

1.

The screenshot shows the Snowflake interface with the 'Practical 3 - Snowflake' tab selected. The left sidebar lists databases: CASESTUDY, ORDERS_PRODUCTS_C..., RETAIL, SALES, SHOPPING, SHOPPINGTRENDS, SNOWFLAKE, SNOWFLAKE_LEARNIN..., and SNOWFLAKE_SAMPLE.... The current worksheet is titled 'Practical 3' and contains the following SQL query:

```
--Question 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 TRENDS  
WHERE SIZE IS NULL AND PURCHASE_AMOUNT > 50;
```

The results table shows 10 rows of data:

CUSTOMER_ID	SIZE	PURCHASE_AMOUNT	ITEM_PURCHASED
11	null	74.0	Handbag
15	null	54.0	Jeans
22	null	88.0	Shirt
32	null	54.0	Blouse
62	null	57.0	Blouse
73	null	65.0	Sandals
91	null	54.0	Shoes
97	null	56.0	Shoes
100	null	55.0	Sneakers
160	null	84.0	Coat

2.

The screenshot shows the Snowflake interface with the 'Practical 3 - Snowflake' tab selected. The left sidebar lists the same databases as the previous screenshot. The current worksheet is titled 'Practical 3' and contains the following SQL query:

```
--Question 2  
--List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'. Expected Columns: Season, Total Purchases  
SELECT  
COALESCE(SEASON, 'Unknown Season') AS SEASON,  
COUNT(*) AS TOTAL_PURCHASES  
FROM TRENDS  
GROUP BY COALESCE(SEASON, 'Unknown Season');
```

The results table shows 5 rows of data:

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

3.

The screenshot shows the Snowflake web interface. The left sidebar lists databases and worksheets. The main area displays a query in the code editor:

```
24
25 --Question 3
26 --Count how many customers used each Payment Method, treating NULLs as 'Not Provided'. Expected Columns: Payment Method,
27 Customer Count
28
29 SELECT
30 COALESCE(PAYMENT_METHOD, 'Not Provided') AS PAYMENT_METHOD,
31 COUNT(DISTINCT customer_id) AS customer_count
32 FROM TRENDS
33 GROUP BY COALESCE(PAYMENT_METHOD, 'Not Provided');
```

The results table shows the count of customers for each payment method:

PAYMENT_METHOD	CUSTOMER_COUNT
PayPal	51
Bank Transfer	38
Debit Card	42
Venmo	53
Not Provided	30
Cash	42
Credit Card	44

4.

The screenshot shows the Snowflake web interface. The left sidebar lists databases and worksheets. The main area displays a query in the code editor:

```
34
35 --Question 4
36 --Show customers where Promo Code Used is NULL and Review Rating is below 3.0. Expected Columns: Customer ID, Promo Code
37 Used, Review Rating, Item Purchased
38
39 SELECT CUSTOMER_ID, PROMO_CODE_USED, REVIEW_RATING, ITEM_PURCHASED
40 FROM TRENDS
41 WHERE PROMO_CODE_USED IS NULL AND REVIEW_RATING < 3.0;
```

The results table shows customers with null promo codes and review ratings below 3.0:

CUSTOMER_ID	PROMO_CODE_USED	REVIEW_RATING	ITEM_PURCHASED
21	null	2.5	Jeans
38	null	2.6	Jeans
61	null	2.5	Jeans
80	null	2.6	Sneakers
125	null	2.8	Sneakers
128	null	2.5	Shoes
180	null	2.5	Shorts
285	null	2.9	Blouse

5.

The screenshot shows the Snowflake web interface. The left sidebar lists databases like CASESTUDY, ORDERS_PRODUCTS_C..., RETAIL, SALES, SHOPPING, SHOPPINGTRENDS, SNOWFLAKE, SNOWFLAKE_LEARNIN..., and SNOWFLAKE_SAMPLE.... The current worksheet is titled "Test 1". The code editor contains the following SQL query:

```
--Question 5  
--Group customers by Shipping Type, and return the average purchase_amount, treating missing values as 0. Expected Columns:  
Shipping Type, Average purchase_amount  
  
SELECT  
    SHIPPING_TYPE,  
    AVG(COALESCE(PURCHASE_AMOUNT, 0)) AS AVERAGE_PURCHASE_AMOUNT  
FROM TRENDS  
GROUP BY SHIPPING_TYPE;
```

The results table shows the average purchase amount for each shipping type:

SHIPPING_TYPE	AVERAGE_PURCHASE_AMOUNT
Standard	47.66666667
Express	53.4545455
Store Pickup	55.3333333
null	52.7037037
Free Shipping	50.2142857
Next Day Air	54.86666667
2-Day Shipping	51.5576923

6.

