E-commerce Order Analysis Dashboard

Project Overview

An e-commerce business wants to analyze order trends, customer preferences, and revenue growth.

- Extract & manipulate data using SQL.
- Process & analyze data using Pandas in Python.
- Visualize insights using Power BI.

SQL Task – Data Extraction & Manipulation

- Retrieve total sales per region.
- Find the top 5 best-selling products.
- Calculate monthly revenue.
- Identify repeat customers.
- Find average order value per region.
- Determine peak sales hour in a day.
- Rank products by sales within each category.

Pandas Task – Data Processing & Analysis

- Load the dataset into Data Frames.
- Handle missing values & data cleaning.
- Analyze total sales per customer.
- Calculate moving average sales per month.
- Segment customers based on total spending (e.g., low, medium, high).
- Calculate product return rate (if return data available).
- Identify top 10 customers by lifetime value.

Power BI Task – Data Visualization

- A sales dashboard with revenue trends.
- A product performance analysis chart.
- A customer segmentation analysis based on age/gender.
- Heatmap of sales by region and time.
- Pie chart of sales contribution by product category.
- Bar chart of sales performance by sales rep (if data available).
- KPI cards for key metrics: total revenue, total orders, repeat rate, average basket size.

Project Tasks

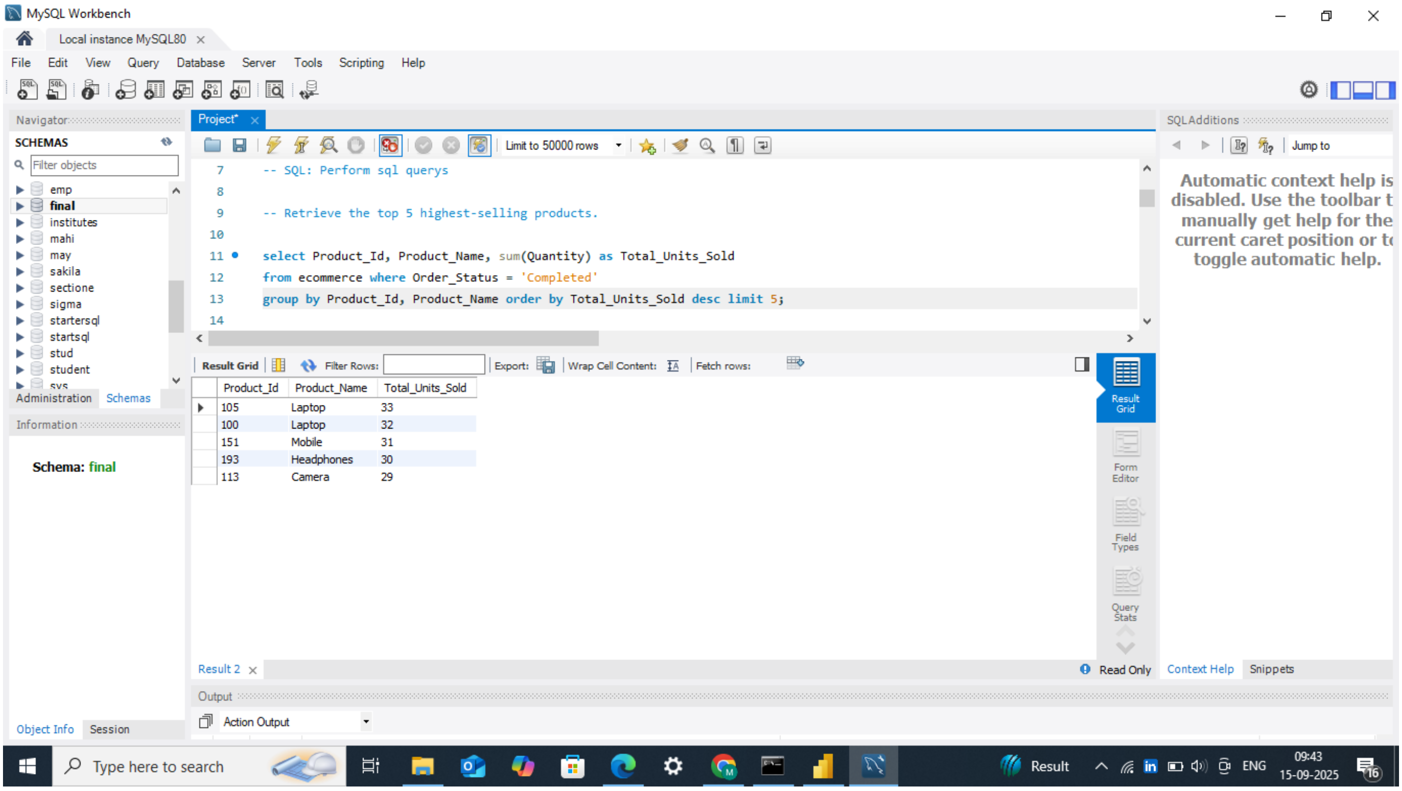
SQL:

1. **Retrieve the top 5 highest-selling products.**

Query: select Product_Id, Product_Name, sum (Quantity) as Total_Units_Sold

from ecommerce where Order_Status = 'Completed'

group by Product_Id, Product_Name order by Total_Units_Sold desc limit 5;



```
-- SQL: Perform sql querys

-- Retrieve the top 5 highest-selling products.

select Product_Id, Product_Name, sum(Quantity) as Total_Units_Sold
from ecommerce where Order_Status = 'Completed'
group by Product_Id, Product_Name order by Total_Units_Sold desc limit 5;
```

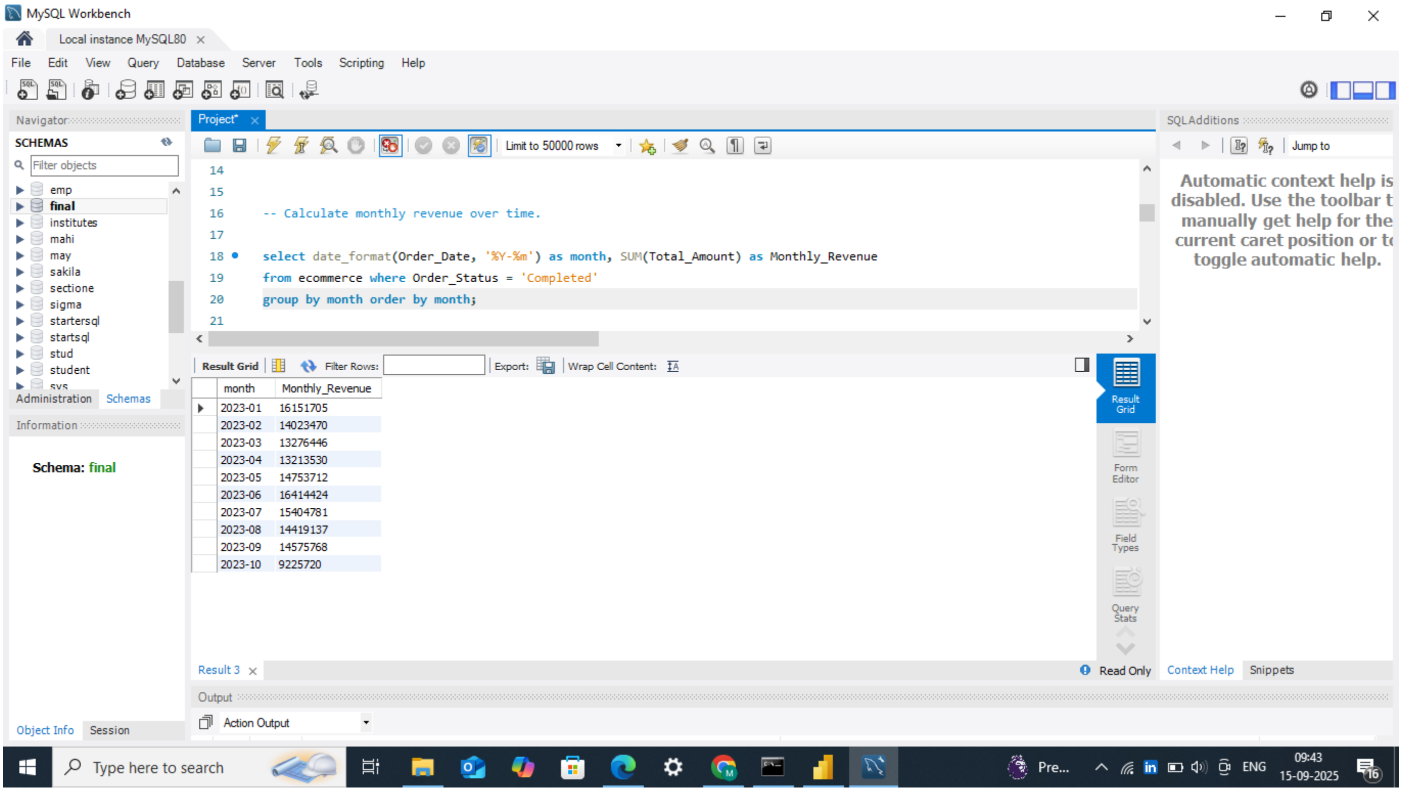
Product_Id	Product_Name	Total_Units_Sold
105	Laptop	33
100	Laptop	32
151	Mobile	31
193	Headphones	30
113	Camera	29

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

2. Calculate monthly revenue over time.

Query: select date_format (Order_Date, '%Y-%m') as month, sum (Total_Amount) as Monthly_Revenue

from ecommerce where Order_Status = 'Completed' group by month order by month;



```
-- Calculate monthly revenue over time.

select date_format(Order_Date, '%Y-%m') as month, SUM(Total_Amount) as Monthly_Revenue
from ecommerce where Order_Status = 'Completed'
group by month order by month;
```

month	Monthly_Revenue
2023-01	16151705
2023-02	14023470
2023-03	13276446
2023-04	13213530
2023-05	14753712
2023-06	16414424
2023-07	15404781
2023-08	14419137
2023-09	14575768
2023-10	9225720

Schema: final

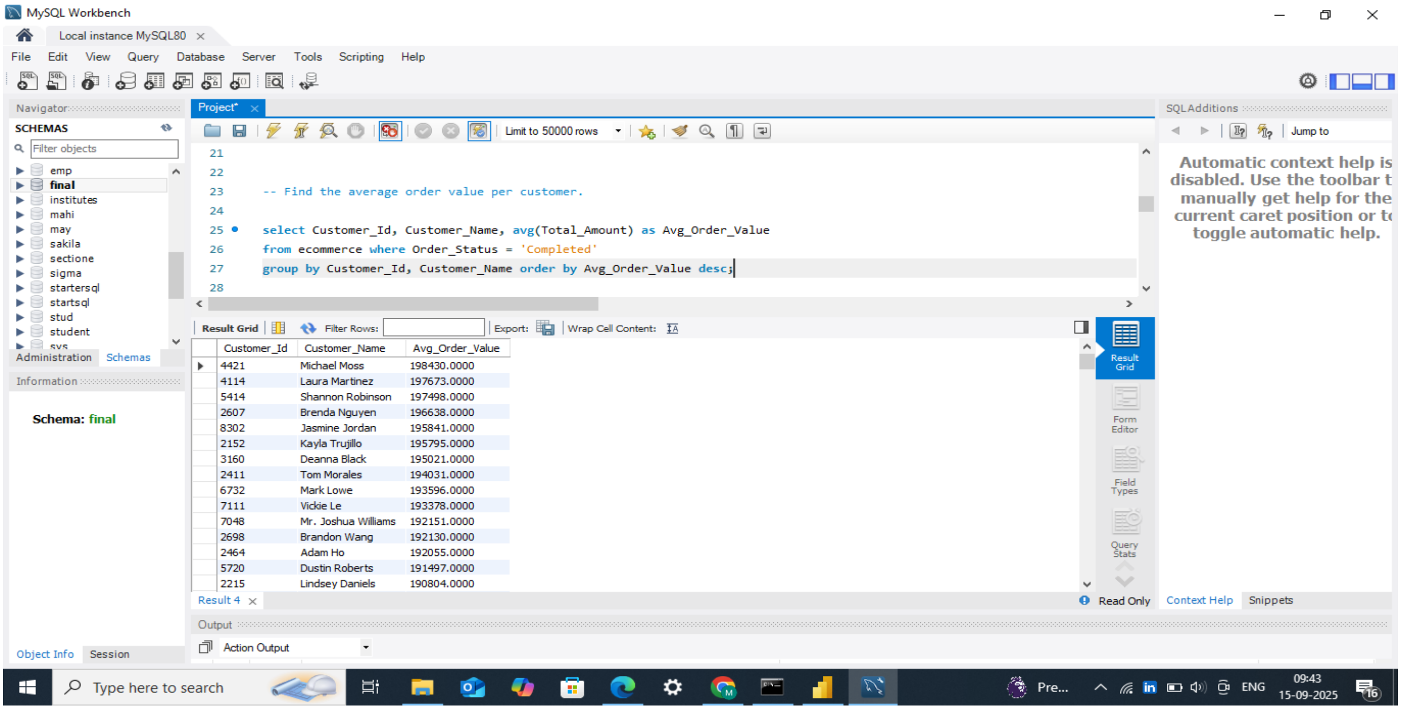
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

