

SQLeo Beginner Users Guide

Revised: 14/06/2012

by Alan Shiers

Revised: 26/09/2016

by PAscal

Table of Contents

Getting started	4
Starting SQLLeo.....	4
Choosing Look and Feel	4
Starting SQLLeo for Mac OS	5
Starting SQLLeo as a portable app	5
Starting SQLLeo to be able to use Unicode exports / imports	5
Choosing Preferences.....	6
Defining JDBC Drivers	6
JDBC Drivers download	7
Creating database connection (datasource).....	8
Main menu Overview	10
Files	10
Actions	10
Tools	10
Windows	10
Help	10
Metadata explorer	10
The Metadata browser mode	11
The Metadata Search Feature	12
The Content Window	13
Displaying data of a Table	13
Sorting Data	15
Filtering Data	15
Finding Terms	17
Jump to Foreign or Parent data	17
Inserting and Deleting Records from a Table	18
Command Editor	18
Using command editor result in text mode	18
Using command editor result in grid mode	18
Command line features	19
New command	19
The Visual Query Designer.....	19
The Designer Mode.....	20

Adding WHERE condition to the query	23
The Syntax Mode	26
Adding ORDER BY clause to the query	27
Adding More Tables to the Query	29
The DEFINITION Window	32
SQL History	33
Data comparer	33
Schema comparer.....	33
Troubleshooting	33
Support.....	33

This guide will cover most of the basic features targeting the beginner user and while SQLeo has many advanced features, these may be covered in another guide for advanced power users.

Getting started

Starting SQLeo

Default is double-click on SQLeoVQB.jar file

or launch :

java -jar SQLeoVQB.jar

java executable should be available in your path but

<java path>/java -jar SQLeoVQB.jar

works as well.

Choosing Look and Feel

Java Look and feel can be chosen at startup time as described in file sqleo-start.help.

javax.swing.plaf.metal.MetalLookAndFeel is the default Look and Feel.

Other available look and feels from jdk can be used with command line:

java -Dcom.sqlleo.laf.class=<L&F> -jar SQLeoVQB.jar

where <L&F> can take one the following values:

- javax.swing.plaf.metal.MetalLookAndFeel
- com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel
- com.sun.java.swing.plaf.windows.WindowsLookAndFeel
- com.sun.java.swing.plaf.motif.MotifLookAndFeel
- com.sun.java.swing.plaf.gtk.GTKLookAndFeel

- com.apple.laf.AquaLookAndFeel

Command files sqleo-start-Metal-Steel.txt, sqleo-start-Nimbus.txt, sqleo-start-Windows.txt are delivered in the application zip file, just change the .txt extension to

- .bat for Windows users,
- .sh for linux users,
- .command for MacOSX users

to get your preferred appearance.

Starting SQLLeo for Mac OS

sqleo-start-MacOS.txt, after being renamed to sqleo-start-MacOS.command gives a full Mac Os integrated application, with the associated Look and Feel, Mac OS short cuts, menu bars, ...

It can also be launched using command line

java -classpath SQLLeoVQB.jar com.sqlleo.environment.SQLLeoMacApp

see [App class for Mac users](#) for more details

Starting SQLLeo as a portable app

You can rename sqleo-start-PortableApp.txt to your OS command file or use the hereafter command line to be able to start application from an USB key or a local directory:

java -Duser.home=<chosen directory path> -jar SQLLeoVQB.jar

<chosen directory path> can be:

- C:\temp
- D:\mydirectory
- /tmp

This directory will contain driver and applications preferences files (driver definition, chosen language, connexion URLs, username, password, ...) that are located in user %HOME% if not specified.

Starting SQLLeo to be able to use Unicode exports / imports

By default SQLLeo use default locale language as code page for exported or imported files.

This can be changed using template `sqleo-start-UTF8.txt` or using command line

`java -Dfile.encoding=UTF-8 -jar SQLeoVQB.jar`

This can be mixed with Look & Feel or other java options.

Choosing Preferences

When navigating to Tools -> Preferences many preferences are available like:

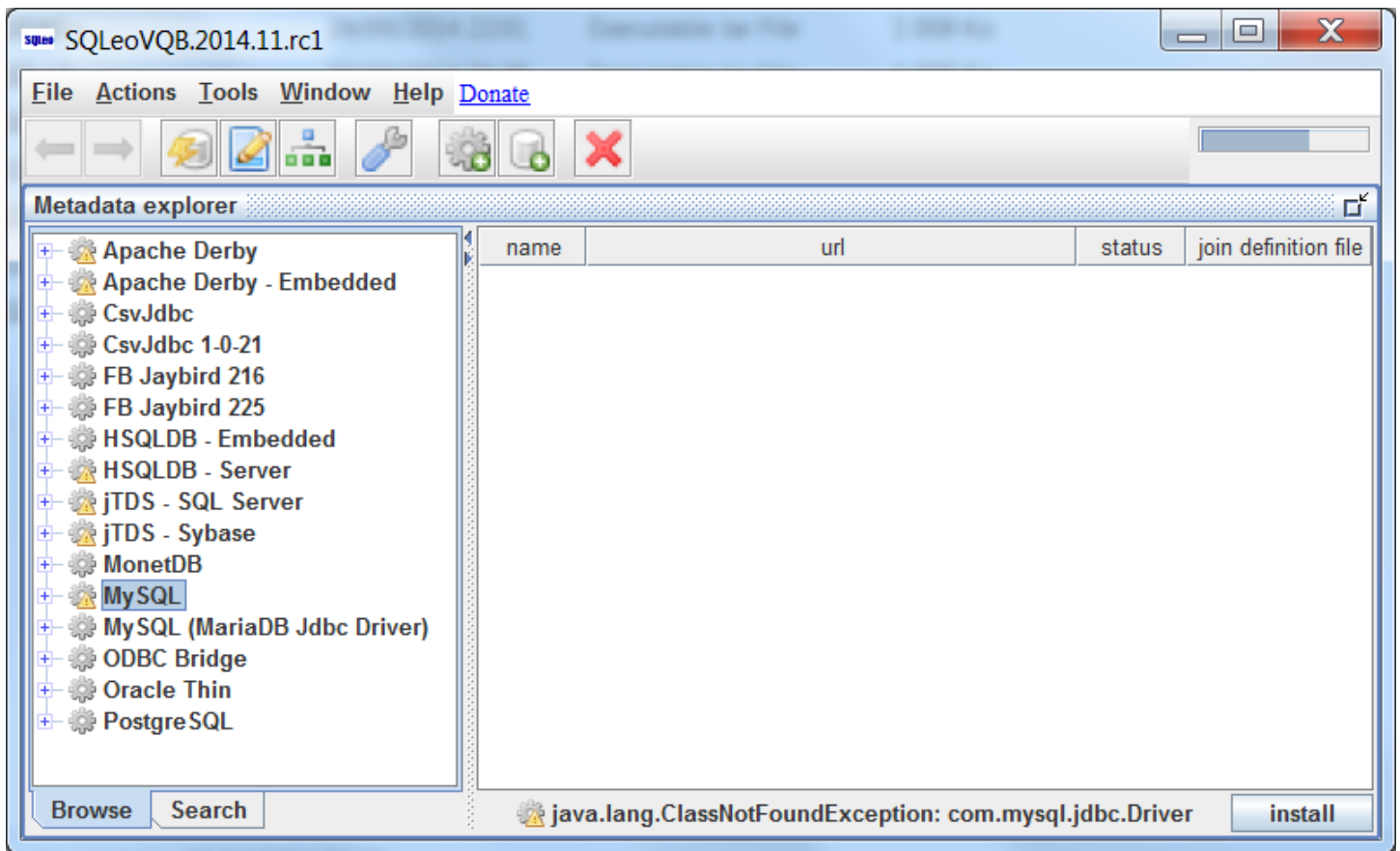
- ➔ Language (French, German, Italian, Polish, Portuguese, Russian, Serbian, Spanish, ...)
- ➔ Font size
- ➔ Icon size
- ➔ Using schema names in SQL syntax
- ➔ ...

More informations can be found in SQLeo Advanced Help document.

Defining JDBC Drivers

When you first launch SQLeo, you are presented with an interface that displays a list of supported database systems on the left hand pane of the Metadata Explorer (many other RDBMS also supports JDBC. They can be added using the menu *New driver* icon or right click in the left hand pane). See IMAGE 1. If you select any one of the items in the list, a message will appear at the bottom of the interface stating that it could not find the JDBC Drivers for that particular database system. The message refers to a *ClassNotFoundException* and names the file it requires. Next to the message is a button labelled “install” that you can use to launch a dialog box that allows you to navigate to the directory on your hard drive where you are storing your JDBC Drivers.

IMAGE 1



JDBC Drivers download

Database Name	Class	URL	Comments
Access		https://sourceforge.net/projects/ucanaccess	No need of ODBC, works on windows AND Linux ...
Apache Derby	org.apache.derby.jdbc.EmbeddedDriver	https://db.apache.org/derby/derby_downloads.html	
Csvjdbc	org.relique.jdbc.csv.CsvDriver	https://sourceforge.net/projects/csvjdbc/	See (2)
DB2			
Firebird	org.firebirdsql.jdbc.FBDriver	http://www.firebirdsql.org/en/jdbc-driver/	Version / java
Interbase	interbase.interclient.Driver		
H2	org.h2.Driver	http://h2database.com/html/download.html	
HSQLDB	org.hsql.jdbcDriver	https://sourceforge.net/projects/hsqldb	
MariaDB	org.mariadb.jdbc.Driver	https://downloads.mariadb.org/connector-java/+releases/	See (1)
MonetDB	nl.cwi.monetdb.jdbc.MonetDriver	http://dev.monetdb.org/downloads/Java/	
MySQL	com.mysql.jdbc.Driver	https://dev.mysql.com/downloads/connector/j/	See (1)
Oracle	oracle.jdbc.driver.OracleDriver	http://www.oracle.com/technetwork/database/features/instant-client/index.html	
PostgreSQL	org.postgresql.Driver	https://jdbc.postgresql.org/download.html	

SQL Anywhere			
SQL server / Jtds	net.sourceforge.jtds.jdbc.Driver	https://sourceforge.net/projects/jtds	
SQLite	org.sqlite.JDBC	https://github.com/xerial/sqlite-jdbc/releases	See (3)

(1) MySQL (MariaDB Jdbc Driver) used to connect for both MariaDB AND MySQL **is already bundled in SQLeoVQB.jar** (no need to install it), it offers a multischema view of MySQL databases.

(2) CsvJdbc driver is provided in lib directory and can be used to read csv file as tables (no need to go on internet to download it, except if you need a newer version ...)

(3) <http://kenfallon.com/adding-sqlite-as-a-datasource-to-sqleo/>

Creating database connection (datasource)


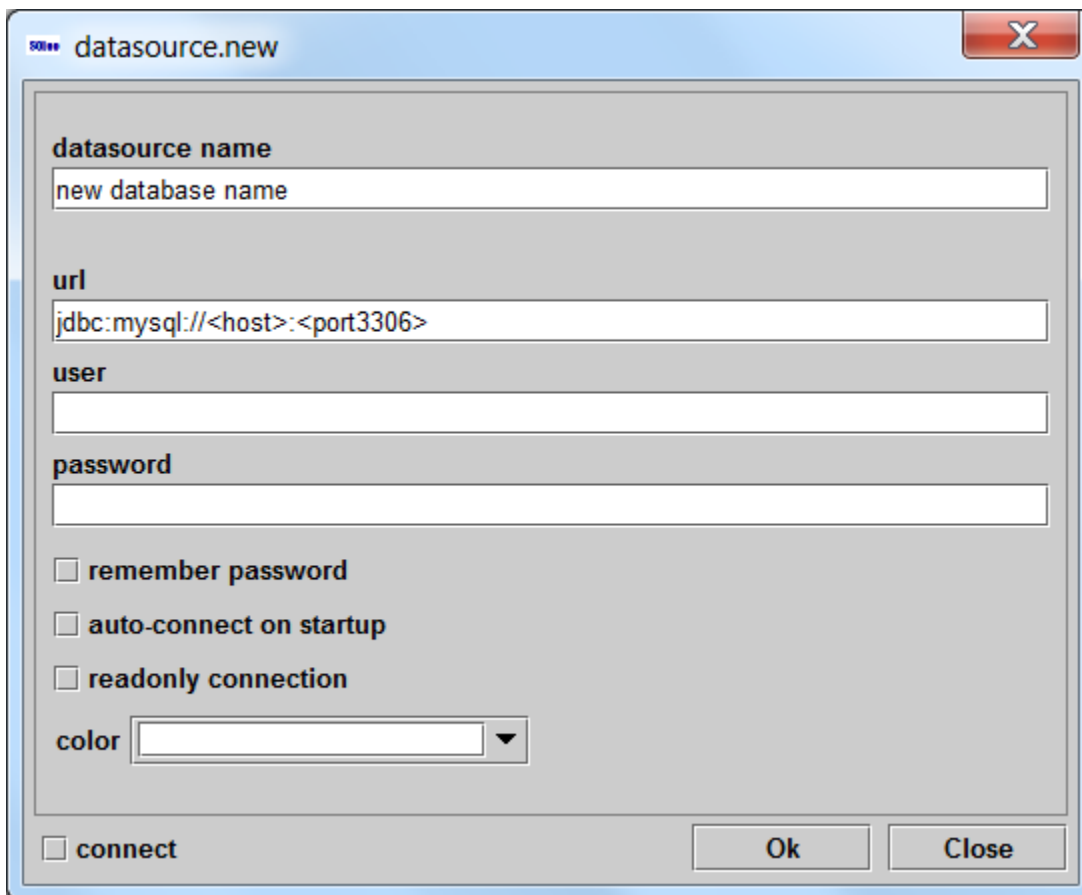
Once you have told SQLeo where to find the Drivers for the database system you are attempting to connect to, you can then provide SQLeo the datasource information it requires to make a connection. To do this, you can click on the button with the image:  and labelled *new datasource* or right-click on the driver name and choose *new datasource*. In the case of connecting to a MySQL database system, you will be presented with the following dialog window:

