Data Analysis

Prepared by: Huzaifa Bin Munir

Dated: 28th October 2024

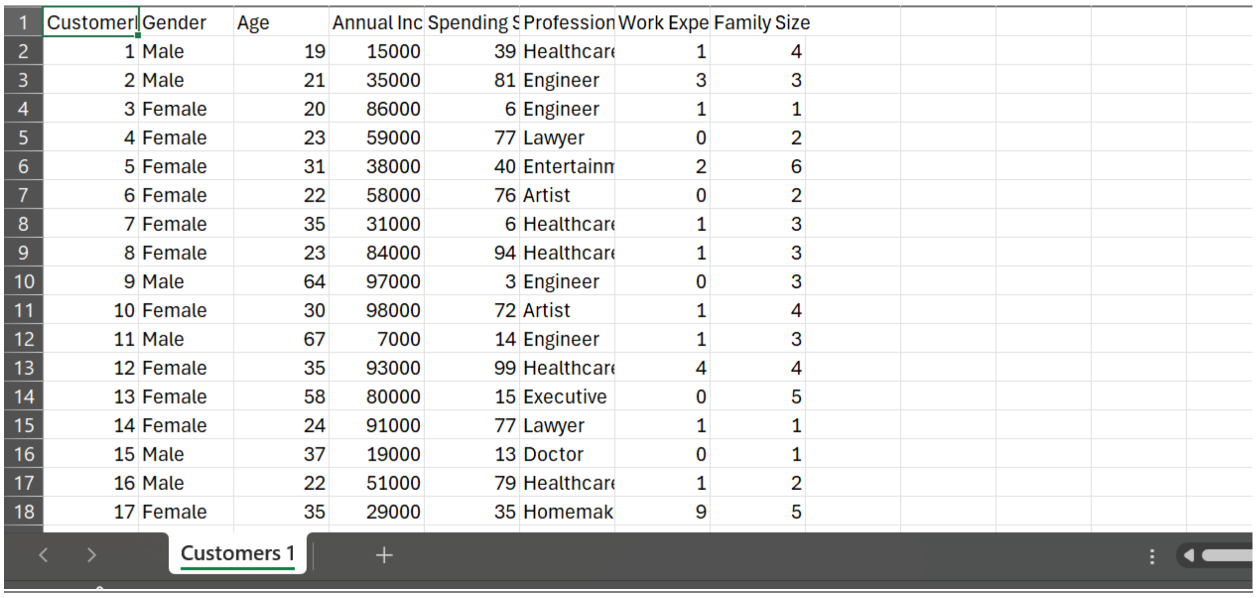
Problem Statement: A customer's dataset has been attached. Your task here is to analyze this data and provide your feedback/ view via exploration.

Instructions:

Follow the following instructions provided:

1. Open the CSV file, you can use any database and import this file as a table with the name of your choice.
2. You can use SQL to search for your answers or to explore the dataset.
3. Provide some key points like types of professions mentioned etc. Average spending etc.
4. Thus provide a detailed analysis against this data (At least 5 points).
5. You can collaborate with other team members and can have the same points.
6. Submit your solution via file share (excel or word). DM me once completed.
7. You can you SQLITE DB here which is a serverless database.
8. Install it from the web following instructions and then import the CSV files as a table.
9. Formulate your queries and run them to collect answers.

Provided Dataset:



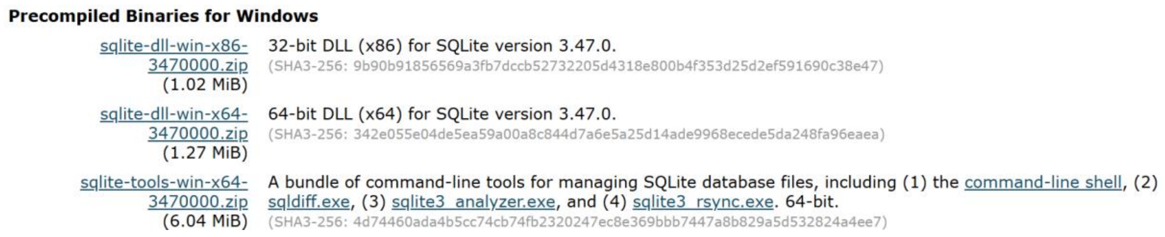
	CustomerI	Gender	Age	Annual Inc	Spending S	Profession	Work Expe	Family Size
1	CustomerI	Gender	Age	Annual Inc	Spending S	Profession	Work Expe	Family Size
2	1	Male	19	15000	39	Healthcare	1	4
3	2	Male	21	35000	81	Engineer	3	3
4	3	Female	20	86000	6	Engineer	1	1
5	4	Female	23	59000	77	Lawyer	0	2
6	5	Female	31	38000	40	Entertainm	2	6
7	6	Female	22	58000	76	Artist	0	2
8	7	Female	35	31000	6	Healthcare	1	3
9	8	Female	23	84000	94	Healthcare	1	3
10	9	Male	64	97000	3	Engineer	0	3
11	10	Female	30	98000	72	Artist	1	4
12	11	Male	67	7000	14	Engineer	1	3
13	12	Female	35	93000	99	Healthcare	4	4
14	13	Female	58	80000	15	Executive	0	5
15	14	Female	24	91000	77	Lawyer	1	1
16	15	Male	37	19000	13	Doctor	0	1
17	16	Male	22	51000	79	Healthcare	1	2
18	17	Female	35	29000	35	Homemak	9	5

Data Analysis:

SQLite is a lightweight serverless database that is easy to use for small scale tasks.

