TASK_2 - PHARMA DATA ANALYSIS

September 30, 2024

1 TASK 2 : PHARMA DATA ANALYTICS

2 1. Connect Notebook with MySQL Database

[1]:
```
# Import required libraries
import pandas as pd
import mysql.connector
```

[2]:
```
# MySQL database connection details
host = 'YOUR_HOST_NAME'
user = 'YOUR_USER_NAME'
password = 'YOUR_PASSWORD'
```

[3]:
```
# CSV file path
csv_file = 'pharma_data.csv'  # Replace 'your_file.csv' with your CSV file path

# Read the first row of the CSV to extract column names
with open(csv_file, 'r') as file:
    first_line = file.readline().strip()  # Read the first line
    column_names = first_line.split(',')  # Assuming columns are comma-separated
    file.close()
```

[4]:
```
# Create MySQL connection
conn = mysql.connector.connect(host=host, user=user, password=password)
cursor = conn.cursor()
```

[5]:
```
# Name of database
database = str(input("Enter database name: "))

# Create the database if it doesn't exist
create_db_query = f"CREATE DATABASE IF NOT EXISTS {database}"
cursor.execute(create_db_query)

# Close the connection as the database is created
conn.close()
```

```
Enter database name: pharmaanalytics
```

[6]:
```
# Reconnect to the newly created or existing database
conn = mysql.connector.connect(host=host, user=user, password=password,␣
 ↪database=database)
cursor = conn.cursor()
```

[9]:
```
column_names
```

[9]:
```
['Distributor',
 'Customer Name',
 'City',
 'Country',
 'Latitude',
 'Longitude',
 'Channel',
 'Sub-channel',
 'Product Name',
 'Product Class',
 'Quantity',
 'Price',
 'Sales',
 'Month',
 'Year',
 'Name of Sales Rep',
 'Manager',
 'Sales Team']
```

[13]:
```
# Ask for table name
table_name = input("Enter table name: ")

# Create table with extracted column names
create_table_query = f"CREATE TABLE IF NOT EXISTS `{table_name}` ({', '.
 ↪join([f'`{col}` TEXT' for col in column_names])})"
cursor.execute(create_table_query)
```

```
Enter table name: pharma_table
```

[14]:
```
# Read the CSV file into a pandas DataFrame
df = pd.read_csv(csv_file)
df.head()
```

[14]:

	Distributor	Customer Name	City	\
0	Gottlieb-Cruickshank	Zieme, Doyle and Kunze	Lublin	
1	Gottlieb-Cruickshank	Feest PLC	Åšwiecie	
2	Gottlieb-Cruickshank	Medhurst-Beer Pharmaceutical Limited	Rybnik	
3	Gottlieb-Cruickshank	Barton Ltd Pharma Plc	CzeladÅº	
4	Gottlieb-Cruickshank	Keeling LLC Pharmacy	Olsztyn	

```
  Country  Latitude  Longitude   Channel  Sub-channel            Product Name  \
0  Poland   51.2333    22.5667  Hospital      Private              Topipizole
1  Poland   53.4167    18.4333  Pharmacy       Retail            Choriotrisin
2  Poland   50.0833    18.5000  Pharmacy  Institution               Acantaine
3  Poland   50.3333    19.0833  Hospital      Private    Lioletine Refliruvax
4  Poland   53.7800    20.4942  Pharmacy       Retail  Oxymotroban Fexoformin

      Product Class  Quantity  Price   Sales    Month  Year Name of Sales Rep  \
0  Mood Stabilizers       4.0    368  1472.0  January  2018      Mary Gerrard
1       Antibiotics       7.0    591  4137.0  January  2018     Jessica Smith
2       Antibiotics      30.0     66  1980.0  January  2018      Steve Pepple
3        Analgesics       6.0    435  2610.0  January  2018      Mary Gerrard
4        Analgesics      20.0    458  9160.0  January  2018           Anne Wu

         Manager Sales Team
0  Britanny Bold      Delta
1  Britanny Bold      Delta
2    Tracy Banks      Bravo
3  Britanny Bold      Delta
4  Britanny Bold      Delta
```

[15]:
```
# Checking shape of dataset
df.shape
```

[15]:
```
(254082, 18)
```

[16]:
```
# Checking column names
df.columns
```

[16]:
```
Index(['Distributor', 'Customer Name', 'City', 'Country', 'Latitude',
       'Longitude', 'Channel', 'Sub-channel', 'Product Name', 'Product Class',
       'Quantity', 'Price', 'Sales', 'Month', 'Year', 'Name of Sales Rep',
       'Manager', 'Sales Team'],
      dtype='object')
```