IMAGE 2



The dialog window titled "datasource.new" contains the following fields and options:

- datasource name:** A text field containing "new database name".
- url:** A text field containing "jdbc:mysql://<host>:<port3306>".
- user:** An empty text field.
- password:** An empty text field.
- remember password:** An unchecked checkbox.
- auto-connect on startup:** An unchecked checkbox.
- readonly connection:** An unchecked checkbox.
- color:** A dropdown menu.
- connect:** An unchecked checkbox.
- Buttons:** "Ok" and "Close" buttons at the bottom right.

In the field labelled *name*, type a new name for the database you are connecting to. In the field labelled *url* edit the existing string: `jdbc:mysql://<host>:<port3306>/<database>`

Replace those parts that are in brackets: `<...>` with:

- `<host>` is the server name where the database system resides on the network or over the internet. Typically this would follow the pattern such as: `www.someplace.com` or an IP Address. If the database resides on your computer and not on the network, then you would replace `<host>` with the term: `localhost` or `127.0.0.1`
- `<port3306>` is the port on which the database listens to incoming requests. Though this can be changed by an administrator, the port number by default is 3306. The port number will be different depending on the RDBMS.
- `<database>` would be the actual name given to the database. **This part can be skipped with MariaDB jdbc driver that permits to see all the databases (schemas) inside the MySQL instance.**

The connexion string should end up looking something like this: `jdbc:mysql://myserver.mydomain:3306/mydb`

Enter the user name and password and check off the additional options as required. Click the OK button to connect to the database. Read only connections are possible (even with non-read only user accounts) and the background color of the tool can be customized here (to be able to distinguish prod and non prod connections easily for exemple)

Main menu Overview

When connected, here are the main windows and commands available:

Files

- New query
- Load query
- Recent queries

Actions

Actions are depending on the window that is activated.

Tools

- METADATA explorer
- COMMAND Editor
- QUERY designer
- SQL History
- DATA comparer
- SCHEMA comparer
- CONTENT window
- DEFINITION window
- Preferences


Windows

- Cascade
- Tile Horizontal
- Close All
- <current opened windows list>

Help

- Keyboard Shortcuts
- Online Documentation

Metadata explorer

Metadata Explorer is the default window that can be recalled by clicking on the button with the image  and labelled metadata explorer on the button toolbar.

The Metadata browser mode

The tree structure found in the left hand pane in IMAGE 3 contains a listing of many of the database types SQLLeo can connect to. In the image, the TABLE node is selected and therefore the content pane will display the list of Tables contained within this database. Currently selected is the *employees* table.

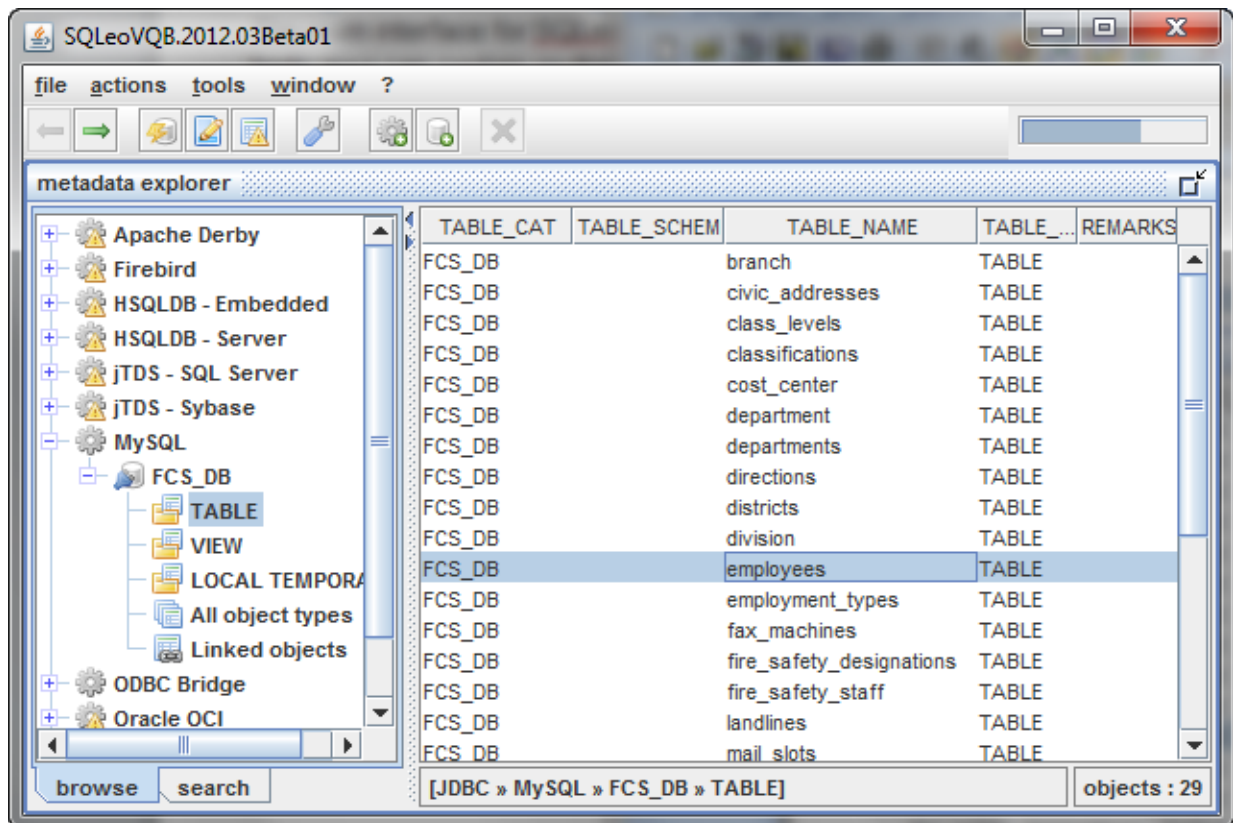


IMAGE 3

With the *employees* table already selected, you can use the right mouse button to bring up additional menu items which provide options on the selected table.

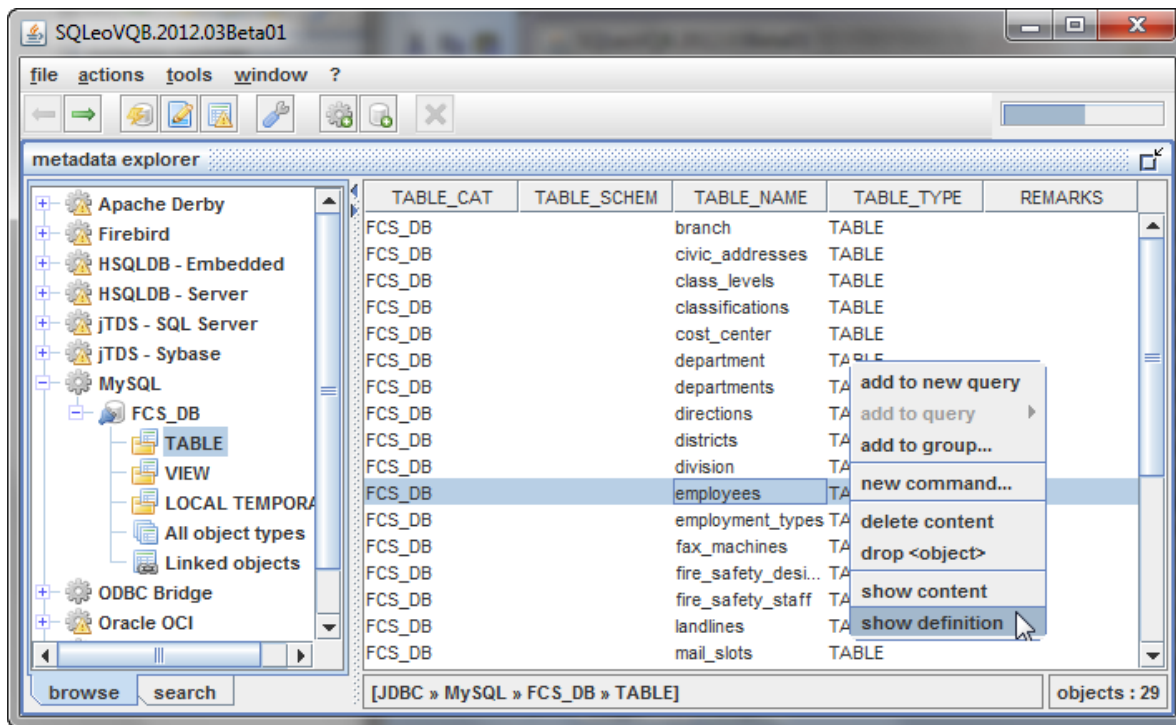


IMAGE 4

The Metadata Search Feature

At the bottom of the Metadata Explorer internal window are two tabs. By default the **browse** tab is selected allowing the user to navigate the tree of database types. Selecting the **search** tab allows to perform a search on the entire database schemas.

As an introduction and an example, text *email_address* being entered into the column field and option *contains* being selected gives results found in IMAGE 5 when running the search with this criteria.

