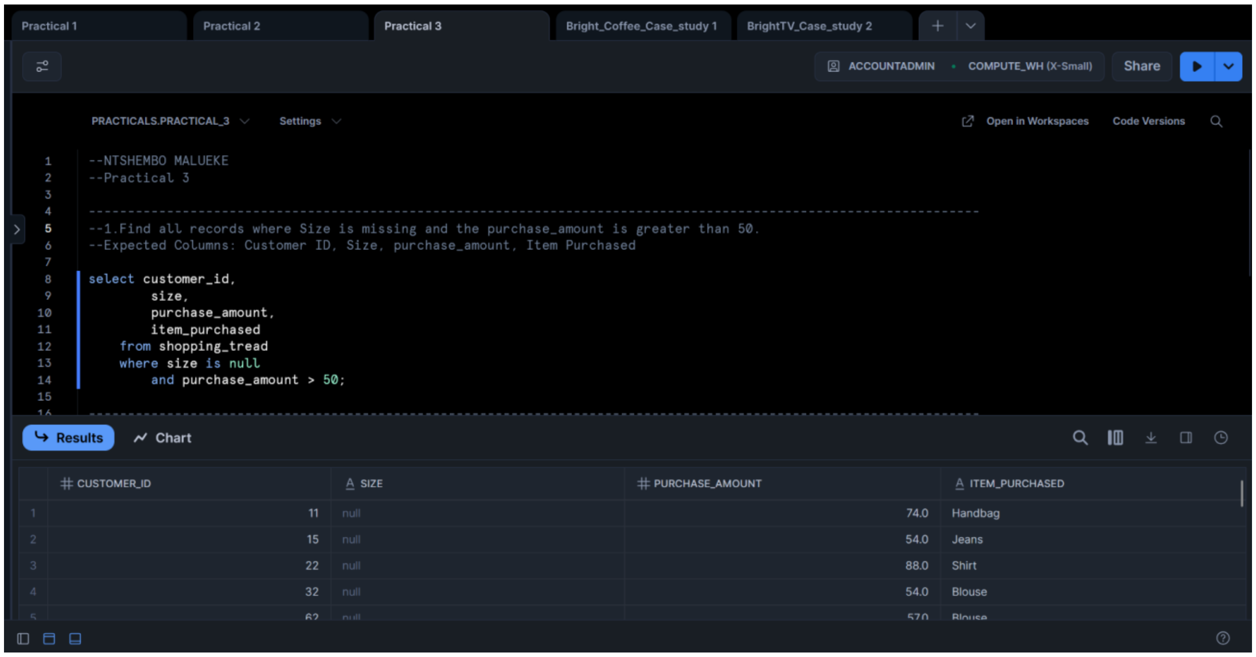
--NTSHEMBO MALUEKE

--Practical 3

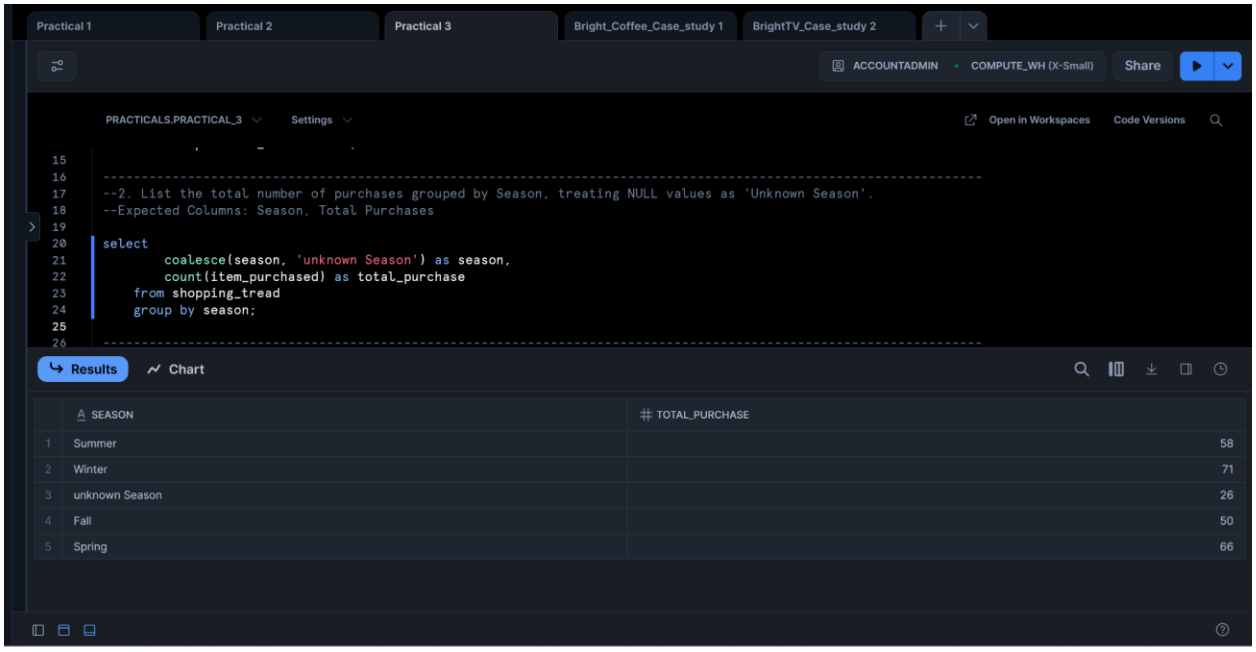
Q1



```
--NTSHEMBO MALUEKE
--Practical 3

------------------------------------------------------------------------------------------
--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 shopping_tread
    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

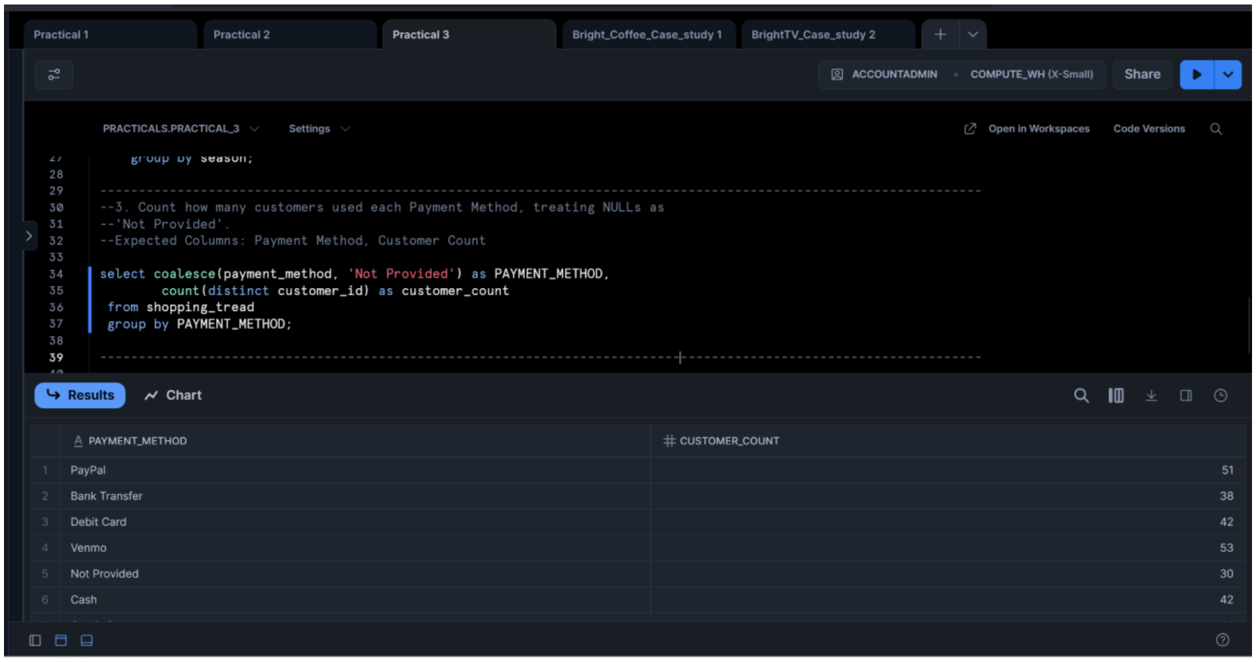
Q2



