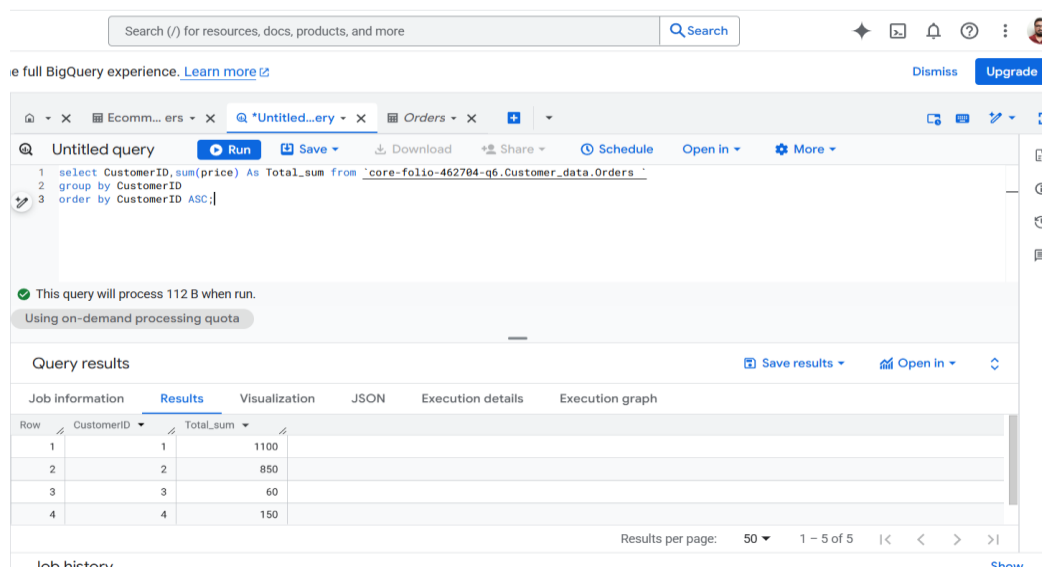


SQL INTERMEDIATE QUESTION ON DATASET

1) List each customer's **total spent** from the Orders table using SUM and arrange the CustomerID in Ascending order.

Query :- select CustomerID,sum(price) As Total_sum from `core-folio-462704-q6.Customer_data.Orders`
group by CustomerID
order by CustomerID ASC;



The screenshot shows the Google BigQuery interface. At the top, there is a search bar and navigation icons. Below the search bar, there is a tab for 'Untitled query'. The query editor shows the following SQL code:

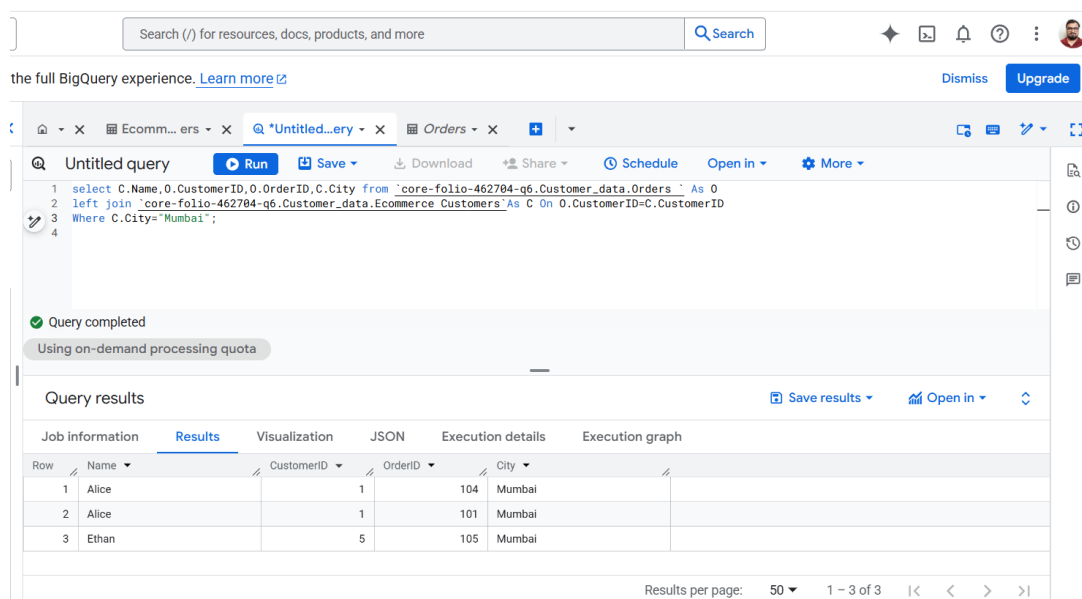
```
1 select CustomerID,sum(price) As Total_sum from `core-folio-462704-q6.Customer_data.Orders`  
2 group by CustomerID  
3 order by CustomerID ASC;
```

Below the query editor, there is a status bar indicating 'This query will process 112 B when run.' and 'Using on-demand processing quota'. The 'Query results' section is visible, showing a table with 4 rows and 2 columns: CustomerID and Total_sum.

| Row | CustomerID | Total_sum |
|-----|------------|-----------|
| 1 | 1 | 1100 |
| 2 | 2 | 850 |
| 3 | 3 | 60 |
| 4 | 4 | 150 |

2) Find all orders made by customers from **Mumbai**.

