Nolwazi Ndiweni

Practical 3

Question 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 prac_3.shoping_trends
WHERE size IS NULL AND purchase_amount>50
```

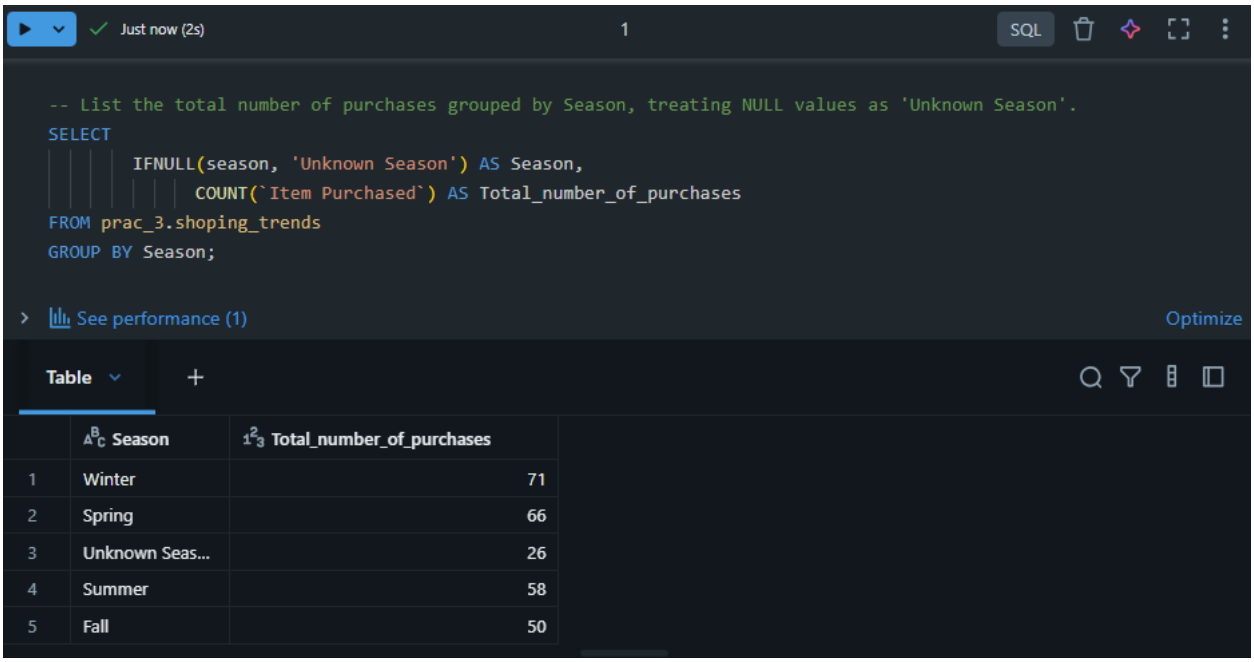
Just now (3s) 1 SQL

See performance (1) Optimize

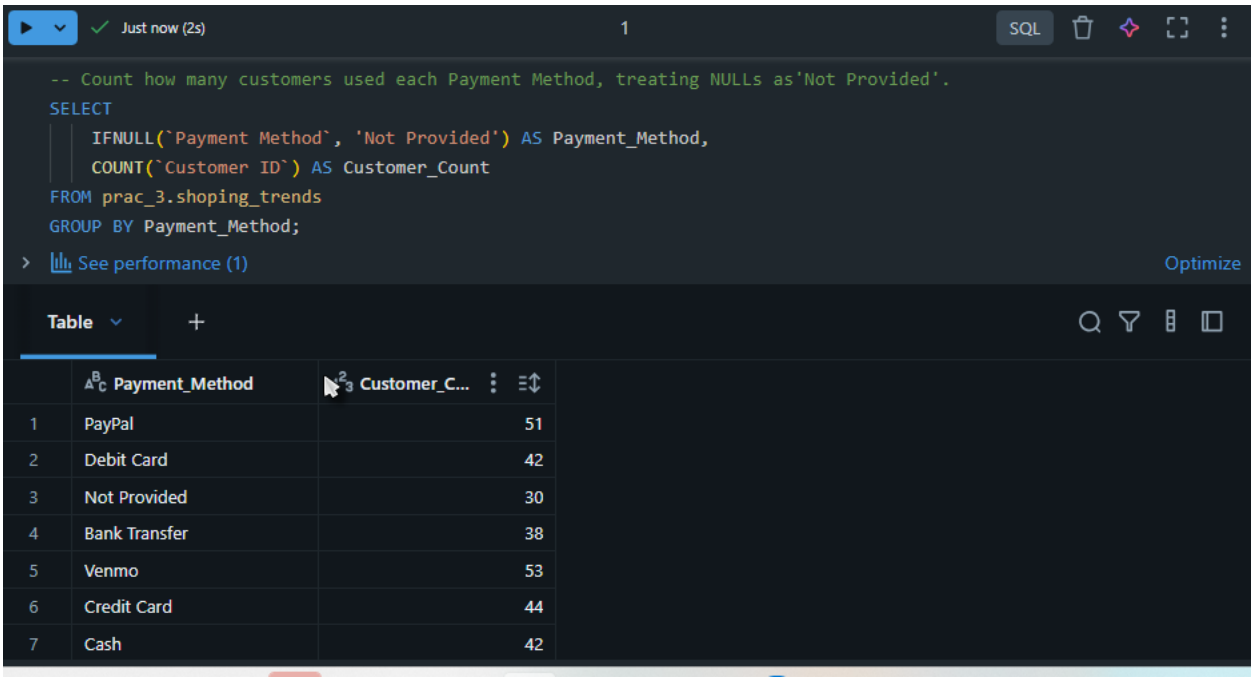
