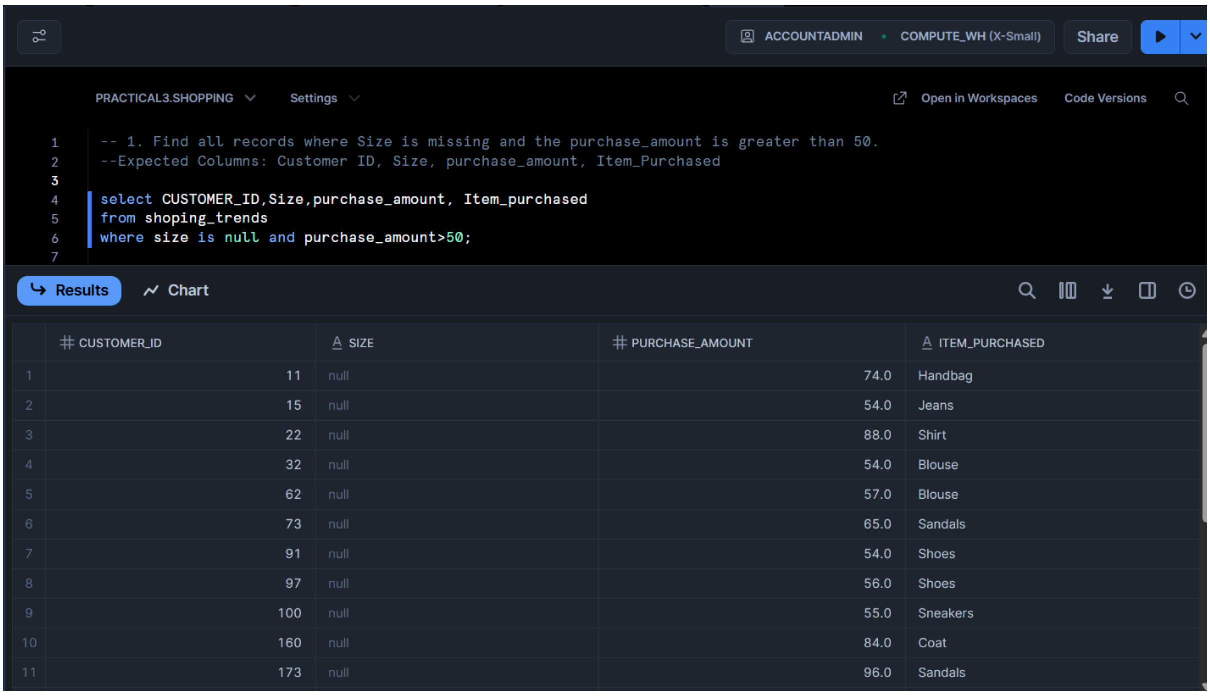
Q.1



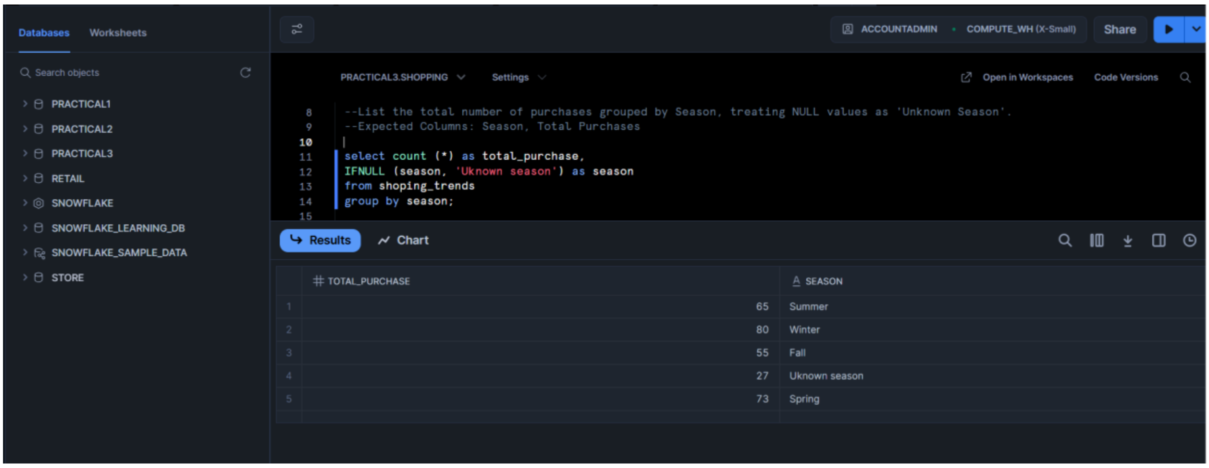
```sql
-- 1. Find all records where Size is missing and the purchase_amount is greater than 50.
--Expected Columns: Customer ID, Size, purchase_amount, Item_Purchased

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	74.0	Handbag
2	15	null	54.0	Jeans
3	22	null	88.0	Shirt
4	32	null	54.0	Blouse
5	62	null	57.0	Blouse
6	73	null	65.0	Sandals
7	91	null	54.0	Shoes
8	97	null	56.0	Shoes
9	100	null	55.0	Sneakers
10	160	null	84.0	Coat
11	173	null	96.0	Sandals

Q.2 Using IFNULL



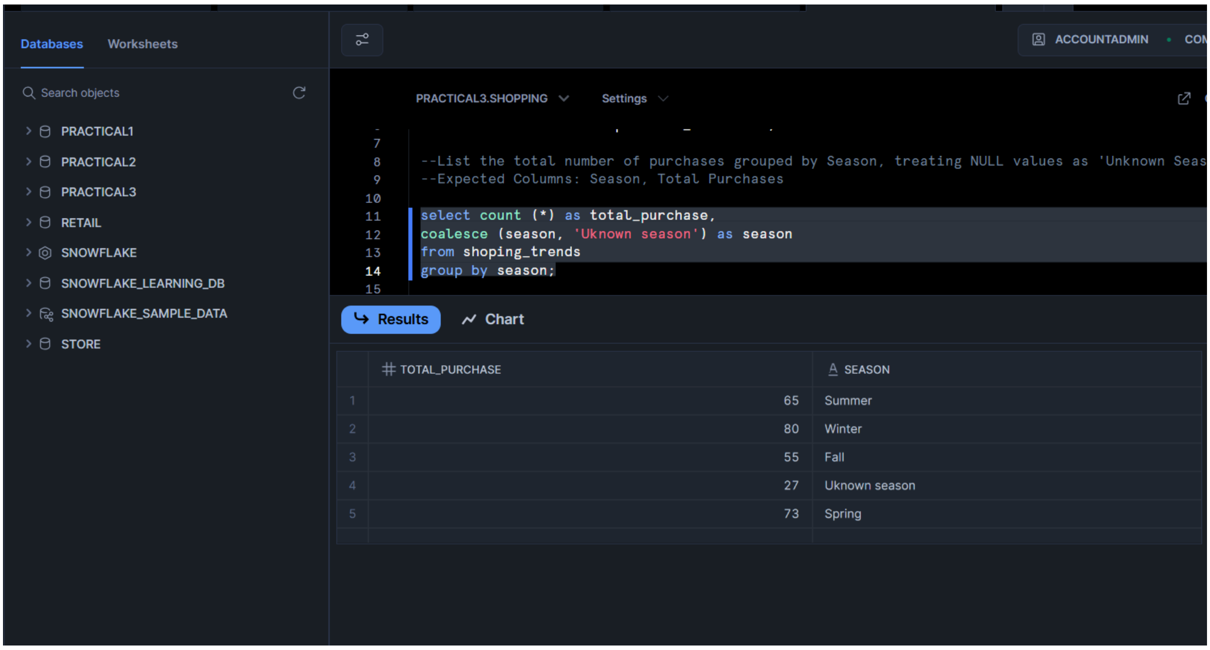
```sql
--List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'.
--Expected Columns: Season, Total Purchases

