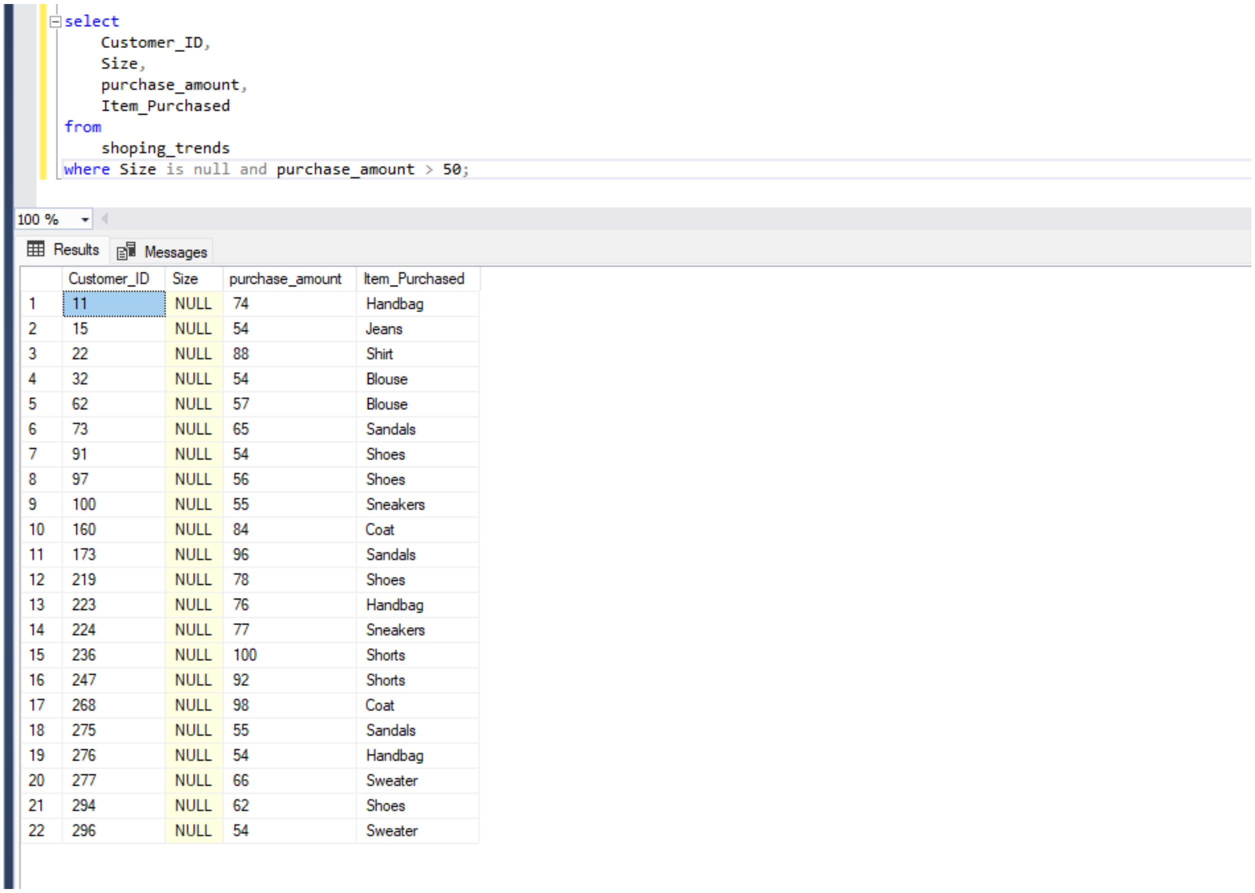
BrightLight Data Analytics Coding Practical (SQL SERVER)

Practical 2.1: Advanced SQL

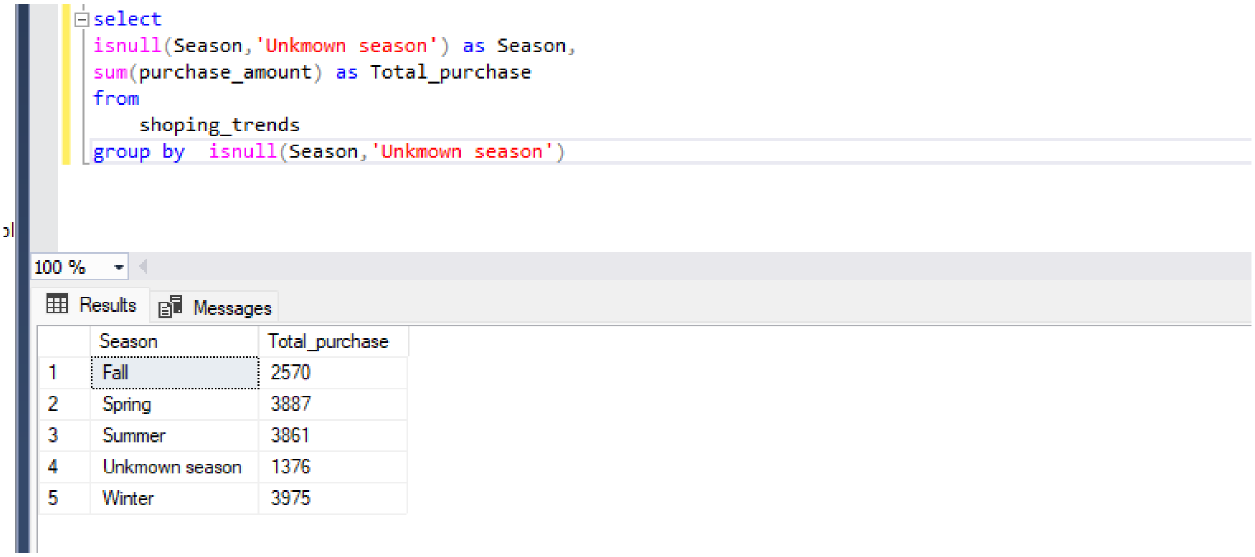
1. **Find all records where Size is missing and the purchase amount is greater than 50.**



```
select
    Customer_ID,
    Size,
    purchase_amount,
    Item_Purchased
from
    shoping_trends
where Size is null and purchase_amount > 50;
```

	Customer_ID	Size	purchase_amount	Item_Purchased
1	11	NULL	74	Handbag
2	15	NULL	54	Jeans
3	22	NULL	88	Shirt
4	32	NULL	54	Blouse
5	62	NULL	57	Blouse
6	73	NULL	65	Sandals
7	91	NULL	54	Shoes
8	97	NULL	56	Shoes
9	100	NULL	55	Sneakers
10	160	NULL	84	Coat
11	173	NULL	96	Sandals
12	219	NULL	78	Shoes
13	223	NULL	76	Handbag
14	224	NULL	77	Sneakers
15	236	NULL	100	Shorts
16	247	NULL	92	Shorts
17	268	NULL	98	Coat
18	275	NULL	55	Sandals
19	276	NULL	54	Handbag
20	277	NULL	66	Sweater
21	294	NULL	62	Shoes
22	296	NULL	54	Sweater

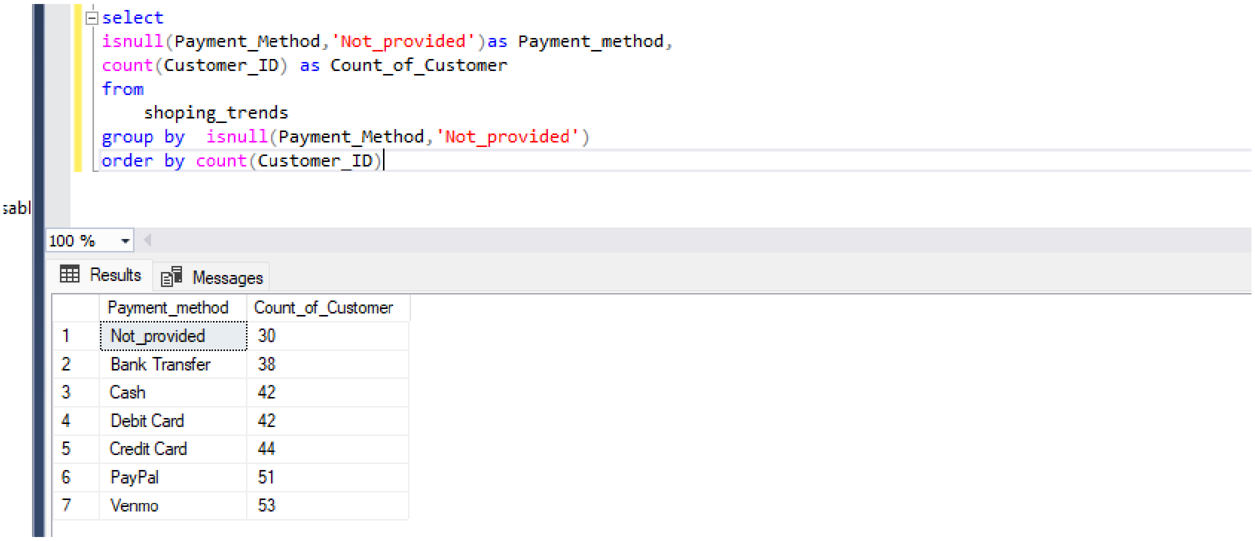
2. **List the total number of purchases grouped by Season, treating NULL values a s 'Unknown Season'.**



```
select
isnull(Season,'Unkmown season') as Season,
sum(purchase_amount) as Total_purchase
from
    shoping_trends
group by  isnull(Season,'Unkmown season')
```

