COMPSCI 410/510 Databases (Spring 2015)

Programming Assignment #0, Due on 1/28/2015, Wednesday (11PM)

50 points

Description:

This assignment will ask you to set up your own copy of `MySQL`. You will be the DBA of your own copy. To do so, you need to run a script on the command line

```
mysql-setup-local port#
```

After you create your own copy of `MySQL`, You need to create a database UNIVERSITY (as attached), and load data into it.

For consistency, I'll provide you a file `university-schema.sql` to define all tables, which can be found in `onyx` under the directory

~vijaydialani/cs410/labs/lab1/files/

After the tables are defined, please accumulate data into the database by inserting tuples to all tables. Again, for consistency, I'll provide you 7 data files (`*.unl`), one for each table. These files can be found in `onyx` under the directory

~vijaydialani/cs410/labs/lab1/files/

To accumulate data, please do not insert tuples one at a time; instead, please use the `LOAD DATA` utility. Example script `load-data-company.sql` describing the usage of the utility can be found in `onyx` under the directory

~vijaydialani/cs410/labs/lab1/files/

This example script is used to load data for the `COMPANY` database. Modify the example script to `load-data-university.sql` to load data for the UNIVERSITY database. The attached page is the university database schema.

Important! Important! Important!

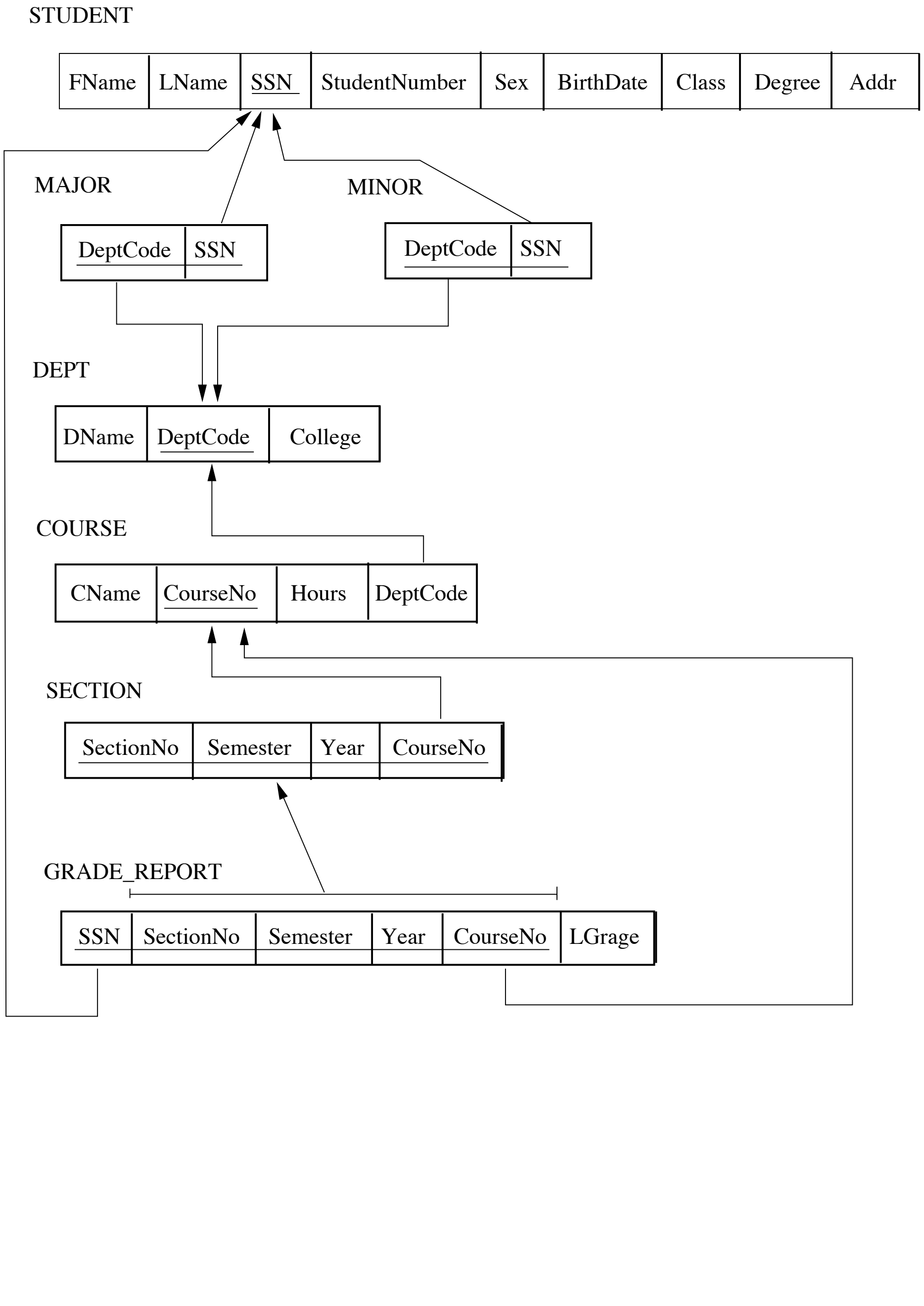
In order for us to execute queries within your `university` database, please add an account "public", only allowing connection from "localhost" with the least privileges. The password for this "public" account is "12345678". An example for creating the user is provided at http://dev.mysql.com/doc/refman/5.1/en/create-user.html.

