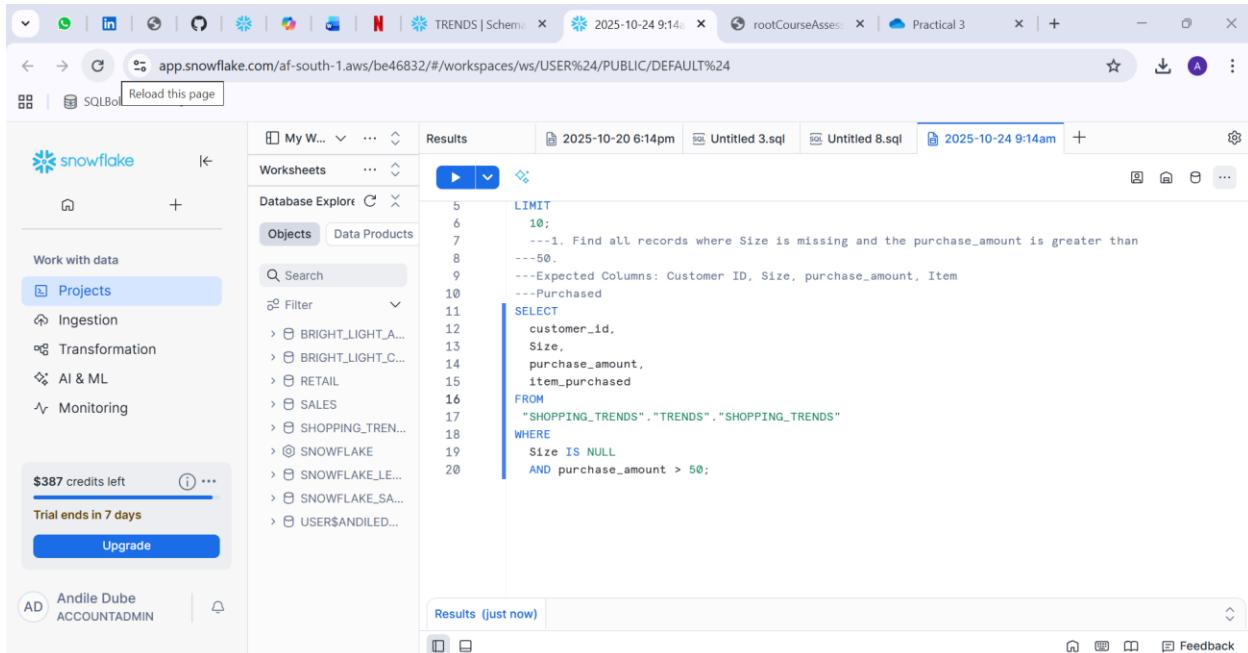


## ANDILE DUBE

### Practical 3: Advanced SQL Null Functions

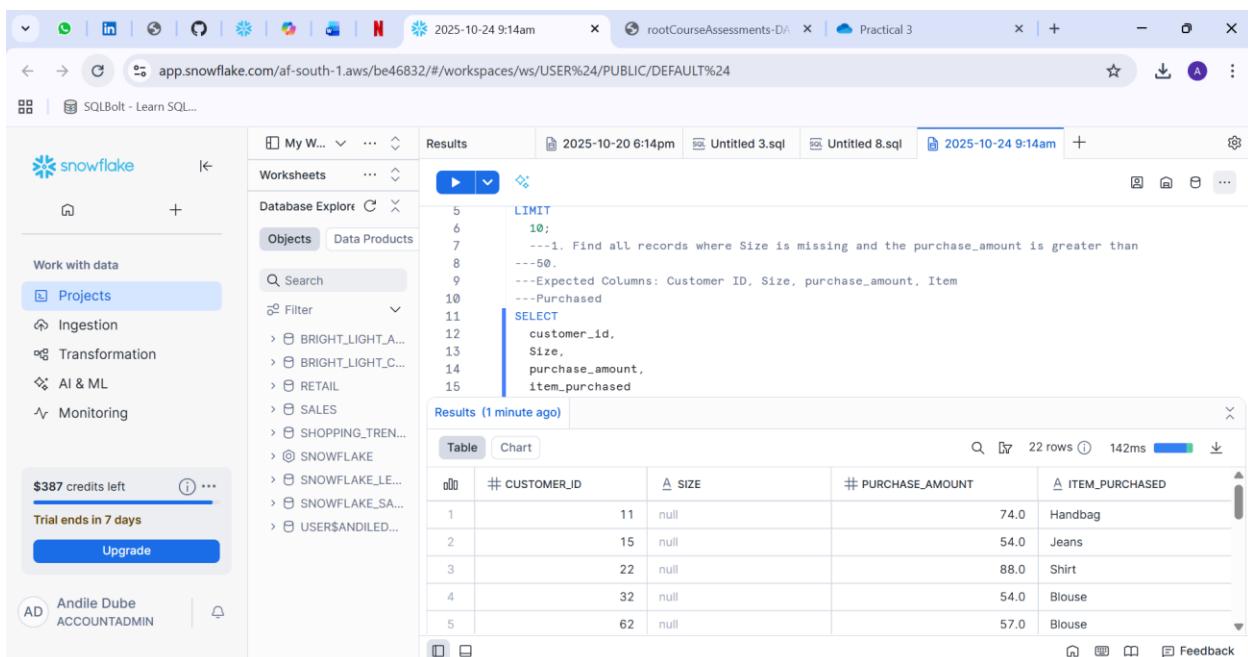
#### Question 1:



The screenshot shows the Snowflake SQL Bolt interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, and Monitoring. It also displays \$387 credits left and a trial ends in 7 days message. On the right, the main area has tabs for Worksheets, Database Explore, Objects (which is selected), and Data Products. A search bar and filter dropdown are present. Below them is a tree view of database objects. The main pane contains a code editor with the following SQL query:

```
5  LIMIT
6  10;
7  ---1. Find all records where Size is missing and the purchase_amount is greater than
8  ---50.
9  ---Expected Columns: Customer ID, Size, purchase_amount, Item
10 ---Purchased
11
12
13
14
15
16
17
18
19
20
```

SELECT  
customer\_id,  
Size,  
purchase\_amount,  
item\_purchased  
FROM  
"SHOPPING\_TRENDS"."TRENDS"."SHOPPING\_TRENDS"  
WHERE  
Size IS NULL  
AND purchase\_amount > 50;



The screenshot shows the same Snowflake SQL Bolt interface after the query has been run. The results tab is active, showing the output of the query. The results table has columns CUSTOMER\_ID, SIZE, PURCHASE\_AMOUNT, and ITEM\_PURCHASED. There are 22 rows returned, each with a customer ID, a null value for size, and a purchase amount between 54.0 and 88.0. The item purchased is either Handbag, Jeans, Shirt, Blouse, or another Blouse entry.

| CUSTOMER_ID | SIZE | PURCHASE_AMOUNT | ITEM_PURCHASED |
|-------------|------|-----------------|----------------|
| 1           | null | 74.0            | Handbag        |
| 2           | null | 54.0            | Jeans          |
| 3           | null | 88.0            | Shirt          |
| 4           | null | 54.0            | Blouse         |
| 5           | null | 57.0            | Blouse         |
| ...         |      |                 |                |

#### Question 2:

SQLBolt - Learn SQL...

```

20     AND purchase_amount > 50;
21 ---2. List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'.
22
23 ---Expected Columns: Season, Total Purchases
24
25 SELECT
26     COALESCE(Season, 'Unknown Season') AS Season,
27     COUNT(*) AS Total_Purchases
28
29 FROM
30     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
31 GROUP BY
32     COALESCE(Season, 'Unknown Season');

```

**Results (just now)**

| SEASON         | TOTAL_PURCHASES |
|----------------|-----------------|
| Summer         | 65              |
| Winter         | 80              |
| Fall           | 55              |
| Spring         | 73              |
| Unknown Season | 27              |

### Question 3:

SQLBolt - Learn SQL...

```

30 ---3. Count how many customers used each Payment Method, treating NULLS as
31 ---'Not Provided'.
32 ---Expected Columns: Payment Method, Customer Count
33
34 SELECT
35     COALESCE(payment_method, 'Not Provided') AS payment_method,
36     COUNT(*) AS "Customer Count"
37
38 FROM
39     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
40 GROUP BY
41     COALESCE(payment_method, 'Not Provided');

```

**Results (just now)**

| PAYMENT_METHOD | Customer Count |
|----------------|----------------|
| Credit Card    | 44             |
| Not Provided   | 30             |
| PayPal         | 51             |
| Debit Card     | 42             |
| Cash           | 42             |

