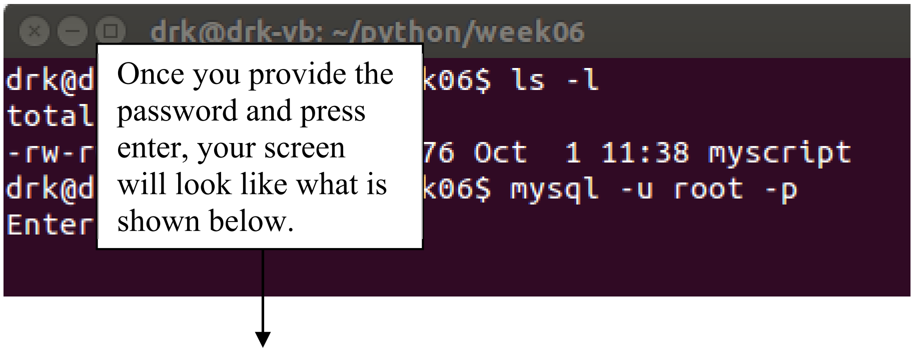
Start-up document for the SQL exercise

This handout is will show you how to load data into your SQL tables. To use this handout you must also have access to the **myscript** text file that contains the SQL code.

As you can see from the first screenshot below, I have only one file (`myscript`) in my week06 folder. Make sure that the `myscript` file in the folder from where you start mysql. The command to start mysql is

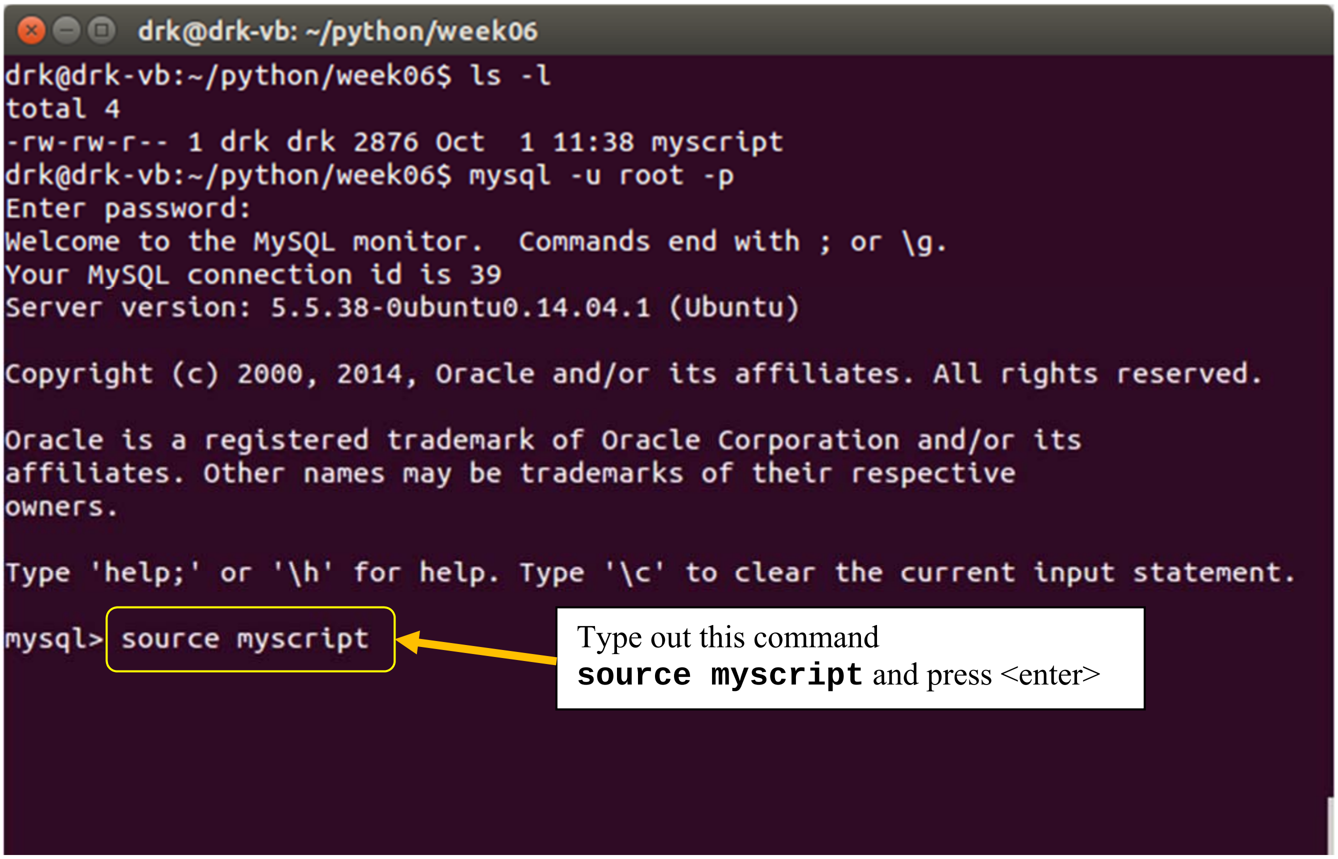
mysql –u root -p

The –u option is followed by a space followed by the name of the user (in this case root). The –p option followed by nothing tells the command line interpreter that the user is going to provide the password (see below).