	Season	Total_purchase
1	Fall	2570
2	Spring	3887
3	Summer	3861
4	Unkmown season	1376
5	Winter	3975

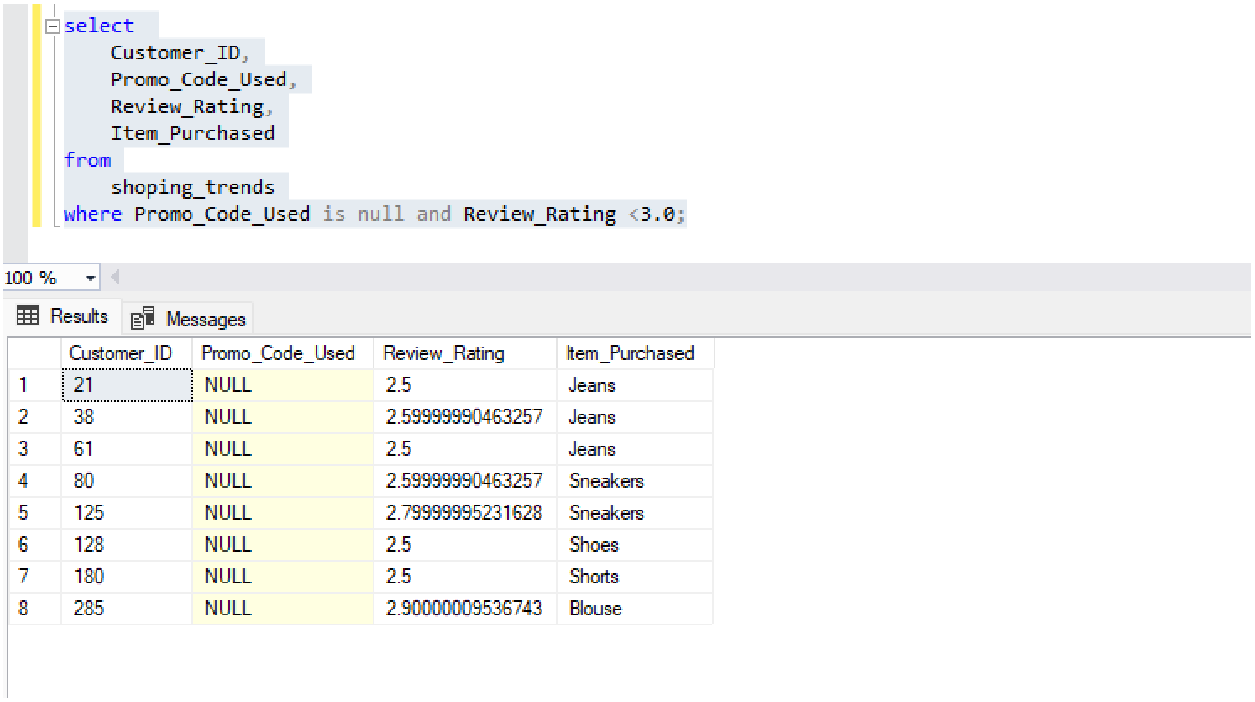
3. **Count how many customers used each Payment Method, treating NULLs as 'Not Provided'.**



```
select
isnull(Payment_Method,'Not_provided')as Payment_method,
count(Customer_ID) as Count_of_Customer
from
    shoping_trends
group by  isnull(Payment_Method,'Not_provided')
order by count(Customer_ID)
```

	Payment_method	Count_of_Customer
1	Not_provided	30
2	Bank Transfer	38
3	Cash	42
4	Debit Card	42
5	Credit Card	44
6	PayPal	51
7	Venmo	53

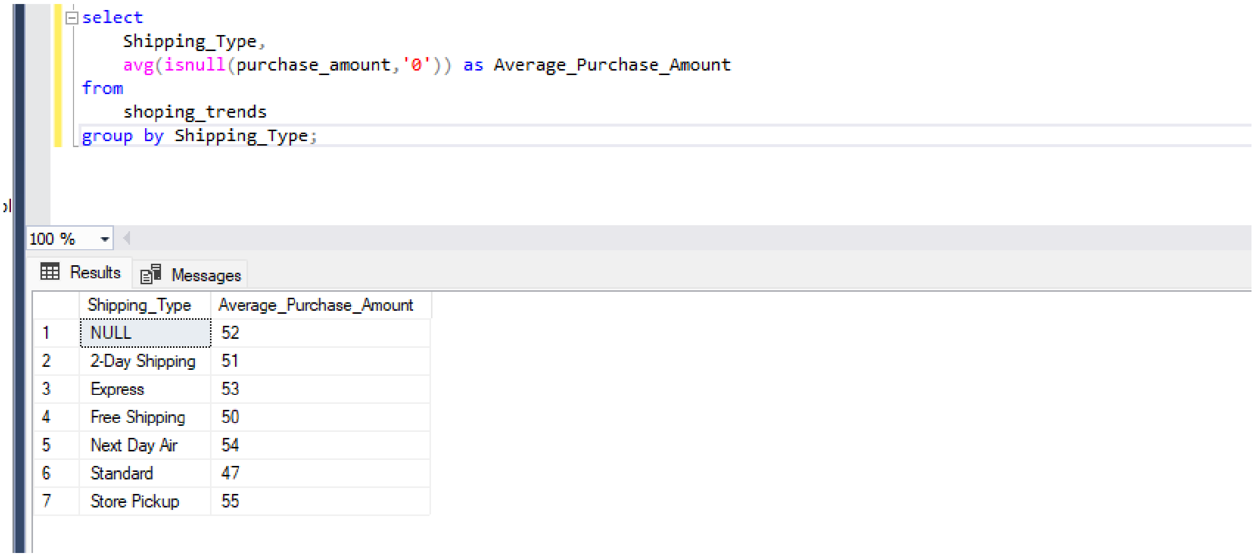
4. **Show customers where Promo Code Used is NULL and Review Rating is below 3.0**



```sql
select
    Customer_ID,
    Promo_Code_Used,
    Review_Rating,
    Item_Purchased
from
    shoping_trends
where Promo_Code_Used is null and Review_Rating <3.0;
```

	Customer_ID	Promo_Code_Used	Review_Rating	Item_Purchased
1	21	NULL	2.5	Jeans
2	38	NULL	2.59999990463257	Jeans
3	61	NULL	2.5	Jeans
4	80	NULL	2.59999990463257	Sneakers
5	125	NULL	2.79999995231628	Sneakers
6	128	NULL	2.5	Shoes
7	180	NULL	2.5	Shorts
8	285	NULL	2.90000009536743	Blouse

