Cal Poly Spring 2013	**CPE/CSC 365** Introduction to Database Systems	Eriq Augustine

Lab 4: SQL Installations

Due date: Mat 1st.

Lab Assignment

Assignment Preparation

This is an individual lab. Each student has to complete all work required in the lab, and submit all required materials **exactly as specified** in this assignment.

Note on machines. For this lab, you will need your own personal machine and will not be able to complete it directly on the lab machines. If you do not have access to a machine that you have root (admin) on (any OS should be fine), then talk to me and we can work something out. Since this lab cannot be done on the lab machines, I suggest that you spend lab time working on the relational algebra assignments.

Tasks

In this lab, you will have to install some database systems on your machine and then populate it with our class datasets. For each installation, copy down the steps that you take in your lab writeup. If you already have any of these systems already installed on your machine, then put down the steps that you would have to, or did, take. If you want to be more adventurous, then you should try installing from source. (Not recommended for Windows users).

Since this lab is highly personalized, I suggest that you throughly research any problems that you are encountering before you contact me for help.

MySQL Installation. Install a copy of MySQL on your system. As discussed in class, there are three major flavors of MySQL to choose from: Orcale MySQL, MariaDB, and Percona MySQL. Our "official" class flavor is MariaDB, but they are similar enough that you can choose whichever one you want.

MySQL is a client/server based database system. (PostgreSQL is also a client/server based database system). This means that there is a server that handles the database, and a client that lets a user connect to the database from anywhere. The server can be some machine running somewhere in cloud, a csl machine (like our class server), or even your local machine. In your MySQL connection command that was given to you along with your passwords, the "-h csc-db0.csc.calpoly.edu" portion is the option that sets the database server that you connect to. ("h" stands for "host"). Leaving off this option will cause the MySQL client to try and connect to the local databse.

Loading and testing the MySQL Dataset. Load the class datasets (the normalized ones from Lab03), into your new MySQL database. This should be fairly straightforward because the datasets are already made for MySQL. The data can be loaded with the provided `build-all.mysql` script.

After all the data is loaded, run the `test-all.mysql` script and put the output into a file called `mysql.counts`.

SQLite Installation. Install a copy of SQLite on your system. (I recommend at least version 3.0). SQLite is very different than MySQL because it does not require a running server. SQLite will just use a single file as a database. This comes in handy when you need to move around your database a lot.

Loading and testing the SQLite Dataset. Load the class datasets (the normalized ones from Lab03), into your new SQLite database. **WARNING:** the unmodified SQL files will not work. MySQL and SQLite are not 100% compatible. You WILL have to modify the SQL files to work in SQLite.

Also make sure to modify the `.mysql` files to work with SQLite (change the extension to `.sqlite`).

HINT: The SQL files use a batch insert (notice that there is only one INSERT statement per table) that SQLite does not support. The easiest (although slowest running) solution to this would be to just expand the one batch insert into many normal inserts. I recommend that you use a text editor with strong find/replace capabilities like Vim or Emacs.

As with the MySQL section, run the newly renamed `test-all.sqlite` script and put the output into a file called `sqlite.counts`.

Deliverables

SQLite Scripts The SQLite version of the dataset files. These files should keep the same directory structure as how they are distributed (see the Example Directory Structure section).

Counts These files were described in the "Loading and testing" sections for MySQL and SQLite. They are the results of running the `test-all.mysql` and `test-all.sqlite` files. They should be called `mysql.counts` and `sqlite.counts`.

Writeup Because this lab has few intellectual components, I will be expecting a writeup that includes the following:

Operating System information. Include what operating system and version you are using. If you are using Linux/UNIX or Mac, the specific flavor (Ubuntu, Arch, MacOSX) and the results of running "uname -a" should be sufficient.

MySQL and SQLite version strings. The output of "mysql --version" and "sqlite3 --version".

Installation Steps. The steps you took to get MySQL and SQLite installed on your system. Be detailed enough so that any of your peers (even if they are not in CSC365) can understand and follow them.

Difference between MySQL and SQLite. List some of the differences that you have noticed (or researched) about MySQL and SQLite.

A short discussion on when to favor MySQL over SQLite and visa-versa.

Any problems you had installing/running any of the database systems in this lab.

Any final comments about this lab or any database system.

Submission Instructions

Please, follow these instructions exactly. Up to 20% of the Lab 04grade will be assigned for conformance to the assignment specifications, **including the subimssion instructions**.

Please, **name your files exactly as requested** (including capitalization). Correct submission simplifies grading, and ensures its correctness.

Please include your name and Cal Poly email address in all files you are submitting. If you are submitting code/scripts, include, at the beginning of the file a few comment lines with this information. Files that cannot be authenticated by observing their content will result in penalties assessed for your work.

Specific Instructions

You must submit all your files in a single archive. Accepted formats are `tarball (gzipped tar)` (`.tar.gz`) or `zip` (`.zip`). The file you are submitting must be named `lab04.ext`, i.e., either `lab04.tar.gz` or `lab04.zip`

Inside it, the archive shall contain a single directory named `sqlite`, the lab writeup (`writeup.txt` or `writeup.pdf`), the count files (`mysql.count` and `sqlite.count`), and your readme (`README.txt`.

As with lab04, it is **INCORRECT** for a submission to unpack into a single directory which contains the actual submission.

Example Directory Structure

You archive should unpack into the same structure outlined below:

```
.
|-- sqlite
|   |-- AIRLINES
|   |   |-- AIRLINES-cleanup.sql
|   |   |-- AIRLINES-insert.sql
|   |   |-- AIRLINES-setup.sql
|   |   `-- AIRLINES-test.sql
|   |-- BAKERY
|   |   |-- BAKERY-cleanup.sql
|   |   |-- BAKERY-insert.sql
|   |   |-- BAKERY-setup.sql
|   |   `-- BAKERY-test.sql
|   |-- CARS
```

```
|   |   |-- CARS-cleanup.sql
|   |   |-- CARS-insert.sql
|   |   |-- CARS-setup.sql
|   |   `-- CARS-test.sql
|   |-- CSU
|   |   |-- CSU-cleanup.sql
|   |   |-- CSU-insert.sql
|   |   |-- CSU-setup.sql
|   |   `-- CSU-test.sql
|   |-- INN
|   |   |-- INN-cleanup.sql
|   |   |-- INN-insert.sql
|   |   |-- INN-setup.sql
|   |   `-- INN-test.sql
|   |-- MARATHON
|   |   |-- MARATHON-cleanup.sql
|   |   |-- MARATHON-insert.sql
|   |   |-- MARATHON-setup.sql
|   |   `-- MARATHON-test.sql
|   |-- STUDENTS
|   |   |-- STUDENTS-cleanup.sql
|   |   |-- STUDENTS-insert.sql
|   |   |-- STUDENTS-setup.sql
|   |   `-- STUDENTS-test.sql
|   |-- WINE
|   |   |-- WINE-cleanup.sql
|   |   |-- WINE-insert.sql
|   |   |-- WINE-setup.sql
|   |   `-- WINE-test.sql
|   |-- build-all.sqlite
|   |-- cleanup-all.sqlite
|   `-- test-all.sqlite
|-- README.txt
|-- mysql.counts
|-- sqlite.counts
`-- writeup.txt
```

Handin

Once you created your submission archive, submit it using the following `handin` command:
$ handin eaugusti lab04 <ARCHIVE>