

1.

The screenshot shows the Snowflake web interface. On the left, the sidebar includes sections like 'Work with data' (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), 'Horizon Catalog' (Catalog, Data sharing, Governance & security), and 'Manage' (Compute). A message at the bottom says '\$396 credits left' and 'Trial ends in 29 days'. The top navigation bar shows 'Practical 1 - Snowflake' and the URL 'app.snowflake.com/af-south-1.aws/ve25906/wsDaz3faaZb#query'. The main area has tabs 'Test 1' and 'Practical 1'. The code editor contains the following SQL:

```
--SELECT Statement  
--Question 1.  
--Display all columns for all transactions. Expected output: All columns  
SELECT *  
FROM RETAIL.SALES.RETAIL_SALES;
```

The results table shows 12 rows of transaction data:

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
5	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
7	7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
8	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
10	10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
11	11	2023-02-14	CUST011	Male	23	Clothing	2	50	100
12	12	2023-10-30	CUST012	Male	35	Beauty	3	25	75

2.

The screenshot shows the Snowflake web interface, identical to the first one but with different query content. The sidebar and navigation bar are the same. The main area has tabs 'Test 1' and 'Practical 1'. The code editor contains the following SQL:

```
--Question 2.  
--Display only the Transaction ID, Date, and Customer ID for all records. Expected output: Transaction ID, Date, Customer ID  
SELECT TRANSACTION_ID  
      , DATE  
      , CUSTOMER_ID  
  FROM RETAIL.SALES.RETAIL_SALES;
```

The results table shows 10 rows of transaction data:

	DATE	CUSTOMER_ID
1		CUST001
2		CUST002
3		CUST003
4		CUST004
5		CUST005
6		CUST006
7		CUST007
8		CUST008
9		CUST009
10		CUST010

3.

The screenshot shows the Snowflake web interface. On the left, there's a sidebar with 'Work with data' sections like Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace, Catalog, Data sharing, Governance & security, Manage, and Compute. A message at the bottom says '\$396 credits left' and 'Trial ends in 29 days' with an 'Upgrade' button. The main area has tabs 'Test 1' and 'Practical 1'. The current tab is 'Practical 1'. It displays a query in the code editor:

```
19
20
21 --Question 3.
22 --Display all the distinct product categories in the dataset. Expected output: Product Category
23
24
25 SELECT DISTINCT Product_Category
FROM RETAIL.SALES.RETAIL_SALES
26
27
28
29
30
31
32
33
34
35
36
37
```

The results table shows the following data:

PRODUCT_CATEGORY
Clothing
Beauty
Electronics

4.

The screenshot shows the Snowflake web interface. The sidebar and layout are identical to the previous screenshot. The main area has tabs 'Test 1' and 'Practical 1'. The current tab is 'Practical 1'. It displays a query in the code editor:

```
27
28
29 --Question 4.
30 --Display all the distinct gender values in the dataset. Expected output: Gender
31
32
33
34
35
36
37
```

The results table shows the following data:

GENDER
Male
Female

5.

The screenshot shows the Snowflake web interface. On the left, the sidebar is open with 'Work with data' selected, showing options like Projects, Ingestion, Transformation, AI & ML, Monitoring, and Marketplace. A message at the bottom left says '\$396 credits left' and 'Trial ends in 29 days'. The main area has a tab bar with 'Test 1' and 'Practical 1'. The query editor contains the following SQL code:

```
36
37 --Question 5.
38 --Display all transactions where the Age is greater than 40. Expected output: All columns
39
40 SELECT *
41 FROM RETAIL.SALES.RETAIL_SALES
42 WHERE Age > 40;
```

The results table shows 10 rows of data from the RETAIL_SALES table, filtered by age > 40. The columns are: TRANSACTION_ID, DATE, CUSTOMER_ID, GENDER, AGE, PRODUCT_CATEGORY, QUANTITY, PRICE_PER_UNIT, and TOTAL_AMOUNT. The data includes various products like Electronics, Clothing, and Beauty items.

TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	2023-01-13	CUST003	Male	50	Electronics	1	30	30
2	2023-04-25	CUST006	Female	45	Beauty	1	30	30
3	2023-03-13	CUST007	Male	46	Clothing	2	25	50
4	2023-12-13	CUST009	Male	63	Electronics	2	300	600
5	2023-10-07	CUST010	Female	52	Clothing	4	50	200
6	2023-01-17	CUST014	Male	64	Clothing	4	30	120
7	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
8	2023-04-30	CUST018	Female	47	Electronics	2	25	50
9	2023-09-16	CUST019	Female	62	Clothing	2	25	50
10	2023-01-14	CUST021	Female	50	Beauty	1	500	500

6.