3. Find the average order value per customer.

Query: select Customer_Id, Customer_Name, avg (Total_Amount) as Avg_Order_Value

from ecommerce where Order_Status = 'Completed'

group by Customer_Id, Customer_Name order by Avg_Order_Value desc;



```
-- Find the average order value per customer.

select Customer_Id, Customer_Name, avg(Total_Amount) as Avg_Order_Value
from ecommerce where Order_Status = 'Completed'
group by Customer_Id, Customer_Name order by Avg_Order_Value desc;
```

Customer_Id	Customer_Name	Avg_Order_Value
4421	Michael Moss	198430.0000
4114	Laura Martinez	197673.0000
5414	Shannon Robinson	197498.0000
2607	Brenda Nguyen	196638.0000
8302	Jasmine Jordan	195841.0000
2152	Kayla Trujillo	195795.0000
3160	Deanna Black	195021.0000
2411	Tom Morales	194031.0000
6732	Mark Lowe	193596.0000
7111	Vickie Le	193378.0000
7048	Mr. Joshua Williams	192151.0000
2698	Brandon Wang	192130.0000
2464	Adam Ho	192055.0000
5720	Dustin Roberts	191497.0000
2215	Lindsey Daniels	190804.0000

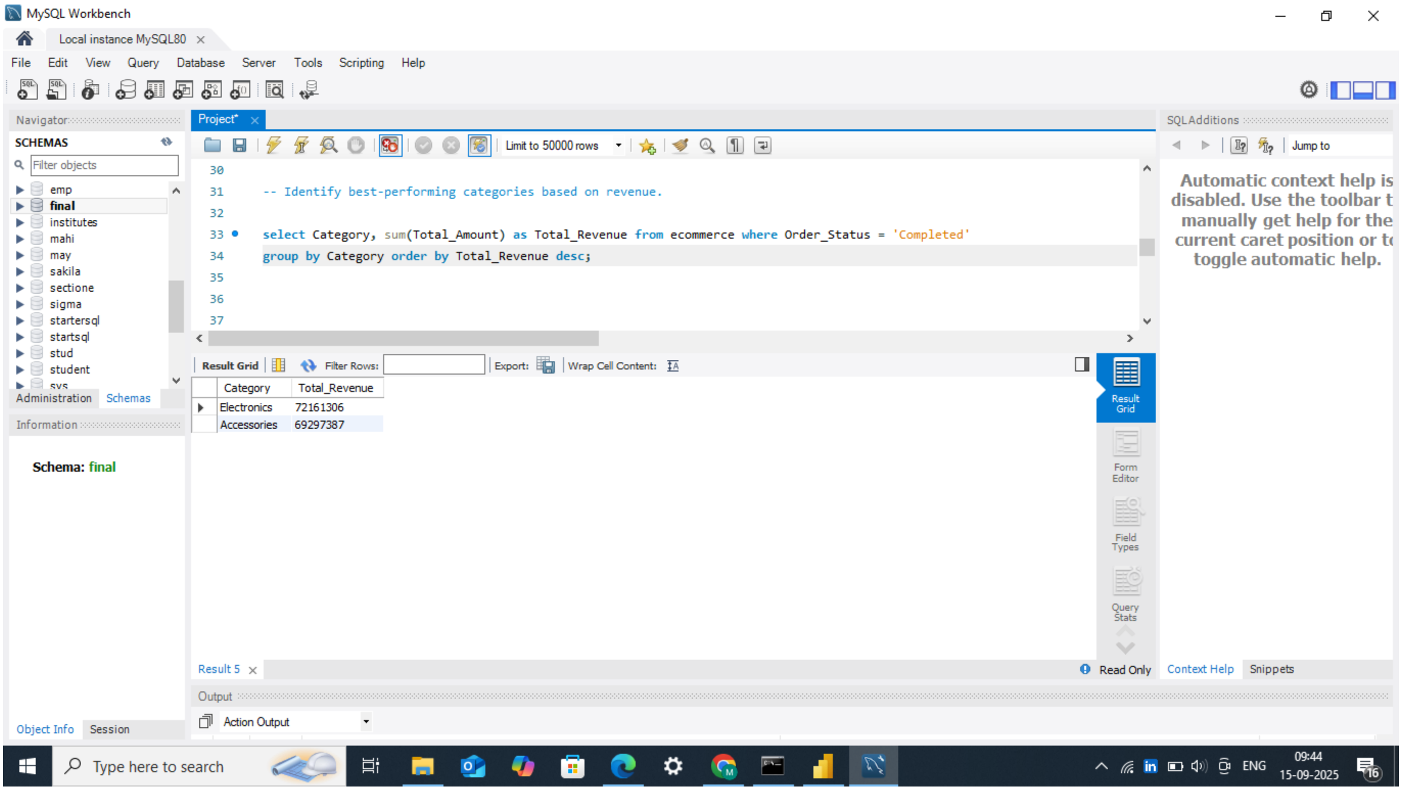
Schema: final

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

4. Identify best-performing categories based on revenue.

Query: select Category, sum (Total_Amount) as Total_Revenue from ecommerce

where Order_Status = 'Completed' group by Category order by Total_Revenue desc;



```sql
-- Identify best-performing categories based on revenue.

select Category, sum(Total_Amount) as Total_Revenue from ecommerce where Order_Status = 'Completed'
group by Category order by Total_Revenue desc;
```

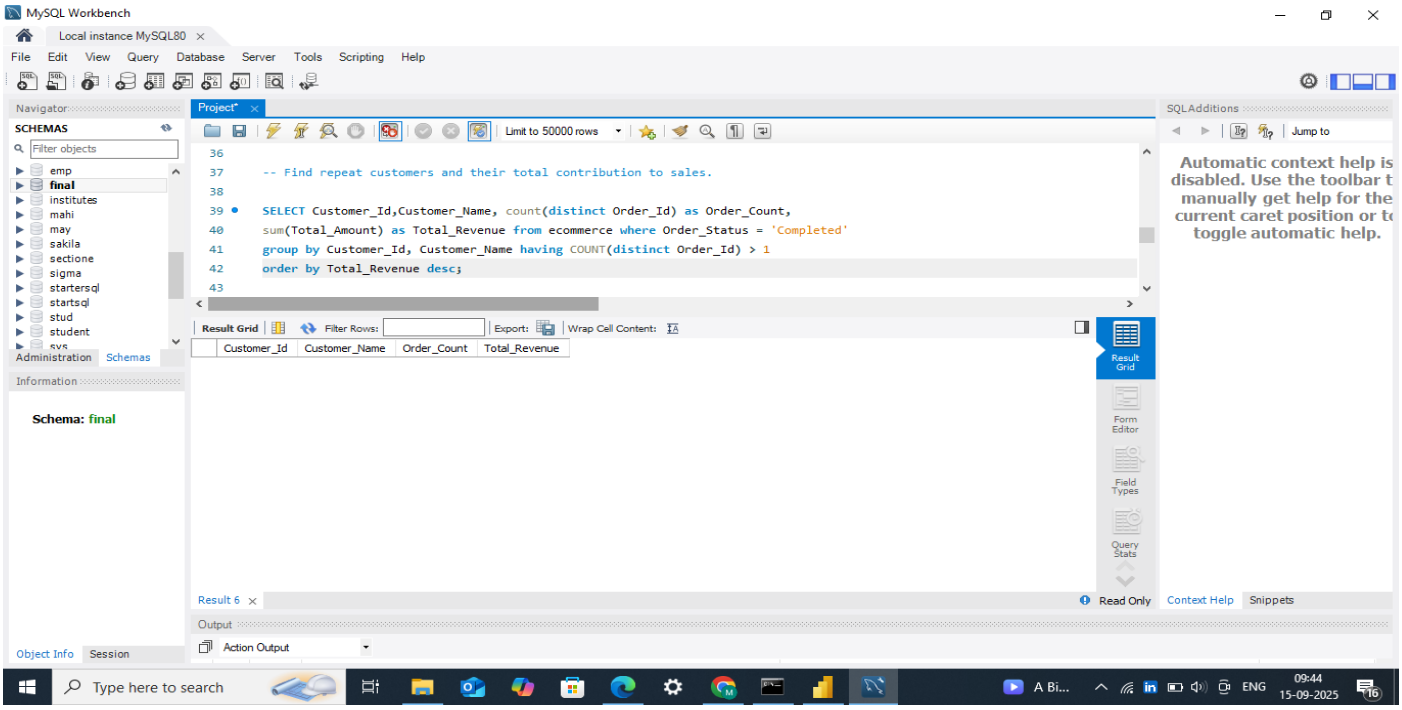
Category	Total_Revenue
Electronics	72161306
Accessories	69297387

5. Find repeat customers and their total contribution to sales.

Query: select customer_id,customer_name,count (order_id) as order_count,sum(total_amount) as total_revenue

from ecommerce where lower(order_status) = 'completed' group by customer_id, customer_name

having count(order_id) > 1 order by total_revenue desc;

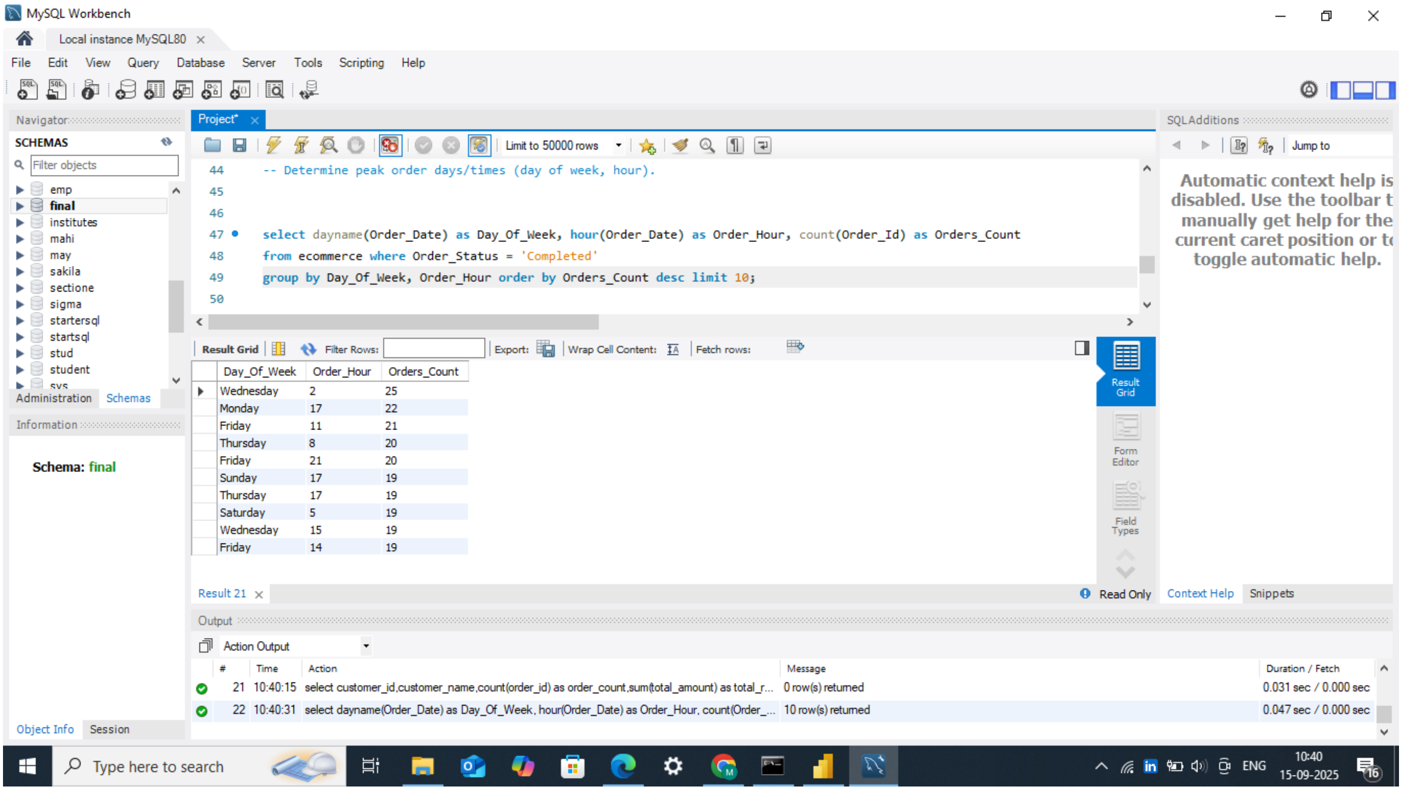


```sql
-- Find repeat customers and their total contribution to sales.

SELECT Customer_Id,Customer_Name, count(distinct Order_Id) as Order_Count,
sum(Total_Amount) as Total_Revenue from ecommerce where Order_Status = 'Completed'
group by Customer_Id, Customer_Name having COUNT(distinct Order_Id) > 1
order by Total_Revenue desc;
```