Syntax: CREATE USER 'public'@'localhost' IDENTIFIED BY '12345678';

Please do not grant un-necessary privileges to "public". Otherwise, someone may maliciously mess up your work.

Submission

Nothing to submit. We'll log in your database as a public user to check yout work.



STUDENT

FName	LName	SSN	StudentNumber	Sex	BirthDate	Class	Degree	Addr

MAJOR

DeptCode	SSN

MINOR

DeptCode	SSN

DEPT

DName	DeptCode	College

COURSE

CName	CourseNo	Hours	DeptCode

SECTION

SectionNo	Semester	Year	CourseNo

GRADE_REPORT

SSN	SectionNo	Semester	Year	CourseNo	LGrage

MySQL Lab Setup

Introduction:

Each user will run their own MySQL database server. So you will act as the Database administrator, Database programmer and Database user! The various MySQL database servers will be kept separate by using a different TCP/IP port and a different socket for each user. For the rest of this document, we will assume that the login name for our user is dbkool, and that their assigned port is 4606. The socket that is created will be /tmp/mysqld-dbkool.sock, where dbkool is your login name.

The MySQL database will only run on onyx. However you will be able to connect it from any machine in the lab (as well as from home by tunneling via Secure Shell).

Setup:

1. Get a port number in the range 4609-4657 from your **TA (Andrew Gable)**. Suppose you have been assigned port 4606.
2. Log in to onyx.
3. Run the local setup script with your assigned port number as an argument.
   mysql-setup-local 4606
   The script will remove any existing MySQL directory from your account and re-initialize a new setup. It will also ask for a root/user password for the MySQL database. Please use a good password (but not the same as the one for your login account). For the curious, the script is located at /usr/local/bin/mysql-setup-local.
4. The database main directory will be $HOME/mysql. You will have full access to it, so be careful if you go peeking around there!
5. Note that the mysql-setup-local script will modify your $HOME/.bash rc file. It also adds a file $HOME/.mysqlrc that is invoked when you login to set up MySQL environment variables.
6. Note that the setup script also creates a config file $HOME/.my.cnf that contains your password. If you ever change your password in MySQL, you will need to change this file as well. This file is readable only by you. Having this file eliminates having to type your password again and again for certain MySQL clients.
7. Using MySQL GUI tool mysql-workbench to create access accounts with different privileges. Don't forget to add an account for your instructor and give them appropriate privileges.
8. The following commands are available after the setup script is run.
   - mysql-check: Check if the database server is running.

- mysqlshow: Check what tables are in your database.
- mysql-stop: Stop the database server.
- mysql-start: Start the database server.

9. What if I want to setup MySQL again in my account from scratch? Just remove the lines added by the setup-script in .bashrc. Then run the mysql-setup-local script again as shown earlier.

