



PSYLIQ

Task 2: Paytm Mall E-purchase Data Analysis



SQL Internship at Psyliq

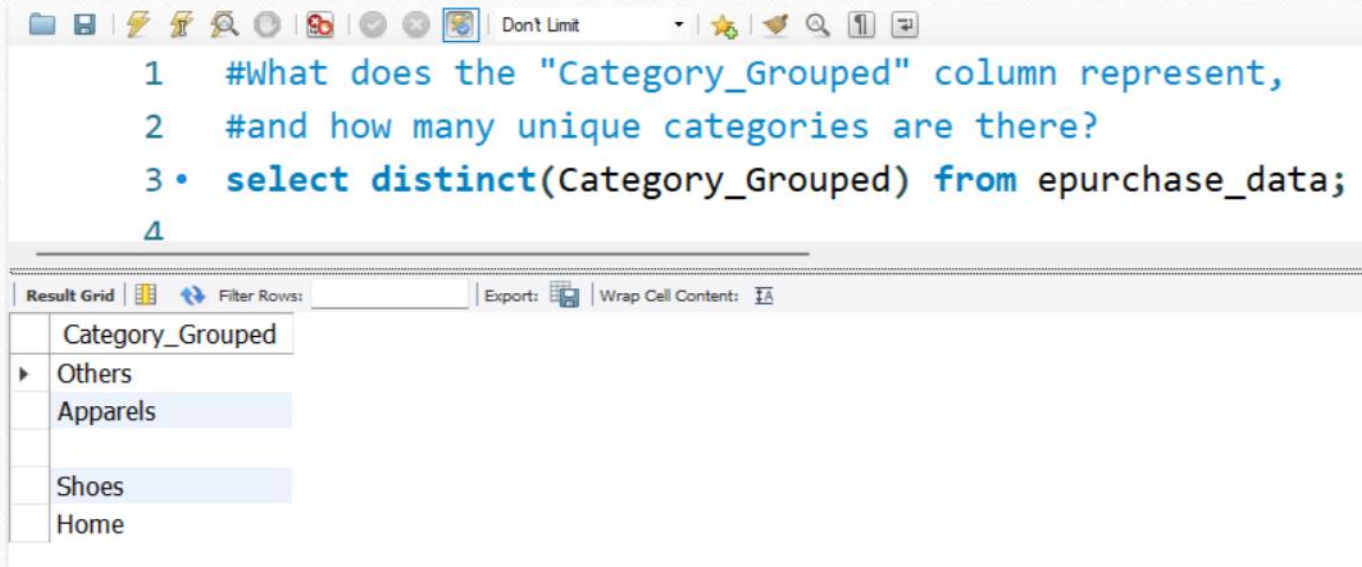
Prepared By: Utsav Raj

LinkedIn: [linkedin.com/in/iamutsavraj](https://www.linkedin.com/in/iamutsavraj)



1. What does the "Category_Grouped" column represent, and how many unique categories are there?

➡ `select distinct(Category_Grouped) from epurchase_data;`
There are 4 unique categories: Home, Shoes, Apparels and Others.



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

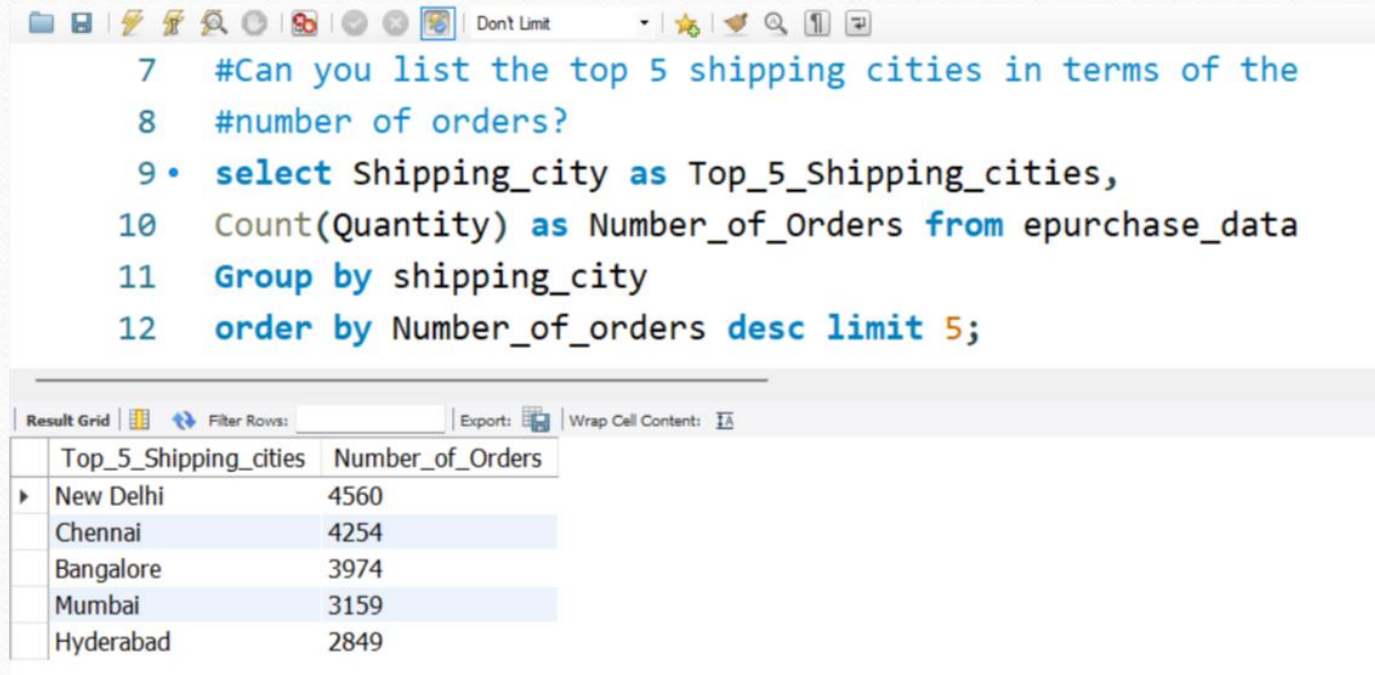
```
1  #What does the "Category_Grouped" column represent,  
2  #and how many unique categories are there?  
3 • select distinct(Category_Grouped) from epurchase_data;  
4
```

Below the query editor, the 'Result Grid' is displayed. It shows a single column titled 'Category_Grouped' with four rows of data: 'Others', 'Apparels', 'Shoes', and 'Home'. The 'Others' row is expanded, indicated by a small triangle icon to its left.

| Category_Grouped |
|------------------|
| Others |
| Apparels |
| Shoes |
| Home |

2. Can you list the top 5 shipping cities in terms of the number of orders?

➡ select Shipping_city as Top_5_Shipping_cities, Count(Quantity) as Number_of_Orders from epurchase_data Group by shipping_city order by Number_of_orders desc limit 5;



