

# Data Explorer User Guide 3.1

---

Introduction .....	8
Data Explorer Editions.....	8
Free .....	8
Professional.....	8
ETL.....	8
Data Migration Suite .....	8
Features .....	9
ETL Engine features.....	12
Running Modes .....	15
Client mode.....	15
Web mode.....	15
Client-server mode.....	15
Compatibility.....	15
Installation .....	15
Client .....	15
Server.....	16
Client-server.....	16
Redistribution of the ETL framework.....	16
Installing License .....	16
Configuration .....	17
Client .....	17
Server .....	17
Logging.....	17
Data Explorer User Interface .....	18
Commands .....	19
Node Search .....	20
User Interface How To .....	21
Add a new Node.....	21
Delete a Node .....	21
Create a new node from the existing one .....	21

Refresh user interface.....	21
Refresh a Node.....	21
Close a Node .....	22
Navigate in the nodes tree.....	22
Search the Node.....	22
Settings.....	23
Settings How To .....	24
Configure Automatic Updates.....	24
Change Data Explorer look & feel.....	24
Hide icons in the tabs.....	24
Display progress indicator in the Web mode.....	25
Change location of the database client for the particular database .....	25
About.....	26
Driver Manager .....	28
Commands .....	28
Fields .....	29
Driver Manager How To.....	30
Use variables in the Url Format .....	30
Use Alias Readers.....	30
How to Install JDBC driver.....	31
Connections .....	32
Commands .....	33
Fields .....	33
Database Connection.....	35
Excel Connection using ODBC .....	36
Excel (*.xls) connection.....	37
Excel (*.xlsx) connection.....	39
XML Connection.....	41
XML Connection with Transformation.....	43
Delimited Text File Connection .....	45
Fixed Length Text File Connection .....	47
Using variables in the URLs .....	48
Browse Database Objects .....	49

View Database Metadata Commands .....	50
View Table Data (table/view/synonym/etc).....	51
Data Viewer.....	53
Features .....	53
Commands .....	54
Data Viewer How To .....	55
View Table Data (table/view/synonym/etc).....	55
Add new data set to view .....	55
View and refresh data.....	56
Change grid parameters.....	56
Apply where clause conditions .....	56
Apply sort criteria.....	58
Interrupt populating data set.....	58
View records of the data set in the Form. View CLOB and BLOB fields.....	58
Describe Data Set.....	59
Search in the Data Set.....	59
Copy data in the grid to the clipboard .....	59
Calculate Function.....	59
Display Data as a Chart .....	59
Export Data Set to the different file formats .....	59
Transform Data Set .....	60
SQL Developer (SQL Editor) .....	61
Features .....	62
Commands .....	62
SQL Developer How To .....	64
Execute SQL statement(s). ....	64
Execute SQL script.....	65
Execute SQL in the External Tool .....	65
Show execution (explain) plan for the query.....	66
Run SQL query against Excel worksheet when Excel ODBC connection is used.....	67
Interrupt SQL execution.....	68
Open new SQL editor tab for the connection in the SQL Developer .....	68
View records of the data set in the Form. View CLOB and BLOB fields.....	69

Use Bind parameters (variables).....	69
Use Typed Bind parameters (variables) .....	69
Use Output parameters .....	71
Use SQL History.....	73
Describe database object at the cursor position .....	74
Describe Data Set.....	76
Change grid parameters.....	76
Manage transactions.....	76
Search in the Data Set.....	76
Copy data in the grid to the clipboard .....	76
Calculate Function.....	76
Display Data as a Chart .....	76
Export Data Set to the different file formats .....	76
Transform Data Set .....	77
Format SQL code.....	77
Search and Replace text in the SQL editor.....	77
Go to the Line in the SQL editor.....	77
ETL Integrated Development Environment .....	78
Glossary.....	78
How ETL Engine Works .....	78
Features .....	79
Examples of the ETL Scenario .....	80
ETL Scenario Editor .....	80
Commands .....	81
ETL Runner .....	82
Commands .....	83
Fields .....	83
ETL Integrated Development Environment How To .....	84
Create ETL scenario.....	84
Add New ETL Scenario .....	84
Save ETL Scenario.....	84
Delete ETL Scenario node .....	85
Create New ETL scenario from the existing one .....	85

Link existing ETL Scenario to the current node.....	86
Expand ETL scenario.....	87
Execute ETL scenario from the editor .....	87
Format ETL code .....	87
Go to the Line in the ETL scenario editor.....	87
Search and Replace text in the ETL scenario editor.....	87
View and Manage ETL Code Snippets .....	88
Open ETL Runner application.....	88
Add scenario to run in the ETL Runner .....	88
Refresh ETL scenario .....	88
Execute ETL Scenario .....	89
Interrupt ETL scenario execution.....	89
One-click Data Migration .....	91
Features .....	92
Commands .....	92
Fields .....	93
Objects List Commands.....	94
Data Migration How To.....	94
Open Data Migration application .....	94
Create new Data Migration Scenario.....	94
Add object to Export List.....	94
Remove object from Export List.....	94
Change a name of the Destination Table.....	94
Change order of objects in the Export List.....	94
Export Data .....	95
Export and Import Data .....	95
Save Data Migration Scenario.....	95
Open existing Data Migration Scenario .....	95
Save Data Migration Scenario as ETL Scenario .....	95
Interrupt Data Migration scenario execution .....	95
Run Extract in the parallel threads .....	95
Stream data from the source to destination .....	95
Create Indexes when creating a table .....	96

Delete data from the destination tables.....	96
Content Management.....	97
Features .....	97
Root level nodes.....	98
Commands .....	99
Root level content node (Home, Local, FTP, SFTP) .....	99
Folder node .....	99
File node.....	99
Excel file node .....	100
Image file node .....	101
Fields .....	101
Local .....	101
Home.....	101
FTP.....	101
SFTP.....	101
Content Management How To .....	102
Add new root level content node such as Home, Local, SFTP and FTP .....	102
Add new folder.....	102
Add file .....	102
Delete node, folder, file .....	102
Copy file from one folder to another.....	102
Move file from one folder to another.....	102
Rename file or folder .....	103
Refresh a folder.....	103
See content of the file.....	103
Edit File.....	103
Security Policy Editor (Web and client-server modes only).....	104
Commands .....	105
Pre-defined user.....	105
Pre-defined roles.....	105
User Editor .....	106
Fields .....	106
Role Editor.....	107

Fields .....	107
Login and Logout.....	108
User Profile (Web and client-server modes only).....	109
Fields .....	109
Common Components .....	110
Data Set Record Viewer .....	110
Describe Data Set.....	112
Calculate Function.....	113
Display Data as a Chart .....	115
Transform Data Set .....	117
Filter Transformation .....	118
Join Transformation .....	119
Minus Transformation .....	120
Order By Transformation .....	121
Pivot Transformation .....	122
De-normalize Transformation.....	124
Remove Duplicates Transformation .....	126
Transpose Transformation.....	127
Union Transformation.....	128
Set Grid Defaults .....	129
Search in the Data Set.....	129
View and Manage Code Snippets .....	130
Grid.....	132
Data Set Export .....	133
Code Formatter.....	136
Search and Replace .....	136
Go to the Line.....	136
Appendix 1. All Keyboard shortcuts.....	138
Appendix 2. System Variables and Folders .....	142

## Introduction

Data Explorer is a complete end-to-end solution for database development, data discovery, data migration, data integration and extract-transform-load (ETL). It runs on all major platforms, including Web browsers, and supports a wide range of databases and other data sources.

This document gives an overview, installation tips and general information about the product. It is mainly concentrated on user interface.

The screenshots throughout the User Guide are produced on Windows XP, Windows 7 and Windows 8 using the default Windows Look and Feel, but Data Explorer lets you choose among other Look and Feels as well.

## Data Explorer Editions

There are four editions of the Data Explorer:

### Free

The free edition is best suited for basic database development tasks, including running queries and scripts, browsing database metadata and viewing data, and exporting data to various file formats. It does not include extended support for databases (only generic JDBC). The ETL, data discovery, data migration, data integration and content management functionality are not included.

### Professional

The professional edition is a feature-rich tool for database development and data discovery. It includes all the functionality of the free edition plus extended support for databases and full data discovery suite including graphical visualizations and interactive transformations. The ETL, data migration, data integration and content management functionality are not included.

### ETL

The ETL edition includes all functionality of the professional edition plus ETL, data migration, data integration and content management functionality. It is best suited for complex database development, ETL, data integration and data migration tasks.

### Data Migration Suite

Data Migration Suite is an edition of the Data Explorer which automates just one task – migration from one or multiple databases (and other data sources) to another. It does not require any programming skills. The functionality of the Data Migration Suite is included in the ETL edition.

## Features

	Free	Professional	ETL	Data Migration Suite
<b>Supported operating systems:</b> Windows, OS X, Linux/Unix	Yes	Yes	Yes	Yes
<b>Supported Browsers:</b> IE 7+, Chrome, Firefox 3.x+, Opera, Safari, any WebKit-based	Yes	Yes	Yes	No
<b>Supported databases:</b> any JDBC, any ODBC	Yes	Yes	Yes	Yes
<b>Extended database support:</b> Oracle,DB2,MS SQL Server,MySQL,ProgreSQL,Informix,Sybase ASE	No	Yes	Yes	Yes
<b>Supported data sources:</b> delimited text, fixed length text, Excel xls, Excel xlsx, XML, XML with transformation, custom using pluggable connectors	Yes	Yes	Yes	Yes
<b>Node Browser:</b> <ul style="list-style-type: none"> <li>global search</li> <li>browser style navigation</li> <li>unified node management (add, modify, delete, copy, paste, etc)</li> </ul>	Yes	Yes	Yes	Yes
<b>Driver manager:</b> manage connection patterns	Yes	Yes	Yes	Yes
<b>Database browser:</b> view database objects, metadata, DDL, search database objects	Yes	Yes	Yes	Yes
<b>Native objects in database browser:</b> packages, triggers, synonyms, sequences, user-defined types,xml schemas, db links, etc	No	Yes	Yes	Yes
<b>Data viewer:</b> <ul style="list-style-type: none"> <li>view data in database objects such as tables, views, synonyms</li> <li>view other data sources such as text, XML and Excel files</li> <li>multiple tabs for multiple data sets</li> <li>filtering and sorting</li> <li>support for CLOB and BLOB fields</li> <li>table and form views</li> <li>limiting number of rows which can be displayed</li> <li>"describe" data set</li> <li>search in the data set</li> </ul>	Yes	Yes	Yes	No
<b>Display data as a Chart:</b> Supported chart types: <ul style="list-style-type: none"> <li>Line</li> <li>Bar</li> <li>Area</li> <li>Pie</li> </ul> Functions: multiple series, 3D charts, save as image, etc. Charts can be displayed for the entire data set or for the selected rows only	No	Yes	Yes	No

		No	Yes	Yes	No
<b>Transform data using graphical UI:</b> Included transformation algorithms: <ul style="list-style-type: none"><li>• Filter</li><li>• Order By</li><li>• Join</li><li>• Union</li><li>• Minus</li><li>• Remove duplicates</li><li>• Transpose</li><li>• Pivot</li><li>• De-normalize</li></ul> Transformation can be performed on the entire data set or on the selected rows only					
<b>Apply functions to the data set:</b> statistical function such as count,sum,min,max,avg,median,etc, date functions such as average age, etc; Functions can be applied to the entire data set or to the selected rows only		No	Yes	Yes	No
<b>Export data set to the following file formats:</b> <ul style="list-style-type: none"><li>• delimited text</li><li>• fixed length text</li><li>• Excel xls</li><li>• Excel xlsx</li><li>• XML</li><li>• XML with transformation</li><li>• SQL</li></ul> Export entire data set or selected rows only; Automatically create multiple output files based on combination of key fields		Yes	Yes	Yes	No
<b>SQL Developer (SQL Editor):</b> <ul style="list-style-type: none"><li>• multiple tabs for multiple connections</li><li>• edit, search and replaces, jump to row number, bookmarks, code highlighting, etc</li><li>• SQL code formatting</li><li>• manual and automatic transaction management</li><li>• execute multiple SQL statements with output to individual tabs</li><li>• execute selected SQL statements</li><li>• execute SQL statements and scripts with input and output parameters including cursors</li><li>• abort currently executed SQL statement</li><li>• persistent history of executed SQL statements</li><li>• code snippets</li><li>• in place “describe” functionality for table/view/synonym</li><li>• all functionality of the Data Viewer</li></ul>		Yes	Yes	Yes	No

<b>Database specific SQL Developer functions:</b> <ul style="list-style-type: none"><li>• explain plan</li><li>• native output cursors</li><li>• execute SQL in the external tool</li><li>• database specific code snippets</li></ul>	No	Yes	Yes	No
<b>ETL Integrated Development Environment:</b> <ul style="list-style-type: none"><li>• edit, search and replace, jump-to-row-number, code highlighting, bookmarks, code folding, etc.</li><li>• ETL code formatting</li><li>• ETL Code snippets</li><li>• <a href="#">embedded ETL engine</a></li><li>• execute ETL scenario from the editor using interactive UI for setting connections and variables</li><li>• manage (add, edit, delete) ETL scenarios as nodes in the Data Explorer node browser</li></ul>	No	No	Yes	No
<b>One-click Data Migration:</b> <ul style="list-style-type: none"><li>• select tables and other data sources from multiple connections to create export list. All connectivity options are supported (jdbc, XML, XML transformation, text, Excel)</li><li>• when editing export list it is possible to change destination table names</li><li>• import (load) all selected data sources into destination. All connectivity options are supported (jdbc, XML, XML transformation, text, Excel)</li><li>• if table doesn't exist in the destination it is automatically created using source data set as a template</li><li>• when creating a table from the source system can also create indexes</li><li>• there are options to extract sources in parallel, stream data from the source to destination, ignore errors on insert (for example duplicated key exception) and delete records from the destination tables before loading data</li></ul>	No	No	Yes	Yes
<b>Content management:</b> <ul style="list-style-type: none"><li>• integrated directly into nodes browser</li><li>• browse folders and files in the local and remote locations</li><li>• local file system, FTP, SFTP and Web are supported</li><li>• Copy/Move/Add/Delete files and folders</li><li>• View and Edit files in multiple formats (text, Excel, images, XML, etc.)</li><li>• For Excel files apply filters, sort, perform transformations, apply functions, export data and display data as a chart</li></ul>	No	No	Yes	No

## ETL Engine features

ETL engine is integrated into Data Explorer but majority of its functions can be used only in the Data Explorer ETL edition.

	<b>Details</b>
<b>Supported databases</b>	<ul style="list-style-type: none"> <li>• Any JDBC and ODBC</li> </ul>
<b>Extended database support</b>	<ul style="list-style-type: none"> <li>• Oracle,</li> <li>• DB2,</li> <li>• MS SQL Server,</li> <li>• MySQL,</li> <li>• ProgreSQL,</li> <li>• Informix,</li> <li>• Sybase ASE</li> </ul>
<b>Supported data sources</b>	<ul style="list-style-type: none"> <li>• delimited text,</li> <li>• fixed length text,</li> <li>• Excel xls,</li> <li>• Excel xlsx,</li> <li>• XML,</li> <li>• XML with transformation</li> <li>• custom using pluggable connectors</li> </ul>
<b>ETL engine</b>	<ul style="list-style-type: none"> <li>• XML-based scenario language</li> <li>• Extract data from multiple sources and load into multiple destinations</li> <li>• All connectivity options are supported (jdbc, XML, XML transformation, text, Excel)</li> <li>• Stream unlimited data sets from the source to destination</li> <li>• All data types supported including CLOBs and BLOBs with automatic or manual conversion between source and destination databases (data sources)</li> <li>• Automatic and manual field's mapping</li> <li>• Extract and Load each data set in parallel with forks and joins</li> <li>• Inner scenarios with conditional and in-loop execution</li> <li>• Automatic table and indexes creation based on the source data set specification</li> <li>• Manual and automatic transactions management (commit intervals)</li> <li>• Per field functions in SQL and JavaScript</li> <li>• Support for automatic primary/foreign key generation with mapping to old primary/foreign key</li> <li>• Validation using JavaScript</li> <li>• Conditional sources and destinations</li> <li>• Conditional (IF-THEN-ELSE) execution</li> <li>• Automatic exception handling</li> <li>• Automatic Insert/Update/Delete/Merge</li> <li>• In-line SQL in scenarios</li> <li>• Pre/post/inline extract and load tasks</li> <li>• OS command execution</li> <li>• File based tasks (file system, ftp and sftp are supported)</li> </ul>

<b>Transformations</b>	<ul style="list-style-type: none"> <li>• Regex transformation</li> <li>• XSL transformation</li> <li>• Transformation using JavaScript</li> <li>• Sorting</li> <li>• Transpose Matrix</li> <li>• Filtering</li> <li>• Remove Duplicates</li> <li>• Union</li> <li>• Join</li> <li>• Minus</li> <li>• Pivot</li> <li>• De-normalize</li> </ul>
<b>Oracle specific Functionality</b>	<ul style="list-style-type: none"> <li>• Using sequences to generate primary keys</li> <li>• Full PL/SQL support including anonymous SQL blocks, inner functions, procedures, named variables, etc.</li> <li>• Cursors as data sources</li> <li>• Extract using SQL*plus and load using SQL*loader (requires Oracle client)</li> <li>• Table copy using SQL*plus COPY command (requires Oracle client)</li> <li>• Support for MERGE, exception handling, date+time conversion, temporary tables</li> </ul>
<b>DB2 specific functionality</b>	<ul style="list-style-type: none"> <li>• Using sequences and auto-increment fields to generate primary keys</li> <li>• Full SQL PL support including functions, procedures, named variables, etc.</li> <li>• Cursors as data sources</li> <li>• Extract and load using SYSPROC.ADMIN_CMD</li> <li>• Support for MERGE, exception handling, date+time conversion, temporary tables</li> </ul>
<b>MS SQL Server specific functionality</b>	<ul style="list-style-type: none"> <li>• Using auto-increment fields to generate primary keys</li> <li>• Full Transact SQL support including functions, procedures, named variables, etc.</li> <li>• Cursors as data sources</li> <li>• Extract and load using BCP (requires MS SQL server client)</li> <li>• Support for exception handling, date+time conversion, temporary tables</li> </ul>
<b>MySQL specific functionality</b>	<ul style="list-style-type: none"> <li>• Using auto-increment fields to generate primary keys</li> <li>• Full MySQL stored procedure language support including functions, procedures, named variables, etc.</li> <li>• Cursors as data sources</li> <li>• Extract using select INTO OUTFILE and Load using LOAD DATA</li> <li>• Support for exception handling, date+time conversion, temporary tables</li> </ul>
<b>PostgreSQL specific functionality</b>	<ul style="list-style-type: none"> <li>• Using sequences and serial fields to generate primary keys</li> <li>• Full PL/pgSQL support including functions, named variables, etc.</li> <li>• Cursors as data sources</li> <li>• Extract and Load using COPY</li> <li>• Support for exception handling, date+time conversion, temporary tables</li> </ul>

<b>Informix specific functionality</b>	<ul style="list-style-type: none"><li>• Using sequences and serial fields to generate primary keys</li><li>• Full SPL support including functions, procedures, named variables, etc</li><li>• Cursors as data sources</li><li>• Extract and load using DBACCESS (requires Informix client)</li><li>• Support for MERGE, exception handling, date+time conversion, temporary tables</li></ul>
<b>Sybase ASE specific functionality</b>	<ul style="list-style-type: none"><li>• Using auto-increment fields to generate primary keys</li><li>• Full T-SQL support including functions, procedures, named variables, etc.</li><li>• Cursors as data sources</li><li>• Extract and load using BCP (requires Sybase Adaptive Server client)</li><li>• Support for exception handling, date+time conversion, temporary tables</li></ul>

## Running Modes

Data Explorer supports three running modes. All modes share the same components, and have a consistent look and functionality.

### Client mode

Desktop application. Everything runs on the local PC. The role-based security is disabled.

### Web mode

Application runs on the web server. Users access application through the Web browser. Role-based security is automatically activated for the Web mode.

### Client-server mode

Desktop application as a client. There is a server component which runs on the web server. All of the heavy lifting is done on the server including connectivity, SQL, ETL, content management, etc. Role-based security is automatically activated for the client-server mode.

## Compatibility

Data Explorer was tested in Windows XP and above (including Windows 8) 32 and 64 bit, OS X Leopard and above (including Mountain Lion). It is expected to work in all major versions of the UNIX and Linux.

In the Web mode Data Explorer works in all modern browsers (Internet Explorer starting from version 7). IE 9 and above must be set to compatibility mode. JavaScript and cookies must be enabled in all browsers. The fast JavaScript engine is required.

Data Explorer requires Java 6 and above. It is tested in Java 7.

## Installation

To run Data Explorer you need a Java runtime. If you are using Windows, you can download a zip archive that includes Java (32 or 64 bit), or let the application automatically check for Java. If Java VM is not found on your computer or you have an older version of Java, application will display a warning message. You can manually install Java for Windows, Linux, Unix and OS X by clicking on this link:  
<http://www.java.com/en/>

If you are planning to use Data Explorer only in the [client mode](#) you don't need to download a server component. Also, there is no need to download a client if you are going to use Data Explorer only in the [web mode](#). [Client-server mode](#) requires downloading the server component and the client.

### Client

1. Download archive file for the particular platform. Use [client](#) downloads.
2. Extract it anywhere in the file system. Example after extracting: c:/dataexplorer. On the OS X it is recommended to extract it to the **applications** folder (or extract anywhere and then copy to **applications** folder).
3. Find executable in the APP\_HOME and create a shortcut/link if needed. The executables are: **dataexplorer.exe** on Windows, **dataexplorer.app** on OS X and **dataexplorer.sh** on Linux/Unix. APP\_HOME is a root folder where application is installed. For example: c:/dataexplorer.
4. Use executable/link to executable to run application.

5. Alternatively you can use dataexplorer.jar.

### **Server**

1. Make sure you have application server which supports Java servlets. We recommend Tomcat but others will work as well.
2. Download archive file. Use **server** downloads.
3. Extract it to the folder on the server, dedicated for web applications. For example for Tomcat it is TOMCAT\_HOME/webapps. Example after extracting: c:/tomcat/webapps/dataexplorer
4. Restart app server
5. Open the following url (example) in your favorite Web browser:  
 http://localhost:8080/dataexplorer/wings. Make sure you are using correct host and port.  
 The best performance will be achieved in the Web browser with the fast JavaScript engine.  
 Internet Explorer 9 and above should be set to compatibility mode.
6. Enter default user name and password to login (admin/admin).

### **Client-server**

1. Install client and server (see above).
2. Uncomment line containing property **app.server.url** in the config.properties file located in the **client's** config folder. Change url and port if required.  
 Example: app.server.url=http://localhost:8080/explorer/ide
3. Use executable/link to executable to run application.
4. Enter default user name and password to login (admin/admin).

### **Redistribution of the ETL framework**

ETL framework is included in Data Explorer and can be embedded and redistributed. Please check out file etl\_framewrok\_redist.txt located in the app\_home/doc folder.

### **Installing License**

Right after installing, application includes a 20 days evaluation license for the Data Explorer ETL edition (or 20 days evaluation license for Data Migration Suite, depending on what you downloaded). If you want to continue using advanced functionality of the professional or ETL editions (or continue using Data Migration Suite, depending on what you downloaded) after evaluation license expire you will need to purchase and install a particular license.

1. Purchase and download license file
2. Put the license file in the APP\_HOME/**config** folder. If you are running Data Explorer in the client-server mode, you need to copy the license file to both - the client and the server.
3. **In the client and client-server modes you need a license for each client's seat**
4. Restart application.

**Client example:** c:/dataexplorer/config/bundle.lic

**Server example:** c:/tomcat/webapps/dataexplorer/config/oracle.lic

**Note.** There can be multiple license files: for the different components, different license terms, etc. Data Explorer always uses the most permissive license for the particular component.

## Configuration

Data Explorer is ready to use right after installation and typically does not require any additional configuration steps. However, if you absolutely have to you can change startup Java system properties on the client (for example minimum and maximum memory limits for jvm). For the Web application you can change servlet configuration and HTTP session properties (for example HTTP session timeout).

**Note:** Please note that it has nothing to do with the Data Explorer [Settings](#).

### Client

To change startup Java system properties:

1. Open APP\_HOME/appstart.properties file in your favorite text editor.
2. Modify the line containing **app.vm.options**. The default min/max memory limits for JVM are app.vm.options=-Xms100m -Xmx1000m
3. Save the file
4. Changes will take effect after application has restarted.

### Server

To change servlet configuration and HTTP session properties:

1. Open APP\_HOME/WEB-INF/web.xml file in your favorite text editor.
2. To modify HTTP session time out change the line containing node **<session-timeout>**. The default value for the session time out is 10 minutes:
 

```
<session-config>
    <session-timeout>10</session-timeout>
  </session-config>
```
3. Change other parameters if needed but make sure you know what you are doing. It is easy to break things here.
4. Save file.
5. Changes will take effect after server has restarted.

### Logging

You can change logging properties such as log level by modifying file log4.properties located under APP\_HOME/config.

## Data Explorer User Interface

Data Explorer user interface includes three resizable panels: Nodes Browser, Applications List and Node Editor/Applications panel (App panel). It is possible to have multiple open applications and in some cases node editors.

When node is selected in the Nodes Browser the appropriate Node Editor is displayed in the first tab of the App panel.

When application is selected in the Applications List the new tab is created in the App panel. If app already opened, the tab with the application is selected. The app can be closed at any time by clicking on the small [x] button on the right corner of the tab.

There is a Data Explorer toolbar on top of the screen with Add/Delete/Navigation commands. Each Node Editor/Application can have its own toolbar which is displayed on top of the App panel.

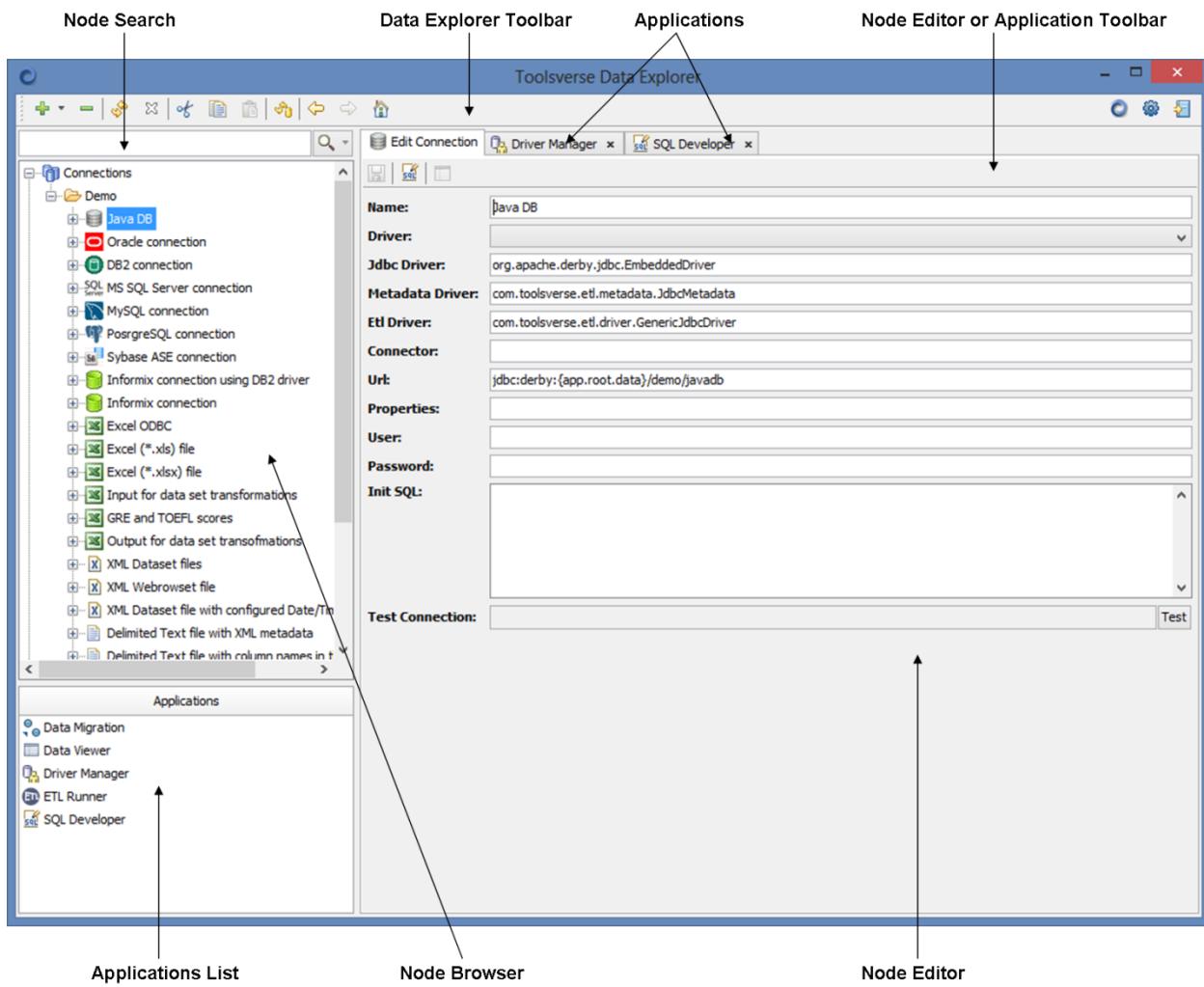


Figure 1: Data Explorer UI

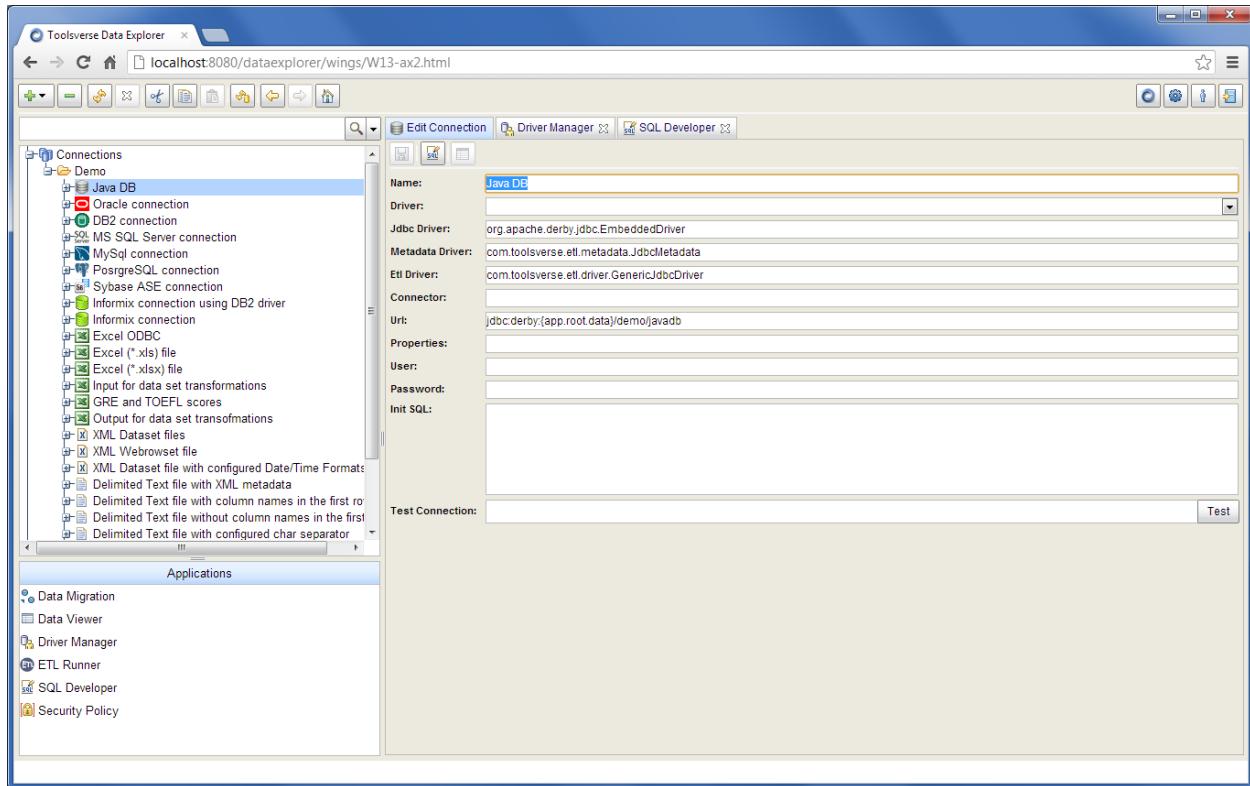


Figure 2: Data Explorer Web UI

## Commands

List of commands in the Data Explorer toolbar, some of them also displayed in the Nodes Browser context menu.

Icon	Command	In context menu
	Add node – opens menu with the list of Nodes which can be added to the currently selected Node	No
	Delete Node – deletes currently selected Node	No
	Refresh Node – refreshes currently selected Node	Yes
	Close Node – closes currently selected Node	Yes
	Cut Node – cuts currently selected Node	Yes
	Copy Node – copies currently selected Node	Yes
	Paste Node – adds previously cut or copied Node to the currently selected Node	Yes

	Refresh All – refreshes Nodes tree. Closes Node Editor and all applications	No
	Go Back one Node – when Nodes are visited (selected) the Data Explorer remembers order in which they were visited. Go Back jumps to the Node located <b>prior</b> currently selected Node in the ordered list. It works similarly to the Web browser Back command.	No
	Go Forward one Node – jumps to the Node located <b>after</b> currently selected Node in the ordered list. It works similarly to the Web browser Forward command.	No
	Go Home – jumps to the first Node in the tree	No
	About – displays About dialog window	No
	Settings – opens Settings app which includes all configuration options. App opens in the separate tab	No
	User Profile – displays User dialog window where password and other attributes for the currently logged user can be changed. Available in the <b>Web</b> and <b>client-server</b> modes only	No
	Logout – in the <b>client</b> mode exits Data Explorer. In the <b>Web</b> and <b>client-server</b> modes logs current user out and displays Login screen.	No

## Node Search

Node Search toolbar is located on top of the Nodes Browser panel. Search works across all nodes, including (if requested) not expanded. Each node has its own “is found” criteria but node name is always included. Search is case insensitive. If not specifically requested (see Extended Search) search starts from the currently selected Node.

When user clicks on the small "down" arrow in the search button the popup menu is displayed. It includes Extended Search menu item and items for quick jump to the one of the top level nodes, such as Connections, ETL Scenarios, Content, etc.

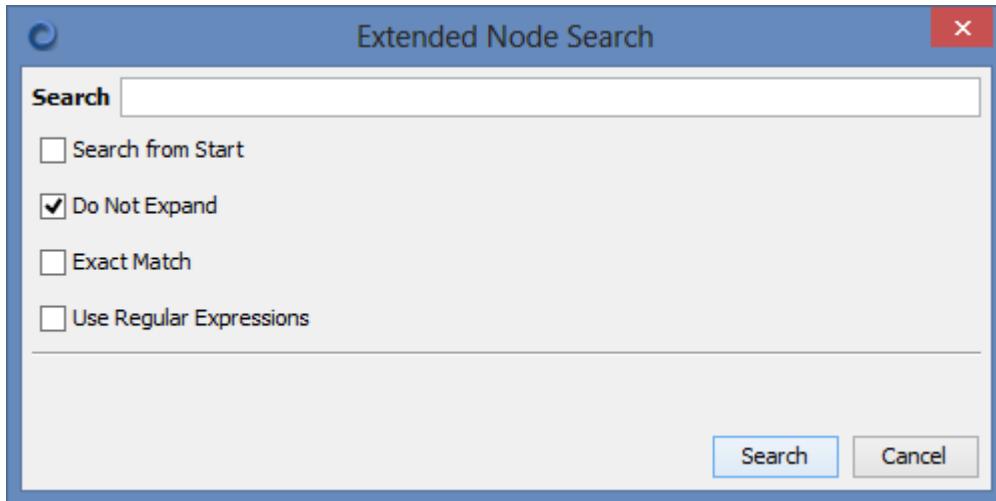


Figure 3: Extended Search Dialog Window

## User Interface How To

### Add a new Node

To add a new node select a node where you want new node to be added to, click on the "add node" button in the Data Explorer toolbar, select a node type in the popup menu. Fill all required fields in the node editor and save your changes by clicking on the "save" button.

In some cases the additional "add node" dialog window is displayed. If you exit this window using "cancel" button the node will not be added.

### Delete a Node

To delete a node select it in the nodes browser and click on the "delete node" button. Confirm that you want to delete it.

### Create a new node from the existing one

Use "copy node" to copy existing node (alternatively you can use a "cut node" ) , select a node in the nodes browser where you want a new node to be added to, click on the "paste node" button. Copy, Cut and Paste are also available in the nodes browser popup menu.

### Refresh user interface

If application is running in one of the multi user modes such as Web or client-server other users might be adding, modifying or deleting nodes. Use "refresh all" button to refresh a nodes tree. It will also close all open editors and applications which can be useful if you want to see changes you made to the user interface in the Settings.

Be aware that Data Explorer does not request confirmation when you click on "refresh all" button.

### Refresh a Node

You might need to refresh a node if you or somebody else made a change to the underlying data. For example if currently selected node is a list of database tables and you just executed a script which added new a table you will need to refresh a node to see new table in the list. Use "refresh node" button to refresh currently selected node.

### ***Close a Node***

In some cases there are additional resources associated with the open node. For example if node is a Connection it can be a physical database connection. When you are done working with a node it might be a good idea to close it. Use "close node"  button when available.

### ***Navigate in the nodes tree***

When you select a node, the nodes browser remembers the order in which nodes where visited. Use "go back"  and "go forward"  buttons to navigate back and forth. It is similar to the Web browser back and forward functions. Click on the "home"  to select a first top level node in the nodes tree.

### ***Search the Node***

Sometime search is a fastest way to find a node you are interested in. Place cursor in the node search toolbar, enter search criteria and click on the "search" button (or hit "Enter"). Use Shift+F7 keyboard shortcut to jump to the search toolbar from anywhere in the Data Explorer. This keyboard shortcut can be changed using the following access path: Settings->General->Go to Node Search.

The extended search is available if you click on the small "down" arrow in the search button. There is also a quick jump list which includes all top level nodes.

## Settings

Data Explorer is highly customizable. You can configure the way it looks, controlled, formats code, etc. Each plug-in has its own set of configurable properties which are automatically added to the configuration widget. There are properties specific for Client and Web modes. For example in the Client mode Data Explorer supports multiple look&feels (skins) but in the Web mode it does not.

Click on the “gears”  button in the Data Explorer toolbar to open “Settings” widget. To save your changes click on the “Save” button. “Save” button is disabled until you changed something.

If you changed something, in some cases you might need to restart Data Explorer or (in the Web and client-server modes) logout/login to see the changes. Data Explorer will let you know if you need to restart but it is generally a good idea to restart/relogin if you are changing keyboard shortcuts.

**Note:** In the Web mode the “local” settings (user personal preferences) are saved in the HTML5 persistant storage which is specific for each browser. For example if you were using Chrome and changed keyboard shortcut for the Search from Shift+F7 to Ctrl+F7 and then switched to the IE 10 you will have to reassign keyboard shortcut again.

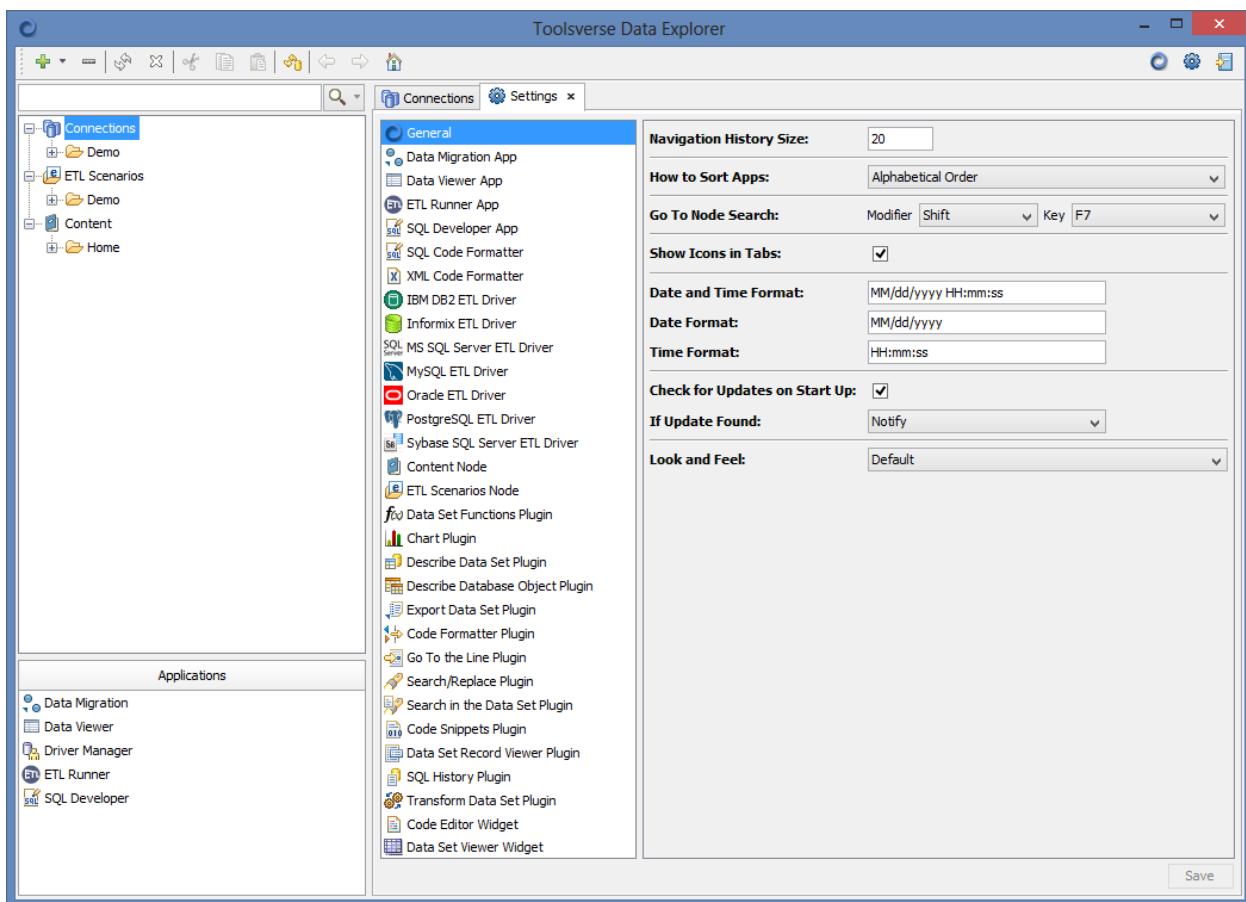


Figure 4: Settings Widget

## Settings How To

### *Configure Automatic Updates*

Data Explorer can automatically check for updates on start up. It is always checking for updates when About dialog window is displayed.

If update is found Data Explorer will either:

- Notify about new version (default)
- Automatically Install and Notify - notify about new version, download and install update
- Automatically Install - silently download and install update (similar to Chrome updates)

Last two options require restarting.

**Note:** Automatic Install (with or without notification) does not work in the Web mode.

To change automatic update parameters use Settings->General->Check for Updates on Start Up and If Update Found.

### *Change Data Explorer look & feel*

Setting->General->Look and Feel. Make a change. Save. You will need to restart Data Explorer to see changes.

The default look & feel for the particular operation system always called "Default". For Windows it depends on version of the Windows, for OS X it is "Quaqua", etc.

The following look & feels are included:

Look & feel	Platforms
Windows	Windows (default in Windows)
<a href="#">Quaqua</a>	OS X (default in OS X)
<a href="#">Nimbus</a>	All
<a href="#">PGS</a>	All
Metal	All
CD/Motif	All

**Note:** Different look &feels are supported only in the client and client-server modes.

### *Hide icons in the tabs*

By default Data Explorer displays icons in the tabs for each application and node editor. If you like more minimalistic look you can hide them. Use Settings->General->Show Icons in Tabs access path. Make a change. Save. You will need to refresh UI or restart Data Explorer to see changes.

Additionally SQL Developer application has its own similar setting. Go to Settings->SQL Developer App->Show Database Icons in Tabs and make a change. Save. Close SQL Developer.

### ***Display progress indicator in the Web mode***

In the client and client-server modes progress indicator is always displayed for the long running tasks, such as search, SQL execution, ETL, etc. You can use Cancel button to interrupt process at any time.

However, it is disabled by default for the Web mode. The reason is an asynchronous nature of the AJAX framework used in the Web mode. You can manually enable progress indicator for the Web mode.

It is controlled for each function individually:

What	Where
Search node	Settings->General->Show search progress
Expand node	Settings->General->Show expand progress
Data viewer->view or refresh data	Settings->Data Viewer App->Show Progress
Data Migration->execute	Settings->Data Migration App->Show Progress
ETL Runner->execute	Settings->ETL Runner App->Show Progress
SQL Developer->execute SQL	Settings->SQL Developer App->Show Progress
Content nodes->file operations	Settings->Content Node->Show Progress
Describe database object	Settings->Describe Database Object Plugin->Show Progress
Export Data Set	Settings->Export Data Set Plugin->Show Progress
Calculate Function	Settings->Data Set Functions Plugin>Show Progress
Display Chart	Settings->Chart Plugin>Show Progress
Transform Data Set	Settings->Transform Data Set Plugin>Show Progress

### ***Change location of the database client for the particular database***

When working with databases in many cases you can get an access to the additional functionality if you have a client for the particular database installed. For example if you have an Oracle client you can execute SQL in the sql\*plus or load data using sql\*loader.

**Note:** The database client must be installed on the same computer which runs SQL.

If you have a license for the ETL driver with support for external tools or evaluation license has not expired yet you can configure a location of the database client tool using the following access path: Settings->Particular ETL Driver ->Location of the command line tools (or Home.)

For example for the Oracle: Settings->Oracle ETL Driver ->Location of the Oracle command line tools.

Make sure location points to the exact folder with the database client binary files. For example c:/oracle/11g/client/bin.

## About

The About dialog window shows product name, version, license type, license expiration date (if it is about to expire) and version of the Java runtime. Click on the “toolseverse”  button in the Data Explorer toolbar to open “About” dialog window.



Figure 5: About

Each time About dialog window is displayed Data Explorer checks for updates. Depending on configuration parameters it can automatically download and install update or just notify user about update status. Possible statuses:

- empty - the update service is off line or computer is not connected to the Internet
- checking for updates - Data Explorer is currently checking for updates
- up to date - the current version of the Data Explorer is up to date
- downloading updates - Data Explorer is currently downloading updates
- update downloaded - the update is downloaded and is ready to be installed

To see detail information about installed add-ons and Data Explorer Credits click on the “Details” button.

**Note:** If add-on is installed but evaluation license has expired or you don't have a license for it - it will not be displayed in the Extensions window.

The screenshot shows a Windows-style dialog box titled "Details". At the top left is a small circular icon with a blue dot. To the right of the title bar are standard window controls: a maximize button, a minimize button, and a close button. Below the title bar is a horizontal tab bar with two tabs: "Extensions" and "Credits". The "Extensions" tab is selected and highlighted with a thin border. The main area of the dialog is a scrollable table with a light gray background. The table has five columns with headers: "Type", "Extension", "Vendor", "Version", and "License Expires". All entries in the table show the same values: Type is "Node" or "Code Formatter", Extension is a specific file type or folder name, Vendor is "Toolsverse", Version is "3.0.1", and License Expires is "Never". There are 20 rows in the table, each representing a different installed extension. At the bottom right of the dialog box is a rectangular button labeled "OK".

Type	Extension	Vendor	Version	License Expires
Code Formatter	SQL Code Formatter	Toolsverse	3.0.1	Never
Code Formatter	XML Code Formatter	Toolsverse	3.0.1	Never
Node	Alias Folder	Toolsverse	3.0.1	Never
Node	Connection	Toolsverse	3.0.1	Never
Node	Connections	Toolsverse	3.0.1	Never
Node	Metadata	Toolsverse	3.0.1	Never
Node	Table Metadata	Toolsverse	3.0.1	Never
Node	ETL Folder	Toolsverse	3.0.1	Never
Node	ETL Scenario	Toolsverse	3.0.1	Never
Node	ETL Scenarios	Toolsverse	3.0.1	Never
Node	Content	Toolsverse	3.0.1	Never
Node	File	Toolsverse	3.0.1	Never
Node	Excel 1997-2003 File	Toolsverse	3.0.1	Never
Node	Excel 2007 File	Toolsverse	3.0.1	Never
Node	Image File	Toolsverse	3.0.1	Never
Node	Text File	Toolsverse	3.0.1	Never
Node	Unknown File Type	Toolsverse	3.0.1	Never
Node	Folder	Toolsverse	3.0.1	Never
Node	Home Folder	Toolsverse	3.0.1	Never

Figure 6: Details

## Driver Manager

Driver Manager is a Data Explorer application for creating and managing templates which can be used to create a database or data source connection. Click on the "driver manager"  icon in the Application list to open Driver Manager.

