



# Shopping Trends

SQL

Shinja Dogra,  
Michelle Marants,  
Stefanny Diaz  
Yashpal Singh



# Data Summary

The Customer Shopping Preferences Dataset offers valuable insights into consumer behavior and purchasing patterns. Understanding customer preferences and trends is critical for businesses to tailor their products, marketing strategies, and overall customer experience. This dataset captures a wide range of customer attributes including age, gender, purchase history, preferred payment methods, frequency of purchases, and more. Analyzing this data can help businesses make informed decisions, optimize product offerings, and enhance customer satisfaction.

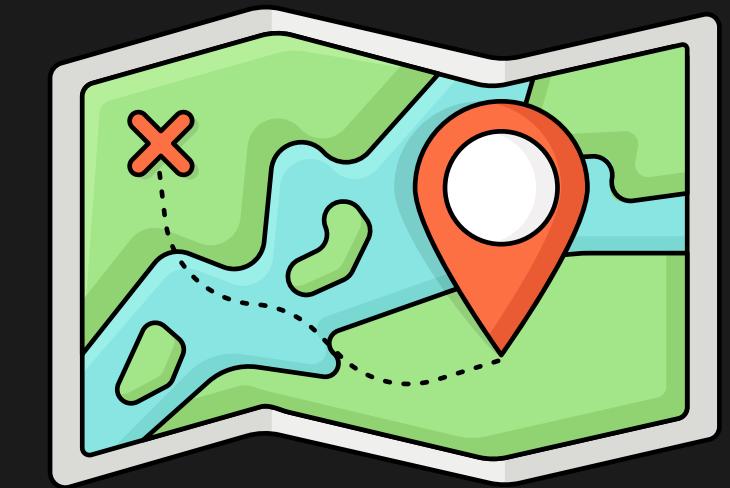
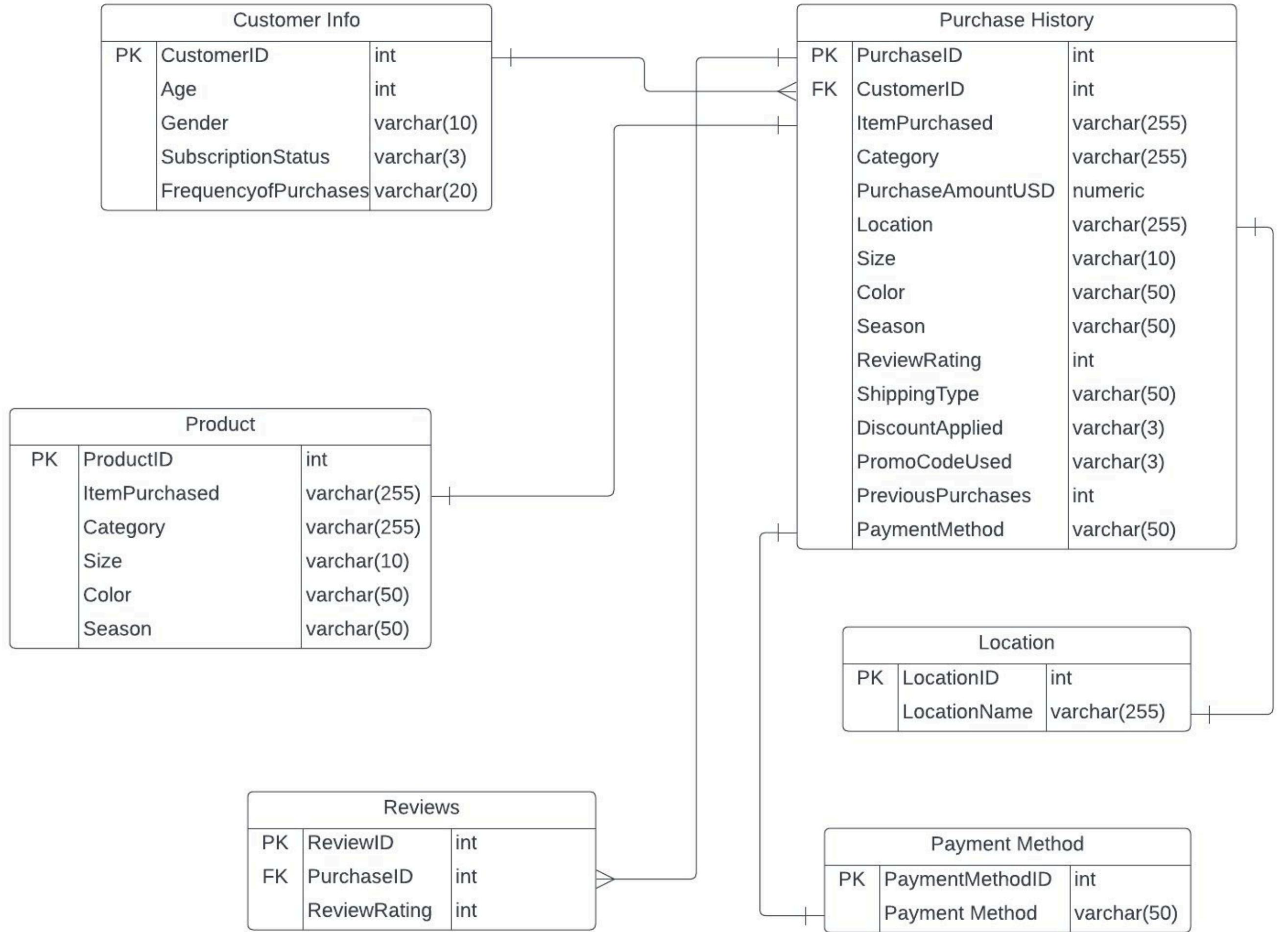


