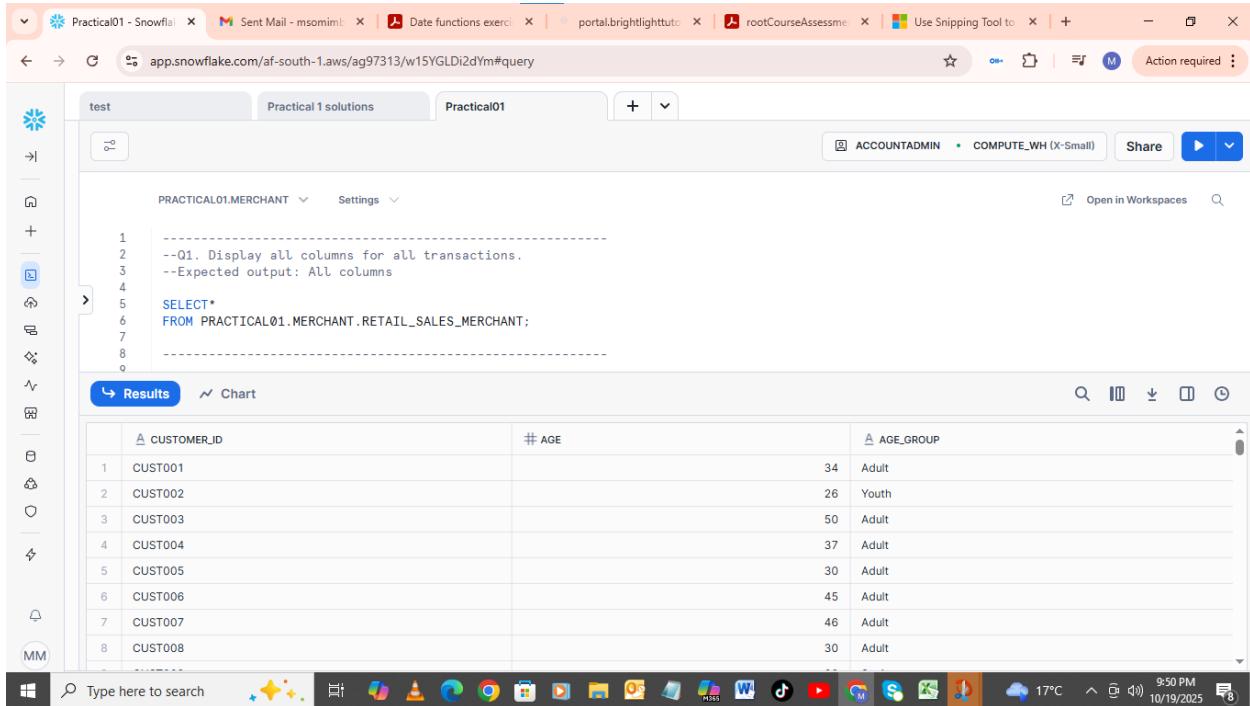


Mbali Zondi

Practical 1: Basic SQL Syntax

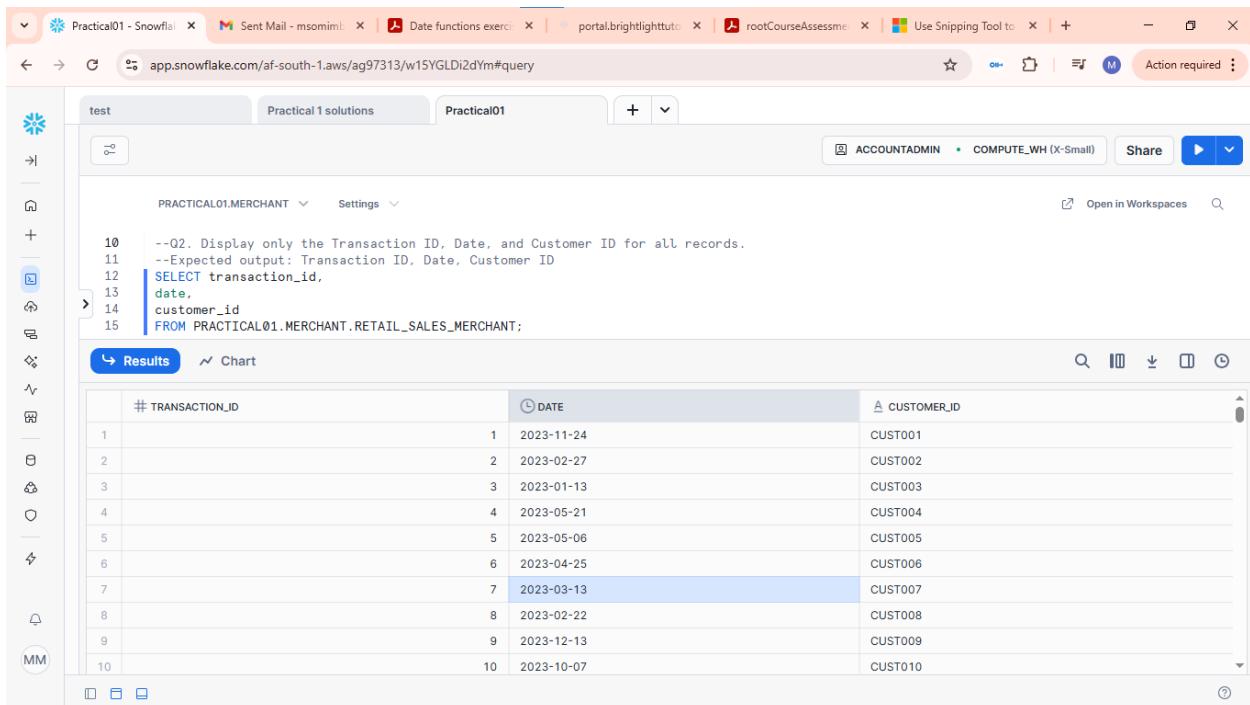
Question 1:



```
PRACTICAL01.MERCHANT
1
2 --Q1. Display all columns for all transactions.
3 --Expected output: All columns
4
5 SELECT*
6   FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
7
8
```