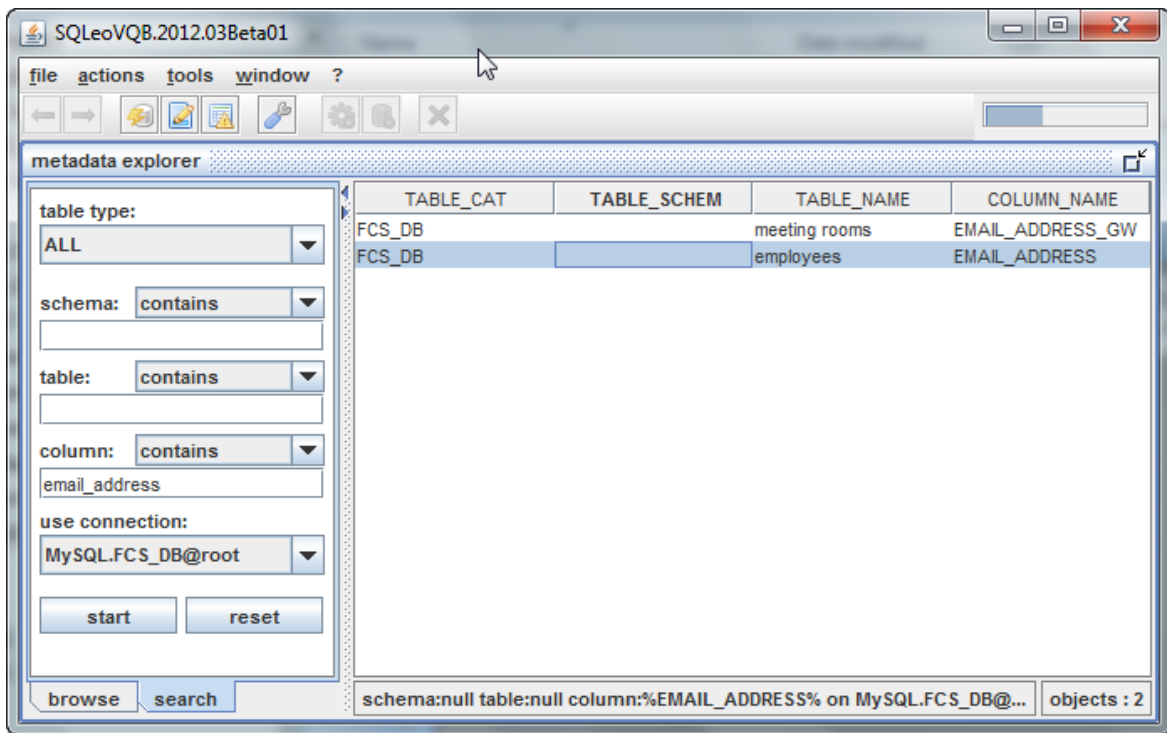


IMAGE 5

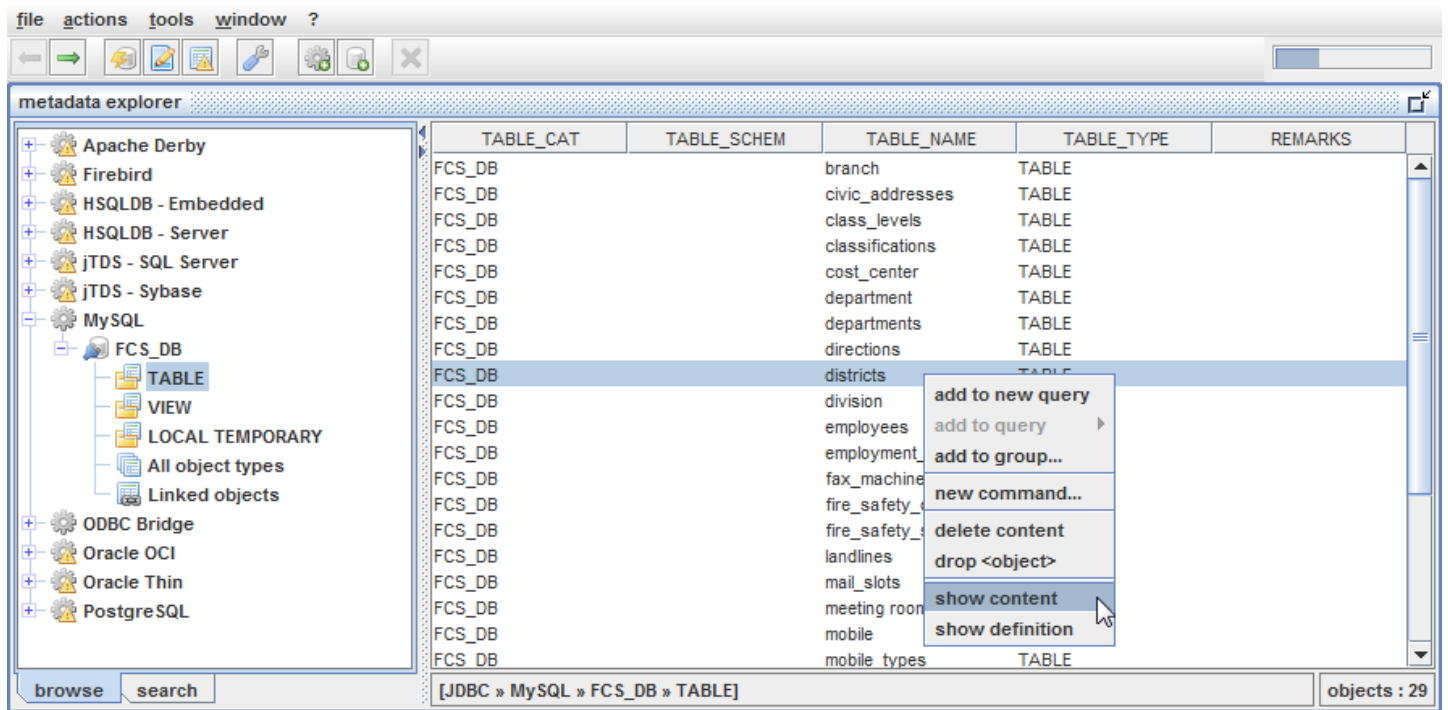
Be careful starting with version 2016.04 : **search text are case sensitive** (not like in this exemple).

The Content Window

Displaying data of a Table

From Metadata Explorer, any table found in the Content pane, can be displayed using **double click** or using the right mouse button: *show content* menu option as shown in image 6.

IMAGE 6



Then Content Window is displayed

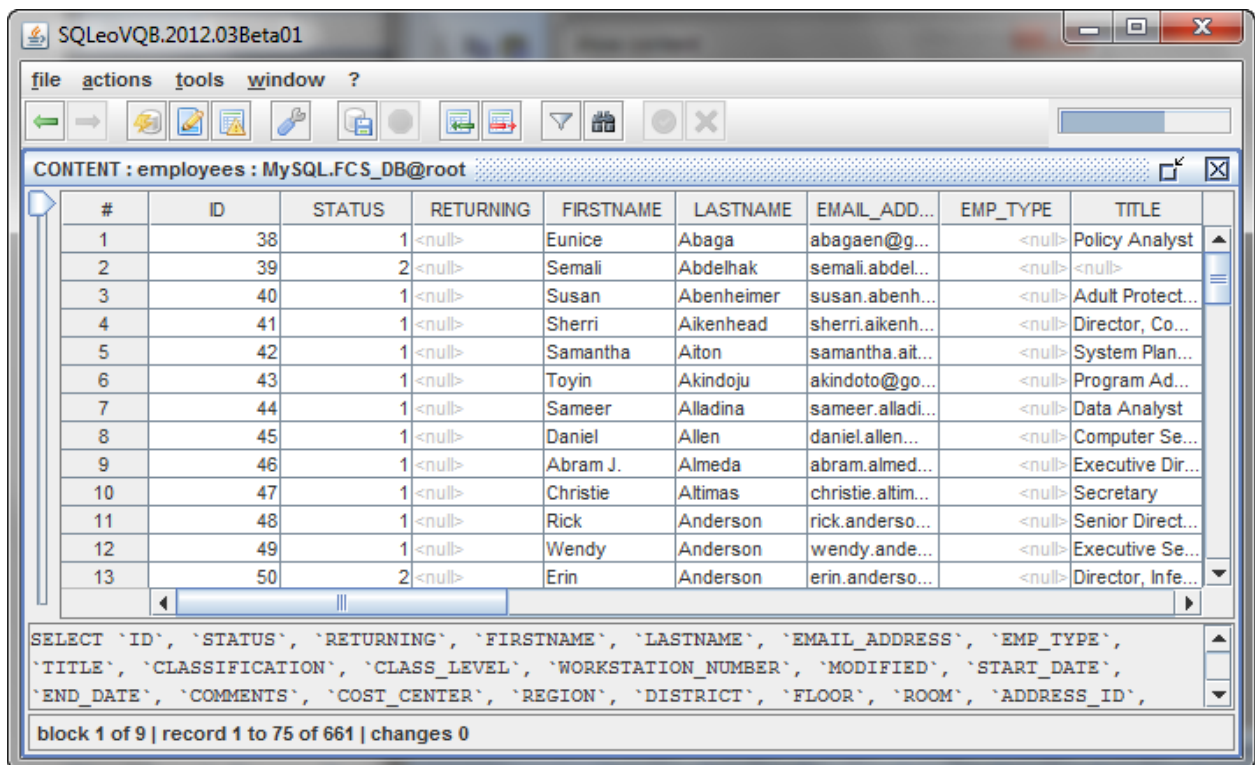


IMAGE 7

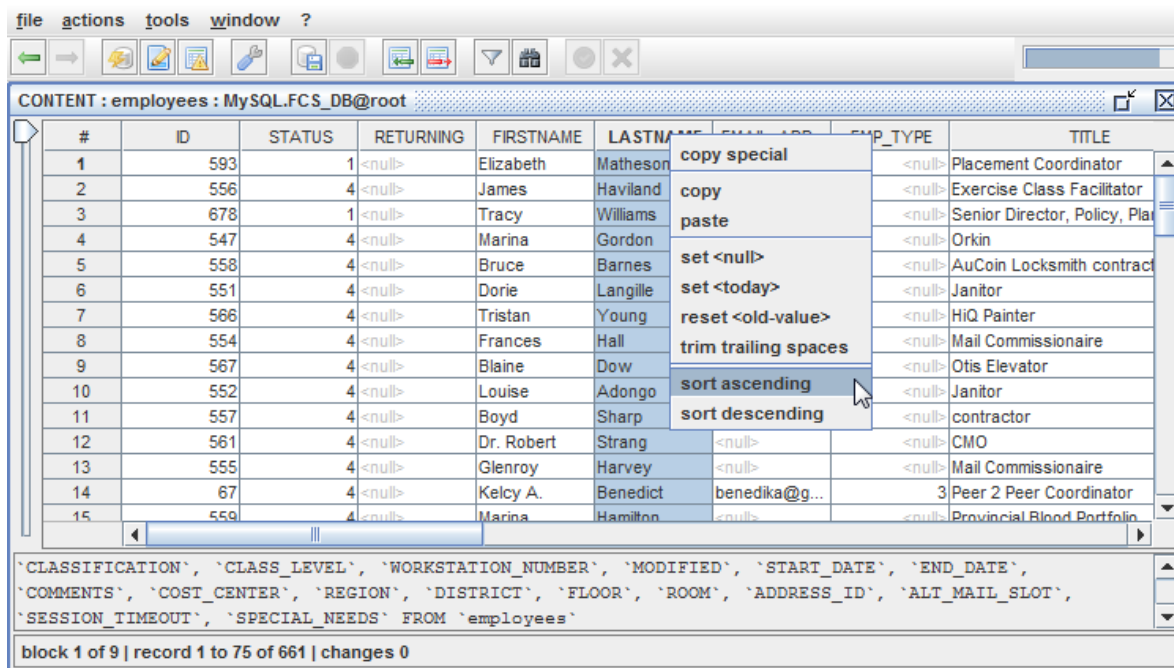
At the bottom of the CONTENT window is the current SQL query.

This Content window will not load all the records in memory, but only the first 100 records (this is a parameter that can be modified in preferences). As the user scrolls down the lists of records SQLLeo will retrieve the next 100 records for display.

Sorting Data

Sorting of data is possible when right clicking on the table column names as shown in image 8

IMAGE 8



The result of the sort appears (data is sorted at the RDBMS level, only the 100 first rows are displayed)

Filtering Data


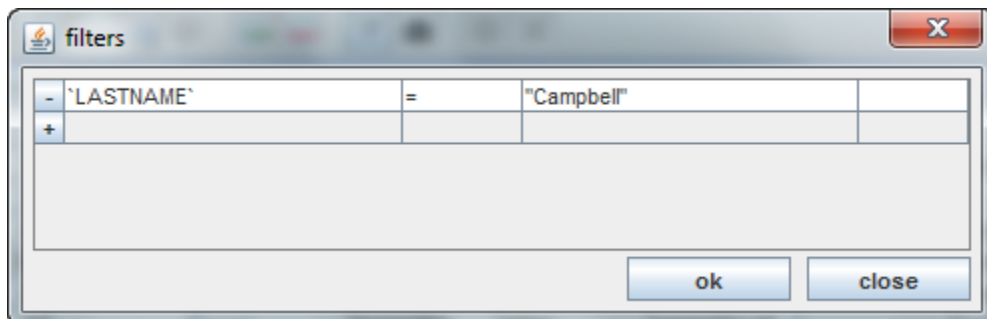
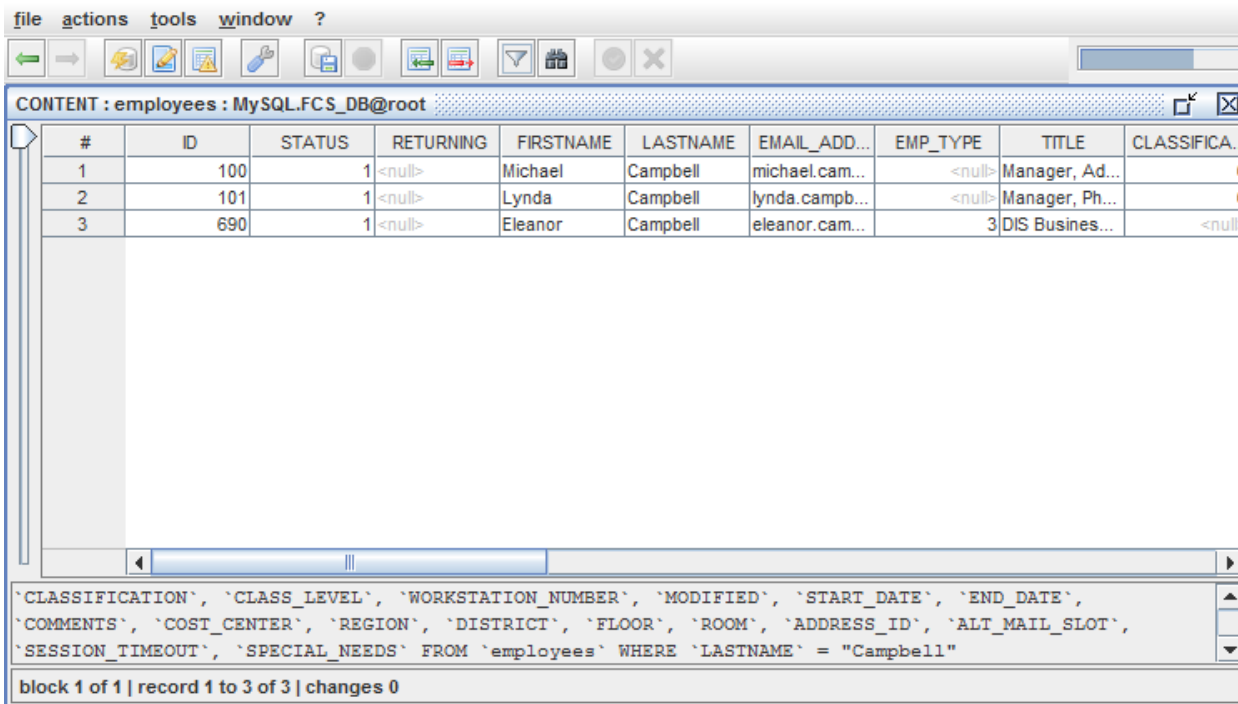
If you wish to narrow your focus on certain records, you can filter the data to only those records that are of interest. From the toolbar select the filter  button. Doing this brings up a dialog box where we can enter criteria. We will propose a simple criteria and enter LASTNAME = "Campbell" as in IMAGE 9.