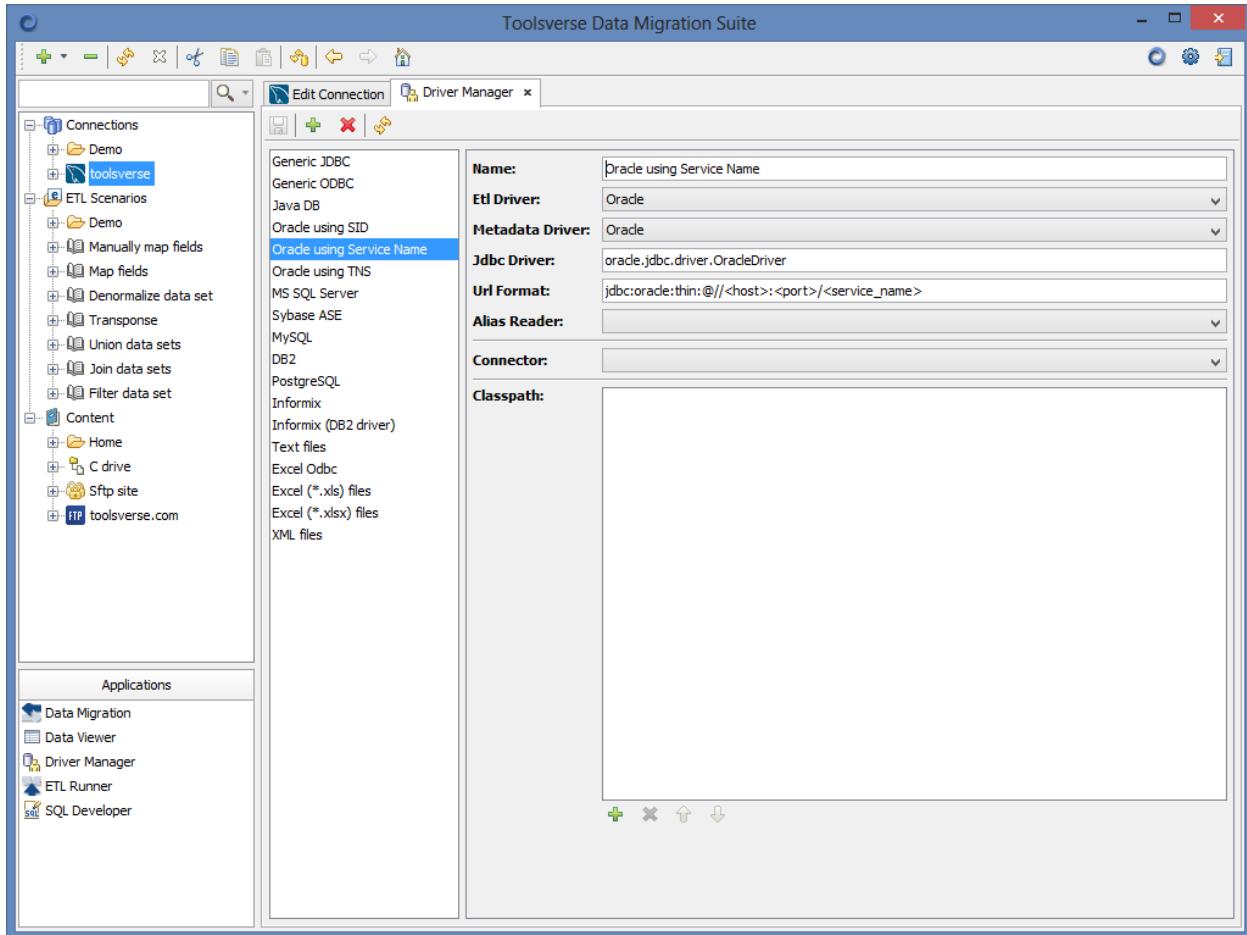


Figure 7: Driver Manager

## Commands

Icon	Command
	Save changes. Disabled until change is made
	Add driver's template
	Delete driver's template
	Refresh driver templates. This command is useful if more than one user can edit templates (for example in the Web or client-server mode).

## Fields

Field	Description
Name	The name of the Driver's pattern. Cannot be empty.
ETL Driver	The ETL Driver. Select ETL Driver from the list of available drivers. The ETL driver is used by various add-ons to perform tasks specific for the particular database.
Metadata Driver	The Metadata Driver. Select Metadata Driver from the list of available drivers. The Metadata Driver is used by Data Explorer to discover metadata for the particular database.
Jdbc Driver	The Jdbc driver class name.
Url Format	The pattern for the url. For example jdbc:oracle:thin:@<host>:<port>:<sid>
Alias Reader	The reader of pre-configured connection aliases. For example tnsnames.ora file reader.
Connector	The Connector. Select connector from the list of available connectors, such as XML connector, Excel connector, etc
Classpath	The classpath to the database specific jar(s). For example c:/oracle/ojdbc6.jar It is not required as long as jdbc jar files are placed under APP_HOME/jdbc folder.

## Driver Manager How To

### *Use variables in the Url Format*

To create a variable use <variable name>. For example you want to create a template for the connection to the Oracle database. The connection url includes a **jdbc driver name**, **host**, **port** and **service name**. Possible Url format can look like this: jdbc:oracle:thin:@//<host>:<port>/<service\_name>.

When creating a connection using this template Data Explorer will add fields **Host**, **Port** and **Service name** to the interface. The Url field will be filled automatically:

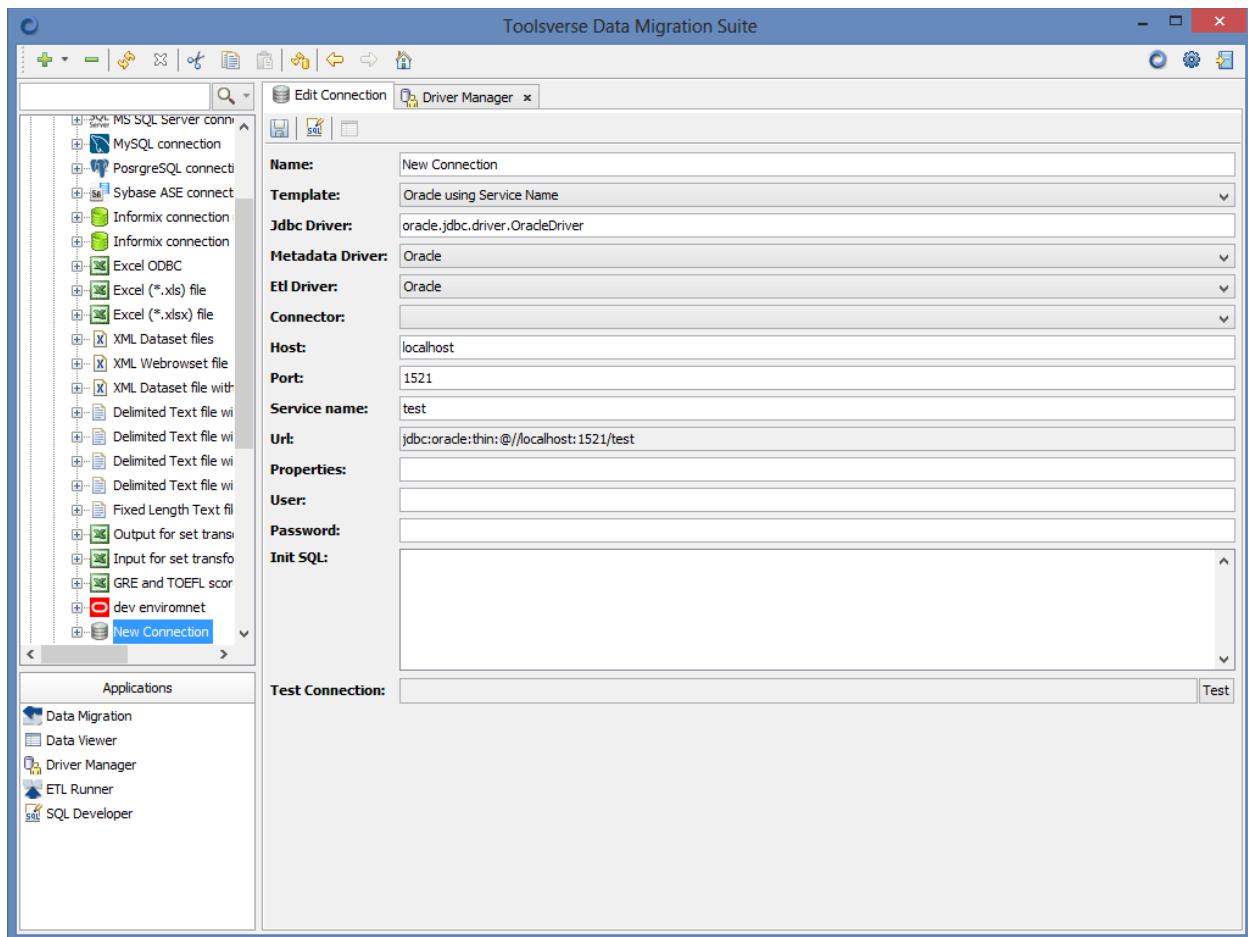


Figure 8: Url Template

### *Use Alias Readers*

Some databases include an external client software which can connect to the database using data from the special configuration files. For example Oracle comes with SQL\*Net client which uses tnsnames.ora file usually located in the ORACLE\_HOME/network/admin directory.

Data Explorer includes pluggable readers for some configuration files. Currently, only reader for the tnsnames.ora is included.

When particular reader is used Data Explorer will provide a drop down to select a pre-configured connection alias from the list.

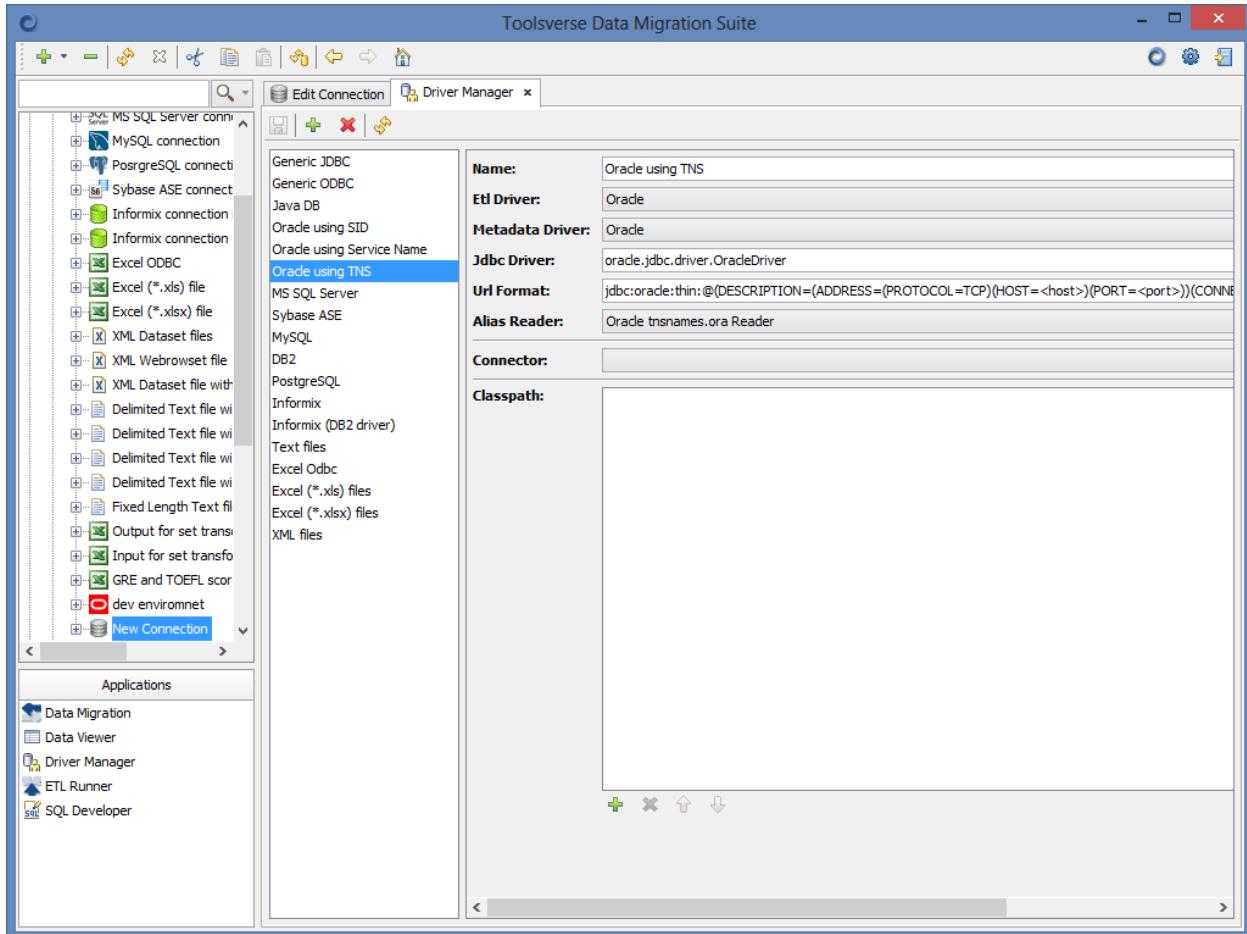


Figure 9: Alias Reader

### ***How to Install JDBC driver***

1. Download JDBC driver from the vendor's website
2. Create a folder under APP\_HOME/jdbc. For example APP\_HOME/jdbc/sqlanywhere
3. Copy all downloaded files into this folder
4. Restart application

The following JDBC drivers are included in Data Explorer:

JDBC Driver	Location
IBM DB2	APP_HOME/jdbc/db2
Informix	APP_HOME/jdbc/informix
MS SQL Server	APP_HOME/jdbc/mssql
MySQL	APP_HOME/jdbc/mysql
Oracle	APP_HOME/jdbc/oracle
PostgreSQL	APP_HOME/jdbc/postgres
Sybase Adaptive Server	APP_HOME/jdbc/sybase
Derby (Java DB)	APP_HOME/lib

## Connections

To access a database or a data source such as Excel spreadsheet, you must first create and setup a connection. Connection describes what type of database or data source it is, which JDBC or other type of driver to use, where it is running, which account to use and more.

The **Connections** node is a root for all database connections. It cannot be deleted or modified. When within a Connections segment of the nodes tree click on the “add node” button to add a new connection or connection’s folder. Use “delete node” to delete currently selected connection or connection’s folder. When you delete a connection’s folder all underlying connections will be deleted as well.

Use copy , cut and paste buttons to create a new connection from the existing one. Use refresh to refresh a connection or any underlying node and close to close connection node (and all associated database connections if any exist).

When new connection is created or existing connection is selected in the nodes browser the Connection Editor is displayed in the App panel.

You can create a database connection (connection to the database server, such as Oracle) or data source connection (text, XML or Excel file) by using different Connectors.

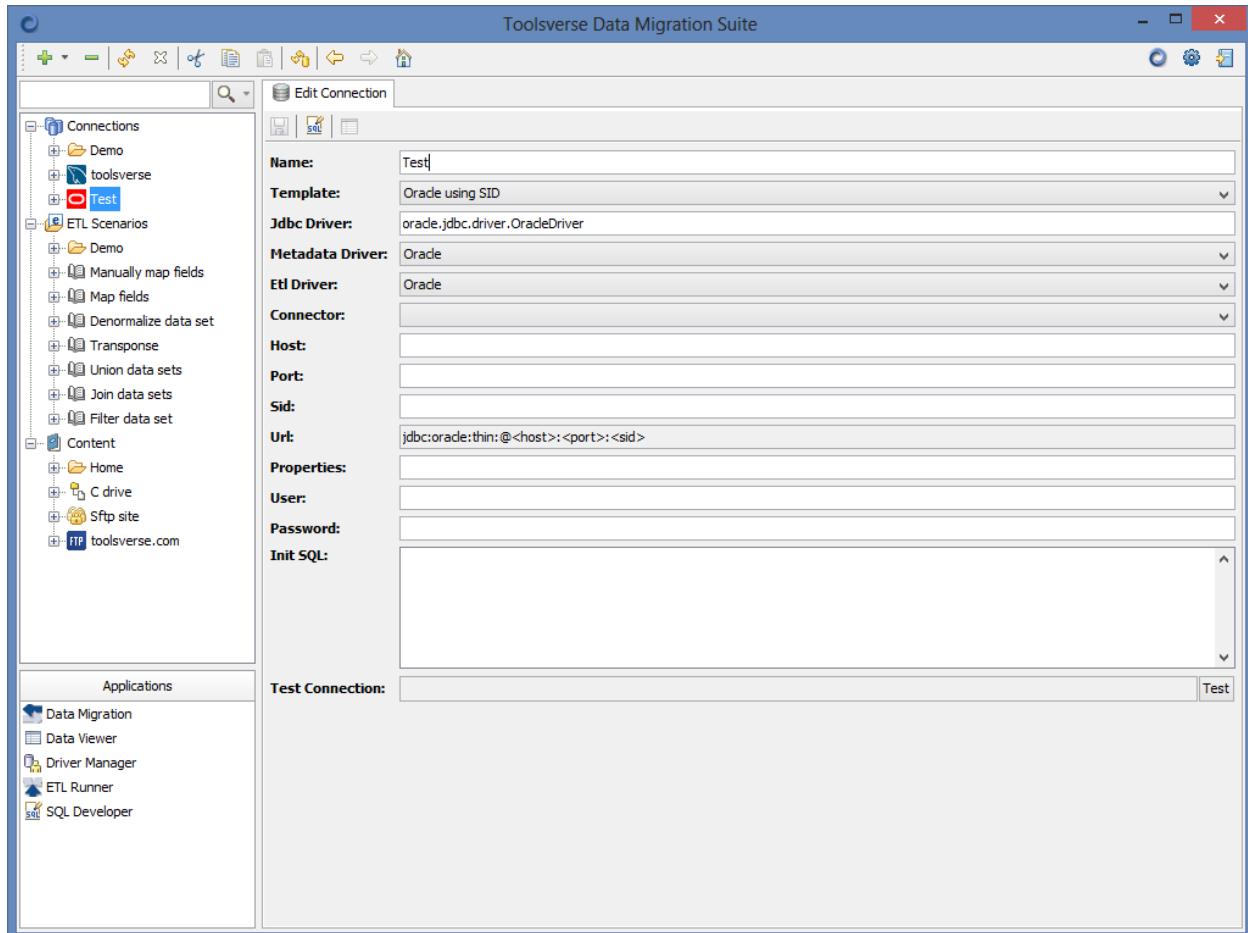


Figure 10: Connection Editor

## Commands

Icon	Command
	Save changes. Disabled until change is made
	Open SQL Editor for the selected connection. Disabled if connection does not support SQL, for example XML connection
	Test Connection - tests database connection. "Test Connection" is disabled for the file based data sources such as text, XML, Excel, etc

## Fields

Field	Description
Name	The name of the connection. Cannot be empty
Template	Choose a driver template from the drop down list. It will automatically populate all other fields. The driver templates are created in the Driver Manager application
Jdbc Drriver	The class name of the Jdbc driver. For example oracle.jdbc.driver.OracleDriver. This field is automatically populated when driver template is selected in the Driver drop down list
Metadata Driver	The name of the driver used to discover database metadata. This field is automatically populated when driver template is selected in the Driver drop down list
ETL Driver	The name of the ETL driver. The ETL driver is used by various add-ons to perform tasks specific for the particular database. This field is automatically populated when driver template is selected in the Driver drop down list
Connector	The name of the data source connector. The Connector is an object responsible for populating and persisting of the data sets. This field is automatically populated when driver template is selected in the Driver drop down list. If this field is empty the regular SQL database connector will be used

Url	The connection's url. For example jdbc:oracle:thin:@localhost:1521:test
Properties	The connection properties. For example informixserver=inf1
User	The user id used to create a connection
Password	The user password used to create a connection
Init SQL	The SQL statement which will be executed each time connection is created. This field works only for SQL-based connections.
Test Connection	Tests the connection.

## Database Connection

Database connection is a connection to the database which supports SQL. Database connection uses JDBC or ODBC driver. To configure database connection you need to enter at least JDBC driver class name and Url. Other fields, including Connector are optional depending on the database type.

**Note:** There are pre-configured connection templates for some databases. Select the one that you need from the **Driver** drop down to populate all required fields.

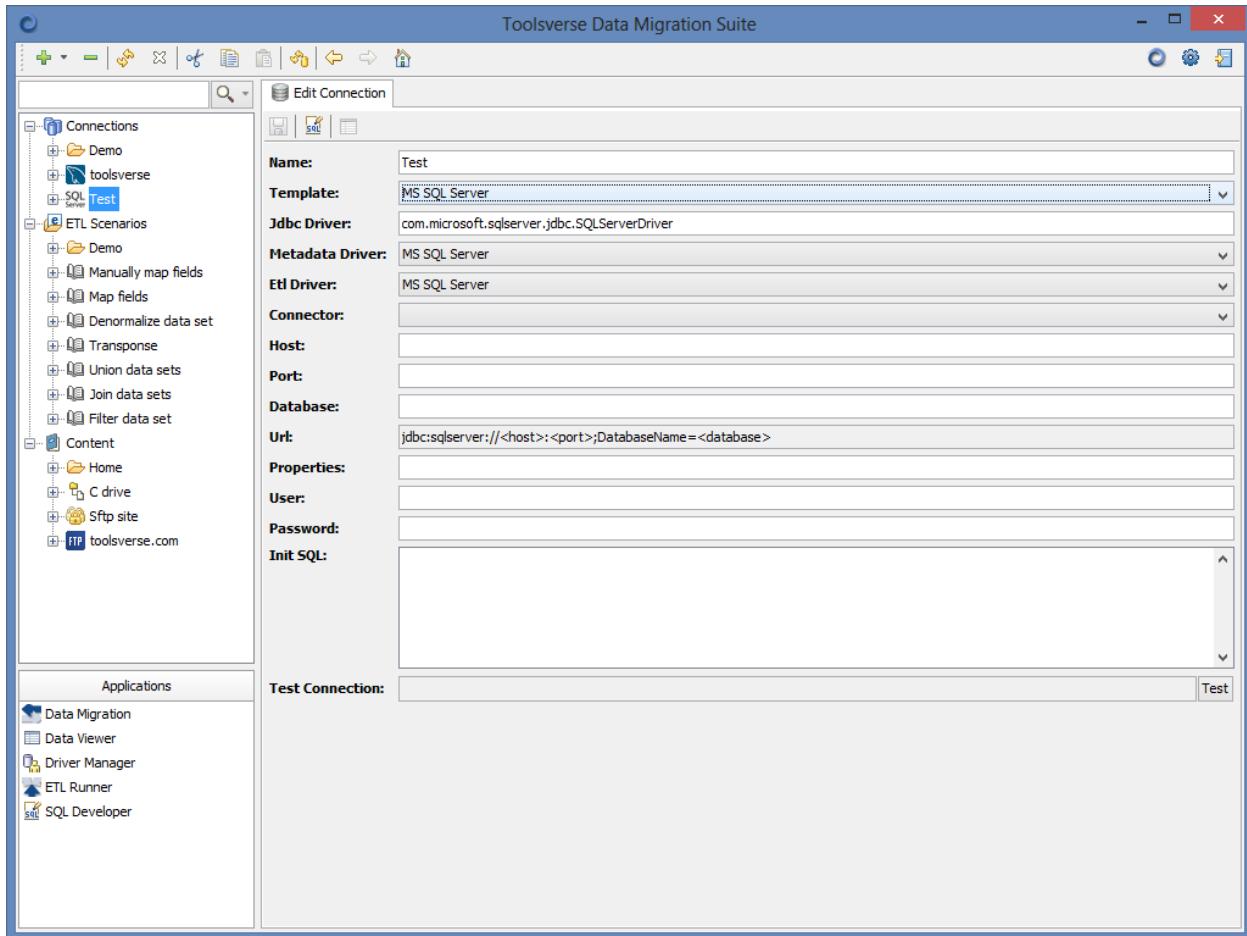


Figure 11: Database connection

If Url points to the file or folder in the file system [system variables](#) can be used as a part of the Url.

## Excel Connection using ODBC

“Excel Connection using ODBC” is a flavor of the database connection. To create an “Excel Connection using ODBC” you will need to the use pre-configured template **Excel ODBC**. Select it from the **Template** drop down to populate all required fields.

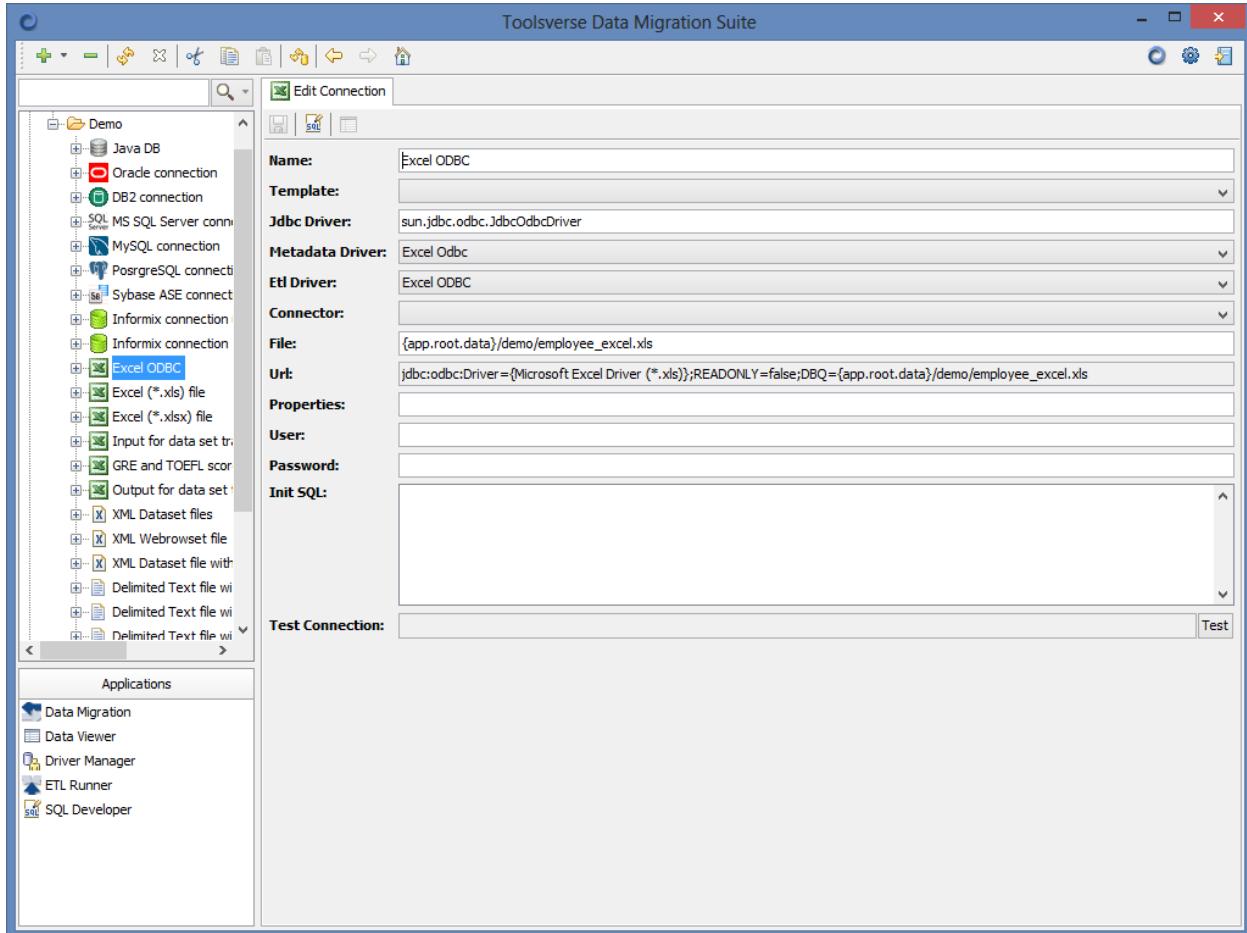


Figure 12: Excel connection using ODBC

## Excel (\*.xls) connection

Create an Excel (\*.xls) connection if you need to view data in the Excel (1997-2003) xls format. Excel connection can be used in the ETL scenarios as well. You cannot use SQL Developer with the Excel connections, however you can browse files and worksheets and view data in the worksheets. To create an Excel (\*.xls) connection you will need.

To create an “Excel (\*.xls) connection” you will need to use pre-configured template **Excel (\*.xls)**. Select it from the **Template** drop down to populate all required fields.

If Url points to the file or folder in the file system [system variables](#) can be used as a part of the Url.

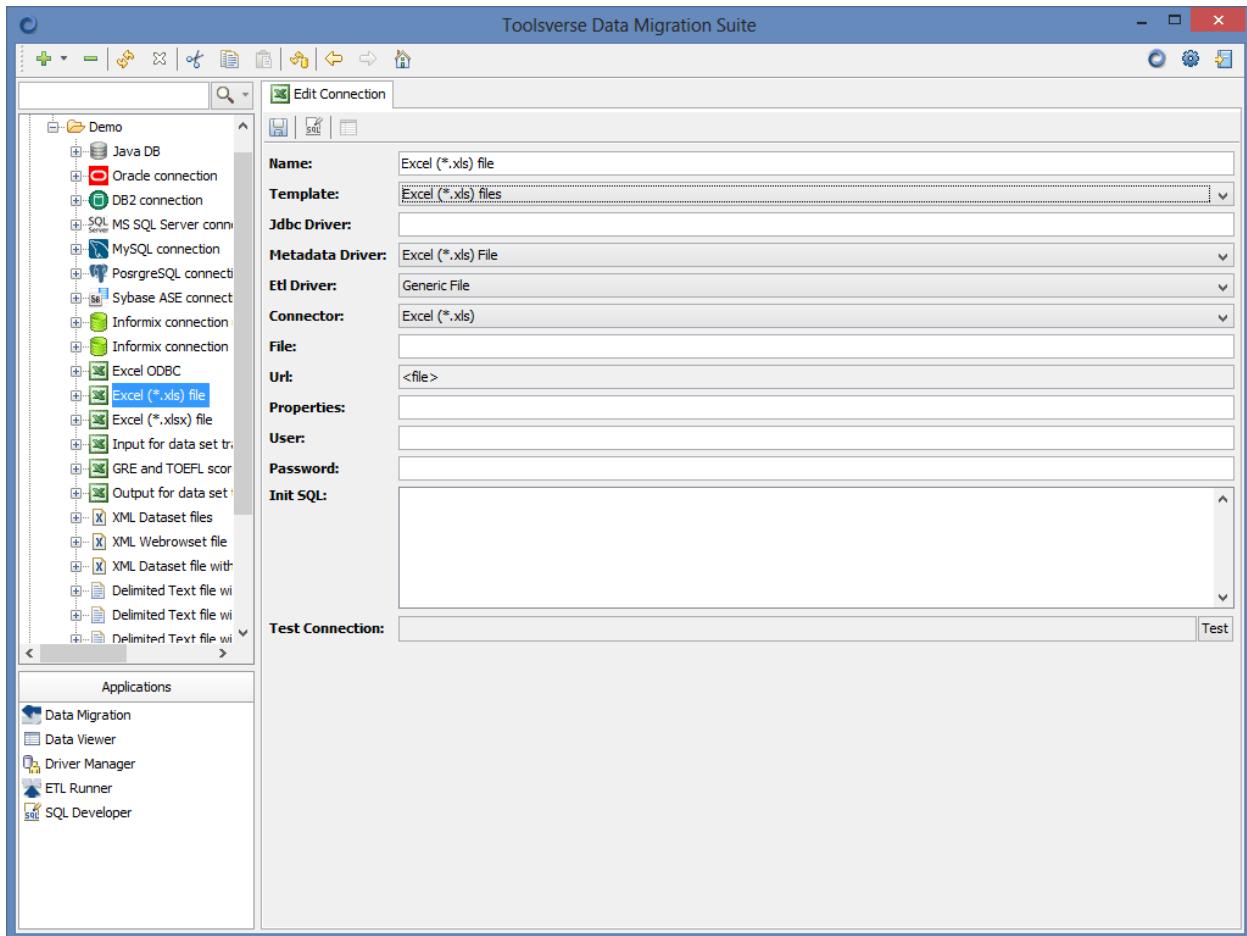


Figure 13: Excel (\*.xls) connection

You can use a wildcard for the URL. For example: {app.root.data}/demo/emp\_\*.xls.

If you are planning to work with the particular worksheet inside Excel spreadsheet, use property **sheetname**.

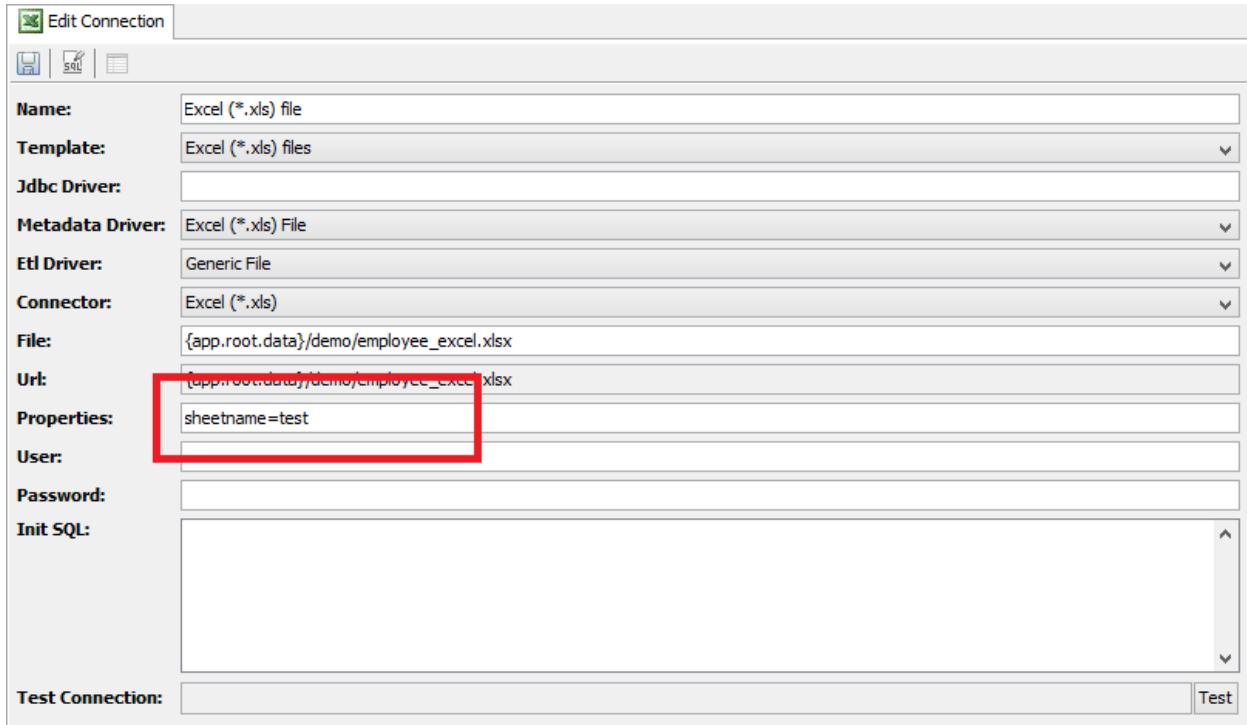


Figure 14: Excel worksheet name

List of all properties (you can put them together in the Properties field using semicolon):

Property	Description	Example	Default
sheetname	The name of worksheet in the spreadsheet	sheetname=Employee	None
date	Date format	date=MMddyyyy	system defined
datetime	Date+time format	datetime=MM/dd/yyyy HH:mm	system defined
time	Time format	time=HH:mm	system defined

**Example:** sheetname=Employee;date=MMddyyyy;datetime=MMddyyyy;time=HH:mm

## Excel (\*.xlsx) connection

Create Excel (\*.xlsx) connection if you need to view data in the Excel (2007 and above) xlsx format. Excel (\*.xlsx) connection can be used in the ETL scenarios as well. You cannot use SQL Developer with the Excel connections, however you can browse files and worksheets and view data in the worksheet.

To create an “Excel (\*.xlsx) connection” you will need to use pre-configured template **Excel (\*.xlsx)**. Select it from the **Template** drop down to populate all required fields.

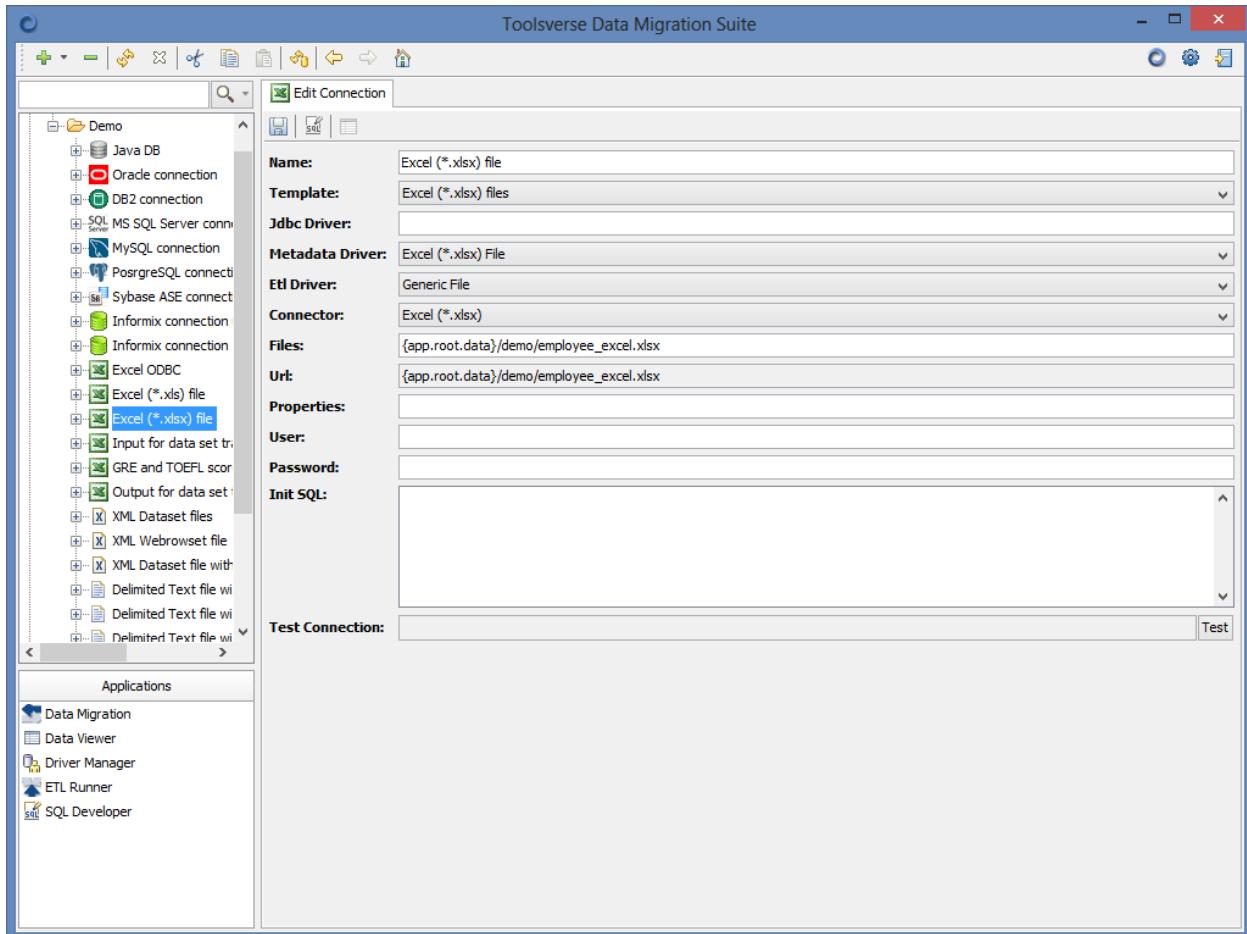


Figure 15: Excel (\*.xlsx) connection

You can use a wildcard for the URL. For example: {app.root.data}/demo/emp\_\*.xlsx.

If Url points to the file or folder in the file system [system variables](#) can be used as a part of the Url.

If you are planning to work with the particular worksheet inside Excel spreadsheet, use property **sheetname**.

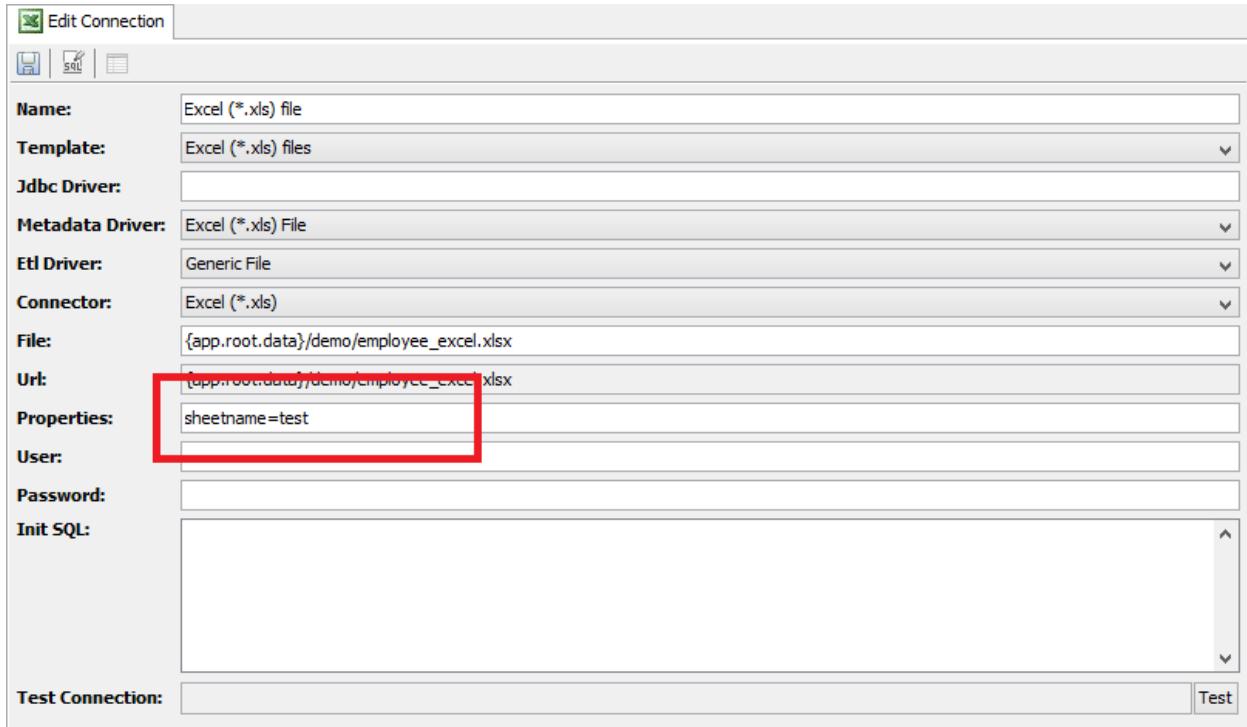


Figure 16: Excel (\*.xlsx) worksheet name

List of all properties (you can put them together in the Properties field using semicolon):

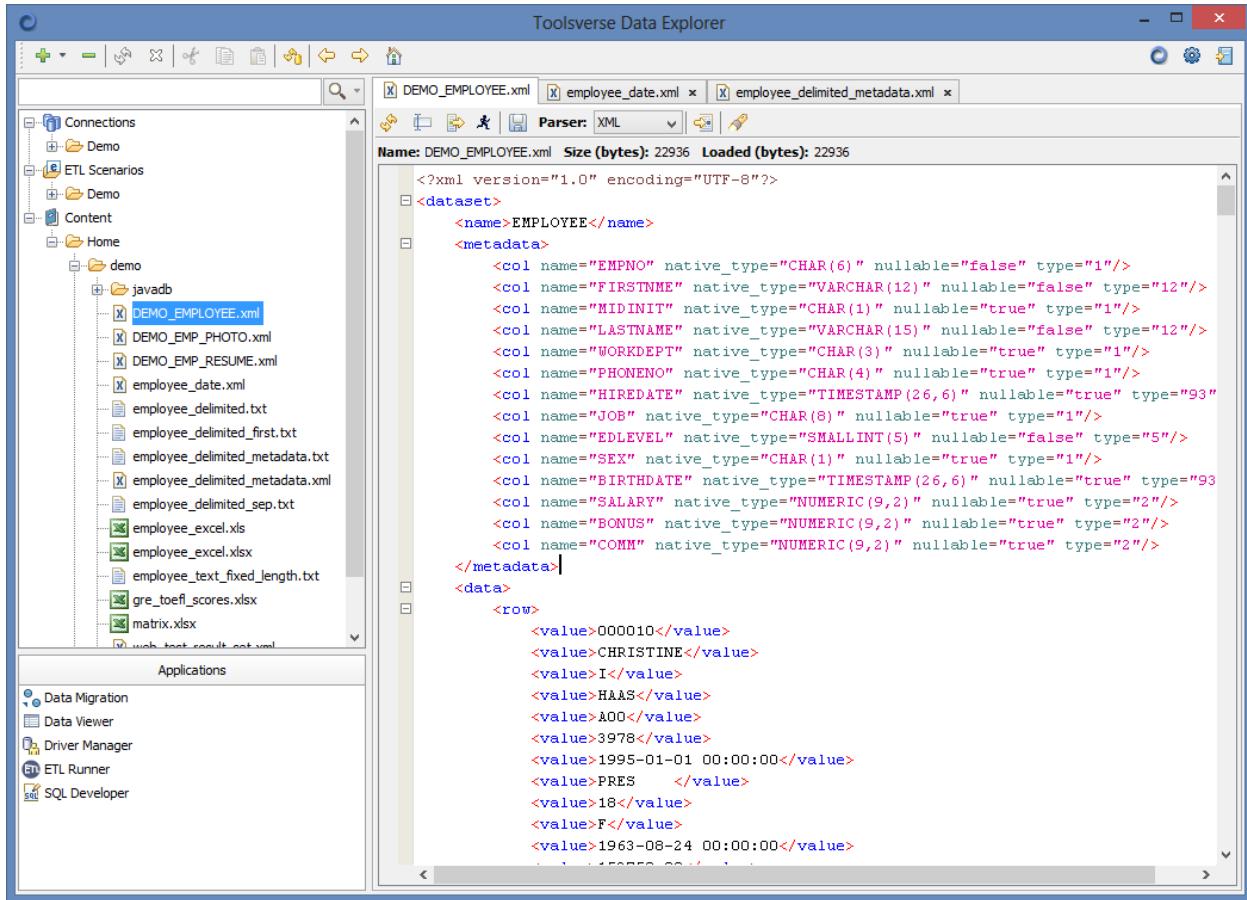
Property	Description	Example	Default
sheetname	The name of worksheet in the spreadsheet	sheetname=Employee	None
date	Date format	date=MMddyyyy	system defined
datetime	Date+time format	datetime=MM/dd/yyyy HH:mm	system defined
time	Time format	time=HH:mm	system defined

**Example:** sheetname=Employee;date=MMddyyyy;datetime=MMddyyyy;time=HH:mm

## XML Connection

XML connection provides an access to the files in the internal Data Explorer XML format called XML dataset. It used by default by the ETL framework to serialize/de-serialize data. You can find schema for the XML dataset format in the APP\_HOME/data/schema/xmldataset.xsd.

Example of the file in the XML dataset format:



The screenshot shows the Toolsverse Data Explorer interface. On the left, there's a tree view of connections, scenarios, and content. Under 'Content' > 'Home' > 'demo' > 'javadb', several XML files are listed: DEMO\_EMPLOYEE.xml, DEMO\_EMP\_PHOTO.xml, DEMO\_EMP\_RESUME.xml, employee\_date.xml, employee\_delimited.txt, employee\_delimited\_first.txt, employee\_delimited\_metadata.xml, employee\_delimited\_sep.txt, employee\_excel.xls, employee\_excel.xlsx, employee\_text\_fixed\_length.txt, gre\_toefl\_scores.xls, matrix.xlsx, and web\_test\_result.out.xml. On the right, the main pane displays the XML code for 'DEMO\_EMPLOYEE.xml'. The code defines a dataset with a name of 'EMPLOYEE', a metadata section mapping columns to database types like CHAR(6) for EMPNO and VARCHAR(12) for FIRSTNAME, and a data section containing a single row of employee data.

```

<?xml version="1.0" encoding="UTF-8"?>
<dataset>
    <name>EMPLOYEE</name>
    <metadata>
        <col name="EMPNO" native_type="CHAR(6)" nullable="false" type="1"/>
        <col name="FIRSTNAME" native_type="VARCHAR(12)" nullable="false" type="12"/>
        <col name="MIDINIT" native_type="CHAR(1)" nullable="true" type="1"/>
        <col name="LASTNAME" native_type="VARCHAR(15)" nullable="false" type="12"/>
        <col name="WORKDEPT" native_type="CHAR(3)" nullable="true" type="1"/>
        <col name="PHONEINO" native_type="CHAR(4)" nullable="true" type="1"/>
        <col name="HIREDATE" native_type="TIMESTAMP(26,6)" nullable="true" type="93"/>
        <col name="JOB" native_type="CHAR(8)" nullable="true" type="1"/>
        <col name="EDLEVEL" native_type="SMALLINT(5)" nullable="false" type="5"/>
        <col name="SEX" native_type="CHAR(1)" nullable="true" type="1"/>
        <col name="BIRTHDATE" native_type="TIMESTAMP(26,6)" nullable="true" type="93"/>
        <col name="SALARY" native_type="NUMERIC(9,2)" nullable="true" type="2"/>
        <col name="BONUS" native_type="NUMERIC(9,2)" nullable="true" type="2"/>
        <col name="COMM" native_type="NUMERIC(9,2)" nullable="true" type="2"/>
    </metadata>
    <data>
        <row>
            <value>000010</value>
            <value>CHRISTINE</value>
            <value>I</value>
            <value>HAAS</value>
            <value>A00</value>
            <value>3978</value>
            <value>1995-01-01 00:00:00</value>
            <value>PRES </value>
            <value>18</value>
            <value>F</value>
            <value>1963-08-24 00:00:00</value>
        </row>
    </data>

```

Figure 17: XML dataset

The XML connection can be used in the ETL scenarios. You cannot use SQL Developer with the XML connections; however you can browse files and view data in the files.