# Data Dictionary (A-Z)



- **Age:** Age of the customer
- **Category:** Category of the item purchased
- **Color:** The color of the purchased item
- **Customer ID:** Unique identifier for each customer
- **Discount Applied:** Indicates if a discount was applied to the purchase (Yes/No)
- **Frequency of Purchases:** Frequency at which the customer makes purchases (e.g., Weekly, Fortnightly, Monthly)
- **Gender:** Gender of the customer (Male/Female)
- **Item Purchased:** The item purchased by the customer
- **Location:** The location where the purchase was made
- **Payment Method:** Customer's most preferred payment method
- **Previous Purchases:** The total count of transactions concluded by the customer at the store, excluding the ongoing transaction
- **Promo Code Used:** Indicates if a promo code was used for the purchase (Yes/No)
- **Purchase Amount (USD):** The amount of the purchase in USD
- **Review Rating:** Rating given by the customer for the purchased item (1-5)
- **Season:** Season during which the purchase was made
- **Shipping Type:** The type of shipping chosen by the customer
- **Size:** Size of the purchased item
- **Subscription Status:** Indicates if the customer has a subscription (Yes/No)





# ERD Diagram



A vibrant photograph of a young woman with long, wavy hair, wearing white sunglasses and a white, ruffled dress. She is captured in a moment of joyful dance, her mouth wide open as if singing or laughing. In her right hand, she holds a small, colorful macaron with purple, yellow, and pink layers. The background is a soft, out-of-focus pink.

# Data Import and View

# Customer Info Table

START

The screenshot shows a database management interface with the following details:

- Schemas:** The current schema is "CustomerShoppingTr...".
- Tables:** Inside this schema, there are tables like CustomerInfo, Location, PaymentMethod, Product, PurchaseHistory, Review, shopping\_trends, Views, Stored Procedures, and Functions.
- Query Editor:** The "Query 1" tab contains the following SQL code:

```
1 • CREATE TABLE CustomerInfo (
2     `Customer ID` INT PRIMARY KEY,
3     Age INT,
4     Gender VARCHAR(10),
5     `Subscription Status` VARCHAR(3),
6     `Frequency of Purchases` VARCHAR(20),
7     UNIQUE KEY (`Customer ID`)
8 );
9
10 • INSERT INTO CustomerInfo (`Customer ID`, Age, Gender, `Subscription Status`, `Frequency of Purchases`)
11 SELECT DISTINCT
12     `Customer ID`,
13     Age,
14     Gender,
15     `Subscription Status`,
16     `Frequency of Purchases`
17 FROM shopping_trends;
18
```
- Toolbar:** Includes standard icons for file operations, database management, and search.
- Status Bar:** Shows "Limit to 1000 rows".

MENU

SIGN IN

# Customer Info Table



Result Grid Filter Rows: Search Edit:

	Customer ID	Age	Gender	Subscription Status	Frequency of Purchases
▶	1	55	Male	Yes	Fortnightly
▶	2	19	Male	Yes	Fortnightly
▶	3	50	Male	Yes	Weekly
▶	4	21	Male	Yes	Weekly
▶	5	45	Male	Yes	Annually
▶	6	46	Male	Yes	Weekly
▶	7	63	Male	Yes	Quarterly
▶	8	27	Male	Yes	Weekly
▶	9	26	Male	Yes	Annually
▶	10	57	Male	Yes	Quarterly
▶	11	53	Male	Yes	Bi-Weekly
▶	12	30	Male	Yes	Fortnightly

SHOP NOW 

# Location Table

Screenshot of a database management tool interface showing the creation of a Location table and its data insertion.

The left pane shows the schema structure:

- SCHEMAS:
  - ap
  - CustomerShoppingTr...
  - Tables
    - CustomerInfo
    - Location
    - PaymentMethod
    - Product
    - PurchaseHistory
    - Review
    - shopping\_trends
  - Views
  - Stored Procedures
  - Functions- sys

The right pane displays the following SQL code:

```
1 • CREATE TABLE Location (
2     LocationID INT AUTO_INCREMENT PRIMARY KEY,
3     LocationName VARCHAR(255) UNIQUE,
4     UNIQUE KEY (LocationID)
5 );
6
7 • INSERT INTO Location (LocationName)
8     SELECT DISTINCT Location
9     FROM shopping_trends;
10
```



# Location Table



100% 1:1

Result Grid Filter Rows: Search Edit:

	LocationID	LocationName
▶	16	Alabama
	30	Alaska
	29	Arizona
	11	Arkansas
	19	California
	25	Colorado
	40	Connecticut
	13	Delaware
	21	Florida
	43	Georgia
	12	Hawaii
	28	Iowa

Location 1

# Payment Method Table

Screenshot of a database management tool interface showing the creation of a Payment Method table.

