BigQuery Exercise -Solutions

Sonali Matadin

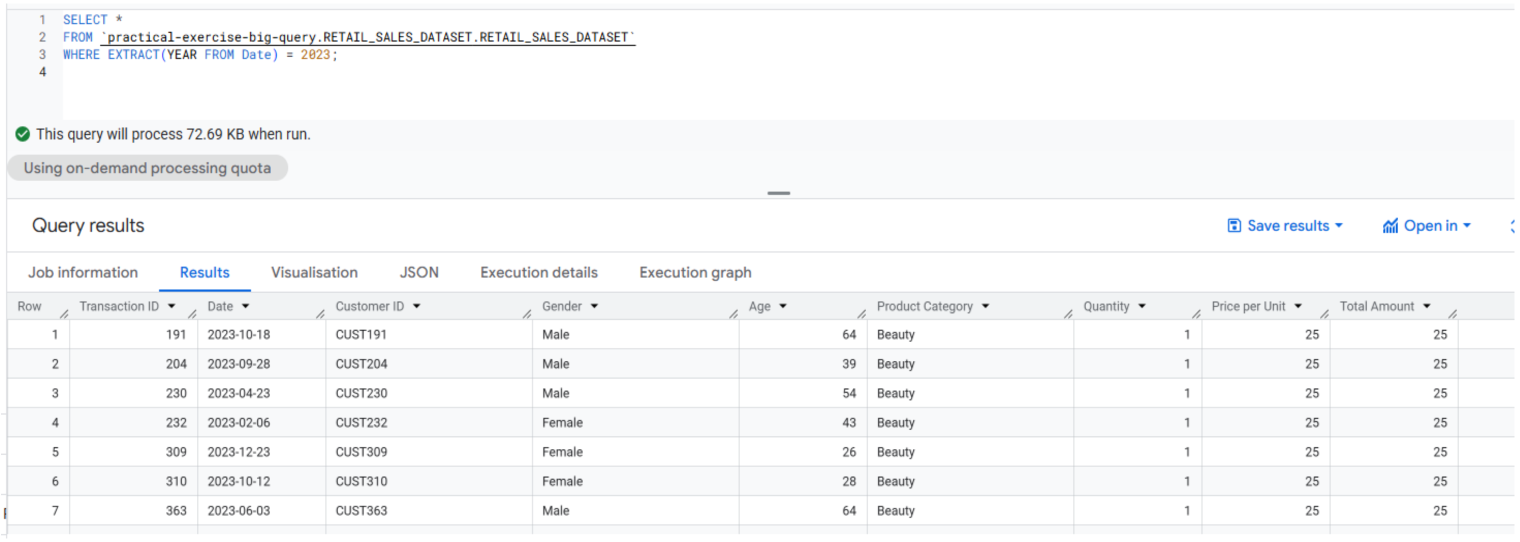
Question 1: WHERE Clause Q1. Filter all transactions that occurred in the year 2023. Expected output: All columns

Sql query:

SELECT *

FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`

WHERE EXTRACT(YEAR FROM Date) = 2023;



```
SELECT *
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
WHERE EXTRACT(YEAR FROM Date) = 2023;
```

This query will process 72.69 KB when run.

Using on-demand processing quota

Query results

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

Question 2: Filtering + Conditions Q2. Display all transactions where the Total Amount is more than the average Total Amount of the entire dataset. Expected output: All columns

Sql query:

SELECT *

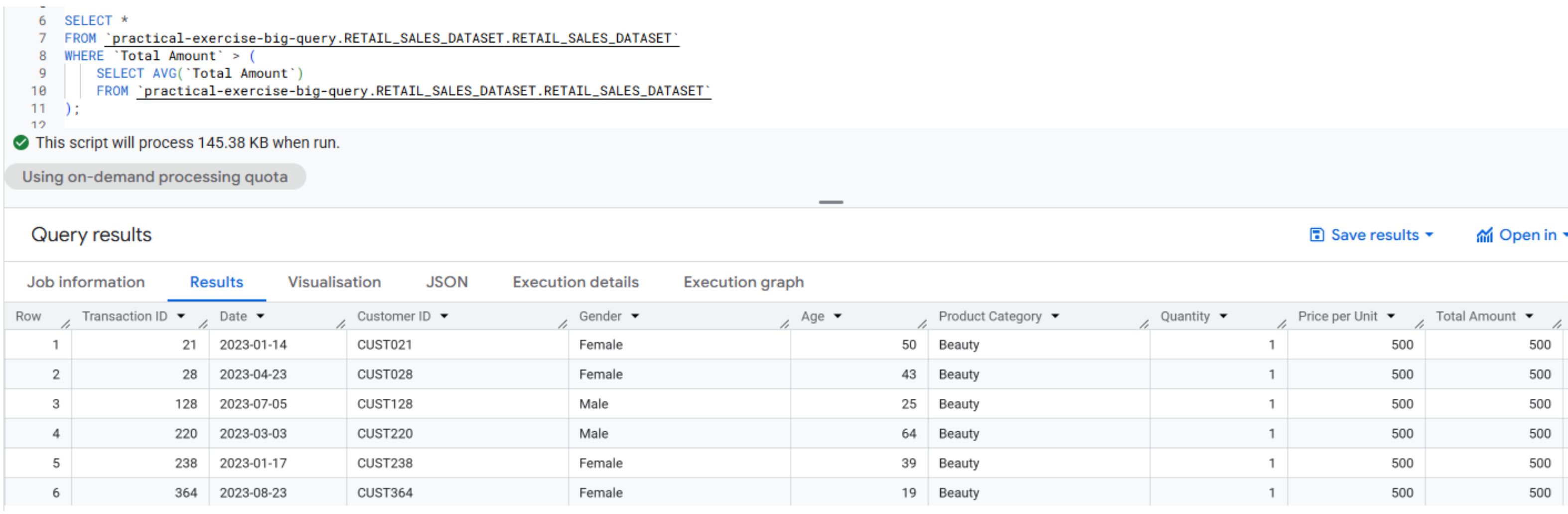
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`

WHERE `Total Amount` > (

SELECT AVG(`Total Amount`)

FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`

);



```
SELECT *
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
WHERE `Total Amount` > (
    SELECT AVG(`Total Amount`)
    FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
);
```

This script will process 145.38 KB when run.

Using on-demand processing quota

Query results

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
1	21	2023-01-14	CUST021	Female	50	Beauty	1	500	500
2	28	2023-04-23	CUST028	Female	43	Beauty	1	500	500
3	128	2023-07-05	CUST128	Male	25	Beauty	1	500	500
4	220	2023-03-03	CUST220	Male	64	Beauty	1	500	500
5	238	2023-01-17	CUST238	Female	39	Beauty	1	500	500
6	364	2023-08-23	CUST364	Female	19	Beauty	1	500	500

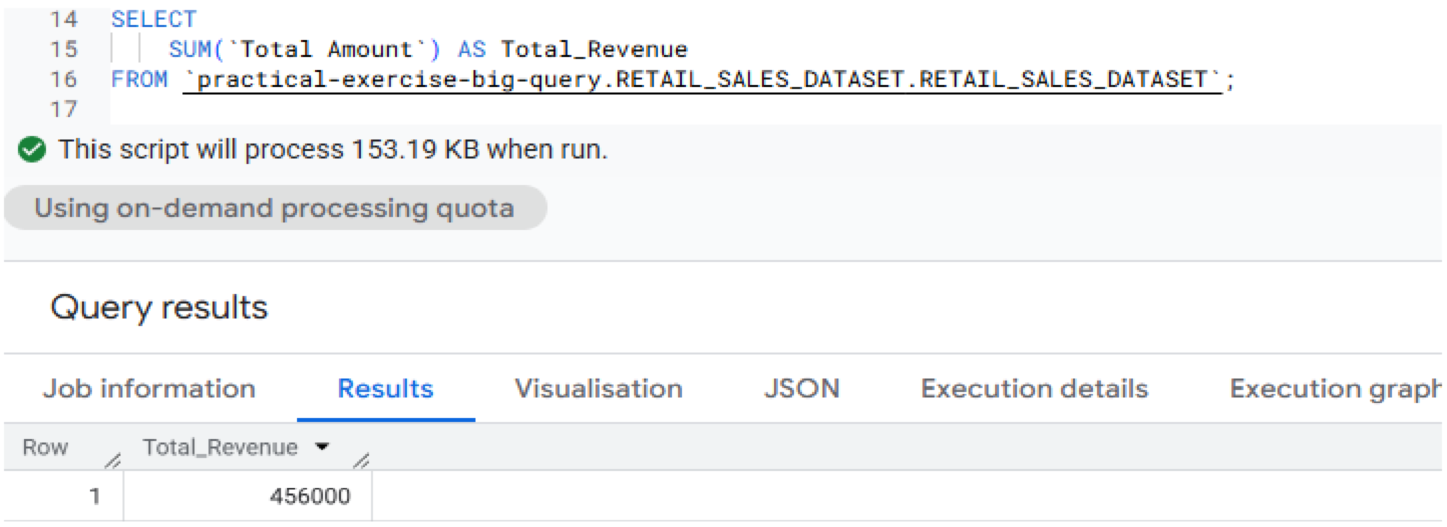
Question 3: Aggregate Functions Q3. Calculate the total revenue (sum of Total Amount). Expected output: Total_Revenue

Sql query:

SELECT

SUM(`Total Amount`) AS Total_Revenue

FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;



```
SELECT
    SUM(`Total Amount`) AS Total_Revenue
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;
```

This script will process 153.19 KB when run.

Row	Total_Revenue
1	456000

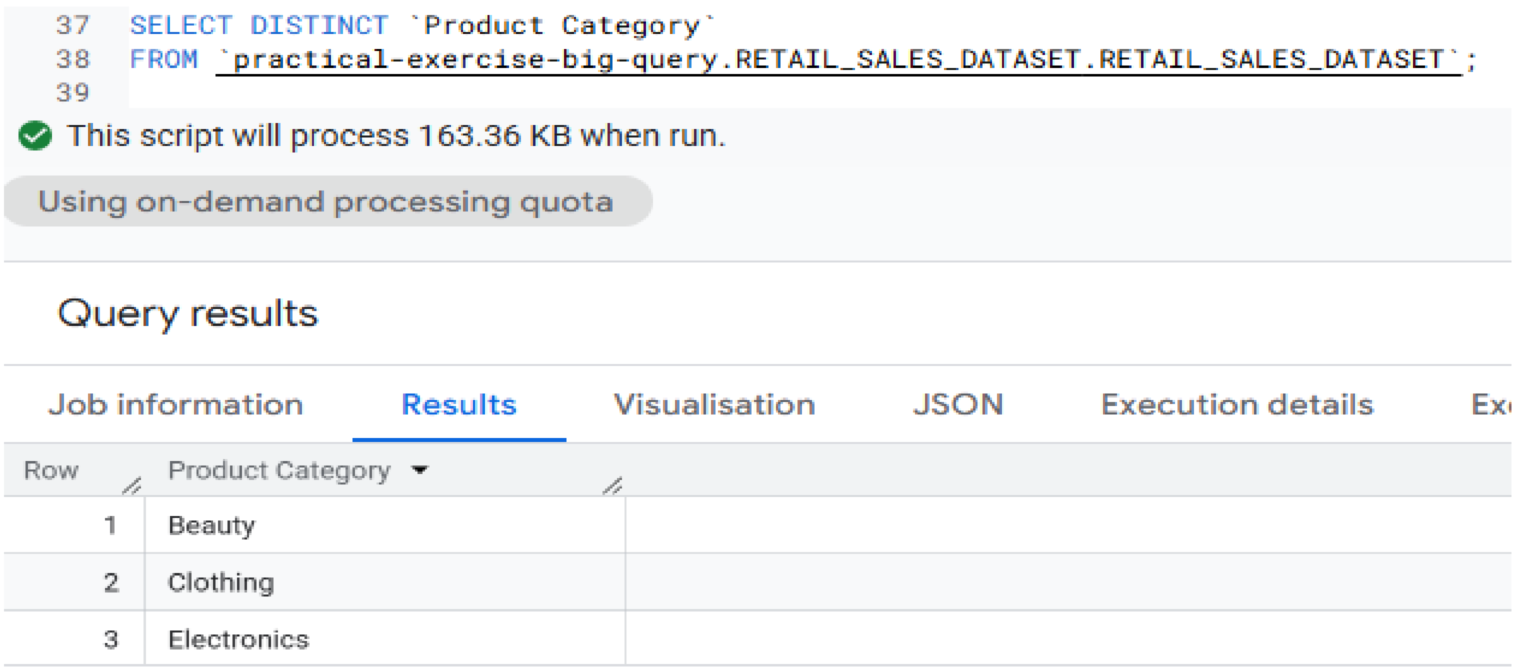
Question 4: DISTINCT

Q4. Display all distinct Product Categories in the dataset. Expected output: Product_Category

Sql query:

SELECT DISTINCT `Product Category`

FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;



```
SELECT DISTINCT `Product Category`
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;
```

This script will process 163.36 KB when run.

Row	Product Category
1	Beauty
2	Clothing
3	Electronics

Question 5: GROUP BY

Q5. For each Product Category, calculate the total quantity sold. Expected output: Product_Category, Total_Quantity

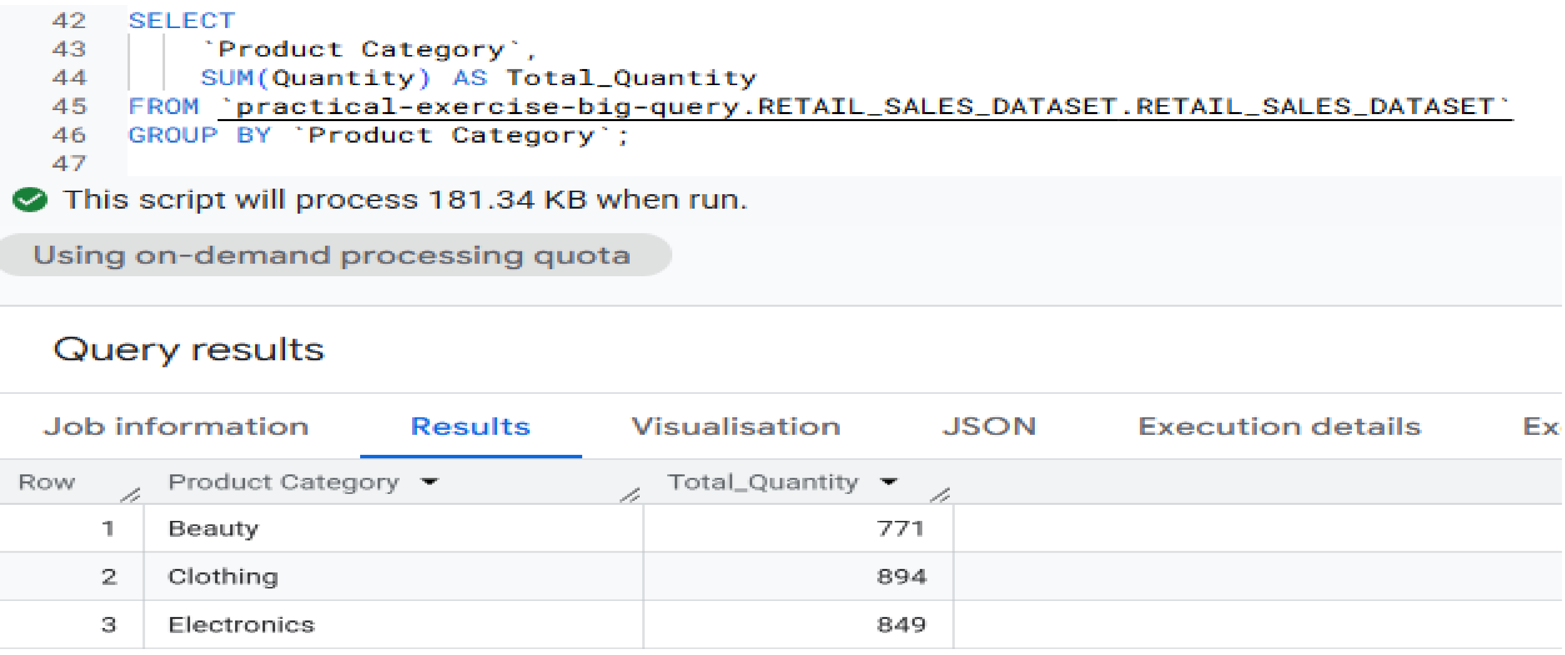
Sql query: SELECT

`Product Category`,

SUM(Quantity) AS Total_Quantity

FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`

GROUP BY `Product Category`;



```
SELECT
    `Product Category`,
    SUM(Quantity) AS Total_Quantity
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
GROUP BY `Product Category`;
```

This script will process 181.34 KB when run.

Using on-demand processing quota

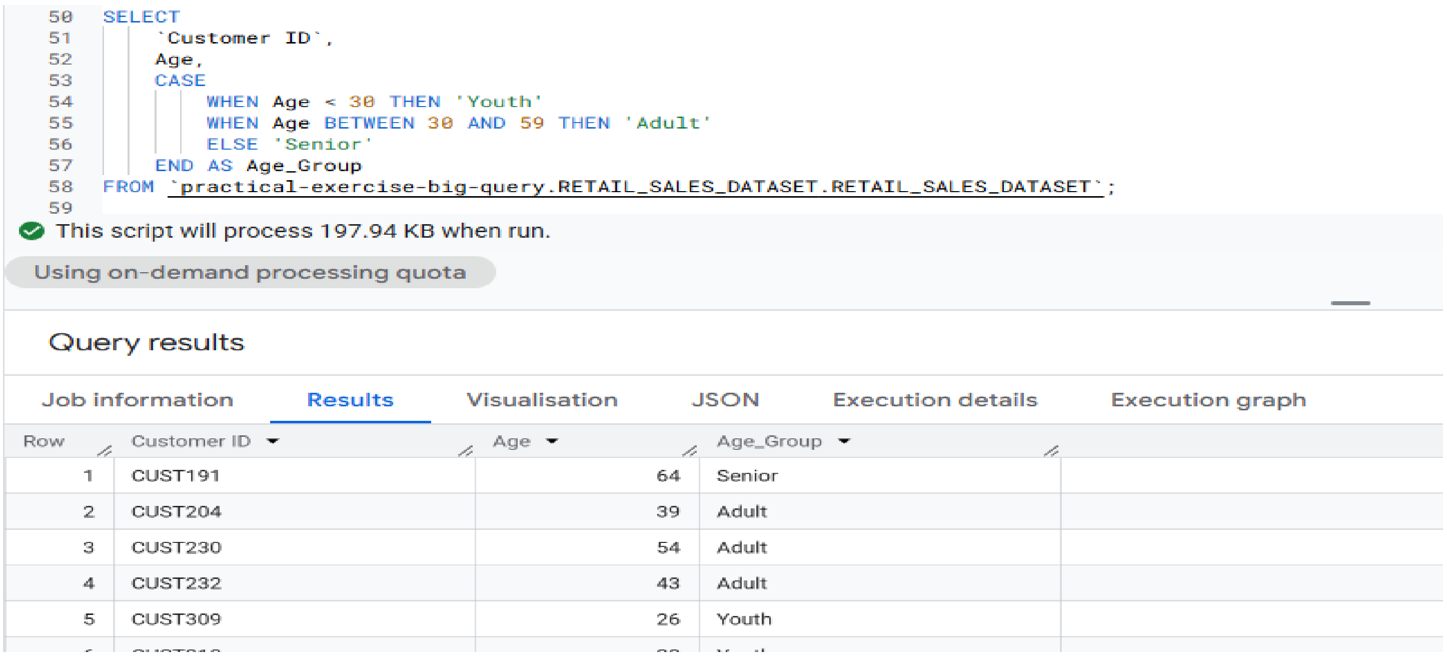
Query results

Row	Product Category	Total_Quantity
1	Beauty	771
2	Clothing	894
3	Electronics	849

Question 6: CASE Statement Q6. Create a column called Age_Group that classifies customers as ‘Youth’ (<30), ‘Adult’ (30–59), and ‘Senior’ (60+). Expected output: Customer_ID, Age, Age_Group

Sql query:

```
SELECT
  `Customer ID`,
  Age,
  CASE
    WHEN Age < 30 THEN 'Youth'
    WHEN Age BETWEEN 30 AND 59 THEN 'Adult'
    ELSE 'Senior'
  END AS Age_Group
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;
```



```
SELECT
    `Customer ID`,
    Age,
    CASE
        WHEN Age < 30 THEN 'Youth'
        WHEN Age BETWEEN 30 AND 59 THEN 'Adult'
        ELSE 'Senior'
    END AS Age_Group
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;
```

This script will process 197.94 KB when run.

Using on-demand processing quota

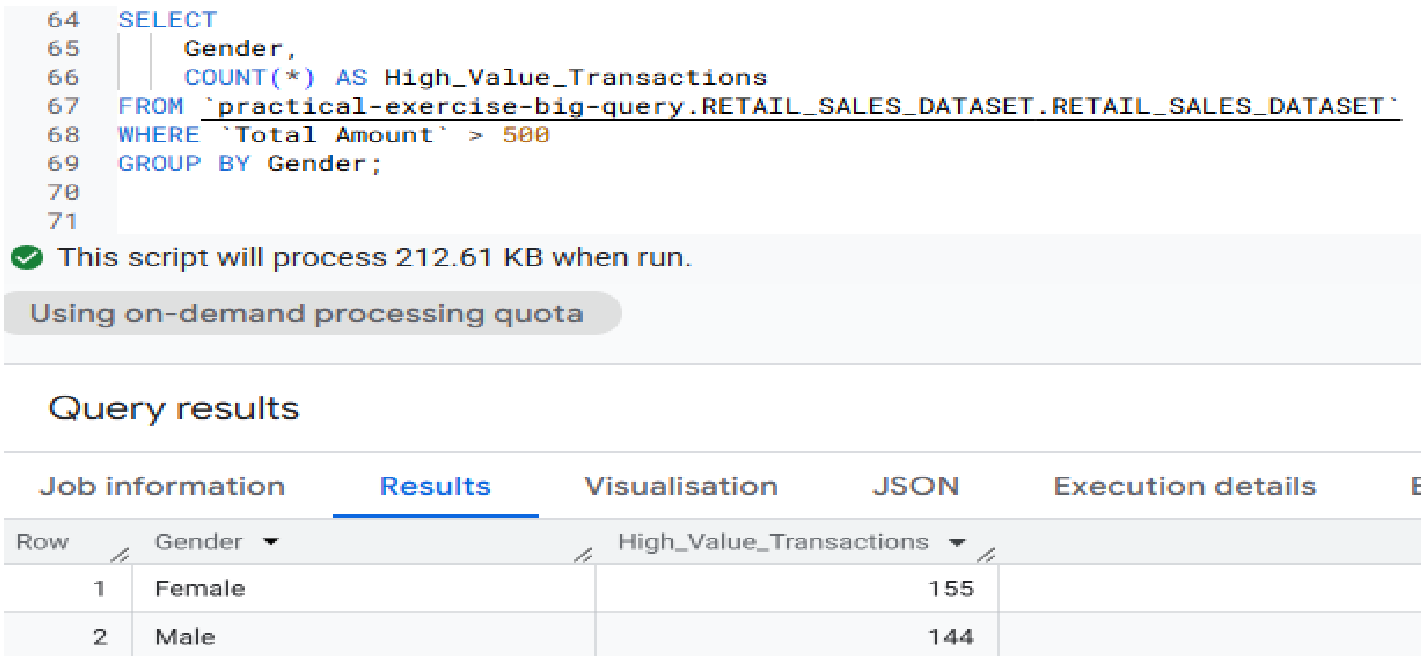
Query results

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

Question 7: Conditional Aggregation Q7. For each Gender, count how many high-value transactions occurred (where Total Amount > 500). Expected output: Gender, High_Value_Transactions

Sql query:

```
SELECT

   Gender,

   COUNT(*) AS High_Value_Transactions

FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`

WHERE `Total Amount` > 500

GROUP BY Gender;
```



```
SELECT
    Gender,
    COUNT(*) AS High_Value_Transactions
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
WHERE `Total Amount` > 500
GROUP BY Gender;
```

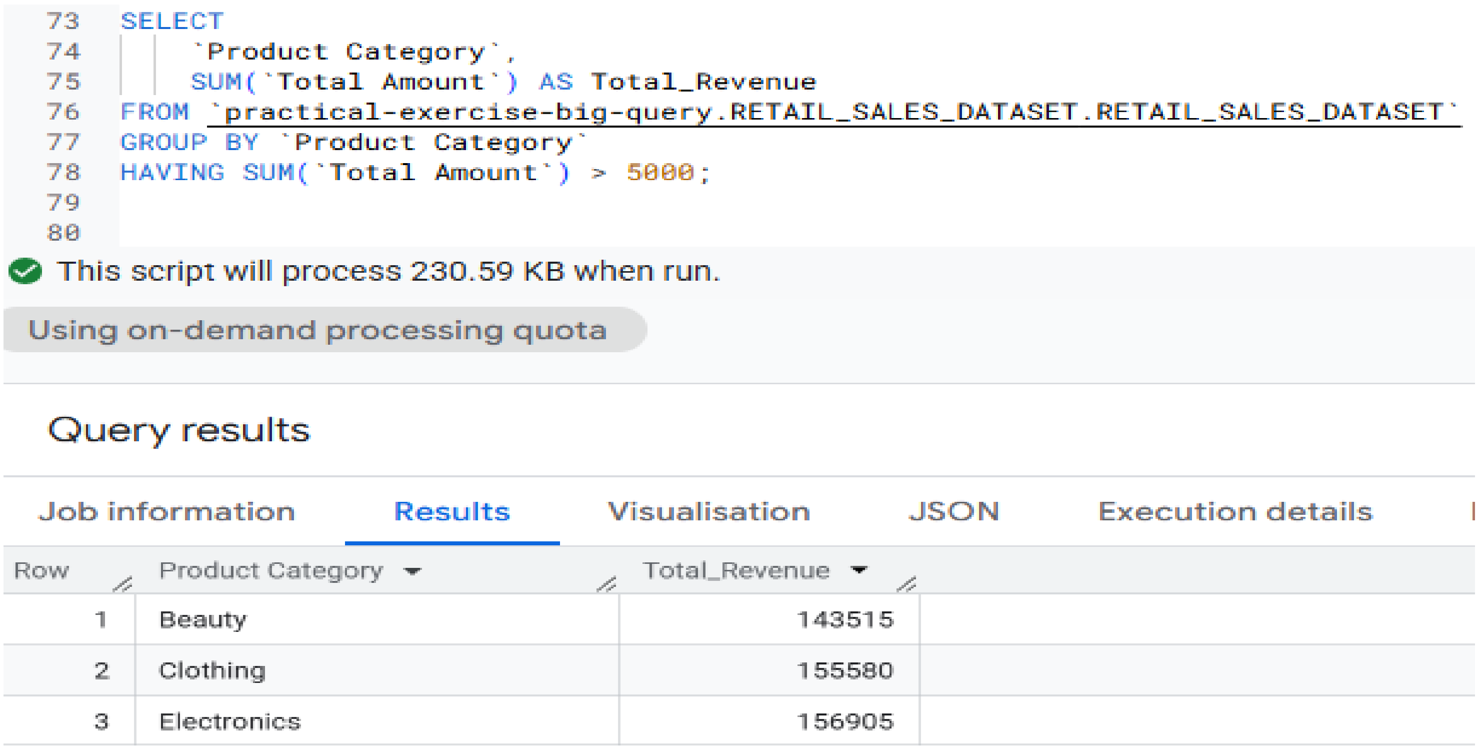
This script will process 212.61 KB when run.

Row	Gender	High_Value_Transactions
1	Female	155
2	Male	144

Question 8: HAVING Clause Q8. For each Product Category, show only those categories where the total revenue exceeds 5,000. Expected output: Product_Category, Total_Revenue

Sql query:

```
SELECT
   `Product Category`,
   SUM(`Total Amount`) AS Total_Revenue
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
GROUP BY `Product Category`
HAVING SUM(`Total Amount`) > 5000;
```



```
SELECT
    `Product Category`,
    SUM(`Total Amount`) AS Total_Revenue
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
GROUP BY `Product Category`
HAVING SUM(`Total Amount`) > 5000;
```

This script will process 230.59 KB when run.

Using on-demand processing quota

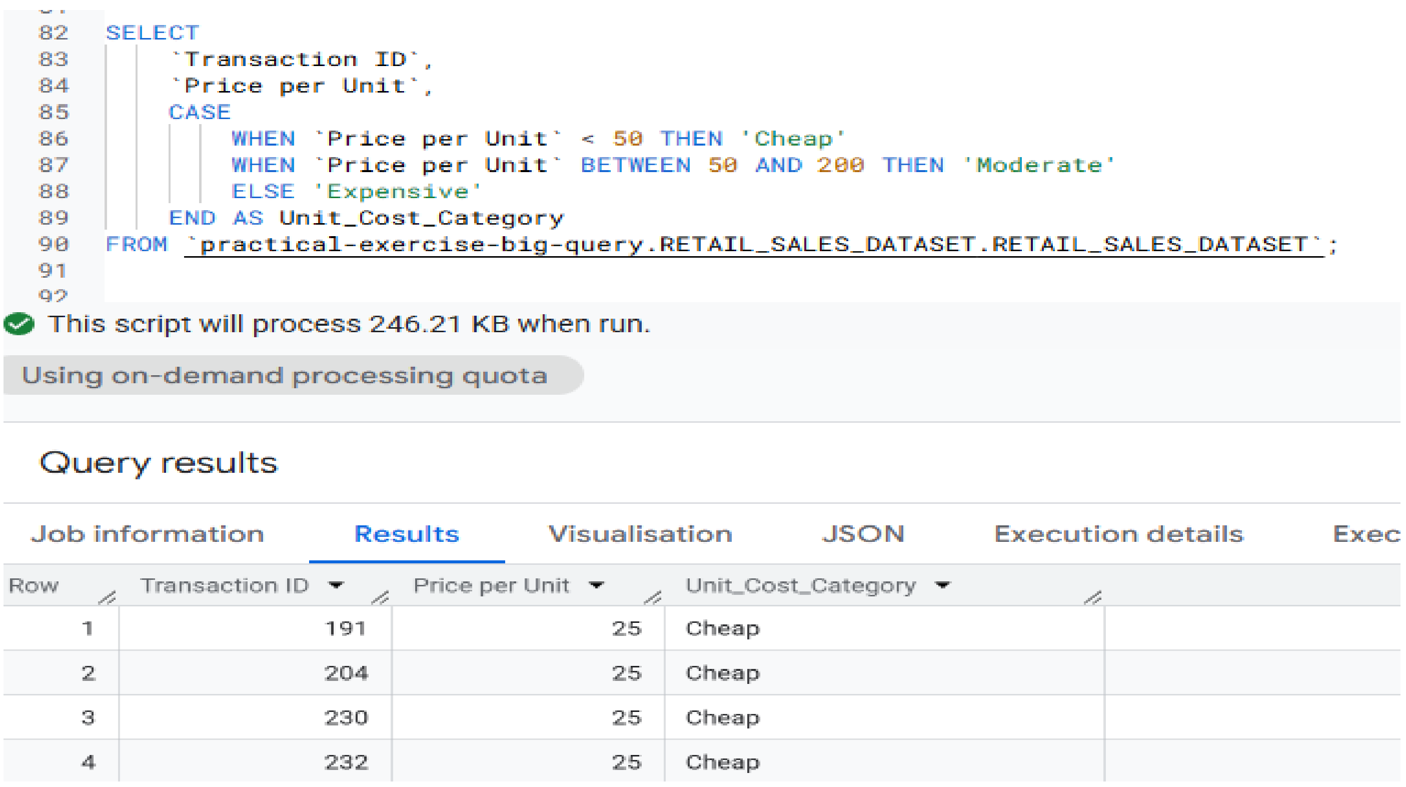
Query results

Row	Product Category	Total_Revenue
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

Question 9: Calculated Fields Q9. Display a new column called Unit_Cost_Category that labels a transaction as: – 'Cheap' if Price per Unit < 50 – 'Moderate' if Price per Unit between 50 and 200 – 'Expensive' if Price per Unit > 200 Expected output: Transaction_ID, Price_per_Unit, Unit_Cost_Category

Sql query:

```
SELECT
  `Transaction ID`,
  `Price per Unit`,
  CASE
    WHEN `Price per Unit` < 50 THEN 'Cheap'
    WHEN `Price per Unit` BETWEEN 50 AND 200 THEN 'Moderate'
    ELSE 'Expensive'
  END AS Unit_Cost_Category
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;
```



```
SELECT
    `Transaction ID`,
    `Price per Unit`,
    CASE
        WHEN `Price per Unit` < 50 THEN 'Cheap'
        WHEN `Price per Unit` BETWEEN 50 AND 200 THEN 'Moderate'
        ELSE 'Expensive'
    END AS Unit_Cost_Category
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`;
```

This script will process 246.21 KB when run.

Using on-demand processing quota

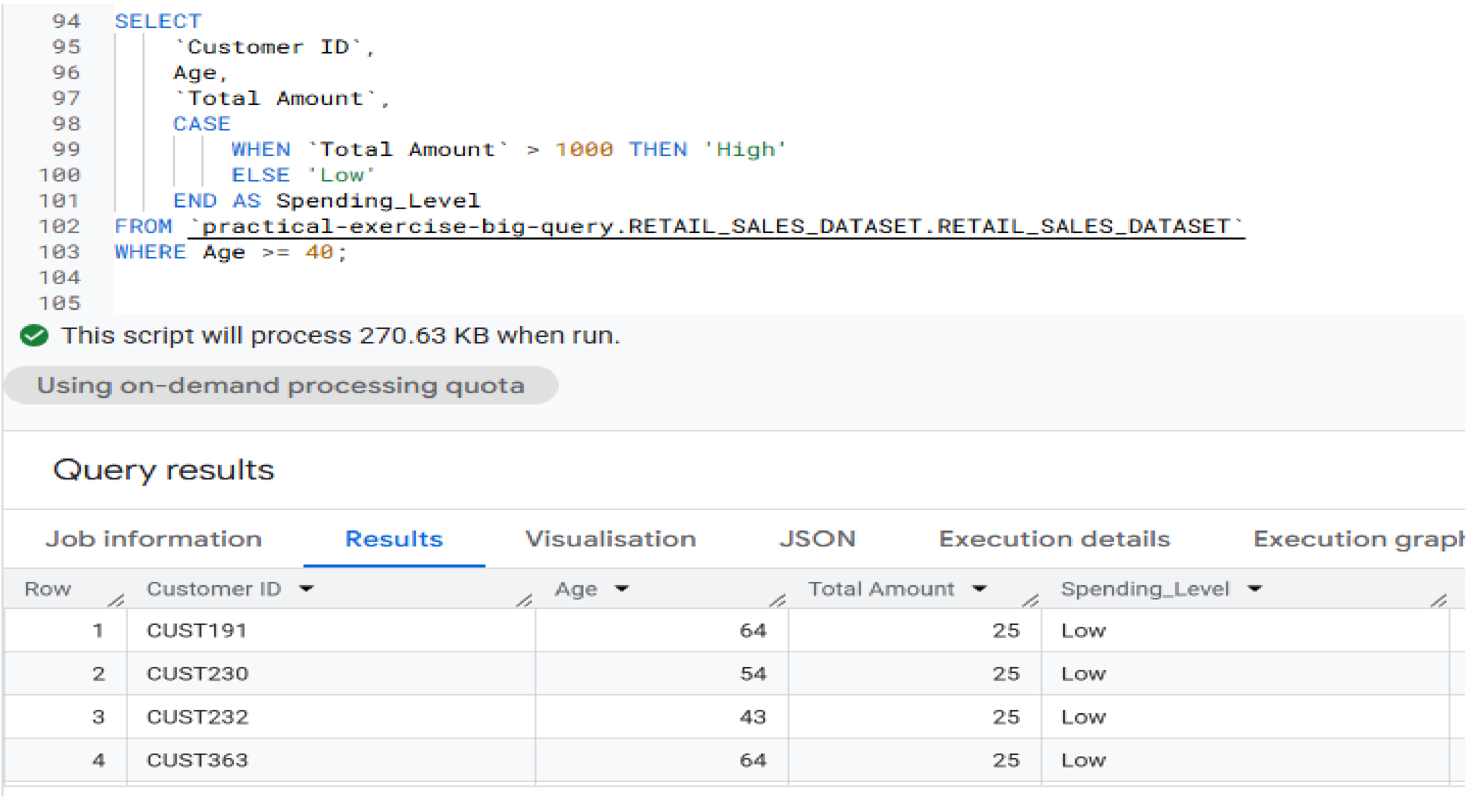
Query results

Row	Transaction ID	Price per Unit	Unit_Cost_Category
1	191	25	Cheap
2	204	25	Cheap
3	230	25	Cheap
4	232	25	Cheap

Question 10: Combining WHERE + CASE Q10. Display all transactions from customers aged 40 or older and add a column Spending_Level showing ‘High’ if Total Amount > 1000, otherwise ‘Low’. Expected output: Customer_ID, Age, Total_Amount, Spending_Level

Sql query:

```
SELECT
  `Customer ID`,
  Age,
  `Total Amount`,
  CASE
    WHEN `Total Amount` > 1000 THEN 'High'
    ELSE 'Low'
  END AS Spending_Level
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
WHERE Age >= 40;
```



```
SELECT
    `Customer ID`,
    Age,
    `Total Amount`,
    CASE
        WHEN `Total Amount` > 1000 THEN 'High'
        ELSE 'Low'
    END AS Spending_Level
FROM `practical-exercise-big-query.RETAIL_SALES_DATASET.RETAIL_SALES_DATASET`
WHERE Age >= 40;
```

This script will process 270.63 KB when run.

Using on-demand processing quota

Query results

Row	Customer ID	Age	Total Amount	Spending_Level
1	CUST191	64	25	Low
2	CUST230	54	25	Low
3	CUST232	43	25	Low
4	CUST363	64	25	Low