The screenshot shows the Snowflake web interface. The sidebar and layout are identical to the previous screenshot. The main area has a tab bar with 'Test 1' and 'Practical 1'. The query editor contains the following SQL code:

```
44
45 --Question 6.
46 --Display all transactions where the Price per Unit is between 100 and 500. Expected output: All columns
47
48 SELECT *
49 FROM RETAIL.SALES.RETAIL_SALES
50 WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
```

The results table shows 10 rows of data from the RETAIL_SALES table, filtered by price_per_unit between 100 and 500. The columns are: TRANSACTION_ID, DATE, CUSTOMER_ID, GENDER, AGE, PRODUCT_CATEGORY, QUANTITY, PRICE_PER_UNIT, and TOTAL_AMOUNT. The data includes various products like Electronics, Clothing, and Beauty items.

TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	2023-05-21	CUST004	Male	37	Clothing	1	500	500
3	2023-12-13	CUST009	Male	63	Electronics	2	300	600
4	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
5	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
6	2023-02-17	CUST016	Male	19	Clothing	3	500	1500
7	2023-11-05	CUST020	Male	22	Clothing	3	300	900
8	2023-01-14	CUST021	Female	50	Beauty	1	500	500
9	2023-11-29	CUST024	Female	49	Clothing	1	300	300
10	2023-10-07	CUST026	Female	28	Electronics	2	500	1000

7.

The screenshot shows the Snowflake web interface. On the left, the sidebar has 'Work with data' expanded, showing 'Projects' selected. The main area shows a query in the 'Practical 1' workspace:

```
--Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.Expected output: All columns
SELECT *
FROM RETAIL.SALES.RETAIL_SALES
WHERE PRODUCT_CATEGORY IN ('Beauty', 'Electronics');
```

The results table shows transactions for Beauty and Electronics categories. The 'PRODUCT_CATEGORY' column is highlighted in blue.

# TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	2023-05-06	CUST005	Male	30	Beauty	2	50	100
4	2023-04-25	CUST006	Female	45	Beauty	1	30	30
5	2023-02-22	CUST008	Male	30	Electronics	4	25	100
6	2023-12-13	CUST009	Male	63	Electronics	2	300	600
7	2023-10-30	CUST012	Male	35	Beauty	3	25	75
8	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
9	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
10	2023-04-30	CUST018	Female	47	Electronics	2	25	50

8.

The screenshot shows the Snowflake web interface. The sidebar is identical to the previous screenshot. The main area shows a query in the 'Practical 1' workspace:

```
--Question 8.
--Display all transactions where the Product Category is not 'Clothing'. Expected output: All columns
SELECT *
FROM RETAIL.SALES.RETAIL_SALES
WHERE PRODUCT_CATEGORY NOT IN ('CLOTHING');
```

The results table shows transactions for categories other than Clothing. The 'PRODUCT_CATEGORY' column is highlighted in blue.

# TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
10	2023-10-07	CUST010	Female	52	Clothing	4	50	200

9.

The screenshot shows the Snowflake web interface. On the left, the sidebar has 'Projects' selected under 'Work with data'. The main area shows a query editor titled 'Test 1' with the tab 'Practical 1' selected. The code entered is:

```
68 | Question 9.
69 | --Display all transactions where the Quantity is greater than or equal to 3. Expected output: All columns
70 |
71 | SELECT *
72 | FROM RETAIL.SALES.RETAIL_SALES
73 | WHERE QUANTITY >= 3;
```

Below the code, the results tab is selected, showing a table with 10 rows of transaction data. The columns are: TRANSACTION_ID, DATE, CUSTOMER_ID, GENDER, AGE, PRODUCT_CATEGORY, QUANTITY, PRICE_PER_UNIT, and TOTAL_AMOUNT. The data includes various items like Beauty, Electronics, and Clothing, with quantities ranging from 3 to 500.

10.

The screenshot shows the Snowflake web interface. The sidebar has 'Projects' selected. The main area shows a query editor titled 'Test 1' with the tab 'Practical 1' selected. The code entered is:

```
77 | --Aggregate Functions
78 |
79 | --Question 10.
80 | --Count the total number of transactions. Expected output: Total_Transactions
81 |
82 | SELECT COUNT ('TransactionID') AS Total_Transactions
83 | FROM RETAIL.SALES.RETAIL_SALES;
```

Below the code, the results tab is selected, showing a table with 1 row labeled 'TOTAL_TRANSACTIONS'. The value is 1000.

