

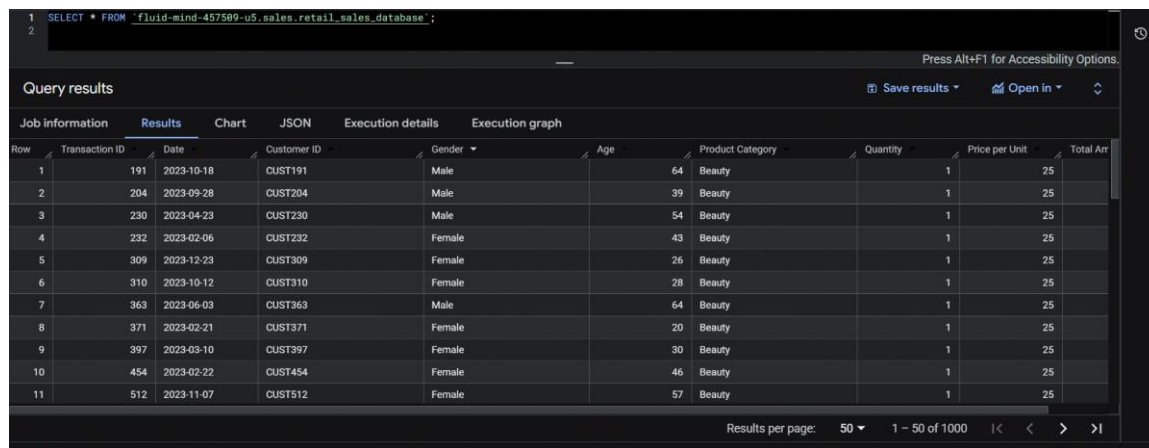
RIBA MAUBANE

BIG QUERY (PRACTICAL 1)

SELECT Statement

Q1. Display all columns for all transactions.

Expected output: All columns



```
1 SELECT * FROM 'fluid-mind-457589-u5.sales.retail_sales_database';
2
```

Query results

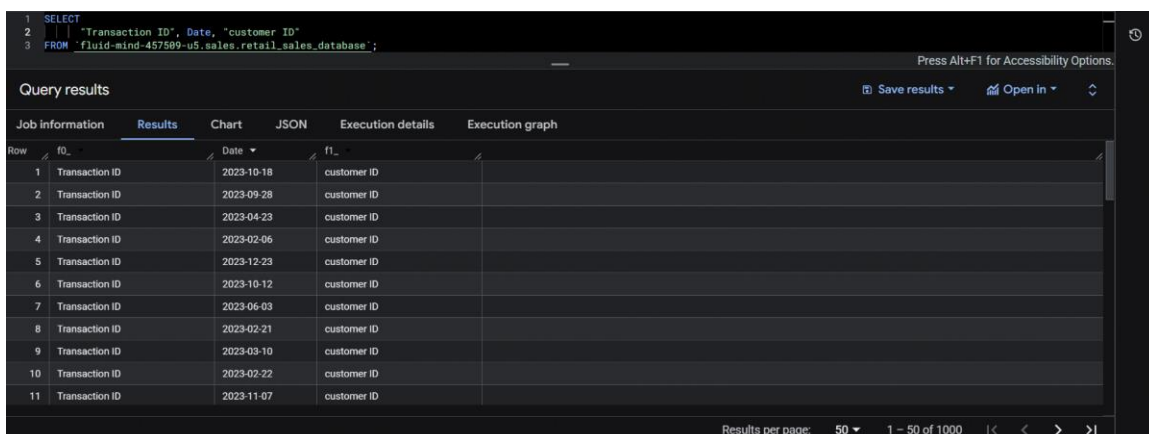
Job information Results Chart JSON Execution details Execution graph

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Arr
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25	
2	204	2023-09-28	CUST204	Male	39	Beauty	1	25	
3	230	2023-04-23	CUST230	Male	54	Beauty	1	25	
4	232	2023-02-06	CUST232	Female	43	Beauty	1	25	
5	309	2023-12-23	CUST309	Female	26	Beauty	1	25	
6	310	2023-10-12	CUST310	Female	28	Beauty	1	25	
7	363	2023-06-03	CUST363	Male	64	Beauty	1	25	
8	371	2023-02-21	CUST371	Female	20	Beauty	1	25	
9	397	2023-03-10	CUST397	Female	30	Beauty	1	25	
10	454	2023-02-22	CUST454	Female	46	Beauty	1	25	
11	512	2023-11-07	CUST512	Female	57	Beauty	1	25	

