

DATA EXPLORATION OF OLIST BRAZILIAN E-COMMERCE USING SQL & BIGQUERY

FATIMA EZZAHRA KABBA

DATA ANALYST | RESEARCH ANALYST | BUSINESS INTELLIGENCE | FREELANCE CONSULTANT



EMAIL: FATIMA.EZZAHRA.KABBA.PRO@GMAIL.COM



LINKEDIN: [LINKEDIN.COM/IN/DR-FATIMA-KABBA](https://www.linkedin.com/in/dr-fatima-kabba)



GITHUB: [GITHUB.COM/FATIMAKABBA](https://github.com/FATIMAKABBA)

PROJECT OVERVIEW

A STRUCTURED DATA EXPLORATION OF OLIST'S E-COMMERCE DATASET USING
SQL & BIGQUERY



What is the Olist Dataset?

- **Brazilian e-commerce transactions** from a marketplace platform.
- Contains **multiple relational tables** (orders, products, customers, sellers).



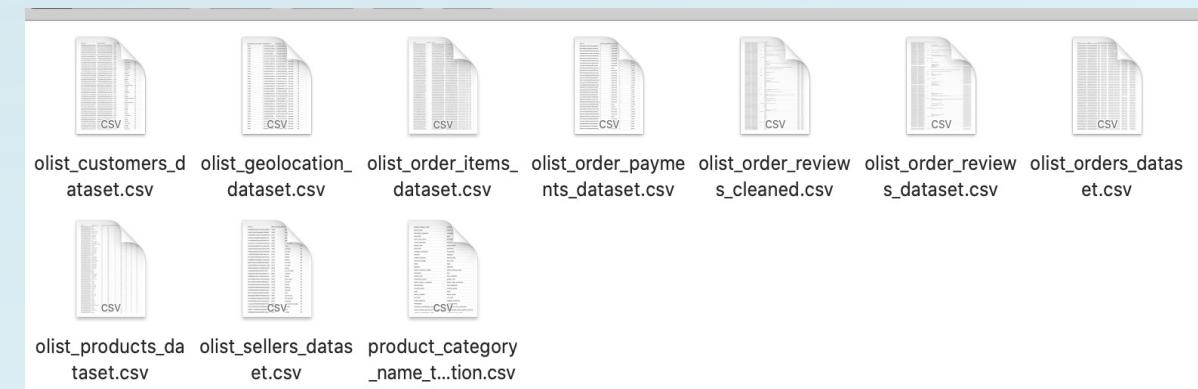
Why This Dataset?

- ✓ **Real-world business data** – Used for e-commerce insights.
- ✓ **Great for SQL & data cleaning** – Perfect for **data exploration & transformation**.



Key project objectives:

- 🔍 **Understand data structure & relationships.**
- 🔗 **Ensure data integrity & quality** before analysis.
- ⚠️ **Detect potential data issues (missing values, duplicates, errors).**



