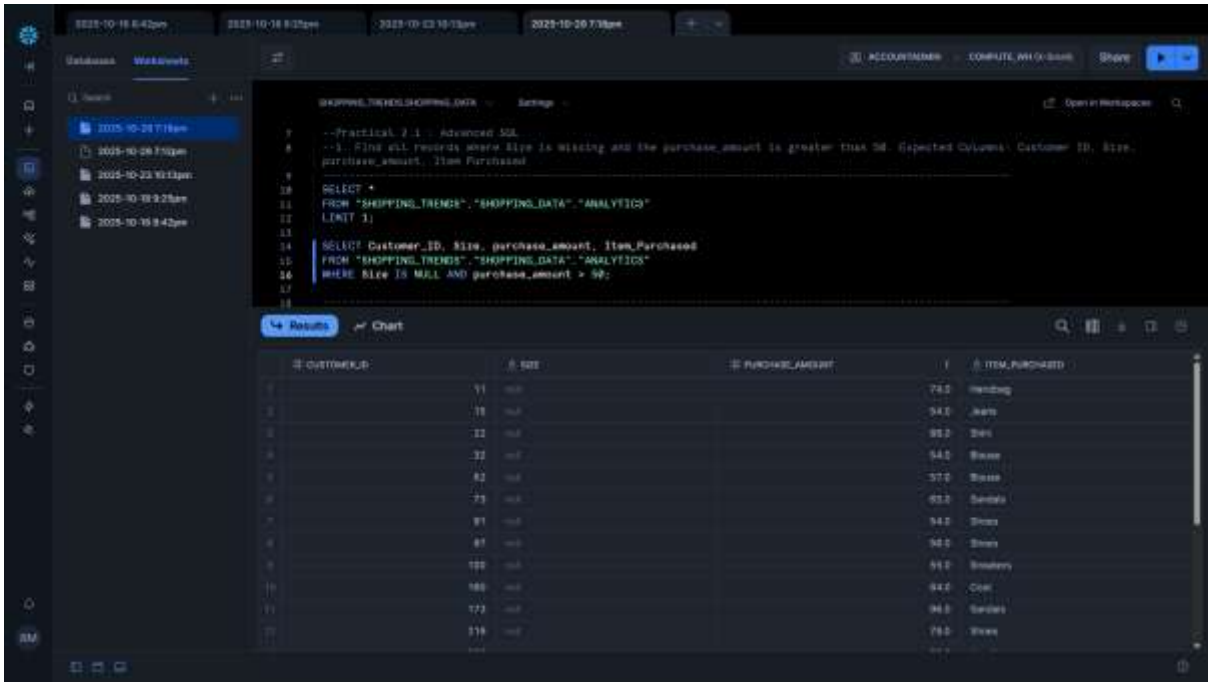
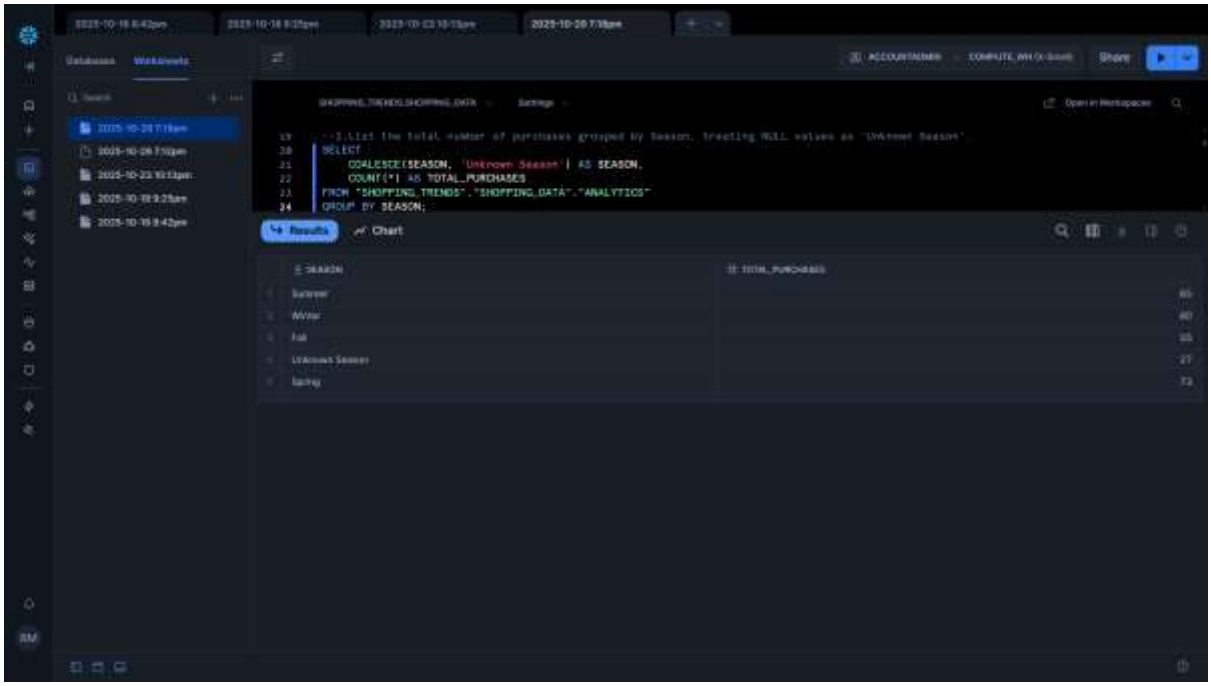
