talend | Connecting the Data-Driven Enterprise

Lab Guide

DQ Essentials

Version 6.3

Talend Inc.
800 Bridge Parkway, Suite 200
Redwood City, CA 94065
United States
+1 (650) 539 3200

Welcome to Talend Training

Congratulations on choosing a Talend training course.

Working through the course

You will develop your skills by working through use cases and practice exercises using live software. Completing the exercises is critical to learning!

If you are following a self-paced, on-demand training (ODT) module, and you need an answer to proceed with a particular exercise, use the help suggestions on your image desktop. If you can't access your image, contact customercare@talend.com.

Exploring

You will be working in actual Talend software, not a simulation. We hope you have fun and get lots of practice using the software! However, if you work on tasks beyond the scope of the training, you could run out of time with the environment, or you could mess up data or Jobs needed for subsequent exercises. We suggest finishing the course first, and if you have remaining time, explore as you wish. Keep in mind that our technical support team can't assist with your exploring beyond the course materials.

For more information

Talend product documentation (help.talend.com)

Talend Community (community.talend.com)

Sharing

This page intentionally left blank to ensure new chapters start on right (odd number) pages.

CONTENTS

LESSON 1

Structural Analysis

This chapter discusses the following.

Structural analysis

Lab overview

Your software company has just merged with a competitor. Combining technologies and data, the new organization will be the industry leader in the American and European markets.

The two companies have not had the same level of commitment to data quality. Your prior company is familiar with Talend Studio and regularly profiles data using available analyses, while your new colleagues did not take the opportunity to actively enhance the quality of their company databases.

Both companies have an online CRM in which people can create customer accounts using a Web form. Your company Web site offered data-entry assistance tools to help people avoid misspelling items like city names, zip codes, and telephone numbers. The other company CRM did not employ these types of tools, so its customer tables are polluted with bad data.

As a member of the merge committee, your role is to improve the other company's CRM database quality to match that of yours.

Before cleansing the bad data, you must closely examine the database and assess the situation. You will do a complete data checkup by using the analyses available in Talend Studio and creating profiling Jobs.

Lesson overview

Talend provides different types of analysis that you can use to manage your data. To complete a particular analysis, you need to connect to the data source. In this course you connect to a MySQL database server. MySQL Server and MySQL Workbench are already installed.

For your convenience, instead of entering new connection information every time you create an analysis, you can store and reuse connection information as metadata.

After starting Talend Studio, you will establish a connection to the local database server. Then you will use a structural analysis to create an overview of the databases stored on the MySQL server, with a spotlight on the CRM database.

Note: Your training environment uses a single virtual machine (VM) that contains all the software you need to complete the exercises, including Talend Studio and MySQL. Although the MySQL connection is "local," remote database servers are also supported.

Objectives

After completing this lesson, you will be able to:

- Start Talend Studio
- Switch to the Profiling perspective
- Store database connection information for reuse
- Run a connection overview analysis on the MySQL server

The first step is to start Talend Studio.

Creating a database connection

Overview

You will be connecting to a MySQL database for most of the analyses in this course. In this lesson, you will store database connection information as metadata that you can reuse in other lessons. Before you create and store the metadata, you will start the Talend software.

Start Talend Studio

1. RUN STUDIO

   To start Studio, on your VM desktop, double-click the **Talend Studio** shortcut.

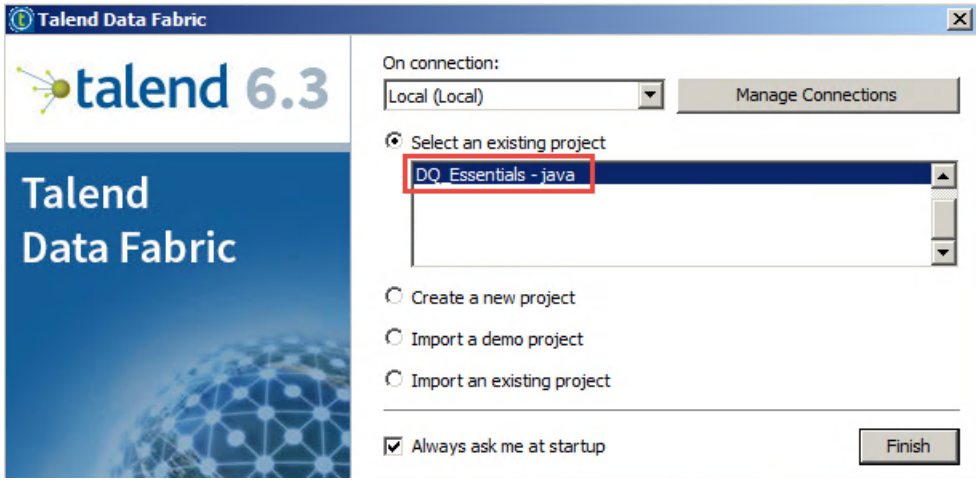
   

   Note: If you ever want to start Studio directly, such as when a shortcut does not exist, it helps to know the location of the binaries. Navigate to **C:\Talend\6.3\studio**. Notice the many executables corresponding to various systems architecture. The training environment is 64-bit Windows, so to start Studio, double-click **Talend-Studio-win-x86_64.exe**. You can create a desktop shortcut at this level.

2. SELECT AN EXISTING PROJECT

   The Talend Data Fabric window opens.

   A project named DQ Essentials has been created for you. Click **Finish**.

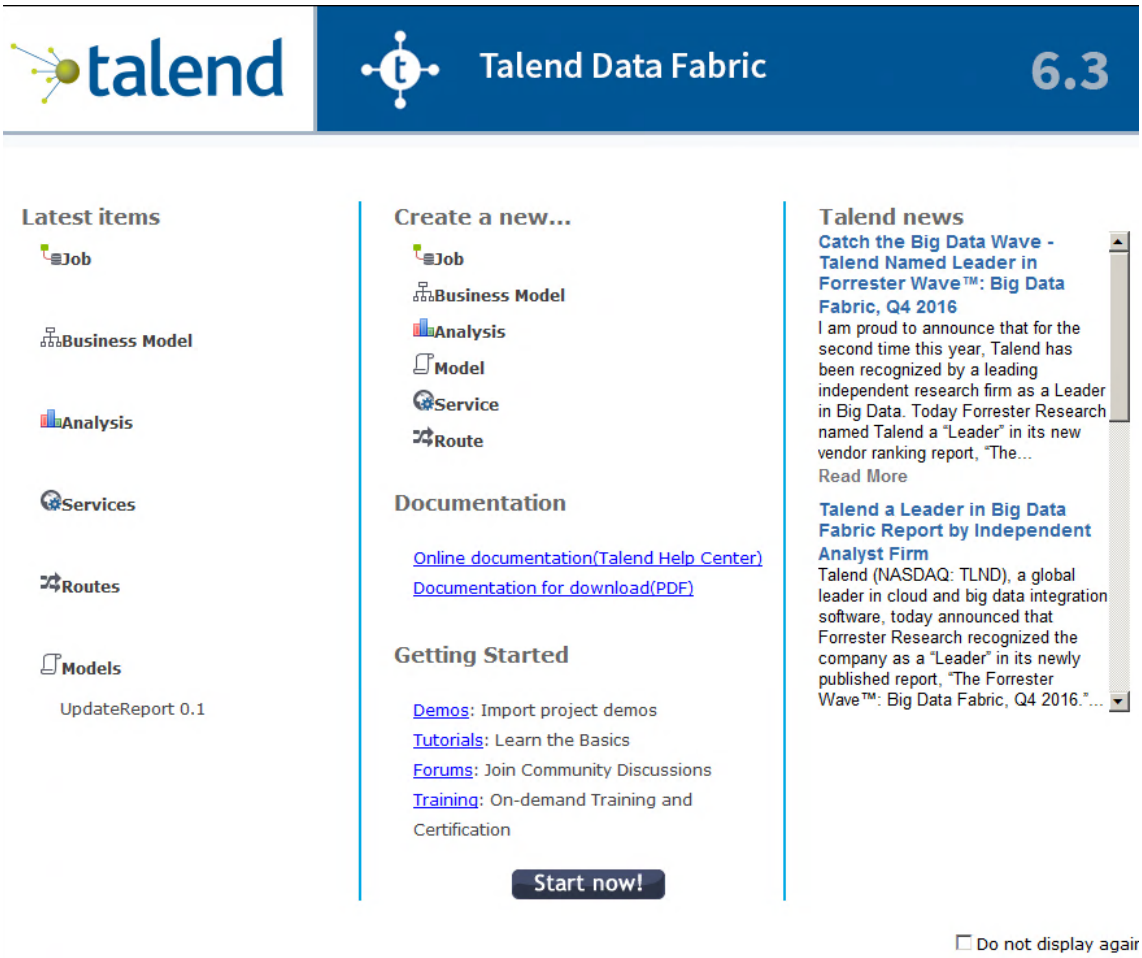
   

   Note: You may have a different version of Studio in your training environment, but you have everything you need for the course.

3. START THE PROJECT

   If you get a Connect to TalendForge window, you can either create an account, connect to an existing account, or skip this step and start Studio. To proceed, click **Skip**.

   The splash screen appears.



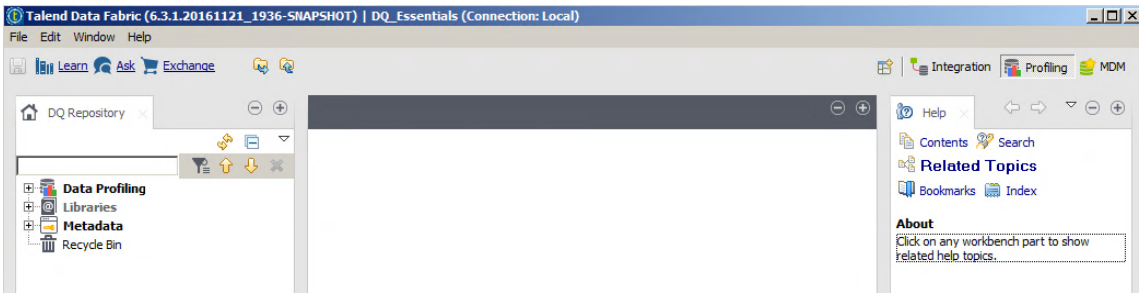
Note: Your screen image may vary depending on the version and type of installation on your VM.

Note: Although you do not use TalendForge in this course, Talend recommends creating an account from your installation environment and becoming an active member of the TalendForge online community, which provides several valuable resources.

4. NAVIGATE IN THE STUDIO MAIN WINDOW

   Click the **Start now!** button (you may need to scroll down).

   The window opens.



Note: Depending on your preferences and available tools, your screen may vary.

You can see the commonly used areas:

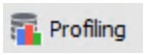
- The DQ Repository, where data analyses and metadata are stored, is in the upper left corner.
- The workspace, where you create and modify Jobs, run analyses, and examine results, is in the center.
- The online help is on the right. Whether you are using Data Quality or Data Integration, the right-side pane also shows a design palette. When building data integration Jobs, you drag and drop components into the workspace.

Create database connection metadata

1. SELECT THE PERSPECTIVE

   The perspective determines the view of visible actions available at any given point. Available perspectives are determined by your installation and license information. For this course, you use the Profiling perspective to set up and run data quality analyses, and you use the Integration perspective to create data profiling Jobs.

   Make sure you are in the Profiling perspective. In the upper part of the window, click the **Profiling** perspective icon:

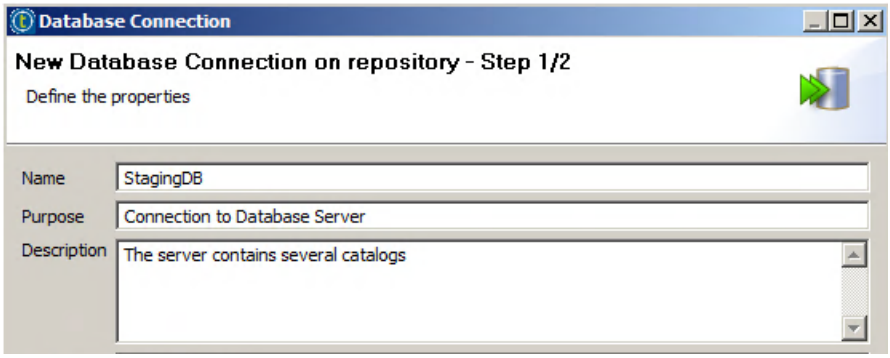
2. CREATE METADATA

   In **DQ Repository**, expand **Metadata**.

   Right-click **DB connections** and click **Create DB Connection.**

   You are creating a connection to a collection of databases already configured in your training environment.

   a. In the **Name** text box, enter *StagingDB*

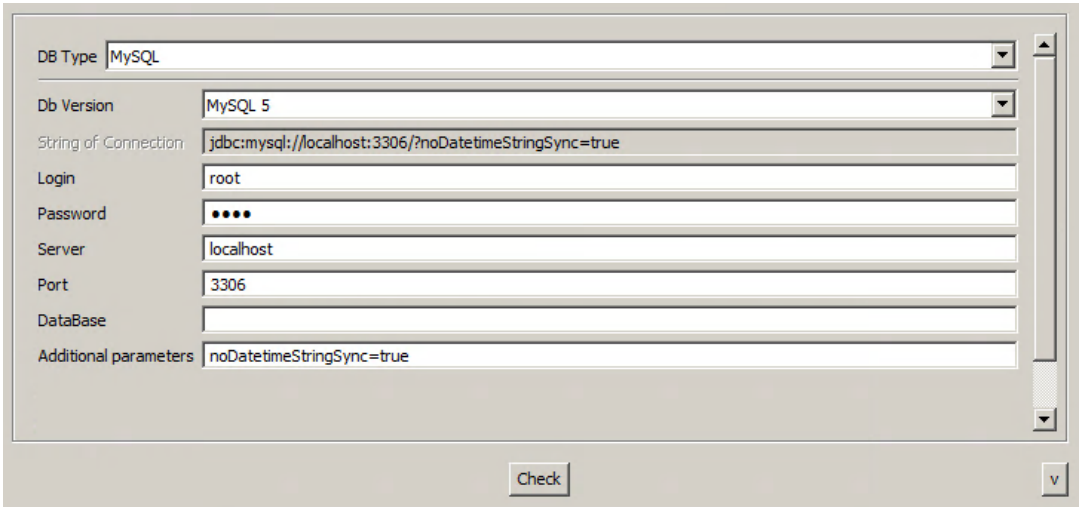
   

   b. In the **Purpose** text box (optional), enter *Connection to Database Server*.

   c. In the **Description** text box (optional), enter *The server contains several catalogs*

   d. Click **Next**.

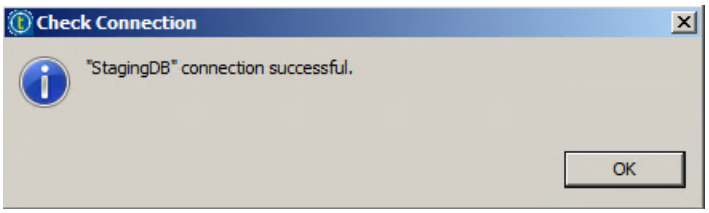
   e. On the **DB Type** list, click **MySQL**. Do not modify the default value in the Db Version box.



f. Enter the settings as described below.

- In the **Login** and **Password** boxes, enter *root*.
- In the **Server** text box, enter *localhost*.
- Leave the **DataBase** box empty (you want to be able to access all databases configured on the local server).

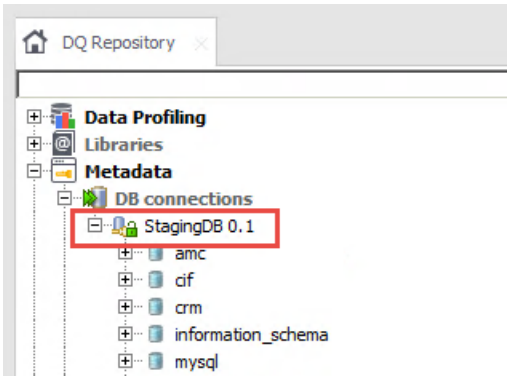
g. To verify your connection information, click the **Check** button.



h. In the Check Connection window, click **OK** or make any necessary corrections and again click **Check**.

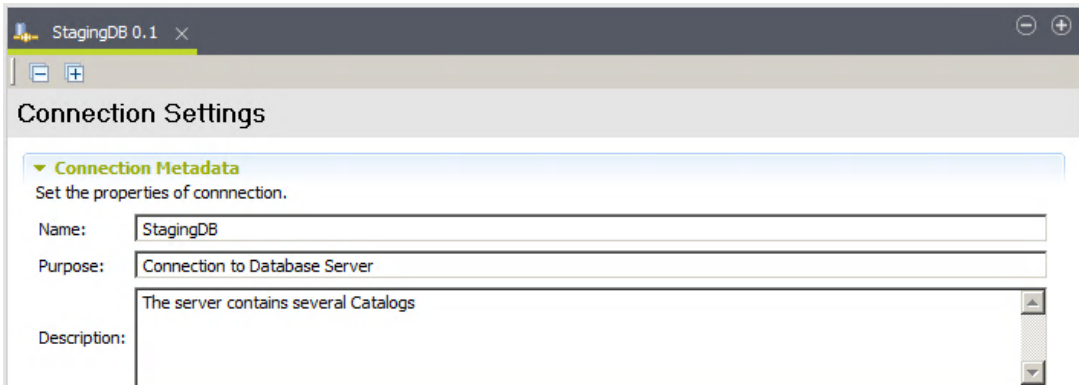
i. Click **Finish**.

Your new database connection, StagingDB, appears in the repository.



It contains all databases and tables configured on the local database server, including the CRM database, which you will use shortly. **Note:** In your course installation, the exact database names may vary.

The workspace displays the configuration information for the connection.



If needed, make changes to the connection settings.

In the next exercise, you will create your first analysis to learn more about the structure of the database server.

Performing structural analyses

Overview

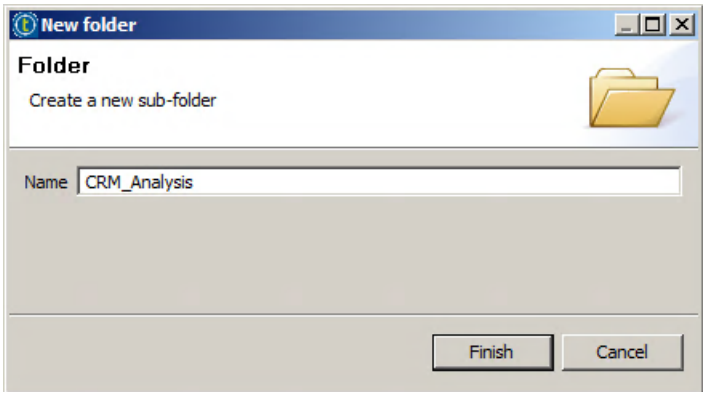
You can profile your data using many types of analysis available in Talend Studio. An analysis from the Structural Analysis category provides basic information about your databases and catalogs, including the number of tables, rows per table, indexes, and primary keys. This overview is often the starting point for a comparison of separate data sources or for making sure data includes the structure needed to meet data quality requirements.

In the following exercise, you will run a connection overview analysis to explore the MySQL server structure. Then you will examine the results, focusing on the CRM database.

Create a connection overview analysis

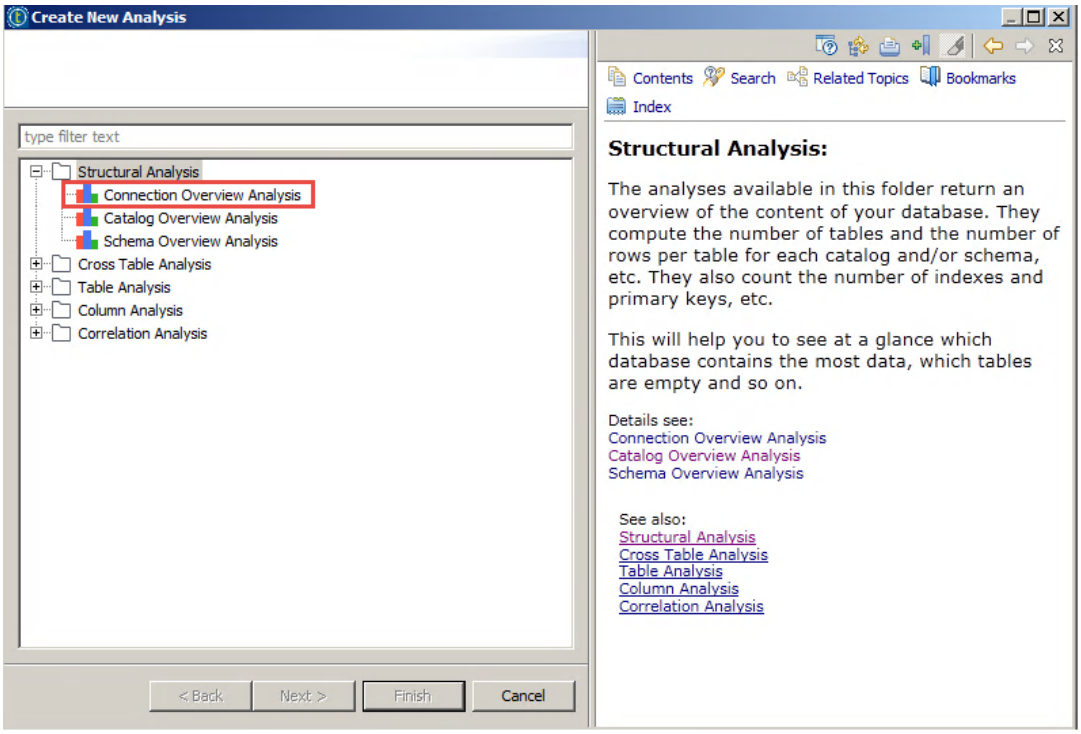
1. CREATE A FOLDER
    a. In **DQ Repository**, expand **Data Profiling**.
    b. To create a folder for your analyses, right-click **Analyses** and select **Create Folder**.

    Name it *CRM_Analysis* and click **Finish**.

    All the analyses that you create in this lab are stored in this folder.
2. CREATE THE ANALYSIS
    a. Right-click the folder and click **New Analysis**.

    The types of analysis are grouped in categories.
    b. Expand **Structural Analysis** and click **Connection Overview Analysis**.



Notice that the right pane displays help information related to the specified analysis. This in-product, context-sensitive help is an easy way to learn more about Talend Studio. Read the background information and click **Next**.

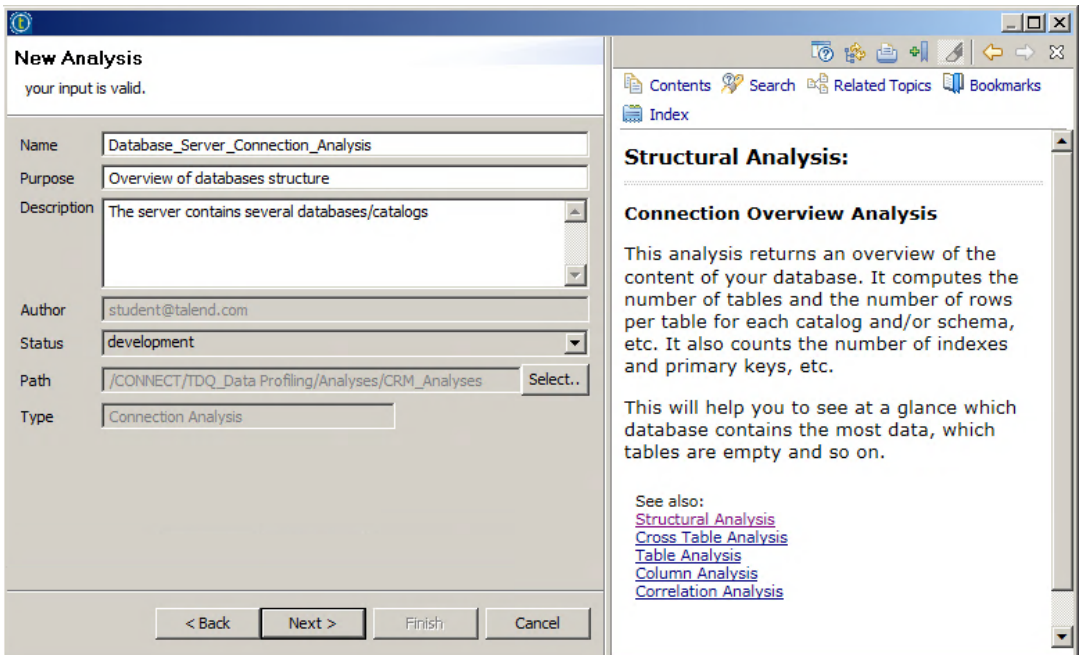
3. SET UP THE ANALYSIS

   a. Now you will identify the analysis.

      In the **Name** text box, enter *Database_Server_Connection_Analysis*

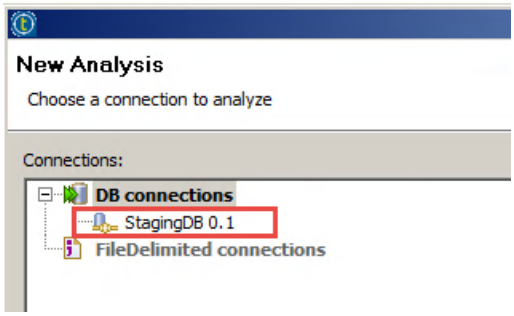
   b. Fill in the **Purpose** and **Description** boxes.

      Note: Get in the habit of filling in these optional text boxes. As with any project, good documentation can be critical for other people to understand an analysis or a Job. If you do not know what to enter, you can copy and paste relevant statements from your in-progress lesson.



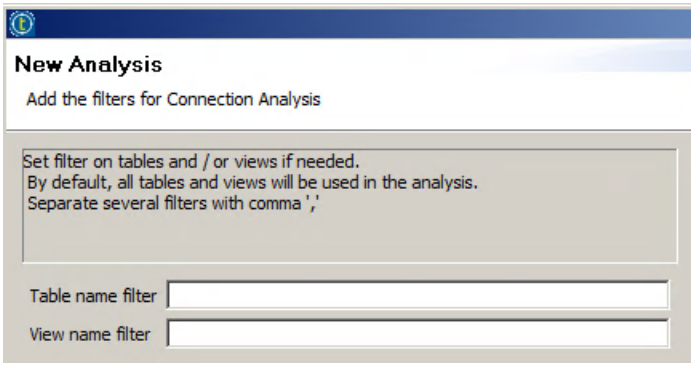
Click **Next**.

c. Choose a connection to analyze. Expand **DB connections** and click **StagingDB** (the database connection metadata you created earlier).



Click **Next.**

d. You can limit the analysis to specific tables or views. For example, entering *customer,contract* filters out tables not named *customer* or *contract*, only reporting statistical information on tables by either name in any catalog. The default is all tables and views.



Leave the boxes empty so the analysis will include all information about the database connection.

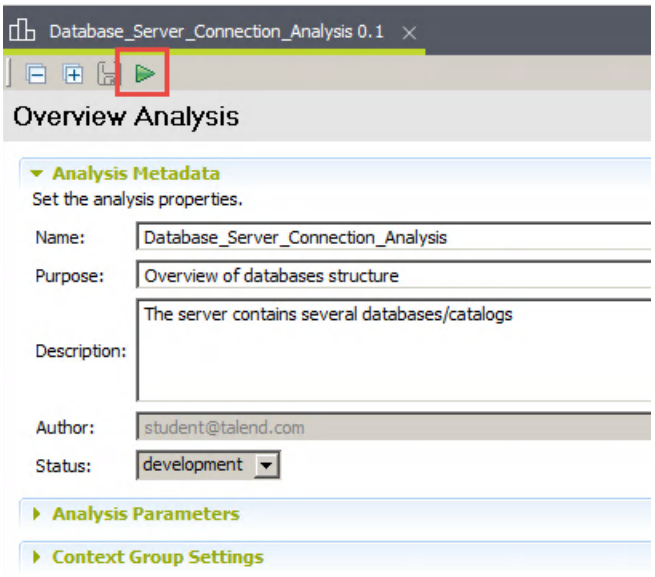
Note: If necessary, you can add information to these boxes later, in your saved analysis.

Click **Finish**. The analysis is open in the workspace with no results.

Examine the results

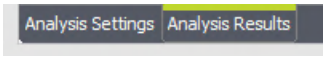
1. RUN THE ANALYSIS

   a. At the top of the workspace, click the **Run** icon.

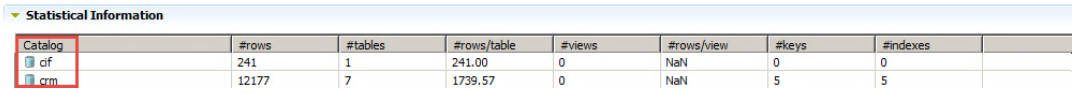


   b. When the analysis runs successfully, the Analysis Results tab appears.

   Note: You can switch between the analysis settings and Analysis Results tab using the tab bar in the lower section of the workspace.



   Global statistics for each databases are displayed in the Statistical Information section.



   Note: The type of database software you analyze affects the terminology shown in the analysis results. This course uses MySQL, so a *catalog* and a *database* are essentially equivalent. Other database implementations are organized differently, so, for instance, a database containing a collection of catalogs would show analysis results differently.

   In your course installation, the list of catalogs and displayed information may vary slightly.

2. EXPLORE THE CRM CATALOG

   a. To display additional information about a catalog, in the **Catalog** column under **Statistical Information**, click **crm**.

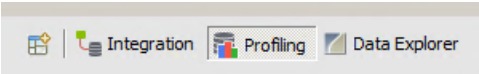
   The table on the lower left side displays the number of rows, primary keys, and indexes for each table in the selected catalog.

Table	#rows	#keys	#indexes
address	967	1	1
claim	100	0	0
contract	800	1	1
country	268	1	1
customer	999	1	1

The CRM catalog has no views; they would appear on the lower right.

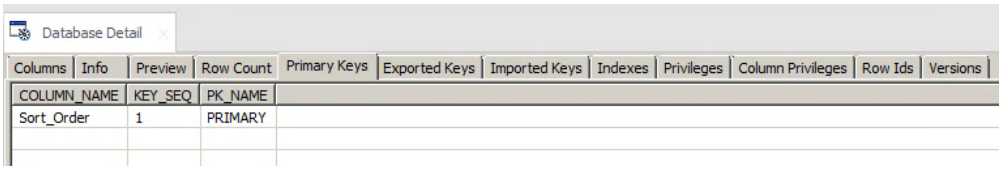
b. In the table on the lower left, right-click in the **country** row and click **View keys**.

The display changes in several ways. By clicking View keys, you change from the Profiling to the Data Explorer perspective. Your current perspective is indicated in the upper right corner, where you can switch between perspectives with a single click.



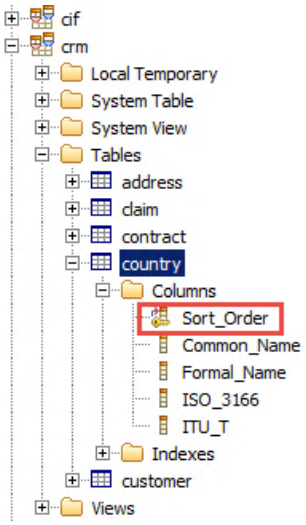
Notice the new elements displayed in the Data Explorer perspective:

- The Database Detail view opens below the analysis results. It shows basic information on the primary key of the country table.



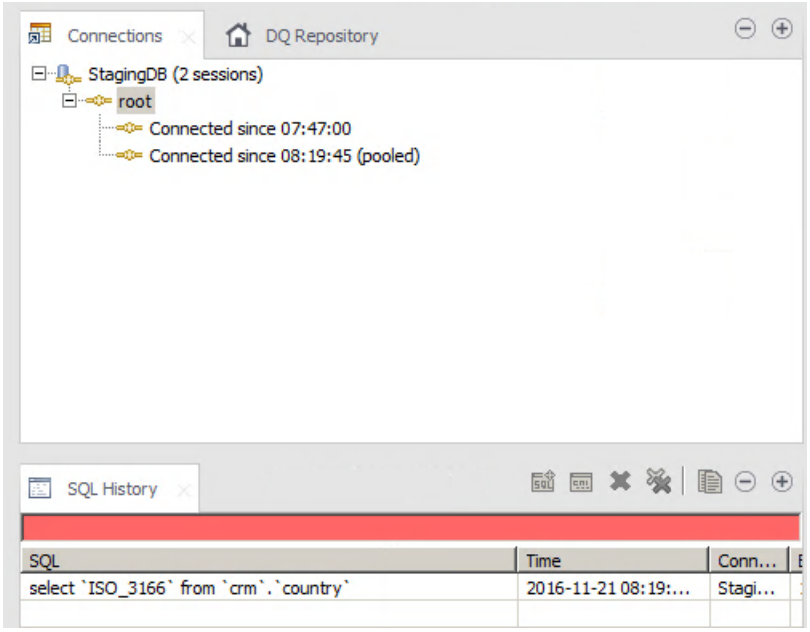
Other details about this table are available in the other tabs in the Database Detail view.

- The Database Structure tree is displayed on the right side of the window.



The database selected in the screenshot is the same as displayed in the Database Detail view. The primary key of the table is displayed with a specific icon.

» In the tabs displayed in the upper left corner, you see basic connection and SQL history information.

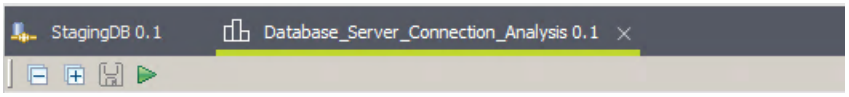


c. To again display the Profiling perspective, in the upper-right corner, click the **Profiling** button.

You use the Profiling perspective to create and run analyses, the Data Explorer perspective to drill down on details, and the Integration perspective to create data integration Jobs.

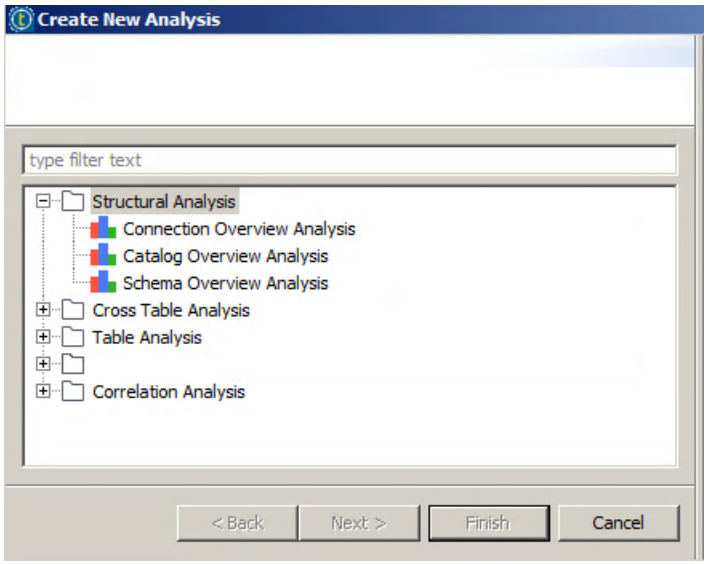
Note: Although your training installation may include additional perspectives, they are not used in this course.

d. To avoid confusion, close all opened tabs before continuing.



Other types of analysis from the same category

Two other types of analysis are available in the Structural Analysis category.



- A catalog analysis generates the same information as a connection analysis for an individual catalog.

  You can run a catalog analysis when you are only interested in one catalog from a database server that contains a large number of catalogs.

- A schema analysis produces the same results as a catalog analysis but is dedicated to other database management systems (like Oracle). The meanings of "catalog," "schema," and even "database" vary among vendors.

Before continuing to the next lesson, read the wrap-up.

Wrap-Up

In this lesson, you started Talend Studio and learned about perspectives. Then you created and stored database connection information as metadata in the DQ repository. You learned that storing connection information as metadata prevents you from having to enter basically the same information each time you create and configure an analysis.

You learned about structural analyses. You created and ran a connection overview analysis to display basic information about the data stored in your database server, including catalog and table information. The analysis results displayed the names of tables and views in catalogs, along with the number of rows, indexes, and keys for each. This analysis provided an excellent overview of the type of information in your database.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

This page intentionally left blank to ensure new chapters start on right (odd number) pages.

LESSON 2

Column Analysis

This chapter discusses the following.

Column analysis

Lesson overview

The first analysis from the Structural Analysis category gave you an overview of the catalogs and tables on your database server.

Now you will take a closer look at the data in individual columns. To do this, you must use analyses from the Column Analysis category. This lesson focuses on basic column analysis.

Basic column analysis is a type of custom analysis for which you can manually set up several indicators. Gathered information such as the length of values, patterns that values match, uniqueness of values, and other statistics can help you identify problem areas and ultimately determine how to resolve data quality issues.

Consider the customer table the kernel of the CRMdatabase. Users directly create customer accounts online, and the email addresses they provide are used as log-in identifiers for future visits to the Web site. In addition, email addresses are used for easy communication with customers. Because no mask for type assistance is available in the CRM Web interface, data issues may occur. However, stored values must be unique and follow a valid email address pattern.

Objectives

After completing this lesson, you will be able to:

- Create a basic column analysis to analyze individual column data statistics and interpret the results
- Use built-in patterns based on regular expressions, modify existing patterns, and generate new patterns (and apply them to an analysis)
- Set indicator thresholds
- Generate an integration Job from analysis results

The first step is to create a new analysis.

Performing a basic column analysis

Overview

For your first in-depth analysis, you will examine the Email column in the customer table of the CRM catalog. The customer table contains primary customer information.

Note: A basic column analysis may also be referred to as a standard or single-column analysis. This is because all indicators are computed against a single column. Essentially, a standard single-column analysis and a basic column analysis are the same thing.

Talend Studio enables you to do a basic column analysis of several columns, but each column is analyzed separately.

Later, in the "Table Analysis" lesson, you will learn about column set analysis, which reports statistics for the dataset represented by all selected columns.

The analysis process

The process to set up an analysis can be summarized in a few steps:

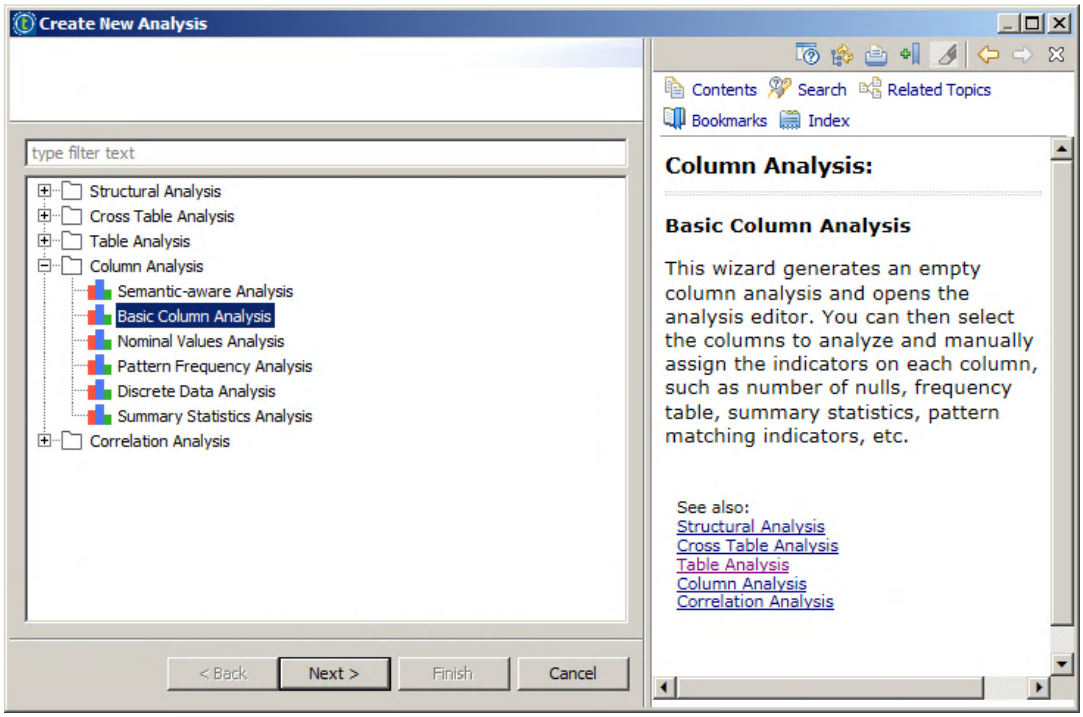
1. Create a new analysis
2. Browse categories and select an analysis type. Provide a name, purpose, and description
3. Select data to analyze (for example, tables, columns, catalogs)
4. Select indicators (for example, number of rows, number of duplicates)
5. Set up selected indicators (for example, patterns, thresholds, data filters)
6. Run the analysis and examine the results

Setting up a basic column analysis

1. CREATE A BASIC COLUMN ANALYSIS

   Right-click the **CRM_Analyses** folder.

   a. Click **New Analysis**, expand **Column Analysis**, and click **Basic Column Analysis**.

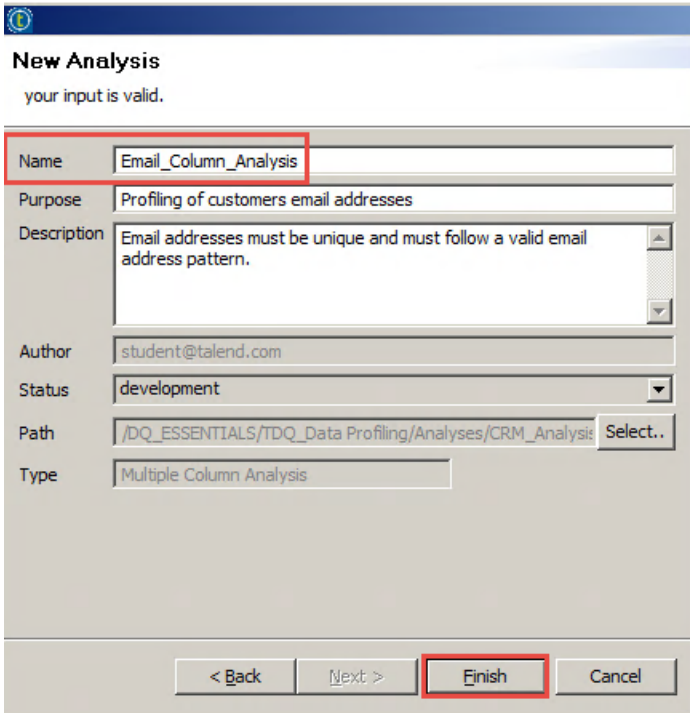


b. Read the context-sensitive help for **Basic Column Analysis**. Click **Next**.

c. In the **Name** text box, enter *Email_Column_Analysis*.

d. As a best practice, fill in the **Purpose** and **Description** boxes.

e. Click **Finish**.

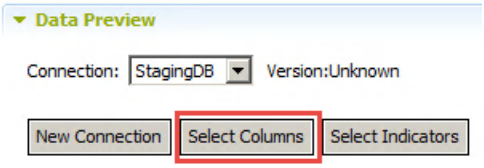


The new analysis appears in the Data Profiling repository in the appropriate folder and is automatically open in the Profiling perspective. The analysis is displayed on two main tabs, Analysis Settings and Analysis Results, selectable in the lower left corner of the window. The Analysis Settings tab is displayed by default, and the analysis configuration options are organized in sections.

2. SELECT DATA

When creating a new analysis, the Data Preview section is open by default. On the **Connection** drop-down menu, **StagingDB** is already selected.

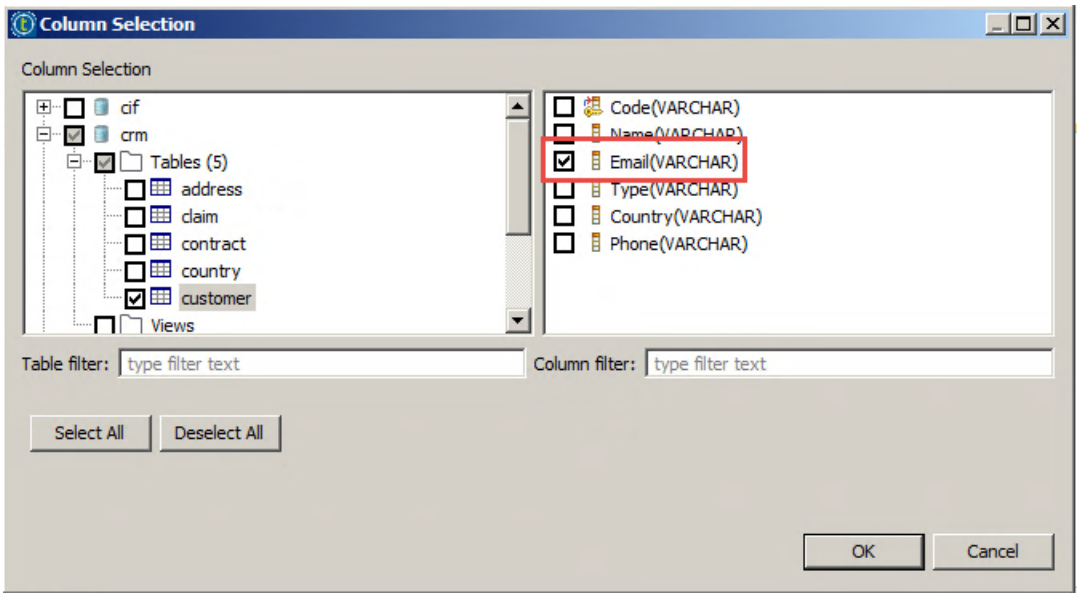
To specify which columns to analyze, click the **Select Columns** button.



a. Expand the **crm** catalog, then the **Tables** subdirectory, and click the **customer** table.

Notice that the right pane shows column names for the customer table.

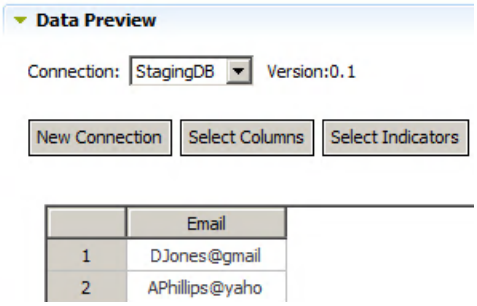
b. Select the **Email** column.



Note: It is possible to select several columns, but for readability reasons, you start with a single column.

c. Click **OK**.

The Email column is shown in the Analyzes Columns section and an overview of the data is displayed in the Data Preview section.



3. SELECT INDICATORS

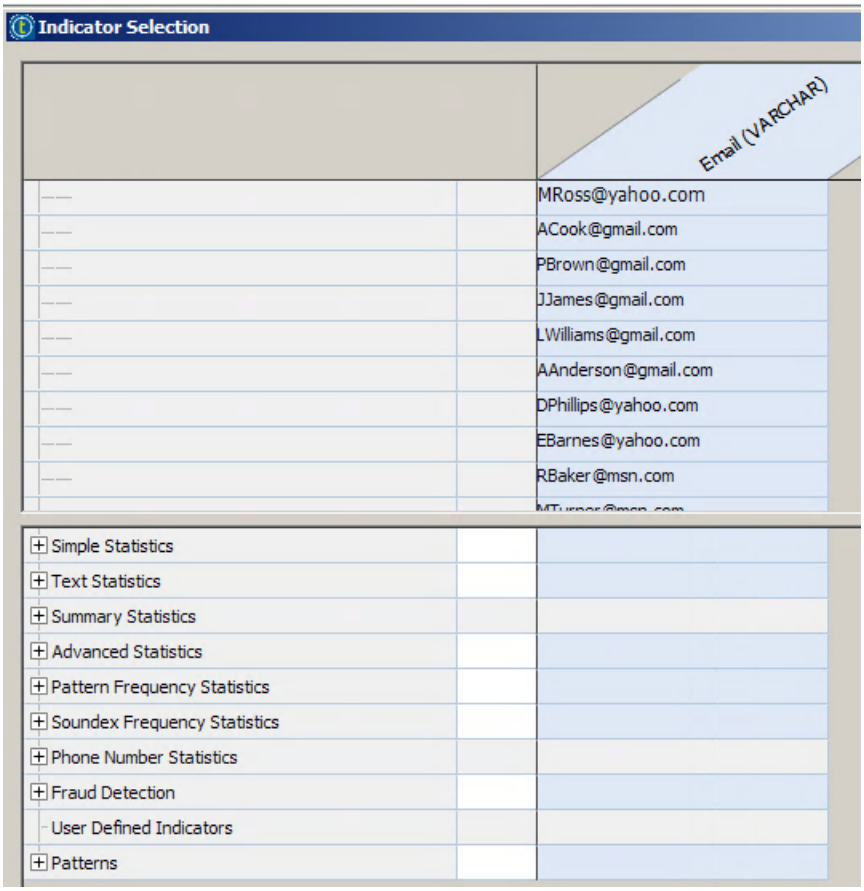
You use the Select Indicators button to specify the statistics to include in the analysis for the previously selected column.

This button is available in both the Data Preview and Analyzed Columns sections.

a. Click the **Select Indicators** button.

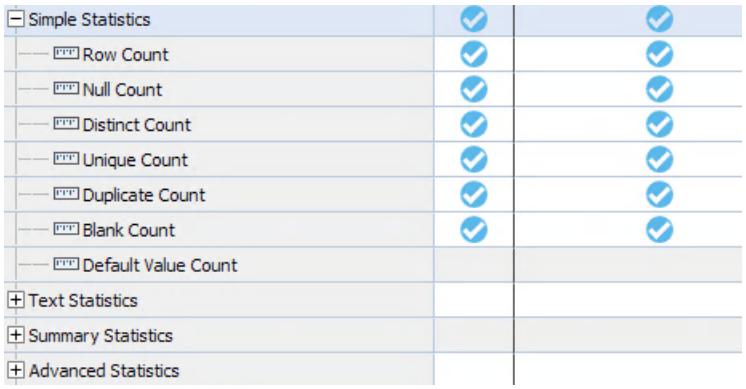
A glimpse of the data is displayed in the Data Preview section, near the top of the window. You can select indicators below.

Notice that the statistics are grouped in categories, for example, **Simple Statistics** and **Text Statistics**.

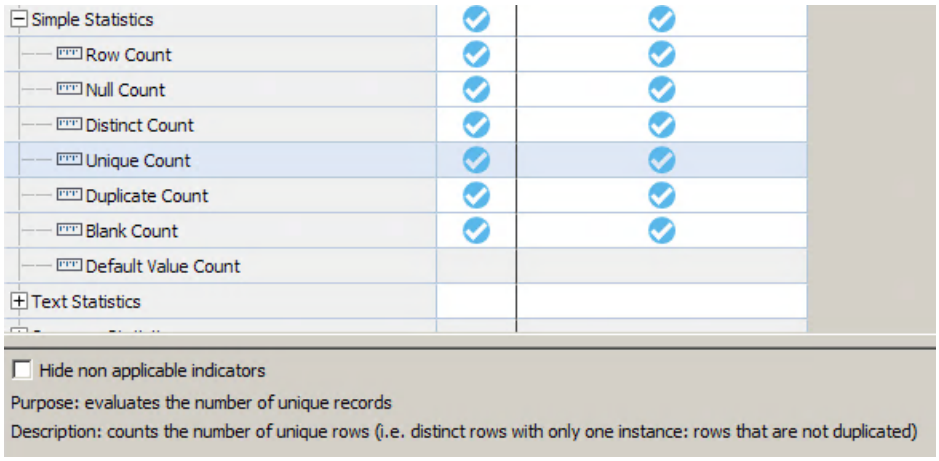


b. Expand **Simple Statistics**. To select all types of simple statistics, click the first column to its right. The table expands to show the details included in the results of the analysis.

Some statistics may not be compatible with the data type (for example, Default Value Count indicator); these are automatically grayed out.

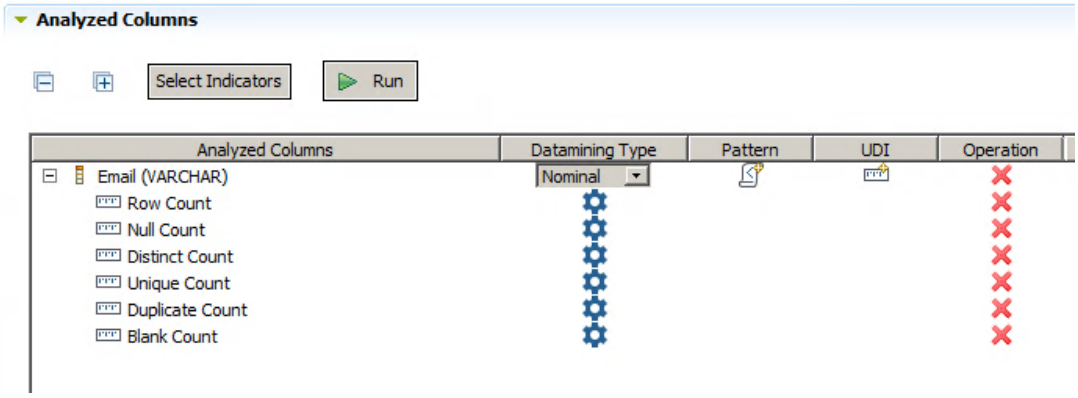


Click some indicators to display a purpose and description in the lower section.



c. Click **OK**.

The Analyzed Columns list shows the statistics selected for the column. Although you can manipulate analysis results by adding, deleting, or moving selected entries, leave them as is for now.