Results per page: 50 1 - 50 of 1000

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

Expected output: Transaction ID, Date, Customer ID



```
1 SELECT
2   "Transaction ID", "Date", "customer ID"
3 FROM 'fluid-mind-457589-u5.sales.retail_sales_database';
```

Query results

Job information Results Chart JSON Execution details Execution graph

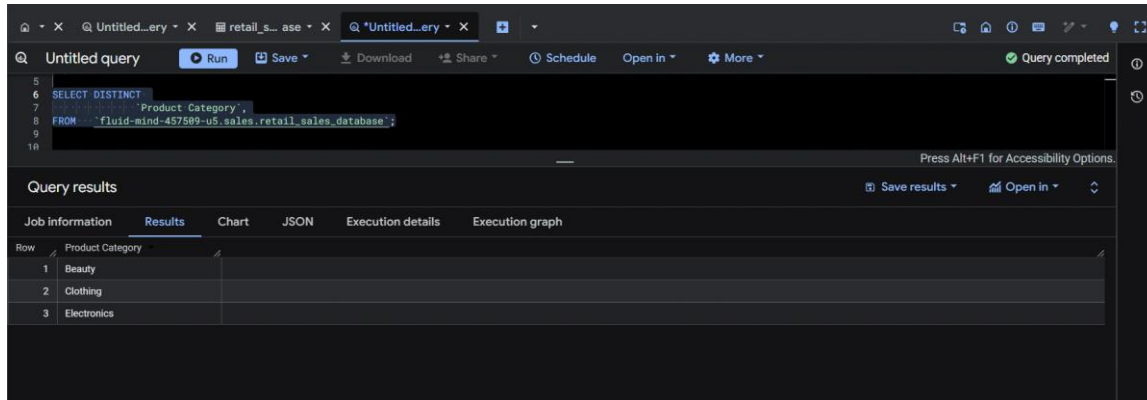
Row	f0_	Date	f1_
1	Transaction ID	2023-10-18	customer ID
2	Transaction ID	2023-09-28	customer ID
3	Transaction ID	2023-04-23	customer ID
4	Transaction ID	2023-02-06	customer ID
5	Transaction ID	2023-12-23	customer ID
6	Transaction ID	2023-10-12	customer ID
7	Transaction ID	2023-06-03	customer ID
8	Transaction ID	2023-02-21	customer ID
9	Transaction ID	2023-03-10	customer ID
10	Transaction ID	2023-02-22	customer ID
11	Transaction ID	2023-11-07	customer ID

Results per page: 50 1 - 50 of 1000

- SELECT DISTINCT Statement

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

Expected output: Product Category



```
5
6 SELECT DISTINCT
7   'Product Category',
8 FROM 'fluid-mind-457589-u5.sales.retail_sales_database';
9
10
```