The left sidebar shows the schema structure:

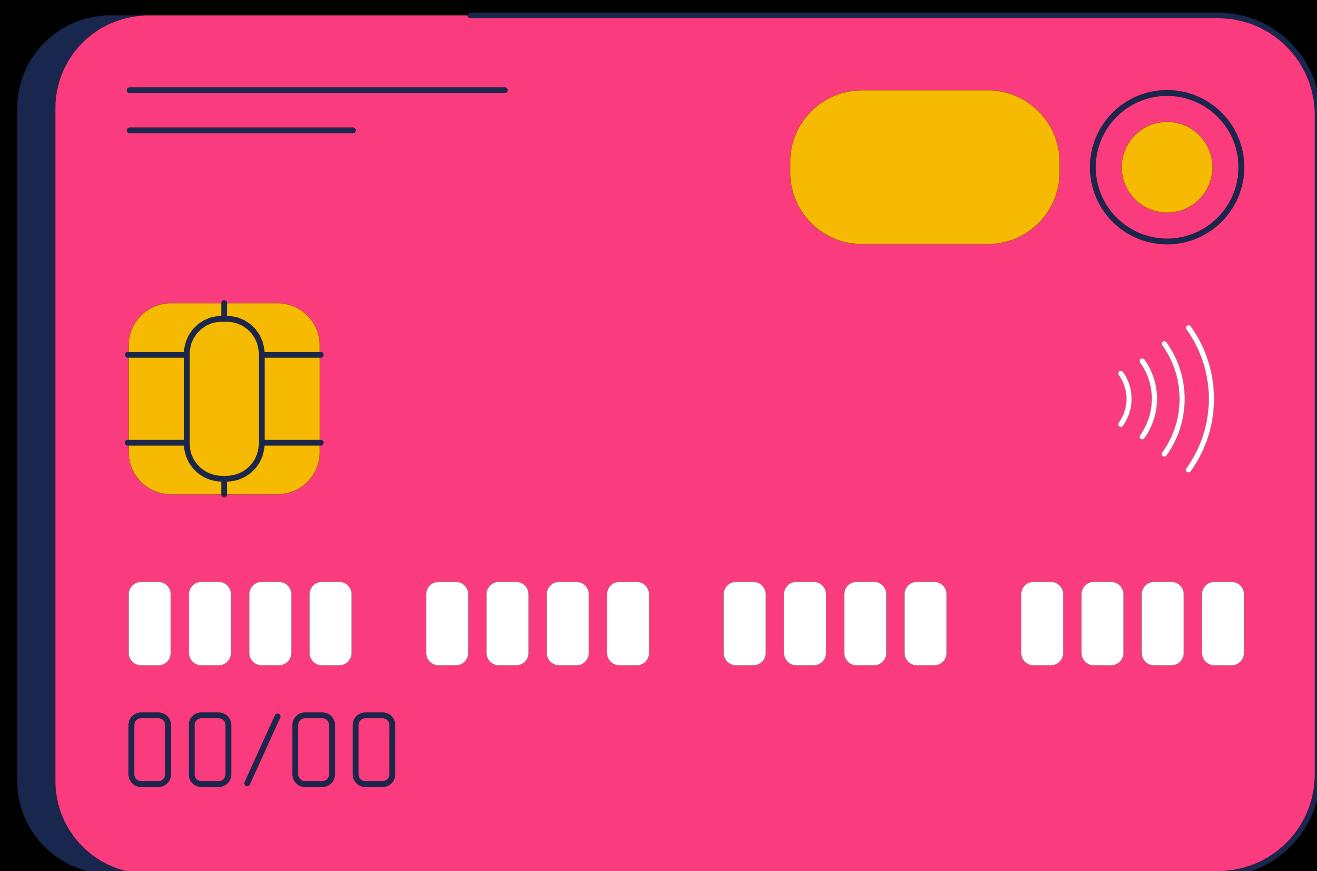
- SCHEMAS
  - ap
  - CustomerShoppingTr...
  - Tables
    - CustomerInfo
    - Location
    - PaymentMethod
    - Product
    - PurchaseHistory
    - Review
    - shopping\_trends
  - Views
  - Stored Procedures
  - Functions- sys

The main area displays the following SQL code:

```
1 * CREATE TABLE PaymentMethod (
2     PaymentMethodID INT AUTO_INCREMENT PRIMARY KEY,
3     `Payment Method` VARCHAR(50) UNIQUE
4 );
5
6 * INSERT INTO PaymentMethod (`Payment Method`)
7     SELECT DISTINCT `Payment Method`
8     FROM shopping_trends;
9
10
11
```



# Payment Method Table



1 | SELECT \* FROM CustomerShoppingTrends.PaymentMethod

100% C 1:1

Result Grid Filter Rows: Search Edit:

PaymentMethodID	Payment Method
2	Bank Transfer
3	Cash
1	Credit Card
6	Debit Card
4	PayPal
5	Venmo
NULL	NULL

PaymentMethodID

# Product Table

Screenshot of a database management tool interface showing the creation of a Product table and an insert query.