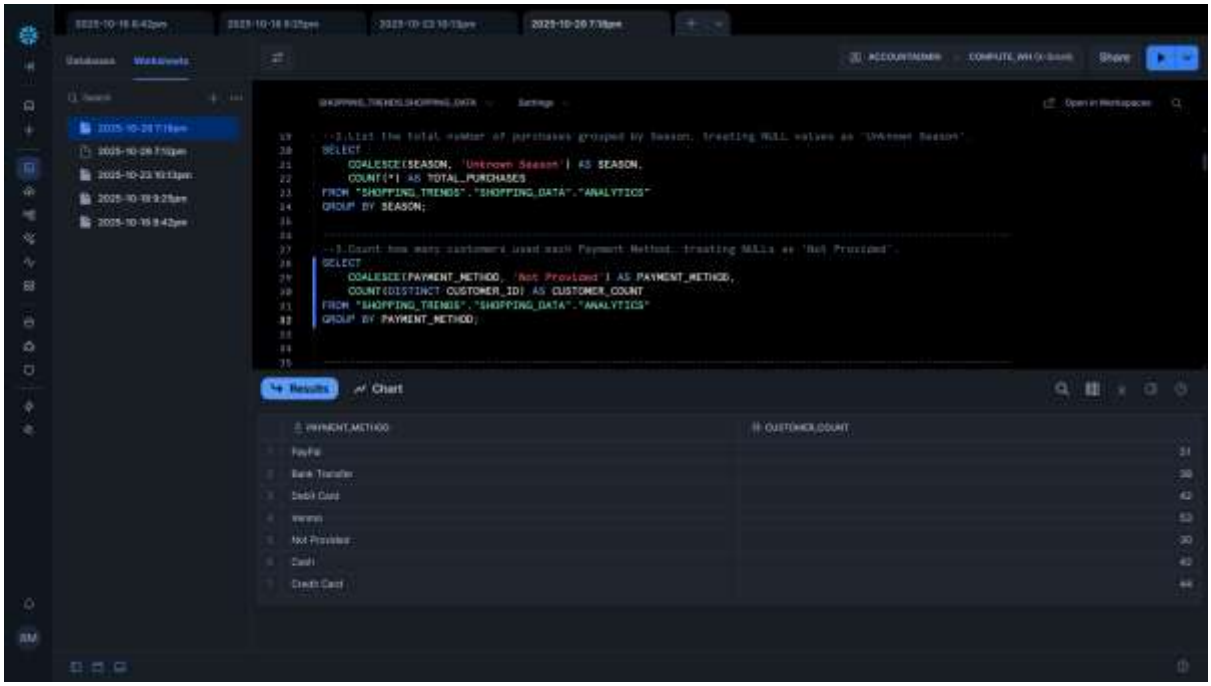
1.