I. Step1:

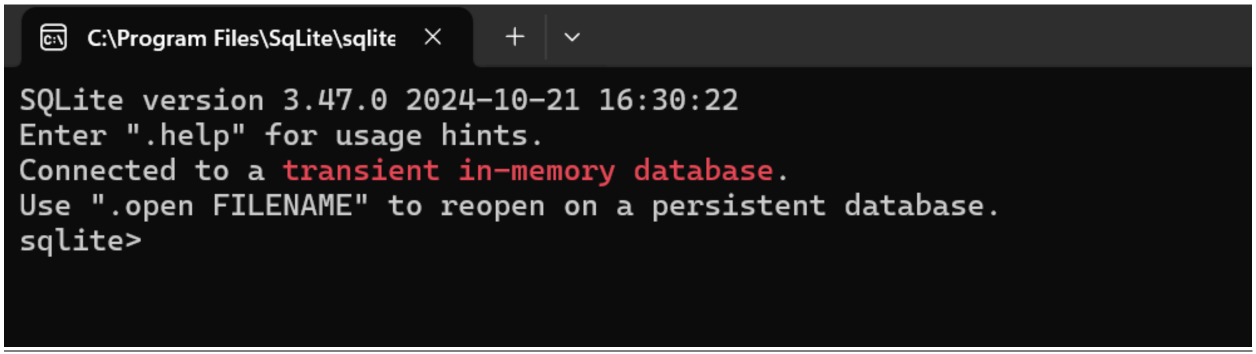
- Install SQLite from the web if it is not installed on your PC.
- Go to the official SQLite website and according to your operating system select the respective option.
- Under the **Precompiled Binaries for Windows** section, download the zip file for sqlite-tools-win-x64-3470000.zip (3rd one).



Precompiled Binaries for Windows

sqlite-dll-win-x86-3470000.zip (1.02 MiB)	32-bit DLL (x86) for SQLite version 3.47.0. (SHA3-256: 9b90b91856569a3fb7dccb52732205d4318e800b4f353d25d2ef591690c38e47)
sqlite-dll-win-x64-3470000.zip (1.27 MiB)	64-bit DLL (x64) for SQLite version 3.47.0. (SHA3-256: 342e055e04de5ea59a00a8c844d7a6e5a25d14ade9968ecede5da248fa96eaea)
sqlite-tools-win-x64-3470000.zip (6.04 MiB)	A bundle of command-line tools for managing SQLite database files, including (1) the command-line shell, (2) sqldiff.exe, (3) sqlite3_analyzer.exe, and (4) sqlite3_rsync.exe. 64-bit. (SHA3-256: 4d74460ada4b5cc74cb74fb2320247ec8e369bbb7447a8b829a5d532824a4ee7)

- After downloading it, now let's set up SQLite. Create a folder by the name of SQLite in the program files (recommended, not necessary) of your C:/ drive and then extract the zip file to this folder.
- Open the folder where you extracted the files. Here, you will see 4 files: sqlite3_rsync.exe, sqlite3_analyzer.exe, sqlite3.exe, sqlite3.exe.
- Double Click to open the sqlite3.exe file and it will open an SQLite command line interface.



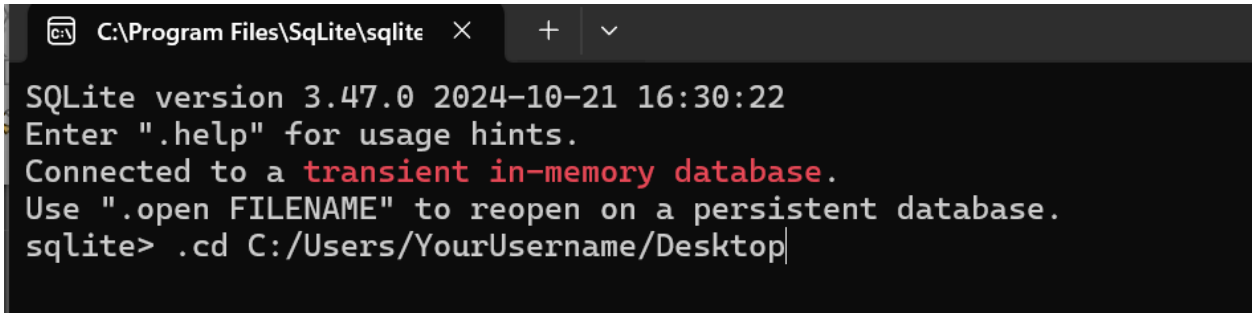
```
SQLite version 3.47.0 2024-10-21 16:30:22
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Note: The above work is for Windows. Kindly, check your OS requirements before downloading and setting.

II. Step2:

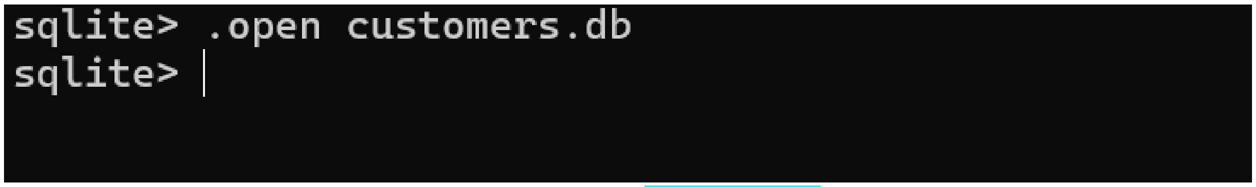
Now create a persistent database.

- To do this, first make sure you are in the right folder where you have full permissions.
- Firstly, change the directory to **Desktop.** For this, type the following command in the SQLite command line interface.



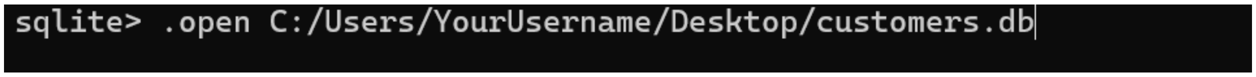
```
SQLite version 3.47.0 2024-10-21 16:30:22
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .cd C:/Users/YourUsername/Desktop
```

- Now, create the database by typing the following command.