OLIST DATASET OVERVIEW: UNDERSTANDING THE DATA STRUCTURE

Key Details About the Dataset

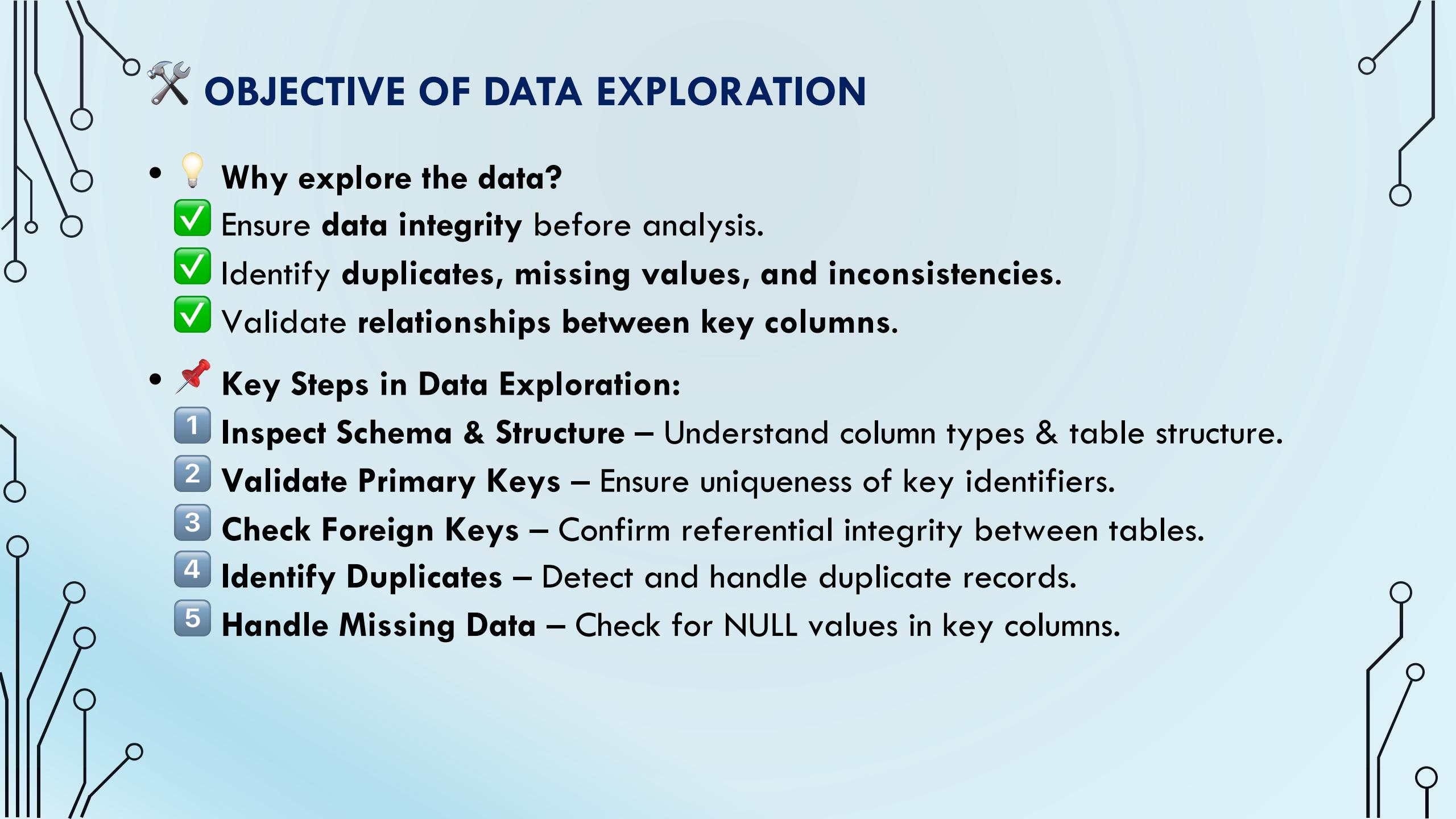
-  **Total Number of Tables:** 9 (covering various aspects of e-commerce).
-  **Project Focus:** This project explores the orders table, ensuring data integrity before further analysis.
-  **Dataset Context:** The Olist dataset represents real-world **Brazilian e-commerce transactions**, containing multiple relational tables.

Key Table Used in This Project

 **olist_orders_dataset** → Contains customer order details, timestamps, and transaction statuses.

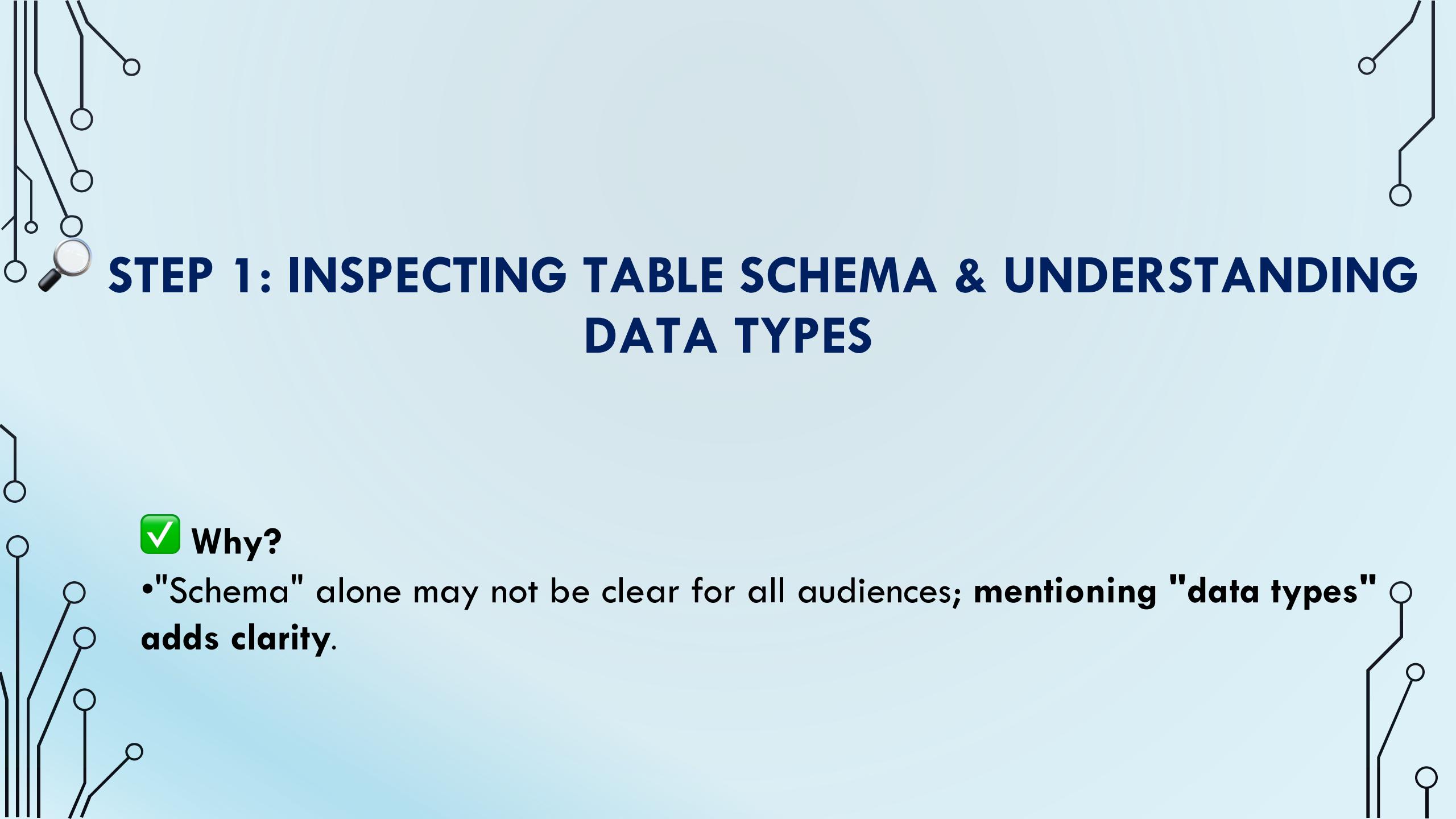
 Other tables such as *products*, *customers*, and *sellers* provide additional details but were not explored in this project.

