HW 2: DB, SQL, and Python

Overview

In this week's exercise you will practice writing basic queries to retrieve data from the Northwind database. In total you will create 12 SQL statements, which will select columns, filter and order rows, and join tables to retrieve data.

Supporting Materials

Starter code:

- Downloaded files from Canvas into the same folder:
  - hw2_db_sql.py (where you'll do your work)
  - hw2_db_sql_test.py. Run this file to check if your functions return expected values (consider this an autograder)

Database:

- Download Northwind_small.sqlite and move it into your starter code folder.

Documentation:

- Lecture and Lab Notes from Week 3
- W3schools SQL tutorials

Main Assignment

Like in the lecture, we will continue work with the Northwind database. It is composed of 13 tables (the bold ones are relevant for the main assignment):

• **Employee** • **Category** • **Customer** • Shipper • Supplier • **Order** • **Product**	• OrderDetail • CustomerCustomerDemo • CustomerDemographic • Region • **Territory** • EmployeeTerritory

Use the starter code provided for you, and **don't change the functions' names**. We encourage though to use of helper functions in order to avoid code duplication where you think it may make sense (e.g. for DB connection set up). You will write 12 SQL queries inside of the given functions to answer questions or show specific rows/entries:

1. Show **all fields** from the **Territory** table.
2. **How many employees** are there?
3. Show the **last 10 rows** of the **Product** table **ordered by Id** in **descending** order.
4. Show the **names and unit prices** of the **three products** that have **the highest unit prices.**
5. Show the **names, unit prices and numbers of units in stock** of all **products** that have **between 60 and 100** (including 60 and 100) **units in stock.**
6. Show the **names** of all the **products**, if any, that **require a reorder** (If the units in stock of a product are lower than its reorder level, the product requires a reorder).
7. Show **Ids** of all the **orders** that were shipped to **France** where **postal code ends with "04".**
8. Show the **company names and contact names** of all **customers** who live in the **UK** and **have a fax number on record**.
9. Show the **names and unit prices of all Beverage products**.
10. Show the **names of all Meat/Poultry products** that were **discontinued** (Discontinued column has a value of 1).
11. Show **order Ids** and **first names** and **last names of associated employees** of all orders that were shipped to **Germany**.
12. Show the **order Ids**, **order dates**, and **names of the companies that placed the respective orders** of all orders that were placed on or before July 10, 2012.

Each functions should just return the **raw fetched result** you get from making each query, which is **always a list of tuples**. We will use the return values of the functions to test your queries.

You need to retrieve and process data using only SQL queries, without python data processing. For example, the 2nd question requires you to directly get the number through your SQL query, instead of retrieving all the rows and counting them using python.

What to Submit

Rename hw2_db_sql.py to hw2_db_sql_<your_uniquename>.py and submit to Canvas. Make sure that you write your name and uniqname in the comment at the top of the file. This will make our awesome GSIs' lives easier!

Rubrics

Req	Description	Category	Point Value
1	Query 1 yields correct results.	Code	7
2	Query 2 yields correct results.	Code	7
3	Query 3 yields correct results.	Code	7
4	Query 4 yields correct results.	Code	7
5	Query 5 yields correct results.	Code	7
6	Query 6 yields correct results.	Code	7
7	Query 7 yields correct results.	Code	7
8	Query 8 yields correct results.	Code	7
9	Query 9 yields correct results.	Code	7
10	Query 10 yields correct results.	Code	7
11	Query 11 yields correct results.	Code	7
12	Query 12 yields correct results.	Code	7
14	Starter code is unchanged (only added to).	Code	8
15	Code style is good and complies with 507 Assignment Guide	Code	8
	Total		100

Extra Credit Note:

This homework includes extra credit questions that include materials we haven't covered in class yet. If you want a challenge, you are welcome to work on them and do some research on what to do. There will be other extra credit opportunities in both future homeworks AND non-coding activities.

Extra Credit #1

Define a function called **print_query_result(raw_query_result)** that takes any raw query result as a parameter and prints the query result to the terminal in a clean format. The specific format is up to you. However, your program (1) should NOT just print the list of tuples returned from each query, and (2) the columns of your printed table should be aligned. That is, below is NOT OK, because "|"s aren't vertically aligned

```
|Id|ProductName|UnitPrice|
|1|Coke|2|
|2|Ginger Ale|3|
|3|Pepsi|1|
```

Notes

- You can research and use any libraries you like. You can also research python's built-in string formatting functionality, such as f-strings.
- You don't have to print out column names.
- You can use fixed-width columns to ensure column alignment. If the value in a field is too long (say, >20 characters), you should show trimmed value in the fixed-width field (e.g. "Four score and seven...").
- Don't worry about extreme cases such as result sets with 100 fields or the like. You can focus on making the function work for the kinds of queries you created for this assignment (with roughly 3-4 fields).
- For any other implementation decisions not specified here, use your best judgment and do something reasonable.

Sample Output:

```
+---------+----------+-----------+
|  10249  | Michael  |   Suyama  |
|  10260  | Margaret |  Peacock  |
```

```
|  10267  | Margaret |  Peacock  |
|  10273  |  Janet   | Leverling |
|  10277  |  Andrew  |   Fuller  |
|  10279  |  Laura   |  Callahan |
|  10284  | Margaret |  Peacock
|  11070  |  Andrew  |   Fuller  |
+---------+----------+-----------+
```

What to Submit

Work under the **print_query_result(query_result)** function provided for you in the starter code (EC1 and the main assignment are in one file).

Rubrics

Req	Description	Category	Point Value
1	The query result is printed in full and in a somewhat readable format. No omission of columns or rows is made.	Code	1
2	The columns of the printed query result are aligned.	Code	1
	Total		**2**

Extra Credit #2

Write an interactive program that keeps prompting user to enter an order date and ship country to find out which employee was responsible for this particular order and print the employee first name and last name to terminal (Hint: (1)How can you dynamically construct SQL query based on user input? (2) You should use EC1 for pretty printing.) The program should quit when the user types “exit.”

Sample Output:

```
Please enter a Order Date and a Ship Country seperated by spac
e (e.g. 2012-07-04 France), or 'exit' to quit: 2012-07-04 Fran
ce
```

```
The employee responsible for this order is:
+---------+----------+
|  Steven | Buchanan |
+---------+----------+


Please enter a Order Date and a Ship Country seperated by spac
e (e.g. 2012-07-04 France), or 'exit' to quit: 2012-08-01 Germ
any
Sorry! Your search returns no results.


Please enter a Order Date and a Ship Country seperated by spac
e (e.g. 2012-07-04 France), or 'exit' to quit: dsjadsal
Invalid Input. Please enter a Order Date and a Ship Country se
perated by space
```

What to Submit

Work under if _name_ == “_main_" in the starter code provided for you (EC2 and the main assignment are in one file).

Rubrics

Req	Description	Category	Point Value
1	Program uses graceful error handling.	Code	1
2	Program retrieves the correct employee first and last names based on order date and order ship country.	Code	1
	Total		**2**