```
sqlite> .open customers.db
sqlite>
```

- Once the command runs successfully, a file called customers.db should appear on your **Desktop** or the specified directory. You can name the file anything you like.
- You can also perform the same task with the following command.



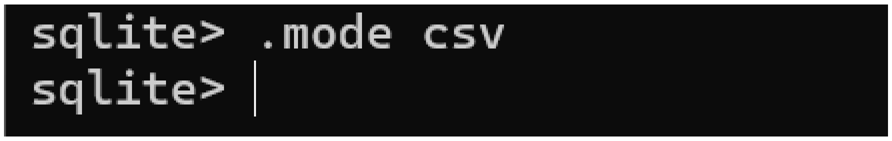
```
sqlite> .open C:/Users/YourUsername/Desktop/customers.db
```

Note: Make sure that you are in a directory where you have permission to create files. The simplest way to avoid this issue is to navigate to a folder like your **Desktop** or **Documents** where you have full access.

III. Step3:

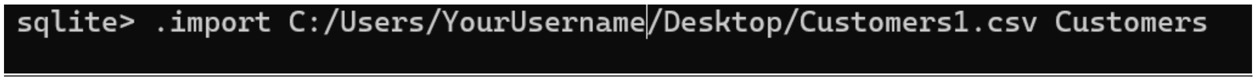
Now, import the CSV file.

- To perform this task, first set the mode to CSV by following command.



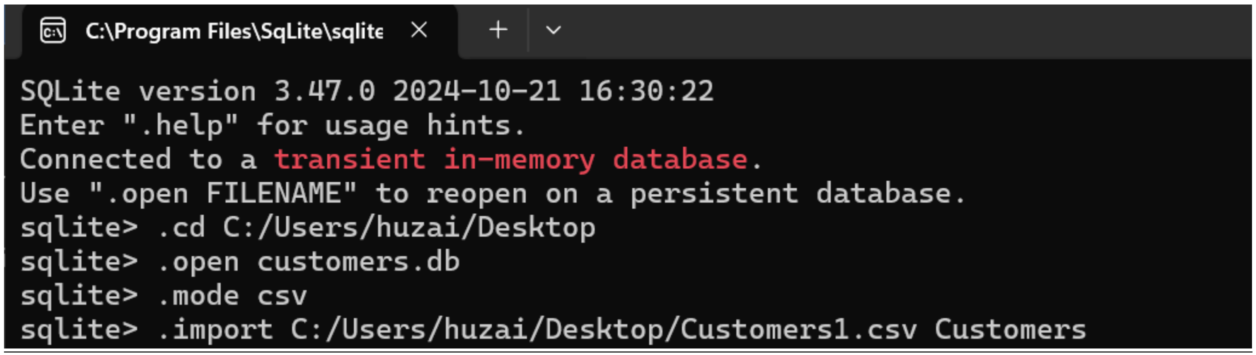
```
sqlite> .mode csv
sqlite>
```

- Now, import your CSV file. (Since, I kept it in my Desktop, you can do the same or choose your preferred directory.)



```
sqlite> .import C:/Users/YourUsername/Desktop/Customers1.csv Customers
```

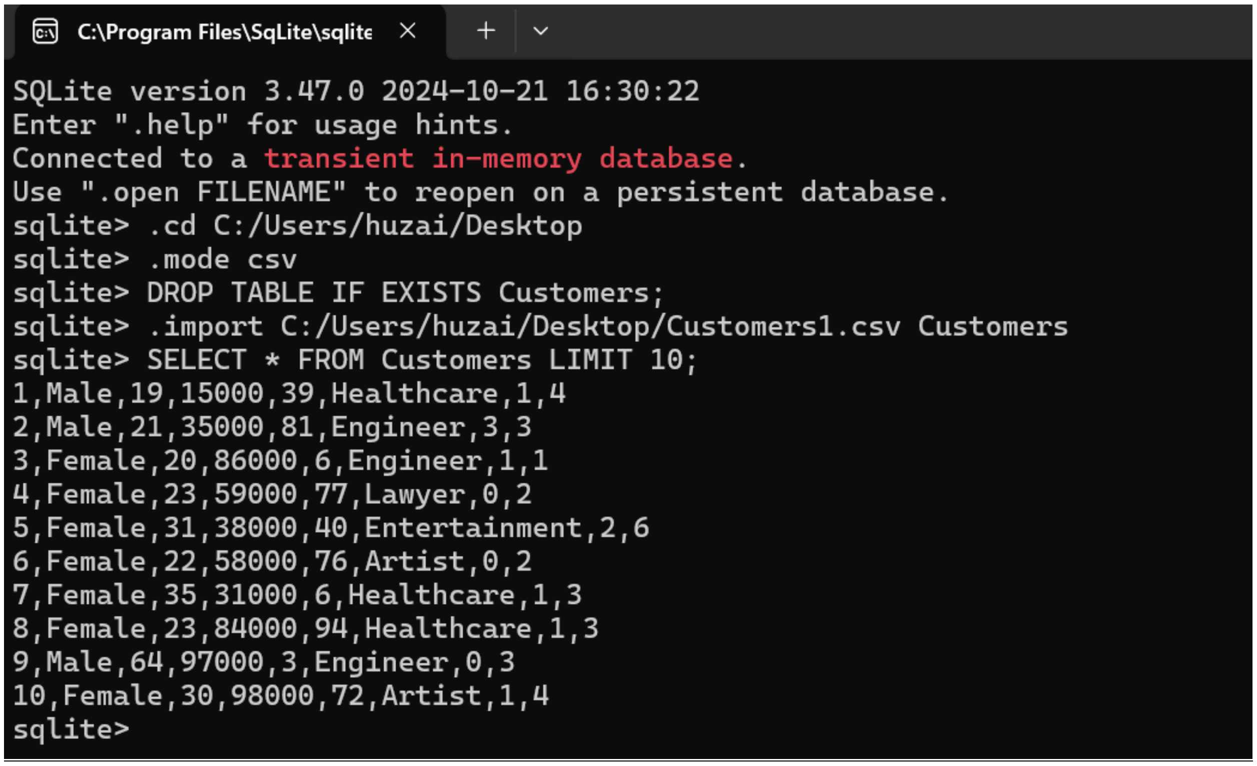
- The CSV is imported into a **table** (in this case, named Customers) inside your SQLite database (customers.db). The original CSV file(Customers1.csv) itself remains unchanged, but its data is copied into the SQLite table(Customers).



