

**PAYTM
EPURCHASE
DATA**

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

The screenshot shows the SQL Server Enterprise Manager interface. In the Object Explorer on the left, the database 'DISHA' is expanded, and the table 'dbo.Purchase_data\$' is selected. The SQL Query window on the right contains the following query:

```
SELECT * FROM DBO.Purchase_data$;  
SELECT DISTINCT(Category_Grouped) FROM DBO.Purchase_data$;
```

The 'Results' tab displays the data from the query. The first table shows the full data from 'Purchase_data\$' with columns: S#no, Name, Shipping_city, Category_Grouped, Category, and Sub_category. The second table shows the distinct values of the 'Category_Grouped' column.

S#no	Name	Shipping_city	Category_Grouped	Category	Sub_category
1	ABHINAV CHATTER	Jabalpur	Others	SUNGLASSES	SUNGLASSES
2	AMIT GALPHADE	Ahmedabad	Apparels	Sports Equipment	Sports Apparel
3	PRABHU NAMBIAPP	Chennai	Others	Bags	Bags
4	MALLIKARJUNA H	Bangalore	Apparels	Sports Equipment	Sports Apparel
5	ANUPAM UPADHYAY	Gurgaon	NULL	Men Footwear	Mens Footwear
6	SITAL DE	Aalo	Shoes	Men Footwear	Mens Footwear
7	Abdul Qadir Sha	Kalyan	NULL	Men Footwear	MENS FOOTWE
8	kunal lavekar	Pune	Shoes	Men Footwear	Mens Footwear
9	Hardeep Mohan	Bangalore	Shoes	Men Footwear	Mens Footwear
10	ASHWIN GIDWANI	Pune	Apparels	Sports Equipment	Sports Apparel

Category_Grouped
1 Apparels
2 Others
3 Shoes
4 NULL
5 Home

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

SQLQuery2.sql - DE...NLE1Q4\Disha (54))* X SQLQu

```
use DISHA  
  
SELECT TOP 5  
    COUNT(*) AS total_orders,  
    Shipping_city  
FROM  
    dbo.Purchase_data$  
GROUP BY  
    Shipping_city  
ORDER BY  
    total_orders DESC;
```

100 %

Results Messages

	total_orders	Shipping_city
1	4560	New Delhi
2	4254	Chennai
3	3974	Bangalore
4	3159	Mumbai
5	2849	Hyderabad

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

There's no such category.

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

SQLQuery2.sql - DE...NLE1Q4\Disha (54)*

SQLQuery1.sql - not connected*

use DISHA

SELECT * FROM dbo.Purchase_data\$ WHERE Sale_Flag='On Sale';

100 %

Results

Messages

	Class	Family	Brand	Brick	Item_NM	Color	Size	Sale_Flag	Payment_Method	coupon_money_ef	
1		TOPS	SPORT & ADVENTURE	SKINS	TIGHTS	SKINS Navy Blue Tights	BLACK	XL	On Sale	COD	0
2		TOPS	SPORT & ADVENTURE	SKINS	TIGHTS	SKINS Navy Blue Tights	BLACK	XL	On Sale	COD	0
3		TOPS	SPORT & ADVENTURE	SKINS	TIGHTS	SKINS Navy Blue Tights	BLACK	XXXL	On Sale	COD	0
4	/EAR	NULL	SPORTS	ADIDAS	RUNNING SHOES	Adizero F50 2 M Black Running Shoes	BLACK	7	On Sale	Prepaid	1000
5	CESSORIES	WATCHES	NULL	PLAYBOY	ANALOG WATCH	Bpb-1004-C Silver/Black Analog Watch	SILVER	FREE SIZE	On Sale	COD	0
6		NULL	NULL	HIDESIGN	HANDBAG	Downing Street 04 Black Handbag	BLACK	STANDARD	On Sale	COD	1099.8
7		TOPS	SPORT & ADVENTURE	SKINS	TIGHTS	SKINS Navy Blue Tights	BLACK	XL	On Sale	COD	0
8	/EAR	NULL	SPORTS	REEBOK	RUNNING SHOES	Ventilator Hls Grey Running Shoes	GREY	10	On Sale	Prepaid	0
9	/EAR	NULL	SPORTS	NIKE	RUNNING SHOES	Nike Lunarglide+ 4 Grey Running Shoes	GREY	7	On Sale	COD	0
10	/EAR	NULL	SPORTS	ADIDAS	RUNNING SHOES	Adizero F50 2 M Black Running Shoes	BLACK	7	On Sale	Prepaid	1000

Query executed successfully.