IMAGE 9



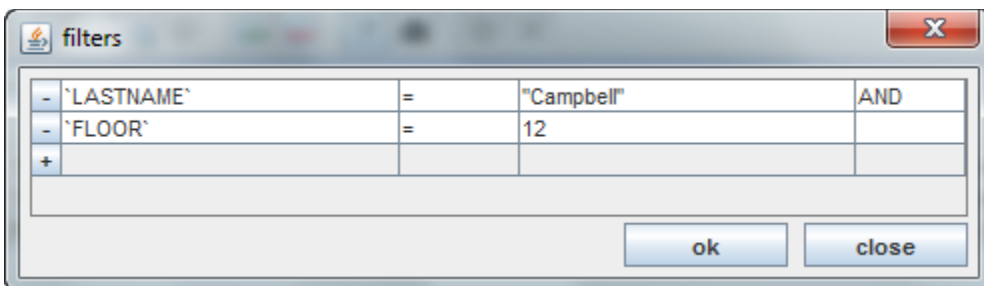
Clicking on the OK button, the CONTENT window displays only those records whose LASTNAME equals "Campbell". See IMAGE 10.

IMAGE 10



Returning to the filters dialog box, note that there are numerous options to set your criteria. Where the equals symbol is displayed in IMAGE 9, you will find the options: =, <, >, <=, >=, <>, LIKE, NOT LIKE, etc. You can also enter more than one criteria using the AND keyword, or OR keyword. See IMAGE 11.

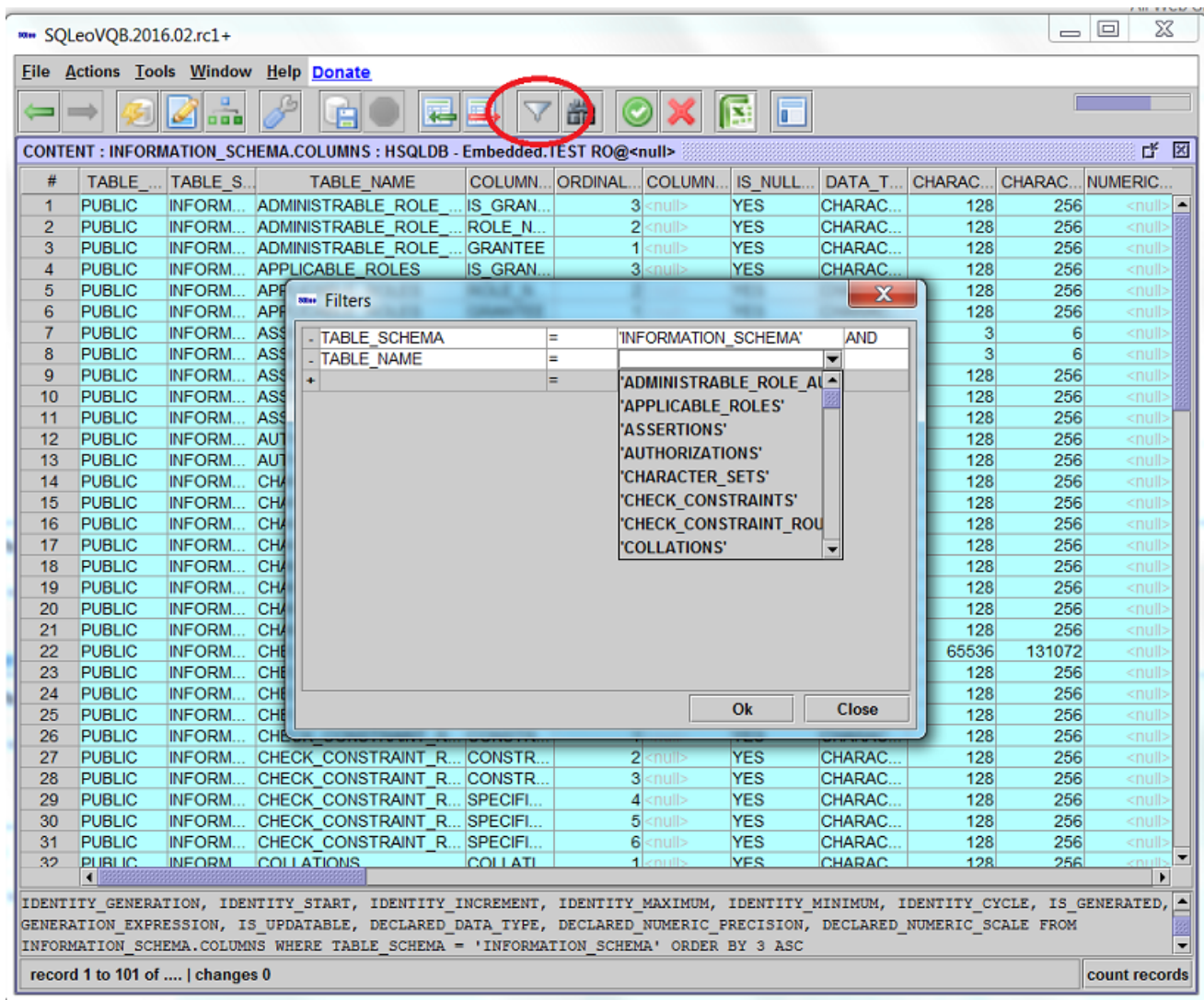
IMAGE 11



Filters populated with values


This is an option in preferences, that queries the distinct values of the select column in the database.

IMAGE 12



Note: this window has a blue background color (chosen in the datasource definition)

Finding Terms

One final search option is available to the user. It is the Find  button. Clicking on this button on the toolbar brings up a familiar find dialog box which you can use to find, and even replace, any term that may reside inside all the **data** within a given window.


Jump to Foreign or Parent data

When using the jump function on cell, SQLLeo checks if that cell's column is referenced via a foreign key (or a Virtual foreign key as described in the join definition file), and displays corresponding foreign tables list. When choosing such a table, SQLLeo will then open a new CONTENT windows displaying data of the primary or foreign table with a filter corresponding to the initial cell's value.

Inserting and Deleting Records from a Table

See SQLLeo Advanced Help document

Command Editor

To open the Command Editor window, click on the Command Editor button  on the toolbar.

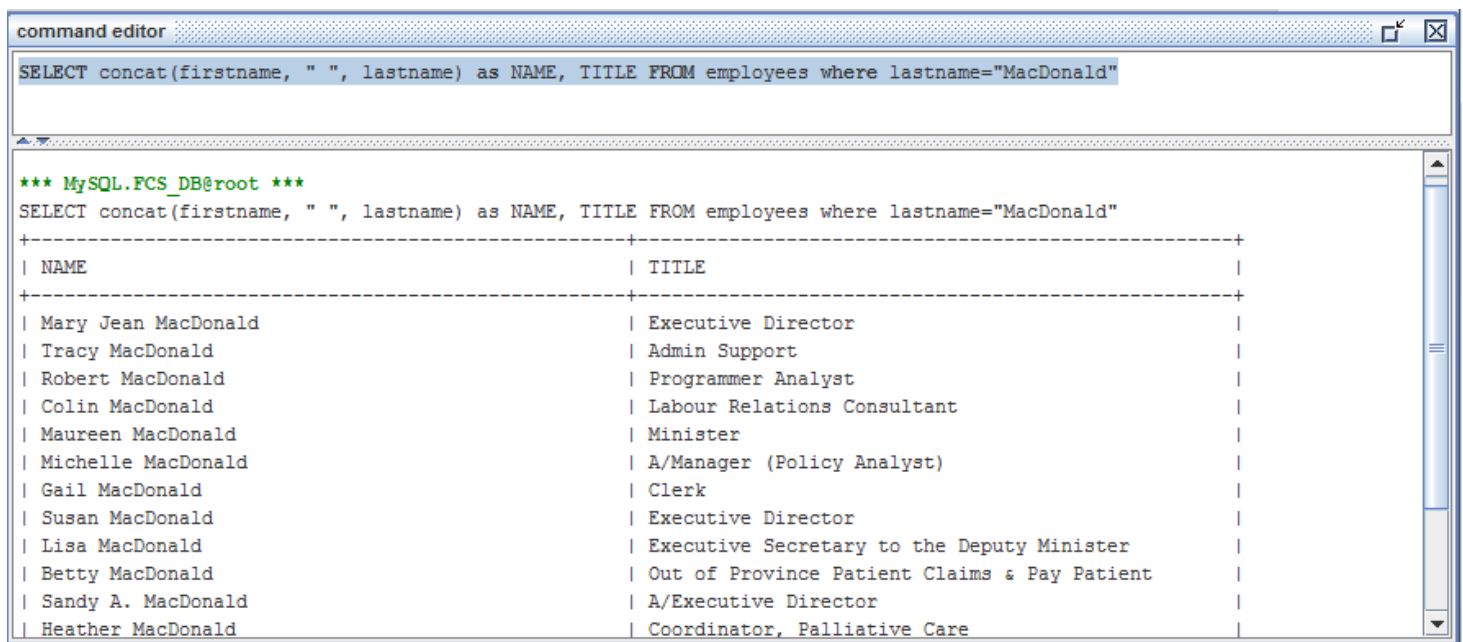
The Command Editor is a separate internal window you can use instead of the Visual Query Designer. The Command Editor is intended for typing **many SQL statements as** ALTER, SELECT, INSERT, DELETE, UPDATE, DROP, ... and even PL/SQL **commands in the same window (when query builder only supports ONE SELECT only).**

Command editor also permits to Switch between different databases connections.

Using command editor result in text mode

When “grid output” check box is not selected, SQL syntax and Query result are displayed in text mode as seen in image

IMAGE 13



Using command editor result in grid mode

When “grid output” check box is selected, Query result is displayed in grid mode as in CONTENT window. Data can be sorted using right click

Command line features

Type “help” to get all the commands available

Command	Usage	description
Clear	Clear	Clears the command editor output text
Connect	connect <datasource>	connects a datasource whose expected format is same value in the connection dropdown
Format	format csv delimiter <char> header true false quote true false	CSV export format, default options : delimiter ; header true quote false
Help	Help	Shows all available commands
Input	input <filename.sql>	Run the commands provided in input file
Output	output text grid csv <filename> append replace	Extract query output to text or grid or csv. default option is replace when not provided
Quit	Quit	Quits the command mode

Exemple:

See [Command Editor: How to choose datasource and output format in script](#)

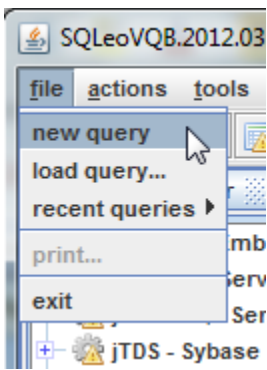
New command

It is an helper to generate syntax for INSERT, UPDATE and DELETE statements ... see See SQLLeo Advanced Help document for more details.