Query :- select C.Name,O.CustomerID,O.OrderID,C.City from `core-folio-462704-q6.Customer_data.Orders` As O
left join `core-folio-462704-q6.Customer_data.Ecommerce Customers`As C On
O.CustomerID=C.CustomerID
Where C.City="Mumbai";



The screenshot shows the Google BigQuery interface. At the top, there is a search bar and navigation icons. Below the search bar, there is a tab for 'Untitled query'. The query editor shows the following SQL code:

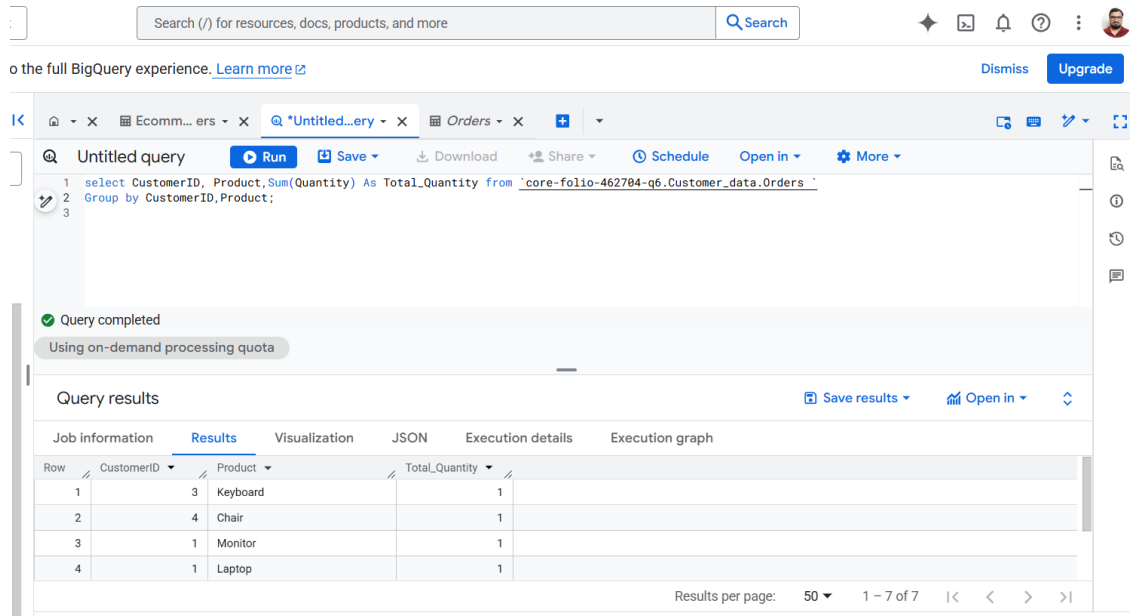
```
1 select C.Name,O.CustomerID,O.OrderID,C.City from `core-folio-462704-q6.Customer_data.Orders` As O  
2 left join `core-folio-462704-q6.Customer_data.Ecommerce Customers`As C On O.CustomerID=C.CustomerID  
3 Where C.City="Mumbai";  
4
```

Below the query editor, there is a status bar indicating 'Query completed' and 'Using on-demand processing quota'. The 'Query results' section is visible, showing a table with 3 rows and 5 columns: Name, CustomerID, OrderID, and City.

| Row | Name | CustomerID | OrderID | City |
|-----|-------|------------|---------|--------|
| 1 | Alice | 1 | 104 | Mumbai |
| 2 | Alice | 1 | 101 | Mumbai |
| 3 | Ethan | 5 | 105 | Mumbai |

3) Show the total quantity of products ordered per customer.

