand **Data Profiling**.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.

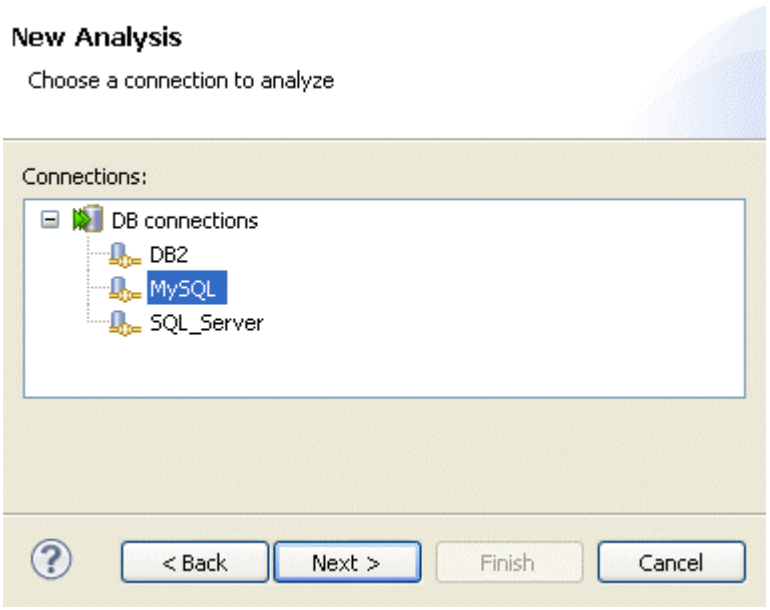


3. Expand the **Connection Analysis** node, click **Database Structure Overview** and then click the **Next** button.



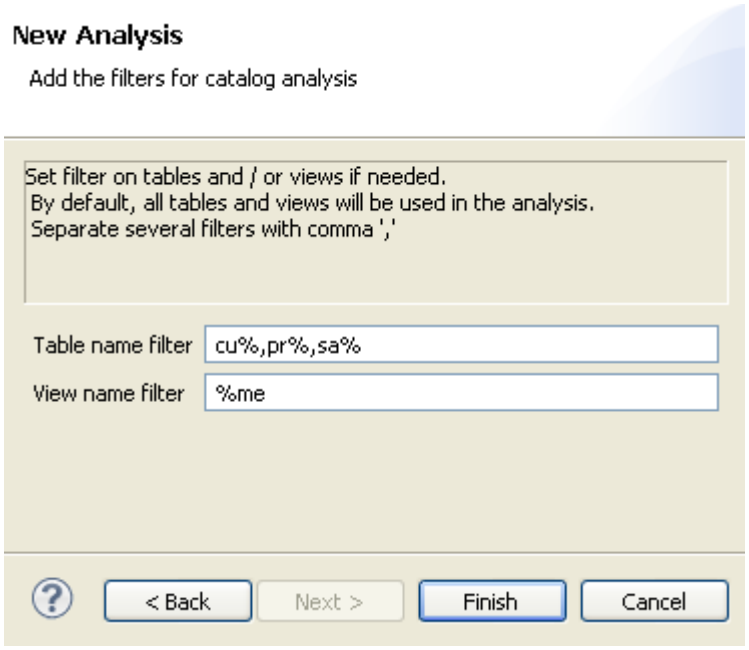
4. In the **Name** field, enter a name for the current analysis.

5. Set the analysis metadata (purpose, description and author name) in the corresponding fields and click **Next**.



Selecting the database connection you want to analyze

1. Expand **DB Connections** and select a database connection to analyze, if more than one exists.
2. Click **Next** to proceed to the next step.

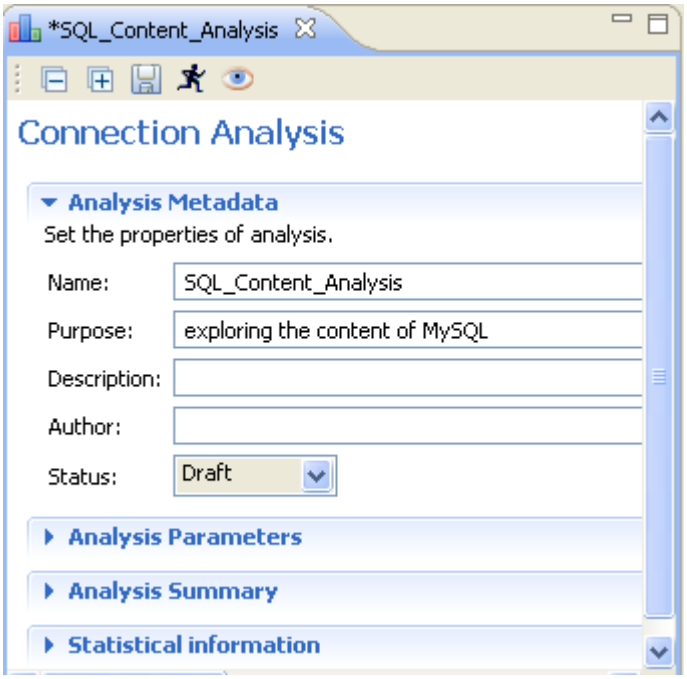


3. Set filters on tables and/or views in their corresponding fields according to your needs using the SQL language.

   By default, the analysis will include all tables and views in the database.

4. Click **Finish** to close the **[Create New Analysis]** wizard.

   A folder for the newly created analysis is listed under the **Analyses** folder in the **DQ Repository** tree view, and the connection editor opens with the defined metadata.



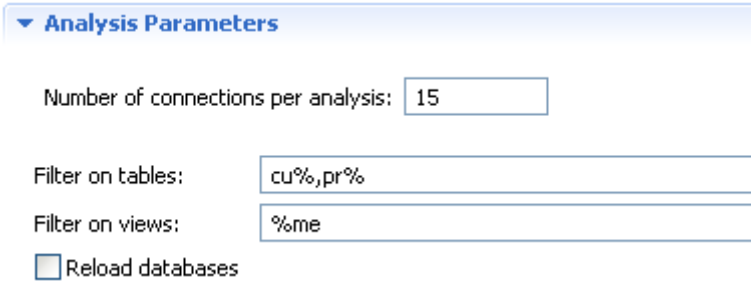
The display of the connection editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

5. Click **Analysis Parameters** and:

   - In the **Number of connections per analysis** field, set the number of concurrent connections allowed per analysis to the selected database connection.

     You can set this number according to the database available resources, that is the number of concurrent connections each database can support.

   - Check/modify filters on table and/or views, if any.



   - Select the **Reload databases** check box if you want to reload all databases in your connection on the server when you run the overview analysis.

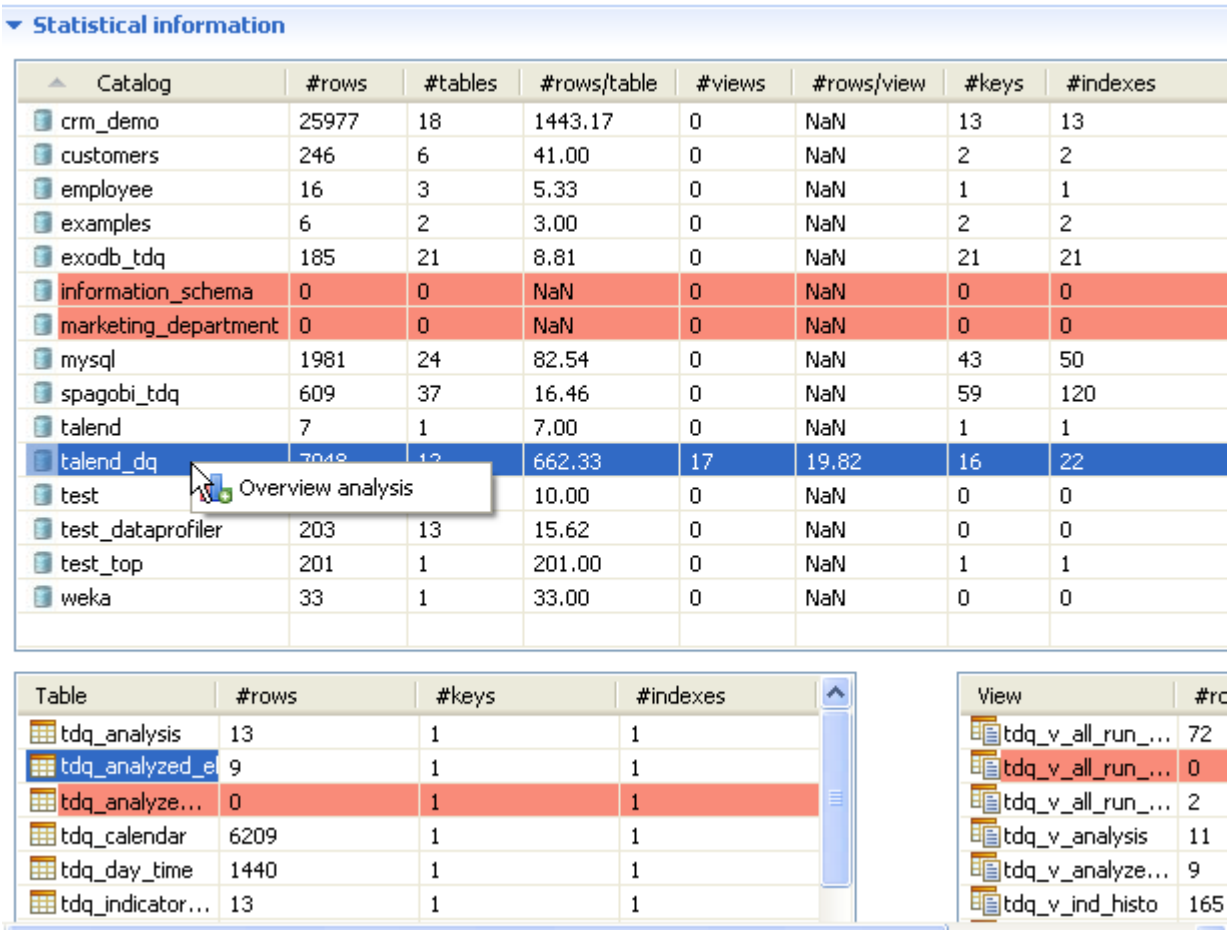
     When you try to reload a database, a message will prompt you for confirmation as any change in the database structure may affect existing analyses.

6. Click **Analysis Summary** to show all the parameters of the current analysis along with the current analysis execution status.

7. Click the save icon on top of the editor and then press **F6** to execute the current analysis. A message opens to confirm that the operation is in progress.

   Analysis results are stored in the **Statistical information** view.

8. Click **Statistical information** to show analytical information about the content of the relevant database.



9. Click a catalog or a schema in the **Statistical information** view to list all tables included in the selected catalog or schema along with a summary of their content: number of rows, keys and indexes.

   The selected catalog or schema is highlighted in blue. Catalogs or schemas highlighted in red indicate potential problems in data.

10. Click any column header in the analytical table to sort alphabetically data listed in catalogs or schemas.

    You can also sort alphabetically all columns in the result table doing the same.

You can create catalog, schema or table analysis directly from the open connection analysis if you right-click the desired catalog, schema or table and select **Overview analysis** or **Table analysis**.

4.1.2. Creating a database content analysis in shortcut procedure

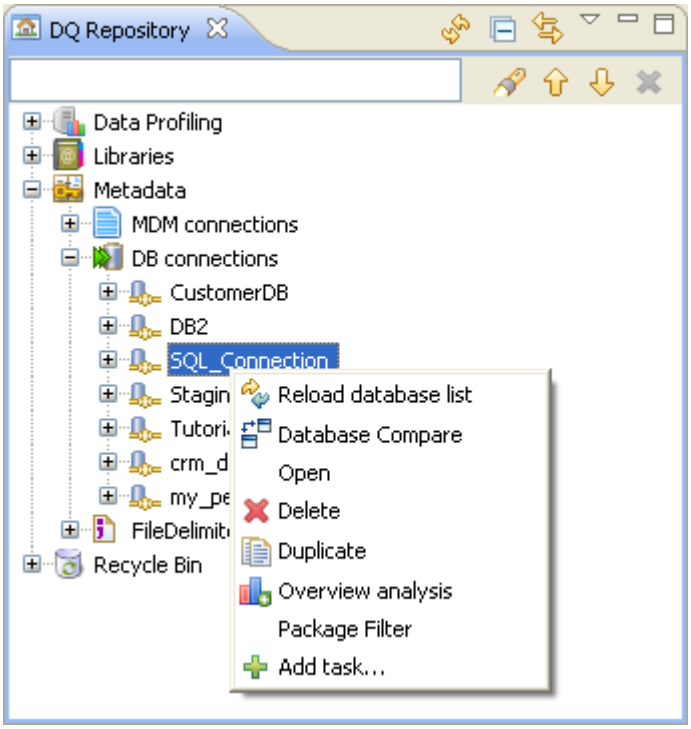
You can create an analysis of the content of a given database directly from the **DB Connection** folder in the **DQ Repository** tree view.

Prerequisite(s): At least, one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

To create a database content analysis, do the following:

1. Right-click the database for which you want to create content analysis.

2. From the contextual menu, select **Overview analysis** .



This way, you do not have to specify in the new analysis wizard either the type of analysis you want to carry out or the DB connection to analyze. Otherwise, all other procedural steps are exactly the same as in section *Creating a database content analysis*.

4.1.3. Creating a catalog analysis

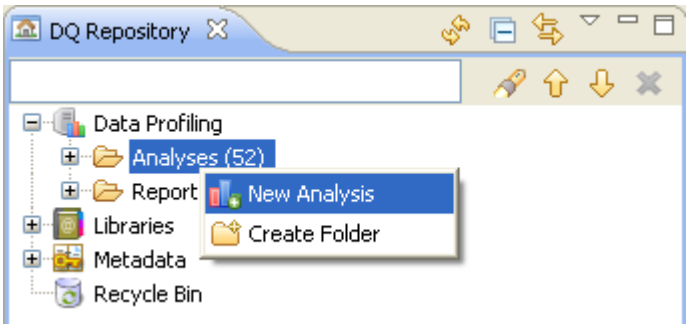
You can analyze one specific catalog in a database, if this entity is used in the physical structure of the database. The result of the analysis gives analytical information about the content of this catalog, for example number of rows, number of tables, number of rows per table and so on.

Prerequisite(s): At least one database connection has been created to connect to a database that uses the “catalog” entity.

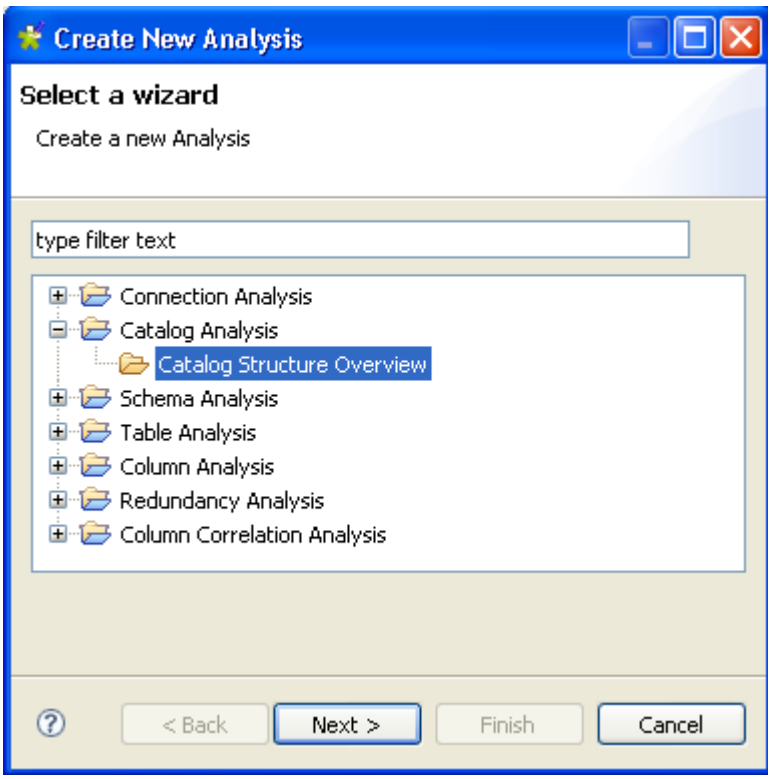
Defining the analysis

1. In the **DQ Repository** tree view, expand **Data Profiling**.

2. Right-click the **Analyses** folder and select **New Analysis**.

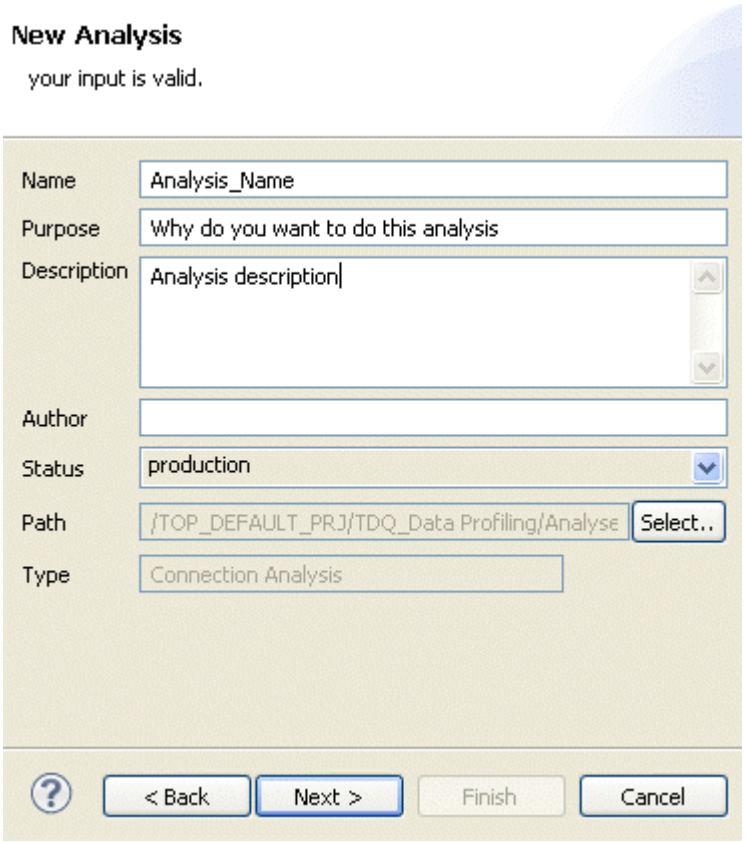


The **[Create New Analysis]** wizard opens.



3. Expand the **Catalog Analysis** node and then click **Catalog Structure Overview**.

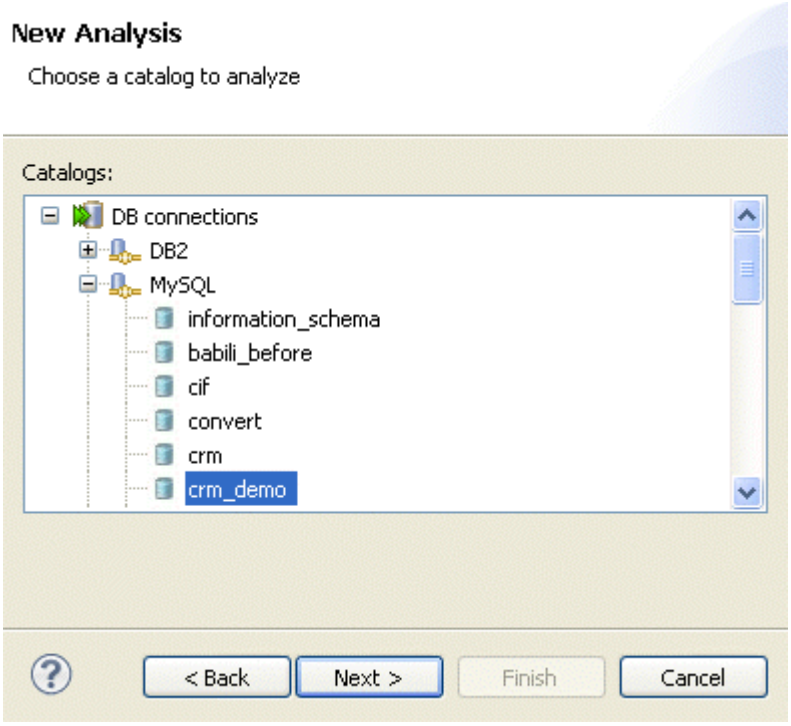
4. Click the **Next** button.



You can directly go to this step in the analysis creation wizard if you right-click the catalog to analyze in **Metadata>DB Connections** and select **Overview analysis**.

5. In the **Name** field, enter a name for the current analysis.

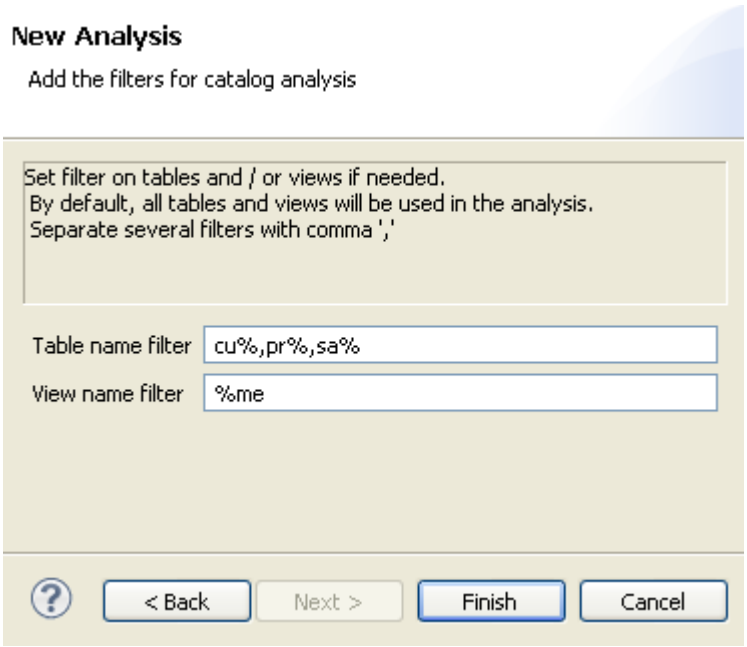
6. Set the analysis metadata (purpose, description and author name) in the corresponding fields and click **Next**.



Selecting the catalog you want to analyze

1. Expand **DB Connections** and the database that include catalog entities in its physical structure and select a catalog to analyze.

2. Click **Next**.

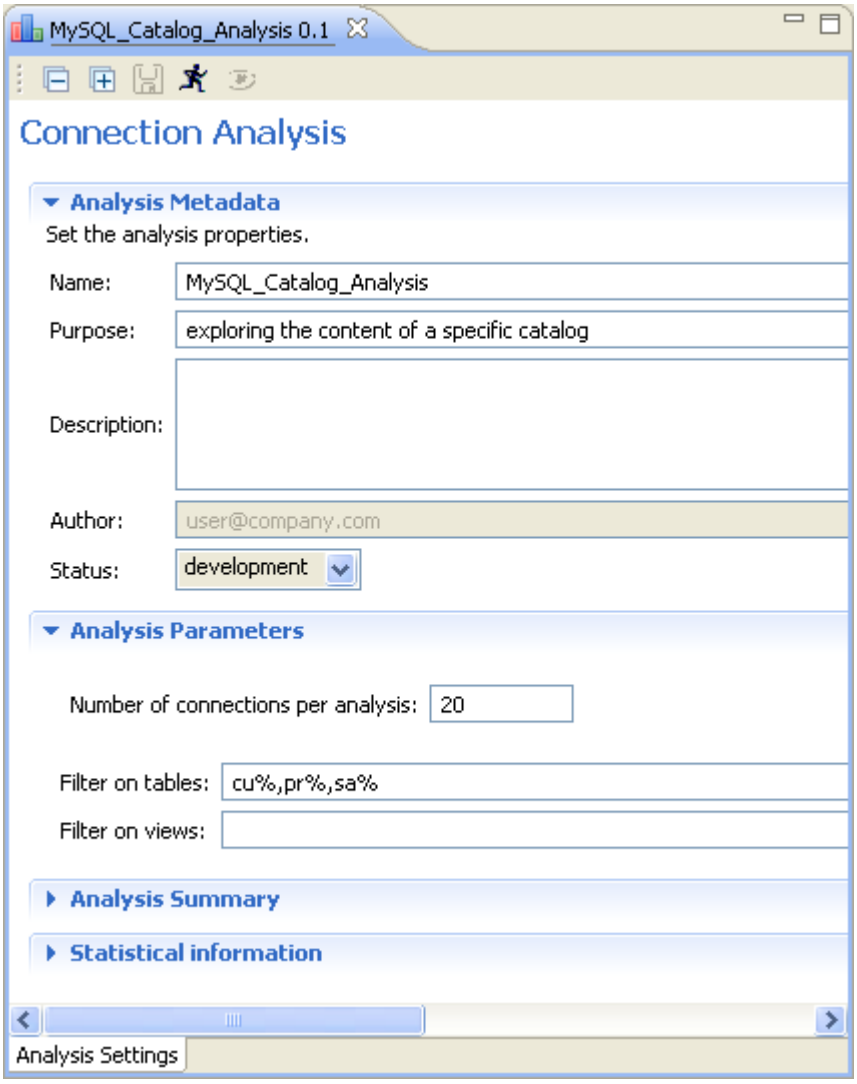


3. Set filters on tables and/or views in their corresponding fields according to your needs using the SQL language.

   By default, the analysis will include all tables and views in the catalog.

4. Click **Finish** to close the **[Create New Analysis]** wizard.

   A folder for the newly created analysis is listed under **Analysis** in the **DQ Repository** tree view, and the analysis editor opens with the defined metadata.



The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

5. Click **Analysis Parameters** and:

   - In the **Number of connections per analysis** field, set the number of concurrent connections allowed per analysis to the selected database connection.

     You can set this number according to the database available resources, that is the number of concurrent connections each database can support.

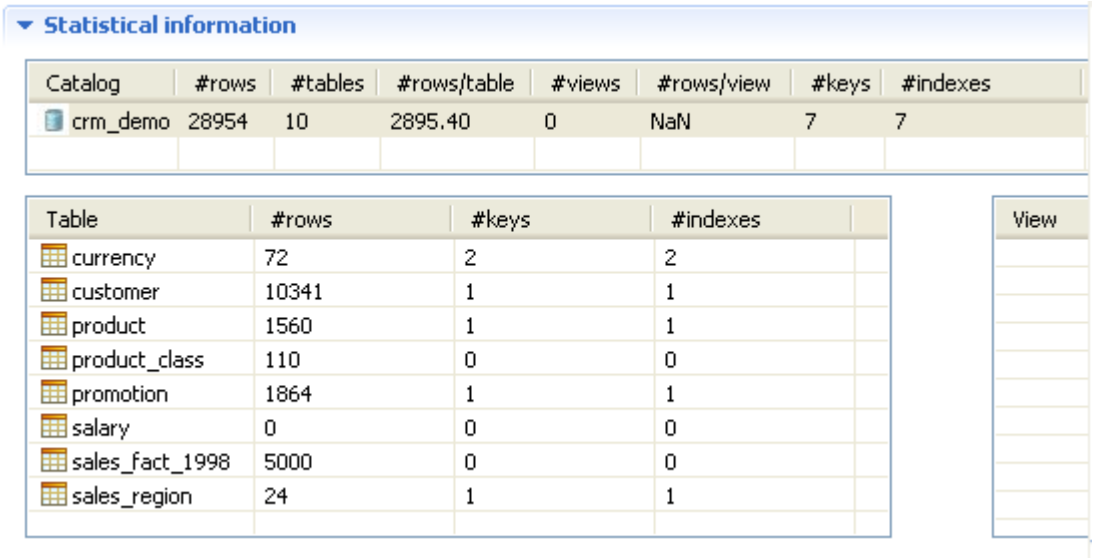
   - Check/modify filters on table and/or views, if any.

6. Click the save icon on top of the editor and then press **F6** to execute the current analysis.

   A message opens to confirm that the operation is in progress.

   Analysis results are stored in the **Statistical informations** view.

7. Click **Statistical informations** to show analytical information about the content of the relevant catalog.



8. If required, click the catalog in the analytical table to open a result list that details all tables included in the selected catalog with a summary of their content.

The selected catalog is highlighted in blue. Catalogs highlighted in red indicate potential problems in data.

9. If required, click any column header in the result table to sort the listed data alphabetically.

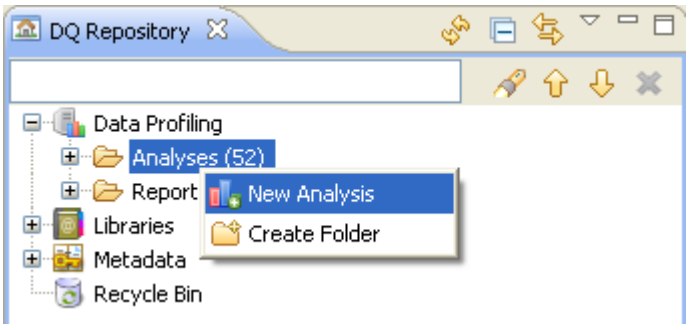
4.1.4. Creating a schema analysis

You can use the **Profiling** perspective of the studio to analyze one specific schema in a database, if this entity is used in the physical structure of the database. The result of the analysis gives analytical information about the content of this schema, for example number of rows, number of tables, number of rows per table and so on.

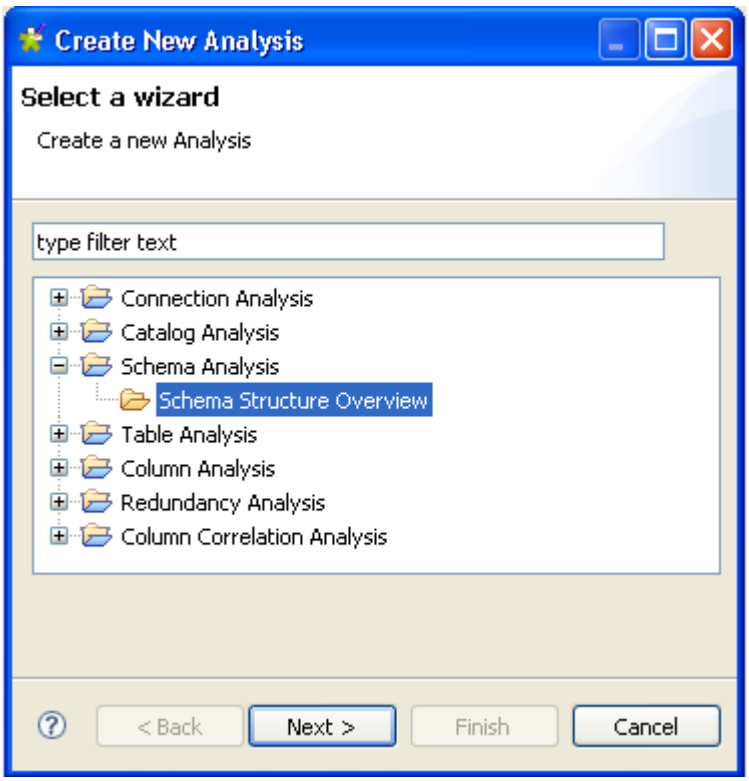
Prerequisite(s): At least one database connection has been created to connect to a database that uses the “schema” entity, for example the DB2 database. For further information, see section *Connecting to a database*.

Defining the analysis

1. In the **DQ Repository** tree view, expand **Data Profiling**.
2. Right-click the **Analyses** folder and select **New Analysis**.

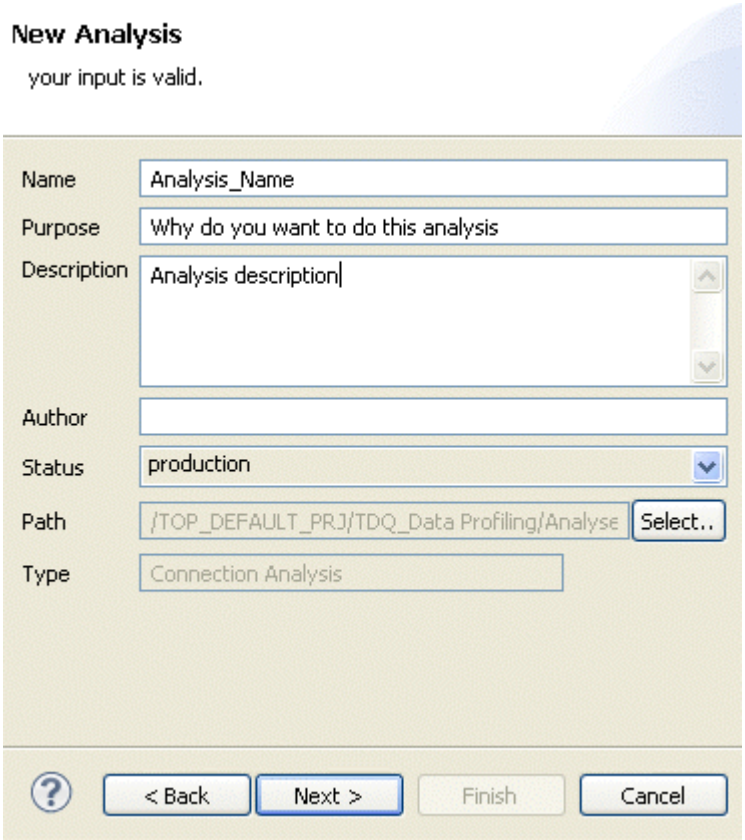


The **[Create New Analysis]** wizard opens.



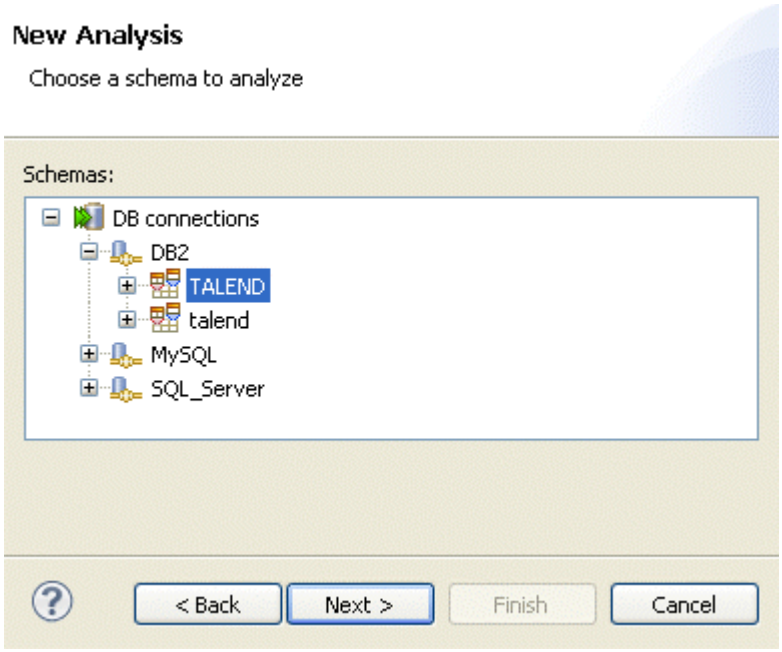
3. Expand the **Schema Analysis** node and then click **Schema Structure Overview**.

4. Click the **Next** button to proceed to the next step.



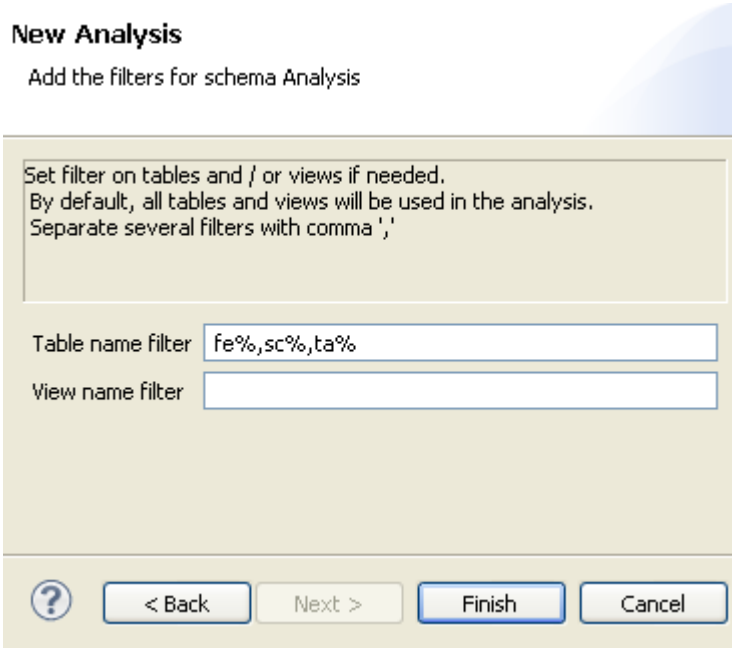
You can directly get to this step in the analysis creation wizard if you right-click the schema to analyze in **Metadata >DB connections** and select **Overview analysis**.

5. In the **Name** field, enter a name for the current analysis.
6. If required, set the analysis metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



Selecting the schema you want to analyze

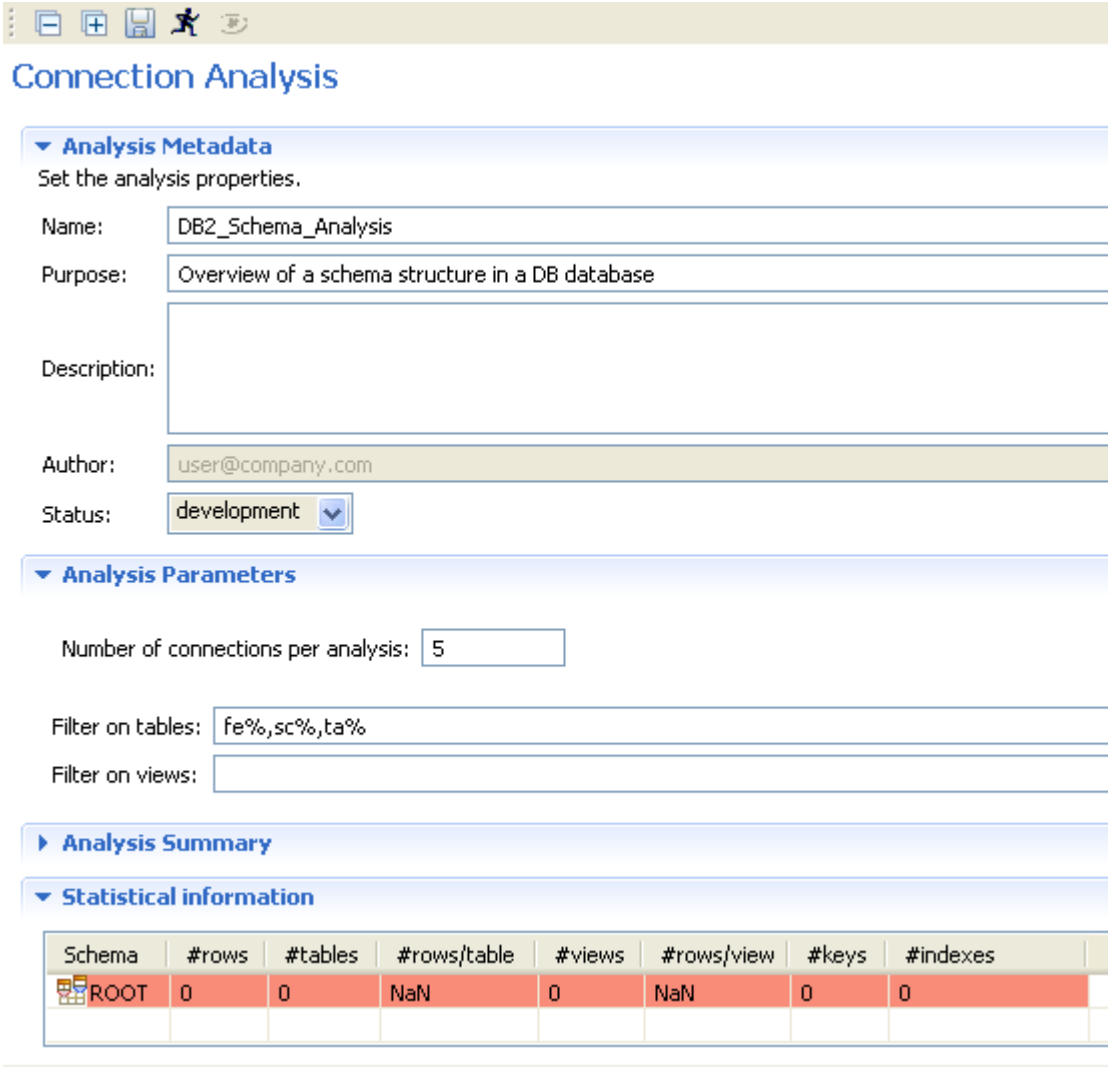
1. Expand in succession **DB Connections** and the database that include schema entities in its physical structure and select a schema to analyze.
2. Click **Next**.



3. Set filters on tables and/or views in their corresponding fields according to your needs using the SQL language.

   By default, the analysis will include all tables and views in the catalog.
4. Click **Finish** to close the **[Create New Analysis]** wizard.

A folder for the newly created analysis is listed under **Analysis** in the **DQ Repository** tree view, and the analysis editor opens with the defined metadata.



The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

5. Click **Analysis Parameters** and:

   - In the **Number of connections per analysis** field, set the number of concurrent connections allowed per analysis to the selected database connection.

     You can set this number according to the database available resources, that is the number of concurrent connections each database can support.

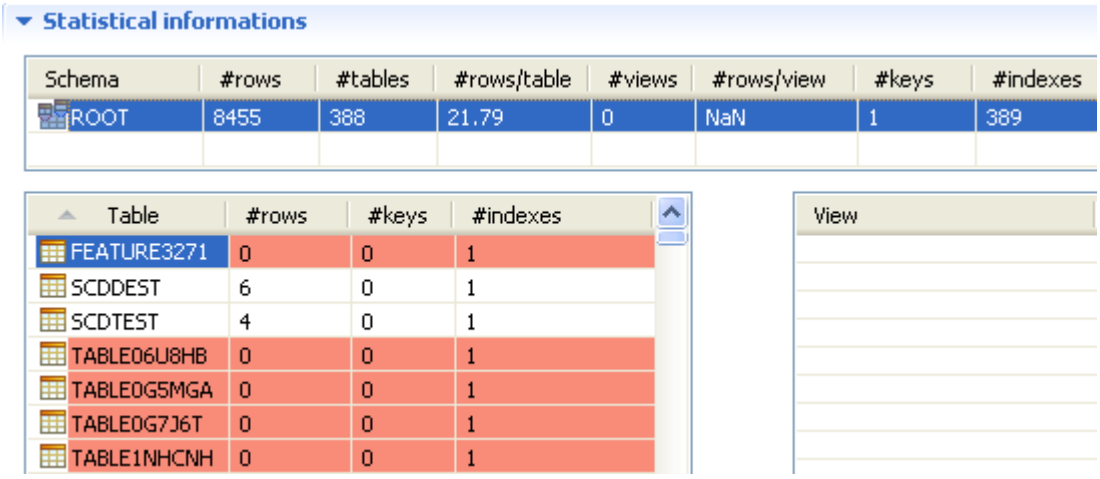
   - Check/modify filters on table and/or views, if any.

6. Click the save icon on top of the editor and then press **F6** to execute the current analysis.

   A message opens to confirm that the operation is in progress.

   Analysis results are stored in the **Statistical informations** area.

7. Click **Statistical informations** to show analytical information about the content of the relevant catalog.



8. Click the schema in the analytical table to open a result list that details all tables included in the selected schema with a summary of their content.

   The selected schema is highlighted in blue. Schemas highlighted in red indicate potential problems in data.

9. Click any column header in the result table to sort the listed data alphabetically.

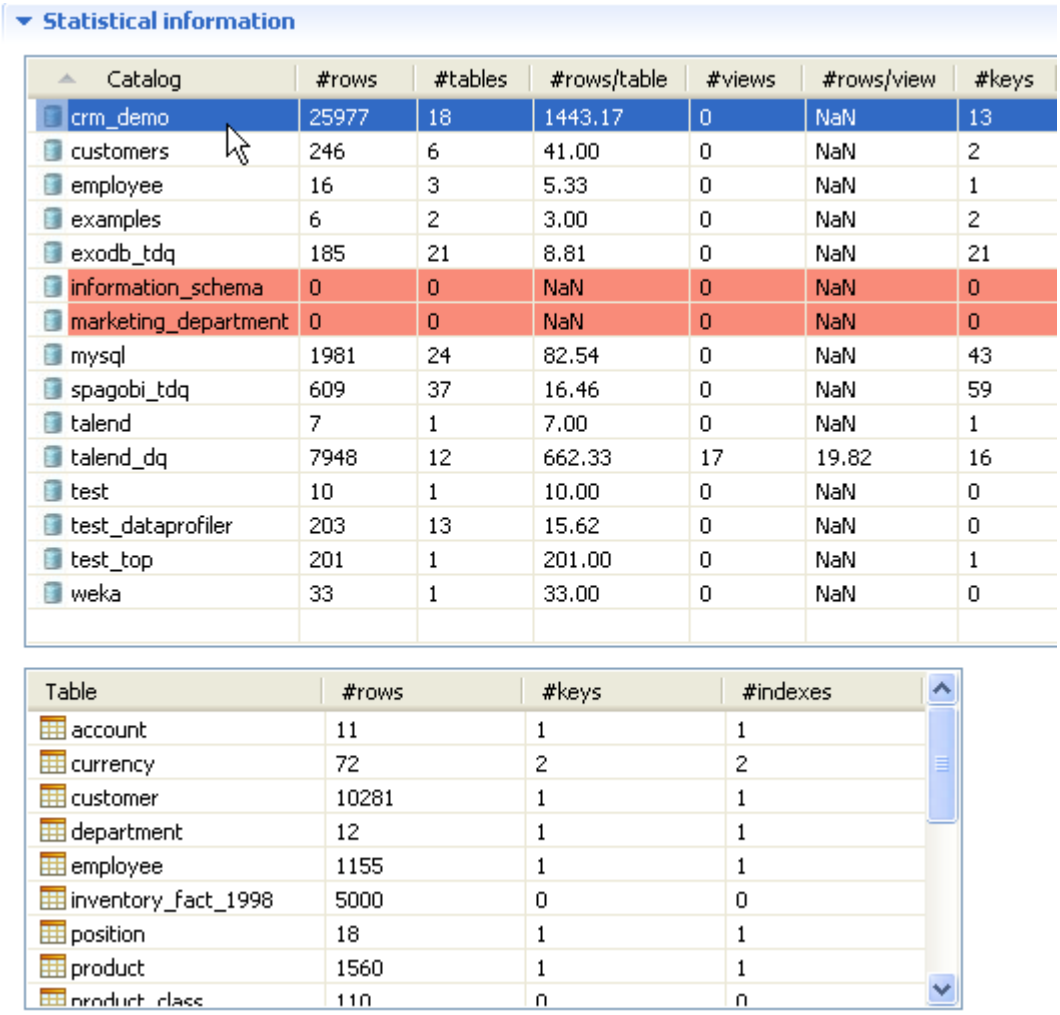
4.2. Displaying a table key and index in the analyzed database

After analyzing the content of a database as outlined in section *Creating a database content analysis*, you can display the details of the key and index of a given table. This information could be very interesting for the database administrator.

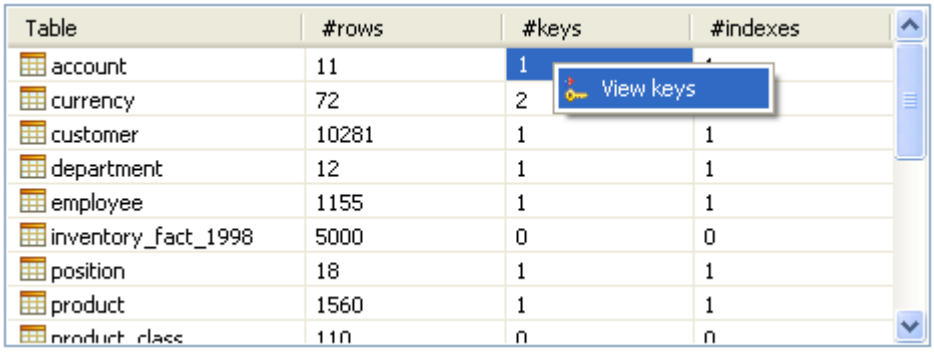
Prerequisite(s): At least one database content analysis has been created and executed in the studio.

To display the details of the key and index of a given table in the analyzed database, do the following:

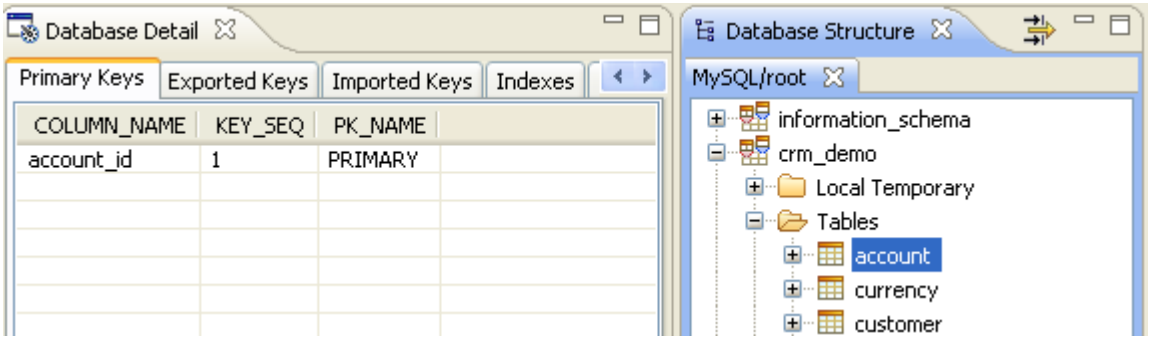
1. In the **Statistical information** view, click a catalog or a schema. All the tables included in the selected catalog or schema are listed along with a summary of their content: number of rows, keys and indexes.



2. In the table list, right-click the table key and select **View keys**.

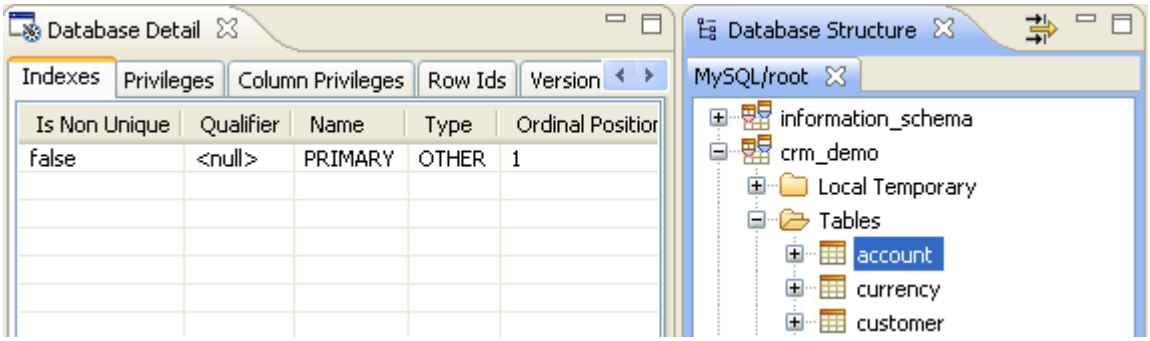


The **Database Structure** and the **Database Detail** views display the structure of the analyzed database and information about the primary key of the selected table.



If one or both views do not show, select **Window** > **Show View** > **Database Structure** or **Window** > **Show View** > **Database Detail**.

3. In the table list, right-click the table index and select **View indexes**.



The **Database Structure** and the **Database Detail** views display the structure of the analyzed database and information about the index of the selected table.

4. If required, click any of the tabs in the **Database Detail** view to display the relevant metadata about the selected table.

4.3. Tracking data changes in source databases

When the data in a source database is changed or updated, it is necessary that the relevant connection structure in the studio follows that change or update as well. Otherwise, errors may occur when trying to analyze a column that has been modified/deleted in a database.

From the studio, you can compare the connection structure displayed in the **DQ Repository** tree view with the database structures itself to locate possible differences. Then you can synchronize the connection structure in the tree view with the actual database structure.

Comparing and synchronizing a database connection with a database structure may take long time. Do not do it unless you are sure that incoherency does exist.

4.3.1. Comparing tree-view metadata structures with database structures

You can quickly and accurately compare the metadata lists displayed in the **DQ Repository** tree view with the database structures on which you create the connection to indicate any incoherences.

The studio takes a connection structure in the **DQ Repository** tree view and compares it to the database trying to locate all structure differences and display these differences in the **Compare** view.

You can then, if necessary, synchronize the connection structure in the tree view with the database structure. For more information, see section *Synchronizing the connection structure with the database structure*.

You can perform the structure comparison at the following three different levels:

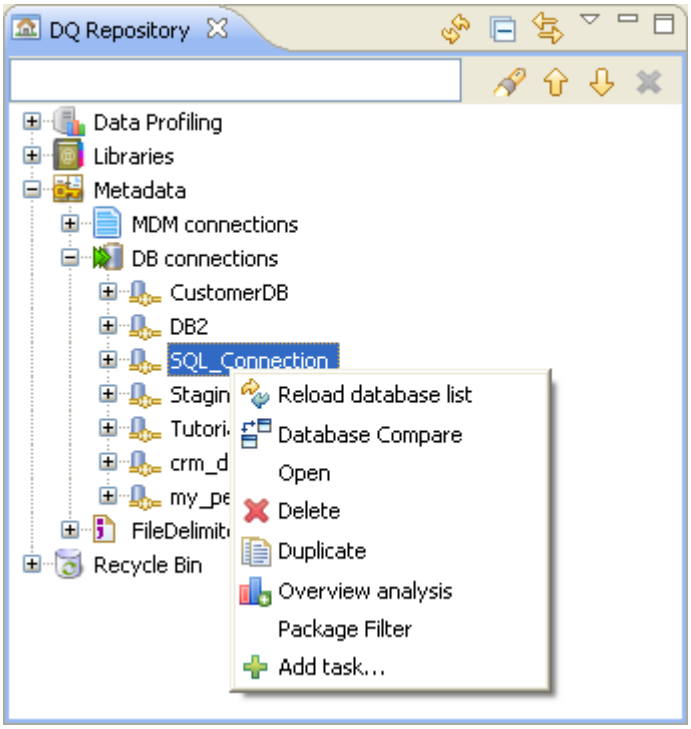
- **DB connection**: to compare the catalog and schema lists,
- **Tables**: to compare the list of tables,
- **Column**: to compare the list of columns.

4.3.1.1. How to compare catalog and schema lists

Prerequisite(s): A database connection has been already created in the **Profiling** perspective of the studio.

To compare the catalog and schema lists, do the following:

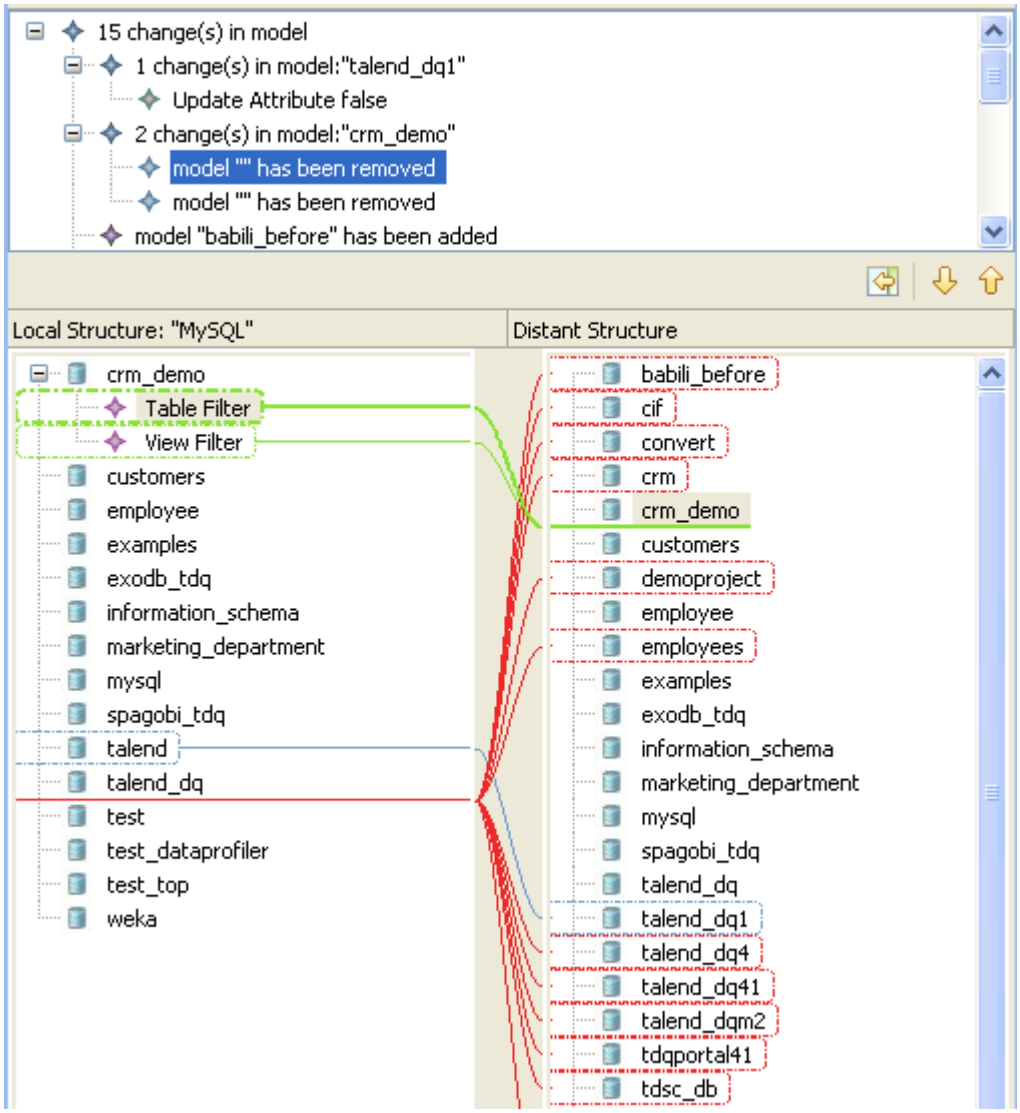
1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Right-click the DB connection for which you want to compare the metadata structure with the database structure and select **Database Compare**.



A message opens to confirm that the operation is in progress.

3. If required, click the **Cancel** button on the message to stop the operation.

A compare view opens displaying the differences between your connection structure and the actual database structure.



In the compare view, colors are used as the following:

Color	Indication
green	highlights any deleted item.
blue	highlights any updated item.
red	highlights any added item.

If you select an item in the top half of the view, the color markers in the bottom half of the view become thicker to highlight the selected item. If you select any database from the **Distant Structure** list in the bottom half of the view, the corresponding description will be highlighted in the top half of the view.

4. If required, right-click a specific catalog in this view to display a contextual menu where you can select **Compare the list of tables** or **Compare the list of views**. This will display respectively the table list or the view list of the selected catalog. For further information about comparing table lists, see section *How to compare table lists*

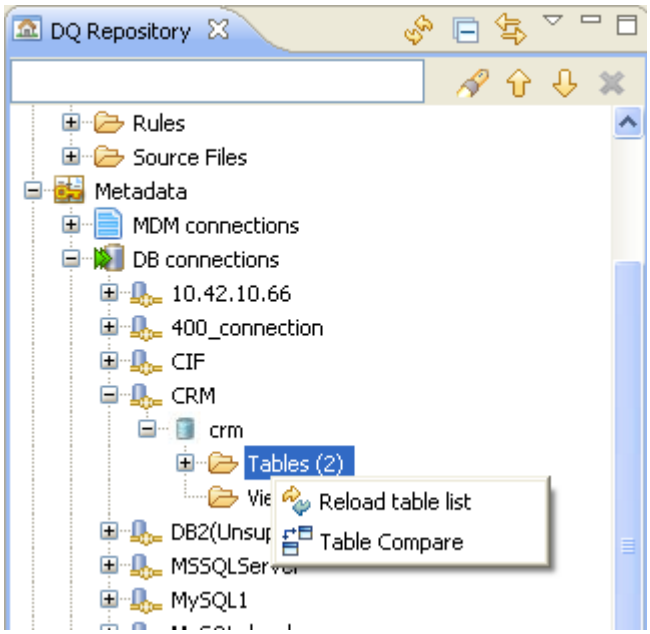
If you select a specific catalog in this list and press the **T** or **V** keys on your keyboard, you can display respectively the table or view lists of the selected catalog.

4.3.1.2. How to compare table lists

Prerequisite(s): A DB connection has already been created in the **Profiling** perspective in the studio.

To compare a table list, do the following:

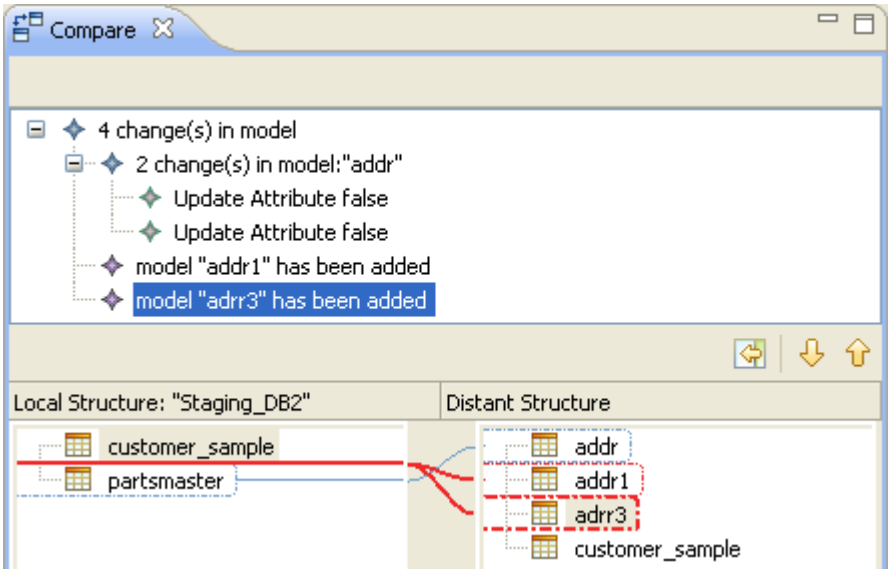
1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Browse through the entities in your database connection to reach the **Table** folder you want to compare with that of the database.
3. Right-click the **Tables** folder and select **Table Compare**.



A message opens to confirm that the operation is in progress.

You can click the **Cancel** button on the confirmation message to stop the operation.

The **Compare** view opens displaying any differences between the table lists in the tree view and the actual database.



In the compare view, colors are used as the following:

Color	Indication
green	highlights any deleted item.
blue	highlights any updated item.
red	highlights any added item.

If you select an item in the top half of the view, the color markers in the bottom half of the view become thicker to highlight the selected item. If you select any database from the **Distant Structure** list in the bottom half of the view, the corresponding description will be highlighted in the top half of the view.

4. If required, right-click a specific table in the **Compare** view to display a contextual menu. Select **Compare the list of columns** to display the columns list of the selected table. For further information, see section *How to compare column lists*

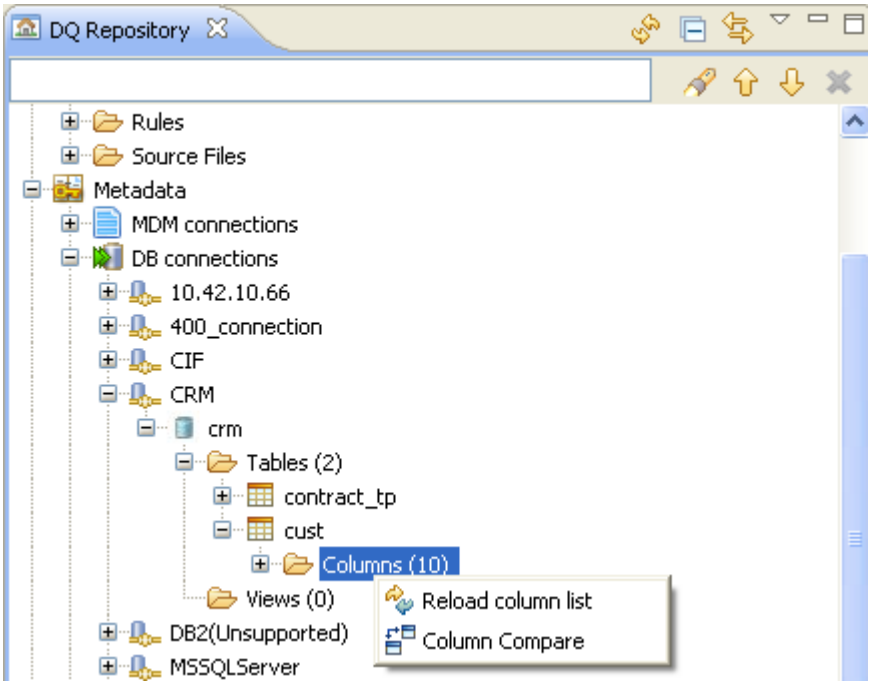
If you select a specific table in the **Compare** list and press the **C** key on your keyboard, you can display the column list of the selected table.

4.3.1.3. How to compare column lists

Prerequisite(s): A database connection has been created in the **Profiling** perspective in the studio.

To compare a column list, do the following:

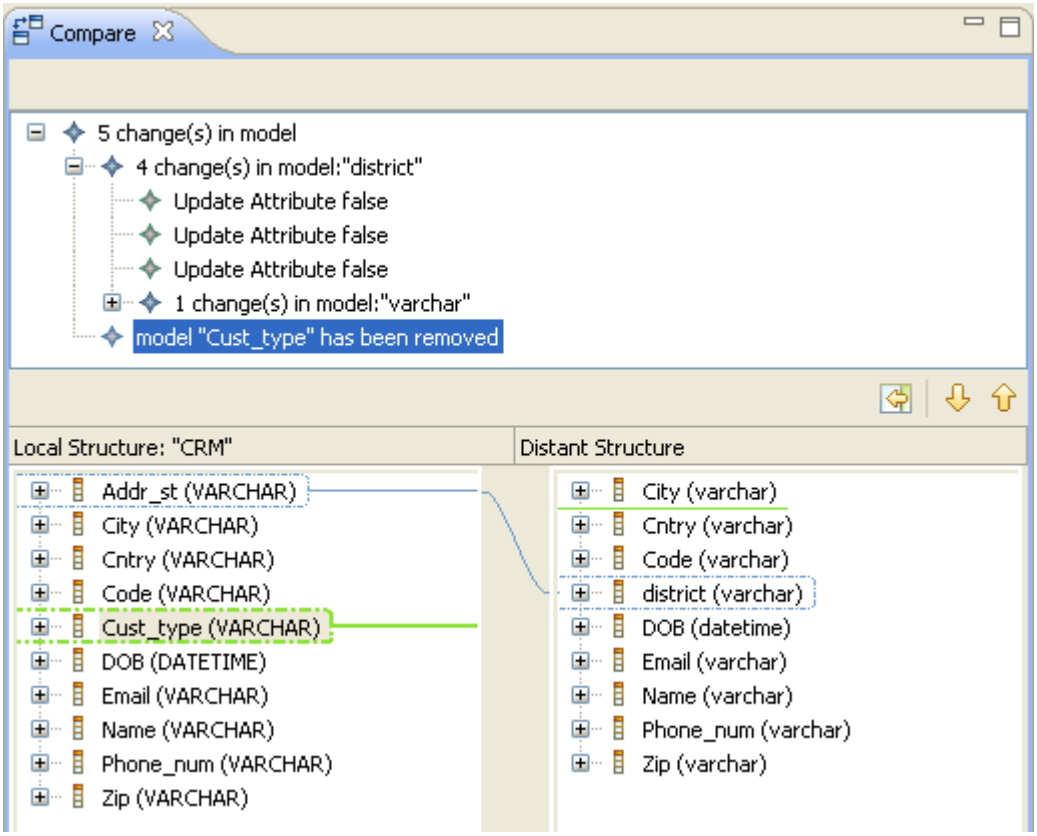
1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Browse through the entities in your database connection to reach the **Columns** folder you want to compare with that of the database.
3. Right-click the **Columns** folder and select **Column Compare**.



A progress information pop-up opens to confirm that the operation is in progress.

You can click the **Cancel** button on the confirmation message to stop the operation.

The **Compare** view opens displaying any differences between the column list in the tree view and the database.



In the compare view, colors are used as the following:

Color	Indication
green	highlights any deleted item.
blue	highlights any updated item.
red	highlights any added item.

If you select an item in the top half of the view, the color markers in the bottom half of the view become thicker to highlight the selected item. If you select any database from the **Distant Structure** list in the bottom half of the view, the corresponding description will be highlighted in the top half of the view.

4.3.2. Synchronizing the connection structure with the database structure

You can synchronize the connection structure displayed in the **DQ Repository** tree view with the database structures to eliminate any incoherences. You can perform synchronization at the following three different levels:

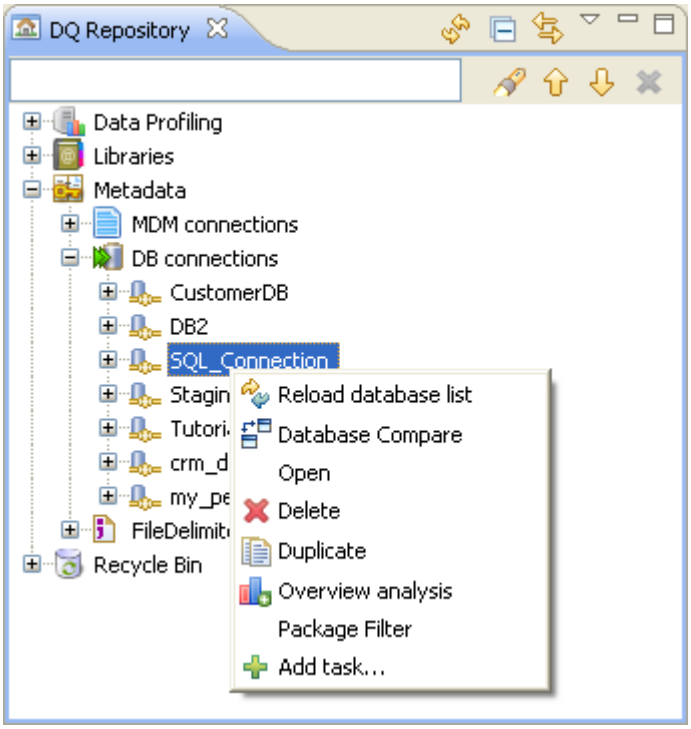
- **DB connection**: to refresh the catalog and schema lists,
- **Tables**: to refresh the list of tables,
- **Column**: to refresh the list of columns.

4.3.2.1. How to synchronize catalog and schema lists

Prerequisite(s): A DB connection has been created in the studio.

To synchronize the catalog and schema lists, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Right-click the DB connection you want to synchronize with the database and select **Reload database list**.



   A message will prompt you for confirmation as any change in the database structure may affect the analyses listed in the Studio.

3. Click **OK** to close the confirmation message, or **Cancel** to stop the operation.

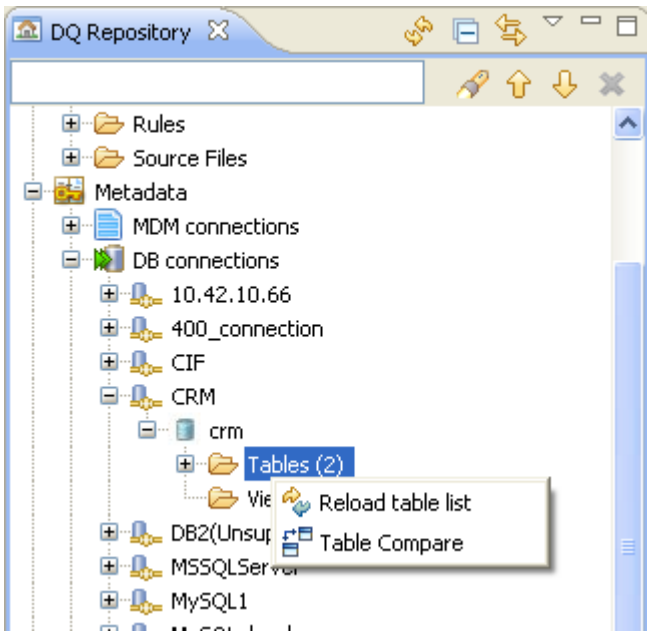
   The selected database connection is updated with the new catalogs and schemas, if any.

4.3.2.2. How to synchronize table lists

Prerequisite(s): A DB connection has already been created in the **Profiling** perspective in the studio.

To synchronize a table list, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections** .
2. Browse through the entities in your database connection to reach the **Table** folder you want to synchronize with the database.
3. Right-click the **Tables** folder and select **Reload table list**.



A message will prompt you for confirmation as any change in the database structure may affect existing analyses.

4. Click **OK** to close the confirmation message, or **Cancel** to stop the operation.

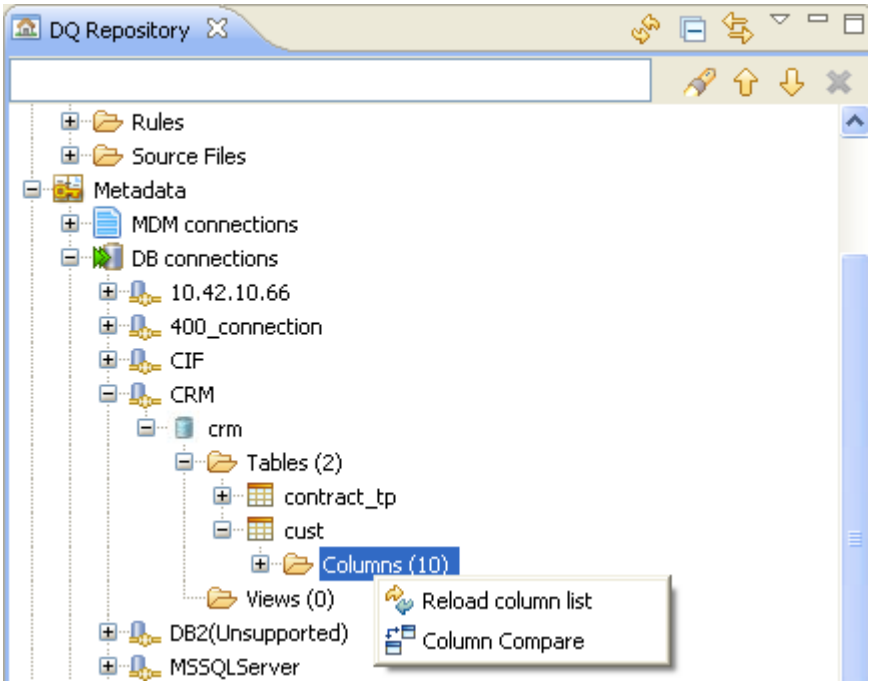
   The selected table list is updated with the new tables in the database, if any.

4.3.2.3. How to synchronize column lists

Prerequisite(s): A database connection has been created in the studio.

To synchronize a column list, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Browse through the entities in your database connection to reach the **Columns** folder you want to synchronize with the database.
3. Right-click the **Columns** folder and select **Reload column list**.



A message will prompt you for confirmation as any change in the database structure may affect existing analyses.

4. Click **OK** to close the confirmation message, or **Cancel** to stop the operation.

   The selected column list is updated with the new column in the database, if any.

Chapter 5. Column analyses

This chapter describes the process of using the studio to examine single columns in databases, data in delimited or excel files or master data on a Master Data Management (MDM) server. It provides detailed information about how to use patterns, indicators and indicator options when analyzing such data.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

5.1. Steps to analyze a column

From the studio, you can examine and collect statistics and information about:

- data available in single columns of database tables,
- data available in delimited or excel files,
- master data available on a Master Data Management (MDM) server. For further information about master data and master data management, see the *Talend Open Studio for MDM Administrator Guide*.

The sequence of profiling data in one or multiple columns involves the following steps:

1. Connecting to the data source being a database, a file or an MDM server. For further information, see chapter *Before you begin profiling data*.
2. Defining one or more columns on which to carry out data profiling processes that will define the content, structure and quality of the data included in the column(s).
3. Settings predefined system indicators or indicators defined by the user on the column(s) that need to be analyzed or monitored. These indicators will represent the results achieved through the implementation of different patterns.
4. Adding to the column analyses the patterns against which you can define the content, structure and quality of the data.

   For further information, see section *How to add a regular expression or an SQL pattern to a column analysis* and

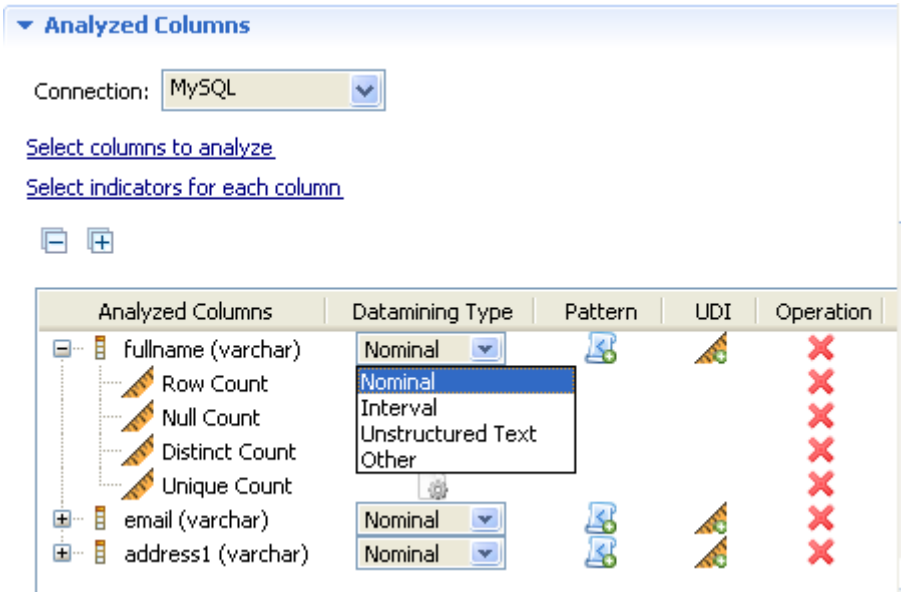
The section *Analyzing columns in a database* explains in detail the procedures to analyze the content of one or multiple columns in a database.

The section *Analyzing master data on an MDM server* explains in detail the procedures to analyze master data on an MDM server.

The section *Analyzing data in a file* explains in detail the procedures to analyze columns in delimited or excel files.

5.2. Data mining types

When you create a column analysis in the studio, you can see a **Datamining Type** box next to each of the columns you want to analyze. The selected type in the box represents the data mining type of the associated column.



These data mining types help the studio to choose the appropriate metrics for the associated column since not all indicators (or metrics) can be computed on all data types.

Available data mining types are: **Nominal**, **Interval**, **Unstructured Text** and **Other**. The sections below describe these data mining types.

5.2.1. Nominal

Nominal data is categorical data which values/observations can be assigned a code in the form of a number where the numbers are simply labels. You can count, but not order or measure nominal data.

In the studio, the mining type of textual data is set to nominal. For example, a column called *WEATHER* with the values: *sun*, *cloud* and *rain* is nominal.

And a column called *POSTAL_CODE* that has the values *52200* and *75014* is nominal as well in spite of the numerical values. Such data is of nominal type because it identifies a postal code in France. Computing mathematical quantities such as the average on such data is non sense. In such a case, you should set the data mining type of the column to Nominal, because there is currently no way in the studio to automatically guess the correct type of data.

The same is true for primary or foreign-key data. Keys are most of the time represented by numerical data, but their data mining type is Nominal.

5.2.2. Interval

This data mining type is used for numerical data and time data. Averages can be computed on this kind of data. In databases, sometimes numerical quantities are stored in textual fields.

In the studio, it is possible to declare the data mining type of a textual column (e.g. a column of type *VARCHAR*) as Interval. In that case, the data should be treated as numerical data and summary statistics should be available.

5.2.3. Unstructured text

This is a new data mining type introduced by the studio. This data mining type is dedicated to handle unstructured textual data.

For example, the data mining type of a column called *COMMENT* that contains commentary text can not be Nominal, since the text in it is unstructured. Still, we could be interested in seeing the duplicate values of such a column and here comes the need for such a new data mining type.

5.2.4. Other

This is another new data mining type introduced in the studio. This type designs the data that the studio does not know how to handle yet.

5.3. Analyzing columns in a database

You can analyze the content of one or multiple columns and execute the created analyses using the Java or the SQL engine. This type of analysis provides statistics about the values within each column.

When you use the Java engine to run a column analysis, you can view the analyzed data according to parameters you set yourself. For more information, see section *Using the Java or the SQL engine*.

When you use the Java engine to run a column analysis on big sets or on data with many problems, it is advisable to define a maximum memory size threshold to execute the analysis as you may end up with a Java heap error. For more information, see section *Defining the maximum memory size threshold*.

You can also analyze a set of columns. This type of analysis provides statistics on the values across all the data set (full records). For more information, see section *Analyzing tables in databases*.

The sequence of analyzing a column involves the following steps:

1. Defining the column(s) to be analyzed.

   For more information, see section *How to define the columns to be analyzed*.

2. Settings predefined system indicators or indicators defined by the user for the column(s).

   For more information, see section *How to set indicators for the column(s) to be analyzed*. For more information on indicator types and indicator management, see section *Indicators*.

3. Adding the patterns against which to define the content, structure and quality of the data.

   For more information, see section *Using regular expressions and SQL patterns in a column analysis*. For more information on pattern types and management, see section *Patterns*.

The following sections provide a detailed description on each of the preceding steps.

5.3.1. Defining the columns to be analyzed and setting indicators

5.3.1.1. How to define the columns to be analyzed

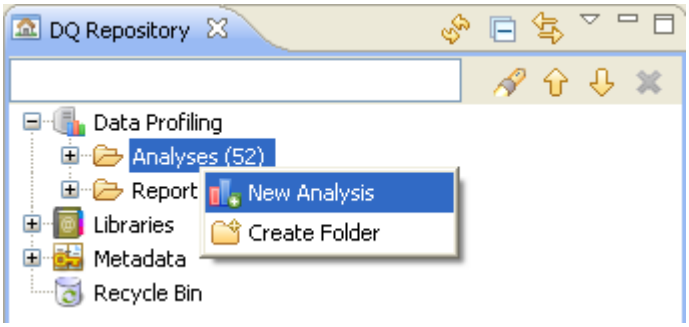
The first step in analyzing the content of one or multiple columns is to define the column(s) to be analyzed. The analysis results provides statistics about the values within each column.

Prerequisite(s): At least one database connection is set in the **Profiling** perspective in the studio. For further information, see section *Connecting to a database*.

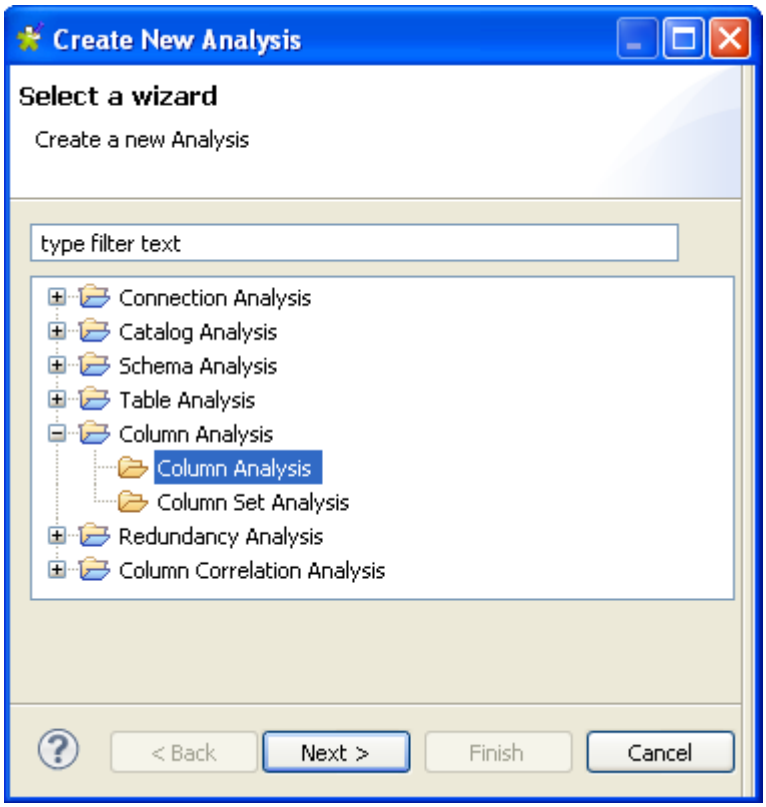
To analyze one or more columns, do the following:

Defining the analysis

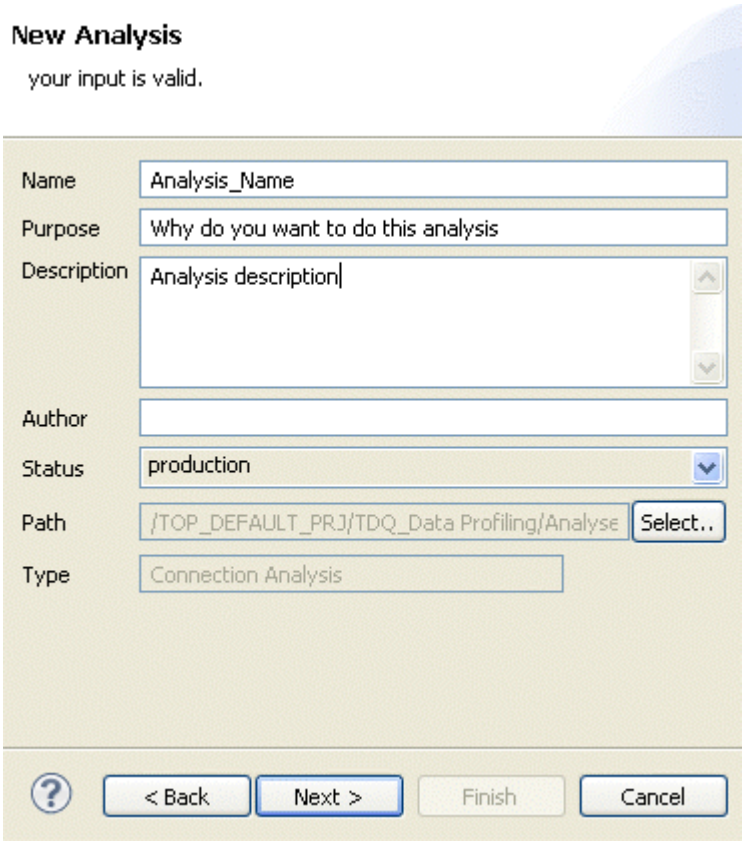
1. In the **DQ Repository** tree view, expand **Data Profiling**.
2. Right-click the **Analysis** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



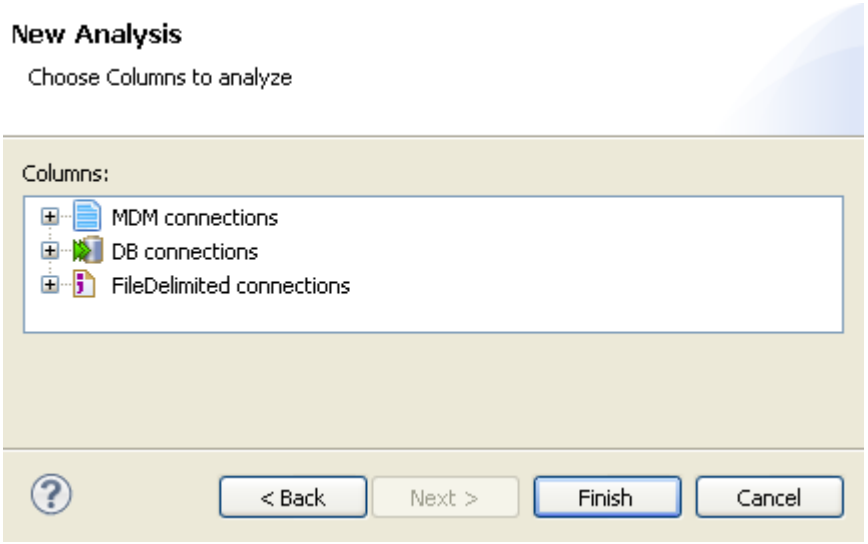
3. Expand the **Column Analysis** node and then click **Column Analysis**.
4. Click the **Next** button.



5. In the **Name** field, enter a name for the current column analysis.

Space is not acceptable when typing in the analysis name in this field.

6. Set column analysis metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.

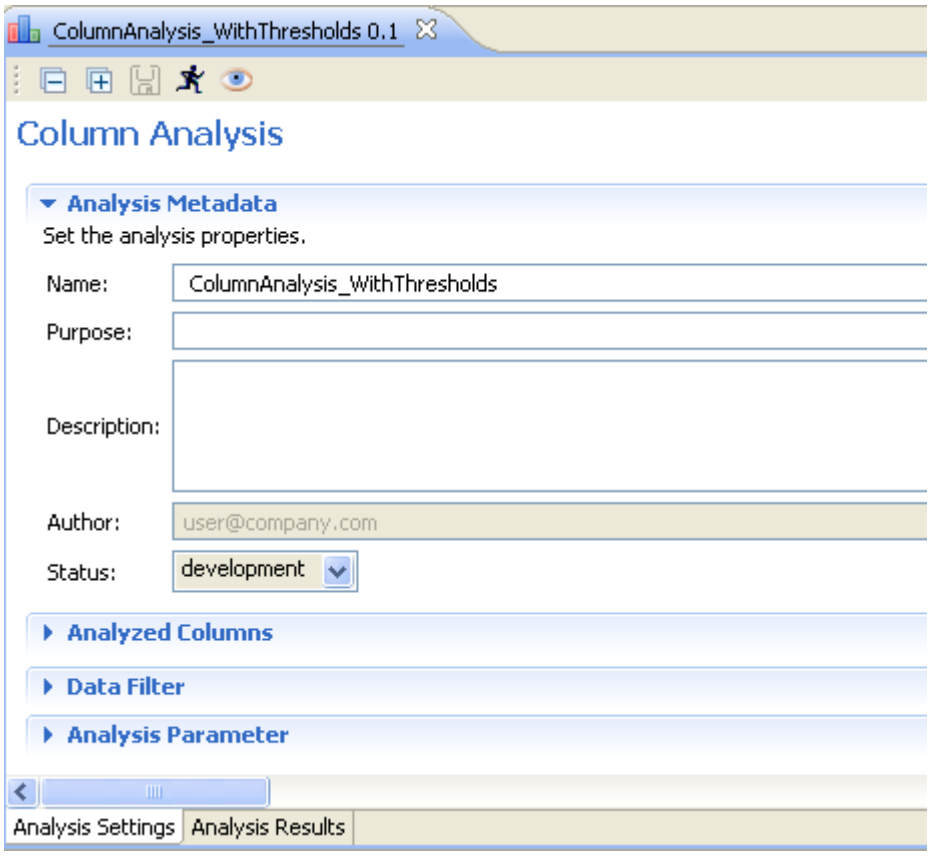


Selecting the column you want to analyze

1. Expand **DB connections** and in the desired database, browse to the columns you want to analyze, select them and then click **Finish** to close the wizard.

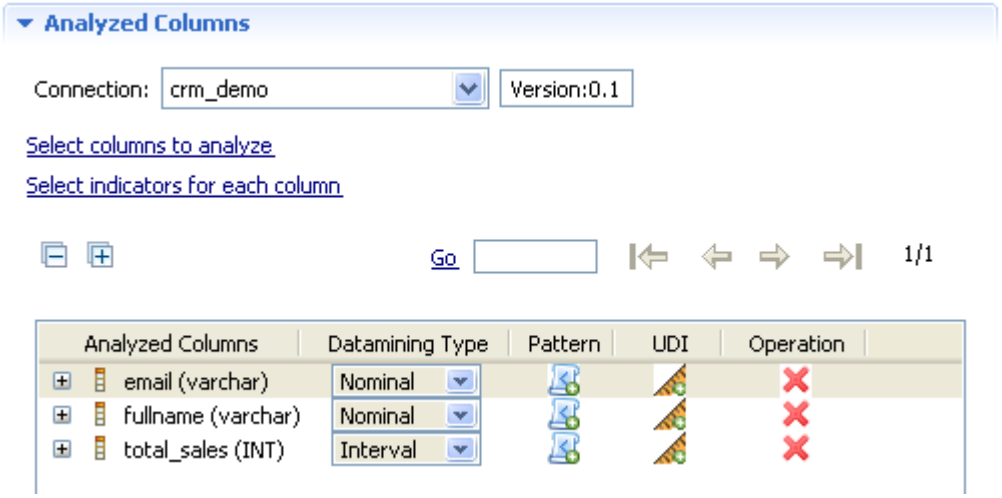
For the DB2 database, if double quotes exist in the column names of a table, the double quotation marks cannot be retrieved when retrieving the column. Therefore, it is recommended not to use double quotes in column names in a DB2 database table.

A file for the newly created column analysis is listed under the **Analysis** node in the **DQ Repository** tree view, and the analysis editor opens with the defined analysis metadata.



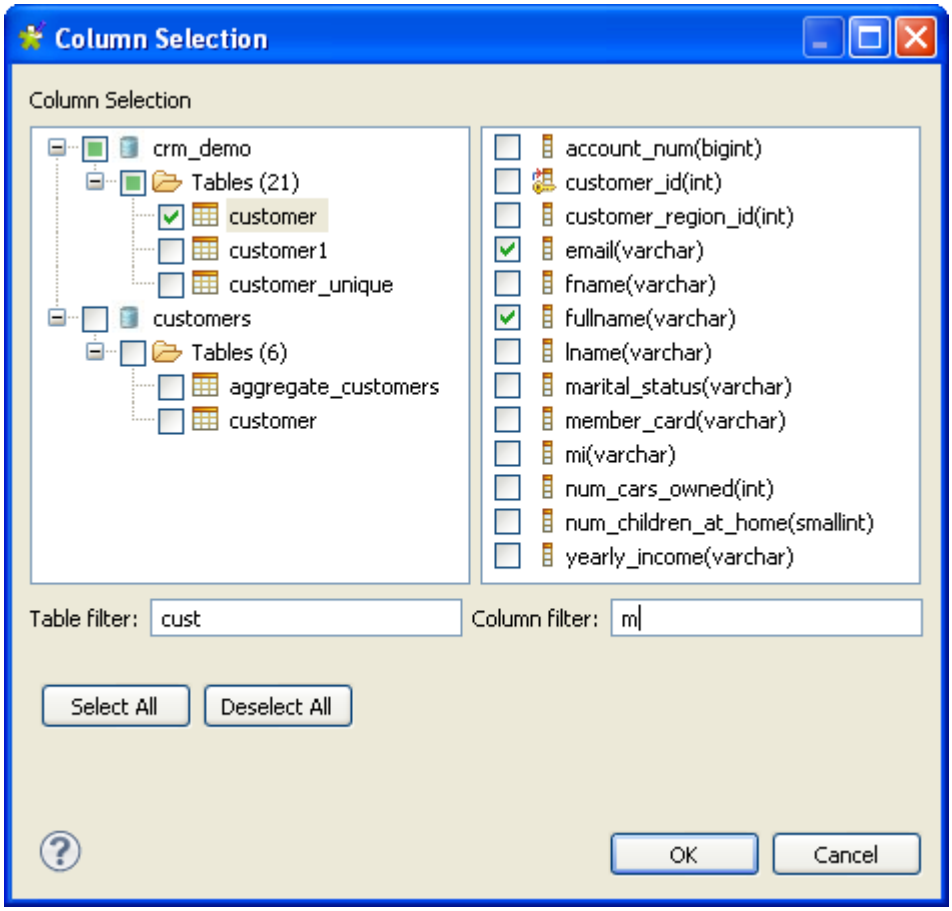
The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

2. Click the **Analyzed Columns** tab to open the corresponding view.



You can drag the columns to be analyzed directly from the **DQ Repository** tree view to the **Analyzed Columns** list in the analysis editor.

3. Click the **Select columns to analyze** link to open a dialog box and select the columns you want to analyze.



You can filter the table or column lists by typing letters or complete words in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

If one of the columns you want to analyze is a primary or a foreign key, its data mining type will automatically become **Nominal** when you list it in the **Analyzed Columns** view. For more information on data mining types, see section *Data mining types*.

4. If required, change your database connection by selecting another connection from the **Connection** box. This box lists all the connections created in the Studio with the corresponding database names.

   If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

   If you select to connect to a database that is not supported in the studio (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on the java engine, see section *Using the Java or the SQL engine*.

5. Click **OK** and then save the column analysis.

   You can right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view** to locate it in the database connection in the **DQ Repository** tree view.

5.3.1.2. How to set indicators for the column(s) to be analyzed

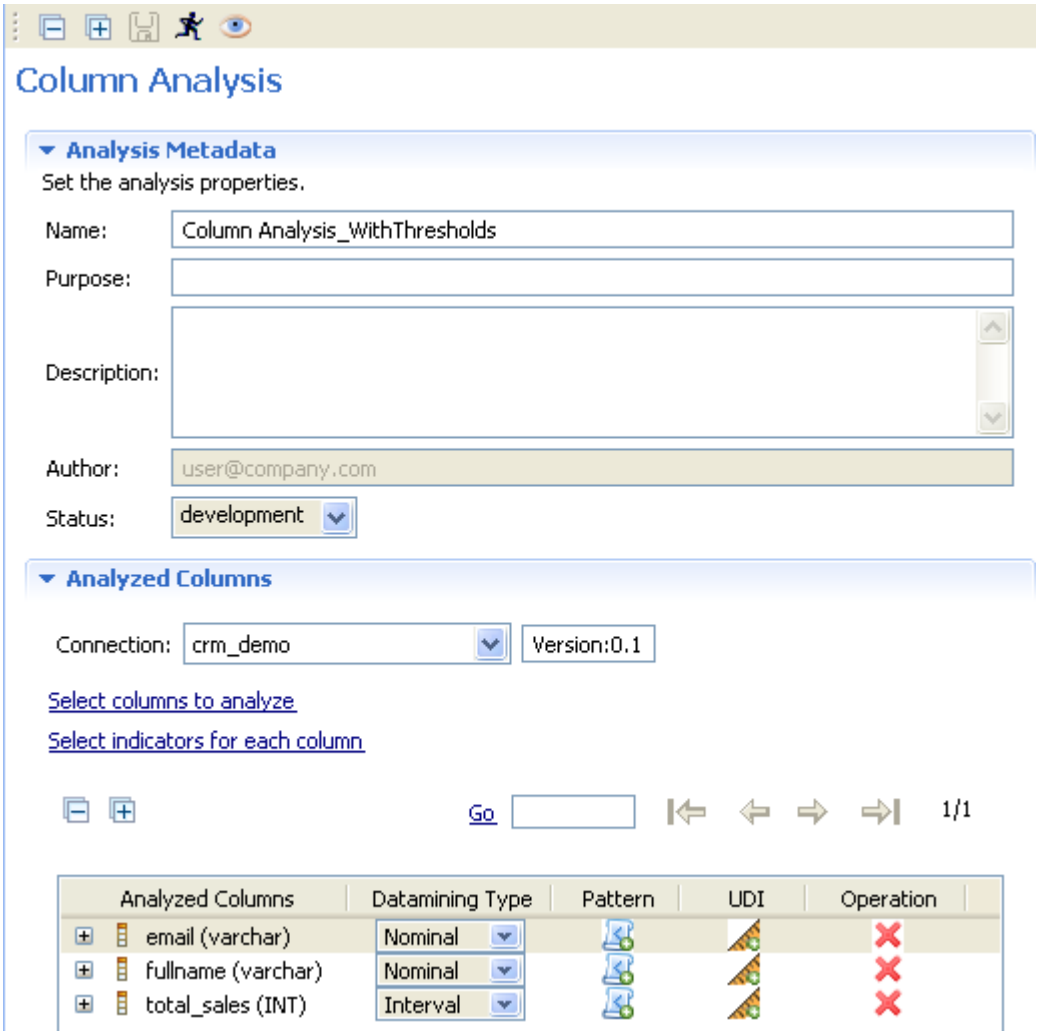
The second step after defining the column(s) to be analyzed is to set either system or user-defined indicators for each of the defined columns.

How to set system indicators

Prerequisite(s): A column analysis is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the columns to be analyzed*.

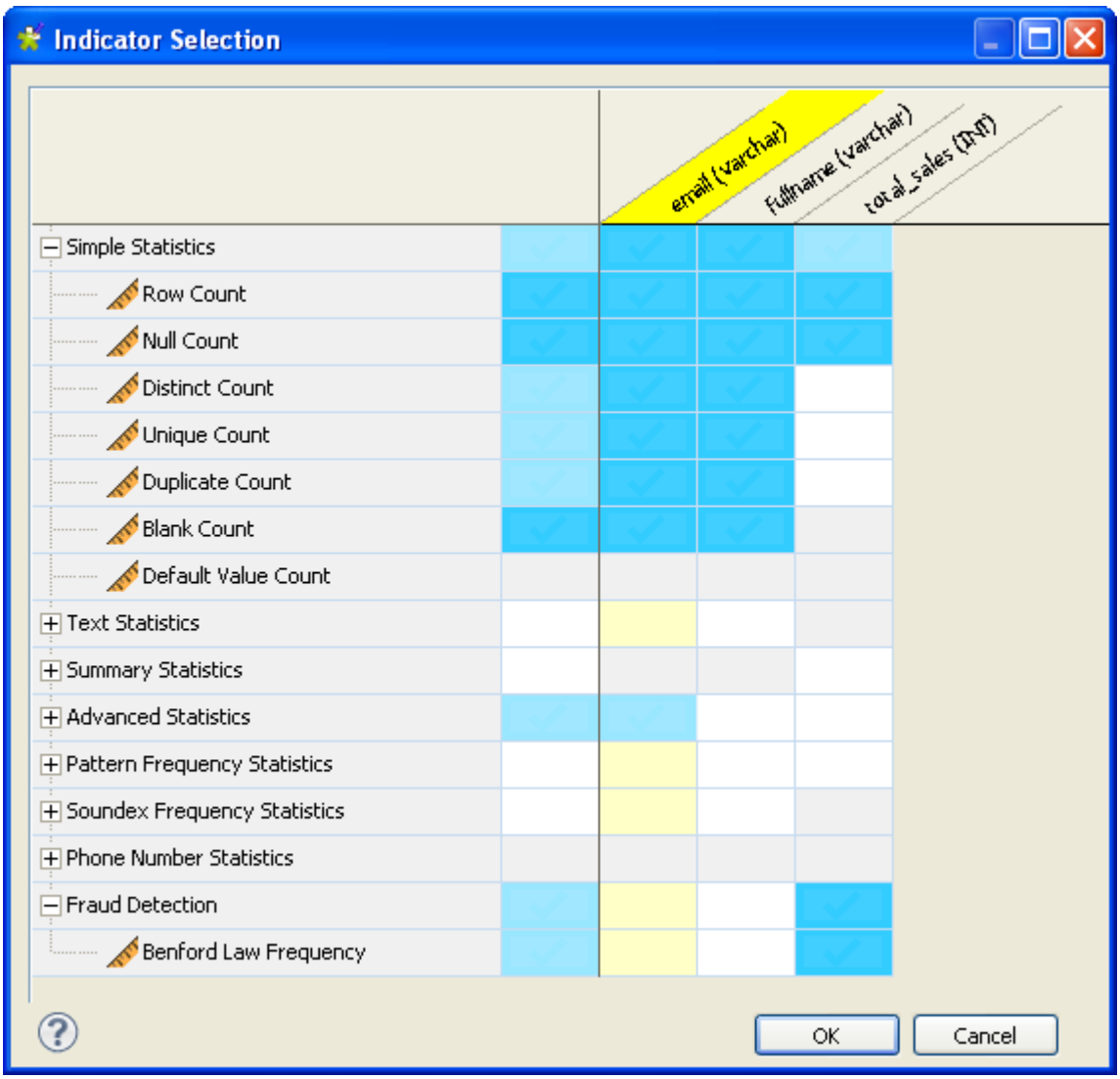
To set system indicators for the column(s) to be analyzed, do the following:

1. In the analysis editor, click **Analyzed Columns** to open the analyzed columns view.



If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

2. Click **Select indicators for each column** to open the **[Indicator Selection]** dialog box.



In this dialog box, you can change column positions by dropping them with the cursor.

3. If you are analyzing very large number of columns, place the cursor in the top/bottom right corner of the **[Indicator Selection]** dialog box to access the columns to the very right.

   Similarly, place the cursor in the top/bottom left corner of the **[Indicator Selection]** dialog box to access the columns to the very left.

4. Click in the cells to set indicator parameters for the analyzed column(s) as needed and then click **OK**.

   Indicators are accordingly attached to the analyzed columns in the **Analyzed Columns** view.

   If you attach the **Data Pattern Frequency Table** to a date column in your analysis, you can generate a date regular expression from the analysis results. For more information, see section *How to generate a regular expression from the Date Pattern Frequency Table*.

5. Click the save icon on the toolbar of the analysis editor.

How to set options for system indicators

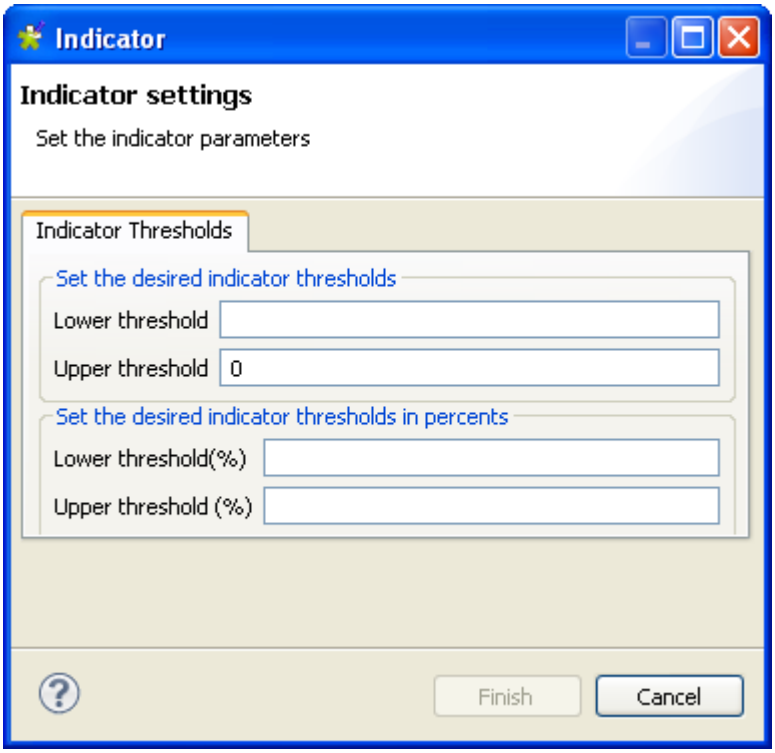
Prerequisite(s): A column analysis is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the columns to be analyzed*.

For more information about setting indicators, see section *How to set system indicators*.

To set options for system indicators, do the following:

1. In the analysis editor, click **Analyzed Columns** to open the analyzed columns view.
2. Click the option icon next to the defined indicator to open the dialog box where you can set options for the given indicator.

   For example, if you want to flag if there are null values in the column you analyze, you can set *0* in the **Upper threshold** field for the *Null Count* indicator.



Indicators settings dialog boxes differ according to the parameters specific for each indicator. For more information about different indicator parameters, see section *Indicator parameters*.

3. Set the parameters for the given indicator.
4. Click **Finish** to close the dialog box.
5. Click the save icon on the toolbar of the analysis editor.

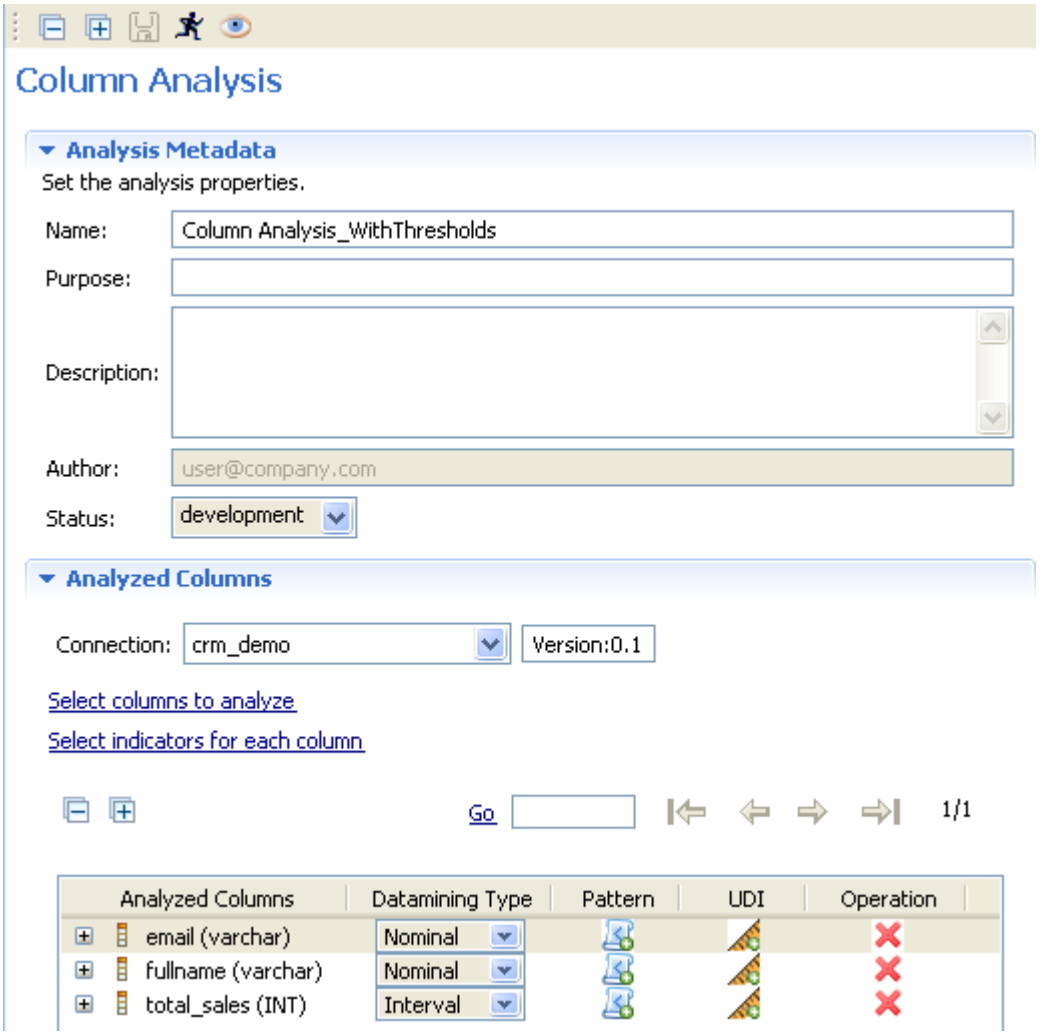
How to set user-defined indicators

Prerequisite(s):

- A column analysis is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the columns to be analyzed*.
- A user-defined indicator is created in the **Profiling** perspective of the studio. For more information, see section *How to create SQL user-defined indicators*.

To set user-defined indicators for the column(s) to be analyzed, do the following:

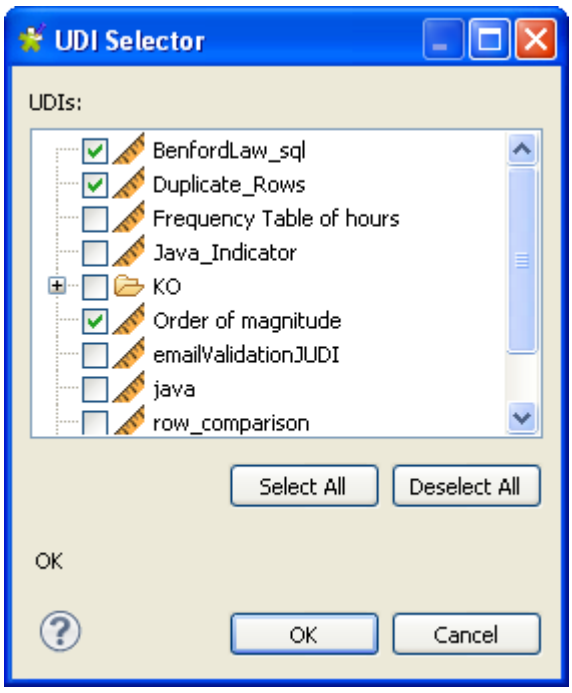
1. In the analysis editor, click **Analyzed Columns** to open the analyzed columns view.



2. Either:

    1. In the **Analyzed Columns** view, click the  icon next to the column name to which you want to define a user-defined indicator.

        The **[UDI selector]** dialog box opens.



2. Select the user-defined indicators you want to use on the column and then click **OK** to close the dialog box.

Or:

1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators**.
2. From the **User Defined Indicator** folder, drop the user-defined indicator(s) against which you want to analyze the column content to the column name(s) in the **Analyzed Columns** view.

   The user-defined indicator is listed under the column name

3. Click the save icon on the toolbar of the analysis editor.

5.3.2. Finalizing the column analysis before execution

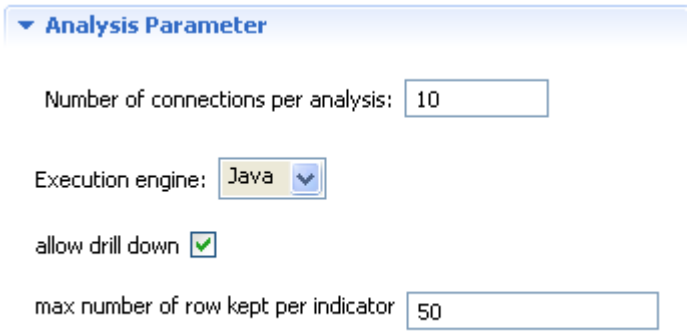
After defining the column(s) to be analyzed and setting indicators, you may want to filter the data that you want to analyze and decide what engine to use to execute the column analysis.

Prerequisite(s):

- The column analysis is open in the analysis editor in the **Profiling** perspective of the studio . For more information, see section *How to define the columns to be analyzed*.
- You have set system or predefined indicators for the column analysis. For more information, see section *How to set indicators for the column(s) to be analyzed*.

To finalize the column analysis defined in the above sections, do the following:

1. In the analysis editor, click **Data Filter** to open the corresponding view and filter data through SQL “WHERE” clauses, if required.
2. Click **Analysis Parameters** and:



- In the **Number of connections per analysis** field, set the number of concurrent connections allowed per analysis to the selected database connection.

  You can set this number according to the database available resources, that is the number of concurrent connections each database can support.

- From the **Execution engine** list, select the engine, Java or SQL, you want to use to execute the analysis.

If you select the Java engine and then select the **Allow drill down** check box in the **Analysis parameters** view, you can store locally the analyzed data and thus access it in the **Analysis Results > Data view**. You can use the **Max number of rows kept per indicator** field to decide the number of the data rows you want to make accessible. For more information on viewing analyzed data, see section *Using the Java or the SQL engine*.

