Customer Shopping Behavior Analysis

1. Project Overview

This project analyzes customer shopping behavior using transactional data from 3,900 purchases across various product categories. The goal is to uncover insights into spending patterns, customer segments, product preferences, and subscription behavior to guide strategic business decisions.

2. Dataset Summary

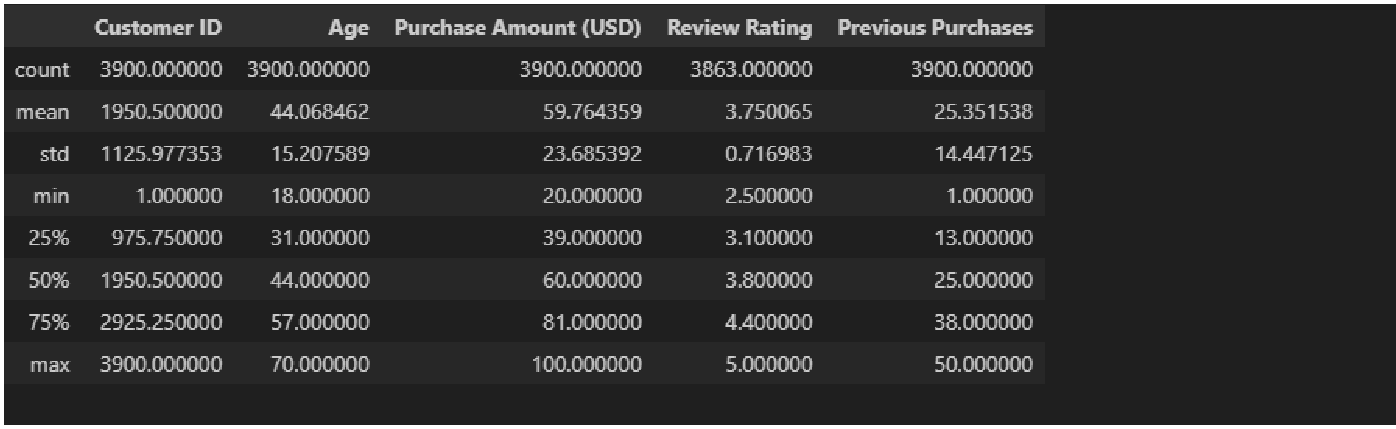
- Rows: 3,900
- Columns: 18
- Key Features:
  - Customer demographics (Age, Gender, Location, Subscription Status)
  - Purchase details (Item Purchased, Category, Purchase Amount, Season, Size, Color)
  - Shopping behavior (Discount Applied, Promo Code Used, Previous Purchases, Frequency of Purchases, Review Rating, Shipping Type)
- Missing Data: 37 values in Review Rating column

3. Exploratory Data Analysis using Python

We began with data preparation and cleaning in Python:

- **Data Loading:** Imported the dataset using `pandas`.

- **Initial Exploration:** Used `df.info()` to check structure and `.describe()` for summary statistics.



	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3863.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.750065	25.351538
std	1125.977353	15.207589	23.685392	0.716983	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.800000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

- **Missing Data Handling:** Checked for null values and imputed missing values in the `Review Rating` column using the median rating of each product category.

- **Column Standardization:** Renamed columns to **snake case** for better readability and documentation.

- **Feature Engineering:**

    - Created **age_group** column by binning customer ages.

    - Created **purchase_frequency_days** column from purchase data.

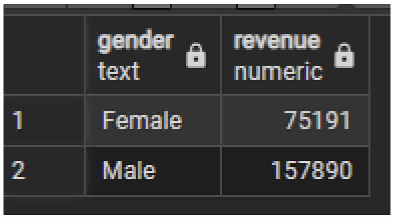
- **Data Consistency Check:** Verified if `discount_applied` and `promo_code_used` were redundant; dropped `promo_code_used`.

- **Database Integration:** Connected Python script to PostgreSQL and loaded the cleaned DataFrame into the database for SQL analysis.

