Using SQL in the Datalab

The aim of this post is to describe why SQL skills are important for IDI users and how to improve skills in SQL coding. We have produced examples of SQL code with commands that are likely to be useful for IDI users.

What is SQL?

SQL (Structured Query Language) is a programming language designed for accessing and manipulating relational databases. The language is best suited to database functions like creating tables, subsetting, joining, grouping, and basic creation of new variables.

Why should I use it?

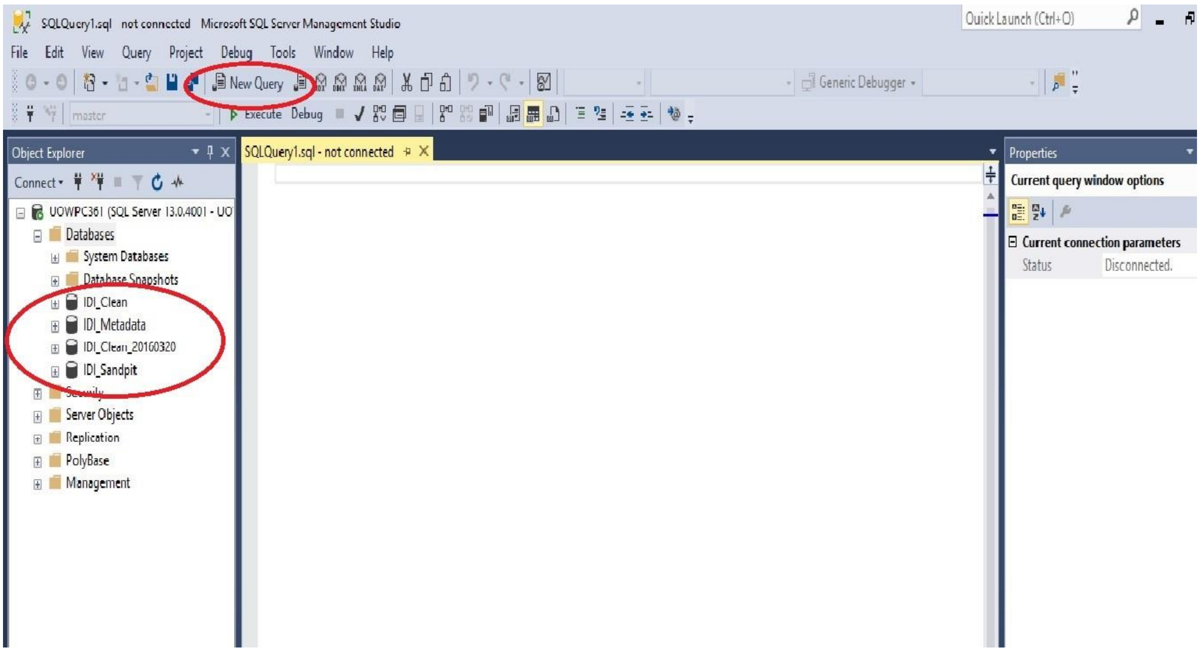
In the IDI datalab environment, programs written in SQL will run faster and use less computing resource. This is particularly useful if you are extracting or joining data from large tables like pharmaceuticals, personal detail, and hospital discharges.

IDI datasets are stored in an SQL database. While it is possible to access these directly from programs like SAS, R and Stata, this process can be slow because it requires data to be moved from the SQL server onto (for example) the SAS server for processing in SAS. If you write your programs in SQL they can be executed on the SQL server without having to move data around.

SQL is not a statistical package. You cannot fit a regression model or create a graph in SQL. To do these things you will need to use another package like R, SAS or Stata. With IDI work a common pattern is to use SQL to create and save a dataset for analysis, and then use R, SAS or Stata for the analysis.

Using SQL server to explore the IDI database structure

One of the simplest and most useful ways to use SQL is to explore the IDI structure. Within the datalab environment we use the program SQL Server Management Studio to access and interact with the available data tables. There is no link to this on the desktop, so search for it on the start screen. After connecting to the server (you will need to use the server name **wprdsql36\ileed**), you will see a window similar to that in Figure 1 (note that this screen shot is not from within the data lab, and is a slightly different version of SQL server).



On the left is the object explorer (circled) which shows the directory structure. Within the IDI, if you expand the *Databases* folder you will see a list of all of the IDI databases. **IDI_Clean** is always the newest data refresh available, and older refreshes have a suffix containing the date of the refresh (e.g. IDI_Clean_20160320). **IDI_Metadata** contains tables of concordances and classifications that can be useful when working with other data sets. These allow you to link, for example, variable descriptions to codes used in IDI_Clean. **IDI_Sandpit** is the location that users can store their own custom tables. **IDI_RnD** is where ad-hoc loaded datasets or other supplementary data might be stored for general or restricted use (historically these were also stored in **IDI_Sandpit**, but now they are being loaded to **IDI_RnD**).

Expanding any one of the databases available will show all of the tables contained with it. By right clicking on any of the tables and selecting *Select Top 1000 Rows* we can view the top 1000 rows of the table, allowing us to view what variables are available in the table and a small sample of their values. Likewise, expanding any of the tables in the object explorer will reveal some sub-folders, and if you expand the one named 'Columns' you can see a list of the column names and data types within that table.

How to run SQL code within SQL Server Management Studio

SQL works similarly to most statistical packages - you can write code (called *queries* in SQL), save these files in the datalab folder, open and edit them at a later date, and run them. To open a blank query (code) file, click the *New Query* button (circled at the top of Figure 1).

Tables that you create in SQL can be saved in your project sandpit (see examples section for how to do this).

Data can be imported and exported from SQL for use in other packages. To do this, right click on IDI_Sandpit, select Tasks, then Export (or Import) Data, then work through the steps in the wizard.

It is also possible to connect directly to an SQL server from R or SAS (see details later in this document).

Where can I get more information?

- The code examples accompanying this guide
- W3school's SQL tutorials: Some simple tutorials that can introduce you to the concepts and functions that you will use with SQL.
- The Ultimate MySQL Bootcamp: this course is not free, but it should only cost $10-15. If you have no experience with SQL it is a thorough introduction. It will cover a reasonable amount of content that you probably won't use in the data lab though.
- TOTN SQL Server: Joins: Introduction and explanation of different joins in SQL.
- This document on Data types in SQL server.
- Check with your institution. Some universities have online SQL courses (for example, Otago Uni)

By Oliver Robertson, Sheree Gibb, Andrea Teng. Thanks to Maddie White for helpful comments on an earlier draft.

Original post 14 August 2018