The Visual Query Designer

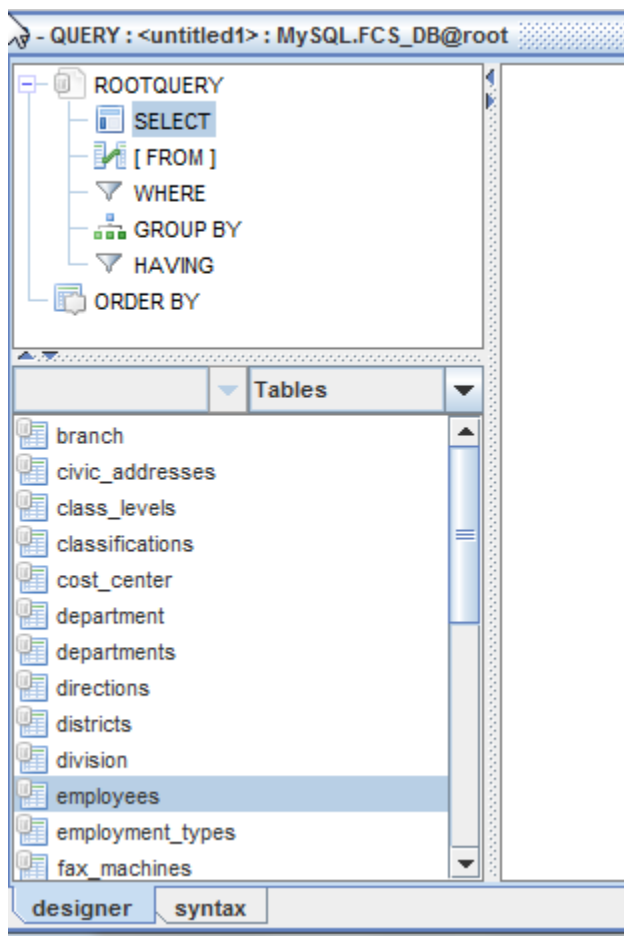
The Visual Query Designer can be started by selecting the File/New Query menu (or CTRL-N shortcut).

IMAGE 15



The Designer Mode

IMAGE 16



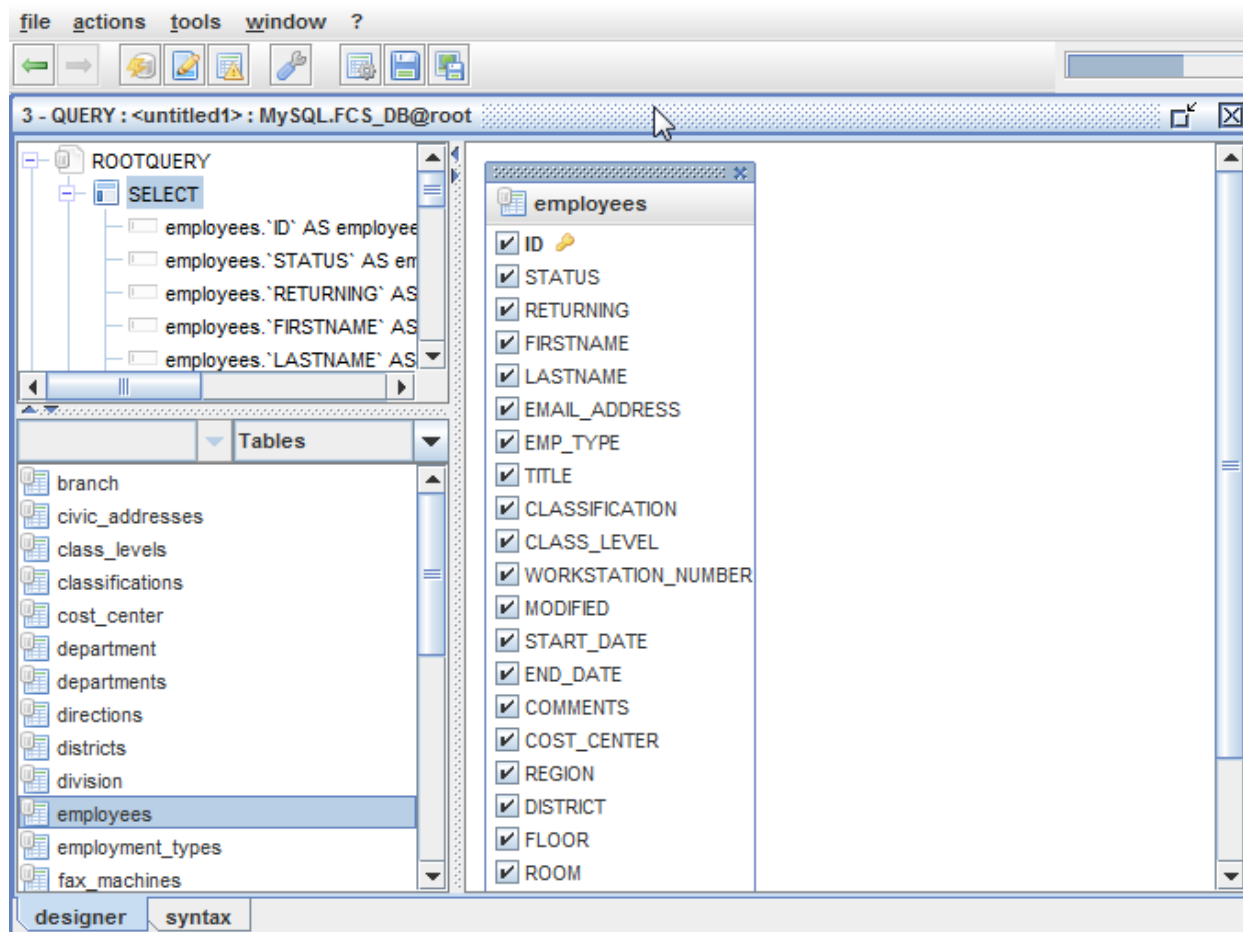
At the bottom of the window are two tabs labelled **designer** and **syntax**. The QUERY window will automatically load all the Table names in the lower portion of the window (this can be modified in the preferences query builder / load table objects at once). In the top portion there is a tree like structure where each node is labelled in accordance with the keywords from the SQL language: SELECT, FROM, WHERE, GROUP BY, HAVING, AND ORDER BY. With the use of the right mouse button you will be able to access popup menus that provide additional options when you select each node in the tree.

First, we want to select a Table from which we want to extract data. To construct a simple query, we will select the *employees* Table. You can either double click the table in the list, or you can drag 'n drop the Table into the Content pane on the right.

You can perform the same procedure in the Metadata Explorer by selecting a Table, right clicking on it to obtain the popup menus with the options: **new query** or **add to query...**

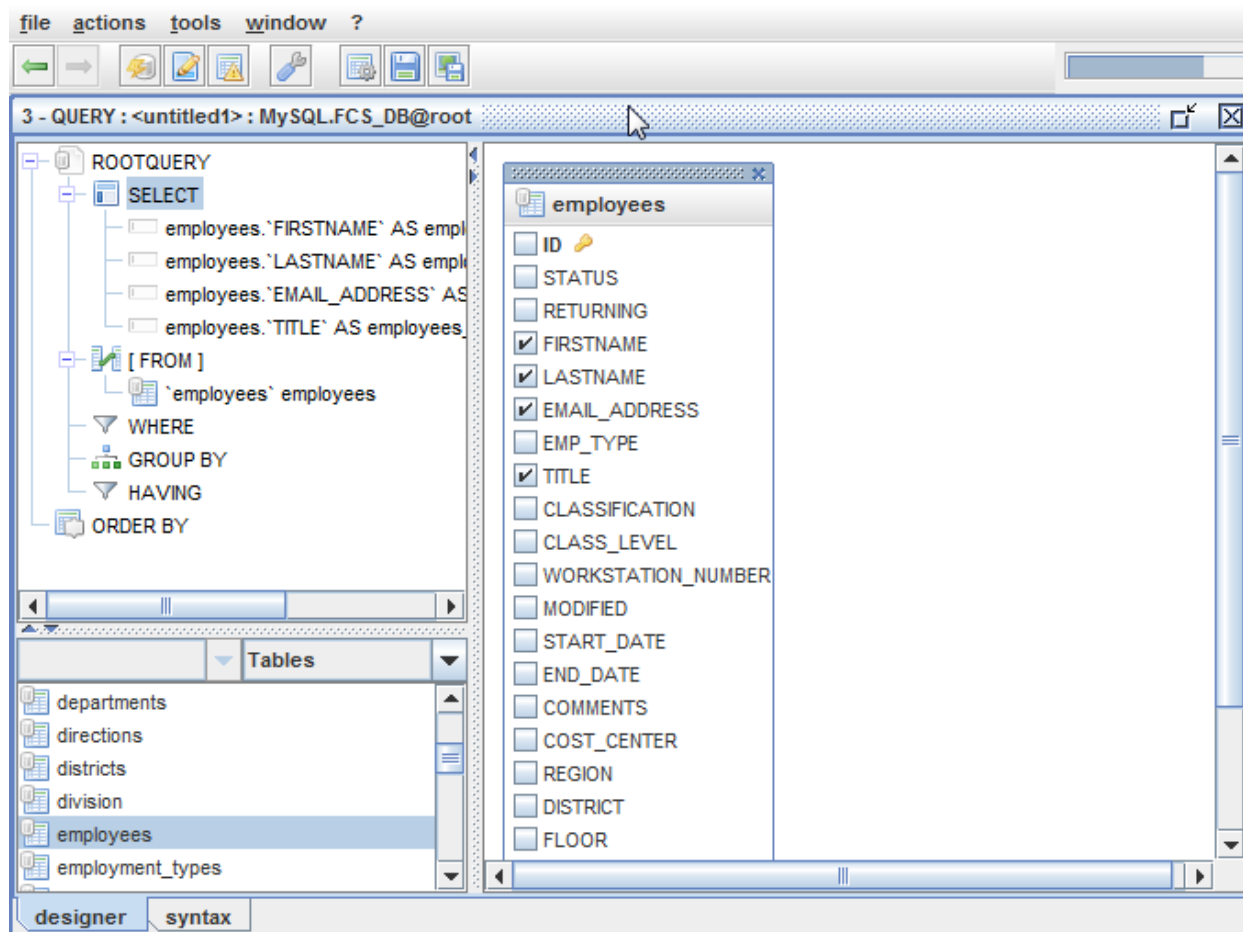
You should see your Table in the Content pane as in IMAGE 17.

IMAGE 17



The Content pane displays another internal window that contains the listing of all the Columns contained within the *employees* Table. Each Column name has next to it a checkbox with a check inside each of these. Notice also, that the SELECT node also displays all the Column names. What we need to decide now is what Columns are we really interested in. We will uncheck all the Columns except for FIRSTNAME, LASTNAME, EMAIL_ADDRESS, and TITLE. As we uncheck the Columns, the list will decrease on the SELECT node. In our example we see the following in IMAGE 18:

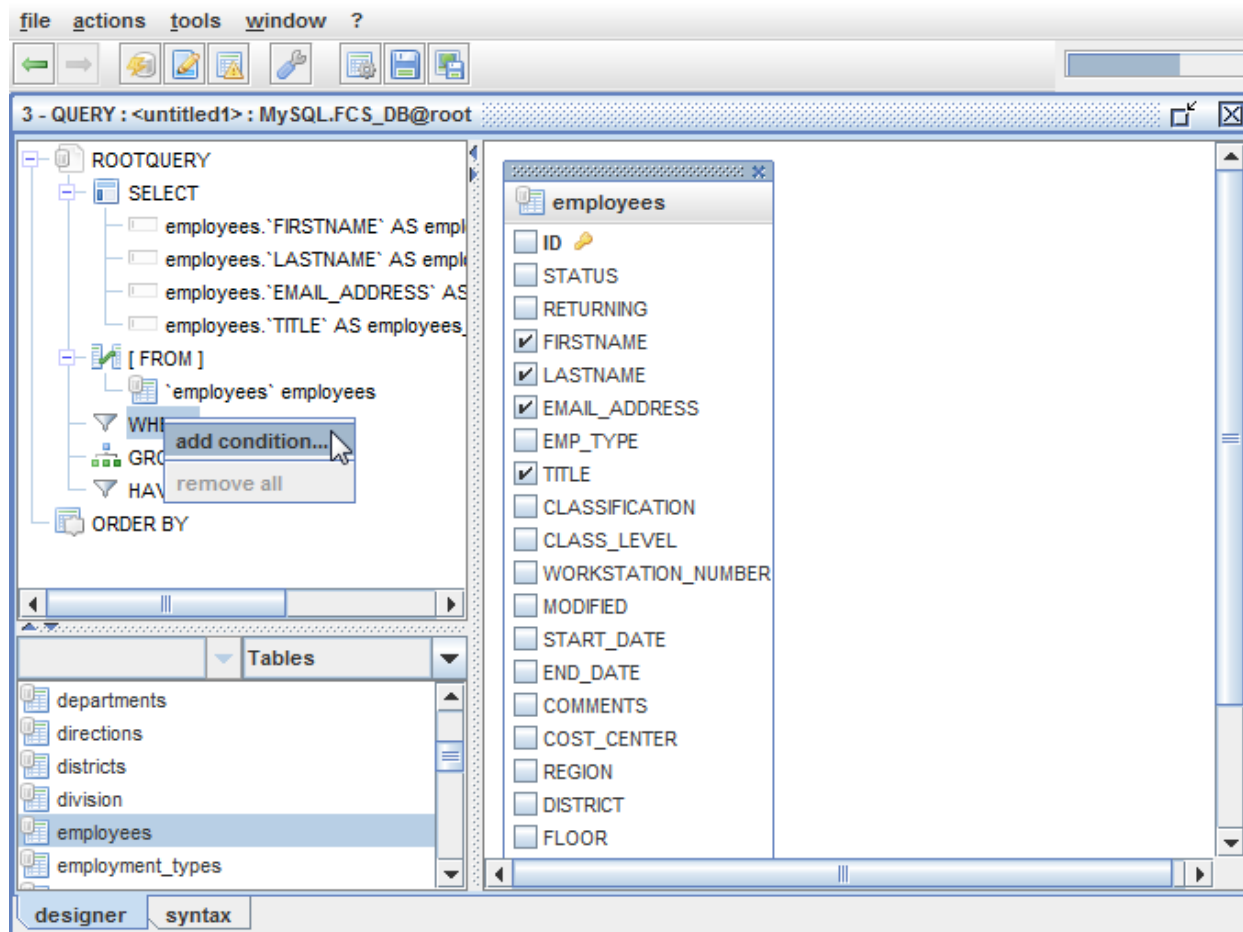
IMAGE 18



If we were to run this query as is, we would have a tremendous listing of employee names. We are really only interested in seeing records on employees whose last name is "Campbell". So, we will place a condition on this query stating just that. To add a condition on a query, we will right click on the WHERE node. This will cause a popup menu to appear with the option: add condition.

Adding WHERE condition to the query

IMAGE 19



Selecting this option will bring up a dialog box that allows you to create a condition based on several expression operators: =, <, >, <=, >=, <>, LIKE, NOT LIKE, etc. These operators can be accessed from the drop down combo box as seen in IMAGE 20.

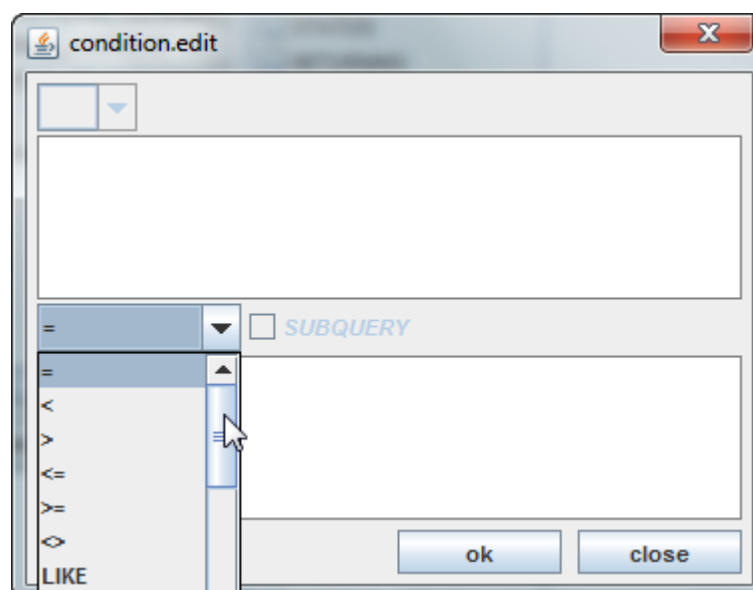
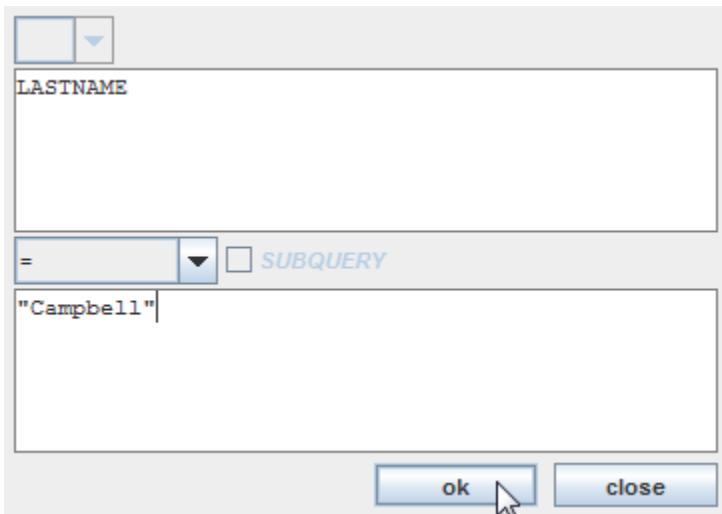


IMAGE 20

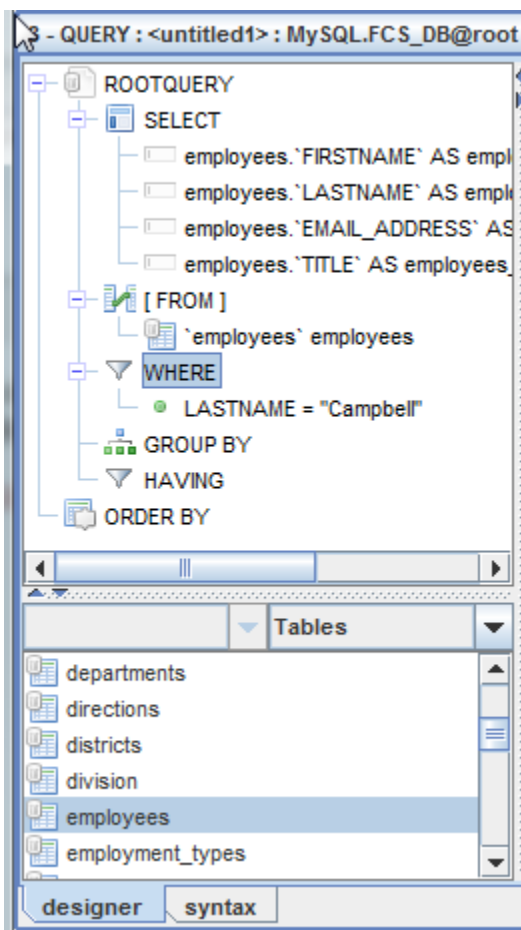
For our example we will leave the default equals sign (=) and type in our condition. See IMAGE 21. The top textbox is where you type the Column name you wish to place the condition on. The bottom textbox is where you type the remainder of the expression. In our case, we type "Campbell" in quotes.

IMAGE 21



After clicking the OK button, we are returned to the QUERY window. Notice now in IMAGE 22 that the WHERE node contains our condition.

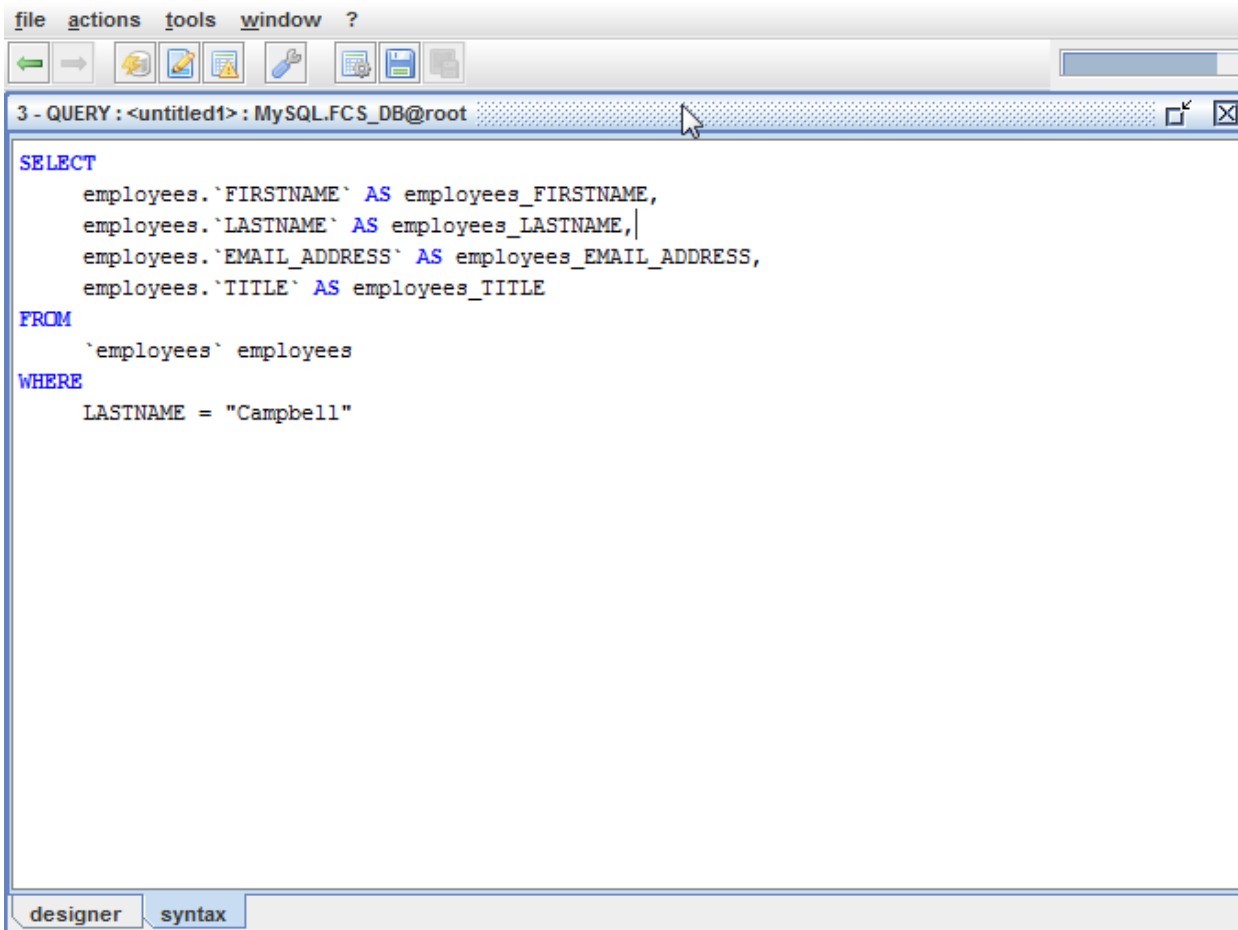
IMAGE 22



The Syntax Mode

When selecting the syntax tab (at the bottom of the window), we see how the actual SQL query has been constructed by SQLeo. See IMAGE 23.