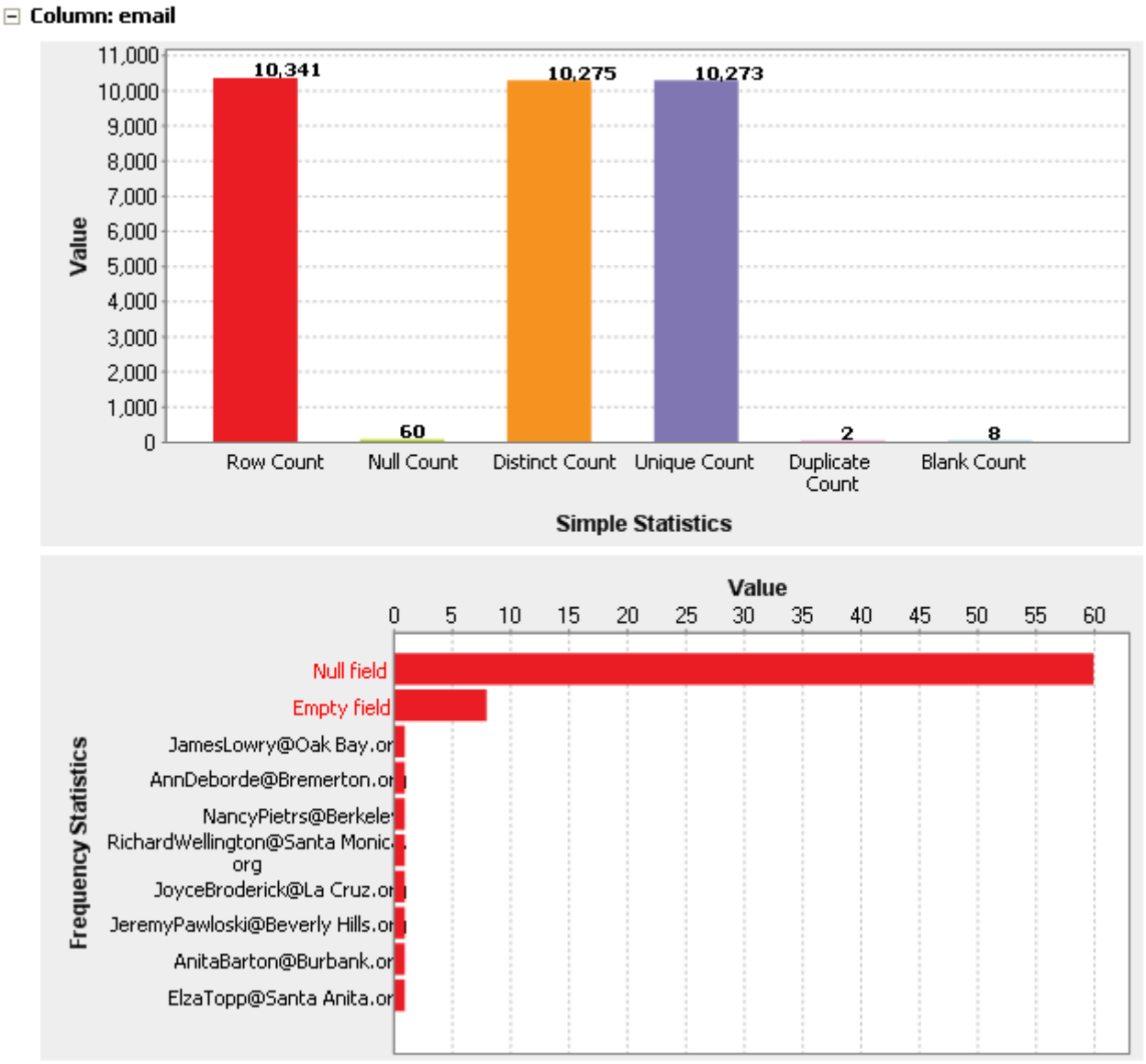
When you select the Java engine, the system will look for Java regular expressions first, if none is found, it looks for SQL regular expressions. For more information on the Java and the SQL engines, see section *Using the Java or the SQL engine*

If you select to connect to a database that is not supported in the studio (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on the java engine, see section *Using the Java or the SQL engine*.

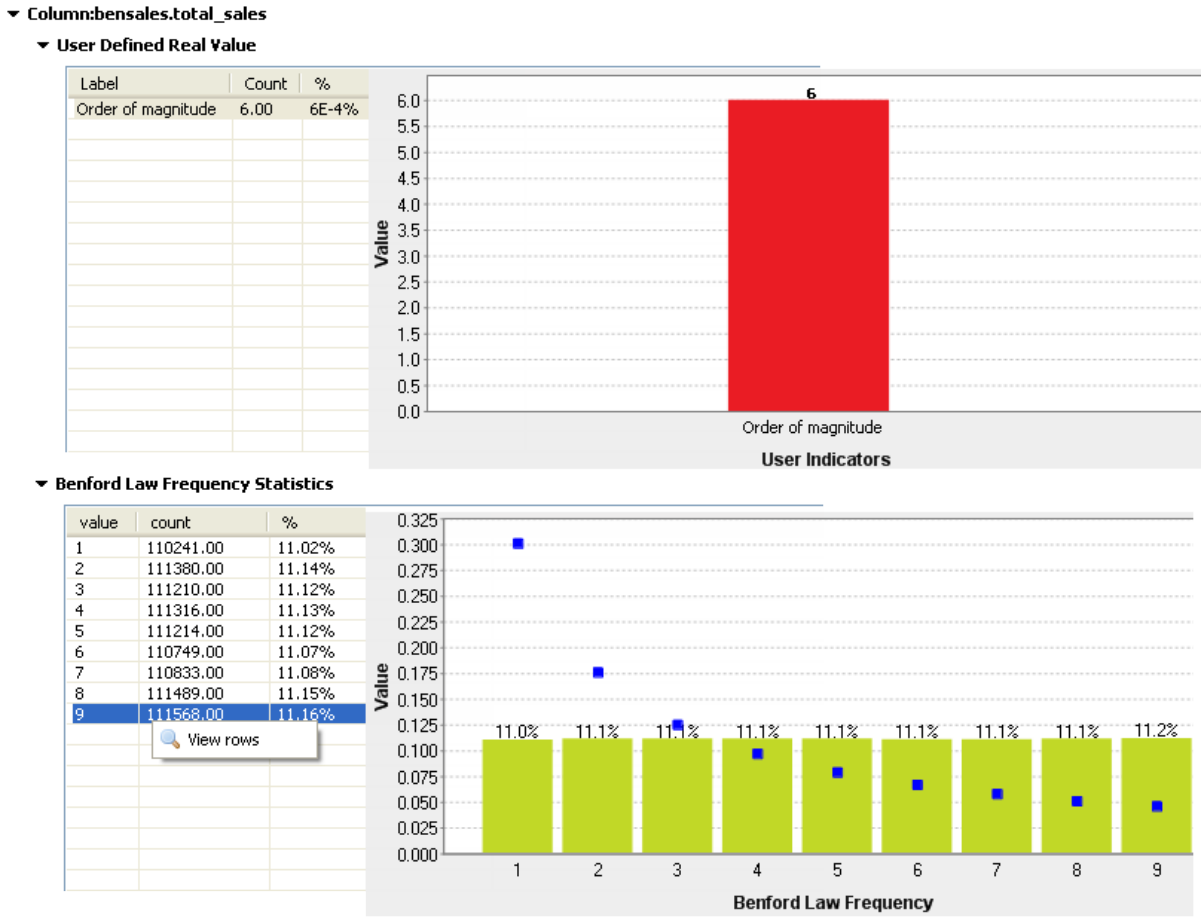
3. Click the save icon on the toolbar of the analysis editor and then press **F6** to execute the column analysis.

A group of graphics is displayed in the **Graphics** panel to the right of the analysis editor, each corresponding to the group of the indicators set for each analyzed column.

Below are the graphics representing the Frequency Statistics and Simple Statistics for the *email* column analyzed in the above procedure.



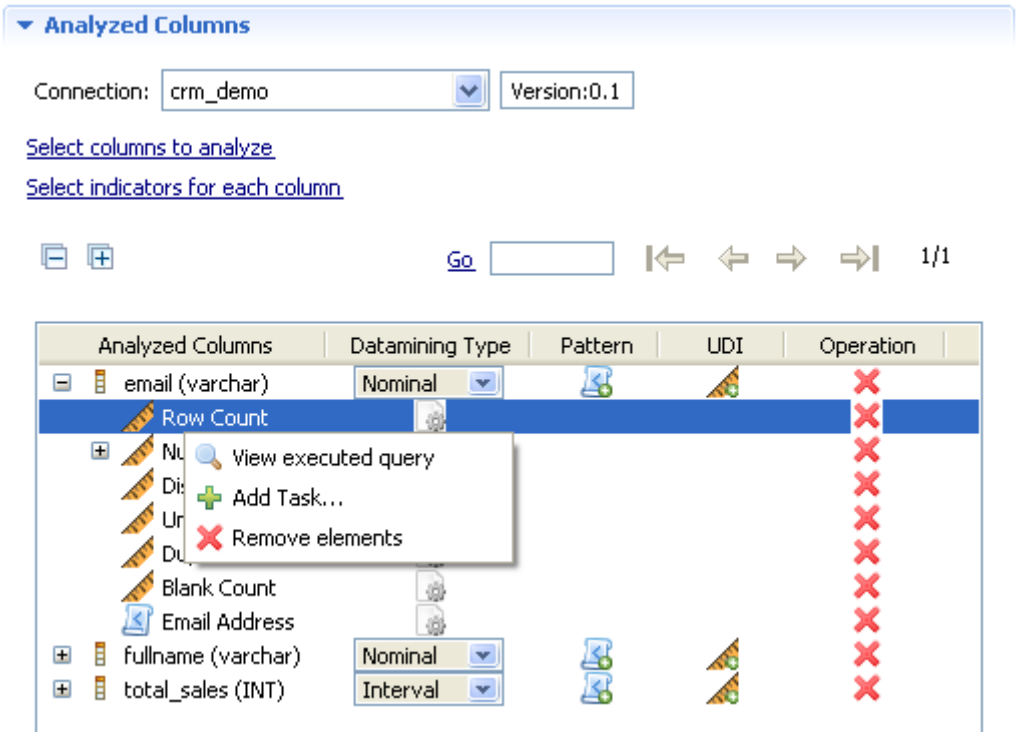
Below are the graphics representing the order of magnitude and the Benford's law frequency statistics for the *total_sales* column analyzed in the above procedure.



For further information about the Benford's law frequency statistics usually used as an indicator of accounting and expenses fraud in lists or tables, see section *Benford's law frequency indicator*.

For information on how to access a detailed view of the results of the analysis, see section *How to access the detailed view of the analysis results*.

If you execute this analysis using the SQL engine, you can view the executed query for each of the attached indicators if you right-click an indicator and then select the **View executed query** option from the list. However, when you use the Java engine, SQL queries will not be accessible and thus clicking this option will open a warning message.



For more information on the Java and the SQL engines, see section *Using the Java or the SQL engine*.

5.3.3. Using the Java or the SQL engine

After setting the analysis parameters in the analysis editor, you can use either the Java or the SQL engine to execute your analysis.

If you use the SQL engine to execute a column analysis:

- an SQL query is generated for each indicator used in the column analysis,
- data monitoring and processing is carried on the DBMS,
- only statistical results are retrieved locally.

Using this engine, you guarantee system better performance. You can also access valid/invalid data in the data explorer, for more information, see section *Viewing and exporting analyzed data*.

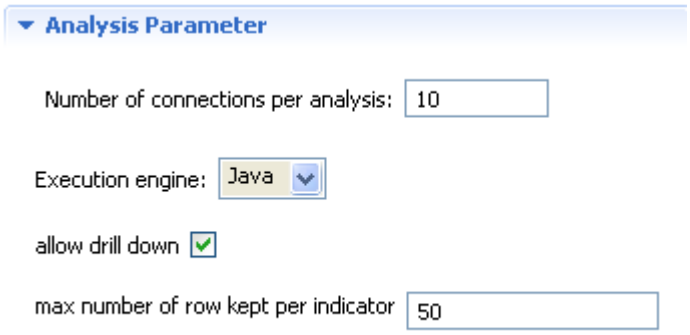
If you use the Java engine to execute a column analysis:

- only one query is generated for all indicators used in the column analysis,
- all monitored data is retrieved locally to be analyzed,
- you can set the parameters to decide whether to access the analyzed data and how many data rows to show per indicator. This will help to avoid memory limitation issues since it is impossible to store all analyzed data.

When you use the Java engine to execute a column analysis you do not need different query templates specific for each database. However, system performance is significantly reduced in comparison with the SQL engine.

To set the parameters to access analyzed data when using the Java engine, do the following:

1. In the **Analysis Parameter** view of the column analysis editor, select **Java** from the **Execution engine** list.



2. Select the **Allow drill down** check box to store locally the data that will be analyzed by the current analysis.

   This check box is usually selected by default.

3. In the **Max number of rows kept per indicator** field enter the number of the data rows you want to make accessible.

   This field is set to 50 by default.

You can now run your analysis and then have access to the analyzed data according to the set parameters. For more information, see section *Viewing and exporting analyzed data*.

5.3.4. Accessing the detailed view of the database column analysis

Prerequisite(s): A column analysis is defined and executed.

To access a more detailed view of the analysis results of the procedures outlined in section *Defining the columns to be analyzed and setting indicators* and section *Finalizing the column analysis before execution*, do the following:

1. Click the **Analysis Results** tab at the bottom of the analysis editor to open the corresponding view.

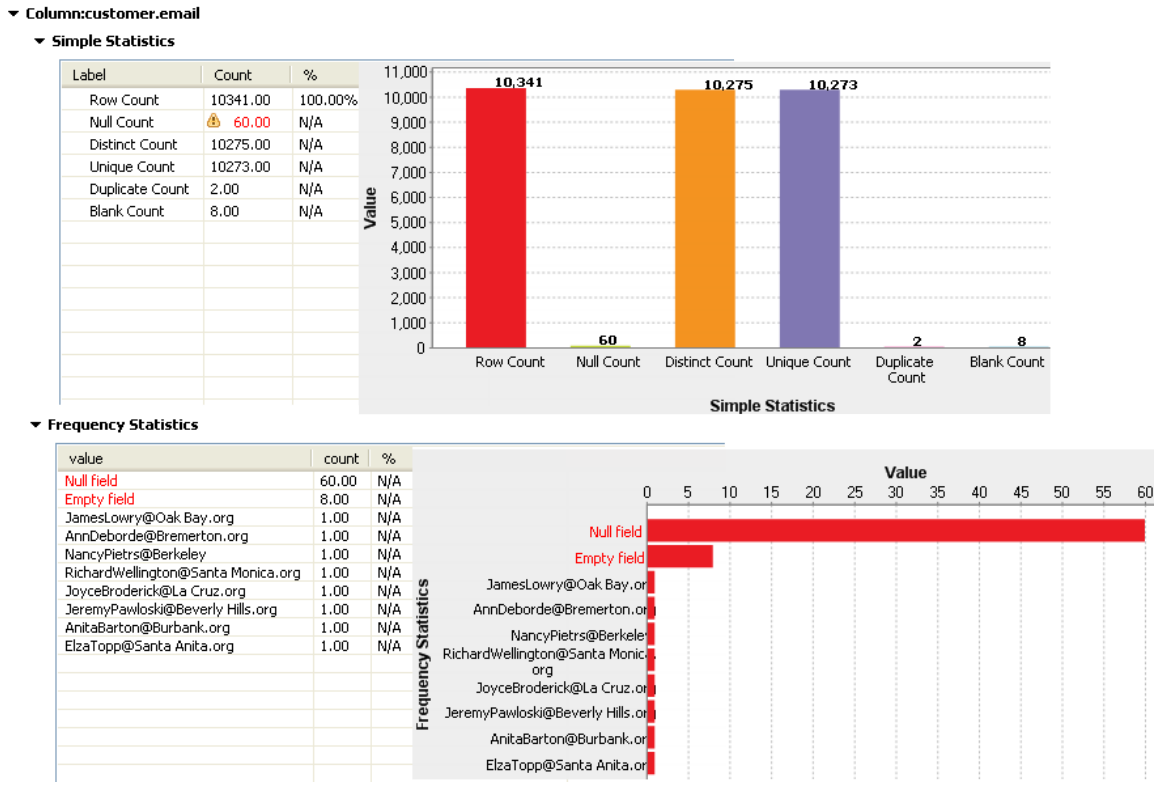
2. Click the **Analysis Result** tab in the view and then the name of the analyzed column for which you want to open the detailed results.

   The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

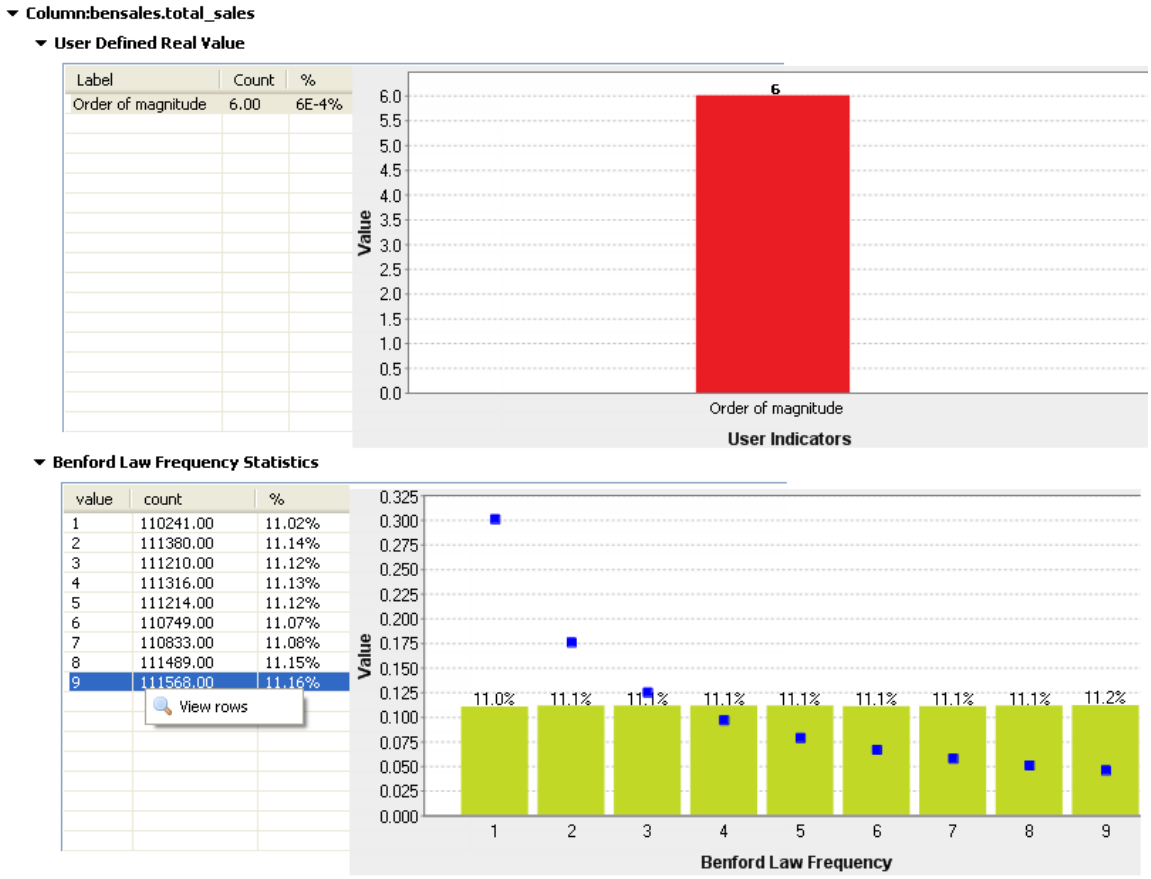
   The detailed analysis results view shows the generated graphics for the analyzed columns accompanied with tables that detail the statistic results.

   Below are the tables that accompany the Frequency and Simple Statistics graphics in the **Analysis Results** view for the analyzed *email* column.



In the **Simple Statistics** table, if an indicator value is displayed in red, this means that a threshold has been set on the indicator in the column analysis editor and that this threshold has been violated. For further information about data thresholds, see section *How to set options for system indicators*.

Below are the tables and the graphics representing the order of magnitude and the Benford's law frequency statistics in the **Analysis Results** view for the analyzed *total_sales* column.



For further information about the Benford's law frequency statistics usually used as an indicator of accounting and expenses fraud in lists or tables, see section *Benford's law frequency indicator*.

3. Right-click any data row in the result tables and select **View rows** to access a view of the analyzed data.

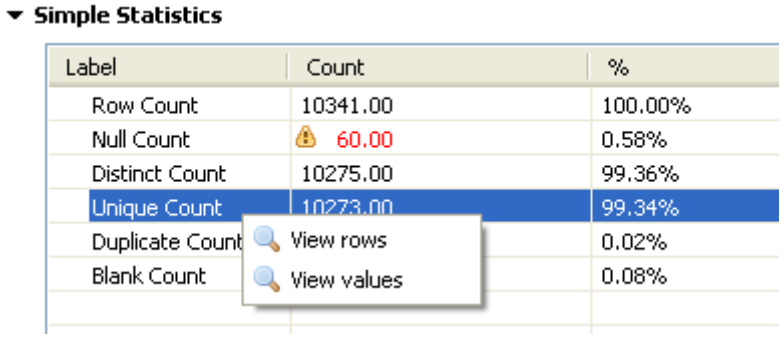
   For more information, see section *Viewing and exporting analyzed data*.

5.3.5. Viewing and exporting analyzed data

After running your analysis using the SQL or the Java engine and from the **Analysis Results** view of the analysis editor, you can right-click any of the rows in the statistic result tables and access a view of the actual data.

After running your analysis using the Java engine, you can use the analysis results to access a view of the actual data.

After running your analysis using the SQL engine, you can use the analysis results to open the **Data Explorer** perspective and access a view of the actual data.



Prerequisite(s): A column analysis has been created and executed.

To view and export the analyzed data, do the following:

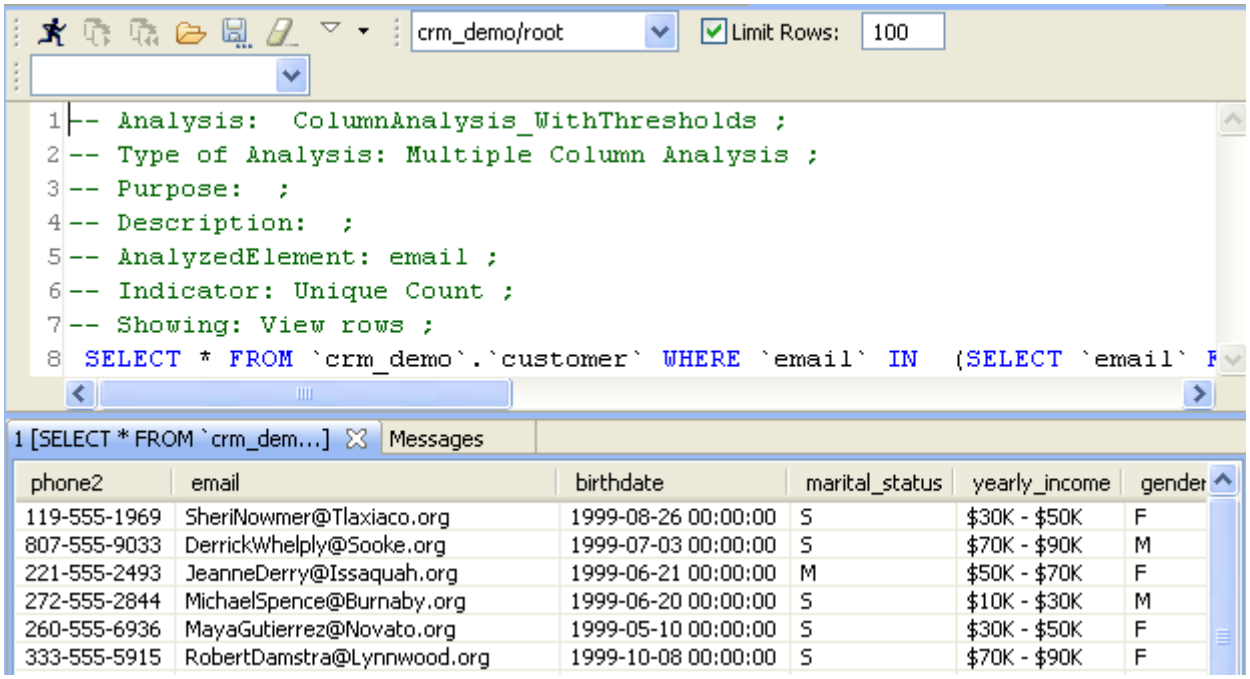
1. At the bottom of the analysis editor, click the **Analysis Results** tab to open a detailed view of the analysis results.

2. Right-click a data row in the statistic results of the analyzed columns and select an option as the following:

Option	Operation
View rows	open a view on a list of all data rows in the analyzed column. For the *Duplicate Count* indicator, the **View rows** option will list all the rows that are duplicated. So if the duplicate count is 12 for example, this option will list 24 rows.
View values	open a view on a list of the actual data values of the analyzed column.

Options other than the above listed ones are available when using regular expressions and SQL patterns in a column analysis. For further information, see section *Using regular expressions and SQL patterns in a column analysis* and section *How to view the data analyzed against patterns*.

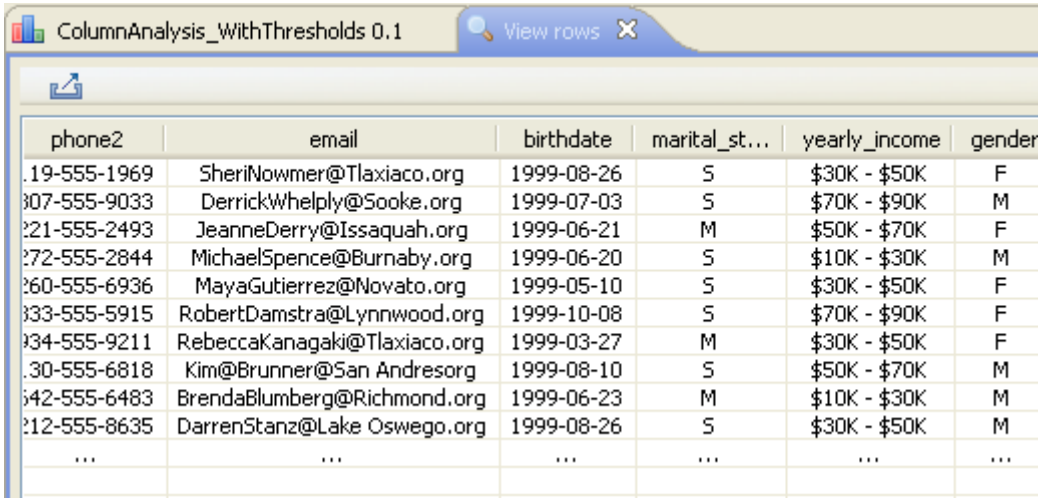
When using the SQL engine, the view opens in the **Data Explorer** perspective listing the rows or the values of the analyzed data according to the limits set in the data explorer.



This explorer view will give also some basic information about the analysis itself. Such information is of great help when working with multiple analysis at the same time.

*The data explorer does not support connections which has empty user name, such as Single sign-on of MS SQL Server. If you analyze data using such connection and you try to view data rows and values in the **Data Explorer** perspective, a warning message prompt you to set your connection credentials to the SQL Server.*

When using the Java engine, the view opens in the studio listing the number of the analyzed data rows you set in the **Analysis parameters** view of the analysis editor. For more information, see section *Using the Java or the SQL engine*.

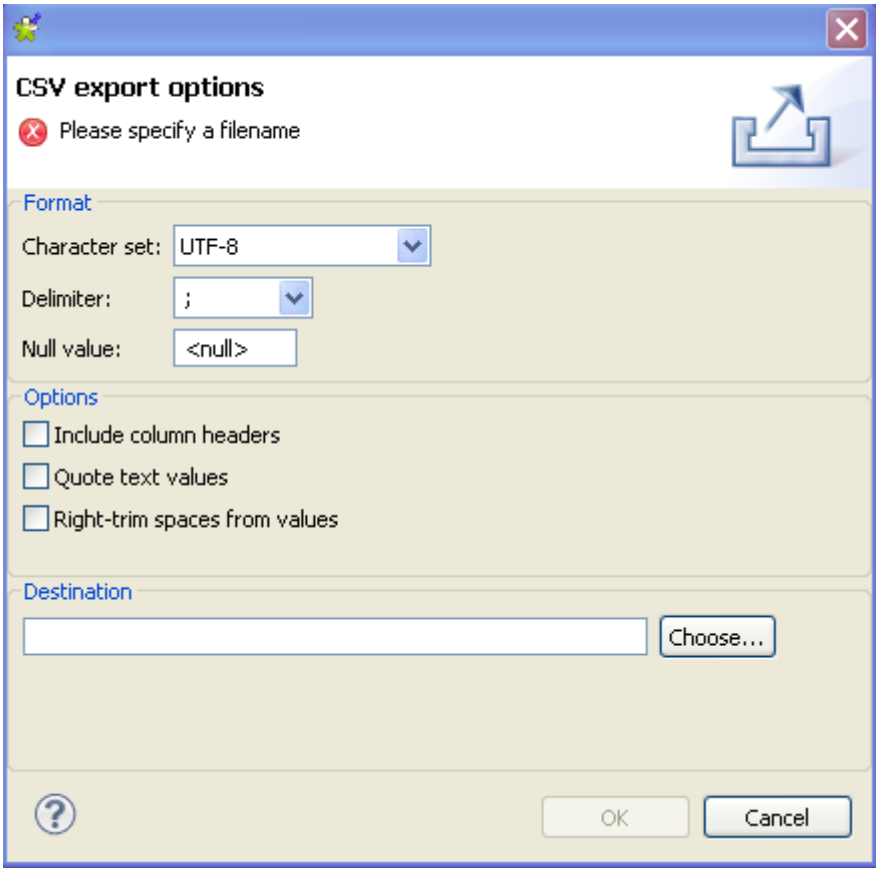


phone2	email	birthdate	marital_st...	yearly_income	gender
.19-555-1969	SheriNowmer@Tlaxiaco.org	1999-08-26	S	$30K - $50K	F
307-555-9033	DerrickWhelply@Sooke.org	1999-07-03	S	$70K - $90K	M
221-555-2493	JeanneDerry@Issaquah.org	1999-06-21	M	$50K - $70K	F
272-555-2844	MichaelSpence@Burnaby.org	1999-06-20	S	$10K - $30K	M
260-555-6936	MayaGutierrez@Novato.org	1999-05-10	S	$30K - $50K	F
333-555-5915	RobertDamstra@Lynnwood.org	1999-10-08	S	$70K - $90K	F
934-555-9211	RebeccaKanagaki@Tlaxiaco.org	1999-03-27	M	$30K - $50K	F
.30-555-6818	Kim@Brunner@San Andresorg	1999-08-10	S	$50K - $70K	M
542-555-6483	BrendaBlumberg@Richmond.org	1999-06-23	M	$10K - $30K	M
212-555-8635	DarrenStanz@Lake Oswego.org	1999-08-26	S	$30K - $50K	M
...	...	...	...	...	...

From this view, you can export the analyzed data into a csv file. To do that:

1. Click the  icon in the upper left corner of the view.

   A dialog box opens.



2. Click the **Choose...** button and browse to where you want to store the csv file and give it a name.
3. Click **OK** to close the dialog box.

   A csv file is created in the specified place holding all the analyzed data rows listed in the view.

5.3.6. Using regular expressions and SQL patterns in a column analysis

You can use regular expressions or SQL patterns in column analyses. These expressions and patterns will help you define the content, structure and quality of the data included in the analyzed columns.

For more information on regular expressions and SQL patterns, see section *Patterns and indicators* and chapter *Table analyses*.

5.3.6.1. How to add a regular expression or an SQL pattern to a column analysis

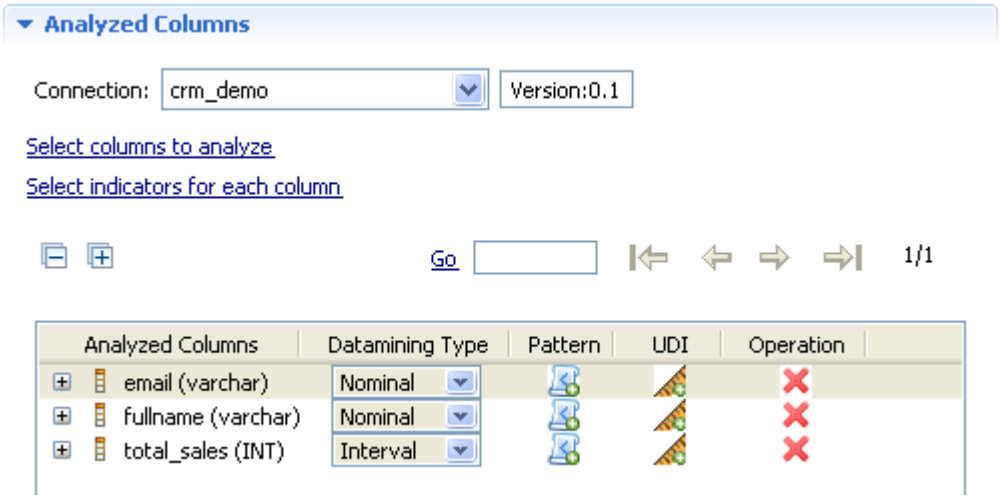
You can add to any column analysis one or more regular expressions or SQL patterns against which you can match the content of the column to be analyzed.

If the database you are using does not support regular expressions or if the query template is not defined in the studio, you need first to declare the user defined function and define the query template before being able to add any of the specified patterns to the column analysis. For more information, see section Managing User-Defined Functions in databases.

Prerequisite(s): A column analysis is open in the analysis editor.

To add a regular expression or an SQL pattern to a column analysis, do the following:

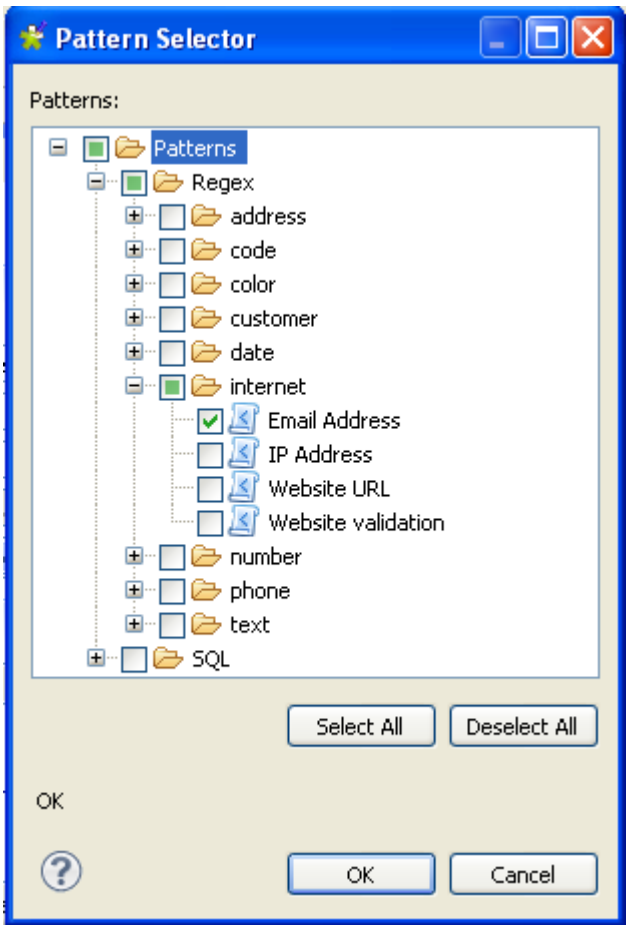
1. Follow the steps outlined in section *How to define the columns to be analyzed* to create a column analysis.
2. In the open analysis editor, click **Analyze Columns** to open the analyzed columns view.



If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

3. Click the  icon next to the column name to which you want to add a regular expression or an SQL pattern, the *email* column in this example.

   The **[Pattern Selector]** dialog box opens.



4. Expand **Patterns** and browse to the regular expression or/and the SQL patterns you want to add to the column analysis.

5. Select the check box(es) of the expression(s) or pattern(s) you want to add to the selected column.

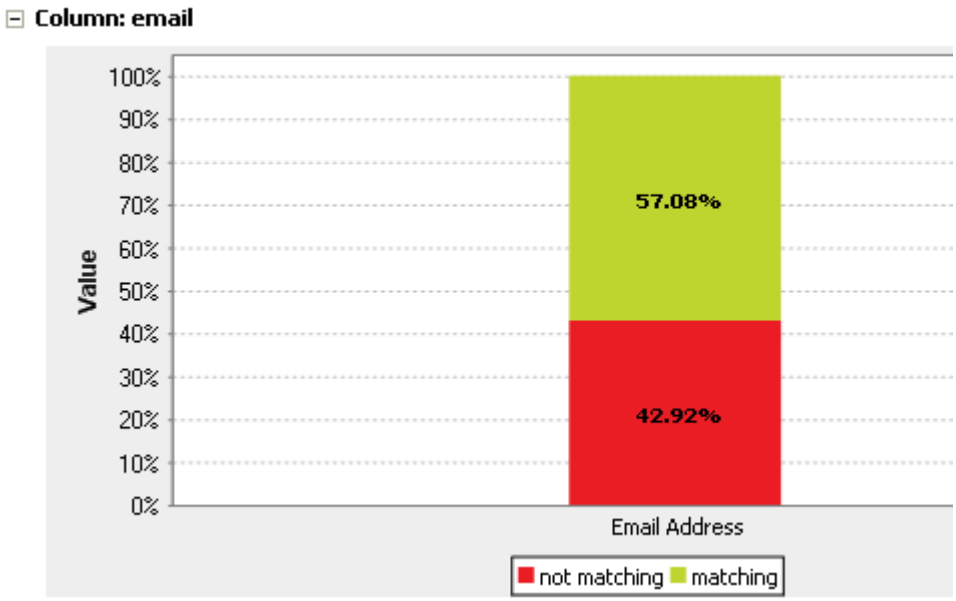
6. Click **OK** to proceed to the next step.

   The added regular expression(s) or SQL pattern(s) are displayed under the analyzed column in the **Analyzed Column** list.

   You can add a regular expression or an SQL pattern to a column simply by a drag and drop operation from the **DQ Repository** tree view onto the analyzed column.

7. Click the save icon on the toolbar of the analysis editor and then press **F6** to execute the column analysis.

   A group of graphics is displayed in the **Graphics** panel to the right of the analysis editor. These graphics show the results of the column analysis including those for pattern matching.

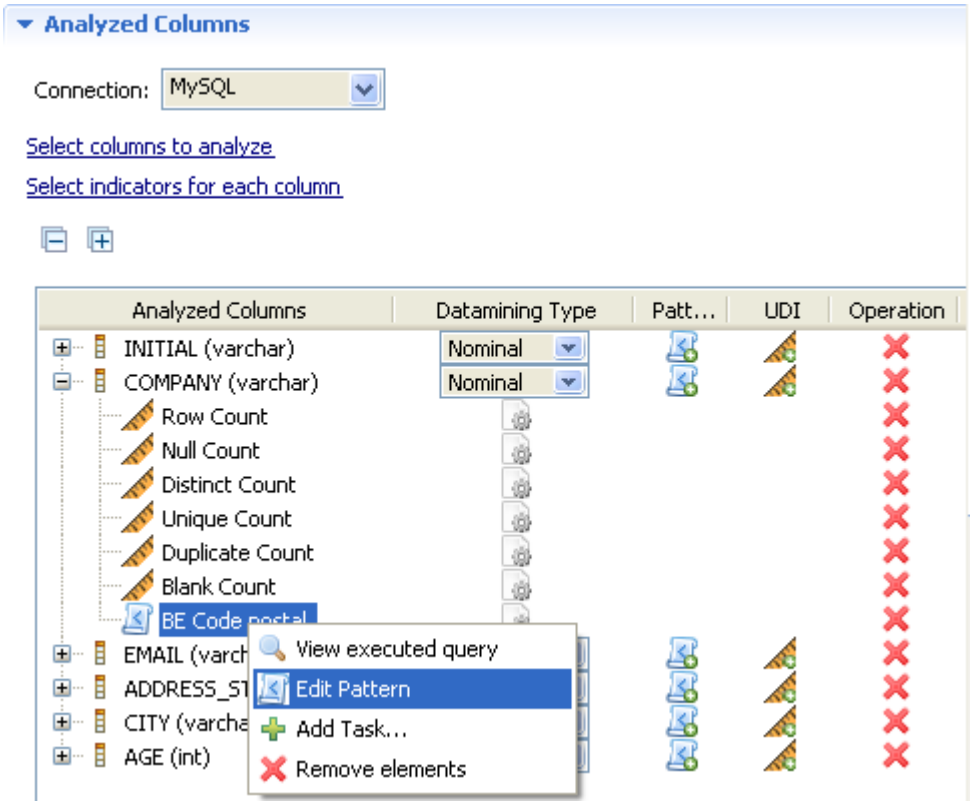


5.3.6.2. How to edit a pattern in the column analysis

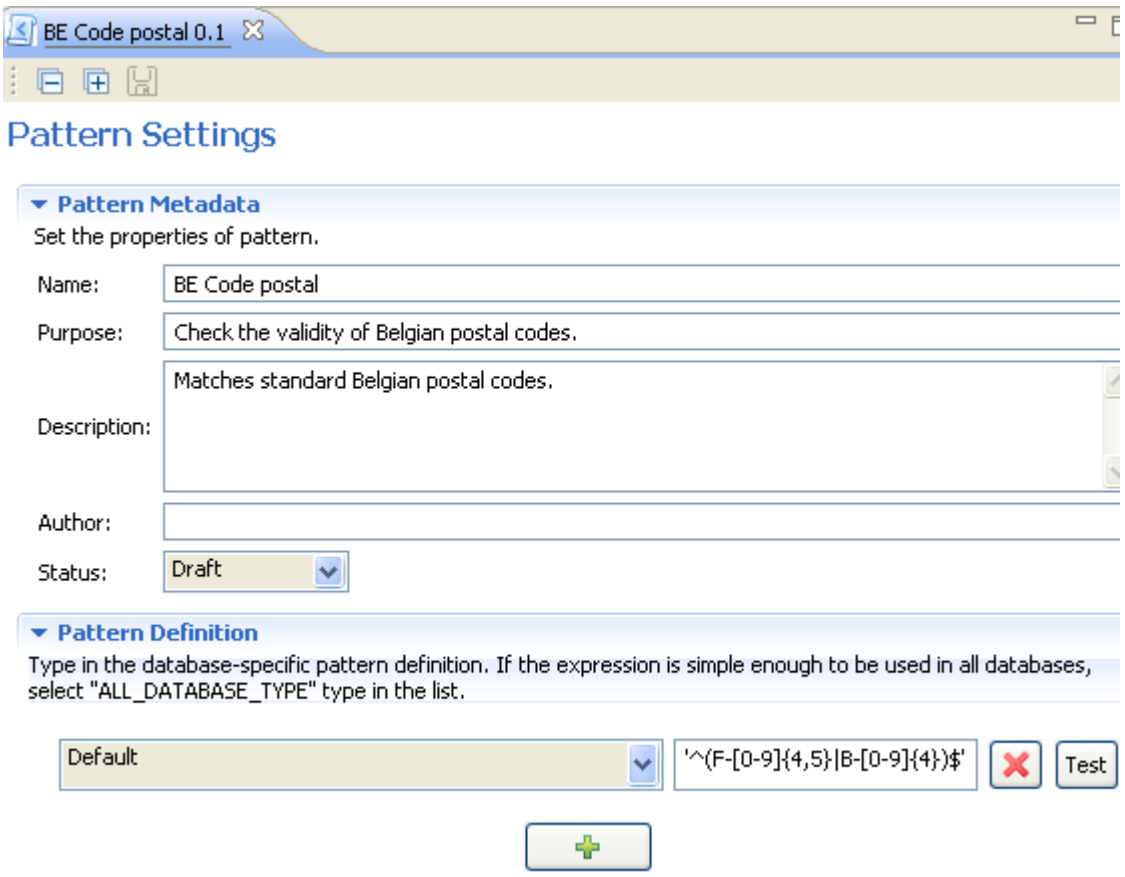
Prerequisite(s): A column analysis is open in the analysis editor.

To edit a pattern added to an analyzed column:

1. Click **Analyze Columns** to open the analyzed columns view.
2. Right-click the pattern you want to edit and select **Edit pattern** from the contextual menu.



The pattern editor opens showing the selected pattern metadata.



3. In the pattern editor, click **Pattern Definition** to edit the pattern definition, or change the selected database, or add other patterns specific to available databases using the **[+]** button.

4. On the toolbar, click the save icon to save your changes.

If the regular pattern is simple enough to be used in all databases, select `Default` in the list.

When you edit a pattern through the analysis editor, you modify the pattern listed in the **DQ Repository** tree view. Make sure that your modifications are suitable for all other analyses that may be using the pattern modified.

5.3.6.3. How to view the data analyzed against patterns

When you add one or more patterns to an analyzed column, you check all existing data in the column against the specified pattern(s). After the execution of the column analysis, using the java or the SQL engine you can access a list of all the valid/invalid data in the analyzed column.

When you use the Java engine to run the analysis, the view of the actual data will open in the studio. While if you use the SQL engine to execute the analysis, the view of the actual data will open in the **Data Explorer** perspective.

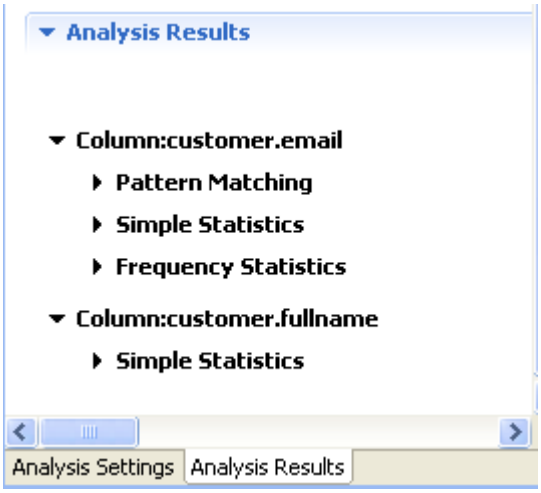
Prerequisite(s): A column analysis that uses patterns has been created and executed.

To view the actual data in the column analyzed against a specific pattern, do the following:

1. Follow the steps outlined in section *How to define the columns to be analyzed* and section *How to add a regular expression or an SQL pattern to a column analysis* to create a column analysis that uses a pattern.

2. Execute the column analysis.

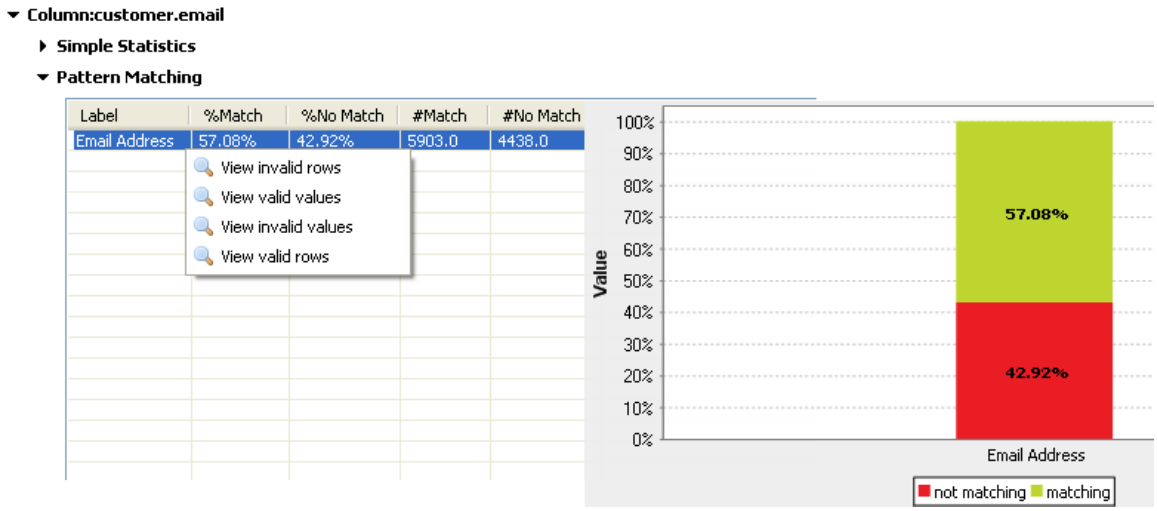
3. In the analysis editor, click the **Analysis Results** tab at the bottom of the editor to open the corresponding view.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

4. Click **Pattern Matching** under the name of the analyzed column.

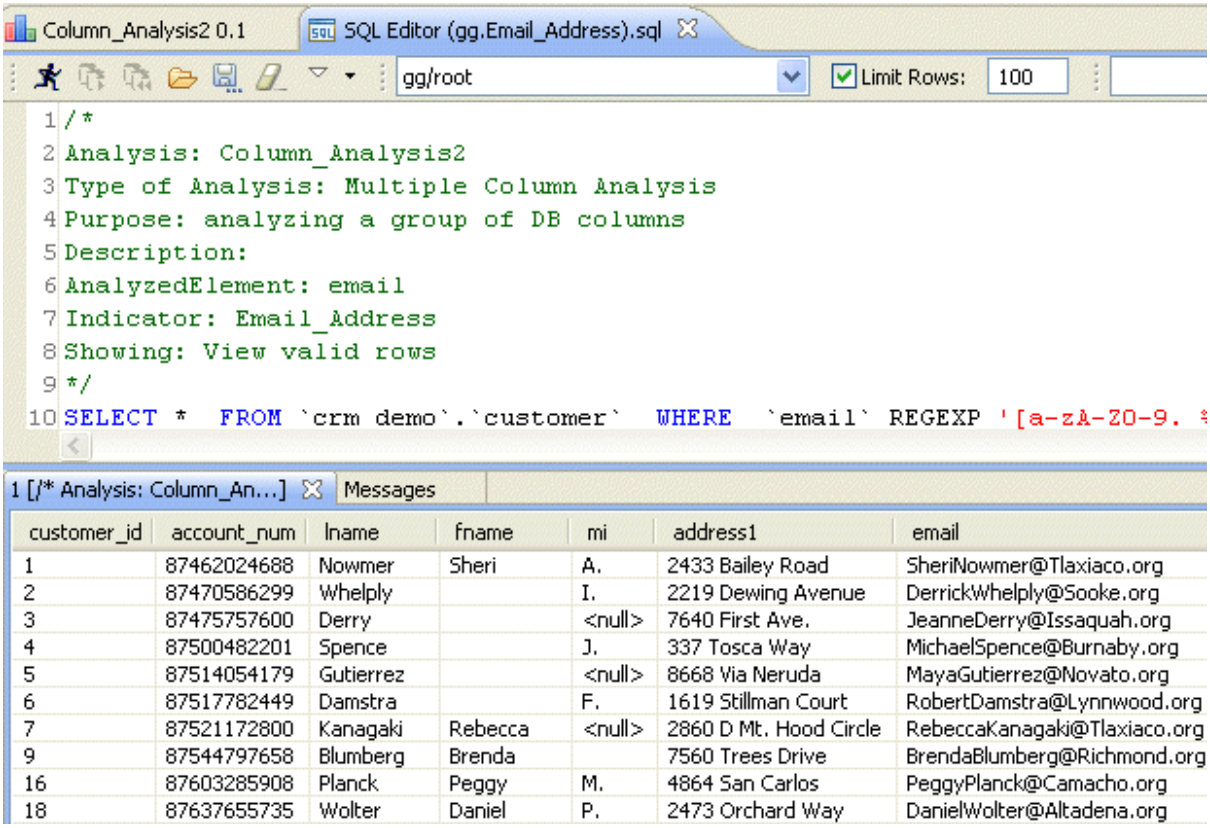
The generated graphic for the pattern matching is displayed accompanied with a table that details the matching results.



5. Right-click the pattern line in the **Pattern Matching** table and select:

Option	To...
View valid/invalid values	open a view of all valid/invalid values measured against the pattern used on the selected column
View valid/invalid rows	open a view of all valid/invalid rows measured against the pattern used on the selected column

When using the SQL engine, the view opens in the **Data Explorer** perspective listing valid/invalid rows or values of the analyzed data according to the limits set in the data explorer.


Column_Analysis2 0.1 | SQL Editor (gg.Email_Address).sql

gg/root | Limit Rows: 100

```
1 /*
2 Analysis: Column_Analysis2
3 Type of Analysis: Multiple Column Analysis
4 Purpose: analyzing a group of DB columns
5 Description:
6 AnalyzedElement: email
7 Indicator: Email_Address
8 Showing: View valid rows
9 */
10 SELECT *  FROM `crm demo`.`customer`  WHERE  `email` REGEXP '[a-zA-Z0-9. %
```

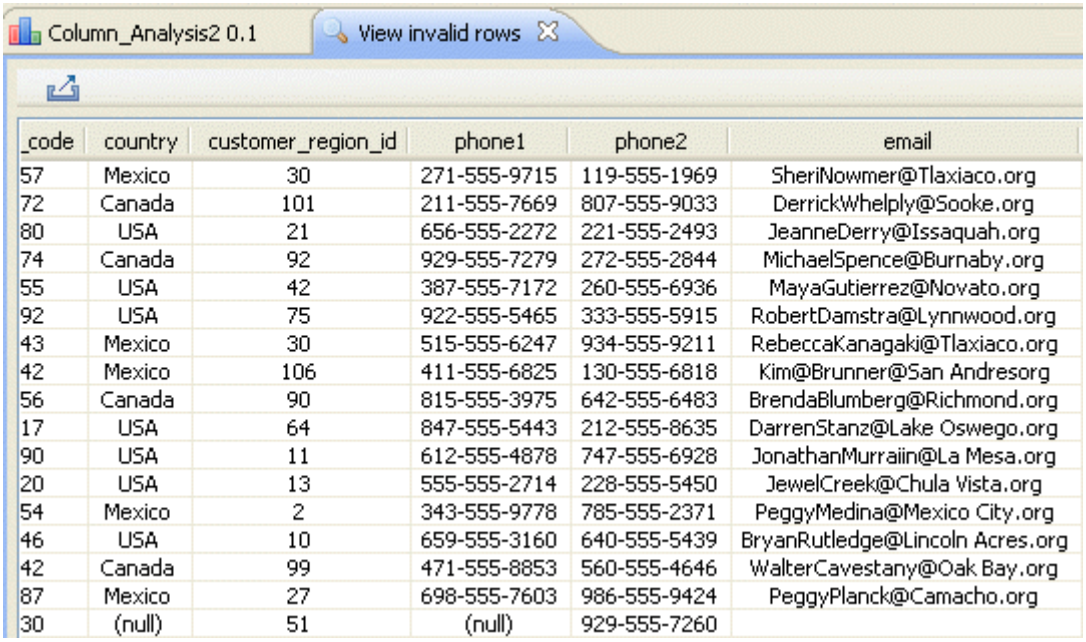
1 [/* Analysis: Column_An...] | Messages

customer_id	account_num	lname	fname	mi	address1	email
1	87462024688	Nowmer	Sheri	A.	2433 Bailey Road	SheriNowmer@Tlaxiaco.org
2	87470586299	Whelply		I.	2219 Dewing Avenue	DerrickWhelply@Sooke.org
3	87475757600	Derry		<null>	7640 First Ave.	JeanneDerry@Issaquah.org
4	87500482201	Spence		J.	337 Tosca Way	MichaelSpence@Burnaby.org
5	87514054179	Gutierrez		<null>	8668 Via Neruda	MayaGutierrez@Novato.org
6	87517782449	Damstra		F.	1619 Stillman Court	RobertDamstra@Lynnwood.org
7	87521172800	Kanagaki	Rebecca	<null>	2860 D Mt. Hood Circle	RebeccaKanagaki@Tlaxiaco.org
9	87544797658	Blumberg	Brenda		7560 Trees Drive	BrendaBlumberg@Richmond.org
16	87603285908	Planck	Peggy	M.	4864 San Carlos	PeggyPlanck@Camacho.org
18	87637655735	Wolter	Daniel	P.	2473 Orchard Way	DanielWolter@Altadena.org

This explorer view will also give some basic information about the analysis itself. Such information is of great help when working with multiple analysis at the same time.

The data explorer does not support connections which has empty user name, such as Single sign-on of MS SQL Server. If you analyze data using such connection and you try to view data rows and values in the ***Data Explorer*** *perspective, a warning message prompt you to set your connection credentials to the SQL Server.*

When using the Java engine, the view opens in the **Profiling** perspective of the studio listing the number of valid/ invalid data according to the row limit you set in the **Analysis parameters** view of the analysis editor. For more information, see section *Using the Java or the SQL engine*.


Column_Analysis2 0.1 | View invalid rows

_code	country	customer_region_id	phone1	phone2	email
57	Mexico	30	271-555-9715	119-555-1969	SheriNowmer@Tlaxiaco.org
72	Canada	101	211-555-7669	807-555-9033	DerrickWhelply@Sooke.org
80	USA	21	656-555-2272	221-555-2493	JeanneDerry@Issaquah.org
74	Canada	92	929-555-7279	272-555-2844	MichaelSpence@Burnaby.org
55	USA	42	387-555-7172	260-555-6936	MayaGutierrez@Novato.org
92	USA	75	922-555-5465	333-555-5915	RobertDamstra@Lynnwood.org
43	Mexico	30	515-555-6247	934-555-9211	RebeccaKanagaki@Tlaxiaco.org
42	Mexico	106	411-555-6825	130-555-6818	Kim@Brunner@San Andresorg
56	Canada	90	815-555-3975	642-555-6483	BrendaBlumberg@Richmond.org
17	USA	64	847-555-5443	212-555-8635	DarrenStanz@Lake Oswego.org
90	USA	11	612-555-4878	747-555-6928	JonathanMurraiin@La Mesa.org
20	USA	13	555-555-2714	228-555-5450	JewelCreek@Chula Vista.org
54	Mexico	2	343-555-9778	785-555-2371	PeggyMedina@Mexico City.org
46	USA	10	659-555-3160	640-555-5439	BryanRutledge@Lincoln Acres.org
42	Canada	99	471-555-8853	560-555-4646	WalterCavestany@Oak Bay.org
87	Mexico	27	698-555-7603	986-555-9424	PeggyPlanck@Camacho.org
30	(null)	51	(null)	929-555-7260	

You can save the executed query and list it under the **Libraries > Source Files** folders in the **DQ Repository** tree view if you click the save icon on the SQL editor toolbar. For more information, see section *Saving the queries executed on indicators*.

For more information about the data explorer Graphical User Interface, see appendix *Data Explorer management GUI*.

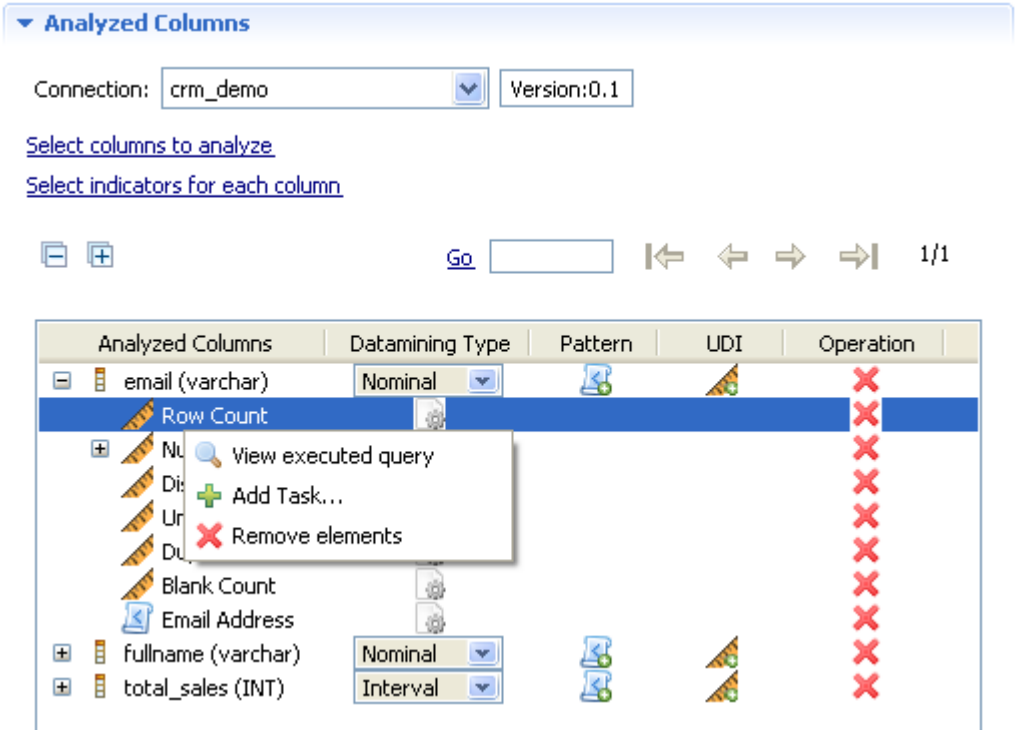
5.3.7. Saving the queries executed on indicators

From the studio and in the **Data Explorer** perspective, you can view the queries executed on different indicators used in an analysis. From the data explorer, you will be able to save the query and list it under the **Libraries > Source Files folders** in the **DQ Repository** tree view.

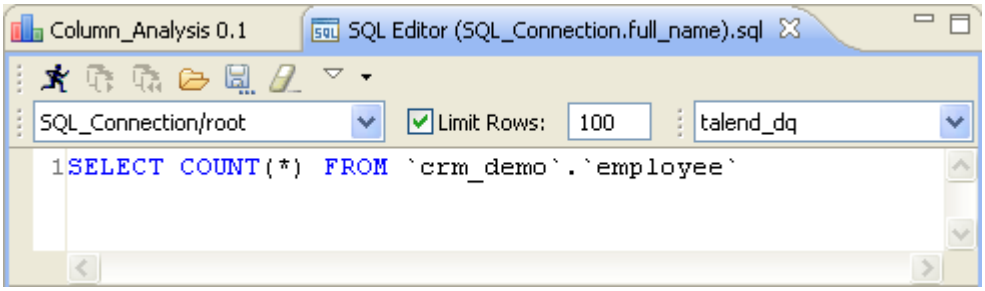
Prerequisite(s): At least one analysis with indicators has been created.

To save any of the queries executed on an indicator set in a column analysis, do the following:

1. In the column analysis editor, right-click any of the used indicators to open a contextual menu.

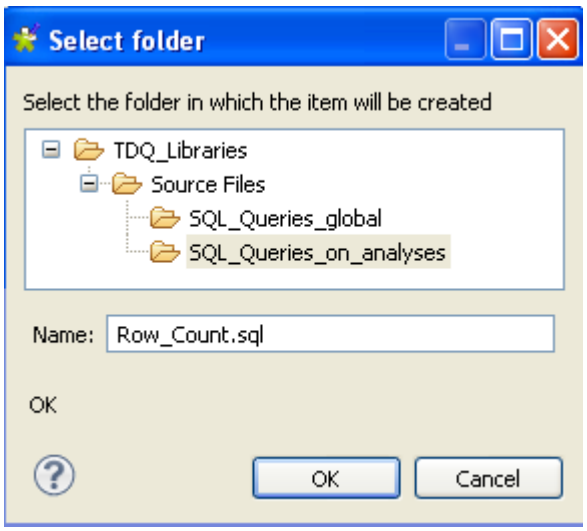


2. Select **View executed query** to open the data explorer on the query executed on the selected indicator.



*The data explorer does not support connections which has empty user name, such as Single sign-on of MS SQL Server. If you analyze data using such connection and you try to view the executed queries in the **Data Explorer** perspective, a warning message prompt you to set your connection credentials to the SQL Server.*

3. Click the save icon on the editor toolbar to open the **[Select folder]** dialog box

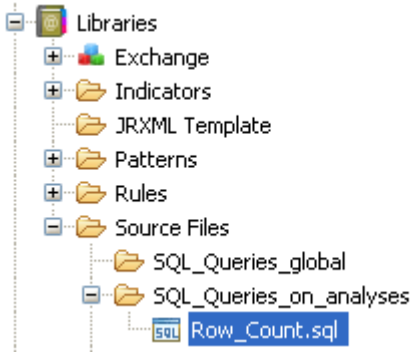


4. Select the **Source Files** folder or any sub-folder under it and enter in the **Name** field a name for the open query.

   Make sure that the name you give to the open query is always followed by *.sql*. Otherwise, you will not be able to save the query.

5. Click **OK** to close the dialog box.

   The selected query is saved under the selected folder in the **DQ Repository** tree view.



5.3.8. Creating table and columns analyses in shortcut procedures

In the studio, you can use simplified ways to create one or multiple column analyses. All what you need to do is to start from the table name or the column name under the relevant **DB Connection** folder in the **DQ Repository** tree view.

However, the options you have to create column analyses if you start from the table name are different from those you have if you start from the column name.

To create a column analysis directly from the relevant table name in the **DB Connection**, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Browse to the table that holds the column(s) you want to analyze and right-click it.
3. From the contextual menu, select:

Item	To...

Table analysis	analyze the selected table using SQL business rules. For more information on the Simple Statistics indicators, see chapter *Table analyses*.
Column analysis	analyze all the columns included in the selected table using the Simple Statistics indicators. For more information on the Simple Statistics indicators, see section *Simple statistics*.
Pattern frequency analysis	analyze all the columns included in the selected table using the Pattern Frequency Statistics along with the Row Count and the Null Count indicators. For more information on the Pattern Frequency Statistics, see section *Pattern frequency statistics*.

The above steps replace the procedures outlined in section *Defining the columns to be analyzed and setting indicators*. Now, you proceed following the steps outlined in section *Finalizing the column analysis before execution*.

To create a column analysis directly from the column name in the **DB Connection**, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Browse to the column(s) you want to analyze and right-click it/them.
3. From the contextual menu, select:

Item	To...
Analyze	create an analysis for the selected column you must later set the indicators you want to use to analyze the selected column. For more information on setting indicators, see section *How to set indicators for the column(s) to be analyzed*. For more information on accomplishing the column analysis, see section *Finalizing the column analysis before execution*.
Analyze correlation	perform column correlation analyses between nominal and interval columns or nominal and date columns in database tables. For more information, see chapter *Table analyses*.
Nominal value analysis	analyze minimal correlations between nominal columns in the same table and gives the result in a chart. For more information, see section *Nominal correlation analysis*.
Simple analysis	analyze the selected column using the Simple Statistics indicators. For more information on the Simple Statistics indicators, see section *Simple statistics*.
Pattern frequency analysis	analyze the selected column using the Pattern Frequency Statistics along with the Row Count and the Null Count indicators. For more information on the Pattern Frequency Statistics, see section *Pattern frequency statistics*.

The above steps replace one of or both of the procedures outlined in section *Defining the columns to be analyzed and setting indicators*. Now, you proceed following the same steps outlined in section *Finalizing the column analysis before execution*.

5.4. Analyzing master data on an MDM server

You can use the studio to analyze master data in one or multiple data containers on the MDM server and execute the created analyses using the SQL or Java engines. For further information on these engines, see section *Using the Java or the SQL engine*.

You can also analyze a set of columns, for more information, see section *Analyzing tables on MDM servers*.

5.4.1. Defining the business entities to be analyzed and setting indicators

The sequence of analyzing a business entity involves the following steps:

1. Defining the business entity to be analyzed.

   For more information, see section *How to define the columns to be analyzed*.

2. Settings predefined system indicators for the business entity.

   For more information, see section *How to set indicators for the column(s) to be analyzed*. For more information on indicator types and indicator management, see section *Indicators*.

You can also use Java user-defined indicators when analyzing master data on the condition that a Java user-defined indicator is already created. For further information, see section *How to define Java user-defined indicators*.

The following sections provide detailed description on each of the preceding steps.

5.4.1.1. How to define the business entities to be analyzed

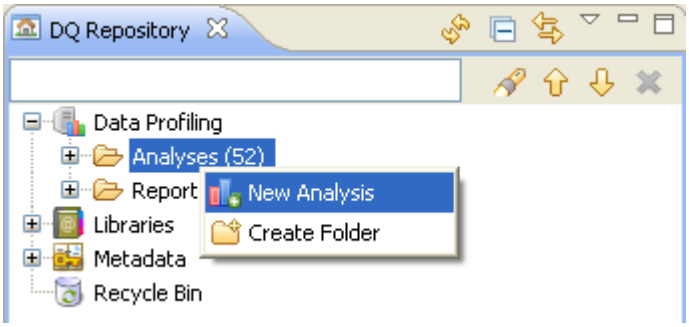
The first step in analyzing the content of one or multiple business entities is to define these entities.

Prerequisite(s): At least one MDM connection is set in the **Profiling** perspective in the studio. For further information, see section *Connecting to an MDM server*.

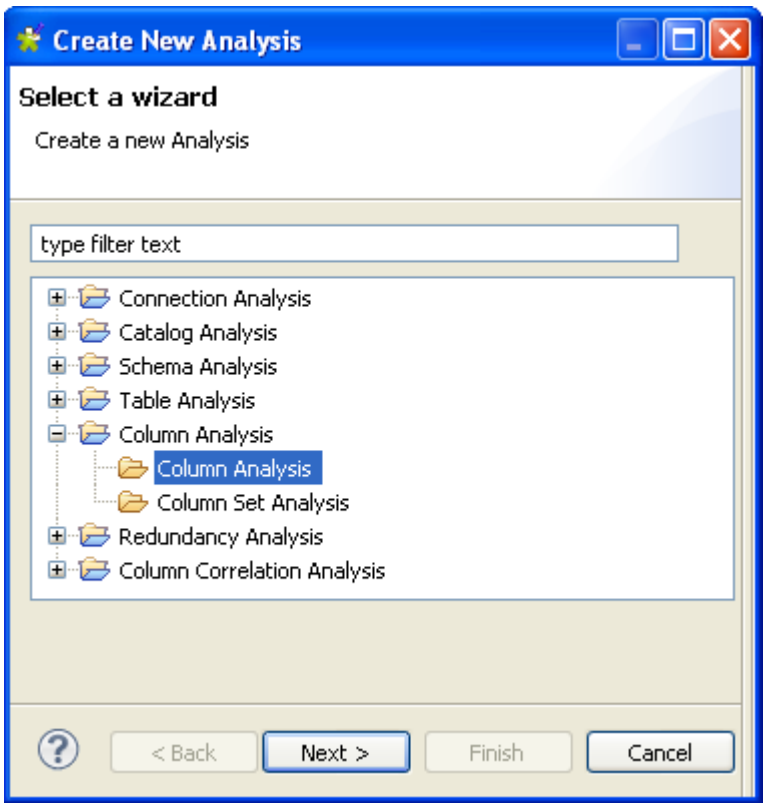
Defining the analysis

1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.

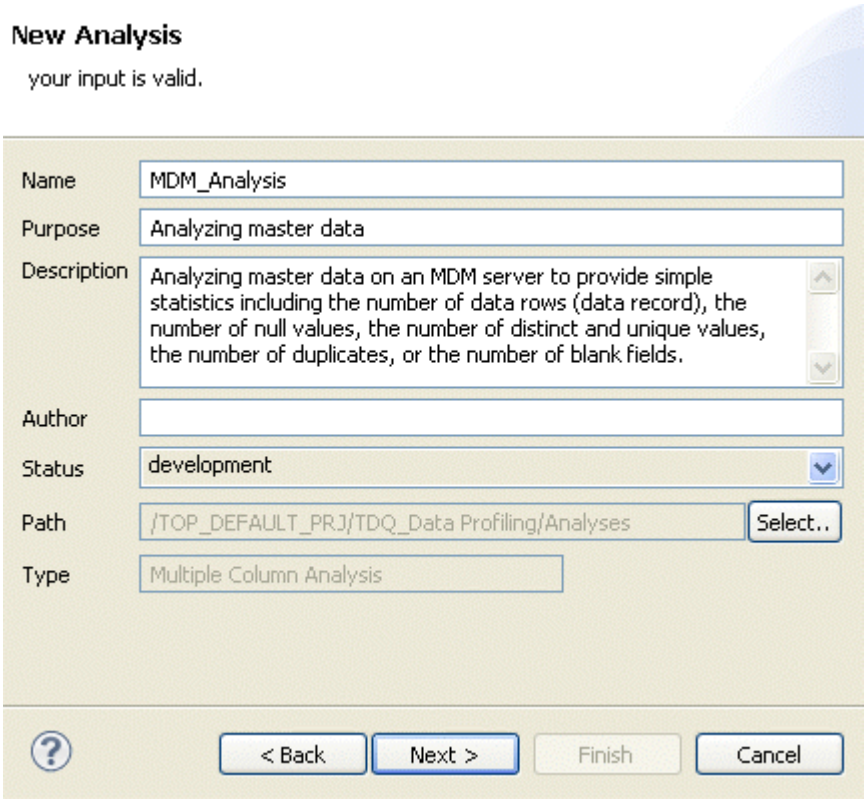
2. Right-click the **Analysis** folder and select **New Analysis**.



   The **[Create New Analysis]** wizard opens.



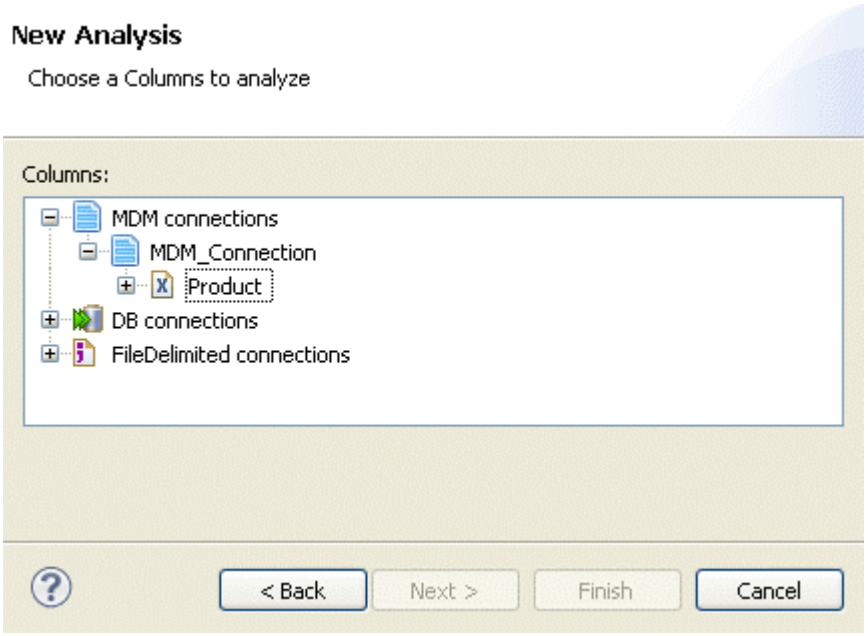
3. Expand the **Column Analysis** folder and click **Column Analysis**.
4. Click the **Next** button to proceed to the next step.



5. In the **Name** field, enter a name for the current column analysis.

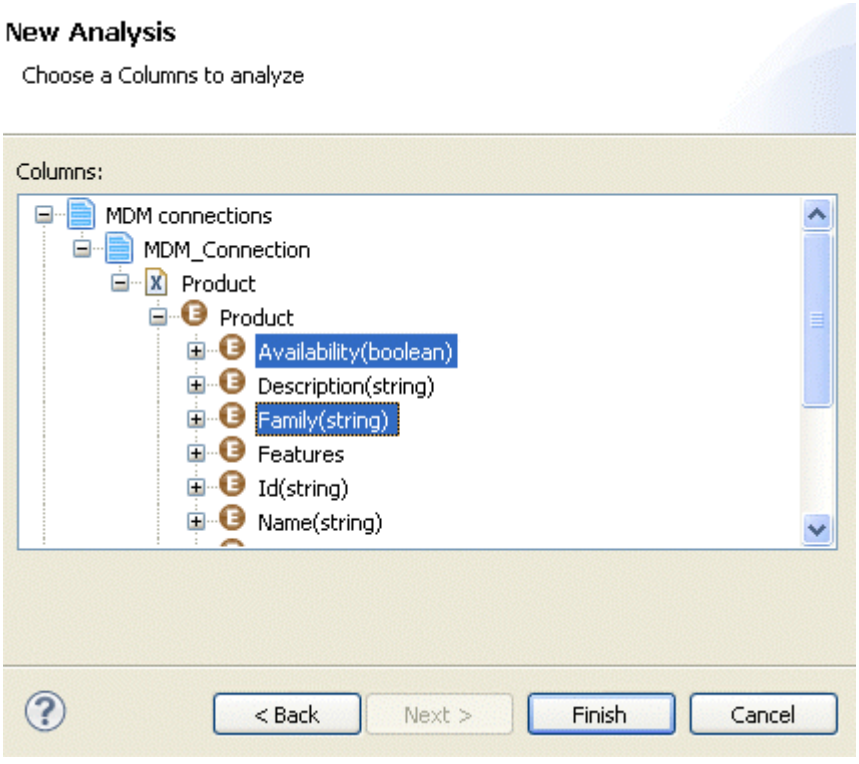
Space is not acceptable when typing in the analysis name in this field.

6. If required, set the analysis metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



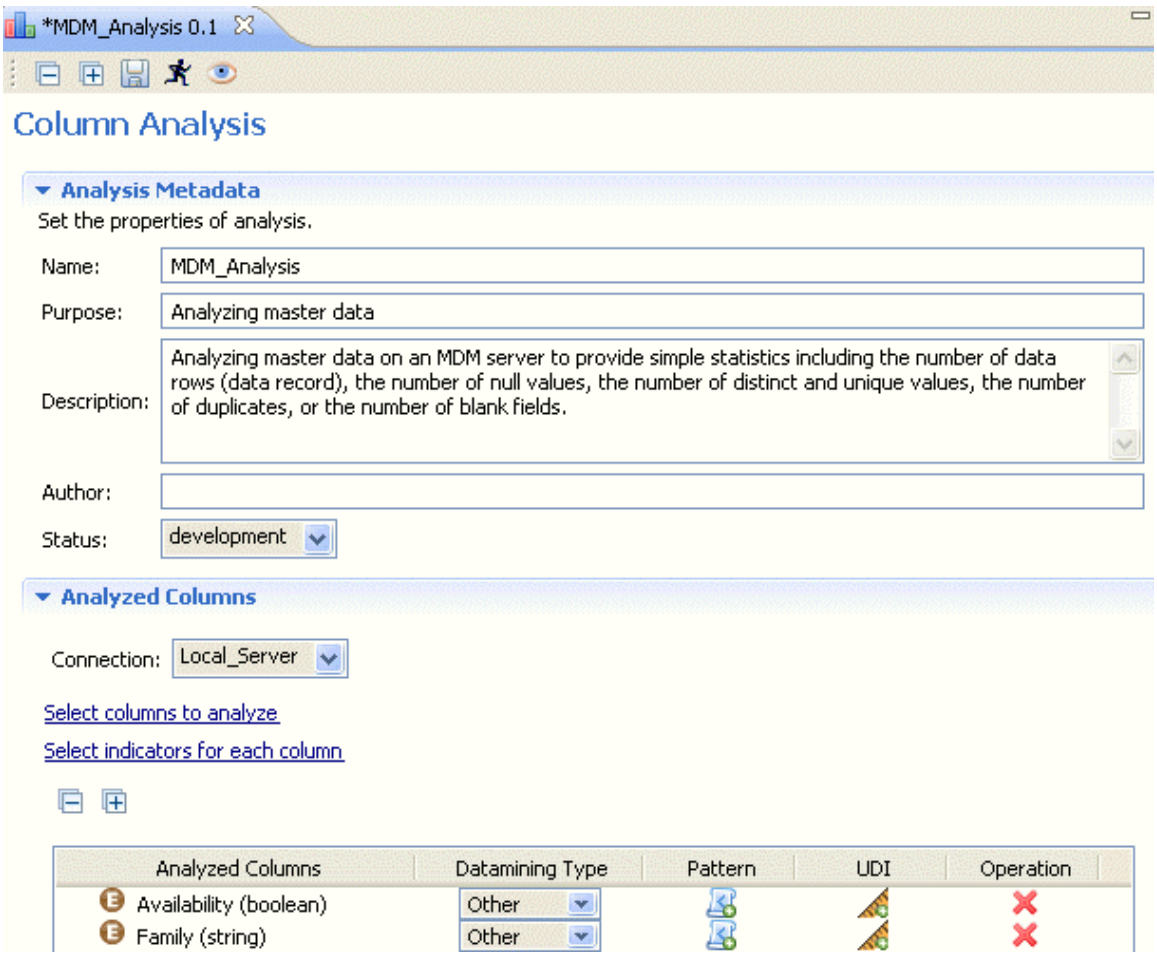
Selecting the business entity you want to analyze

1. Expand **MDM connections** and browse through the data containers on the MDM server to reach the business entity (column) holding the data you want to analyze.



2. Select the columns to analyze and then click **Finish** to close the wizard.

   A file for the newly created analysis is displayed under the **Analysis** node in the **DQ Repository** tree view, and the analysis editor opens with the defined analysis metadata.

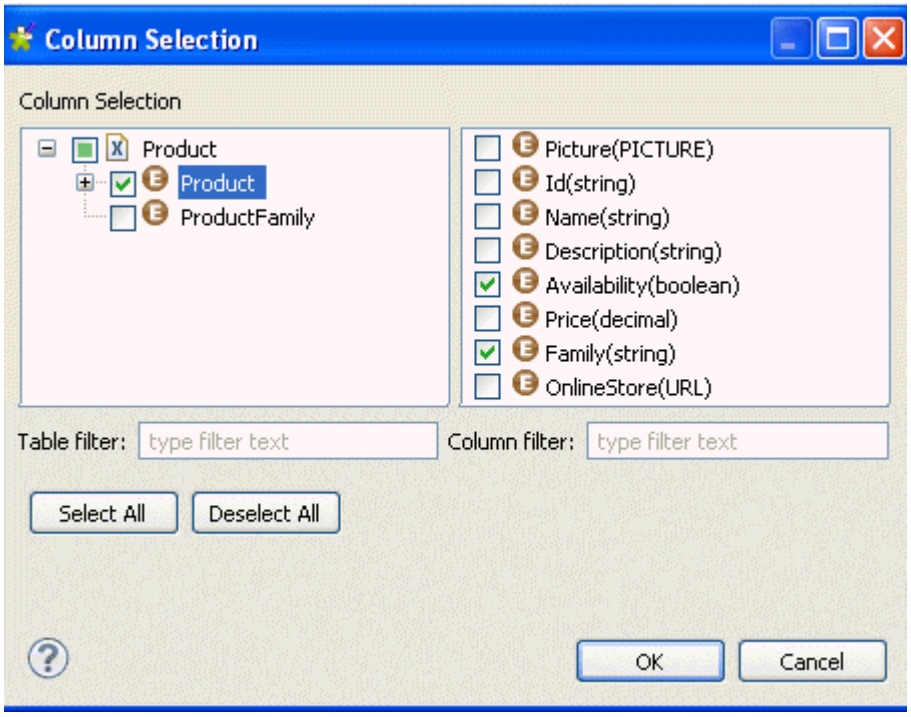


The display of the connection editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

3. Click the **Analyzed Column** tab to open the corresponding view, if not already open.

   The **Connection** field has the connection name to the MDM server that holds the items you want to analyze and these items (columns) are already listed in the column list.

4. If required, click the **Select columns to analyze** link to open a dialog box where you can modify your column selection. You can filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

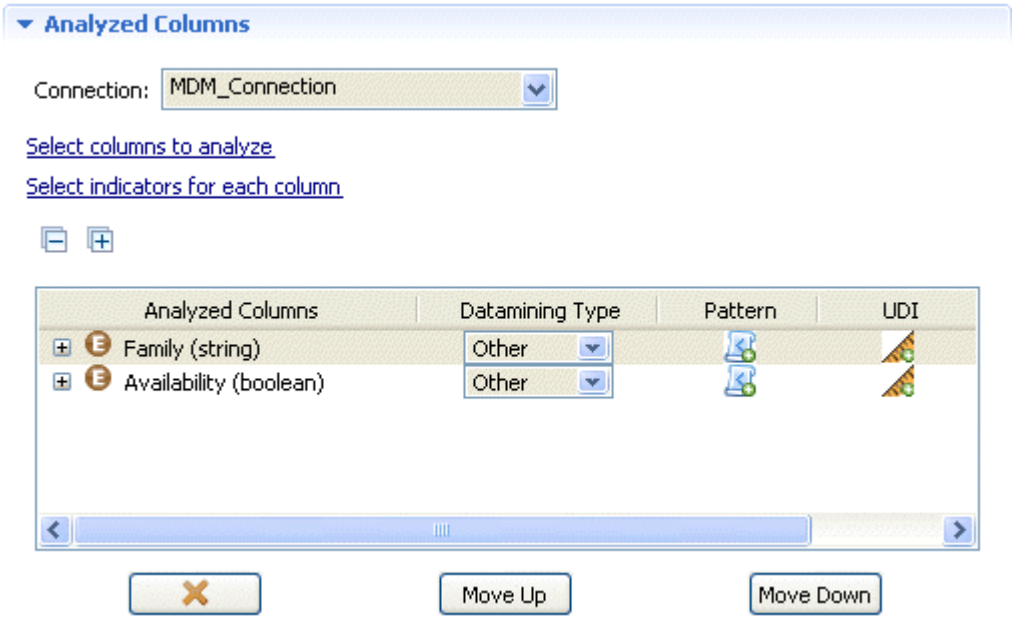


5. Click the business entity name to display all its record in the right-hand panel of the **[Column Selection]** dialog box.

6. In the list to the right, select the check boxes of the column(s) you want to analyze and click **OK** to proceed to the next step.

   The selected records display in the **Analyzed Column** view of the analysis editor.

   You can drag the records to be analyzed directly from the **DQ Repository** tree view to the column analysis editor.



7. If required, use the delete, move up or move down buttons to manage the analyzed columns.

   The data mining type is set to **Other** by default. For more information on data mining types in the studio, see section *Data mining types*.

If you right-click any of the listed records in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected record will be automatically located under the corresponding MDM connection in the tree view.

8. Click the save icon on the toolbar of the analysis editor.

5.4.1.2. How to set system indicators for the records to be analyzed

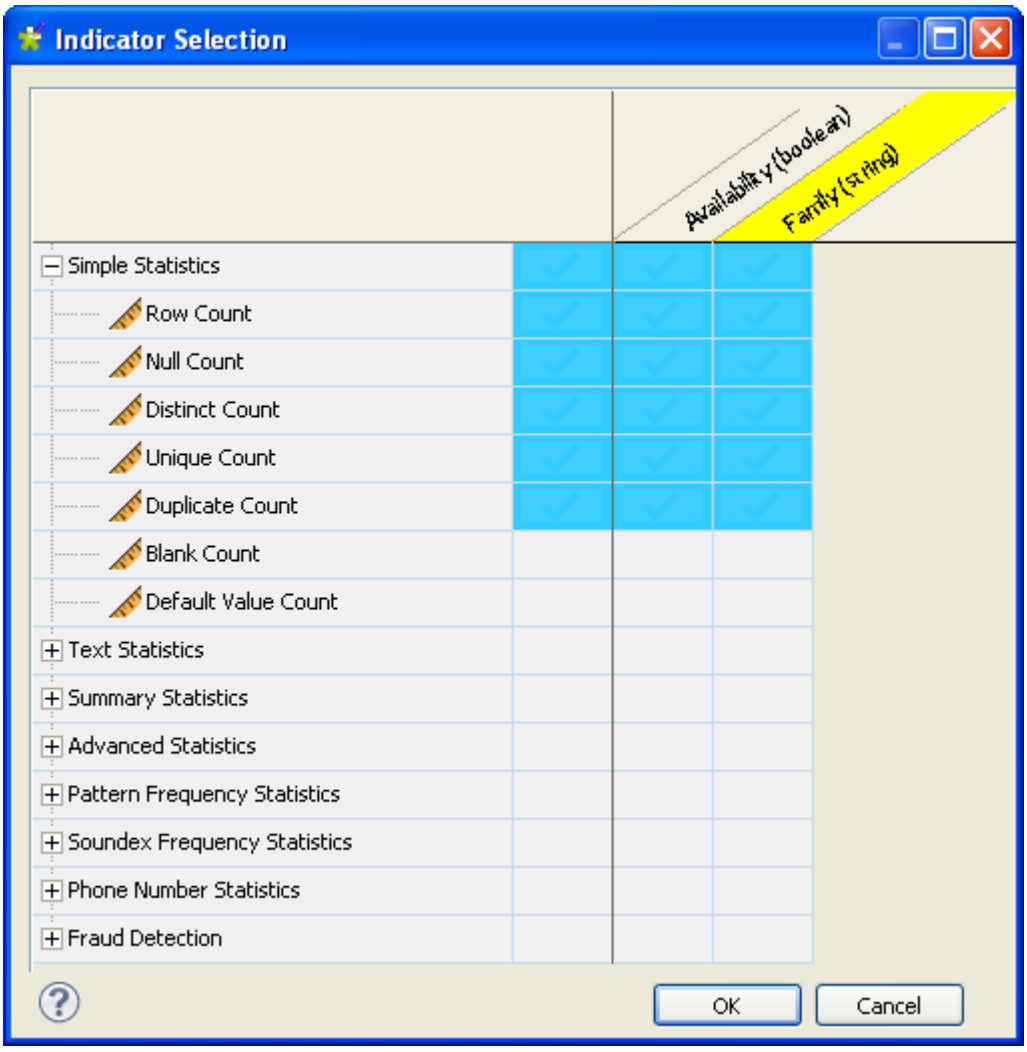
The second step after defining the records to be analyzed is to set the simple statistics indicators for each of the defined records.

You can also use Java user-defined indicators when analyzing master data on the condition that a Java user-defined indicator is already created. For further information, see section *How to define Java user-defined indicators*.

Prerequisite(s): An analysis of a business entity is open in the analysis editor in the studio. For more information, see section *How to define the columns to be analyzed*.

To set system indicators for the record(s) to be analyzed, do the following:

1. In the analysis editor, click **Analyzed Columns** to open the analyzed columns view.
2. Click **Select indicators for each column** to open the **[Indicator Selection]** dialog box.



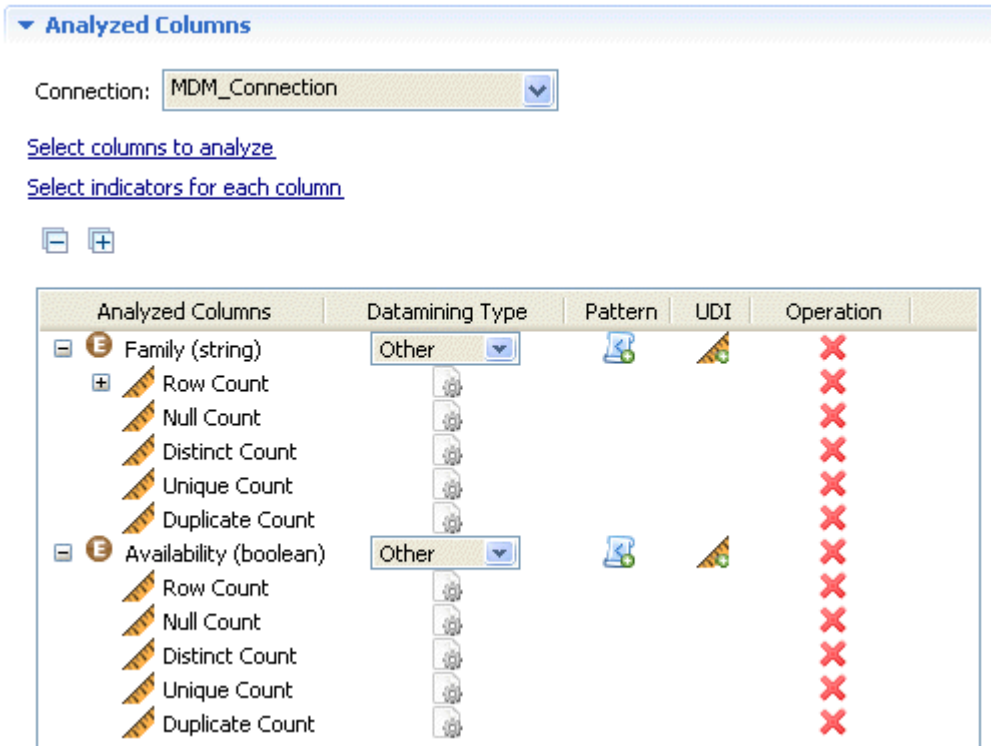
In this dialog box, you can change column positions by dropping them with the cursor.

3. If you are analyzing very large number of columns, place the cursor in the top/bottom right corner of the **[Indicator Selection]** dialog box to access the columns to the very right.

   Similarly, place the cursor in the top/bottom left corner of the **[Indicator Selection]** dialog box to access the columns to the very left.

4. Click in the simple statistics cell to set these indicators for the MDM records and then click **OK** to proceed to the next step.

   The selected indicators are attached to the analyzed records in the **Analyzed Columns** view.



5. Click the save icon on the toolbar of the analysis editor.

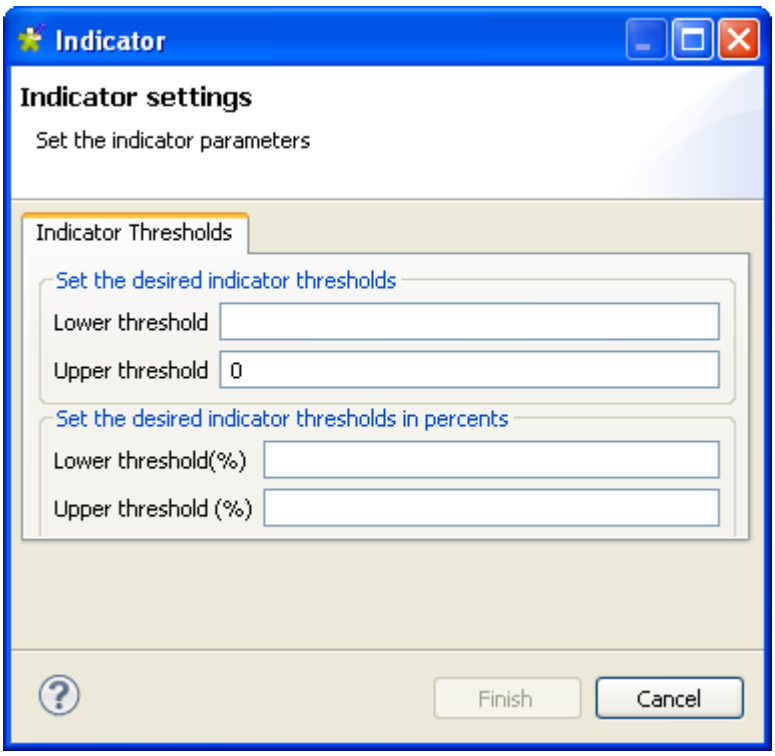
5.4.1.3. How to set options for system indicators

Prerequisite(s): An analysis of MDM records is open in the analysis editor in the studio. For more information, see section *Defining the columns to be analyzed and setting indicators*.

To set options for system indicators, do the following:

1. In the analysis editor, click **Analyzed Columns** to open the analyzed columns view.

2. Click the option icon next to the defined indicator to open the dialog box where you can set options for the given indicator.



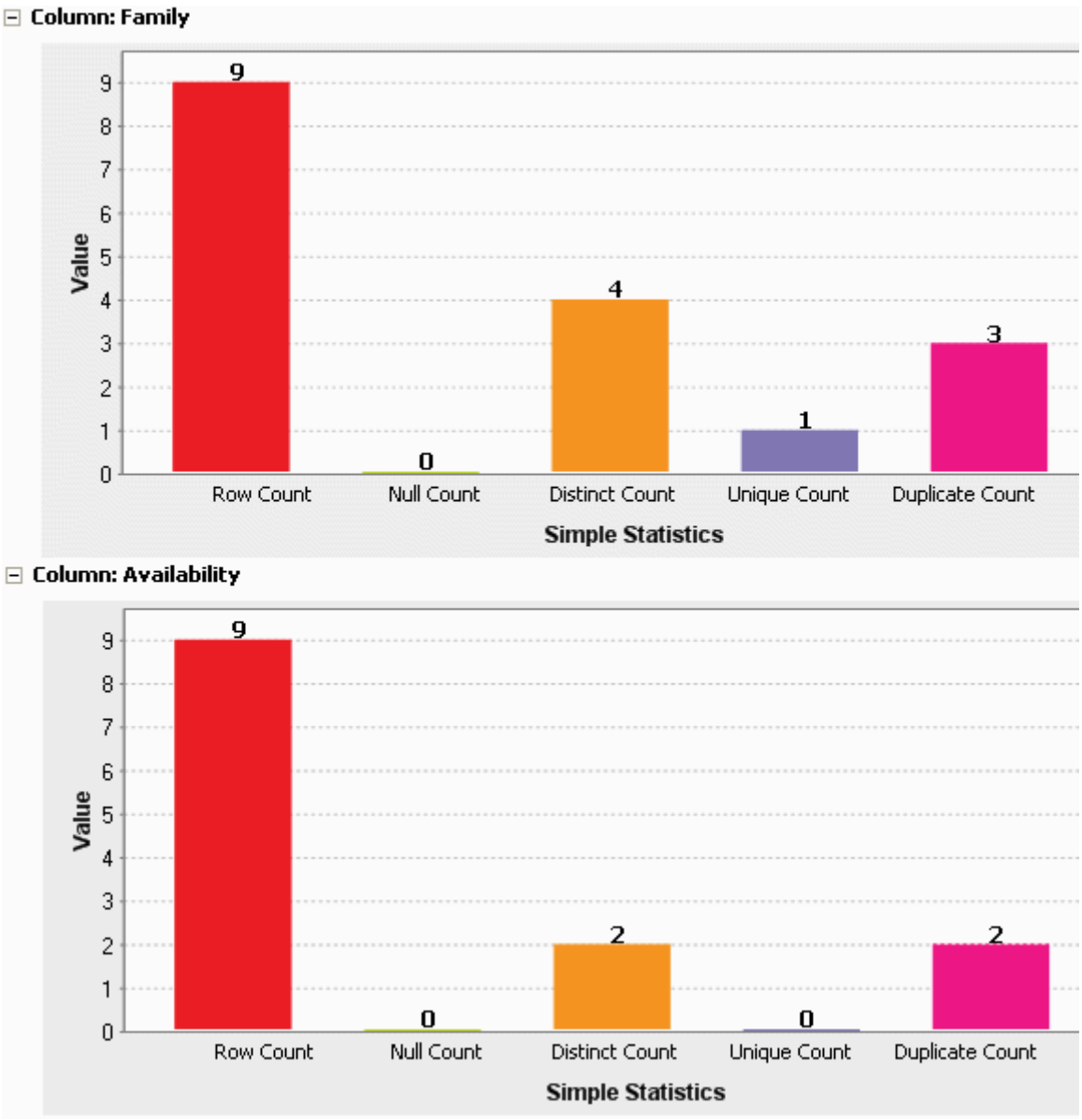
Running the analysis will show if these thresholds are violated through appending a warning icon on such a result and the result itself will be in red. For further information, see section *How to access the detailed view of the analysis results*.

Indicators settings dialog boxes differ according to the parameters specific for each indicator. For more information about different indicator parameters, see section *Indicator parameters*.

3. Set the parameters for the given indicator.
4. Click **Finish** to close the dialog box.
5. In the analysis editor, click the **Data Filter** tab to display the corresponding view and filter master data through XQuery clauses, if required.
6. In the analysis editor, click the **Analysis Parameters** tab to display the corresponding view and select the engine you want to use to run the analysis. For more information on available engines, see section *Using the Java or the SQL engine*.
7. Click the save icon on the toolbar of the analysis editor and then press **F6** to execute the analysis.

The **Graphics** panel to the right of the analysis editor displays a group of graphic(s), each corresponding to one of the analyzed records.

To view the different graphics associated with all analyzed records, you may need to navigate through the different pages in the **Graphics** panel using the toolbar on the upper-right corner.



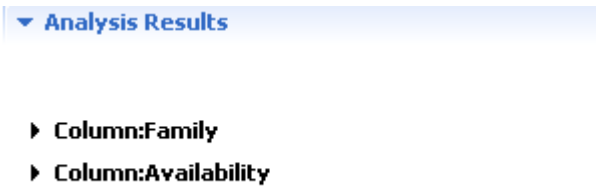
5.4.2. Accessing the detailed view of the master data analysis

Prerequisite(s): An analysis of a business entity is defined and executed in the **Profiling** perspective of the studio. For more information, see section *Defining the columns to be analyzed and setting indicators*.

To access a more detailed view of the analysis results, do the following:

1. Click the **Analysis Results** tab at the bottom of the analysis editor to open the corresponding view.

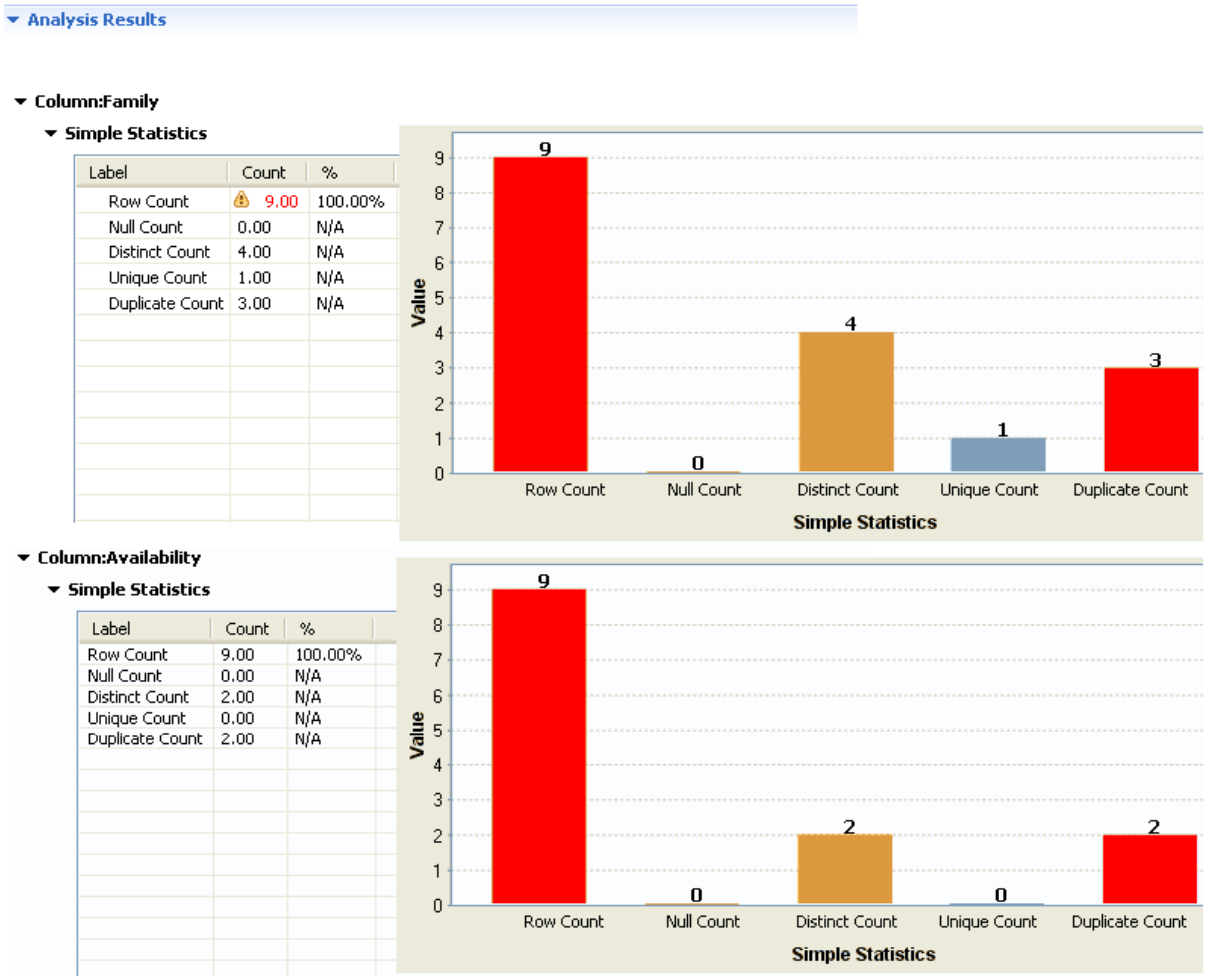
2. Click **Analysis Results** and then the name of the analyzed column for which you want to display the detailed results.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

The detailed analysis results view shows the generated graphics for the analyzed columns accompanied with tables that detail the statistic results.

Below are the tables that accompany the Simple Statistics graphics in the **Analysis Results** view for the analyzed records in the procedure outlined in section *Defining the columns to be analyzed and setting indicators*.



5.4.3. Analyzing master data in shortcut procedures

From the studio, you can profile the data on an MDM server using a simplified way. All what you need to do is to start from the column name under **Metadata > MDM connections** folders in the **DQ Repository** tree view.

For further information, see section *Creating table and columns analyses in shortcut procedures*.

5.5. Analyzing data in a file

You can create a column analysis on a delimited file and execute the created analyses using the Java engine.

From the studio, you can also analyze a set of columns, for more information, see section *Analyzing tables in delimited files*.

5.5.1. Analyzing columns in a delimited file

The sequence of profiling data in a delimited file involves the following steps:

1. defining the columns to be analyzed.

   For more information, see section *How to define the columns to be analyzed*.

2. settings predefined system indicators for the defined columns.

   For more information, see section *How to set indicators for the column(s) to be analyzed*. For more information on indicator types and indicator management, see section *Indicators*.

3. setting patterns for the defined columns. For more information, see section *Patterns*.

You can also use Java user-defined indicators when analyzing columns in a delimited file on the condition that a Java user-defined indicator is already created. For further information, see section *How to define Java user-defined indicators*.

The following sections provide a detail description on each of the preceding steps.

5.5.1.1. How to define the columns to be analyzed

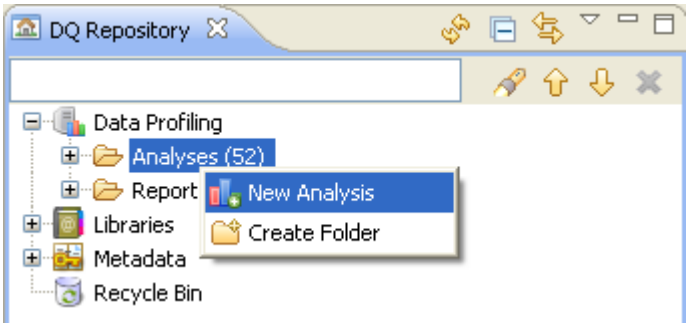
The first step in analyzing the content of a delimited file is to define the columns to be analyzed.

Prerequisite(s): At least one connection to a delimited file is set in the **Profiling** perspective of the studio. For further information, see section *How to connect to a delimited file*.

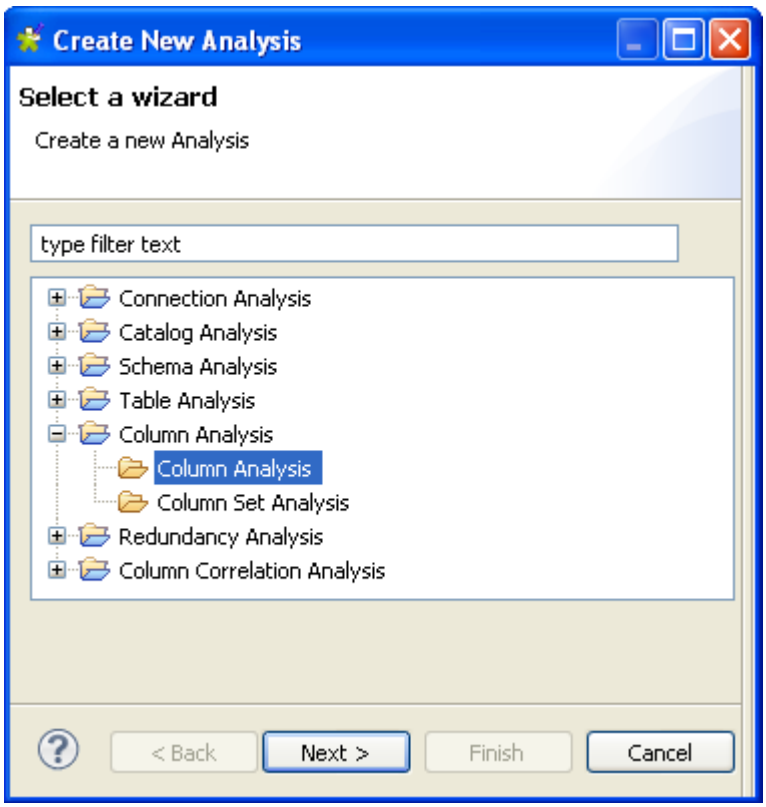
Defining the analysis

1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.

2. Right-click the **Analysis** folder and select **New Analysis**.

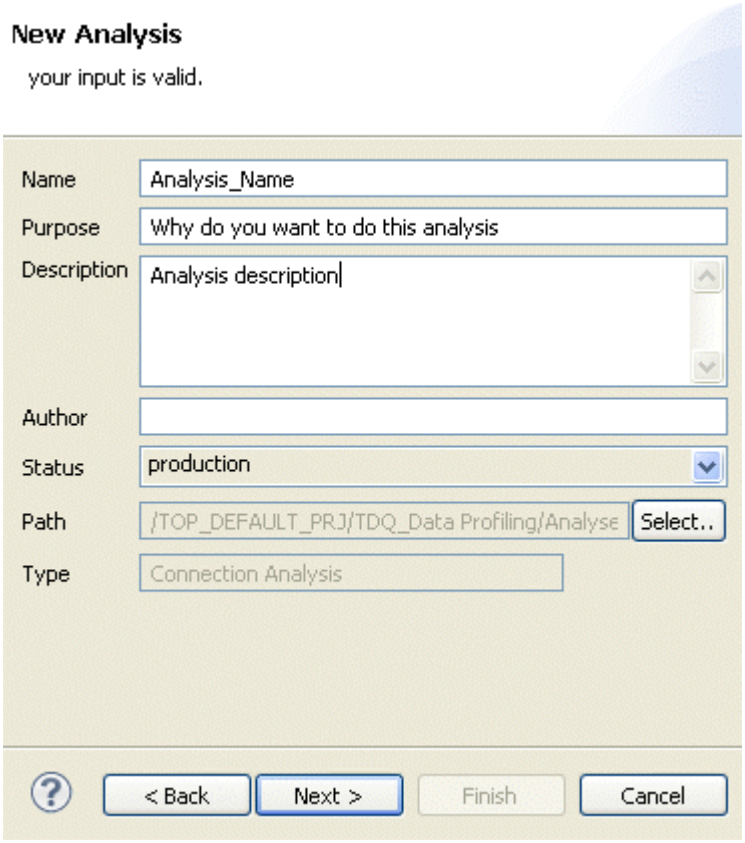


   The **[Create New Analysis]** wizard opens.



3. Expand the **Column Analysis** folder and click **Column Analysis**.

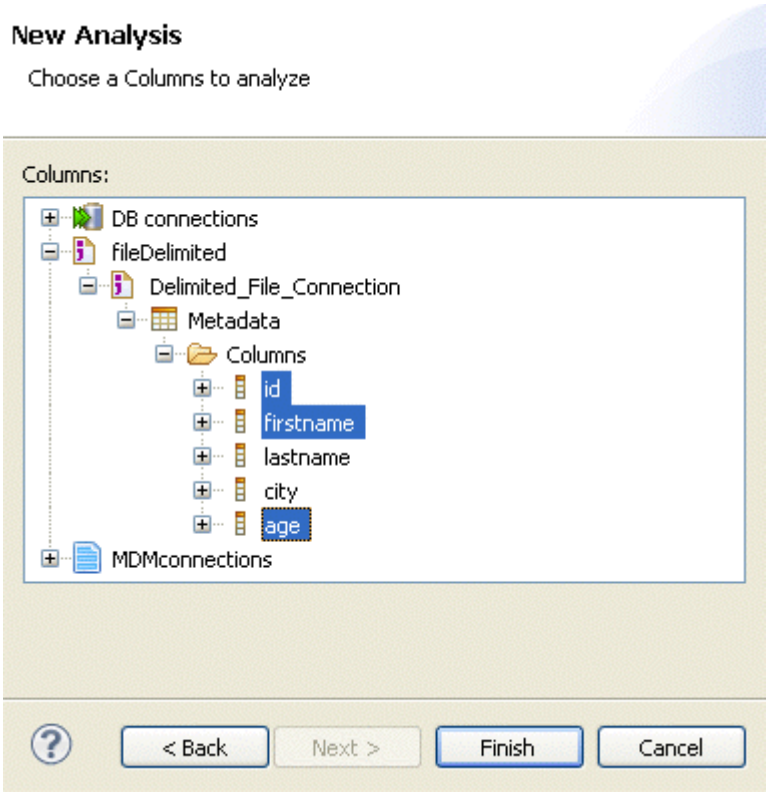
4. Click the **Next** button to proceed to the next step.



You can directly get to this step in the analysis creation wizard if you right-click the column to analyze in **Metadata > FileDelimited** and select **Column Analysis > Analyze**. For further information, see section *Creating table and columns analyses in shortcut procedures*.

5. In the **Name** field, enter a name for the current column analysis.

6. If required, set the analysis metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.

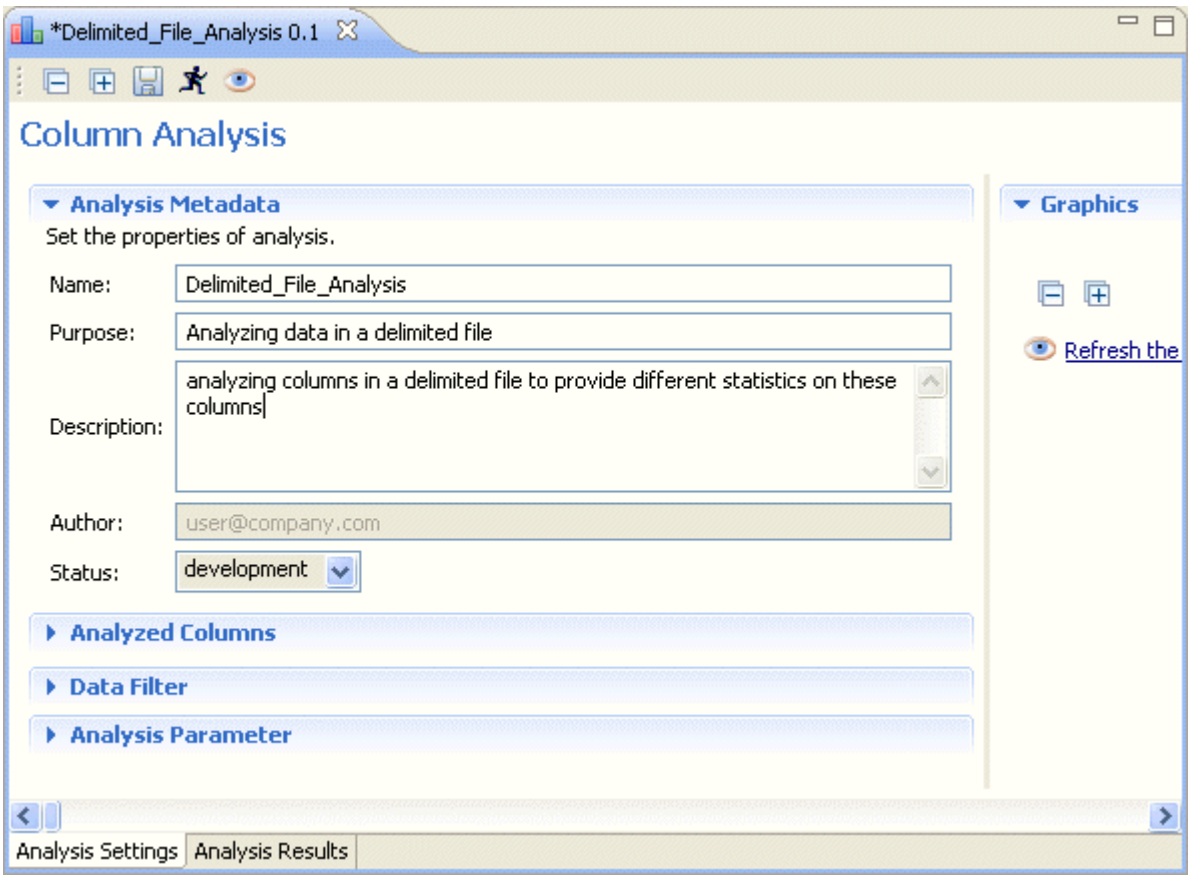


Selecting the columns in the delimited file

1. Expand **FileDelimited** and then browse to the columns you want to analyze.

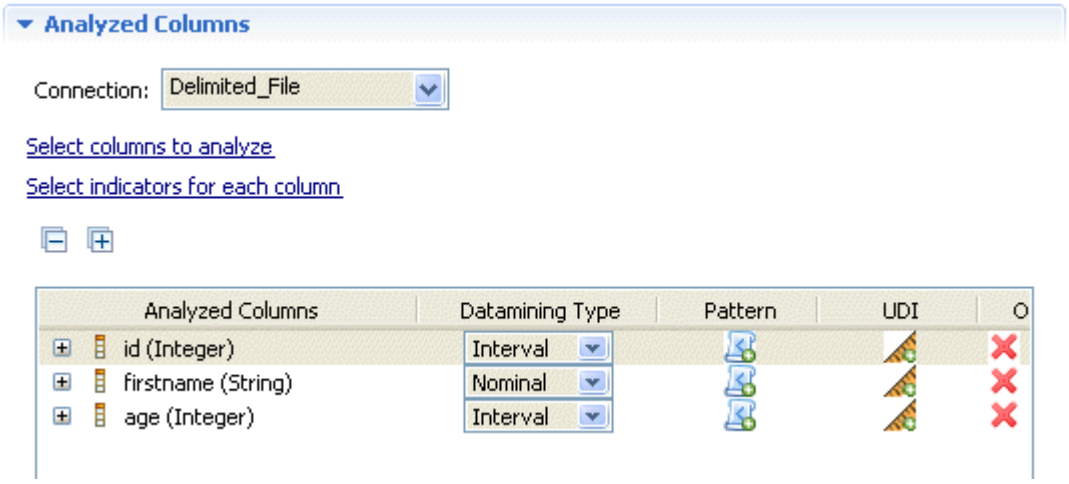
2. Select these columns and then click **Finish** to close the wizard.