```
------------------------------------------------------------------------------------------
--2. List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'.
--Expected Columns: Season, Total Purchases

select
        coalesce(season, 'unknown Season') as season,
        count(item_purchased) as total_purchase
    from shopping_tread
    group by season;
------------------------------------------------------------------------------------------
```

	SEASON	TOTAL_PURCHASE
1	Summer	58
2	Winter	71
3	unknown Season	26
4	Fall	50
5	Spring	66

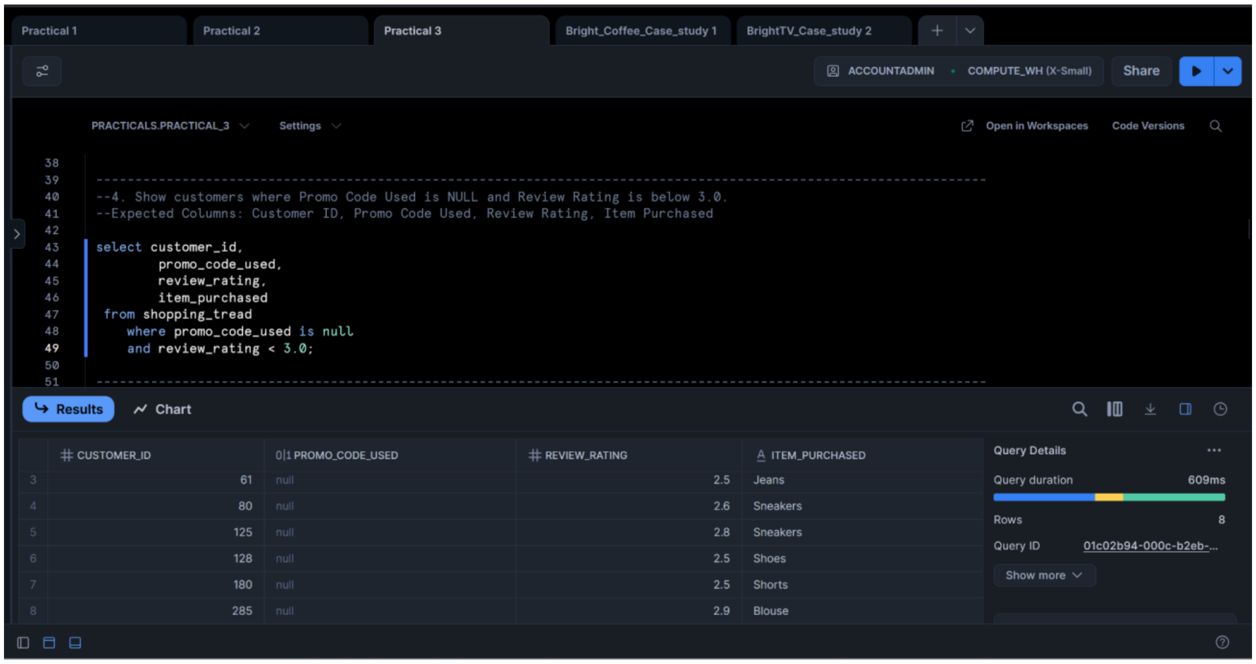
Q3



```sql
    group by season;

------------------------------------------------------------------------------------------
--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 Provided') as PAYMENT_METHOD,
        count(distinct customer_id) as customer_count
 from shopping_tread
 group by PAYMENT_METHOD;

------------------------------------------------------------------------------------------
```

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

Q4



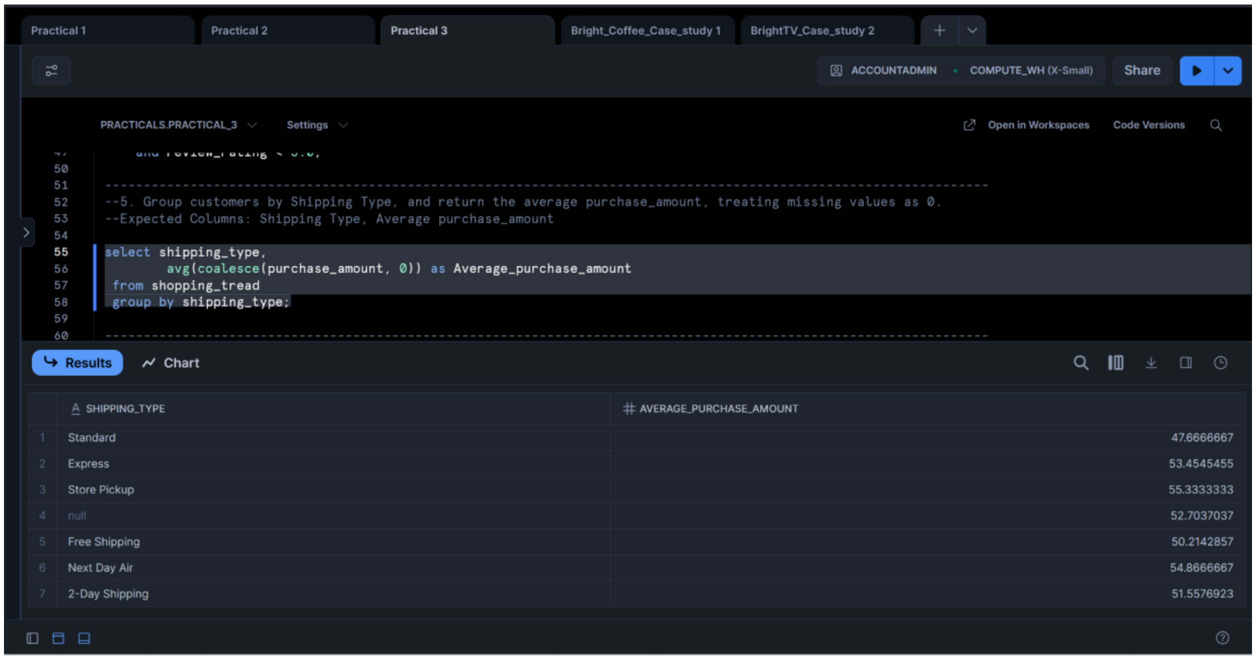
```sql

------------------------------------------------------------------------------------------
--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 shopping_tread
    where promo_code_used is null
    and review_rating < 3.0;