CUSTOMER_ID	AGE	AGE_GROUP
CUST001	34	Adult
CUST002	26	Youth
CUST003	50	Adult
CUST004	37	Adult
CUST005	30	Adult
CUST006	45	Adult
CUST007	46	Adult
CUST008	30	Adult

Question 2:



```
PRACTICAL01.MERCHANT
10
11 --Q2. Display only the Transaction ID, Date, and Customer ID for all records.
12 --Expected output: Transaction ID, Date, Customer ID
13
14   transaction_id,
15   date,
16   customer_id
17   FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
```

TRANSACTION_ID	DATE	CUSTOMER_ID
1	2023-11-24	CUST001
2	2023-02-27	CUST002
3	2023-01-13	CUST003
4	2023-05-21	CUST004
5	2023-05-06	CUST005
6	2023-04-25	CUST006
7	2023-03-13	CUST007
8	2023-02-22	CUST008
9	2023-12-13	CUST009
10	2023-10-07	CUST010

Question 3:

The screenshot shows a Snowflake query editor window. The query being run is:

```
17 -----  
18  
19 --Q3. Display all the distinct product categories in the dataset.  
20 --Expected output: Product Category  
21 | SELECT DISTINCT product_category  
22 | FROM practical01.merchant.retail_sales_merchant;
```

The results pane displays the following data:

PRODUCT_CATEGORY
Clothing
Beauty
Electronics

Question 4:

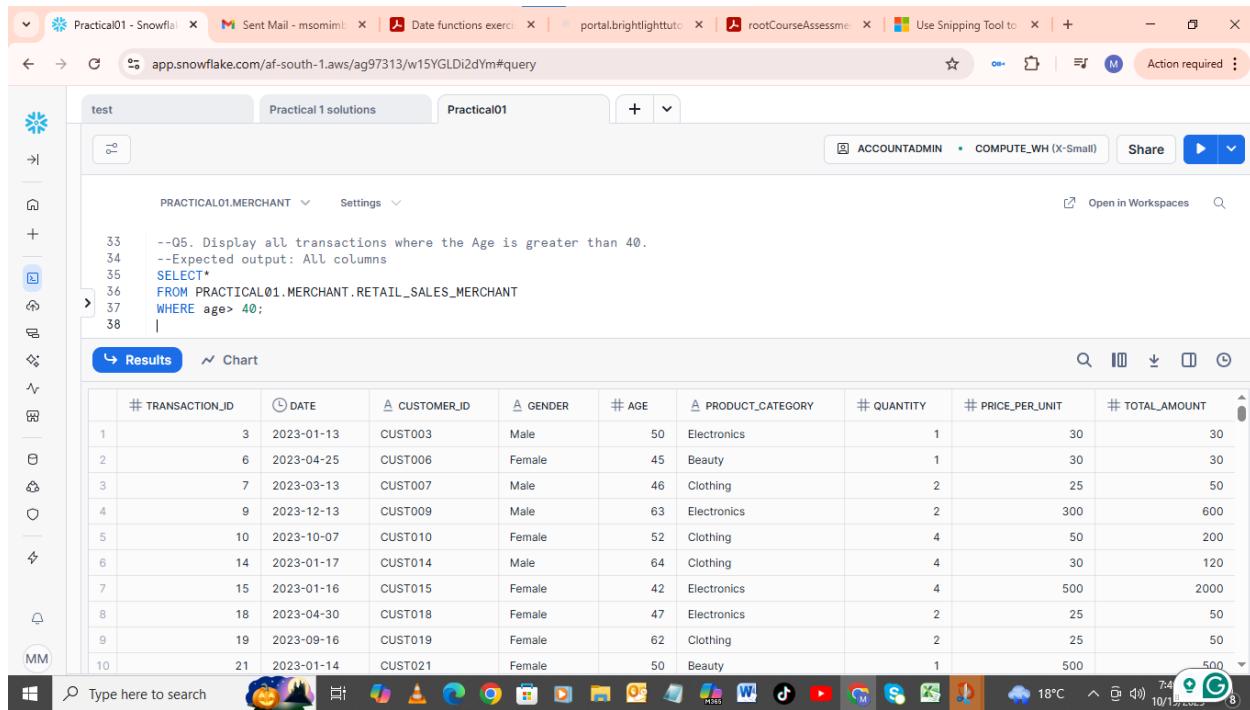
The screenshot shows a Snowflake query editor window. The query being run is:

```
25  
26 --Q4. Display all the distinct gender values in the dataset.  
27 --Expected output: Gender  
28 | SELECT DISTINCT gender  
29 | FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;  
30
```