```
SQLite version 3.47.0 2024-10-21 16:30:22
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .cd C:/Users/huzai/Desktop
sqlite> .open customers.db
sqlite> .mode csv
sqlite> .import C:/Users/huzai/Desktop/Customers1.csv Customers
```

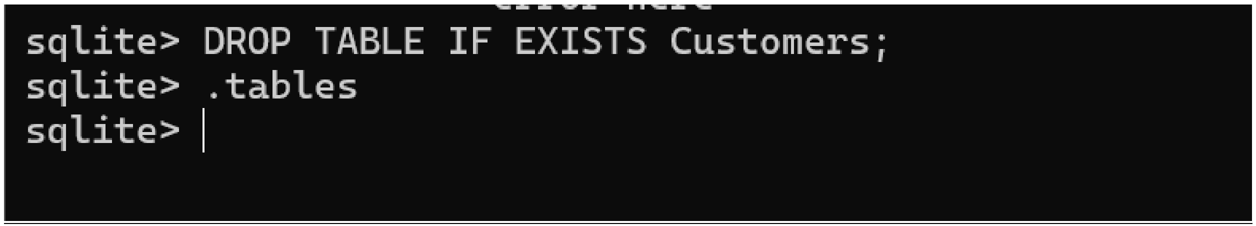
IV. Step4:

Now, verify if the above commands were successful.



```
SQLite version 3.47.0 2024-10-21 16:30:22
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .cd C:/Users/huzai/Desktop
sqlite> .mode csv
sqlite> DROP TABLE IF EXISTS Customers;
sqlite> .import C:/Users/huzai/Desktop/Customers1.csv Customers
sqlite> SELECT * FROM Customers LIMIT 10;
1,Male,19,15000,39,Healthcare,1,4
2,Male,21,35000,81,Engineer,3,3
3,Female,20,86000,6,Engineer,1,1
4,Female,23,59000,77,Lawyer,0,2
5,Female,31,38000,40,Entertainment,2,6
6,Female,22,58000,76,Artist,0,2
7,Female,35,31000,6,Healthcare,1,3
8,Female,23,84000,94,Healthcare,1,3
9,Male,64,97000,3,Engineer,0,3
10,Female,30,98000,72,Artist,1,4
sqlite>
```

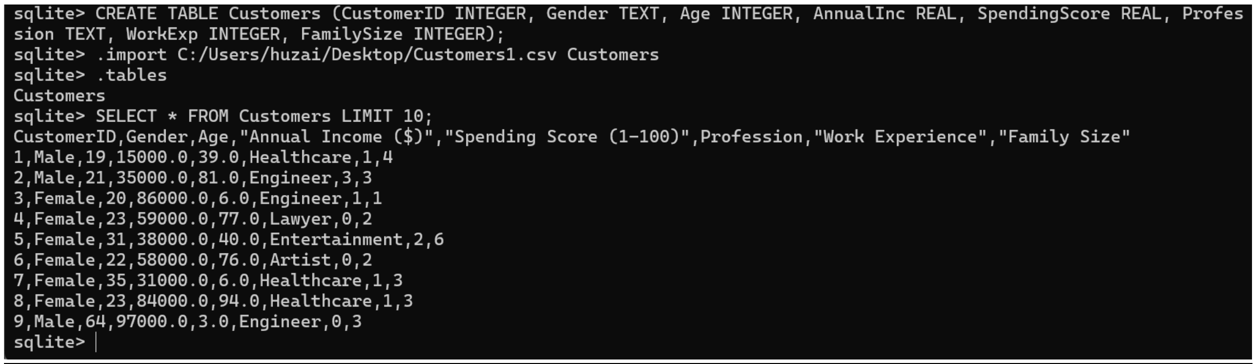
- If it doesn't work, you can try creating the table manually and then copying the data to it. I will first drop the table and then check if there are any tables.



```
sqlite> DROP TABLE IF EXISTS Customers;
sqlite> .tables
sqlite>
```

If there were, the output would have been "Customers".

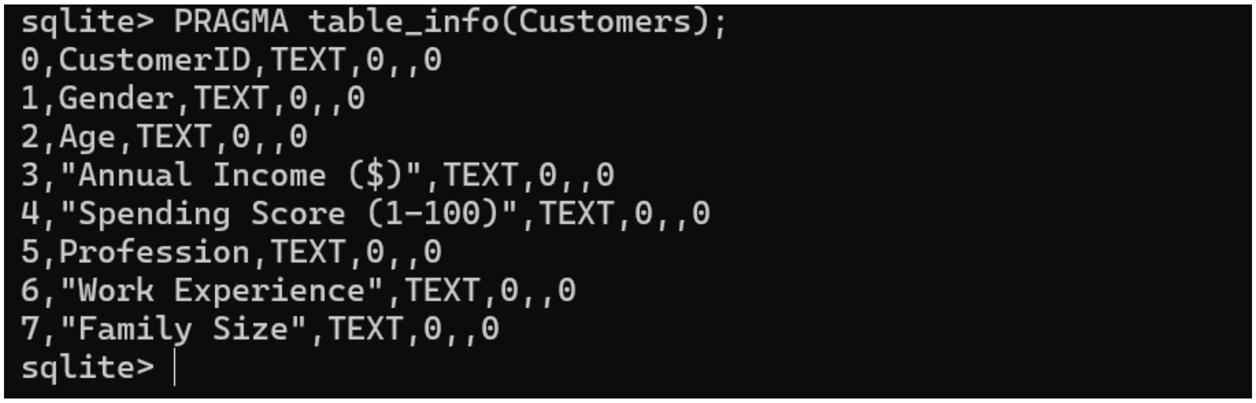
- Now, we will be manually creating the table and importing all of the data into it. Also, check with '.tables' command to verify.