select count (*) as total_purchase,
IFNULL (season, 'Uknown season') as season
from shoping_trends
group by season;

```

	TOTAL_PURCHASE	SEASON
1	65	Summer
2	80	Winter
3	55	Fall
4	27	Uknown season
5	73	Spring

Q.2 Using Coalesce

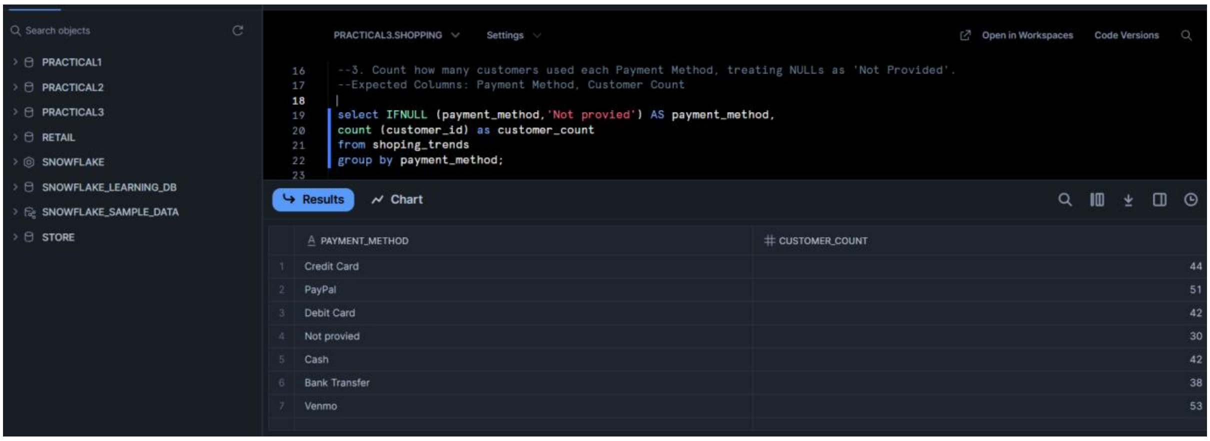


```
--List the total number of purchases grouped by Season, treating NULL values as 'Unknown Seas
--Expected Columns: Season, Total Purchases

select count (*) as total_purchase,
coalesce (season, 'Uknown season') as season
from shoping_trends
group by season;
```

	TOTAL_PURCHASE	SEASON
1	65	Summer
2	80	Winter
3	55	Fall
4	27	Uknown season
5	73	Spring

Q.3 Using IFNULL

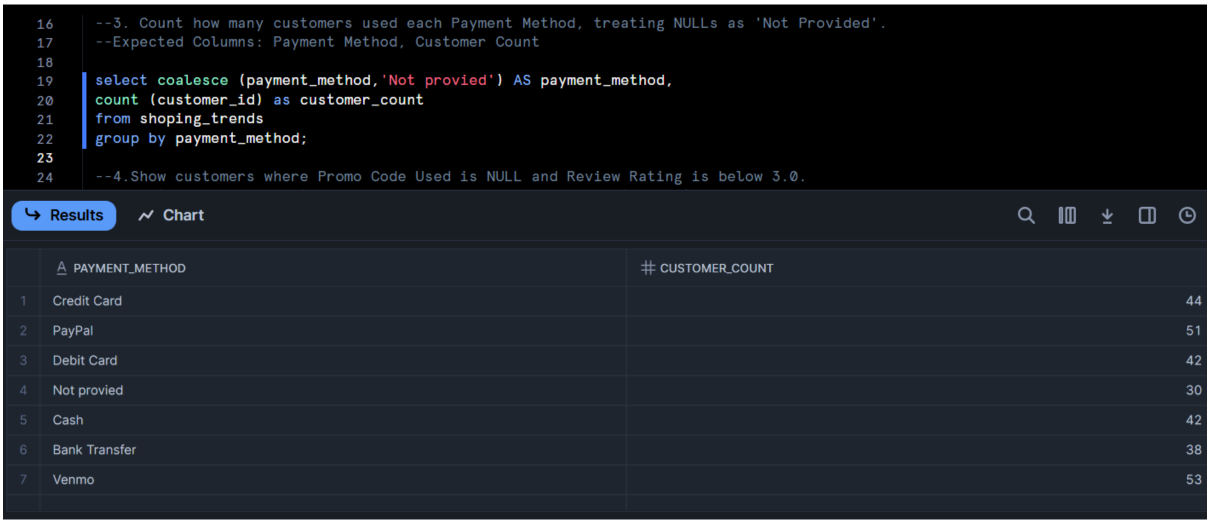


```
--3. Count how many customers used each Payment Method, treating NULLs as 'Not Provided'.
--Expected Columns: Payment Method, Customer Count

select IFNULL (payment_method,'Not provied') AS payment_method,
count (customer_id) as customer_count
from shoping_trends
group by payment_method;
```

	PAYMENT_METHOD	CUSTOMER_COUNT
1	Credit Card	44
2	PayPal	51
3	Debit Card	42
4	Not provied	30
5	Cash	42
6	Bank Transfer	38
7	Venmo	53

Q.3 Using Coalesce



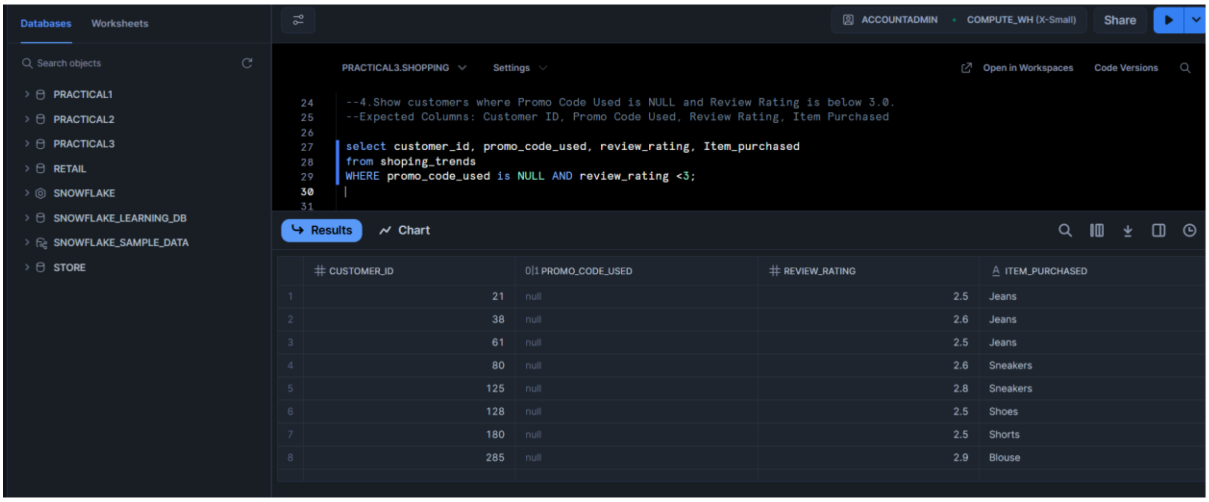
```
--3. Count how many customers used each Payment Method, treating NULLs as 'Not Provided'.
--Expected Columns: Payment Method, Customer Count

select coalesce (payment_method,'Not provied') AS payment_method,
count (customer_id) as customer_count
from shoping_trends
group by payment_method;

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

	PAYMENT_METHOD	CUSTOMER_COUNT
1	Credit Card	44
2	PayPal	51
3	Debit Card	42
4	Not provied	30
5	Cash	42
6	Bank Transfer	38
7	Venmo	53

Q.4

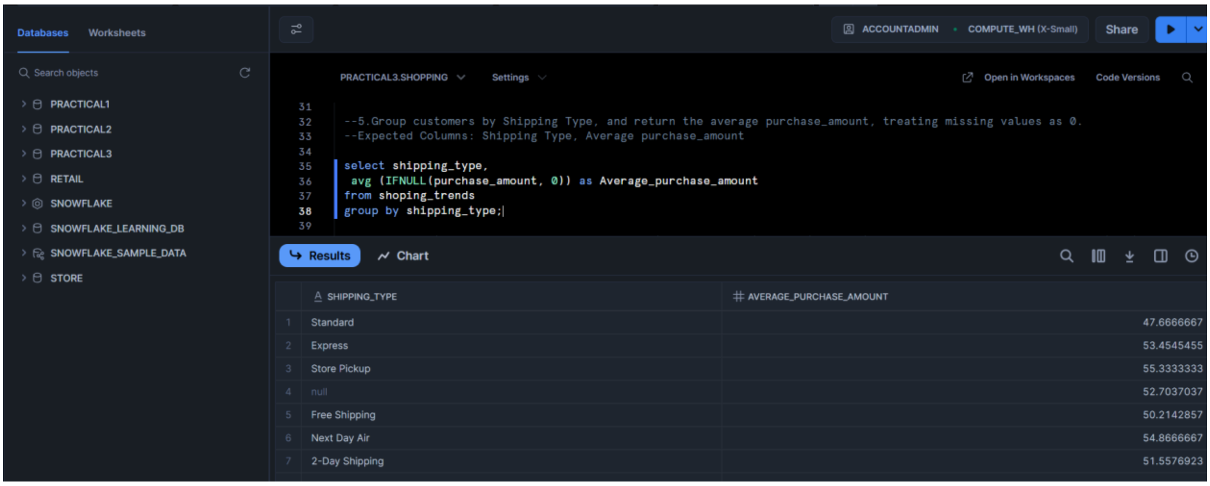


```sql
--4.Show customers where Promo Code Used is NULL and Review Rating is below 3.0.
--Expected Columns: Customer ID, Promo Code Used, Review Rating, Item Purchased

select customer_id, promo_code_used, review_rating, Item_purchased
from shoping_trends
WHERE promo_code_used is NULL AND review_rating <3;
```