### Question 4:

The screenshot shows the Snowflake SQL Bolt interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, and Monitoring. A message says '\$387 credits left' and 'Trial ends in 7 days'. Below that is a user profile for 'Andile Dube' (ACCOUNTADMIN). The main area has tabs for Worksheets, Database Explore (selected), Objects, and Data Products. A search bar and filter dropdown are also present. The 'Objects' tab is selected. A code editor window shows a query:

```

43   SELECT
44     customer_id,
45     promo_code_used,
46     review_rating,
47     item_purchased
48   FROM
49     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
50   WHERE
51     promo_code_used IS NULL
52     AND review_rating < 3.0;
53   Ctrl+I to generate

```

The results table shows 8 rows with the following data:

|   | CUSTOMER_ID | 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       |

## Question 5:

The screenshot shows the Snowflake SQL Bolt interface. The sidebar and user info are identical to the previous screenshot. The 'Database Explore' tab is selected. A code editor window shows a query:

```

55   --Expected Columns: Shipping Type, Average purchase_amount
56
57   SELECT
58     shipping_type,
59     AVG(COALESCE(purchase_amount, 0)) AS "Average purchase_amount"
60   FROM
61     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
62   GROUP BY
63     shipping_type;

```

The results table shows 7 rows with the following data:

| SHIPPING_TYPE | Average purchase_amount |
|---------------|-------------------------|
| Standard      | 47.6666667              |
| Express       | 53.4545455              |
| Store Pickup  | 55.3333333              |
| null          | 52.7037037              |
| Free Shipping | 50.2142857              |

## Question 6:

```

53. Display the number of purchases per location only for those with more than 5 purchases and no NULL Payment Method.
54. ---Expected Columns: Location, Total Purchases
55.
56. SELECT
57.     location,
58.     COUNT(*) AS "Total Purchases"
59. FROM
60.     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
61. WHERE
62.     payment_method IS NOT NULL
63. GROUP BY
64.     location
65. HAVING
66.     COUNT(*) > 5;

```

**Results (1 minute ago)**

| LOCATION | Total Purchases |
|----------|-----------------|
| Maine    | 41              |
| Kentucky | 30              |
| null     | 24              |

## Question 7:

```

1. Create a column Spender Category that classifies customers using CASE
2. ---'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80,
3. ---'Low' otherwise. Replace NULLs in purchase_amount with 0.
4. ---Expected Columns: Customer ID, purchase_amount, Spender Category
5.
6. SELECT
7.     customer_id,
8.     COALESCE(purchase_amount, 0) AS purchase_amount,
9.     CASE
10.        WHEN COALESCE(purchase_amount, 0) > 80 THEN 'High'
11.        WHEN COALESCE(purchase_amount, 0) BETWEEN 50 AND 80 THEN 'Medium'
12.        ELSE 'Low'
13.    END AS "Spender Category"
14. FROM
15.     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS" ;

```

**Results (Just now)**

| CUSTOMER_ID | PURCHASE_AMOUNT | Spender Category |
|-------------|-----------------|------------------|
| 1           | 20.0            | Low              |
| 2           | 21.0            | Low              |
| 3           | 27.0            | Low              |

## Question 8:

The screenshot shows the Snowflake SQL Bolt interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, and Monitoring. A message says '\$387 credits left' and 'Trial ends in 7 days'. Below that is a user profile for 'Andile Dube' with the role 'ACCOUNTADMIN'. The main area has tabs for Worksheets, Database Explore, and Objects. The Database Explore tab is active, showing a search bar and a tree view of objects. A code editor window displays the following SQL query:

```

92  ---Purchases value but whose Color is not NULL.
93  ---Expected Columns: Customer ID, Color, Previous Purchases
94  SELECT
95    customer_id,
96    Color,
97    previous_purchases
98  FROM
99    "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
100 WHERE
101   previous_purchases IS NULL
102   AND Color IS NOT NULL;

```

The results table shows 36 rows with columns CUSTOMER\_ID, COLOR, and PREVIOUS\_PURCHASES. The data is as follows:

| CUSTOMER_ID | COLOR | PREVIOUS_PURCHASES |
|-------------|-------|--------------------|
| 1           | 8     | null               |
| 2           | 21    | null               |
| 3           | 25    | null               |
| 4           | 37    | null               |
| 5           | 66    | null               |

## Question 9:

This screenshot is identical to the one above, showing the same interface and the same SQL query and results. The results table is identical, showing 36 rows with CUSTOMER\_ID, COLOR, and PREVIOUS\_PURCHASES columns.