```
sqlite> CREATE TABLE Customers (CustomerID INTEGER, Gender TEXT, Age INTEGER, AnnualInc REAL, SpendingScore REAL, Profes
sion TEXT, WorkExp INTEGER, FamilySize INTEGER);
sqlite> .import C:/Users/huzai/Desktop/Customers1.csv Customers
sqlite> .tables
Customers
sqlite> SELECT * FROM Customers LIMIT 10;
CustomerID,Gender,Age,"Annual Income ($)","Spending Score (1-100)",Profession,"Work Experience","Family Size"
1,Male,19,15000.0,39.0,Healthcare,1,4
2,Male,21,35000.0,81.0,Engineer,3,3
3,Female,20,86000.0,6.0,Engineer,1,1
4,Female,23,59000.0,77.0,Lawyer,0,2
5,Female,31,38000.0,40.0,Entertainment,2,6
6,Female,22,58000.0,76.0,Artist,0,2
7,Female,35,31000.0,6.0,Healthcare,1,3
8,Female,23,84000.0,94.0,Healthcare,1,3
9,Male,64,97000.0,3.0,Engineer,0,3
sqlite>
```

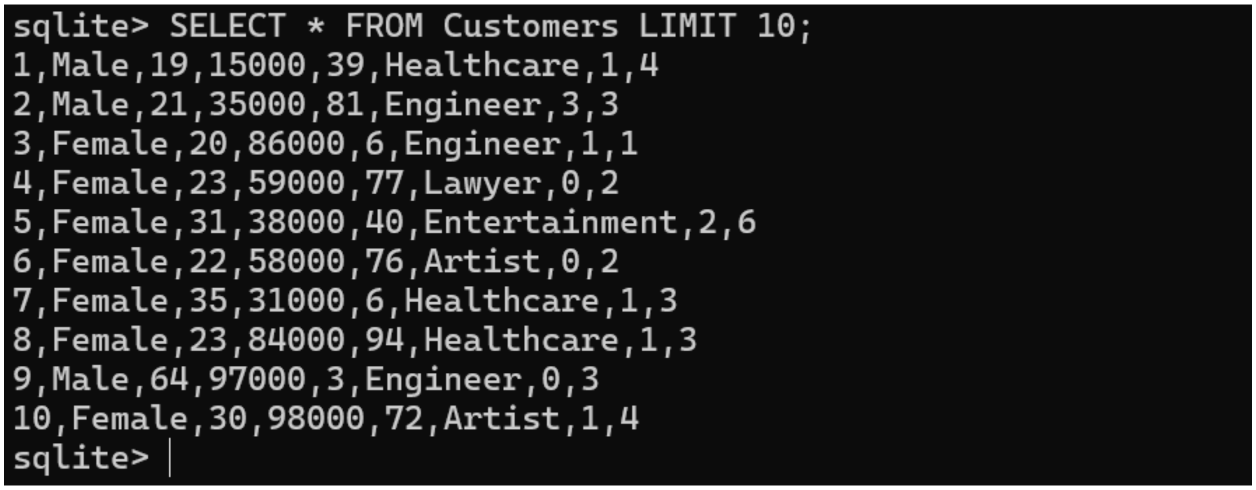
V. Step5:

- Check the structure of the imported table.



```
sqlite> PRAGMA table_info(Customers);
0,CustomerID,TEXT,0,,0
1,Gender,TEXT,0,,0
2,Age,TEXT,0,,0
3,"Annual Income ($)",TEXT,0,,0
4,"Spending Score (1-100)",TEXT,0,,0
5,Profession,TEXT,0,,0
6,"Work Experience",TEXT,0,,0
7,"Family Size",TEXT,0,,0
sqlite>
```

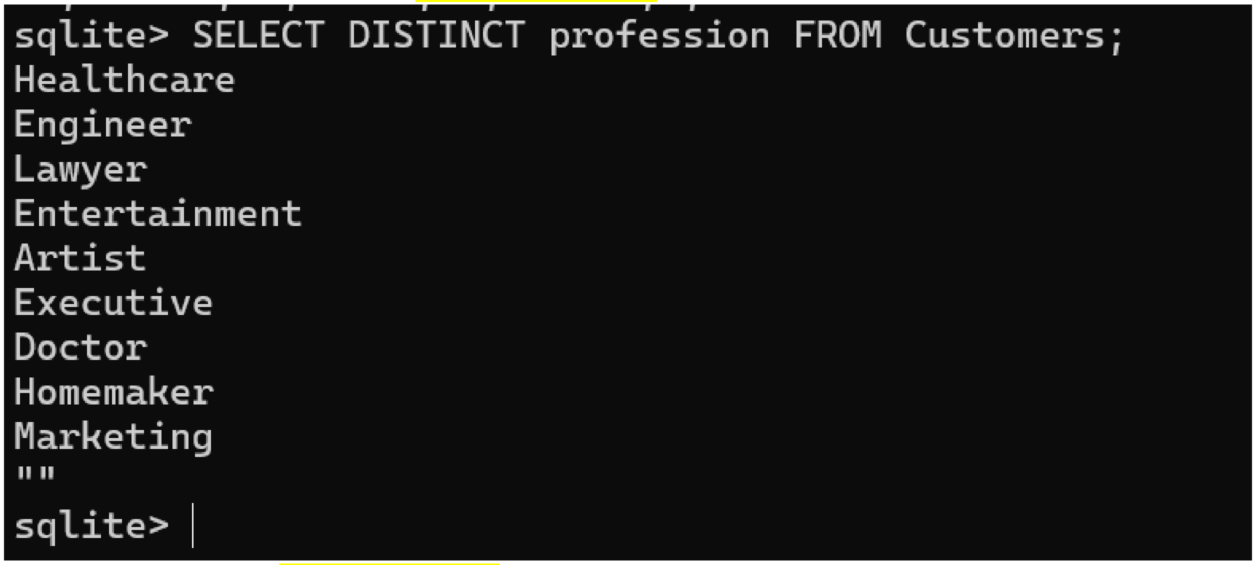
- To view the first 10 rows of the table