[18]:
```
# Insert data into the created table
for _, row in df.iterrows():
    # Handle NaN values by converting them to None
    row = [None if pd.isna(value) else value for value in row]

    # Prepare and execute the insert query
    insert_query = f"INSERT INTO `{table_name}` ({', '.join([f'`{col}`' for col␣
 ↪in column_names])}) VALUES ({', '.join(['%s'] * len(column_names))})"
    cursor.execute(insert_query, tuple(row))

# Commit changes and close the connection
conn.commit()
```

```
conn.close()

print(f"Table '{table_name}' created and data loaded successfully.")
```

```
Table 'pharma_table' created and data loaded successfully.
```

3 2. Excess Database Information as per your Requirement

[21]:
```
# Pass the query through this function ant get your results
from IPython.display import display

def execute_query(query):
    # MySQL database connection details
    host = 'YOUR_HOST_NAME'
    user = 'YOUR_USER_NAME'
    password = 'YOUR_PASSWORD'
    database = 'pharmaanalytics'  # Replace with your database name

    # Connect to MySQL
    conn = mysql.connector.connect(host=host, user=user, password=password,␣
 ↪database=database)

    # Create a cursor
    cursor = conn.cursor()

    try:
        # Execute the query
        cursor.execute(query)

        # Fetch all results
        results = cursor.fetchall()


        # Display results in a table format
        if results:
            header = [desc[0] for desc in cursor.description]
            data = list(results)
            df = pd.DataFrame(data, columns=header)
            display(df)
        else:
            print("No results found.")

    except mysql.connector.Error as error:
        print(f"Error: {error.msg}")

    finally:
```

```
        # Close the cursor and connection
        cursor.close()
        conn.close()
```

3.1 1. Retrieve all columns for all records in the dataset.

[22]:
```
# Define the query
query = "SELECT * FROM pharma_table;"
# Call the function with the query
execute_query(query)
```

```
                Distributor                                Customer Name  \
0       Gottlieb-Cruickshank                 Zieme, Doyle and Kunze
1       Gottlieb-Cruickshank                            Feest PLC
2       Gottlieb-Cruickshank      Medhurst-Beer Pharmaceutical Limited
3       Gottlieb-Cruickshank                  Barton Ltd Pharma Plc
4       Gottlieb-Cruickshank                   Keeling LLC Pharmacy
…                       …                                       …
254077    Bashirian-Kassulke  Koch, Borer and Hagenes Pharmaceutical Ltd
254078    Bashirian-Kassulke              Hane Ltd Pharmaceutical Ltd
254079    Bashirian-Kassulke                   Harris-Conroy Pharmacy
254080    Bashirian-Kassulke                   Balistreri Group Pharm
254081    Bashirian-Kassulke          Heathcote, Kovacek and Parker

                  City  Country Latitude Longitude   Channel  Sub-channel  \
0               Lublin   Poland  51.2333   22.5667  Hospital      Private
1            Åšwiecie   Poland  53.4167   18.4333  Pharmacy       Retail
2              Rybnik   Poland  50.0833      18.5  Pharmacy  Institution
3            CzeladÅº   Poland  50.3333   19.0833  Hospital      Private
4             Olsztyn   Poland    53.78   20.4942  Pharmacy       Retail
…                 …       …       …         …         …            …
254077          Lauf  Germany  49.5103   11.2772  Hospital      Private
254078       Aichach  Germany    48.45   11.1333  Hospital      Private
254079  Wilhelmshaven  Germany  53.5167    8.1333  Pharmacy       Retail
254080     BÃ¶blingen  Germany  48.6833       9.0  Hospital   Government
254081           Hof  Germany  50.3167   11.9167  Pharmacy       Retail

                  Product Name     Product Class Quantity Price     Sales  \
0                   Topipizole  Mood Stabilizers      4.0   368    1472.0
1                  Choriotrisin       Antibiotics      7.0   591    4137.0
2                     Acantaine       Antibiotics     30.0    66    1980.0
3          Lioletine Refliruvax        Analgesics      6.0   435    2610.0
4       Oxymotroban Fexoformin        Analgesics     20.0   458    9160.0
…                          …                 …      …     …       …
254077              Pentastrin       Antibiotics    919.0   497  456743.0
254078   Abranatal Lysoprosate       Antiseptics    432.0   681  294192.0
254079                Adideine  Mood Stabilizers    320.0   678  216960.0
```

```
254080             Feruprazole  Mood Stabilizers    565.0   115   64975.0
254081             Feruprazole  Mood Stabilizers   1080.0   115  124200.0

           Month  Year  Name of Sales Rep          Manager Sales Team
0        January  2018       Mary Gerrard    Britanny Bold      Delta
1        January  2018      Jessica Smith    Britanny Bold      Delta
2        January  2018       Steve Pepple      Tracy Banks      Bravo
3        January  2018       Mary Gerrard    Britanny Bold      Delta
4        January  2018            Anne Wu    Britanny Bold      Delta
…          …    …                 …               …         …
254077  December  2020  Thompson Crawford   James Goodwill       Alfa
254078  December  2020            Anne Wu    Britanny Bold      Delta
254079  December  2020   Abigail Thompson      Tracy Banks      Bravo
254080  December  2020       Stella Given  Alisha Cordwell    Charlie
254081  December  2020           Alan Ray   James Goodwill       Alfa

[254082 rows x 18 columns]
```

3.2 2. How many unique countries are represented in the dataset?

[26]:
```
# Define the query
query = "SELECT COUNT(DISTINCT country) AS Unique_Countries FROM pharma_table;"
# Call the function with the query
execute_query(query)
```

```
   Unique_Countries
0                 2
```

3.3 3. Select the names of all the customers on the 'Retail' channel.

[27]:
```
# Define the query
query = "SELECT `Customer Name` AS Customers_Retail FROM pharma_table WHERE␣
 ↪`Sub-channel` = 'Retail';"
# Call the function with the query
execute_query(query)
```

```
                            Customers_Retail
0                                Feest PLC
1                          Keeling LLC Pharmacy
2               Blick, Pacocha and Schowalter
3                         Leuschke PLC Pharmacy
4       McClure, Zemlak and Dibbert Pharma Plc
…                                       …
68346                            Paucek PLC
68347                   Walsh-Brown Pharma Plc
68348  Schinner, Gaylord and Treutel Pharmacy
68349                   Harris-Conroy Pharmacy
68350           Heathcote, Kovacek and Parker
```

[68351 rows x 1 columns]

3.4 4. Find the total quantity sold for the ' Antibiotics' product class.

[28]:
```
# Define the query
query = "SELECT SUM(Quantity) AS Total_Quantity FROM pharma_table WHERE␣
 ↪`Product Class` = 'Antibiotics';"
# Call the function with the query
execute_query(query)
```

```
   Total_Quantity
0    4.154322e+06
```

3.5 5. List all the distinct months present in the dataset.

[29]:
```
# Define the query
query = "SELECT DISTINCT Month FROM pharma_table;"
# Call the function with the query
execute_query(query)
```

```
        Month
0     January
1    February
2       March
3       April
4         May
5        June
6        July
7      August
8   September
9     October
10   November
11   December
```

3.6 6. Calculate the total sales for each year.

[30]:
```
# Define the query
query = "SELECT Year, SUM(Sales) AS Total_Sales FROM pharma_table GROUP BY 1;"
# Call the function with the query
execute_query(query)
```

```
   Year   Total_Sales
0  2018  3.506897e+09
1  2017  2.701481e+09
2  2019  2.930937e+09
3  2020  2.659672e+09
```

3.7 7. Find the customer with the highest sales value.

[31]:
```
# Define the query
query = "SELECT `Customer Name`, SUM(Sales) AS Total_Sales FROM pharma_table␣
 ↪GROUP BY 1 ORDER BY 2 DESC LIMIT 1;"
# Call the function with the query
execute_query(query)
```

```
          Customer Name  Total_Sales
0  Mraz-Kutch Pharma Plc   93561780.0
```

3.8 8. Get the names of all employees who are Sales Reps and are managed by 'James Goodwill'.

[33]:
```
# Define the query
query = "SELECT DISTINCT `Name of Sales Rep` AS Employee_Names FROM␣
 ↪pharma_table WHERE Manager = 'James Goodwill';"
# Call the function with the query
execute_query(query)
```

```
     Employee_Names
0  Thompson Crawford
1        Erica Jones
2           Alan Ray
```

3.9 9. Retrieve the top 5 cities with the highest sales.