DESKTOP-DNLE1Q4\SQLEXPRESS...DESKTOP-DNLE1Q4\Disha ...DISHA00:00:0016,171 rows

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

SQLQuery2.sql - DE...NLE1Q4\Disha (54))* SQLQuery1.sql - not connected*

use DISHA

```
SELECT * FROM dbo.Purchase_data$ ORDER BY Item_Price DESC;
```

Results Messages

	Payment_Method	coupon_money_effective	Coupon_Percentage	Quantity	Cost_Price	Item_Price	Special_Price_effective	paid_pr_effective	Value_CM1	Value_CM2	Special_price	Paid_pr
1	COD	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
2	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
3	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
4	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
5	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
6	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
7	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
8	COD	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
9	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990
10	Prepaid	0	NULL	1	3822.45	13500	4990	4990	588.6	-2956.4	4990	4990

Query executed successfully. DESKTOP-DNLE1Q4\SQLEXPRESS0... DESKTOP-DNLE1Q4\Disha ... DISHA 00:00:02 50,846 rows

SQLQuery2.sql - DE...NLE1Q4\Disha (54))* SQLQuery1.sql - not connected*

use DISHA

```
SELECT TOP 1 Item_Price, S#no FROM dbo.Purchase_data$ ORDER BY Item_Price DESC;
```

Results Messages

	Item_Price	S#no
1	13500	36

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

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL statement:

```
use DISHA  
  
SELECT * FROM dbo.Purchase_data$ where Special_Price_effective<50 AND Color='RED';
```

The bottom pane shows the results of the query. The results grid has the following columns: SNo, Name, Shipping_city, Category_Grouped, Category, Sub_category, Product_Gender, Segment, Class, Family, Brand, Brick, Item_NM, Color, Size, Sale_Hag, and Payment_Method. The results area is currently empty, displaying the message "NOTHING FOUND HERE!!".

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

use DISHA

```
SELECT SUM(Item_Price) AS TOTAL_SALES, Category_Grouped FROM dbo.Purchase_data$ GROUP BY Category_Grouped;
```

%

Results Messages

TOTAL_SALES	Category_Grouped
56208837	Apparels
83734877	Others
68406549	Shoes
85054962	NULL
3051213	Home

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

SQLQuery2.sql - DE...NLE1Q4\Disha (54))* SQLQuery1.sql - not connected*

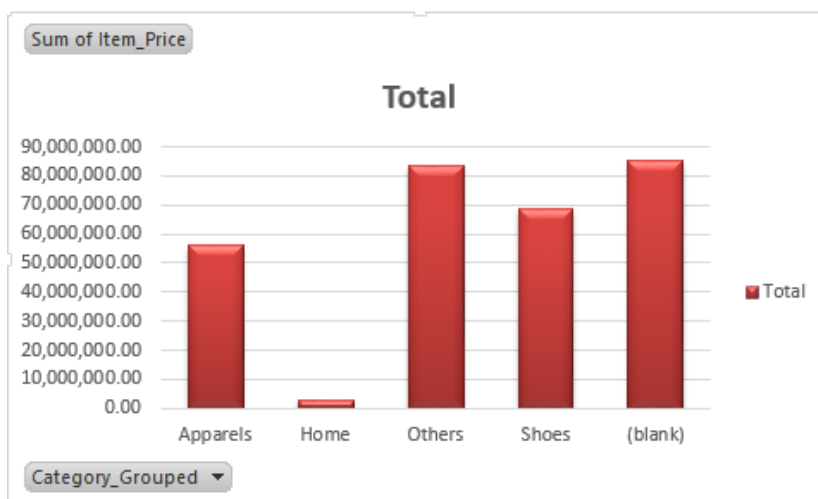
```
use DISHA
```

```
SELECT SUM(Item_Price) AS TOTAL_SALES,  
Category_Grouped  
FROM dbo.Purchase_data$  
GROUP BY Category_Grouped  
ORDER BY TOTAL_SALES DESC;
```

100 %

Results Messages

	TOTAL_SALES	Category_Grouped
1	85054962	NULL
2	83734877	Others
3	68406549	Shoes
4	56208837	Apparels
5	3051213	Home