   A file for the newly created analysis is displayed under the **Analyses** node in the **DQ Repository** tree view, and the analysis editor opens with the defined analysis metadata.



The display of the connection editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

3. Click **Analyzed Columns** to display the **Analyzed Columns** view.



You can also drop the columns to analyze directly from the **DQ Repository** tree view to the analysis editor.

The **Connection** field shows the selected connection and the columns you want to analyze are already listed in the column list.

4. If required, click the **Select columns to analyze** link to open a dialog box where you can modify your column selection.

   In this example, you want to analyze the *id*, *firstname* and *age* columns from the selected connection.

5. If required, use the delete, move up or move down buttons to manage the analyzed columns.

If you right-click any of the listed columns in the **Analyzed Columns** table and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding delimited file connection in the tree view.

6. Click the save icon on the toolbar of the analysis editor.

5.5.1.2. How to set system indicators for the columns to be analyzed

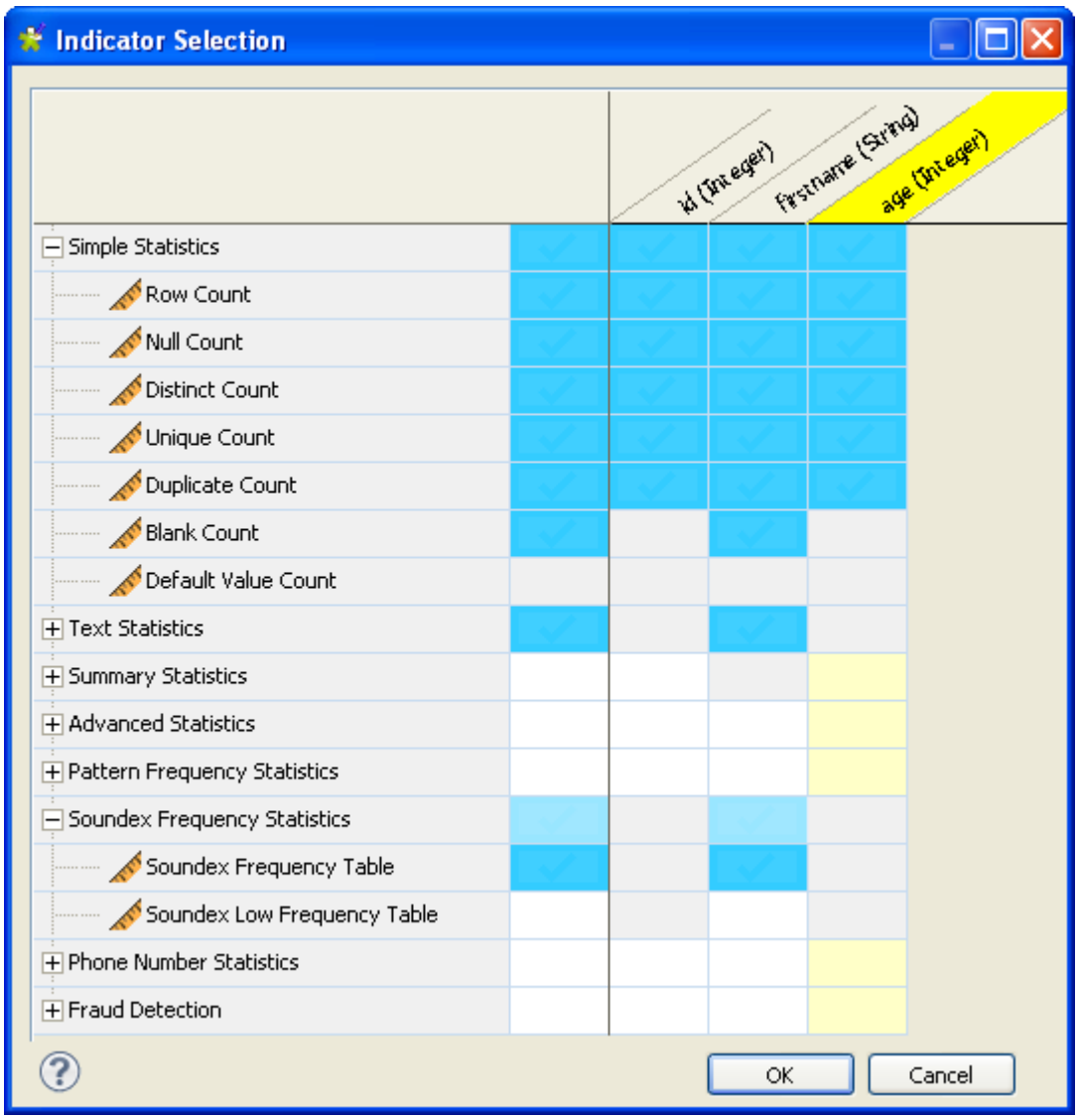
The second step after defining the columns to be analyzed is to set statistics indicators for each of the defined columns.

You can also use Java user-defined indicators when analyzing columns in a delimited file on the condition that a Java user-defined indicator is already created. For further information, see section *How to define Java user-defined indicators*.

Prerequisite(s): An analysis of a delimited file is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the columns to be analyzed*.

To set system indicators for the column(s) to be analyzed, do the following:

1. Follow the procedure outlined in section *How to define the columns to be analyzed*.
2. In the analysis editor, click **Analyzed Columns** to open the analyzed columns view.
3. Click **Select indicators for each column** to open the **[Indicator Selection]** dialog box.



In this dialog box, you can change column positions by dropping them with the cursor.

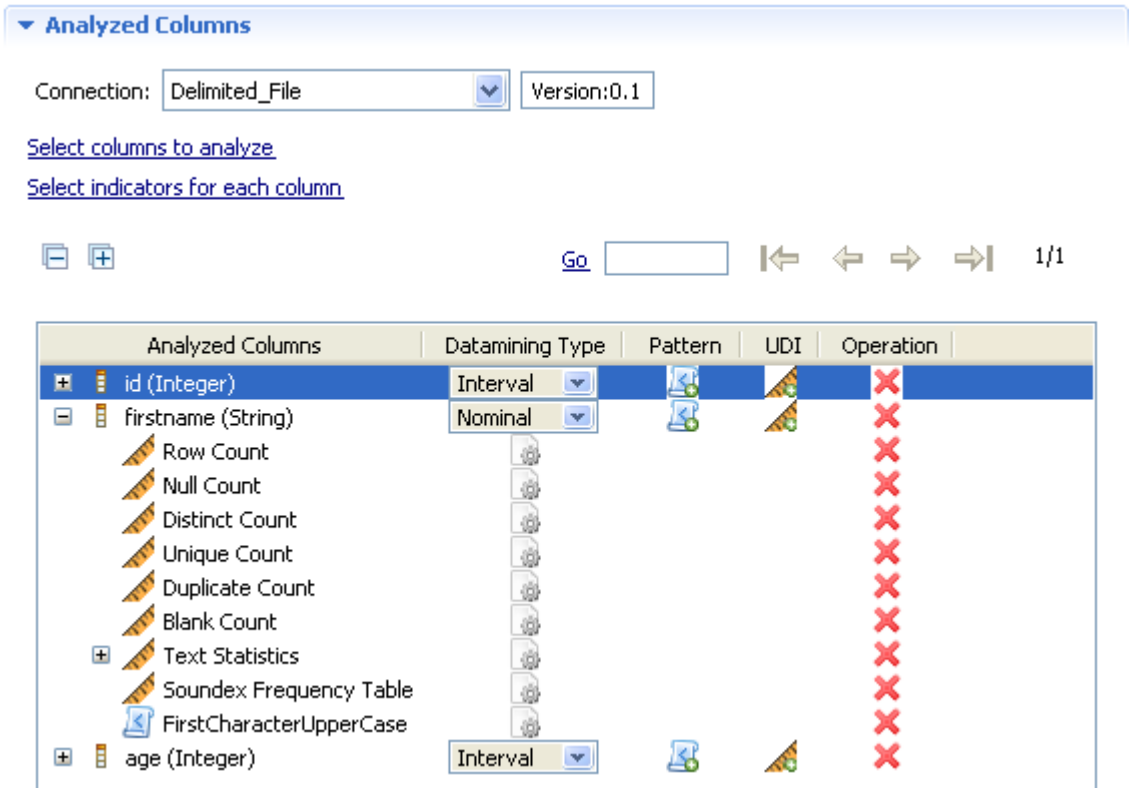
4. If you are analyzing very large number of columns, place the cursor in the top/bottom right corner of the **[Indicator Selection]** dialog box to access the columns to the very right.

   Similarly, place the cursor in the top/bottom left corner of the **[Indicator Selection]** dialog box to access the columns to the very left.

5. Click in the cells to set indicator parameters for the columns to be analyzed and then click **OK** to proceed to the next step.

   In this example, you want to set the **Simple Statistics** indicators on all columns, the **Text Statistics** indicators on the *firstname* column and the **Soundex Frequency Table** on the *firstname* column as well.

   The selected indicators are attached to the analyzed columns in the **Analyzed Columns** view.



6. Click the save icon on the toolbar of the analysis editor.

5.5.1.3. How to set options for system indicators

Prerequisite(s): An analysis of a delimited file is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the columns to be analyzed*, section *How to set indicators for the column(s) to be analyzed*.

To set options for system indicators used on the columns to be analyzed, do the following:

1. Follow the procedures outlined in section *How to define the columns to be analyzed* and section *How to set indicators for the column(s) to be analyzed*.
2. In the analysis editor, click **Analyzed Columns** to open the analyzed columns view.
3. In the **Analyzed Columns** list, click the option icon next to the indicator to open the dialog box where you can set options for the given indicator.

   Indicators settings dialog boxes differ according to the parameters specific for each indicator. For more information about different indicator parameters, see section *Indicator parameters*.
4. Set the parameters for the given indicator.
5. Click **Finish** to close the dialog box.
6. Click the save icon on the toolbar of the analysis editor.

5.5.1.4. How to set regular expressions and finalize the analysis

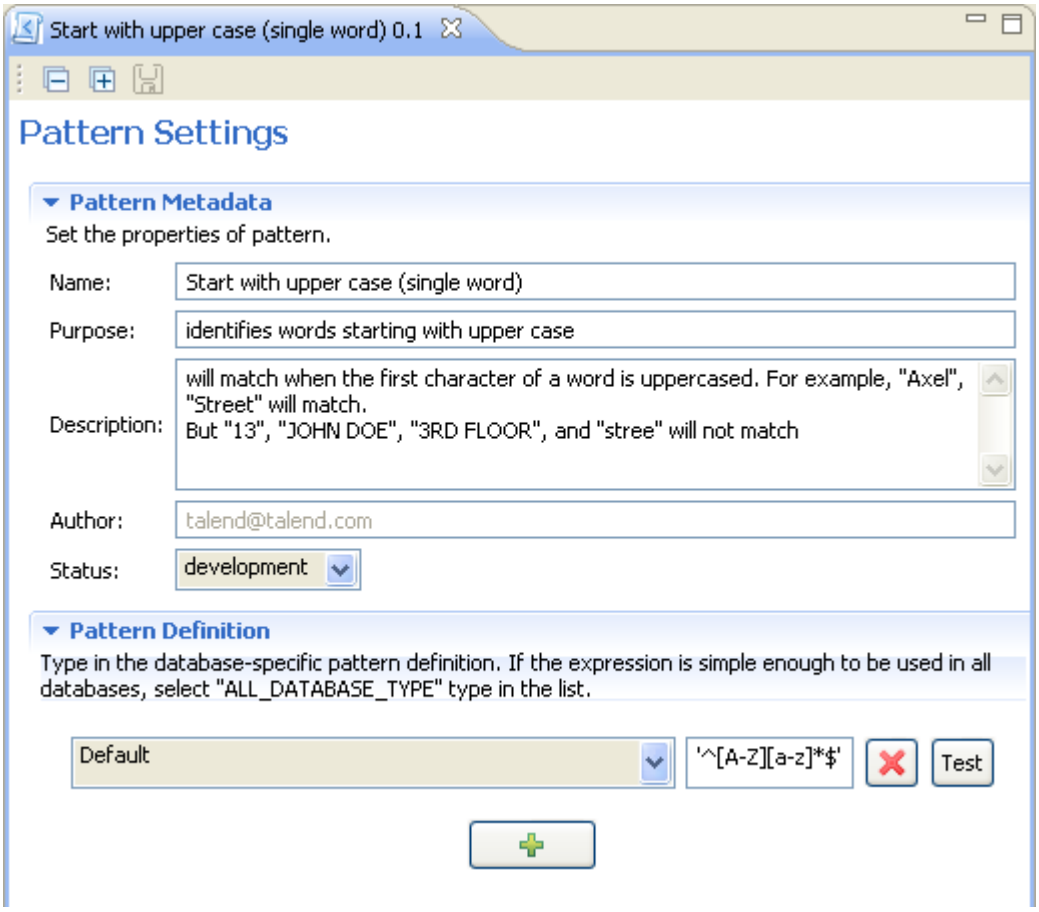
You can add one or more regular expressions to one or more of the analyzed columns.

Prerequisite(s): An analysis of a delimited file is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the columns to be analyzed*, section *How to set indicators for the column(s) to be analyzed* and section *How to set options for system indicators*.

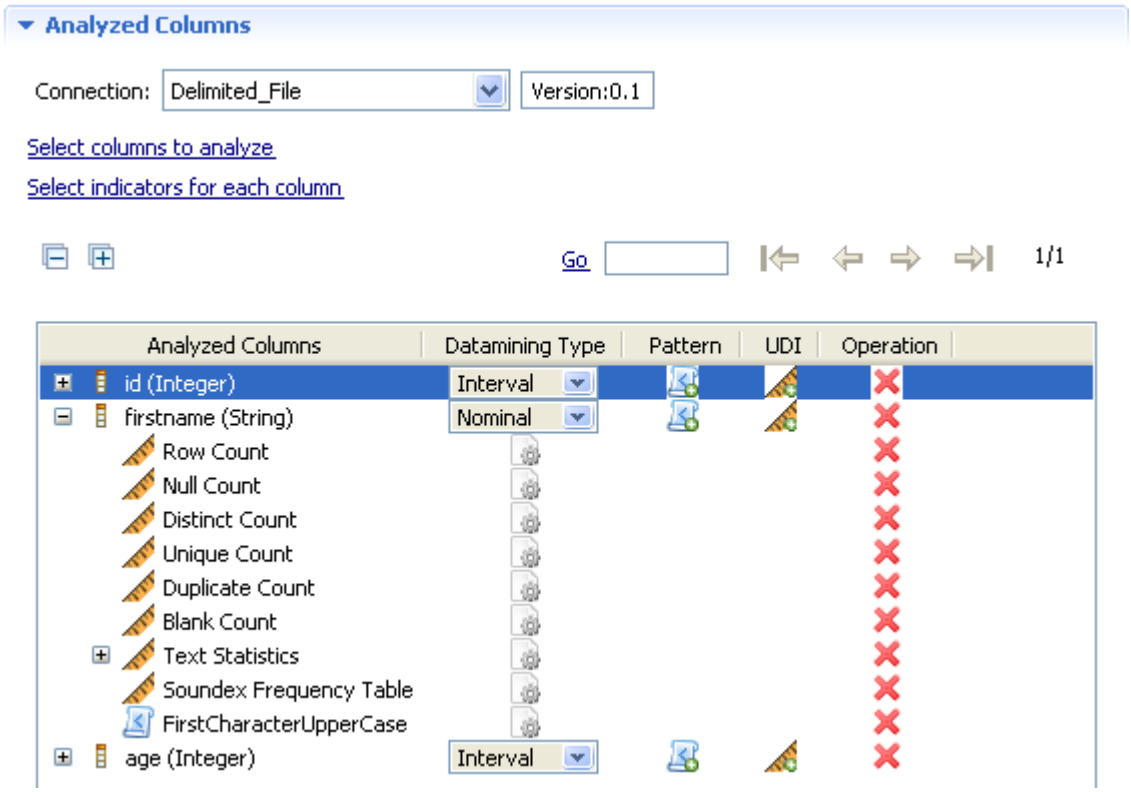
To set regular expressions to the analyzed columns, do the following:

1. Define the regular expression you want to add to the analyzed column. For further information on creating regular expressions, see section *How to create a new regular expression or SQL pattern*.

   In this example, the regular expression checks for all words that start with uppercase.



2. Add the regular expression to the analyzed column in the open analysis editor, the *firstname* column in this example. For further information, see section *How to add a regular expression or an SQL pattern to a column analysis*.



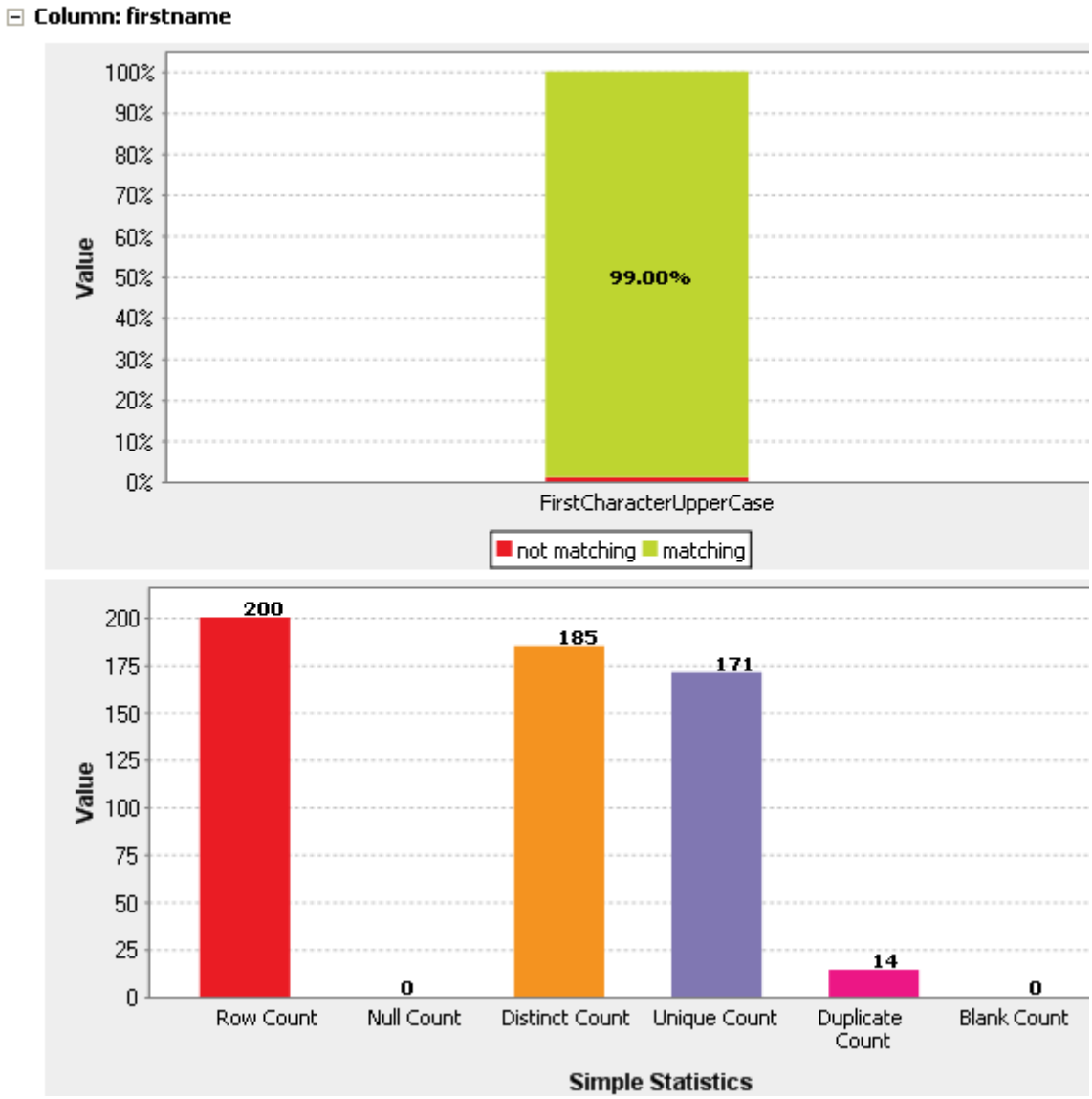
3. Click the save icon on the toolbar of the analysis editor and then press **F6** to execute the analysis.

   If the format of the file you are using has problems, you will have an error message to indicate which row causes the problem.

   The **Graphics** panel to the right of the analysis editor displays a group of graphic(s), each corresponding to one of the analyzed columns.

4. If you analyze more than one column, navigate through the different pages in the **Graphics** panel using the toolbar on the upper-right corner in order to view the different graphics associated with all analyzed columns.

   Below is a sample of the graphical results of one of the analyzed columns: *firstname*.



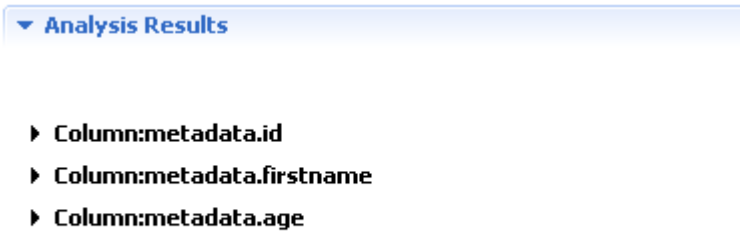
In order to view detail results of the analyzed columns, see section *How to access the detailed view of the analysis results*.

5.5.1.5. How to access the detailed view of the file analysis

Prerequisite(s): An analysis of a delimited file is defined and executed in the **Profiling** perspective in the studio. For more information, see section *Analyzing columns in a delimited file*.

To access a more detailed view of the analysis results, do the following:

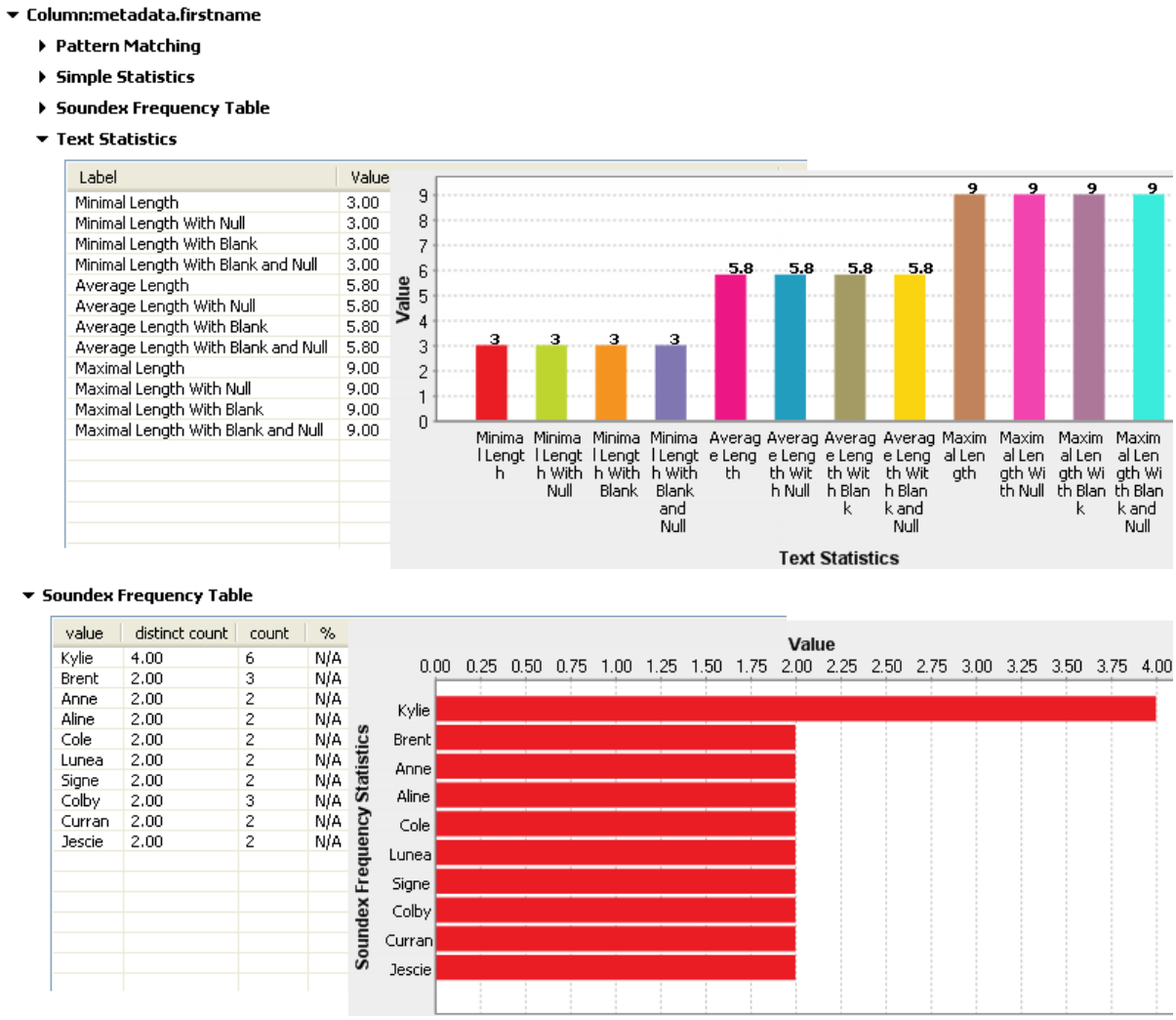
1. Click the **Analysis Results** tab at the bottom of the analysis editor to open the corresponding view.
2. Click **Analysis Result** and then the name of the analyzed column for which you want to display the detailed results.

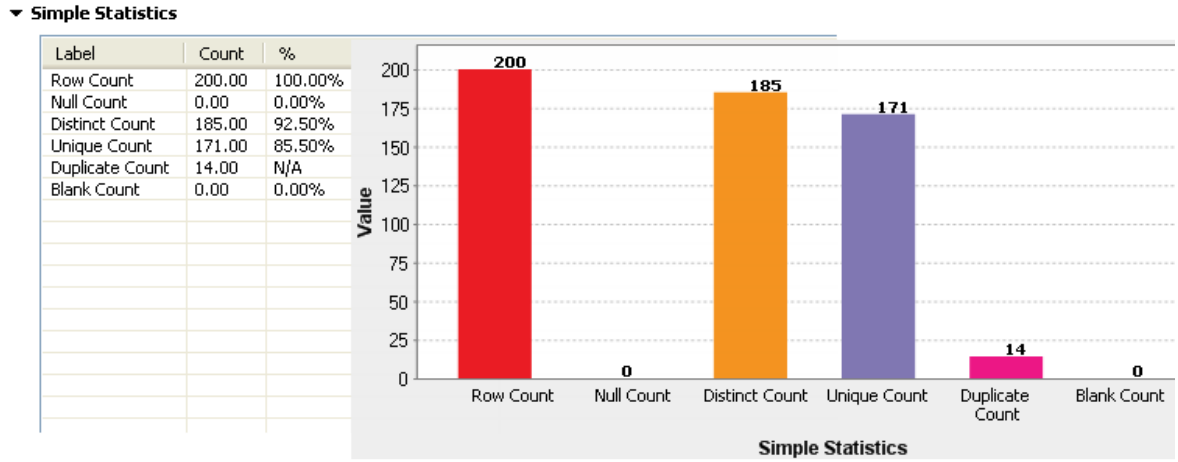


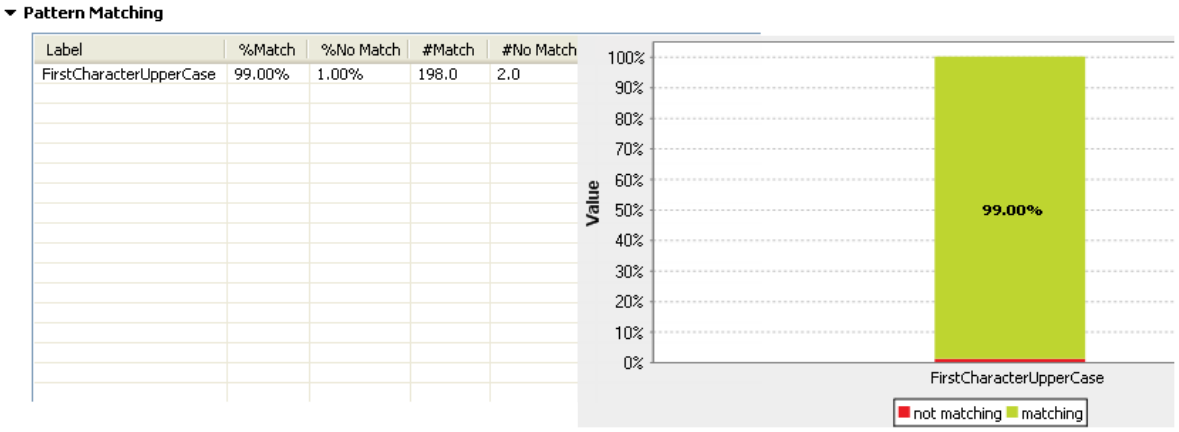
The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

The detailed analysis results view shows the generated graphics for the analyzed columns accompanied with tables that detail the statistic results.

Below are the tables that accompany the statistics graphics in the **Analysis Results** view for the analyzed *firstname* column in the procedure outlined in section *Analyzing columns in a delimited file*.







5.5.1.6. How to view and export the analyzed data in a file

After running your file analysis using the Java engine and from the **Analysis Results** view of the analysis editor, you can right-click any of the rows in the statistic result tables and access a view of the actual data.

Prerequisite(s): A file analysis has been created and executed.

To view and export the analyzed data, do the following:

1. At the bottom of the analysis editor, click the **Analysis Results** tab to open a detailed view of the analysis results.

2. Right-click a data row in the statistic results of any of the analyzed columns and select an option as the following:

Option	Operation
View rows	open a view on a list of all data rows in the analyzed column. For the *Duplicate Count* indicator, the **View rows** option will list all the rows that are duplicated. So if the duplicate count is 12 for example, this option will list 24 rows.
View values	open a view on a list of the actual data values of the analyzed column.

For Pattern Matching results, select an option as the following:

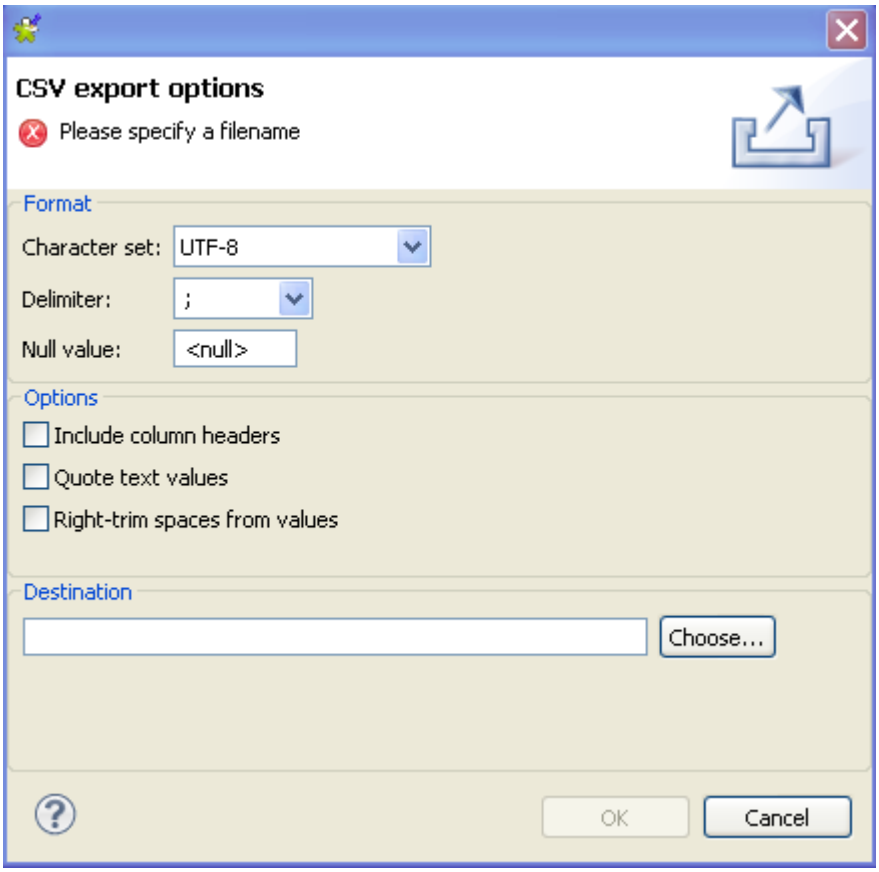
Option	Operation
View valid/invalid rows	open a view on a list of all valid/invalid rows measured against a pattern.

Option	Operation
View valid/invalid values	open a view on a list of all valid/invalid values measured against a pattern.

From this view, you can export the analyzed data into a csv file. To do that:

1. Click the icon in the top left corner of the view.

   A dialog box opens.



2. Click the **Choose...** button and browse to where you want to store the csv file and give it a name.
3. Click **OK** to close the dialog box.

   A csv file is created in the specified place holding all the analyzed data rows listed in the view.

5.5.1.7. How to analyze delimited data in shortcut procedures

You can profile data in a delimited file using a simplified way. All what you need to do is to start from the column name under **Metadata > FileDelimited** folders in the **DQ Repository** tree view.

For further information, see section *Creating table and columns analyses in shortcut procedures*.

5.5.2. Analyzing columns in an excel file

You can analyze data in an excel file and execute the created analyses using the Java engine.

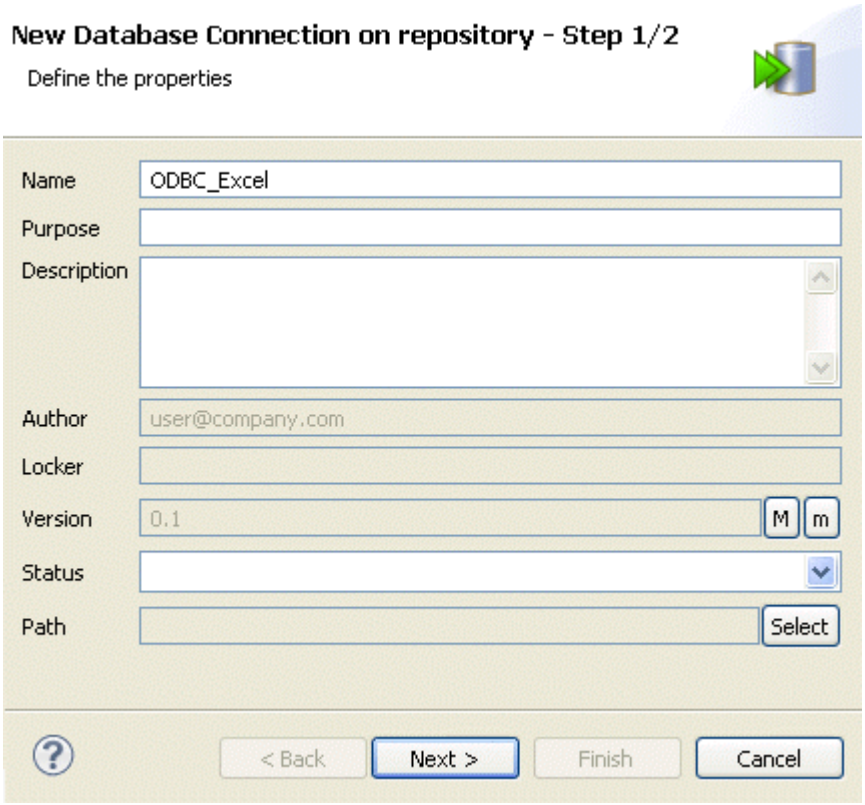
Profiling excel files is done via ODBC for the time being. In later releases, you will be able to analyze excel files directly as you do with delimited files.

Prerequisite(s): At least one connection to an excel file is set in the **Profiling** perspective of the studio. For further information, see section *How to connect to an Excel file*.

To set up an ODBC connection to a Data Source, do the following:

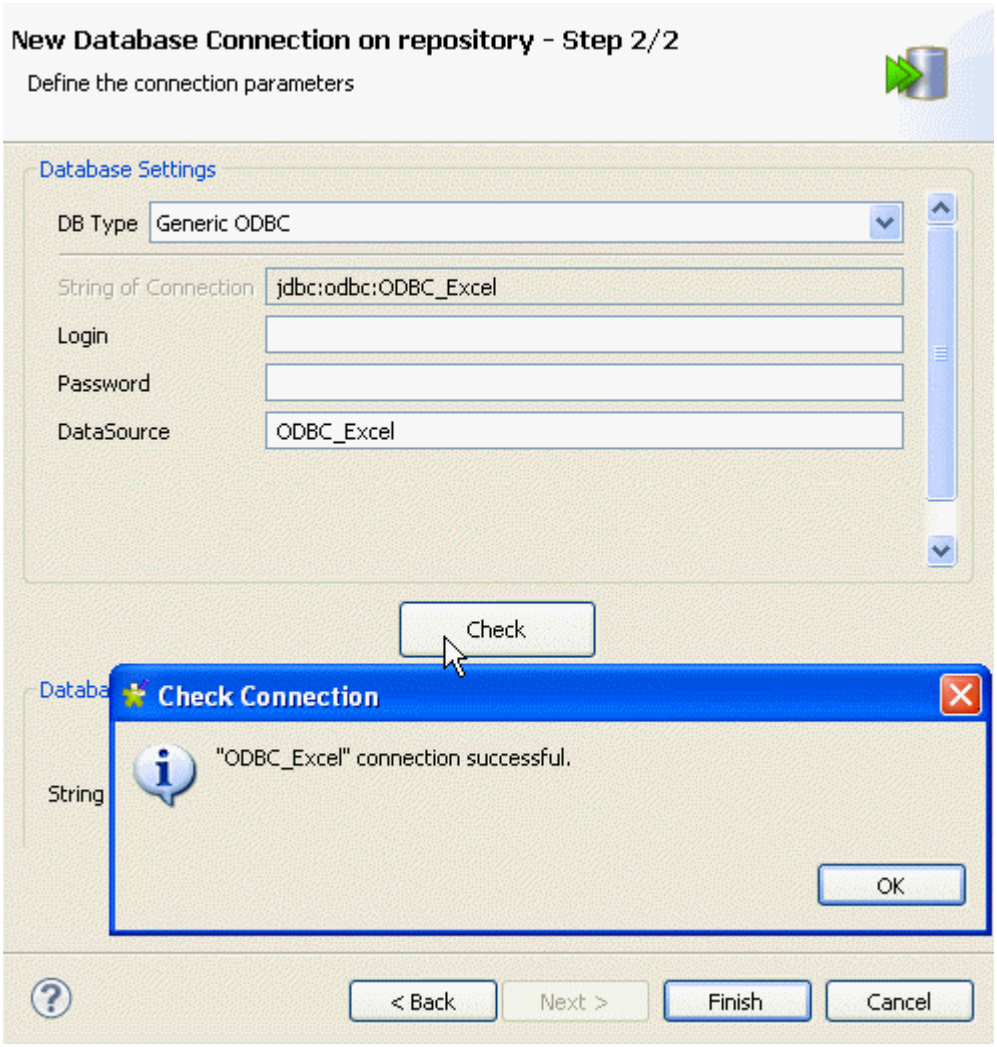
1. In the **DQ Repository** tree view, expand **Metadata**, and then right-click **DB connections**.

   The connection wizard is displayed.

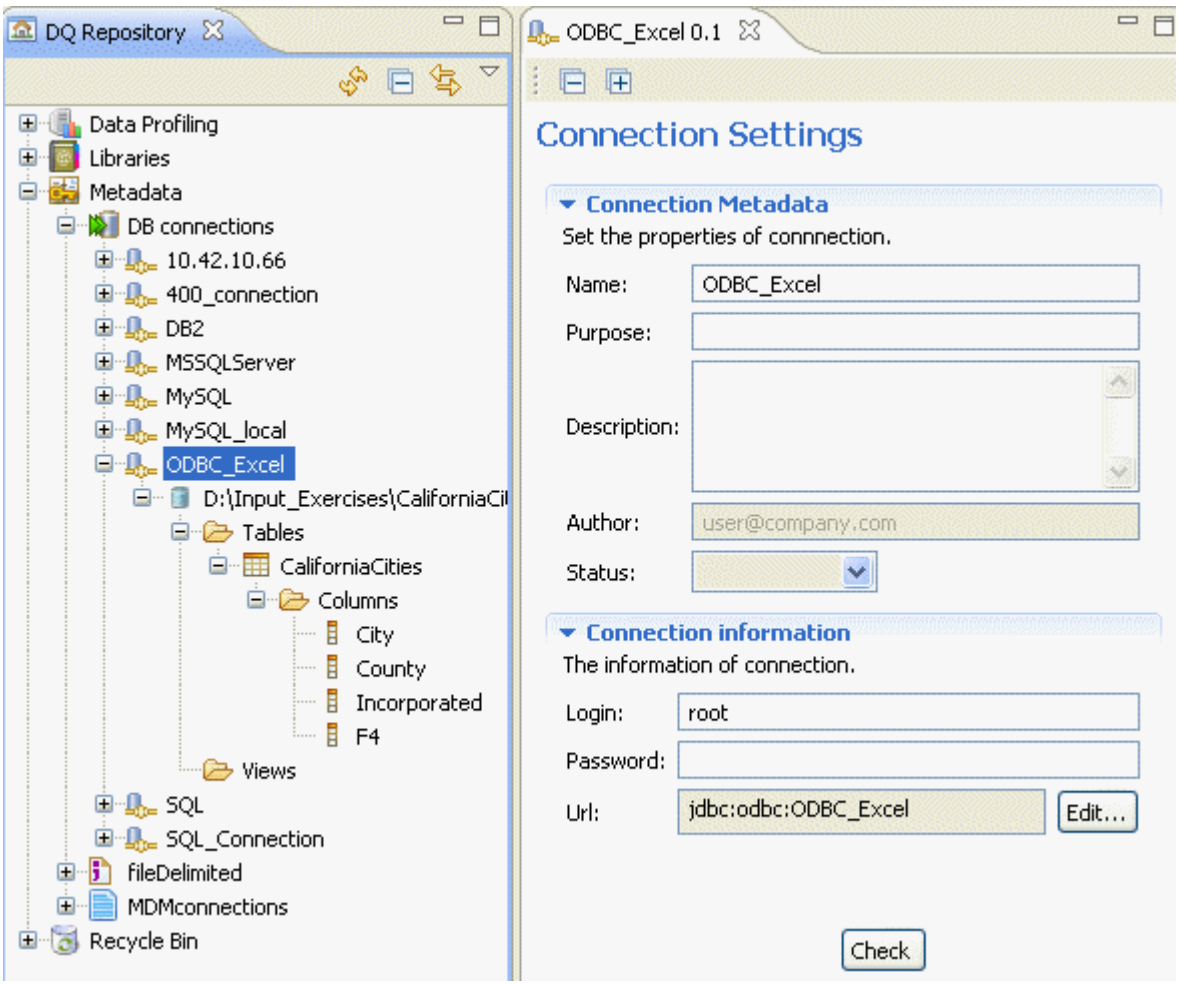


2. In the **Name** field, enter a name for the connection.

3. If required, fill in a purpose and a description for the connection, and then click **Next** to proceed to the next step.



4. From the **DB Type** list, select **Generic ODBC**.
5. In the **DataSource** field, enter the exact name of the Data Source you created in the previous procedure.
6. Click the **Check** button to display a confirmation message about the status of the connection.
7. If your connection is successful, click **OK** to close the message, and then click **Finish** to close the wizard.
8. The connection is listed under **DB connections** in the **DQ Repository** tree view and the connection editor opens in the Studio.



If you have difficulty retrieving the columns from the excel file, give the worksheet in the excel file the same name of the table. To do that, select the whole table in the excel file and then press Ctrl + F3 and modify the name.

You can now create a column analysis in the **Profiling** perspective of the studio to profile the columns in the excel file.

The procedures to analyze columns in an excel file are exactly the same as those for analyzing columns in a delimited file. For further information on analyzing columns in an excel files, see section *Analyzing columns in a delimited file*, section *How to access the detailed view of the analysis results* and section *Analyzing master data in shortcut procedures*.

Make sure to select the Java engine in the Analysis Parameter view in the analysis editor before executing the analysis of the excel columns, otherwise you will have an error message when running the analysis.

Chapter 6. Table analyses

This chapter provides all the information you need to perform table analyses on databases, delimited files or Master Data Management (MDM) servers.

It describes how to set up SQL business rules based on WHERE clauses and add them as indicators to database table analyses.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

6.1. Steps to analyze a table

You can examine the data available in single tables of a database and collect information and statistics about this data.

The sequence of profiling data in one or multiple tables may involve the following steps:

1. Defining one or more tables on which to carry out data profiling processes that will define the content, structure and quality of the data included in the table(s).
2. Creating SQL business rules based on WHERE clauses and add them as indicators to table analyses.
3. Creating column functional dependencies analyses to detect anomalies in the column dependencies of the defined table(s) through defining columns as either “determinant” or “dependent”.

section *Analyzing tables in databases* explains in detail the different options to analyze a table.

6.2. Analyzing tables in databases

Table analyses can range from simple table analyses to table analyses that uses SQL business rules or table analyses that detect anomalies in the table columns.

Using the studio, you can better explore the quality of data in a database table through either:

- Creating a simple table analysis through analyzing all columns in the table using patterns. For more information, see section *Creating a simple table analysis: the analysis of a set of columns*.
- Adding data quality rules as indicators to table analysis. For more information, see section *Creating a table analysis with SQL business rules*.
- Detecting anomalies in column dependencies. For more information, see section *Detecting anomalies in the table columns: column functional dependency analysis*.

The sections below explain in detail all types of analysis that can be executed against tables.

6.2.1. Creating a simple table analysis: the analysis of a set of columns

You can analyze the content of a set of columns. This set can represent only some of the columns in the defined table or the table as a whole.

The analysis of a set of columns focuses on a column set (full records) and not on separate columns as it is the case with the column analysis. The statistics presented in the analysis results (row count, distinct count, unique count and duplicate count) are measured against the values across all the data set and thus do not analyze the values separately within each column.

With the Java engine, you may also apply patterns on each column and the result of the analysis will give the number of records matching all the selected patterns together. For further information, see section *How to add patterns to the analyzed columns*.

When you use the Java engine to run a column set analysis on big sets or on data with many problems, it is advisable to define a maximum memory size threshold to execute the analysis as you may end up with a Java heap error. For more information, see section *Defining the maximum memory size threshold*.

6.2.1.1. How to create an analysis of a set of columns using patterns

This type of analysis provides simple statistics on the full records of the analyzed column set and not on the values within each column separately. For more information about simple statistic indicators, see section *Simple statistics*.

With this analysis, you can use patterns to validate the full records against all patterns and have a single-bar result chart that shows the number of the rows that match “all” the patterns..

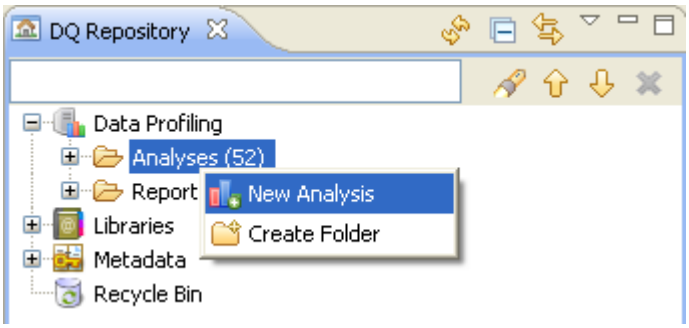
How to define the set of columns to be analyzed

Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

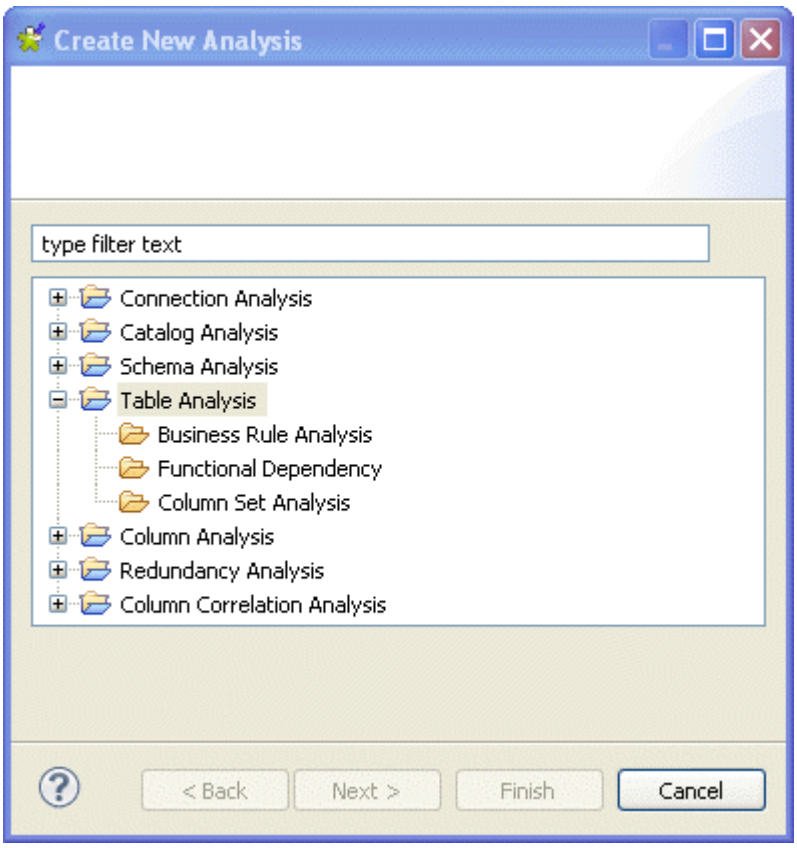
To define the set of columns to analyzed, do the following:

Defining the analysis

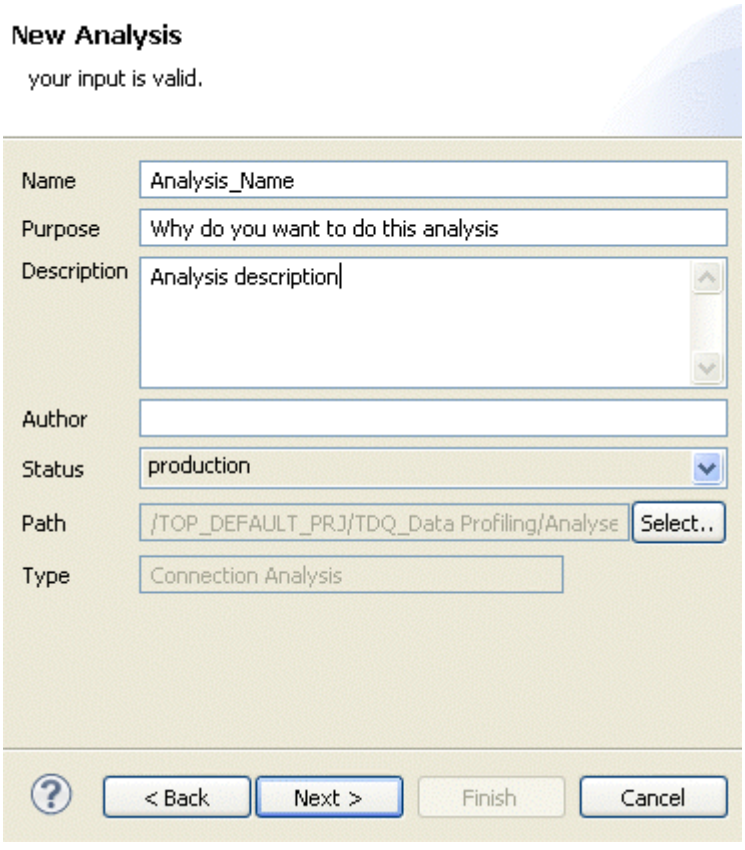
1. In the **DQ Repository** tree view, expand **Data Profiling**.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



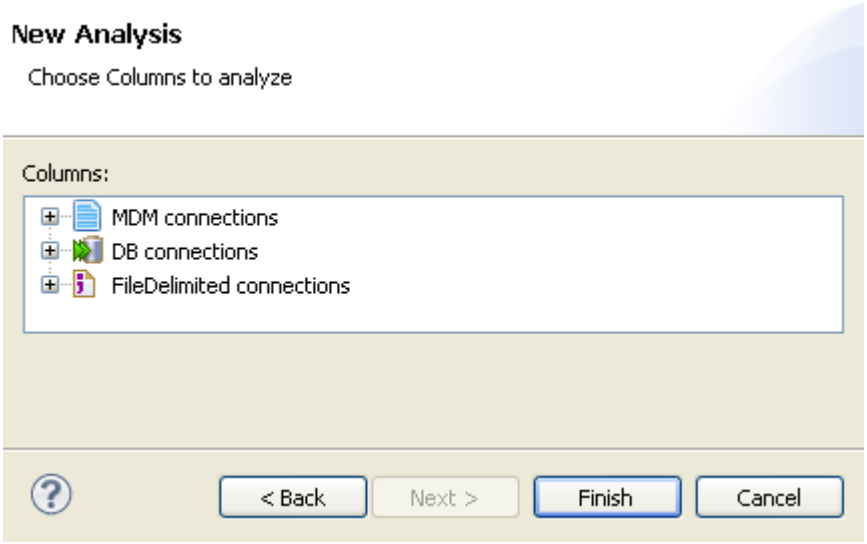
3. Expand the **Table Analysis** node and then click **Column Set Analysis**.
4. Click the **Next** button.



5. In the **Name** field, enter a name for the current analysis.

Space is not acceptable when typing in the analysis name in this field.

6. Set column analysis metadata (purpose, description and author name) in the corresponding fields and then click **Next**.

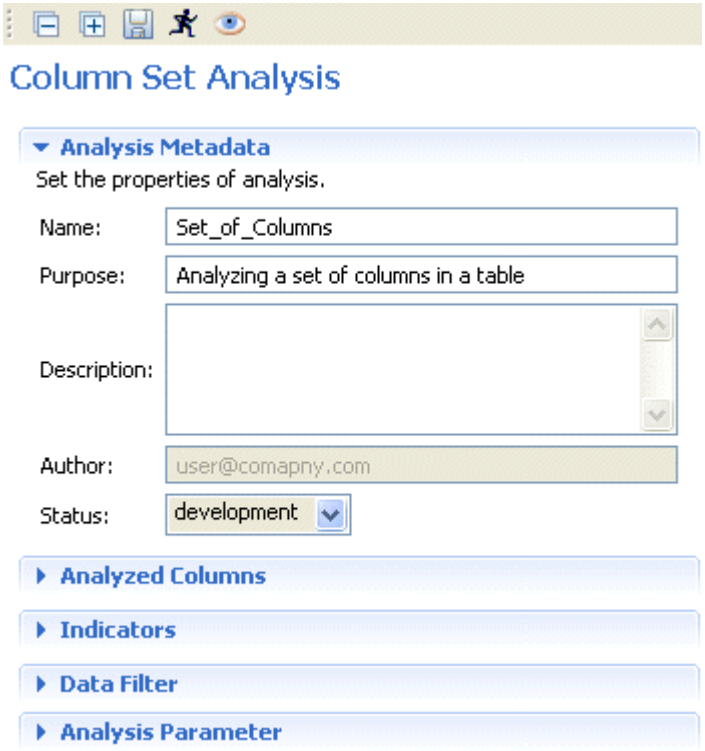


Selecting the set of columns you want to analyze

1. Expand **DB connections**.

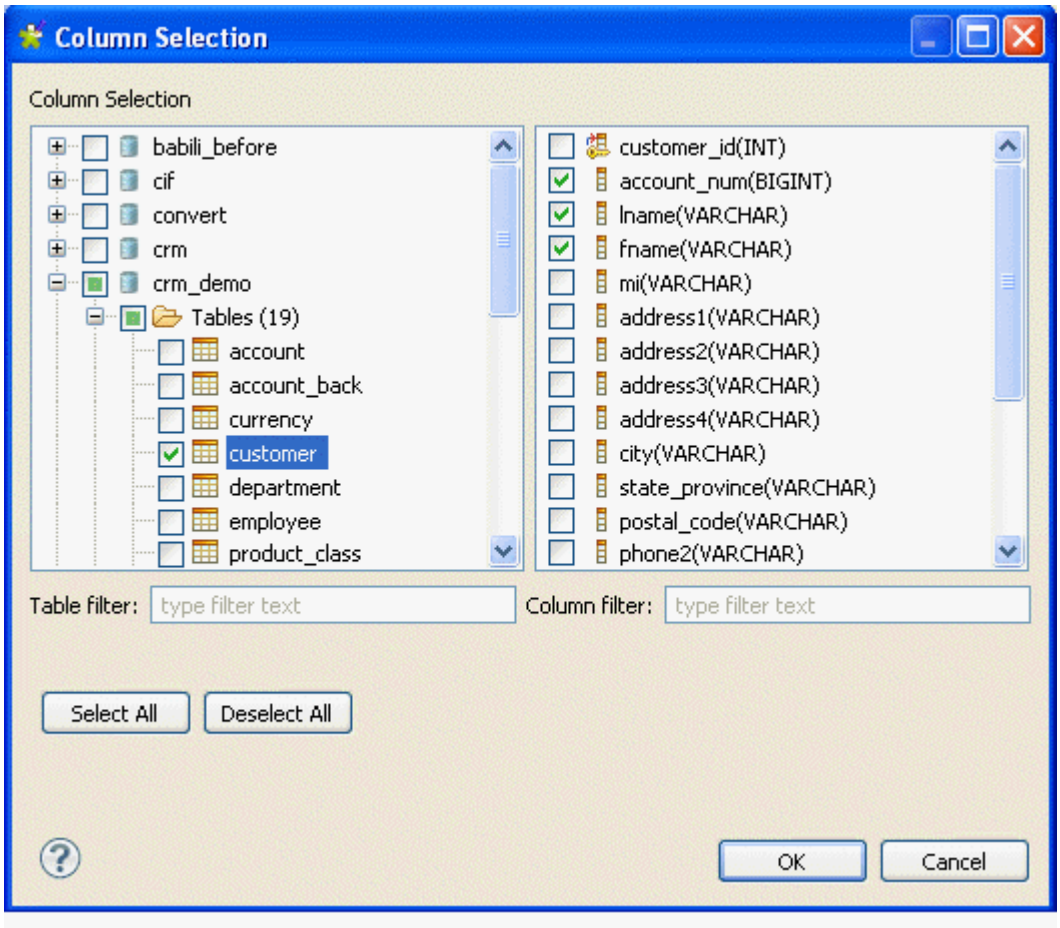
2. In the desired database, browse to the columns you want to analyze, select them and then click **Finish** to close this **[New analysis]** wizard.

   A folder for the newly created analysis is listed under **Analysis** in the **DQ Repository** tree view, and the analysis editor opens with the defined analysis metadata.



The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

3. Click the **Analyzed Columns** tab to open the corresponding view. Click the **Select columns to analyze** link to open a dialog box where you can modify your table or column selection.



If you select to connect to a database that is not supported in the studio (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on the Java engine, see section *Using the Java or the SQL engine*.

4. Either:

   - expand the **DB Connections** folder and browse through the catalog/schemas to reach the table holding the columns you want to analyze, or,
   - filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

   As this analysis retrieves as many rows as the number of distinct rows in order to compute the statistics, it is advised to avoid selecting a primary key column.

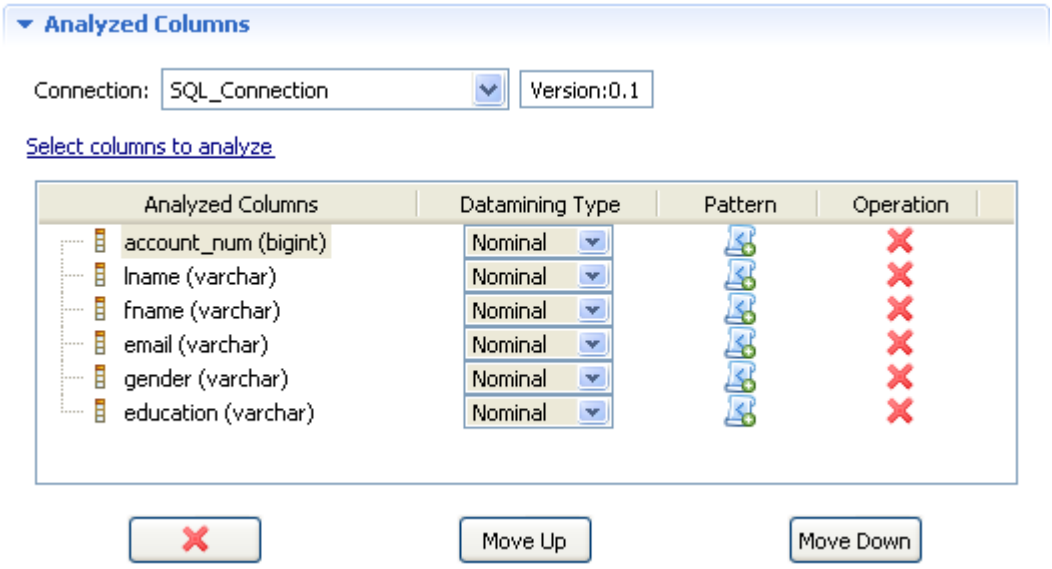
   In this example, you want to analyze a set of six columns in the *customer* table: account number (*account_num*), education (*education*), email (*email*), first name (*fname*), second name (*Iname*) and gender (*gender*). you want to identify the number of rows, the number of distinct and unique values and the number of duplicates.

5. Click the table name to list all its columns in the right-hand panel of the **[Column Selection]** dialog box.

6. In the column list, select the check boxes of the column(s) you want to analyze and click **OK**.

Select the check boxes of all the columns if you want to get simple statistics on the whole table.

The selected columns is displayed in the **Analyzed Column** view of the analysis editor.



7. If required, select to connect to a different database by selecting a different connection from the **Connection** box. This box lists all the connections created in the Studio with the corresponding database names.

   If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

8. If required, right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**. The selected column is automatically located under the corresponding connection in the tree view.

9. Use the delete, move up or move down buttons to manage the analyzed columns when necessary.

How to add patterns to the analyzed columns

You can add patterns to one or more of the analyzed columns to validate the full record (all columns) against all the patterns, and not to validate each column against a specific pattern as it is the case with the column analysis. The results chart is a single bar chart for the totality of the used patterns. This chart shows the number of the rows that match “all” the patterns.

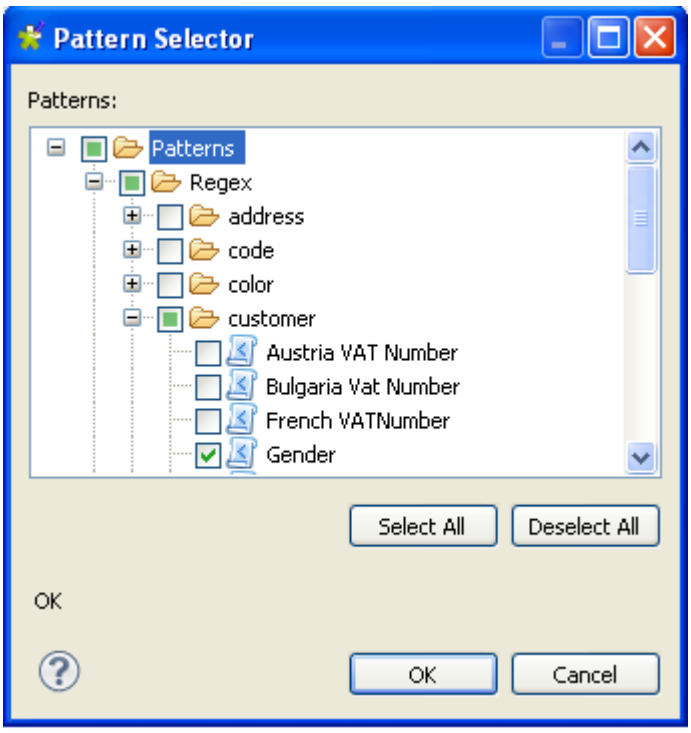
Before being able to use a specific pattern with a set of columns analysis, you must manually set the pattern definition for Java in the pattern settings, if it does not already exist. Otherwise, a warning message opens prompting you to set the definition of the Java regular expression.

Prerequisite(s): An analysis of a set of columns is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the set of columns to be analyzed*.

To add patterns to the analysis of a set of columns, do the following:

1. Click the icon next to each of the columns you want to validate against a specific pattern.

   The **[Pattern Selector]** dialog box is displayed.



You can add only regular expressions to the analyzed columns.

You can drop the regular expression directly from the **Patterns** folder in the **DQ Repository** tree view directly to the column name in the column analysis editor.

If no Java expression exists for the pattern you want to add, a warning message opens prompting you to add the pattern definition for Java. Click Yes to open the pattern editor and add the Java regular expression, then proceed to add the pattern to the analyzed columns.

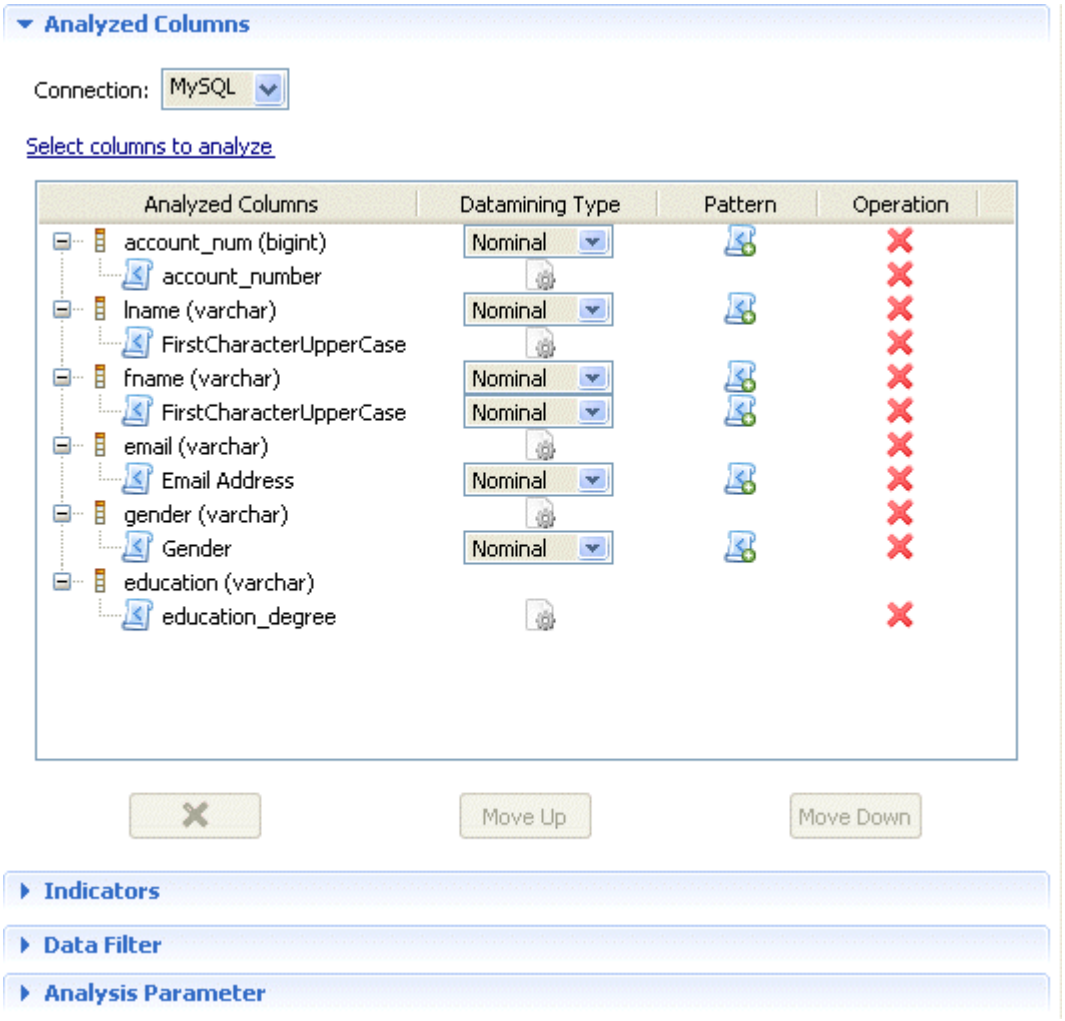
In this example, you want to add a corresponding pattern to each of the analyzed columns to validate data in these columns against the selected patterns. The result chart will show the percentage of the matching/non-matching values, the values that respect the totality of the used patterns.

2. In the **[Pattern Selector]** dialog box, expand **Patterns** and browse to the regular expression you want to add to the selected column.

3. Select the check box(es) of the expression(s) you want to add to the selected column.

4. Click **OK**.

   The added regular expression(s) are displayed under the analyzed column(s) in the **Analyzed Columns** list, and the `All Match` indicator is displayed in the **Indicators** list in the **Indicators** view.

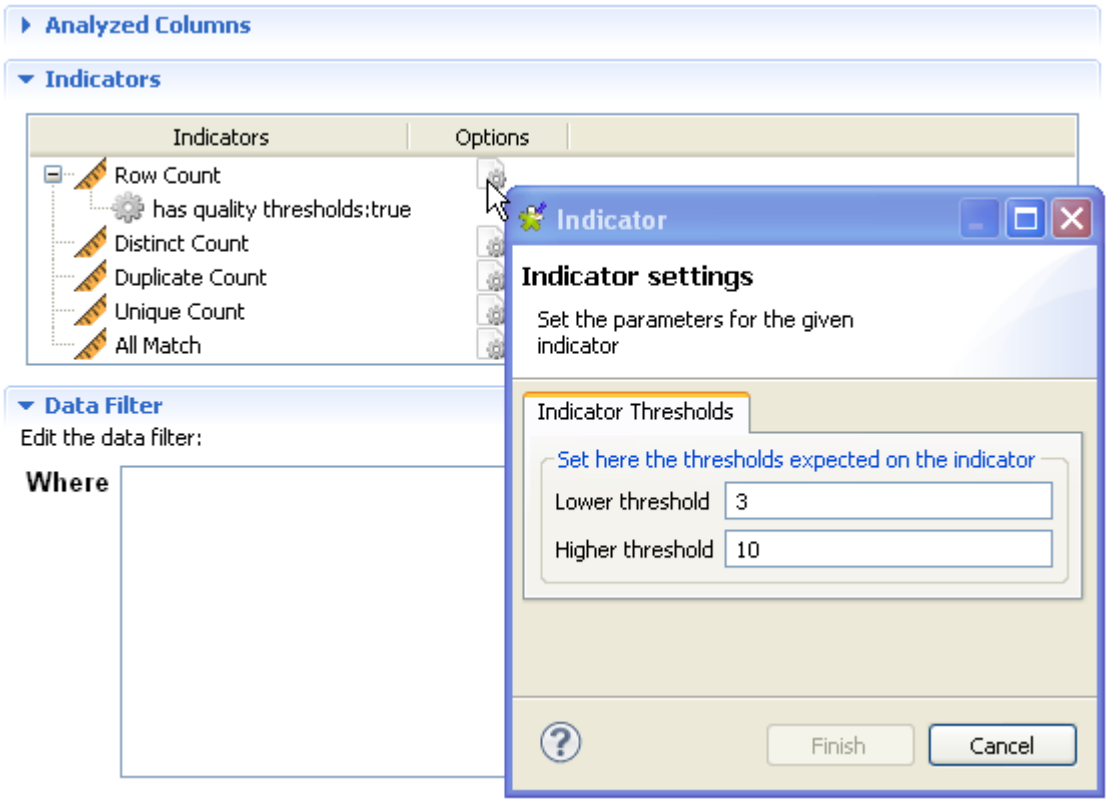


How to finalize and execute the analysis of a set of columns

What is left before executing this set of columns analysis is to define indicators, data filter and analysis parameters.

Prerequisite(s): A column set analysis has already been defined in the **Profiling** perspective of the studio. For further information, see section *How to define the set of columns to be analyzed* and section *How to add patterns to the analyzed columns*.

1. Click **Indicators** in the analysis editor to open the corresponding view.



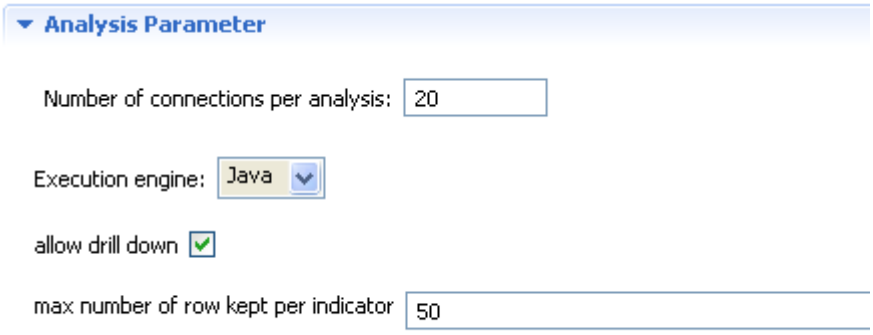
The indicators representing the simple statistics are by-default attached to this type of analysis. For further information about the indicators for simple statistics, see section section *Simple statistics*.

2. Click the option icon to open a dialog box where you can set options for each indicator according to your needs.

   For more information about indicators management, see section *Indicators*.

3. Click **Data Filter** in the analysis editor to open its view and filter data through SQL “WHERE” clauses according to your needs.

4. Click **Analysis Parameters** and:



   - In the **Number of connections per analysis** field, set the number of concurrent connections allowed per analysis to the selected database connection.

     You can set this number according to the database available resources, that is the number of concurrent connections each database can support.

   - From the **Execution engine** list, select the engine, Java or SQL, you want to use to execute the analysis.

If you select the **Java** engine and then select the **Allow drill down** check box in the **Analysis parameters** view, you can store locally the analyzed data and thus access it in the **Analysis Results** > **Data** view. You can use the **Max number of rows kept per indicator** field to decide the number of the data rows you want to make accessible.

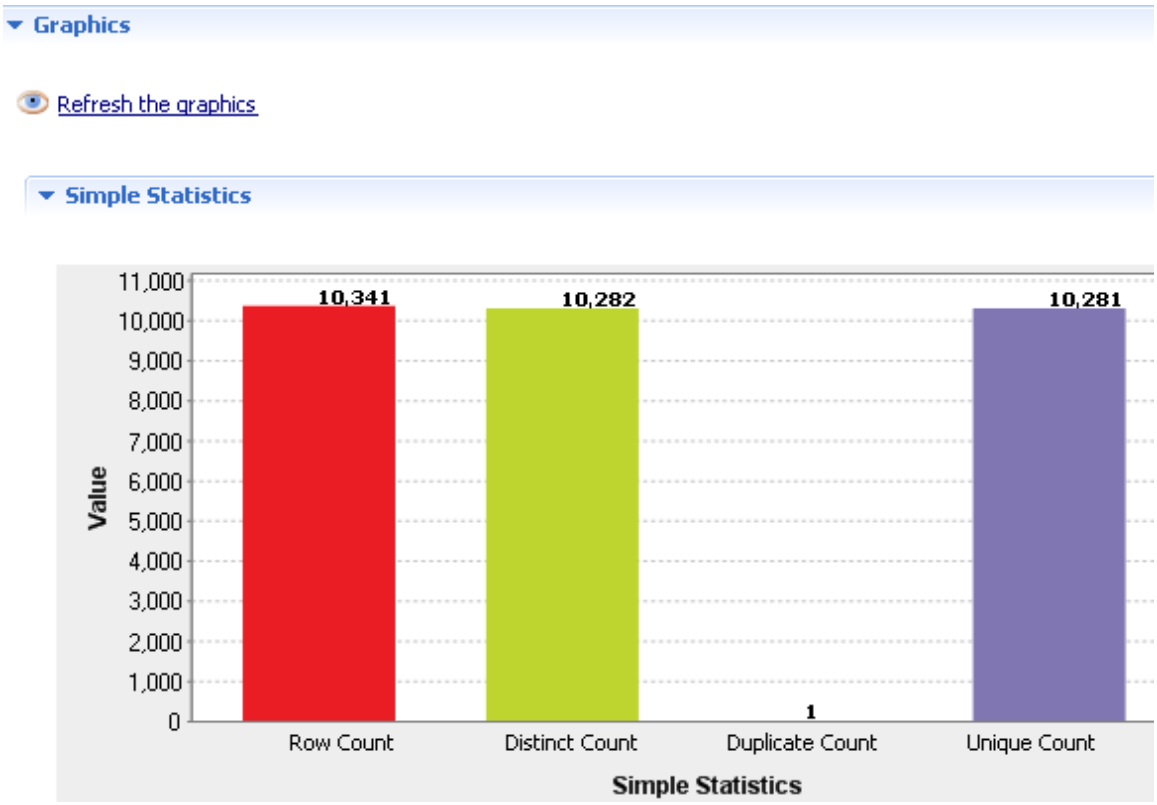
For further information, see section *How to access the detailed result view*.

If you select the **SQL** engine, select the **Store data** check box if you want to store locally the list of all analyzed rows and thus access it in the **Analysis Results** > **Data** view. For further information, see section *How to access the detailed result view*.

If the data you are analyzing is very big, it is advisable to leave this check box unchecked in order to have only the analysis results without storing analyzed data at the end of the analysis computation.

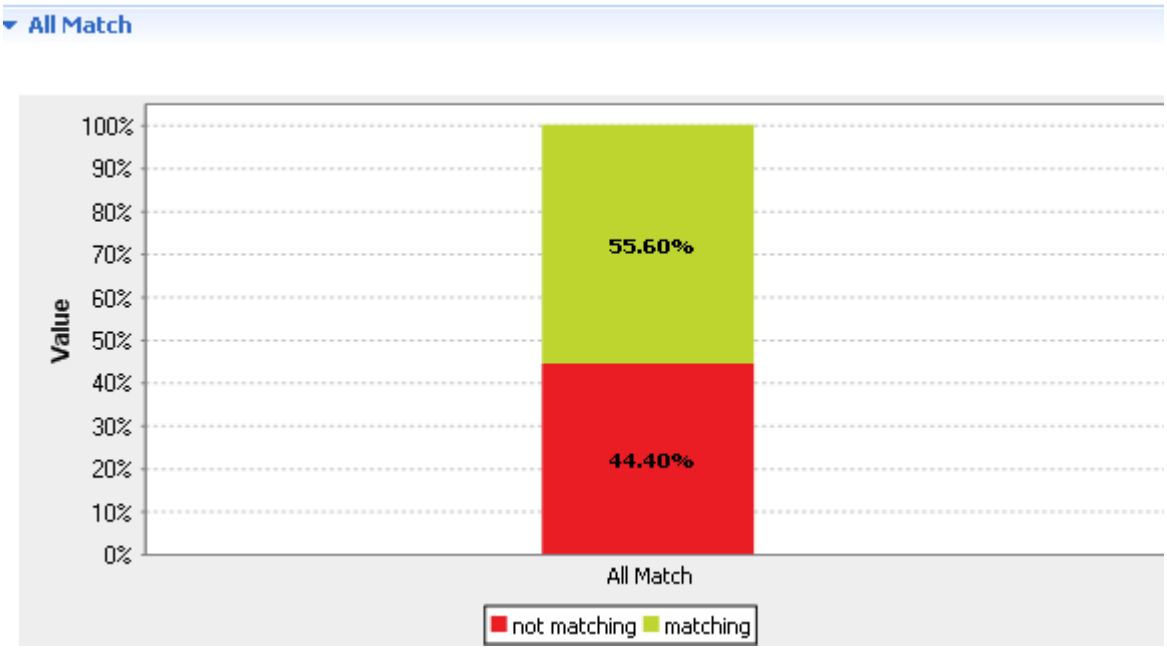
5. Click the save icon on top of the analysis editor and then press **F6** to execute the analysis.

The graphical result of the set of columns analysis is displayed in the **Graphics** panel to the right of the analysis editor.



This graphical result provides the simple statistics on the full records of the analyzed column set and not on the values within each column separately.

When you use patterns to match the content of the set of columns, another graphic is displayed to illustrates the match and non-match results against the totality of the used patterns.



How to access the detailed result view

Prerequisite(s): An analysis of a set of columns is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the set of columns to be analyzed* and section *How to add patterns to the analyzed columns*.

To access a more detailed view of the analysis results:

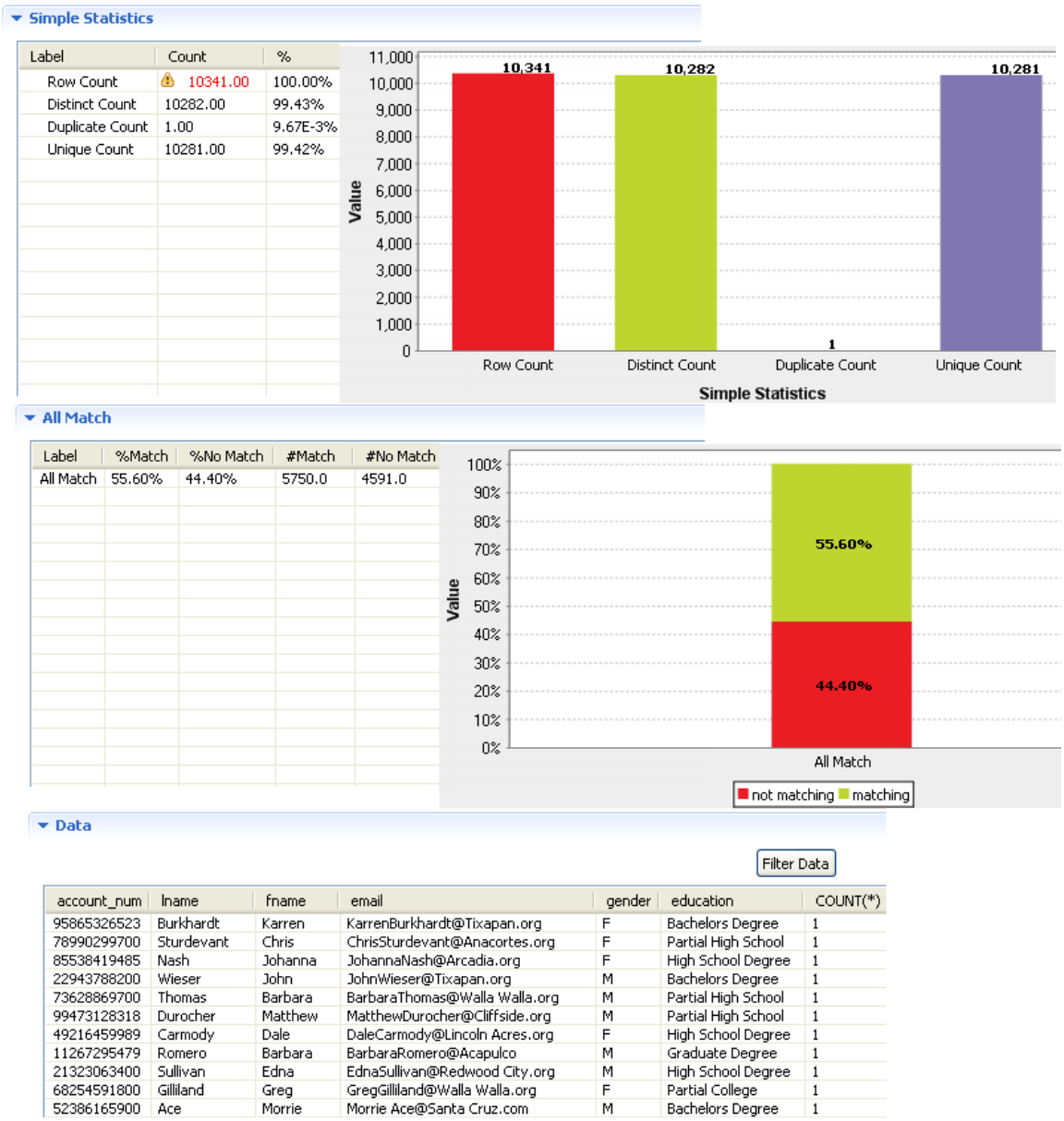
1. Click the **Analysis Results** tab at the bottom of the analysis editor.

   The corresponding view is displayed. Here you can read the analysis results in a table that accompanies the **Simple Statistics** and **All Match** graphics.

   The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

2. To have a view of the actual analyzed data, click **Data** in the **Analysis Results** view.

   In order to have the analyzed data stored in this view, you must select the Store data check box in the Analysis Parameter view. For further information, see section *How to finalize and execute the analysis of a set of columns*.



You can filter analyzed data according to any of the used patterns. For further information, see section *How to filter data against patterns*.

How to filter data against patterns

After analyzing a set of columns against a group of patterns and having the results of the rows that match or do not match “all” the patterns, you can filter the valid/invalid data according to the used patterns.

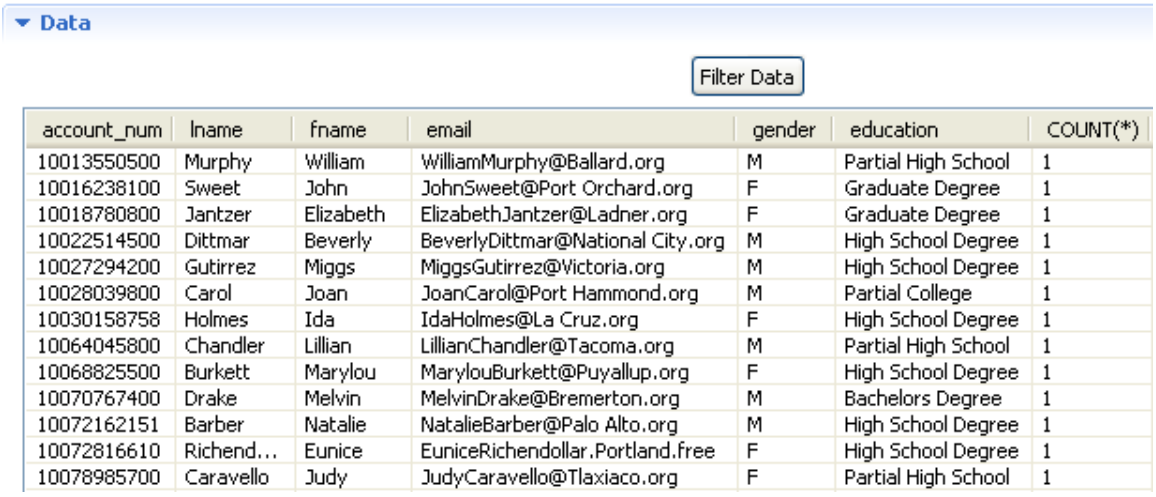
Prerequisite(s): An analysis of a set of columns is open in the analysis editor in the **Profiling** perspective of the studio. For more information, see section *How to define the set of columns to be analyzed* and section *How to add patterns to the analyzed columns*.

To filter data resulted from the analysis of a set of columns, do the following:

1. In the analysis editor, click the **Analysis Results** tab at the bottom of the editor to open the detailed result view.

The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

2. Click **Data** to open the corresponding table.

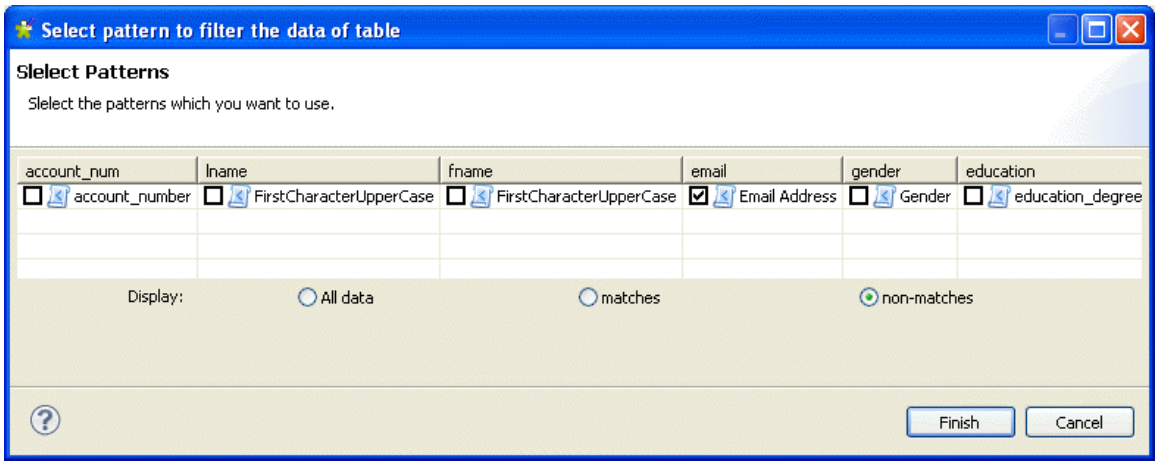


account_num	lname	fname	email	gender	education	COUNT(*)
10013550500	Murphy	William	WilliamMurphy@Ballard.org	M	Partial High School	1
10016238100	Sweet	John	JohnSweet@Port Orchard.org	F	Graduate Degree	1
10018780800	Jantzer	Elizabeth	ElizabethJantzer@Ladner.org	F	Graduate Degree	1
10022514500	Dittmar	Beverly	BeverlyDittmar@National City.org	M	High School Degree	1
10027294200	Gutirrez	Miggs	MiggsGutirrez@Victoria.org	M	High School Degree	1
10028039800	Carol	Joan	JoanCarol@Port Hammond.org	M	Partial College	1
10030158758	Holmes	Ida	IdaHolmes@La Cruz.org	F	High School Degree	1
10064045800	Chandler	Lillian	LillianChandler@Tacoma.org	M	Partial High School	1
10068825500	Burkett	Marylou	MarylouBurkett@Puyallup.org	F	High School Degree	1
10070767400	Drake	Melvin	MelvinDrake@Bremerton.org	M	Bachelors Degree	1
10072162151	Barber	Natalie	NatalieBarber@Palo Alto.org	M	High School Degree	1
10072816610	Richend...	Eunice	EuniceRichendollar.Portland.free	F	High School Degree	1
10078985700	Caravello	Judy	JudyCaravello@Tlaxiaco.org	F	Partial High School	1

This table lists the actual analyzed data in the analyzed columns.

3. Click **Filter Data** on top of the table.

A dialog box is displayed listing all the patterns used in the column set analysis.

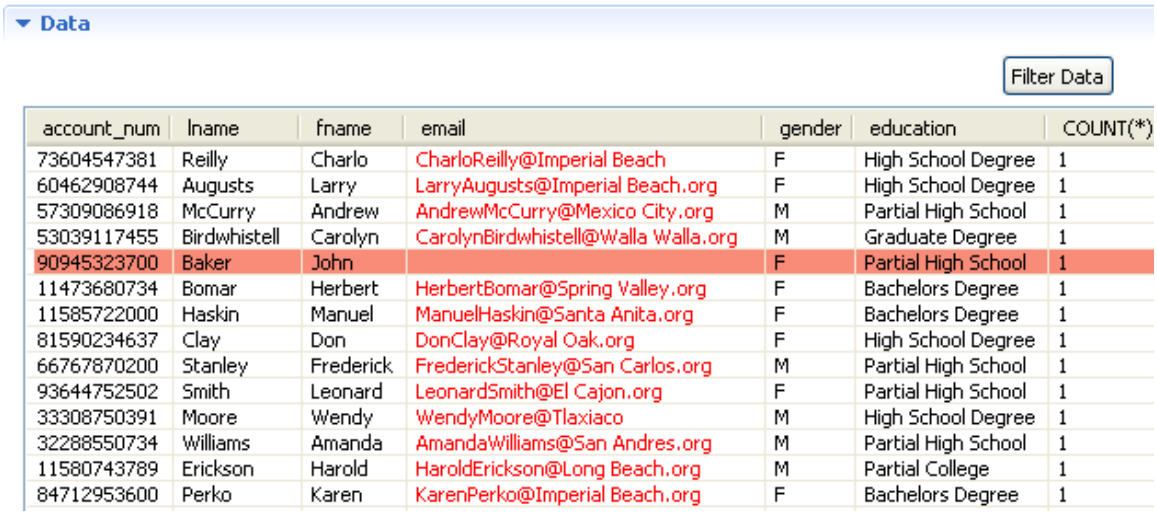


4. Select the check box(es) of the pattern(s) according to which you want to filter the data, and then select a display option according to your needs.

5. Select **All data** to show all analyzed data, or **matches** to show only the data that matches the pattern, or **non-matches** to show the data that does not match the selected pattern(s).

6. Click **Finish** to close the dialog box.

In this example, data is filtered against the *Email Address* pattern, and only the data that does not match is displayed.



account_num	lname	fname	email	gender	education	COUNT(*)
73604547381	Reilly	Charlo	CharloReilly@Imperial Beach	F	High School Degree	1
60462908744	Augusts	Larry	LarryAugusts@Imperial Beach.org	F	High School Degree	1
57309086918	McCurry	Andrew	AndrewMcCurry@Mexico City.org	M	Partial High School	1
53039117455	Birdwhistell	Carolyn	CarolynBirdwhistell@Walla Walla.org	M	Graduate Degree	1
90945323700	Baker	John		F	Partial High School	1
11473680734	Bomar	Herbert	HerbertBomar@Spring Valley.org	F	Bachelors Degree	1
11585722000	Haskin	Manuel	ManuelHaskin@Santa Anita.org	F	Bachelors Degree	1
81590234637	Clay	Don	DonClay@Royal Oak.org	F	High School Degree	1
66767870200	Stanley	Frederick	FrederickStanley@San Carlos.org	M	Partial High School	1
93644752502	Smith	Leonard	LeonardSmith@El Cajon.org	F	Partial High School	1
33308750391	Moore	Wendy	WendyMoore@Tlaxiaco	M	High School Degree	1
32288550734	Williams	Amanda	AmandaWilliams@San Andres.org	M	Partial High School	1
11580743789	Erickson	Harold	HaroldErickson@Long Beach.org	M	Partial College	1
84712953600	Perko	Karen	KarenPerko@Imperial Beach.org	F	Bachelors Degree	1

All email addresses that do not match the selected pattern appear in red. Any data row that has a missing value appear with a red background.

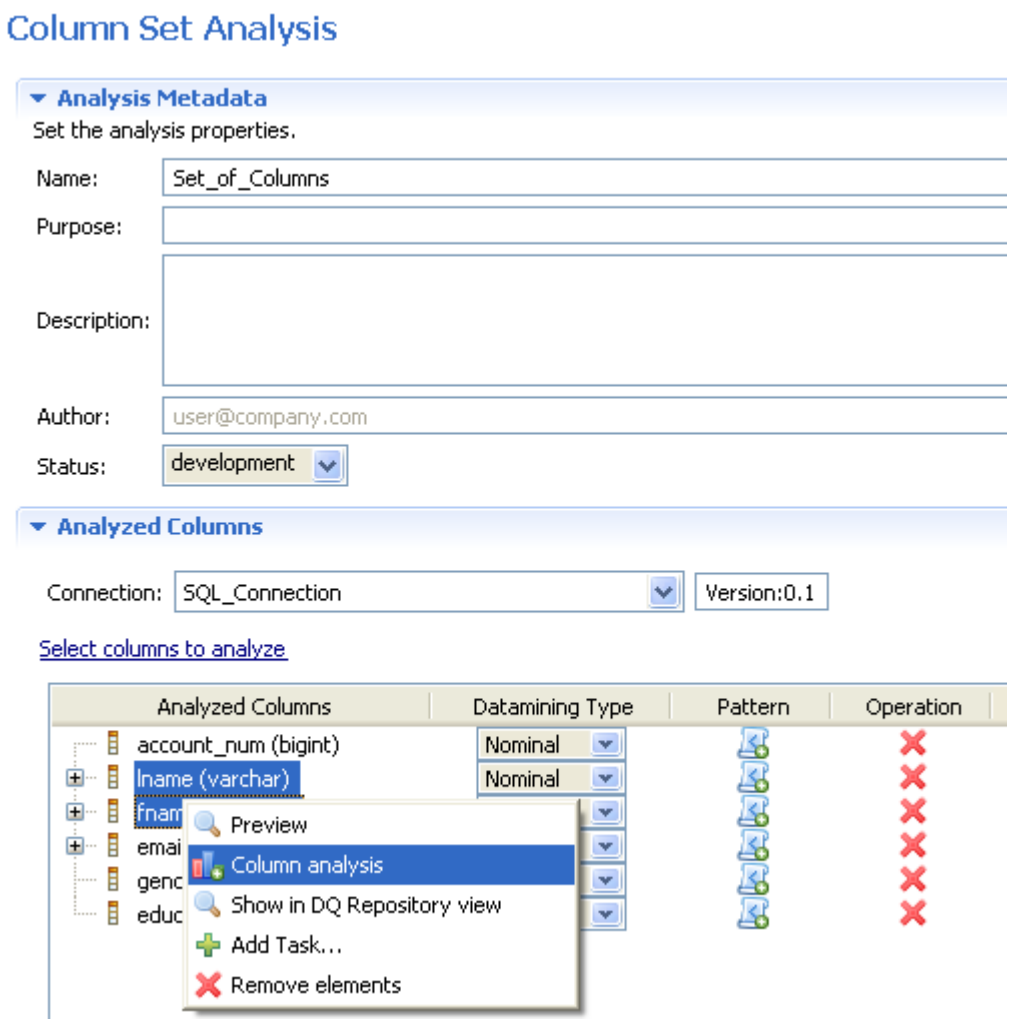
6.2.1.2. How to create a column analysis from a simple table analysis

You can create a column analysis on one or more columns defined in a simple table analysis (column set analysis).

Prerequisite(s): A simple table analysis is defined in the analysis editor in the **Profiling** perspective of the studio.

To create a column analysis on one or more columns defined in a simple table analysis, do the following:

1. Open the simple table analysis.
2. In the **Analyzed Columns** view, right-click the column(s) you want to create a column analysis on.



3. Select **Column analysis** from the contextual menu.

   The **[New Analysis]** wizard opens.

4. In the **Name** field, enter a name for the new column analysis and then click **Next** to proceed to the next step.

   The analysis editor opens with the defined metadata and a folder for the newly created analysis is listed under the **Analyses** folder in the **DQ Repository** tree view.

5. Follow the steps outlined in section *Analyzing columns in a database* to continue creating the column analysis.

6.2.2. Creating a table analysis with SQL business rules

You can set up SQL business rules based on WHERE clauses and add them as indicators to table analyses. You can as well define expected thresholds on the SQL business rule indicator's value. The range defined is used for measuring the quality of the data in the selected table.

It is also possible to create an analysis with SQL business rules on views in a database. The procedure is exactly the same as that for tables. For more information, see section *How to create a table analysis with an SQL business rule with a join condition*.

When you use the Java engine to run a column set analysis on big sets or on data with many problems, it is advisable to define a maximum memory size threshold to execute the analysis as you may end up with a Java heap error. For more information, see section *Defining the maximum memory size threshold*.

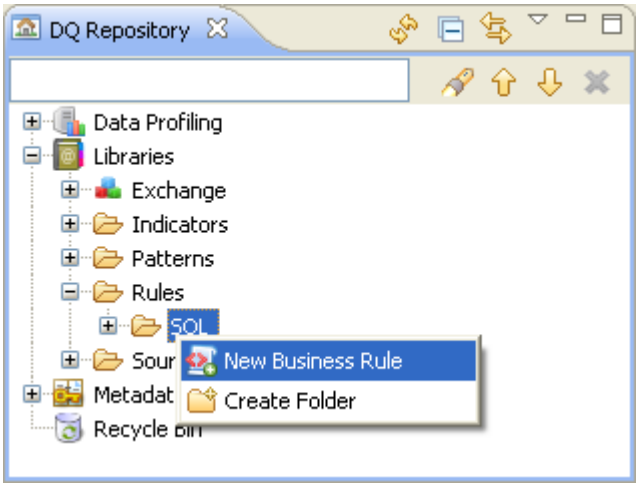
6.2.2.1. How to create an SQL business rule

SQL business rules can be simple rules with WHERE clauses. They can also have join conditions in them to combine common values between columns in database tables and give a result data set.

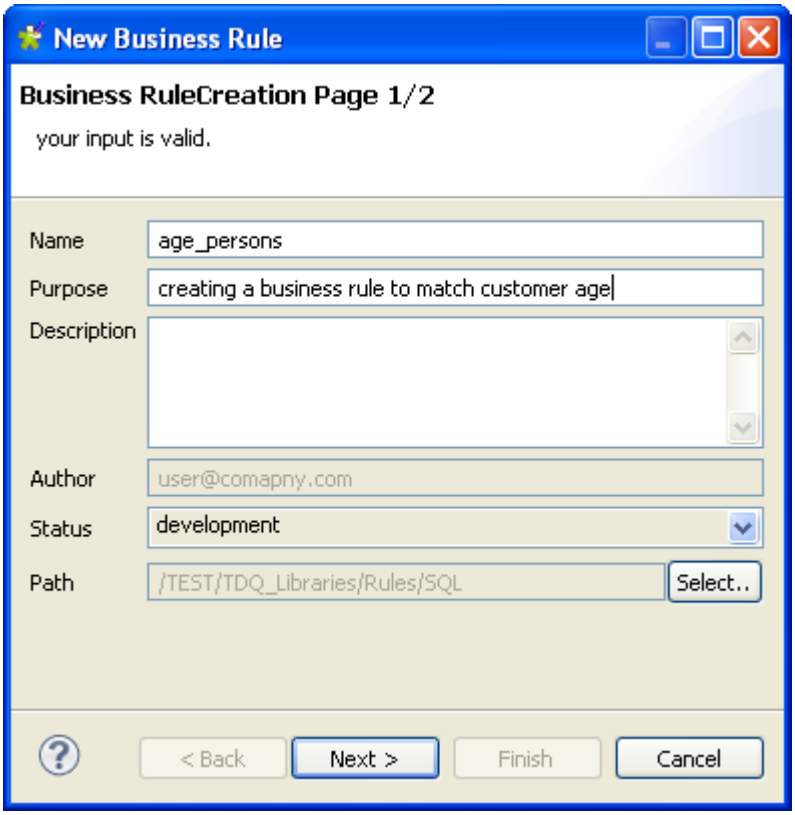
For an example of a table analysis with a simple business rule, see section *How to create a table analysis with a simple SQL business rule*. For an example of a table analysis with a business rule that has a join condition, see section *How to create a table analysis with an SQL business rule with a join condition*.

Creating the business rule

1. In the **DQ Repository** tree view, expand **Libraries > Rules**.
2. Right-click **SQL**.



3. From the contextual menu, select **New Business Rule** to open the **[New Business Rule]** wizard.

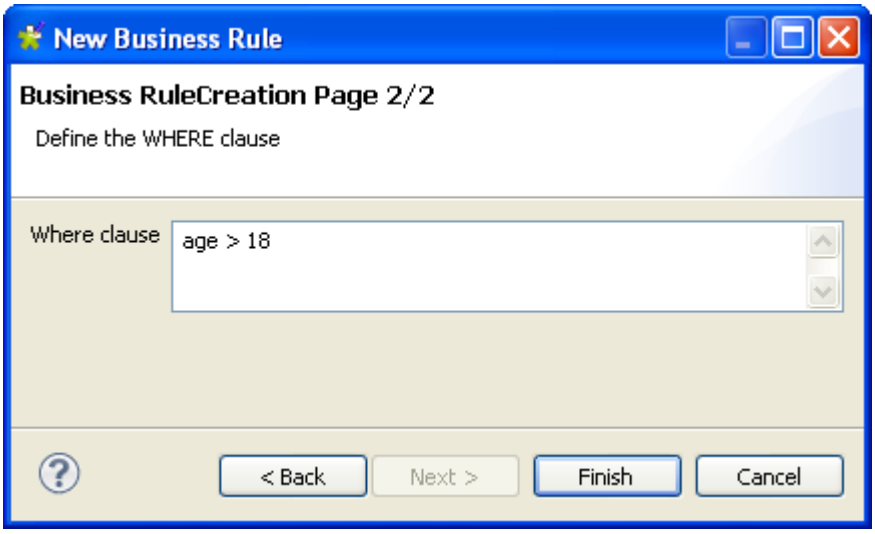


Consider as an example that you want to create a business rule to match the age of all customers listed in the *age* column of a defined table. You want to filter all the age records to identify those that fulfill the specified criterion.

4. In the **Name** field, enter a name for this new SQL business rule.

   Space is not acceptable when typing in the business rule name in this field.

5. Set other metadata (purpose, description and author name) in the corresponding fields and then click **Next**.

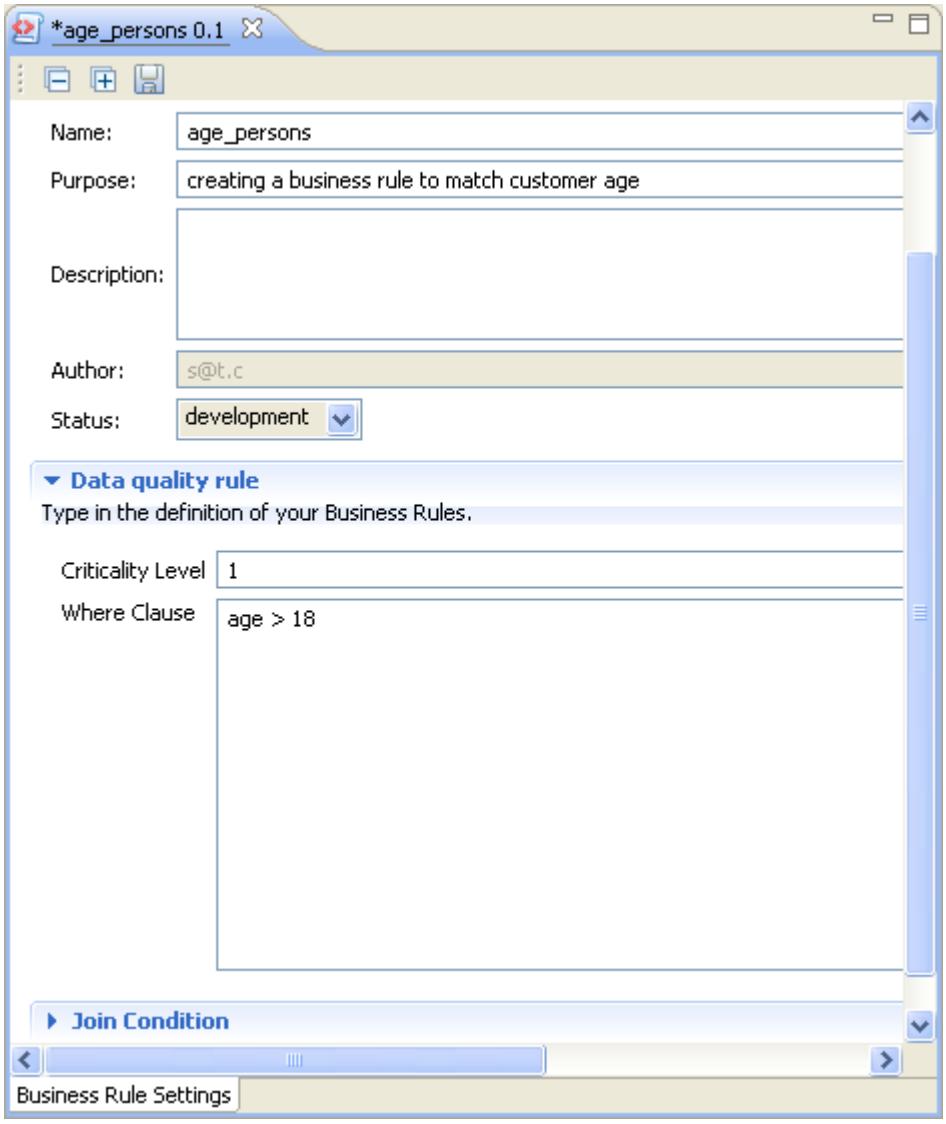


6. In the **Where clause** field, enter the WHERE clause to be used in the analysis.

   In this example, the WHERE clause is used to match the records where customer age is greater than 18.

7. Click **Finish** to close the **[New Business Rule]** wizard.

   A sub-folder for this new SQL business rule is displayed under the **Rule**s folder in the **DQ Repository** tree view. The SQL business rule editor opens with the defined metadata.



In the SQL business rule editor, you can modify the WHERE clause or add a new one directly in the **Data quality rule** view.

8. If required, set a value in the **Criticality Level** field.

   This will act as an indicator to measure the importance of the SQL business rule.

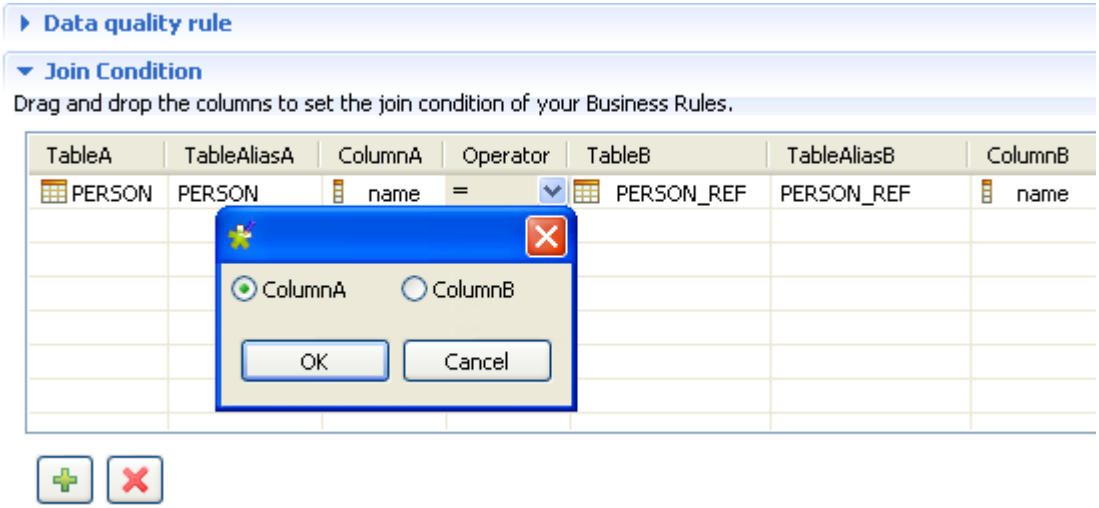
Creating a join condition

This step is not obligatory. You can decide to create a business rule without a join condition and use it with only the WHERE clause in the table analysis.

For an example of a table analysis with a simple business rule, see section *How to create a table analysis with a simple SQL business rule*. For an example of a table analysis with a business rule that has a join condition, see section *How to create a table analysis with an SQL business rule with a join condition*.

1. In the SQL business rule editor, click **Join Condition** to open the corresponding view.

2. Click the **[+]** button to add a row in the **Join Condition** table.



3. Expand the **Metadata** folder in the **DQ Repository** tree view, and then browse to the columns in the tables on which you want to create the join condition.

   This join condition will define the relationship between a table A and a table B using a comparison operator on a specific column in both tables. In this example, the join condition will compare the "name" value in the *Person* and *Person_Ref* tables that have a common column called *name*.

   You must be careful when defining the join clause. In order to get an easy to understand result, it is advisable to make sure that the joined tables do not have duplicate values. For further information, see section *How to create a table analysis with an SQL business rule with a join condition*.

4. Drop the columns from the **DQ Repository** tree view to the **Join Condition** table.

   A dialog box is displayed prompting you to select where to place the column: in **TableA** or in **TableB**.

5. Select a comparison condition operator between the two columns in the tables and save your modifications.

   In the analysis editor, you can now drop this newly created SQL business rule onto a table that has an "age" column. When you run the analysis, the join to the second column is done automatically.

   The table to which to add the business rule must contain at least one of the columns used in the SQL business rule.

For more information about using SQL business rules as indicators on a table analysis, see section *Creating a table analysis with SQL business rules*.

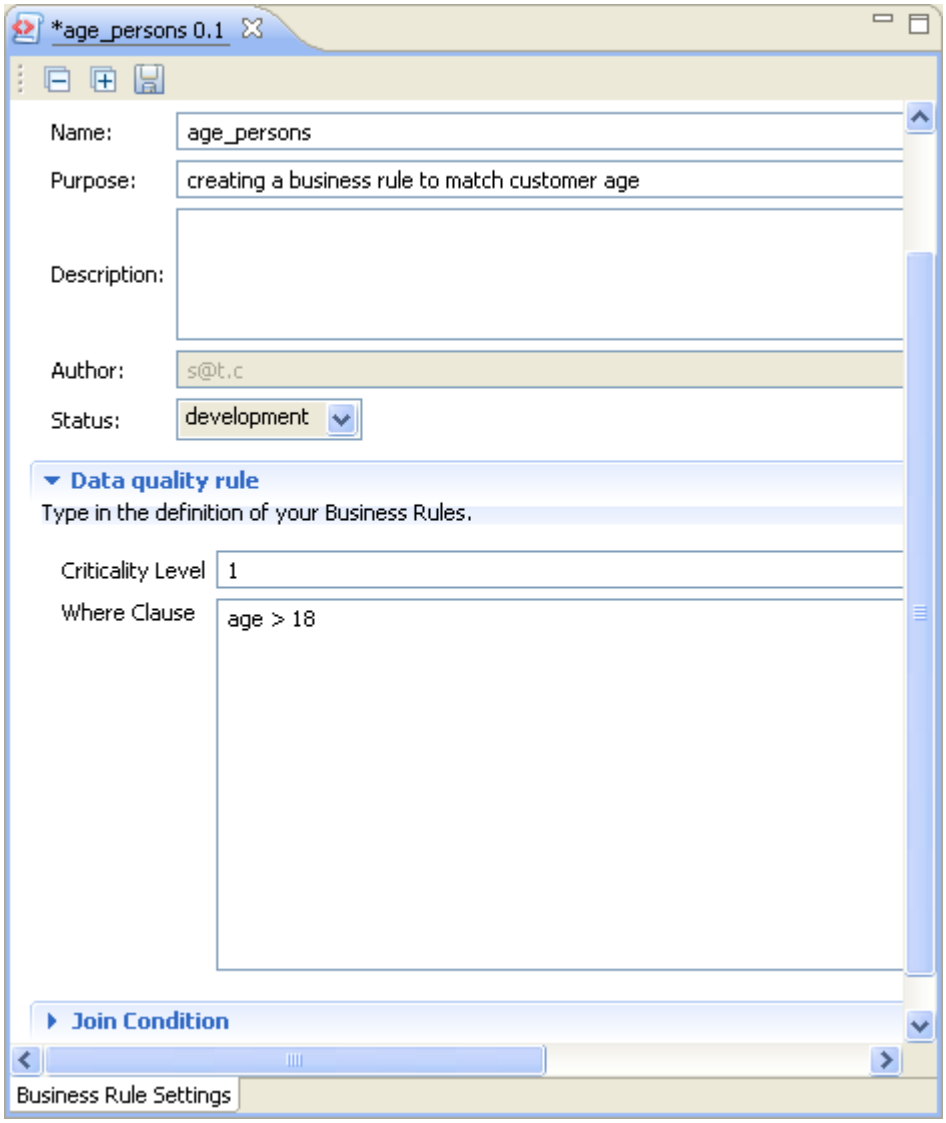
6.2.2.2. How to edit an SQL business rule

To edit an SQL business rule, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Rules > SQL**.

2. Right-click the SQL business rule you want to open and select **Open** from the contextual menu.

   The SQL business rule editor opens displaying the rule metadata.



3. Modify the business rule metadata or the WHERE clause as required.
4. Click the save icon on top of the editor to save your modifications.

   The SQL business rule is modified as defined.

6.2.2.3. How to create a table analysis with a simple SQL business rule

You can create analyses on either tables or views in a database using SQL business rules. The procedure for creating such analysis is the same for a table or a view.

