

ORIE 5270 HW7

Ko-Cheng Wang (kw582)

Repo url: <https://github.coecis.cornell.edu/kw582/ORIE5270/tree/master/HW7>

Files Description:

1. Solution.pdf: Contain the detailed solution to HW7
2. problem1.py: Contain the solution code to Problem 1
3. problem2.py: Contain the solution code to Problem 2

Problem 1)

Run problem1.py with the following command:

```
python3 problem1.py
```

Screenshot of the result of running problem1.py (including results of Question 1-3 and 1-4)

```
[LuckydeMacBook-Air:HW7 luckywang$ python3 problem1.py

Answer to Question 3:
      Max SepalWidth Min SepalWidth  Avg SepalWidth
Result                4.4             2.0           3.054

Answer to Question 4:
      Max SepalWidth Min SepalWidth  Avg SepalWidth
Class
Iris-setosa          4.4             2.3           3.418
Iris-versicolor      3.4             2.0           2.770
Iris-virginica        3.8             2.2           2.974
```

Comment:

By comparing the above two results, it shows that the max, min and average sepal width of class "Iris-setosa" are all the highest among three classes (4.4, 2.3, 3.418). On the other hand, the class "Iris-versicolor" has the shorter sepal width than the other two classes. The overall max and min sepal width come from the class "Iris-setosa" and "Iris versicolor" respectively.

Screenshot of the first five rows in the Iris table (Question 1-1, 1-2)

```
[sqlite> select * from Iris limit 5;
ID      SepalLength  SepalWidth  PetalLength  PetalWidth  Class
-----
1        5.1           3.5         1.4          0.2         Iris-setosa
2        4.9           3.0         1.4          0.2         Iris-setosa
3        4.7           3.2         1.3          0.2         Iris-setosa
4        4.6           3.1         1.5          0.2         Iris-setosa
5        5.0           3.6         1.4          0.2         Iris-setosa
```

Problem 2)

Run problem2.py with the following command:

```
python3 problem2.py
```

Screenshot of the result of running problem2.py (including results of Question 2-2 and 2-3)

```
[LuckydeMacBook-Air:HW7 luckywang$ python3 problem2.py
Answer to Question 2:
('Paul Novak', '2009-22-11')
('Paul Novak', '2009-29-11')
('Terry Neils', '2009-28-11')
('Terry Neils', '2009-29-11')
('Jack Fonda', '2009-02-12')

Answer to Question 3:
('Paul Novak', '2009-22-11')
('Paul Novak', '2009-29-11')
('Terry Neils', '2009-28-11')
('Terry Neils', '2009-29-11')
('Jack Fonda', '2009-02-12')
('Tom Willis', None)
```

From the above result, I notice that the only difference between Question 2-2 and 2-3 is that in Question 2-3, there is an extra row of ('Tom Willis', None).

The reason why the date is "None" is that Tom Willis did not really make a reservation according to the table "Reservations". For "LEFT OUTER JOIN" (used in 2-3), the join will start with the first (left) table and then try to match the result in the second table. However, all the rows in the left table will be presented in the final result, no matter whether there is a match in the second table or not; On the other hand, "INNER JOIN" (used in 2-2) will only output the records that have matching values in both tables. This is the reason why 'Tom Willis' record is not presented in the final output.