



Talend Open Studio for Data Quality

User Guide



Talend Open Studio : User Guide

Adapted for Talend Open Studio for Data Quality v5.1.x.

Copyleft

This documentation is provided under the terms of the Creative Commons Public License (CCPL).

For more information about what you can and cannot do with this documentation in accordance with the CCPL, please read: http://creativecommons.org/licenses/by-nc-sa/2.0/

Notices

All brands, product names, company names, trademarks and service marks are the properties of their respective owners.

Table of Contents

1. General information ix
1.1. Purpose ix
1.2. Audience ix
1.3. Typographical
conventions ix
2. History of changes ix
3. Feedback and Support xi
Chapter 1. Overview of <i>Talend</i>
Open Studio for Data Quality 1
1.1. Data profiling main concepts
1.1.1. The problems data
profiling addresses 2
1.1.2. How does data
profiling promote better data
quality 2
1.2. About <i>Talend Open Studio for</i>
Data Quality 2
1.2.1. What is <i>Talend Open</i>
Studio for Data Quality 2
1.2.2. A simple-to-use data profiler
1.2.3. Core features of <i>Talend</i>
Open Studio for Data Quality
Chapter 2. Getting started with
Talend Open Studio for Data
Quality
\sim ·
2.1. Working principles of the data quality Studio 6
2.2. Launching <i>Talend Open</i>
Studio for Data Quality
2.3. Defining the maximum
memory size threshold
2.4. Setting preferences of
analysis editors and analysis results
analysis editors and analysis
analysis editors and analysis results
analysis editors and analysis results
analysis editors and analysis results82.5. Displaying/hiding the help content92.5.1. Cheat sheets102.5.2. Help panel11
analysis editors and analysisresults82.5. Displaying/hiding the helpcontent92.5.1. Cheat sheets102.5.2. Help panel112.6. Displaying the error log view
analysis editors and analysisresults82.5. Displaying/hiding the helpcontent92.5.1. Cheat sheets102.5.2. Help panel112.6. Displaying the error log viewand managing the log files12
analysis editors and analysis results82.5. Displaying/hiding the help content92.5.1. Cheat sheets102.5.2. Help panel112.6. Displaying the error log view and managing the log files122.7. Icons appended on analyses
analysis editors and analysis results82.5. Displaying/hiding the help content92.5.1. Cheat sheets102.5.2. Help panel112.6. Displaying the error log view and managing the log files122.7. Icons appended on analyses names in the DQ Repository15
analysis editors and analysis results82.5. Displaying/hiding the help content92.5.1. Cheat sheets102.5.2. Help panel112.6. Displaying the error log view and managing the log files122.7. Icons appended on analyses names in the DQ Repository152.8. Opening new editors16
analysis editors and analysis results
analysis editors and analysis results82.5. Displaying/hiding the help content92.5.1. Cheat sheets102.5.2. Help panel112.6. Displaying the error log view and managing the log files122.7. Icons appended on analyses
analysis editors and analysis results

	3.1.3. Connecting to an	
	MDM server	26
3.2. I	Managing the connections to	
	sources	30
	3.2.1. Managing database	
	connections	30
	3.2.2. Managing MDM	
	connections	40
	3.2.3. Managing file	
	connections	46
	Catalogs and schemas in	
data	base systems	46
Chapter	4. Profiling database	
content .		47
	Managing database content	
	yses	48
	4.1.1. Creating a database	
	content analysis	48
	4.1.2. Creating a database	
	content analysis in shortcut	
	procedure	52
	4.1.3. Creating a catalog	
	analysis	53
	4.1.4. Creating a schema	
	analysis	57
4.2.]	Displaying a table key and	
index	x in the analyzed database	60
13 -	Fracking data changes in	
4.3.	ruching duta changes m	
	ce databases	62
		62
	ce databases	62
	ce databases 4.3.1. Comparing tree-view	
	4.3.1. Comparing tree-view metadata structures with	
	4.3.1. Comparing tree-view metadata structures with database structures	
	4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the	62
sour	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the	62 67
sour	 ce databases	62 67 71
sour Chapter 5.1. S	 4.3.1. Comparing tree-view metadata structures with database structures	62 67 71
sour Chapter 5.1. S	 ce databases	62 67 71 72
sour Chapter 5.1. S	cc databases4.3.1. Comparing tree-viewmetadata structures withdatabase structures4.3.2. Synchronizing theconnection structure with thedatabase structure5. Column analysesSteps to analyze a columnData mining types	62 67 71 72 72
sour Chapter 5.1. S	cc databases4.3.1. Comparing tree-viewmetadata structures withdatabase structures4.3.2. Synchronizing theconnection structure with thedatabase structure5. Column analysesSteps to analyze a columnData mining types5.2.1. Nominal	62 67 71 72 72 73
sour Chapter 5.1. S	cc databases4.3.1. Comparing tree-viewmetadata structures withdatabase structures4.3.2. Synchronizing theconnection structure with thedatabase structure 5. Column analyses Steps to analyze a column 5.2.1. Nominal5.2.2. Interval	62 67 71 72 73 73
sour Chapter 5.1. § 5.2.]	cc databases4.3.1. Comparing tree-viewmetadata structures withdatabase structures4.3.2. Synchronizing theconnection structure with thedatabase structure 5. Column analyses Steps to analyze a columnData mining types5.2.1. Nominal5.2.3. Unstructured text	62 67 71 72 73 73 73 74
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column	62 67 71 72 73 73 73 74
sour Chapter 5.1. § 5.2. 1	cc databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other Analyzing columns in a	62 67 71 72 73 73 73 74 74
sour Chapter 5.1. § 5.2. 1	cc databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other 5.3.1. Defining the columns to be analyzed and setting	62 67 71 72 73 73 73 74 74
sour Chapter 5.1. § 5.2. 1	cc databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other 5.3.1. Defining the columns	62 67 71 72 73 73 73 74 74
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column	62 67 71 72 73 73 73 74 74 74 74
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column. Data mining types 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column	62 67 71 72 73 73 73 74 74 74 74
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other Stase 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column analysis before execution 5.3.3. Using the Java or the	62 67 71 72 73 73 74 74 74 74 74
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other Analyzing columns in a base 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column analysis before execution 5.3.3. Using the Java or the SQL engine	62 67 71 72 73 73 74 74 74 74 74
sour Chapter 5.1. § 5.2. 1	cc databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column Data mining types 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column analysis before execution 5.3.3. Using the Java or the SQL engine 5.3.4. Accessing the detailed	62 67 71 72 73 73 74 74 74 74 75 83
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column Data mining types 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other Analyzing columns in a base 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column analysis before execution 5.3.3. Using the Java or the SQL engine 5.3.4. Accessing the detailed view of the database column	62 67 71 72 73 73 74 74 74 74 75 83 85
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column Data mining types 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other Stase 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column analysis before execution 5.3.3. Using the Java or the SQL engine 5.3.4. Accessing the detailed view of the database column analysis	62 67 71 72 73 73 74 74 74 74 75 83
sour Chapter 5.1. § 5.2. 1	ce databases 4.3.1. Comparing tree-view metadata structures with database structures 4.3.2. Synchronizing the connection structure with the database structure 5. Column analyses Steps to analyze a column Data mining types 5.2.1. Nominal 5.2.2. Interval 5.2.3. Unstructured text 5.2.4. Other Analyzing columns in a base 5.3.1. Defining the columns to be analyzed and setting indicators 5.3.2. Finalizing the column analysis before execution 5.3.3. Using the Java or the SQL engine 5.3.4. Accessing the detailed view of the database column	62 67 71 72 73 73 74 74 74 74 75 83 85

	5.3.6. Using regular
	expressions and SQL patterns
	in a column analysis 90
	5.3.7. Saving the queries
	executed on indicators
	5.3.8. Creating table and
	columns analyses in shortcut
	procedures
5.4. <i>A</i>	Analyzing master data on an
	A server 100
	5.4.1. Defining the business
	entities to be analyzed and
	setting indicators 100
	5.4.2. Accessing the detailed
	view of the master data
	analysis 109
	5.4.3. Analyzing master data
	in shortcut procedures 110
5.5. A	Analyzing data in a file 111
	5.5.1. Analyzing columns in a
	delimited file 111
	5.5.2. Analyzing columns in
	an excel file 121
Chapter	6. Table analyses 125
-	Steps to analyze a table 126
	Analyzing tables in databases
	6.2.1. Creating a simple table
	analysis: the analysis of a set
	of columns 126
	6.2.2. Creating a table
	analysis with SQL business
	rules 140
	6.2.3. Detecting anomalies in
	the table columns: column
	functional dependency
	analysis 153
	6.2.4. Creating a column
	analysis from a simple table
	analysis 159
6.3. A	Analyzing tables in delimited
	160
	6.3.1. Creating a column set
	analysis on a delimited file
	using patterns 160
	6.3.2. Creating a column
	analysis from the analysis of
	a set of columns 169
6.4. <i>A</i>	Analyzing tables on MDM
serve	ers 170
	6.4.1. Creating a column set
	analysis on an MDM server 170
	6.4.2. Creating a column
	analysis from the column set
	analysis 176
Chapter	7. Redundancy analysis
•••••	179

	7.1. What are redundancy	
	analyses	180
	7.2. Comparing identical columns	
	n different tables	180
	7.3. Matching primary and	
f	oreign keys	185
Chap	ter 8. Correlation analyses	
-		191
8	8.1. What are column correlation	
	analyses	192
	8.2. Numerical correlation	
	analysis	192
	8.2.1. Creating numerical	
	correlation analysis	193
	8.2.2. Accessing the detailed	
	view of the analysis results	199
8	8.3. Time correlation analysis	
	8.3.1. Creating time	
	correlation analysis	201
	8.3.2. Accessing the detailed	
	view of the analysis results	206
8	8.4. Nominal correlation analysis	
		208
	8.4.1. Creating nominal	
	correlation analysis	208
	8.4.2. Accessing the detailed	
	view of the analysis results	212
Chan	oter 9. Extended	
_	ionality: patterns and	
Tunci	ionanty. patterns and	
• ••		A 1 F
	ators	
	9.1. Patterns	216
	9.1.1. Patterns	216
	9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User-	216
	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in 	216 216
	9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases	216 216
	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular 	216 216 216
	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns 	216 216 216
	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 	216 216 216
	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular 	216 216 216
	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 	216216216222
5	 9.1. Patterns	 216 216 216 222 222
5	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Managing regular expressions and SQL patterns 	 216 216 216 222 222 245
5	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Managing regular 9.1.4. Managing reg	 216 216 216 222 222 245
5	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Indicators 9.2.1. Indicator types 9.2.2. Managing system 	 216 216 216 222 222 245
5	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Indicators 9.2.1. Indicator types 9.2.2. Managing system indicators 	 216 216 216 222 222 245
5	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Indicator types 9.2.1. Indicator types 9.2.2. Managing system indicators 9.2.3. Managing user-defined 	 216 216 216 222 222 245 248
5	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.2.1. Indicator types 9.2.2. Managing system indicators 9.2.3. Managing user-defined indicators 	 216 216 216 222 245 248 2500
S S	 9.1. Patterns	 216 216 216 222 245 248 2500
S S	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.2.1. Indicator types 9.2.2. Managing system indicators 9.2.3. Managing user-defined indicators 	 216 216 216 222 245 248 2500
s	 9.1. Patterns	 216 216 216 222 245 248 250 268
s Chap mana	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.2.1. Indicator types 9.2.2. Managing system indicators 9.2.3. Managing user-defined indicators 9.2.4. Indicator parameters oter 10. Other important agement procedures 	 216 216 216 222 245 248 250 268
s S Chap mana	 9.1. Patterns 9.1.1. Pattern types 9.1.2. Managing User- Defined Functions in databases 9.1.3. Adding regular expressions and SQL patterns to column analyses 9.1.4. Managing regular expressions and SQL patterns 9.1.4. Managing regular expressions and SQL patterns 9.2.1. Indicator types 9.2.2. Managing system indicators 9.2.3. Managing user-defined indicators 9.2.4. Indicator parameters 9.2.4. Indicator parameters 9.2.4. Indicator parameters 9.2.4. Indicator s 9.2.5. Managing user-defined 10. Other important 10.1. Creating and storing SQL 	 216 216 216 222 245 248 250 268
s Chap mana	 9.1. Patterns	 216 216 216 222 245 245 248 250 268 271
S Chap mana 1 G	 9.1. Patterns	 216 216 216 222 245 245 248 250 268 271
S Chap mana 1 c f 1	 9.1. Patterns	 216 216 216 222 245 248 250 268 271 272

10.3. Exporting data profiling	
items 270	5
10.4. Upgrading projects items	
from older versions 278	\$
Chapter 11. Managing existing	
analyses 279)
11.1. Procedures for all types of	
analyses	
11.1.2. Executing an analysis 200	
)
11.1.3. Duplicating an	
analysis 280)
11.1.4. Adding a task to an	
analysis	
11.1.5. Deleting or restoring	1
an analysis	
11.2.1. Adding a task to	L
a column in a database	
connection 282	2
11.2.2. Adding a task to an	
item in a specific analysis 283	;
11.2.3. Adding a task to an	
indicator in a column analysis	4
	ł
list 285	5
11.2.5. Deleting a completed	,
task 286	5
Appendix A. Talend Open Studio	
for Data Quality management	
GUI 289)
A.1. Main window of Talend Open	
Studio for Data Quality 290)
A.2. Menu bar of <i>Talend Open</i>	
Studio for Data Quality 291	
A.3. Toolbar of <i>Talend Open</i>	
Studio for Data Quality 292 A.4. Tree view of Talend Open	2
Studio for Data Quality	,
A.5. Detailed View of <i>Talend Open</i>	~
Studio for Data Quality	3
A.6. The Profiling perspective of	
Talend Open Studio for Data	
<i>Quality</i> 294	ł
A.7. Tab panel of the analysis	
editors	ł
A.8. Selecting a task from <i>Talend</i> Open Studio for Data Quality	
management GUI 296	5
Appendix B. Data Explorer	-
management GUI 299)
B.1. Main window of the Data	
Explorer)

B.2. Menu bar of the Data Explorer	
-	300
B.3. Toolbar of the Data Explorer	301
B.4. Connections view	301
B.5. SQL History view	301
B.6. The SQL editor view	302
B.7. Database Structure view	
B.8. Database Detail view	303
Appendix C. Regular	
expressions on SQL Server 3	305
C.1. Main concept	306
C.2. How to create a regular	
expression function on SQL Server	
	306
C.2.1. How to create a project	
in Visual Studio	306
C.2.2. How to deploy the	
regular expression function to	
the SQL server	308
C.2.3. How to set up Talend	
Open Studio for Data Quality	
	312
C.3. How to test the created	
function via the SQL Server editor	
	314

List of Tables

A.1. Table 1—Management menus	291
A.2. Table 2—Management toolbar	292

Preface

1. General information

1.1. Purpose

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

Information presented in this document applies to *Talend Open Studio for Data Quality* releases beginning with **5.1.x**.

1.2. Audience

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



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

1.3. Typographical conventions

This guide uses the following typographical conventions:

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

2. History of changes

The below table lists changes made in the Talend Open Studio for Data Quality User Guide.

Version	Date	History of Changes
v5.0_a	15/12/2011	Updates in <i>Talend Open Studio for Data Quality User Guide</i> include:

Version	Date	History of Changes		
		• Modifications throughout the document to reflect the new column and table filters in the columns and table selection dialog boxes.		
		• Updated documentation to reflect new product names. For further information on these changes, see the Talend website.		
		• New sections about the new "Phone Number Statistics" indicators.		
v5.0_b	13/02/2012	Updates in Talend Open Studio for Data Quality User Guide include:		
		• Added legal notices to the User Guide.		
		• The <procedure> element is used in place of the <orderedlist> element throughout the document.</orderedlist></procedure>		
		• Updates in "regular expressions on SQL server" appendix: reorganization of the section "how to create a regular expression function on an SQL server + new section "how to set up talend data quality studio".		
v5.1_a	03/05/2012	Updates in Talend Open Studio for Data Quality User Guide include:		
		• Modifications in the Table analysis chapter + the section about setting editor preferences in the studio to replace "DQ rule" with "Business rule".		
		• Modifications in the "Indicator parameters" section to define the new indicators: blank options, java options and phone numbers.		
		• A new section about database structure "Catalogs and schemas in database systems" (before you begin chapter).		
		• A new section about setting the maximum memory size threshold for analyses (Getting Started chapter).		
		• A new section about how to create a new pattern from the Pattern Test View (Patterns and Indicators chapter)		
		• A new section about filtering database connections (Before you begin chapter)		
		• Slight modification in the Table analysis chapter to show the difference between column analysis and column set analysis.		
		• Modifications throughout the whole book in the procedural steps and the notes that talk about the Connection list to add a sentence about the modified Connection combo box that lists now the connection and the database name.		
		• Many captures in the column and column set analyses results are replaced to show the new vertical bar chart that takes the place of the horizontal bar chart.		

3. Feedback and Support

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

http://talendforge.org/forum



Chapter 1. Overview of *Talend Open Studio* for *Data Quality*

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

This chapter points out that data profiling, data quality, data cleansing and data integration go together since all these address related issues in data assessment, acquisition and improvement.

1.1. Data profiling main concepts

Beginning a data-driven initiative in your enterprise without first understanding the enterprise data will lead to great losses. Data improvement efforts must start with an understanding of the integrity of the data of the enterprise.

1.1.1. The problems data profiling addresses

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

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

A clear, up-front picture of all the potential issues is essential to plan projects effectively. Data analyzing, data cleansing and data transformation requirements must be understood before timescales and costs are finalized, and not after.

1.1.2. How does data profiling promote better data quality

The first step in improving the quality of data is to "profile" or evaluate that data. Data profiling helps you understand the data you manage and the rules that govern that data. Without this knowledge, no effective data management plan can be formulated.

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

1.2. About *Talend Open Studio for Data Quality*

The following sections introduce Talend Open Studio for Data Quality and list its key features.

1.2.1. What is Talend Open Studio for Data Quality

Talend Open Studio for Data Quality is a stand-alone data profiling tool that provides several perspectives including:

- Profiling the main perspective, used for profiling data,
- Data Explorer another perspective which allows the user to browse data.

Talend Open Studio for Data Quality helps you discover and understand the quality of your data. With Talend Open Studio for Data Quality, you can carry out accurate data profiling processes and thus reduce the time and

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

Its comprehensive data profiling features will help you enhance and accelerate your data analysis projects.

Talend Open Studio for Data Quality helps you communicate with others in your organization what you have discovered about the data.

1.2.2. A simple-to-use data profiler

Talend Open Studio for Data Quality is a sophisticated yet simple-to-use and easy to implement data profiler. It centralizes a:

- data profiler, for more information about the data profiler, see Appendix A, *Talend Open Studio for Data Quality management GUI*.
- data automated explorer, for more information about the data explorer, see Appendix B, *Data Explorer* management GUI.
- pattern manager, for more information about the pattern manager, see Section 1.2.3.2, "Patterns and indicators".
- metadata manager, for more information about the metadata manager, see Section 1.2.3.1, "Metadata repository".

1.2.3. Core features of *Talend Open Studio for Data Quality*

This section describes the basic features of Talend Open Studio for Data Quality.

1.2.3.1. Metadata repository

Talend Open Studio for Data Quality connects to databases and delimited files to analyze their structure (catalogs, schemas and tables), and stores the description of their metadata in its metadata repository. You can then use this metadata to set up metrics and indicators.

For more information, see Section 3.1.1, "Connecting to a database" and Chapter 6, Table analyses.

1.2.3.2. Patterns and indicators

Patterns are sets of strings against which you can define the content, structure and quality of high complex data. *Talend Open Studio for Data Quality* lists two types of patterns: regular expressions, which are predefined regular patterns, and SQL patterns which are the patterns you add using LIKE clauses.

For more information about patterns, see Section 9.1, "Patterns".

Indicators are the results achieved through the implementation of different patterns. They can represent the results of data matching and different other data-related operations. *Talend Open Studio for Data Quality* lists two types

of indicators: system indicators, a list of predefined indicators, and user-defined indicators, a list of those defined by the user.

For more information about indicators, see Section 9.2, "Indicators".



Chapter 2. Getting started with *Talend Open Studio for Data Quality*

This chapter introduces Talend Open Studio for Data Quality and guides you through the basics for launching it.

This chapter explains the typical sequence of profiling data using *Talend Open Studio for Data Quality* and many other important miscellaneous subjects.

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

2.1. Working principles of the data quality Studio

From *Talend Open Studio for Data Quality*, you can examine the data available in different data sources and collect statistics and information about this data.

A typical sequence of profiling data using Talend Open Studio for Data Quality involves the following steps:

- 1. Connecting to a data source including databases, a Master Data Management (MDM) servers and delimited files or excel files in order to be able to access the tables and columns on which you want to define and execute analyses. For more information, see Chapter 3, *Before you begin profiling data*.
- 2. Defining any of the available data quality analyses including database content analysis, column analysis, table analysis, redundancy analysis, correlation analysis, etc. These analyses will carry out data profiling processes that will define the content, structure and quality of highly complex data structures. The analysis results will be displayed graphically next to each of the analysis editors, or in more detail in the **Analysis Results** view.



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

2.2. Launching *Talend Open Studio for Data Quality*

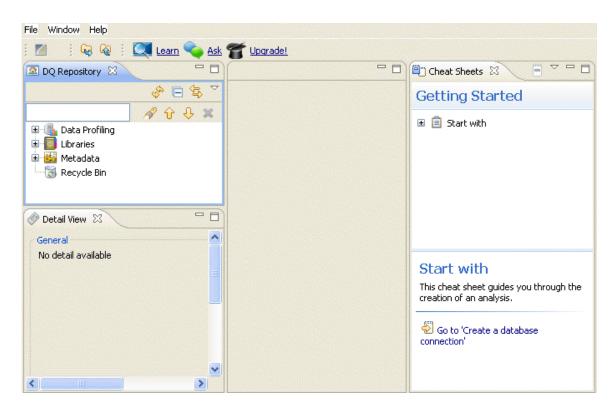
To open Talend Open Studio for Data Quality for the first time, do the following:

1. Unzip the *Talend Open Studio for Data Quality* zip file and, in the folder, double-click the executable file corresponding to your operating system.



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

- 2. In the **[License]** window that is displayed, read and accept the terms of the license agreement to proceed to the next step. A registration window is displayed.
- 3. If required, follow the instructions provided to join **Talend** community or click **Register later** to open a welcome window.
- 4. In the welcome window, click **Start now** to open *Talend Open Studio for Data Quality*.



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

For more information about creating new analyses, see Section 2.1, "Working principles of the data quality Studio".

For more information about importing analyses and data quality items created in other Studios, see Section 10.2, "Importing data profiling items" and Section 10.4, "Upgrading projects items from older versions".

2.3. Defining the maximum memory size threshold

Talend Open Studio for Data Quality enables you to control memory usage when using the Java engine to run two types of analyses: column analysis and the analysis of a set of columns.

Why would you like to set a memory limit when running such analyses? If you use column analysis or column set analysis to profile very big sets of data or data with many problems, you may run out of memory and end up with a Java heap error. By defining the maximum memory size threshold for these analyses, the Studio will stop the analysis execution when the memory limit size is reached and provide you with the analysis results that were measured on the data before the analysis execution was terminated by the memory limit size.

Prerequisite(s): Talend Open Studio for Data Quality is open.

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

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

🗳 Preferences	
type filter text	Analysis tuning $\Leftrightarrow \bullet \bullet \bullet \bullet \bullet$
 General ATL EMF Compare Help Install/Update 	Limit Analyzed columns limit: 200 Memory
 Java Run/Debug Talend Data Explorer Metadata Profiling Analysis tuning Editor Internationalization Web Browser Usage Data Collector Team 	 Enable analysis thread memory control. Force to stop analysis on heap size exceed 462 (ME Heap status: 40M of 93M The Java virtual machine may fail to allocate memory even if the amount of used memory does not exceed this maximum size.
	Restore Defaults Apply
?	OK Cancel

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

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

2.4. Setting preferences of analysis editors and analysis results

Talend Open Studio for Data Quality enables you to decide what sections to fold by default when you open any of the connection or analysis editors. It also offers the possibility to set up the display of all analysis results and whether to show or hide the graphical results in the different analysis editors.

Prerequisite(s): *Talend Open Studio for Data Quality* is open.

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

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

📽 Preferences	
type filter text General ATL EMF Compare Help Install/Update Java Run/Debug Talend Otata Explorer Metadata Profiling Cittor Internationalization Web Browser Usage Data Collector Team	Editor Folding Choose how to fold the section when an editor is open Outfold all sections Fold all sections Outfold first section Analysis results folding Choose how to present the indicator results when the editor is open Outfold all analyzed elements Outfold all indicators Graphics Hide graphics in analysis results page Analyzed Items Per Page 5 Business Rules Per Page 5 Restore Defaults Apply
?	OK Cancel

- 3. In the **Folding** area, select the check box(es) corresponding to the display mode you want to set for the different sections in all the editors.
- 4. In the **Analysis results folding** area, select the check boxes corresponding to the display mode you want to set for the statistic results in the **Analysis Results** view of the analysis editor.
- 5. In the **Graphics** area, select the **Hide graphics in analysis results page** option if you do not want to show the graphical results of the executed analyses in the analysis editor. This will optimize system performance when you have so many graphics to generate.
- 6. In the Analyzed Items Per Page field, set the number for the analyzed items you want to group on each page.
- 7. In the Business Rules Per Page field, set the number for the business rules you want to group in each page.



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

8. Click Apply and then Ok to validate the changes and close the [Preferences] dialog box.

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

2.5. Displaying/hiding the help content

Talend Open Studio for Data Quality provides cheat sheets that can be opened in the Studio. You can use these cheat sheets as a quick reference that guides you through all common tasks in Talend Open Studio for Data Quality.

Talend Open Studio for Data Quality also provides a help panel attached to all wizards used in *Talend Open Studio for Data Quality* to create the different types of analyses or to set thresholds on indicators.

2.5.1. Cheat sheets

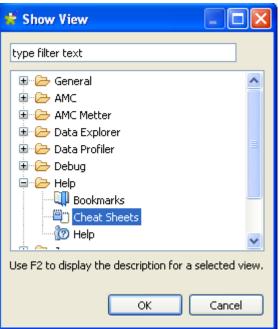
When you open the **Profiling** perspective for the first time, the cheat sheet panel will open by default. However, this view is closed when you switch away from the **Profiling** perspective.

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

To display the cheat sheets in the **Profiling** perspective, do one of the following:

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

The [Show View] dialog box opens.



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

Or,

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

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

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

The selected cheat sheet opens in the *Talend Open Studio for Data Quality* main window. Use the local toolbar icons to manage the display of the cheat sheets.

File Edit Window Help			
: 🔚 : 🖳 🏟		🖹 📔 Data Profiler 😴 Data Integrat	
🔯 DQ Rep 🛛 🗖 🗖	📊 catalog_Analysis 0.1 🛛 💙 🎦 🗖	🖺 Cheat Sheets 🛛 🗎 🔽 🗖	
🗞 🖻 🖨 🗸	i 🖻 🖶 🔡 🛣	Create a new catalog analysis	
	Connection Analysis	▼ Introduction	
🗄 👪 Metadata 🗄 🗑 Recycle Bin	Analysis Metadata Set the properties of analysis.	This type of analysis analyzes one specific catalog in a database. The result of the analysis gives information about the content of the catalog.	
	Name: catalog_Analysis	Click to Begin	
	Purpose:	 select a catalog to analyze 	
	Description:	Note that catalog analysis is possible only with databases that include catalog entities in their physical structure. This step opens the Create New Analysis wizard. You can right click on the "Analysis" folder or on a catalog (or click the link below). This wizard helps you to fill the	
	Author: user@company.com Status: development	analysis metadata (name, purpose, description). At the end it opens the corresponding editor in which you can define the settings of the analysis.	
	▼ Analysis Parameters	Review your settings	
🔗 Detail Vi 🛛 🗖 🗖	Filter on tables:	This step allows you to review and change your analysis	
General	Filter on views:	settings, if necessary, in the open catalog editor.	
Name: Purpose:	Analysis Summary		
Description:	✓ Statistical information	✓ Analyze	
Type:	Analysis Settings	To execute the analysis, click on the "Run" button at the bottom of the editor (or click the link below). You can run	

2.5.2. Help panel



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

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

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

The [Web Browser] view opens.

📽 Preferences	
type filter text	Web Browser 🔶 🔹 🗢
Components Data Explorer Data Profiler Components Web Browser Documentation	Block browser help
Import/Export	Restore Defaults Apply
?	OK Cancel

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

🗳 Create New Analysis		×
Select a wizard	¯® ∰ 👜 🖣 🖋 🗢 ⇒	×
Create a new Analysis	Catalog Analysis:	^
type filter text		
Connection Analysis Catalog Analysis	Catalog Structure Overview	
Catalog Structure Overview Catalog Structure Overview Ca	This kind of analysis gives you an overview of the content of a catalog. It computes the number of tables and the number of rows per table for each catalog and/or schema	
E Column Correlation Analysis	Note that Oracle (with some other DBMS) does not define catalogs but schemas. If you want to analyze a database on Oracle, choose a "Schema analysis" instead.	
	See also: <u>Connection Analysis</u> <u>Catalog Analysis</u> <u>Schema Analysis</u> <u>Table Analysis</u>	~
C Back Next > Fin	Go To:	lex

All the wizards in Talend Open Studio for Data Quality will display with the help panel.

2.6. Displaying the error log view and managing the log files

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

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

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

The [Show View] dialog box opens.

😤 Show View	
type filter text	
General General General General General General General Component Designer General Console General General Console General Ge	
Ок (Cancel

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

The Error Log view opens in Talend Open Studio for Data Quality.

	🗙 🗎 🧬 🏹 🗖 I
/orkspace Log	
201	4
Message Plug-in	Date 🔻
🙆 2011-03-03 13:57:45,382 WARN org.talend.core.mc org.talend.platform.logging	3/3/11 1:57 PM
& 2011-03-03 13:57:45,382 WARN org.talend.core.mc org.talend.platform.logging	3/3/11 1:57 PM
i 2011-03-03 10:42:06,445 INFO org.talend.core.rep: org.talend.platform.logging	3/3/11 10:42 AM
i 2011-03-02 17:04:03,808 INFO org.talend.commons org.talend.platform.logging	3/2/11 5:04 PM
i 2011-03-02 16:38:53,644 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:38 PM
i 2011-03-02 16:27:22,760 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:27 PM
i 2011-03-02 16:27:05,604 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:27 PM
i 2011-03-02 16:27:05,495 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:27 PM
i 2011-03-02 16:21:24,490 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:21 PM
i 2011-03-02 16:21:21,178 INFO org.talend.core.moc org.talend.platform.logging	3/2/11 4:21 PM
i 2011-03-02 16:21:21,178 INFO org.talend.core.moc org.talend.platform.logging	3/2/11 4:21 PM
🕙 2011-03-02 16:20:34,851 ERROR org.talend.commor org.talend.platform.logging	3/2/11 4:20 PM
😢 2011-03-02 16:19:58,743 ERROR org.talend.commor org.talend.platform.logging	3/2/11 4:19 PM
9 2011-03-02 16:19:50,337 ERROR org.talend.commor org.talend.platform.logging	3/2/11 4:19 PM
9 2011-03-02 16:19:42,493 ERROR org.talend.commor org.talend.platform.logging	3/2/11 4:19 PM
9 2011-03-02 16:19:34,665 ERROR org.talend.commor org.talend.platform.logging	3/2/11 4:19 PM
9 2011-03-02 16:19:26,962 ERROR org.talend.commor org.talend.platform.logging	3/2/11 4:19 PM
2011-03-02 16:18:56,760 ERROR org.talend.commor org.talend.platform.logging	3/2/11 4:18 PM
🚯 2011-03-02 16:18:40,886 WARN org.talend.core.md org.talend.platform.logging	3/2/11 4:18 PM
🚯 2011-03-02 16:18:40,886 WARN org.talend.core.mc org.talend.platform.logging	3/2/11 4:18 PM
i 2011-03-02 16:12:53,928 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:12 PM
i 2011-03-02 16:12:43,053 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:12 PM
i 2011-03-02 16:12:42,943 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:12 PM
i 2011-03-02 16:11:28,695 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:11 PM
i 2011-03-02 16:11:16,196 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:11 PM
i 2011-03-02 16:11:16,055 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:11 PM
i 2011-03-02 16:10:57,884 INFO org.talend.commons org.talend.platform.logging	3/2/11 4:10 PM
1	

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

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

Each error log in the list is preceded by an icon that indicates the severity of the log: \mathfrak{D} for errors, $\overset{\texttt{a}}{}$ for warnings and \mathbf{i} for information.

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

9

😤 Event Details 📃 🗖 🔀		
Date: Severity:	3/2/11 4:20 PM	
Message:	2011-03-02 16:20:34,851 ERROR org.talend.commons.ui.runtime.exception.ExceptionHandl er - java.sql.SQLException: Access denied for user	
Exception	n Stack Trace:	
at org.t. at org.t. at org.t. at org.t. at org.t. at org.t. at org.t.	ng.RuntimeException: java.sql.SQLException: Access denied for user 'rool alend.core.model.metadata.builder.database.JDBCDriverLoader.getConr alend.core.model.metadata.builder.database.ExtractMetaDataUtils.conn alend.core.model.metadata.builder.database.ExtractMetaDataFromData alend.repository.ui.utils.ManagerConnection.check(Unknown Source) alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor alend.repository.ui.wizards.metadata.connection.database.DatabaseFor	
Session D	Data:	
java.ver java.ver BootLoa	buildId=unknown rsion=1.6.0_21 ndor=Sun Microsystems Inc. ader constants: OS=win32, ARCH=x86, WS=win32, NL=en_US nd-line arguments: -os win32 -ws win32 -arch x86	
?	ОК	

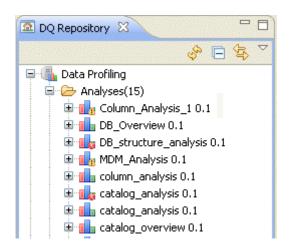
5. If required, click the 🗐 icon in the [Event Detail] dialog box to copy the event detail to the Clipboard and then paste it anywhere you like.

2.7. Icons appended on analyses names in the DQ Repository

When you create any analysis type from *Talend Open Studio for Data Quality*, a corresponding analysis item is listed under the **Analyses** folder in the **DQ Repository** tree view.



The number of the analyses created in *Talend Open Studio for Data Quality* will be indicated next to this **Analyses** folder in the **DQ Repository** tree view.



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

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

2.8. Opening new editors

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

Prerequisite(s): An analysis editor or an SQL query editor is open in Talend Open Studio for Data Quality.

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

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

In the analysis editor:

📊 Column_Anal	ysis 0.1 🛛		1
) 🗆 🗄 🖁 🖈 🔛 🖃		Restore Move	
Column Analysis		Size ► Minimize	
▼ Analysis I		Maximize	
Set the analy	sis properties.	Close	
Name:	Column_Analysis	Close Others	
Purpose:		Close All	
		New Editor	~
Description:			
			~
Author:	user@company.c	om	
Status:	development 🗸		
 Analyzed 	Columns		
Connection:	MySQL1	Versio	n:0.1
Select column	<u>ns to analyze</u>		
Select indicat	ors for each colum	<u>n</u>	
🖻 🕀 🤖			⇒∥ 1/1

In the SQL editor:

□ TEST_TOP_0.1.sql 🛛	Deshaus	L
i 🗶 🗟 🕼 🗁 🖳 🖉 🛸 i	Restore Move •	Limit Rows: 100
1 MySQL dump 10.11	Size 🕨	
2 3 Host: localhost	Minimize Maximize	5T TOP
4		
5 Server version 5	Close Close Others	
6 7 /*!40101 SET @OLD_CHA:	Close All	CENT=00CHARACTER_SET_CLIENT */;
<	New Editor	
Messages 🔀		

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

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

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

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

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

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

Connections	🛛 🖸 DQ Reposito 🛛 🖓 🗖	🗊 SQL Editor (1).sql 🛛 📾 SQL Editor (2).sql 🛛
 ■ Image MySQL (2 sessions) Image model Image model		i 🖈 ि ि ि ि ि 🖳 🖉 マ ▾ i MySQL/root 💌 ✔Limit
		Messages 🔀 Status Location SQL Text

A contextual menu is displayed.

2. Select New SQL Editor.

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

To open an empty SQL editor from the **Profiling** perspective, see the procedure outlined in Section 10.1, "Creating and storing SQL queries via *Talend Open Studio for Data Quality*".



Chapter 3. Before you begin profiling data

Talend Open Studio for Data Quality enables you to profile data in databases, in files, or on Master Data Management (MDM) servers.

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

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

3.1. Creating connections to different data sources

Talend Open Studio for Data Quality enables you to create connections to databases, files or MDM servers in order to profile data in such different data sources.

3.1.1. Connecting to a database

Before proceeding to analyze data in a specific database, you must first set up the connection to this database. *Talend Open Studio for Data Quality* enables you to create a connection on the DataBase Management System (DBMS) and show the content of the database in the **DQ Repository** tree view.

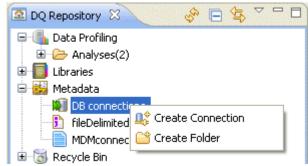


The logical and physical structure of data differs from one relational database to another. The highest level structure "Catalog" followed by "Schema" and finally by "Table" is not applicable to all database types. Thus connection to different databases are reflected by different tree levels and different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

Prerequisite(s): Talend Open Studio for Data Quality is open.

To create a database connection, do the following:

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



The [Database Connection] wizard opens.

😴 Database Connection		
New Database Connection on repository - Step 1/2 Define the properties		
Name	SQL_Connection	
Purpose	Connecting to MySQL database	
Description		
	~	
Author	user@company.com	
Locker		
Version	0.1 Mm	
Status	development 💌	
Path	Select	
?	< Back Next > Finish Cancel	

- 2. In the **Name** field, enter a name for this new database connection.
- 3. If required, set other connection metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.

😴 Database Connection 📃 🗖 🔀		
New Database Connection on repository - Step 2/2 Define the connection parameters		
Database Settings		
DB Type MySQL		•
Db Version	MySQL 5	~
String of Connection	jdbc:mysql://localhost:3306/?noDatetimeStringSync=true	
Login	root	
Password	••••	
Server	localhost	
Port	3306	
DataBase		
Additional parameters	noDatetimeStringSync=true	
Database Properties	Check Null Char 000	
?	< Back Next > Finish	Cancel

- 4. In the **DB Type** field and from the drop-down list, select the type of database to which you want to connect. For example, *MySQL*.
 - If you select to connect to a database that is not supported in *Talend Open Studio for Data Quality* (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on column analyses, see Section 5.3.1, "Defining the columns to be analyzed and setting indicators", and for more information on the Java engine, see Section 5.3.3, "Using the Java or the SQL engine".
- 5. In the **DB Version** field, select the version of the database to which you are creating the connection.
- 6. Enter your login, password, server and port information in their corresponding fields.
- 7. In the **Database** field, enter the database name you are connecting to. If you need to connect to all of the catalogs within one connection, if the database allows you to, leave this field empty.
- 8. Click the Check button to verify if your connection is successful.
- 9. Click Finish to close the [Database Connection] wizard.

A folder for the created MySQL database connection is displayed under **DB** Connection in the **DQ** Repository tree view. The connection editor opens with the defined metadata in *Talend Open Studio for Data Quality*.

🚛 SQL_Conne	ction 0.1 🛛 🗖
i 🖻 🕀	
Connecti	on Settings
▼ Connect	ion Metadata
Set the prop	erties of connection.
Name:	SQL_Connection
Purpose:	Connection to MySQL database
Description:	
Author:	user@company.com
Status:	development 💽
▼ Connect	ion information
The informal	tion of connection.
Login:	root
Password:	••••
Url:	jdbc:mysql://localhost:3306/?zeroDateTimeBehavior=convertToNull&noDatetimeS
	Check

From the connection editor, you can:

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

For information on how to set up a connection to a file, see Section 3.1.2, "Connecting to a file". For information on how to set up a connection to an MDM server, see Section 3.1.3, "Connecting to an MDM server".

3.1.2. Connecting to a file

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

3.1.2.1. How to connect to a delimited file

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

Prerequisite(s): Talend Open Studio for Data Quality is open.

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

1. Expand the **Metadata** node.



- 2. Right-click **FileDelimited connections** and then select **Create File Delimited Connection** to open the **[New Delimited File]** wizard.
- 3. Follow the steps defined in the wizard to create a connection to a delimited file. For further information, see the *Talend Open Studio for Data IntegrationUser Guide*.

You can then create a column analysis and drop the columns to analyze from the delimited file metadata in the **DQ Repository** tree view to the open analysis editor. For further information, see Section 5.5.1, "Analyzing columns in a delimited file".

For information on how to set up a connection to a database, see Section 3.1.1, "Connecting to a database". For further information on how to set up a connection to an MDM server, see Section 3.1.3, "Connecting to an MDM server".



You can create a file delimited connection either from the **Profiling** or the **Integration** perspectives. Once created, this connection is always listed in both perspectives.

3.1.2.2. How to connect to an Excel file

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

Prerequisite(s): Talend Open Studio for Data Quality is open.

To create the Data Source, do the following:

- 1. On the task bar of your desktop, click the **Start** button and then select **Control Panel** to open the corresponding page.
- 2. Double-click Tools and Administrator to open the corresponding page.
- 3. Double-click Data sources (ODBC).

ODBC Data Source	Administrator				
User DSN System DSI	N File DSN Drivers Tracing Connection Pooling About				
User Data Sources:					
Name	Driver Add				
dBASE Files Excel Files MS Access Database	Microsoft Access dBASE Driver (*.dbf, *.ndx Microsoft Excel Driver (*xls, *xlsx, *xlsm, *x Microsoft Access Driver (*.mdb, *.accdb)				
•	4				
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.					
	OK Cancel Apply Help				

A dialog box opens.

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

Create New Data Source	Select a driver for which you want to set up a	a data source.
	Name	Version 🔺
	Driver da Microsoft para arquivos texto (* Driver do Microsoft Access (*.mdb) Driver do Microsoft dBase (*.dbf) Driver do Microsoft Excel(*xls) Driver do Microsoft Paradox (*.db) Driver para o Microsoft Visual FoxPro Ingres	. 6.01.760 6.01.760 6.01.760 6.01.760 6.01.760 1.00.02.(3.50.101 ↓
	< Back Finish	Cancel

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

ODBC Microsoft Exc	? ×	
Data Source Name:	ODBC_Excel	ОК
Description:		Cancel
Database		
Version: Exc	el 97-2000 🔻	Help
Workbook:		
	Select Workbook	
Use Current Dir	ectory	Options>>

- 6. In the **Data Source Name** field, enter a name for the Data Source, and then click the **Select Workbook...** tab to proceed to the step where you link this Data Source to the excel file you want to profile.
- 7. In the open dialog box, browse to the excel file to which you want to link your Data Source.



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

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

You can then create a column analysis and drop the columns to analyze from the excel file metadata in the **DQ Repository** tree view to the open analysis editor. For further information, see Section 5.5.2, "Analyzing columns in an excel file".

For information on how to set up a connection to a database, see Section 3.1.1, "Connecting to a database". For further information on how to set up a connection to an MDM server, see Section 3.1.3, "Connecting to an MDM server".

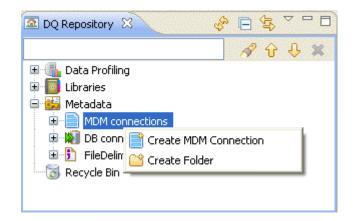
3.1.3. Connecting to an MDM server

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

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

To create an MDM connection, do the following:

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



The [MDM Connection] wizard opens.

*							
Talend MDM							
Create a MI	DMConnection						
Name	MDM_Connection						
Purpose	Connecting to an MDM server						
Description							
	~						
Author	talend@talend.com						
Locker							
Version	0.1 Mm						
Status	production						
Path	Select						
?	< Back Next > Finish Cancel						

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



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

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

-	1	
T	alend MD	м
	←Link Param	eter
	Username	administrator
	password	•••••
	Server	localhost
	Port	8080
		Check
	?	<pre>< Back Next > Finish Cancel</pre>

To set the connection parameters, do the following:

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

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

- 2. Set the connection parameters to the MDM server in the Server and Port fields.
- 3. Click the **Check** button to verify if your connection is successful. A confirmation message is displayed.
- 4. Click **OK** to close the message and then **Next** to proceed to the next step.

*	
Talend MDM	
- Select Version -	
Version	[HEAD]
Data-model	Product 💌
Data-Container	Product 💌
· · · · · · · · · · · · · · · · · · ·	
?	< Back Next > Finish Cancel

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

- 6. From the Data-Model list, select the data model against which master data is validated.
- 7. From the Data-Container list, select the data container that holds the master data you want to access.
- 8. Click **Finish** to validate your changes and close the wizard.

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

Connection	Settings	

▼ Connect	Connection Metadata		
Set the prop	erties of connection.		
Name:	MDM_Connection		
Purpose:	Connecting to an MDM server		
Description:			
Author:	user@company.com		
Status:	development 👽		
▼ Connect	ion information		
The informa	tion of connection.		
Login:	administrator		
Password:	•••••		
Url:	http://localhost:8080/talend/TalendPort Edit		

The display of the connection editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

Check

From the analysis editor, you can:

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

For information on how to set up a connection to a database, see Section 3.1.1, "Connecting to a database". For further information on how to set up a connection to a file, see Section 3.1.2, "Connecting to a file".

3.2. Managing the connections to data sources

Several management options are available for each of the connections created in *Talend Open Studio for Data Quality*.

3.2.1. Managing database connections

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

The sections below explain in detail these management options.

3.2.1.1. How to open/edit a database connection

From *Talend Open Studio for Data Quality*, you can edit the connection to a specific database and change the connection metadata and the connection information.

Prerequisite(s): A database connection is created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

To edit an existing database connection, do the following:

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

🗟 DQ Repository 🛛	Ŷ	E	\$₽¢	\bigtriangledown	
		A	Û	Ŷ	×
🕀 🜗 Data Profiling					
😟 🔟 Libraries					
📮 👪 Metadata					
MDM connections					
🖨 🔯 DB connections					
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □		_			
🗷 🖳 CIF 🦓 Reload database list					
🗉 🦺 CRM 🚰 Database Compare					
DB2 Open					
🖳 🦺 MSS 💥 Delete					
Balling MySi Colores					
ODB OB					
I → ↓ SQL Package Filter					
	_				
FileDelimited connections					
🗄 🐻 Recycle Bin					

The connection editor for the selected database connection is displayed.

: E 🕀	
Connecti	on Settings
Connect	ion Metadata
 Connect 	ion information
The informa	tion of connection.
Login:	root
Password:	••••
Url:	jdbc:mysql://localhost:3306/?zeroDateTimeBehavior=convert
	Check
Connection Set	tings

- 3. Modify the connection metadata in the **Connection Metadata** view, as required.
- 4. Click the **Edit...** button in the **Connection information** view to open the **[Database Connection]** wizard again.

📽 Database Connection				
New Database Connection on repository - Step 1/2 Define the properties				
Name	SQL_Connection			
Purpose	Connecting to MySQL database			
Description				
	~			
Author	user@company.com			
Locker				
Version	0.1 Mm			
Status	development			
Path	Select			
?	< Back Next > Finish Cancel			

5. Modify the database connection information as required and then click **Finish** to validate the modifications.

A message is displayed to show the progress of the operation. Then a dialog box is displayed to list all the analyses that use the selected database connection. It alerts you that if you save the new connection information, all the analyses using the connection will become unusable although they will be always listed in the **DQ Repository** tree view.

😤 URL has been changed 🛛 👂	×
Check connection failed! The following analyses will never be used. Continue saving anyway?	
Column_Analysis requires < <sql_connection>> One_Column requires <<sql_connection>> Set_of_Columns</sql_connection></sql_connection>	
requires < <sql_connection>> MySQLContent requires <<sql_connection>> Age_Analysis requires <<sql_connection>></sql_connection></sql_connection></sql_connection>	
OK Cancel)

6. Click **OK** to save the modifications or **Cancel** to cancel the operation and close the dialog box.

3.2.1.2. How to filter a database connection

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

Talend Open Studio for Data Quality enables you to filter your database connections to list the databases that match the filter you set. This option is very helpful when the number of databases in a specific connection is very big.