DATA EXPLORATION – OBJECTIVE & STEPS



OBJECTIVE OF DATA EXPLORATION

-  **Why explore the data?**
 - ✓ Ensure **data integrity** before analysis.
 - ✓ Identify **duplicates, missing values, and inconsistencies**.
 - ✓ Validate **relationships between key columns**.
-  **Key Steps in Data Exploration:**
 - 1 **Inspect Schema & Structure** – Understand column types & table structure.
 - 2 **Validate Primary Keys** – Ensure uniqueness of key identifiers.
 - 3 **Check Foreign Keys** – Confirm referential integrity between tables.
 - 4 **Identify Duplicates** – Detect and handle duplicate records.
 - 5 **Handle Missing Data** – Check for NULL values in key columns.



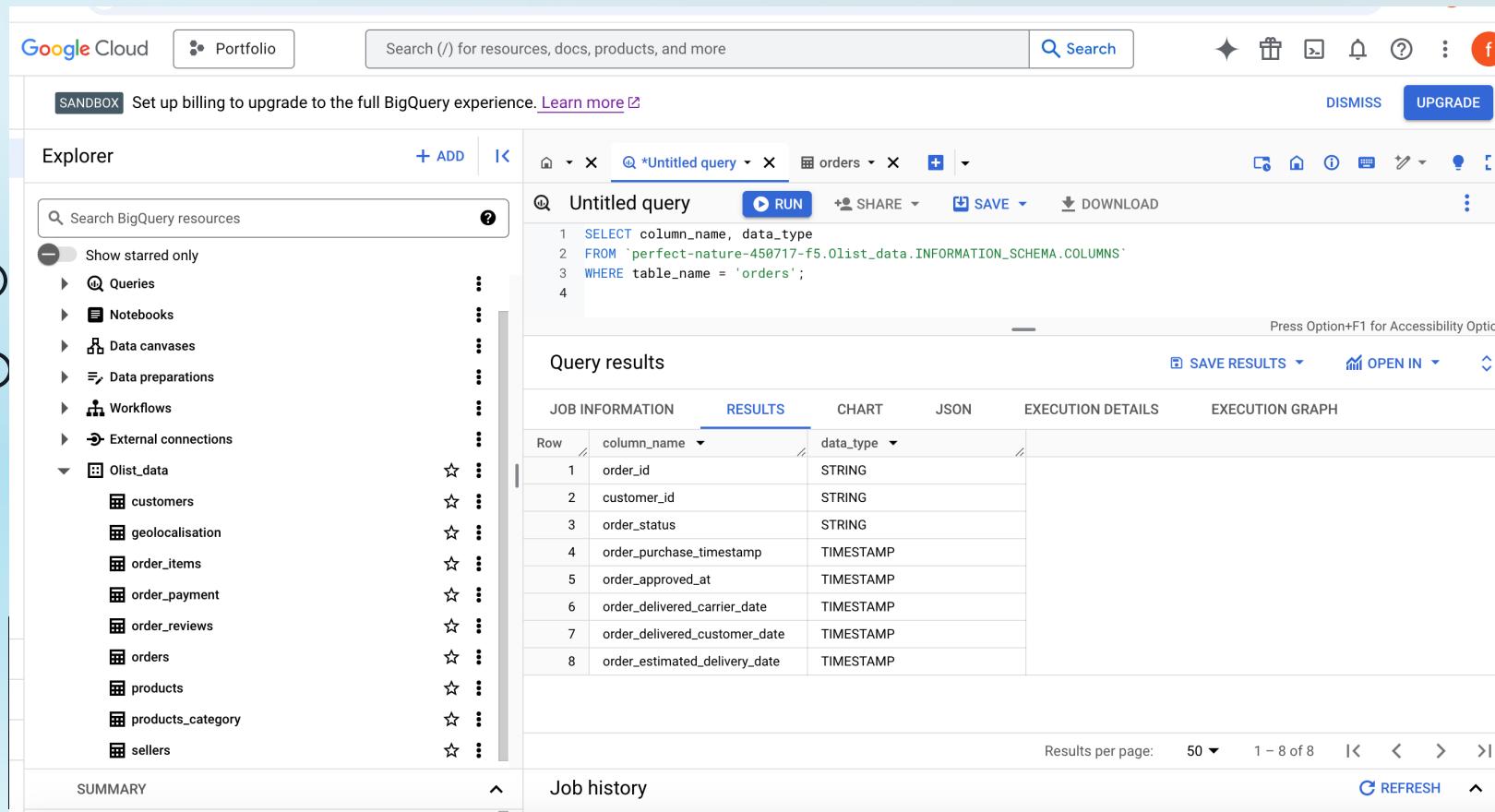
STEP 1: INSPECTING TABLE SCHEMA & UNDERSTANDING DATA TYPES



- "Schema" alone may not be clear for all audiences; **mentioning "data types" adds clarity.**

 **Objective:** Identify column names, data types, and table structure for better data understanding.

```
SELECT column_name, data_type
FROM `perfect-nature-450717-f5.olist_data.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'orders';
```



The screenshot shows the Google Cloud BigQuery interface. The top navigation bar includes 'Google Cloud', 'Portfolio', a search bar, and various icons. A 'Sandbox' message with a 'Learn more' link and a 'DISMISS' button is visible. The main area has an 'Explorer' sidebar on the left containing a tree view of datasets and tables, including 'Olist_data' and its sub-tables like 'customers', 'geolocalisation', and 'orders'. The central workspace shows an 'Untitled query' tab with the previously provided SQL code. Below it, the 'Query results' section displays the results of the query in a table format. The table has two columns: 'column_name' and 'data_type'. The data is as follows:

Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

 The *orders* table is a key dataset in the Olist e-commerce database.

This step inspects its schema to identify column names, data types, and potential data quality issues before analysis.”

Key Findings: Schema & Structure Overview

 Total Columns: 8

 Primary Identifiers:

- order_id, customer_id (**Both are STRING data types**)

 Timestamps Available:

- order_purchase_timestamp
- order_approved_at
- order_delivered_carrier_date (and more)

 Potential Data Cleaning Needs:

- Check for **NULL values** in date fields.



Step 2: Identifying Primary Keys in the orders Table

✓ Objective:

- Verify if `order_id` is a **Primary Key** (i.e., each `order_id` is unique).
- Ensure there are **no duplicate orders** in the dataset.
- Confirm `order_id` can be used for **joins with other tables**.

Objectif: Verify if order_id is a Primary Key (i.e., each order_id

is unique).

```
SELECT COUNT(order_id) AS total_orders,  
       COUNT(DISTINCT order_id) AS unique_orders  
  FROM `perfect-nature-450717-  
f5.olist_data.orders`;
```

The screenshot shows the Google Cloud BigQuery interface. The left sidebar displays the project structure under 'perfect-nature-450717-f5'. The main area is titled 'Untitled query' and contains the following SQL code:

```
1 SELECT COUNT(order_id) AS total_orders,  
2      COUNT(DISTINCT order_id) AS unique_orders  
3  FROM `perfect-nature-450717-f5.olist_data.orders`;
```

The 'RESULTS' tab is selected, showing the query results in a table:

Row	total_orders	unique_orders
1	99441	99441

Below the results, there are tabs for 'JOB INFORMATION', 'CHART', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH'. At the bottom, there are buttons for 'SAVE RESULTS', 'OPEN IN', and 'REFRESH'.

🏆 Key Findings (Bullet Points):

- ✓ Total Orders: 99,441
- ✓ Unique Orders: 99,441
- ✓ Primary Key Status: ✓ Valid (No duplicate order IDs found)

📌 This confirms that `order_id` is unique and can be used as a primary key for further analysis.