Customer_Id	Customer_Name	Order_Count	Total_Revenue

6. Determine peak order days/times (day of week, hour).

Query: select dayname (Order_Date) as Day_Of_Week, hour (Order_Date) as Order_Hour, count (Order_Id) as Orders_Count ecommerce where Order_Status = 'Completed' group by Day_Of_Week, Order_Hour order by Orders_Count desc limit 10;



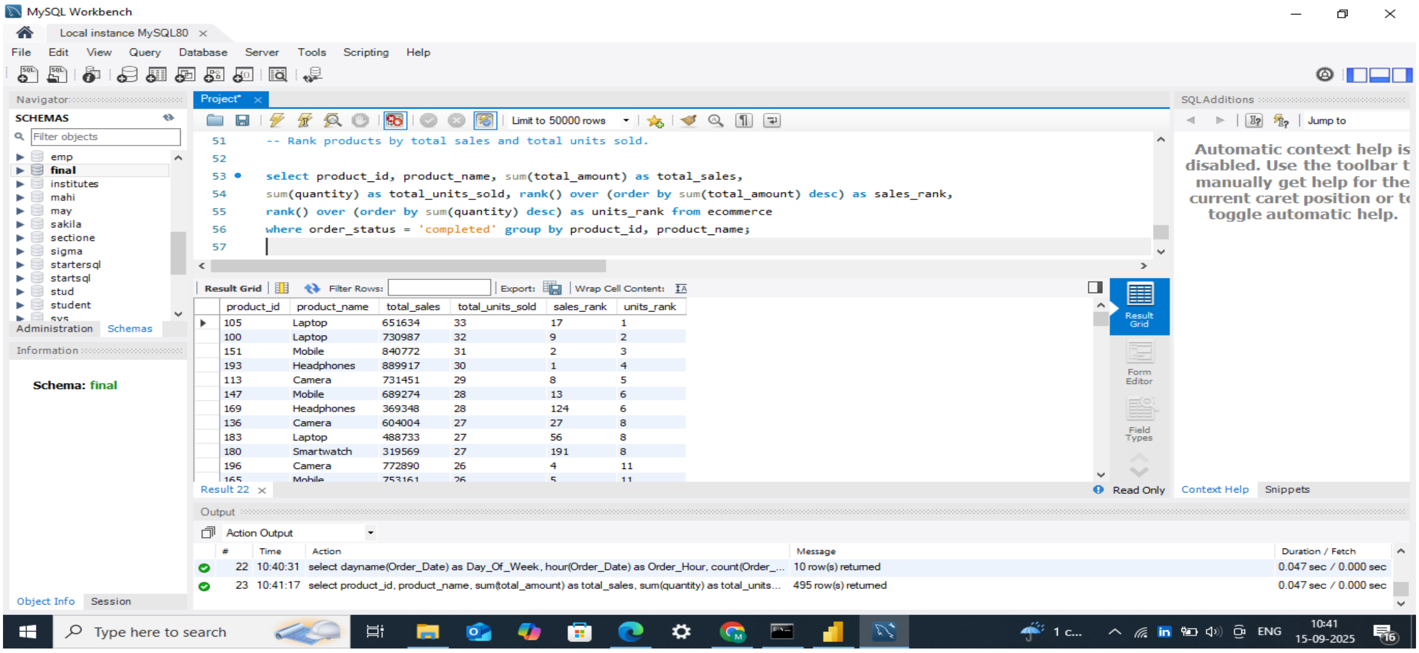
```
-- Determine peak order days/times (day of week, hour).


select dayname(Order_Date) as Day_Of_Week, hour(Order_Date) as Order_Hour, count(Order_Id) as Orders_Count
from ecommerce where Order_Status = 'Completed'
group by Day_Of_Week, Order_Hour order by Orders_Count desc limit 10;

```

Day_Of_Week	Order_Hour	Orders_Count
Wednesday	2	25
Monday	17	22
Friday	11	21
Thursday	8	20
Friday	21	20
Sunday	17	19
Thursday	17	19
Saturday	5	19
Wednesday	15	19
Friday	14	19

```
   #   Time      Action                                                                              Message             Duration / Fetch
✓  21  10:40:15  select customer_id,customer_name,count(order_id) as order_count,sum(total_amount) as total_r...  0 row(s) returned   0.031 sec / 0.000 sec
✓  22  10:40:31  select dayname(Order_Date) as Day_Of_Week, hour(Order_Date) as Order_Hour, count(Order_...  10 row(s) returned  0.047 sec / 0.000 sec
```

7. Rank products by total sales and total units sold.

Query: select product_id, product_name, sum(total_amount) as total_sales, sum(quantity) as total_units_sold, rank() over (order by sum(total_amount) desc) as sales_rank, rank() over (order by sum(quantity) desc) as units_rank from ecommerce where order_status = 'completed' group by product_id, product_name;



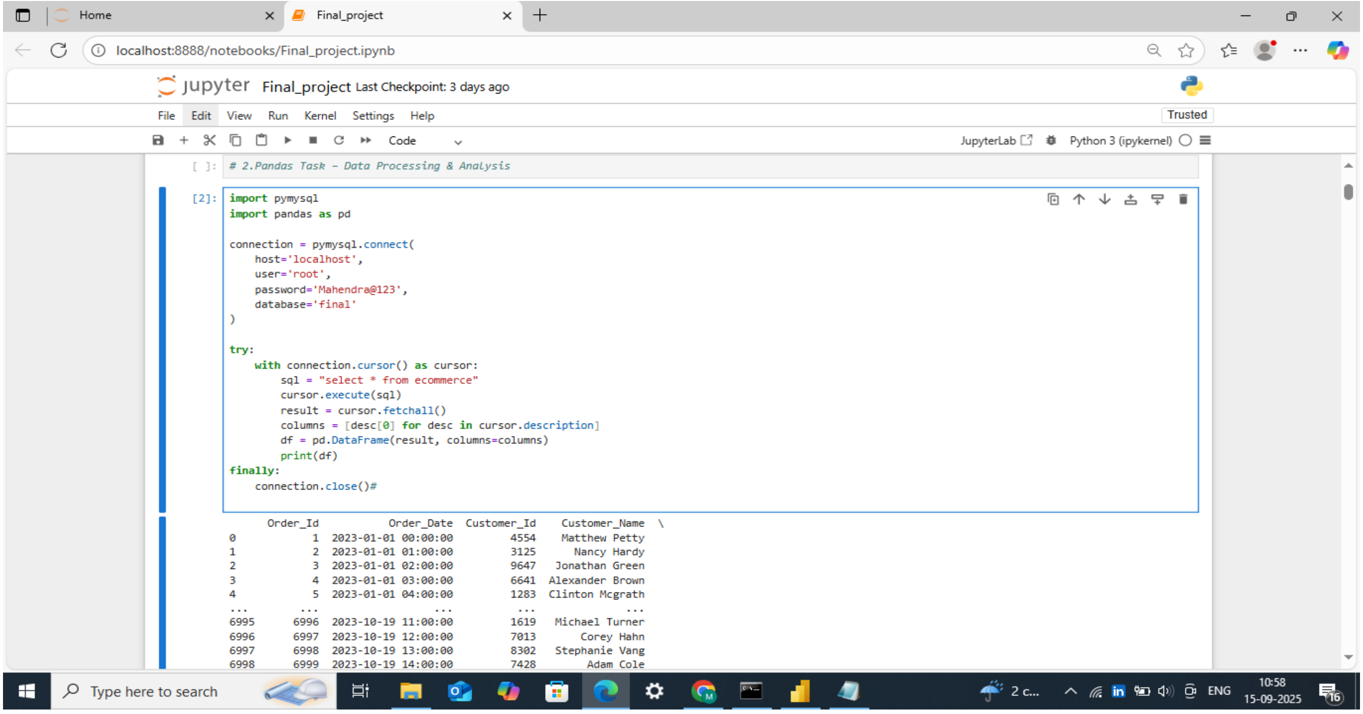
```
-- Rank products by total sales and total units sold.

select product_id, product_name, sum(total_amount) as total_sales,
sum(quantity) as total_units_sold, rank() over (order by sum(total_amount) desc) as sales_rank,
rank() over (order by sum(quantity) desc) as units_rank from ecommerce
where order_status = 'completed' group by product_id, product_name;

```

product_id	product_name	total_sales	total_units_sold	sales_rank	units_rank
105	Laptop	651634	33	17	1
100	Laptop	730987	32	9	2
151	Mobile	840772	31	2	3
193	Headphones	889917	30	1	4
113	Camera	731451	29	8	5
147	Mobile	689274	28	13	6
169	Headphones	369348	28	124	6
136	Camera	604004	27	27	8
183	Laptop	488733	27	56	8
180	Smartwatch	319569	27	191	8
196	Camera	772890	26	4	11
165	Mobile	753161	26	5	11

```
   #   Time      Action                                                                              Message              Duration / Fetch
✓  22  10:40:31  select dayname(Order_Date) as Day_Of_Week, hour(Order_Date) as Order_Hour, count(Order_...  10 row(s) returned   0.047 sec / 0.000 sec
✓  23  10:41:17  select product_id, product_name, sum(total_amount) as total_sales, sum(quantity) as total_units...  495 row(s) returned  0.047 sec / 0.000 sec
```

Pandas:

- **Data Processing & Analysis , Process & analyze data using Pandas in Python.**
- **Connection:**



```
# 2.Pandas Task - Data Processing & Analysis
```

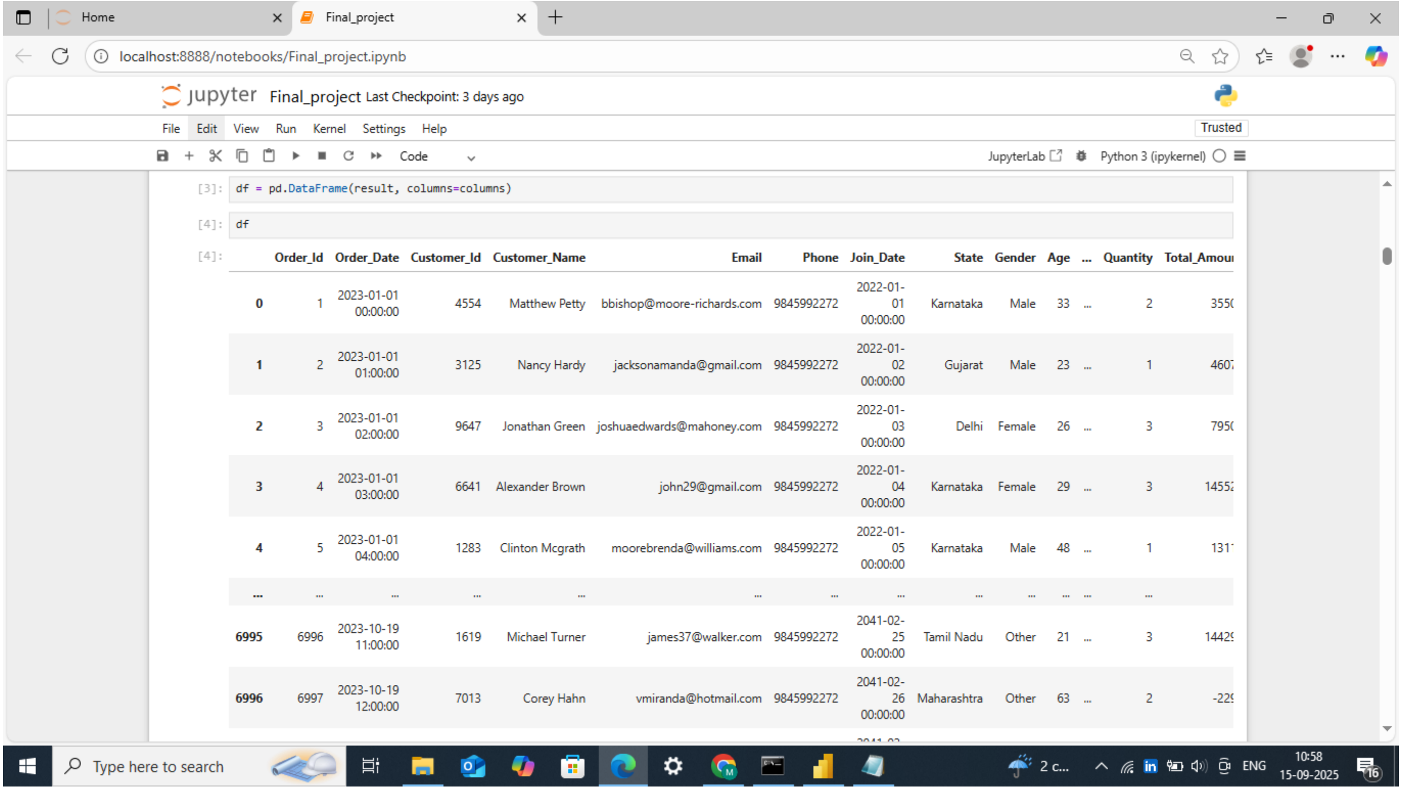
```
import pymysql
import pandas as pd

connection = pymysql.connect(
    host='localhost',
    user='root',
    password='Mahendra@123',
    database='final'
)

try:
    with connection.cursor() as cursor:
        sql = "select * from ecommerce"
        cursor.execute(sql)
        result = cursor.fetchall()
        columns = [desc[0] for desc in cursor.description]
        df = pd.DataFrame(result, columns=columns)
        print(df)
finally:
    connection.close()#
```

```
      Order_Id          Order_Date  Customer_Id    Customer_Name  \
0            1 2023-01-01 00:00:00         4554    Matthew Petty
1            2 2023-01-01 01:00:00         3125      Nancy Hardy
2            3 2023-01-01 02:00:00         9647   Jonathan Green
3            4 2023-01-01 03:00:00         6641  Alexander Brown
4            5 2023-01-01 04:00:00         1283  Clinton Mcgrath
...        ...                 ...          ...              ...
6995      6996 2023-10-19 11:00:00         1619   Michael Turner
6996      6997 2023-10-19 12:00:00         7013       Corey Hahn
6997      6998 2023-10-19 13:00:00         8302   Stephanie Vang
6998      6999 2023-10-19 14:00:00         7428        Adam Cole
```

