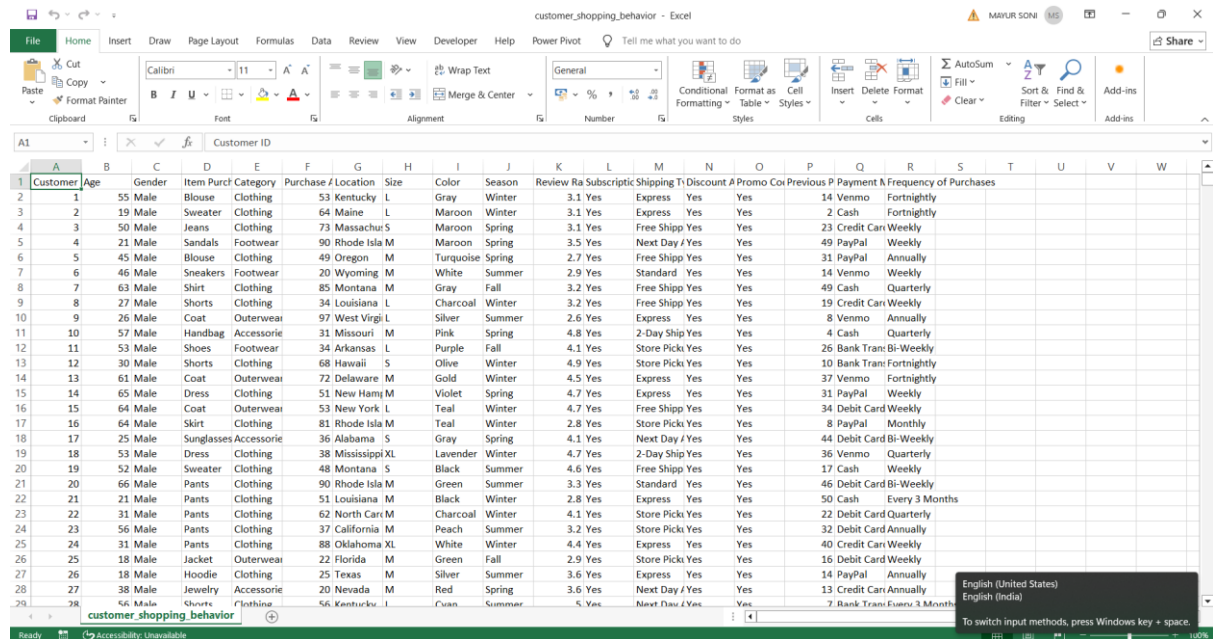


Customer Shopping Behaviour Analysis Report

1. Project Goal

I looked closely at **3,900 customer purchases** to understand how people shop. The main goal was to find out who spends what, what they buy most often, and if they're subscribing. These insights will help us make smarter business decisions.



Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method
1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo
2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	Cash
3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	Credit Card
4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	PayPal
5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	PayPal
6	46	Male	Sneakers	Footwear	20	Wyoming	M	White	Summer	2.9	Yes	Standard	Yes	Yes	14	Venmo
7	63	Male	Shirt	Clothing	85	Montana	M	Gray	Fall	3.2	Yes	Free Shipping	Yes	Yes	49	Cash
8	27	Male	Shorts	Clothing	34	Louisiana	L	Charcoal	Winter	3.2	Yes	Free Shipping	Yes	Yes	19	Credit Card
9	26	Male	Coat	Outerwear	97	West Virginia	L	Silver	Summer	2.6	Yes	Express	Yes	Yes	8	Venmo
10	57	Male	Handbag	Accessories	31	Missouri	M	Pink	Spring	4.8	Yes	2-Day Ship	Yes	Yes	4	Cash
11	53	Male	Shoes	Footwear	34	Arkansas	L	Purple	Fall	4.1	Yes	Store Pick	Yes	Yes	26	Bank Trans
12	30	Male	Shorts	Clothing	68	Hawaii	S	Olive	Winter	4.9	Yes	Store Pick	Yes	Yes	10	Bank Trans
13	61	Male	Coat	Outerwear	72	Delaware	M	Gold	Winter	4.5	Yes	Express	Yes	Yes	37	Venmo
14	65	Male	Dress	Clothing	51	New Hampshire	M	Violet	Spring	4.7	Yes	Express	Yes	Yes	31	PayPal
15	64	Male	Coat	Outerwear	53	New York	L	Teal	Winter	4.7	Yes	Free Shipping	Yes	Yes	34	Debit Card
16	64	Male	Skirt	Clothing	81	Rhode Island	M	Teal	Winter	2.8	Yes	Store Pick	Yes	Yes	8	PayPal
17	25	Male	Sunglasses	Accessories	36	Alabama	S	Gray	Spring	4.1	Yes	Next Day	Yes	Yes	44	Debit Card
18	53	Male	Dress	Clothing	38	Mississippi	XL	Lavender	Winter	4.7	Yes	2-Day Ship	Yes	Yes	36	Venmo
19	52	Male	Sweater	Clothing	48	Montana	S	Black	Summer	4.6	Yes	Free Shipping	Yes	Yes	17	Cash
20	66	Male	Pants	Clothing	90	Rhode Island	M	Green	Summer	3.3	Yes	Standard	Yes	Yes	46	Debit Card
21	21	Male	Pants	Clothing	51	Louisiana	M	Black	Winter	2.8	Yes	Express	Yes	Yes	50	Cash
22	31	Male	Pants	Clothing	62	North Carolina	M	Charcoal	Winter	4.1	Yes	Store Pick	Yes	Yes	22	Debit Card
23	56	Male	Pants	Clothing	37	California	M	Peach	Summer	3.2	Yes	Store Pick	Yes	Yes	32	Debit Card
24	31	Male	Pants	Clothing	88	Oklahoma	XL	White	Winter	4.4	Yes	Express	Yes	Yes	40	Credit Card
25	18	Male	Jacket	Outerwear	22	Florida	M	Green	Fall	2.9	Yes	Store Pick	Yes	Yes	16	Debit Card
26	26	Male	Hoodie	Clothing	25	Texas	M	Silver	Summer	3.6	Yes	Express	Yes	Yes	14	PayPal
27	38	Male	Jewelry	Accessories	20	Nevada	M	Red	Spring	3.6	Yes	Next Day	Yes	Yes	13	Credit Card
28	46	Male	Shirts	Clothing	56	Kentucky	L	Cyan	Summer	5	Yes	Next Day	Yes	Yes	7	Bank Trans