------------------------------------------------------------------------------------------
```

	CUSTOMER_ID	PROMO_CODE_USED	REVIEW_RATING	ITEM_PURCHASED
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

Query Details

Query duration	609ms
Rows	8
Query ID	01c02b94-000c-b2eb-...

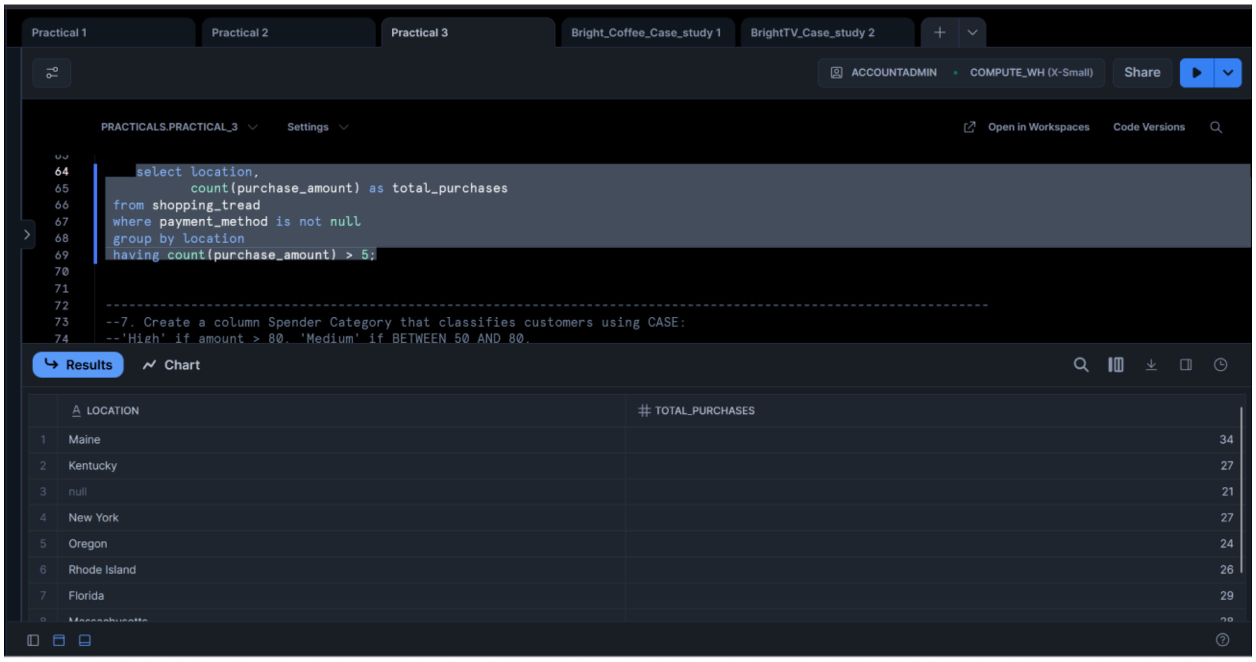
Q5



```
--------------------------------------------------------------------------------------------------------------------
--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 shopping_tread
 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

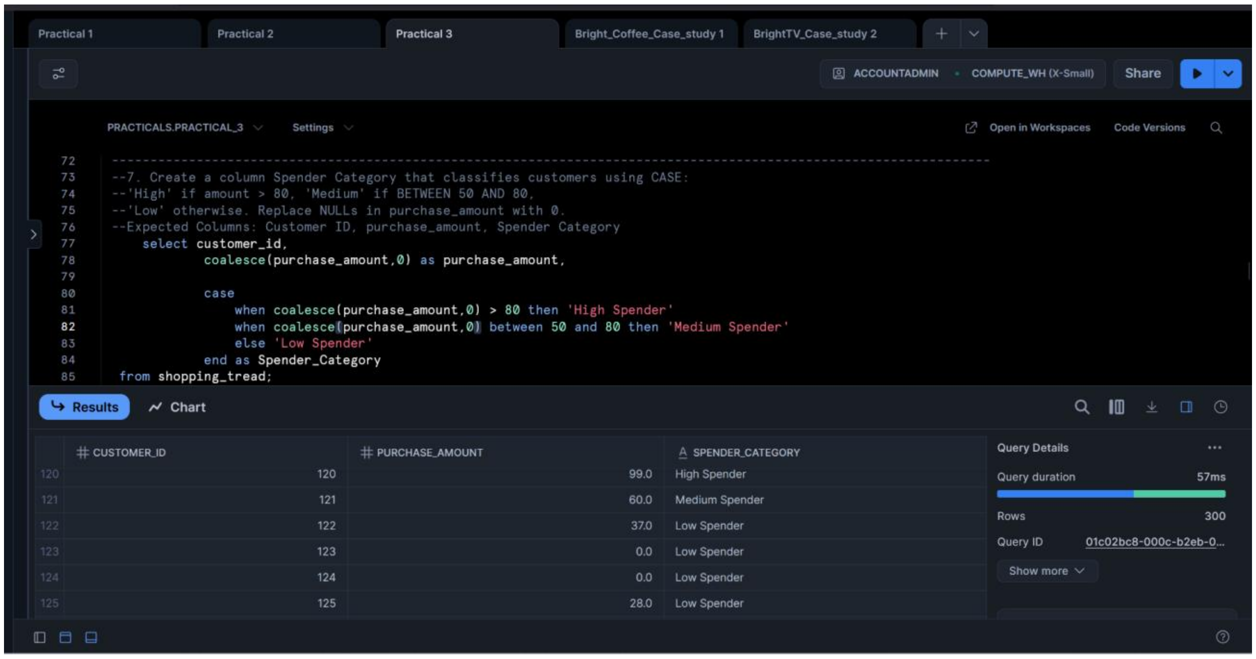
Q6



```
    select location,
           count(purchase_amount) as total_purchases
 from shopping_tread
 where payment_method is not null
 group by location
 having count(purchase_amount) > 5;



--------------------------------------------------------------------------------------------------------------------
--7. Create a column Spender Category that classifies customers using CASE:
--'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80,
```

	LOCATION	TOTAL_PURCHASES
1	Maine	34
2	Kentucky	27
3	null	21
4	New York	27
5	Oregon	24
6	Rhode Island	26
7	Florida	29

Q7

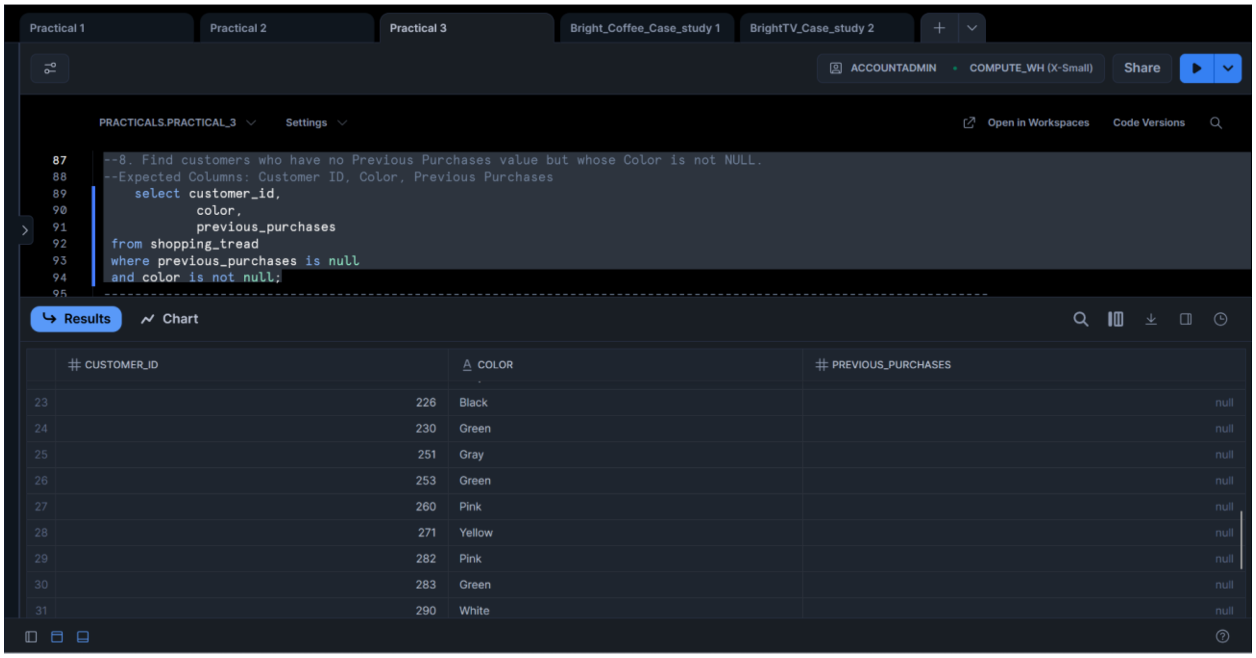


```
------------------------------------------------------------------------------------------------------------------------
--7. Create a column Spender Category that classifies customers using CASE:
--'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80,
--'Low' otherwise. Replace NULLs in purchase_amount with 0.
--Expected Columns: Customer ID, purchase_amount, Spender Category
    select customer_id,
            coalesce(purchase_amount,0) as purchase_amount,

            case
                when coalesce(purchase_amount,0) > 80 then 'High Spender'
                when coalesce(purchase_amount,0) between 50 and 80 then 'Medium Spender'
                else 'Low Spender'
            end as Spender_Category
 from shopping_tread;
```