Convert data into DataFrame:

df = pd.DataFrame(result, columns=columns)

df



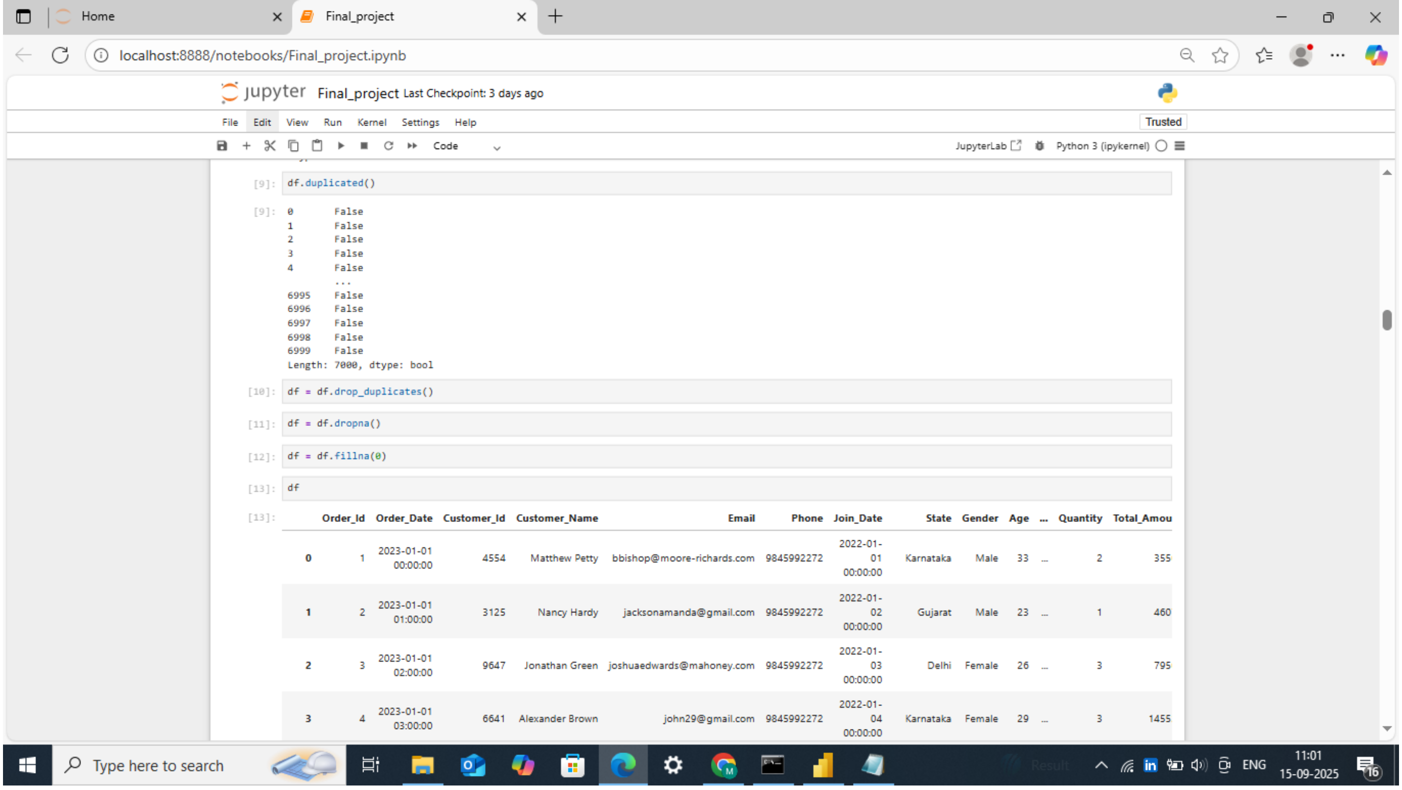
```
[3]: df = pd.DataFrame(result, columns=columns)
```

```
[4]: df
```

	Order_Id	Order_Date	Customer_Id	Customer_Name	Email	Phone	Join_Date	State	Gender	Age	...	Quantity	Total_Amou
0	1	2023-01-01 00:00:00	4554	Matthew Petty	bbishop@moore-richards.com	9845992272	2022-01-01 00:00:00	Karnataka	Male	33	...	2	355
1	2	2023-01-01 01:00:00	3125	Nancy Hardy	jacksonamanda@gmail.com	9845992272	2022-01-02 00:00:00	Gujarat	Male	23	...	1	460
2	3	2023-01-01 02:00:00	9647	Jonathan Green	joshuaedwards@mahoney.com	9845992272	2022-01-03 00:00:00	Delhi	Female	26	...	3	795
3	4	2023-01-01 03:00:00	6641	Alexander Brown	john29@gmail.com	9845992272	2022-01-04 00:00:00	Karnataka	Female	29	...	3	1455
4	5	2023-01-01 04:00:00	1283	Clinton Mcgrath	moorebrenda@williams.com	9845992272	2022-01-05 00:00:00	Karnataka	Male	48	...	1	131
...	...	...	...	...	...	...	...	...	...	...	...	...	
6995	6996	2023-10-19 11:00:00	1619	Michael Turner	james37@walker.com	9845992272	2041-02-25 00:00:00	Tamil Nadu	Other	21	...	3	1442
6996	6997	2023-10-19 12:00:00	7013	Corey Hahn	vmiranda@hotmail.com	9845992272	2041-02-26 00:00:00	Maharashtra	Other	63	...	2	-22

1. Load and clean the dataset (handle duplicates, missing values, etc.).

```
df.duplicated()

df = df.drop_duplicates()

df = df.dropna()

df = df.fillna(0)

df
```



Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
ENG
11:01
15-09-2025

```
[9]: df.duplicated()
```

```
[9]: 0       False
     1       False
     2       False
     3       False
     4       False
             ...
     6995    False
     6996    False
     6997    False
     6998    False
     6999    False
     Length: 7000, dtype: bool
```

```
[10]: df = df.drop_duplicates()
```

```
[11]: df = df.dropna()
```

```
[12]: df = df.fillna(0)
```

```
[13]: df
```

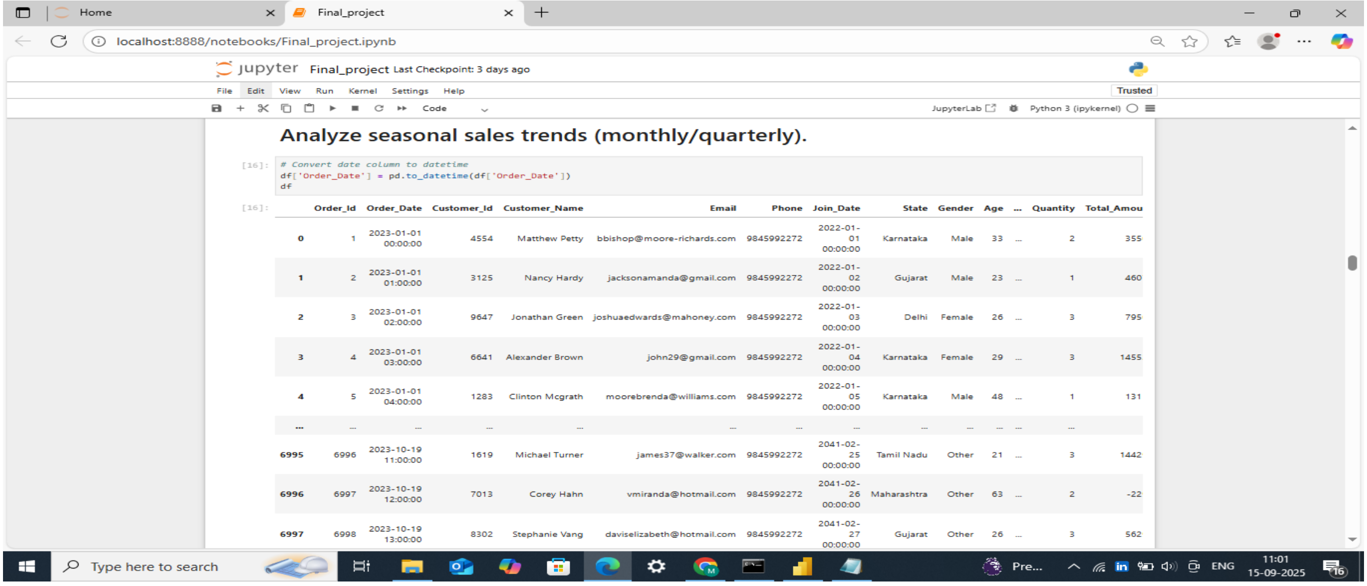
[13]:

	Order_Id	Order_Date	Customer_Id	Customer_Name	Email	Phone	Join_Date	State	Gender	Age	...	Quantity	Total_Amou
0	1	2023-01-01 00:00:00	4554	Matthew Petty	bbishop@moore-richards.com	9845992272	2022-01-01 00:00:00	Karnataka	Male	33	...	2	355
1	2	2023-01-01 01:00:00	3125	Nancy Hardy	jacksonamanda@gmail.com	9845992272	2022-01-02 00:00:00	Gujarat	Male	23	...	1	460
2	3	2023-01-01 02:00:00	9647	Jonathan Green	joshuaedwards@mahoney.com	9845992272	2022-01-03 00:00:00	Delhi	Female	26	...	3	795
3	4	2023-01-01 03:00:00	6641	Alexander Brown	john29@gmail.com	9845992272	2022-01-04 00:00:00	Karnataka	Female	29	...	3	1455

2. Analyze seasonal sales trends (monthly/quarterly).

- **Convert date column to datetime**

```
df['Order_Date'] = pd.to_datetime(df['Order_Date'])
df
```



Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
Pre...
ENG
11:01
15-09-2025

Analyze seasonal sales trends (monthly/quarterly).

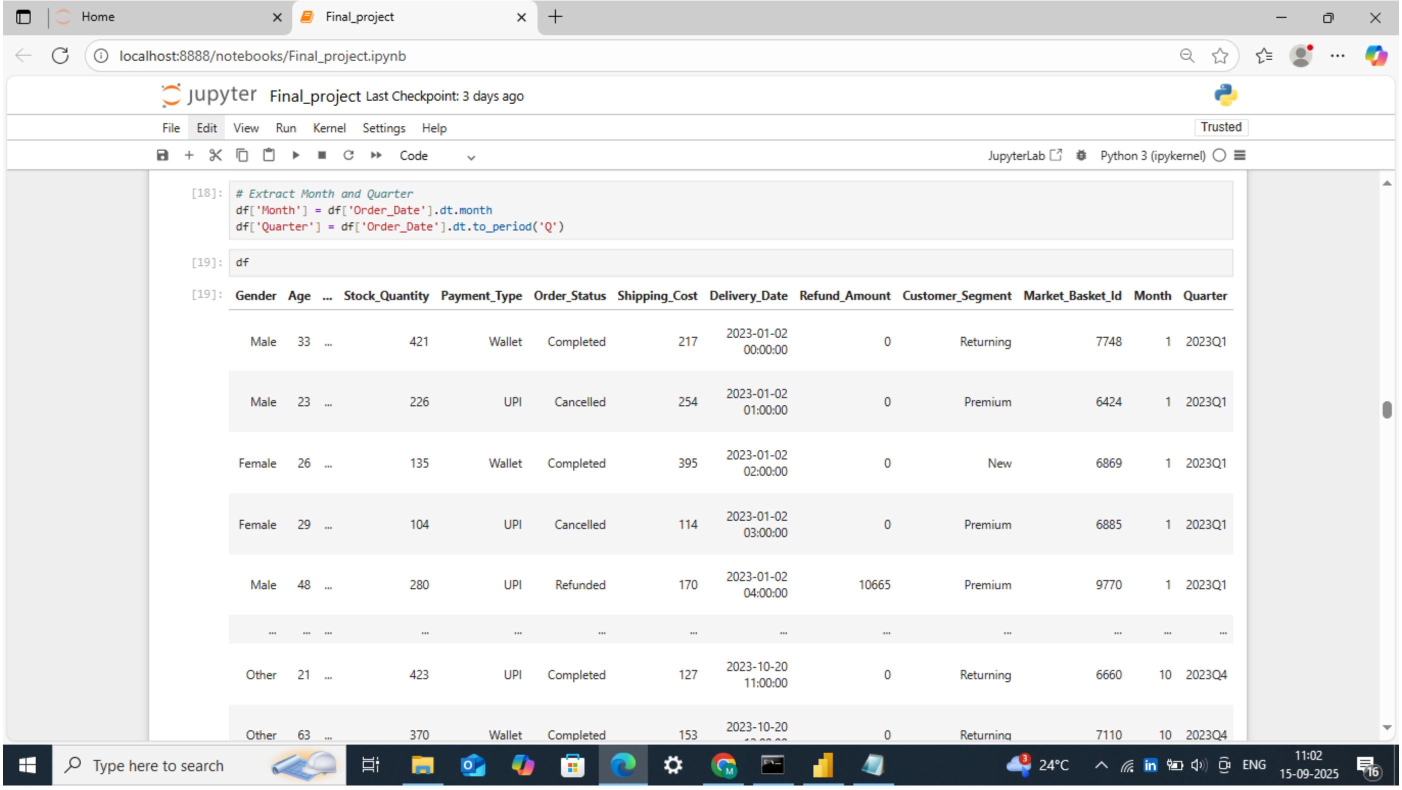
```
[16]: # Convert date column to datetime
      df['Order_Date'] = pd.to_datetime(df['Order_Date'])
      df
```