2. The Data Snapshot

- Total Purchases: 3,900

Home

Customer_Shopping_Behavior

localhost:8888/notebooks/Customer_Shopping_Behavior_Analysis.ipynb?

Intership Adobe Acrobat Dashboard | ExcelR All Bookmarks

jupyter Customer_Shopping_Behavior_Analysis Last Checkpoint: 4 days ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python [conda env:base] Anaconda Toolbox

[4]: import pandas as pd
df = pd.read_csv(r"C:\Users\Lenovo\Desktop\dataset\customer_shopping_behavior.csv")

[5]: df.head()

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	Cash
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	Credit Card
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	PayPal
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	PayPal

[6]: df.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 39000 entries, 0 to 38999  
Data columns (total 18 columns):  
# Column Non-Null Count Dtype  
-----  
0 1 38999 non-null int64  
1 2 38999 non-null int64  
2 3 38999 non-null object  
3 4 38999 non-null object  
4 5 38999 non-null object  
5 6 38999 non-null float64  
6 7 38999 non-null object  
7 8 38999 non-null object  
8 9 38999 non-null object  
9 10 38999 non-null object  
10 11 38999 non-null float64  
11 12 38999 non-null object  
13 14 38999 non-null object  
14 15 38999 non-null object  
15 16 38999 non-null int64  
16 17 38999 non-null object
```

- **Details Tracked:** 18 pieces of information per purchase (e.g., age, what they bought, price, if they used a discount, shipping type, and review rating).

```
[6]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   Customer ID           3900 non-null   int64   
 1   Age                   3900 non-null   int64   
 2   Gender                3900 non-null   object  
 3   Item Purchased        3900 non-null   object  
 4   Category              3900 non-null   object  
 5   Purchase Amount (USD) 3900 non-null   int64   
 6   Location              3900 non-null   object  
 7   Size                  3900 non-null   object  
 8   Color                 3900 non-null   object  
 9   Season                3900 non-null   object  
10   Review Rating         3863 non-null   float64  
11   Subscription Status   3900 non-null   object  
12   Shipping Type         3900 non-null   object  
13   Discount Applied      3900 non-null   object  
14   Promo Code Used       3900 non-null   object  
15   Previous Purchases    3900 non-null   int64   
16   Payment Method        3900 non-null   object  
17   Frequency of Purchases 3900 non-null   object  
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

- **Small Fix Needed:** I had to fill in **37 missing review scores**.