	CUSTOMER_ID	PURCHASE_AMOUNT	SPENDER_CATEGORY
120	120	99.0	High Spender
121	121	60.0	Medium Spender
122	122	37.0	Low Spender
123	123	0.0	Low Spender
124	124	0.0	Low Spender
125	125	28.0	Low Spender

Query Details
Query duration 57ms
Rows 300
Query ID 01c02bc8-000c-b2eb-0...
Show more

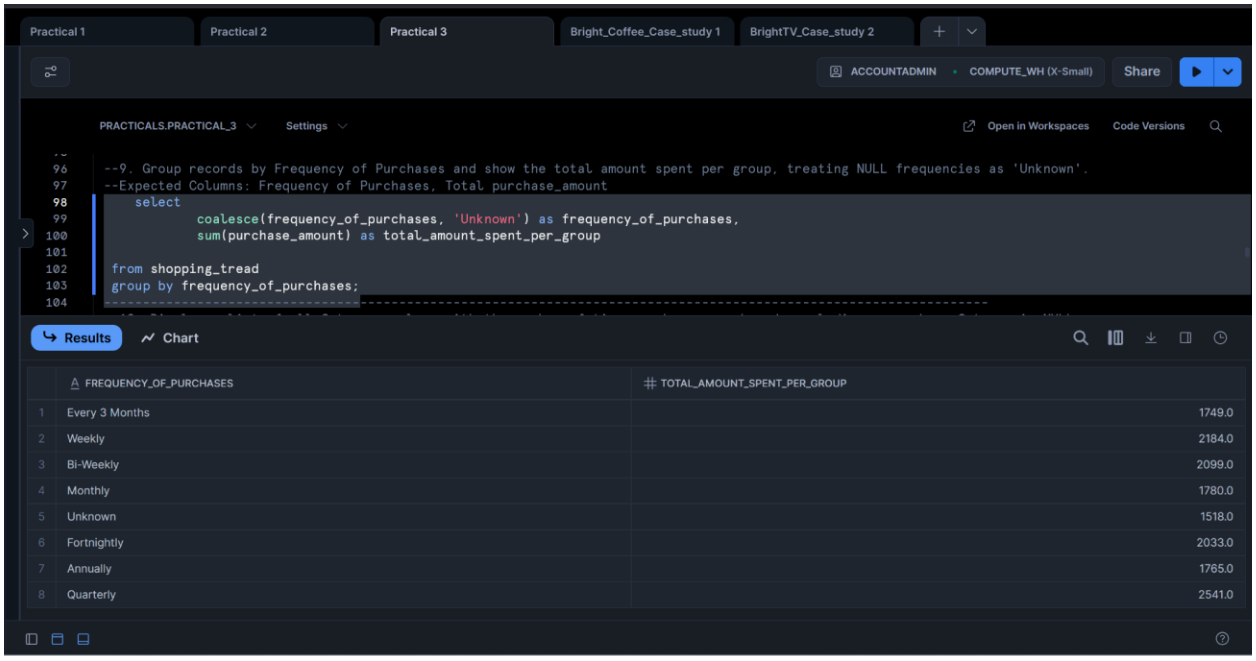
Q8



```
--8. 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 shopping_tread
 where previous_purchases is null
 and color is not null;
------------------------------------------------------------------------------------------------------------------------
```

	CUSTOMER_ID	COLOR	PREVIOUS_PURCHASES
23	226	Black	null
24	230	Green	null
25	251	Gray	null
26	253	Green	null
27	260	Pink	null
28	271	Yellow	null
29	282	Pink	null
30	283	Green	null
31	290	White	null

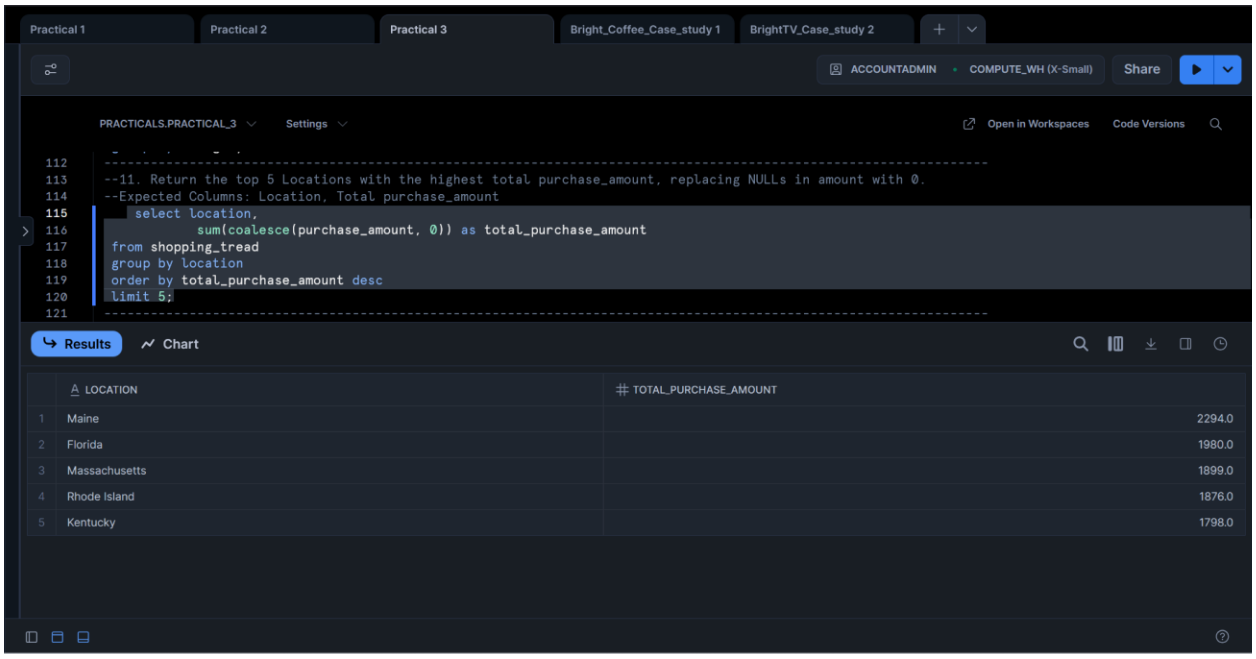
Q9



```
--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
            coalesce(frequency_of_purchases, 'Unknown') as frequency_of_purchases,
            sum(purchase_amount) as total_amount_spent_per_group

from shopping_tread
group by frequency_of_purchases;
------------------------------------------------------------------------------------------------------------------------------
```

	FREQUENCY_OF_PURCHASES	TOTAL_AMOUNT_SPENT_PER_GROUP
1	Every 3 Months	1749.0
2	Weekly	2184.0
3	Bi-Weekly	2099.0
4	Monthly	1780.0
5	Unknown	1518.0
6	Fortnightly	2033.0
7	Annually	1765.0
8	Quarterly	2541.0