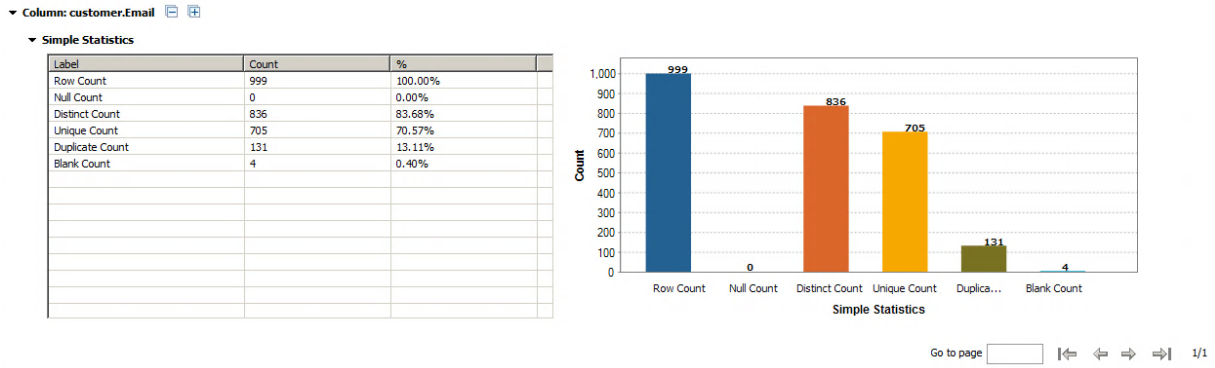


d. To run the analysis, click the **Run** icon.

Examining the results

The Analysis Results tab is automatically displayed.

This view shows the results as both tables and graphs.



Look closely at the indicators you selected.

- **Row Count** is the number of records in the column.
- **Null Count** is the number of rows without any data.
- **Distinct Count** is the number of distinct values with one or several occurrences in data.
- **Unique Count** is the number of distinct values with exactly one occurrence. If Unique Count equals Row Count, the column is a good candidate for primary key.
- **Duplicate Count** is the number of distinct values with several occurrences. Simply put, Duplicate Count and Unique Count equal Distinct Count.
- **Blank Count** is the number of non-null records that contain only spaces (or nothing).

The meanings of these indicators slightly differ according to the relational databases they are used for. These definitions are true for a MySQL database. For example, Oracle does not distinguish between the empty string and the null value.

Analysis outcome

The email address is an important customer attribute because it is used as both a log-in ID to access the Web site and a recipient address for messages. For security reasons, a log-in ID must be unique. The analysis reveals that this simple control was not done or was not efficient enough during the user-creation process. Having some blank or null records is also a problem.

How can you be sure that values stored in the Email column can be used to send electronic messages? You must study the structure of the data.

Adding regular expressions

Overview

Beyond basic and text statistics, a data analysis allows you to study the data structure by collecting information about the patterns of values. You can then compare the outcome to specified regular expressions to compute matching statistics.

A regular expression (regex) is a special character string for describing a data pattern. As part of your analysis, you can compare the contents of a given column with a regex to determine if the data matches that expression.

In this lab you will create the regex from existing data. In addition, you can:

- Create the expression from scratch
- Download expressions from Talend Exchange
- Use built-in expressions
- Modify built-in expressions

Adding pattern statistics

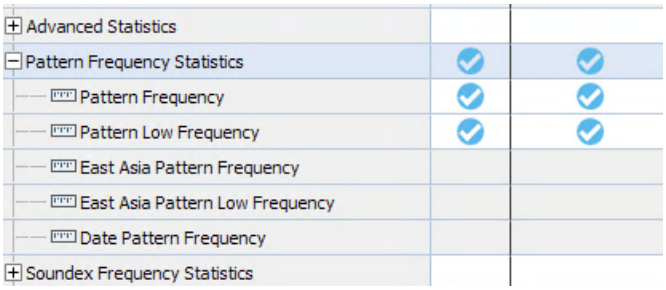
1. SELECT MORE INDICATORS

   First you need to add the Pattern Statistics indicators.

   a. At the bottom of the **Email_Column_Analysis** you recently created, click the **Analysis Settings** tab, then the **Select indicators** button.
   The Indicator Selection window appears.

   b. To select the Pattern Frequency and Pattern Low Frequency indicators, Click the **Pattern Frequency Statistics** row.

   Notice that you cannot select some indicators: they are not compatible with the data type. For instance, you cannot select the Date Pattern Frequency Table indicator because the Email column is not a date/time column.



   c. Click **OK**.

2. RUN THE ANALYSIS

   The analysis is not saved yet, as indicated by the asterisk (*) in the Email_Column_Analysis tab name.

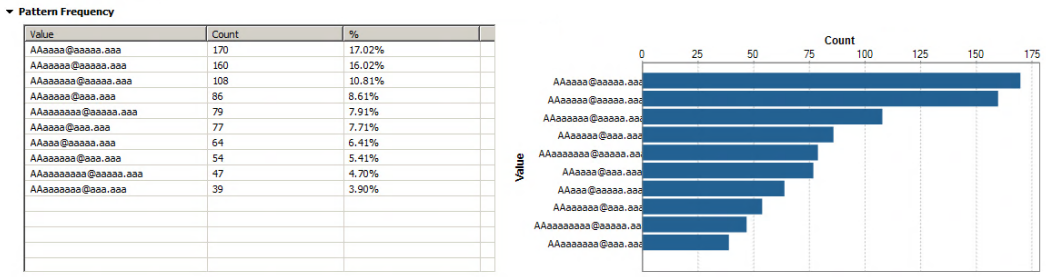
   Click **Run**.

   The analysis is saved and run, the asterisk (*) has been removed.

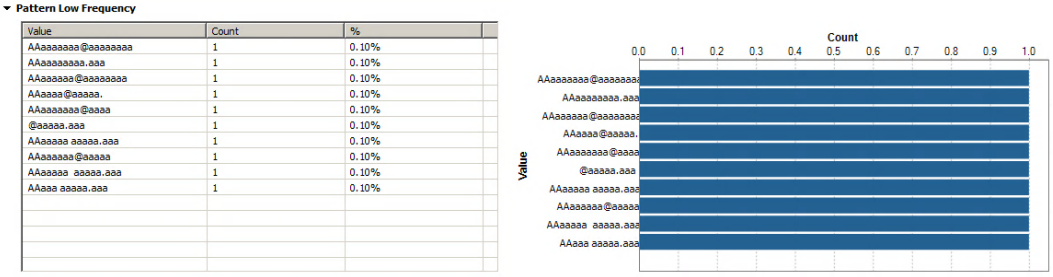
3. EXAMINE THE RESULTS

   Browse the **Analysis Results** tab and examine the **Pattern Frequency Statistics**:

   - The Pattern Frequency indicator shows the pattern values most frequently found in the column. Letters in lower-case or uppercase, numbers, and specific characters are recognized. Most of the values should correspond to a classic email address format: a more or less complex alphanumeric string on the left, followed by an at symbol (@), followed by another alphanumeric string (the domain), followed by a period and ending with a two-to-four-character string.



» In the Pattern Low Frequency statistics, the order is reversed: only the less-frequent patterns are shown. Having both lists can be useful when a column contains values that match many different patterns. In general, when a specific pattern frequency is very low, that pattern may represent values that are incorrect and warrant further examination.



Some patterns in this list generally do not look at usual email addresses. This confirms the original hypothesis of a lack of data control on the input form that leads to poor data quality.

Generating a regex from pattern indicators

As part of the analysis, you can compare the contents of a given column to a regex to see if the data matches that expression. The regex can be either built in, created by you from scratch, retrieved from the Talend Exchange, or (in this case) generated from existing data.

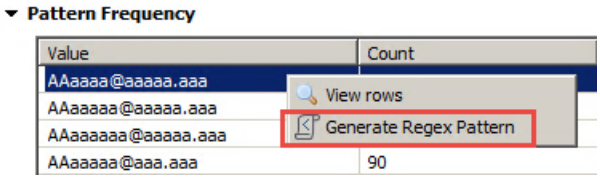
Your first approach is to reuse the most frequent pattern revealed by the Pattern Frequency indicator to create a regex.

1. CREATE THE PATTERN

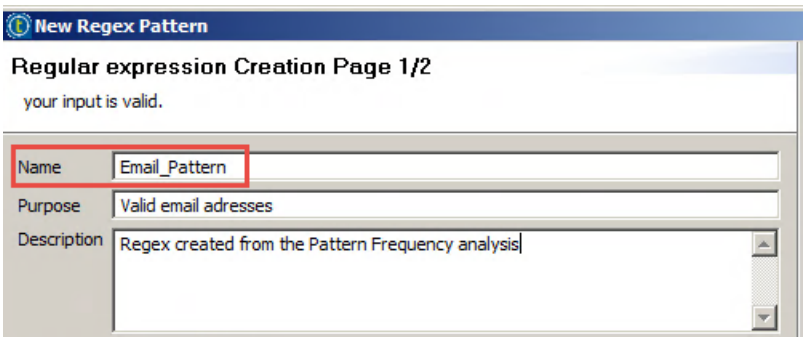
   Create the pattern from the **Email_Column_Analysis** results.

   a. In the **Pattern Frequency** section, locate the pattern with the highest row count (the pattern most frequently found in data).

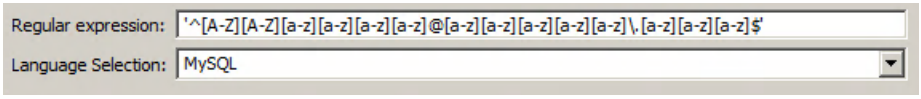
   b. Right-click it and select **Generate Regex Pattern**.



   c. On the first page of the **New Regex Pattern** wizard, name the pattern *Email_Pattern* and add a purpose and description. Click **Next**.



d. Examine the pattern definition:



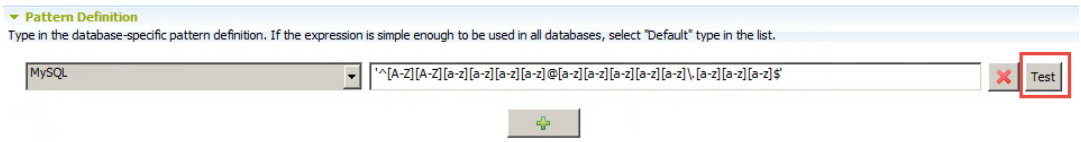
Notice that you can specify the particular database type for this pattern to allow for variations in the regular expressions. In this course, your patterns are for a MySQL database. The regex itself simply matches the selected email pattern, with no variable.

Click **Finish**.

2. TEST THE PATTERN

In the Pattern Settings window automatically open in the workspace, you can modify the pattern and even test it.

a. To the right of the **Pattern definition** section, click the **Test** button.



The Pattern Test View appears in the lower pane.

b. In the **Test Area** text box, enter an email address and click **Test** to see if the string you entered matches the pattern. Only addresses that exactly match the pattern are accepted. To become efficient, the regular expression must be enhanced.

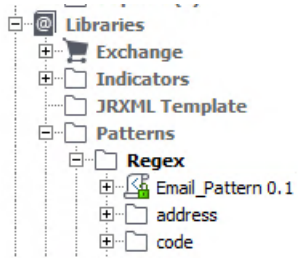
The Pattern Test View lets you verify a regex pattern to make sure it matches the values you want to match. This can be useful if you create a pattern from scratch or if the pattern is very complex.

Hint: When testing patterns, be careful to avoid pressing the ENTER key unless you want to match that in your pattern as well. It can produce undesired results because the row return is part of the character string that is tested.

c. Close the **Pattern Settings** window and **Pattern Test View**.

d. In the **Repository**, navigate below **Libraries** and notice that the generated regular expression patterns you just cre-

ated are saved there.

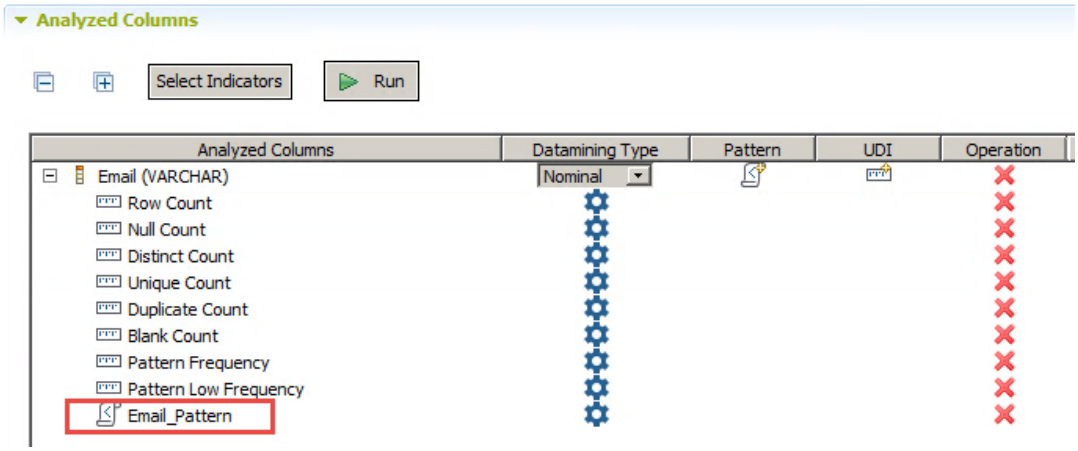


3. USE THE PATTERN

Reuse the pattern in the **Email_Column_Analysis**.

a. Click the **Analysis Settings** tab for **Email_Column_Analysis**.

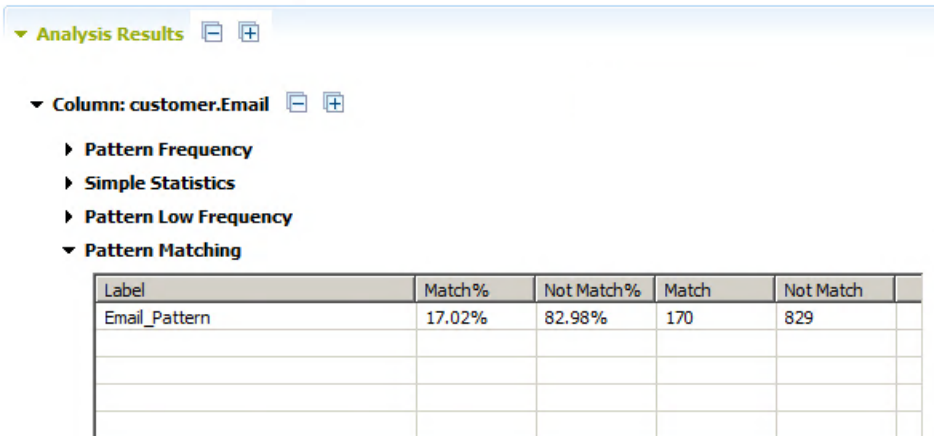
b. Drag the **Email_Pattern** regex from the **Repository** onto **Email** in the **Analyzed Columns** list.



The pattern appears at the bottom of the indicators list. The analysis will now compare the values in the Email column with the Email_Pattern regex you created.

c. Run the analysis.

d. Click the **Analysis Results** tab and locate the **Pattern Matching** indicator. Notice that only a few values match the pattern.

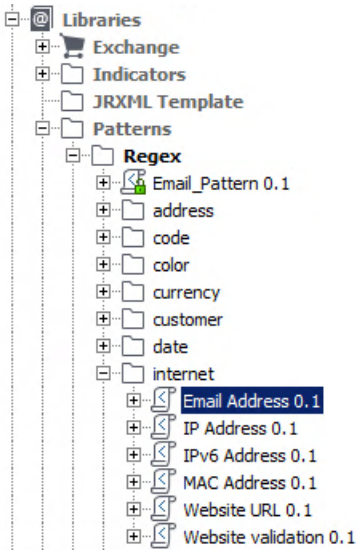


Using a built-in regex produces more-valuable results.

Using a built-in regex

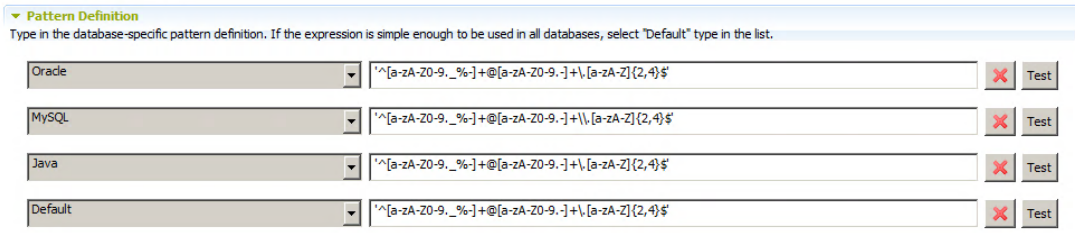
1. EXAMINE THE BUILT-IN PATTERN

   In the **Repository**, expand **Libraries**, **Patterns**, **Regex**, and **internet**.



   a. Double-click **Email Address**. The pattern settings for email addresses open.

      In the **Pattern Definition** section, notice that the pattern includes regexes for databases, Java, and the default entry. All of these display the standard format of an email address.



The salient points with respect to the regex are as follows:

- **^**

  Start of a string

- **[0-9]** Any digit
- **[a-zA-Z]** Any letter (uppercase or lowercase)
- **[a-zA-Z0-9]** Any letter or digit
- **$** End of string

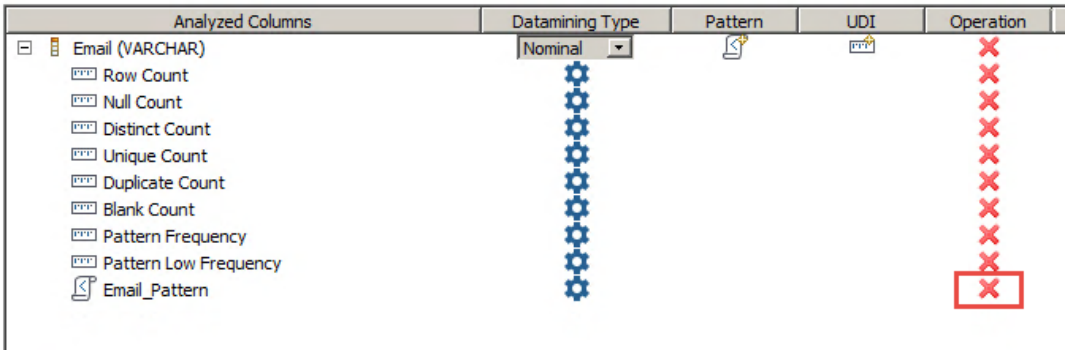
  Note: In regular expressions, a backslash (\) is used to precede a character you want matched literally. In this example, the backslash means to literally match the period (.); do not expand it to mean *any character* (which is the interpretation of a period in a regex).

b. Close the **Email Address** pattern tab.

2. USE THE BUILT-IN PATTERN

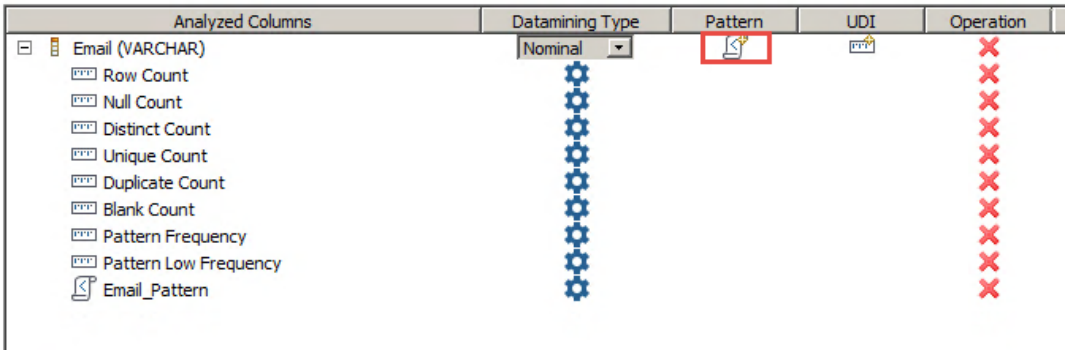
To replace the Email Pattern control you previously set up with the built-in Email Address regex from the repository, open the **Analysis Settings** tab for the **Email_Column_Analysis**.

a. To remove the **Email Pattern** control, click the **red cross symbol** next to it.

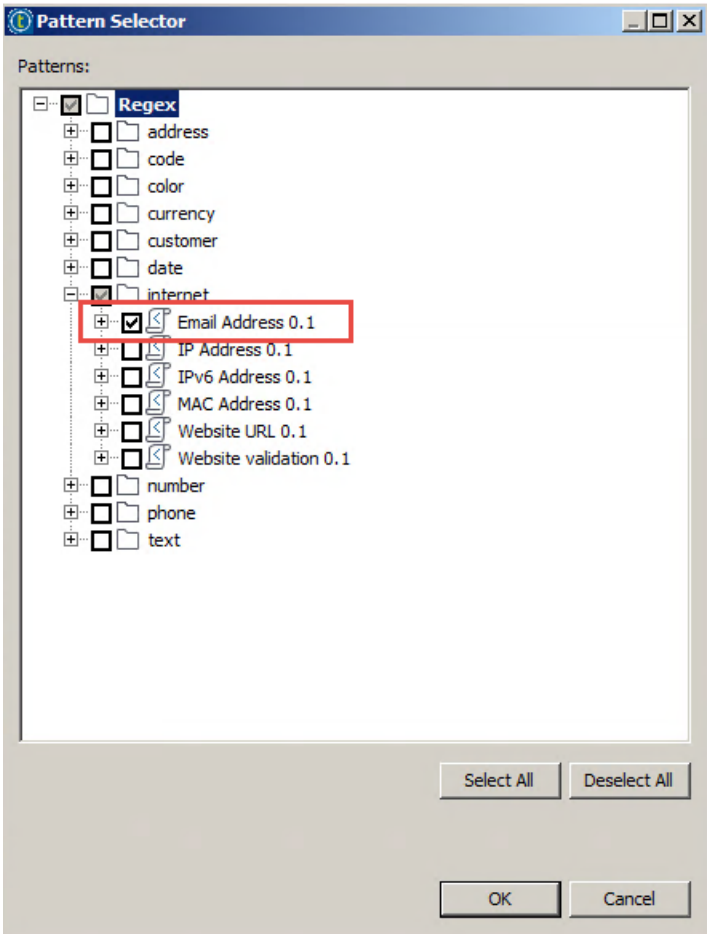


You will use an alternative process to add another pattern comparison to the analysis.

b. In the **Pattern** column of the **Analyzed Columns** list, click the **Add pattern** icon (the only one in that column). The Pattern Selection window opens.

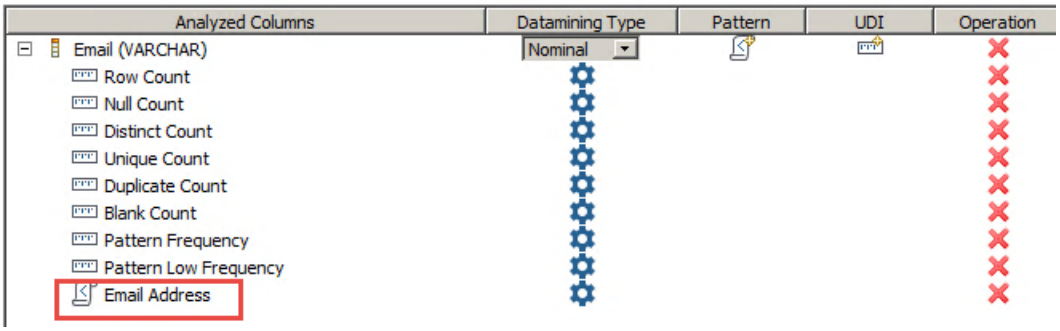


c. Expand **Patterns**, **Regex**, and **internet**, then select **Email_Address**.



d. Click **OK**.

The pattern appears at the bottom of the indicators list.



This process has the same effect as dragging the pattern from the repository.

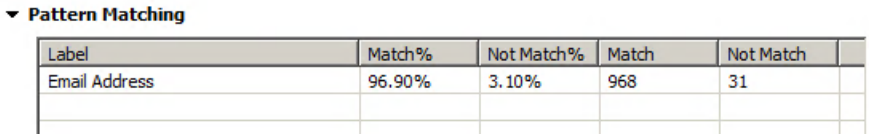
Now the analysis compares the values in this column to the built-in pattern for email addresses.

3. EXAMINE THE RESULTS

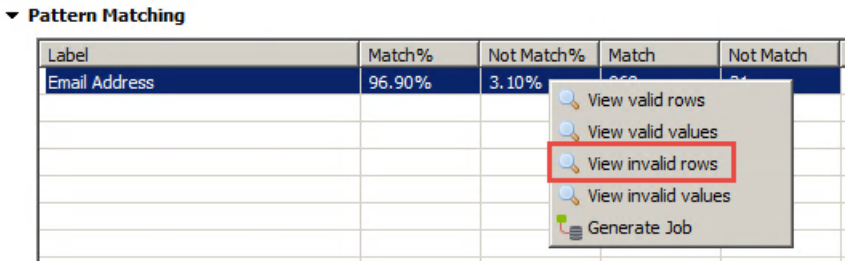
Run the analysis.

a. Click the **Analysis Results** tab and locate **Pattern Matching**.

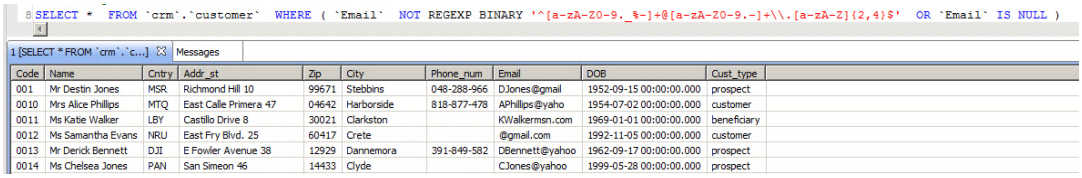
The pattern matching results indicate that even though the Email column contains a wide variety of text patterns, most of them match the basic structure of an email address.



b. Right-click the pattern row and click **View invalid rows**.



SQL Editor opens, displaying the results of a SQL query.



The View invalid rows option generates and runs an SQL command to show you the rows with values that do not match the pattern. These rows are likely to require action. Notice that the same command is available in the bar chart as in the table—in the bar chart, you can right-click and then click **View invalid rows**.

c. Close **SQL Editor**.

d. You can also use SQL Editor for other indicators.

In the **Pattern Frequency** statistics, right-click a pattern, then click **View rows**. Email addresses that follow this pattern are displayed in SQL Editor.

Now that your analysis is correctly set up, you can set up thresholds to highlight important data quality issues.

Defining indicator thresholds

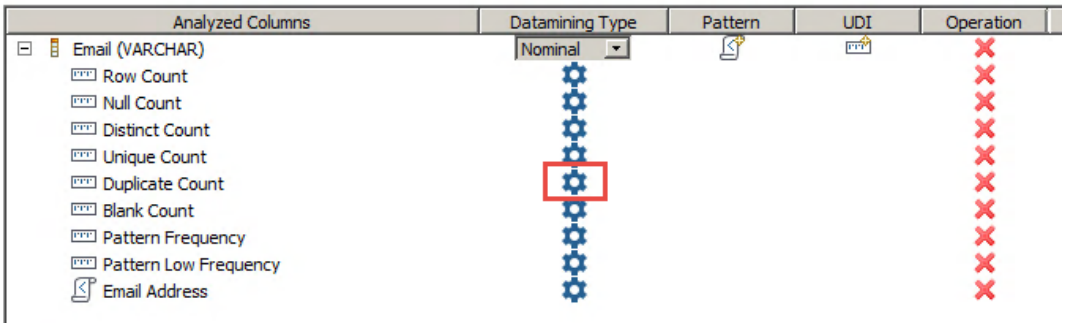
As explained in the introduction to the Basic Column Analysis lab, you must make sure that the values stored in the Email column do not contain duplicates, and that they strictly follow the Email Address pattern.

You will now enhance your analysis with thresholds to highlight major data quality issues with the most critical indicators.

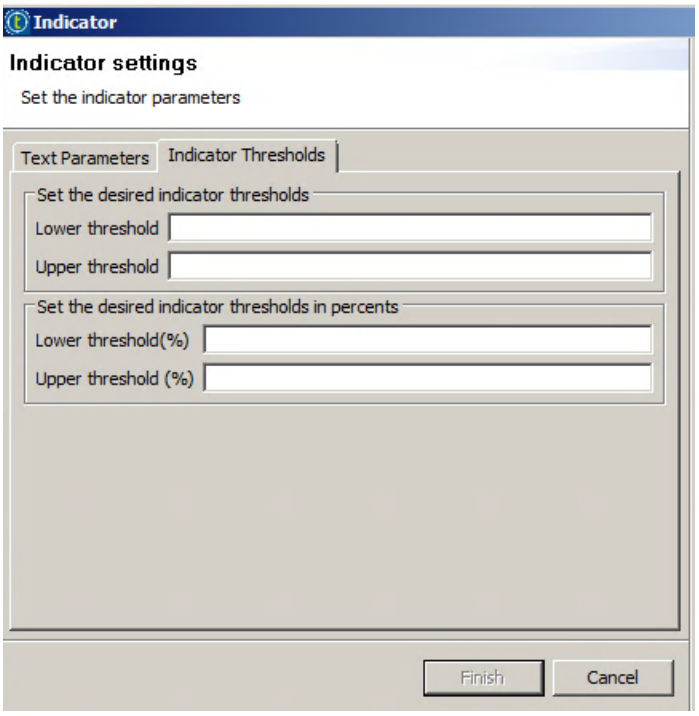
1. SET UP THRESHOLDS

   To set up thresholds for the **Duplicate Count** and **Email Adress** indicators, display the **Analysis Settings** tab.

   a. In the **Datamining Type** column, next to **Duplicate Count**, click the **Options** icon (shaped like a gear).



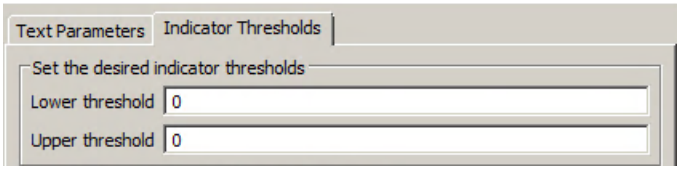
   b. The **Indicator** window opens. Select the **Indicator Thresholds** tab.



   This is where you can set thresholds for the values of this statistic. Take a moment to read the context-sensitive help for indicator thresholds.

   Data ranges can be defined by value or percentage. If the value of the indicator is outside the data range, it appears in red in the analysis results.

c. In the top two threshold boxes, enter zero. Any value other than zero will be displayed in red.

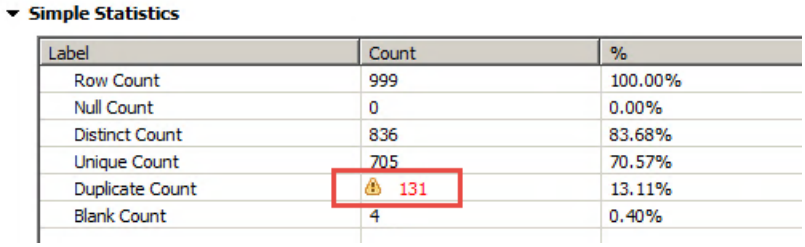


Click **Finish**.

2. SHOW THRESHOLDS

Run the analysis and display the **Analysis Result** tab.

Notice that the value of the **Duplicate Count** indicator appears in red.



Thanks to the threshold, the simple statistics are now easier to read. You will continue to enhance the analysis with advanced statistics.

Applying advanced statistics

Overview

In this lesson you will add a new column to the existing analysis, use the Advanced Statistics indicators, and set up a data filter to analyze a portion of data.

Adding a new column with new indicators

Your first basic column analysis was created with only one column selected. For a more consistent analysis, you can add columns.

Note: You need to select indicators for new columns as columns being analyzed separately; the indicators selection is column-specific.

You have setup several a column analysis, so some of this will be familiar. Some instructions, with screenshots, are grouped inside collapsed sections, which you can click to expand.

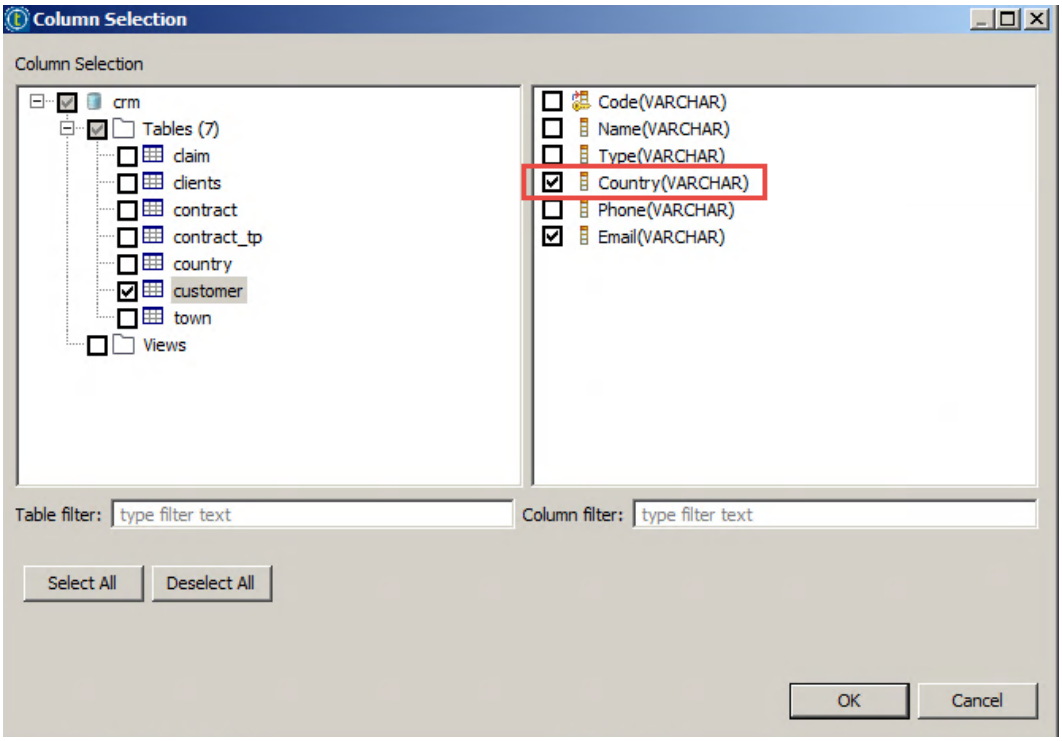
1. ADD THE NEW COLUMN

   Add the **Country** column to the **Email_Column_Analysis**.

   a. In the **Data Preview** section of the **Analysis Settings** tab, click the **Select Columns** button.

   b. Expand the **crm** catalog, then the **Tables** subdirectory, and click the **customer** table.

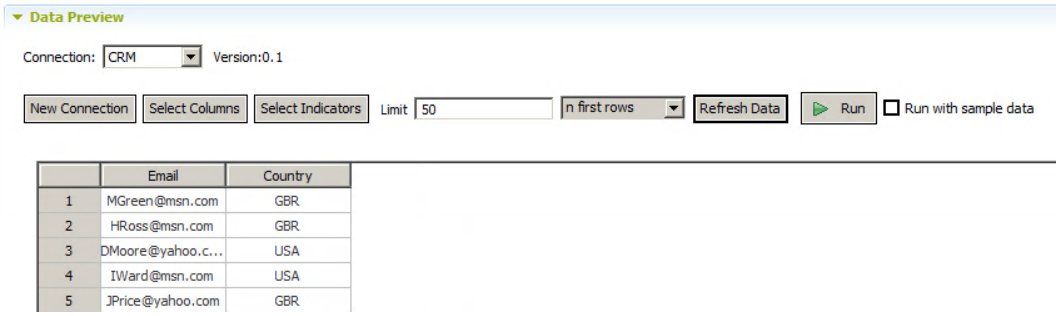
   c. Select the **Country** column in addition to the **Email** column.



   d. Click **OK**.

   The Country column is listed in the Analyzed Columns section, and a data overview appears in the Data Preview

section.



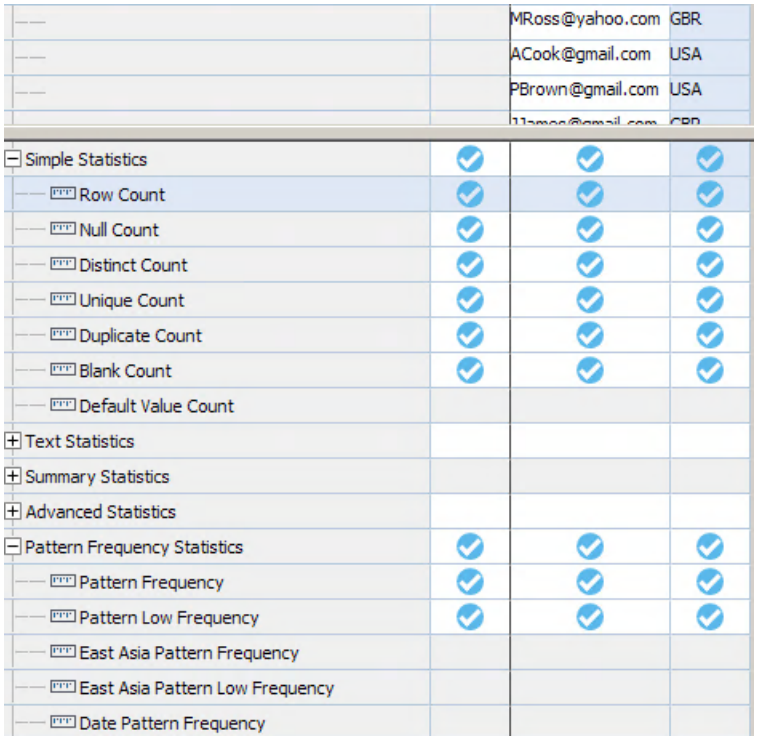
2. SELECT THE INDICATORS

   Select all the **Simple Statistics** indicators, as well as the **Pattern Frequency** and **Pattern Low Frequency** indicators, for the new column.

   a. Click the **Select Indicators** button.

   b. To select all types of simple statistics, expand **Simple Statistics** and click the **Simple Statistics** section header in the **Country** column.

   c. Expand the **Pattern Frequency Statistics** and click the **Pattern Frequency Statistics** section header in the **Country** column. Only the Pattern Frequency and Pattern Low Frequency indicators are selected, as the others are not compatible with the data type.



   d. Click **OK**.

3. RUN THE ANALYSIS

   Click **Run** and browse the results.

   The Pattern Frequency indicators reveal that most of the values contain only three characters. We can conclude that the Country column contains ISO country codes. If a value contains something other than three characters, it is erroneous.

The Distinct Count indicator shows how many specific country codes are present in the data. To generate a list, right-click the **Distinct Rows** indicator and, on the contextual menu, choose **View values**. A list of country codes is displayed in SQL Editor.

Adding advanced statistics

Close **SQL Editor** and go back to the analysis.

1. ADD NEW INDICATORS

   Display the **Analysis Settings** tab and click the **Select Indicators** button.

   a. Select all the compatible advanced statistics by clicking the **Advanced Statistics** section header in the **Country** column.

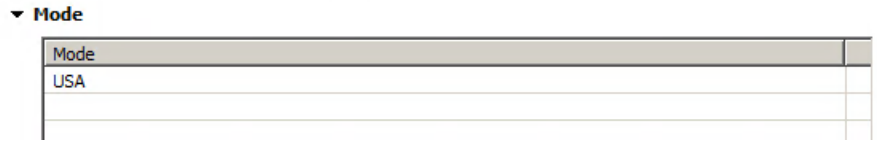
   b. Click **OK**.

2. EXAMINE THE RESULTS

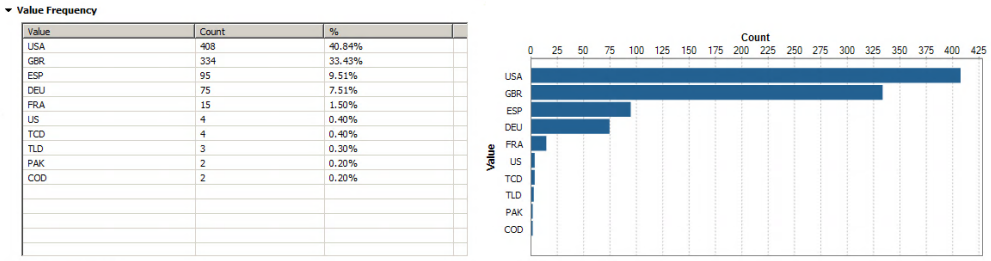
   Run the analysis and display the **Analysis Results** tab.

   a. Collapse the **Colum:Customer.Email** section in order to focus on the results for the **Country** column and browse the indicators you added.

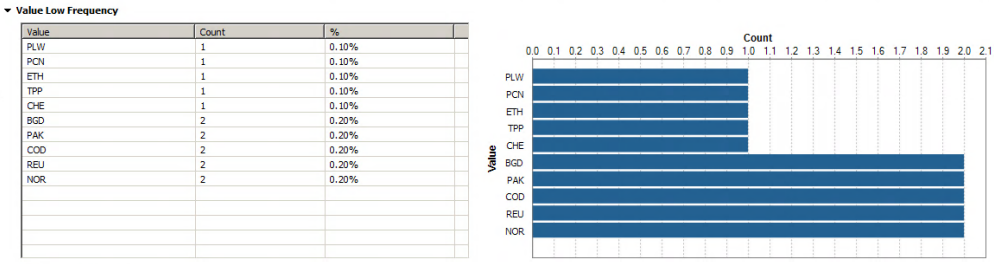
   » The Mode statistic displays the most frequently appearing country code.



   » The Value Frequency statistic lists the most frequently appearing records. You can see that the majority of records are from the US, followed by the UK (GBR).

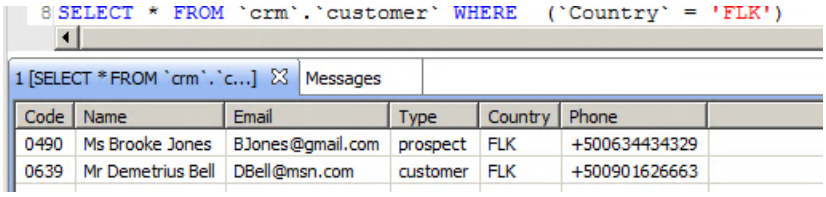


   » The **Value Low Frequency** statistic lists the less frequently appearing records. Only a few records are from a country besides the UK or US.



   b. In the **Value Low Frequency** table, right-click a country code and click the **View** rows. Rows with that country code are displayed in SQL Editor. The SQL query, with the filter applied to the selected country code, appears in the text

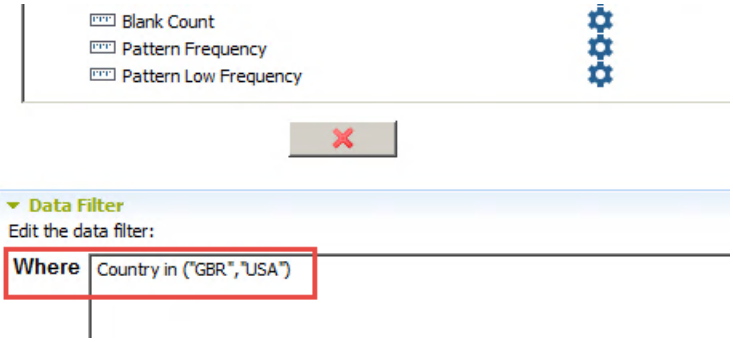
box at the top.



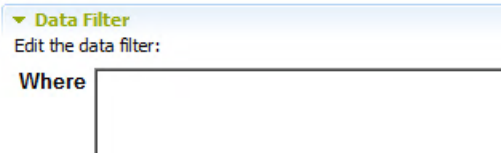
Setting up a data filter

According to the Advanced Statistics indicators, most of the records are from two countries. The analyzed data can be filtered for these countries to facilitate the analysis and prevent inconsistent results.

1. Display the **Analysis Settings** tab and expand the **Data Filter** section (below the **Analyzed Columns** section).
2. In the **Where** text box, enter *Country in ("GBR","USA")*



3. Run the analysis.
   Confirm that all the records are from the US and UK (GBR).
4. Remove the data filter and run the analysis before continuing.



In the next chapter, you will generate an integration job from the analysis.

Generating Jobs from an analysis

The Marketing Department has requested a list of valid email addresses in order to start an electronic campaign. In the first part of this lesson, you will use the analysis results to easily create an integration job to export all the email addresses that match the regular expression (regex) pattern in a CSV file.

In the second part, you will use a similar process to create another job to identify duplicates.

Generating a job to export valid email addresses

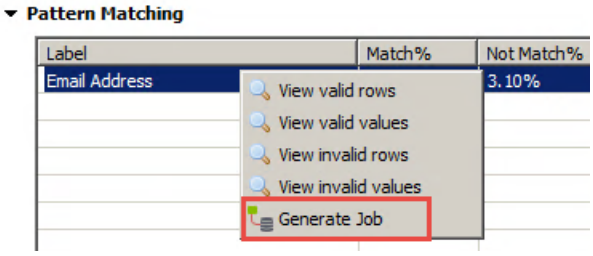
You can automatically create this integration Job from the analysis.

1. GENERATE THE EXPORT JOB

   Display the **Analysis Results** tab of the **Email_Column_Analysis**.

   a. Browse the results and display the **Pattern Matching** indicator for the **Email** column.

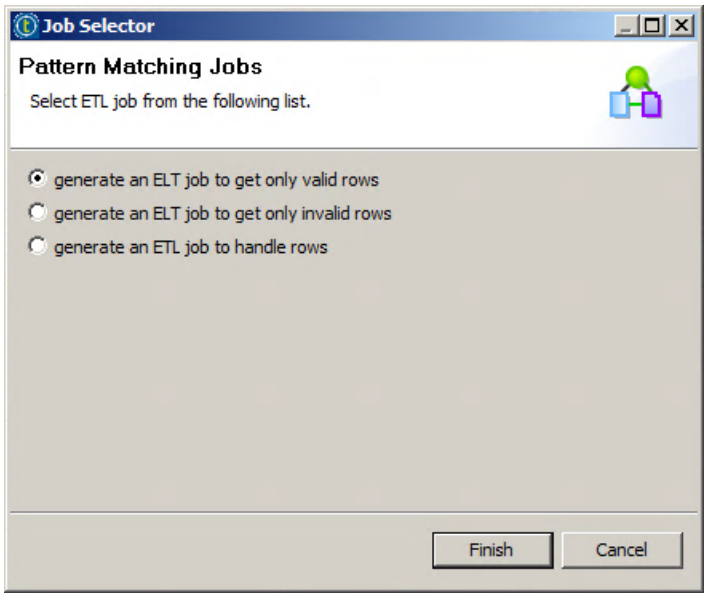
   b. Right-click somewhere in the results row.



   - You can use the first four items of the contextual menu to display valid and invalid rows and values in SQL Editor.
   - You can use the Generate Job item to automatically generate an integration job.

   c. Select **Generate Job**.

   Three job templates appear in the **Job Selector** window.

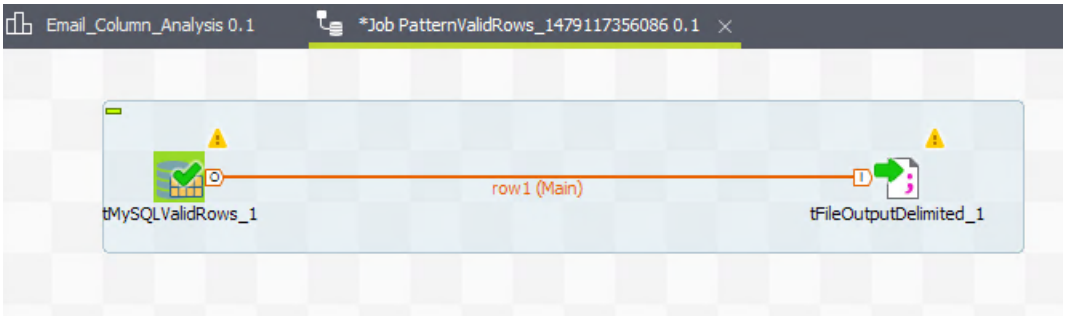


- The first template generates a Job that exports valid rows in a CSV file.
- The second generates a Job that exports invalid rows in a CSV file.
- The third combines the first two and generates a Job that exports rows in two CSV files—one for valid rows and one for invalid rows.

d. The first template is selected by default. Click **Finish**.

2. EXAMINE THE JOB

The requested Job is automatically created and displayed in the Integration perspective.



This very simple job consists of two components:

- **tMySQLValidRows** uses the Email Address regex to select the valid rows from the MySQL database
- **tFileOuputDelimited** exports the valid rows in a CSV file

Before running the job, you need to update two settings.

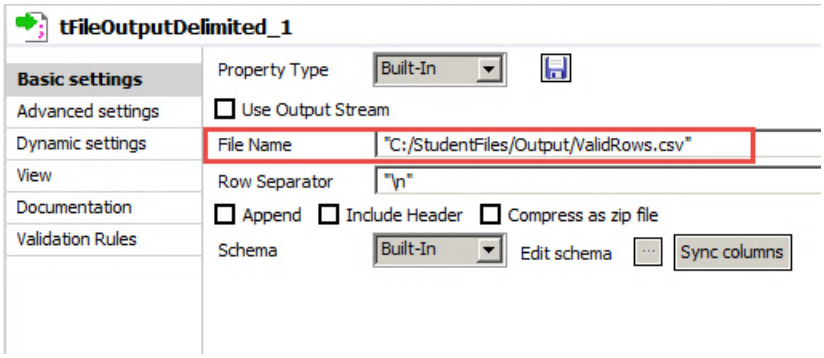
Setting up the Job

1. SET UP THE OUTPUT FILE

Correctly set up the output path and file name of the CSV file.

a. To display the **Component** view, double-click the **tFileOuputDelimited** component.

b. On the **Basic settings** tab, update the default **File Name** value with the path *"C:/StudentFiles/Output/ValidRows.csv"*

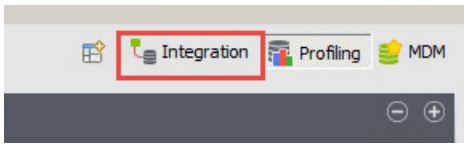


2. SET UP THE JOB TITLE

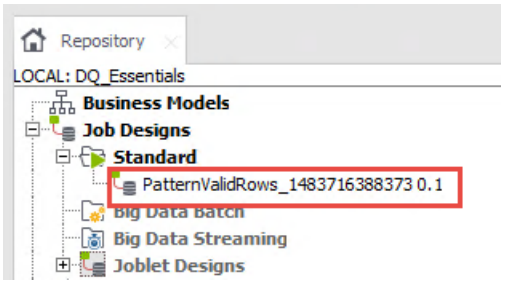
You must update the default Job title (a best practice).

a. When the Job is open in the Designer, you cannot update its name. On the upper left toolbar, click the **Save** icon, then close it from the tab bar of the Designer.

b. The Email_Column_Analysis is the only element left open in the Designer. To switch back to integration profiling, in the upper right corner of the window, click the **Integration** button.



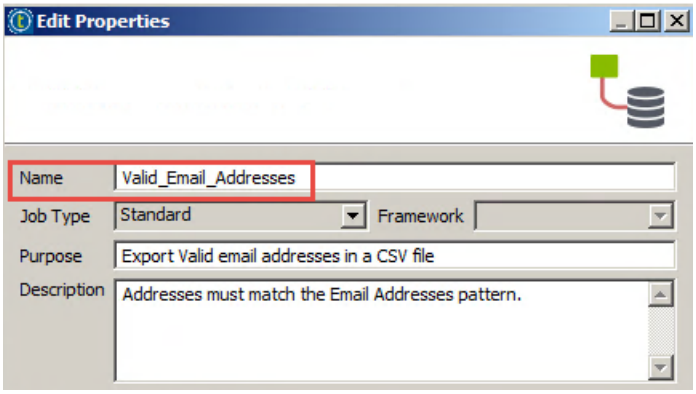
c. The job has been saved in the In the **Repository**.



Right-click the standard Job, and on the contextual menu, choose **Edit properties**.

d. Change the default name to *Valid_Email_Addresses*

You can also enter a purpose and a description.



e. Click **Finish**.

Running the Job

In the **Repository**, double-click the **Valid_Email_Addresses** job to reopen it.

1. RUN THE JOB

   Press **F6** or click the **Run** button on the **Run** tab below the Designer.

   a. In Windows Explorer, open the **StudentFiles** directory.

   b. Navigate to the **Output** subdirectory and confirm that the CSV file has been created.

Generating a Job to identify duplicates

You can also automatically create this integration job from the analysis.

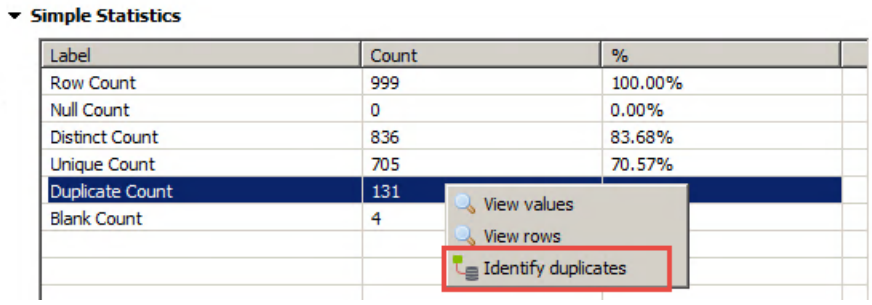
1. GENERATE THE JOB

   To switch to the **Profiling** perspective, at the top of the screen, click the **Profiling** button.

   a. If the **Email_Column_Analysis** is not open in the Designer, open it from the repository.

