Objectives:

Upon completing this exercise, you will know how to find and use MySQL's command-line interface tool and learn about your database environment. You will also begin to use the interface to extract information from a database and to produce simple reports. The knowledge you gain from this lab will prepare you for the more advanced work you will encounter in subsequent labs.

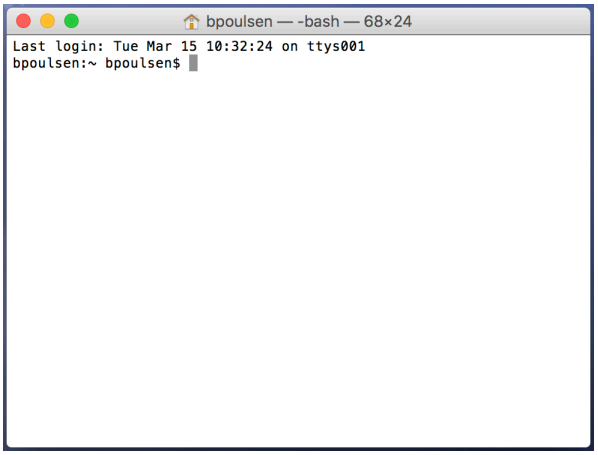
Part I - Introduction

What is a *relational database*? In simple terms, a relational database is a set of data organized in *tables, which* consist of *columns* (fields) and *rows* (records). The data contained in a table is very often *related* to the data contained in other tables. The separation of related data into different tables occurs during the process of *normalization*. Related data are normalized to provide greater flexibility and efficiency than would be possible otherwise. You will begin to learn how to display the related data of different tables in the labs later in the semester.

Displaying data from the database occurs through a process known as *querying* the database. Queries are an important component of the *Structured Query Language (SQL)*. You will learn about queries and about the other components of SQL in this course.

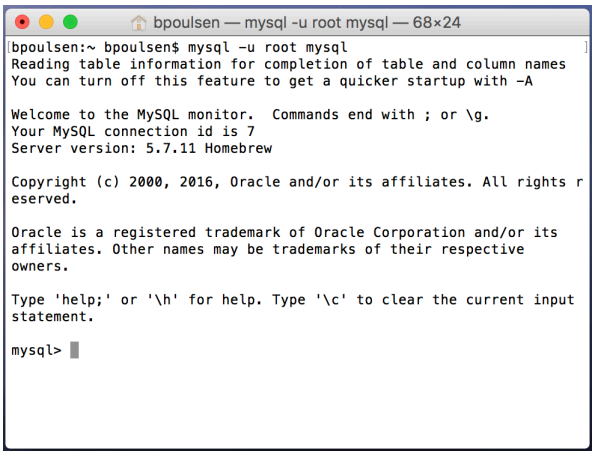
MySQL provides a command line tool for interfacing with its database products. You will be using this interface to query the database, and to perform other SQL against the database during this course, so let's see what it looks like!

On Mac OSX systems, MySQL needs to be accessed via the terminal (Finder > Go > Utilities > Terminal).



Once the terminal window is open issue the following command:

```
mysql -u root mysql
```



This logs you in as the root user which is the most powerful user in MySQL. You will rarely do this for typical operations in MySQL because it is very easy to make large mistakes.

Part II - Saving and Printing Your Work

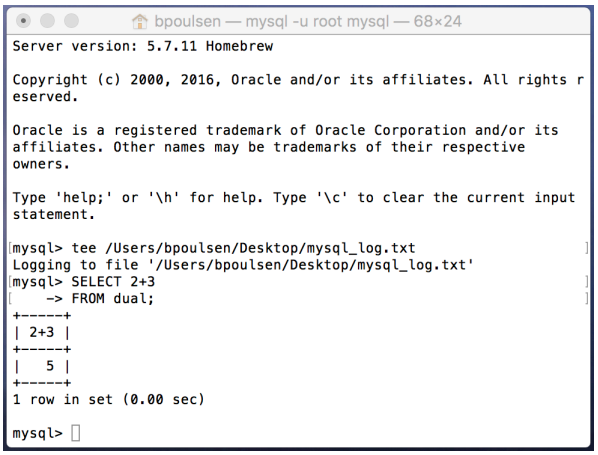
Let's prepare MySQL to save our commands and its responses through a process called spooling.

First create a text file (mysql_log.txt) and save it to your Desktop. Next in the terminal window running MySQL type "tee" followed by a space then drag the file you created from the desktop into the terminal window (this is the easiest way to get the absolute path to the text file you want to save your spooling data to) and press enter.