The screenshot shows the Snowflake web interface. The left sidebar lists the same databases as the previous screenshot. The current worksheet is titled "Test 1". The code editor contains the following SQL query:

```
--Question 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 TRENDS  
WHERE PAYMENT_METHOD IS NOT NULL  
GROUP BY LOCATION  
HAVING COUNT(*) > 5;
```

The results table shows the total number of purchases for each location:

LOCATION	TOTAL_PURCHASES
Maine	41
Kentucky	30
null	24
New York	31
Oregon	30
Rhode Island	29
Florida	32
Massachusetts	31
Texas	22

7.

The screenshot shows the Snowflake web interface. The top navigation bar includes tabs for 'Test 1', 'Case Study - Coffe eSales', 'Practical 1', 'Practical 2', and 'Practical 3'. The 'Practical 3' tab is active. The sidebar on the left lists various databases and worksheets, with 'CASESTUDY' expanded. The main workspace displays a query for 'SHOPPINGTRENDS.DATASET'. The code is as follows:

```
--Question 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 TRENDS;
```

The results table shows the following data:

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
7	7	50.0 Medium
8	8	29.0 Low

8.

The screenshot shows the Snowflake web interface. The top navigation bar includes tabs for 'Test 1', 'Case Study - Coffe eSales', 'Practical 1', 'Practical 2', and 'Practical 3'. The 'Practical 3' tab is active. The sidebar on the left lists various databases and worksheets, with 'CASESTUDY' expanded. The main workspace displays a query for 'SHOPPINGTRENDS.DATASET'. The code is as follows:

```
--Question 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 TRENDS  
WHERE PREVIOUS_PURCHASES IS NULL AND color IS NOT NULL;
```

The results table shows the following data:

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
8	70	White null
9	73	Maroon null
10	75	Pink null

9.

The screenshot shows the Snowflake web interface. The left sidebar lists databases and worksheets. The main area displays a query for 'Question 9' in the 'SHOPPINGTRENDS.DATASET'. The results table shows the frequency of purchases and total purchase amount for various categories.

```
--Question 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 TRENDS
GROUP BY COALESCE(FREQUENCY_OF_PURCHASES, 'Unknown');
```

FREQUENCY_OF_PURCHASES	TOTAL_PURCHASE_AMOUNT
Every 3 Months	1749.0
Weekly	2184.0
Bi-Weekly	2099.0
Monthly	1780.0
Fortnightly	2033.0
Annually	1765.0
Unknown	1518.0
Quarterly	2541.0

10.

The screenshot shows the Snowflake web interface. The left sidebar lists databases and worksheets. The main area displays a query for 'Question 10' in the 'SHOPPINGTRENDS.DATASET'. The results table shows the category and total purchases for each item.

```
--Question 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 TRENDS
WHERE CATEGORY IS NOT NULL
GROUP BY CATEGORY;
```

CATEGORY	TOTAL_PURCHASES
Outerwear	60
Footwear	70
Clothing	59
Accessories	78

11.

The screenshot shows the Snowflake web interface with the URL `app.snowflake.com/af-south-1.aws/ve25906/w4mLDVdgDSAE#query`. The sidebar on the left lists databases like CASESTUDY, ORDERS_PRODUCTS_C..., RETAIL, SALES, SHOPPING, SHOPPINGTRENDS, SNOWFLAKE, SNOWFLAKE_LEARNIN..., and SNOWFLAKE_SAMPLE.... The main area displays a query for Question 11:

```
--Question 11  
--Return the top5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0. Expected Columns:  
Location, Total purchase_amount  
104  
105  
106  
107 SELECT location, SUM(COALESCE(PURCHASE_AMOUNT, 0)) AS TOTAL_PURCHASE_AMOUNT  
108 FROM TRENDS  
109 GROUP BY location  
110 ORDER BY TOTAL_PURCHASE_AMOUNT DESC  
111  
112 LIMIT 5;
```

The results table shows the top 5 locations with their total purchase amounts:

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

12.

The screenshot shows the Snowflake web interface with the URL `app.snowflake.com/af-south-1.aws/ve25906/w4mLDVdgDSAE#query`. The sidebar on the left lists the same databases as the previous screenshot. The main area displays a query for Question 12:

```
--Question 12  
--Group customers by Gender and Size, and count how many entries have a NULL Color. Expected Columns: Gender, Size, Null  
Color Count  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
SELECT  
    GENDER,  
    SIZE,  
    COUNT(*) AS NULL_COLOUR_COUNT  
FROM TRENDS  
WHERE COLOR IS NULL  
GROUP BY GENDER, SIZE;
```

The results table shows the count of entries with NULL color for each gender and size combination:

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

13.