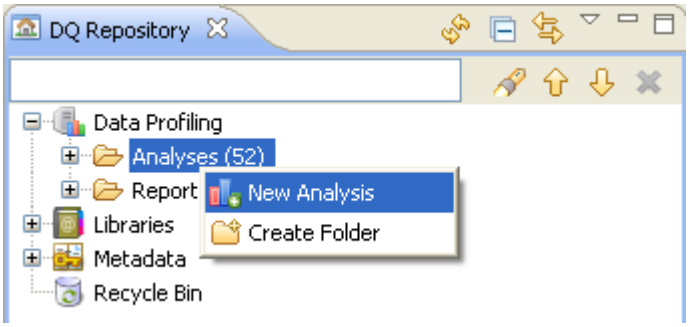
Prerequisite(s):

- At least one SQL business rule has been created in the **Profiling** perspective of the studio. For further information about creating SQL business rules, see section *How to create an SQL business rule*
- At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

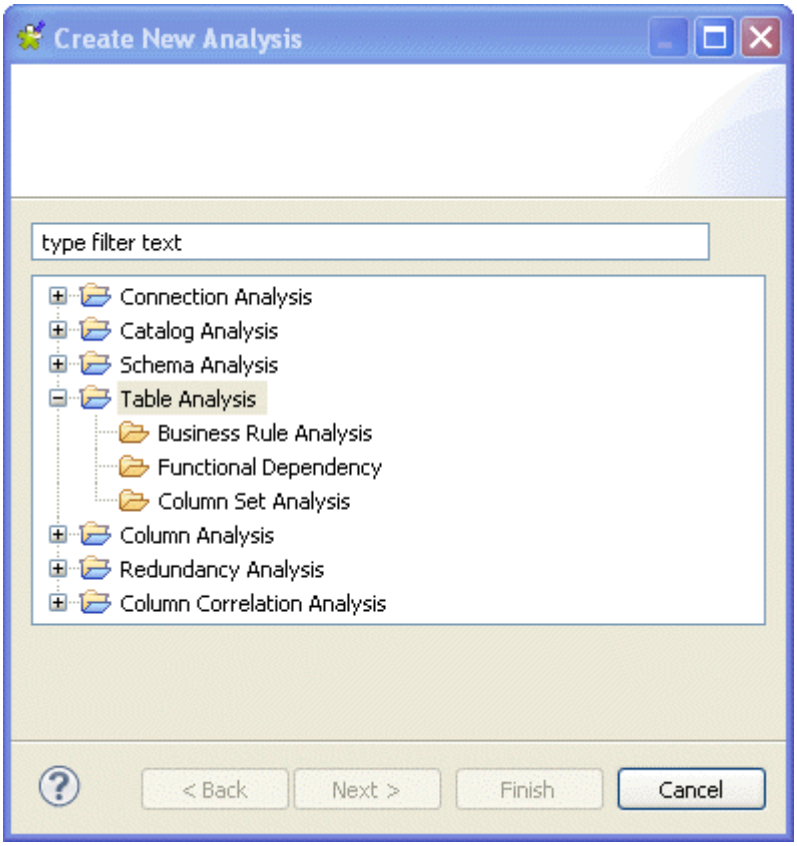
In this example, you want to add the SQL business rule created in section *How to create an SQL business rule* to a *top_custom* table that contains an *age* column. This SQL business rule will match the customer ages to define those who are older than 18.

Defining the analysis

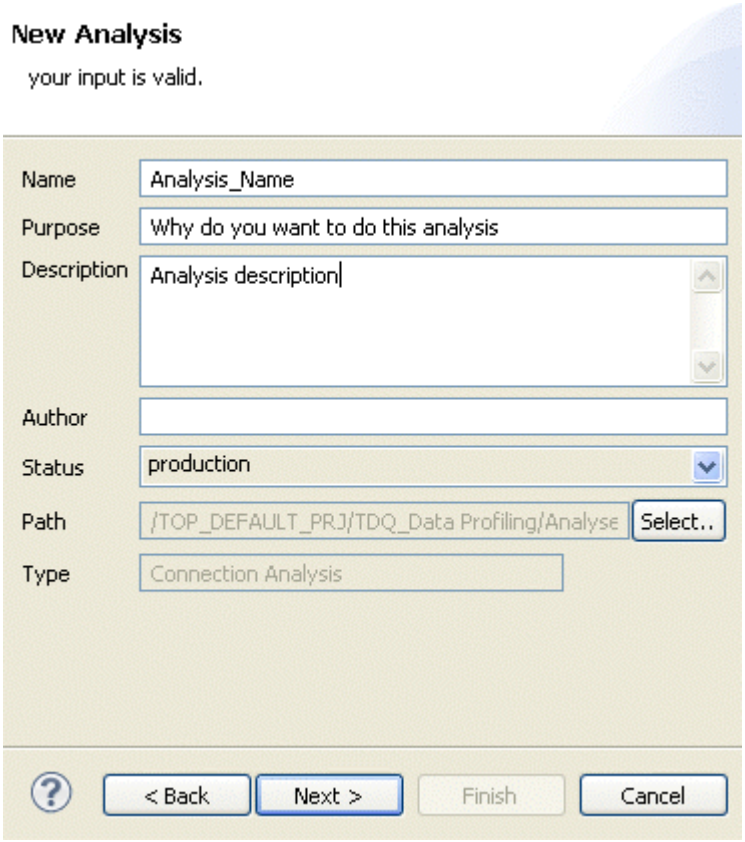
1. In the **DQ Repository** tree view, expand **Data Profiling**.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



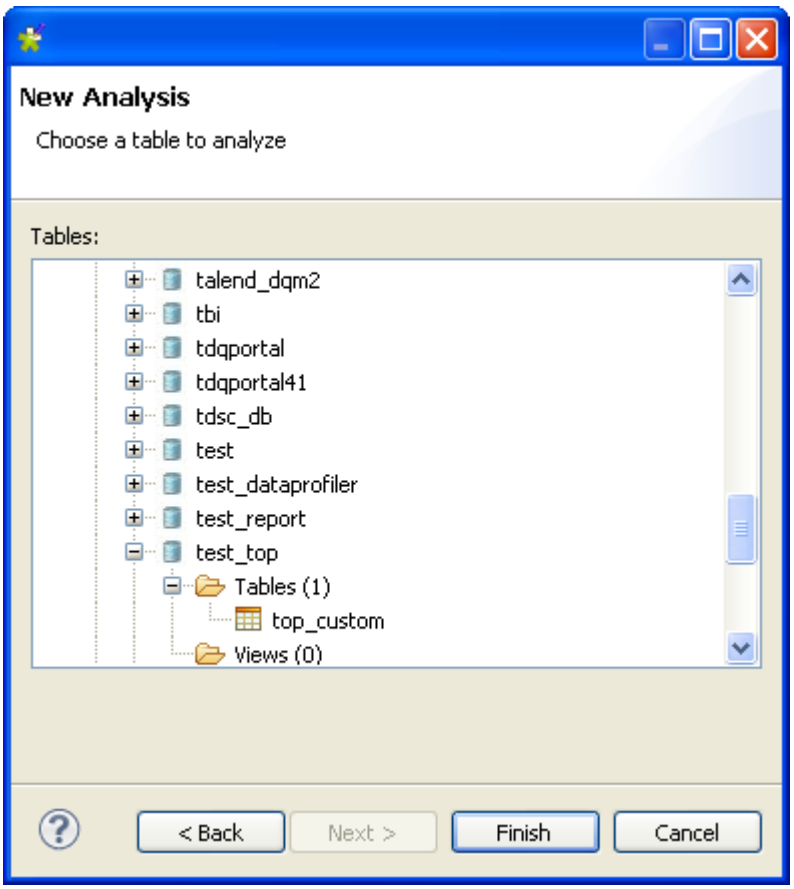
3. Expand the **Table Analysis** node and then select **Business Rule Analysis**.
4. Click the **Next** button to proceed to the next step.



5. In the **Name** field, enter a name for the current analysis.

   Space is not acceptable when typing in the analysis name in this field.

6. Set the analysis metadata (purpose, description and author name) in the corresponding fields and then click **Next**.

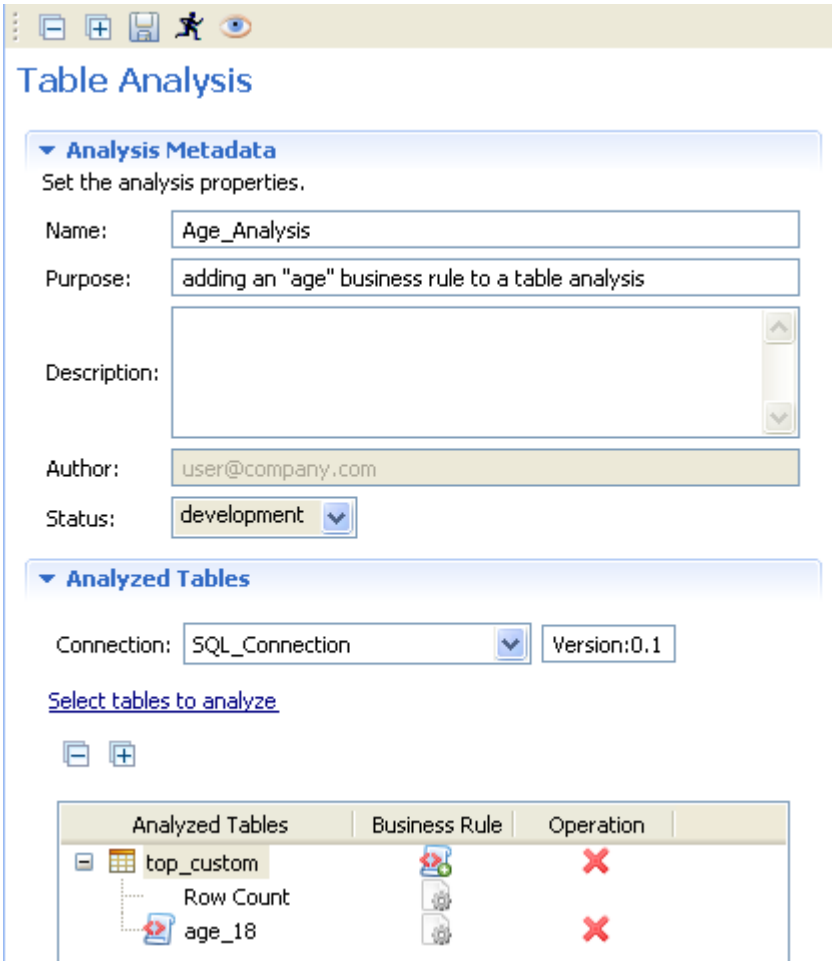


Selecting the table you want to analyze

1. Expand **DB Connections**, browse to the table to be analyzed and select it.
2. Click **Finish** to close the **[Create New Analysis]** wizard.

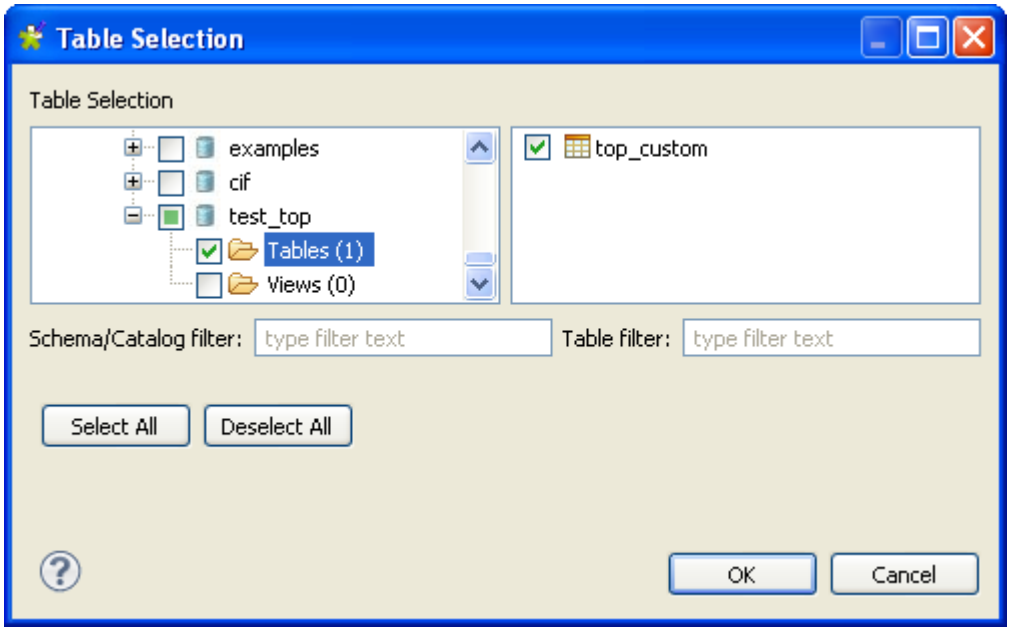
You can directly select the data quality rule you want to add to the current analysis by clicking the **Next** button in the **[New Analysis]** wizard or you can do that at later stage in the **Analyzed Tables** view as shown in the following steps.

A folder for the newly created table analysis is listed under the **Analyses** folder in the **DQ Repository** tree view, and the analysis editor opens with the defined metadata.



3. Click the **Analyzed Tables** tab to open the **Analyzed Tables** view.

4. If required, click **Select tables to analyze** to open the **[Table Selection]** dialog box and modify the selection and/or select new table(s).

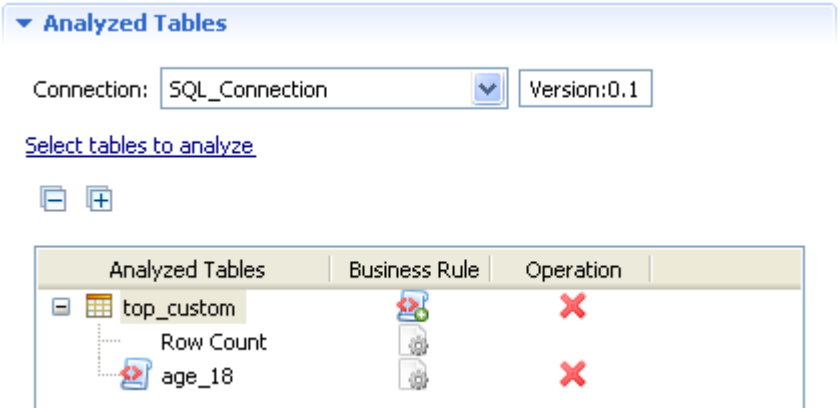


5. Expand **DB Connections** and browse to the table(s) you want to analyze.

You can filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

6. Select the check box next to the table name and click **OK**.

The selected table(s) is listed in the **Analyzed Tables** view.



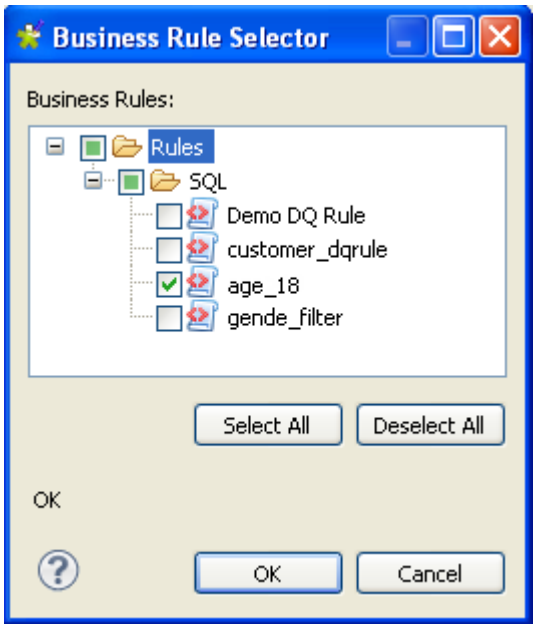
You can connect to a different database by selecting another connection from the **Connection** box. This box lists all the connections created in the Studio with the corresponding database names. If the tables listed in the **Analyzed Tables** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column is automatically located under the corresponding connection in the tree view.

Selecting the business rule

1. Click the icon next to the table name where you want to add the SQL business rule.

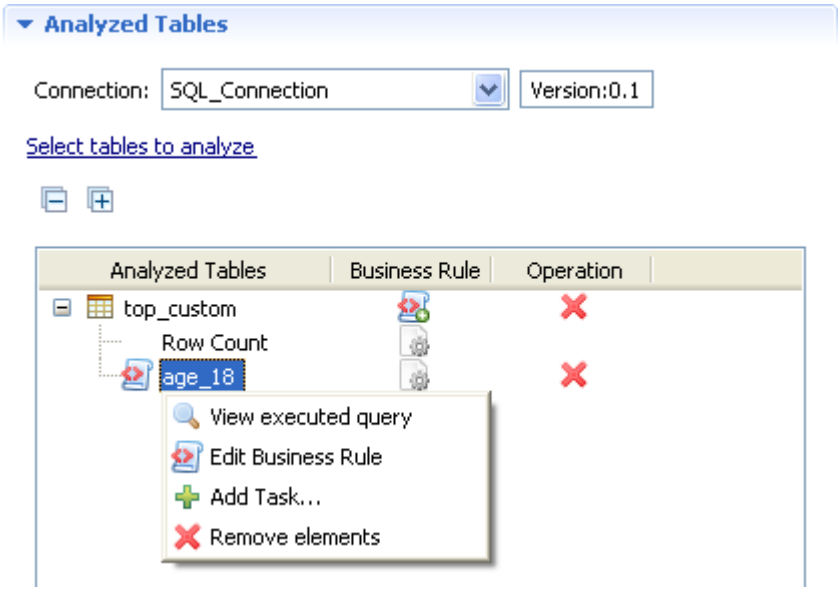
The **[Business Rule Selector]** dialog box is displayed.



2. Expand the **Rules** folder and select the check box(es) of the predefined SQL business rule(s) you want to use on the corresponding table(s).

3. Click **OK**.

The selected business rule is listed below the table name in the **Analyzed Tables** view.



You can also drag the business rule directly from the **DQ Repository** tree view to the table in the analysis editor.

4. If required, right-click the business rule and select **View executed query**.

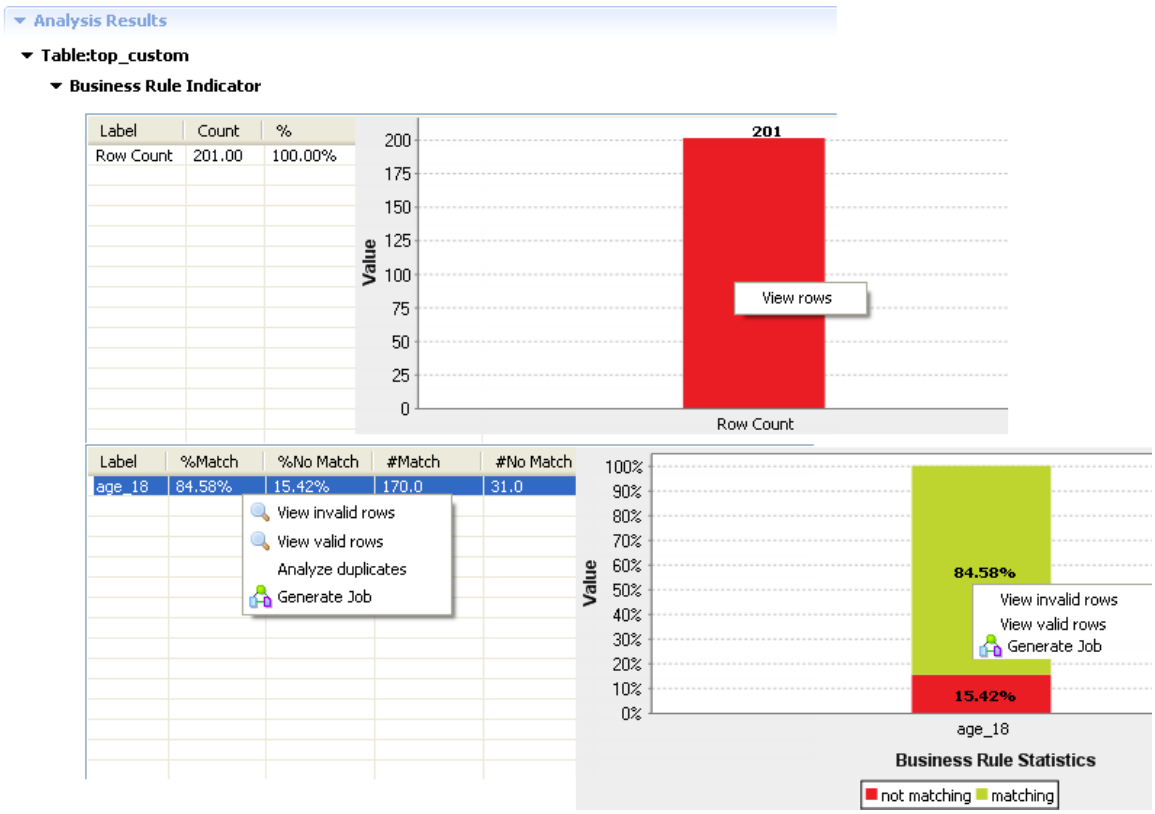
   The SQL editor opens in the Studio to display the query.

5. Click **Data Filter** in the analysis editor to open the view where you can set a filter on the data of the analyzed table(s).

6. Save the table analysis and press **F6** to execute it.

   An information pop-up opens to confirm that the operation is in progress. The table analysis results are displayed in the **Graphics** panel to the right.

7. Click **Analysis Results** at the bottom of the analysis editor to switch to the detail result view.



All age records in the selected table are evaluated against the defined SQL business rule. The analysis results has two bar charts: the first is a row count indicator that shows the number of rows in the analyzed table, and the second is a match and non-match indicator that indicates in red the age records from the "analyzed result set" that do not match the criteria (age below 18).

8. Right-click the business rule results in the second table, or right-click the result bar in the chart itself and select:

Option	To...
View valid rows	access a list in the SQL editor of all valid rows measured against the pattern used on the selected table
View invalid rows	access a list in the SQL editor of all invalid rows measured against the pattern used on the selected table
Analyze duplicates	generates a ready-to-use analysis that analyzes duplicates in the table, if any, and give the row and duplicate counts. For further information, see section *How to generate an analysis on the join results to analyze duplicates*.

For further information about the **Analysis Results** view, see section *How to access the detailed view of the analysis results*.

You can carry out a table analysis in a direct and more simplified way. For further information, see section *How to create a table analysis with an SQL business rule in a shortcut procedure*.

6.2.2.4. How to create a table analysis with an SQL business rule with a join condition

In some cases, you may need to analyze database tables or views using an SQL business rule that has a join clause that combines records from two tables in a database. This join clause will compare common values between two columns and give a result data set. Then the data in this set will be analyzed against the business rule.

Depending on the analyzed data and the join clause itself, several different results of the join are possible, for example #match + #no match > #row count, #match + #no match < #row count or #match + #no match = #row count.

The example below explains in detail the case where the data set in the join result is bigger than the row count (#match + #no match > #row count) which indicates duplicates in the processed data.

Prerequisite(s):

- At least one SQL business rule has been created in the **Profiling** perspective of the studio. For further information about creating SQL business rules, see section *How to create an SQL business rule*
- At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

In this example, you want to add the SQL business rule created in section *How to create an SQL business rule* to a *Person* table that contains the *age* and *name* columns. This SQL business rule will match the customer ages to define those who are older than 18. The business rule also has a join condition that compares the "name" value between the *Person* table and another table called *Person_Ref* through analyzing a common column called *name*.

Below is a capture of both tables:

age	name
7	John Smith
14	Edward Silver
23	John Doe
34	Jennifer Monroe
35	Jennifer Monroe
45	James Came

RefId	Name
1	John Doe
2	Jennifer Monroe
3	Jennifer Monroe
4	John Smith
5	Edward Silver
6	Fanny Compton
7	Maria Lapaloo

Below is a capture of the result of the join condition between these two tables:

1 [SELECT * FROM `my_pers...] | Messages

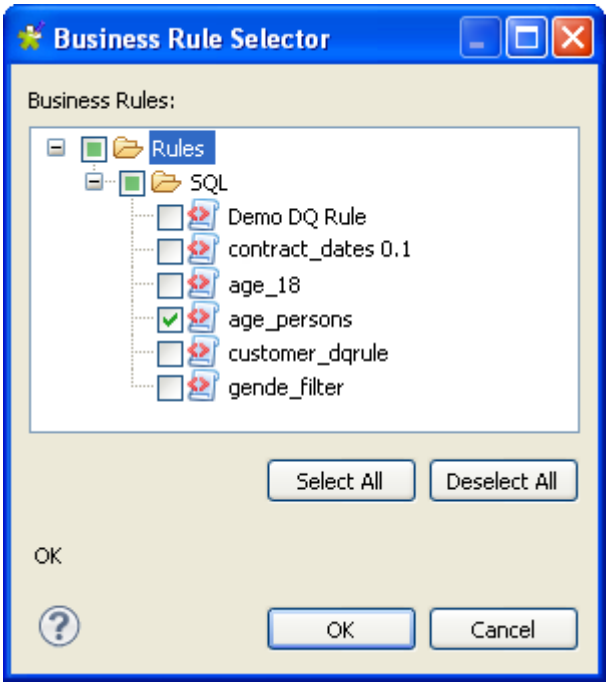
age	name	RefId	Name
23	John Doe	1	John Doe
34	Jennifer Monroe	2	Jennifer Monroe
35	Jennifer Monroe	2	Jennifer Monroe
34	Jennifer Monroe	3	Jennifer Monroe
35	Jennifer Monroe	3	Jennifer Monroe
7	John Smith	4	John Smith
14	Edward Silver	5	Edward Silver

The result set may give duplicate rows as it is the case here. Thus the results of the analysis may become a bit harder to understand. The analysis here will not analyze the rows of the table that match the business rule but it will run on the result set given by the business rule. See the end of the section for detail explanation of the analysis results.

1. Define the table analysis and select the table you want to analyze as outlined in section *How to create a table analysis with a simple SQL business rule*.

   The selected table is listed in the **Analyzed Tables** view.

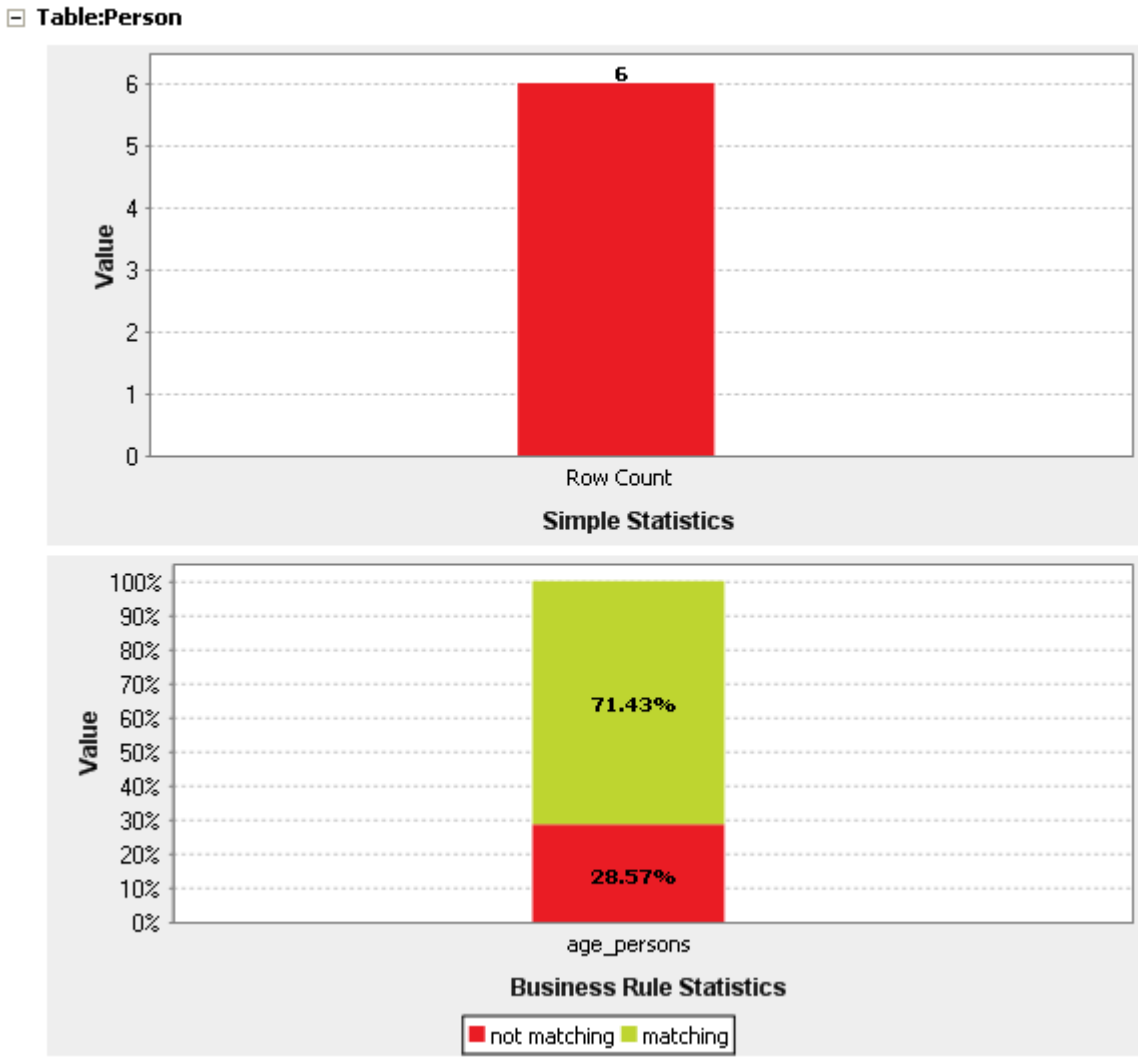
2. Add the business rule with the join condition to the selected table through clicking the icon next to the table name.



This business rule has a join condition that compares the "name" value between two different tables through analyzing a common column. For further information about SQL business rules, see section *How to create an SQL business rule*.

3. Save the table analysis and press **F6** to execute it.

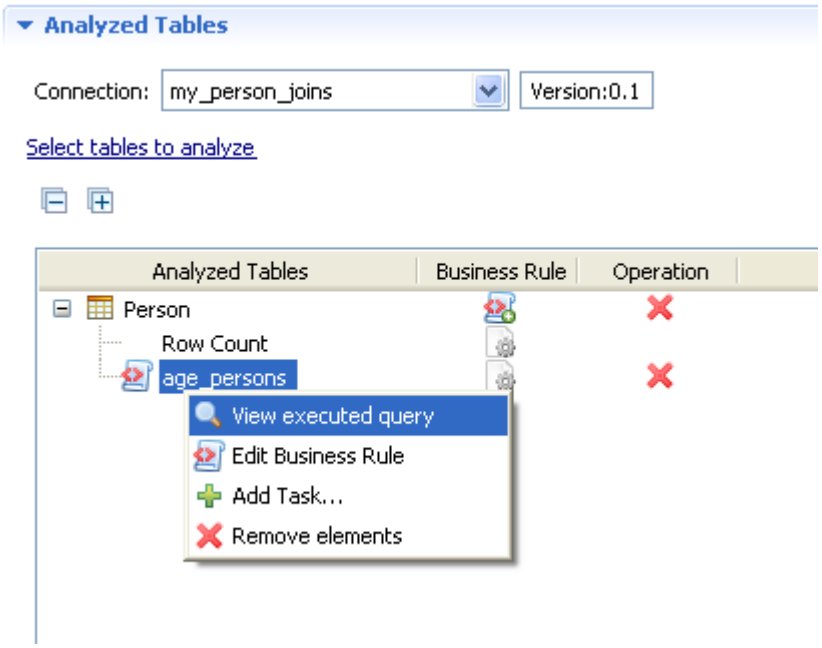
An information pop-up opens to confirm that the operation is in progress. The table analysis results are displayed in the **Graphics** panel to the right.



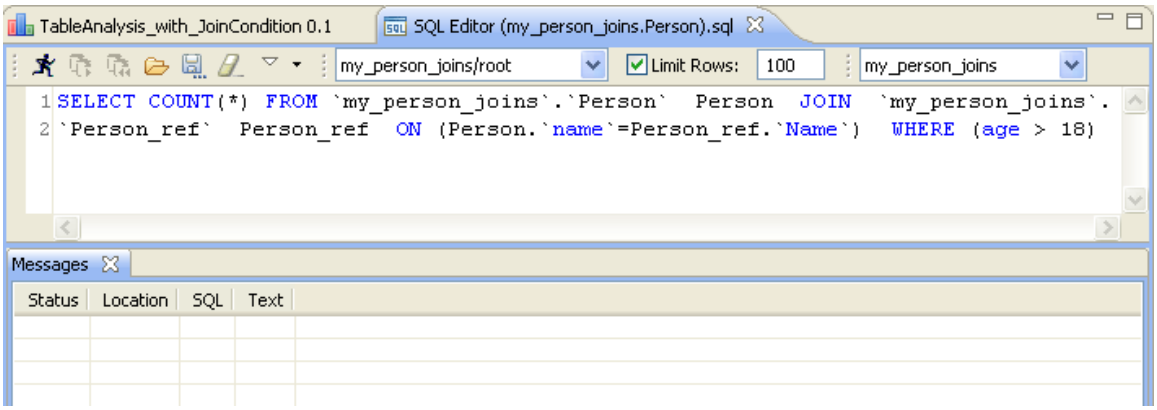
All age records in the selected table are evaluated against the defined SQL business rule. The analysis results has two bar charts: the first is a row count indicator that shows the number of rows in the analyzed table, and the second is a match and non-match indicator that indicates in red the age records from the "analyzed result set" that do not match the criteria (age below 18).

To better understand the **Business Rule Statistics** bar chart in the analysis results, do the following:

1. In the analysis editor, right-click the business rule and select **View executed query**.



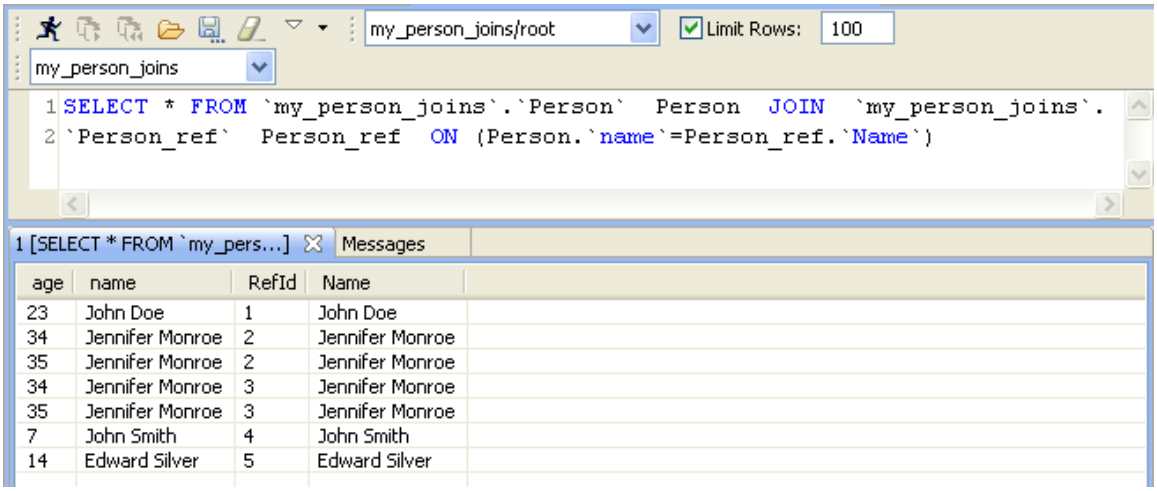
The SQL editor opens in the Studio.



2. Modify the query in the top part of the editor to read as the following: `SELECT * FROM `person_joins`.`PERSON` PERSON JOIN `person_joins`.`PERSON_REF` PERSON_REF ON (PERSON.`name`=PERSON_REF.`name`)`.

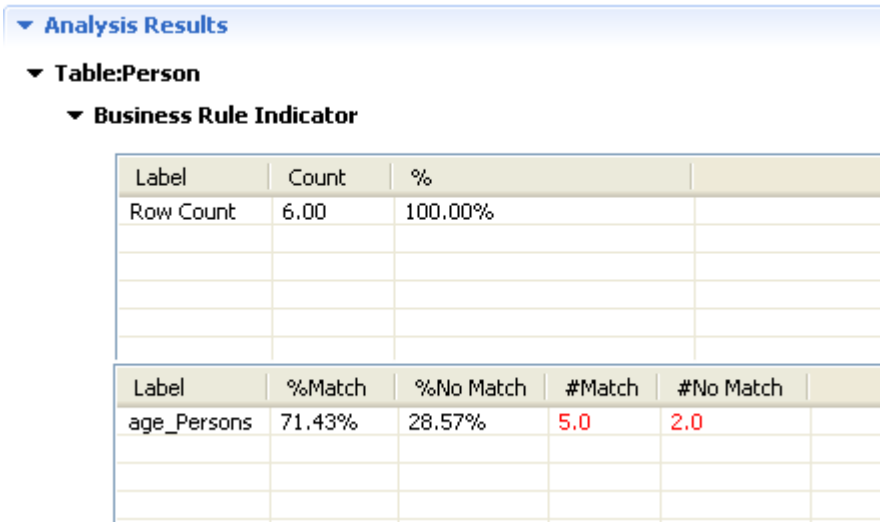
   This will list the result data set of the join condition in the editor.

3. In the top left corner of the editor, click the icon to execute the query.



The query result, that is the analyzed result set, is listed in the bottom part of the editor.

4. Click the **Analysis Results** tab at the bottom of the analysis editor to open a detail view of the analysis results.



The analyzed result set may contain more or fewer rows than the analyzed table. In this example, the number of match and non-match records (5 + 2 = 7) exceeds the number of analyzed records (6) because the join of the two tables generates more rows than expected.

Here 5 rows (71.43%) match the business rule and 2 rows do not match. Because the join generates duplicate rows, this result does not mean that 5 rows of the analyzed table match the business rule. It only means that 5 rows among the 7 rows of the result set match the business rule. Actually, some rows of the analyzed tables may not be even analyzed against the business rule. This happens when the join excludes these rows. For this reason, it is advised to check for duplicates on the columns used in the join of the business rule in order to make sure that the join does not remove or add rows in the analyzed result set. Otherwise the interpretation of the result is more complex.

For further information on the result detail view, see section *How to access the detailed view of the analysis results*.

In the **Analysis Results** view, if the number of match and non-match records exceeds the number of analyzed records, you can generate a ready-to-use analysis that will analyze the duplicates in the selected table. For further information, see section *How to access the detailed view of the analysis results*.

6.2.2.5. How to access the detailed view of the analysis results

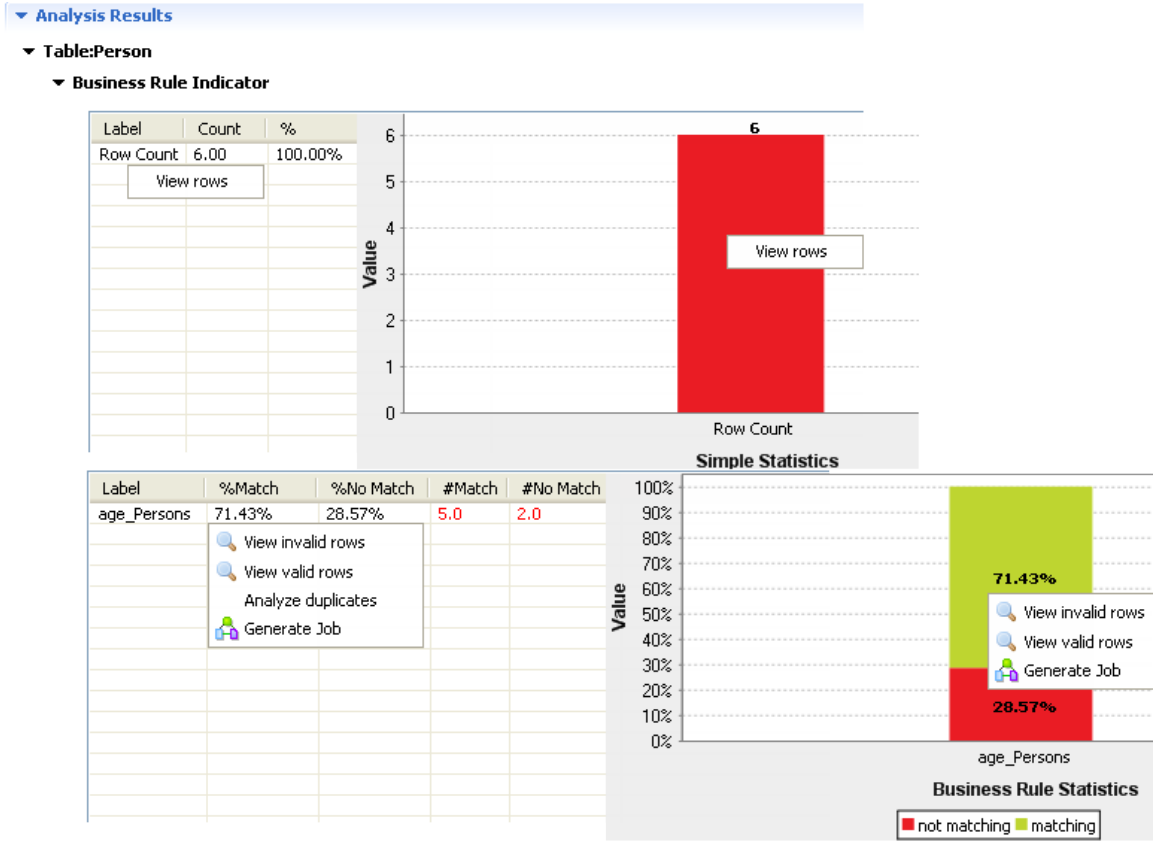
Prerequisite(s): A table analysis with an SQL business rule, that may have a join condition, is defined and executed in the **Profiling** perspective of the studio. For further information, see section *How to create a table analysis with an SQL business rule with a join condition*.

To access a more detailed view of a table analysis that uses an SQL business rule, do the following:

1. Click the **Analysis Results** tab at the bottom of the analysis editor to open the corresponding view.

   The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

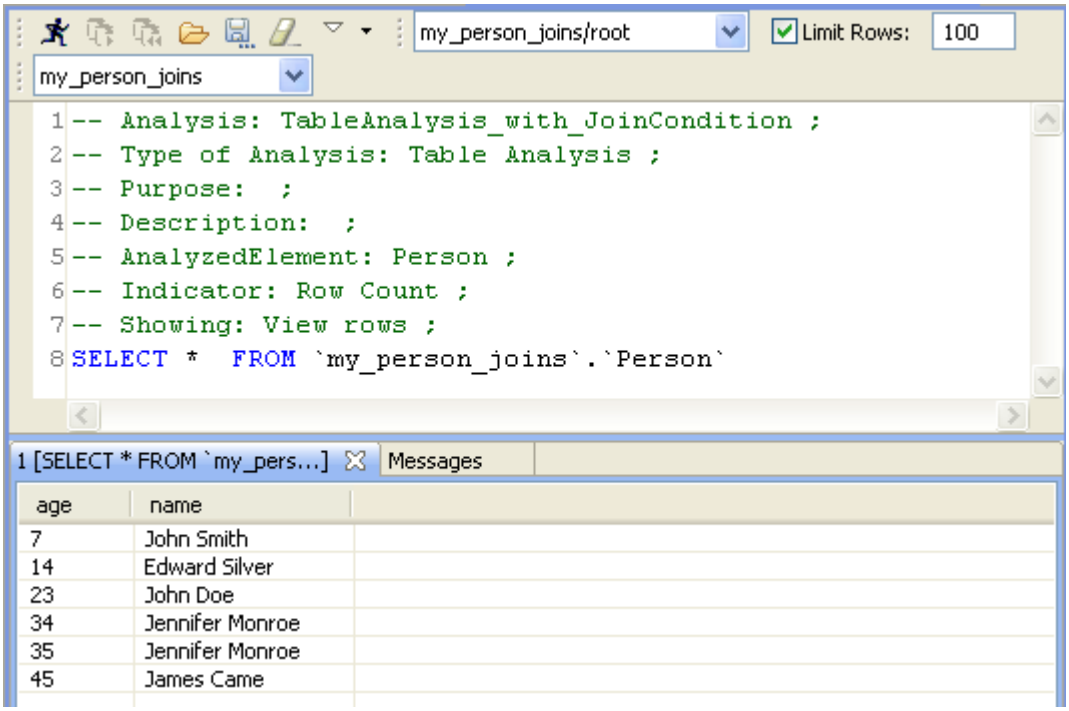
   The detailed analysis results view shows the two bar charts that indicate the number of the analyzed rows in the selected table and the percentage of the rows that match and non-match the SQL business rule. The bar charts are also accompanied with the tables that detail the statistic results.



   If a join condition is used in the SQL business rule, the number of the rows of the join (#match + # no match) can be different from the number of the analyzed rows (row count). For further information, see section *How to create a table analysis with an SQL business rule with a join condition*.

2. Right-click the **Row Count** row in the first table and select **View rows**.

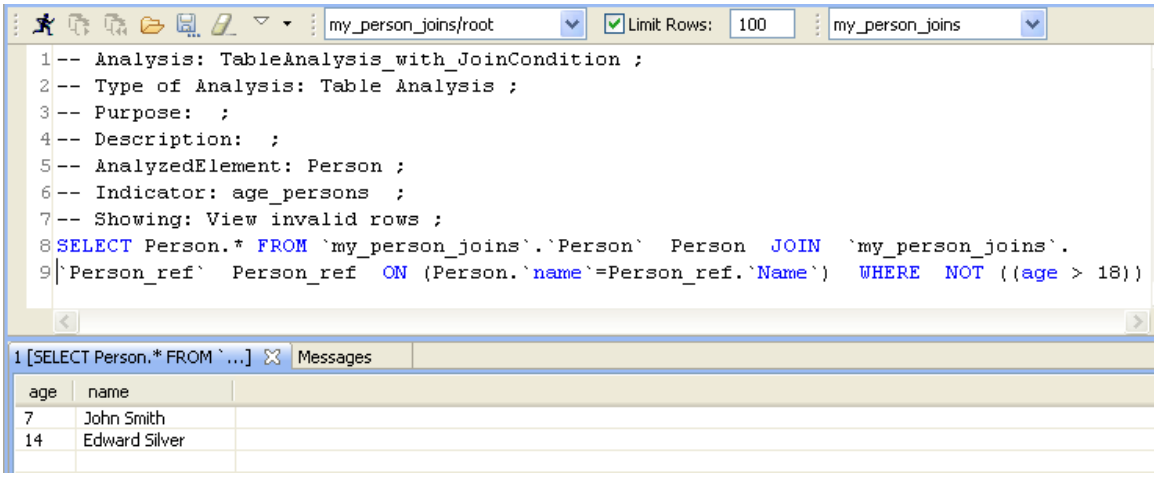
   The SQL editor opens in the Studio to display a list of the analyzed rows.



3. Right-click the business rule results in the second table, or right-click the result bar in the chart itself and select:

Option	To...
View valid rows	access a list in the SQL editor of all valid rows measured against the pattern used on the selected table
View invalid rows	access a list in the SQL editor of all invalid rows measured against the pattern used on the selected table
Analyze duplicates	generates a ready-to-use analysis that analyzes duplicates in the table and give the row and duplicate counts. For further information, see section *How to generate an analysis on the join results to analyze duplicates*.

Below is the list of the invalid rows in the analyzed table.



4. In the SQL editor, click the save icon on the toolbar to save the executed query on the SQL business rule and list it under the **Libraries > Source Files** folder in the **DQ Repository** tree view.

   For further information, see section *Saving the queries executed on indicators*.

6.2.2.6. How to generate an analysis on the join results to analyze duplicates

In some cases, when you analyze database tables using an SQL business rule that has a join clause, the join results show that there are more rows in the joint than in the analyzed table. This is because the columns in the analyzed table has some duplicate records, for an example see section *How to create a table analysis with an SQL business rule with a join condition*.

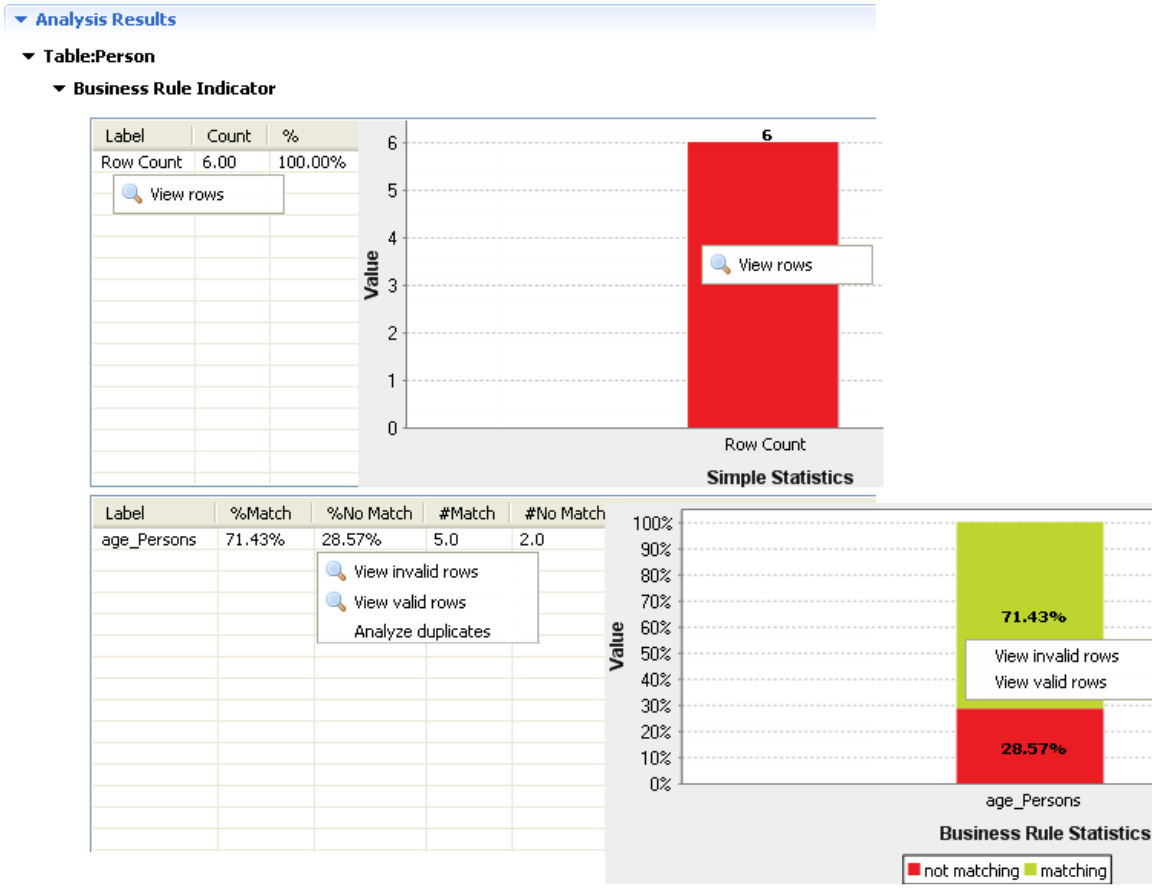
You can generate a ready-to-use analysis to analyze these duplicate records. The results of this analysis help you to better understand why there are more records in the join results than in the table.

Prerequisite(s): A table analysis with an SQL business rule, that has a join condition, is defined and executed in the **Profiling** perspective of the studio. The join results must show that there are duplicates in the table. For further information, see section *How to create a table analysis with an SQL business rule with a join condition*.

To generate an analysis that analyzes the duplicate records in a table, do the following:

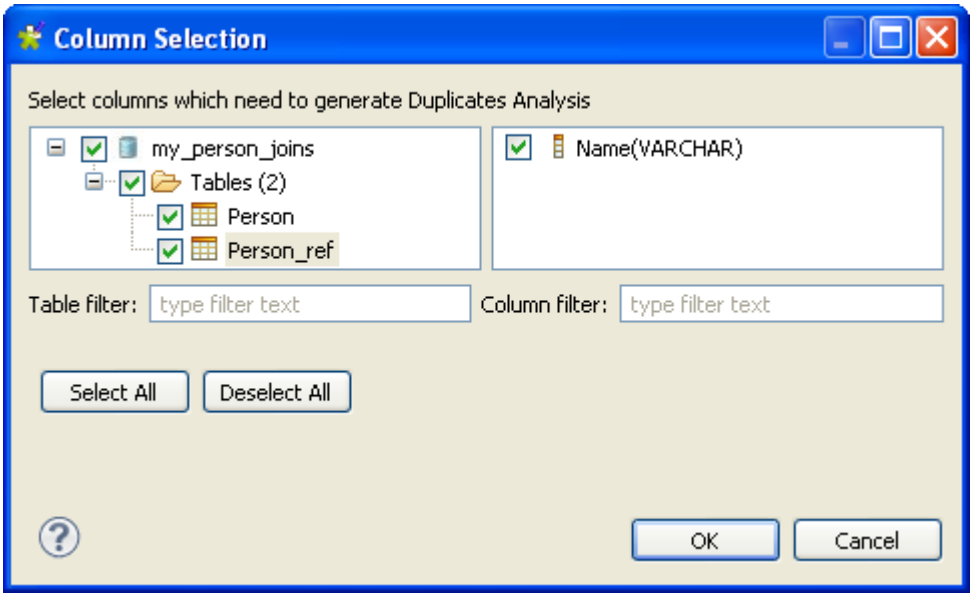
1. After creating and executing an analysis on a table that has duplicate records as outlined in section *How to create a table analysis with an SQL business rule with a join condition*, click the **Analysis Results** tab at the bottom of the analysis editor.

2. Right-click the join results in the second table and select **Analyze duplicates**.

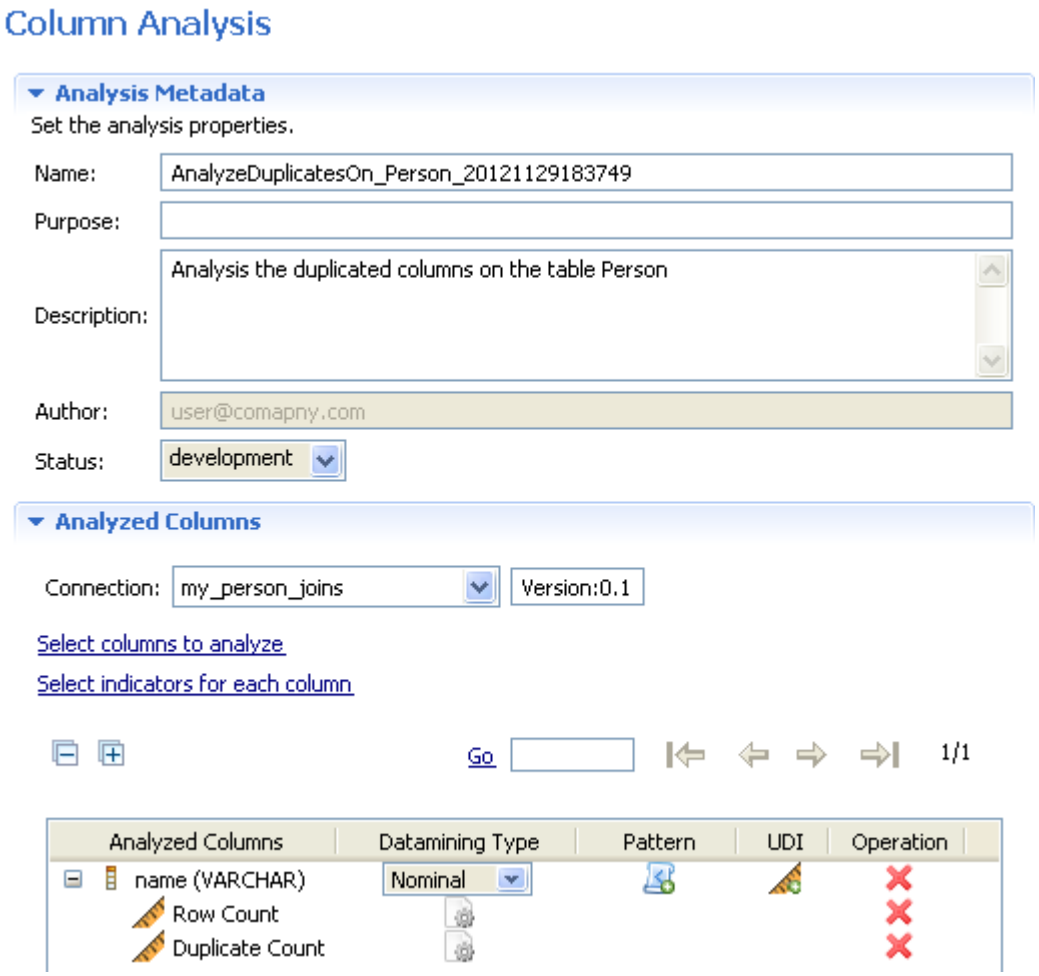


   The **[Column Selection]** dialog box opens with the analyzed tables selected by default.

3. Modify the selection in the dialog box if needed and then click **OK**.



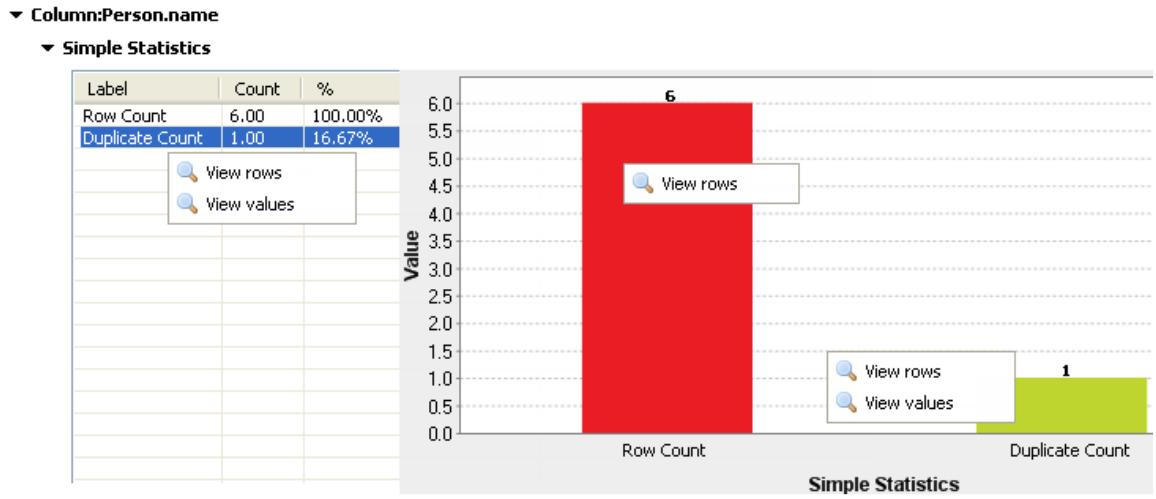
Two column analyses are generated and listed under the **Analyses** folder in the **DQ Repository** tree view and the analysis editor opens in the Studio on the settings of the generated analysis.



4. Press **F6** to execute the analysis.

The analysis results show two bars, one representing the row count of the data records in the analyzed column and the other representing the duplicate count.

5. Click **Analysis Results** at the bottom of the analysis editor to access the detail result view.



6. Right-click the row count or duplicate count results in the table, or right-click the result bar in the chart itself and select:

Option	To...
View rows	open a view on a list of all data rows or duplicate rows in the analyzed column.
View values	open a view on a list of the duplicate data values of the analyzed column.

6.2.2.7. How to create a table analysis with an SQL business rule in a shortcut procedure

You can use a simplified way to create a table analysis with a predefined business rule. All what you need to do is to start from the table name under the relevant **DB Connection** folder.

Prerequisite(s):

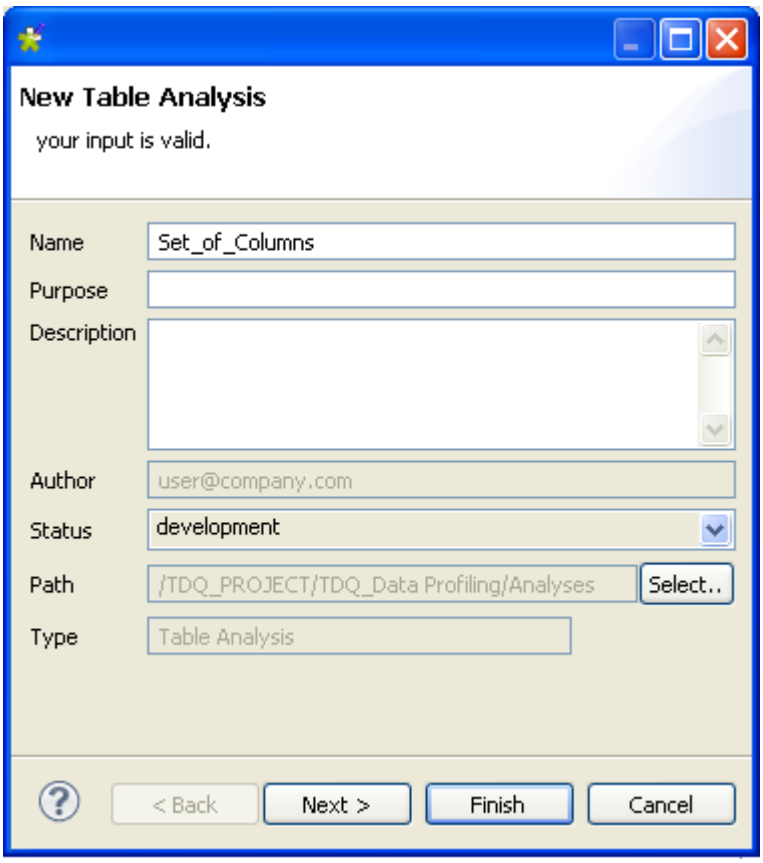
- At least one SQL business rule is created in the **Profiling** perspective of the studio.
- At least one database connection is set in the **Profiling** perspective of the studio.

For more information about creating SQL business rules, see section *How to create an SQL business rule*.

To create a table analysis with an SQL business rule in a shortcut procedure, do the following:

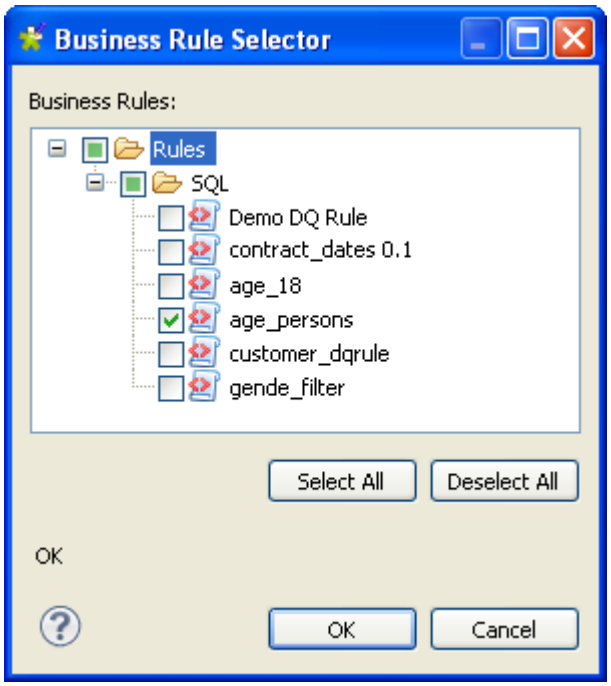
1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**, and then browse to the table you want to analyze.
2. Right-click the table name and select **Table analysis** from the list.

   The **[New Table Analysis]** wizard is displayed.



3. Enter the metadata for the new analysis in the corresponding fields and then click **Next** to proceed to the next step.

Space is not acceptable when typing in the table analysis name in the Name field.



4. Expand **Rules** > **SQL** and then select the check box(es) of the predefined SQL business rule(s) you want to use on the corresponding table(s).

5. Click **OK** to proceed to the next step.

   The table name along with the selected business rule are listed in the **Analyzed Tables** view.

6. If required, click **Data Filter** in the analysis editor to open the view where you can set a filter on the data of the analyzed table(s).

7. Save the table analysis and press **F6** to execute it.

   An information pop-up opens to confirm that the operation is in progress. The table analysis results are displayed in the **Graphics** panel to the right.

6.2.3. Detecting anomalies in the table columns: column functional dependency analysis

This type of analysis helps you to detect anomalies in column dependencies through defining columns as either “determinant” or “dependent” and then analyzing values in dependant columns against those in determinant columns.

This type of analysis detects to what extent a value in a determinant column functionally determines another value in a dependant column.

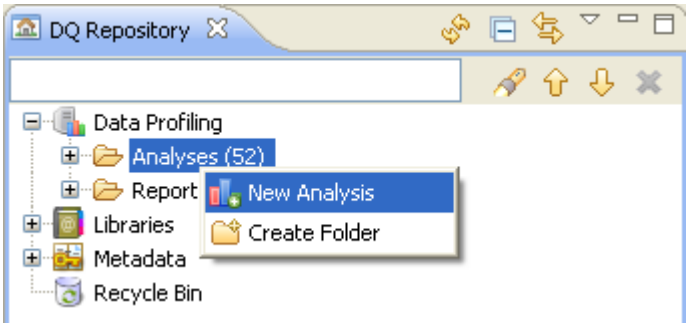
This can help you identify problems in your data, such as values that are not valid. For example, if you analyze the dependency between a column that contains United States Zip Codes and a column that contains states in the United States, the same Zip Code should always have the same state. Running the functional dependency analysis on these two columns will show if there are any violations of this dependency.

Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

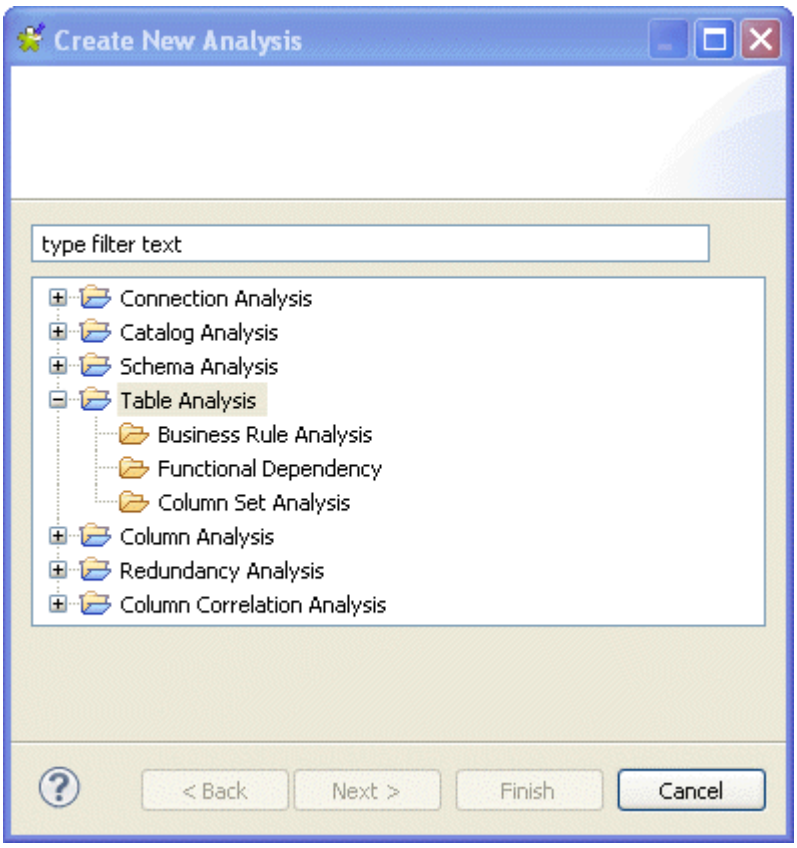
Defining the analysis

1. In the **DQ Repository** tree view, expand **Data Profiling**.

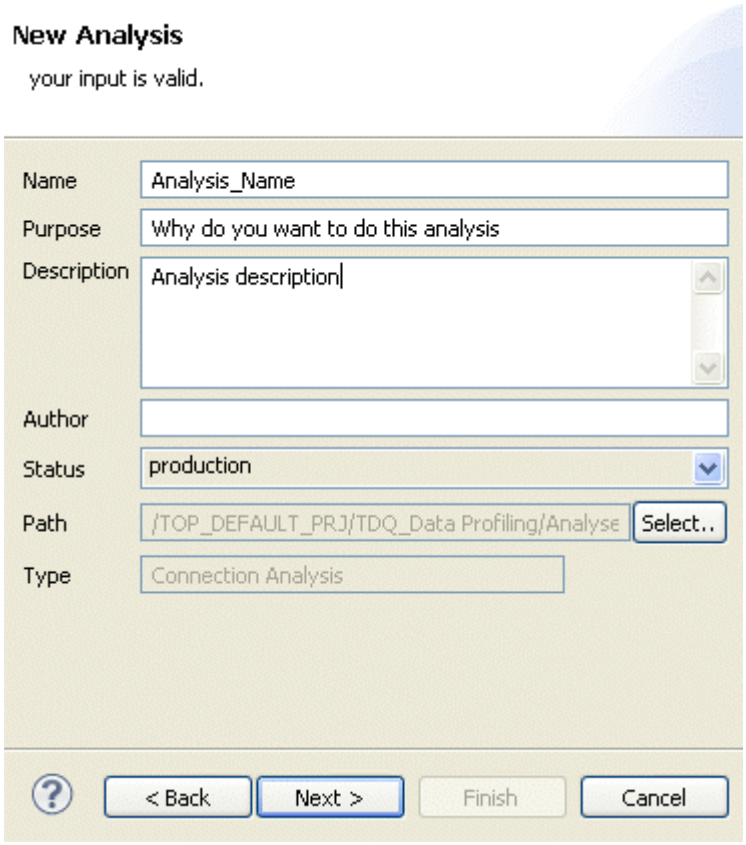
2. Right-click the **Analyses** folder and select **New Analysis**.



   The **[Create New Analysis]** wizard opens.



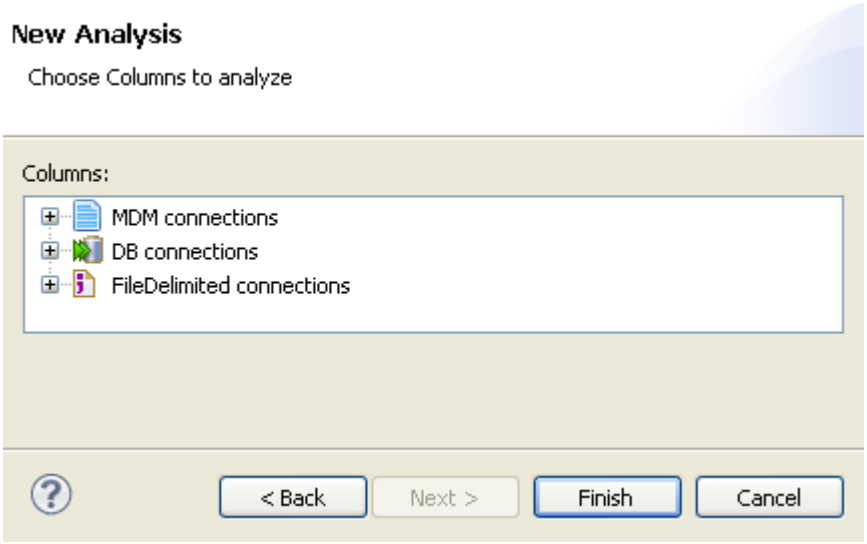
3. Expand the **Table Analysis** node and select **Functional Dependency**.
4. Click the **Next** button to proceed to the next step.



5. In the **Name** field, enter a name for the current analysis.

Space is not acceptable when typing in the analysis name in this field.

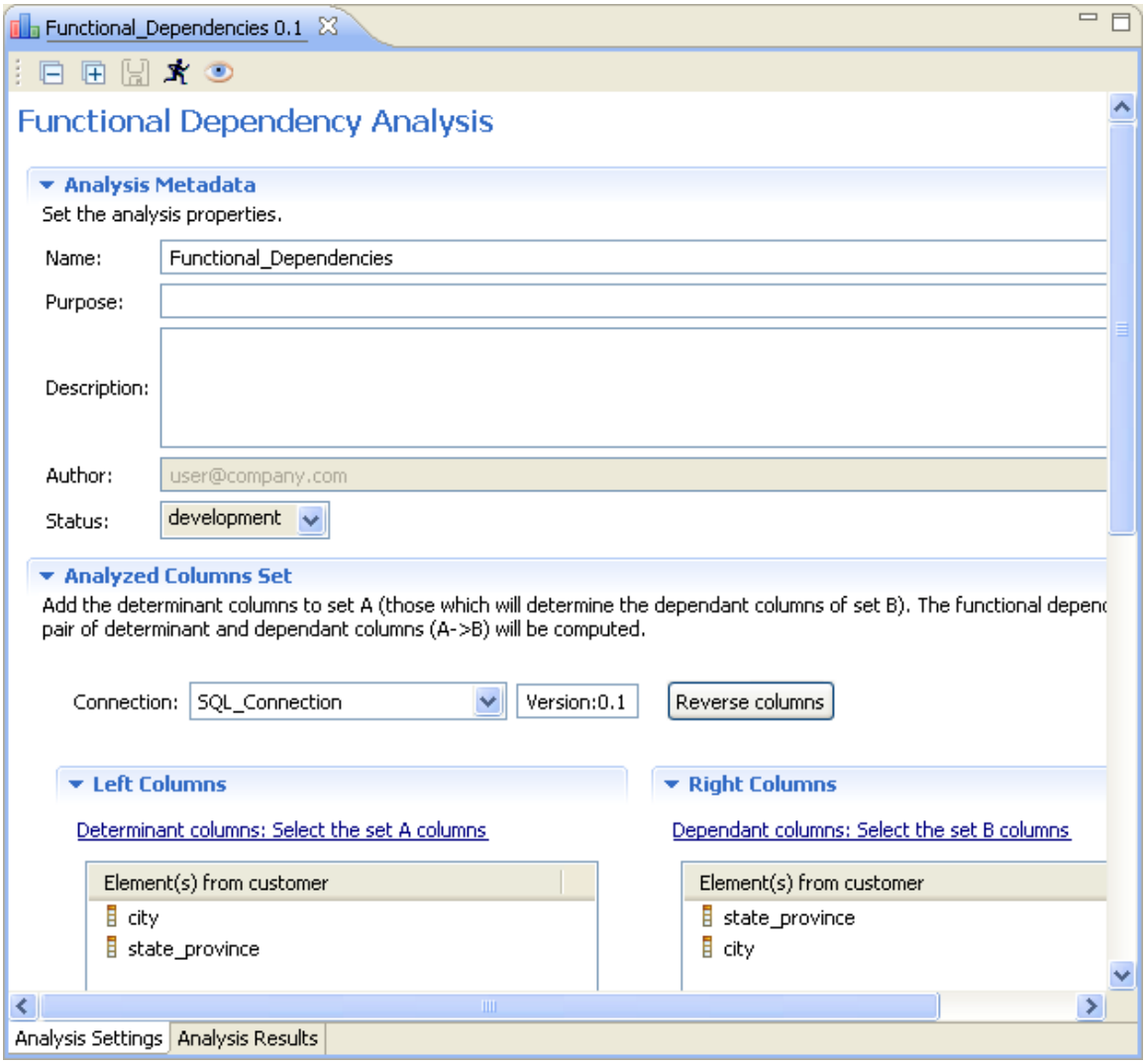
6. Set the analysis metadata (purpose, description and author name) in the corresponding fields, and then click **Next**.



Selecting the columns and executing the functional dependency analysis

1. Expand **DB connections**, and then browse to the columns you want to analyze, select them and then click **Finish** to close the **[New Analysis]** wizard.

   A folder for the newly created functional dependency analysis is listed under **Analysis** in the **DQ Repository** tree view, and the analysis editor opens with the defined metadata.



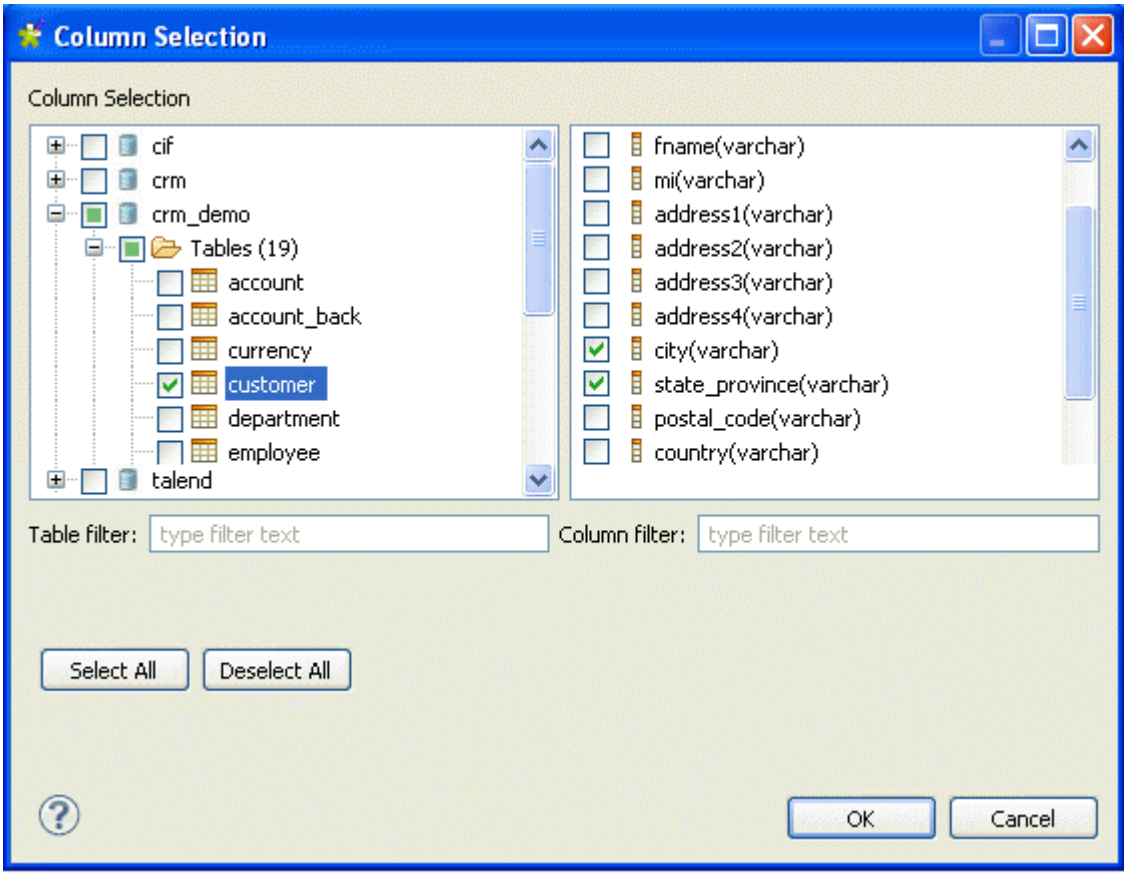
The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

2. Click the **Analyzed Column Set** tab to open the corresponding view.

3. Click **Determinant columns: Select columns from set A** to open the **[Column Selection]** dialog box.

   Here you can select the first set of columns against which you want to analyze the values in the dependant columns. You can also drag the columns directly from the **DQ Repository** tree view to the left column panel.

   In this example, you want to evaluate the records present in the *city* column and those present in the *state_province* column against each other to see if state names match to the listed city names and vice versa.

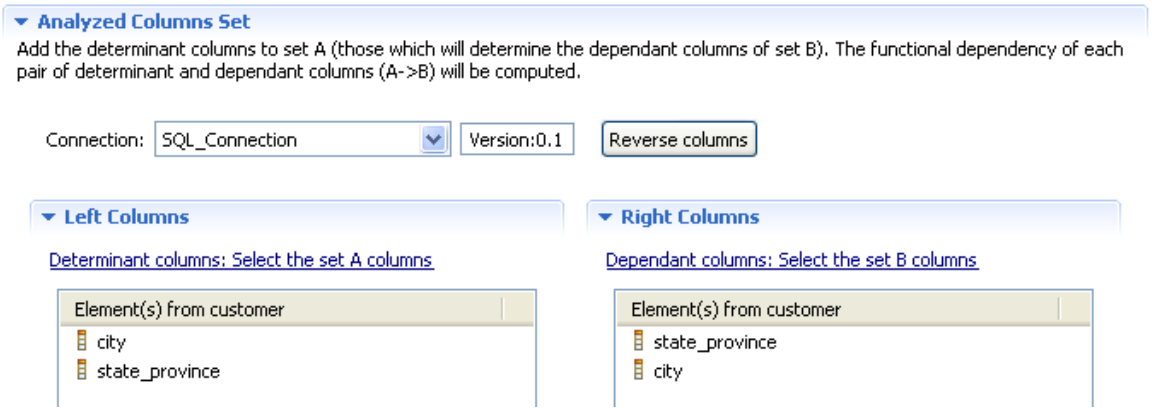


4. In the **[Column Selection]** dialog box, expand **DB Connections** and browse to the column(s) you want to define as determinant columns.

   You can filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

5. Select the check box(es) next to the column(s) you want to analyze and click **OK** to proceed to the next step.

   The selected column(s) are displayed in the **Left Columns** panel of the **Analyzed Columns Set** view. In this example, we select the *city* column as the determinant column.



6. Do the same to select the dependant column(s) or drag it/them from the **DQ Repository** tree view to the **Right Columns** panel. In this example, we select the *state_province* column as the dependent column. This relation will show if the state names match to the listed city names.

   If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column is automatically located under the corresponding connection in the tree view.

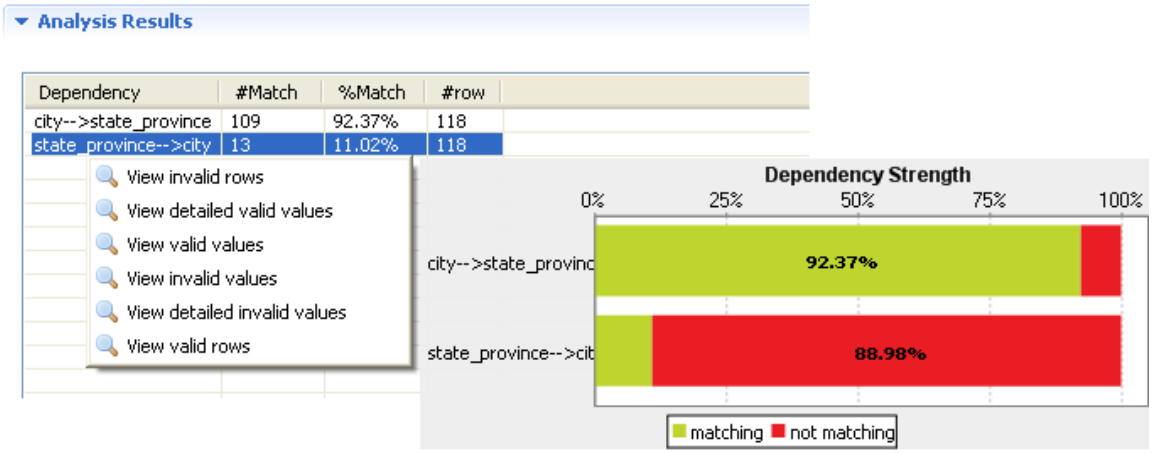
7. Click the **Reverse columns** tab to automatically reverse the defined columns and thus evaluate the reverse relation, what city names match to the listed state names.

You can select to connect to a different database by selecting another connection from the **Connection** box. This box lists all the connections created in the Studio with the corresponding database names. If the columns listed in the **Analyzed Columns Set** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

8. Click the save icon on top of the editor, and then press **F6** to execute the current analysis.

A progress information pop-up opens to confirm that the operation is in progress. The results of column functional dependency analysis are displayed in the **Analysis Results** view.

The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.



This functional dependency analysis evaluated the records present in the *city* column and those present in the *state_province* column against each other to see if the city names match to the listed state names and vice versa. The returned results, in the **%Match** column, indicate the functional dependency strength for each determinant column. The records that do not match are indicated in red.

The **#Match** column in the result table lists the numbers of the distinct determinant values in each of the analyzed columns. The **#row** column in the analysis results lists the actual relations between the determinant attribute and the dependant attribute. In this example, **#Match** in the first row of the result table represents the number of distinct cities, and **#row** represents the number of distinct pairs (city, state_province). Since these two numbers are not equal, then the functional dependency relationship here is only partial and the ratio of the numbers (**%Match**) measures the actual dependency strength. When these numbers are equal, you have a "strict" functional dependency relationship, i.e. each city appears only once with each state.

The presence of null values in either of the two analyzed columns will lessen the “dependency strength”. The system does not ignore null values, but rather calculates them as values that violates the functional dependency.

9. In the **Analysis Results** view, right-click any of the dependency lines and select:

Option	To...
View valid/invalid rows	access a list in the SQL editor of all valid/invalid rows measured according to the functional dependencies analysis
View valid/invalid values	access a list in the SQL editor of all valid/invalid values measured according to the functional dependencies analysis
View detailed valid/detailed invalid values	access a detailed list in the SQL editor of all valid/invalid values measured according to the functional dependencies analysis

From the SQL editor, you can save the executed query and list it under the **Libraries > Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see section *Saving the queries executed on indicators*.

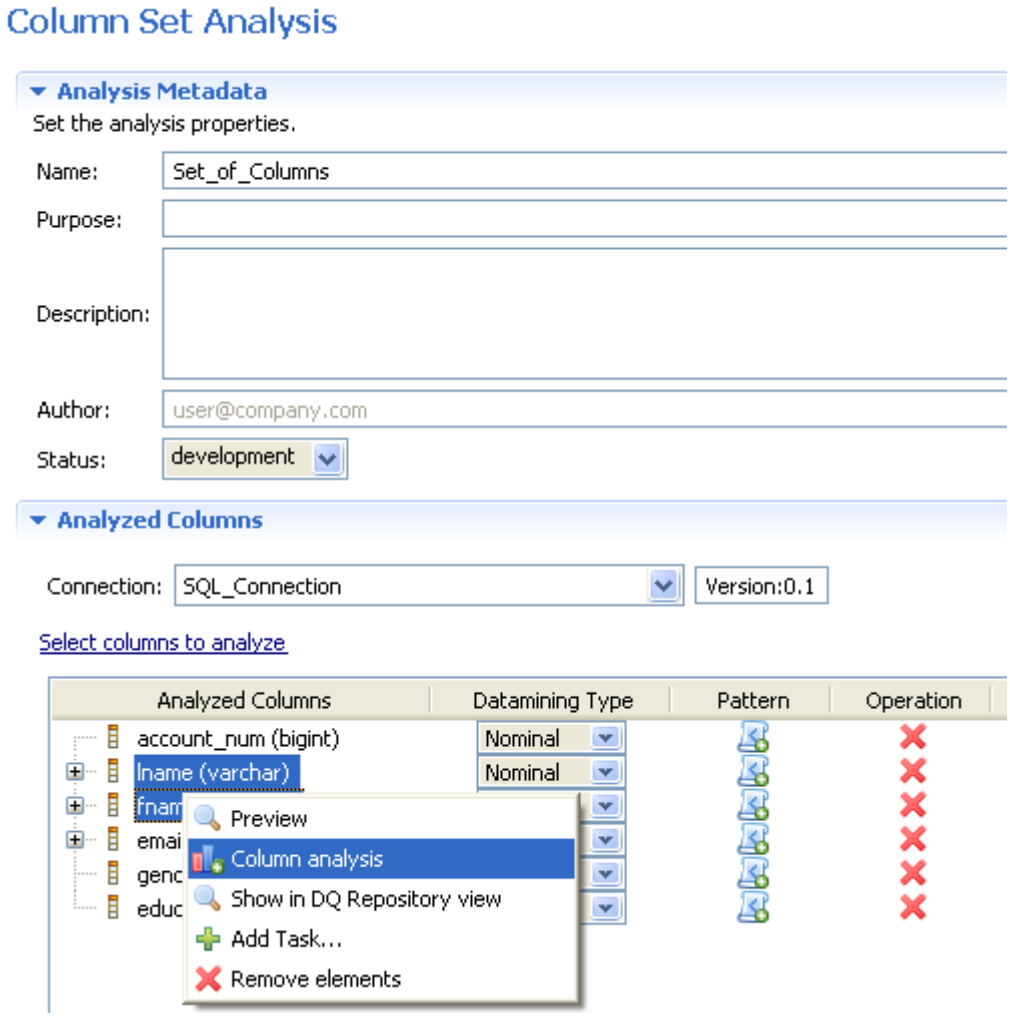
6.2.4. Creating a column analysis from a simple table analysis

You can create a column analysis on one or more columns defined in a simple table analysis (column set analysis).

Prerequisite(s):A simple table analysis is defined in the analysis editor in the **Profiling** perspective of the studio.

To create a column analysis on one or more columns defined in a simple table analysis, do the following:

1. Open the simple table analysis.
2. In the **Analyzed Columns** view, right-click the column(s) you want to create a column analysis on.



3. Select **Column analysis** from the contextual menu.

   The **[New Analysis]** wizard opens.

4. In the **Name** field, enter a name for the new column analysis and then click **Finish** to proceed to the next step.

   The analysis editor opens with the defined metadata and a folder for the newly created analysis is listed under the **Analyses** folder in the **DQ Repository** tree view.

5. Follow the steps outlined in section *Analyzing columns in a database* to continue creating the column analysis.

6.3. Analyzing tables in delimited files

You can analyze the content of a set of columns in a delimited file. This set can represent only some of the columns in the defined table or the table as a whole.

You can then execute the created analysis using the Java engine.

6.3.1. Creating a column set analysis on a delimited file using patterns

This type of analysis provide simple statistics on the number of records falling in certain categories, including the number of rows, the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields. For more information about these indicators, see section *Simple statistics*.

It is also possible to add patterns to this type of analysis and have a single-bar result chart that shows the number of the rows that match “all” the patterns.

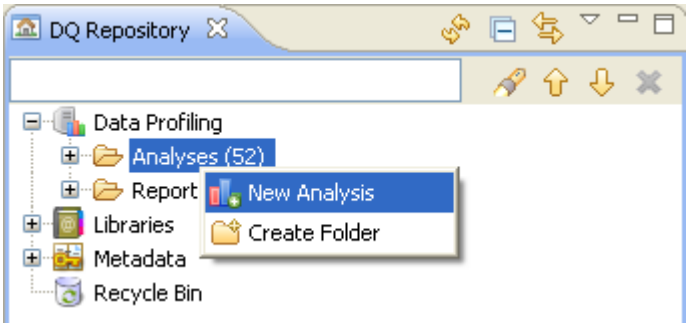
6.3.1.1. How to define the set of columns to be analyzed in a delimited file

Prerequisite(s): At least one connection to a delimited file is set in the studio. For further information, see section *Connecting to a database*.

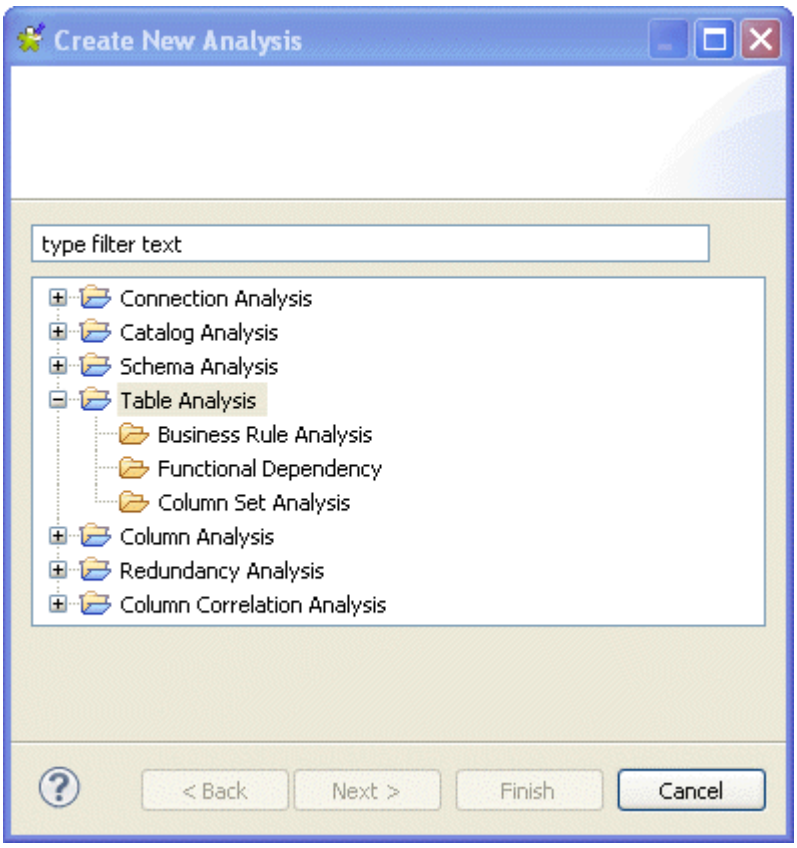
When carrying out this type of analysis, the set of columns to be analyzed must not include a primary key column.

To define the set of columns to analyzed, do the following:

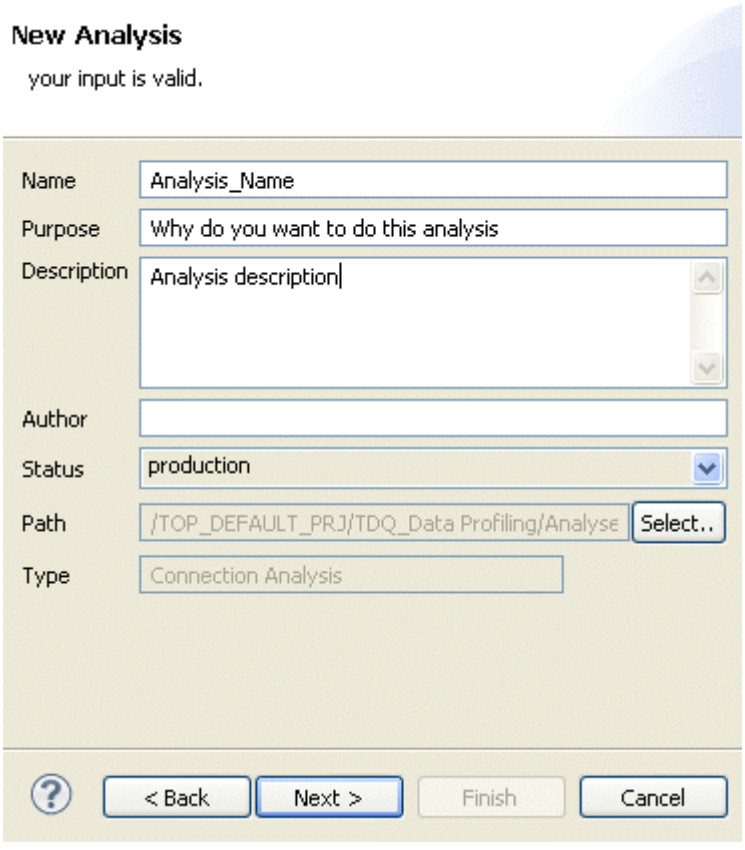
1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



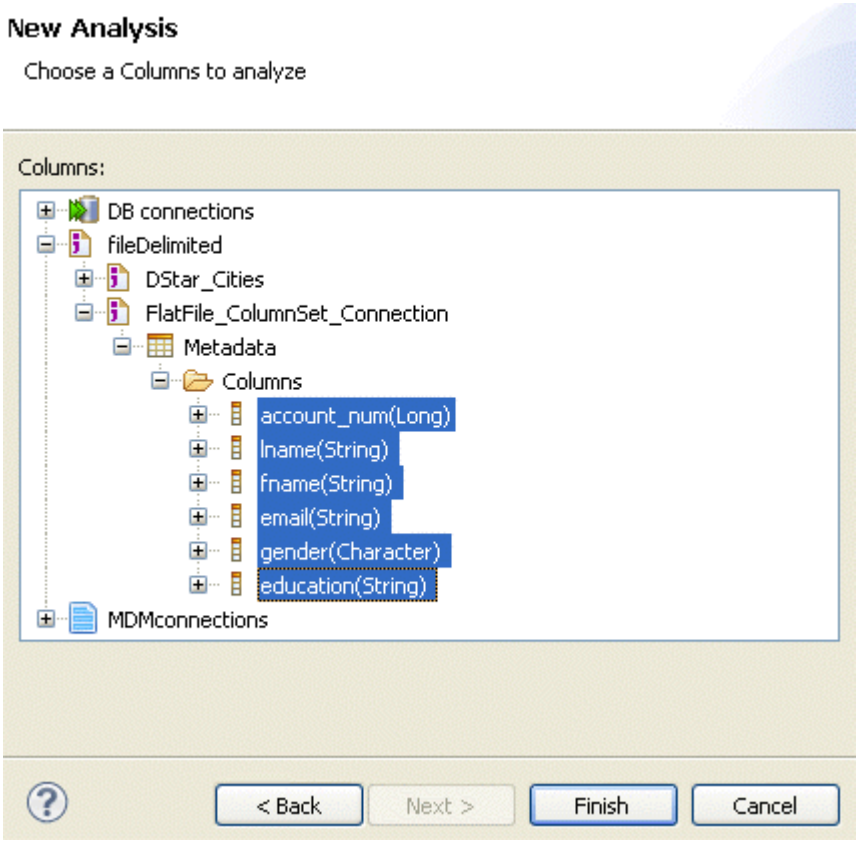
3. Expand the **Table Analysis** folder and click **Column Set Analysis**.
4. Click the **Next** button to proceed to the next step.



5. In the **Name** field, enter a name for the current analysis.

Space is not acceptable when typing in the analysis name in this field.

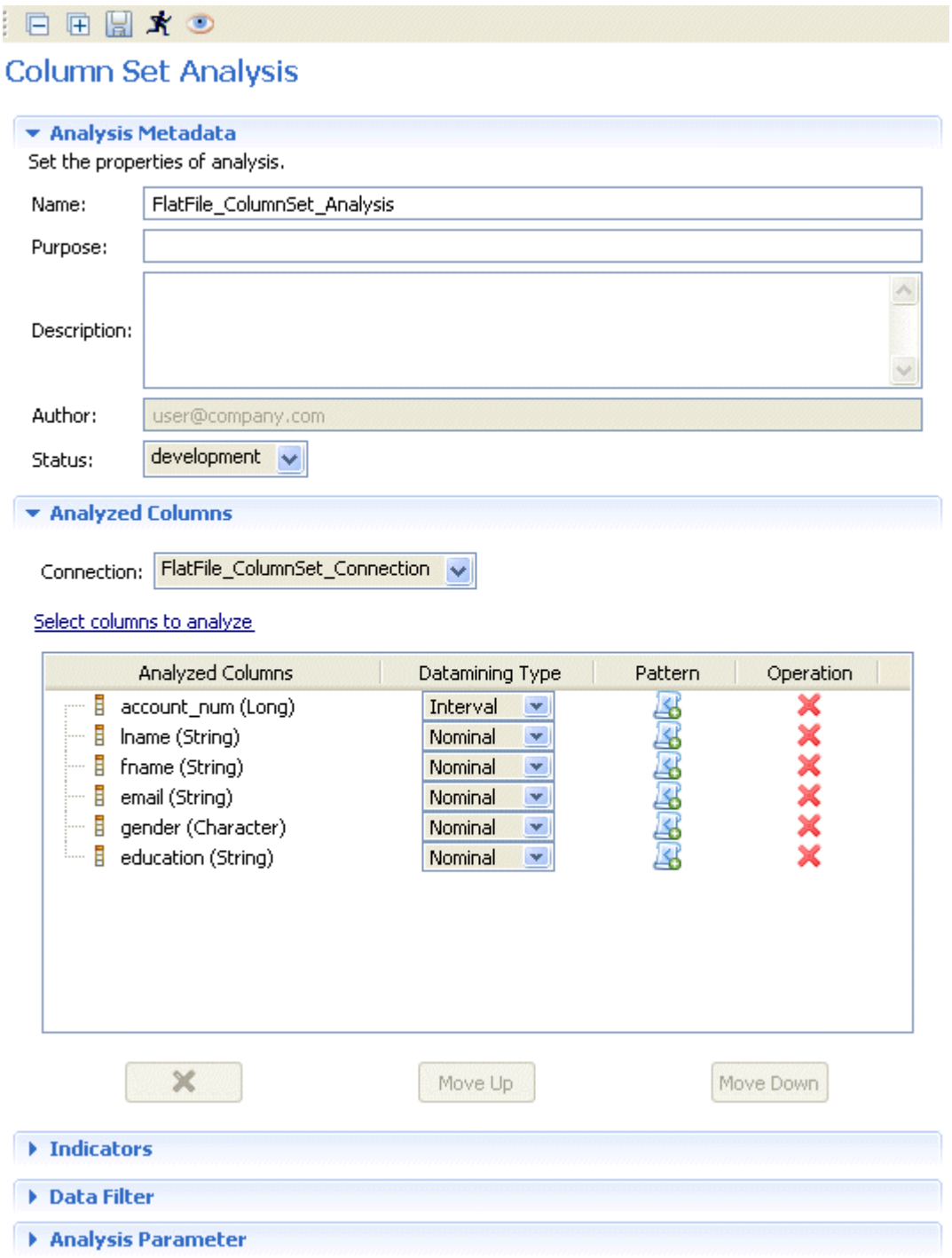
6. If required, set column analysis metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



7. Expand the **FileDelimited** connection and browse to the set of columns you want to analyze.

8. Select the columns to be analyzed, and then click **Finish** to close this **[New analysis]** wizard.

   The analysis editor opens with the defined analysis metadata, and a folder for the newly created analysis is displayed under **Analysis** in the **DQ Repository** tree view.

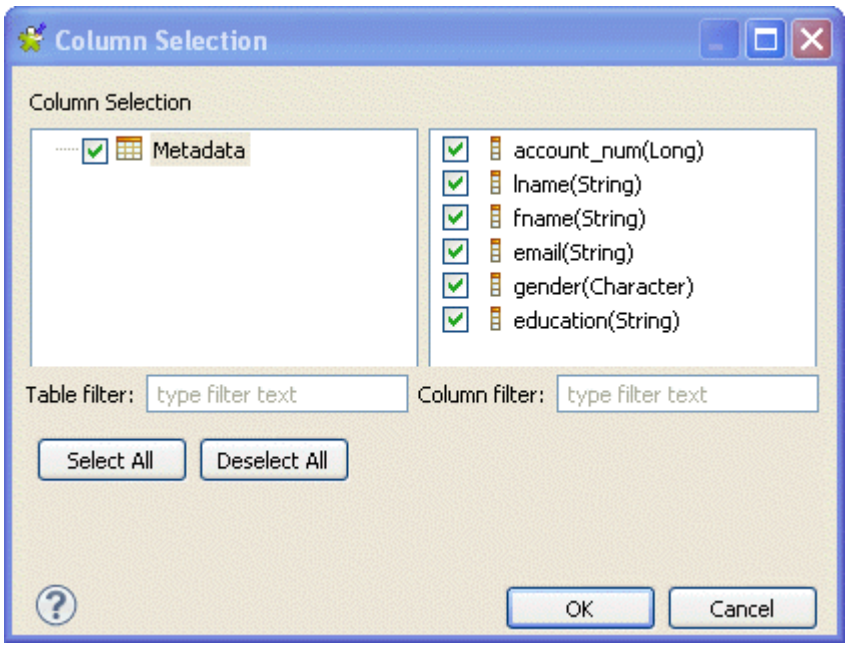


The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

9. If required, select another connection from the **Connection** box in the **Analyzed Columns** view. This box lists all the connections created in the Studio with the corresponding database names.

   By default, the delimited file connection you have selected in the previous step is displayed in the **Connection** box.

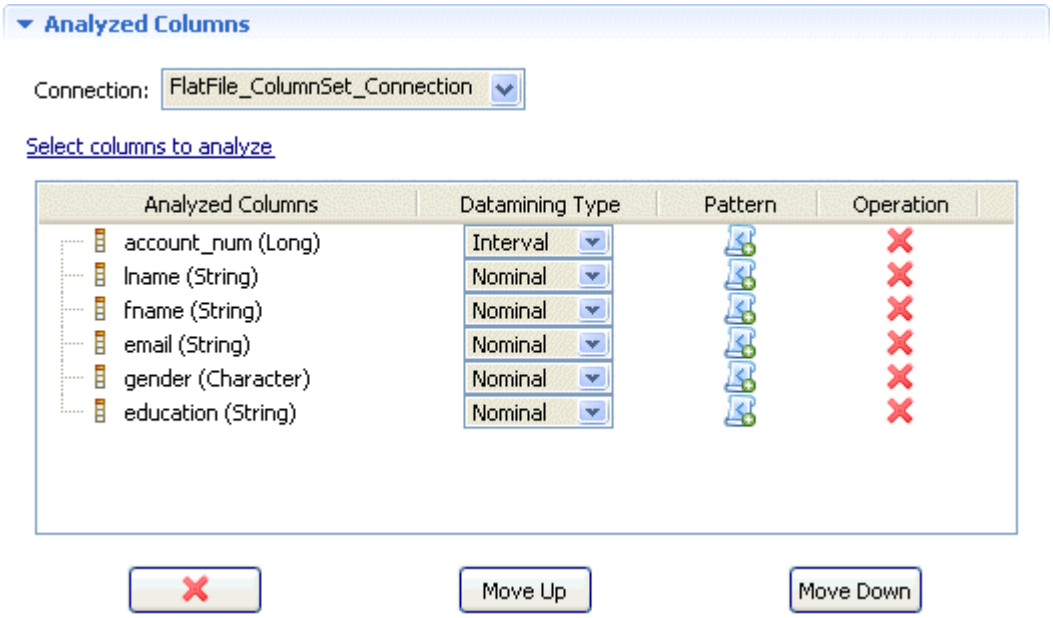
10. If required, click the **Select columns to analyze** link to open a dialog box where you can modify your column selection.



You can filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

11. In the column list, select the check boxes of the column(s) you want to analyze and click **OK** to proceed to the next step.

    In this example, you want to analyze a set of six columns in the delimited file: account number (*account_num*), education (*education*), email (*email*), first name (*fname*), second name (*lname*) and gender (*gender*). You want to identify the number of rows, the number of distinct and unique values and the number of duplicates.



12. If required, use the delete, move up or move down buttons to manage the analyzed columns.

If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

6.3.1.2. How to add patterns to the analyzed columns in the delimited file

Now, you can add patterns to one or more of the analyzed columns to validate the full record (all columns) against all the patterns, and not to validate each column against a specific pattern as it is the case with the column analysis. The results chart is a single bar chart for the totality of the used patterns. This chart shows the number of the rows that match “all” the patterns.

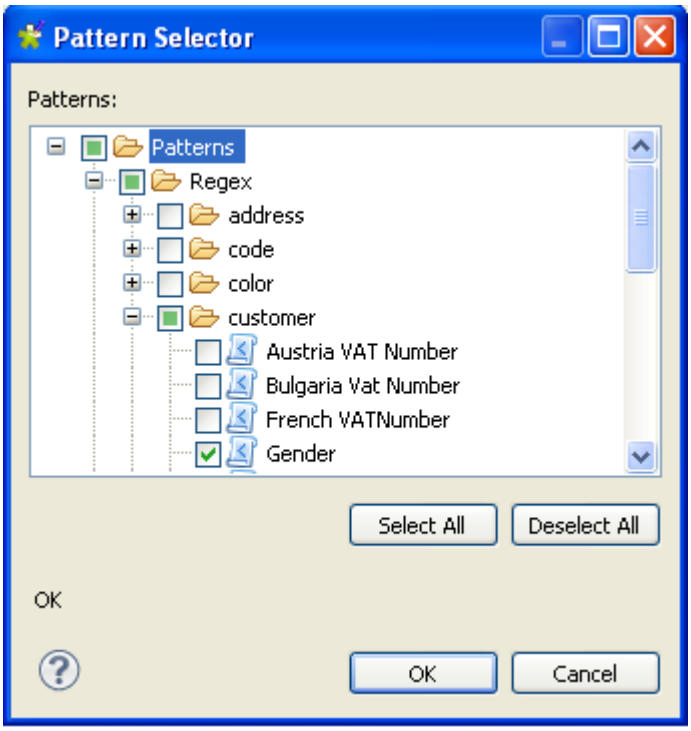
Before being able to use a specific pattern with a set of columns analysis, you must manually set in the patterns settings the pattern definition for Java, if it does not already exist. Otherwise, a warning message will display prompting you to set the definition of the Java regular expression.

Prerequisite(s): An analysis of a set of columns is open in the analysis editor in the studio. For more information, see section *How to define the set of columns to be analyzed*.

To add patterns to the analysis of a set of columns, do the following:

1. Click the icon next to each of the columns you want to validate against a specific pattern.

   The **[Pattern Selector]** dialog box is displayed.



   You can add only regular expressions to the analyzed columns.

   You can drop the regular expression directly from the **Patterns** folder in the **DQ Repository** tree view directly to the column name in the column analysis editor.

   If no Java expression exists for the pattern you want to add, a warning message will display prompting you to add the pattern definition for Java. Click Yes to open the pattern editor and add the Java regular expression, then proceed to add the pattern to the analyzed columns.

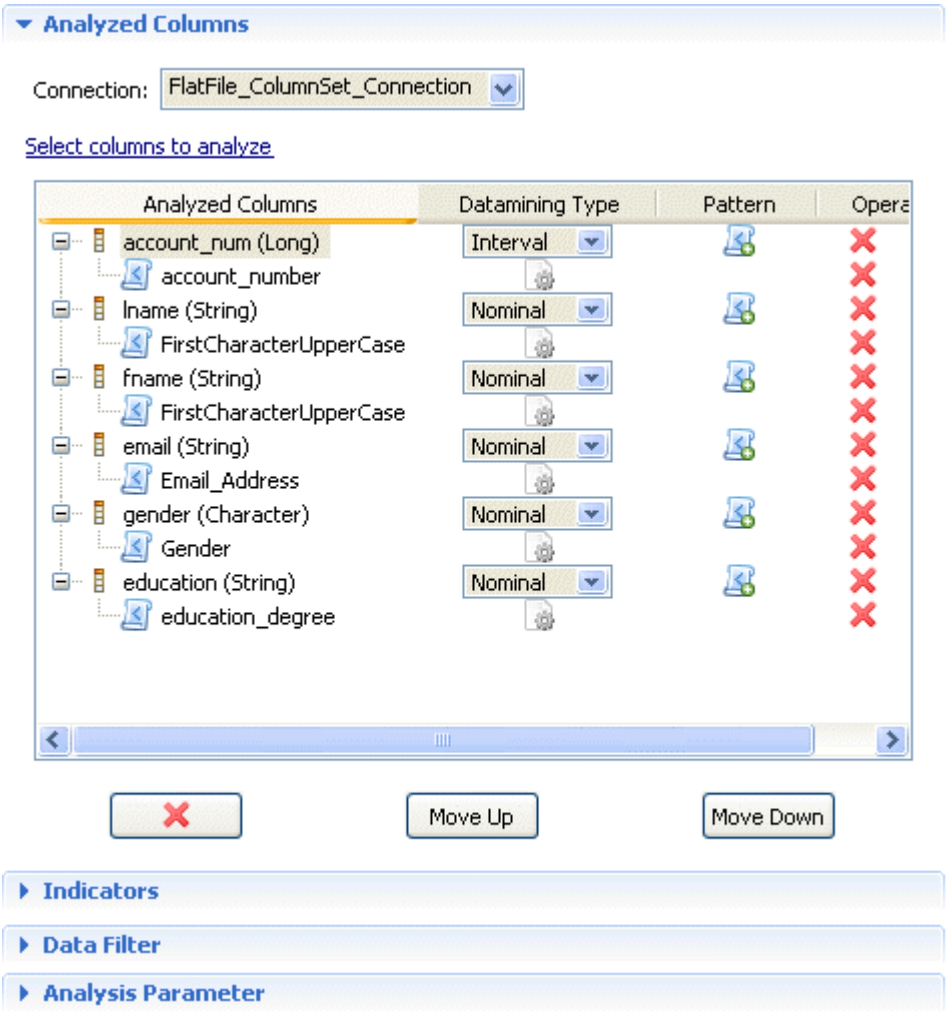
   In this example, you want to add a corresponding pattern to each of the analyzed columns to validate data in these columns against the selected patterns. The result chart will show the percentage of the matching/non-matching values, the values that respect the totality of the used patterns.

2. In the **[Pattern Selector]** dialog box, expand **Patterns** and browse to the regular expression you want to add to the selected column.

3. Select the check box(es) of the expression(s) you want to add to the selected column.

4. Click **OK** to proceed to the next step.

   The added regular expression(s) display(s) under the analyzed column(s) in the **Analyzed Columns** view and the `All Match` indicator is displayed in the **Indicators** list in the **Indicators** view.

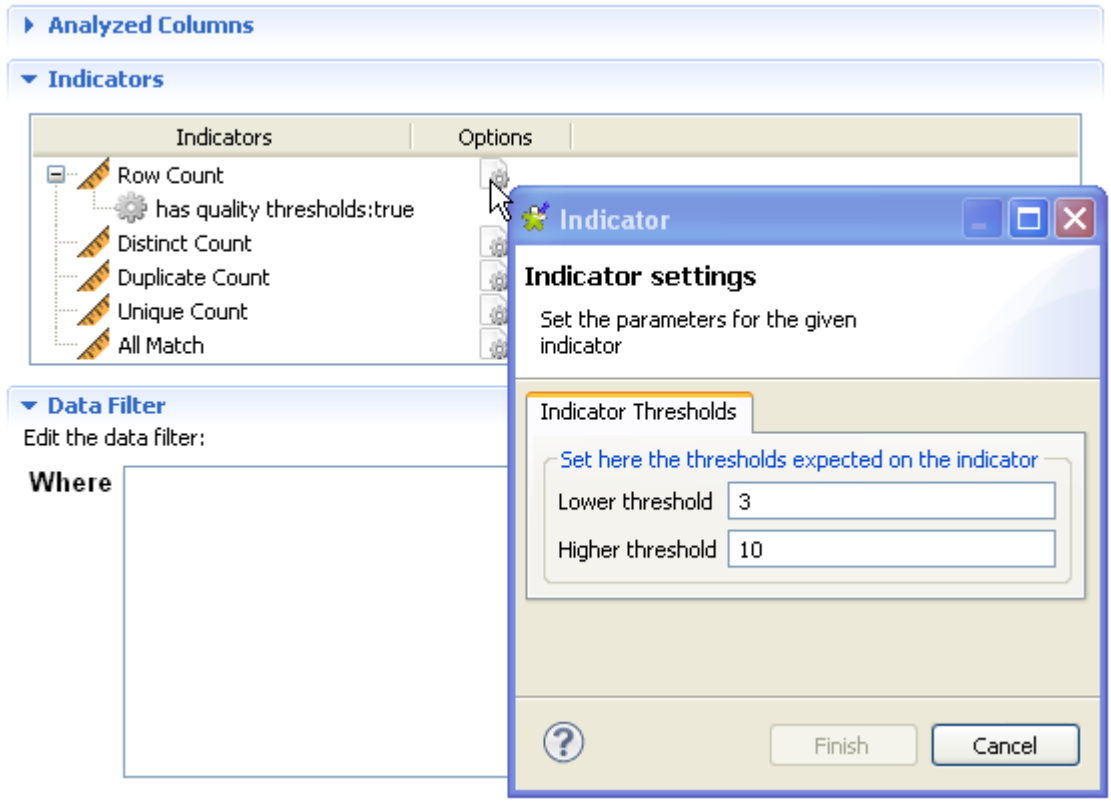


6.3.1.3. How to finalize and execute the column set analysis on a delimited file

What is left before executing this set of columns analysis is to define indicators, data filter and analysis parameters.

Prerequisite(s):A column set analysis is defined in the **Profiling** perspective of the studio. For further information, see section *How to define the set of columns to be analyzed in a delimited file* and section *How to add patterns to the analyzed columns in the delimited file*.

1. Click **Indicators** in the analysis editor to open the corresponding view.

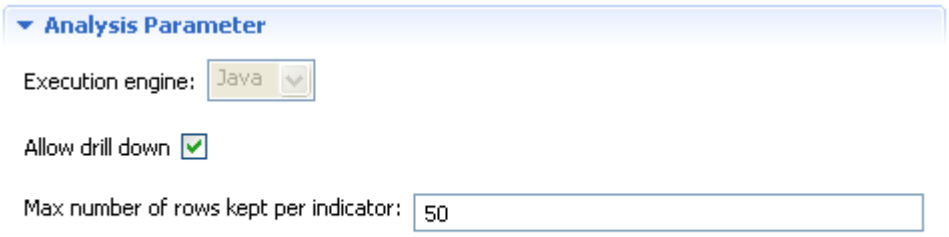


The indicators representing the simple statistics are by-default attached to this type of analysis. For further information about the indicators for simple statistics, see section section *Simple statistics*.

2. If required, click the option icon to open a dialog box where you can set options for each indicator. For more information about indicators management, see section *Indicators*.

3. If required, click **Data Filter** in the analysis editor to display its view and filter data through SQL “WHERE” clauses.

4. In the **Analysis Parameters** view, select the **Allow drill down** check box to store locally the data that will be analyzed by the current analysis.