**Administration** | **Schemas** | **Query 1**

**SCHEMAS**

Filter objects

- > ap
- < CustomerShoppingTr...
- < Tables
  - > CustomerInfo
  - > Location
  - > PaymentMethod
  - > Product
  - > PurchaseHistory
  - > Review
  - > shopping\_trends
- < Views
- < Stored Procedures
- < Functions
- > sys

Object Info | Session  
No object selected

```
1 • CREATE TABLE Product (
2     ProductID INT AUTO_INCREMENT PRIMARY KEY,
3     `Item Purchased` VARCHAR(255),
4     Category VARCHAR(255),
5     Size VARCHAR(10),
6     Color VARCHAR(50),
7     Season VARCHAR(50),
8     UNIQUE KEY (`Item Purchased`, Category, Size, Color, Season)
9 );
10
11 • INSERT INTO Product(`Item Purchased`, Category, Size, Color, Season)
12 SELECT DISTINCT `Item Purchased`, Category, Size, Color, Season
13 FROM shopping_trends;
14
```



# Product Table



1 ▶ SELECT \* FROM CustomerShoppingTrends.Product;

100% C 1:1

Result Grid Filter Rows: Search Edit:

	ProductID	Item Purchased	Category	Size	Color	Season
▶	676	Backpack	Accessories	L	Beige	Summer
	175	Backpack	Accessories	L	Black	Spring
	415	Backpack	Accessories	L	Black	Winter
	1823	Backpack	Accessories	L	Blue	Spring
	1738	Backpack	Accessories	L	Blue	Summer
	121	Backpack	Accessories	L	Brown	Spring
	390	Backpack	Accessories	L	Cyan	Fall
	1983	Backpack	Accessories	L	Gold	Spring
	1215	Backpack	Accessories	L	Gold	Summer
	2565	Backpack	Accessories	L	Green	Fall
	952	Backpack	Accessories	L	Indigo	Spring
	546	Backpack	Accessories	L	Teal	Fall

Product 1

# Purchase History Table



Administration   Schemas   Query 1

SCHEMAS

Filter objects

> ap  
CustomerShoppingTr...  
Tables  
CustomerInfo  
Location  
PaymentMethod  
Product  
PurchaseHistory  
Review  
shopping\_trends  
Views  
Stored Procedures  
Functions  
> sys

Object Info   Session

No object selected

```
1 CREATE TABLE PurchaseHistory (
2     PurchaseID INT PRIMARY KEY,
3     CustomerID INT,
4     ItemPurchased VARCHAR(255),
5     Category VARCHAR(255),
6     PurchaseAmount(USD) DECIMAL(10, 2),
7     Location VARCHAR(255),
8     Size VARCHAR(10),
9     Color VARCHAR(50),
10    Season VARCHAR(50),
11    ReviewRating INT,
12    ShippingType VARCHAR(50),
13    DiscountApplied VARCHAR(3),
14    PromoCodeUsed VARCHAR(3),
15    PreviousPurchases INT,
16    PaymentMethod VARCHAR(50),
17    FOREIGN KEY (CustomerID) REFERENCES CustomerInfo(CustomerID),
18    INDEX (CustomerID)
19 );
20
21 INSERT INTO PurchaseHistory (PurchaseID, CustomerID, ItemPurchased, Category, PurchaseAmount(USD), Location, Size, Color, Season, ReviewRating, ShippingType, DiscountApplied, PromoCodeUsed, PreviousPurchases, PaymentMethod)
22     SELECT
23         ROW_NUMBER() OVER () AS PurchaseID,
24         CustomerID,
25         ItemPurchased,
26         Category,
27         PurchaseAmount(USD),
28         Location,
29         Size,
30         Color,
31         Season,
32         ReviewRating,
33         ShippingType,
34         DiscountApplied,
35         PromoCodeUsed,
36         PreviousPurchases,
37         PaymentMethod
38     FROM shopping_trends;
39
40
```