[34]:
```
# Define the query
query = "SELECT City, SUM(Sales) AS Highest_Sales FROM pharma_table GROUP BY 1␣
 ↪ORDER BY 2 DESC LIMIT 5;"
# Call the function with the query
execute_query(query)
```

```
         City  Highest_Sales
0    Butzbach     93561780.0
1  Baesweiler     64890501.0
2    Cuxhaven     56006680.0
3   Friedberg     52183634.6
4   Altenburg     50885320.0
```

3.10 10. Calculate the average price of products in each sub-channel.

[36]:
```
# Define the query
query = "SELECT `Sub-channel`, AVG(Price) AS Average_Price FROM pharma_table␣
 ↪GROUP BY 1 ORDER BY 2 DESC;"
# Call the function with the query
execute_query(query)
```

```
   Sub-channel  Average_Price
0   Government     413.149440
1       Retail     412.807040
2  Institution     411.954398
3      Private     410.718371
```

3.11 11. Join the 'Employees' table with the 'Sales' table to get the name of the Sales Rep and the corresponding sales records.

[38]:
```
# Define the query
query = "SELECT `Name of Sales Rep`, SUM(Sales) AS Sales_Record FROM␣
 ↪pharma_table GROUP BY 1 ORDER BY 2 DESC;"
# Call the function with the query
execute_query(query)
```

```
    Name of Sales Rep  Sales_Record
0          Jimmy Grey  9.859700e+08
1     Abigail Thompson  9.810570e+08
2        Sheila Stones  9.582039e+08
3         Daniel Gates  9.506586e+08
4              Anne Wu  9.201683e+08
5        Morris Garcia  9.011955e+08
6         Stella Given  8.883409e+08
7        Jessica Smith  8.816984e+08
8         Steve Pepple  8.754500e+08
9         Mary Gerrard  8.752708e+08
10         Erica Jones  8.713722e+08
11  Thompson Crawford   8.669649e+08
12           Alan Ray  8.426372e+08
```

3.12 12. Retrieve all sales made by employees from ' Rendsburg ' in the year 2018.

[39]:
```
# Define the query
query = "SELECT `Name of Sales Rep`, SUM(Sales) AS Total_Sales, `YEAR` FROM␣
 ↪pharma_table WHERE City = 'Rendsburg' AND `Year` = 2018 GROUP BY 1 ORDER BY␣
 ↪2 DESC;"
# Call the function with the query
execute_query(query)
```

```
    Name of Sales Rep  Total_Sales  YEAR
0        Jessica Smith    5059318.0  2018
1        Sheila Stones    1581159.0  2018
2          Erica Jones     980046.0  2018
3        Morris Garcia     405500.0  2018
4              Anne Wu     383869.0  2018
5            Alan Ray      366832.0  2018
6           Jimmy Grey     253399.0  2018
```

```
7         Stella Given     226347.0  2018
8   Thompson Crawford      81915.0  2018
9         Mary Gerrard     74042.0  2018
10   Abigail Thompson      65022.0  2018
11        Daniel Gates     49801.0  2018
12        Steve Pepple      1377.0  2018
```

3.13 13. Calculate the total sales for each product class, for each month, and order the results by year, month, and product class.

[40]:
```
# Define the query
query = "SELECT  `Product Class`, `Month`, `Year`, SUM(Sales) AS Total_Sales␣
 ↪FROM pharma_table GROUP BY 1, 2, 3 ORDER BY `Year`, `Month`, `Product Class`;
 ↪"
# Call the function with the query
execute_query(query)
```

```
        Product Class      Month  Year  Total_Sales
0          Analgesics      April  2017   32223716.0
1          Antibiotics     April  2017   40029226.0
2         Antimalarial     April  2017   17789675.0
3         Antipiretics     April  2017   22868812.0
4          Antiseptics     April  2017   42712211.0
..                 …        …    …           …
283        Antibiotics  September  2020   34985330.0
284       Antimalarial  September  2020   19328987.0
285       Antipiretics  September  2020   30939171.0
286        Antiseptics  September  2020   62348015.0
287  Mood Stabilizers   September  2020   43643444.0

[288 rows x 4 columns]
```

3.14 14. Find the top 3 sales reps with the highest sales in 2019.