The screenshot shows the Snowflake web interface. The left sidebar lists databases and worksheets. The main area shows a query editor with the following code:

```
123 --Question 13
124 --Identify all Item Purchased where more than 3 purchases had NULL Shipping Type. Expected Columns: Item Purchased, NULL
125 Shipping Type Count
126
127 SELECT
128     ITEM_PURCHASED,
129     COUNT(*) AS null_shipping_type_count
130     FROM TRENDS
131     WHERE SHIPPING_TYPE IS NULL
132     GROUP BY ITEM_PURCHASED
133     HAVING COUNT(*) > 3;
134
```

The results table shows the following data:

ITEM_PURCHASED	null_shipping_type_count
null	4
Shirt	5
Shoes	4

14.

The screenshot shows the Snowflake web interface. The left sidebar lists databases and worksheets. The main area shows a query editor with the following code:

```
134 --Question 14
135 --Show a count of how many customers per Payment Method have NULL Review Rating. Expected Columns: Payment Method, Missing
136 Review Rating Count
137
138 SELECT
139     PAYMENT_METHOD,
140     COUNT(*) AS MISSING_REVIEW_RATING_COUNT
141     FROM TRENDS
142     WHERE REVIEW_RATING IS NULL
143     GROUP BY PAYMENT_METHOD;
144
145
```

The results table shows the following data:

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

15.

The screenshot shows the Snowflake UI with a query editor. The query is:

```
146 --Question 15.
147 --Group by Category and return the average Review Rating, replacing NULLs with 0, and filter only where average is greater
148 than 3.5.Expected Columns: Category, Average Review Rating
149
150     SELECT
151         CATEGORY,
152             ROUND(AVG(COALESCE(REVIEW_RATING, 0)), 2) AS AVERAGE REVIEW RATING
153     FROM TRENDS
154     WHERE CATEGORY IS NOT NULL
155     GROUP BY CATEGORY
156     HAVING ROUND(AVG(COALESCE(REVIEW_RATING, 0)), 2) > 3.5
157     OR COUNT(*) > 0
158     ORDER BY AVERAGE REVIEW RATING DESC;
159
```

The results table shows:

CATEGORY	AVERAGE REVIEW RATING
Outerwear	3.31
Accessories	3.26
Footwear	3.19
Clothing	3.02

16.

The screenshot shows the Snowflake UI with a query editor. The query is:

```
159 --Question 16
160 --List all Colors that are missing (NULL) in at least 2 rows and the average Age of customers for those rows.Expected
161 Columns: Color, Average Age
162
163     SELECT
164         COLOR,
165             AVG(AGE) AS AVERAGE AGE
166     FROM TRENDS
167     WHERE COLOR IS NULL
168     GROUP BY COLOR
169     HAVING COUNT(*) >= 2;]
```

The results table shows:

COLOR	AVERAGE AGE
null	47.8461538

17.

The screenshot shows the Snowflake interface with a query editor tab titled "Practical 3 - Snowflake". The code is as follows:

```
--Question 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_ID
FROM TRENDS
GROUP BY
CASE
WHEN SHIPPING_TYPE IN ('Express', 'Next Day Air') THEN 'Fast'
WHEN SHIPPING_TYPE = 'Standard' THEN 'Slow'
ELSE 'Other'
END;|
```

The results table shows the following data:

DELIVERY_SPEED	CUSTOMER_ID
Fast	89
Slow	45
Other	166

18.

The screenshot shows the Snowflake interface with a query editor tab titled "Practical 3 - Snowflake". The code is as follows:

```
--Question 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 TRENDS
WHERE PURCHASE_AMOUNT IS NULL AND PROMO_CODE_USED = 'Yes';|
```

The results table shows the following data:

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
10	168	null TRUE
11	177	null TRUE

19.

The screenshot shows a Snowflake worksheet titled "Practical 3 - Snowflake". The code in the editor is as follows:

```
--Question19  
--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  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222
```

The results table shows the following data:

LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE REVIEW RATING
Rhode Island	50.0	3.54
Massachusetts	47.0	3.44
Oregon	50.0	3.39
Maine	47.0	3.19
New York	49.0	3.07
Kentucky	46.0	3.06
Florida	49.0	3.05

20.

The screenshot shows a Snowflake worksheet titled "Practical 3 - Snowflake". The code in the editor is as follows:

```
--Question 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  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222
```

The results table shows the following data:

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