Table

	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

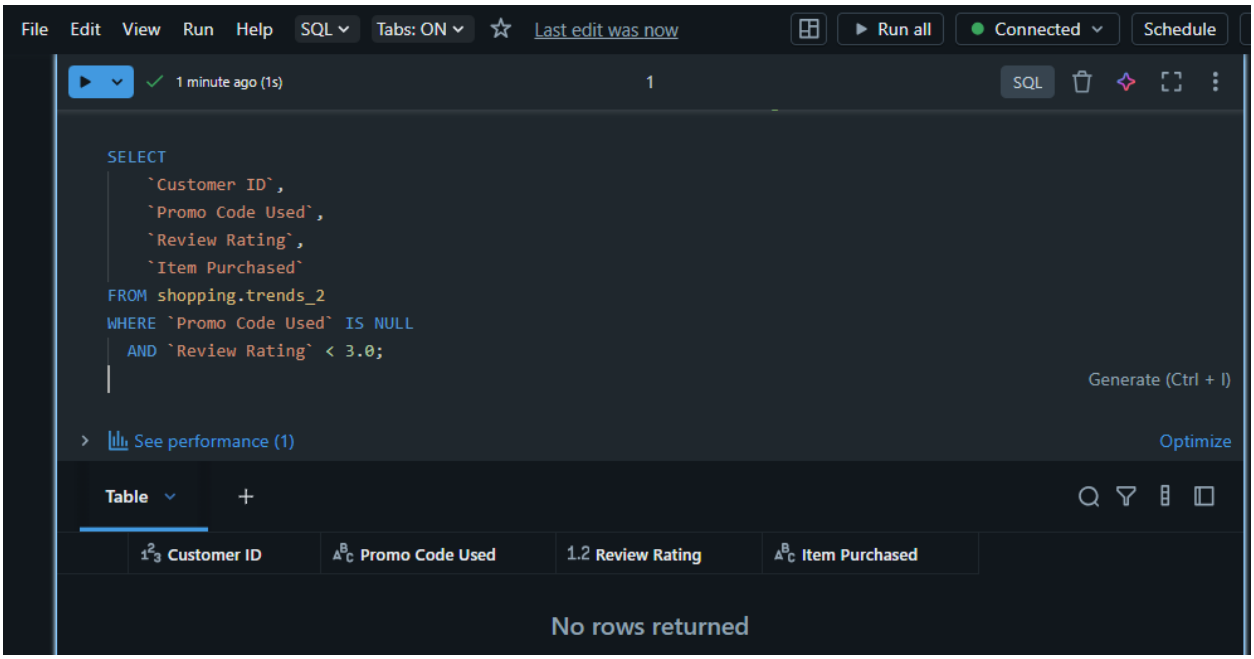
Question 2



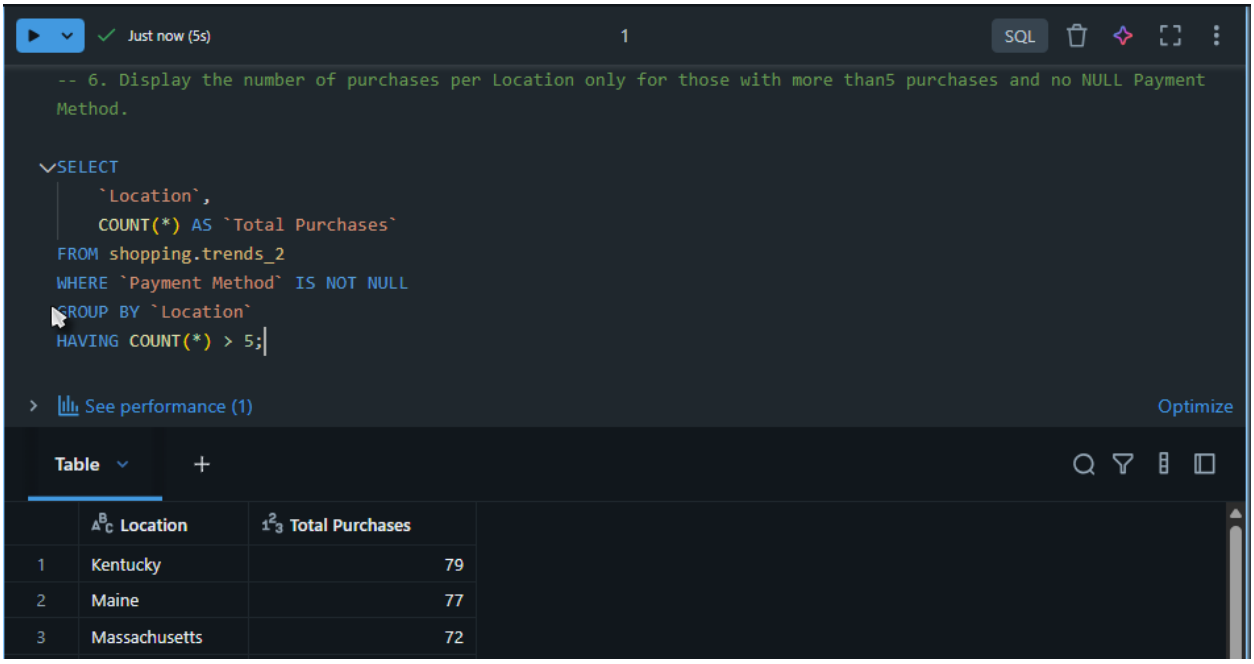
Question 3



Question 4



Question 6



Question 7

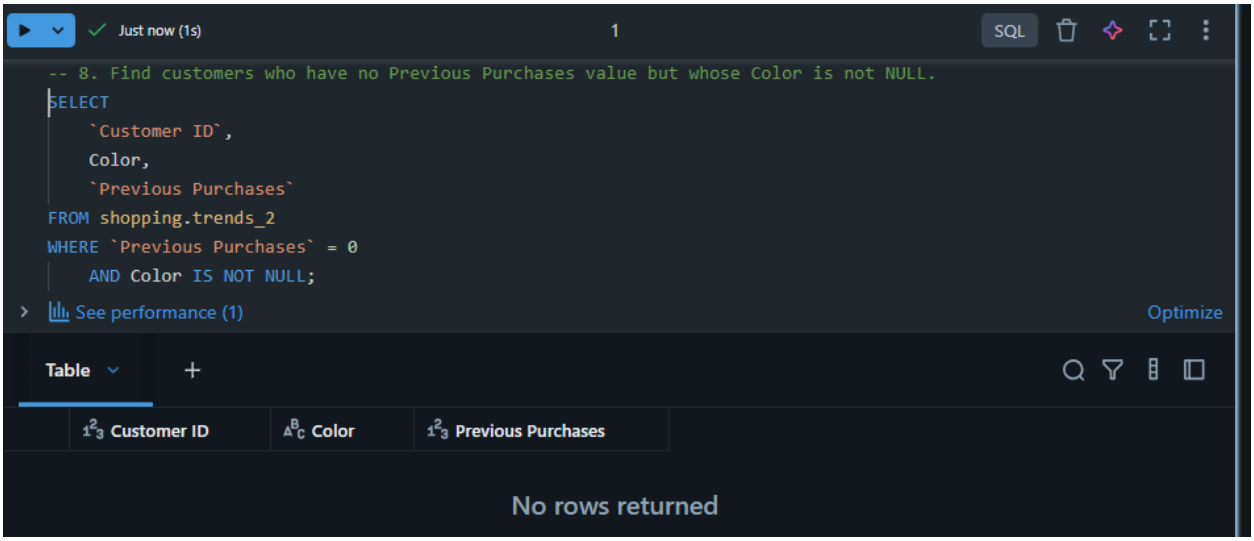
```sql