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

```
use DISHA

SELECT Family, Count(*) AS TOTAL_PRODUCTS
FROM dbo.Purchase_data$
GROUP BY Family;
```

100 %

Results Messages

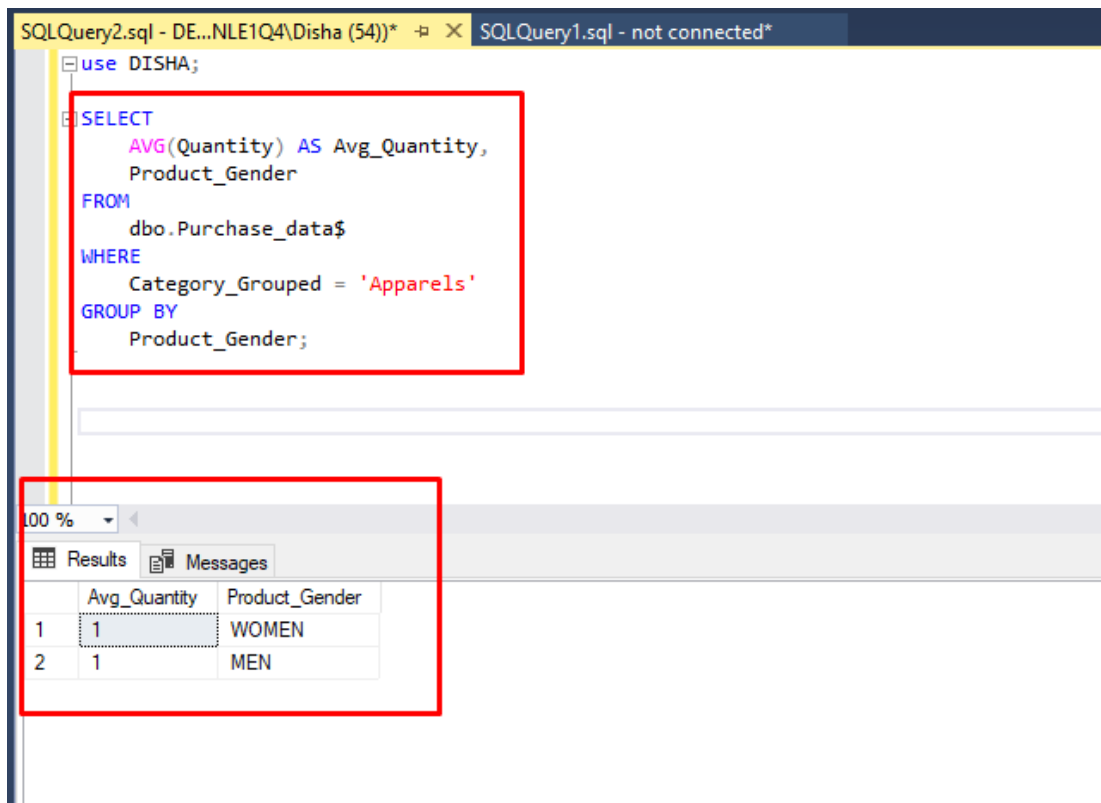
	Family	TOTAL_PRODUCTS
1	UNISEX	2912
2	NULL	15533
3	FLOORING	641
4	WOMEN	576
5	EQUIPMENTS	527
6	CASUAL WEAR	121
7	CASUAL	3670
8	ETHNIC	6426
9	ETHNIC WEAR	505
10	SPORT & ADVENTURE	3087
11	TABLES	524

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

```
use DISHA;

UPDATE dbo.Purchase_data$
SET Payment_Method =
CASE
    WHEN Payment_Method NOT IN ('Visa', 'MasterCard') THEN 'Null'
    ELSE Payment_Method
END;
```

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



The screenshot shows a SQL Server Enterprise Manager window with two tabs: "SQLQuery2.sql - DE...NLE1Q4\Disha (54))" and "SQLQuery1.sql - not connected". The active tab displays a SQL query that calculates the average quantity sold for products in the "Apparels" category, grouped by "Product_Gender". The query is as follows:

```
use DISHA;

SELECT
    AVG(Quantity) AS Avg_Quantity,
    Product_Gender
FROM
    dbo.Purchase_data$
WHERE
    Category_Grouped = 'Apparels'
GROUP BY
    Product_Gender;
```

The results of the query are displayed in a table with two columns: "Avg_Quantity" and "Product_Gender". The table shows two rows of data:

	Avg_Quantity	Product_Gender
1	1	WOMEN
2	1	MEN

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

SQLQuery2.sql - DE...NLE1Q4\Disha (54))* SQLQuery1.sql - not connected*

```
use DISHA;  
  
SELECT TOP 5  
FROM  
dbo.Purchase_data$  
ORDER BY  
Value_CM1 / NULLIF(Value_CM2, 0) DESC;
```

100 %

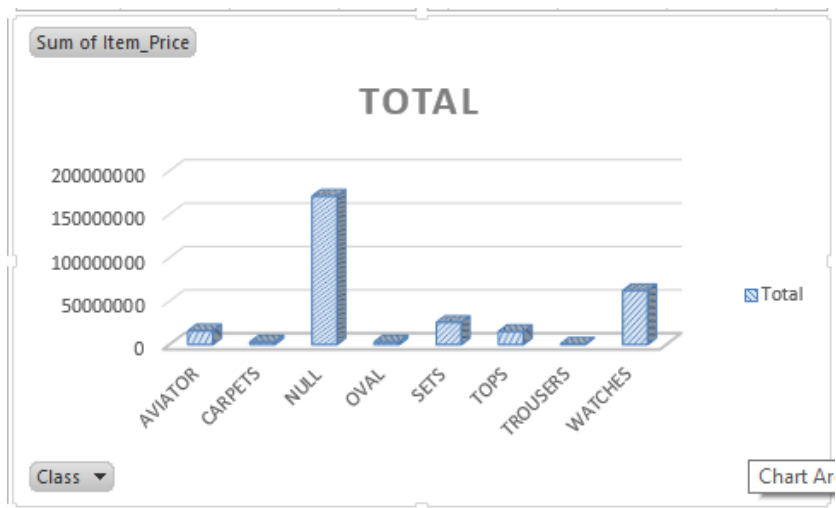
Results Messages

	Payment_Method	coupon_money_effective	Coupon_Percentage	Quantity	Cost_Price	Item_Price	Special_Price_effective	paid_pr_effective	Value_CM1	Value_CM2	Special_price	Paid_pr
1	NULL	0	NULL	1	3503.31	5999	4499	4499	473.71	91.71	4499	4499
2	NULL	0	NULL	1	3503.31	5999	4499	4499	473.71	91.71	4499	4499
3	NULL	0	NULL	1	3503.31	5999	4499	4499	473.71	91.71	4499	4499
4	NULL	0	NULL	1	3503.31	5999	4499	4499	473.71	91.71	4499	4499
5	NULL	0	NULL	1	3503.31	5999	4499	4499	473.71	91.71	4499	4499