	CUSTOMER_ID	PROMO_CODE_USED	REVIEW_RATING	ITEM_PURCHASED
1	21	null	2.5	Jeans
2	38	null	2.6	Jeans
3	61	null	2.5	Jeans
4	80	null	2.6	Sneakers
5	125	null	2.8	Sneakers
6	128	null	2.5	Shoes
7	180	null	2.5	Shorts
8	285	null	2.9	Blouse

Q.5 Using IfNull

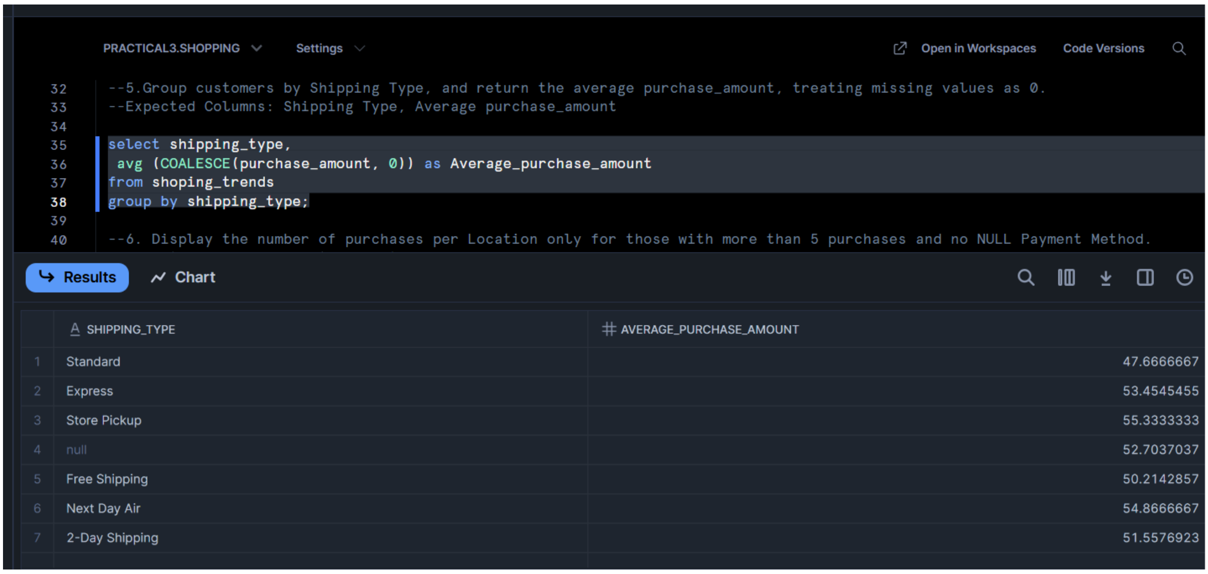


```sql
--5.Group customers by Shipping Type, and return the average purchase_amount, treating missing values as 0.
--Expected Columns: Shipping Type, Average purchase_amount

select shipping_type,
 avg (IFNULL(purchase_amount, 0)) as Average_purchase_amount
from shoping_trends
group by shipping_type;
```

	SHIPPING_TYPE	AVERAGE_PURCHASE_AMOUNT
1	Standard	47.6666667
2	Express	53.4545455
3	Store Pickup	55.3333333
4	null	52.7037037
5	Free Shipping	50.2142857
6	Next Day Air	54.8666667
7	2-Day Shipping	51.5576923

Q.5 Using Coalesce



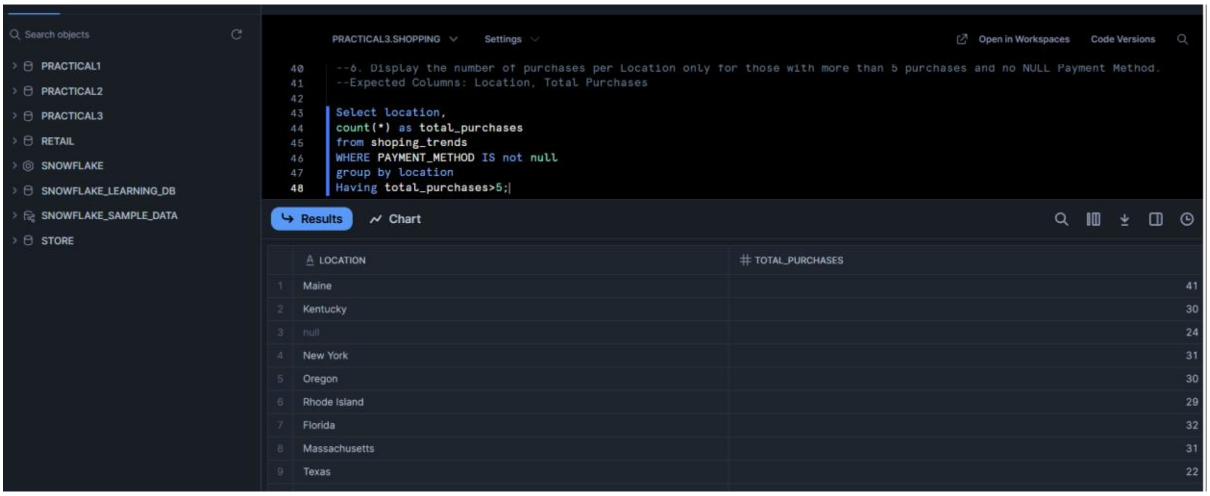
```sql
--5.Group customers by Shipping Type, and return the average purchase_amount, treating missing values as 0.
--Expected Columns: Shipping Type, Average purchase_amount

select shipping_type,
 avg (COALESCE(purchase_amount, 0)) as Average_purchase_amount
from shoping_trends
group by shipping_type;

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

	SHIPPING_TYPE	AVERAGE_PURCHASE_AMOUNT
1	Standard	47.6666667
2	Express	53.4545455
3	Store Pickup	55.3333333
4	null	52.7037037
5	Free Shipping	50.2142857
6	Next Day Air	54.8666667
7	2-Day Shipping	51.5576923

Q.6

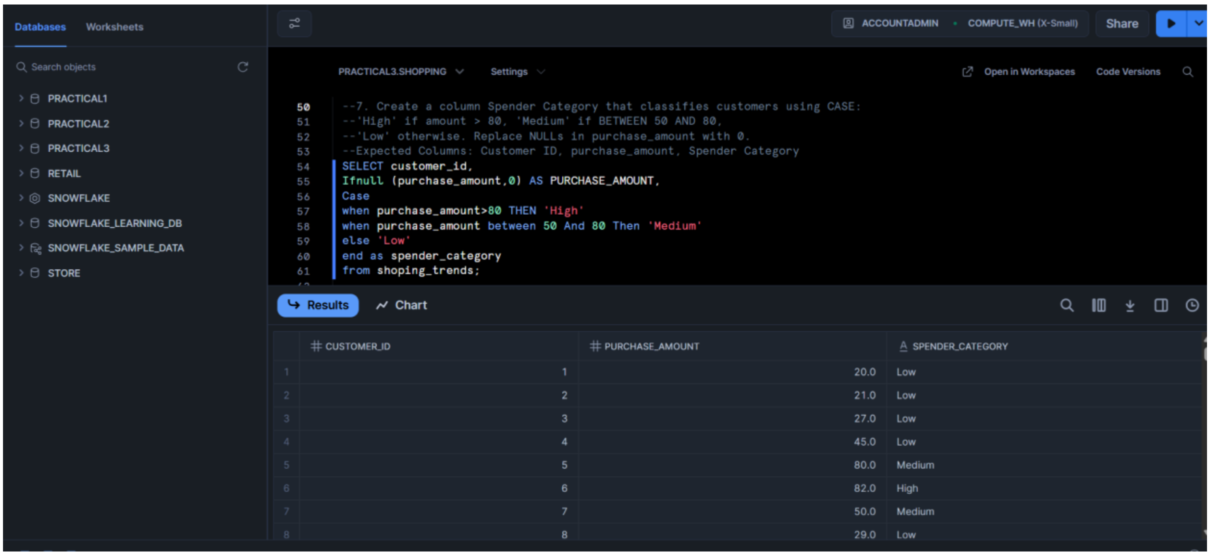