The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

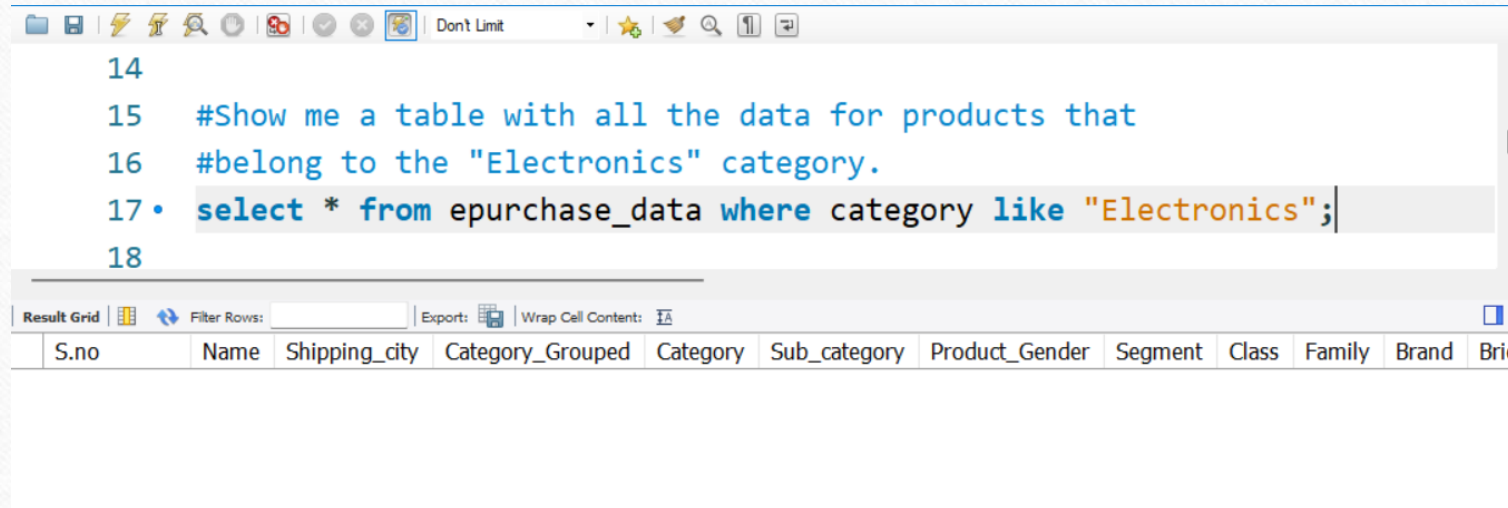
```
7 #Can you list the top 5 shipping cities in terms of the
8 #number of orders?
9 • select Shipping_city as Top_5_Shipping_cities,
10 Count(Quantity) as Number_of_Orders from epurchase_data
11 Group by shipping_city
12 order by Number_of_orders desc limit 5;
```

Below the query editor, the results are displayed in a table with the following columns: Top_5_Shipping_cities and Number_of_Orders.

| Top_5_Shipping_cities | Number_of_Orders |
|-----------------------|------------------|
| New Delhi | 4560 |
| Chennai | 4254 |
| Bangalore | 3974 |
| Mumbai | 3159 |
| Hyderabad | 2849 |

3. Show me a table with all the data for products that belong to the "Electronics" category.

➡ `select * from epurchase_data where category like "Electronics";`



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

```
14  
15 #Show me a table with all the data for products that  
16 #belong to the "Electronics" category.  
17 • select * from epurchase_data where category like "Electronics";  
18
```

Below the query editor is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid table has the following columns:

| S.no | Name | Shipping_city | Category_Grouped | Category | Sub_category | Product_Gender | Segment | Class | Family | Brand | Brick |
|------|------|---------------|------------------|----------|--------------|----------------|---------|-------|--------|-------|-------|
|------|------|---------------|------------------|----------|--------------|----------------|---------|-------|--------|-------|-------|

4. Filter the data to show only rows with a "Sale_Flag" of 'Yes'.

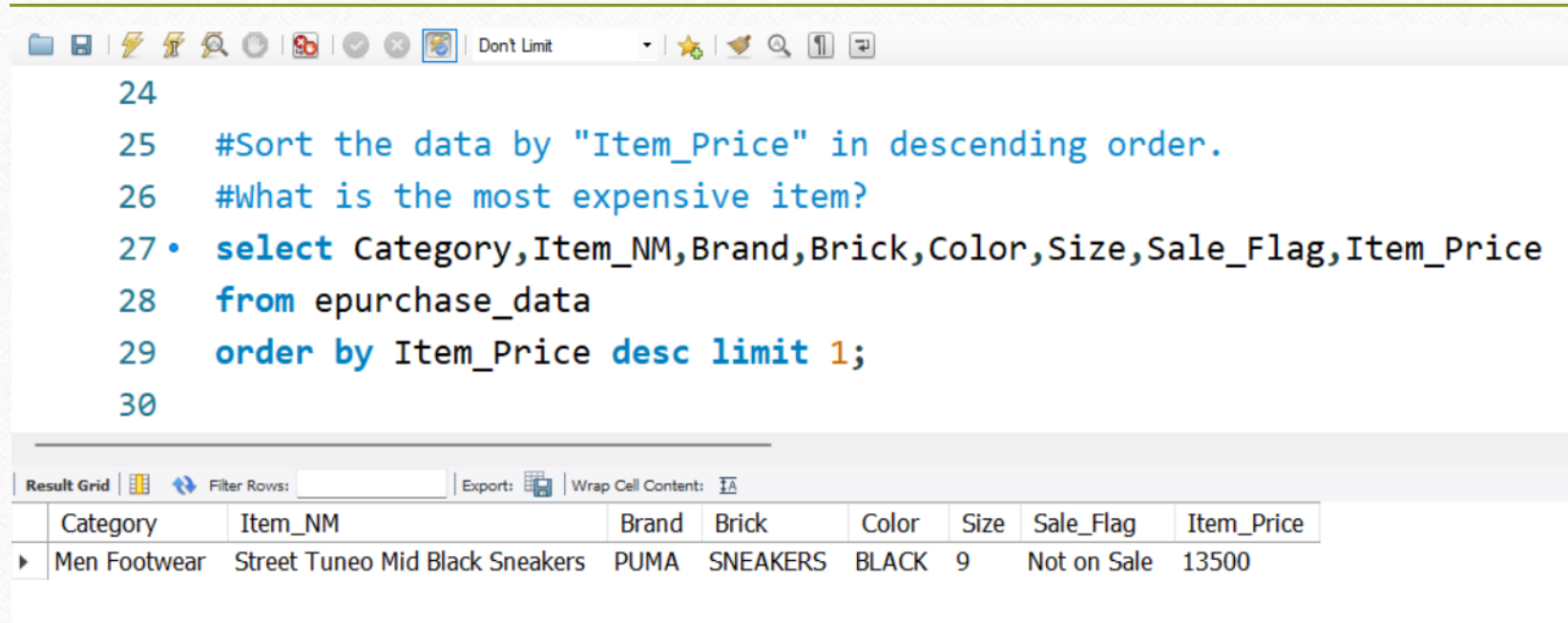
→ select * from epurchase_data where sale_flag like "on sale";

The screenshot shows a data analysis interface with a SQL query editor and a result grid. The query is: `select * from epurchase_data where sale_flag like "on sale";`. The result grid displays 13 rows of data with columns: S.no, Name, Shipping_city, Category_Grouped, Category, Sub_category, Product_Gender, and Segment. The data is as follows:

| S.no | Name | Shipping_city | Category_Grouped | Category | Sub_category | Product_Gender | Segment |
|------|-----------------|---------------|------------------|------------------|----------------|----------------|-------------|
| 2 | AMIT GALPHADE | Ahmedabad | Apparels | Sports Equipment | Sports Apparel | MEN | MENS WEAR |
| 4 | MALLIKARJUNA H | Bangalore | Apparels | Sports Equipment | Sports Apparel | MEN | MENS WEAR |
| 10 | ASHWIN GIDWANI | Pune | Apparels | Sports Equipment | Sports Apparel | MEN | MENS WEAR |
| 16 | Rompelli GopalK | Salem | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTW |
| 20 | prabhakar reddy | Jhansi | | WATCHES | WATCHES | MEN | WOMENS ACC |
| 22 | RAHUL SINGH PAT | Jabalpur | Others | Bags | Bags | WOMEN | WOMEN |
| 23 | NAGA KISHORE | Bangalore | Apparels | Sports Equipment | Sports Apparel | MEN | MENS WEAR |
| 24 | kamla singh | Lucknow | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTW |
| 26 | ankit patni | Indore | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTW |
| 27 | pc marwah | New Delhi | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTW |
| 28 | Ram Prasath | Coimbatore | Shoes | Women Footwear | Womens Foot... | WOMEN | LADIES FOOT |
| 29 | SATHIVA NARAYAN | Chennai | Apparels | Sports Equipment | Sports Apparel | MEN | MENS WEAR |

