

Q1 Assignment Setup

0 Points

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

- [Ed discussion board](#)
- 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

You will submit SQL commands that satisfy the requirements described by each question. Utilize SQLite to test your commands before submitting them here.

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](#) (the "bundle of command-line tools" option from the precompiled windows binaries part of 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). This is the more powerful option because you can run other command line tools later. If you install cygwin, make sure to check all the boxes for sqlite3. You may also choose to get python and vim which are nice to have!

Q2

20 Points

First, create a simple table using the following steps:

Q2.1

3 Points

Create a table Edges(Source, Destination) where both Source and Destination are integers.

Q2.2

3 Points

Insert the tuples (10,5), (6,25), (1,3), and (4,4)

Q2.3

3 Points

Write a SQL statement that returns all tuples.

Q2.4

3 Points

Write a SQL statement that returns only column Source for all tuples.

Q2.5

3 Points

Write a SQL statement that returns all tuples where Source > Destination.

Q2.6

5 Points

Now write a statement that inserts the tuple ('-1','2000'). Do you get an error? Why?
This is a tricky question, you might want to check the [documentation](#).

Q3

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

## Q4

13 Points

Write the commands to 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.

## Q5

13 Points

Write a SQL query that returns every row in your MyRestaurants table. Experiment with a few of SQLite's output formats and write this query six times, formatting the output in the following ways.



Remember to include both the command you use to format the output along with your query. When we run your code for 5.1 and 5.2 we should see the table printed 6 times.

## Q5.1

11 Points

Write the code to turn column headers on, then output the results in these three formats:

1. print the results in comma-separated form
2. print the results in list form, delimited by "|"
3. print the results in column form and make **every** column have width 15 (be sure that every column has width 15 and not just the first one)

## Q5.2

2 Points

Now write the code to turn the column headers off, and output the results again in the three formats.

## Q6

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.

## Q7

13 Points

Write a SQL query that returns all restaurants that you like, but has not visited since more than 3 months ago. Make sure that you use the date() function to calculate the date 3 months ago.

**Q8**

13 Points

Write a SQL query that returns all restaurants that are within and including 10 mins from your house.