```
--6. Display the number of purchases per Location only for those with more than 5 purchases and no NULL Payment Method.
--Expected Columns: Location, Total Purchases

Select location,
count(*) as total_purchases
from shoping_trends
WHERE PAYMENT_METHOD IS not null
group by location
Having total_purchases>5;
```

	LOCATION	TOTAL_PURCHASES
1	Maine	41
2	Kentucky	30
3	null	24
4	New York	31
5	Oregon	30
6	Rhode Island	29
7	Florida	32
8	Massachusetts	31
9	Texas	22

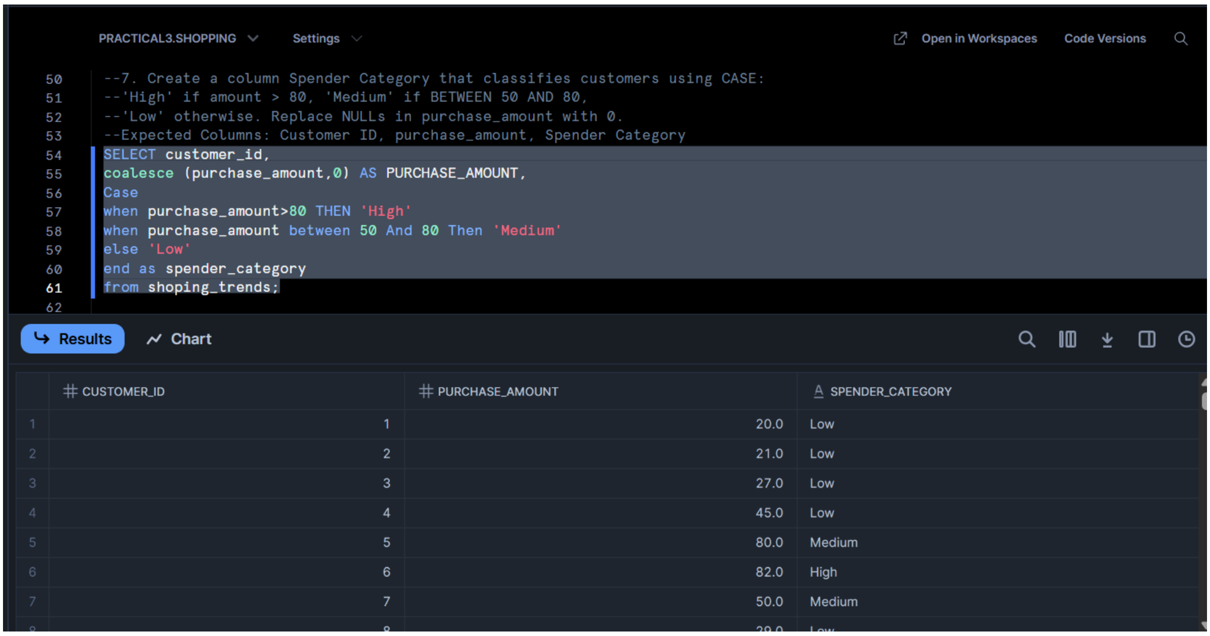
Q.7Using IFNULL



```
--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.
--Expected Columns: Customer ID, purchase_amount, Spender Category
SELECT customer_id,
Ifnull (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 as spender_category
from shoping_trends;
```

	CUSTOMER_ID	PURCHASE_AMOUNT	SPENDER_CATEGORY
1	1	20.0	Low
2	2	21.0	Low
3	3	27.0	Low
4	4	45.0	Low
5	5	80.0	Medium
6	6	82.0	High
7	7	50.0	Medium
8	8	29.0	Low

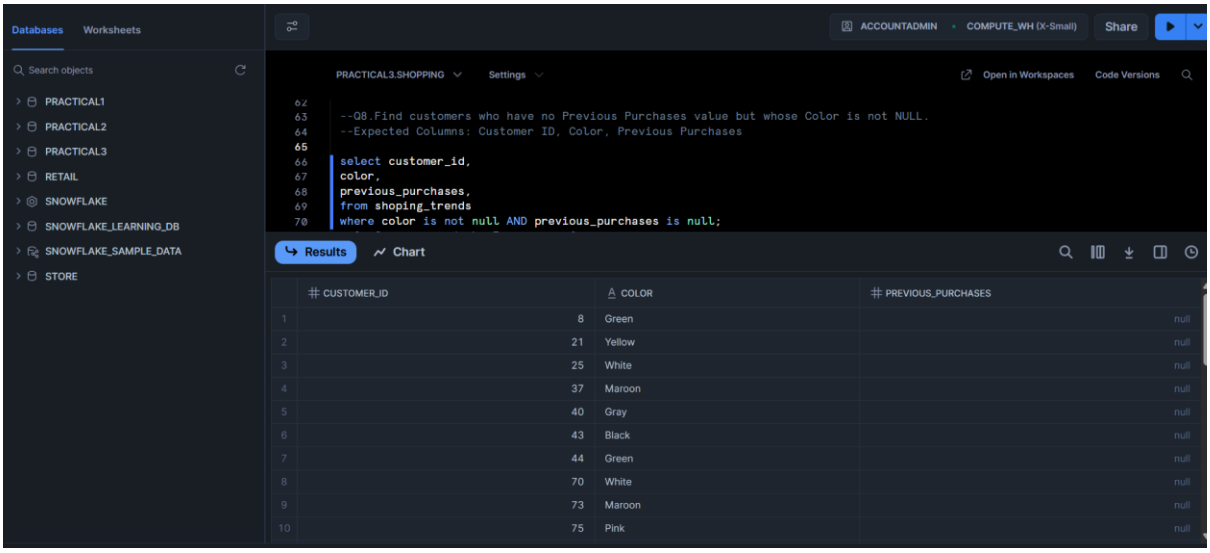
Q.7 Using Coalesce



```
--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.
--Expected Columns: Customer ID, purchase_amount, Spender Category
SELECT customer_id,
coalesce (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 as spender_category
from shoping_trends;
```

	CUSTOMER_ID	PURCHASE_AMOUNT	SPENDER_CATEGORY
1	1	20.0	Low
2	2	21.0	Low
3	3	27.0	Low
4	4	45.0	Low
5	5	80.0	Medium
6	6	82.0	High
7	7	50.0	Medium
8	8	29.0	Low

Q.8

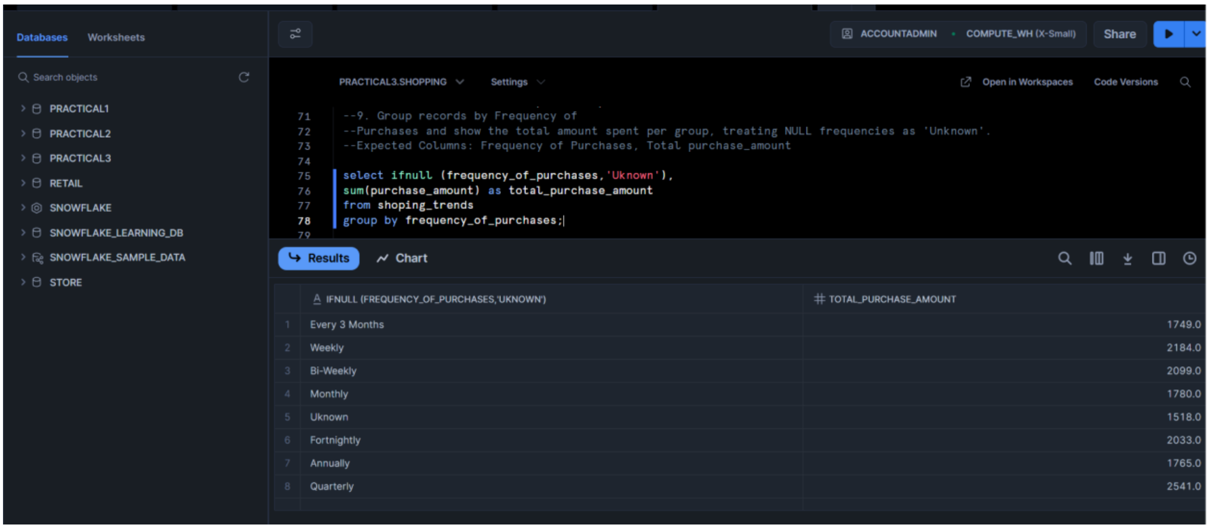