## Question 10:

The screenshot shows the Snowflake web interface. On the left, there's a sidebar with 'Work with data' sections like 'Ingestion', 'Transformation', 'AI & ML', and 'Monitoring'. A message at the top says '\$387 credits left' and 'Trial ends in 7 days' with a 'Upgrade' button. The main area has tabs for 'Results', '2025-10-20 6:14pm', 'Untitled 3.sql', 'Untitled 8.sql', and '2025-10-24 9:14am'. The 'Results' tab is active, displaying a SQL query and its output.

```

116    SELECT
117      Category,
118      COUNT(*) AS "Total Purchases"
119    FROM
120      "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
121    WHERE
122      Category IS NOT NULL
123    GROUP BY
124      Category;
125

```

**Results (just now)**

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

## Question 11:

This screenshot is similar to the previous one but shows a different query. The sidebar and top navigation are identical. The 'Results' tab is active, showing a query that includes a comment about expected columns and a result table.

```

128    --Expected Columns: Location, Total purchase_amount
129    SELECT
130      Location,
131      SUM(COALESCE(purchase_amount, 0)) AS "Total purchase_amount"
132    FROM
133      "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
134    WHERE
135      Location IS NOT NULL
136    GROUP BY
137      Location
138    ORDER BY
139      "Total purchase_amount" DESC
140    LIMIT 5;

```

**Results (just now)**

|   | LOCATION      | Total purchase_amount |
|---|---------------|-----------------------|
| 1 | Maine         | 2294.0                |
| 2 | Florida       | 1980.0                |
| 3 | Massachusetts | 1899.0                |

## Question 12:

The screenshot shows the Snowflake SQL Bolt interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, and Monitoring. A message indicates '\$387 credits left' and 'Trial ends in 7 days'. Below that is a user profile for 'Andile Dube' with the role 'ACCOUNTADMIN'. The main area has tabs for 'Results', '2025-10-20 6:14pm', 'Untitled 3.sql', 'Untitled 8.sql', and '2025-10-24 9:14am'. The code editor contains a query:

```

143   SELECT
144     Gender,
145     Size,
146     COUNT(*) AS "Null Color Count"
147   FROM
148     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
149   WHERE
150     Color IS NULL
151   GROUP BY
152     Gender,
153     Size;
154

```

The results table shows 5 rows:

|   | GENDER | SIZE | Null Color Count |
|---|--------|------|------------------|
| 1 | Male   | null | 6                |
| 2 | Male   | M    | 7                |
| 3 | Male   | L    | 6                |
| 4 | Male   | S    | 5                |

### Question 13:

The screenshot shows the Snowflake SQL Bolt interface. The sidebar and user info are identical to the previous screenshot. The main area has tabs for 'Results', '2025-10-20 6:14pm', 'Untitled 3.sql', 'Untitled 8.sql', and '2025-10-24 9:14am'. The code editor contains a query:

```

152   Gender,
153   Size;
154   ---13. Identify all Item Purchased where more than 3 purchases had NULL ShippingType.
155   ---Expected Columns: Item Purchased, NULL Shipping Type Count
156
157   SELECT
158     Item_purchased,
159     COUNT(*) AS "NULL Shipping Type Count"
160   FROM
161     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
162   WHERE
163     SHIPPING_TYPE IS NULL
164     AND Item_purchased IS NOT NULL
165   GROUP BY
166     Item_purchased
167     HAVING
168     COUNT(*) > 3;

```

The results table shows 2 rows:

|   | ITEM_PURCHASED | NULL Shipping Type Count |
|---|----------------|--------------------------|
| 1 | Shirt          | 5                        |
| 2 | Shoes          | 4                        |

### Question 14:

The screenshot shows the Snowflake SQL Bolt interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, and Monitoring. A message indicates '\$387 credits left' and 'Trial ends in 7 days'. The main area has tabs for 'Results', '2025-10-20 6:14pm', 'Untitled 3.sql', 'Untitled 8.sql', and '2025-10-24 9:14am'. The code editor shows a query:

```
170 ---Expected Columns: Payment Method, Missing Review Rating Count
171
172 SELECT
173     payment_method,
174     COUNT(*) AS "Missing Review Rating Count"
175 FROM
176     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
177 WHERE
178     review_rating IS NULL
179     AND payment_method IS NOT NULL
180 GROUP BY
181     payment_method;
```

The results table shows the following data:

| PAYMENT_METHOD | Missing Review Rating Count |
|----------------|-----------------------------|
| Credit Card    | 8                           |
| Cash           | 4                           |
| Debit Card     | 7                           |
| Venmo          | 9                           |