5. Sort the data by "Item_Price" in descending order. What is the most expensive item?

➡ select Category, Item_NM, Brand, Brick,Color ,Size, Sale_Flag, Item_Price from epurchase_data order by Item_Price desc limit 1;



The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

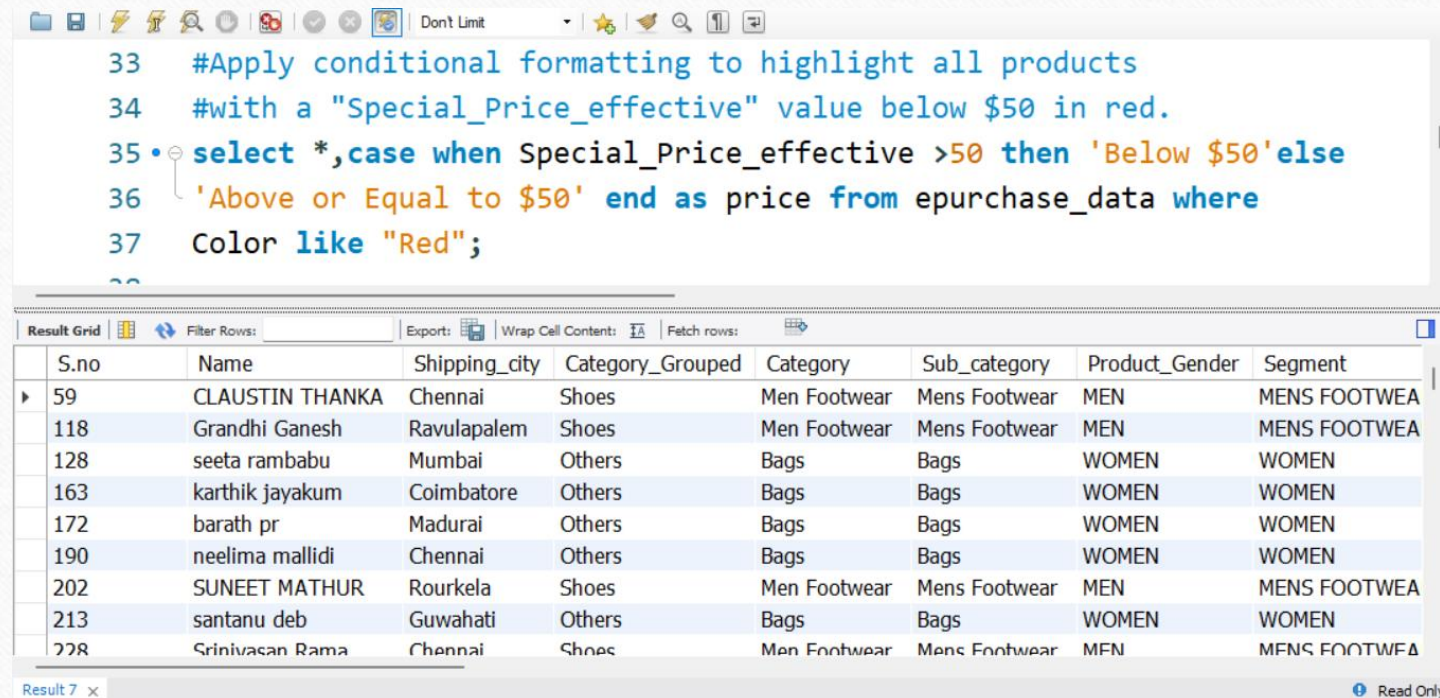
```
24
25 #Sort the data by "Item_Price" in descending order.
26 #What is the most expensive item?
27 • select Category,Item_NM,Brand,Brick,Color,Size,Sale_Flag,Item_Price
28 from epurchase_data
29 order by Item_Price desc limit 1;
30
```

Below the query editor, the results are displayed in a table format. The table has a toolbar with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The table contains one row of data:

| Category | Item_NM | Brand | Brick | Color | Size | Sale_Flag | Item_Price |
|--------------|---------------------------------|-------|----------|-------|------|-------------|------------|
| Men Footwear | Street Tuneo Mid Black Sneakers | PUMA | SNEAKERS | BLACK | 9 | Not on Sale | 13500 |

6. Apply conditional formatting to highlight all products with a "Special_Price_effective" value below \$50 in red.

select *, case when Special_Price_effective >50 then 'Below \$50' else 'Above or Equal to \$50' end as price from epurchase_data where Color like "Red";



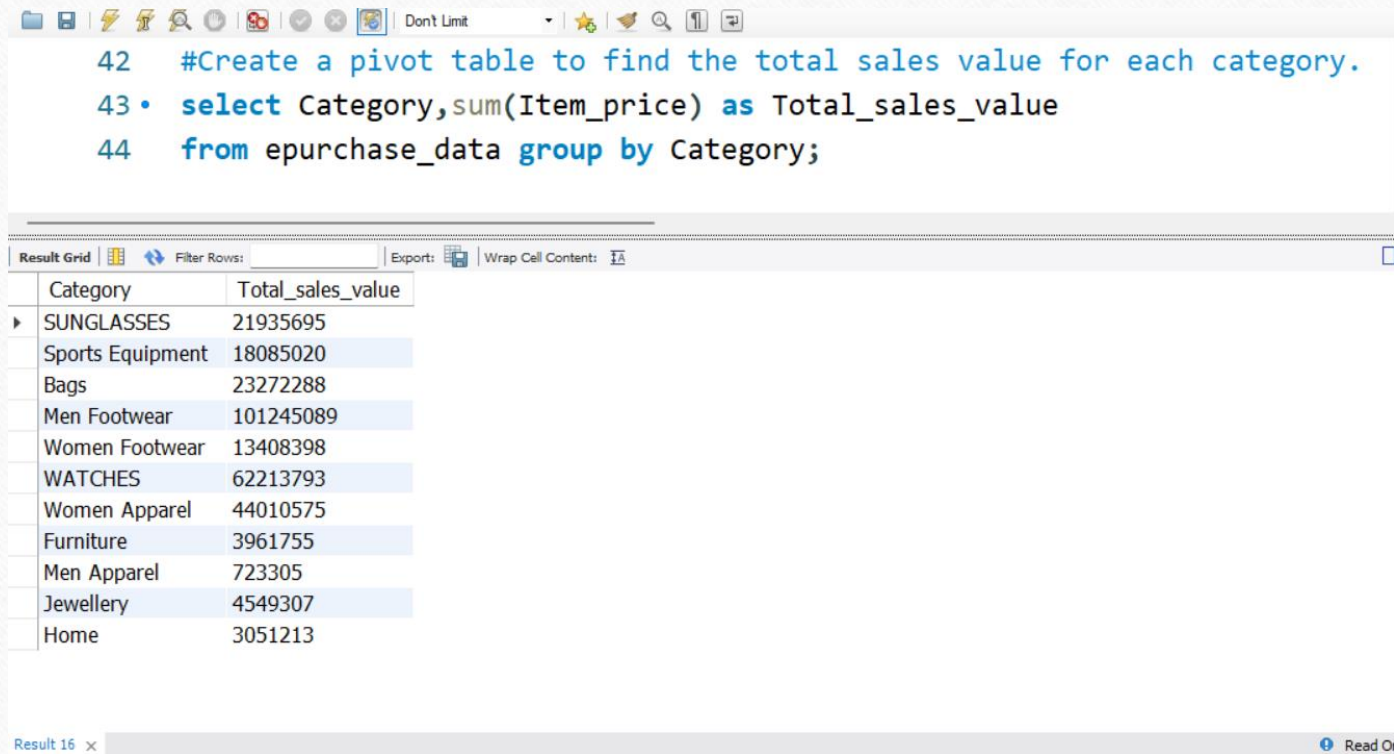
The screenshot shows a SQL IDE interface. The top pane contains a SQL query with line numbers 33 to 37. The query is:
33 #Apply conditional formatting to highlight all products
34 #with a "Special_Price_effective" value below \$50 in red.
35 select *, case when Special_Price_effective >50 then 'Below \$50' else
36 'Above or Equal to \$50' end as price from epurchase_data where
37 Color like "Red";
The bottom pane shows the 'Result Grid' with 8 columns: S.no, Name, Shipping_city, Category_Grouped, Category, Sub_category, Product_Gender, and Segment. It displays 9 rows of data. The status bar at the bottom indicates 'Result 7' and 'Read Only'.

```
33 #Apply conditional formatting to highlight all products
34 #with a "Special_Price_effective" value below $50 in red.
35 select *, case when Special_Price_effective >50 then 'Below $50' else
36 'Above or Equal to $50' end as price from epurchase_data where
37 Color like "Red";
```