3. Cleaning Up the Data (Using Python)

I started by getting the data ready for analysis:

- **Load and Check:** I opened the data and checked its basic structure.
- **Missing Scores:** I filled in the missing product review scores by using the **median rating** for that specific product type.

```
[8]: df.isnull().sum()

[8]: Customer ID           0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             37
Subscription Status       0
Shipping Type             0
Discount Applied          0
Promo Code Used           0
Previous Purchases        0
Payment Method            0
Frequency of Purchases    0
dtype: int64

[9]: df['Review Rating'] = df.groupby('Category')['Review Rating'].fillna(df.groupby('Category')['Review Rating'].median())

[9]: Customer ID           0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             0
Subscription Status       0
Shipping Type             0
Discount Applied          0
Promo Code Used           0
Previous Purchases        0
Payment Method            0
Frequency of Purchases    0
dtype: int64
```

- **Tidying Up:** I renamed columns to be simple and easy to read (like item purchased instead of ItemPurchased).
- **New Insights Created:**
 - I grouped customers into **Age Groups** (like 'Young Adults', 'Middle Aged', etc.).

- I figured out how many days pass between purchases.
- **Removing Duplicates:** I found that the 'promo code' column was basically the same as the 'discount applied' column, so I only kept the '**discount applied**' one.
