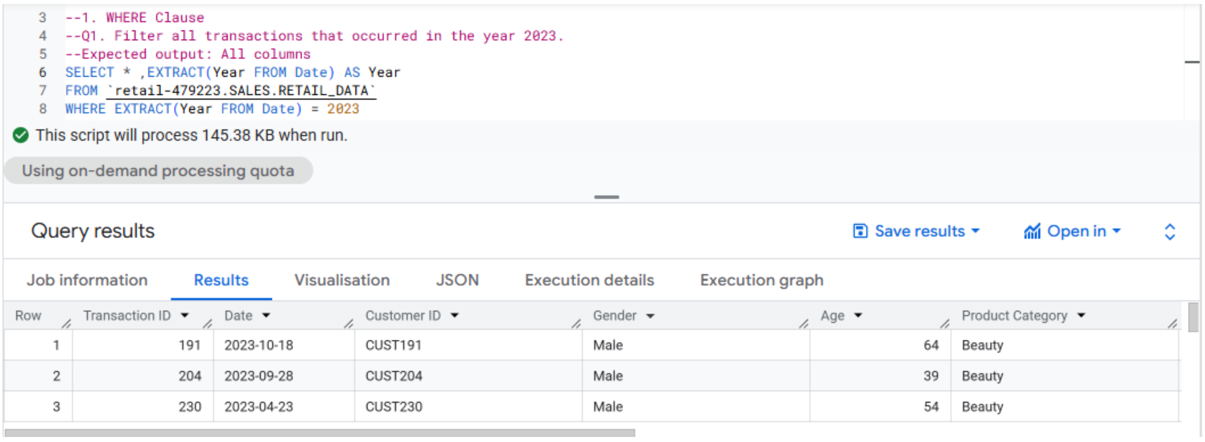
Google Big Query

Question 1



```
--1. WHERE Clause
--Q1. Filter all transactions that occurred in the year 2023.
--Expected output: All columns
SELECT * ,EXTRACT(Year FROM Date) AS Year
FROM `retail-479223.SALES.RETAIL_DATA`
WHERE EXTRACT(Year FROM Date) = 2023
```

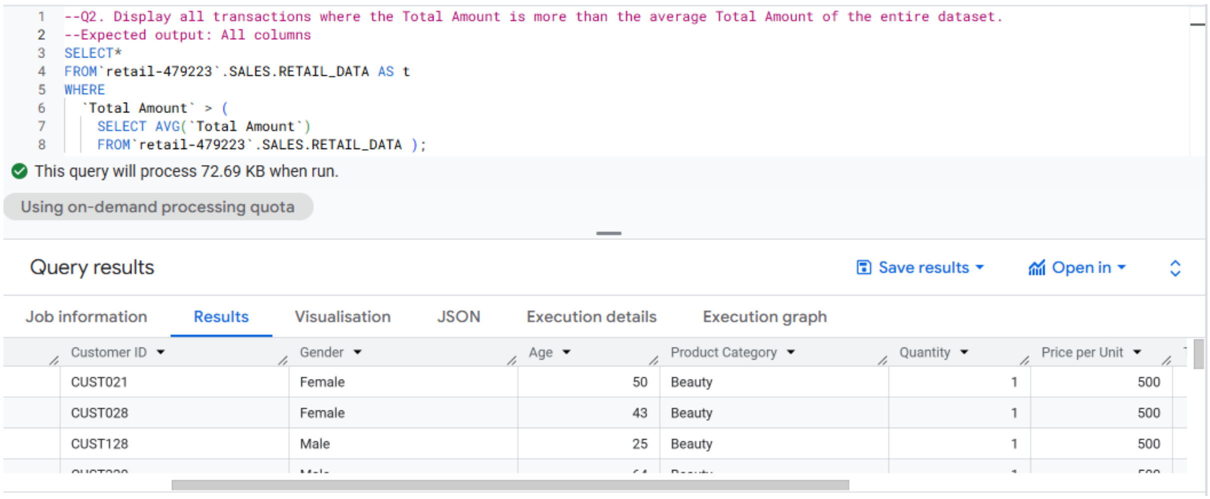
This script will process 145.38 KB when run.