| S.no | Name | Shipping_city | Category_Grouped | Category | Sub_category | Product_Gender | Segment |
|------|-----------------|---------------|------------------|--------------|---------------|----------------|--------------|
| 59 | CLAUSTIN THANKA | Chennai | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTWEA |
| 118 | Grandhi Ganesh | Ravulapalem | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTWEA |
| 128 | seeta rambabu | Mumbai | Others | Bags | Bags | WOMEN | WOMEN |
| 163 | karthik jayakum | Coimbatore | Others | Bags | Bags | WOMEN | WOMEN |
| 172 | barath pr | Madurai | Others | Bags | Bags | WOMEN | WOMEN |
| 190 | neelima mallidi | Chennai | Others | Bags | Bags | WOMEN | WOMEN |
| 202 | SUNEET MATHUR | Rourkela | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTWEA |
| 213 | santanu deb | Guwahati | Others | Bags | Bags | WOMEN | WOMEN |
| 228 | Srinivasan Rama | Chennai | Shoes | Men Footwear | Mens Footwear | MEN | MENS FOOTWEA |

7. Create a pivot table to find the total sales value for each category.

➡ select Category, sum(Item_price) as Total_sales_value from
epurchase_data group by Category;



The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

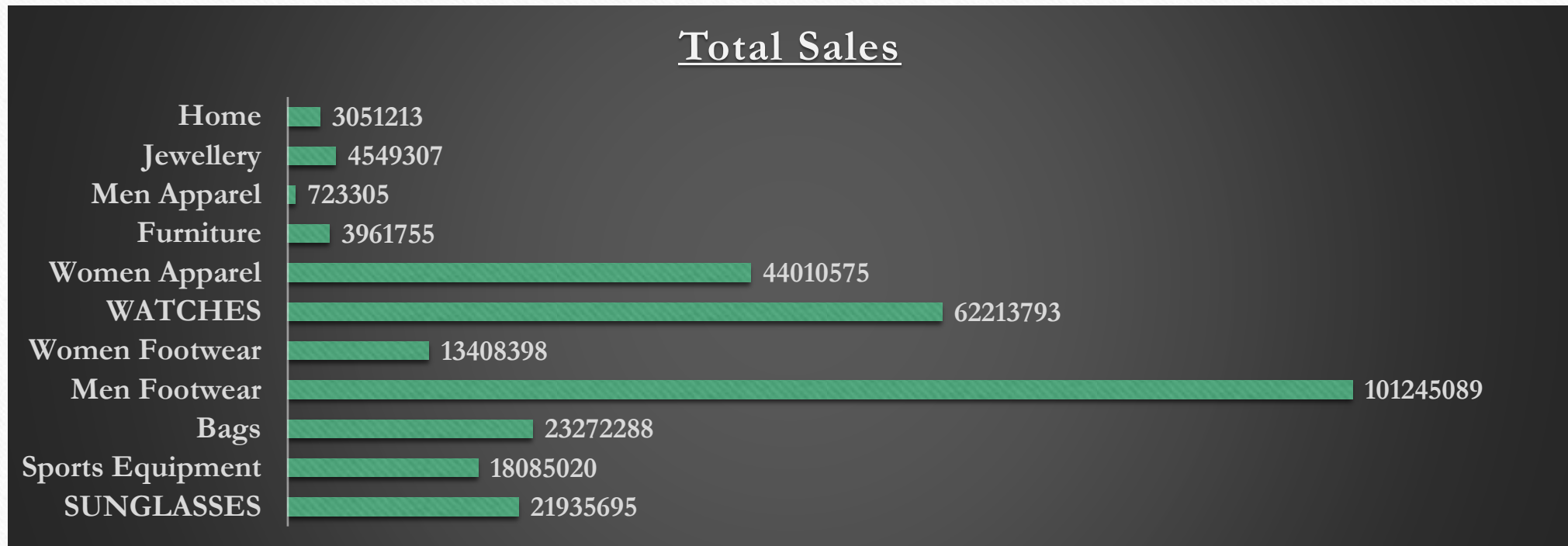
```
42 #Create a pivot table to find the total sales value for each category.  
43 • select Category, sum(Item_price) as Total_sales_value  
44 from epurchase_data group by Category;
```

The results pane displays a table with two columns: Category and Total_sales_value. The table contains 12 rows of data. The first row is expanded, showing a sub-table for SUNGLASSES.

| Category | Total_sales_value |
|------------------|-------------------|
| SUNGLASSES | 21935695 |
| Sports Equipment | 18085020 |
| Bags | 23272288 |
| Men Footwear | 101245089 |
| Women Footwear | 13408398 |
| WATCHES | 62213793 |
| Women Apparel | 44010575 |
| Furniture | 3961755 |
| Men Apparel | 723305 |
| Jewellery | 4549307 |
| Home | 3051213 |