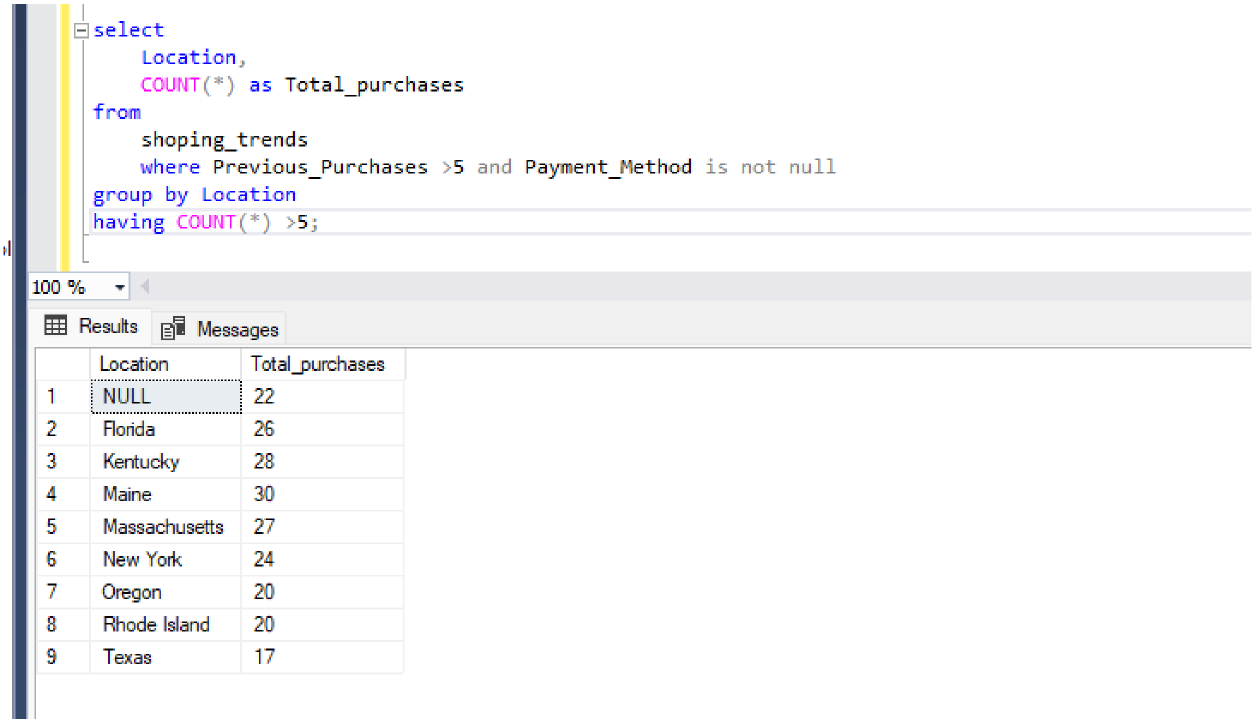
5. **Group customers by Shipping Type, and return the average purchase amount, treating missing values as 0.**



```sql
select
    Shipping_Type,
    avg(isnull(purchase_amount,'0')) as Average_Purchase_Amount
from
    shoping_trends
group by Shipping_Type;
```

	Shipping_Type	Average_Purchase_Amount
1	NULL	52
2	2-Day Shipping	51
3	Express	53
4	Free Shipping	50
5	Next Day Air	54
6	Standard	47
7	Store Pickup	55

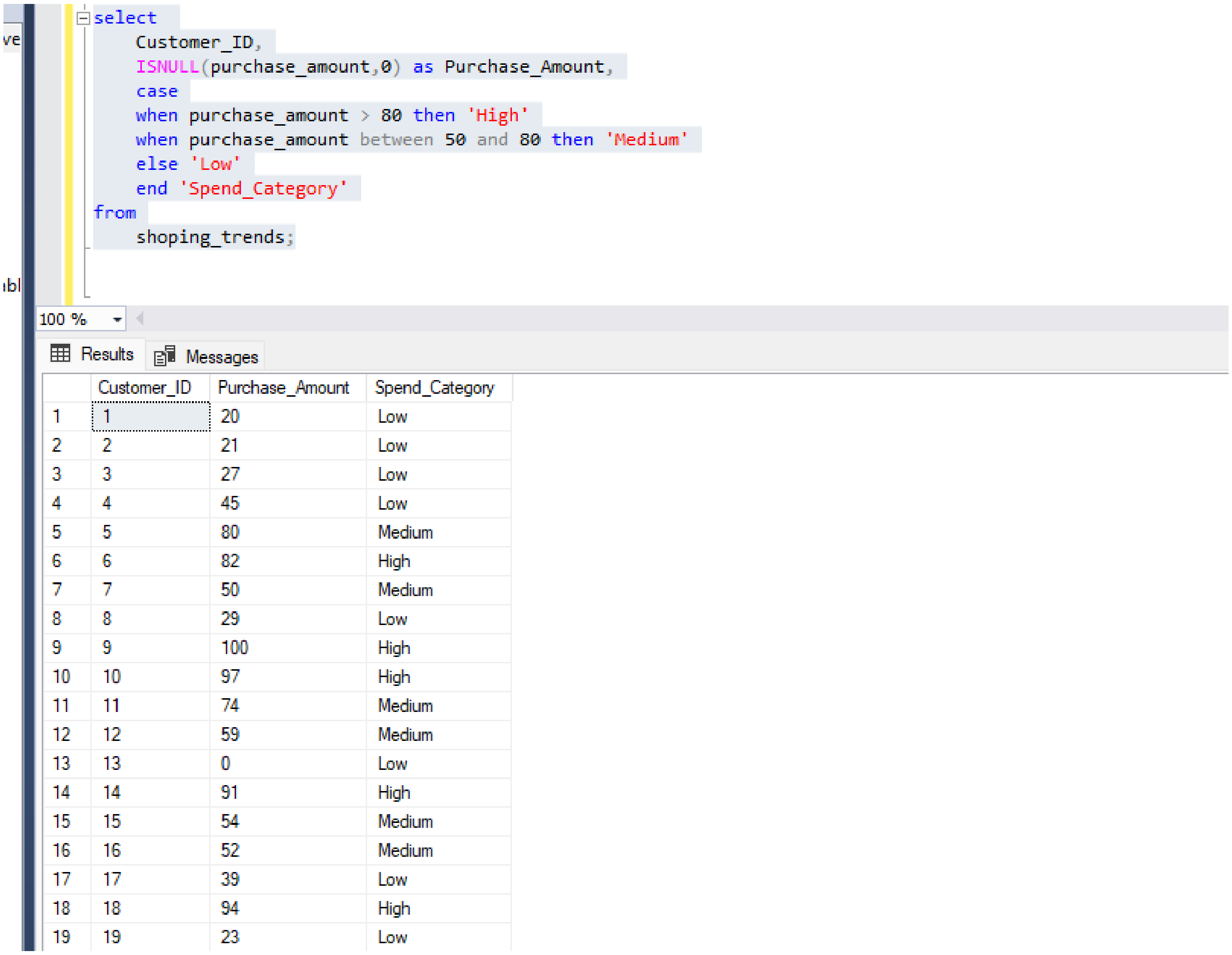
6. **Display the number of purchases per Location only for those with more than 5 purchases and no NULL Payment Method.**



```
select
    Location,
    COUNT(*) as Total_purchases
from
    shoping_trends
    where Previous_Purchases >5 and Payment_Method is not null
group by Location
having COUNT(*) >5;
```

	Location	Total_purchases
1	NULL	22
2	Florida	26
3	Kentucky	28
4	Maine	30
5	Massachusetts	27
6	New York	24
7	Oregon	20
8	Rhode Island	20
9	Texas	17

