

DATA514 Homework 1: SQLITE and SQL Basics

Objectives: To be able to create and manipulate tables in sqlite3, and write simple queries using SQL.

Assignment tools: [SQLite 3](#)

Assigned date: Jan. 08, 2019

Due date: Jan 14. You have 1 week for this homework.

Questions: Make sure you post them on the [discussion board](#).

What to turn in:

One file per each of the question below, containing your SQL and SQLite commands, and SQL comments for your responses that are not in SQL (i.e., submit a `.sql` file that can be executed directly against the database system).

No need to include any inputs or outputs. Name each file `hw1-q1.sql`, `hw1-q2.sql`, etc. You will need to learn how to write SQL comments if you have not done so before.

Motivation: We will use [SQLite](#) for this assignment. SQLite is a software library that implements a SQL database engine. We will use SQLite in this assignment because it offers an extremely lightweight method to create and analyze structured datasets (by structured we mean datasets in the form of tables rather than, say, free text). Using SQLite is a minimal hassle approach to realizing the benefits of a relational database management system.

Of course, SQLite does not do everything, but we will get to that point in later assignments. In the meantime, you can also learn [when to use SQLite and when not to use it](#).

Resources:

- Some important SQLite commands:
 - To view help contents: `.help`
 - To view a list of all your tables: `.tables`
 - To exit: `.exit`
- [A simple guide](#) for commonly used command-line functions in SQLite.
- [More information](#) on formatting output in SQLite.
- [An index](#) of more detailed information for SQL commands in SQLite.
- A [SQL style guide](#) in case you are interested (FYI only).

Assignment Details

To run SQLite do the following:

- Mac OS X or Linux: open a terminal and type `sqlite3` (if installed)
- Windows: there are two reasonable options:

- Install the stand-alone windows program from the [SQLite web site](#) (precompiled windows command-line shell on the [download](#) page)
- (maybe a bit more complicated): Install [cygwin](#) to get a linux command shell, then open cygwin and type `sqlite3` (you may have to install it by running `setup → database → sqlite3`; it is probably already installed in the CSE labs).

Note that the course staff will only support the linux version of sqlite3 running on CSE machines (e.g., [attu](#)) and will also use that version for grading purposes. *Make sure your submissions runs there.*

Problems

1. (20 points) First, create a simple table using the following steps:

- Create a table `Edges(Source, Destination)` where both `Source` and `Destination` are integers.
- Insert the tuples `(10,5)`, `(6,25)`, `(1,3)`, and `(4,4)`
- Write a SQL statement that returns all tuples.
- Write a SQL statement that returns only column `Source` for all tuples.
- Write a SQL statement that returns all tuples where `Source > Destination`.
- Now insert the tuple `(-1, '2000')`. Do you get an error? Why? This is a tricky question, you might want to [check the documentation](#).

2. (15 points) Next, you will create a table with attributes of types integer, varchar, date, and Boolean.

However, SQLite does not have date and Boolean: you will use `varchar` and `int` instead. Some notes:

- 0 (false) and 1 (true) are the values used to interpret Booleans.
- Date strings in SQLite are in the form: 'YYYY-MM-DD'. Examples of valid date strings include: `'1988-01-15'`, `'0000-12-31'`, and `'2011-03-28'`.
Examples of invalid date strings include: `'11-11-01'`, `'1900-1-20'`, `'2011-03-5'`, and `'2011-03-50'`.
- Examples of date operations on date strings (feel free to try them): `select date('2011-03-28');`
`select date('now');`
`select date('now', '-5 year');`
`select date('now', '-5 year', '+24 hour');`
`select case when date('now') < date('2011-12-09') then 'Taking classes' when date('now') < date('2011-12-16') then 'Exams' else 'Vacation' end;`
What does this query do? (no need to turn in your answer)
Create a table called `MyRestaurants` with the following attributes (you can pick your own names for the attributes, just make sure it is clear which one is for which):
 - Name of the restaurant: a `varchar` field
 - Type of food they make: a `varchar` field
 - Distance (in minutes) from your house: an `int`
 - Date of your last visit: a `varchar` field, interpreted as date
 - Whether you like it or not: an `int`, interpreted as a Boolean

3. (13 points) Insert at least five tuples using the SQL `INSERT` command five (or more) times. You should insert at least one restaurant you liked, at least one restaurant you did not like, and at least one restaurant where you leave the "I like" field `NULL`.

4. (13 points) Write a SQL query that returns all restaurants in your table. Experiment with a few of SQLite's output formats and show the command you use to format the output along with your query:
 - print the results in comma-separated form
 - print the results in list form, delimited by " | "
 - print the results in column form, and make each column have width 15
 - for each of the formats above, try printing/not printing the column headers with the results
5. (13 points) Write a SQL query that returns only the name and distance of all restaurants within and including 20 minutes of your house. The query should list the restaurants in alphabetical order of names.
6. (13 points) Write a SQL query that returns all restaurants that you like, but have not visited since more than 3 months ago.
7. (13 points) Write a SQL query that returns all restaurants that are within and including 10 mins from your house.

Collaboration

All DATA 514 assignments are to be completed **INDIVIDUALLY!** However, you may discuss your high-level approach to solving each lab with other students in the class.

Submitting your assignment

Put your `.sql` files in `hw1/submission`. Your directory structure should look like this after you have completed the assignment:

```
\-- hw1
  \-- hw1.pdf      # this is the file that you are currently reading
  \-- submission
    \-- hw1-q1.sql # your solution to question 1
    \-- hw1-q2.sql # your solution to question 2
    \-- hw1-q3.sql # your solution to question 3
    ...
```

Make a zip/tarball of the hw1 directory and submit the zipped/tarball file.