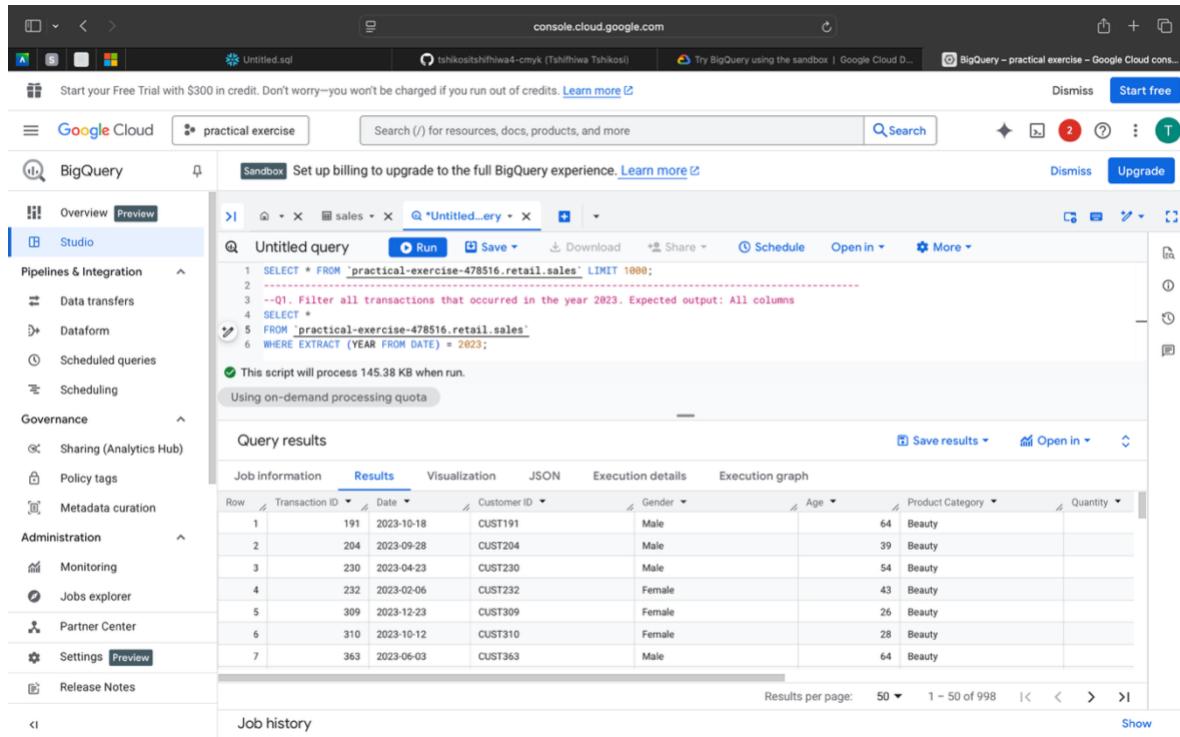


BIG QUERY PRACTICAL EXERCISE – TSHIKOSI TSHIFHIWA

QUESTION 1



The screenshot shows the Google Cloud BigQuery interface. On the left, there's a sidebar with various navigation options like Overview, Studio, Pipelines & Integration, Data transfers, Dataform, Scheduled queries, Scheduling, Governance, Policy tags, Metadata curation, Administration, Monitoring, Jobs explorer, Partner Center, Settings, and Release Notes. The main area is titled "Untitled query" and contains the following SQL code:

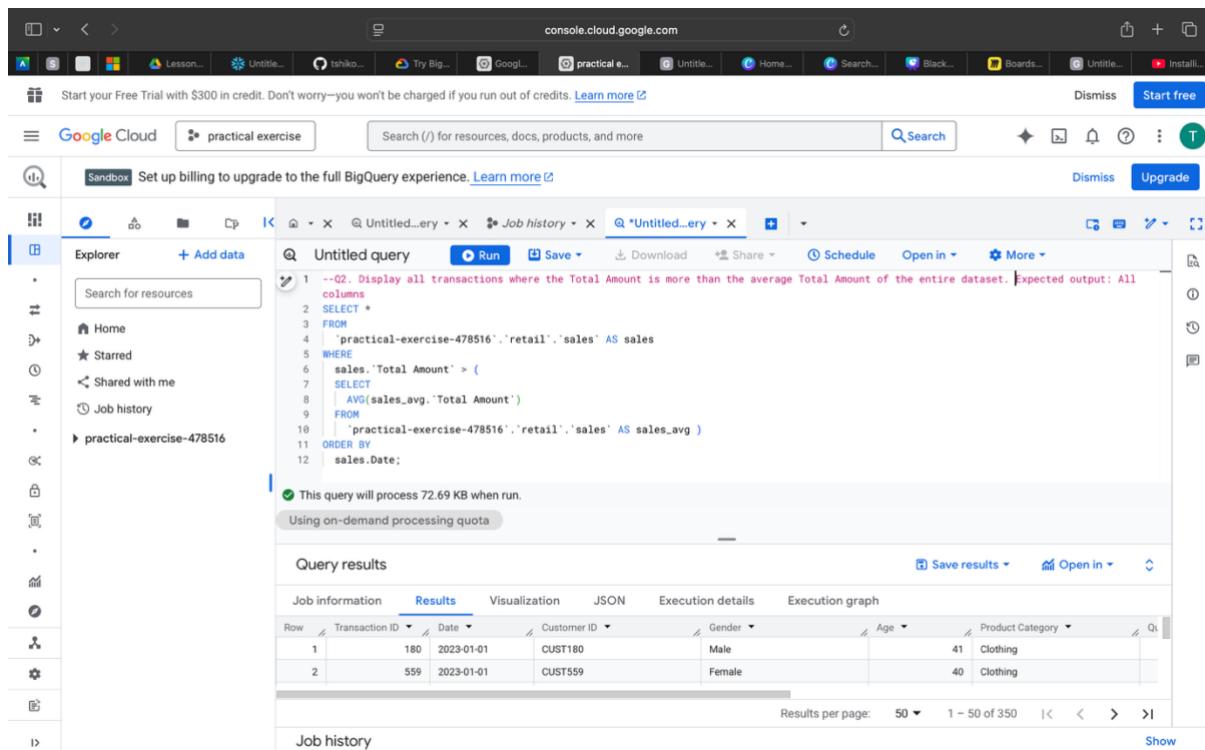
```
1 SELECT * FROM `practical-exercise-478516.retail.sales` LIMIT 1000;
2
3 --Q1. Filter all transactions that occurred in the year 2023. Expected output: All columns
4
5 SELECT *
6 FROM `practical-exercise-478516.retail.sales`
7 WHERE EXTRACT (YEAR FROM DATE) = 2023;
```

A note below the code says, "This script will process 145.38 KB when run." The "Results" tab is selected in the "Query results" section, displaying a table with 7 rows of data:

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity
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	
6	310	2023-10-12	CUST310	Female	28	Beauty	
7	363	2023-06-03	CUST363	Male	64	Beauty	

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

QUESTION 2



The screenshot shows the Google Cloud BigQuery interface. On the left, there's a sidebar with various navigation options like Explorer, Home, Starred, Shared with me, Job history, and practical-exercise-478516. The main area is titled "Untitled query" and contains the following SQL code:

```
1 --Q2. Display all transactions where the Total Amount is more than the average Total Amount of the entire dataset. Expected output: All columns
2
3 SELECT *
4 FROM `practical-exercise-478516`.`retail`.`sales` AS sales
5 WHERE
6 sales.Total Amount' > (
7 SELECT
8 AVG(sales_avg.`Total Amount`)
9 FROM `practical-exercise-478516`.`retail`.`sales` AS sales_avg
10 ORDER BY
11 sales.Date;
```

A note below the code says, "This query will process 72.69 KB when run." The "Results" tab is selected in the "Query results" section, displaying a table with 2 rows of data:

Row	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity
1	180	2023-01-01	CUST180	Male	41	Clothing	
2	559	2023-01-01	CUST559	Female	40	Clothing	

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

QUESTION 3

The screenshot shows the Google Cloud BigQuery interface. On the left, there's a sidebar with navigation links like Home, Starred, Shared with me, Job history, and the current project practical-exercise-478516. The main area has tabs for Explorer, Untitled query, Job history, and a search bar. The Untitled query tab is active, showing a code editor with the following SQL query:

```
12 sales.Date;
13 -----
14 --Q3. Calculate the total revenue (sum of Total_Amount). Expected output: Total_Revenue
15 SELECT SUM(sales.'Total_Amount') AS TOTAL_REVENUE
16 FROM 'practical-exercise-478516'.'retail'.'sales';
17
18 This script will process 80.5 KB when run.
```

Below the code editor is a "Using on-demand processing quota" message. The "Results" tab is selected under the "Query results" section. It shows a single row of data:

Row	TOTAL_REVENUE
1	456000

At the bottom, there are buttons for Save results and Open in, and a results per page dropdown set to 50.

QUESTION 4

This screenshot shows the same Google Cloud BigQuery interface as the previous one. The sidebar and tabs are identical. The Untitled query tab is active, displaying the following SQL query:

```
18
19 --Q4. Display all distinct Product Categories in the dataset. Expected output: Product_Category
20 SELECT DISTINCT sales.'product category'
21
22 This script will process 90.67 KB when run.
```

Below the code editor is a "Using on-demand processing quota" message. The "Results" tab is selected under the "Query results" section. It shows three rows of data:

Row	product category
1	Beauty
2	Clothing
3	Electronics

At the bottom, there are buttons for Save results and Open in, and a results per page dropdown set to 50.