```
sqlite> SELECT * FROM Customers LIMIT 10;
1,Male,19,15000,39,Healthcare,1,4
2,Male,21,35000,81,Engineer,3,3
3,Female,20,86000,6,Engineer,1,1
4,Female,23,59000,77,Lawyer,0,2
5,Female,31,38000,40,Entertainment,2,6
6,Female,22,58000,76,Artist,0,2
7,Female,35,31000,6,Healthcare,1,3
8,Female,23,84000,94,Healthcare,1,3
9,Male,64,97000,3,Engineer,0,3
10,Female,30,98000,72,Artist,1,4
sqlite>
```

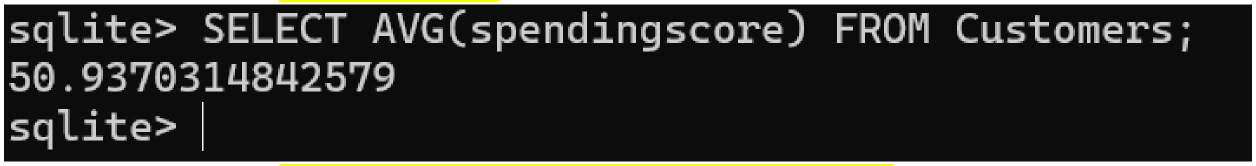
VI. <u>Step6:</u>

- If you want to find the number of unique professions in the data, perform the following.



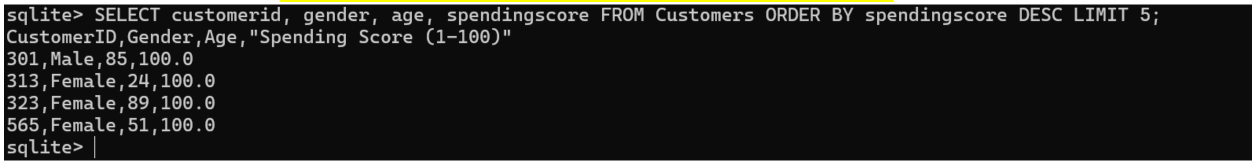
```
sqlite> SELECT DISTINCT profession FROM Customers;
Healthcare
Engineer
Lawyer
Entertainment
Artist
Executive
Doctor
Homemaker
Marketing
""
sqlite>
```

- If you want to find the average spending of the customers, then do the following.



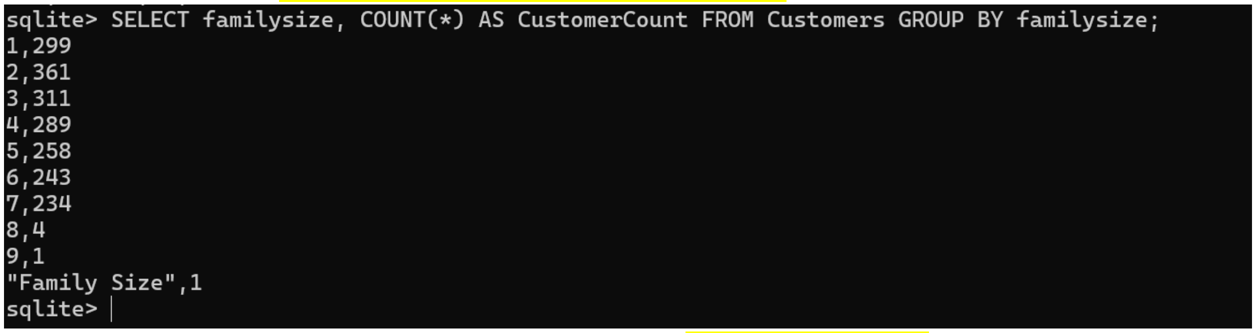
```
sqlite> SELECT AVG(spendingscore) FROM Customers;
50.9370314842579
sqlite>
```

- If you want to find the highest spenders based on their genders and age, then do this.



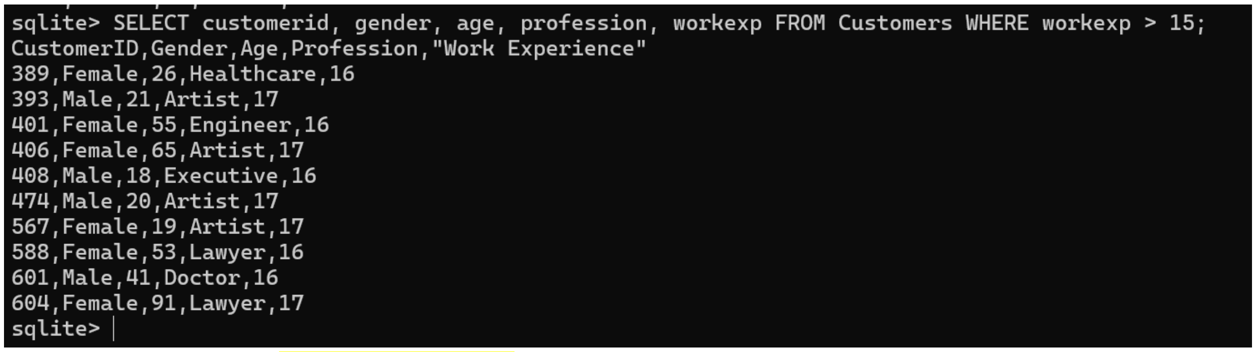
```
sqlite> SELECT customerid, gender, age, spendingscore FROM Customers ORDER BY spendingscore DESC LIMIT 5;
CustomerID,Gender,Age,"Spending Score (1-100)"
301,Male,85,100.0
313,Female,24,100.0
323,Female,89,100.0
565,Female,51,100.0
sqlite>
```

- If you want to find the customer count based on the family size, run this query.



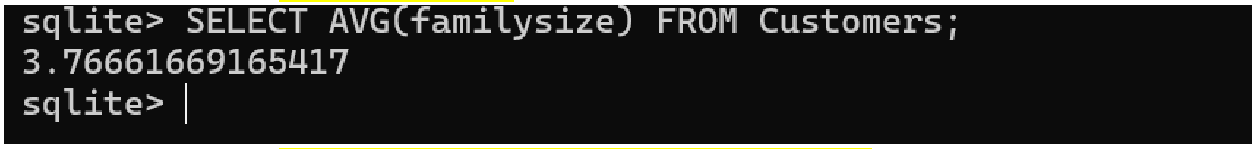
```
sqlite> SELECT familysize, COUNT(*) AS CustomerCount FROM Customers GROUP BY familysize;
1,299
2,361
3,311
4,289
5,258
6,243
7,234
8,4
9,1
"Family Size",1
sqlite>
```

- If you want to find customers with a certain number of years of experience, e.g., 15 years.



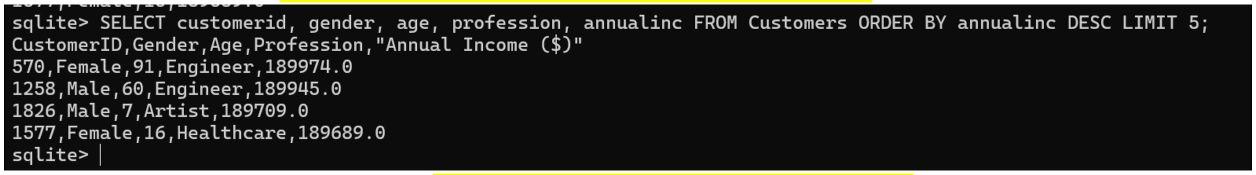
```
sqlite> SELECT customerid, gender, age, profession, workexp FROM Customers WHERE workexp > 15;
CustomerID,Gender,Age,Profession,"Work Experience"
389,Female,26,Healthcare,16
393,Male,21,Artist,17
401,Female,55,Engineer,16
406,Female,65,Artist,17
408,Male,18,Executive,16
474,Male,20,Artist,17
567,Female,19,Artist,17
588,Female,53,Lawyer,16
601,Male,41,Doctor,16
604,Female,91,Lawyer,17
sqlite>
```

- If you want to find the average family size, then write this query in SQLite CLI.



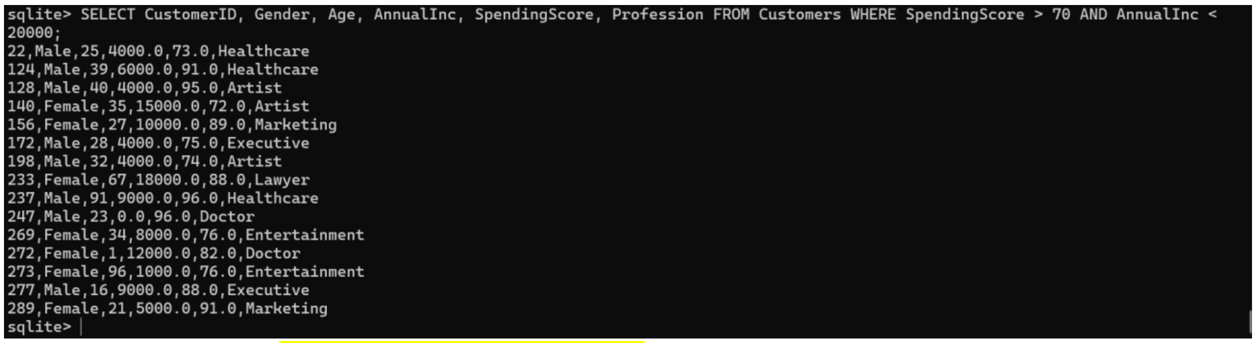
```
sqlite> SELECT AVG(familysize) FROM Customers;
3.76661669165417
sqlite>
```

- If you want to find the highest earners based on their genders and age, then do this.



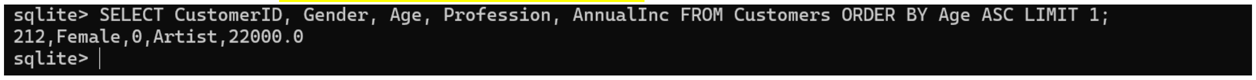
```
sqlite> SELECT customerid, gender, age, profession, annualinc FROM Customers ORDER BY annualinc DESC LIMIT 5;
CustomerID,Gender,Age,Profession,"Annual Income ($)"
570,Female,91,Engineer,189974.0
1258,Male,60,Engineer,189945.0
1826,Male,7,Artist,189709.0
1577,Female,16,Healthcare,189689.0
sqlite>
```

- If you want to find customers with a high spending score but low income based on their genders and age, then do this.

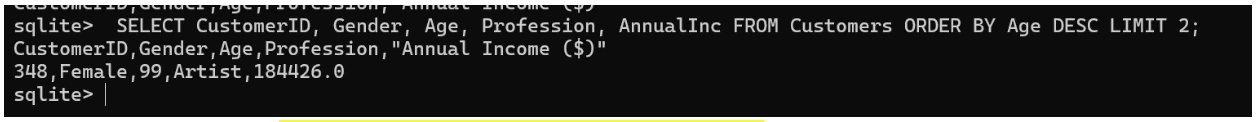