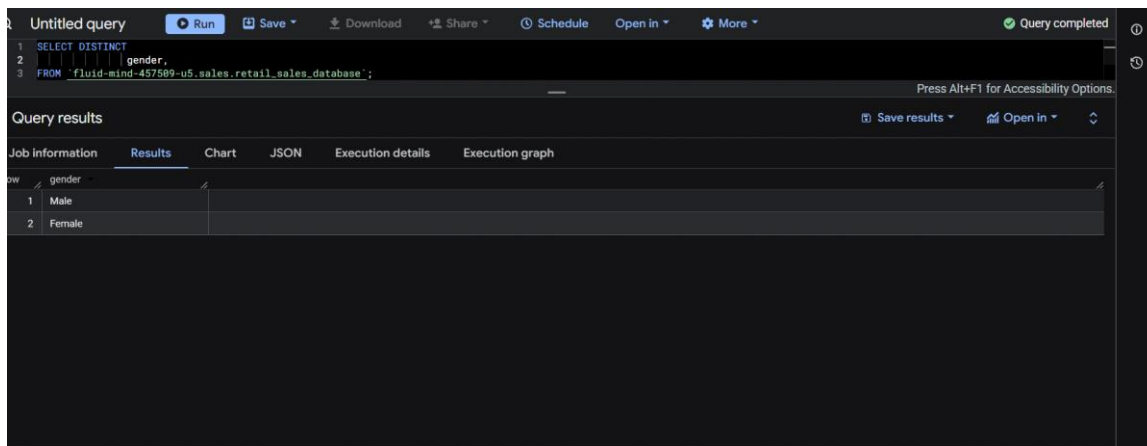
Query results

Job information Results Chart JSON Execution details Execution graph

Row	Product Category
1	Beauty
2	Clothing
3	Electronics

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

Expected output: Gender



```
1 SELECT DISTINCT
2   gender,
3 FROM 'fluid-mind-457589-u5.sales.retail_sales_database';
```

Query results

Job information Results Chart JSON Execution details Execution graph

Row	gender
1	Male
2	Female

- **WHERE Clause**

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

Expected output: All columns

1SELECT

2

3FROM "fluid-mind-457589-u5.sales.retail_sales_database"

4WHERE

5age > 40;

6

7

Press Alt+F1 for Accessibility Options.

Query results

Save results

Open in

Job informationResultsChartJSONExecution detailsExecution graph

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Arr
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25	
2	230	2023-04-23	CUST230	Male	54	Beauty	1	25	
3	232	2023-02-06	CUST232	Female	43	Beauty	1	25	
4	363	2023-06-03	CUST363	Male	64	Beauty	1	25	
5	454	2023-02-22	CUST454	Female	46	Beauty	1	25	
6	512	2023-11-07	CUST512	Female	57	Beauty	1	25	
7	791	2023-12-05	CUST791	Female	51	Beauty	1	25	
8	825	2023-08-26	CUST825	Female	46	Beauty	1	25	

Results per page:501 - 50 of 534

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

Expected output: All columns

Untitled...ery ×retail...se ×*Untitled...ery ×

Run

Save

Download

Share

Schedule

Open in

More

Query completed

```
1 SELECT *,
2 FROM "fluid-mind-457589-u5.sales.retail_sales_database"
3 WHERE
4 "Price per Unit" BETWEEN 100 AND 500;
5
```

Press Alt+F1 for Accessibility Options.

Query results

Save results

Open in

Job informationResultsChartJSONExecution detailsExecution graph

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Arr
1	52	2023-03-05	CUST052	Female	36	Beauty	1	300	
2	79	2023-04-18	CUST079	Male	34	Beauty	1	300	
3	174	2023-04-12	CUST174	Female	39	Beauty	1	300	
4	240	2023-02-06	CUST240	Female	23	Beauty	1	300	
5	358	2023-05-16	CUST358	Female	32	Beauty	1	300	
6	378	2023-06-28	CUST378	Male	50	Beauty	1	300	
7	555	2023-10-19	CUST555	Male	25	Beauty	1	300	
8	794	2023-09-17	CUST794	Female	60	Beauty	1	300	
9	905	2023-04-02	CUST905	Male	58	Beauty	1	300	
10	947	2023-03-02	CUST947	Male	50	Beauty	1	300	

Results per page: 501 - 50 of 396<>>>

Job history

Refresh

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

Expected output: All columns

Untitled query

Run

Save

Download

Share

Schedule

Open in

More

Query completed

```
6 SELECT
7 *
8 FROM 'fluid-mind-457589-u5.sales.retail_sales_database'
9 WHERE
10   'Product Category' = "Beauty"
11 OR
12   'Product Category' = "Electronics";
13
14
15
```

Press Alt+F1 for Accessibility Options.

Query results

Save results

Open in

Job information

Results

Chart

JSON

Execution details

Execution graph

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Arr
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25	
2	204	2023-09-28	CUST204	Male	39	Beauty	1	25	
3	230	2023-04-23	CUST230	Male	54	Beauty	1	25	
4	232	2023-02-06	CUST232	Female	43	Beauty	1	25	
5	309	2023-12-23	CUST309	Female	26	Beauty	1	25	
6	310	2023-10-12	CUST310	Female	28	Beauty	1	25	
7	363	2023-06-03	CUST363	Male	64	Beauty	1	25	

Results per page:

50

1 - 50 of 649

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

Expected output: All columns

16

17 SELECT *

18 FROM

19 "fluid-mind-457589-u5.sales.retail_sales_database"

20

21 WHERE

22 "Product Category" <> 'Clothing';

Press Alt+F1 for Accessibility Options.