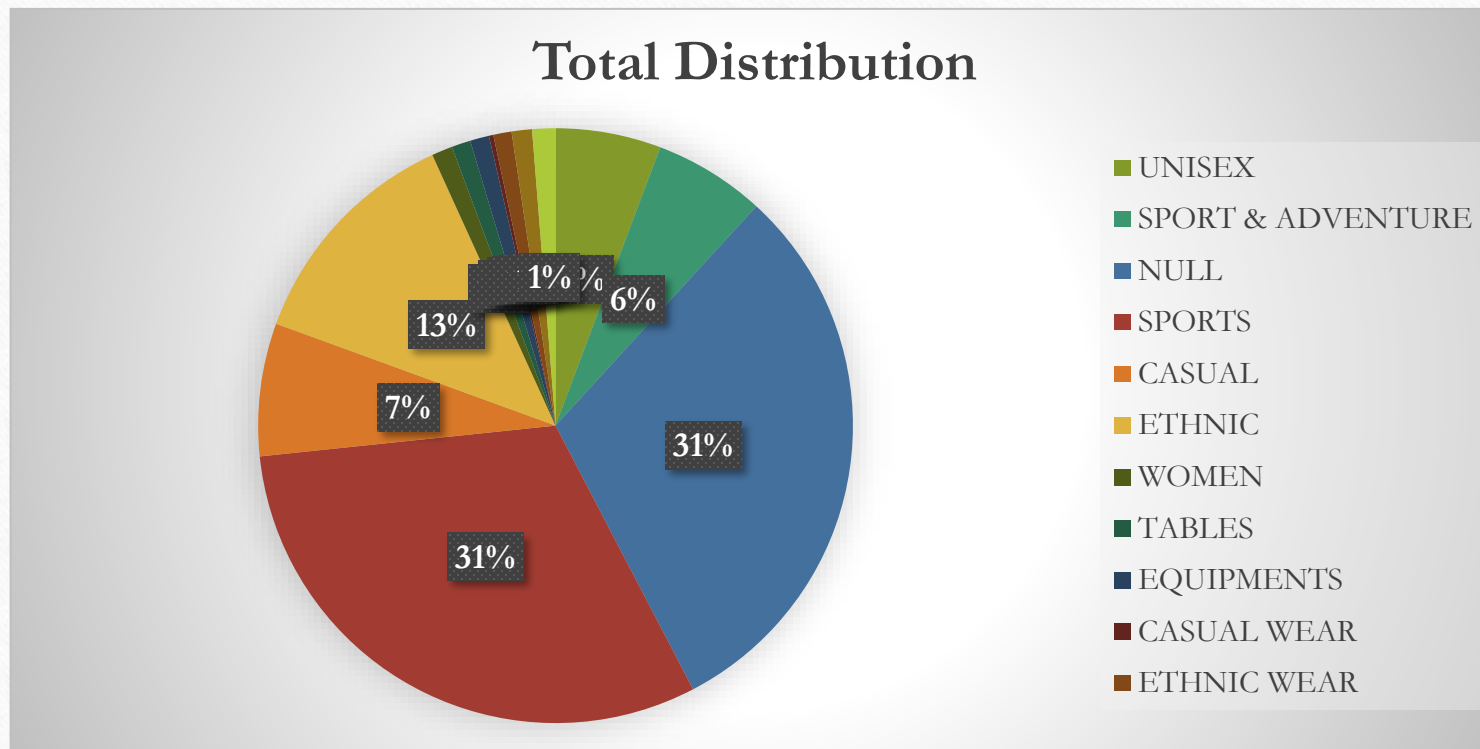
Result 16 x Read Or

8. Create a bar chart to visualize the total sales for each category.



9. Create a pie chart to show the distribution of products in the "Family" category.

→ select Family,count(*) as Total_distribution from epurchase_data group by Family;



10. Ensure that the "Payment_Method" column only contains valid payment methods (e.g., Visa, MasterCard).

➡ select * from epurchase_data where Payment_Method in ("Cod","Prepaid","visa","MasterCard");

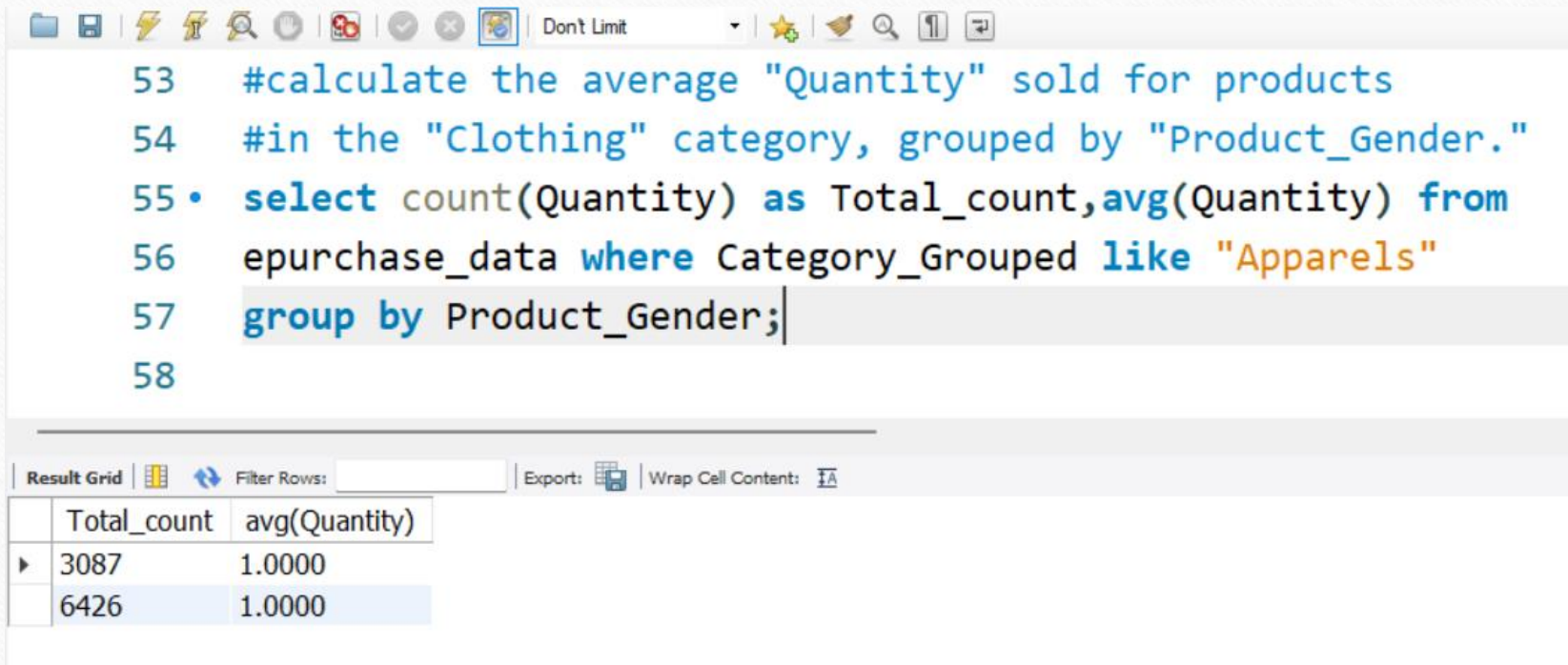
```
47 #Ensure that the "Payment_Method" column only contains valid payment
48 #methods (e.g., Visa, MasterCard).
49 • select * from epurchase_data where Payment_Method
50 in ("Cod","Prepaid","visa","MasterCard");
51
```

| S.no | Name | Shipping_city | Category_Grouped | Category | Sub_category | Product_Gender | Segmen |
|------|-----------------|---------------|------------------|------------------|----------------|----------------|---------|
| 1 | ABHINAV CHATTER | Jabalpur | Others | SUNGLASSES | SUNGLASSES | UNISEX | SUNGLA |
| 2 | AMIT GALPHADE | Ahmedabad | Apparels | Sports Equipment | Sports Apparel | MEN | MENS W |
| 3 | PRABHU NAMBIAPP | Chennai | Others | Bags | Bags | UNISEX | UNISEX |
| 4 | MALLIKARJUNA H | Bangalore | Apparels | Sports Equipment | Sports Apparel | MEN | MENS W |
| 5 | ANUPAM UPADHYAY | Gurgaon | | Men Footwear | Mens Footwear | MEN | MENS FC |
| 6 | SITAL DE | Aalo | Shoes | Men Footwear | Mens Footwear | MEN | MENS FC |
| 7 | Abdul Qadir Sha | Kalyan | | Men Footwear | MENS FOOTWEAR | MEN | MENS FC |
| 8 | kunal lavekar | Pune | Shoes | Men Footwear | Mens Footwear | MEN | MENS FC |
| 9 | Hardeep Mohan | Bangalore | Shoes | Men Footwear | Mens Footwear | MEN | MENS FC |
| 10 | ASHWIN GIDWANI | Pune | Apparels | Sports Equipment | Sports Apparel | MEN | MENS W |
| 11 | R RAMAKRISHNAN | Pune | | Men Footwear | Mens Footwear | MEN | MENS FC |
| 12 | 1AYAKTSHOR RACH | Hyderahad | | Men Footwear | MENS FOOTWFAR | MFN | MFNS FC |

epurchase_data 18 x Read Only

11. Calculate the average "Quantity" sold for products in the "Clothing" category, grouped by "Product_Gender."

```
select count(Quantity) as Total_count, avg(Quantity) from epurchase_data  
where Category_Grouped like "Apparels" group by Product_Gender;
```