10. How do I restart the MySQL server. Stop the server with the command mysql-stop and then start the server with the command mysql-start.

11. Does the MySQL database server keep running when I log out? Yes. You should normally leave the server running at all times so you can connect from home or other places.

Using MySQL Workbench GUI:

After you run the local setup script and have your MySQL server running, you can use the GUI tool mysql-workbench to administering user accounts, creating database model and adding data to your databases. Please following the step-by-step tutorial on the following MySQL websites.

https://www.youtube.com/watch?v=X_umYKqKaF0
http://downloads.mysql.com/docs/workbench-en.pdf

Remote Access to MySQL

Normally MySQL runs on port 3306, which is blocked by the firewall (a good thing). Your MySQL on `onyx` will be running on some other port in the range `4609-4657`, which are also blocked by the firewall (another good thing). However, you can still use the tools `mysql-query-browser` and `mysql-administrator` from home to connect to your server on `onyx` by tunneling via Secure Shell.

First you will need to use the `mysql-administrator` and add the host `onyx.boisestate.edu` under your user name (by right-clicking on the user name and then choosing the *Add Host* option) and grant the appropriate privileges.

Linux

Suppose your assigned port is 4606 and login id is `dbkool`. Then login to onyx with the following command.

```
ssh -L 4606:onyx:4606 dbkool@onyx
```

Now tunneling has been established. Use `mysql-query-browser` or `mysql-administrator` and specify the host address as `127.0.0.1` (instead of `localhost`), specify the correct port and socket file name and viola you are now sitting at home but using the MySQL server at `onyx`. If you get "access denied" message, then you need to use the `mysql-administrator` and add the host `onyx.boisestate.edu` under your user name and grant the appropriate privileges. And yet our system admin is happy since he does not have to worry about holes in MySQL. How cool is that?

MS Windows

This depends upon which ssh client you are using. Tunneling can be setup with either the client from ssh.com or with putty. See this article on how to set it up:

http://quintagroup.com/services/support/tutorials/mysql-windows

Once you have established tunneling, then use mysql-query-browser or mysql-administrator and specify the host address as 127.0.0.1 (instead of localhost), specify the correct port and socket file name and viola you are now sitting at home but using the MySQL server at onyx.

Tunneling needs to be enabled for MySQL Workbench to access MySQL Server running on onyx. Make sure you provide *localhost* as the server name and your allocated port number to access the MySQL server on onyx.

Using MySQL

Using the GUI. Use MySQL workbench to run queries.

Using the command line interface. The client mysql gives you a command line interface to MySQL. This is convenient if you are coming over the network to onyx. It can also be used to run SQL statements in batch mode. For example, if you have all your SQL statements typed in a file, say queries.sql, then you can run them with the command source queries.sql; in the mysql command line interface. See an example session below.

```
[dbkool@onyx soln]$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 4.1.10-standard-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use company;
Reading table information for completion of table and column names You
can turn off this feature to get a quicker startup with -A

Database changed
mysql> source
queries.sql; ...
<lots of output>
...
mysql> show warnings;
...
```

```
<hopefully, not much
output :-)> ...
mysql> quit
```

You can even redirect them in as:

```
mysql < queries.sql
```

Dumping your database in SQL. Use the mysqldump command to export your entire database into SQL statements. This can be very convenient in migrating your database to another database server. An example is shown below.

```
[dbkool@onyx dbkool] mysqldump company > company.sql
```

Making a hot copy of your database. The term "hot copy" refers to making a consistent copy of a database that is being used. Use the mysqlhotcopy command on onyx. An example is shown below.

```
[dbkool@onyx dbkool] mysqlhotcopy company mysql.backup/
Locked 6 tables in 0 seconds.
Flushed tables (‘company‘.‘department‘, ‘company‘.‘dependent‘,
‘company‘.‘dept_locations‘, ‘company‘.‘employee‘, ‘company‘.‘project‘,
‘company‘.‘works_on‘) in 0 seconds.
Copying 19 files...
Copying indices for 0 files...
Unlocked tables.
mysqlhotcopy copied 6 tables (19 files) in 0 seconds (0 seconds overall).
[dbkool@onyx dbkool]
```