Using on-demand processing quota

Query results

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	191	2023-10-18	CUST191	Male	64	Beauty
2	204	2023-09-28	CUST204	Male	39	Beauty
3	230	2023-04-23	CUST230	Male	54	Beauty

Question 2



```
--Q2. Display all transactions where the Total Amount is more than the average Total Amount of the entire dataset.
--Expected output: All columns
SELECT*
FROM`retail-479223`.SALES.RETAIL_DATA AS t
WHERE
  `Total Amount` > (
    SELECT AVG(`Total Amount`)
    FROM`retail-479223`.SALES.RETAIL_DATA );
```

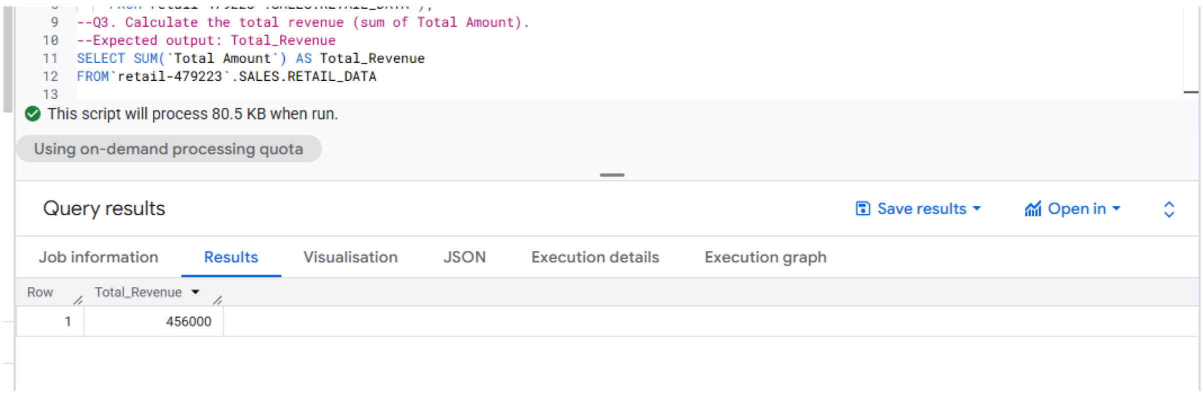
This query will process 72.69 KB when run.

Using on-demand processing quota

Query results

Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
CUST021	Female	50	Beauty	1	500
CUST028	Female	43	Beauty	1	500
CUST128	Male	25	Beauty	1	500

Question 3



```
--Q3. Calculate the total revenue (sum of Total Amount).
--Expected output: Total_Revenue
SELECT SUM(`Total Amount`) AS Total_Revenue
FROM`retail-479223`.SALES.RETAIL_DATA
```

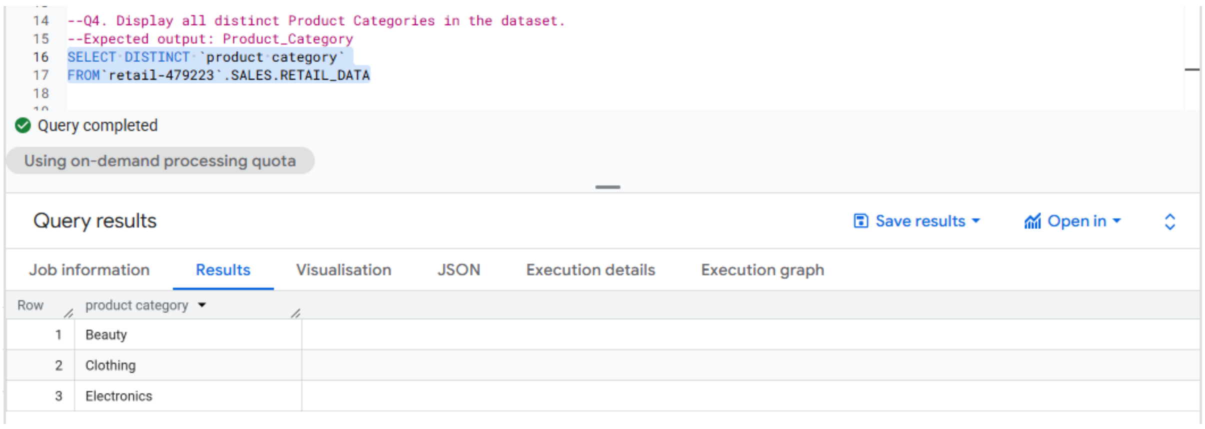
This script will process 80.5 KB when run.

Using on-demand processing quota

Query results

Row	Total_Revenue
1	456000

Question 4



```
--Q4. Display all distinct Product Categories in the dataset.
--Expected output: Product_Category
SELECT DISTINCT `product category`
FROM`retail-479223`.SALES.RETAIL_DATA
```

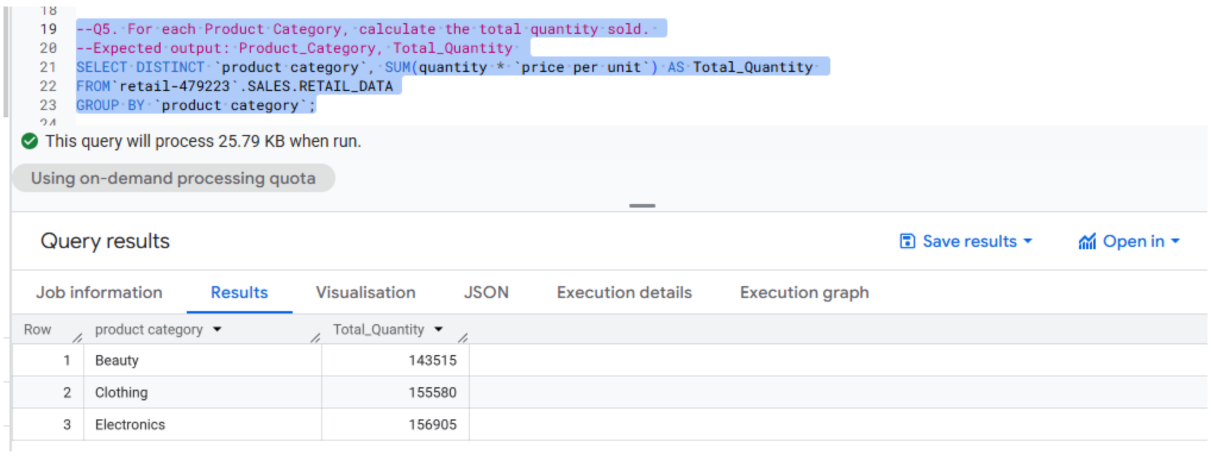
Query completed

Using on-demand processing quota

Query results

Row	product category
1	Beauty
2	Clothing
3	Electronics

Question 5



```
--Q5. For each Product Category, calculate the total quantity sold.
--Expected output: Product_Category, Total_Quantity
SELECT DISTINCT `product category`, SUM(quantity * `price per unit`) AS Total_Quantity
FROM`retail-479223`.SALES.RETAIL_DATA
GROUP BY `product category`;
```