-- 7. Create a column Spender Category that classifies customers using CASE:
-- 'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80,
-- 'Low' otherwise. Replace NULLs in purchase_amount with 0.
SELECT
    `Customer ID`,
    COALESCE(`purchase amount (usd)`, 0) AS `purchase_amount`,
    CASE
        WHEN COALESCE(`purchase_amount`, 0) > 80 THEN 'High'
        WHEN COALESCE(`purchase_amount`, 0) BETWEEN 50 AND 80 THEN 'Medium'
        ELSE 'Low'
    END AS `Spender Category`
FROM shopping.trends_2;
```

See performance (1) Optimi

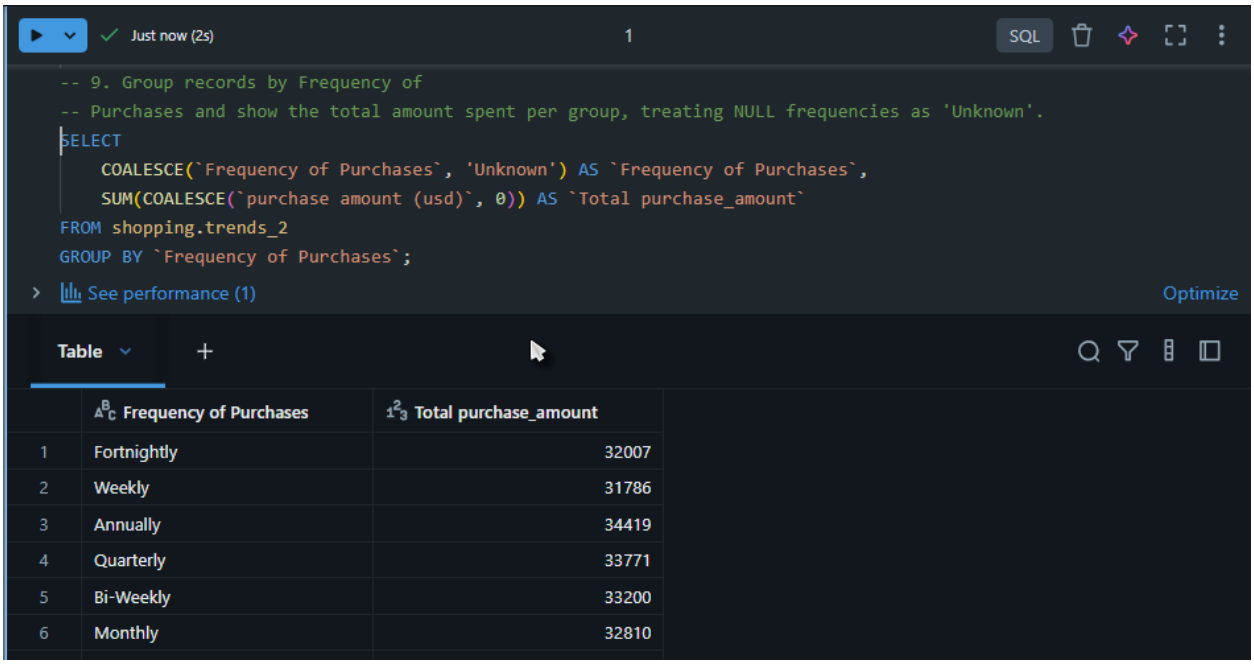