To create an “XML connection” you will need to the use pre-configured template **XML Files**. Select it from the **Template** drop down to populate all required fields.

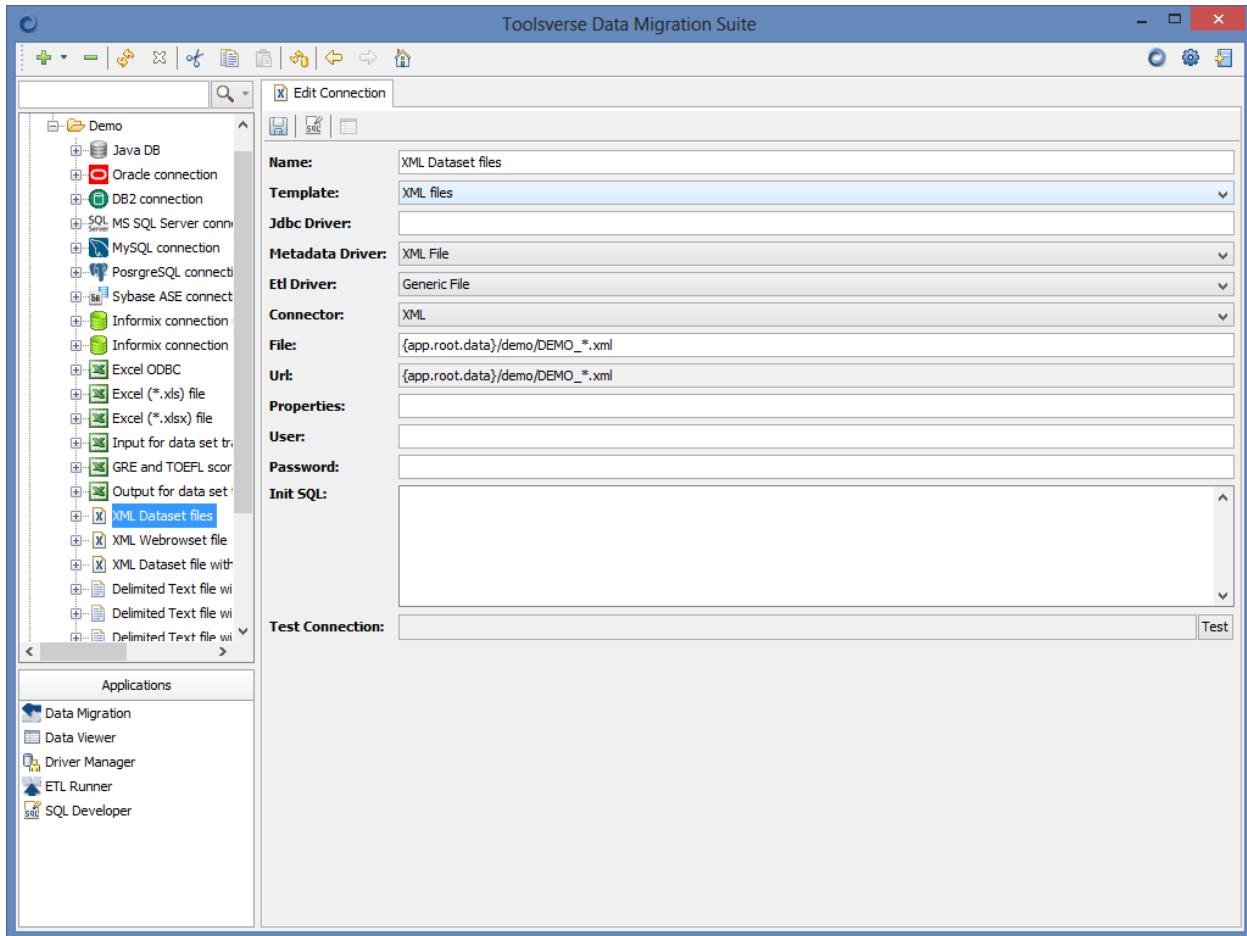


Figure 18: XML connection

You can use a wildcard for the Url. For example: {app.root.data}/demo/demo\_\* .xml.

If Url points to the file or folder in the file system [system variables](#) can be used as a part of the Url.

List of all properties (you can put them together in the Properties field using semicolon):

Property	Description	Example	Default
date	Date format	date=MMddyyyy	system defined
datetime	Date+time format	datetime=MM/dd/yyyy HH:mm	system defined
time	Time format	time=HH:mm	system defined

**Example:** date=MMddyyyy;datetime=MMddyyyy;time=HH:mm

## XML Connection with Transformation

If you need to view data in the XML format different from the XML dataset you can use XML connection with Transformation. You must have a XSL transformation style sheet. Please see examples of the XML dataset to WebRowSet and WebRowSet to XML dataset style sheets in APP\_HOME/data/schema folder. The WebRowSet is an XML document representation of a JDBC result set which was introduced by Sun in JDK 1.5. You can use XML connection with Transformation in the ETL scenarios as well.

XML connection with Transformation is a basically XML connection with a property which defines XSL transformation style sheet. It can be used in the ETL scenarios.

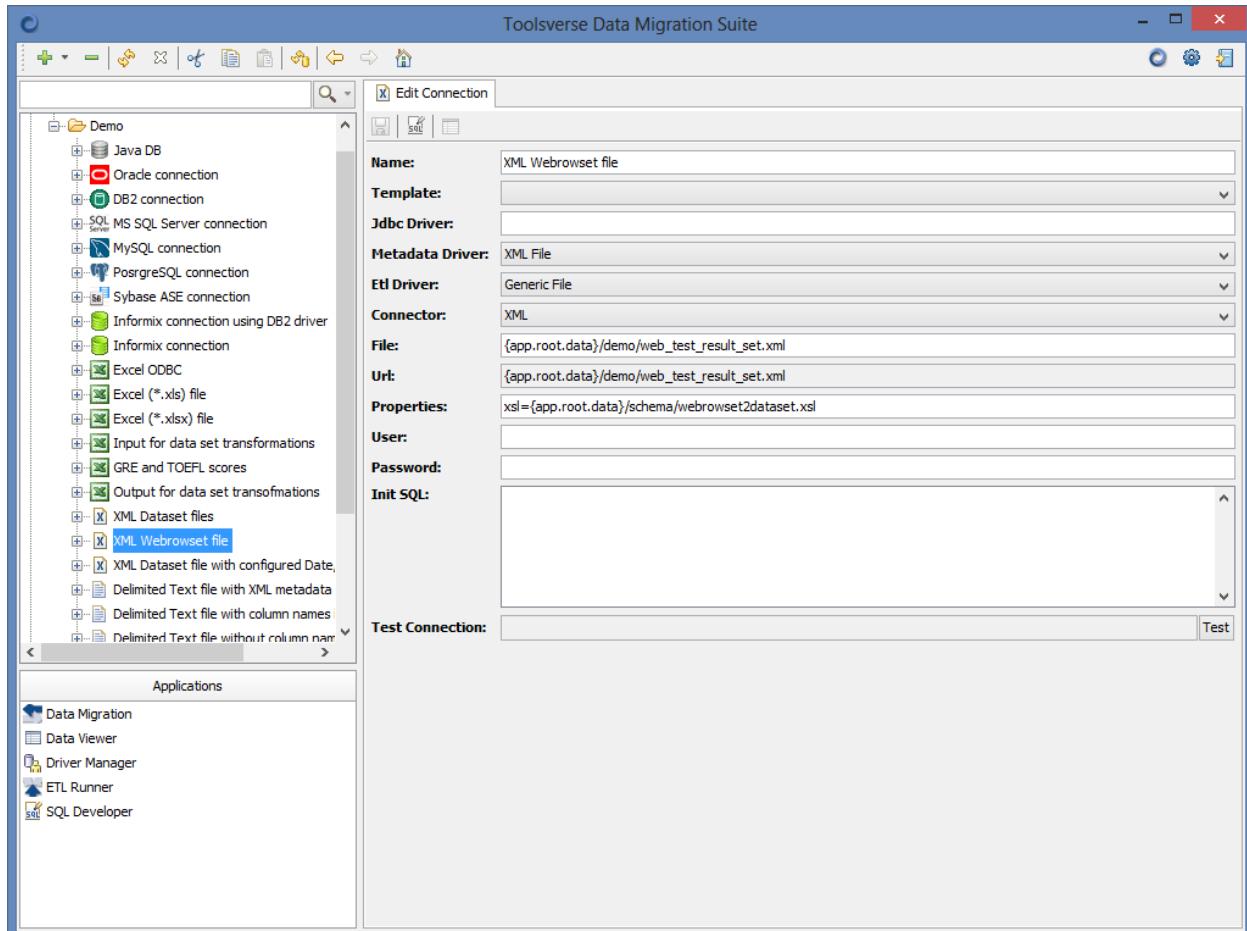


Figure 19: XML connection with Transformation

You can use a wildcard for the Url. For example: {app.root.data}/demo/demo\_\*.xml.

If Url points to the file or folder in the file system [system variables](#) can be used as a part of the Url.

List of all properties (you can put them together in the Properties field using semicolon):

Property	Description	Example	Default
xsl	Name of the xlst scenario file	xsl={app.root.data}/schema/webrowset2dataset.xsl	None
xslfrom	Name of the xlst scenario file used to transform <b>from</b> other XML format to XML dataset	xslfrom={app.root.data}/schema/dataset2webrowset.xsl	None
xsltto	Name of the xlst scenario file used to transform to other XML format from XML dataset	xslfrom={app.root.data}/schema/webrowset2dataset.xsl	None
date	Date format	date=MMddyyyy	system defined
datetime	Date+time format	datetime=MM/dd/yyyy HH:mm	system defined
time	Time format	time=HH:mm	system defined

**Example:**

xsl={app.root.data}/schema/webrowset2dataset.xsl;date=MMddyy;datetime=MMddyyyy;time=HH:mm

## Delimited Text File Connection

You can create a connection to the delimited text file using wide range or properties. You cannot use SQL Developer with the Delimited text file connections; however you can browse files and view data in files. It can be used in the ETL scenarios.

To create an “Delimited Text file connection” you will need to use pre-configured template **Text Files**. Select it from the **Template** drop down to populate all required fields.

Below is an example of the connection for the delimited text file:

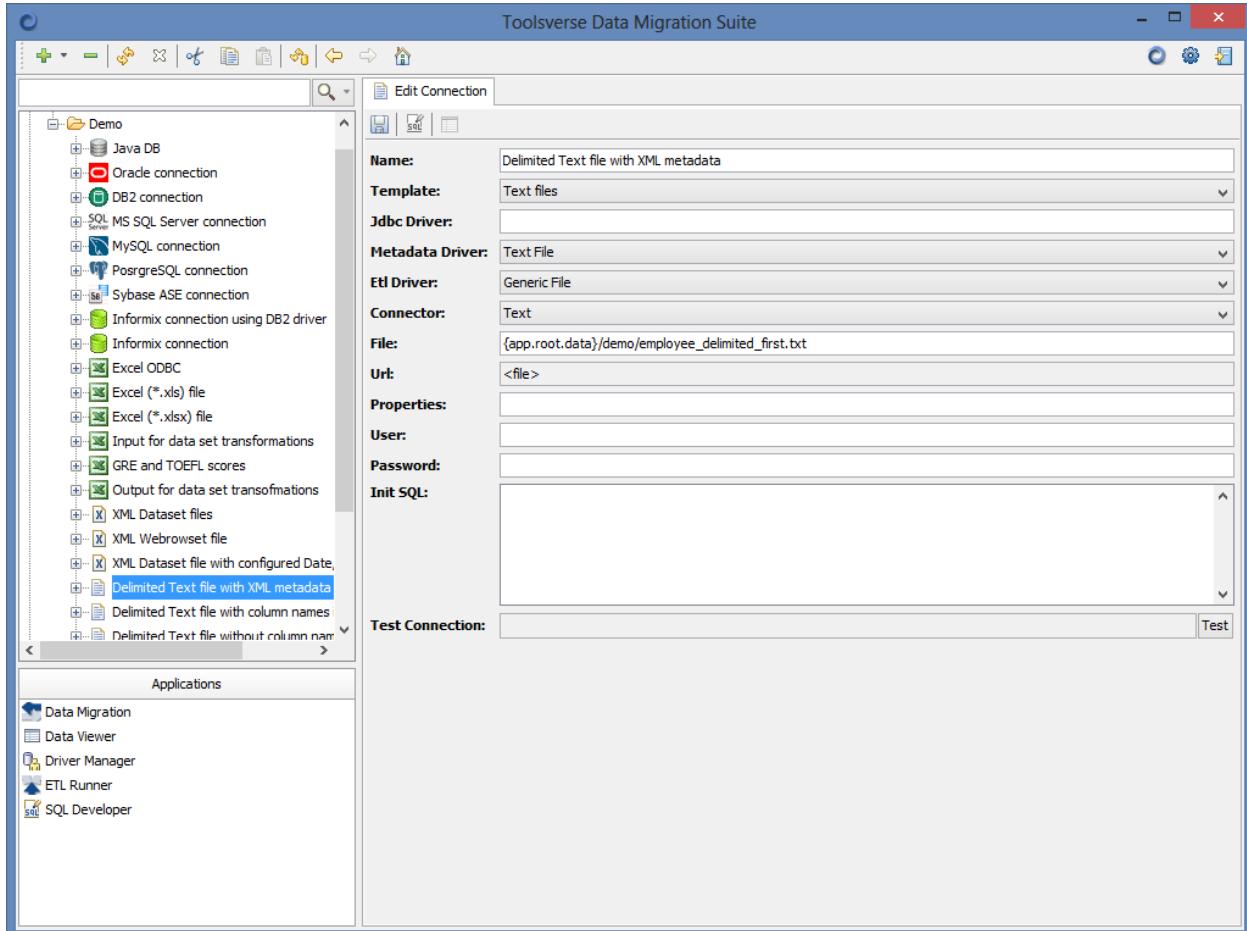


Figure 20: Delimited text file connection

You can use a wildcard for the Url. For example: {app.root.data}/demo/demo\_\*.txt.

If Url points to the file or folder in the file system [system variables](#) can be used as a part of the Url.

List of all properties (you can put them together in the Properties field using semicolon):

Property	Description	Example	Default
delimiter	The field delimiter	delimiter=';'	' '
firstrow	Use first row for data	firstrow=false	true

metadata	Store metadata in XML dataset format	metadata=false	false
charseparator	The character used to enclose string values into	charseparator=';'	Nothing
lineseparator	The separator between lines	<p>lineseparator=w.</p> <p>Possible values:</p> <p>s – os default</p> <p>w – windows</p> <p>u - unix</p>	s
date	Date format	date=MMddyyyy	system defined
datetime	Date+time format	datetime=MM/dd/yyyy HH:mm	system defined
time	Time format	time=HH:mm	system defined

**Example:** delimiter=';';charseparator="";firstrow=false

## Fixed Length Text File Connection

Fixed length text file connection is created by using the same drivers and connector as Delimited text file connection. You cannot use SQL Developer with the Fixed Length Text File connections; however you can browse files and view data in files. It can be used in the ETL scenarios.

The **fields** property is what differentiates it from the Delimited text file connection. The **fields** property must include a length of the each field in the data set. The numbers must be delimited by value of the **delimiter** property. Example: `delimiter='';fields='6;12;15;8'` defines a file with a 4 fields with a length 6, 12,15 and 8 respectfully.

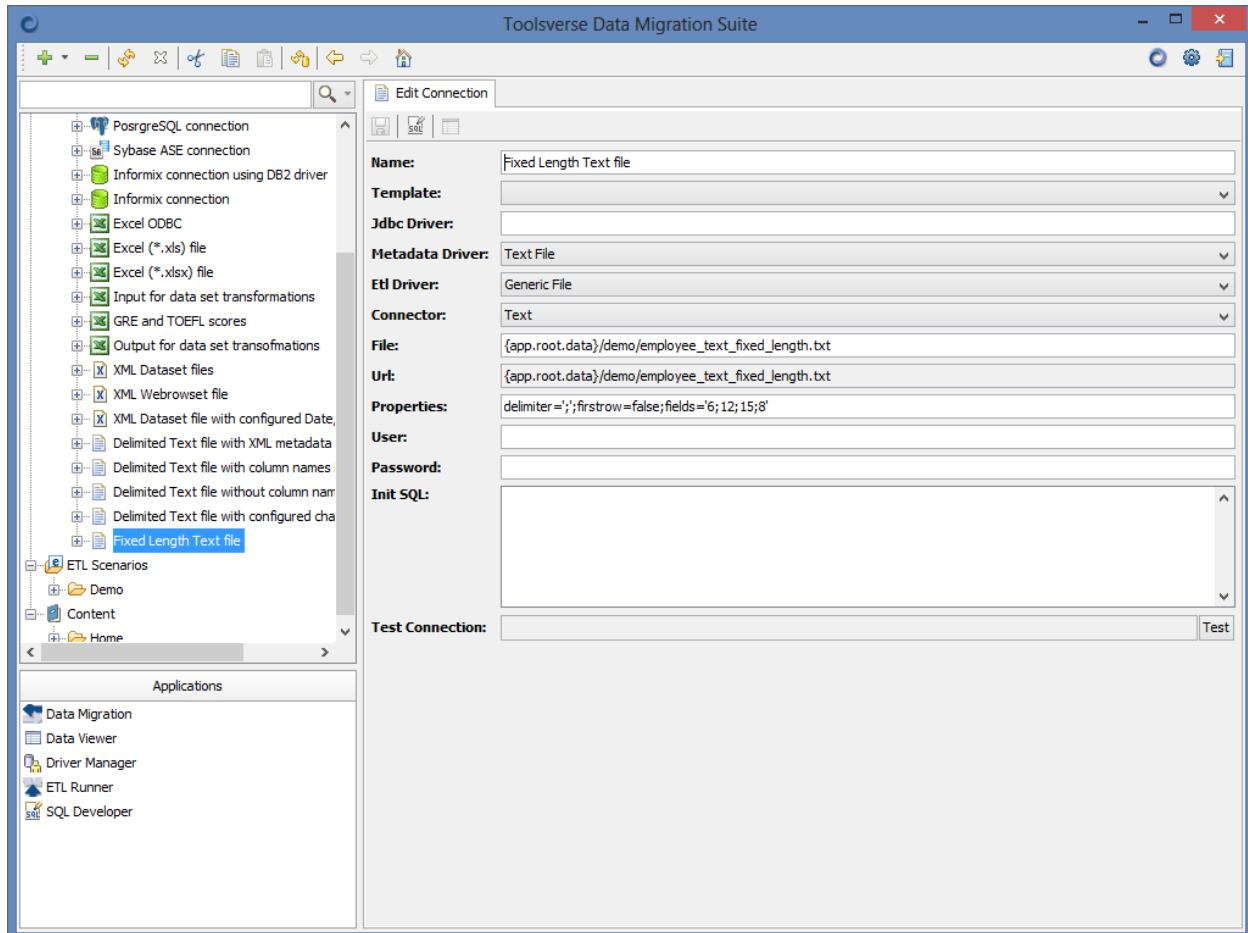


Figure 21: Fixed length text file connection

You can use a wildcard for the Url. For example: `{app.root.data}/demo/demo_*.txt`.

If Url points to the file or folder in the file system [system variables](#) can be used as a part of the Url.

List of all properties (you can put them together in the Properties field using semicolon):

Property	Description	Example	Default
delimiter	The field delimiter	<code>delimiter=''</code>	<code>' '</code>

fields	The length of the each field in the data set	fields='6;12;15;8'	none
firstrow	Use first row for data	firstrow=false	true
metadata	Store metadata in XML dataset format	metadata=false	false
chardata	The character used to enclose string values into	chardata=';'	Nothing
lineseparator	The separator between lines	lineseparator=w.  Possible values:  s – os default  w – windows  u - unix	s
date	Date format	date=MMddyyyy	system defined
datetime	Date+time format	datetime=MM/dd/yyyy HH:mm	system defined
time	Time format	time=HH:mm	system defined

**Example:** delimiter=';';firstrow=false;fields='6;12;15;8

### Using variables in the URLs

You can use variables {app.root.data} and {app. data} as a path component in the URL. For example: jdbc:derby:{app.root.data}/demo/javadb. Please see definition of these variables [here](#).

## Browse Database Objects

To expand connection and explore underlying database objects, such as tables, views, etc click expand (+) button in the nodes browser. The objects displayed in the tree are different for different databases and data sources.

When database object is selected in the nodes browser Data Explorer displays all available metadata. For example for the table these are columns, constraints, triggers, etc.

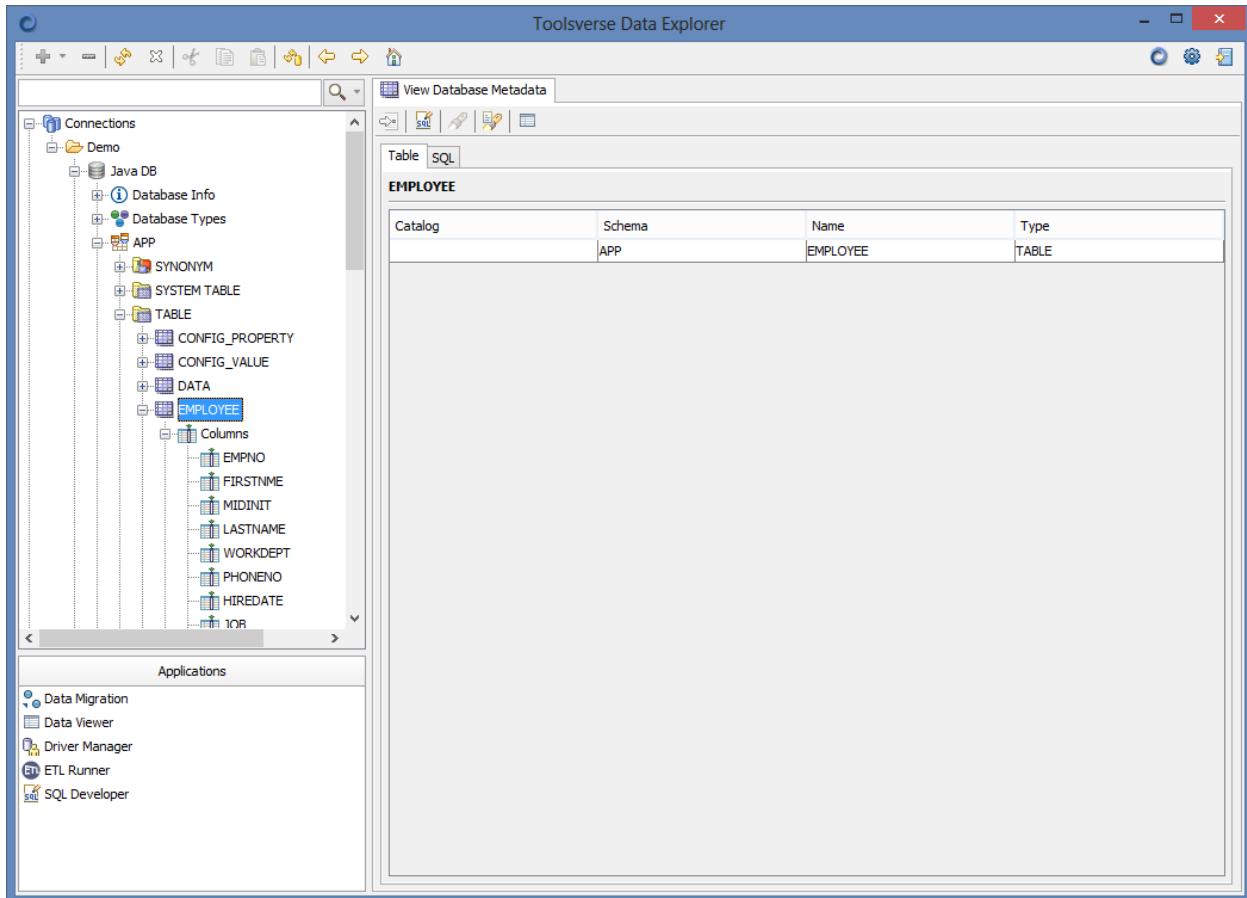


Figure 22: Database Objects

Some objects, such as tables, views, stores procedures, etc (depending on used metadata driver) can be displayed as SQL. If SQL is available for the object - the SQL tab is displayed.

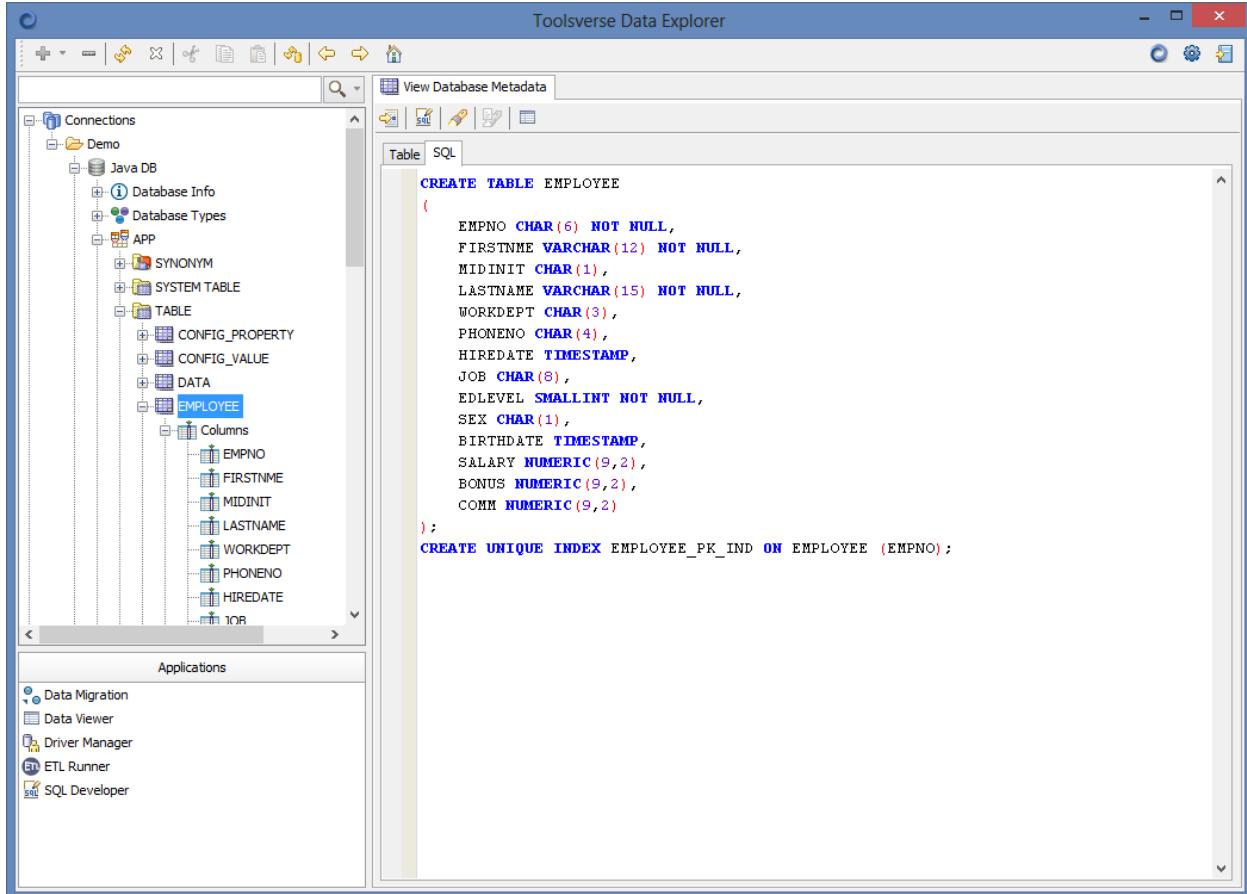


Figure 23: Table DDL

### *View Database Metadata Commands*

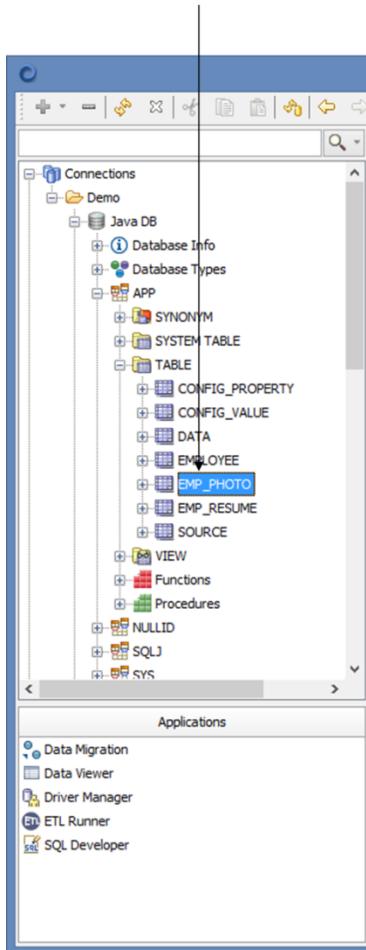
When Data Explorer is in the "view database metadata" mode the following commands are available:

Icon	Function	Windows and others	OS X	Web browser
	Open SQL Developer	Alt+F9	Alt+F9	Alt+F9
	Search in the Data Set. The data set is a currently displayed grid	F7	F7	F7
	View Data - opens a Data Viewer application. Used to display data in table/view/synonymous.	Alt+F10	Alt+F10	Alt+F10

## View Table Data (table/view/synonym/etc)

When node, selected in the nodes browser is a database table, view, synonym (or other table type) or a text or XML file or an Excel spreadsheet, the "View Data" command is available in the "View Database Metadata" tab. When you click on the "view data"  button the new "Data Viewer" tab is added to the App panel.

1. Select table/view/synonym to view



2. Click on this button to open Data Viewer

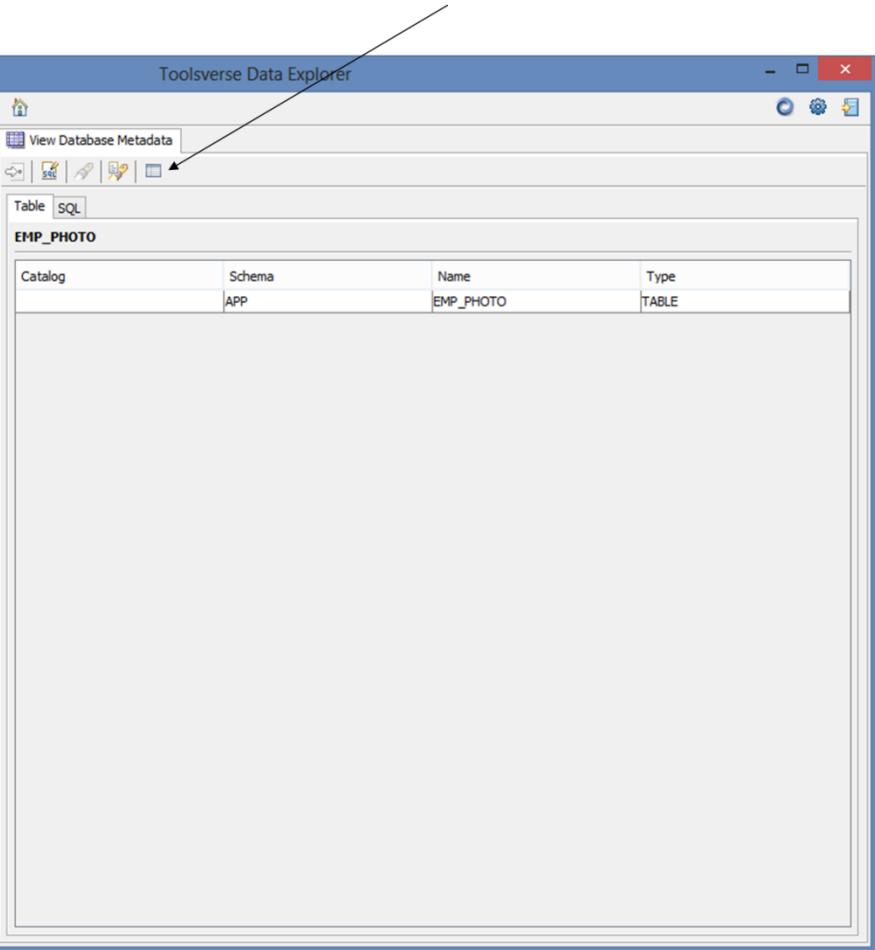


Figure 24: Select a table

By default the data will be displayed automatically. However if Settings->Data Viewer App->Preview Data Set is unchecked you will need to click on "refresh"  button to see data.

The screenshot shows the Toolsverse Data Explorer interface. On the left, the 'Connections' tree view displays a connection to 'Java DB' under 'Demo'. The 'Java DB' node has several sub-nodes: Database Info, Database Types, APP, SYSTEM TABLE, TABLE, VIEW, Functions, Procedures, NULLID, SQL, and SYS. The 'APP' node is expanded, showing SYNONYM, SYSTEM TABLE, and TABLE. The 'TABLE' node is expanded, showing CONFIG\_PROPERTY, CONFIG\_VALUE, DATA, EMPLOYEE, EMP\_PHOTO, EMP\_RESUME, SOURCE, and VIEW. The 'EMP\_PHOTO' node is selected and highlighted in blue. On the right, the 'Data Viewer' tab is active, showing a table titled 'Java DB:APP.EMP\_PHOTO'. The table has three columns: EMPNO, PHOTO\_FORMAT, and PICTURE. The data consists of eight rows:

EMPNO	PHOTO_FORMAT	PICTURE
000130	bitmap	Blob
000130	gif	Blob
000140	bitmap	Blob
000140	gif	Blob
000150	bitmap	Blob
000150	gif	Blob
000190	bitmap	Blob
000190	gif	Blob

Figure 25: Database table data

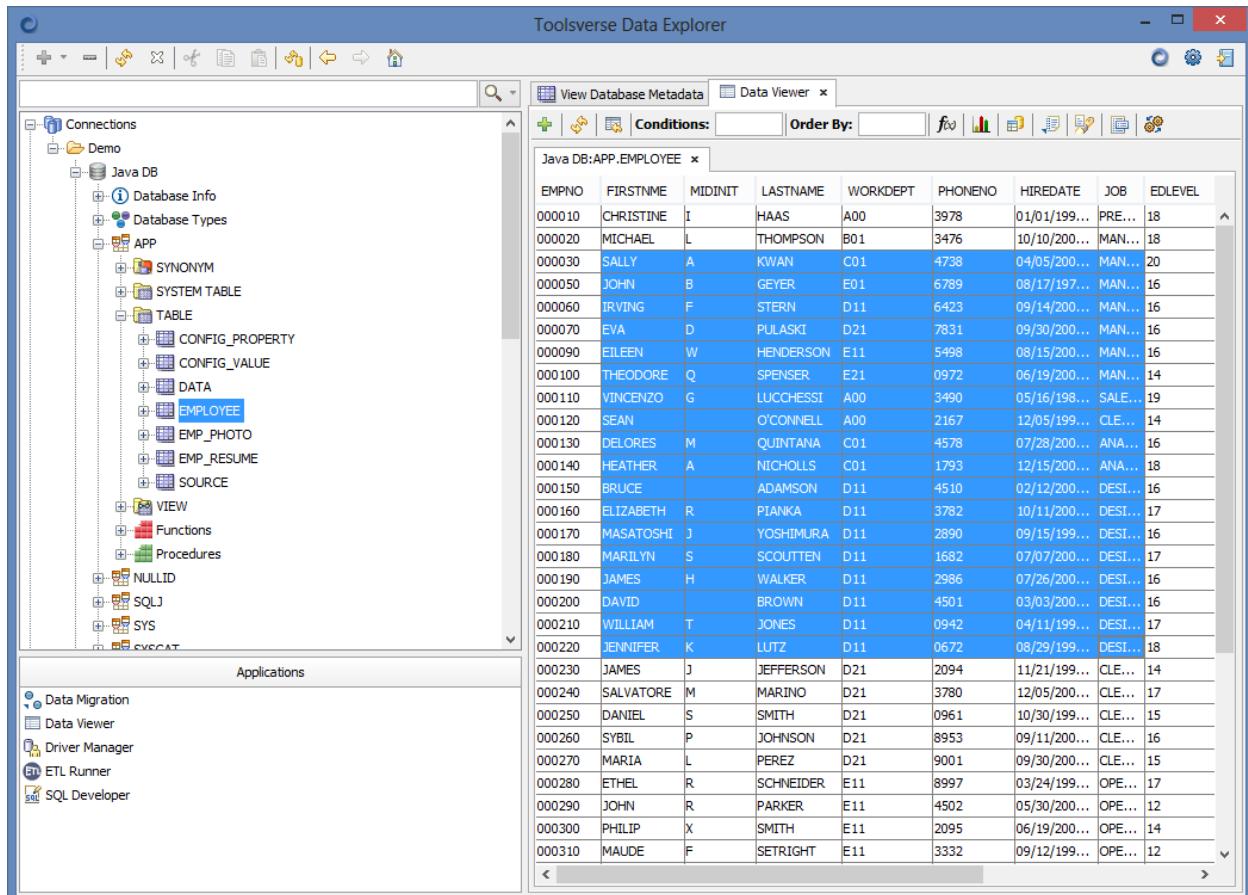
Please see [Data Viewer](#) for more details.

## Data Viewer

Data Viewer is a Data Explorer application which can display data in the different formats, including database tables, Excel spreadsheets, text and XML files. Multiple data sets can be viewed at the same time, each in its own tab.

Data sets can be displayed in the grid and form views and as charts. The “*where clause*” conditions can be applied to the data set as well as “*order by*” sort criteria.

To open Data Viewer click on "data viewer"  icon in the Applications list or expand Connection node to the particular table/view/synonym or Excel worksheet, text or XML file and click on "view data"  icon.



EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL
000010	CHRISTINE	I	HAAS	A00	3978	01/01/199...	PRES...	18
000020	MICHAEL	L	THOMPSON	B01	3476	10/10/200...	MANAG...	18
000030	SALLY	A	KWAN	C01	4738	04/05/200...	MANAG...	20
000050	JOHN	B	GEYER	E01	6769	08/17/197...	MANAG...	16
000060	IRVING	F	STERN	D11	6423	09/14/200...	MANAG...	16
000070	EVA	D	PULASKI	D21	7831	09/30/200...	MANAG...	16
000090	EILEEN	W	HENDERSON	E11	5498	08/15/200...	MANAG...	16
000100	THEODORE	Q	SPENSER	E21	0972	06/19/200...	MANAG...	14
000110	VINCENZO	G	LUCCHESI	A00	3490	05/16/198...	SALE...	19
000120	SEAN		O'CONNELL	A00	2167	12/05/199...	CLE...	14
000130	DELORES	M	QUINTANA	C01	4578	07/28/200...	ANA...	16
000140	HEATHER	A	NICHOLLS	C01	1793	12/15/200...	ANA...	18
000150	BRUCE		ADAMSON	D11	4510	02/12/200...	DESL...	16
000160	ELIZABETH	R	PIANKA	D11	3782	10/11/200...	DESL...	17
000170	MASATOSHI	J	YOSHIMURA	D11	2890	09/15/199...	DESL...	16
000180	MARYLN	S	SCOUTTEN	D11	1682	07/07/200...	DESL...	17
000190	JAMES	H	WALKER	D11	2986	07/26/200...	DESL...	16
000200	DAVID		BROWN	D11	4501	03/03/200...	DESL...	16
000210	WILLIAM	T	JONES	D11	0942	04/11/199...	DESL...	17
000220	JENNIFER	K	LUTZ	D11	0672	08/29/199...	DESL...	18
000230	JAMES	J	JEFFERSON	D21	2094	11/21/199...	CLE...	14
000240	SALVATORE	M	MARINO	D21	3780	12/05/200...	CLE...	17
000250	DANIEL	S	SMITH	D21	0961	10/30/199...	CLE...	15
000260	SYBIL	P	JOHNSON	D21	8953	09/11/200...	CLE...	16
000270	MARIA	L	PEREZ	D21	9001	09/30/200...	CLE...	15
000280	ETHEL	R	SCHNEIDER	E11	8997	03/24/199...	OPE...	17
000290	JOHN	R	PARKER	E11	4502	05/30/200...	OPE...	12
000300	PHILIP	X	SMITH	E11	2095	06/19/200...	OPE...	14
000310	MAUDE	F	SETRIGHT	E11	3332	09/12/199...	OPE...	12

Figure 26: Data Viewer

## Features

Please see a list of all Data Viewer features [here](#).

## Commands

Icon	Function	Windows and others	OS X	Web browser
	Opens a dialog window with a tree of available connections. You must expand connection node and find an object you are interested in. The "Ok" button is enabled only for the objects that can be displayed in the Data Viewer. For example database table or Excel worksheets can be stored procedure can not	Ctrl+N	Command+N	Shift+F4
	Refreshes data. When tab is initially opened for the object the data might not be displayed yet (depending on configuration). In this case you must click on "refresh" to see the data. Also you need to "refresh" when you change conditions or ordering	Ctrl+F2	Command+F2	Ctrl+F2
	<a href="#">Set Grid Defaults</a> such as maximum number of rows in the data set, etc	Ctrl+F9	Command+F9	Ctrl+F9
	Condition - the <i>where clause</i> . Example: lastname = 'STERN' or job = 'MANAGER'  <b>Note:</b> You must click on the "refresh"  button when you change conditions			
	Order by - the sort criteria. Example: firstname, lastname desc  <b>Note:</b> You must click on the "refresh"  button when you change <b>order by</b> .			
	<a href="#">Calculate function</a>	Alt+F12	Alt+F12	Alt+F12
	<a href="#">Display Data as a Chart</a>	Shift+F10	Shift+F10	Shift+F10
	<a href="#">Describe Data Set.</a>	Shift+F12	Shift+F12	Shift+F12
	<a href="#">Export Data Set.</a>	F9	F9	F9
	<a href="#">Search in the Data Set.</a>	F7	F7	F7

	<a href="#">Show Data Set Record in the form view.</a>	F4	F4	F4
	<a href="#">Transform Data Set</a>	Shift+F8	Shift+F8	Shift+F8

## Data Viewer How To

### *[View Table Data \(table/view/synonym/etc\)](#)*

See [View Table Data](#)

### *[Add new data set to view](#)*

Data Viewer can display multiple data sets, each in its own tab. They are not necessary database objects such as tables or views but also can be Excel worksheets, text and XML files, etc.

To add a new data set to the Data Viewer click on "select table"  button. When Select Node dialog window is displayed navigate to the object you are interested in. Click "Ok" button.

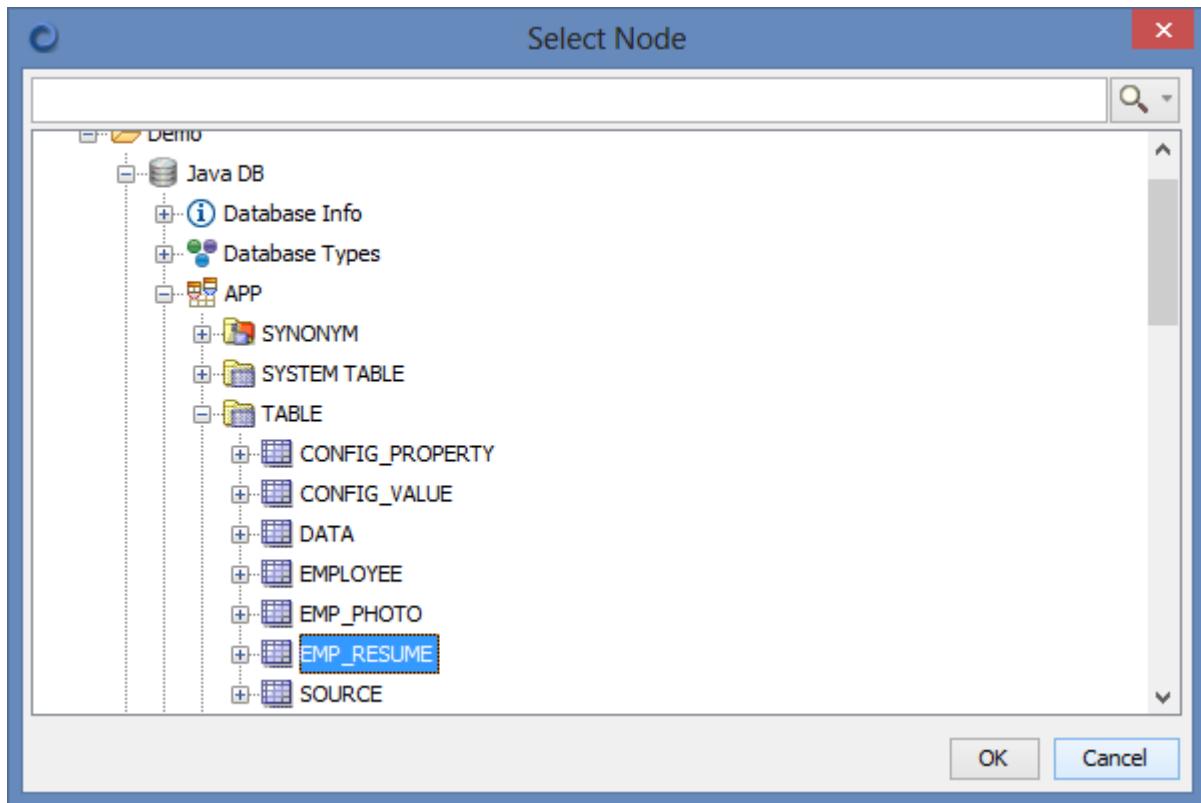


Figure 27: Select database table to view

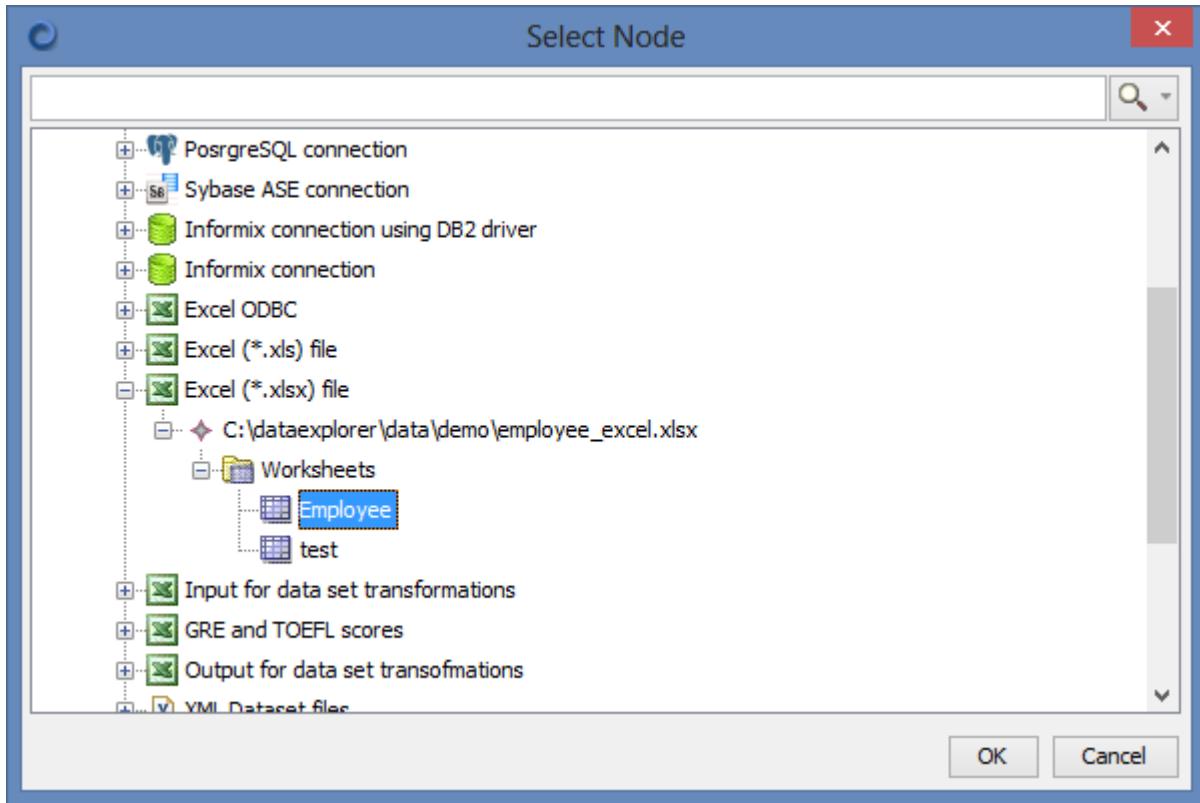


Figure 28: Select Excel worksheet to view

### ***View and refresh data***

Click on "refresh" button to refresh already displayed data set. You can change a maximum number of rows that can be displayed by setting [grid defaults](#).

### ***Change grid parameters***

See [Set Grid Defaults](#)

### ***Apply where clause conditions***

To apply where clause conditions to the data set enter those in the **Conditions** field and press "refresh" button. If you are viewing database object such as table, view, synonym, etc **Conditions** must be a valid SQL "where clause".