- **Ready for Deep Dive:** I saved the cleaned data into a **PostgreSQL database** for My detailed analysis using SQL.

```

Downloaded psycopg2-binary-2.9.11-cp313-cp313-win_amd64.whl.metadata (5.1 kB)
Requirement already satisfied: sqlalchemy in c:\users\lenovo\anaconda3\lib\site-packages (2.0.39)
Requirement already satisfied: greenlet==0.4.17 in c:\users\lenovo\anaconda3\lib\site-packages (from sqlalchemy) (3.1.1)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\lenovo\anaconda3\lib\site-packages (from sqlalchemy) (4.12.2)
Downloading psycopg2-binary-2.9.11-cp313-cp313-win_amd64.whl (2.7 MB)
----- 0.0/2.7 MB ? eta -:--:--
----- 2.6/2.7 MB 13.5 MB/s eta 0:00:01
----- 2.7/2.7 MB 12.0 MB/s eta 0:00:00
Installing collected packages: psycopg2-binary
Successfully installed psycopg2-binary-2.9.11

[29]: from sqlalchemy import create_engine

[35]: username = "postgres"
password = "Password"
host = "localhost"
port = "5432"
database = "customer_shopping_behavior"

engine = create_engine(f"postgresql+psycopg2://{username}:{password}@{host}:{port}/{database}")

[36]: table_name = "customer"
df.to_sql(table_name, engine, if_exists="replace", index=False)

print(f"Data successfully loaded into table '{table_name}' in database '{database}'.")

username = "postgres"

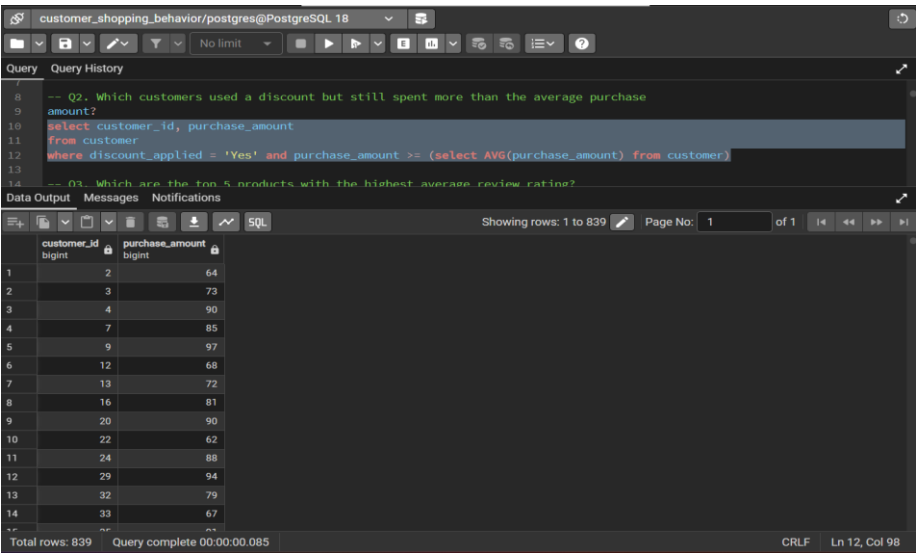
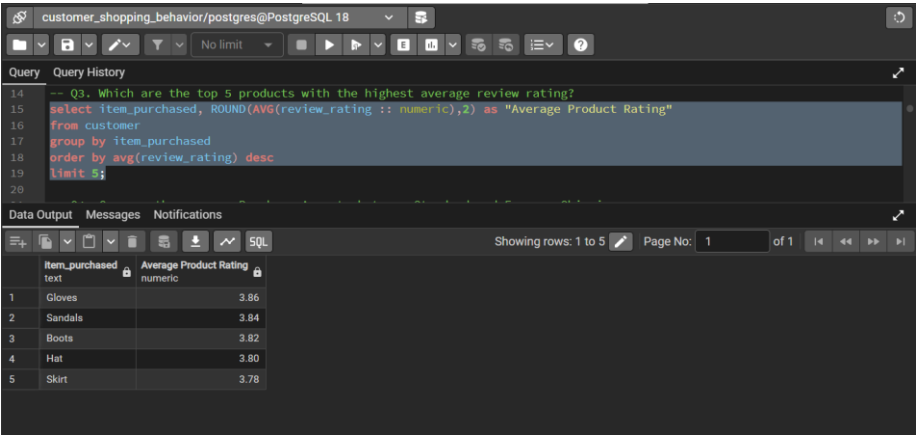
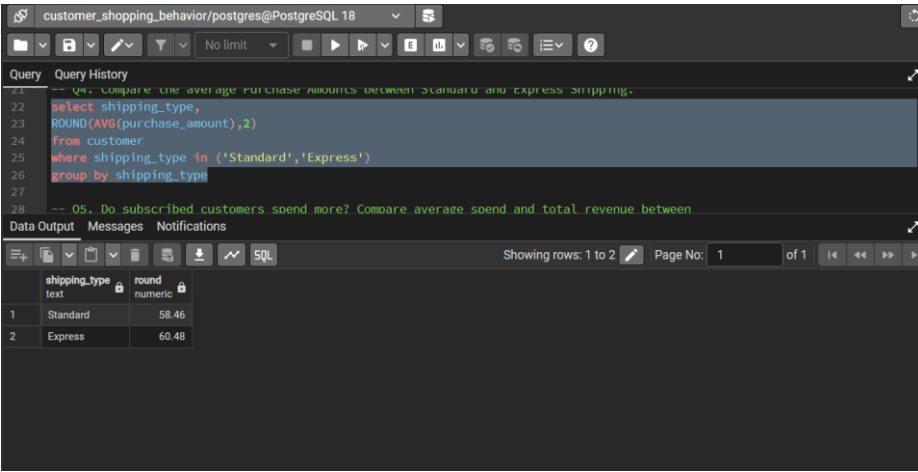
Data successfully loaded into table 'customer' in database 'customer_shopping_behavior'.

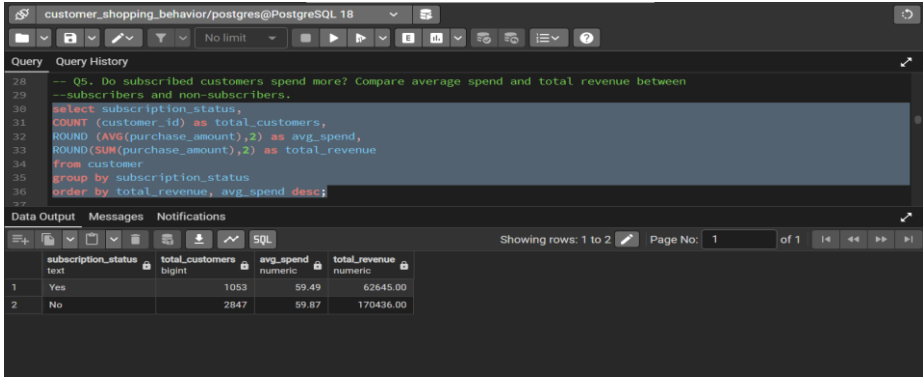
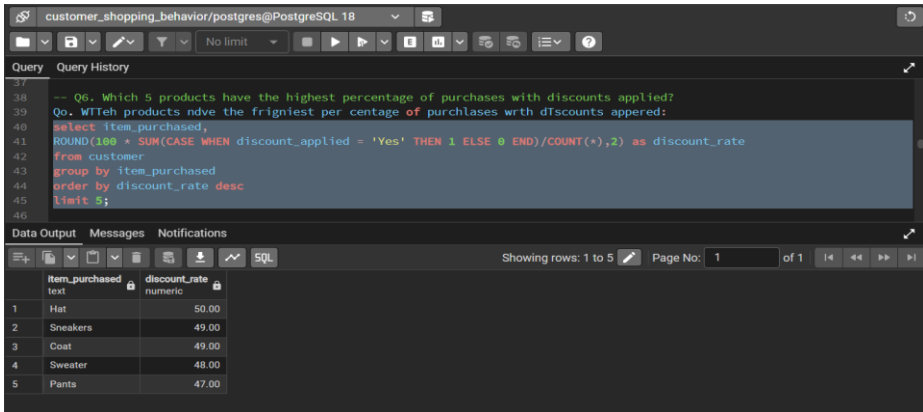
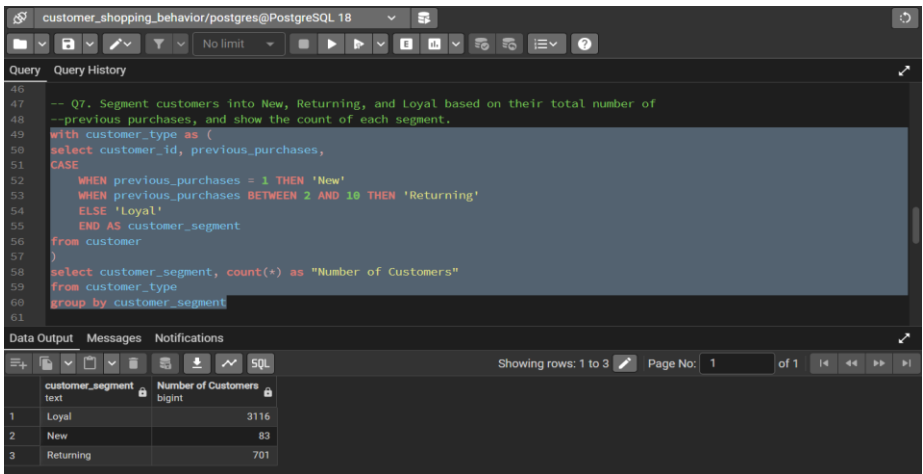
```