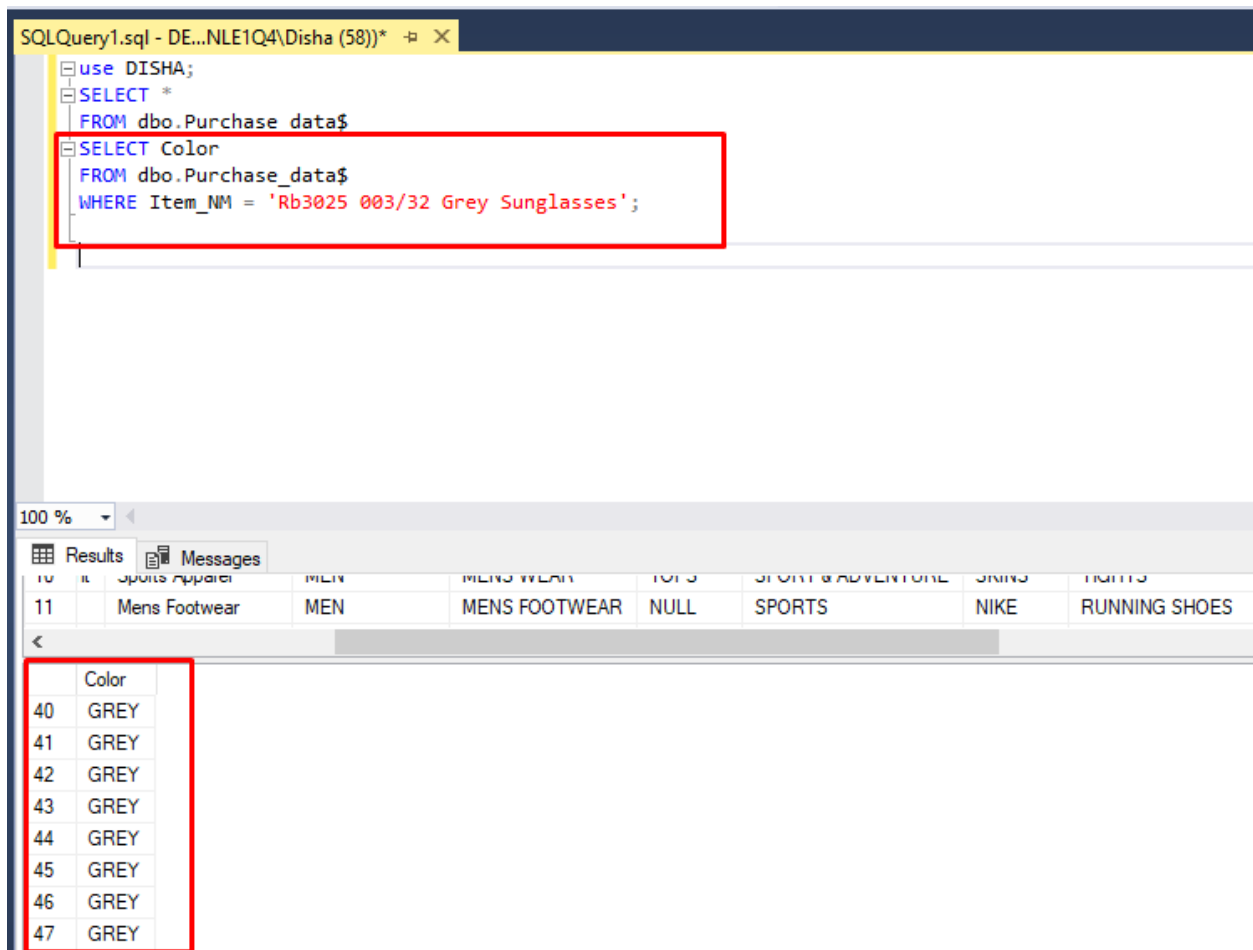
13. Identify the top 3 "Class" categories with the highest total sales. Create a stacked bar chart to represent this data.

```
use DISHA;  
  
SELECT TOP 3  
    Class, SUM(Item_Price) AS TotalSales  
FROM  
    dbo.Purchase_data$  
GROUP BY  
    Class  
ORDER BY  
    TotalSales DESC;
```

	Class	TotalSales
1	NULL	170968090
2	WATCHES	62213793
3	SETS	25529167



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



The screenshot shows a SQL query window with the following code:

```
use DISHA;  
SELECT *  
FROM dbo.Purchase_data$  
SELECT Color  
FROM dbo.Purchase_data$  
WHERE Item_NM = 'Rb3025 003/32 Grey Sunglasses';
```

The query results are displayed in a table with the following columns: Item, Sports Apparel, MEN, MENS WEAR, TOYS, SPORTS & ADVENTURE, SHIRTS, and TIGHTS. The results show a single row for Item 11, which is 'Mens Footwear'.

Item	Sports Apparel	MEN	MENS WEAR	TOYS	SPORTS & ADVENTURE	SHIRTS	TIGHTS
11	Mens Footwear	MEN	MENS FOOTWEAR	NULL	SPORTS	NIKE	RUNNING SHOES

Below the table, a list of results is shown, with the 'Color' column highlighted. The results are:

Color
40 GREY
41 GREY
42 GREY
43 GREY
44 GREY
45 GREY
46 GREY
47 GREY

=INDEX(C1:C100, MATCH("specific_item_nm", A1:A100, 0))

0 in MATCH ensures an exact match

A1:A100 is the range of "Item_NM," and C1:C100 is the range of "Color."

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

SQLQuery1.sql - DE...NLE1Q4\Disha (58)) * -> X

```
use DISHA;  
SELECT  
    SUM(coupon_money_effective) AS TotalCouponMoneyEffective,  
    SUM(Coupon_Percentage) AS TotalCouponPercentage  
FROM  
    dbo.Purchase_data$  
WHERE  
    Category = 'Electronics';
```