--Practical 2.1 : Advanced 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 *
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
LIMIT 1;
SELECT Customer_ID, Size, purchase_amount, Item_Purchased
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE Size IS NULL AND purchase_amount > 50;
Results
Chart

2.


--2.List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'
SELECT
COALESCE(SEASON, 'Unknown Season') AS SEASON,
COUNT(*) AS TOTAL_PURCHASES
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
GROUP BY SEASON;
Results
Chart

3.



4.

5.

6.

```sql
--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 "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE PAYMENT_METHOD IS NOT NULL
GROUP BY LOCATION
HAVING COUNT(*) > 5;
```

Results | Chart

	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

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. 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'
        WHEN COALESCE(PURCHASE_AMOUNT, 0) BETWEEN 50 AND 80 THEN 'Medium'
        ELSE 'Low'
    END AS SPENDER_CATEGORY
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS";
```

Results | Chart

	CUSTOMER_ID	PURCHASE_AMOUNT	SPENDER_CATEGORY
1	1	30.0	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
9	9	100.0	High

8. 

```
--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_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE PREVIOUS_PURCHASES IS NULL AND COLOR IS NOT NULL;
```

Results | Chart

CUSTOMER_ID	COLOR	PREVIOUS_PURCHASES
8	Green	null
21	Yellow	null
25	White	null
37	Maroon	null
40	Gray	null
43	Black	null
44	Green	null
70	White	null
73	Maroon	null

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
    COALESCE(FREQUENCY_OF_PURCHASES, 'Unknown') AS FREQUENCY_OF_PURCHASES,
    SUM(COALESCE(PURCHASE_AMOUNT, 0)) AS TOTAL_PURCHASE_AMOUNT
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
GROUP BY FREQUENCY_OF_PURCHASES;
```

Results | Chart

FREQUENCY_OF_PURCHASES	TOTAL_PURCHASE_AMOUNT
Every 3 Months	1749.0
Weekly	2184.0
Bi-Weekly	2099.0
Monthly	1780.0
Unknown	1518.0
Fortnightly	2033.0
Annually	1785.0
Quarterly	2541.0

10.

```
--10. Display a list of all Category values with the number of times each was purchased, excluding rows where Category is NULL.
Expected Columns: Category, Total Purchases
SELECT
    CATEGORY,
    COUNT(*) AS TOTAL_PURCHASES
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE CATEGORY IS NOT NULL
GROUP BY CATEGORY;
```

Results | Chart

CATEGORY	TOTAL_PURCHASES
Outerwear	60
Footwear	70
Clothing	59
Accessories	78

11.

```
--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_TRENDS"."SHOPPING_DATA"."ANALYTICS"
GROUP BY LOCATION
ORDER BY TOTAL_PURCHASE_AMOUNT DESC
LIMIT 5;
```

Results | Chart

LOCATION	TOTAL_PURCHASE_AMOUNT
Maine	2294.0
Florida	1980.0
Massachusetts	1899.0
Rhode Island	1876.0
Kentucky	1798.0

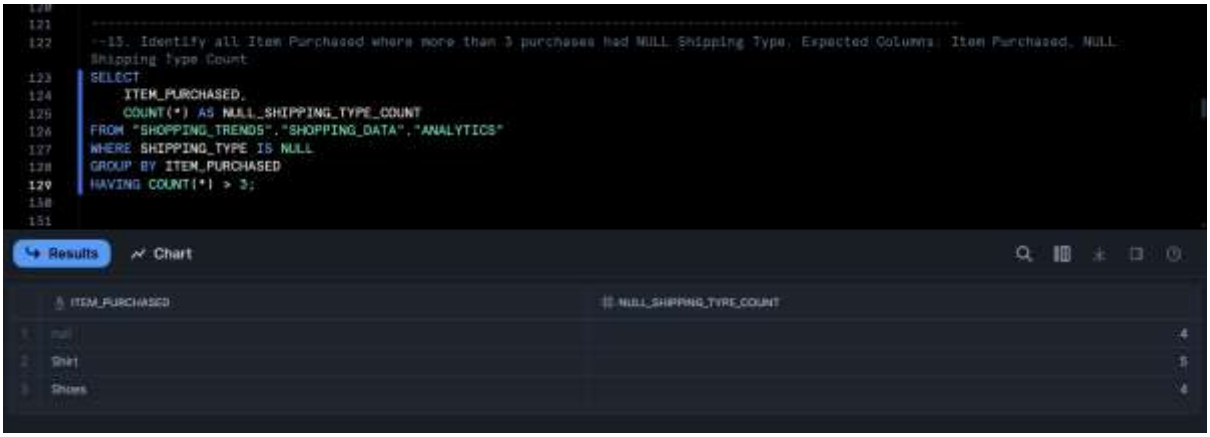
12.

```
--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 "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE COLOR IS NULL
GROUP BY GENDER, SIZE;
```

