ORIE5270 - Homework 7
Due March 15th, 2019 at 1:25pm

You are not allowed to collaborate with your classmates. Submit this assignment by pushing a folder HW7 to your ORIE5270 repository on cornell's github with your solutions as well as a solution file in pdf format containing your repo `url` to Blackboard. Make sure all your code are readable. You should also tell us how to run your code in your solution file.

Problem 1 (Iris dataset) We have seen the Iris dataset in class. Now you will have a chance to manipulate it by yourself. Write a python file `problem1.py` so that once we run it in command line, it can finish the following tasks:

(1) It directly scribes the `iris.data` from "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data". You can check "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.names" for the meaning of the values.

(2) It creates a SQLite database called `problem1` with a table `Iris` by using the package `sqlite3`. The table `Iris` should consist of columns

`ID`, `SepalLength, SepalWidth, PetalLength, PetalWidth, Class`,

and the rows are the `iris.data`. Make sure the correspondence of columns between the `iris.data` and your table are correct. Since `ID` is not given in the original dataset, you should just assign ID `i` to the `i`-th row. So the first row `5.1,3.5,1.4, 0.2,Iris-setosa` should have ID `1` and the second row `4.9,3.0,1.4,0.2,Iris-setosa` should have ID `2` and so on.

(3) It calls `sqlite3` to compute the maximum, minimum, average of the column `SepalWidth`. Print the results in a readable form (a table with column and row name is fine).

(4) It calls `sqlite3` to compute the maximum, minimum, average of the column `SepalWidth` for each class: Iris Setosa, Iris Versicolour, and Iris Virginica. So you should have nine numbers this time. Print the results in a readable form (a table with column and row name is fine).

You solution file should contain the two printed results in (3) and (4). Comment on your results.

Problem 2 (Join Operation) We have seen the `JOIN` operation provided by `SQL` in class. In this problem, you will get some experience of using two different `JOIN` operations: `INNER JOIN` and `LEFT OUTER JOIN`. Write a python file `problem2.py` so that once we run it in command line, it can finish the following tasks:

(1) It creates a `SQLite` database called `problem2` with a table called `Customer` with content:

CustomerId	Name
1	Paul Novak
2	Terry Neils
3	Jack Fonda
4	Tom Willis

and another table `Reservations` with content

Id	CustomerId	**Day**
1	1	2009−22−11
2	2	2009−28−11

3	2	2009−29−11
4	1	2009−29−11
5	3	2009−02−12

(2) It calls `sqlite3` to execute the command

```
SELECT Name, Day FROM Customers AS C JOIN Reservations
AS R ON C.CustomerId = R.CustomerId
```

It will then fetch the result and print it.

(3) It calls `sqlite3` to execute the command

```
SELECT Name, Day FROM Customers LEFT JOIN Reservations
ON Customers.CustomerId = Reservations.CustomerId
```

It will then fetch the result and print it.

The command in step (2) is called `INNER JOIN` and the one in step (3) is called `LEFT OUTER JOIN`. In your solution file, you should write what do you observem, explain in your own words what is the difference between the two, and include the printed results of step (2) and (3).