The results pane displays the following data:

GENDER
Male
Female

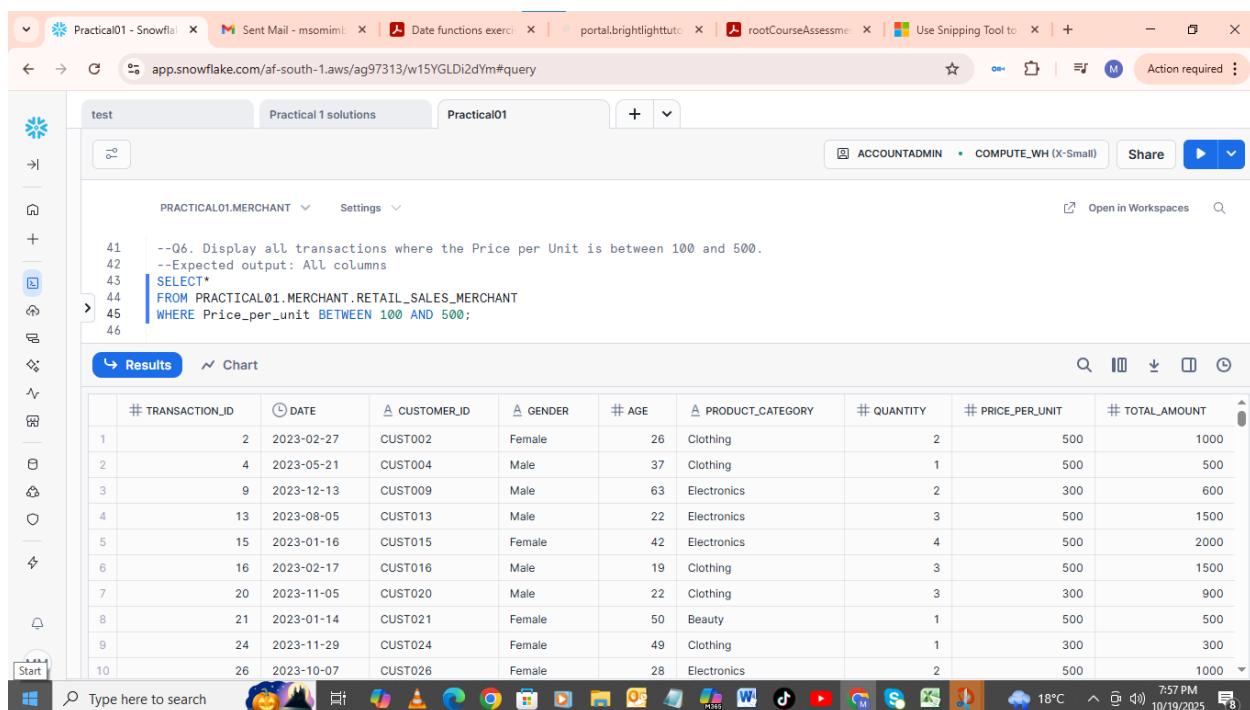
Question 5:



```
--Q5. Display all transactions where the Age is greater than 40.  
--Expected output: All columns  
SELECT*  
FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT  
WHERE age> 40;
```

# 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

Question 6:



```
--Q6. Display all transactions where the Price per Unit is between 100 and 500.  
--Expected output: All columns  
SELECT*  
FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT  
WHERE Price_per_unit BETWEEN 100 AND 500;
```

# 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

Question 7:

--Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.
--Expected output: All columns

```
49
50
51
52
53
54
```

SELECT*
FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
WHERE Product_category IN ('Beauty', 'Electronics');

# 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

Question 8:

--Q8. Display all transactions where the Product Category is not 'Clothing'.
--Expected output: All columns

```
57
58
59
60
61
62
63
64
```

SELECT*
FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
WHERE Product_category != 'Clothing';

# 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

Question 9:

Practical01 - Snowflake | Sent Mail - msomimi | Date functions exercise | portal.brightlighttuto | rootCourseAssessment | Use Snipping Tool to | + | - | X

<app.snowflake.com/af-south-1.aws/ag97313/w15YGLDi2dYm#query>

test Practical 1 solutions Practical01 +

PRACTICAL01.MERCHANT Settings

```

65 --Q9. Display all transactions where the Quantity is greater than or equal to 3.
66 --Expected output: All columns
67
68 SELECT*
69   FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
70   WHERE Quantity >=3;
    
```

Results Chart

# 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	2023-02-22	CUST008	Male	30	Electronics	4	25	100
3	2023-10-07	CUST010	Female	52	Clothing	4	50	200
4	2023-10-30	CUST012	Male	35	Beauty	3	25	75
5	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
6	2023-01-17	CUST014	Male	64	Clothing	4	30	120
7	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
8	2023-02-17	CUST016	Male	19	Clothing	3	500	1500
9	2023-04-22	CUST017	Female	27	Clothing	4	25	100
10	2023-11-05	CUST020	Male	22	Clothing	3	300	900

Question 10:

Practical01 - Snowflake | Sent Mail - msomimi | Date functions exercise | portal.brightlighttuto | rootCourseAssessment | Use Snipping Tool to | + | - | X

