

BRIGHTLEARN

Practical exercise (Google BigQuery)

Dataset: Retail Sales Dataset

QUESTIONS

1. WHERE Clause

Q1. Filter all transactions that occurred in the year 2023.

Expected output: All columns

The screenshot shows the Google BigQuery Studio interface. On the left is a sidebar with navigation options: Overview, Studio, Pipelines and integration, Data transfers, Dataform, Scheduled queries, Scheduling, Governance, Sharing (Analytics Hub), Release notes, Partner Centre, and Settings. The main area displays a query editor with the following SQL code:

```
1 SELECT * FROM `retail-479317.Sales.data` LIMIT 1000;
2 -- 1. WHERE Clause Q1. Filter all transactions that occurred in the year 2023.
3 SELECT *
4 FROM `retail-479317.Sales.data`
5 WHERE EXTRACT(YEAR FROM DATE) = 2023;
```

Below the query editor, a message indicates "Query completed" and "Using on-demand processing quota". The "Query results" section is active, showing a table with the following data:

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

At the bottom right, it says "Results per page: 50" and "1 - 50 of 998".

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

The screenshot shows the Google BigQuery Studio interface. On the left is a sidebar with navigation options: Overview, Studio, Pipelines and integration, Data transfers, Dataform, Scheduled queries, Scheduling, Governance, Sharing (Analytics Hub), Release notes, Partner Centre, and Settings. The main area displays a query editor with the following SQL code:

```
9 SELECT
10 *
11 FROM
12 `retail-479317.Sales.data`
13 WHERE
14 `Total Amount` > (
15 SELECT
```

Below the query editor, a message indicates "Query completed" and "Using on-demand processing quota". The "Query results" section is active, showing a table with the following data:

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	21	2023-01-14	CUST021	Female	50	Beauty
2	28	2023-04-23	CUST028	Female	43	Beauty
3	128	2023-07-05	CUST128	Male	25	Beauty

At the bottom right, it says "Results per page: 50" and "1 - 50 of 350".

3. Aggregate Functions

Q3. Calculate the total revenue (sum of Total Amount).

Expected output: Total_Revenue

The screenshot shows the Google BigQuery Studio interface. On the left is a sidebar with navigation options: Overview, Studio, Pipelines and integration, Data transfers, Dataform, Scheduled queries, Scheduling, Governance, Sharing (Analytics Hub), Release notes, Partner Centre, Settings, and Release notes. The main area displays a query editor with the following SQL code:

```
11 FROM
12 `retail-479317`.`Sales.data`
13 WHERE
14 `Total Amount` > (
15   SELECT
16     AVG(`Total Amount`)
17   FROM
18     `retail-479317`.`Sales.data`
19 );
20
21 -- 3. Aggregate FunctionsQ3. Calculate the total revenue (sum of Total Amount).Expected output: Total_Revenue
22 SELECT SUM(`Total Amount`) AS Total_Revenue
23 FROM `retail-479317`.`Sales.data`;
```

Below the query editor, a status bar indicates "Query completed" and "Using on-demand processing quota". The "Query results" section is visible, showing a table with one column, "Total_Revenue", and one row with the value "456000".

4. DISTINCT

Q4. Display all distinct Product Categories in the dataset.

Expected output: Product_Category

The screenshot shows the Google BigQuery Studio interface. On the left is a sidebar with navigation options: Overview, Studio, Pipelines and integration, Data transfers, Dataform, Scheduled queries, Scheduling, Governance, Sharing (Analytics Hub), Release notes, Partner Centre, Settings, and Release notes. The main area displays a query editor with the following SQL code:

```
23 FROM `retail-479317`.`Sales.data`;
24
25 -- 4. DISTINCT Q4. Display all distinct Product Categories in the dataset.
26 SELECT DISTINCT `Product Category`
27 FROM `retail-479317`.`Sales.data`;
```

Below the query editor, a status bar indicates "Query completed" and "Using on-demand processing quota". The "Query results" section is visible, showing a table with one column, "Product Category", and three rows with the values "Beauty", "Clothing", and "Electronics".

5. GROUP BY

Q5. For each Product Category, calculate the total quantity sold.