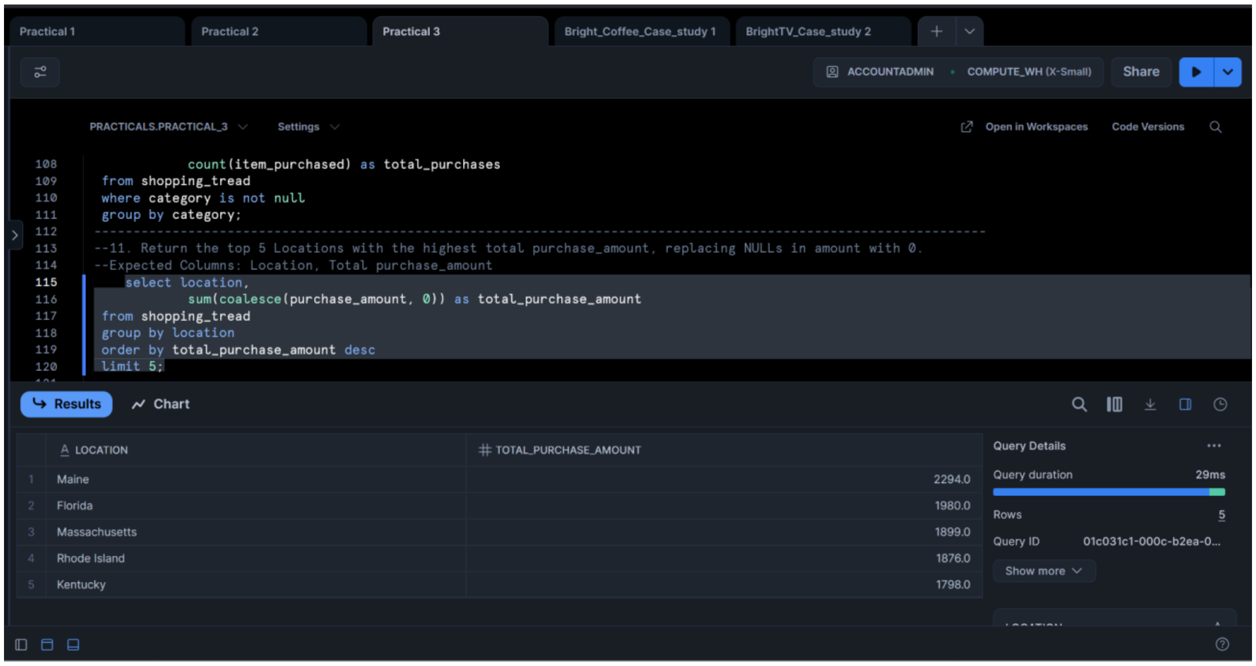
Q10



```
------------------------------------------------------------------------------------------------------------------------------
--11. Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0.
--Expected Columns: Location, Total purchase_amount
    select location,
            sum(coalesce(purchase_amount, 0)) as total_purchase_amount
from shopping_tread
group by location
order by total_purchase_amount desc
limit 5;
------------------------------------------------------------------------------------------------------------------------------
```

	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

Q11

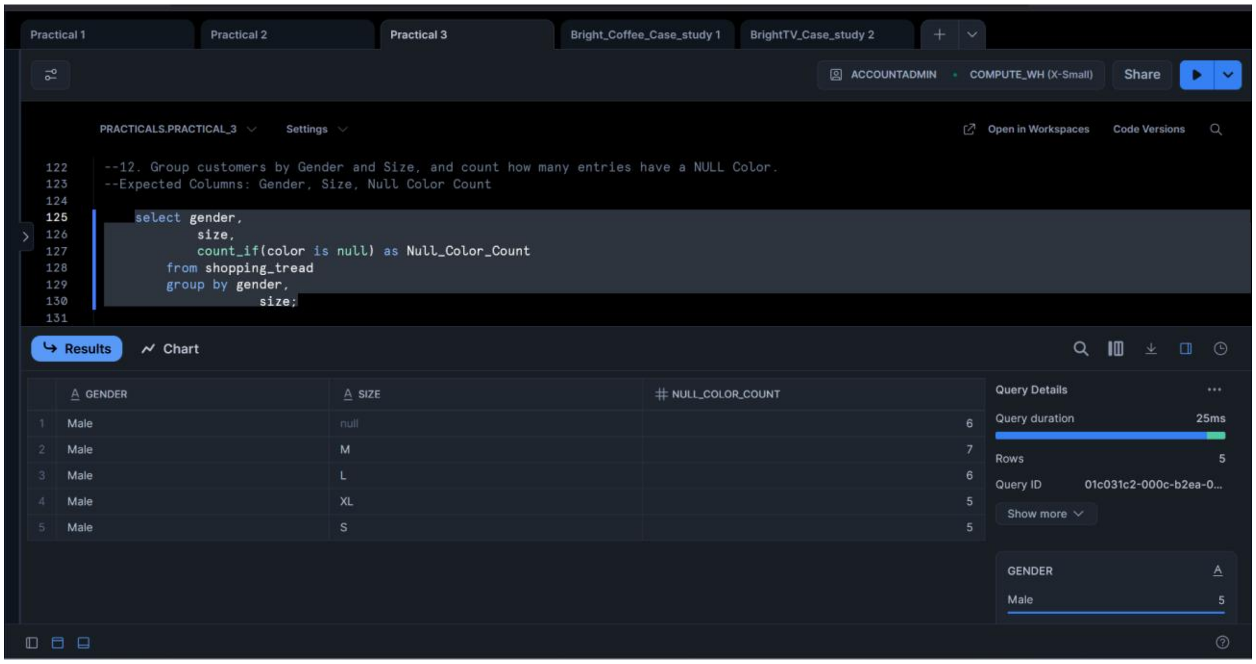


```sql
        count(item_purchased) as total_purchases
    from shopping_tread
    where category is not null
    group by category;
   ---------------------------------------------------------------------------------------------------------------
   --11. Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0.
   --Expected Columns: Location, Total purchase_amount
       select location,
               sum(coalesce(purchase_amount, 0)) as total_purchase_amount
    from shopping_tread
    group by location
    order by total_purchase_amount desc
    limit 5;
```

	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

Query Details
Query duration 29ms
Rows 5
Query ID 01c031c1-000c-b2ea-0...
Show more

Q12



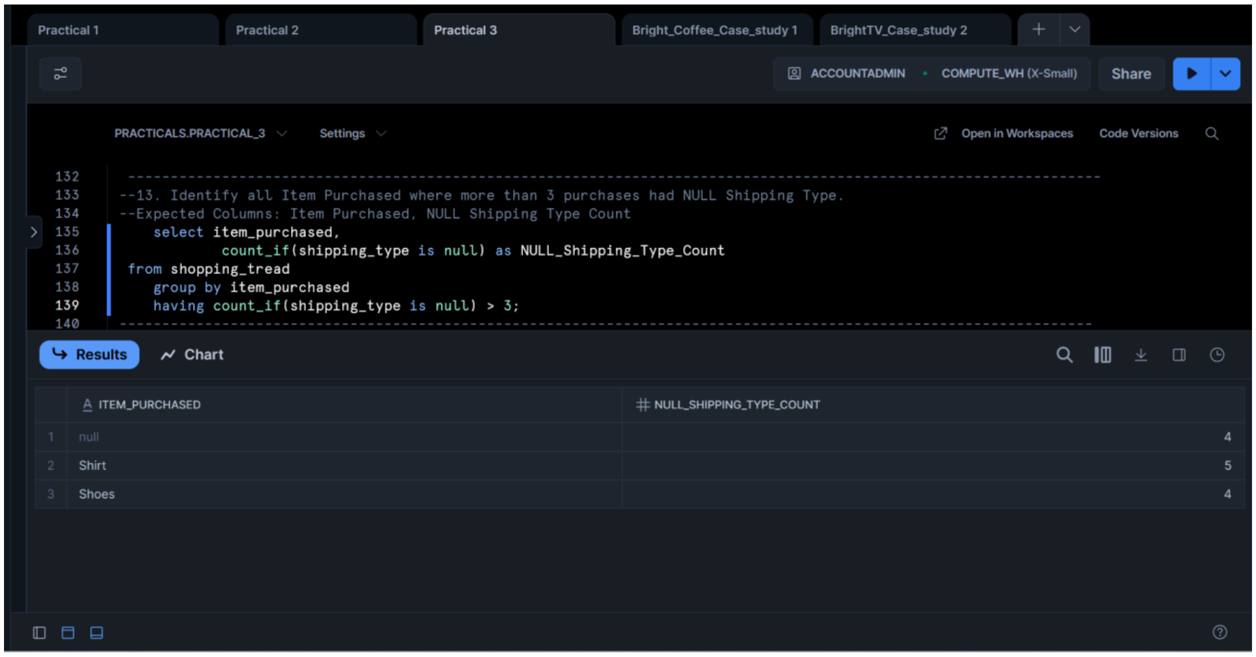
```sql
   --12. Group customers by Gender and Size, and count how many entries have a NULL Color.
   --Expected Columns: Gender, Size, Null Color Count

       select gender,
               size,
               count_if(color is null) as Null_Color_Count
           from shopping_tread
           group by gender,
                       size;
```