Results | Chart

GENDER	SIZE	NULL_COLOR_COUNT
Male	null	6
Male	M	7
Male	L	6
Male	S	5
Male	XL	6

13.



```
--15. 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 "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE SHIPPING_TYPE IS NULL
GROUP BY ITEM_PURCHASED
HAVING COUNT(*) > 3;
```

Results | Chart

ITEM_PURCHASED	NULL_SHIPPING_TYPE_COUNT
null	4
Shirt	5
Shoes	4

14.

```
--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 "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE REVIEW_RATING IS NULL
GROUP BY PAYMENT_METHOD;
```

Results | Chart

PAYMENT_METHOD	MISSING_REVIEW_RATING_COUNT
Credit Card	8
Cash	4
null	3
Debit Card	7
Venmo	8
PayPal	3
Bank Transfer	4

15.

```
--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_TRENDS"."SHOPPING_DATA"."ANALYTICS"
GROUP BY CATEGORY
HAVING AVG(COALESCE(REVIEW_RATING, 0)) > 3.5;
```

Results | Chart

CATEGORY	AVERAGE_REVIEW_RATING

Query produced no results

16.

```
--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_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE COLOR IS NULL
GROUP BY COLOR
HAVING COUNT(*) >= 2;
```

Results | Chart

COLOR	AVERAGE_AGE
null	47.8461538

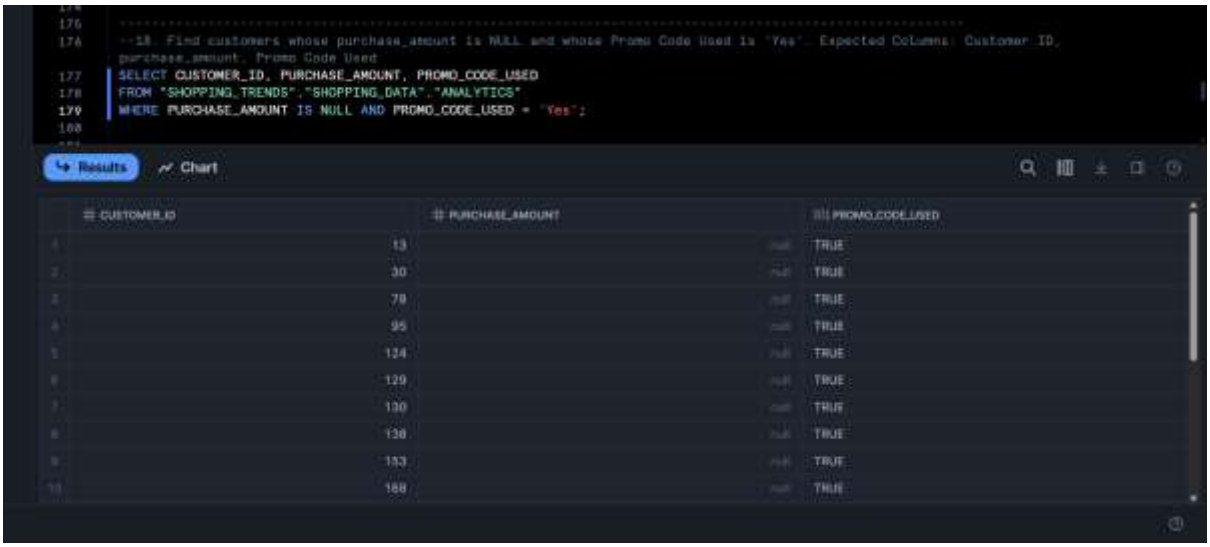
17.

```
--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(DISTINCT CUSTOMER_ID) AS CUSTOMER_COUNT
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
GROUP BY DELIVERY_SPEED;
```

Results | Chart

DELIVERY_SPEED	CUSTOMER_COUNT
Fast	89
Slow	45
Other	166

18.

