

# Querying a Customer Database

As an example of using a “real” database, in this exercise you will run some queries on a large table of survey results. To do this exercise you will need to have installed a browser and editor for SQLite databases. See the instructions on Blackboard for installing either the *DB Browser for SQLite* standalone application or the *SQLite Manager* add-on for Firefox.

## A. Import the Customer Data

Accompanying these instructions you will find an SQLite script `customers.sql`. Create a database file `customers.db` and populate it by running the script. In the *DB Browser for SQLite* application this can be done in two separate steps, but an easier way is to choose the ‘Import database from SQL file’ option and follow the prompts. This script will create a large table called `customers` in your database file.

(There are minor differences in different versions of the SQLite language. Just in case you have any problems running the SQLite script provided we have also supplied a copy of the `customers.db` file, but it would be better for you to run the script yourself to see what it contains.)

## B. Browse the Customer Data

Browse the data in the `customers` table. This table contains information describing over 3,000 people who completed a survey. The data has been anonymised by using a ‘customerID’ field instead of the respondents’ names. Familiarise yourself with the kind of data stored in the table.

## C. Query the Customer Data

Now imagine that you want to find out some things about the people in the `customers` table. Given its large size, it would obviously be impractical to browse through it manually. Therefore, you will need to write some SQLite queries (**select** statements) to solve the following problems. In the *DB Browser for SQLite* application this is done via the ‘Execute SQL’ tab.

Write and execute **select** queries to find solutions to the following problems.

1. Show all details of customers who are 60 or older.
2. How many customers in the survey have a doctorate? (You can answer this question just by asking for all customers with this qualification and noting the number of rows returned. Alternatively, you can apply SQLite’s **count** function to the list of columns to be returned, in which case the result produced will be the number of rows.)
3. Show all details of customers who are male and divorced.
4. Show just the ages of male, divorced customers, ordered from youngest to oldest.
5. Show the average age of male, divorced customers. (You can do this by applying SQLite’s **avg** function to the column returned by your previous query.)
6. Show the education level and number of cars of all customers with either a Master or Doctorate degree, ordered by number of cars from highest to lowest.

7. Show all details of the female customers who have a doctorate and have never married or have three or more cars.
8. Show the customer ids and marital status of all customers with six or more cars.
9. Choose one of the customers returned by the previous query and show all the details for just that person.
10. Show the customer ids, ages and genders for all customers who have never married but have children.
11. Find the youngest age of any customer who has a doctorate.
12. Determine the average number of cars owned by any customer who has a higher degree (master or doctorate).