**Belinda Thando Mhlanga**

**Practical 3 - Advanced SQL**

## Question 1.

```sql
-- Question 1: Find all records where Size is missing and purchase_amount > 50
SELECT
    Customer_ID,
    Size,
    purchase_amount,
    Item_Purchased
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Size IS NULL
  AND purchase_amount > 50;
```

Results (2 minutes ago)

Table | Chart — 22 rows — 434ms

| | 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 |

## Question 2.

```sql
--Question 2: List 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 SHOP.SHOPING.SHOPING_TRENDS
GROUP BY COALESCE(Season, 'Unknown Season')
ORDER BY Total_Purchases DESC;
```

Results (just now)

Table | Chart — 5 rows — 86ms

| | SEASON | TOTAL_PURCHASES |
|---|---|---|
| 1 | Winter | 80 |
| 2 | Spring | 73 |
| 3 | Summer | 65 |
| 4 | Fall | 55 |
| 5 | Unknown Season | 27 |

## Question 3

```sql
-- Question 3: Count customers by Payment Method, treating NULLs as 'Not Provided'
SELECT
    COALESCE(Payment_Method, 'Not Provided') AS Payment_Method,
    COUNT(DISTINCT Customer_ID) AS Customer_Count
FROM SHOP.SHOPING.SHOPING_TRENDS
GROUP BY COALESCE(Payment_Method, 'Not Provided')
ORDER BY Customer_Count DESC;
```

esults (just now)

Table | Chart — 7 rows — 94ms

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

## Question 4

```sql
-- Question 4: Show customers where Promo Code Used is NULL and Review Rating < 3.0
SELECT
    Customer_ID,
    Promo_Code_Used,
    Review_Rating,
    Item_Purchased
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Promo_Code_Used IS NULL
  AND Review_Rating < 3.0;
```

Results (just now)

Table | Chart — 8 rows — 65ms

| | CUSTOMER_ID | 0\|1 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 |

## Question 5

```sql
--Question 5: Group by Shipping Type, return avg purchase_amount,
--treating missing values as 0
SELECT
    Shipping_Type,
    AVG(COALESCE(purchase_amount, 0)) AS Average_purchase_amount
FROM SHOP.SHOPING.SHOPING_TRENDS
GROUP BY Shipping_Type
ORDER BY Average_purchase_amount DESC;
```

Results (just now)

Table | Chart — 7 rows — 91ms

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

## Question 6

```sql
-- Question 6: Display purchases per Location with >5 purchases and
-- non-NULL Payment Method
SELECT
    Location,
    COUNT(*) AS Total_Purchases
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Payment_Method IS NOT NULL
GROUP BY Location
HAVING COUNT(*) > 5
ORDER BY Total_Purchases DESC;
```

Results (just now)

Table | Chart — 9 rows — 88ms

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

## Question 7

```sql
--Question 7: Create Spender Category column using CASE
--High: >80, Medium: 50-80, Low: otherwise
-- Replace NULLs in purchase_amount with 0
SELECT
    Customer_ID,
    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 SHOP.SHOPING.SHOPING_TRENDS
ORDER BY purchase_amount DESC;
```

Results (just now)

Table | Chart — 300 rows — 65ms

| | CUSTOMER_ID | PURCHASE_AMOUNT | SPENDER_CATEGORY |
|---|---|---|---|
| 1 | 259 | null | Low |
| 2 | 41 | null | Low |
| 3 | 286 | null | Low |
| 4 | 65 | null | Low |
| 5 | 71 | null | Low |
| 6 | 78 | null | Low |

## Question 8

```sql
-- Question 8: Find customers with NULL Previous Purchases but non-NULL Color
SELECT
    Customer_ID,
    Color,
    Previous_Purchases
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Previous_Purchases IS NULL
  AND Color IS NOT NULL;
```

Results (just now)

Table | Chart — 36 rows — 21ms

| | 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 |

**Question 9**

```sql
-- Question 9: Group by Frequency of Purchases, show total amount spent,
-- treating NULL frequencies as 'Unknown'
SELECT
    COALESCE(Frequency_of_Purchases, 'Unknown') AS Frequency_of_Purchases,
    SUM(purchase_amount) AS Total_purchase_amount
FROM SHOP.SHOPING.SHOPING_TRENDS
GROUP BY COALESCE(Frequency_of_Purchases, 'Unknown')
ORDER BY Total_purchase_amount DESC;
```

Results (just now)

Table | Chart — 8 rows — 106ms

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

**Question 10**

```sql
-- Question 10: Display all Category values with purchase counts,
--excluding NULL Category
SELECT
    Category,
    COUNT(*) AS Total_Purchases
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Category IS NOT NULL
GROUP BY Category
ORDER BY Total_Purchases DESC;
```

Results (just now)

Table | Chart — 4 rows — 93ms

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

**Question 11**

```sql
--Question 11: Return top 5 Locations with highest total purchase_amount,
--replacing NULLs in amount with 0
SELECT
    Location,
    SUM(COALESCE(purchase_amount, 0)) AS Total_purchase_amount
FROM SHOP.SHOPING.SHOPING_TRENDS
GROUP BY Location
ORDER BY Total_purchase_amount DESC
LIMIT 5;
```

Results (just now)

Table | Chart — 5 rows — 123ms

| | 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 |

**Question 12**

```sql
-- Question 12: Group by Gender and Size, count entries with NULL Color
SELECT
    Gender,
    Size,
    COUNT(CASE WHEN Color IS NULL THEN 1 END) AS Null_Color_Count
FROM SHOP.SHOPING.SHOPING_TRENDS
GROUP BY Gender, Size
HAVING COUNT(CASE WHEN Color IS NULL THEN 1 END) > 0
ORDER BY Null_Color_Count DESC;
```

