

## Task 2 Query implementation & Visualization using python

May 12, 2023

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import sqlalchemy
```

```
[2]: engine = sqlalchemy.create_engine('mysql+pymysql://root:pass@123@localhost:3306/
    ↪shop')
```

```
[3]: df = pd.read_sql_table('sales_data',engine)
df.head()
```

```
[3]:  id      order_id order_date  ship_date      ship_mode customer_id \
0    1  CA-2017-152156 2017-11-08 2017-11-11    Second Class    CG-12520
1    2  CA-2017-152156 2017-11-08 2017-11-11    Second Class    CG-12520
2    3  CA-2017-138688 2017-06-12 2017-06-16    Second Class    DV-13045
3    4  US-2016-108966 2016-10-11 2016-10-18  Standard Class    SO-20335
4    5  US-2016-108966 2016-10-11 2016-10-18  Standard Class    SO-20335

      customer_name  segment      country      city      state \
0      Claire Gute  Consumer  United States    Henderson  Kentucky
1      Claire Gute  Consumer  United States    Henderson  Kentucky
2  Darrin Van Huff  Corporate  United States  Los Angeles  California
3  Sean O'Donnell  Consumer  United States  Fort Lauderdale  Florida
4  Sean O'Donnell  Consumer  United States  Fort Lauderdale  Florida

      postal_code region      product_id      category sub_category \
0         42420  South  FUR-BO-10001798    Furniture  Bookcases
1         42420  South  FUR-CH-10000454    Furniture    Chairs
2         90036  West  OFF-LA-10000240  Office Supplies    Labels
3         33311  South  FUR-TA-10000577    Furniture    Tables
4         33311  South  OFF-ST-10000760  Office Supplies    Storage

      product_name  sales
0      Bush Somerset Collection Bookcase  261.96
1  Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.94
2  Self-Adhesive Address Labels for Typewriters b...  14.62
3      Bretford CR4500 Series Slim Rectangular Table  957.58
4      Eldon Fold 'N Roll Cart System  22.37
```

[4]: *#Q1: Total number of orders count per year*

```
query=''

SELECT EXTRACT(YEAR FROM order_date) AS year, COUNT(*) AS order_count
FROM sales_data
GROUP BY year
ORDER BY year;

'''

df = pd.read_sql_query(query,engine)
df
```

```
[4]:   year  order_count
0  2015         1953
1  2016         2055
2  2017         2534
3  2018         3258
```

[5]: *# Q2: Total count of distinct customers*

```
query=''

SELECT COUNT(DISTINCT customer_id) AS total_distinct_customers
FROM sales_data;

'''

df = pd.read_sql_query(query,engine)
df
```

```
[5]:   total_distinct_customers
0                        793
```

[6]: *# Q3: Top 3 customers who have ordered the most with their total amount of transactions.*

```
query=''

SELECT customer_id, customer_name, SUM(sales) AS total_amount
FROM sales_data
GROUP BY customer_id, customer_name
ORDER BY total_amount DESC
LIMIT 3;
```

```
'''

df = pd.read_sql_query(query,engine)
df
```

```
[6]:  customer_id  customer_name  total_amount
0     SM-20320    Sean Miller    25043.07
1     TC-20980    Tamara Chand    19052.22
2     RB-19360    Raymond Buch    15117.35
```

```
[7]: # Q4: Customer Transactions per Year (from the beginning year to last year)
```

```
query='''

SELECT EXTRACT(YEAR FROM order_date) AS year, COUNT(*) AS transaction_count
FROM sales_data
GROUP BY year
ORDER BY year;

'''

df = pd.read_sql_query(query,engine)
df
```

```
[7]:  year  transaction_count
0  2015                 1953
1  2016                 2055
2  2017                 2534
3  2018                 3258
```

```
[8]: # Q5: Most selling items sub-category names
```

```
query='''

SELECT sub_category, SUM(sales) AS total_sales
FROM sales_data
GROUP BY sub_category
ORDER BY total_sales DESC
LIMIT 5;

'''

df = pd.read_sql_query(query,engine)
df
```

```
[8]:  sub_category  total_sales
0      Phones    327782.49
```

1	Chairs	322822.75
2	Storage	219343.37
3	Tables	202810.77
4	Binders	200028.82

[9]: *# Q6: Region basis sales performance PIE CHART*

```
query=''

SELECT region, SUM(sales) AS total_sales
FROM sales_data
GROUP BY region
ORDER BY total_sales DESC;