Query :- select CustomerID, Product,Sum(Quantity) As Total_Quantity from `core-folio-462704-q6.Customer_data.Orders`
Group by CustomerID,Product;



The screenshot shows the Google BigQuery web interface. At the top, there's a search bar and navigation icons. Below the header, the query editor displays the following SQL query:

```
1 select CustomerID, Product,Sum(Quantity) As Total_Quantity from `core-folio-462704-q6.Customer_data.Orders`  
2 Group by CustomerID,Product;  
3
```

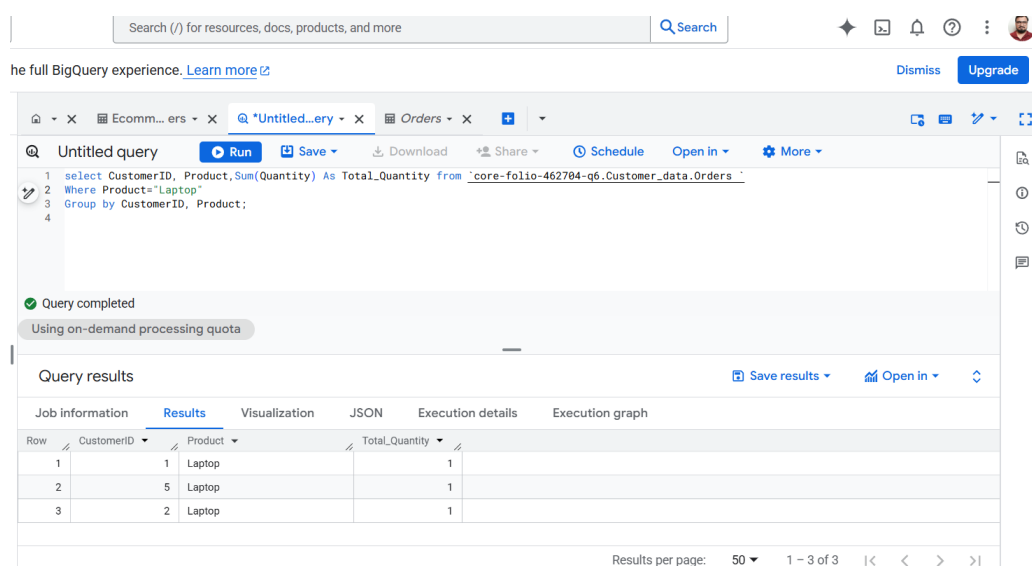
The query has been executed successfully, as indicated by the "Query completed" status. The results are displayed in a table with the following columns: Row, CustomerID, Product, and Total_Quantity.

| Row | CustomerID | Product | Total_Quantity |
|-----|------------|----------|----------------|
| 1 | 3 | Keyboard | 1 |
| 2 | 4 | Chair | 1 |
| 3 | 1 | Monitor | 1 |
| 4 | 1 | Laptop | 1 |

The interface also shows options to save results, open in a new tab, and view execution details.

4) Find customers who have purchased a **Laptop**.

Query : select CustomerID, Product,Sum(Quantity) As Total_Quantity from `core-folio-462704-q6.Customer_data.Orders`
Where Product="Laptop"
Group by CustomerID, Product;



The screenshot shows the Google BigQuery web interface with a filtered query. The SQL query in the editor is:

```
1 select CustomerID, Product,Sum(Quantity) As Total_Quantity from `core-folio-462704-q6.Customer_data.Orders`  
2 Where Product="Laptop"  
3 Group by CustomerID, Product;  
4
```

The query has been executed successfully. The results table shows only the rows where the product is a Laptop.

| Row | CustomerID | Product | Total_Quantity |
|-----|------------|---------|----------------|
| 1 | 1 | Laptop | 1 |
| 2 | 5 | Laptop | 1 |
| 3 | 2 | Laptop | 1 |

The interface also shows options to save results, open in a new tab, and view execution details.

5) Join Customers and Orders to list Customer Name, Product, Quantity, and OrderDate.

QUERY :- select C.Name,O.Product,O.Quantity,O.OrderDate from `core-folio-462704-q6.Customer_data.Orders` As O
left join `core-folio-462704-q6.Customer_data.Ecommerce Customers`As C On
C.CustomerID=O.CustomerID;

The screenshot displays the Google BigQuery web interface. At the top, there's a search bar and navigation icons. Below the header, a message prompts the user to 'to the full BigQuery experience' with a 'Learn more' link. The main workspace shows a query editor with the following SQL code:

```
1 select C.Name,O.Product,O.Quantity,O.OrderDate from `core-folio-462704-q6.Customer_data.Orders` As O
2 left join `core-folio-462704-q6.Customer_data.Ecommerce Customers`As C On C.CustomerID=O.CustomerID;
3
```

Below the editor, a status bar indicates 'Query completed' and 'Using on-demand processing quota'. The 'Query results' section is active, showing a table with 4 rows and 5 columns: Row, Name, Product, Quantity, and OrderDate.

| Row | Name | Product | Quantity | OrderDate |
|-----|---------|----------|----------|------------|
| 1 | Charlie | Keyboard | 1 | 2024-05-25 |
| 2 | Diana | Chair | 1 | 2024-03-01 |
| 3 | Alice | Monitor | 1 | 2024-03-10 |
| 4 | Alice | Laptop | 1 | 2024-02-01 |

At the bottom right of the results table, it says 'Results per page: 50' and '1 - 7 of 7'.