In the file mode this is a simple filter, similar to SQL where clause. You can use field names (case sensitive), logical operators: AND, OR, NOT, comparison operators: =,<=,<,> and round brackets: (, ). You cannot use advanced SQL operators such as IN, EXISTS, etc. The condition is evaluated by JavaScript engine so JavaScript functions are also allowed.

**Example:** (FIRSTNAME = 'John' and LASTNAME = 'Smith') or SEX.toUpperCase() = 'F'

The screenshot shows a database application interface with two tabs at the top: "Edit Connection" and "Data Viewer". The "Data Viewer" tab is active. Below the tabs, there are icons for creating a connection, opening a connection, closing a connection, and a search bar labeled "Conditions: lastname = 'STERN' or job in ('MANAGER', 'CLERK')". To the right of the search bar is an "Order By:" dropdown menu and several other toolbar icons.

The main area displays two tables:

- Java DB:APP.DATA**: This table has 15 rows of data. The columns are: EMPNO, FIRSTNAME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB, EDLEVEL, SEX, and BIRTHDATE.
- Java DB:APP.EMPLOYEE**: This table also has 15 rows of data. The columns are: EMPNO, FIRSTNAME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB, EDLEVEL, SEX, and BIRTHDATE.

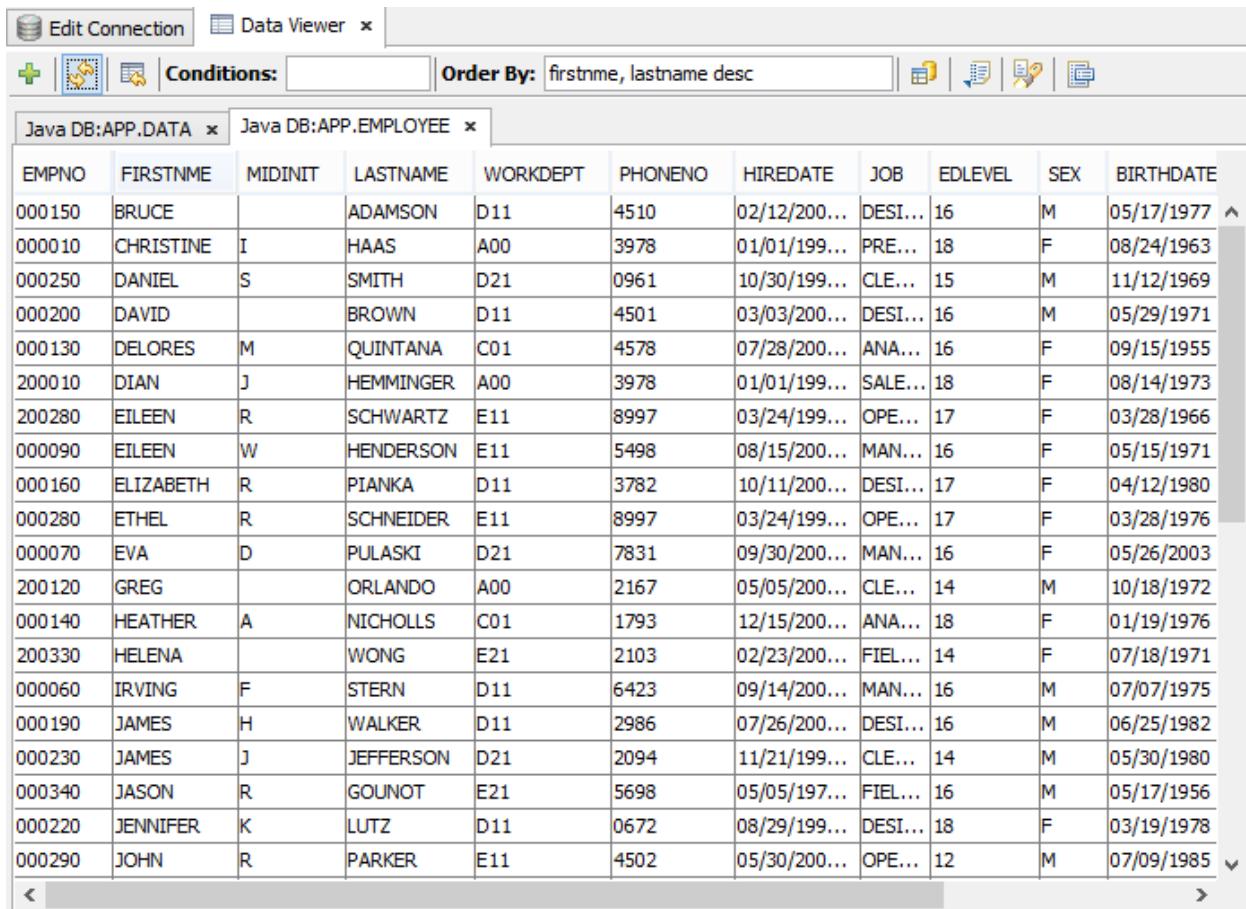
The data from the APP.EMPLOYEE table is as follows:

EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE
000020	MICHAEL	L	THOMPSON	B01	3476	10/10/200...	MAN...	18	M	02/02/1978 ...
000030	SALLY	A	KWAN	C01	4738	04/05/200...	MAN...	20	F	05/11/1971 ...
000050	JOHN	B	GEYER	E01	6789	08/17/197...	MAN...	16	M	09/15/1955 ...
000060	IRVING	F	STERN	D11	6423	09/14/200...	MAN...	16	M	07/07/1975 ...
000070	EVA	D	PULASKI	D21	7831	09/30/200...	MAN...	16	F	05/26/2003 ...
000090	EILEEN	W	HENDERSON	E11	5498	08/15/200...	MAN...	16	F	05/15/1971 ...
000100	THEODORE	Q	SPENSER	E21	0972	06/19/200...	MAN...	14	M	12/18/1980 ...
000120	SEAN		O'CONNELL	A00	2167	12/05/199...	CLE...	14	M	10/18/1972 ...
000230	JAMES	J	JEFFERSON	D21	2094	11/21/199...	CLE...	14	M	05/30/1980 ...
000240	SALVATORE	M	MARINO	D21	3780	12/05/200...	CLE...	17	M	03/31/2002 ...
000250	DANIEL	S	SMITH	D21	0961	10/30/199...	CLE...	15	M	11/12/1969 ...
000260	SYBIL	P	JOHNSON	D21	8953	09/11/200...	CLE...	16	F	10/05/1976 ...
000270	MARIA	L	PEREZ	D21	9001	09/30/200...	CLE...	15	F	05/26/2003 ...
200120	GREG		ORLANDO	A00	2167	05/05/200...	CLE...	14	M	10/18/1972 ...
200240	ROBERT	M	MONTEVERDE	D21	3780	12/05/200...	CLE...	17	M	03/31/1984 ...

Figure 29: Conditions

### Apply sort criteria

To apply a sort criteria to the data set enter it in the **Order By** field and press "refresh"  button. Sort criteria must be a valid SQL "order by". Filed numbers are not allowed in the file mode.



The screenshot shows the Data Viewer interface with the following details:

- Toolbar:** Includes "Edit Connection", "Data Viewer", "Conditions" (with a dropdown menu), "Order By" (set to "firstname, lastname desc"), and various toolbar icons for filtering, sorting, and refresh.
- Tables:** Two tables are listed: "Java DB:APP.DATA" and "Java DB:APP.EMPLOYEE".
- Data View:** The "Java DB:APP.EMPLOYEE" table is displayed as a grid with the following columns: EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB, EDLEVEL, SEX, and BIRTHDATE.
- Sorting:** The data is sorted by "firstname, lastname desc". The "BIRTHDATE" column shows a downward arrow icon, indicating it is sorted in descending order.
- Content:** The grid contains 30 rows of employee data, such as Bruce Adamson (EMPNO 000150) and John Parker (EMPNO 000290).

Figure 30: Order by

### Interrupt populating data set

When Data Viewer is populating data set (for example by executing SQL query or reading Excel worksheet) the progress dialog window is displayed. You cannot do anything in Data Explorer while SQL is running but you can interrupt (cancel) it at any time by pressing on "Cancel" button.

**Note:** In the Web mode the Data Viewer progress dialog is disabled by default. You can enable it using the following access path: Settings->Data Viewer App-> Show Progress.

### View records of the data set in the Form. View CLOB and BLOB fields

In the Form view one record of the data set can be displayed at the time but it is an only view which can display CLOB and BLOB fields.

To display a record of the data set in the form view click on  button in the Data Viewer toolbar.

See [Data Set Record Viewer](#)

## ***Describe Data Set***

It is possible to display detailed information about data set fields, such as name, type, etc.

To describe a data set click on  button in the Data Viewer toolbar.

See [Describe Data Set](#)

## ***Search in the Data Set***

To search in the data set click on  button in the Data Viewer toolbar.

This is a full text search; text value of the field is compared to the search string.

See [Search in Data Set](#)

## ***Copy data in the grid to the clipboard***

In the desktop modes (Client and Client-Server) you can copy selected cells to the clipboard.

See [Grid](#)

## ***Calculate Function***

It is possible to calculate statistical functions such as count(), min(), max() etc for the entire data set or selected rows only. The wide range of functions is available.

To calculate a function click on  button in the Data Viewer toolbar.

See [Calculate Function](#)

## ***Display Data as a Chart***

It is possible to display a chart for the entire data set or for the selected rows only. There is a wide selection of chart types and customization options, including multi-series and 3D charts.

To display data as a chart click on  button in the Data Viewer toolbar.

See [Display Data as a Chart](#)

## ***Export Data Set to the different file formats***

Data set or selected rows of the data set can be exported to the multiple file formats such as text, XML, Excel, etc.

To export data to the various file formats click on  button in the Data Viewer toolbar.

See [Export Data Set](#)

### ***Transform Data Set***

There is a graphical UI for transforming data sets using a wide range of transformation algorithms such as: pivot, de-normalization, filtering, sorting, remove duplicates, set operations, etc.

To transform data set click on  button in the Data Viewer toolbar.

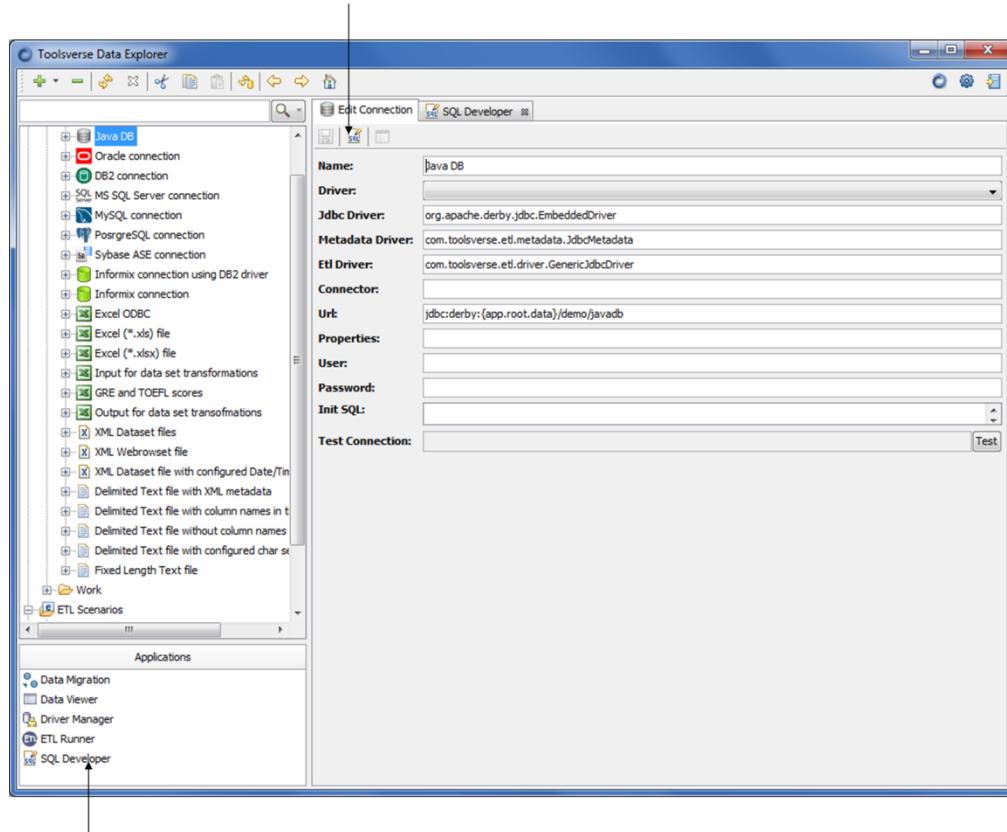
See [Transform Data Set](#)

## SQL Developer (SQL Editor)

SQL Developer is a Data Explorer application which is used to edit, format and execute SQL statements and scripts. Multiple editors may be open at the same time, each controlling its own SQL log and data sets. Data sets can be displayed in the grid and form views and as charts.

To open SQL Developer click on SQL Developer  icon in the Applications list or in the Connection node editor panel.

Click on this icon to open SQL Developer for this database connection



Click on this icon to open SQL Developer for the currently selected database connection. If none is selected — just open SQL Developer

**Figure 31: Open SQL Developer**

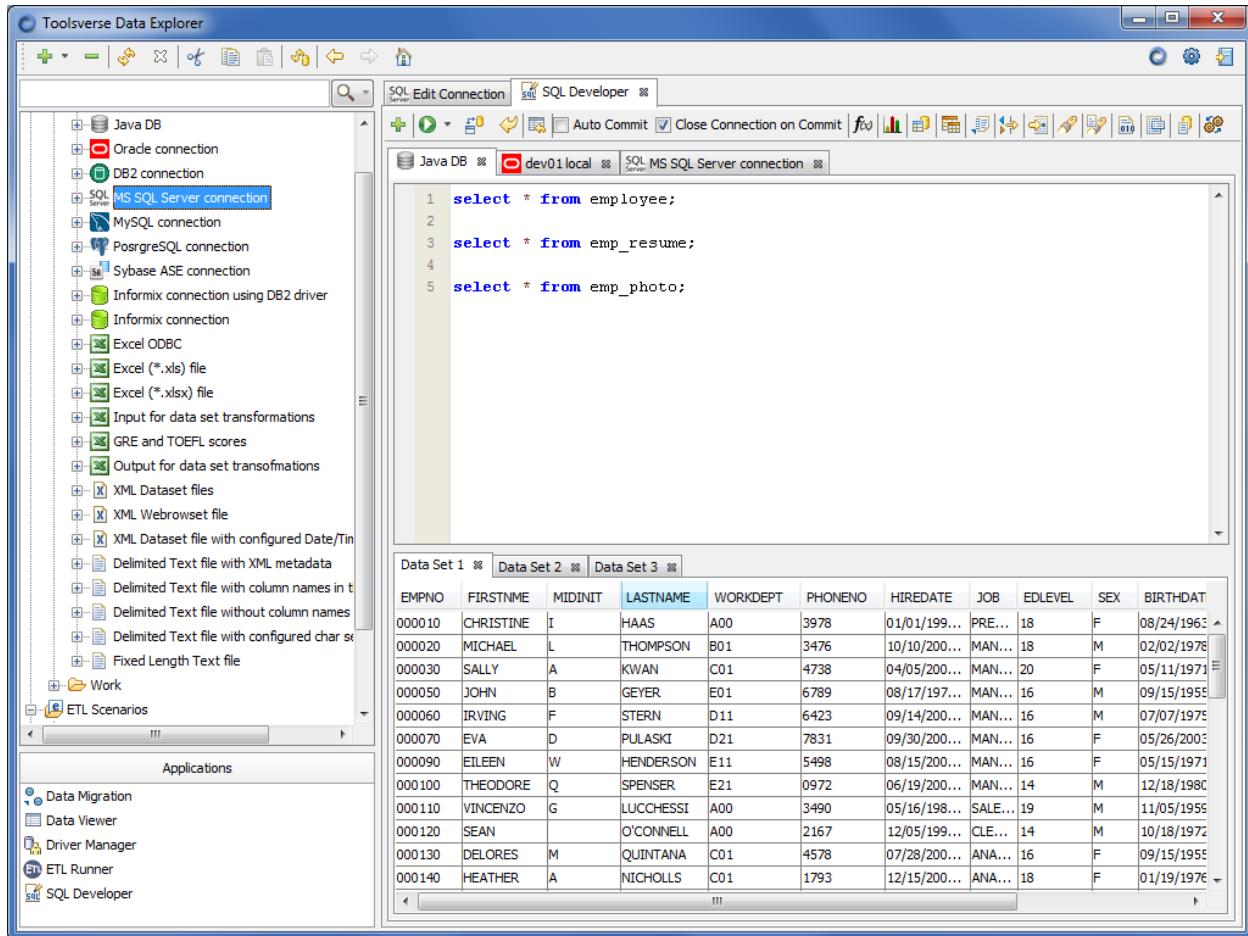


Figure 32: SQL Developer with multiple opened tabs

## Features

Please see a full list of SQL Developer features [here](#).

## Commands

Icon	Function	Windows and others	OS X	Web browser
	Opens a dialog window with a tree of available connections. After connection is selected adds SQL editor tab to the SQL Developer for the chosen connection	Ctrl+N	Command +N	Shift+F4
	Opens menu which includes the following commands:			
	Execute SQL - executes one or multiple SQL statements divided by semicolon	Ctrl+F2	Ctrl+F2	Ctrl+F2
	Execute SQL script, for example anonymous SQL block	Ctrl+F3	Ctrl+F3	Ctrl+F3

	Execute External Tool - executes SQL statement in the external tool, for example sql*plus if the ETL driver for the current connection supports external tools. Otherwise this menu item is not visible	Alt+F2	Alt+F2	Alt+F2
	Show Execution Plan - displays an execution (explain) plan for the SQL statement in the editor. If ETL driver for the current connection does not support execution plan this menu item is not visible	Alt+F3	Alt+F3	Alt+F3
	Commits database transaction. It is disabled if there is nothing to commit			
	Rollback database transaction. It is disabled if there is nothing to rollback			
	<a href="#">Set Grid Defaults such as maximum number of rows in the data set, etc</a>	Ctrl+F9	Command +F9	Ctrl+F9
	Enable\disable auto commit			
	Enable\disable close connection on commit			
	<a href="#">Describe Data Set</a>	Shift+F12	Shift+F12	Shift+F12
	Describe Database Object (table/view/synonym). The object name must be either selected in the SQL editor or cursor must be somewhere within object name (not supported in the Web mode)	Shift+F11	Shift+F11	Shift+F11
	<a href="#">Calculate function</a>	Alt+F12	Alt+F12	Alt+F12
	<a href="#">Display Data as a Chart</a>	Shift+F10	Shift+F10	Shift+F10
	<a href="#">Format SQL</a>	Shift+F9	Shift+F9	Shift+F9
	<a href="#">Go To the Line</a>	Ctrl+L	Command +L	Ctrl+F8
	<a href="#">Search and Replace</a>	Ctrl+F	Command +F	Ctrl +F7
	<a href="#">Search in the Data Set</a>	F7	F7	F7

	<a href="#">Show Code Snippets for the particular database</a>	Ctrl+F12	Command +F12	Ctrl+F12
	<a href="#">Show Data Set Record in the Form view</a>	F4	F4	F4
	Show SQL History (history of executed SQL statements and scripts)	Ctrl+F10	Command +F10	Ctrl+F10
	<a href="#">Transform Data Set</a>	Shift+F8	Shift+F8	Shift+F8

## SQL Developer How To

### Execute SQL statement(s).

Click on “execute sql” button and select “Execute SQL” menu item from the menu. All SQL statements in the current tab of the editor will be executed with output to the individual tabs. Multiple statements must be separated by semicolon character.

You can change a shortcut used to execute SQL statements using the following access path:  
Settings->SQL Developer App->Execute SQL.

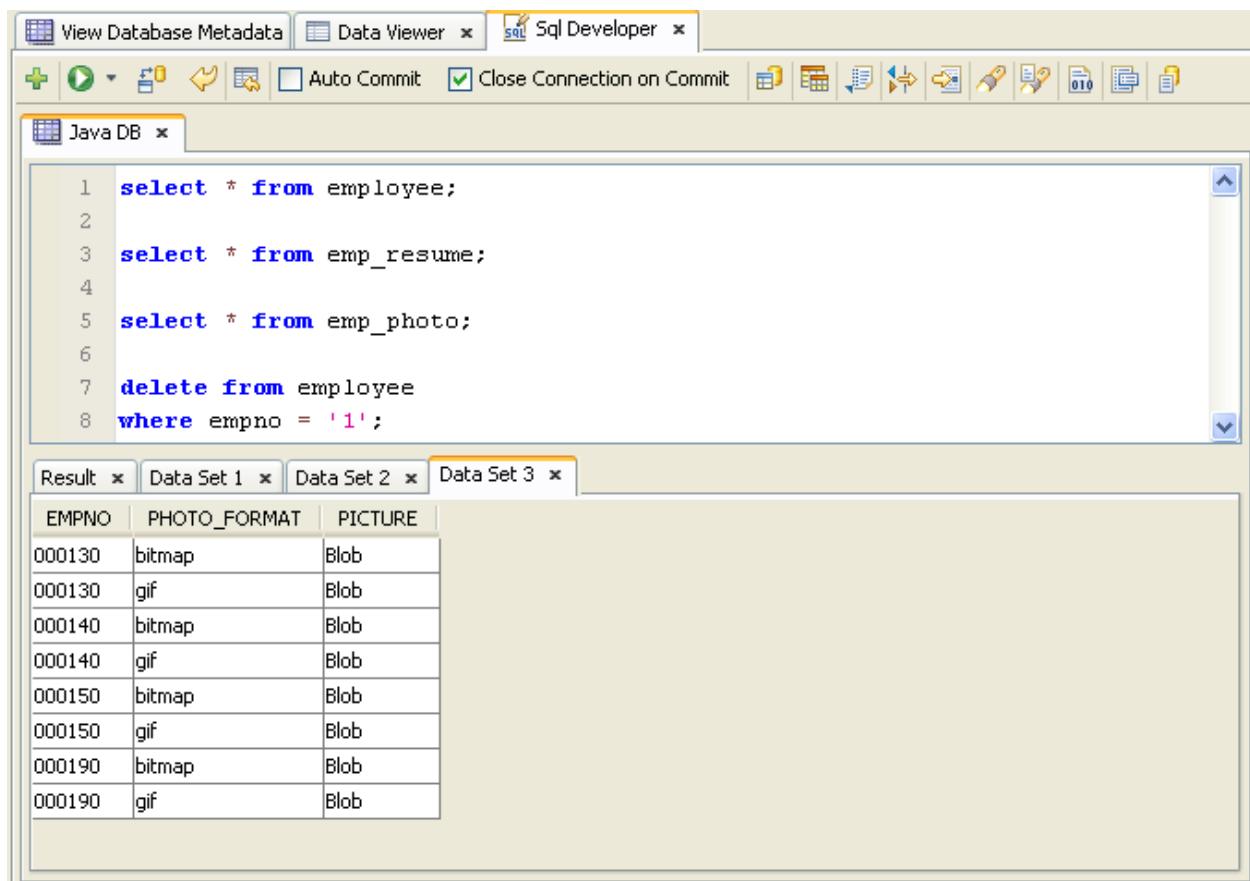


Figure 33: Execute SQL statements

If there is a selected text in the editor it will be executed instead of entire editor's buffer.

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes 'Edit Connection' and the current tab 'Sql Developer'. Below the menu is a toolbar with various icons. The main workspace has two tabs: 'Oracle connection' and 'Java DB'. The 'Java DB' tab is active, displaying a SQL script in the editor:

```
select * from employee;
select * from emp_photo;
select * from emp_resume;
delete from employee
where empno = '1';
```

A portion of the script is highlighted in blue, indicating it is selected. Below the editor is a table titled 'Data Set 1' showing the results of the executed query:

EMPNO	RESUME_FORMAT	RESUME
000130	ascii	Resum...
000130	html	<!DOCT...
000140	ascii	Resum...
000150	ascii	Resum...
000140	html	<!DOCT...
000150	html	<!DOCT...
000190	ascii	Resum...
000190	html	<!DOCT...

Figure 34: Execute selected SQL

### Execute SQL script

Click on “execute sql” button and select “Execute SQL Script” menu item from the menu.

You can change a shortcut used to execute SQL script using the following access path:

Settings->SQL Developer App->Execute SQL Script.

**Note:** If ETL driver associated with the connection does not support SQL scripts this menu item is not visible.

### Execute SQL in the External Tool

Click on “execute sql” button and select “Execute External Tool” menu item from the popup menu.

You can change a shortcut used to execute SQL in the external tool using the following access path:

Settings->SQL Developer App->Execute External Tool.

**Note:** If ETL driver associated with the connection does not support external tool this menu item is not visible. The generic JDBC ETL driver does not support it.

**Note:** The external tool for the particular database must be installed on the computer which executes SQL statement.

The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for 'Edit Connection' and 'Sql Developer'. Below the tabs are various icons for connection management and toolbars. A toolbar below the top bar includes 'Auto Commit' and 'Close Connection on Commit' checkboxes. The main window has a title bar 'Oracle connection x'. In the top-left of the main area, there is a code editor window containing the SQL command 'desc emp\_photo'. To the right of the code editor is a results grid titled 'Result x'. The results grid displays the following data:

	Name	Null?	Null?
1	EMPNO	NOT NULL	CHAR (6)
2	PHOTO_FORMAT	NOT NULL	VARCHAR2 (10)
3	PICTURE		BLOB
4			
5			
6			

Figure 35: Execute SQL in the External Tool

#### *Show execution (explain) plan for the query*

Click on “execute sql” button and select “Show Execution Plan” menu item from the menu.

You can change a shortcut used to display an execution plan using the following access path:  
Settings->SQL Developer App->Show Execution Plan.

**Note:** If ETL driver associated with the connection does not support displaying of the execution plan this menu item is not visible. The generic JDBC ETL driver does not support it.

**Note:** In some cases (for example IBM DB2) this function requires client for the particular database to be installed on the same computer which executes SQL statement.

The screenshot shows the Oracle SQL Developer interface. In the top left, there's a connection list with 'Edit Connection' and 'Sql Developer'. Below it is a toolbar with various icons. A connection named 'Oracle connection' is selected. In the main area, there's a code editor window containing the following SQL:

```

1 select * from employee
2 where empno = :empno
3

```

Below the code editor is an 'Execution Plan' window. It starts with a header:

```

1 Plan hash value: 2119105728
2
3
4 | Id | Operation          | Name      | Rows   | Bytes | Cost (%CPU) | Time
5 |
6 | 0 | SELECT STATEMENT |          | 1 | 61 | 12 (0) | 00:00:01
7 |* 1 | TABLE ACCESS FULL| EMPLOYEE | 1 | 61 | 12 (0) | 00:00:01
8
9

```

Following the execution plan is a section titled 'Predicate Information (identified by operation id):' which contains:

```

10 1 - filter ("EMPNO"=:EMPNO)
11
12
13
14

```

Figure 36: Execution (explain) plan

### Run SQL query against Excel worksheet when Excel ODBC connection is used

To execute SQL query against Excel spreadsheet, use [sheetname\$] for the sheet name. Example:

The screenshot shows the Toolsverse Data Explorer interface. On the left, there's a tree view of connections under 'Demo'. One connection is highlighted: 'Excel ODBC'. The main workspace has a connection list with 'Edit Connection' and 'Sql Developer'. Below it is a toolbar. A connection named 'Excel ODBC' is selected. In the main area, there's a code editor window containing the following SQL:

```

1 select * from [employee$]
2 order by 4

```

Below the code editor is a 'Data Set 1' window showing the results of the query. The results are as follows:

EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONE NO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SA
000150	BRUCE		ADAMSON	D11	4510	02/12/200...	DESI...	16	M	05/17/1977 ...	5528
200340	ROY	R	ALONZO	E21	5698	07/05/199...	FIEL...	16	M	05/17/1956 ...	3184
000200	DAVID		BROWN	D11	4501	03/03/200...	DESI...	16	M	05/29/1971 ...	5774
000050	JOHN	B	GEYER	E01	6789	08/17/197...	MAN...	16	M	09/15/1955 ...	8011
000340	JASON	R	GOONOT	E21	5698	05/05/197...	FIEL...	16	M	05/17/1956 ...	4386
000010	CHRISTINE	I	HAAS	A00	3978	01/01/199...	PRE...	18	F	08/24/1963 ...	1521
200010	DIAN	J	HEMMINGER	A00	3978	01/01/199...	SALE...	18	F	08/14/1973 ...	4650
000090	EILEEN	W	HENDERSON	E11	5498	08/15/200...	MAN...	16	F	05/15/1971 ...	8975
000230	JAMES	J	JEFFERSON	D21	2094	11/21/199...	CLE...	14	M	05/30/1980 ...	4218
200220	REBA	K	JOHN	D11	0672	08/29/200...	DESI...	18	F	03/19/1978 ...	6984
000260	SYBIL	P	JOHNSON	D21	8953	09/11/200...	CLE...	16	F	10/05/1976 ...	4725
000210	WILLIAM	T	JONES	D11	0942	04/11/199...	DESI...	17	M	02/23/2003 ...	6821
000030	SALLY	A	KWAN	C01	4738	04/05/200...	MAN...	20	F	05/11/1971 ...	9825

Figure 37: SQL query against Excel worksheet

### ***Interrupt SQL execution***

When SQL is running the progress dialog window is displayed. You cannot do anything in Data Explorer while SQL is running but you can interrupt (cancel) it at any time by pressing on “Cancel” button.

**Note:** In the Web mode the SQL progress dialog is disabled by default. You can enable it using the following access path: Settings->SQL Developer App-> Show Progress

### ***Open new SQL editor tab for the connection in the SQL Developer***

Click on “select connection to add tab”  button in the SQL Developer toolbar to add a new tab to the SQL Developer.

You can also select a connection node in the nodes browser and click on the “open sql developer”  button. If SQL Developer already has a tab for this connection the tab will be selected.

You can have as many open tabs as you need (even for the same logical connection) and each will have its own physical connection so you can commit/rollback transactions separately in each tab. For example if you have an Oracle connection named “dev” you can open multiple tabs for “dev”, they will be named dev1, dev2, etc and will **not** share the same physical connection.

You can close tab at any time by clicking on the small “x” button on the top of the tab. The associated physical connection will be automatically closed as well as open log and data sets.

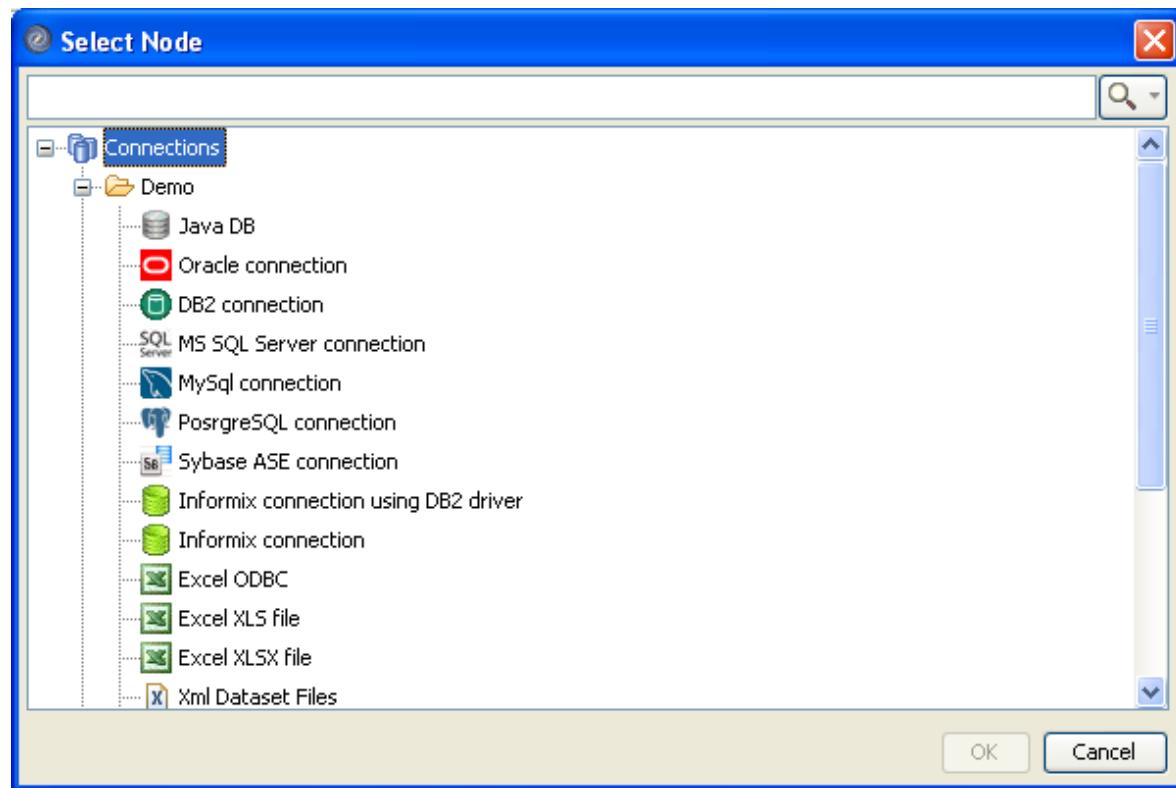


Figure 38: Select connection to add tab

### ***View records of the data set in the Form. View CLOB and BLOB fields***

In the Form view one record of the data set can be displayed at the time but it is an only view which can display CLOB and BLOB fields.

To see a data set record in the form view click on  button in the SQL Developer toolbar.

See [Data Set Record Viewer](#)

### ***Use Bind parameters (variables)***

To execute SQL statement or script with the bind variables use token **:param** for the variable. Using bind parameters (variables) is a way to make your query more generic. For example, instead of hard-coded conditions for the birthday (`birthday = '01/02/2002'`) and sex (`sex = 'F'`) you can use `birthday = :birthdate` and `sex = :sex`. The Data Explorer will ask you to input parameters in the special dialog window before executing query. It will remember parameters you entered last time.

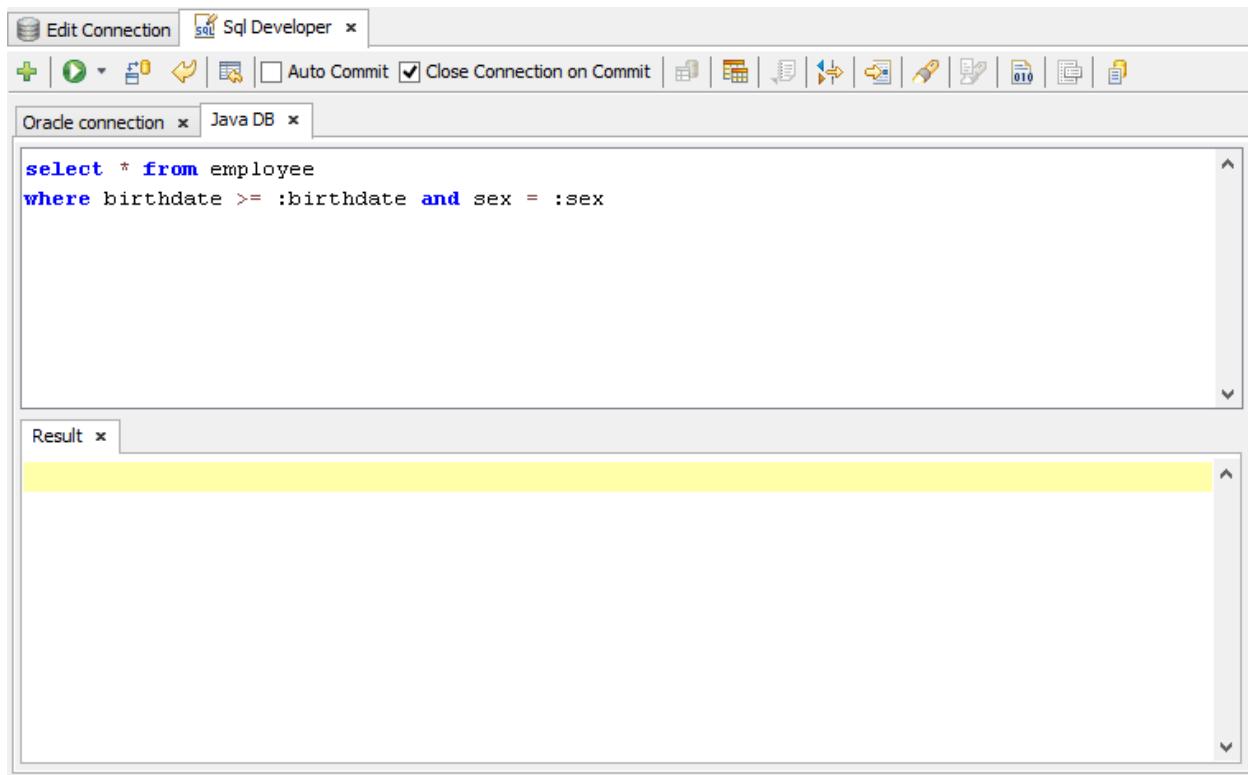


Figure 39: Bind variables

### ***Use Typed Bind parameters (variables)***

If you need to specify a type of the bind variable use token: **type\_param** for the variable.

The list of the supported types (case insensitive):

Type	Example
INTEGER	:integer_age
NUMBER	:number_latitude
STRING	:string_last_name
DATE	:date_date_of_birth
TIME	:time_departure_time

TIMESTAMP	:timestamp_event
BOOLEAN	:boolean_is_included

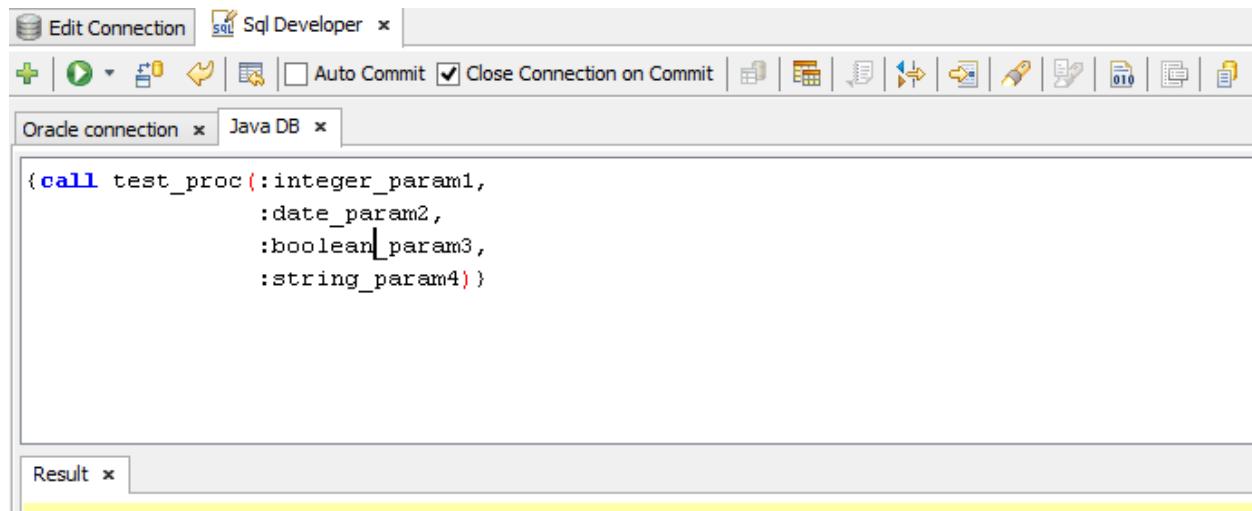


Figure 40: Typed bind variables

When SQL is executed with the typed bind variables the Data Explorer will open Parameters dialog window. The actual input fields for the parameters will not allow you to enter invalid values. For example, you will not be able to enter a string if parameter was defined as integer.

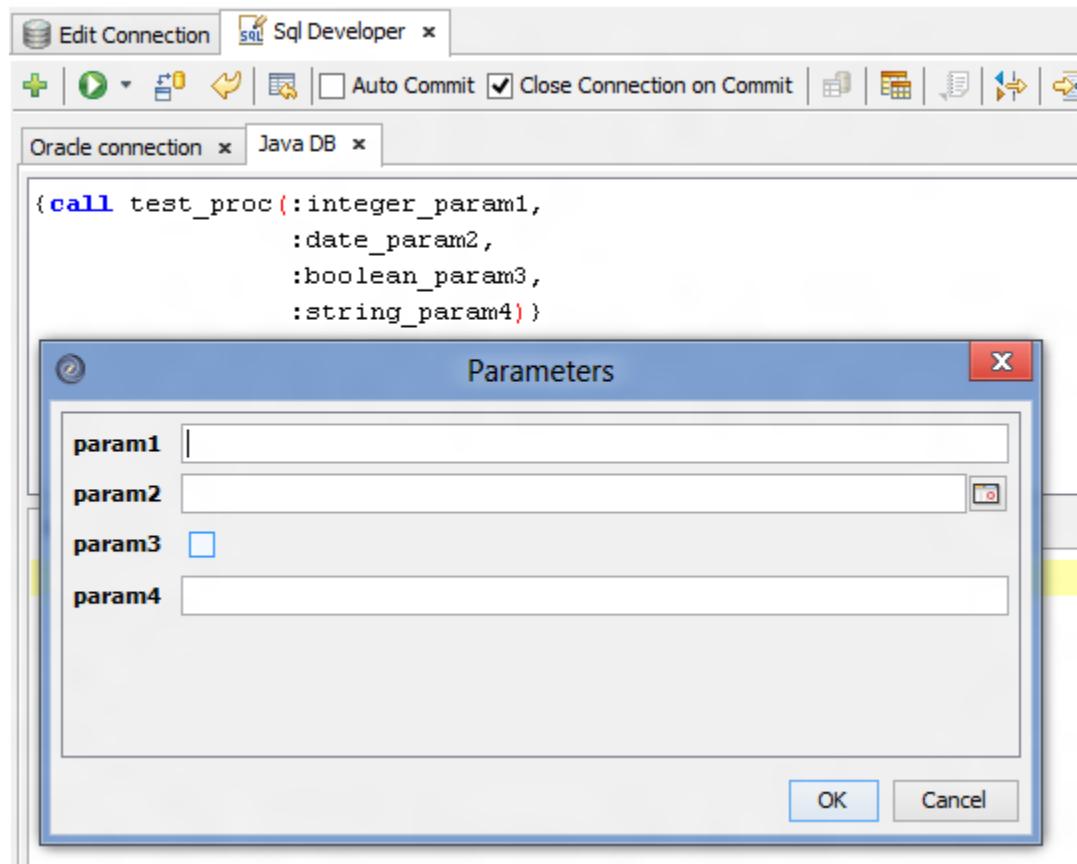


Figure 41: SQL parameters

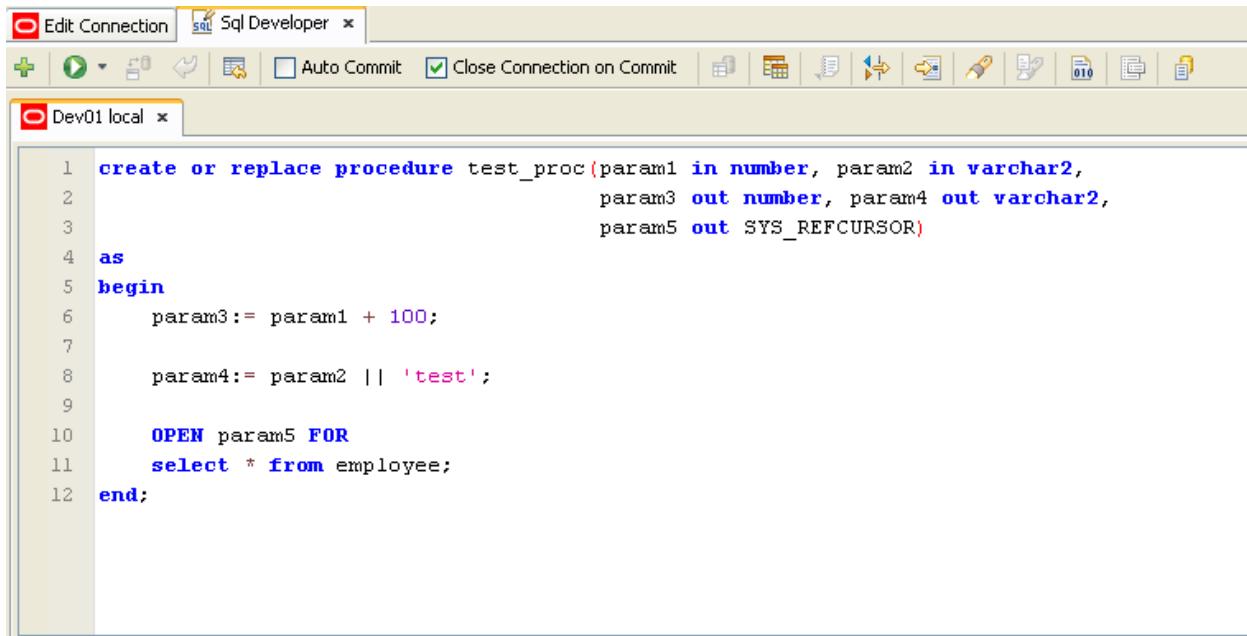
### *Use Output parameters*

If you want to see an output from the stored procedure (output parameters) use token:

**out\_type\_param** for the output parameter. You must specify a parameter type. The list of the supported types (case insensitive):

Type	Example
BIT	:out_bit_value
TINYINT	:out_tinyint_abc
SMALLINT	:out_smallint_age
INTEGER (or INT)	:out_integer_value, :out_int_value
BIGINT	:out_bigint_really_big_int
FLOAT	:out_float_float_value
REAL	:out_real_some_real_value
DOUBLE	:out_double_latitude
NUMERIC (or NUMBER)	:out_numeric_xyz, :out_number_acb
DECIMAL	:out_decimal_mnn
CHAR	:out_char_yes_no
VARCHAR (or STRING)	:out_varchar_last_name or :out_string_last_name
LONGVARCHAR	:out_longvarchar_some_long_string
DATE	:out_date:dob
TIME	:out_date_event_time
TIMESTAMP	:out_timestamp_happened
BOOLEAN	:out_boolean_yes_no
NCHAR	:out_nchar_yes_no
NVARCHAR	:out_nvchar_something
LONGNVARCHAR	:out_longnvchar_test
CURSOR	:out_cursor_employees
Actual number used by the particular JDBC driver	:out_15_param, :out_minus10_param2

For example, there an Oracle stored procedure with input and output parameters:



The screenshot shows the SQL Developer interface with a connection named "Dev01 local". In the central workspace, a script is displayed with the following PL/SQL code:

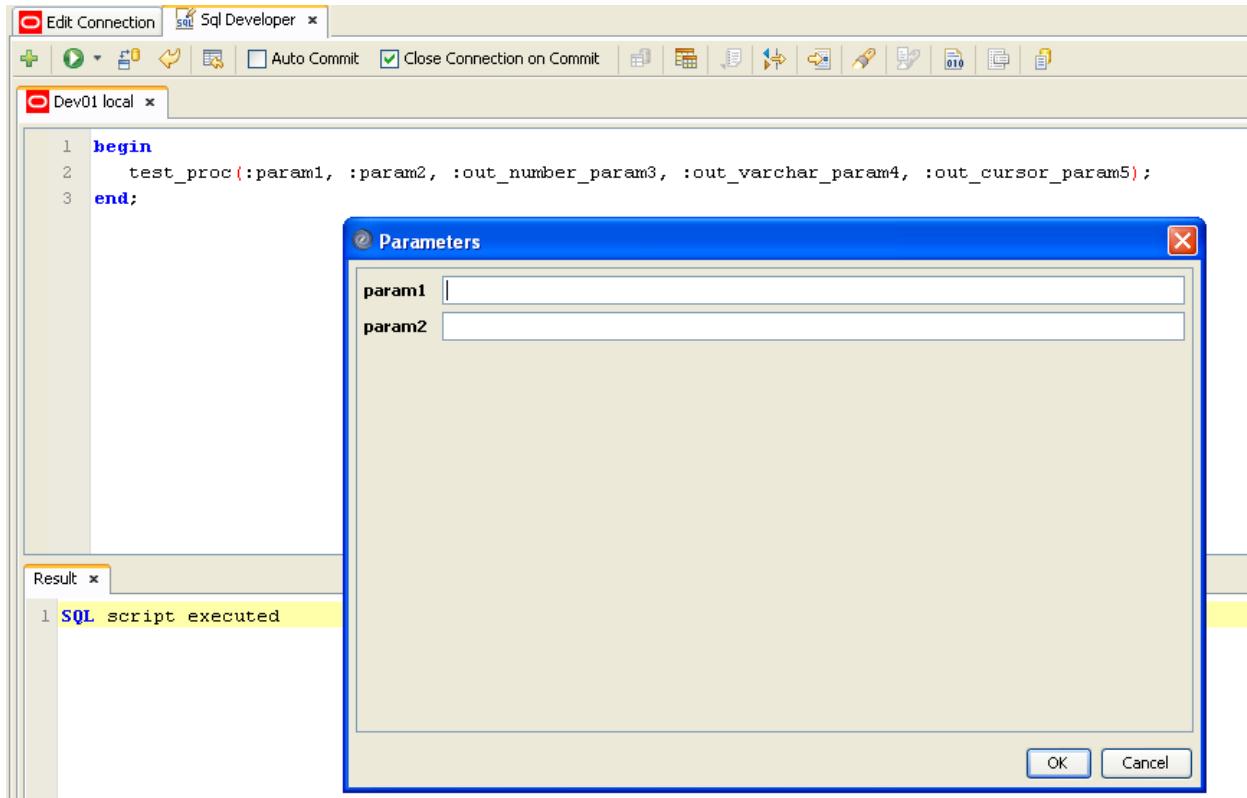
```

1 create or replace procedure test_proc(param1 in number, param2 in varchar2,
2                                     param3 out number, param4 out varchar2,
3                                     param5 out sys_refcursor)
4 as
5 begin
6     param3:= param1 + 100;
7
8     param4:= param2 || 'test';
9
10    OPEN param5 FOR
11        select * from employee;
12 end;

```

Figure 42: Stored procedure with input and output parameters

If you want to execute it from the Data Explorer you can use the code below. The Data Explorer will ask you to enter input parameters:



The screenshot shows the SQL Developer interface with a connection named "Dev01 local". In the central workspace, a script is displayed with the following PL/SQL code:

```

1 begin
2     test_proc(:param1, :param2, :out_number_param3, :out_varchar_param4, :out_cursor_param5);
3 end;

```

A "Parameters" dialog box is overlaid on the interface, prompting for values for "param1" and "param2". The "Result" tab at the bottom shows the message "1 SQL script executed".