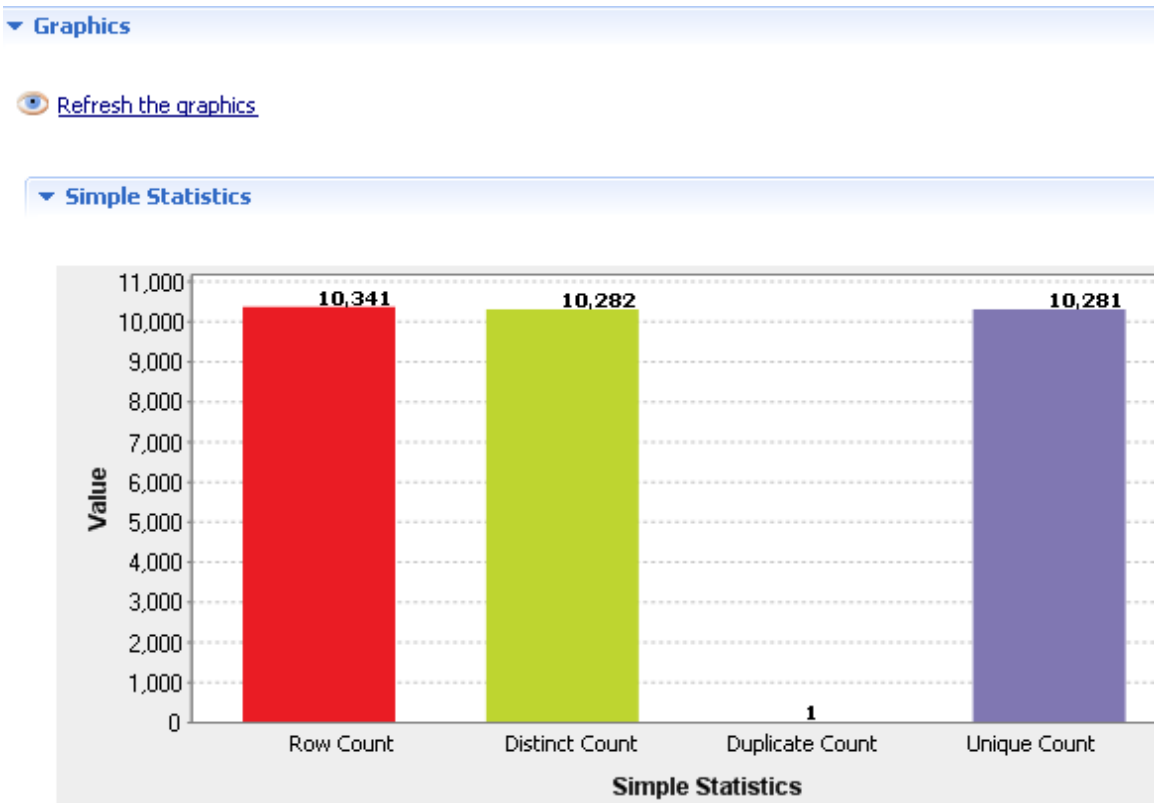


5. In the **Max number of rows kept per indicator** field enter the number of the data rows you want to make accessible.

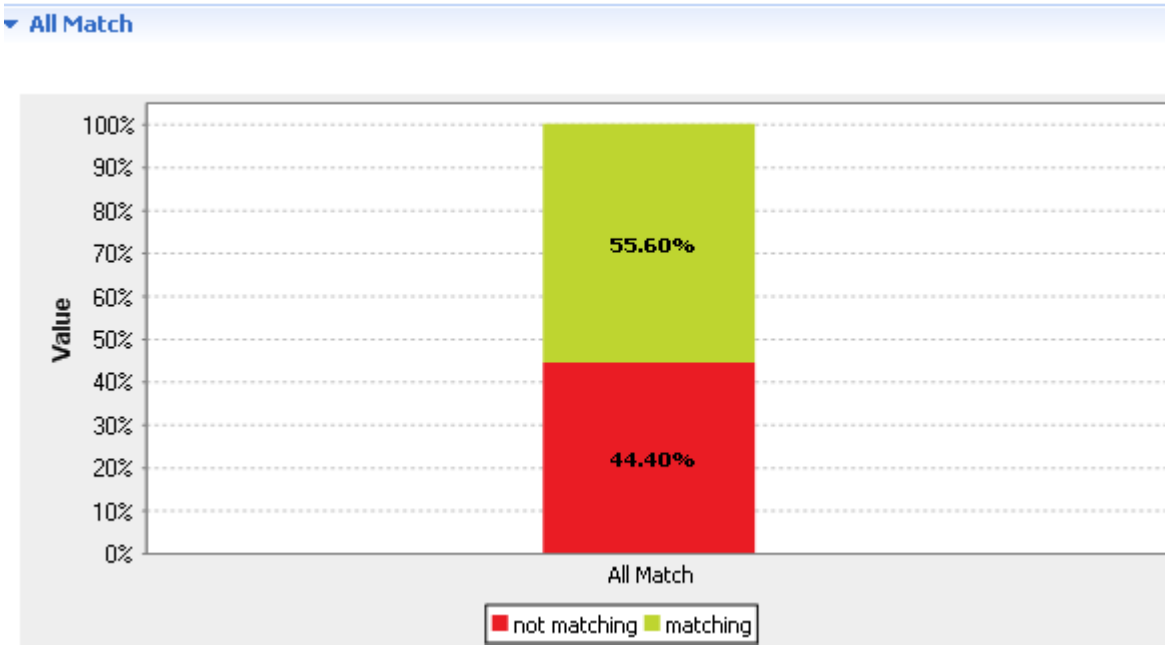
The **Allow drill down** check box is selected by default, and the maximum analyzed data rows to be shown per indicator is set to 50.

6. Click the save icon on top of the analysis editor and then press **F6** to execute the analysis.

   The **Graphics** panel to the right of the analysis editor displays the graphical result corresponding to the Simple Statistics indicators used to analyze the defined set of columns.



When you use patterns to match the content of the columns to be analyzed, another graphic is displayed to illustrates the match results against the totality of the used patterns.



6.3.1.4. How to access the detailed result view for the delimited file analysis

The procedure to access the detailed results for the delimited file analysis is the same as that for the database analysis. For further information, see section *How to access the detailed result view*.

6.3.1.5. How to filter analysis data against patterns

The procedure to filter the data of the analysis of a delimited file is the same as that for the database analysis. For further information, see section *How to filter data against patterns*.

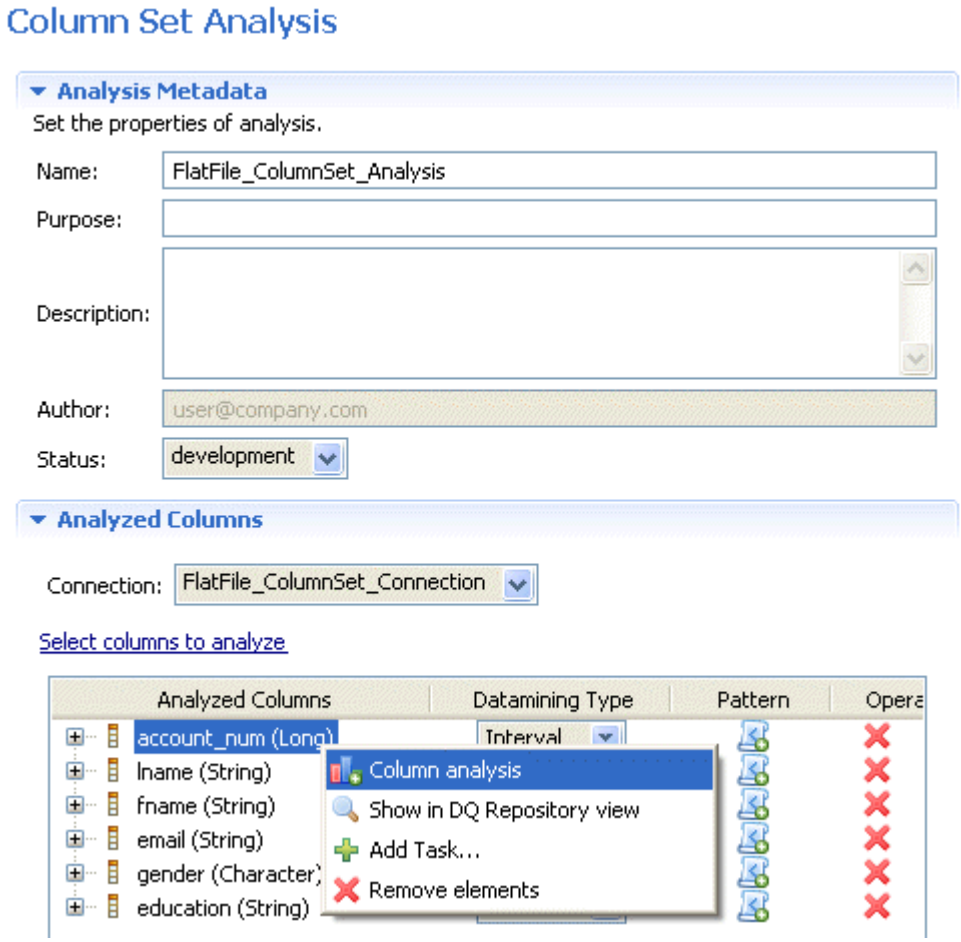
6.3.2. Creating a column analysis from the analysis of a set of columns

You can create a column analysis on one or more columns defined in the set of columns analysis.

Prerequisite(s): A simple table analysis is defined in the analysis editor in the **Profiling** perspective of the studio.

To create a column analysis on one or more columns defined in the set of columns analysis, do the following:

1. Open the set of columns analysis.
2. In the **Analyzed Columns** view, right-click the column(s) you want to create a column analysis on.



3. Select **Column analysis** from the contextual menu. The **[New Analysis]** wizard opens.
4. In the **Name** field, enter a name for the new column analysis and then click **Next** to proceed to the next step.

   The analysis editor opens with the defined metadata and a folder for the newly created analysis is displayed under the **Analyses** folder in the **DQ Repository** tree view.
5. Follow the steps outlined in section *Analyzing columns in a delimited file* to continue creating the column analysis on a delimited file.

6.4. Analyzing tables on MDM servers

You can analyze the content of a set of columns “attributes” in a specific table “entity” on the MDM server. This set can represent only some of the attributes in the defined entity or the entity as a whole.

You can then execute the created analysis using the Java engine.

6.4.1. Creating a column set analysis on an MDM server

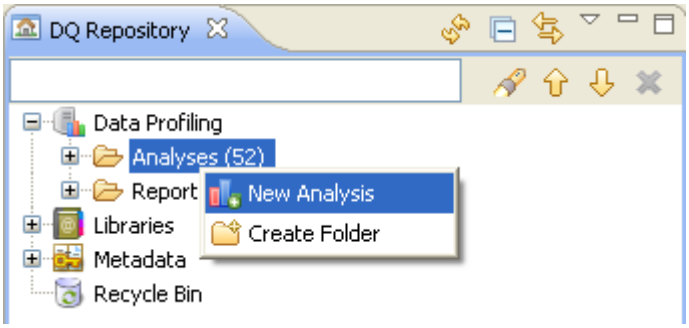
This type of analysis provide simple statistics on the number of records falling in certain categories, including the number of rows, the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields. For more information about these indicators, see section *Simple statistics*.

6.4.1.1. How to define the set of columns to be analyzed on the MDM server

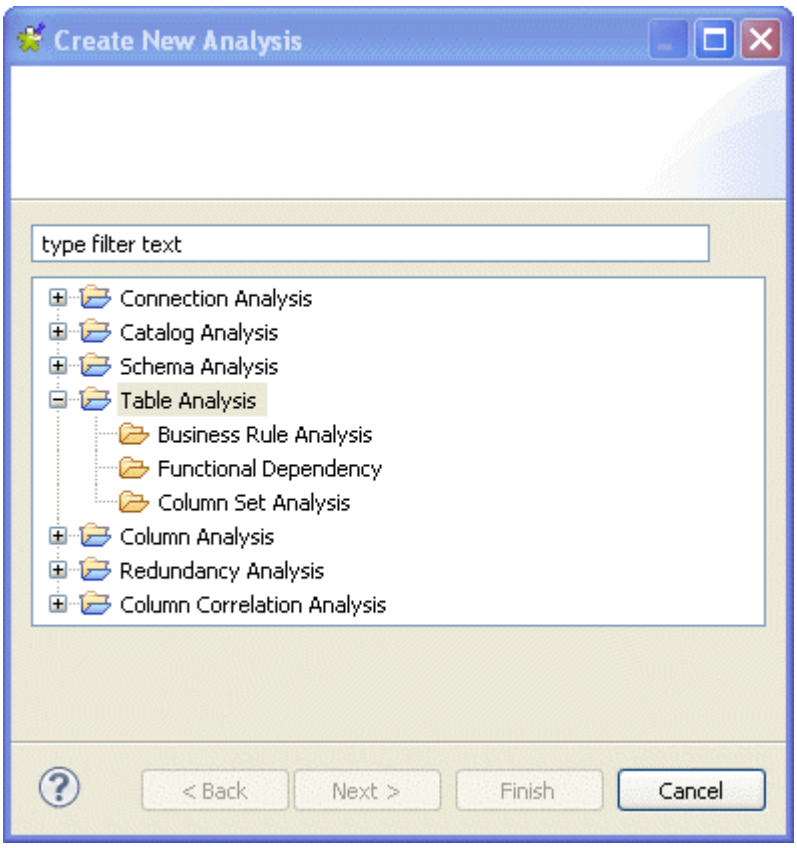
Prerequisite(s): At least one connection to an MDM server is set in the studio. For further information, see section *Connecting to an MDM server*.

To define the set of columns “attributes” to be analyzed, do the following:

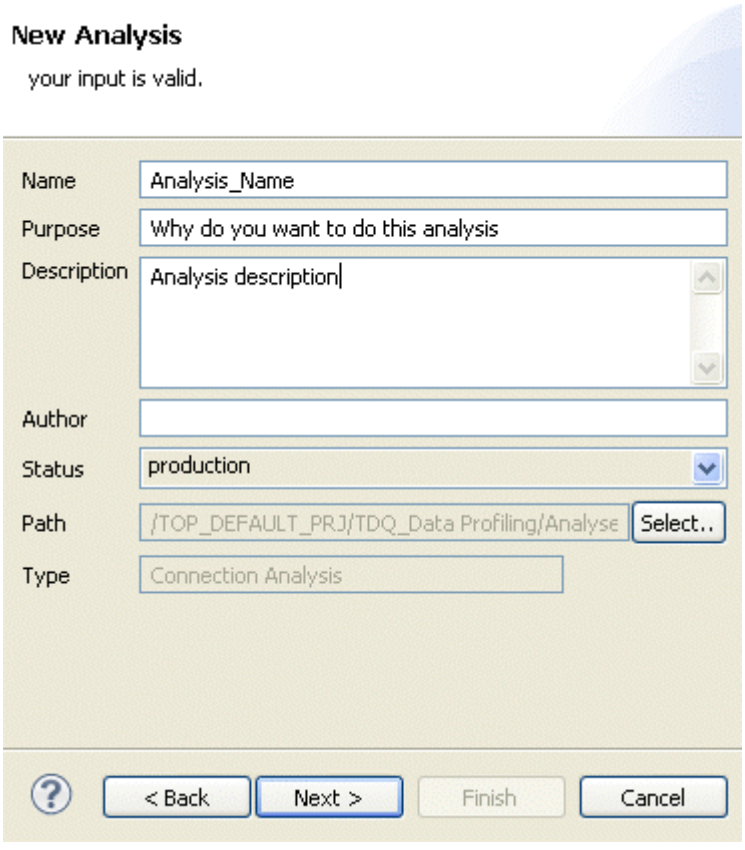
1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



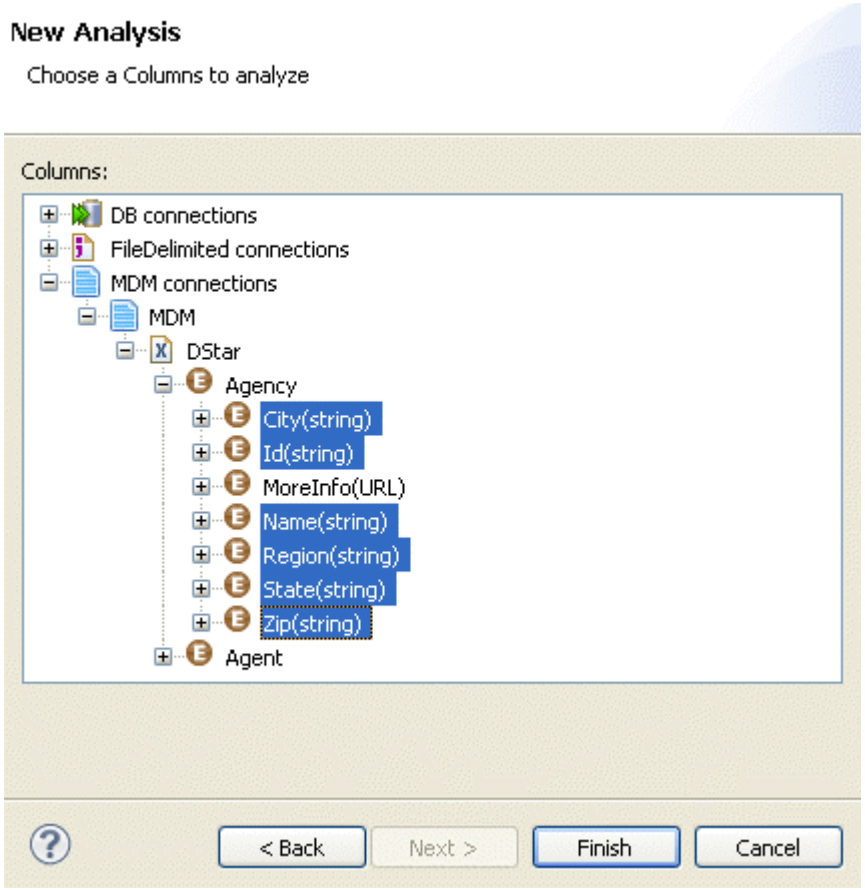
3. Expand the **Table Analysis** folder and click **Column Set Analysis**.
4. Click the **Next** button to proceed to the next step.



5. In the **Name** field, enter a name for the current analysis.

Space is not acceptable when typing in the analysis name in this field.

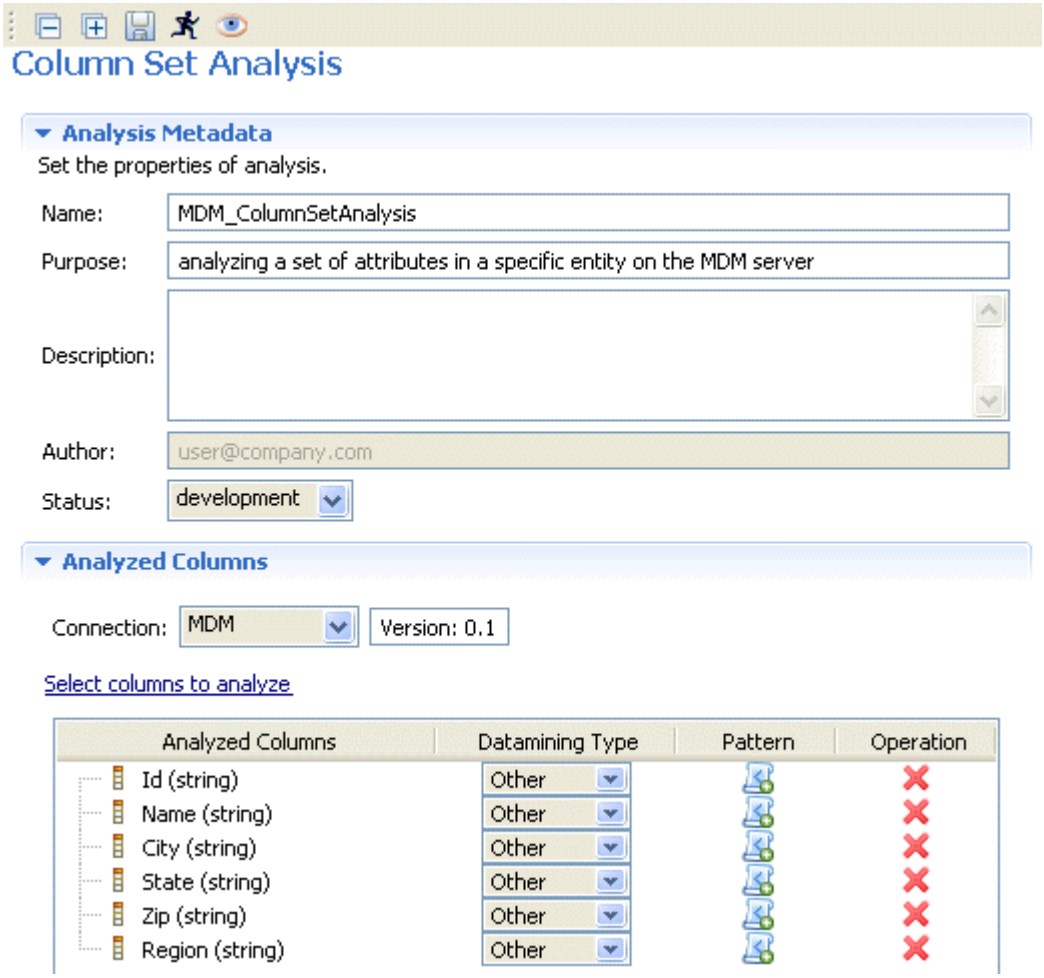
6. If required, set column analysis metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



7. Expand **MDM connections** and browse to the set of columns “attributes” you want to analyze.

8. Select the attributes to be analyzed, and then click **Finish** to close this **[New analysis]** wizard.

   The analysis editor opens with the defined analysis metadata, and a folder for the newly created analysis is displayed under **Analysis** in the **DQ Repository** tree view.

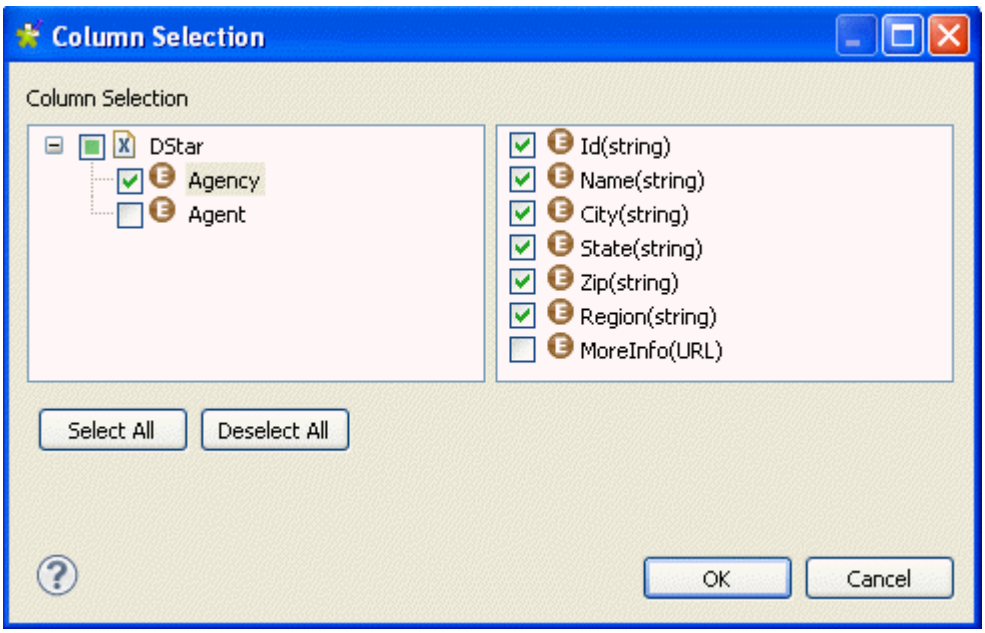


The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

9. If required, select another connection from the **Connection** box in the **Analyzed Columns** view. This box lists all the connections created in the Studio with the corresponding database names.

   By default, the connection you have selected in the previous step is displayed in the **Connection** box.

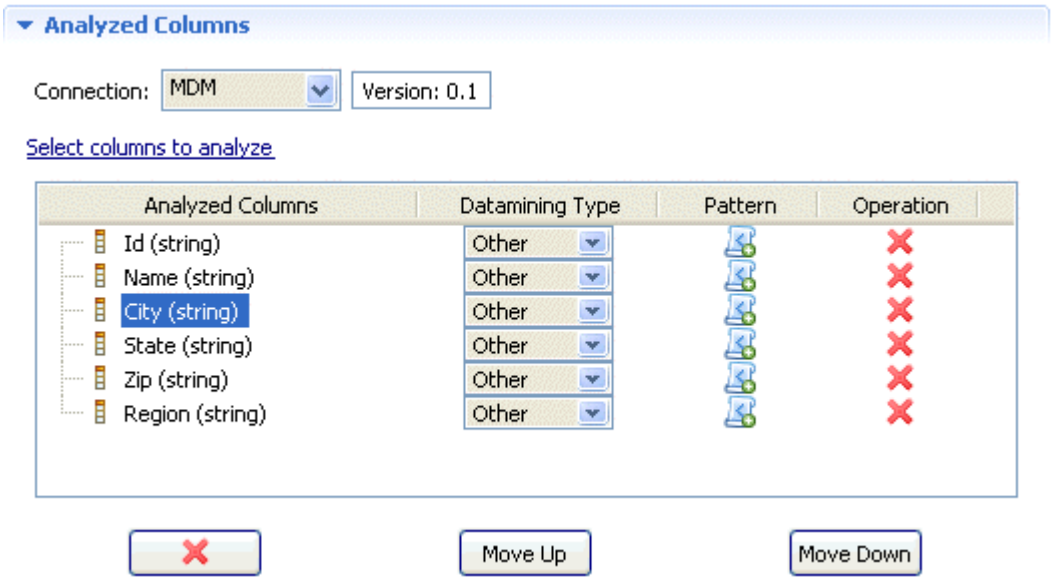
10. If required, click the **Select columns to analyze** link to open a dialog box where you can modify your column selection.



When carrying out this type of analysis, the set of columns to be analyzed must not include a primary key column.

11. In the column list, select the check boxes of the attributes you want to analyze and click **OK** to proceed to the next step.

    Selected attributes are listed in the **Analyzed Columns** view.



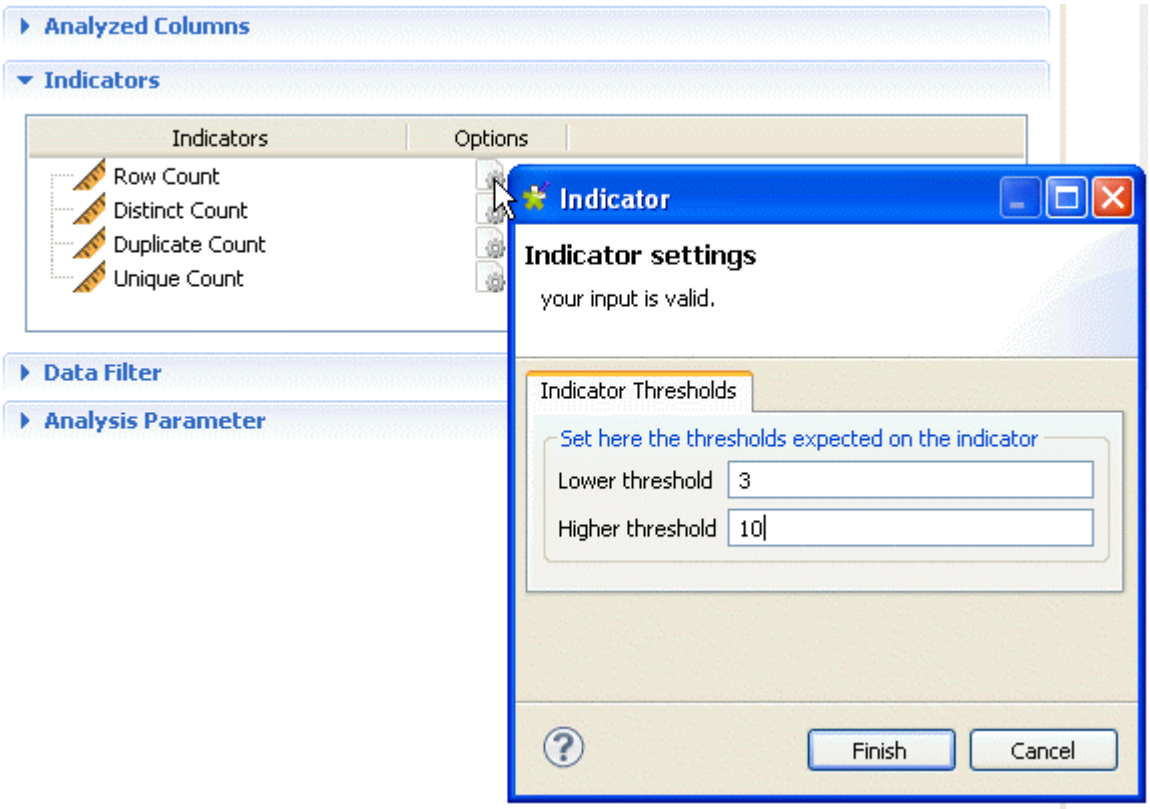
12. If required, use the delete, move up or move down buttons to manage the analyzed columns.

6.4.1.2. How to finalize and execute the analysis of a set of columns on a delimited file

What is left before executing this set of columns analysis is to define indicators and analysis parameters.

Prerequisite(s):A column set analysis has been defined in the **Profiling** perspective of the studio. For further information, see section *How to define the set of columns to be analyzed on the MDM server*.

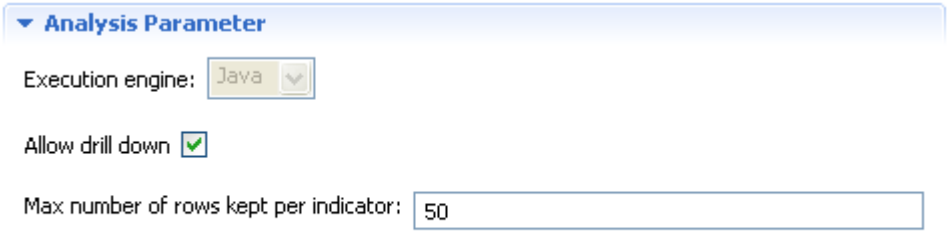
1. Click **Indicators** in the analysis editor to open the corresponding view.



The indicators representing the simple statistics are by-default attached to this type of analysis. For further information about the indicators for simple statistics, see section section *Simple statistics*.

2. If required, click the option icon to open a dialog box where you can set options for each indicator. For more information about indicators management, see section *Indicators*.

3. In the **Analysis Parameters** view, select the **Allow drill down** check box to store locally the data that will be analyzed by the current analysis.



4. In the **Max number of rows kept per indicator** field enter the number of the data rows you want to make accessible.

The **Allow drill down** check box is selected by default, and the maximum analyzed data rows to be shown per indicator is set to 50.

5. Click the save icon on top of the analysis editor and then press **F6** to execute the analysis.

   The **Graphics** panel to the right of the analysis editor displays the graphical result corresponding to the Simple Statistics indicators used to analyze the defined set of columns.

6.4.1.3. How to access the detail result view

The procedure to access the detail results for the column set analysis on an MDM server is the same as that for the same analysis on databases. For further information, see section *How to access the detailed result view*.

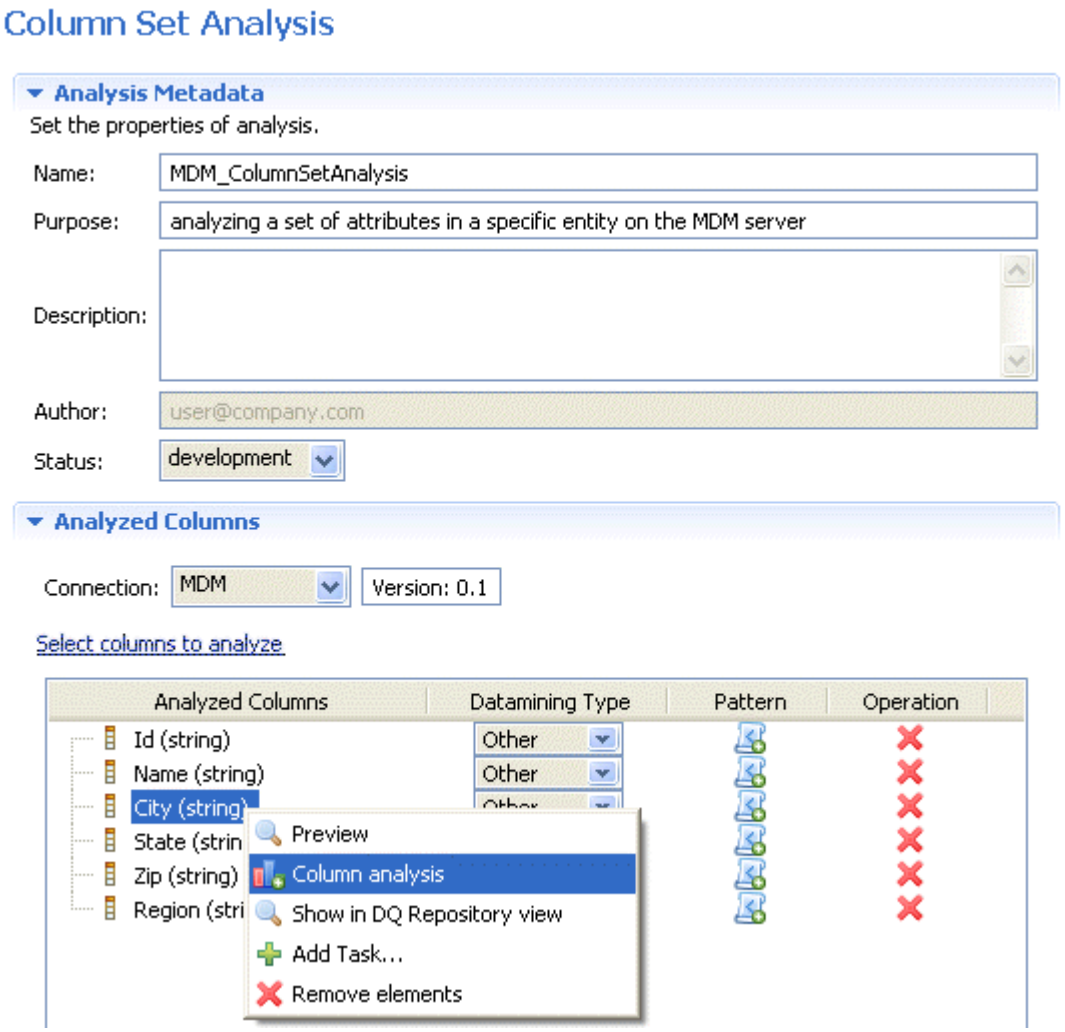
6.4.2. Creating a column analysis from the column set analysis

You can create a column analysis on one or more columns defined in the set of columns analysis.

Prerequisite(s): A column set analysis has been defined in the **Profiling** perspective of the studio. For further information, see section *How to define the set of columns to be analyzed on the MDM server*.

To create a column analysis on one or more columns defined in the column set analysis, do the following:

1. Open the column set analysis.
2. In the **Analyzed Columns** view, right-click the column(s) you want to create a column analysis on.



3. Select **Column analysis** from the contextual menu. The **[New Analysis]** wizard opens.
4. In the **Name** field, enter a name for the new column analysis and then click **Next** to proceed to the next step. The analysis editor opens with the defined metadata and a folder for the newly created analysis is displayed under the **Analyses** folder in the **DQ Repository** tree view.

5. Follow the steps outlined in section *Analyzing master data on an MDM server* to continue creating the column analysis on a delimited file.

Chapter 7. Redundancy analysis

This chapter provides all the information you need to perform redundancy analysis that can compare table content or identify overlapping values between two sets of columns.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

7.1. What are redundancy analyses

Redundancy analyses are column comparison analyses that better explore the relationships between tables through:

- Comparing identical columns in different tables,
- Matching foreign keys in one table to primary keys in the other table and vice versa.

The sections below provide detailed information about these two types of redundancy analyses.

The number of the analyses created in the **Profiling** perspective of the studio is indicated next to the **Analyses** folder in the **DQ Repository** tree view.

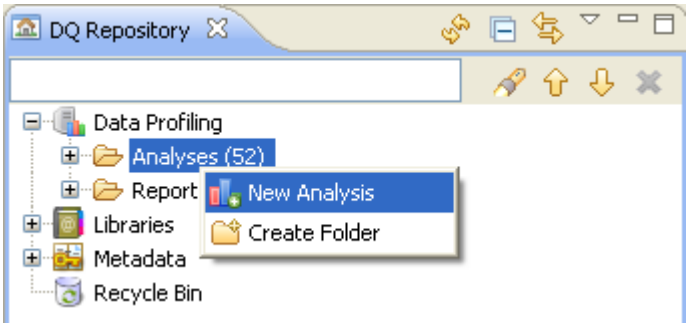
7.2. Comparing identical columns in different tables

From your studio, you can create an analysis that compares two identical sets of columns in two different tables.

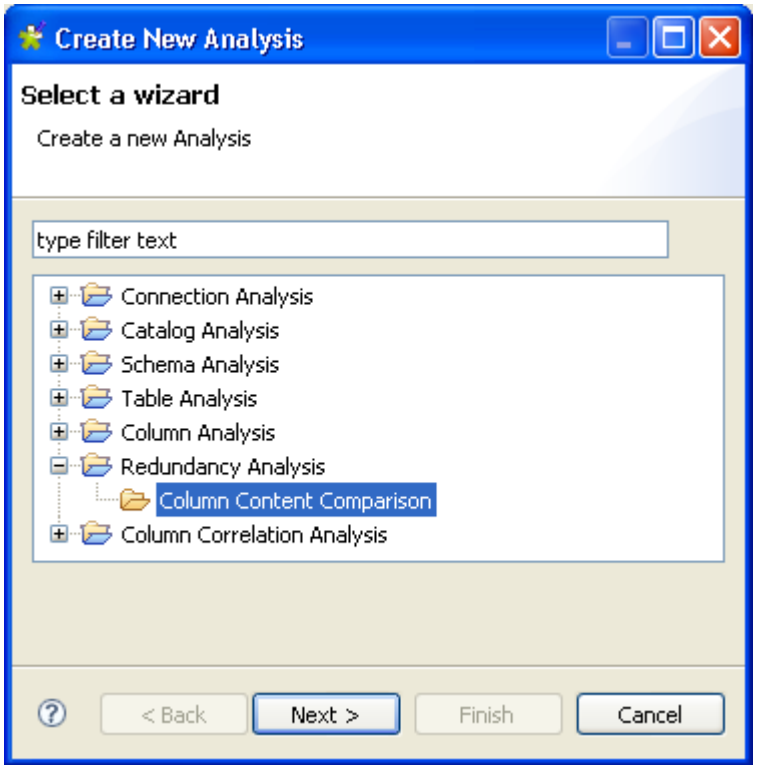
Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

Defining the analysis

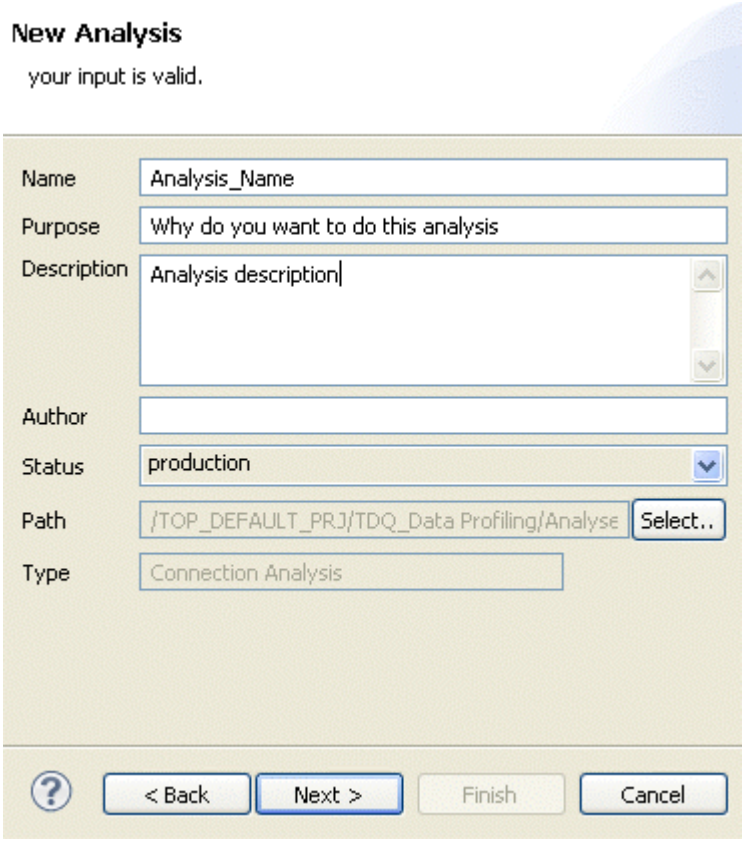
1. In the **DQ Repository** tree view, expand **Data Profiling**.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.

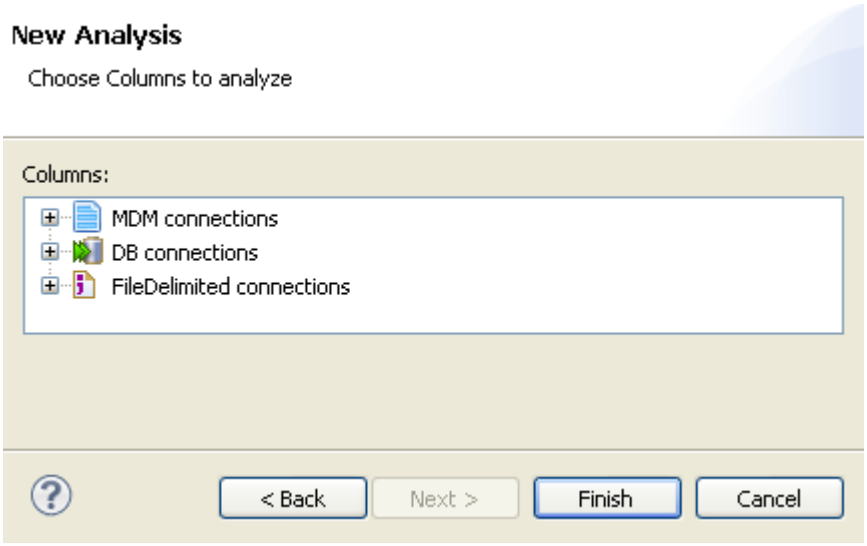


3. Expand the **Redundancy Analysis** node and then select **Column Content Comparison**.
4. Click **Next**.



5. In the **Name** field, enter a name for the current analysis.

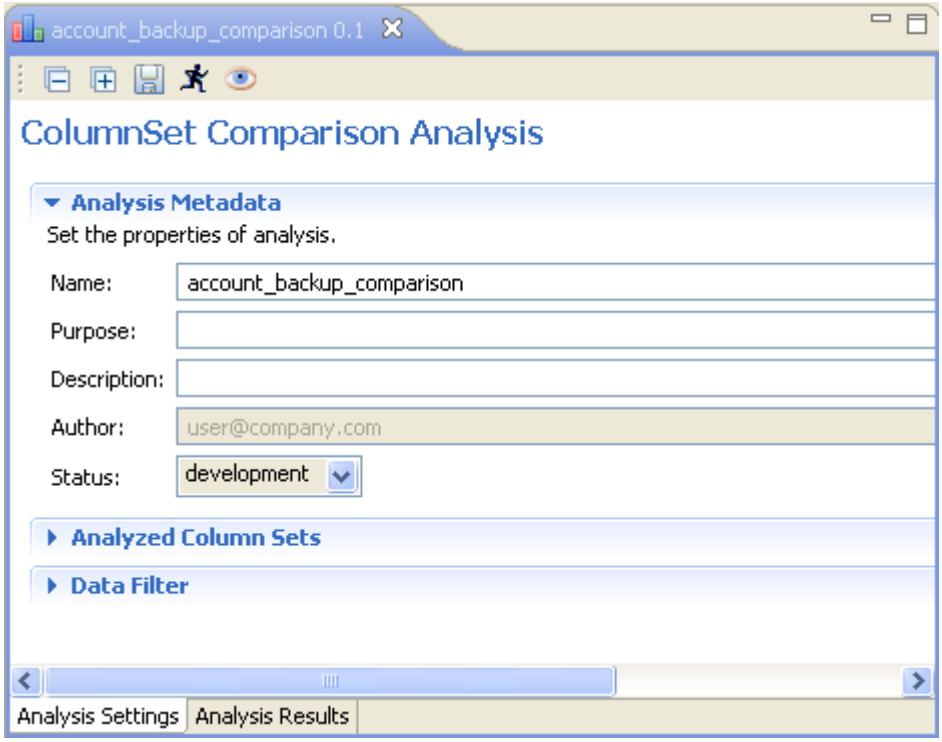
6. Set the analysis metadata (purpose, description and author name) in the corresponding fields and then click **Next**.



Selecting the identical columns you want to compare

1. Expand **DB connections** and in the desired database, browse to the columns you want to analyze, select them and then click **Finish** to close the wizard.

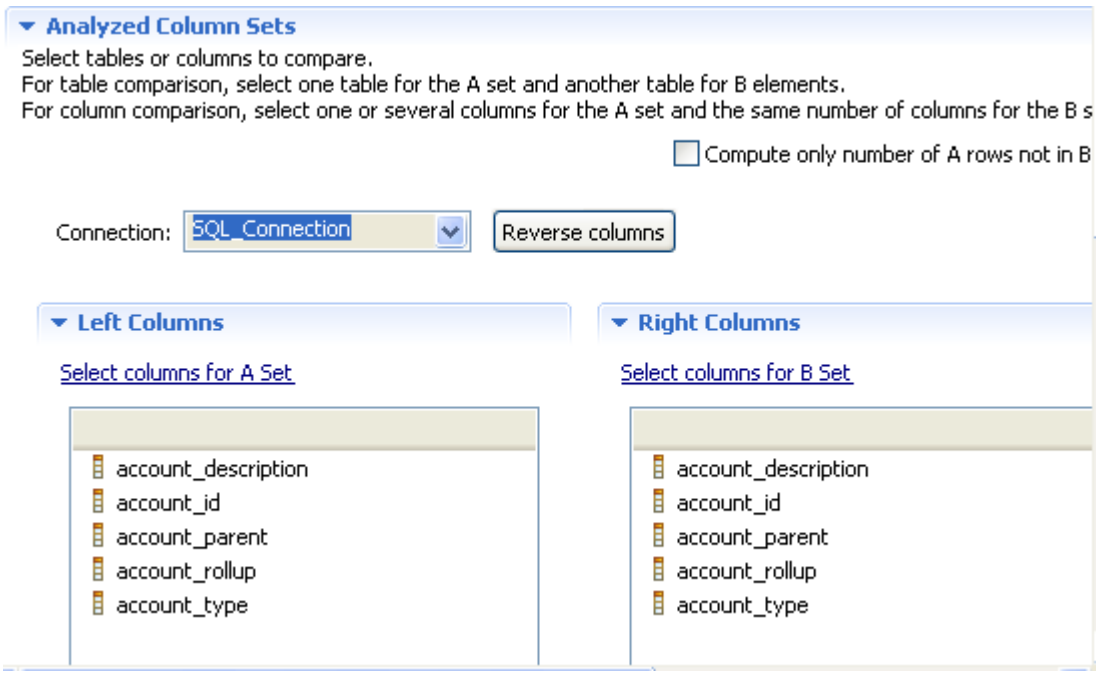
   A file for the newly created analysis is listed under the **Analysis** folder in the **DQ Repository** tree view. The analysis editor opens with the defined analysis metadata.



   The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

2. Click **Analyzed Column Sets** to open the view where you can set the columns or modify your selection.

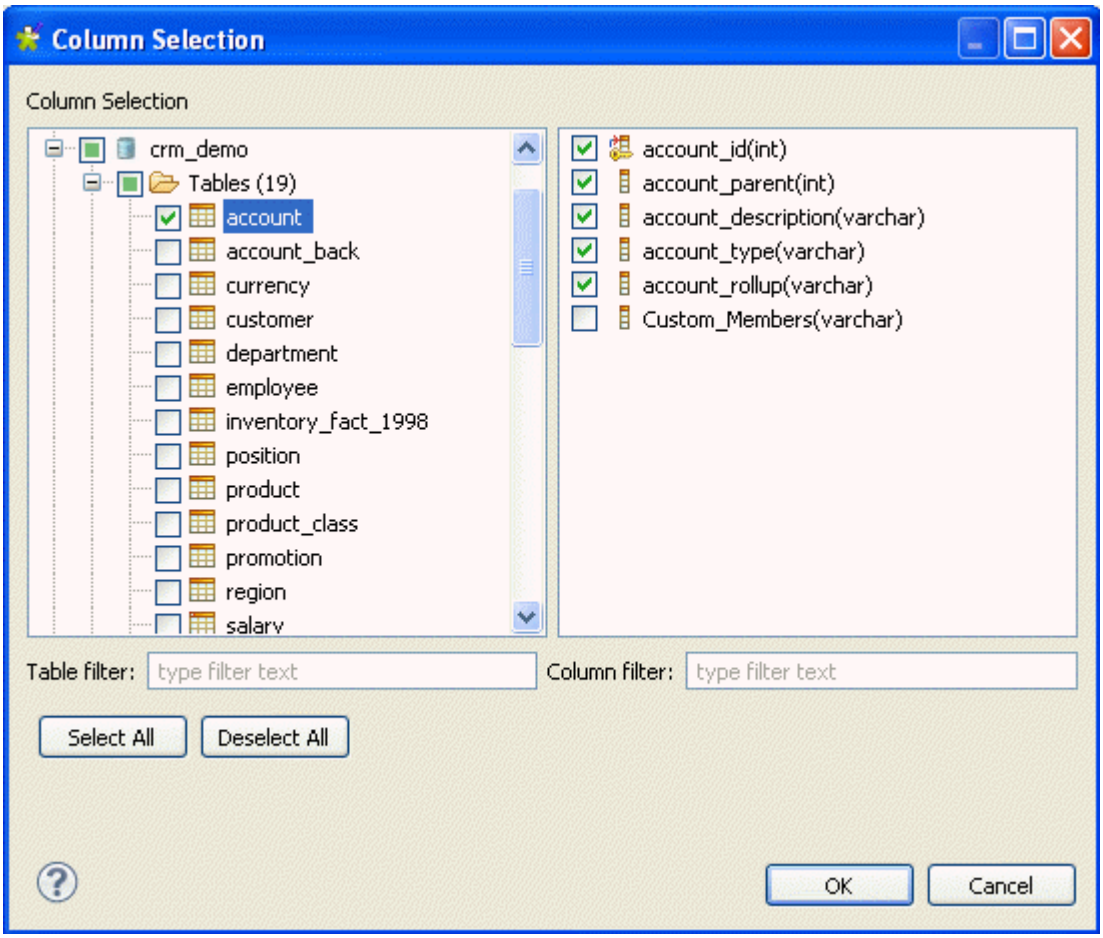
   In this example, you want to compare identical columns in the *account* and *account_back* tables.



3. From the **Connection** box, select the database connection relevant to the database to which you want to connect.

   This box lists all the connections created in the Studio with the corresponding database names.

4. Click **Select columns for the A set** to open the **[Column Selection]** dialog box.



5. Expand **DB Connections** and then browse through the catalogs/schemas to reach the table holding the columns you want to analyze.

   You can filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

6. Click the table name to list all its columns in the right-hand panel of the **[Column Selection]** dialog box.

7. In the list to the right, select the check boxes of the column(s) you want to analyze and click **OK** to proceed to the next step.

   You can drag the columns to be analyzed directly from the **DQ Repository** tree view to the editor.

   If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

8. Click **Select Columns from the B set** and follow the same steps to select the second set of columns or drag it to the right column panel.

9. Select the **Compute only number of A rows not in B** check box if you want to match the data from the A set against the data from the B set and not vice versa.

10. Click **Data Filter** in the analysis editor to open the view where you can set a filter on each of the column sets.

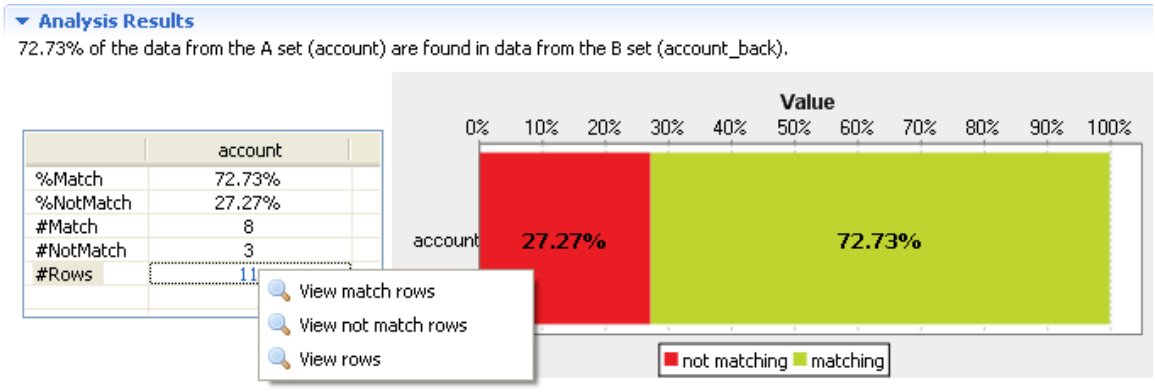
11. Click the save icon on top of the editor and then press **F6** to execute the column comparison analysis.

    A confirmation message is displayed.

12. Read the confirmation message and click **OK** if you want to continue the operation.

    The **Analysis Results** view opens showing the analysis results.

    In this example, 72.73% of the data present in the columns in the *account* table could be matched with the same data in the columns in the *account_back* table.



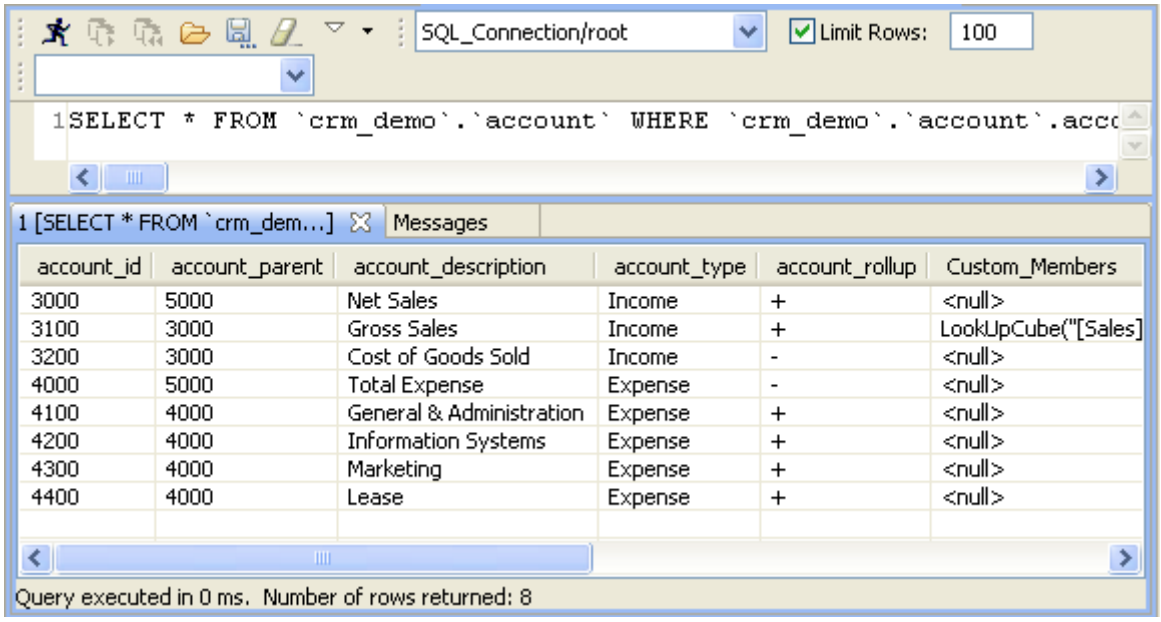
Through this view, you can also access the actual analyzed data via the Data Explorer.

To access the analyzed data rows, right-click any of the lines in the table and select:

Option	To...
View match rows	access a list of all rows that could be matched in the two identical column sets
View not match rows	access a list of all rows that could not be matched in the two identical column sets
View rows	access a list of all rows in the two identical column sets

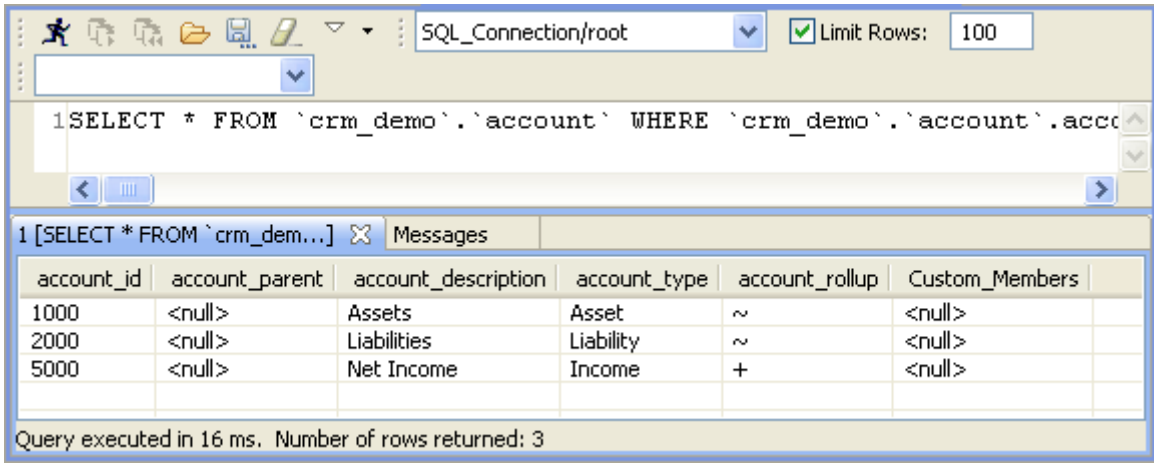
*The data explorer does not support connections which has empty user name, such as Single sign-on of MS SQL Server. If you analyze data using such connection and you try to view data rows in the **Data Explorer** perspective, a warning message prompt you to set your connection credentials to the SQL Server.*

The figure below illustrates the data explorer list of all rows that could be matched in the two sets, eight in this example.



From the SQL editor, you can save the executed query and list it under the **Libraries** > **Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see section *Saving the queries executed on indicators*.

The figure below illustrates the data explorer list of all rows that could not be matched in the two sets, three in this example.



For more information about the data explorer Graphical User Interface, see appendix *Data Explorer management GUI*.

7.3. Matching primary and foreign keys

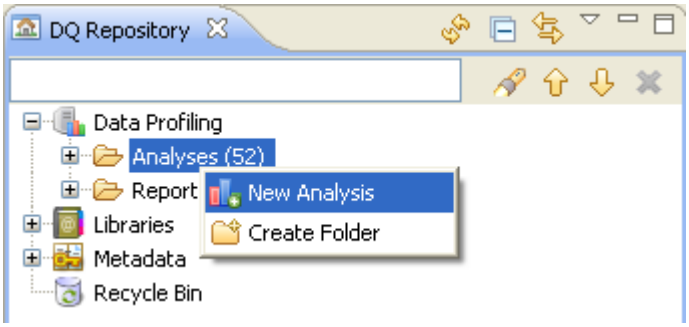
You can create an analysis that matches foreign keys in one table to primary keys in the other table and vice versa.

Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

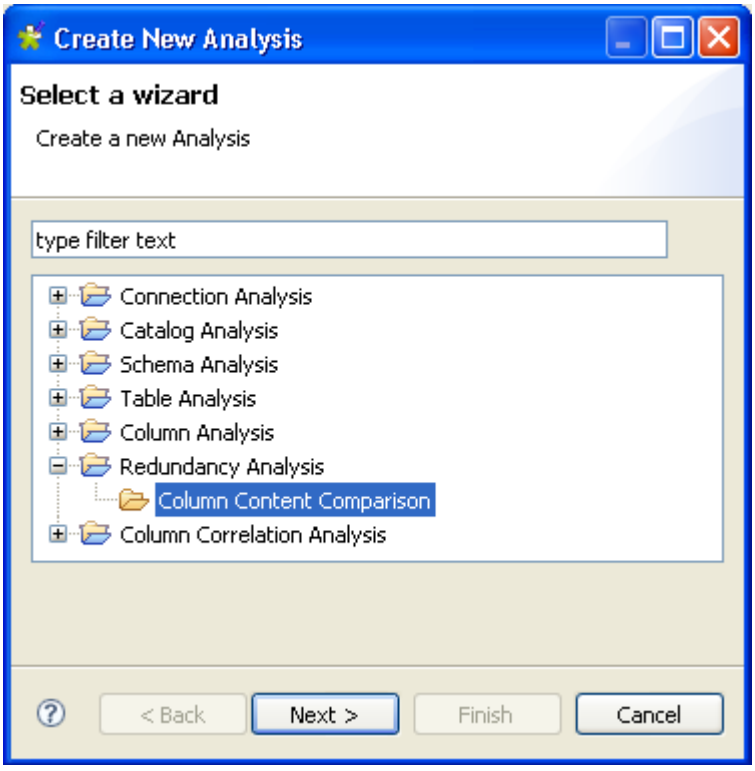
To match primary and foreign keys in tables, do the following:

Defining the analysis

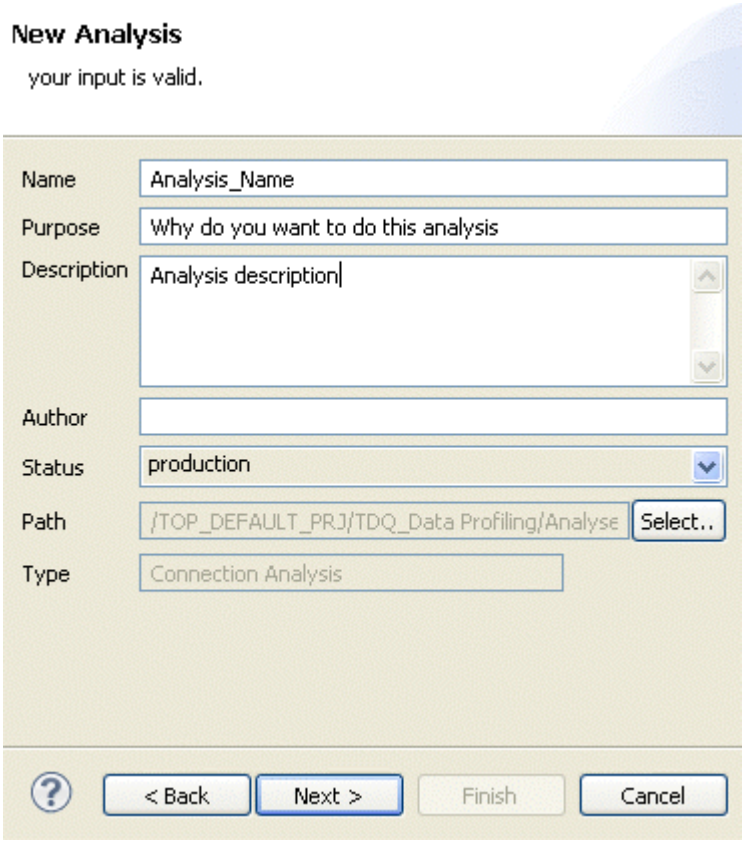
1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



3. Expand the **Redundancy Analysis** folder and select **Column Content Comparison**.
4. Click **Next**.

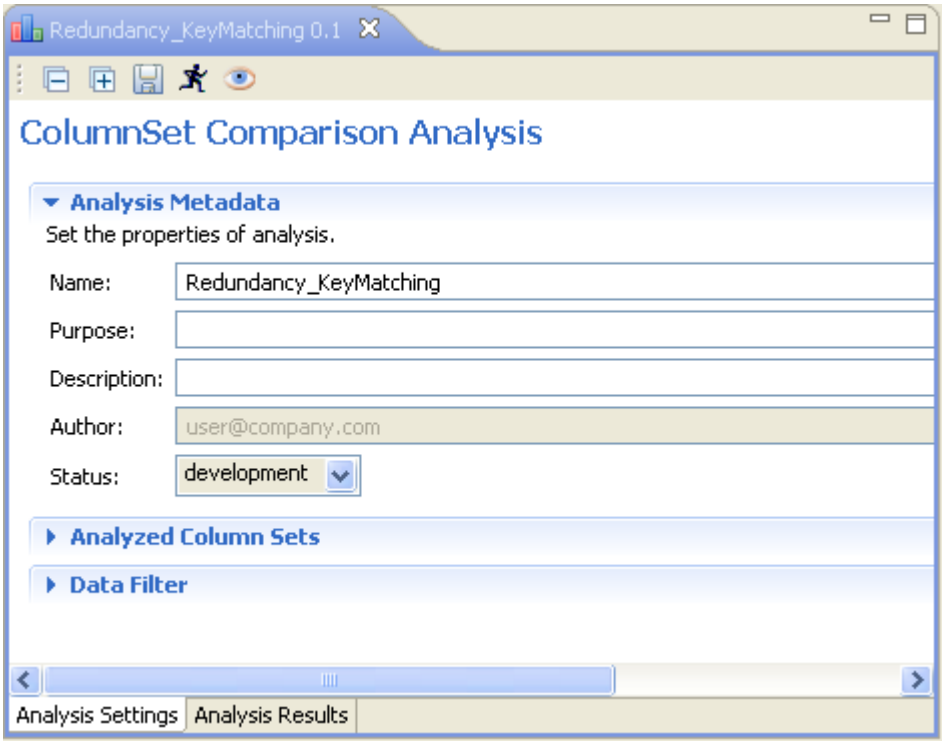


5. In the **Name** field, enter a name for the current analysis.

   Space is not acceptable when typing in the analysis name in this field.

6. Set the analysis metadata (purpose, description and author name) in the corresponding fields and then click **Finish**.

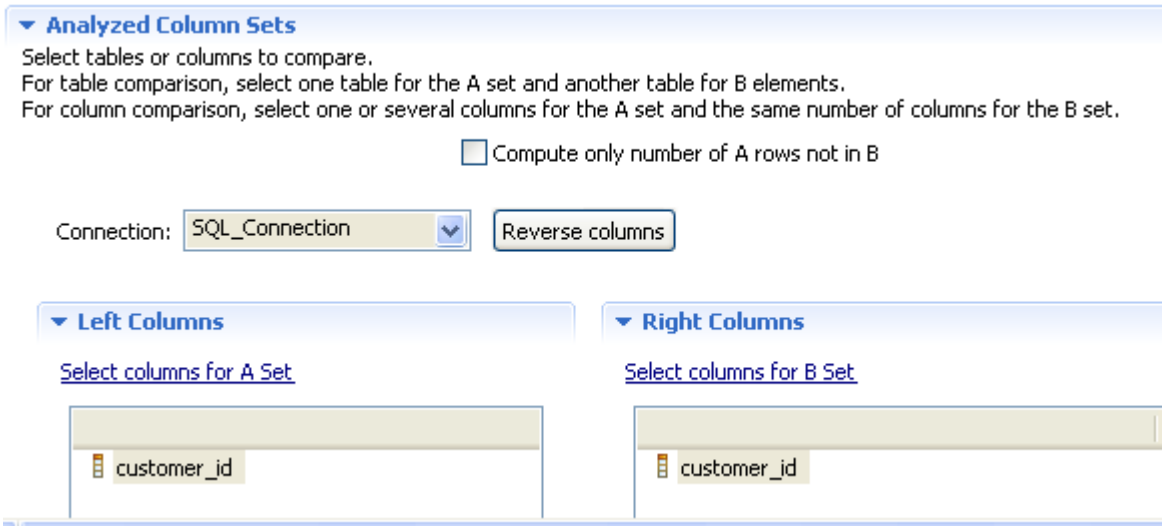
   A file for the newly created analysis is displayed under the **Analysis** folder in the **DQ Repository** tree view. The analysis editor opens with the defined analysis metadata.



Selecting the primary and foreign keys

1. Click **Analyzed Column Sets** to display the corresponding view.

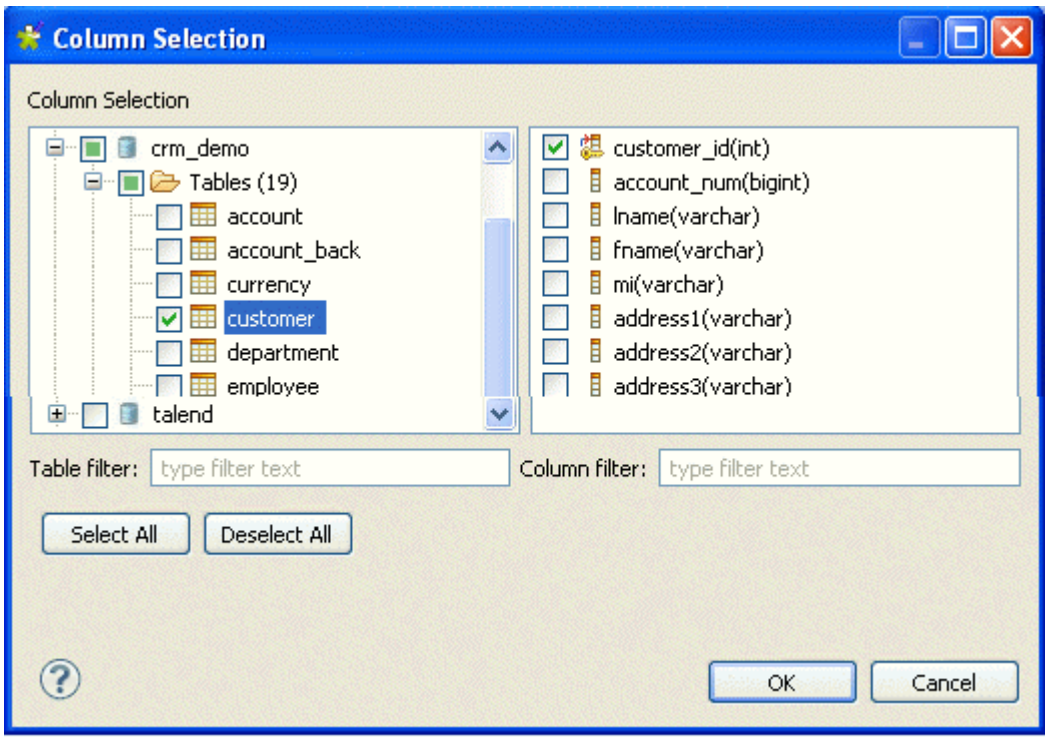
   In this example, you want to match the foreign keys in the *customer_id* column of the *sales_fact_1998* table with the primary keys in the *customer_id* column of the *customer* table, and vice versa. This will explore the relationship between the two tables to show us for example if every customer has an order in the year 1998.



2. From the **Connection** box, select the database connection relevant to the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.

3. Click **Select columns for the A set** to open the **[Column Selection]** dialog box.

   If you want to check the validity of the foreign keys, select the column holding the foreign keys for the A set and the column holding the primary keys for the B set.



4. Expand the **DB Connections** folder and browse through the catalogs/schemas to reach the table holding the column you want to match. In this example, the column to be analyzed is *customer_id* that holds the foreign keys.

   You can filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

5. Click the table name to display all its columns in the right-hand panel of the **[Column Selection]** dialog box.

6. In the list to the right, select the check box of the column holding the foreign keys and then click **OK** to proceed to the next step.

   You can drag the columns to be analyzed directly from the **DQ Repository** tree view to the editor.

   If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

7. Click **Select Columns from the B set** and follow the same steps to select the column holding the primary keys or drag it from the **DQ Repository** to the right column panel.

   If you select the **Compute only number of rows not in B** check box, you will look for any missing primary keys in the column in the B set.

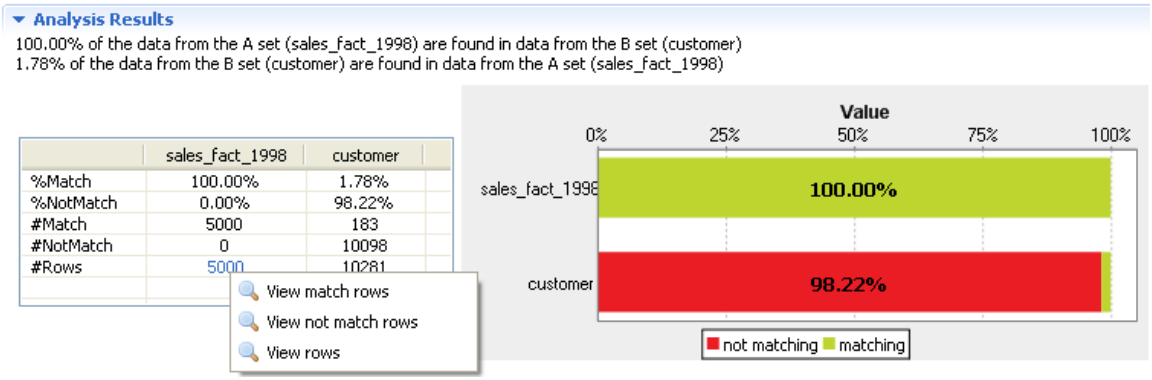
8. Click **Data Filter** in the analysis editor to display the view where you can set a filter on each of the analyzed columns.

9. Click the save icon on top of the editor, and then press **F6** to execute this key-matching analysis. A confirmation message is displayed.

10. Read the confirmation message and click **OK** if you want to continue the operation.

    The **Analysis Results**view opens to display the analysis results.

    The execution of this type of analysis may takes some time. Wait till the **Analysis Results** view opens automatically showing the analysis results.



In this example, every foreign key in the *sales_fact_1998* table is identified with a primary key in the *customer* table. However, 98.22% of the primary keys in the *customer* table could not be identified with foreign keys in the *sales_fact_1998* table. These primary keys are for the customers who did not order anything in 1998.

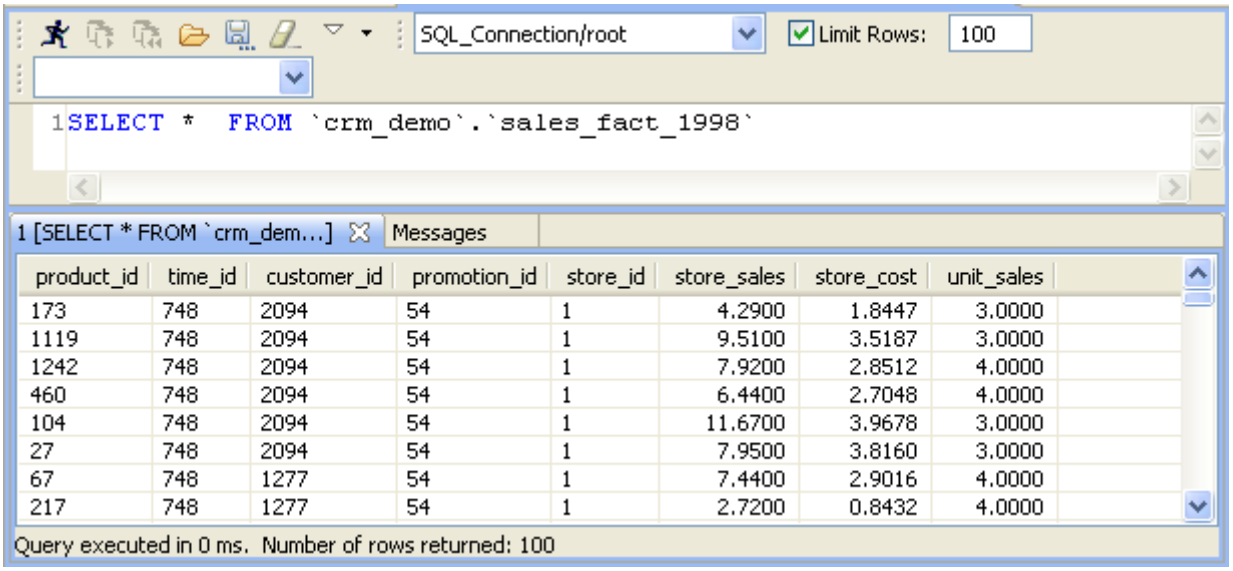
Through this view, you can also access the actual analyzed data via the data explorer.

To access the analyzed data rows, right-click any of the lines in the table and select::

Option	To...
View match rows	access a list of all rows that could be matched in the two identical column sets
View not match rows	access a list of all rows that could not be matched in the two identical column sets
View rows	access a list of all rows in the two identical column sets

*The data explorer does not support connections which has empty user name, such as Single sign-on of MS SQL Server. If you analyze data using such connection and you try to view data rows in the **Data Explorer** perspective, a warning message prompt you to set your connection credentials to the SQL Server.*

The figure below illustrates the data explorer list of all analyzed rows in the two columns.



From the SQL editor, you can save the executed query and list it under the **Libraries** > **Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see section *Saving the queries executed on indicators*.

For more information about the data explorer Graphical User Interface, see appendix *Data Explorer management GUI*.

Chapter 8. Correlation analyses

This chapter provides all the information you need to perform column correlation analyses between nominal and interval columns or nominal and date columns in database tables. A column correlation analysis can also investigate minimal correlations between nominal columns in the same table.

Column correlation analyses are usually used to explore relationships and correlations in data. They are not used to provide statistics about the quality of data.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

8.1. What are column correlation analyses

Your studio provides the possibility to explore relationships and correlations between two or more columns so that these relationships and correlations give a new interpretation of the data through describing how data values are correlated at different positions.

It is very important to make the distinction between column correlation analyses and all other types of data quality analyses. Column correlation analyses are usually used to explore relationships and correlations in data and not to provide statistics about the quality of data.

Several types of column correlation analysis are possible. For more information, see section *Creating numerical correlation analysis*, section *Creating time correlation analysis* and section *Creating nominal correlation analysis*.

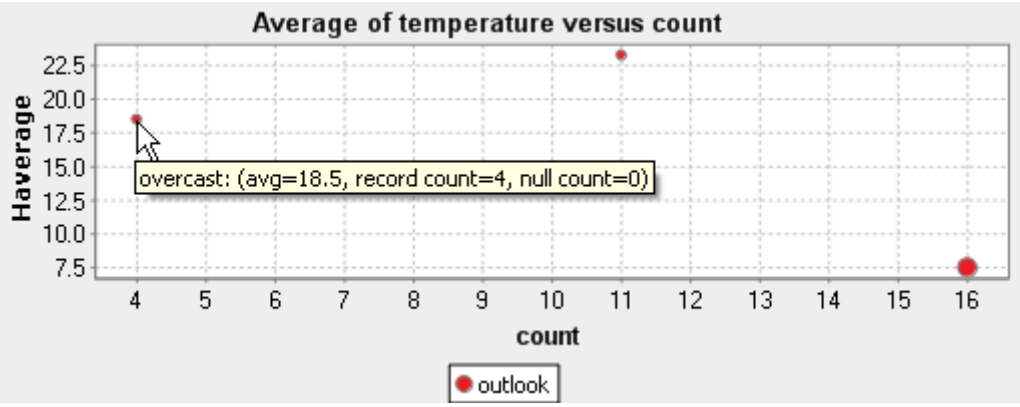
For more information about the use of data mining types in the studio, see section *Data mining types*.

The number of the analyses created in the studio is indicated next to the **Analyses** folder in the **DQ Repository** tree view.

8.2. Numerical correlation analysis

This type of analysis analyzes correlation between nominal and interval columns and gives the result in a kind of a bubble chart.

A bubble chart is created for each selected numeric column. In a bubble chart, each bubble represents a distinct record of the nominal column. For example, a nominal column called *outlook* with 3 distinct nominal instances: *sunny* (11 records), *rainy* (16 records) and *overcast* (4 records) will generate a bubble chart with 3 bubbles.



The second column in this example is the *temperature* column where temperature is in degrees Celsius. The analysis in this example will show the correlation between the *outlook* and the *temperature* columns and will give the result in a bubble chart. The vertical axis represents the average of the numeric column and the horizontal axis represents the number of records of each nominal instance. The average temperature would be 23.273 for the "sunny" instances, 7.5 for the "rainy" instances and 18.5 for the "overcast" instances.

The two things to pay attention to in such a chart is the position of the bubble and its size.

Usually, outlier bubbles must be further investigated. The more the bubble is near the left axis, the less confident we are in the average of the numeric column. For example, the *overcast* nominal instance here has only 4 records, hence the bubble is near the left axis. We cannot be confident in the average with only 4 records. When looking for data quality issues, these bubbles could indicate problematic values.

The bubbles near the top of the chart and those near the bottom of the chart may suggest data quality issues too. A too high or too low temperature in average could indicate a bad measure of the temperature.

The size of the bubble represents the number of null numeric values. The more there are null values in the interval column, the bigger will be the bubble.

When several nominal columns are selected, the order of the columns plays a crucial role in this analysis. A series of bubbles (with one color) is displayed for the average temperature and the weather. Another series of bubbles is displayed for the average temperature and each record of any other nominal column.

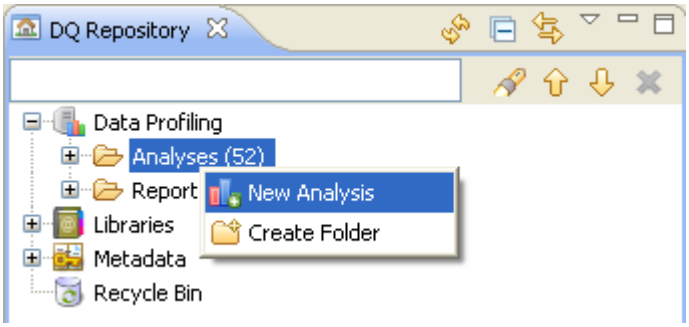
8.2.1. Creating numerical correlation analysis

In the example below, you want to create a numerical correlation analysis to compute the age average of the personnel of different enterprises located in different states. Three columns are used for the analysis: *STATE*, *AGE* and *COMPANY*.

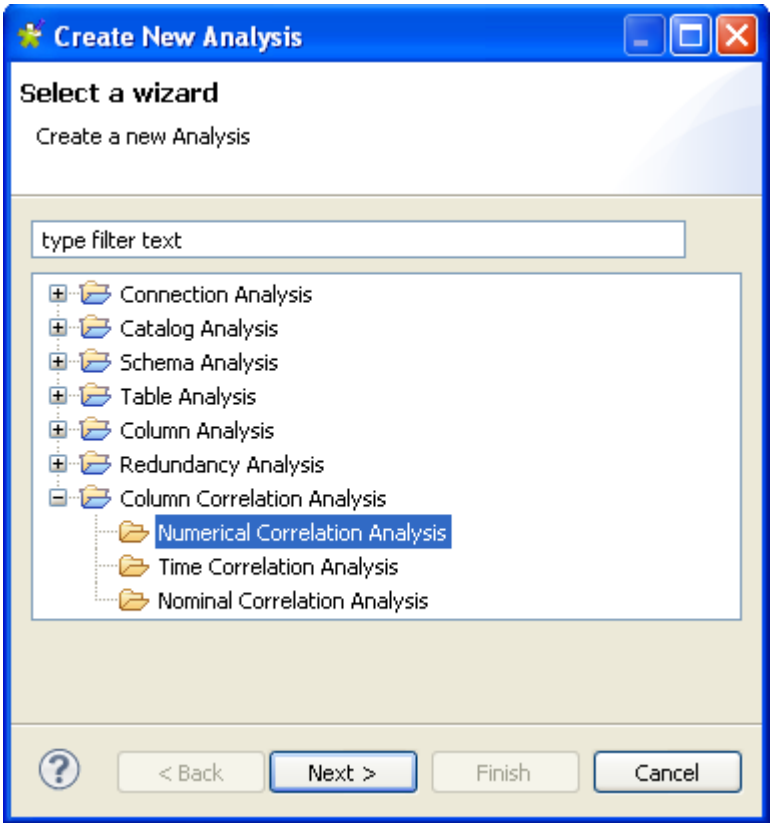
Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

Defining the analysis

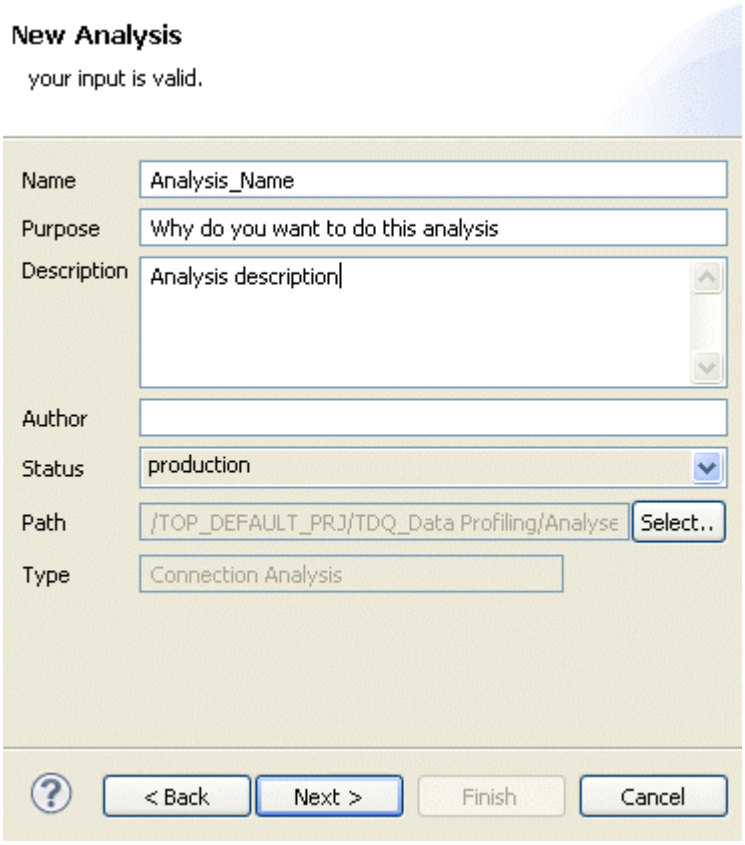
1. In the **DQ Repository** tree view, expand **Data Profiling**.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



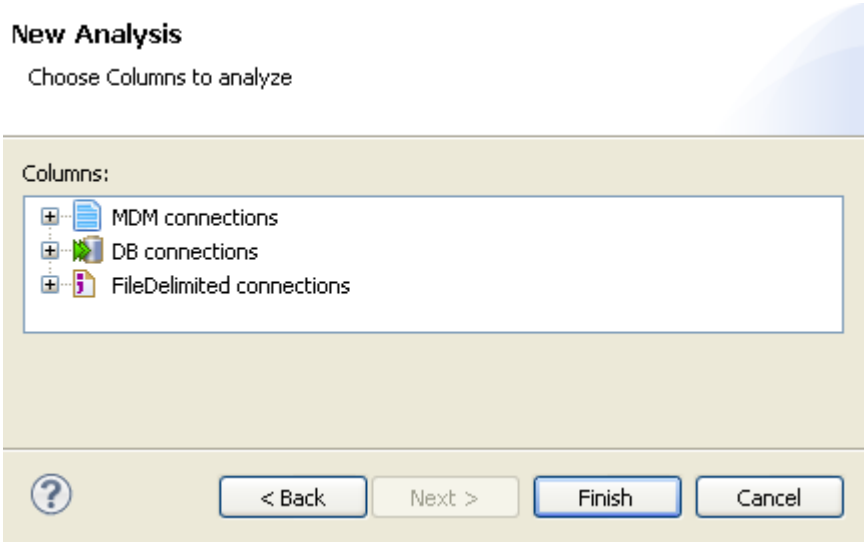
3. Expand the **Column Correlation Analysis** node and select **Numerical Correlation Analysis**.
4. Click **Next**.



5. In the **Name** field, enter a name for the current analysis.

Space is not acceptable when typing in the analysis name in this field.

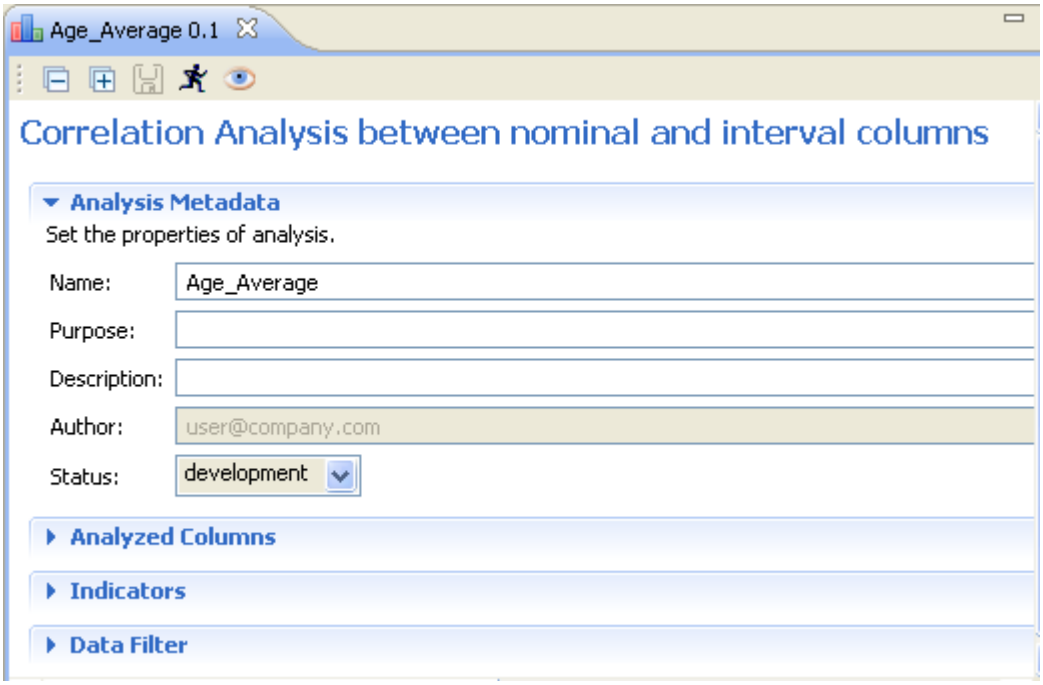
6. Set the analysis metadata (purpose, description and author name) in the corresponding fields and then click **Next**.



Selecting the columns you want to analyze

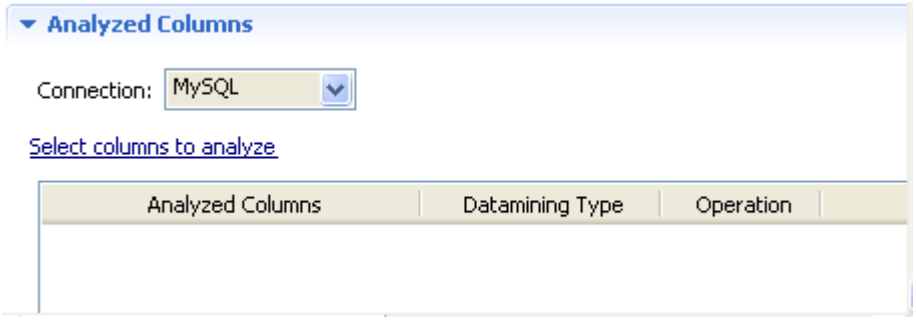
1. Expand **DB connections** and in the desired database, browse to the columns you want to analyze, select them and then click **Finish**.

A folder for the newly created analysis is listed under **Analysis** in the **DQ Repository** tree view, and the analysis editor opens with the defined analysis metadata.



The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

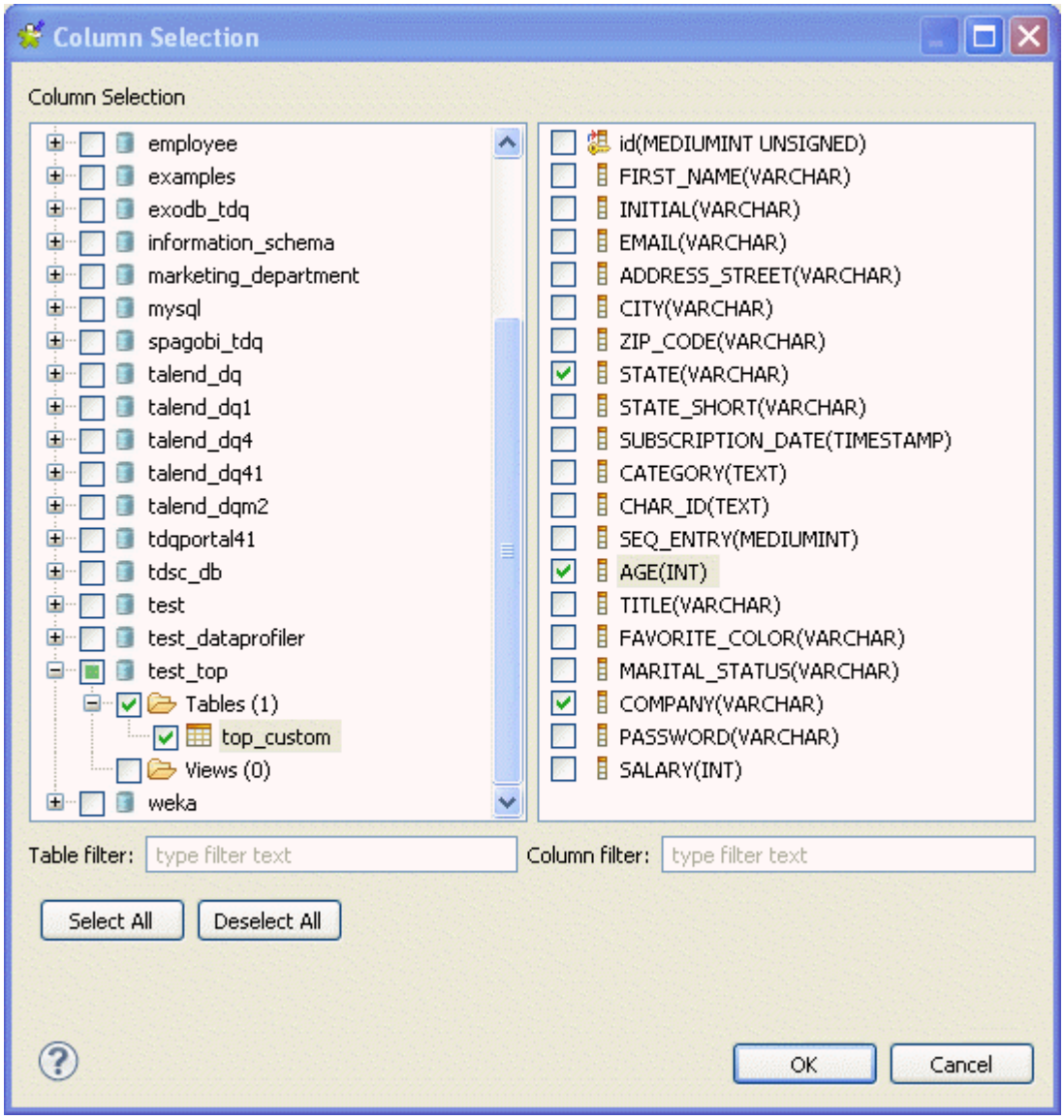
2. Click **Analyzed Columns** to open the corresponding view.



3. From the **Connection** box, select the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.

You can change your database connection by selecting another connection from the **Connection** box. If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

4. Click **Select columns to analyze** to open the **[Column Selection]** dialog box.



5. Expand **DB Connections** and browse the catalogs/schemas in your database connection to reach the table that holds the column(s) you want to analyze.

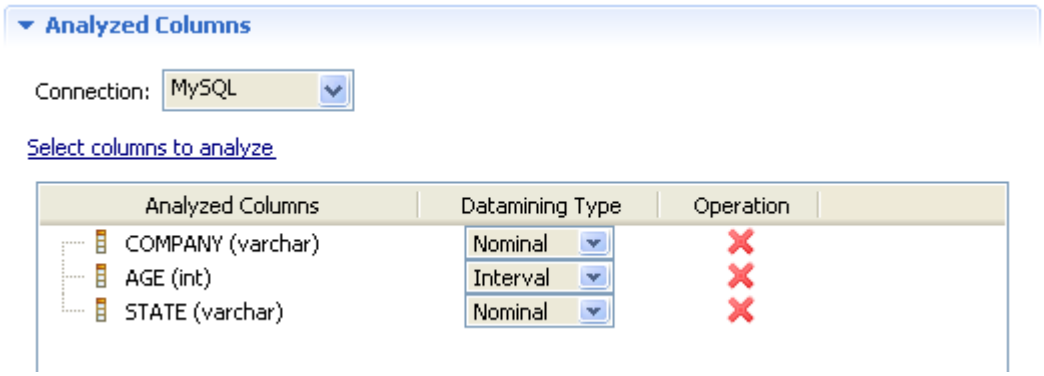
   You can filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

6. Click the table name to list all its columns in the right-hand panel of the **[Column Selection]** dialog box.

7. In the column list, select the check boxes of the column(s) you want to analyze and click **OK** to proceed to the next step.

   In this example, you want to compute the age average of the personnel of different enterprises located in different states. Then the columns to be analyzed are *AGE*, *COMPANY* and *STATE*.

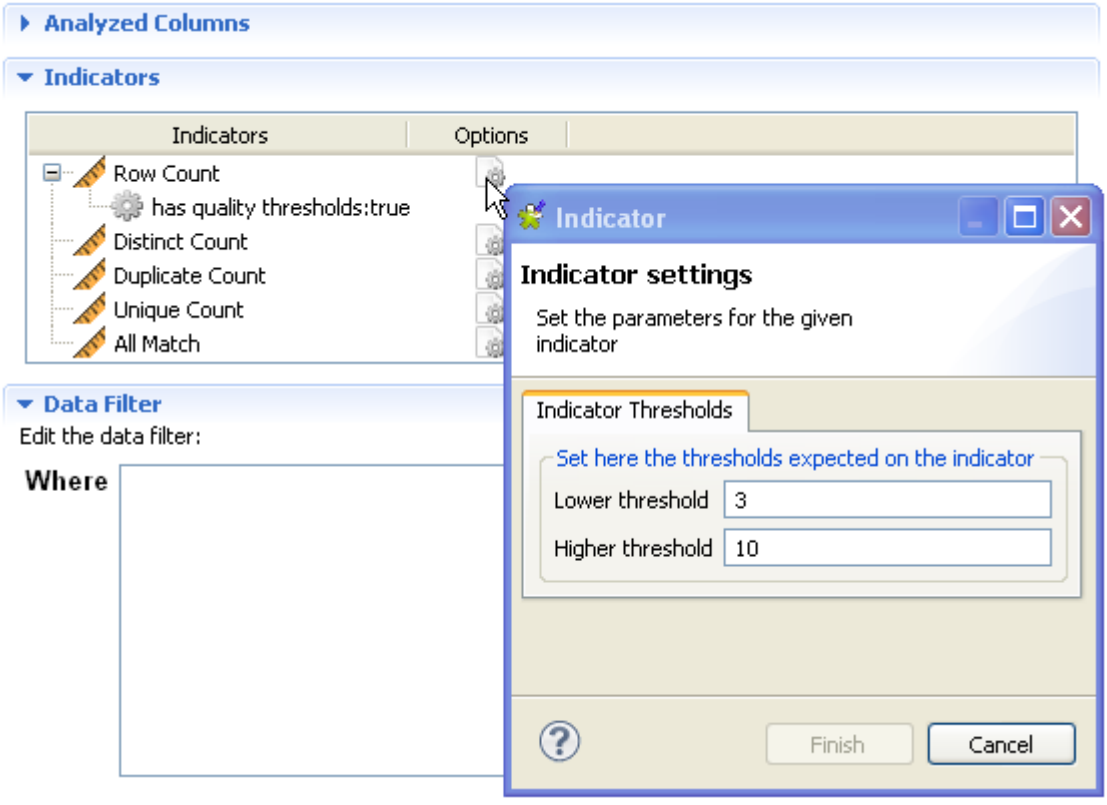
   The selected columns are displayed in the **Analyzed Column** view of the analysis editor.



You can drag the columns to be analyzed directly from the corresponding database connection in the **DQ Repository** tree view into the **Analyzed Columns** area.

If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

8. Click **Indicators** in the analysis editor and then click the option icon to open a dialog box where you can set thresholds for each indicator.

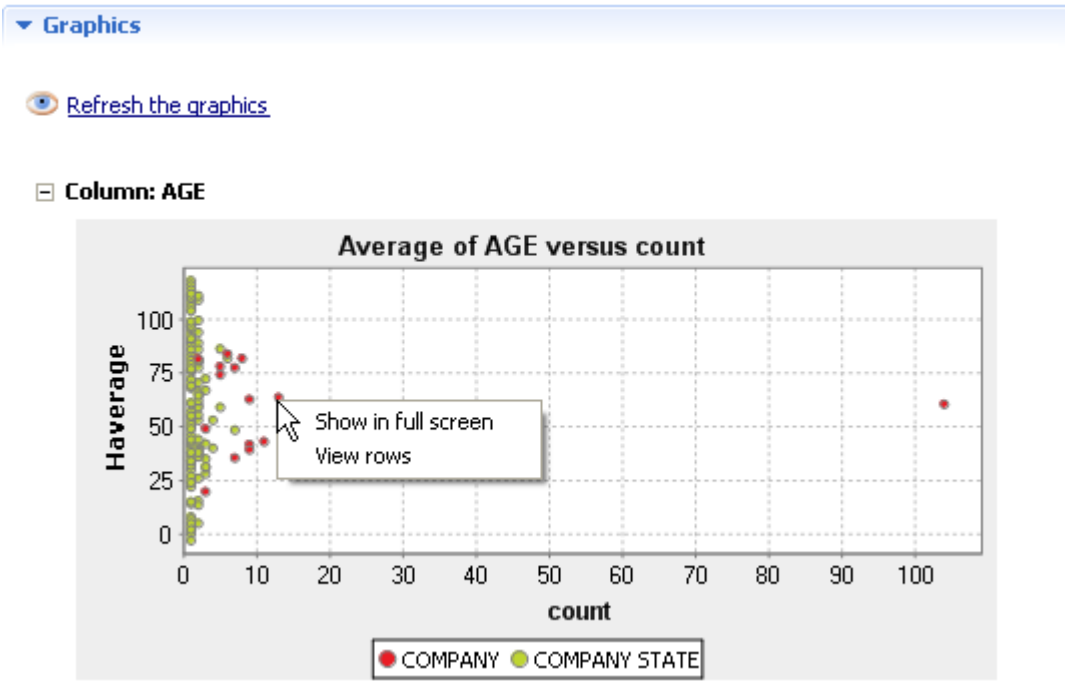


The indicators representing the simple statistics are by-default attached to this type of analysis.

9. Click **Data Filter** in the analysis editor to open the view where you can set a filter on the data of the analyzed columns.

10. Click the save icon on top of the editor and then press **F6** to execute the column comparison analysis.

    The graphical result is displayed in the **Graphics** panel to the right of the editor.



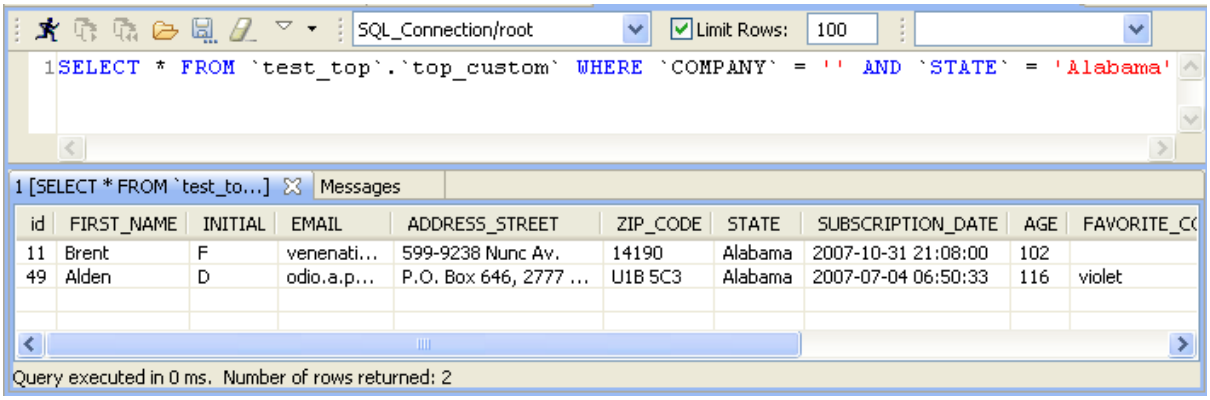
The data plotted in the bubble chart have different colors with the legend pointing out which color refers to which data.

From the generated graphic, you can:

- place the pointer on any of the bubbles to see the correlated data values at that position,
- right-click any of the bubbles and select:

Option	To...
Show in full screen	open the generated graphic in a full screen
View rows	access a list of all analyzed rows in the selected position

The below figure illustrates an example of the SQL editor listing the correlated data values at the selected position.



From the SQL editor, you can save the executed query and list it under the **Libraries** > **Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see section *Saving the queries executed on indicators*.

For more information on the bubble chart, see the below section.

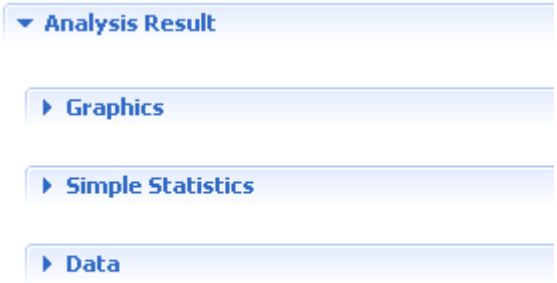
8.2.2. Accessing the detailed view of the analysis results

Prerequisite(s): A numerical correlation analysis is defined and executed in the **Profiling** perspective of the studio.

To access a more detailed view of the analysis results of the procedure outlined in section *Creating numerical correlation analysis*, do the following:

1. Click the **Analysis Results** tab at the bottom of the analysis editor to open the corresponding view.

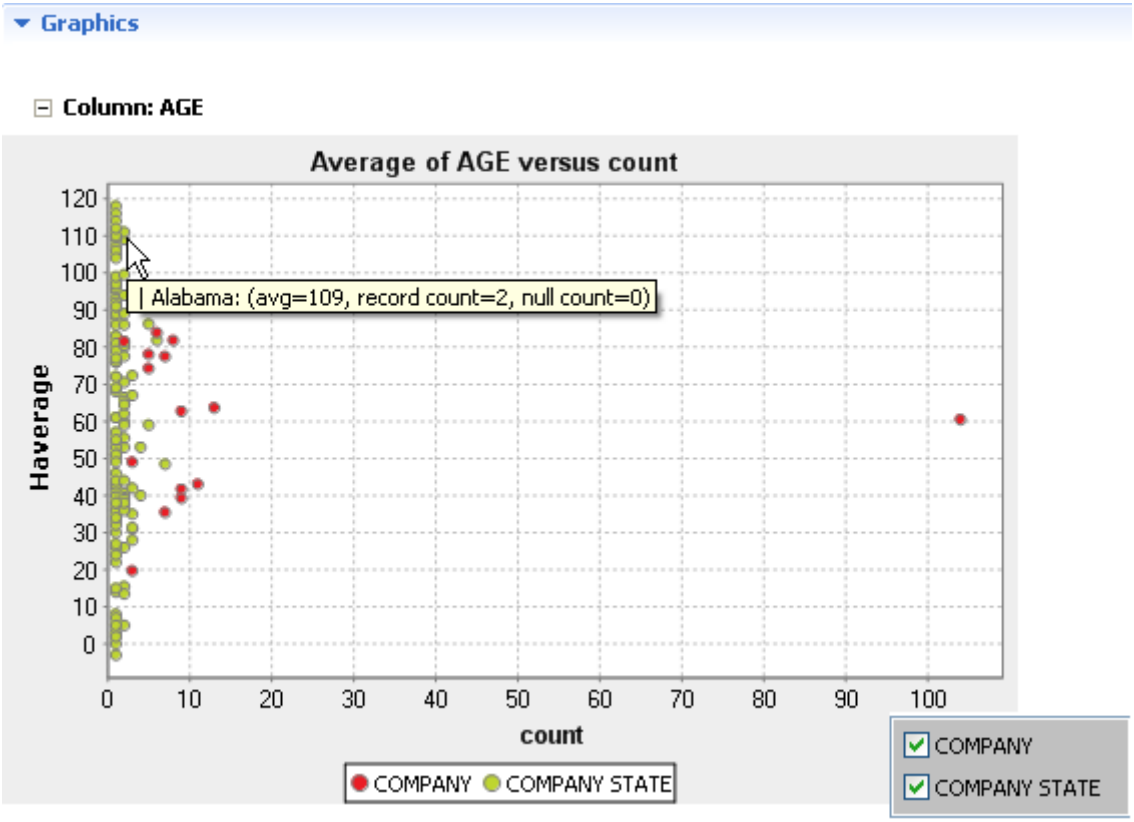
2. Click on **Analysis Result** to see more detail of the analysis results in the three different views: **Graphics**, **Simple Statistics** and **Data**.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

3. Click **Graphics**, **Simple Statistics** or **Data** to show the generated graphic, the number of the analyzed records or the actual analyzed data respectively.

In the **Graphics** view, the data plotted in the bubble chart have different colors with the legend pointing out which color refers to which data.



The more the bubble is near the left axis the less confident we are in the average of the numeric column. For the selected bubble in the above example, the company name is missing and there are only two data records, hence the bubble is near the left axis. We cannot be confident about age average with only two records. When looking for data quality issues, these bubbles could indicate problematic values.

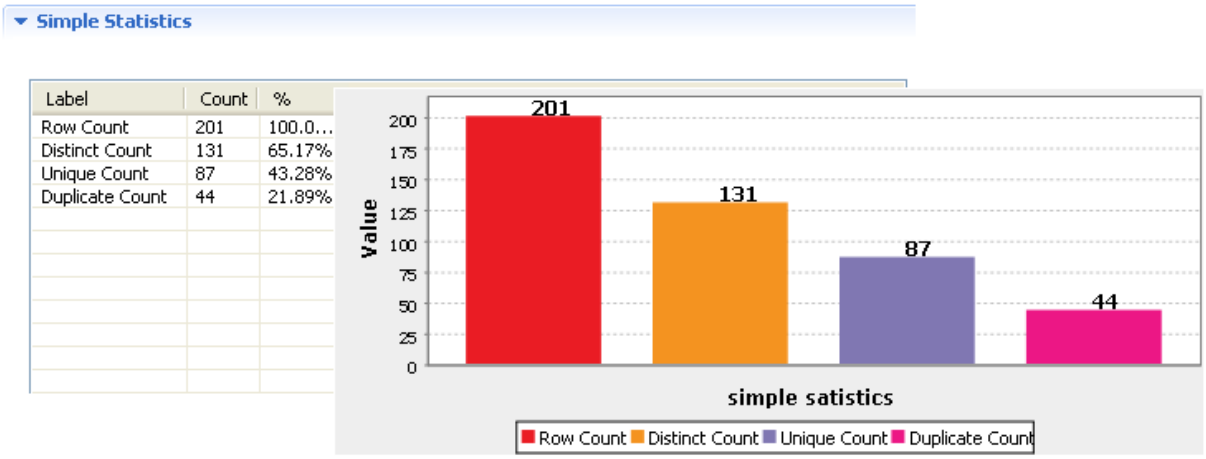
The bubbles near the top of the chart and those near the bottom of the chart may suggest data quality issues too, too big or too small age average in the above example.

From the generated graphic, you can:

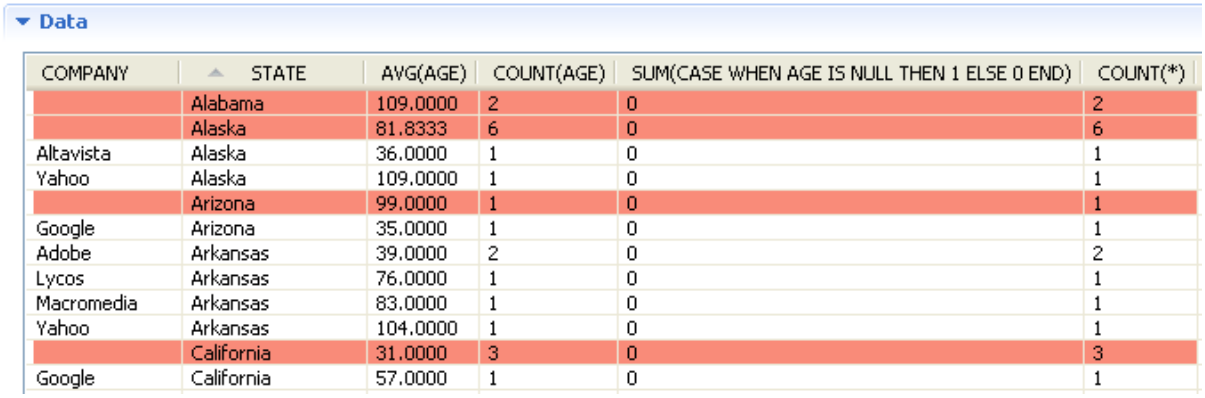
- clear the check box of the value(s) you want to hide in the bubble chart,
- place the pointer on any of the bubbles to see the correlated data values at that position,
- right-click any of the bubbles and select:

Option	To...
Show in full screen	open the generated graphic in a full screen
View rows	access a list of all analyzed rows in the selected column

The **Simple Statistics** view shows the number of the analyzed records falling in certain categories, including the number of rows, the number of distinct and unique values and the number of duplicates.



The **Data** view displays the actual analyzed data.



COMPANY	STATE	AVG(AGE)	COUNT(AGE)	SUM(CASE WHEN AGE IS NULL THEN 1 ELSE 0 END)	COUNT(*)
	Alabama	109.0000	2	0	2
	Alaska	81.8333	6	0	6
Altavista	Alaska	36.0000	1	0	1
Yahoo	Alaska	109.0000	1	0	1
	Arizona	99.0000	1	0	1
Google	Arizona	35.0000	1	0	1
Adobe	Arkansas	39.0000	2	0	2
Lycos	Arkansas	76.0000	1	0	1
Macromedia	Arkansas	83.0000	1	0	1
Yahoo	Arkansas	104.0000	1	0	1
	California	31.0000	3	0	3
Google	California	57.0000	1	0	1

You can sort the data listed in the result table by simply clicking any column header in the table.

8.3. Time correlation analysis

This type of analysis analyzes correlation between nominal and date columns and gives the result in a gantt chart that illustrates the start and finish dates of each value of the nominal column.

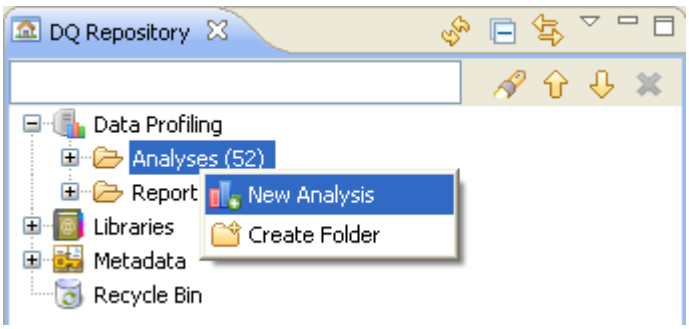
8.3.1. Creating time correlation analysis

In the example below, you want to create time correlation analysis to compute the minimal and maximal birth dates for each listed country in the selected nominal column. Two columns are used for the analysis: *birthdate* and *country*.