IMAGE 23

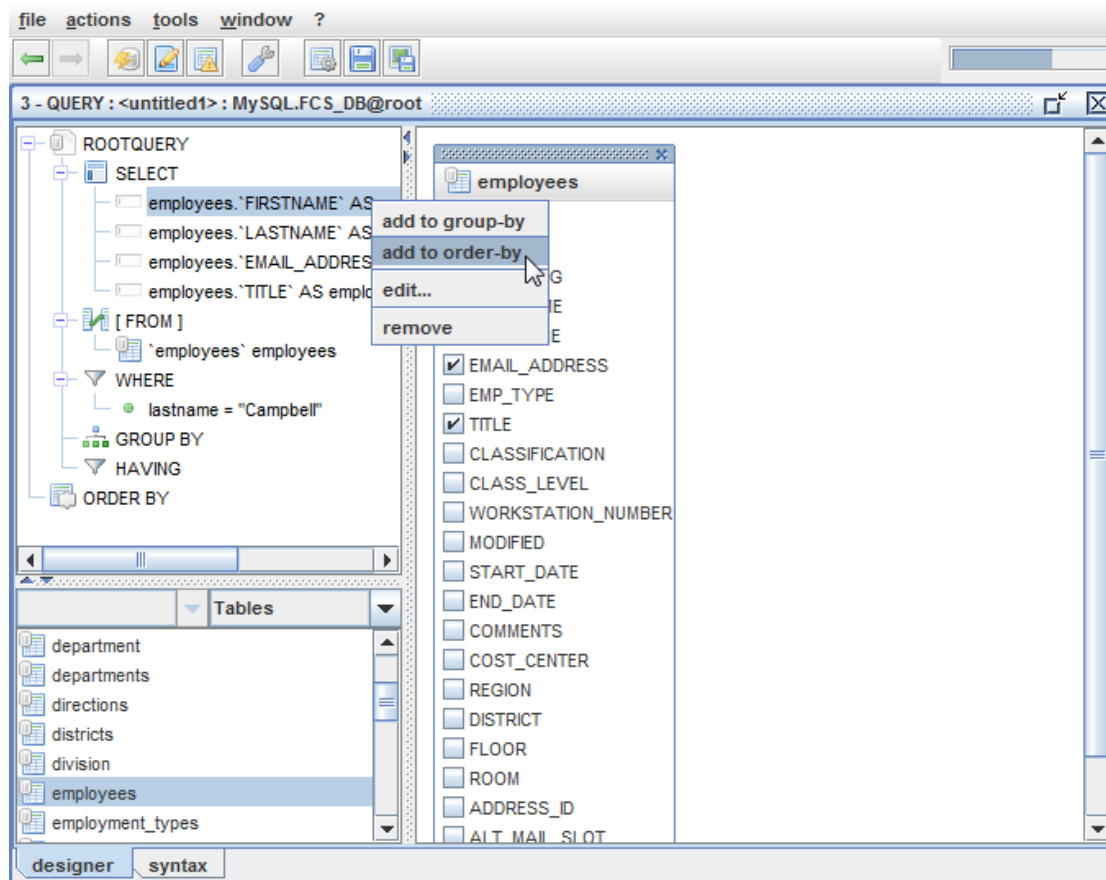


Run the query by clicking on the button with the image ► and labelled *launch query*.

Adding ORDER BY clause to the query

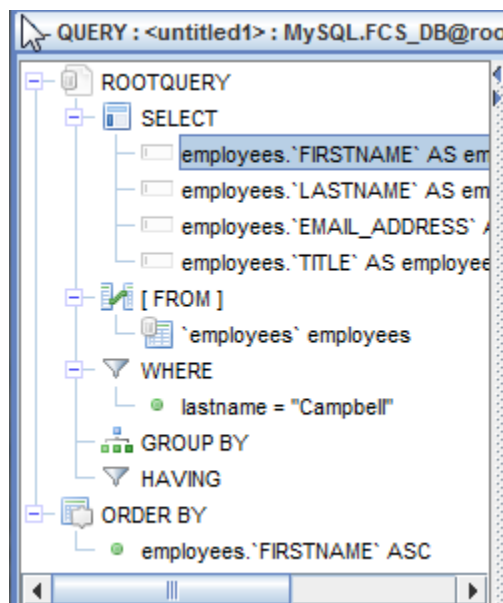
From the tree under the SELECT node, we right click the child node labelled *employees.FIRSTNAME*. A popup menu displays more options. We select **add to order-by**.

IMAGE 25



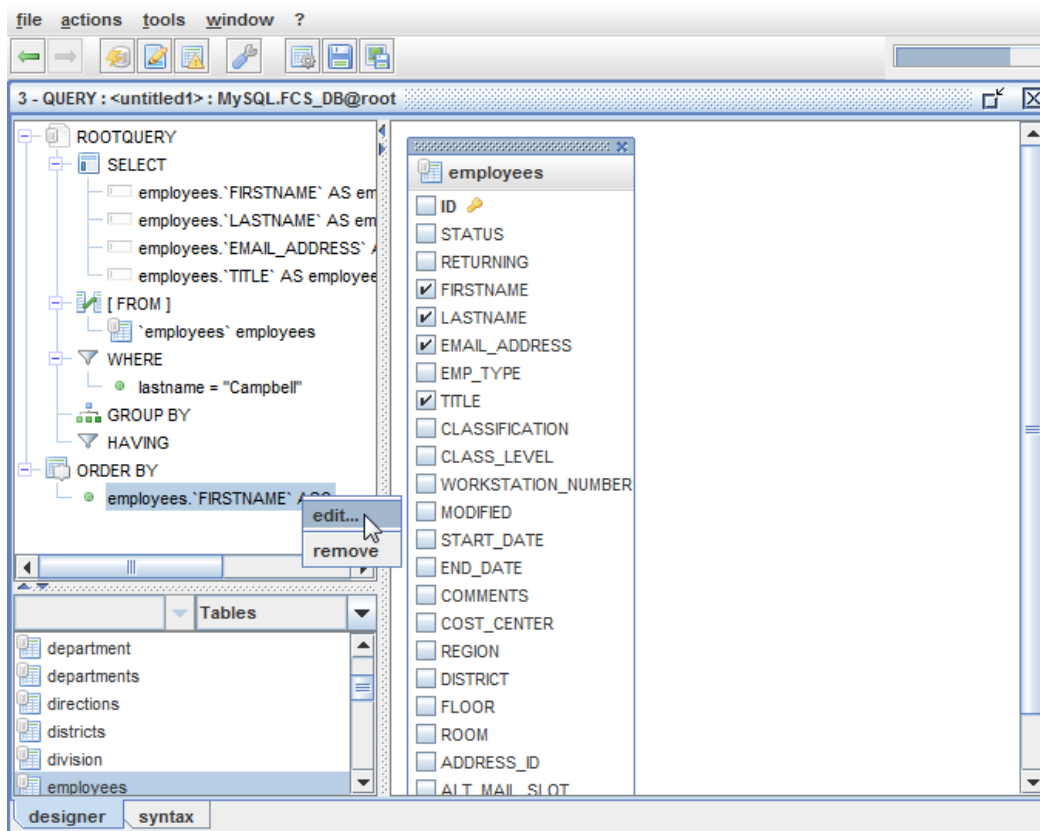
Notice what happens in IMAGE 26. The ORDER BY node now contains the new condition.

IMAGE 26



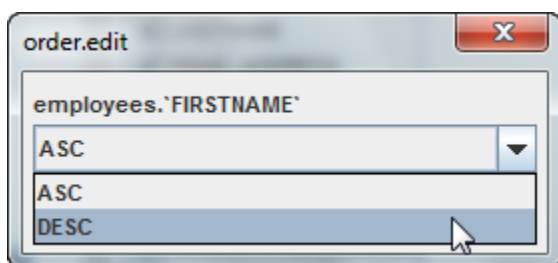
If you look closer, you can see that the ORDER BY condition by default will arrange the resulting records in ascending order. The suffix ASC is displayed. You may have an occasion when you would prefer to display your records in descending order. You can change this if you wish. Right click on the condition as it appears under the ORDER BY node and select the *edit...* option as in IMAGE 27.

IMAGE 27



This will launch a dialog box as in IMAGE 28. From here you can select the descending order option.

IMAGE 28





We now launch the query by clicking on the *launch query* button , we will see that our resulting records are now in alphabetical order by first name. See IMAGE 29.

IMAGE 29

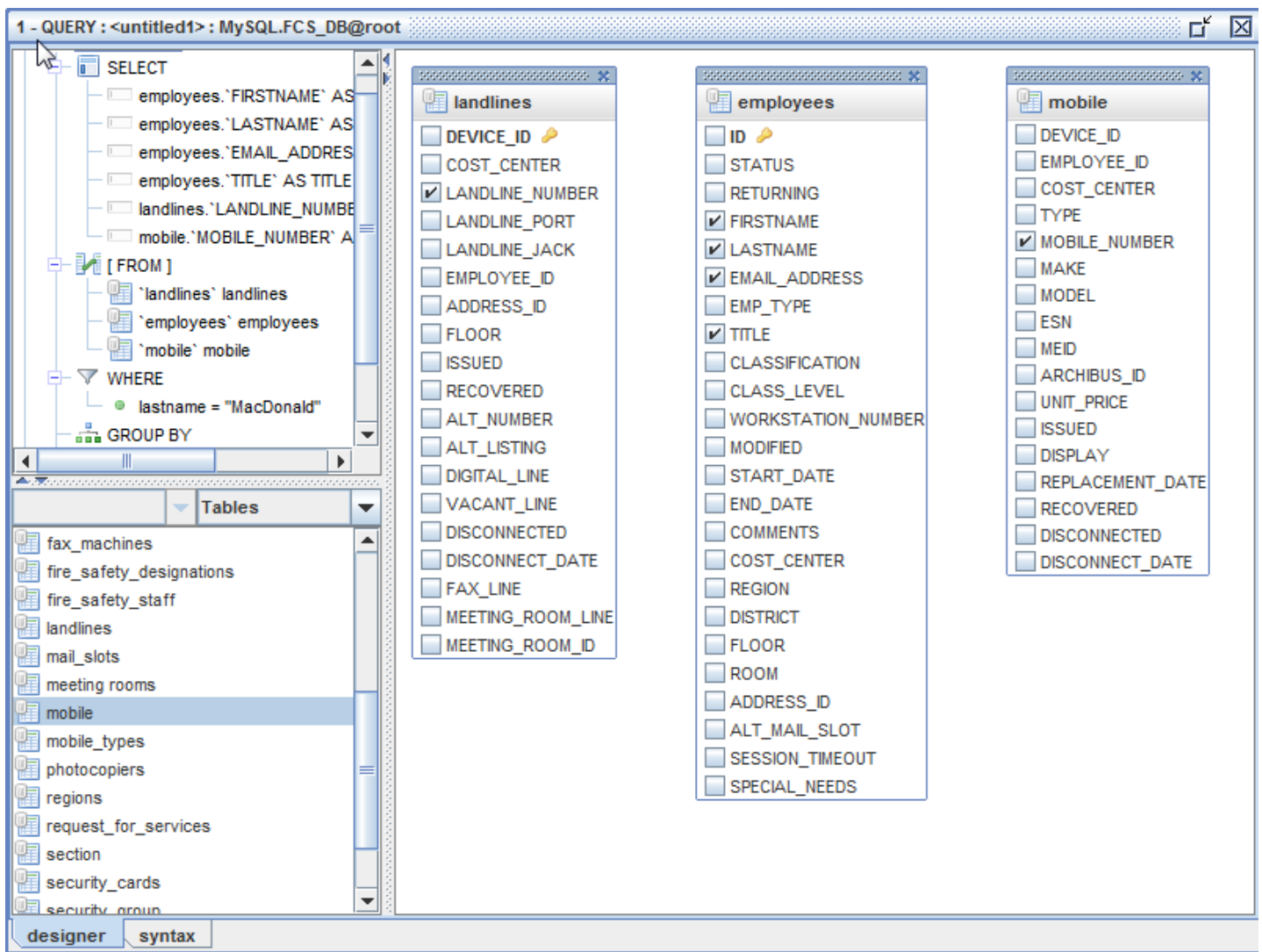
PREVIEW :: <untitled1>: MySQL.FCS_DB@root				
#	employees_FIRSTNAME	employees_LASTNAME	employees_EMAIL_...	employees_TITLE
1	Eleanor	Campbell	eleanor.campbell@...	DIS Business Analyst
2	Lynda	Campbell	lynda.campbell@go...	Manager, Physician Resources
3	Michael	Campbell	michael.campbell@...	Manager, Administrative Services

At this point you have the option to save your query to a file so that you can recall it later. Use the Save Query button . Step through the Save As dialog wizard; provide a name for your query file and save it to a directory of your choosing. By default query files are generally saved using the .sql extension.

Adding More Tables to the Query

To do this, either double-click **landlines** and **mobile** tables as they are listed or drag 'n drop them into the Content pane. When task is completed, all three Tables are displayed in the Content pane. See IMAGE 30.

