Homework 1 Part 2: SQL

To start with, if you felt the class was unclear, check out the following tutorial: https://mode.com/sql-tutorial/introduction-to-sql/

Now! We'll be using sqlite to access a database. Start by downloading the sql lite file and putting it in the same directory as this notebook: https://www.kaggle.com/datasets/kaggle/sf-salaries (hit the 'download' button in the upper right). Check out the description of the data so you know the table / column names.

The following code will use sqlite to create a database connection.

```
import sqlite3
import pandas as pd

conn = sqlite3.connect("database.sqlite")
crsr = conn.cursor()
```

Before we proceed, please note that every task **must be completed using a single SQL query**, unless mentioned otherwise or given as prompt. (e.g. using print statements are fine, but you should not be using pandas library to work with the dataset.) The query should be stored in the query variable, like so:

query = 'your query here'

Exploration

Problem 1:

Try to create a query that gives you a data frame of the EmployeeName, JobTitle, and BasePay from the salaries table.

```
query = 'select EmployeeName, JobTitle, BasePay from salaries'

df = pd.read_sql(query, conn)
df.head()
```

	EmployeeName	JobTitle	BasePay
0	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18
1	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02
2	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13
3	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916
4	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.6

Problem 2.

Modify your query from Problem 1 to limit it to the year 2012.

```
query = 'select EmployeeName, JobTitle, BasePay from salaries where Year = 2012'

df = pd.read_sql(query, conn)
df.head()
```

	EmployeeName	JobTitle	BasePay
0	Gary Altenberg	Lieutenant, Fire Suppression	128808.87
1	Gregory Suhr	Chief of Police	302578.00
2	Khoa Trinh	Electronic Maintenance Tech	111921.00
3	Joanne Hayes-White	Chief, Fire Department	296943.01
4	Frederick Binkley	EMT/Paramedic/Firefighter	126863.19

Problem 3:

Further limit the table to the year 2012, employees making under 150,000, and sort in descending order by salary.

```
query = 'select EmployeeName, JobTitle, BasePay, TotalPay from salaries where Year = 2012 and TotalPay < 150000 order by TotalPay desc'

df = pd.read_sql(query, conn)
df.head()
```

	EmployeeName	JobTitle	BasePay	TotalPay
0	Kevin Salas	Firefighter	110847.11	149989.01
1	Delene Wolf	Dept Head I	146497.50	149983.50
2	Steven Stocker	Sergeant 3	135977.79	149956.09
3	Wayne Yu	Electronic Maintenance Tech	101088.00	149934.34
4	Eric Altorfer	Sergeant 2	129175.75	149931.31

Aggregation

Problem 4:

Get the average base pay from the table.

```
query = 'select AVG(BasePay) as AverageBasePay from salaries'

df = pd.read_sql(query, conn)
df.head()
```

	AverageBasePay
0	66053.729288

Problem 5:

Produce and print the head of a dataframe that shows the average pay for each year (only use a single, simple query). Your result should have a column for the year and a column for the average base pay.

```
query = 'select Year, AVG(BasePay) as AverageBasePay from salaries group by Year'

df = pd.read_sql(query, conn)
df.head()
```

	Year	AverageBasePay
0	2011	63595.956517
1	2012	65436.406857
2	2013	68509.832156
3	2014	66557.437750

Problem 6:

Create a dataframe with averages of base pay, benefits, and overtime for each job title, as well as a column with the average of these three values.

```
avg = 'AVG(BasePay) as AvgBasePay, AVG(Benefits) as AvgBenefits, AVG(OvertimePay) as AvgOvertime'
avgs = f'{avg}, (AVG(BasePay) + AVG(Benefits) + AVG(OvertimePay)) / 3 as AvgOfAverages'
query = f'select JobTitle, {avgs} from salaries group by JobTitle'

df = pd.read_sql(query, conn)
df.head()
```