'''

df = pd.read_sql_query(query,engine)
df
```

[9]:

	region	total_sales
0	West	710219.77
1	East	669518.85
2	Central	492646.90
3	South	389151.45

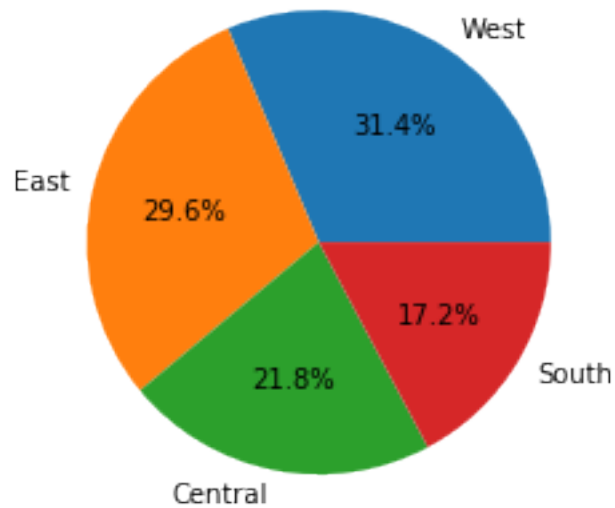
[10]: `import matplotlib.pyplot as plt`

```
# Replace the data below with the actual result of your SQL query
data = [('West', 710219.77),
        ('East', 669518.85),
        ('Central', 492646.90),
        ('South', 389151.45)]

regions = [row[0] for row in data]
sales = [row[1] for row in data]

plt.pie(sales, labels=regions, autopct='%1.1f%%')
plt.title('Total Sales by Region')
plt.show()
```

Total Sales by Region



```
[11]: # Q7: Sales performance LINE CHART over the years
```

```
query='''
```

```
SELECT EXTRACT(YEAR FROM order_date) AS year, SUM(sales) AS total_sales
FROM sales_data
GROUP BY year
ORDER BY year;
```

```
'''
```

```
df = pd.read_sql_query(query,engine)
df
```

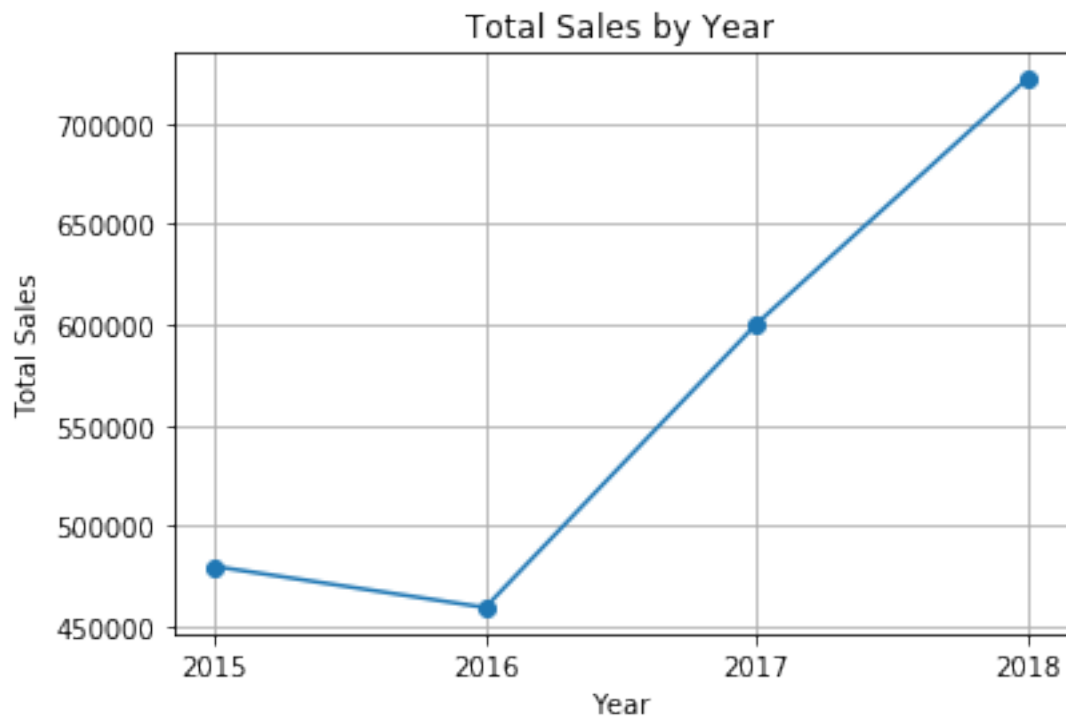
```
[11]:
```

	year	total_sales
0	2015	479856.27
1	2016	459435.94
2	2017	600192.80
3	2018	722051.96

```
[12]: import matplotlib.pyplot as plt
```

```
# Replace the data below with the actual result of your SQL query
data = [(2015, 479856.27),
```

```
(2016, 459435.94),  
(2017, 600192.80),  
(2018, 722051.96)]  
  
years = [row[0] for row in data]  
sales = [row[1] for row in data]  
  
plt.plot(years, sales, marker='o')  
plt.xlabel('Year')  
plt.ylabel('Total Sales')  
plt.title('Total Sales by Year')  
plt.xticks(years)  
plt.grid(True)  
plt.show()
```



[ ]: