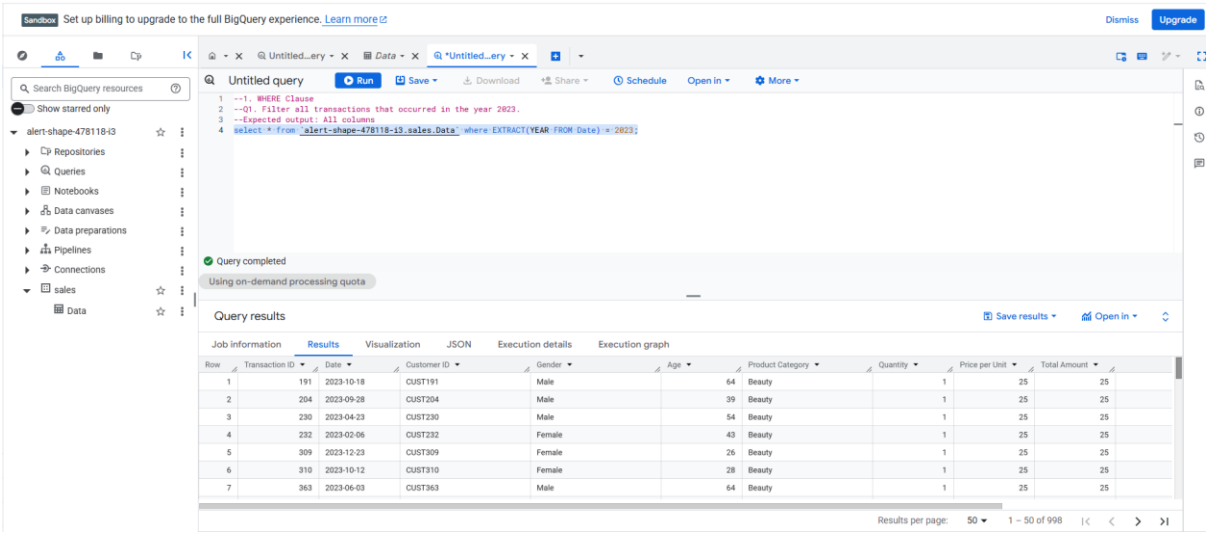
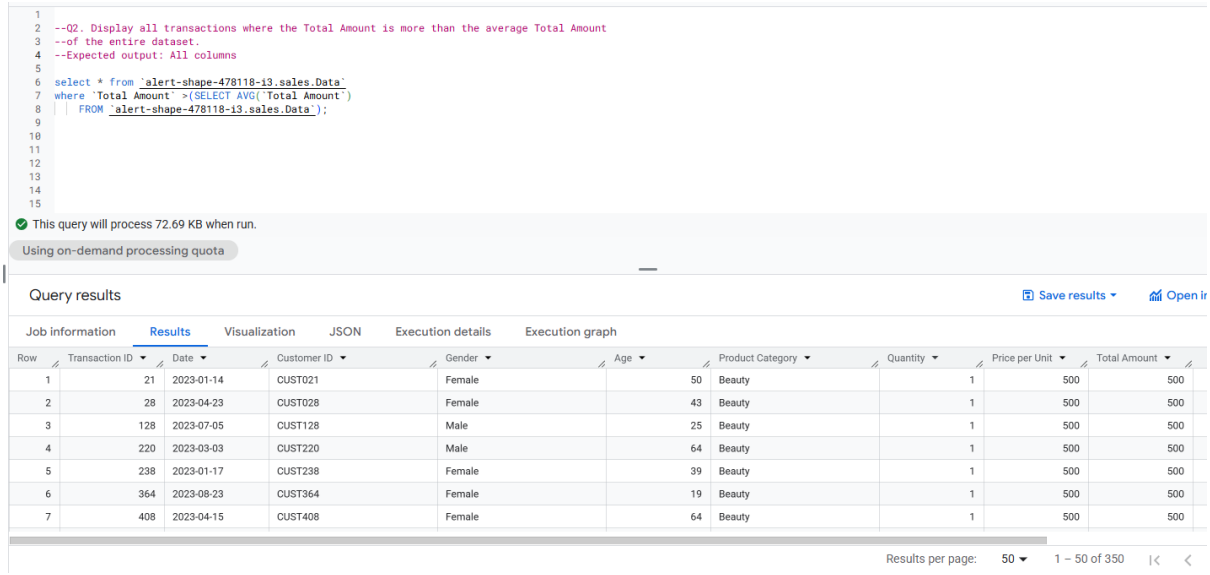
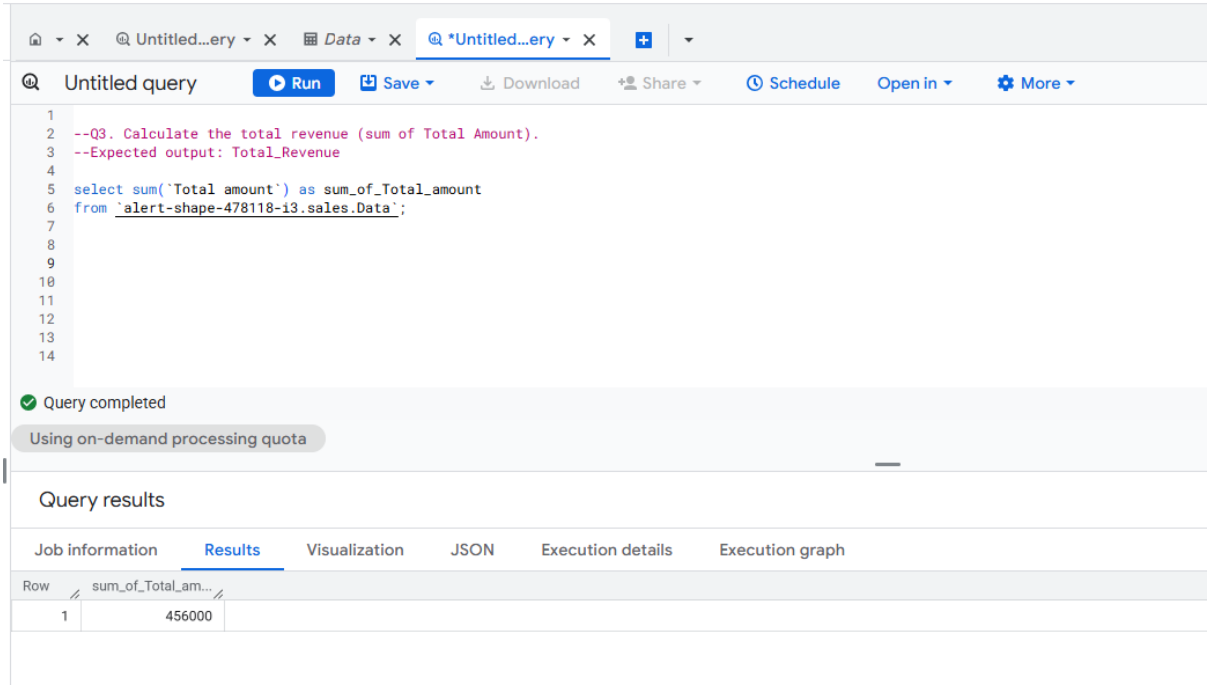
1


Sandbox Set up billing to upgrade to the full BigQuery experience. Learn more
Dismiss
Upgrade
Search BigQuery resources
Show starred only
alert-shape-478118-i3
Repositories
Queries
Notebooks
Data canvases
Data preparations
Pipelines
Connections
sales
Data
Untitled query
Run
Save
Download
Share
Schedule
Open in
More
--1. WHERE Clause
--Q1. Filter all transactions that occurred in the year 2023.
--Expected output: All columns
select * from `alert-shape-478118-i3.sales.Data` where EXTRACT(YEAR FROM Date) = 2023;
Query completed
Using on-demand processing quota
Query results
Save results
Open in
Job information
Results
Visualization
JSON
Execution details
Execution graph
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
Results per page: 50
1 – 50 of 998

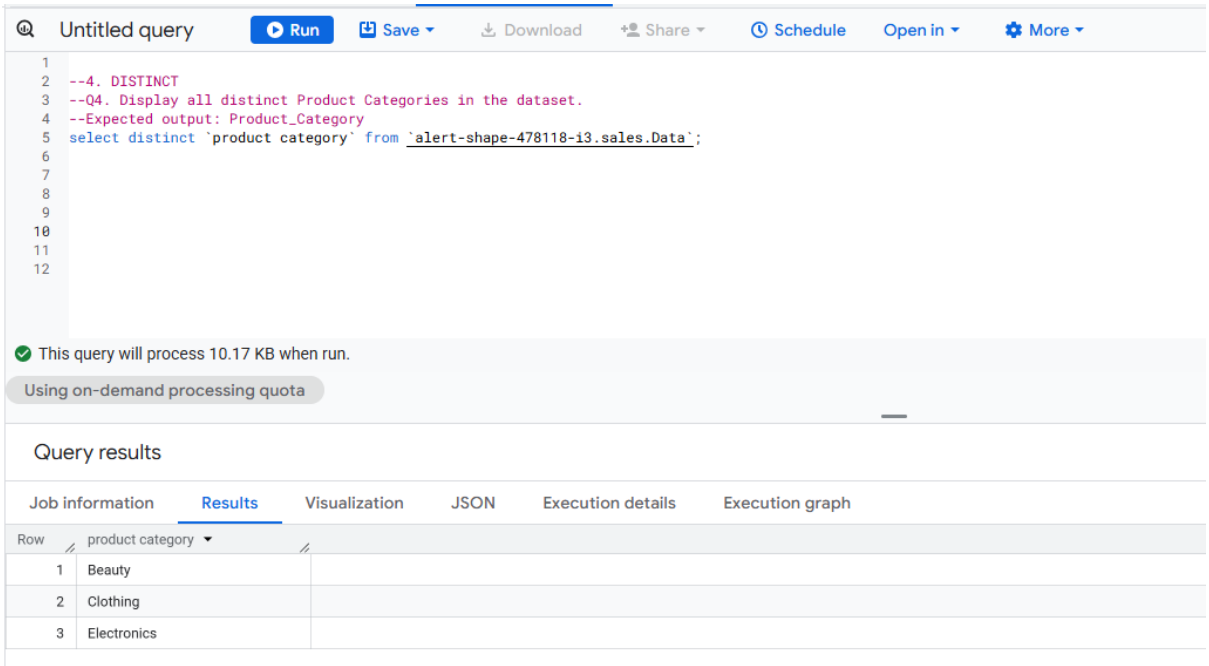
2


--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 `alert-shape-478118-i3.sales.Data`
where `Total Amount` >(SELECT AVG(`Total Amount`)
FROM `alert-shape-478118-i3.sales.Data`);
This query will process 72.69 KB when run.
Using on-demand processing quota
Query results
Save results
Open in
Job information
Results
Visualization
JSON
Execution details
Execution graph
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
7 | 408 | 2023-04-15 | CUST408 | Female | 64 | Beauty | 1 | 500 | 500
Results per page: 50
1 – 50 of 350

3


Untitled...ery
Data
*Untitled...ery
Untitled query
Run
Save
Download
Share
Schedule
Open in
More
--Q3. Calculate the total revenue (sum of Total Amount).
--Expected output: Total_Revenue
select sum(`Total amount`) as sum_of_Total_amount
from `alert-shape-478118-i3.sales.Data`;
Query completed
Using on-demand processing quota
Query results
Job information
Results
Visualization
JSON
Execution details
Execution graph
Row
sum_of_Total_am...
1
456000

4


Untitled query
Run
Save
Download
Share
Schedule
Open in
More
--4. DISTINCT
--Q4. Display all distinct Product Categories in the dataset.
--Expected output: Product_Category
select distinct `product category` from `alert-shape-478118-i3.sales.Data`;
This query will process 10.17 KB when run.
Using on-demand processing quota
Query results
Job information
Results
Visualization
JSON
Execution details
Execution graph
Row
product category
1
Beauty
2
Clothing
3
Electronics

5

Untitled query

Run | Save | Download | Share | Schedule | Open in | More

```sql
--5. GROUP BY
--Q5. For each Product Category, calculate the total quantity sold.
--Expected output: Product_Category, Total_Quantity
select `product category`, sum(quantity) as Total_Quantity
from `alert-shape-478118-i3.sales.Data`
group by `product category`;
```

This query will process 17.98 KB when run.

Using on-demand processing quota

Query results

Job information | Results | Visualization | JSON | Execution details | Execution graph

Row	product category	Total_Quantity
1	Beauty	771
2	Clothing	894
3	Electronics	849

6

Untitled query

Run | Save | Download | Share | Schedule | Open in | More

```sql
--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

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 `alert-shape-478118-i3.sales.Data`;
```

This query will process 16.6 KB when run.

Using on-demand processing quota

Query results

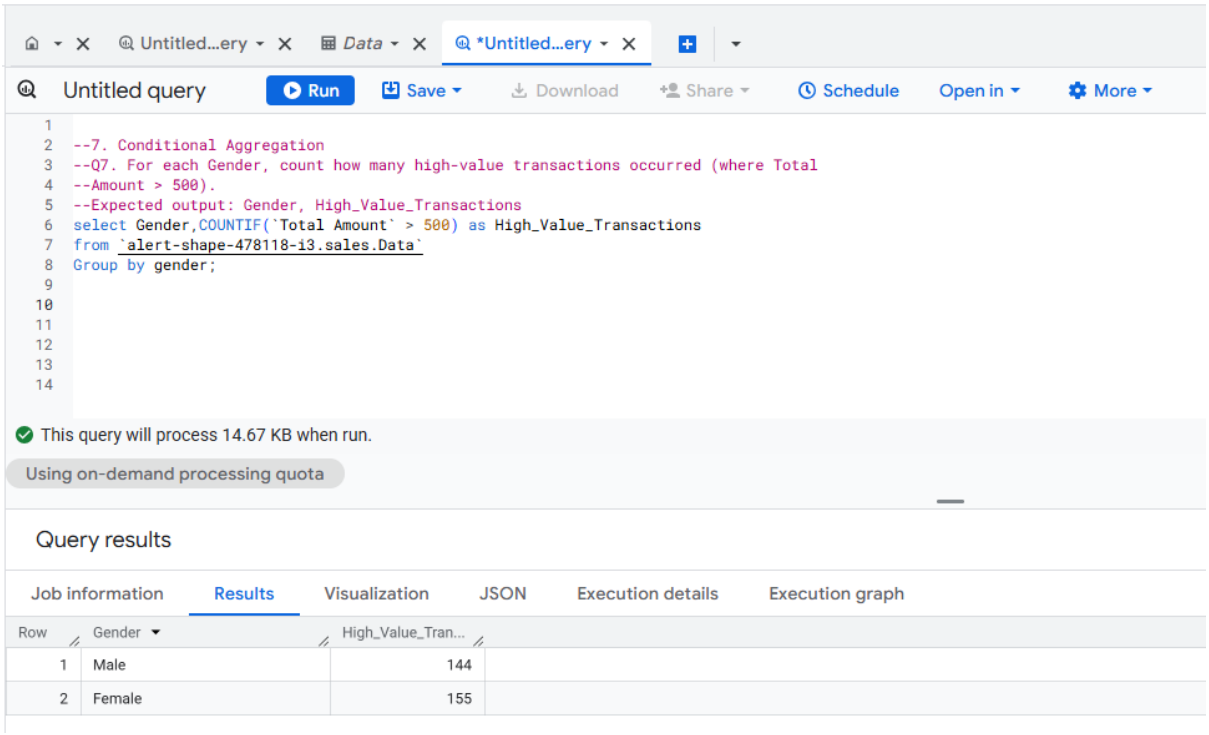
Save results | Open in

Job information | Results | Visualization | JSON | Execution details | Execution graph

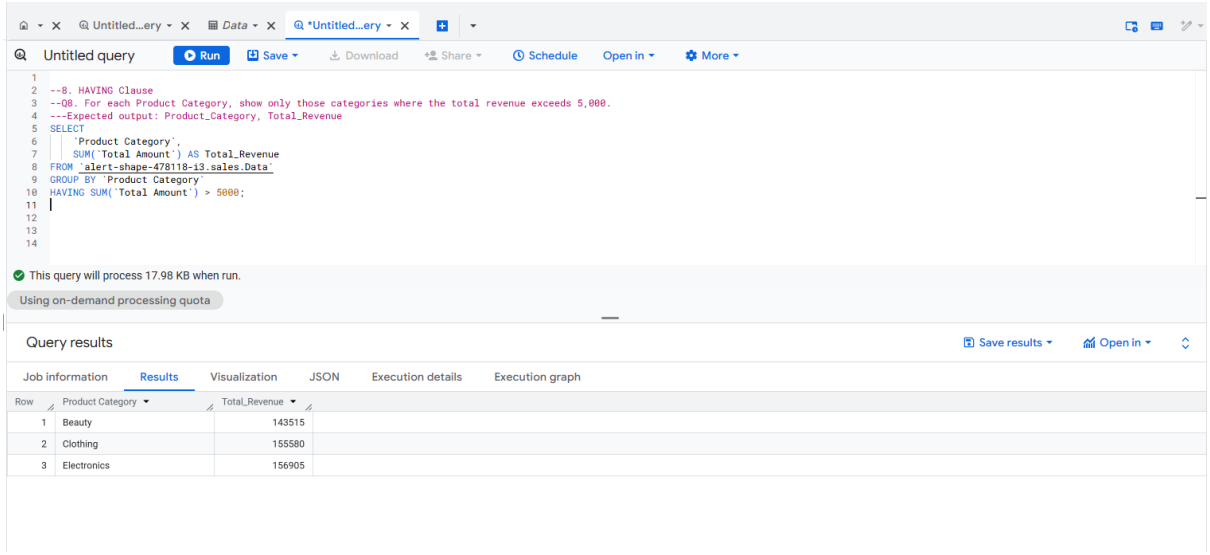
Row	customer id	age	Age_group
1	CUST191	64	senior
2	CUST204	39	Adult
3	CUST230	54	Adult
4	CUST232	43	Adult
5	CUST309	26	youth
6	CUST310	28	youth
7	CUST363	64	senior
8	CUST371	20	youth

Results per page: 50 | 1 – 50 of 1000

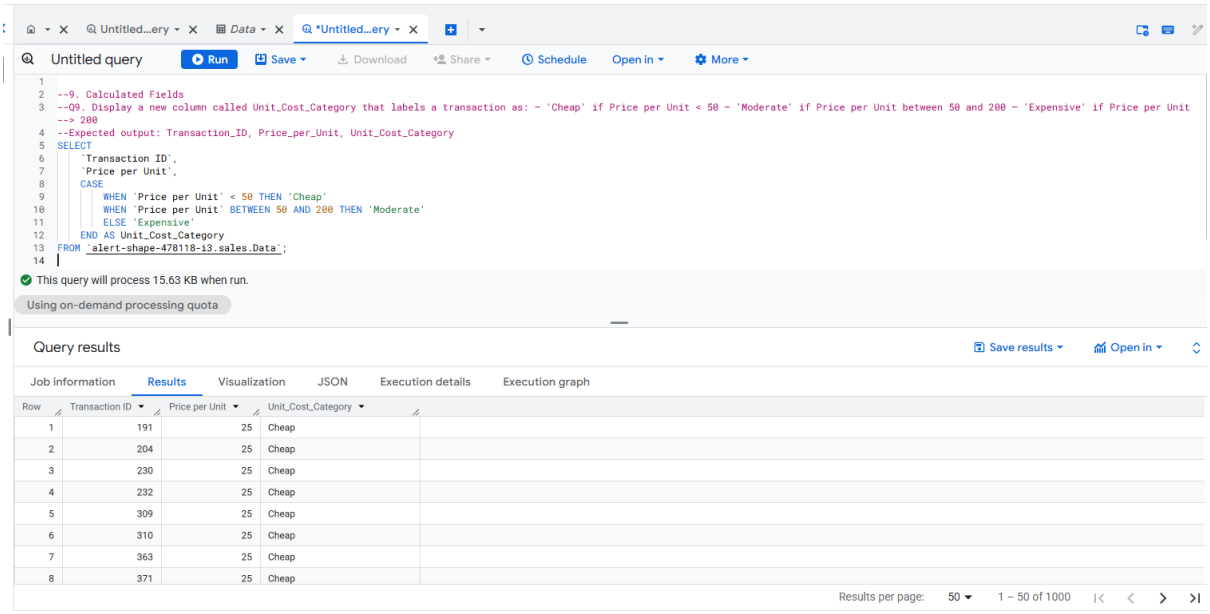
7


Untitled query
Run
Save
Download
Share
Schedule
Open in
More
--7. Conditional Aggregation
--Q7. For each Gender, count how many high-value transactions occurred (where Total
--Amount > 500).
--Expected output: Gender, High_Value_Transactions
select Gender,COUNTIF(`Total Amount` > 500) as High_Value_Transactions
from `alert-shape-478118-i3.sales.Data`
Group by gender;
This query will process 14.67 KB when run.
Using on-demand processing quota
Query results
Job information
Results
Visualization
JSON
Execution details
Execution graph
Row
Gender
High_Value_Tran...
1
Male
144
2
Female
155

8


Untitled query
Run
Save
Download
Share
Schedule
Open in
More
--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
SELECT
`Product Category`,
SUM(`Total Amount`) AS Total_Revenue
FROM `alert-shape-478118-i3.sales.Data`
GROUP BY `Product Category`
HAVING SUM(`Total Amount`) > 5000;
This query will process 17.98 KB when run.
Using on-demand processing quota
Query results
Save results
Open in
Job information
Results
Visualization
JSON
Execution details
Execution graph
Row
Product Category
Total_Revenue
1
Beauty
143515
2
Clothing
155580
3
Electronics
156905

9



```sql
--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
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 `alert-shape-478118-i3.sales.Data`;
```

This query will process 15.63 KB when run.

Using on-demand processing quota

Query results

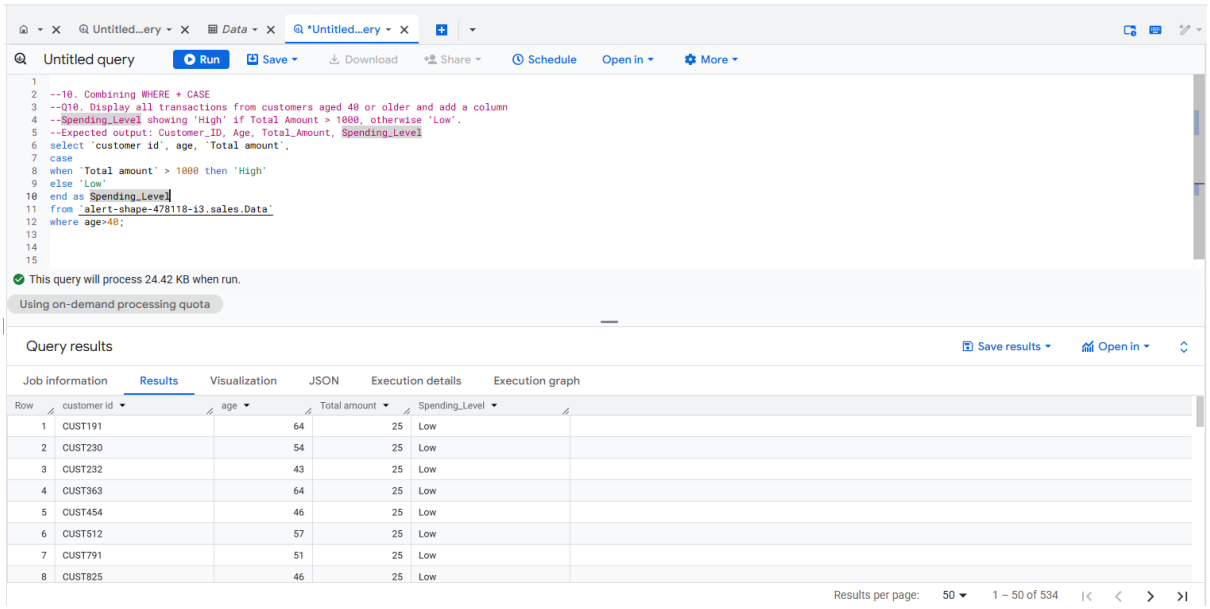
Save results | Open in

Job information | Results | Visualization | JSON | Execution details | Execution graph

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
5	309	25	Cheap
6	310	25	Cheap
7	363	25	Cheap
8	371	25	Cheap

Results per page: 50 | 1 – 50 of 1000

10



```sql
--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
select `customer id`, age, `Total amount`,
case
when `Total amount` > 1000 then 'High'
else 'Low'
end as Spending_Level
from `alert-shape-478118-i3.sales.Data`
where age>40;
```

This query will process 24.42 KB when run.

Using on-demand processing quota

Query results

Save results | Open in

Job information | Results | Visualization | JSON | Execution details | Execution graph

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
5	CUST454	46	25	Low
6	CUST512	57	25	Low
7	CUST791	51	25	Low
8	CUST825	46	25	Low

Results per page: 50 | 1 – 50 of 534