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

Query Details
Query duration 25ms
Rows 5
Query ID 01c031c2-000c-b2ea-0...
Show more

GENDER
Male 5

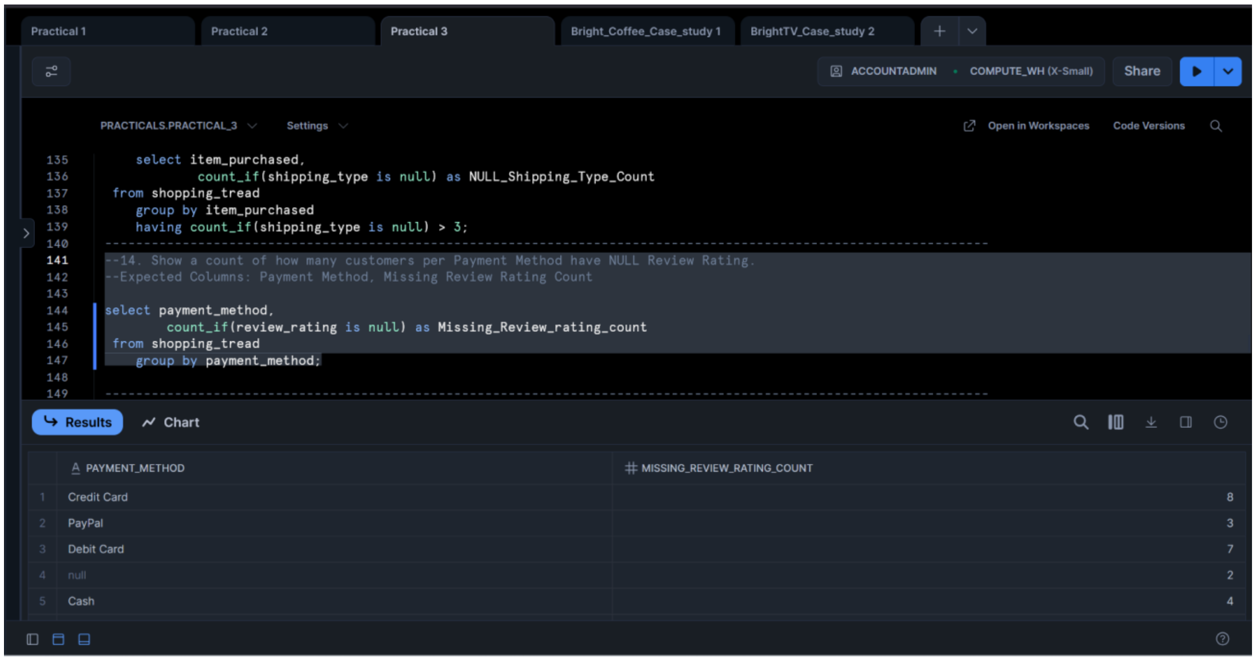
Q13



```
------------------------------------------------------------------------------------------------------------------------
--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_if(shipping_type is null) as NULL_Shipping_Type_Count
 from shopping_tread
    group by item_purchased
    having count_if(shipping_type is null) > 3;
------------------------------------------------------------------------------------------------------------------------
```

```
  ITEM_PURCHASED          NULL_SHIPPING_TYPE_COUNT
1 null                                           4
2 Shirt                                          5
3 Shoes                                          4
```

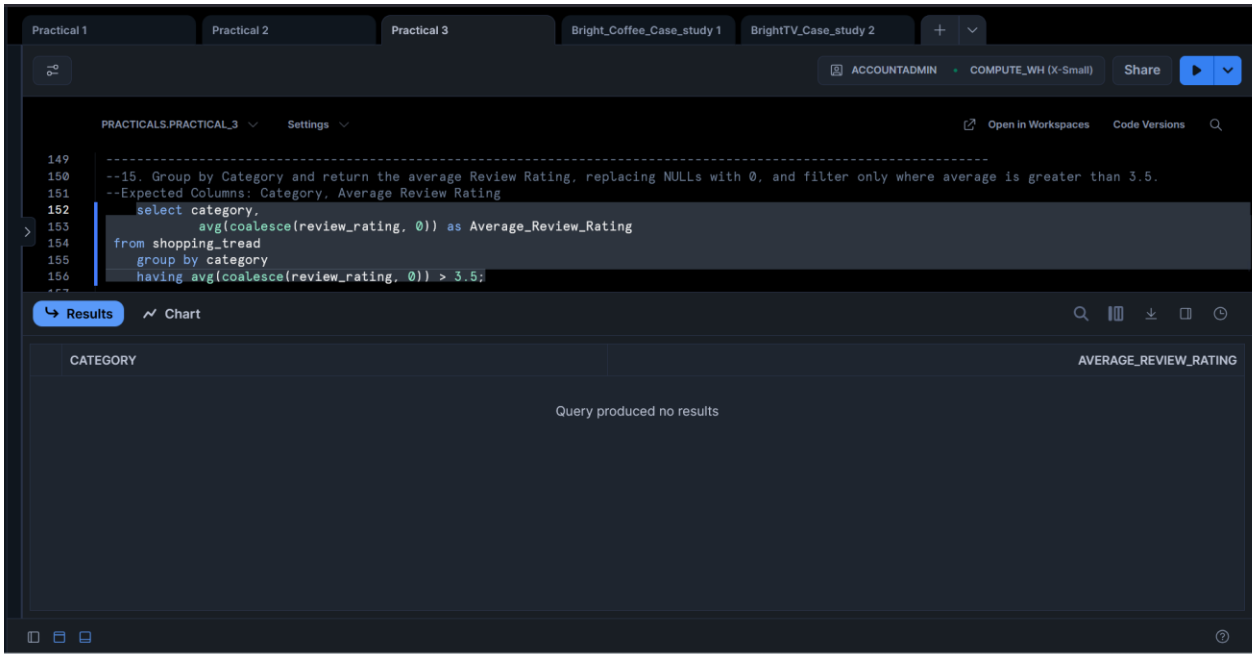
Q14



```
    select item_purchased,
            count_if(shipping_type is null) as NULL_Shipping_Type_Count
 from shopping_tread
    group by item_purchased
    having count_if(shipping_type is null) > 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_if(review_rating is null) as Missing_Review_rating_count
 from shopping_tread
    group by payment_method;

------------------------------------------------------------------------------------------------------------------------
```

```
  PAYMENT_METHOD          MISSING_REVIEW_RATING_COUNT
1 Credit Card                                      8
2 PayPal                                           3
3 Debit Card                                       7
4 null                                             2
5 Cash                                             4
```