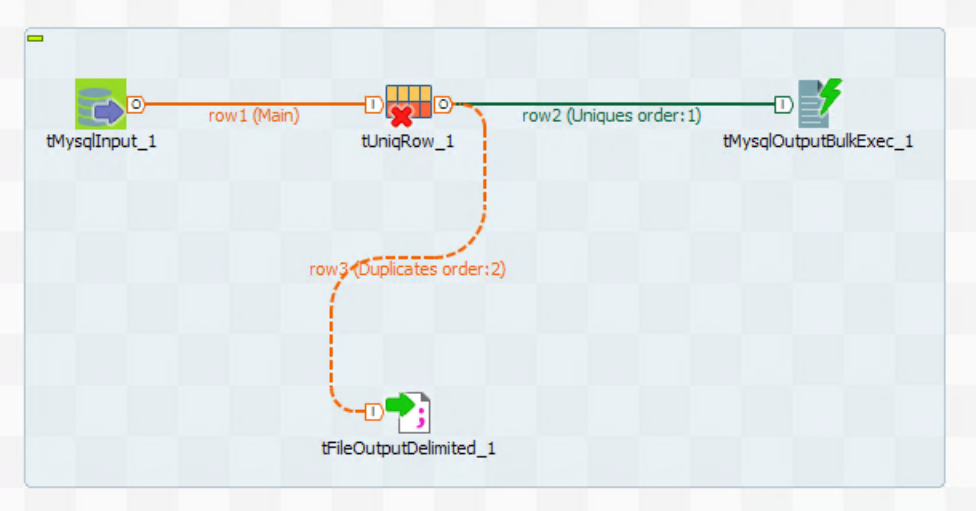
   b. Display the **Analysis Results** tab and go to the **Simple Statistics** table of the **column:Customer.Email** section.

c. Right-click the **Duplicates Count** row, and on the contextual menu, choose **Identify duplicates**.



2. EXAMINE THE JOB

The requested Job is automatically created and displayed in the **Integration** perspective.



This Job has four components:

- **tMySQLInput** exports email addresses from the MySQL database
- **tUniqRow** determines whether email addresses are unique
- **tMysqlOutputBulkExec** writes unique records to a new MySQL table named *Customer_unique*
- **tFileOuputDelimited** exports duplicate records in CSV format

Setting up the Job

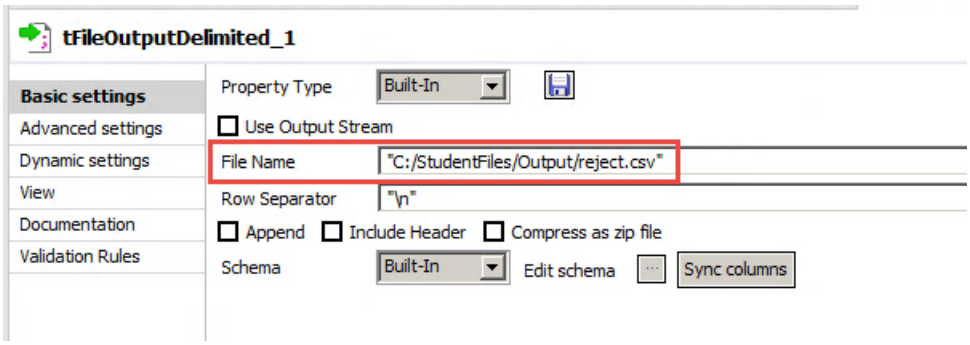
The components are again automatically configured, but it is a best practice to update some settings.

1. SETUP THE OUTPUT FILE

Correctly set up the output path and name of the CSV file.

a. Double-click the **tFileOuputDelimited** component to display the **Component** view.

b. On the **Basic settings** tab, update the default **File Name** value with the path *"C:/StudentFiles/Output/reject.csv"*

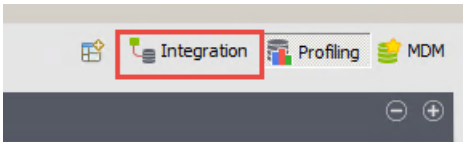


2. SET UP THE JOB TITLE

   You must update the default Job title (a best practice).

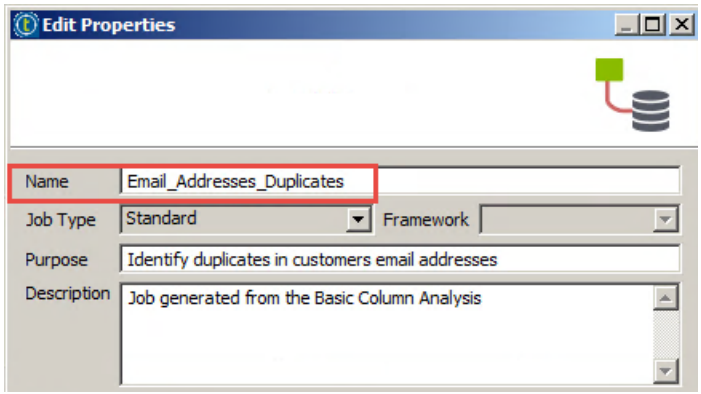
   a. You cannot update the Job name when the Job is open in the Designer. On the toolbar in the upper left corner, click the **Save** icon, then close it from the tab bar of the **Designer**.

   b. The **Email_Column_Analysis** is the only element still open in the Designer. To switch back to integration profiling, in the upper right corner of the window, click the **Integration** button.



   c. In the **Repository**, right-click the standard Job, and on the contextual menu, choose **Edit properties**.

   d. Change the default name to *Email_Addresses_Duplicates*



   You can also enter a purpose and description.

   e. Click **Finish**.

Running the job

In the **Repository**, double-click the **Email_Addresses_Duplicates** job to reopen it.

1. RUN THE JOB

   Press **F6** or click the **Run** button in the **Run** tab below the Designer.

   a. In Windows Explorer, open the **StudentFiles** directory.

   b. Navigate to the **Output** subdirectory and view the new CSV file. It contains only the duplicate records.

2. VIEW THE NEW TABLE

   You can confirm the creation of the new table using MySQL Workbench.

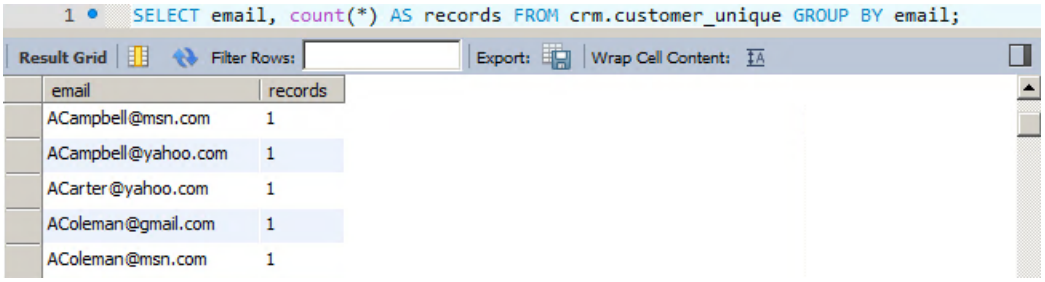
a. On the Windows Start menu, start MySQL Workbench.

b. Under **MySQL Connections**, click **Local instance MySQL56**.

c. On the left, under **Navigator**, under **Schemas**, expand **crm** and **Tables**, then click the **Customer_unique** table. Notice the single **Email** column.

d. Execute the query *SELECT email, count(*) AS records FROM crm.customer_unique GROUP BY email;*

Notice that the new table contains only unique records.



You have now finished this lesson. Complete the exercises to reinforce your understanding of the topics covered.

Challenge

Complete these exercises to further explore the use of standard column analysis and regular expressions (regexes). See the Solutions chapter for answers.

Update the Email_Column analysis

Continue to enhance the analysis with the following tasks:

- Filter the analysis to select the five most frequently appearing countries in the customer table.
- Add the Phone column, which stores telephone numbers with international prefixes, to the other selected columns. Use a predefined regex to control its pattern. Compare the results with the output of the Simple Statistics.

Analyze the primary key

In the **DQ Repository**, reopen the connection overview analysis you created and view the statistics of the **Country** table.

Consider the following questions:

- Which column has been identified as the primary key of the country table?
- How can you use a column analysis to confirm that this column is a good candidate for primary key?

In a future exercise, you will perform a cross table analysis to explore the relationships between the country and customer tables.

Solutions

Here are solutions to the challenge. Your solutions may be slightly different and still valid.

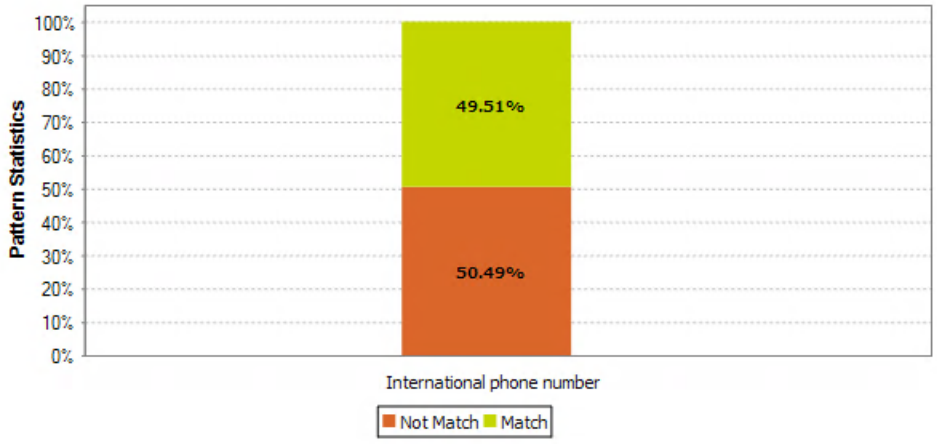
Updat the Email_Column analysis

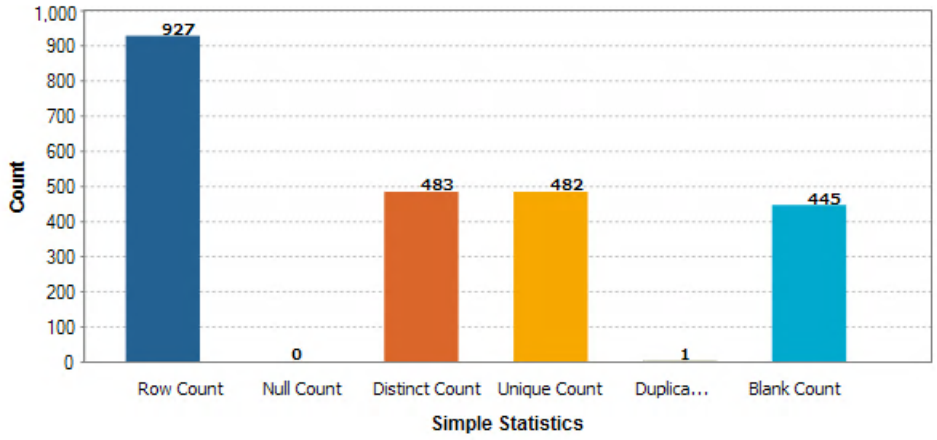
Answers:

- Use the results of the **Value Frequency** statistic to determine the five most frequently appearing countries. Then update the data filter accordingly. The new data filter should be:

  Country in ("GBR","USA","ESP","DEU","FRA")

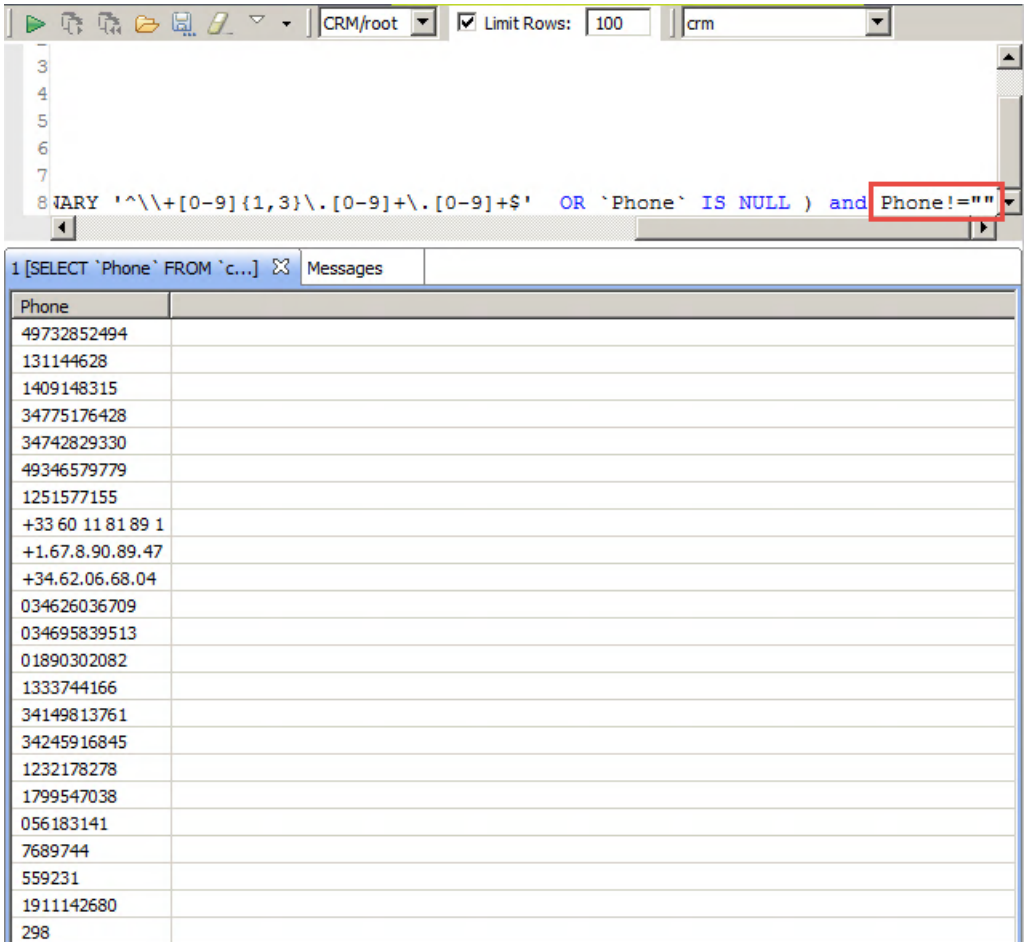
- Display the **Analysis Setting** tab. To add the **Phone** column, click the **Select Columns** button. Then click the **Select Indicators** button to select the Simple Statistics indicators. Drag the predefined regex **International phone number** from the **Repository** to the **Phone** column in the **Analyzed Columns** section. Run the analysis.
  - The Simple Statistics show that many values are missing.
  - The Pattern Matching statistic shows that half of the rows do not match the pattern.





- Right-click the **Pattern Matching** results to display invalid values in SQL Editor. You can see that the blank values are counted as invalid phone numbers. You can enhance the SQL query to remove the blank values from the

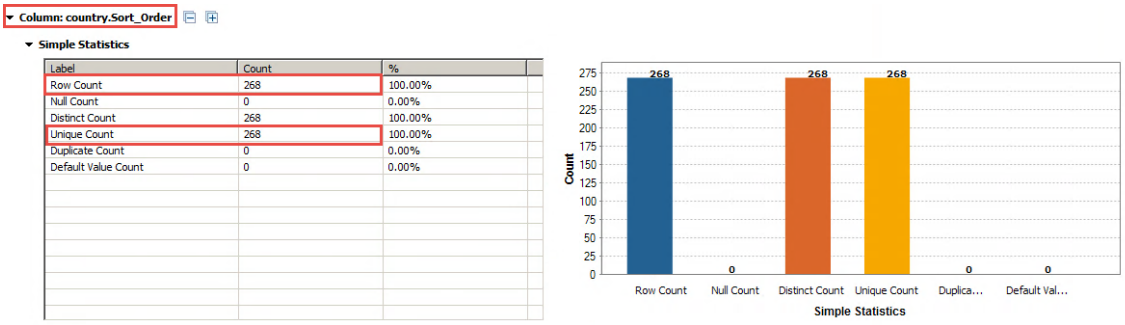
returned rows. Just add *and Phone!=""* to the end of the query.



Null and blank count

Answers:

- The Sort_Order column has been identified as the primary key.
- You can use the simple statistics of a basic column analysis to identify which column in the table is a good candidate for primary key. If the Unique Count and Row Count statistics are equal, the column can be used as primary key.



You have finished this lesson and it is time to wrap-up.

Wrap-Up

In this lesson, you created and ran several basic column analyses on individual columns of fairly typical CRM data. The analyses included a more detailed look at column comparisons within a single table, including pattern matching to filter data. You learned that regular expression (regex) patterns are built in to the repository by Talend as a convenience so you do not need to create common regexes yourself. In this lesson you used built-in regexes to help analyze data and discover anomalies. You also set up indicators as a means of flagging column data if various thresholds are violated.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

LESSON 3

Table Analysis

This chapter discusses the following.

Table analysis

Lesson overview

In this lesson you will run several types of table analysis.

Unlike with basic column analysis, in which all indicators are computed against a single column, the level of granularity for table analysis is a table row.

For example, you can use a table analysis to determine if several rows are identical or similar, or to check whether values of a table row conform to a specific business rule.

Two other points to consider:

- A row is a set of cells that belong to all columns. In this type of analysis, the row is the indivisible element that is analyzed.
- In Talend Studio, you *can choose* which columns to focus on, bypassing the others. Running a table analysis on such a set of cells is a relevant way to look for duplicates, as they can be revealed only if you do not include the primary key in the analysis.

You will start by setting up a column set analysis to identify potential duplicates from a selection of cells in the customer table. Then you will run a business rule analysis to validate data with a data quality rule written in SQL.

The table match analysis identifies duplicates, but in a much more advanced way. You will use this analysis in another lesson in this lab. This will be the starting point of a progressive exercise on matching features.

Finally, this lab does not cover the functional dependency analysis. You can find additional information online.

Objectives

After completing this lesson, you will be able to:

- Create and run a column set analysis
- Create business rules to validate data
- Use a business rule to execute a business rule analysis

The first step is to create a column set analysis.

Using a column set analysis

Overview

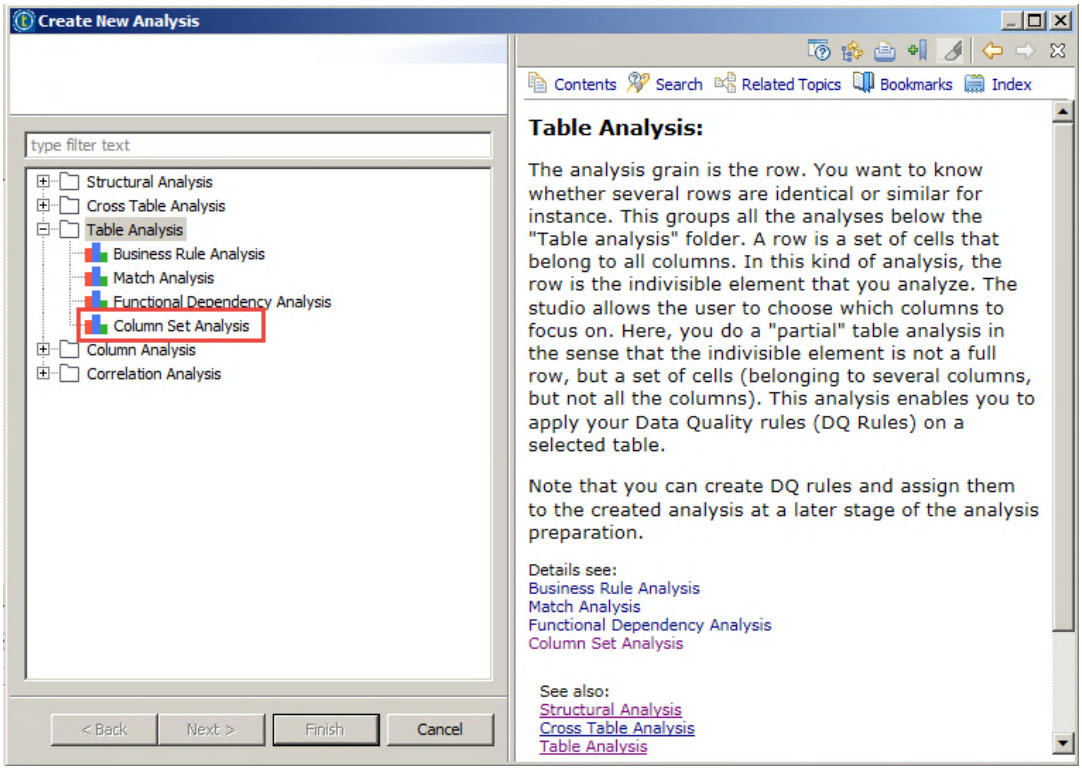
In this lesson you will set up and run a column set analysis to search for duplicates in the customer table. You will see that the number of duplicates depends on the columns selected. Unlike the basic column analysis, which reports individual statistics for each column selected, the column set analysis reports statistics for the full column set.

Column set analysis

1. CREATE THE ANALYSIS

   Right-click the **CRM_Analyses** folder.

   a. Click **New Analysis**, expand **Table Analysis**, and click **Column Set Analysis**.

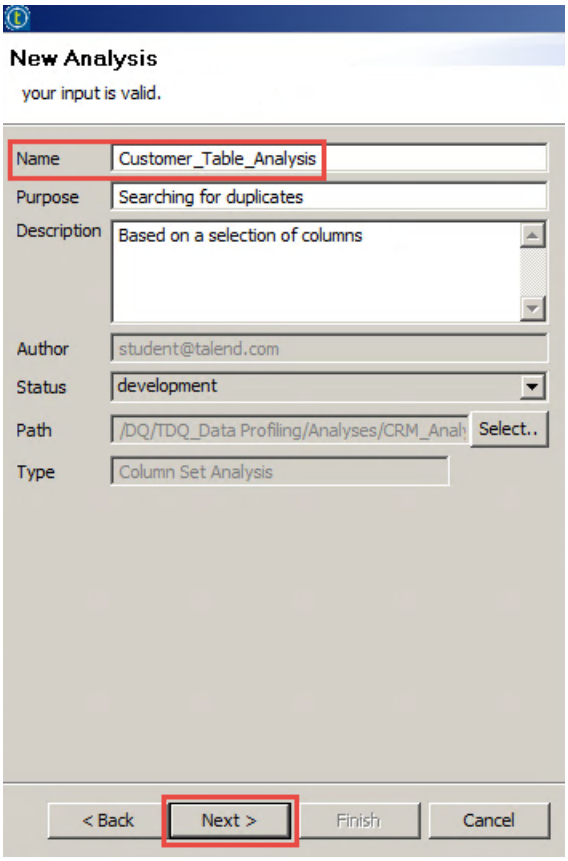


   b. Read the context-sensitive help, then click **Next**.

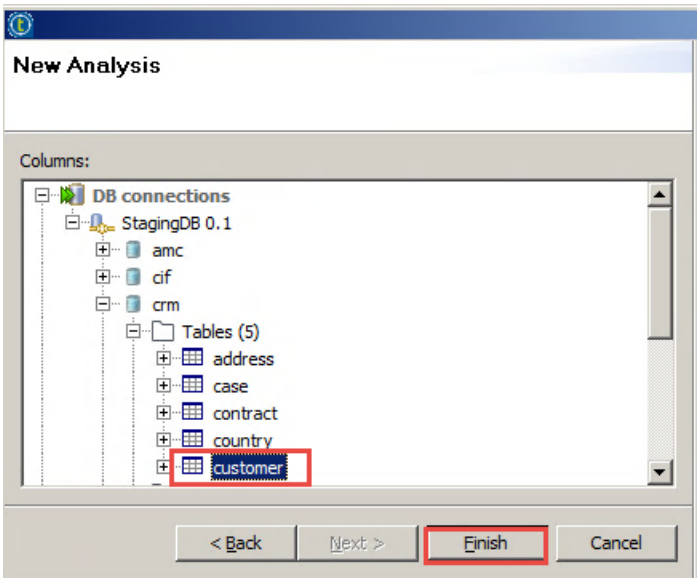
   c. In the **Name** text box, enter *Customer_Table_Analysis*

   d. Fill in the **Purpose** and **Description** boxes (optional, but a best practice).

   e. Click **Next**.



f. Expand **DB connections**, then **StagingDB**, and in the **crm** catalog, select the **customer** table. Click **Finish**.



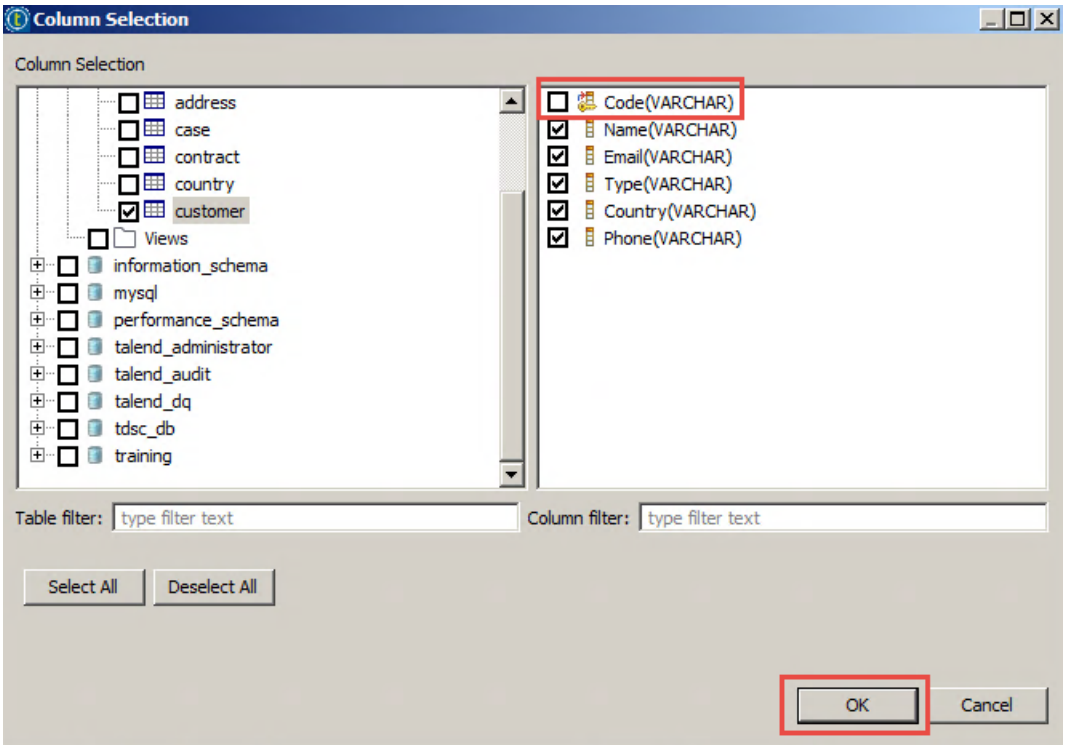
2. SET UP THE ANALYSIS

   The new analysis is open in the Profiling perspective.

   You do not need to select indicators; just select a set of columns to analyze.

a. Click the **Select Columns** button.

b. If the primary key of the table is selected, no duplicates are discovered, as the primary key is supposed to be unique. Deselect the **Code** column and click **OK**.

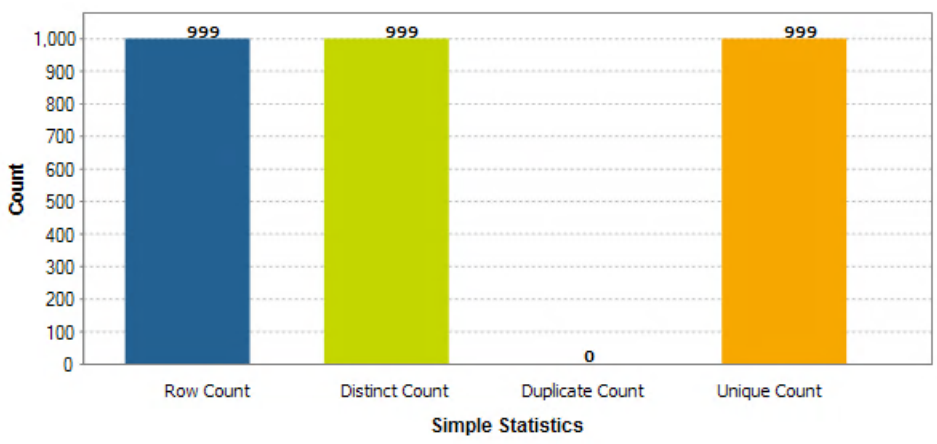


c. Click the **Run** button.

3. EXAMINE THE RESULTS

Only simple statistics are displayed on the Analysis Results tab.

At first glance, you can see that there are no duplicates in the set of selected columns.

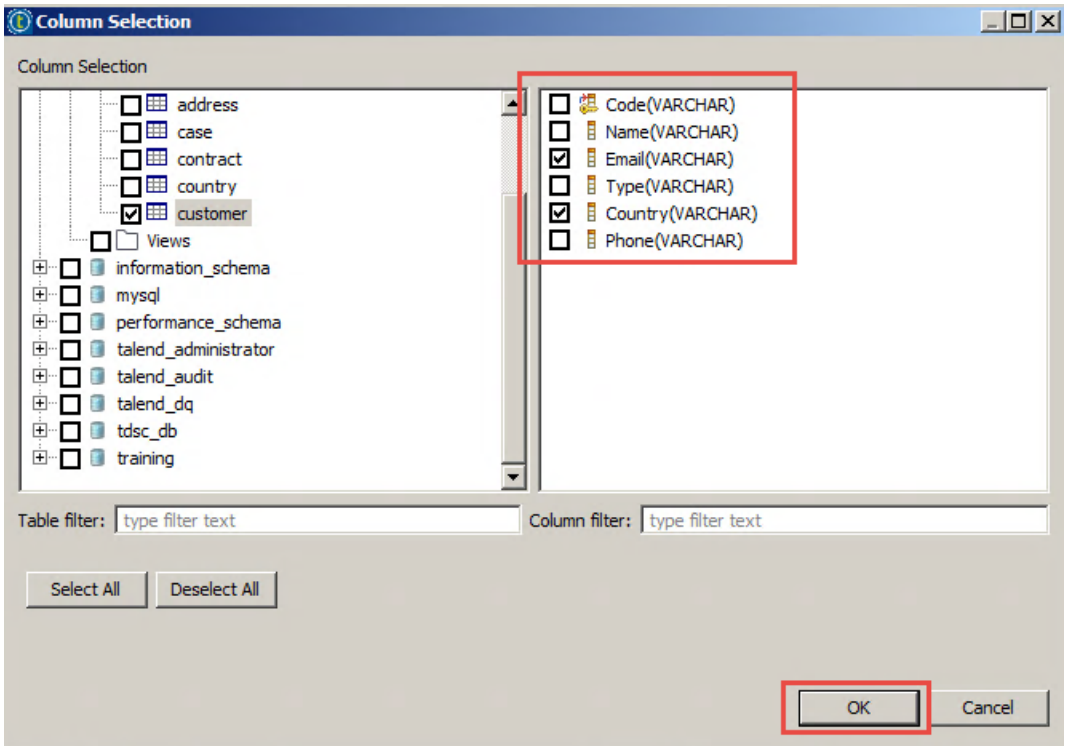


Removing some columns will make some duplicates appear.

4. CHANGE THE COLUMN SELECTION

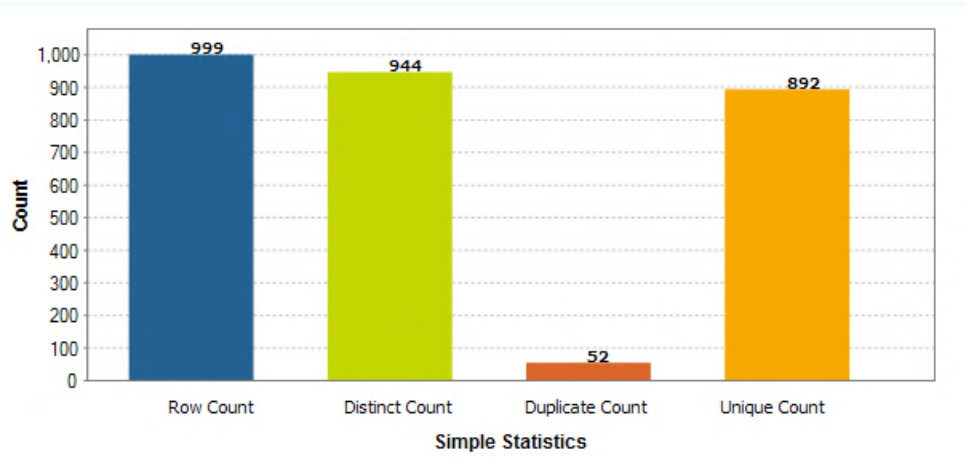
Display the **Analysis Settings** tab and click the **Select Columns** button.

a. To show how many email duplicates exist per country, deselect all columns except **Email** and **Country**, then click **OK**.



b. Run the analysis again.

This time, several duplicates appear on the chart.



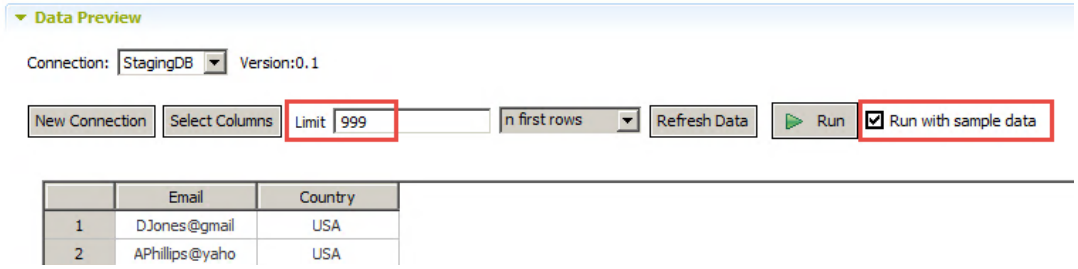
Advanced features

To avoid system overload, some features are enabled only when the analysis is run on a limited number of rows.

1. RUN THE ANALYSIS WITH SAMPLE DATA

   Display the **Analysis Settings** tab.

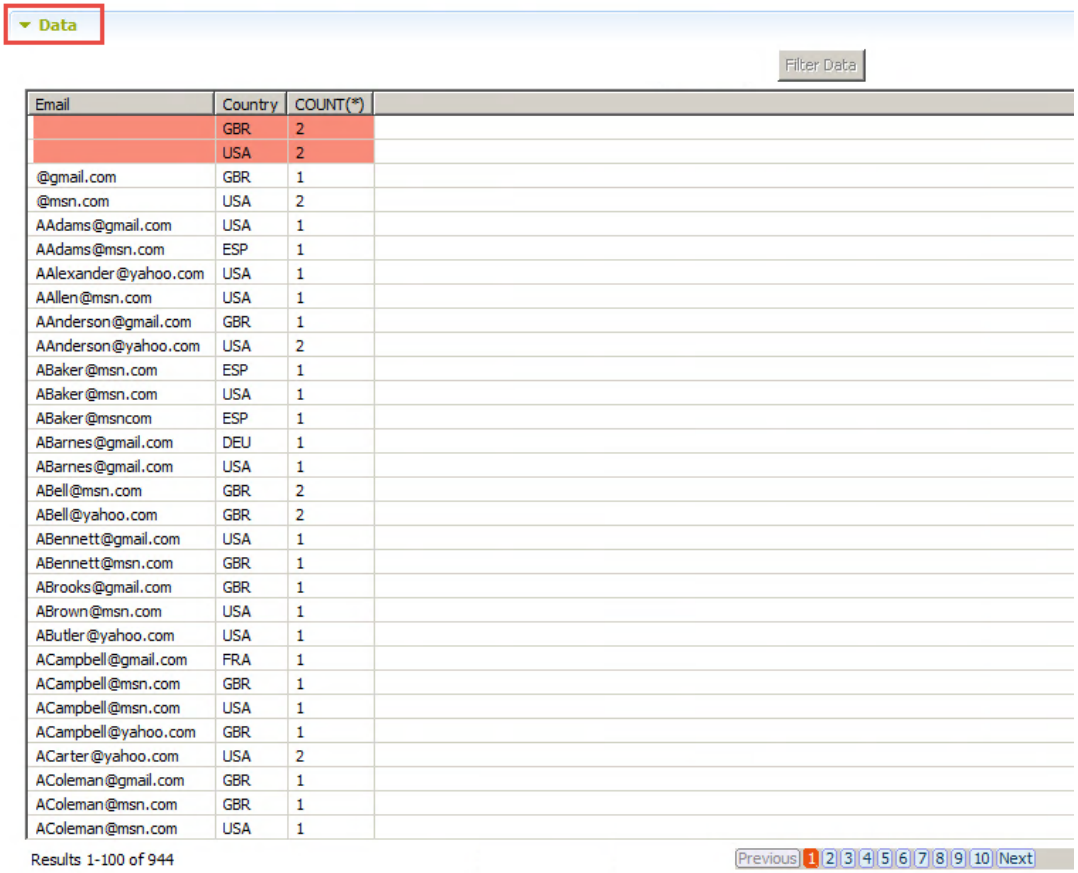
a. In the top left corner of the **Data Preview** section, select the **Run with sample data** check box. Since the training database does not contain a lot of data, set the highest limit by entering *999* in the **Limit** box.



b. Run the analysis. The Data section is now active at the bottom of the Analysis Results tab.

c. Expand the section to display the analyzed data and notice the following from the output:

- A count column was added to the column set you specified for the analysis.
- If the count is equal to 1, the column set is not a duplicate.
- A row with any missing column data (including blank rows) is highlighted in salmon.
- The total number of results corresponds to the distinct count.
- You can jump to the previous, next, or any other numbered page using the buttons in the lower right corner of the results.
- Notice that the Filter Data button is not active.

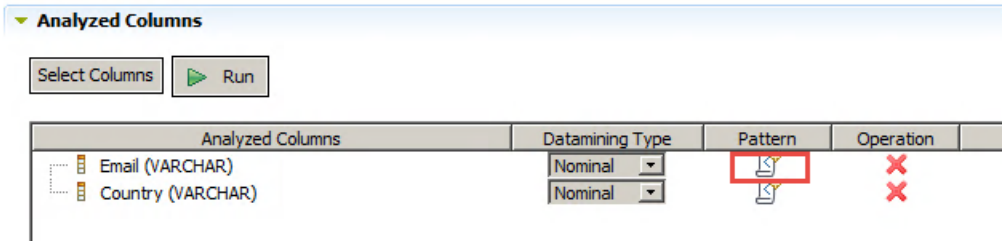


2. ADDING PATTERN CONTROL

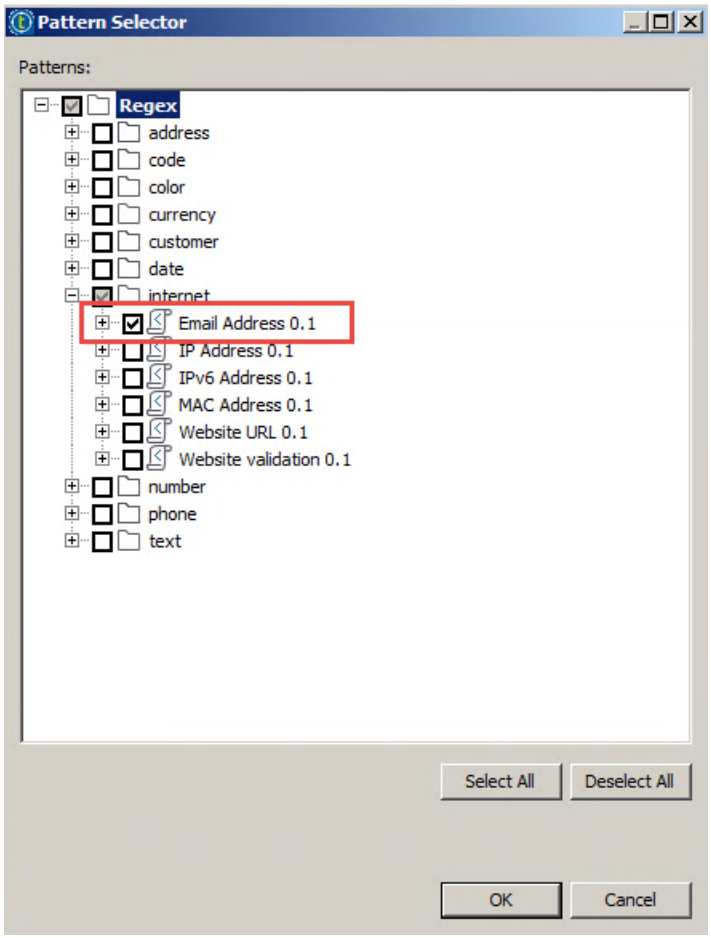
As you did for the basic column analysis, add the Email Address pattern control to the Email column.

a. Again display the **Analysis Settings** tab.

b. In the **Pattern** column of the **Analyzed Columns** list, click the **Add pattern** icon (the only one in that column). The Pattern Selection window appears.



c. Expand **Patterns**, **Regex**, and **internet**, then select **Email_Address**.



d. Click **OK**.

4. VIEW AND FILTER DATA

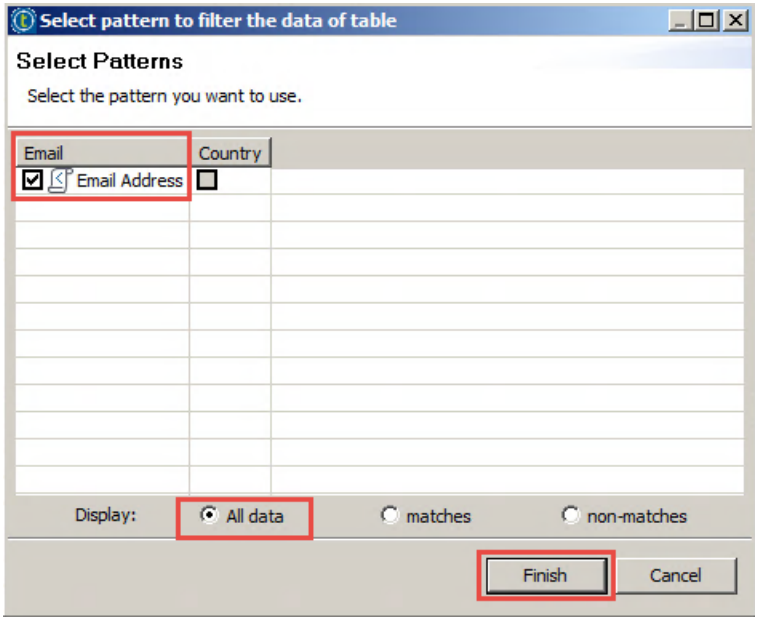
Run the analysis. In the Data section, the Filter Data button is now active.

a. Click the **Filter Data** button.

b. In the Select Patterns window, select **Email Address**. Several filter options are available:

- **All data** displays all rows, but uses specific formatting for emails that do not match the pattern
- **Matches** displays only rows with emails that match the pattern
- **Non-matches** displays only rows with emails that do not match the pattern

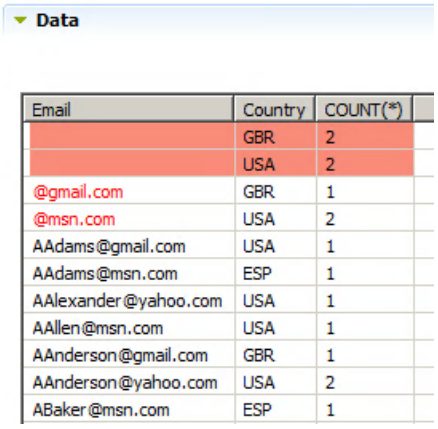
Select **All data**, then click **Finish**.



c. The output in the Data section changes.

Notice in the results shown below:

- Emails that do not match the regular expression are marked in red type.
- Rows including one or more blank columns are highlighted in salmon.



The next step is to create a business rule analysis.

Using a business rule analysis

Overview

The business rule analysis is used to check for anomalies in your data that are tied to business rules you define. You will use it to confirm the validity of the dates in the contract table.

This table contains two dates:

- Begin_dt, the start date of the contract
- End_dt, the end date of the contract

To be considered valid, the end date must be strictly greater than the start date.

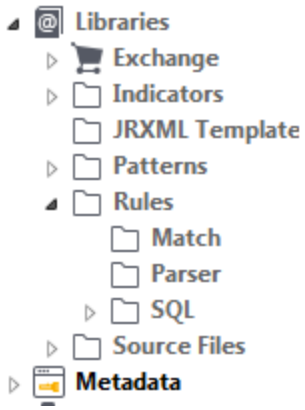
First you will create a SQL business rule in the repository. Then you will create a business rule analysis and associate the business rule with it.

For the second analysis, you will join two tables in another SQL rule to confirm the validity of a third date in the claim table. To be valid, the date of the creation of the claim (incident ticket) must fall between the contract start and end dates.

Creating a business rule analysis

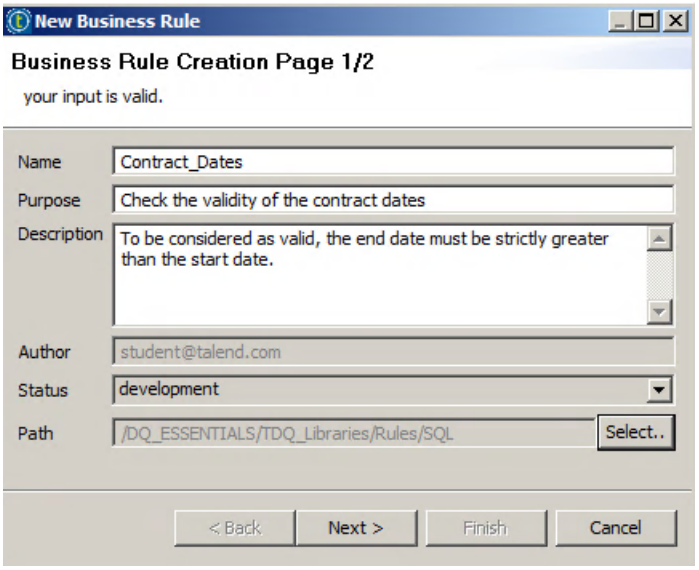
1. CREATE A BUSINESS RULE

   In the **DQ Repository**, expand **Libraries,** then **Rules.**

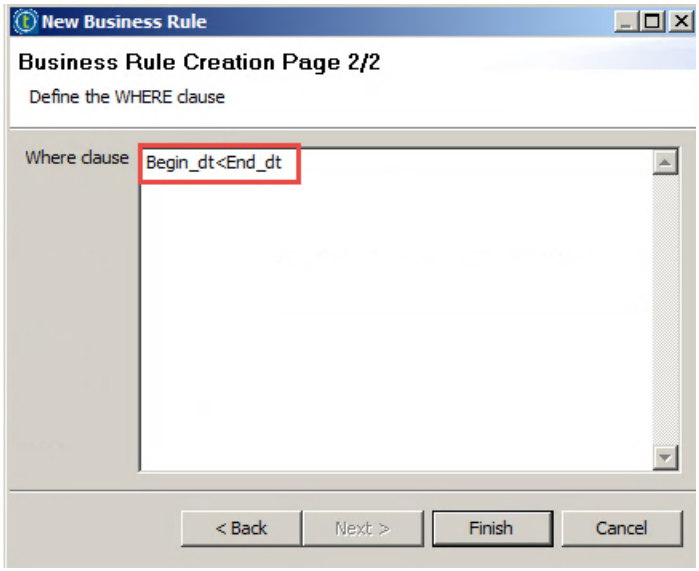


   a. Right-click the **SQL** folder and click **New Business Rule.**

      Fill in the **Name**, **Purpose**, and **Description** boxes and click **Next**.



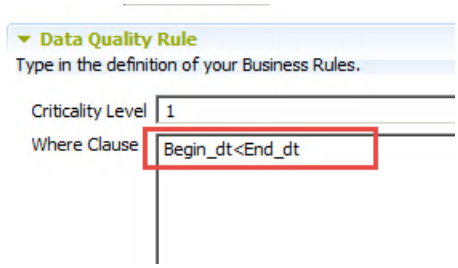
b. You must define the WHERE clause used for the business rule. Data is considered valid only if it matches or passes this clause.



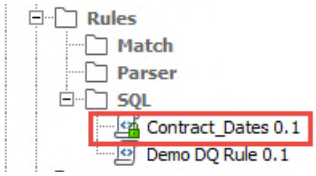
In the **WHERE clause** text box, enter *Begin_dt < End_dt*

This ensures that the value in the end date is greater than the value in the start date.

c. Click **Finish**. The new rule appears in the workspace with the WHERE clause in the Data Quality Rule area.



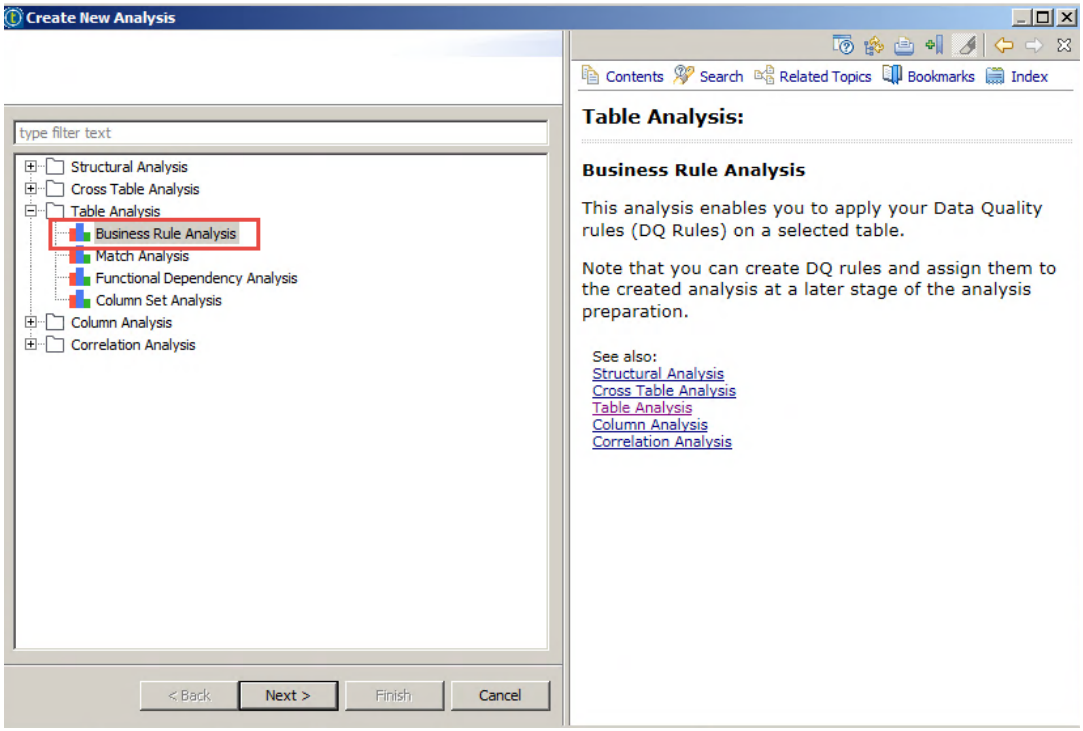
Notice that the new business rule appears in the repository.



2. CREATE A BUSINESS RULE ANALYSIS

   Right-click the **CRM_Analyses** folder and choose **New Analysis**.

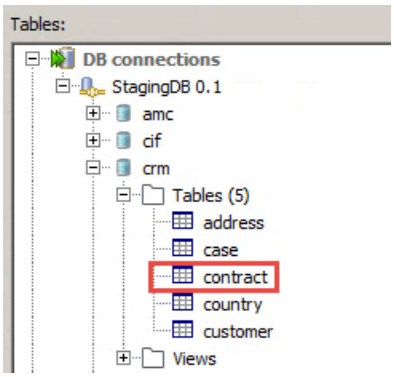
   a. Expand **Table Analysis** and click **Business Rule Analysis**.



   b. Read the context-sensitive help, then click **Next**.

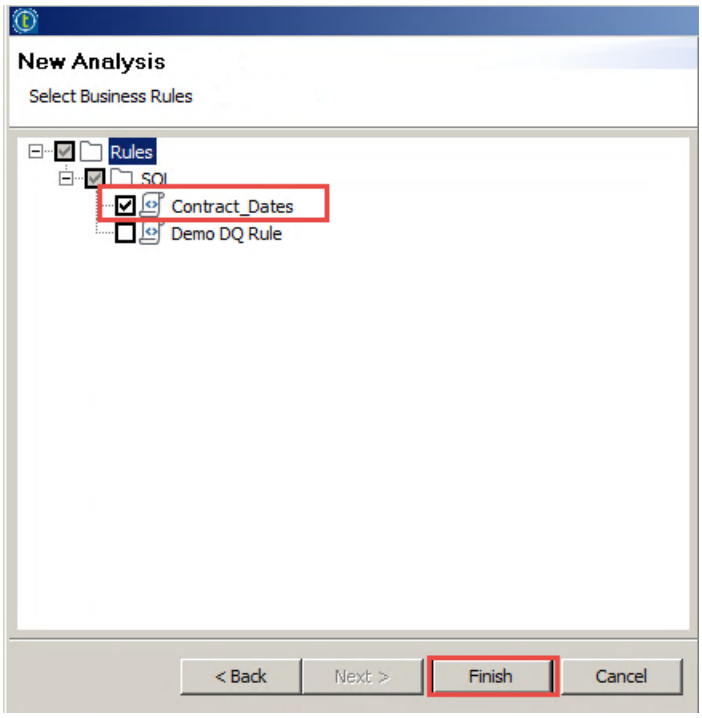
   c. Name the analysis *Contract_Dates_Analysis*. Fill in the **Purpose** and **Description** boxes, then click **Next**.

   d. To identify the table to be analyzed, extend **DB Connections**, then **crm** catalog, and click the **contract** table.