```
sqlite> SELECT CustomerID, Gender, Age, AnnualInc, SpendingScore, Profession FROM Customers WHERE SpendingScore > 70 AND AnnualInc <
20000;
22,Male,25,4000.0,73.0,Healthcare
124,Male,39,6000.0,91.0,Healthcare
128,Male,40,4000.0,95.0,Artist
140,Female,35,15000.0,72.0,Artist
156,Female,27,10000.0,89.0,Marketing
172,Male,28,4000.0,75.0,Executive
198,Male,32,4000.0,74.0,Artist
233,Female,67,18000.0,88.0,Lawyer
237,Male,91,9000.0,96.0,Healthcare
247,Male,23,0.0,96.0,Doctor
269,Female,34,8000.0,76.0,Entertainment
272,Female,1,12000.0,82.0,Doctor
273,Female,96,1000.0,76.0,Entertainment
277,Male,16,9000.0,88.0,Executive
289,Female,21,5000.0,91.0,Marketing
sqlite>
```

- If you want to find the youngest & oldest customers, then do this.

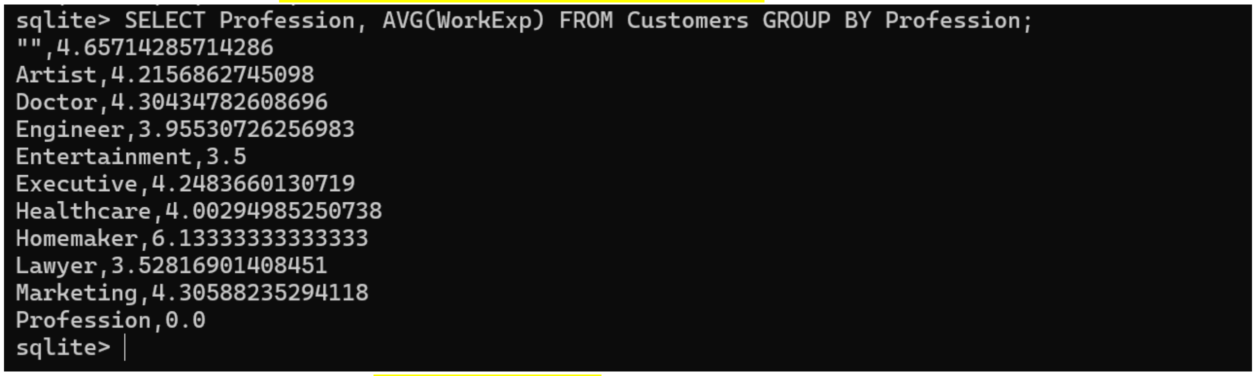