The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

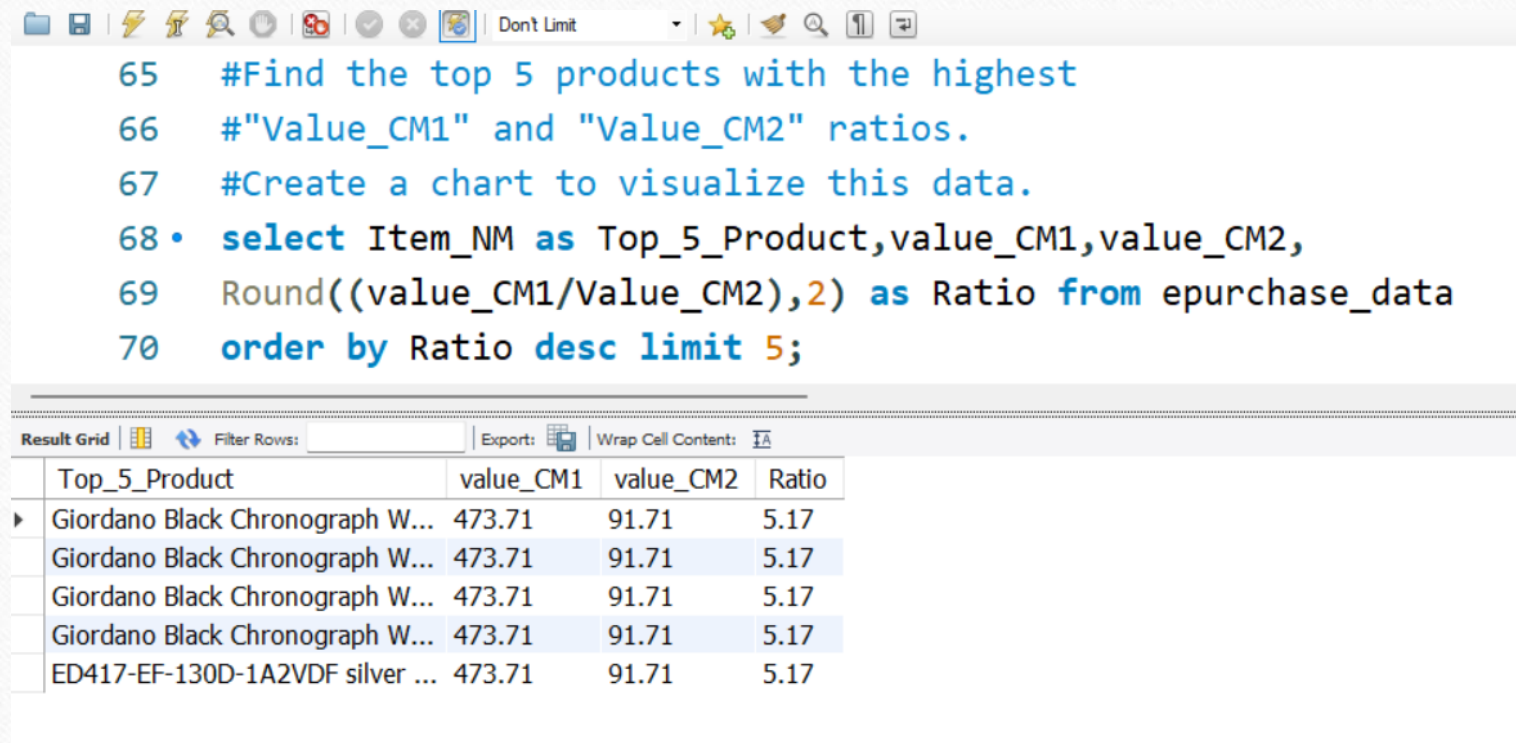
```
53  #calculate the average "Quantity" sold for products  
54  #in the "Clothing" category, grouped by "Product_Gender."  
55  • select count(Quantity) as Total_count, avg(Quantity) from  
56  epurchase_data where Category_Grouped like "Apparels"  
57  group by Product_Gender;  
58
```

Below the query editor, the "Result Grid" is displayed. It includes a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox. The results table has two columns: "Total_count" and "avg(Quantity)".

| | Total_count | avg(Quantity) |
|---|-------------|---------------|
| ▶ | 3087 | 1.0000 |
| | 6426 | 1.0000 |

12. Find the top 5 products with the highest "Value_CM1" and "Value_CM2" ratios. Create a chart to visualize this data.

➡ Select Item_NM as Top_5_Product, value_CM1, value_CM2, Round((value_CM1/Value_CM2),2) as Ratio from epurchase_data order by Ratio desc limit 5;



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

```
65 #Find the top 5 products with the highest
66 #"Value_CM1" and "Value_CM2" ratios.
67 #Create a chart to visualize this data.
68 • select Item_NM as Top_5_Product,value_CM1,value_CM2,
69 Round((value_CM1/Value_CM2),2) as Ratio from epurchase_data
70 order by Ratio desc limit 5;
```

Below the query editor is a "Result Grid" window. It has a toolbar with "Filter Rows", "Export", and "Wrap Cell Content" options. The grid displays the following data:

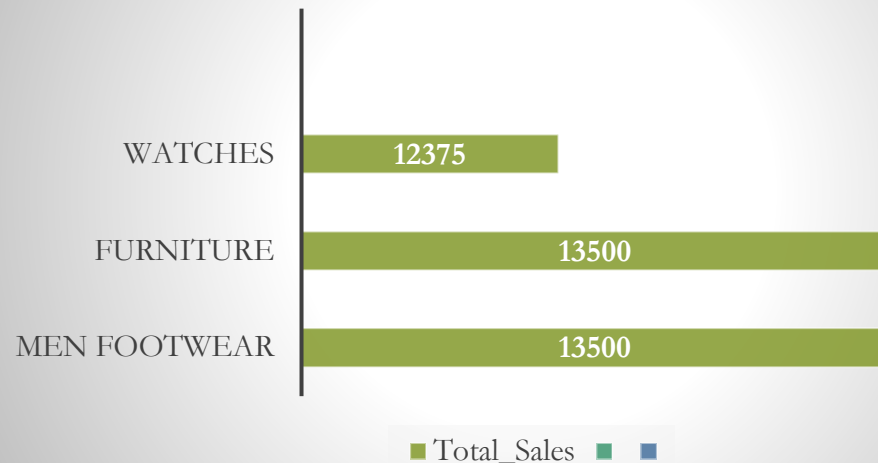
| Top_5_Product | value_CM1 | value_CM2 | Ratio |
|---------------------------------|-----------|-----------|-------|
| Giordano Black Chronograph W... | 473.71 | 91.71 | 5.17 |
| Giordano Black Chronograph W... | 473.71 | 91.71 | 5.17 |
| Giordano Black Chronograph W... | 473.71 | 91.71 | 5.17 |
| Giordano Black Chronograph W... | 473.71 | 91.71 | 5.17 |
| ED417-EF-130D-1A2VDF silver ... | 473.71 | 91.71 | 5.17 |

13. Identify the Top 3 "Class" categories with the highest total sales.

Create a stacked bar chart to represent this data.

select Category,Item_Price as Total_Sales from epurchase_data group by category,Item_Price order by Item_Price desc limit 3;