Figure 43: Executing stored procedure

After stored procedure is executed the output will look like this:

The screenshot shows two instances of the SQL Developer interface. The top instance is connected to 'Dev01 local' and displays the following PL/SQL code in the script editor:

```

1 begin
2   test_proc (:param1, :param2, :out_number_param3, :out_varchar_param4, :out_cursor_param5);
3 end;

```

The 'Result' tab below the code shows the output:

```

1 SQL script executed
2 Output variable param3 = 115
3 Output variable param4 = abctest

```

An arrow points from the text 'Output parameters' to the third line of the result.

The bottom instance is also connected to 'Dev01 local' and shows the same PL/SQL code. Its 'Result' tab displays a cursor output parameter as a table:

EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONE NO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
000010	CHRISTINE	I	HAAS	A00	3978	01/01/199...	PRE...	18	F	08/24/1963 ...	152750	1000	4220
000020	MICHAEL	L	THOMPSON	B01	3476	10/10/200...	MAN...	18	M	02/02/1978 ...	94250	800	3300
000030	SALLY	A	KWAN	C01	4738	04/05/200...	MAN...	20	F	05/11/1971 ...	98250	800	3060
000050	JOHN	B	GEYER	E01	6789	08/17/197...	MAN...	16	M	09/15/1955 ...	80175	800	3214

An arrow points from the text 'Cursor output parameter' to the second row of the table.

Figure 44: Output parameters

### Use SQL History

All executed SQL statements and scripts are stored in the history, along with entered parameters. The history is persistent which means it is still there when you restart Data Explorer. Use history button in the SQL Developer toolbar to access SQL history. You can search by any part of the SQL. Search is case insensitive.

You can change the size of the history using the following access path:

Settings->SQL Developer App->SQL History Size. You can change a keyboard shortcut used to open SQL History using the following access path Settings->SQL History Plugin->Show SQL History.

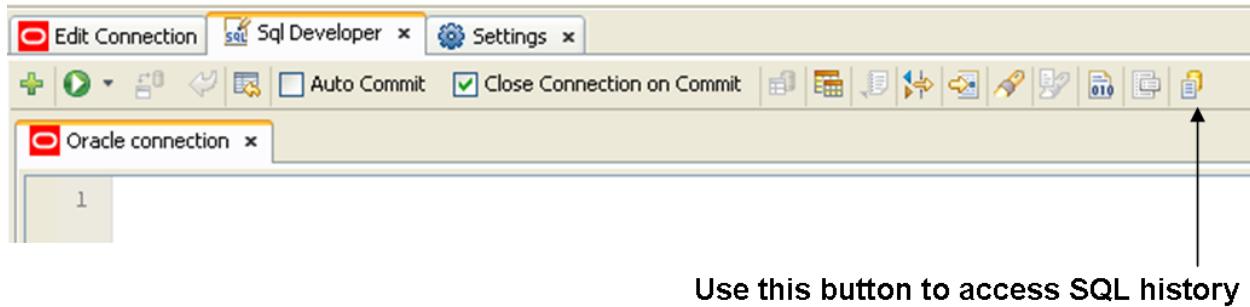


Figure 45: Access SQL history

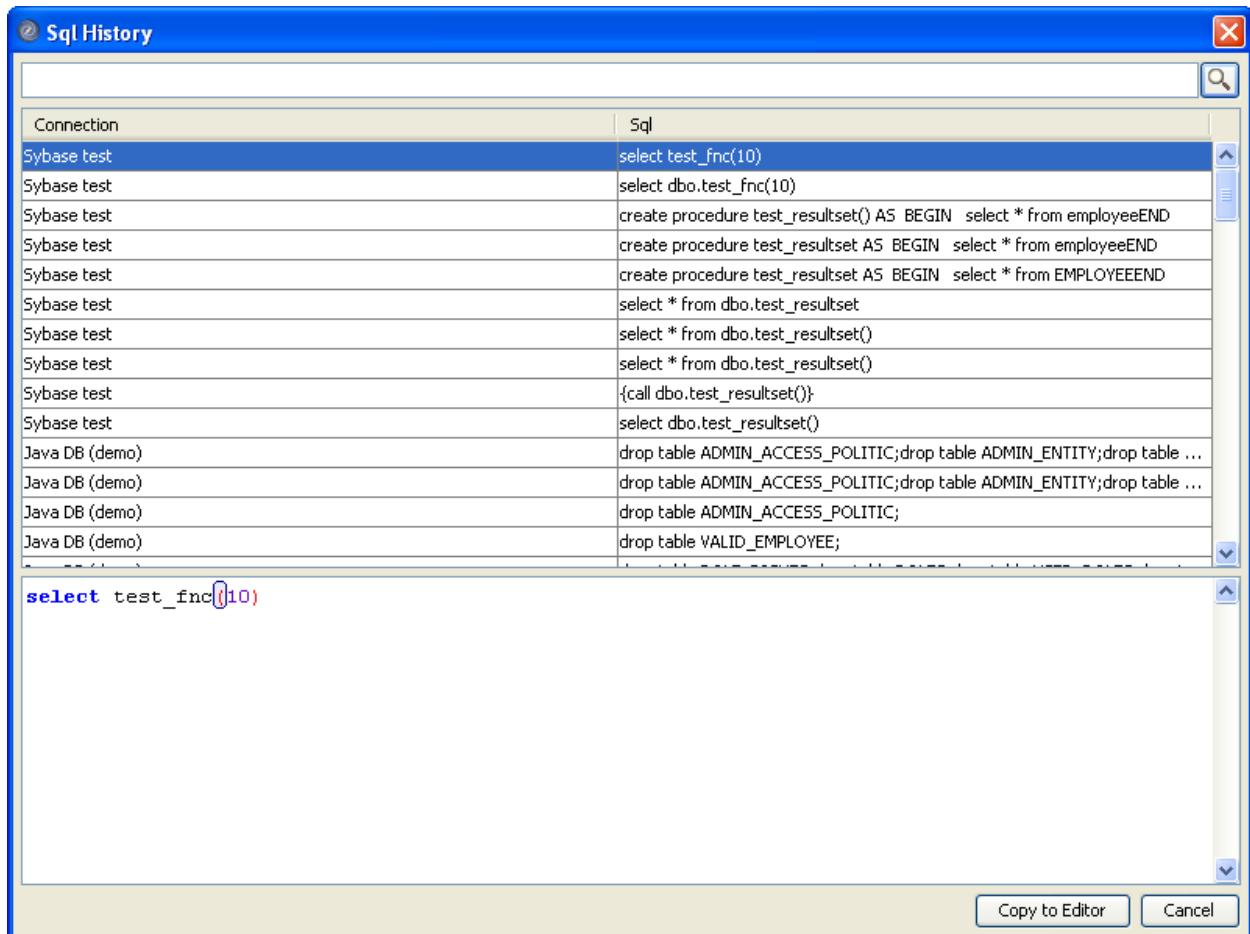


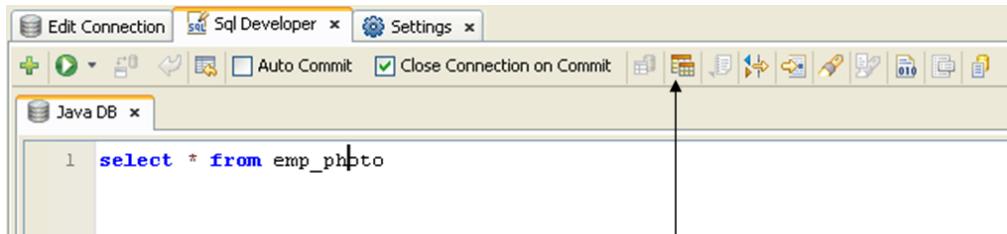
Figure 46: SQL History

Click on “Copy to Editor” to copy item from the SQL history list to the current SQL Editor window. The entire editor’s buffer gets replaced with the SQL from the history. If you want to copy just part of the SQL or insert SQL at the cursor position in the editor use **copy and paste**.

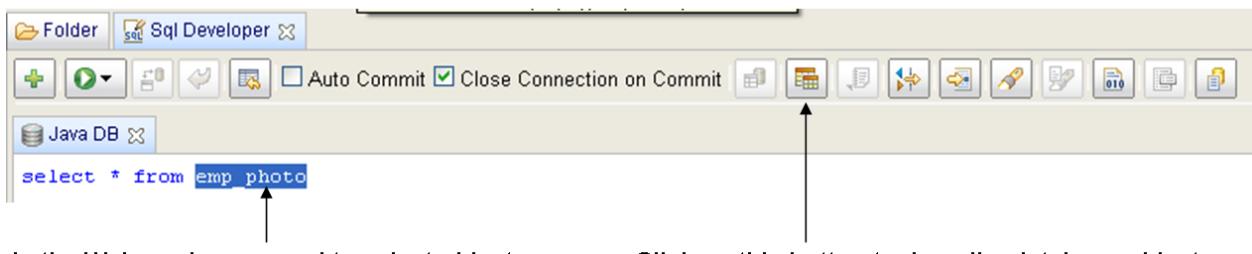
#### ***Describe database object at the cursor position***

If you want to see a table fields (table/view/synonym and other table types) without digging into database object browser just place a cursor in the SQL editor on the object name (client and client-server modes only) or select an object name in the SQL editor and click on the describe object  button

in the SQL Developer toolbar. You can change a keyboard shortcut used to describe database object using the following access path: Settings->Describe Database Object Plugin-> Describe Database Object.



Click on this button to describe database object



In the Web mode you need to select object

Click on this button to describe database object

Figure 47: Describe database object

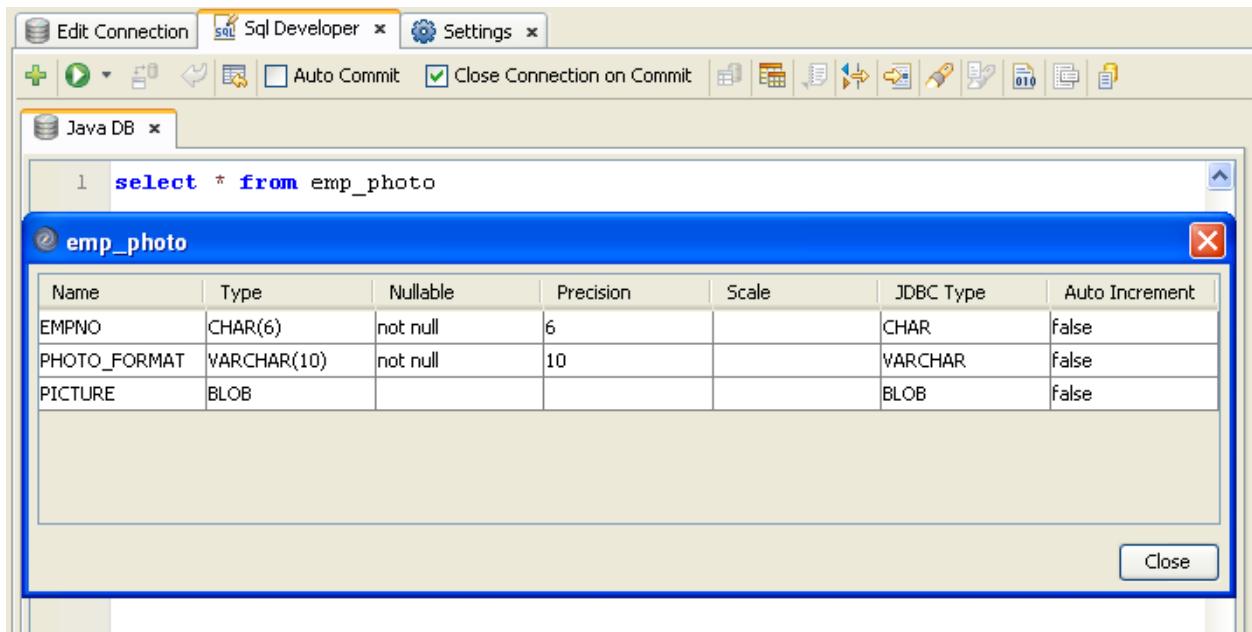


Figure 48: Database object fields

You can select cells and copy them to the clipboard using Ctrl+C (or Command+C on Mac).

### ***Describe Data Set***

It is possible to display detailed information about data set fields, such as name, type, etc. To describe data set click on  button in the SQL Developer toolbar.

See [Describe Data Set](#)

### ***Change grid parameters***

To change grid parameters click on  button in the SQL Developer toolbar.

See [Set Grid Defaults](#).

### ***Manage transactions***

SQL Developer allows automatic and manual transaction management. If check box “Auto Commit” is unchecked you can manually commit  or rollback  transactions. If “Close Connection on Commit” check box is checked the database connection is getting closed each time you commit or rollback transaction. The default values for both parameters can be changed using the following access path: Settings->SQL Developer App->Default Auto Commit On/Default Close Connection on Commit and Rollback.

### ***Search in the Data Set***

To search in the data set click on  button in the SQL Developer toolbar.

See [Search in Data Set](#)

### ***Copy data in the grid to the clipboard***

In the desktop modes (Client and Client-Server) you can copy selected cells to the clipboard.

See [Grid](#)

### ***Calculate Function***

It is possible to calculate statistical functions such as count(), min(), max() etc for the entire data set or selected rows only. The wide range of functions is available.

To calculate a function click on  button in the SQL Developer toolbar.

See [Calculate Function](#)

### ***Display Data as a Chart***

It is possible to display a chart for the entire data set or for the selected rows only. There is a wide selection of chart types and customization options, including multi-series and 3D charts.

To display data as a chart click on  button in the SQL Developer toolbar.

See [Display Data as a Chart](#)

### ***Export Data Set to the different file formats***

Data set or selected rows of the data set can be exported to the multiple file formats such as text, XML, Excel, etc.

To export data to the various file formats click on the  button in the SQL Developer toolbar.

See [Export Data Set](#)

### ***Transform Data Set***

There is a graphical UI for transforming data sets using a wide range of transformation algorithms such as: pivot, de-normalization, filtering, sorting, remove duplicates, set operations, etc.

To transform data set click on  button in the SQL Developer toolbar.

See [Transform Data Set](#)

### ***Format SQL code***

To format an SQL click on the  button in the SQL Developer toolbar.

See [Code Formatter](#)

### ***Search and Replace text in the SQL editor***

To search and replace text click on the  button in the SQL Developer toolbar.

See [Search and Replace](#)

### ***Go to the Line in the SQL editor***

To jump to line in the editor click on the  button in the SQL Developer toolbar.

See [Go to the Line](#)

## ETL Integrated Development Environment

ETL Integrated Development Environment is a Data Explorer application that provides comprehensive facilities for ETL development. You need it if you want to create and run ETL scenarios directly from the Data Explorer. ETL IDE incorporates free and open-source [ETL Framework](#).

## Glossary

Term	Definition
Data integration	A process which involves combining data residing in different sources and providing users with a unified view of these data
Data Migration	A process of transferring data between storage types, formats, or computer systems
ETL	Extract Transform Load. ETL is a process which involves: <ul style="list-style-type: none"> <li>• Extracting data from outside sources</li> <li>• Transforming it to fit operational needs (which can include quality levels)</li> <li>• Loading it into the end target database or data source</li> </ul>
ETL Scenario	A program in the declarative XML-based language which describes extract, transform and load steps of the ETL process
Inner ETL Scenario	ETL scenario included in other ETL scenario
ETL Framework	The set of classes and interfaces coded in Java which implement feature rich ETL engine. Includes multiple Toolsverse and third-party jar files
ETL Engine	Same as ETL Framework
Standalone ETL tool	A standalone program which executes one or multiple ETL Scenarios
Embedding	A way to integrate ETL framework into customer's application using open API (application programming interface)
Source	The data set to extract. Can be populated by executing SQL query or reading file-based sources such as Excel worksheet, text and XML files, etc
Destination	The load target. Can be a database table or file-based data set such as Excel worksheet, text and XML files, etc
Data Set	The in-memory representation of the database table or file-based data source such as Excel worksheet, text and XML files, etc
Connection	Either database connection or connection to the file-based data set such as Excel worksheet, text and XML files, etc.
Connector	A pluggable add-on which reads and writes data in the particular format.
Streaming	A way to copy data from the source to destination using very small memory footprint. Basically only current row (record) is stored in the memory
Mapping	A way to map a field in the source to the field in the destination
Automatic mapping	Field in the source is mapped to the field in the destination by name
Scenario variable	Input parameter
Destination variable	Data set field or calculated variable

## How ETL Engine Works

Short version - ETL engine reads data sources, performs transformations and generates database-specific SQL code which is then executed within a transaction. If destination is not a SQL database, ETL engine uses pluggable connectors to write data in the designated format. The ETL scenarios are written in the XML-based language, but it is also possible to create them as Java objects.

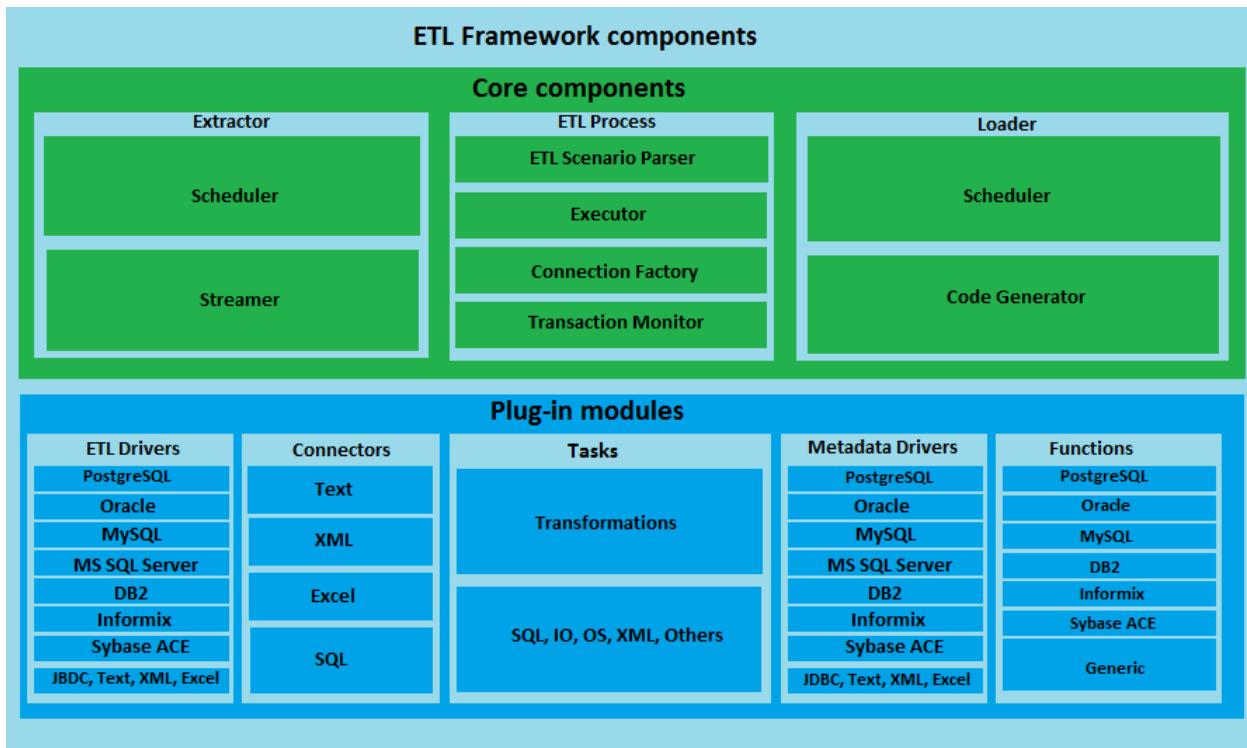


Figure 49: ETL Framework components

There are three main pillars:

**First**, the code generated by the ETL engine is specific to the database. For example for Oracle, it is PL/SQL code, for Microsoft – Transact-SQL, etc. The code is extremely efficient and supports a wide range of techniques from temporary tables and cursors to the native extract and bulk load. It's all done automatically, behind the scenes, by the translation layer and usually does not require any specific knowledge about the target database.

**Second**, the ETL engine supports data streaming where “reading” and “writing” are combined in one operation. Basically it allows moving practically unlimited sets of data from the source to destination. It is also done automatically.

**The transformation suite is a third pillar.** The high-level transformations such as de-duplication, pivoting, de-normalization, etc are all built in. There are also programmable transformations and validations. The ETL engine supports multithreading at all levels: from extract to load to executing individual ETL scenarios.

Perhaps most important, the ETL engine is easily expandable. All core components (such as drivers, connectors, transformations, functions, code generators etc.) are dynamically loaded plug-in modules. It is easy to add new or modify existing functionality. It is also easy to integrate it into your application by either embedding the ETL engine or running it in the client-server mode.

## Features

The full list of features of the ETL Integrated Development Environment can be found [here](#).

## Examples of the ETL Scenario

You can find examples of the ETL scenario [here](#).

### ETL Scenario Editor

ETL Scenario Editor is integrated directly into nodes browser. The **ETL Scenarios** node is a root for all scenarios. It cannot be deleted or modified. When within an ETL Scenarios segment of the nodes tree click on the “add node” button to add a new scenario or scenario’s folder. Use “delete node” button to delete currently selected scenario or scenario’s folder. When you delete a scenario’s folder all underlying scenario nodes will be deleted as well. The actual scenario files will not be deleted.

ETL Scenario Editor supports multiple tabs.

Use copy , cut and paste buttons to create a new ETL scenario from the existing one.

When new ETL scenario is created or existing scenario node is selected in the nodes browser the ETL Editor is displayed in the App panel.

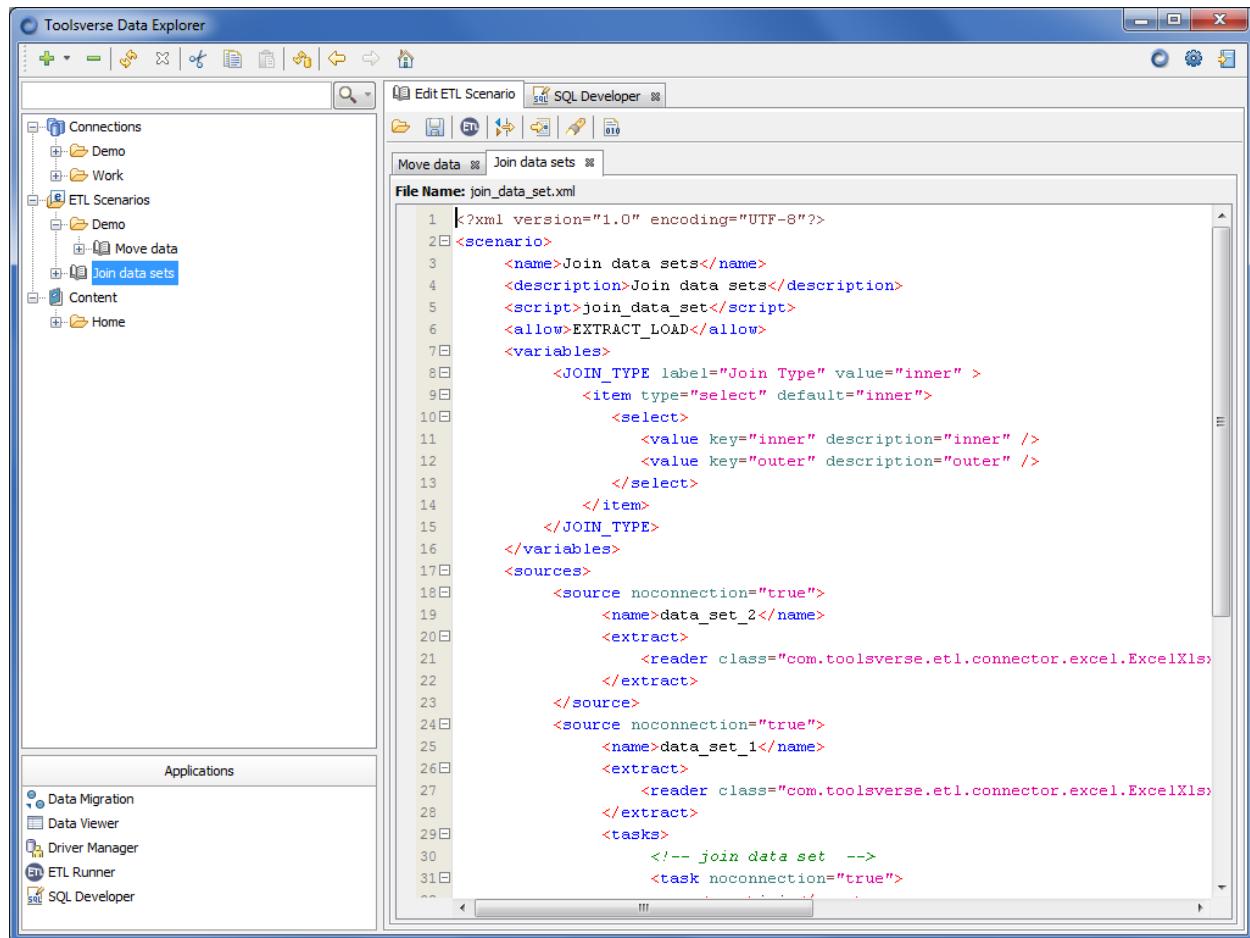


Figure 50: ETL Editor with multiple opened tabs

### *Commands*

Icon	Function	Windows and others	OS X	Web browser
	Open existing ETL scenario	Ctrl+O	Command+O	Ctrl+O
	Save ETL scenario file and update node	Ctrl+S	Command+S	Ctrl+S
	Execute ETL Scenario	Alt+F8	Alt+F8	Alt+F8
	<a href="#">Format Code</a>	Shift+F9	Shift+F9	Shift+F9
	<a href="#">Go To the Line</a>	Ctrl+L	Command+L	Ctrl+F8
	<a href="#">Search and Replace</a>	Ctrl+F	Command+F	Ctrl +F7
	<a href="#">Show ETL Code Snippets</a>	Ctrl+F12	Command+F12	Ctrl+F12

## ETL Runner

ETL Runner provides user interface for executing ETL scenarios from the Data Explorer. It can be opened from the ETL scenario editor or as an independent app from the Applications list.

ETL Runner parses ETL scenario and creates UI for entering scenario variables and connections. It also validates syntax, connections and variables before executing scenario. The ETL scenario is executed in the separate thread and it is possible to interrupt it at any time - the ETL engine will rollback all changes.

To open ETL Runner application use “run etl scenario”  button from the ETL Scenario Editor or click “etl runner”  icon in the Applications list.

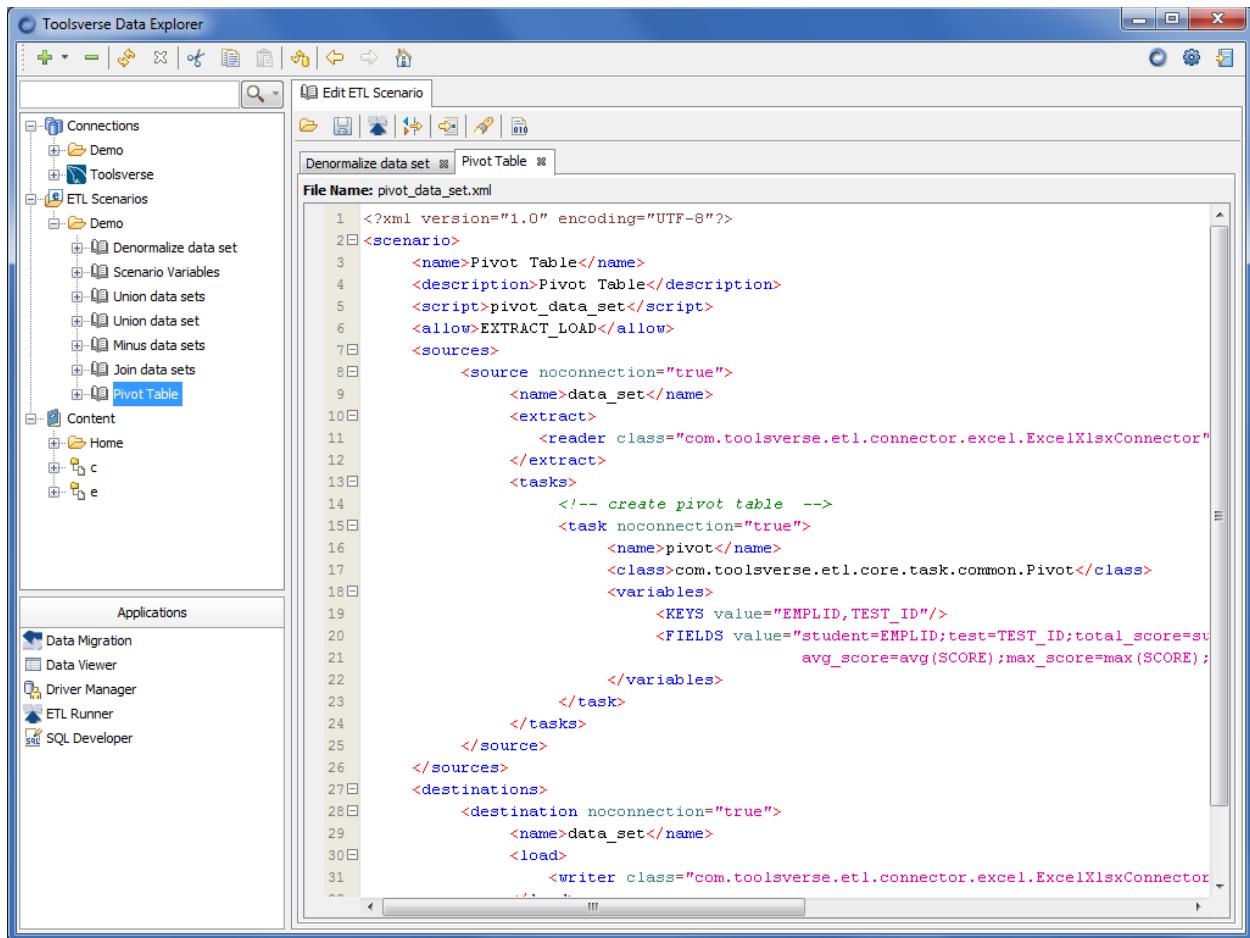


Figure 51: ETL Runner

### Commands

Icon	Function	Windows and others	OS X	Web browser
	Add ETL Scenario to run. Opens a dialog window with a tree of available ETL scenarios. If scenario has inner scenarios you can expand the node and select the one that you need.	Ctrl+N	Command+N	Shift+F4
	Re-parses ETL scenario and updates UI. You need it if you or somebody else made a change to the scenario already opened in the ETL Runner.	Ctrl+F9	Command+F9	Ctrl+F9
	Execute ETL scenario	Ctrl+F2	Command+F2	Ctrl+F2

### Fields

Field	Description
Action	The ETL action. Possible actions: <ul style="list-style-type: none"> <li>• Extract</li> <li>• Load</li> <li>• Extract and Load</li> </ul>
Connections	The list of the source and destination connections. The list is changing depending on the selected action
Variables	The scenario variables

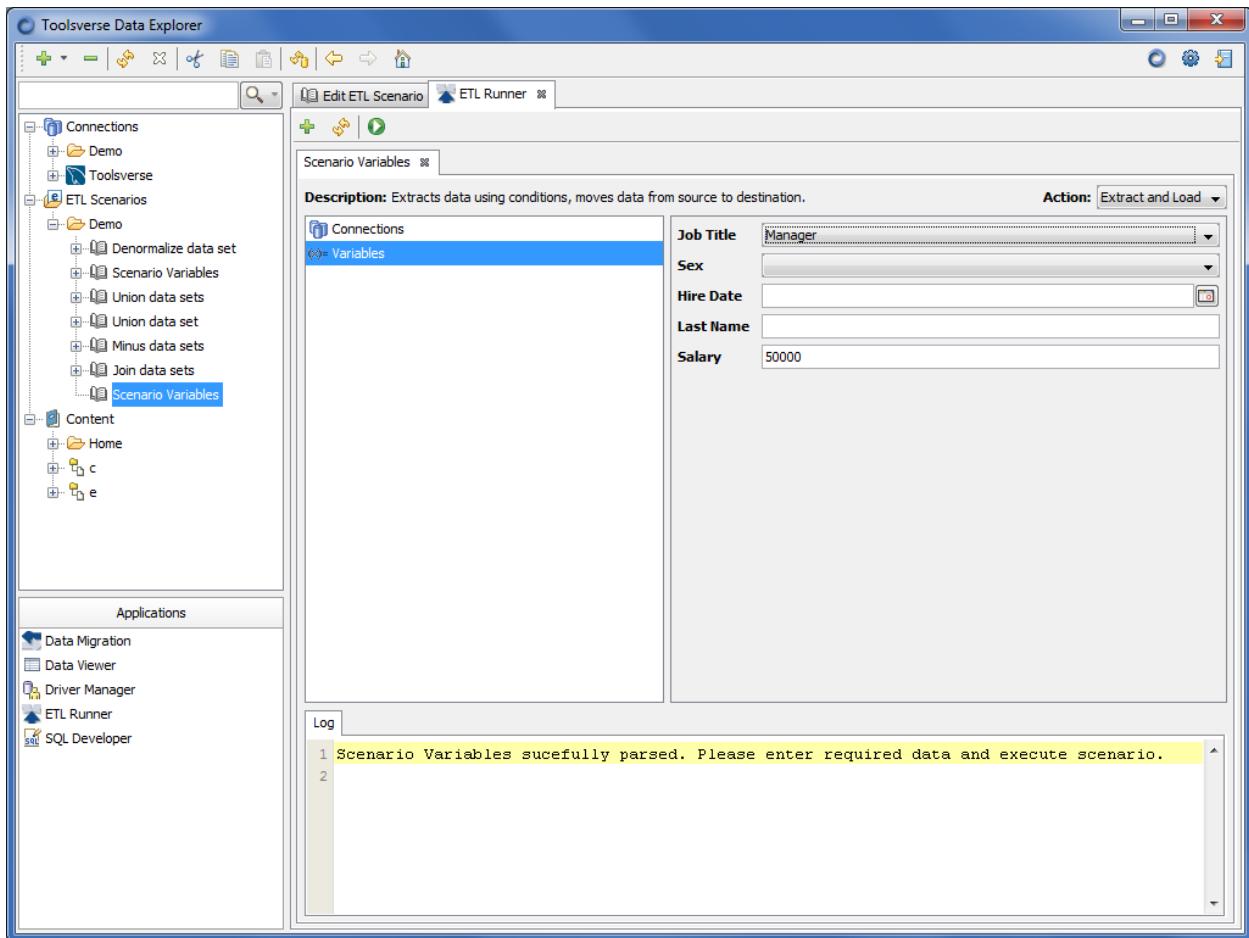


Figure 52: ETL scenario variables

## ETL Integrated Development Environment How To

### *Create ETL scenario*

You can find examples of the ETL scenarios [here](#).

### *Add New ETL Scenario*

To add new ETL scenario select any node inside ETL Scenarios segment in the nodes tree and click on the “add node” button in the Data Explorer toolbar. You can also add scenarios folder.

### *Save ETL Scenario*

If you made any changes to the existing or new scenario, click on the “save” button in the editor’s toolbar to save them.

The value of the attribute `<name/>` will be used as a node name and the value of the attribute `<script/>` as ETL scenario file name.

**Note:** You must save any changes before scenario can be executed.

All ETL scenarios are automatically stored in the APP\_HOME/data/scenario folder.

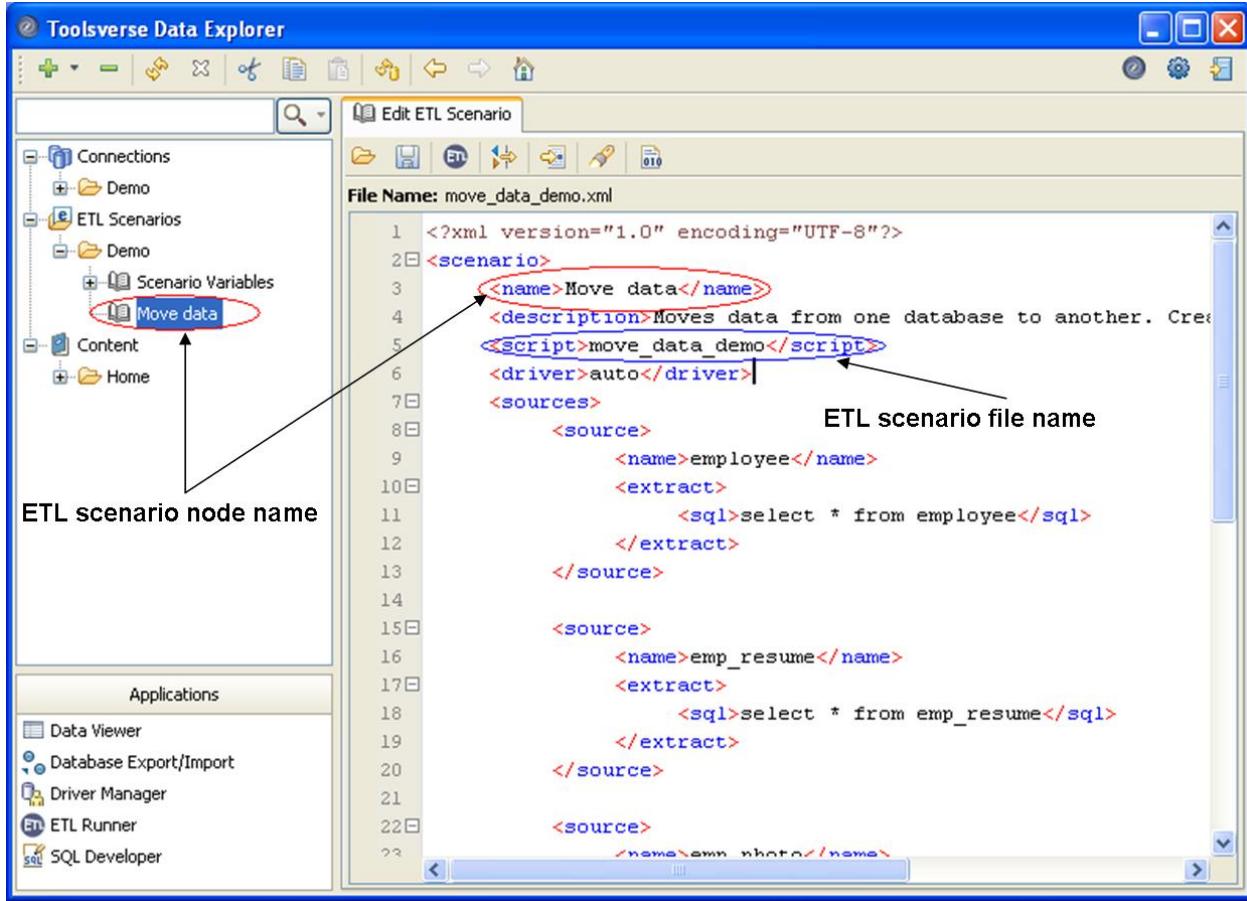


Figure 53: Save ETL Scenario

### Delete ETL Scenario node

To delete ETL Scenario node select node in the browser and click on the “delete node” button. Confirm deletion.

**Note:** The actual scenario file will not be deleted.

### Create New ETL scenario from the existing one

To create a new ETL scenario node form the existing one use “copy” , “cut” and “paste” commands.

### ***Link existing ETL Scenario to the current node***

To link existing ETL scenario to the currently selected node click on “open etl scenario”  button in the scenario editor toolbar. Choose ETL scenario to link and click on “Select Scenario” button. Click on the “Save”  button in the editor’s toolbar to save changes.

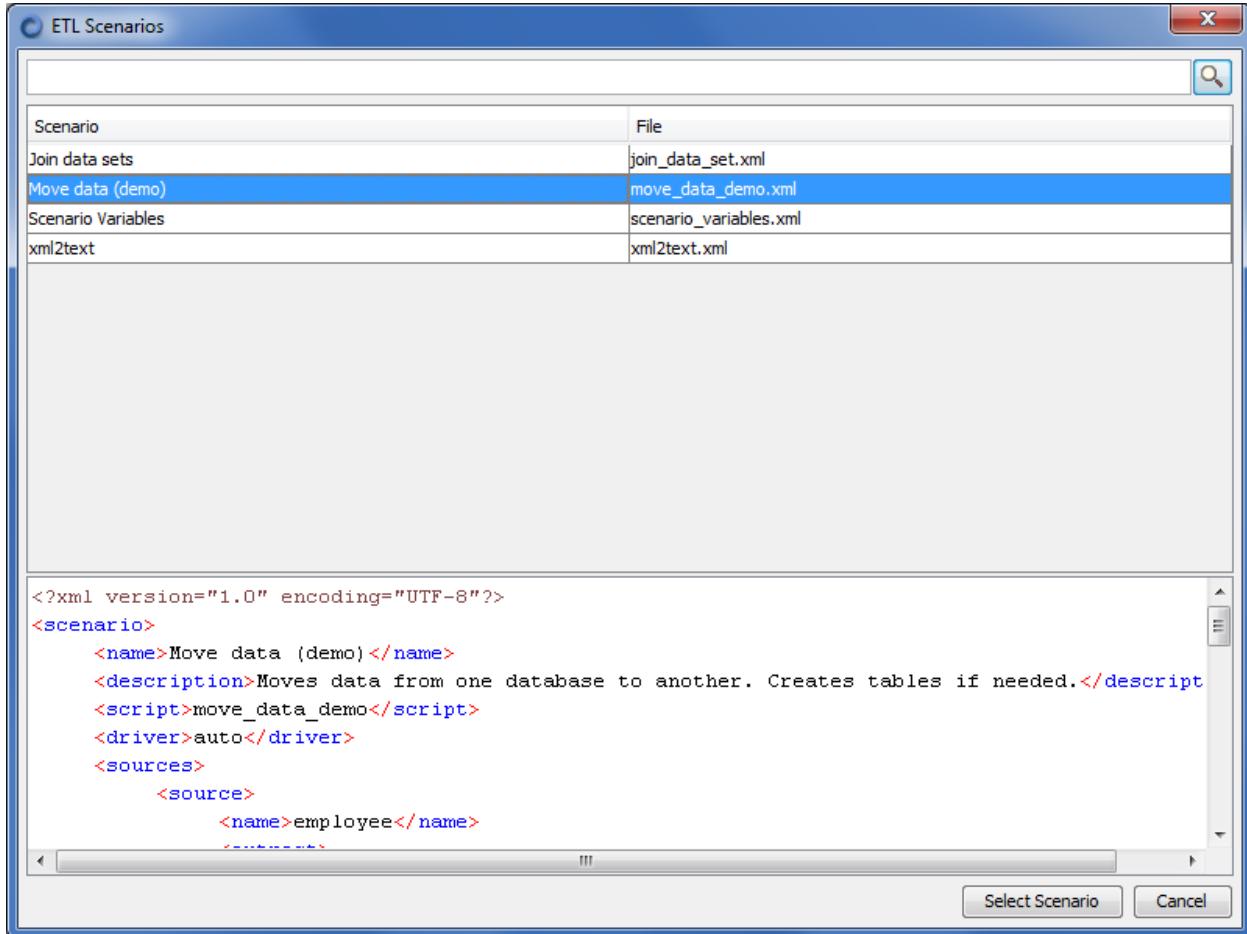


Figure 54: Open ETL Scenario

## **Expand ETL scenario**

If selected in the nodes tree ETL scenario has inner scenarios, the selected scenario can be expanded. When you change a list of inner scenarios and save your changes the nodes browser will reflect it.

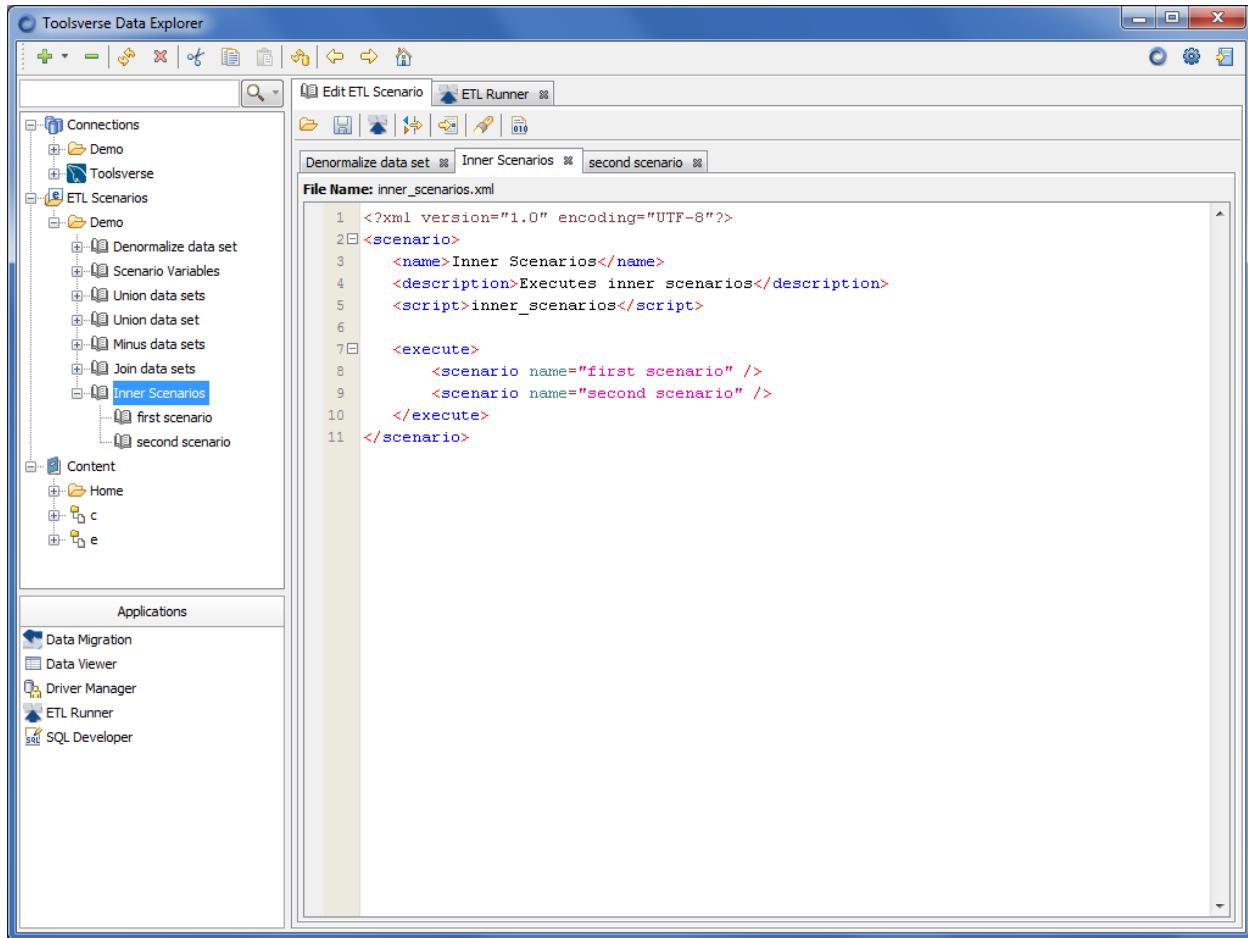


Figure 55 : Inner ETL scenarios

## **Execute ETL scenario from the editor**

To execute ETL scenario from the editor click on the “execute etl scenario”  button in the editor’s toolbar. If ETL Runner application is already running the new tab will be added to the ETL Runner. If there is already a tab for the selected scenario in the ETL Runner it will be selected.

## **Format ETL code**

To format ETL code click on  button in the ETL Editor toolbar.

See [Code Formatter](#)

## **Go to the Line in the ETL scenario editor**

To jump to the line click on  button in the ETL Editor toolbar.

See [Go to the Line](#)

## **Search and Replace text in the ETL scenario editor**

To search and replace in the text editor click on  button in the ETL Editor toolbar.

See [Search and Replace](#)

### ***View and Manage ETL Code Snippets***

To use ETL code snippets click on  button in the ETL Editor toolbar.

See [View and Manage Code Snippets](#)

### ***Open ETL Runner application***

To open ETL Runner application select “etl runner”  icon in the Applications list. You can also open it right from the ETL Scenario editor but clicking on “execute etl scenario”  button on the editor’s toolbar.

### ***Add scenario to run in the ETL Runner***

To add scenario to the ETL Runner click on the “select scenario”  button on the ETL Runner toolbar. Choose a scenario in the “select node” dialog window and click “ok” button. If scenario has inner scenarios you can expand the node and select the one that you need.