Prerequisite(s): A database connection is already created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

To filter a database connection, do the following:

1. In the DQ Repository tree view, expand Metadata and DB Connection in succession.

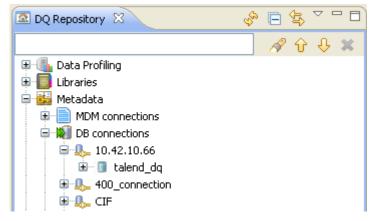
Image: Arror of the second
🕀 🔟 Libraries
MDM connections MDM connections DB connections DB connections DB connections CIF CIF CRM Database Compare Open Database Compare Open Delete Duplicate Open SQL Package Filter Staging_DBZ

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

💕 Package F	ilter	
Package Filt Filter is valid, o	er different filter separated by commas	×
Connection Package Filter	10.42.10.66 talend_dq]
?	Finish	Cancel

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

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



To cancel the filter, do the following:

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

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

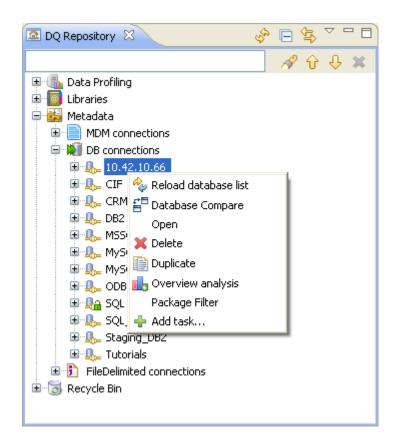
3.2.1.3. How to duplicate a database connection

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

Prerequisite(s):A database connection is created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

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

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



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

3.2.1.4. How to add a task to a database connection

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

Prerequisite(s): A database connection is created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

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

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

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

Properti	es 📃 🗖 🔀
Description:	added a task to reload the database list of this connection in a month time
Priority:	Normal 🔽 Completed
On element:	MySQL
In folder:	RELEASE_421/metadata/connections
Location:	D:/MDM_builds/TMDM_EE/4.2.1_latest/Talend-All-r60995-V4.2.1/workspace/RELEASE
?	OK Cancel

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

The created task is added to the Tasks list.

For more information on how to access the task list, see Section 11.2.4, "Displaying the task list".

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

Talend Open Studio for Data Quality enables you to filter the tables/views to list under any database connection.

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

Prerequisite(s): A database connection is already created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

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

- 1. In the **DQ Repository** tree view, expand **Metadata** and **DB Connection** in succession.
- 2. Expand the database connection in which you want to filter tables/views and right-click the desired catalog/ schema.

🖸 DQ Repository 🛛 🔗 🗖 🕏 🏹 🗖	
🕀 🜗 Data Profiling	^
🕀 🐻 Libraries	
🚍 👪 Metadata	
🖨 🔊 DB connections	
🖨 🦺 MySQL	
😟 🧊 information_schema	
😟 🗊 babili_before	
😟 🗊 cif	
😟 🧻 convert	
😟 🗊 crm	
😟 🗊 crm_demo	
🗄 📋 custon 🙀 Export to Design Workspace	
😟 🧧 demor 📊 Overview analysis	
Table/View Filter	

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

🍯 Table/View	Filter	
Table/View Fi		jt.
Filter is valid, dif	erent filter separated by commas	
Connection	MySQL	
Catalog/Schema	crm_demo	
Table Filter	cu%,pr%	
View Filter	%me	
0	Finish	Cancel

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

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

3.2.1.6. How to delete or restore a database connection

In the *Talend Open Studio for Data Quality*, you can delete a database connection whether it is used by analyses or not. You can also restore a deleted database connection.



When you delete a database connection from the Profiling perspective in Talend Open Studio for Data Quality, it will be automatically deleted from the Metadata node in the Data Integration perspective and vice versa. The deleted connection will go to the Recycle Bins of both perspectives.

Prerequisite(s): A database connection is created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

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

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

DQ Repository	Ş	Ξ	Þ	× '	
		R	Ŷ	÷	×
🕀 🜗 Data Profiling					
🗄 🐻 Libraries					
🚍 👪 Metadata					
MDM connections					
🖃 👹 DB connections					
🗈 🖺 10.42.10.66					
🕀 🦺 CIF 🤣 Reload database list					
🕀 🦺 CRM 🚰 Database Compare					
🕀 🖺 DB2 👘 Open					
🖳 🦺 MSS 💥 Delete					
Hyperate Myperate					
🖬 🔔 ODB 📊 Overview analysis					
🗈 🦺 SQL 🛛 Package Filter					
🖻 🦺 SQL 🕂 Add task					
🕀 🦺 Staging_D82		_			
🕀 Ille Tutorials					
FileDelimited connections					
🗄 📷 Recycle Bin					

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

?	You will always be able to run any analysis that uses the connection moved to the recycle bin. However, an alert message will be displayed next to the connection name in the analysis editor.
	r Analyzed Columns
	Connection: SQL_Connection Version:0.1 This connection "SQL_Connection" is logical deleted!
	Select columns to analyze

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

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

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

📽 Delet	e forever 🔀
ځ 😲	MySQL will be deleted forever (cannot be retrieved). Continue anyway?
	Yes No

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

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

📽 Cannot delete this resource	×
Other items depend on item("SQL_Connection").	
Force to delete all the dependencies.	
🗐 📶 Age_Analysis	~
in requires < <sql_connection>></sql_connection>	
🚊 💼 Age_Average	
requires < <sql_connection>></sql_connection>	_
🗐 📶 Subscription_Date	
in the second	
🖨 📶 SQL_Content_Analysis	
in the second se	
🖨 📶 temperature_correlation	
interpretation in the second s	×
	ОК

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

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

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

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

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

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

🗳 Cannot delete this resource	×
You can't empty the Recycle Bin. Other items depend on these ones. Please delete them one by one.	
🖃 📶 Age_Analysis	~
in requires < <sql_connection>></sql_connection>	
🚊 📶 Age_Average	
in the second se	_
🗐 📶 Subscription_Date	
in requires < <sql_connection>></sql_connection>	
🚊 📶 SQL_Content_Analysis	
in requires < <sql_connection>></sql_connection>	
🗐 📶 temperature_correlation	
👘 🙀 requires < <sql connection="">></sql>	×
	ОК

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

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

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

The database connection is moved back to the Metadata node.

3.2.2. Managing MDM connections

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

The sections below explain in detail these management options.

3.2.2.1. How to open/edit an MDM connection

Prerequisite(s): An MDM connection is already created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.3, "Connecting to an MDM server".

To open an existing MDM connection, do the following:

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

🗟 DQ Repository 🛛		Ş		¢Ę\$	~ (- 8
			P	Û	Ŷ	ж
🕀 🜗 Data Profiling						
🗈 🐻 Libraries						
MDM connections						
E	nection					
🗷 📄 MDM_Connectic	Open			1		
🖻 📄 MDM_connectio	💢 Delete					
🕀 💓 DB connections	📄 Duplicate	1				
🗄 🗓 FileDelimited connec	🐈 Add task					

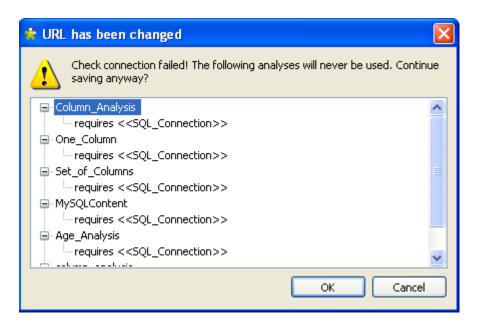
The analysis editor for the selected MDM connection is displayed.

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

*	
Talend MD Create a MI	DMConnection
Name	MDM_Connection
Purpose	Connecting to an MDM server
Description	
Author	talend@talend.com
Locker	
Version	0.1 Mm
Status	production
Path	Select
?	< Back Next > Finish Cancel

5. Modify the MDM connection information as required and then click **Finish** to validate the modifications.

A message is displayed to show the progress of the operation. Then a dialog box lists all the analyses that use the selected MDM connection. It alerts you that if you save the new connection information, all the analyses using the connection will become unusable although they will be listed in the **DQ Repository** tree view.



6. Click **OK** to save the modifications or **Cancel** to cancel the operation and close the dialog box.

3.2.2.2. How to duplicate an MDM connection

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

Prerequisite(s): At least one MDM connection is created in*Talend Open Studio for Data Quality*. For further information, see Section 3.1.3, "Connecting to an MDM server".

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

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

🗟 DQ Repository 🛛		Ş	E	\$		
			P	Û	Ŷ	×
🗉 🜗 Data Profiling						
😟 🐻 Libraries						
🖨 👪 Metadata						
🖃 📄 MDM connections						
😟 📄 Local_MDM_Cor	nection					
😥 📄 MDM_Connectic	Open			1		
😟 📄 MDM_connectio	💢 Delete					
🕀 💓 DB connections	Duplicate	;				
🗄 🗓 FileDelimited connec	💠 Add task					

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

3.2.2.3. How to add a task to an MDM connection

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

Prerequisite(s): An MDM connection is created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.3, "Connecting to an MDM server".

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

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

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

💕 Properti	es 📃 🗖 🔀
Description:	task description
Priority:	Normal 🖌 Completed
On element:	MDM
In folder:	RELEASE_421/metadata/MDMconnections
Location:	D:/MDM_builds/TMDM_EE/4.2.1_latest/Talend-All-r60995-V4.2.1/workspace/RELEASE
?	OK Cancel

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

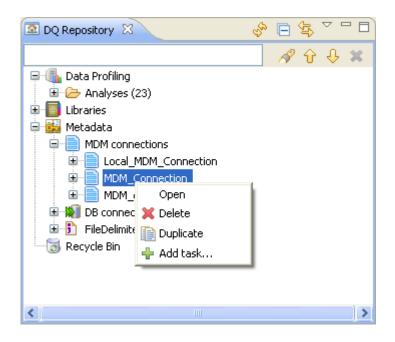
For more information on how to access the task list, see Section 11.2.4, "Displaying the task list".

3.2.2.4. How to delete or restore an MDM connection

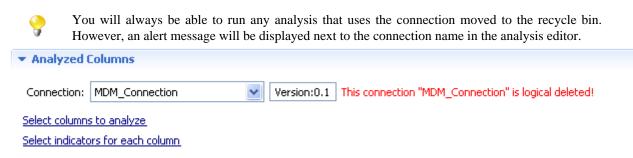
Prerequisite(s): An MDM connection is created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.3, "Connecting to an MDM server".

To delete an MDM connection, do the following:

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



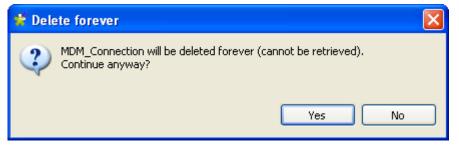
The MDM connection is moved to the Recycle Bin.



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

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

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



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

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

📽 Cannot delete this resource	
Other items depend on item("MDM_Connection").	
Force to delete all the dependencies.	
MDM_ColumnAnalysis1 requires < <mdm_connection>> MDM_ColumnAnalysis3 requires <<mdm_connection>></mdm_connection></mdm_connection>	
	ОК

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

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

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

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

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

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

📽 Cannot delete this resource	×
You can't empty the Recycle Bin. Other items depend on these ones. Please delete them one by one.	
MDM_Column_Analysis2 requires < <mdm_connection1>> min MDM_ColumnAnalysis1 requires <<mdm_connection>> min MDM_ColumnAnalysis3 requires <<mdm_connection>></mdm_connection></mdm_connection></mdm_connection1>	
ОК	כ

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

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

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

The MDM connection is moved back to the Metadata node.

3.2.3. Managing file connections

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

The procedures to manage file connections are the same as those for managing MDM connections. For further information, see Section 3.2.2, "Managing MDM connections".

3.3. Catalogs and schemas in database systems

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

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

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



Chapter 4. Profiling database content

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

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

4.1. Managing database content analyses

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

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

4.1.1. Creating a database content analysis

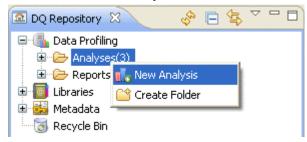
From Talend Open Studio for Data Quality, you can create an analysis of the content of a given database.

Prerequisite(s): At least, one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

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

Procedure 4.1. Defining the analysis

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



The [Create New Analysis] wizard opens.

😤 Create New Analysis	
type filter text	
Connection Analysis Database Structure Overview Catalog Analysis Catalog Analysis Catalog Analysis Column Analysis Column Analysis Column Analysis Column Correlation Analysis	
Rack Next > Finish	Cancel

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

New Analy	ysis
your input i	s valid.
Name	An alusia Mana
Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	~
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis
?	< Back Next > Finish Cancel

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

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

New Analysis	
Choose a connection to analyze	
Connections:	
DB connections	
MySQL	
SQL_Server	
(?) < Back Next > Finish Cancel	

Procedure 4.2. Selecting the database connection you want to analyze

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

New Analysis Add the filters for	Connection Analysis
By default, all tabl	and / or views if needed. es and views will be used in the analysis. ilters with comma ','
Table name filter	cu%,pr%
Table Hame Hiter	
View name filter	%me
? < Ba	ck Next > Finish Cancel

- 3. If required, set filters on tables and/or views in their corresponding fields using the SQL language. By default, the analysis will include all tables and views in the database.
- 4. Click Finish to close the [Create New Analysis] wizard and proceed to the next step.

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

💼 *SQL_Conte	nt_Analysis 🛛	- 0
i 🖻 🕀 🔡	考 💿	
Connectio	on Analysis	^
💌 Analysis	Metadata	
Set the prope	erties of analysis.	
Name:	SQL_Content_Analysis	
Purpose:	exploring the content of MySQL	
Description:		=
Author:		
Status:	Draft 🖌	
Analysis	Parameters	
Analysis	Summary	
Statistica	al information	~

- The display of the connection editor depends on the parameters you set in the [**Preferences**] dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".
- 5. If required, click Analysis Parameters to check/modify filters on table and/or views, if any.

▼ Analysis Parameters			
Filter on tables:	cu%,pr%		
Filter on views:	%me		
📃 Reload databases			

- You can reload all databases on the server if you select the **Reload databases** check box in the **Analysis Parameter** view and run the overview analysis. When you try to reload a database, a message will prompt you for confirmation as any change in the database structure may affect existing analyses.
- 6. If required, click **Analysis Summary** to show all the parameters of the current analysis along with the current analysis execution status.
- 7. Click the save icon on top of the editor and then press **F6** to execute the current analysis. A message opens to confirm that the operation is in progress.

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

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

🔺 Catalog		#rows	#tables	#rows/table	e #views	#rows/view	#keys	#indexes	
🔋 crm_demo		25977	18	1443.17	0	NaN	13	13	
customers		246	6	41.00	0	NaN	2	2	
📑 employee		16	3	5.33	0	NaN	1	1	
🔋 examples		6	2	3.00	0	NaN	2	2	
🔋 exodb_tdq		185	21	8.81	0	NaN	21	21	
🔋 information_sche	ema	0	0	NaN	0	NaN	0	0	
🔋 marketing_depai	rtment	0	0	NaN	0	NaN	0	0	
🔳 mysql		1981	24	82.54	0	NaN	43	50	
🔋 spagobi_tdq		609	37	16.46	0	NaN	59	120	
🔋 talend		7	1	7.00	0	NaN	1	1	
talend_dq 📐		7040	12	662.33	17	19.82	16	22	
🛭 test 🛛 🐧	0ver\	/iew analys	is	10.00	0	NaN	0	0	
🔋 test_dataprofile	r	203	13	15.62	0	NaN	0	0	
🔋 test_top		201	1	201.00	0	NaN	1	1	
🗊 weka		33	1	33.00	0	NaN	0	0	
Table	#rows	;	#keys		indexes	<u>_</u> _	View		#
🔝 tdq_analysis	13		1		1			_v_all_run	
tdq_analyzed_el			1	1				_v_all_run	
🔣 tdq_analyze	0		1	1				_v_all_run	
🔣 tdq_calendar	6209		1	1			tdq_v_analysis		11
🔣 tdq_day_time	1440		1	1				_v_analyze	9
🔢 tdq_indicator	13		1	1			🔚 tdq	_v_ind_histo	16

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



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

10. If required, click any column header in the analytical table to sort alphabetically data listed in catalogs or schemas. You can also sort alphabetically all columns in the result table doing the same.



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

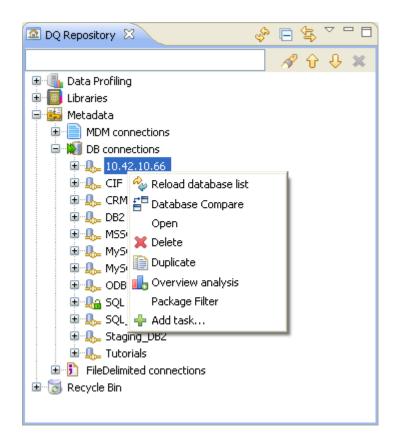
4.1.2. Creating a database content analysis in shortcut procedure

In Talend Open Studio for Data Quality, you can create an analysis of the content of a given database directly from the **DB** Connection folder in the **DQ** Repository tree view.

Prerequisite(s): At least, one database connection is set in Talend Open Studio for Data Quality. For further information, see Section 3.1.1, "Connecting to a database".

To create a database content analysis, do the following:

- 1. Right-click the database for which you want to create content analysis.
- 2. From the contextual menu, select Overview analysis .



This way, you do not have to specify in the new analysis wizard either the type of analysis you want to carry out or the DB connection to analyze. Otherwise, all other procedural steps are exactly the same as in Section 4.1.1, "Creating a database content analysis".

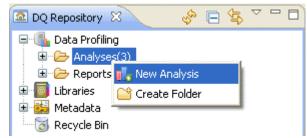
4.1.3. Creating a catalog analysis

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

Prerequisite(s): At least one database connection has been created in *Talend Open Studio for Data Quality* to connect to a database that uses the "catalog" entity.

Procedure 4.3. Defining the analysis

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



The [Create New Analysis] wizard opens.

🗳 Create New Analysis	
Select a wizard	
Create a new Analysis	
type filter text	
Connection Analysis	
Catalog Analysis	
🕀 🔁 Schema Analysis	
🗄 🗁 Table Analysis	
 Column Analysis Redundancy Analysis 	
Column Correlation Analysis	
? < Back Next > Finish	Cancel

- 3. Expand the Catalog Analysis node and click Catalog Structure Overview.
- 4. Click the **Next** button to proceed to the next step.

Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	×
Author	
Status	production 💌
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis

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

9

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

atalogs: 🖃 💓 DB conr	nections	
🗷 😥 DB2		
🖨 🦺 Mys	5QL	
	information_schema	
	babili_before	
🗊	cir convert	
- 1		
🗊	crm_demo	
	crm_demo	

Procedure 4.4. Selecting the catalog you want to analyze

- 1. Expand **DB Connections** and the database that include catalog entities in its physical structure and select a catalog to analyze.
- 2. Click **Next** to proceed to the next step.

New Analysis

Add the filters for Connection Analysis

By default, all tabl	and / or views if needed. es and views will be used in the analysis. 'ilters with comma ','
Table name filter	cu%,pr%
View name filter	%me
?	ck Next > Finish Cancel

- 3. If required, set filters on tables and/or views in their corresponding fields using the SQL language. By default, the analysis will include all tables and views in the catalog.
- 4. Click Finish to close the [Create New Analysis] wizard.

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

🚹 MySQL_Cata	log_Analys 🛛 🎽	
: 🖻 🕀 🔛	* 💿	
Connectio	on Analysis	^
💌 Analysis I	Metadata	
Set the prope	rties of analysis.	
Name:	MySQL_Catalog_Analysis	
Purpose:	exploring the content of a specific catalog	
Description:		≣
Author:		
Status:	Draft 💽	
Analysis I	Parameters	
Analysis !	Summary	
Statistica	l information	~

The display of the analysis editor depends on the parameters you set in the [Preferences] dialog box.
For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

- 5. If required, click **Analysis Parameters** to check/modify filters on table and/or views, if any.
- 6. If required, click **Analysis Summary** to show all the parameters of the current analysis along with the current analysis execution status.
- 7. Click the save icon on top of the editor and then press **F6** to execute the current analysis. A message opens to confirm that the operation is in progress.

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

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

Catalog ;	#rows	#tables	#rows/table	#views	#rows/view	#keys	#indexes
🔋 crm_demo 2	5977	18	1443.17	0	NaN	13	13
Table	#rows		#keys	#inde	exes	~	View
🛄 account	11		1	1			
🛄 currency	72		2	2		=	
📰 customer	10281		1	1			
🧮 department	12		1	1		_	
📰 employee	1155		1	1			
🧰 inventory_fa	5000		0	0			
🛄 position	18		1	1			
💷 product	1560		1	1			
🚥 product i class	110		0	0	1	✓	

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



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

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

4.1.4. Creating a schema analysis

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

Prerequisite(s): At least one database connection has been created in *Talend Open Studio for Data Quality* to connect to a database that uses the "schema" entity, for example the DB2 database. For further information, see Section 3.1.1, "Connecting to a database".

Procedure 4.5. Defining the analysis

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



The [Create New Analysis] wizard opens.

🎽 Create New Analysis	
Select a wizard Create a new Analysis	
type filter text	
 Connection Analysis Catalog Analysis Catalog Analysis Cohema Analysis Cohema Structure Overview 	
(?) < Back Next > Finish	Cancel

- 3. Expand the Schema Analysis node and click Schema Structure Overview.
- 4. Click the **Next** button to proceed to the next step.

New Analy	/sis
your input is	s valid.
Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
Author	
Status	production 😪
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis
?	< Back Next > Finish Cancel

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

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

New Analysis Choose a schema to analyze	
Schemas:	
DB connections DB2 DB2 TALEND talend MySQL SQL_Server	
Rack Next > Fire	nish Cancel

9

Procedure 4.6. Selecting the schema you want to analyze

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

New Analysis Add the filters for	Connection Analysis
Set filter on tables	and / or views if needed.
	les and views will be used in the analysis. filters with comma ','
Table name filter	cu%,pr%
View name filter	%me
? < Ba	ack Next > Finish Cancel

- 3. If required, set filters on tables and/or views in their corresponding fields using the SQL language. By default, the analysis will include all tables and views in the catalog.
- 4. Click **Finish** to close the **[Create New Analysis]** wizard.

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

🚹 DB2_Schema	a_Analysis 0.1 🕱	
i 🖻 🕀 🔡	X 🕑	
Connectio	on Analysis	
💌 Analysis	Metadata	
Set the prope	erties of analysis.	
Name:	DB2_Schema_Analysis	
Purpose:	Overview of schema structure in a DB2 database	
Description:		
Author:	user@company.com	
Status:	development 💽	
	Parameters	
Analysis	Summary	
→ Statistica	al information	
<		>



The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

- 5. If required, click Analysis Parameters to check/modify filters on table and/or views, if any.
- 6. If required, click **Analysis Summary** to show all the parameters of the current analysis along with the current analysis execution status.
- 7. Click the save icon on top of the editor and then press **F6** to execute the current analysis. A message opens to confirm that the operation is in progress.

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

Statistical informations

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

Schema	#rows	#tables	#rows/table	#views	#rows/view	#keys	#indexes
ROOT	8455	388	21.79	0	NaN	1	389
🔺 Table	#rows	#keys	#indexes	~	Viev	J	
FEATURE3271	_	0	1				
SCDDEST	6	0	1				
E SCDTEST	4	0	1				
TABLE06U8HB	0	0	1				
TABLEOG5MGA	¥ 0	0	1				
TABLE0G7J6T	0	0	1				
TABLE1NHCNH	1 0	0	1				

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



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

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

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

After analyzing the content of a database as outlined in Section 4.1.1, "Creating a database content analysis", you can display the details of the key and index of a given table. This information could be very interesting for the database administrator.

Prerequisite(s): At least one database content analysis has been created and executed in *Talend Open Studio for Data Quality*.

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

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

🔺 Catalog	#rows	#tables	#rows/table	#views	#rows/view	#keys
🗑 crm_demo	25977	18	1443.17	0	NaN	13
🔳 customers 🛛 😽	246	6	41.00	0	NaN	2
🔋 employee	16	3	5.33	0	NaN	1
examples	6	2	3.00	0	NaN	2
🔋 exodb_tdq	185	21	8.81	0	NaN	21
🔋 information_schema 👘	0	0	NaN	0	NaN	0
🔋 marketing_department	0	0	NaN	0	NaN	0
📑 mysql	1981	24	82.54	0	NaN	43
🔋 spagobi_tdq	609	37	16.46	0	NaN	59
🔋 talend	7	1	7.00	0	NaN	1
🔋 talend_dq	7948	12	662.33	17	19.82	16
🔋 test	10	1	10.00	0	NaN	0
🔋 test_dataprofiler	203	13	15.62	0	NaN	0
<pre>test_top</pre>	201	1	201.00	0	NaN	1
🔋 weka	33	1	33.00	0	NaN	0

Statistical information

Table	#rows	#keys	#indexes	^
🧮 account	11	1	1	
eurrency	72	2	2	
🧱 customer	10281	1	1	
🧱 department	12	1	1	
🎞 employee	1155	1	1	
inventory_fact_1998	5000	0	0	
🛄 position	18	1	1	
== product	1560	1	1	
🚥 product i class	110	n	n	×

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

Table	#rows	#keys	#indexes	
🥅 account	11	1		
currency	72	2 🍎 ^{Vie}	w keys	
🧱 customer	10281	1	1	
🧮 department	12	1	1	
🧮 employee	1155	1	1	
inventory_fact_1998	5000	0	0	
i position	18	1	1	
🥅 product	1560	1	1	_
🚥 product, class	110	n	n	~

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

🗔 Database De	etail 🛛			' 🗆	🗄 Database Structure 🛛 📑 🗖
Primary Keys	Exported Keys	Imported K	eys Indexes	•	MySQL/root 🔀
COLUMN_NAM	ME KEY_SEQ	PK_NAME			Information_schema
account_id	1	PRIMARY			🖨 📲 crm_demo
					🗄 🧀 Local Temporary
					📮 🗁 Tables
					🕮 📲 account
					😟 🔠 currency
					🗉 🧮 customer



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

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

💩 Database Det	ail 🛛				🛱 Database Structure 🛛 📑 🗖 🗖
Indexes Privile	ges Colun	nn Privileges	Row Ids	s Version 🔹	MySQL/root 🔀
Is Non Unique	Qualifier	Name	Туре	Ordinal Positior	😐 🔡 information_schema
false	<null></null>	PRIMARY	OTHER	1	📮 🖼 crm_demo
					🗉 🗀 Local Temporary
					📮 🗁 Tables
					🖳 🕀 📰 account
					😟 🖽 customer

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

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

4.3. Tracking data changes in source databases

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

From *Talend Open Studio for Data Quality*, you can compare the connection structure displayed in the **DQ Repository** tree view with the database structures itself to locate possible differences. Then you can synchronize the connection structure in the tree view with the actual database structure.



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

4.3.1. Comparing tree-view metadata structures with database structures

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

Talend Open Studio for Data Quality takes a connection structure in the **DQ Repository** tree view and compares it to the database trying to locate all structure differences and display these differences in the **Compare** view.

You can then, if necessary, synchronize the connection structure in the tree view with the database structure. For more information, see Section 4.3.2, "Synchronizing the connection structure with the database structure".

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

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

4.3.1.1. How to compare catalog and schema lists

Prerequisite(s): A database connection has been already created in *Talend Open Studio for Data Quality*.

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

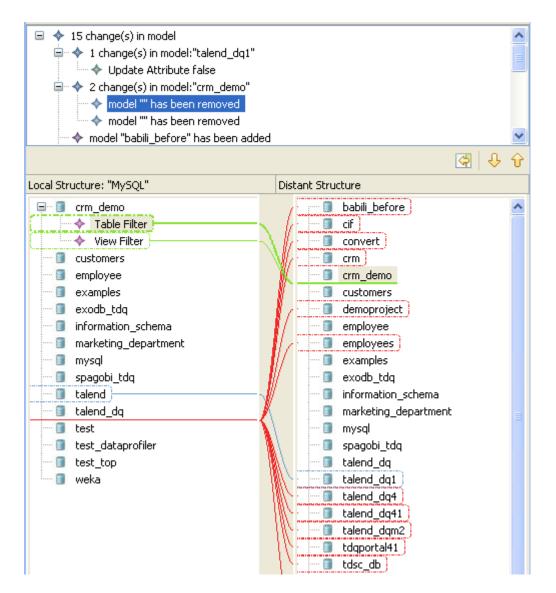
- 1. In the **DQ Repository** tree view, expand the **Metadata** and the **DB connection** folders in succession.
- 2. Right-click the DB connection for which you want to compare the metadata structure with the database structure and select **Database Compare**.

Data Profiling Data Profiling Data Profiling Data Profiling Dibraries Metadata MDM connections DB connections D connections	🗟 DQ Repository 🛛	Ŷ		\$		
Libraries Metadata MDM connections MDB connections DB connections Delete Dopen Delete Duplicate Duplicate Duplicate DD connections DD connections DD connections DD connections DB connections DB connections DD connections DD connections DB connections D			A	Û	Ŷ	ж
Staging_D82 S	Libraries Metadata Motadata Motadata Motadata Motadata Motadata DB connections DB connections DB connections Package Filter Package Filter<		R	Ŷ	Ŷ	*

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

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

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



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

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

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

4. If required, right-click a specific catalog in this view to display a contextual menu where you can select Compare the list of tables or Compare the list of views. This will display respectively the table list or the view list of the selected catalog. For further information about comparing table lists, see Section 4.3.1.2, "How to compare table lists"



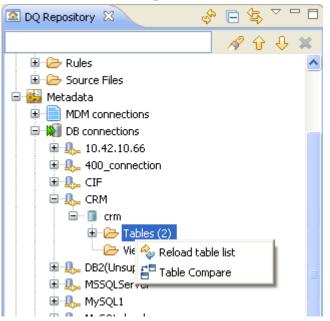
If you select a specific catalog in this list and press the \mathbf{T} or \mathbf{V} keys on your keyboard, you can display respectively the table or view lists of the selected catalog.

4.3.1.2. How to compare table lists

Prerequisite(s): A DB connection has already been created in Talend Open Studio for Data Quality.

To compare a table list, do the following:

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



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



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

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

E Compare 🛛	
 4 change(s) in model 2 change(s) in model:"addr" Update Attribute false Update Attribute false model "addr1" has been added model "adrr3" has been added 	
	🔄 🕂 🗘
Local Structure: "Staging_DB2"	Distant Structure
customer_sample	addr addr1 addr1 customer_sample

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

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

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

4. If required, right-click a specific table in the **Compare** view to display a contextual menu. Select **Compare the list of columns** to display the columns list of the selected table. For further information, see Section 4.3.1.3, "How to compare column lists"



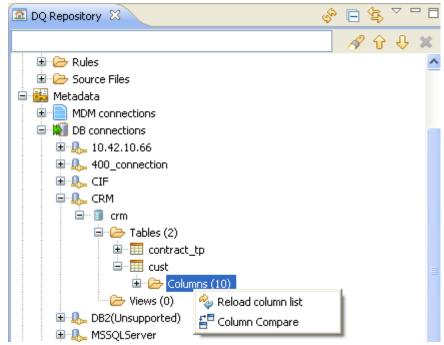
If you select a specific table in the **Compare** list and press the **C** key on your keyboard, you can display the column list of the selected table.

4.3.1.3. How to compare column lists

Prerequisite(s): A database connection has been created in Talend Open Studio for Data Quality.

To compare a column list, do the following:

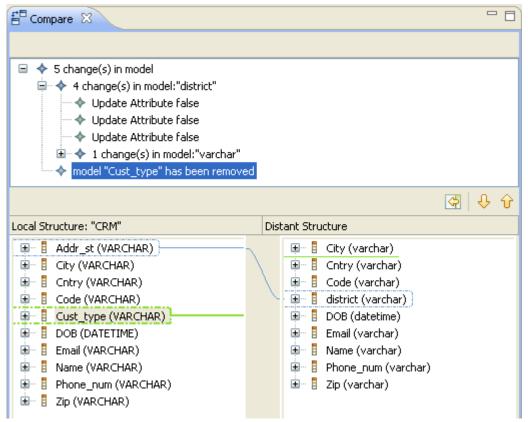
- 1. In the DQ Repository tree view, expand the Metadata and the DB connection folders in succession.
- 2. Browse through the entities in your database connection to reach the **Columns** folder you want to compare with that of the database.
- 3. Right-click the **Columns** folder and select **Column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



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

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

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



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

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

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

4.3.2. Synchronizing the connection structure with the database structure

From *Talend Open Studio for Data Quality*, you can synchronize the connection structure displayed in the **DQ Repository** tree view with the database structures to eliminate any incoherences. You can perform synchronization at the following three different levels:

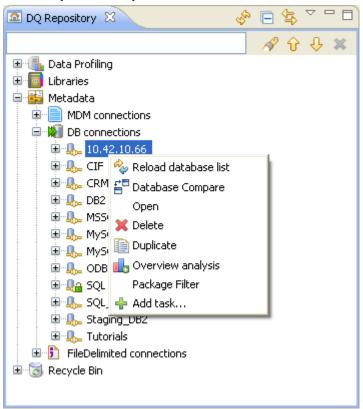
- **DB connection**: to refresh the catalog and schema lists,
- Tables: to refresh the list of tables,
- Column: to refresh the list of columns.

4.3.2.1. How to synchronize catalog and schema lists

Prerequisite(s): A DB connection has been created in Talend Open Studio for Data Quality.

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

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



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

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

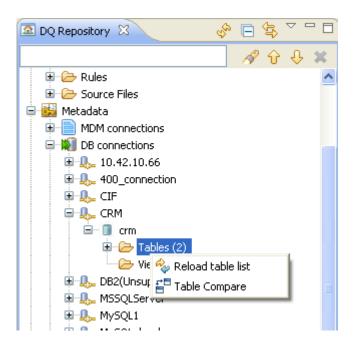
The selected database connection is updated with the new catalogs and schemas, if any.

4.3.2.2. How to synchronize table lists

Prerequisite(s): A DB connection has already been created in Talend Open Studio for Data Quality.

To synchronize a table list, do the following:

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



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

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

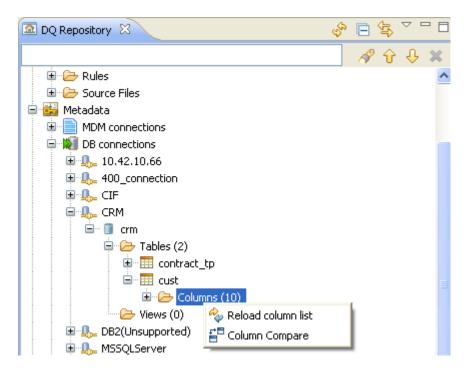
The selected table list is updated with the new tables in the database, if any.

4.3.2.3. How to synchronize column lists

Prerequisite(s): A DB connection has been created in Talend Open Studio for Data Quality.

To synchronize a column list, do the following:

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



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

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

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



Chapter 5. Column analyses

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

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

5.1. Steps to analyze a column

From Talend Open Studio for Data Quality, you can examine and collect statistics and information about:

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

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

- 1. Connecting to the data source being a database, a file or an MDM server. For further information, see Chapter 6, *Table analyses*.
- 2. Defining one or more columns on which to carry out data profiling processes that will define the content, structure and quality of the data included in the column(s).
- 3. Settings predefined system indicators or indicators defined by the user on the column(s) that need to be analyzed or monitored. These indicators will represent the results achieved through the implementation of different patterns.
- 4. Adding to the column analyses the patterns against which you can define the content, structure and quality of the data.

For further information, see Section 5.3.6.1, "How to add a regular expression or an SQL pattern to a column analysis" and

The Section 5.3, "Analyzing columns in a database" explains in detail the procedures to analyze the content of one or multiple columns in a database.

The Section 5.4, "Analyzing master data on an MDM server" explains in detail the procedures to analyze master data on an MDM server.

The Section 5.5, "Analyzing data in a file" explains in detail the procedures to analyze columns in delimited or excel files.

5.2. Data mining types

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

▼ Analyzed Columns				
Connection: MySQL	~			
Select columns to analyze Select indicators for each column	L			
Analyzed Columns	Datamining Type	Pattern	UDI	Operation
fullname (varchar) for Row Count Null Count Jostinct Count Jostint Jostinct Count Jostint Jost	Nominal Nominal Interval Unstructured Text Other Nominal Nominal			****

These data mining types help *Talend Open Studio for Data Quality* to choose the appropriate metrics for the associated column since not all indicators (or metrics) can be computed on all data types.

Available data mining types in *Talend Open Studio for Data Quality* are: **Nominal, Interval, Unstructured Text** and **Other**. The sections below describe these data mining types.

5.2.1. Nominal

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

In *Talend Open Studio for Data Quality*, the mining type of textual data is set to nominal. For example, a column called *WEATHER* with the values: *sun, cloud* and *rain* is nominal.

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

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

5.2.2. Interval

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

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

5.2.3. Unstructured text

This is a new data mining type introduced by *Talend Open Studio for Data Quality*. This data mining type is dedicated to handle unstructured textual data.

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

5.2.4. Other

This is another new data mining type in *Talend Open Studio for Data Quality*. This type designs the data that *Talend Open Studio for Data Quality* does not know how to handle yet.

5.3. Analyzing columns in a database

Talend Open Studio for Data Quality enables you to analyze the content of one or multiple columns and execute the created analyses using the Java or the SQL engine. This type of analysis provides statistics about the values within each column.

When you use the Java engine to run a column analysis, you can view the analyzed data according to parameters you set yourself. For more information, see Section 5.3.3, "Using the Java or the SQL engine".



When you use the Java engine to run a column analysis on big sets or on data with many problems, it is advisable to define a maximum memory size threshold to execute the analysis as you may end up with a Java heap error. For more information, see Section 2.3, "Defining the maximum memory size threshold".

Talend Open Studio for Data Quality enables you as well to analyze a set of columns. This type of analysis provides statistics on the values across all the data set (full records). For more information, see Section 6.2, "Analyzing tables in databases".

The sequence of analyzing a column involves the following steps:

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

For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

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

For more information, see Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed". For more information on indicator types and indicator management, see Section 9.2, "Indicators".

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

For more information, see Section 5.3.6, "Using regular expressions and SQL patterns in a column analysis". For more information on pattern types and management, see Section 9.1, "Patterns".

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

5.3.1. Defining the columns to be analyzed and setting indicators

5.3.1.1. How to define the columns to be analyzed

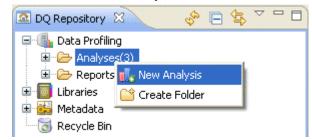
The first step in analyzing the content of one or multiple columns is to define the column(s) to be analyzed. The analysis results provides statistics about the values within each column.

Prerequisite(s): At least one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

To analyze one or more columns, do the following:

Procedure 5.1. Defining the analysis

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



The [Create New Analysis] wizard opens.

😤 Create New Analysis	×
Select a wizard Create a new Analysis	
type filter text Connection Analysis Catalog Analysis Catalog Analysis Column Analysis Column Analysis Column Analysis Column Set Analysis Column Set Analysis Column Correlation Analysis Column Correlation Analysis	
Reck Next > Finish Cancel	

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

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

New Analy your input is	
Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	~
Author	
AUCTOR	
Status	production 🔽
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis
?	< Back Next > Finish Cancel

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



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

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

New Analysis				
Choose Columns I	o analyze:			
Columns:				
🖽 📄 MDM cor	nections			
🗉 😥 🕅 DB conne				
🗄 🗊 FileDelim	ited connections			
?	< Back	Next >	Finish	Cancel

Procedure 5.2. Selecting the column you want to analyze

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



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

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

🚺 Column_Anal	ysis-1 🔀							
: 🖻 🕀 🔛								
Column A	Column Analysis							
🔻 Analysis I	▼ Analysis Metadata							
Set the prope	e properties of analysis.							
Name:	Column_Analysis_1							
Purpose:	Purpose:							
Description:								
Author:								
Status:	Draft 🛛 🗸							
Analyzed	Analyzed Columns							
🕨 Data Filte)ata Filter							
Analysis I	Parameter							
<		>						
Analysis Settings	Analysis Results							

The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

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

 Analyzed Columns 				
Connection: MySQL1	Version:0.1			
Select columns to analyze				
Select indicators for each column				
	Go		₩ \$	⇒ ⇒∥ 1/1
Analyzed Columns	Datamining Type	Pattern	UDI	Operation
🚦 fullname (varchar)	Nominal 💌	<u>-</u>		×
🚦 email (varchar)	Nominal 💌	<u>_</u>		×

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

🗳 Column Selection	
Column Selection	<pre>customer_id(int) scount_num(bigint) Iname(varchar) fname(varchar) sv(varchar) </pre>
customer department department membrane membrane	 mi(varchar) address1(varchar) address2(varchar) address3(varchar) address4(varchar) fullname(varchar)
Table filter: type filter text	Column filter: type filter text
Select All Deselect All	
?	OK Cancel

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

If one of the columns you want to analyze is a primary or a foreign key, its data mining type will automatically become **Nominal** when you list it in the **Analyzed Columns** view. For more information on data mining types in *Talend Open Studio for Data Quality*, see Section 5.2, "Data mining types".

4. If required, right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**.

The selected column will be automatically located under the corresponding connection in the tree view.

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

Ç

If you select to connect to a database that is not supported in *Talend Open Studio for Data Quality* (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on the java engine, see Section 5.3.3, "Using the Java or the SQL engine".

- 6. If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.
- 7. Click the save icon on the toolbar of the analysis editor.

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

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

How to set system indicators

Prerequisite(s): A column analysis is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

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

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

 Analysis I Set the analy 	sis properties.
Name:	Column_Analysis1
Purpose:	
Description:	
Descripcion;	
Author:	talend@talend.com
Status:	development 💌
 Analyzed 	Columns
Connection:	MySQL1 Version:0.1
Connoccion	
	ns to analyze
Select colum	ns to analyze .ors for each column
Select colum Select indical	ns to analyze cors for each column
Select colum	
Select column Select indical	<u>Go</u>
Select column Select indical	<u>Go</u>

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

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

.		All columns	full_name	email	^
	Simple Statistics	✓	Image: A start and a start		
	💉 Row Count	 Image: A set of the set of the	✓	Image: A start of the start	
	💉 Null Count	 Image: A set of the set of the	~	Image: A start of the start	
	💉 Distinct Count	 Image: A set of the set of the	✓	Image: A start and a start	
	💉 Unique Count	 Image: A set of the set of the	~	Image: A start of the start	
	💉 Duplicate Count	 Image: A set of the set of the	~	Image: A start and a start	
	💉 Blank Count	 Image: A set of the set of the	✓	Image: A start of the start	
	💉 Default Value Count	Image: A start of the start			
÷	Text Statistics				
÷	Summary Statistics				
ė	Advanced Statistics		~		
	💉 Mode		~		
	💉 Frequency Table		✓		
	💉 Date Frequency Table				
	💉 Week Frequency Table				
	💉 Month Frequency Table				~

If you are analyzing a very large number of columns, forward and backward arrows are displayed at the bottom of the **[Indicator Selection]** dialog box. Use these arrows to move through different pages and have access to all analyzed columns

	_					
2/7 🗸	k=	4	\Rightarrow	⇒	ОК	Cancel

- 3. Set indicator parameters for the analyzed column(s) as needed and then click **OK** to proceed to the next step.
- 4. Click **OK** to proceed to the next step.

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



9

If you attach the **Data Pattern Frequency Table** to a date column in your analysis, you can generate a date regular expression from the analysis results. For more information, see Section 9.1.4.4, "How to generate a regular expression from the Date Pattern Frequency Table".

 Analyzed Columns 				
Connection: MySQL				
Select columns to analyze				
Select indicators for each column				
Analyzed Columns	Datamining Type	Pattern	UDI	Operation
🖃 🗄 email (varchar)	Nominal 💌	<u> </u>		×
Row Count	and a			×
Null Ca 🔍 View executi	ed query			×
🥂 💉 Distinc 🕂 Add Task				×
🛛 💉 Unique 💥 Remove eler	ments			×
Duplicate Count	39			×
Blank Count	100			X
Mode	1			X
Frequency Table	1			X
Low Frequency Table	131			X
표 🛛 🚦 full_name (varchar)	Nominal 💌	4	<u></u>	×

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

For more information on the Java and the SQL engines, see Section 5.3.3, "Using the Java or the SQL engine".

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

How to set options for system indicators

Prerequisite(s): A column analysis is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

For more information about setting indicators, see the section called "How to set system indicators".

To set options for system indicators, do the following:

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

👻 Indicator 📃 🗖 🔀
Indicator settings Set the parameters for the given indicator
Indicator Thresholds Set here the thresholds expected on the indicator Lower threshold 3 Higher threshold 5
Pinish Cancel

Indicators settings dialog boxes differ according to the parameters specific for each indicator. For more information about different indicator parameters, see Section 9.2.4, "Indicator parameters".

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

How to set user-defined indicators

Prerequisite(s):

- A column analysis is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".
- A user-defined indicator is created in *Talend Open Studio for Data Quality*. For more information, see Section 9.2.3.1, "How to create SQL user-defined indicators".

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

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

1		* 💿						
: 4		-						
Co	lumn A	nalysis						
-	Analysis	Metadata						
		sis properties.						
Ν	lame:	Column_Analysis	;1					
F	urpose:							
0	escription:							
Ą	Author:	talend@talend.c	om					
-	itatus:	development N	•					
_								
•	Analyzed	Columns						
	Connection:	MySQL1	~	Version:0.	1			
	Select colum	ns to analyze						
		ors for each colu	nn					
-								
					Go		I∕ =	¢
		ed Columns	Datamining T	Pattern	UDI C	Operation		
		Iname (varchar)	Nominal 💌	4		×.		
	веп	iail (varchar)		20	A 13	^		

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



You can also click the \checkmark icon next to the column name to which you want to define a userdefined indicator. This will open the **[UDI selector]** dialog box where you can select the user-defined indicators you want to use on the corresponding columns.

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

5.3.2. Finalizing the column analysis before execution

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

Prerequisite(s):

• The column analysis is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

• You have set system or predefined indicators for the column analysis. For more information, see Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed".

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

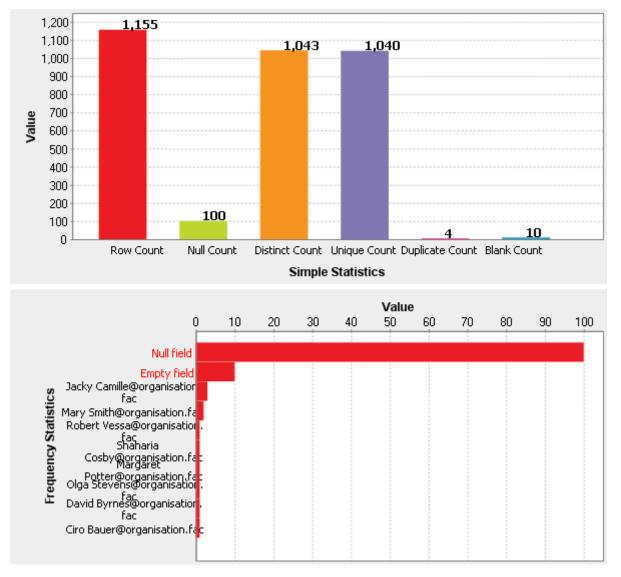
- 1. In the analysis editor, click **Data Filter** to open the corresponding view and filter data through SQL "WHERE" clauses, if required.
- 2. Click **Analysis Parameter** in the analysis editor to open the corresponding view and then select from the **Execution engine** list the engine, Java or SQL, you want to use to execute the analysis.
 - If you select to connect to a database that is not supported in *Talend Open Studio for Data Quality* (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on the java engine, see Section 5.3.3, "Using the Java or the SQL engine".
 - If you select the Java engine, the system will look for Java regular expressions first, if none is found, it looks for SQL regular expressions. For more information on the Java and the SQL engines, see Section 5.3.3, "Using the Java or the SQL engine"

For more information on viewing analyzed data, see Section 5.3.3, "Using the Java or the SQL engine".

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

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

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



🖃 Column: email

9

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

For information on how to access a detailed view of the results of the analysis, see Section 6.2.2.4, "How to access the detailed view of the analysis results".

5.3.3. Using the Java or the SQL engine

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

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

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

Using this engine, you guarantee system better performance. You can also access valid/invalid data in the Data Explorer, for more information, see Section 5.3.5, "Viewing and exporting analyzed data".

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

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

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

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

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

▼ Analysis Parameter
Execution engine: Java 💌
Allow drill down 🔽
Max number of rows kept per indicator: 10

- 2. Select the Allow drill down check box to store locally the data that will be analyzed by the current analysis.
- 3. In the **Max number of rows kept per indicator** field enter the number of the data rows you want to make accessible.



The **Allow drill down** check box is selected by default, and the maximum analyzed data rows to be shown per indicator is set to 50.