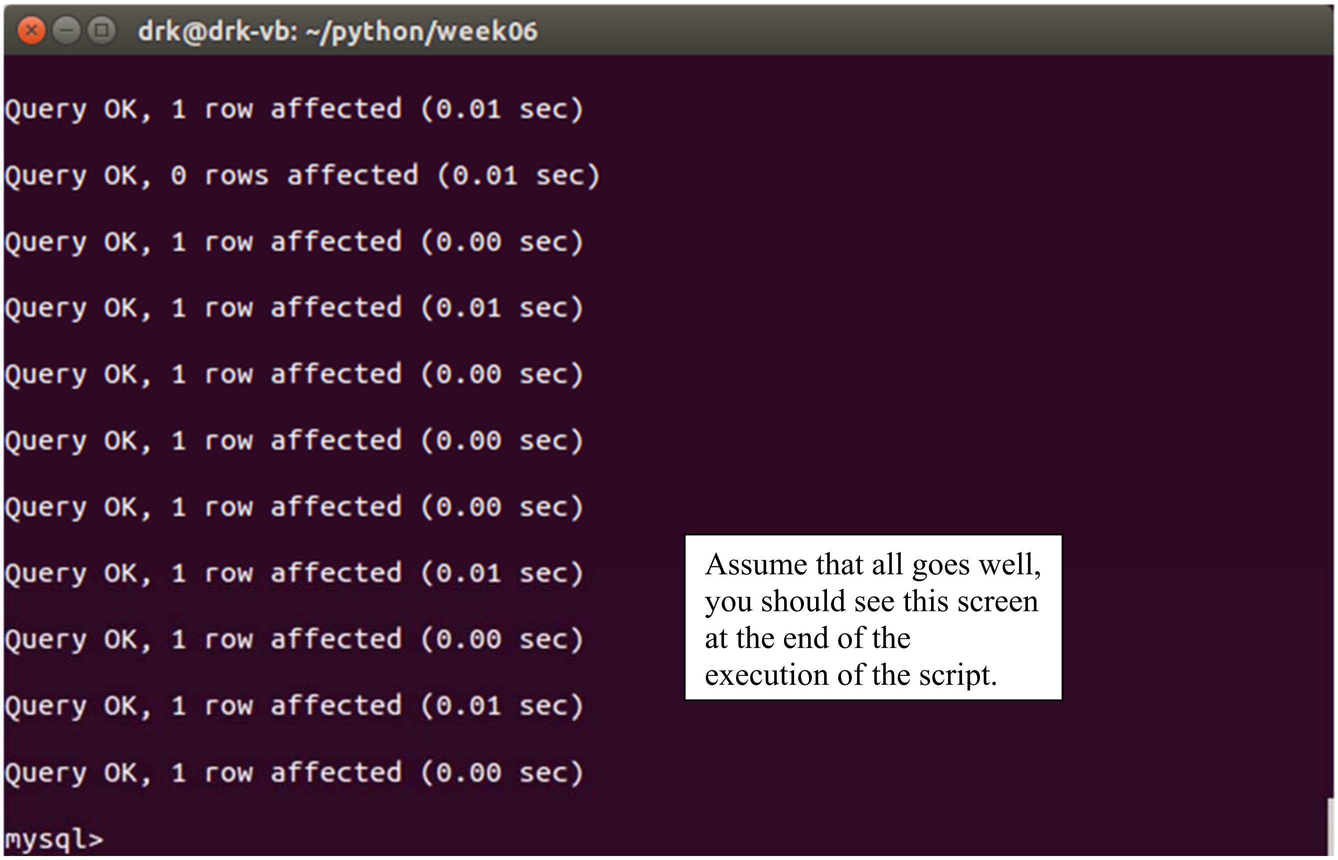
```
drk@d            k06$ ls -l
total
-rw-r            76 Oct  1 11:38 myscript
drk@d            k06$ mysql -u root -p
Enter
```

Once you provide the password and press enter, your screen will look like what is shown below.



```
drk@drk-vb:~/python/week06$ ls -l
total 4
-rw-rw-r-- 1 drk drk 2876 Oct  1 11:38 myscript
drk@drk-vb:~/python/week06$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.5.38-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source myscript
```

Type out this command
`source myscript` and press <enter>



```
Query OK, 1 row affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

mysql>
```

Assume that all goes well, you should see this screen at the end of the execution of the script.