Part III - Beginning SQL (Using dual)

Let's look at a simple database query (an SQL command). Type the following in the MySQL interface. Press enter after each line, don't forget the semicolon (;) following the word dual:

```
SELECT 2+3
FROM dual;
```



This simple query indicates some important features of all database queries:

1. A query is a special type of SQL command used to obtain and display data from the database, and must begin with the SELECT keyword.
2. One or more column names (each separated by a comma, if more than one) must immediately follow the SELECT command.
3. The SELECTed columns must come FROM some MySQL object (like a table).
4. All queries must be terminated with a semicolon (;).

But wait! What is this "dual" thing? "dual" is a special "table" provided by MySQL. You may find many uses for "dual" while testing and debugging tricky SQL statements, especially those involving arithmetic and dates.

What is the single column identified in your results from the above query? (Hint: don't think about this too much – what does the column heading appear to be on the screen?) That's right; the column name is the same as the arithmetic expression passed to dual (i.e., 2+3). This odd sounding column heading is coincidental, and related only to the way we used dual in this example. What is important here is that anything entered after the SELECT command, and before the FROM qualifier, will be considered a "column name" by MySQL.

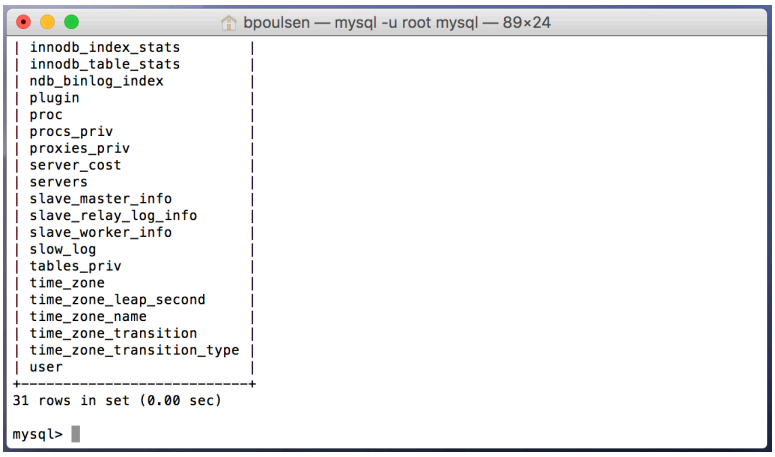
Finally, each database query is terminated with a semicolon (;). The semicolon signals the SQL engine running on the database server that entry is complete, and the command is ready for execution. You may already know that the semicolon is used in many

programming languages to terminate individual elements in the code, utilizing a concept known as tokenization.

Let's look at a couple other commands available in MySQL. From the MySQL interface, execute the following command;

```
SHOW tables;
```

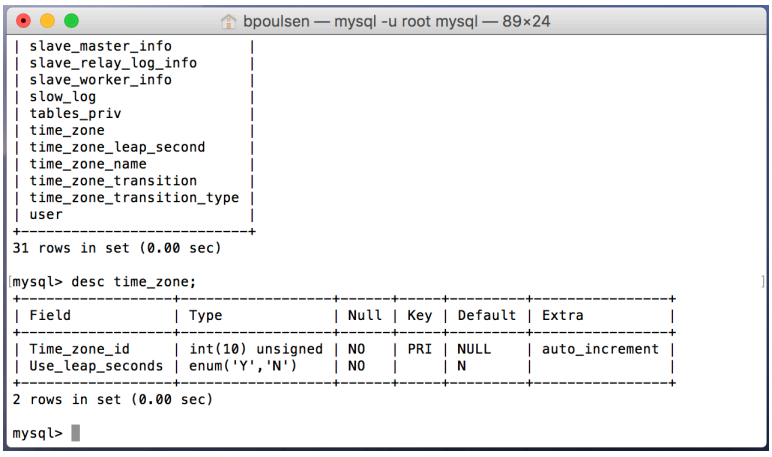
This will list the tables available in the current database. We are currently using the "mysql" database which we selected when logging into mysql.



Next enter the command:

```
desc time_zone;
```

The command desc is shorthand for "describe". What is the describe telling us about "time_zone"? We are told that time_zone is an object (a table in this case), consisting of two columns named "Time_zone_id" and "Use_leap"seconds". It also shows us the type of data that each is and a few other pieces of information. We won't go into too much detail about this for the time being. We just wanted to see an example of what the "desc" command does.



Part IV - Getting Some Real Tables to Work With

What Tables are out there now?

One of the first things all MySQL users need to learn is the way to determine what tables are available for use, and what their columns and data types include. We already looked that the "show tables;" command which lists all tables in the current database.

We are going to create a new database and new user in MySQL to use for future labs.

1. Creating a new database in MySQL by issuing the following command in the MySQL interface:

```
CREATE DATABASE ict4300;
```

This will create a database named "ict4300"

2. Create a new user in MySQL by issuing the following command in the MySQL interface:

```
CREATE USER 'student'@'localhost' IDENTIFIED BY 'password';
```

This will create a new user "student" allowed to connect from the host "localhost" with a password equal to "password".