You can now run your analysis and then have access to the analyzed data according to the set parameters. For more information, see Section 5.3.5, "Viewing and exporting analyzed data".

5.3.4. Accessing the detailed view of the database column analysis

Prerequisite(s): Talend Open Studio for Data Quality is open. A column analysis is defined and executed.

To access a more detailed view of the analysis results of the procedures outlined in Section 5.3.1, "Defining the columns to be analyzed and setting indicators" and Section 5.3.2, "Finalizing the column analysis before execution", do the following:

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



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

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

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

Label	Count	%		1,200)+	1,15	5										
Row Count	1155.00	100.0.		1,100)+					1,04	3	1.1	040				
Null Count	100.00	8.66%		1,000)					-,-		-,					
Distinct Count	1043.00	90.30		900													
Unique Count	1040.00	90.049															
Duplicate Count	4.00	0.35%		800													
Blank Count	10.00	0.87%	e	700													
	10100	0.01.1	Value	600													
			>	500													
				400													
				300													
				200													
				100				100									
														4	1		
				0	-						_						-
						Row Cou	unt Nu	Il Count	Disti	net Cou	int Un	iaue Co	unt Du	iplicate.	Count	Blank	Ca
						Row Cou	unt Nu	Ill Count	Disti		int Un			iplicate	Count	Blank	Co
						Row Cou	unt Nu	Il Count	Disti		unt Un Iple St			ıplicate	Count	Blank	Co
equency Statisl	tics					Row Cou	unt Nu	ill Count	Disti			atistic	s	iplicate	Count	Blank	Co
· ·	tics	count	%			Row Cou	unt Nu	Il Count 10	Disti 20				s	iplicate 70	Count 80	Blank 90	
value	tics	count	%			Row Cou				Sim	iple St	atistic Valu	s e				
value Null field	tics	100.00	8.66%							Sim	iple St	atistic Valu	s e				
value Null field Empty field		100.00 9.00	8.66% 0.78%	6		N	0 Null field			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga	anisation	100.00 9.00 2.00	8.66% 0.78% 0.17%	6		N Em	0 Null field			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or	nisation	100.00 9.00 2.00 1.00	8.66% 0.78% 0.17% N/A	6	Mary 9	N	0 Null field			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or	nisation rganisat rganisat	100.00 9.00 2.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A	6	Mary 9	N Em	0 Null field pty field anisati			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@organ	nisation rganisat rganisat isation.fac	100.00 9.00 2.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A	6	Mary S	N Em	0 Null field pty field anisati			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@organ Gayle Winfrey@o	nisation rganisat rganisat isation.fac rganisa	100.00 9.00 2.00 1.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A N/A	Statistics	Mary 9	N Em	0 Null field pty field anisati			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@orga Gayle Winfrey@o Janice Vrins@orga	nisation rganisat rganisat isation.fac rganisa anisatio	100.00 9.00 2.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A N/A	Statistics	-	N Em Gmith@org	0 Null field pty field anisati Lo Dav			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@organ Gayle Winfrey@o Janice Vins@orga Ann Evans@orga	nisation rganisat rganisat isation.fac rganisa anisatio nisation	100.00 9.00 2.00 1.00 1.00 1.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A N/A N/A	Statistics	-	N Em	0 Null field pty field anisati Lo Dav			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@orga Gayle Winfrey@o Janice Vrins@orga	nisation rganisat rganisat isation.fac rganisa anisatio nisation	100.00 9.00 2.00 1.00 1.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A N/A	Statistics	-	N Em Gmith@org	0 Null field pty field anisati Lo Dav			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@organ Gayle Winfrey@o Janice Vins@orga Ann Evans@orga	nisation rganisat rganisat isation.fac rganisa anisatio nisation	100.00 9.00 2.00 1.00 1.00 1.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A N/A N/A	Statistics	Amir	N Em Gmith@org Netz@org	0 pty field anisati Lo Dav Gay			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@organ Gayle Winfrey@o Janice Vins@orga Ann Evans@orga	nisation rganisat rganisat isation.fac rganisa anisatio nisation	100.00 9.00 2.00 1.00 1.00 1.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A N/A N/A	quency Statistics	Amir Janice	N Em Smith@org Netz@org Vrins@org	0 Juli field anisati Dav anisati Gay janisati			Sim	iple St	atistic Valu	s e				
value Null field Empty field Mary Smith@orga Lori Anderson@or David Carlson@or Amir Netz@organ Gayle Winfrey@o Janice Vins@orga Ann Evans@orga	nisation rganisat rganisat isation.fac rganisa anisatio nisation	100.00 9.00 2.00 1.00 1.00 1.00 1.00 1.00 1.00	8.66% 0.78% 0.17% N/A N/A N/A N/A N/A	Statistics	Amir Janice	N Em Gmith@org Netz@org	0 Juli field anisati Dav anisati Gay janisati			Sim	iple St	atistic Valu	s e				

If you right-click a data row in the table and select **View rows**, you can access a view of the analyzed data. For more information, see Section 5.3.5, "Viewing and exporting analyzed data".

Below is an example of a Summary Statistics table and graphic for an *age* analyzed column.

Column:age

9

Summary Statistics

Label	Count		40.8	
mean	32.17		39.6	
median	30.00		00.4	
Inter Quartile Range	null		38.4	
lower quartile	30.00		37.2	
upper quartile Range	35.00 12.00		36	
min.value	28.00	Value	34.8	
max.value	40.00	val	33.6	
			32.4	
			31.2	
			30	
			28.8	
			27.6	

After defining the column(s) to be analyzed and setting indicators, you may set options for the attached indicators. For more information, see the following section.

5.3.5. Viewing and exporting analyzed data

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

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

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

✓ Analysis Results
 ✓ Column:email

 Simple Statistics Label Count Row Count 10281.00 Null Count 0.00 Distinct Count 10274.00 Unique Count 10273.00 Duj 1.00 View rows Bla 8.00 🔍 View values 🖧 Remove duplicates

Prerequisite(s): Talend Open Studio for Data Quality is open. A column analysis has been created and executed.

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

- 1. At the bottom of the analysis editor, click the **Analysis Results** tab to open a detailed view of the analysis results.
- 2. Right-click a data row in the statistic results of any of the analyzed columns and select an option as the following:

Option	Operation
View rows	open a view on a list of all data rows in the analyzed column.
View values	open a view on a list of the actual data values of the analyzed column.

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

* 🛱 🕅	i 🗁 🖳 🖉	▽ • gg	i/root		🐱 🔽 Limit	Rows: 100
1/*						
2 Analys	sis: Column	n Analys:	is2			
3 Type o	of Analysis	 s: Multin	ple Colu	nn Anal	vsis	
	se: analyzi					
5 Descri	-	ing a gr	oup or p.	0 00100		
	-					
-	zedElement:					
	ator: Uniqu					
8 Showin	ng: View ro	ວພຣ				
9 */						
	T + FROM S		- ` `		UEDE Sawadi S. TH	CELECT
	CT * FROM `	crm_dem	o`.`cust	omer` 🕻	HERE `email` IN	(SELECT
	CT * FROM `	crm_dem	o`.`cust	omer` G	HERE `email` IN	(SELECT
10 SELEC	CT * FROM ` Column_An] {	_		omer` l		(SELECT
10 SELEC		 ⊠Message		omer` G		(SELECT
10 SELEC	Column_An] {	 ⊠Message	is		address1	•
10 SELEC	Column_An] { account_num	— X Message Iname	s fname	mi	Ш	address2
10 SELEC	Column_An] { account_num 87462024688	_ ⊠ Message Iname Nowmer	s fname	mi A.	address1 2433 Bailey Road	address2
10 SELEC Analysis: customer_id 1 2	Column_An] 2 account_num 87462024688 87470586299	Message Iname Nowmer Whelply	s fname	mi A. I.	address1 2433 Bailey Road 2219 Dewing Avenue	address2 <null> <null></null></null>
10 SELEC / [/* Analysis: customer_id 1 2 3	Column_An] 2 account_num 87462024688 87470586299 87475757600	Message Iname Nowmer Whelply Derry	s fname	mi A. I. <null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave.	address2 <null> <null> <null></null></null></null>
10 SELEC I/* Analysis: customer_id 1 2 3 4 5 6	Column_An] 2 account_num 87462024688 87470586299 87475757600 87500482201	Message Iname Nowmer Whelply Derry Spence Gutierrez Damstra	rs fname Sheri	mi A. I. <null> J. <null> F.</null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way 8668 Via Neruda 1619 Stillman Court	address2 <null> </null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null>
10 SELEC [/* Analysis: customer_id 1 2 3 4 5 6 7	Column_An] S account_num 87462024688 87470586299 87475757600 87500482201 87514054179 87517782449 87521172800	Message Iname Nowmer Whelply Derry Spence Gutierrez Damstra Kanagaki	s fname Sheri Rebecca	mi A. I. <null> J. <null></null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way 8668 Via Neruda 1619 Stillman Court 2860 D Mt. Hood Circle	address2 <null> <</null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null></null>
10 SELEC I [/* Analysis: customer_id 1 2 3 4 5 6 7 8	Column_An] 2 account_num 87462024688 87470586299 87475757600 87500482201 87514054179 87517782449 87521172800 87539744377	Message Iname Nowmer Whelply Derry Spence Gutierrez Damstra Kanagaki Brunner	s fname Sheri Rebecca Kim	mi A. I. <null> J. <null> F.</null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way 8668 Via Neruda 1619 Stillman Court 2860 D Mt. Hood Circle 6064 Brodia Court	<pre>address2 <pre>cnull> <pre>cnull> <pre>cnull> <pre>cnull> <pre>cnull> <pre>cnull> <pre>cnull> <pre>cnull> <pre>cnull></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
10 SELEC Customer_id 1 2 3 4 5 6 7	Column_An] S account_num 87462024688 87470586299 87475757600 87500482201 87514054179 87517782449 87521172800	Message Iname Nowmer Whelply Derry Spence Gutierrez Damstra Kanagaki	s fname Sheri Rebecca	mi A. I. <null> J. <null> F.</null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way 8668 Via Neruda 1619 Stillman Court 2860 D Mt. Hood Circle	address2 anul>

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

When using the Java engine, the view opens in the **Profiling** perspective listing the number of the analyzed data rows you set in the **Analysis parameters** view of the analysis editor. For more information, see Section 5.3.3, "Using the Java or the SQL engine".

Column_Ana	alysis2 0.1	👆 View rows	8			
4						
customer_id	account_num	Iname	fname	mi	address1	address2
1	87462024688	Nowmer	Sheri	Α.	2433 Bailey Road	(null)
2	87470586299	Whelply		Ι.	2219 Dewing Avenue	(null)
3	87475757600	Derry		(null)	7640 First Ave.	(null)
4	87500482201	Spence		J.	337 Tosca Way	(null)
5	87514054179	Gutierrez		(null)	8668 Via Neruda	(null)
6	87517782449	Damstra		F.	1619 Stillman Court	(null)
7	87521172800	Kanagaki	Rebecca	(null)	2860 D Mt. Hood Circle	(null)
8	87539744377	Brunner	Kim		6064 Brodia Court	(null)
9	87544797658	Blumberg	Brenda		7560 Trees Drive	(null)
10	87568712234	Stanz	Darren		1019 Kenwal Rd.	(null)
11	87572821378	Murraiin	Jonathan		5423 Camby Rd.	(null)
12	87579237222	Creek	Jewel		1792 Belmont Rd.	(null)
13	87587122917	Medina	Peggy		3796 Keller Ridge	(null)
14	87592626810	Rutledge	Bryan		3074 Ardith Drive	(null)

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

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

A dialog box opens.

"	×
CSV export options Please specify a filename	N
Format Character set: UTF-8 Delimiter: ; Null value: <	
Destination	Choose OK Cancel

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

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

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

Talend Open Studio for Data Quality allows you to use regular expressions or SQL patterns in column analyses. These expressions and patterns will help you define the content, structure and quality of the data included in the analyzed columns.

For more information on regular expressions and SQL patterns, see Section 1.2.3.2, "Patterns and indicators" and Chapter 6, *Table analyses*.

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

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



If the database you are using does not support regular expressions or if the query template is not defined in Talend Open Studio for Data Quality, you need first to declare the user defined function and define the query template before being able to add any of the specified patterns to the column analysis. For more information, see Section 9.1.2, "Managing User-Defined Functions in databases".

Prerequisite(s): *Talend Open Studio for Data Quality* is open. A column analysis is open in the analysis editor.

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

- 1. Follow the steps outlined in Section 5.3.1.1, "How to define the columns to be analyzed" to create a column analysis.
- 2. In the open analysis editor, click Analyze Columns to open the analyzed columns view.

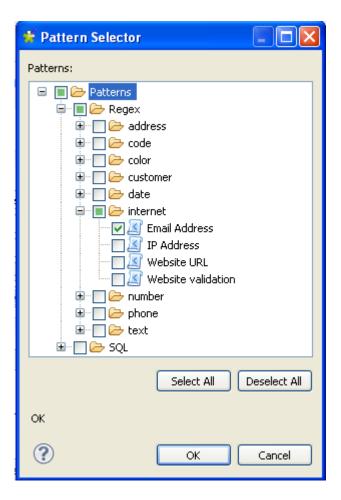
 Analyzed Columns 					
Connection: MySQL1	Version:0.1				
Select columns to analyze					
Select indicators for each column					
	Go		¢ ∳	⇒ ⇒∥ 1/1	
Analyzed Columns	Datamining Type	Pattern	UDI	Operation	
🚦 fullname (varchar)	Nominal 💌	<u></u>		×	
🚦 email (varchar)	Nominal 💌	<u> </u>		×	

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

Click the $\boxed{3}$ icon next to the column name to which you want to add a regular expression or an SQL pattern.

The [Pattern Selector] dialog box opens.

3.



- 4. Expand **Patterns** and browse to the regular expression or/and the SQL patterns you want to add to the column analysis.
- 5. Select the check box(es) of the expression(s) or pattern(s) you want to add to the selected column.
- 6. Click **OK** to proceed to the next step.

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

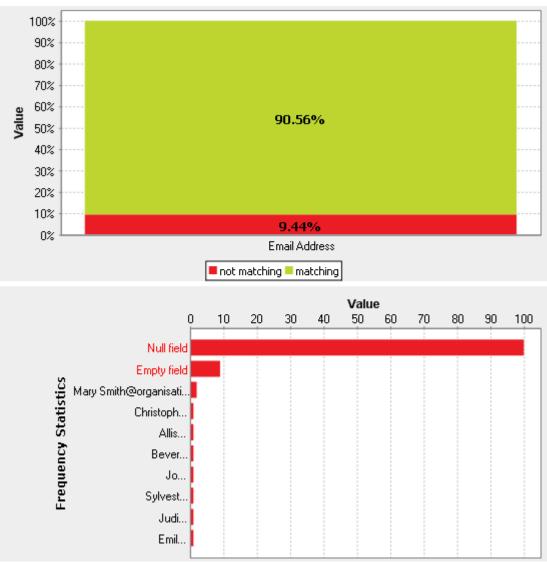


You can add a regular expression or an SQL pattern to a column simply by a drag and drop operation from the **DQ Repository** tree view onto the analyzed column.

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

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

- 🛨 Column: full_name
- 🖃 Column: email



5.3.6.2. How to edit a pattern in the column analysis

Prerequisite(s): Talend Open Studio for Data Quality is open. A column analysis is open in the analysis editor.

To edit a pattern added to an analyzed column:

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

▼ Analyzed Columns				
Connection: MySQL				
Select columns to analyze				
Select indicators for each column				
Analyzed Columns	Datamining Type	Patt	UDI	Operation
INITIAL (varchar)		R	601	
🗐 🚦 COMPANY (varchar)	Nominal 💌	Ā		×
Row Count	1			×
Null Count	- 494			****
🦳 💉 Distinct Count				×
Unique Count	100			×
Duplicate Count	100			×
Blank Count	100			X
BE Code postal	ed query	5		- Č
		4	A.5	Č I
ADDRESS_S1 Si Edit Pattern Em CITY (varcha Add Task	1		<u></u>	- C
	2			- C
Remove eler	ments	<u> </u>	<u> </u>	^

The pattern editor opens showing the selected pattern metadata.

BE Code pos	tal 0.1 🛛 🖓 🖓 🖓
: 🖻 🕀 🔛	
Pattern S	ettings
💌 Pattern M	letadata
Set the prope	rties of pattern.
Name:	BE Code postal
Purpose:	Check the validity of Belgian postal codes.
	Matches standard Belgian postal codes.
Description:	
Author:	
Status:	Draft 🛛 🗸
💌 Pattern D	efinition
Type in the da select "ALL_D/	tabase-specific pattern definition. If the expression is simple enough to be used in all databases, ATABASE_TYPE" type in the list.
Default	✓ 「^(F-[0-9]{4,5} B-[0-9]{4})\$」 Test
	+

- 3. In the pattern editor, click **Pattern Definition** to edit the pattern definition, or change the selected database, or add other patterns specific to available databases using the [+] button.
- 4. On the toolbar, click the save icon to save your changes.
 - If the regular pattern is simple enough to be used in all databases, select Default in the list.

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

5.3.6.3. How to view the data analyzed against patterns

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

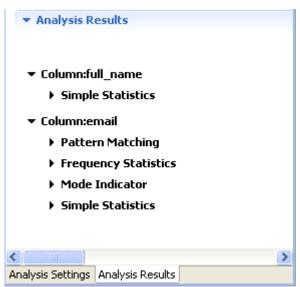


When you use the Java engine to run the analysis, the view of the actual data will open in the **Profiling** perspective. While if you use the SQL engine to execute the analysis, the view of the actual data will open in the **Data Explorer** perspective.

Prerequisite(s): *Talend Open Studio for Data Quality* is open. A column analysis that uses patterns has been created and executed.

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

- 1. Follow the steps outlined in Section 5.3.1.1, "How to define the columns to be analyzed" and Section 5.3.6.1, "How to add a regular expression or an SQL pattern to a column analysis" to create a column analysis that uses a pattern.
- 2. Execute the column analysis.
- 3. In the analysis editor, click the **Analysis Results** tab at the bottom of the editor to open the corresponding view.





The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

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

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

- ▼ Column:email
 - Pattern Matching %Match %No Match #Match Label #No 100% Email Add 9.44% 1046.0 109.0 90% 💫 view invalid values 80% 🔍 view invalid rows 👢 view valid values 70% view valid rows 60% Value 90.56% Generate jobs 50% 40% 30% 20% 10% 9.44% 0% Email Address not matching matching
- 5. Right-click the pattern line in the **Pattern Matching** table and select:

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

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

		and a de core	or (gg.Email_A	Address).sc		
: X 47 47		▽ • gg	/root		🖌 🗹 Lim	it Rows: 100
1/*						
2 Analys	sis: Column	h Analys:	is2			
	of Analysis			nn Anal	vsis	
	se: analyzi					
	_	ing a gro	oup or pr	o corun	115	
5 Descri	-					
6 Analyz	zedElement:	email				
7 Indica	ator: Email	l_Addres:	3			
8 Showir	ng: View va	alid row:	3			
9 */						
10 SELECT						
		'crm demo	o'.'custo	nmer`	WHERE `email``	REGEXP '[a-z]-ZO-9, %-
<	I - FROM	crm dem	o'.'custo	omer`	WHERE `email` :	REGEXP '[a-zA-ZO-9. %-
<				omer`	WHERE `email`	REGEXP '[a-zA-ZO-9. %-
<	Column_An]			omer`	WHERE `email` :	REGEXP '[a-z1-20-9. %-
<	Column_An]			omer`	WHERE `email` ` address1	REGEXP '[a-zA-20-9. %-
1 [/* Analysis:	Column_An]	X Message	s			•
I [/* Analysis: customer_id	Column_An] { account_num	Message	s fname	mi	address1	email
I [/* Analysis: customer_id	Column_An] 2 account_num 87462024688	Message	s fname	mi A.	address1 2433 Bailey Road	email SheriNowmer@Tlaxiaco.org
I [/* Analysis: customer_id 1 2	Column_An] account_num 87462024688 87470586299	X Message Iname Nowmer Whelply	s fname	mi A. I.	address1 2433 Bailey Road 2219 Dewing Avenue	email SheriNowmer@Tlaxiaco.org DerrickWhelply@Sooke.org
1 [/* Analysis: customer_id 1 2 3	Column_An] account_num 87462024688 87470586299 87475757600	Message Iname Nowmer Whelply Derry	s fname	mi A. I. <null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave.	email SheriNowmer@Tlaxiaco.org DerrickWhelply@Sooke.org JeanneDerry@Issaquah.org
I [/* Analysis: customer_id 1 2 3 4	Column_An] account_num 87462024688 87470586299 87475757600 87500482201	Message Iname Nowmer Whelply Derry Spence	s fname	mi A. I. <null> J.</null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way	email SheriNowmer@Tlaxiaco.org DerrickWhelply@Sooke.org JeanneDerry@Issaquah.org MichaelSpence@Burnaby.org
1 [/* Analysis: customer_id 1 2 3 4 5	Column_An] S account_num 87462024688 87470586299 87475757600 87500482201 87514054179	Message Iname Nowmer Whelply Derry Spence Gutierrez Damstra	s fname Sheri Rebecca	mi A. I. <null> J. <null></null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way 8668 Via Neruda 1619 Stillman Court 2860 D Mt. Hood Circle	email SheriNowmer@Tlaxiaco.org DerrickWhelply@Sooke.org JeanneDerry@Issaquah.org MichaelSpence@Burnaby.org MayaGutierrez@Novato.org RobertDamstra@Lynnwood.org RebeccaKanagaki@Tlaxiaco.org
<pre>1 [/* Analysis: customer_id 1 2 3 4 5 6 7 9</pre>	Column_An] S account_num 87462024688 87470586299 87475757600 87500482201 87514054179 87517782449 87521172800 87544797658	Message Iname Nowmer Whelply Derry Spence Gutierrez Damstra Kanagaki Blumberg	fname Sheri	mi A. I. <null> J. <null> F. <null></null></null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way 8668 Via Neruda 1619 Stillman Court 2860 D Mt. Hood Circle 7560 Trees Drive	email SheriNowmer@Tlaxiaco.org DerrickWhelply@Sooke.org JeanneDerry@Issaquah.org MichaelSpence@Burnaby.org MayaGutierrez@Novato.org RobertDamstra@Lynnwood.org RebeccaKanagaki@Tlaxiaco.org BrendaBlumberg@Richmond.org
1 [/* Analysis: customer_id 1 2 3 4 5 6 7	Column_An] S account_num 87462024688 87470586299 87475757600 87500482201 87514054179 87517782449 87521172800	Message Iname Nowmer Whelply Derry Spence Gutierrez Damstra Kanagaki	s fname Sheri Rebecca	mi A. I. <null> J. <null> F.</null></null>	address1 2433 Bailey Road 2219 Dewing Avenue 7640 First Ave. 337 Tosca Way 8668 Via Neruda 1619 Stillman Court 2860 D Mt. Hood Circle	email SheriNowmer@Tlaxiaco.org DerrickWhelply@Sooke.org JeanneDerry@Issaquah.org MichaelSpence@Burnaby.org MayaGutierrez@Novato.org RobertDamstra@Lynnwood.org RebeccaKanagaki@Tlaxiaco.org

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

When using the Java engine, the view opens in the **Profiling** perspective listing the number of valid/invalid data according to the row limit you set in the **Analysis parameters** view of the analysis editor. For more information, see Section 5.3.3, "Using the Java or the SQL engine".

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

You can save the executed query and list it under the **Libraries** > **Source Files** folders in the **DQ Repository** tree view if you click the save icon on the SQL editor toolbar. For more information, see Section 5.3.7, "Saving the queries executed on indicators".

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

5.3.7. Saving the queries executed on indicators

Talend Open Studio for Data Quality enables you to view, in the Data Explorer, the queries executed on different indicators used in an analysis. From the Data Explorer, you will be able to save the query and list it under the **Libraries > Source Files folders** in the **DQ Repository** tree view.

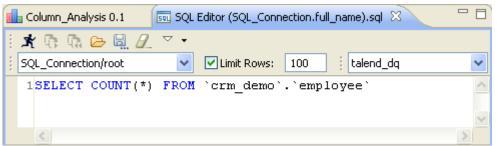
Prerequisite(s): *Talend Open Studio for Data Quality* is open. At least one analysis with indicators has been created.

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

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

 Analyzed Columns 				
Connection: MySQL				
Select columns to analyze				
Select indicators for each column				
Analyzed Columns	Datamining Type	Pattern	UDI	Operation
🖃 🗄 email (varchar)	Nominal 💌	<u> </u>		×
Row Count	140			×
Null Cc 🔍 View execute	ed query			×
🚽 🖉 Distinc 🕂 Add Task				×
🖉 🖉 Unique 💥 Remove elen	nents			×
Duplicace counc	- m			×
Blank Count	100			×
Mode	100			×
Frequency Table	100			*****
🧹 Low Frequency Table	100			×
🗄 📲 🗄 full_name (varchar)	Nominal 💌	<u>_</u>	<u></u>	×

2. Select **View executed query** to open the Data Explorer on the query executed on the selected indicator.



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

¥ Select folder 📃 🗖 🔀
Select the folder in which the item will be created
 TDQ_Libraries Source Files SQL_Queries_global SQL_Queries_on_analyses
Name: Row_Count.sql
ок
OK Cancel

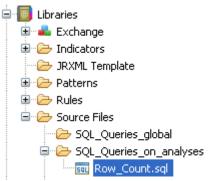
4. Select the **Source Files** folder or any sub-folder under it and enter in the **Name** field a name for the open query.



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

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

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



5.3.8. Creating table and columns analyses in shortcut procedures

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

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

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

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

Item	То	
Table analysis	analyze the selected table using SQL business rules.	
	For more information on the Simple Statistics indicators, see Chapter 6, <i>Table analyses</i> .	
Column analysis	analyze all the columns included in the selected table using the Simple Statistics indicators.For more information on the Simple Statistics indicators, see Section 9.2.1.1, "Simple statistics".	
Pattern frequency analysis	analyze all the columns included in the selected table using the Pattern Frequency Statistics along with the Row Count and the Null Count indicators.For more information on the Pattern Frequency Statistics, see Section 9.2.1.5, "Pattern frequency statistics".	

The above steps replace the procedures outlined in Section 5.3.1, "Defining the columns to be analyzed and setting indicators". Now, you proceed following the steps outlined in Section 5.3.2, "Finalizing the column analysis before execution".

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

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

Item	То	
Analyze	create an analysis for the selected column	
	you must later set the indicators you want to use to analyze the selected column.	
	For more information on setting indicators, see Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed". For more information on accomplishing the column analysis, see Section 5.3.2, "Finalizing the column analysis before execution".	
Analyze correlation	perform column correlation analyses between nominal and interval columns or nominal and date columns in database tables.	
	For more information, see Chapter 6, Table analyses.	
Nominal value analysis	analyze minimal correlations between nominal columns in the same table and gives the result in a chart.	
	For more information, see Section 8.4, "Nominal correlation analysis".	
Simple analysis	analyze the selected column using the Simple Statistics indicators.	
	For more information on the Simple Statistics indicators, see Section 9.2.1.1, "Simple statistics".	
Pattern frequency analysis	analyze the selected column using the Pattern Frequency Statistics along with the Row Count and the Null Count indicators.	
	For more information on the Pattern Frequency Statistics, see Section 9.2.1.5, "Pattern frequency statistics".	

The above steps replace one of or both of the procedures outlined in Section 5.3.1, "Defining the columns to be analyzed and setting indicators". Now, you proceed following the same steps outlined in Section 5.3.2, "Finalizing the column analysis before execution".

5.4. Analyzing master data on an MDM server

Talend Open Studio for Data Quality enables you to analyze master data in one or multiple data containers on the MDM server and execute the created analyses using the SQL or Java engines. For further information on these engines, see Section 5.3.3, "Using the Java or the SQL engine".

Talend Open Studio for Data Quality enables you as well to analyze a set of columns, for more information, see Section 6.4, "Analyzing tables on MDM servers".

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

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

1. Defining the business entity to be analyzed.

For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

2. Settings predefined system indicators for the business entity.

For more information, see Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed". For more information on indicator types and indicator management, see Section 9.2, "Indicators".



You can also use Java user-defined indicators when analyzing master data on the condition that a Java user-defined indicator is already created. For further information, see Section 9.2.3.2, "How to define Java user-defined indicators".

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

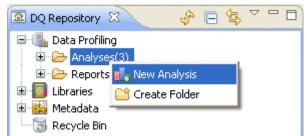
5.4.1.1. How to define the business entities to be analyzed

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

Prerequisite(s): At least one MDM connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.3, "Connecting to an MDM server".

Procedure 5.3. Defining the analysis

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



The [Create New Analysis] wizard opens.

📽 Create New Analysis	
Select a wizard	
Create a new Analysis	
type filter text	
Connection Analysis Catalog Analysis Catalog Analysis Schema Analysis Column Analysis Column Analysis Column Analysis Column Set Analysis Redundancy Analysis Column Correlation Analysis Column Correlation Analysis	
(?) < Back Next > Finish	Cancel

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

Name	MDM_Analysis				
Purpose	Analyzing master data				
Description	Analyzing master data on an MDM server to provide simple statistics including the number of data rows (data record), the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields.				
Author					
Status	development				
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyses Select				
Туре	Multiple Column Analysis				

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

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

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

New Analys Choose a Colu	is umns to analyze			
	connections IDM_Connection Product onnections elimited connections			
(?)	< Back	Next >	Finish	Cancel

Procedure 5.4. Selecting the business entity you want to analyze

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

New Analysis Choose a Columns to analyze	
Columns: MDM connections MDM_Connection Product Product Availability(boolean) Description(string) Family(string) Features Id(string) Name(string) Name(string)	
Rext Next > Finish Cancel	el

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

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

🔓 *MDM_Analy	sis 0.1 🕱	
	大 ③	
Column A	nalysis	
Analysis I Set the prope	Metadata erties of analysis.	
Name:	MDM_Analysis	
Purpose:	Analyzing master data	
Description:	Analyzing master data on an MDM server to provide simple statistics including the number of data rows (data record), the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields.	
Author:		Ī
Status:	development	
▼ Analyzed	Columns	
Connection:	Local_Server 🖌	
Select column	ns to analyze	
Select indicat	tors for each column	
	Analyzed Columns Datamining Type Pattern UDI Operation	
O Av	railability (boolean) Other 💌 🧟 💉 🗙	
🙂 Fa	mily (string) Other 💌 🔏 🗶	

- The display of the connection editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".
- 3. Click the Analyzed Column tab to open the corresponding view, if not already open.

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

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

🗳 Column Selection	
Column Selection	
Product Product Product ProductFamily	 Picture(PICTURE) Id(string) Name(string) Description(string) Availability(boolean) Price(decimal) Family(string) OnlineStore(URL)
Table filter: type filter text	Column filter: type filter text
Select All Deselect All	
?	OK Cancel

- Click the business entity name to display all its record in the right-hand panel of the [Column Selection] 5. dialog box.
- 6. In the list to the right, select the check boxes of the column(s) you want to analyze and click OK to proceed to the next step.

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

analysis editor.

ect columns to analyze			
ect indicators for each column			
∃ ⊞			
Analyzed Columns	Datamining Type	Pattern	UDI
🗄 🧿 Family (string)	Other 💌	<u>R</u>	
🗄 🟮 Availability (boolean)	Other 💌	R	
Availability (boolean)			

You can drag the records to be analyzed directly from the DQ Repository tree view to the column

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



The data mining type is set to **Other** by default. For more information on data mining types in *Talend* Open Studio for Data Quality, see Section 5.2, "Data mining types".



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

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

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

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



You can also use Java user-defined indicators when analyzing master data on the condition that a Java user-defined indicator is already created. For further information, see Section 9.2.3.2, "How to define Java user-defined indicators".

Prerequisite(s): An analysis of a business entity is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

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

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

	Indicators	All columns	Family	Availability
.	Simple Statistics	V	Image: A start of the start	 Image: A start of the start of
	💉 Row Count	V	~	~
	💉 Null Count	V	~	~
	💉 Distinct Count	V	~	~
	💉 Unique Count	V	~	~
	💉 Duplicate Count		~	~
	💉 Blank Count			
	Default Value Count			
÷	Text Statistics			
÷	Summary Statistics			
÷	Advanced Statistics			
<u>+</u>	Pattern Frequency Statistics			
÷	Soundex Frequency Statistics			
	se: analyze the quantity of records ption: contain several count indicato	ors		

3. Select the simple statistics check boxes for the MDM records and then click **OK** to proceed to the next step.

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

 Analyzed Columns 					
Connection: MDM_Connection	~				
Select columns to analyze					
Select indicators for each column					
E E					
Analyzed Columns	Datamining Type	Pattern	UDI	Operation	1
□	Other 💌			X	
🗄 💉 Row Count			~	×	
📈 Null Count				×	
🧹 Distinct Count				×	
📈 Unique Count				×	
💉 Duplicate Count	-			×	
🖃 🟮 Availability (boolean)	Other 💌	<u>_</u>		×	
💉 Row Count				×	
💉 Null Count	100			×	
🧖 Distinct Count	100			×	
📈 Unique Count	-			×	
💉 Duplicate Count	-			×	

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

5.4.1.3. How to set options for system indicators

Prerequisite(s): An analysis of MDM records is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1, "Defining the columns to be analyzed and setting indicators".

To set options for system indicators, do the following:

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

👻 Indicator 📃 🗖 🔀
Indicator settings Set the parameters for the given indicator
Indicator Thresholds Set here the thresholds expected on the indicator Lower threshold Higher threshold 5
Pinish Cancel

Running the analysis will show if these thresholds are violated through appending a warning icon on such a result and the result itself will be in red. For further information, see Section 6.2.2.4, "How to access the detailed view of the analysis results".

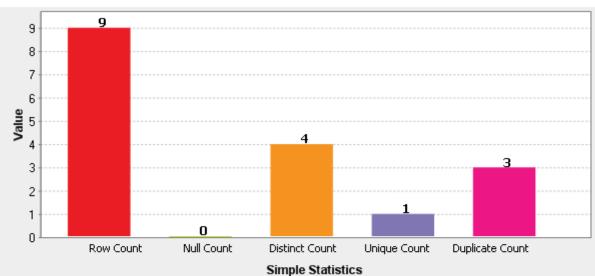
Indicators settings dialog boxes differ according to the parameters specific for each indicator. For more information about different indicator parameters, see Section 9.2.4, "Indicator parameters".

- 3. Set the parameters for the given indicator.
- 4. Click **Finish** to close the dialog box.
- 5. In the analysis editor, click the **Data Filter** tab to display the corresponding view and filter master data through XQuery clauses, if required.
- 6. In the analysis editor, click the **Analysis Parameters** tab to display the corresponding view and select the engine you want to use to run the analysis. For more information on available engines, see Section 5.3.3, "Using the Java or the SQL engine".
- 7. Click the save icon on the toolbar of the analysis editor and then press F6 to execute the analysis.

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

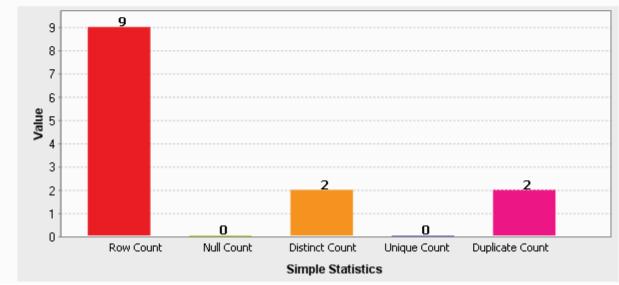


To view the different graphics associated with all analyzed records, you may need to navigate through the different pages in the **Graphics** panel using the toolbar on the upper-right corner.



🖃 Column: Family



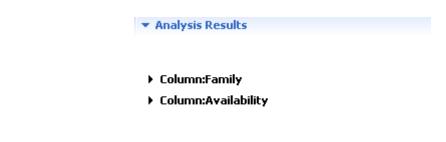


5.4.2. Accessing the detailed view of the master data analysis

Prerequisite(s): An analysis of a business entity is defined and executed in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1, "Defining the columns to be analyzed and setting indicators".

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

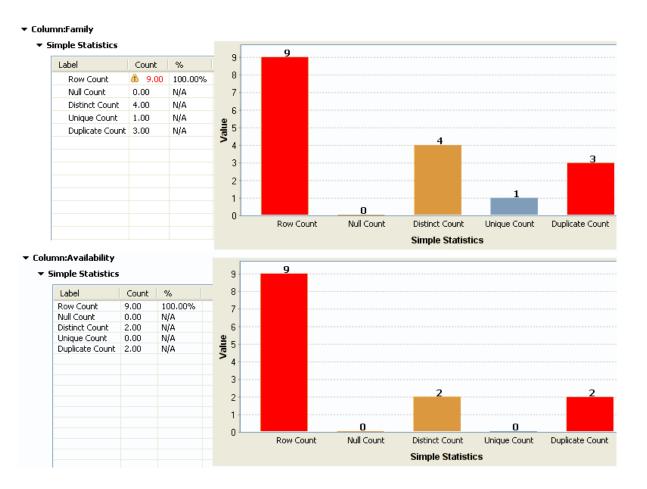
- 1. Click the Analysis Results tab at the bottom of the analysis editor to open the corresponding view.
- 2. Click **Analysis Results** and then the name of the analyzed column for which you want to display the detailed results.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

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

Below are the tables that accompany the Simple Statistics graphics in the **Analysis Results** view for the analyzed records in the procedure outlined in Section 5.3.1, "Defining the columns to be analyzed and setting indicators".



5.4.3. Analyzing master data in shortcut procedures

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

For further information, see Section 5.3.8, "Creating table and columns analyses in shortcut procedures".

5.5. Analyzing data in a file

Talend Open Studio for Data Quality enables you to create a column analysis on a delimited file and execute the created analyses using the Java engine.

Talend Open Studio for Data Quality enables you as well to analyze a set of columns, for more information, see Section 6.3, "Analyzing tables in delimited files".

5.5.1. Analyzing columns in a delimited file

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

1. defining the columns to be analyzed.

For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

2. settings predefined system indicators for the defined columns.

For more information, see Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed". For more information on indicator types and indicator management, see Section 9.2, "Indicators".

3. setting patterns for the defined columns. For more information, see Section 9.1, "Patterns".

You can also use Java user-defined indicators when analyzing columns in a delimited file on the condition that a Java user-defined indicator is already created. For further information, see Section 9.2.3.2, "How to define Java user-defined indicators".

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

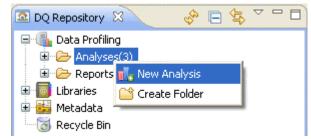
5.5.1.1. How to define the columns to be analyzed

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

Prerequisite(s): At least one connection to a delimited file is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.2.1, "How to connect to a delimited file".

Procedure 5.5. Defining the analysis

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



The [Create New Analysis] wizard opens.

📽 Create New Analysis	
Select a wizard Create a new Analysis	
type filter text Connection Analysis Catalog Analysis Catalog Analysis Column Analysis Column Analysis Column Analysis Column Set Analysis Column Set Analysis Column Correlation Analysis Column Correlation Analysis	
Rext > Finish	Cancel

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

Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	~
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis



You can directly get to this step in the analysis creation wizard if you right-click the column to analyze in **Metadata > FileDelimited** and select **Column Analysis > Analyze**. For further information, see Section 5.3.8, "Creating table and columns analyses in shortcut procedures".

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

DB connections Image: Second secon	Columns	
Delimited_File_Connection Metadata Metadata Columns firstname I lastname i lastname i city i lage MDMconnections	1 1	
Metadata Columns Columns Firstname I astname City City City City MDMconnections		
Columns Columns Columns Firstname Columns Firstname Columns Firstname Columns Column		
firstname i I lastname i city i gg MDMconnections		
		🖮 🗝 🚦 📊
city if city if age MDMconnections		📺 🖷 🚦 firstname
MDMconnections		
MDMconnections		
	±=	MDMconnections

Procedure 5.6. Selecting the columns in the delimited file

- 1. Expand FileDelimited and then browse to the columns you want to analyze.
- 2. Select these columns and then click **Finish** to close the wizard.

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

💼 *Delimited_Fi	ile_Analysis 0.1 🛛	- 8
	• 2	
Column A	nalysis	
💌 Analysis I	Metadata	▼ Graphics
Set the prope	rties of analysis.	
Name:	Delimited_File_Analysis	ĒĒ
Purpose:	Analyzing data in a delimited file	Refresh the
Description:	analyzing columns in a delimited file to provide different statistics on these columns	Concontate.
Author:	user@company.com	
Status:	development 💌	
Analyzed	Columns	
🕨 Data Filte	ir -	
Analysis I	Parameter	
Analysis Settings	Analysis Results	>

The display of the connection editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

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

Analyzed Columns				
Connection: Delimited_File	~			
elect columns to analyze				
elect indicators for each column				
elect indicators for each column				
	1	1		-
	Datamining Type	Pattern	UDI	0
	Datamining Type	Pattern		0
Analyzed Columns		Pattern		0 X X



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

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

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

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

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



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

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

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

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



You can also use Java user-defined indicators when analyzing columns in a delimited file on the condition that a Java user-defined indicator is already created. For further information, see Section 9.2.3.2, "How to define Java user-defined indicators".

Prerequisite(s): An analysis of a delimited file is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed".

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

- 1. Follow the procedure outlined in Section 5.3.1.1, "How to define the columns to be analyzed".
- 2. In the analysis editor, click Analyzed Columns to open the analyzed columns view.
- 3. Click Select indicators for each column to open the [Indicator Selection] dialog box.

1	i Indi	cator Selection					×
		Indicators Simple Statistics Text Statistics Summary Statistics Advanced Statistics Pattern Frequency Statistics Soundex Frequency Statistics Soundex Frequency Table Soundex Low Frequency Table	All columns	id	firstname		
	65/26/26/26	e: analyze the quantity of records otion: contain several count indicate	ors		ок	Cancel	

4. Select the check boxes for the indicators you want to use on the columns to be analyzed and then click **OK** to proceed to the next step.

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

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

Analyzed Columns				
Connection: Delimited_File				
elect columns to analyze				
elect indicators for each column				
Analyzed Columns	Datamining Type	Pattern	UDI	0
🗄 🚦 id (Integer)	Interval 💌	8		×
 firstname (String) Row Count Null Count Distinct Count Unique Count Duplicate Count Blank Count Text Statistics Soundex Frequency Table Start with upper case (s 				****
C (-	Interval 💌	<2	<u> </u>	× ×

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

5.5.1.3. How to set options for system indicators

Prerequisite(s): An analysis of a delimited file is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed", Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed".

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

- 1. Follow the procedures outlined in Section 5.3.1.1, "How to define the columns to be analyzed" and Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed".
- 2. In the analysis editor, click Analyzed Columns to open the analyzed columns view.
- 3. In the **Analyzed Columns** list, click the option icon a next to the indicator to open the dialog box where you can set options for the given indicator.



Indicators settings dialog boxes differ according to the parameters specific for each indicator. For more information about different indicator parameters, see Section 9.2.4, "Indicator parameters".

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

5.5.1.4. How to set regular expressions and finalize the analysis

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

Prerequisite(s): An analysis of a delimited file is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see Section 5.3.1.1, "How to define the columns to be analyzed", Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed" and Section 5.5.1.3, "How to set options for system indicators".

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

1. Define the regular expression you want to add to the analyzed column. For further information on creating regular expressions, see Section 9.1.4.1, "How to create a new regular expression or SQL pattern".

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

🔄 Start with up	per case (single word) 0.1 🖾 📃 🗖
: 🖻 🕀 🔛	
Pattern S	ettings
💌 Pattern N	
Set the prope	erties of pattern.
Name:	Start with upper case (single word)
Purpose:	identifies words starting with upper case
Description:	will match when the first character of a word is uppercased. For example, "Axel", Street" will match. But "13", "JOHN DOE", "3RD FLOOR", and "stree" will not match
Author:	talend@talend.com
Status:	development 💌
	Definition stabase-specific pattern definition. If the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list. Image: the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list. Image: the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list. Image: the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list. Image: the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list. Image: the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list. Image: the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list.

2. Add the regular expression to the analyzed column in the open analysis editor, the *firstname* column in this example. For further information, see Section 5.3.6.1, "How to add a regular expression or an SQL pattern to a column analysis".

		ion: Delimited_File				
		<u>lumns to analyze</u> dicators for each column				
<u></u>						
	Ŧ					
		Analyzed Columns	Datamining Type	Pattern	UDI	
Ŧ	B	id (Integer)	Interval 💌	<u></u>		>
	Ð	Soundex Frequency Table	Nominal	25		
Ŧ		当 Start with upper case (sing age (Integer)	Interval 💌	B	*	5

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



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

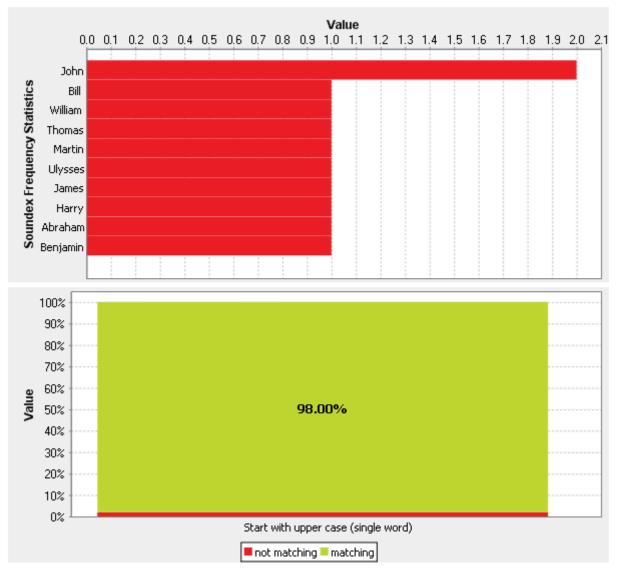
The **Graphics** panel to the right of the analysis editor displays a group of graphic(s), each corresponding to one of the analyzed columns.

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

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

🛨 Column: id

🖃 Column: firstname



In order to view detail results of the analyzed columns, see Section 6.2.2.4, "How to access the detailed view of the analysis results".

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

Prerequisite(s): An analysis of a delimited file is defined and executed in *Talend Open Studio for Data Quality*. For more information, see Section 5.5.1, "Analyzing columns in a delimited file".

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

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



- Column:id
- Column:firstname
- Column:age



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

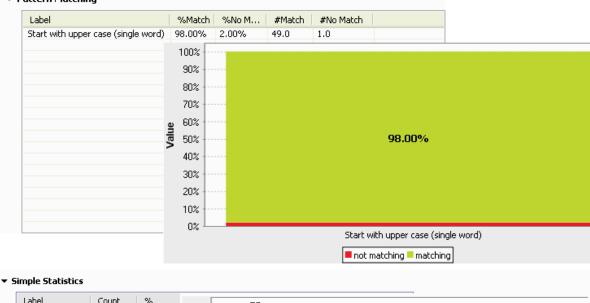
The detailed analysis results view shows the generated graphics for the analyzed columns accompanied with tables that detail the statistic results.

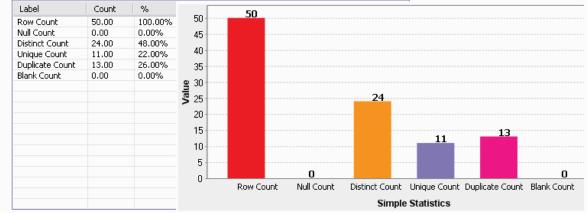
Below are the tables that accompany the statistics graphics in the **Analysis Results** view for the analyzed *firstname* column in the procedure outlined in Section 5.5.1, "Analyzing columns in a delimited file".

Column:id

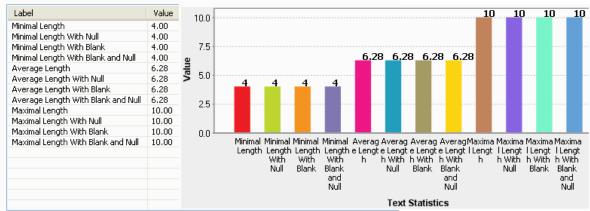
Column:firstname

🔻 Pattern Matching





Text Statistics



5.5.1.6. How to analyze delimited data in shortcut procedures

In *Talend Open Studio for Data Quality*, you can profile data in a delimited file using a simplified way. All what you need to do is to start from the column name under **Metadata** > **FileDelimited** folders in the **DQ Repository** tree view.

For further information, see Section 5.3.8, "Creating table and columns analyses in shortcut procedures".

5.5.2. Analyzing columns in an excel file

Talend Open Studio for Data Quality enables you to analyze data in an excel file and execute the created analyses using the Java engine.