[16]:

	Order_Id	Order_Date	Customer_Id	Customer_Name	Email	Phone	Join_Date	State	Gender	Age	...	Quantity	Total_Amou
0	1	2023-01-01 00:00:00	4554	Matthew Petty	bbishop@moore-richards.com	9845992272	2022-01-01 00:00:00	Karnataka	Male	33	...	2	355
1	2	2023-01-01 01:00:00	3125	Nancy Hardy	jacksonamanda@gmail.com	9845992272	2022-01-02 00:00:00	Gujarat	Male	23	...	1	460
2	3	2023-01-01 02:00:00	9647	Jonathan Green	joshuaedwards@mahoney.com	9845992272	2022-01-03 00:00:00	Delhi	Female	26	...	3	795
3	4	2023-01-01 03:00:00	6641	Alexander Brown	john29@gmail.com	9845992272	2022-01-04 00:00:00	Karnataka	Female	29	...	3	1455
4	5	2023-01-01 04:00:00	1283	Clinton Mcgrath	moorebrenda@williams.com	9845992272	2022-01-05 00:00:00	Karnataka	Male	48	...	1	131
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6995	6996	2023-10-19 11:00:00	1619	Michael Turner	james37@walker.com	9845992272	2041-02-25 00:00:00	Tamil Nadu	Other	21	...	3	1442
6996	6997	2023-10-19 12:00:00	7013	Corey Hahn	vmiranda@hotmail.com	9845992272	2041-02-26 00:00:00	Maharashtra	Other	63	...	2	-22
6997	6998	2023-10-19 13:00:00	8302	Stephanie Vang	daviselizabeth@hotmail.com	9845992272	2041-02-27 00:00:00	Gujarat	Other	26	...	3	562

- **Extract Month and Quarter**

df['Month'] = df['Order_Date'].dt.month
df['Quarter'] = df['Order_Date'].dt.to_period('Q')



Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
24°C
ENG
11:02
15-09-2025

```
[18]: # Extract Month and Quarter
      df['Month'] = df['Order_Date'].dt.month
      df['Quarter'] = df['Order_Date'].dt.to_period('Q')

[19]: df
```

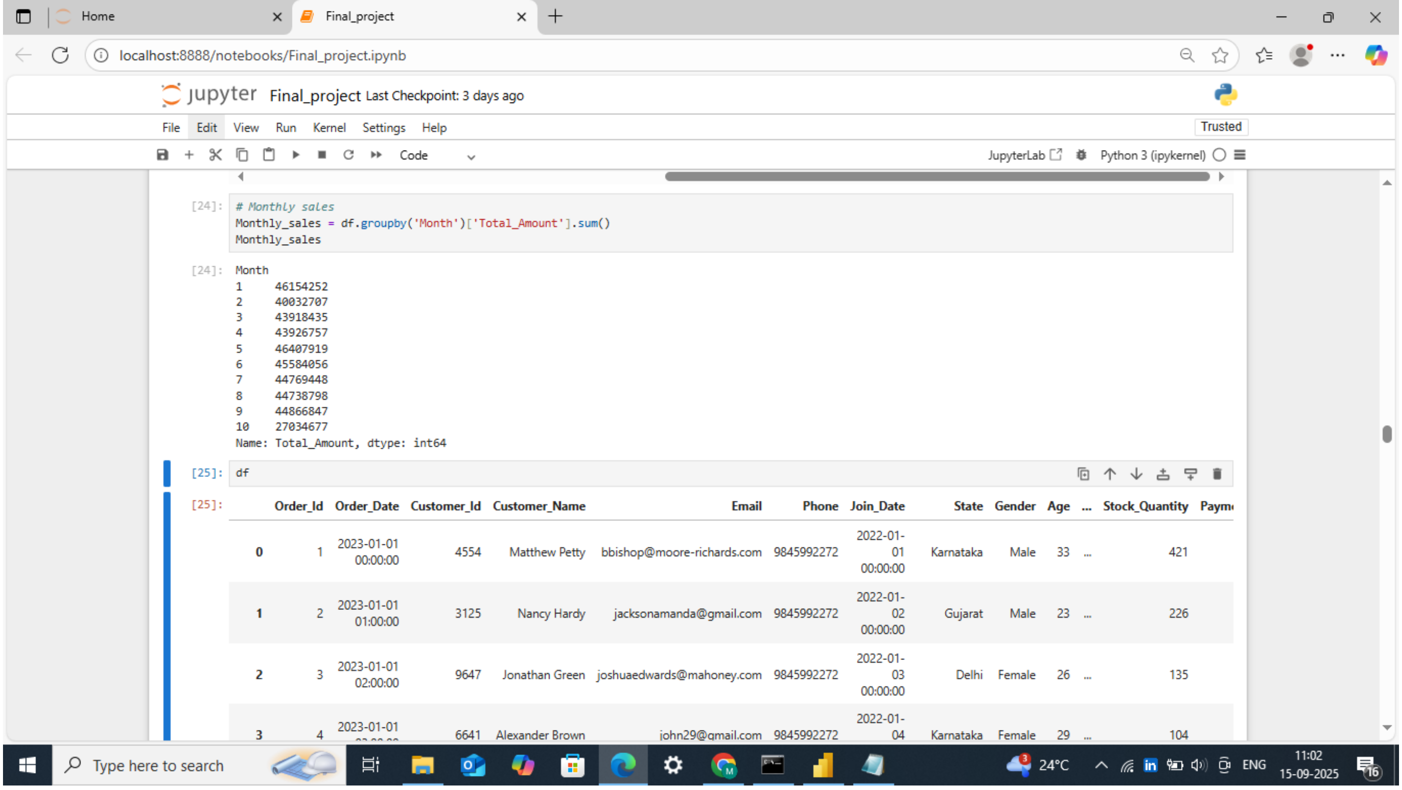
[19]:

Gender	Age	...	Stock_Quantity	Payment_Type	Order_Status	Shipping_Cost	Delivery_Date	Refund_Amount	Customer_Segment	Market_Basket_Id	Month	Quarter
Male	33	...	421	Wallet	Completed	217	2023-01-02 00:00:00	0	Returning	7748	1	2023Q1
Male	23	...	226	UPI	Cancelled	254	2023-01-02 01:00:00	0	Premium	6424	1	2023Q1
Female	26	...	135	Wallet	Completed	395	2023-01-02 02:00:00	0	New	6869	1	2023Q1
Female	29	...	104	UPI	Cancelled	114	2023-01-02 03:00:00	0	Premium	6885	1	2023Q1
Male	48	...	280	UPI	Refunded	170	2023-01-02 04:00:00	10665	Premium	9770	1	2023Q1
...	...	...	...	...	...	...	...	...	...	...	...	...
Other	21	...	423	UPI	Completed	127	2023-10-20 11:00:00	0	Returning	6660	10	2023Q4
Other	63	...	370	Wallet	Completed	153	2023-10-20	0	Returning	7110	10	2023Q4

- **Monthly sales**

Monthly_sales = df.groupby('Month')['Total_Amount'].sum()

Monthly_sales



Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
24°C
ENG
11:02
15-09-2025

```
[24]: # Monthly sales
      Monthly_sales = df.groupby('Month')['Total_Amount'].sum()
      Monthly_sales

[24]: Month
      1     46154252
      2     40032707
      3     43918435
      4     43926757
      5     46407919
      6     45584056
      7     44769448
      8     44738798
      9     44866847
      10    27034677
      Name: Total_Amount, dtype: int64

[25]: df
```

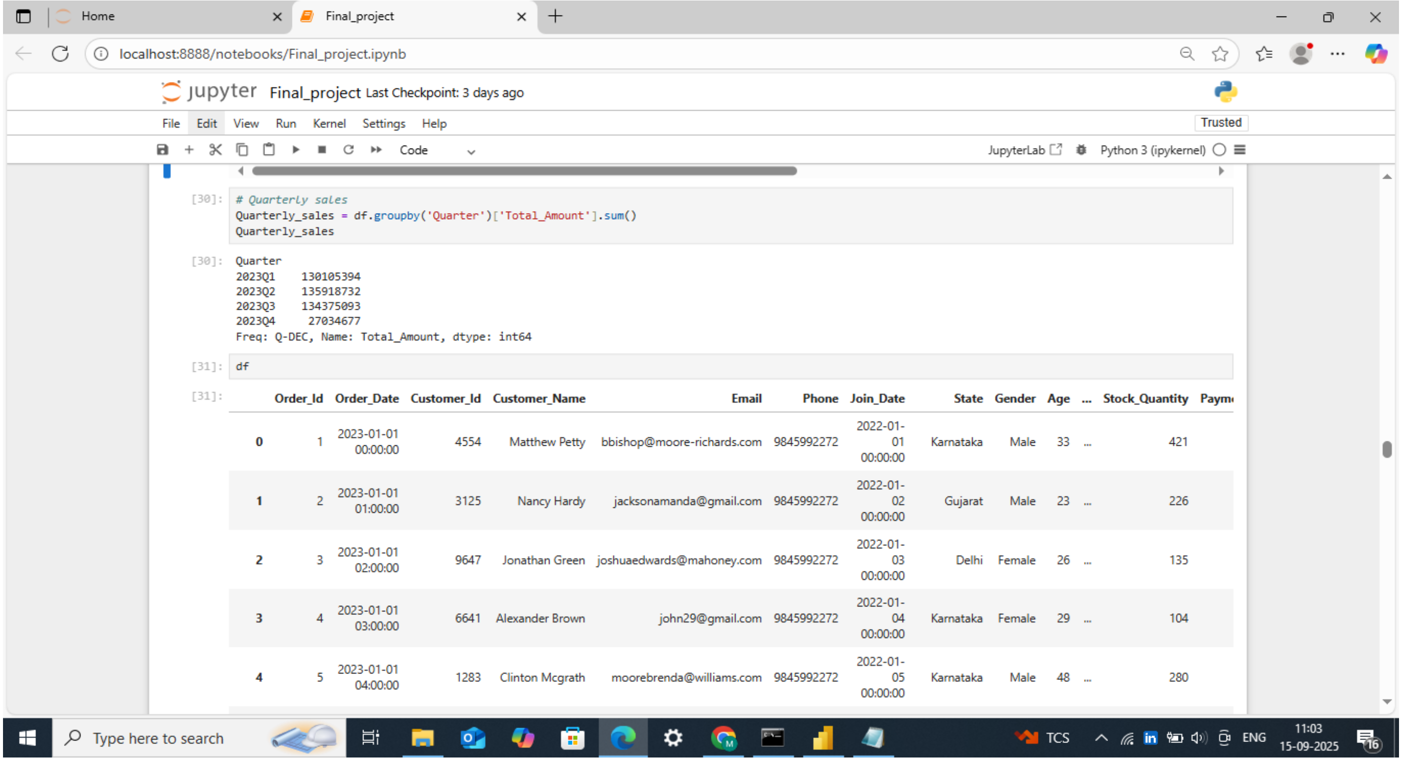
[25]:

	Order_Id	Order_Date	Customer_Id	Customer_Name	Email	Phone	Join_Date	State	Gender	Age	...	Stock_Quantity	Paym
0	1	2023-01-01 00:00:00	4554	Matthew Petty	bbishop@moore-richards.com	9845992272	2022-01-01 00:00:00	Karnataka	Male	33	...	421	
1	2	2023-01-01 01:00:00	3125	Nancy Hardy	jacksonamanda@gmail.com	9845992272	2022-01-02 00:00:00	Gujarat	Male	23	...	226	
2	3	2023-01-01 02:00:00	9647	Jonathan Green	joshuaedwards@mahoney.com	9845992272	2022-01-03 00:00:00	Delhi	Female	26	...	135	
3	4	2023-01-01	6641	Alexander Brown	john29@gmail.com	9845992272	2022-01-04	Karnataka	Female	29	...	104	