<app.snowflake.com/af-south-1.aws/ag97313/w15YGLDi2dYm#query>

test Practical 1 solutions Practical01 +

PRACTICAL01.MERCHANT Settings

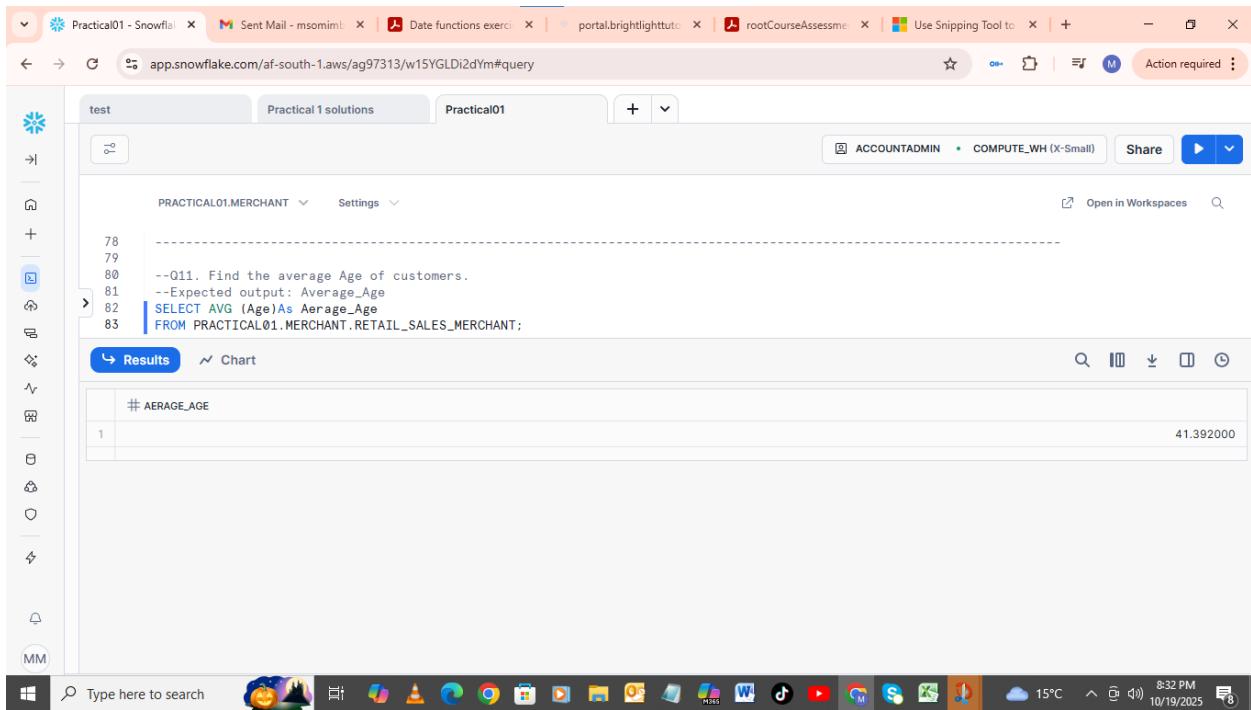
```

71 -----
72
73 --Q10. Count the total number of transactions.
74 --Expected output: Total_Transactions
75
76   SELECT COUNT(*) As Total_Transactions
      FROM practical01.merchant.retail_sales_merchant;
    
```

Results Chart

TOTAL_TRANSACTIONS	
1	1000

Question 11:



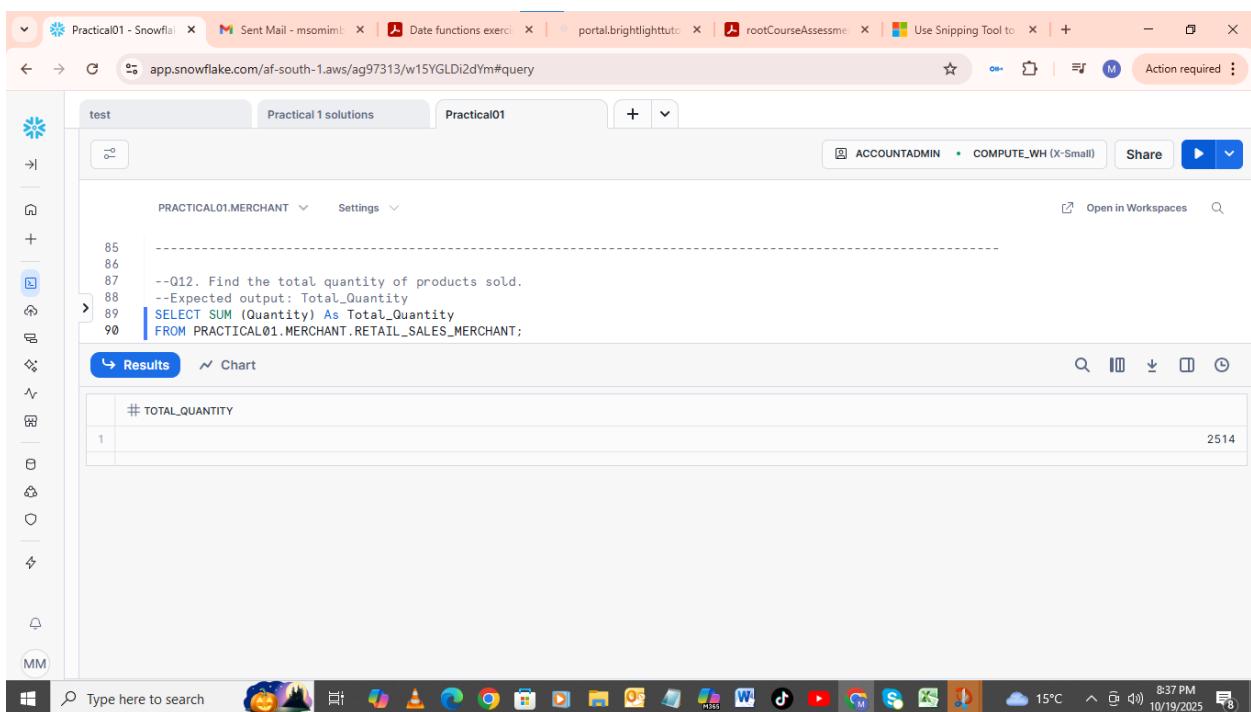
```
78
79
80    --Q11. Find the average Age of customers.
81    --Expected output: Average_Age
82    SELECT AVG (Age)As Aeration_Age
83    FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
```