Profiling excel files is done via ODBC for the time being. In later releases, you will be able to analyze excel files directly as you do with delimited files.

Prerequisite(s): At least one connection to an excel file is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.2.2, "How to connect to an Excel file".

To set up an ODBC connection to a Data Source, do the following:

1. In the DQ Repository tree view, expand Metadata, and then right-click DB connections.

The connection wizard is displayed.

New Data Define the p	base Connection on repository - Step 1/2 properties	
Name	ODBC_Excel	
Purpose		
Description		~
Author	user@company.com	
Locker		
Version	0.1	Mm
Status		~
Path		Select
?	< Back Next > Finish	Cancel

- 2. In the **Name** field, enter a name for the connection.
- 3. If required, fill in a purpose and a description for the connection, and then click **Next** to proceed to the next step.

Database Settings		
DB Type Generic OD	DBC	× 1
String of Connection	jdbc:odbc:ODBC_Excel	
Login		
Password		
DataSource	ODBC_Excel	
	Check	
	Check	
Databa 😴 Check C		
in the second	onnection	
(i) "ODE	onnection	

- 4. From the **DB Type** list, select **Generic ODBC**.
- 5. In the **DataSource** field, enter the exact name of the Data Source you created in the previous procedure.
- 6. Click the **Check** button to display a confirmation message about the status of the connection.
- 7. If your connection is successful, click **OK** to close the message, and then click **Finish** to close the wizard.
- 8. The connection is listed under **DB connections** in the **DQ Repository** tree view and the connection editor opens in the Studio.

🗟 DQ Repository 🛛 🗖 🗖	🗛 ODBC_Excel 0.1 🛛 🗖 🗖		
🕹 🗖 🗞	: 🖻 🕀		
Data Profiling Libraries	Connecti	on Settings	
 Metadata DB connections 		on Metadata erties of connection.	
🖅 🦺 400_connection	Name:	ODBC_Excel	
B B DB2	Purpose:		
MySQLServer MySQL MySQL MySQL_local ODBC_Excel ODBC_Excel	Description:		
⊡… 📳 D:\Input_Exercises\CaliforniaCi ⊡ 🗁 Tables	Author:	user@company.com	
Ġ~⊞ CaliforniaCities Ġ~⊘ Columns	Status:		
City		on information ion of connection.	
Incorporated	Login:	root	
→ Views	Password:		
	Url:	jdbc:odbc:ODBC_Excel	
Image: Bin		Check	

You can create a connection to an excel file either from the **Profiling** or the **Integration** perspectives.Once created, this connection is always displayed simultaneously in both perspectives.



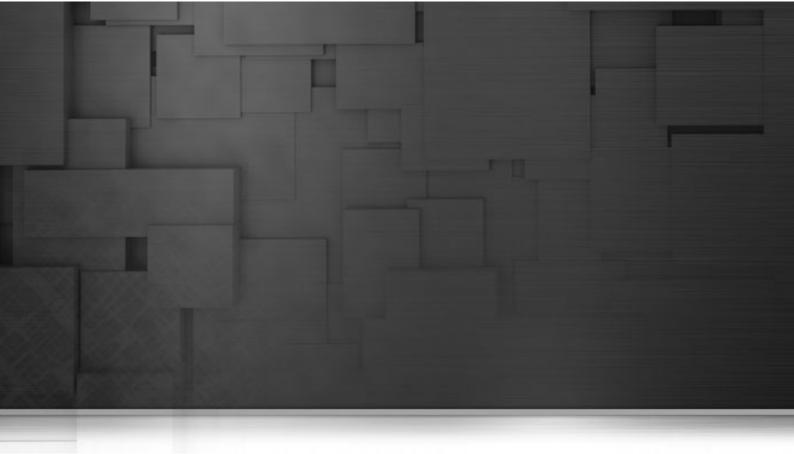
If you have difficulty retrieving the columns from the excel file, give the worksheet in the excel file the same name of the table. To do that, select the whole table in the excel file and then press Ctrl + F3 and modify the name.

You can now create a column analysis in the **Profiling** perspective to profile the columns in the excel file.

The procedures to analyze columns in an excel file are exactly the same as those for analyzing columns in a delimited file. For further information on analyzing columns in an excel files, see Section 5.5.1, "Analyzing columns in a delimited file", Section 6.2.2.4, "How to access the detailed view of the analysis results" and Section 5.4.3, "Analyzing master data in shortcut procedures".



Make sure to select the Java engine in the Analysis Parameter view in the analysis editor before executing the analysis of the excel columns, otherwise you will have an error message when running the analysis.



Chapter 6. Table analyses

This chapter provides all the information you need to perform table analyses on databases, delimited files or Master Data Management (MDM) servers.

It describes how to set up SQL business rules based on WHERE clauses and add them as indicators to database table analyses.

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

6.1. Steps to analyze a table

From *Talend Open Studio for Data Quality*, you can examine the data available in single tables of a database and collect information and statistics about this data.

The sequence of profiling data in one or multiple tables may involve the following steps:

- 1. Defining one or more tables on which to carry out data profiling processes that will define the content, structure and quality of the data included in the table(s).
- 2. Creating SQL business rules based on WHERE clauses and add them as indicators to table analyses.
- 3. Creating column functional dependencies analyses to detect anomalies in the column dependencies of the defined table(s) through defining columns as either "determinant" or "dependent".

Section 6.2, "Analyzing tables in databases" explains in detail the different options to analyze a table.

6.2. Analyzing tables in databases

Table analyses can range from simple table analyses to table analyses that uses SQL business rules or table analyses that detect anomalies in the table columns.

Talend Open Studio for Data Quality allows you to better explore the quality of data in a database table through either:

- Creating a simple table analysis through analyzing all columns in the table using patterns. For more information, see Section 6.2.1, "Creating a simple table analysis: the analysis of a set of columns".
- Adding data quality rules as indicators to table analysis. For more information, see Section 6.2.2, "Creating a table analysis with SQL business rules".
- Detecting anomalies in column dependencies. For more information, see Section 6.2.3, "Detecting anomalies in the table columns: column functional dependency analysis".

The sections below explain in detail all types of analysis that can be executed against tables.

6.2.1. Creating a simple table analysis: the analysis of a set of columns

Talend Open Studio for Data Quality enables you to analyze the content of a set of columns. This set can represent only some of the columns in the defined table or the table as a whole.

The analysis of a set of columns focuses on a column set (full records) and not on separate columns as it is the case with the column analysis. The statistics presented in the analysis results are measured against the values across all the data set and thus do not analyze the values separately within each column.



When you use the Java engine to run a column set analysis on big sets or on data with many problems, it is advisable to define a maximum memory size threshold to execute the analysis as you may end up with a Java heap error. For more information, see Section 2.3, "Defining the maximum memory size threshold".

6.2.1.1. How to create an analysis of a set of columns using patterns

This type of analysis provides simple statistics on the full records of the analyzed column set and not on the values within each column separately. For more information about simple statistic indicators, see Section 9.2.1.1, "Simple statistics".

With this analysis, you can use patterns to validate the full records against all patterns and have a single-bar result chart that shows the number of the rows that match "all" the patterns..

How to define the set of columns to be analyzed

Prerequisite(s): At least one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

To define the set of columns to analyzed, do the following:

Procedure 6.1. Defining the analysis

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



The [Create New Analysis] wizard opens.

🗳 Create New Analysis	
type filter text	
 Connection Analysis Catalog Analysis Schema Analysis Table Analysis Business Rule Analysis Functional Dependency Column Set Analysis Column Analysis Redundancy Analysis Column Correlation Analysis 	
<pre></pre>	Cancel

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

Purpose Why do you want to do this analysis Description Analysis description Author Image: Constant of the second sec	^
Author Status production Path /TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select	~
Status production Path /TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse	
Status production Path /TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse	
Status production Path /TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse	~
Path /TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select	
	~
Type Connection Analysis	Select

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



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

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

٦

Procedure 6.2. Selecting the set of columns you want to analyze

- 1. Expand **DB connections**.
- 2. In the desired database, browse to the columns you want to analyze, select them and then click **Finish** to close this **[New analysis]** wizard.

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

	考 🖲
Column S	et Analysis
 Analysis I Set the property 	Metadata arties of analysis.
Name:	Set_of_Columns
Purpose:	Analyzing a set of columns in a table
Description:	
Author:	user@comapny.com
Status:	development 💌
Analyzed	Columns
Indicator	S
🕨 Data Filte	ar -
Analysis I	Parameter

The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

3. If required, click the **Analyzed Columns** tab to open the corresponding view. Click the **Select columns to analyze** link to open a dialog box where you can modify your table or column selection.

🎽 Column Selection	ainan an		
Column Selection			
 babili_before cif convert crm crm_demo <licrm_demo< li=""> <licrm_demo< l<="" td=""><td></td><td>Image: customer_id(INT) Image: customer_id(INT)</td><td></td></licrm_demo<></licrm_demo<>		Image: customer_id(INT) Image: customer_id(INT)	
Table filter: type filter text		Column filter: type filter text	
Select All Deselect All			
?		ОК	Cancel

If you select to connect to a database that is not supported in *Talend Open Studio for Data Quality* (using the ODBC or JDBC methods), it is recommended to use the Java engine to execute the column analyses created on the selected database. For more information on the java engine, see Section 5.3.3, "Using the Java or the SQL engine".

- 4. Either:
 - expand the **DB Connections** folder and browse through the catalog/schemas to reach the table holding the columns you want to analyze, or,
 - filter the table or column lists by typing the desired text in the **Table filter** or **Column filter** fields respectively. The lists will show only the tables/columns that correspond to the text you type in.

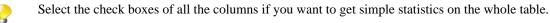


When carrying out this type of analysis, the set of columns to be analyzed must not include a primary key column.

In this example, you want to analyze a set of six columns in the *customer* table: account number (*account_num*), education (*education*), email (*email*), first name (*fname*), second name (*Iname*) and gender (*gender*). you want to identify the number of rows, the number of distinct and unique values and the number of duplicates.

5. Click the table name to list all its columns in the right-hand panel of the [Column Selection] dialog box.

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



The selected columns is displayed in the Analyzed Column view of the analysis editor.

Analyzed Columns Connection: SQL_Connection Select columns to analyze	Version:0.1		
Analyzed Columns Analyzed Columns account_num (bigint) account_num (bigint) fname (varchar) account_num (varc	Datamining Type Nominal Nominal Nominal Nominal Nominal Nominal Nominal Nominal Nominal	Pattern	Operation X X X X X
×	Move Up	(Move Down

7. If required, select to connect to a different database by selecting a different connection from the **Connection** box. This box lists all the connections created in the Studio with the corresponding database names.



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

- 8. If required, right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**. The selected column is automatically located under the corresponding connection in the tree view.
- 9. Use the delete, move up or move down buttons to manage the analyzed columns when necessary.

How to add patterns to the analyzed columns

You can add patterns to one or more of the analyzed columns to validate the full record (all columns) against all the patterns, and not to validate each column against a specific pattern as it is the case with the column analysis. The results chart is a single bar chart for the totality of the used patterns. This chart shows the number of the rows that match "all" the patterns.



Before being able to use a specific pattern with a set of columns analysis, you must manually set the pattern definition for Java in the pattern settings, if it does not already exist. Otherwise, a warning message opens prompting you to set the definition of the Java regular expression.

Prerequisite(s): An analysis of a set of columns is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see the section called "How to define the set of columns to be analyzed".

To add patterns to the analysis of a set of columns, do the following:

1. Click the 🗷 icon next to each of the columns you want to validate against a specific pattern.

The [Pattern Selector] dialog box is displayed.

📽 Pattern Selector	_ 🗆 🔀
Patterns:	
	< III
Select All De	eselect All
С СК	Cancel

?

You can add only regular expressions to the analyzed columns.

You can drop the regular expression directly from the **Patterns** folder in the **DQ Repository** tree view directly to the column name in the column analysis editor.

If no Java expression exists for the pattern you want to add, a warning message opens prompting you to add the pattern definition for Java. Click Yes to open the pattern editor and add the Java regular expression, then proceed to add the pattern to the analyzed columns.

In this example, you want to add a corresponding pattern to each of the analyzed columns to validate data in these columns against the selected patterns. The result chart will show the percentage of the matching/non-matching values, the values that respect the totality of the used patterns.

- 2. In the **[Pattern Selector]** dialog box, expand **Patterns** and browse to the regular expression you want to add to the selected column.
- 3. Select the check box(es) of the expression(s) you want to add to the selected column.
- 4. Click **OK** to proceed to the next step.

The added regular expression(s) are displayed under the analyzed column(s) in the **Analyzed Columns** list, and the All Match indicator is displayed in the **Indicators** list in the **Indicators** view.

▼ Analyzed Columns			
Connection: MySQL			
Select columns to analyze			
Analyzed Columns	Datamining Type	Pattern	Operation
account_num (bigint)	Nominal 💌	<u>_</u>	×
account_number	Nominal 💌	R	- Č
FirstCharacterUpperCase		60	Î X I
🗐 🖥 fname (varchar)	Nominal 💌	<u>_</u>	×
FirstCharacterUpperCase	Nominal 💌	<u> </u>	X
😑 📲 email (varchar)	Nominal 💌	R	×××××××××××××××××××××××××××××××××××××××
🗐 - 🚦 gender (varchar)		∈0	x I
Gender	Nominal 💌	<u>k</u>	×
education (varchar)			×
education_degree	(j)		^
×	Move Up	Mo	ive Down
Indicators			
▶ Data Filter			
· · · · · · · · · · · · · · · · · · ·			
Analysis Parameter			

How to finalize and execute the analysis of a set of columns

What is left before executing this set of columns analysis is to define indicators, data filter and analysis parameters.

Prerequisite(s): A column set analysis has already been defined in *Talend Open Studio for Data Quality*. For further information, see the section called "How to define the set of columns to be analyzed" and the section called "How to add patterns to the analyzed columns".

1. Click **Indicators** in the analysis editor to open the corresponding view.

Analyzed Columns	
▼ Indicators	
Indicators Opti Row Count Image: Count Image: Count Image: Count Ima	ons Indicator Indicator settings Set the parameters for the given indicator Indicator Thresholds Set here the thresholds expected on the indicator Lower threshold 3 Higher threshold 10 Finish Cancel

- The indicators representing the simple statistics are by-default attached to this type of analysis. For further information about the indicators for simple statistics, see section Section 9.2.1.1, "Simple statistics".
- 2. If required, click the option icon \square to open a dialog box where you can set options for each indicator.

For more information about indicators management, see Section 9.2, "Indicators".

- 3. If required, click **Data Filter** in the analysis editor to open its view and filter data through SQL "WHERE" clauses.
- 4. In the **Analysis Parameters** view, select the execution engine between SQL and Java. For further formation, see Section 5.3.3, "Using the Java or the SQL engine".

▼ Analysis Parameter
Execution engine: Dava
Allow drill down 🔽
Max number of rows kept per indicator: 50

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

For further information, see the section called "How to access the detailed result view".

💌 Analysis Paran	neter
Execution engine:	SQL 💌
Store data: 🔽	

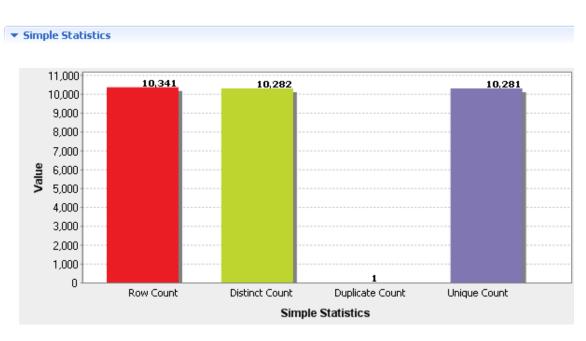
If you select the **SQL** e, select the **Store data** check box if you want to store locally the list of all analyzed rows and thus access it in the **Analysis Results** > **Data** view. For further information, see the section called "How to access the detailed result view".



If the data you are analyzing is very big, it is advisable to leave this check box unchecked in order to have only the analysis results without storing analyzed data at the end of the analysis computation.

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

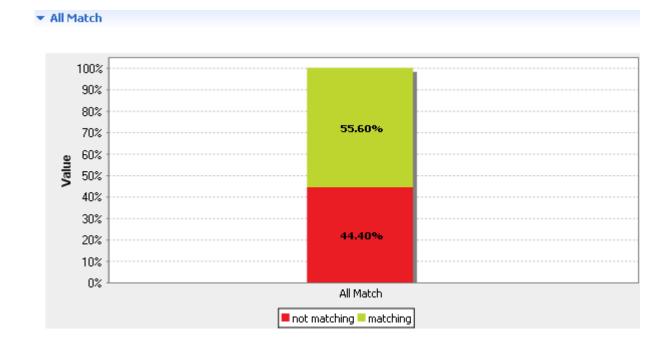
The graphical result of the set of columns analysis is displayed in the **Graphics** panel to the right of the analysis editor.



This graphical result provides the simple statistics on the full records of the analyzed column set and not on the values within each column separately.

When you use patterns to match the content of the set of columns, another graphic is displayed to illustrates the match and non-match results against the totality of the used patterns.

Refresh the graphics



How to access the detailed result view

Prerequisite(s): An analysis of a set of columns is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see the section called "How to define the set of columns to be analyzed" and the section called "How to add patterns to the analyzed columns".

To access a more detailed view of the analysis results:

1. Click the Analysis Results tab at the bottom of the analysis editor.

The corresponding view is displayed. Here you can read the analysis results in a table that accompanies the **Simple Statistics** and **All Match** graphics.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

2. To have a view of the actual analyzed data, click **Data** in the **Analysis Results** view.



In order to have the analyzed data stored in this view, you must select the Store data check box in the Analysis Parameter view. For further information, see the section called "How to finalize and execute the analysis of a set of columns".

.abel		Count	%	11,000			10,341	10,282		10,281
Row Co	ount	ا10341.00 🕲	100.00%	6 10,000				10,202		
Distinct	t Count	10282.00	99.43%	9,000			·····			
Duplica	ite Count	1.00	9.67E-3							
Unique	Count	10281.00	99.42%							
				7,000						
				6,000		-				••••••••••
				≥ 5,000				•••••		•••••••••••••••••••••••••••••••••••••••
				4,000						
				3,000						
				2,000						
				1,000					•	
				0 -	-	Ro	w Count	Distinct Count	Duplicate Count	Unique Count
ll Match	1								ple Statistics	
abel	%Match	No Match	n #Mati	h #No Match		100%				
l Match	55.60%	44.40%	5750.0							
						90%				
						80% -			•••••	
						70% -			55.60%	
						60%				
					Value	50%				
					2					
						40%				
						30% -			·····	
						20%			44.40%	
						10%				
						0%				
						0/8			All Match	
ata									not matching <mark>=</mark> matching	3
									Filter Data	
account_i			me en	ail			gender	education	COUNT(*)	
5865326		khardt Karr		renBurkhardt@T			F	Bachelors Degree	1	
8990299 5538419		rdevant Chri			Sturdevant@Anacortes.org InaNash@Arcadia.org		F	Partial High School High School Degree	1	
2943788				nWieser@Tixapa			M	Bachelors Degree	1	
3628869				baraThomas@W				Partial High School	1	
9473128	318 Dur	ocher Mat	hew Ma	thewDurocher@	Cliff	side.org	M	Partial High School	1	
9216459		mody Dale		eCarmody@Linco				High School Degree		
1267295	479 Ror	nero Bart	bara Bar	baraRomero@Ac	apu	lco	M	Graduate Degree	1	
1323063		ivan Edn		aSullivan@Redw			M	High School Degree		



You can filter analyzed data according to any of the used patterns. For further information, see the section called "How to filter data against patterns".

How to filter data against patterns

After analyzing a set of columns against a group of patterns and having the results of the rows that match or do not match "all" the patterns, you can filter the valid/invalid data according to the used patterns.

Prerequisite(s): An analysis of a set of columns is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see the section called "How to define the set of columns to be analyzed" and the section called "How to add patterns to the analyzed columns".

To filter data resulted from the analysis of a set of columns, do the following:

1. In the analysis editor, click the Analysis Results tab at the bottom of the editor to open the detailed result view.



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

2. Click **Data** to open the corresponding table.

	•	Data
--	---	------

			Filte	r Data		
account_num	Iname	fname	email	gender	education	COUNT(*)
10013550500	Murphy	William	WilliamMurphy@Ballard.org	M	Partial High School	1
10016238100	Sweet	John	JohnSweet@Port Orchard.org	F	Graduate Degree	1
10018780800	Jantzer	Elizabeth	ElizabethJantzer@Ladner.org	F	Graduate Degree	1
10022514500	Dittmar	Beverly	BeverlyDittmar@National City.org	M	High School Degree	1
10027294200	Gutirrez	Miggs	MiggsGutirrez@Victoria.org	M	High School Degree	1
10028039800	Carol	Joan	JoanCarol@Port Hammond.org	M	Partial College	1
10030158758	Holmes	Ida	IdaHolmes@La Cruz.org	F	High School Degree	1
10064045800	Chandler	Lillian	LillianChandler@Tacoma.org	M	Partial High School	1
10068825500	Burkett	Marylou	MarylouBurkett@Puyallup.org	F	High School Degree	1
10070767400	Drake	Melvin	MelvinDrake@Bremerton.org	M	Bachelors Degree	1
10072162151	Barber	Natalie	NatalieBarber@Palo Alto.org	M	High School Degree	1
10072816610	Richend	Eunice	EuniceRichendollar.Portland.free	F	High School Degree	1
10078985700	Caravello	Judy	JudyCaravello@Tlaxiaco.org	F	Partial High School	1

This table lists the actual analyzed data in the analyzed columns.

3. Click **Filter Data** on top of the table.

A dialog box is displayed listing all the patterns used in the column set analysis.

😤 Select pattern to filter the data of table							
Slelect Patterns Slelect the patterns which you want to use.							
account_num	Iname	fname	email 🗹 🔀 Email Address	gender	education		
Display:	🔿 All data	Omatches		⊙ non-match	es		
?				Fir	nish Cancel		

- 4. Select the check box(es) of the pattern(s) according to which you want to filter the data, and then select a display option according to your needs.
- 5. Select **All data** to show all analyzed data, or **matches** to show only the data that matches the pattern, or **non-matches** to show the data that does not match the selected pattern(s).
- 6. Click **Finish** to close the dialog box.

In this example, data is filtered against the *Email Address* pattern, and only the data that does not match is displayed.

Data						
					Filte	er Data
account_num	Iname	fname	email	gender	education	COUNT(*)
73604547381	Reilly	Charlo	CharloReilly@Imperial Beach	F	High School Degree	1
60462908744	Augusts	Larry	LarryAugusts@Imperial Beach.org	F	High School Degree	1
57309086918	McCurry	Andrew	AndrewMcCurry@Mexico City.org	M	Partial High School	1
53039117455	Birdwhistell	Carolyn	CarolynBirdwhistell@Walla Walla.org	M	Graduate Degree	1
90945323700	Baker	John		F	Partial High School	1
11473680734	Bomar	Herbert	HerbertBomar@Spring Valley.org	F	Bachelors Degree	1
11585722000	Haskin	Manuel	ManuelHaskin@Santa Anita.org	F	Bachelors Degree	1
81590234637	Clay	Don	DonClay@Royal Oak.org	F	High School Degree	1
66767870200	Stanley	Frederick	FrederickStanley@San Carlos.org	M	Partial High School	1
93644752502	Smith	Leonard	LeonardSmith@El Cajon.org	F	Partial High School	1
33308750391	Moore	Wendy	WendyMoore@Tlaxiaco	M	High School Degree	1
32288550734	Williams	Amanda	AmandaWilliams@San Andres.org	M	Partial High School	1
11580743789	Erickson	Harold	HaroldErickson@Long Beach.org	M	Partial College	1
84712953600	Perko	Karen	KarenPerko@Imperial Beach.org	F	Bachelors Degree	1

All email addresses that do not match the selected pattern appear in red. Any data row that has a missing value appear with a red background.

6.2.1.2. How to create a column analysis from a simple table analysis

Talend Open Studio for Data Quality allows to create a column analysis on one or more columns defined in a simple table analysis (column set analysis).

Prerequisite(s): A simple table analysis is defined in the analysis editor in *Talend Open Studio for Data Quality*.

To create a column analysis on one or more columns defined in a simple table analysis, do the following:

- 1. Open the simple table analysis.
- 2. In the Analyzed Columns view, right-click the column(s) you want to create a column analysis on.

Column Set Analysis

 Analysis Set the analy 	Metadata sis properties.			
Name:	Set_of_Columns			
Purpose:				
Description:				
Author:	user@company.com			
Status:	development 💽			
▼ Analyzed	Columns			
Connection	SQL_Connection	~	Version:0.1	
Select colum	ns to analyze			
	Analyzed Columns	Datamining Type	Pattern	Operation
	count_num (bigint)	Nominal 💌	4	×.
	ame (varchar)	Nominal 💌		Ŷ
🖻 🖷 🚦 en	nai		2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	X
	nc 🔍 Show in DQ Repository v	riew		×.
	🕂 Add Task			**
	💢 Remove elements			

3. Select **Column analysis** from the contextual menu.

The [New Analysis] wizard opens.

4. In the Name field, enter a name for the new column analysis and then click Next to proceed to the next step.

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

5. Follow the steps outlined in Section 5.3, "Analyzing columns in a database" to continue creating the column analysis.

6.2.2. Creating a table analysis with SQL business rules

Talend Open Studio for Data Quality allows you to set up SQL business rules based on WHERE clauses and add them as indicators to table analyses. You can as well define expected thresholds on the SQL business rule indicator's value. The range defined is used for measuring the quality of the data in the selected table.



It is also possible to create an analysis with SQL business rules on views in a database. The procedure is exactly the same as that for tables. For more information, see Section 6.2.2.3, "How to create a table or a view analysis with an SQL business rule".

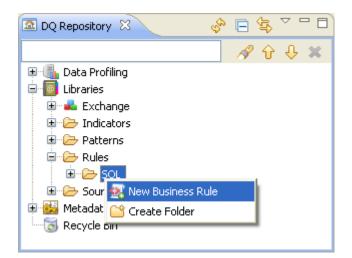
When you use the Java engine to run a column set analysis on big sets or on data with many problems, it is advisable to define a maximum memory size threshold to execute the analysis as you may end up with a Java heap error. For more information, see Section 2.3, "Defining the maximum memory size threshold".

6.2.2.1. How to create an SQL business rule

Prerequisite(s): Talend Open Studio for Data Quality is open.

To create an SQL business rule, do the following:

- 1. In the DQ Repository tree view, expand the Libraries and Rules folders in succession.
- 2. Right-click SQL.



3. From the contextual menu, select New Business Rule to open the [New Business Rule] wizard.

💕 New Bu	siness Rule	
Business I your input i	RuleCreation Page 1/2 s valid.	
Name	age	
Purpose	creating a business rule to match customer age	
Description		~
Author		
Status	development	~
Path	/TOP_DEFAULT_PRJ/TDQ_Libraries/Rules/SQL	Select
?	< Back Next > Finish	Cancel

Consider as an example that you want to create a business rule to match the age of all customers listed in the *Age* column of a defined table. You want to filter all the age records to identify those that fulfill the specified criterion.

4. In the **Name** field, enter a name for this new SQL business rule.



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

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

💕 New Busi	iness Rule	
Business Ru Define the WH	uleCreation Page 2/2 HERE clause	
Where clause	age > 18	~
?	< Back Next > Finish	Iancel

6. In the Where clause field, enter the WHERE clause to be used in the analysis.

In this example, the WHERE clause is used to match the records where customer age is greater than 18.

7. Click Finish to close the [New Business Rule] wizard.

A sub-folder for this new SQL business rule is displayed under the **Rule**s folder in the **DQ Repository** tree view. The SQL business rule editor opens with the defined metadata.

🙋 age 0.1 🔀							
: 🖻 🕀 🔛							
Business	Rule Settings		^				
▼ Business	Rule Metadata						
Set the prope	erties of Business Rule.						
Name:	age						
Purpose:	creating a business rule to match customer age						
Description:							
Author:	talend@talend.com		-				
Status:	sopment						
 Data qual Type in the de 	lity rule efinition of your Business Rules.						
Criticality Le	avel 1						
Where Clau	se age > 18		~				
<		>					
Business Rule Se	ettings						



In the SQL business rule editor, you can modify the WHERE clause or add a new one directly in the **Data quality rule** view.

- 8. If required, set a value in the **Criticality Level** field. This will act as an indicator to measure the importance of the SQL business rule.
- 9. In the SQL business rule editor, click **Join Condition** to open the corresponding view.
- 10. Click the [+] button to add as many join conditions as you want on the selected columns.

Data qua	ality rule					
r Join Con rag and dro be left table	dition op the columns t e name should b	o set the join e the same as	condition of y	our DQ Rule:	s. the da rule will	be applied
TableA	TableAliasA	ColumnA	Operator	TableB	TableAliasB	
			=			8
			= 🔤			8
			= ^^	{		
			~			
			>=			
			<=			
+	·		1			

11. Select the desired sign from the join operator box and save your modifications.

In the analysis editor, you can now drop this newly created SQL business rule onto the table that has the *Age* column. When you run the analysis, the join to the second column is done automatically.



The table to which to add the business rule must contain at least one of the columns used in the SQL business rule.

For more information about using SQL business rules as indicators on a table analysis, see Section 6.2.2, "Creating a table analysis with SQL business rules".

6.2.2.2. How to edit an SQL business rule

Prerequisite(s): Talend Open Studio for Data Quality is open.

To edit an SQL business rule, do the following:

- 1. In the DQ Repository tree view, expand the Libraries, the Rules and the SQL folders in succession.
- 2. Right-click the SQL business rule you want to open and select **Open** from the contextual menu.

The SQL business rule editor opens displaying the rule metadata.

🙍 age 0.1 🔀		
: 🖻 🖻 🔛		
Business	Rule Settings	^
▼ Business	Rule Metadata	
Set the prope	erties of Business Rule.	
Name:	age	
Purpose:	creating a business rule to match customer age	∃
Description:		
Author:	talend@talend.com	
Status:	development 💟	
💌 Data qua		
Type in the de	efinition of your Business Rules.	
Criticality Le	vel 1	
Where Clau	se age > 18	
<		>
Business Rule Se	ttings	

- 3. Modify the business rule metadata or the WHERE clause as required.
- 4. Click the save icon on top of the editor to save your modifications.

The SQL business rule is modified as defined.

6.2.2.3. How to create a table or a view analysis with an SQL business rule

Talend Open Studio for Data Quality enables you to create analyses on either tables or views in a database using SQL business rules. The procedure for creating such analysis is the same for a table or a view.

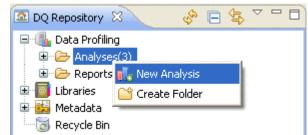
Prerequisite(s):

- At least one SQL business rule has been created in Talend Open Studio for Data Quality.
- At least one database connection is set in Talend Open Studio for Data Quality.

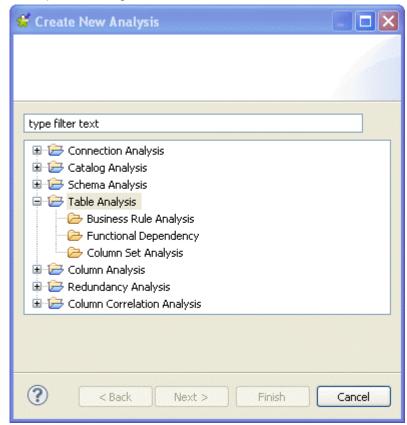
For more information about creating SQL business rules, see Section 6.2.2.1, "How to create an SQL business rule".

Procedure 6.3. Defining the analysis

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



The [Create New Analysis] wizard opens.



- 3. Expand the Table Analysis folder and select Business Rule Analysis.
- 4. Click the **Next** button to proceed to the next step.

New Analy your input is	
Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis
?	< Back Next > Finish Cancel

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



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

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

4	- 🗆 🗙
New Analysis Choose a table to analyze	
Tables: Imarketing_department Imarketing <td></td>	
Reck Next > Finish	Cancel

Procedure 6.4. Selecting the table you want to analyze

- 1. Expand **DB** Connections, browse to the table to be analyzed and select it.
- 2. Click Finish to close the [Create New Analysis] wizard.
 - 9

You can directly select the data quality rule you want to add to the current analysis by clicking the **Next** button in the **[New Analysis]** wizard or you can do that at later stage in the **Analyzed Tables** view as shown in the following steps.

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

In this example, you want to add the SQL business rule created in Section 6.2.2.1, "How to create an SQL business rule" to the *top_custom* table that contains the *Age* column. This SQL business rule will match the customer ages to define those greater than 18.

🚹 *age analys	iis 0.1 🛛	
i 🗆 🕀 🔡	 ★ 	
Table An	alysis	
	Metadata erties of analysis.	
Name:	age analysis	
Purpose:	adding age indicator to a table analysis	
Description:		
Author:	user@company.com	
Status:	development 💽	
▼ Analyzed	d Tables	
Connection	n: MySQL	
Select table:	s to analyze	
	Analyzed Tables DQ Rule Operation	
🖃 🎛 to	pp_custom 🛃 🗙 Row Count 🍙	

- 3. Click the Analyzed Tables tab to open the Analyzed Tables view.
- 4. If required, click **Select tables to analyze** to open the **[Table Selection]** dialog box and modify the selection and/or select new table(s).

📽 Table/view Selection	
Table/view Selection	
Image: Information_schema Image:	
Schema/Catalog filter: type filter text Table filter: type filter text	
Select All Deselect All	
OK Cance	el

5. Expand **DB** Connections and browse to the table(s) you want to analyze.

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

6. Select the check box next to the table name and click **OK** to proceed to the next step.

u table(s	s) is listed in the Analyzed	ables view.	
-	Analyzed Tables		
I	Connection: SQL_Connection	Ve	rsion:0.1
S	ielect tables to analyze		
	Analyzed Tables	Business Rul	e Operation
	top_custom Row Count	25	×

The selected table(s) is listed in the **Analyzed Tables** view.

- You can connect to a different database by selecting another connection from the **Connection** box. This box lists all the connections created in the Studio with the corresponding database names. If the tables listed in the **Analyzed Tables** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.
- 7. If required, right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**. The selected column is automatically located under the corresponding connection in the tree view.

Procedure 6.5. Selecting the business rule

1. Click the solution is to the table name where you want to add the SQL business rule to open the [Business Rule Selector] dialog box.

📽 Business Rule Selector 💦 🔲 🔀
Business Rules:
Rules Rules Rules Rule Rule
Select All Deselect All
ок
OK Cancel

2. Expand the **Rules** folder and select the check box(es) of the predefined SQL business rule(s) you want to use on the corresponding table(s).

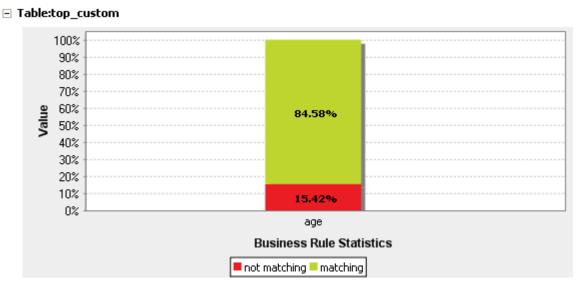
The SQL business rule used in this example will match the customer ages to define those greater than 18.

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

The selected business rule is listed below the table name in the Analyzed Tables view.

- 4. If required, click **Data Filter** in the analysis editor to open the view where you can set a filter on the data of the analyzed table(s).
- 5. Save the table analysis and press **F6** to execute it.

An information pop-up opens to confirm that the operation is in progress. The table analysis results are displayed in the **Graphics** panel to the right.



All age records in the selected table are evaluated against the defined SQL business rule. The returned results indicate in red the age records that do not match the criteria (age below 18).

You can carry out an analysis of a set of columns in a direct and more simplified way. For more information, see the section below.

6.2.2.4. How to access the detailed view of the analysis results

Prerequisite(s): A table analysis with SQL business rule is defined and executed in *Talend Open Studio for Data Quality*.

To access a more detailed view of the analysis results of the procedures outlined in Section 6.2.2.3, "How to create a table or a view analysis with an SQL business rule", do the following:

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



The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

The detailed analysis results view shows the generated graphics for the table analyzed with an SQL business rule accompanied with the table that detail the statistic results.

Analysis Results

Table:top_custom
 Business Rule Indical

%M	atch 🕴 %No Match	#Match	#No Match		100%		
84.5	3% 15.42%	170.0	31.0		90%		
🔍 View	invalid rows	L			80%		
– 🔍 View	valid rows	L			70%		
		-		ne	60%		84.58%
				Value	50%		
				-	40%		
					30%		
					20%		
					10%		15.42%
					0%		age
						Busi	ness Rule St
							ching <mark>=</mark> match

2. Right-click the SQL business rule in the table and select:

Option	То
View valid rows	access a list in the SQL editor of all valid rows measured against the pattern used on the selected table
View invalid rows	access a list in the SQL editor of all invalid rows measured against the pattern used on the selected table

1	(फ़ि फ़ि 🗁	B. 2	🛛 🝷 🕴 SQL_Connection/ro	ot 🔽 🗹 Limit Rows: 🛛	.00		*	
1	Analysi	is: Age	Analysis ;					
2			- sis: Table Analysi	is ;				
3	Purpose							
4	Descrip	tion:	;					
5	Analyze	edEleme	nt: top custom ;					
	Indicat							
		-	invalid rows ;					
		·	est top`.`top cust	om' WHERE NOT	((age>18)	11		
					11-9			
	<							
1 [SE	LECT * FROM `te	est_to]	Messages					
id	FIRST_NAME	INITIAL	EMAIL	ADDRESS_STREET	CITY	ZIP_CODE	STATE	S
[3]	Autumn	Y	eu.dolor.orci.org	P.O. Box 230, 1869 Mo	Maywood	69372	Wisconsin	IA
12	Maya	Z	netus.et.malesuada@a	5013 Magna, St.	Hopkinsville	40729	Mississippi	NY
13	Blake	I	non@montes.org	Ap #727-9073 A Ave	Lockport	90389	Pennsylv	TN

10	DIANE	1	non@monces.org	mp #727-9075 m mile	LOCKPOIL	90309	Ferinsylviii	11.4
14	Elaine	В	Proin@Maurisvelturpis.c	P.O. Box 202, 675 Alig	Watertown	65656	Georgia	NP
18	Leandra	N	montes@cursus.org	Ap #478-9720 Ornare,	Moultrie	43715	Illinois	NE
24	Coby	Р	justo.Praesent.luctus@l	P.O. Box 799, 3552 Ut	Duarte	50446	Iowa	DC
41	Marny	Н	a@liberoatauctor.org	P.O. Box 176, 7869 Est	Paducah	T1F 508	Wisconsin	W٩
50	Frances	0	nonummy.ut.molestie@	6011 Enim, Street	Newport	87552	New Mexico	PA

3. In the SQL editor, click the save icon on the toolbar to save the executed query on the SQL business rule and list it under the **Libraries > Source Files** folder in the **DQ Repository** tree view. For further information, see Section 5.3.7, "Saving the queries executed on indicators".

6.2.2.5. How to create a table analysis with an SQL business rule in a shortcut procedure

In *Talend Open Studio for Data Quality*, you can use a simplified way to create a table analysis with a predefined business rule. All what you need to do is to start from the table name under the relevant **DB Connection** folder.

Prerequisite(s):

- At least one SQL business rule is created in *Talend Open Studio for Data Quality*.
- At least one database connection is set in Talend Open Studio for Data Quality.

For more information about creating SQL business rules, see Section 6.2.2.1, "How to create an SQL business rule".

To create a table analysis with an SQL business rule in a shortcut procedure, do the following:

- 1. In the **DQ Repository** tree view, expand **Metadata** and **DB Connections** in succession and browse to the table you want to analyze.
- 2. Right-click the table name and select Table analysis from the list.

The [New Table Analysis] wizard is displayed.

#	
New Table your input is	-
Name	Set_of_Columns
Purpose	
Description	
Author	user@company.com
Status	development 💌
Path	/TDQ_PROJECT/TDQ_Data Profiling/Analyses Select
Туре	Table Analysis
?	< Back Next > Finish Cancel

3. Enter the metadata for the new analysis in the corresponding fields and then click **Next** to proceed to the next step.



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

🎽 Business Rule Selector 💦 🔲 🖾
Business Rules:
 Rules Rules SQL SQL SQE age age age_rule customer_dqrule gende_filter
Select All Deselect All
ок
OK Cancel

- 4. Expand the **Rules > SQL** folders and select the check box(es) of the predefined SQL business rule(s) you want to use on the corresponding table(s).
- 5. Click **OK** to proceed to the next step.

The table name along with the selected business rule are listed in the Analyzed Tables view.

- 6. If required, click **Data Filter** in the analysis editor to open the view where you can set a filter on the data of the analyzed table(s).
- 7. Save the table analysis and press **F6** to execute it.

An information pop-up opens to confirm that the operation is in progress. The table analysis results are displayed in the **Graphics** panel to the right.

6.2.3. Detecting anomalies in the table columns: column functional dependency analysis

This type of analysis helps you to detect anomalies in column dependencies through defining columns as either "determinant" or "dependent" and then analyzing values in dependant columns against those in determinant columns.

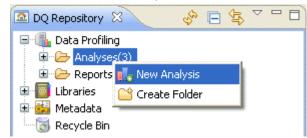
This type of analysis detects to what extent a value in a determinant column functionally determines another value in a dependant column.

This can help you identify problems in your data, such as values that are not valid. For example, if you analyze the dependency between a column that contains United States Zip Codes and a column that contains states in the United States, the same Zip Code should always have the same state. Running the functional dependency analysis on these two columns will show if there are any violations of this dependency.

Prerequisite(s): At least one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

Procedure 6.6. Defining the analysis

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



The [Create New Analysis] wizard opens.

🗳 Create New Analysis	
type filter text	
 Connection Analysis Catalog Analysis Catalog Analysis Schema Analysis Table Analysis Business Rule Analysis Business Rule Analysis Functional Dependency Column Set Analysis Column Analysis Redundancy Analysis Column Correlation Analysis 	
Reck Next > Finish	Cancel

- 3. Expand the **Table Analysis** folder and select **Functional Dependency**.
- 4. Click the **Next** button to proceed to the next step.

New Analysis							
your input is valid.							
Name	Analysis_Name						
Purpose	Why do you want to do this analysis						
Description	Analysis description						
	~						
Author							
Status	production 🕑						
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select						
Туре	Connection Analysis						
?	< Back Next > Finish Cancel						

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



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

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

New Analysis

Choose Columns to analyze

🗄 👿 DB conn	nnections ections ited connections			
?	< Back	Next >	Finish	Cancel

Procedure 6.7. Selecting the columns and executing the functional dependency analysis

1. Expand the **DB connections** node, and then browse to the columns you want to analyze, select them and then click **Finish** to close the **[New Analysis]** wizard.

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

i 🕞 🕀 🔚 📩 🔊								
Functional Dependency Analysis								
▼ Analysis Metadata								
Set the analysis properties.								
Name:	Functional_Dependencies							
Purpose:								
Description:								
Author:	user@company.com							
Status:	development 💽							
Add the dete	Analyzed Columns Set Add the determinant columns to set A (those which will determine the dependant columns of set B). The functional dependency of each pair of determinant and dependant columns (A->B) will be computed.							
Connectio	Connection: MySQL Version:0.1 Reverse columns							
💌 Left Co	▼ Left Columns ▼ Right Columns							
Dependant columns: Select the set A columns Dependant columns: Select the set B columns								
Eleme	Element(s) from customer							
E city E state_province								
🚦 sta	te_province	E city						

The display of the analysis editor depends on the parameters you set in the [**Preferences**] dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

- 2. Click the Analyzed Column Set tab to open the corresponding view.
- 3. Click **Determinant columns: Select columns from set A** to open the **[Column Selection]** dialog box.

Here you can select the first set of columns against which you want to analyze the values in the dependant columns. You can also drag the columns directly from the **DQ Repository** tree view to the left column panel.

In this example, you want to evaluate the records present in the *city* column and those present in the *state_province* column against each other to see if state names match to the listed city names and vice versa.

📽 Column Selection		
Column Selection Column Selection cif cif crm crm crm crm crm crm_demo crm_demo crm_Tables (19)	 fname(varchar) fni(varchar) mi(varchar) address1(varchar) address2(varchar) 	
customer department ment ment ment ment ment ment ment ment ment	 address3(varchar) address3(varchar) address4(varchar) city(varchar) state_province(varchar) postal_code(varchar) country(varchar) 	
Table filter: type filter text Select All Deselect All	Column filter: type filter text	
?	ОК	Cancel

4. In the [Column Selection] dialog box, expand DB Connections and browse to the column(s) you want to define as determinant columns.



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

5. Select the check box(es) next to the column(s) you want to analyze and click **OK** to proceed to the next step.

The selected column(s) are displayed in the **Left Columns** panel of the **Analyzed Columns Set** view. In this example, we select the *city* column as the determinant column.

· · ·								
 Analyzed Columns Set Add the "determinant" columns in the A set (those which will determine the "dependent" columns of the B set). The functional dependency is computed for each pair of determinant column and dependent column (A->B). 								
Connection: SQL_Connection Reverse columns								
▼ Left Columns	▼ Right Columns							
Determinant columns: Select columns from A set	Dependent columns: Select columns from B set							
Element(s) from customer	Element(s) from customer							
🚦 city	state_province							
state_province	🚦 city							

6. Do the same to select the dependant column(s) or drag it/them from the **DQ Repository** tree view to the **Right Columns** panel. In this example, we select the *state_province* column as the dependent column. This relation will show if the state names match to the listed city names.

- 7. If required, right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**. The selected column is automatically located under the corresponding connection in the tree view.
- 8. Click the **Reverse columns** tab to automatically reverse the defined columns and thus evaluate the reverse relation, what city names match to the listed state names.
 - ?

You can select to connect to a different database by selecting another connection from the **Connection** box. This box lists all the connections created in the Studio with the corresponding database names. If the columns listed in the **Analyzed Columns Set** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

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

A progress information pop-up opens to confirm that the operation is in progress. The results of column functional dependency analysis are displayed in the **Analysis Results** view.

The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

Dependency	#Match	%Mat	ch 🛛	#row						
city>state_province	109	92.379	6	118						
stațe province>citv	13	11.029	6	118						
🔜 🔍 View invalid rou	NS						Depend	ency Stre	ngth	
🔍 View detailed v	alid values		ŀ	(1%	25%		50%	75%	100%
🔍 🔍 View valid valu	es		Ŀ.,							
— 🔍 View invalid va	lues		city>state_provinc		92.37%					
🔍 View detailed ir	nvalid value:	s					_	:		
— 🔍 View valid rows	5		stal	te_province>ci				88.98%		
			-			matchin	g 📕 not m	atching		

This functional dependency analysis evaluated the records present in the *city* column and those present in the *state_province* column against each other to see if the state names match to the listed city names and vice versa.

The returned results indicate the functional dependency strength for each determinant column. The records that do not match are indicated in red.

?

The presence of null values in either of the two analyzed columns will lessen the "dependency strength". The system does not ignore null values, but rather calculates them as values that violates the functional dependency.

10. In the Analysis Results view, right-click any of the dependency lines and select:

Option	То
View valid/invalid rows	access a list in the SQL editor of all valid/invalid rows measured according to the functional dependencies analysis
View valid/invalid values	access a list in the SQL editor of all valid/invalid values measured according to the functional dependencies analysis
View detailed valid/detailed invalid values	access a detailed list in the SQL editor of all valid/invalid values measured according to the functional dependencies analysis



From the SQL editor, you can save the executed query and list it under the **Libraries > Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see Section 5.3.7, "Saving the queries executed on indicators".

6.2.4. Creating a column analysis from a simple table analysis

Talend Open Studio for Data Quality enables you to create a column analysis on one or more columns defined in a simple table analysis (column set analysis).

Prerequisite(s): A simple table analysis is defined in the analysis editor in *Talend Open Studio for Data Quality*.

To create a column analysis on one or more columns defined in a simple table analysis, do the following:

- 1. Open the simple table analysis.
- 2. In the Analyzed Columns view, right-click the column(s) you want to create a column analysis on.

Column Set Analysis

🝷 Analysis Metadata							
Set the analysis properties.							
Name:	Set_of_Columns						
Purpose:							
Description:							
Author:	user@company.com						
Status:	development 💟						
 Analyzed 	Columns						
Connection	SQL_Connection		Version:0.1				
Select colum	ns to analyze						
	Analyzed Columns	Datamining Type	Pattern	Operation			
	count_num (bigint)	Nominal 💌	4	X			
	ame (varchar)	Nominal 💌	a a b b b b b b b b b b b b b b b b b b	- Č			
	al Preview		R	- -			
- 🗄 ge	nc Column analysis		<u>R</u>	×			
🛿 ed	luc 🔍 Show in DQ Repositor	ry view	<u></u>	×			
	🐈 Add Task						
	X Remove elements						

3. Select **Column analysis** from the contextual menu.

The [New Analysis] wizard opens.

4. In the Name field, enter a name for the new column analysis and then click Finish to proceed to the next step.

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

5. Follow the steps outlined in Section 5.3, "Analyzing columns in a database" to continue creating the column analysis.

6.3. Analyzing tables in delimited files

Talend Open Studio for Data Quality enables you to analyze the content of a set of columns in a delimited file. This set can represent only some of the columns in the defined table or the table as a whole.

You can then execute the created analysis using the Java engine.

6.3.1. Creating a column set analysis on a delimited file using patterns

This type of analysis provide simple statistics on the number of records falling in certain categories, including the number of rows, the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields. For more information about these indicators, see Section 9.2.1.1, "Simple statistics".

It is also possible to add patterns to this type of analysis and have a single-bar result chart that shows the number of the rows that match "all" the patterns.

6.3.1.1. How to define the set of columns to be analyzed in a delimited file

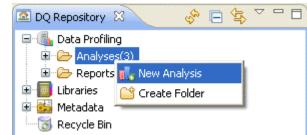
Prerequisite(s): At least one connection to a delimited file is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".



When carrying out this type of analysis, the set of columns to be analyzed must not include a primary key column.

To define the set of columns to analyzed, do the following:

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



The [Create New Analysis] wizard opens.

🗳 Create New Analysis	
type filter text	
 Connection Analysis Catalog Analysis Catalog Analysis Schema Analysis Table Analysis Business Rule Analysis Functional Dependency Column Set Analysis Column Analysis Redundancy Analysis Column Correlation Analysis 	
(?) < Back Next > Finish Ca	ancel

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

	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	N 100
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select.
Туре	Connection Analysis

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



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

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

New Analysis
Choose a Columns to analyze
Columns:
🕀 🔊 DB connections
🗐 🗊 fileDelimited
DStar_Cities
FlatFile_ColumnSet_Connection
🖮 🎫 Metadata
🖻 🗁 Columns
🚊 🖷 🚦 account_num(Long)
🖳 🗄 Iname(String)
🗐 🖷 🚦 fname(String)
🖳 🖷 🚦 email(String)
😟 🖷 🚦 gender(Character)
🛓 🖻 education(String)
MDMconnections
(?) < Back Next > Finish Cancel

- 7. Expand the **FileDelimited** connection and browse to the set of columns you want to analyze.
- 8. Select the columns to be analyzed, and then click **Finish** to close this **[New analysis]** wizard.

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

	考 💿				
Column S	et Analysis				
▼ Analysis	Metadata				
Set the prope	erties of analysis.				
Name:	FlatFile_ColumnSet_Analysi	s			
Purpose:					
Description:					< >
Author:	user@company.com				
Status:	development 🖌				
▼ Analyzed	Columns				
ac	FlatFile_ColumnSet_Conne ns to analyze Analyzed Columns count_num (Long) ame (String) ame (String)	Datamining Type	Pattern	Operation	
	hail (String)	Nominal 💌	4	X	
	nder (Character) ucation (String)	Nominal 💌		Ŷ	
	×	Move Up		we Down	
Indicator					
Data Filte					
Analysis	Parameter				

- The display of the analysis editor depends on the parameters you set in the [Preferences] dialog box.For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".
- 9. If required, select another connection from the **Connection** box in the **Analyzed Columns** view. This box lists all the connections created in the Studio with the corresponding database names.

By default, the delimited file connection you have selected in the previous step is displayed in the **Connection** box.

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

🗳 Column Selection	
Column Selection	
🔽 🎛 Metadata	Image: String
Table filter: type filter text	Column filter: type filter text
Select All Deselect All	
?	OK Cancel



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

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

In this example, you want to analyze a set of six columns in the delimited file: account number (*account_num*), education (*education*), email (*email*), first name (*fname*), second name (*Iname*) and gender (*gender*). You want to identify the number of rows, the number of distinct and unique values and the number of duplicates.

Analyzed Columns	Datamining Type	Pattern	Operation
 account_num (Long) Iname (String) fname (String) email (String) gender (Character) education (String) 	Interval Nominal Nominal Nominal Nominal Nominal Nominal		×××××

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



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

6.3.1.2. How to add patterns to the analyzed columns in the delimited file

Now, you can add patterns to one or more of the analyzed columns to validate the full record (all columns) against all the patterns, and not to validate each column against a specific pattern as it is the case with the column analysis. The results chart is a single bar chart for the totality of the used patterns. This chart shows the number of the rows that match "all" the patterns.



Before being able to use a specific pattern with a set of columns analysis, you must manually set in the patterns settings the pattern definition for Java, if it does not already exist. Otherwise, a warning message will display prompting you to set the definition of the Java regular expression.

Prerequisite(s):An analysis of a set of columns is open in the analysis editor in *Talend Open Studio for Data Quality*. For more information, see the section called "How to define the set of columns to be analyzed".

To add patterns to the analysis of a set of columns, do the following:

1. Click the 🗷 icon next to each of the columns you want to validate against a specific pattern.

The [Pattern Selector] dialog box is displayed.

🗳 Pattern Selector	
Patterns:	
 Patterns Patterns Regex Regex Code Code Color Color Color Regex Regex	
Select All	Deselect All
? ок	Cancel



You can add only regular expressions to the analyzed columns.

You can drop the regular expression directly from the **Patterns** folder in the **DQ Repository** tree view directly to the column name in the column analysis editor.



If no Java expression exists for the pattern you want to add, a warning message will display prompting you to add the pattern definition for Java. Click Yes to open the pattern editor and add the Java regular expression, then proceed to add the pattern to the analyzed columns.

In this example, you want to add a corresponding pattern to each of the analyzed columns to validate data in these columns against the selected patterns. The result chart will show the percentage of the matching/non-matching values, the values that respect the totality of the used patterns.

2. In the **[Pattern Selector]** dialog box, expand **Patterns** and browse to the regular expression you want to add to the selected column.

- 3. Select the check box(es) of the expression(s) you want to add to the selected column.
- 4. Click **OK** to proceed to the next step.

The added regular expression(s) display(s) under the analyzed column(s) in the **Analyzed Columns** view and the All Match indicator is displayed in the **Indicators** list in the **Indicators** view.

ielect columns to analyze Analyzed Columns	Datamining Type	Pattern	Opera
🗐 📲 account_num (Long)	Interval 💌	23	×
account_number			×
🗐 📲 Iname (String)	Nominal 💌	<u>I</u>	×
FirstCharacterUpperCase			****
🗐 🗄 fname (String)	Nominal 💌	<u> </u>	×
			×
🖃 🗄 email (String)	Nominal 💌	<u> </u>	×
Email_Address			×
🚊 📲 gender (Character)	Nominal 💌	<u> </u>	×
🦾 🧾 Gender			×
🖮 🚦 education (String)	Nominal 🗾	<u> </u>	×
<pre> education_degree</pre>			- ×
		Maria Da) <u>></u>
Indicators Data Filter	Move Up	Move Dow	

6.3.1.3. How to finalize and execute the column set analysis on a delimited file

What is left before executing this set of columns analysis is to define indicators, data filter and analysis parameters.

Prerequisite(s):A column set analysis is defined in *Talend Open Studio for Data Quality*. For further information, see Section 6.3.1.1, "How to define the set of columns to be analyzed in a delimited file" and Section 6.3.1.2, "How to add patterns to the analyzed columns in the delimited file".

1. Click **Indicators** in the analysis editor to open the corresponding view.

Analyzed Columns	
▼ Indicators	
Indicators Row Count The sholds:true The shold sho	Options Indicator Indicator settings Set the parameters for the given indicator Indicator Thresholds Set here the thresholds expected on the indicator Lower threshold 3 Higher threshold 10 Finish Cancel

- The indicators representing the simple statistics are by-default attached to this type of analysis. For further information about the indicators for simple statistics, see section Section 9.2.1.1, "Simple statistics".
- 2. If required, click the option icon icon to open a dialog box where you can set options for each indicator. For more information about indicators management, see Section 9.2, "Indicators".
- 3. If required, click **Data Filter** in the analysis editor to display its view and filter data through SQL "WHERE" clauses.
- 4. In the **Analysis Parameters** view, select the **Allow drill down** check box to store locally the data that will be analyzed by the current analysis.

Analysis Parameter
Execution engine: Java 🔛
Allow drill down 🔽
Max number of rows kept per indicator: 50

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



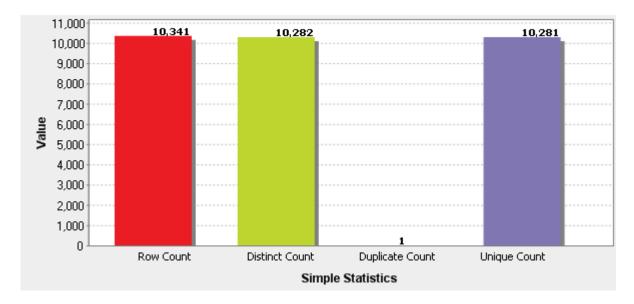
The **Allow drill down** check box is selected by default, and the maximum analyzed data rows to be shown per indicator is set to 50.

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

The **Graphics** panel to the right of the analysis editor displays the graphical result corresponding to the Simple Statistics indicators used to analyze the defined set of columns.

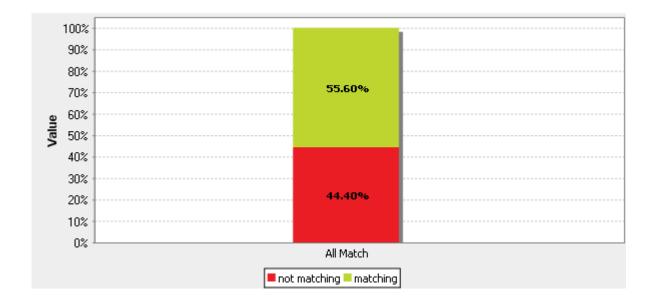
Refresh the graphics.





When you use patterns to match the content of the columns to be analyzed, another graphic is displayed to illustrates the match results against the totality of the used patterns.

All Match



6.3.1.4. How to access the detailed result view for the delimited file analysis

The procedure to access the detailed results for the delimited file analysis is the same as that for the database analysis. For further information, see the section called "How to access the detailed result view".

6.3.1.5. How to filter analysis data against patterns

The procedure to filter the data of the analysis of a delimited file is the same as that for the database analysis. For further information, see the section called "How to filter data against patterns".

6.3.2. Creating a column analysis from the analysis of a set of columns

Talend Open Studio for Data Quality enables you to create a column analysis on one or more columns defined in the set of columns analysis.

Prerequisite(s): A simple table analysis is defined in the analysis editor in Talend Open Studio for Data Quality.

To create a column analysis on one or more columns defined in the set of columns analysis, do the following:

- 1. Open the set of columns analysis.
- In the Analyzed Columns view, right-click the column(s) you want to create a column analysis on. Column Set Analysis

	Metadata erties of analysis.
Name:	FlatFile_ColumnSet_Analysis
Purpose:	
Description:	
Author:	user@company.com
Status:	development
▼ Analyzed	Columns
Connection	FlatFile_ColumnSet_Connection
Select colum	ns to analyze
	Analyzed Columns Datamining Type Pattern Opera
ina ina ina ina ina ina ina ina ina ina	count_num (Long) Interval Image: String) ame (String) Image: Column analysis Image: String) amil (String) Show in DQ Repository view Image: String) ander (Character) Add Task Image: String) ucation (String) Remove elements Image: String)

- 3. Select Column analysis from the contextual menu. The [New Analysis] wizard opens.
- 4. In the Name field, enter a name for the new column analysis and then click Next to proceed to the next step.

The analysis editor opens with the defined metadata and a folder for the newly created analysis is displayed under the **Analyses** folder in the **DQ Repository** tree view.

5. Follow the steps outlined in Section 5.5.1, "Analyzing columns in a delimited file" to continue creating the column analysis on a delimited file.

6.4. Analyzing tables on MDM servers

Talend Open Studio for Data Quality enables you to analyze the content of a set of columns "attributes" in a specific table "entity" on the MDM server. This set can represent only some of the attributes in the defined entity or the entity as a whole.

You can then execute the created analysis using the Java engine.

6.4.1. Creating a column set analysis on an MDM server

This type of analysis provide simple statistics on the number of records falling in certain categories, including the number of rows, the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields. For more information about these indicators, see Section 9.2.1.1, "Simple statistics".

6.4.1.1. How to define the set of columns to be analyzed on the MDM server

Prerequisite(s):At least one connection to an MDM server is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.3, "Connecting to an MDM server".

To define the set of columns "attributes" to be analyzed, do the following:

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

DQ Repository	3	Ŷ	E	\$ 1 }		
🖃 🚮 Data Profiling						
	🚮 New 4	Analysis e Folder				

The [Create New Analysis] wizard opens.

📽 Create New Analysis 📃 🗖 🗙
type filter text
 Connection Analysis Catalog Analysis Catalog Analysis Catalog Analysis Table Analysis Business Rule Analysis Functional Dependency Column Set Analysis Column Analysis Redundancy Analysis Column Correlation Analysis
(?) < Back Next > Finish Cancel

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

your input i:	s valid.
Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis

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



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

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

New Analysis Choose a Columns to analyze	
Columns: DB connections FileDelimited connections MDM connections MDM	
DStar Agency City(string) C	
Reck Next > Finish	Cancel

- 7. Expand **MDM connections** and browse to the set of columns "attributes" you want to analyze.
- 8. Select the attributes to be analyzed, and then click **Finish** to close this **[New analysis]** wizard.

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

Column S	🖈 💿 et Analysis				
 Analysis I 	•				
······	at the properties of analysis.				
Name:	MDM_ColumnSetAnalysis				
Purpose:	analyzing a set of attributes in	a specific entity on t	he MDM server		
Description:				< >	
Author:	user@company.com				
Status:	development 💽				
▼ Analyzed	Columns				
Connection:	MDM Version: C).1			
Select column	ns to analyze				
	Analyzed Columns	Datamining Type	Pattern	Operation	
_	(string)	Other 💌	23	×	
	me (string) y (string)	Other 💌		- X	
	ate (string)	Other 💌	R	- Ŷ	
	(string)	Other 💌	A A A A A A A A A A A A A A A A A A A	×	
🔤 🗄 Re	gion (string)	Other 💌	<u>k</u>	×	

The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

9. If required, select another connection from the **Connection** box in the **Analyzed Columns** view. This box lists all the connections created in the Studio with the corresponding database names.

By default, the connection you have selected in the previous step is displayed in the Connection box.

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

🗳 Column Selection	
Column Selection	
 DStar Agency Agent 	Id(string) Id(string)
Select All Deselect All	OK Cancel



When carrying out this type of analysis, the set of columns to be analyzed must not include a primary key column.

11. In the column list, select the check boxes of the attributes you want to analyze and click **OK** to proceed to the next step.

Selected attributes are listed in the Analyzed Columns view.

 Analyzed Columns 				
Connection: MDM 💽 Versi	ion: 0.1			
Select columns to analyze				
Analyzed Columns	Datamining Type	Pattern Operation		
💼 🚦 Id (string)	Other 💌	<u>R</u> ×		
🚦 Name (string)	Other 💌	<u>R</u> ×		
🚦 City (string)	Other 💌			
State (string)	Other 💌	<u>R</u> ×		
🔤 🖥 Zip (string)	Other 💌			
🔤 Region (string)	Other 💌	<u>R</u> ×		
×	Move Up	Move Down		

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

6.4.1.2. How to finalize and execute the analysis of a set of columns on a delimited file

What is left before executing this set of columns analysis is to define indicators and analysis parameters.

Prerequisite(s): A column set analysis has been defined in *Talend Open Studio for Data Quality*. For further information, see Section 6.4.1.1, "How to define the set of columns to be analyzed on the MDM server".

1. Click Indicators in the analysis editor to open the corresponding view.

r Indicators	
Indicators Row Count Distinct Count Duplicate Count Unique Count	Options Indicator Indicator settings your input is valid.
Data Filter Analysis Parameter	Indicator Thresholds Set here the thresholds expected on the indicator Lower threshold 3 Higher threshold 10 Finish Cancel

- The indicators representing the simple statistics are by-default attached to this type of analysis. For further information about the indicators for simple statistics, see section Section 9.2.1.1, "Simple statistics".
- 2. If required, click the option icon icon to open a dialog box where you can set options for each indicator. For more information about indicators management, see Section 9.2, "Indicators".
- 3. In the **Analysis Parameters** view, select the **Allow drill down** check box to store locally the data that will be analyzed by the current analysis.

▼ Analysis Parameter			
Execution engine: Java 🔛			
Allow drill down 🔽			
Max number of rows kept per indicator: 50			

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



The **Allow drill down** check box is selected by default, and the maximum analyzed data rows to be shown per indicator is set to 50.

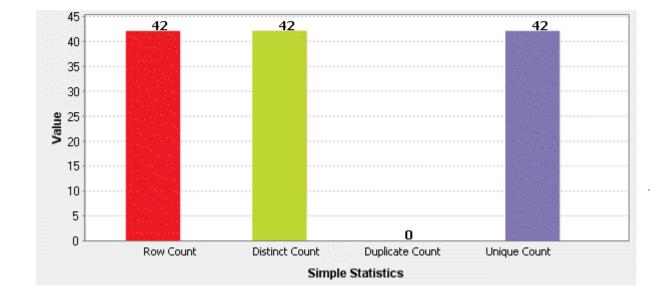
5. Click the save icon on top of the analysis editor and then press F6 to execute the analysis.

The **Graphics** panel to the right of the analysis editor displays the graphical result corresponding to the Simple Statistics indicators used to analyze the defined set of columns.

Graphics

Refresh the graphics.

```
    Simple Statistics
```



6.4.1.3. How to access the detail result view

The procedure to access the detail results for the column set analysis on an MDM server is the same as that for the same analysis on databases. For further information, see the section called "How to access the detailed result view".

6.4.2. Creating a column analysis from the column set analysis

Talend Open Studio for Data Quality enables you to create a column analysis on one or more columns defined in the set of columns analysis.

Prerequisite(s): A column set analysis has been defined in *Talend Open Studio for Data Quality*. For further information, see Section 6.4.1.1, "How to define the set of columns to be analyzed on the MDM server".

To create a column analysis on one or more columns defined in the column set analysis, do the following:

- 1. Open the column set analysis.
- 2. In the Analyzed Columns view, right-click the column(s) you want to create a column analysis on.

Column Set Analysis

▼ Analysis	Metadata
Set the prope	erties of analysis.
Name:	MDM_ColumnSetAnalysis
Purpose:	analyzing a set of attributes in a specific entity on the MDM server
Description:	
Author:	user@company.com
Status:	development 💽
▼ Analyzed	d Columns
Instance descent and have	r: MDM Version: 0.1
2010010101001001001000000	
	ame (string) Other 💌 🔣 🗙
St	d (string) Other ame (string) Other ame (string) Other ity (string) <td< td=""></td<>
	+ Add Task
	X Remove elements

- 3. Select Column analysis from the contextual menu. The [New Analysis] wizard opens.
- 4. In the **Name** field, enter a name for the new column analysis and then click **Next** to proceed to the next step. The analysis editor opens with the defined metadata and a folder for the newly created analysis is displayed under the **Analyses** folder in the **DQ Repository** tree view.
- 5. Follow the steps outlined in Section 5.4, "Analyzing master data on an MDM server" to continue creating the column analysis on a delimited file.



Chapter 7. Redundancy analysis

This chapter provides all the information you need to perform redundancy analysis that can compare table content or identify overlapping values between two sets of columns.

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

7.1. What are redundancy analyses

Redundancy analyses are column comparison analyses that better explore the relationships between tables through:

- Comparing identical columns in different tables,
- Matching foreign keys in one table to primary keys in the other table and vice versa.

The sections below provide detailed information about these two types of redundancy analyses.



The number of the analyses created in *Talend Open Studio for Data Quality* is indicated next to the **Analyses** folder in the **DQ Repository** tree view.

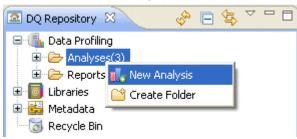
7.2. Comparing identical columns in different tables

From *Talend Open Studio for Data Quality*, you can create an analysis that compares two identical sets of columns in two different tables.

Prerequisite(s):At least one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

Procedure 7.1. Defining the analysis

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



The [Create New Analysis] wizard opens.

🗳 Create New Analysis	
Select a wizard	
Create a new Analysis	
type filter text	
Connection Analysis Cotalog Analysis Catalog Analysis Cotalog Analysis Schema Analysis Table Analysis Column Analysis Column Analysis Column Content Comparison Column Correlation Analysis	
(?) < Back Next > Finish	Cancel

- 3. Expand the **Redundancy Analysis** folder and select **Column Content Comparison**.
- 4. Click the **Next** button to proceed to the next step.

Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	~
Author	
Status	production 💌
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis

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

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

New Analysis
Choose Columns to analyze
Columns:
MDM connections
DB connections
<pre></pre>

Procedure 7.2. Selecting the identical columns you want to compare

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

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

🚹 account_back	kup_comparison 0.1 🗙	- 8		
i 🕞 🕀 🔛 🖈 💿				
ColumnSe	et Comparison Analysis			
💌 Analysis N	Metadata			
Set the prope	erties of analysis.			
Name:	account_backup_comparison			
Purpose:				
Description:				
Author:	user@company.com			
Status:	development 💟			
Analyzed	Column Sets			
🕨 Data Filte	r			
<		>		
Analysis Settings	Analysis Results			

9

The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

2. Click Analyzed Column Sets to open the view where you can set the columns or modify your selection.

In this example, you want to compare identical columns in the account and account_back tables.

Analyzed Column Sets Select tables or columns to compare. For table comparison, select one table for the A set and another table for B elements. For column comparison, select one or several columns for the A set and the same number of columns for the B s Compute only number of A rows not in B				
Connection: SQL_Connection Rev	verse columns			
Select columns for A Set	Select columns for B Set			
 account_description account_id account_parent account_rollup account_type 	 account_description account_id account_parent account_rollup account_type 			

- 3. From the **Connection** box, select the database connection relevant to the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.
- 4. Click Select columns for the A set to open the [Column Selection] dialog box.

👹 Column Selection		
Column Selection		
Image: marginal system Image: marginal system Image: marginal syst	Image: Second	r) ar)
Table filter: type filter text	olumn filter: type filter text	
Select All Deselect All		
?	OK	Cancel

5. Expand the **DB Connections** folder and browse through the catalogs/schemas to reach the table holding the columns you want to analyze.

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

- 6. Click the table name to list all its columns in the right-hand panel of the [Column Selection] dialog box.
- 7. In the list to the right, select the check boxes of the column(s) you want to analyze and click **OK** to proceed to the next step.



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



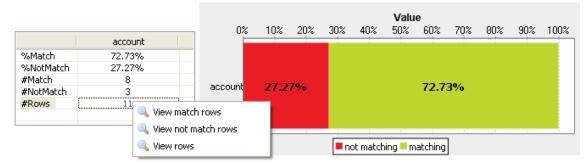
If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.

- 8. Click **Select Columns from the B set** and follow the same steps to select the second set of columns or drag it to the right column panel.
- 9. Select the **Compute only number of A rows not in B** check box if you want to match the data from the A set against the data from the B set and not vice versa.
- 10. If required, click **Data Filter** in the analysis editor to open the view where you can set a filter on each of the column sets.
- 11. Click the save icon on top of the editor and then press **F6** to execute the column comparison analysis. A confirmation message is displayed.
- 12. Read the confirmation message and click **OK** if you want to continue the operation. The **Analysis Results** view opens showing the analysis results.

In this example, 72.73% of the data present in the columns in the *account* table could be matched with the same data in the columns in the *account_back* table.



72.73% of the data from the A set (account) are found in data from the B set (account_back).



Through this view, you can also access the actual analyzed data via the Data Explorer.

To access the analyzed data rows, right-click any of the lines in the table and select:

Option	То
View match rows	access a list of all rows that could be matched in the two identical column sets
View not match rows	access a list of all rows that could not be matched in the two identical column sets
View rows	access a list of all rows in the two identical column sets

The figure below illustrates the data explorer list of all rows that could be matched in the two sets, eight in this example.

🕺 🕼 🗁 🖳 🖉 🔻 🕴 SQL_Connection/root 🛛 🕑 Limit Rows: 100					
1SELECT * FROM `crm demo`.`account` WHERE `crm demo`.`account`.accd					
<					>
1 [SELECT * FROM `crm_dem] 🔀 Messages					
account_id	account_parent	account_description	account_type	account_rollup	Custom_Members
3000	5000	Net Sales	Income	+	<null></null>
3100	3000	Gross Sales	Income	+	LookUpCube("[Sales
3200	3000	Cost of Goods Sold	Income	-	<null></null>
4000	5000	Total Expense	Expense	-	<null></null>
4100	4000	General & Administration	Expense	+	<null></null>
4200	4000	Information Systems	Expense	+	<null></null>
4300	4000	Marketing	Expense	+	<null></null>
4400	4000	Lease	Expense	+	<null></null>
Query executed in 0 ms. Number of rows returned: 8					

From the SQL editor, you can save the executed query and list it under the **Libraries > Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see Section 5.3.7, "Saving the queries executed on indicators".

The figure below illustrates the data explorer list of all rows that could not be matched in the two sets, three in this example.

1 🖈 🖻 🖗	i 🗁 🖳 🖉 🚿	7 🝷 🕴 SQL_Connecti	on/root	🔽 🗹 Limit Ro	ows: 100
1	*				
1 SELECT	1SELECT * FROM `crm_demo`.`account` WHERE `crm_demo`.`account`.acco				
					× .
<					>
1 [SELECT * FI	1 [SELECT * FROM `crm_dem] 🔀 Messages				
account_id	account_parent	account_description	account_type	account_rollup	Custom_Members
1000	<null></null>	Assets	Asset	~	<null></null>
2000	<null></null>	Liabilities	Liability	~	<null></null>
5000	<null></null>	Net Income	Income	+	<null></null>
Query execute	Query executed in 16 ms. Number of rows returned: 3				

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

7.3. Matching primary and foreign keys

From *Talend Open Studio for Data Quality*, you can create an analysis that matches foreign keys in one table to primary keys in the other table and vice versa.

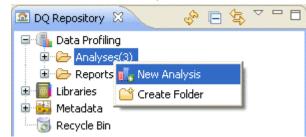
Prerequisite(s):At least one database connection is set in *Talend Open Studio for Data Quality* is open. For further information, see Section 3.1.1, "Connecting to a database".

To match primary and foreign keys in tables, do the following:

9

Procedure 7.3. defining the analysis

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



The [Create New Analysis] wizard opens.

📽 Create New Analysis	
Select a wizard Create a new Analysis	
type filter text	
E Connection Analysis	
 Catalog Analysis Schema Analysis 	
🕀 🗁 Table Analysis	
🖻 🗁 Column Analysis 🗐 🗁 Redundancy Analysis	
Column Content Comparison	
E Column Correlation Analysis	
Rext > Finish	Cancel

- 3. Expand the Redundancy Analysis folder and select Column Content Comparison.
- 4. Click the **Next** button to proceed to the next step.

New Analysis your input is valid.			
Name	Analysis_Name		
Purpose	Why do you want to do this analysis		
Description	Analysis description		
Author			
Status	production 💽		
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select		
Туре	Connection Analysis		
?	< Back Next > Finish Cancel		

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



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

6. If required, set the analysis metadata (purpose, description and author name) in the corresponding fields and click **Finish** to close the **[Create New Analysis]** wizard.

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

🚹 Redundancy	_KeyMatching 0.1 🗙	° 🗆		
i 🗆 🗄 👷 💌				
ColumnSe	et Comparison Analysis			
	Metadata arties of analysis.			
Name:	Redundancy_KeyMatching			
Purpose:				
Description:				
Author:	user@company.com			
Status:	development 💌			
Analyzed	Column Sets			
🕨 Data Filte	er			
<		>		
Analysis Settings Analysis Results				

Procedure 7.4. Selecting the primary and foreign keys

1. Click Analyzed Column Sets to display the corresponding view.

In this example, you want to match the foreign keys in the *customer_id* column of the *sales_fact_1998* table with the primary keys in the *customer_id* column of the *customer* table, and vice versa. This will explore the relationship between the two tables to show us for example if every customer has an order in the year 1998.

 Analyzed Column Sets Select tables or columns to compare. For table comparison, select one table for the A set and another table for B elements. For column comparison, select one or several columns for the A set and the same number of columns for the B set. 				
Comput	e only number of A rows not in B			
Connection: SQL_Connection Revers	e columns			
▼ Left Columns	▼ Right Columns			
Select columns for A Set	Select columns for B Set			
customer_id	customer_id			

- 2. From the **Connection** box, select the database connection relevant to the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.
- 3. Click Select columns for the A set to open the [Column Selection] dialog box.



If you want to check the validity of the foreign keys, select the column holding the foreign keys for the A set and the column holding the primary keys for the B set.

🗳 Column Selection	
Column Selection	 Customer_id(int) account_num(bigint) Iname(varchar) fname(varchar) fname(varchar) address1(varchar) address2(varchar) address3(varchar)
Table filter: type filter text	Column filter: type filter text
Select All Deselect All	
?	OK Cancel

4. Expand the **DB Connections** folder and browse through the catalogs/schemas to reach the table holding the column you want to match. In this example, the column to be analyzed is *customer_id* that holds the foreign keys.



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

- 5. Click the table name to display all its columns in the right-hand panel of the [Column Selection] dialog box.
- 6. In the list to the right, select the check box of the column holding the foreign keys and then click **OK** to proceed to the next step.



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

- If you right-click any of the listed columns in the **Analyzed Columns** view and select **Show in DQ Repository view**, the selected column will be automatically located under the corresponding connection in the tree view.
- 7. Click **Select Columns from the B set** and follow the same steps to select the column holding the primary keys or drag it from the **DQ Repository** to the right column panel.



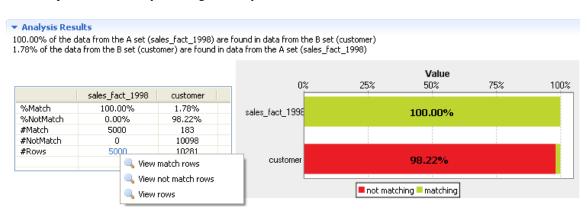
If you select the **Compute only number of rows not in B** check box, you will look for any missing primary keys in the column in the B set.

- 8. If required, click **Data Filter** in the analysis editor to display the view where you can set a filter on each of the analyzed columns.
- 9. Click the save icon on top of the editor, and then press F6 to execute this key-matching analysis. A confirmation message is displayed.
- 10. Read the confirmation message and click **OK** if you want to continue the operation.

The Analysis Results view opens to display the analysis results.

8

The execution of this type of analysis may takes some time. Wait till the **Analysis Results** view opens automatically showing the analysis results.



In this example, every foreign key in the *sales_fact_1998* table is identified with a primary key in the *customer* table. However, 98.22% of the primary keys in the *customer* table could not be identified with foreign keys in the *sales_fact_1998* table. These primary keys are for the customers who did not order anything in 1998.

Through this view, you can also access the actual analyzed data via the Data Explorer.

To access the analyzed data rows, right-click any of the lines in the table and select::

Option	То	
View match rows	access a list of all rows that could be matched in the two identical column sets	
View not match rowsaccess a list of all rows that could not be matched in the two identical columnication		
View rows	access a list of all rows in the two identical column sets	

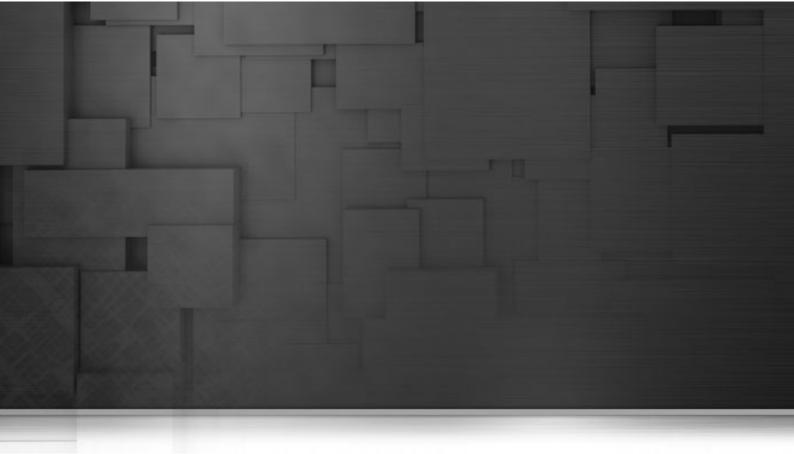
The figure below illustrates the data explorer list of all analyzed rows in the two columns.

1 🖈 🕏 🖣	i 🗁 🖫	. 🖉 🔻 🔹	SQL_Connect	ion/root	~	Limit Rows:	100	
		*						
1SELECT * FROM `crm_demo`.`sales_fact_1998`						~		
							~	
<								>
1 [SELECT * F	ROM `crm	_dem] 🔀 🛛	Messages					
product_id	time_id	customer_id	promotion_id	store_id	store_sales	store_cost	unit_sales	^
173	748	2094	54	1	4.2900	1.8447	3.0000	
1119	748	2094	54	1	9.5100	3.5187	3.0000	
1242	748	2094	54	1	7.9200	2.8512	4.0000	
460	748	2094	54	1	6.4400	2.7048	4.0000	
104	748	2094	54	1	11.6700	3.9678	3.0000	
27	748	2094	54	1	7.9500	3.8160	3.0000	
67	748	1277	54	1	7.4400	2.9016	4.0000	
217	748	1277	54	1	2.7200	0.8432	4.0000	~
Query executed in 0 ms. Number of rows returned: 100								



From the SQL editor, you can save the executed query and list it under the **Libraries > Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see Section 5.3.7, "Saving the queries executed on indicators".

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



Chapter 8. Correlation analyses

This chapter provides all the information you need to perform column correlation analyses between nominal and interval columns or nominal and date columns in database tables. A column correlation analysis can also investigate minimal correlations between nominal columns in the same table.

Column correlation analyses are usually used to explore relationships and correlations in data. They are not used to provide statistics about the quality of data.

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

8.1. What are column correlation analyses

Talend Open Studio for Data Quality provides the possibility to explore relationships and correlations between two or more columns so that these relationships and correlations give a new interpretation of the data through describing how data values are correlated at different positions.

It is very important to make the distinction between column correlation analyses and all other types of data quality analyses. Column correlation analyses are usually used to explore relationships and correlations in data and not to provide statistics about the quality of data.

Several types of column correlation analysis are possible. For more information, see Section 8.2.1, "Creating numerical correlation analysis", Section 8.3.1, "Creating time correlation analysis" and Section 8.4.1, "Creating nominal correlation analysis".

For more information about the use of data mining types in *Talend Open Studio for Data Quality*, see Section 5.2, "Data mining types".

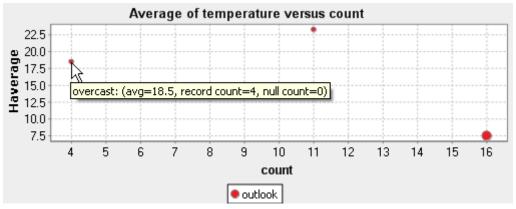


The number of the analyses created in *Talend Open Studio for Data Quality* is indicated next to the **Analyses** folder in the **DQ Repository** tree view.

8.2. Numerical correlation analysis

This type of analysis analyzes correlation between nominal and interval columns and gives the result in a kind of a bubble chart.

A bubble chart is created for each selected numeric column. In a bubble chart, each bubble represents a distinct record of the nominal column. For example, a nominal column called *outlook* with 3 distinct nominal instances: *sunny* (11 records), *rainy* (16 records) and *overcast* (4 records) will generate a bubble chart with 3 bubbles.



The second column in this example is the *temperature* column where temperature is in degrees Celsius. The analysis in this example will show the correlation between the *outlook* and the *temperature* columns and will give the result in a bubble chart. The vertical axis represents the average of the numeric column and the horizontal axis represents the number of records of each nominal instance. The average temperature would be 23.273 for the "sunny" instances, 7.5 for the "rainy" instances and 18.5 for the "overcast" instances.

The two things to pay attention to in such a chart is the position of the bubble and its size.

Usually, outlier bubbles must be further investigated. The more the bubble is near the left axis, the less confident we are in the average of the numeric column. For example, the *overcast* nominal instance here has only 4 records, hence the bubble is near the left axis. We cannot be confident in the average with only 4 records. When looking for data quality issues, these bubbles could indicate problematic values.

The bubbles near the top of the chart and those near the bottom of the chart may suggest data quality issues too. A too high or too low temperature in average could indicate a bad measure of the temperature.

The size of the bubble represents the number of null numeric values. The more there are null values in the interval column, the bigger will be the bubble.

When several nominal columns are selected, the order of the columns plays a crucial role in this analysis. A series of bubbles (with one color) is displayed for the average temperature and the weather. Another series of bubbles is displayed for the average temperature and each record of any other nominal column.

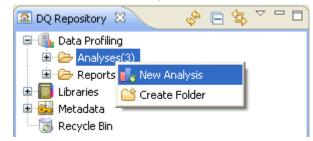
8.2.1. Creating numerical correlation analysis

In the example below, you want to create a numerical correlation analysis to compute the age average of the personnel of different enterprises located in different states. Three columns are used for the analysis: *STATE*, *AGE* and *COMPANY*.

Prerequisite(s):At least one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

Procedure 8.1. Defining the analysis

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



The [Create New Analysis] wizard opens.

📽 Create New Analysis	
Select a wizard Create a new Analysis	
type filter text Connection Analysis Catalog Analysis Catalog Analysis Catalog Analysis Column Analysis Column Analysis Column Analysis Column Correlation Analysis Numerical Correlation Analysis	
Rext > Finish	Cancel

- 3. Expand the Column Correlation Analysis folder and select Numerical Correlation Analysis.
- 4. Click the **Next** button to proceed to the next step.

Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	~
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis

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



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

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

New Analysis	
Choose Columns to analyze	
Columns:	
I MDM connections	
DB connections	
∃ FileDelimited connections	
Cancel	el

Procedure 8.2. Selecting the columns you want to analyze

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

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

🚹 Age_Average	• 0.1 🛛 🗖					
: 🖻 🕀 🔛						
Correlatio	Correlation Analysis between nominal and interval columns					
💌 Analysis I	Metadata					
Set the prope	erties of analysis.					
Name:	Age_Average					
Purpose:						
Description:						
Author:	user@company.com					
Status:	development 💟					
Analyzed	Columns					
Indicator	Indicators					
🕨 Data Filte						

The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

2. Click Analyzed Columns to open the corresponding view.

▼ Analyzed Columns	
Connection: MySQL	
Select columns to analyze	
Analyzed Columns	Datamining Type Operation

- 3. From the **Connection** box, select the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.
 - ?

You can change your database connection by selecting another connection from the **Connection** box. If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

4. Click Select columns to analyze to open the [Column Selection] dialog box.

📽 Column Selection		
Column Selection	~	 id(MEDIUMINT UNSIGNED) FIRST_NAME(VARCHAR) INITIAL(VARCHAR) EMAIL(VARCHAR) ADDRESS_STREET(VARCHAR) CITY(VARCHAR) ZIP_CODE(VARCHAR) STATE(VARCHAR)
<pre>talend_dq1 talend_dq4 talend_dq4 talend_dq4 talend_dqm2 talend_ddm2 talend_db talend_db talend_db top_custom test_top test_top test_top test_top test_top test_top test_top test_top test_top top_custom Views (0)</pre>	III	STATE_SHORT(VARCHAR) SUBSCRIPTION_DATE(TIMESTAMP) CATEGORY(TEXT) CHAR_ID(TEXT) SEQ_ENTRY(MEDIUMINT) AGE(INT) TITLE(VARCHAR) AGARITAL_STATUS(VARCHAR) MARITAL_STATUS(VARCHAR) ASSWORD(VARCHAR) SALARY(INT)
 Image: Boost of the second sec		Column filter: type filter text
(?)		OK Cancel

5. Expand **DB Connections** and browse the catalogs/schemas in your database connection to reach the table that holds the column(s) you want to analyze.



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

- 6. Click the table name to list all its columns in the right-hand panel of the [Column Selection] dialog box.
- 7. In the column list, select the check boxes of the column(s) you want to analyze and click **OK** to proceed to the next step.

In this example, you want to compute the age average of the personnel of different enterprises located in different states. Then the columns to be analyzed are *AGE*, *COMPANY* and *STATE*.

The selected columns are displayed in the Analyzed Column view of the analysis editor.

•	Analyzed Columns	
(Connection: MySQL	
<u>S</u>	elect columns to analyze	
	Analyzed Columns	Datamining Type Operation
	💬 🚦 COMPANY (varchar)	Nominal 💌 🔀
	🚦 AGE (int)	Interval 💌 🗶
	🔤 STATE (varchar)	Nominal 💌 🗶



You can drag the columns to be analyzed directly from the corresponding database connection in the **DQ Repository** tree view into the **Analyzed Columns** area.

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

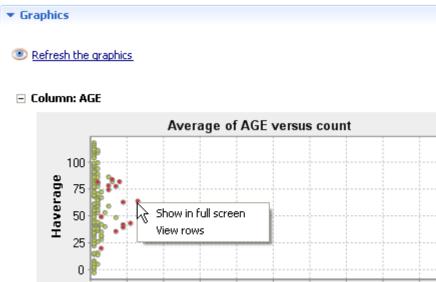
8. If required, click **Indicators** in the analysis editor to see the indicators that are used in the current numerical correlation analysis.

Analyzed Columns	
▼ Indicators	
Indicators Cont Row Count Distinct Count Duplicate Count Unique Count All Match Value Ata filter Edit the data filter: Where	Options Indicator Indicator settings Set the parameters for the given indicator Indicator Thresholds Set here the thresholds expected on the indicator Lower threshold 3 Higher threshold 10
	Finish Cancel



The indicators representing the simple statistics are by-default attached to this type of analysis.

- 9. Click the option icon icon to open a dialog box where you can set thresholds for each indicator.
- 10. If required, click **Data Filter** in the analysis editor to open the view where you can set a filter on the data of the analyzed columns.
- 11. Click the save icon on top of the editor and then press **F6** to execute the column comparison analysis. The graphical result is displayed in the **Graphics** panel to the right of the editor.



COMPANY COMPANY STATE

count

The data plotted in the bubble chart have different colors with the legend pointing out which color refers to which data.

From the generated graphic, you can:

- place the pointer on any of the bubbles to see the correlated data values at that position,
- right-click any of the bubbles and select:

Option	То	
Show in full screen open the generated graphic in a full screen		
View rows access a list of all analyzed rows in the selected position		

The below figure illustrates an example of the SQL editor listing the correlated data values at the selected position.

1	(कि कि 🗁	8. /	∽ 🔸 🕴 sql	_Connection/root	V VLi	mit Rows:	100		*
1	SELECT * F	ROM 't	est_top`.	.`top_custom` 🛛	HERE COM	IPANY`=	• '' AND 'STATE'	- 1	Alabama ' 📥
									~
	<								>
1 [SE	LECT * FROM `t	est to]	🔀 Message	s					
id	FIRST_NAME	INITIAL	EMAIL	ADDRESS STREET	ZIP CODE	STATE	SUBSCRIPTION DATE	AGE	FAVORITE CO
11	_	F	venenati		14190	Alabama	2007-10-31 21:08:00	102	_
49	Alden	D	odio.a.p	P.O. Box 646, 2777	U1B 5C3	Alabama	2007-07-04 06:50:33	116	violet
<									>
	y executed in 0 r	ns. Numbe	r of rows retu	rned: 2					

From the SQL editor, you can save the executed query and list it under the **Libraries > Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see Section 5.3.7, "Saving the queries executed on indicators".

For more information on the bubble chart, see the below section.

8.2.2. Accessing the detailed view of the analysis results

Prerequisite(s): A numerical correlation analysis is defined and executed in Talend Open Studio for Data Quality.

To access a more detailed view of the analysis results of the procedure outlined in Section 8.2.1, "Creating numerical correlation analysis", do the following:

- 1. Click the Analysis Results tab at the bottom of the analysis editor to open the corresponding view.
- 2. Click on **Analysis Result** to see more detail of the analysis results in the three different views: **Graphics**, **Simple Statistics** and **Data**.

▼ Analysis Result	
Graphics	
• Simple Statistics	
🕨 Data	



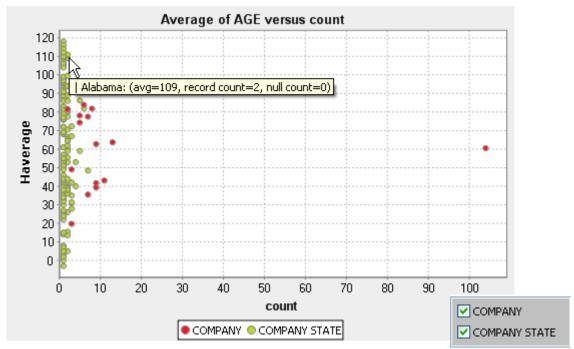
The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

3. Click **Graphics**, **Simple Statistics** or **Data** to show the generated graphic, the number of the analyzed records or the actual analyzed data respectively.

In the **Graphics** view, the data plotted in the bubble chart have different colors with the legend pointing out which color refers to which data.







The more the bubble is near the left axis the less confident we are in the average of the numeric column. For the selected bubble in the above example, the company name is missing and there are only two data records, hence the bubble is near the left axis. We cannot be confident about age average with only two records. When looking for data quality issues, these bubbles could indicate problematic values.

The bubbles near the top of the chart and those near the bottom of the chart may suggest data quality issues too, too big or too small age average in the above example.

From the generated graphic, you can:

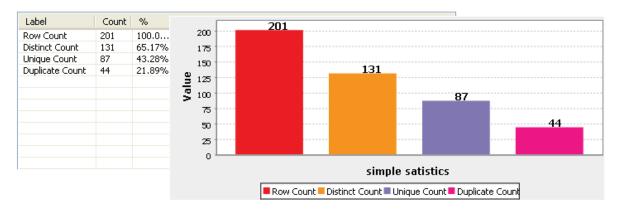
• clear the check box of the value(s) you want to hide in the bubble chart,

- place the pointer on any of the bubbles to see the correlated data values at that position,
- right-click any of the bubbles and select:

Option	То
Show in full screen	open the generated graphic in a full screen
View rows	access a list of all analyzed rows in the selected column

The **Simple Statistics** view shows the number of the analyzed records falling in certain categories, including the number of rows, the number of distinct and unique values and the number of duplicates.

Simple Statistics



The **Data** view displays the actual analyzed data.

COMPANY	STATE	AVG(AGE)	COUNT(AGE)	SUM(CASE WHEN AGE IS NULL THEN 1 ELSE 0 END)	COUNT(*)
COMPANY		AVG(AGE)		SUMICASE WHEN AGE IS NULL THEN I ELSE U END)	COOMI(*)
	Alabama	109.0000	2	0	2
	Alaska	81.8333	6	0	6
Altavista	Alaska	36.0000	1	0	1
Yahoo	Alaska	109.0000	1	0	1
	Arizona	99.0000	1	0	1
Google	Arizona	35.0000	1	0	1
Adobe	Arkansas	39.0000	2	0	2
Lycos	Arkansas	76.0000	1	0	1
Macromedia	Arkansas	83.0000	1	0	1
Yahoo	Arkansas	104.0000	1	0	1
	California	31.0000	3	0	3
Google	California	57.0000	1	0	1

You can sort the data listed in the result table by simply clicking any column header in the table.

8.3. Time correlation analysis

This type of analysis analyzes correlation between nominal and date columns and gives the result in a gantt chart that illustrates the start and finish dates of each value of the nominal column.

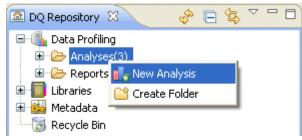
8.3.1. Creating time correlation analysis

In the example below, you want to create time correlation analysis to compute the minimal and maximal birth dates for each listed country in the selected nominal column. Two columns are used for the analysis: *birthdate* and *country*.

Prerequisite(s):At least one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

Procedure 8.3. Defining the analysis

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



The [Create New Analysis] wizard opens.

📽 Create New Analysis	
Select a wizard Create a new Analysis	
type filter text	
 Connection Analysis Catalog Analysis Catalog Analysis Cohema Analysis Column Analysis Column Analysis Column Correlation Analysis Column Correlation Analysis Correlation Analysis Numerical Correlation Analysis Numerical Correlation Analysis Nominal Correlation Analysis 	
Rext > Finish	Cancel

- 3. Expand the Column Correlation Analysis folder and select Time Correlation Analysis.
- 4. Click the **Next** button to proceed to the next step.