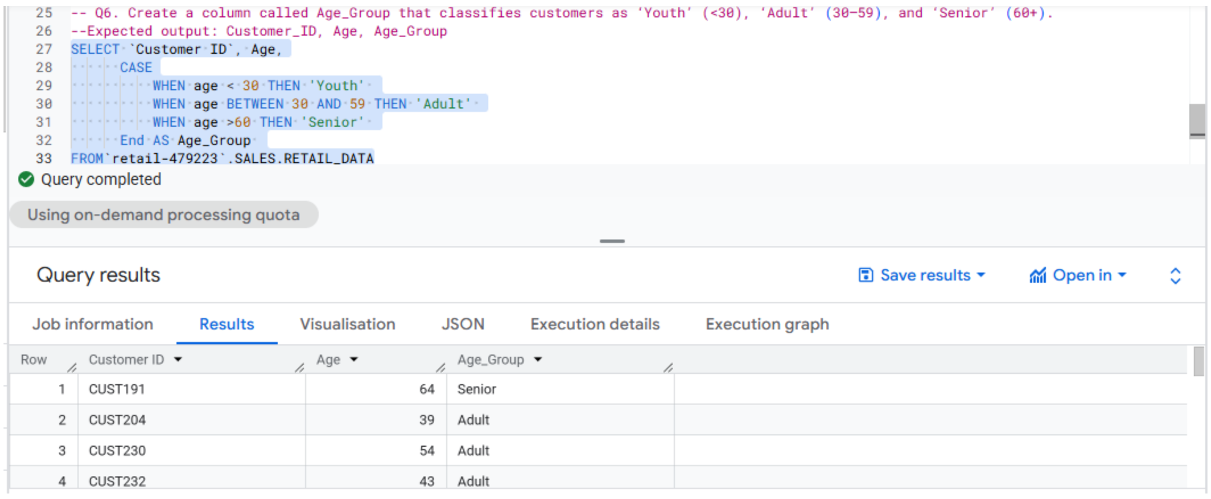
This query will process 25.79 KB when run.

Using on-demand processing quota

Query results

Row	product category	Total_Quantity
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

Question 6



```
-- 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
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-479223`.SALES.RETAIL_DATA
```

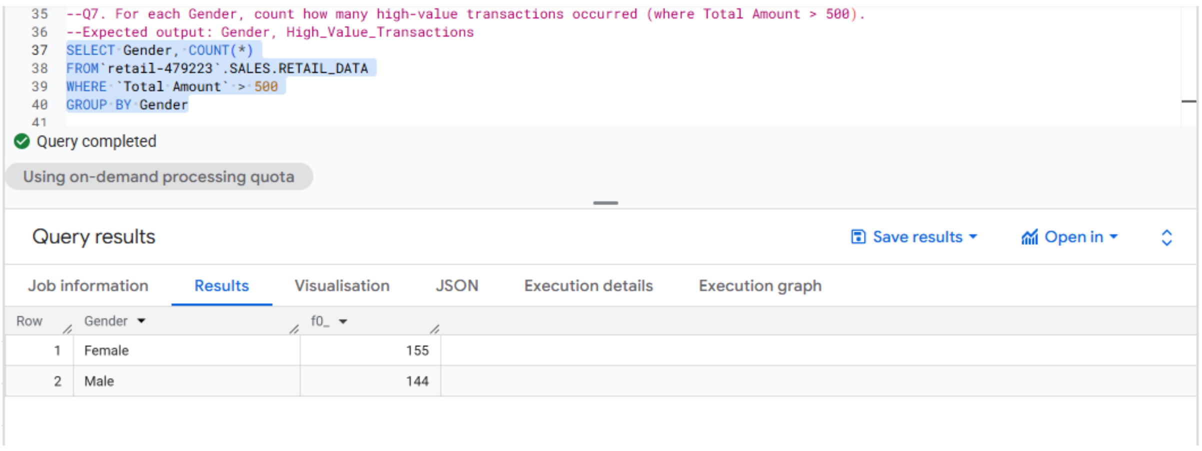
Query completed

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

Question 7



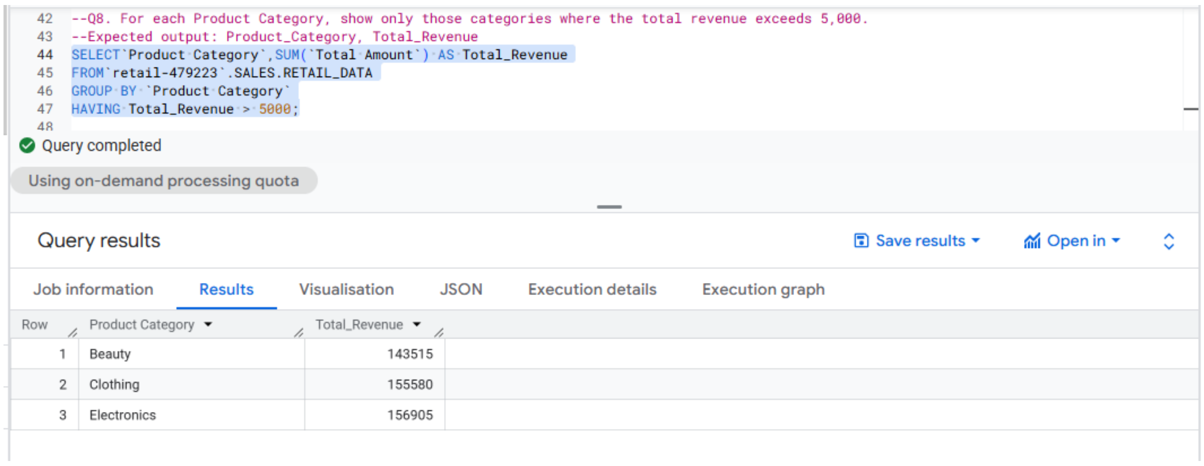
```sql
--Q7. For each Gender, count how many high-value transactions occurred (where Total Amount > 500).
--Expected output: Gender, High_Value_Transactions
SELECT Gender, COUNT(*)
FROM`retail-479223`.SALES.RETAIL_DATA
WHERE `Total Amount` > 500
GROUP BY Gender
```

Query completed

Using on-demand processing quota

Row	Gender	f0_
1	Female	155
2	Male	144

Question 8



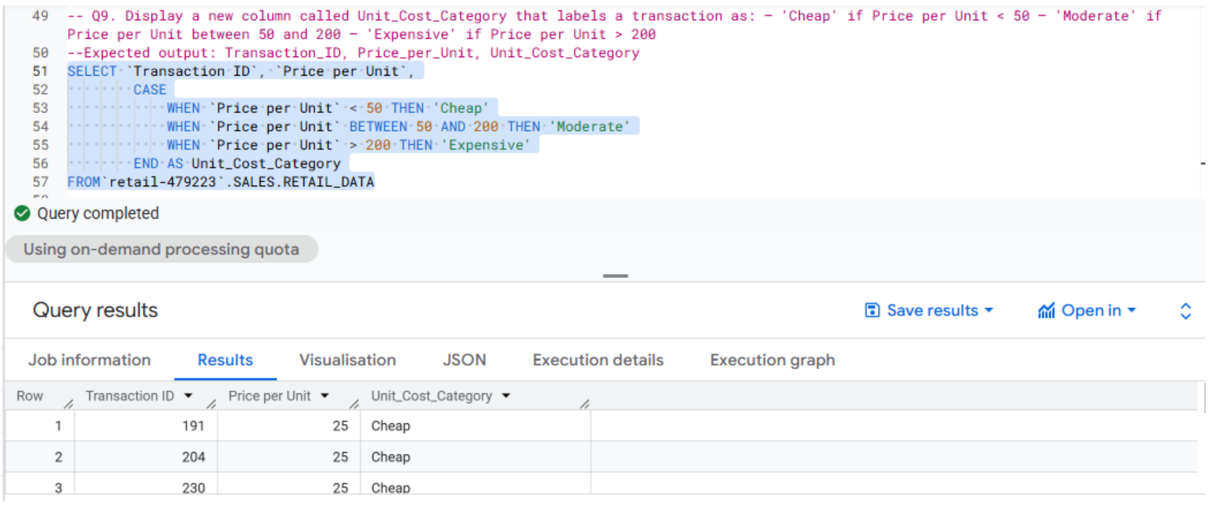
```sql
--Q8. For each Product Category, show only those categories where the total revenue exceeds 5,000.
--Expected output: Product_Category, Total_Revenue
SELECT`Product Category`,SUM(`Total Amount`) AS Total_Revenue
FROM`retail-479223`.SALES.RETAIL_DATA
GROUP BY `Product Category`
HAVING Total_Revenue > 5000;
```

Query completed

Using on-demand processing quota

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

Question 9



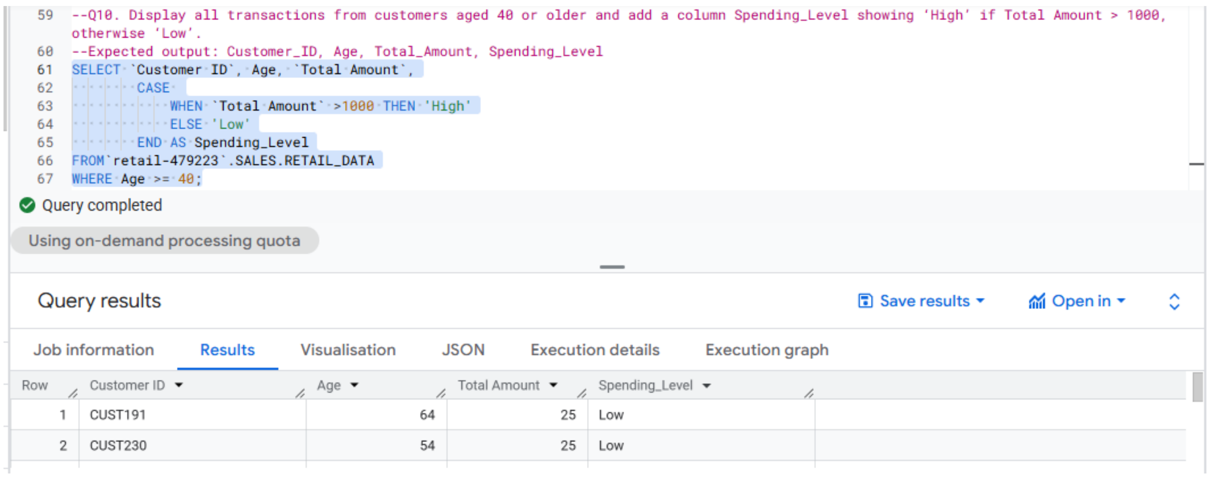
```sql
-- Q9. Display a new column called Unit_Cost_Category that labels a transaction as: – 'Cheap' if Price per Unit < 50 – 'Moderate' if
Price per Unit between 50 and 200 – 'Expensive' if Price per Unit > 200
--Expected output: Transaction_ID, Price_per_Unit, Unit_Cost_Category
SELECT `Transaction ID`, `Price per Unit`,
        CASE
            WHEN `Price per Unit` < 50 THEN 'Cheap'
            WHEN `Price per Unit` BETWEEN 50 AND 200 THEN 'Moderate'
            WHEN `Price per Unit` > 200 THEN 'Expensive'
        END AS Unit_Cost_Category
FROM`retail-479223`.SALES.RETAIL_DATA
```

Query completed

Using on-demand processing quota

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

Question 10



```
--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
SELECT `Customer ID`, Age, `Total Amount`,
       CASE
           WHEN `Total Amount` >1000 THEN 'High'
           ELSE 'Low'
       END AS Spending_Level
FROM`retail-479223`.SALES.RETAIL_DATA
WHERE Age >= 40;
```

Query completed

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