The screenshot shows a Windows desktop with a Snowflake query editor window open. The query is as follows:

```
78
79
80    --Q11. Find the average Age of customers.
81    --Expected output: Average_Age
82    SELECT AVG (Age)As Aeration_Age
83    FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
```

The results pane shows a single row with the value 41.392000.

Question 12:



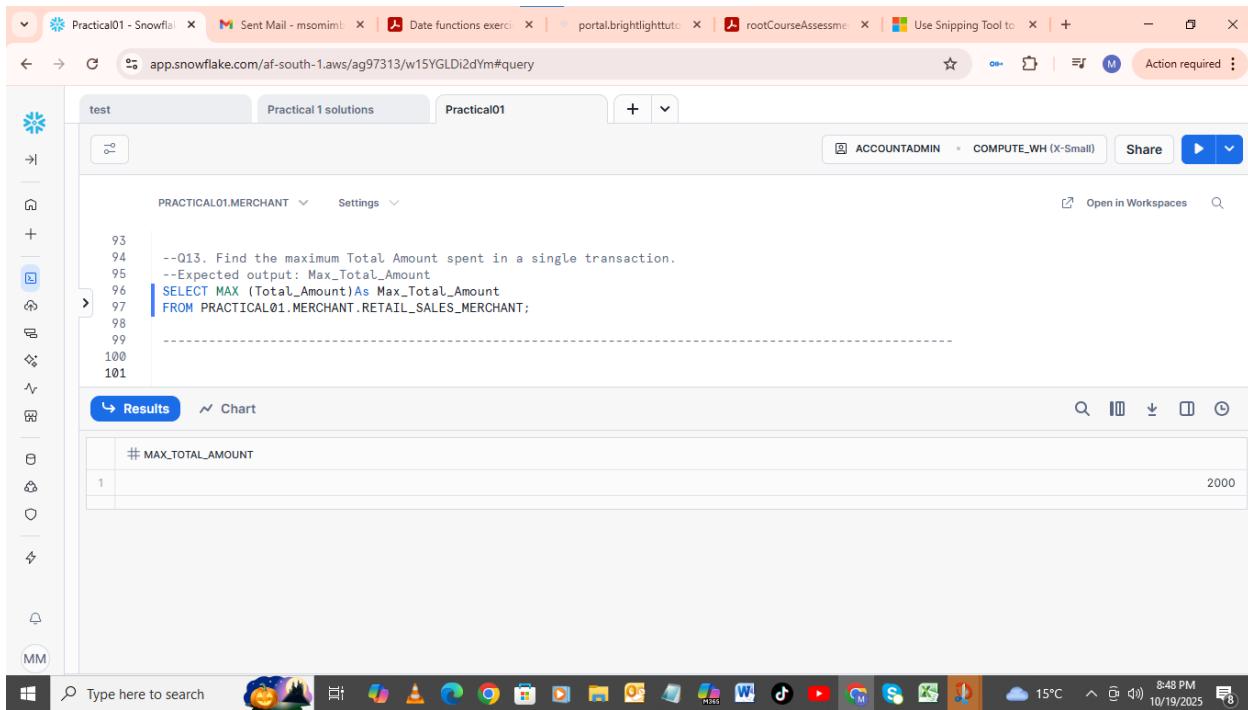
```
85
86
87    --Q12. Find the total quantity of products sold.
88    --Expected output: Total_Quantity
89    SELECT SUM (Quantity) As Total_Quantity
90    FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
```

The screenshot shows a Windows desktop with a Snowflake query editor window open. The query is as follows:

```
85
86
87    --Q12. Find the total quantity of products sold.
88    --Expected output: Total_Quantity
89    SELECT SUM (Quantity) As Total_Quantity
90    FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
```

The results pane shows a single row with the value 2514.