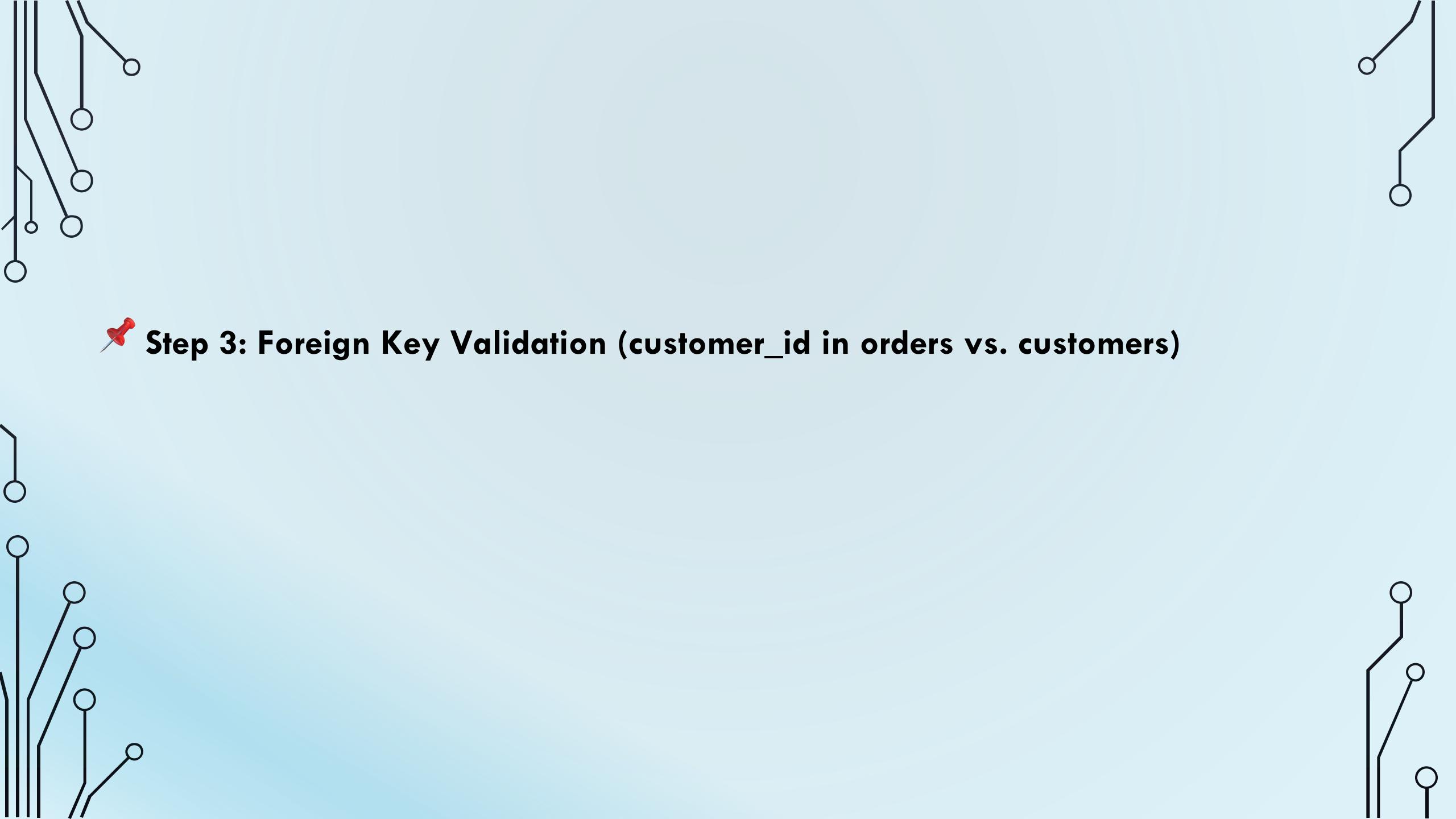
The screenshot shows the Google Cloud BigQuery interface. The left sidebar displays the project structure under 'perfect-nature-450717-f5' with 'Olist_data' selected, which contains tables like 'customers', 'geolocalisation', 'order_items', etc. The main area shows an 'Untitled query' with the following SQL code:

```
1 SELECT COUNT(order_id) AS total_orders,
2      COUNT(DISTINCT order_id) AS unique_orders
3 FROM `perfect-nature-450717-f5.olist_data.orders`;
```