Query results

Save results

Open in

Job information	Results	Chart	JSON	Execution details	Execution graph				
Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Arr
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25	
2	204	2023-09-28	CUST204	Male	39	Beauty	1	25	
3	230	2023-04-23	CUST230	Male	54	Beauty	1	25	
4	232	2023-02-06	CUST232	Female	43	Beauty	1	25	
5	309	2023-12-23	CUST309	Female	26	Beauty	1	25	
6	310	2023-10-12	CUST310	Female	28	Beauty	1	25	
7	363	2023-06-03	CUST363	Male	64	Beauty	1	25	
8	371	2023-02-21	CUST371	Female	20	Beauty	1	25	

Results per page: 50 1 - 50 of 1000 < < > >

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

Expected output: All columns

23 SELECT *,

24 FROM "fluid-mind-457589-u5.sales.retail_sales_database"

25 WHERE

26 "Quantity" > "3"

27 OR

28 "Quantity" = "3";

29

30

Press Alt+F1 for Accessibility Options.

Query results

Save results

Open in

Job information

Results

Chart

JSON

Execution details

Execution graph

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Arr
1	191	2023-10-18	CUST191	Male	64	Beauty	1	25	
2	204	2023-09-28	CUST204	Male	39	Beauty	1	25	
3	230	2023-04-23	CUST230	Male	54	Beauty	1	25	
4	232	2023-02-06	CUST232	Female	43	Beauty	1	25	
5	309	2023-12-23	CUST309	Female	26	Beauty	1	25	
6	310	2023-10-12	CUST310	Female	28	Beauty	1	25	
7	363	2023-06-03	CUST363	Male	64	Beauty	1	25	
8	371	2023-02-21	CUST371	Female	20	Beauty	1	25	

Results per page: 501 - 50 of 1000<<<>>>

- **Aggregate Functions**

Q10. Count the total number of transactions.

Expected output: Total_Transactions

```
29
30
31 SELECT COUNT ("Transaction") AS total_transaction,
32
33 FROM "fluid-mind-457509-u5.sales.retail_sales_database";
34
35
36
37
38
```

Query results Save results Open in

Job information Results Chart JSON Execution details Execution graph

Row	total_transaction
1	1000

Q11. Find the average Age of customers.

Expected output: Average_Age

```
35
36 SELECT avg(Age) AS Average_Age,
37 FROM "fluid-mind-457509-u5.sales.retail_sales_database";
38
39
40
```

Query results Save results Open in

Job information Results Chart JSON Execution details Execution graph

Row	Average_Age
1	41.39199999999999

Q12. Find the total quantity of products sold.

Expected output: Total_Quantity

```
38
39
40 SELECT SUM(Quantity) AS Total_Quantity,
41 FROM "fluid-mind-457509-u5.sales.retail_sales_database";
42
43
44
```

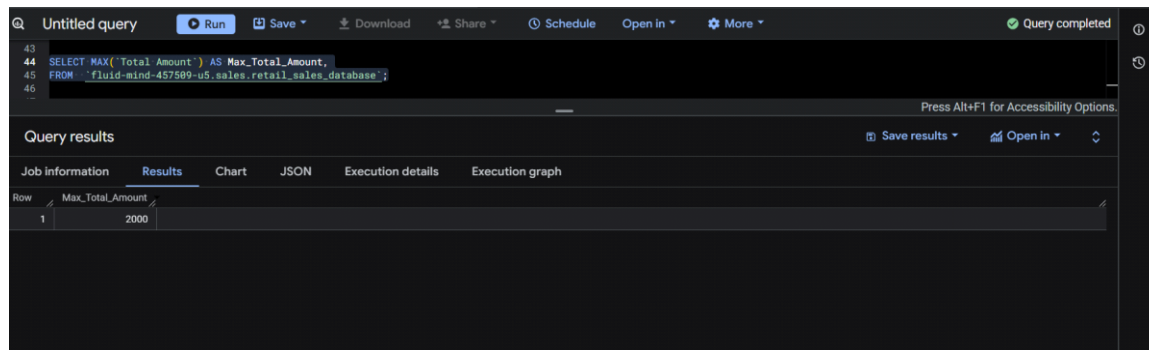
Query results Save results Open in

Job information Results Chart JSON Execution details Execution graph

Row	Total_Quantity
1	2514

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

Expected output: Max_Total_Amount



The screenshot shows a SQL query editor with a query to find the maximum total amount. The query is executed, and the result is displayed in a table.

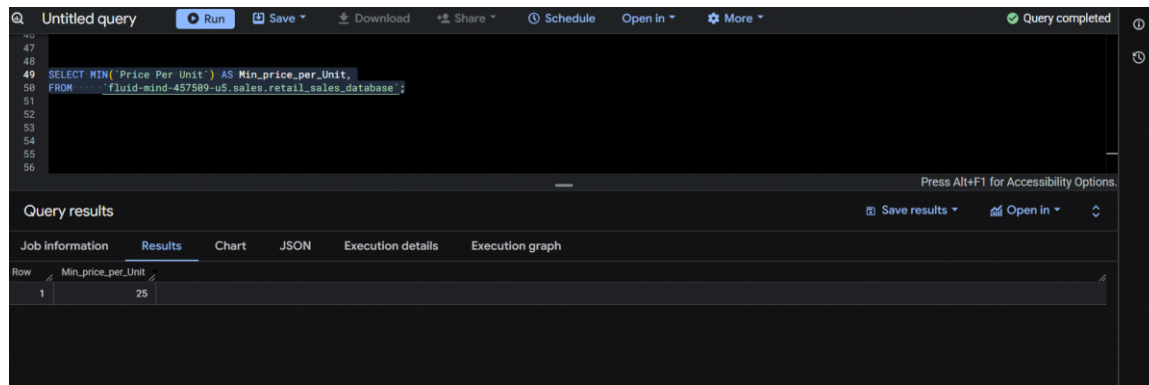
```
43
44 SELECT MAX('Total Amount') AS Max_Total_Amount,
45 FROM 'fluid-mind-457589-u5.sales.retail_sales_database';
46
```

Query results

Row	Max_Total_Amount
1	2000

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

Expected output: Min_Price_per_Unit



The screenshot shows a SQL query editor with a query to find the minimum price per unit. The query is executed, and the result is displayed in a table.

```
47
48
49 SELECT MIN('Price Per Unit') AS Min_price_per_Unit,
50 FROM 'fluid-mind-457589-u5.sales.retail_sales_database';
51
52
53
54
55
56
```

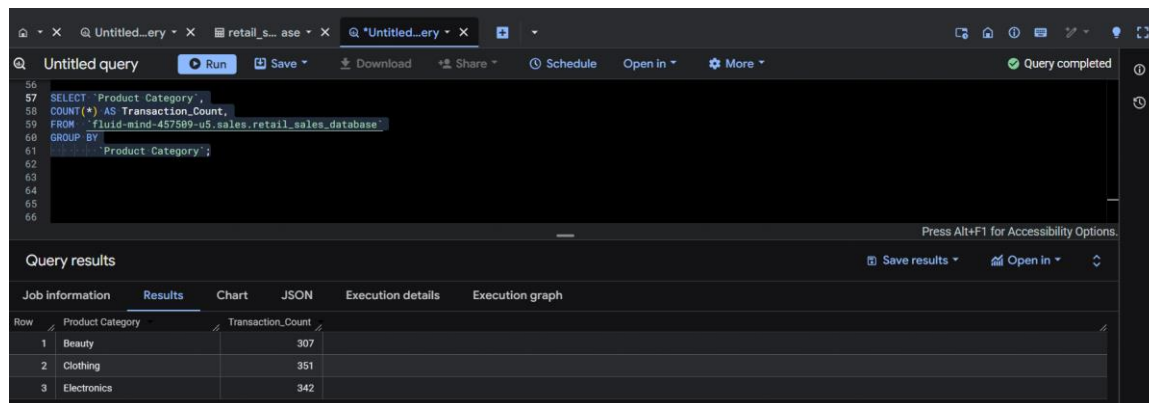
Query results

Row	Min_price_per_Unit
1	25

- **GROUP BY Statement**

Q15. Find the number of transactions per Product Category.

Expected output: Product Category, Transaction_Count



The screenshot shows a SQL query editor with a query to find the number of transactions per product category. The query is executed, and the result is displayed in a table.

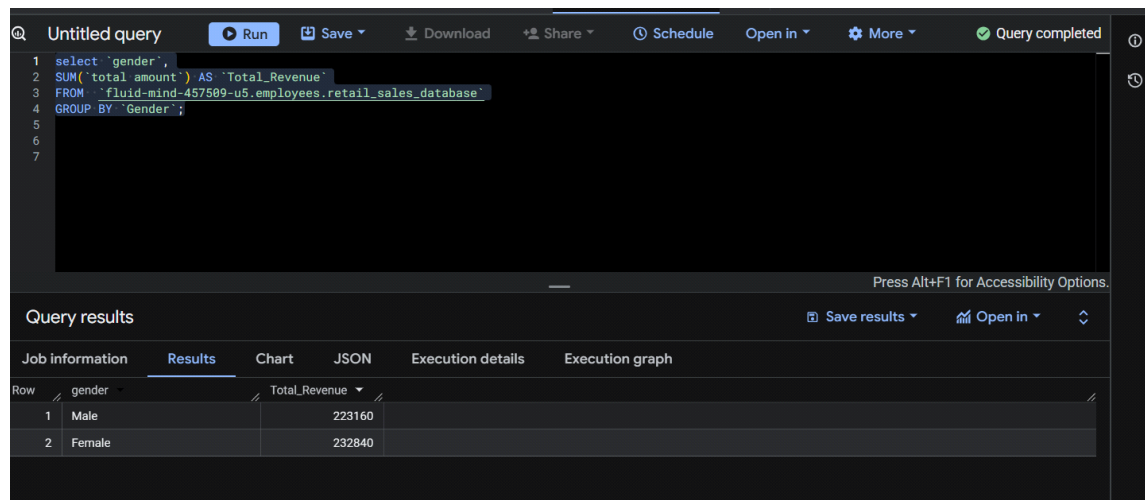
```
56
57 SELECT 'Product Category',
58 COUNT(*) AS Transaction_Count,
59 FROM 'fluid-mind-457589-u5.sales.retail_sales_database'
60 GROUP BY
61 'Product Category';
62
63
64
65
66
```

Query results

Row	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 query editor with a query to calculate total revenue by gender. The query is as follows:

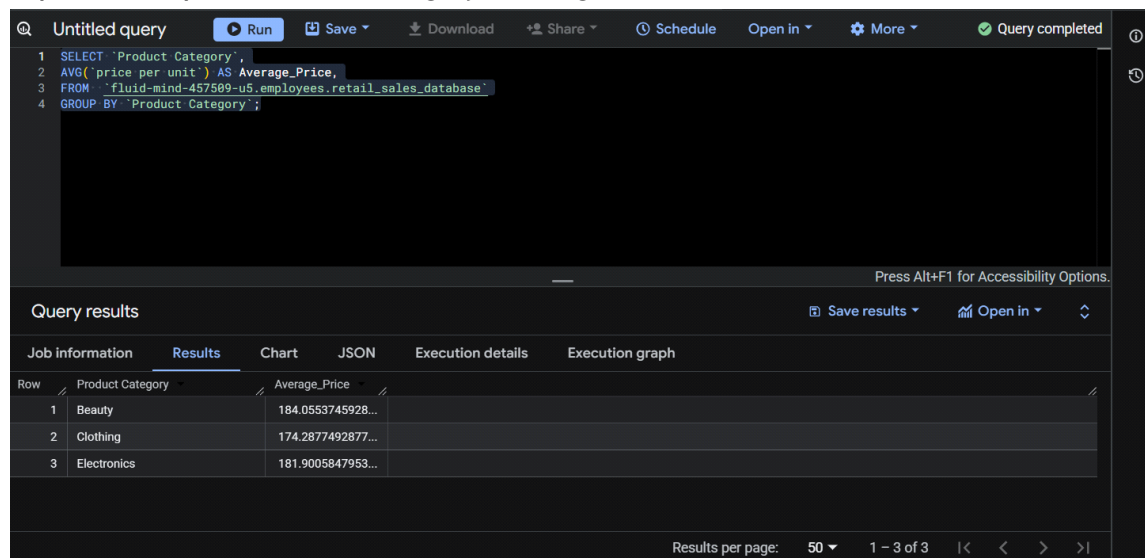
```
1 select 'gender',
2 SUM('total amount') AS 'Total_Revenue'
3 FROM 'fluid-mind-457509-u5.employees.retail_sales_database'
4 GROUP BY 'Gender';
```

The query results are displayed in a table with two columns: gender and Total_Revenue. The results are as follows:

Row	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 query editor with a query to calculate average price per unit by product category. The query is as follows:

```
1 SELECT 'Product Category',
2 AVG('price per unit') AS 'Average_Price',
3 FROM 'fluid-mind-457509-u5.employees.retail_sales_database'
4 GROUP BY 'Product Category';
```

The query results are displayed in a table with two columns: Product Category and Average_Price. The results are as follows:

Row	Product Category	Average_Price
1	Beauty	184.0553745928...
2	Clothing	174.2877492877...
3	Electronics	181.9005847953...

- **HAVING Clause**

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

Expected output: Product Category, Total_Revenue

Untitled query Run Save Download Share Schedule Open in More Query completed

```

12 SELECT 'Product Category',
13 SUM('Total Amount') AS 'Total_Revenue'
14 FROM 'fluid-mind-457589-u5.employees.retail_sales_database'
15 GROUP BY 'Product Category'
16 HAVING Total_Revenue > 10000;
17

```

Press Alt+F1 for Accessibility Options.

Query results Save results Open in

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	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

Untitled query Run Save Download Share Schedule Open in More Query completed

```

5
6 SELECT 'Product Category',
7 AVG('QUANTITY') AS 'Average_Quantity'
8 FROM 'fluid-mind-457589-u5.employees.retail_sales_database'
9 GROUP BY 'Product Category'
10 HAVING 'Average_Quantity' > 2;
11
12
13

```

Press Alt+F1 for Accessibility Options.

Query results Save results Open in

Job information	Results	Chart	JSON	Execution details	Execution graph
Row	Product Category	Average_Quantity			
1	Beauty	2.511400651465...			
2	Clothing	2.547008547008...			
3	Electronics	2.482456140350...			

- CASE Statement

Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.

Expected output: Transaction ID, Total Amount, Spending_Level

Untitled query Run Save Download Share Schedule Open in More Query completed

```

1
2
3
4 SELECT 'Transaction ID',
5       'Total Amount',
6       CASE
7         WHEN 'Total Amount' > 1000 THEN 'High'
8         ELSE 'Low'
9       END AS Spending_Level
10 FROM   'fluid-mind-457509-u5.sales.retail_sales_database';
11

```

Press Alt+F1 for Accessibility Options.

Query results Save results Open in

Job Information	Results	Chart	JSON	Execution details	Execution graph
Row	Transaction ID	Total Amount	Spending_Level		
1	191	25	Low		
2	204	25	Low		
3	230	25	Low		
4	232	25	Low		
5	309	25	Low		
6	310	25	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

```

37
38 SELECT 'Customer ID',
39       'AGE',
40       CASE
41         WHEN 'AGE' BETWEEN 30 AND 59 THEN 'Adult'
42         WHEN 'AGE' >= 60 THEN 'SENIOR'
43         ELSE 'YOUTH'
44       END AS Age_Group
45 FROM   'fluid-mind-457509-u5.sales.retail_sales_database';
46
47

```

Press Alt+F1 for Accessibility Options.

Query results Save results Open in

Job Information	Results	Chart	JSON	Execution details	Execution graph
Row	Customer ID	AGE	Age_Group		
1	CUST191	64	SENIOR		
2	CUST204	39	Adult		
3	CUST230	54	Adult		
4	CUST232	43	Adult		
5	CUST309	26	YOUTH		
6	CUST310	28	YOUTH		