Q15

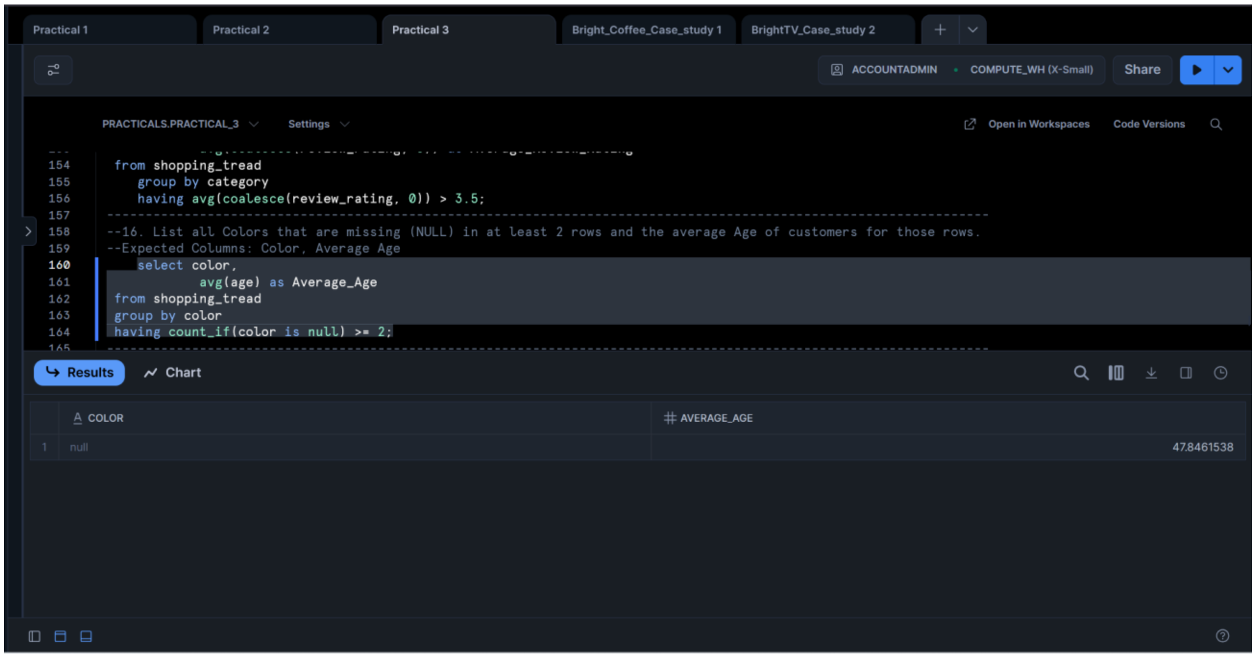


```
-----------------------------------------------------------------------------------------------------------------
--15. Group by Category and return the average Review Rating, replacing NULLs with 0, and filter only where average is greater than 3.5.
--Expected Columns: Category, Average Review Rating
   select category,
           avg(coalesce(review_rating, 0)) as Average_Review_Rating
from shopping_tread
   group by category
   having avg(coalesce(review_rating, 0)) > 3.5;
```

CATEGORY	AVERAGE_REVIEW_RATING

Query produced no results

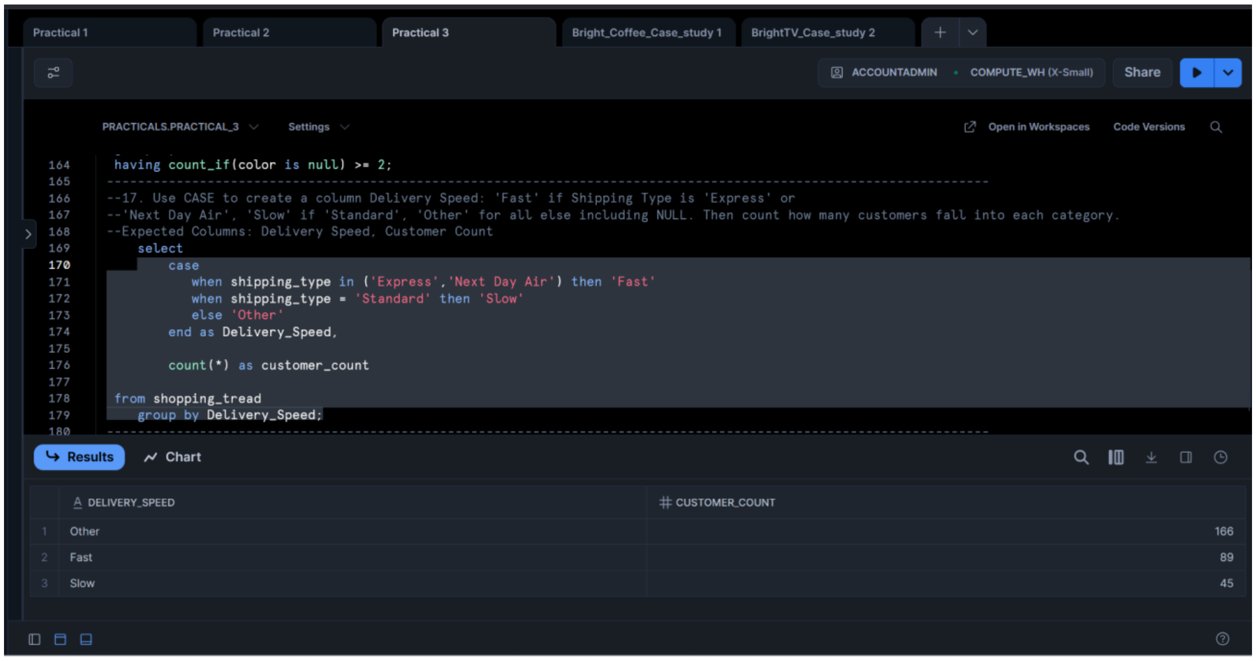
Q16



```
from shopping_tread
   group by category
   having avg(coalesce(review_rating, 0)) > 3.5;
-----------------------------------------------------------------------------------------------------------------
--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 shopping_tread
group by color
having count_if(color is null) >= 2;
```

	COLOR	AVERAGE_AGE
1	null	47.8461538

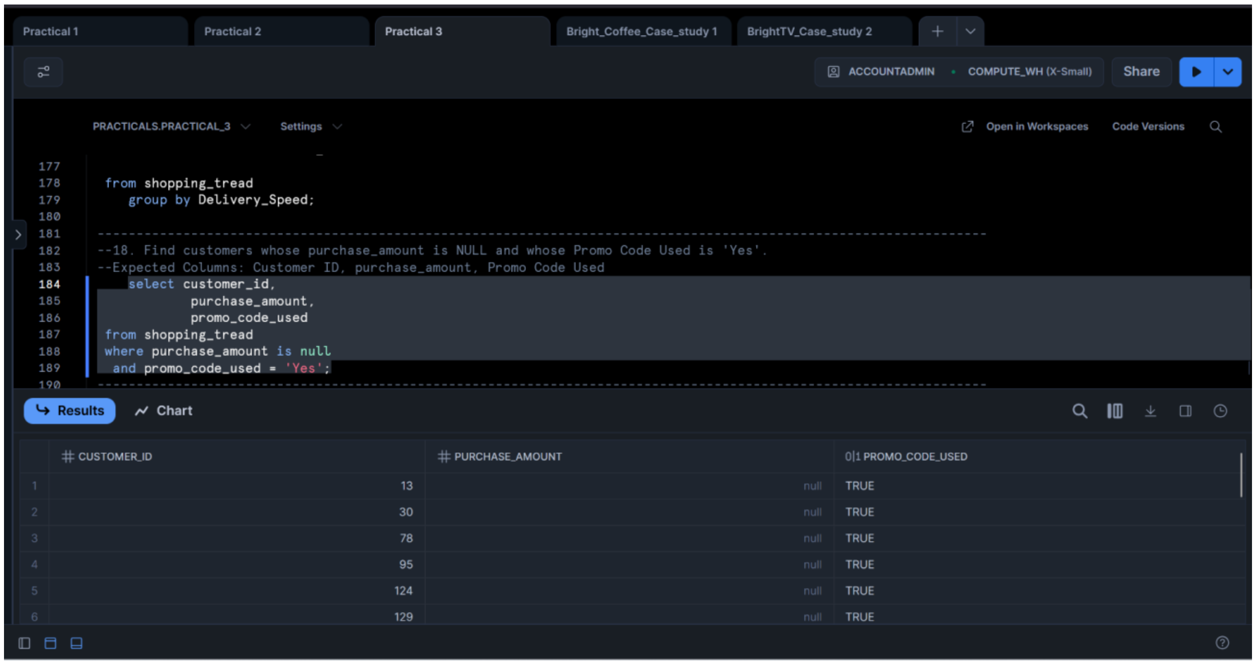
Q17



```
having count_if(color is null) >= 2;
------------------------------------------------------------------------------------------------------------------------
--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 in ('Express','Next Day Air') then 'Fast'
           when shipping_type = 'Standard' then 'Slow'
           else 'Other'
        end as Delivery_Speed,

        count(*) as customer_count

 from shopping_tread
    group by Delivery_Speed;
------------------------------------------------------------------------------------------------------------------------
```