Top 3 Categories (Highest sales)



```
66
67 #Identify the top 3 "Class" categories with the highest total sales.
68 #Create a stacked bar chart to represent this data
69 • select Category,Item_Price as Total_Sales from epurchase_data
70   group by category,Item_Price
71   order by Item_Price desc limit 3;
72
```

| Result Grid | | Filter Rows: | Exports: | Wrap Cell Contents: |
|--------------|-------------|--------------|----------|---------------------|
| Category | Total_Sales | | | |
| Men Footwear | 13500 | | | |
| Furniture | 13500 | | | |
| WATCHES | 12375 | | | |

14. Use VLOOKUP or INDEX-MATCH to retrieve the "Color" of a product with a specific "Item_NM".

➡ select color, Item_NM from epurchase_data where Item_NM="Puma Sneakers Black";

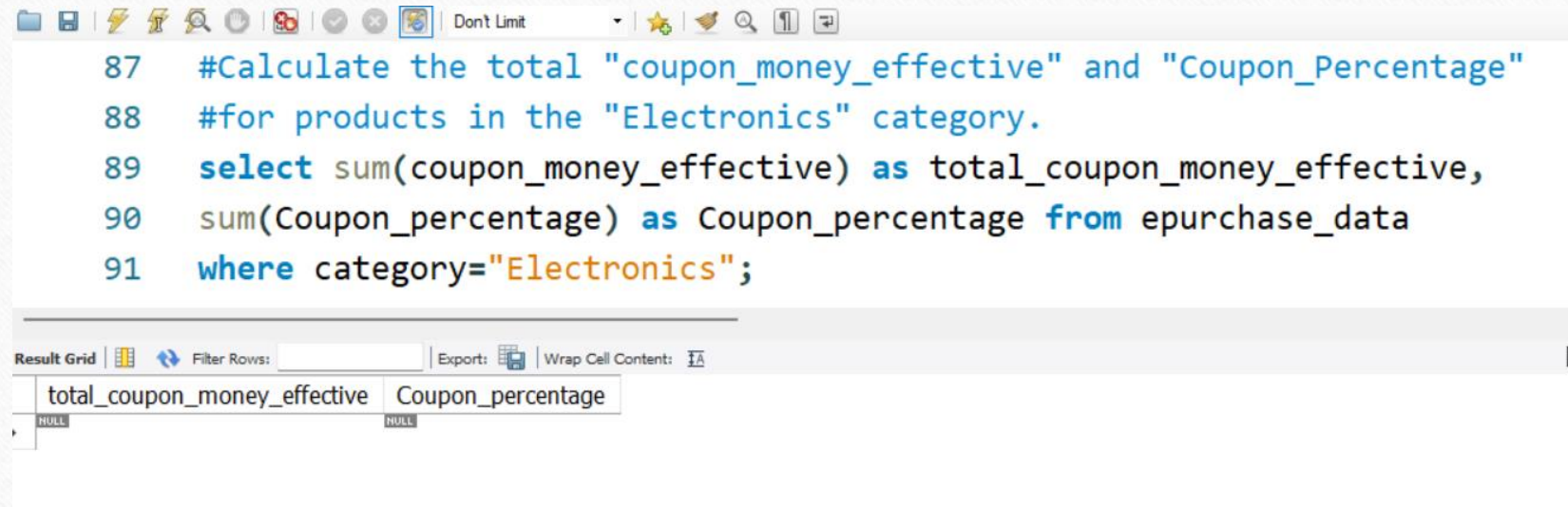
//

=VLOOKUP([@[Item_NM]],A1:AB50847,14,0)

//

15. Calculate the total "coupon_money_effective" and "Coupon_Percentage" for products in the "Electronics" category.

➡ select sum(coupon_money_effective) as total_coupon_money_effective, sum(Coupon_percentage) as Coupon_percentage from epurchase_data where category="Electronics";



The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

```
87 #Calculate the total "coupon_money_effective" and "Coupon_Percentage"
88 #for products in the "Electronics" category.
89 select sum(coupon_money_effective) as total_coupon_money_effective,
90 sum(Coupon_percentage) as Coupon_percentage from epurchase_data
91 where category="Electronics";
```

Below the query editor is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid itself is a table with two columns: "total_coupon_money_effective" and "Coupon_percentage". The first row of data shows "NULL" for both columns.

| total_coupon_money_effective | Coupon_percentage |
|------------------------------|-------------------|
| NULL | NULL |

16. Perform a time series analysis to identify the month with the highest total sales.

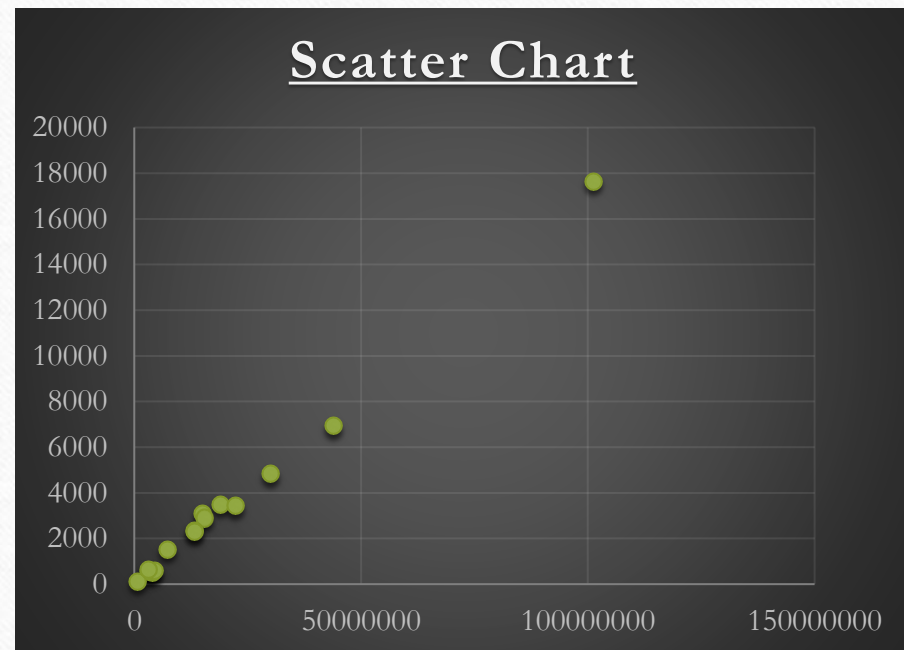
```
select sum(Item_price) as Total_Price, Extract Month() as Monthly_sales from  
epurchase_data group by Monthly_sales order by Total_Price;
```

17. Calculate the total sales for each "Segment" and create a scatter plot to visualize the relationship between "Item_Price" and "Quantity" in this data.