### Question 15:

The screenshot shows the Snowflake SQL Bolt interface. The sidebar and credits message are identical to the previous screenshot. The main area shows a query:

```
182 ---Expected Columns: Category, Average Review Rating
183
184
185
186
187
188
189
190
191
192
193
194
195
```

The results table shows the following data:

| CATEGORY                  | Average Review Rating |
|---------------------------|-----------------------|
| Query produced no results |                       |

### Question 16:

The screenshot shows the SQLBolt interface with a query editor and results viewer. The query is:

```

197 --- Rows showing average age of customers for each color.
198 --- Expected Columns: Color, Average Age
199
200 SELECT
201     Color,
202         AVG(Age) AS "Average Age"
203     FROM
204     "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
205     WHERE
206         Color IS NULL
207     GROUP BY
208         Color
209     HAVING
210         COUNT(*) >= 2;

```

The results table shows one row:

| Color | Average Age |
|-------|-------------|
| null  | 47.8461538  |

### Question 17:

The screenshot shows the SQLBolt interface with a query editor and results viewer. The query is:

```

209 ---17. Use CASE to create a column Delivery Speed: 'Fast' if Shipping Type is 'Express' or
210     ---'Next Day Air', 'Slow' if 'Standard',
211     ---'Other' for all else including NULL. Then count how many customers fall into
212     ---each category.
213     |
214     SELECT
215         CASE
216             WHEN shipping_type IN ('Express', 'Next Day Air') THEN 'Fast'
217             WHEN shipping_type = 'Standard' THEN 'Slow'
218             ELSE 'Other'
219         END AS "Delivery Speed",
220         COUNT(*) AS "Customer Count"
221     FROM
222         "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
223     GROUP BY
224         "Delivery Speed";

```

The results table shows three rows:

| Delivery Speed | Customer Count |
|----------------|----------------|
| Other          | 166            |
| Fast           | 89             |

### Question 18:

The screenshot shows the Snowflake SQLBolt interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, and Monitoring. A trial message says '\$387 credits left' and 'Trial ends in 7 days'. Below it, the user is identified as 'Andile Dube' with the role 'ACCOUNTADMIN'. The main area has tabs for 'My W...', 'Worksheets', and 'Database Explore'. The 'Database Explore' tab is active, showing a results table for a query. The query itself is:

```

225 ---'Yes'.
226 ---Expected Columns: Customer_ID, purchase_amount, Promo_Code_Used
227
228
229
230
231
232
233
SELECT Customer_ID, purchase_amount, Promo_Code_Used
FROM
    "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
WHERE purchase_amount IS NULL
    AND Promo_Code_Used = 'Yes';

```

The results table shows 5 rows:

| # CUSTOMER_ID | # PURCHASE_AMOUNT | 0 1 PROMO_CODE_USED |
|---------------|-------------------|---------------------|
| 1             | 13                | null TRUE           |
| 2             | 30                | null TRUE           |
| 3             | 78                | null TRUE           |
| 4             | 95                | null TRUE           |
| 5             | 124               | null TRUE           |

### Question 19:

This screenshot is similar to the previous one but shows an error. The user has run a query that includes a function call 'COALESCE' which is causing a compilation error. The error message is: 'SQL compilation error: error line 238 at position 8 not enough arguments for function [COALESCE(SHOPPING\_TRENDS.PREVIOUS\_PURCHASES)], expected 2, got 1'. The query is:

```

235 ---Review Rating
236
237
238
239
240
241
242
243
244
245
SELECT
    Location,
    MAX(COALESCE (previous_purchases),0) AS Max_previous_purchases,
    AVG(review_rating) AS Average_review_rating
FROM
    "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
GROUP BY
    Location
HAVING
    AVG(review_rating) > 4.0;

```

### Question 20:

The screenshot shows the Snowflake SQL Bolt interface. On the left, there's a sidebar with navigation links like 'Work with data' (Projects, Ingestion, Transformation, AI & ML, Monitoring), a credit balance (\$387), and user information (Andile Dube, ACCOUNTADMIN). The main area has tabs for 'Results' (selected) and 'Database Explorer'. A SQL query is being run:

```
250 SELECT
251   customer_id,
252   shipping_type,
253   purchase_amount,
254   item_purchased
255 FROM
256   "SHOPPING_TRENDS"."TRENDS"."SHOPPING_TRENDS"
257 WHERE
258   shipping_type IS NULL
259   AND purchase_amount BETWEEN 30 AND 70;
260
```

The results table shows 7 rows of data:

|   | 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          |