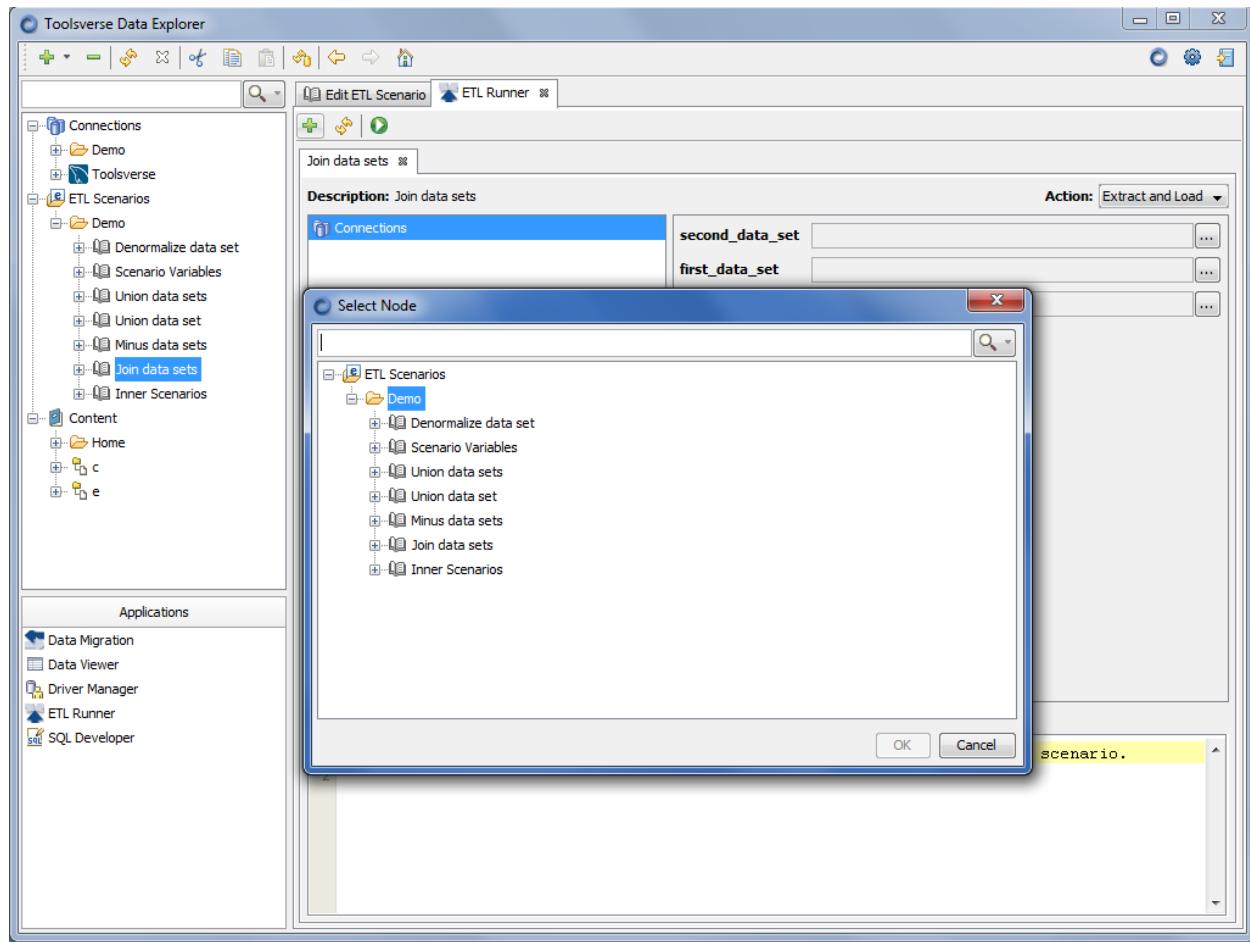


Figure 56: Add ETL Scenario

### ***Refresh ETL scenario***

If you or somebody else made a change to the scenario which is already opened in the ETL Runner you will need to refresh it. Refresh re-parses ETL scenario and updates UI.

To refresh ETL scenario click on “refresh”  button in the ETL Runner toolbar.

If you are accessing ETL Runner from ETL scenario editor and scenario is saved () you don't need to refresh it.

### **Execute ETL Scenario**

To execute ETL scenario click on “execute etl scenario”  button in the ETL Runner toolbar.

Most ETL scenarios can be executed in the “Extract” (only extract will be performed), “Load” (only load) and “Extract and Load” (extract and load will be performed) modes. Select appropriate action from the **Action** drop down list.

ETL Runner validates syntax and input data so you must enter all required variables and connections. If validation fails the Log is updated with all appropriate information.

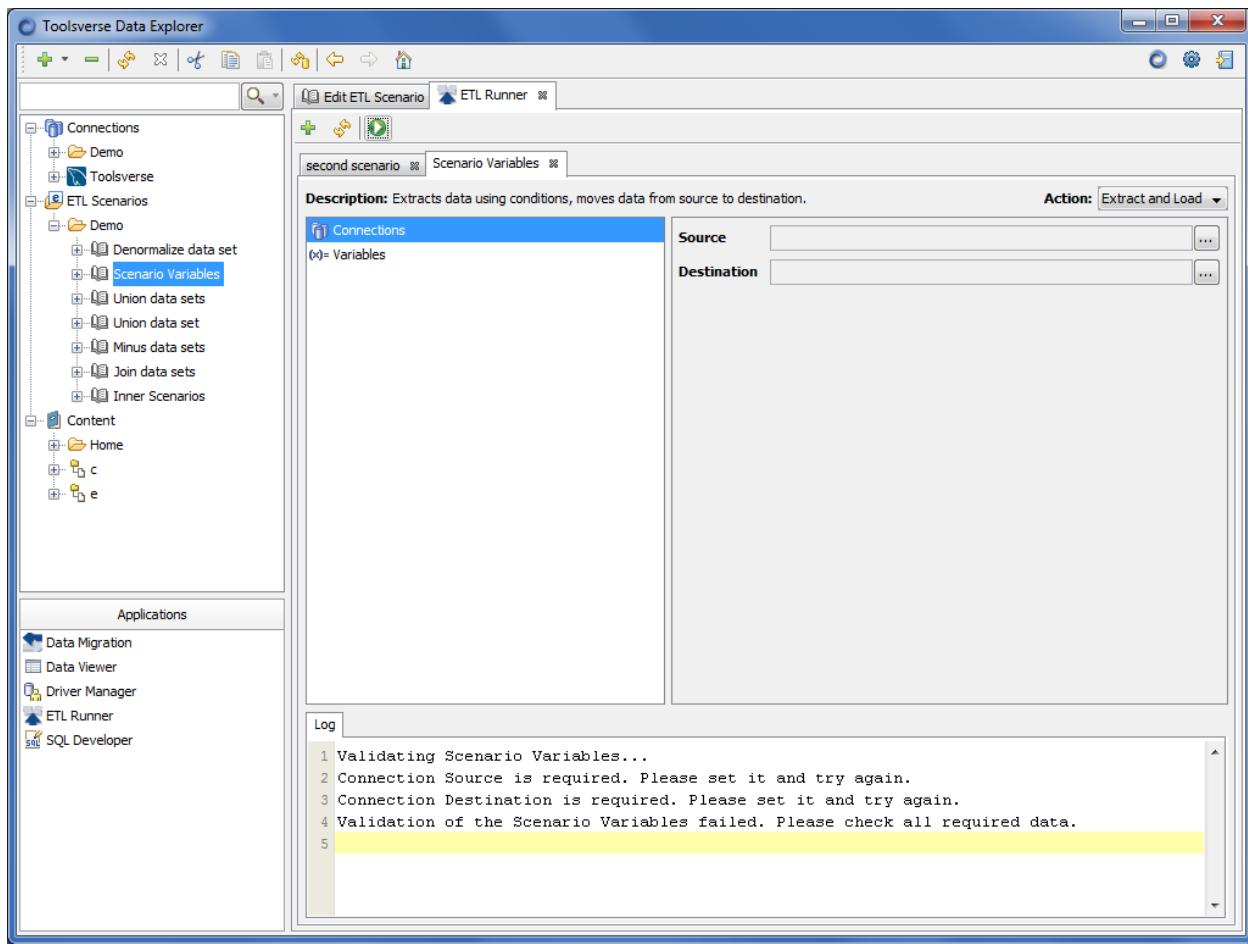


Figure 57: ETL validation failed

### **Interrupt ETL scenario execution**

When ETL scenario is running the progress dialog window is displayed. You cannot do anything in Data Explorer while ETL is running but you can interrupt (cancel) it at any time by pressing on “Cancel” button. When you cancel ETL process all modifications made in all affected databases and in the file system will be rolled back.

**Note:** In the Web mode the ETL progress dialog is disabled by default. You can enable it using the following access path: Settings->ETL Runner App-> Show Progress.

## One-click Data Migration

Data Migration application automates migration from one or multiple databases (and other data sources) to another. It does not require any programming skills.

All connectivity options (jdbc/odbc/XML/text/Excel) are supported. For example, you can copy data from Excel spreadsheet to PostgreSQL database or from Oracle to DB2 database using interactive graphical UI and without writing a single line of ETL code. If required, destination tables and indexes will be automatically created.

It is possible to change names of the destination tables.

Data Migration application supports data streaming which allows copying practically unlimited sets of data. It also supports parallel extract.

**Note:** You can combine data sets from the different sources in one export list. For example, export list can include tables from the Oracle database and worksheets from the Excel spreadsheet.

All data types including CLOBs and BLOBs are supported.

### Typical usage scenario:

You have a customer's database in the Excel or Access and want to move data to the more "mature" database, for example PostgreSQL or Oracle. All you need to do is:

1. Create a source connection for the Excel spreadsheet or Access database
2. Create a destination connection for the PostgreSQL or Oracle database
3. Add tables/worksheets from the source to the export list
4. Select "Extract and Load" action
5. Select a PostgreSQL or Oracle connection as Destination
6. Execute Data Migration scenario

The tables (and indexes, if requested) in the destinations database will be automatically created and data will be copied from the source database to destination. If table in the destination already exists it will be used. There is an option to delete all data from destination tables before loading.

To open Data Migration application click “database migration”  icon in the Applications list.

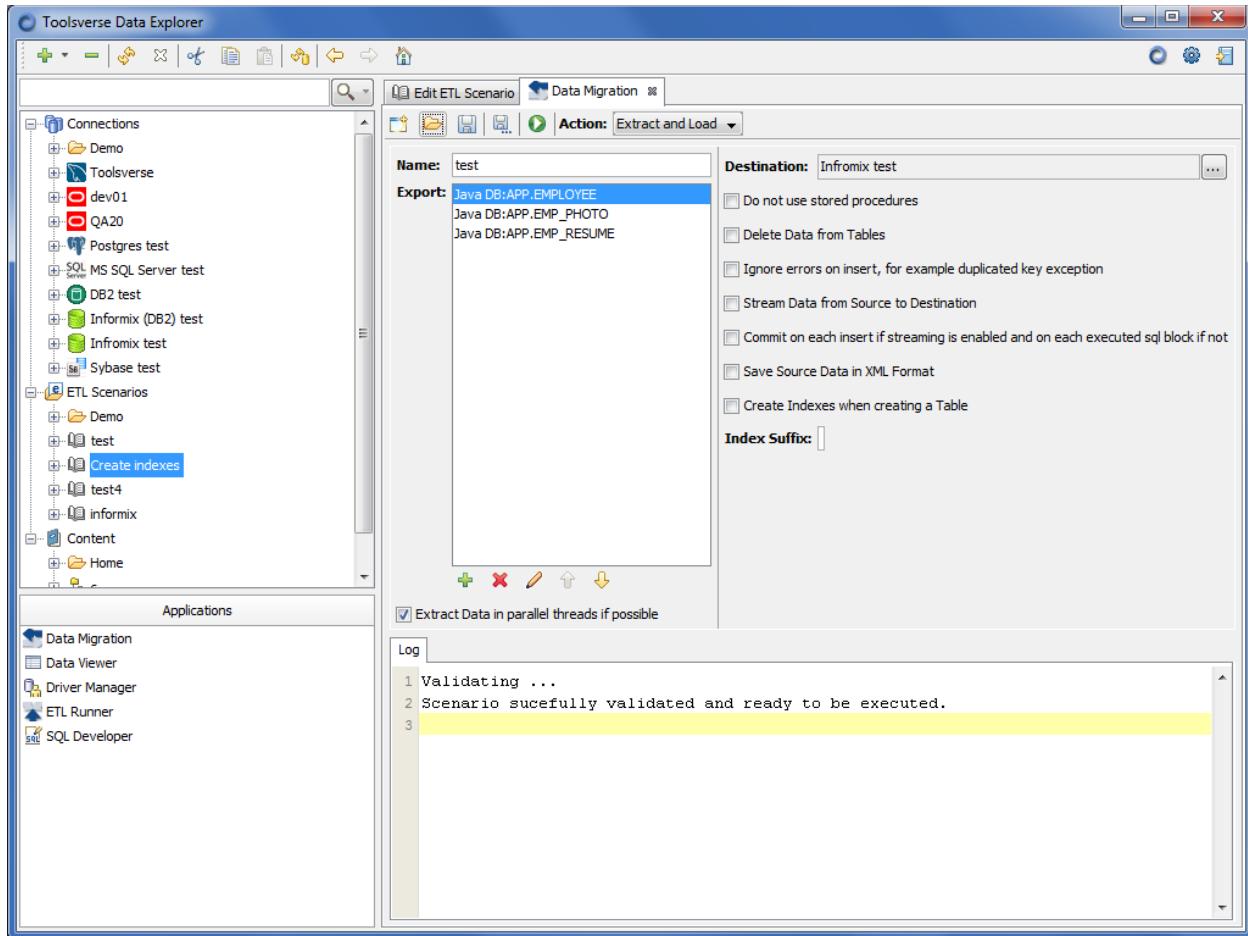


Figure 58: Data Migration application

## Features

The full list of features can be found [here](#).

## Commands

Icon	Function	Windows and others	OS X	Web browser
	Create new Data Migration Scenario	Ctrl+N	Command+N	Shift+F4
	Open Data Migration Scenario. Opens existing Data Migration scenario	Ctrl+O	Command+O	Ctrl F3
	Save Data Migration Scenario. Saves currently opened Data Migration scenario	Ctrl+S	Command+S	Ctrl+F6
	Save as ETL Scenario. Saves currently opened Data Migration scenario as ETL scenario	Ctrl+F11	Command+F11	Ctrl+F11

	Execute Data Migration Scenario.	Ctrl+F2	Command+F2	Ctrl+F2
	Action. Possible actions: <ul style="list-style-type: none"> <li>• Extract – extract data sets and creates XML files</li> <li>• Extract and Load – extracts source data sets and loads them into destination connection</li> </ul>			

## Fields

Field	Description	Visible for Action
Name	Name of the Data Migration scenario.	“Extract” and “Extract and Load”
Export:	Data sets to export. Can be: <ul style="list-style-type: none"> <li>• database tables (views, synonyms, etc)</li> <li>• Excel worksheets</li> <li>• Text files</li> <li>• XML files</li> </ul>	“Extract” and “Extract and Load”
Extract data in parallel threads if possible	If option is enabled each data set will be extracted in the parallel thread if it is possible	“Extract” and “Extract and Load”
Destination	The name of the destination connection	“Extract and Load”
Do not use stored procedures	If option is enabled system <b>will not</b> be using stored procedures or anonymous SQL blocks to load data	“Extract and Load”
Delete Data From Tables	If option is enabled and table already exists in the destination database – all records from the table will be deleted before load	“Extract and Load”
Ignore errors on insert, for example duplicated key exception	If option is enabled any error which occurred when executing insert SQL will be ignored. No exception will be generated and system will continue loading data. It can be useful to ignore errors such as duplicated key	“Extract and Load”
Stream Data from Source to Destination	If option is enabled the data sets will be streamed from the source to destination. Streaming takes much less memory because only current record is stored in memory. <b>Note:</b> Using this option you can transfer practically unlimited sets of data	“Extract and Load”
Commit on each insert if streaming is enabled and on each executed sql block if not	If option is enabled system will commit transaction on each insert (if streaming is enabled) or on each executed SQL block	“Extract and Load”
Save Source Data in XML format	If option is enabled the source data sets will be additionally “saved” in XML format	“Extract and Load”

Create Indexes and creating a Table	If option is enabled , destination table doesn't exist yet and source table has indexes the same indexes will be created for the destination table as well	"Extract and Load"
Index Suffix	The suffix will be added to the index name. Use it if you think that index with the same name already exists	"Extract and Load"

## Objects List Commands

Icon	Command
	Add Database Objects to the Export List.
	Delete Database Object from the Export List.
	Edit Table Name. You can use it to change a name of the destination table. By default destination table name = source table name.
	Move Database Object up the Export List. If "Extract data in parallel threads" checkbox is "checked" the order of objects is ignored
	Move Database Object down the Export List. If "Extract data in parallel threads" checkbox is "checked" the order of objects is ignored

## Data Migration How To

### Open Data Migration application

To open Data Migration application click "Data Migration" icon in the Applications list.

### Create new Data Migration Scenario

To create a new data migration scenario click icon in the Data Migration app toolbar and confirm selection.

### Add object to Export List

To add objects (tables, views, synonyms, worksheets, files, etc) to the export list click on "add database object" button located below Export list. Select object from the tree. Click "Ok" button to confirm. The "Select Node" dialog window will remain opened until you are done.

**Note:** Only specific types of objects can be added to the list. "Ok" button is enabled only for those objects.

### Remove object from Export List

To remove object from the export list click on "delete database object" button located below Export list.

### Change a name of the Destination Table

By default the name of the destination name is equal to the name of the source table. You can change a name by clicking on the "edit" button located below Export list.

### Change order of objects in the Export List

To change order of objects in the export list use "up" and "down" buttons located below Export list. If "Extract data in parallel threads" checkbox is "checked" the order of objects is ignored.

## **Export Data**

If you only want to export data – select “Extract” action and click on “execute”  button in the Data Migration toolbar. The XML files (each represents one data set) will be created in your local [DATA folder](#).

## **Export and Import Data**

If you want export and import data (extract and load):

1. Select “Extract and Load” action
2. Set the Destination connection
3. Click on “execute”  button in the Data Migration toolbar

## **Save Data Migration Scenario**

You can save current Data Migration scenario including name, export list, destination connection and all settings by clicking on “save”  button in the Data Migration toolbar.

## **Open existing Data Migration Scenario**

You can open previously saved Data Migration scenario by clicking on “open”  button in the Data Migration toolbar. Select scenario from the list and click on “Select Scenario” button.

## **Save Data Migration Scenario as ETL Scenario**

Data Migration scenarios and ETL scenarios have different formats and stored in different places but internally Data Migration scenario is converted to ETL scenario before it is executed. To save current Data Migration scenario as ETL scenario, click on “save as”  button in the Data Migration toolbar.

## **Interrupt Data Migration scenario execution**

When scenario is running the progress dialog window is displayed. You cannot do anything in Data Explorer while ETL is running but you can interrupt (cancel) it at any time by pressing on “Cancel” button. When you cancel ETL process all modifications made in all affected databases and in the file system will be rolled back.

**Note:** In the Web mode the progress dialog is disabled by default. You can enable it using the following access path: Settings->Data Migration App-> Show Progress.

## **Run Extract in the parallel threads**

In most cases extract of the **each data set** can be executed in the parallel thread. On the computer that supports hardware threads it can greatly reduce a time needed to finish the task. To enable multithreading for the extract check option "Extract data in the parallel threads is possible".

**Note:** if multithreading is enable the order of the objects in the Export list does not really matter.

## **Stream data from the source to destination**

By default when data migration scenario is executed application creates an in-memory table for the each data source and then loads it into the destination. In most cases it is possible to stream data from the source to destination with a little to none memory allocation. Using this option you can copy practically unlimited data sets. To enable streaming check option "Stream data from Source to Destination".

### *Create Indexes when creating a table*

When data migration scenario is executed and table does not exist in the destination, application automatically creates it using source table data structure as a pattern. It is possible to create indexes as well (by default it is disabled). To create indexes check "Create Indexes when creating a Table" option. By default index created using the same names as in the source database. You can use "Index Suffix" to change index name from the default. Suffix will be added to the end of the each index name.

**Note:** if table already exists indexes will not be created.

### *Delete data from the destination tables*

It is possible to run a data migration scenario on the database which already has destination tables. Some (or all) of these tables might have data. By checking option "Delete Data from Tables" you will enable deleting from the destination tables before loading.

**Note:** The delete might fail if there are primary-foreign key constraints which enforce data integrity.

## Content Management

Content Management application is a file manager which supports a wide range of file formats including Excel, images, XML, text, etc. It is integrated directly into the Data Explorer nodes browser. You can view, in some cases edit, delete, copy and move files and folders between different nodes.

For the Excel files the following additional functions are available:

- filtering
- sorting
- perform transformations
- apply functions
- export data
- display data as a chart

## Features

The full list of features can be found [here](#).

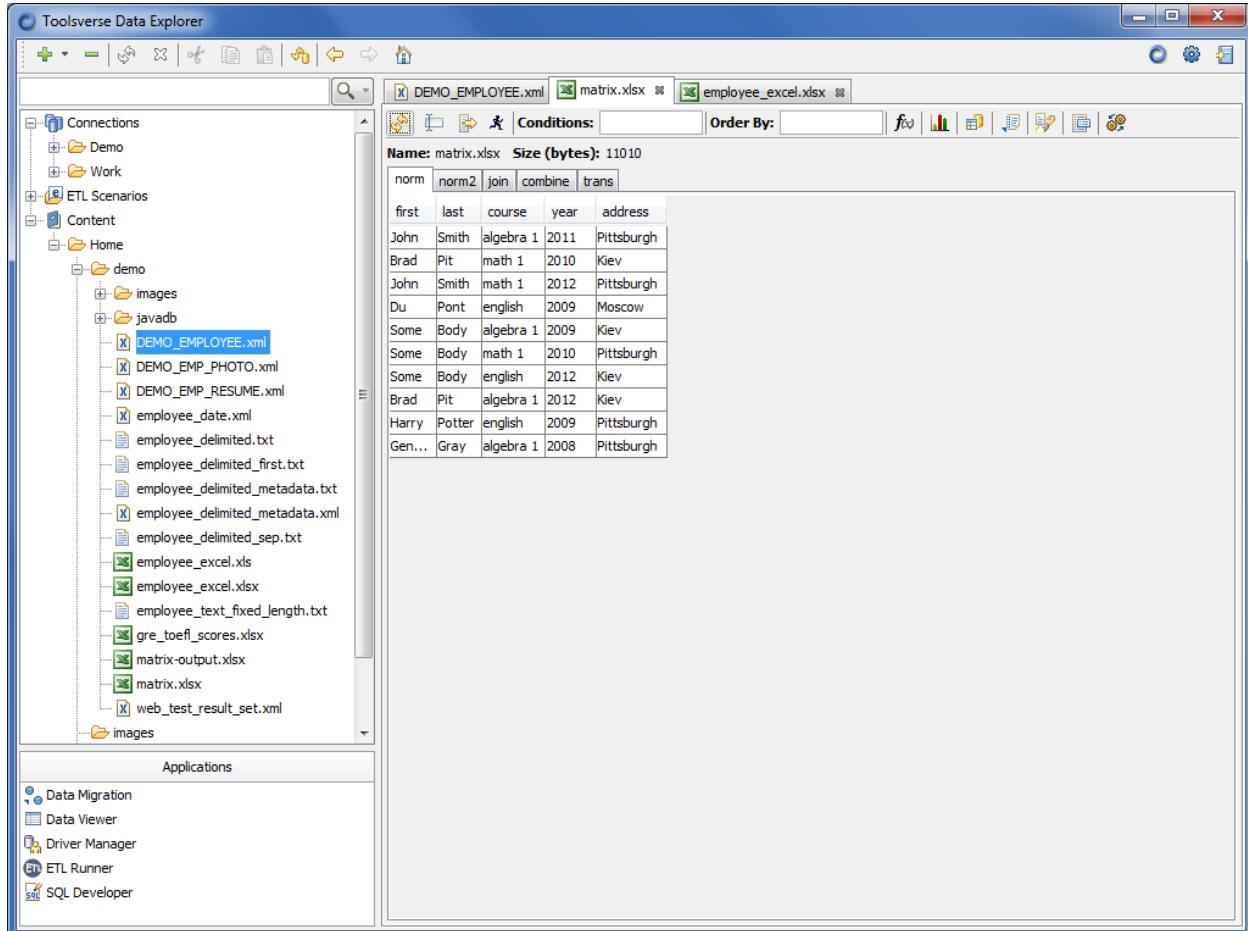
The following node types are supported:

- Local – the local computer hard drives (client and client-server modes only)
- Home - the shared DATA folder and user's personal folder in the Web and client-server mode
- FTP site
- SFTP site

For example, you can copy data files from the remote FTP server to your personal folder on the Web server where they will be available to the ETL engine. Or you can view an Excel spreadsheet in the shared data folder.

**Note:** Data Explorer uses Home folder by default to store all sort of files, from ETL scenarios to data files. In the Web and client-server modes there is a shared Home folder and personal folder under Home. Each user has his/her own personal folder. The personal folder name is calculated using login name so it is important to have unique logins for each user in the multi user modes such as Web and client-server.

To view and manage content (files and folders) select any node within “content”  segment of the nodes browser. Expand nodes and folders to find files that you need.



**Figure 59: Content Management**

**Note:** The number of files which can be opened in the separate tabs is limited. You can change a limit using following access path: Settings->Content Node->Maximum Number of Tabs.

Use “add node”  button in the Data Explorer toolbar and select a node type in the popup menu to add a new root level node, folder or file and “delete node”  to delete it. Use “refresh”  button to refresh node or folder.

### Root level nodes

Icon	Type
	Local - the local hard driver (client and client-server modes only)
	Home - the shared DATA folder and user's home folder in the Web and client-server mode
	FTP site
	SFTP site

## Commands

The list of commands is different for each node and file type.

### *Root level content node (Home, Local, FTP, SFTP)*

Icon	Function	Windows and others	OS X	Web browser
	Save node	Ctrl+S	Command+S	Ctrl+S
	Sort files: • By Name • By Type • By Size • By Date	<ul style="list-style-type: none"> <li>• Ctrl+F7</li> <li>• Ctrl+F8</li> <li>• Ctrl+F9</li> <li>• Ctrl+F10</li> </ul>	<ul style="list-style-type: none"> <li>• Command+F7</li> <li>• Command +F8</li> <li>• Command +F9</li> <li>• Command +F10</li> </ul>	<ul style="list-style-type: none"> <li>• Ctrl+F7</li> <li>• Ctrl+F8</li> <li>• Ctrl+F9</li> <li>• Ctrl+F10</li> </ul>

### *Folder node*

Icon	Function	Windows and others	OS X	Web browser
	Rename folder			
	Sort files: • By Name • By Type • By Size • By Date	<ul style="list-style-type: none"> <li>• Ctrl+F7</li> <li>• Ctrl+F8</li> <li>• Ctrl+F9</li> <li>• Ctrl+F10</li> </ul>	<ul style="list-style-type: none"> <li>• Command+F7</li> <li>• Command +F8</li> <li>• Command +F9</li> <li>• Command +F10</li> </ul>	<ul style="list-style-type: none"> <li>• Ctrl+F7</li> <li>• Ctrl+F8</li> <li>• Ctrl+F9</li> <li>• Ctrl+F10</li> </ul>

### *File node*

Icon	Function	Windows and others	OS X	Web browser
	Refresh file content (reload file)			
	Rename file			
	Copy file to (folder, node)			
	Move file to (folder, node)			
	Save file if editing is enabled	Ctrl+S	Command+S	Ctrl+S
	Parser – select a syntax highlighter			

	Go To the Line	Ctrl+L	Command+L	Ctrl+F8
	Search and Replace	Ctrl+F	Command+F	Ctrl +F7

### *Excel file node*

Icon	Function	Windows and others	OS X	Web browser
	Refresh file content (reload file)			
	Rename file			
	Copy file to (folder, node)			
	Move file to (folder, node)			
	Condition - the <i>where clause</i> . Example: lastname = 'STERN' or job = 'MANAGER.  <b>Note:</b> You must click on the "refresh"  button			
	Order by - the sort criteria. Example: firstnme, lastname desc  <b>Note:</b> You must click on the "refresh"  button when you change <b>order by</b> .			
	<a href="#">Calculate function</a>	Alt+F12	Alt+F12	Alt+F12
	<a href="#">Display Data as a Chart</a>	Shift+F10	Shift+F10	Shift+F10
	<a href="#">Search in the Data Set</a> . The data set is a currently displayed worksheet	F7	F7	F7
	<a href="#">Show Data Set Record in the Form view</a> . The data set is a currently displayed worksheet	F4	F4	F4
	<a href="#">Transform Data Set</a>	Shift+F8	Shift+F8	Shift+F8

### *Image file node*

Icon	Function	Windows and others	OS X	Web browser
	Refresh file content (reload file)			
	Rename file			
	Copy file to (folder, node)			
	Move file to (folder, node)			
	Scale Image (%)			

### *Fields*

Depending on the node type the following fields can be used to describe root level content node:

#### *Local*

Field	Description
Name	The name of the node
Folder	The root level folder. For example c:\

#### *Home*

Field	Description
Name	The name of the node

#### *FTP*

Field	Description
Name	The name of the node
Folder	The root level folder.
Url	The ftp url. Must start with ftp://. Can include port . The default port is 21
User	The account name
Password	The account password
Passive mode	If checked the passive ftp mode will be used
Proxy Host	The socks proxy host name
Proxy Port	The socks proxy port

#### *SFTP*

Field	Description
Name	The name of the node
Folder	The root level folder.
Url	The sftp url. Can include port . The default port is 22
User	The account name
Password	The account password

## Content Management How To

### Add new root level content node such as Home, Local, SFTP and FTP

To add new root level content node click on “add node”  button in the Data Explorer toolbar. Choose “add home”  for Home node, “add local”  for Local node, “add sftp”  for SFTP and “add ftp”  for FTP. Enter all required fields. Use “save”  button in the node editor toolbar to save changes.

### Add new folder

To add new folder to the root level content node or to another folder, select a node where you want folder to be added to. Click on “add node”  button in the Data Explorer toolbar and select “add folder”  menu item. Enter folder name. Click on “add” button.

### Add file

To add file to the folder select a folder where you want file to be added to. Click on “add node”  button in the Data Explorer toolbar and select “add file”  menu item. Select a local file (file in the local hard driver) in the file chooser dialog window. Click on “add” button.

**Note:** When in the Web mode the uploaded file must be a file in your local PC. It will be uploaded into designated node/folder including remote FTP and SFTP sites.

**Note:** When file is uploaded into Web server its size is limited. To change a limit from default 64K, use the following access path: Settings->Content Node->Maximum File Size to Upload.

### Delete node, folder, file

To delete a root level content node, folder or file click on “delete node”  button in the Data Explorer toolbar. Confirm that you want to delete it.

### Copy file from one folder to another

You can copy a file from one folder to another including folders in the different root content nodes. For example you can copy a file from the folder /usr/local/your\_name in the FTP site to your personal data folder in the Web server where it will be available to the ETL engine.

To copy file you need to select it in the nodes browser and click on “copy to folder”  button in the node editor toolbar. Select a destination folder in the “Select Node” dialog window. Click “Ok” button to confirm. You will need to select a destination folder and click on “refresh”  button in the Data Explorer toolbar to see the file.

### Move file from one folder to another

You can move a file from one folder to another including folders in the different root content nodes. For example you can move a file from the folder /usr/local/your\_name in the FTP site to your personal data folder in the Web server where it will be available to the ETL engine.

To move file you need to select it in the nodes browser and click on “move to folder”  button in the node editor toolbar. Select a destination folder in the “Select Node” dialog window. Click “Ok” button to confirm. You will need to select a destination folder and click on “refresh”  button in the Data Explorer toolbar to see the file.

### ***Rename file or folder***

To rename file or folder click on “rename”  button in the node editor toolbar. Click “Ok” button to confirm changes.

### ***Refresh a folder***

To refresh a folder – select it in the nodes browser and click on “refresh”  button in the Data Explorer toolbar.

### ***See content of the file***

If file format is recognized as one of the supported text formats, such as xml, txt, etc the content of the file is automatically displayed once you select file in the nodes browser. Text files can be edited. In the Web mode the size of the text file which can be edited is limited to the value of the configuration properly which can be changed using the following access path: Settings->Content Node->Maximum Text File Size.

If file format is recognized as one of the supported binary formats, such as Excel, image formats, etc you must click on “preview or refresh file”  button in the nodes editor toolbar to see content of the file. You can automatically preview file right after it is selected in the nodes browser if you set configuration properly Settings->Content Node->Preview files in Binary Formats to “checked”.

In all other cases (when format is not recognized) just click on “preview or refresh file”  button in the nodes editor toolbar to see content of the file.

The content of the Excel (\*.xls and \*.xlsx) files can be displayed in multiple tabs (one tab for each worksheet).

### ***Edit File***

To edit content of the file select a file in the nodes browser. When file is displayed look for “save”  button in the node editor toolbar. If it is present and enabled you can modify file and save changes by pressing on the “save” button.

**Note:** In the Web mode the size of the text file which can be edited is limited to the value of the configuration properly which can be changed using the following access path: Settings->Content Node->Maximum Text File Size. Currently you can edit files only in the text (xml included) format.

## Security Policy Editor (Web and client-server modes only)

When Data Explorer works in the Web or client-server modes the role based security is automatically activated. User must login in to the application using unique credentials.

Each user's account has a list of roles associated with it. Role can have one or multiple permissions (rights). There are two types of permissions: permission to access particular application (for example SQL Developer) and permission to access node. Some permissions can have one or more associated actions such as "edit", "expand", etc.

Only allowed nodes and applications are displayed and only allowed actions are permitted.

**Note:** Data Explorer uses Home folder by default to store all sort of files, from ETL scenarios to data files. In the Web and client-server modes there is a shared Home folder and personal folder under Home. The personal folder name is calculated using account name so it is important to have unique logins for each user.

Security Policy editor is a Data Explorer application for creating and managing user accounts and roles. It is only available in the Web and client-server modes.

To open Security Policy editor click on "security policy"  icon in the Applications list.

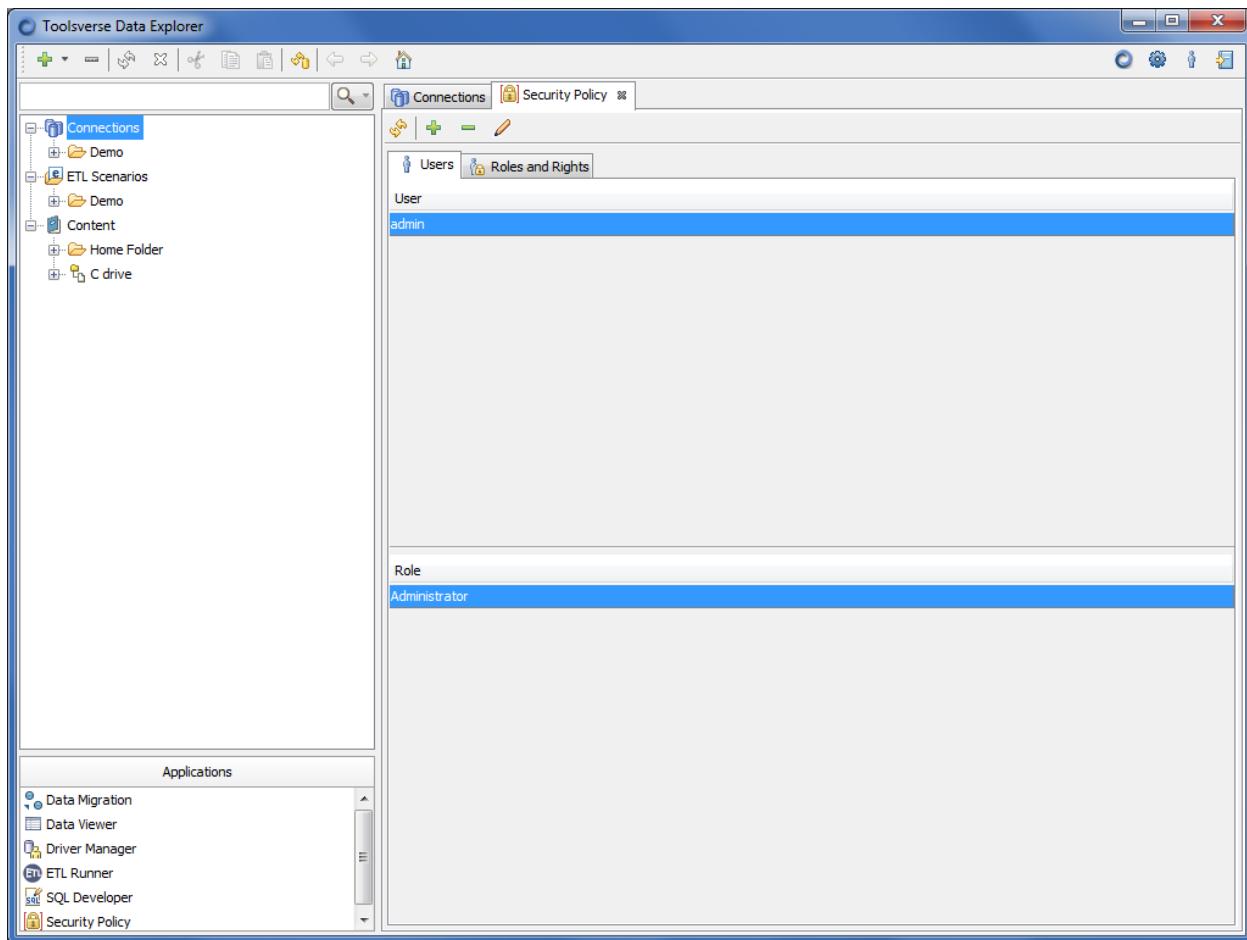


Figure 60: Security Policy Editor

## Commands

Icon	Command
	Refresh security policy. Refreshes users, roles and user to role associations. Can be useful if other user is modifying security policy at the same time
	Add new user or role.
	Edit user or role. The pre-defined “admin” roles cannot be deleted.  <b>Note:</b> It is possible to delete a single existing user. In this case there will be no way to login so be careful.
	Edit user or role. The pre-defined “admin” roles cannot be edited

## Pre-defined user

When Data Explorer is installed first time there is user **admin** which has **administrator** role assigned. The default password is **admin**. Password can be changed and user can be deleted at any time.

## Pre-defined roles

The pre-defined roles cannot be edited or deleted.

Role	Description
Administrator	No restrictions
Node Administrator	Can create, modify, delete, view and expand all nodes
Node View Only Administrator	Can view and expand all nodes
Apps Administrator	Can use all apps and plug-ins

## User Editor

To open User Editor click on “add user”  or “edit user”  button in the Security Policy editor. The “Users” tab must be selected.

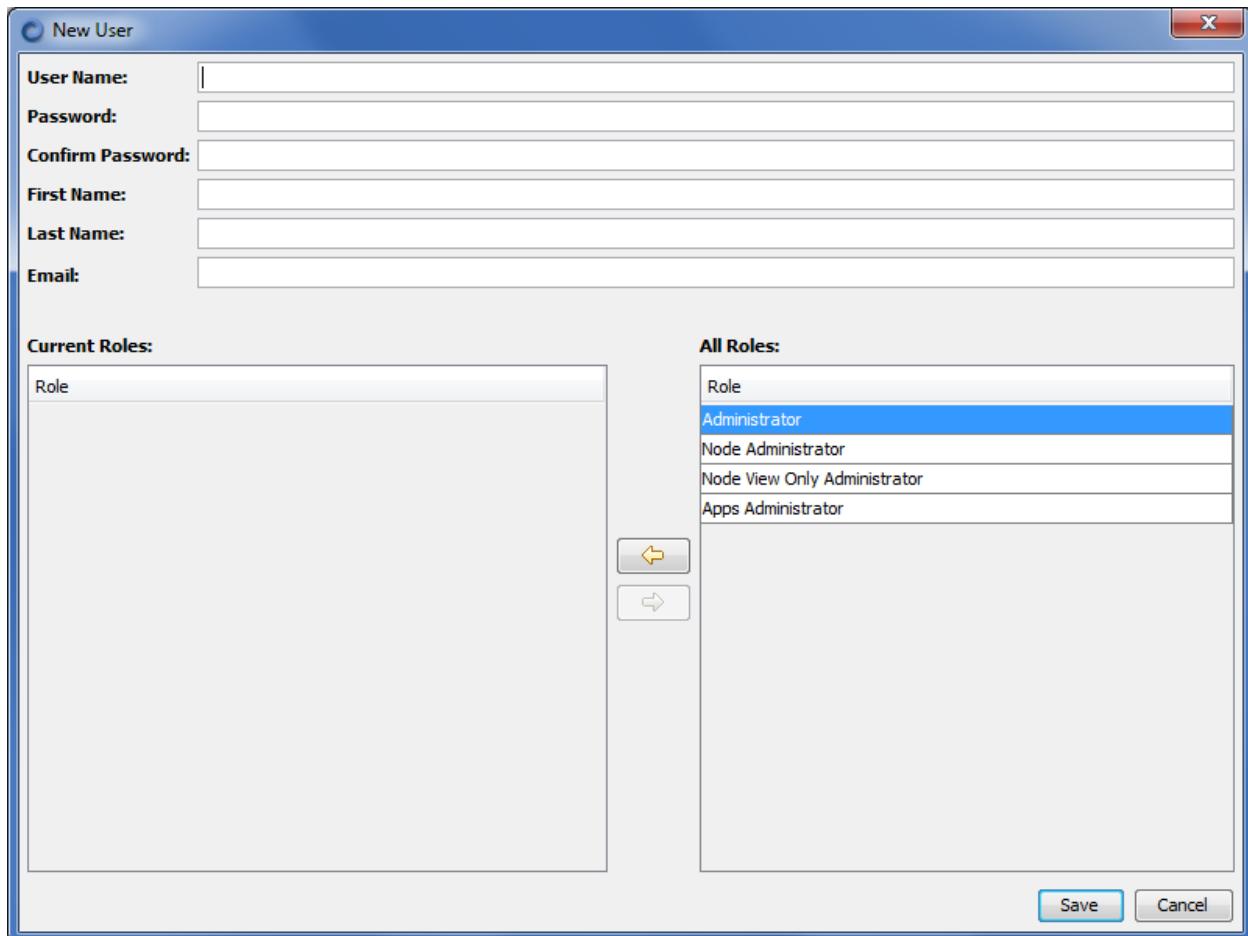


Figure 61: User Editor

### Fields

Field	Description	Is required
User name	The account name. Once saved cannot be changed	Yes
Password	The password	Yes
Confirm Password	The password confirmation. Should be the same as “password”. Field is enabled only when password has changed	No
First Name	User's first name	No
Last Name	User's last name	No
Email	User's email	No
Current roles	The list of assigned roles. Use  to assign role to the user and  to un-assign	No
All Roles	The list of roles	
Save	Saves changes	
Cancel	Closes dialog window	

## Role Editor

To open Role Editor click on “add role”  or “edit role”  button in the Security Policy editor. Roles and Rights” tab must be selected.

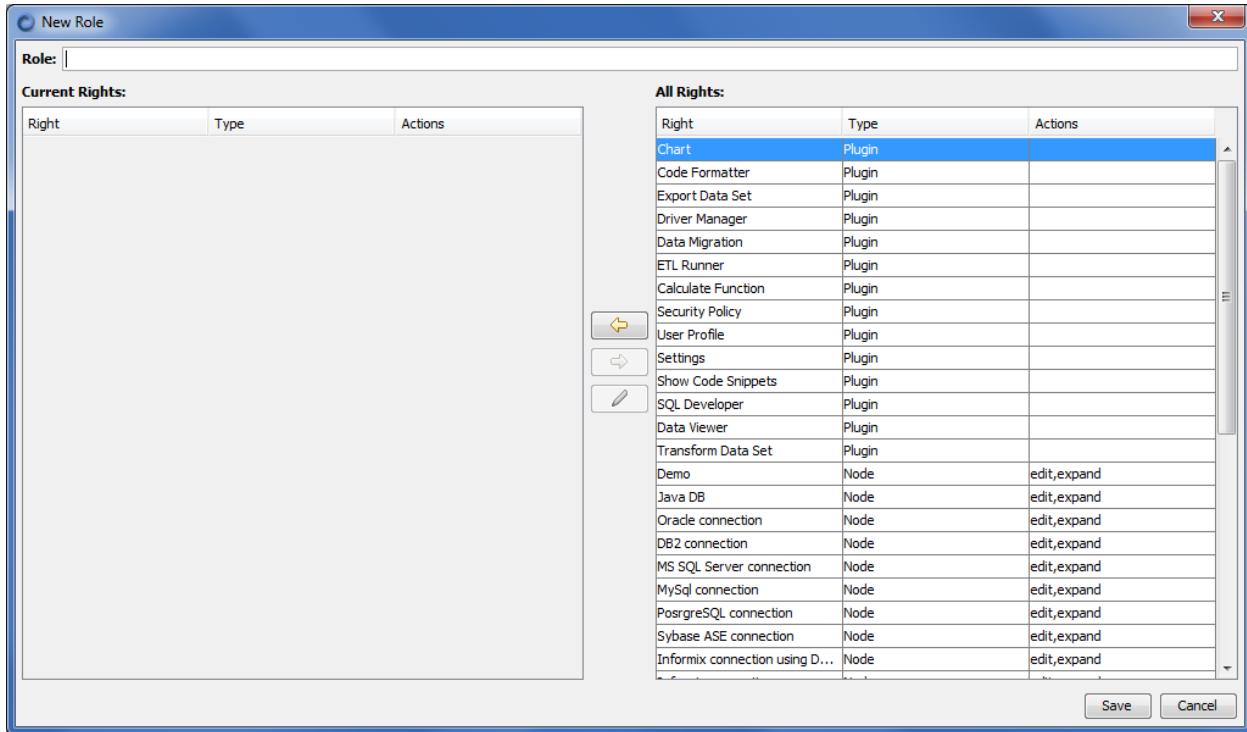


Figure 62: Role Editor

## Fields

Field	Description	Is required
Role	The Role name	Yes
Current rights	The list of rights (permissions) assigned to the role. Use  to assign right to the role and  to un-assign. Use  to edit actions associated with the right (if any exist)	No
All Rights	The list of all Rights	
Save	Saves changes	
Cancel	Closes dialog window	

### ***Login and Logout***

When in the Web or client-server mode the “logout”  button in the Data Explorer toolbar takes user back to the login screen.

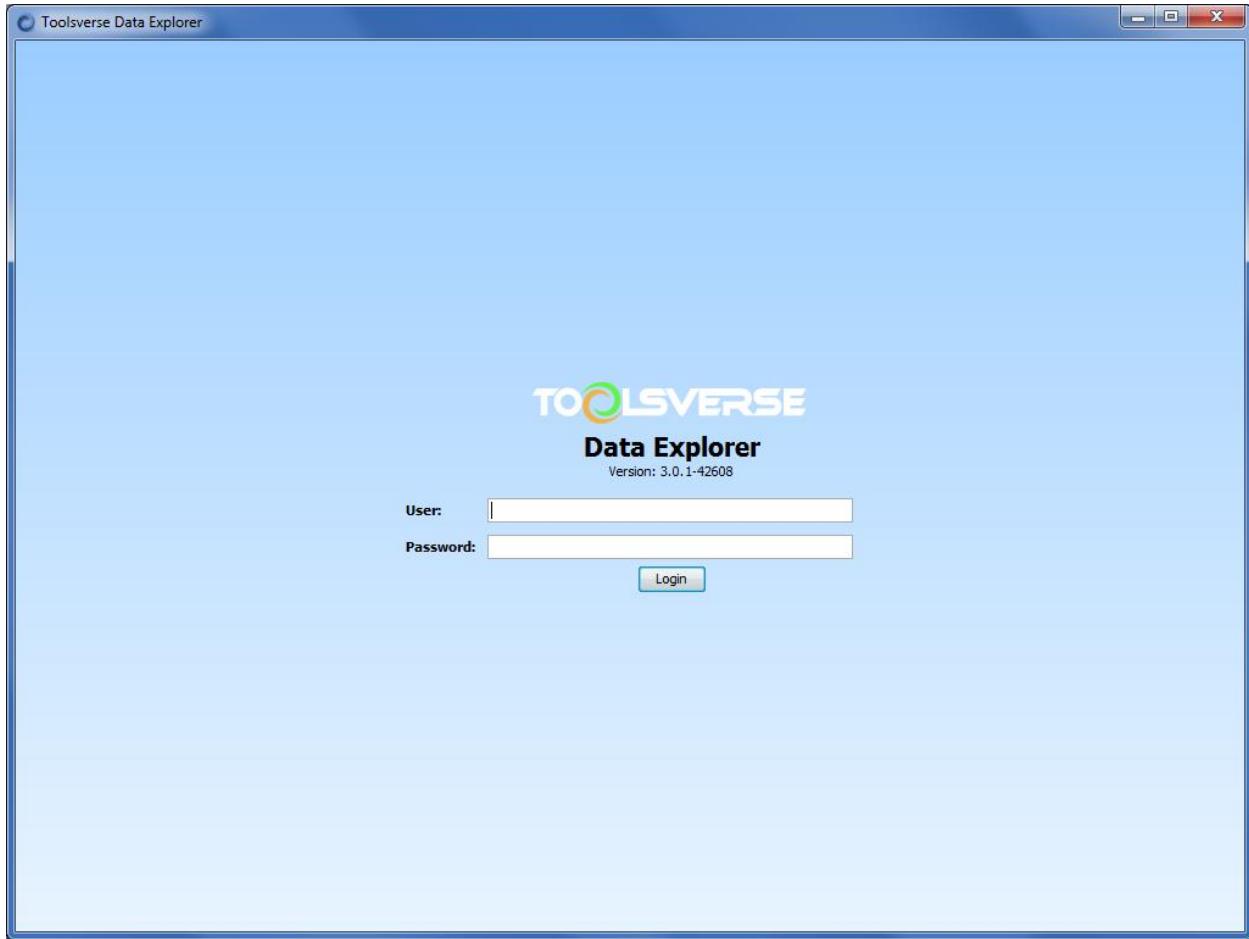
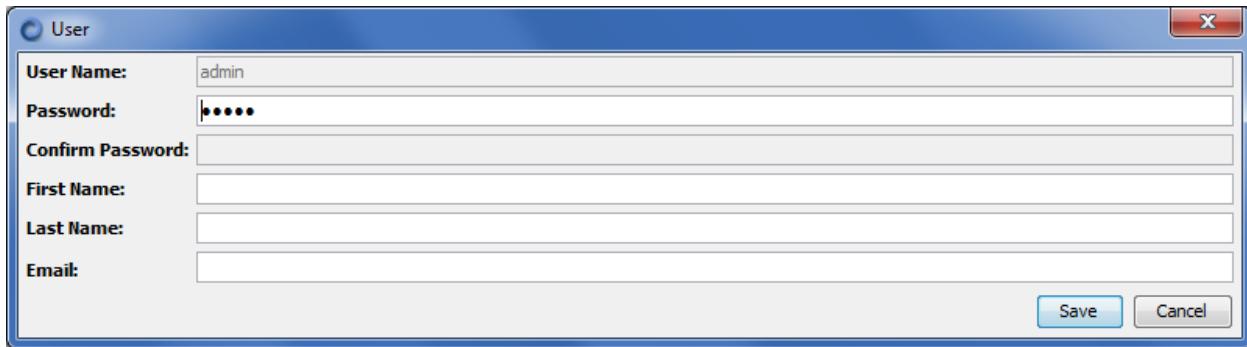


Figure 63: Login Screen

## User Profile (Web and client-server modes only)

In the Web and client-server modes currently logged user can view and change his/her own password and “principals”, such as First Name, Last Name and Email.