```
--Q8.Find customers who have no Previous Purchases value but whose Color is not NULL.
--Expected Columns: Customer ID, Color, Previous Purchases

select customer_id,
color,
previous_purchases,
from shoping_trends
where color is not null AND previous_purchases is 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

Q.9

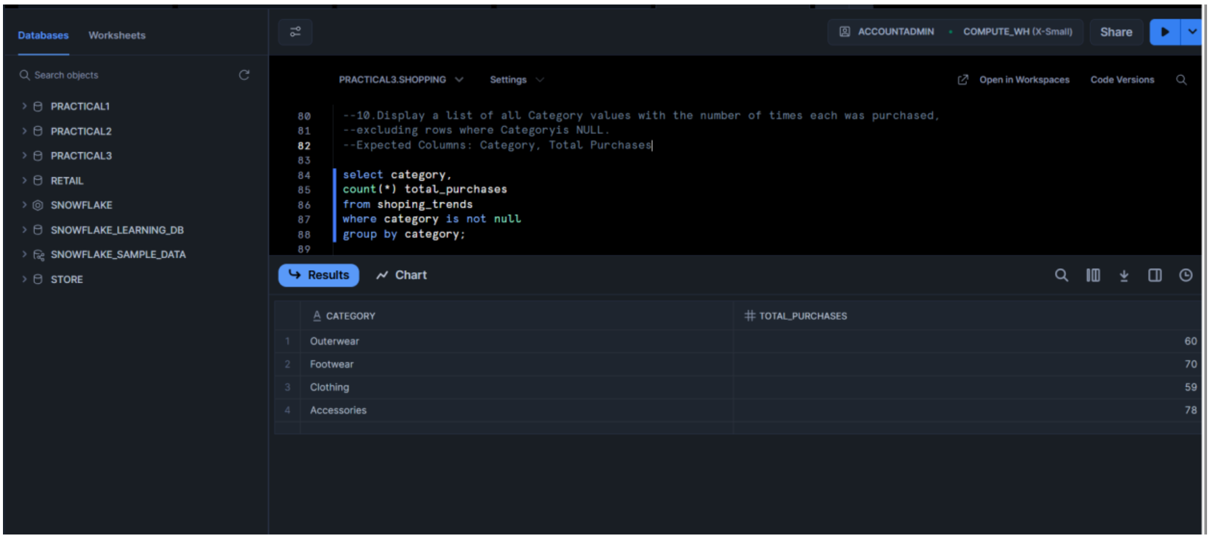


```
--9. Group records by Frequency of
--Purchases and show the total amount spent per group, treating NULL frequencies as 'Unknown'.
--Expected Columns: Frequency of Purchases, Total purchase_amount

select ifnull (frequency_of_purchases,'Uknown'),
sum(purchase_amount) as total_purchase_amount
from shoping_trends
group by frequency_of_purchases;
```

	IFNULL (FREQUENCY_OF_PURCHASES,'UKNOWN')	TOTAL_PURCHASE_AMOUNT
1	Every 3 Months	1749.0
2	Weekly	2184.0
3	Bi-Weekly	2099.0
4	Monthly	1780.0
5	Uknown	1518.0
6	Fortnightly	2033.0
7	Annually	1765.0
8	Quarterly	2541.0

Q.10

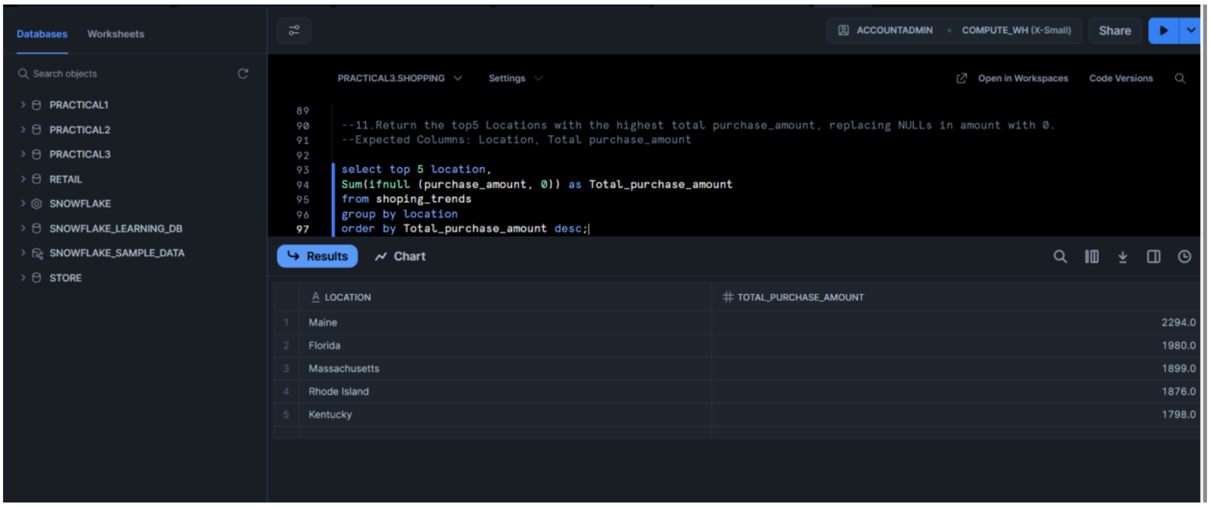


```
--10.Display a list of all Category values with the number of times each was purchased,
--excluding rows where Categoryis NULL.
--Expected Columns: Category, Total Purchases

select category,
count(*) total_purchases
from shoping_trends
where category is not null
group by category;
```

	CATEGORY	TOTAL_PURCHASES
1	Outerwear	60
2	Footwear	70
3	Clothing	59
4	Accessories	78

Q.11

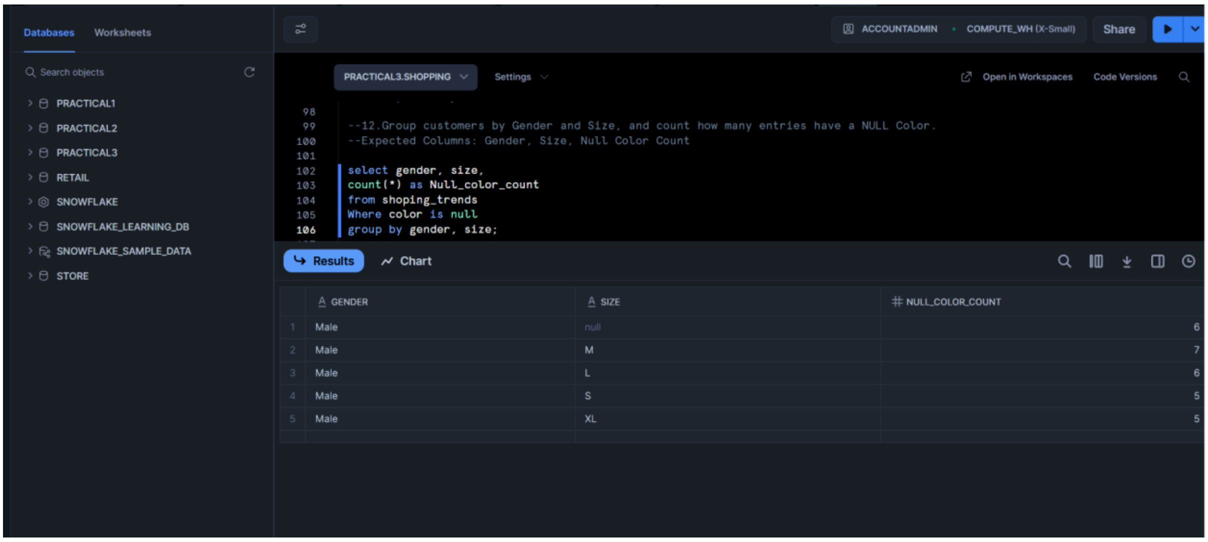


```
--11.Return the top5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0.
--Expected Columns: Location, Total purchase_amount

select top 5 location,
Sum(ifnull (purchase_amount, 0)) as Total_purchase_amount
from shoping_trends
group by location
order by Total_purchase_amount desc;
```

