**SI 507**

**Homework 9:** Basic SQL statements

**Homework Objectives**

* Understand databases and SQL
* Write basic SQL statements
* Be able to join two tables or a table to itself

**Supporting Material**

* Northwind Database

<https://drive.google.com/open?id=15v9Up9EUDSWFTKLBb4cwgDSsbQAK-hPm>

* Lecture 9, 10, and 11
* W3schools SQL tutorial

<https://www.w3schools.com/sql/>

**Starter Files**

We have provided you with the following file:

* [hw10.py](https://www.dropbox.com/s/godush2lm0vz3ic/hw10.py?dl=0)

For HW10, we will continue to use *Northwind\_small.sqlite*. Inside of the database, there are 13 tables:

1. Employee
2. Category
3. Customer
4. Shipper
5. Supplier
6. Order
7. Product
8. OrderDetail
9. CustomerCustomerDemo
10. CustomerDemographic
11. Region
12. Territory
13. EmployeeTerritory

Write out SQL statements for the following 10 queries about the Northwind database (2 points per query: 1.5 (correct query result, no partial credit for incorrect query result) + 0.5 (pretty print)). *Print out the results nicely* - *there is no required format, but printing raw outputs (lists or tuples) will result in -0.5 point per query.*

Please use the skeleton code provided to you, and **don’t change the functions’ names**. Write your query inside the function for each question. It is ok to either print out the result inside or outside the function, but your function should return the **raw result** you get from query, which is **always** **a list of tuples**. We will use the return value of the function to grade whether your query returns the correct result (order doesn’t matter as long as it’s not specified in the question). If your function returns a different type you may lose points for it. Your printing process should only change how the data is presented without changing the data. For example, if the question asks for name, the result you get from database should only includes names, instead of getting everything and only displaying name during printing.

You need to get the data for these questions only using a SQL query, without python data processing (except for the 7th one, which I didn’t find an easy way to accomplish using only sql query, but if you can you are awesome). For example, for the 2nd question, you need to directly get a number back (still in a form of a list of tuples, though) from your SQL query instead of getting all the rows and counting it using python. Or for the 4th question you need to make sure you are only getting 5 records from the database, instead of getting all and only returning the first 5.

1. Show all rows from the *Region* table
2. How many customers are there?
3. How many orders have been made?
4. Show the first five rows from the *Product* table
5. Show the names of the five cheapest products
6. Show the names and number of units in stock of all products that have more than 100 units in stock
7. Show all column names in the *Order* table
8. Show the contactnames of all customers who lives in USA and have a fax number on record.
9. Show the names of all the products, if any, that require a reorder.

(If the units in stock of a product is lower than its reorder level but there's no units of that product currently on order, the product requires a reorder.)

1. Show ids of all the orders that ship to France where postal code starts with "44"

**Extra Credit 1 (2 points)**

For EC1, we are going to see the number of days passed between orders for each customer. Identify each customer id, order date, previous order date, and the number of days passed between two order dates *in order of Customer’s Id and then the order date*. Output should look like below.

CustomerID,Order date,Previous order date, days passed

ALFKI,2013-10-03,2013-08-25,39.0

ALFKI,2013-10-13,2013-10-03,10.0

ALFKI,2014-01-15,2013-10-13,94.0

ALFKI,2014-03-16,2014-01-15,60.0

ALFKI,2014-04-09,2014-03-16,24.0

ANATR,2013-08-08,2012-09-18,324.0

ANATR,2013-11-28,2013-08-08,112.0

ANATR,2014-03-04,2013-11-28,96.0

.

.

.

WILMK,2013-09-18,2013-07-30,50.0

WILMK,2013-10-07,2013-09-18,19.0

WILMK,2014-02-06,2013-10-07,122.0

WILMK,2014-02-10,2014-02-06,4.0

WILMK,2014-02-26,2014-02-10,16.0

WILMK,2014-04-07,2014-02-26,40.0

WOLZA,2013-07-25,2012-12-05,232.0

WOLZA,2013-12-23,2013-07-25,151.0

WOLZA,2014-02-04,2013-12-23,43.0

WOLZA,2014-02-25,2014-02-04,21.0

WOLZA,2014-04-03,2014-02-25,37.0

WOLZA,2014-04-23,2014-04-03,20.0

**Extra Credit 2 (2 points)**

For EC2, we are going to use two user inputs to complete the query: 1) the first character of ShipCity and 2) the number of characters in employee's first names. Your program should show the number of orders made based on the inputs. For example, bogus inputs would give us 0 order.

Sample Command:

$ python hw9\_ec2.py s 5

Sample Output:

Searched for

1) ShipCity starts with: s

2) The number of characters in employee's first name: 5

The number of orders: 40

\*Reference: the 40 orders captured by the query are as follows.

('Sao Paulo', 'Laura')

('Sao Paulo', 'Nancy')

('Stuttgart', 'Laura')

('Stavern', 'Nancy')

('Strasbourg', 'Janet')

('Strasbourg', 'Janet')

('Seattle', 'Nancy')

('San Cristóbal', 'Laura')

('San Cristóbal', 'Nancy')

('San Cristóbal', 'Laura')

('Salzburg', 'Janet')

('San Francisco', 'Nancy')

('Sao Paulo', 'Janet')

('Seattle', 'Laura')

('Stuttgart', 'Laura')

('Stuttgart', 'Laura')

('Stuttgart', 'Nancy')

('Strasbourg', 'Laura')

('Seattle', 'Janet')

('Seattle', 'Laura')

('San Francisco', 'Laura')

('Seattle', 'Janet')

('Sao Paulo', 'Laura')

('San Cristóbal', 'Janet')

('Stavern', 'Janet')

('Sao Paulo', 'Nancy')

('Sao Paulo', 'Janet')

('Salzburg', 'Laura')

('Sevilla', 'Nancy')

('Seattle', 'Janet')

('Stavern', 'Nancy')

('Sevilla', 'Janet')

('Sevilla', 'Janet')

('San Cristóbal', 'Laura')

('San Cristóbal', 'Janet')

('Sao Paulo', 'Laura')

('Sao Paulo', 'Nancy')

('San Cristóbal', 'Nancy')

('Stuttgart', 'Laura')

('Sao Paulo', 'Laura')

**What to turn in on Canvas**

* Your hw9.py file
* Any extra credit file
* **No need to submit the .sqlite file**