To open User Profile dialog window click on the “user profile”  button in the Data Explorer toolbar.



The screenshot shows a Windows-style dialog box titled "User". Inside, there are several input fields:

- User Name:** admin
- Password:** \*\*\*\*\*
- Confirm Password:** (empty)
- First Name:** (empty)
- Last Name:** (empty)
- Email:** (empty)

At the bottom right of the dialog are two buttons: "Save" and "Cancel".

Figure 64: User Profile

### Fields

Field	Description	Is required
User name	The account name. Cannot be changed	Yes
Password	The password	Yes
Confirm Password	The password confirmation. Should be the same as “password”. Field is enabled only when password has changed	No
First Name	User's first name	No
Last Name	User's last name	No
Email	User's email	No
Save	Saves changes	
Cancel	Closes dialog window	

## Common Components

### Data Set Record Viewer

Along with a grid view Data Explorer provides a Form View. It displays one record of the data set at the time but allows back and forward navigation.

**Note:** You can see content of the CLOB and BLOB fields only in the Form view.

When CLOB field is displayed you can choose an appropriate syntax highlighter, for example XML, HTML, etc.

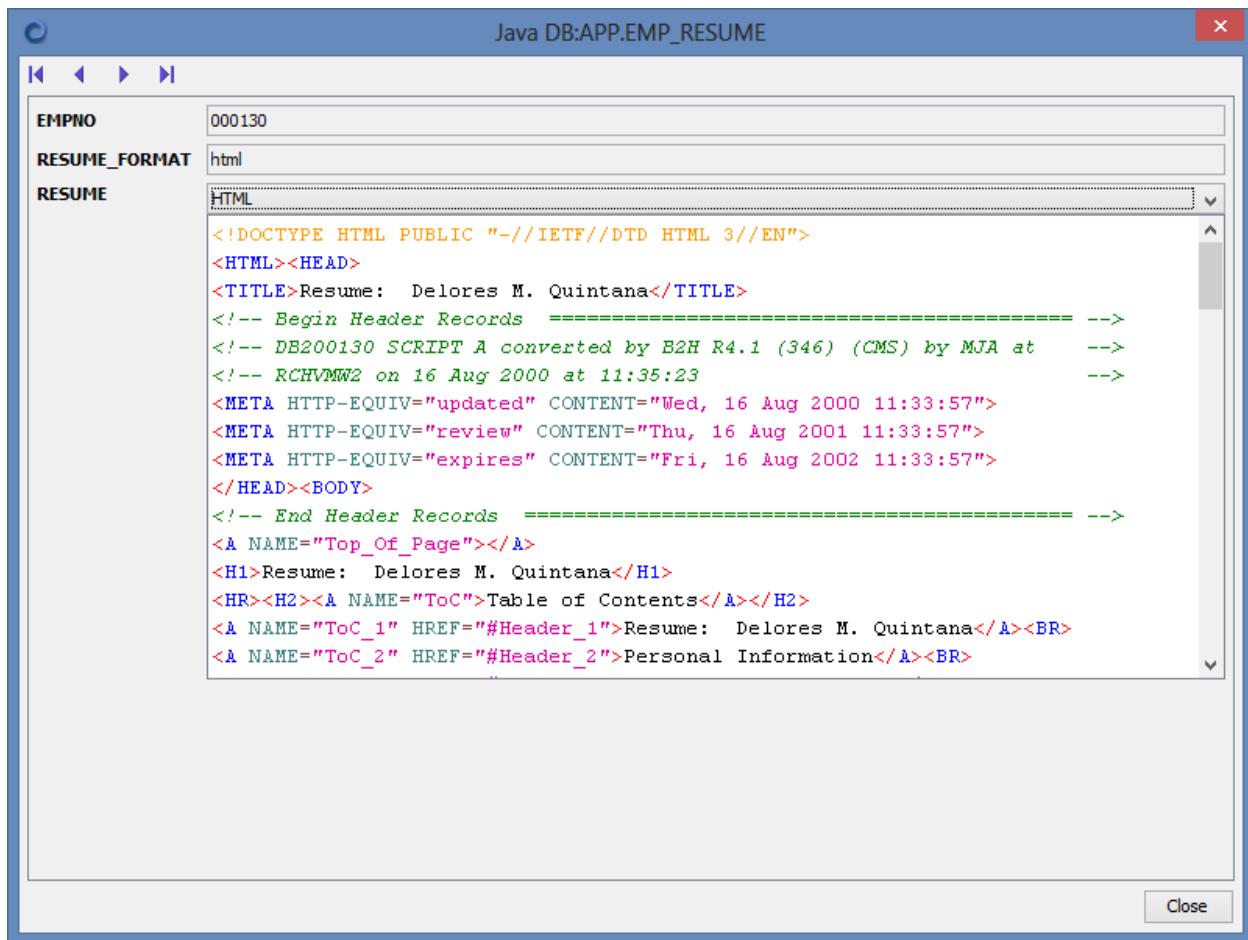
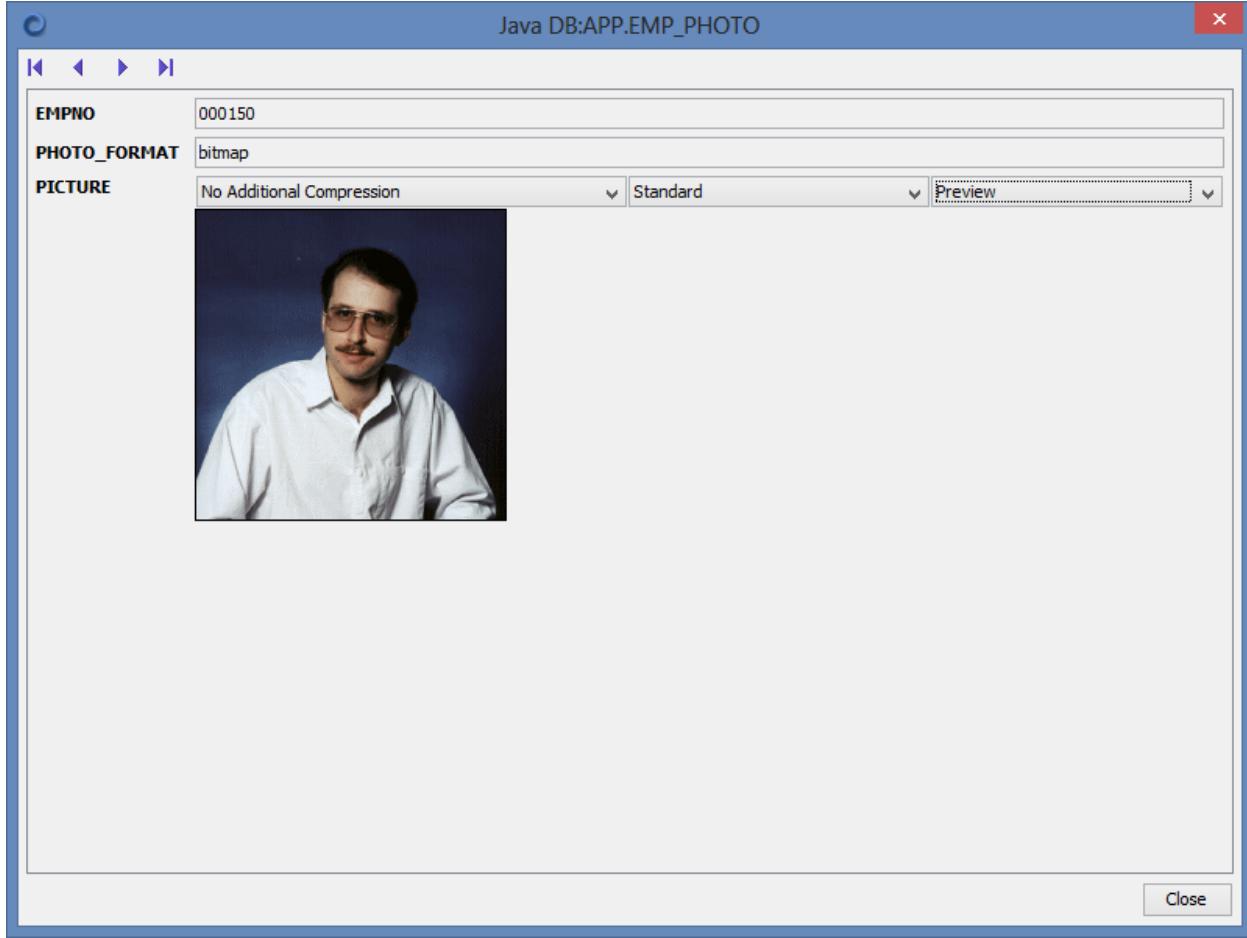


Figure 65: Form View (CLOB)

When BLOB field is displayed the viewer is currently limited to the various image formats. It is possible to view a BLOB field which is additionally compressed using zip or gzip. The images can be scaled up and down compare to the configurable "preview size". To change default preview size use the following access path: Settings->Data Set Record Viewer plugin->Preview Image With (Height).



**Figure 66: Form View (BLOB)**

Use “show data set record” button to open a Form view. You can change a keyboard shortcut used to open Form view using the following access path: Settings->Data Set Record Viewer Plugin->Show Data Set Record. You can change other parameters for the Form View using the following access path: Settings->Data Set Record Viewer Widget and Settings->Data Set Record Viewer Plugin.

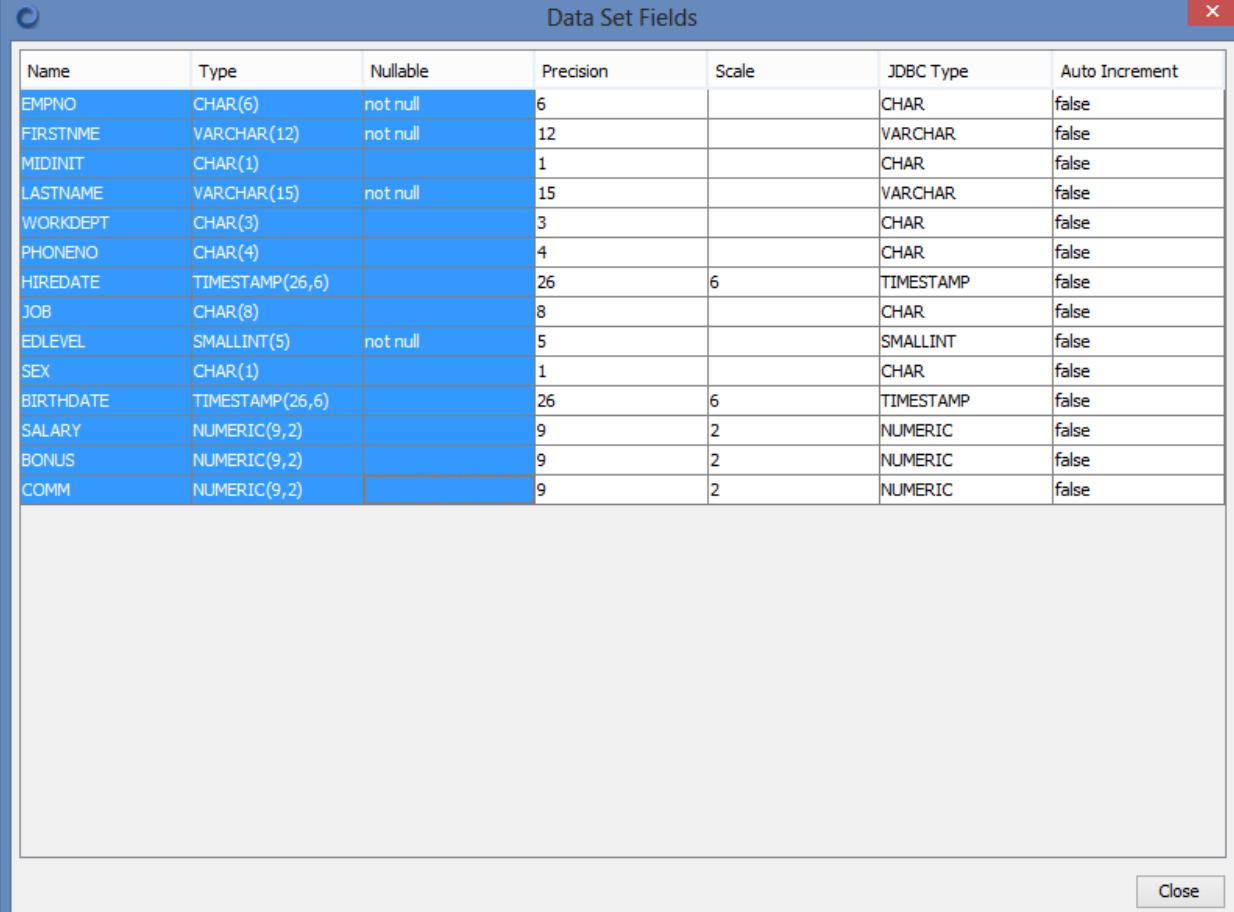
<b>Configuration property</b>	<b>Description</b>
Show Data Set Record	Keyboard shortcut to open record viewer
Select record in Table	When "checked" and you navigate through data set using record viewer navigation buttons the record in the underlying grid is also selected
View BLOB as CLOB	When "checked" the BLOB fields are displayed as CLOBS (text)
View as CLOB if Length is More Than	When "checked" the VARCHAR/CHAR fields with a length bigger than configured value are displayed as CLOBS (larger display area, syntax highlighters, etc)
CLOB/BLOB height	The height of the display area for the CLOB and BLOB fields
Preview Image size	The width and height of the previewed image

## Describe Data Set

You can describe (show detail information about data set fields) a data set by clicking on the  button (if available) in the application toolbar.

You can change a keyboard shortcut used to describe a data set using the following access path:

Settings->Describe Data Set Plugin-> Describe Data Set.



The screenshot shows a Windows-style dialog box titled "Data Set Fields". The table has the following columns: Name, Type, Nullable, Precision, Scale, JDBC Type, and Auto Increment. The data is as follows:

Name	Type	Nullable	Precision	Scale	JDBC Type	Auto Increment
EMPNO	CHAR(6)	not null	6		CHAR	false
FIRSTNAME	VARCHAR(12)	not null	12		VARCHAR	false
MIDINIT	CHAR(1)		1		CHAR	false
LASTNAME	VARCHAR(15)	not null	15		VARCHAR	false
WORKDEPT	CHAR(3)		3		CHAR	false
PHONE NO	CHAR(4)		4		CHAR	false
HIREDATE	TIMESTAMP(26,6)		26	6	TIMESTAMP	false
JOB	CHAR(8)		8		CHAR	false
EDLEVEL	SMALLINT(5)	not null	5		SMALLINT	false
SEX	CHAR(1)		1		CHAR	false
BIRTHDATE	TIMESTAMP(26,6)		26	6	TIMESTAMP	false
SALARY	NUMERIC(9,2)		9	2	NUMERIC	false
BONUS	NUMERIC(9,2)		9	2	NUMERIC	false
COMM	NUMERIC(9,2)		9	2	NUMERIC	false

Figure 67: Describe Data Set

You can select cells and copy them to the clipboard using Ctrl+C (or Command+C on Mac).

## Calculate Function

You can calculate statistical function such as count(), min(), max(), avg(), etc for the entire data set or for the selected rows only. Click on the "function"  button (if available) in the application toolbar to open "Calculate Function" dialog window.

You can change a keyboard shortcut used to open "Calculate Function" dialog window using the following access path: Settings->Data Set Functions Plugin-> Calculate Function.

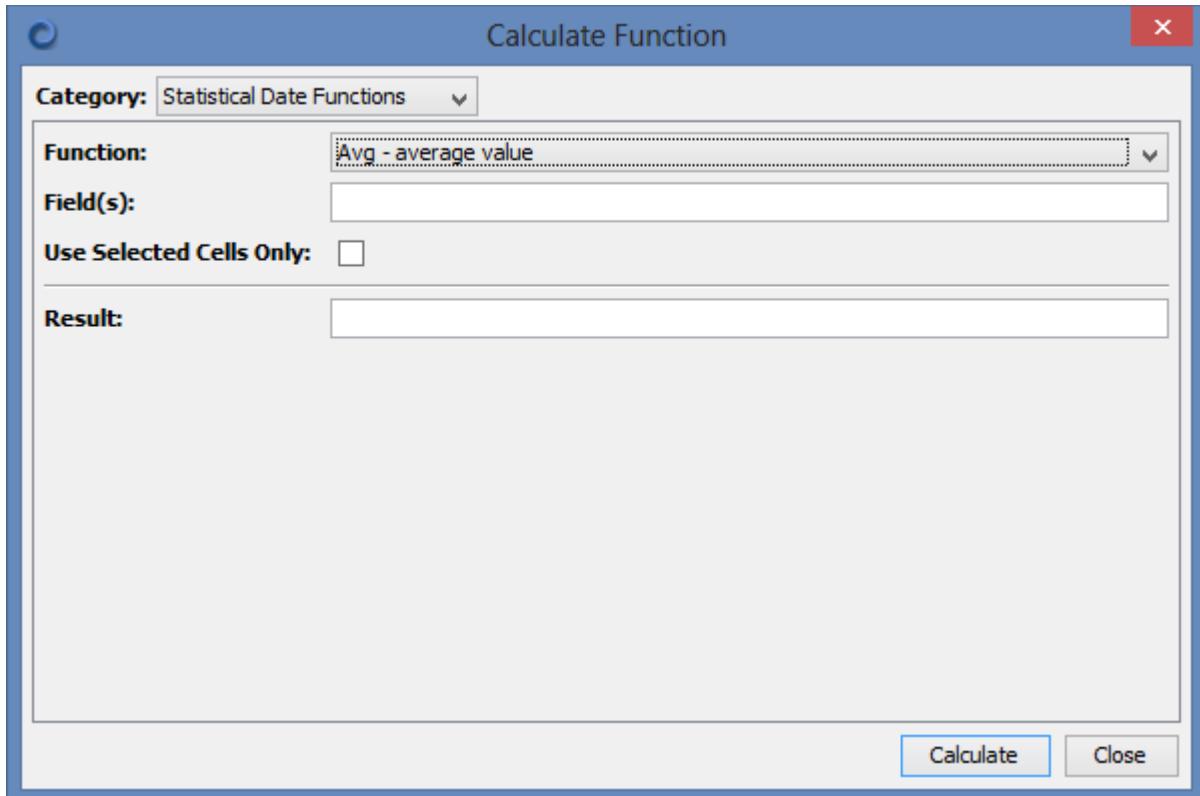


Figure 68: Calculate Function

**Note:** By checking "Use Selected Cells Only" option you will be able to calculate function for the selected rows only. This option is disabled in the Web mode.

The function is calculated for either:

- Fields entered in the **Field(s)** input box. Use ',' as a separator for multiple fields, **filed names are case sensitive**
- Currently selected field
- If "Use Selected Cells Only" option is checked and nothing is entered in the **Field(s)** input box - all selected fields

Use **Category** drop down list to define a category of the functions you want to calculate. Select a particular function from the **Function** drop down and enter parameters. Click on Calculate button to get a result(s).

Available functions and parameters:

Category	Function	Parameters
<b>Statistical Date Functions</b> These functions can be applied to the date fields only. If field is not a date the return value is going to be "Unknown". Fields with a "null" value are ignored.	Avg - average value	<ul style="list-style-type: none"> <li>Field(s) - fields to calculate function for</li> <li>Use Selected Cells Only</li> </ul>
	AvgAge - average age. Age is a difference between now and evaluated date	See above
	Max - maximum value	See above
	Min - minimum value	See above
	Median - median	See above
	Median - median age. Age is a difference between now and evaluated date	See above
	AgeStandardDeviation - age std deviation. Age is a difference between now and evaluated date	See above
<b>Data Set Functions</b> These functions can be applied to the entire data set or to the any type of data	Count - number of rows in the data set	<ul style="list-style-type: none"> <li>Field(s) - fields to calculate function for</li> <li>Use Selected Cells Only</li> </ul>
	Min - minimum value	See above
	Max - maximum value	See above
<b>Statistical Numeric Functions</b> These functions can be applied to the numeric fields only. If field is not a numeric the return value is going to be "Not a Number". Fields with a "null" value are ignored.	Avg - average value	<ul style="list-style-type: none"> <li>Field(s) - fields to calculate function for</li> <li>Use Selected Cells Only</li> <li>Number format - the output format for the number: Example1: #.00 - 123.12 Example2: ## - 123 Example 3: 00.# - 12.123</li> </ul>
	Min - minimum value	See above
	Max - maximum value	See above
	Median - median	See above
	PopulationVariance - population variance	See above
	Product - product	See above
	StandardDeviation - standard deviation	See above
	Sum - sum of all values	See above
	Variance - variance	See above

## Display Data as a Chart

Using Data Explorer it is possible to create charts of various types for the entire data set or for the selected rows only. To display a chart click on the  button (if available) in the application toolbar.

You can change a keyboard shortcut used to create a chart using the following access path: Settings->Chart Plugin-> Display Chart.

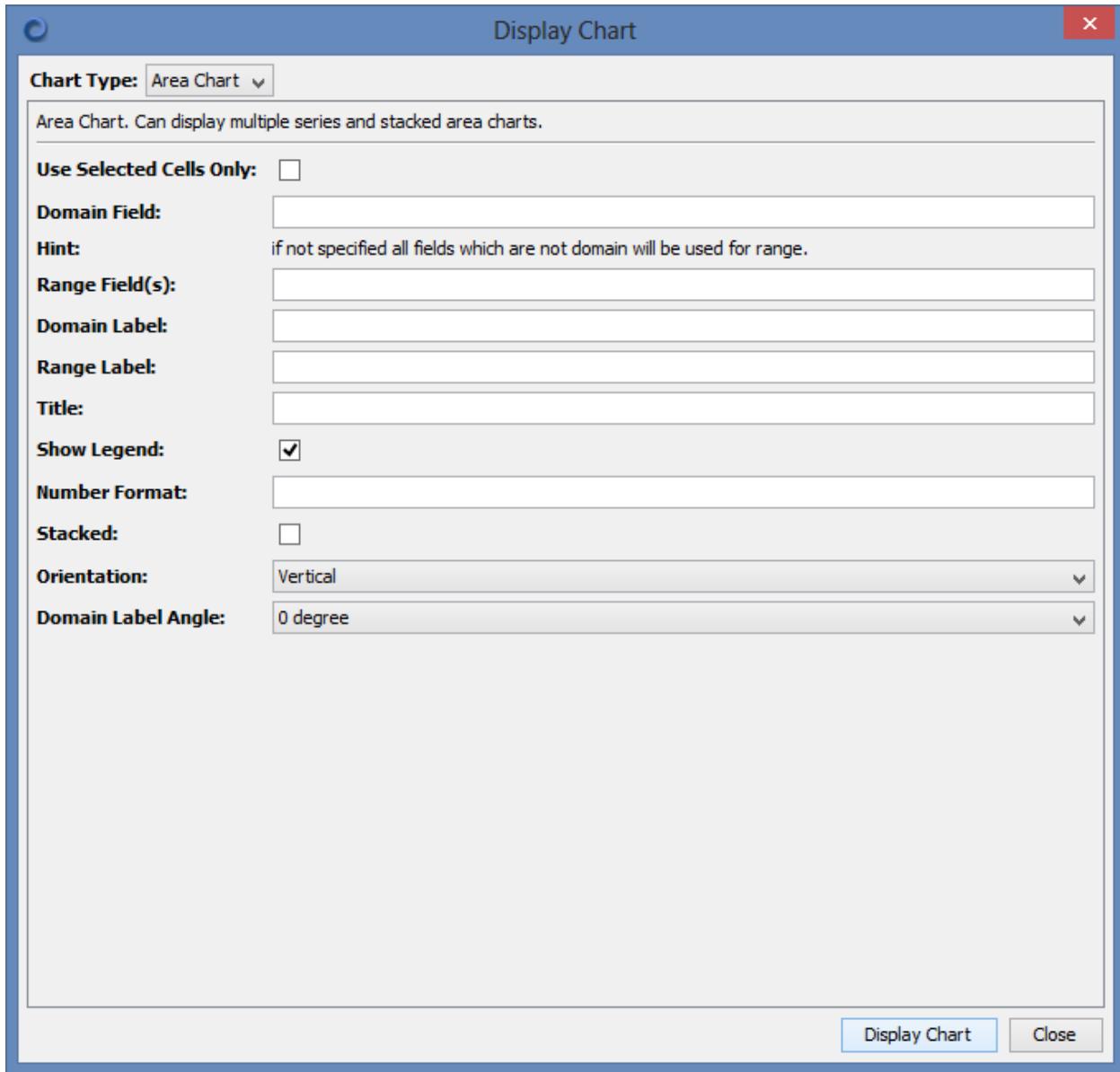


Figure 69: Display Chart

**Note:** By checking "Use Selected Cells Only" option you will be able to create a chart for the selected rows only. This option is disabled in the Web mode.

Use **Chart Type** drop down to define chart type and chart parameters. Available chart types:

- Area Chart (supports multiple series)
- Bar Chart (supports multiple series, supports 3D)
- Line Chart (supports multiple series)
- Pie Chart (does not support multiple series, supports 3D)

The following rules are used to select rows and columns to display on the chart:

- If there are selected rows and **Use Selected Cells Only** option is checked only selected rows will be used to create a chart, otherwise – entire data set
- **Domain Field** is required. It is case sensitive. If there are selected rows and **Use Selected Cells Only** option is checked **Domain Field** must be inside selected block.
- **Range field (s)** is not required and if it is empty all fields which are not **Domain** will be used for range (if chart supports multiple series) or first not **Domain** field will be used for range (if chart does not support multiple series). If there are selected rows and **Use Selected Cells Only** option is checked **Range Field(s)** must inside selected block. Field is case sensitive. Use coma as a separator for multiple fields. Example: salary, bonus
- Other fields are not required

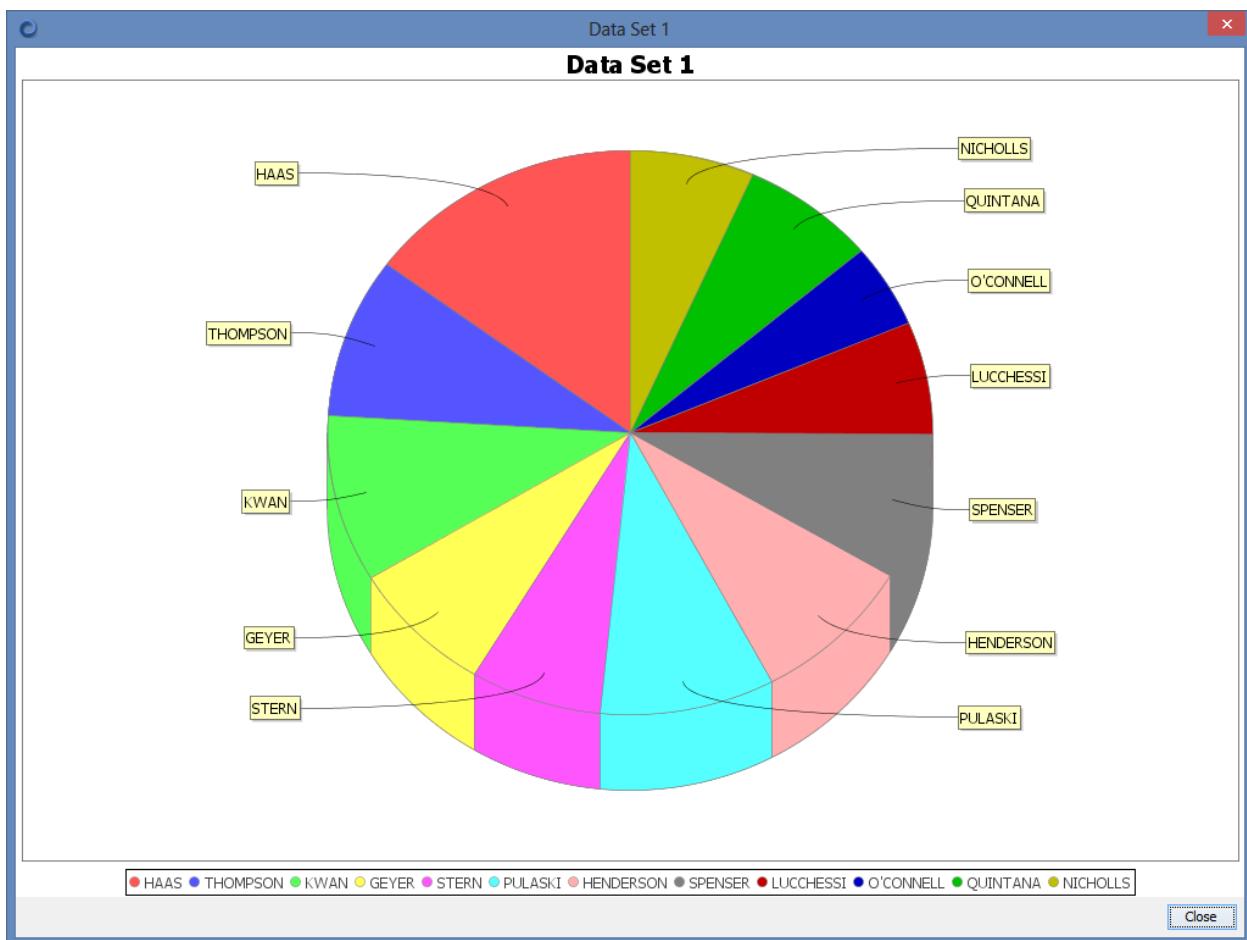


Figure 70: Pie Chart

## Transform Data Set

Data Explorer includes a graphical UI which can be used to perform complex transformations on data sets, such as pivot, de-normalization, filtering, sorting, remove duplicates, set operations, etc.

Transformation can be performed on the entire data set or selected rows only. Some transformations, such as set operations involve two data sets.

You can change a keyboard shortcut used to transform data set using the following access path:  
Settings->Transform Data Set Plugin-> Transform Data Set.

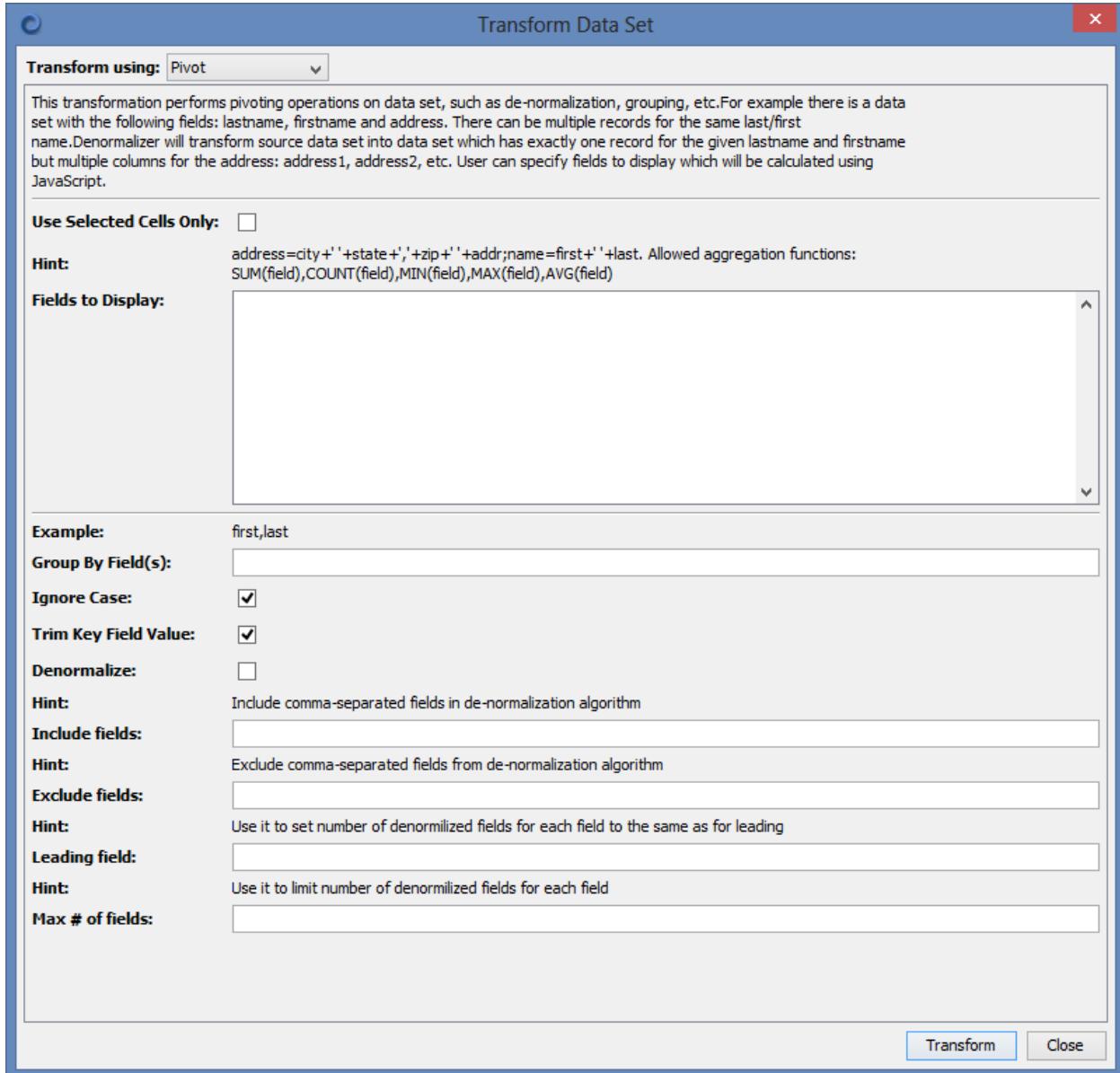


Figure 71: Transform Data Set

**Note:** By checking "Use Selected Cells Only" option you will be able to perform transformation on the selected rows only. This option is disabled in the Web mode.

Use **Transform using** drop down to select a transformation algorithm, enter parameters and click on **Transform** button.

**Note:** Successful transformation does not open new window. If you want to go back to the original data set just click on “refresh”  button in the application toolbar.

### *Filter Transformation*

This is a simple filter, similar to SQL where clause. You can use field names (case sensitive), logical operators: AND, OR, NOT, comparison operators: =,<=,<,> and round brackets: (, ). You cannot use advanced SQL operators such as IN, EXISTS, etc. The condition is evaluated by JavaScript engine so JavaScript functions are also allowed.

**Example:** (FIRSTNME = 'John' and LASTNAME = 'Smith') or SEX.toUpperCase() = 'F'

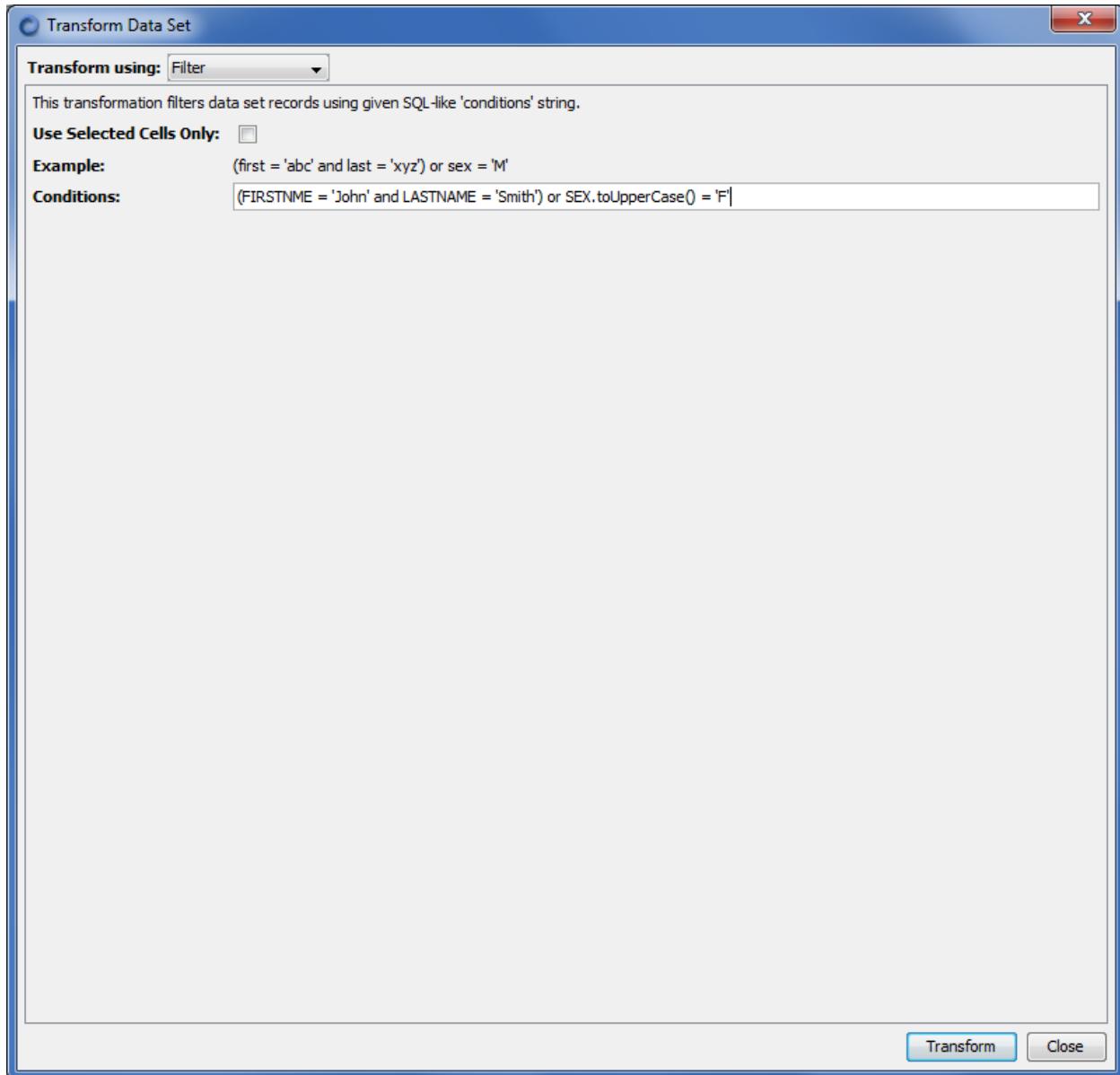


Figure 72: Filter Transformation

### **Join Transformation**

Join transformation is similar to SQL JOIN operation. Basically it combines records and fields from two data sets. INNER and OUTER join are supported. The second data set can be selected by browsing all currently opened by Data Explore data sets and selecting the one to join with. The Transform Data Set dialog window is **not modal** so you can just:

1. Select a tab with a data set to join with
2. Click on the [...] button in the **Join With** field

The **Key Field(s)** is required and case sensitive. Use “,” as a separator if multiple fields should be used to join two data sets. Both data sets must have the same key fields.

**Include** and **Exclude** fields are not required and used to define what fields should be included and excluded in the join of the two data sets. By default all fields from both data sets are included.

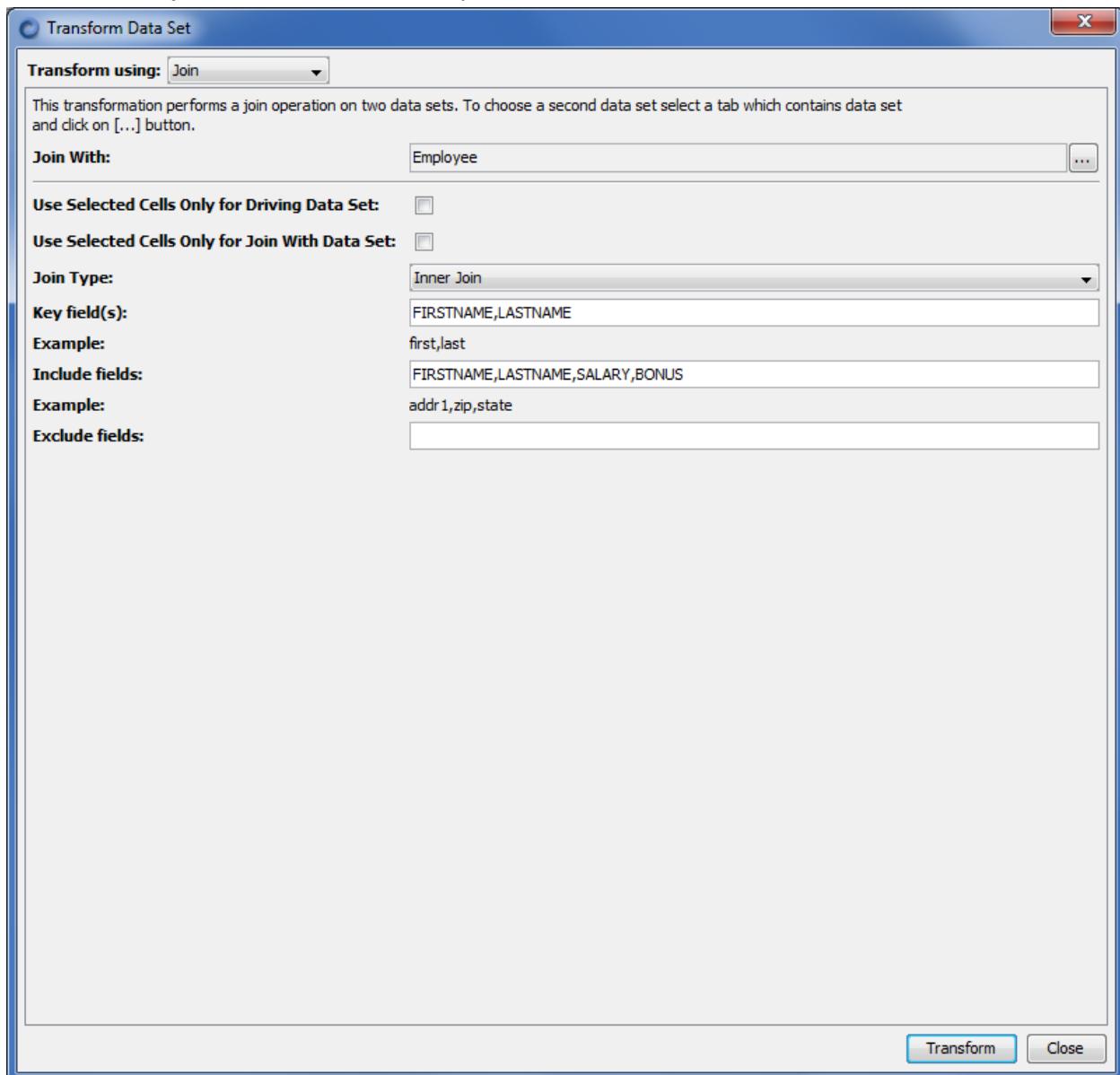


Figure 73: Join Transformation

### **Minus Transformation**

Minus transformation is similar to SQL MINUS operation. Basically it displays records from the first data set which are **not** in the second data set. The second data set can be selected by browsing all currently opened by Data Explore data sets and selecting the one to minus. The Transform Data Set dialog window is **not modal** so you can just:

1. Select a tab with a data set to minus
2. Click on the [...] button in the **Minus** field

The **Key Field(s)** is required and case sensitive. Use “,” as a separator if multiple fields should to be used to minus one data set from another. Both data sets must have the same fields.

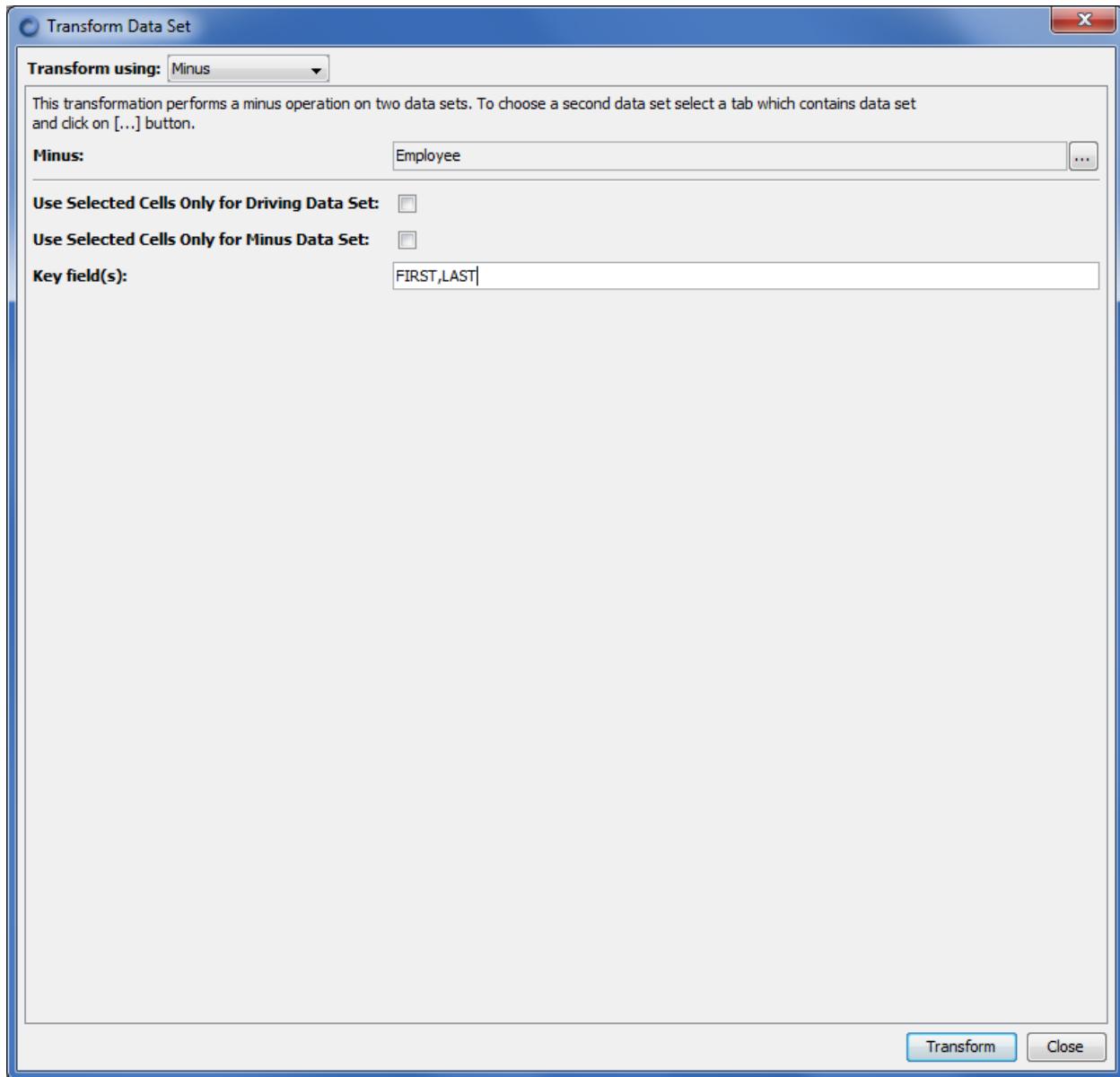


Figure 74: Minus Transformation

### *Order By Transformation*

Order By transformation is similar to SQL order by clause. You can use field names (case sensitive) and **ask** and **desc** suffixes after field names. The default is **ask**. Filed numbers are not allowed.