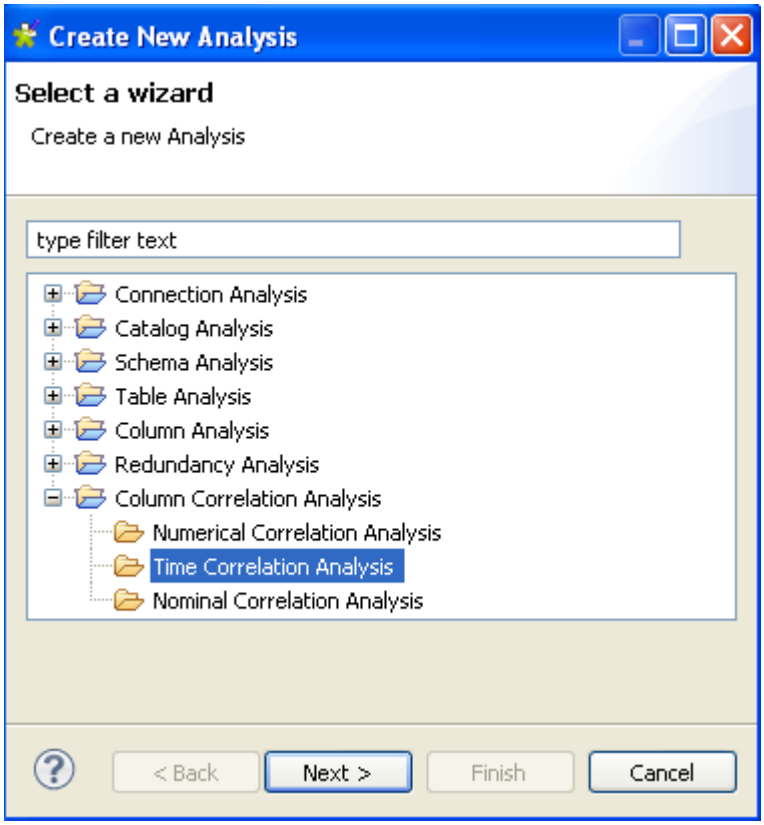
Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

Defining the analysis

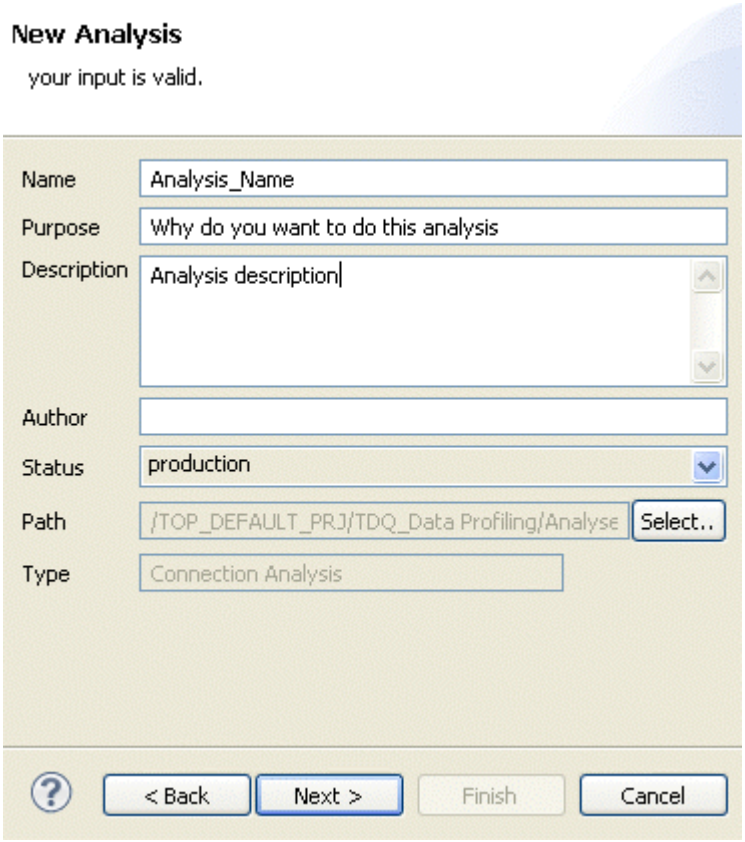
1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.
2. Right-click the **Analyses** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.

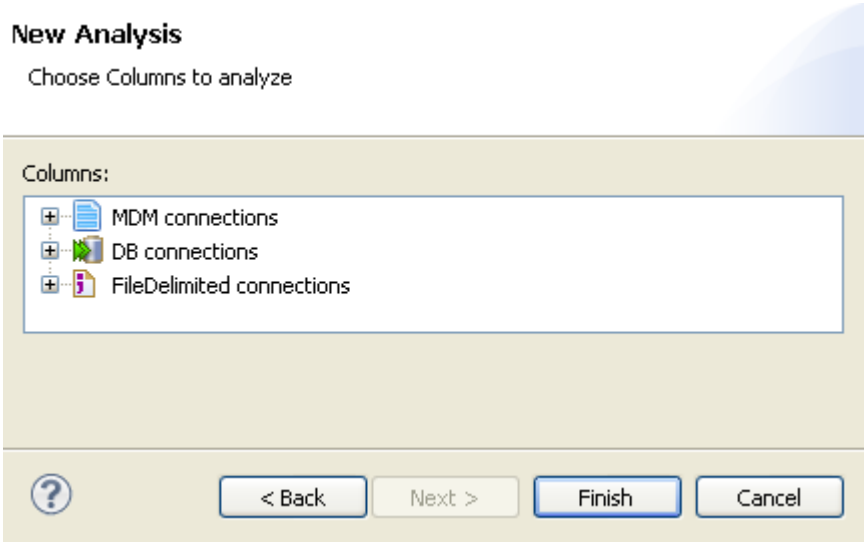


3. Expand the **Column Correlation Analysis** folder and select **Time Correlation Analysis**.
4. Click **Next**.



5. In the **Name** field, enter a name for the current analysis.

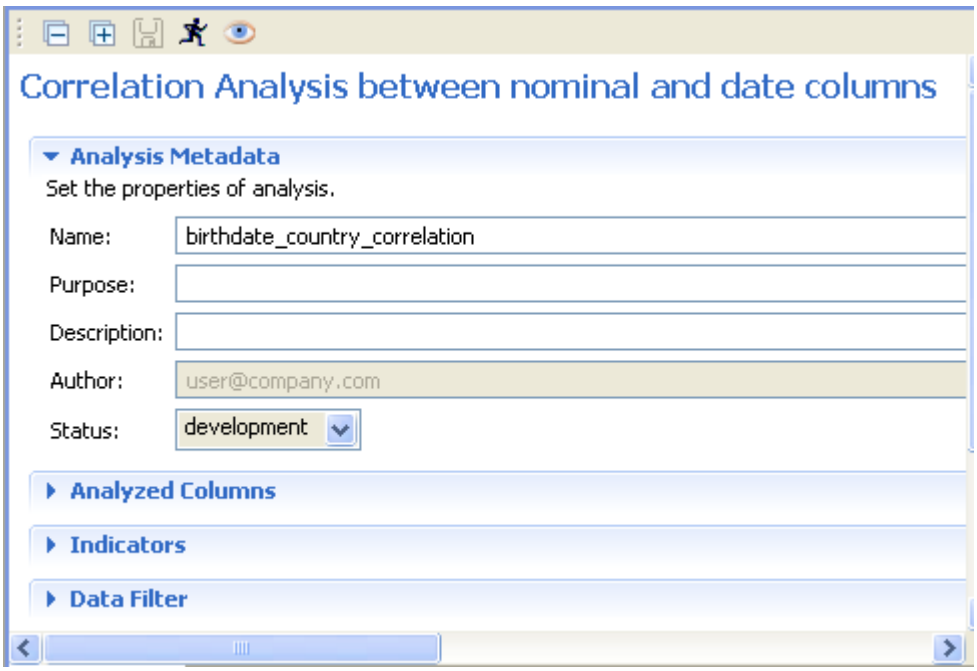
6. Set the analysis metadata (purpose, description and author name) in the corresponding fields and then click **Next**.



Selecting the columns you want to analyze

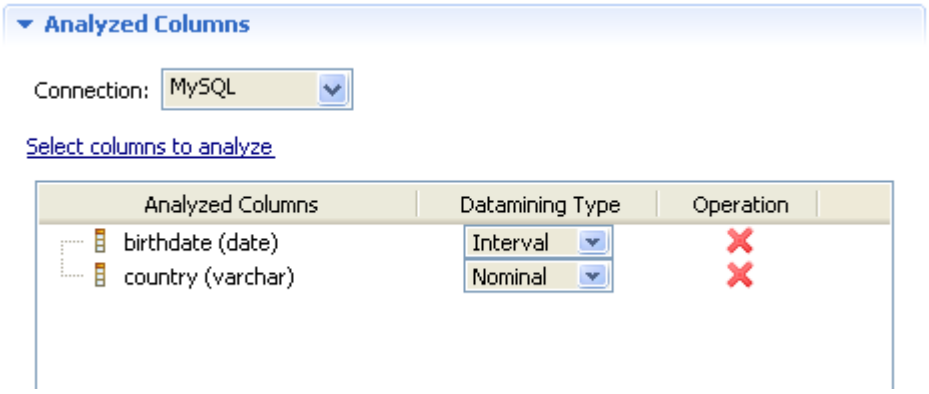
1. Expand **DB connections** and in the desired database, browse to the columns you want to analyze, select them and then click **Finish**.

   A folder for the newly created analysis is displayed under **Analysis** in the **DQ Repository** tree view, and the time analysis editor opens with the defined analysis metadata.



The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

2. Click **Analyzed Columns** to display the corresponding view.



3. From the **Connection** box, select the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.

You can change your database connection by selecting another connection from the **Connection** box. If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

4. Click **Select columns to analyze** to open the **[Column Selection]** dialog box and select the columns, or drag them directly from the **DQ Repository** tree view into the **Analyzed Columns** view.

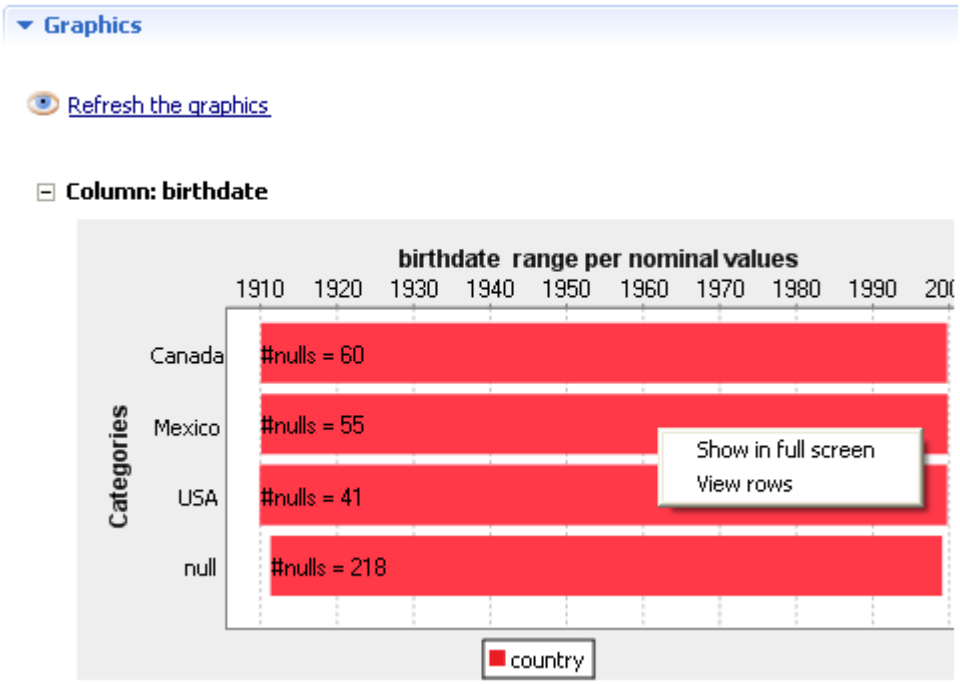
If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

5. If required, click **Indicators** in the analysis editor to display the indicators used in the current time correlation analysis.

6. Click **Data Filter** in the analysis editor to display the view where you can set a filter on the analyzed column set.

7. Click the save icon on top of the editor and press **F6** to execute the column comparison analysis.

   The graphical result is displayed in the **Graphics** panel to the right.



This gantt chart displays a range showing the minimal and maximal birth dates for each country listed in the selected nominal column. It also highlights the range bars that contain null values for birth dates.

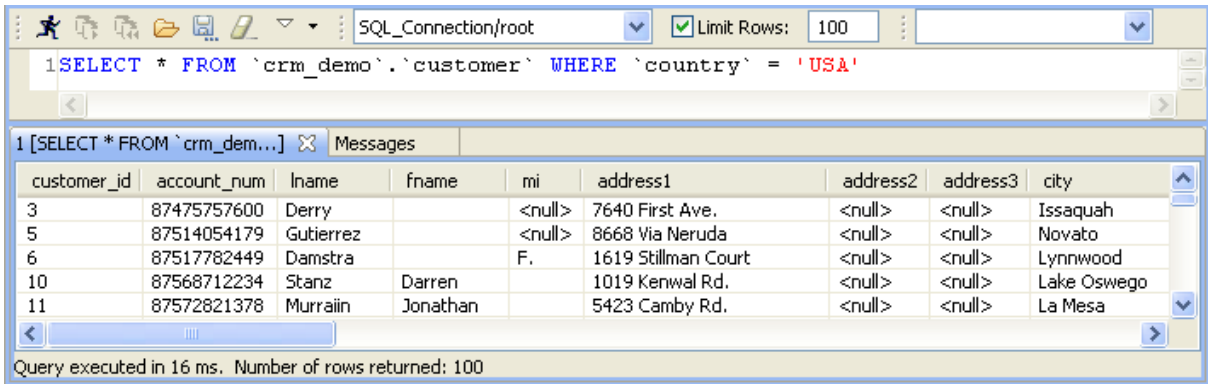
For example, in the above chart, the minimal birth date for Mexico is 1910 and the maximal is 2000. And of all the data records where the country is Mexico, 41 records have null value as birth date.

From the generated graphic, you can:

- place the pointer on any of the range bars to display the correlated data values at that position,
- put the pointer on a specific birth date and drag it to another birth date to change the chart and show the minimal and maximal birth dates related only to your selection.
- right-click any of the range bars and select:

Option	To...
Show in full screen	open the generated graphic in a full screen
View rows	access a list of all analyzed rows in the selected nominal column

The below figure illustrates an example of the SQL editor listing the correlated data values at the selected range bar.



From the SQL editor, you can save the executed query and list it under the **Libraries** > **Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see section *Saving the queries executed on indicators*.

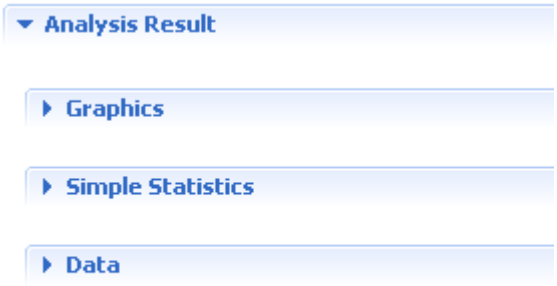
For more information on the gantt chart, see the below section.

8.3.2. Accessing the detailed view of the analysis results

Prerequisite(s): A time correlation analysis is defined and executed is the **Profiling** perspective of the studio.

To access a more detailed view of the analysis results of the procedure outlined in section *Creating time correlation analysis*, do the following:

1. Click the **Analysis Results** tab at the bottom of the analysis editor to open the corresponding view.
2. Click on **Analysis Result** to display the analysis more detailed results in the three different views: **Graphics**, **Simple Statistics** and **Data**.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

3. Click **Graphics**, **Simple Statistics** or **Data** to show the generated graphic, the number of the analyzed records or the actual analyzed data respectively.

In the **Graphics** view, you can clear the check box of the value(s) you want to hide in the chart.

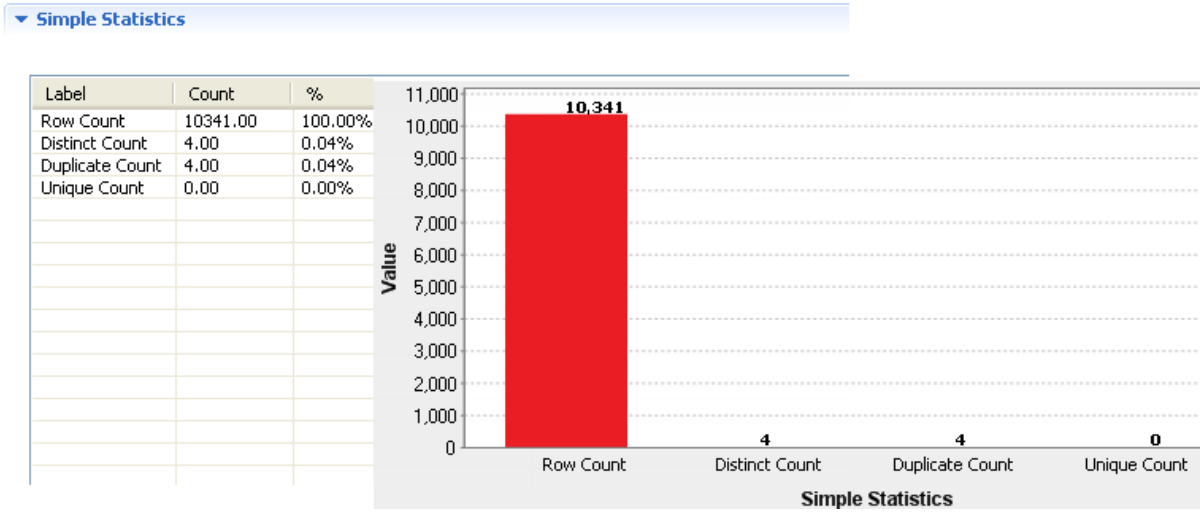
You can also select a specific birth date range to show if you put the pointer at the start nominal value you want to show and drag it to the end nominal value you want to show.

From the generated graphic, you can:

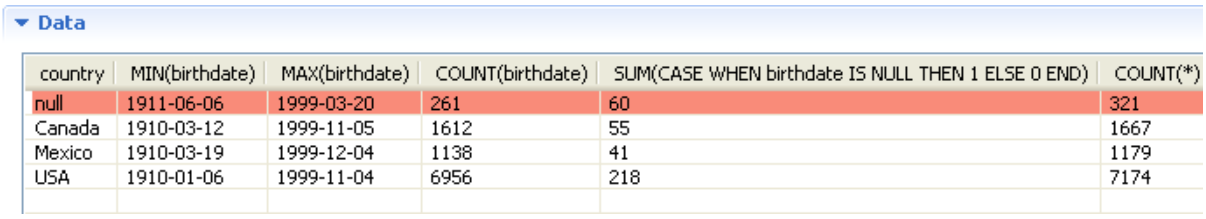
- clear the check box of the value(s) you want to hide in the chart,
- place the pointer on any of the range bars to display the correlated data values at that position,
- right-click any of the bars and select:

Option	To...
Show in full screen	open the generated graphic in a full screen
View rows	access a list of all analyzed rows in the selected column

The **Simple Statistics** view shows the number of the analyzed records falling in certain categories, including the number of rows, the number of distinct and unique values and the number of duplicates.



The **Data** view displays the actual analyzed data.



You can sort the data listed in the result table by simply clicking any column header in the table.

8.4. Nominal correlation analysis

This type of analysis analyzes minimal correlations between nominal columns in the same table and gives the result in a chart.

In the chart, each column will be represented by a node that has a given color. The correlations between the nominal values are represented by lines. The thicker the line is, the weaker the association is. Thicker lines can identify problems or correlations that need special attention. However, you can always inverse edge weight, that is give larger edge thickness to higher correlation, by selecting the **Inverse Edge Weight** check box below the nominal correlation chart.

The correlations in the chart are always pairwise correlations: show associations between pairs of columns.

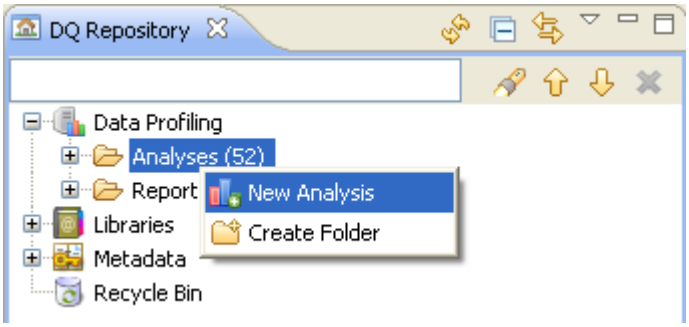
8.4.1. Creating nominal correlation analysis

In the example below, you want to create nominal correlation analysis to compute the minimal and maximal birth dates for each listed country in the selected nominal column. Two columns are used for the analysis: *birthdate* and *country*.

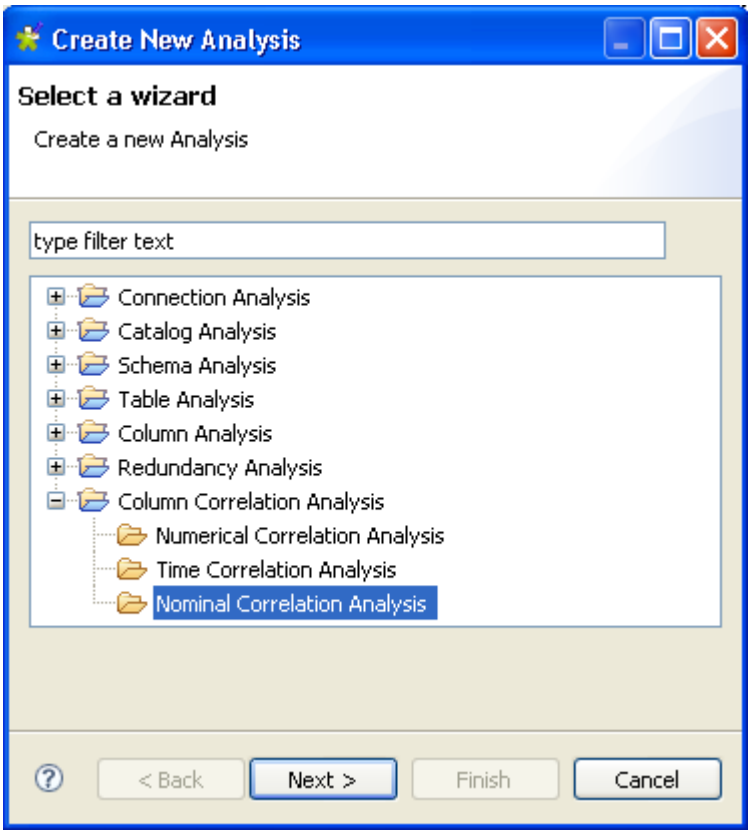
Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

Defining the analysis

1. In the **DQ Repository** tree view, expand the **Data Profiling** folder.
2. Right-click the **Analyses** folder and select **New Analysis**.

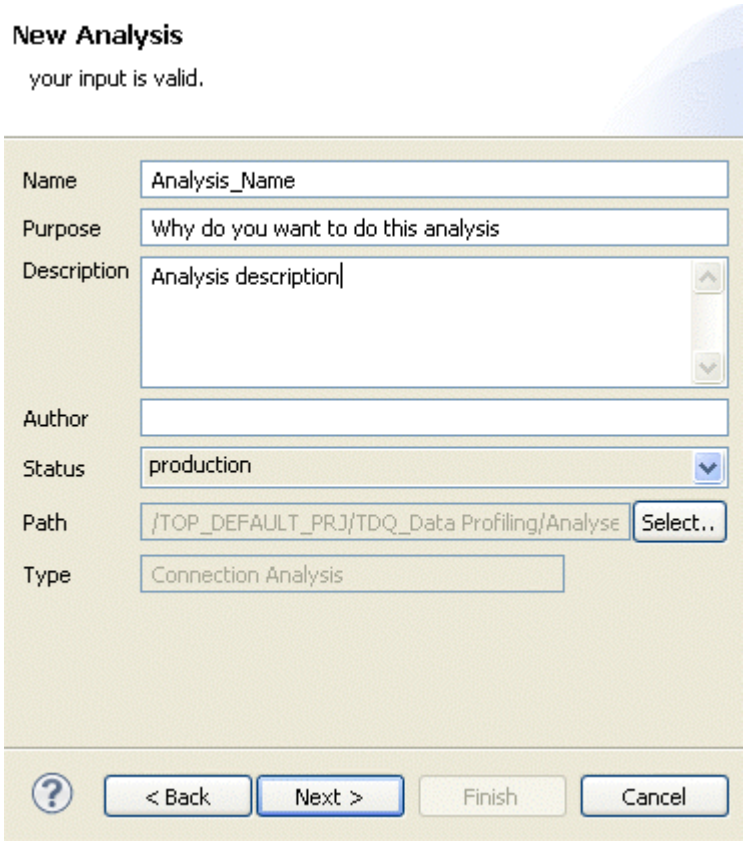


The **[Create New Analysis]** wizard opens.



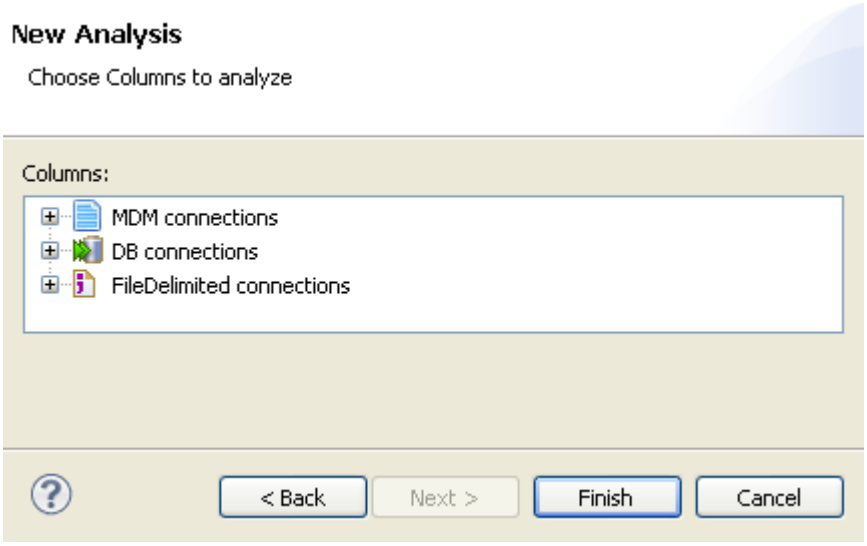
3. Expand **Column Correlation Analysis** and select **Nominal Correlation Analysis**.

4. Click **Next**.



5. In the **Name** field, enter a name for the current analysis.

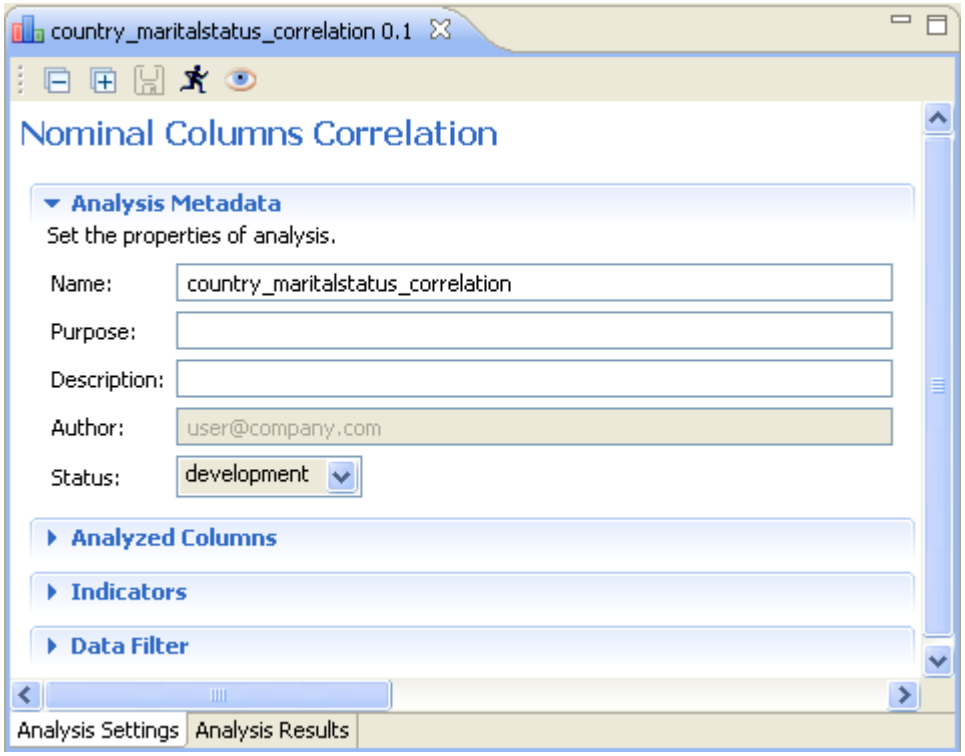
6. Set the analysis metadata (purpose, description and author name) in the corresponding fields and then click **Next**.



Selecting the columns you want to analyze

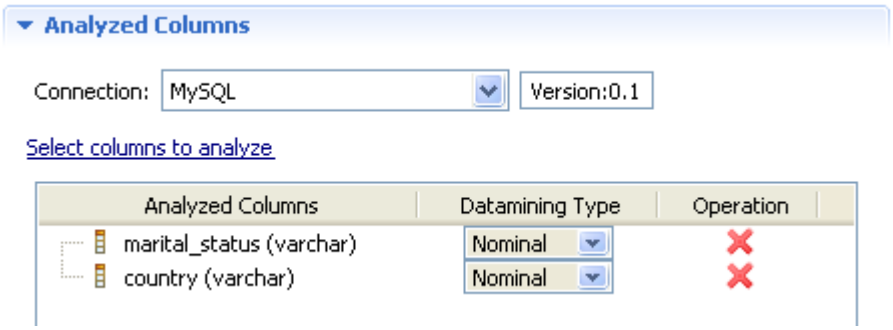
1. Expand **DB connections** and in the desired database, browse to the columns you want to analyze, select them and then click **Finish** to close the wizard.

   A folder for the newly created analysis is displayed under **Analysis** in the **DQ Repository** tree view, and the analysis editor opens with the defined analysis metadata.



The display of the analysis editor depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

2. Click **Analyzed Columns** to display the corresponding view.



3. From the **Connection** box, select the database to which you want to connect.

   This box lists all the connections created in the Studio with the corresponding database names.

You can change your database connection by selecting another connection from the **Connection** box. If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

4. Click **Select columns to analyze** to open the **[Column Selection]** dialog box and select as many nominal columns as you want, or drag them directly from the **DQ Repository** tree view.

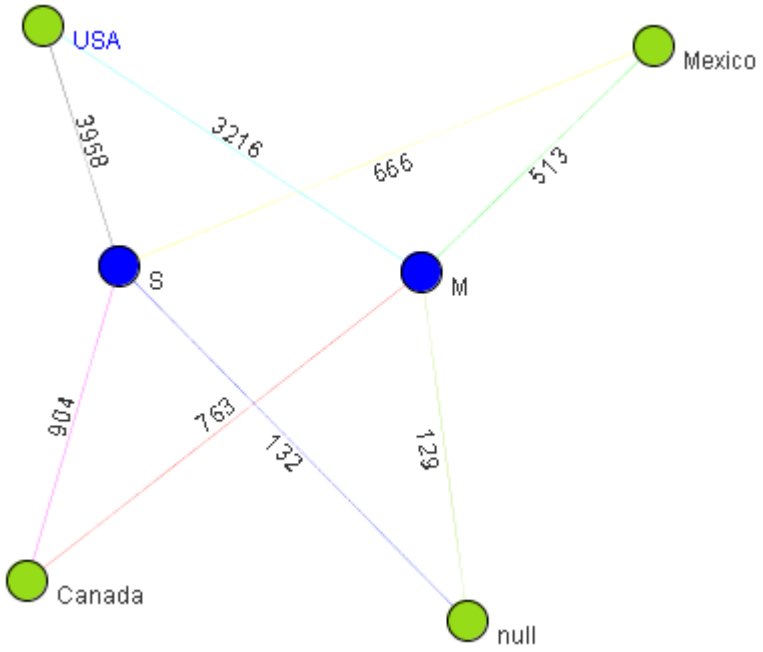
If you select too many columns, the analysis result chart will be very difficult to read.

If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

5. If required, click **Indicators** in the analysis editor to display the indicators used in the current nominal correlation analysis.

6. Click **Data Filter** in the analysis editor to display the view where you can set a filter on the data of the analyzed columns.

7. Click the save icon on top of the editor and then press **F6** to execute the nominal correlation analysis. The graphical result is displayed in the **Graphics** panel to the right of the editor.



In the above chart, each value in the *country* and *marital-status* columns is represented by a node that has a given color. Nominal correlation analysis is carried out to see the relationship between the number of married or single people and the country they live in. Correlations are represented by lines.

To better view the graphical result of the nominal correlation analysis, right-click the graphic in the **Graphics** panel and select **Show in full screen**. For more information on the chart, see the below section.

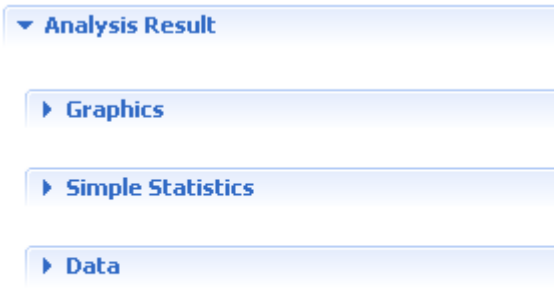
8.4.2. Accessing the detailed view of the analysis results

Prerequisite(s): A nominal correlation analysis is defined and executed in the **Profiling** perspective of the studio.

To access a more detailed view of the analysis results of the procedure outlined in section *Creating nominal correlation analysis*, do the following:

1. Click the **Analysis Results** tab at the bottom of the analysis editor to open the corresponding view.

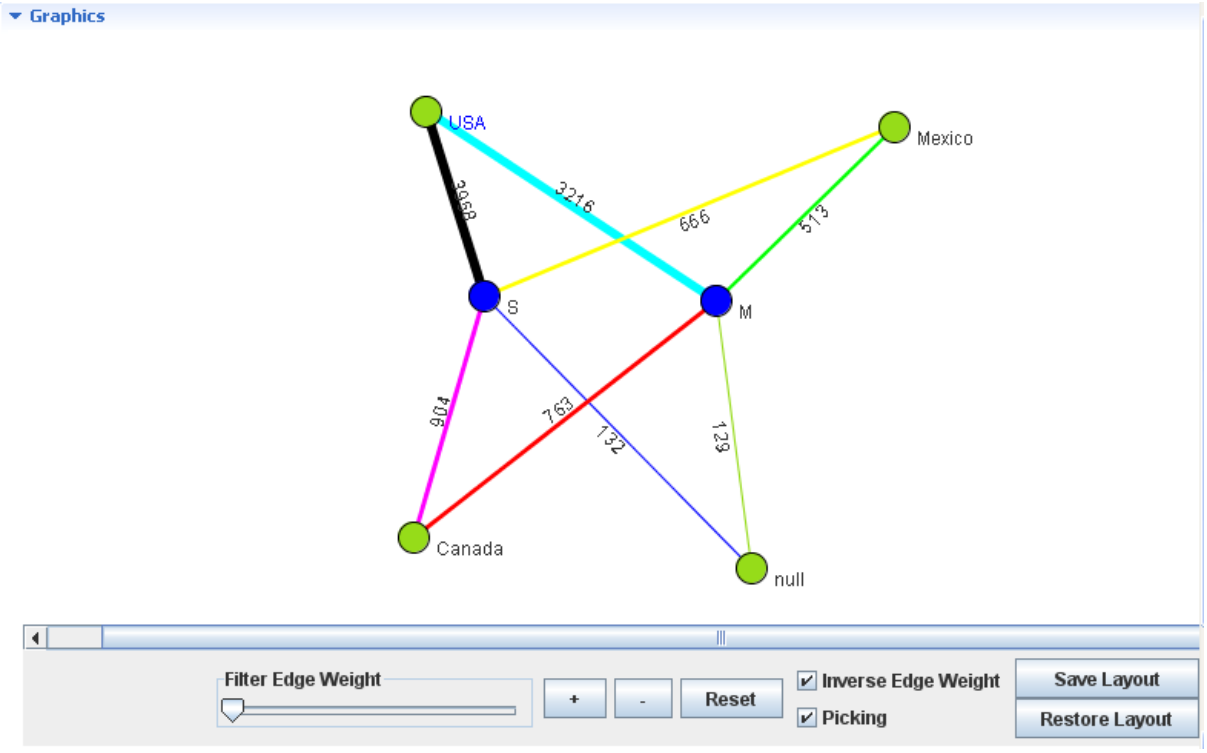
2. Click on **Analysis Result** to display the analysis more detailed results in three different views: **Graphics, Simple Statistics** and **Data**.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** window. For more information, see section *Setting preferences of analysis editors and analysis results*.

3. Click **Graphics**, **Simple Statistics** or **Data** to show the generated graphic, the number of the analyzed records or the actual data respectively.

The **Graphics** view shows the generated graphic for the analyzed columns.



In the above chart, each value in the *country* and *marital-status* columns is represented by a node that has a given color. Nominal correlation analysis is carried out to see the relationship between the number of married or single people and the country they live in. Correlations are represented by lines, the thicker the line is, the higher the association is - if the **Inverse Edge Weight** check box is selected.

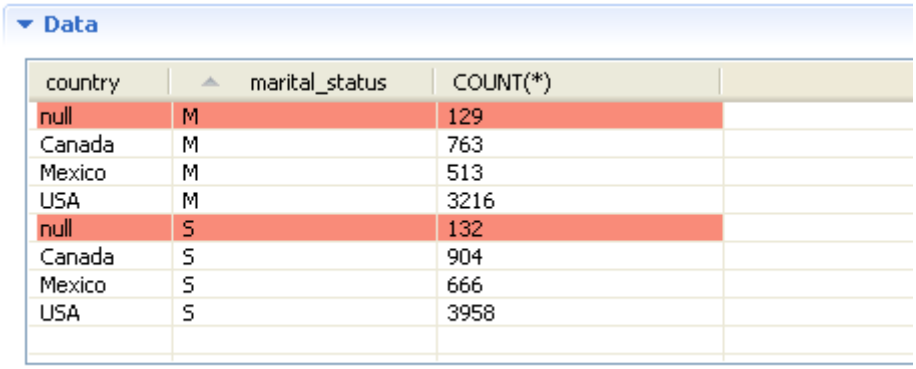
The buttons below the chart help you manage the chart display. The following table describes these buttons and their usage:

Button	Description
Filter Edge Weight	Move the slider to the right to (filter out edges with small weight) visualize the more important edges.
plus and minus	Click the **[+]** or **[-]** buttons to respectively zoom in and zoom out the chart size.
Reset	Click to put the chart back to its initial state.
Inverse Edge Weight	By default, the thicker the line is, the weaker the correlation is.

Button	Description
	Select this check box to inverse the current edge weight, that is give larger edge thickness to higher correlation.
Picking	Select this check box to be able to pick any node and drag it to anywhere in the chart.
Save Layout	Click this button to save the chart layout.
Restore Layout	Click this button to restore the chart to its previously saved layout.

The **Simple Statistics** view shows the number of the analyzed records falling in certain categories, including the number of rows, the number of distinct and unique values and the number of duplicates.

The **Data** view displays the actual analyzed data.



You can sort the data listed in the result table by simply clicking any column header in the table.

Chapter 9. Extended functionality: patterns and indicators

This chapter provides detailed information about how to use regular expressions and SQL patterns to analyze and monitor data in columns. It also explains how to use system and user-defined indicators when analyzing columns.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

9.1. Patterns

Patterns are sets of strings against which you can match the content of the columns to be analyzed.

9.1.1. Pattern types

Two types of patterns are listed under the **Patterns** folder in the **DQ Repository** tree view: regular expressions and SQL patterns.

Regular expressions (regex) are predefined patterns that you can use to search and manipulate text in the databases to which you connect. You can also create your own regular expressions and use them to analyze columns.

SQL patterns are a kind of personalized patterns that are used in SQL queries. These patterns usually contain the percent sign (%). For more information on SQL wildcards, see http://www.w3schools.com/SQL/sql_wildcards.asp.

You can use any of the above two pattern types either with column analyses or with the analyses of a set of columns (simple table analyses). These pattern-based analyses illustrate the frequencies of various data patterns found in the values of the analyzed columns. For more information, see section *Analyzing columns in a database*and section *How to create an analysis of a set of columns using patterns*.

From the studio, you can generate graphs to represent the results of analyses using patterns. You can also view tables in the **Analysis Results** view that write in words the generated graphs. From those graphs and analysis results you can easily determine the percentage of invalid values based on the listed patterns. For more information, see section *Tab panel of the analysis editors*.

Management processes for SQL patterns and regular expressions, including those for Java, are the same. For more information, see section *Managing regular expressions and SQL patterns*.

Some databases do not support regular expressions. To work with such databases, some configuration is necessary before being able to use regular expressions. For more information, see section Managing User-Defined Functions in databases.

9.1.2. Managing User-Defined Functions in databases

The regular expression function is built in several databases, but many other databases do not support it. The databases that natively support regular expressions include: MySQL, PostgreSQL, Oracle 10g, and Ingres while Microsoft SQL server does not, for example.

A different case is when the regular expression function is built in the database but the query template of the regular expression indicator is not defined.

From the **Profiling** perspective of the studio, you can:

- extend the functionality of certain database servers to support the regular expression function. For more information, see section *How to declare a User-Defined Function in a specific database*.
- define the query template for a database that supports the regular expression function. For more information, see section *How to define a query template for a specific database*.

9.1.2.1. How to declare a User-Defined Function in a specific database

The regular expression function is not built into all different database environments. If you want to use the studio to analyze columns against regular expressions in databases that do not natively support regular expressions, you can:

Either:

1. Install the relevant regular expressions libraries on the database. For an example of creating a regular expression function on a database, see appendix *Regular expressions on SQL Server*.

2. Create a query template for the database in the studio. For more information, see section *How to define a query template for a specific database*.

Or:

- Execute the column analysis using the Java engine. In this case, the system will use the Java regular expressions to analyze the specified column(s) and not SQL regular expressions. For more information on the Java engine, see section *Using the Java or the SQL engine*.

9.1.2.2. How to define a query template for a specific database

A query template defines the query logic required to analyze columns against regular expressions. The steps to define a query template in the studio include the following:

- Create a query template for a specific database,

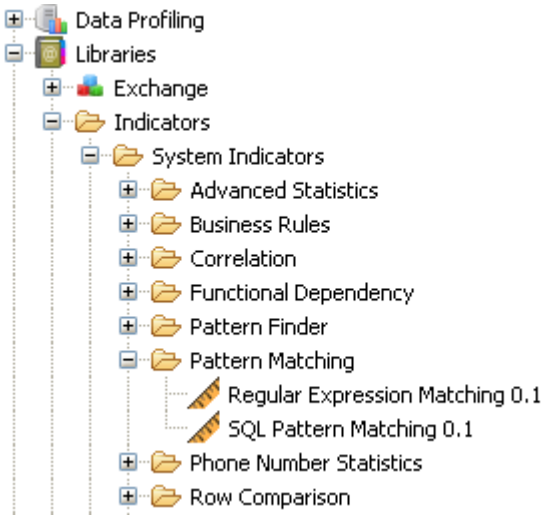
- Set the database-specific regular expression if this expression is not simple enough to be used with all databases.

The below example shows how to define a query template specific for the Microsoft SQL Server database. appendix *Regular expressions on SQL Server* gives a detailed example on how to create a user-defined regular expression function on an SQL server.

To define a query template for a specific database, do the following:

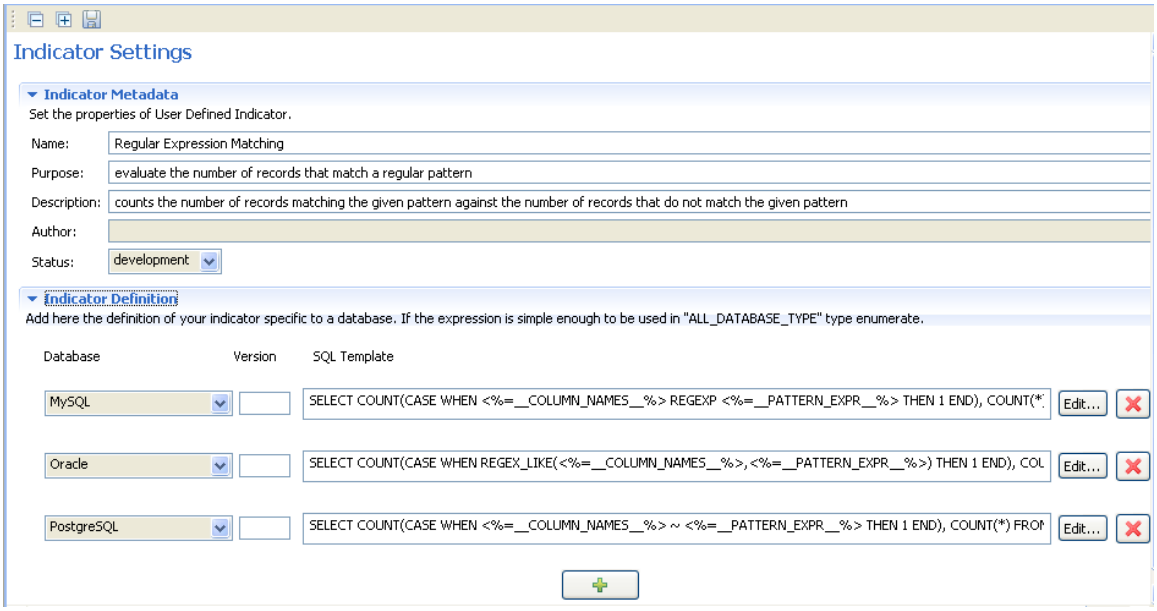
1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators**.

2. Expand **System Indicators** > **Pattern Matching**.



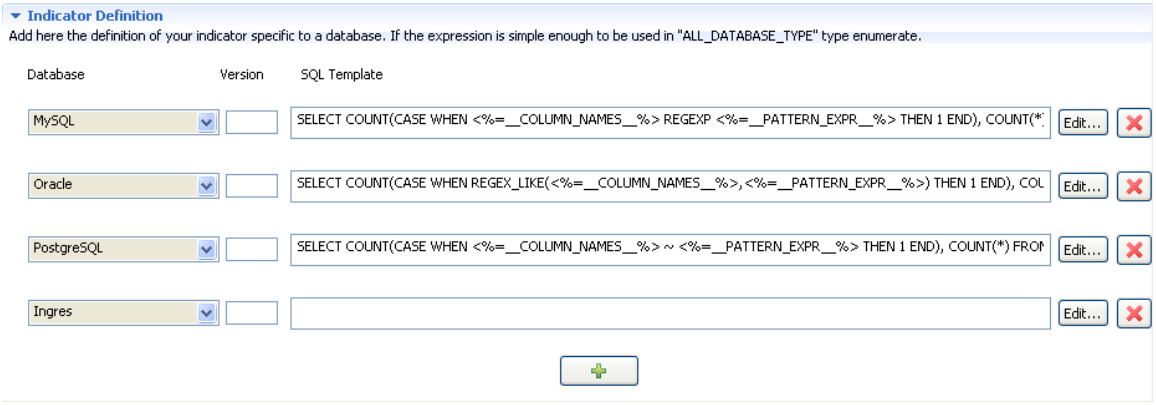
3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

   The corresponding view is displayed to show the indicator metadata and its definition.



You need now to add to the list of databases the database for which you want to define a query template. This query template will compute the regular expression matching.

4. Click the **[+]** button at the bottom of the **Indicator Definition** view to add a field for the new template.

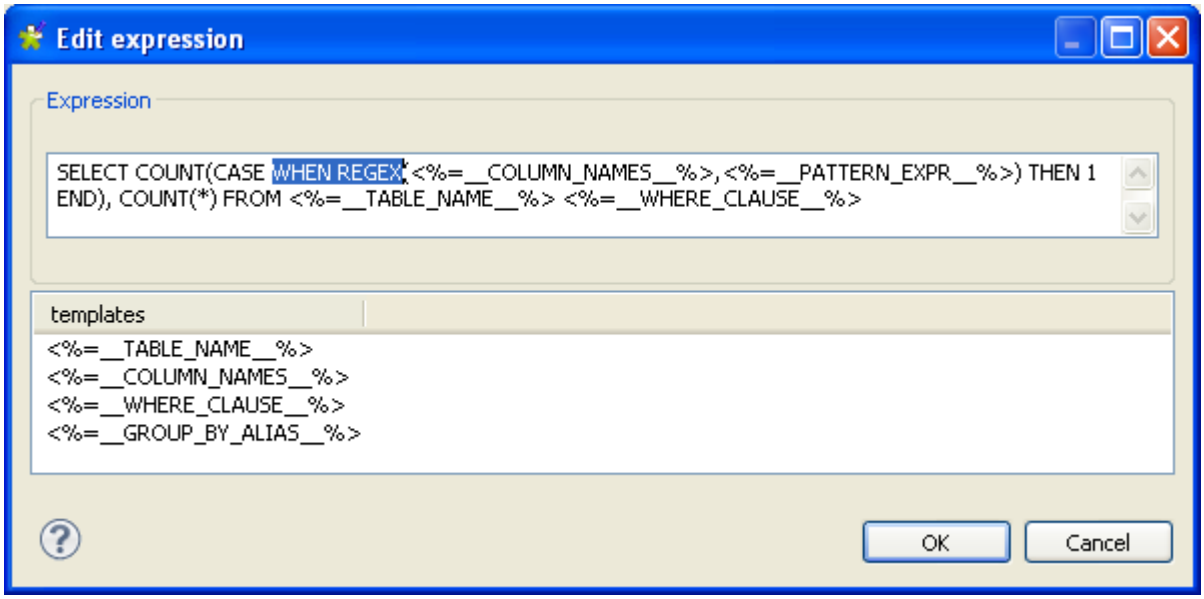


5. In the new field, click the arrow and select the database for which you want to define the template. In this example, select **Ingres**.

6. Copy the indicator definition of any of the other databases.

7. Click the **Edit...** button next to the new field.

   The **[Edit expression]** dialog box is displayed.



8. Paste the indicator definition (template) in the **Expression** box and then modify the text after `WHEN` in order to adapt the template to the selected database. In this example, replace the text after `WHEN` with `WHEN REGEX`.

9. Click **OK** to proceed to the next step. The new template is displayed in the field.

10. Click the save icon on top of the editor to save your changes.

You have finalized creating the query template specific for the **Ingres** database. You can now start analyzing the columns in this database against regular expressions.

If the regular expression you want to use to analyze data on this server is simple enough to be used with all databases, you can start your column analyses immediately. If not, you must edit the definition of the regular expression to work with this specific database, **Ingres** in this example.

For more information on how to set the database-specific regular expression definition, see section *How to edit a regular expression or an SQL pattern* and section *How to duplicate a regular expression or an SQL pattern*.

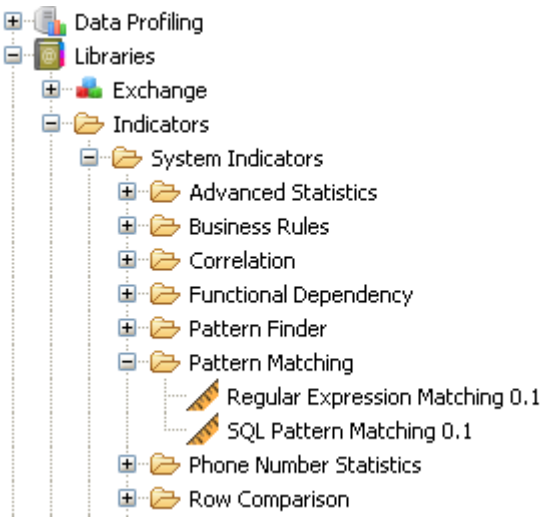
9.1.2.3. How to edit a query template

You can edit the query template you create for a specific database.

To edit a query template for a specific database, do the following:

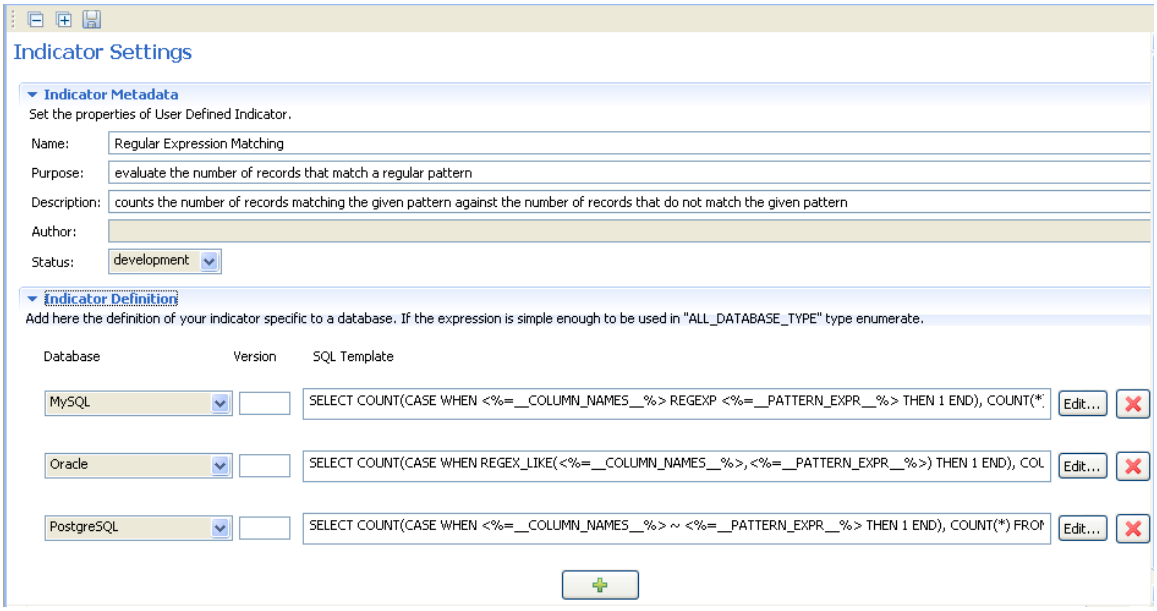
1. In the **DQ Repository** tree view, expand **Libraries > Indicators**.

2. Expand **System Indicator > Pattern Matching**.



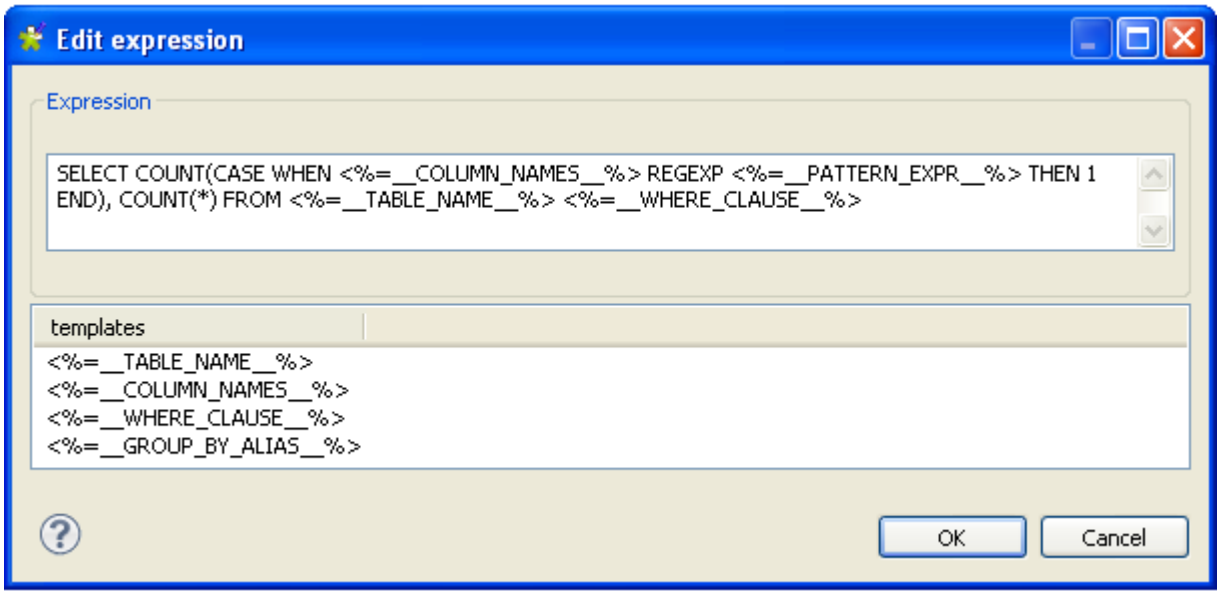
3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

   The corresponding view is displayed to show the indicator metadata and its definition.



4. Click the Edit... button next to the database for which you want to edit the query template.

   The **[Edit expression]** dialog box is displayed.



5. In the **Expression** area, edit the regular expression template as required and then click OK to close the dialog box and proceed to the next step.

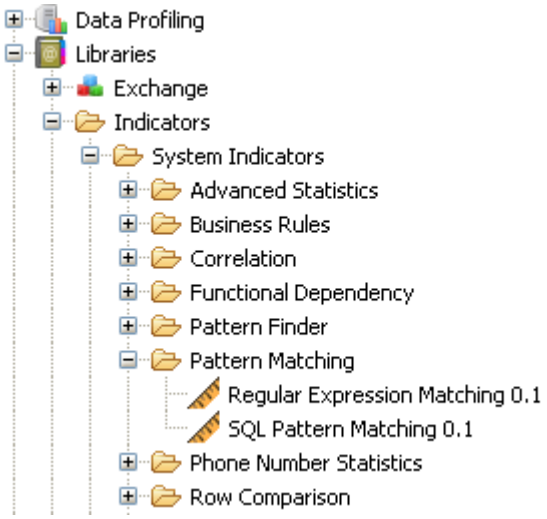
   The regular expression template is modified accordingly.

9.1.2.4. How to delete a query template

You can delete the query template you create for a specific database.

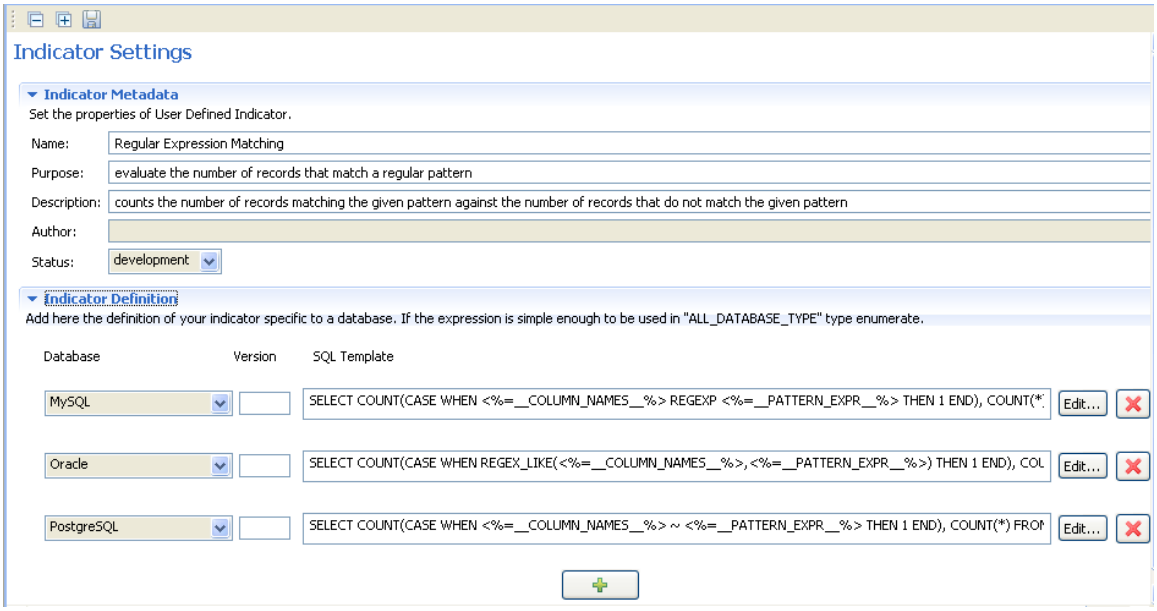
To delete a query template for a specific database, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators**.
2. Expand **System Indicators** > **Pattern Matching**.



3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

   The corresponding view is displayed to show the indicator metadata and its definition.



4. Click the button next to the database for which you want to delete the query template.

   The selected query template is deleted from the list in the Indicator definition view.

9.1.3. Adding regular expressions and SQL patterns to column analyses

You can use regular expressions and SQL patterns in column analyses in order to check all existing data in the analyzed columns against these expressions and patterns. For more information, see section *How to add a regular expression or an SQL pattern to a column analysis*.

You can also edit the regular expression or SQL pattern parameters after attaching it to a column analysis. For more information, see section *How to edit a pattern in the column analysis*.

After the execution of the column analysis that uses a specific expression or pattern, you can:

- access a list of all valid/invalid data in the analyzed column. For more information, see section *How to view the data analyzed against patterns*.

9.1.4. Managing regular expressions and SQL patterns

The management procedures of regular expressions and SQL patterns include operations like creating, testing, duplicating, importing and exporting.

The sections below explain in detail each of the management option for regular expressions and SQL patterns. Management processes for both types of patterns are exactly the same.

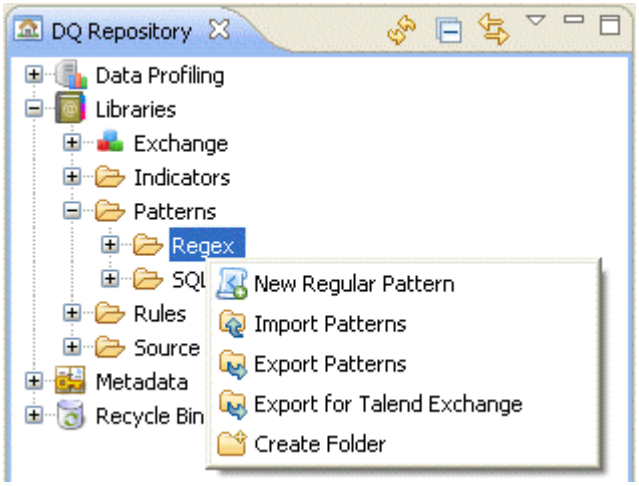
9.1.4.1. How to create a new regular expression or SQL pattern

You can create new regular expressions or SQL patterns, including those for Java to be used in column analyses.

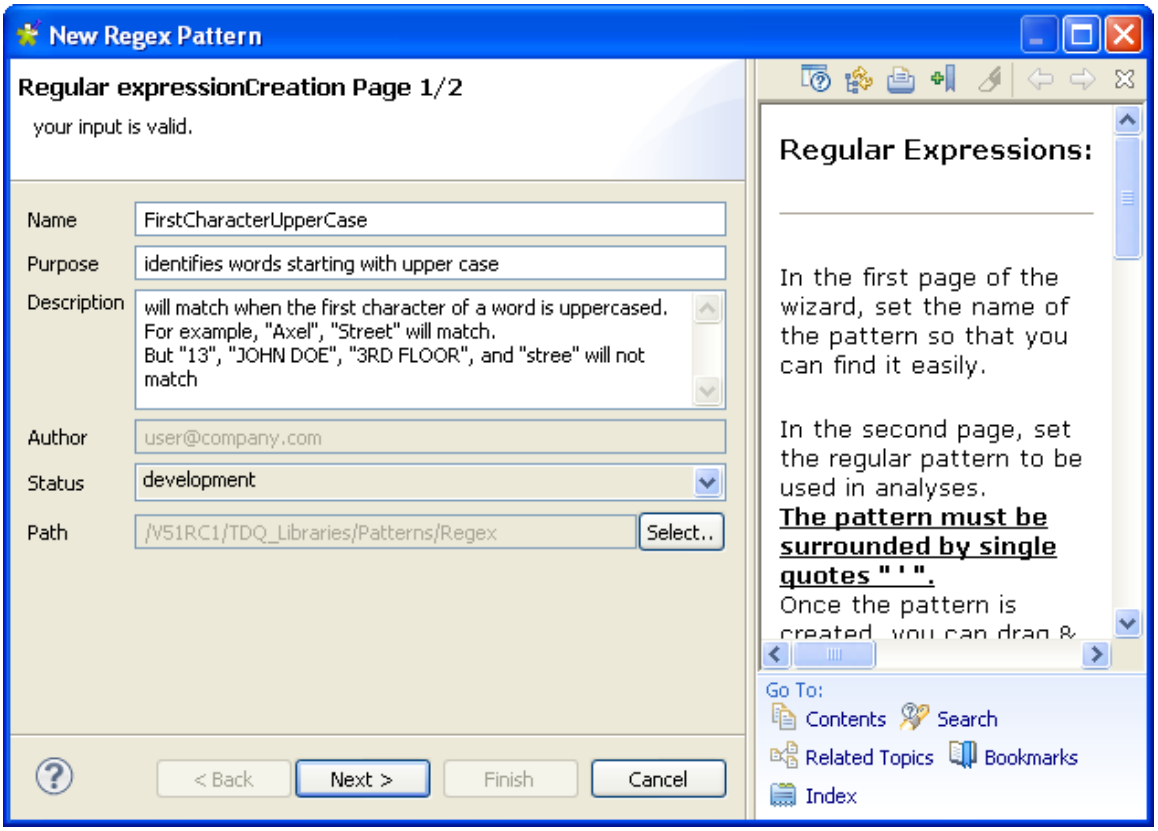
Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to create a regular expression. You can follow the same steps to create an SQL pattern.

To create a new pattern, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns**, and then right-click **Regex**.



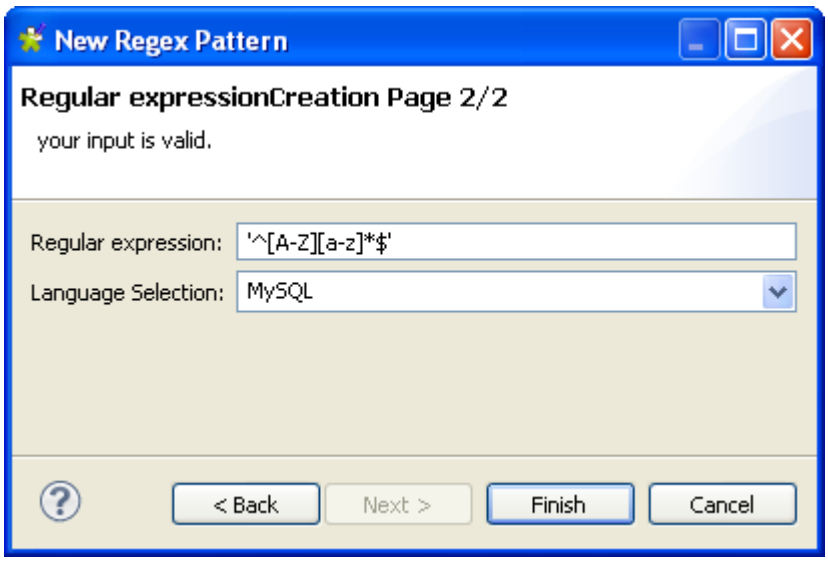
2. From the contextual menu, select **New regular pattern** to open the corresponding wizard.



When you open the wizard, a help panel automatically opens with the wizard. This help panel guides you through the steps of creating new regular patterns.

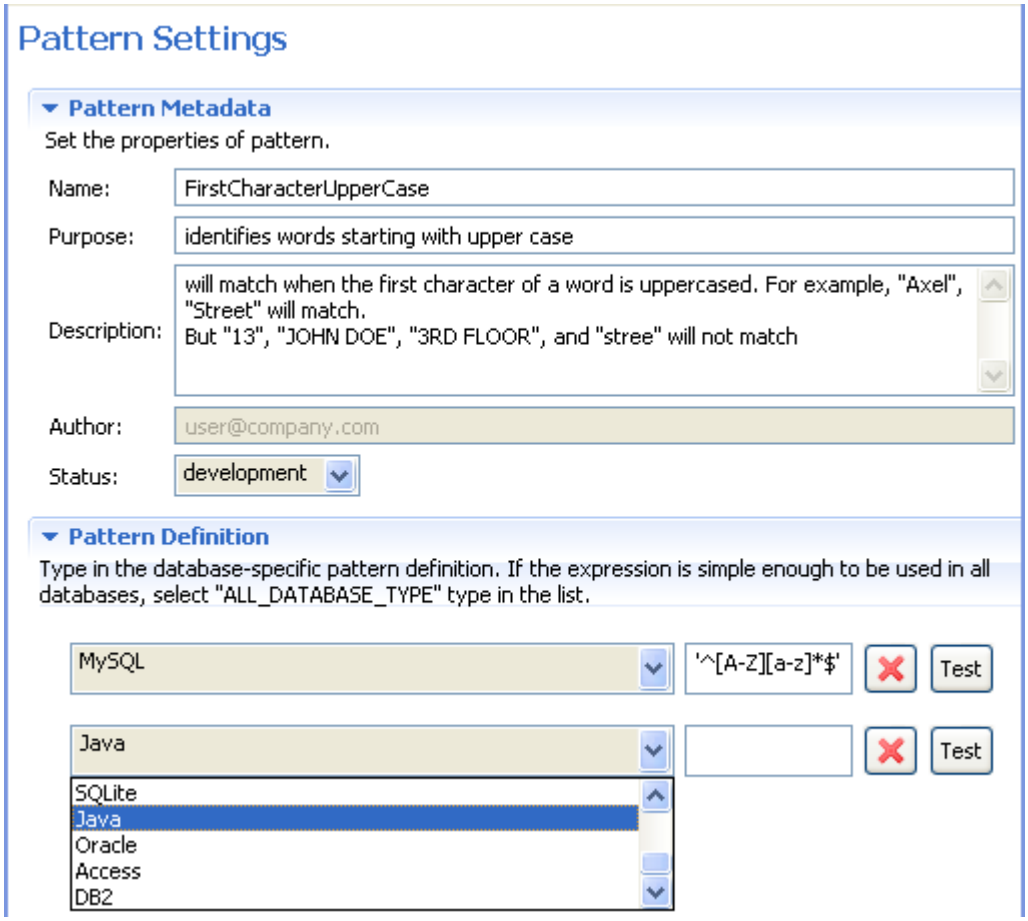
3. In the **Name** field, enter a name for this new regular expression.

4. If required, set other metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



5. In the **Regular expression** field, enter the syntax of the regular expression to be created. The regular expression must be surrounded by single quotes.
6. From the **Language Selection** list, select the language (a specific database or Java).
7. Click **Finish** to close the dialog box.

   A sub-folder for this new regular expression is listed under the **Regex** folder in the **DQ Repository** tree view, and the pattern editor opens with the defined metadata and the defined regular expression.

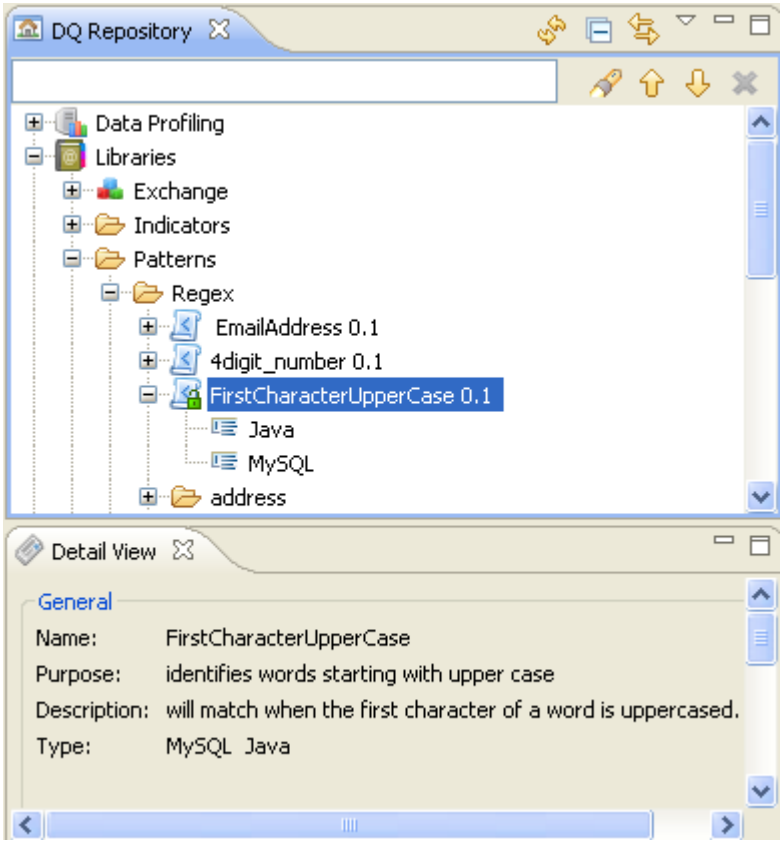


8. In the **Pattern Definition** view, click the **[+]** button and add as many regular expressions as necessary in the new pattern.

You can define the regular expressions specific to any of the available databases or specific to Java.

If the regular expression is simple enough to be used in all databases, select `Default` from the list.

Sub-folders labeled with the specified database types or Java are listed below the name of the new pattern under the **Patterns** folder in the **DQ Repository** tree view.



9. Save the new pattern.

   Once the pattern is created, you can drop it directly onto a database column in the open analysis editor.

10. If required, click the pattern name to display its detail in the **Detail View** in the Studio.

In the pattern editor, you can click **Test** next to the regular expression to test the regular pattern definition. For more information, see section *How to test a regular expression in the Pattern Test View*. Also, from the **[Pattern Test View]**, you can create a new pattern based on the regular expression you are testing. For further information, see xsection *How to create a new pattern from the Pattern Test View* .

9.1.4.2. How to test a regular expression in the Pattern Test View

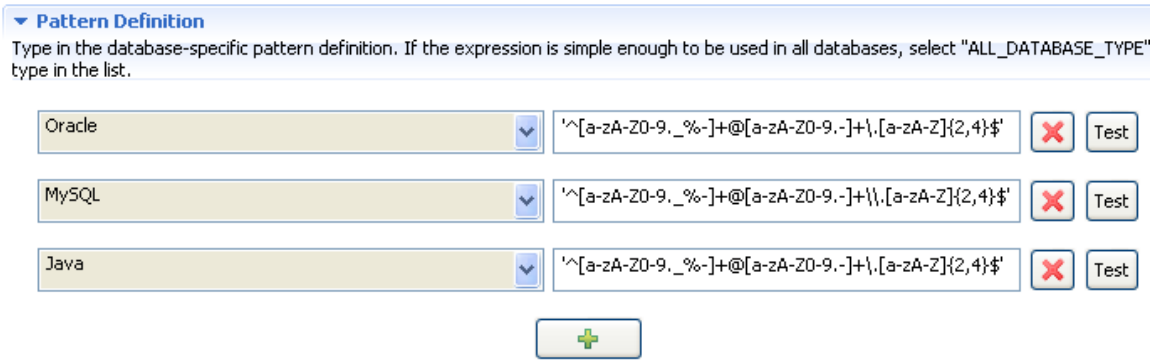
It is possible to test character sequences against a predefined or newly created regular expression.

Prerequisite(s): At least one database connection is set in the **Profiling** perspective of the studio.

To test a character sequence against a regular expression, do the following:

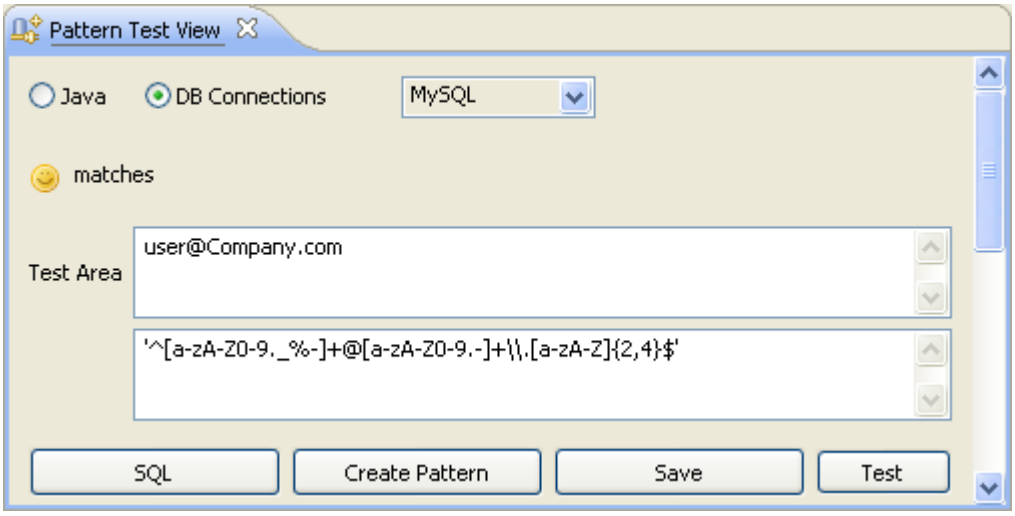
1. Follow the steps outlined in section *How to create a new regular expression or SQL pattern* to create a new regular expression.

2. In the open pattern editor, click **Pattern Definition** to open the relevant view.



3. Click the **Test** button next to the definition against which you want to test a character sequence to proceed to the next step.

   The test view is displayed in the Studio showing the selected regular expression.



4. In the **Test Area**, enter the character sequence you want to check against the regular expression

5. From the **DB Connection** list, select the database in which you want to use the regular expression.

   If you select to test a regular expression in Java, the **Java** option will be selected by default and the **DB Connections** option and list will be unavailable in the test view.

6. Click **Test**.

   An icon is displayed in the upper left corner of the view to indicate if the character sequence matches or does not match the selected pattern definition.

7. If required, modify the regular expression according to your needs and then click **Save** to save your modifications.

   The pattern definition is modified accordingly in the pattern editor.

You can create/modify patterns directly from the **Pattern Test View** via the **Create Pattern** button. For further information, see section *How to create a new pattern from the Pattern Test View*

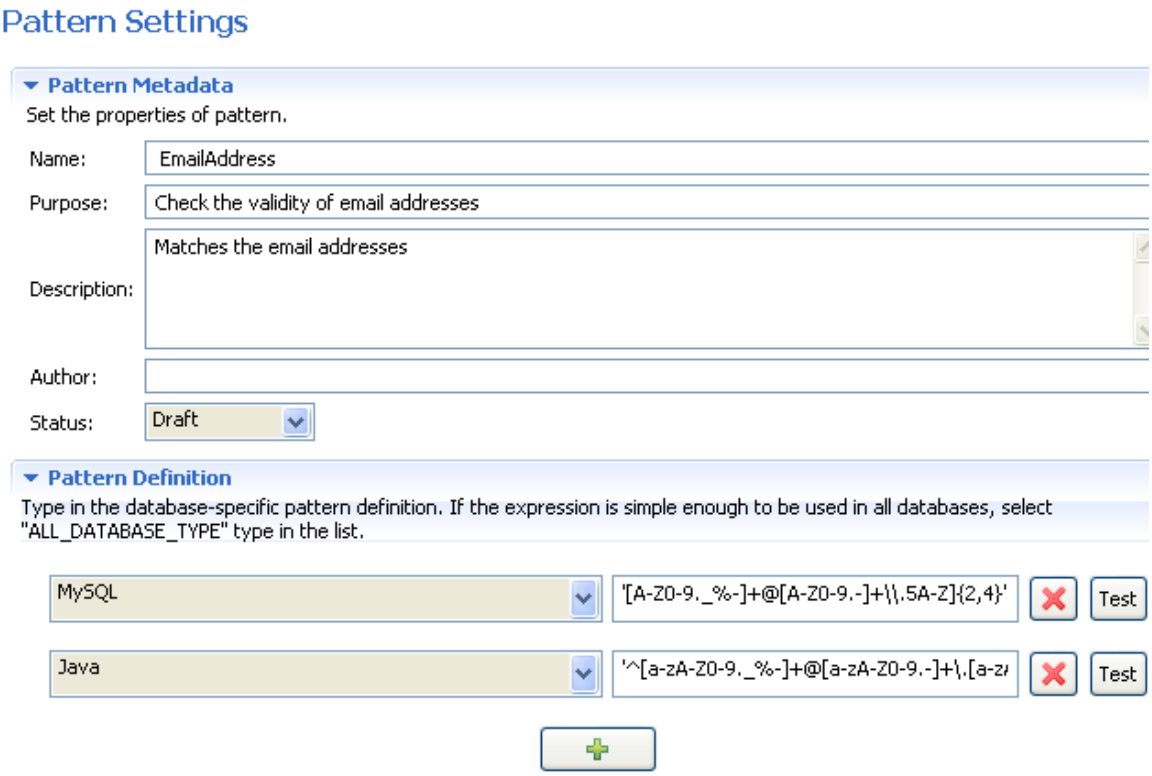
9.1.4.3. How to create a new pattern from the Pattern Test View

You can create your own customized patterns from the **[Pattern Test View]**.

The advantage of creating a pattern from this view is that you can create your customized pattern based on an already tested regular expression. All you need to do is to customize the expression definition according to your needs and save it to create a new pattern.

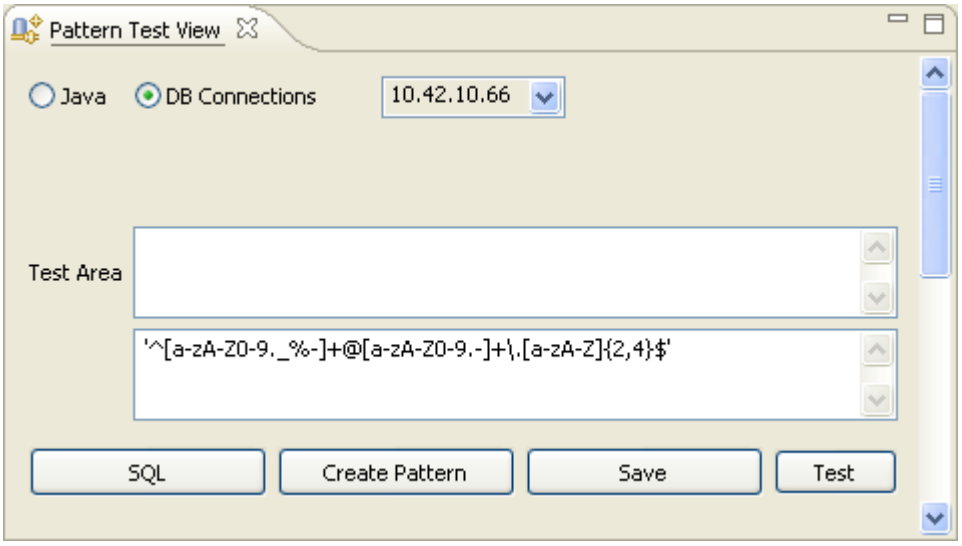
To create a new pattern based on a predefined or a newly created regular expression, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns > Regex** and double-click the pattern you want to use to create your customized pattern to open the pattern editor.



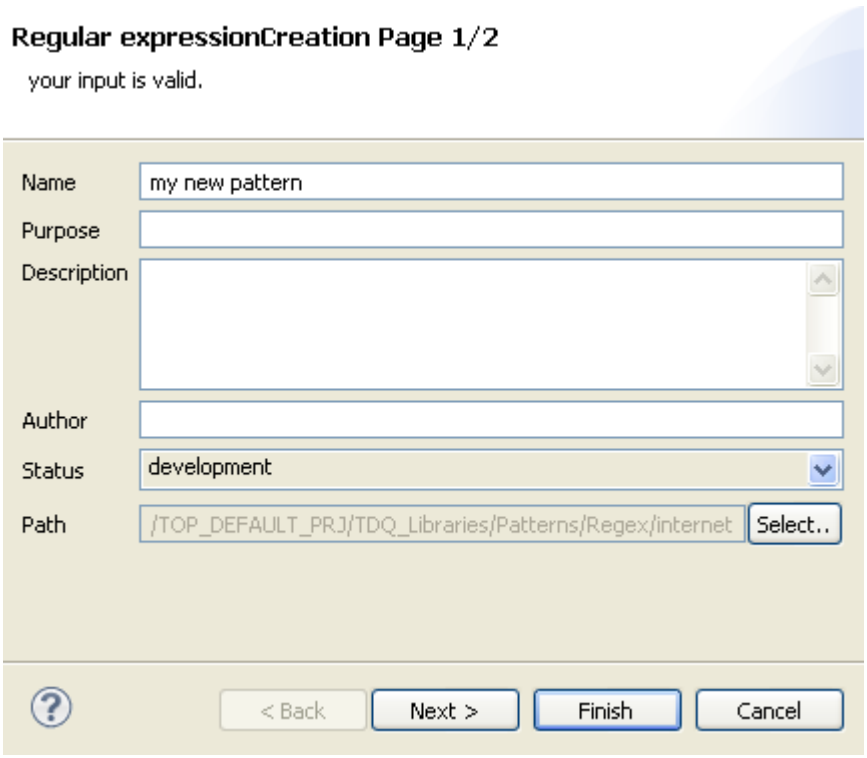
2. Click **Test** next to the definition you want to use as a base to create the new pattern.

   The **[Pattern Test View]** is opened on the definition of the selected regular expression.



3. If required, test the regular expression through entering text in the **Test Area**. For further information, see section *How to test a regular expression in the Pattern Test View*.

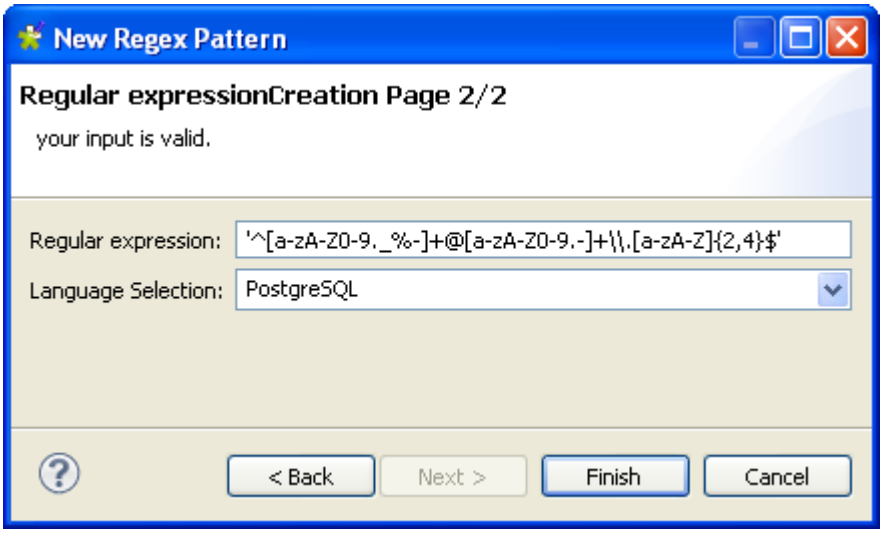
4. Click **Create Pattern** to open the **[New Regex pattern]** wizard.



5. In the **Name** field, enter a name for this new regular expression.

6. If required, set other metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.

   The definition of the initial regular expression is already listed in the **Regular expression** field.

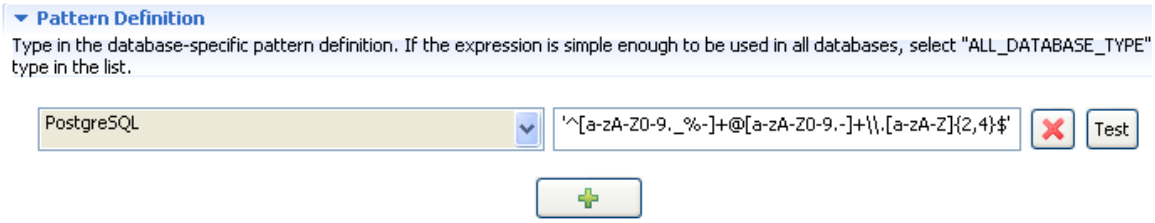


7. Customize the syntax of the initial regular expression according to your needs. The regular expression definition must be surrounded by single quotes.

8. From the **Language Selection** list, select the database in which you want to use the new regular expression.

9. Click **Finish** to close the wizard.

   A sub-folder for the new pattern is listed under the **Regex** folder in the same file of the initial regular pattern. The pattern editor opens on the pattern metadata and pattern definition.



Once the new pattern is created, you can drop it onto a column in the open analysis editor.

9.1.4.4. How to generate a regular expression from the Date Pattern Frequency Table

You can generate a regular pattern from the results of an analysis that uses the **Date Pattern Frequency Table** indicator on a date column.

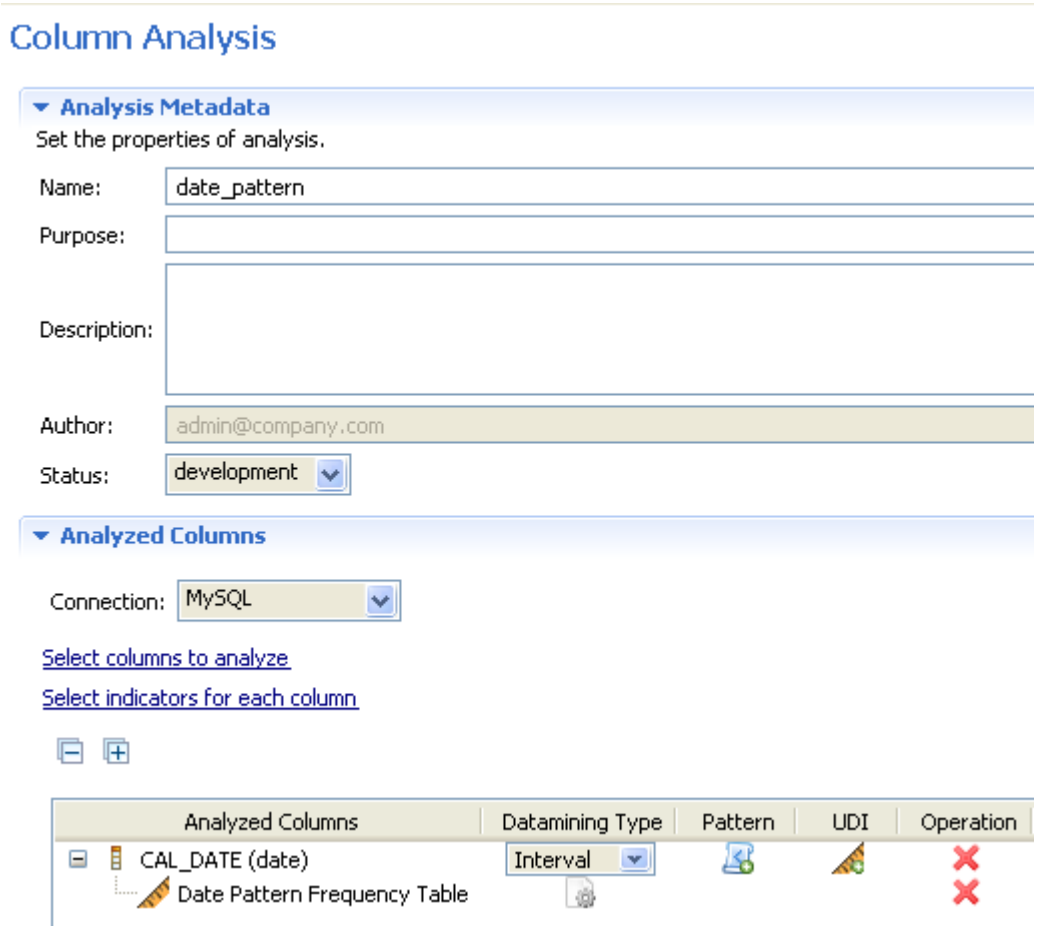
Prerequisite(s): In the **Profiling** perspective of the studio, a column analysis is created on a date column using the **Date Pattern Frequency Table** indicator.

To be able to use the Date Pattern Frequency Table indicator on date columns, you must set the execution engine to Java in the Analysis Parameter view of the column analysis editor. For more information on execution engines, see section Using the Java or the SQL engine.

For more information on how to create a column analysis, see section *Analyzing columns in a database*.

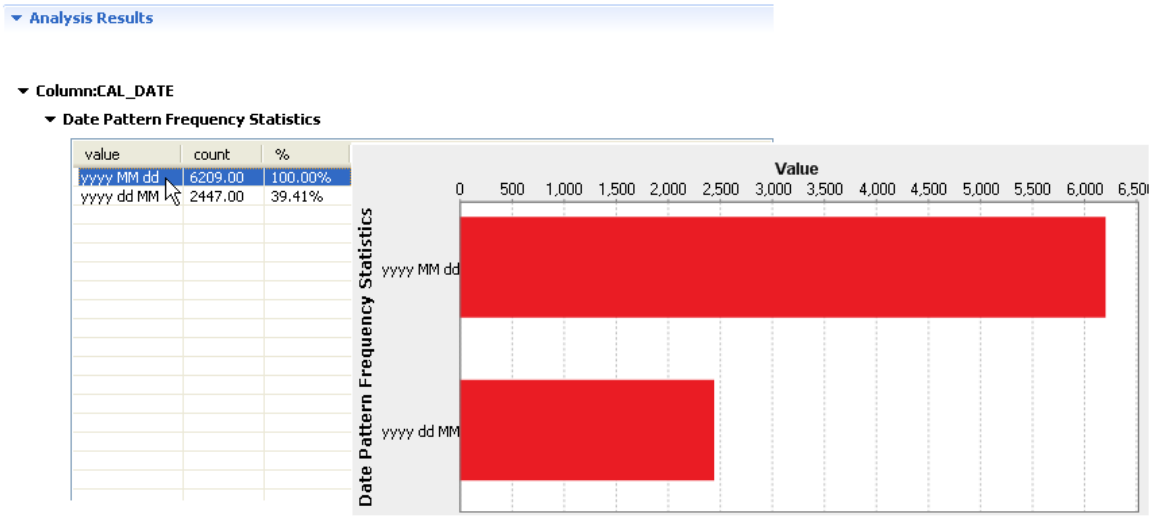
To generate a regular expression from the results of a column analysis, do the following:

1. In the **DQ Repository** tree view, right-click the column analysis that uses the date indicator on a date column.
2. Select **Open** from the contextual menu to open the corresponding analysis editor.



3. Press **F6** to execute the analysis and display the analysis results in the **Graphics** panel to the right of the Studio.

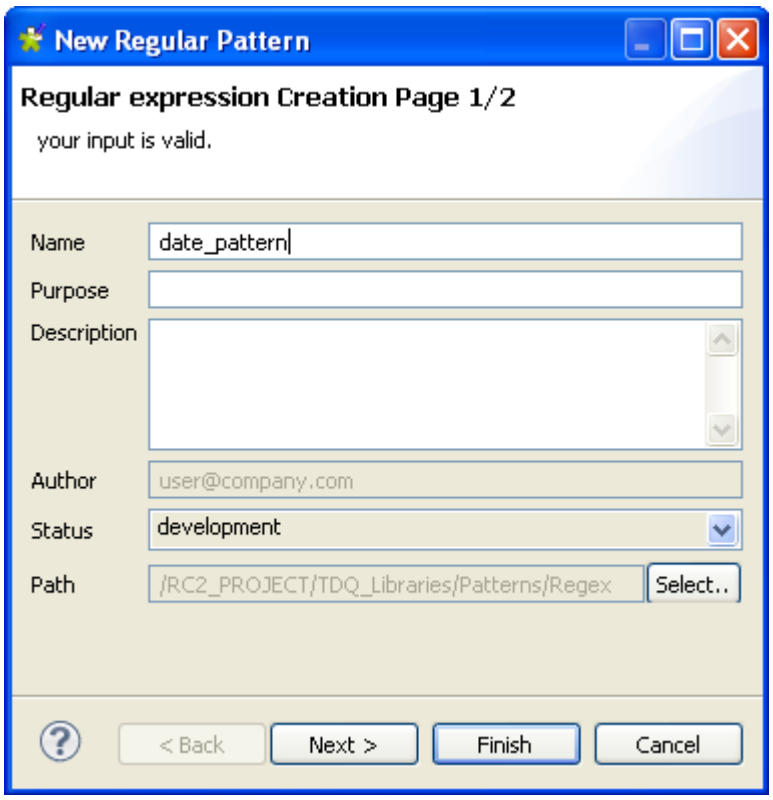
4. At the bottom of the editor, click the **Analysis Results** tab to display a more detailed result view.



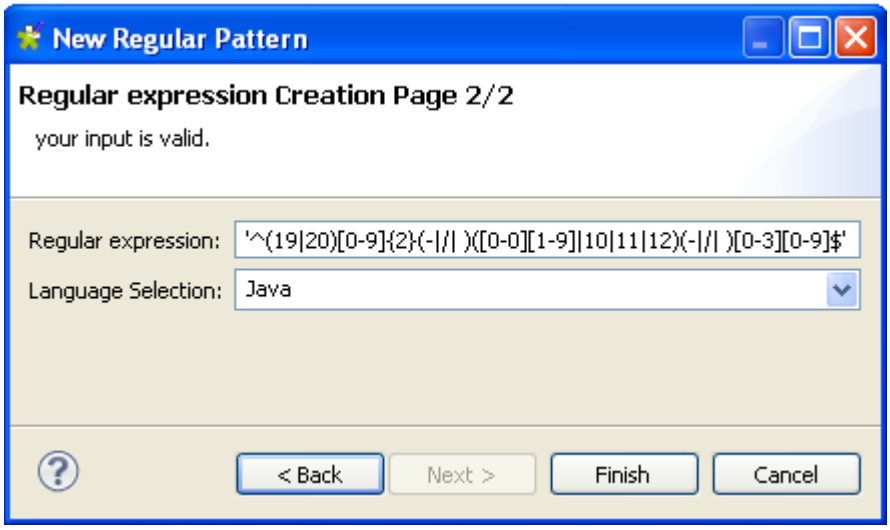
In this example, 100.00% of the date values follow the pattern `yyyy MM dd` and 39.41% follow the pattern `yyyy dd MM`.

5. Right-click the date value for which you want to generate a regular expression and select **Generate Regular Pattern** from the contextual menu.

The **[New Regular Pattern]** dialog box is displayed.



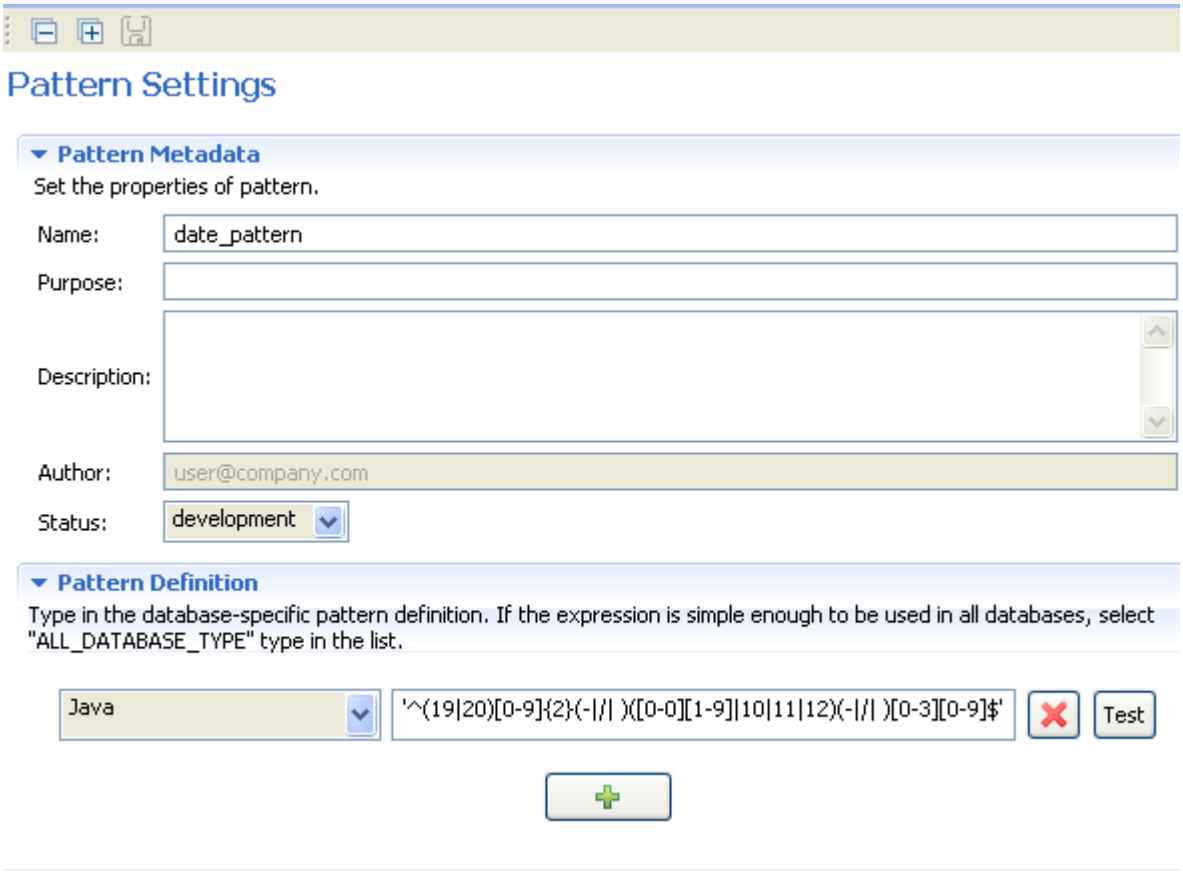
6. Click **Next** to proceed to the next step.



The date regular expression is already defined in the corresponding field.

7. Click **Finish** to proceed to the next step.

The pattern editor opens with the defined metadata and the generated pattern definition.



The new regular expression is listed under **Pattern** > **Regex** in the **DQ Repository** tree view. You can drag it onto any date column in the analysis editor.

8. If required, click the **Test** button to test a character sequence against this date regular expression as outlined in the following section.

9.1.4.5. How to edit a regular expression or an SQL pattern

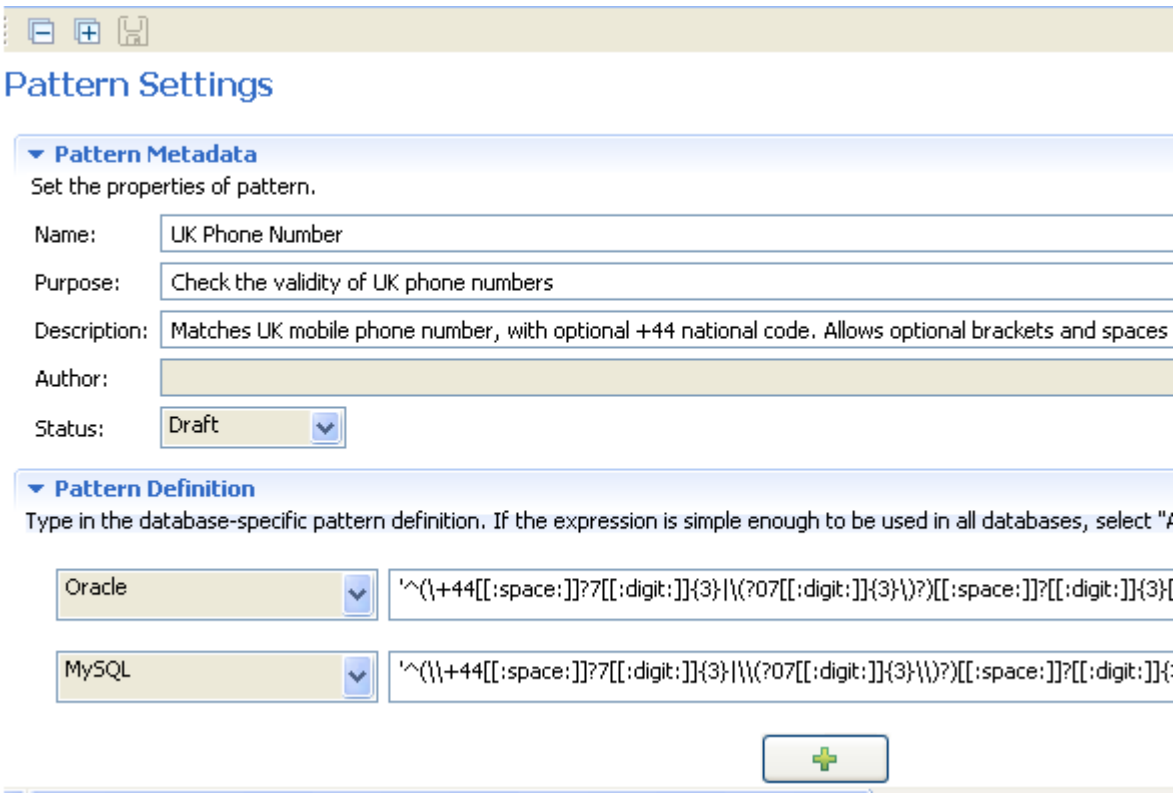
You can open the editor of any regular expression or SQL pattern to check its settings and/or edit its definition in order to:

- adapt it to a specific database type, or
- adapt it to a specific use.

To open/edit a regular expression or an SQL pattern, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Patterns**.
2. Browse through the regular expression or SQL pattern lists to reach the expression or pattern you want to open/edit.
3. Right-click its name and select **Open** from the contextual menu.

   The pattern editor opens displaying the regular expression or SQL pattern settings.



4. Modify the pattern metadata, if required, and then click **Pattern Definition** to display the relevant view. In this view, you can: edit pattern definition, change the selected database and add other patterns specific to available databases through the **[+]** button.
5. If the regular expression or SQL pattern is simple enough to be used in all databases, select `Default` in the list.
6. Click the save icon on top of the editor to save your changes.

You can test regular expressions before start using them against data in the specified database. For more information, see section *How to test a regular expression in the Pattern Test View*.

When you edit a regular expression or an SQL pattern, make sure that your modifications are suitable for all the analyses that may be using this regular expression or SQL pattern.

9.1.4.6. How to export regular expressions or SQL patterns

You can export regular expressions and SQL patterns and store them locally in a csv file. For more information about the content lay out of the csv file, see section *How to import regular expressions or SQL patterns from a csv file*.

Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to export regular expressions. You can follow the same steps to export SQL patterns.

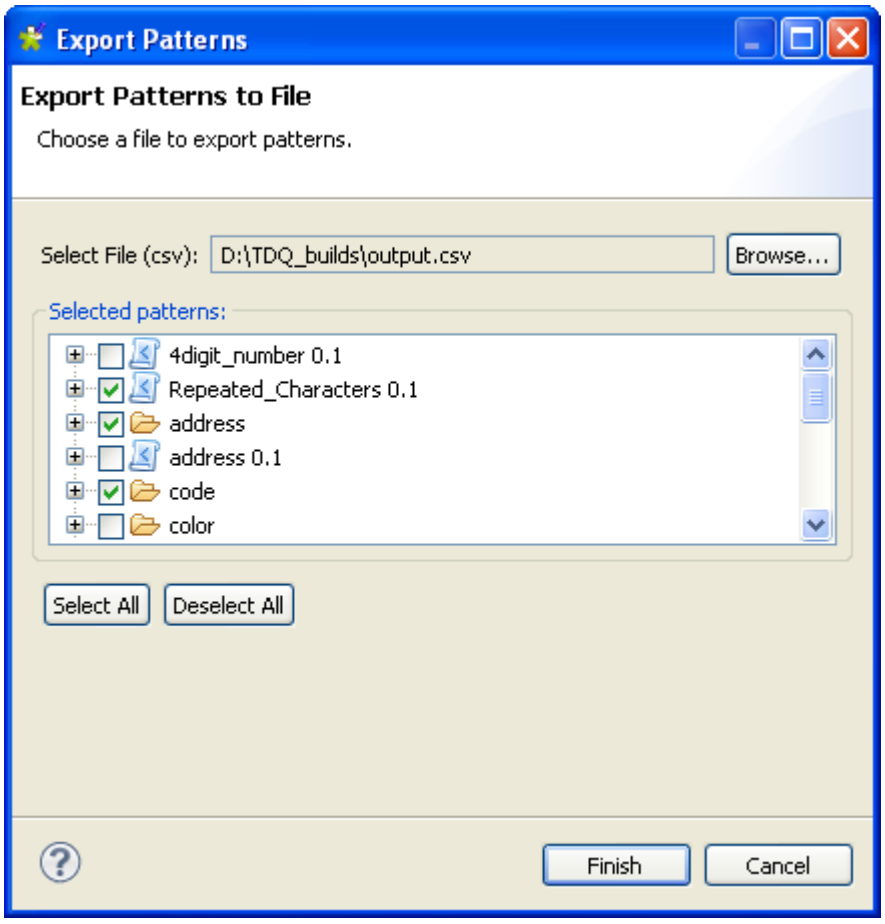
How to export expressions or patterns to a csv file

To export regular expressions to a csv file, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns**, and then right-click **Regex**.

2. From the contextual menu, select **Export Patterns**.

   The **[Export Patterns]** wizard opens.



3. Browse to the csv file where to save the regular expressions.

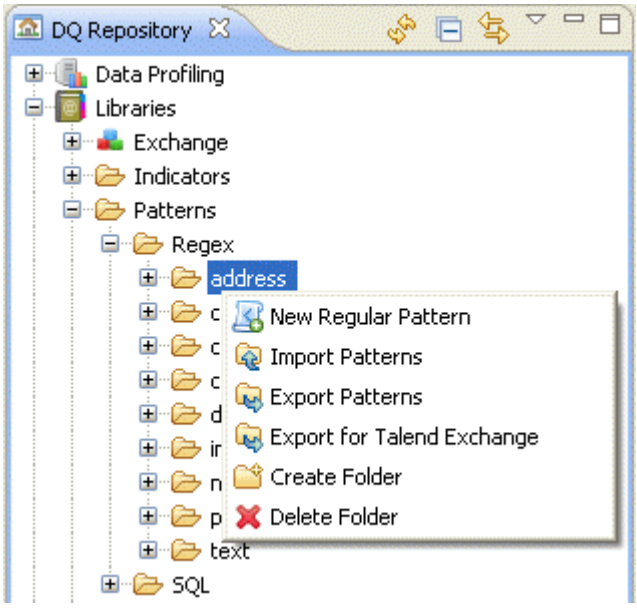
4. Click **Select All** to select all listed regular expressions or select the check boxes of the regular expressions you want to export to the csv file.

5. Click **Finish** to close the wizard.

   All exported regular expressions are saved in the defined csv file.

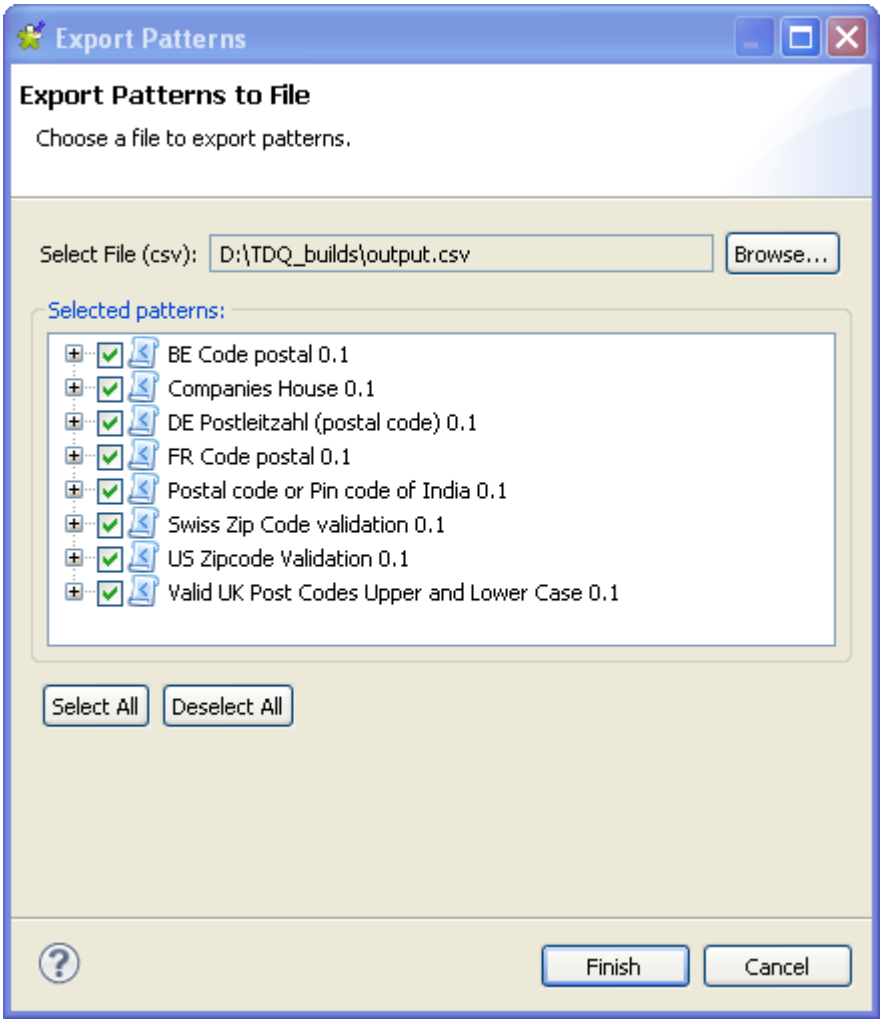
To export a single regular expression family to a csv file, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns**, and then browse to the regular expression family you want to export.



2. From the contextual menu, select **Export Patterns**.

   The **[Export Patterns]** wizard opens.



3. Click **Select All** to select all the check boxes of the regular expressions or select the check boxes of the regular expressions you want to export to the csv file.

4. Click **Finish** to close the wizard.

All exported regular expressions are saved in the defined csv file.

How to export expressions or patterns to Talend Exchange

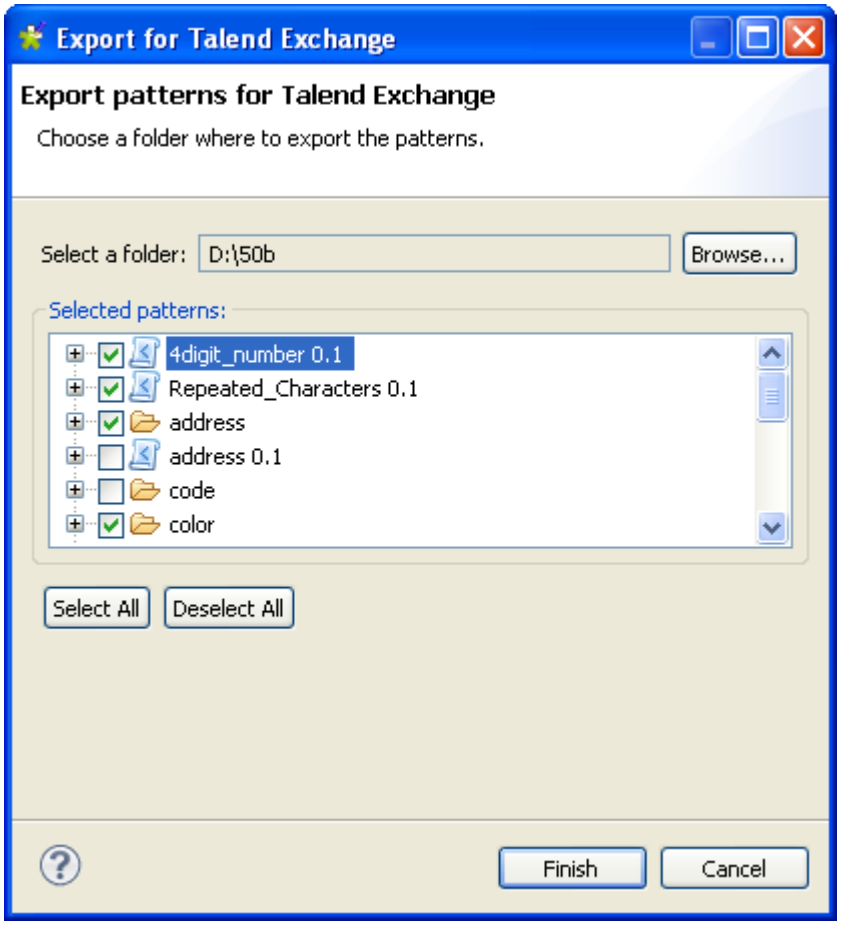
You can export regular expressions or SQL patterns from your current version of studio to **Talend Exchange** where you can share them with other users.

Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to export regular expressions to **Talend Exchange**. You can follow the same steps to export SQL patterns to **Talend Exchange**.

To export regular expressions to **Talend Exchange**, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns**.
2. Right-click **Regex** and select **Export for Talend Exchange**.

   The **[Export for Talend Exchange]** wizard is displayed.

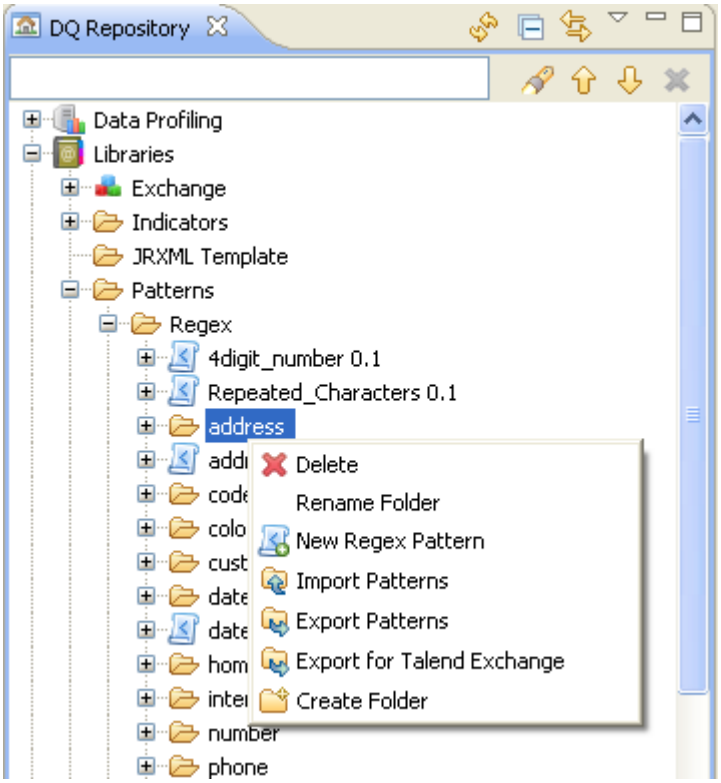


3. Browse to the folder where to save regular expressions.
4. Click Select All to select all the regular expressions in the list or select the check boxes of the regular expressions you want to export to the specified folder.
5. Click **Finish** to close the wizard.