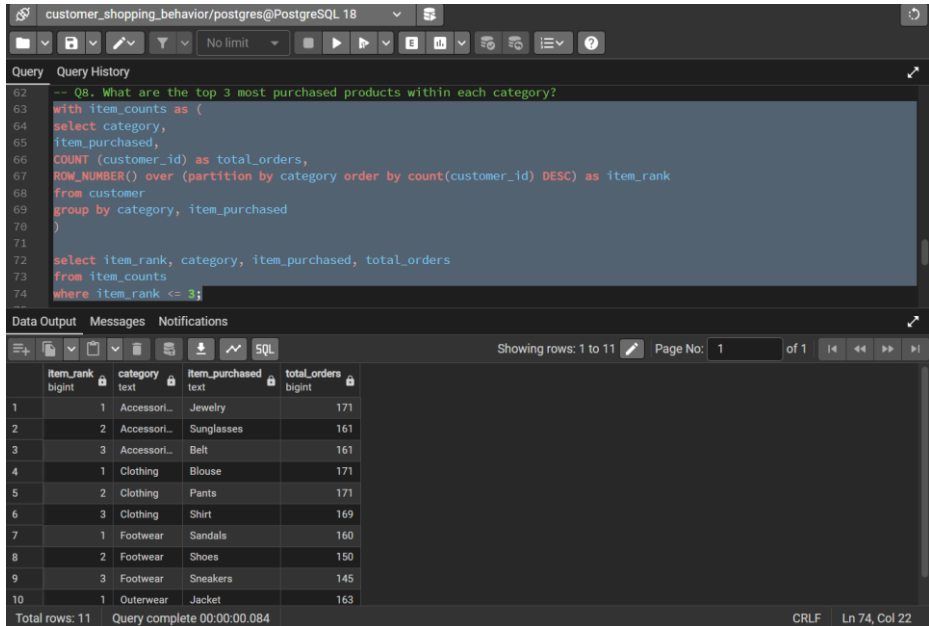
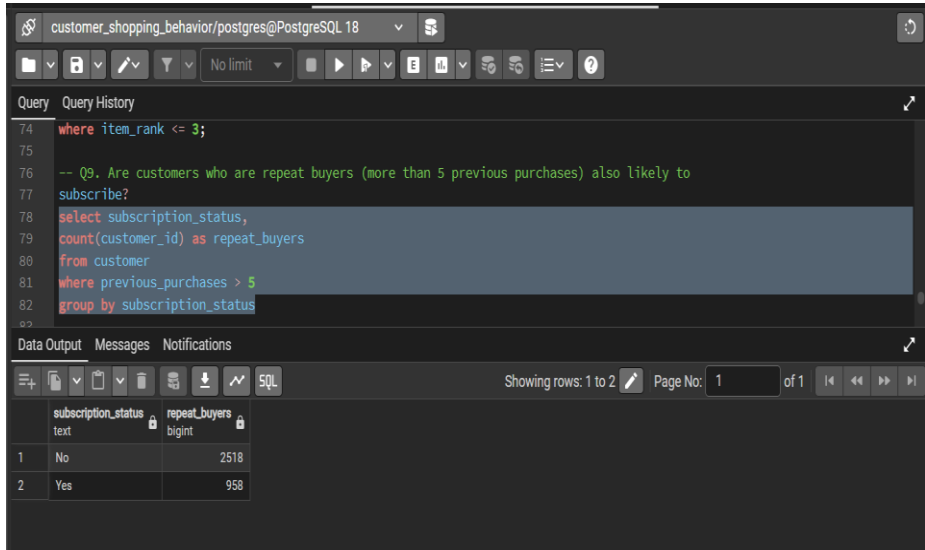
4. Key Findings from the Analysis (Using SQL)