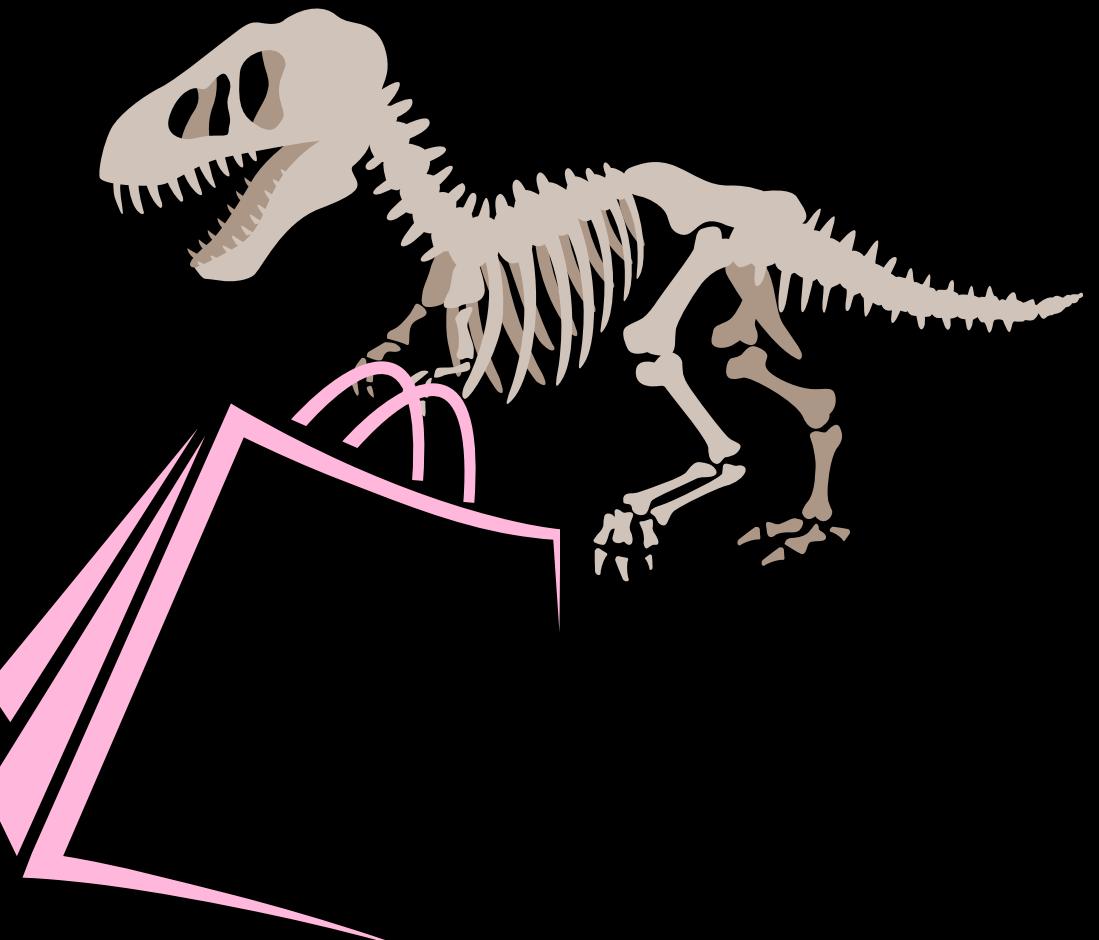
# Purchase History Table

1 • `SELECT * FROM CustomerShoppingTrends.PurchaseHistory;`

100% 1:1

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

PurchaseID	Customer ID	Item Purchased	Category	Purchase Amount (US\$)	Location	Size	Color	Season	Review Rating	Shipping Type	Discount Appli...	Promo Code Used	Previous Purchase ID
1	1	Blouse	Clothing	53.00	Kentucky	L	Gray	Winter	3	Express	Yes	Yes	14
2	2	Sweater	Clothing	64.00	Maine	L	Maroon	Winter	3	Express	Yes	Yes	2
3	3	Jeans	Clothing	73.00	Massachusetts	S	Maroon	Spring	3	Free Shipping	Yes	Yes	23
4	4	Sandals	Footwear	90.00	Rhode Island	M	Maroon	Spring	4	Next Day Air	Yes	Yes	49
5	5	Blouse	Clothing	49.00	Oregon	M	Turquoise	Spring	3	Free Shipping	Yes	Yes	31
6	6	Sneakers	Footwear	20.00	Wyoming	M	White	Summer	3	Standard	Yes	Yes	14
7	7	Shirt	Clothing	85.00	Montana	M	Gray	Fall	3	Free Shipping	Yes	Yes	49
8	8	Shorts	Clothing	34.00	Louisiana	L	Charcoal	Winter	3	Free Shipping	Yes	Yes	19
9	9	Coat	Outerwear	97.00	West Virginia	L	Silver	Summer	3	Express	Yes	Yes	8
10	10	Handbag	Accessories	31.00	Missouri	M	Pink	Spring	5	2-Day Shipping	Yes	Yes	4
11	11	Shoes	Footwear	34.00	Arkansas	L	Purple	Fall	4	Store Pickup	Yes	Yes	26
12	12	Shorts	Clothing	68.00	Hawaii	S	Olive	Winter	5	Store Pickup	Yes	Yes	10
13	13	Coat	Outerwear	72.00	Delaware	M	Gold	Winter	4	Express	Yes	Yes	37
14	14	Dress	Clothing	51.00	New Hampshire	M	Violet	Spring	5	Express	Yes	Yes	31
15	15	Coat	Outerwear	53.00	New York	L	Teal	Winter	5	Free Shipping	Yes	Yes	34
16	16	Skirt	Clothing	81.00	Rhode Island	M	Teal	Winter	3	Store Pickup	Yes	Yes	8
17	17	Sunglasses	Accessories	36.00	Alabama	S	Gray	Spring	4	Next Day Air	Yes	Yes	44
18	18	Dress	Clothing	38.00	Mississippi	XL	Lavender	Winter	5	2-Day Shipping	Yes	Yes	36
19	19	Sweater	Clothing	48.00	Montana	S	Black	Summer	5	Free Shipping	Yes	Yes	17



# Review Table

Schemas

SCHEMAS

Filter objects

ap

CustomerShoppingTr...