4. Data Analysis using SQL (Business Transactions)

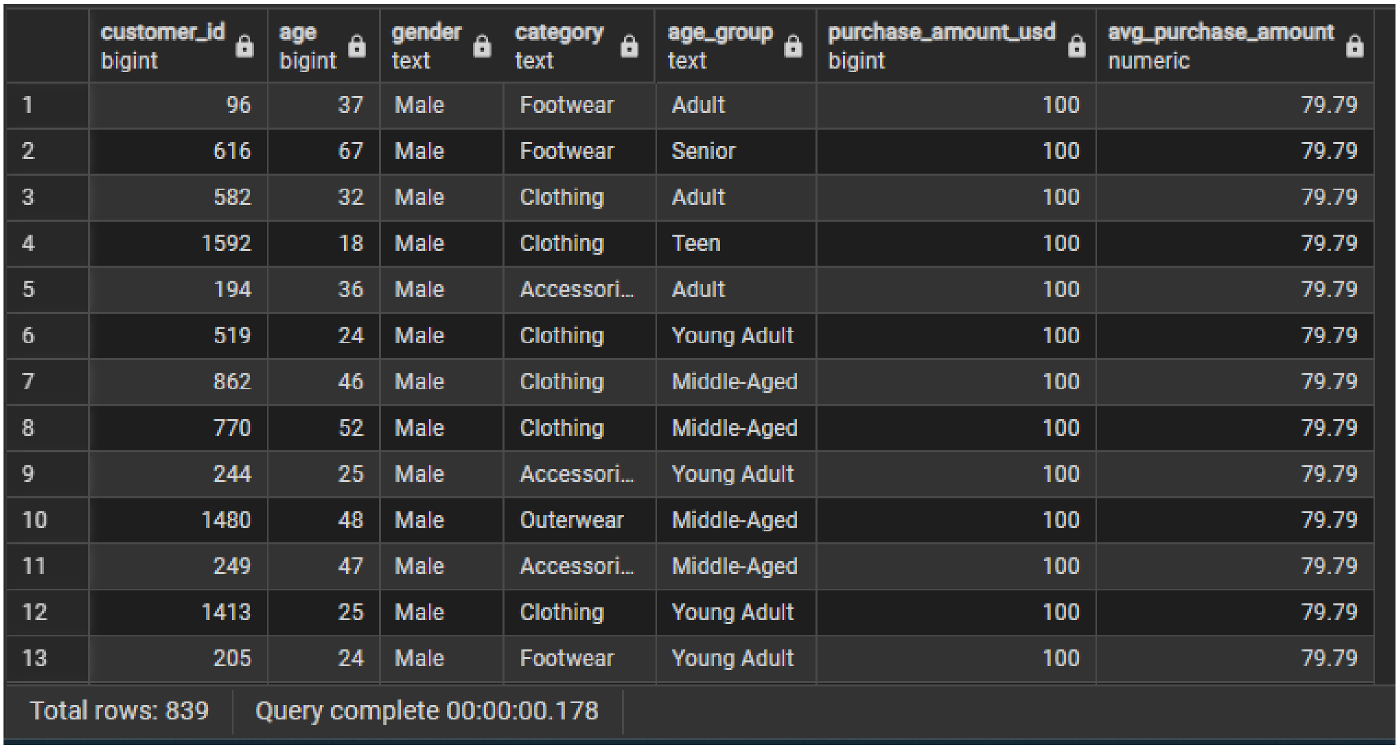
We performed structured analysis in PostgreSQL to answer key business questions:

1. **Revenue by Gender** – Compared total revenue generated by male vs. female customers.



	gender text	revenue numeric
1	Female	75191
2	Male	157890

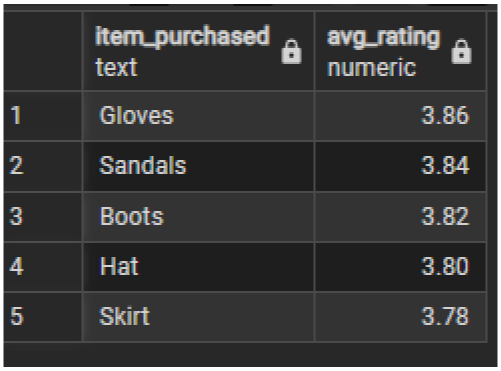
2. **High-Spending Discount Users** – Identified customers who used discounts but still spent above the average purchase amount.