IMAGE 30



The screenshot shows the SQL query designer interface. The main window title is "1 - QUERY : <untitled1>: MySQL.FCS_DB@root". The SQL editor on the left contains the following query:

```

SELECT
  employees.'FIRSTNAME' AS FIRSTNAME,
  employees.'LASTNAME' AS LASTNAME,
  employees.'EMAIL_ADDRESS' AS EMAIL_ADDRESS,
  employees.'TITLE' AS TITLE,
  landlines.'LANDLINE_NUMBER' AS LANDLINE_NUMBER,
  mobile.'MOBILE_NUMBER' AS MOBILE_NUMBER
FROM
  'landlines' landlines,
  'employees' employees,
  'mobile' mobile
WHERE
  lastname = "MacDonald"
GROUP BY

```

The right pane displays the structure of the three tables involved in the query:

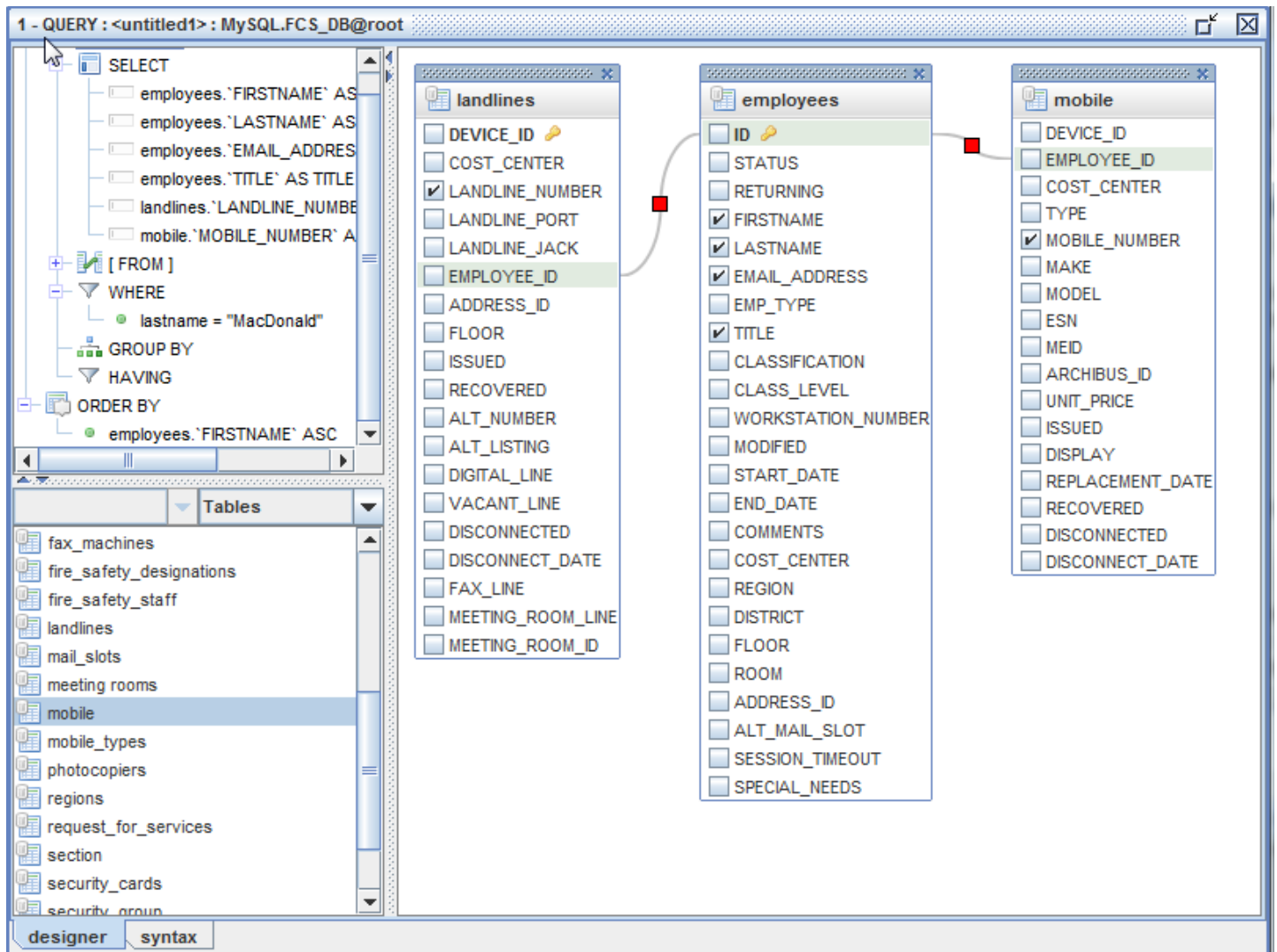
- landlines**
 - ☐ DEVICE_ID
 - ☐ COST_CENTER
 - ☒ LANDLINE_NUMBER
 - ☐ LANDLINE_PORT
 - ☐ LANDLINE_JACK
 - ☐ EMPLOYEE_ID
 - ☐ ADDRESS_ID
 - ☐ FLOOR
 - ☐ ISSUED
 - ☐ RECOVERED
 - ☐ ALT_NUMBER
 - ☐ ALT_LISTING
 - ☐ DIGITAL_LINE
 - ☐ VACANT_LINE
 - ☐ DISCONNECTED
 - ☐ DISCONNECT_DATE
 - ☐ FAX_LINE
 - ☐ MEETING_ROOM_LINE
 - ☐ MEETING_ROOM_ID
- employees**
 - ☐ ID
 - ☐ STATUS
 - ☐ RETURNING
 - ☒ FIRSTNAME
 - ☒ LASTNAME
 - ☒ EMAIL_ADDRESS
 - ☐ EMP_TYPE
 - ☒ TITLE
 - ☐ CLASSIFICATION
 - ☐ CLASS_LEVEL
 - ☐ WORKSTATION_NUMBER
 - ☐ MODIFIED
 - ☐ START_DATE
 - ☐ END_DATE
 - ☐ COMMENTS
 - ☐ COST_CENTER
 - ☐ REGION
 - ☐ DISTRICT
 - ☐ FLOOR
 - ☐ ROOM
 - ☐ ADDRESS_ID
 - ☐ ALT_MAIL_SLOT
 - ☐ SESSION_TIMEOUT
 - ☐ SPECIAL_NEEDS
- mobile**
 - ☐ DEVICE_ID
 - ☐ EMPLOYEE_ID
 - ☐ COST_CENTER
 - ☐ TYPE
 - ☒ MOBILE_NUMBER
 - ☐ MAKE
 - ☐ MODEL
 - ☐ ESN
 - ☐ MEID
 - ☐ ARCHIBUS_ID
 - ☐ UNIT_PRICE
 - ☐ ISSUED
 - ☐ DISPLAY
 - ☐ REPLACEMENT_DATE
 - ☐ RECOVERED
 - ☐ DISCONNECTED
 - ☐ DISCONNECT_DATE

The bottom left pane shows a list of tables in the database, including fax_machines, fire_safety_designations, fire_safety_staff, landlines, mail_slots, meeting_rooms, mobile, mobile_types, photocopiers, regions, request_for_services, section, security_cards, and security_group. The 'mobile' table is currently selected.

Notice that every column from the LANDLINES Table except for the LANDLINE_NUMBER column have been deselected. Also, we have deselected every column from the MOBILE Table except for the MOBILE_NUMBER column. These two columns are all we are interested in adding to our query. Note that we have also changed the criteria on the WHERE clause to: lastname="MacDonald".

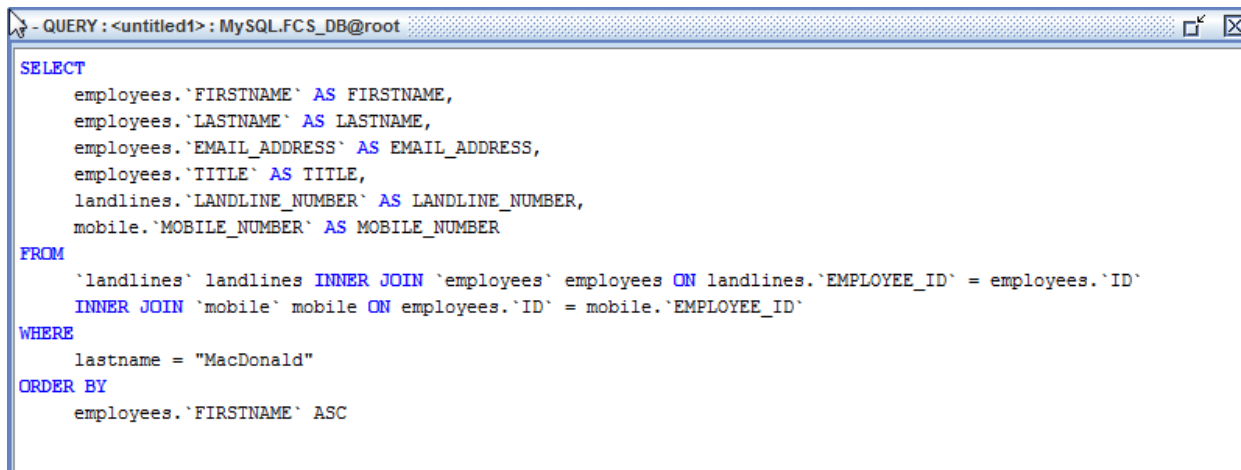
To JOIN the Tables together : perform a little drag 'n drop operation. Using the left mouse button drag 'n drop the EMPLOYEES ID column to the **EMPLOYEE_ID** column on the LANDLINES Table. Do the same operation to the MOBILE Table. When done, we have two links joining the three Tables as seen in IMAGE 31.

IMAGE 31



The EMPLOYEE_ID columns in both the LANDLINES and MOBILE Tables contain integer values referencing every employee record in the EMPLOYEES Table. To see how this looks as an SQL query, we click on the **syntax** tab. See IMAGE 32.

IMAGE 32




Now we will launch the query by clicking on the *launch query* button . See IMAGE 33 for the results.

IMAGE 33

#	FIRSTNAME	LASTNAME	EMAIL_ADDRESS	TITLE	LANDLINE_NUMBER	MOBILE_NUMBER
1	Colin	MacDonald	colin.h.macdonald@g...	Labour Relations Consultant	424-0066	219-1113
2	Douglas Lamont	MacDonald	doug.macdonald@go...	Adult Protection Social Worker	893-5393	897-8356
3	Heather	MacDonald	heather.macdonald@...	Coordinator, Palliative Care	424-2542	237-0049
4	Mary Jean	MacDonald	maryjean.macdonald...	Executive Director	424-4868	237-5535
5	Maureen	MacDonald	maureen.macdonald...	Minister	424-6653	496-3630
6	Michelle	MacDonald	michelle.macdonald@...	A/Manager (Policy Analyst)	424-3573	430-5208
7	Sandy A.	MacDonald	sandy.macdonald@g...	A/Executive Director	424-0128	471-3893

The query results in seven records.

Let us return to IMAGE 31. Note that the JOIN lines connecting the Tables contain a RED square in the center. At any time you may edit the JOINS by using your right mouse button and clicking on the RED square. You will be provided with two menu options: EDIT or REMOVE. If you choose EDIT, you will be presented with a dialog box that allows you to make some refinements on the JOIN criteria.

The DEFINITION Window

IMAGE 5

[illegible]

From IMAGE 5 you can see that the DEFINITION internal window displays details on the *employees* Table providing such information as Column names and their data types: INT, DATETIME, VARCHAR, etc.

You will notice that the DEFINITION internal window has a number of tabs you can select to obtain other information on the *employees* Table. If we select the *primary keys* tab, we can discover which Column has been set as having a Primary Key. In this case, as seen in IMAGE 6, the Column named ID has the Primary Key. Typically one Primary Key is assigned to each Table by the database designer, however not all Tables will necessarily have a Primary Key.

IMAGE 6

5 - DEFINITION : 'employees' : MySQL.FCS_DB@root					
columns (24)		primary keys (1)	indices (1)	exported keys (0)	imported keys (0)
TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	KEY_SEQ	PK_NAME
FCS_DB		employees	ID	1	PRIMARY

If we select the *indices* tab as in IMAGE 7, we can see that the Column ID has not only a Primary Key, but also is indexed to allow for faster searches when the database performs a query on this particular Table. The *exported keys* and *imported keys* tabs provide additional information regarding Reference Primary Keys and Foreign Keys respectively. Not every database system supports these features, which is why you will notice the value zero indicated on the tabs.

TABLE_...	TABLE_...	TABLE_NAME	NON_UNIQUE	INDEX_QUALI...	INDEX_NAME	TYPE	ORDINA...	COLUMN_NAME
FCS_DB		employees	false		PRIMARY	3	1	ID	A	...	0

SQL History

This stores all the SQL commands that have been executed, with a timestamp and the Origin window / datasource. This is useful when such a SQL command has to be recalled.

Data comparer

Description to come.

Schema comparer

Description to come.

Troubleshooting

The first place to look at in case of trouble, is in the command line output after launching SQLLeo with:

Java -jar SQLLeoVQB.jar

Any Java error should be found here and would be useful for development team.

Support

Support and discussion of issues can be obtained from the SourceForge website:

<http://sourceforge.net/p/sqlleo/discussion/>