Tables

CustomerInfo

Location

PaymentMethod

Product

PurchaseHistory

Review

shopping\_trends

Views

Stored Procedures

Functions

sys

Object Info Session

No object selected

Query 1

CREATE TABLE Review (

    ReviewID INT AUTO\_INCREMENT PRIMARY KEY,

    PurchaseID INT,

    `Review Rating` INT,

    FOREIGN KEY (PurchaseID) REFERENCES PurchaseHistory(PurchaseID),

    INDEX (PurchaseID)

);

INSERT INTO Review (PurchaseID, `Review Rating`)

SELECT

    PurchaseID,

    `Review Rating`

FROM PurchaseHistory;



# Review Table

1 • SELECT \* FROM CustomerShoppingTrends.Review;

100% 1:1

Result Grid Filter Rows: Search Edit: Export

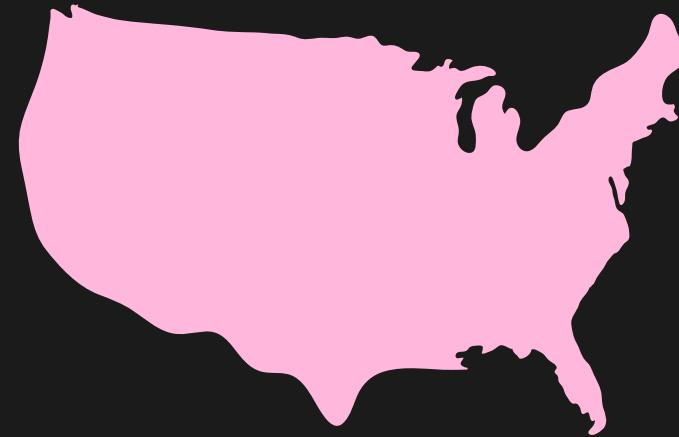
	ReviewID	PurchaseID	Review	Rating
▶	1	1	3	
	2	2	3	
	3	3	3	
	4	4	4	
	5	5	3	
	6	6	3	
	7	7	3	
	8	8	3	
	9	9	3	
	10	10	5	
	11	11	4	

Review 1



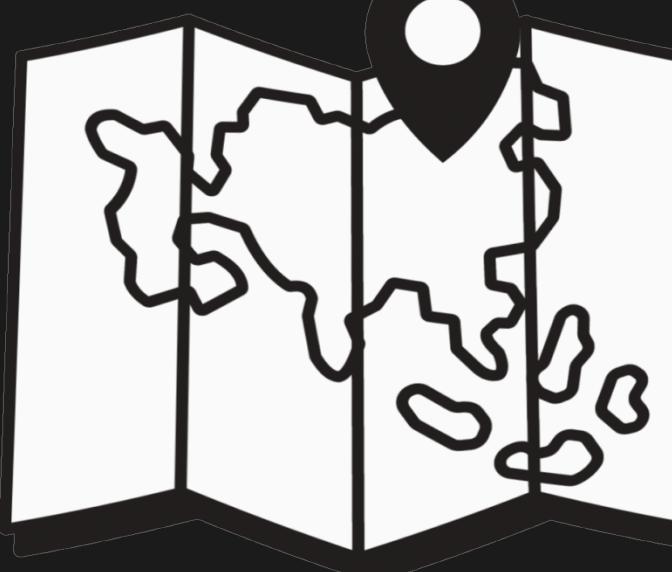
# Roll up: Questions and Answers





# Question 1:

## Who are the Top Spenders?



Retrieve the top 5 customers who spent the most, including their location and total spending.

```
1 • SELECT `CustomerInfo`.`Customer ID`, `PurchaseHistory`.`Location`, SUM(`PurchaseHistory`.`Purchase Amount (USD)`) AS `TotalSpending`
2   FROM `CustomerInfo`
3     JOIN `PurchaseHistory` ON `CustomerInfo`.`Customer ID` = `PurchaseHistory`.`Customer ID`
4   GROUP BY `CustomerInfo`.`Customer ID`, `PurchaseHistory`.`Location`
5   ORDER BY `TotalSpending` DESC
6   LIMIT 5;
```

1:7

Result Grid   Filter Rows:  Search   Export:  Fetch rows:

Customer ID	Location	TotalSpendi...
205	Arizona	100.00
43	Tennessee	100.00
96	Missouri	100.00
194	North Dakota	100.00
244	Kentucky	100.00

