

✅ Task 1: Explore the Data

◆ Goal:

Explore the `thelook_ecommerce` dataset, focusing on the `orders` and `order_items` tables.

🧭 Steps:

1. Open **BigQuery** from the Navigation menu.
2. Expand your project and locate the **thelook_ecommerce** dataset.
3. Explore the **orders** table:
 - **Schema tab**: Review column names and types.
 - **Details tab**: Check table metadata.
 - **Preview tab**: View the first 1000 rows.

✅ Correct Answers: Which column names are used in the orders table?

- ✅ **order_id**
 - ✅ **status**
 - customer_name ❌ (*Not present in the schema*)
 - quantity ❌ (*Belongs to order_items, not orders*)
-

✅ Correct Answers: Benefits of using the Preview tab

- Create a simple report ❌ (*Preview is not for reporting*)
 - ✅ **Verify data has been loaded correctly**
 - Identify relationships in the data ❌ (*Requires querying or joins*)
 - ✅ **Inspect table contents without running a query**
-

✅ Task 2: Find the 10 Most Recent Orders

◆ Goal:

Calculate **recency** — how long ago each customer made their last purchase.

🧠 SQL Query:

```
SQL
SELECT
  user_id AS customer_id,
```

```
DATE_DIFF(CURRENT_TIMESTAMP(), MAX(created_at), DAY) AS recency
FROM `thelook_ecommerce.orders`
GROUP BY user_id
ORDER BY recency DESC
LIMIT 10;
```

✅ Correct Outcome:

- Returns 10 customers with the **oldest last purchase dates** (i.e., highest recency values).
 - Helps identify **at-risk customers** who haven't purchased in a long time.
-

✅ Task 3: Determine the Order Frequency

◆ Goal:

Calculate **frequency** — total number of orders placed by each customer in 2022.

🧠 SQL Query:

```
SQL
SELECT
  user_id AS customer_id,
  COUNT(order_id) AS frequency
FROM `thelook_ecommerce.orders`
WHERE created_at >= '2022-01-01' AND created_at < '2023-01-01'
GROUP BY customer_id
ORDER BY frequency DESC
LIMIT 10;
```

✅ Correct Answer: What is the highest order frequency over the past year?

- ✅ 10
Correct: Sorting the results in descending order reveals the top frequency value.