Expected output: Product_Category, Total_Quantity

The screenshot shows the Google Cloud BigQuery Studio interface. On the left is a navigation sidebar with options like Overview, Studio, Pipelines and integration, Data transfers, Dataform, Scheduled queries, Scheduling, Governance, Sharing (Analytics Hub), Partner Centre, Settings, and Release notes. The main panel displays an 'Untitled query' with the following SQL code:

```

27 FROM `retail-479317.Sales_data`;
28
29 -- GROUP BY Q5. For each Product Category, calculate the total quantity sold.
30 SELECT
31   'Product Category',
32   SUM(Quantity) AS Total_Quantity
33 FROM `retail-479317.Sales_data`
34 GROUP BY 'Product Category';
35

```

Below the query editor, a status bar indicates 'Query completed' and 'Using on-demand processing quota'. The 'Query results' section is active, showing a table with 3 rows:

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

The bottom of the results section shows 'Results per page: 50' and '1 - 3 of 3'.

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

The screenshot shows the Google Cloud BigQuery Studio interface. The main panel displays an 'Untitled query' with the following SQL code:

```

37 SELECT
38   'Customer ID',
39   Age,
40   CASE
41     WHEN Age < 30 THEN 'Youth'
42     WHEN Age BETWEEN 30 AND 59 THEN 'Adult'
43     ELSE 'Senior'
44   END AS Age_Group
45 FROM `retail-479317.Sales_data`;
46

```

Below the query 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:

Row	Customer ID	Age	Age_Group
1	CUST191	64	Senior
2	CUST204	39	Adult
3	CUST230	54	Adult
4	CUST232	43	Adult

The bottom of the results section shows 'Results per page: 50' and '1 - 50 of 1000'.

7. Conditional Aggregation

Q7. For each Gender, count how many high-value transactions occurred (where Total Amount > 500).

Expected output: Gender, High_Value_Transactions

BigQuery Studio interface showing a query for Q7. The query is:

```

45 FROM `retail-479317.Sales_data`;
46
47 -- 7... Conditional Aggregation Q7. For each Gender, count how many high-value transactions occurred (where Total Amount > 500).
48 SELECT
49   Gender,
50   COUNTIF('Total Amount' > 500) AS High_Value_Transactions
51 FROM `retail-479317.Sales_data`
52 GROUP BY Gender;
53
54
55 Query completed
56 Using on-demand processing quota

```

The query results table is as follows:

Row	Gender	High_Value_Trans...
1	Male	144
2	Female	155

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

BigQuery Studio interface showing a query for Q8. The query is:

```

52 GROUP BY Gender;
53
54 -- 8. HAVING Clause Q8. For each Product Category, show only those categories where the total revenue exceeds 5,000
55 SELECT
56   'Product Category',
57   SUM('Total Amount') AS Total_Revenue
58 FROM `retail-479317.Sales_data`
59 GROUP BY 'Product Category'
60 HAVING SUM('Total Amount') > 5000;
61
62 Query completed
63 Using on-demand processing quota

```

The query results table is as follows:

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

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

BigQuery Studio interface showing a query execution result.

```

62 -- 9. Calculated Fields Q9. Display a new column called Unit_Cost_Category that labels a transaction as:
63 SELECT
64   'Transaction ID',
65   'Price per Unit',
66   CASE
67     WHEN 'Price per Unit' < 50 THEN 'Cheap'
68     WHEN 'Price per Unit' BETWEEN 50 AND 200 THEN 'Moderate'
69     ELSE 'Expensive'
70   END AS Unit_Cost_Category
71 FROM `retail-479317.Sales.data`;

```

Query completed. Using on-demand processing quota.

Query results table:

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

Results per page: 50 | 1 - 50 of 1000

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

BigQuery Studio interface showing a query execution result.

```

73 -- 10. Combining WHERE + CASE Q10. Display all transactions from customers
74 SELECT
75   'Customer ID',
76   Age,
77   'Total Amount',
78   CASE
79     WHEN 'Total Amount' > 1000 THEN 'High'
80     ELSE 'Low'
81   END AS Spending_Level
82 FROM `retail-479317.Sales.data`
83 WHERE Age >= 40;

```

Query completed. Using on-demand processing quota.

Query results table:

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

Results per page: 50 | 1 - 50 of 558