Table +

	Customer ID	purchase_amount	Spender Category
1	1	53	Medium
2	2	64	Medium

Question 8



Question 9



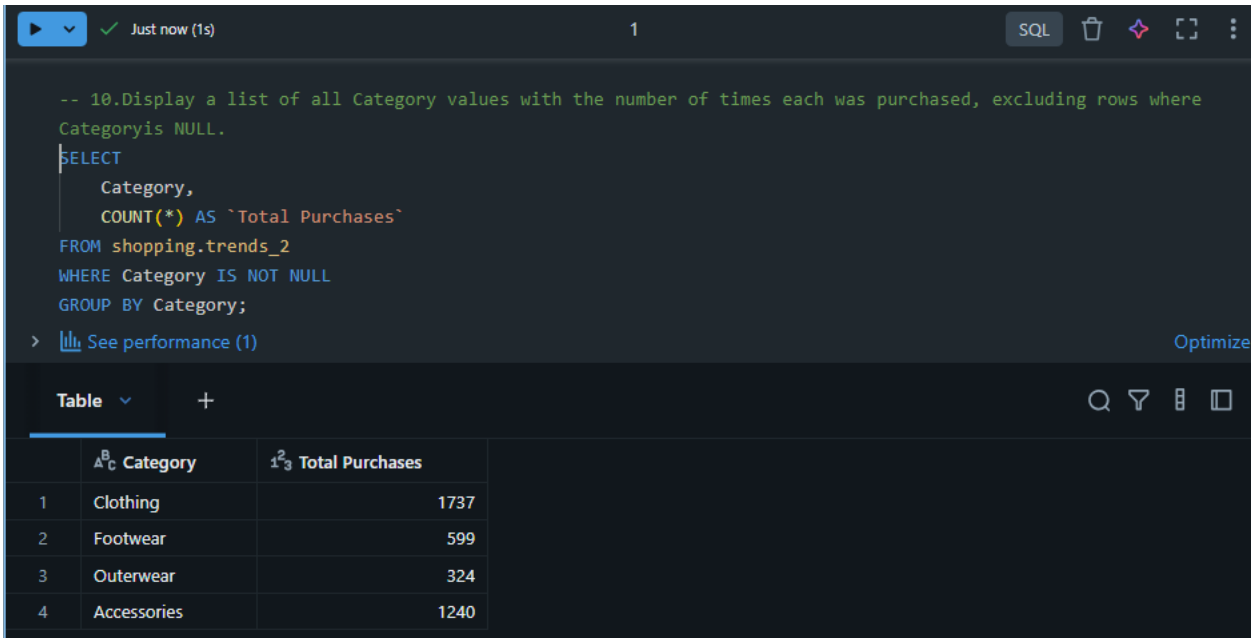
```
-- 9. Group records by Frequency of
-- Purchases and show the total amount spent per group, treating NULL frequencies as 'Unknown'.