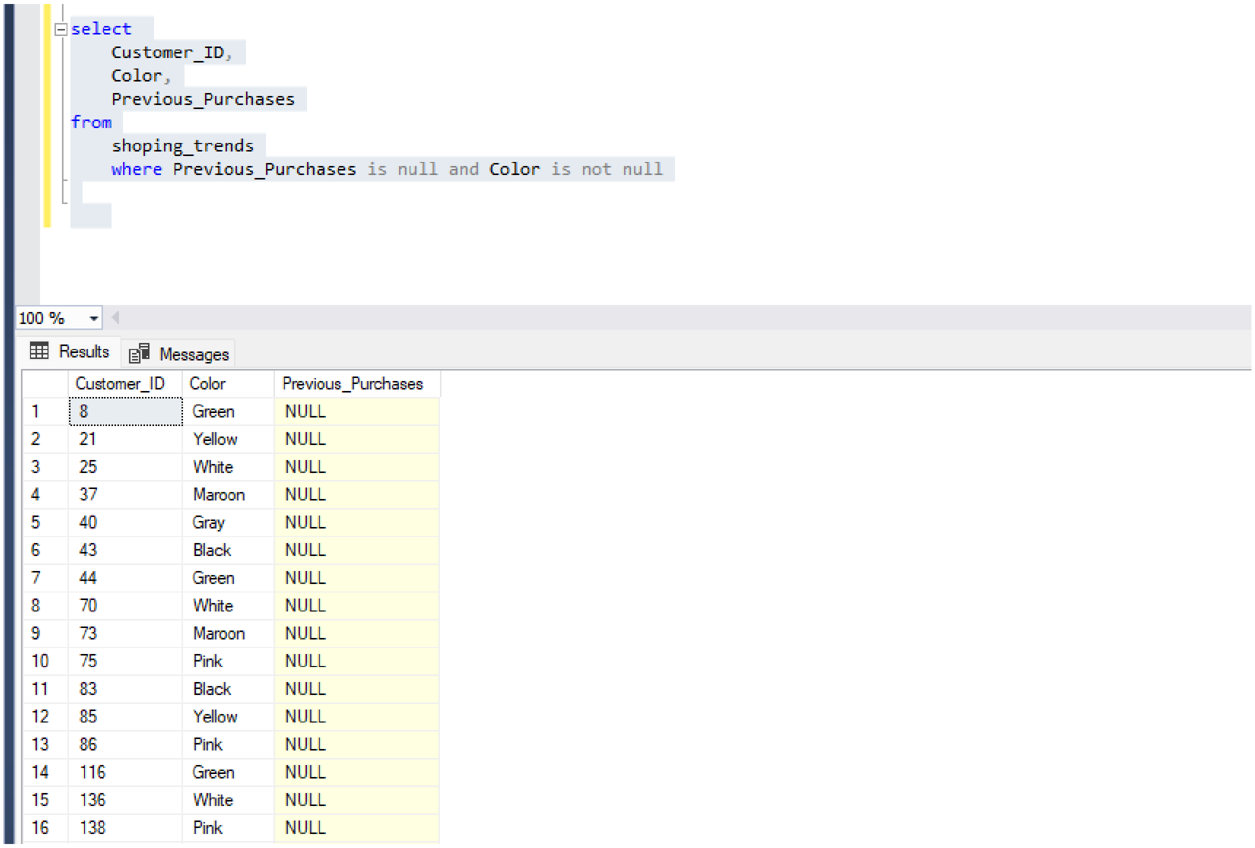
7. **Create a column Spender Category that classifies customers using CASE: 'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80, 'Low' otherwise. Replace NULLs in purchase amount with 0.**



```sql
select
    Customer_ID,
    ISNULL(purchase_amount,0) as Purchase_Amount,
    case
    when purchase_amount > 80 then 'High'
    when purchase_amount between 50 and 80 then 'Medium'
    else 'Low'
    end 'Spend_Category'
from
    shoping_trends;
```

	Customer_ID	Purchase_Amount	Spend_Category
1	1	20	Low
2	2	21	Low
3	3	27	Low
4	4	45	Low
5	5	80	Medium
6	6	82	High
7	7	50	Medium
8	8	29	Low
9	9	100	High
10	10	97	High
11	11	74	Medium
12	12	59	Medium
13	13	0	Low
14	14	91	High
15	15	54	Medium
16	16	52	Medium
17	17	39	Low
18	18	94	High
19	19	23	Low

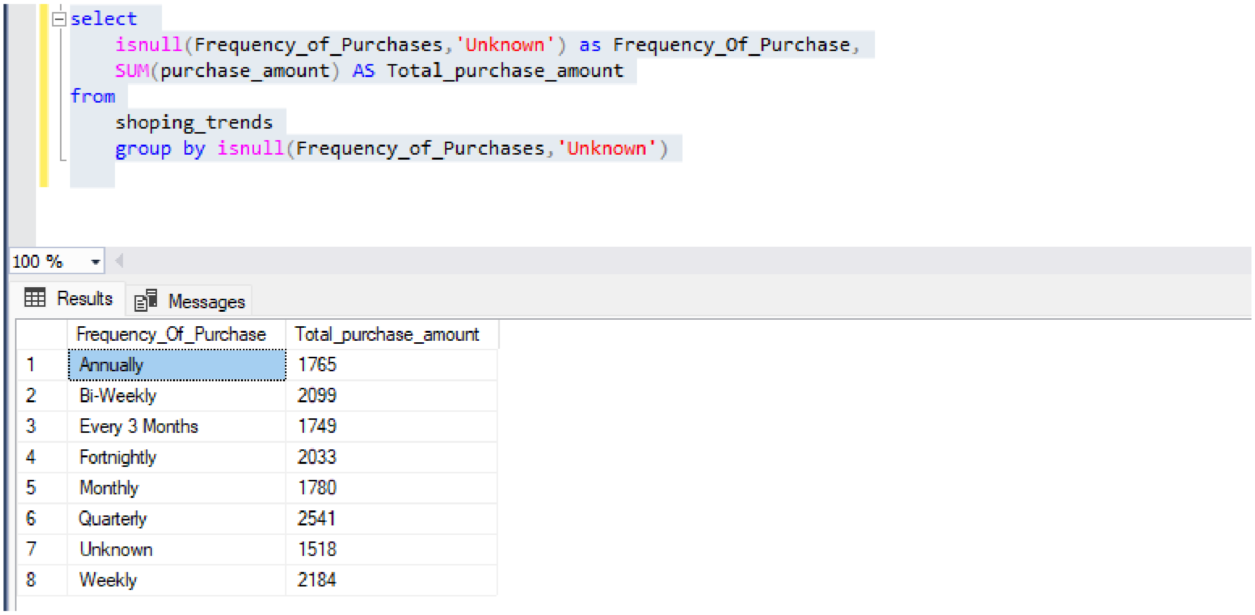
8. **Find customers who have no Previous Purchases value but whose Color is not NULL.**



```
select
    Customer_ID,
    Color,
    Previous_Purchases
from
    shoping_trends
    where Previous_Purchases is null and Color is not null
```

	Customer_ID	Color	Previous_Purchases
1	8	Green	NULL
2	21	Yellow	NULL
3	25	White	NULL
4	37	Maroon	NULL
5	40	Gray	NULL
6	43	Black	NULL
7	44	Green	NULL
8	70	White	NULL
9	73	Maroon	NULL
10	75	Pink	NULL
11	83	Black	NULL
12	85	Yellow	NULL
13	86	Pink	NULL
14	116	Green	NULL
15	136	White	NULL
16	138	Pink	NULL

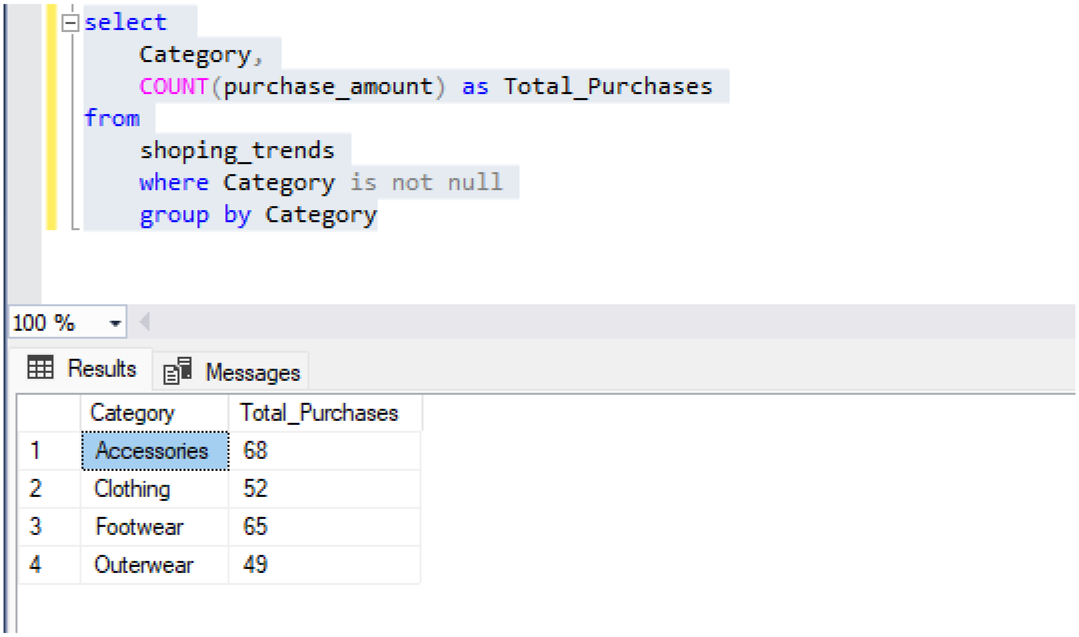
9. **Group records by Frequency of Purchases and show the total amount spent per group, treating NULL frequencies as 'Unknown'.**



```
select
    isnull(Frequency_of_Purchases,'Unknown') as Frequency_Of_Purchase,
    SUM(purchase_amount) AS Total_purchase_amount
from
    shoping_trends
    group by isnull(Frequency_of_Purchases,'Unknown')
```