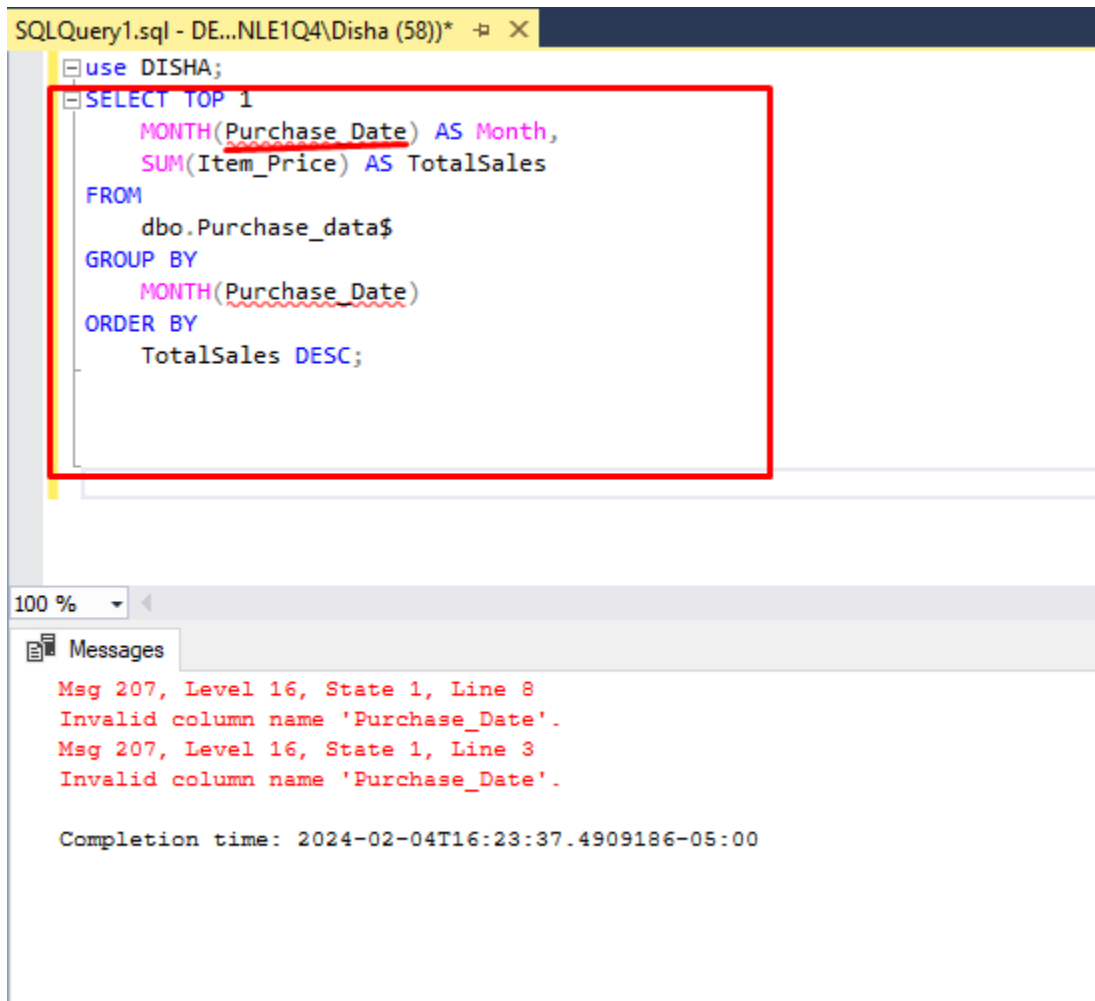
100 %

Results Messages

	TotalCouponMoneyEffective	TotalCouponPercentage
1	NULL	NULL

2 of 2

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



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a messages pane. The query editor contains the following SQL code:

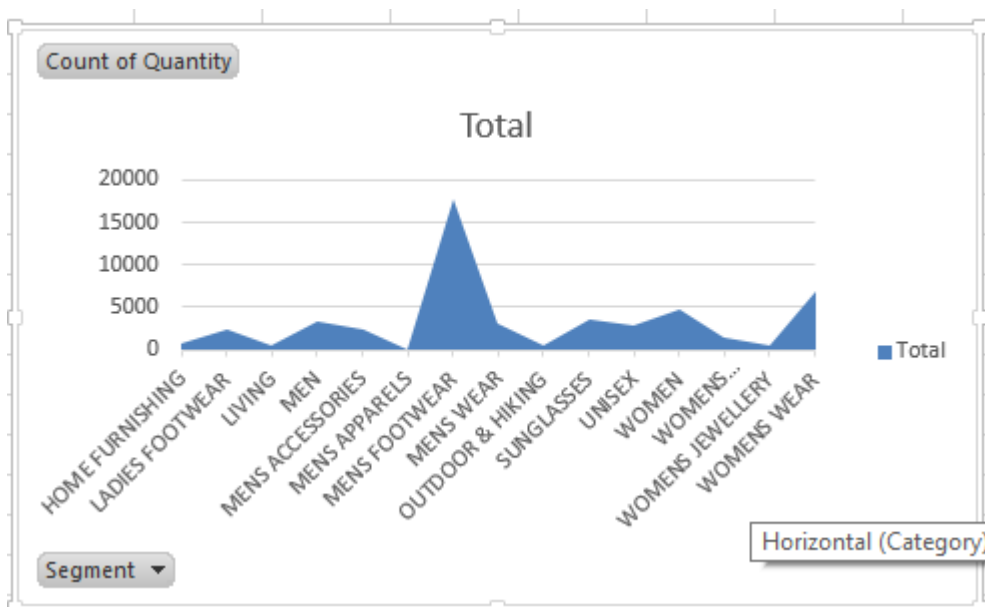
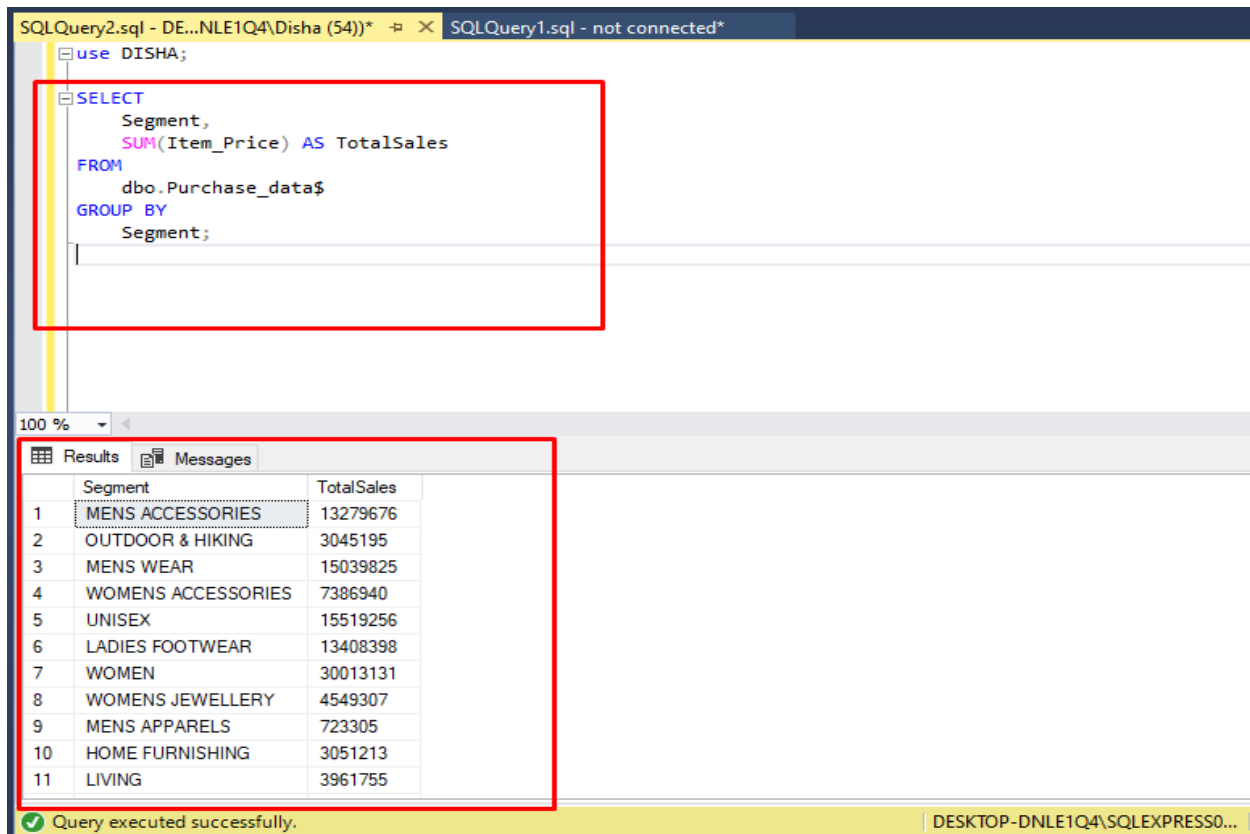
```
use DISHA;  
SELECT TOP 1  
    MONTH(Purchase_Date) AS Month,  
    SUM(Item_Price) AS TotalSales  
FROM  
    dbo.Purchase_data$  
GROUP BY  
    MONTH(Purchase_Date)  
ORDER BY  
    TotalSales DESC;
```