- **Quarterly sales**

Quarterly_sales = df.groupby('Quarter')['Total_Amount'].sum()

Quarterly_sales



Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
TCS
ENG
11:03
15-09-2025

```
[30]: # Quarterly sales
      Quarterly_sales = df.groupby('Quarter')['Total_Amount'].sum()
      Quarterly_sales
```

```
[30]: Quarter
      2023Q1    130105394
      2023Q2    135918732
      2023Q3    134375093
      2023Q4     27034677
      Freq: Q-DEC, Name: Total_Amount, dtype: int64
```

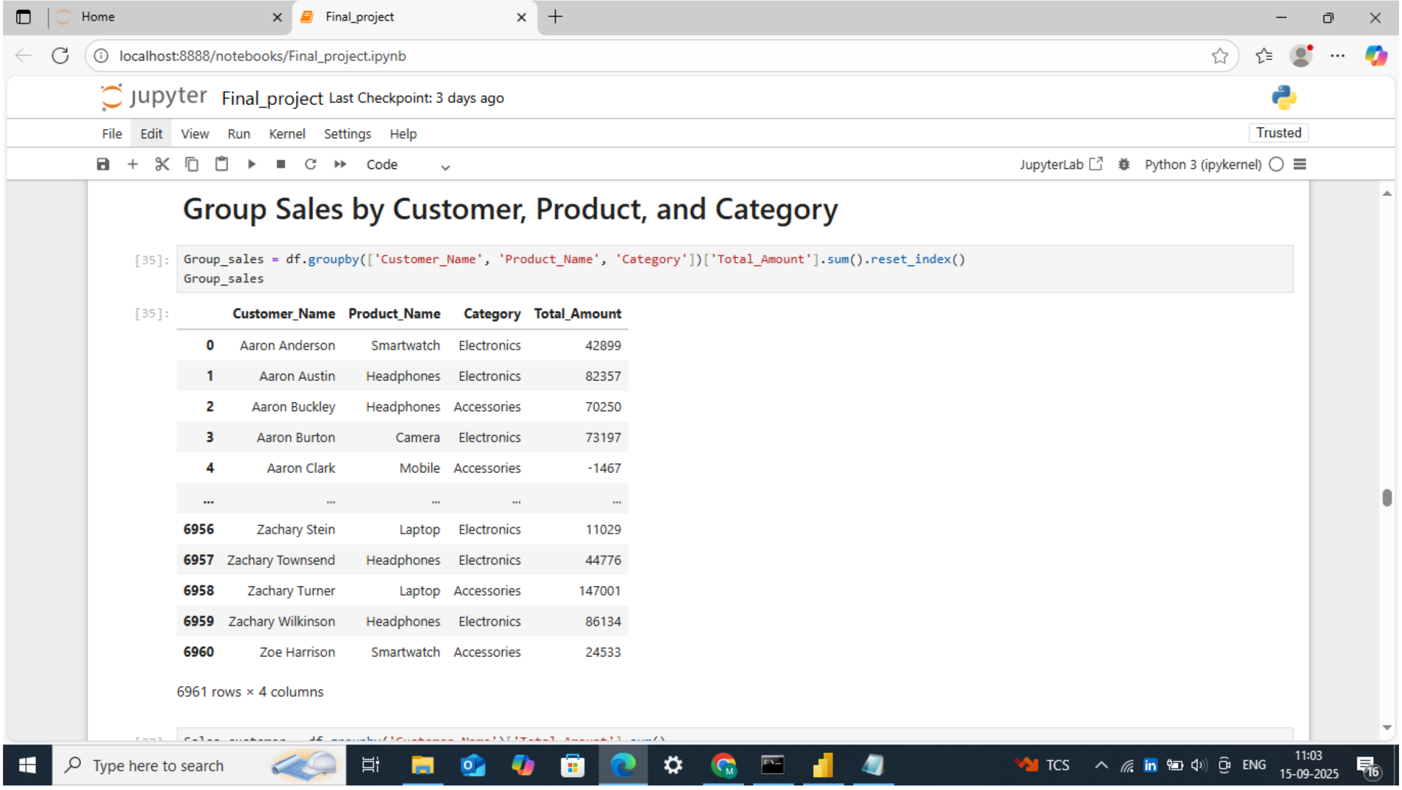
```
[31]: df
```

[31]:

	Order_Id	Order_Date	Customer_Id	Customer_Name	Email	Phone	Join_Date	State	Gender	Age	...	Stock_Quantity	Paym
0	1	2023-01-01 00:00:00	4554	Matthew Petty	bbishop@moore-richards.com	9845992272	2022-01-01 00:00:00	Karnataka	Male	33	...	421	
1	2	2023-01-01 01:00:00	3125	Nancy Hardy	jacksonamanda@gmail.com	9845992272	2022-01-02 00:00:00	Gujarat	Male	23	...	226	
2	3	2023-01-01 02:00:00	9647	Jonathan Green	joshuaedwards@mahoney.com	9845992272	2022-01-03 00:00:00	Delhi	Female	26	...	135	
3	4	2023-01-01 03:00:00	6641	Alexander Brown	john29@gmail.com	9845992272	2022-01-04 00:00:00	Karnataka	Female	29	...	104	
4	5	2023-01-01 04:00:00	1283	Clinton Mcgrath	moorebrenda@williams.com	9845992272	2022-01-05 00:00:00	Karnataka	Male	48	...	280	

3. Group sales by customer, product, and category.

Group_sales = df.groupby(['Customer_Name', 'Product_Name', 'Category'])['Total_Amount'].sum().reset_index()
Group_sales



Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
TCS
ENG
11:03
15-09-2025

Group Sales by Customer, Product, and Category

```
[35]: Group_sales = df.groupby(['Customer_Name', 'Product_Name', 'Category'])['Total_Amount'].sum().reset_index()
      Group_sales
```

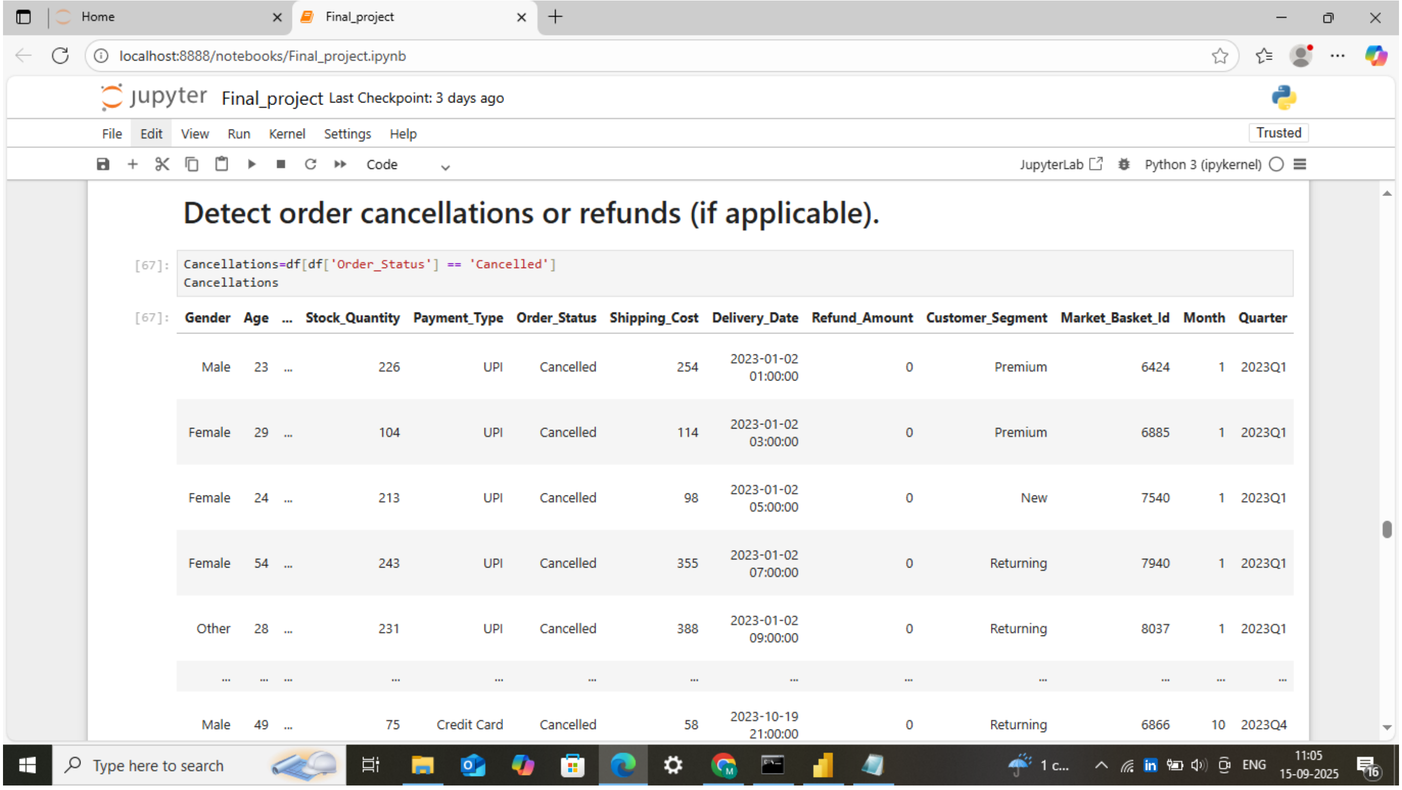
[35]:

	Customer_Name	Product_Name	Category	Total_Amount
0	Aaron Anderson	Smartwatch	Electronics	42899
1	Aaron Austin	Headphones	Electronics	82357
2	Aaron Buckley	Headphones	Accessories	70250
3	Aaron Burton	Camera	Electronics	73197
4	Aaron Clark	Mobile	Accessories	-1467
...	...	...	...	...
6956	Zachary Stein	Laptop	Electronics	11029
6957	Zachary Townsend	Headphones	Electronics	44776
6958	Zachary Turner	Laptop	Accessories	147001
6959	Zachary Wilkinson	Headphones	Electronics	86134
6960	Zoe Harrison	Smartwatch	Accessories	24533

6961 rows × 4 columns

4. Detect order cancellations or refunds (if applicable).

Cancellations=df[df['Order_Status'] == 'Cancelled']
Cancellations



Detect order cancellations or refunds (if applicable).

```
[67]: Cancellations=df[df['Order_Status'] == 'Cancelled']
      Cancellations
```

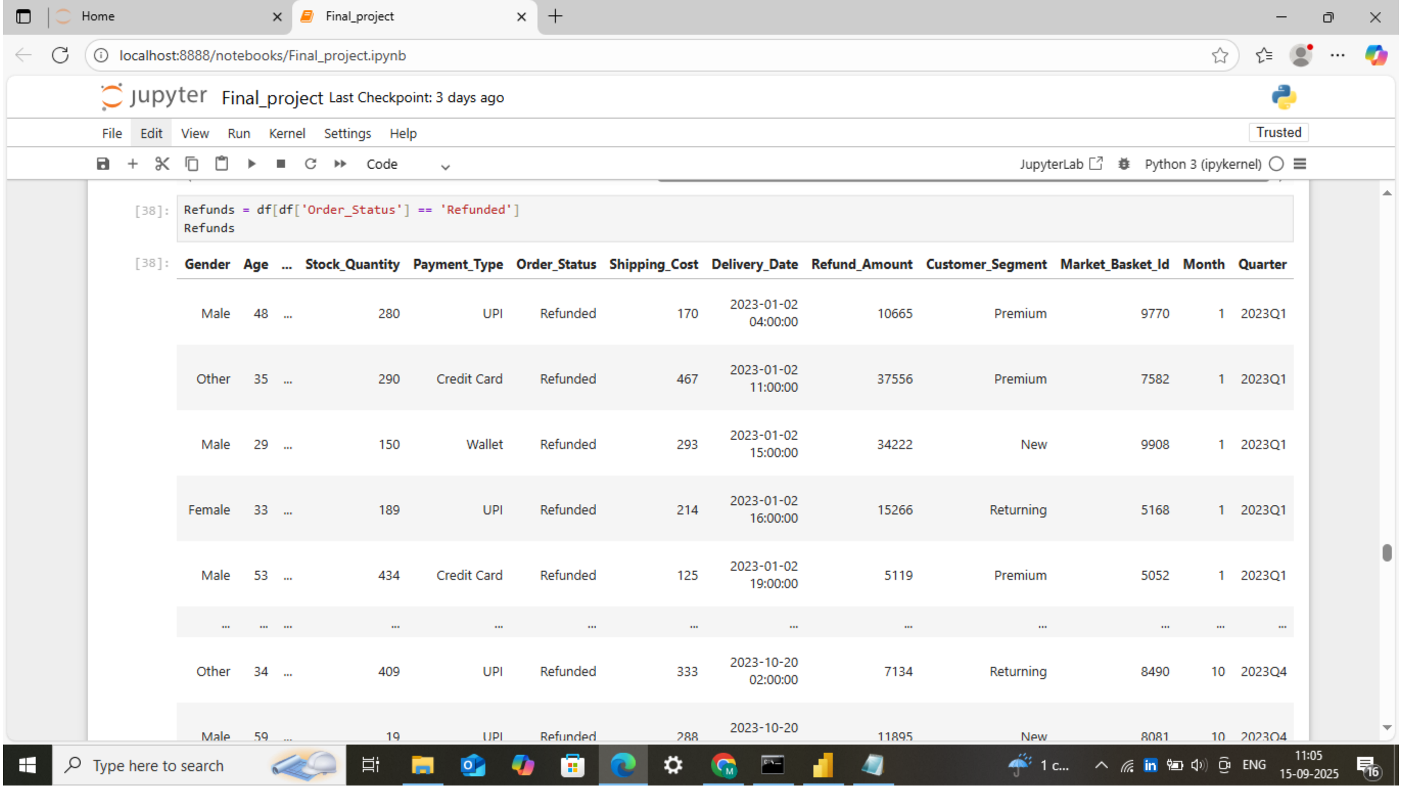
[67]:

Gender	Age	...	Stock_Quantity	Payment_Type	Order_Status	Shipping_Cost	Delivery_Date	Refund_Amount	Customer_Segment	Market_Basket_Id	Month	Quarter
Male	23	...	226	UPI	Cancelled	254	2023-01-02 01:00:00	0	Premium	6424	1	2023Q1
Female	29	...	104	UPI	Cancelled	114	2023-01-02 03:00:00	0	Premium	6885	1	2023Q1
Female	24	...	213	UPI	Cancelled	98	2023-01-02 05:00:00	0	New	7540	1	2023Q1
Female	54	...	243	UPI	Cancelled	355	2023-01-02 07:00:00	0	Returning	7940	1	2023Q1
Other	28	...	231	UPI	Cancelled	388	2023-01-02 09:00:00	0	Returning	8037	1	2023Q1
...	...	...	...	...	...	...	...	...	...	...	...	...
Male	49	...	75	Credit Card	Cancelled	58	2023-10-19 21:00:00	0	Returning	6866	10	2023Q4

Refunds

Refunds = df[df['Order_Status'] == 'Refunded']

Refunds



```
[38]: Refunds = df[df['Order_Status'] == 'Refunded']
      Refunds
```

[38]:

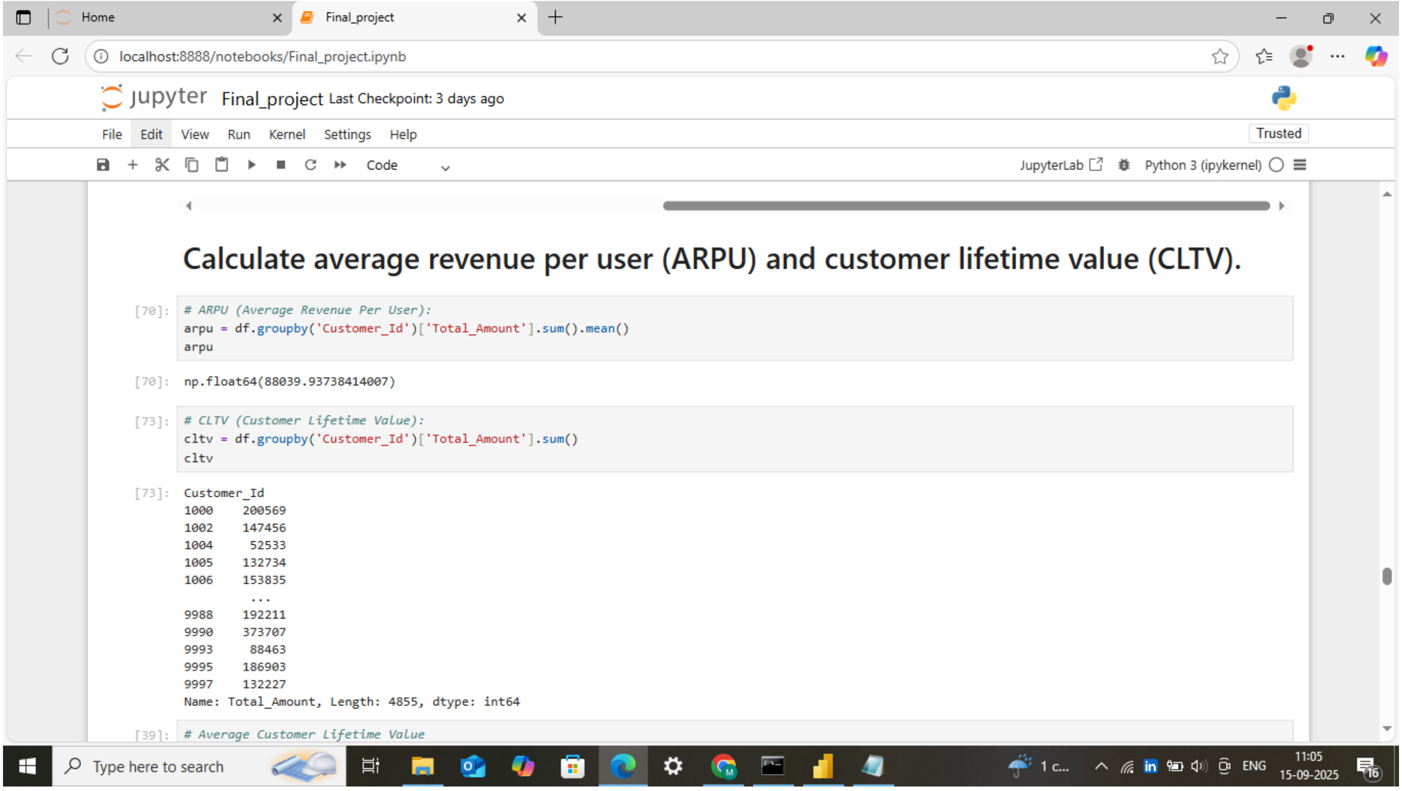
Gender	Age	...	Stock_Quantity	Payment_Type	Order_Status	Shipping_Cost	Delivery_Date	Refund_Amount	Customer_Segment	Market_Basket_Id	Month	Quarter
Male	48	...	280	UPI	Refunded	170	2023-01-02 04:00:00	10665	Premium	9770	1	2023Q1
Other	35	...	290	Credit Card	Refunded	467	2023-01-02 11:00:00	37556	Premium	7582	1	2023Q1
Male	29	...	150	Wallet	Refunded	293	2023-01-02 15:00:00	34222	New	9908	1	2023Q1
Female	33	...	189	UPI	Refunded	214	2023-01-02 16:00:00	15266	Returning	5168	1	2023Q1
Male	53	...	434	Credit Card	Refunded	125	2023-01-02 19:00:00	5119	Premium	5052	1	2023Q1
...	...	...	...	...	...	...	...	...	...	...	...	...
Other	34	...	409	UPI	Refunded	333	2023-10-20 02:00:00	7134	Returning	8490	10	2023Q4
Male	59	...	19	UPI	Refunded	288	2023-10-20	11895	New	8081	10	2023Q4

5. **Calculate average revenue per user (ARPU) and customer lifetime value (CLTV).**
    - **ARPU (Average Revenue Per User):**

arpu = df.groupby('Customer_Id')['Total_Amount'].sum().mean()
arpu

- **CLTV (Customer Lifetime Value):**

cltv = df.groupby('Customer_Id')['Total_Amount'].sum()
cltv

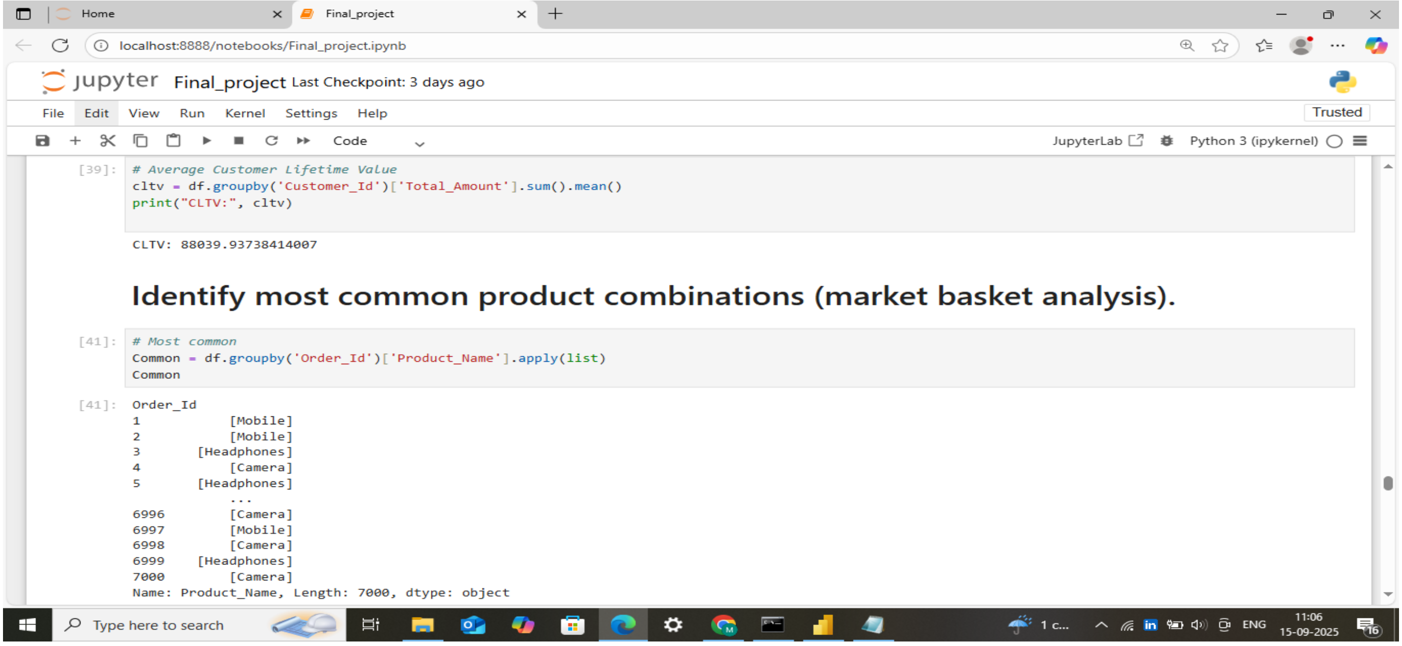

Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
1 c...
ENG
11:05
15-09-2025

Calculate average revenue per user (ARPU) and customer lifetime value (CLTV).

```
[70]: # ARPU (Average Revenue Per User):
      arpu = df.groupby('Customer_Id')['Total_Amount'].sum().mean()
      arpu

[70]: np.float64(88039.93738414007)

[73]: # CLTV (Customer Lifetime Value):
      cltv = df.groupby('Customer_Id')['Total_Amount'].sum()
      cltv

[73]: Customer_Id
      1000    200569
      1002    147456
      1004     52533
      1005    132734
      1006    153835
               ...
      9988    192211
      9990    373707
      9993     88463
      9995    186903
      9997    132227
      Name: Total_Amount, Length: 4855, dtype: int64

[39]: # Average Customer Lifetime Value
```

6. **Identify most common product combinations (market basket analysis).**
    - **Most common**

    Common = df.groupby('Order_Id')['Product_Name'].apply(list)
    Common


Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
1 c...
ENG
11:06
15-09-2025