	customer_id bigint	age bigint	gender text	category text	age_group text	purchase_amount_usd bigint	avg_purchase_amount numeric
1	96	37	Male	Footwear	Adult	100	79.79
2	616	67	Male	Footwear	Senior	100	79.79
3	582	32	Male	Clothing	Adult	100	79.79
4	1592	18	Male	Clothing	Teen	100	79.79
5	194	36	Male	Accessori...	Adult	100	79.79
6	519	24	Male	Clothing	Young Adult	100	79.79
7	862	46	Male	Clothing	Middle-Aged	100	79.79
8	770	52	Male	Clothing	Middle-Aged	100	79.79
9	244	25	Male	Accessori...	Young Adult	100	79.79
10	1480	48	Male	Outerwear	Middle-Aged	100	79.79
11	249	47	Male	Accessori...	Middle-Aged	100	79.79
12	1413	25	Male	Clothing	Young Adult	100	79.79
13	205	24	Male	Footwear	Young Adult	100	79.79

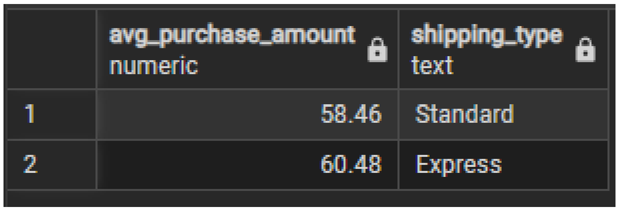
Total rows: 839 Query complete 00:00:00.178

3. **Top 5 Products by Rating** – Found products with the highest average review ratings.



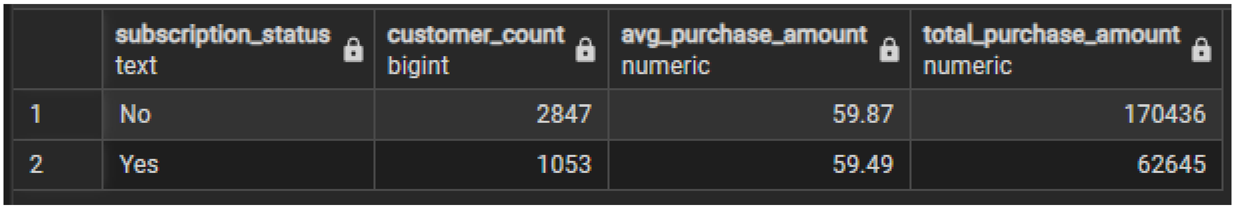
	item_purchased text	avg_rating numeric
1	Gloves	3.86
2	Sandals	3.84
3	Boots	3.82
4	Hat	3.80
5	Skirt	3.78

4. **Shipping Type Comparison** – Compared average purchase amounts between Standard and Express shipping.



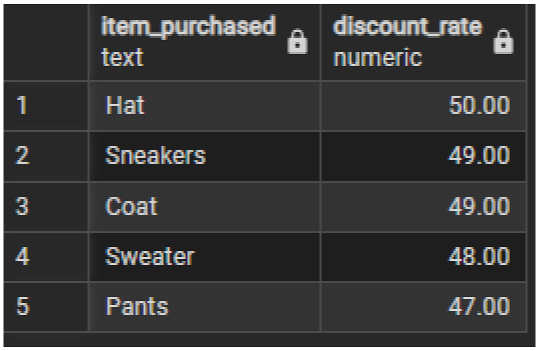
	avg_purchase_amount numeric	shipping_type text
1	58.46	Standard
2	60.48	Express

5. **Subscribers vs. Non-Subscribers** – Compared average spend and total revenue across subscription status.



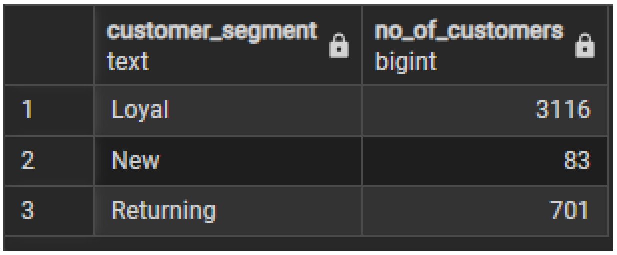
	subscription_status text	customer_count bigint	avg_purchase_amount numeric	total_purchase_amount numeric
1	No	2847	59.87	170436
2	Yes	1053	59.49	62645

6. **Discount-Dependent Products** – Identified 5 products with the highest percentage of discounted purchases.