The messages pane shows two error messages:

```
Msg 207, Level 16, State 1, Line 8  
Invalid column name 'Purchase_Date'.  
Msg 207, Level 16, State 1, Line 3  
Invalid column name 'Purchase_Date'.  
  
Completion time: 2024-02-04T16:23:37.4909186-05:00
```

Date hasn't been provided to us.

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



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

SQLQuery1.sql - DE...NLE1Q4\Disha (58))*

```
use DISHA;  
SELECT  
*  
FROM  
    dbo.Purchase_data$  
  
SELECT  
    AVG(Item_Price) AS AverageItemPrice  
FROM  
    dbo.Purchase_data$  
WHERE  
    Sale_Flag = 'On Sale';
```

100 %

Results Messages

	S#no	Name	Shipping_city	Category_Grouped	Category	Sub_category	Product_Gender
1	1	ABHINAV CHATTER	Jabalpur	Others	SUNGLASSES	SUNGLASSES	UNISEX
2	2	AMIT GALPHADE	Ahmedabad	Apparels	Sports Equipment	Sports Apparel	MEN
3	3	PRABHU NAMBIAPP	Chennai	Others	Bags	Bags	UNISEX
4	4	MALLIKARJUNA H	Bangalore	Apparels	Sports Equipment	Sports Apparel	MEN
5	5	ANUBAM UBADHYAY	Gurgaon	NULL	Men Footwear	Men Footwear	MEN

	AverageItemPrice
1	5803.22960855853

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

```
use DISHA;  
  
SELECT  
    S#no,  
    Family,  
    Brand,  
    Paid_pr  
FROM  
    dbo.Purchase_data$ AS pd  
WHERE  
    Paid_pr > (  
        SELECT AVG(Paid_pr)  
        FROM dbo.Purchase_data$ AS sub_pd  
        WHERE sub_pd.Family = pd.Family AND sub_pd.Brand = pd.Brand  
    );
```

100 %

Results Messages

	S#no	Family	Brand	Paid_pr
1	2	SPORT & ADVENTURE	SKINS	4999
2	4	SPORT & ADVENTURE	SKINS	4999
3	5	SPORTS	NIKE	5621
4	7	CASUAL	CLARKS	4560
5	8	SPORTS	NIKE	4995
6	13	CASUAL	WOODLAND	5621
7	17	SPORTS	REEBOK	5670
8	20	NULL	PLAYBOY	4799
9	26	SPORTS	NIKE	5299
10	30	ETHNIC	SANGRIA	6396
11	31	NULL	VIP	5983

Query executed successfully. DESKTOP-DNLE1Q4\SQLEXPRESS0...

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.

There's no Clothing category but there is "Apparels" category.

Men Apparel	723305
BEIGE	723305

Women Apparel	44010575
BLACK	2841563
BLUE	9073153
GREEN	2891719
MULTI	6589440
NAVY BLUE	22614700

SQL

The screenshot shows a SQL query window with the following code:

```
use DISHA;  
SELECT  
    Color,  
    SUM(Item_Price) as PRICE  
FROM  
    dbo.Purchase_data$  
WHERE  
    Category_Grouped = 'Apparels'  
GROUP BY  
    Color;
```

Below the query window, the 'Results' tab is active, displaying the following data:

	Color	PRICE
1	GREEN	2891719
2	NAVY BLUE	22614700
3	MULTI	6589440
4	BLACK	15039825
5	BLUE	9073153

Item_Price	Special_Price_effective
4999	4999
4999	4999
4095	4095
4999	4999
7495	7495
6495	6495
4560	4560
5995	5995
5690	5690
4020	4020
4999	4999
5795	4636
7495	5621
5795	4636
5090	5090
5499	5499
6300	5670
4295	4295
4020	4020
4799	4799
4999	4999
5499	5499
4925	4186
4990	4990
5499	5499
5299	5299
5499	5499
4195	4195
4550	4095
7995	6396
11966	5983
4925	4186