	JobTitle	AvgBasePay	AvgBenefits	AvgOvertime	AvgOfAverages
0	ACCOUNT CLERK	43300.806506	0.00	373.200843	14558.002450
1	ACCOUNTANT	46643.172000	0.00	0.000000	15547.724000
2	ACCOUNTANT INTERN	28732.663958	0.00	24.430625	9585.698194
3	ACPO,JuvP, Juv Prob (SFERS)	62290.780000	17975.59	0.000000	26755.456667
4	ACUPUNCTURIST	66374.400000	0.00	0.000000	22124.800000

Table Creation

Problem 7:

Now we'll create our own table in our database. Separate the Salaries table by Year, and add it back to the database. (You may use basic python to complete the task, however, should still use SQL to query the data.)

```
for y in ['2011','2012','2013','2014']:
    query = 'select * from salaries group by Year'
    df = pd.read_sql(query, conn)
    df.to_sql(name='Y'+y, con=conn, if_exists='replace')
```

Table Joining

Problem 8:

We'll move on to a new dataset for the next steps. Download the dataset from here (https://www.kaggle.com/datasets/luizpaulodeoliveira/imdb-project-sql) and load the sqlite file same as before. Start by just selecting everything in the "movies" table to see what it looks like.

```
conn = sqlite3.connect("movies.sqlite")
query = 'select * from movies'

df = pd.read_sql(query, conn)
df.head()
```

	id	original_title	budget	popularity	release_date	revenue	title	vote_average	vote_count	overview	tagline	u
0	43597	Avatar	237000000	150	2009-12-10	2787965087	Avatar	7.2	11800	In the 22nd century, a paraplegic Marine is di...	Enter the World of Pandora.	199
1	43598	Pirates of the Caribbean: At World's End	300000000	139	2007-05-19	961000000	Pirates of the Caribbean: At World's End	6.9	4500	Captain Barbossa, long believed to be dead, ha...	At the end of the world, the adventure begins.	2

Problem 9:

Create a dataframe that includes the entire contents of "movies" table as well as the director's name.

```
query = 'select m.*, d.name  from movies m inner join directors d where m.director_id = d.id'

df = pd.read_sql(query, conn)
df.head(5)
```

	id	original_title	budget	popularity	release_date	revenue	title	vote_average	vote_count	overview	tagline	u
0	43597	Avatar	237000000	150	2009-12-10	2787965087	Avatar	7.2	11800	In the 22nd century, a paraplegic Marine is di...	Enter the World of Pandora.	199
1	43598	Pirates of the Caribbean: At World's End	300000000	139	2007-05-19	961000000	Pirates of the Caribbean: At World's End	6.9	4500	Captain Barbossa, long believed to be dead, ha...	At the end of the world, the adventure begins.	2

Analysis

The next few problems will be more involved! You'll need to combine some concepts you've learned. For each cell, show your work. Remember, the answers should be in **a single query**.

Problem 10:

What is the average budget used for the top 10 grossing movies?

```
query = 'select AVG(budget) as top_10_avg_budget from movies where popularity <= 10'

df = pd.read_sql(query, conn)
df.head(5)
```

	top_10_avg_budget
0	9.973870e+06

Problem 11:

Which directors have the highest overall voting average? - show the top 5 directors' name and their average rating

```
query = 'select d.name, m.vote_average from directors d inner join movies m where m.director_id = d.id order by m.vote_average desc'

df = pd.read_sql(query, conn)
df.head(5)
```

	name	vote_average
0	Gary Sinyor	10.0
1	Tim McCanlies	10.0
2	Rohit Jugraj	9.5
3	Lance Hool	9.3
4	Frank Darabont	8.5

Problem 12:

What are the top five directors by how much their average budget is?

```
subquery = 'select director_id, AVG(budget) as avg_budget_dir from movies group by director_id'
query = f'select d.name, m.avg_budget_dir from ({subquery}) m inner join directors d where d.id = m.director_id order by m.avg_budget_dir desc

df = pd.read_sql(query, conn)
df.head(5)
```

	name	avg_budget_dir
0	Byron Howard	2.600000e+08
1	Dan Scanlon	2.000000e+08
2	Lee Unkrich	2.000000e+08
3	David Yates	1.933333e+08
4	Brenda Chapman	1.850000e+08