[41]:
```
# Define the query
query = "SELECT  `Name of Sales Rep` AS Top_3_Sales_Rep, SUM(Sales) AS␣
 ↪Total_Sales FROM pharma_table WHERE year = 2019 GROUP BY 1 ORDER BY 2 DESC␣
 ↪LIMIT 3;"
# Call the function with the query
execute_query(query)
```

```
  Top_3_Sales_Rep   Total_Sales
0      Jimmy Grey  3.105511e+08
1   Sheila Stones  2.669244e+08
2    Daniel Gates  2.453639e+08
```

3.15 15. Calculate the monthly total sales for each sub-channel, and then calculate the average monthly sales for each sub-channel over the years.

[42]:
```
# Define the query
query = "SELECT `Sub-channel`, `Month`, `Year`, SUM(Sales) AS Total_Sales,␣
 ↪AVG(SUM(Sales)) OVER (PARTITION BY `Sub-channel`, `Month`) AS Average_Sales␣
 ↪FROM pharma_table GROUP BY 1,2,3 ORDER BY 3,2;"
# Call the function with the query
execute_query(query)
```

	Sub-channel	Month	Year	Total_Sales	Average_Sales
0	Government	April	2017	45892380.0	59112240.75
1	Private	April	2017	43680022.0	38498738.50
2	Retail	April	2017	49076812.0	53068274.50
3	Institution	April	2017	50151370.0	49329388.45
4	Institution	August	2017	57379276.0	58881548.75
..	…	…	…	…	…
187	Government	October	2020	46367808.0	56208502.00
188	Institution	September	2020	58791970.0	65737833.75
189	Retail	September	2020	67426858.0	75422382.75
190	Private	September	2020	50514794.0	55488251.75
191	Government	September	2020	58376813.0	60848428.00

[192 rows x 5 columns]

3.16 16. Create a summary report that includes the total sales, average price, and total quantity sold for each product class.

[43]:
```
# Define the query
query = "SELECT `Product Class`, SUM(Sales) AS Total_Sales, AVG(Price) AS␣
 ↪Average_Price, SUM(Quantity) AS Total_Quantity FROM pharma_table GROUP BY 1;"
# Call the function with the query
execute_query(query)
```

	Product Class	Total_Sales	Average_Price	Total_Quantity
0	Mood Stabilizers	2.058910e+09	400.493353	5.169781e+06
1	Antibiotics	1.750277e+09	419.671057	4.154322e+06
2	Analgesics	2.371515e+09	432.571071	5.553144e+06
3	Antiseptics	2.237525e+09	412.396699	5.499913e+06
4	Antipiretics	1.883306e+09	469.047680	4.052544e+06
5	Antimalarial	1.497455e+09	337.667208	4.249075e+06

3.17 17. Find the top 5 customers with the highest sales for each year.

[44]:
```
# Define the query
```

```
query = "WITH Top_Customers AS (SELECT `Customer Name`, Sales, `Year`,␣
 ↪DENSE_RANK() OVER(PARTITION BY year ORDER BY Sales DESC) AS Top_5_Customers␣
 ↪FROM pharma_table) SELECT * FROM Top_Customers WHERE Top_5_Customers <= 5;"
# Call the function with the query
execute_query(query)
```

```
                                      Customer Name      Sales  Year  \
0     Casper, Hyatt and Jakubowski Pharmaceutical Ltd    99990.0  2017
1                                   Konopelski-Blick     99981.0  2017
2                             Kozey-Emmerich Pharmacy    99936.0  2017
3                             Jacobs-Jones Pharma Plc   999000.0  2017
4                  Waelchi LLC Pharmaceutical Limited     9990.0  2017
5              VonRueden-Adams Pharmaceutical Limited     9999.0  2018
6                                   Paucek PLC Pharm     99936.0  2018
7                    Purdy Ltd Pharmaceutical Limited    99936.0  2018
8                  Feest-Kshlerin Pharmaceutical Ltd     99900.0  2018
9                           Armstrong Inc Pharma Plc     9990.0  2018
10                        McKenzie-Zemlak Pharma Plc     9990.0  2018
11                                  Fahey-Flatley        9990.0  2018
12                          Cremin-Zemlak Pharma Plc     99840.0  2018
13                                  Yundt-Crona Pharm     99840.0  2018
14                   Funk, Ratke and Heaney Pharmacy     99840.0  2018
15                               Crist LLC Pharma Plc     9999.0  2019
16                                Turner Ltd Pharmacy   9996000.0  2019
17                               Koch-Osinski Pharmacy    99900.0  2019
18                            Rohan and Sons Pharma Plc   998400.0  2019
19         Leuschke, Waters and Schowalter Pharma Plc     998400.0  2019
20           Pfannerstill, Upton and Balistreri Pharm     99840.0  2019
21                      Balistreri-Watsica Pharma Plc     99840.0  2019
22                 Hirthe, Williamson and Macejkovic      99840.0  2019
23    Casper, Hyatt and Jakubowski Pharmaceutical Ltd     99840.0  2019
24               Reinger, Kihn and Goyette Pharma Plc     99840.0  2019
25                               Reilly Ltd Pharma Plc     9996.0  2020
26  Prohaska, Bogisich and Gutkowski Pharmaceutica…    999360.0  2020
27                                  Wehner-Stehr Pharm    99905.0  2020
28                                   Fritsch LLC Pharm     9990.0  2020
29                                         Bayer LLC      9990.0  2020
30                             Reinger Group Pharma Plc     9990.0  2020
31                                 Crist Inc Pharma Plc     99840.0  2020
32                                   Crona PLC Pharmacy     99840.0  2020
33                       Reynolds, Jast and Mante Pharm     99840.0  2020

    Top_5_Customers
0                 1
1                 2
2                 3
3                 4
```

