CS 166: Lab01 Assignment

April 1, 2019

This is an introductory lab to help you familiarize yourself with the tools that will be used throughout the course.

1 Initialize PSQL Environment

Following, we describe the steps associated with initializing our execution environment.

1. Execute the following command to initialize the PSQL environment

   ```
   source ./startPostgreSQL.sh
   ```

   Note: Examine the script you just executed. Look at every command and try to figure out its functionality. Pay special attention to the *PGPORT* variable. After executing the above script type *pg_ctl status* to view the status of the server, it should indicate that it is running correctly.

2. Execute the following command to create your database

   ```
   source ./createPostgreDB.sh
   ```

   Note: Examine the script you just executed. Look at every command and try to figure out its functionality. What is the name of the database you just created?

3. Once you finished with the whole assignment, **DO NOT FORGET!** to call the following command to stop the server and shutdown the database.

   ```
   source ./stopPostgreDB.sh
   ```

   Note: Examine the script you just executed. Look at every command and try to figure out its functionality.

2 Execute SQL Statements

After initializing your environment you should execute a series of SQL statements. **DO NOT OPEN!** a new terminal window, the scripts you just executed rely on system variables initialized through the previous scripts. Opening a new window will require initializing the values from scratch, hence running each script again after stopping the database correctly.

First, you will use the interactive environment PSQL environment to execute some SQL statements.

1. Type the following command to launch the PSQL interactive environment

   ```
   psql −h localhost −p $PGPORT $USER"_DB"
   ```

   We use $ to specify the value of the system variable with the corresponding name. In this case, $USER is your username. Alternatively, you can type the values directly if you know them. For example, if your username is *vzois001* and the port number is set to 8192 then the command should look like:

   ```
   psql −h localhost −p 8192 "vzois001_DB"
   ```

2. In the terminal type the following statement to create a table with name students

   ```
   CREATE TABLE Students (SID numeric (9, 0), Name
      ↪ text, Grade float);
   ```

3. Type \dt to view a list of all tables in the database. You should be able to see the table you just created.

4. Insert a single row in the table using the following statement

   ```
   INSERT INTO Students VALUES (860507041, 'John
      ↪ Anderson', 3.67);
   ```

   This statement will create a record in the table Students for a new student with name John Anderson, SID 860507041 and GPA 3.67.

5. Insert a single row in the table using the following statement

   ```
   INSERT INTO Students VALUES (860309067, 'Tom Kamber
      ↪ ', 3.12);
   ```

This statement will create a record in the table Students for a new student with name Tom Kamber, SID 860309041 and GPA 3.12.

6. Execute a query using the following command

```
SELECT SID, Name, Grade FROM Students WHERE SID =
    ↪ 860507041;
```

This statement will retrieve all records from the table Students which satisfy the condition that the column SID has value 860507041.

7. Try to insert a new student in the table with name George Haggerty SID = 860704039 and GPA = 3.67.

8. Try to retrieve all records from the table which have GPA = 3.67.

9. Exit from the PSQL terminal (type \q).

10. Using the text editor (gedit), create a .sql file containing all of the previous statements (except \dt). At the top of the file include the following statement

```
DROP TABLE IF EXISTS Students;
```

This statement is often used at the beginning of each script to avoid errors when re-initializing the tables.

11. Use the following command to execute all statements in the .sql file you just created

```
psql −h localhost −p $PGPORT $USER_"DB" < script.
    ↪ sql
```

Replace "script" with the name of your script.

12. Shutdown the database using the appropriate script (look at previous section).

13. Submit the script you created in iLearn.