A distinct csv file is created for each exported regular expression. Each csv file is compressed as a zip. All these zip files are saved in the defined folder. You need now to upload them to **Talend Exchange** at http://www.talendforge.org/exchange/top/help_guest.php. For information about how to upload a file to **Talend Exchange**, see *Talend Open Studio for Data Integration User Guide*.

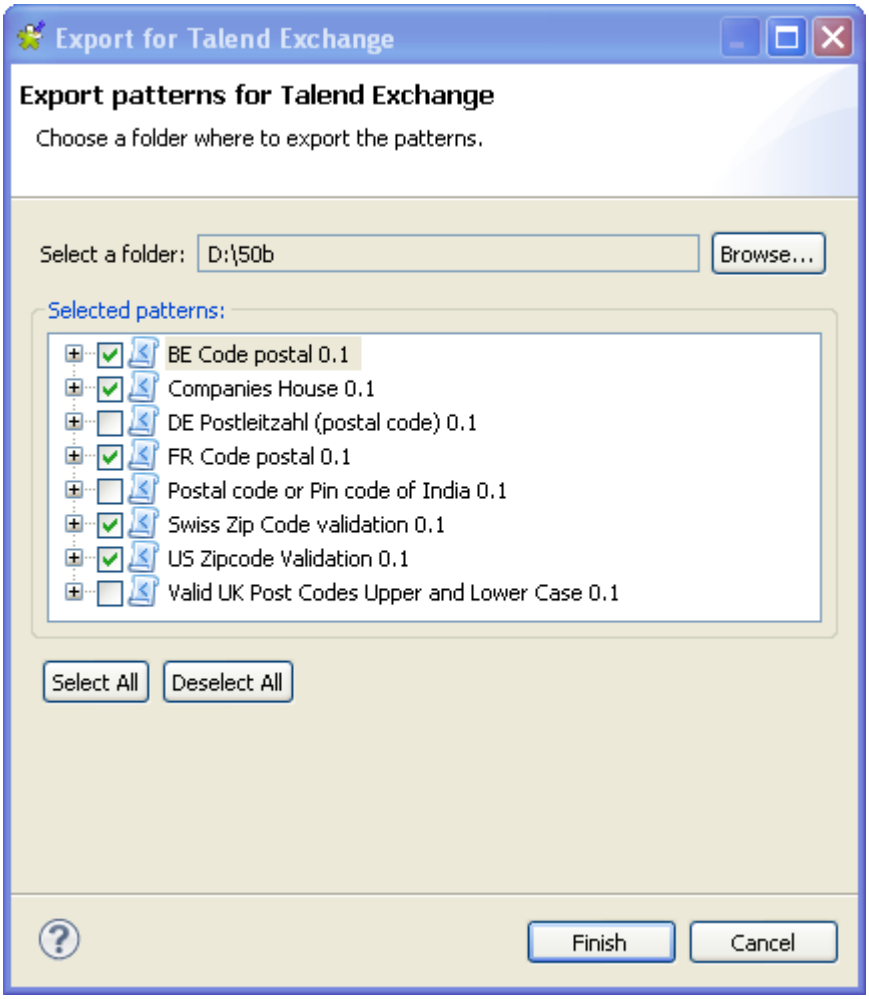
To export a single regular expression family to **Talend Exchange**, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns**, and then browse to the regular expression you want to export.



2. Right-click it and then select **Export for Talend Exchange** from the contextual menu.

   The **[Export for Talend Exchange]** wizard opens.



3. Click **Select All** to select all the regular expressions in the list, or select the check boxes of the regular expressions or SQL patterns you want to export to the folder.

4. Click **Finish** to close the wizard.

A distinct csv file is created for each exported regular expression or SQL pattern. Each csv file is compressed as zip. All these zip files are saved in the defined folder.

You need now to upload them to **Talend Exchange** at http://www.talendforge.org/exchange/top/help_guest.php. For information about how to upload a file to **Talend Exchange**, see *Talend Open Studio for Data Integration User Guide*.

9.1.4.7. How to import regular expressions or SQL patterns

You can import the regular expressions or SQL patterns stored locally in a csv file. The csv file must have 11 columns laid out as follows:

Column name	Description
Label	the label of the pattern (must not be empty)
Purpose	the purpose of the pattern (can be empty)
Description	the description of the pattern (can be empty)
Author	the author of the regular expression (can be empty)
Relative Path	the relative path to the root folder (can be empty)
All DB Regular	the regular expression applicable to all databases (can be empty)

Column name	Description
DB2 Regexp	the regular expression applicable to DB2 databases (can be empty)
MySQL Regexp	the regular expression applicable to MySQL databases (can be empty)
Oracle Regexp	the regular expression applicable to Oracle databases (can be empty)
PostgreSQL Regexp	the regular expression applicable to PostgreSQL databases (can be empty)
SQL Server Regexp	the regular expression applicable to SQL Server databases (can be empty)

Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to import regular expressions. You can follow the same steps to import SQL patterns.

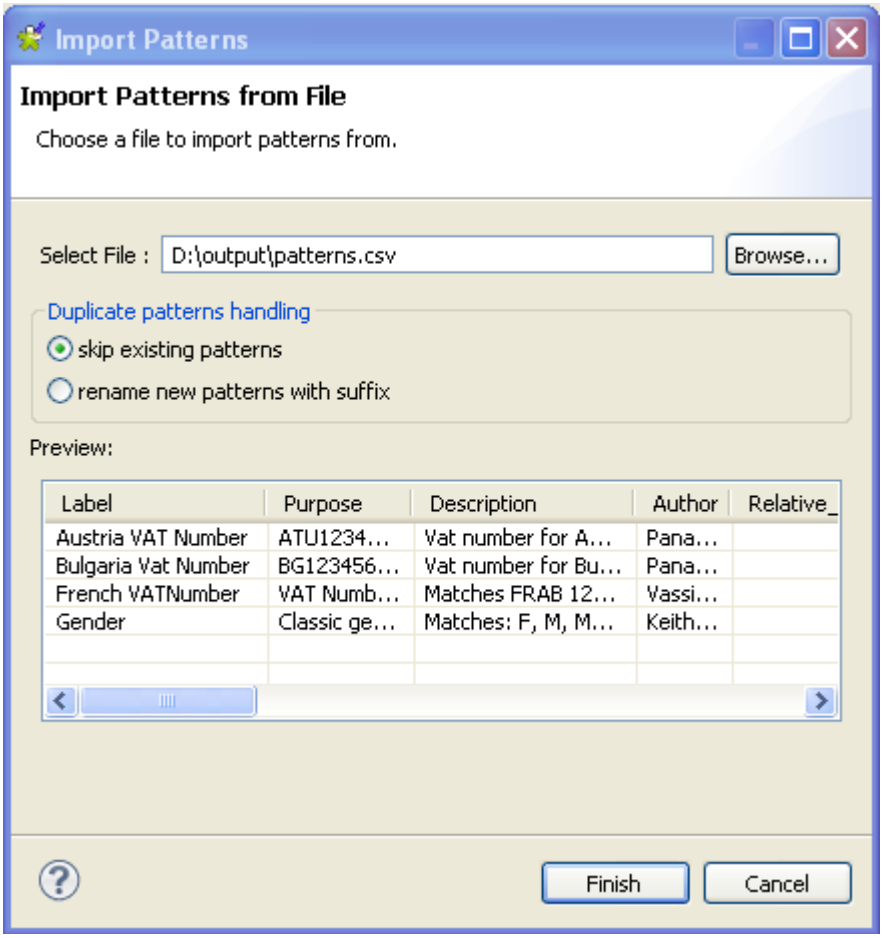
How to import regular expressions or SQL patterns from a csv file

Prerequisite(s): The csv file is stored locally.

To import regular expressions from a csv file, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns**.
2. Right-click **Regex** and select **Import patterns**.

   The **[Import Patterns]** wizard opens.



3. Browse to the csv file holding the regular expressions.
4. In the **Duplicate patterns handling** area, select:

Option	To...

skip existing patterns	import only the regular expressions that do not exist in the corresponding lists in the **DQ Repository** tree view. A warning message is displayed if the imported patterns already exist under the **Patterns** folder.
rename new patterns with suffix	identify each of the imported regular expressions with a suffix. All regular expression will be imported even if they already exist under the **Patterns** folder.

5. Click **Finish** to close the wizard.

All imported regular expressions are listed under the **Regex** folder in the **DQ Repository** tree view.

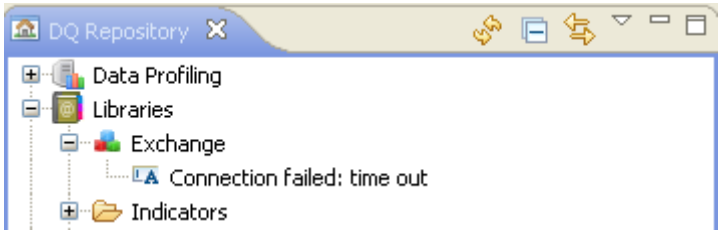
A warning icon next to the name of the imported regular expression or SQL pattern in the tree view identifies that it is not correct. You must open the expression or the pattern and try to figure out what is wrong. Usually, problems come from missing quotes. Check your regular expressions and SQL patterns and ensure that they are encapsulated in single quotes.

How to import regular expressions or SQL patterns from Talend Exchange

You can import regular expressions or SQL patterns from **Talend Exchange** to your current version of studio and use them on analyzed columns.

Prerequisite(s): Your network is up and running.

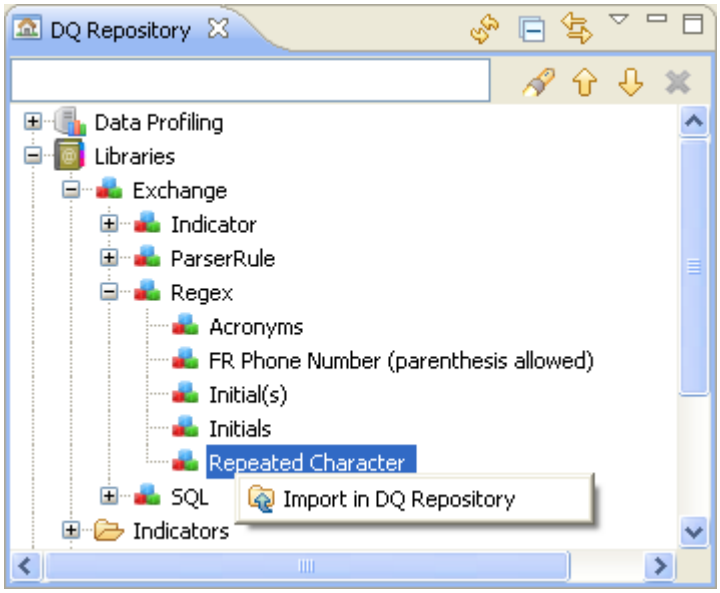
If you have connection problems, you will not be able to access any of the regular expressions or SQL patterns under the **Exchange** node in the **DQ Repository** tree view.



Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to import regular expressions from **Talend Exchange**. You can import SQL patterns following the same steps.

To import regular expressions from **Talend Exchange**, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Exchange**.
2. Under **Exchange**, expand **Regex** and right-click the name of the pattern you want to import.



You will have access only to versions that are compatible with the version of your current Studio.

3. Select **Import in DQ Repository** from the contextual menu.

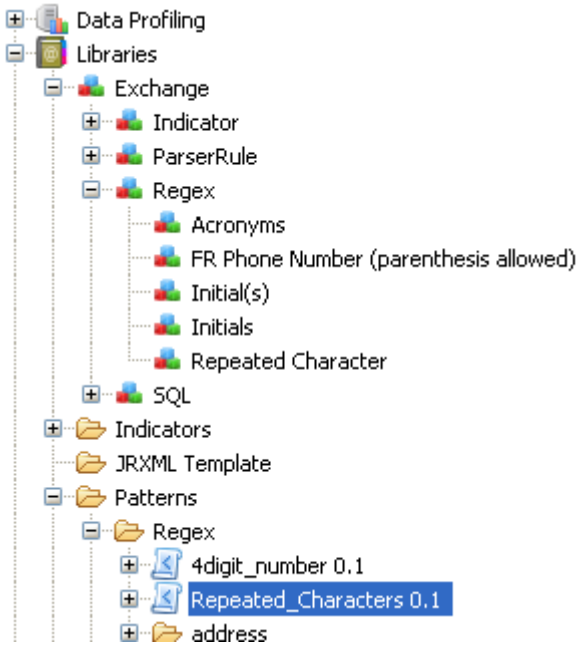
   If more than one version for the selected regular expression is available on **Talend Exchange**, a dialog box is displayed to list the versions that are compatible with your current Studio version.

4. Select a version from the list and then click **OK**.

   A message is displayed to confirm the operation.

5. Click **OK** in the confirmation message.

   The imported regular expression is listed under the **Patterns** > **Regex** folders in the **DQ Repository** tree view.



9.1.4.8. How to duplicate a regular expression or an SQL pattern

To avoid creating a regular expression or an SQL pattern from scratch, you can duplicate an existing one and work around its metadata to have a new regular expression or SQL pattern to be used in data profiling analyses.

To duplicate a regular expression or an SQL pattern, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **patterns**.

2. Browse through the regular expression/SQL pattern lists to reach the expression/pattern you want to duplicate.

3. Right-click its name and select **Duplicate...** from the contextual menu.

The duplicated regular expression/SQL pattern is displayed under the **Regex**/**SQL** folder in the **DQ Repository** tree view.

You can now double-click the duplicated pattern to modify its metadata as needed.

You can test new regular expressions before start using them against data in the specified database. For more information, see section *How to test a regular expression in the Pattern Test View*.

9.1.4.9. How to delete a regular expression or an SQL pattern

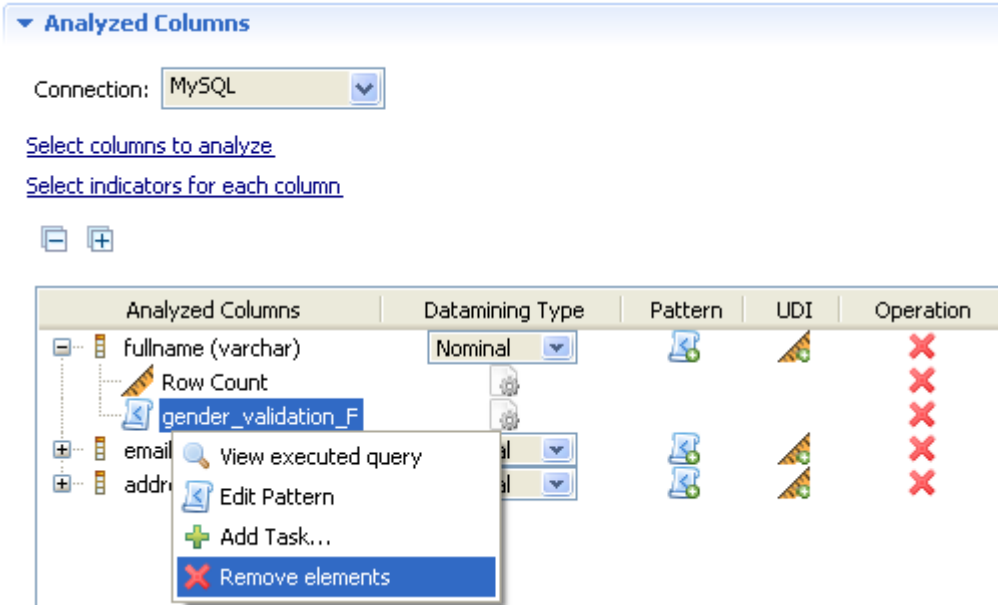
You can delete regular expressions or SQL patterns directly from the **Analyzed Columns** view or from the **DQ Repository** tree view.

How to delete a regular expression or an SQL pattern from the analyzed column

Prerequisite(s): A column analysis is open in the analysis editor in the **Profiling** perspective of the studio.

To delete a regular expression or an SQL pattern from the analyzed column, do the following:

1. Click **Analyze Columns** to display the analyzed columns view.
2. Right-click the regular expression/SQL pattern you want to delete and select **Remove Element**s from the contextual menu.

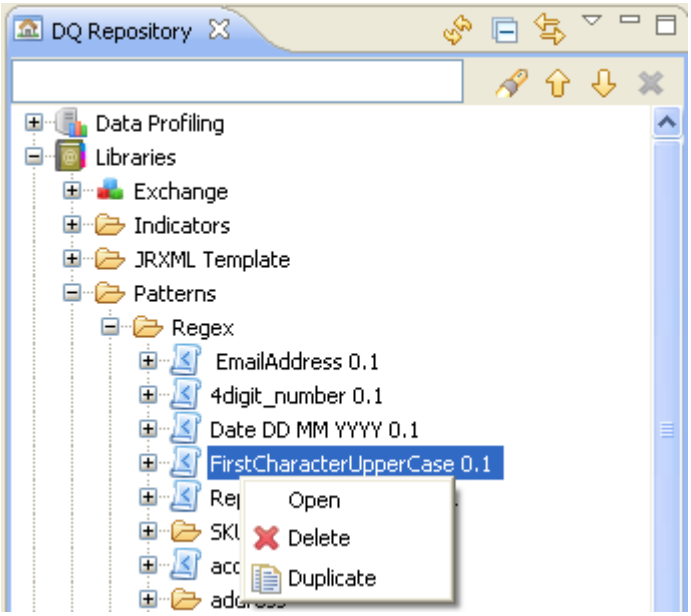


The selected regular expression/SQL pattern disappears from the **Analyzed Column** list.

How to delete and restore a regular expression or an SQL pattern from the DQ Repository

To delete a regular expression or an SQL pattern from the **DQ Repository** tree view, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Patterns**.
2. Browse to the regular expression or SQL pattern you want to remove from the list.
3. Right-click the expression or pattern and select **Delete** from the contextual menu.

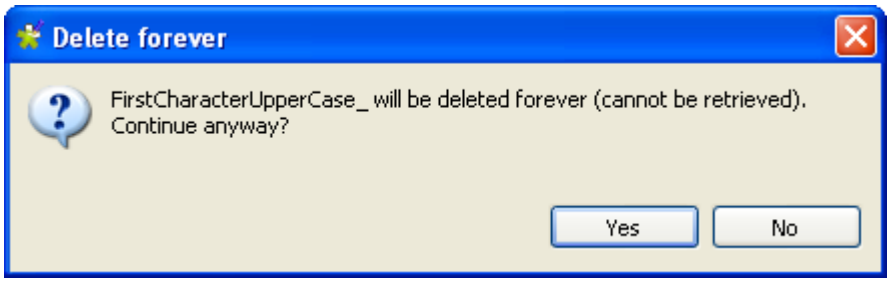


The regular expression or SQL pattern is moved to the **Recycle Bin**.

To delete it from the **Recycle Bin**, do the following:

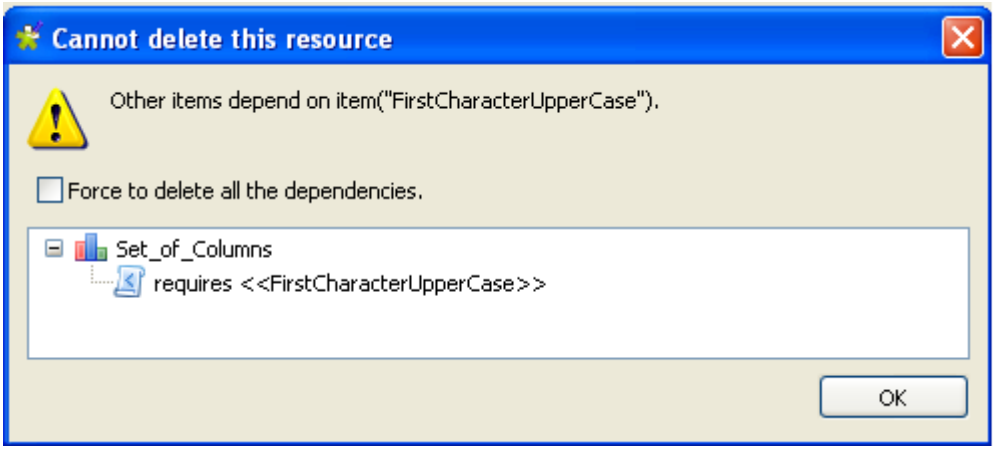
1. Right-click it in the **Recycle Bin** and choose **Delete** from the contextual menu.

   If it is not used by any analysis in the current Studio, a **[Delete forever]** dialog box is displayed.



2. Click **Yes** to confirm the operation and close the dialog box.

   If it is used by one or more analyses in the current Studio, a dialog box is displayed to list all the analyses that use the pattern.



3. Either:

   - Click **OK** to close the dialog box without deleting the pattern from the recycle bin.

- Select the **Force to delete all the dependencies** check box and then click **OK** to delete the pattern from the recycle bin and to delete all the dependent analyses from the **Data Profiling** node.

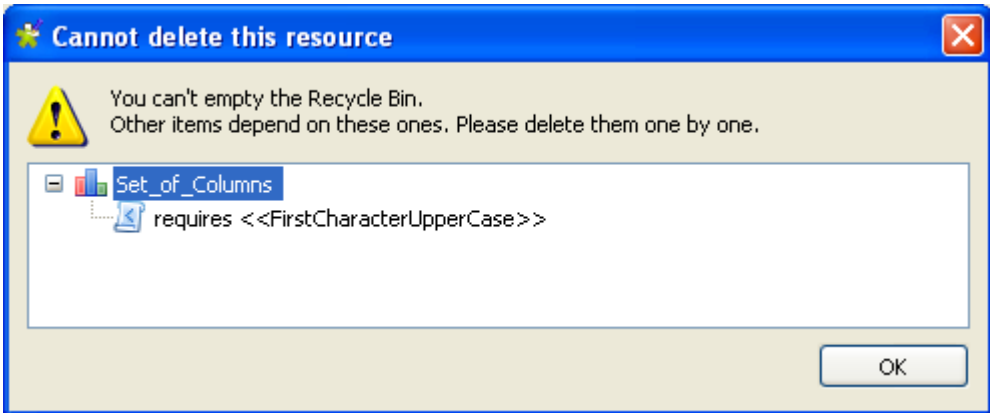
You can also delete the pattern permanently by emptying the recycle bin. To empty the **Recycle Bin**, do the following:

1. Right-click the **Recycle Bin** and select **Empty recycle bin**.

   If the pattern is not used by any analysis in the current Studio, a confirmation dialog box is displayed.

2. Click **Yes** to empty the recycle bin.

   If the pattern is used by one or more analyses in the current Studio, a dialog box is displayed to list all the analyses that use the pattern.



3. Click **OK** to close the dialog box without removing the pattern from the recycle bin.

To restore a pattern from the **Recycle Bin**, do the following:

- In the **Recycle Bin**, right-click the pattern and select **Restore**.

  The pattern is moved back to the **Libraries** node.

9.2. Indicators

Indicators can be the results achieved through the implementation of different patterns that are used to define the content, structure and quality of your data.

Indicators represent as well the results of highly complex analyses related not only to data-matching, but also to different other data-related operations.

9.2.1. Indicator types

Two types of indicators are listed under the **Indicators** folder in the **DQ Repository** tree view: system indicators and user-defined indicators.

User-defined indicators, as their name indicates, are indicators created by the user. You can use them through a simple drag-and-drop operation from the **User Defined Indicators** folder in the tree view. User-defined indicators are used only with column analyses. For more information on how to set user-defined indicators for columns, see section *How to set user-defined indicators*.

System indicators are predefined indicators grouped under different categories in the **System Indicators** folder in the **DQ Repository** tree view. Each category of the system indicators is used with a corresponding analysis type.

You can not create a system indicator or drag it directly from the **DQ Repository** tree view to an analysis. However, you can open and modify the parameters of a system indicator to adapt it to a specific database for example. For further information, see section *How to edit a system indicator*

Several management options including editing, duplicating, importing and exporting are possible for both types of indicators. For more information, see section *Managing user-defined indicators* and section *Managing system indicators*.

The below sections describe the system indicators used on column analyses. These system indicators can range from simple or advanced statistics to text strings analysis, including summary data and statistical distributions of records.

You can see under the **System Indicators** folder in the **DQ Repository** tree view system indicators other than the indicators in the below sections. Those different system indicators are used on the other analysis types, for example redundancy, correlation and overview analyses.

9.2.1.1. Simple statistics

They provide simple statistics on the number of records falling in certain categories including the number of rows, the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields.

- Blank Count: counts the number of blank rows. A “blank” is a non null textual data that contains only white space. Note that Oracle does not distinguish between the empty string and the null value.
- Default Value Count: counts the number of default values.
- Distinct Count: counts the number of distinct values of your column.
- Duplicate Count: counts the number of values appearing more than once. You have the relation: Duplicate count + Unique count = Distinct count. For example, a,a,a,a,b,b,c,d,e => 9 values, 5 distinct values, 3 unique values, 2 duplicate values.
- Null Count: counts the number of null rows.
- Row Count: counts the number of rows.
- Unique Count: counts the number of distinct values with only one occurrence. It is necessarily less or equal to Distinct counts.

9.2.1.2. Text statistics

They analyze the characteristics of textual fields in the columns, including minimum, maximum and average length.

- Minimal Length: computes the minimal length of a text field.
- Maximal Length: computes the maximal length of a text field.
- Average Length: computes the average length of a field.

Other text indicators are available to count each of the above indicators with null values, with blank values or with null and blank values.

Null values will be counted as data of 0 length, i.e. the minimal length of null values is 0. This means that the **Minimal Length With Null** and the **Maximal Length With Null** will compute the minimal/maximal length of a text field including null values.

Blank values will be counted as data of 0 length, i.e. the minimal length of blank values is 0. This means that the **Minimal Length With Blank** and the **Maximal Length With Blank** will compute the minimal/maximal length of a text field including blank values.

The same will be applied for all average indicators.

The below table gives an example of computing the length of few textual fields in a column using all different types of text statistic indicators.

Data	Current length	With blank values	With null values	With blank and null values
Brayan	6	6	6	6
Ava	3	3	3	3
“ “	1	0	1	0
““	0	0	0	0
Null	—	—	0	0
Minimal, Maximal and Average lengths				
Minimal length	0	0	0	0
Maximal length	6	6	6	6
Average length	9/4 = 2.25	8/4 = 2	9/5 = 1.8	8/5 = 1.6

9.2.1.3. Summary statistics

They perform statistical analyses on numeric data, including the computation of location measures such as the median and the average, the computation of statistical dispersions such as the inter quartile range and the range.

- Mean: computes the average of the records.
- Median: computes the value separating the higher half of a sample, a population, or a probability distribution from the lower half.
- Inter quartile range: computes the difference between the third and first quartiles.
- Range: computes the difference between the highest and lowest records.

9.2.1.4. Advanced statistics

They determine the most probable and the most frequent values and builds frequency tables. The main advanced statistics include the following values:

- Mode: computes the most probable value. For numerical data or continuous data, you can set bins in the parameters of this indicator. It is different from the “average” and the “median”. It is good for addressing categorical attributes.
- Frequency table: computes the number of most frequent values for each distinct record. Other frequency table indicators are available to aggregate data with respect to “date”, “week”, “month”, “quarter”, “year” and "bin".

  Frequency table statistics are applied only on columns that have "date" data.

- Low frequency table: computes the number of less frequent records for each distinct record. Other low frequency table indicators are available for each of the following values: “date”, “week”, “month”, “quarter”, “year” and “bin” where “bin” is the aggregation of numerical data by intervals.

9.2.1.5. Pattern frequency statistics

Indicators in this group determine the most and less frequent patterns.

- Pattern frequency table: computes the number of most frequent records for each distinct pattern.
- Pattern low frequency table: computes the number of less frequent records for each distinct pattern.
- Date pattern frequency table: retrieves the date patterns from date or text columns. It works only with the Java engine.

9.2.1.6. Soundex frequency statistics

Indicators in this group use the Soundex algorithm built in the DBMS.

They index records by sounds. This way, records with the same pronunciation (only English pronunciation) are encoded to the same representation so that they can be matched despite minor differences in spelling.

- Soundex frequency table: computes the number of most frequent distinct records relative to the total number of records having the same pronunciation.
- Soundex low frequency table: computes the number of less frequent distinct records relative to the total number of records having the same pronunciation.

9.2.1.7. Phone number statistics

Indicators in this group count phone numbers. They return the count for each phone number format. They validate the phone formats using the *org.talend.libraries.google.libphonumber* library.

- Valid phone number count: computes the valid phone numbers.
- Possible phone number count: computes the supposed valid phone numbers.
- Valid region code number count: computes phone numbers with valid region code.
- Invalid region code count. computes phone numbers with invalid region code.
- Well formed national phone number count: computes well formatted national phone numbers.
- Well formed international phone number count: computes the international phone numbers that respect the international phone format (phone numbers that start with the country code) .
- Well formed E164 phone number count: computes the international phone numbers that respect the international phone format (maximum of fifteen digits written with a + prefix.
- Format Frequency Pie: shows the results of the phone number count in a pie chart divided into sectors.

9.2.1.8. Benford's law frequency indicator

The **Benford Law Frequency** indicator (first-digit law) is based on examining the actual frequency of the digits *1* through *9* in numerical data. It is usually used as an indicator of accounting and expenses fraud in lists or tables.

Benford's law states that in lists and tables the digit *1* tends to occur as a leading digit about 30% of the time. Larger digits occur as the leading digits with lower frequency, for example the digit *2* about 17%, the digit *3* about 12% and so on. Valid, unaltered data will follow this expected frequency. A simple comparison of first-digit frequency distribution from the data you analyze with the expected distribution according to Benford's law ought to show up any anomalous results.

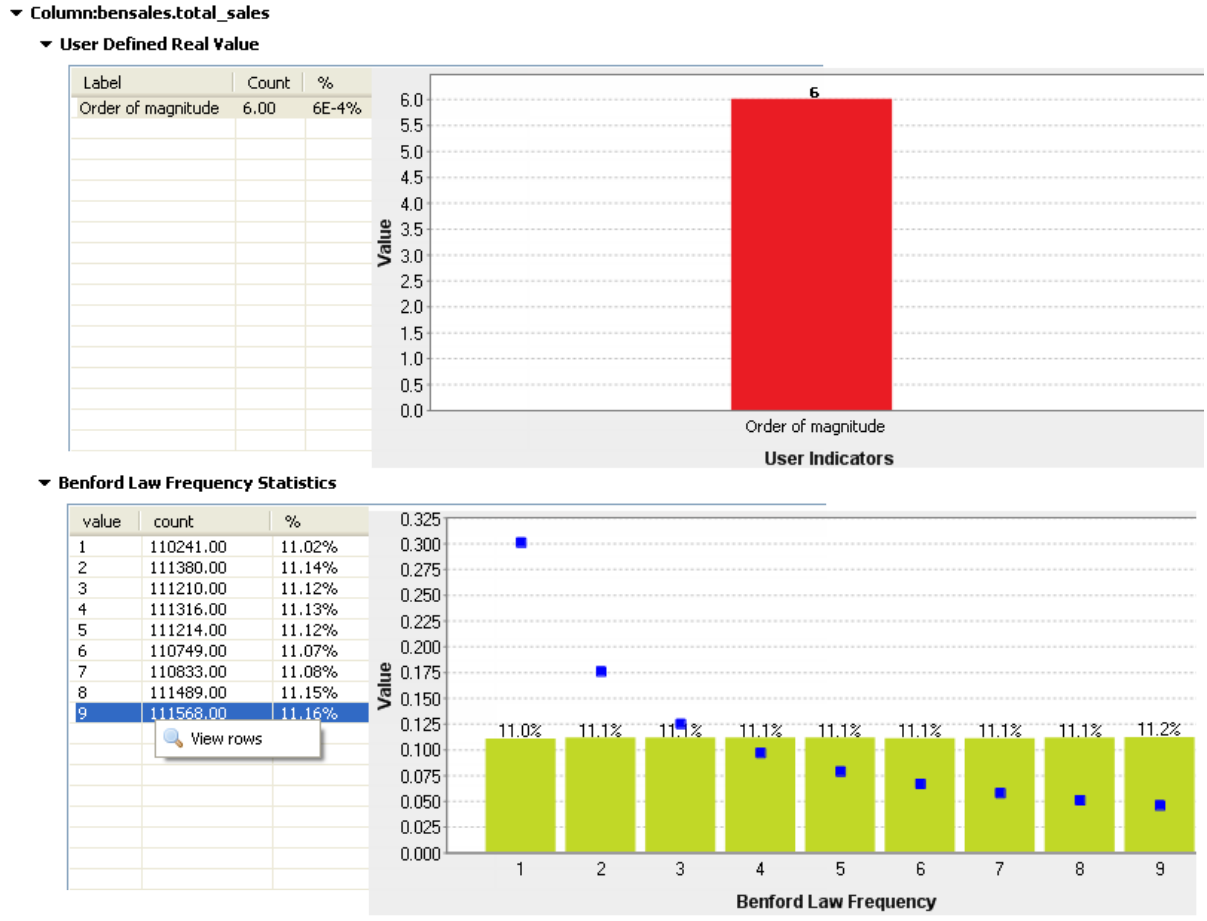
For example, let's assume an employee has committed fraud by creating and sending payments to a fictitious vendor. Since the amounts of these fictitious payments are made up rather than occurring naturally, the leading digit distribution of all fictitious and valid transactions (mixed together) will no longer follow Benford's law. Furthermore, assume many of these fraudulent payments have *2* as the leading digit, such as 29, 232 or 2,187. By using the Benford Law indicator to analyze such data, you should see the amounts that have the leading digit *2* occur more frequently than the usual occurrence pattern of 17%.

When using the **Benford Law Frequency** indicator, it is advised to:

- make sure that the numerical data you analyze do not start with *0* as Benford's law expects the leading digit to range only from *1* to *9*. This can be verified by using the **number > Integer values** pattern on the column you analyze.
- check the order of magnitude of the data either by selecting the min and max value indicators or by using the **Order of Magnitude** indicator you can import from **Talend Exchange**. This is because Benford's law tends to be most accurate when values are distributed across multiple orders of magnitude. For further information about importing indicators from **Talend Exchange**, see section *How to import user-defined indicators from Talend Exchange*.

In the result chart of the **Benford Law Frequency** indicator, digits *1* through *9* are represented by bars and the height of the bar is the percentage of the first-digit frequency distribution of the analyzed data. The dots represent the expected first-digit frequency distribution according to Benford's law.

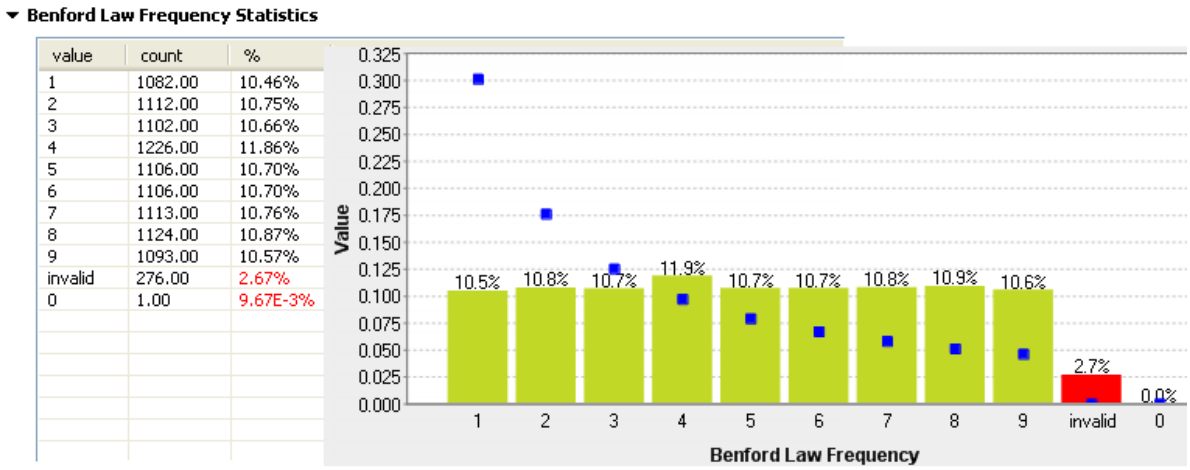
Below is an example of the results of an analysis after using the **Benford Law Frequency** indicator and the **Order of Magnitude** user-defined indicator on a *total_sales* column.



The first chart shows that the analyzed data varies over *6* orders of magnitude, that is there are *6* digits between the minimal value and maximal value of the numerical column.

The second chart shows that the actual distribution of the data (height of bars) does not follow the Benford's law (dot values). The differences are very big between the frequency distribution of the sales figures and the expected distribution according to Benford's law. For example, the usual occurrence pattern for sales figures that start with *1* is 30% and those figures in the analyzed data represent only 11%. Some fraud could be suspected here, sales figures may have been modified by someone or some data may be missing.

Below is another example of the result chart of a column analysis after using the **Benford Law Frequency** indicator.



The red bar labeled as invalid means that this percentage of the analyzed data does not start with a digit. And the *0* bar represents the percentage of data that starts with 0. Both cases are not expected when analyzing columns using the **Benford Law Frequency** indicator and this is why they are represented in red.

For further information about analyzing columns, see section *Analyzing columns in a database*.

9.2.2. Managing system indicators

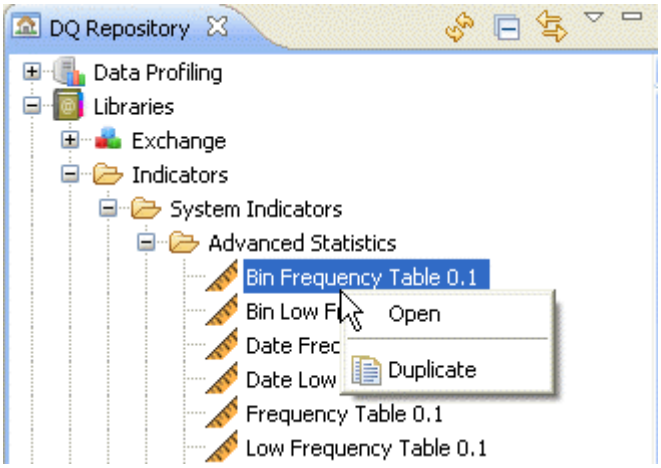
System indicators are predefined but editable indicators that are automatically used on the relevant analyses. For more information on the system indicators available in the studio, see section *Indicator types*.

9.2.2.1. How to edit a system indicator

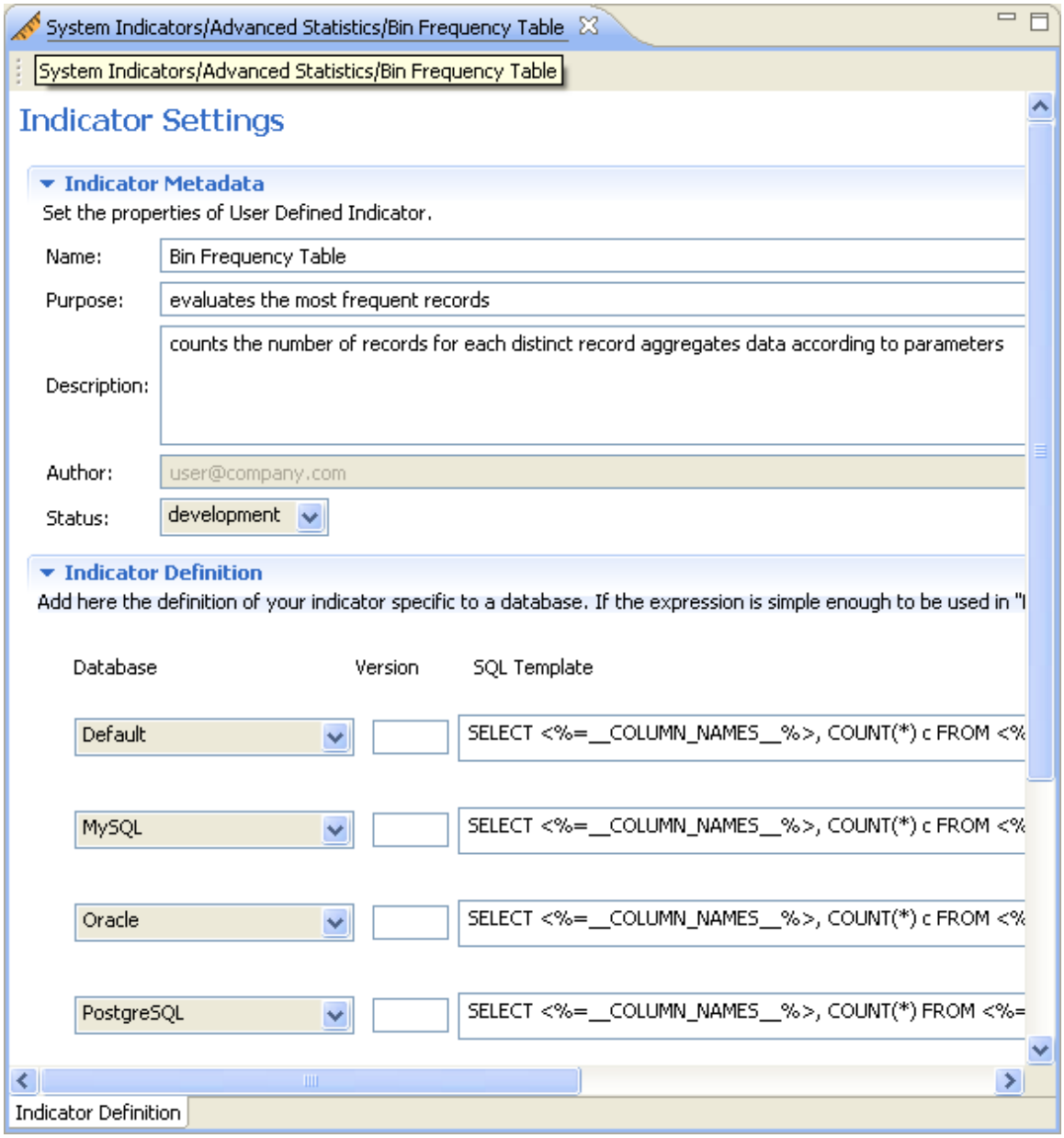
Although system indicators are predefined indicators, you can open their editors to check their settings or to edit their definition in order to adapt them to a specific database version or to a specific need, for example.

To edit a system indicator, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Indicators**, and then browse through the indicator lists to reach the indicator you want to modify.
2. Right-click the indicator name and select **Open** from the contextual menu.



The indicator editor opens displaying the selected indicator parameters.



3. Modify the indicator metadata, if required, and then click **Indicator Definition**.

   In this view, you can edit the indicator definition, change the selected database and add other indicators specific to available databases using the **[+]** button at the bottom of the editor.

4. Click the save icon on top of the editor to save your changes.

If the indicator is simple enough to be used in all databases, select `Default` in the database list.

When you edit an indicator, you modify the indicator listed in the DQ Repository tree view. Make sure that your modifications are suitable for all analyses that may be using the modified indicator.

9.2.2.2. How to set system indicators and indicator options to column analyses

You can define system indicators and indicator parameters for columns of database tables that need to be analyzed or monitored. For more information, see section *How to set indicators for the column(s) to be analyzed* and section *How to set options for system indicators*.

9.2.2.3. How to export or import system indicators

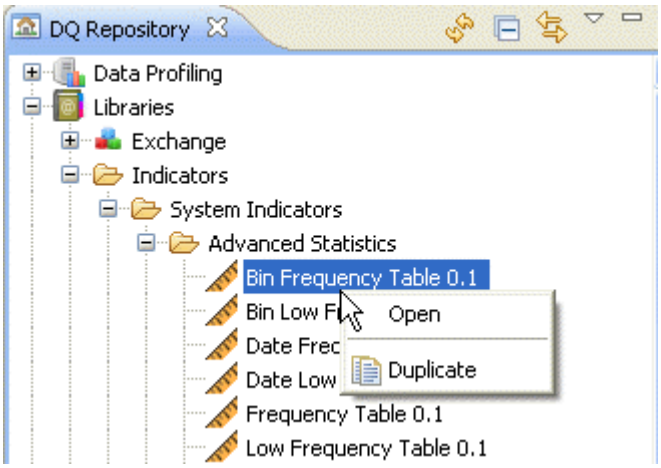
You can export system indicators to folders or archive files and import them again in the studio on the condition that the export and import operations are done in compatible versions of the Studio. For further information, see section *Exporting data profiling items* and section *Importing data profiling items or projects*.

9.2.2.4. How to duplicate a system indicator

To avoid creating a system indicator from scratch, you can duplicate an existing one in the indicator list. once the copy is created, you can work around its metadata to have a new indicator and use it in data profiling analyses.

To duplicate a system indicator, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators**.
2. Browse through the indicator lists to reach the indicator you want to duplicate, right-click its name and select **Duplicate...** from the contextual menu.



The duplicated indicator is displayed under the **System** folder in the **DQ Repository** tree view.

You can now open the duplicated indicator to modify its metadata and definition as needed. For more information on editing system indicators, see section *How to edit a system indicator*.

9.2.3. Managing user-defined indicators

User-defined indicators, as their name indicates, are indicators created by the user himself/herself. You can use these indicators to analyzed columns through a simple drag-and-drop operation from the **DQ Repository** tree view to the analyzed columns.

The management options available for user-defined indicators include: create, export and import, edit and duplicate. For detailed information, see the following sections.

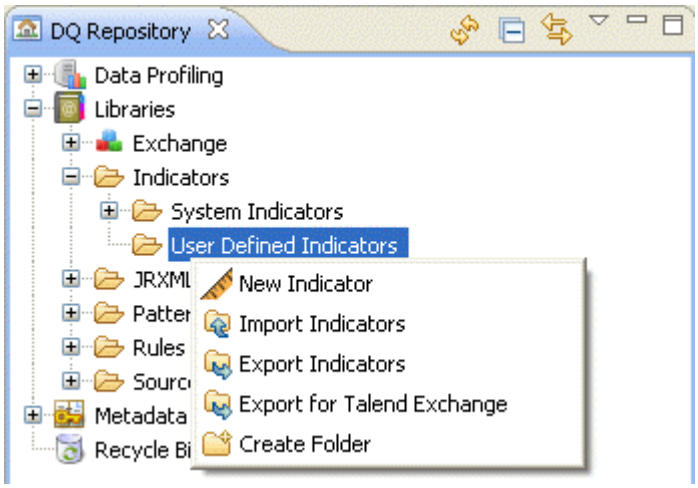
9.2.3.1. How to create SQL user-defined indicators

You can create your own personalized indicators from the studio.

Management processes for user-defined indicators are the same as those for system indicators.

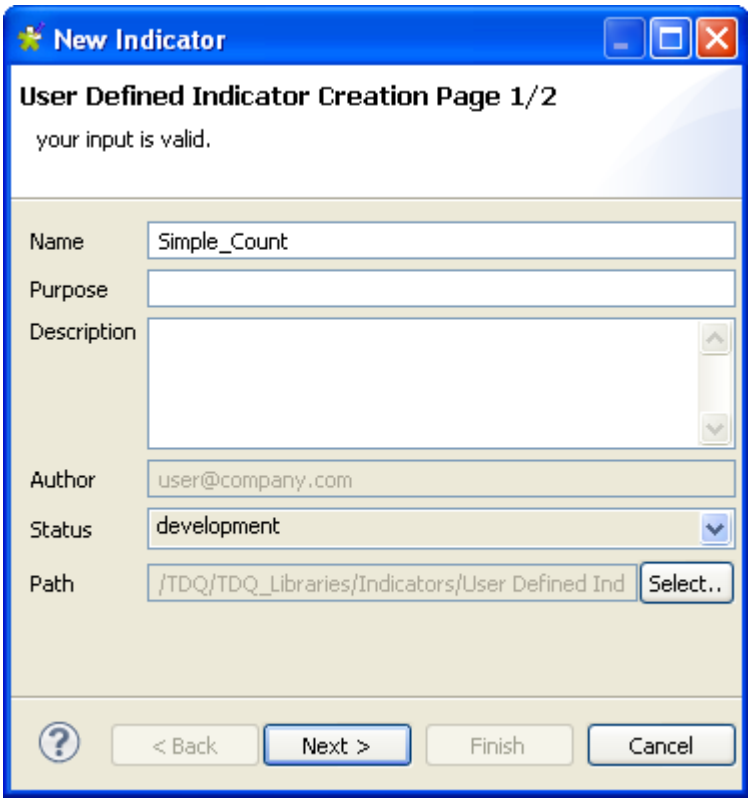
Defining the indicator

1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators**.
2. Right-click **User Defined Indicators**.



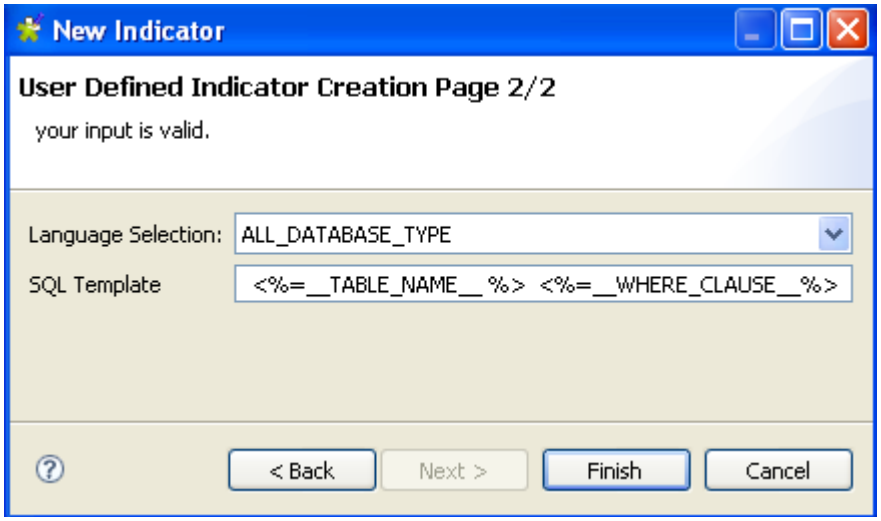
3. Select **New Indicator** from the contextual menu.

   The **[New Indicator]** wizard is displayed.



4. In the **Name** field, enter a name for the indicator you want to create.

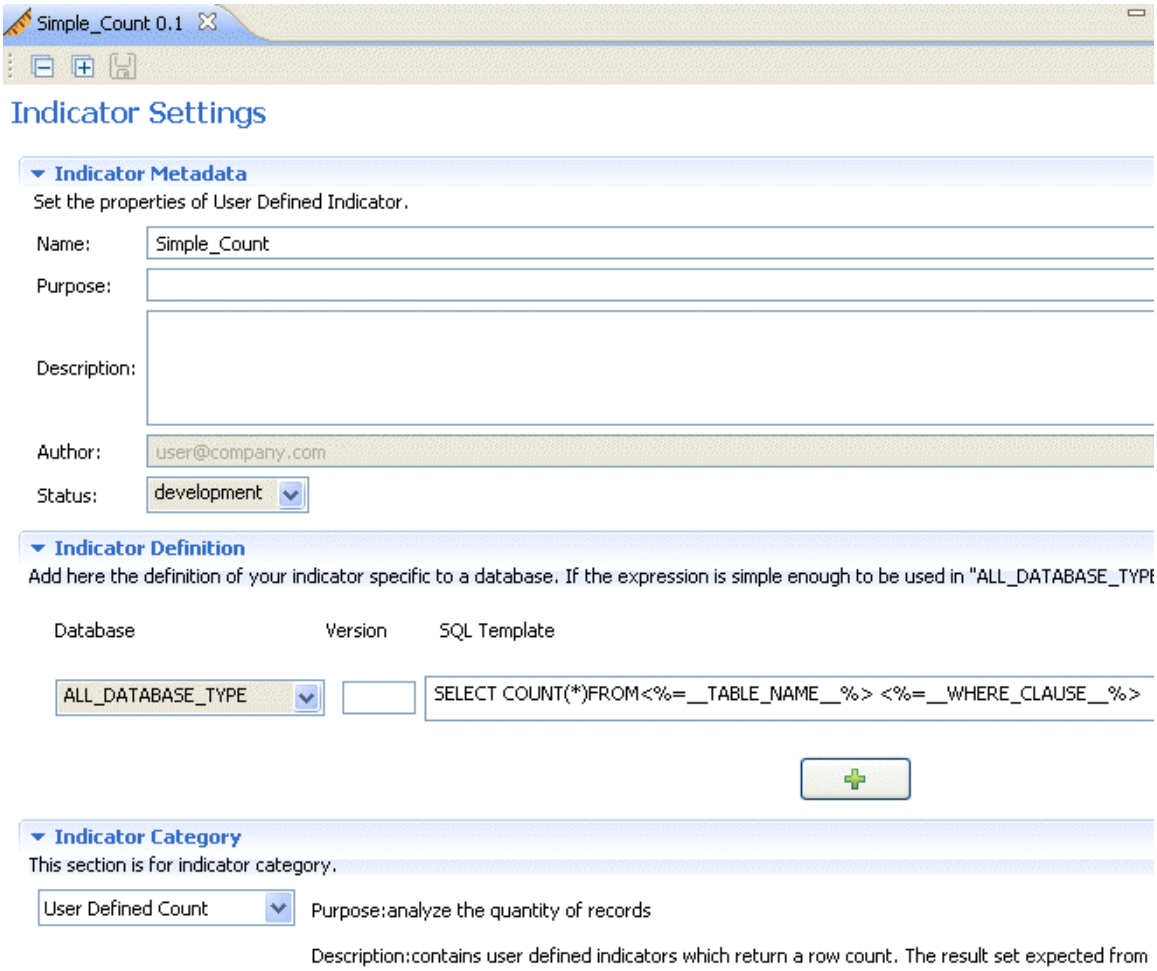
If required, set other metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



5. From the **Language Selection** list, select the database that will support the created indicator.

6. In the **SQL Template** field, enter the SQL template statement corresponding to the indicator you want to create and then click **Finish** to close the wizard and proceed to the next step.

   The indicator editor opens displaying the metadata of the user-defined indicator.



Setting the indicator definition and category

1. In the editor, click **Indicator Definition** to display the corresponding view.
2. If required, change the selected database or click the **Edit...** button to the right of the view to edit the indicator definition.
3. If required, click the **[+]** button and add other indicators specific to available databases.
4. Enter the database version in the **Version** field.
5. Click **Indicator Category** to display the corresponding view. In this view, you can select from the list a category for the created indicator. The selected category will determine the type of chart that will represent the results of the executed analysis that uses the created indicator.
6. From the **Indicator Category** list, select a category for the created indicator.

   The table below explains available categories.

Indicator category	Description
User Defined Match (by-default category)	Uses user-defined indicators to evaluate the number of the data records that match a regular expression or an SQL pattern. The analysis results show the record matching count and the record total count.
User Defined Frequency	Uses user-defined indicators for each distinct data record to evaluate the record frequency that match a regular expression or an SQL pattern. The analysis results show the distinct count giving a label and a label-related count.
User Defined Real Value	Uses user-defined indicators which return real value to evaluate any real function of the data.
User Defined Count	Uses user-defined indicators that return a row count.

7. Click the save icon on top of the editor.

   The created indicator is listed under the **User Defined Indicators** folder in the **DQ Repository** tree view.

9.2.3.2. How to define Java user-defined indicators

You can create your own personalized Java indicators from the studio. Management processes for Java user-defined indicators are the same as those for system indicators.

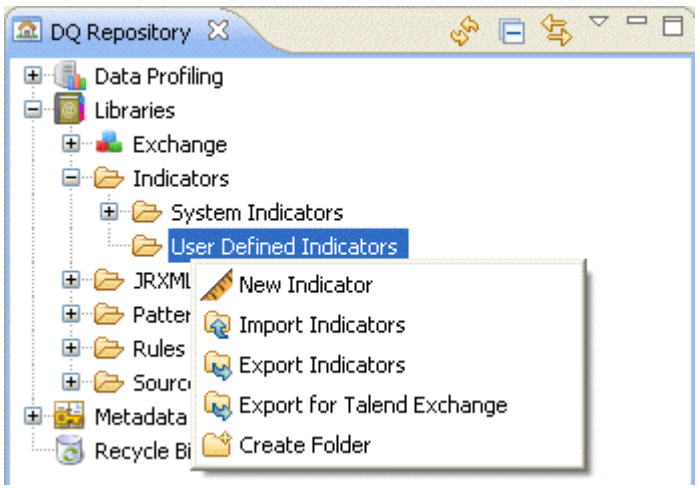
You can also import a ready-to-use Java user-defined indicator from the **Exchange** folder in the **DQ Repository** tree view. This Java user-defined indicator connects to the mail server and checks if the email exists. For further information on importing indicators from **Talend Exchange**, see section *How to import user-defined indicators from Talend Exchange*.

The two sections below detail the procedures to create Java user-defined indicators.

How to create Java user-defined indicators

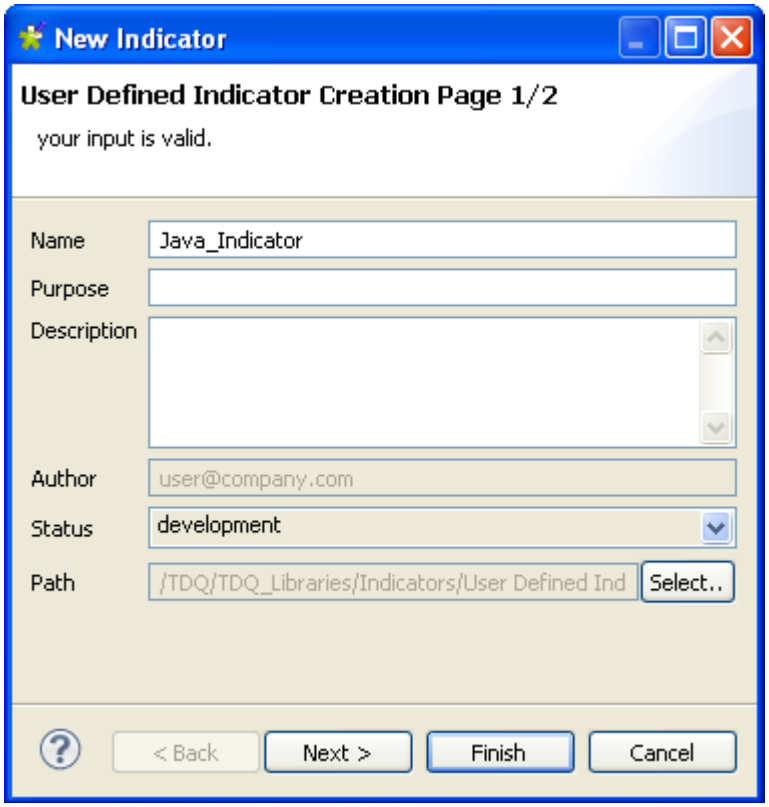
Defining the indicator

1. In the **DQ Repository** tree view, expand **Libraries > Indicators**.
2. Right-click **User Defined Indicators**.



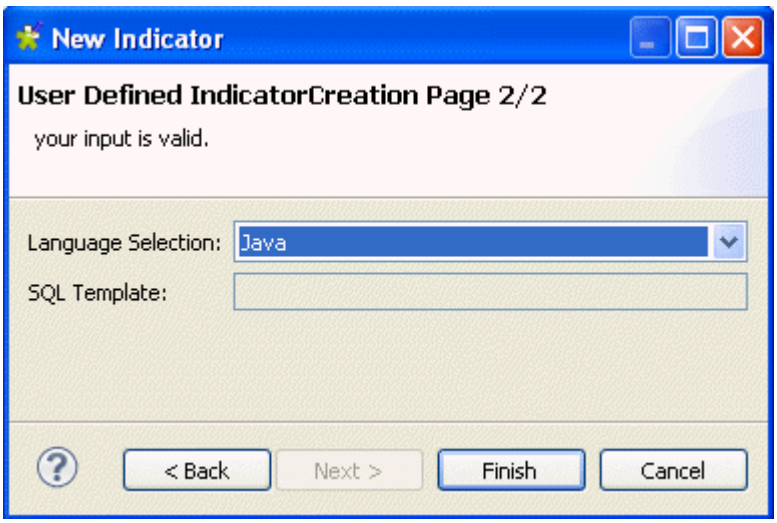
3. Select **New Indicator** from the contextual menu.

   The **[New Indicator]** wizard is displayed.



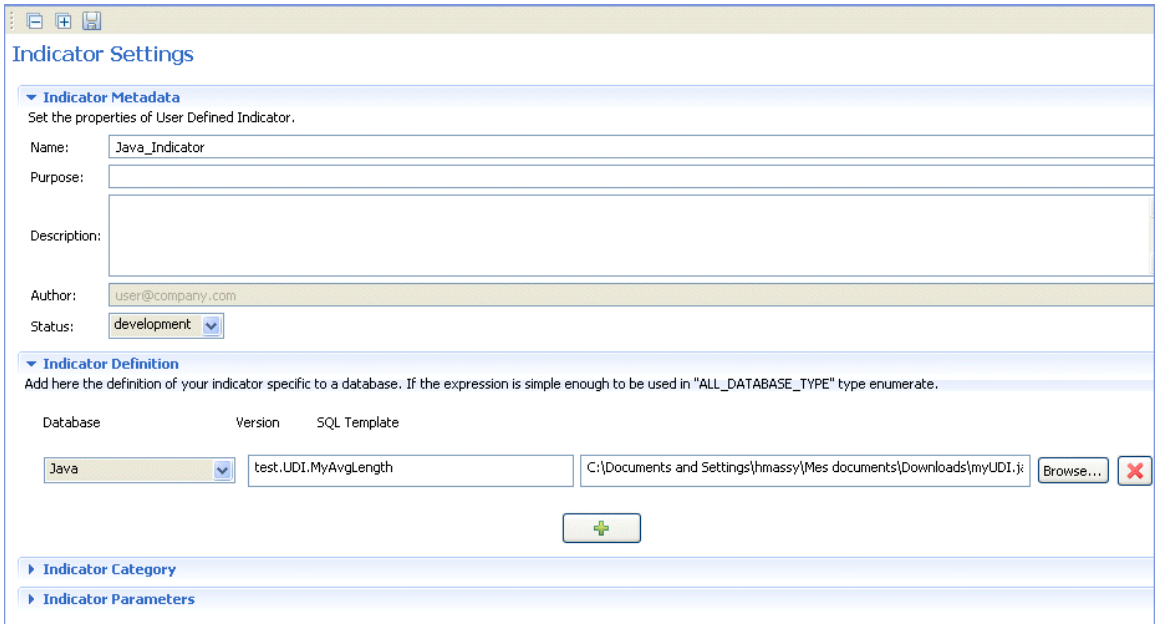
4. In the **Name** field, enter a name for the Java indicator you want to create.

5. If required, set other metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.



6. From the **Language Selection** list, select **Java** and then click **Finish** to open the indicator settings.

   The indicator editor opens displaying the metadata of the Java indicator.



Setting the indicator definition and category

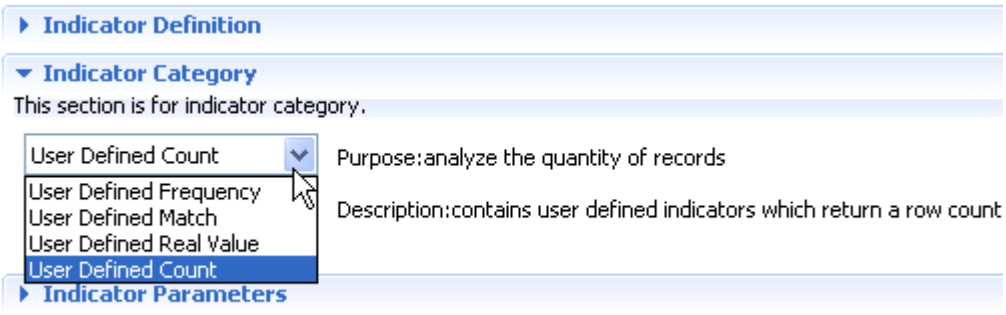
1. In the editor, click **Indicator Definition** to display the corresponding view. **Java** is selected by default.

2. Click the browse button to the right of the view and browse to the Java archive holding the Java classes. For more information on creating a Java archive, see section *How to create a Java archive for the user-defined indicator*.

3. Enter the Java class in the **Version** field.

   Make sure that the class name includes the package path, if this string parameter was not correctly specified, an error message will display when you try to save the Java user-defined indicator.

4. Click **Indicator Category** to display the corresponding view.

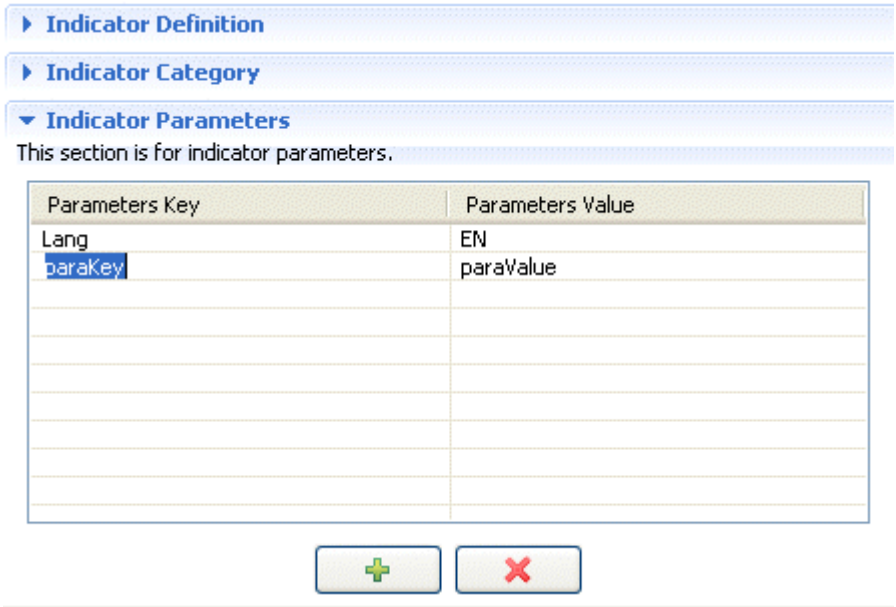


5. From the **Indicator Category** list, select a category for the created Java indicator.

   The selected category will determine the type of chart that will represent the results of the executed analysis that uses the created Java indicator.

Indicator category	Description
User Defined Count	Uses user-defined indicators that return a row count.
User Defined Real Value	Uses user-defined indicators which return real value to evaluate any real function of the data.
User Defined Match (by-default category)	Uses user-defined indicators to evaluate the number of the data records that match a regular expression or an SQL pattern. The analysis results show the record matching count and the record total count.
User Defined Frequency	Uses user-defined indicators for each distinct data record to evaluate the record frequency. The analysis results show the distinct count giving a label and a label-related count.

6. Click **Indicator Parameter** to display the corresponding view.



   In this table, you can set the default parameters for this new Java indicator. These default parameters are stored in a *Map* object.

7. Click the **[+]** button at the bottom of the table to add as many lines as needed and define the parameter key and value.

8. Click in the line and define the parameter key and the parameter value.

You can edit these default parameters or even add new parameters any time you use the indicator in a column analysis. To do this, click the indicator option icon in the analysis editor to open a dialog box where you can edit the default parameters according to your needs or add new parameters.

9. Click the save icon on top of the editor.

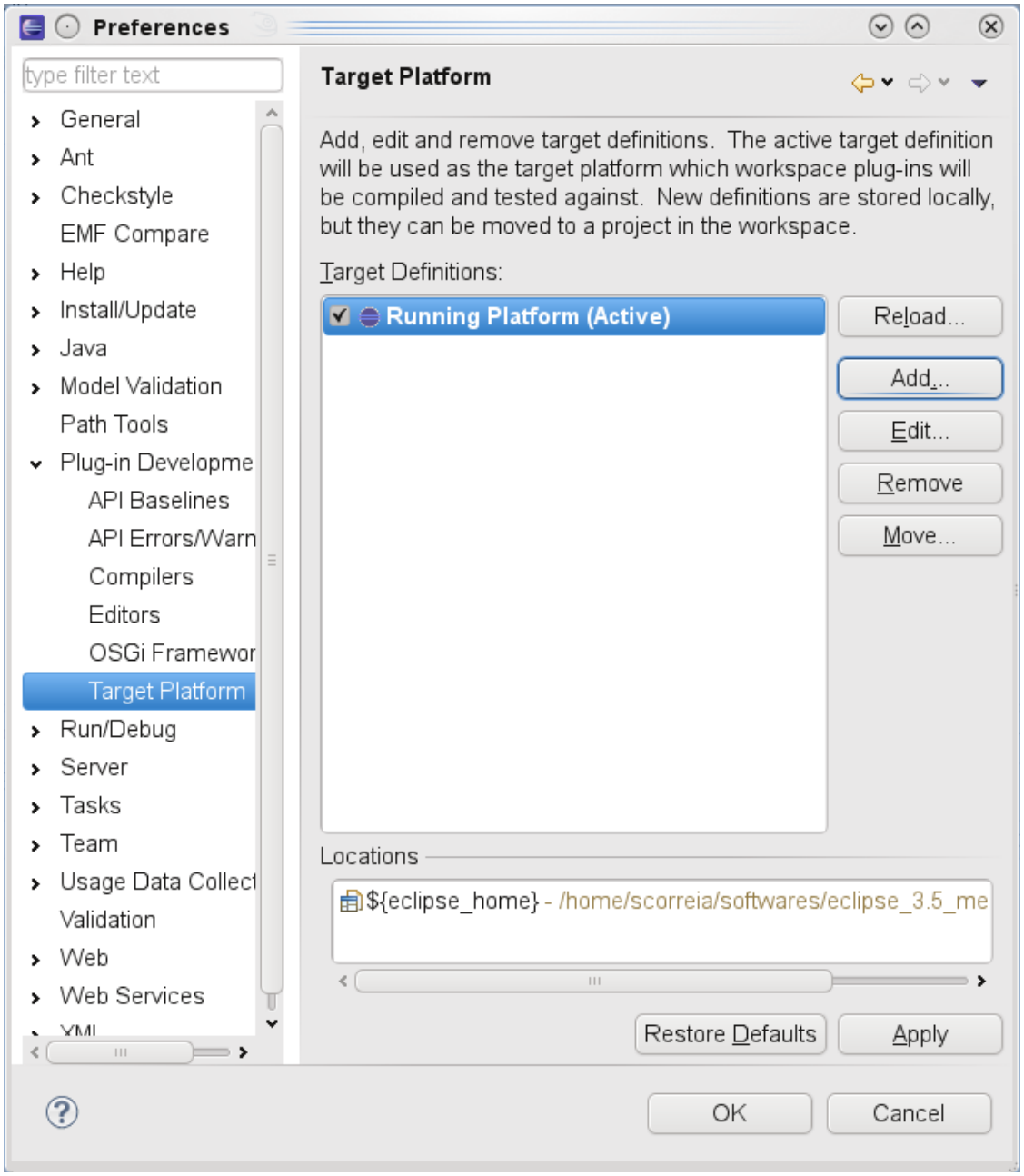
   The created indicator is listed under the **User Defined Indicators** folder in the **DQ Repository** tree view.

How to create a Java archive for the user-defined indicator

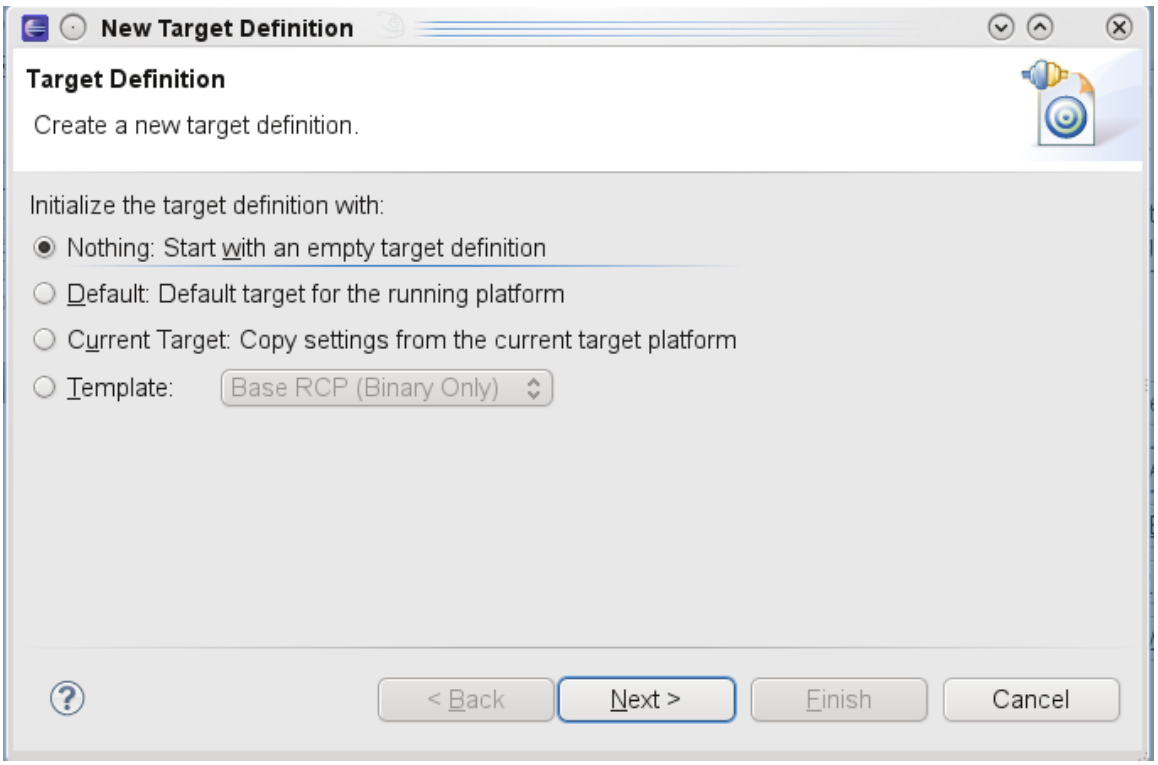
Before creating a Java archive for the user defined indicator, you must define, in Eclipse, the target platform against which the workspace plug-ins will be compiled and tested.

To define the target platform, do the following:

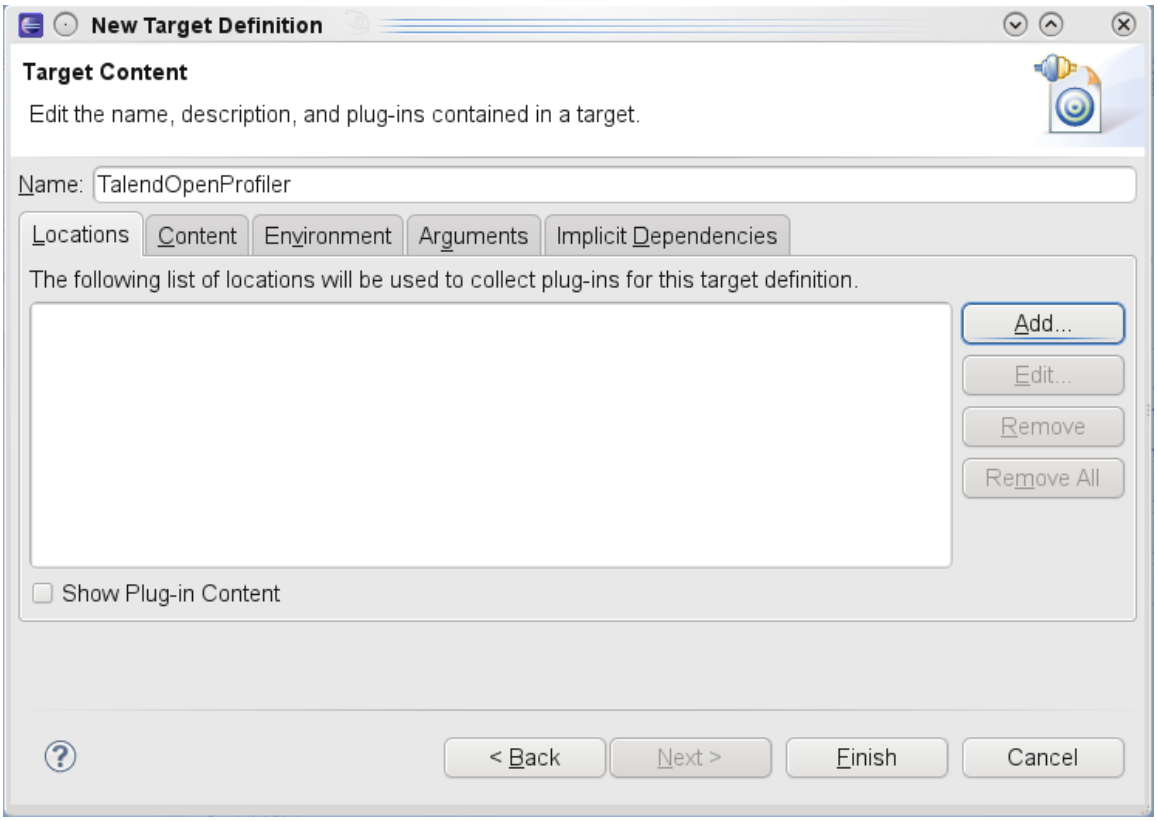
1. In Eclipse, select **Preferences** to display the **[Preferences]** window.



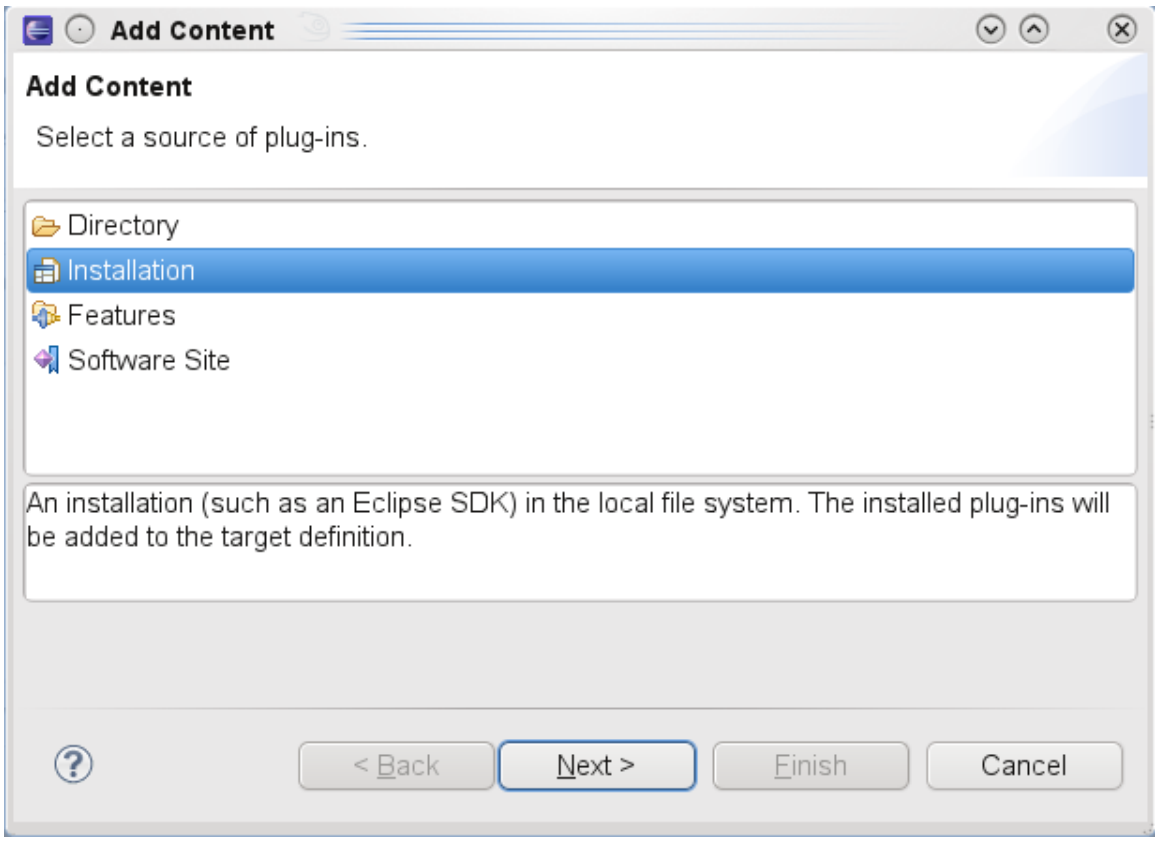
2. Expand **Plug-in Development** and select **Target Platform** then click **Add...** to open a view where you can create the target definition.



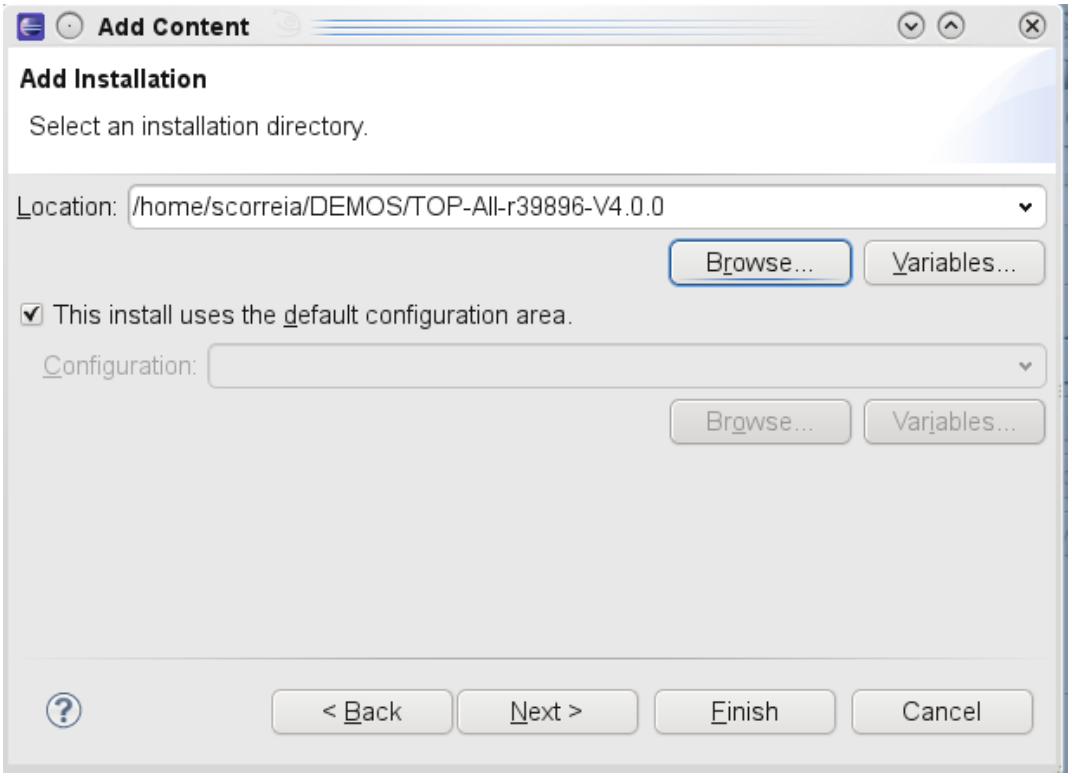
3. Select the **Nothing: Start with an empty target definition** option and then click **Next** to proceed to the next step.



4. In the **Name** field, enter a name for the new target definition and then click the **Add...** button to proceed to the next step.

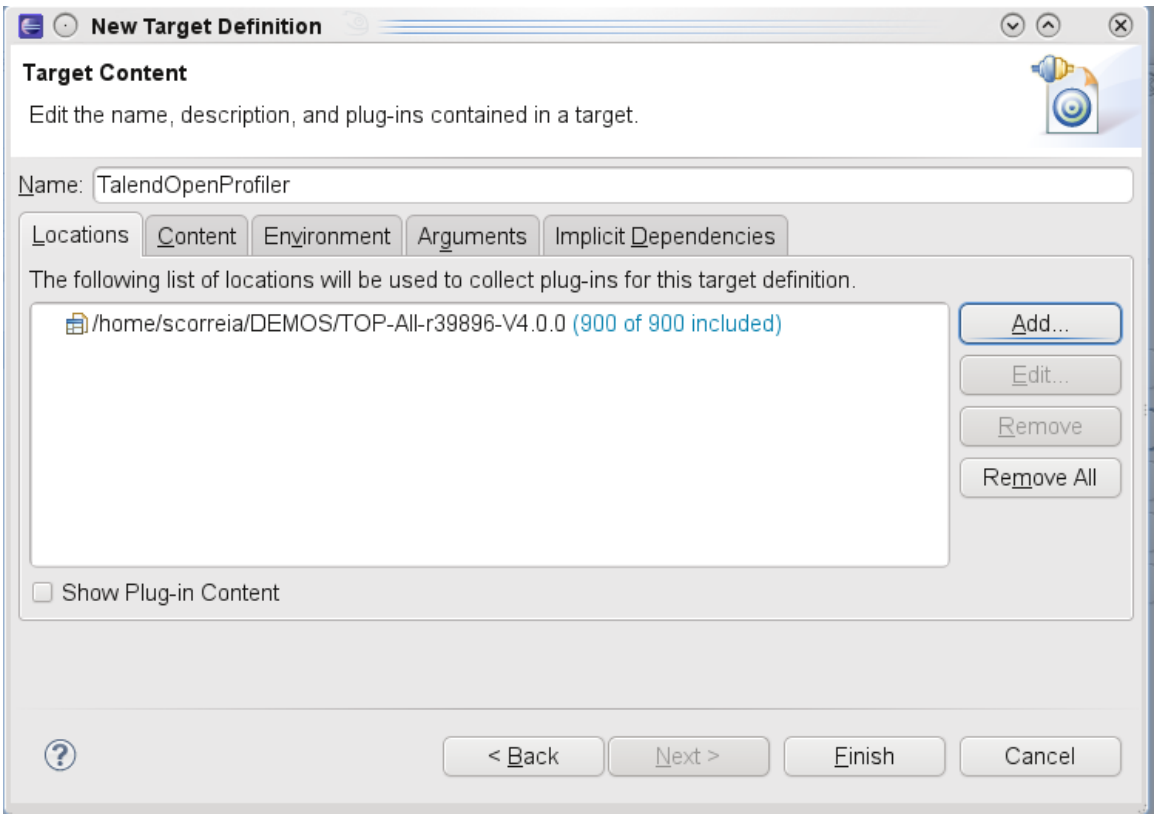


5. Select **Installation** from the **Add Content** list and then click **Next** to proceed to the next step.



6. Use the **Browse...** button to set the path of the installation directory and then click **Next** to proceed to the next step.

   The new target definition is displayed in the location list.



7. Click **Finish** to close the dialog box.

To create a Java archive for the user defined indicator, do the following:

1. In Eclipse, check out the project from svn at http://talendforge.org/svn/top/branches/branch-4_0/test.myudi.

   In this Java project, you can find four Java classes that correspond to the four indicator categories listed in the **Indicator Category** view in the indicator editor.

   - ..
   - MyAvgLength.java
   - MyFrequencyUDI.java
   - MyNotNullMatchingUDI.java
   - MyNotNullUDI.java

   Each one of these Java classes extends the `UserDefIndicatorImpl` indicator. The figure below illustrates an example using the `MyAvgLength` Java class.

```
package test.udi;

import org.talend.dataquality.indicators.sql.impl.UserDefIndicatorImpl;

/**
 * @author mzhao
 *
 * A very simple example of a java implementation of a user defined indicator. This in
 * real value. It implements the minimum number of required methods.
 */
public class MyAvgLength extends UserDefIndicatorImpl {

    private double length = 0;

    @Override
    public boolean reset() {
        super.reset();
        length = 0;
        return true;
    }

    @Override
    public boolean handle(Object data) {
        super.handle(data);
        // an indicator which computes the average text length on data which are more
        // text values with less than 2 characters are not taken into account).
        int dataLength = (data != null) ? data.toString().length() : 0;
        if (dataLength > 2) {
            length += dataLength;
        }
        return true;
    }

    /*
     * (non-Javadoc)
     *
     * @see org.talend.dataquality.indicators.impl.IndicatorImpl#finalizeComputation()
     */
    @Override
    public boolean finalizeComputation() {
        value = String.valueOf(this.length / (this.getCount() - this.getNullCount()));
        return super.finalizeComputation();
    }
```

2. Modify the code of the methods that follow each `@Override` according to your needs.
3. If required, use the following methods in your code to retrieve the indicator parameters:
4. use `Indicator.getParameter()` which returns an `IndicatorParameters` object.
5. call `IndicatorParameters.getIndicatorValidDomain()` which returns a `Domain` object.
6. call `Domain.getJavaUDIIndicatorParameter()` which returns a list of `JavaUDIIndicatorParameter` that stores each key/value pair that defines the parameter.
7. Save your modifications.
8. Using Eclipse, export this new Java archive.

The Java archive is now ready to be attached to any Java indicator you want to create in from the **Profiling** perspective of the studio.

9.2.3.3. How to export user-defined indicators

You can export user-defined indicators to a local csv file or to **Talend Exchange** to be shared with other users.

You can also export user-defined indicators to folders or archive files. For further information, see section *Exporting data profiling items*.

You can only export user-defined indicators based on SQL templates. It is not possible to export Java user-defined indicators.

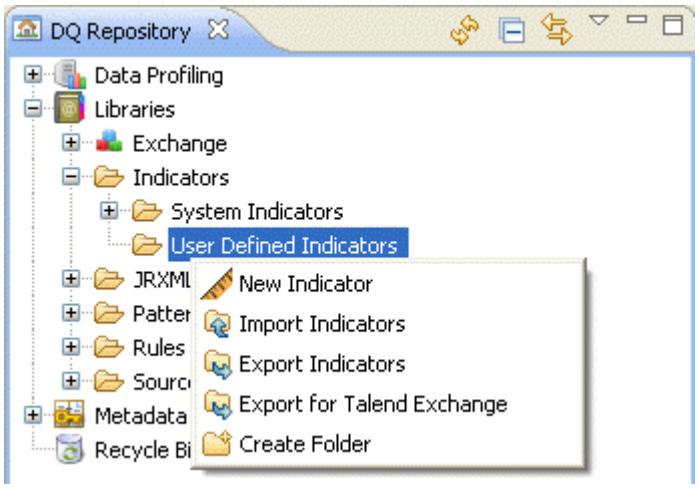
How to export user-defined indicators to a csv file

You can export user-defined indicators and store them locally in a csv file.

Prerequisite(s): At least one user-defined indicator is created in the **Profiling** perspective of the studio.

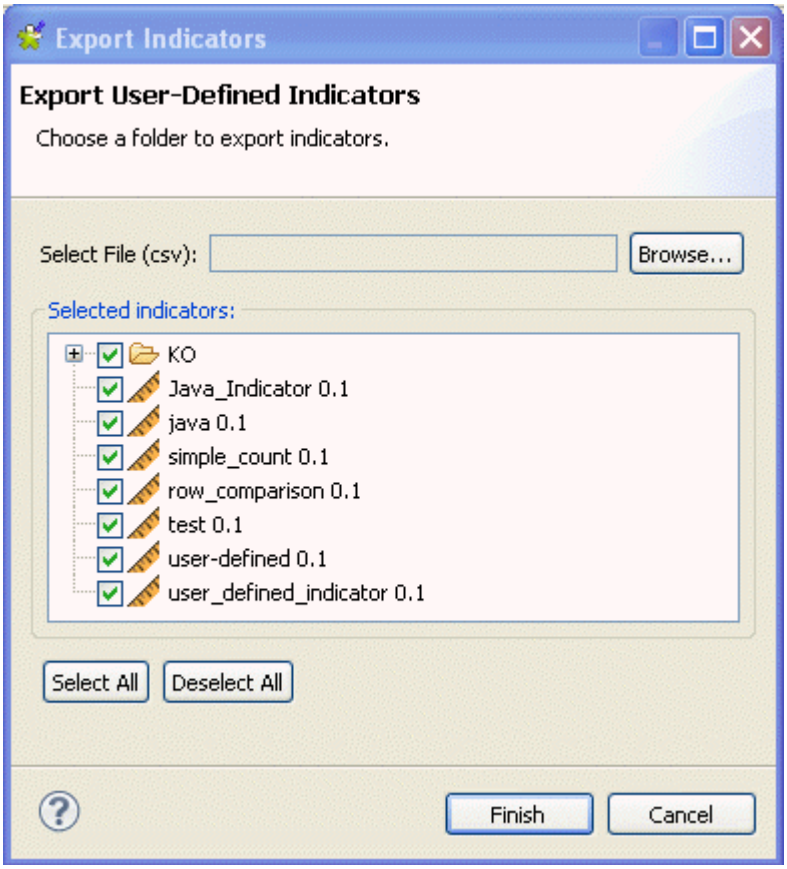
To export user-defined indicators to a csv file, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators** and then right-click **User Defined Indicators**.



2. From the contextual menu, select **Export Indicators**.

   The **[Export Indicators]** wizard opens with the check boxes of all indicators selected by default.



3. Browse to the csv file where to save the indicators.
4. If required, clear the check boxes of the indicators you do not want to export to the csv file.
5. Click **Finish** to close the wizard.

   All exported user-defined indicators are saved in the defined csv file.

How to export user-defined indicators to Talend Exchange

You can export user-defined indicators from your current version of studio to **Talend Exchange** where you can share them with other users.

Prerequisite(s): At least one user-defined indicator is created in the **Profiling** perspective of the studio.

To export user-defined indicators to **Talend Exchange**, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Indicators**.
2. Right-click the **User Defined Indicator** folder and select **Export for Talend Exchange**.

   The **[Export for Talend Exchange]** wizard is displayed.

3. Browse to the folder where to save indicators.
4. If required, clear the check boxes of the indicators you do not want to export to the specified folder.
5. Click **Finish** to close the wizard.

A distinct csv file is created for each exported indicator. Each csv file is compressed as a zip. All these zip files are saved in the defined folder. You need now to upload them to **Talend Exchange**at http://www.talendforge.org/exchange/top/help_guest.php.

9.2.3.4. How to import user-defined indicators

You can import indicators from a local csv file or from **Talend Exchange** into your studio and use them, as needed, on your column analyses.

You can also import user-defined indicators from folders or archive files. For further information, see section *Importing data profiling items or projects*.

How to import user-defined indicators from a csv file

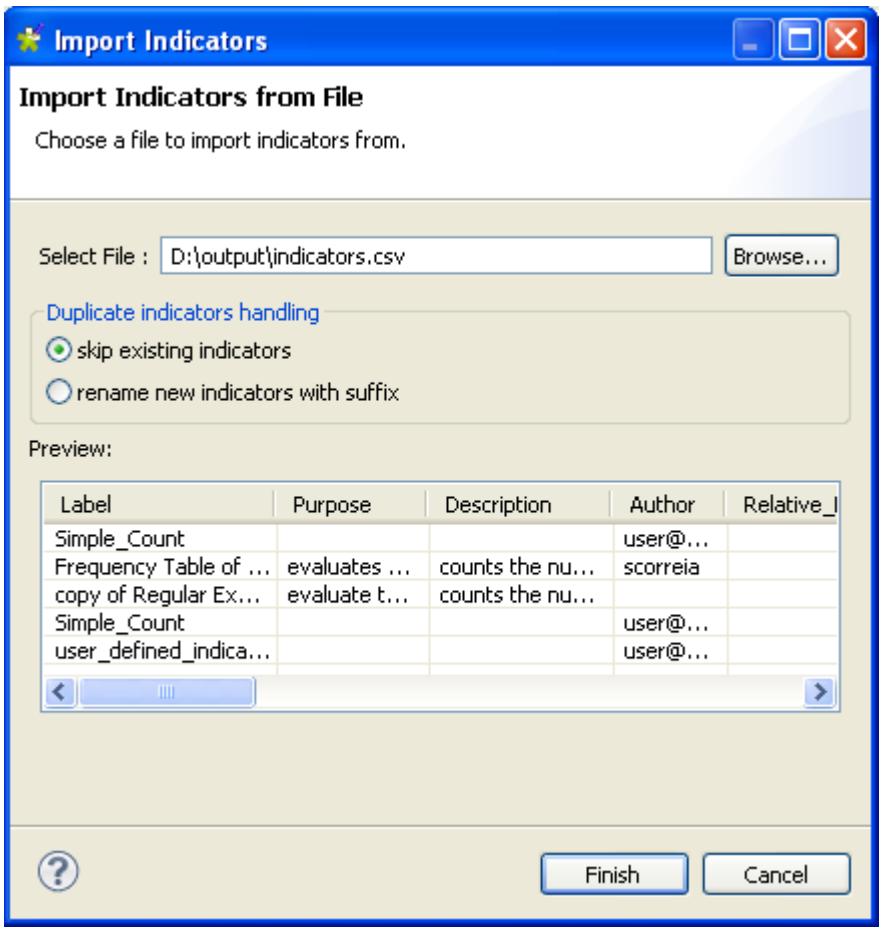
You can import indicators stored locally in a csv file to use them on your column analyses.

Prerequisite(s): You have already selected the **Profiling** perspective of the studio. The csv file is stored locally.

To import user-defined indicators from a csv file, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Indicators**.
2. Right-click **User Defined Indicators** and select **Import Indicators**.

   The **[Import Indicators]** wizard opens.



3. Browse to the csv file holding the user-defined indicators.
4. In the **Duplicate patterns handling** area, select:

Option	To...
skip existing indicators	import only the indicators that do not exist in the corresponding lists in the **DQ Repository** tree view. A warning message is displayed if the imported indicators already exist under the **Indicators** folder.
rename new indicators with suffix	identify each of the imported indicators with a suffix. All indicators will be imported even if they already exist under the **Indicators** folder.

5. Click **Finish** to close the wizard.

All imported indicators are listed under the **User Defined Indicators** folder in the **DQ Repository** tree view.

A warning icon next to the name of the imported user-defined indicator in the tree view identifies that it is not correct. You must open the indicator and try to figure out what is wrong.

How to import user-defined indicators from Talend Exchange

You can import user-defined indicators created by other users and stored in **Talend Exchange** into your current version of studio and use them, as needed, on your column analyses.

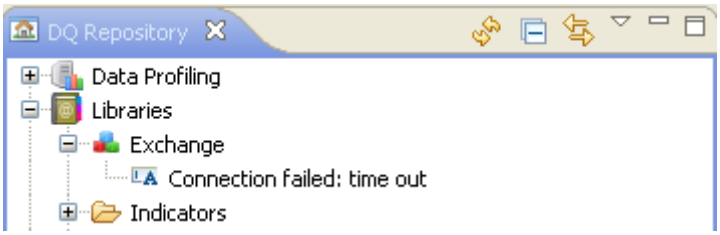
The indicators you can import from **Talend Exchange** include for example:

- *Order of Magnitude*: It computes the number of digits between the minimal value and maximal value of a numerical column.

- *Email validation via mail server*: This Java user-defined indicator connects to the mail server and checks if the email exists.

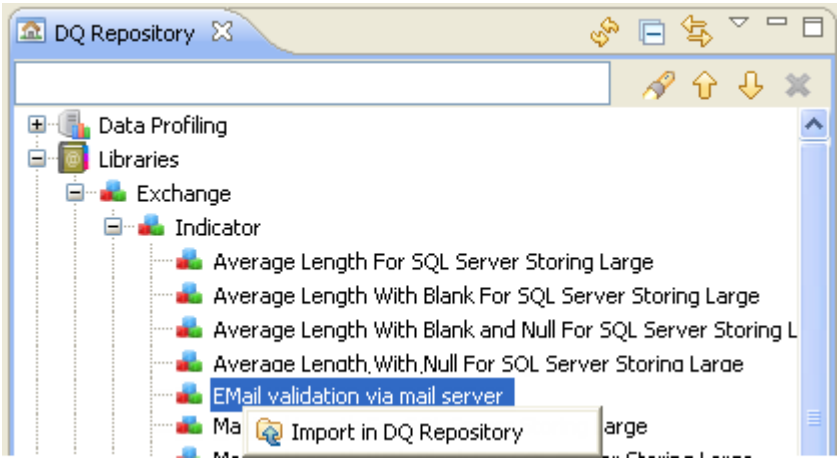
Prerequisite(s): You have already selected the **Profiling** perspective of the studio. Your network is up and running.

If you have connection problems, you will not be able to access any of the regular expressions or SQL patterns under the **Exchange** node in the **DQ Repository** tree view.



To import user-defined indicators from **Talend Exchange**, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Exchange**.
2. Under **Exchange**, expand **indicator** and right-click the name of the indicator you want to import, a Java user-defined indicator in this example.



You will have access only to versions that are compatible with the version of your current Studio.

3. Select **Import in DQ Repository**.

   If more than one version for the selected indicator is available on **Talend Exchange**, a dialog box is displayed to list the versions that are compatible with your current Studio version.

4. Select a version from the list and then click **OK**.

   A message is displayed to confirm the operation.

5. Click **OK** in the confirmation message.

The user-defined indicator is imported from **Talend Exchange** and listed under the **User Defined Indicators** folder in the **DQ Repository** tree view. You can now use this indicator on a column analysis to check emails by sending an SMTP request to the mail server.

You can also use the Studio to create an SQL user-defined indicator or a Java user-defined indicator from scratch. For further information, see section *How to create SQL user-defined indicators* and section *How to define Java user-defined indicators* respectively.

9.2.3.5. How to edit a user-defined indicator

You can open the editor of any system or user-defined indicator to check its settings and/or edit its definition and metadata in order to adapt it to a specific database type or need, if required.

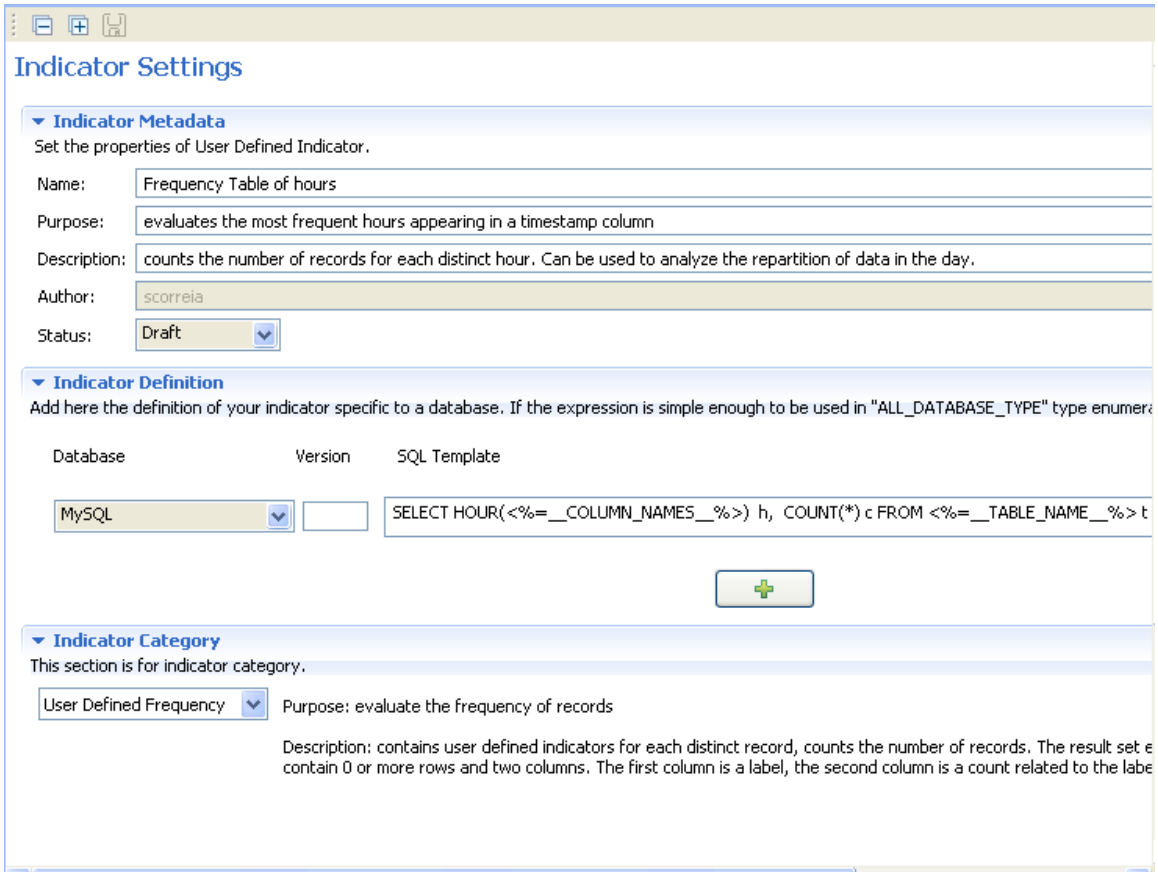
Prerequisite(s): At least one user-defined indicator is created in the **Profiling** perspective of the studio.

To edit the definition of a user-defined indicator, do the following:

1. In the **DQ Repository** tree view, expand **Libraries > Indicators**, and then browse through the indicator lists to reach the indicator you want to modify the definition of.

2. Right-click the indicator name and select **Open** from the contextual menu.

   The indicator editor opens displaying the selected indicator settings.



3. Modify the indicator metadata, if required, and then click **Indicator Definition** to display the relevant view.

   In this view, you can: edit indicator definition, change the selected database and add other indicators specific to available databases using the **[+]** button.

   If the indicator is simple enough to be used in all databases, select `Default` in the list.

4. Click **Indicator Category** to display the relevant view.

5. If required, change the indicator category in the list.

   The table below describes the different categories.

Indicator category	Description

User Defined Match (by-default category)	Uses user-defined indicators to evaluate the number of the data records that match a regular expression or an SQL pattern. The analysis results show the record matching count and the record total count.
User Defined Frequency	Uses user-defined indicators for each distinct data record to evaluate the record frequency that match a regular expression or an SQL pattern. The analysis results show the distinct count giving a label and a label-related count.
User Defined Real Value	Uses user-defined indicators which return real value to evaluate any real function of the data.
User Defined Count	Uses user-defined indicators that return a row count.

6. Click the save icon on top of the editor to save your changes.

When you edit an indicator, you modify the indicator listed in the DQ Repository tree view. Make sure that your modifications are suitable for all analyses that may be using the modified indicator.

9.2.3.6. How to duplicate a user-defined indicator

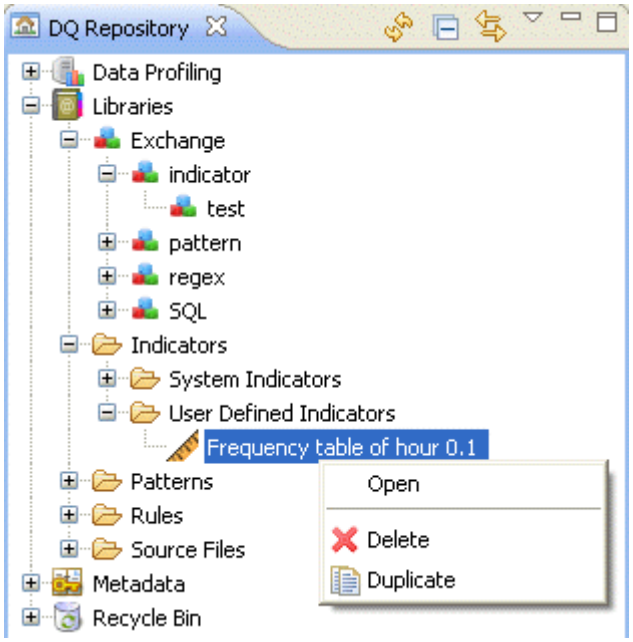
To avoid creating an indicator from scratch, you can duplicate an existing one in the indicator list. Once the copy is created, you can work around its metadata to have a new indicator and use it in data profiling analyses.

Prerequisite(s): At least one user-defined indicator has been defined in the **Profiling** perspective of the studio.

To duplicate a user defined indicator, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators**.

2. Browse through the user-defined indicator lists to reach the indicator you want to duplicate, right-click its name and select **Duplicate...** from the contextual menu.



The duplicated indicator is displayed under the **User Defined Indicators** folder in the **DQ Repository** tree view.

You can now open the duplicated indicator to modify its metadata and definition as needed. for more information on editing user-defined indicators, see section *How to edit a user-defined indicator*.

9.2.3.7. How to delete or restore a user-defined indicator

In the studio, you can delete a user-defined indicator definitely or restore it from the **Recycle Bin**. For details, refer to section *How to delete a regular expression or an SQL pattern*.

9.2.4. Indicator parameters

This section describes indicator parameters displayed in the different **Indicators Settings** dialog boxes.

Bins Designer

Possible value	Description
Minimal value	Beginning of the first bin.
Maximal value	End of the last bin.
Number of bins	Number of bins.

Blank Options

Possible value	Description
Aggregate nulls with blanks	When selected, null data is counted as zero length text field. This means that null data is treated as an empty string. When not selected, null data is treated as any other text data.
Aggregate blanks	When selected, blank texts (e.g. " ") are all grouped together and considered as an empty string. When not selected, blank texts are treated as any other text data.  In Oracle, empty strings and null strings are the same objects. Therefore, you must select or clear both check boxes in order to get consistent results.

Data Thresholds

Possible value	Description
Lower threshold	Data smaller than this value should not exist.
Upper threshold	Data greater than this value should not exist.

Frequency Table Parameters

Possible value	Description
Number of results shown	Number of displayed results.

Indicator Thresholds

Possible value	Description
Lower threshold	Lower threshold of matching indicator values.
Upper threshold	Higher threshold of matching indicator values.
Lower threshold(%)	Lower threshold of matching indicator values in percentage relative to the total row count.
Upper threshold(%)	Higher threshold of matching indicator values in percentage relative to the total row count.
Expected value	Only for the **Mode** indicator in the **Advanced Statistics**. Most probable value that should exist in the selected column.

Java Options

Possible value	Description
Characters to replace	List of the characters to be replaced.

Possible value	Description
Replacement characters	List of the characters that will take the place of the replaced characters.  Each character of the first field will be replaced by the character at the same position from the second field. For example, with the values "abc0123ABC,;.:" in the first field and "aaa9999AAApppp" in the second field any "a", "b" or "c" will be replaced by "a" and any "0", "1", "2" or "3" will be replaced by "9".

Phone number

Possible value	Description
Country	Country ISO2 code of the phone number.

Text Parameters

Possible value	Description
Ignore case	When selected, comparison of text data is not case sensitive.

Text Length

Possible value	Description
Count nulls	When selected, null data is counted as zero length text field.
Count blanks	When selected, blank texts (e.g. " ") are counted as zero length text fields.

Chapter 10. Other important management procedures

This chapter provides the information you need to carry out some basic procedures including setting preferences of analysis editors and analysis results, creating SQL queries, setting data parser rules, importing/exporting data quality items and upgrading projects from older versions.

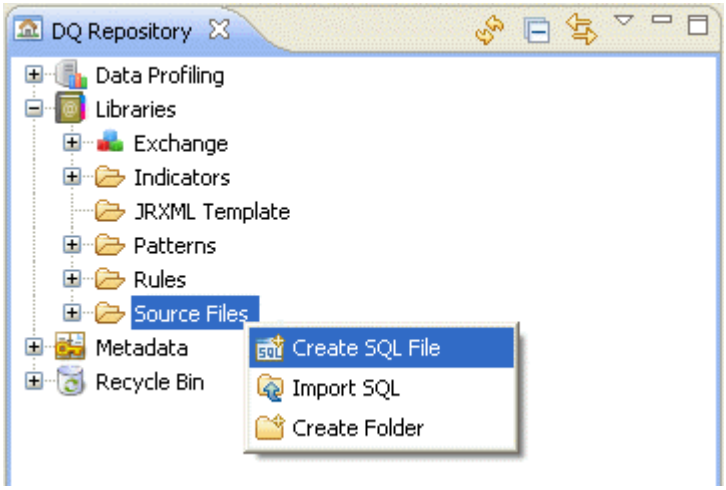
Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

10.1. Creating and storing SQL queries

From the studio, you can query and browse a selected database using the SQL Editor and then to store these SQL queries under the **Source Files** folder in the **DQ Repository** tree view. You can then open the SQL Editor on any of these stored queries to rename, edit or execute the query.

To create an SQL query, do the following:

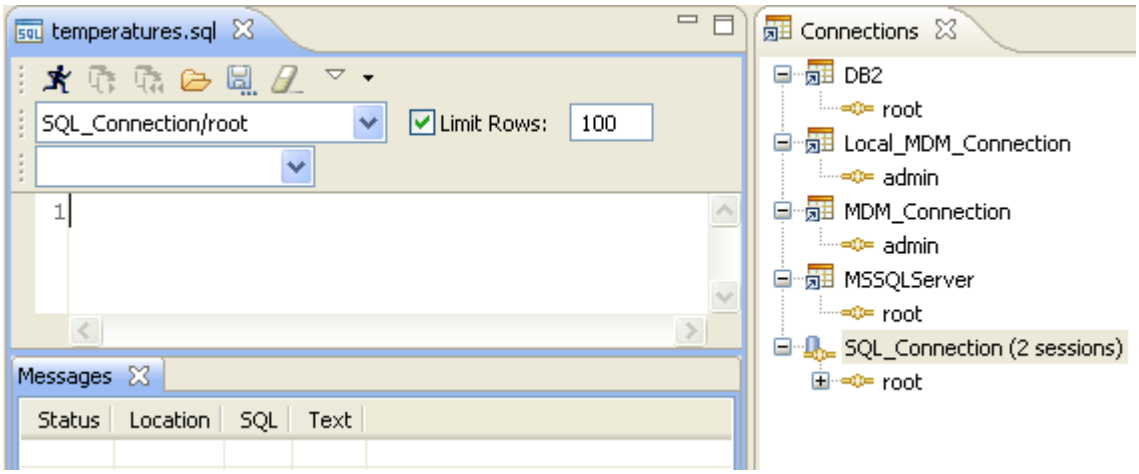
1. In the **DQ Repository** tree view, expand **Libraries**.



2. Right-click **Source Files** and select **Create SQL File** from the contextual menu. The **[Create SQL File]** dialog box is displayed.

3. In the **Name** field, enter a name for the SQL query you want to create and then click **Finish** to proceed to the next step.

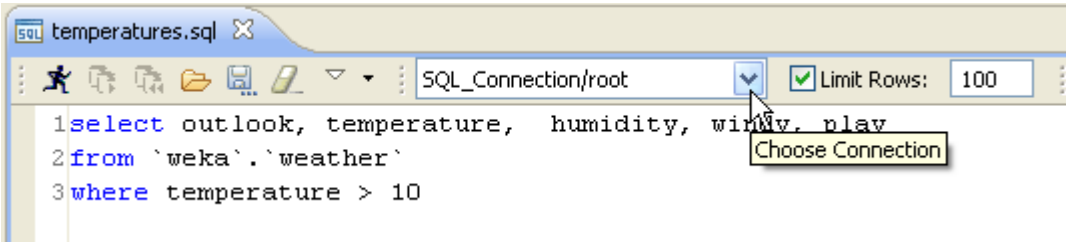
   The SQL Editor opens on the new SQL query.



If the **Connections** view is not open, use the combination **Window > Show View > Data Explorer > Connections** to open it.

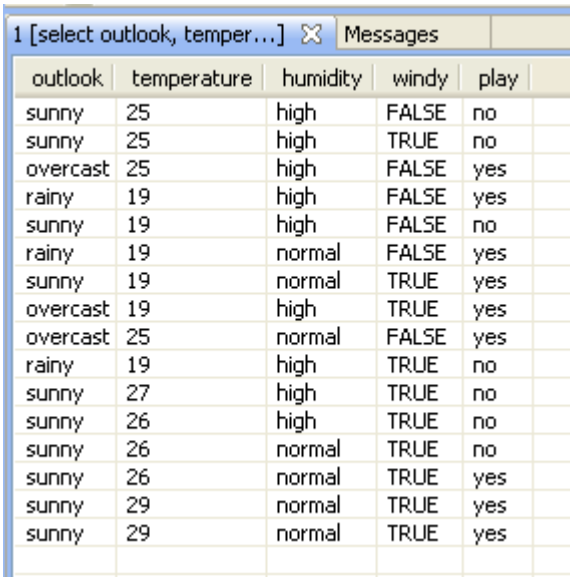
4. Enter your SQL statement in the SQL Editor.