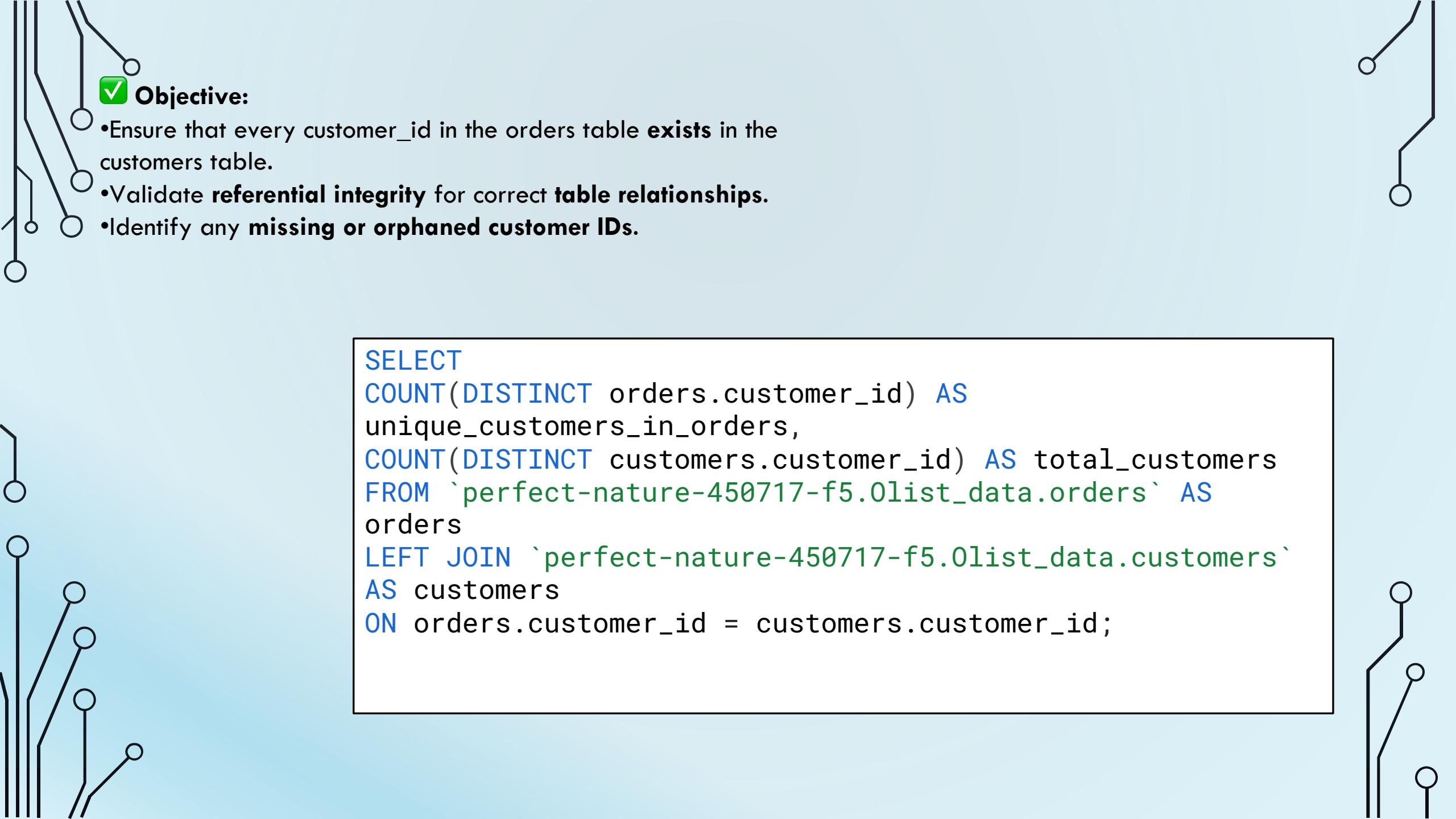
The 'Query results' section shows the output of the query:

Row	total_orders	unique_orders
1	99441	99441

Below the results, there are tabs for 'JOB INFORMATION', 'RESULTS' (which is selected), 'CHART', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH'. At the bottom, there are navigation links for 'SUMMARY' and 'Job history'.



📌 **Step 3: Foreign Key Validation (`customer_id` in `orders` vs. `customers`)**



Objective:

- Ensure that every `customer_id` in the `orders` table **exists** in the `customers` table.
- Validate **referential integrity** for correct **table relationships**.
- Identify any **missing or orphaned customer IDs**.