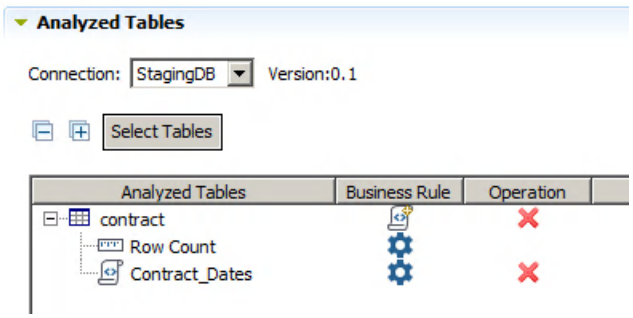
Click **Next**.

e. Expand **Rules**, then **SQL**, and select the business rule you created earlier, **Contract_Dates**.



Click **Finish**.

The Analyzed Tables section displays the information you provided earlier to create the analysis.



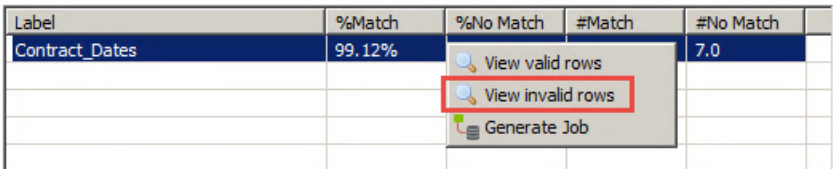
f. Run the analysis.

3. EXPORT INVALID ROWS

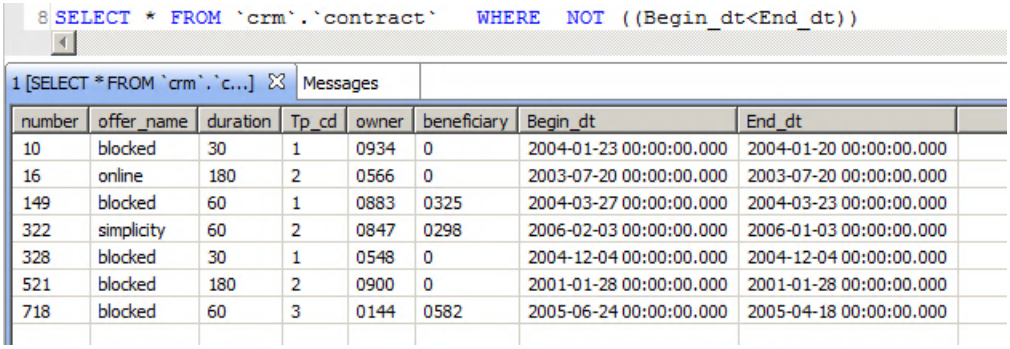
Examine the results of the analysis.

Approximately 99% of the data is considered valid and 1% is invalid.

a. Right-click the **Contract_Dates** indicator and click **View invalid rows**.



SQL Editor displays the invalid rows.

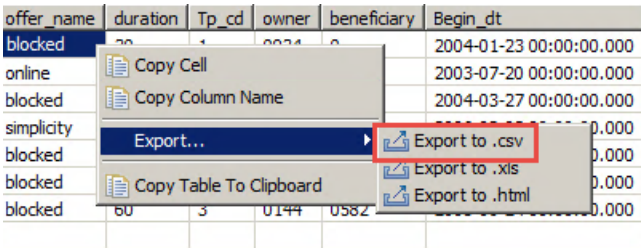


number	offer_name	duration	Tp_cd	owner	beneficiary	Begin_dt	End_dt
10	blocked	30	1	0934	0	2004-01-23 00:00:00.000	2004-01-20 00:00:00.000
16	online	180	2	0566	0	2003-07-20 00:00:00.000	2003-07-20 00:00:00.000
149	blocked	60	1	0883	0325	2004-03-27 00:00:00.000	2004-03-23 00:00:00.000
322	simplicity	60	2	0847	0298	2006-02-03 00:00:00.000	2006-01-03 00:00:00.000
328	blocked	30	1	0548	0	2004-12-04 00:00:00.000	2004-12-04 00:00:00.000
521	blocked	180	2	0900	0	2001-01-28 00:00:00.000	2001-01-28 00:00:00.000
718	blocked	60	3	0144	0582	2005-06-24 00:00:00.000	2005-04-18 00:00:00.000

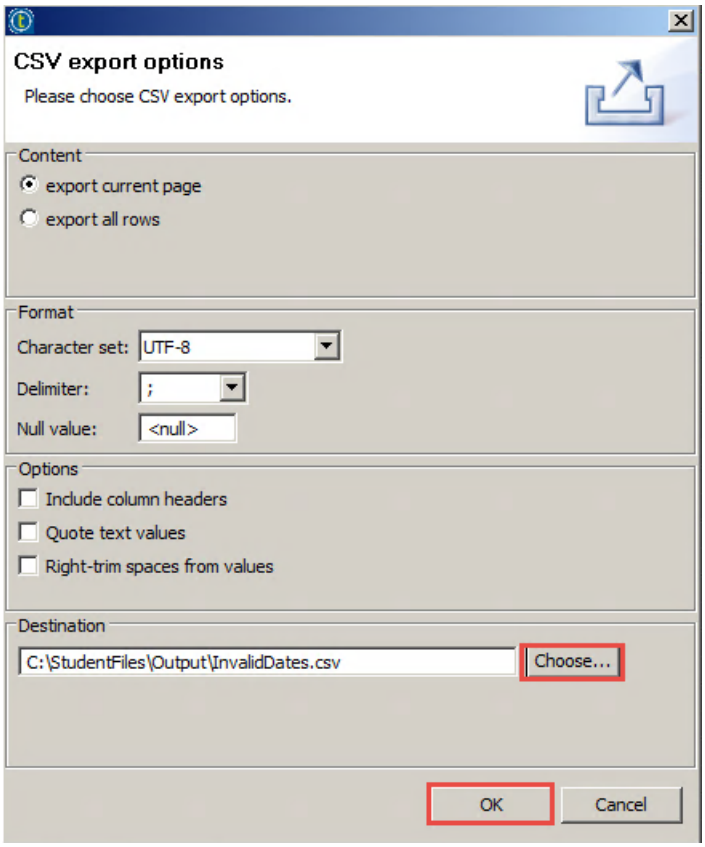
Notice the **Begin_dt** and **End_dt** columns. The values in Begin_dt are greater than (or equal to) the values in End_dt, making them invalid, according to your business rule.

Notice how the WHERE clause of the SQL statement reuses the business rule you created.

b. To export the results, right-click anywhere in the table, click **Export**, and click **Export to .csv**.



c. Do not modify anything in the CSV export options except the **Destination** path.
Click **Choose** and navigate to *C:\StudentFiles\Output*. Name the file *InvalidDates* and click **Save**.



d. In the CSV export options window, click **OK**.

e. Use Windows Explorer to locate and open the file. The content should be the same as what SQL Editor displays (with a semicolon field delimiter).

Using a join in a SQL rule

1. CREATE A NEW SQL BUSINESS RULE

   Use the same process to create a new SQL rule.

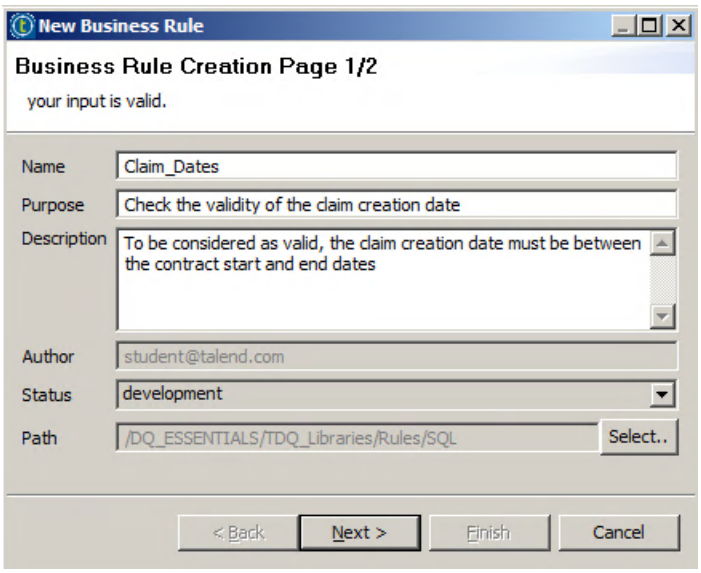
   Name the rule *Claim_Dates* and use WHERE clause *Date >= Begin_dt and Date <= End_dt*

   Date is a date column in the claim table, and Begin_dt and End_dt are the two date columns in the contract table.

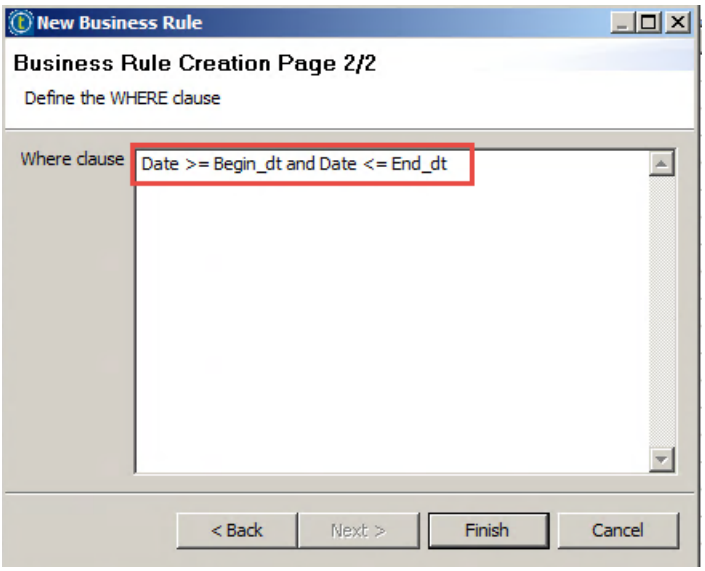
   a. In the **DQ Repository**, expand **Libraries,** then **Rules.**

      Right-click the **SQL** folder and click **New Business Rule.**

      Fill in the **Name**, **Purpose**, and **Description** boxes and click **Next**.



b. In the **Where clause** text box, enter *Date >= Begin_dt and Date <= End_dt*

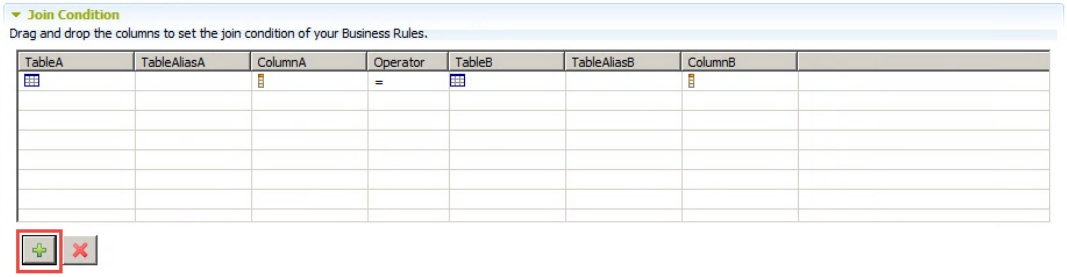


c. Click **Finish**.

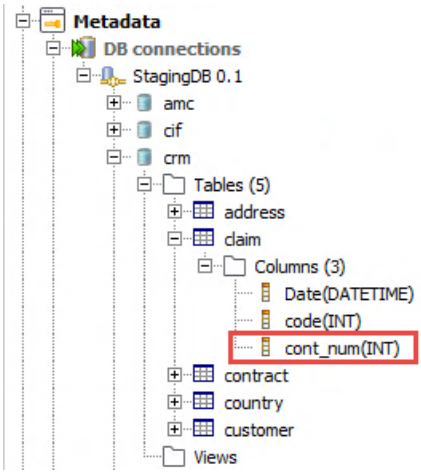
2. JOIN THE TWO TABLES

The SQL rule uses fields from the contract and claim tables. A join must be created on the contract number, which is present in the two tables.

a. To add a condition, below the Data Quality Rule section, in the Join Condition section, click the green **plus symbol (+)**.

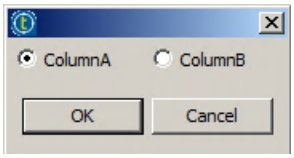


b. In the **Repository**, locate the **cont_num** column in the **claim** table.



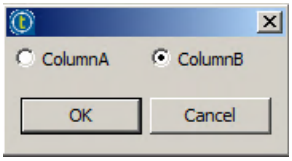
c. Drag the **cont_num** column into the first row of the **Join Condition** table.

d. Select **ColumnA** and click **OK**.

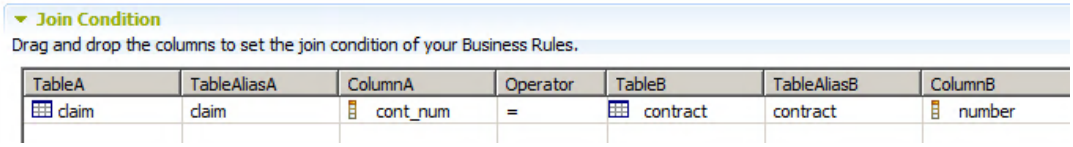


e. Following the same process, drag the **number** column from the contract table into the first row of the **Join Condition** table.

f. Select **ColumnB** and click **OK**.



The two tables are now joined by their common key.



g. Save the rule by clicking the **disk icon**.

3. CREATE A NEW BUSINESS RULE ANALYSIS

   Following the steps at the beginning of this lesson, create a new business rule analysis.

   You must name it **Claim_Dates_Analysis** and reuse the **Claim_Dates** business rule.

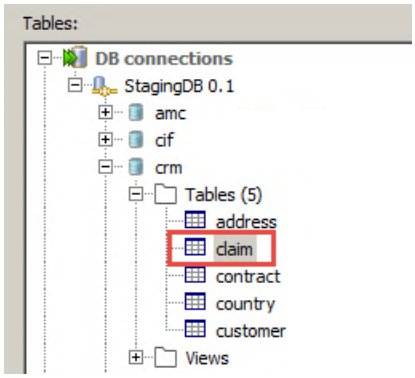
   a. Right-click the **CRM_Analyses** folder and choose **New Analysis**.

   b. Expand **Table Analysis** and click **Business Rule Analysis**.

   c. Read the context-sensitive help and click **Next**.

   d. Name the analysis *Claim_Dates_Analysis*. Fill in the **Purpose** and **Description** text boxes and click **Next**.

   e. In **DB connections**, in the **crm** catalog, click the **claim** table.



   Click **Next**.

   f. Expand **Rules**, then **SQL**, and select the business rule you created earlier, **Claim_Dates**.

   Click **Finish**.

4. EXAMINE THE RESULTS

   Run the analysis

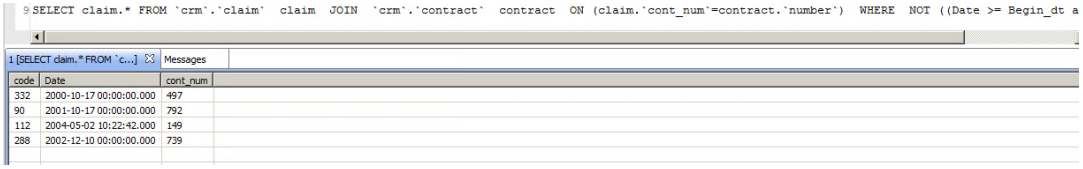
   a. Examine the **Claim_Dates** indicator. Four rows were not validated by the business rule.

Label	%Match	%No Match	#Match	#No Match
Claim_Dates	96.00%	4.00%	96.0	4.0

   b. Right-click the results table and click **View invalid rows**.

   c. SQL Editor displays the invalid rows.

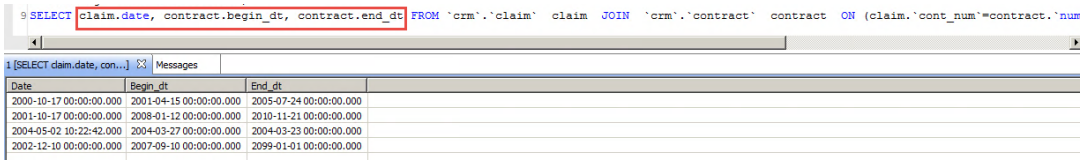
   Notice how the WHERE clause of the SQL statement reuses the business rule you created.



   d. You can easily compare dates by updating the statement. Replace the first part of the select query with the following statement:

   SELECT claim.date, contract.begin_dt, contract.end_dt

   On the toolbar of SDL Editor, click the **Run** button.



The date listed for each claim in the four invalid rows is not within the start and end dates of the contract.

Read the lesson wrap-up before continuing.

Wrap-Up

In this lesson you performed a column set analysis, which is similar to a basic column analysis but with statistics based on rows (sets of columns you specify) rather than individual columns.

You learned how to create business rules in the DQ repository, and how to reuse them in a business rule analysis.

Then you used SQL Editor to export invalid rows to a CSV file in your local file system.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

LESSON 4

Cross-Table Analysis

This chapter discusses the following.

Cross-table analysis

Lesson overview

The Cross Table Analysis category contains only one type of analysis: redundancy. A redundancy analysis can compare the data in two columns (or two sets of columns) in two tables. People most commonly use this type of analysis to verify foreign key/primary key relationships between two tables.

Objectives

After completing this lesson, you will be able to:

- Create a redundancy analysis
- Use the analysis to discover primary key/foreign key relationships between two tables
- Configure the analysis to include an SQL Where clause to qualify the results

The first step is to create a redundancy analysis.

Using redundancy analysis

Overview

This exercise compares the relationship between columns from two tables as part of a redundancy analysis.

After completing this exercise, you will have the opportunity to perform a redundancy analysis on your own.

Check foreign keys

In this exercise, you will compare the country codes in the customer table with the countries available in the country table. The country table contains descriptive information (for example, name and currency) for a long list of countries. The country codes are in the three-letter ISO format. To be able to correctly reuse this data, you must ensure that all the country codes used in the customer table exist in the country table.

You have created several other types of analysis, so some of this will be familiar.

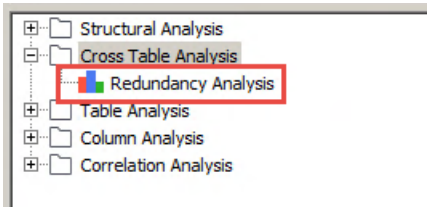
Note: In this course, you can view brief, default summary instructions, or you can expand the sections to see complete, detailed instructions that often include screenshots.

1. CREATE THE ANALYSIS

   In the **Repository**, in the **CRM_Analyses** folder, create a new redundancy analysis. Name it *Country_Match_Keys*.

   a. Right-click the **CRM_Analyses** folder and select **New Analysis**.

   b. Expand the **Cross Table Analysis** category, click **Redundancy Analysis**, and click **Next**.

   c. In the **Name** text box, enter *Country_Match_Keys*

   d. As a best practice, fill in the **Purpose** and **Description** boxes.

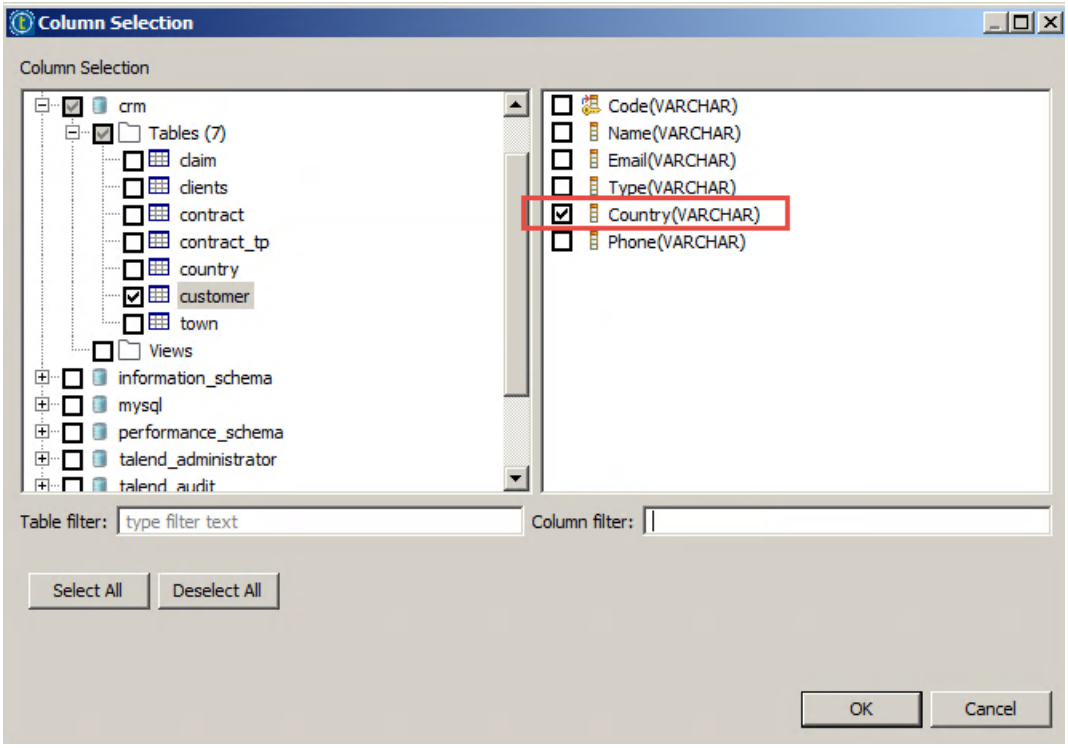
   e. Click **Finish**.

2. SET UP THE ANALYSIS

   The Analyzed Column Sets section is where you select columns to compare.

   Note: Columns must be from two different tables (labeled A and B on your screen). In this exercise, you will select only one column (the country code) per table, but keep in mind that you can compare a set of columns (for example, the country code plus the currency) if the columns belong to tables A and B (sometimes the columns have different names).

a. To select the first column in the analysis, click the **A Column Set** button (below Left Columns).

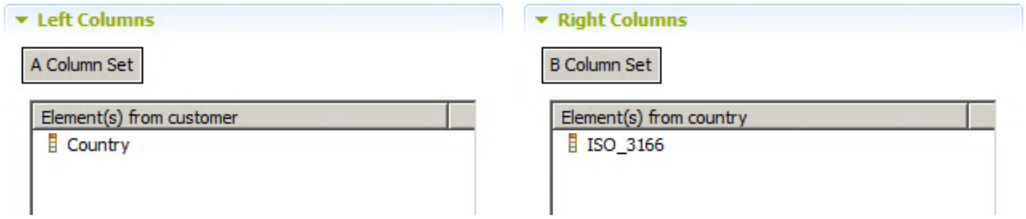


b. Expand the **crm** catalog, then locate and select the **Country** column in the customer table.

c. Click **OK**.

The Country column is now selected in the Left Columns list.

d. Click the **B Column Set** button (below Right Columns) and use the same process to select the **ISO-3166** column in the country table in the **Right Columns** list.

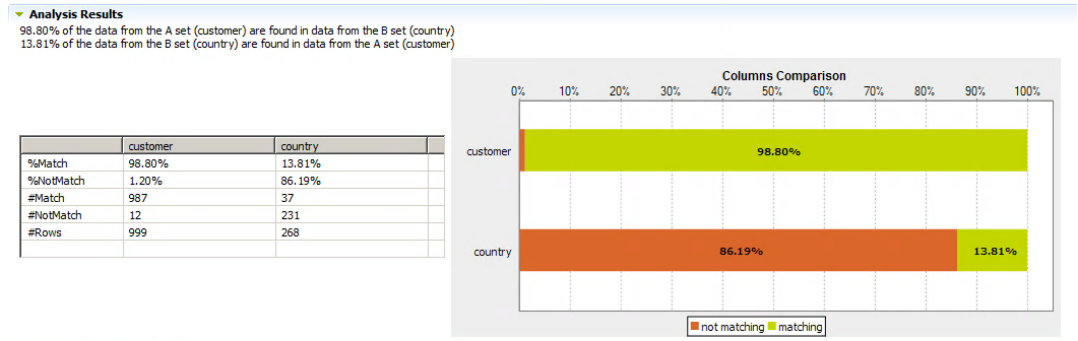


3. RUN THE ANALYSIS

To run the analysis, click **Run**.

a. In the **Confirm your operation** warning message window about left joins, click **OK** (the training tables contain a reasonable number of records).

b. Examine the results.



The table and chart show the percentages and total numbers of records that match or do not match, as well as the total number of rows examined in both the customer and country tables.

Many country codes from the country table are not appearing in the customer table. This is OK, as the country table contains almost all the countries in the world.

A few country codes from the customer table do not appear in the country table. You can examine these codes in SQL Editor to find out why they do not have matches in the country table.

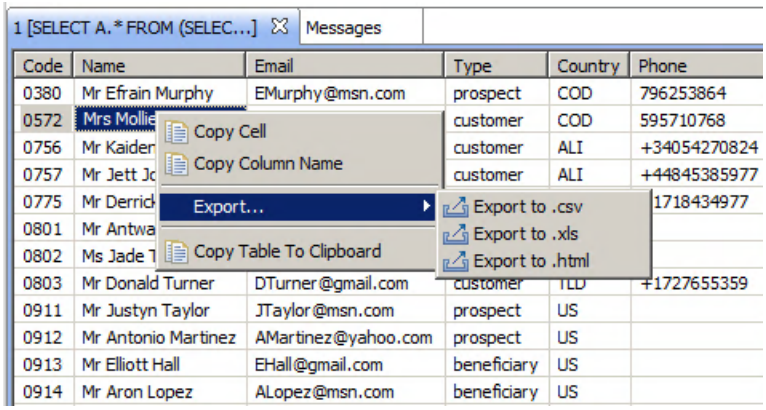
4. USE SQL EDITOR

In SQL Editor, display the **#NotMatch** rows.

a. Right-click somewhere in the customer column in the table (or in the customer bar on the chart). On the contextual menu, choose **View not match rows**.

The #NotMatch rows from the customer table appear in SQL Editor. Look at the **Country** column. Either the codes are in the wrong format (for instance, they have two characters instead of three), or they are not known country codes.

b. Right-click somewhere on the list and view the options on the contextual menu.



You can copy a cell or the full table to the clipboard. You can also export this data in TXT, CSV, or HTML format.

You have completed this lesson and can move on to the exercises.

Challenge

Complete this exercise to further explore the use of redundancy analysis. For one way to do this, see the solutions.

Customers and contracts

In this analysis, you will compare the Owner column of the contract table to the primary key of the customer table to find out if all the contract owners exist in the customer table.

Create a redundancy analysis and name it *contract_owners*. Set the **Code** column of the customer table in the **Left Columns**. Set the Owner column of the contract table in the **Right Columns**.

Run the analysis and examine the results.

1. Look at the **Owner** column. How many records do not match? Why do they not match?
2. Look at the **Code** column. The results seem unbalanced. Take a closer look. Again display the **Analysis Settings** tab and see if you can determine how to improve the analysis.

Solutions

Here is a solution to the challenge. Your version may be slightly different but still valid.

Customers and contracts

The challenge was to create a new redundancy analysis with left and right columns as dictated in the exercise. Here are the answers:

1. Only two records do not match (in a total of 800 rows). Display the two nonmatching rows in SQL Editor. You can see that the Owner column is empty; data is clearly missing.
2. According to the analysis results, a majority of customers do not have a contract. Use SQL Editor to study the nonmatching rows. Take a look at the Type column. Most of the customers without contracts are prospects (the type can have one of three values: customers, beneficiary, prospect).

   It is possible to enhance this analysis by filtering the customer type: in the **Data Filter** section, in the **Where** clause located below the Left Columns section, simply type *customer.type = "customer"*

   Run the analysis again.
   The results are much more balanced, with only 10 percent of customers not having a contract. In the **Contract** column, the number of nonmatching rows has increased a little (owners of those contracts are prospects or beneficiaries).

Read the lesson wrap-up before continuing.

Wrap-Up

In this lesson you learned how to compare column data in one table with column data in another table.

You have used the redundancy analysis to check the validity of the foreign keys in the customer table. Then you have used SQL Editor to visualize the #NotMatch keys.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

LESSON 5

Advanced Matching

This chapter discusses the following.

Advanced matching

Lesson overview

In this lesson, you will set up and run a match analysis to identify duplicates in the address table of the CRM database.

In the column set analysis material, you started learning about duplicate detection.

The match analysis is a more thorough solution available in the Table Analysis category. It provides:

- More available parameters: you can create blocks to optimize the matching process, you have several algorithms available for comparing values, and you can set up thresholds
- More-accurate results: duplicates are grouped according to their degree of likeness

In the first exercise, you will explore the available match analysis settings and fine-tune them for the best output. You will complete the analysis and export all the settings in a match rule.

Then you will create an integration Job to export unique, suspect, and match rows in three separate outputs. To retrieve the analysis settings, you will import the match rule into the matching component of the Job (tMatchGroup).

Creating a matching integration Job gives you access to additional possibilities, and you will see that linking several matching components can enhance output quality.

Objectives

After completing this lesson, you will be able to:

- Create and run a match analysis
- Configure and rerun a match analysis, observing the different results
- Export match analysis settings in a match rule
- Create an integration Job using matching components

Before using the analysis, you will examine the address table.

Getting ready for match analysis

Overview

In this lesson, you will reuse the connection overview analysis and create a column set analysis to explore the content in the address table of the CRM database.

Customer addresses

The address table of the CRM database stores customer mailing addresses.

These addresses were populated in several ways:

- Customers manually entered them on the Web site.
- Customers updated addresses and created additional addresses without removing the existing ones (a customer can have several addresses).
- When support technicians called in response to claims, they updated or created new customer addresses.

The lack of control in this process led to bad data capture and duplicated addresses.

Connection overview analysis

To learn more about the address table and its relationship to the customer table, you can reuse the connection overview analysis you created in the first lesson.

1. OPEN THE CONNECTION OVERVIEW ANALYSIS

   Open the first analysis you created in the CRM_Analysis folder.

   a. In the **Profiling** perspective, expand **DQ Repository**, **Analysis**, and **CRM_Analysis**.

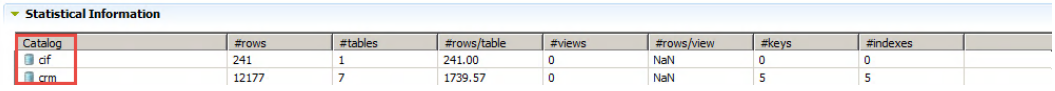
   b. Double-click **Database_Server_Connection_Analysis**.

   c. Display the **Analysis Results** tab, which shows the latest run of the analysis. (You can rerun it by clicking the **Run** icon or pressing **F6**.)

2. DISPLAY ADDRESS TABLE DATA IN THE DATA EXPLORER PERSPECTIVE

   You can directly access the Data Explorer perspective from the analysis results.

   a. In the **Statistical Information** section of the **Analysis Results** tab, click the **crm** catalog.



Statistical Information

Catalog	#rows	#tables	#rows/table	#views	#rows/view	#keys	#indexes
dif	241	1	241.00	0	NaN	0	0
crm	12177	7	1739.57	0	NaN	5	5

   The list of tables appears, displaying the number or rows, keys, and indexes in each table.

g

Table	#rows	#keys	#indexes
address	967	1	1
claim	100	0	0
contract	800	1	1
country	268	1	1
customer	999	1	1

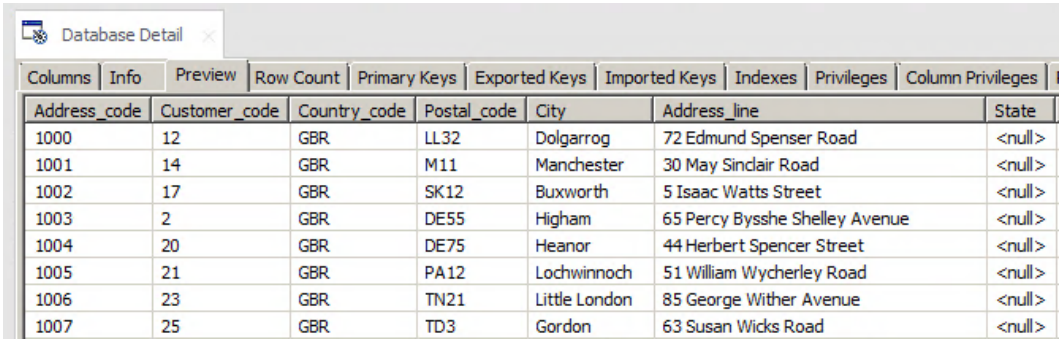
   b. Right-click in the **address** row, then click **View keys**. The display changes in several ways.

      By clicking **View keys**, you change the application to the Data Explorer perspective.

   The Database Detail view shows basic information in the Address table. Take a look at the following tabs:

   - **Primary Keys** shows that the primary key of the Address table is the Address_code column.
   - **Columns** shows the properties of the columns of the table.

» **Preview** displays the first rows of the table.



Address_code	Customer_code	Country_code	Postal_code	City	Address_line	State
1000	12	GBR	LL32	Dolgarrog	72 Edmund Spenser Road	<null>
1001	14	GBR	M11	Manchester	30 May Sinclair Road	<null>
1002	17	GBR	SK12	Buxworth	5 Isaac Watts Street	<null>
1003	2	GBR	DE55	Higham	65 Percy Bysshe Shelley Avenue	<null>
1004	20	GBR	DE75	Heanor	44 Herbert Spencer Street	<null>
1005	21	GBR	PA12	Lochwinnoch	51 William Wycherley Road	<null>
1006	23	GBR	TN21	Little London	85 George Wither Avenue	<null>
1007	25	GBR	TD3	Gordon	63 Susan Wicks Road	<null>

From here, you may recognize some codes:

» **Country_code** is the primary key in the Country table

» **Customer_code** is the primary key in the Customer table

The other columns contain the elements needed to form a complete postal address: an address line, city, and postal code.

Additional information not shown by the analysis:

» The address table contains addresses from only five countries: the United Kingdom, United States, Germany, Spain, and France.

» The State column cells are populated only with addresses within the US, otherwise they are empty.

Before continuing, switch to the Profiling perspective.

Column set analysis

1. CREATE AND RUN A COLUMN SET ANALYSIS OF THE ADDRESS TABLE

   Note: You already created a column set analysis of the customer table in the "Table Analysis" lesson.

   a. Right-click the **CRM_Analyses** folder. Click **New Analysis**, expand **Table Analysis**, and click **Column Set Analysis**.

   b. Fill in the **Name**, **Purpose**, and **Description** fields, then click **Next**.

   c. Expand **DB connections**, **StagingDB**, and **crm**, then select the **address** table. Click **Finish**.

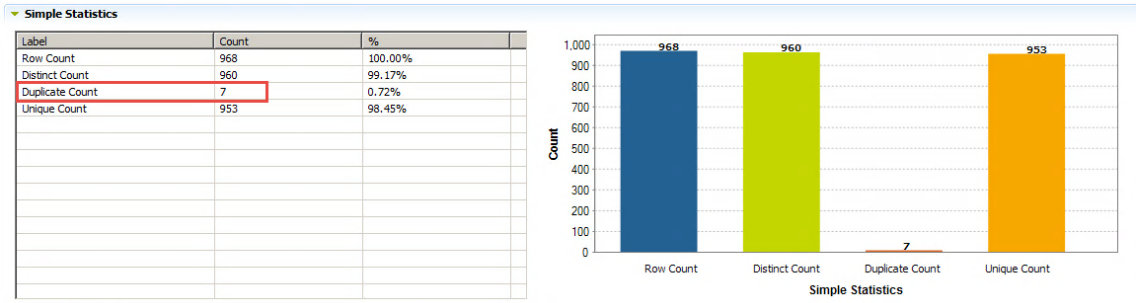
      The new analysis appears in the Profiling perspective.

   d. Click the **Select Columns** button.

   e. Because no duplicates are discovered if the primary key is selected, deselect the **Address_code** column. Click **Ok**.

   f. To launch the analysis, click the **Run** button.

2. EXAMINE THE RESULTS OF THE COLUMN SET ANALYSIS



Despite the lack of control when creating addresses and potential for incorrect address creation, only a few duplicates were identified by the column set analysis. This is because a column set analysis considers two column sets duplicates only if they are *identical*.

In addition, the analysis does not mention whether match records are duplicated once, twice, or more.

In the next chapter, you will use a match analysis that can produce much more detailed output.

Reviewing the match analysis process

Overview

This lesson covers how match analysis works. It is intended to give you enough knowledge to create and configure a table match analysis. You do not need to practice until the next lesson.

The output of a match analysis

For the same selected columns, match analysis produces more-detailed information than column set analysis.

- Match analysis compares rows. It classifies them by similarity, examining those that are strictly similar as well as those that look alike.

  The comparison is done only on selected columns. Comparison algorithms are used to estimate whether two records are similar.

Input Column	Matching Function
City	Jaro-Winkler
Address_line	q-grams

  If content is similar enough, the match analysis identifies misspelled words, incomplete records, and reverse-ordered addresses as potential duplicates.

- The analysis sorts data into groups of similar rows, and you can see how many times a row is duplicated.

  Each group is composed of a potential master record—also called a survivor or golden record—plus all the similar rows.

City	Address_line	MASTER	GRP_SIZE
Guadalupe	88 Black Elk Drive	true	3
Guadalupe	Black Elk Drive 88	false	0
Guadalupe	88 Black Elk Drive	false	0
San Antonio	21 Cate Chopin Drive	true	2
San Antonio	21 Kate Chopin Drive	false	0

- The match analysis computes the level of similarity between rows. If rows are not similar enough, they are considered *suspect*.

Duplicate Record Statistics

Label	Count	%
Row Count	967	100.00%
Unique Records	906	93.69%
Matched Records	49	5.07%
Suspect Records	12	1.24%

  You can output three separate lists of rows: unique, matched, and suspect.

  A data steward must manually process the list of suspect rows to determine whether records are duplicates.

How match analysis works

A match analysis compares rows and assigns a similarity score to each combination of rows.

- The analysis can use a different match function (or algorithm) for each analyzed column to determine if values are unique or similar.
- The similarity scores between rows are computed by combining the outputs of the column algorithms.

Then the analysis compares the similarity scores with two thresholds, which are configured when setting up the analysis:

- The match threshold, which defines whether or not rows are similar
- The confident match threshold, which defines whether the rows are similar enough to be considered duplicates, or if they must be classified simply as suspects

The confident match threshold is always higher than the match threshold.

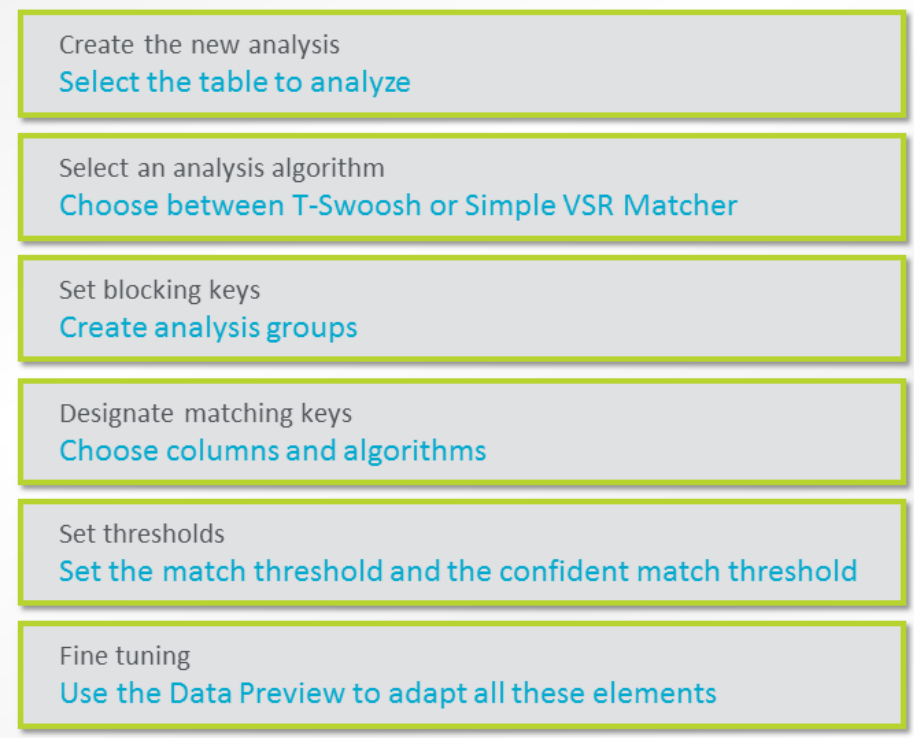
Here is how the analysis compares the similarity score with these two thresholds:

- If the similarity score is below the match threshold, the rows are not considered similar.
- If the similarity score is above the match threshold but below the confident match threshold, the rows are incorporated in the same group and considered suspect.
- If the similarity score is above the confident match threshold, the rows are incorporated in the same group and considered duplicates.

Note: A row cannot belong to several groups. If a single row is similar enough to be included in two groups, it is added to the group with the highest similarity score.

The match analysis set-up process

Setting up a match analysis is more complex than setting up other types of analysis you have used in this lab, so before you start, it is a good idea to review all the steps. You will put that into practice in a moment.



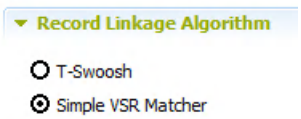
1. Create a new analysis and select data to analyze.

   As with other types of analysis, browse the categories and select the analysis type. Provide an analysis name, purpose, and description.

   Select the table to analyze.

   Note: Like the other types of analysis in the Table Analysis category, the match analysis is always set up for a single table.

2. Select an analysis algorithm: T-Swoosh or Simple VSR Matcher.

   These algorithms use dissimilar methods to compare rows. They produce different numbers of duplicates, so the composition of the groups is different.

Another difference is that the Simple VSR Matcher method designates one row as master of the group whileen T-Swoosh creates a brand new row for the master (which is then called the survivor). You can configure the values used to create this new row by using the survivorship rules.

In this lab, you will use both Simple VSR Matcher and T-Swoosh.

3. Set blocking keys.

   The analysis runs faster if data is already divided into blocks. A column can be selected as a blocking key to filter data into the different blocks. The analysis searches for duplicates inside each block; rows from different blocks are not compared. This step is optional.

4. Designate matching keys.

   You must create a match rule for the analysis:

   - Choose the columns used to compare rows.
   - Set a match algorithm for each selected column.

   Select only columns that contain relevant data to be compared. Do not select the primary key.

   Several algorithms are available for comparing data. Some of them phonetically compare similarities between strings. Others compute the distance between two strings (the number of edits needed to change one string to another).

   In this lab, you will use different kinds of algorithms.

   When setting up matching keys, you can assign key weight to more importance or less importance in the computation of the similarity score.

   Note: It is possible to create several match rules for the same analysis. Match rules are applied in succession: rows defined as single records by the first rule are analyzed by the second one.

5. Set thresholds

   Set the match threshold and confident match threshold for the analysis.

   If you are using the T-Swoosh method, there is also an individual threshold by matching key.

   If you created several match rules, the match threshold is rule-specific while the confident match threshold is applied at the end, after all the rules.

6. Use the Data Preview feature in the wizard to fine-tune all of these elements. To refine the results, you may need to run the analysis several times.

You are ready to create a table match analysis.

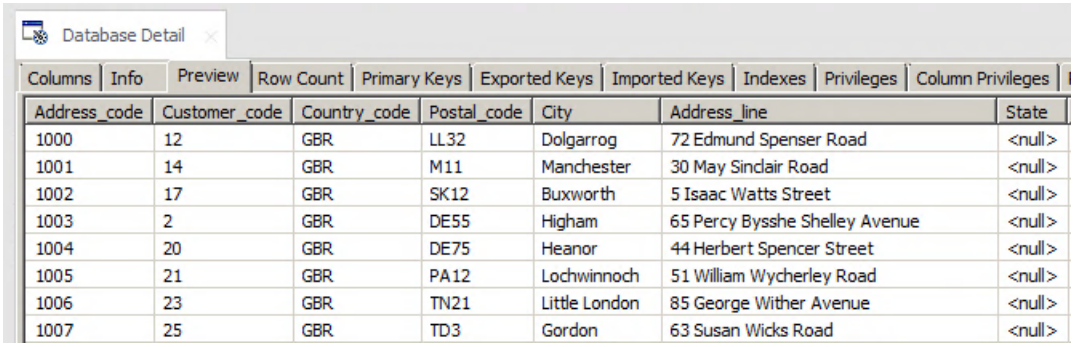
Performing a match analysis

Overview

It is time to implement what you have learned. Using the Simple VSR algorithm, you will set up a match analysis to identify duplicates in the address table.

The goal is to identify how many duplicate addresses exist per customer.

Remember what you learned from the connection overview analysis. An address is composed of four elements: the address, city, state, and postal code.



Address_code	Customer_code	Country_code	Postal_code	City	Address_line	State
1000	12	GBR	LL32	Dolgarrog	72 Edmund Spenser Road	<null>
1001	14	GBR	M11	Manchester	30 May Sinclair Road	<null>
1002	17	GBR	SK12	Buxworth	5 Isaac Watts Street	<null>
1003	2	GBR	DE55	Higham	65 Percy Bysshe Shelley Avenue	<null>
1004	20	GBR	DE75	Heanor	44 Herbert Spencer Street	<null>
1005	21	GBR	PA12	Lochwinnoch	51 William Wycherley Road	<null>
1006	23	GBR	TN21	Little London	85 George Wither Avenue	<null>
1007	25	GBR	TD3	Gordon	63 Susan Wicks Road	<null>

Note: Be aware that a customer may have several addresses, and the same address may be used by two different customers.

Creating the analysis with the Simple VSR Matcher algorithm

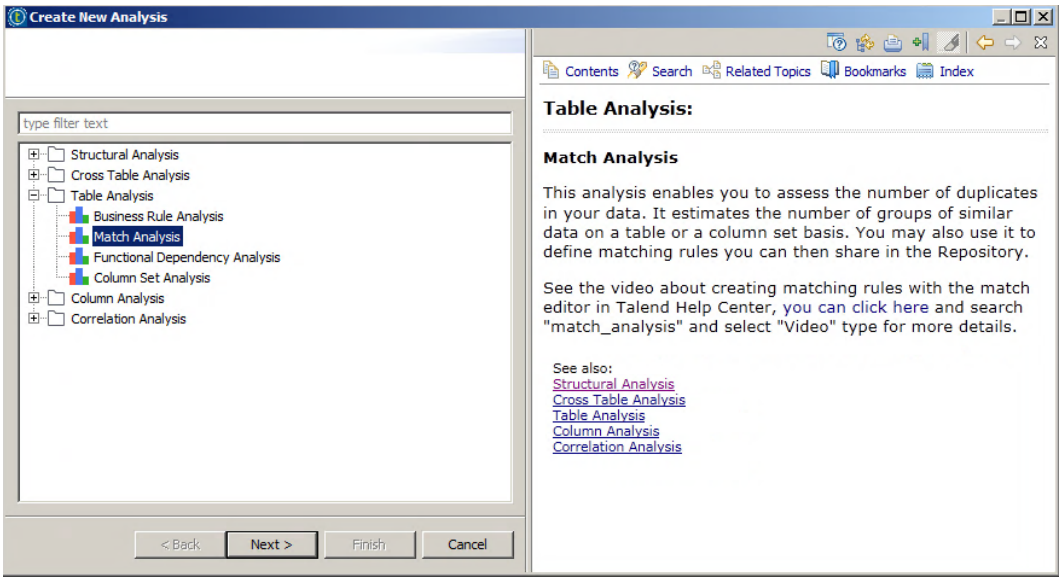
To set up a match analysis, follow the steps you completed in the previous lesson.

Note: Configuring a match analysis is a relatively long process, so you may want to periodically save your work.

1. CREATE A NEW ANALYSIS AND SELECT DATA TO ANALYZE

   Create an analysis in the CRM_Analyses folder.

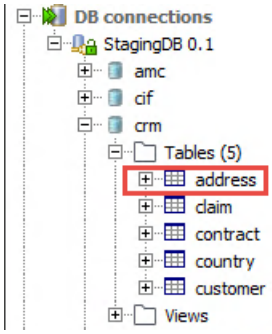
   a. Right-click the **CRM_Analyses** folder and choose **New Analysis**. Expand **Table Analysis** and click **Match Analysis**.



b. Read the context-sensitive help, then click **Next**.

c. Name the analysis *Match_Address_Analysis*. Fill in the **Purpose** and **Description** text boxes and click **Next**.

d. In **DB Connections**, in the **crm** catalog, select the **address** table.



This identifies the table to be analyzed.

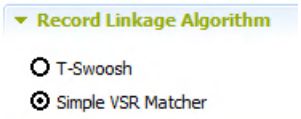
Click **Finish**.

2. SELECT AN ANALYSIS ALGORITHM

The Analysis Settings tab appears. The window is divided into sections that remind the steps of a match analysis creation.

Below the Data Preview section is the Record Linkage Algorithm section, where you select T-Swoosh or Simple VSR Matcher.

Simple VSR Matcher is selected by default.



You will use T-Swoosh later.

Set blocking keys

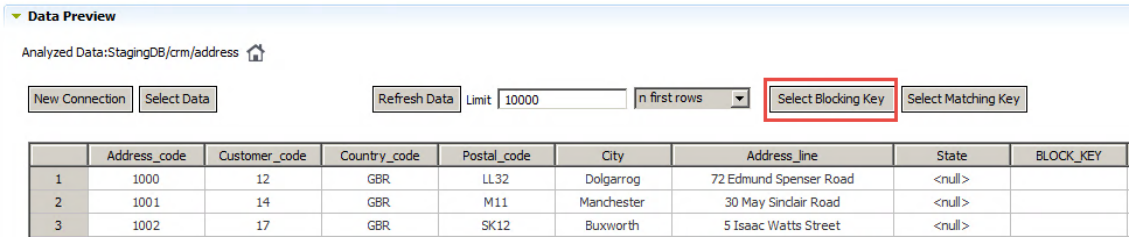
Below the Record Linkage Algorithm section is the Blocking Keys section.

You can select blocking keys there, or, to do it more interactively, in the **Data Preview** section, click **Select Blocking Key**.

Selecting a blocking key (or keys) is not mandatory but is strongly encouraged, especially when working with large datasets.

1. SELECT A BLOCKING KEY

In the Data Preview section, click **Select Blocking Key**. The button turns light gray and the Select Matching Key button turns white, but the Blocking Key section remains blank until you specify the key.

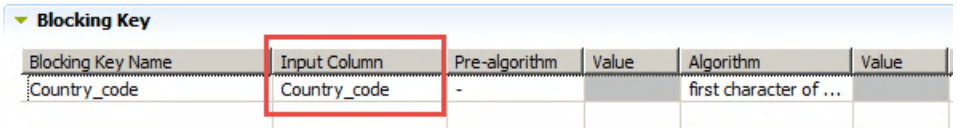


Select the **Country_code** column header. It turns white.

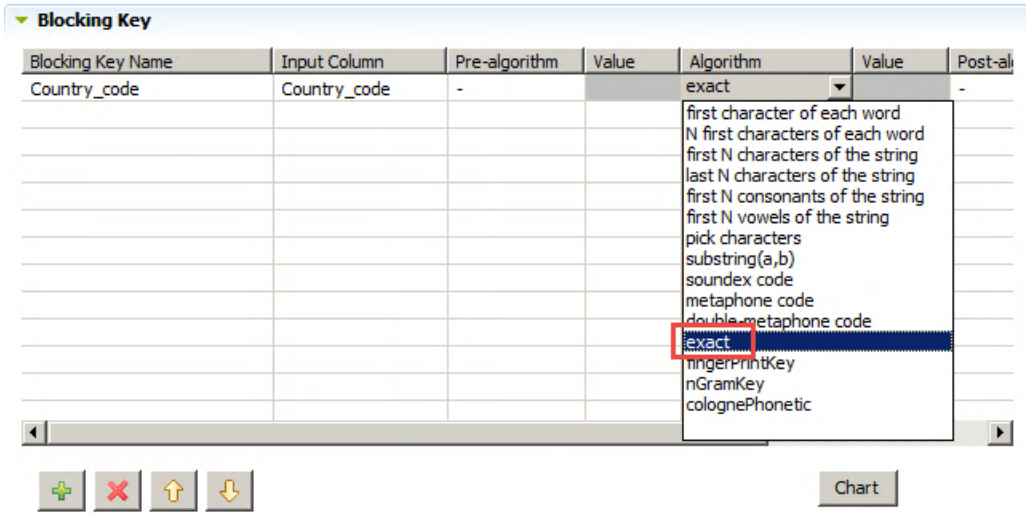
To switch back to the original display, again click **Select Blocking Key**.

2. SELECT AN ALGORITHM

The Country_code key is added to the table in the Blocking Key section. This way, the analysis creates one block per country, and customer addresses are compared in these blocks.