4	5
5	1
6	2
7	2
8	3
9	4
10	4
11	4
12	5
13	5
14	5
15	1
16	2
17	3
18	4
19	4
20	5
21	5
22	5
23	5
24	5
25	1
26	2
27	3
28	4
29	4
30	4
31	5
32	5
33	5

3.18 18. Calculate the year-over-year growth in sales for each country.

[45]:
```
# Define the query
query = "SELECT Country, `Year`, SUM(Sales) AS Total_Sales, LAG(SUM(Sales), 1,␣
 ↪0) OVER(PARTITION BY Country ORDER BY year) AS Previous_Year_Sales,␣
 ↪SUM(Sales) - LAG(SUM(Sales), 1, 0) OVER(PARTITION BY Country ORDER BY year)␣
 ↪AS Year_Over_Year_Growth FROM pharma_table GROUP BY 1,2 ORDER BY 2,1;"
# Call the function with the query
execute_query(query)
```

	Country	Year	Total_Sales	Previous_Year_Sales	Year_Over_Year_Growth
0	Germany	2017	2.701481e+09	0.000000e+00	2.701481e+09
1	Germany	2018	2.826018e+09	2.701481e+09	1.245368e+08
2	Poland	2018	6.808798e+08	0.000000e+00	6.808798e+08
3	Germany	2019	2.930937e+09	2.826018e+09	1.049196e+08
4	Germany	2020	2.659672e+09	2.930937e+09	-2.712647e+08

3.19 19. List the months with the lowest sales for each year.

[46]:
```
# Define the query
query = "WITH Lowest_Sales AS (SELECT  `Month`, `Year`, SUM(Sales) AS␣
 ↪Total_Sales, DENSE_RANK() OVER(PARTITION BY year ORDER BY SUM(Sales) ASC) AS␣
 ↪Ranks FROM pharma_table GROUP BY 1,2) SELECT * FROM Lowest_Sales WHERE Ranks␣
 ↪= 1;"
# Call the function with the query
execute_query(query)
```

	Month	Year	Total_Sales	Ranks
0	January	2017	151872184.0	1
1	December	2018	214882167.0	1
2	January	2019	97664076.0	1
3	April	2020	135409908.0	1

3.20 20. Calculate the total sales for each sub-channel in each country, and then find the country with the highest total sales for each sub-channel.

[47]:
```
# Define the query
query = "WITH CTE AS (SELECT `Sub-channel`, Country, SUM(Sales) AS Total_Sales␣
 ↪FROM pharma_table GROUP BY 1,2), Highest_Total_Sales AS (SELECT *,␣
 ↪DENSE_RANK() OVER (PARTITION BY `Sub-channel` ORDER BY Total_Sales DESC) AS␣
 ↪Ranks FROM CTE) SELECT * FROM Highest_Total_Sales WHERE Ranks = 1;"
# Call the function with the query
execute_query(query)
```

	Sub-channel	Country	Total_Sales	Ranks
0	Government	Germany	2.920913e+09	1
1	Institution	Germany	2.719605e+09	1
2	Private	Germany	2.315302e+09	1
3	Retail	Germany	3.162287e+09	1