3. Finally we need to grant the user permissions to use the database we created. Issue the following command in the MySQL interface:

```
GRANT ALL PRIVILEGES ON ict4300.* TO 'student'@'localhost';
```

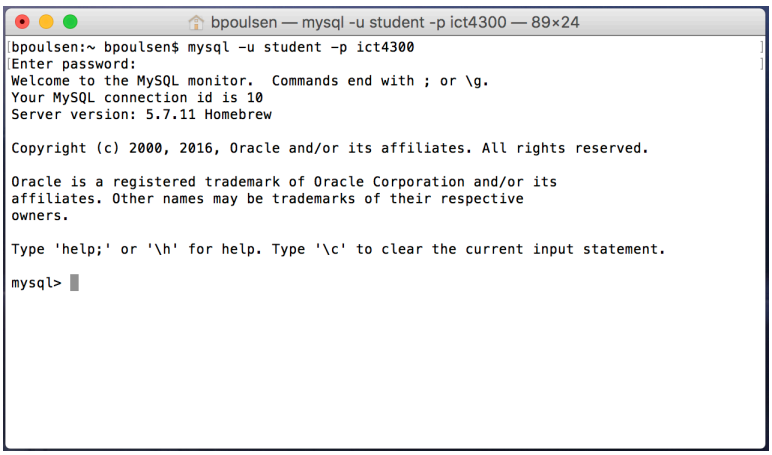
This will give the user "student" all permissions for any table on database "ict4300"

Now that we have a new user and database we need to `exit` the current MySQL session and login as the new user and connect to the new database we created.

To log in MySQL as the new user and connect to the new database issue the following command in a new terminal window:

```
mysql –u student –p ict4300
```

The "-u student" mean use user "student". The "-p" asks for a password prompt (don't forget the password! we used "password" in the example above incase you forgot). Finally the "ict4300" mean use the database "ict4300".



Running Scripts in MySQL

Now, let's talk about a special command in MySQL "source". The source command in MySQL allows you to run or execute MySQL scripts. Scripts can be written to do anything imaginable with a database. For instance, scripts can be written to create entirely new databases, tables, and user accounts. The use of scripts is a convenience feature preferred by database professional because of the easy and powerful editing capabilities provided by text editors.

In fact, your are about to run your first script! You have been provided a file named **demobld_mysql.sql**. It can be downloaded from the files area of canvas. This is a text file containing the instructions for MySQL to build a set of tables for you. The script will also instruct MySQL to populate your new tables with data, enable a series of integrity constraints on those tables (presented in future labs), and build a series of views and sequences. Because demobld_mysql.sql is just a text file, you may open it with notepad or some other simple text editor. You are encouraged to do so, but be very careful not to alter the file in any way.

To run this (or any other) script, issue the following command in the MySQL interface. (like earlier with the "tee" command, you can drag the demobld_mysql.sql file into the MySQL interface instead of typing the full path to the file.

```
source <path_to_folder>/demo_mysql.sql
```

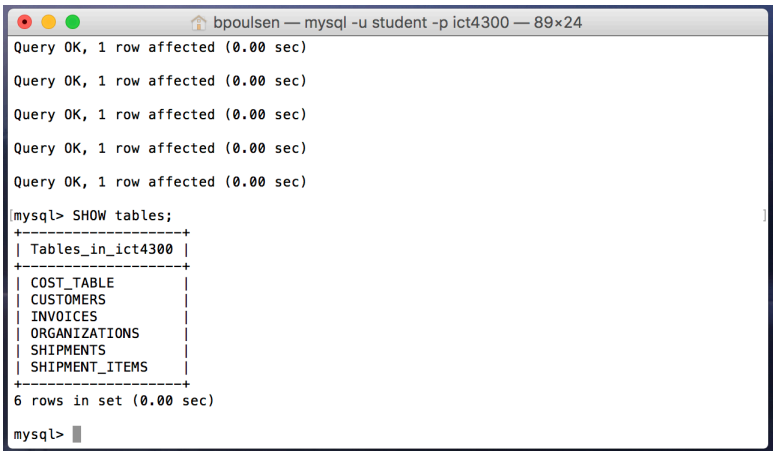
You should get a series of "Query OK, 1 row affected" statements. Congratulations you have just executed the demobld_mysql.sql script in MySQL.

Part V - Briefly Exploring Your New Tables

Execute the following command in the MySQL interface:

```
SHOW tables;
```

The demobld_mysql.sql script you just rand created the tables that we will use throughout the class.



Part VI - Ending Your Session

Always properly end your SQL session by issuing the following command:

```
exit;
```