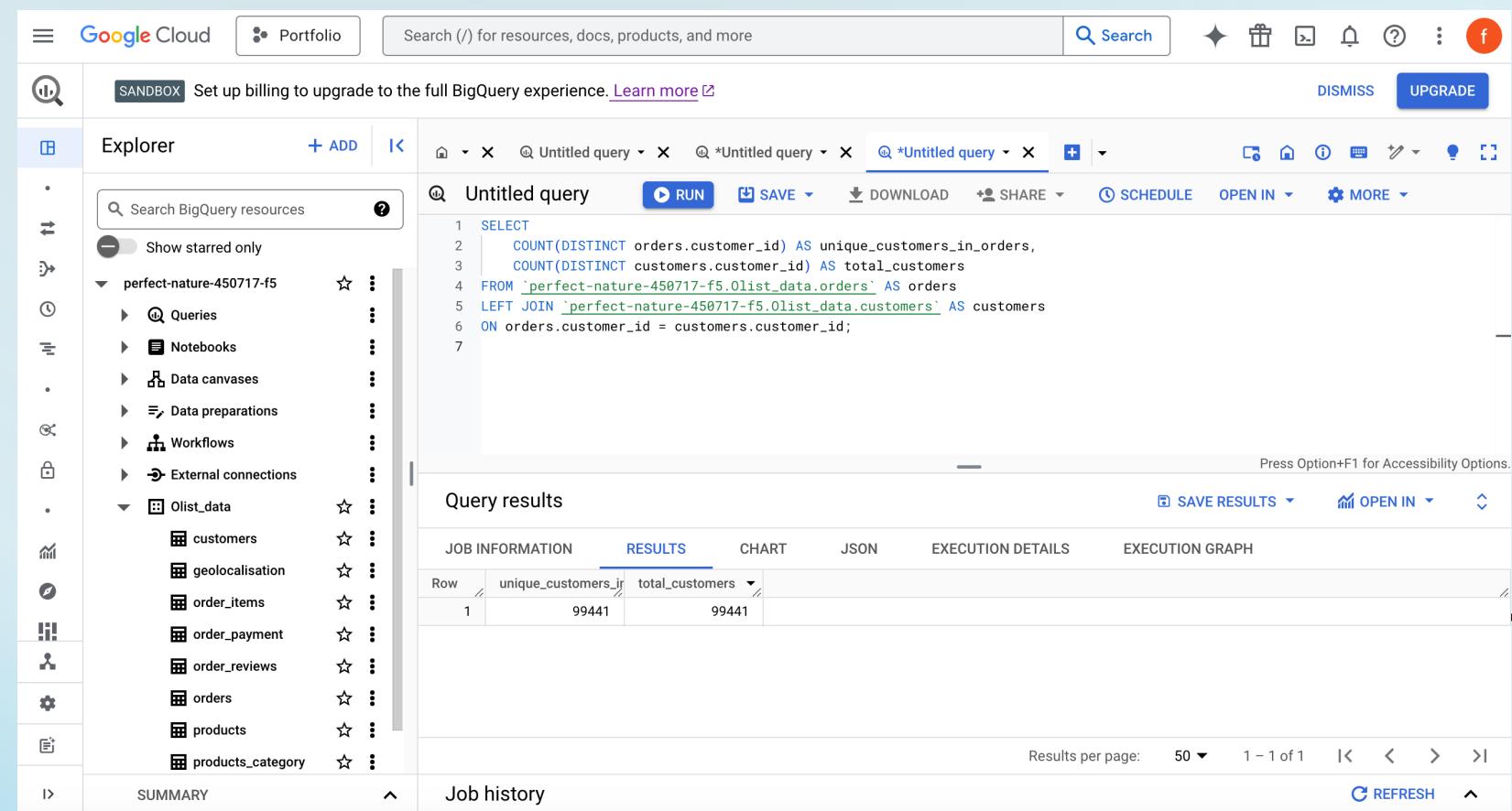
```
SELECT
  COUNT(DISTINCT orders.customer_id) AS
  unique_customers_in_orders,
  COUNT(DISTINCT customers.customer_id) AS total_customers
FROM `perfect-nature-450717-f5.olist_data.orders` AS
  orders
  LEFT JOIN `perfect-nature-450717-f5.olist_data.customers` AS
  customers
  ON orders.customer_id = customers.customer_id;
```

 Unique Customers in Orders: 99,441

 Total Customers in Dataset: 99,441

 Foreign Key Status:  Valid (No missing customer_id values)

 This confirms that every customer_id in the orders table exists in the customers table, ensuring accurate relationships.