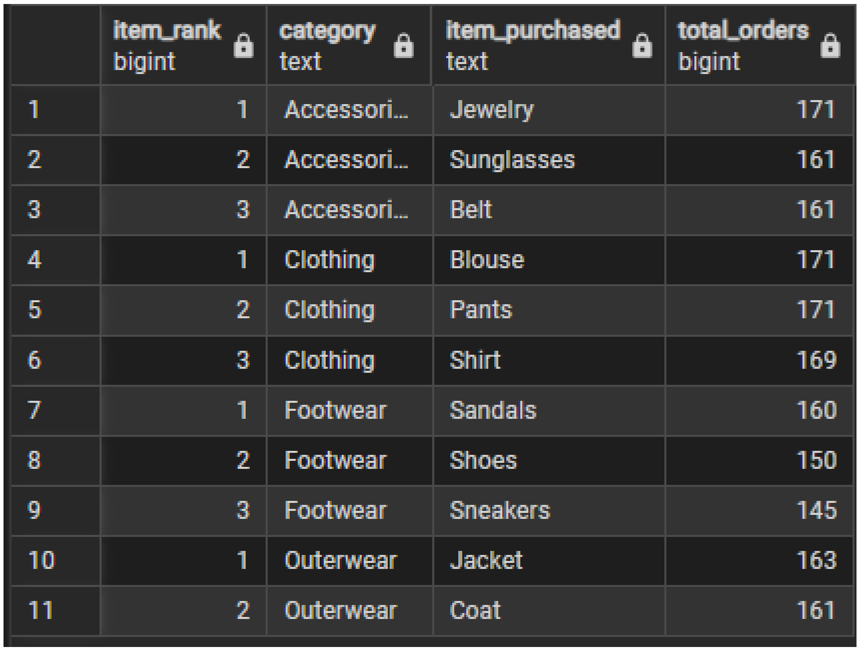
	item_purchased text	discount_rate numeric
1	Hat	50.00
2	Sneakers	49.00
3	Coat	49.00
4	Sweater	48.00
5	Pants	47.00

7. **Customer Segmentation** – Classified customers into New, Returning, and Loyal segments based on purchase history.



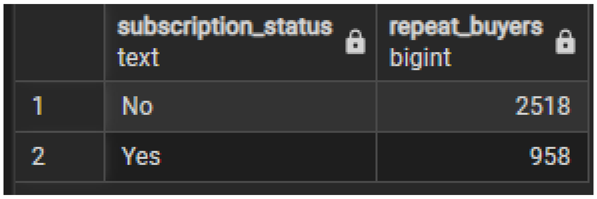
	customer_segment text	no_of_customers bigint
1	Loyal	3116
2	New	83
3	Returning	701

8. **Top 3 Products per Category** – Listed the most purchased products within each category.



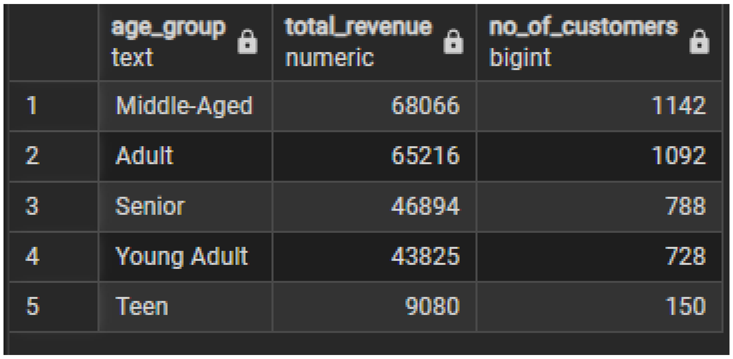
	item_rank	category	item_purchased	total_orders
	bigint	text	text	bigint
1	1	Accessori...	Jewelry	171
2	2	Accessori...	Sunglasses	161
3	3	Accessori...	Belt	161
4	1	Clothing	Blouse	171
5	2	Clothing	Pants	171
6	3	Clothing	Shirt	169
7	1	Footwear	Sandals	160
8	2	Footwear	Shoes	150
9	3	Footwear	Sneakers	145
10	1	Outerwear	Jacket	163
11	2	Outerwear	Coat	161

9. **Repeat Buyers & Subscriptions** – Checked whether customers with >5 purchases are more likely to subscribe.



