SQL Continued

Before you begin please do the following:

1. Watch the video lecture posted. Many of the commands I will show you on the video.
2. You can also look at the SQLCheatsheet in the "Resources" section for many of the common SQL commands.
3. The below exercises corresponds to the lecture titled SQL Continued.

This homework continues from the previous one.

Submission:
A SQL file is a text file produced by MySql workbench or any other SQL IDE. It is not a Word file, nor a PDF**.** You must create a SQL file with all your answers: The SQL file should have your name in it commented by a "#" on the first line. The following line is for the "use" command you will issue, commented with a "#". The following line will contain the date and number that results from:
`SELECT @d:=NOW(),SHA(@d);` separated by a TAB in a "#" comment, followed by a blank line.

Then, for every answer, a comment line with the question number and the remaining lines should contain the answer. Then, skip a blank line and repeat the process. For example, a file from John Doe with 2 questions, where the first question has 2 items, may look like this:

```
# John Doe
#use jdoe_db;
#2017-01-05 17:38:27  41860934238f8b3a1fdd51dee11e2d4219c92068

#1.1
Some SQL code goes
Here as the answer to
The question. SELECTs, CREATEs, whatever.

#1.2
More SQL code answering
Question number 1.2 (or I.2)
The answer can take up
As many lines as you need

#2.1
Yet more SQL commands
That answer
The first part of the second question
```

If you want to add additional comments, use the – notation.
You should be able to load your entire SQL file into your IDE, uncomment the second line and run your code without **syntax** errors, at least, before submitting.

You will need to submit the SQL file. **Other formats will not be graded**. The SQL file ends in **.sql** you can choose whatever name you like WITHOUT spaces. A good idea is your last and first name and homework number. For example: `DoeJohnHW2.sql`

I. Update Continued

1. Guest 101 has asked for a discount in rent. Write the statement to update that information for guest 101 only for their June 2014 entry.
2. Write a statement to view that your changes took place correctly.
3. Guest 102, who arrived on July 1st, 2014 extends his stay to August 31, 2014. Write the statement to update BOTH the price and dateDepart for that entry. *Note: In the same statement.*
4. Write a statement to view that your changes took place correctly.

II. Aggregates

1. Write a statement to COUNT how many **bookings** there are.
2. Write the code to find the AVERAGE price, MINIMUM price, MAXIMUM price, and SUM of all prices for all of the bookings.
3. Write the select statement to count how many bookings belong to guest 101.

III. Group by

1. Select the guest number, and the average price. Group the results by guestNo and order the results by the guest number DESCENDING.
   Hint: In addition to Group By you will need to use above aggregates
2. Select the guest number, and the average price, grouped by guestNo, but ONLY display it when the average price is greater than or equal to 500.
   Hint:You cannot use 'where' in Group By when using Aggregates - use Having!

IV. Subqueries

1. Use a **subquery** to select the arrival and departure attributes where the guest number corresponds to a guest number whose last name is Washington.

V. Joins

1. Select the guest number, guest first name, last name, and the arrival and departure dates for those guests that have bookings.
2. Select the guest number, guest first name, last name, and arrival and departure dates whether or not there is a booking. (All guests in the guest table should appear)
   Hint: Use Left or Right Join