Question 13:



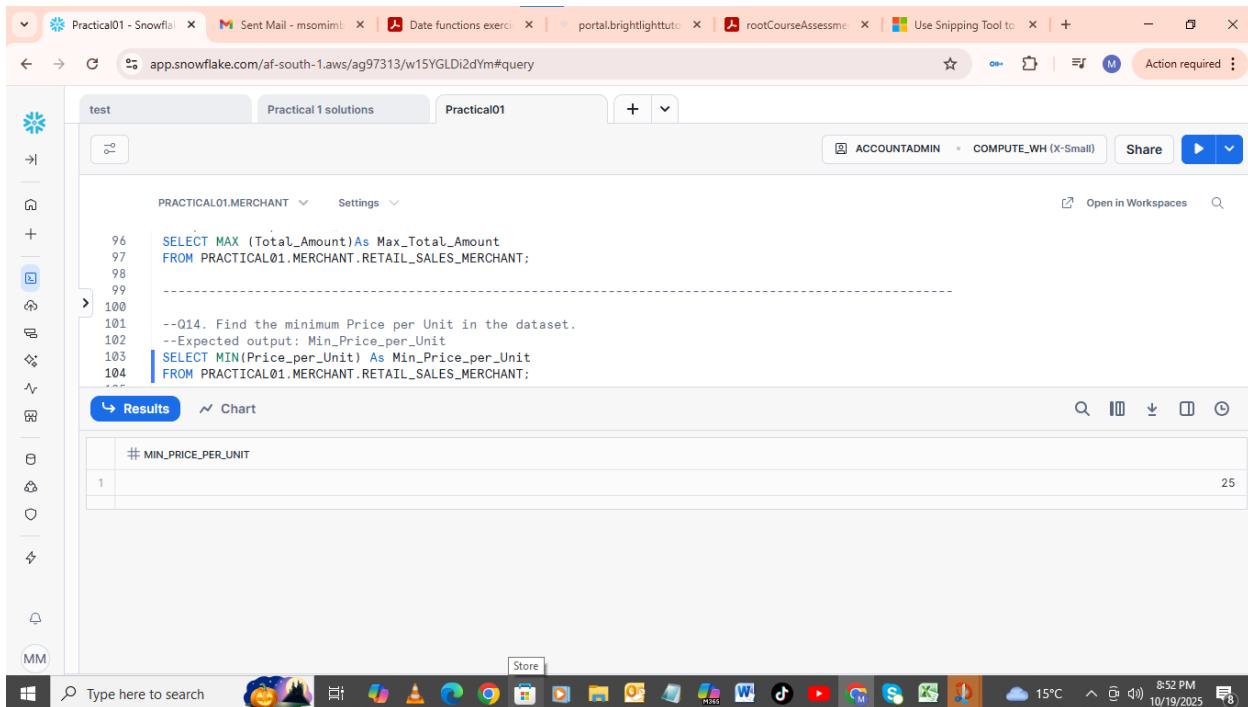
The screenshot shows a Windows desktop with a Snowflake query editor window open. The window title is "Practical01 - Snowflake". The query being run is:

```
93
94 --Q13. Find the maximum Total Amount spent in a single transaction.
95 --Expected output: Max_Total_Amount
96 SELECT MAX (Total_Amount)As Max_Total_Amount
97 FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
98
99
100
101
```

The results pane shows a single row with the value 2000.

MAX_TOTAL_AMOUNT
1 2000

Question 14:



The screenshot shows a Windows desktop with a Snowflake query editor window open. The window title is "Practical01 - Snowflake". The query being run is:

```
96
97 SELECT MAX (Total_Amount)As Max_Total_Amount
98
99
100
101 --Q14. Find the minimum Price per Unit in the dataset.
102 --Expected output: Min_Price_per_Unit
103 SELECT MIN(Price_per_Unit) As Min_Price_per_Unit
104 FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
105
```

The results pane shows a single row with the value 25.

MIN_PRICE_PER_UNIT
1 25

Question 15:

The screenshot shows a Windows desktop with a Snowflake query editor window open. The window title is "Practical01 - Snowflake". The query being run is:

```
105
106
107
108 --015. Find the number of transactions per Product Category.
109 --Expected output: Product Category, Transaction_Count
110
111
112
113
```

SELECT Product_Category,
COUNT (*) As Transaction_Count
FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
GROUP BY Product_Category;

The results table shows the following data:

PRODUCT_CATEGORY	TRANSACTION_COUNNT
Clothing	351
Beauty	307
Electronics	342

Question 16:

The screenshot shows a Windows desktop with a Snowflake query editor window open. The window title is "Practical01 - Snowflake". The query being run is:

```
114
115
116
117 --016. Find the total revenue (Total_Amount) per gender.
118 --Expected output: Gender, Total_Revenue
119
120
121
122
```

SELECT Gender,
SUM (Total_Amount) As Total_Revenue
FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
GROUP BY Gender;

The results table shows the following data:

GENDER	TOTAL_REVENUE
Male	223160
Female	232840

Question 17:

The screenshot shows a Windows desktop with a Snowflake query editor window open. The window title is "Practical01 - Snowflake". The query being run is:

```
123
124
125
126 --Q17. Find the average Price per Unit per product category.
127 --Expected output: Product Category, Average_Price
128 SELECT Product_Category,
129     AVG (Price_per_Unit) AS Average_Price
130 FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
131 GROUP BY Product_Category;
```

The results table shows the average price for three categories:

PRODUCT_CATEGORY	AVERAGE_PRICE
Beauty	184.055375
Clothing	174.287749
Electronics	181.900585

Question 18:

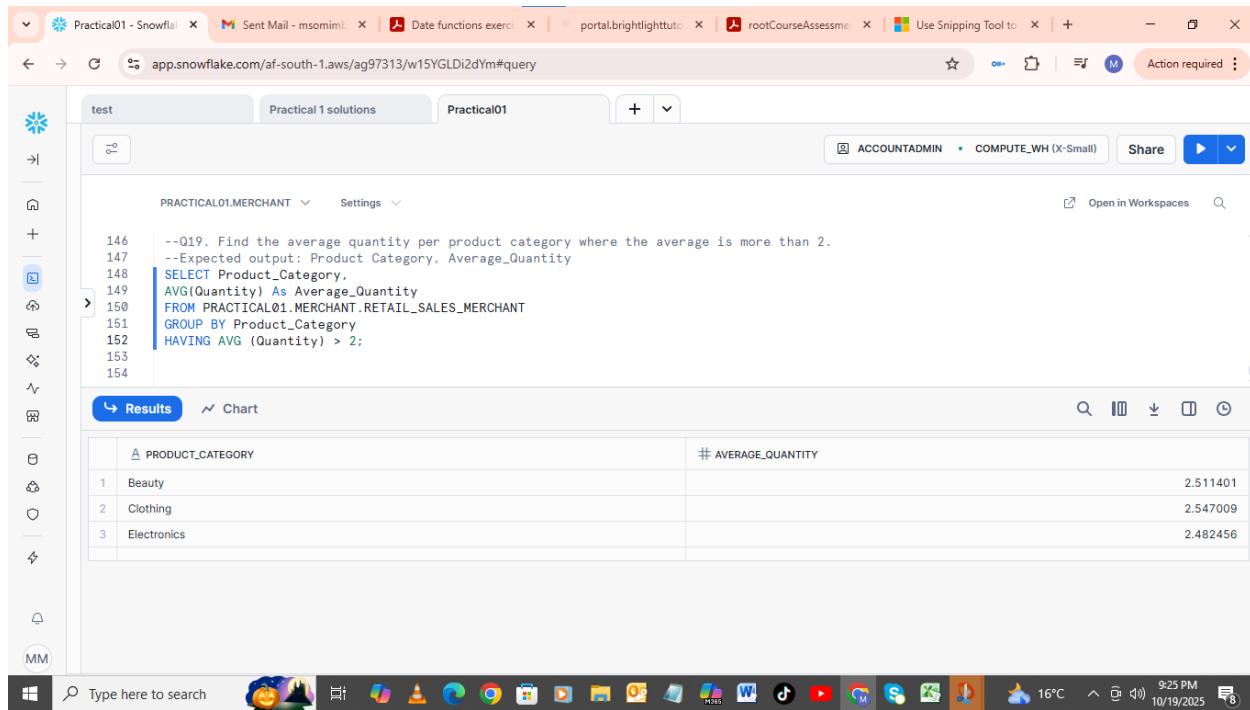
The screenshot shows a Windows desktop with a Snowflake query editor window open. The window title is "Practical01 - Snowflake". The query being run is:

```
134
135 --Q18. Find the total revenue per product category where total revenue is greater than 10,000.
136 --Expected output: Product Category, Total_Revenue
137
138 SELECT Product_Category,
139     SUM(Total_Amount) AS Total_Revenue
140 FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
141 GROUP BY Product_Category
142 HAVING SUM(Total_Amount) > 10000;
```

The results table shows the total revenue for three categories:

PRODUCT_CATEGORY	TOTAL_REVENUE
Beauty	143515
Clothing	155580
Electronics	156905

Question 19:

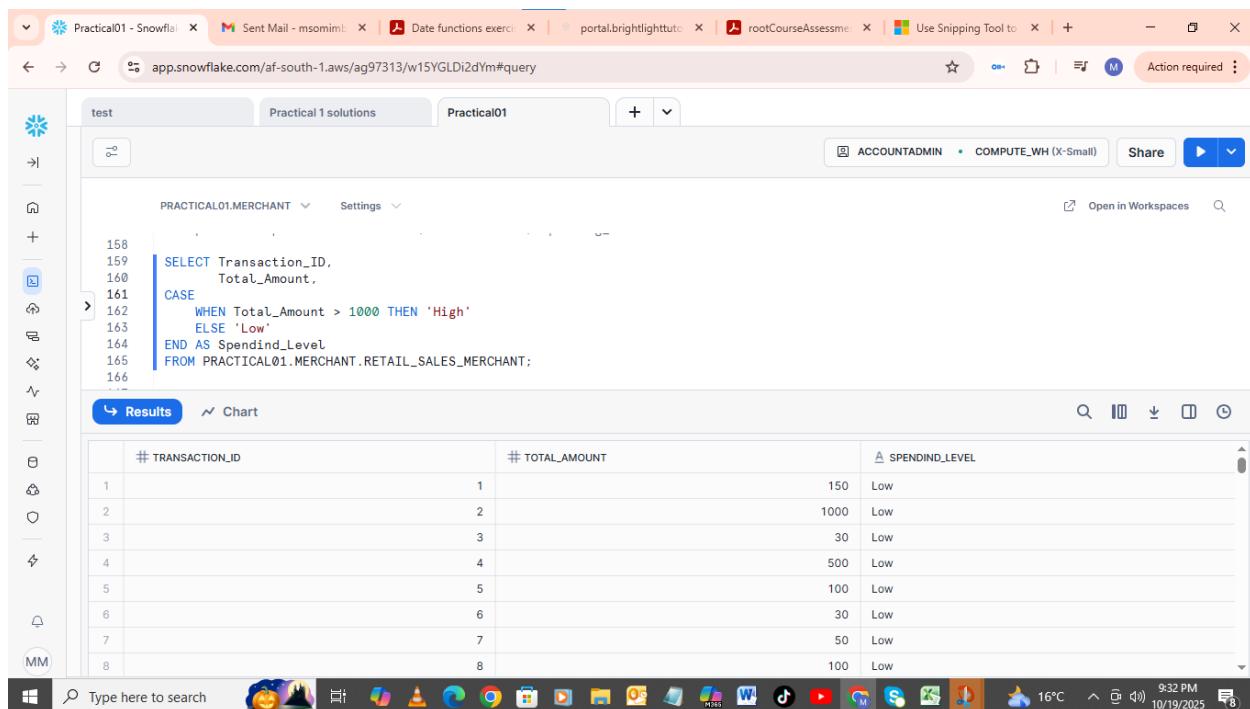


```
--Q19. Find the average quantity per product category where the average is more than 2.
--Expected output: Product Category, Average_Quantity
SELECT Product_Category,
AVG(Quantity) As Average_Quantity
FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT
GROUP BY Product_Category
HAVING AVG (Quantity) > 2;