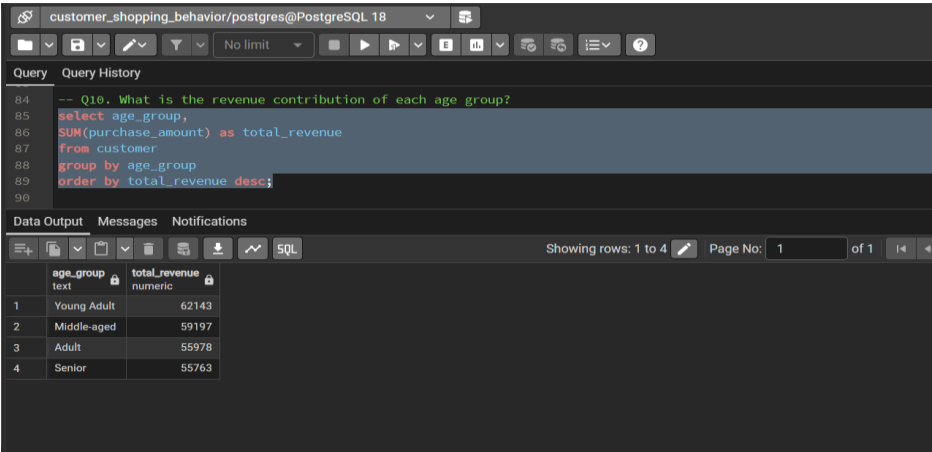
I ran specific checks to Answer important business questions:

Question	What I Looked For						
Spending by Gender	<p>Who brings in more total sales: men or women?</p> <pre> 1 select * from customer limit 20 2 3 -- Q1. What is the total revenue generated by male vs. female customers? 4 select gender, SUM(purchase_amount) as revenue 5 from customer 6 group by gender 7 </pre> <table border="1"> <thead> <tr> <th>gender</th><th>revenue</th></tr> </thead> <tbody> <tr> <td>Female</td><td>75191</td></tr> <tr> <td>Male</td><td>157890</td></tr> </tbody> </table>	gender	revenue	Female	75191	Male	157890
gender	revenue						
Female	75191						
Male	157890						
Smart Discount Users	Which customers used a discount but still spent more than the average amount?						

Question	What I Looked For																														
	 <p>The screenshot shows a PostgreSQL query window with the following SQL code:</p> <pre>-- Q2. Which customers used a discount but still spent more than the average purchase amount? select customer_id, purchase_amount from customer where discount_applied = 'Yes' and purchase_amount >= (select AVG(purchase_amount) from customer)</pre> <p>The data output shows 14 rows of results:</p> <table border="1"><thead><tr><th>customer_id</th><th>purchase_amount</th></tr></thead><tbody><tr><td>1</td><td>64</td></tr><tr><td>2</td><td>73</td></tr><tr><td>3</td><td>90</td></tr><tr><td>4</td><td>85</td></tr><tr><td>5</td><td>97</td></tr><tr><td>6</td><td>68</td></tr><tr><td>7</td><td>72</td></tr><tr><td>8</td><td>81</td></tr><tr><td>9</td><td>90</td></tr><tr><td>10</td><td>62</td></tr><tr><td>11</td><td>88</td></tr><tr><td>12</td><td>94</td></tr><tr><td>13</td><td>79</td></tr><tr><td>14</td><td>67</td></tr></tbody></table> <p>Total rows: 839 Query complete 00:00:00.085</p>	customer_id	purchase_amount	1	64	2	73	3	90	4	85	5	97	6	68	7	72	8	81	9	90	10	62	11	88	12	94	13	79	14	67
customer_id	purchase_amount																														
1	64																														
2	73																														
3	90																														
4	85																														
5	97																														
6	68																														
7	72																														
8	81																														
9	90																														
10	62																														
11	88																														
12	94																														
13	79																														
14	67																														
Best Loved Products	<p>The Top 5 products based on their high average review score.</p>  <p>The screenshot shows a PostgreSQL query window with the following SQL code:</p> <pre>-- Q3. Which are the top 5 products with the highest average review rating? select item_purchased, ROUND(AVG(review_rating::numeric),2) as "Average Product Rating" from customer group by item_purchased order by avg(review_rating) desc limit 5;</pre> <p>The data output shows 5 rows of results:</p> <table border="1"><thead><tr><th>item_purchased</th><th>Average Product Rating</th></tr></thead><tbody><tr><td>Gloves</td><td>3.86</td></tr><tr><td>Sandals</td><td>3.84</td></tr><tr><td>Boots</td><td>3.82</td></tr><tr><td>Hat</td><td>3.80</td></tr><tr><td>Skirt</td><td>3.78</td></tr></tbody></table>	item_purchased	Average Product Rating	Gloves	3.86	Sandals	3.84	Boots	3.82	Hat	3.80	Skirt	3.78																		
item_purchased	Average Product Rating																														
Gloves	3.86																														
Sandals	3.84																														
Boots	3.82																														
Hat	3.80																														
Skirt	3.78																														
Shipping Impact	<p>Does the purchase amount change between Standard and Express shipping?</p>  <p>The screenshot shows a PostgreSQL query window with the following SQL code:</p> <pre>-- Q4. Compare the average purchase amounts between standard and express shipping. select shipping_type, ROUND(AVG(purchase_amount),2) from customer where shipping_type in ('Standard','Express') group by shipping_type</pre> <p>The data output shows 2 rows of results:</p> <table border="1"><thead><tr><th>shipping_type</th><th>round</th></tr></thead><tbody><tr><td>Standard</td><td>58.46</td></tr><tr><td>Express</td><td>60.48</td></tr></tbody></table>	shipping_type	round	Standard	58.46	Express	60.48																								
shipping_type	round																														
Standard	58.46																														
Express	60.48																														