SELECT
    COALESCE(`Frequency of Purchases`, 'Unknown') AS `Frequency of Purchases`,
    SUM(COALESCE(`purchase amount (usd)`, 0)) AS `Total purchase_amount`
FROM shopping.trends_2
GROUP BY `Frequency of Purchases`;
```

Just now (2s) | SQL | See performance (1) | Optimize

Table

	Frequency of Purchases	Total purchase_amount
1	Fortnightly	32007
2	Weekly	31786
3	Annually	34419
4	Quarterly	33771
5	Bi-Weekly	33200
6	Monthly	32810

Question 10



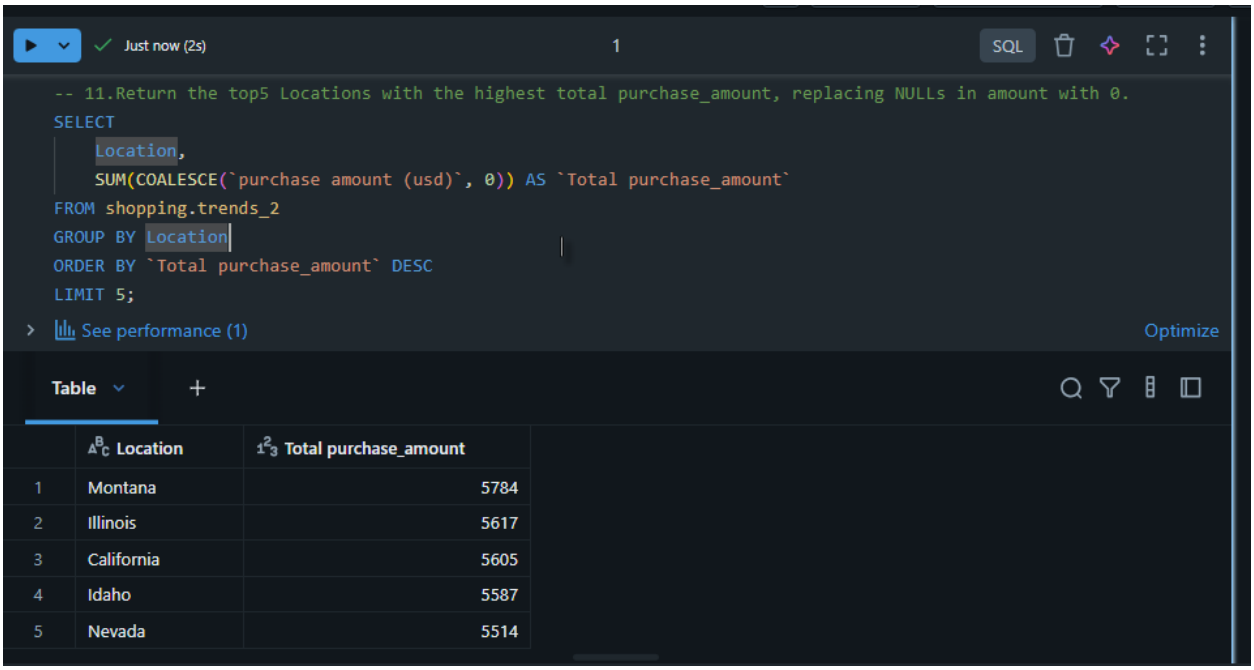
```
-- 10.Display a list of all Category values with the number of times each was purchased, excluding rows where
Categoryis NULL.
