

BigQuery Exercise

The screenshot shows the Google Cloud BigQuery interface. On the left is a sidebar with project navigation and a search bar. The main area has a toolbar at the top with options like Run, Save, Download, Share, Schedule, and Open in. A message says "Query completed Using on-demand processing quota". Below is a "Query results" table with columns: Row, Transaction ID, Date, Customer ID, Gender, Age, and Product Categ. The data shows 5 rows of customer information. At the bottom, there's a "Job history" section and a system tray.

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Categ
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
4	232	2023-02-06	CUST232	Female	43	Beauty
5	309	2023-12-23	CUST309	Female	26	Beauty

Table

This screenshot is similar to the first one but shows a different query. The query is: `SELECT * FROM `Retail_Sales.2112` WHERE EXTRACT(YEAR FROM Date) = 2023;`. The results table is identical to the one above, showing 5 rows of customer data for the year 2023.

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Categ
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
4	232	2023-02-06	CUST232	Female	43	Beauty
5	309	2023-12-23	CUST309	Female	26	Beauty

Question 1

The screenshot shows the Google Cloud BigQuery interface. A query has been run on the 'Retail_Sales_2112' dataset:

```
18
19 SELECT *
20 FROM `Retail_Sales_2112`
21 WHERE `Total_Amount` > (
22     SELECT AVG(`Total_Amount`)
23     FROM `Retail_Sales_2112`)
```

The results show four rows of data:

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per
1	21	2023-01-14	CUST021	Female	50	Beauty	1	
2	28	2023-04-23	CUST028	Female	43	Beauty	1	
3	128	2023-07-05	CUST128	Male	25	Beauty	1	
4	220	2023-09-03	CUST220	Male	64	Beauty	1	

Job history is shown at the bottom.

Question 2

The screenshot shows the Google Cloud BigQuery interface. A query has been run on the 'Retail_Sales_2112' dataset:

```
--Q3 : Aggregate Functions
--Q3 . Calculate the total revenue (sum of Total_Amount).
---Expected output: Total_Revenue
31
32 SELECT COUNT(`Total_Amount`) AS Total_Revenue
33 FROM `Retail_Sales_2112`
```

The results show one row of data:

Row	Total_Revenue
1	1000

Using on-demand processing quota is indicated. Job history is shown at the bottom.

Question 3

The screenshot shows the Google Cloud BigQuery interface. A query has been run on the 'Retail_Sales.2112' dataset. The results show three distinct product categories: Beauty, Clothing, and Electronics.

```
37 ----4. DISTINCT
38 ----4. Display all distinct Product Categories in the dataset.
39 -----Expected output: Product_Category
40
41 SELECT DISTINCT `Product Category`
42   FROM `Retail_Sales.2112`
43
44
45
46
47
48
49
50
51
52
```

Row	Product Category
1	Beauty
2	Clothing
3	Electronics

Question 4

The screenshot shows the Google Cloud BigQuery interface. A query has been run on the 'Retail_Sales.2112' dataset. The results show the total quantity sold for each product category.

```
47 -----5. For each Product Category, calculate the total quantity sold.
48 -----Expected output: Product_Category
49
50
51
52
```

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

Question 5

The screenshot shows the Google Cloud BigQuery interface. A query has been run to create an Age_Group column based on customer age. The results show three rows of data: CUST191 (Age 64, null for Age Group), CUST204 (Age 39, Adult), and CUST230 (Age 54, Adult). The interface includes a toolbar at the top with various icons, a sidebar on the left, and a bottom navigation bar.

```
59 ---Q6. Create a column called Age_Group that classifies customers as 'Youth' (<30), 'Adult'
60 (30->59), and [Senior] (60+).
61 ----Expected output: Customer_ID, Age, Age_Group.
62
63 SELECT `Customer_ID`, Age,
64   CASE WHEN Age BETWEEN 16 AND 30 THEN 'Youth'
65   WHEN Age BETWEEN 31 AND 59 THEN 'Adult'
66   ELSE 'Senior'
67 END AS Age_Group
68
69
70
71
72
73
74
75
76
77
78
79
80
```

Row	Customer ID	Age	Age Group
1	CUST191	64	null
2	CUST204	39	Adult
3	CUST230	54	Adult

Question 6

The screenshot shows the Google Cloud BigQuery interface. A query has been run to count high-value transactions for each gender. The results show two rows: Male (144 transactions) and Female (155 transactions). The interface includes a toolbar at the top with various icons, a sidebar on the left, and a bottom navigation bar.

```
73 ----Q7. For each Gender, count how many high-value transactions occurred (where Total
74 Amount > 500).
75 -----Expected output: Gender, High_Value_Transactions
76
77
78
79
80
```

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

Question 7

A screenshot of the Google Cloud BigQuery interface. The query window contains the following SQL code:

```
90  
91 SELECT  
92   'Product Category',  
93   SUM('Total Amount') AS Total_Revenue  
94 FROM `Retail_Sales.2112`
```

The results table shows the total revenue for three product categories:

Product Category	Total_Revenue
Beauty	143515
Clothing	155580
Electronics	156905

Job history and system status information are visible at the bottom.

Question 8

A screenshot of the Google Cloud BigQuery interface. The query window contains the following SQL code:

```
107 SELECT 'Transaction ID', 'Price per Unit',  
108   CASE  
109     WHEN 'Price per Unit' < 50 THEN 'Cheap'  
110     WHEN 'Price per Unit' BETWEEN 50 AND 200 THEN 'Moderate'  
111   END AS 'Unit Cost Category'
```

The results table shows transaction IDs and their unit cost categories based on price per unit:

Transaction ID	Price per Unit	Unit Cost Category
1	191	Cheap
2	204	Cheap
3	230	Cheap
4	232	Cheap

Job history and system status information are visible at the bottom.

Question 9

The screenshot shows the Google Cloud BigQuery interface. A modal window titled "Untitled query" is open, displaying the following SQL code:

```
122 SELECT 'Customer ID',  
123      'Age',  
124      'Total Amount',  
125      CASE  
126      WHEN 'Total Amount' > 1000 THEN 'High'  
127      ELSE 'Low'  
128 END AS Spending Level
```

The status bar at the bottom indicates "Query completed". Below the modal, the "Query results" section is visible, showing a table with three rows of data:| Row | Customer ID | Age | Total Amount | Spending Level |
| --- | --- | --- | --- | --- |
| 1 | CUST191 | 64 | 25 | Low |
| 2 | CUST230 | 54 | 25 | Low |
| 3 | CUST232 | 43 | 25 | Low |

Question 10