

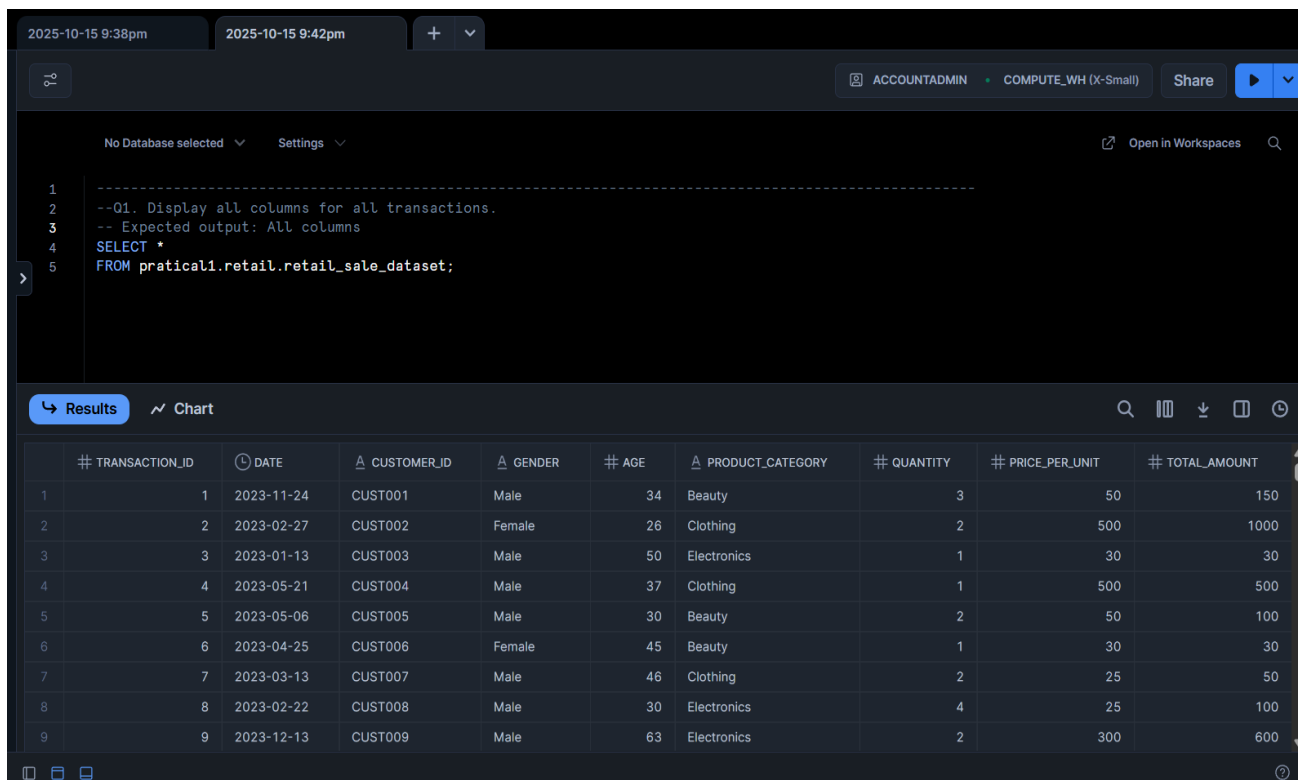
## Sindiswa Jachin Shongwe

### Practical 1: BrightLight Data Analytics Coding Practical Basic SQL Syntax

#### Question 1

-- Q1. Display all columns for all transactions.

-- Expected output: All columns



The screenshot shows a SQL IDE interface. The top bar displays the time '2025-10-15 9:38pm' and '2025-10-15 9:42pm'. The main area contains a SQL query:

```
1 -----  
2 --Q1. Display all columns for all transactions.  
3 -- Expected output: All columns  
4 SELECT *  
5 FROM pratical1.retail.retail_sale_dataset;
```

Below the query editor, the 'Results' tab is active, showing a table with 10 columns: # TRANSACTION\_ID, DATE, CUSTOMER\_ID, GENDER, AGE, PRODUCT\_CATEGORY, QUANTITY, PRICE\_PER\_UNIT, and TOTAL\_AMOUNT. The table contains 9 rows of data.

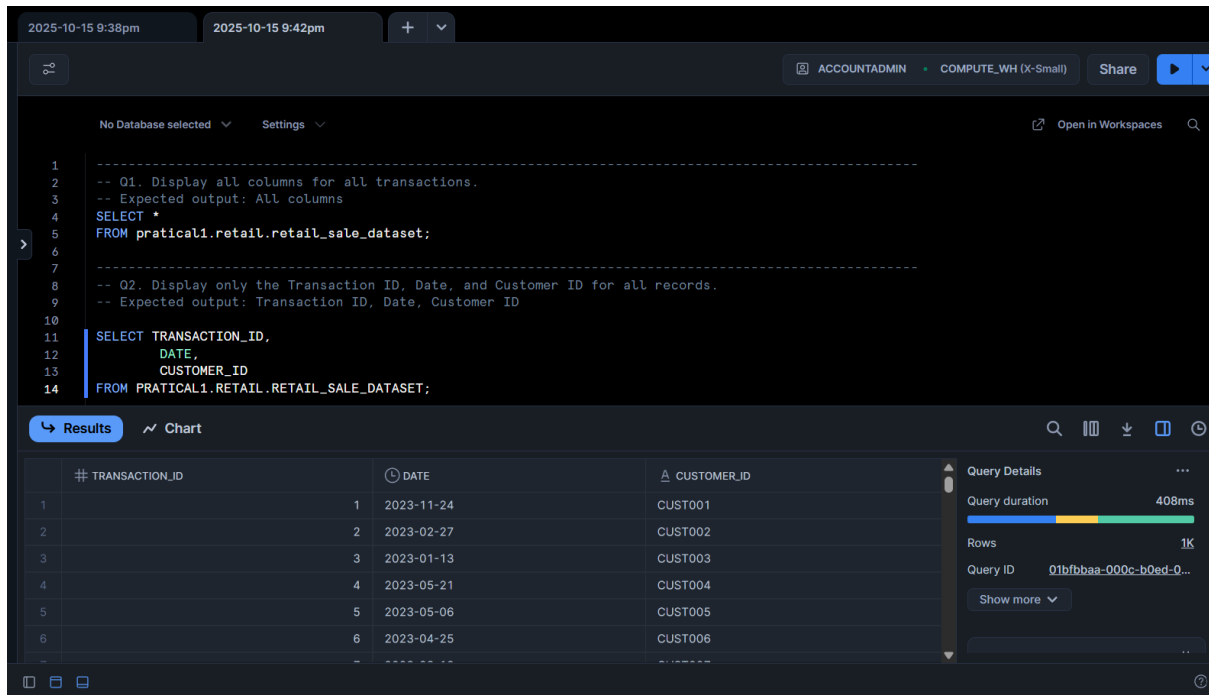
	# TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
5	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

## Question 2

---

-- Q2. Display only the Transaction ID, Date, and Customer ID for all records.

-- Expected output: Transaction ID, Date, Customer ID



The screenshot shows a SQL query editor interface. The top bar indicates the user is 'ACCOUNTADMIN' and the database is 'COMPUTE\_WH (X-Small)'. The editor contains two queries. Query 1 is commented out. Query 2 is active and shows the expected output: Transaction ID, Date, and Customer ID for all records. The results table shows 6 rows of data.

```
1  -----
2  -- Q1. Display all columns for all transactions.
3  -- Expected output: All columns
4  SELECT *
5  FROM practical1.retail.retail_sale_dataset;
6  -----
7
8  -- Q2. Display only the Transaction ID, Date, and Customer ID for all records.
9  -- Expected output: Transaction ID, Date, Customer ID
10
11 SELECT TRANSACTION_ID,
12        DATE,
13        CUSTOMER_ID
14 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
```

#	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

Query Details

- Query duration: 408ms
- Rows: 1K
- Query ID: 01bf7baa-000c-b0ed-0...
- Show more

### Question 3

---

-- Q3. Display all the distinct product categories in the dataset.

-- Expected output: Product Category

The screenshot shows a SQL IDE interface. The top bar displays the date and time '2025-10-15 9:38pm' and '2025-10-15 9:42pm', along with a '+' icon and a dropdown arrow. The main area contains a SQL query with line numbers 7 through 22. The query is as follows:

```
7 -----  
8 -- Q2. Display only the Transaction ID, Date, and Customer ID for all records.  
9 -- Expected output: Transaction ID, Date, Customer ID  
10  
11 SELECT TRANSACTION_ID,  
12        DATE,  
13        CUSTOMER_ID  
14 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;  
15  
16 -----  
17 -- Q3. Display all the distinct product categories in the dataset.  
18 -- Expected output: Product Category  
19  
20 SELECT DISTINCT product_category  
21 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;  
22 |
```

The bottom panel shows the 'Results' tab with a table of product categories. The table has a header row 'PRODUCT\_CATEGORY' and three data rows: 'Clothing', 'Beauty', and 'Electronics'. To the right of the table, the 'Query Details' panel shows 'Query duration' as 66ms, 'Rows' as 3, and 'Query ID' as 01bfbbb0-000c-b0ed-0... with a 'Show more' button.

PRODUCT_CATEGORY
1 Clothing
2 Beauty
3 Electronics

Query Details

- Query duration: 66ms
- Rows: 3
- Query ID: 01bfbbb0-000c-b0ed-0...
- Show more

-- Q4. Display all the distinct gender values in the dataset.

-- Expected output: Gender

The screenshot shows a SQL IDE interface with a dark theme. The top bar displays the date and time '2025-10-15 9:38pm' and '2025-10-15 9:42pm', along with user 'ACCOUNTADMIN' and database 'COMPUTE\_WH (X-Small)'. The main editor area contains SQL code for Q4. The results pane at the bottom shows a table with two rows: 'Male' and 'Female'.

```
12      DATE,  
13      CUSTOMER_ID  
14 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;  
15  
16 -----  
17 -- Q3. Display all the distinct product categories in the dataset.  
18 -- Expected output: Product Category  
19  
20 SELECT DISTINCT product_category  
21 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;  
22  
23 -----  
24 -- Q4. Display all the distinct gender values in the dataset.  
25 -- Expected output: Gender  
26 SELECT DISTINCT GENDER  
27 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
```

	GENDER
1	Male
2	Female

Query Details: Query duration 69ms, Rows 2, Query ID 01bfbbb8-000c-b0ed-0...

-- Q5. Display all transactions where the Age is greater than 40.

-- Expected output: All columns

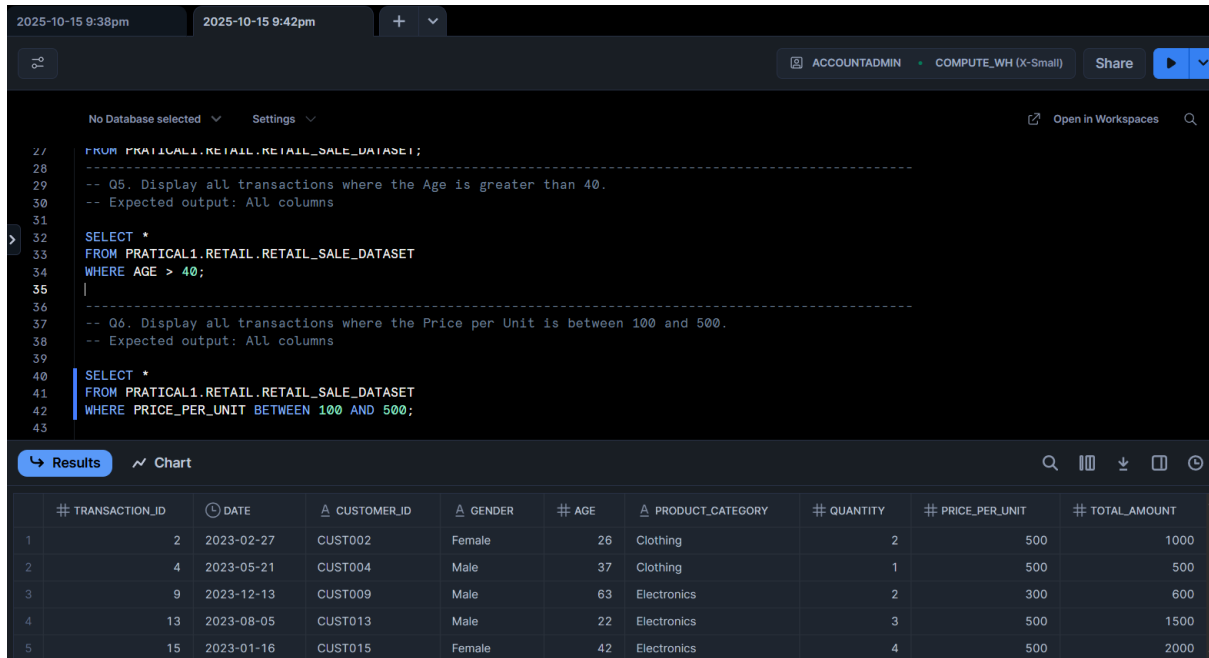
The screenshot shows the same SQL IDE interface. The main editor area contains SQL code for Q5. The results pane at the bottom shows a table with 5 rows of transaction data.

```
19  
20 SELECT DISTINCT product_category  
21 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;  
22  
23 -----  
24 -- Q4. Display all the distinct gender values in the dataset.  
25 -- Expected output: Gender  
26 SELECT DISTINCT GENDER  
27 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;  
28  
29 -- Q5. Display all transactions where the Age is greater than 40.  
30 -- Expected output: All columns  
31  
32 SELECT *  
33 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET  
34 WHERE AGE > 40;  
35
```

	# TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	# AGE	PRODUCT_CATEGORY	# QUANTITY	# PRICE_PER_UNIT	# TOTAL_AMOUNT
1	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
2	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
3	7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
4	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
5	10	2023-10-07	CUST010	Female	52	Clothing	4	50	200

-- Q6. Display all transactions where the Price per Unit is between 100 and 500.

-- Expected output: All columns



The screenshot shows a SQL IDE interface. The top bar displays the time '2025-10-15 9:38pm' and '2025-10-15 9:42pm'. The main editor area contains the following SQL code:

```
27 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;  
28  
29 -- Q5. Display all transactions where the Age is greater than 40.  
30 -- Expected output: All columns  
31  
32 SELECT *  
33 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET  
34 WHERE AGE > 40;  
35  
36  
37 -- Q6. Display all transactions where the Price per Unit is between 100 and 500.  
38 -- Expected output: All columns  
39  
40 SELECT *  
41 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET  
42 WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;  
43
```

The bottom panel shows the 'Results' tab with a table of 5 rows and 10 columns. The columns are: #, TRANSACTION\_ID, DATE, CUSTOMER\_ID, GENDER, AGE, PRODUCT\_CATEGORY, QUANTITY, PRICE\_PER\_UNIT, and TOTAL\_AMOUNT. The data is as follows:

#	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
3	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
4	13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
5	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000

-- Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.

--Expected output: All columns

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for '2025-10-15 9:38pm' and '2025-10-15 9:42pm'. The main editor area contains SQL code for two queries. Query Q6 is commented out, and Query Q7 is active. The results of Query Q7 are displayed in a table below the editor.

```
35 -----
36 -- Q6. Display all transactions where the Price per Unit is between 100 and 500.
37 -- Expected output: All columns
38
39
40 SELECT *
41 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
42 WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
43
44 -----
45 -- Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.
46 --Expected output: All columns
47
48 SELECT *
49 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
50 WHERE PRODUCT_CATEGORY='Beauty' OR PRODUCT_CATEGORY='Electronics';
51
```

The results table has the following columns: TRANSACTION\_ID, DATE, CUSTOMER\_ID, GENDER, AGE, PRODUCT\_CATEGORY, QUANTITY, PRICE\_PER\_UNIT, and TOTAL\_AMOUNT. It contains 5 rows of 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	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100

-- Q8. Display all transactions where the Product Category is not 'Clothing'.

-- Expected output: All columns

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for '2025-10-15 9:38pm' and '2025-10-15 9:42pm'. Below the tabs, there's a toolbar with 'ACCOUNTADMIN', 'COMPUTE\_WH (X-Small)', 'Share', and a play button. The main area contains a SQL query editor with the following text:

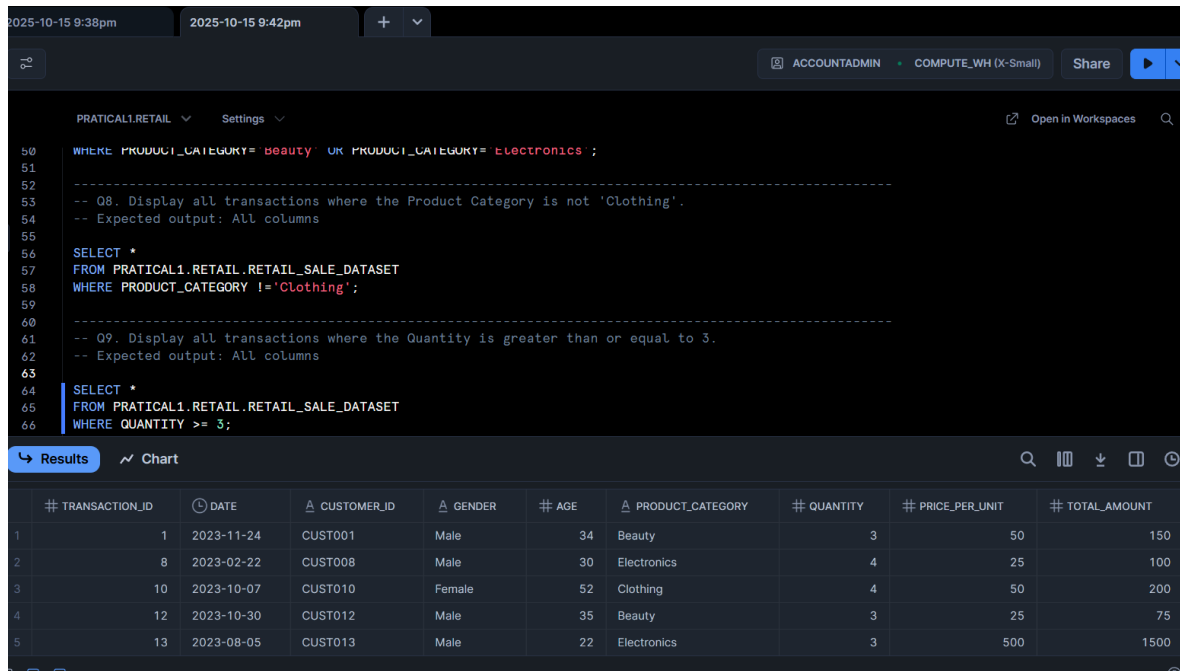
```
42 WHERE PRICE_PER_UNIT BETWEEN 100 AND 500;
43
44 -----
45 -- Q7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.
46 --Expected output: All columns
47
48 SELECT *
49 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
50 WHERE PRODUCT_CATEGORY='Beauty' OR PRODUCT_CATEGORY='Electronics';
51
52 -----
53 -- Q8. Display all transactions where the Product Category is not 'Clothing'.
54 -- Expected output: All columns
55
56 SELECT *
57 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
58 WHERE PRODUCT_CATEGORY !='Clothing';]
```

Below the query editor, there's a 'Results' tab selected, showing a table with 10 columns: TRANSACTION\_ID, DATE, CUSTOMER\_ID, GENDER, AGE, PRODUCT\_CATEGORY, QUANTITY, PRICE\_PER\_UNIT, and TOTAL\_AMOUNT. The table contains 5 rows of 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	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100

-- Q9. Display all transactions where the Quantity is greater than or equal to 3.

-- Expected output: All columns



The screenshot shows a SQL IDE interface with a query editor and a results table. The query editor contains the following SQL code:

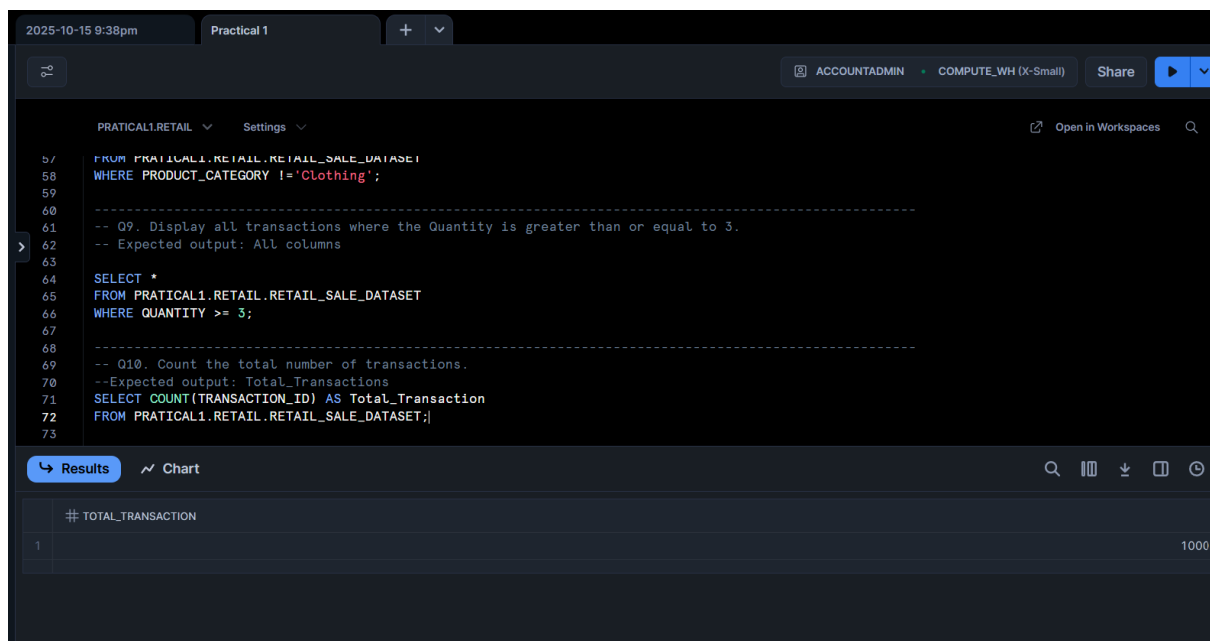
```
50 WHERE PRODUCT_CATEGORY = 'Beauty' OR PRODUCT_CATEGORY = 'Electronics';
51
52 -----
53 -- Q8. Display all transactions where the Product Category is not 'Clothing'.
54 -- Expected output: All columns
55
56 SELECT *
57 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
58 WHERE PRODUCT_CATEGORY != 'Clothing';
59
60 -----
61 -- Q9. Display all transactions where the Quantity is greater than or equal to 3.
62 -- Expected output: All columns
63
64 SELECT *
65 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
66 WHERE QUANTITY >= 3;
```

The results table displays the following 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	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
3	10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
4	12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
5	13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500

-- Q10. Count the total number of transactions.

--Expected output: Total\_Transactions



The screenshot shows a SQL IDE interface with a query editor and a results table. The query editor contains the following SQL code:

```
57 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
58 WHERE PRODUCT_CATEGORY != 'Clothing';
59
60 -----
61 -- Q9. Display all transactions where the Quantity is greater than or equal to 3.
62 -- Expected output: All columns
63
64 SELECT *
65 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
66 WHERE QUANTITY >= 3;
67
68 -----
69 -- Q10. Count the total number of transactions.
70 --Expected output: Total_Transactions
71 SELECT COUNT(TRANSACTION_ID) AS Total_Transaction
72 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
73
```

The results table displays the following data:

#	TOTAL_TRANSACTION
1	1000



---

-- Q11. Find the average Age of customers.

-- Expected output: Average\_Age

The screenshot shows a SQL IDE interface with a dark theme. The top bar displays the time '2025-10-15 9:38pm' and the workspace name 'Practical 1'. The main editor area contains SQL code for Q11. The code is as follows:

```
64 SELECT *
65 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
66 WHERE QUANTITY >= 3;
67
68 -----
69 -- Q10. Count the total number of transactions.
70 --Expected output: Total_Transactions
71 SELECT COUNT(TRANSACTION_ID) AS Total_Transaction
72 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
73
74 -----
75 -- Q11. Find the average Age of customers.
76 -- Expected output: Average_Age
77
78 SELECT AVG(AGE) AS Average_Age
79 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
```

Below the editor, the 'Results' tab is active, showing a single row of data:

AVERAGE_AGE
41.392000

---

-- Q12. Find the total quantity of products sold.

-- Expected output: Total\_Quantity

The screenshot shows the same SQL IDE interface. The main editor area contains SQL code for Q12. The code is as follows:

```
69 -- Q10. Count the total number of transactions.
70 --Expected output: Total_Transactions
71 SELECT COUNT(TRANSACTION_ID) AS Total_Transaction
72 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
73
74 -----
75 -- Q11. Find the average Age of customers.
76 -- Expected output: Average_Age
77
78 SELECT AVG(AGE) AS Average_Age
79 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
80
81 -----
82 -- Q12. Find the total quantity of products sold.
83 -- Expected output: Total_Quantity
84 SELECT COUNT(QUANTITY) AS Total_Quantity
85 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
```

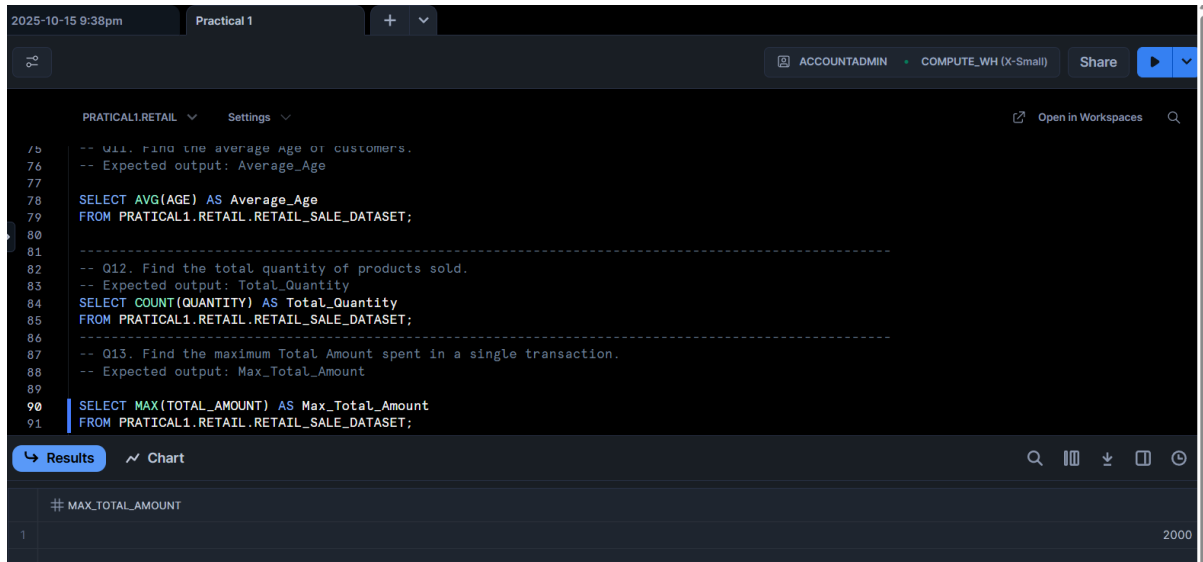
Below the editor, the 'Results' tab is active, showing a single row of data:

TOTAL_QUANTITY
1000

---

-- Q13. Find the maximum Total Amount spent in a single transaction.

-- Expected output: Max\_Total\_Amount



The screenshot shows a SQL IDE interface with a dark theme. The top bar displays the time '2025-10-15 9:38pm' and the workspace name 'Practical 1'. The main editor area contains SQL code for three queries. Query 13 is highlighted, showing a SELECT statement to find the maximum total amount. The results pane at the bottom shows a single row with the value 2000 for the column MAX\_TOTAL\_AMOUNT.

```
75 -- Q11. Find the average Age of customers.
76 -- Expected output: Average_Age
77
78 SELECT AVG(AGE) AS Average_Age
79 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
80
81 -----
82 -- Q12. Find the total quantity of products sold.
83 -- Expected output: Total_Quantity
84 SELECT COUNT(QUANTITY) AS Total_Quantity
85 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
86
87 -----
88 -- Q13. Find the maximum Total Amount spent in a single transaction.
89 -- Expected output: Max_Total_Amount
90 SELECT MAX(TOTAL_AMOUNT) AS Max_Total_Amount
91 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
```

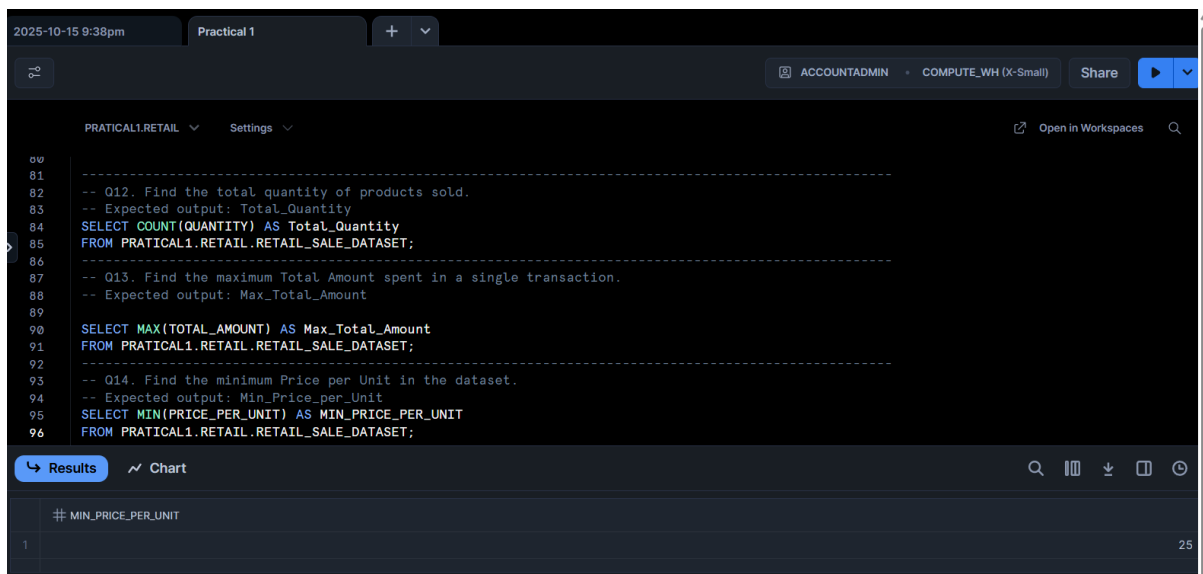
Results

MAX_TOTAL_AMOUNT
2000

---

-- Q14. Find the minimum Price per Unit in the dataset.

-- Expected output: Min\_Price\_per\_Unit



The screenshot shows the same SQL IDE interface. The main editor area now displays SQL code for three queries, with Query 14 highlighted. Query 14 is a SELECT statement to find the minimum price per unit. The results pane at the bottom shows a single row with the value 25 for the column MIN\_PRICE\_PER\_UNIT.

```
81 -----
82 -- Q12. Find the total quantity of products sold.
83 -- Expected output: Total_Quantity
84 SELECT COUNT(QUANTITY) AS Total_Quantity
85 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
86
87 -----
88 -- Q13. Find the maximum Total Amount spent in a single transaction.
89 -- Expected output: Max_Total_Amount
90 SELECT MAX(TOTAL_AMOUNT) AS Max_Total_Amount
91 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
92
93 -----
94 -- Q14. Find the minimum Price per Unit in the dataset.
95 -- Expected output: Min_Price_per_Unit
96 SELECT MIN(PRICE_PER_UNIT) AS MIN_PRICE_PER_UNIT
97 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
```

Results

MIN_PRICE_PER_UNIT
25

-- Q15. Find the number of transactions per Product Category.

-- Expected output: Product Category, Transaction\_Count

The screenshot shows a SQL IDE interface with a dark theme. At the top, there's a tab labeled 'Practical 1' and a user profile 'ACCOUNTADMIN'. The main editor area contains SQL code for three queries. The third query, Q15, is highlighted and shows the expected output: Product Category, Transaction\_Count. Below the editor, the 'Results' tab is active, displaying a table with three rows: Beauty (307), Clothing (351), and Electronics (342).

```
87 -- Q13. Find the maximum total amount spent in a single transaction.
88 -- Expected output: Max_Total_Amount
89
90 SELECT MAX(TOTAL_AMOUNT) AS Max_Total_Amount
91 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
92
93 -- Q14. Find the minimum Price per Unit in the dataset.
94 -- Expected output: Min_Price_per_Unit
95 SELECT MIN(PRICE_PER_UNIT) AS MIN_PRICE_PER_UNIT
96 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
97
98
99 -- Q15. Find the number of transactions per Product Category.
100 -- Expected output: Product Category, Transaction_Count
101 SELECT PRODUCT_CATEGORY, COUNT(TRANSACTION_ID) AS Transaction_Count
102 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
103 GROUP BY PRODUCT_CATEGORY;
```

	PRODUCT_CATEGORY	TRANSACTION_COUNT
1	Beauty	307
2	Clothing	351
3	Electronics	342

-- Q16. Find the total revenue (Total Amount) per gender.

-- Expected output: Gender, Total\_Revenue

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
-- Q15. Find the number of transactions per Product Category.
-- Expected output: Product Category, Transaction_Count

SELECT PRODUCT_CATEGORY, COUNT(TRANSACTION_ID) AS Transaction_Count
FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
GROUP BY PRODUCT_CATEGORY;

-- Q16. Find the total revenue (Total Amount) per gender.
-- Expected output: Gender, Total_Revenue

SELECT GENDER, SUM(TOTAL_AMOUNT) AS Total_Revenue
FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
GROUP BY GENDER;
```

The results pane shows the output for the Q16 query:

	GENDER	# TOTAL_REVENUE
1	Male	223160
2	Female	232840

-- Q17. Find the average Price per Unit per product category.

-- Expected output: Product Category, Average\_Price

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
GROUP BY PRODUCT_CATEGORY;

-- Q16. Find the total revenue (Total Amount) per gender.
-- Expected output: Gender, Total_Revenue

SELECT GENDER, SUM(TOTAL_AMOUNT) AS Total_Revenue
FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
GROUP BY GENDER;

-- Q17. Find the average Price per Unit per product category.
-- Expected output: Product Category, Average_Price

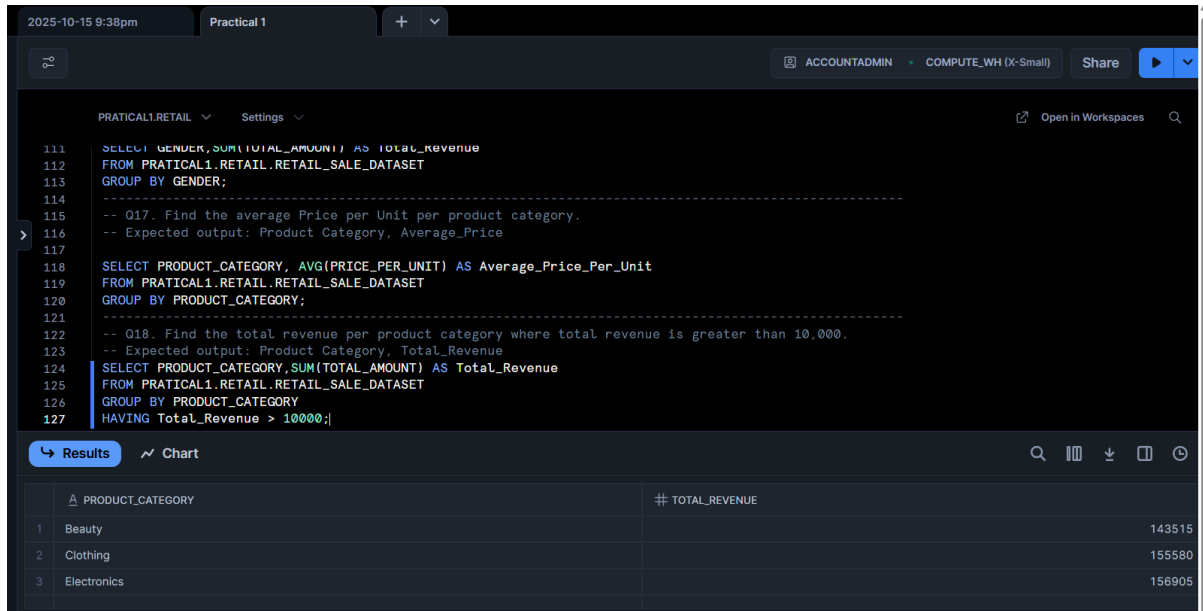
SELECT PRODUCT_CATEGORY, AVG(PRICE_PER_UNIT) AS Average_Price_Per_Unit
FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
GROUP BY PRODUCT_CATEGORY;
```

The results pane shows the output for the Q17 query:

	PRODUCT_CATEGORY	# AVERAGE_PRICE_PER_UNIT
1	Beauty	184.055375
2	Clothing	174.287749
3	Electronics	181.900585

-- Q18. Find the total revenue per product category where total revenue is greater than 10,000.

-- Expected output: Product Category, Total\_Revenue



The screenshot shows a SQL IDE interface with a dark theme. The top bar displays the date and time '2025-10-15 9:38pm' and the workspace name 'Practical 1'. The main editor area contains SQL code for Q18, which is highlighted. The code is as follows:

```
111 SELECT GENDER, SUM(TOTAL_AMOUNT) AS Total_Revenue
112 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
113 GROUP BY GENDER;
114
115 -- Q17. Find the average Price per Unit per product category.
116 -- Expected output: Product Category, Average_Price
117
118 SELECT PRODUCT_CATEGORY, AVG(PRICE_PER_UNIT) AS Average_Price_Per_Unit
119 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
120 GROUP BY PRODUCT_CATEGORY;
121
122 -- Q18. Find the total revenue per product category where total revenue is greater than 10,000.
123 -- Expected output: Product Category, Total_Revenue
124 SELECT PRODUCT_CATEGORY, SUM(TOTAL_AMOUNT) AS Total_Revenue
125 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
126 GROUP BY PRODUCT_CATEGORY
127 HAVING Total_Revenue > 10000;
```

Below the editor, the 'Results' tab is active, showing a table with the following data:

	PRODUCT_CATEGORY	TOTAL_REVENUE
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

-- Q19. Find the average quantity per product category where the average is more than 2.

-- Expected output: Product Category, Average\_Quantity

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
121 -----
122 -- Q18. Find the total revenue per product category where total revenue is greater than 10,000.
123 -- Expected output: Product Category, Total_Revenue
124
125 SELECT PRODUCT_CATEGORY, SUM(TOTAL_AMOUNT) AS Total_Revenue
126 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
127 GROUP BY PRODUCT_CATEGORY
128 HAVING Total_Revenue > 10000;
129 -----
130 -- Q19. Find the average quantity per product category where the average is more than 2.
131 -- Expected output: Product Category, Average_Quantity
132
133 SELECT PRODUCT_CATEGORY, AVG(QUANTITY) AS Average_Quantity
134 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
135 GROUP BY PRODUCT_CATEGORY
136 HAVING Average_Quantity > 2;
137 |
```

The results pane shows the output for Q19:

	PRODUCT_CATEGORY	AVERAGE_QUANTITY
1	Beauty	2.511401
2	Clothing	2.547009
3	Electronics	2.482456

-- Q20. Display a column called Spending\_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.

-- Expected output: Transaction ID, Total Amount, Spending\_Level

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
132
133 SELECT PRODUCT_CATEGORY, AVG(QUANTITY) AS Average_Quantity
134 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET
135 GROUP BY PRODUCT_CATEGORY
136 HAVING Average_Quantity > 2;
137 -----
138 -- Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.
139 -- Expected output: Transaction ID, Total Amount, Spending_Level
140
141 SELECT Transaction_ID,
142        TOTAL_AMOUNT,
143        CASE
144            WHEN TOTAL_AMOUNT > 1000 THEN 'High'
145            ELSE 'Low'
146        END AS Spending_Level
147 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
148
```

The results pane shows the output for Q20:

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

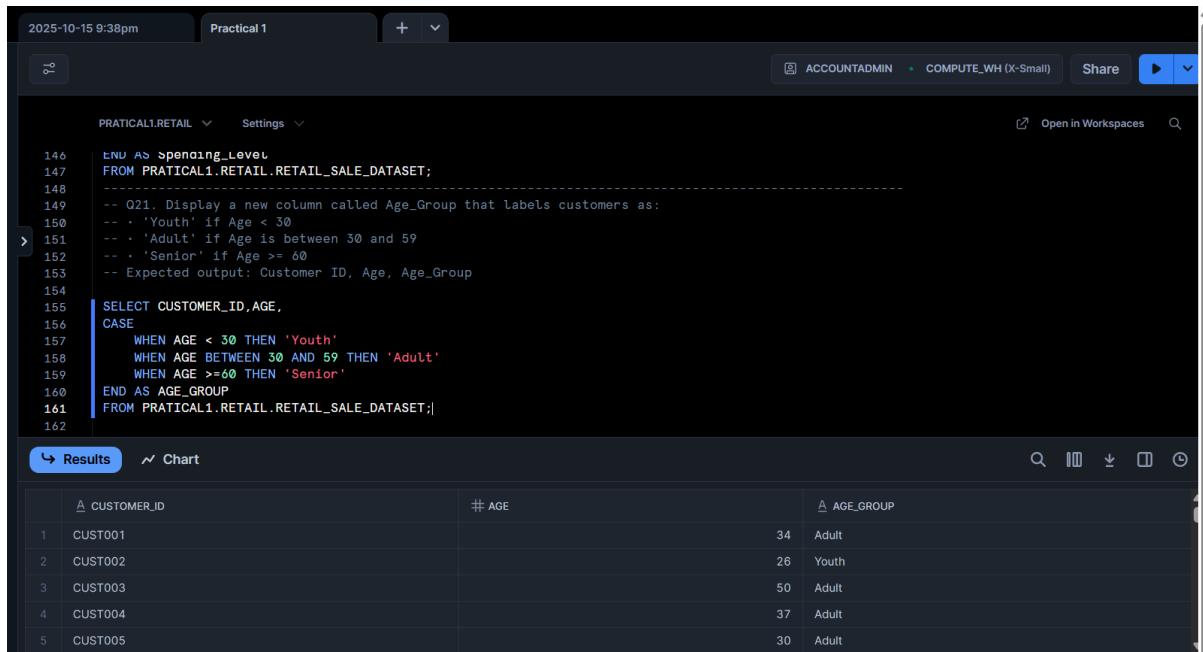
-- Q21. Display a new column called Age\_Group that labels customers as:

-- • 'Youth' if Age < 30

-- • 'Adult' if Age is between 30 and 59

-- • 'Senior' if Age >= 60

-- Expected output: Customer ID, Age, Age\_Group



The screenshot shows a SQL IDE interface with a dark theme. The top bar displays the time '2025-10-15 9:38pm' and the workspace name 'Practical 1'. The main editor area contains a SQL query with line numbers 146 to 162. The query is as follows:

```
146 END AS Spending_Level
147 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
148 -----
149 -- Q21. Display a new column called Age_Group that labels customers as:
150 -- • 'Youth' if Age < 30
151 -- • 'Adult' if Age is between 30 and 59
152 -- • 'Senior' if Age >= 60
153 -- Expected output: Customer ID, Age, Age_Group
154
155 SELECT CUSTOMER_ID,AGE,
156 CASE
157     WHEN AGE < 30 THEN 'Youth'
158     WHEN AGE BETWEEN 30 AND 59 THEN 'Adult'
159     WHEN AGE >=60 THEN 'Senior'
160 END AS AGE_GROUP
161 FROM PRATICAL1.RETAIL.RETAIL_SALE_DATASET;
162
```

Below the editor, the 'Results' tab is active, showing a table with 5 rows and 3 columns: CUSTOMER\_ID, AGE, and AGE\_GROUP. The data is as follows:

	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