New Analy your input is	
Name	Analysia Mana
	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
	~
A. 46	
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis
?	< Back Next > Finish Cancel

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

New Analysis

Choose Colur	nns to analyze			
Columns:				
🕀 🙀 DB c	l connections onnections velimited connections			
?	< Back	Next >	Finish	Cancel

Procedure 8.4. Selecting the columns you want to analyze

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

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

: 🖻 🖻					
Correlation Analysis between nominal and date columns					
🔷 💌 Ana	ilysis Metadata				
Set the	e properties of analysis.				
Name:	birthdate_country_correlation				
Purpos	se:				
Descriț	ption:				
Author	r: user@company.com				
Status	development 👽				
Analyzed Columns					
🕨 Indi	icators				
🕨 Dat	a Filter				
<					



The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

2. Click Analyzed Columns to display the corresponding view.

•	Analyzed Columns		
(Connection: MySQL		
<u>s</u>	elect columns to analyze		
	Analyzed Columns	Datamining Type	Operation
	birthdate (date)	Interval 💌 Nominal 💌	××

3. From the **Connection** box, select the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.



You can change your database connection by selecting another connection from the **Connection** box. If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

4. Click **Select columns to analyze** to open the **[Column Selection]** dialog box and select the columns, or drag them directly from the **DQ Repository** tree view into the **Analyzed Columns** view.



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

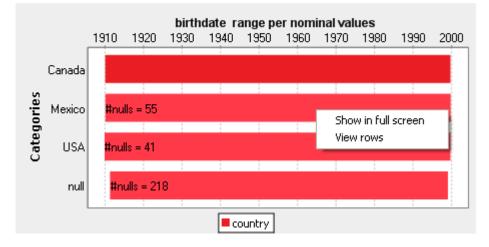
- 5. If required, click **Indicators** in the analysis editor to display the indicators used in the current time correlation analysis.
- 6. If required, click **Data Filter** in the analysis editor to display the view where you can set a filter on the analyzed column set.
- 7. Click the save icon on top of the editor and press F6 to execute the column comparison analysis.

The graphical result is displayed in the Graphics panel to the right.

▼ Graphics

Refresh the graphics.

🖃 Column: birthdate



This gantt chart displays a range showing the minimal and maximal birth dates for each country listed in the selected nominal column. It also highlights the range bars that contain null values for birth dates.

For example, in the above chart, the minimal birth date for Mexico is 1910 and the maximal is 2000. And of all the data records where the country is Mexico, 41 records have null value as birth date.

From the generated graphic, you can:

- place the pointer on any of the range bars to display the correlated data values at that position,
- put the pointer on a specific birth date and drag it to another birth date to change the chart and show the minimal and maximal birth dates related only to your selection.
- right-click any of the range bars and select:

Option	То
Show in full screen	open the generated graphic in a full screen
View rows	access a list of all analyzed rows in the selected nominal column

The below figure illustrates an example of the SQL editor listing the correlated data values at the selected range bar.

🗄 🖈 🔃 🖾 🗁 🖳 🖉 🔻 🕴 SQL_Connection/root 🛛 💉 🔽 Limit Rows: 100 👘 💽									
1SELECT * FROM `crm_demo`.`customer` WHERE `country` = 'USA'							-		
<								>	
1 [SELECT * FR	OM `crm_dem] 🔀 Messa	iges						
customer_id	account_num	Iname	fname	mi	address1	address2	address3	city	^
3	87475757600	Derry		<null></null>	7640 First Ave.	<null></null>	<null></null>	Issaguah	
5	87514054179	Gutierrez		<null></null>	8668 Via Neruda	<null></null>	<null></null>	Novato	
6	87517782449	Damstra		F.	1619 Stillman Court	<null></null>	<null></null>	Lynnwood	
10	87568712234	Stanz	Darren		1019 Kenwal Rd.	<null></null>	<null></null>	Lake Oswego	
11	87572821378	Murraiin	Jonathan		5423 Camby Rd.	<null></null>	<null></null>	La Mesa	~
<									
Query executed in 16 ms. Number of rows returned: 100									



From the SQL editor, you can save the executed query and list it under the **Libraries > Source Files** folders in the **DQ Repository** tree view if you click the save icon on the editor toolbar. For more information, see Section 5.3.7, "Saving the queries executed on indicators".

For more information on the gantt chart, see the below section.

8.3.2. Accessing the detailed view of the analysis results

Prerequisite(s): A time correlation analysis is defined and executed is *Talend Open Studio for Data Quality*.

To access a more detailed view of the analysis results of the procedure outlined in Section 8.3.1, "Creating time correlation analysis", do the following:

- 1. Click the Analysis Results tab at the bottom of the analysis editor to open the corresponding view.
- 2. Click on **Analysis Result** to display the analysis more detailed results in the three different views: **Graphics**, **Simple Statistics** and **Data**.

▼ Analysis Result
▶ Graphics
Simple Statistics
▶ Data

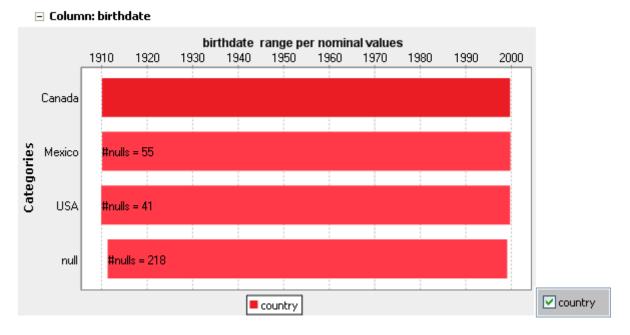


The display of the **Analysis Results** view depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

3. Click **Graphics**, **Simple Statistics** or **Data** to show the generated graphic, the number of the analyzed records or the actual analyzed data respectively.

In the Graphics view, you can clear the check box of the value(s) you want to hide in the chart.

Graphics



You can also select a specific birth date range to show if you put the pointer at the start nominal value you want to show and drag it to the end nominal value you want to show.

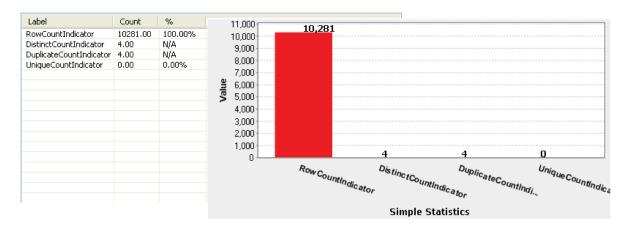
From the generated graphic, you can:

- clear the check box of the value(s) you want to hide in the chart,
- place the pointer on any of the range bars to display the correlated data values at that position,
- right-click any of the bars and select:

Option	То
Show in full screen	open the generated graphic in a full screen
View rows	access a list of all analyzed rows in the selected column

The **Simple Statistics** view shows the number of the analyzed records falling in certain categories, including the number of rows, the number of distinct and unique values and the number of duplicates.

Simple Statistics



The **Data** view displays the actual analyzed data.

🕶 Data 👘					
🔺 c.,	MIN(birthdate)	MAX(birthdate)	COUNT(birthdate)	SUM(CASE WHEN birthdate IS NULL THEN 1 ELSE 0 END)	COUNT(*)
null	1911-06-06	1999-03-20	261	0	261
Canada	1910-03-12	1999-11-05	1612	55	1667
Mexico	1910-03-19	1999-12-04	1138	41	1179
USA	1910-01-06	1999-11-04	6956	218	7174

You can sort the data listed in the result table by simply clicking any column header in the table.

8.4. Nominal correlation analysis

This type of analysis analyzes minimal correlations between nominal columns in the same table and gives the result in a chart.

In the chart, each column will be represented by a node that has a given color. The correlations between the nominal values are represented by lines. The thicker the line is, the weaker the association is. Thicker lines can identify problems or correlations that need special attention. However, you can always inverse edge weight, that is give larger edge thickness to higher correlation, by selecting the **Inverse Edge Weight** check box below the nominal correlation chart.

The correlations in the chart are always pairwise correlations: show associations between pairs of columns.

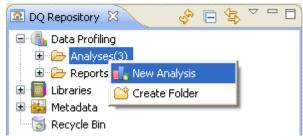
8.4.1. Creating nominal correlation analysis

In the example below, you want to create nominal correlation analysis to compute the minimal and maximal birth dates for each listed country in the selected nominal column. Two columns are used for the analysis: *birthdate* and *country*.

Prerequisite(s):At least one database connection is set in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

Procedure 8.5. Defining the analysis

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



The [Create New Analysis] wizard opens.

🗳 Create New Analysis	
Select a wizard Create a new Analysis	
type filter text	
Catalog Analysis Catalog Analysis Schema Analysis Field Analysis Field Analysis Column Analysis	
 Redundancy Analysis Column Correlation Analysis Mumerical Correlation Analysis 	
Time Correlation Analysis Nominal Correlation Analysis	
Reck Next > Finish	Cancel

- 3. Expand Column Correlation Analysis and select Nominal Correlation Analysis.
- 4. Click the **Next** button to proceed to the next step.

Name	Analysis_Name
Purpose	Why do you want to do this analysis
Description	Analysis description
Author	
Status	production
Path	/TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyse Select
Туре	Connection Analysis

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

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

New Analysis
Choose Columns to analyze
Columns:
MDM connections
DB connections
Cancel

Procedure 8.6. Selecting the columns you want to analyze

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

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

🚹 country_mari	italstatus_correlation 0.1 🛛		
: 🖻 🕀 🔛	大 ③		
Nominal (Columns Correlation		^
 Analysis I Set the property 	Metadata erties of analysis.		
Name:	country_maritalstatus_correlation		
Purpose:			
Description:			
Author:	user@company.com		
Status:	development 💽		
Analyzed	Columns		
▶ Indicator	S		
🕨 Data Filte	er		~
<		>	
Analysis Settings	Analysis Results		

The display of the analysis editor depends on the parameters you set in the **[Preferences]** dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".

2. Click Analyzed Columns to display the corresponding view.

•	Analyzed Columns	
C	Connection: MySQL	Version:0.1
Se	elect columns to analyze	
	Analyzed Columns	Datamining Type Operation

- 3. From the **Connection** box, select the database to which you want to connect. This box lists all the connections created in the Studio with the corresponding database names.
 - 9

You can change your database connection by selecting another connection from the **Connection** box. If the columns listed in the **Analyzed Columns** view do not exist in the new database connection you want to set, you will receive a warning message that enables you to continue or cancel the operation.

4. Click **Select columns to analyze** to open the **[Column Selection]** dialog box and select as many nominal columns as you want, or drag them directly from the **DQ Repository** tree view.

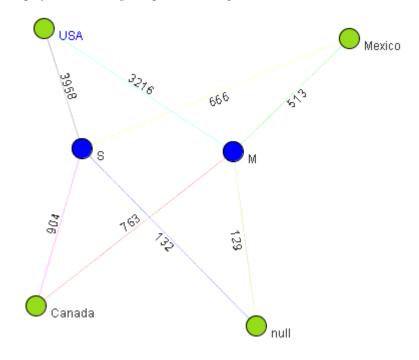


If you select too many columns, the analysis result chart will be very difficult to read.



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

- 5. If required, click **Indicators** in the analysis editor to display the indicators used in the current nominal correlation analysis.
- 6. If required, click **Data Filter** in the analysis editor to display the view where you can set a filter on the data of the analyzed columns.
- 7. Click the save icon on top of the editor and then press **F6** to execute the nominal correlation analysis. The graphical result is displayed in the **Graphics** panel to the right of the editor.



In the above chart, each value in the *country* and *marital-status* columns is represented by a node that has a given color. Nominal correlation analysis is carried out to see the relationship between the number of married or single people and the country they live in. Correlations are represented by lines.



To better view the graphical result of the nominal correlation analysis, right-click the graphic in the **Graphics** panel and select **Show in full screen**. For more information on the chart, see the below section.

8.4.2. Accessing the detailed view of the analysis results

Prerequisite(s): A nominal correlation analysis is defined and executed in Talend Open Studio for Data Quality.

To access a more detailed view of the analysis results of the procedure outlined in Section 8.4.1, "Creating nominal correlation analysis", do the following:

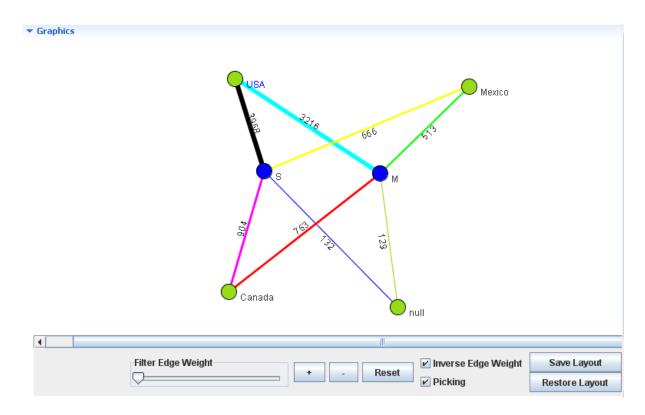
- 1. Click the Analysis Results tab at the bottom of the analysis editor to open the corresponding view.
- 2. Click on **Analysis Result** to display the analysis more detailed results in three different views: **Graphics**, **Simple Statistics** and **Data**.

▼ Analysis Result	
Graphics	
• Simple Statistics	
▶ Data	



- The display of the **Analysis Results** view depends on the parameters you set in the [**Preferences**] dialog box. For more information, see Section 2.4, "Setting preferences of analysis editors and analysis results".
- 3. Click **Graphics**, **Simple Statistics** or **Data** to show the generated graphic, the number of the analyzed records or the actual data respectively.

The Graphics view shows the generated graphic for the analyzed columns.



In the above chart, each value in the *country* and *marital-status* columns is represented by a node that has a given color. Nominal correlation analysis is carried out to see the relationship between the number of married or single people and the country they live in. Correlations are represented by lines, the thicker the line is, the higher the association is - if the **Inverse Edge Weight** check box is selected.

The buttons below the chart help you manage the chart display. The following table describes these buttons and their usage:

Button	Description
Filter Edge Weight	Move the slider to the right to (filter out edges with small weight) visualize the more important edges.
plus and minus	Click the [+] or [-] buttons to respectively zoom in and zoom out the chart size.
Reset	Click to put the chart back to its initial state.
Inverse Edge Weight	By default, the thicker the line is, the weaker the correlation is.
	Select this check box to inverse the current edge weight, that is give larger edge thickness to higher correlation.
Picking	Select this check box to be able to pick any node and drag it to anywhere in the chart.
Save Layout	Click this button to save the chart layout.
Restore Layout	Click this button to restore the chart to its previously saved layout.

The **Simple Statistics** view shows the number of the analyzed records falling in certain categories, including the number of rows, the number of distinct and unique values and the number of duplicates.

The Data view displays the actual analyzed data.

country	🔺 marital_status	COUNT(*)
null	M	129
Canada	M	763
Mexico	M	513
USA	M	3216
null	S	132
Canada	S	904
Mexico	S	666
USA	S	3958

You can sort the data listed in the result table by simply clicking any column header in the table.



Chapter 9. Extended functionality: patterns and indicators

This chapter provides detailed information about how to use regular expressions and SQL patterns to analyze and monitor data in columns. It also explains how to use system and user-defined indicators when analyzing columns.

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

9.1. Patterns

Patterns are sets of strings against which you can match the content of the columns to be analyzed.

9.1.1. Pattern types

Talend Open Studio for Data Quality lists two types of patterns under the **Patterns** folder in the **DQ Repository** tree view: regular expressions and SQL patterns.

Regular expressions (regex) are predefined patterns that you can use to search and manipulate text in the databases to which you connect. You can also create your own regular expressions and use them to analyze columns.

SQL patterns are a kind of personalized patterns that are used in SQL queries. These patterns usually contain the percent sign (%). For more information on SQL wildcards, see http://www.w3schools.com/SQL/sql_wildcards.asp.

You can use any of the above two pattern types either with column analyses or with the analyses of a set of columns (simple table analyses). These pattern-based analyses illustrate the frequencies of various data patterns found in the values of the analyzed columns. For more information, see Section 5.3, "Analyzing columns in a database" and Section 6.2.1.1, "How to create an analysis of a set of columns using patterns".

From *Talend Open Studio for Data Quality*, you can generate graphs to represent the results of analyses using patterns. You can also view tables in the **Analysis Results** view that write in words the generated graphs. From those graphs and analysis results you can easily determine the percentage of invalid values based on the listed patterns. For more information, see Section A.7, "Tab panel of the analysis editors".

Management processes for SQL patterns and regular expressions, including those for Java, are the same. For more information, see Section 9.1.4, "Managing regular expressions and SQL patterns".



Some databases do not support regular expressions. To work with such databases, some configuration is necessary before being able to use regular expressions. For more information, see Section 9.1.2, "Managing User-Defined Functions in databases".

9.1.2. Managing User-Defined Functions in databases

The regular expression function is built in several databases, but many other databases do not support it. The databases that natively support regular expressions include: MySQL, PostgreSQL, Oracle 10g, and Ingres while Microsoft SQL server does not, for example.

A different case is when the regular expression function is built in the database but the query template of the regular expression indicator is not defined.

Talend Open Studio for Data Quality enables you to:

- extend the functionality of certain database servers to support the regular expression function. For more information, see Section 9.1.2.1, "How to declare a User-Defined Function in a specific database".
- define the query template for a database that supports the regular expression function. For more information, see Section 9.1.2.2, "How to define a query template for a specific database from *Talend Open Studio for Data Quality*".

9.1.2.1. How to declare a User-Defined Function in a specific database

The regular expression function is not built into all different database environments. If you want to use *Talend Open Studio for Data Quality* to analyze columns against regular expressions in databases that do not natively support regular expressions, you can:

Either:

- 1. Install the relevant regular expressions libraries on the database. For an example of creating a regular expression function on a database, see Appendix C, *Regular expressions on SQL Server*.
- 2. Create a query template for the database in *Talend Open Studio for Data Quality*. For more information, see Section 9.1.2.2, "How to define a query template for a specific database from *Talend Open Studio for Data Quality*".

Or:

• Execute the column analysis using the Java engine. In this case, the system will use the Java regular expressions to analyze the specified column(s) and not SQL regular expressions. For more information on the Java engine, see Section 5.3.3, "Using the Java or the SQL engine".

9.1.2.2. How to define a query template for a specific database from *Talend Open Studio for Data Quality*

A query template defines the query logic required to analyze columns against regular expressions. The steps to define a query template in *Talend Open Studio for Data Quality* include the following:

- Create a query template for a specific database,
- Set the database-specific regular expression if this expression is not simple enough to be used with all databases.

The below example shows how to define a query template specific for the Microsoft SQL Server database. Appendix C, *Regular expressions on SQL Server* gives a detailed example on how to create a user-defined regular expression function on an SQL server.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To define a query template for a specific database, do the following:

- 1. In the **DQ Repository** tree view, expand **Libraries** and **Indicators** in succession.
- 2. Expand the System Indicators folder and then the Pattern Matching indicator.

🗉 🜗 Data Profilir	ng
😑 🔟 Libraries	
🗄 🔒 Exchan	ge
🚊 🗁 Indicato	ors
🖨 🗁 Sys	tem Indicators
😟 🖓 🗁	Advanced Statistics
😟 🗁	Business Rules
😟 🗁	Correlation
😟 🗁	Functional Dependency
😟 🗁	Pattern Finder
🗎 🗎 🗁	Pattern Matching
	Aregular Expression Matching 0.1
	💉 SQL Pattern Matching 0.1
🕒 🕀 🔁	Phone Number Statistics
😟 - 🔁	Row Comparison

3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

The corresponding view is displayed to show the indicator metadata and its definition.

Indicator	Indicator Settings				
 Indicator 	Metadata				
Set the prope	erties of User Defined Indicator.				
Name:	Regular Expression Matching				
Purpose:	evaluate the number of record	that match a regular pattern			
Description:	counts the number of records matching the given pattern against the number of records that do not match the given pattern				
Author:					
Status:	development 🐱				
 Indicator 	Definition				
Add here the	definition of your indicator specil	ic to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerate.			
Database	Version	SQL Template			
MySQL		SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%> REGEXP <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*) Edit			
Oracle	v	SELECT COUNT(CASE WHEN REGEX_LIKE(<%=_COLUMN_NAMES_%>,<%=_PATTERN_EXPR_%>) THEN 1 END), COL			
PostgreS0	۲.	SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%>~ <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*) FROP			
		+			

You need now to add to the list of databases the database for which you want to define a query template. This query template will compute the regular expression matching.

4. Click the [+] button at the bottom of the **Indicator Definition** view to add a field for the new template.

 Indicator Definition Add here the definition of y 	our indicator specif	ic to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerate.		
Database	Version	SQL Template		
MySQL	•	SELECT COUNT(CASE WHEN <%=COLUMN_NAMES%> REGEXP <%=PATTERN_EXPR%> THEN 1 END), COUNT(*	Edit 🗙	
Oracle		SELECT COUNT(CASE WHEN REGEX_LIKE(<%=COLUMN_NAMES%>,<%=PATTERN_EXPR%>) THEN 1 END), COL	Edit 🗙	
PostgreSQL	•	SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%>~ <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*) FROM	Edit 🔀	
Ingres	×		Edit 🔀	
		+		

- 5. In the new field, click the arrow and select the database for which you want to define the template. In this example, select **Ingres**.
- 6. Copy the indicator definition of any of the other databases.
- 7. Click the Edit... button next to the new field.

The [Edit expression] dialog box is displayed.

Ť	Edit expression	
ſ	Expression	
	SELECT COUNT(CASE WHEN REGEX <%=COLUMN_NAMES%>, <%=PATTERN_EXPR%>) THEN END), COUNT(*) FROM <%=TABLE_NAME%> <%=WHERE_CLAUSE%>	
	templates	
	<%=TABLE_NAME%> <%=COLUMN_NAMES%> <%=WHERE_CLAUSE%> <%=GROUP_BY_ALIAS%>	
(? ОК Са	incel

- 8. Paste the indicator definition (template) in the **Expression** box and then modify the text after WHEN in order to adapt the template to the selected database. In this example, replace the text after WHEN with WHEN REGEX.
- 9. Click **OK** to proceed to the next step. The new template is displayed in the field.
- 10. Click the save icon on top of the editor to save your changes.

You have finalized creating the query template specific for the **Ingres** database. You can now start analyzing the columns in this database against regular expressions.

If the regular expression you want to use to analyze data on this server is simple enough to be used with all databases, you can start your column analyses immediately. If not, you must edit the definition of the regular expression to work with this specific database, **Ingres** in this example.

For more information on how to set the database-specific regular expression definition, see Section 9.1.4.5, "How to edit a regular expression or an SQL pattern" and Section 9.1.4.8, "How to duplicate a regular expression or an SQL pattern".

9.1.2.3. How to edit a query template

Talend Open Studio for Data Quality enables you to edit the query template you create for a specific database.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To edit a query template for a specific database, do the following:

- 1. In the DQ Repository tree view, expand Libraries and Indicators in succession.
- 2. Expand the System folder and then the Pattern Matching indicator.

🗉 🜗 Data Profilir	ng
😑 🔟 Libraries	
🗄 🔒 Exchan	ge
🚊 🗁 Indicato	ors
🖨 🗁 Sys	tem Indicators
😟 🖓 🗁	Advanced Statistics
😟 🗁	Business Rules
😟 🗁	Correlation
😟 🗁	Functional Dependency
😟 🗁	Pattern Finder
🗎 🗎 🗁	Pattern Matching
	Aregular Expression Matching 0.1
	💉 SQL Pattern Matching 0.1
🕒 🕀 🔁	Phone Number Statistics
😟 - 🔁	Row Comparison

3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

The corresponding view is displayed to show the indicator metadata and its definition.

: 🖃 🖽 🗐			
Indicator	Settings		
 Indicator 	Metadata		
Set the prope	rties of User Defined Indicator.		
Name:	Regular Expression Matching		
Purpose:	evaluate the number of records that match a regular pattern		
Description:	counts the number of records matching the given pattern against the number of records that do not match the given pattern		
Author:			
Status:	development 🗸		
▼ Indicator	D-C-IN-		
·····		ic to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerate.	
Database	Version	SQL Template	
MySQL	•	SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%> REGEXP <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*; Edit	
Oracle	•	SELECT COUNT(CASE WHEN REGEX_LIKE(<%=_COLUMN_NAMES_%>,<%=_PATTERN_EXPR_%>) THEN 1 END), COL	
PostgreSQ		SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%> ~ <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*) FROP	
		+	

4. Click the Edit... button next to the database for which you want to edit the query template.

The [Edit expression] dialog box is displayed.

📽 Edit expression	×
Expression	
SELECT COUNT(CASE WHEN <%=COLUMN_NAMES%> REGEXP <%=PATTERN_EXPR%> THEN 1 END), COUNT(*) FROM <%=TABLE_NAME%> <%=WHERE_CLAUSE%>	
templates	
<%=TABLE_NAME%> <%=COLUMN_NAMES%> <%=WHERE_CLAUSE%> <%=GROUP_BY_ALIAS%>	
OK Cancel	

5. In the **Expression** area, edit the regular expression template as required and then click OK to close the dialog box and proceed to the next step.

The regular expression template is modified accordingly.

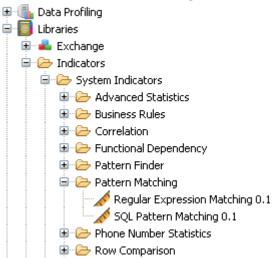
9.1.2.4. How to delete a query template

Talend Open Studio for Data Quality enables you to delete the query template you create for a specific database.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To delete a query template for a specific database, do the following:

- 1. In the DQ Repository tree view, expand Libraries and Indicators in succession.
- 2. Expand the System folder and then the Pattern Matching indicator.



3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

The corresponding view is displayed to show the indicator metadata and its definition.

t the prope	perties of User Defined Indicator.	
ame:	Regular Expression Matching	
irpose:	evaluate the number of records that match a regular pattern	
escription:	counts the number of records matching the given pattern against the number of records that do not match the given pattern	
uthor:		
atus:	development 🤜	
Indicator	r Definition	
	r Definition of your indicator specific to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerate.	
D-1-1	Under COLTextelate	
Database	e Version SQL Template	
Database MySQL	Version SQL Template SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%> REGEXP <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*)	Edit

4. Click the *solution* button next to the database for which you want to delete the query template.

The selected query template is deleted from the list in the Indicator definition view.

9.1.3. Adding regular expressions and SQL patterns to column analyses

Talend Open Studio for Data Quality allows you to use regular expressions and SQL patterns in column analyses in order to check all existing data in the analyzed columns against these expressions and patterns. For more information, see Section 5.3.6.1, "How to add a regular expression or an SQL pattern to a column analysis".

You can also edit the regular expression or SQL pattern parameters after attaching it to a column analysis. For more information, see Section 5.3.6.2, "How to edit a pattern in the column analysis".

After the execution of the column analysis that uses a specific expression or pattern, you can:

• access a list of all valid/invalid data in the analyzed column. For more information, see Section 5.3.6.3, "How to view the data analyzed against patterns".

9.1.4. Managing regular expressions and SQL patterns

In *Talend Open Studio for Data Quality*, the management procedures of regular expressions and SQL patterns include operations like creating, testing, duplicating, importing and exporting.

The sections below explain in detail each of the management option for regular expressions and SQL patterns. Management processes for both types of patterns are exactly the same.

9.1.4.1. How to create a new regular expression or SQL pattern

Talend Open Studio for Data Quality enables you to create new regular expressions or SQL patterns, including those for Java to be used in column analyses.

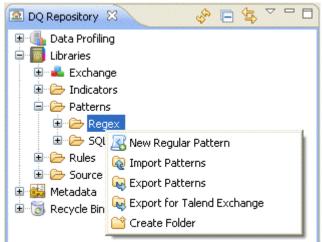


Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to create a regular expression. You can follow the same steps to create an SQL pattern.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To create a new pattern, do the following:

1. In the **DQ Repository** tree view, expand the **Libraries** and **Pattern** folders in succession and right-click **Regex**.



2. From the contextual menu, select New regular pattern to open the corresponding wizard.

📽 New Re	gex Pattern	
Regular e	xpressionCreation Page 1/2	x () 🗞 🍋 📲 🖉
your input i	s valid.	Regular Expressions:
Name	FirstCharacterUpperCase	
Purpose	identifies words starting with upper case	In the first page of the
Description	will match when the first character of a word is uppercased. For example, "Axel", "Street" will match. But "13", "JOHN DOE", "3RD FLOOR", and "stree" will not match	wizard, set the name of the pattern so that you can find it easily.
Author	user@company.com	In the second page, set
Status	development	the regular pattern to be used in analyses.
Path	/V51RC1/TDQ_Libraries/Patterns/Regex Select	The pattern must be <u>surrounded by single</u> <u>quotes " ' ".</u> Once the pattern is created you can drag & Go To: Contents % Search
?	< Back Next > Finish Cancel	📽 Related Topics 💷 Bookmarks

When you open the wizard, a help panel automatically opens with the wizard. This help panel guides you through the steps of creating new regular patterns.

3. In the Name field, enter a name for this new regular expression.

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

📽 New Regex Pat	tern 📃 🗖	
Regular expressi your input is valid.	ionCreation Page 2/2	
Regular expression: Language Selection:	'^[A-Z][a-z]*\$' MySQL	>
?	Back Next > Finish Cance	3

- 5. In the **Regular expression** field, enter the syntax of the regular expression to be created. The regular expression must be surrounded by single quotes.
- 6. From the Language Selection list, select the language (a specific database or Java).
- 7. Click **Finish** to close the dialog box.

A sub-folder for this new regular expression is listed under the **Regex** folder in the **DQ Repository** tree view, and the pattern editor opens with the defined metadata and the defined regular expression.

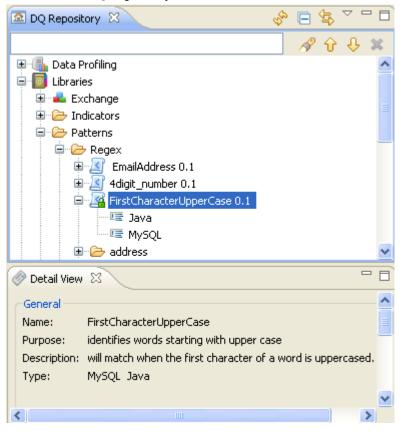
Pattern S	ettings
💌 Pattern N	1etadata
Set the prope	erties of pattern.
Name:	FirstCharacterUpperCase
Purpose:	identifies words starting with upper case
Description:	will match when the first character of a word is uppercased. For example, "Axel", Street" will match. But "13", "JOHN DOE", "3RD FLOOR", and "stree" will not match
Author:	user@company.com
Status:	development 🔜
 Pattern D Type in the databases, see 	Definition atabase-specific pattern definition. If the expression is simple enough to be used in all elect "ALL_DATABASE_TYPE" type in the list.
MySQL	▼ [^[A-Z][a-z]*\$] X Test
Java	🔽 🔀 Test
SQLite	^
Java	
Oracle Access	
DB2	×

8. In the **Pattern Definition** view, click the [+] button and add as many regular expressions as necessary in the new pattern.

You can define the regular expressions specific to any of the available databases or specific to Java.

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

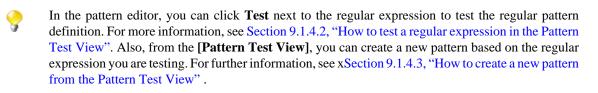
Sub-folders labeled with the specified database types or Java are listed below the name of the new pattern under the **Patterns** folder in the **DQ Repository** tree view.



9. Save the new pattern.

Once the pattern is created, you can drop it directly onto a database column in the open analysis editor.

10. If required, click the pattern name to display its detail in the **Detail View** in the Studio.



9.1.4.2. How to test a regular expression in the Pattern Test View

It is possible to test character sequences against a predefined or newly created regular expression.

Prerequisite(s): At least one database connection is set in Talend Open Studio for Data Quality.

To test a character sequence against a regular expression, do the following:

- 1. Follow the steps outlined in Section 9.1.4.1, "How to create a new regular expression or SQL pattern" to create a new regular expression.
- 2. In the open pattern editor, click **Pattern Definition** to open the relevant view.

Тур	Pattern Definition e in the database-specific pattern definition. If the expressio e in the list.	on is	simple enough to be used in all databases, select "ALL_DATABASE_TYPE"
	Oracle	,	'^[a-zA-Z0-9%-]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,4}\$' Test
	MySQL	•	'^[a-zA-Z0-9%-]+@[a-zA-Z0-9]+\\.[a-zA-Z]{2,4}\$'
	Java		'^[a-zA-Z0-9%-]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,4}\$' 🔀 Test
			+

3. Click the **Test** button next to the definition against which you want to test a character sequence to proceed to the next step.

🔓 Pattern	Test View 🛛	
🔿 Java	DB Connections MySQL	^
🥹 match	nes	
Test Area	user@Company.com	
	'^[a-zA-Z0-9%-]+@[a-zA-Z0-9]+\\.[a-zA-Z]{2,4}\$'	
	SQL Create Pattern Save Test	~

The test view is displayed in the Studio showing the selected regular expression.

- 4. In the **Test Area**, enter the character sequence you want to check against the regular expression
- 5. From the **DB Connection** list, select the database in which you want to use the regular expression.



If you select to test a regular expression in Java, the **Java** option will be selected by default and the **DB Connections** option and list will be unavailable in the test view.

6. Click Test.

An icon is displayed in the upper left corner of the view to indicate if the character sequence matches or does not match the selected pattern definition.

7. If required, modify the regular expression according to your needs and then click **Save** to save your modifications.

The pattern definition is modified accordingly in the pattern editor.



You can create/modify patterns directly from the **Pattern Test View** via the **Create Pattern** button. For further information, see Section 9.1.4.3, "How to create a new pattern from the Pattern Test View"

9.1.4.3. How to create a new pattern from the Pattern Test View

Talend Open Studio for Data Quality enables you to create your own customized patterns from the **[Pattern Test View]**.

The advantage of creating a pattern from this view is that you can create your customized pattern based on an already tested regular expression. All you need to do is to customize the expression definition according to your needs and save it to create a new pattern.

Prerequisite(s):*Talend Open Studio for Data Quality* is open.

To create a new pattern based on a predefined or a newly created regular expression, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Patterns** > **Regex** and double-click the pattern you want to use to create your customized pattern to open the pattern editor.

Pattern Settings

 Pattern Metadata Set the properties of pattern. 				
Name:	EmailAddress			
Purpose:	Check the validity of email addresses			
	Matches the email addresses	1		
Description:				
Author:				
Status:	Draft 💽			

▼ Pattern Definition

Type in the database-specific pattern definition. If the expression is simple enough to be used in all databases, select "ALL_DATABASE_TYPE" type in the list.

MySQL	[A-Z0-9%-]+@[A-Z0-9]+\\.5A-Z]{2,4}
Java	✓ '^[a-zA-Z0-9%-]+@[a-zA-Z0-9]+\.[a-z/ X Test
	+

2. Click **Test** next to the definition you want to use as a base to create the new pattern.

The [Pattern Test View] is opened on the definition of the selected regular expression.

👫 Pattern Test View	3		3
🔿 Java 💿 DB Conne	ections 10.42.10.66 👽	~	
Test Area			
'^[a-zA-Z0-	9%-]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,4}\$'		
SQL	Create Pattern Save Test		•

- 3. If required, test the regular expression through entering text in the **Test Area**. For further information, see Section 9.1.4.2, "How to test a regular expression in the Pattern Test View".
- 4. Click Create Pattern to open the [New Regex pattern] wizard.

your input i	xpressionCreation Page 1/2 s valid.
Name	my new pattern
Purpose	
Description	
	×
Author	
Status	development 💌
Path	/TOP_DEFAULT_PRJ/TDQ_Libraries/Patterns/Regex/internet Select
?	<pre>< Back Next > Finish Cancel</pre>

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

The definition of the initial regular expression is already listed in the **Regular expression** field.

¥ New Regex Pat	tern 📃 🗖 🔀
Regular expressi your input is valid.	ionCreation Page 2/2
Regular expression: Language Selection:	'^[a-zA-Z0-9%-]+@[a-zA-Z0-9]+\\.[a-zA-Z]{2,4}\$' PostgreSQL
? (< Back Next > Finish Cancel

- 7. Customize the syntax of the initial regular expression according to your needs. The regular expression definition must be surrounded by single quotes.
- 8. From the Language Selection list, select the database in which you want to use the new regular expression.
- 9. Click **Finish** to close the wizard.

A sub-folder for the new pattern is listed under the **Regex** folder in the same file of the initial regular pattern. The pattern editor opens on the pattern metadata and pattern definition.

 Pattern Definition 	
Type in the database-specific pattern definition. If the expression is simple enough to be us type in the list.	ed in all databases, select "ALL_DATABASE_TYPE"
PostgreSQL	[a-zA-Z0-9]+\\.[a-zA-Z]{2,4}\$' 🔀 Test
*	

Once the new pattern is created, you can drop it onto a column in the open analysis editor.

9.1.4.4. How to generate a regular expression from the Date Pattern Frequency Table

Talend Open Studio for Data Quality enables you to generate a regular pattern from the results of an analysis that uses the **Date Pattern Frequency Table** indicator on a date column.

Prerequisite(s):In *Talend Open Studio for Data Quality*, a column analysis is created on a date column using the **Date Pattern Frequency Table** indicator.



To be able to use the Date Pattern Frequency Table indicator on date columns, you must set the execution engine to Java in the Analysis Parameter view of the column analysis editor. For more information on execution engines, see Section 5.3.3, "Using the Java or the SOL engine".

For more information on how to create a column analysis, see Section 5.3, "Analyzing columns in a database".

To generate a regular expression from the results of a column analysis, do the following:

- 1. In the DQ Repository tree view, right-click the column analysis that uses the date indicator on a date column.
- 2. Select **Open** from the contextual menu to open the corresponding analysis editor.

Со	Column Analysis					
	Analysis Metadata Set the properties of analysis.					
	Name: date_pattern					
P	urpose:					
D	escription:					
А	uthor:	admin@company.com				
Status: development 💽						
-	▼ Analyzed Columns					
¢	Connection:	MySQL				
_		ns to analyze				
<u>S</u>	elect indical	tors for each column				
		Analyzed Columns	Datamining Type	Pattern	UDI	Operation
		L_DATE (date) Date Pattern Frequency Table	Interval 💌	<u> </u>		××

- 3. Press **F6** to execute the analysis and display the analysis results in the **Graphics** panel to the right of the Studio.
- At the bottom of the editor, click the Analysis Results tab to display a more detailed result view.
 Analysis Results



In this example, 100.00% of the date values follow the pattern yyyy MM dd and 39.41% follow the pattern yyyy dd MM.

5. Right-click the date value for which you want to generate a regular expression and select **Generate Regular Pattern** from the contextual menu.

The [New Regular Pattern] dialog box is displayed.

📽 New Re	gular Pattern					
Regular expression Creation Page 1/2						
your input i:	your input is valid.					
Name	date_pattern					
Purpose						
Description						
Author	user@company.com					
Status	development 💌					
Path	/RC2_PROJECT/TDQ_Libraries/Patterns/Regex Select					
?	< Back Next > Finish Cancel					

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

📽 New Regular Pa	attern 📃 🗖 🔀				
Regular expression Creation Page 2/2 your input is valid.					
Regular expression: Language Selection:	'^(19 20)[0-9]{2}(- /)([0-0][1-9] 10 11 12)(- /)[0-3][0-9] \$ ' Java				
?	< Back Next > Finish Cancel				

The date regular expression is already defined in the corresponding field.

7. Click **Finish** to proceed to the next step.

The pattern editor opens with the defined metadata and the generated pattern definition.

: 🖻 🕀 🔛	
Pattern S	ettings
▼ Pattern M	
Set the prope	rties of pattern.
Name:	date_pattern
Purpose:	
Description:	
Description:	
Author:	user@company.com
Status:	development 💌
💌 Pattern D	afinition
	stabase-specific pattern definition. If the expression is simple enough to be used in all databases, select
"ALL_DATABA	SE_TYPE" type in the list.
Java	Y ^{(19 20)[0-9]{2}(- /)([0-0][1-9] 10 11 12)(- /)[0-3][0-9]\$}
	+

The new regular expression is listed under **Pattern > Regex** in the **DQ Repository** tree view. You can drag it onto any date column in the analysis editor.

8. If required, click the **Test** button to test a character sequence against this date regular expression as outlined in the following section.

9.1.4.5. How to edit a regular expression or an SQL pattern

You can open the editor of any regular expression or SQL pattern to check its settings and/or edit its definition in order to:

- adapt it to a specific database type, or
- adapt it to a specific use.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To open/edit a regular expression or an SQL pattern, do the following:

- 1. In the **DQ Repository** tree view, expand the **Libraries** and the **Patterns** folders in succession.
- 2. Browse through the regular expression or SQL pattern lists to reach the expression or pattern you want to open/edit.
- 3. Right-click its name and select **Open** from the contextual menu.

The pattern editor opens displaying the regular expression or SQL pattern settings.

: 🖻 🖻 🔛			
Pattern S	ettings		
▼ Pattern N	1etadata		
	erties of pattern.		
Name:	UK Phone Number		
Purpose:	Check the validity of UK phone numbers		
Description:	Matches UK mobile phone number, with optional +44 national code. Allows optional brackets and spaces at a		
Author:			
Status:	Draft 💽		
▼ Pattern D	Pefinition		
Type in the da	atabase-specific pattern definition. If the expression is simple enough to be used in all databases, select "ALL_I		
Oracle	<pre></pre>		
MySQL	<pre></pre>		
	+		

- 4. Modify the pattern metadata, if required, and then click **Pattern Definition** to display the relevant view. In this view, you can: edit pattern definition, change the selected database and add other patterns specific to available databases through the [+] button.
- 5. If the regular expression or SQL pattern is simple enough to be used in all databases, select Default in the list.
- 6. Click the save icon on top of the editor to save your changes.
- 9

You can test regular expressions before start using them against data in the specified database. For more information, see Section 9.1.4.2, "How to test a regular expression in the Pattern Test View".



When you edit a regular expression or an SQL pattern, make sure that your modifications are suitable for all the analyses that may be using this regular expression or SQL pattern.

9.1.4.6. How to export regular expressions or SQL patterns

In *Talend Open Studio for Data Quality* you can export regular expressions and SQL patterns and store them locally in a csv file. For more information about the content lay out of the csv file, see the section called "How to import regular expressions or SQL patterns from a csv file".



Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to export regular expressions. You can follow the same steps to export SQL patterns.

How to export expressions or patterns to a csv file

Prerequisite(s): Talend Open Studio for Data Quality is open.

To export regular expressions to a csv file, do the following:

- 1. In the **DQ Repository** tree view, expand the **Libraries** and **Patterns** folders in succession and right-click **Regex**.
- 2. From the contextual menu, select **Export Patterns**.
 - The [Export Patterns] wizard opens.

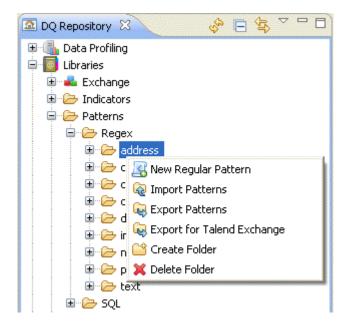
📽 Export Patterns	
Export Patterns to File Choose a file to export patterns.	
Select File (csv): D:\TDQ_builds\output.csv Selected patterns: 4digit_number 0.1 4digit_number 0.1 4digit_number 0.1 4digit_number 0.1 4digit_number 0.1 4digit_number 0.1 Characters 0.1 Composition of the select of the select All Select All Deselect All	Browse
?	Finish Cancel

- 3. Browse to the csv file where to save the regular expressions.
- 4. Click **Select All** to select all listed regular expressions or select the check boxes of the regular expressions you want to export to the csv file.
- 5. Click **Finish** to close the wizard.

All exported regular expressions are saved in the defined csv file.

To export a single regular expression family to a csv file, do the following:

1. In the **DQ Repository** tree view, expand the **Libraries** and **Patterns** in succession and browse to the regular expression family you want to export.



2. From the contextual menu, select **Export Patterns**.

The [Export Patterns] wizard opens.

🗳 Export Patterns	
Export Patterns to File Choose a file to export patterns.	
Select File (csv): D:\TDQ_builds\output.csv Selected patterns: BE Code postal 0.1 Companies House 0.1 Companies House 0.1 FR Code postal code) 0.1 FR Code postal 0.1 FR Code postal 0.1 FR Code or Pin code of India 0.1 FR Code validation 0.1 Code Validation 0.1 FR Code Validation 0.1	Browse
Select All Deselect All	Finish Cancel

3. Click **Select All** to select all the check boxes of the regular expressions or select the check boxes of the regular expressions you want to export to the csv file.

4. Click **Finish** to close the wizard.

All exported regular expressions are saved in the defined csv file.

How to export expressions or patterns to Talend Exchange

You can export regular expressions or SQL patterns from your current version of *Talend Open Studio for Data Quality* to **Talend Exchange** where you can share them with other users.



Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to export regular expressions to **Talend Exchange**. You can follow the same steps to export SQL patterns to **Talend Exchange**.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To export regular expressions to Talend Exchange, do the following:

- 1. In the DQ Repository tree view, expand Libraries and Patterns in succession.
- 2. Right-click **Regex** and select **Export for Talend Exchange**.

The [Export for Talend Exchange] wizard is displayed.

📽 Export for Talend Exchange	
Export patterns for Talend Exchange Choose a folder where to export the patterns.	
Select a folder: D:\50b Selected patterns:	Browse
?	Finish Cancel

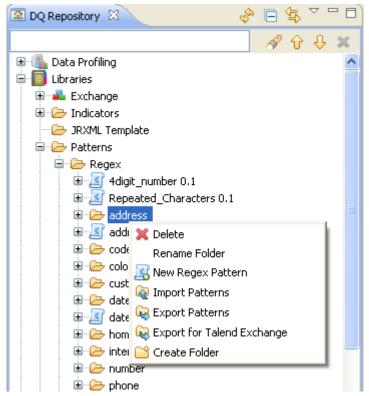
- 3. Browse to the folder where to save regular expressions.
- 4. Click Select All to select all the regular expressions in the list or select the check boxes of the regular expressions you want to export to the specified folder.

5. Click **Finish** to close the wizard.

A distinct csv file is created for each exported regular expression. Each csv file is compressed as a zip. All these zip files are saved in the defined folder. You need now to upload them to **Talend Exchange** at http://www.talendforge.org/exchange/top/help_guest.php. For information about how to upload a file to **Talend Exchange**, see *Talend Open Studio for Data Integration User Guide*.

To export a single regular expression family to **Talend Exchange**, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** and **Patterns** in succession browse to the regular expression you want to export.



2. Right-click it and then select **Export for Talend Exchange** from the contextual menu.

The [Export for Talend Exchange] wizard opens.

📽 Export for Talend Exchange
Export patterns for Talend Exchange Choose a folder where to export the patterns.
Select a folder: D:\50b Browse Selected patterns: Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House 0.1 Image: Companies House Codes Upper and Lower Case 0.1
Select All Deselect All
? Finish Cancel

- 3. Click **Select All** to select all the regular expressions in the list, or select the check boxes of the regular expressions or SQL patterns you want to export to the folder.
- 4. Click **Finish** to close the wizard.

A distinct csv file is created for each exported regular expression or SQL pattern. Each csv file is compressed as zip. All these zip files are saved in the defined folder.

You need now to upload them to **Talend Exchange** at http://www.talendforge.org/exchange/top/help_guest.php. For information about how to upload a file to **Talend Exchange**, see *Talend Open Studio for Data Integration User Guide*.

9.1.4.7. How to import regular expressions or SQL patterns

In *Talend Open Studio for Data Quality* you can import the regular expressions or SQL patterns stored locally in a csv file. The csv file must have 11 columns laid out as follows:

Column name	Description
Label	the label of the pattern (must not be empty)
Purpose	the purpose of the pattern (can be empty)
Description	the description of the pattern (can be empty)
Author	the author of the regular expression (can be empty)

Column name	Description
Relative Path	the relative path to the root folder (can be empty)
All DB Regular	the regular expression applicable to all databases (can be empty)
DB2 Regexp	the regular expression applicable to DB2 databases (can be empty)
MySQL Regexp	the regular expression applicable to MySQL databases (can be empty)
Oracle Regexp	the regular expression applicable to Oracle databases (can be empty)
PostgreSQL Regexp	the regular expression applicable to PostgreSQL databases (can be empty)
SQL Server Regexp	the regular expression applicable to SQL Server databases (can be empty)

Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to import regular expressions. You can follow the same steps to import SQL patterns.