```
sqlite> SELECT CustomerID, Gender, Age, Profession, AnnualInc FROM Customers ORDER BY Age ASC LIMIT 1;
212,Female,0,Artist,22000.0
sqlite>
```



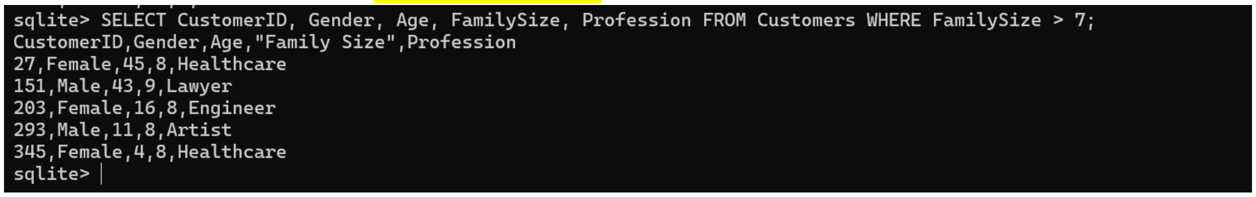
```
sqlite>  SELECT CustomerID, Gender, Age, Profession, AnnualInc FROM Customers ORDER BY Age DESC LIMIT 2;
CustomerID,Gender,Age,Profession,"Annual Income ($)"
348,Female,99,Artist,184426.0
sqlite>
```

- If you want to find the average work experience by profession, then do this.



```
sqlite> SELECT Profession, AVG(WorkExp) FROM Customers GROUP BY Profession;
"",4.65714285714286
Artist,4.2156862745098
Doctor,4.30434782608696
Engineer,3.95530726256983
Entertainment,3.5
Executive,4.2483660130719
Healthcare,4.00294985250738
Homemaker,6.13333333333333
Lawyer,3.52816901408451
Marketing,4.30588235294118
Profession,0.0
sqlite>
```

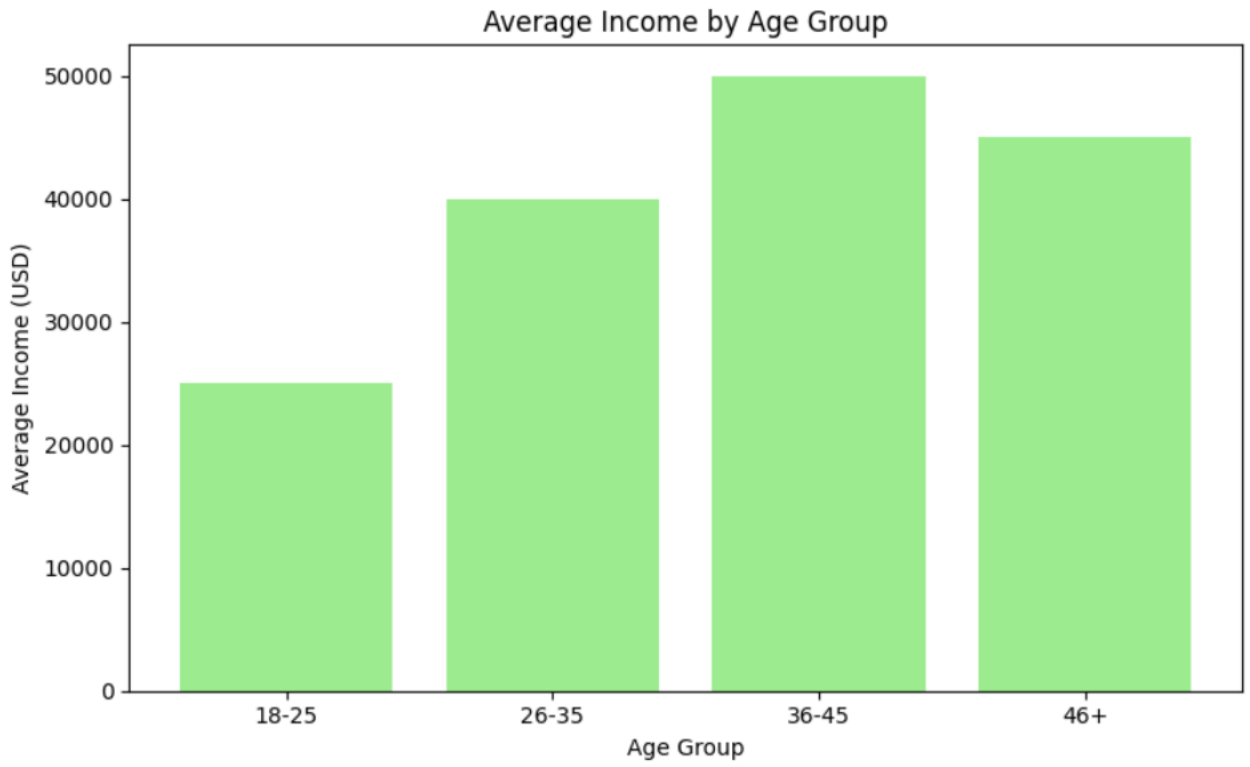
- If you want to find customers with large families, then do this.



```
sqlite> SELECT CustomerID, Gender, Age, FamilySize, Profession FROM Customers WHERE FamilySize > 7;
CustomerID,Gender,Age,"Family Size",Profession
27,Female,45,8,Healthcare
151,Male,43,9,Lawyer
203,Female,16,8,Engineer
293,Male,11,8,Artist
345,Female,4,8,Healthcare
sqlite>
```

Conclusion:

The data analysis reveals key insights into customer demographics, spending behavior, and income distribution across various professions, age groups, and family sizes. Notably, high earners tend to have higher spending scores, and significant variations exist in work experience and family size across professions.
The analysis shows that healthcare professionals have the highest average spending score, while customers aged 26-35 have the highest spending across all age groups. Additionally, customers with large families (5+ members) tend to have lower average spending scores despite higher annual incomes.



Average Income by Age Group
Average Income (USD)
50000
40000
30000
20000
10000
0
18-25
26-35
36-45
46+
Age Group