	Frequency_Of_Purchase	Total_purchase_amount
1	Annually	1765
2	Bi-Weekly	2099
3	Every 3 Months	1749
4	Fortnightly	2033
5	Monthly	1780
6	Quarterly	2541
7	Unknown	1518
8	Weekly	2184

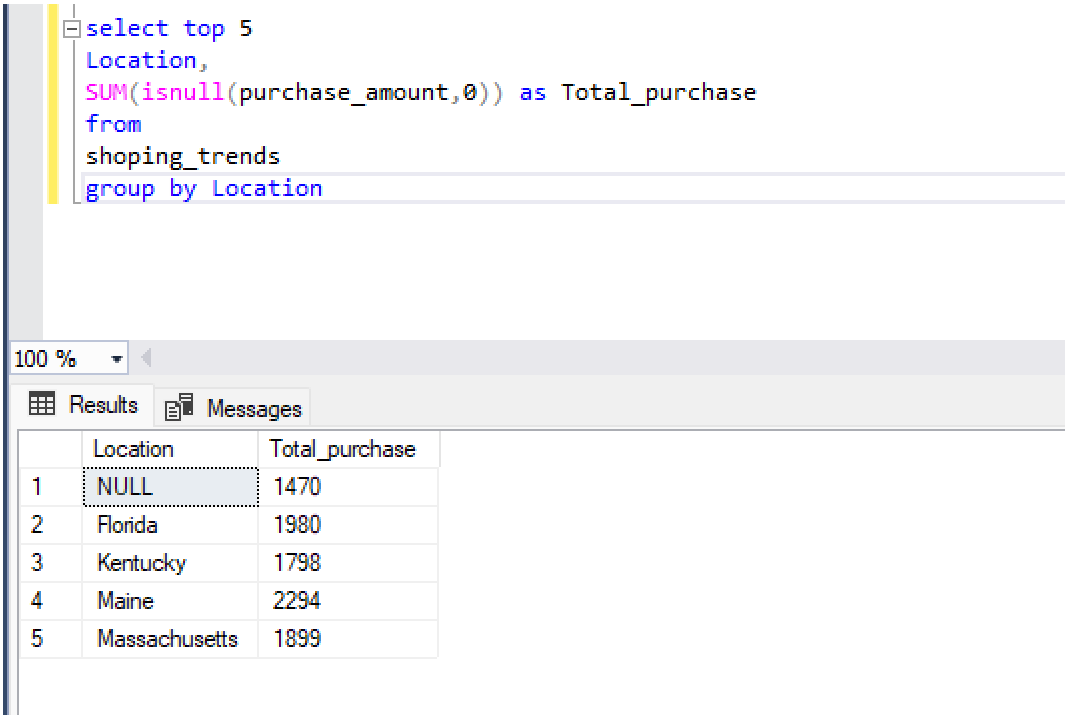
10. . Display a list of all Category values with the number of times each was purcha sed, excluding rows where Categoryis NULL.



```
select
    Category,
    COUNT(purchase_amount) as Total_Purchases
from
    shoping_trends
    where Category is not null
    group by Category
```

	Category	Total_Purchases
1	Accessories	68
2	Clothing	52
3	Footwear	65
4	Outerwear	49

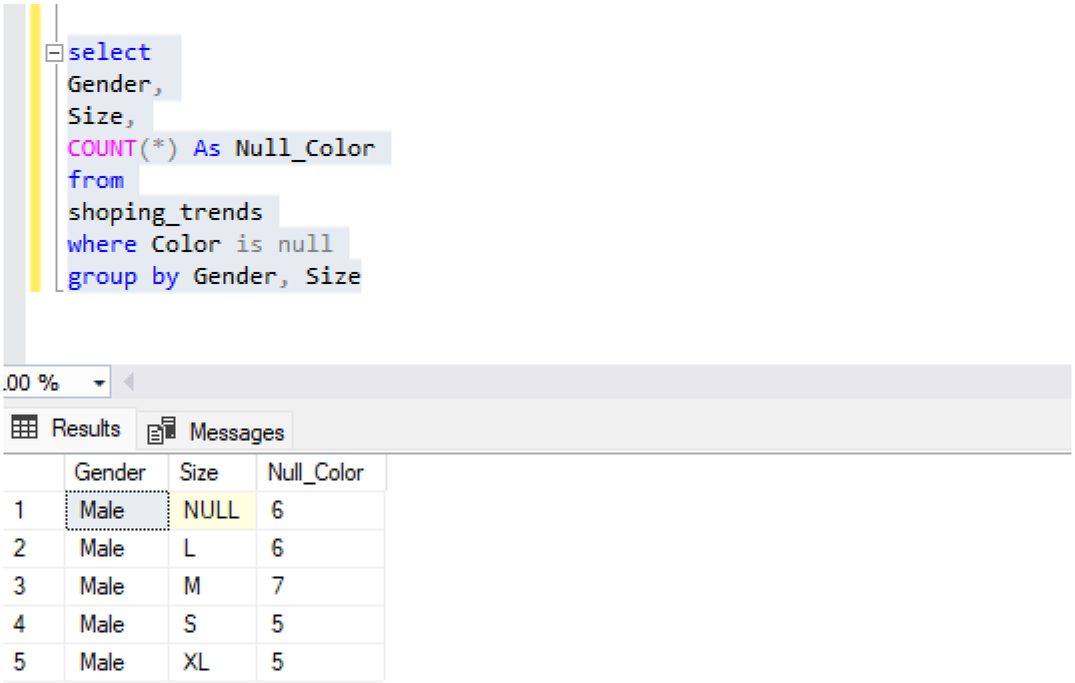
11. . Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0



```
select top 5
Location,
SUM(isnull(purchase_amount,0)) as Total_purchase
from
shoping_trends
group by Location
```

	Location	Total_purchase
1	NULL	1470
2	Florida	1980
3	Kentucky	1798
4	Maine	2294
5	Massachusetts	1899

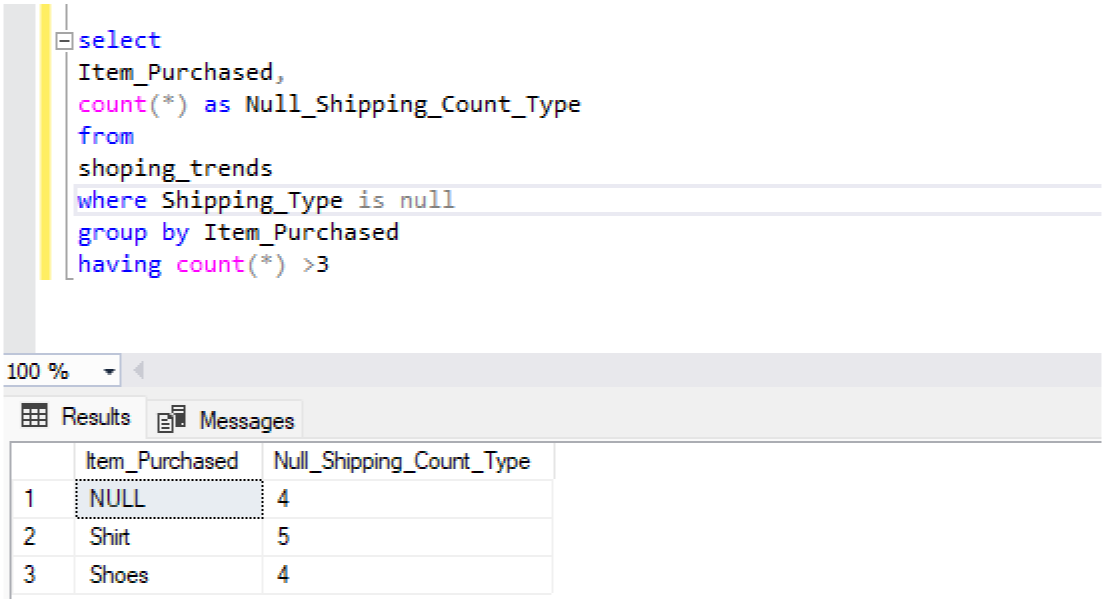
12. Group customers by Gender and Size, and count how many entries have a NUL L Color.



```
select
Gender,
Size,
COUNT(*) As Null_Color
from
shoping_trends
where Color is null
group by Gender, Size
```