Result Grid

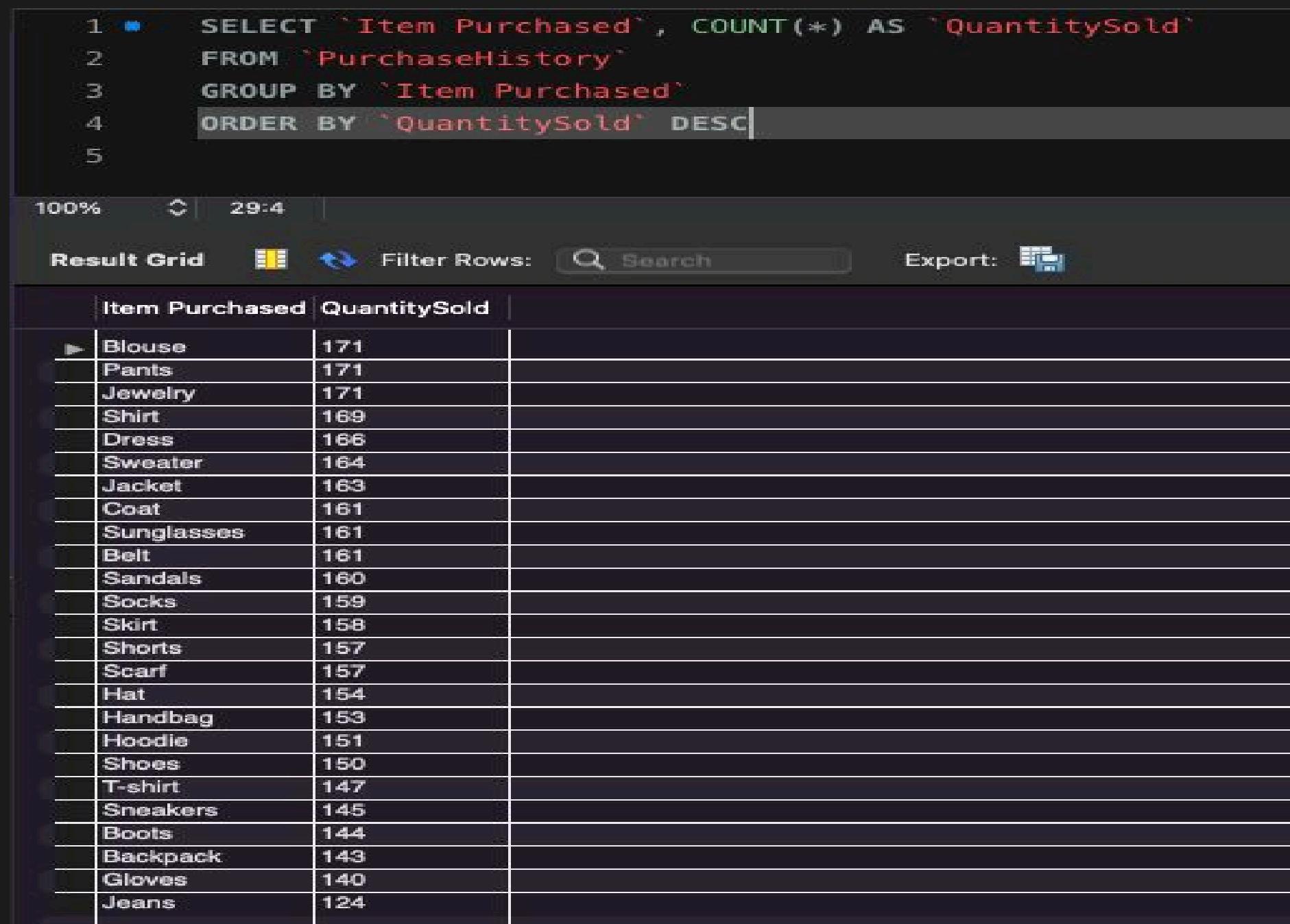
Form Editor

Find

# Question 2:

## What are the most popular and unpopular products?

Find the top 3 most and least purchased products along with the total quantity



The screenshot shows a MySQL command-line interface with a query window and a results window. The query is:

```
1 •  SELECT `Item Purchased`, COUNT(*) AS `QuantitySold`
2   FROM `PurchaseHistory`
3   GROUP BY `Item Purchased`
4   ORDER BY `QuantitySold` DESC
```

The results grid displays the following data:

Item Purchased	QuantitySold
Blouse	171
Pants	171
Jewelry	171
Shirt	169
Dress	166
Sweater	164
Jacket	163
Coat	161
Sunglasses	161
Belt	161
Sandals	160
Socks	159
Skirt	158
Shorts	157
Scarf	157
Hat	154
Handbag	153
Hoodie	151
Shoes	150
T-shirt	147
Sneakers	145
Boots	144
Backpack	143
Gloves	140
Jeans	124

- 01 Blouse
- 02 Pants
- 03 Jewelry

- 01 Jeans
02. Gloves
- 03 Backpack

# Question 3:

## Analyze the Review ratings



Calculate the average review rating for each product category.