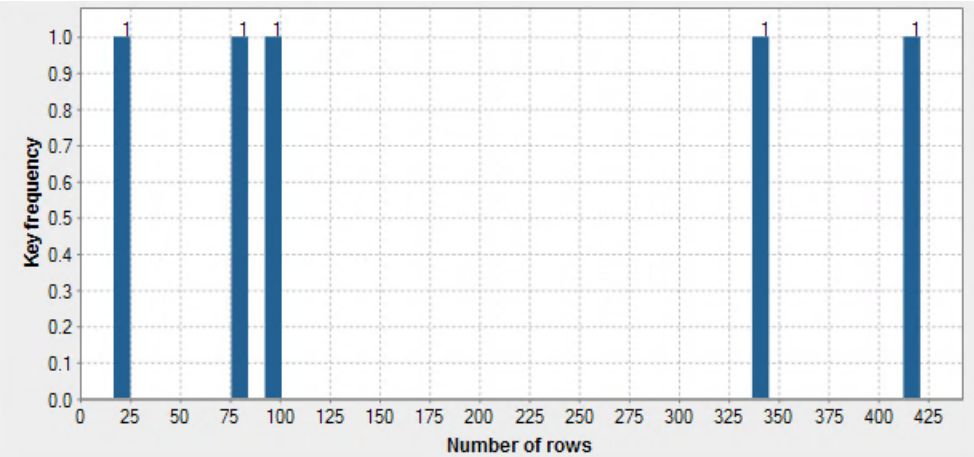


A default algorithm is automatically set up for the key. This algorithm is used to maximize data selection in complex character strings. It is totally useless for three letters country codes made. Change the default **Algorithm** to **exact**.



3. DISPLAY THE KEY FREQUENCY

Below the table, click the **Chart** button.



The key frequency is charted. This shows the number of records by country code.

Notice that only five country codes are found in the data.

Set matching keys

The Matching Key section is located below the Blocking Key section. Like blocking keys, matching keys can be selected from the table in the section or by using the dedicated button in the Data Preview section.

Matching key selection must be consistent with what you are seeking. Your goal is to identify how many duplicate addresses exist per customer. Therefore, you need to select the customer code and columns used to compose an address (to keep it simple, the city and address line).

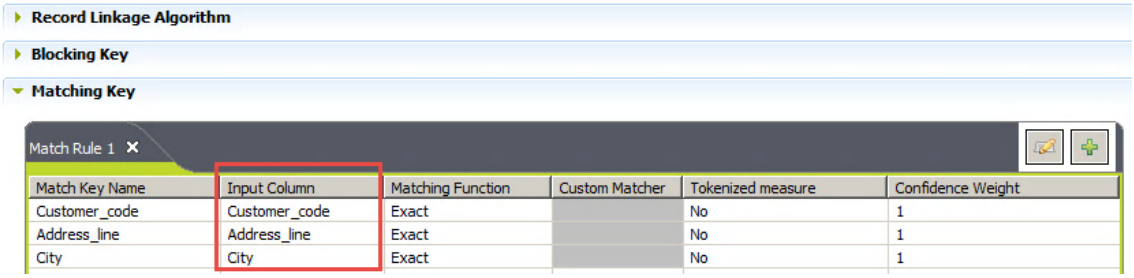
1. ORGANIZE THE ANALYSIS SECTIONS

   To display the **Data Preview** and **Matching Key** sections on the same screen, collapse the **Record Linkage Algorithm** and **Blocking Key** sections.

2. SELECT THE MATCHING KEYS

   In the Data Preview section, click the **Select Matching Key** button. Select the column headers **Customer_code**, **City**, and **Address_line**. To switch back to the original display, again click **Select Matching Key**.

   The selected keys are added to the table in the Matching Keys section.



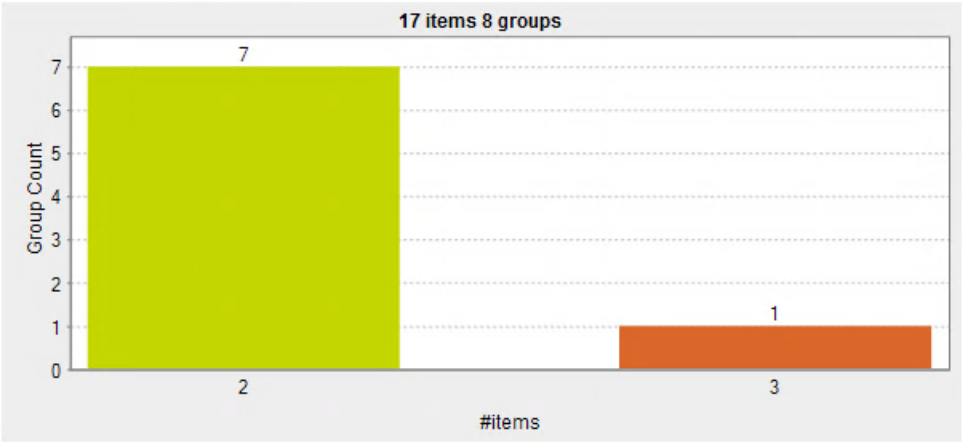
   By default, the Exact matching function is used.

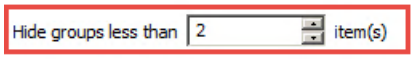
   Matching functions can be updated according to the data stored in the column, but first you can see how many duplicates are found without additional changes.

3. REFRESH THE DATA PREVIEW

   Below the table, click the **Chart** button.

   The groups distribution is charted and the Data Preview section is updated.





   By default, the chart shows only groups with at least two items. This value can be adapted below the chart.

   This also applies to The Data Preview section which shows what is in each group.

	Address_code	Customer_code	Country_code	Postal_code	City	Address_line	State	BLOCK_KEY	GID	GRP_SIZE	MASTER	SCORE	GRP_QUALITY	ATTRIBUTE_SCORES
9	4085	948	ESP	32558	Cepedelo	Bulevar Garcilaso de la Vega 76	<null>	ESP	e3e21229-8b66-4...	3	true	1.0	1.0	
10	4086	948	ESP	32558	Cepedelo	Bulevar Garcilaso de la Vega 76	<null>	ESP	e3e21229-8b66-4...	0	false	1.0	0.0	Customer_code: 1.0 | City: 1.0 | Address_line: 1.0
11	4087	948	ESP	32558	Cepedelo	Bulevar Garcilaso de la Vega 76	<null>	ESP	e3e21229-8b66-4...	0	false	1.0	0.0	Customer_code: 1.0 | City: 1.0 | Address_line: 1.0
12	4014	103	ESP	28016	Madrid	Calle Miguel de Cervantes 39	<null>	ESP	8e23407b-d077-4...	2	true	1.0	1.0	
13	4095	103	ESP	28016	Madrid	Calle Miguel de Cervantes 39	<null>	ESP	8e23407b-d077-4...	0	false	1.0	0.0	Customer_code: 1.0 | City: 1.0 | Address_line: 1.0
14	2364	434	USA	02814	Chepachet	20 Pauline Hopkins Avenue	RI	USA	32e61d3d-aeb4-4...	2	true	1.0	1.0	
15	2395	434	USA	0284	Chepachet	20 Pauline Hopkins Avenue	<null>	USA	32e61d3d-aeb4-4...	0	false	1.0	0.0	Customer_code: 1.0 | City: 1.0 | Address_line: 1.0

The same colors are used in the chart and Data Preview.

Several new columns are added:

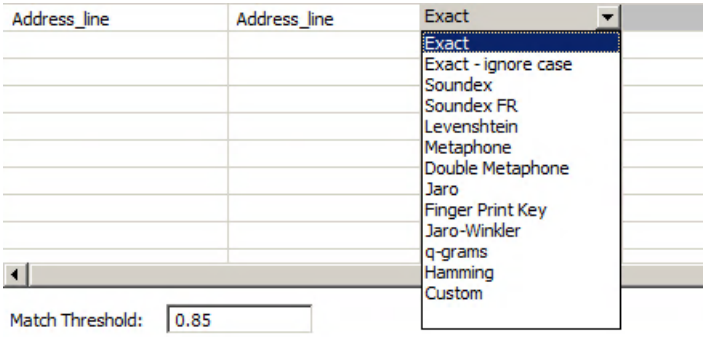
- **GID** is the group ID, a unique identifier
- **GRP-SIZE** indicates the size of the group (single, pair, triplet or more)
- **MASTER** identifies the golden record of the group. There is one master for each group.
- **SCORE** is the similarity score computed by the analysis algorithm. It reflects how close the row is to the master. A score of 1 means that it is the master or a duplicate of the master. The closer it is to 1, the closer a match is to the master record.
- **GRP_QUALITY** defines the group quality. Its value is the lowest score in the group.
- **ATTRIBUTE_SCORES** are the individual scores of matching keys.

Note: Approximate numbers are used above. As you rechart after making minor adjustments in various parameters, your numbers may vary.

4. ADAPT THE MATCHING FUNCTION FOR THE ADRESS_LINE KEY

   For the time being, the analysis detects only exact match records. You can amend the search by using more-relevant matching functions. You will start with the Address_Line key.

   a. To open the drop-down list of available algorithms, click the **Matching Function** parameter in the **Address_line** key.

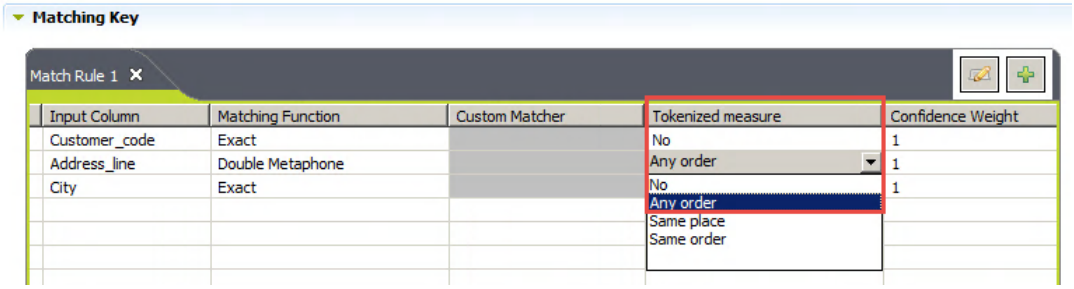


   b. Select the **Double Metaphone** algorithm.

      The Double Metaphone is one of the more efficient phonetic algorithms. It was developed to search for phonetic similarities in the English language. Unfortunately, the Address table stores addresses from five countries.

      Click the **Chart** button. More groups of pairs (two rows) and triplets (three rows) are identified. Take a moment to look at the identified duplicates in the **Data Preview** section.

   c. To take into account the strings that have words in a different order, in the **Address_line key**, click the **Tokenized measure** parameter and select **Any order**.



Click the **Chart** button and confirm that more groups of similar rows are identified.

The Tokenized measure parameter does not have the same level of effectiveness with all the algorithms.

d. Again click the **Matching function** parameter in the **Address_Line** key and select **q-grams**.

Q-grams is a distance comparison algorithm that is well adapted to long text strings. It compares small sequences of items (phonemes, syllables, letters, words) and adds up their differences to compute the score.

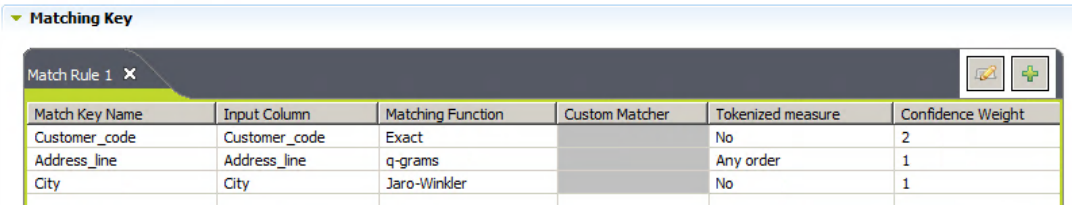
Click the **Chart** button. More groups are identified.

Below is a triplet found by the q-grams algorithm but ignored by the Double Metaphone, which, despite the omission of the first name, was not able to recognize that the street is the same.

4. ADAPT THE MATCHING FUNCTION FOR THE OTHER KEYS

a. Take your time testing several algorithms for the **City** key, then select the **Jaro-Winkler** algorithm. Jaro-Winkler is a distance algorithm well adapted to short text strings. It computes the distance between two strings, giving more importance to the beginnings of the strings.

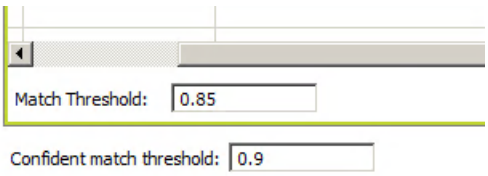
b. To finish, keep the **Exact** matching function for the **Customer_code**, as you want to ensure that the analysis identifies address duplication for each customer. To give the Customer_code key more importance, update the **Confidence Weight**. Change the weight from *1* to *2*.



Set the thresholds

You need to set the match threshold and confident threshold for the analysis.

By default, the match threshold is set at 0.85 and the confident match threshold at 0.9.



Modifying these values impacts the number of duplicates and number of suspect records revealed by the analysis.

1. SET THE MATCH THRESHOLD

   Start by modifying the match threshold.

   a. In the match threshold box, enter *0.70* , then click **chart**. Since the comparison criteria are less stringent, more duplicates are identified.

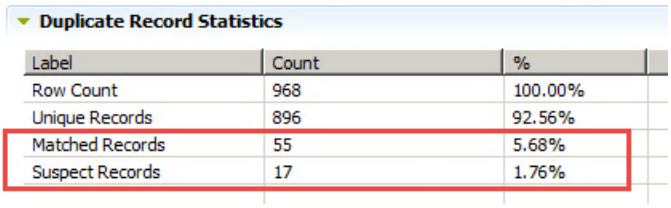
   b. Enter *0.90* in the match threshold box, then click **chart**. This time the analysis finds fewer duplicates.

   c. To better understand the impact of the matching threshold, you can test other values. To finish, enter *0.75* and validate the value by clicking **chart** .

2. SET THE CONFIDENT THRESHOLD

   To see the impact of the confident match threshold, you must run the analysis.

   a. Run the analysis. On the Analysis result tab, notice the numbers of suspect and matched records.

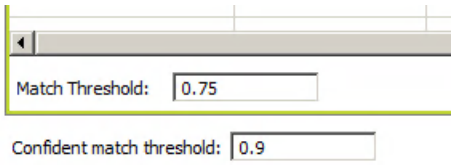


   Display the Analysis Settings tab to modify the confident match threshold.

   b. Enter *0.95* in the confident match threshold box, then run the analysis. The number of suspect records increases and the number of matched records decreases equally.

   c. Enter *0.85* in the confident match threshold box, then run the analysis. This time, more similar rows are considered matched records, and fewer as suspect records.

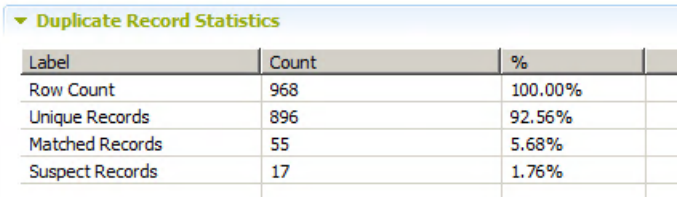
   d. To better understand the impact of the confident match threshold, you can test other values. To finish testing, enter *0.9* and save the analysis.



Run the analysis

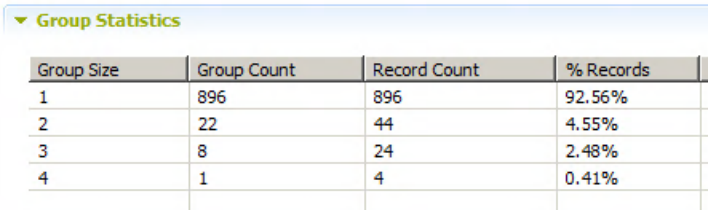
Run the analysis. The results are displayed in two sections:

» The table and chart in the Duplicate Record Statistics section shows the rows distributed by single, matched or suspect records.



» The table and chart in the Group Statistics section shows the matched and suspect records distribution in groups by sim-

ilarity.



Group Statistics

Group Size	Group Count	Record Count	% Records
1	896	896	92.56%
2	22	44	4.55%
3	8	24	2.48%
4	1	4	0.41%

Before exporting the match rule, additional settings must be configured.

Configuring additional settings for the table match analysis

Overview

In this lesson you will duplicate the table match analysis and configure it with the T-Swoosh algorithm.

Then you will save the analysis parameters in a match rule.

Using the T-Swoosh algorithm

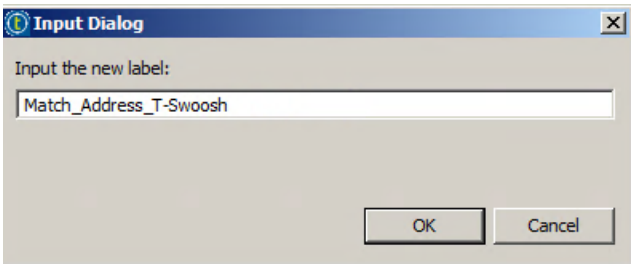
In this exercise, you will duplicate the table match analysis and modify the record linkage algorithm.

1. DUPLICATE THE TABLE MATCH ANALYSIS

   Duplicate the analysis. Having an analysis using Simple VSR Matcher and another one using T-Swoosh will help you understand the distinctive features of each algorithm.

   a. In the **CRM_Analysis** folder, right-click **Address_Match_Analysis,** and on the contextual menu, choose **Duplicate**.

   b. In the **Input the new label:** text box, enter *Match_Address_T-Swoosh*.

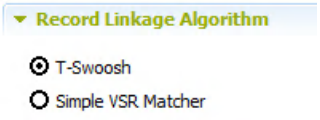


   c. To open the analysis, double-click it.

2. SELECT THE T-SWOOSH ALGORITHM

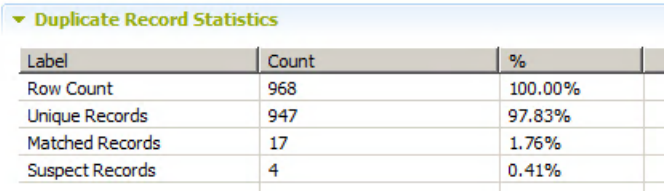
   Change the default record linkage algorithm.

   a. In the **Record Linkage Algorithm** section of the **Analysis Settings** tab, select the **T-Swoosh** algorithm.



   b. Without modifying any other settings, run the analysis.

   Notice that the numbers of duplicates and suspect records, as well as the group distributions, have changed.



Label	Count	%
Row Count	968	100.00%
Unique Records	947	97.83%
Matched Records	17	1.76%
Suspect Records	4	0.41%

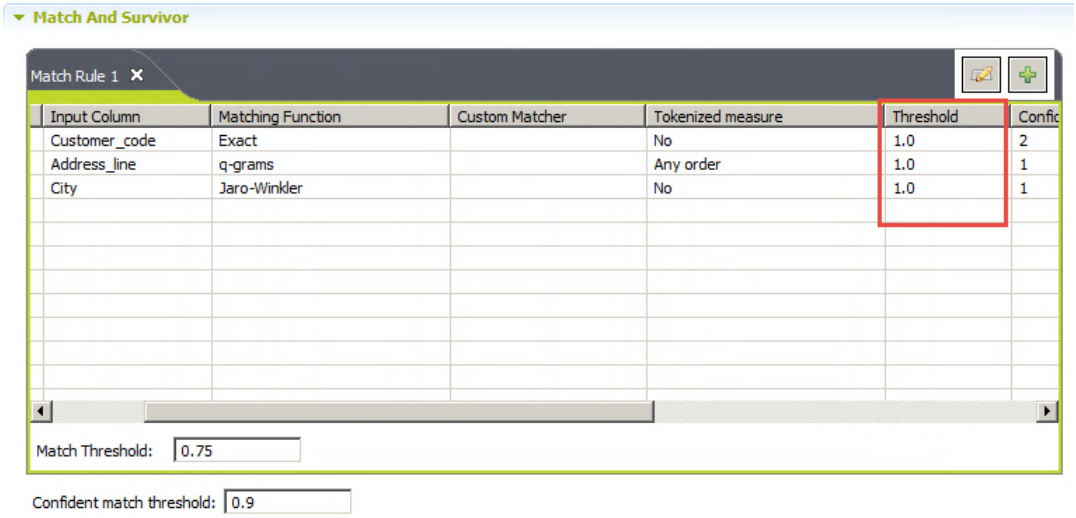
3. SET THE KEY THRESHOLD

   A new threshold is available.

   a. Display the **Analysis Settings** tab.

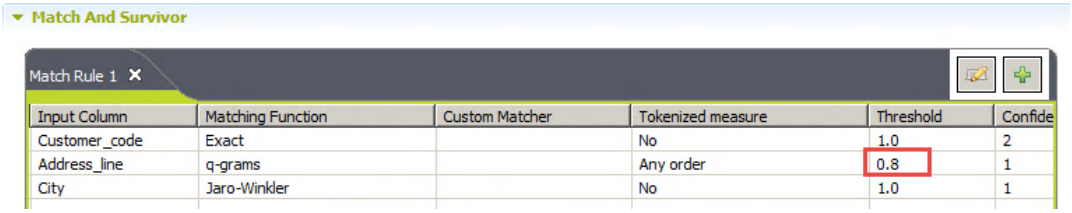
   Notice that the Matching Keys section is now called Match and Survivor. New features are available. You will use the survivor parameters, but first view the new threshold.

You may need to scroll to display the new Threshold column, located on the right side of the Match Rule table.

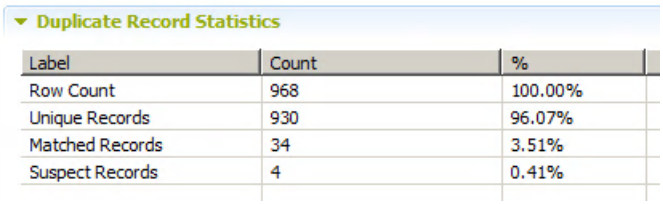


This individual threshold has a distinct value for each matching key. This value is factored in to the calculation of the similarity score computed by the T-Swoosh algorithm.

b. Lower the **Threshold** value of the **Address_line** key from *1.0* to *0.8*.



c. Run the analysis.



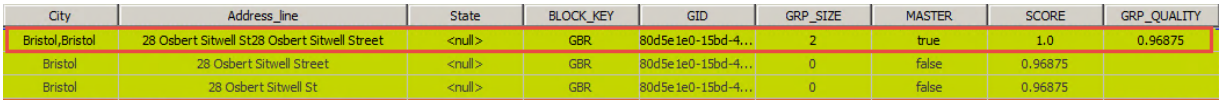
The comparison on Address_line is more permissive, so the analysis identifies more duplicates.

Setting up a new survivor rule

Display the **Analysis Settings** tab and click the **Chart** button at the bottom of the **Match and Survivor** section. Then use the slide to display the data preview on the upper part of the tab.

A survivor row was created for each group.

The image below focuses only on one group. The survivor is displayed on top.

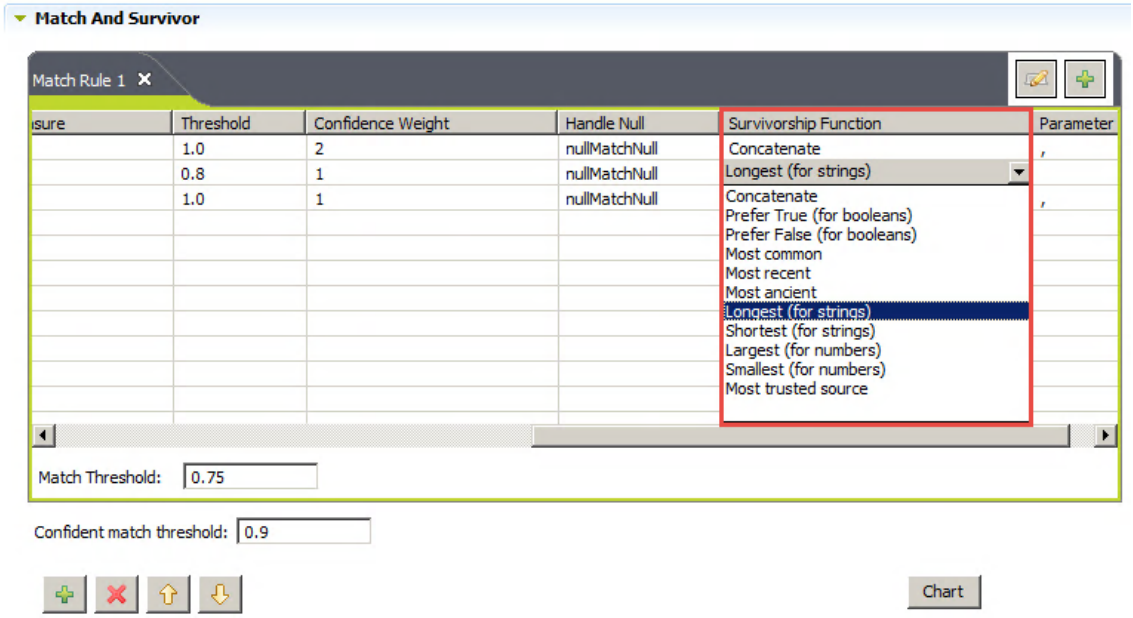


By default, values for the survivor row are a concatenation of the other records in the group. Survivor rules can be set up in the Match and Survivor section.

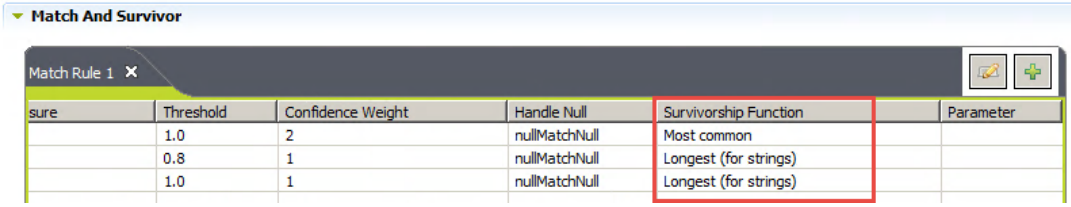
1. SET UP SURVIVOR RULES

   Display the **Match and Survivor** section. To the right of the Match Rule table is a Survivorship Function column specific to the T-Swoosh algorithm.

   To update the default function, use the drop-down list.



   a. For the **Address_line** and **City** keys, select **Longest (for strings)**. For the **Customer_code** key, select **Most common**.



   Note: A Longest address string is usually considered more complete.

   b. To refresh the data preview, click the **Chart** button.

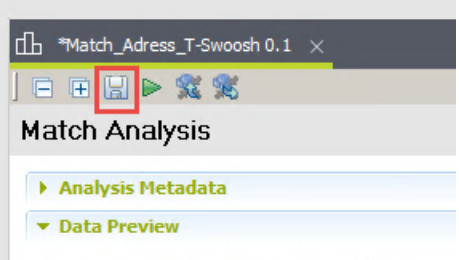
2. EXAMINE THE RESULTS

   Display the Data Preview Section.

   a. Notice that the values stored in the **Address_Line** and **City** columns are updated for all survivor rows.

Customer_code	Country_code	Postal_code	City	Address_line
301	GBR	BS41	Bristol	28 Osbert Sitwell Street
301	GBR	BS41	Bristol	28 Osbert Sitwell Street
301	GBR	BS41	Bristol	28 Osbert Sitwell St

b. To save the analysis, on the analysis toolbar, click the **Save** button.



Note: You can also set up survivorship rules for the other columns that are not matching keys in the Default Survivorship Rules section, located below the Match And Survivor section.

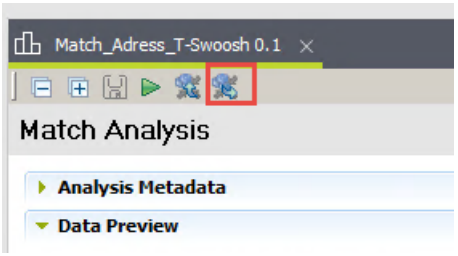
Exporting the match rule

The table match analysis settings can be exported in a match rule, then reused in another analysis or an integration job.

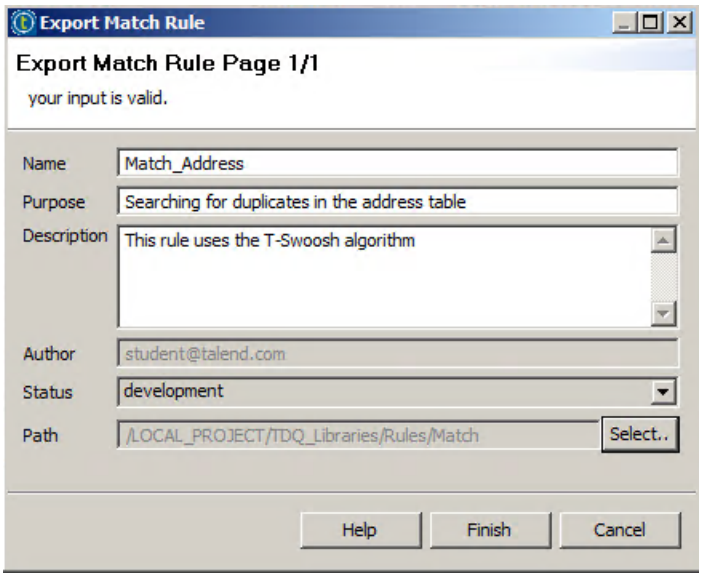
1. EXPORT THE MATCH RULE

   Display the **Analysis Settings** tab of the **Match_Address_T-Swoosh** analysis.

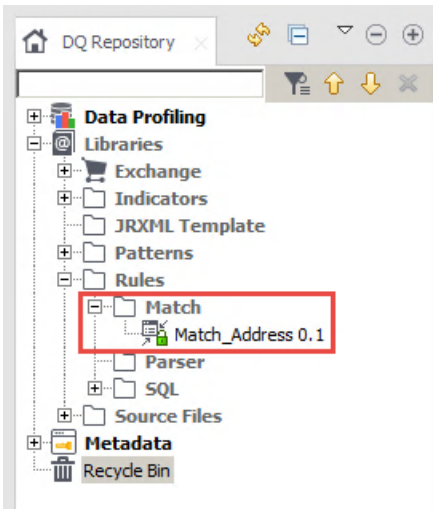
   a. Click the last button on the analysis toolbar, **Export Match Rule**.



   b. In the **Name** text box, enter *Match_Address*. As a best practice, fill in the **Purpose** and **Description** text boxes, then click **Finish**.



   The match rule is now available in the DQ Repository.



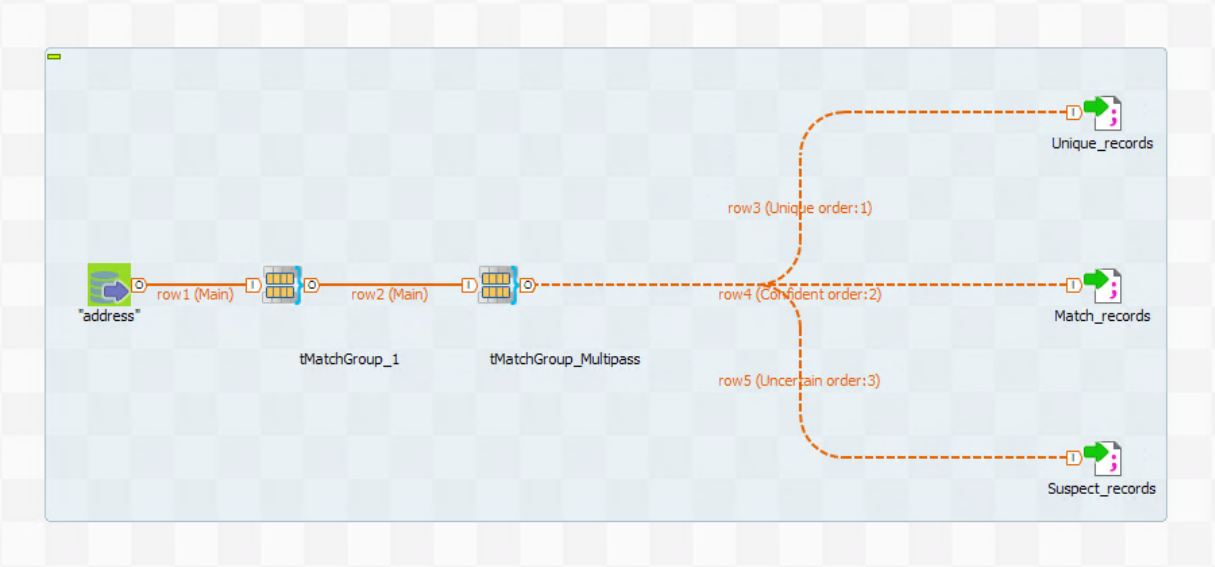
The next step is to reuse the match rule in an integration Job.

Using a matching integration Job

In this lesson you will create an integration Job for data matching. You will use a tMatchGroup component to analyze the address table and then export single, suspect, and duplicate rows in separate files.

To easily set up the job, you will reuse the match rule you created in the previous exercise. Then you will improve the sharpness of the Job by using two tMatchGroup components in a series.

At the end of this lab, your Job should be as follows.



Before building the Job, you must adapt the metadata you created for the database server connection.

Adapting the DB connection metadata

In the first lesson, you created the StagingDB metadata to provide access to all the databases stored on the local MySQL server.

To facilitate the creation of the integration Job, you must restrict the connection to just the CRM database.

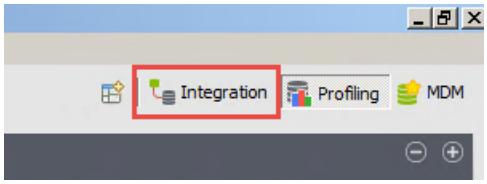
It is safest to duplicate the StagingDB connection and then modify the copy.

1. DUPLICATE THE METADATA

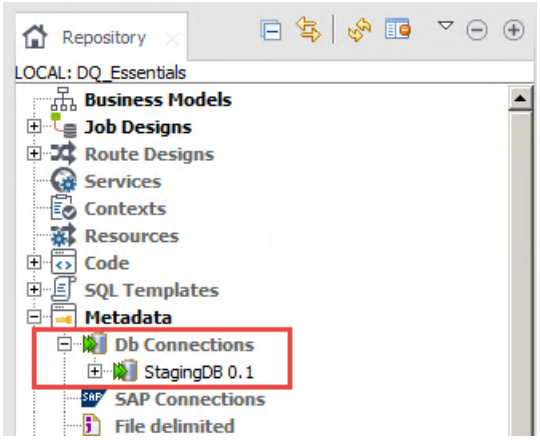
   The metadata is available in both the Profiling and Integration perspective.

   a. To avoid confusion, in the **Designer**, close all the open analyses and rules.

   b. To switch to the Integration perspective, above the **Designer**, click the **Integration** button.

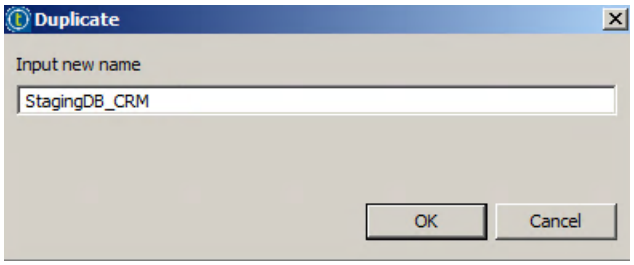


c. In the **Repository**, in the **Metadata** folder, right-click the **StagingDB** connection.

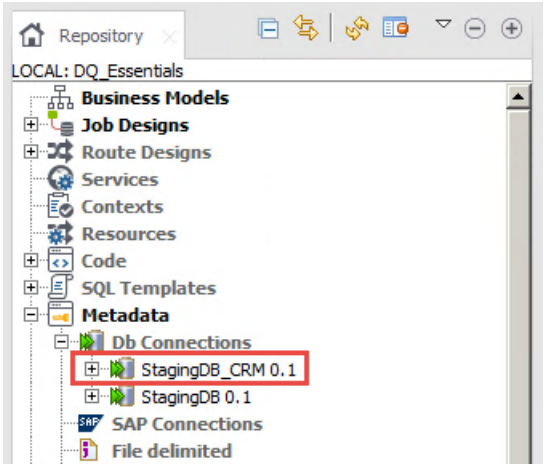


On the contextual menu, choose **Duplicate**.

d. In the **Input New Name** text box, enter *StagingDB_CRM* and click **OK**.



The new metadata is available in the repository.



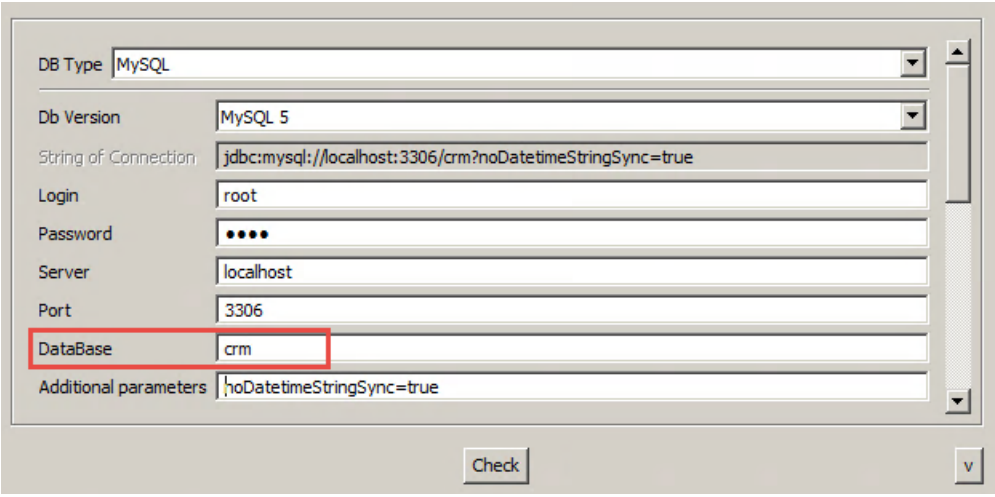
2. MODIFY THE CLONE

Modifying the clone will not affect the analyses previously set up with the original StagingDB metadata.

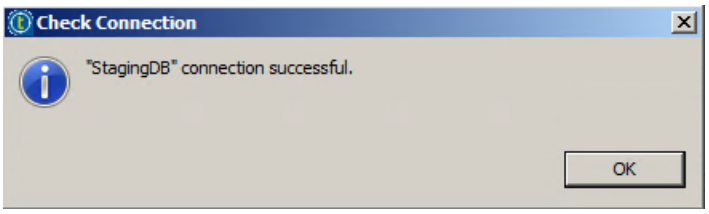
a. Right-click **StagingDB_CRM**, and on the contextual menu, choose **Edit connection**.

b. Update the **Purpose** and **Description** text boxes and click **Next**.

c. In the **DataBase** text box, enter *crm* and click the **Check** button.

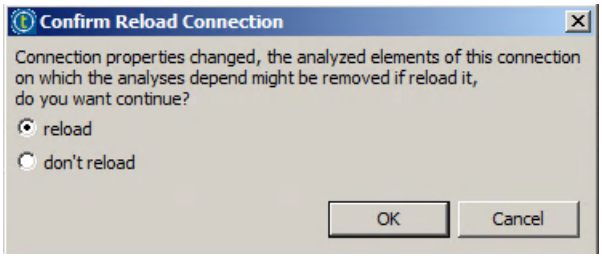


d. In the **Check Connection** window, click **OK,** or make corrections and again click **Check**.

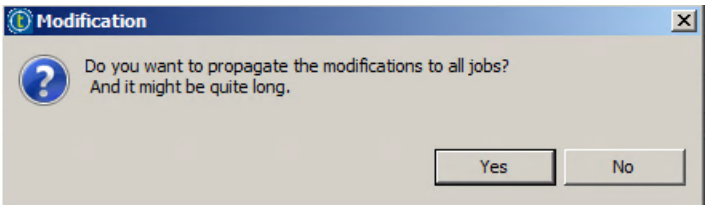


e. Click **Finish**.

f. In the **Confirm Reload Connection** window, without modifying anything, click **OK**.



g. In the Modification window, click **No** (this new metadata is not used in any Jobs).



The metadata is ready to use.

Creating the matching integration Job

The Job you are about to create uses:

- A tMysqlInput component to extract data from the CRM database
- A tMatchGroup component to profile the data

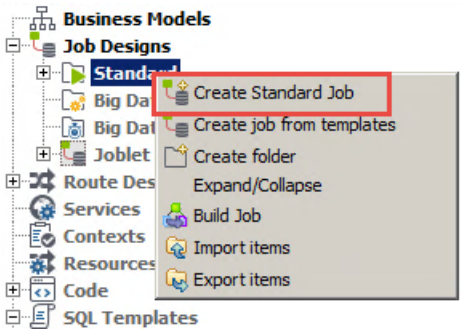
» Three tFileOutputDelimited components to export single, suspect, and duplicate rows in separate files

The tMatchGroup component uses the same settings as the table match analysis. Therefore, you can easily set it for importing the match rule.

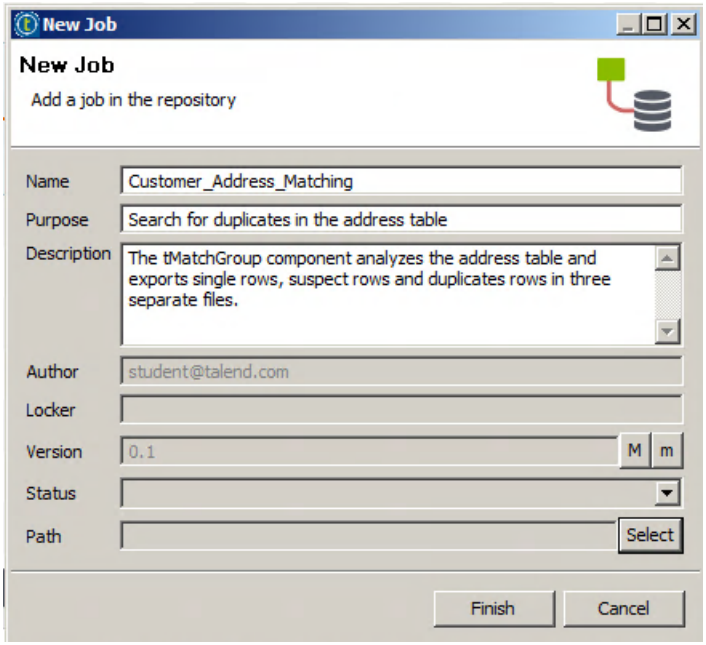
1. CREATE THE JOB

   The Job must be created in the Standard directory of the repository, alongside the other Jobs you created in a previous lesson.

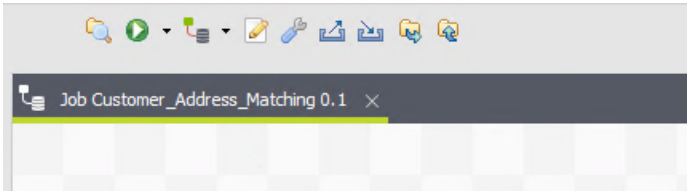
   a. In the **Repository**, expand **Job Designs** and right-click the **Standard** directory. On the contextual menu, choose **Create Standard Job**.



   b. Fill in the **Name**, **Purpose**, and **Description** text boxes and click **Finish**.



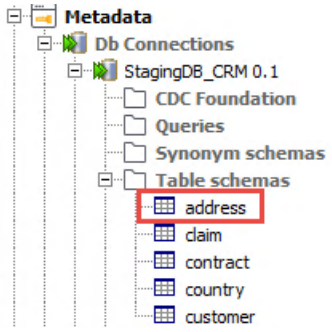
   The Job is open in the work area.



2. ADD A tMysqlInput COMPONENT

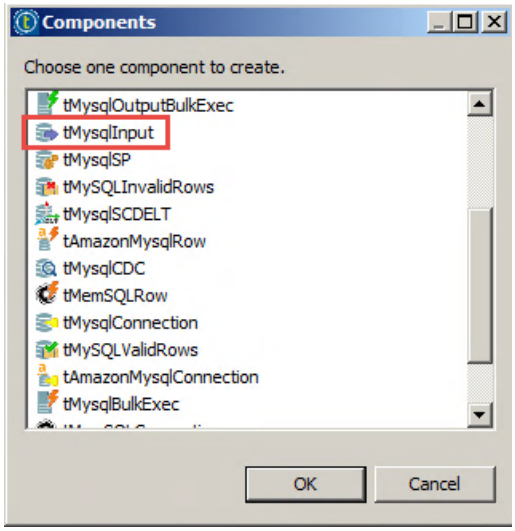
You need to connect to the database using a tMysqlInput component.

a. In the **Repository**, go to **Metadata**, **Db Connections>StagingDB_CRM>Table schemas**.



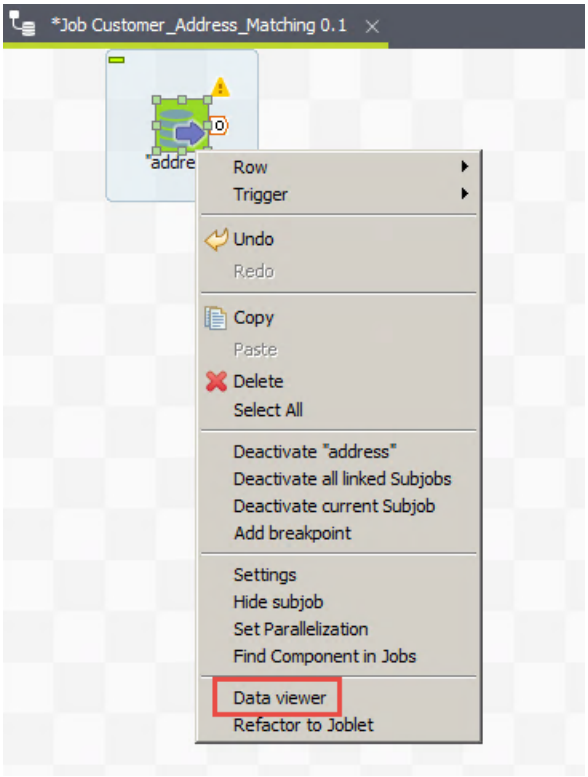
b. Click the **address table** icon and place it in the work space.

c. Select the **tMysqlInput** component and click **OK**.

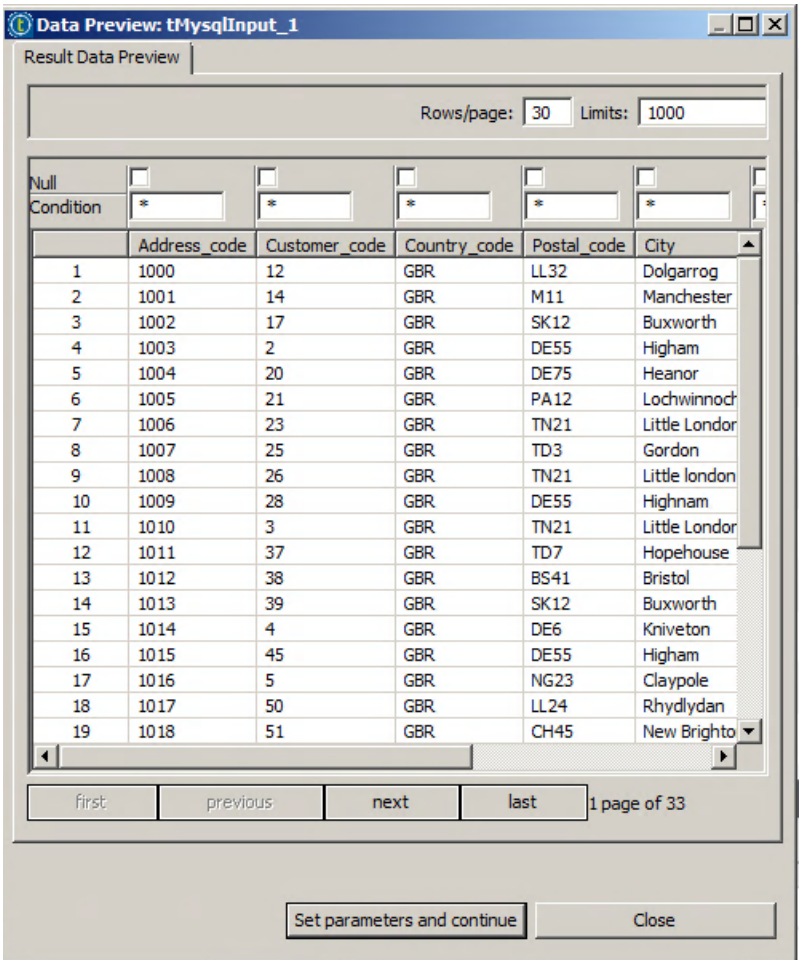


d. The component is already set up to extract data from the address table. To check the connection, right-click the com-

ponent, and on the contextual menu, choose **Data viewer**.



e. The first table rows are displayed in the Data Preview window.



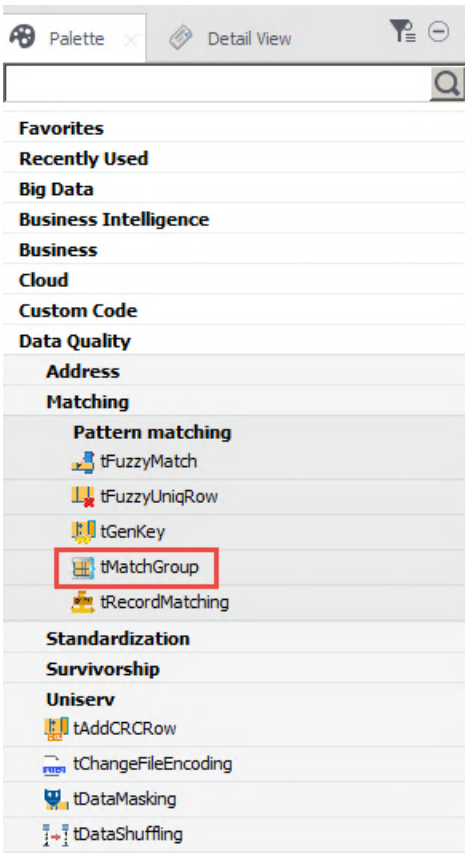
	Address_code	Customer_code	Country_code	Postal_code	City
1	1000	12	GBR	LL32	Dolgarrog
2	1001	14	GBR	M11	Manchester
3	1002	17	GBR	SK12	Buxworth
4	1003	2	GBR	DE55	Higham
5	1004	20	GBR	DE75	Heanor
6	1005	21	GBR	PA12	Lochwinnoch
7	1006	23	GBR	TN21	Little Londor
8	1007	25	GBR	TD3	Gordon
9	1008	26	GBR	TN21	Little london
10	1009	28	GBR	DE55	Highnam
11	1010	3	GBR	TN21	Little Londor
12	1011	37	GBR	TD7	Hopehouse
13	1012	38	GBR	BS41	Bristol
14	1013	39	GBR	SK12	Buxworth
15	1014	4	GBR	DE6	Kniveton
16	1015	45	GBR	DE55	Higham
17	1016	5	GBR	NG23	Claypole
18	1017	50	GBR	LL24	Rhydlydan
19	1018	51	GBR	CH45	New Brighto

Click **Close**.

3. ADD A tMatchGroup COMPONENT

To add a new component, use the palette to the right of the work space.

a. In the **Palette**, expand the **Data Quality** section to display all the DQ components. tMatchGroup is in the Pattern matching subsection.



b. Drag the **tMatchGroup** component into the work space next to the **address** component.

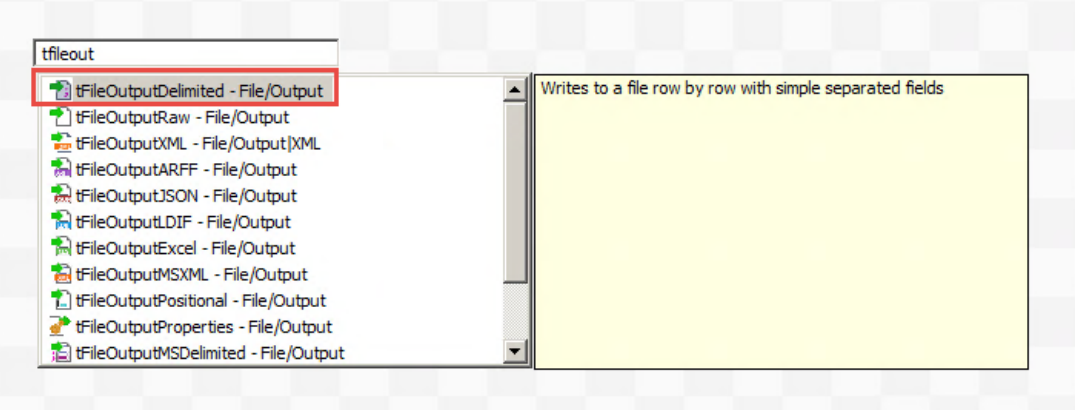
4. ADD THREE tFileOutputDelimited COMPONENTS

   You will use a different method to place new components in the work space.

   a. Click in the space to the right of the **tMatchGroup** component.

      Enter the first letters of the *tFileOutputDelimited* component name.