The contents of the myscript file are shown below. You can edit that file and check the statements out on your own.

```
SHOW DATABASES;

DROP DATABASE IF EXISTS classwork;
```

← delete a database called 'classwork' if it exists

```
CREATE DATABASE classwork;
```

← create a database named 'classwork'

```
USE classwork;
```

← Use the classwork database for subsequent commands

```
SHOW TABLES;
```

← There shouldn't be any tables — we just created the database

```
CREATE TABLE Student (
  stuId CHAR(6),
  lastName CHAR(20) NOT NULL,
  firstName CHAR(20) NOT NULL,
  major CHAR(10),
  credits SMALLINT DEFAULT 0,
  CONSTRAINT Student_stuId_pk PRIMARY KEY (stuId),
  CONSTRAINT Student_credits_cc CHECK ((CREDITS>=0) AND (credits < 150))
);
```

← create the Student table

```
INSERT INTO Student  VALUES ("S1001", "Smith", "Tom", "History", 90);
INSERT INTO Student  VALUES ("S1002", "Chin",  "Ann", "Math", 36);
INSERT INTO Student  VALUES ("S1005", "Lee",  "Perry", "History", 3);
INSERT INTO Student  VALUES ("S1010", "Burns", "Edward", "Art", 63);
INSERT INTO Student  VALUES ("S1013", "McCarthy", "Owen", "Math", 0);
```

← populate the Student table

```
INSERT INTO Student  VALUES ("S1015", "Jones",  "Mary", "Math", 42);
INSERT INTO Student  VALUES ("S1020", "Rivera",  "Jane", "CSC", 15);
```

← Create the Faculty table

```
CREATE TABLE Faculty (
  facId CHAR(6),
  name CHAR(20) NOT NULL,
  department CHAR(20) NOT NULL,
  rank CHAR(10),
  CONSTRAINT Faculty_facId_pk PRIMARY KEY (facId)
);
```

← populate the Faculty table

```
INSERT INTO Faculty VALUES ("F101", "Adams", "Art", "Professor");
INSERT INTO Faculty VALUES ("F105", "Tanaka", "CSC", "Instructor");
INSERT INTO Faculty VALUES ("F110", "Byrne", "Math", "Assistant");
INSERT INTO Faculty VALUES ("F115", "Smith", "History", "Associate");
INSERT INTO Faculty VALUES ("F221", "Smith", "CSC", "Professor");
```

← Create the Class table

```
CREATE TABLE Class (
  classNumber CHAR(8),
  facId CHAR(6) NOT NULL,
  schedule CHAR(8),
  room CHAR(6),
  CONSTRAINT Class_classNumber_pk PRIMARY KEY (classNumber),
  CONSTRAINT Class_facId_fk FOREIGN KEY (facId) REFERENCES Faculty (facId) ON
DELETE NO ACTION
);
```

← populate the Class table

```
INSERT INTO Class VALUES ("ART103A", "F101", "MWF9", "H221");
INSERT INTO Class VALUES ("CSC201A", "F105", "TuThF10", "M110");
INSERT INTO Class VALUES ("CSC203A", "F105", "MThF12", "M110");
INSERT INTO Class VALUES ("HST205A" ,"F115", "MWF11", "H221");
INSERT INTO Class VALUES ("MTH101B", "F110", "MTuTh9", "H225");
INSERT INTO Class VALUES ("MTH103C", "F110", "MWF11", "H225");
```

← Create the Enroll table

```
CREATE TABLE Enroll (
  classNumber CHAR(8),
  stuId CHAR(6),
  grade CHAR(2),
  CONSTRAINT Enroll_classNumber_stuId_pk PRIMARY KEY (classNumber, stuId),
  CONSTRAINT Enroll_classNumber_fk FOREIGN KEY (classNumber) REFERENCES Class
(classNumber) ON DELETE NO ACTION,
  CONSTRAINT Enroll_stuId_fk FOREIGN KEY (stuId) REFERENCES Student (stuId) ON
DELETE CASCADE
);
```

← populate the Enroll table

```
INSERT INTO Enroll VALUES ("ART103A", "S1001", "A");
INSERT INTO Enroll VALUES ("HST205A", "S1001", "C");
INSERT INTO Enroll VALUES ("ART103A", "S1002", "D");
INSERT INTO Enroll VALUES ("CSC201A", "S1002", "F");
INSERT INTO Enroll VALUES ("MTH103C", "S1002", "B");
INSERT INTO Enroll VALUES ("ART103A", "S1010", NULL);
INSERT INTO Enroll VALUES ("MTH103C", "S1010", NULL);
INSERT INTO Enroll VALUES ("CSC201A", "S1020", "B");
INSERT INTO Enroll VALUES ("MTH101B", "S1020", "A");
```