BNTOMBI ANNA M

Brightlight Data Analytics Coding Practical Practical 2.1 :

Advanced SQL The following questions are designed to help you build a strong foundation in basic SQL syntax. You are provided with a dataset named shoping_trends.csv. Upload this dataset to your Snowflake account and use it to answer the questions below. Please follow the instructions below carefully:

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 shoppingtrends_db
where SIZE IS NULL AND PURCHASE_AMOUNT > 50;
```

Results | Chart

	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

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

```
 SELECT
   COALESCE('SEASON', 'Unknown Season') AS SEASON,
   COUNT(*) AS Total_Purchases
FROM SHOPPINGTRENDS_DB
GROUP BY COALESCE(SEASON, 'Unknown Season');
```

Results | Chart

	SEASON	TOTAL_PURCHASES
1	SEASON	80
2	SEASON	73
3	SEASON	27
4	SEASON	65
5	SEASON	55

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 shoppingtrends_db
GROUP BY COALESCE("PAYMENT_METHOD", 'Not Provided');
```

Results | Chart

	PAYMENT_METHOD	CUSTOMER_COUNT
1	PAYMENT_METHOD	51
2	PAYMENT_METHOD	42
3	PAYMENT_METHOD	30
4	PAYMENT_METHOD	38
5	PAYMENT_METHOD	53

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

```sql
SELECT
    CUSTOMER_ID,
    PROMO_CODE_USED,
    ITEM_PURCHASED,
    CAST(REVIEW_RATING AS DECIMAL(10,2)) AS REVIEW_RATING_NUM
FROM shoppingtrends_db
WHERE PROMO_CODE_USED IS NULL
    AND CAST(REVIEW_RATING AS DECIMAL(10,2)) < 3.0;
```

Results | Chart

	CUSTOMER_ID	PROMO_CODE_USED	ITEM_PURCHASED	REVIEW_RATING_NUM
1	21	null	Jeans	2.50
2	38	null	Jeans	2.60
3	61	null	Jeans	2.50
4	80	null	Sneakers	2.60
5	125	null	Sneakers	2.80
6	128	null	Shoes	2.50

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

```sql
SELECT
    "SHIPPING_TYPE",
    AVG(COALESCE(PURCHASE_AMOUNT, 0)) AS Average_Purchase_Amount
FROM shoppingtrends_db
GROUP BY "SHIPPING_TYPE";
```

Results | Chart

	SHIPPING_TYPE	AVERAGE_PURCHASE_AMOUNT
1	Standard	47.6666667
2	Store Pickup	55.3333333
3	null	52.7037037
4	Express	53.4545455
5	2-Day Shipping	51.5576923

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

```sql
SELECT
    LOCATION,
    COUNT(*) AS Total_Purchased
FROM shoppingtrends_db
WHERE PAYMENT_METHOD IS NOT NULL
GROUP BY LOCATION
HAVING COUNT(*) > 5;
```

Results | Chart

	LOCATION	TOTAL_PURCHASED
1	Maine	41
2	Rhode Island	29
3	null	24
4	Florida	32

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

```sql
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 SHOPPINGTRENDS_DB;
```

Results | Chart

	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

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

```sql
SELECT
    CUSTOMER_ID,
    COLOR,
    PREVIOUS_PURCHASES
FROM shoppingtrends_db
WHERE PREVIOUS_PURCHASES IS NULL
    AND COLOR IS NOT NULL;
```

Results | Chart

	CUSTOMER_ID	COLOR	PREVIOUS_PURCHASES
1	8	Green	null
2	21	Yellow	null
3	25	White	null
4	37	Maroon	null

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

```sql
SELECT
    COALESCE(FREQUENCY_OF_PURCHASES, 'Unknown') AS Frequency,
    SUM(COALESCE(purchase_amount, 0)) AS Total_Purchase_Amount
FROM shoppingtrends_db
GROUP BY COALESCE(FREQUENCY_OF_PURCHASES, 'Unknown');
```

Results | Chart

	FREQUENCY	TOTAL_PURCHASE_AMOUNT
1	Every 3 Months	1749.0
2	Weekly	2184.0
3	Bi-Weekly	2099.0
4	Monthly	1780.0

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

```
77  SELECT
78      CATEGORY,
79      COUNT(*) AS Total_Purchases
80  FROM shoppingtrends_db
81  WHERE CATEGORY IS NOT NULL
82  GROUP BY CATEGORY;
83
84  SELECT
```

Results | Chart

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

11. Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0. Expected Columns: Location, Total purchase_amount

```
84  SELECT
85      Location,
86      SUM(COALESCE(purchase_amount, 0)) AS Total_Purchase_Amount
87  FROM shoppingtrends_db
88  GROUP BY Location
89  ORDER BY Total_Purchase_Amount DESC
90  LIMIT 5;
```

Results | Chart

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

12. Group customers by Gender and Size, and count how many entries have a NUL L Color. Expected Columns: Gender, Size, Null Color Count

```
93  SELECT
94      GENDER,
95      SIZE,
96      COUNT(*) AS Null_COLOR_Count
97  FROM SHOPPINGTRENDS_DB
98  WHERE COLOR IS NULL
99  GROUP BY GENDER, SIZE;
100
```

Results | Chart

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

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 shoppingtrends_db
WHERE Shipping_Type IS NULL
GROUP BY Item_Purchased
HAVING COUNT(*) > 3;
```

Results | Chart

	ITEM_PURCHASED	NULL_SHIPPING_TYPE_COUNT
1	null	4
2	Shoes	4
3	Shirt	5

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 shoppingtrends_db
WHERE Review_Rating IS NULL
GROUP BY Payment_Method;
```

Results | Chart

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

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

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 shoppingtrends_db
WHERE COLOR IS NULL
GROUP BY COLOR
HAVING COUNT(*) >= 2;
```

Results | Chart

	COLOR	AVERAGE_AGE
1	null	47.8461538

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

```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(DISTINCT "CUSTOMER_ID") AS Customer_Count
FROM shoppingtrends_db
GROUP BY DELIVERY_SPEED;
```

Results | Chart

	DELIVERY_SPEED	CUSTOMER_COUNT
1	Slow	45
2	Other	166
3	Fast	89

18. Find customers whose purchase_amount is NULL and whose Promo Code Used is 'Yes'. Expected Columns: Customer ID, purchase_amount, Promo Code Used

```sql
SELECT
    CUSTOMER_ID,
    PURCHASE_AMOUNT,
    PROMO_CODE_USED
FROM shoppingtrends_db
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

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

```sql
SELECT
    Location,
    MAX(COALESCE(PREVIOUS_PURCHASES, 0)) AS Max_PREVIOUS_PURCHASES,
    AVG(REVIEW_RATING) AS Average_Review_Rating
FROM shoppingtrends_db
GROUP BY Location
HAVING AVG(REVIEW_RATING) > 4.0;
```

Results | Chart

	LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE_REVIEW_RATING
	Query produced no results		

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 shoppingtrends_db
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