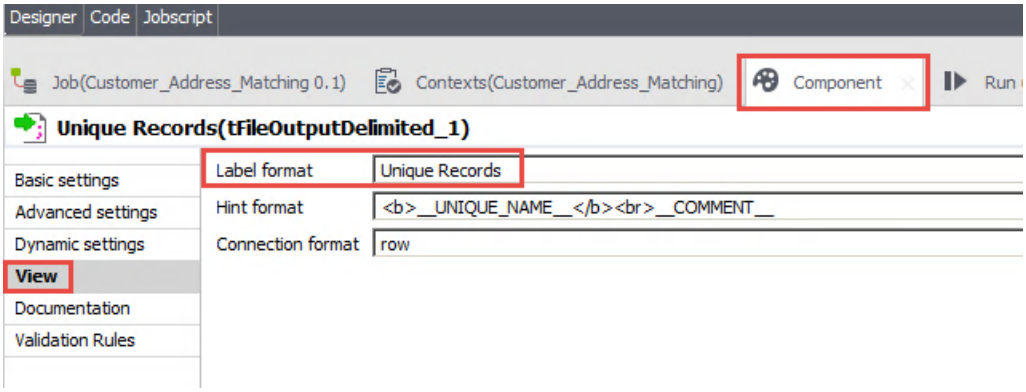
      On the list of compatible components, double-click **tFileOutputDelimited**.



To add three **tFileOutputDelimited** components, do this three times.

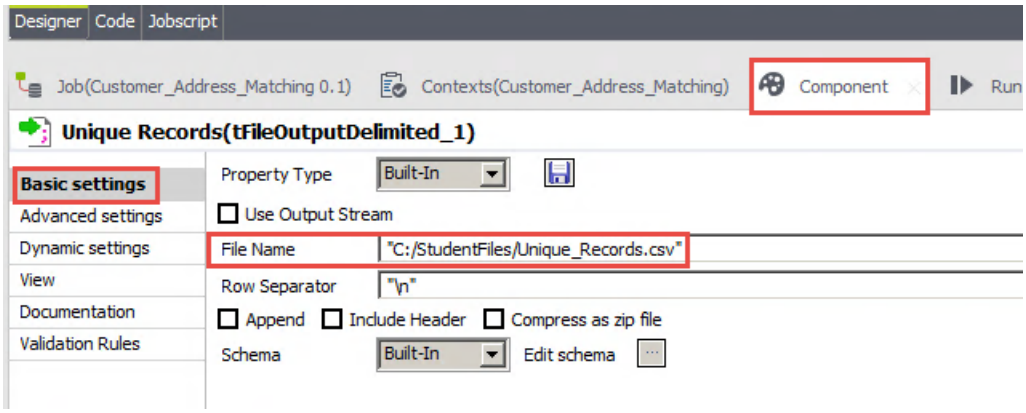
In the **Component** view, update the component settings.

b. To rename the component, select the **View** tab. In the **Label format** text box, enter the component name.



- For the first component, enter *Unique _Records*
- For the second component, enter *Match_Records*
- For the third component, enter *Suspect_Records*

c. To change the default path and file name, select the **Basic settings** tab. Enter the new path in the **File Name** text box.



- For the first component, enter *"C:/StudentFiles/Unique _Records.csv"*
- For the second component, enter *"C:/StudentFiles/Match_Records.csv"*
- For the third component, enter *"C:/StudentFiles/Suspect_Records.csv"*

The three components are ready to use.

Configuring the components

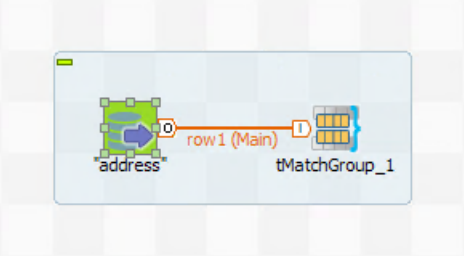
1. SET UP THE tMatchGroup COMPONENT

   You can easily set up the tMatchGroup component by importing the match rule.

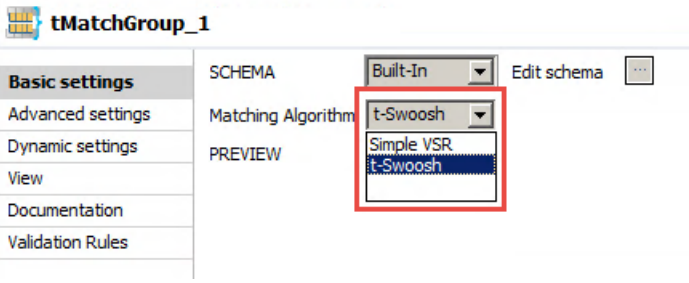
   Note: When the rule is imported, you must still manually update some settings.

a. Connect the **tMysqlInput** component to the **tMatchGroup** component using the **Main** row.

Note: If there is a warning or an error on the component, click the **Sync columns** button to update the schema.

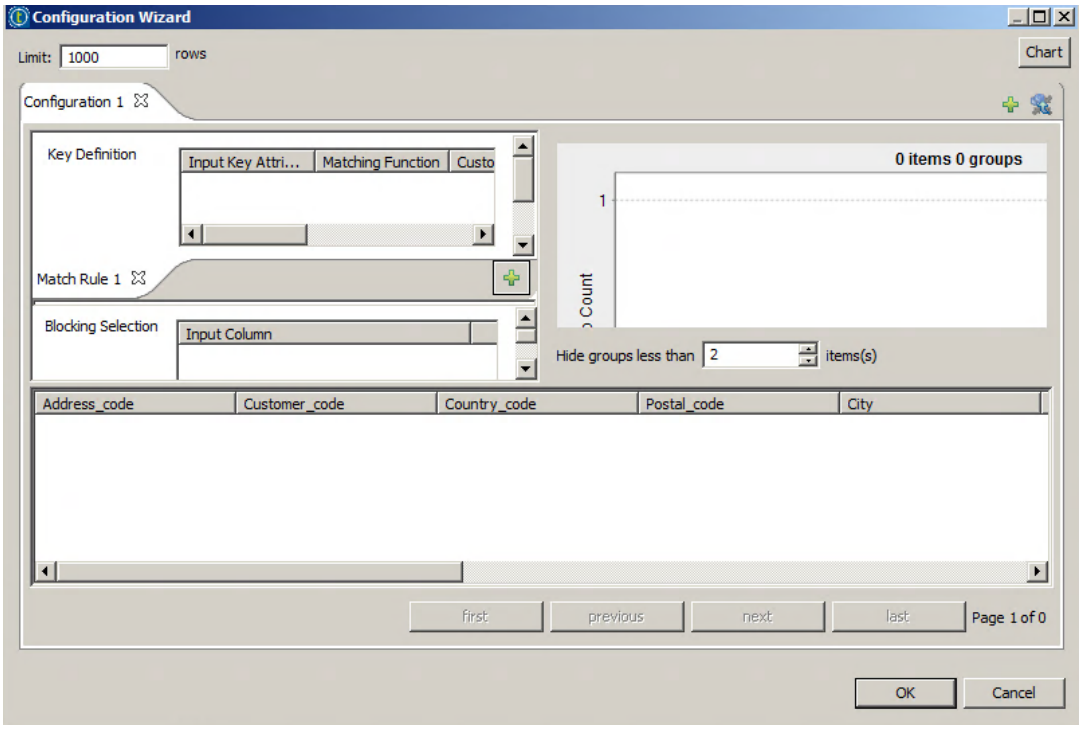


b. Select the **tMatchGroup** component, and on the **Basic settings** tab of the **Component** view, in the **Matching Algorithm** box, select **t-Swoosh**.



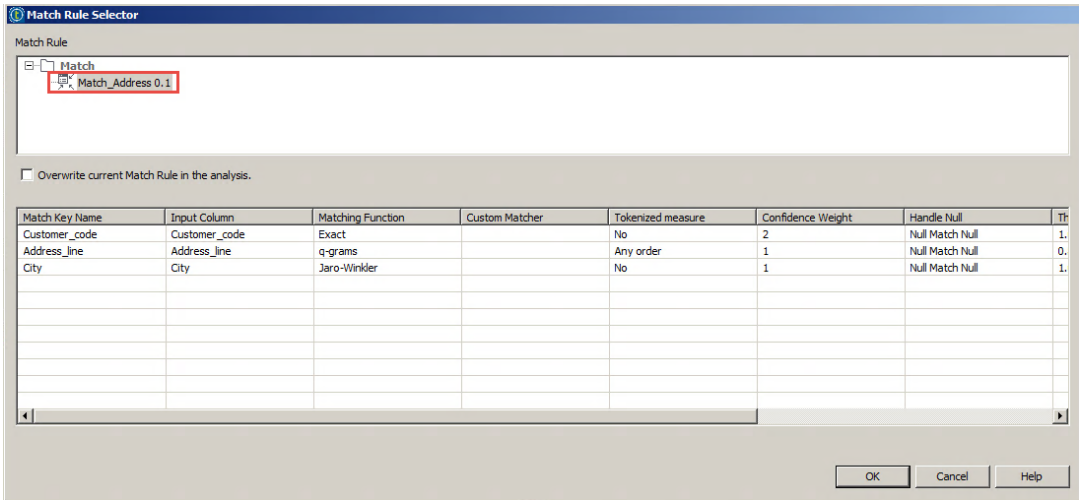
c. To open the Configuration wizard, double click the **tMatchGroup** component.

This wizard gathers almost all the settings you used to configure the match analysis.



d. To import the match rule, in the upper right corner of the wizard, click the **Import match rule from repository** icon.

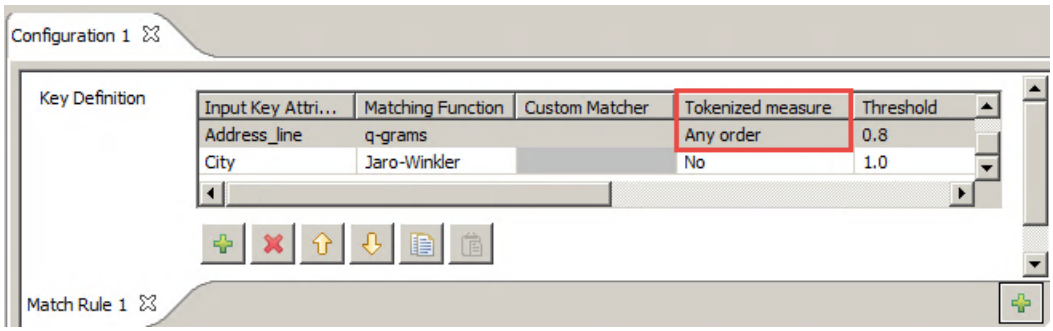
e. In the **Match Rule Selector** window, expand **Match** and select the match rule. The matching keys and their parameters appear.



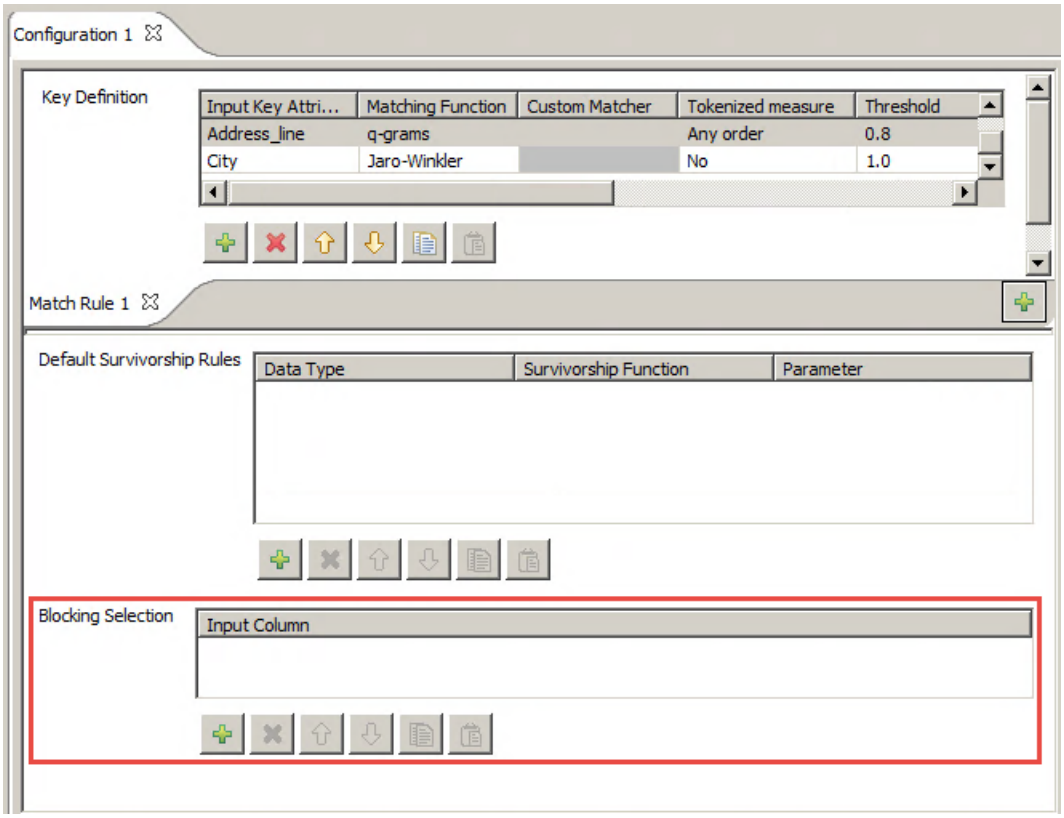
Click **OK**.

The matching key configuration has been imported, but you must update some settings.

f. Ensure that the **Tokenized measure** parameter of the **Address_line** key is designated **Any order**.

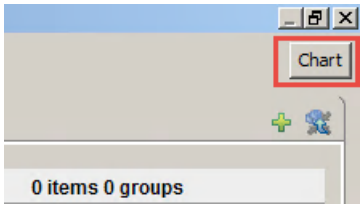


g. The Blocking Selection section (below Key Definition and Default Survivorship Rules) is empty.



To create a new key, click the **plus symbol (+)**, and in **Input Column**, select **Country_code**.

h. To refresh the data preview, in the upper right corner of the wizard, click the **Chart** button.



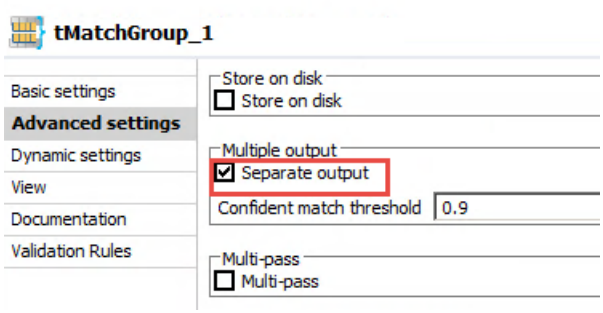
Similar to what you saw in the table matching analysis, the wizard displays the distribution of duplicated rows in groups.

i. Examine the preview, then click **OK**.

2. CONNECT THE tFileOutputDelimited COMPONENTS

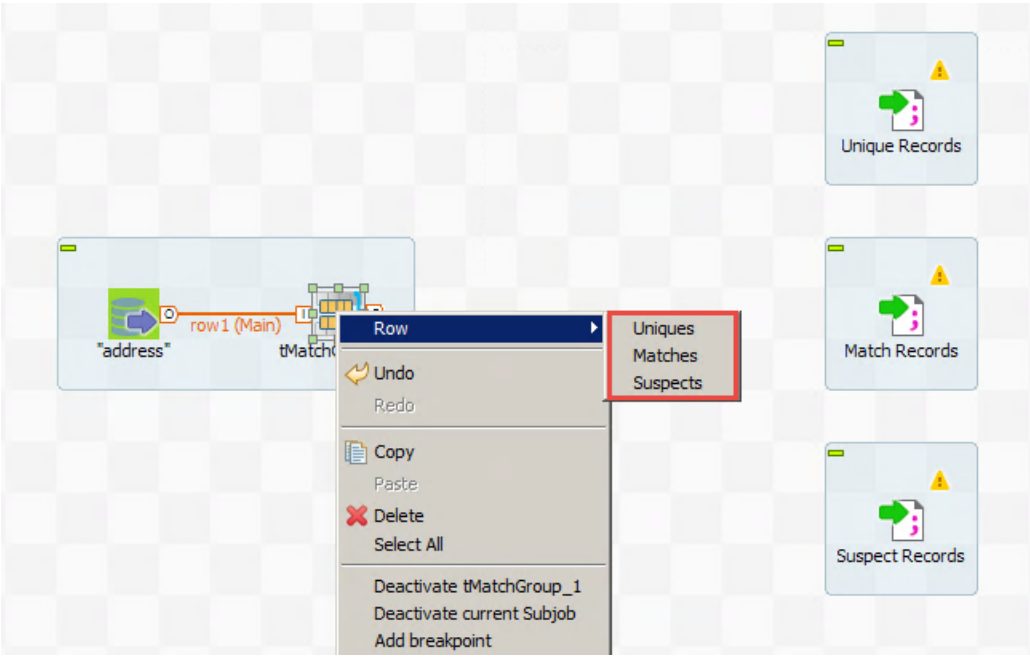
The results of the matching Job must be exported in three separate files. Then, data stewards take in charge the suspect records and decide whether they are single or duplicate records.

a. To activate the three outputs of the tMatchGroup component, on the **Advanced settings** tab of the **Component** view, select **Separate output**.

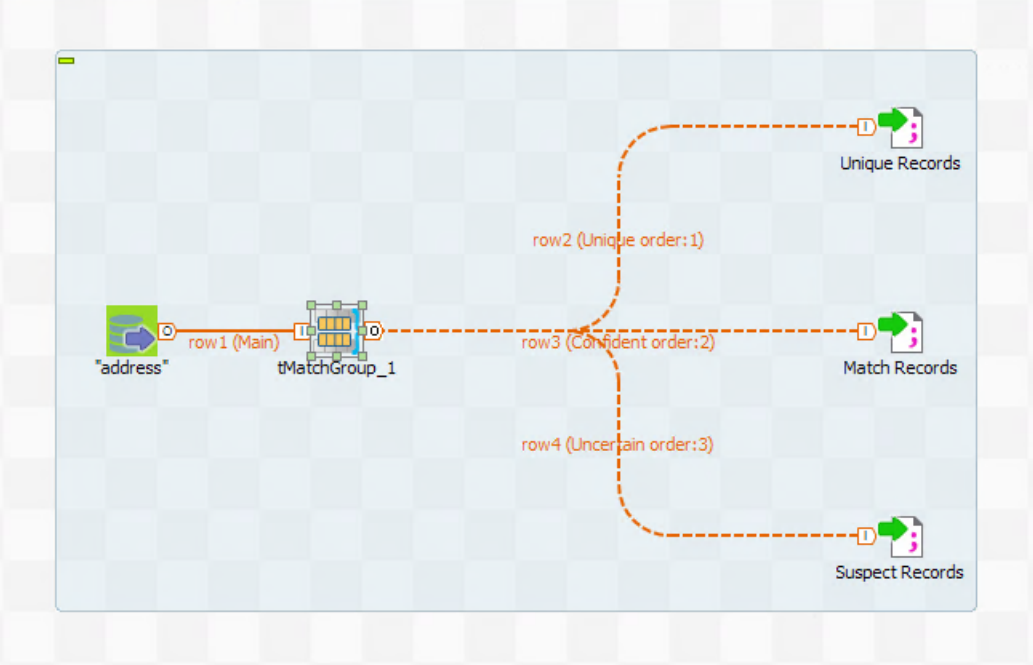


Note: This option activates the confident match threshold. Make sure the value corresponds to the value imported with the match rule.

b. In the work space, right-click the **tMatchGroup** component. Three output rows are available.

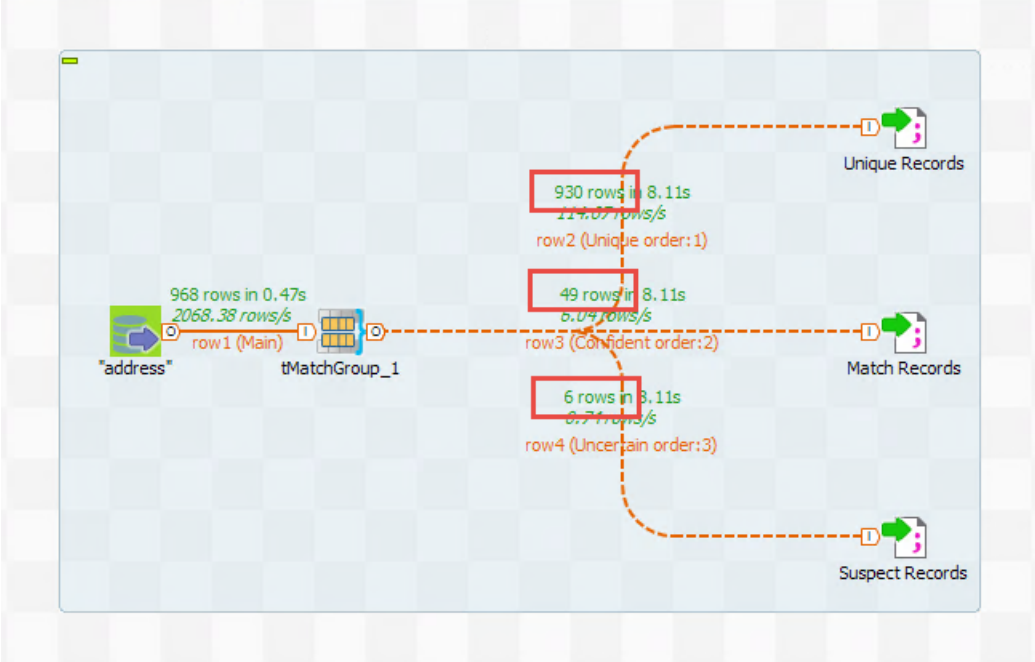


Connect the **Uniques** row to **Unique Records**, the **Matches** row to **Match Records**, and the **Suspects** row to **Suspect Records**.



Running the analysis

1. To run the Job, press **F6**.
2. You can see the number of rows that were extracted in each file.

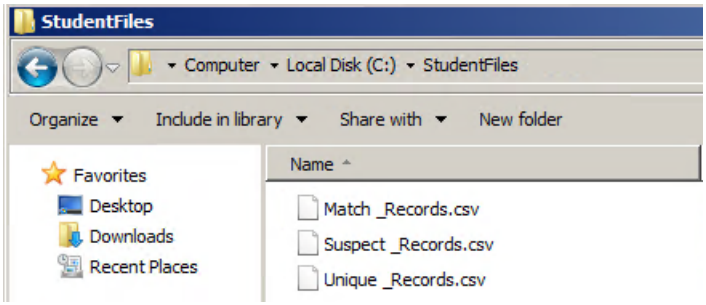


Note: The Match Records file contains the details for each group, meaning that the number of rows equals the number of duplicate records plus the number of survivors (one golden record per group).

Right-click the **tFileOutputDelimited** components, and to see their content, on the contextual menu, select **Data Viewer**.

Address_code	Customer_code	Country_code	Postal_code	City	Address_line	State	GID	GRP_SIZE	MASTER	SCORE	GRP_QUALITY
4095	103	ESP	28016	Madrid	Calle Miguel de Cervantes 39		392b81f6-d421-4a5b-b8fe-58d1fb89dfbd	2	true	1.0	1.0
4014	103	ESP	28016	Madrid	Calle Miguel de Cervantes 39		392b81f6-d421-4a5b-b8fe-58d1fb89dfbd	0	false	1.0	0.0
4095	103	ESP	28016	Madrid	Calle Miguel de Cervantes 39		392b81f6-d421-4a5b-b8fe-58d1fb89dfbd	0	false	1.0	0.0
4086	948	ESP	32558	Cepedelo	Bulevar Garcilaso de la Vega 76		87b9af8f-3d11-4dc6-94be-ffa123fa4bab	3	true	1.0	1.0
4085	948	ESP	32558	Cepedelo	Bulevar Garcilaso de la Vega 76		87b9af8f-3d11-4dc6-94be-ffa123fa4bab	0	false	1.0	0.0
4086	948	ESP	32558	Cepedelo	Bulevar Garcilaso de la Vega 76		87b9af8f-3d11-4dc6-94be-ffa123fa4bab	0	false	1.0	0.0
4087	948	ESP	32558	Cepedelo	Bulevar Garcilaso de la Vega 76		87b9af8f-3d11-4dc6-94be-ffa123fa4bab	0	false	1.0	0.0
4010	62	ESP	28017	Madrid	Avenida Fray Luis de Leon 60		cfce0fc0-a7d3-4712-aad2-d086d2c1245c	2	true	1.0	0.9583333333333334

3. To check the output files, open the **StudentFiles** directory.



Adding a second tMatchGroup component

The purpose of this exercise is to enhance the matching Job results by trying to identity duplicates that may exist across countries (for example, same customer, similar address line, similar city but different country code).

If a bad address association exists in the CRM database, the current Job cannot find it because blocks have been created to limit the duplicate searches per country. Because the search is optimized to consume fewer resources, some duplicates may slip though.

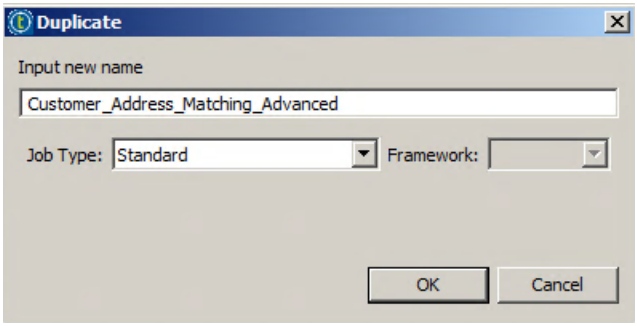
This is why you need a second tMatchGroup component set up without blocking keys. The first tMatchGroup component, optimized with blocking keys, identifies most of the duplicates. The second tMatchGroup component, without blocking keys, analyzes the output of the first component to search for duplicates in survivors and single records.

1. DUPLICATE THE JOB

   To easily compare the results, you must duplicate the Job.

   a. In the **Repository**, right-click the **Customer_Address_Matching** Job, and on the contextual menu, choose **Duplicate**.

   b. in the **Input new name** text box, enter *Customer_Address_Matching_Advanced*



   c. To open the **Customer_Address_Matching_Advanced** Job in the work space, double-click it.

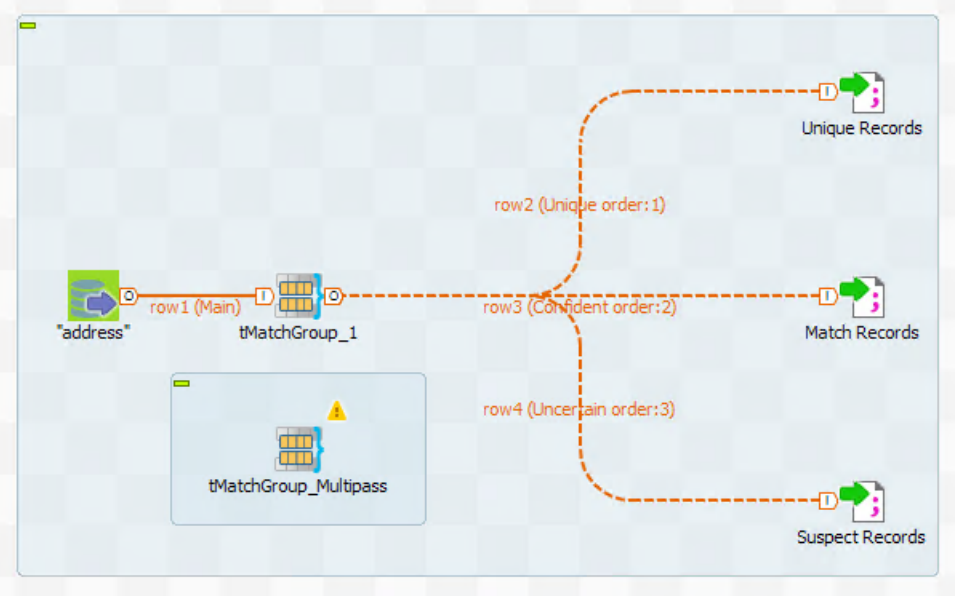
2. ADD A NEW tMatchGroup COMPONENT

   To avoid extra configuration, copy the first tMatchGroup component.

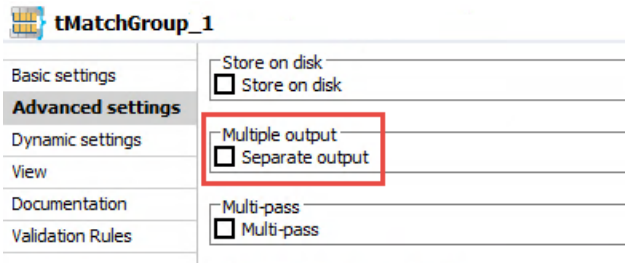
   a. Right-click the first **tMatchGroup** component, and on the contextual menu, choose **Copy**.

   b. In the work space, right-click in an open area, and on the contextual menu, choose **Paste**.

   c. To avoid confusion, rename the second component. Select the **View** tab, and in the **Label format** text box, enter *tMatchGroup_Multipass*

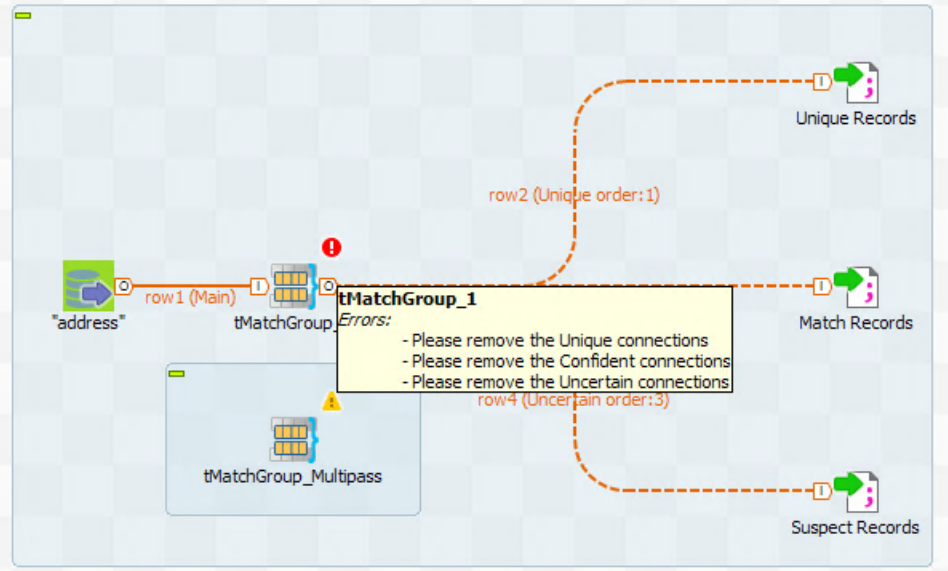


d. Click the primary **tMatchGroup** component, and in the **Component** view, on the **Advanced settings** tab, deselect **Separate output**.



A red warning icon appears next to the component.

e. To display the text of the warning, hover your mouse over the icon.

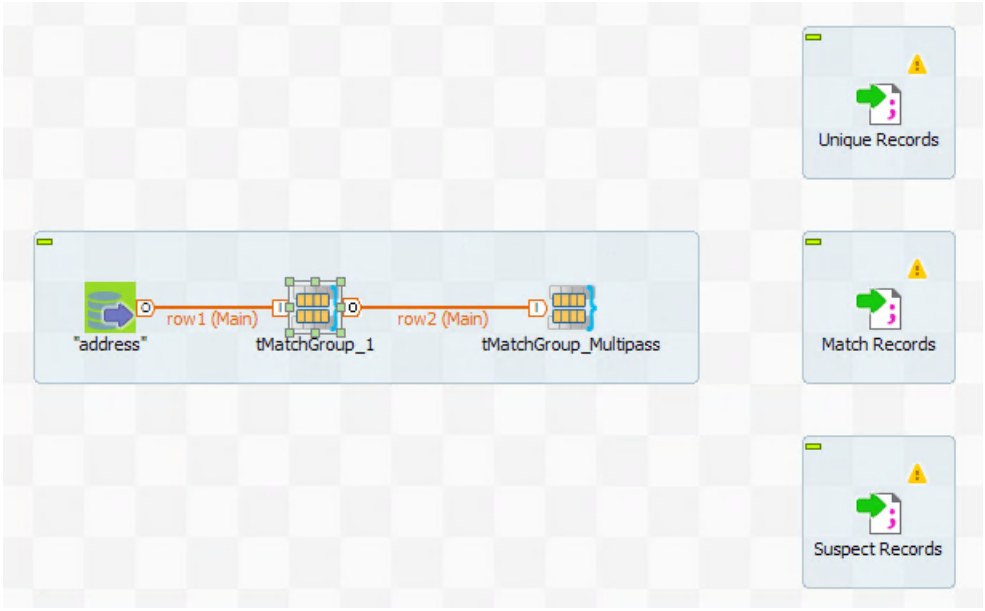


You need to remove the three connections.

f. Move the **tMatchGroup_Multipass** component between the first **tMatchGroup** component and the three **tFileOutputDelimited** components.

Connect the first **tMatchGroup** component to the **tMatchGroup_Multipass** component using the **Main row**.

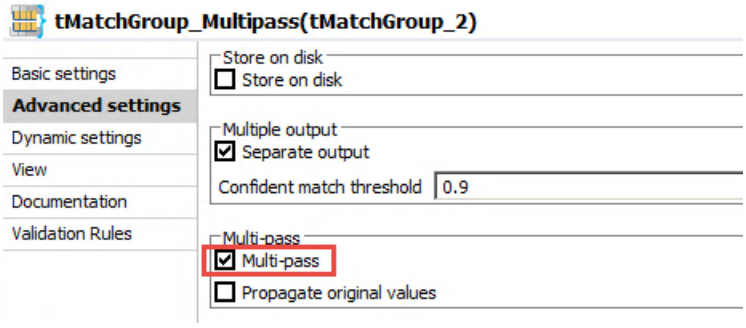
Note: If there is a warning or an error on the component, click the **Sync columns** button to update the schema.



3. SET UP THE tMatchGroup_Multipass COMPONENT

Set up the new component for multi-pass usage.

a. Click the **tMatchGroup_Multipass** component, and in the **Component** view, on the **Advanced settings** tab, select the **Multi-pass** check box.



With the Multi-pass parameter activated, a tMatchGroup component can profile the output of another tMatchGroup component. It searches for duplicates in the single records and survivors sent by the first component.

With the Propagate original values parameter activated, the search covers all records sent by the first component: singles, survivors, and duplicates.

Note: Do not select the **Propagate original values** parameter.

b. To open the configuration wizard, double-click the **tMatchGroup_Multipass** component.

c. To remove the blocking keys, in the **Blocking Selection** section, select the **Country_code** key and click the **[X]** button.

d. To refresh the data preview, in the upper right corner, click the **Chart** button.

Customer_code	Country_code	Postal_code	City	Address_line	State	GID	GRP_SIZE	MASTER	SCORE	GRP_QUALITY
667	USA	15663	Madison	18 Sui Sin Far Street	PA	0b0bf1c4-649e-4b6c-8e8...	2	true	1.0	1.0
667	DEU	15663	Madison	18 Sui Sin Far Street	PA	0b0bf1c4-649e-4b6c-8e8...		false	1.0	0.0
667	USA	15663	Madison	18 Sui Sin Far Street	PA	0b0bf1c4-649e-4b6c-8e8...		false	1.0	0.0
301	GBR	B541	Bristol	28 Osbert Sitwell Street		181d7d02-32af-42a8-ac3...	3	true	1.0	1.0
301	USA	B541	Bristol	28 Osbert Sitwell Street		181d7d02-32af-42a8-ac3...		false	1.0	0.0
301	GBR	B541	Bristol	28 Osbert Sitwell Street		181d7d02-32af-42a8-ac3...		false	0.96875	0.0
301	GBR	B541	Bristol	28 Osbert Sitwell St		181d7d02-32af-42a8-ac3...		false	0.96875	0.0

Some groups contain duplicate records from different countries.

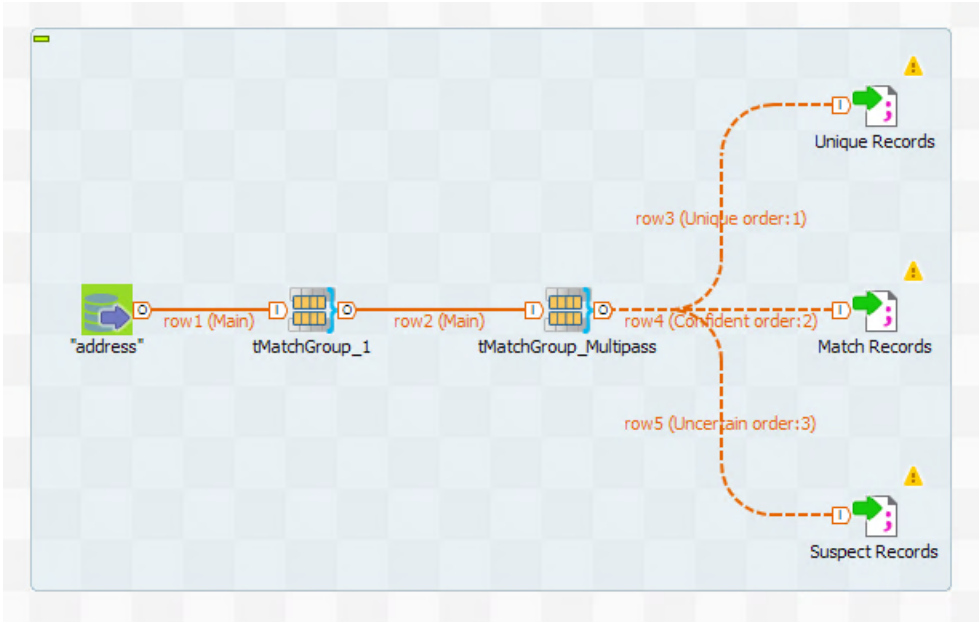
e. To close the wizard, click **OK**.

4. RECONNECT THE tFileOutputDelimitedCOMPONENTS

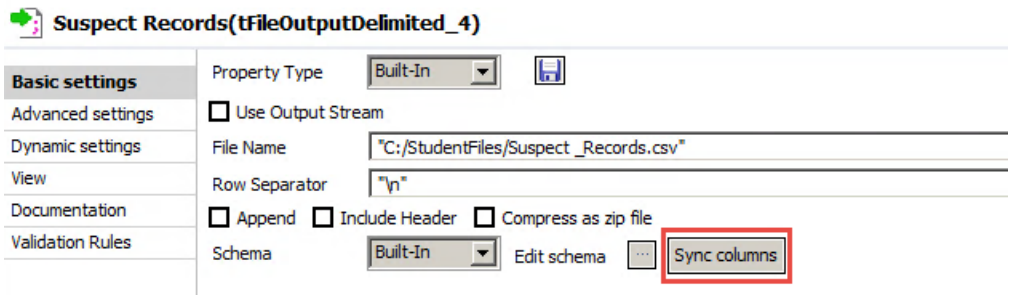
You must reconnect the three tFileOutputDelimited components to the three outputs of the tMatchGroup_Multipass component.

a. In the work space, select the **tMatchGroup_Multipass** component.

Connect the **Uniques** row to **Unique Records**, the **Matches** row to **Match Records**, and the **Suspects** row to **Suspect Records**.



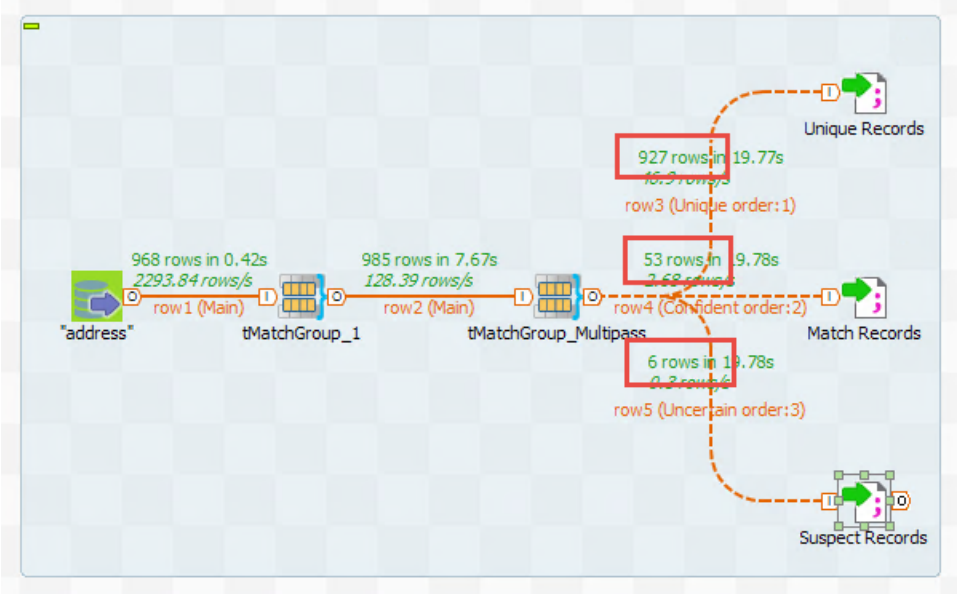
b. The schema of the three tFileOutputDelimited components must be re-synced with the tMatchGroup_Multipass component. On the **Basic settings** tab, click the **Sync columns** button.



Do this three times—once for each tFileOutputDelimited component.

Checking the results

1. To run the Job, press **F6**.
2. Notice the number of rows extracted in each file.



Right-click the **tFileOutputDelimited** components, and to see their content, on the contextual menu, select **Data Viewer**.

3. To access the output files, open the **StudentFiles** directory.

Congratulations! You have completed this comprehensive lesson and it is time to wrap up.

Wrap-Up

In this lesson, you learned the how the match analysis process works. Then you created and ran a match analysis. You played with several parameters to better understand their impact on the detection of match records. Then you saved the configuration of the match analysis in a match rule.

You also created an integration Job and reuse the match rule inside a tMatchGroup component to retrieve the configuration of the analysis. You setup the Job to export the unique, suspect and match rows in three distinct files. And finally you inserted a second tMatchGroup component after the primary one to find more duplicates.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

LESSON 6

Data Privacy

This chapter discusses the following.

Data privacy

Lesson overview

You have been asked to send data extracted from the CRM database to a business intelligence partner agency. The agency provides dashboards to summarize customer activities.

Data analysts need to know the number of customer accounts, contracts, and claims, but they must not access customer names, email addresses, and telephone numbers.

To guarantee data anonymity, you will create an integration Job and use Data Quality components to remove sensitive personal information from extracted data.

In this Job, you will shuffle and mask private data from the customer table.

Objectives

After completing this lesson, you will be able to:

- Shuffle data, creating groups and partitions to keep logical relationships between columns
- Mask data by using masking functions according to column content

The first step is to shuffle customer data.

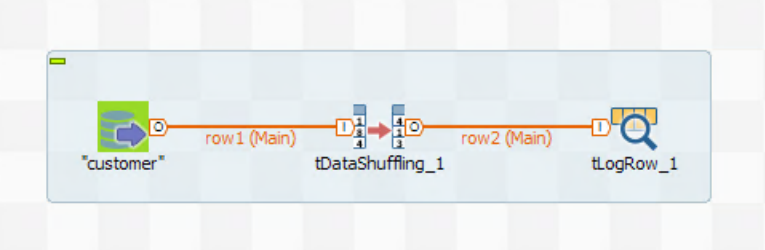
Shuffling data for privacy

In this lesson you will create an integration Job to anonymize customer data from the CRM database. You will load the customer table and use the tDataShuffling Data Quality component to mix up the data.

In the second section, you will use groups and partitions to shuffle data while preserving its logical structure.

- You can group columns. They are shuffled together and values from the same row are always associated.
- You can create partitions. Data is shuffled inside partitions; values from different partitions are never associated.

At the end of this lab, your Job should be as follows.



Creating a Job

First you must create a new integration Job.

Note: To avoid repetition, tasks you have already learned, such as creating a new Job or connecting a database, are summarized. To see detailed instructions that may include screenshots, expand the sections.

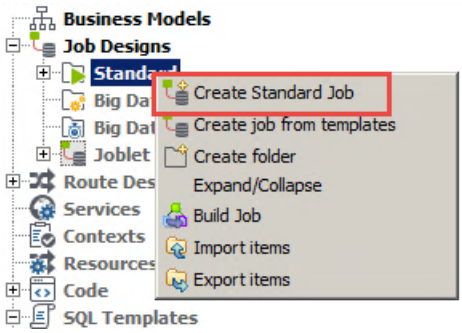
The Job you are about to create uses:

- A tMysqlInput component for extracting data from the CRM database
- A tDataShuffling component for shuffling data
- A tLogRow components for displaying output of the tDataShuffling component on the Run tab

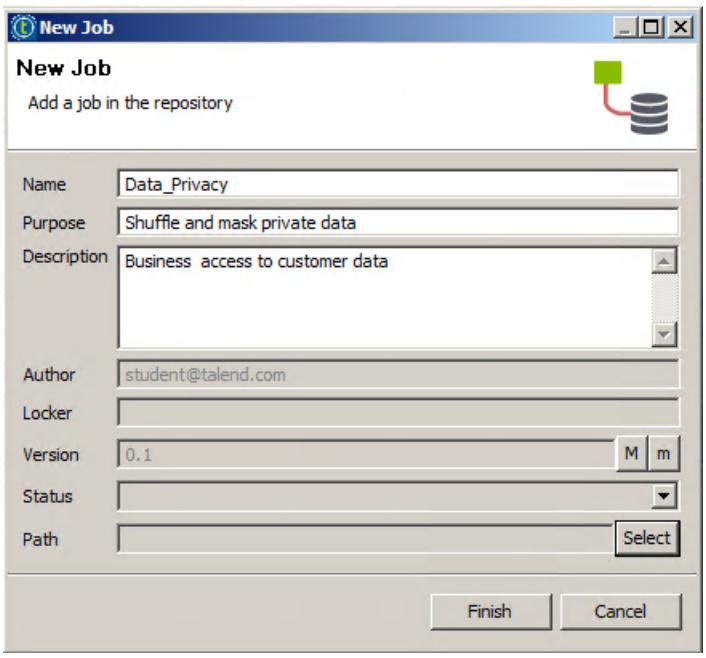
1. CREATE A JOB

   Create a Job and name it *Data_Privacy*.

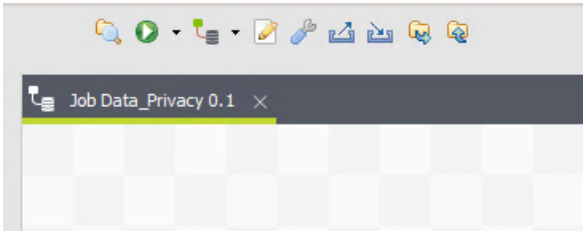
   a. In the **Repository**, expand **Job Designs**, right-click **Standard**, and on the contextual menu, choose **Create Standard Job**.



   b. Fill in the **Name**, **Purpose**, and **Description** text boxes and click **Finish**.



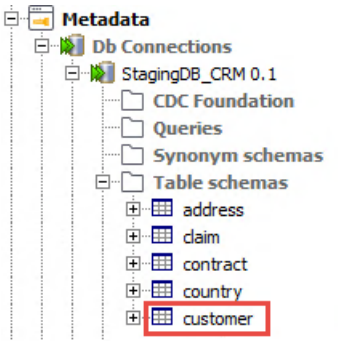
The Job opens in the work space.



2. ADD THE tMysqlInput COMPONENT

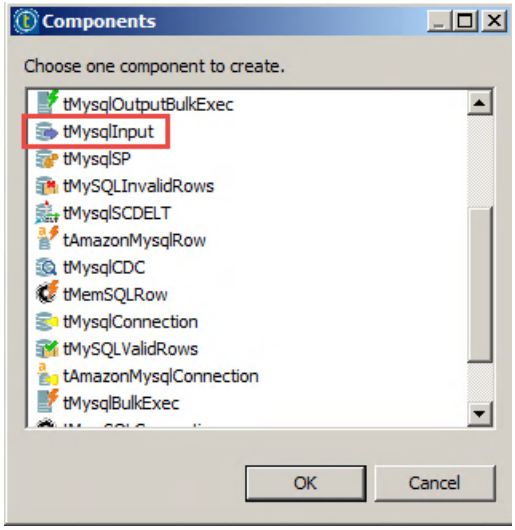
   You must first connect to the database using a tMysqlInput component.

   a. In the **Repository**, go to **Metadata>Db Connections>StagingDB_CRM>Table schemas**.

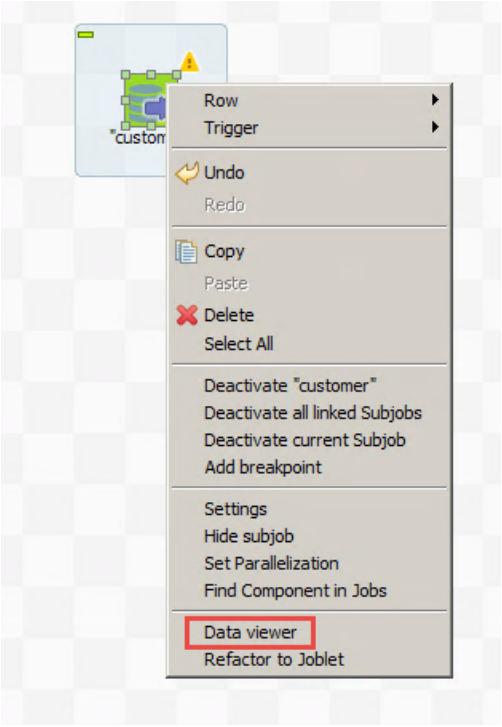


   b. Click the **customer** table and place it in the work space.

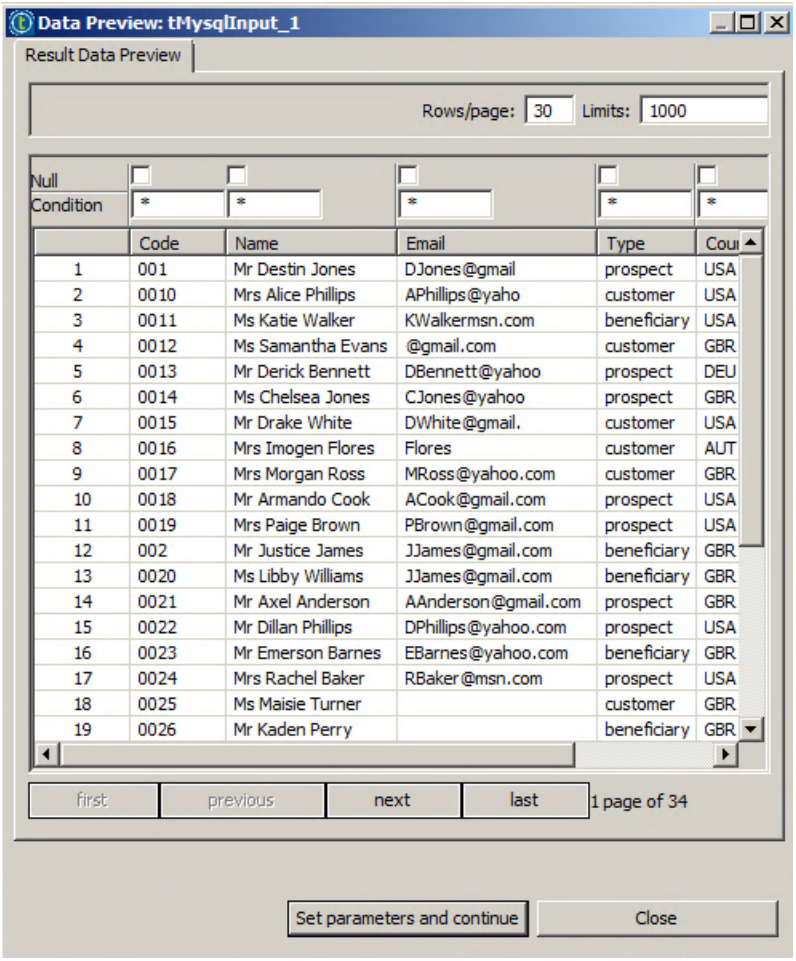
c. Select **tMysqlInput** and click **OK**.



d. The component is automatically set up to extract data from the customer table. To check the connection, right-click the component, and on the contextual menu, choose **Data viewer**.



e. The first rows of the table are displayed in the Data Preview window.