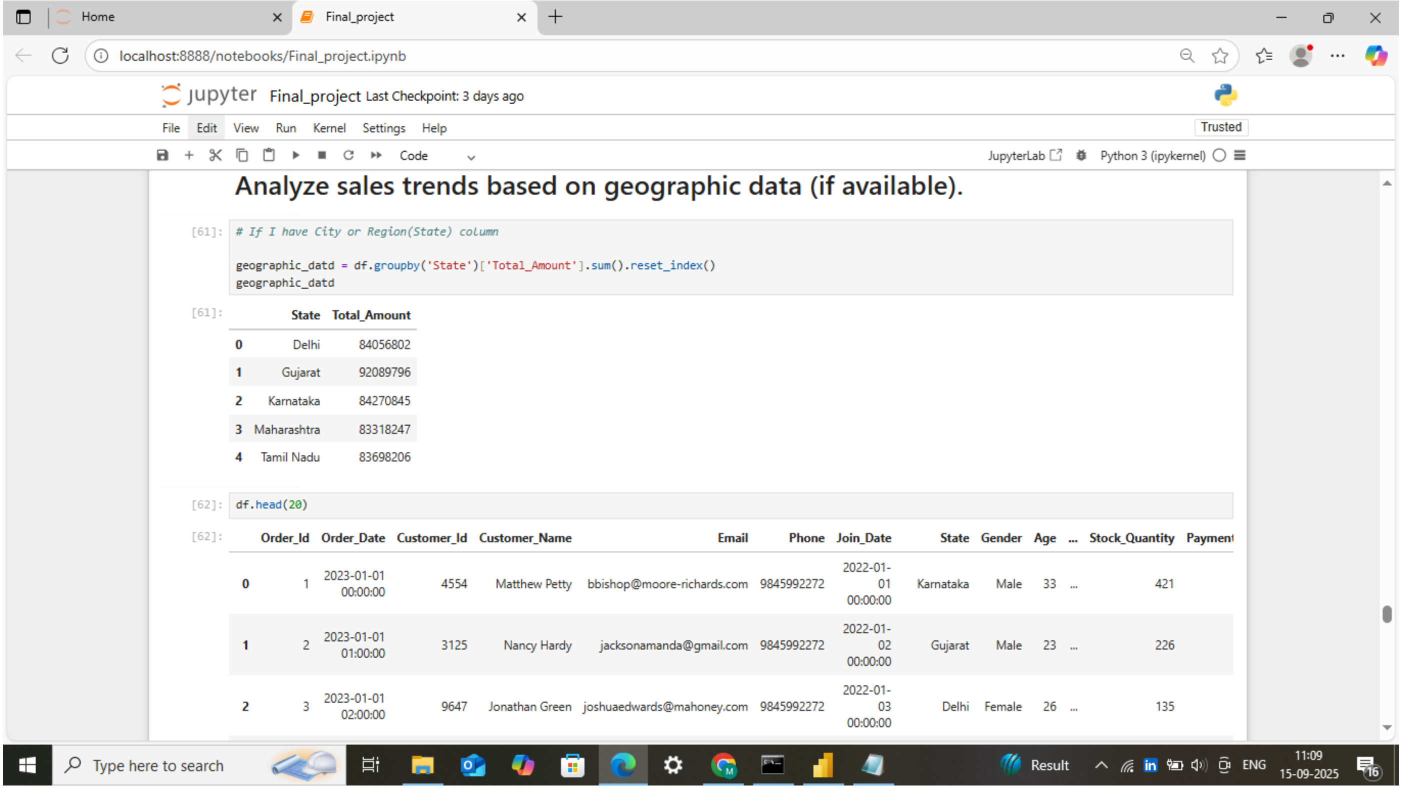
```
[39]: # Average Customer Lifetime Value
      cltv = df.groupby('Customer_Id')['Total_Amount'].sum().mean()
      print("CLTV:", cltv)

      CLTV: 88039.93738414007
```

Identify most common product combinations (market basket analysis).

```
[41]: # Most common
      Common = df.groupby('Order_Id')['Product_Name'].apply(list)
      Common

[41]: Order_Id
      1            [Mobile]
      2            [Mobile]
      3        [Headphones]
      4            [Camera]
      5        [Headphones]
                 ...
      6996         [Camera]
      6997         [Mobile]
      6998         [Camera]
      6999     [Headphones]
      7000         [Camera]
      Name: Product_Name, Length: 7000, dtype: object
```

7. **Analyze sales trends based on geographic data (if available).**
    - **If I have City or Region(State) column**

```
geographic_datd = df.groupby('State')['Total_Amount'].sum().reset_index()
geographic_datd
```


Home
Final_project
localhost:8888/notebooks/Final_project.ipynb
jupyter Final_project Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
Result
ENG
11:09
15-09-2025

Analyze sales trends based on geographic data (if available).

```
[61]: # If I have City or Region(State) column

      geographic_datd = df.groupby('State')['Total_Amount'].sum().reset_index()
      geographic_datd
```

[61]:

	State	Total_Amount
0	Delhi	84056802
1	Gujarat	92089796
2	Karnataka	84270845
3	Maharashtra	83318247
4	Tamil Nadu	83698206

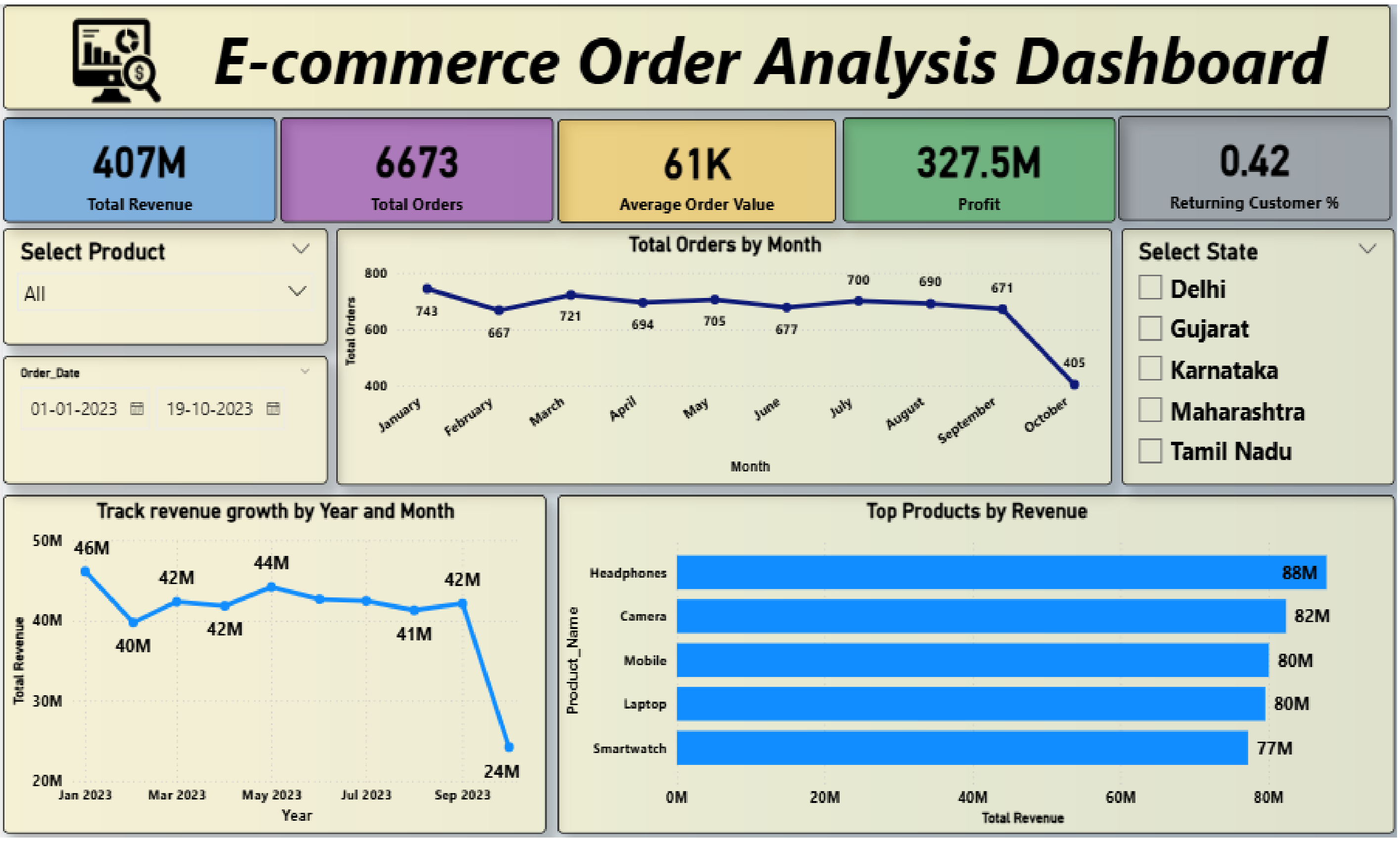
```
[62]: df.head(20)
```

[62]:

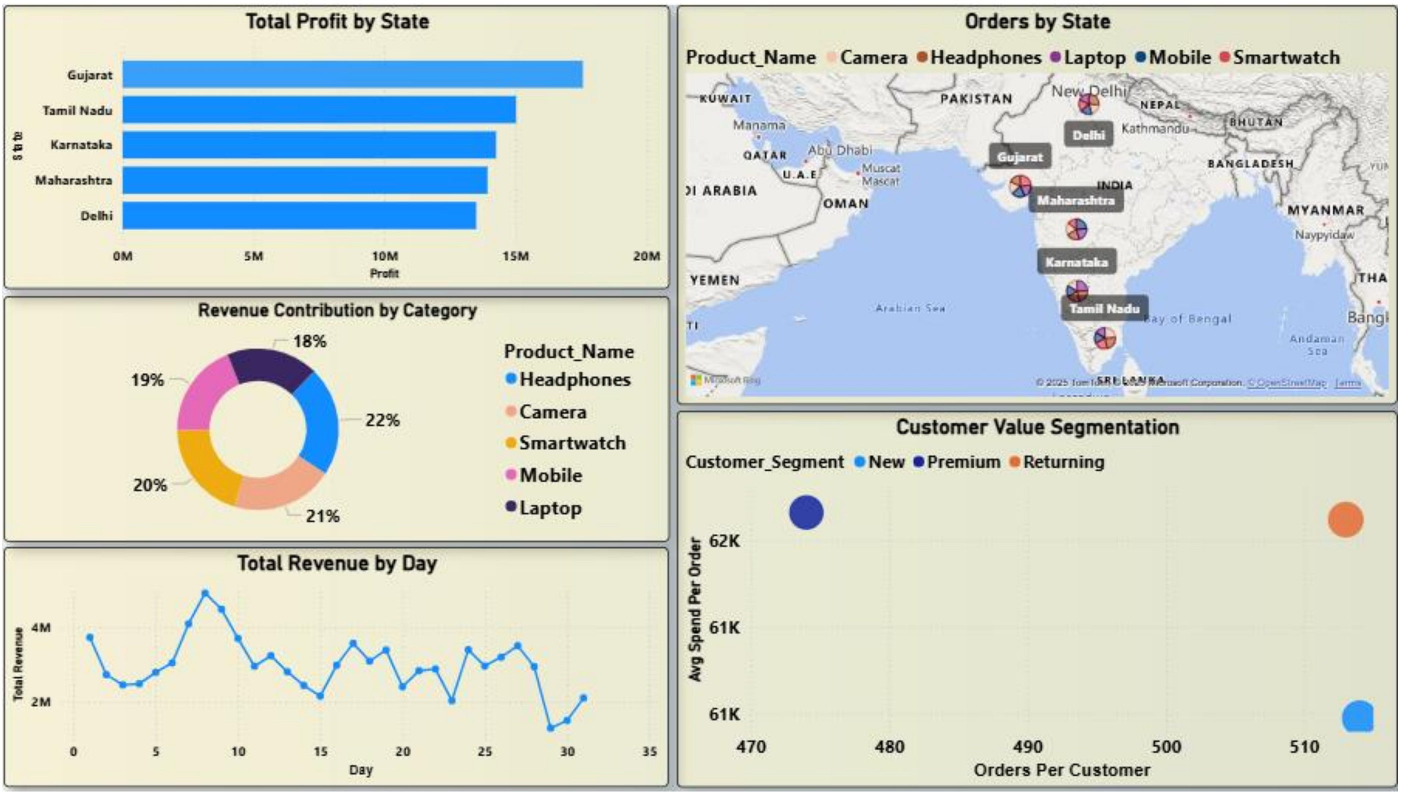
	Order_Id	Order_Date	Customer_Id	Customer_Name	Email	Phone	Join_Date	State	Gender	Age	...	Stock_Quantity	Payment
0	1	2023-01-01 00:00:00	4554	Matthew Petty	bbishop@moore-richards.com	9845992272	2022-01-01 00:00:00	Karnataka	Male	33	...	421	
1	2	2023-01-01 01:00:00	3125	Nancy Hardy	jacksonamanda@gmail.com	9845992272	2022-01-02 00:00:00	Gujarat	Male	23	...	226	
2	3	2023-01-01 02:00:00	9647	Jonathan Green	joshuaedwards@mahoney.com	9845992272	2022-01-03 00:00:00	Delhi	Female	26	...	135	

Power BI

Order Analysis Overview



Insight Analysis



<u>Project Summary</u>

In this project, I analyze E-commerce sales data using SQL, Python, and Power BI. With SQL, I retrieved top-selling products, monthly revenue, best categories, and repeat customers. Using Python Pandas, I cleaned the dataset, removed duplicates, handled missing values, and analyze sales trends.
I also calculated average revenue per customer, customer lifetime value, and studied product combinations. Finally, in Power BI, I built an interactive dashboard with charts, KPIs, and slicers. This dashboard shows order trends, revenue growth, product demand, customer segmentation, and revenue contribution by category.

Overall, the project gives a clear and meaningful view of business performance and helps in better decision-making.

Conclusion

This project helped to understand E-commerce sales patterns and customer behavior. By using SQL, I got important business insights like top products, revenue trends, and best categories. With Python Pandas, I cleaned and analyze data to find seasonal trends, customer value, and product combinations.

Finally, with Power BI dashboard, I created an interactive report to visualize sales, revenue, and customer segmentation.

The analysis shows that such insights can help businesses improve sales strategies, identify growth opportunities, and make better decisions.