How to import regular expressions or SQL patterns from a csv file

Prerequisite(s):*Talend Open Studio for Data Quality* is open. The csv file is stored locally.

To import regular expressions from a csv file, do the following:

- 1. In the **DQ Repository** tree view, expand the **Libraries** and **Patterns** folders in succession.
- 2. Right-click Regex and select Import patterns.

The [Import Patterns] wizard opens.

📽 Import Patterns 📃 🗖 📔				
Import Patterns Choose a file to impo				
Select File : D:\output\patterns.csv Browse Duplicate patterns handling Skip existing patterns rename new patterns with suffix Preview:				
Label	Purpose	Description	Author	Relative_
Austria VAT Numbe		Vat number for A	Pana	
Bulgaria Vat Numbe		Vat number for Bu		
French VATNumber		Matches FRAB 12		
Gender	Classic ge	Matches: F, M, M	Keith	
<				>
? Finish Cancel				

- 3. Browse to the csv file holding the regular expressions.
- 4. In the **Duplicate patterns handling** area, select:

Option	То
skip existing patterns	import only the regular expressions that do not exist in the corresponding lists in the DQ Repository tree view. A warning message is displayed if the imported patterns already exist under the Patterns folder.
rename new patterns with suffix	identify each of the imported regular expressions with a suffix. All regular expression will be imported even if they already exist under the Patterns folder.

5. Click **Finish** to close the wizard.

All imported regular expressions are listed under the **Regex** folder in the **DQ Repository** tree view.

A warning icon ^(a) next to the name of the imported regular expression or SQL pattern in the tree view identifies that it is not correct. You must open the expression or the pattern and try to figure out what is wrong. Usually, problems come from missing quotes. Check your regular expressions and SQL patterns and ensure that they are encapsulated in single quotes.

How to import regular expressions or SQL patterns from Talend Exchange

You can import regular expressions or SQL patterns from **Talend Exchange** to your current version of *Talend Open Studio for Data Quality* and use them on analyzed columns.

Make sure that the Talend Exchange extension you want to import from is compatible with your current Studio version. Compatibility means that Talend Exchange extension has the same two first sequences of the unique identifier of your current version of Talend Open Studio for Data Quality. For example, if your current version of Talend Open Studio for Data Quality is 3.2.0, compatible extensions could be 3.2.1, 3.2.0M1, 3.2.0M2, 3.2.0RC1, etc.

Prerequisite(s): Talend Open Studio for Data Quality is open. Your network is up and running.



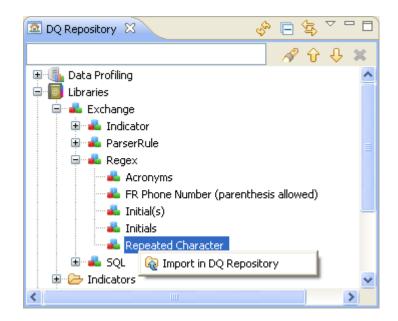
If you have connection problems, you will not be able to access any of the regular expressions or SQL patterns under the **Exchange** node in the **DQ Repository** tree view.



Management processes for regular expressions and SQL patterns are the same. The procedure below with all the included screen captures reflect the steps to import regular expressions from **Talend Exchange**. You can import SQL patterns following the same steps.

To import regular expressions from Talend Exchange, do the following:

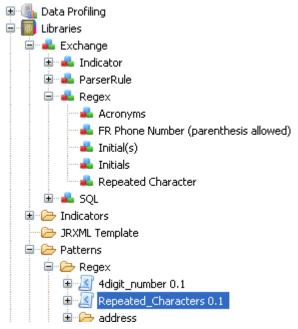
- 1. In the **DQ Repository** tree view, expand **Libraries** and **Exchange** in succession.
- 2. Under Exchange, expand Regex and right-click the name of the pattern you want to import.
- 3. Select **Import in DQ Repository** from the contextual menu.



A message is displayed to confirm the operation.

4. Click **OK** in the confirmation message to close it.

The imported regular expression is listed under the **Patterns > Regex** folders in the **DQ Repository** tree view.



9.1.4.8. How to duplicate a regular expression or an SQL pattern

To avoid creating a regular expression or an SQL pattern from scratch, you can duplicate an existing one and work around its metadata to have a new regular expression or SQL pattern to be used in data profiling analyses.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To duplicate a regular expression or an SQL pattern, do the following:

1. In the **DQ Repository** tree view, expand the **Libraries** and the **Patterns** folders in succession.

- 2. Browse through the regular expression/SQL pattern lists to reach the expression/pattern you want to duplicate.
- 3. Right-click its name and select **Duplicate...** from the contextual menu.

The duplicated regular expression/SQL pattern is displayed under the **Regex/SQL** folder in the **DQ Repository** tree view.

You can now double-click the duplicated pattern to modify its metadata as needed.



You can test new regular expressions before start using them against data in the specified database. For more information, see Section 9.1.4.2, "How to test a regular expression in the Pattern Test View".

9.1.4.9. How to delete a regular expression or an SQL pattern

You can delete regular expressions or SQL patterns directly from the **Analyzed Columns** view or from the **DQ Repository** tree view.

How to delete a regular expression or an SQL pattern from the analyzed column

Prerequisite(s): A column analysis is open in the analysis editor in Talend Open Studio for Data Quality.

To delete a regular expression or an SQL pattern from the analyzed column, do the following:

- 1. Click **Analyze Columns** to display the analyzed columns view.
- 2. Right-click the regular expression/SQL pattern you want to delete and select **Remove Elements** from the contextual menu.

 Analyzed Columns 				
Connection: MySQL				
Select columns to analyze				
Select indicators for each column				
Analyzed Columns	Datamining Type	Pattern	UDI	Operation
🖃 🚦 fullname (varchar)	Nominal 💌	<u>k</u>		×
Row Count	(j)			×
gender_validation_F	<u>a</u>			×
🖳 🗄 email 🔍 View executed query	31 💌	<u> 4</u>	A	X
🖮 🗄 addri 🧟 Edit Pattern	al 💌	4	A	×
💠 Add Task				
🔀 Remove elements				

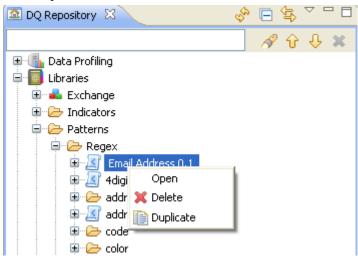
The selected regular expression/SQL pattern disappears from the Analyzed Column list.

How to delete and restore a regular expression or an SQL pattern from the DQ Repository

Prerequisite(s): Talend Open Studio for Data Quality is open.

To delete a regular expression or an SQL pattern from the **DQ Repository** tree view, do the following:

- 1. In the **DQ Repository** tree view, expand the **Libraries** and the **Patterns** folders in succession.
- 2. Browse to the regular expression or SQL pattern you want to remove from the list.
- 3. Right-click the expression or pattern and select **Delete** from the contextual menu.



The regular expression or SQL pattern is moved to the Recycle Bin.

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

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

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

📽 Delete forever 🛛 🔀
_Email_Address will be deleted forever (cannot be retrieved). Continue anyway?
Yes No

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

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

📽 Cannot delete this resource 🛛 🛛 🔀
Other items depend on item("Email Address").
Force to delete all the dependencies.
Set_of_Columns
ОК

- 3. Either:
 - Click **OK** to close the dialog box without deleting the pattern from the recycle bin.
 - Select the **Force to delete all the dependencies** check box and then click **OK** to delete the pattern from the recycle bin and to delete all the dependent analyses from the **Data Profiling** node.

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

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

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

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

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

🗳 Cannot delete this resource	×
You can't empty the Recycle Bin. Other items depend on these ones. Please delete them one by one.	
Set_of_Columns Image:	
ОК	

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

To restore a pattern from the Recycle Bin, do the following:

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

The pattern is moved back to the Libraries node.

9.2. Indicators

Indicators can be the results achieved through the implementation of different patterns that are used to define the content, structure and quality of your data.

Indicators represent as well the results of highly complex analyses related not only to data-matching, but also to different other data-related operations.

9.2.1. Indicator types

Talend Open Studio for Data Quality lists two types of indicators under the **Indicators** folder in the **DQ Repository** tree view: system indicators and user-defined indicators.

User-defined indicators, as their name indicates, are indicators created by the user. You can use them through a simple drag-and-drop operation from the **DQ Repository** tree view. User-defined indicators are used only with column analyses. For more information on how to set user-defined indicators for columns, see the section called "How to set user-defined indicators".

System indicators are predefined indicators that can be used on any of the analysis types available in *Talend Open Studio for Data Quality*. The Studio automatically uses a system indicator on the corresponding analysis type.

It is not possible to create a system indicator or to drag it directly from the **DQ Repository** tree view to an analysis. However, you can open and modify the parameters of a system indicator according to a specific database.

Several management options including editing, duplicating, importing and exporting are possible for both types of indicators. For more information, see Section 9.2.3, "Managing user-defined indicators" and Section 9.2.2, "Managing system indicators".

The below sections describe the system indicators used only on column analyses. These system indicators can range from simple or advanced statistics to text strings analysis, including summary data and statistical distributions of records.



You can see under the **System Indicators** folder in the **DQ Repository** tree view system indicators other than the indicators in the below sections. Those different system indicators are used on the other analysis types, for example redundancy, correlation and overview analyses.

9.2.1.1. Simple statistics

They provide simple statistics on the number of records falling in certain categories including the number of rows, the number of null values, the number of distinct and unique values, the number of duplicates, or the number of blank fields.

• Blank Count: counts the number of blank rows. A "blank" is a non null textual data that contains only white space. Note that Oracle does not distinguish between the empty string and the null value.

- Default Value Count: counts the number of default values.
- Distinct Count: counts the number of distinct values of your column.
- Duplicate Count: counts the number of values appearing more than once. You have the relation: Duplicate count + Unique count = Distinct count. For example, a,a,a,b,b,c,d,e => 9 values, 5 distinct values, 3 unique values, 2 duplicate values.
- Null Count: counts the number of null rows.
- Row Count: counts the number of rows.
- Unique Count: counts the number of distinct values with only one occurrence. It is necessarily less or equal to Distinct counts.

9.2.1.2. Text statistics

They analyze the characteristics of textual fields in the columns, including minimum, maximum and average length.

- Minimal Length: computes the minimal length of a text field.
- Maximal Length: computes the maximal length of a text field.
- Average Length: computes the average length of a field.

Other text indicators are available to count each of the above indicators with null values, with blank values or with null and blank values.

Null values will be counted as data of 0 length, i.e. the minimal length of null values is 0. This means that the **Minimal Length With Null** and the **Maximal Length With Null** will compute the minimal/maximal length of a text field including null values.

Blank values will be counted as data of 0 length, i.e. the minimal length of blank values is 0. This means that the **Minimal Length With Blank** and the **Maximal Length With Blank** will compute the minimal/maximal length of a text field including blank values.

The same will be applied for all average indicators.

The below table gives an example of computing the length of few textual fields in a column using all different types of text statistic indicators.

Data	Current length	With blank values	With null values	With blank and null values	
Brayan	6	6	6	6	
Ava	3	3	3	3	
	1	0	1	0	
	0	0	0	0	
Null	—	—	0	0	
Minimal, Maximal and Average lengths					
Minimal length	0	0	0	0	
Maximal length	6	6	6	6	
Average length	9/4 = 2.25	8/4 = 2	9/5 = 1.8	8/5 = 1.6	

9.2.1.3. Summary statistics

They perform statistical analyses on numeric data, including the computation of location measures such as the median and the average, the computation of statistical dispersions such as the inter quartile range and the range.

- Mean: computes the average of the records.
- Median: computes the value separating the higher half of a sample, a population, or a probability distribution from the lower half.
- Inter quartile range: computes the difference between the third and first quartiles.
- Range: computes the difference between the highest and lowest records.

9.2.1.4. Advanced statistics

They determine the most probable and the most frequent values and builds frequency tables. The main advanced statistics include the following values:

- Mode: computes the most probable value. For numerical data or continuous data, you can set bins in the parameters of this indicator. It is different from the "average" and the "median". It is good for addressing categorical attributes.
- Frequency table: computes the number of most frequent values for each distinct record. Other frequency table indicators are available to aggregate data with respect to "date", "week", "month", "quarter", "year" and "bin".
- Low frequency table: computes the number of less frequent records for each distinct record. Other low frequency table indicators are available for each of the following values: "date", "week", "month", "quarter", "year" and "bin" where "bin" is the aggregation of numerical data by intervals.

9.2.1.5. Pattern frequency statistics

Indicators in this group determine the most and less frequent patterns.

- Pattern frequency table: computes the number of most frequent records for each distinct pattern.
- Pattern low frequency table: computes the number of less frequent records for each distinct pattern.
- Date pattern frequency table: retrieves the date patterns from date or text columns. It works only with the Java engine.

9.2.1.6. Soundex frequency statistics

Indicators in this group use the Soundex algorithm built in the DBMS.

They index records by sounds. This way, records with the same pronunciation (only English pronunciation) are encoded to the same representation so that they can be matched despite minor differences in spelling.

- Soundex frequency table: computes the number of most frequent distinct records relative to the total number of records having the same pronunciation.
- Soundex low frequency table: computes the number of less frequent distinct records relative to the total number of records having the same pronunciation.

9.2.1.7. Phone number statistics

Indicators in this group count phone numbers. They return the count for each phone number format. They validate the phone formats using the *org.talend.libraries.google.libphonumber* library.

- Valid phone number count: computes the valid phone numbers.
- Possible phone number count: computes the supposed valid phone numbers.
- Valid region code number count: computes phone numbers with valid region code.
- Invalid region code count. computes phone numbers with invalid region code.
- Well formed national phone number count: computes well formatted national phone numbers.
- Well formed international phone number count: computes the international phone numbers that respect the international phone format (phone numbers that start with the country code).
- Well formed E164 phone number count: computes the international phone numbers that respect the international phone format (maximum of fifteen digits written with a + prefix.
- Format Frequency Pie: shows the results of the phone number count in a pie chart divided into sectors.

9.2.2. Managing system indicators

System indicators are predefined indicators that are automatically used on the relevant analyses. For more information on the system indicators available in *Talend Open Studio for Data Quality*, see Section 9.2.1, "Indicator types".

The management options available for system indicators include: setting indicators to column analyses, export/ import and edit functions. For detailed information, see the following sections.

9.2.2.1. How to set system indicators and indicator options to column analyses

You can define system indicators and indicator parameters for columns of database tables that need to be analyzed or monitored. For more information, see Section 5.3.1.2, "How to set indicators for the column(s) to be analyzed" and the section called "How to set options for system indicators".

9.2.2.2. How to export or import system indicators

You can export system indicators to folders or archive files and import them again in *Talend Open Studio for Data Quality* on the condition that the export and import operations are done in compatible versions of the Studio. For further information, see Section 10.3, "Exporting data profiling items" and Section 10.2, "Importing data profiling items".

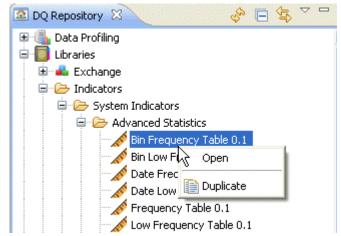
9.2.2.3. How to edit a system indicator

You can open the editor of any system indicator to check its settings and/or edit its definition and metadata in order to adapt it to a specific database type or need.

Prerequisite(s):*Talend Open Studio for Data Quality* is open.

To edit a system indicator, do the following:

- 1. In the **DQ Repository** tree view, expand the **Libraries** and the **Indicators** folders in succession and browse through the indicator lists to reach the indicator you want to modify.
- 2. Right-click the indicator name and select **Open** from the contextual menu.



The indicator editor opens displaying the selected indicator parameters.

🔶 Mode 🗙	- 8
: 🖻 🖻 🔛	
Indicator	Settings
 Indicator 	
Set the prope	rties of User Defined Indicator.
Name:	Mode
Purpose:	evaluates the most frequent value
Description:	computes the mode (most probable value). For numerical data or continuous data, bins can
Author:	
Status:	Draft 🔽
▼ Indicator	Definition
	definition of your indicator specific to a database. If the expression is simple enough to be us
ALL_DATA	BASE_TYPE SELECT <%=_COLUMN_NAMES_%>, COUNT(*) FROM <%=_T
Oracle	SELECT <%=COLUMN_NAMES%>, COUNT(*) FROM <%=T
DB2	SELECT <%=COLUMN_NAMES%>, COUNT(*) FROM <%=T
MySQL	SELECT <%=_COLUMN_NAMES%>, COUNT(*) c FROM <%='
	+

3. Modify the indicator metadata, if required, and then click **Indicator Definition** to display the relevant view. In this view, you can edit the indicator definition, change the selected database and add other indicators specific to available databases using the [+] button.

4. Click the save icon on top of the editor to save your changes.

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



When you edit an indicator, you modify the indicator listed in the DQ Repository tree view. Make sure that your modifications are suitable for all analyses that may be using the modified indicator.

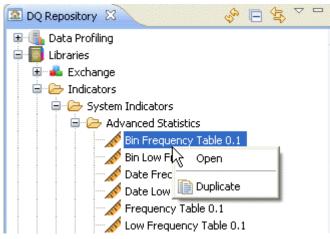
9.2.2.4. How to duplicate a system indicator

To avoid creating a system indicator from scratch, you can duplicate an existing one in the indicator list. once the copy is created, you can work around its metadata to have a new indicator and use it in data profiling analyses.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To duplicate a system indicator, do the following:

- 1. In the **DQ Repository** tree view, expand the **Libraries** and the **Indicators** folders in succession.
- 2. Browse through the indicator lists to reach the indicator you want to duplicate, right-click its name and select **Duplicate...** from the contextual menu.



The duplicated indicator is displayed under the System folder in the DQ Repository tree view.

You can now open the duplicated indicator to modify its metadata and definition as needed. For more information on editing system indicators, see Section 9.2.2.3, "How to edit a system indicator".

9.2.3. Managing user-defined indicators

User-defined indicators, as their name indicates, are indicators created by the user himself/herself. You can use these indicators to analyzed columns through a simple drag-and-drop operation from the **DQ Repository** tree view to the analyzed columns.

The management options available for user-defined indicators include: create, export and import, edit and duplicate. For detailed information, see the following sections.

9.2.3.1. How to create SQL user-defined indicators

Talend Open Studio for Data Quality enables you to create your own personalized indicators.

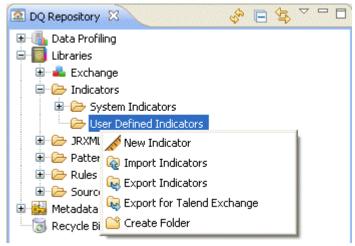


Management processes for user-defined indicators are the same as those for system indicators.

Prerequisite(s): Talend Open Studio for Data Quality is open.

Procedure 9.1. Defining the indicator

- 1. In the DQ Repository tree view, expand the Libraries and Indicators folders in succession.
- 2. Right-click User Defined Indicators.



3. Select **New Indicator** from the contextual menu.

The [New Indicator] wizard is displayed.

😤 New Inc	dicator 📃 🗖 🔀
User Defir your input is	ned Indicator Creation Page 1/2 s valid.
Name	Simple_Count
Purpose	
Description	
Author	user@company.com
Status	development 💌
Path	/TDQ/TDQ_Libraries/Indicators/User Defined Ind Select
?	< Back Next > Finish Cancel

4. In the **Name** field, enter a name for the indicator you want to create.

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

📽 New Indicator	
User Defined Ind	icator Creation Page 2/2
Language Selection: SQL Template	ALL_DATABASE_TYPE
0	< Back Next > Finish Cancel

- 5. From the Language Selection list, select the database that will support the created indicator.
- 6. In the **SQL Template** field, enter the SQL template statement corresponding to the indicator you want to create and then click **Finish** to close the wizard and proceed to the next step.

The indicator editor opens displaying the metadata of the user-defined indicator.

💉 Simple_Coun	t 0.1 🛛		
Indicator	Settings		
 Indicator Set the property 	Metadata erties of User Define	l Indicator.	
Name:	Simple_Count		
Purpose:			
Description:			
Author:	user@company.co	n	
Status:	development 🔽		
 Indicator Add here the 		icator specific to a database. If the expression is simple enough to be used in "ALL_DATABASE	_TYPE
Database		Version SQL Template	
ALL_DAT/	ABASE_TYPE	SELECT COUNT(*)FROM<%=TABLE_NAME%> <%=WHERE_CLAUSE	%>
		+	
▼ Indicator	and the second		
This section is	for indicator categ	у.	
User Define	d Count 🛛 🔽	Purpose:analyze the quantity of records	
		Description:contains user defined indicators which return a row count. The result set expected l	from

Procedure 9.2. Setting the indicator definition and category

1. In the editor, click **Indicator Definition** to display the corresponding view.

- 2. If required, change the selected database or click the **Edit...** button to the right of the view to edit the indicator definition.
- 3. If required, click the [+] button and add other indicators specific to available databases.
- 4. Enter the database version in the **Version** field.
- 5. Click **Indicator Category** to display the corresponding view. In this view, you can select from the list a category for the created indicator. The selected category will determine the type of chart that will represent the results of the executed analysis that uses the created indicator.
- 6. From the Indicator Category list, select a category for the created indicator.

The table below explains available categories.

Indicator category	Description
User Defined Match (by- default category)	Uses user-defined indicators to evaluate the number of the data records that match a regular expression or an SQL pattern. The analysis results show the record matching count and the record total count.
User Defined Frequency	Uses user-defined indicators for each distinct data record to evaluate the record frequency that match a regular expression or an SQL pattern. The analysis results show the distinct count giving a label and a label-related count.
User Defined Real Value	Uses user-defined indicators which return real value to evaluate any real function of the data.
User Defined Count	Uses user-defined indicators that return a row count.

7. Click the save icon on top of the editor.

The created indicator is listed under the User Defined Indicators folder in the DQ Repository tree view.

9.2.3.2. How to define Java user-defined indicators

Talend Open Studio for Data Quality enables you to create your own personalized Java indicators. Management processes for Java user-defined indicators are the same as those for system indicators.

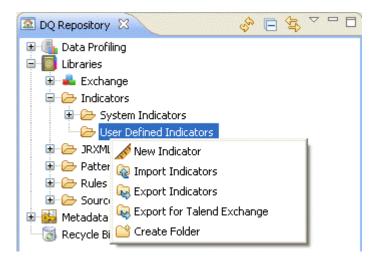
The two sections below detail the procedures to create Java user-defined indicators.

How to create Java user-defined indicators

Prerequisite(s): Talend Open Studio for Data Quality is open.

Procedure 9.3. Defining the indicator

- 1. In the **DQ Repository** tree view, expand the **Libraries** and **Indicators** folders in succession.
- 2. Right-click User Defined Indicators.



3. Select **New Indicator** from the contextual menu.

The [New Indicator] wizard is displayed.

😤 New Inc	dicator 📃 🗖 🔀
User Defin your input is	ned Indicator Creation Page 1/2 s valid.
Name	Java_Indicator
Purpose	
Description	~
Author	user@company.com
Status	development
Path	/TDQ/TDQ_Libraries/Indicators/User Defined Ind Select
?	< Back Next > Finish Cancel

- 4. In the **Name** field, enter a name for the Java indicator you want to create.
- 5. If required, set other metadata (purpose, description and author name) in the corresponding fields and click **Next** to proceed to the next step.

📽 New Indicator	
User Defined IndicatorCreation Page 2/2 your input is valid.	
Language Selection: Java SQL Template:	→
Contract	Cancel

6. From the Language Selection list, select Java and then click Finish to open the indicator settings.

i 🖻 🕀 🔡	
Indicator	Settings
▼ Indicator	Metadata
Set the prope	erties of User Defined Indicator.
Name:	Java_Indicator
Purpose:	
Description:	
Author:	user@company.com
Status:	development 💽
▼ Indicator	Definition
Add here the	definition of your indicator specific to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerate.
Database	Version SQL Template
Java	test.UDI.MyAvgLength C:\Documents and Settings\hmassy\Mes documents\Downloads\myUDI.jz Browse 🗶
	+
Indicator	Category
Indicator	Parameters

The indicator editor opens displaying the metadata of the Java indicator.

Procedure 9.4. Setting the indicator definition and category

- 1. In the editor, click Indicator Definition to display the corresponding view. Java is selected by default.
- 2. Click the browse button to the right of the view and browse to the Java archive holding the Java classes. For more information on creating a Java archive, see the section called "How to create a Java archive for the user-defined indicator".
- 3. Enter the Java class in the **Version** field.

Make sure that the class name includes the package path, if this string parameter was not correctly specified, an error message will display when you try to save the Java user-defined indicator.

4. Click **Indicator Category** to display the corresponding view.

Indicator Definition	
Indicator Category This section is for indicator category	jory.
User Defined Count	Purpose:analyze the quantity of records Description:contains user defined indicators which return a row count

5. From the Indicator Category list, select a category for the created Java indicator.

The selected category will determine the type of chart that will represent the results of the executed analysis that uses the created Java indicator.

Indicator category	Description	
User Defined Count	Uses user-defined indicators that return a row count.	
User Defined Real Value	Uses user-defined indicators which return real value to evaluate any real function of the data.	
User Defined Match (by- default category)	Uses user-defined indicators to evaluate the number of the data records that match a regular expression or an SQL pattern. The analysis results show the record matching count and the record total count.	
User Defined Frequency	Uses user-defined indicators for each distinct data record to evaluate the record frequency. The analysis results show the distinct count giving a label and a label-related count.	

- 6. Click **Indicator Parameter** to display the corresponding view.
 - Indicator Definition
 - Indicator Category
 - Indicator Parameters

This section is for indicator parameters.

Parameters Key	Parameters Value
Lang	EN
paraKey	paraValue
ſ	(¥
	+ ×

In this table, you can set the default parameters for this new Java indicator. These default parameters are stored in a *Map* object.

- 7. Click the [+] button at the bottom of the table to add as many lines as needed and define the parameter key and value.
- 8. Click in the line and define the parameter key and the parameter value.



You can edit these default parameters or even add new parameters any time you use the indicator in a column analysis. To do this, click the indicator option icon in the analysis editor to open a dialog box where you can edit the default parameters according to your needs or add new parameters.

9. Click the save icon on top of the editor.

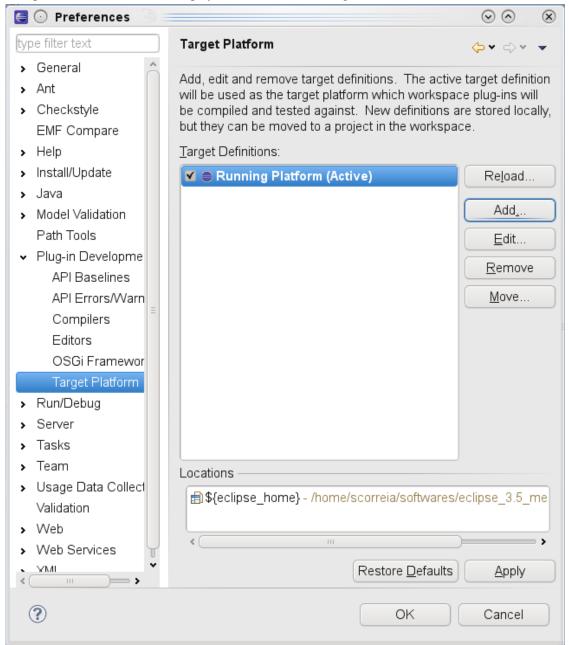
The created indicator is listed under the User Defined Indicators folder in the DQ Repository tree view.

How to create a Java archive for the user-defined indicator

Before creating a Java archive for the user defined indicator, you must define, in Eclipse, the target platform against which the workspace plug-ins will be compiled and tested.

To define the target platform, do the following:

1. In Eclipse, select **Preferences** to display the [**Preferences**] dialog box.



2. Expand **Plug-in Development** and select **Target Platform** then click **Add...** to open a view where you can create the target definition.

🥃 📀 New Target Definition	\odot	\otimes
Target Definition		
Create a new target definition.	\odot	
Initialize the target definition with:		
Nothing: Start with an empty target definition		
Default: Default target for the running platform		
O Current Target: Copy settings from the current target platform		
○ <u>T</u> emplate: Base RCP (Binary Only) \$		
< Back Next > Einish	Cancel	
	Cancer	

3. Select the **Nothing: Start with an empty target definition** option and then click **Next** to proceed to the next step.

Sew Target Definition	\odot	\otimes
Target Content		
Edit the name, description, and plug-ins contained in a target.	\odot	
Name: TalendOpenProfiler		
Locations Content Environment Arguments Implicit Dependencies		
The following list of locations will be used to collect plug-ins for this target definition.		
	<u>A</u> dd	
	<u> </u>	
	<u>R</u> emove	
	Re <u>m</u> ove Al	
Show Plug-in Content		
< Back	Cancel	

4. In the **Name** field, enter a name for the new target definition and then click the **Add...** button to proceed to the next step.

E 🖸 Add Content	\odot	\otimes
Add Content		
Select a source of plug-ins.		
🔁 Directory		
Installation		
🌆 Features		
🖏 Software Site		
An installation (such as an Eclipse SDK) in the local file system. The installed be added to the target definition.	d plug-ins v	vill
(?) < <u>Back</u> <u>Next</u> > <u>Einish</u>	Cancel	

5. Select **Installation** from the **Add Content** list and then click **Next** to proceed to the next step.

들 📀 Add Content 🕓 🖃				\odot \odot \otimes
Add Installation				
Select an installation directo	iry.			
Location: /home/scorreia/DE	MOS/TOP-A	ll-r39896-V4	.0.0	~
			Browse	
✓ This install uses the <u>d</u> efail	ult configuratio	on area.		
<u>C</u> onfiguration:				~ .
			Br <u>o</u> wse	Variables
?	<u>B</u> ack	<u>N</u> ext >	<u> </u>	Cancel

6. Use the **Browse...** button to set the path of the installation directory and then click **Next** to proceed to the next step.

The new target definition is displayed in the location list.

E 🖸 New Target Definition	\odot \otimes \otimes
Target Content Edit the name, description, and plug-ins contained in a target.	
Name: TalendOpenProfiler	
Locations Content Environment Arguments Implicit Dependencies The following list of locations will be used to collect plug-ins for this target definition.	
	Add Edit Remove Remove All
Show Plug-in Content	
(?) < <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

7. Click **Finish** to close the dialog box.

To create a Java archive for the user defined indicator, do the following:

1. In Eclipse, check out the project from svn at http://talendforge.org/svn/top/branches/branch-4_0/test.myudi.

In this Java project, you can find four Java classes that correspond to the four indicator categories listed in the **Indicator Category** view in the indicator editor.

<u>MyAvgLength.java</u>
<u>MyFrequencyUDI.java</u>
<u>MyNotNutlMatchingUDI.java</u>
MyNotNutlUDI.java

Each one of these Java classes extends the UserDefIndicatorImpl indicator. The figure below illustrates an example using the MyAvgLength Java class.

```
package test.udi;
import org.talend.dataquality.indicators.sql.impl.UserDefIndicatorImpl;
/**
 * @author mzhao
 * A very simple example of a java implementation of a user defined indicator. This in
 * real value. It implements the minimum number of required methods.
 */
public class MyAvgLength extends UserDefIndicatorImpl {
    private double length = 0;
    00verride
    public boolean reset() {
        super.reset();
        length = 0;
        return true;
    3
    @Override
    public boolean handle(Object data) {
        super.handle(data);
        // an indicator which computes the average text length on data which are more
        // text values with less than 2 characters are not taken into account).
        int dataLength = (data != null) ? data.toString().length() : 0;
        if (dataLength > 2) {
             length += dataLength;
        }
        return true;
    }
    /*
     *
       (non-Javadoc)
     * @see org.talend.dataquality.indicators.impl.IndicatorImpl#finalizeComputation()
     */
    @Override
    public boolean finalizeComputation() {
        value = String.valueOf(this.length / (this.getCount() - this.getNullCount()));
        return super.finalizeComputation();
    3
   2.
       Modify the code of the methods that follow each @Override according to your needs.
   3.
       If required, use the following methods in your code to retrieve the indicator parameters:
   4.
      use Indicator.getParameter() which returns an IndicatorParameters object.
   5.
       call IndicatorParameters.getIndicatorValidDomain() which returns a Domain object.
                                                                               list
                                                                                     of
   6.
       call
             Domain.getJavaUDIIndicatorParameter()
                                                          which
                                                                  returns
                                                                          а
       JavaUDIIndicatorParameter that stores each key/value pair that defines the parameter.
   7. Save your modifications.
```

```
8. Using Eclipse, export this new Java archive.
```

The Java archive is now ready to be attached to any Java indicator you want to create in *Talend Open Studio for Data Quality*.

9.2.3.3. How to export user-defined indicators

Talend Open Studio for Data Quality enables you to export user-defined indicators to a local csv file or to **Talend Exchange** to be shared with other users.

You can also export user-defined indicators to folders or archive files. For further information, see Section 10.3, "Exporting data profiling items".



You can only export user-defined indicators based on SQL templates. It is not possible to export Java user-defined indicators.

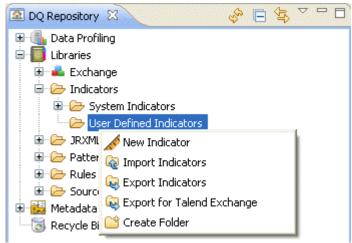
How to export user-defined indicators to a csv file

In Talend Open Studio for Data Quality you can export user-defined indicators and store them locally in a csv file.

Prerequisite(s): At least one user-defined indicator is created in Talend Open Studio for Data Quality.

To export user-defined indicators to a csv file, do the following:

1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators** and then right-click **User Defined Indicators**.



2. From the contextual menu, select **Export Indicators**.

The [Export Indicators] wizard opens with the check boxes of all indicators selected by default.

📽 Export Indicators	
Export User-Defined Indicators	
Choose a folder to export indicators.	
Select File (csv):	Browse
Selected indicators:	
€ KO	
Java_Indicator 0.1	
simple_count 0.1	
row_comparison 0.1	
user-defined 0.1	
user_defined_indicator 0.1	
Select All Deselect All	
Deselect All	
? Finis	h Cancel

- 3. Browse to the csv file where to save the indicators.
- 4. If required, clear the check boxes of the indicators you do not want to export to the csv file.
- 5. Click **Finish** to close the wizard.

All exported user-defined indicators are saved in the defined csv file.

How to export user-defined indicators to Talend Exchange

You can export user-defined indicators from your current version of *Talend Open Studio for Data Quality* to **Talend Exchange** where you can share them with other users.

Prerequisite(s): At least one user-defined indicator is created in Talend Open Studio for Data Quality.

To export user-defined indicators to Talend Exchange, do the following:

- 1. In the **DQ Repository** tree view, expand **Libraries** > **Indicators** in succession.
- 2. Right-click the User Defined Indicator folder and select Export for Talend Exchange.

The [Export for Talend Exchange] wizard is displayed.

📽 Export for Talend Exchange 📃 🗖 🔀				
Export Indicators for Talend Exchange Choose a folder where to export the indicators.				
Select a folder: D:\output Browse Selected indicators: Select a folder: Select a folder: Image: Select All row_comparison 0.1 Select All Select All Deselect All				
? Finish Cancel				

- 3. Browse to the folder where to save indicators.
- 4. If required, clear the check boxes of the indicators you do not want to export to the specified folder.
- 5. Click **Finish** to close the wizard.

A distinct csv file is created for each exported indicator. Each csv file is compressed as a zip. All these zip files are saved in the defined folder. You need now to upload them to **Talend Exchange**at http://www.talendforge.org/exchange/top/help_guest.php.

9.2.3.4. How to import user-defined indicators

From *Talend Open Studio for Data Quality*, you can import indicators from a local csv file or from **Talend Exchange** and use them, as needed, on your column analyses.

You can also import user-defined indicators from folders or archive files. For further information, see Section 10.2, "Importing data profiling items".

How to import user-defined indicators from a csv file

In *Talend Open Studio for Data Quality* you can import indicators stored locally in a csv file to use them on your column analyses.

Prerequisite(s):*Talend Open Studio for Data Quality* is open. The csv file is stored locally.

To import user-defined indicators from a csv file, do the following:

- 1. In the DQ Repository tree view, expand Libraries and Indicators in succession.
- 2. Right-click User Defined Indicators and select Import Indicators.

The [Import Indicators] wizard opens.

💰 Import Indicate	ors				
Import Indicators from File Choose a file to import indicators from.					
Select File : D:\out Duplicate indicators Skip existing indir rename new indir Preview:	ators			Browse	
Label Simple_Count Frequency Table o copy of Regular Ex		Description counts the nu counts the nu	Author user@ scorreia	Relative_I	
Simple_Count user_defined_indic	a		user@ user@	>	
? Finish Cancel					

- 3. Browse to the csv file holding the user-defined indicators.
- 4. In the **Duplicate patterns handling** area, select:

Option	То
	import only the indicators that do not exist in the corresponding lists in the DQ Repository tree view. A warning message is displayed if the imported indicators already exist under the Indicators folder.
rename new indicators with suffix	identify each of the imported indicators with a suffix. All indicators will be imported even if they already exist under the Indicators folder.

5. Click **Finish** to close the wizard.

All imported indicators are listed under the User Defined Indicators folder in the DQ Repository tree view.

A warning icon (a) next to the name of the imported user-defined indicator in the tree view identifies that it is not correct. You must open the indicator and try to figure out what is wrong.

How to import user-defined indicators from Talend Exchange

You can import user-defined indicators created by other users and stored in **Talend Exchange** into your current version of *Talend Open Studio for Data Quality* and use them, as needed, on your column analyses.



9

Make sure that the Talend Exchange extension you want to import from is compatible with your current Studio version. Compatibility means that Talend Exchange extension has the same two first sequences of the unique identifier of your current version of Talend Open Studio for Data Quality. For example, *if your current version of* Talend Open Studio for Data Quality *is 3.2.0, compatible extensions could be 3.2.0M1, 3.2.0M2, 3.2.0RC1, 3.2.1, etc.*

Prerequisite(s): Talend Open Studio for Data Quality is open. Your network is up and running.

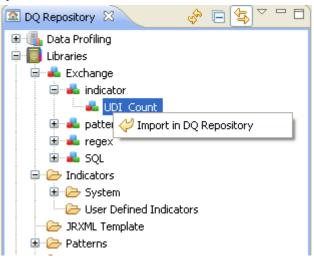


If you have connection problems, you will not be able to access any of the regular expressions or SQL patterns under the **Exchange** node in the **DQ Repository** tree view.



To import user-defined indicators from Talend Exchange, do the following:

- 1. In the DQ Repository tree view, expand Libraries and Exchange in succession.
- 2. Under Exchange, expand indicator and right-click the indicator name you want to import and then select Import in DQ Repository.



A message is displayed to confirm the operation.

3. Click **OK** in the confirmation message to close it.

The indicator imported from **Talend Exchange** is listed under the **User Defined Indicators** folder in the **DQ Repository** tree view.

9.2.3.5. How to edit a user-defined indicator

You can open the editor of any system or user-defined indicator to check its settings and/or edit its definition and metadata in order to adapt it to a specific database type or need, if required.

Prerequisite(s): At least one user-defined indicator is created in Talend Open Studio for Data Quality.

To edit the definition of a user-defined indicator, do the following:

1. In the **DQ Repository** tree view, expand the **Libraries** and **Indicators** folders in succession and browse through the indicator lists to reach the indicator you want to modify the definition of.

2. Right-click the indicator name and select **Open** from the contextual menu.

The indicator editor opens displaying the selected indicator settings.

: 🖻 🕀 🔛	
Indicator	Settings
▼ Indicator	Metadata
Set the prope	rties of User Defined Indicator.
Name:	Frequency Table of hours
Purpose:	evaluates the most frequent hours appearing in a timestamp column
Description:	counts the number of records for each distinct hour. Can be used to analyze the repartition of data in the day.
Author:	scorreia
Status:	Draft 💽
Indicator Add here the Database MySQL Indicator	definition of your indicator specific to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerated Version SQL Template SELECT HOUR(<%=COLUMN_NAMES%>) h, COUNT(*) c FROM <%=TABLE_NAME%> t
	Category for indicator category.
User Defined	

3. Modify the indicator metadata, if required, and then click **Indicator Definition** to display the relevant view.

In this view, you can: edit indicator definition, change the selected database and add other indicators specific to available databases using the [+] button.



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

- 4. Click **Indicator Category** to display the relevant view.
- 5. If required, change the indicator category in the list.

The table below describes the different categories.

Indicator category	Description
User Defined Match (by- default category)	Uses user-defined indicators to evaluate the number of the data records that match a regular expression or an SQL pattern. The analysis results show the record matching count and the record total count.
User Defined Frequency	Uses user-defined indicators for each distinct data record to evaluate the record frequency that match a regular expression or an SQL pattern. The analysis results show the distinct count giving a label and a label-related count.
User Defined Real Value	Uses user-defined indicators which return real value to evaluate any real function of the data.
User Defined Count	Uses user-defined indicators that return a row count.

6. Click the save icon on top of the editor to save your changes.



When you edit an indicator, you modify the indicator listed in the DQ Repository tree view. Make sure that your modifications are suitable for all analyses that may be using the modified indicator.

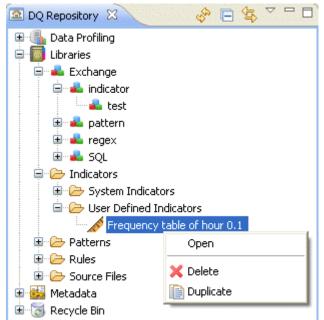
9.2.3.6. How to duplicate a user-defined indicator

To avoid creating an indicator from scratch, you can duplicate an existing one in the indicator list. Once the copy is created, you can work around its metadata to have a new indicator and use it in data profiling analyses.

Prerequisite(s): At least one user-defined indicator has been defined in Talend Open Studio for Data Quality.

To duplicate a user defined indicator, do the following:

- 1. In the DQ Repository tree view, expand the Libraries and the Indicators folders in succession.
- 2. Browse through the user-defined indicator lists to reach the indicator you want to duplicate, right-click its name and select **Duplicate...** from the contextual menu.



The duplicated indicator is displayed under the User Defined Indicators folder in the DQ Repository tree view.

You can now open the duplicated indicator to modify its metadata and definition as needed. for more information on editing user-defined indicators, see Section 9.2.3.5, "How to edit a user-defined indicator".

9.2.3.7. How to delete or restore a user-defined indicator

In the *Talend Open Studio for Data Quality*, you can delete a user-defined indicator definitely or restore it from the **Recycle Bin**. For details, refer to Section 9.1.4.9, "How to delete a regular expression or an SQL pattern".

9.2.4. Indicator parameters

This section describes indicator parameters displayed in the different Indicators Settings dialog boxes.

Bins Designer

Possible value	Description
Minimal value	Beginning of the first bin.
Maximal value	End of the last bin.
Number of bins	Number of bins.

Blank Options

Possible value	Description
Aggregate nulls with blanks	When selected, null data is counted as zero length text field. This means that null data is treated as an empty string. When not selected, null data is treated as any other text data.
Aggregate blanks	When selected, blank texts (e.g. " ") are all grouped together and considered as an empty string. When not selected, blank texts are treated as any other text data.
	In Oracle, empty strings and null strings are the same objects. Therefore, you must select or clear both check boxes in order to get consistent results.

Data Thresholds

Possible value	Description
Lower threshold	Data smaller than this value should not exist.
Upper threshold	Data greater than this value should not exist.

Frequency Table Parameters

Possible value	Description
Number of results shown	Number of displayed results.

Indicator Thresholds

Possible value	Description
Lower threshold	Lower threshold of matching indicator values.
Upper threshold	Higher threshold of matching indicator values.
Lower threshold(%)	Lower threshold of matching indicator values in percentage relative to the total row count.
Upper threshold(%)	Higher threshold of matching indicator values in percentage relative to the total row count.
Expected value	Only for the Mode indicator in the Advanced Statistics . Most probable value that should exist in the selected column.

Java Options

Possible value	Description
Characters to replace	List of the characters to be replaced.
Replacement characters	List of the characters that will take the place of the replaced characters. Each character of the first field will be replaced by the character at the same position from the second field. For example, with the values "abc0123ABC,;.:" in the first field and "aaa9999AAApppp" in the second field any "a", "b" or "c" will

Possible value	Description
	be replaced by "a" and any "0", "1", "2" or "3" will be replaced by "9".

Phone number

Possible value	Description
Country	Country ISO2 code of the phone number.

Text Parameters

Possible value	Description
Ignore case	When selected, comparison of text data is not case sensitive.

Text Length

Possible value	Description
Count nulls	When selected, null data is counted as zero length text field.
Count blanks	When selected, blank texts (e.g. " ") are counted as zero length text fields.



Chapter 10. Other important management procedures

This chapter provides the information you need to carry out some basic procedures including setting preferences of analysis editors and analysis results, creating SQL queries, setting data parser rules, importing/exporting data quality items and upgrading projects from older versions.

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

10.1. Creating and storing SQL queries via *Talend Open Studio for Data Quality*

Talend Open Studio for Data Quality allows to query and browse a selected database using the SQL Editor and then to store these SQL queries under the **Source Files** folder in the **DQ Repository** tree view. You can then open the SQL Editor on any of these stored queries to rename, edit or execute the query.

Prerequisite(s): Talend Open Studio for Data Quality is open.

To create an SQL query from Talend Open Studio for Data Quality, do the following:

1. In the **DQ Repository** tree view, expand **Libraries**.

🗟 DQ Repository 🔀		🔶 🕒	€ ₽	
🗉 🚯 Data Profiling				
😑 🐻 Libraries				
🕀 📥 Exchange				
😟 🗁 Indicators				
- 🗁 JRXML Tem	plate			
🗄 🗁 Patterns				
🕀 🗁 Rules				
🗄 🗁 Source File	ş			
🕀 🔛 Metadata	📸 Create SQL File			
🗄 👸 Recycle Bin	🧔 Import SQL			
	😂 Create Folder			

- 2. Right-click **Source Files** and select **Create SQL File** from the contextual menu. The **[Create SQL File]** dialog box is displayed.
- 3. In the **Name** field, enter a name for the SQL query you want to create and then click **Finish** to proceed to the next step.

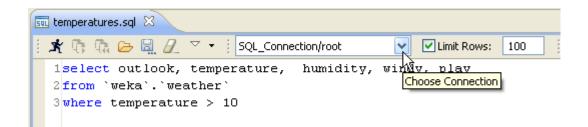
🗊 temperatures.sql 🛛 🗖 🗖	👼 Connections 🖾
★ Is Image: Connection/root ✓ SQL_Connection/root ✓ ✓	□
	MDM_Connection
	MSSQLServer MosQLServer MosQLServer MosQLServer SQL_Connection (2 sessions)
Messages 🛛	🗄 🗝 root
Status Location SQL Text	

The SQL Editor opens on the new SQL query.



If the **Connections** view is not open, use the combination **Window > Show View > Data Explorer > Connections** to open it.

- 4. Enter your SQL statement in the SQL Editor.
- 5. From the **Choose Connection** list, select the database you want to run the query on.

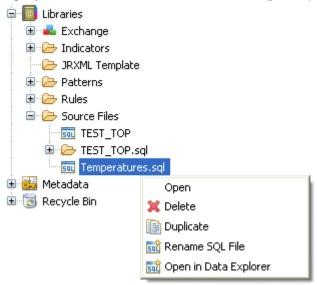


6. On the SQL Editor toolbar, click and then \mathbf{X} to execute the query on the defined base table(s).

			-		
l [select ou	utlook, temper.] 🔀 Me:	ssages		
outlook	temperature	humidity	windy	play	
sunny	25	high	FALSE	no	
sunny	25	high	TRUE	no	
overcast	25	high	FALSE	yes	
rainy	19	high	FALSE	yes	
sunny	19	high	FALSE	no	
rainy	19	normal	FALSE	yes	
sunny	19	normal	TRUE	yes	
overcast	19	high	TRUE	yes	
overcast	25	normal	FALSE	yes	
rainy	19	high	TRUE	no	
sunny	27	high	TRUE	no	
sunny	26	high	TRUE	no	
sunny	26	normal	TRUE	no	
sunny	26	normal	TRUE	yes	
sunny	29	normal	TRUE	yes	
sunny	29	normal	TRUE	yes	

Data rows are retrieved from the defined base table(s) and displayed in the editor.

A file for the new SQL query is listed under Source Files in the DQ Repository tree view.



7. Right-click an SQL file and from the contextual menu select:

Option	То
Open	open the selected Query file
Duplicate	create a copy of the selected Query file

Rename SQL File	open a dialog box where you can edit the name of the query file
Open in Data Explorer	open in the Data Explorer the SQL Editor on the selected query file
Delete	delete the query file
	The deleted item will go to the Recycle Bin in the DQ Repository tree view. You can restore or delete such item via a right-click on the item. You can also empty the recycle bin via a right-click on it.

When you open a query file in the SQL Editor, make sure to select the database connection from the **Choose Connection** list before executing the query. Otherwise the run icon on the editor toolbar will be unavailable.

When you create or modify a query in a query file in the SQL Editor and try to close the editor, you will be prompted to save the modifications. The modifications will not be taken into account unless you click the save icon on the editor toolbar.

10.2. Importing data profiling items

From *Talend Open Studio for Data Quality*, you can import the data profiling items (analyses, database connections, patterns and indicators, etc.) already created in other versions of *Talend Open Studio for Data Quality*.



You can not import into your current Studio data profiling items created in versions older than 4.0.0. To use such items in your current Studio, you must carry out an upgrade operation. For further information, see Section 10.4, "Upgrading projects items from older versions".

Prerequisite(s):*Talend Open Studio for Data Quality* is open. You have access to another Studio version in which data profiling items have been created.

To import one or more data profiling items, do the following:

1. Click the $\boxed{\mathbf{Q}}$ icon on the toolbar.

The [Import Item] wizard is displayed.

🗳 Import Item		
Import item to current proj	ect.	
 Select root directory: Select archive file: 	D:\TOP_builds\TO5_DQ-r81684-V5.1.0RC1\workspace\	Browse Browse
metadata metadata metadata TDQ_Data Prof metadata metadatata metadatata metadatata metadatatatatatatatatatatatatatatatatatat	nt_Backup_Comparison nalysis	
	ate_country_correlation	
✓ Overwrite existing items		
?	Finish	Cancel

- 2. Select the root directory or the archive file option according to whether the data profiling items you want to import are in the *workspace* file of the release directory or are already exported into a zip file.
- 3. If you select the root directory option, click **Browse** and set the path to the project folder in the release containing the items to be imported. All data profiling items that are listed in the dialog box and are selected by default.
- 4. If you select the archive file option, click **Browse** and set the path to the archive file that holds the exported data profiling items.

All data profiling items that are listed in the dialog box and are selected by default

9

If the data profiling items you try to import already exists in the Studio, they will be listed in the **Error and Warning** area in the **[Import Item]** wizard. Select the **Overwrite existing items** check box if you want to replace the existing items with the imported ones.

🙆 [ITEM]" 10.42.10	0.66 " conflict : the same item with different name exists!	~
🙆 [ITEM]" 400_con	nection " conflict : the same item with different name exists!	_
🙆 [ITEM]" Account[DataForDeDuplication " conflict : the same item with different name e	xists!
🙆 [ITEM]" address_	_template " conflict : the same item with different name exists!	~
<		>

- 5. If required, clear the check boxes of the data profiling items you do not want to import.
 - When you clear the check box of an analysis, the check boxes of the dependencies of this analysis (metadata connection, patterns and indicators) are automatically cleared as well and thus will not be imported. Also, an error message will display on top of the dialog box if you clear the check box of a metadata connection, any patterns or system indicators that are used in any of the analyses you want to import.
- 6. Click **Finish** to validate the operation.

The imported items display under the corresponding folders in the DQ Repository tree view.

10.3. Exporting data profiling items

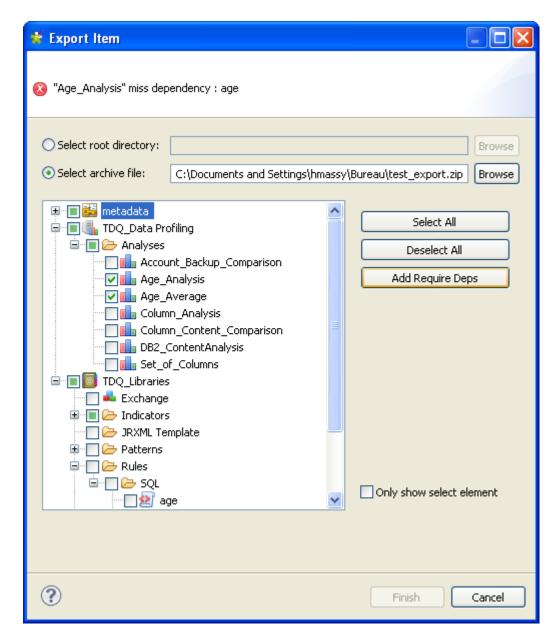
Talend Open Studio for Data Quality allows you to export the data profiling items (analyses, database connections, patterns and indicators, etc.) created or imported in the current instance of the Studio to folders or archive files.

Prerequisite(s): At least, one data profiling item has been created in Talend Open Studio for Data Quality.

To export data profiling items, do the following:

1. Click the \bigcirc icon on the toolbar.

The [Export Item] wizard is displayed.



- 2. Select the root directory or archive file option and then click **Browse...** and browse to the file/archive where you want to export the data profiling items.
- 3. Select the check boxes of the data profiling items you want to export or use the Select All or Deselect All tabs.

When you select an analysis check box, all analysis dependencies including the metadata connection and any patterns or system indicators used in this analysis are selected by default. Otherwise, if you have an error message on top of the dialog box that indicates any missing dependencies, click the **Add Require Deps** tab to automatically select the check boxes of all items necessary to use the exported data profiling analysis.

- 4. If required, select the **Only show selected elements** check box to have in the export list only the selected data profiling elements.
- 5. Click **Finish** to validate the operation.

A progress bar is displayed to indicate the progress of the export operation and the data profiling items are exported in the defined place.

10.4. Upgrading projects items from older versions

The below procedure concerns only the migration of data profiling items from versions older than 4.0.0. To migrate your data profiling items from version 4.0.0 onward, you simply need to import them into your current Studio or to import the project itself. For further information, see Section 10.2, "Importing data profiling items", and also see the section on importing projects in the Talend Open Studio for Data Integration User Guide.

To migrate data profiling items (analyses, database connections, patterns and indicators, etc.) created in versions older than 4.0.0, do the following:

- 1. From the folder of the old version studio, copy the workspace file and paste it in the folder of your current Studio. Accept to replace the current workspace file with the old file.
- 2. Launch the Studio connecting to this workspace.

The upgrade operation is completed once the Studio is completely launched, and you should have access to all your data profiling items.



Regarding system indicators during migration, please pay attention to the following:

- When you upgrade the repository items to version 4.2 from a prior version, the migration process overwrites any changes you made to the system indicators.
- When you upgrade the repository items from version 4.2 to version 5.0, you do not lose any changes you made to the system indicators.



Chapter 11. Managing existing analyses

This chapter provides the information you need to perform basic management procedures for all analysis created in *Talend Open Studio for Data Quality*.

Before starting data profiling management procedures, you need to be familiar with *Talend Open Studio for Data Quality* Graphical User Interface (GUI). For more information, see Appendix A, *Talend Open Studio for Data Quality management GUI*.

11.1. Procedures for all types of analyses

The procedures below provide detailed information on basic management options for all types of the analyses listed under the **Analyses** folder in the **DQ Repository** tree view.

From the contextual menu of the selected analysis, you can open, execute, duplicate or delete this analysis. You can also add a task to the selected analysis.

11.1.1. Opening an analysis

Prerequisite(s): At least one analysis has been created in Talend Open Studio for Data Quality.

To open an analysis, do the following:

- 1. In the DQ Repository tree view, expand the Data Profiling and the Analyses folders in succession.
- 2. Either:
 - · double-click the analysis you want to open, or,
 - right-click the analysis you want to open and select **Open** from the contextual menu.

The corresponding analysis editor is displayed.

- 3. If required, click **Refresh the graphics** to the right of the editor to display the results of the analysis.
- 4. If required, click the **Analysis results** button at the bottom of the editor to open a more detailed view of the analysis results.

11.1.2. Executing an analysis

Prerequisite(s): At least one analysis has been created in Talend Open Studio for Data Quality.

To execute an analysis, do the following:

- 1. In the DQ Repository tree view, expand the Data Profiling and Analyses folders in succession.
- 2. Right-click the analysis you want to execute and select **Run** from the contextual menu.

A progress bar is displayed to convey the progress of the analysis execution.



You can execute many analyses simultaneously if you select them, right-click the selection and finally click **Run**.

11.1.3. Duplicating an analysis

To avoid creating an analysis from scratch, you can duplicate an existing one in the **Analyses** folder and work around its metadata to have a new analysis.

Prerequisite(s): At least one analysis has been created in Talend Open Studio for Data Quality.

To duplicate an analysis, do the following:

- 1. In the **DQ Repository** tree view, expand the **Data profiling** and the **Analyses** folders in succession.
- 2. Right-click the analysis you want to duplicate and select **Duplicate...** from the contextual menu.

The duplicated analysis shows in the analysis list in the **DQ Repository** tree view. You can now open the duplicated analysis and modify its metadata as needed.

11.1.4. Adding a task to an analysis

You can add a task to an analysis to indicate a problem that needs to be solved later, for example.

For more information, see Section 11.2, "Managing tasks".

11.1.5. Deleting or restoring an analysis

Prerequisite(s): At least one analysis has been created in Talend Open Studio for Data Quality.

To delete an analysis, do the following:

- 1. In the DQ Repository tree view, expand the Data Profiling and Analyses folders in succession.
- 2. Right-click the analysis you want to delete and select **Delete** from the contextual menu.

The analysis is moved to the Recycle Bin.

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

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

A confirmation dialog box is displayed.

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

To restore an analysis from the Recycle Bin, do the following:

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

The analysis is moved back to the **Data profiling** node.

11.2. Managing tasks

In *Talend Open Studio for Data Quality*, it is possible to add tasks to different items, display the task list and delete any completed task from the task list.

From Talend Open Studio for Data Quality, you can add tasks to different items either:

- in the **DQ Repository** tree view on catalogs, schemas, tables, columns and created analyses,
- or, on columns, or patterns and indicators set on columns directly in the current analysis editor.

For example, you can add a general task to any item in a database connection via the **Metadata** node in the **DQ Repository** tree view. You can add a more specific task to the same item defined in the context of an analysis through the **Analyses** node. And finally, you can add a task to a column in an analysis context (also to a pattern or an indicator set on this column) directly in the current analysis editor.

The procedure to add a task to any of these items is exactly the same. Adding tasks to such items will list these tasks in the **Tasks** list accessible through the **Window > Show view...** combination. Later, you can open the editor corresponding to the relevant item by double-clicking the appropriate task in the **Tasks** list.

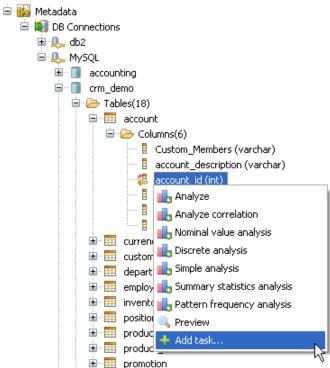
For examples on how to add a task to different items, see the sections below.

11.2.1. Adding a task to a column in a database connection

Prerequisite(s):At least, one database connection has been created in *Talend Open Studio for Data Quality*. For further information, see Section 3.1.1, "Connecting to a database".

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

- 1. In the DQ Repository tree view, expand the Metadata and the DB Connections folders in succession.
- 2. Navigate to the column you want to add a task to, account_id in this example.
- 3. Right-click the account_id and select Add task... from the contextual menu.



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

👹 Properti	es 📃 🗖 🔀
Description:	check for null values?
Priority:	Normal Completed
On element:	account_id
In folder:	Metadata/DB Connections
Location:	C:/TOP_builds/TOP-All-r21394-V1.2.0M3/workspace/Metadata/DB Connections/U
0	OK Cancel

- 4. In the **Description** field, enter a short description for the task you want to carry on the selected item.
- 5. In the **Priority** list, select the priority level and then click **OK** to close the dialog box.

The created task is added to the **Tasks** list. For more information on how to access the task list, see Section 11.2.4, "Displaying the task list".

🚈 Tasks 🛛 🏾	3				~ - 6
items					
- 🔺 📔 1	Description	Resource	Path	Locat	Туре
\checkmark	check for null values?	account_id	TOP_DEFAULT_PRJ/TD	D:/TO	Task
\checkmark	pattern task	Address	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	chek this report?	catalog_Analysis	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	test this indicator?	Address	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	test this pattern?	product_id	TOP_DEFAULT_PRJ/TD	D:/TO	Task

From the task list, you can:

- double-click a task to open the editor where this task has been set.
- select the task check box once the task is completed in order to be able to delete it.
- filter the task view according to your needs using the options in a contextual menu accessible through the dropdown arrow on the top-right corner of the **Tasks** view.

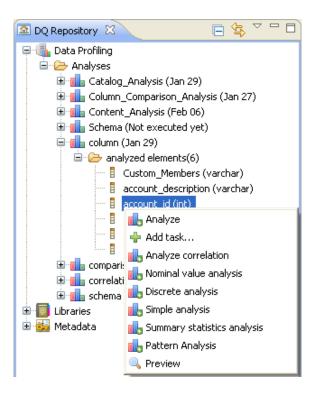
11.2.2. Adding a task to an item in a specific analysis

The below procedure gives an example of adding a task to a column in an analysis. You can follow the same steps to add tasks to other elements in the created analyses.

Prerequisite(s): The analysis has been created in *Talend Open Studio for Data Quality*.

To add a task to an item in an analysis, do the following:

- 1. In the **DQ Repository** tree view, expand the **Analyses** node.
- 2. Expand an analysis and navigate to the item you want to add a task to, the account_id column in this example.
- 3. Right-click account_id and select Add task... from the contextual menu.



4. Follow the steps outlined in Section 11.2.1, "Adding a task to a column in a database connection" to add a task to account_id in the selected analysis.

For more information on how to access the task list, see Section 11.2.4, "Displaying the task list".

11.2.3. Adding a task to an indicator in a column analysis

In the open analysis editor, you can add a task to the indicators set on columns. This task can be used, for example, as a reminder to modify the indicator or to flag a problem that needs to be solved later.

Prerequisite(s):

- A column analysis is open in the analysis editor in Talend Open Studio for Data Quality.
- At least one indicator is set for the columns to be analyzed.

To add a task to an indicator, do the following:

- 1. In the open analysis editor, click Analyzed columns to open the relevant view.
- 2. In the Analyzed Columns list, right-click the indicator name and select Add task... from the contextual menu.

▼ Analyzed Columns			
Connection: MySQL			
Select columns to analyze			
Select indicators for each column			
Analyzed Columns	Datamining Type	Pattern	Operation
First_Name (varchar)		R	
I Last_Name (varchar)	Nominal 💌	Ř	×
🖃 🚦 Address (varchar)	Nominal 💌	<u>I</u>	××××
Row Courts			×
	query		×.
Distinct 🕂 Add Task			*
X Remove eleme	nts		

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

📽 Properti	es 📃 🗖 🔀
Description:	test this indicator?
Priority:	Normal Completed
On element:	Address
In folder:	TOP_DEFAULT_PRJ/TDQ_Data Profiling/Analyses
Location:	D:/TOP_builds/TOP-All-r26328-V3.2.0M1/workspace/TOP_DEFAULT_PRJ/TDQ_Data Profi
0	OK Cancel

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

For more information on how to access the task list, see Section 11.2.4, "Displaying the task list".

11.2.4. Displaying the task list

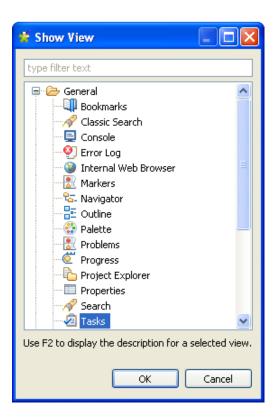
Adding tasks to items will list these tasks in the Tasks list.

Prerequisite(s): At least, one task is added to an item in Talend Open Studio for Data Quality.

To access the **Tasks** list, do the following:

1. On the menu bar of Talend Open Studio for Data Quality, select Window > Show view.....

The [Show View] dialog box is displayed.



- 2. Expand General and then select Tasks.
- 3. Click **OK** to proceed to the next step.

The Tasks view opens in Talend Open Studio for Data Quality listing the added task(s).

⁄ Tasks 🛛	3				~ - 8
5 items					
A 1	Description	Resource	Path	Locat	Туре
	check for null values?	account_id	TOP_DEFAULT_PRJ/TD	D:/TO	Task
\checkmark	pattern task	Address	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	chek this report?	catalog_Analysis	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	test this indicator?	Address	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	test this pattern?	product_id	TOP_DEFAULT_PRJ/TD	D:/TO	Task



You can open the editor corresponding to the item to which a task is attached by double-clicking the appropriate task in the **Tasks** list.

11.2.5. Deleting a completed task

When a task goal is met, you can delete this task from the Tasks list after labeling it as completed.

Prerequisite(s): At least one task is added to an item in Talend Open Studio for Data Quality.

To delete a completed task, do the following:

1. Follow the steps outlined in Section 11.2.4, "Displaying the task list" to access the Tasks list.

🗿 Tasks 🛛	×				~ 0 6
items					
· 🔺 📔	! Description	Resource	Path	Locat	Туре
\checkmark	check for null values?	account_id	TOP_DEFAULT_PRJ/TD	D:/TO	Task
\checkmark	pattern task	Address	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	chek this report?	catalog_Analysis	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	test this indicator?	Address	TOP_DEFAULT_PRJ/TD	D:/TO	Task
	test this pattern?	product_id	TOP_DEFAULT_PRJ/TD	D:/TO	Task

2. Select the check boxes next to each of the tasks and right-click anywhere in the list.

-			
🙆 Tasks	5 23		
4 items			
- A	! Description	Resource	Path
\checkmark	check for null value:	s crm demo	TOP_DEFAULT_P
	🔕 Go to		TOP_DEFAULT_P
			TOP_DEFAULT_P
	E Copy	Ctrl+C	TOP_DEFAULT_P
	🗙 Delete	Delete	
	Select All	Ctrl+A	
	Show In	Alt+Shift+W 🕨	
	Add Task		
	🗙 Delete	Delete	
	Delete Completed T	Tasks 💦	
	Mark Completed	4	
	Properties	Alt+Enter	

- 3. From the contextual menu, select **Delete Completed Tasks**. A confirmation message is displayed to validate the operation.
- 4. Click **OK** to close the confirmation message.

All tasks marked as completed are deleted from the **Tasks** list.



Appendix A. *Talend Open Studio for Data Quality* management GUI

This appendix describes the Graphical User Interfaces (GUI) of Talend Open Studio for Data Quality.

A.1. Main window of *Talend Open Studio for Data Quality*

Talend Open Studio for Data Quality main window is the interface from which you manage data profiling.

The Talend Open Studio for Data Quality main window is divided into:

- the menu bar,
- the toolbar,
- the tree view area,
- a detailed view
- the workspace,
- a tab panel (specific to the Column Analysis editors),
- a cheat sheet view.

The Uniteday

The figure below illustrates Talend Open Studio for Data Quality main window and its possible views.

🔷 DQ Repository 🛿 🦳 🗖	💼 column_analysis 0.1 🕱	🗖 🕮 Cheat Sheets 🕺 📃 🗖
🗞 🖻 🚭 🗸		■ ▽
□ ● ● Data Profiling □ ● ● Analyses(2) ● ● ● Column_analysis 0	Column Analysis	Talend Data Profiler
📃 🗄 📶 mysql_content 0.1	Set the properties of analysis.	🖃 📋 Talend Data Profiler 🛛 🔻
🖨 🐻 Libraries	Name: column analysis	Create a database co
⊞ 📥 Exchange ⊞ 🏱 Indicators	Purpose:	Create a connection
Patterns		Create a catalog ana
Cource Files Cource Files Metadata DB connections	Description:	Create a schema ana Create a column ana Create a columns cor Create a columns cor Create a numerical co
ileDelimited	Author:	Create a nominal cori
MDMconnections	Status: Draft	Create a time correla
🔤 🐻 Recycle Bin	▼ Analyzed Columns	Create an MDM conn
	* Analyzea columns	Create a column set
	Connection: MySQL	Create a functional d
	Select columns to analyze	
	Select indicators for each column	
		Create a column analysis
<	Analyzed Columns Datamining Type Pattern UDI Operation	This tutorial guides you through the creation of a column
Detail View 🛛 🗖 🗖	🖃 🖥 id (int unsigned) 🛛 💽 🔣 🔏 🗙	analysis.
	Row Count 📓 💥	
General	Mull Count	🖲 Start working on this task
Name:	Distinct Count	
Purpose:	Duplicate Count	
Description:	🗄 🖥 First_Name (varchai Nominal 💌 🔏 🔏 🗙	
Number of analyzed element:	🗉 🗄 Last_Name (varchar 🛛 Nominal 💌 🔣 🏑 🗙	
Connection:	Null Count Image: Count Distinct Count Image: Count Duplicate Count Image: Count Image: Count <td></td>	
		<u>×</u>
< >>	Analysis Settings Analysis Results	

The following sections give detailed information about each of the above views.

A.2. Menu bar of *Talend Open Studio for Data Quality*

The menu bar headers and submenus help you perform operations on your enterprise data.

Table 1 describes menus and menu items available to you.

Menu	Menu item	Description	
File	Close	Closes the current open editor in the workspace	
	Close All	Closes all open editors in the workspace	
	Save	Unavailable option.	
	Save All	Unavailable option.	
	Exit	Closes Talend Open Studio for Data Quality main window	
	Open File	Opens a file	
Window	Perspective	Profiling: Opens the data profiler GUI	
	Data Explorer	Data Explorer Opens the data explorer GUI	
	Show View	Opens the [Show View] dialog box which enables you to display different views on <i>Talend Open Studio for Data Quality</i>	
	Preferences	Opens the [Preferences] window which enables you to set your preferences	
	Reset Perspective	Resets the current perspective to its default view after confirmation	
Help	Welcome	Opens a welcoming page which has links to <i>Talen</i> <i>Open Studio for Data Quality</i> documentation and Talen practical sites	
	Help Contents	Opens the Eclipse help system documentation	
	About Talend Open Studio Data Quality	for Displays:	
		-the software version you are using	
		-detailed information on your software configuration that may be useful if there is a problem	
		-detailed information about plug-in(s)	
		-detailed information about <i>Talend Open Studio for Data Quality</i> features	
	Cheat Sheets	Displays a dialog box where you can select a cheat sheet to open	
	Software Updates	Find and Install: Opens the [Install/Update] wizard that helps searching for updates for the currently installed features, or searching for new features to install	
		Manage Configuration: Opens the [Product Configuration] window where you can manage <i>Talend</i> Open Studio for Data Quality configuration	

Table A.1. Table 1—Management menus

Menu	Menu item	Description
	View bookmarks	Opens a bookmarks panel that holds few useful links. These links enable you to easily access specific information related to the usage of <i>Talend Open Studio for</i> <i>Data Quality</i> and/or its database management system
	Key Assist	Opens a list of all short-cut keys

A.3. Toolbar of *Talend Open Studio for Data Quality*

The toolbar contains icons that provide you with quick access to the commonly used operations you can perform from *Talend Open Studio for Data Quality* main window.

Table 2 describes the toolbar icons and their functions.

Table A.2. Table 2—Management toolbar

Icon	Function
	Saves modifications
Q	Import data quality items
Q	Export data quality items
7	Switches to Data Explorer

A.4. Tree view of *Talend Open Studio for Data Quality*

The **DQ Repository** tree view area shows folders for data profiling analyses, patterns and metadata.

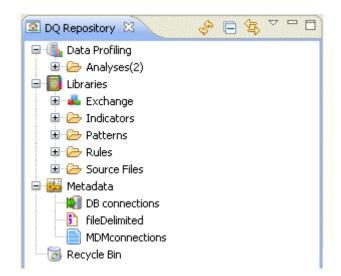
When expanding the **Data profiling** folder in the tree view list, you display the created analyses (either executed or not executed yet).

When expanding the **Libraries** folder in the tree view list, you display the list of the pre-defined patterns and SQL patterns. Imported patterns and patterns created by you will also show under the **Patterns** folder.

Under Libraries as well, you have all created SQL business rules and all imported patterns from Talend Exchange.

When expanding the Metadata folder in the tree view list, you display the list of all created DB connections.

The figure below shows an example of an expanded DQ Repository tree view.

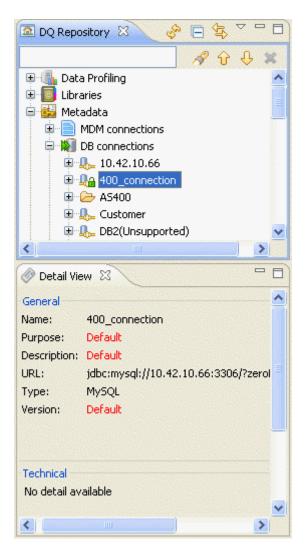


You can use the local toolbar icons to manage the display of the DQ Repository tree view.

A.5. Detailed View of *Talend Open Studio for Data Quality*

This view is located below the **DQ Repository** tree view area. It displays detailed information about the selected element in the tree view area.

The figure below shows an example of the detailed view of the selected DB connection.



You can use the local toolbar icons to manage the display of Detail View.

A.6. The Profiling perspective of *Talend Open Studio for Data Quality*

This perspective contains:

- nothing if no analysis, pattern or DB connection is open,
- the parameter values of the open analysis, pattern or DB connection.

When you open a column analysis, a pattern or a DB connection through the tree view area, the relevant editor opens in *Talend Open Studio for Data Quality* workspace.

You can use the local toolbar icons to manage the display of the workspace.

A.7. Tab panel of the analysis editors

This management tab panel is located at the bottom of the analysis editors. It contains a pair of tabs:

- Analysis Settings,
- Analysis Results.

The Analysis Settings tab lists the settings for the current analysis in the currenty editor.

The figure below is an example of the parameters of a column analysis.

🚹 One_Columr	10.1 🛛 🗖 🗖	
i 🖻 🕀 🔡	* 💿	
Column A	nalysis	
💌 Analysis	Metadata	
Set the prop	erties of analysis.	
Name:	One_Column	
Purpose:		
Description:		
Author:	user@company.com	
Status:	development 💽	
► Analyzed	l Columns	
▶ Data Filter		
Analysis Parameter		
<	>	
Analysis Setting	s Analysis Results	

The Analysis Results tab lists:

- a summary of the executed analysis in the **Analysis Summary** view in which it specifies the connection, the database and the table names for the current analysis,
- the results of the executed analysis, graphics and tables, in the Analysis Results view.

The figure below is an example of a column analysis results.



In the Analysis Results view, you can:

- click the arrow located next to a column name to display the types of analyses done on that column,
- select a type of analysis to display the corresponding generated graphics and tables.

A.8. Selecting a task from *Talend Open* Studio for Data Quality management GUI

You have several ways to select a task from the *Talend Open Studio for Data Quality* main window. You can, for example, use:

- a menu submenu combination, or
- a toolbar icon, or
- a right-click list, or
- shortcut keys.

Example 1: To show a view in the Talend Open Studio for Data Quality main window, either:

- use the Window > Show View... menu submenu combination, or,
- use the Alt+Shift+Q, Q shortcut key.

Example 2: To execute an analysis, do one of the followings:

- use the run icon on the toolbar, or
- right-click the analysis you want to execute and select Run from the contextual menu, or

- click the **Run** button at the bottom of the editor, or
- use the **F6** shortcut key.



Appendix B. Data Explorer management GUI

The Data Explorer embedded in Talend Open Studio for Data Quality allows you to query and browse databases.

This appendix introduces the Graphical User Interfaces (GUI) of the Data Explorer which is based on the SQL Explorer for which you can find documentation at http://www.sqlexplorer.org/.

B.1. Main window of the Data Explorer

The main window of the Data Explorer is the interface from which you manage your database.

The Data Explorer main window is divided into:

- the menu bar,
- the toolbar,
- Connections view,
- SQL History view
- SQL editor view,
- Database Detail view,
- Database Structure view.

The figure below illustrates an example of Data Explorer main window and its components.

File Window Help		
: 💹 🔚		
🗟 Co 🛛 🖸 🗖 🗖	💀 *SQL Editor (0).sql 🛛 🔭 🗖 🗖	🗄 Database 🛛 🗖
MySQL (2 sessions) ······· root ······· Connected sinc	★ ↑ ↓	₩ MySQL/root 🛛
Connected sinc		Crm_demo
	1Select * from examples.insert_id	🖃 🧰 Tables
	×	account ⊕
BQETHISCORY 20	1 [Select * from examples] 🔀 Messages	🗄 🧮 customer
📃 🖬 📾 🗰 🎇 🗈	id firstname lastname title address company	🗈 🧮 departm
type filter text	16 Martin Richards Mr.	🗄 🧮 employe 👘
SQL	17 Janet Brown Mrs.	inventor
Select * from examples.ins	18 Elizabeth Clinton Ms. 19 <null> brown Mr. 4785mlk hall</null>	
	20 <null> <null> <null> 5487mlkkjh mlk</null></null></null>	i product
		i product_
	Query executed in 63 ms. Number of rows returned: 5	⊞ region —
	Database Detail	🗄 🛄 salary
	🕞 Database Detail 🛛 👘 🗖	🕀 🧮 sales_fa
	Columns Info Preview Row Count Primary Keys E	🕀 🤠 sales_re
	Column Name 🛛 Data Type Type Name Column Size 📥	😟 🧮 store
	account_id 4 int 10	🗉 🥅 time_by_
	account_parent 4 int 10	🕀 🎹 warehou
	account_descript 12 varchar 30	🗄 🖽 warehou
		🗄 🗀 Views 🔽
	Columns for `crm_demo`,`account`	

The following sections give detailed information about each of the above components.

B.2. Menu bar of the Data Explorer

The menu bar headers and submenus help you perform operations on your enterprise data.

Table A.1, "Table 1—Management menus" of Appendix A describes menus and menu items available to you.

B.3. Toolbar of the Data Explorer

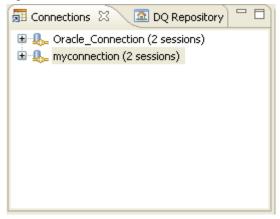
The toolbar contains icons that provide you with quick access to the commonly used operations you can perform from the Data Explorer main window.

Table A.2, "Table 2—Management toolbar" of Appendix A describes the toolbar icons and their functions.

B.4. Connections view

The Connections view shows all the connection profiles that you have set up.

The figure below shows an example of the Connections view.



You can use the local toolbar icons to manage the display of the Connections view.

B.5. SQL History view

This view shows below the Connections view area. Every statement that was successfully executed is logged in the SQL History view.

The view shows the statement, the date and time when the statement was last executed, which connection was used and how many times the statement has been executed. The SQL statements can be filtered, sorted, removed and opened in or appended to the [SQL Editor].

The figure below shows an example of the SQL History view.

SQL History 🛛		式 📖 🕽	s 🖗 🗈 🗖
type filter text			
SQL	Time	Connection	Executions
SELECT * FROM test_top.top_custom WH	2008-09-22 15:27:57	myconnect	2
select * from examples.insert_id	2008-09-22 15:21:52	myconnect	1
select AN_PK, AN_UUID, AN_LABEL, AN_C		Copy To Clipt	ditor h History rom History

You can use the local toolbar icons to manage the display of SQL History View.

B.6. The SQL editor view

This area contains nothing if no [SQL Editor] is open. The [SQL Editor] provides the following features:

- Executing queries using the CTRL-ENTER combination,
- · Basic syntax coloring
- Basic Content Assist
- Overriding result limit
- Word wrapping (if enabled in preferences)
- Session/Catalog/Schema switching
- Loading/Saving SQL scripts
- Commit/Rollback buttons (if session is not in auto-commit mode)
- Display of query execution time of last run query

The figure below shows an example of the [SQL Editor] view.

ன *SQL Editor (16).sql 🛛	
🗄 🛣 🕼 🗁 🖳 🖉 🔻 🕴 💽 🔽 🔽 🔽 🚽	
1select AN_PK, AN_UUID, AN_LABEL, AN_CREATION_DATE, AN_AUTHOR,	an_da 🗠
	<u> </u>
Messages 🔀	
Status Location SQL Text	
Status line 1 0 records updated in 62 ms	

The lower part of the **[SQL Editor]** view, the Messages area, detailed information about your data exploring actions. When you execute a query in the SQL query editor, the **Messages** area displays the query results.



You can save all the queries you execute in the Data Explorer under **Libraries > Source Files** in the DQ Repository tree view of *Talend Open Studio for Data Quality*.

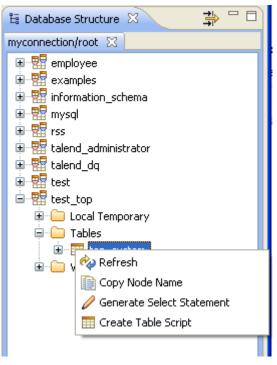
The figure below shows an example of the Messages area.

1 [se	lect * from e	xamples]	🔀 Messages	
id	firstname	lastname	title	
16	action	date	result	
I				
Quer	Query executed in 375 ms. Number of rows returned: 1			

B.7. Database Structure view

Using the **Database Structure** view, you can explore multiple databases simultaneously.

When you select a node in the Database Structure view, the corresponding detail is shown in the Database Detail view. For more information, see Section B.8, "Database Detail view". If the detailed view is not active, double-clicking the node will bring the detail view to the front.



B.8. Database Detail view

Database Detail view shows detailed information for whatever node you select in the database structure view. What is displayed will depend on the type of database that you are using.

The figures below show two examples of a database detailed view.

Connection Info	
Property	Value
Database Product Name	MySQL
Driver Major Version	5
Driver Minor Version	1
Driver Name	MySQL-AB JDBC Driver
Driver Version	mysql-connector-java-5.1.0 (\$Date: 2007-04-05 23:56:0
User Name	root@localhost
URL	jdbc:mysql://localhost:3306/?noDatetimeStringSync=true
Autocommit Mode	true
Commit On Close	true
All Procedures Are Callable	false
All Tables Are Selectable	false
Nulls Are Sorted High	false

Connection info for myconnection/root

💩 Database Detail 🛛					~	
Columns Info Pre	view Row C	ount Primary Keys	Exported Keys	Imported Keys	; Index	es Privileges 🚺
Column Name	Data Type	Type Name	Column Size	Decimal Digits	Radix	Is Null Allowed
id	4	mediumint unsigned	7	0	10	0
FIRST_NAME	12	varchar	255	0	10	1
INITIAL	12	varchar	255	0	10	1
EMAIL	12	varchar	255	0	10	1
ADDRESS_STREET	12	varchar	255	0	10	1
CITY	12	varchar	50	0	10	1
ZIP_CODE	12	varchar	10	0	10	1
STATE	12	varchar	50	0	10	1
STATE_SHORT	12	varchar	50	0	10	1
SUBSCRIPTION_D	93	timestamp	0	0	10	0
CATEGORY	-1	text	65535	0	10	1
CHAR_ID	-1	text	65535	0	10	1
<						>



Appendix C. Regular expressions on SQL Server

This appendix describes in detail how to create a regular expression function on SQL Server databases.

C.1. Main concept

The regular expression function is not built into all different databases environments. This is why you need, when using some databases, to create a User-Defined Function (UDF) to extend the functionality of the database server.

For example, the following databases natively support regular expressions: MySQL, PostgreSQL, Oracle 10g, Ingres, etc., while Microsoft SQL server does not.

After you create the regular expression function, you should use *Talend Open Studio for Data Quality* to declare that function in a specific database before being able to use regular expressions on analyzed columns.

For more information on how to declare a regular expression function in *Talend Open Studio for Data Quality*, see Section 9.1.2.2, "How to define a query template for a specific database from *Talend Open Studio for Data Quality*" and Section 9.1.2.1, "How to declare a User-Defined Function in a specific database".

C.2. How to create a regular expression function on SQL Server

Prerequisite(s): You should have Visual Studio 2005 or 2008. The Visual Studio main window is open.

To create a regular expression function in SQL Server, follow the steps outlined in the sections below.

C.2.1. How to create a project in Visual Studio

You must start by creating an SQL server database project. To do that:

1. On the menu bar, select **File > New > Project** to open the **[New Project]** window.

New Project			? 🔀
Project types:		Templates:	
 Visual Basic Windows Smart Devi Database Starter Kits Visual C# Windows Smart Devi Smart Devi Smart Devi Smart Devi Starter Kits Visual J# Visual C++ Other Project 1 	5 5	Visual Studio installed templates Image: SQL Server Project My Templates Image: Search Online Templates	
A project for creati	ing classes to use in S	QL Server	
Name:	UDF function		
Location:	C:\Documents and S	Settings\mhallam\Mes documents\Visual Studio 2005\Projects	Browse
Solution Name:	UDF function	Create directory for solution	
		ОК	Cancel

- 2. In the **Project types** tree view, expand **Visual C#** and select **Database**.
- 3. In the **Templates** area to the right, select **SQL Server Project** and then enter a name in the **Name** field for the project you want to create, *UDF function* in this example.
- 4. Click **OK** to validate your changes and close the window.

The [Add Database Reference] dialog box is displayed.

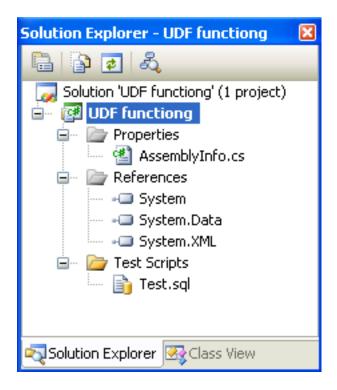
Add Database Reference 🛛 🛛 🔀
Choose a database connection to be added as a database reference.
Available References:
🎯 talend-sql2005\talend.master.dbo
🎯 talend-sql2005\talend.Talend.dbo
Add New Reference
OK Cancel

5. From the **Available References** list, select the database in which you want to create the project and then click **OK** to close the dialog box.



If the database you want to create the project in is not listed, you can add it to the **Available Reference** list through the **Add New Reference** tab.

The project is created and listed in the Solution Explorer panel to the right of the Visual Studio main window.



C.2.2. How to deploy the regular expression function to the SQL server

You need now to add the new regular expression function to the created project, and then deploy the function to the SQl server. To do that:

- 1. In the project list in the **Solution Explorer** panel, expand the node of the project you created and right-click the **Test Scripts** node.
- 2. From the contextual menu, select Add > New Item....

	👻 🗙 Solution	n Explo	orer - Solution ' 👻 📮 🗙	
		🖶 🔁 🕢		
-		Solution 'UDF function' (1 project)		
		·	F function	
			Properties 🕙 AssemblyInfo.cs	
			References	
			- System	
	. 🔗 📗		• System.Data	
to review an f ATL, <u>and th</u>		System.XML		
-	New Item		Add 🔸	
f code i 😐	Existing Item		Add Test Script	
1	New Folder		Exclude From Project	
f code i 🗎	User-Defined Function	Ж	Cut	
· 🗎	Stored Procedure		Сору	
B	Aggregate		Paste	
:o modi 🎦	Trigger	×	Delete	
311G IIIII	User-Defined Type		Rename	
rstem. 🛰	Class		Properties	

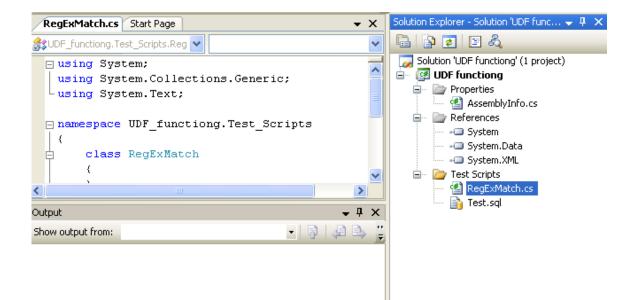
The [Add New Item] dialog box is displayed.

emplates: Visual Studio ir	nstalled template	·s		
📄 Stored Procec 📝 Trigger My Templates		्रि User-Defined Function स्रि ⁹⁹² Aggregate	■ User-Defined Type ▲ Class	
Search Online	Templates			
lame:	RegExMatch.cs			_

3. From the **Templates** list, select **Class** and then in the **Name** field, enter a name to the user-defined function you want to add to the project, *RegExMatch* in this example.

The added function is listed under the created project node in the Solution Explorer panel to the right.

4. Click **Add** to validate your changes and close the dialog box.



5. In the code space to the left, enter the instructions corresponding to the regular expression function you already added to the created project.

Below is the code for the regular expression function we use in this example.

```
Using System;
Using Microsoft.SqlServer.Server;
Using System.Text.RegularExpressions;
Public partial class RegExBase
ł
     [SqlFunction(IsDeterministic = true, IsPrecise = true)]
     Public static int RegExMatch( string matchString , string pattern)
     {
            Regex r1 = new Regex(pattern.TrimEnd(null));
            if (r1.Match(matchString.TrimEnd(null)).Success == true)
            {
                 return 1 ;
            }
            else
            {
                 return 0 ;
Using
     ł
};
```

6. Press **Ctrl+S** to save your changes and then on the menu bar, click **Build** and in the contextual menu select the corresponding item to build the project you created, **Build** *UDF function* in this example.

Buile	ld Debug Tools Wi	indow (
₩	Build Solution Ctrl+Sh	ift+B			
	Rebuild Solution				
	Deploy Solution				
	Clean Solution				
	Build UDF functiong				
	Rebuild UDF functiong				
	Deploy UDF functiong				
	Clean UDF functiong				
	Batch Build				
	Configuration Manager.				

The lower pane of the window displays a message to confirm that the "build" operation was successful or not.

7. On the menu bar, click **Build** and in the contextual menu select the corresponding item to deploy the project you created, **Deploy** *UDF function* in this example.

Buil	d Debug T	ools	Window	(
₩	Build Solution	Ctrl	+Shift+B			
	Rebuild Soluti	Rebuild Solution				
	Deploy Solutio	Deploy Solution				
	Clean Solution	Clean Solution				
	Build UDF fun	tiong				
	Rebuild UDF f	unctio	ing			
	Deploy UDF fi	unctio	ng			
	Clean UDF fur	nction	g			
	Batch Build					
	Configuration	Mana	iger			

The lower pane of the window displays a message to confirm that the "deploy" operation was successful, or not.

Output		
Show output from:	Build	- 🗟 🚑 🚉 🛒 🖸
	: UDF functiong.dll . started: Project: UD	 DF functiong, Configuration: Debug Any CPU
		, Path: C:\Documents and Settings\mhallam\Mes documents\Visu , Path: C:\Documents and Settings\mhallam\Mes documents\Visu
	-	olyInfo.cs, Path: C:\Documents and Settings\mhallam\Mes docu NxMatch.cs, Path: C:\Documents and Settings\mhallam\Mes docu
		up-to-date, O failed, O skipped ===================================

If required:

- 1. launch SQL Server and check if the created function exists in the function list,
- 2. check if the function works well, for more information, see Section C.3, "How to test the created function via the SQL Server editor".

C.2.3. How to set up *Talend Open Studio for Data Quality*

Before being able to use regular expressions on analyzed columns in a database, you must first declare the created regular expression function, *RegExMatch* in this example, in *Talend Open Studio for Data Quality* in the specified database. To do that:

- 1. In the **DQ Repository** tree view, expand **Libraries** and **Indicators** in succession.
- 2. Expand the System folder and then the Pattern Matching indicator.
- 3. Double-click **Regular Expression Matching**, or right-click it and select **Open** from the contextual menu.

The corresponding view displays the indicator metadata and its definition.

💉 *System Indi	💉 *System Indicators/Pattern Matching/Regular Expression Matching 🛛 🗧						
i 🖻 🕀 🔛							
Indicator Settings							
▼ Indicator							
	erties of User Defined Indicator.						
Name:	Regular_Expression_Matching						
Purpose:	pose: evaluate the number of records that match a regular pattern						
	counts the number of records ma	tching the given pattern against the number of records that do not match the given pattern	~				
Description:							
			V				
Author:	xqliu@talend.com						
Status: development							
▼ Indicator							
Add here the	definition of your indicator specific	to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerate.					
Database	Version	SQL Template					
MySQL		SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%> REGEXP BINARY <%=_PATTERN_EXPR_%> THEN 1 END), C	×				
Oracle		SELECT COUNT(CASE WHEN REGEXP_LIKE(<%=_COLUMN_NAMES_%>,<%=_PATTERN_EXPR_%>) THEN 1 END), CC	×				
Postgree	iqL	SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%> ~ <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*) FROP	×				
+							
<			>				
Indicator Definiti	on						

You need now to add to the list of databases the database for which you want to define a query template. This query template will compute the regular expression matching.

4. Click the [+] button at the bottom of the **Indicator Definition** view to add a field for the new template.

	Indicator Definition Add here the definition of your indicator specific to a database. If the expression is simple enough to be used in "ALL_DATABASE_TYPE" type enumerate.							
Da	tabase	Version	SQL Template					
Μ	ysql 💌		SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%> REGEXP BINARY <%=_PATTERN_EXPR_%> THEN 1 END), C	Edit 🗙				
0	racle 💌		SELECT COUNT(CASE WHEN REGEXP_LIKE(<%=COLUMN_NAMES%>, <%=PATTERN_EXPR%>) THEN 1 END), CC	Edit 🗙				
P	ostgre5QL 🗸		SELECT COUNT(CASE WHEN <%=_COLUMN_NAMES_%>~ <%=_PATTERN_EXPR_%> THEN 1 END), COUNT(*) FROP	Edit 🔀				
Μ	icrosoft SQL Server 🛛 😽		SELECT COUNT(CASE WHEN dbo.RegExMatch(<%=_COLUMN_NAMES_%>,<%=_PATTERN_EXPR_%>)=1 THEN 1 EN	Edit 🔀				

- 5. In the new field, click the arrow and select the database for which you want to define the template, **Microsoft SQL Server**.
- 6. Copy the indicator definition of any of the other databases.
- 7. Click the **Edit...** button next to the new field.

The [Edit expression] dialog box is displayed.

-	🖌 Edit expression	_ 🗆 🗙
	Expression	
	SELECT COUNT(CASE WHEN dbo.RegExMatch(<%=COLUMN_NAMES%>,<%=PATTERN_EXPR%>)=1 THEN 1 END), COUNT(*) FROM <%=TABLE_NAME%> <%=WHERE_CLAUSE%>	~
	templates	~
	<%=TABLE_NAME%> <%=COLUMN_NAMES%>	~
	ОК	Cancel

- 8. Paste the indicator definition (template) in the **Expression** box and then modify the text after WHEN in order to adapt the template to the selected database.
- 9. Click **OK** to proceed to the next step. The new template is displayed in the field.
- 10. Click the save icon on top of the editor to save your changes.

For more detailed information on how to declare a regular expression function in *Talend Open Studio for Data Quality*, see Section 9.1.2.2, "How to define a query template for a specific database from *Talend Open Studio for Data Quality*" and Section 9.1.2.1, "How to declare a User-Defined Function in a specific database".

C.3. How to test the created function via the SQL Server editor

• To test the created function via the SQL server editor, copy the below code and execute it:

```
create table Contacts (
FirstName nvarchar(30),
LastName nvarchar(30),
EmailAddress nvarchar(30) CHECK
 (dbo.RegExMatch('[a-zA-Z0-9])-]+@([a-zA-Z0-9])-]+).)
 +(com|org|edu|nz)',
EmailAddress)=1),
USPhoneNo nvarchar(30) CHECK
 (dbo.RegExMatch('\([1-9][0-9][0-9]\) [0-9][0-9][0-9]
\ [0-9][0-9][0-9][0-9]',
UsPhoneNo)=1))
INSERT INTO [talend].[dbo].[Contacts]
                         ([FirstName]
                        , [LastName]
                        , [EmailAddress]
                        , [USPhoneNo])
       VALUES
                      ('Hallam'
                     , 'Amine'
                     , 'mhallam@talend.com'
                     , '0129-2090-1092')
                     , ( 'encoremoi'
                      , 'nimportequoi'
                       'amine@zichji.org'
                      , '(122) 190-9090')
GO
```

• To search for the expression that match, use the following code:

```
SELECT [FirstName]
   , [LastName]
   , [EmailAddress]
   , [USPhoneNo]
   FROM [talend].[dbo].[Contacts]
   where [talend].[dbo].RegExMatch([EmailAddress],
   '[a-zA-Z0-9_\-]+@([a-zA-Z0-9_\-]+\.)+(com|org|edu|nz|au)')
= 1
```

• To search for the expression that do not match, use the following code:

```
SELECT [FirstName]
, [LastName]
, [EmailAddress]
, [USPhoneNo]
FROM [talend].[dbo].[Contacts]
where [talend].[dbo].RegExMatch([EmailAddress],
'[a-zA-Z0-9_\-]+@([a-zA-Z0-9_\-]+\.)+(com|org|edu|nz|au)')
= 0
```