	LOCATION	TOTAL_PURCHASE_AMOUNT
1	Maine	2294.0
2	Florida	1980.0
3	Massachusetts	1899.0
4	Rhode Island	1876.0
5	Kentucky	1798.0

Q.12

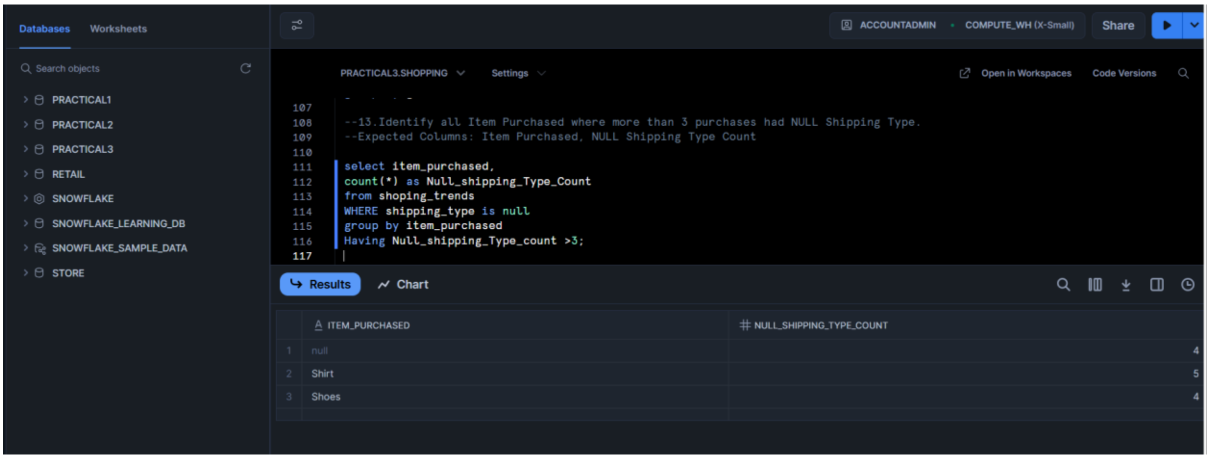


```sql
--12.Group customers by Gender and Size, and count how many entries have a NULL Color.
--Expected Columns: Gender, Size, Null Color Count

select gender, size,
count(*) as Null_color_count
from shoping_trends
Where color is null
group by gender, size;
```

	GENDER	SIZE	NULL_COLOR_COUNT
1	Male	null	6
2	Male	M	7
3	Male	L	6
4	Male	S	5
5	Male	XL	5

Q13.

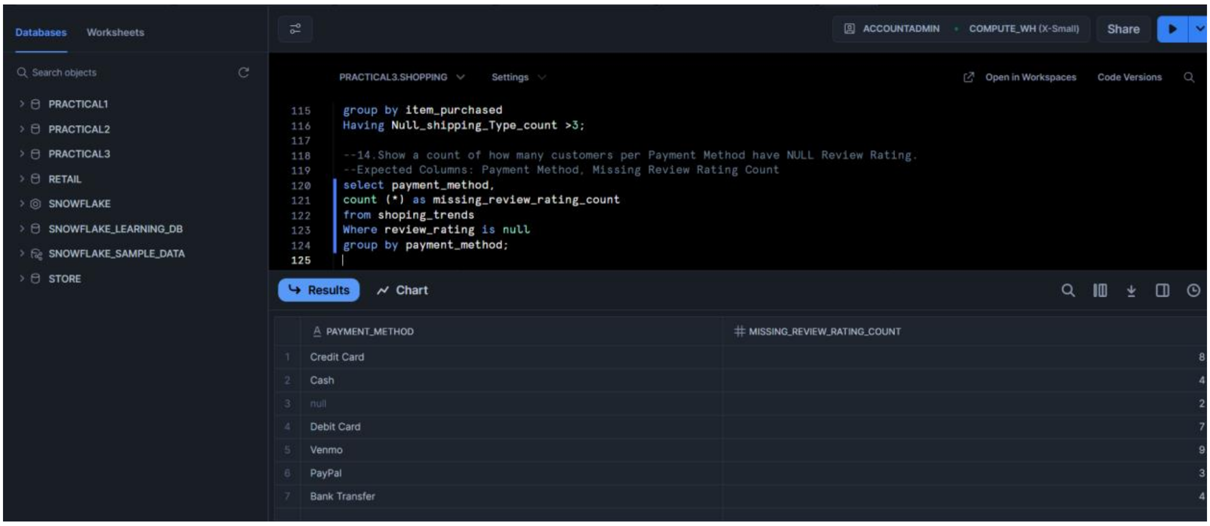


```sql
--13.Identify all Item Purchased where more than 3 purchases had NULL Shipping Type.
--Expected Columns: Item Purchased, NULL Shipping Type Count

select item_purchased,
count(*) as Null_shipping_Type_Count
from shoping_trends
WHERE shipping_type is null
group by item_purchased
Having Null_shipping_Type_count >3;
```

	ITEM_PURCHASED	NULL_SHIPPING_TYPE_COUNT
1	null	4
2	Shirt	5
3	Shoes	4

Q.14

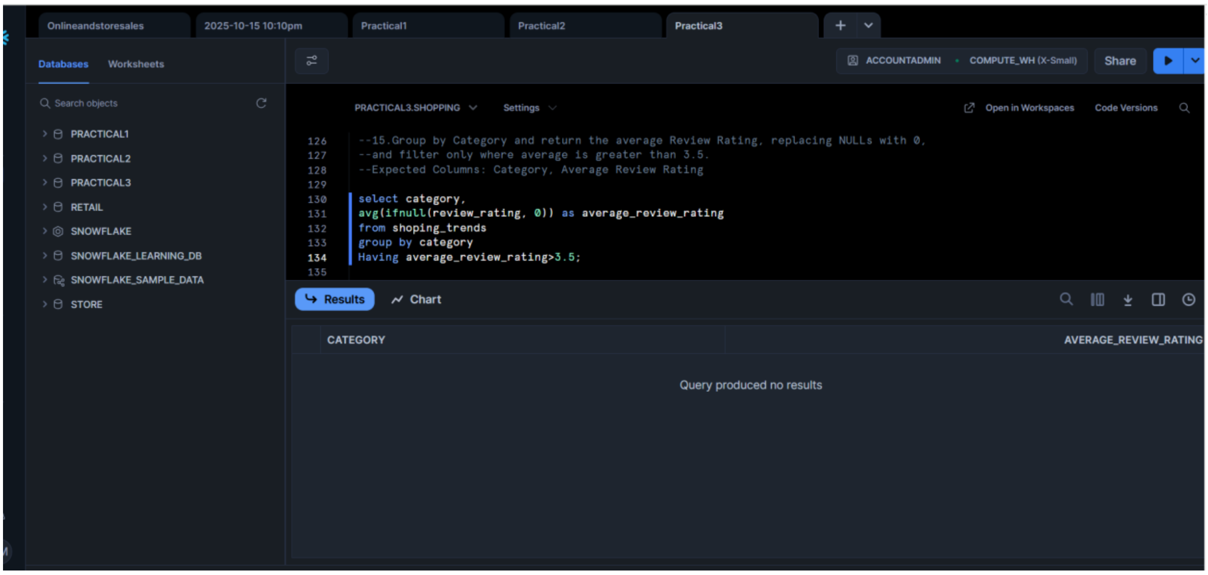


```sql
group by item_purchased
Having Null_shipping_Type_count >3;

--14.Show a count of how many customers per Payment Method have NULL Review Rating.
--Expected Columns: Payment Method, Missing Review Rating Count
select payment_method,
count (*) as missing_review_rating_count
from shoping_trends
Where review_rating is null
group by payment_method;
```