**Example:** first desc, last, salary desc

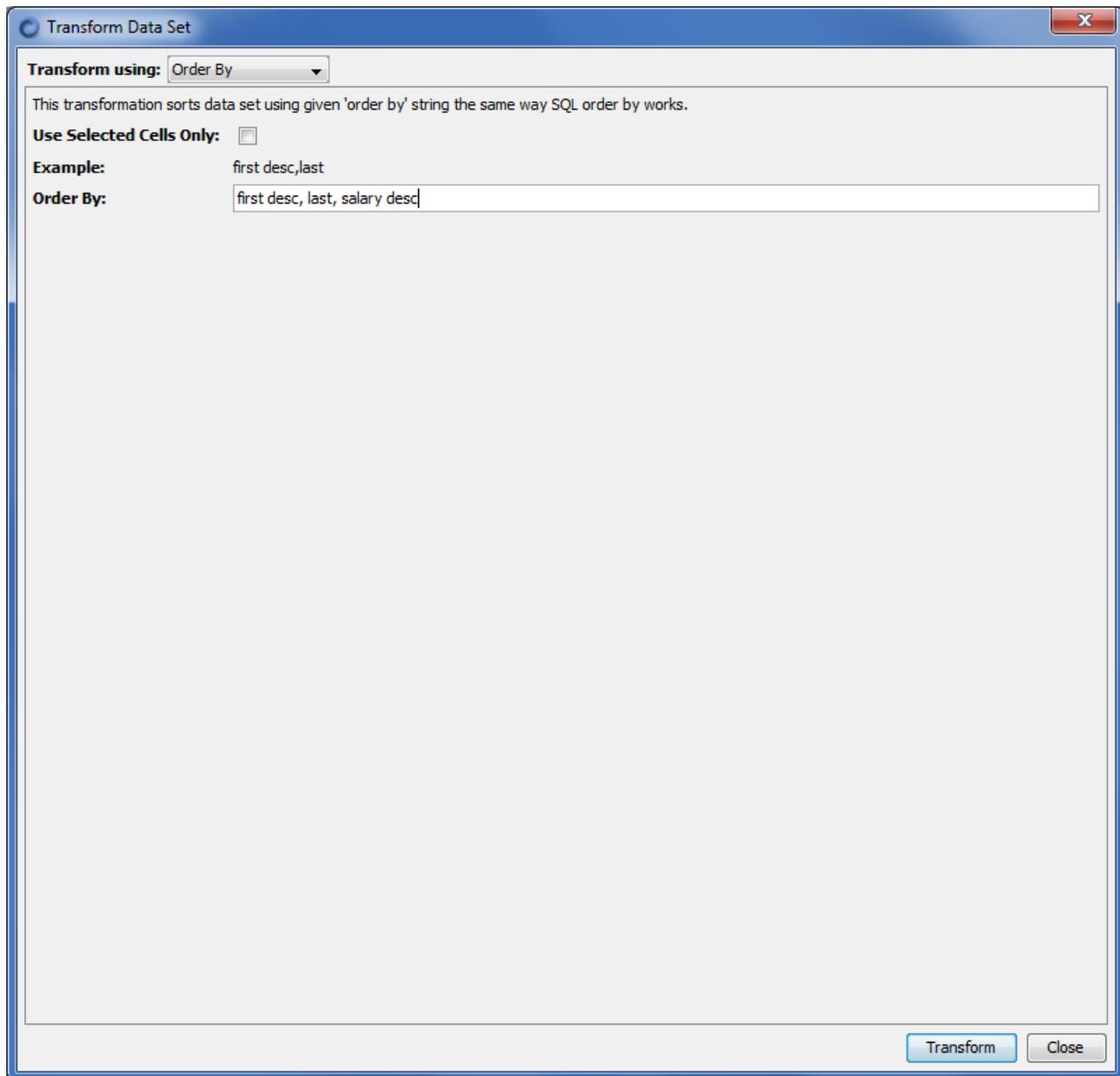


Figure 75: Order By Transformation

### Pivot Transformation

Pivot transformation is similar to SQL GROUP BY. You can “group by” data by key fields and use aggregation function such as SUM, COUNT, MIN, MAX, AVG as well as expressions to calculate field values. The expressions are calculated using JavaScript engine.

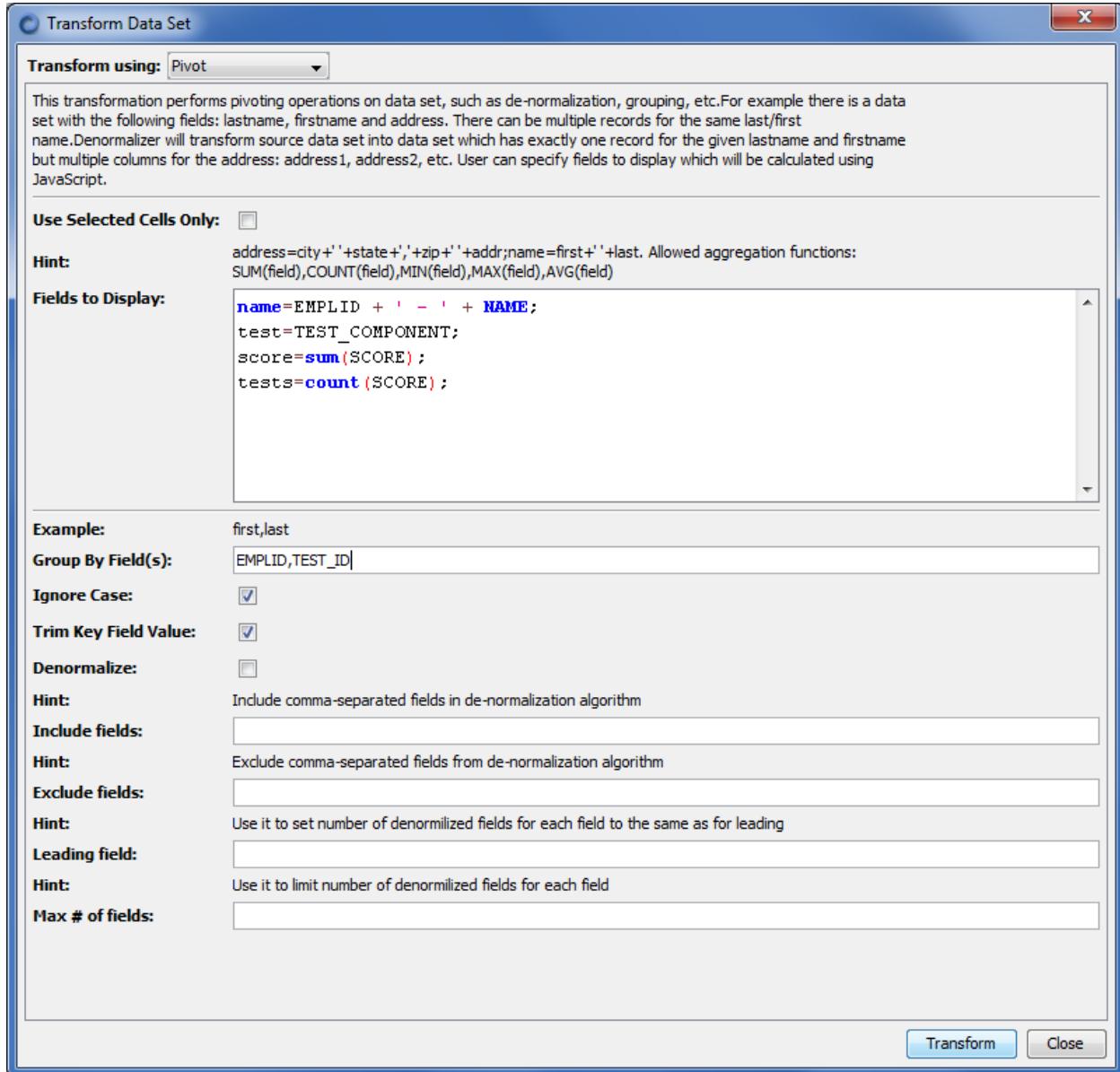


Figure 76: Pivot Transformation

**Example:**

EMPID	NAME	TEST_ID	TEST_COMPONENT	SCORE
3806882	Steve Jobs	GRE	QUANS	162.00
3806882	Steve Jobs	GRE	ANLY	3.00
3806882	Steve Jobs	GRE	VERBS	138.00
3806882	Steve Jobs	TOEFL	COMPI	86.00
3806882	Steve Jobs	TOEFL	READI	24.00
3806882	Steve Jobs	TOEFL	LISTN	21.00
3806882	Steve Jobs	TOEFL	WRITI	24.00
3806882	Steve Jobs	TOEFL	SPEAK	17.00

**Fields to Display:**

```
name=EMPLID + ' - ' + NAME;
test=TEST_COMPONENT;
score=sum(SCORE);
tests=count(SCORE);
```

**Group By Fields:**

EMPLID,TEST\_ID

**Result:**

name	test	score	tests
3806882 - Steve Jobs	GRE	303	3
3806882 - Steve Jobs	TOEFL	172	5

If **Ignore Case** option is selected Data Explorer will ignore case of field's **value** when checking for duplicates. Also if **Trim Key Field Value** option is selected the leading and trailing spaces in the field's **value** will be ignored.

Other fields are ignored.

### **De-normalize Transformation**

The de-normalize transformation is a sub set of the Pivot transformation and uses the same UI (you need to select a Pivot Transformation to perform a de-normalization).

**Example for the de-normalization:** there is a data set with the following fields: lastname, firstname and address. There can be multiple records for the same last/first name. De-normalizer will transform source data set into data set which has exactly one record for the given lastname and firstname but multiple columns for the address: address1, address2, etc.

You must check a **Denormalize** option. **Group By Field(s)** is required.

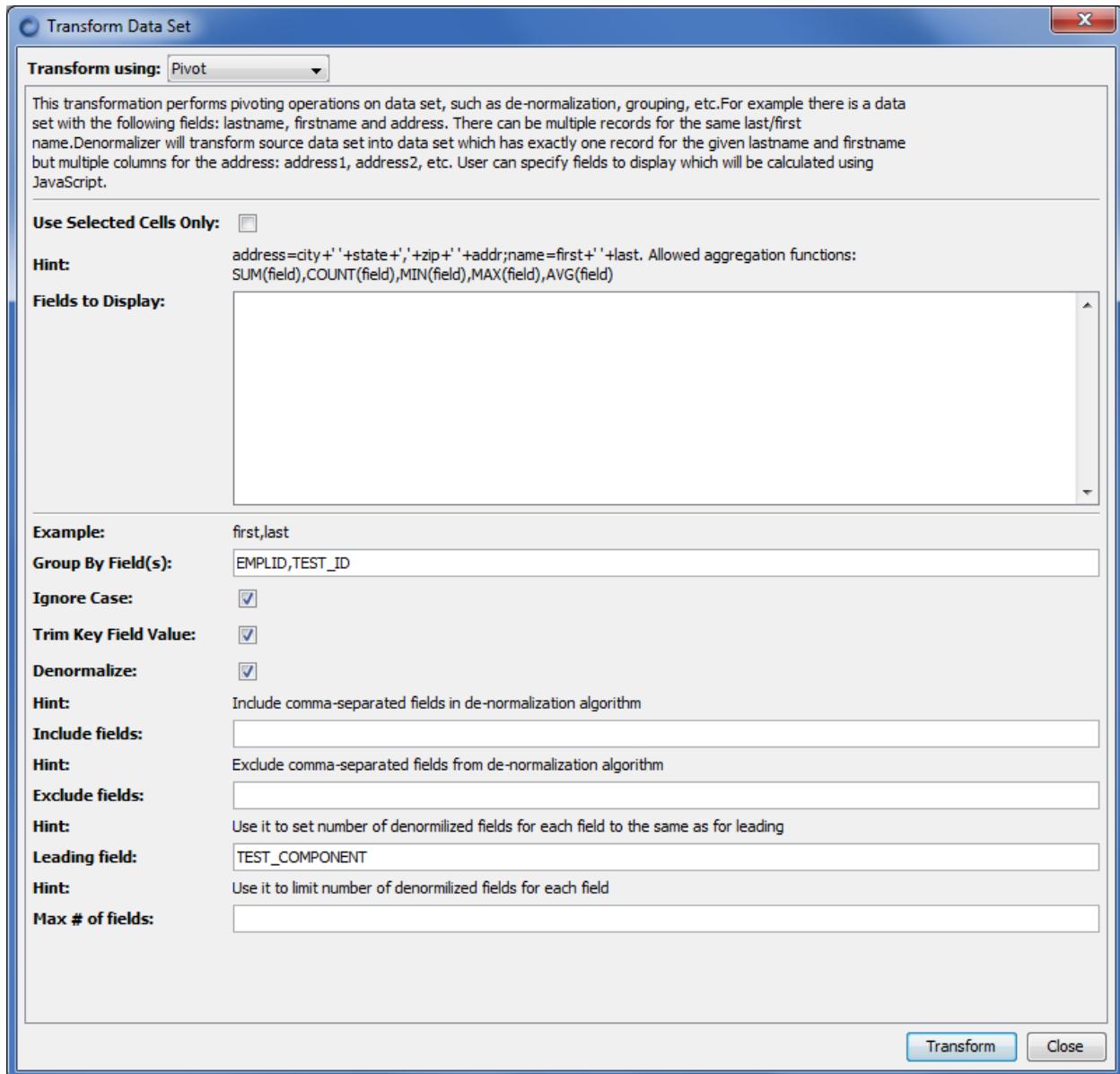


Figure 77: De-normalize Transformation

Other fields:

If **Ignore Case** option is selected Data Explorer will ignore case of field's **value** when checking for duplicates. Also if **Trim Key Field Value** option is selected the leading and trailing spaces in the field's **value** will be ignored.

**Include fields** – if not empty (you can have multiple comma delimited, case sensitive field names here) fields in this list will be included in de-normalization. The others will not.

**Exclude fields** – if not empty (you can have multiple comma delimited, case sensitive field names here) fields in this list will be excluded from de-normalization. The others will not.

**Leading field** – if not empty (you can have single, case sensitive field name here) this field will be used as a pattern for all others. For example if a leading field "abc" is repeated 4 times, all others will be repeated 4 times as well.

**Max # of fields:** the maximum number of demoralized fields for each sub-field. For example if there are fields abc, xyz, and yyy which can be de-normalized the maximum number of abc, xyz and yyy will be less or equal to entered value.

### **Remove Duplicates Transformation**

Remove Duplicates transformation simply removes duplicated rows. Basically only one row with the same values of the key field's remains after transformation. Example:

#### **Before**

<b>Id</b>	<b>Value</b>
100	abc
200	xyz
300	mnm
100	xxx
100	yyy
300	aaa

#### **After**

<b>Id</b>	<b>Value</b>
100	abc
200	xyz
300	mnm

The **Key Field(s)** is required and case sensitive. Use “,” as a separator for multiple fields. If **Ignore Case** option is selected Data Explorer will ignore case of field's **value** when checking for duplicates. Also if **Trim Key Field Value** option is selected the leading and trailing spaces in the field's **value** will be ignored.

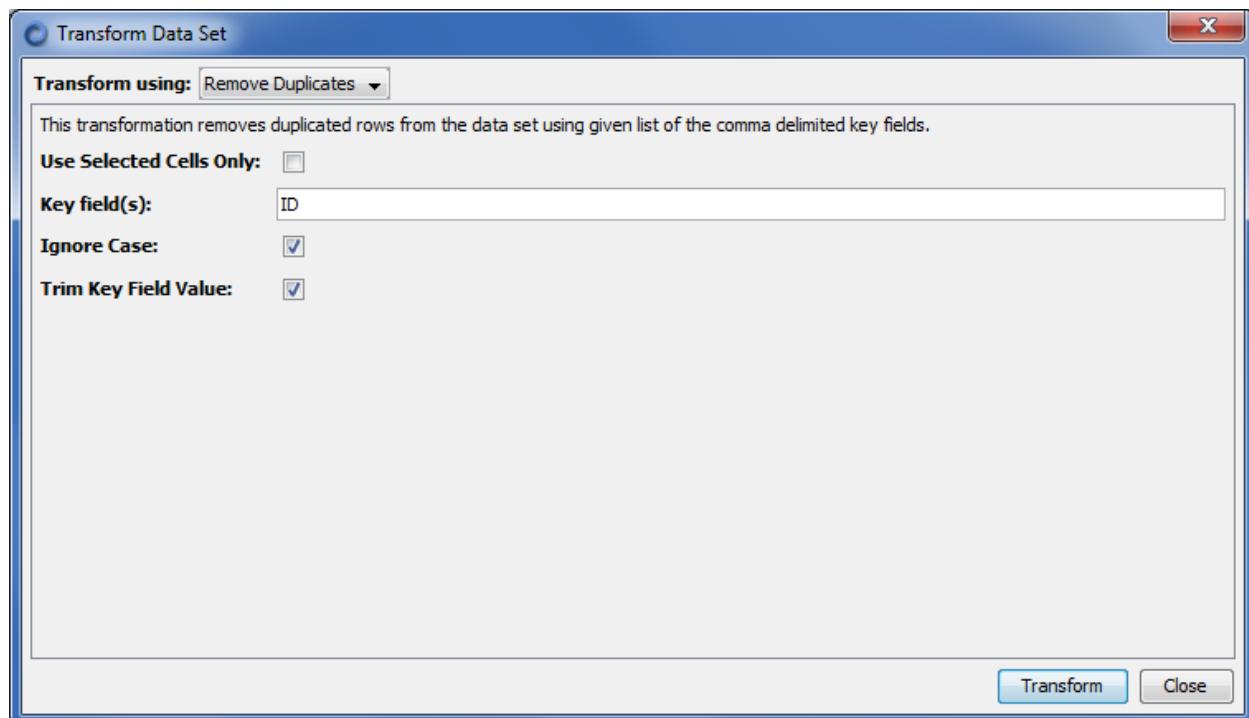


Figure 78: Remove Duplicates Transformation

### Transpose Transformation

Transpose transformation transposes data set (swaps rows and columns). Example:

#### Before

Id	Value
100	abc
200	xyz
300	mnm

#### After

column_0	column_1	column_2
100	200	300
abc	xyz	mnm

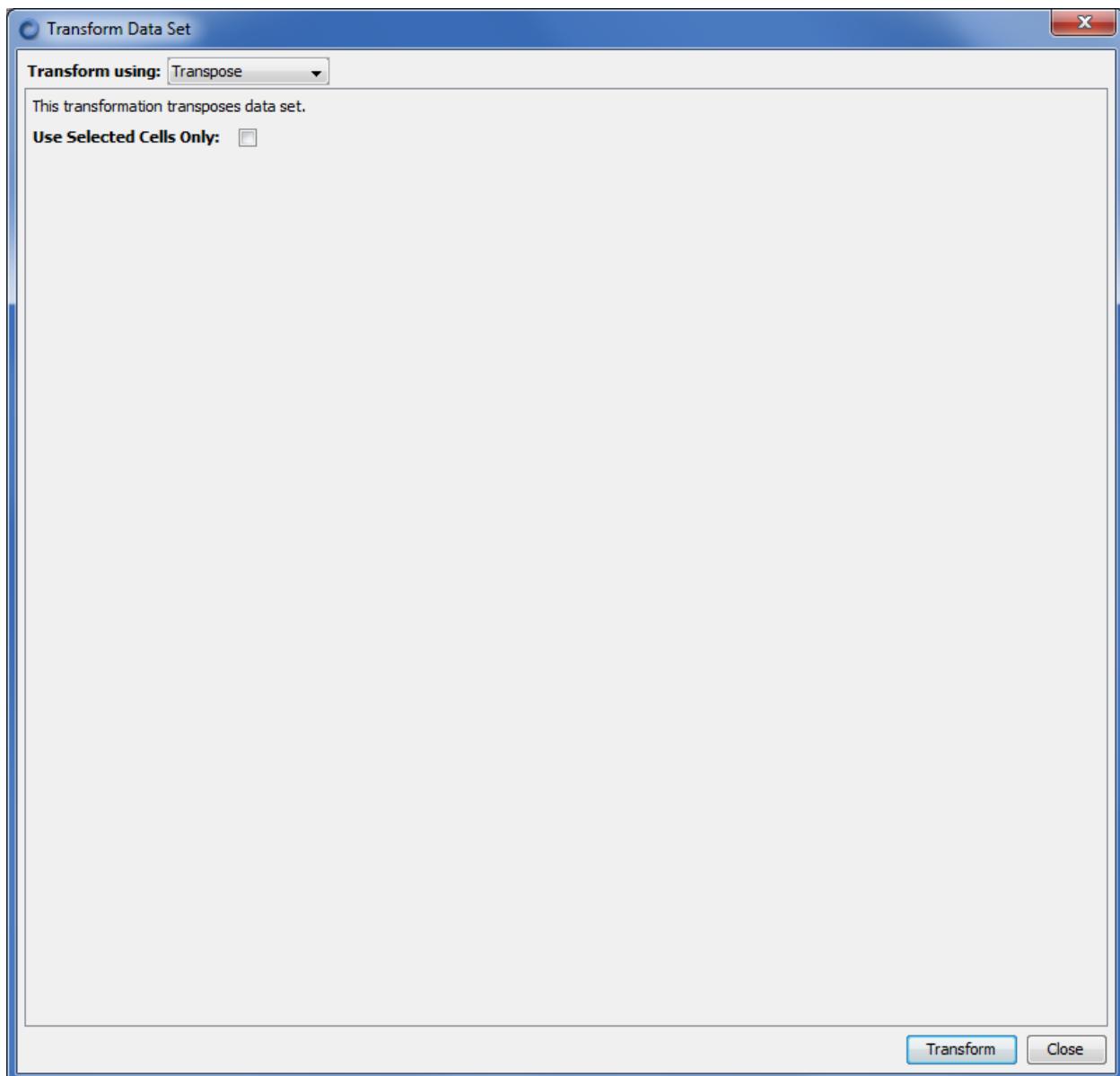


Figure 79: Transpose Transformation

### **Union Transformation**

Union transformation is similar to SQL UNION operation. Basically it adds records from one data set to another. UNION ALL and UNION are supported. The second data set can be selected by browsing all currently opened by Data Explore data sets and selecting the one to union with. The Transform Data Set dialog window is **not modal** so you can just:

1. Select a tab with a data set to join with
2. Click on the [...] button in the **Union With** field

The **Key Field(s)** is required for the UNION operation (not required for UNION ALL) and case sensitive. Use “,” as a separator if multiple fields should be used to union two data sets.

**Note:** Both data sets must have the same fields.

**Include** and **Exclude** fields are not required and used to define what fields should be included and excluded in the union of the two data sets. By default all fields are included.

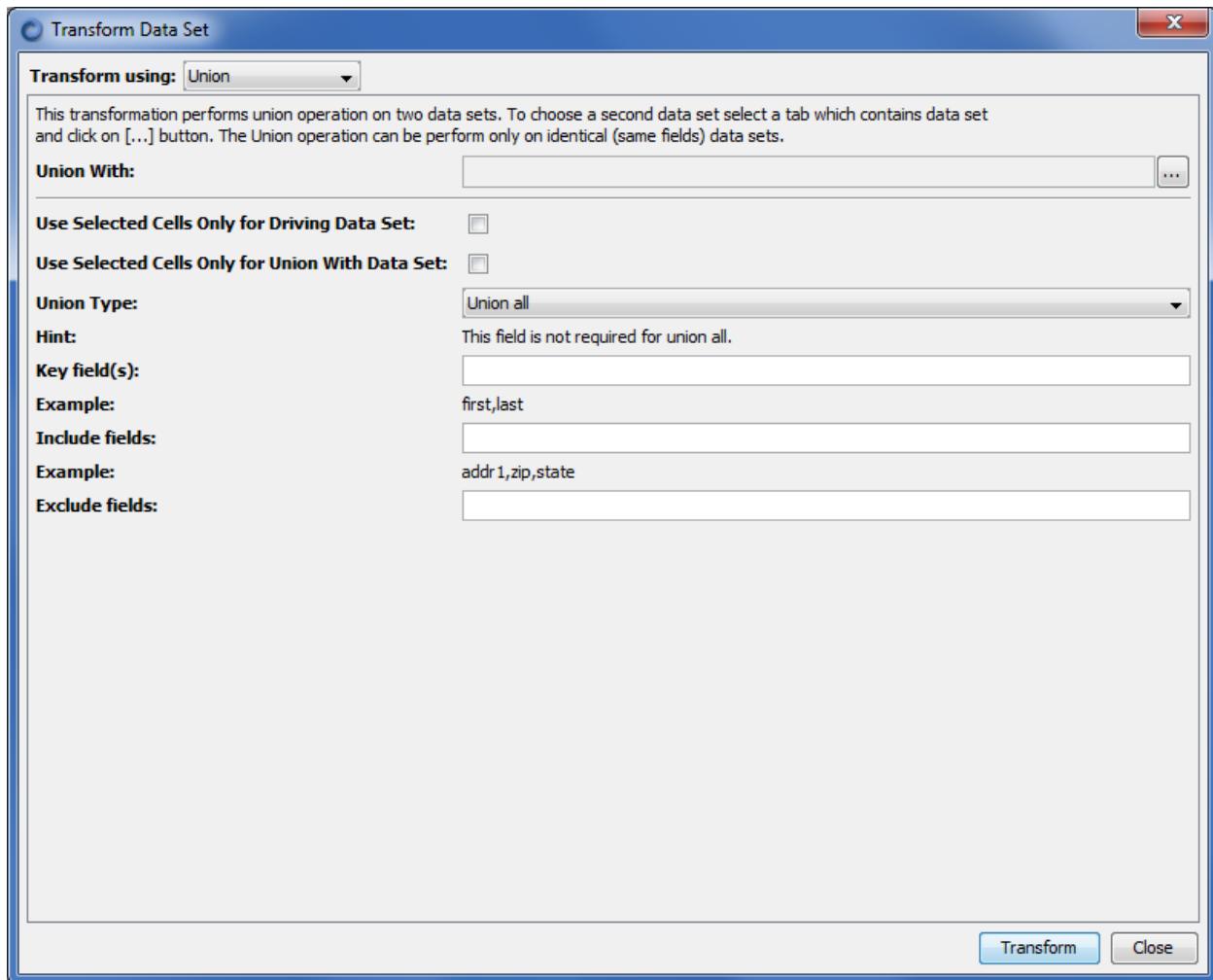


Figure 80: Union Transformation

## Set Grid Defaults

To change grid parameters such as: maximum number of rows displayed (all modes) or maximum number of rows and columns displayed per page (Web mode only), click on the set grid defaults  button.

You can change a keyboard shortcut used to open Grid Default dialog window using the following access path: Settings->SQL Developer App-> Set Grid Defaults. You can also change constraints applied to the Grid Defaults fields using the following access path: Settings->Data Set Viewer Widget

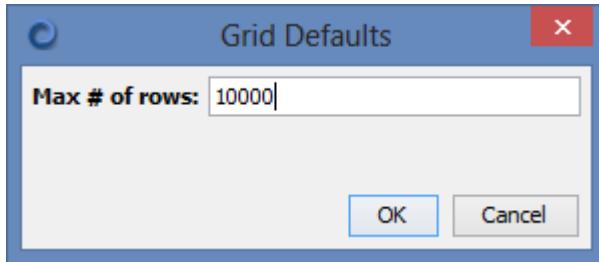


Figure 81: Grid Defaults

## Search in the Data Set

You can search in the current data set by clicking on the “search in the data set”  button. Search is global, all rows and columns are included.

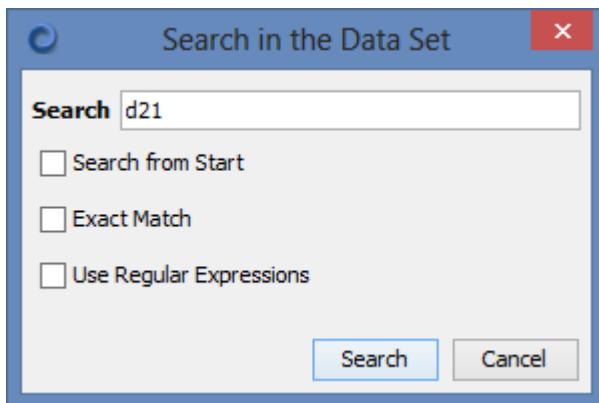


Figure 82: Search in the Data Set

## View and Manage Code Snippets

Code snippets are useful fragments of the code. You can view them, edit, delete, copy to the editor, etc.

Click on the code snippets  button in the toolbar to open Code Snippets dialog window.

You can change a keyboard shortcut used to open "Code Snippets" dialog window using the following access path: Settings->Code Snippets Plugin->Show Code Snippets.

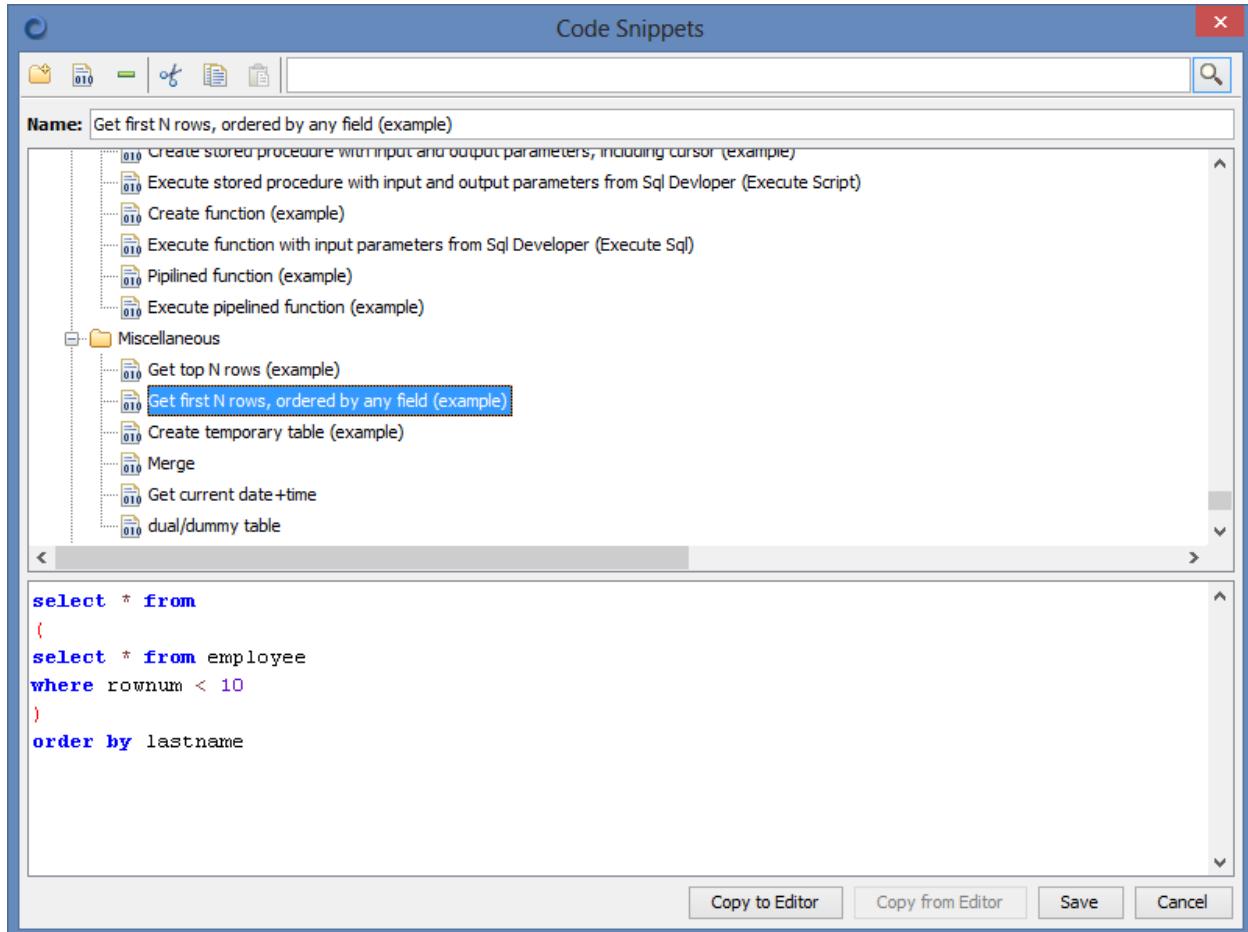


Figure 83: Code Snippets

**Note:** If you are accessing code snippets from the SQL Developer using data Explorer Professional or ETL Edition the database specific code snippets will be displayed.

Code Snippets functions:

Icon	Function	Description
	Add folder	Adds the folder
	Add snippet	Adds the snippet
	Delete folder or snippet	Deletes the folder or snippet

	Cut folder or snippet	Cuts folder or snippet
	Copy folder or snippet	Copies folder or snippet
	Paste folder or snippet	Paste folder or snippet
	Copy to Editor	<p>Copies snippet code to the editor.</p> <p><b>Note:</b> Copy to Editor replaces all code in the editor with a snippet. To insert a code into the specific position in the editor use text copy and paste</p>
	Copy from Editor	<p>Copies code from the editor to the current snippet.</p> <p><b>Note:</b> If there is a selected text in the editor it will be copied instead of entire buffer.</p>
	Save	Saves all changes

## Grid

Grid is a common component used to display multiple rows of the data set.

CONFIG_PROPERTY_NUM	NAME	DESCRIPTION	DEFAULT_VALUE
1	name1	description1	123
2	name2	description2	abc
3	name3	description3	12.23
4	name4	description4	01/01/2012
5	name5	description5	test
6	name6	description6	
7	name7	description7	true

Figure 84: Grid

You can select cells in the grid using default method for the host operation system and copy them to the system clipboard. Use Ctrl+C in Windows and Unix/Linux and Command+C in OS X. Copy to clipboard is also available from the popup menu (right mouse click).

**Note:** This function is not available per-se in the web mode but you can select a text and use browser copy command to archive the same goal.

## Data Set Export

Using data Set Export plug-in you can export data set to the following file formats:

- Excel (\*.xls)
- Excel (\*.xlsx)
- Delimited text
- Fixed Length text
- XML dataset
- any XML using XSL transformation
- SQL (insert statements)

**Note:** By checking "Use Selected Cells Only" you will be able to export selected rows only. This option is disabled in the Web mode.

To export data set click on the “export data set”  button in the toolbar. You can change a keyboard shortcut used to open "Export Data Set" dialog window using the following access path: Settings-> Export Data Set Plugin-> Export Data Set.

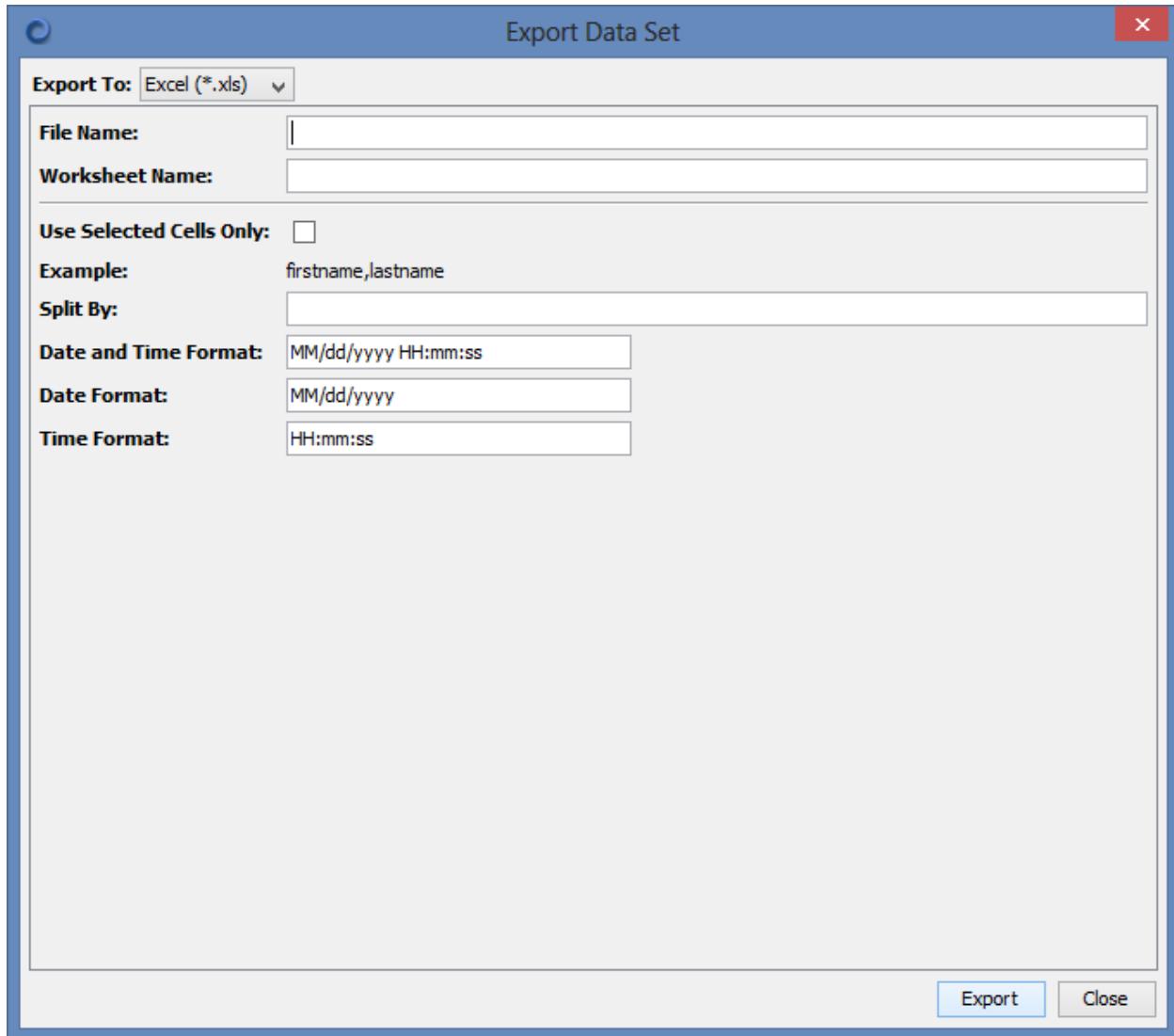


Figure 85: Export Data Set

**Note:** If you specify one or multiple (comma-delimited) key fields in the **Split By** input box the system will automatically group data by key fields and create multiple output files: one for each group. For example you can have an Excel spreadsheet “report” with a data for 12 months. The spreadsheet has a field Month (January, February, etc). You can enter Month as a key field in the **Split By** input box and system will create a separate file for each month: report\_January.txt, report\_February.txt, etc.

Use **Export To** drop down list to define format and parameters.

Available formats and parameters:

Format	Parameter	Example	Default value
<b>Excel (*.xls)</b>	File Name	abc.xls	None. Required field
	Worksheet Name	test	None. Required field
	Date and Time format	MM/dd/yyyy HH:mm	System defined
	Date format	MM/dd/yyyy	System defined
	Time format	HH:mm	System defined
<b>Excel (*.xlsx)</b>	File Name	abc.xls	None. Required field
	Worksheet Name	test	None. Required field
	Date and Time format	MM/dd/yyyy HH:mm	System defined
	Date format	MM/dd/yyyy	System defined
	Time format	HH:mm	System defined
<b>SQL</b>	File name	abc.sql	None. Required field
	Table name	test	If not entered the file name without extension will be used
<b>Text</b>	File name	abc.txt	None. Required field
	Delimiter	;	
	Store Metadata	false	true
	Use first row for data	false	true
	Fields	10;12;15;8	None. If used the fixed length file will be created. Use the same separator as defined by

			<b>Delimiter field</b>
	Date and Time format	MM/dd/yyyy HH:mm	System defined
	Date format	MM/dd/yyyy	System defined
	Time format	HH:mm	System defined
<b>XML</b>	File name	abc.xml	None. Required field
	Xsl file name	xyz.xslt	None. If used the xsl transformation will be performed
	Date and Time format	MM/dd/yyyy HH:mm	System defined
	Date format	MM/dd/yyyy	System defined
	Time format	HH:mm	System defined

Please note that if File Name field does not include path the output file will be created in the local DATA folder. Please see definition of the local DATA folder [here](#).

You can use variables {app.root.data} and {app. data} as a path of the file name. For example: {app.root.data}/abc.txt. Please see definition of these variables [here](#).

## Code Formatter

To format a code in the editor, click on the “format code”  button. The particular language must be supported by code formatter. The generic SQL code formatter is trying to do its best but it does not recognize all dialects of the SQL.

**Note:** Formatter formats entire editor’s buffer. Formatting of the selected text is not supported yet. Use undo function if you don’t like results.

You can change some parameters of the formatter for the particular language using the following access path: Settings->XXX Code Formatter. For example: Settings->SQL Code Formatter.

## Search and Replace

The text editor supports Search and Replace. Click on the “search and replace”  button to open “Search/Replace” dialog window.

**Note:** search using regular expressions is not currently supported in the Web mode.

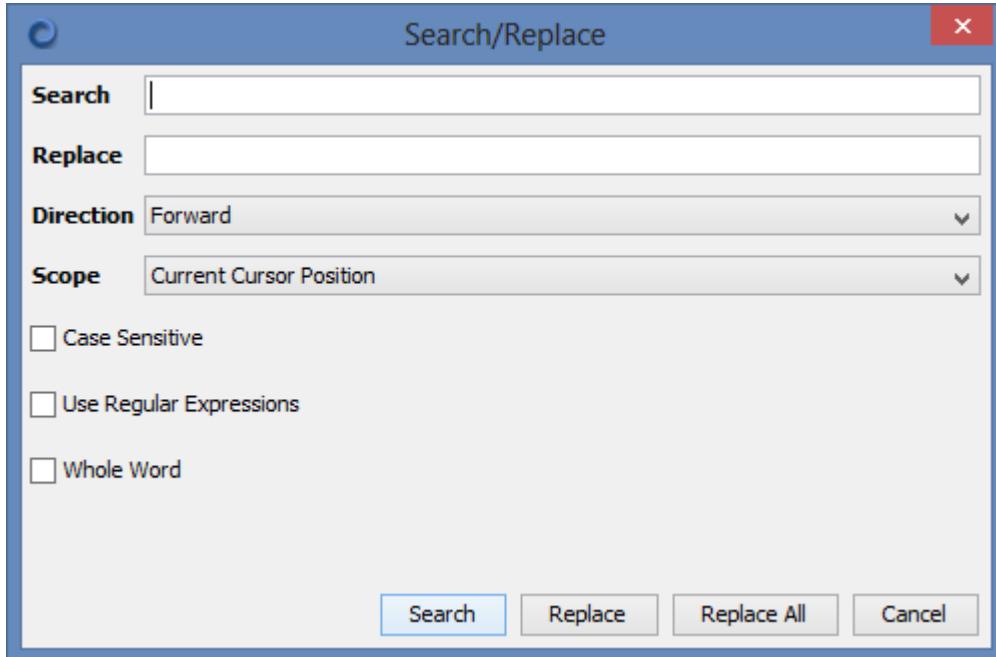


Figure 86: Search and Replace

You can change keyboard shortcut used to open “Search/Replace” dialog window using the following access path: Settings->Search/Replace Plugin->Search and Replace.

## Go to the Line

To execute Go to the Line function click on the “go to the line”  button.

You can change keyboard shortcut used to open “Go to the Line” dialog window using the following access path: Settings-> Go to the Line Plugin-> Go to the Line.

You can enable displaying line numbers in the editor using the following access path: Settings->Code Editor Widget->Show Line Numbers. You will need to re-open currently opened editor windows to see an effect.

## Appendix 1. All Keyboard shortcuts

Keyboard shortcuts are customizable. Use the following access path to change keyboard shortcuts:  
Settings->Plugin name->Function.

Default keyboard shortcuts:

Icon	Component	Function	Windows and others	OS X	Web browser
	Data Explorer	Go To Node Search	Shift+F7	Shift+F7	Shift+F7
	Data Explorer	Open SQL Developer (available for Connection Nodes and below)	Alt+F9	Alt+F9	Alt+F9
	Data Explorer	View Data (available for tables/views/synonyms and other table types)	Alt+F10	Alt+F10	Alt+F10
	Data Explorer	Execute ETL Scenario (available for ETL Scenarios)	Alt+F8	Alt+F8	Alt+F8
	Any Text Editor	Search and Replace	Ctrl+F	Command+F	Ctrl +F7
	Any Text Editor	Go To the Line	Ctrl+L	Command+L	Ctrl+F8
	SQL Developer Code Editor	Format SQL	Shift+F9	Shift+F9	Shift+F9
	SQL Developer Code Editor	Show Code Snippets	Ctrl+F12	Command+F12	Ctrl+F12
	SQL Developer Code Editor	Describe Database Object (table/view/synonym)	Shift+F11	Shift+F11	Shift+F11
	SQL Developer	Describe Data Set (dataset displayed in the selected grid)	Shift+F12	Shift+F12	Shift+F12
	SQL Developer	Select Connection to open new SQL Editor window	Ctrl+N	Command+N	Shift+F4
	SQL Developer	Execute SQL	Ctrl+F2	Command+F2	Ctrl+F2

	SQL Developer	Execute SQL Script	Ctrl+F3	Command+F3	Ctrl+F2
	SQL Developer	Execute SQL in the External tool (if driver supports it)	Alt+F2	Alt+F2	Alt+F2
	SQL Developer	Show Execution Plan (if driver supports it)	Alt F3	Alt F3	Alt F3
	SQL Developer	Set Grid Defaults	Ctrl+F9	Command+F9	Ctrl+F9
	SQL Developer	Show SQL History	Ctrl+F10	Command+F10	Ctrl+F10
	Data Viewer	Select Table, View, Synonym or other Data Source to view data	Ctrl+N	Command+N	Shift+F4
	Data Viewer	View or Refresh Data	Ctrl+F2	Command+F2	Ctrl+F2
	Data Viewer	Set Grid Defaults	Ctrl+F9	Command+F9	Ctrl+F9
	Data Viewer	Describe Data Set (dataset displayed in the selected grid)	Shift+F12	Shift+F12	Shift+F12
	SQL Developer, Data Viewer, Excel Viewer	Export Data Set	F9	F9	F9
	SQL Developer, Data Viewer, Excel Viewer	Show Data Set Record	F4	F4	F4
	SQL Developer, Data Viewer, Excel Viewer	Copy data from the selected cells to the clipboard	Ctrl+C	Command+C	Ctrl+C
	SQL Developer, Data Viewer, Excel Viewer	Search in the Data Set	F7	F7	F7
	SQL Developer, Data Viewer, Excel Viewer	Calculate Function	Alt+F12	Alt+F12	Alt+F12
	SQL Developer, Data Viewer, Excel Viewer	Display Data as a Chart	Shift+F10	Shift+F10	Shift+F10

	SQL Developer, Data Viewer, Excel Viewer	Transform Data Set	Shift+F8	Shift+F8	Shift+F8
	ETL Scenario Code Editor	Open ETL Scenario	Ctrl+O	Command+O	Ctrl+F3
	ETL Scenario Code Editor	Save ETL Scenario	Ctrl+S	Command+S	Ctrl+F6
	ETL Scenario Code Editor	Format XML	Shift+F9	Shift+F9	Shift+F9
	ETL Scenario Code Editor	Show Code Snippets	Ctrl+F12	Command+F12	Ctrl+F12
	ETL Scenario Code Editor	Execute ETL Scenario	Alt+F8	Alt+F8	Alt+F8
	ETL Runner App	Execute ETL Scenario	Ctrl+F2	Command+F2	Ctrl+F2
	ETL Runner App	Refresh ETL Scenario	Ctrl+F9	Command+F9	Ctrl+F9
	Data Migration App	Create new Data Migration Scenario	Ctrl+N	Command+N	Shift+F4
	Data Migration App	Open Data Migration Scenario	Ctrl+O	Command+O	Ctrl+F3
	Data Migration App	Save Data Migration Scenario	Ctrl+S	Command+S	Ctrl+F6
	Data Migration App	Save as ETL Scenario	Ctrl+F11	Command+F11	Ctrl+F11
	Data Migration App	Execute Data Migration Scenario	Ctrl+F2	Command+F2	Ctrl+F2
	Content Node	Save File	Ctrl+S	Command+S	Ctrl+F6
	Content Node	Sort By Name	Ctrl+F7	Command+F7	Ctrl+F7
	Content Node	Sort By Type	Ctrl+F8	Command+F8	Ctrl+F8

	Content Node	Sort By Size	Ctrl+F9	Command+F9	Ctrl+F9
	Content Node	Sort By Date	Ctrl+F10	Command+F10	Ctrl+F9

## Appendix 2. System Variables and Folders

Data Explorer uses Home folder by default to store all sort of files, from ETL scenarios to data files. In the Web and client-server modes there is a shared Home folder and personal folder under Home. Each user has his/her own personal folder. Personal folder name is calculated using login name so it is important to have unique logins for each user in the multi user modes such as Web and client-server.

System variables can be used when defining URLs for the connections, file names etc.

Variable	Definition	Example
{app.home}	The root folder where application is installed	{app.home}/doc
{app.data}	The local DATA folder. In the single user mode (client) this is a {app.home}/data. In the multi user modes (Web and client-server) each user has his/her own home folder under {app.home}/data. Only user with a particular login name has an access to this home folder.	{app.data }/errors
{app.root.data}	The root DATA folder. All users have access to this folder	{app.root.data}/scenario