

SQL ADVANCE QUESTION ON DATASET

- 1) Find the top 2 customers by total spent in Orders.

```
Query:- select O.CustomerID, C.TotalSpent 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
        Order by C.TotalSpent DESC
        limit 2;
```

The screenshot shows the BigQuery web interface. At the top, there's a search bar and a navigation bar with tabs for 'Ecommerce' and 'Orders'. Below the tabs, an 'Untitled query' is selected. The query code is displayed:

```
1 select O.CustomerID, C.TotalSpent 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 Order by C.TotalSpent DESC
4 limit 2;
5
```

A note below the code says 'This query will process 136 B when run.' and 'Using on-demand processing quota'. The 'Results' tab is selected under 'Job information'. The results table has columns 'Row', 'CustomerID', and 'TotalSpent', with two rows of data:

Row	CustomerID	TotalSpent
1	4	1340
2	1	1200

- 2) Identify customers who have not ordered any product.

```
QUERY:- select C.CustomerID,C.Name from `core-folio-462704-q6.Customer_data.Ecommerce Customers` As C
          LEFT JOIN `core-folio-462704-q6.Customer_data.Orders` ` As O On
            C.CustomerID=O.CustomerID
          Where O.CustomerID Is Null;
```

NOTE:-There is no null value in both the table that's why the result showing there is no data to display.

The screenshot shows the BigQuery web interface. The 'Untitled query' is selected. The query code is:

```
1 select C.CustomerID,C.Name from `core-folio-462704-q6.Customer_data.Ecommerce Customers` As C
2 LEFT JOIN `core-folio-462704-q6.Customer_data.Orders` ` As O On C.CustomerID=O.CustomerID
3 Where O.CustomerID Is Null;
```

A note below the code says 'This query will process 131 B when run.' and 'Using on-demand processing quota'. The 'Results' tab is selected under 'Job information'. A message in a box says 'There is no data to display.' At the bottom, it shows 'Results per page: 50 ▾ 1 - 0 of 0 |< < > >|'

3) Calculate the **running total of price** ordered by each customer by OrderDate.

QUERY :- select CustomerID,Price,OrderDate,Sum(Price) Over(Partition By CustomerID Order By OrderDate)As Running_Total from `core-folio-462704-q6.Customer_data.Orders`;

The screenshot shows the BigQuery web interface. At the top, there is a search bar and a 'Search' button. Below the search bar, a message says 'full BigQuery experience. [Learn more](#)' with 'Dismiss' and 'Upgrade' buttons. The main area shows an 'Untitled query' tab with the following SQL code:

```
1 select CustomerID,Price,OrderDate,Sum(Price) Over(Partition By CustomerID Order By OrderDate )As Running_Total from `core-folio-462704-q6.Customer_data.Orders`;
```

A note below the code says 'This query will process 168 B when run.' and 'Using on-demand processing quota'. The results section shows a table with the following data:

Row	CustomerID	Price	OrderDate	Running_Total
3	2	50	2024-03-20	50
4	2	800	2024-04-05	850
5	3	60	2024-05-25	60
6	4	150	2024-03-01	150
7	5	800	2024-05-01	800

At the bottom of the results table, there are buttons for 'Save results' and 'Open in'.

4) Find the most expensive product ordered by each customer.

QUERY :- with ranked As (SELECT CustomerID,Product,Price,DENSE_RANK() Over(Partition By CustomerID ORDER BY Price DESC)As Expensive from `core-folio-462704-q6.Customer_data.Orders`)

Select CustomerID,Product,Price From Ranked

Where Expensive =1;

The screenshot shows the BigQuery web interface. At the top, there is a search bar and a 'Search' button. Below the search bar, a message says 'the full BigQuery experience. [Learn more](#)' with 'Dismiss' and 'Upgrade' buttons. The main area shows an 'Untitled query' tab with the following SQL code:

```
1 with ranked As (SELECT CustomerID,Product,Price,DENSE_RANK() Over(Partition By CustomerID ORDER BY Price DESC)As Expensive from `core-folio-462704-q6.Customer_data.Orders`)
2 Select CustomerID,Product,Price,Rank From Ranked
3 Where Rank =1;
4
```

A note below the code says 'Query completed' and 'Using on-demand processing quota'. The results section shows a table with the following data:

Row	CustomerID	Product	Price	Rank
1	1	Laptop	800	1
2	2	Laptop	800	1
3	3	Keyboard	60	1
4	4	Chair	150	1
5	5	Laptop	800	1

At the bottom of the results table, there are buttons for 'Save results' and 'Open in'.

5) List customers along with the number of distinct products they purchased.

```
QUERY :- Select C.Name,O.CustomerID,COUNT(DISTINCT O.Product) AS  
Distinct_Products_Purchased,CONCAT(COUNT(DISTINCT O.Product), '(',  
STRING_AGG(DISTINCT O.Product, ', '), ')') AS Product_List_Display 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  
Group By C.Name,O.CustomerID;
```

The screenshot shows the BigQuery web interface with the following details:

- Search Bar:** Search (/) for resources, docs, products, and more
- Toolbar:** Includes icons for Help, Search, Dismiss, Upgrade, and user profile.
- Query Editor:** Untitled query tab, showing the SQL code for the query.
- Query Status:** A green checkmark indicates the query will process 188 B when run, and a note about using on-demand processing quota.
- Results Tab:** Selected tab, showing the query results.
- Job Information:** Row, Name, CustomerID, Distinct_Products_Purchased, Product_List_Display.
- Results Data:** 5 rows of data:

Row	Name	CustomerID	Distinct_Products_Purchased	Product_List_Display
1	Charlie	3	1	1 (Keyboard)
2	Diana	4	1	1 (Chair)
3	Alice	1	2	2 (Monitor, Laptop)
4	Ethan	5	1	1 (Laptop)
5	Bob	2	2	2 (Laptop, Mouse)
- Page Controls:** Results per page: 50, 1 – 5 of 5, navigation arrows.