	PAYMENT_METHOD	MISSING_REVIEW_RATING_COUNT
1	Credit Card	8
2	Cash	4
3	null	2
4	Debit Card	7
5	Venmo	9
6	PayPal	3
7	Bank Transfer	4

Q.15 Using ifnull



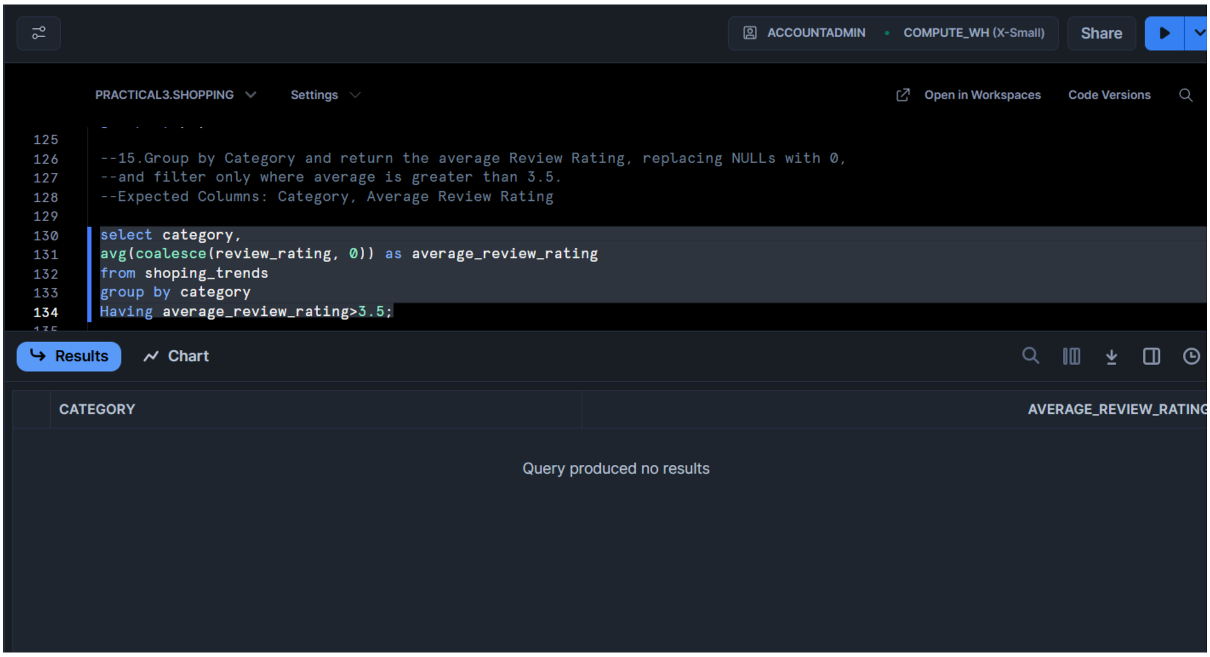
```sql
--15.Group by Category and return the average Review Rating, replacing NULLs with 0,
--and filter only where average is greater than 3.5.
--Expected Columns: Category, Average Review Rating

select category,
avg(ifnull(review_rating, 0)) as average_review_rating
from shoping_trends
group by category
Having average_review_rating>3.5;
```

CATEGORY	AVERAGE_REVIEW_RATING

Query produced no results

Q.15 Using Coalesce



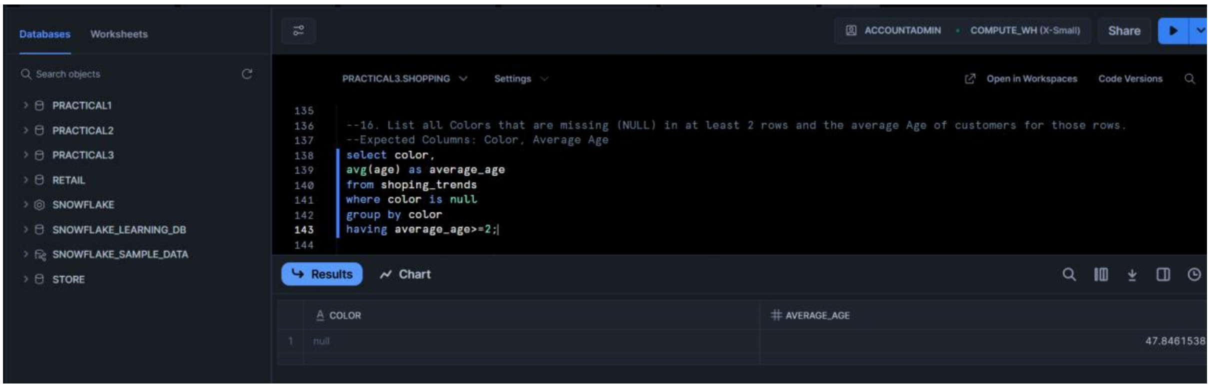
```sql
--15.Group by Category and return the average Review Rating, replacing NULLs with 0,
--and filter only where average is greater than 3.5.
--Expected Columns: Category, Average Review Rating

select category,
avg(coalesce(review_rating, 0)) as average_review_rating
from shoping_trends
group by category
Having average_review_rating>3.5;
```

CATEGORY	AVERAGE_REVIEW_RATING

Query produced no results

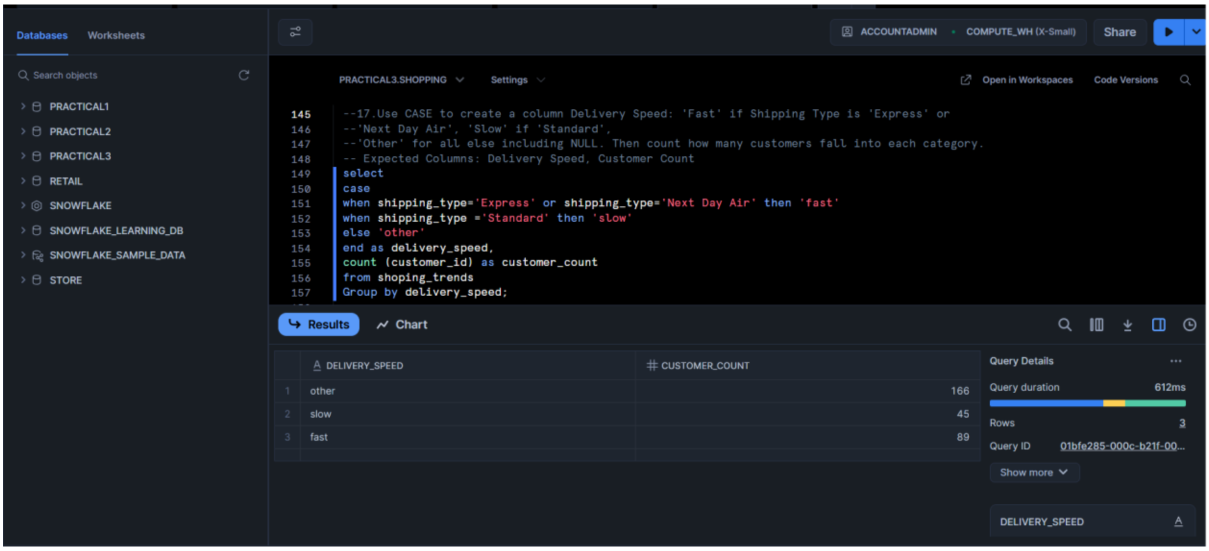
Q.16



```sql
--16. List all Colors that are missing (NULL) in at least 2 rows and the average Age of customers for those rows.
--Expected Columns: Color, Average Age
select color,
avg(age) as average_age
from shoping_trends
where color is null
group by color
having average_age>=2;
```

	COLOR	AVERAGE_AGE
1	null	47.8461538

Q.17



```sql
--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.
-- Expected Columns: Delivery Speed, Customer Count
select
case
when shipping_type='Express' or shipping_type='Next Day Air' then 'fast'
when shipping_type ='Standard' then 'slow'
else 'other'
end as delivery_speed,
count (customer_id) as customer_count
from shoping_trends
Group by delivery_speed;
```