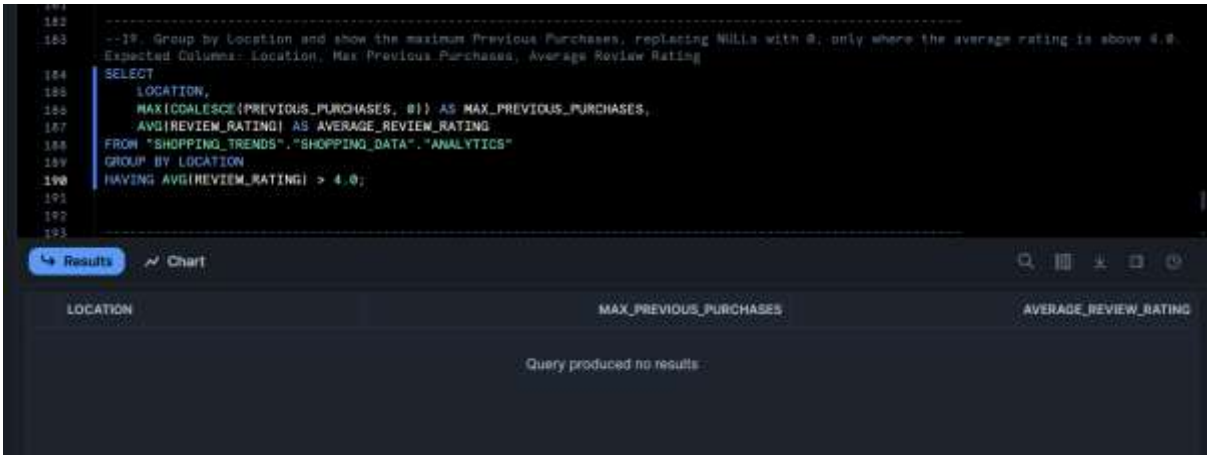


```
--18. Find customers whose purchase_amount is NULL and whose Promo Code Used is 'Yes'. Expected Columns: Customer ID,
purchase_amount, Promo Code Used
SELECT CUSTOMER_ID, PURCHASE_AMOUNT, PROMO_CODE_USED
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE PURCHASE_AMOUNT IS NULL AND PROMO_CODE_USED = 'Yes';
```

Results | Chart

CUSTOMER_ID	PURCHASE_AMOUNT	PROMO_CODE_USED
13	null	TRUE
30	null	TRUE
78	null	TRUE
95	null	TRUE
124	null	TRUE
129	null	TRUE
130	null	TRUE
138	null	TRUE
153	null	TRUE
168	null	TRUE

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(COALESCE(PREVIOUS_PURCHASES, 0)) AS MAX_PREVIOUS_PURCHASES,
AVG(REVIEW_RATING) AS AVERAGE_REVIEW_RATING
FROM "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
GROUP BY LOCATION
HAVING AVG(REVIEW_RATING) > 4.0;
Results
Chart
LOCATION
MAX_PREVIOUS_PURCHASES
AVERAGE_REVIEW_RATING
Query produced no results

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 "SHOPPING_TRENDS"."SHOPPING_DATA"."ANALYTICS"
WHERE SHIPPING_TYPE IS NULL
AND PURCHASE_AMOUNT BETWEEN 30 AND 70;
Results
Chart
CUSTOMER_ID
SHIPPING_TYPE
PURCHASE_AMOUNT
ITEM_PURCHASED
15 null 54.0 Jeans
105 null 43.0 Shirt
141 null 37.0 Shorts
188 null 68.0 Coat
213 null 36.0 Shirt
235 null 39.0 Sandals
293 null 35.0 null