QUESTION 5

The screenshot shows the Google Cloud BigQuery interface. In the center, there is a query editor window titled "Untitled query". The code input field contains the following SQL query:

```
--Q5. For each Product Category, calculate the total quantity sold. Expected output: Product_Category, Total_Quantity
SELECT sales.'product category',
       SUM(sales.quantity) AS TOTAL_QUANTITY SOLD
  FROM `practical-exercise-478516`.`retail`.`sales`
 GROUP BY sales.'product category';
```

Below the code, a note says: "This script will process 108.65 KB when run." and "Using on-demand processing quota".

On the right side of the interface, there is a "Query results" section. It has tabs for "Job information", "Results", "Visualization", "JSON", "Execution details", and "Execution graph". The "Results" tab is selected, showing a table with three rows:

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

At the bottom of the results table, it says "Results per page: 50 1 - 3 of 3".

QUESTION 6

The screenshot shows the Google Cloud BigQuery interface. In the center, there is a query editor window titled "Untitled query". The code input field contains the following SQL query:

```
--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
  sales.'customer id',
  sales.age,
  CASE
    WHEN sales.age < 30 THEN 'Youth'
    WHEN sales.age BETWEEN 30 AND 59 THEN 'Adult'
    WHEN sales.age >= 60 THEN 'Senior'
  END AS Age_Group
FROM `practical-exercise-478516`.`retail.sales`;
```

Below the code, a note says: "This query will process 16.6 KB when run." and "Using on-demand processing quota".

On the right side of the interface, there is a "Query results" section. It has tabs for "Job information", "Results", "Visualization", "JSON", "Execution details", and "Execution graph". The "Results" tab is selected, showing a table with four rows:

Row	customer id	age	Age_Group
1	CUST191	64	Senior
2	CUST204	39	Adult
3	CUST230	54	Adult
4	CUST232	43	Adult

At the bottom of the results table, it says "Results per page: 50 1 - 50 of 1000".

QUESTION 7

The screenshot shows the Google Cloud BigQuery interface. The query is:

```
--Q7. For each Gender, count how many high-value transactions occurred (where Total Amount > 500). Expected output: Gender, High_Value_Transactions
SELECT sales.gender,
       COUNT(*) AS high_value_transactions
  FROM `practical-exercise-478516`.retail.sales
 WHERE sales.total_amount > 500
 GROUP BY sales.gender;
```

The results table shows:

gender	high_value_trans...
Female	155
Male	144

QUESTION 8

The screenshot shows the Google Cloud BigQuery interface. The query is:

```
--Q8. For each Product Category, show only those categories where the total revenue exceeds 5,000. Expected output: Product_Category, Total_Revenue
SELECT sales.Product_Cat..., SUM(sales.total_amount) AS total_revenue
  FROM `practical-exercise-478516`.retail.sales
 GROUP BY sales.Product_Cat...
 HAVING SUM(sales.total_amount)>5000;
```

The results table shows:

Product Category	total_revenue
Beauty	143515
Clothing	155580
Electronics	156905

QUESTION 9

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section with a search bar and a tree view of projects and datasets, including 'practical-exercise-478516'. The main area is titled 'Untitled query' with a code editor containing the following SQL:

```
--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 sales.'Transaction ID',
       sales.'Price per Unit',
       CASE
         WHEN sales.'Price per Unit' < 50 THEN 'Cheap'
         WHEN sales.'Price per Unit' BETWEEN 50 AND 200 THEN 'Moderate'
         WHEN sales.'Price per Unit' > 200 THEN 'Expensive'
         ELSE 'Unknown'
       END AS Unit_Cost_Category
FROM `practical-exercise-478516`.`retail`.`sales` AS sales;
```

A note at the bottom says, "This script will process 64.88 KB when run." Below the code editor is a 'Using on-demand processing quota' message. The 'Results' tab is selected in the 'Query results' section, showing a table with four rows:

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

At the bottom right of the results table, it says "Results per page: 50 1 - 50 of 1000".

QUESTION 10

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section with a search bar and a tree view of projects and datasets, including 'practical-exercise-478516'. The main area is titled 'Untitled query' with a code editor containing the following SQL:

```
otherwise 'Low'. Expected output: Customer_ID, Age, Total_Amount, Spending_Level
SELECT sales.'Customer ID',
       sales.'Age',
       sales.'Total Amount',
       CASE
         WHEN sales.'Total Amount' > 1000 THEN 'High'
         ELSE 'Low'
       END AS Spending_Level
FROM `practical-exercise-478516`.`retail`.`sales`
WHERE sales.'Age' >= 40
ORDER BY sales.'Customer ID';
```

A note at the bottom says, "This script will process 89.29 KB when run." Below the code editor is a 'Using on-demand processing quota' message. The 'Results' tab is selected in the 'Query results' section, showing a table with three rows:

Row	Customer ID	Age	Total Amount	Spending_Level
1	CUST003	50	30	Low
2	CUST006	45	30	Low
3	CUST007	46	50	Low

At the bottom right of the results table, it says "Results per page: 50 1 - 50 of 558".