➡ select segment, sum(Item_price) as Total_sales, sum(Quantity) as Total_quantity from epurchase_data group by segment;

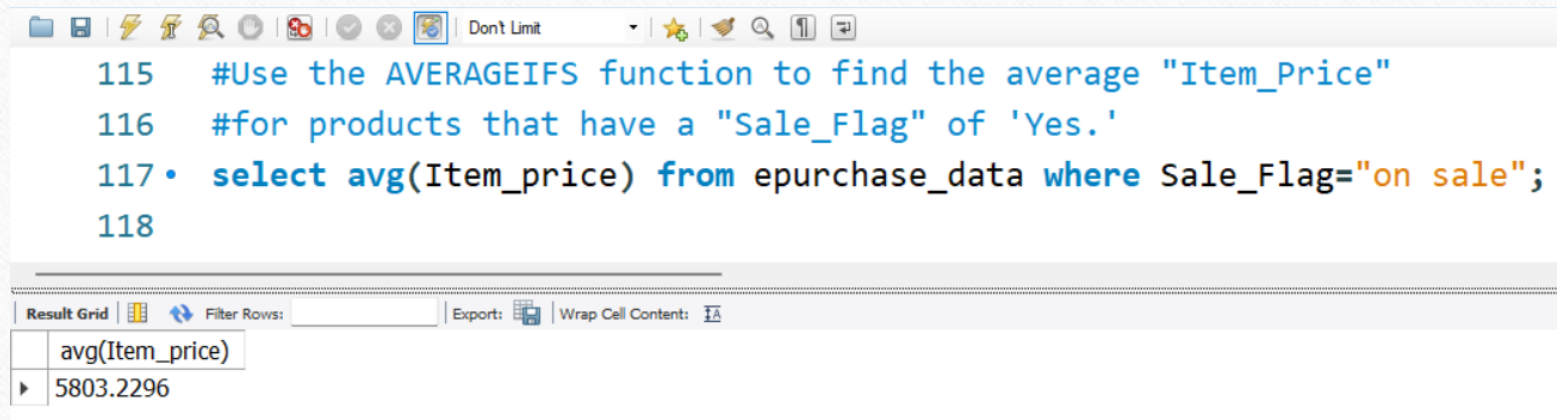
```
107 #Calculate the total sales for each "Segment" and create a scatter plot
108 #to visualize the relationship between "Item_Price"
109 #and "Quantity" in this data.
110 • select segment, sum(Item_price) as Total_sales, sum(Quantity) as Total_quantity
111 from epurchase_data group by segment;
```

| segment | Total_sales | Total_quantity |
|--------------------|-------------|----------------|
| MENS WEAR | 15039825 | 3087 |
| UNISEX | 15519256 | 2890 |
| MENS FOOTWEAR | 101245089 | 17653 |
| LADIES FOOTWEAR | 13408398 | 2347 |
| WOMEN | 30013131 | 4828 |
| WOMENS ACCESSORIES | 7386940 | 1529 |
| WOMENS WEAR | 44010575 | 6931 |
| MENS ACCESSORIES | 13279676 | 2287 |
| MEN | 22291728 | 3422 |
| LIVING | 3961755 | 524 |
| OUTDOOR & HIKING | 3045195 | 527 |
| MENS APPARELS | 723305 | 121 |
| WOMENS JEWELLERY | 4549307 | 605 |
| HOME FURNISHING | 3051213 | 641 |



18. Use the AVERAGEIFS function to find the average "Item_Price" for products that have a "Sale_Flag" of 'Yes.'

→ `select avg(Item_Price) from epurchase_data where Sale_Flag="on sale";
//=AVERAGEIFS(V2:V50847, P2:P50847, "On Sale")//`



The screenshot shows a software interface with a toolbar at the top containing icons for file operations, search, and execution. Below the toolbar is a text area with a SQL query. The query is as follows:

```
115 #Use the AVERAGEIFS function to find the average "Item_Price"  
116 #for products that have a "Sale_Flag" of 'Yes.'  
117 • select avg(Item_price) from epurchase_data where Sale_Flag="on sale";  
118
```

Below the query editor is a section labeled "Result Grid". It includes a "Filter Rows:" field and buttons for "Export:" and "Wrap Cell Content:". The result grid itself is a table with one column and one row:

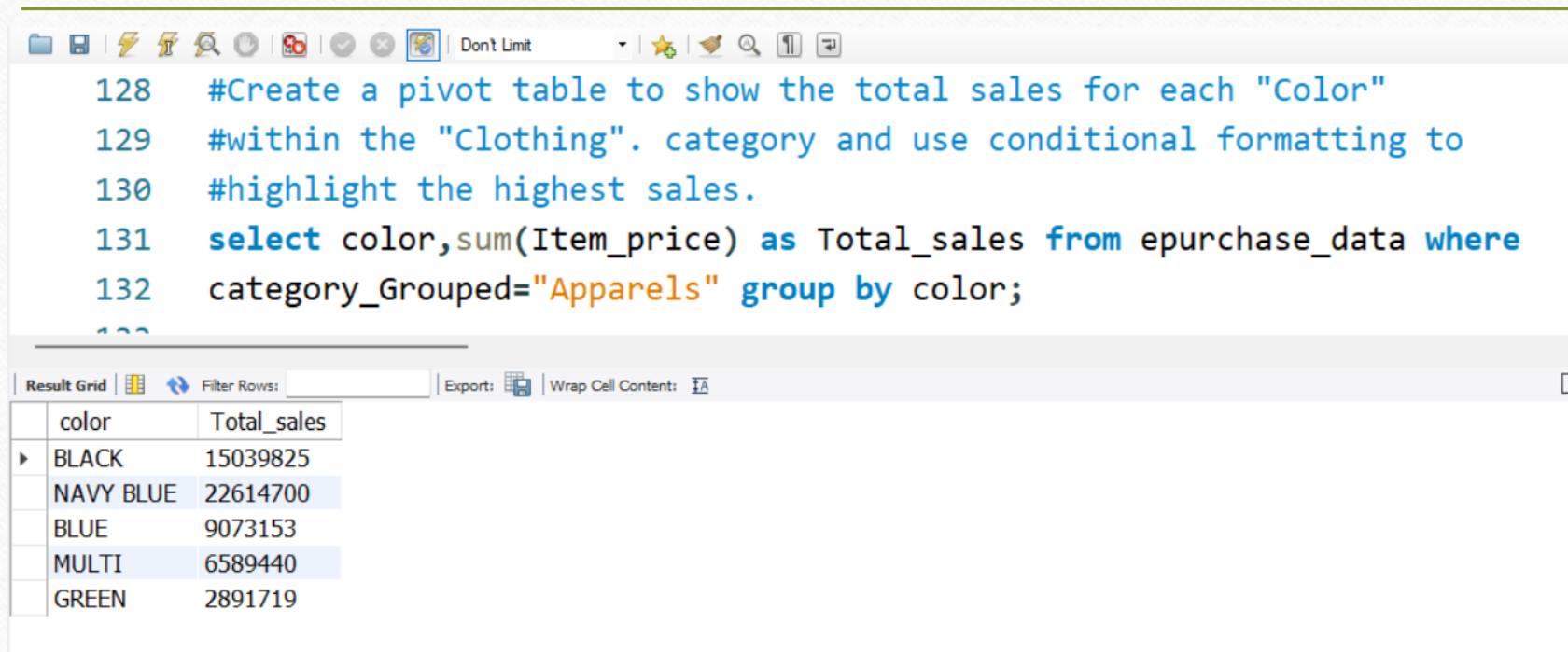
| avg(Item_price) |
|-----------------|
| 5803.2296 |

19. Identify products with a "Paid_pr" higher than the average in their respective "Family" and "Brand" groups.

➡ `select * from epurchase_data a where paid_pr > (select avg(paid_pr) as avg_price_pr from epurchase_data b where a.Family=b.Family and a.Brand=b.Brand);`

20. Create a pivot table to show the total sales for each "Color" within the "Clothing". Category and use conditional formatting to highlight the highest sales.

➡ select color,sum(Item_price) as Total_sales from epurchase_data where category_Grouped="Apparels" group by color;



The screenshot shows a software interface with a SQL query editor at the top and a results grid at the bottom. The query is a SQL statement to select color and total sales from a table named epurchase_data, filtered by category_Grouped = 'Apparels' and grouped by color. The results grid displays a table with two columns: color and Total_sales. The data is as follows:

| color | Total_sales |
|-----------|-------------|
| BLACK | 15039825 |
| NAVY BLUE | 22614700 |
| BLUE | 9073153 |
| MULTI | 6589440 |
| GREEN | 2891719 |



PSYLIQ



Thank You



SQL Internship at Psyliq

Prepared By: Utsav Raj

LinkedIn: [linkedin.com/in/iamutsavraj](https://www.linkedin.com/in/iamutsavraj)