	DELIVERY_SPEED	CUSTOMER_COUNT
1	Other	166
2	Fast	89
3	Slow	45

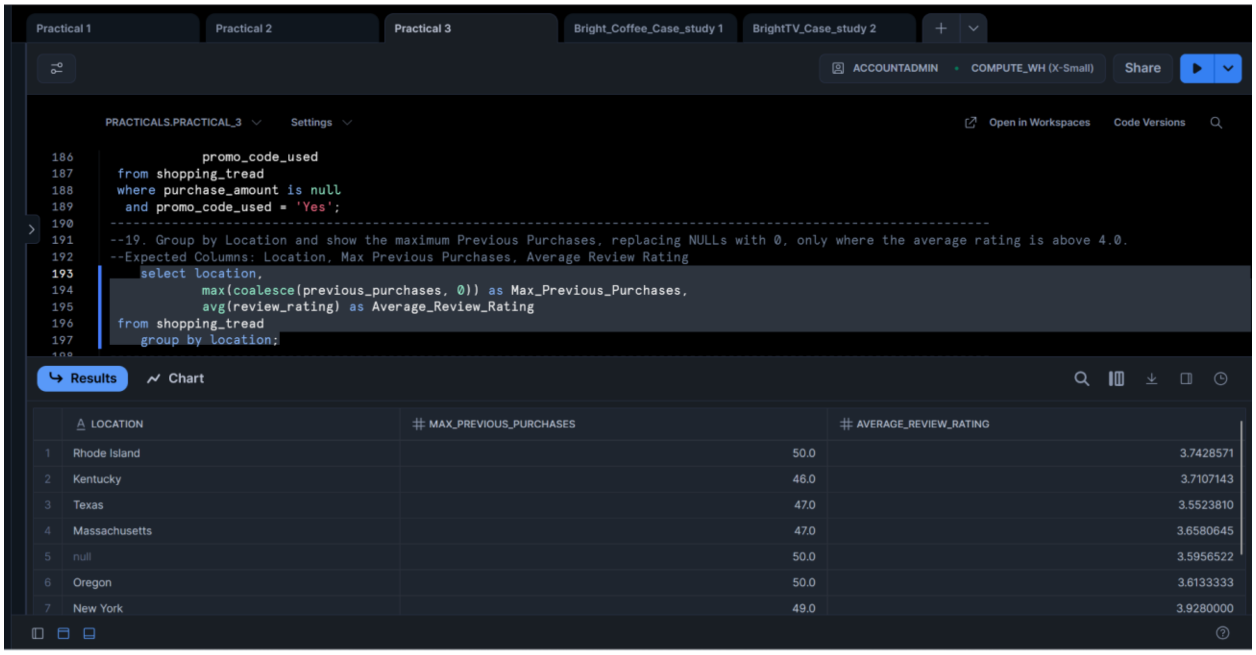
Q18



```
 from shopping_tread
    group by Delivery_Speed;

------------------------------------------------------------------------------------------------------------------------
--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 shopping_tread
 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

Q19


Practical 1 | Practical 2 | Practical 3 | Bright_Coffee_Case_study 1 | BrightTV_Case_study 2

ACCOUNTADMIN · COMPUTE_WH (X-Small) | Share

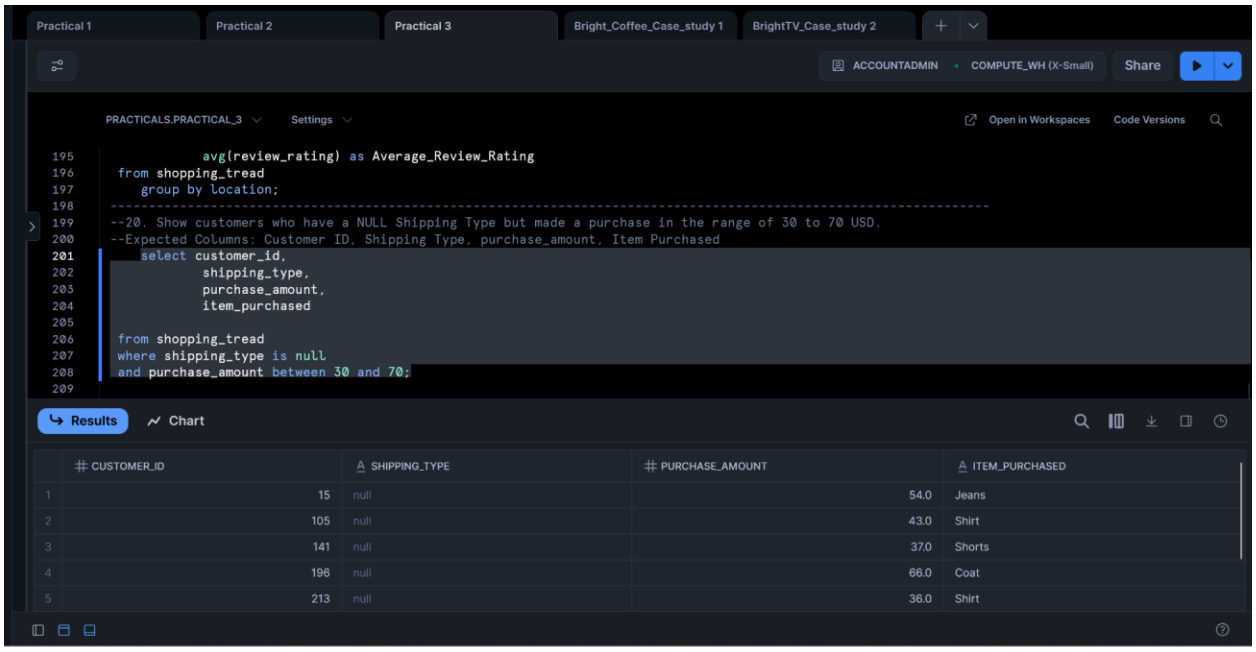
PRACTICALS.PRACTICAL_3 | Settings | Open in Workspaces | Code Versions

```sql
            promo_code_used
 from shopping_tread
 where purchase_amount is null
  and promo_code_used = 'Yes';
-----------------------------------------------------------------------------------------------------------------
--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(coalesce(previous_purchases, 0)) as Max_Previous_Purchases,
            avg(review_rating) as Average_Review_Rating
 from shopping_tread
    group by location;
```

Results | Chart

	LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE_REVIEW_RATING
1	Rhode Island	50.0	3.7428571
2	Kentucky	46.0	3.7107143
3	Texas	47.0	3.5523810
4	Massachusetts	47.0	3.6580645
5	null	50.0	3.5956522
6	Oregon	50.0	3.6133333
7	New York	49.0	3.9280000

Q20


Practical 1 | Practical 2 | Practical 3 | Bright_Coffee_Case_study 1 | BrightTV_Case_study 2

ACCOUNTADMIN · COMPUTE_WH (X-Small) | Share

PRACTICALS.PRACTICAL_3 | Settings | Open in Workspaces | Code Versions

```sql
            avg(review_rating) as Average_Review_Rating
 from shopping_tread
    group by location;
-----------------------------------------------------------------------------------------------------------------
--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 shopping_tread
 where shipping_type is null
 and purchase_amount between 30 and 70;
```

Results | Chart

	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