	subscription_status	repeat_buyers
	text	bigint
1	No	2518
2	Yes	958

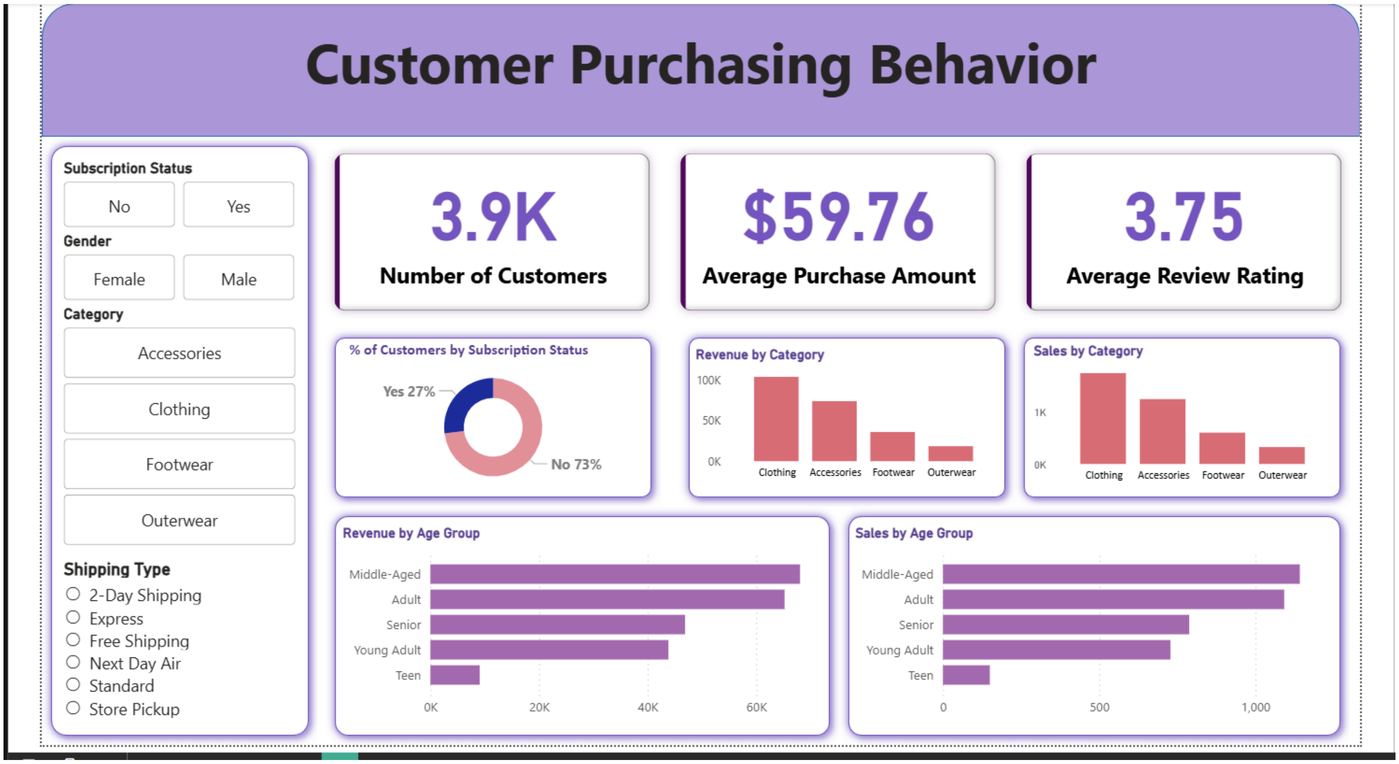
10. **Revenue by Age Group** – Calculated total revenue contribution of each age group.



	age_group	total_revenue	no_of_customers
	text	numeric	bigint
1	Middle-Aged	68066	1142
2	Adult	65216	1092
3	Senior	46894	788
4	Young Adult	43825	728
5	Teen	9080	150

5. Dashboard in Power BI

Finally, we built an interactive dashboard in **Power BI** to present insights visually.



6. Business Recommendations

- **Boost Subscriptions** – Promote exclusive benefits for subscribers.

- **Customer Loyalty Programs** – Reward repeat buyers to move them into the “Loyal” segment.
- **Review Discount Policy** – Balance sales boosts with margin control.
- **Product Positioning** – Highlight top-rated and best-selling products in campaigns.
- **Targeted Marketing** – Focus efforts on high-revenue age groups and express-shipping users.