\*Note\* : The reviews range from 1-5 with 5 being the highest.

```
1 • SELECT `Product`.`Category` , AVG(`PurchaseHistory`.`Review Rating`) AS 'AvgReviewRating'  
2   FROM `PurchaseHistory`  
3   JOIN `Product` ON `PurchaseHistory`.`Item Purchased` = `Product`.`Item Purchased`  
4   GROUP BY `Product`.`Category`;  
5
```

100% 1:5

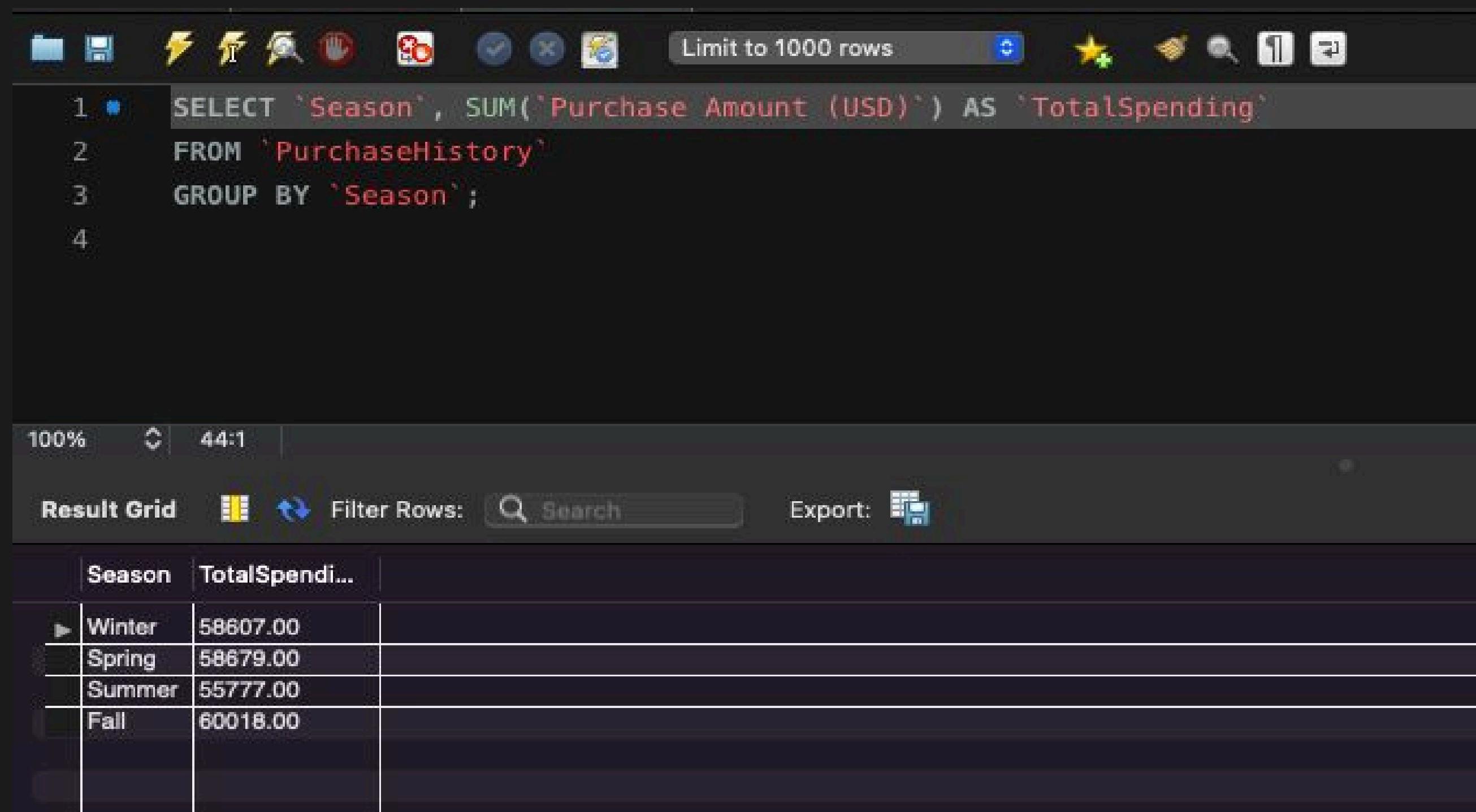
Result Grid Filter Rows: Search Export:

Category	AvgReviewRating
Clothing	3.7576
Footwear	3.8253
Outerwear	3.7750
Accessories	3.7921

# Question 4:

## Are there any Seasonal Trends?

Determine the total spending for each season



The screenshot shows a database interface with a dark theme. At the top, there is a toolbar with various icons. Below the toolbar, a SQL query is displayed in a code editor:

```
1 •   SELECT `Season`, SUM(`Purchase Amount (USD)`) AS `TotalSpending`
2     FROM `PurchaseHistory`
3   GROUP BY `Season`;
4
```

The interface includes a "Limit to 1000 rows" button and several other buttons for managing the query. Below the code editor, there are some status indicators: "100%" and "44:1". At the bottom, there are buttons for "Result Grid", "Filter Rows:", "Search", and "Export".

Season	TotalSpend...
Winter	58607.00
Spring	58679.00
Summer	55777.00
Fall	60018.00

# Thank You!

---

HAPPY  
*holidays*