```

PRODUCT_CATEGORY	AVERAGE_QUANTITY
1 Beauty	2.511401
2 Clothing	2.547009
3 Electronics	2.482456

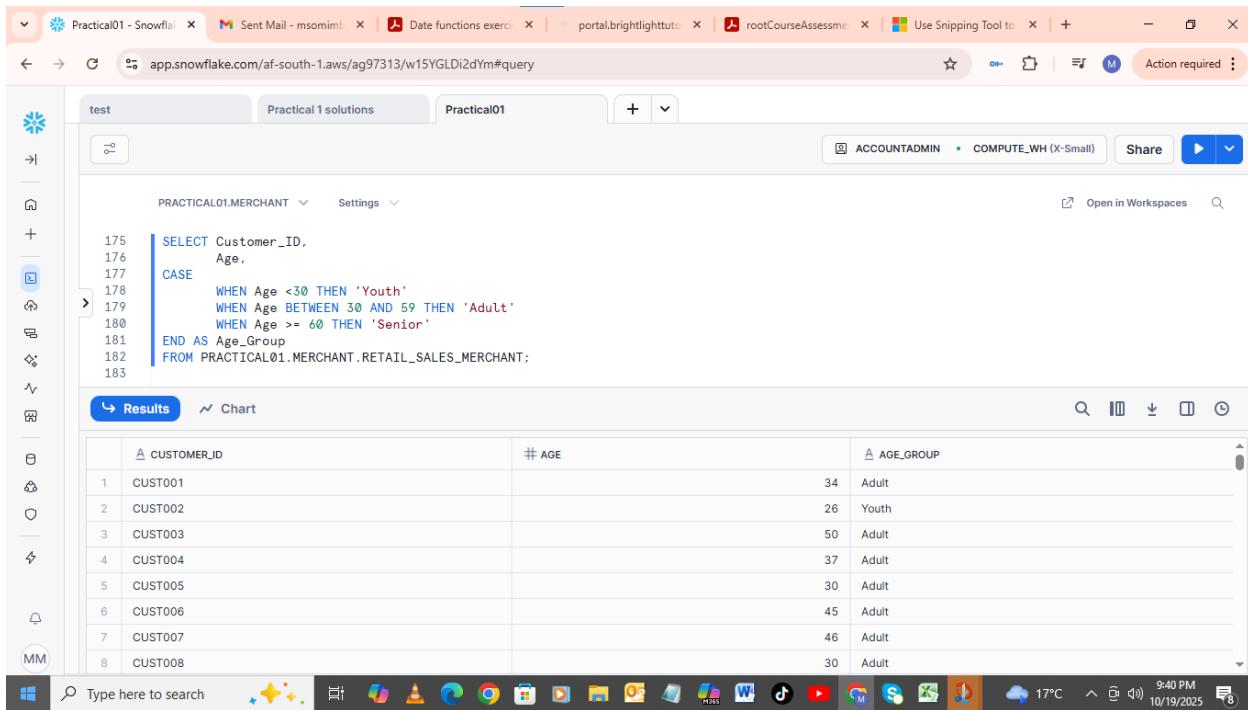
Question 20:



```
158
159 | SELECT Transaction_ID,
160 |     Total_Amount,
161 |     CASE
162 |         WHEN Total_Amount > 1000 THEN 'High'
163 |         ELSE 'Low'
164 |     END AS Spendind_Level
165 | FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
166
```

TRANSACTION_ID	TOTAL_AMOUNT	SPENDIND_LEVEL
1	150	Low
2	1000	Low
3	30	Low
4	500	Low
5	100	Low
6	30	Low
7	50	Low
8	100	Low

Question 21:



The screenshot shows a Windows desktop environment with a Snowflake query editor window open. The window title is "test" and the tab is "Practical01". The query editor contains the following SQL code:

```
175  SELECT Customer_ID,
176      Age,
177      CASE
178          WHEN Age < 30 THEN 'Youth'
179          WHEN Age BETWEEN 30 AND 59 THEN 'Adult'
180          WHEN Age >= 60 THEN 'Senior'
181      END AS Age_Group
182  FROM PRACTICAL01.MERCHANT.RETAIL_SALES_MERCHANT;
```

The results pane shows the following data:

	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
8	CUST008	30	Adult