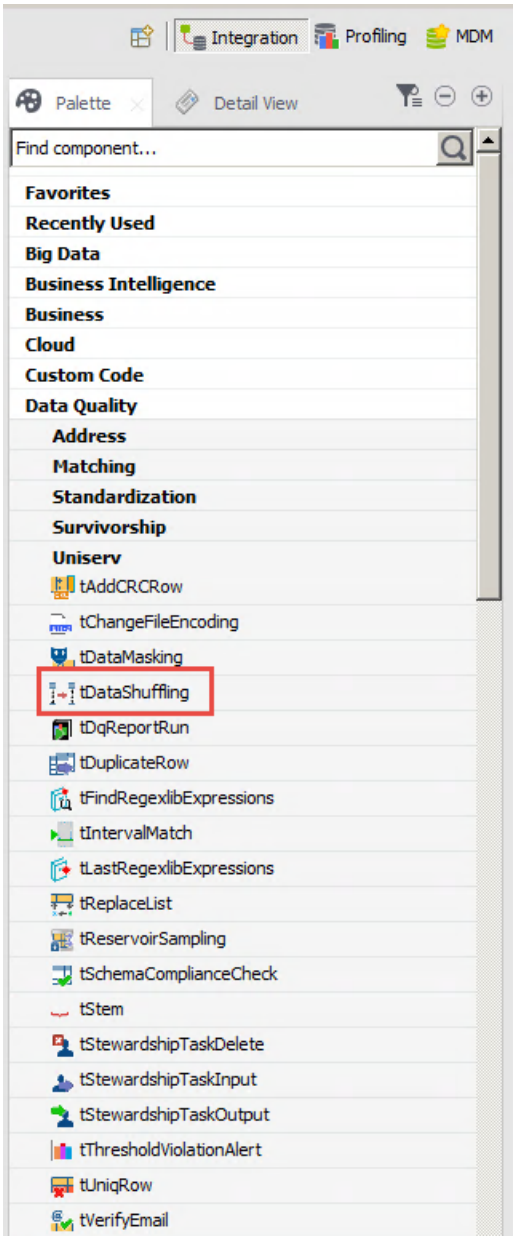
	Code	Name	Email	Type	Cou
1	001	Mr Destin Jones	DJones@gmail	prospect	USA
2	0010	Mrs Alice Phillips	APhillips@yaho	customer	USA
3	0011	Ms Katie Walker	KWalkermsn.com	beneficiary	USA
4	0012	Ms Samantha Evans	@gmail.com	customer	GBR
5	0013	Mr Derick Bennett	DBennett@yahoo	prospect	DEU
6	0014	Ms Chelsea Jones	CJones@yahoo	prospect	GBR
7	0015	Mr Drake White	DWhite@gmail.	customer	USA
8	0016	Mrs Imogen Flores	Flores	customer	AUT
9	0017	Mrs Morgan Ross	MRoss@yahoo.com	customer	GBR
10	0018	Mr Armando Cook	ACook@gmail.com	prospect	USA
11	0019	Mrs Paige Brown	PBrown@gmail.com	prospect	USA
12	002	Mr Justice James	JJames@gmail.com	beneficiary	GBR
13	0020	Ms Libby Williams	JJames@gmail.com	beneficiary	GBR
14	0021	Mr Axel Anderson	AAnderson@gmail.com	prospect	GBR
15	0022	Mr Dillan Phillips	DPhillips@yahoo.com	prospect	USA
16	0023	Mr Emerson Barnes	EBarnes@yahoo.com	beneficiary	GBR
17	0024	Mrs Rachel Baker	RBaker@msn.com	prospect	USA
18	0025	Ms Maisie Turner		customer	GBR
19	0026	Mr Kaden Perry		beneficiary	GBR

Click **Close**.

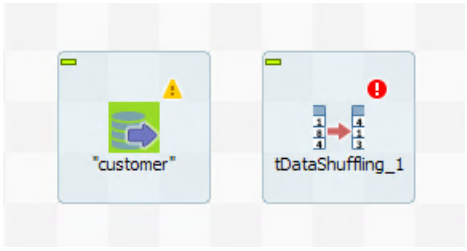
3. ADD A tDataShuffling COMPONENT

   Add a data shuffling component.

a. To display the DQ components, on the **Palette**, expand the **Data Quality** section.



b. Drag the **tDataShuffling** component into the work area.

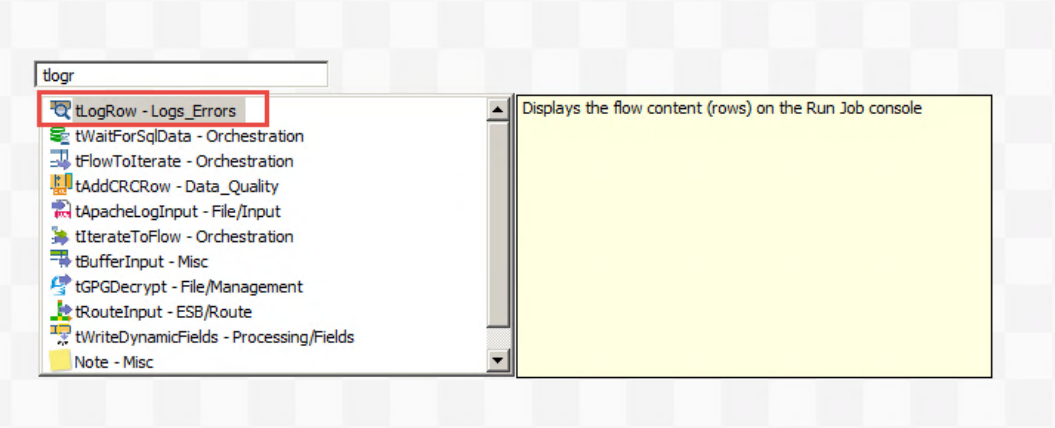


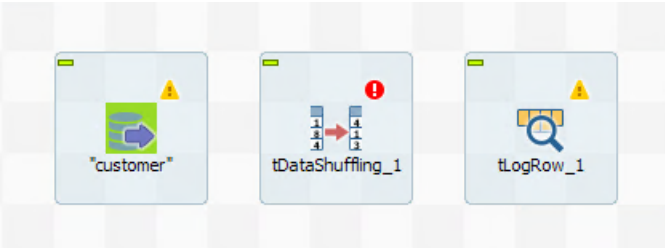
4. ADD A tLogRow COMPONENT

a. In the work space, click to the right of the tDataShuffling component.

Enter the first letters of the *tLogRow* component name.

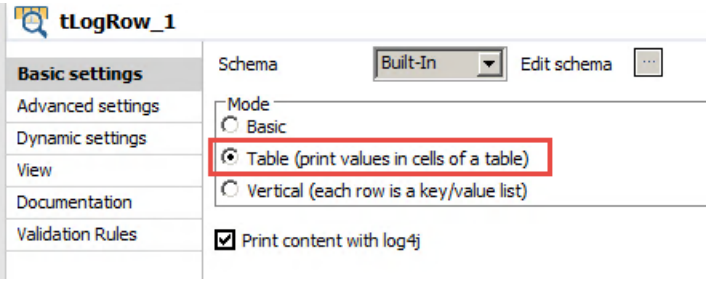
On the list of compatible components, double-click **tLogRow**.





In the **Component** view of the **tLogRow**, below the work space, update the component settings.

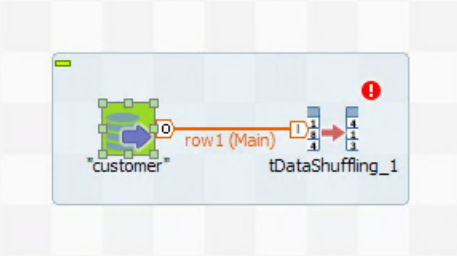
For a clear view of the output data, on the **Basic settings** tab, for **Mode**, select **Table**.



5. SET UP THE tDataShuffling COMPONENT

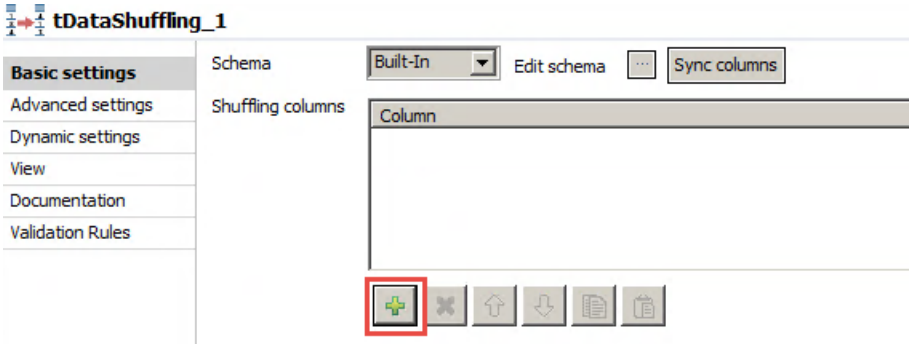
You have to select shuffling columns. To create accurate tables and charts, the dashboard designers need the actual customer codes, types, and country associations. Other fields in the row contain personal data and must be shuffled.

a. Connect **tMysqlInput** to **tDataShuffling** using the **Main row**.

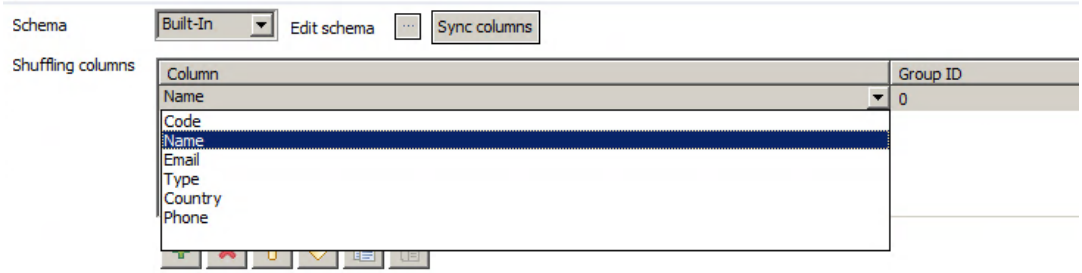


b. To display the **tDataShuffling** settings, double-click the component. Display the **Basic settings** tab.

c. To add a row to the **Shuffling columns** table, click the **plus symbol (+)**.

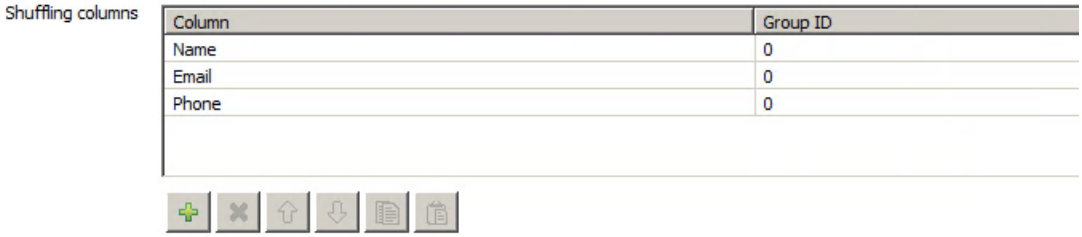


On the **Column** drop-down list, select **Name**.

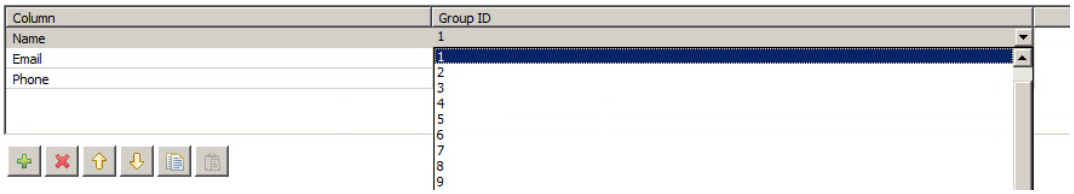


Use the same process to add the **Email** and **Phone** columns.

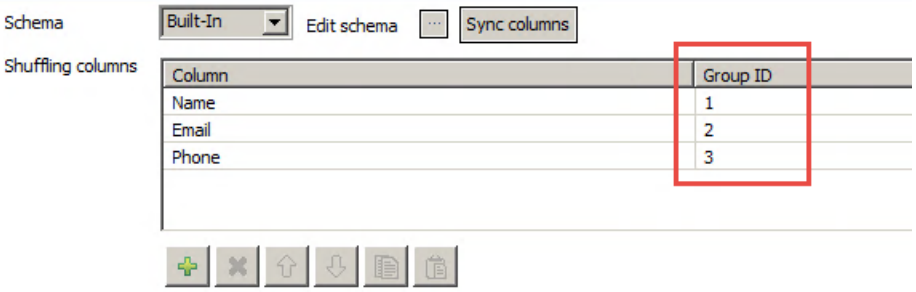
d. By default, all new columns are included in Group 0.



This is a bypass group; all of its columns are not shuffled.
Use the drop-down list to change the **Group ID**.



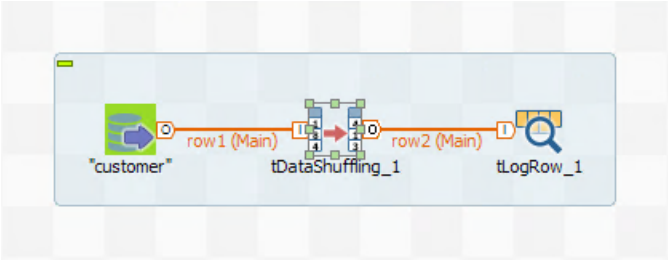
Select a different **Group ID** for each column.



6. CONNECT THE tLogRow COMPONENT AND RUN THE JOB

   Connect all the components before running the Job.

   a. Connect **tDataShuffling** to **tLogRow** using the **Main** row.

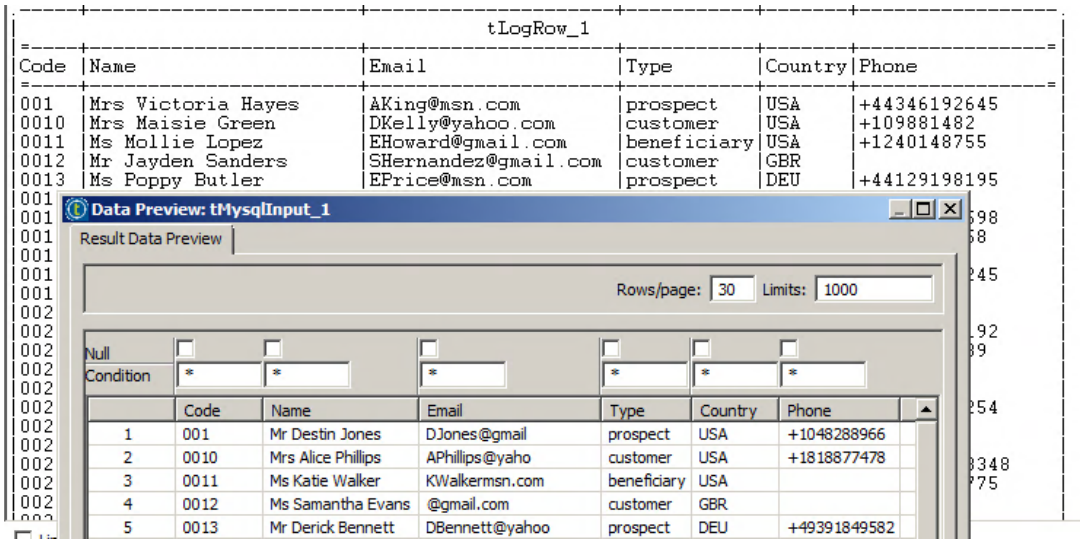


   b. To run the Job, press the **F6** key.

   c. The results are displayed on the Run tab.

```
.-----+----------------------+--------------------+-----------+-------+----------------.
|                                        tLogRow_1                                       |
|=----+----------------------+--------------------+-----------+-------+----------------=|
|Code |Name                  |Email               |Type       |Country|Phone           |
|=----+----------------------+--------------------+-----------+-------+----------------=|
|001  |Mrs Victoria Hayes    |AKing@msn.com       |prospect   |USA    |+44346192645    |
|0010 |Mrs Maisie Green      |DKelly@yahoo.com    |customer   |USA    |+109881482      |
|0011 |Ms Mollie Lopez       |EHoward@gmail.com   |beneficiary|USA    |+1240148755     |
|0012 |Mr Jayden Sanders     |SHernandez@gmail.com|customer   |GBR    |                |
|0013 |Ms Poppy Butler       |EPrice@msn.com      |prospect   |DEU    |+44129198195    |
|0014 |Mr Jayson Hernandez   |ACook@yahoo.com     |prospect   |GBR    |                |
|0015 |Mr Jovan Baker        |EWilliams@yahoo.com |customer   |USA    |+34152610598    |
|0016 |Mr Josiah Young       |JBaker@gmail.com    |customer   |AUT    |+1512225758     |
|0017 |Ms Rachel Phillips    |ALopez@msn.com      |customer   |GBR    |                |
|0018 |Mr Jude Anderson      |KLee@msn.com        |prospect   |USA    |+49748835245    |
|0019 |Mrs Millie Scott      |DPhillips@yahoo.com |prospect   |USA    |                |
```

   To easily compare data, display the **Data Preview** from the **customer** table and the **tLogRow** output side by side.



You can see that the Code, Type, and Country columns were not shuffled, while the personal data was mixed up.

This shuffling method caused some inconsistencies. For instance, the telephone prefixes vary within the same country. Email addresses do not have anything to do with the associated customer names.

These inconsistencies can be corrected using specific group and partition settings.

Designating groups

Groups can help preserve value association between fields in the same row. Columns that belong to the same group are connected and their values are shuffled together.

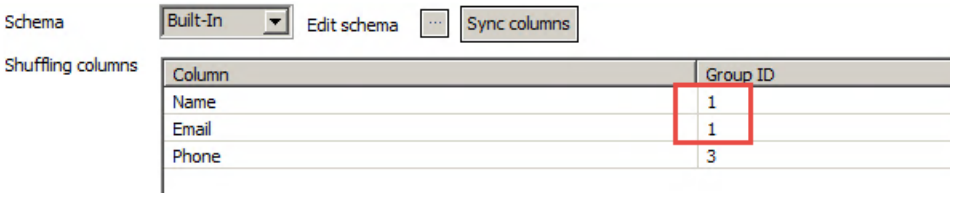
To preserve the association between the Name and Email columns, select the same group for these columns.

1. UPDATE THE GROUPS CONFIGURATION

   To display the **Component** view, double-click **tDataShuffling**.

   a. Display the **Basic Settings** tab.

   b. Give the **Email** and **Name** column the same **GroupID**.



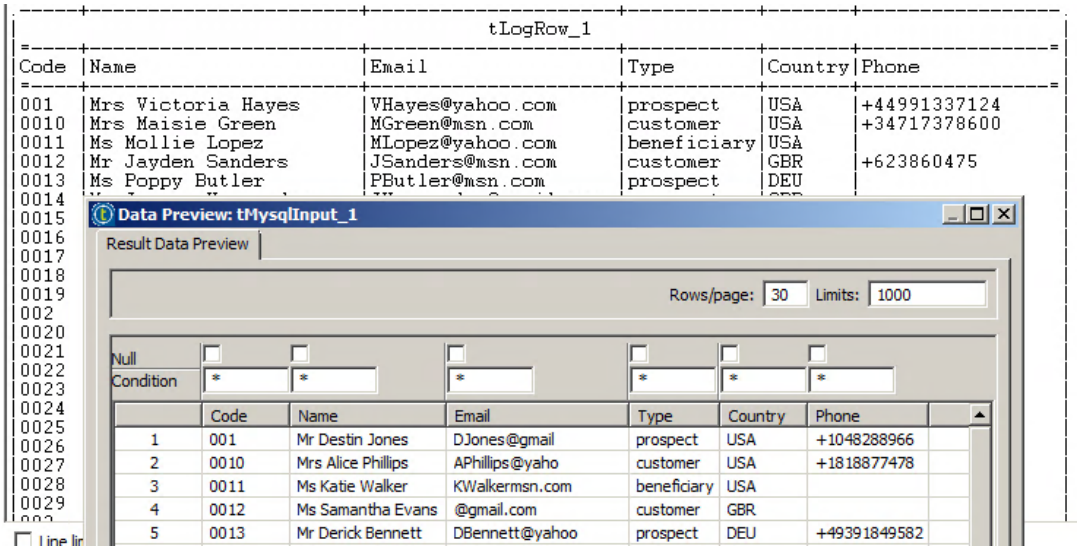
   Note: You must select the group ID in the drop-down list; you cannot type it.

2. RUN THE JOB

   To better understand the influence of the groups, run the Job.

a. Press the **F6** key.

b. View the Job results on the **Run** tab and again compare the output of the Job with the input data.



After the shuffling process, values in the Name and Email columns are still associated.

Designating partitions

Partitions can help preserve dependencies between columns. You can create a partition for a specific country or customer type. Data is shuffled within partitions, and values from different partitions are never associated.

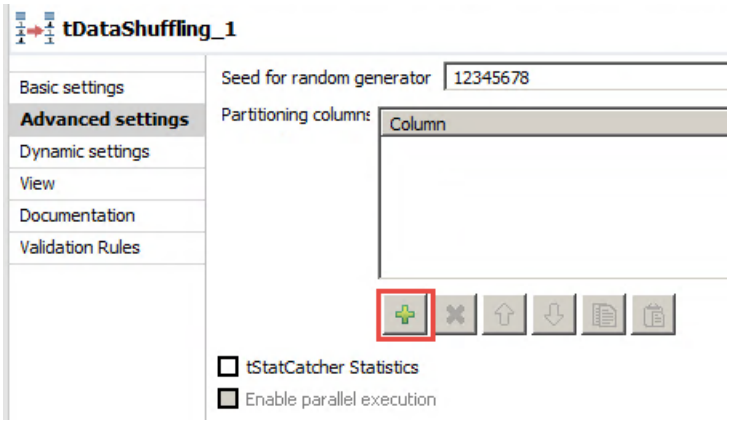
To preserve dependency between a phone number and a country, you must create a partition.

1. CREATE A PARTITION

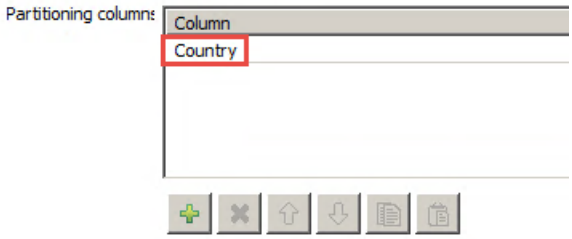
   Create a partition on the Advanced Settings tab of the Component view.

   a. Double-click **tDataShuffling** to display the **Component** view, then select the **Advanced Settings** tab.

   b. To create a partition in the **Partitioning columns** table, click the **plus symbol (+)**.



c. On the drop-down **Column** list, select **Country**.

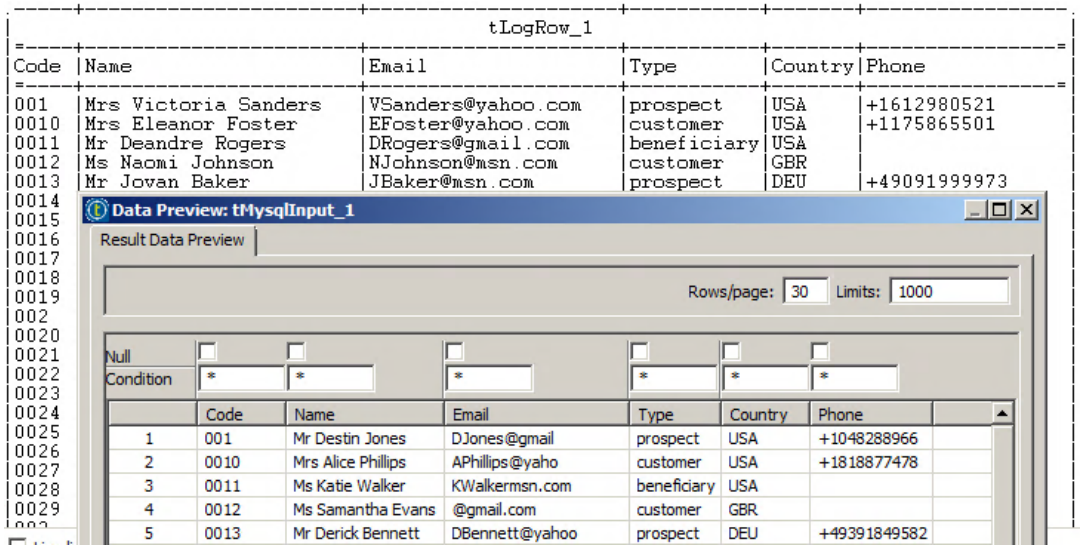


4. RUN THE JOB

To better understand the impact of the partition, run the Job.

a. Press the **F6** key.

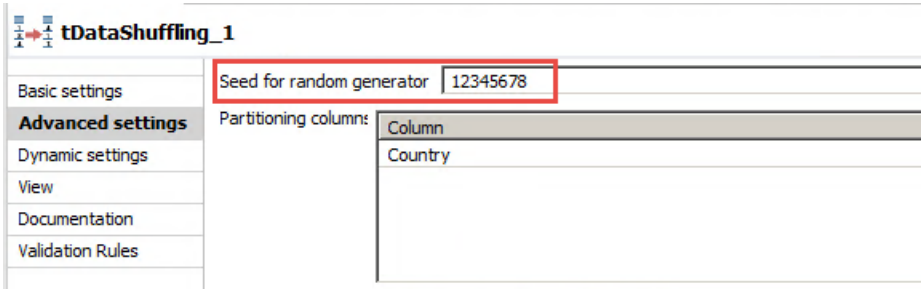
b. On the **Run** tab, view the results and compare the results of the Job with the input data.



The shuffling process preserves the dependency between the country code and telephone prefix. This way, the telephone prefix*+1* is always associated with the country *USA*, *+33* with *FRA*, +49 with *DEU*, and so on.

Note: Because the data is shuffled, your results may vary.

If you rerun the Job without changing any settings, you get the same results. This is due to the parameter in the Seed for random generator box on the Advanced Settings tab of the Component view.



- To shuffle data in the same order in each execution of the Job, keep the same value in this text box.
- To shuffle data in a different order, change the value.
- To shuffle data in random order each time you execute the Job, leave the box empty.

Another way to preserve data privacy is to mask private data.

Masking data for privacy

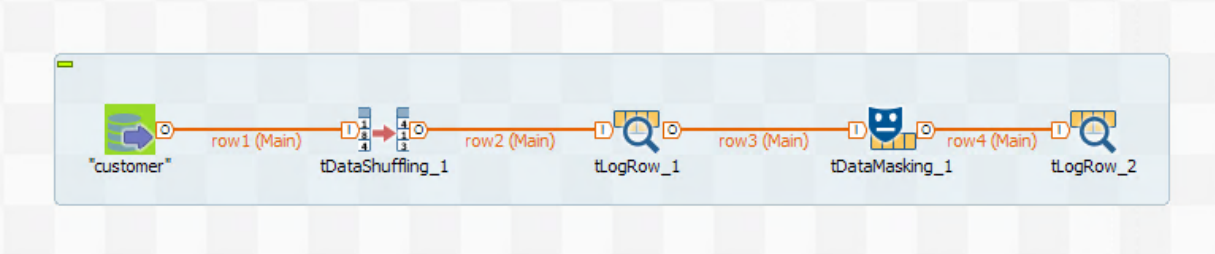
In this lesson you will use a second Data Quality component: tDataMasking.

This component hides original data by simply using random characters or more advanced masking functions. Masked data continues looking and consistent and remains usable for purposes such as business intelligence and software demonstration.

In this exercise, you will hide customer names using random characters. For that, you will add a tDataMasking component at the end of the shuffle Job..

Then, in the second section, you will explore several masking functions to hide names, email addresses, and phone numbers.

At the end of this lab, your Job should be as follows.



Masking data with random characters

You must add the following components to the *Data_Privacy* Job:

- A tDataMasking component for masking data
- A tLogRow component for displaying the output of the tDataMasking component on the Run tab

1. PLACE AND LINK THE COMPONENTS

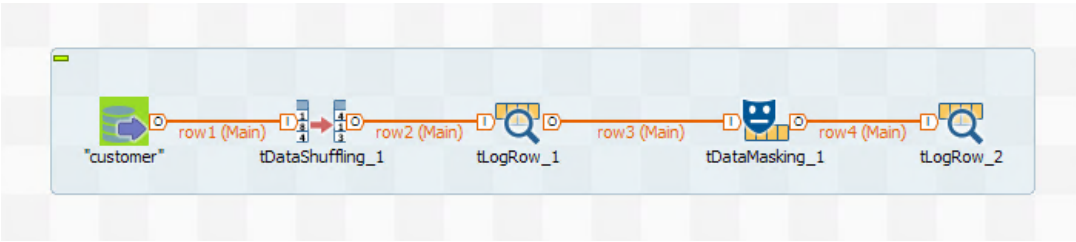
   Place the new components to the right of the last component in the *Data_Privacy* Job, a tLogRow.

   a. Drag and drop **tDataMasking** from the **Data Quality** section of the **Palette** to the right side of the **tLogRow**.

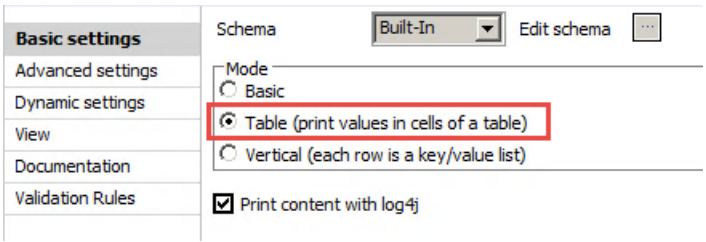
   b. Drag and drop a second **tLogRow** component from the **Recently Used** section of the **Palette** to the right side of **tDataMasking**.

   c. Connect the first **tLogRow** component to **tDataMasking** using the **Main** row.

   d. Connect **tDataMasking** to the second **tLogRow** component using the **Main** row.



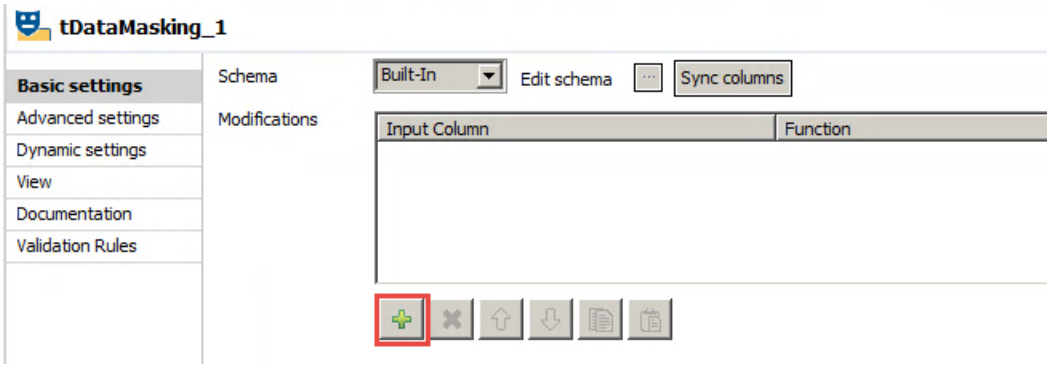
   e. Double-click the second **tLogRow**, and on the **Basic settings** tab, in **Mode**, select **Table**.



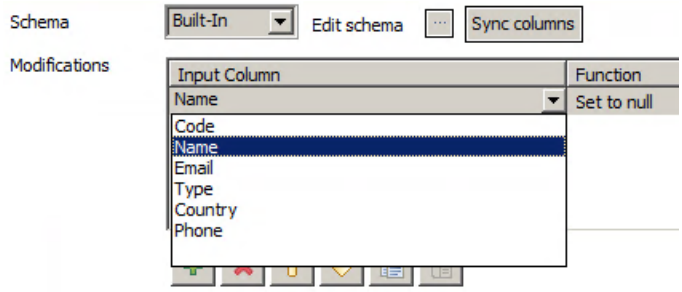
2. SET UP THE tDataMasking COMPONENT

To access the component view, double-click **tDataMasking**.

a. To add a new column to the **Modifications** table, click the **plus symbol (+)**.

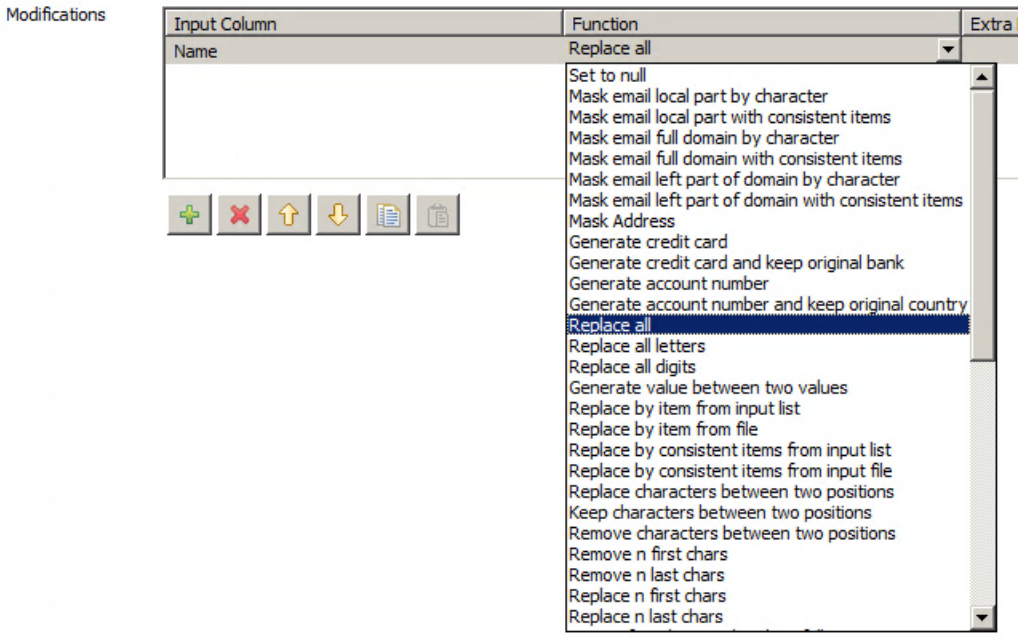


b. On the drop-down list, select the **Name** column.



c. The drop-down list in the Function column displays all the data-masking functions. Some can be used with any kind of content, while others are designed for a specific format, such as email address, credit card number, or Social Security number (SSN).

To mask all the content in the Name column using random characters, select **Replace all**.

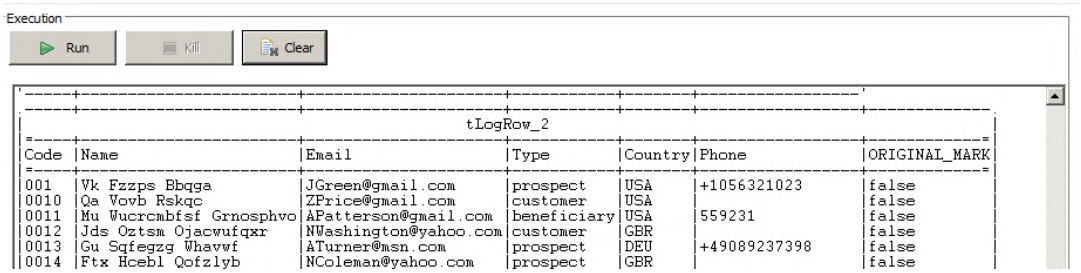


3. RUN THE JOB

   Run the Job, to shuffle and mask data.

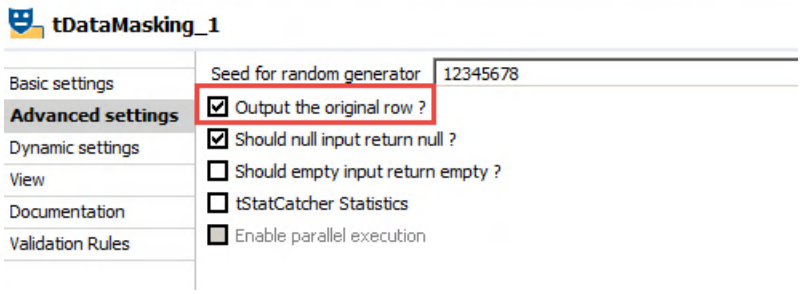
   a. To run the Job, press the **F6** key.

   b. The results are displayed on the **Run** tab.



   Notice the new column, **ORIGINAL_MARK**. It defines the row status and can be useful when original rows are added to the output.

   c. To activate this option, double-click **tDataMasking**, display the **Advanced settings** tab, and select **Output the original row**.



d. Run the Job and again check the results on the **Run** tab.

tLogRow_2

Code	Name	Email	Type	Country	Phone	ORIGINAL_MARK
001	Mr Josue Green	JGreen@gmail.com	prospect	USA	+1056321023	true
001	Vk Fzzps Bbqga	JGreen@gmail.com	prospect	USA	+1056321023	false
0010	Ms Zara Price	ZPrice@gmail.com	customer	USA		true
0010	Qa Vovb Rskqc	ZPrice@gmail.com	customer	USA		false
0011	Mr Alexzander Patterson	APatterson@gmail.com	beneficiary	USA	559231	true
0011	Mu Wucrcmbfsf Grnosphvo	APatterson@gmail.com	beneficiary	USA	559231	false
0012	Mrs Naomi Washington	NWashington@yahoo.com	customer	GBR		true
0012	Jds Oztsm Ojacwufqxr	NWashington@yahoo.com	customer	GBR		false

There are two rows for each customer: the original version (true) and the masked version (false). This is useful if you want to compare masked values with original values.

c. Before continuing, deselect **Output the original row**. You do not need this because the original rows are already displayed on the Run tab, in the output logs of the first tLogRow.

Using other masking functions

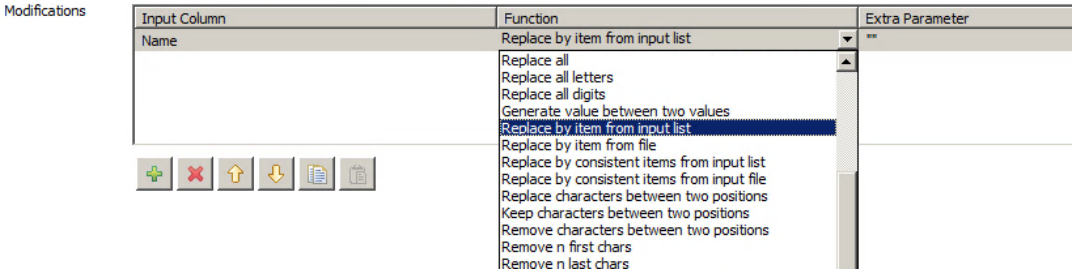
In this section, you will use several masking functions adapted to the input data.

1. USING AN INPUT LIST OF NAMES

   Customer names composed of random characters are not consistent enough to use in a software demonstration. The Job can be set up to pick up anonymous names on a list.

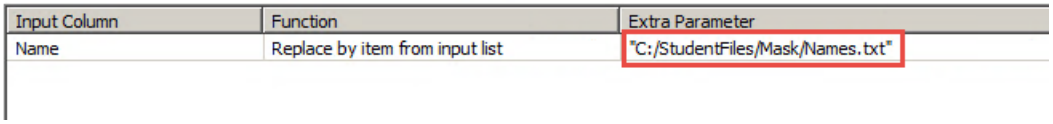
   a. To display the Component view, double click **tDataMasking**.

   b. On the drop-down list, in the **Function** column, select **Replace by item from input list**.

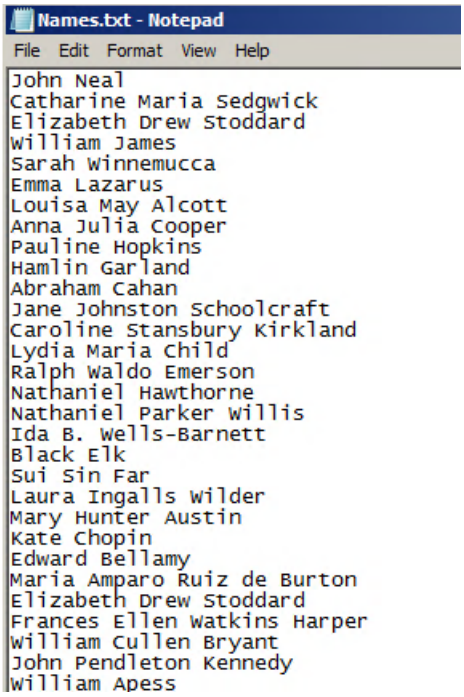


The list of anonymous names is stored on the VM in the StudentFiles directory in a text file.

   c. In the **Extra Parameter** column, enter *"C:/StudentFiles/Mask/Names.txt"*



d. Using Windows Explorer, find and double-click the **Names.txt** file.



The file contains a list of names used to randomly replace customer names.

e. Run the Job and examine the results.

tLogRow_2

Code	Name	Email	Type	Country	Phone	ORIGINAL_MARK
001	Edward Bellamy	JGreen@gmail.com	prospect	USA	+1056321023	false
0010	Elizabeth Drew Stoddard	ZPrice@gmail.com	customer	USA		false
0011	Emma Lazarus	APatterson@gmail.com	beneficiary	USA	559231	false
0012	Catharine Maria Sedgwick	NWashington@yahoo.com	customer	GBR		false
0013	Nathaniel Hawthorne	ATurner@msn.com	prospect	DEU	+49089237398	false
0014	Jane Johnston Schoolcraft	NColeman@yahoo.com	prospect	GBR		false

Customer names were randomly replaced with names from the text file.

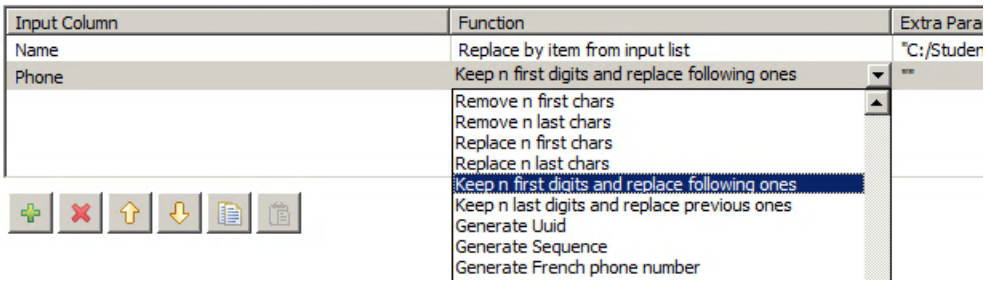
2. KEEPING THE FIRST DIGITS OF THE PHONE NUMBER

To preserve the phone number prefix, you can keep the first three digits and mask the rest.

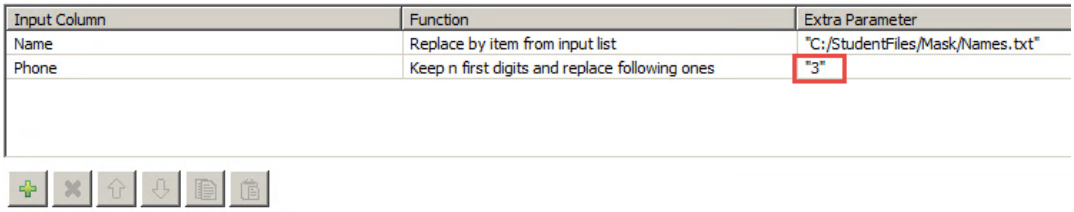
a. To display the **Component** view, double-click **tDataMasking**.

b. To add the column to the **Modifications** table, click the **plus symbol (+).** On the drop-down list, select the **Phone** column.

c. Select **Keep n first digits and replace following ones**.



d. In the **Extra Parameter** column, enter *"3"*



e. Run the Job and examine the results.

tLogRow_2

Code	Name	Email	Type	Country	Phone	ORIGINAL_MARK
001	Edward Bellamy	JGreen@gmail.com	prospect	USA	+1052873888	false
0010	Emma Lazarus	ZPrice@gmail.com	customer	USA		false
0011	John Neal	APatterson@gmail.com	beneficiary	USA	559283	false
0012	Caroline Stansbury Kirkland	NWashington@yahoo.com	customer	GBR		false
0013	Maria Amparo Ruiz de Burton	ATurner@msn.com	prospect	DEU	+49080533052	false
0014	Pauline Hopkins	NColeman@yahoo.com	prospect	GBR		false
0015	Black Elk	KCollins@gmail.com	customer	USA	+1487340830	false
0016	Nathaniel Hawthorne	Flores	customer	AUT		false

Any original prefixes were preserved and the other digits were randomly replaced.

3. MASKING EMAIL ADDRESSES

Several functions are available for masking email addresses. Some were designed for the complete address and others for the local or domain segment (assuming that email addresses fit the standard format, *local@domain*).

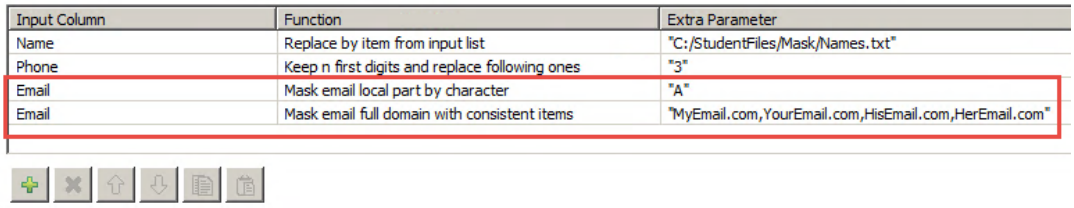
You will use two functions: one for the local segment and one for the domain.

a. To display the **Component** view, double-click **tDataMasking**.

b. To add two columns to the **Modification** table, click twice on the **plus symbol (+)** and select **Email** for both.

c. For the first one, select **Mask email local part by character**, and in the **Extra Parameter** column, enter *"A"*

d. For the second, select **Mask email full domain with consistent items**, and in the **Extra Parameter** column, type *"MyEmail.com,YourEmail.com,HisEmail.com,HerEmail.com"*



e. Run the Job and examine the results.

tLogRow_2

Code	Name	Email	Type	Country	Phone	ORIGINAL_MARK
001	Edward Bellamy	AAAAAA@HisEmail.com	prospect	USA	+1058738888	false
0010	John Neal	AAAAAA@HerEmail.com	customer	USA		false
0011	John Neal	AAAAAAAAAA@YourEmail.com	beneficiary	USA	559260	false
0012	Laura Ingalls Wilder	AAAAAAAAAAA@HisEmail.com	customer	GBR		false
0013	William James	AAAAAAA@MyEmail.com	prospect	DEU	+49085283408	false
0014	Black Elk	AAAAAAAA@HisEmail.com	prospect	GBR		false
0015	Nathaniel Hawthorne	AAAAAAAA@MyEmail.com	customer	USA	+1487386631	false

The email addresses are still in a usable format but completely anonymized. Your data is ready to be extracted and sent to the business intelligence agency.

You have finished this section. Before you continue, read the wrap-up.

Wrap-Up

In this lesson, you learned how to preserve data privacy using Data Quality components. You created an integration Job to anonymize data extracted from the customer table. With the tDataShuffling component, you shuffled data while preserving its logical structure. With the tDataMasking component, you masked data using different functions for names, phone numbers, and email addresses.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

LESSON 7

Reports and Data Quality Portal

This chapter discusses the following.

Reports and the Data Quality portal

Lesson overview

To easily monitor your data quality, in Talend Studio, you can set up and generate reports based on analyses you created.

By default, a report is generated in two steps:

- Report data is stored in a database
- A PDF file is created in the workspace file system

Business users can monitor data quality through a Web interface tool, Talend Data Quality Portal (DQP). To display reports, the portal uses data saved in the database. This can be useful because not everyone has access to or knows how to use Studio, but people need to stay informed and influence data quality decisions.

Objectives

After completing this section, you will be able to:

- Configure a database connection for reporting purposes
- In Studio, create basic reports based on saved analyses
- Create reports that show the evolution of data quality over time
- Launch and access DQP
- Run reports from DQP

First you will get familiar with the data quality database.

Configuring the Data Quality database

Overview

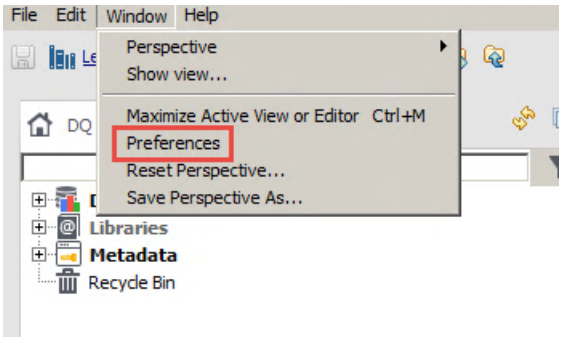
Before a report is displayed in DQP, analytical data is fetched from a database. Data is saved to the database when the report is generated.

The training environment includes a preconfigured connection to a remote database—the data quality datamart. The first step is to connect Talend Studio to this datamart.

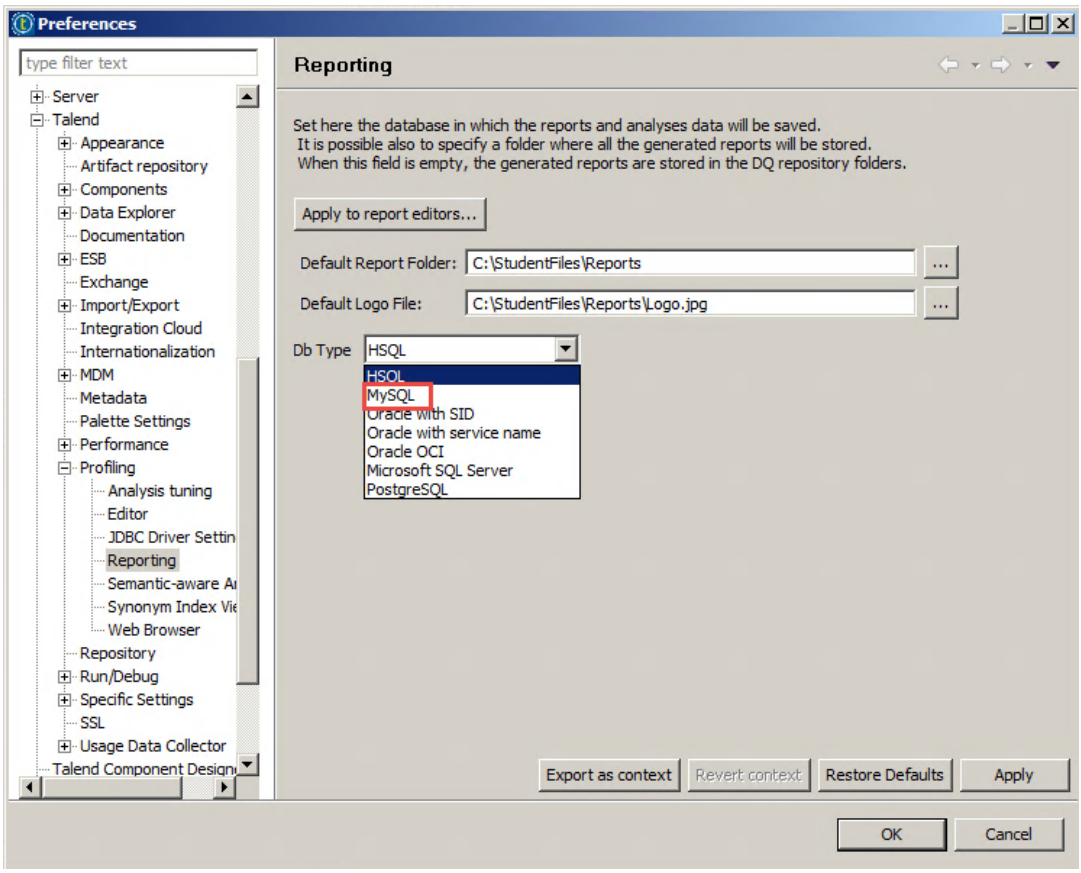
Connecting to the datamart

Before running reports, you need to specify a few preferences. This one-time configuration task includes changing the database type for saving reports from the default, embedded HSQL, to MySQL.

1. CONNECT TO THE DATAMART
    a. Click the **Window** menu and select **Preferences**.



    b. Navigate to **Talend>Profiling>Reporting**.
    c. When reports are generated, they are saved locally on the workspace directory tree. To save them in a more easily accessible directory, next to the **Default Report Folder** box, click the **ellipsis (...)** button and select **C:\StudentFiles\Reports**.
    d. To automatically add the same logo to all generated reports, next to the **Default Logo File** text box, click the **ellipsis (...)** button and select the *C:\StudentFiles\Reports\Logo.jpg* file.
    e. Change **Db Type** to **MySQL**.



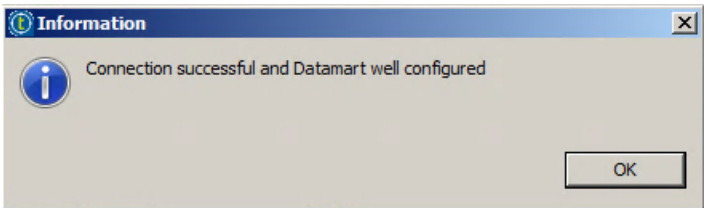
The connection settings for the datamart appearbelow.

f. Use the default configuration. You must update only the user and password; for both, enter *root*. View the settings summary and use these parameters:

- **Db Type:** *MySQL*
- **Db Version:** *MYSQL_5*
- **Host:** *localhost*
- **Port:** *3306*
- **Db Name:** *talend_dq*
- **User:** *root*
- **Password:** *root*
- **Url:** *jdbc:mysql://localhost:3306/talend_dq?characterEncoding=UTF8*

g. Click **OK**.

h. The first time you connect to the datamart, you see the Information window.

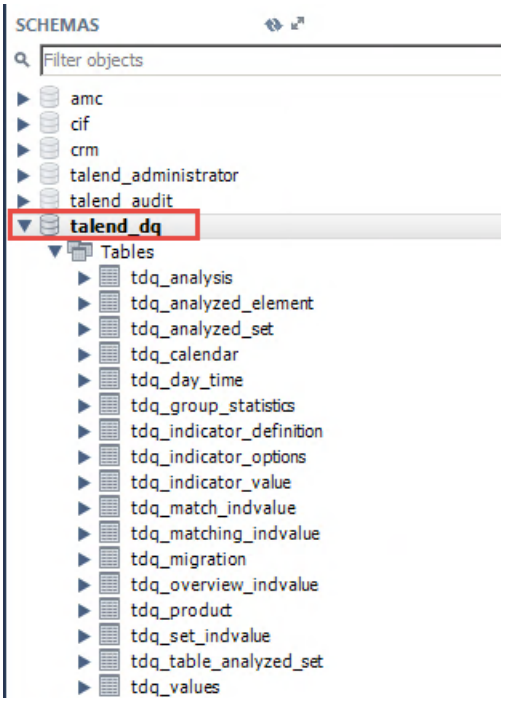


Click **OK**.