	Gender	Size	Null_Color
1	Male	NULL	6
2	Male	L	6
3	Male	M	7
4	Male	S	5
5	Male	XL	5

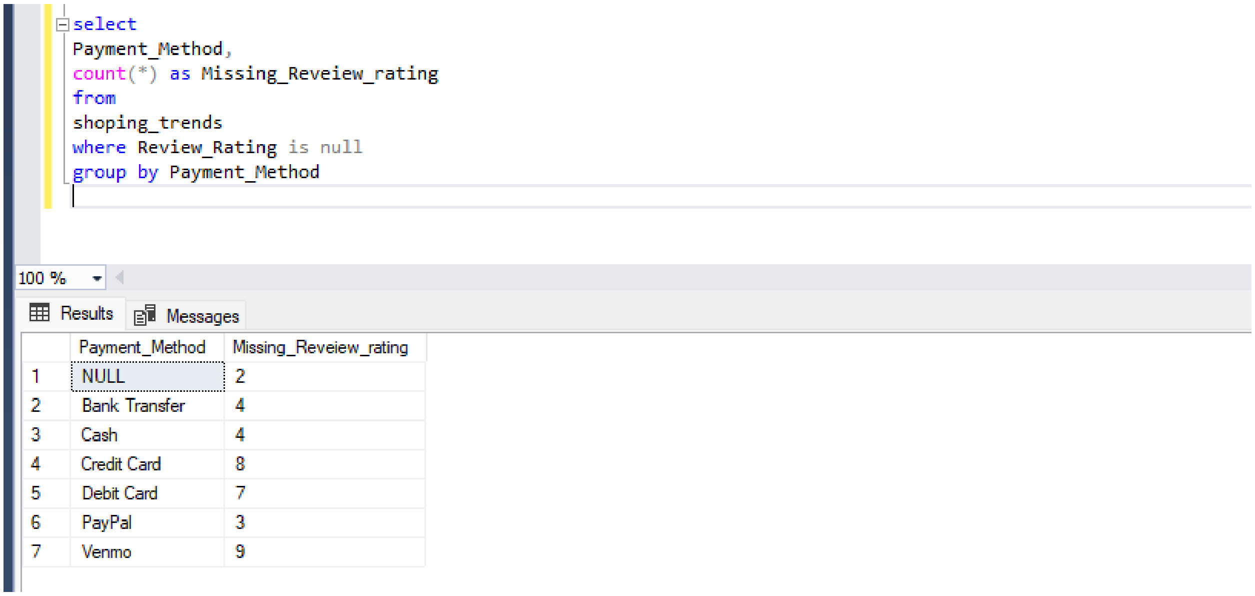
13. Identify all Item Purchased where more than 3 purchases had NULL Shipping Type.



```
select
Item_Purchased,
count(*) as Null_Shipping_Count_Type
from
shoping_trends
where Shipping_Type is null
group by Item_Purchased
having count(*) >3
```

	Item_Purchased	Null_Shipping_Count_Type
1	NULL	4
2	Shirt	5
3	Shoes	4

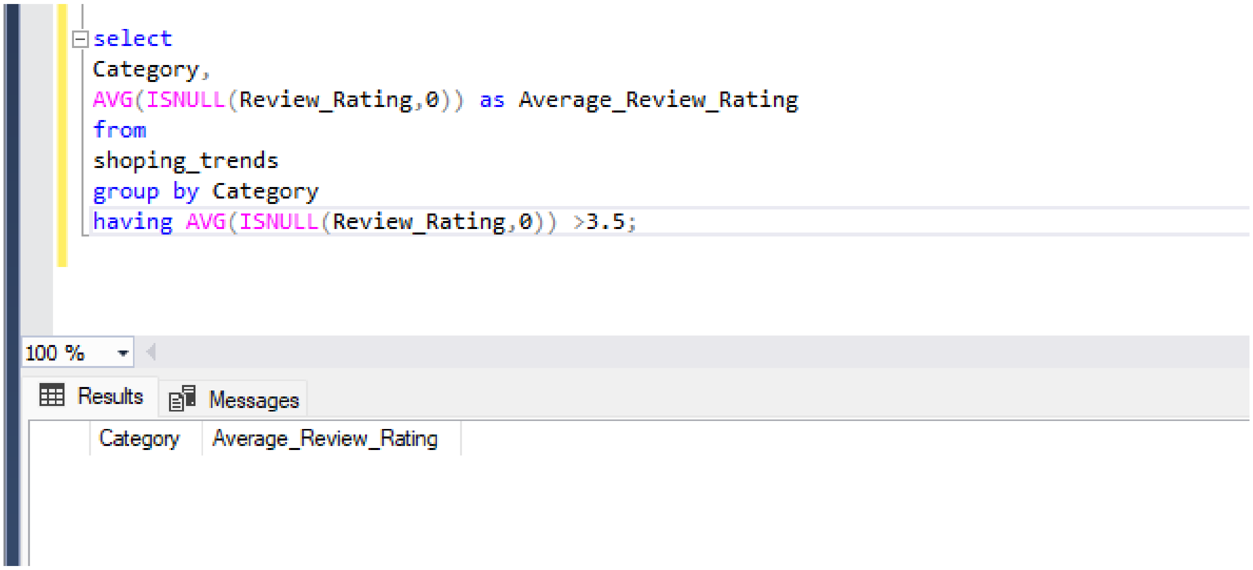
14. . Show a count of how many customers per Payment Method have NULL Review Rating.



```
select
Payment_Method,
count(*) as Missing_Reveiew_rating
from
shoping_trends
where Review_Rating is null
group by Payment_Method
```

	Payment_Method	Missing_Reveiew_rating
1	NULL	2
2	Bank Transfer	4
3	Cash	4
4	Credit Card	8
5	Debit Card	7
6	PayPal	3
7	Venmo	9

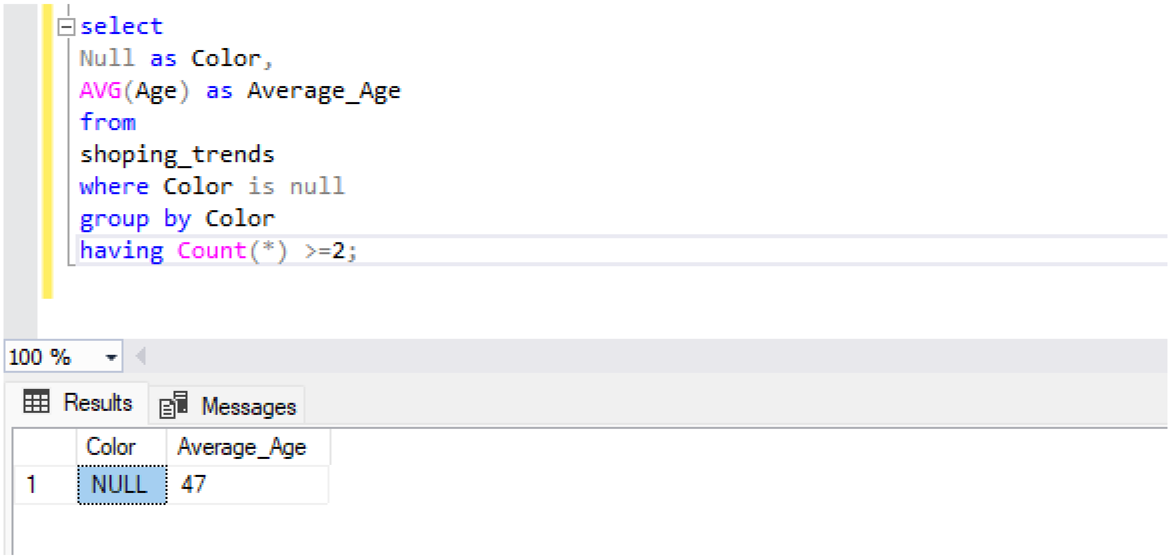
15. Group by Category and return the average Review Rating, replacing NULLs with 0, and filter only where average is greater than 3.5.



```
select
Category,
AVG(ISNULL(Review_Rating,0)) as Average_Review_Rating
from
shoping_trends
group by Category
having AVG(ISNULL(Review_Rating,0)) >3.5;
```