Results (just now)

Table | Chart — 5 rows — 94ms

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

**Question 13**

```sql
-- Question 13: Identify Item Purchased where >3 purchases had NULL Shipping Type
SELECT
    Item_Purchased,
    COUNT(CASE WHEN Shipping_Type IS NULL THEN 1 END) AS NULL_Shipping_Type_Count
FROM SHOP.SHOPING.SHOPING_TRENDS
GROUP BY Item_Purchased
HAVING COUNT(CASE WHEN Shipping_Type IS NULL THEN 1 END) > 3
ORDER BY NULL_Shipping_Type_Count DESC;
```

Results (just now)

Table | Chart — 3 rows — 84ms

| | ITEM_PURCHASED | NULL_SHIPPING_TYPE_COUNT |
|---|---|---|
| 1 | Shirt | 5 |
| 2 | null | 4 |
| 3 | Shoes | 4 |

**Question 14**

```sql
-- Question 14: Count customers per Payment Method with NULL Review Rating
SELECT
    Payment_Method,
    COUNT(CASE WHEN Review_Rating IS NULL THEN 1 END) AS Missing_Review_Rating_Count
FROM SHOP.SHOPING.SHOPING_TRENDS
GROUP BY Payment_Method
HAVING COUNT(CASE WHEN Review_Rating IS NULL THEN 1 END) > 0
ORDER BY Missing_Review_Rating_Count DESC;
```

Results (just now)

Table | Chart — 7 rows — 92ms

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

**Question 15**

```sql
-- Question 15: Group by Category, return avg Review Rating (NULLs as 0),filter
where average >3.5
SELECT
    Category,
    AVG(COALESCE(Review_Rating, 0)) AS Average_Review_Rating
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Category IS NOT NULL
GROUP BY Category
HAVING AVG(COALESCE(Review_Rating, 0)) > 3.5
ORDER BY Average_Review_Rating DESC;
```

Results (1 minute ago)

Table | Chart | 0 rows | 104ms

| CATEGORY | AVERAGE_REVIEW_RATING |
|---|---|

Query produced no results

**Question 16**

```sql
-- Question 16: List Colors that are NULL in ≥2 rows with average Age
SELECT
    Color,
    AVG(Age) AS Average_Age
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Color IS NULL
GROUP BY Color
HAVING COUNT(*) >= 2;
```

Results (just now)

Table | Chart | 1 row | 85ms

| | COLOR | AVERAGE_AGE |
|---|---|---|
| 1 | null | 47.8461538 |

**Question 17**

```sql
-- Question 17: Create Delivery Speed column using CASE and count customers. Fast: Express/Next Day Air, Slow: Standard, Other: else/NULL
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 SHOP.SHOPING.SHOPING_TRENDS
GROUP BY
    CASE
        WHEN Shipping_Type IN ('Express', 'Next Day Air') THEN 'Fast'
        WHEN Shipping_Type = 'Standard' THEN 'Slow'
        ELSE 'Other'
    END
ORDER BY Customer_Count DESC;
```

Results (just now)

Table | Chart | 3 rows | 104ms

| | DELIVERY_SPEED | CUSTOMER_COUNT |
|---|---|---|
| 1 | Other | 166 |
| 2 | Fast | 89 |
| 3 | Slow | 45 |

**Question 18**

```sql
-- Question 18: Find customers with NULL purchase_amount and Promo Code Used = 'Yes'
SELECT
    Customer_ID,
    purchase_amount,
    Promo_Code_Used
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE purchase_amount IS NULL
  AND Promo_Code_Used = 'Yes';
```

Results (just now)

Table | Chart — 20 rows — 80ms

| | 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 |

**Question 19**

```sql
-- Question 19: Group by Location, show max Previous Purchases (NULLs as 0),only
where average rating >4.0
SELECT
    Location,
    MAX(COALESCE(Previous_Purchases, 0)) AS Max_Previous_Purchases,
    AVG(Review_Rating) AS Average_Review_Rating
FROM SHOP.SHOPING.SHOPING_TRENDS
WHERE Review_Rating IS NOT NULL  -- Ensures we only average non-NULL ratings
GROUP BY Location
HAVING AVG(Review_Rating) > 4.0
ORDER BY Average_Review_Rating DESC;
```

Results (just now)

Table | Chart — 0 rows — 35ms

| LOCATION | MAX_PREVIOUS_PURCHASES | AVERAGE_REVIEW_RATING |
|---|---|---|

Query produced no results

**Question 20**

```
211  --Show customers who have a NULL Shipping
212  --Type but made a purchase in the range of 30 to 70 USD.
213  SELECT
214      Customer_ID,
215      Shipping_Type,
216      purchase_amount,
217      Item_Purchased
218  FROM SHOP.SHOPING.SHOPING_TRENDS
219  WHERE Shipping_Type IS NULL
220    AND purchase_amount BETWEEN 30 AND 70
221  ORDER BY purchase_amount;
```

Results (just now)

Table | Chart | 7 rows | 78ms

| | CUSTOMER_ID | SHIPPING_TYPE | PURCHASE_AMOUNT | ITEM_PURCHASED |
|---|---|---|---|---|
| 1 | 293 | null | 35.0 | null |
| 2 | 213 | null | 36.0 | Shirt |
| 3 | 141 | null | 37.0 | Shorts |
| 4 | 235 | null | 38.0 | Sandals |
| 5 | 105 | null | 43.0 | Shirt |
| 6 | 15 | null | 54.0 | Jeans |
| 7 | 196 | null | 66.0 | Coat |