	DELIVERY_SPEED	CUSTOMER_COUNT
1	other	166
2	slow	45
3	fast	89

Query Details

Query duration 612ms

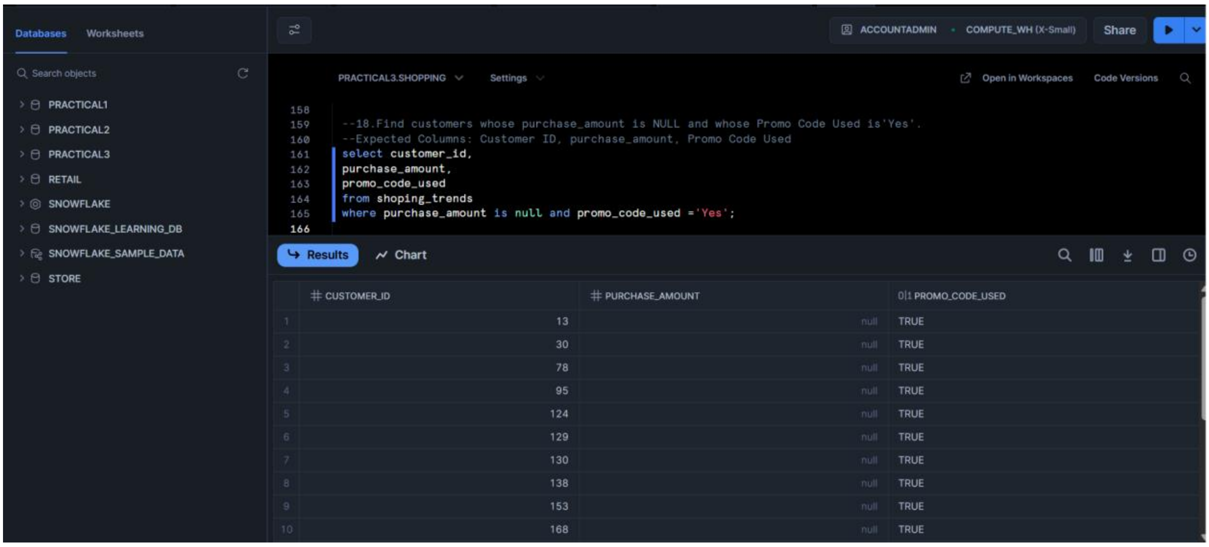
Rows 3

Query ID 01bfe285-000c-b21f-00...

Show more

DELIVERY_SPEED

Q.18



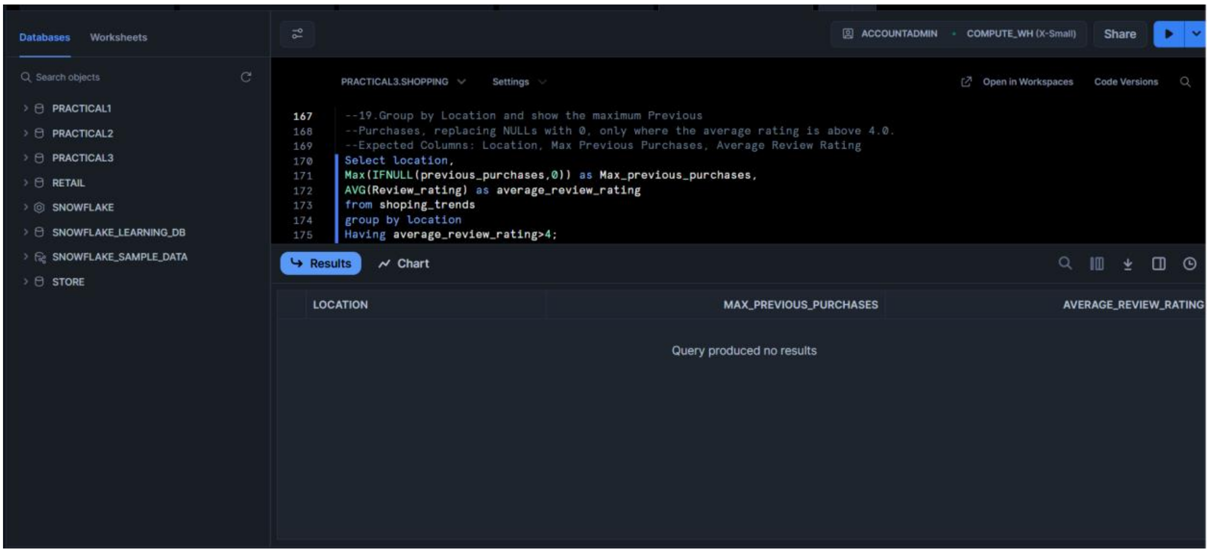
```sql

--18.Find customers whose purchase_amount is NULL and whose Promo Code Used is'Yes'.
--Expected Columns: Customer ID, purchase_amount, Promo Code Used
select customer_id,
purchase_amount,
promo_code_used
from shoping_trends
where purchase_amount is null and promo_code_used ='Yes';

```

	CUSTOMER_ID	PURCHASE_AMOUNT	PROMO_CODE_USED
1	13	null	TRUE
2	30	null	TRUE
3	78	null	TRUE
4	95	null	TRUE
5	124	null	TRUE
6	129	null	TRUE
7	130	null	TRUE
8	138	null	TRUE
9	153	null	TRUE
10	168	null	TRUE

Q.19

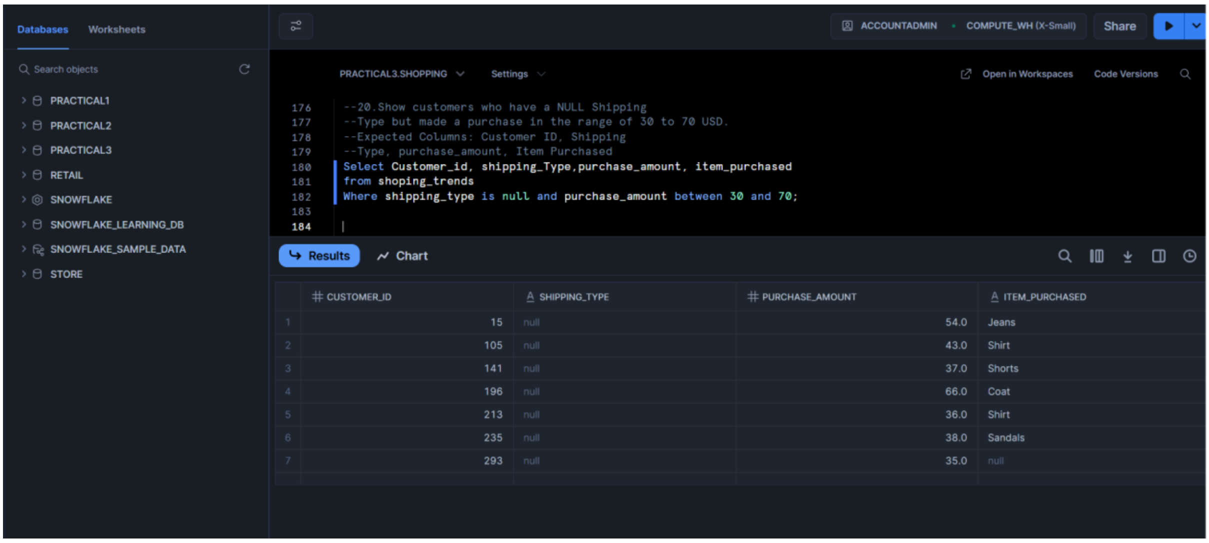


```
--19.Group by Location and show the maximum Previous
--Purchases, replacing NULLs with 0, only where the average rating is above 4.0.
--Expected Columns: Location, Max Previous Purchases, Average Review Rating
Select location,
Max(IFNULL(previous_purchases,0)) as Max_previous_purchases,
AVG(Review_rating) as average_review_rating
from shoping_trends
group by location
Having average_review_rating>4;
```

LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE_REVIEW_RATING

Query produced no results

Q.20



```
--20.Show customers who have a NULL Shipping
--Type but made a purchase in the range of 30 to 70 USD.
--Expected Columns: Customer ID, Shipping
--Type, purchase_amount, Item Purchased
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	54.0	Jeans
2	105	null	43.0	Shirt
3	141	null	37.0	Shorts
4	196	null	66.0	Coat
5	213	null	36.0	Shirt
6	235	null	38.0	Sandals
7	293	null	35.0	null