	Category	Average_Review_Rating

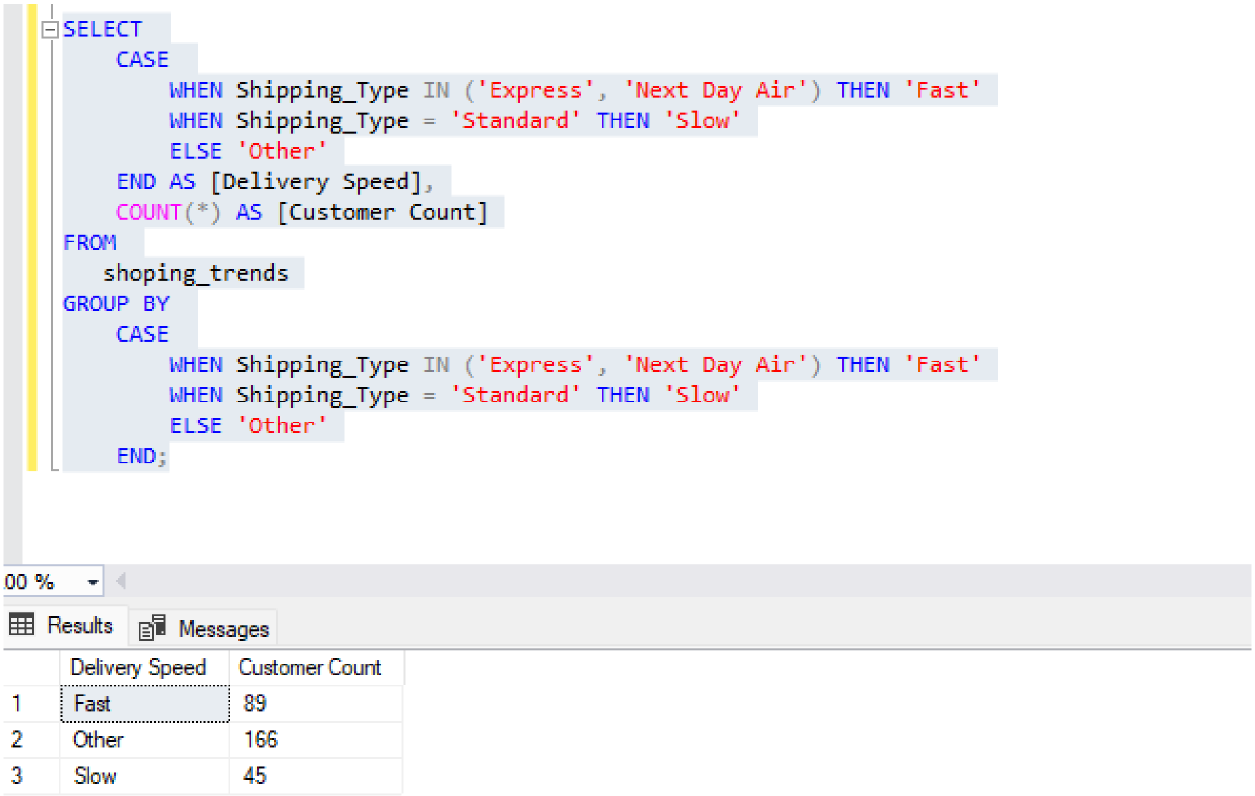
16. . List all Colors that are missing (NULL) in at least 2 rows and the average Age of customers for those rows.



```sql
select
Null as Color,
AVG(Age) as Average_Age
from
shoping_trends
where Color is null
group by Color
having Count(*) >=2;
```

	Color	Average_Age
1	NULL	47

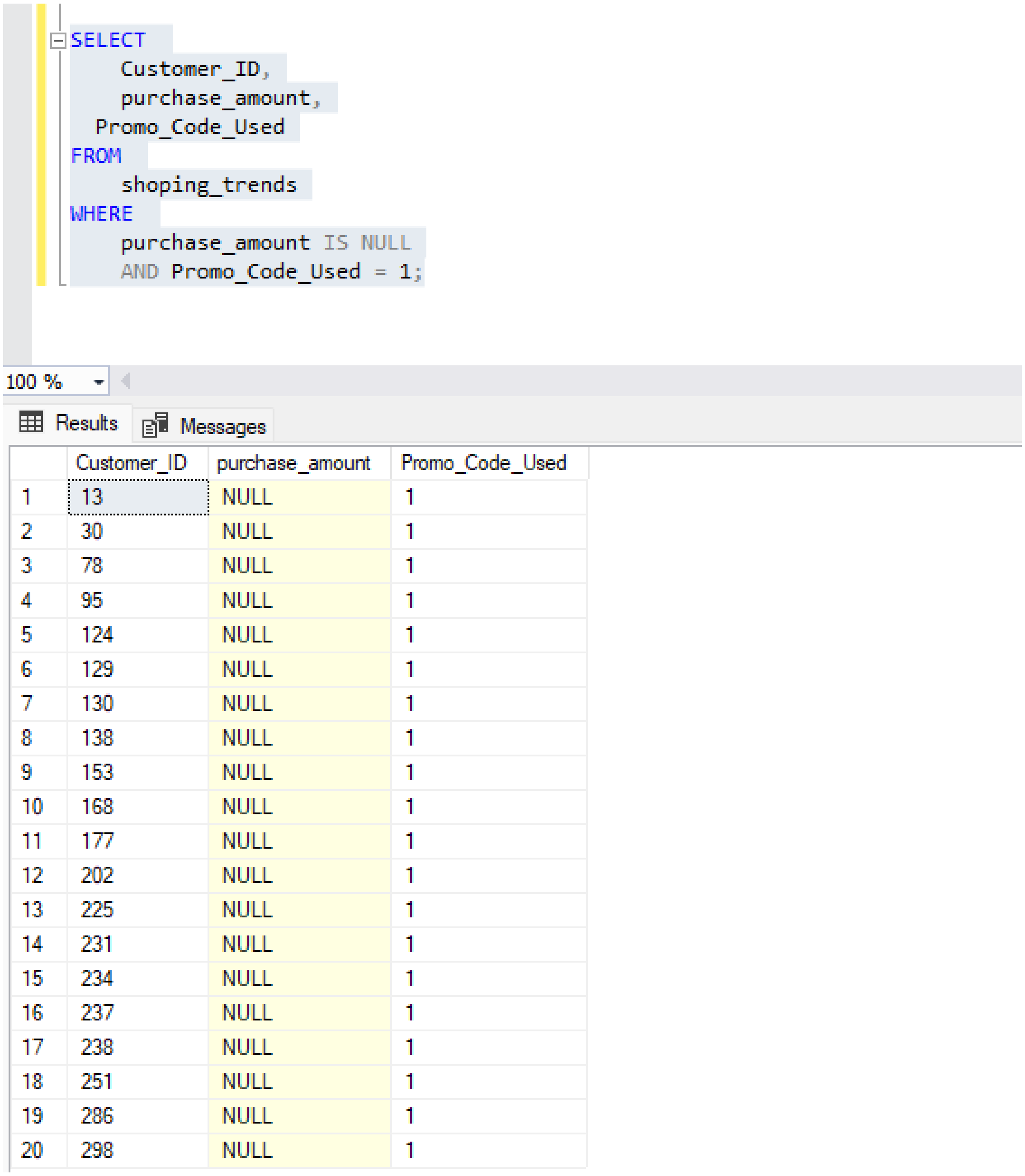
17. Use CASE to create a column Delivery Speed: 'Fast' if Shipping Type is 'Express' or 'Next Day Air', 'Slow' if 'Standard', 'Other' for all else including NULL. Then count how many customers fall into each category.



```sql
SELECT
    CASE
        WHEN Shipping_Type IN ('Express', 'Next Day Air') THEN 'Fast'
        WHEN Shipping_Type = 'Standard' THEN 'Slow'
        ELSE 'Other'
    END AS [Delivery Speed],
    COUNT(*) AS [Customer Count]
FROM
   shoping_trends
GROUP BY
    CASE
        WHEN Shipping_Type IN ('Express', 'Next Day Air') THEN 'Fast'
        WHEN Shipping_Type = 'Standard' THEN 'Slow'
        ELSE 'Other'
    END;
```