Displaying database content

You can use MySQL WorkBench to display the datamart content.

1. BROWSE THE DATAMART
   a. In MySQL WorkBench, in the **Navigator** pane on the left, in **SCHEMAS**, expand **talend_dq**.



   b. View the names of the various tables in the datamart. The details are not important right now.

In Talend Studio you will create a report.

Creating a report

Overview

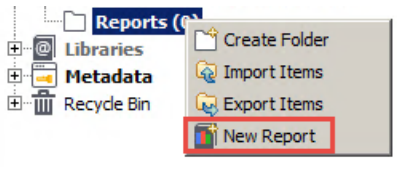
In this lesson, you will generate a report for the table match analysis that you created earlier. You will experiment with different settings before generating a PDF file.

Setting up a report

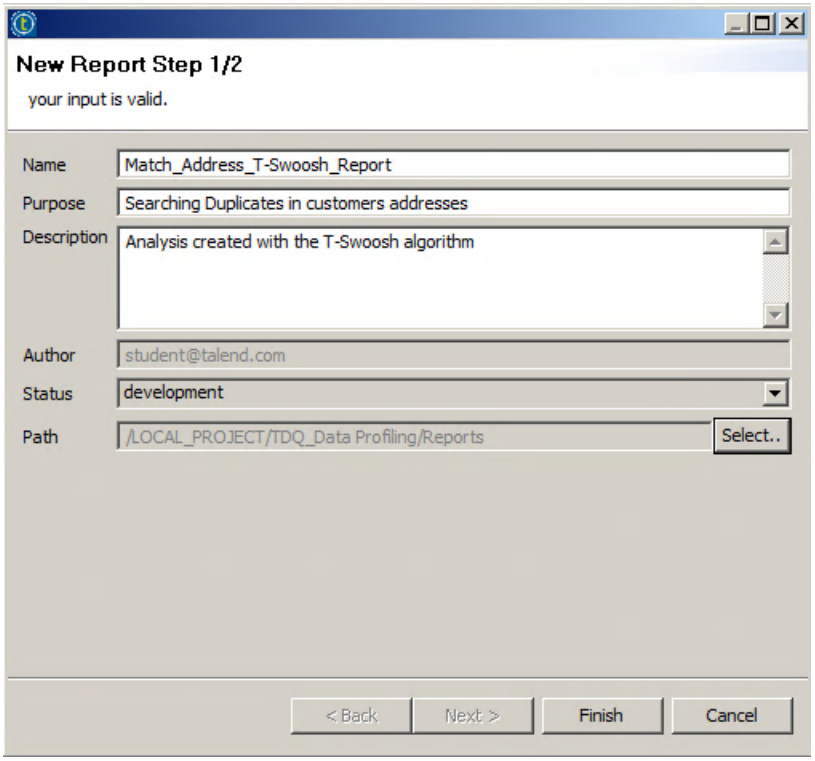
Before creating a report, switch to the Profiling perspective.

1. CREATE A REPORT

   a. In **DQ Repository**, in **Data Profiling**, right-click the **Reports** folder and select **New Report**.

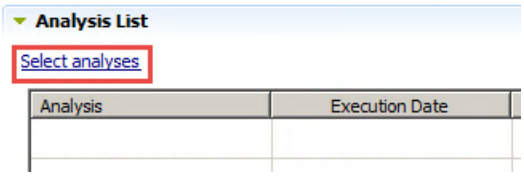


   b. Fill in the **Name**, **Purpose,** and **Description** text boxes and click **Finish**.



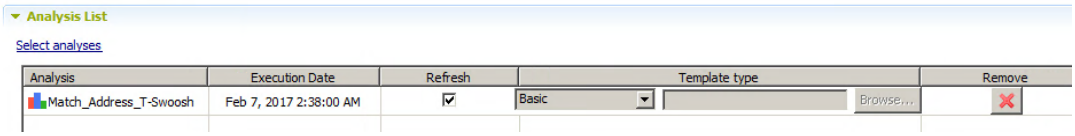
   The Report Settings window opens.

   c. In the **Analysis List** section, click **Select analyses**.



   Select the *Match_Address_T-Swoosh* analysis.

The Refresh box is selected by default—to guarantee the latest data, the analysis automatically runs before the report is executed.

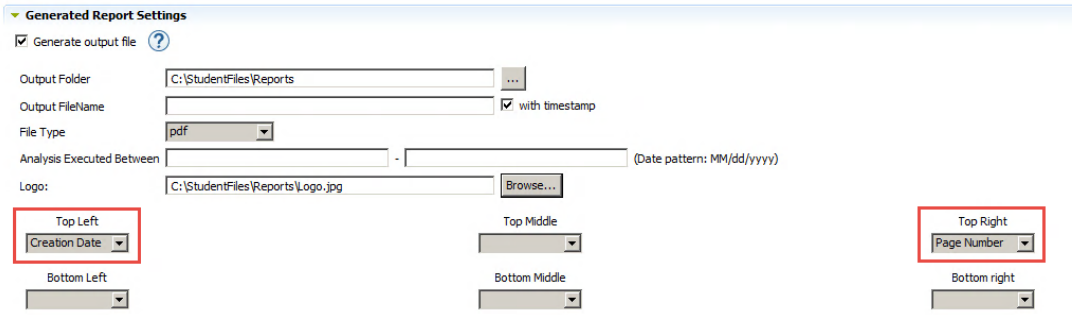


Note: You can select several analyses, which are displayed in succession in the same report.

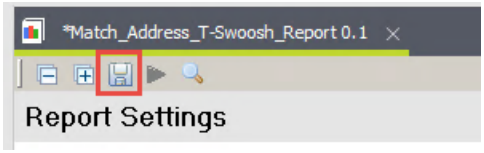
d. Locate the **Generated Report Settings** section. The Output Folder and Logo boxes already contain the values you defined when setting up the preferences.

Do not enter anything in the Output FileName box. This way, the generated file is saved with the report name and a timestamp.

In the **Top Left** box, select *Creation Date*, and in the **Top Right** box, select *Page Number*.



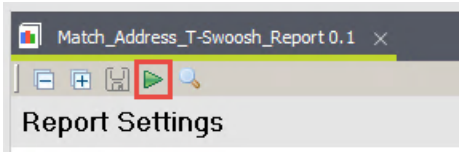
e. You must save the report before running it. Click the **Save** icon.



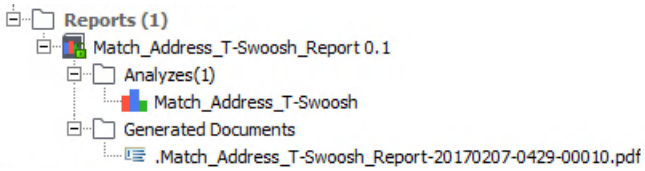
2. EXECUTE THE REPORT

The Run icon is active.

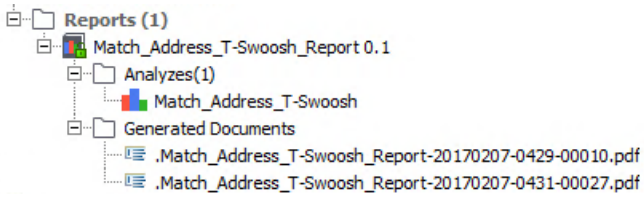
a. Click the **Run** icon.



b. The report appears in the Reports folder. To display the PDF file, expand **Match_Address_T-Swoosh_Report**, then the **Generated Documents** folder.



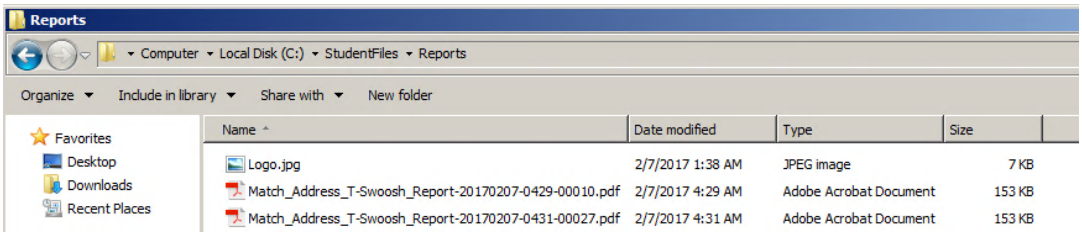
c. Run the report again. A second file is created and appears below the first.



Notice the timestamp in the file names.

d. Open a PDF file and examine the format and results. Scroll down to see all the statistics and bar charts.

e. To confirm the creation of the two PDF files, in **Windows Explorer**, navigate to **C:\StudentFiles\Reports**.



These reports show the analysis results at specific times.

To show data fluctuation for a period of time, you must create an evolution report.

Creating an evolution report

Overview

In this lesson, you will create an evolution report that tracks changes in data quality over time. With each execution, an evolution report saves the state of each indicator. Then the evolution of the indicators is charted on a graph for easy comparison of current and historical results.

You will create an evolution report for a business rule analysis you created earlier. You configured the Claim Dates Analysis to control the validity of the claim creation date. To be valid, this date must fall between the contract start and end dates.

To fake a data evolution, you will run a Job that corrects a few dates between two runs of the report.

Before creating the report, run and display the Claim Dates Analysis. The analysis shows two indicators:

- **Row Count** is the number of rows (100) in the claim table
- **Claim_Dates** is the number of valid dates (96 rows)

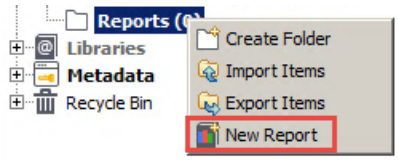
Creating the evolution report

The first step is to create the report.

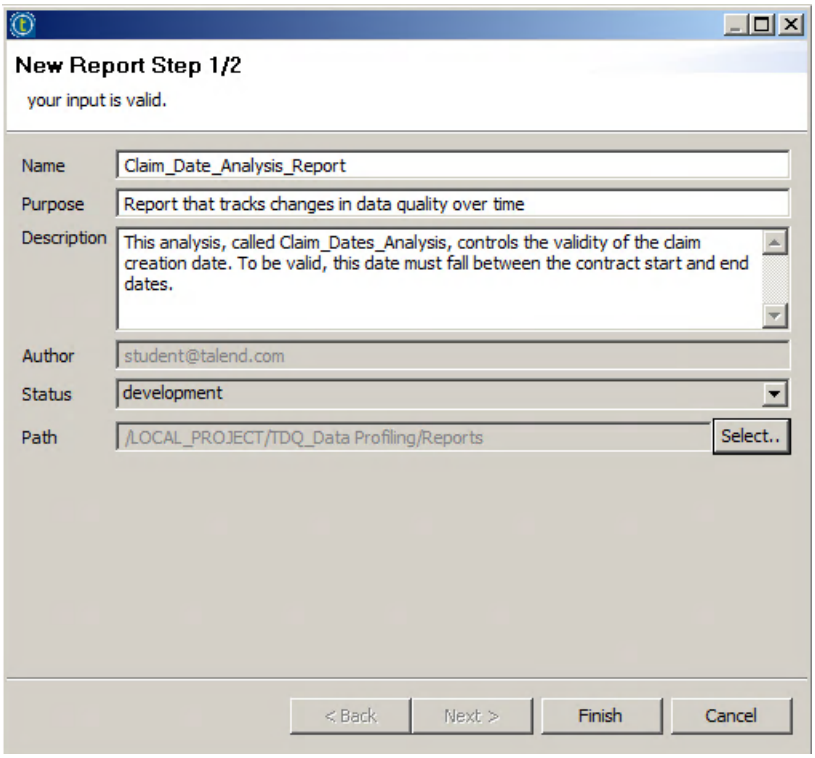
1. CREATE THE REPORT

   Use the same process you used to create a report for the Claim Dates Analysis.

   a. In **DQ Repository**, in **Data Profiling**, right-click the **Reports** folder and select **New Report.**

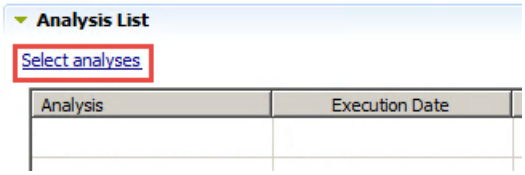


b. Fill in the **Name**, **Purpose**, and **Description** text boxes and click **Finish**.

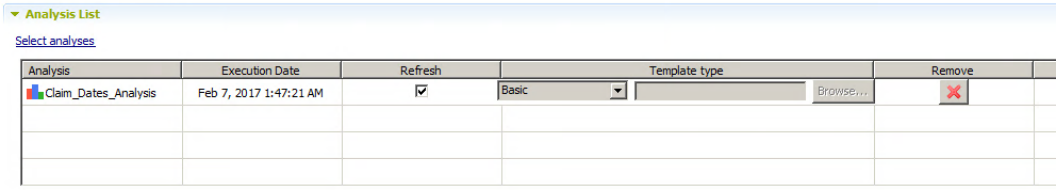


The Report Settings window opens.

c. In the **Analysis List** section, click **Select analyses**.



Select **Claim_Dates_Analysis**.

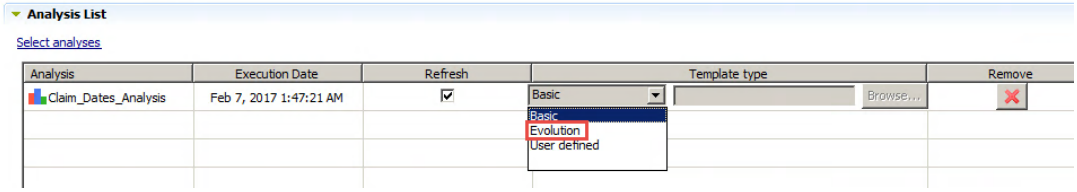


d. In the **Generated Report Settings** section, in the **Top Left** box, select **Creation Date**. In the **Top Right** box, select **Page Number**.

2. SET UP AND RUN THE EVOLUTION REPORT

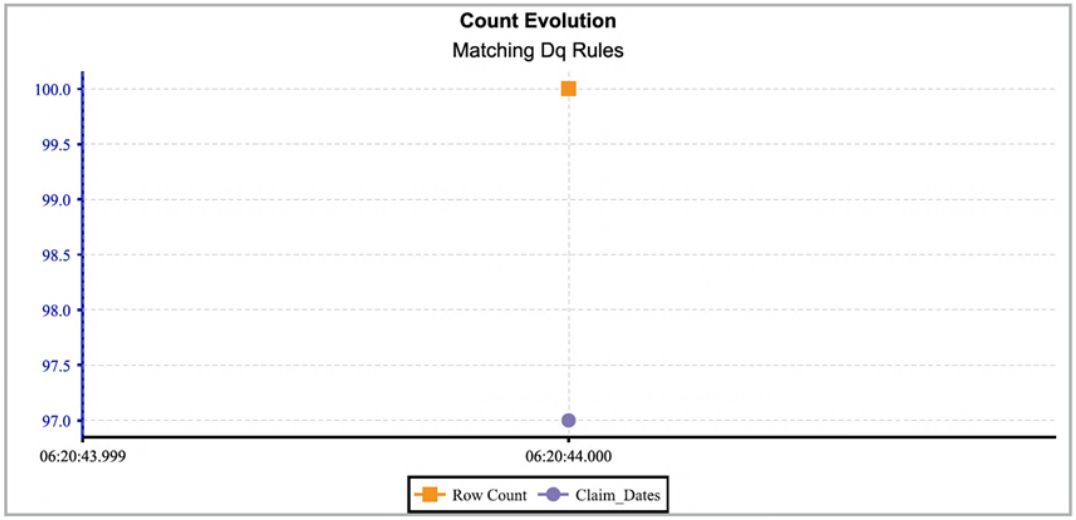
In order to track changes, you must set up the report as an evolution report.

a. Change **Template type** from **Basic** to **Evolution**.



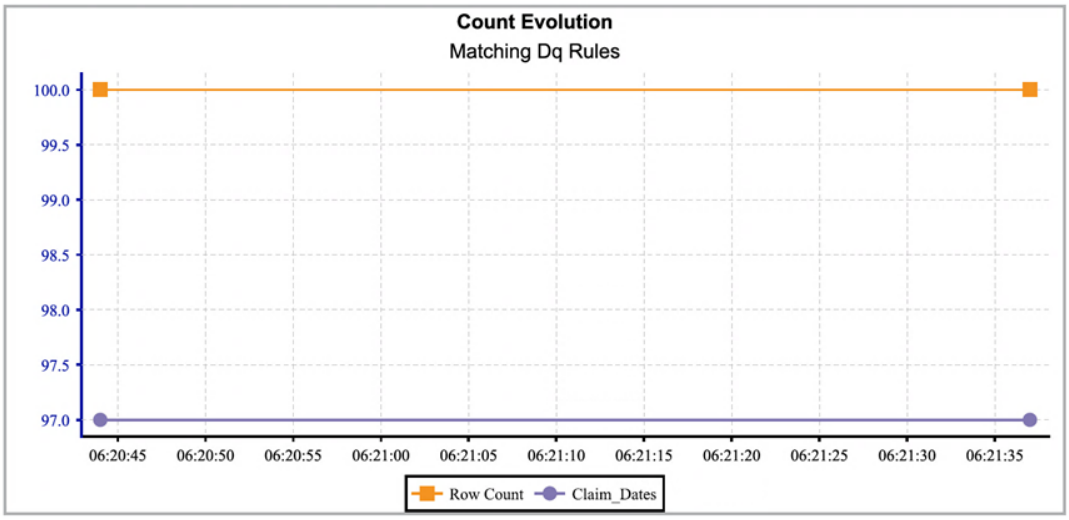
Until this point, you have used the default Basic template.

b. Save and run the report, then display the PDF output.



The graph contains only one point per indicator related to the current execution of the Job.

c. Execute the report a second time and display the new output.



Now there is a second data point representing the second time you generated the report. The line is flat because the data is unchanged; the analysis results are the same. If reports are run on changing data over time, they track the evolution and can help you identify anomalies. For example, you might see a rise or drop on the graph. Before running another evolution report, you can run a simple Job that updates the analyzed data, making the report slightly more

insightful.

Running the Job and tracking evolution

In order to make the next run of the evolution report more insightful, you first need to run the correct_dates Job. It corrects some bad dates in the claim table.

You must upload the Job from an archive stored in the C:\StudentFiles folder.

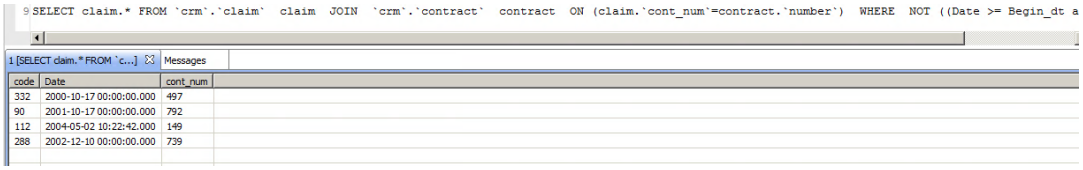
1. TRACK THE BAD DATES IN THE CLAIM_DATES_ANALYSIS

   With the analysis complete, you can monitor the impact of the Job.

   a. Run the **Claim_Dates_Analysis**.

   b. Display the **Analysis Results** tab.

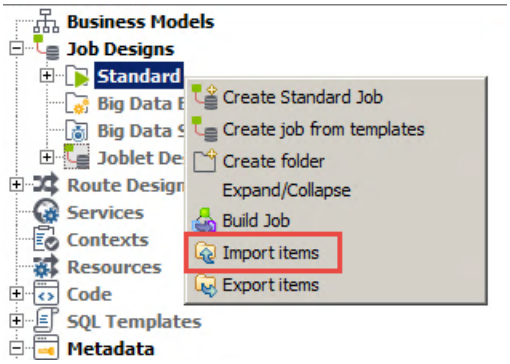
   c. Right-click the results table and click **View invalid rows**. SQL Editor displays the invalid rows.



2. IMPORT THE JOB

   Before uploading the Job, switch to the Integration perspective.

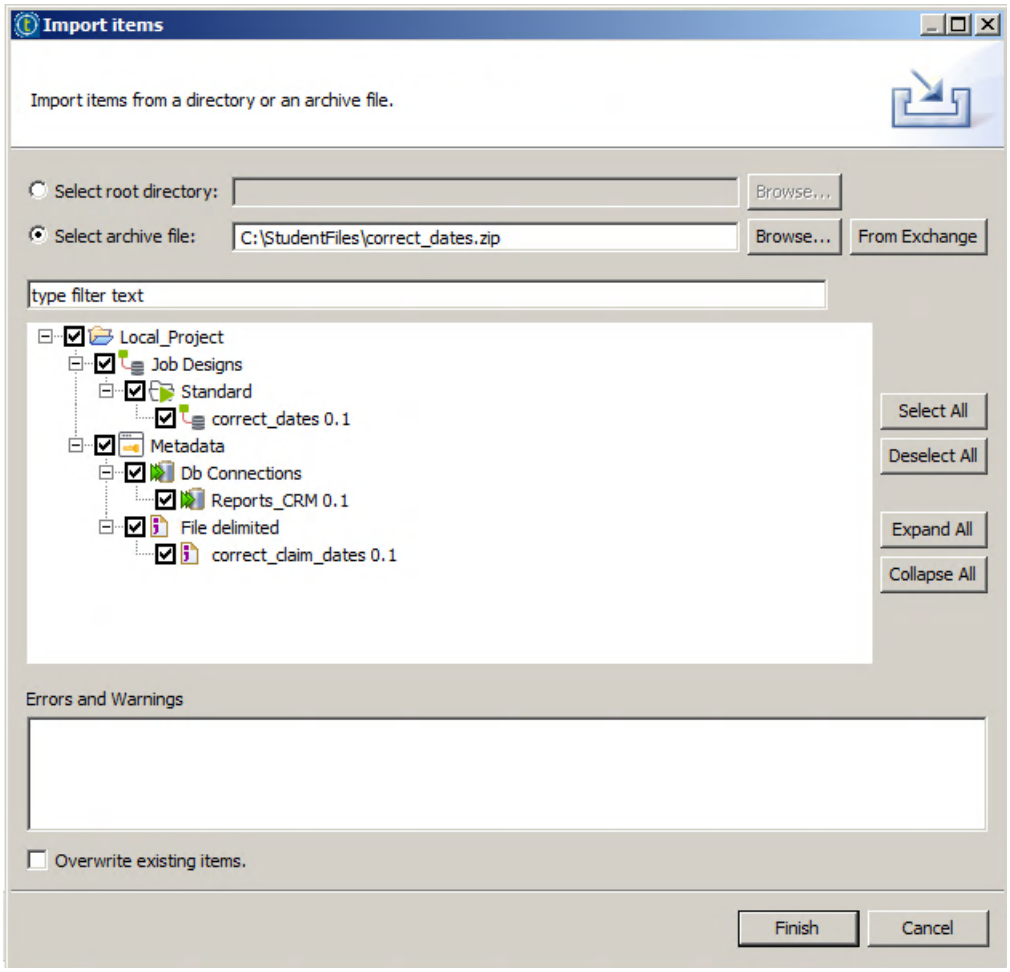
   a. In the **Repository**, expand **Job Designs**, right-click **Standard**, and on the contextual menu, select **Import items**.



   b. Select **Select archive file** and click the **Browse** button.

   c. Navigate to the **C:\StudentFiles** folder and select the **correct_dates.zip** file.

d. Click the **Select All** button. The Job is imported with all of its dependencies.



e. Click **Finish**.

3. RUN THE JOB AND MONITOR THE CHANGES

The Job is available in the repository.

a. In the **Repository**, find and double-click the **correct_dates** Job.

b. To run the Job, press the **F6** key.

c. Switch to the **Claim_Dates_Analysis,** run it, and display the **Analysis Results** tab.

d. Only two dates are invalid.

Label	%Match	%No Match	#Match	#No Match
Claim_Dates	98.00%	2.00%	98.0	2.0

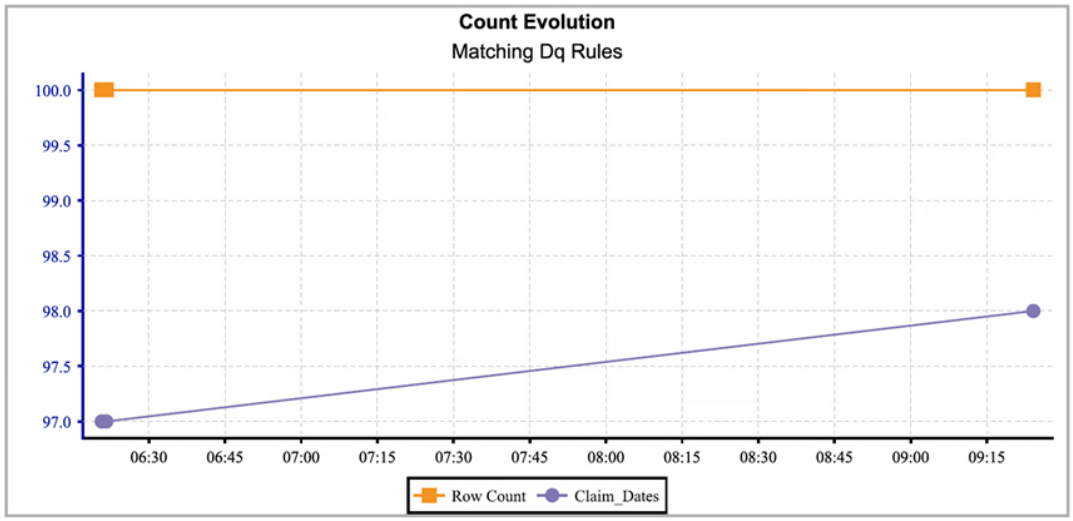
To display them in SQL Editor, right-click the results table and click **View invalid rows**.

4. RUN THE EVOLUTION REPORT

The evolution report tracks the data changes.

a. Switch to the Claim_Dates_Analysis_Report and click the **Run** button.

b. Display the new PDF output.



The graph shows the data quality evolution.

You created two reports. You will display them in the DQP, but first you must access the Web portal.

Configuring the Data Quality portal

Overview

In this lesson, you will launch and access a local version of DQP to view various types of reports.

Accessing DQP

DQP is installed on your virtual machine. To run it locally, you just need to launch the Web server.

Talend Services can be configured for manual or automatic startup, so you may need to manually start the Tomcat Web server.

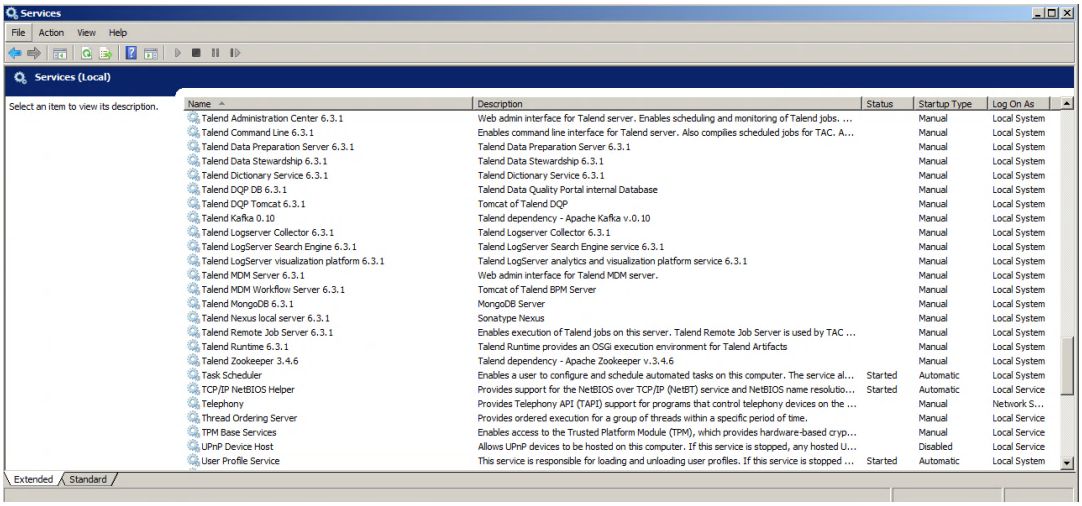
1. CHECK THE DQP STATUS

   You can start the DQP service on the Windows Services console.

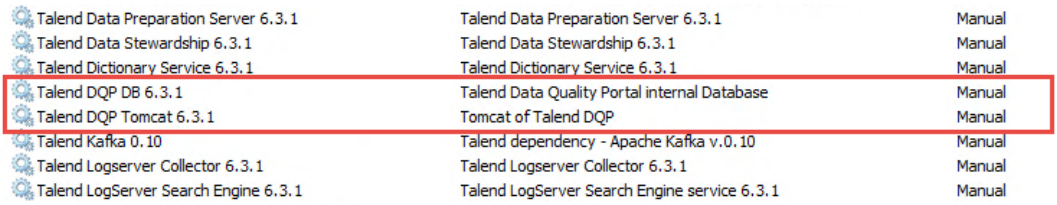
   a. On the Windows task bar, click the **Services** icon.

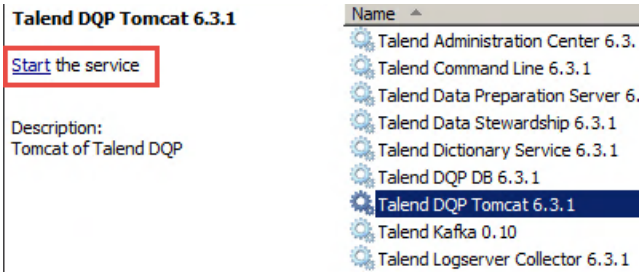
   b. On the **Windows Services** console, scroll down to display the Talend services.

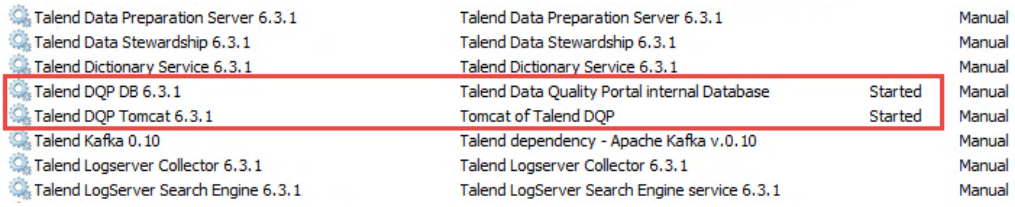
   c. View the status of the **Talend DQP Tomcat** and **Talend DQP DB** services. Neither has been started.

d. Select **Talend DQP Tomcat** and click **Start**.



This runs both services at once.

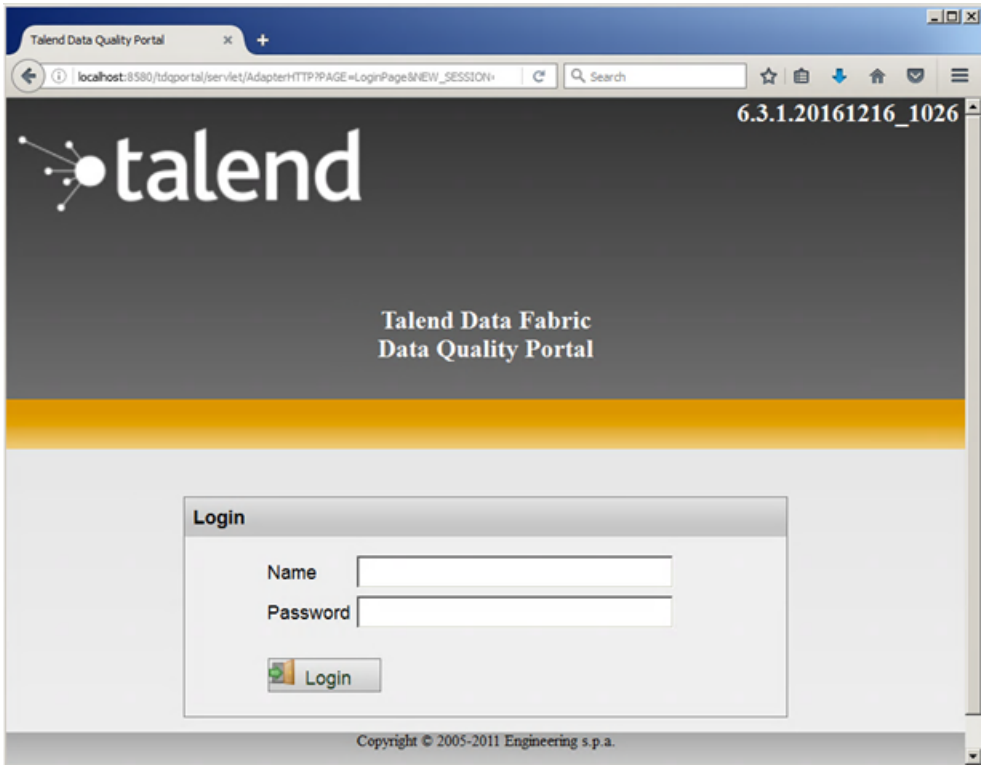


Note: Based on the installation and hardware specifications, starting the Tomcat Web server may take about 60 seconds.

2. LOG IN

Now that the Web server is running locally, you can access DQP from a Web browser.

a. Open a browser, and in the address bar, enter *http://localhost:8580/tdqportal/*

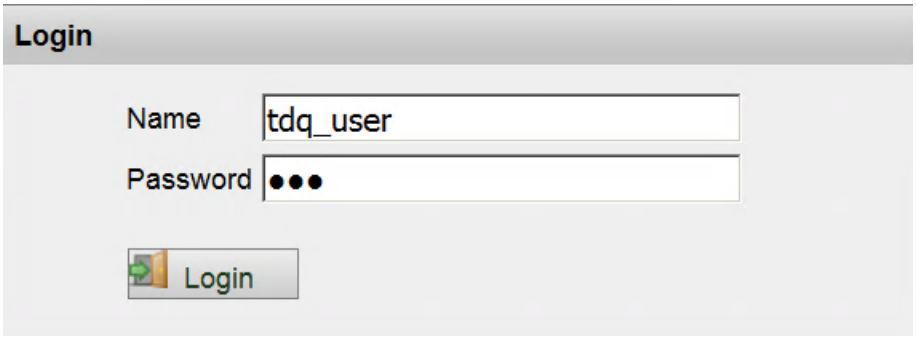


The DQP log-in page opens.

If this page does not appear, Tomcat is probably not running yet. Wait a little longer, or use the correct start-up script to make sure it is started.

b. Log in with **Name** *tdq_user* and **Password** *tdq*.

The DQP main page opens.



This is essentially a blank screen with the menu options on the left pane. Hover over any icon to view its function.

Now you are ready to run reports from DQP.

Running reports on Data Quality portal

Overview

In this lesson, you will load and display the reports you just created on DQP.

Note: On the portal, Talend Data Quality users can also create reports based on SpagoBI, an open-source business intelligence suite. To learn more about SpagoBI, view the resources in the wrap-up.

Displaying basic reports

You can display the reports you created in Studio.

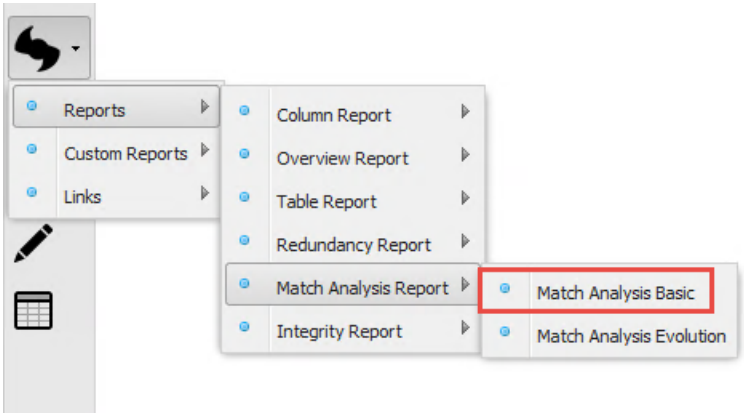
1. DISPLAY THE MATCH REPORT

   Use the toolbar to the left.

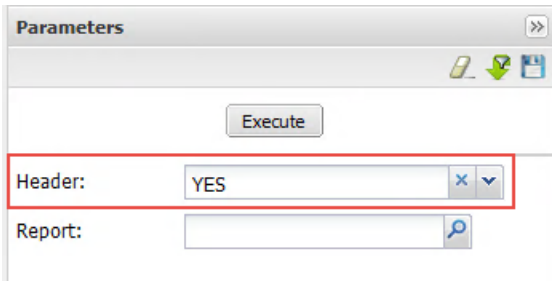
   a. Click the **User Menu** button.

   b. Select **Reports>Match Analysis Report>Match Analysis Basic**.

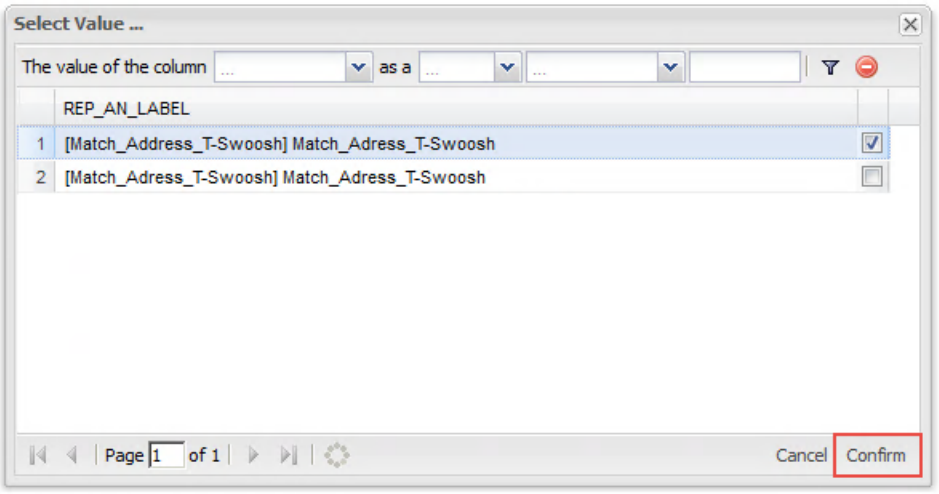
   

   c. Next to the **Header:** box, click the down arrow and select **YES**.



In this case, the header is the Talend logo.

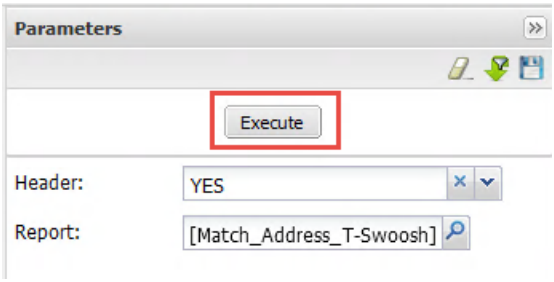
d. Next to the **Report** box, click the **magnifying glass** icon. You get a list of available reports based on the report type you selected.



The name of the report is in square brackets, for example, [Match_Address_T-Swoosh], followed by the name of the analysis selected in the report (in this example, Match_Address_T-Swoosh). The reports on your VM may vary.

Select the first report on the list and click **Confirm**.

e. In the upper right corner, click the **Execute** button.



f. When the report appears in the browser, you can scroll down to view the other sections.

Displaying evolution reports

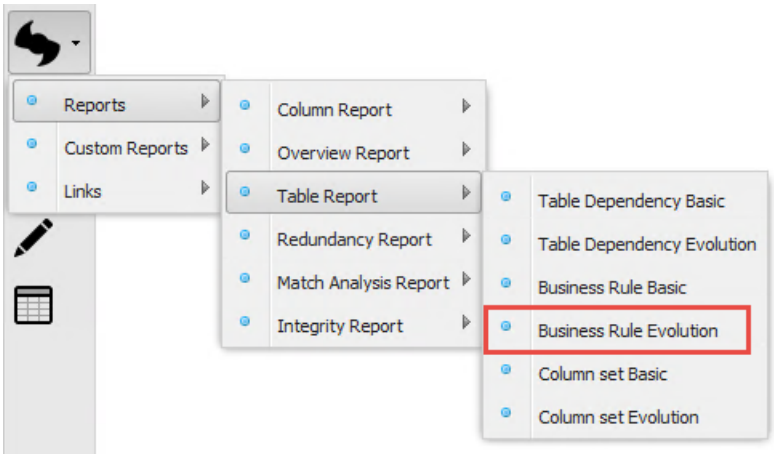
You can also display evolution reports created in Studio.

1. DISPLAY THE MATCH REPORT

   Use the toolbar to the left.

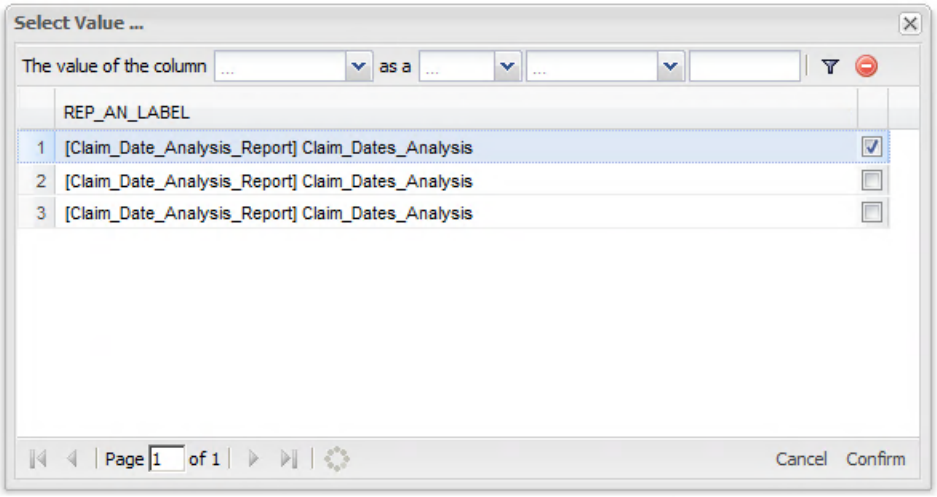
a. Click the **User Menu** button.

b. Select **Reports>Table Report>Business Rule Evolution**.



c. Change the **Header** option to **YES**.

d. In the report **Select Value** window, select the first **Claim_Dates_Analysis**.



Remember, in the previous exercise, you ran this evolution report in Studio.

e. Execute the report and examine the results.

The results should be the same as those in the Claim Dates Analysis report that you ran in Studio.

Displaying integrity reports

Integrity reports are advanced analyses available in DQP. They compile data from the datamart and display statistics on database quality. That means they reuse data created by reports configured and run in Studio.

The Potential PK report analyzes simple statistics indicators: row count, distinct count, unique count, and duplicate count.

To generate the necessary data in the datamart, you will reuse the analysis created for the column analysis challenge. You created this analysis to identify the potential primary key of the Country table, and this is exactly what the Potential PK report does.

1. CREATE AND RUN THE REPORT IN STUDIO

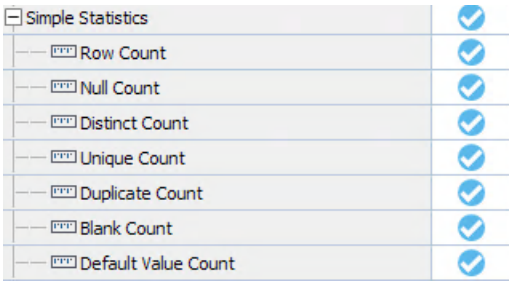
   Switch to Studio and locate the analysis you created for the column analysis challenge.

a. Run the analysis and examine the results.

This basic column analysis is supposed to display simple statistics indicators for all columns in the country table. If it does not, update the analysis settings.

» Confirm that all columns are selected.

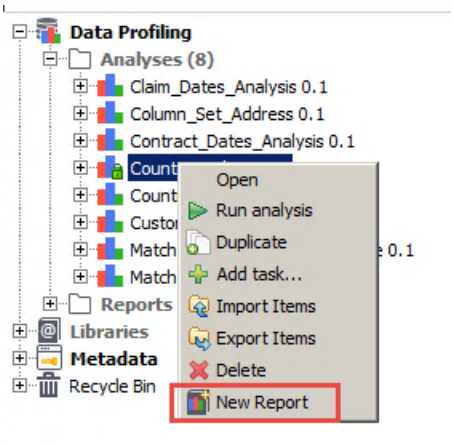
» Confirm that the **Simple Statistics** indicators are selected for all columns.



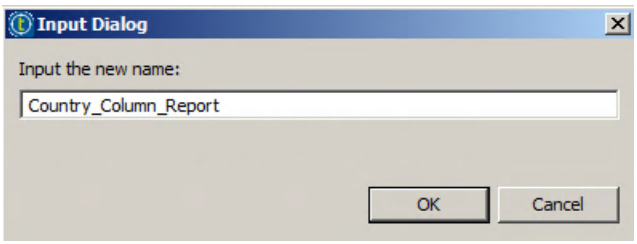
Run the analysis again.

b. Create a simple report for the analysis and run it.

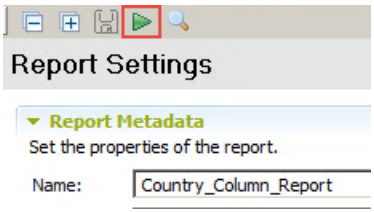
To quickly create a report from DQ Repository, right-click the analysis, and on the contextual menu, select **New Report**.



Name the report.



Click the **Run** icon.

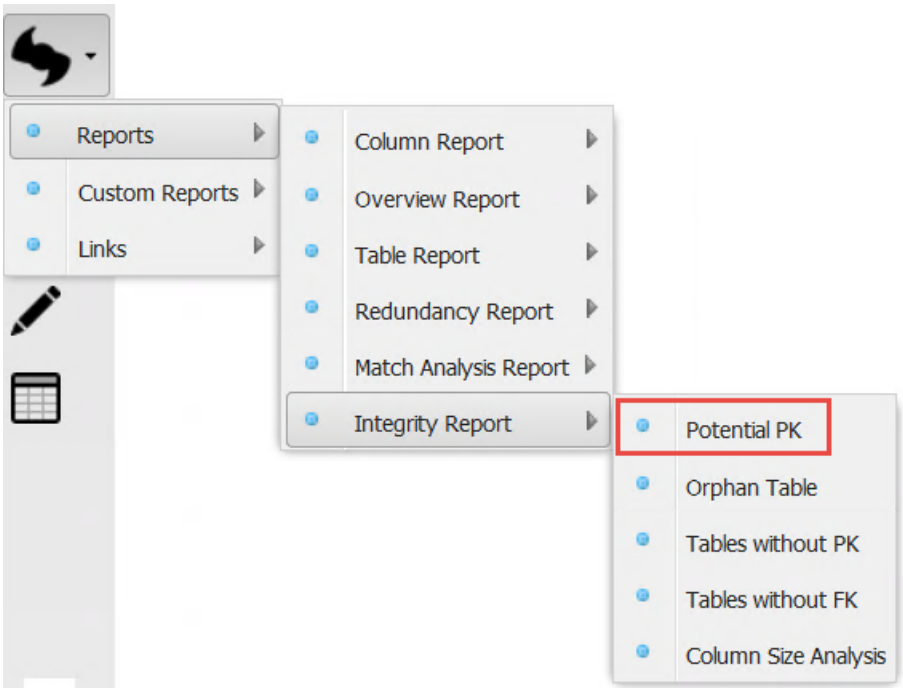


2. DISPLAY THE INTEGRITY REPORT

   Switch back to the browser where DQP is displayed.

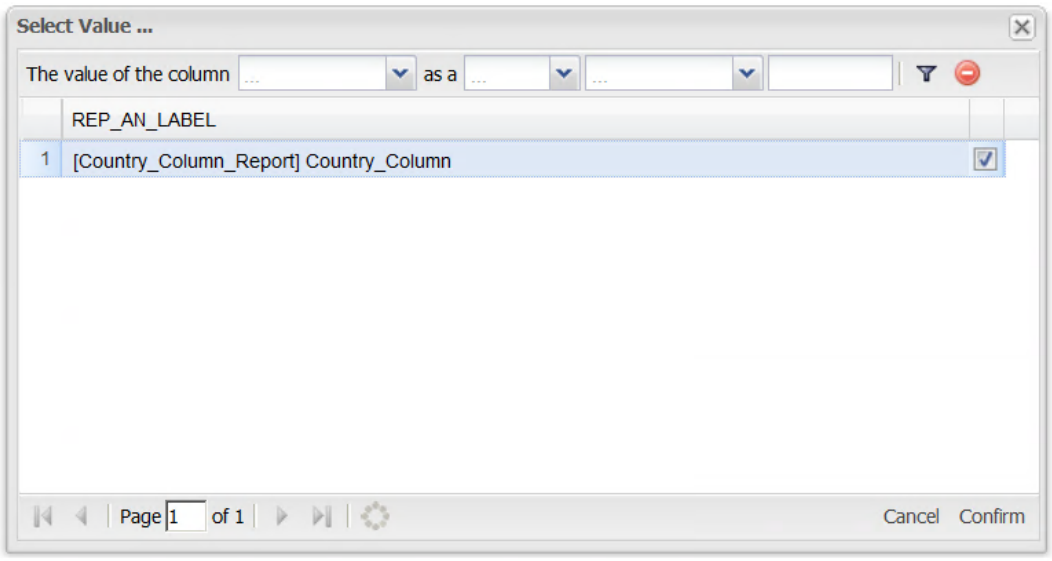
   a. Click the **User Menu** button.

   b. Select **Reports>Integrity Report>Potential PK**.

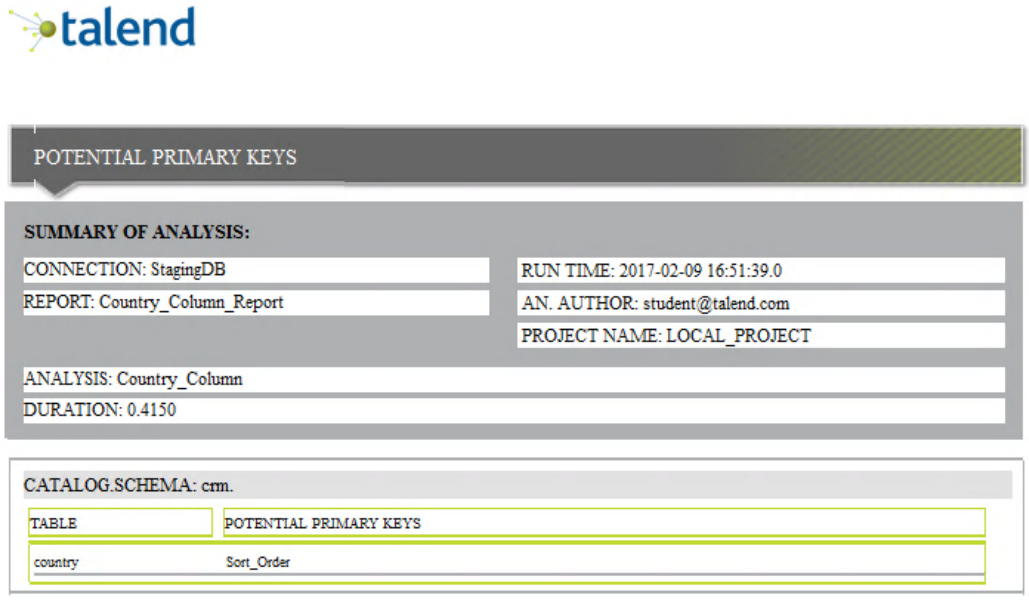


   c. Change the **Header** to **YES**.

   d. In the report selection window, select the report you created.



e. Execute the report and examine the results.



The Sort_Order column is identified as the primary key in the country table. This is exactly what you determined during the column analysis challenge.

You have finished this lesson, so proceed to the wrap-up.

Wrap-Up

In this lesson, you set up the reporting datamart. Then you used analyses to create basic reports as well as evolution reports in Studio.

You launched the DQP, logged in, and ran several reports. The DQP is a good way for business users to view important report data.

Additional details on SpagoBI are beyond the scope of this course. However, clicking the Help icon (?) in the lower left corner of the DQP main menu opens a tab to its wiki page.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

APPENDIX

Additional Information

NOTE:
If the links below yield no search results, please make sure the language filter is set to English.

Talend Documentation:

- Talend Help Center

Lesson 01 - Structural analysis

- Talend Data Fabric - Getting Started Guide
  - Getting Started Guide
- Talend Data Fabric - Data Profiling
  - Data Profiling Getting Started
  - Structural analyses

Lesson 02 - Column analysis

- Talend Data Fabric - Data Profiling
  - Column analyses
  - Patterns and indicators

Lesson 03 - Table analysis

- Talend Data Fabric - Data Profiling
  - Table analyses

Lesson 04 - Cross-table analysis

- Talend Data Fabric - Data Profiling
  - Redundancy analyses

Lesson 05 - Advanced Matching

- Talend Data Fabric - Data Profiling
  - Analyzing duplicates
- Component Reference Guide
  - tMatchGroup

Lesson 06 - Data Privacy

- Component Reference Guide
  - tDataShuffling
  - tDataMasking

Lesson 07 - Reports and Data Quality portal

- Talend Data Fabric - Data Profiling
  - Reports

- Talend Data Quality Portal
  - About Talend Data Quality Portal