11.

The screenshot shows the Snowflake web interface. On the left, the sidebar is open with 'Work with data' selected, showing options like Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace, Horizon Catalog, Catalog, Data sharing, Governance & security, Manage, and Compute. A message at the bottom indicates '\$390 credits left' and 'Trial ends in 28 days' with a 'Upgrade' button. The main area has tabs 'Test 1' and 'Practical 1' with 'Practical 1' currently active. The query editor shows the following SQL code:

```
84  
85  
86 --Question 11.  
87 --Find the average Age of customers. Expected output: Average_Age  
88  
89 SELECT AVG(Age) AS Average_Age  
FROM RETAIL.SALES.RETAIL_SALES
```

The results pane shows a single row with the header '# AVERAGE_AGE' and the value '41.392000'.

12.

The screenshot shows the Snowflake web interface, identical to the previous one but with a different query. The sidebar and tabs are the same. The query editor shows the following SQL code:

```
91  
92  
93 --Question 12.  
94 --Find the total quantity of products sold. Expected output: Total_Quantity  
95  
96 SELECT SUM(Quantity) AS Total_Quantity  
FROM RETAIL.SALES.RETAIL_SALES
```

The results pane shows a single row with the header '# TOTAL_QUANTITY' and the value '2514'.

13.

The screenshot shows the Snowflake web interface. On the left, the sidebar is open with 'Work with data' selected, showing options like Projects, Ingestion, Transformation, AI & ML, Monitoring, and Marketplace. The main area displays a query editor titled 'Test 1' under the 'Practical 1' tab. The query is:

```
--Question 13.  
--Find the maximum Total Amount spent in a single transaction. Expected output: Max_Total_Amount  
SELECT MAX(Total_amount) AS Max_Total_Amount  
FROM RETAIL.SALES.RETAIL_SALES;
```

The results section shows a single row with the header '# MAX_TOTAL_AMOUNT' and the value '2000'. At the bottom of the interface, there is a message: '\$390 credits left' and 'Trial ends in 28 days' with a blue 'Upgrade' button.

14.

The screenshot shows the Snowflake web interface. The sidebar is identical to the previous screenshot. The main area displays a query editor titled 'Test 1' under the 'Practical 1' tab. The query is:

```
--Question 14.  
--Find the minimum Price per Unit in the dataset. Expected output: Min_Price_per_Unit  
SELECT MIN(Price_Per_Unit) AS Min_Price_Per_Unit  
FROM RETAIL.SALES.RETAIL_SALES;
```

The results section shows a single row with the header '# MIN_PRICE_PER_UNIT' and the value '25'. At the bottom of the interface, there is a message: '\$390 credits left' and 'Trial ends in 28 days' with a blue 'Upgrade' button.

15.

The screenshot shows the Snowflake web interface. On the left, the sidebar has 'Projects' selected under 'Work with data'. The main area shows a query editor with the following code:

```
--GROUP BY Statement  
--Question 15.  
--Find the number of transactions per Product Category. Expected output: Product Category, Transaction_Count  
SELECT Product_Category,  
       COUNT(Quantity) AS Transactions_Count,  
FROM RETAIL.SALES.RETAIL_SALES  
GROUP BY Product_Category;
```

The results tab displays the following data:

PRODUCT_CATEGORY	TRANSACTIONS_COUNT
Clothing	351
Beauty	307
Electronics	342

16.

The screenshot shows the Snowflake web interface. On the left, the sidebar has 'Projects' selected under 'Work with data'. The main area shows a query editor with the following code:

```
--Question 16.  
--Find the total revenue (Total_Amount) per gender. Expected output: Gender, Total_Revenue  
SELECT Gender,  
       SUM(Total_Amount * Quantity) as Total_Revenue,  
FROM RETAIL.SALES.RETAIL_SALES  
GROUP BY Gender;
```

The results tab displays the following data:

GENDER	TOTAL_REVENUE
Male	677750
Female	705450

17.

The screenshot shows the Snowflake web interface. On the left, the sidebar has 'Work with data' expanded, with 'Projects' selected. The main area shows a code editor with a query:

```
131 GROUP BY Gender;
132
133
134 --Question 17.
135 --Find the average Price per Unit per product category. Expected output: Product Category, Average_Price
136
137
138 SELECT Product_Category,
139     AVG(Price_per_Unit) AS Average_Price
140 FROM RETAIL_SALES.RETAIL_SALES
141 GROUP BY Product_Category;
```

Below the code editor is a results table:

PRODUCT_CATEGORY	AVERAGE_PRICE
Beauty	184.055375
Clothing	174.287749
Electronics	181.900585

18.