	Delivery Speed	Customer Count
1	Fast	89
2	Other	166
3	Slow	45

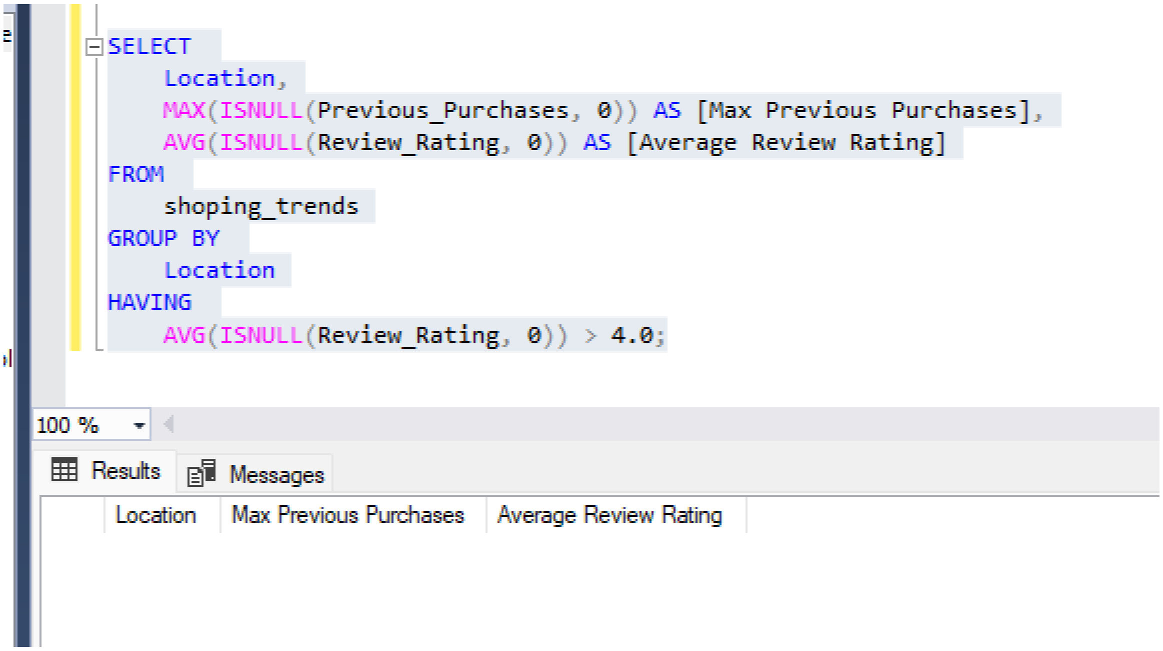
18. Find customers whose purchase_amount is NULL and whose Promo Code Used is 'Yes'



```sql
SELECT
    Customer_ID,
    purchase_amount,
  Promo_Code_Used
FROM
    shoping_trends
WHERE
    purchase_amount IS NULL
    AND Promo_Code_Used = 1;
```

	Customer_ID	purchase_amount	Promo_Code_Used
1	13	NULL	1
2	30	NULL	1
3	78	NULL	1
4	95	NULL	1
5	124	NULL	1
6	129	NULL	1
7	130	NULL	1
8	138	NULL	1
9	153	NULL	1
10	168	NULL	1
11	177	NULL	1
12	202	NULL	1
13	225	NULL	1
14	231	NULL	1
15	234	NULL	1
16	237	NULL	1
17	238	NULL	1
18	251	NULL	1
19	286	NULL	1
20	298	NULL	1

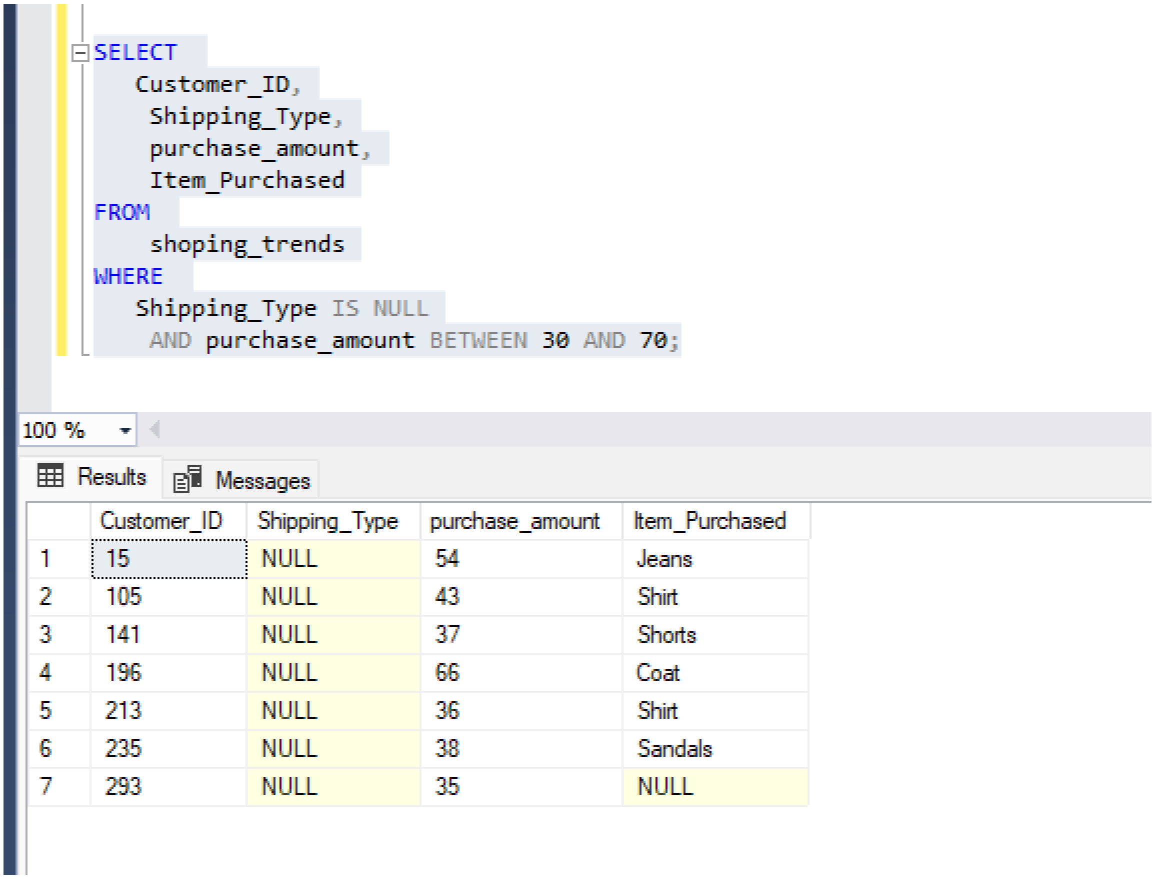
19. **Group by Location and show the maximum Previous Purchases, replacing NULLs with 0, only where the average rating is above 4.0.**



```
SELECT
    Location,
    MAX(ISNULL(Previous_Purchases, 0)) AS [Max Previous Purchases],
    AVG(ISNULL(Review_Rating, 0)) AS [Average Review Rating]
FROM
    shoping_trends
GROUP BY
    Location
HAVING
    AVG(ISNULL(Review_Rating, 0)) > 4.0;
```

Location	Max Previous Purchases	Average Review Rating

20. **Show customers who have a NULL Shipping Type but made a purchase in the range of 30 to 70 USD.**



```sql
SELECT
   Customer_ID,
    Shipping_Type,
    purchase_amount,
    Item_Purchased
FROM
    shoping_trends
WHERE
   Shipping_Type IS NULL
    AND purchase_amount BETWEEN 30 AND 70;
```

	Customer_ID	Shipping_Type	purchase_amount	Item_Purchased
1	15	NULL	54	Jeans
2	105	NULL	43	Shirt
3	141	NULL	37	Shorts
4	196	NULL	66	Coat
5	213	NULL	36	Shirt
6	235	NULL	38	Sandals
7	293	NULL	35	NULL