Question	What I Looked For												
Subscribers vs. Others	<p>How much more do subscribers spend, and how much total revenue do they generate compared to non-subscribers?</p>  <pre>-- Q5. Do subscribed customers spend more? Compare average spend and total revenue between --subscribers and non-subscribers. select subscription_status, count(customer_id) as total_customers, round(avg(purchase_amount),2) as avg_spend, round(sum(purchase_amount),2) as total_revenue from customer group by subscription_status order by total_revenue, avg_spend desc;</pre> <table><tr><th>subscription_status</th><th>total_customers</th><th>avg_spend</th><th>total_revenue</th></tr><tr><td>Yes</td><td>1053</td><td>59.49</td><td>62645.00</td></tr><tr><td>No</td><td>2847</td><td>59.87</td><td>170436.00</td></tr></table>	subscription_status	total_customers	avg_spend	total_revenue	Yes	1053	59.49	62645.00	No	2847	59.87	170436.00
subscription_status	total_customers	avg_spend	total_revenue										
Yes	1053	59.49	62645.00										
No	2847	59.87	170436.00										
"Discount Only" Items	<p>The Top 5 products that are most often bought <i>only</i> when a discount is offered.</p>  <pre>-- Q6. Which 5 products have the highest percentage of purchases with discounts applied? -- With products ndve the frigniest per centage of purchLases wrth dIscounts appered: select item_purchased, round(100 * sum(case when discount_applied = 'Yes' then 1 else 0 end)/count(*),2) as discount_rate from customer group by item_purchased order by discount_rate desc limit 5;</pre> <table><tr><th>item_purchased</th><th>discount_rate</th></tr><tr><td>Hat</td><td>50.00</td></tr><tr><td>Sneakers</td><td>49.00</td></tr><tr><td>Coat</td><td>49.00</td></tr><tr><td>Sweater</td><td>48.00</td></tr><tr><td>Pants</td><td>47.00</td></tr></table>	item_purchased	discount_rate	Hat	50.00	Sneakers	49.00	Coat	49.00	Sweater	48.00	Pants	47.00
item_purchased	discount_rate												
Hat	50.00												
Sneakers	49.00												
Coat	49.00												
Sweater	48.00												
Pants	47.00												
Customer Groups	<p>How to categorize customers based on how much they've bought before: New, Returning, or Loyal?</p>  <pre>-- Q7. Segment customers into New, Returning, and Loyal based on their total number of --previous purchases, and show the count of each segment. with customer_type as (select customer_id, previous_purchases, case when previous_purchases = 1 then 'New' when previous_purchases between 2 and 10 then 'Returning' else 'Loyal' end as customer_segment from customer) select customer_segment, count(*) as "Number of Customers" from customer_type group by customer_segment;</pre> <table><tr><th>customer_segment</th><th>Number of Customers</th></tr><tr><td>Loyal</td><td>3116</td></tr><tr><td>New</td><td>83</td></tr><tr><td>Returning</td><td>701</td></tr></table>	customer_segment	Number of Customers	Loyal	3116	New	83	Returning	701				
customer_segment	Number of Customers												
Loyal	3116												
New	83												
Returning	701												