The screenshot shows the Snowflake web interface. On the left, the sidebar has 'Work with data' expanded, with 'Projects' selected. The main area shows a code editor with a query:

```
148 SELECT Product_Category,
149     SUM(Price_per_Unit * Quantity) AS Total_Revenue
150 FROM RETAIL_SALES.RETAIL_SALES
151 GROUP BY Product_Category
152 HAVING SUM(Price_per_Unit * Quantity) > 10000;
153
154 --Question 19.
155 --Find the average quantity per product category where the average is more than 2. Expected output: Product Category, Average_Quantity
156
157
```

Below the code editor is a results table:

PRODUCT_CATEGORY	TOTAL_REVENUE
Beauty	143515
Clothing	155580
Electronics	156905

19.

The screenshot shows the Snowflake web interface. On the left, the sidebar is visible with sections like 'Work with data' (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), 'Horizon Catalog' (Catalog, Data sharing, Governance & security), and 'Manage' (Compute). A message at the bottom left says '\$390 credits left' and 'Trial ends in 28 days'. The top navigation bar has tabs for 'rootCourseAssessments-DAT', 'Practical 1 - Snowflake', and 'Syntax error fix'. The main area shows a query editor with a code block:

```
--Question 19.  
--Find the average quantity per product category where the average is more than 2. Expected output: Product Category, Average_Quantity  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
SELECT Product_Category,  
       AVG(Quantity) AS Average_Quantity  
  FROM RETAIL.SALES.RETAIL_SALES  
 GROUP BY Product_Category  
 HAVING AVG(Quantity) > 2;
```

The results table shows three rows:

PRODUCT_CATEGORY	AVERAGE_QUANTITY
Beauty	2.511401
Clothing	2.547009
Electronics	2.482456

20.

The screenshot shows the Snowflake web interface, identical to the previous one in layout and sidebar. The main area shows a query editor with a code block:

```
--CASE Statement  
164  
165  
166  
167  
--Question 20.  
--Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'. Expected output: Transaction ID, Total Amount, Spending_Level  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
SELECT  
      Transaction_ID,  
      Total_Amount,  
      CASE  
        WHEN Total_Amount > 1000 THEN 'High'  
        ELSE 'Low'  
      END AS Spending_Level  
  FROM RETAIL.SALES.RETAIL_SALES;
```

The results table shows six rows:

TRANSACTION_ID	TOTAL_AMOUNT	SPENDING_LEVEL
1	150	Low
2	1000	Low
3	30	Low
4	500	Low
5	100	Low
6	20	Low

21.

The screenshot shows the Snowflake web interface with the following details:

- Header:** portal.brightlighttutorials.co... x rootCourseAssessments-DAT x Practical 1 - Snowflake x Syntax error fix x +
- Toolbar:** app.snowflake.com/af-south-1.aws/ve25906/wsDaz3faaZb#query
- Left Sidebar:** Work with data (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), Horizon Catalog, Catalog, Data sharing, Governance & security.
- Compute Section:** \$390 credits left, Trial ends in 28 days, Upgrade button.
- User Information:** Sibulelo Mafrika, ACCOUNTADMIN.
- Query Editor:** Test 1 tab selected, Practical 1 tab open. The code is:

```
--Question 21.  
--Display a new column called Age_Group that labels customers as:  
--- 'Youth' if Age < 30  
--- 'Adult' if Age is between 30 and 59  
--- 'Senior' if Age >= 60  
--Expected output: Customer_ID, Age, Age_Group  
  
SELECT  
    Customer_ID,  
    Age,  
    CASE  
        WHEN Age < 30 THEN 'Youth'  
        WHEN Age BETWEEN 30 AND 59 THEN 'Adult'  
        WHEN Age >= 60 THEN 'Senior'  
    END AS Age_Group  
FROM RETAIL.SALES.RETAIL_SALES;
```

- Results Tab:** Results tab selected, Chart tab available. The results table is:

	CUSTOMER_ID	AGE	AGE_GROUP
1	CUST001	34	Adult
2	CUST002	26	Youth
3	CUST003	50	Adult
4	CUST004	37	Adult
5	CUST005	30	Adult
6	CUST006	45	Adult
7	CUST007	46	Adult