SELECT
    Category,
    COUNT(*) AS `Total Purchases`
FROM shopping.trends_2
WHERE Category IS NOT NULL
GROUP BY Category;
```

Just now (1s) | SQL | See performance (1) | Optimize

Table

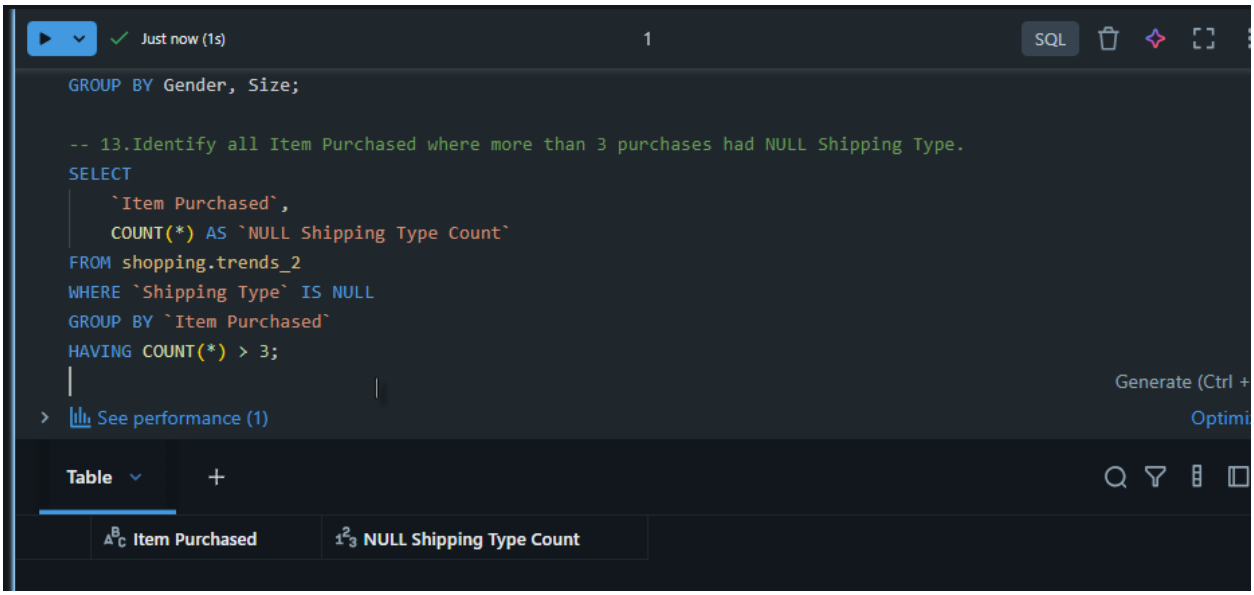
	Category	Total Purchases
1	Clothing	1737
2	Footwear	599
3	Outerwear	324
4	Accessories	1240

Question 11

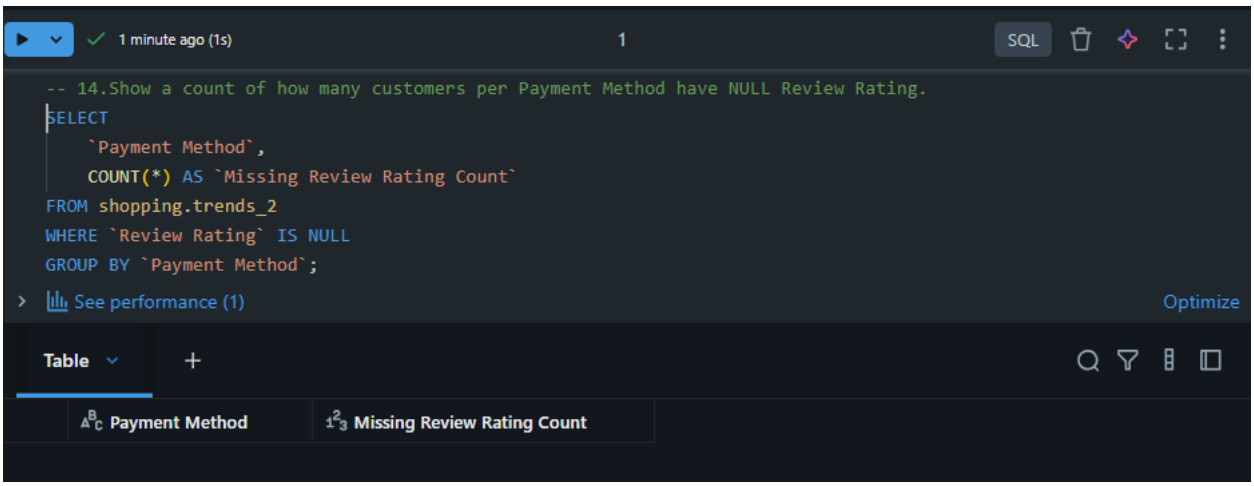


Question 12

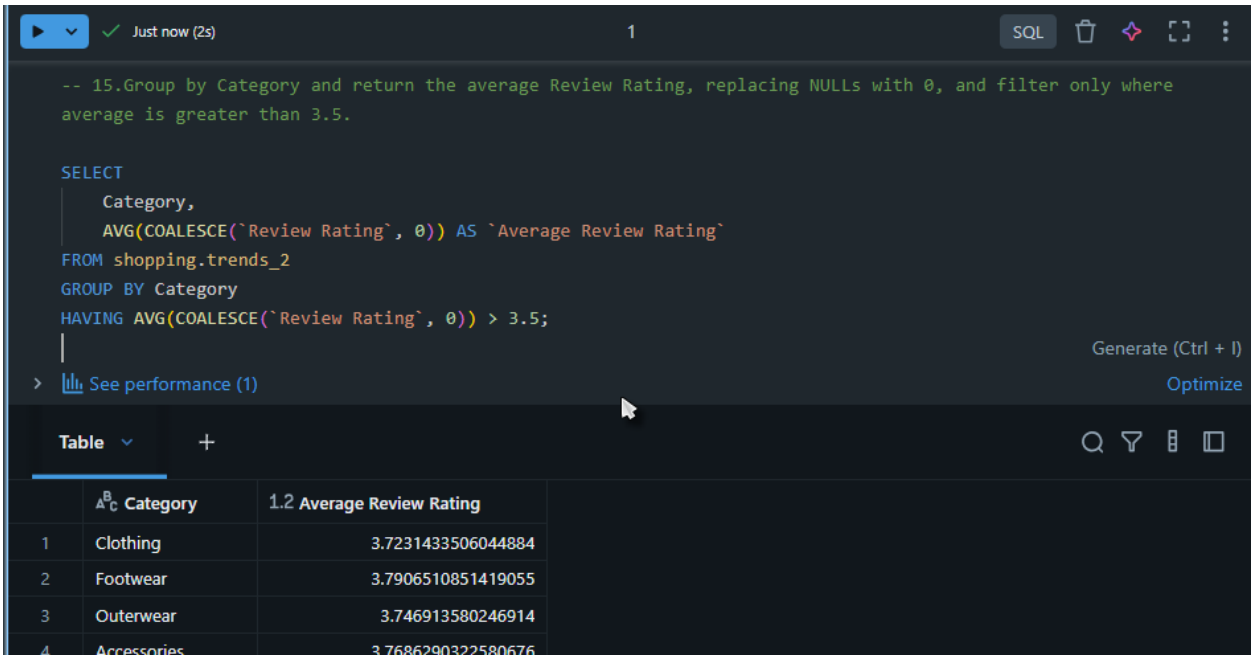
Question 13



Question 14



Question 15


Just now (2s)
1
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.
SELECT
Category,
AVG(COALESCE(`Review Rating`, 0)) AS `Average Review Rating`
FROM shopping.trends_2
GROUP BY Category
HAVING AVG(COALESCE(`Review Rating`, 0)) > 3.5;
Generate (Ctrl + I)
See performance (1)
Optimize
Table
Category
1.2 Average Review Rating
1 Clothing 3.7231433506044884
2 Footwear 3.7906510851419055
3 Outerwear 3.746913580246914
4 Accessories 3.7686290322580676