Question	What I Looked For																																												
Top Items in Each Category	<p>The three most purchased products within <i>each</i> product category (e.g., most purchased in "Apparel").</p>  <pre>-- Q8. What are the top 3 most purchased products within each category? with item_counts as (select category, item_purchased, COUNT(customer_id) as total_orders, ROW_NUMBER() over (partition by category order by count(customer_id) DESC) as item_rank from customer group by category, item_purchased) select item_rank, category, item_purchased, total_orders from item_counts where item_rank <= 3;</pre> <table><thead><tr><th>item_rank</th><th>category</th><th>item_purchased</th><th>total_orders</th></tr></thead><tbody><tr><td>1</td><td>Accessori...</td><td>Jewelry</td><td>171</td></tr><tr><td>2</td><td>Accessori...</td><td>Sunglasses</td><td>161</td></tr><tr><td>3</td><td>Accessori...</td><td>Belt</td><td>161</td></tr><tr><td>4</td><td>Clothing</td><td>Blouse</td><td>171</td></tr><tr><td>5</td><td>Clothing</td><td>Pants</td><td>171</td></tr><tr><td>6</td><td>Clothing</td><td>Shirt</td><td>169</td></tr><tr><td>7</td><td>Footwear</td><td>Sandals</td><td>160</td></tr><tr><td>8</td><td>Footwear</td><td>Shoes</td><td>150</td></tr><tr><td>9</td><td>Footwear</td><td>Sneakers</td><td>145</td></tr><tr><td>10</td><td>Outerwear</td><td>Jacket</td><td>163</td></tr></tbody></table>	item_rank	category	item_purchased	total_orders	1	Accessori...	Jewelry	171	2	Accessori...	Sunglasses	161	3	Accessori...	Belt	161	4	Clothing	Blouse	171	5	Clothing	Pants	171	6	Clothing	Shirt	169	7	Footwear	Sandals	160	8	Footwear	Shoes	150	9	Footwear	Sneakers	145	10	Outerwear	Jacket	163
item_rank	category	item_purchased	total_orders																																										
1	Accessori...	Jewelry	171																																										
2	Accessori...	Sunglasses	161																																										
3	Accessori...	Belt	161																																										
4	Clothing	Blouse	171																																										
5	Clothing	Pants	171																																										
6	Clothing	Shirt	169																																										
7	Footwear	Sandals	160																																										
8	Footwear	Shoes	150																																										
9	Footwear	Sneakers	145																																										
10	Outerwear	Jacket	163																																										
Loyalty & Subscriptions	<p>Are customers who buy more than 5 times more likely to be subscribers?</p>  <pre>-- Q9. Are customers who are repeat buyers (more than 5 previous purchases) also likely to subscribe? select subscription_status, count(customer_id) as repeat_buyers from customer where previous_purchases > 5 group by subscription_status;</pre> <table><thead><tr><th>subscription_status</th><th>repeat_buyers</th></tr></thead><tbody><tr><td>No</td><td>2518</td></tr><tr><td>Yes</td><td>958</td></tr></tbody></table>	subscription_status	repeat_buyers	No	2518	Yes	958																																						
subscription_status	repeat_buyers																																												
No	2518																																												
Yes	958																																												
Revenue by Age	Which age groups bring in the most money?																																												

Question	What I Looked For										
	 <p>The screenshot shows a PostgreSQL query editor with the following SQL query:</p> <pre>-- Q10. What is the revenue contribution of each age group? select age_group, SUM(purchase_amount) as total_revenue from customer group by age_group order by total_revenue desc;</pre> <p>The results table shows the following data:</p> <table><thead><tr><th>age_group</th><th>total_revenue</th></tr></thead><tbody><tr><td>Young Adult</td><td>62143</td></tr><tr><td>Middle-aged</td><td>59197</td></tr><tr><td>Adult</td><td>55978</td></tr><tr><td>Senior</td><td>55763</td></tr></tbody></table>	age_group	total_revenue	Young Adult	62143	Middle-aged	59197	Adult	55978	Senior	55763
age_group	total_revenue										
Young Adult	62143										
Middle-aged	59197										
Adult	55978										
Senior	55763										

5. Visualizing the Data (Polr BI)

I created an **interactive dashboard** in Power BI to easily see all these findings in charts and graphs.



6. My Recommendations for the Business

Based on the data, here's how I should move forward:

- **Grow Subscriptions:** Offer **exclusive perks** to encourage more people to sign up for subscriptions.
 - **Reward Loyalty:** Create a rewards program to turn **returning customers** into **loyal, high-value buyers**.
 - **Fine-Tune Discounts:** Carefully check My discount strategy to ensure sales boosts don't hurt My profit margins too much.
 - **Showcase the Best:** Run marketing campaigns that **highlight the top-rated and best-selling products**.
 - **Focus your Ads:** Direct marketing efforts toward the **age groups** that spend the most, and consider special promotions for those who choose **Express shipping**.
-