5. From the **Choose Connection** list, select the database you want to run the query on.



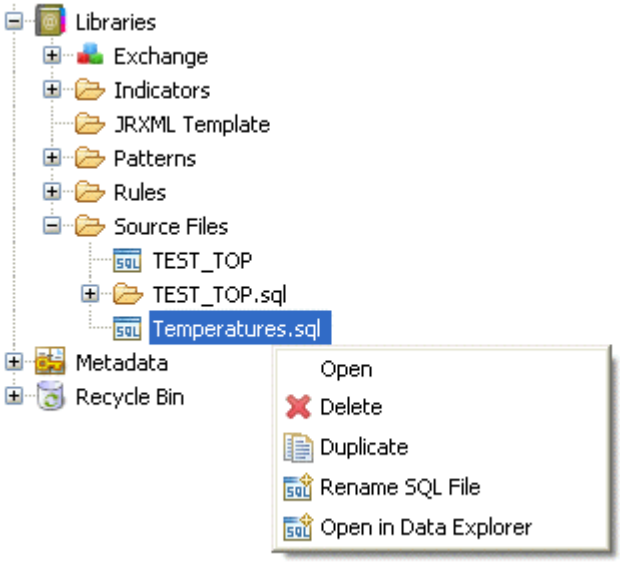
6. On the SQL Editor toolbar, click to execute the query on the defined base table(s).

   Data rows are retrieved from the defined base table(s) and displayed in the editor.



outlook	temperature	humidity	windy	play
sunny	25	high	FALSE	no
sunny	25	high	TRUE	no
overcast	25	high	FALSE	yes
rainy	19	high	FALSE	yes
sunny	19	high	FALSE	no
rainy	19	normal	FALSE	yes
sunny	19	normal	TRUE	yes
overcast	19	high	TRUE	yes
overcast	25	normal	FALSE	yes
rainy	19	high	TRUE	no
sunny	27	high	TRUE	no
sunny	26	high	TRUE	no
sunny	26	normal	TRUE	no
sunny	26	normal	TRUE	yes
sunny	29	normal	TRUE	yes
sunny	29	normal	TRUE	yes

   A file for the new SQL query is listed under **Source Files** in the **DQ Repository** tree view.



7. Right-click an SQL file and from the contextual menu select:

Option	To...
Open	open the selected Query file
Duplicate	create a copy of the selected Query file
Rename SQL File	open a dialog box where you can edit the name of the query file

Open in Data Explorer	open in the data explorer the SQL editor on the selected query file
Delete	delete the query file The deleted item will go to the **Recycle Bin** in the **DQ Repository** tree view. You can restore or delete such item via a right-click on the item. You can also empty the recycle bin via a right-click on it.

When you open a query file in the SQL Editor, make sure to select the database connection from the **Choose Connection** list before executing the query. Otherwise the run icon on the editor toolbar will be unavailable.

When you create or modify a query in a query file in the SQL Editor and try to close the editor, you will be prompted to save the modifications. The modifications will not be taken into account unless you click the save icon on the editor toolbar.

10.2. Importing data profiling items or projects

You can import data profiling items (analyses, database connections, patterns and indicators, etc.) from various projects or different versions of the studio. You can import these items from the **import items** icon in the studio.

You can not import an item without all its dependencies. When you try to import an analysis for example, all its dependencies such as a metadata connection and the patterns and indicators used in this analysis will be selected by default and imported with the analysis.

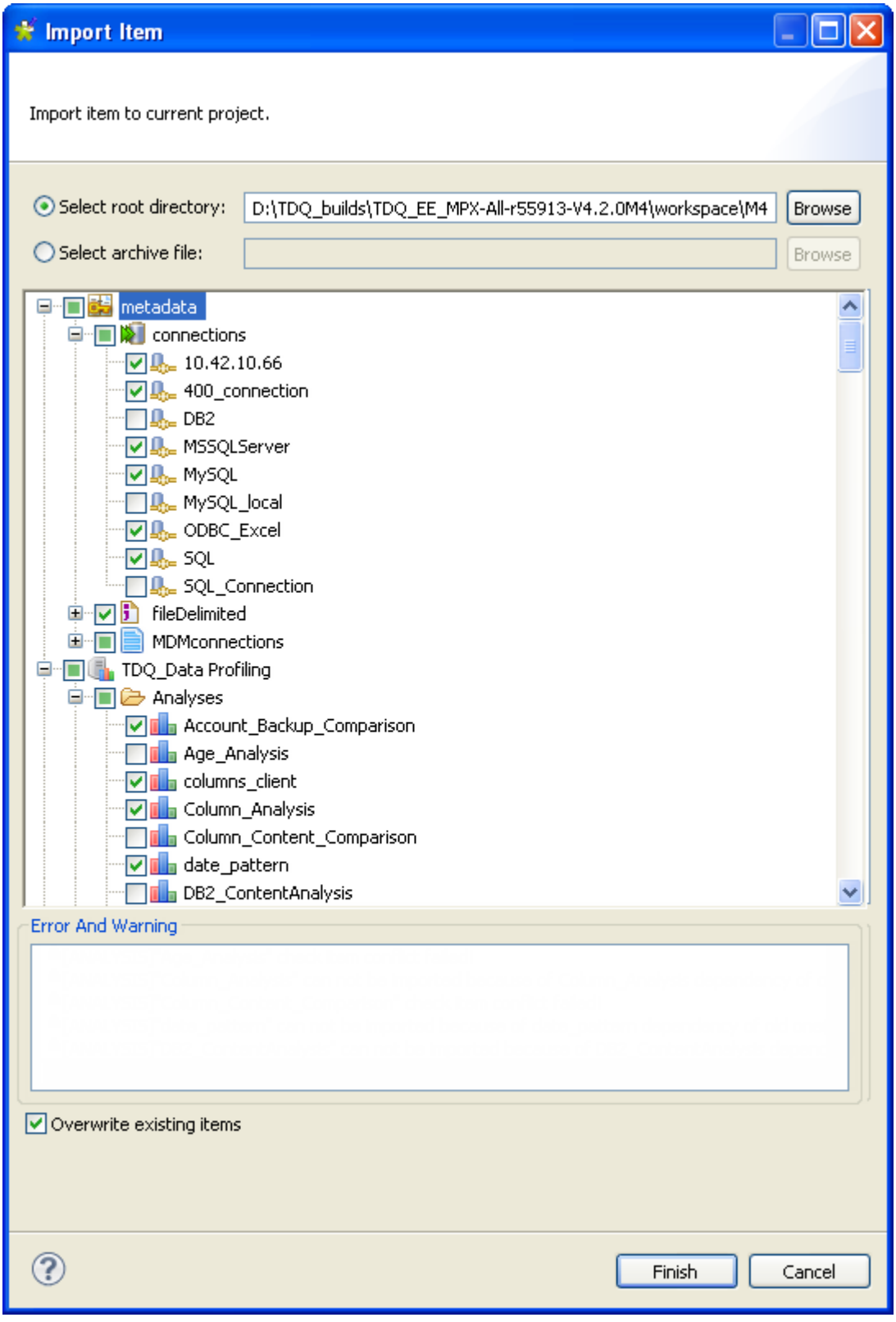
You can not import into your current Studio data profiling items created in versions older than 4.0.0. To use such items in your current Studio, you must carry out an upgrade operation. For further information, see section Upgrading projects items from older versions.

Prerequisite(s): You have access to another studio version in which data profiling items have been created.

To import one or more data profiling items, do the following:

1. In the **Profiling** perspective, click the  icon on the toolbar.

   The **[Import Item]** wizard is displayed.



2. Select the root directory or the archive file option according to whether the data profiling items you want to import are in the *workspace* file within the Studio directory or are already exported into a zip file.

   - If you select the root directory option, click **Browse** and set the path to the project folder containing the items to be imported within the *workspace* file of the Studio directory.

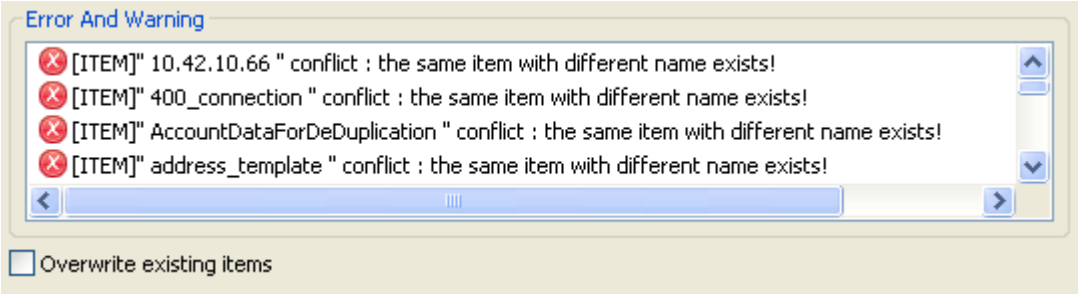
All items and their dependencies that do not exist in your current Studio are selected by default in the dialog box.

- If you select the archive file option, click **Browse** and set the path to the archive file that holds the data profiling items you want to import.

  All items and their dependencies that do not exist in your current Studio are selected by default in the dialog box.

3. Select the **Overwrite existing items** check box if some error and warning messages are listed in the **Error and Warning** area.

   This means that items with the same names already exist in the current Studio.



   The imported items will replace the existing ones.

   For release 5.2.0 and above, when you try to import some system indicators that are modified in a Studio version, they will not overwrite the system indicators in the current Studio. All modifications from older versions will be integrated with the system indicators in the current Studio. This enables you to use these indicators on your analyses in the current Studio without a problem.

4. Select or clear the check boxes of the data profiling items you want or do not want to import according to your needs.

   All dependencies for the selected item are selected by default. When you clear the check box of an item, the check boxes of the dependencies of this item are automatically cleared as well. Also, an error message will display on top of the dialog box if you clear the check box of any of the dependencies of the selected item.

5. Click **Finish** to validate the operation.

   The imported items display under the corresponding folders in the **DQ Repository** tree view.

   *If you import SQL Servers (2005 or 2008) connections into your current Studio, a warning icon is docked on the connection names in the **DB connections** folder. This indicates that the driver path for these connections is empty. You must open the connection wizard and redefine the connection manually to set the path to a JDBC driver you can download from the Microsoft download center.*

   For further information on editing a database connection, see section How to open or edit a database connection.

   You can also set the path to a JDBC driver for a group of database connections simultaneously in order not to define them one by one. For further information, see section Migrating a group of connections.

You can also import local project folders from the login window of your studio. For further information, see section *Launching the studio*.

10.3. Exporting data profiling items

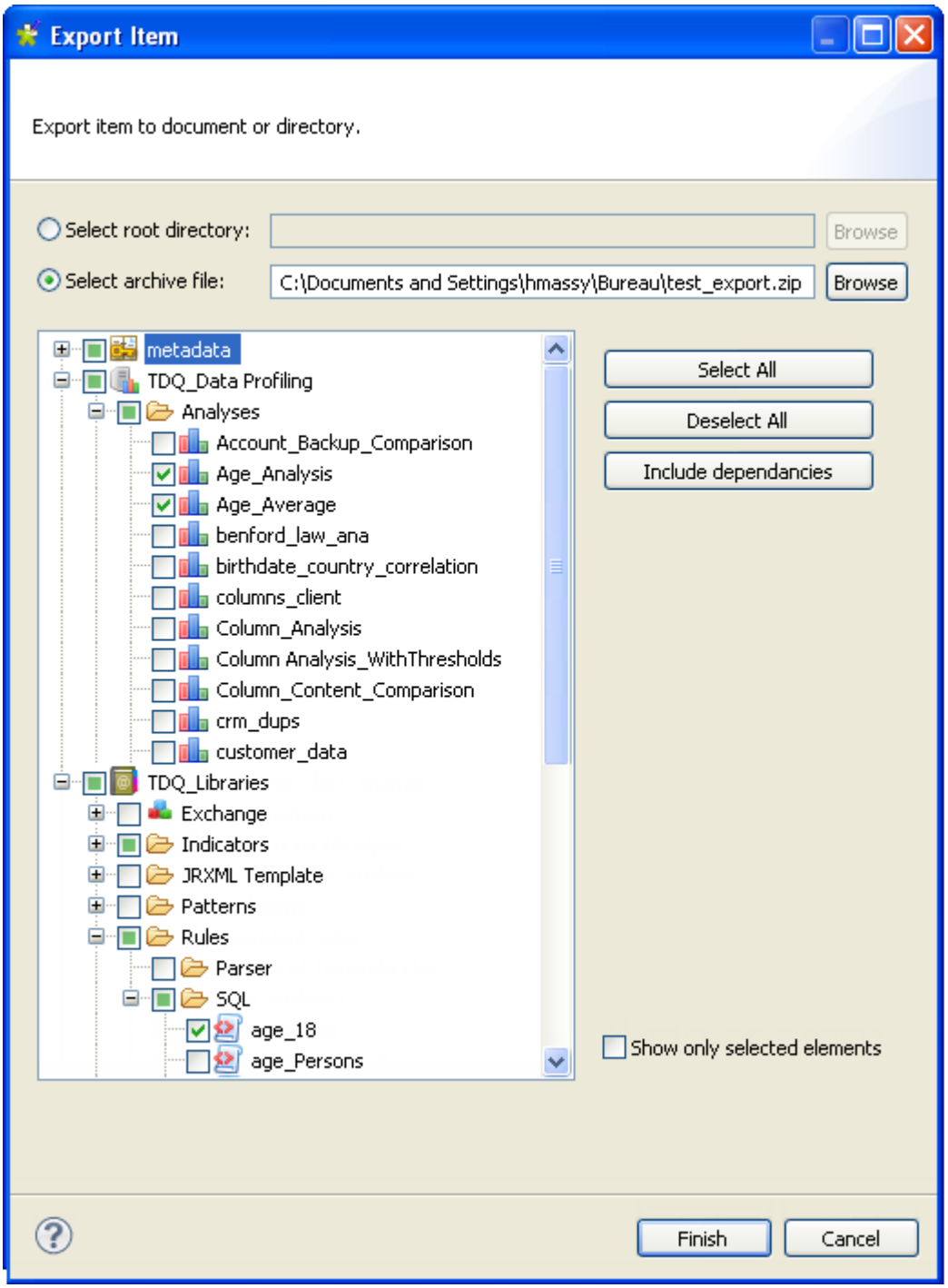
You can export data profiling items (analyses, database connections, patterns and indicators, etc.) in the current instance of the Studio to folders or archive files.

Prerequisite(s): At least, one data profiling item has been created in the studio.

To export data profiling items, do the following:

1. In the **Profiling** perspective, click the icon on the toolbar.

   The **[Export Item]** wizard is displayed.



2. Select the root directory or archive file option and then click **Browse...** and browse to the file/archive where you want to export the data profiling items.

3. Select the check boxes of the data profiling items you want to export or use the **Select All** or **Deselect All** tabs.

   When you select an analysis check box, all analysis dependencies including the metadata connection and any patterns or indicators used in this analysis are selected by default. Otherwise, if you have an error message on top of the dialog

box that indicates any missing dependencies, click the **Include dependencies** tab to automatically select the check boxes of all items necessary to the selected data profiling analysis.

4. If required, select the **Show only selected elements** check box to have in the export list only the selected data profiling elements.

5. Click **Finish** to validate the operation.

   A progress bar is displayed to indicate the progress of the export operation and the data profiling items are exported in the defined place.

10.4. Migrating a group of connections

You can import database connections from various projects or various versions of the Studio.

Some of the migrated JDBC connections may have a warning icon docked on their names in the **DB connections** folder in the **DQ Repository** tree view. This indicates that the driver path for these connections is empty after migration.

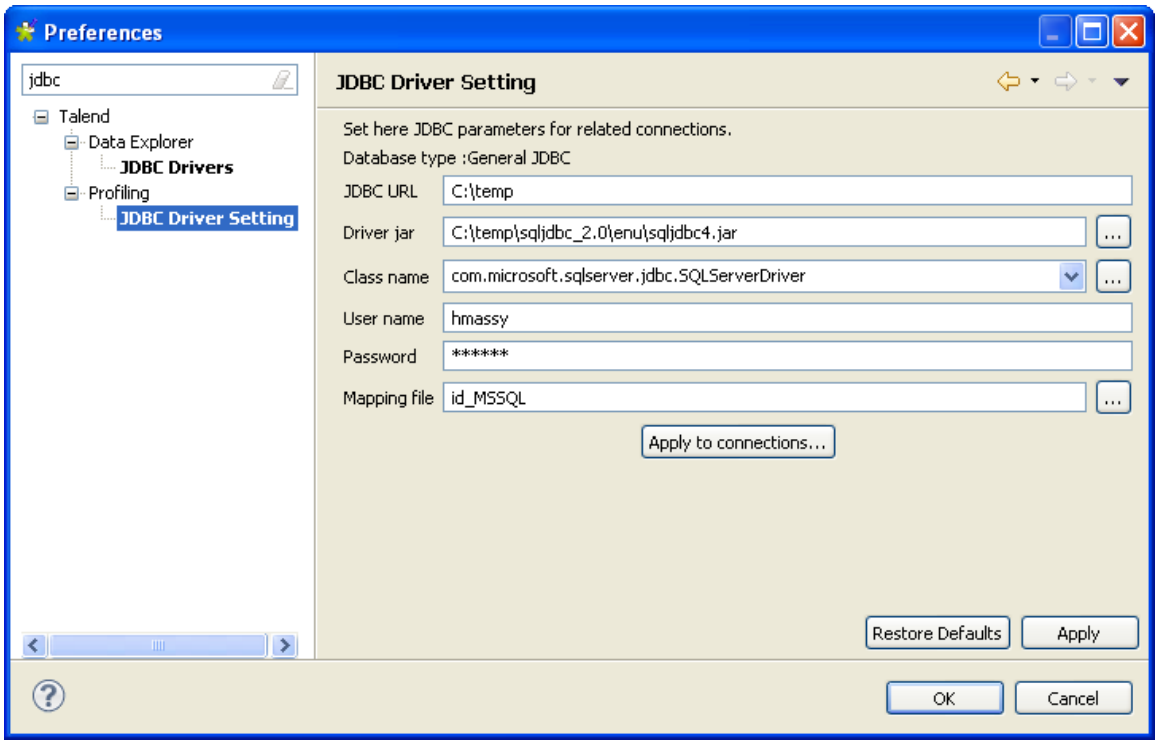
Setting the driver path manually for each of the connections could be tedious especially if you have imported big number. The studio enables you to set the driver path once for all. You may download such a driver from the Microsoft download center, for example.

Prerequisite(s): You have already migrated your database connections from an older version of the studio as outlined in section *Importing data profiling items or projects*.

To migrate a group of connections simultaneously, do the following:

1. In the menu bar, select **Window > Preferences** to display the **[Preferences]** window.

2. In the search field, type *jdbc* and then select **JDBC Driver Setting** to open the corresponding view.



3. Set the JDBC parameters in the corresponding fields, and then click **Apply to connections...**.

A dialog box is displayed to list all the JDBC connections that do not have the required JDBC driver after migration.

4. Select the check boxes of the connections for which you want to apply the driver settings and then click **OK**.

   A confirmation message is displayed.

5. Click **OK** to close the confirmation message.

6. In the **[Preferences]** window, click **OK**.

   A confirmation message is displayed.

7. Click **OK** to close the message and the **[Preferences]** window.

10.5. Upgrading projects items from older versions

The below procedure concerns only the migration of data profiling items from versions older than 4.0.0. To migrate your data profiling items from version 4.0.0 onward, you simply need to import them into your current Studio or to import the project itself. For further information, see section Importing data profiling items or projects, and also see the section on importing projects in the Talend Open Studio for Data Integration User Guide.

To migrate data profiling items (analyses, database connections, patterns and indicators, etc.) created in versions older than 4.0.0, do the following:

1. From the folder of the old version studio, copy the workspace file and paste it in the folder of your current Studio. Accept to replace the current workspace file with the old file.

2. Launch the Studio connecting to this workspace.

The upgrade operation is completed once the Studio is completely launched, and you should have access to all your data profiling items.

Regarding system indicators during migration, please pay attention to the following:

- *When you upgrade the repository items to version 4.2 from a prior version, the migration process overwrites any changes you made to the system indicators.*
- *When you upgrade the repository items from version 4.2 to version 5.0, you do not lose any changes you made to the system indicators.*

Chapter 11. Managing existing analyses

This chapter provides the information you need to perform basic management procedures for all analysis created in *Talend Open Studio for Data Quality*.

Before starting data profiling management procedures, you need to be familiar with the studio Graphical User Interface (GUI). For more information, see appendix *The studio management GUI*.

11.1. Procedures for all types of analyses

The procedures below provide detailed information on basic management options for all types of the analyses listed under the **Analyses** folder in the **DQ Repository** tree view.

From the contextual menu of the selected analysis, you can open, execute, duplicate or delete this analysis. You can also add a task to the selected analysis.

11.1.1. Opening an analysis

Prerequisite(s): At least one analysis has been created in the **Profiling** perspective of the studio.

To open an analysis, do the following:

1. In the **DQ Repository** tree view, expand **Data Profiling > Analyses**.
2. Either:
   - double-click the analysis you want to open, or,
   - right-click the analysis you want to open and select **Open** from the contextual menu.

   The corresponding analysis editor is displayed.
3. If required, click **Refresh the graphics** to the right of the editor to display the results of the analysis.
4. If required, click the **Analysis results** button at the bottom of the editor to open a more detailed view of the analysis results.

11.1.2. Executing an analysis

Prerequisite(s): At least one analysis has been created in the **Profiling** perspective of the studio.

To execute an analysis, do the following:

1. In the **DQ Repository** tree view, expand **Data Profiling > Analyses**.
2. Right-click the analysis you want to execute and select **Run** from the contextual menu.

   A progress bar is displayed to convey the progress of the analysis execution.

You can execute many analyses simultaneously if you select them, right-click the selection and finally click **Run**.

11.1.3. Duplicating an analysis

To avoid creating an analysis from scratch, you can duplicate an existing one in the **Analyses** folder and work around its metadata to have a new analysis.

Prerequisite(s): At least one analysis has been created in the **Profiling** perspective of the studio.

To duplicate an analysis, do the following:

1. In the **DQ Repository** tree view, expand **Data profiling > Analyses**.
2. Right-click the analysis you want to duplicate and select **Duplicate...** from the contextual menu.

   The duplicated analysis shows in the analysis list in the **DQ Repository** tree view. You can now open the duplicated analysis and modify its metadata as needed.

11.1.4. Adding a task to an analysis

You can add a task to an analysis to indicate a problem that needs to be solved later, for example.

For more information, see section *Managing tasks*.

11.1.5. Deleting or restoring an analysis

Prerequisite(s): At least one analysis has been created in the **Profiling** perspective of the studio.

To delete an analysis, do the following:

1. In the **DQ Repository** tree view, expand **Data Profiling > Analyses**.
2. Right-click the analysis you want to delete and select **Delete** from the contextual menu.

   The analysis is moved to the **Recycle Bin**.

You can also delete the analysis permanently by emptying the recycle bin. To empty the **Recycle Bin**, do the following:

1. Right-click the **Recycle Bin** and select **Empty recycle bin**.

   A confirmation dialog box is displayed.
2. Click **Yes** to empty the recycle bin.

To restore an analysis from the **Recycle Bin**, do the following:

- In the **Recycle Bin**, right-click the analysis and select **Restore**.

  The analysis is moved back to the **Data profiling** node.

11.2. Managing tasks

In the studio, it is possible to add tasks to different items, display the task list and delete any completed task from the task list.

You can add tasks to different items either:

- in the **DQ Repository** tree view on connections, catalogs, schemas, tables, columns and created analyses,
- or, on columns, or patterns and indicators set on columns directly in the current analysis editor.

For example, you can add a general task to any item in a database connection via the **Metadata** node in the **DQ Repository** tree view. You can add a more specific task to the same item defined in the context of an analysis through the **Analyses** node. And finally, you can add a task to a column in an analysis context (also to a pattern or an indicator set on this column) directly in the current analysis editor.

The procedure to add a task to any of these items is exactly the same. Adding tasks to such items will list these tasks in the **Tasks** list accessible through the **Window > Show view...** combination. Later, you can open the editor corresponding to the relevant item by double-clicking the appropriate task in the **Tasks** list.

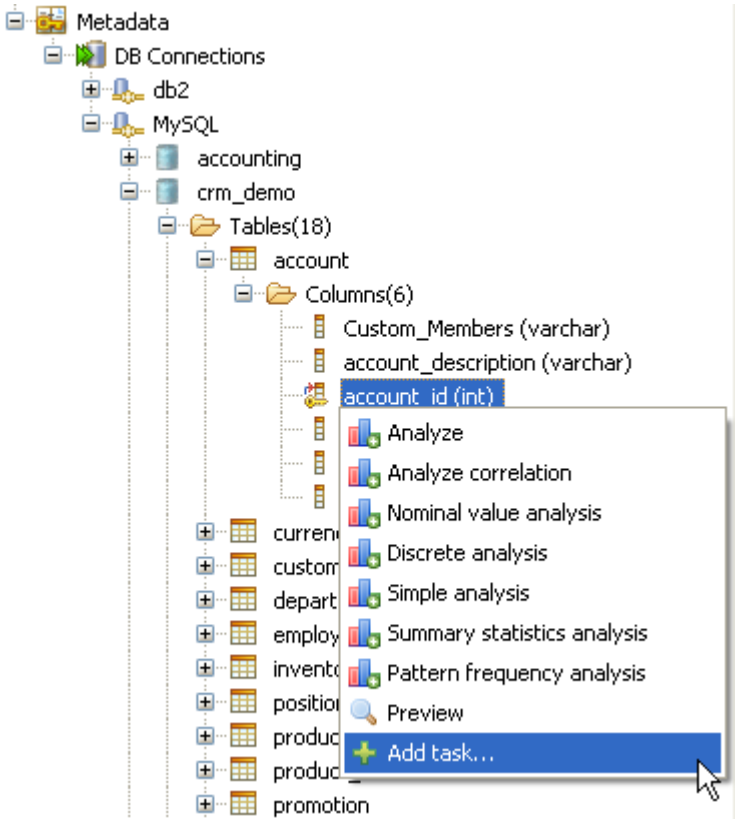
For examples on how to add a task to different items, see the sections below.

11.2.1. Adding a task to a column in a database connection

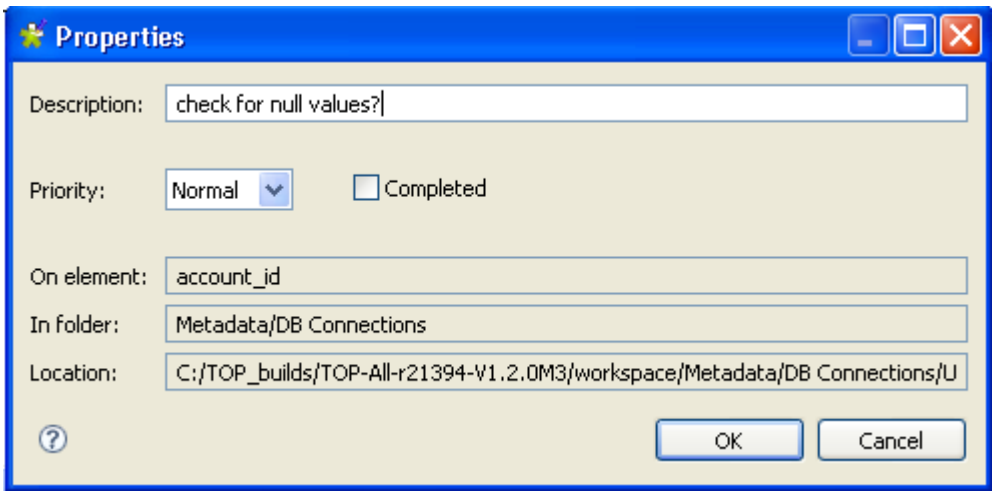
Prerequisite(s): At least, one database connection has been created in the **Profiling** perspective of the studio. For further information, see section *Connecting to a database*.

To add a task to a column in a database connection, do the following:

1. In the **DQ Repository** tree view, expand **Metadata > DB Connections**.
2. Navigate to the column you want to add a task to, `account_id` in this example.
3. Right-click the `account_id` and select **Add task...** from the contextual menu.



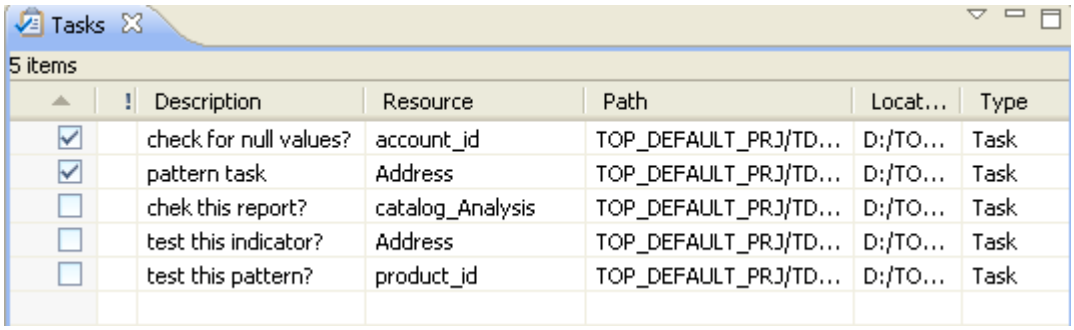
The **[Properties]** dialog box opens showing the metadata of the selected column.



4. In the **Description** field, enter a short description for the task you want to carry on the selected item.

5. In the **Priority** list, select the priority level and then click **OK** to close the dialog box.

   The created task is added to the **Tasks** list. For more information on how to access the task list, see section *Displaying the task list*.



From the task list, you can:

- double-click a task to open the editor where this task has been set.
- select the task check box once the task is completed in order to be able to delete it.
- filter the task view according to your needs using the options in a menu accessible through the drop-down arrow on the top-right corner of the **Tasks** view. For further information about filtering the task list, see section *Filtering the task list*.

11.2.2. Adding a task to an item in a specific analysis

The below procedure gives an example of adding a task to a column in an analysis. You can follow the same steps to add tasks to other elements in the created analyses.

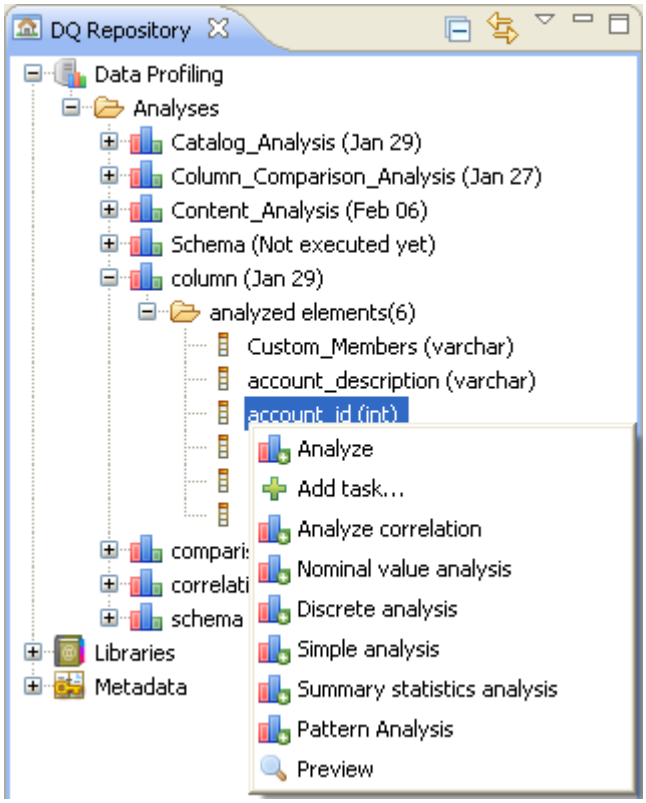
Prerequisite(s): The analysis has been created in the **Profiling** perspective of the studio.

To add a task to an item in an analysis, do the following:

1. In the **DQ Repository** tree view, expand **Analyses**.

2. Expand an analysis and navigate to the item you want to add a task to, the `account_id` column in this example.

3. Right-click `account_id` and select **Add task...** from the contextual menu.



4. Follow the steps outlined in section *Adding a task to a column in a database connection* to add a task to `account_id` in the selected analysis.

For more information on how to access the task list, see section *Displaying the task list*.

11.2.3. Adding a task to an indicator in a column analysis

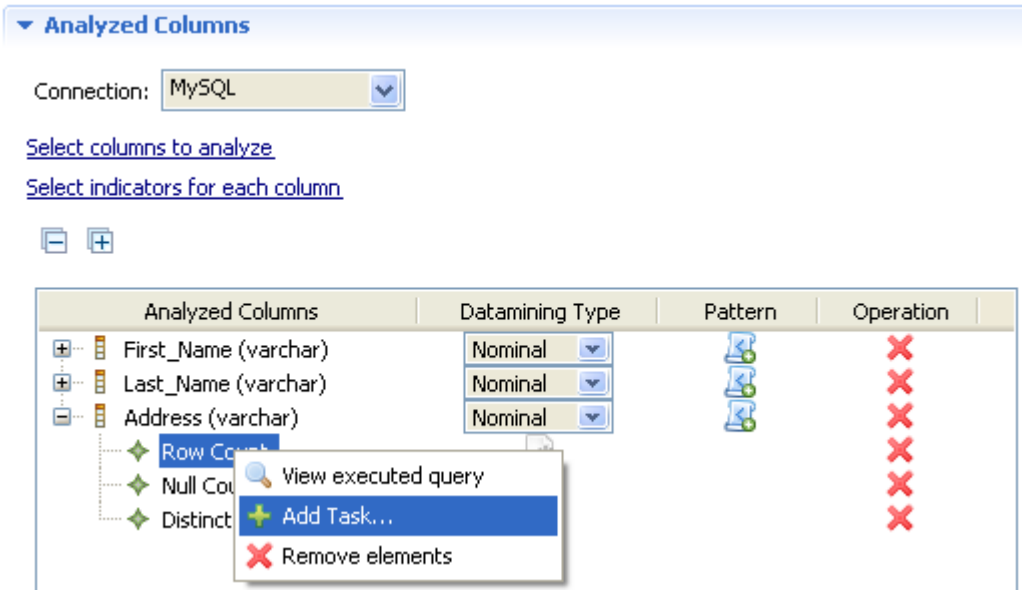
In the open analysis editor, you can add a task to the indicators set on columns. This task can be used, for example, as a reminder to modify the indicator or to flag a problem that needs to be solved later.

Prerequisite(s):

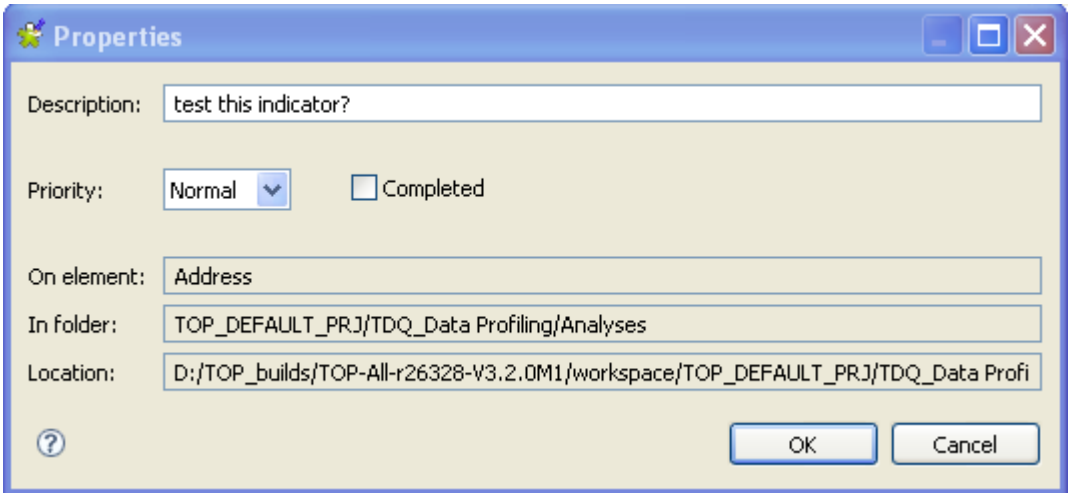
- A column analysis is open in the analysis editor in the **Profiling** perspective of the studio.
- At least one indicator is set for the columns to be analyzed.

To add a task to an indicator, do the following:

1. In the open analysis editor, click **Analyzed columns** to open the relevant view.
2. In the **Analyzed Columns** list, right-click the indicator name and select **Add task...** from the contextual menu.



The **[Properties]** dialog box opens showing the metadata of the selected indicator.



3. In the **Description** field, enter a short description for the task you want to attach to the selected indicator.
4. On the **Priority** list, select the priority level and then click **OK** to close the dialog box. The created task is added to the **Tasks** list.

For more information on how to access the task list, see section *Displaying the task list*.

11.2.4. Displaying the task list

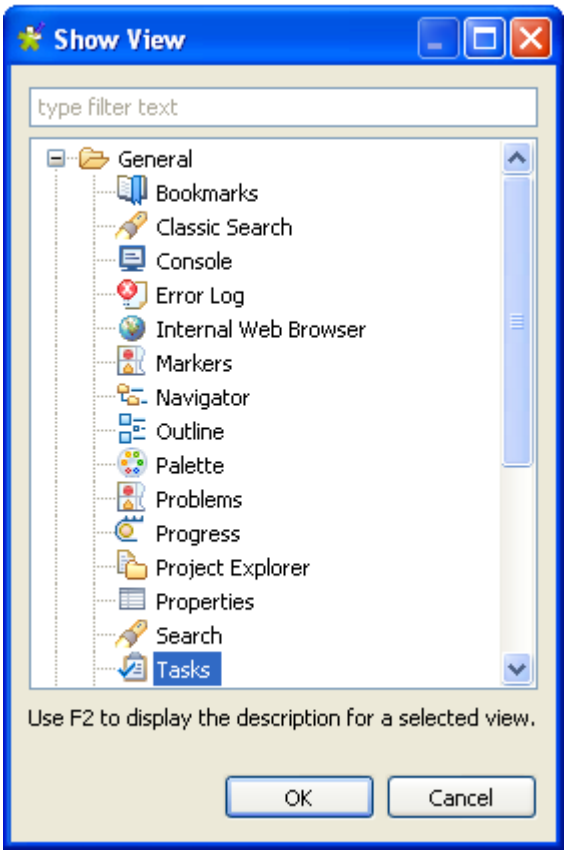
Adding tasks to items will list these tasks in the **Tasks** list.

Prerequisite(s): At least, one task is added to an item in the **Profiling** perspective of the studio.

To access the **Tasks** list, do the following:

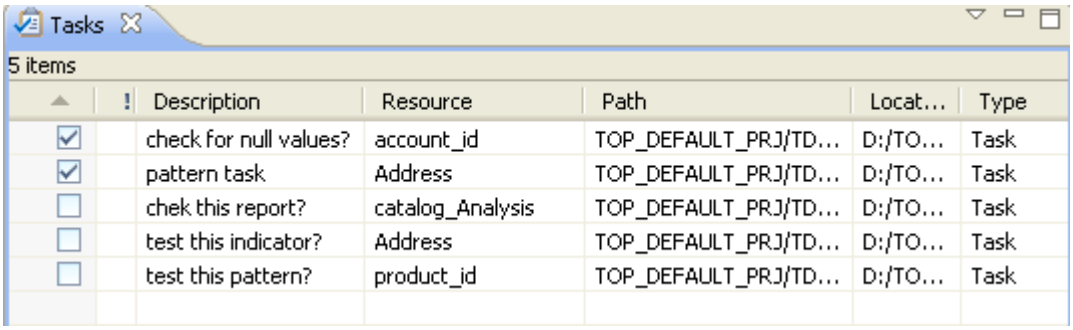
1. On the menu bar of *Talend Open Studio for Data Quality*, select **Window > Show view...** .

   The **[Show View]** dialog box is displayed.



2. Expand **General** and then select **Tasks**.
3. Click **OK** to proceed to the next step.

   The **Tasks** view opens in the **Profiling** perspective of the studio listing the added task(s).



	!	Description	Resource	Path	Locat...	Type
☑		check for null values?	account_id	TOP_DEFAULT_PRJ/TD...	D:/TO...	Task
☑		pattern task	Address	TOP_DEFAULT_PRJ/TD...	D:/TO...	Task
☐		chek this report?	catalog_Analysis	TOP_DEFAULT_PRJ/TD...	D:/TO...	Task
☐		test this indicator?	Address	TOP_DEFAULT_PRJ/TD...	D:/TO...	Task
☐		test this pattern?	product_id	TOP_DEFAULT_PRJ/TD...	D:/TO...	Task

4. If required, double-click any task in the **Tasks** list to open the editor corresponding to the item to which the task is attached.

You can create different filters for the content of the task list. For further information, see section *Filtering the task list*.

11.2.5. Filtering the task list

In the **Profiling** perspective of the studio, the **Tasks** view lists all the tasks you create in the studio.

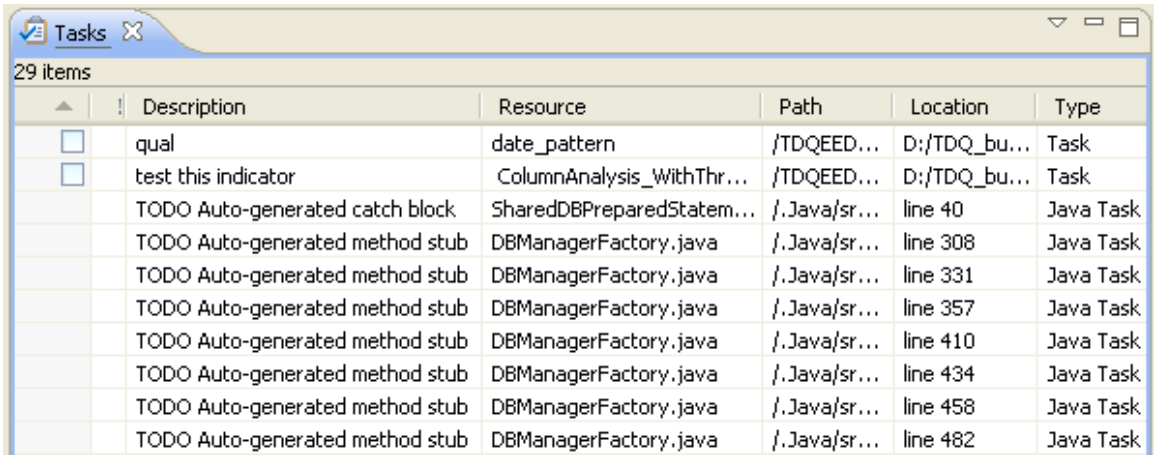
You can create filters to decide what to list in the task view.

Prerequisite(s): At least, one task is added to an item in the **Profiling** perspective of the studio.

To filter tasks in the **Tasks** view, do the following:

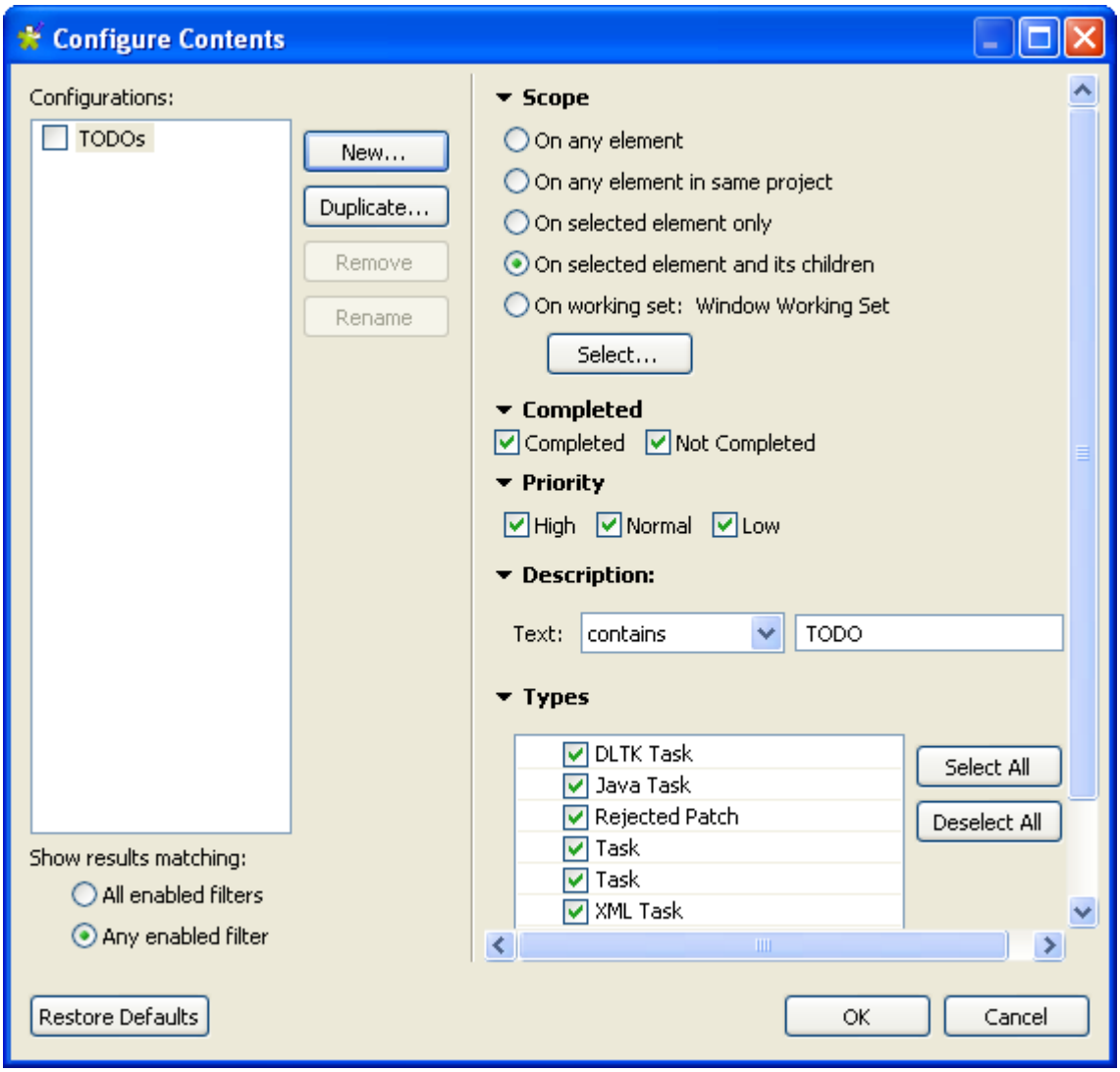
1. Follow the steps outlined in section *Displaying the task list* to open the task list.

   The **Tasks** view is displayed.



2. Click the drop-down arrow in the top right corner of the view, and then select **Configure contents...**.

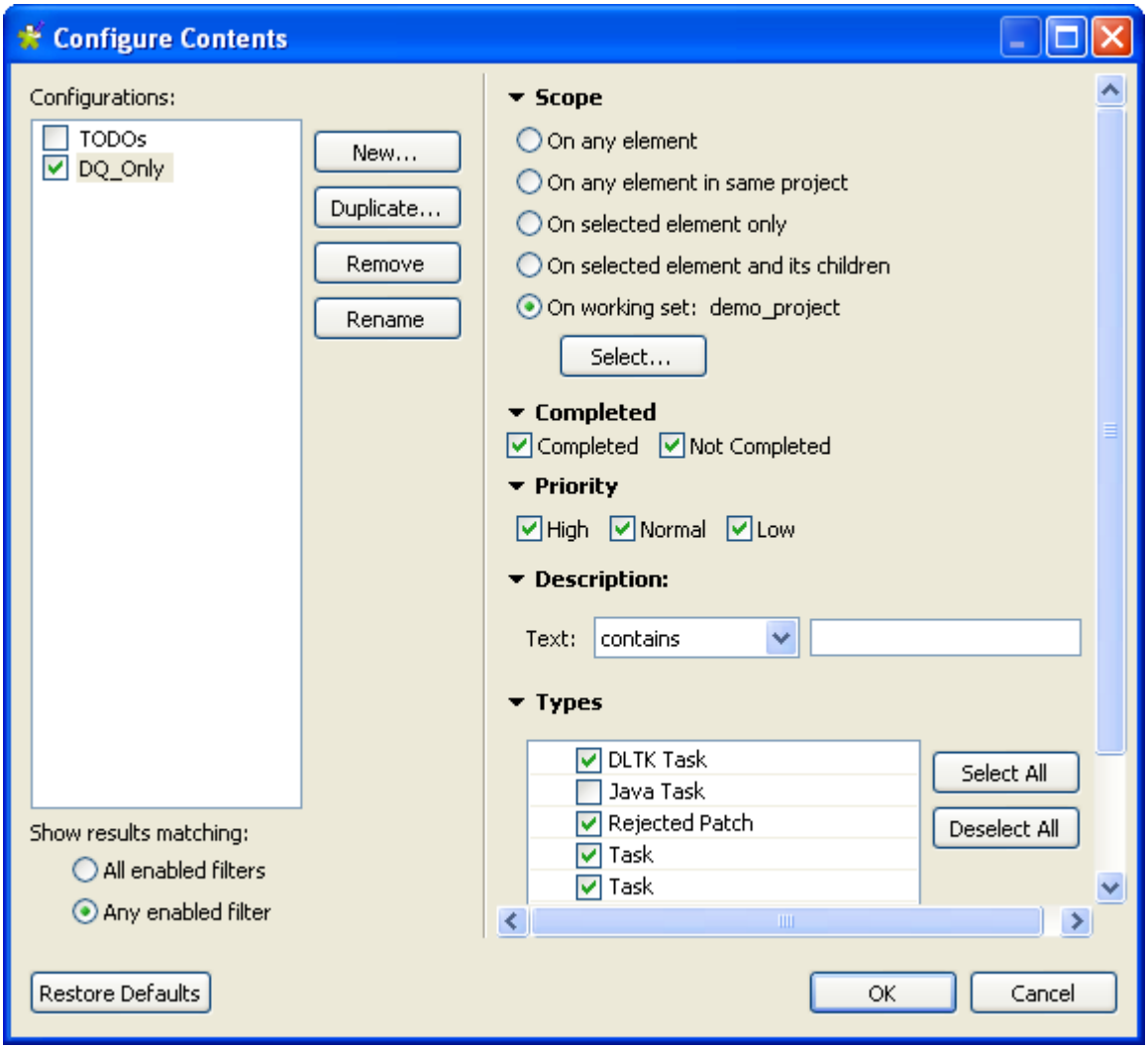
   The **[Configure contents...]** dialog box is displayed showing the by-default configuration.



3. Click **New** to open a dialog box and then enter a name for the new filter.

4. Click **OK** to close the dialog box

   The new filter is listed in the **Configurations** list.

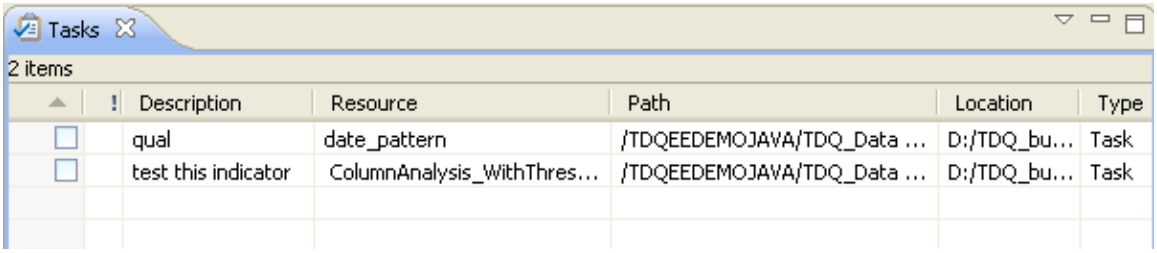


5. Set the different options for the new filter as the following:

   - From the **Scope** list, select a filter scope option, and then click **Select...** to open a dialog box where you can select a working set for your filter.
   - Select whether you want to display completed or not completed tasks or both of them.
   - Select to display tasks according to their priority or according to the text they have.
   - Finally, select the check boxes of the task types you want to list.

6. Click **OK** to confirm your changes and close the dialog box.

   The task list shows only the tasks that confirm to the new filter options.



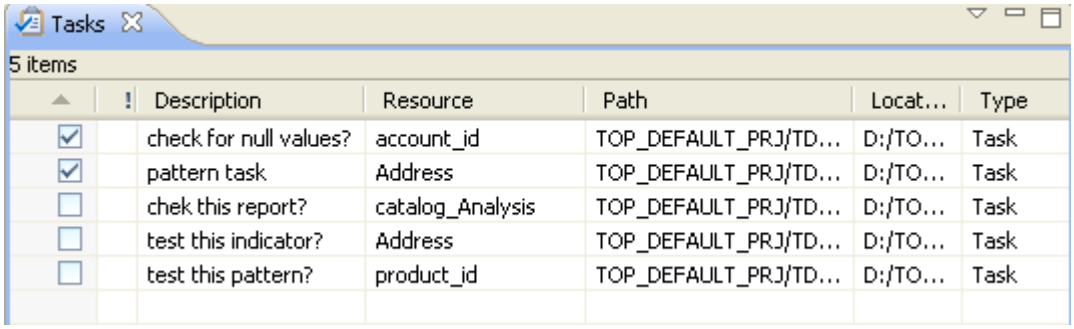
11.2.6. Deleting a completed task

When a task goal is met, you can delete this task from the **Tasks** list after labeling it as completed.

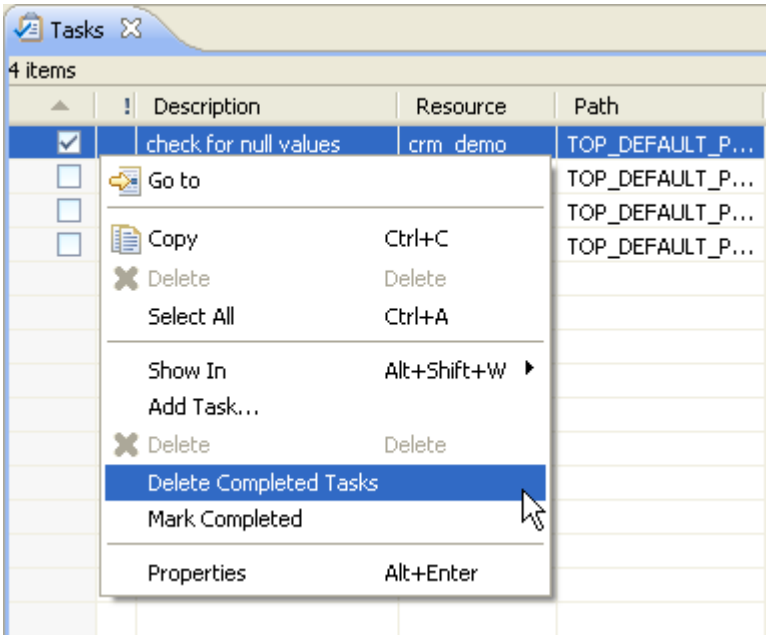
Prerequisite(s): At least one task is added to an item in the **Profiling** perspective of the studio.

To delete a completed task, do the following:

1. Follow the steps outlined in section *Displaying the task list* to access the **Tasks** list.



2. Select the check boxes next to each of the tasks and right-click anywhere in the list.



3. From the contextual menu, select **Delete Completed Tasks**. A confirmation message is displayed to validate the operation.

4. Click **OK** to close the confirmation message.

All tasks marked as completed are deleted from the **Tasks** list.

Appendix A. The studio management GUI

This appendix describes the Graphical User Interfaces (GUI) of the studio.

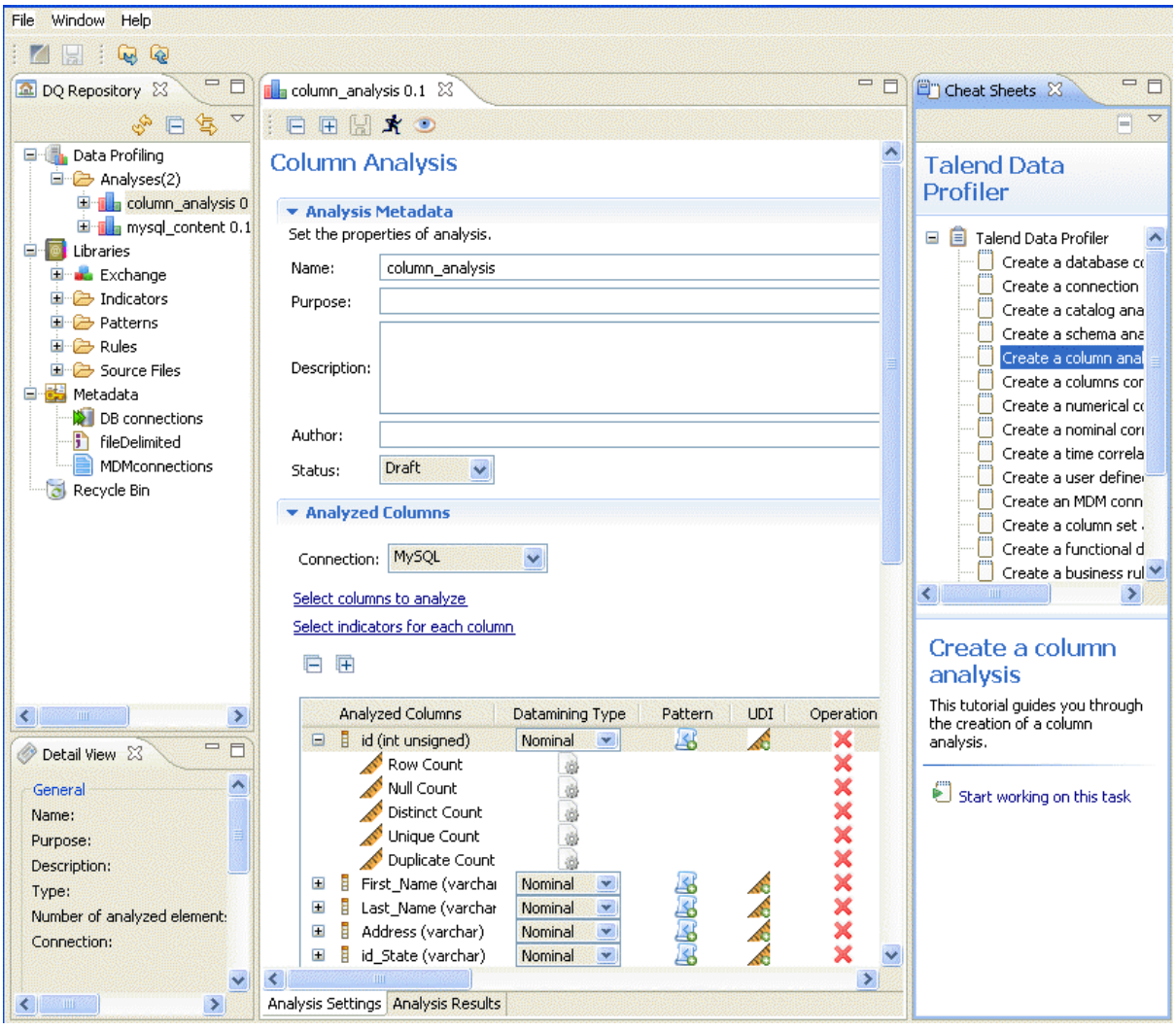
A.1. Main window

The studio main window is the interface from which you manage data profiling.

The main window is divided into:

- the menu bar,
- the toolbar,
- the tree view area,
- a detailed view
- the workspace,
- a tab panel (specific to the Column Analysis editors),
- a cheat sheet view.

The figure below illustrates the main window and its possible views.



The following sections give detailed information about each of the above views.

A.2. Menu bar

The menu bar headers and submenus help you perform operations on your enterprise data.

Table 1 describes menus and menu items available to you.

Table A.1. Table 1—Management menus

Menu	Menu item	Description
File	**Close**	Closes the current open editor in the workspace
	Close All	Closes all open editors in the workspace
	Save	Unavailable option.
	Save All	Unavailable option.
	Exit	Closes the studio main window
	Open File...	Opens a file
Window	**Perspective**	**Profiling:** Opens the data profiler perspective **Data Explorer:** Opens the data explorer perspective **Other...:** Opens a dialog box where you can select any of the available perspectives
	Show View...	Opens the **[Show View]** dialog box which enables you to display different views in the studio
	Preferences	Opens the **[Preferences]** window which enables you to set your preferences
	Reset Perspective...	Resets the current perspective to its default view after confirmation
Help	**Welcome**	Opens a welcoming page which has links to the user documentation and **Talend** practical sites
	Help Contents	Opens the Eclipse help system documentation
	About *Talend studio*	Displays: -the software version you are using -detailed information on your software configuration that may be useful if there is a problem -detailed information about plug-in(s) -detailed information about the studio features
	Cheat Sheets...	Displays a dialog box where you can select a cheat sheet to open
	Software Updates	**Find and Install...:** Opens the **[Install/Update]** wizard that helps searching for updates for the currently installed features, or searching for new features to install **Manage Configuration...:** Opens the **[Product Configuration]** window where you can manage the studio configuration
	View bookmarks	Opens a bookmarks panel that holds few useful links. These links enable you to easily access specific information related to the usage of the studio and/or its database management system
	Key Assist...	Opens a list of all short-cut keys

A.3. Toolbar

The toolbar contains icons that provide you with quick access to the commonly used operations you can perform from the studio main window.

Table 2 describes the toolbar icons and their functions.

Table A.2. Table 2—Management toolbar

Icon	Function
	Saves modifications
	Import data quality items
	Export data quality items
	Switches to data explorer

A.4. Tree view

The **DQ Repository** tree view of the studio shows folders for data profiling analyses, patterns and metadata.

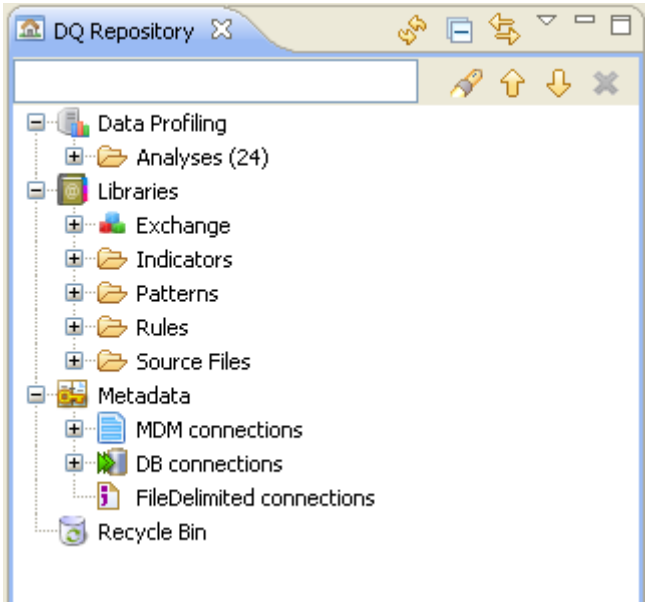
When expanding the **Data profiling** folder in the tree view, you display the created analyses (either executed or not executed yet).

When expanding the **Libraries** folder in the tree view list, you display the list of the pre-defined patterns and SQL patterns. Imported patterns and patterns created by you will also show under the **Patterns** folder.

Under **Libraries** as well, you have all created SQL business rules and all imported patterns from **Talend Exchange**.

When expanding the **Metadata** folder in the tree view list, you display the list of all created DB connections.

The figure below shows an example of an expanded **DQ Repository** tree view.

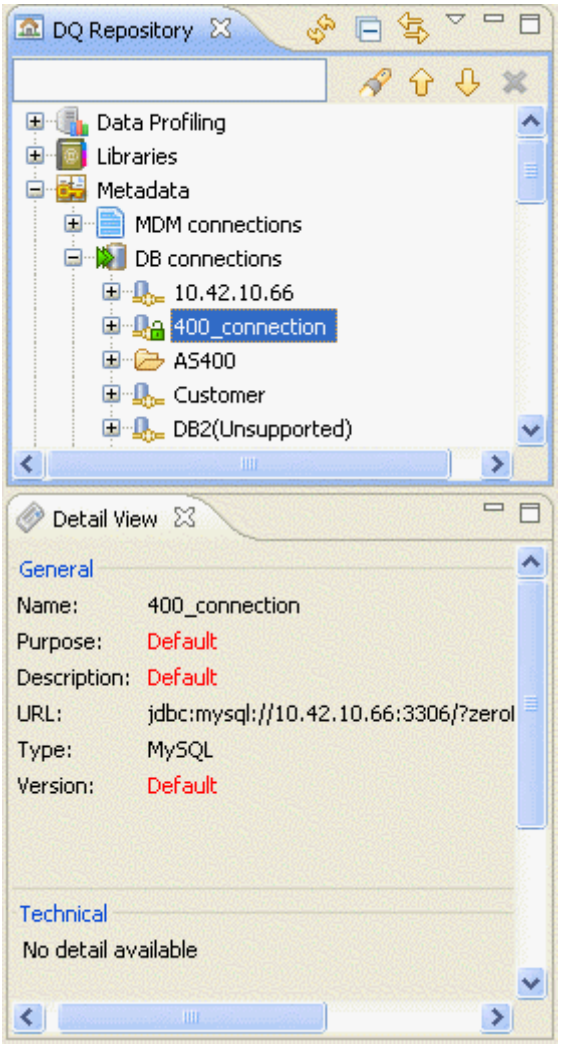


You can use the local toolbar icons to manage the display of the **DQ Repository** tree view.

A.5. Detailed View

This view is located below the **DQ Repository** tree view of the studio. It displays detailed information about the selected element in the tree view area.

The figure below shows an example of the detailed view of the selected DB connection.



You can use the local toolbar icons to manage the display of Detail View.

A.6. The Profiling perspective of the studio

This perspective contains:

- nothing if no analysis, pattern or DB connection is open,
- the parameter values of the open analysis, pattern or DB connection.

When you open a column analysis, a pattern or a DB connection through the tree view area, the relevant editor opens in the studio workspace.

You can use the local toolbar icons to manage the display of the workspace.

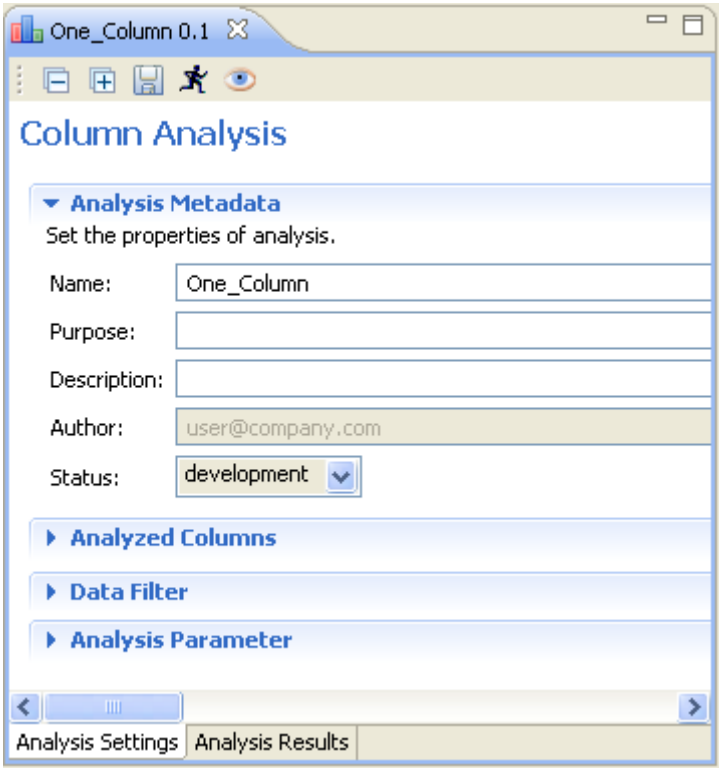
A.7. Tab panel of the analysis editors

This management tab panel is located at the bottom of the analysis editors. It contains a pair of tabs:

- **Analysis Settings**,
- **Analysis Results**.

The **Analysis Settings** tab lists the settings for the current analysis in the currenty editor.

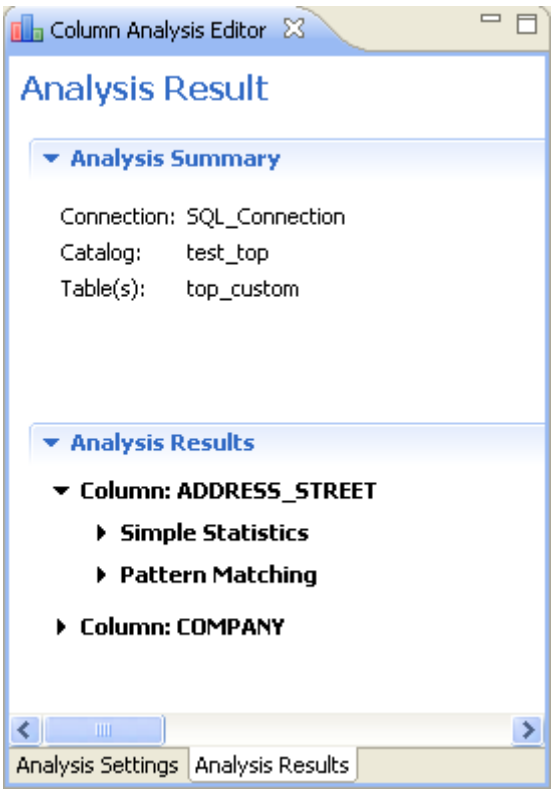
The figure below is an example of the parameters of a column analysis.



The **Analysis Results** tab lists:

- a summary of the executed analysis in the **Analysis Summary** view in which it specifies the connection, the database and the table names for the current analysis,
- the results of the executed analysis, graphics and tables, in the **Analysis Results** view.

The figure below is an example of a column analysis results.



In the **Analysis Results** view, you can:

- click the arrow located next to a column name to display the types of analyses done on that column,
- select a type of analysis to display the corresponding generated graphics and tables.

A.8. Selecting a task from the studio management GUI

You have several ways to select a task from the *Talend Open Studio for Data Quality* main window. You can, for example, use:

- a menu - submenu combination, or
- a toolbar icon, or
- a right-click list, or
- shortcut keys.

Example 1: To show a view in the *Talend Open Studio for Data Quality* main window, either:

- use the **Window > Show View...** menu - submenu combination, or,
- use the **Alt+Shift+Q, Q** shortcut key.

Example 2: To execute an analysis, do one of the followings:

- use the run icon on the toolbar, or
- right-click the analysis you want to execute and select **Run** from the contextual menu, or

- click the **Run** button at the bottom of the editor, or
- use the **F6** shortcut key.

Appendix B. Data Explorer management GUI

The data explorer embedded in the studio allows you to query and browse databases.

This appendix introduces the Graphical User Interfaces (GUI) of the data explorer which is based on the SQL Explorer for which you can find documentation at http://www.sqlexplorer.org/.

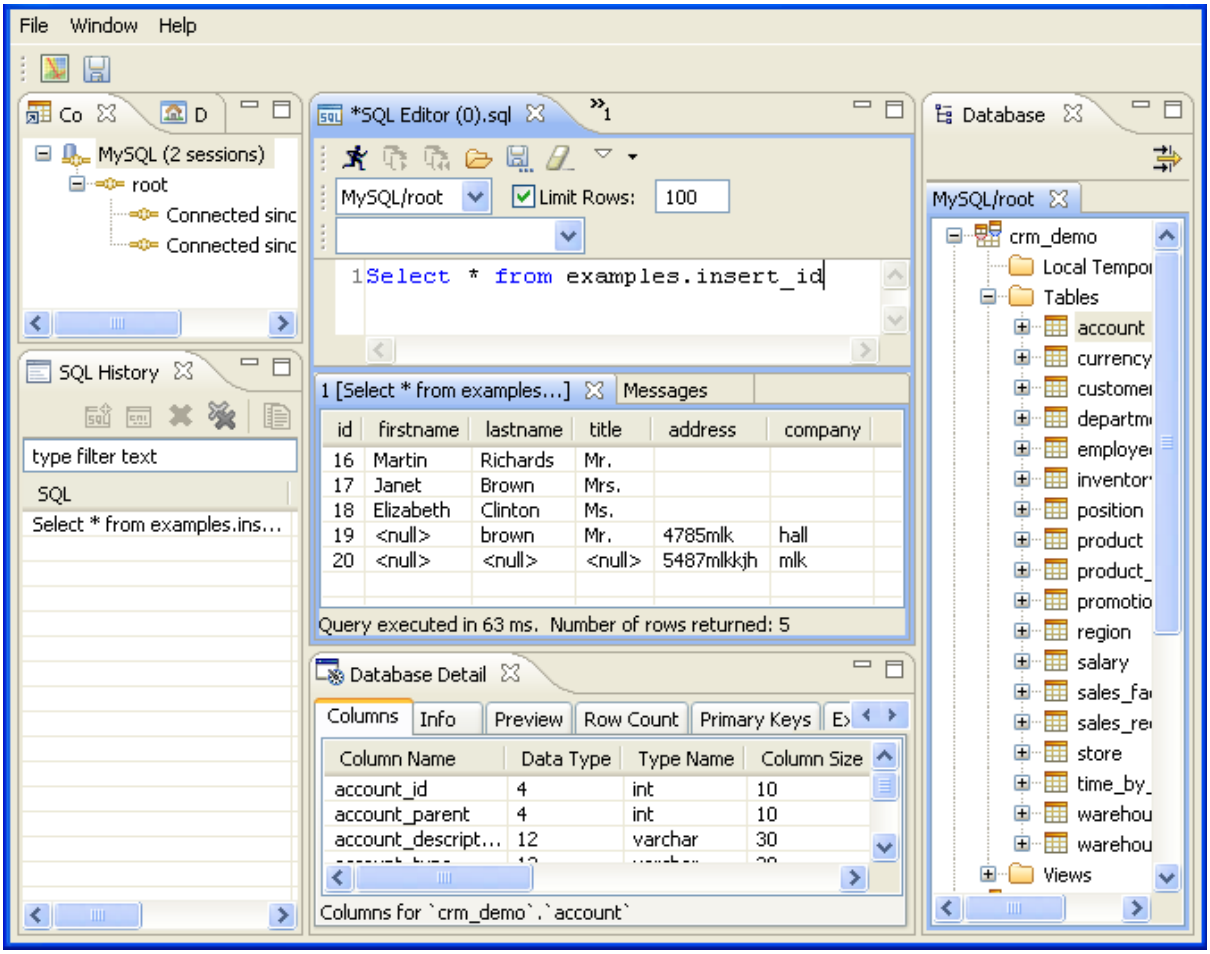
B.1. Main window of the data explorer

The main window of the data explorer is the interface from which you manage your database.

The data explorer main window is divided into:

- menu bar,
- toolbar,
- Connections view,
- SQL History view
- SQL editor view,
- Database Detail view,
- Database Structure view.

The figure below illustrates an example of the data explorer main window and its components.



The following sections give detailed information about each of the above components.

B.2. Menu bar of the data explorer

The menu bar headers and submenus help you perform operations on your enterprise data.

Table A.1, “Table 1—Management menus” of Appendix A describes menus and menu items available to you.

B.3. Toolbar of the data explorer

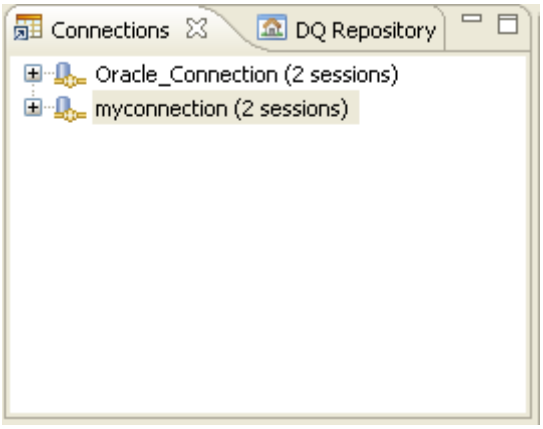
The toolbar contains icons that provide you with quick access to the commonly used operations you can perform from the data explorer main window.

Table A.2, “Table 2—Management toolbar” of Appendix A describes the toolbar icons and their functions.

B.4. Connections view

The **Connections** view shows all the connection profiles that you have set up.

The figure below shows an example of the Connections view.



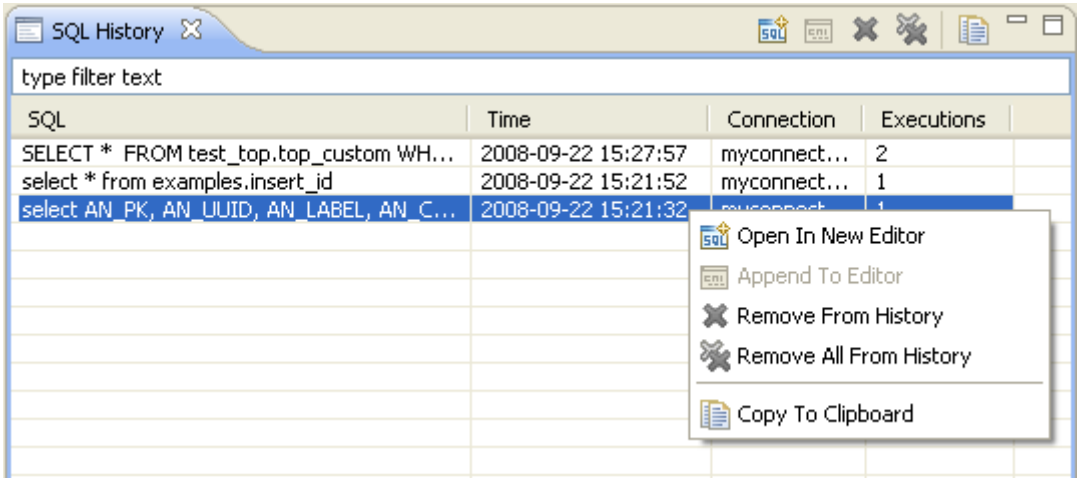
You can use the local toolbar icons to manage the display of the **Connections** view.

B.5. SQL History view

This view shows below the Connections view area. Every statement that was successfully executed is logged in the SQL History view.

The view shows the statement, the date and time when the statement was last executed, which connection was used and how many times the statement has been executed. The SQL statements can be filtered, sorted, removed and opened in or appended to the **[SQL Editor]**.

The figure below shows an example of the SQL History view.



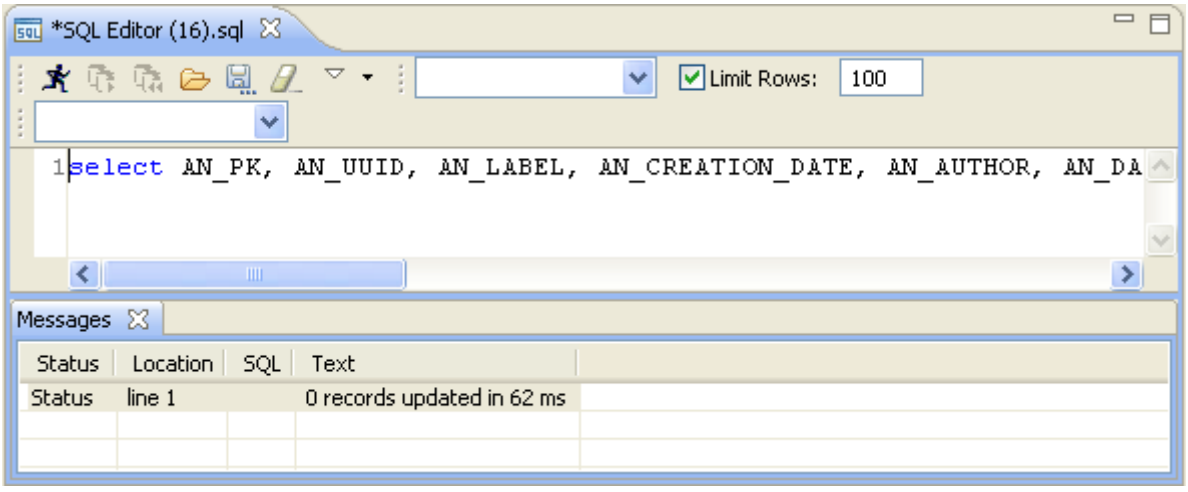
You can use the local toolbar icons to manage the display of SQL History View.

B.6. SQL editor view

This area contains nothing if no **[SQL Editor]** is open. The **[SQL Editor]** provides the following features:

- Executing queries using the CTRL-ENTER combination,
- Basic syntax coloring
- Basic Content Assist
- Overriding result limit
- Word wrapping (if enabled in preferences)
- Session/Catalog/Schema switching
- Loading/Saving SQL scripts
- Commit/Rollback buttons (if session is not in auto-commit mode)
- Display of query execution time of last run query

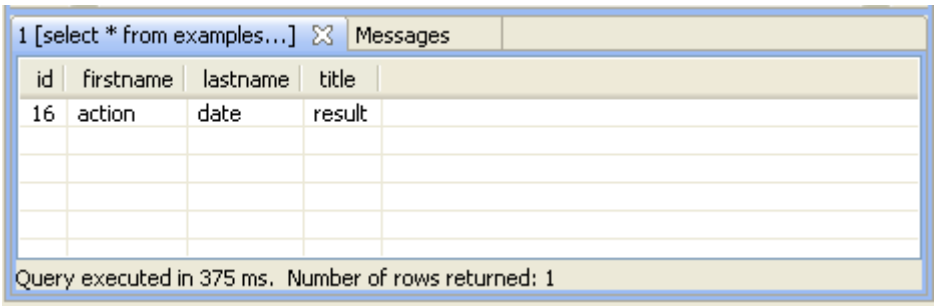
The figure below shows an example of the **[SQL Editor]** view.



The lower part of the **[SQL Editor]** view, the Messages area, detailed information about your data exploring actions. When you execute a query in the SQL query editor, the **Messages** area displays the query results.

You can save all the queries you execute in the data explorer under **Libraries** > **Source Files** in the **DQ Repository** tree view in the studio.

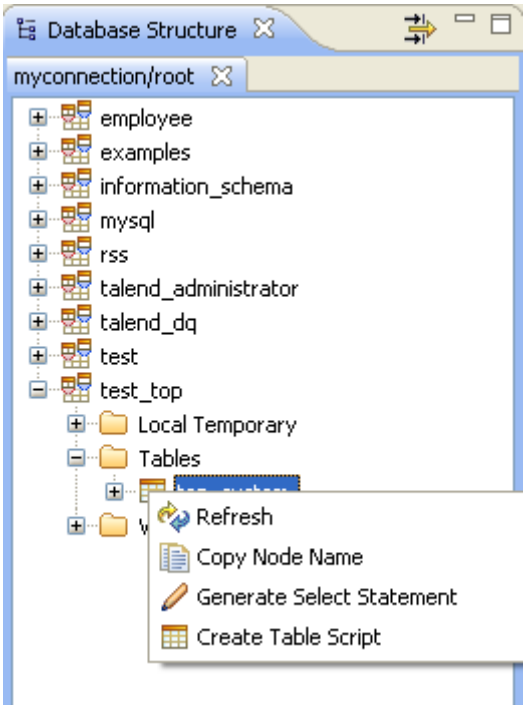
The figure below shows an example of the Messages area.



B.7. Database Structure view

Using the **Database Structure** view, you can explore multiple databases simultaneously.

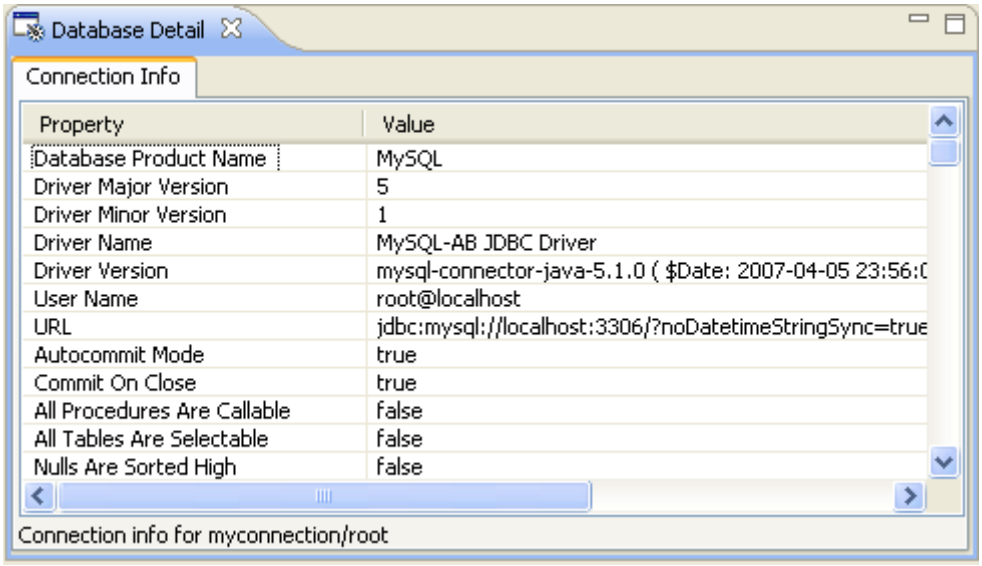
When you select a node in the Database Structure view, the corresponding detail is shown in the Database Detail view. For more information, see section *Database Detail view*. If the detailed view is not active, double-clicking the node will bring the detail view to the front.



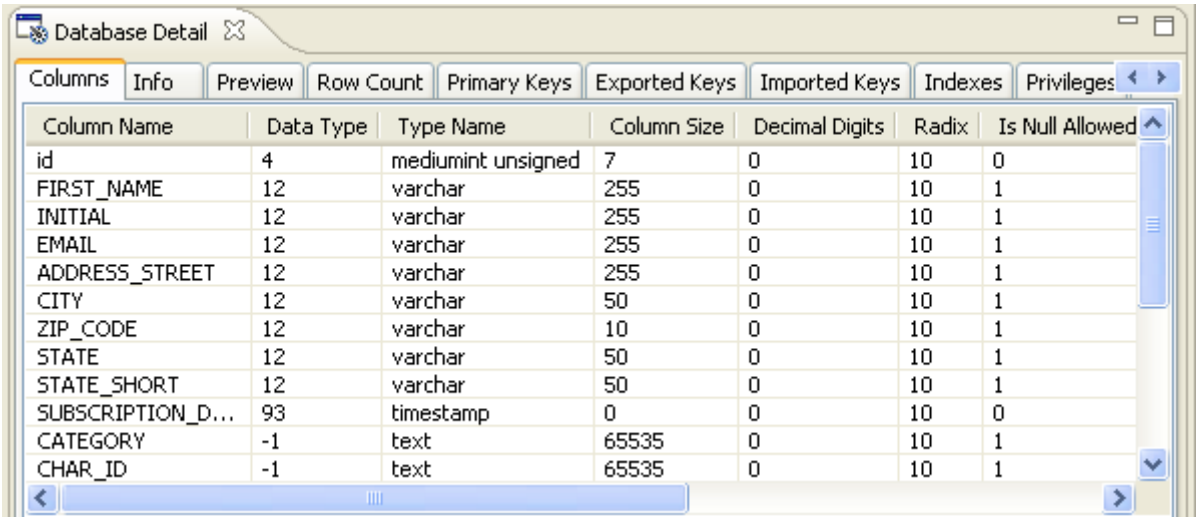
B.8. Database Detail view

Database Detail view shows detailed information for whatever node you select in the **Database Structure** view. What is displayed will depend on the database type that you are using.

When you select a database node in the **Database Structure** view, the **Database Detail** view will show you the connection information as shown in the figure below.



When you select a specific table in the database connection in the **Database Structure** view, the **Database Detail** view shows you detail information about the selected table including **Exported Keys** and **Imported Keys**.



The **Imported Keys** column shows how the table references other tables based on primary and foreign key declarations.

The **Exported Keys** column shows how other tables reference the selected table based on primary and foreign key declarations.

Appendix C. Regular expressions on SQL Server

This appendix describes in detail how to create a regular expression function on SQL Server databases.

C.1. Main concept

The regular expression function is not built into all different databases environments. This is why you need, when using some databases, to create a User-Defined Function (UDF) to extend the functionality of the database server.

For example, the following databases natively support regular expressions: MySQL, PostgreSQL, Oracle 10g, Ingres, etc., while Microsoft SQL server does not.

After you create the regular expression function, you should use the studio to declare that function in a specific database before being able to use regular expressions on analyzed columns.

For more information on how to declare a regular expression function in the studio, see section *How to define a query template for a specific database* and section *How to declare a User-Defined Function in a specific database*.

C.2. How to create a regular expression function on SQL Server

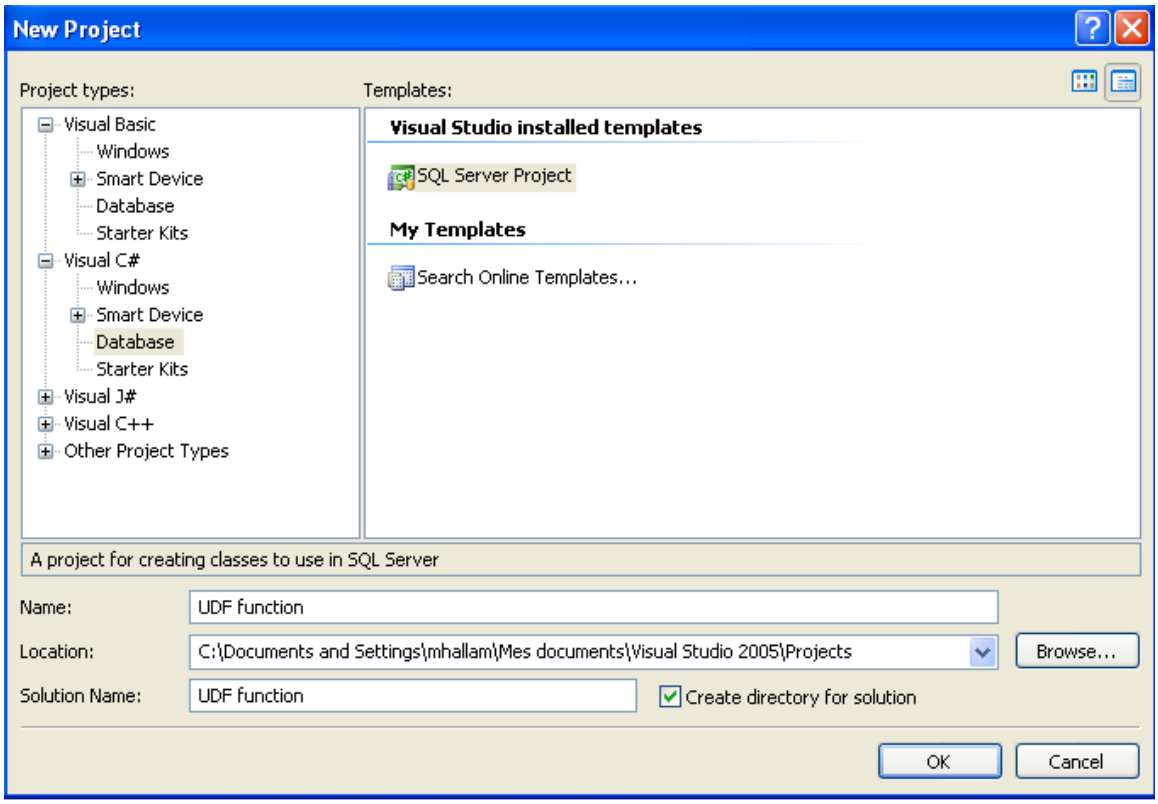
Prerequisite(s): You should have Visual Studio 2005 or 2008. The Visual Studio main window is open.

To create a regular expression function in SQL Server, follow the steps outlined in the sections below.

C.2.1. How to create a project in Visual Studio

You must start by creating an SQL server database project. To do that:

1. On the menu bar, select **File > New > Project** to open the **[New Project]** window.

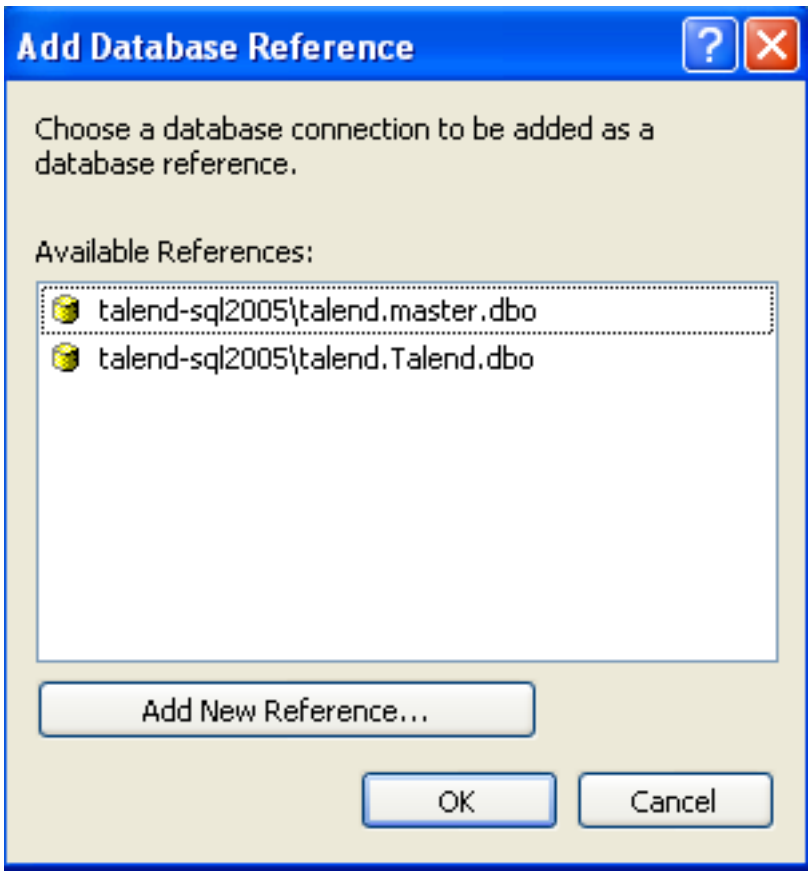


2. In the **Project types** tree view, expand **Visual C#** and select **Database**.

3. In the **Templates** area to the right, select **SQL Server Project** and then enter a name in the **Name** field for the project you want to create, *UDF function* in this example.

4. Click **OK** to validate your changes and close the window.

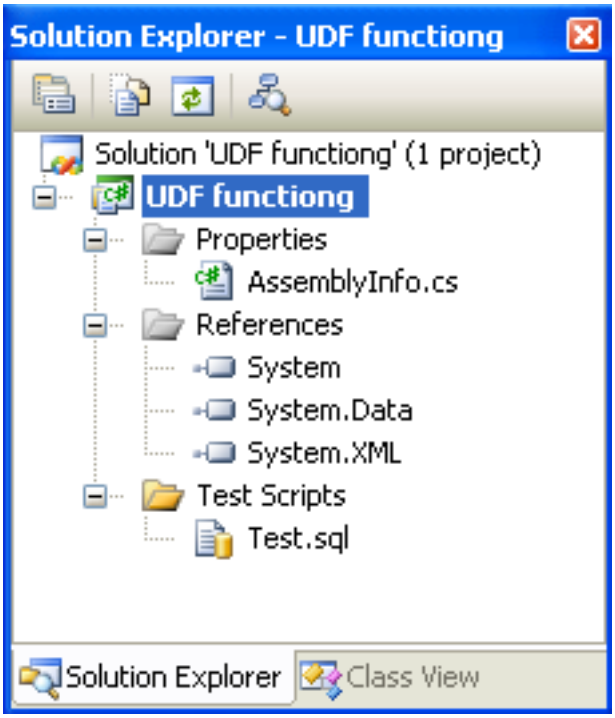
   The **[Add Database Reference]** dialog box is displayed.



5. From the **Available References** list, select the database in which you want to create the project and then click **OK** to close the dialog box.

   If the database you want to create the project in is not listed, you can add it to the **Available Reference** list through the **Add New Reference** tab.

   The project is created and listed in the **Solution Explorer** panel to the right of the Visual Studio main window.

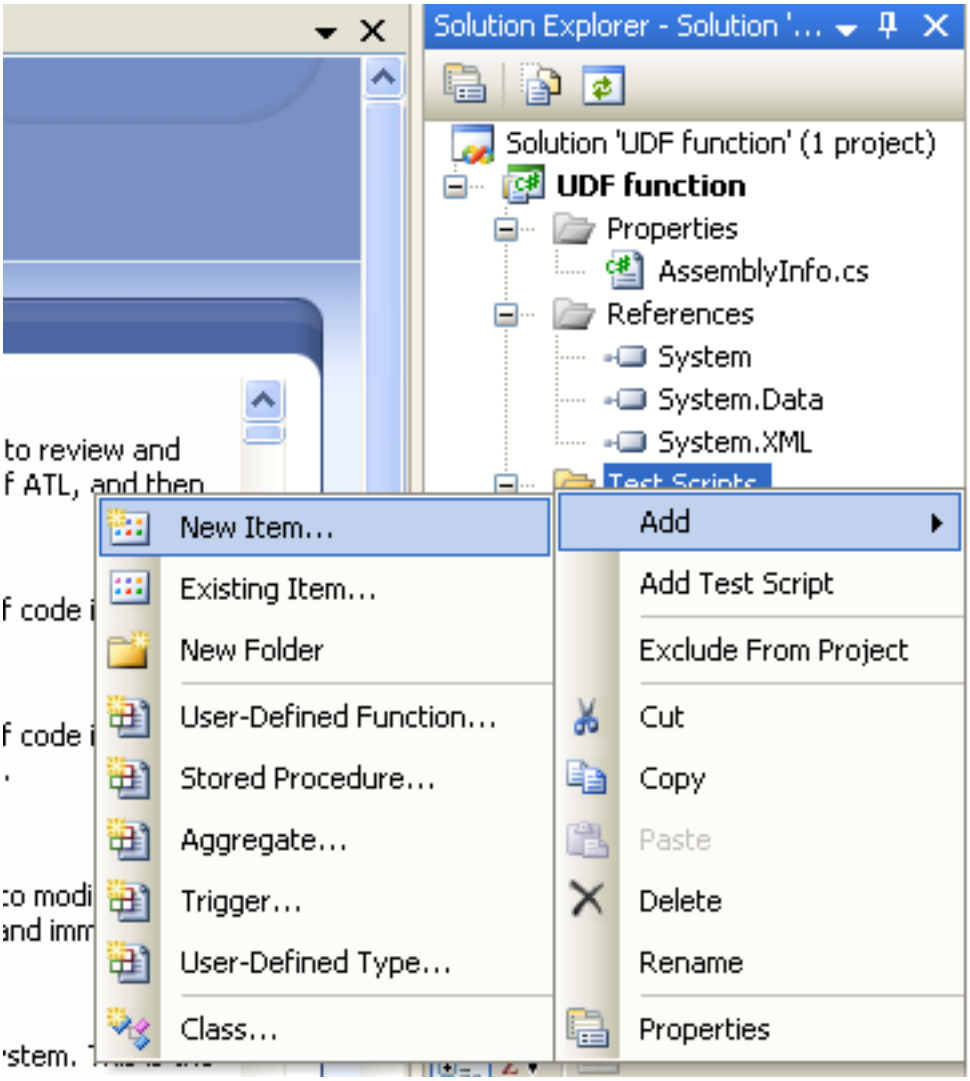


C.2.2. How to deploy the regular expression function to the SQL server

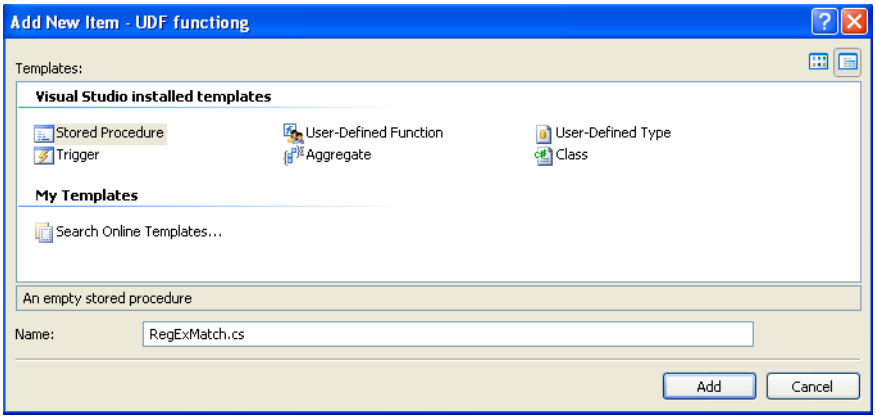
You need now to add the new regular expression function to the created project, and then deploy the function to the SQl server. To do that:

1. In the project list in the **Solution Explorer** panel, expand the node of the project you created and right-click the **Test Scripts** node.

2. From the contextual menu, select **Add > New Item...**.



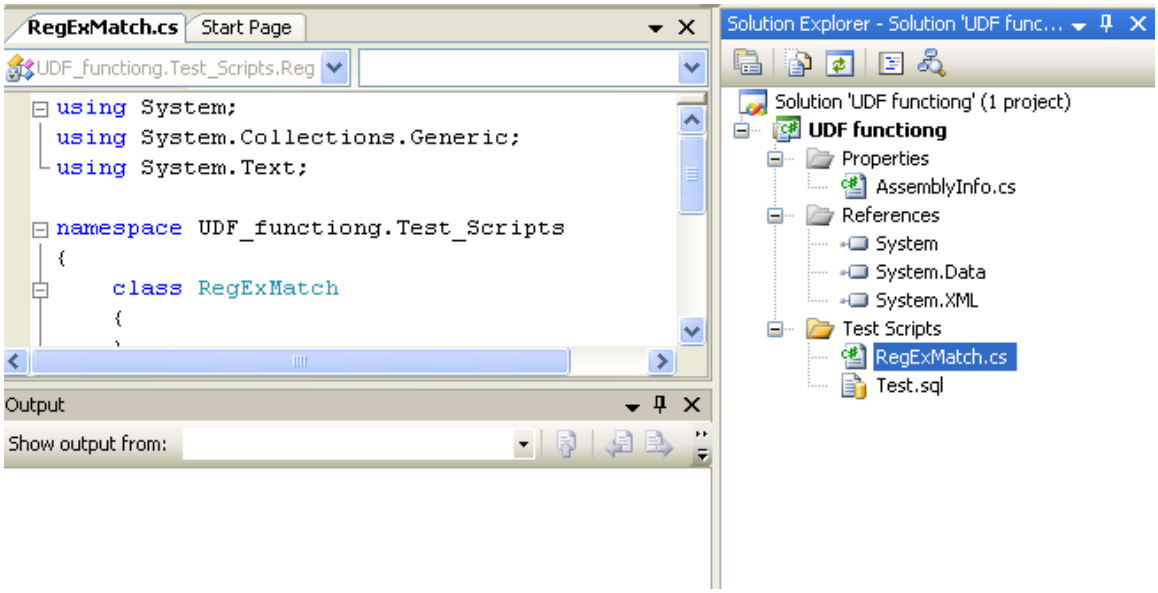
The **[Add New Item]** dialog box is displayed.



3. From the **Templates** list, select **Class** and then in the **Name** field, enter a name to the user-defined function you want to add to the project, *RegExMatch* in this example.

   The added function is listed under the created project node in the **Solution Explorer** panel to the right.

4. Click **Add** to validate your changes and close the dialog box.

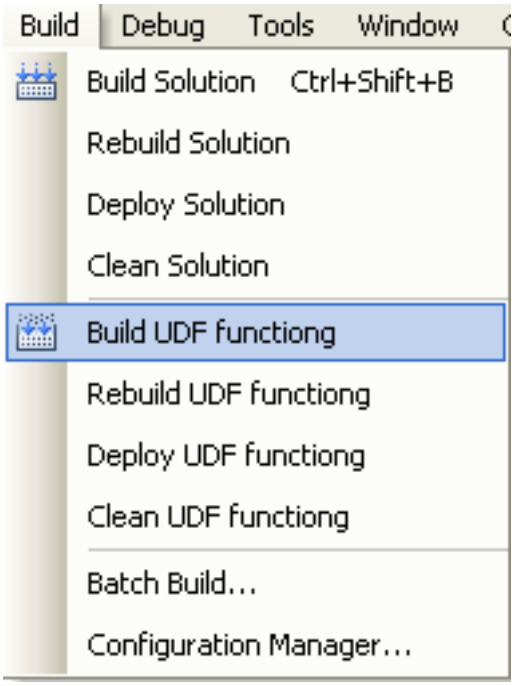


5. In the code space to the left, enter the instructions corresponding to the regular expression function you already added to the created project.

   Below is the code for the regular expression function we use in this example.

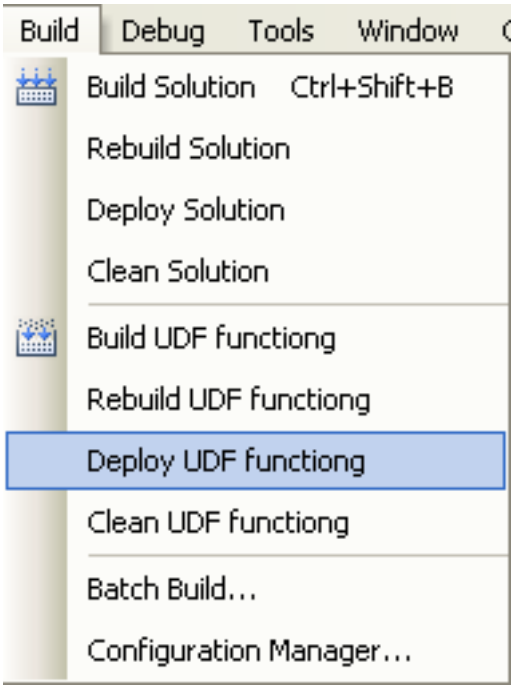
```
Using System;
Using Microsoft.SqlServer.Server;
Using System.Text.RegularExpressions;
Public partial class RegExBase
{
     [SqlFunction(IsDeterministic = true, IsPrecise = true)]
     Public static int RegExMatch( string matchString , string pattern)
     {
            Regex r1 = new Regex(pattern.TrimEnd(null));
            if (r1.Match(matchString.TrimEnd(null)).Success == true)
            {
                 return 1 ;
            }
            else
            {
                 return 0 ;
            }
Using
     }
};
```

6. Press **Ctrl+S** to save your changes and then on the menu bar, click **Build** and in the contextual menu select the corresponding item to build the project you created, **Build** *UDF function* in this example.

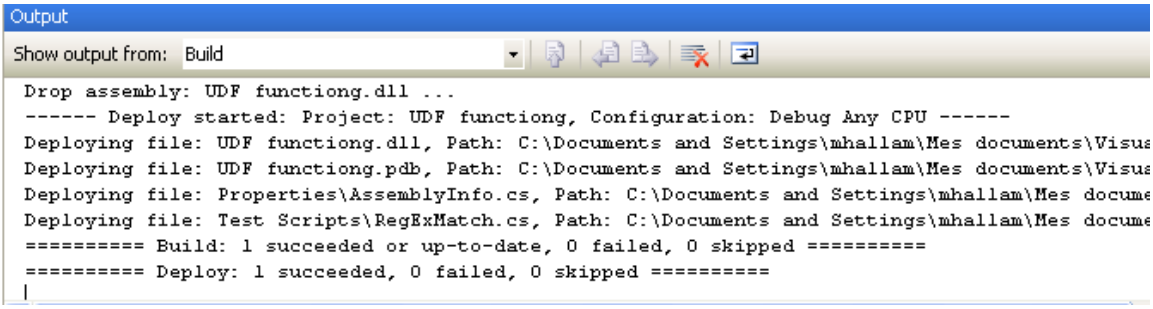


The lower pane of the window displays a message to confirm that the “build” operation was successful or not.

7. On the menu bar, click **Build** and in the contextual menu select the corresponding item to deploy the project you created, **Deploy** *UDF function* in this example.



The lower pane of the window displays a message to confirm that the “deploy” operation was successful, or not.



If required:

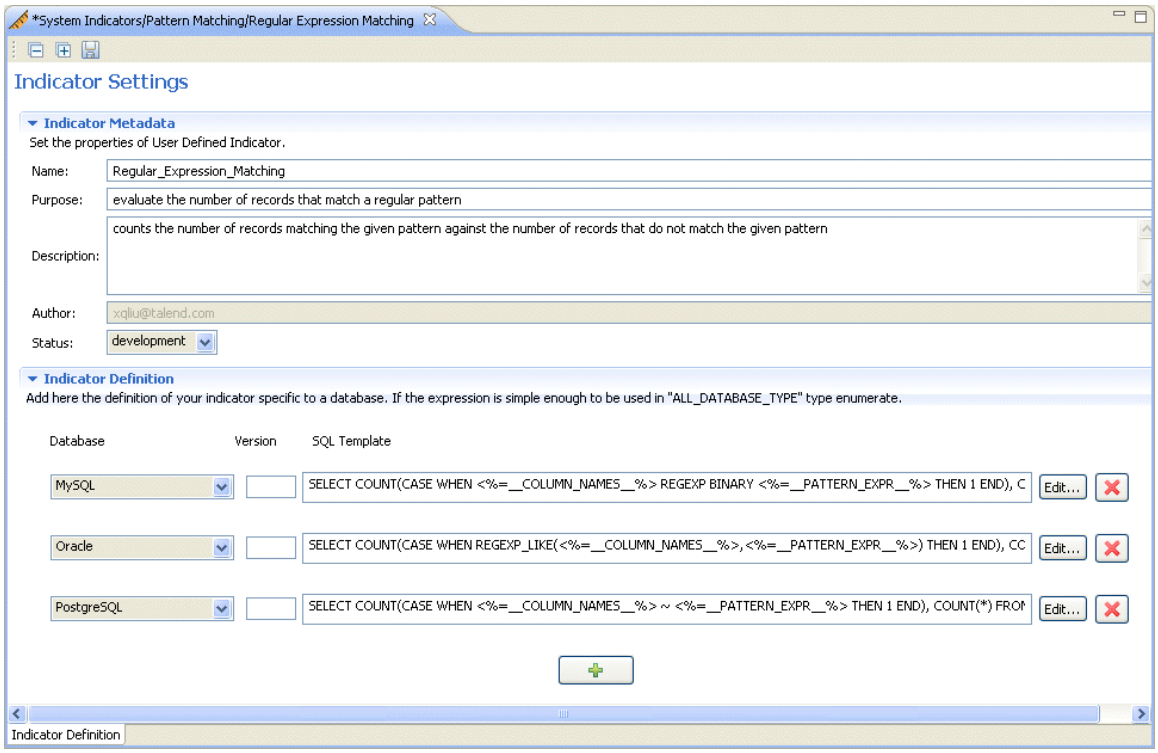
1. launch SQL Server and check if the created function exists in the function list,
2. check if the function works well, for more information, see section *How to test the created function via the SQL Server editor*.

C.2.3. How to set up the studio

Before being able to use regular expressions on analyzed columns in a database, you must first declare the created regular expression function, *RegExMatch* in this example, in the specified database via the studio. To do that:

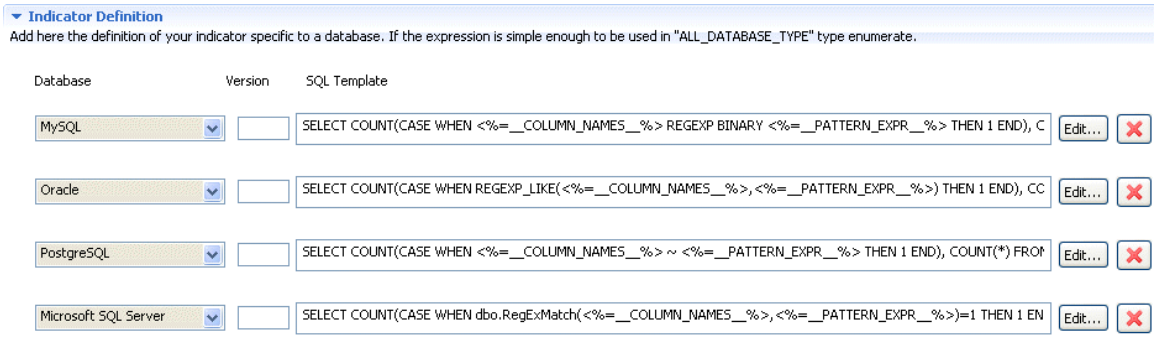
1. In the **DQ Repository** tree view, expand **Libraries > Indicators**.
2. Expand **System Indicators > Pattern Matching**.
3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

   The corresponding view displays the indicator metadata and its definition.



You need now to add to the list of databases the database for which you want to define a query template. This query template will compute the regular expression matching.

4. Click the **[+]** button at the bottom of the **Indicator Definition** view to add a field for the new template.

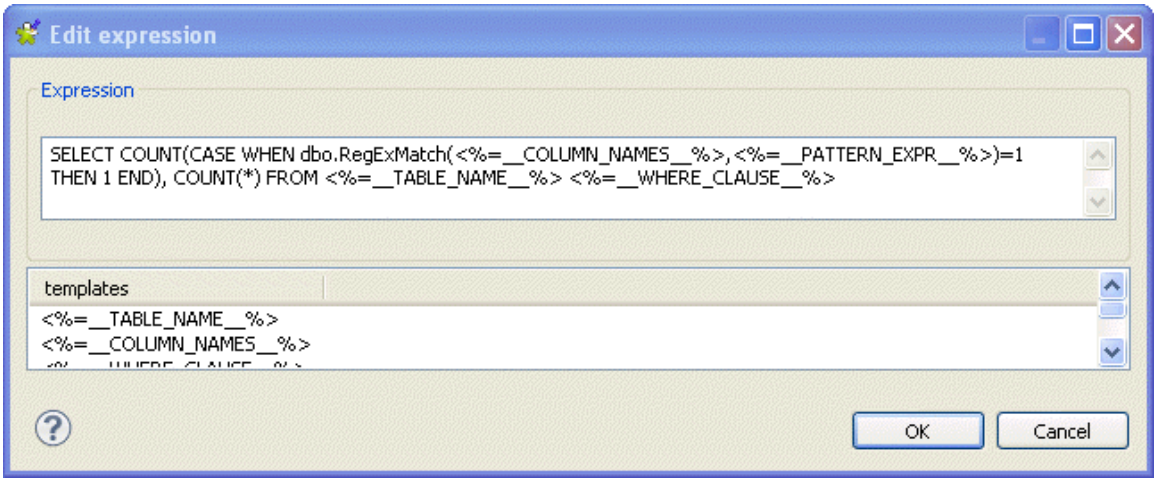


5. In the new field, click the arrow and select the database for which you want to define the template, **Microsoft SQL Server**.

6. Copy the indicator definition of any of the other databases.

7. Click the **Edit...** button next to the new field.

   The **[Edit expression]** dialog box is displayed.



8. Paste the indicator definition (template) in the **Expression** box and then modify the text after `WHEN` in order to adapt the template to the selected database.

9. Click **OK** to proceed to the next step. The new template is displayed in the field.

10. Click the save icon on top of the editor to save your changes.

For more detailed information on how to declare a regular expression function in the studio, see section *How to define a query template for a specific database* and section *How to declare a User-Defined Function in a specific database*.

C.3. How to test the created function via the SQL Server editor

- To test the created function via the SQL server editor, copy the below code and execute it:

```
create table Contacts (
FirstName nvarchar(30),
```

```
 LastName nvarchar(30),
EmailAddress nvarchar(30) CHECK
 (dbo.RegExMatch('[a-zA-Z0-9_\-]+@([a-zA-Z0-9_\-]+\.)
 +(com|org|edu|nz)',
EmailAddress)=1),
USPhoneNo nvarchar(30) CHECK
 (dbo.RegExMatch('\([1-9][0-9][0-9]\) [0-9][0-9][0-9]
\-[0-9][0-9][0-9][0-9]',
UsPhoneNo)=1))

INSERT INTO [talend].[dbo].[Contacts]
                       ([FirstName]
                      , [LastName]
                      , [EmailAddress]
                      , [USPhoneNo])
      VALUES
                     ('Hallam'
                    , 'Amine'
                    , 'mhallam@talend.com'
                    , '0129-2090-1092')
                    , ( 'encoremoi'
                     , ‘nimportequoi'
                     , 'amine@zichji.org'
                     , '(122) 190-9090')
GO
```

- To search for the expression that match, use the following code:

```
SELECT [FirstName]
     , [LastName]
     , [EmailAddress]
     , [USPhoneNo]
  FROM [talend].[dbo].[Contacts]
  where [talend].[dbo].RegExMatch([EmailAddress],
'[a-zA-Z0-9_\-]+@([a-zA-Z0-9_\-]+\.)+(com|org|edu|nz|au)')
= 1
```

- To search for the expression that do not match, use the following code:

```
SELECT [FirstName]
     , [LastName]
     , [EmailAddress]
     , [USPhoneNo]
  FROM [talend].[dbo].[Contacts]
  where [talend].[dbo].RegExMatch([EmailAddress],
'[a-zA-Z0-9_\-]+@([a-zA-Z0-9_\-]+\.)+(com|org|edu|nz|au)')
= 0
```