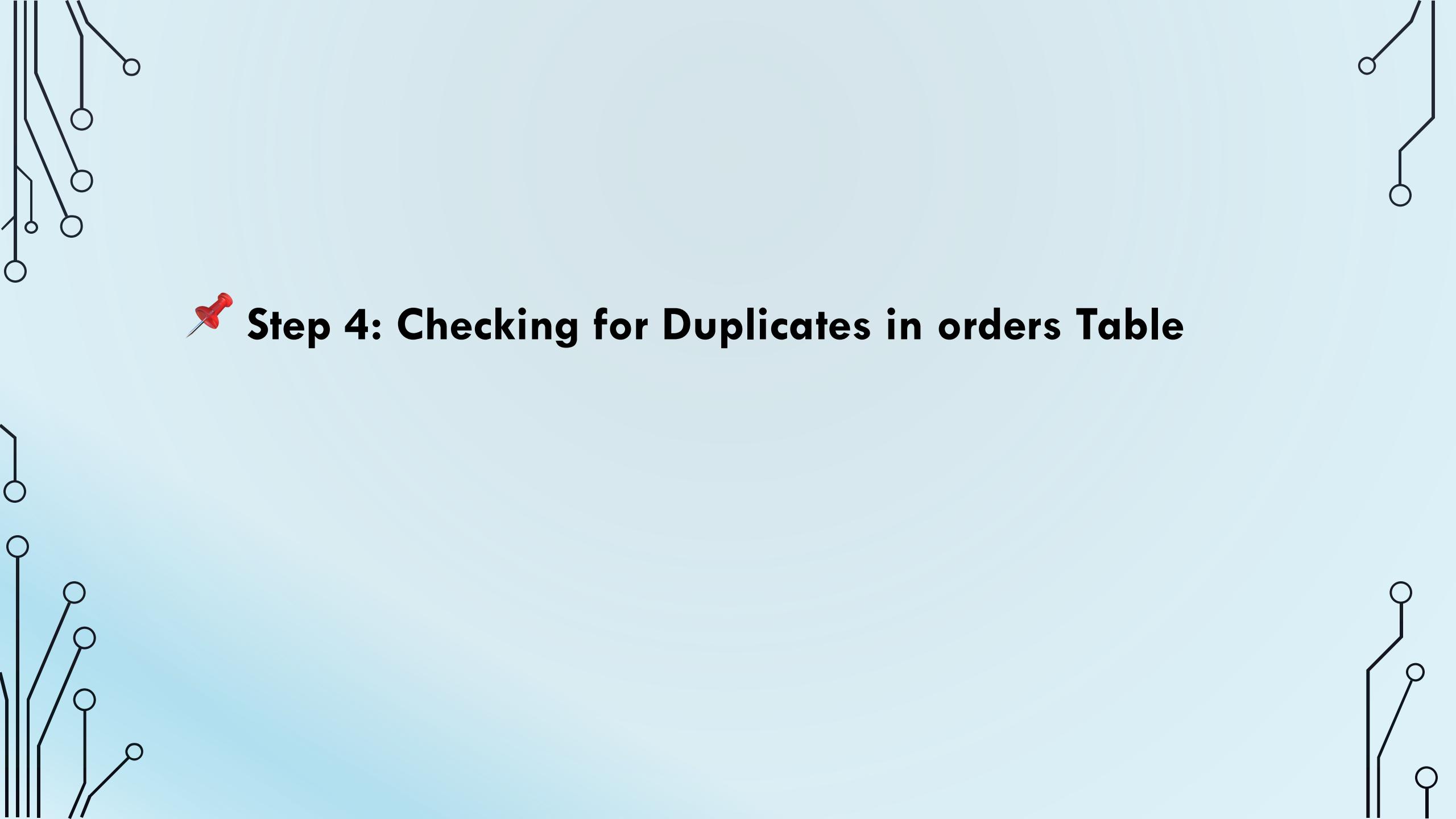
The screenshot shows the Google Cloud BigQuery interface. On the left is the Explorer sidebar with project and dataset navigation. The main area is titled "Untitled query" and contains the following SQL code:

```
1 SELECT
2     COUNT(DISTINCT orders.customer_id) AS unique_customers_in_orders,
3     COUNT(DISTINCT customers.customer_id) AS total_customers
4 FROM `perfect-nature-450717-f5.olist_data.orders` AS orders
5 LEFT JOIN `perfect-nature-450717-f5.olist_data.customers` AS customers
6 ON orders.customer_id = customers.customer_id;
```

The "Query results" section displays the output of the query:

Row	unique_customers_in_orders	total_customers
1	99441	99441

At the bottom, there are tabs for "JOB INFORMATION", "RESULTS" (which is selected), "CHART", "JSON", "EXECUTION DETAILS", and "EXECUTION GRAPH".



Step 4: Checking for Duplicates in orders Table

 **Objective** The goal of this step is to ensure that each `order_id` in the `orders` table is unique and there are no duplicate records. Duplicate orders can lead to incorrect analysis, inflated sales numbers, and misleading insights. Detecting and removing them is crucial for maintaining data integrity.

```
SELECT order_id, COUNT(*) AS order_count  
FROM `perfect-nature-450717-f5.olist_data.orders`  
GROUP BY order_id  
HAVING COUNT(*) > 1;
```

This step verifies whether duplicate `order_id` values exist in the `orders` table. Duplicates can cause inaccurate sales calculations, reporting errors, and inconsistencies in data analysis.



Key Findings:

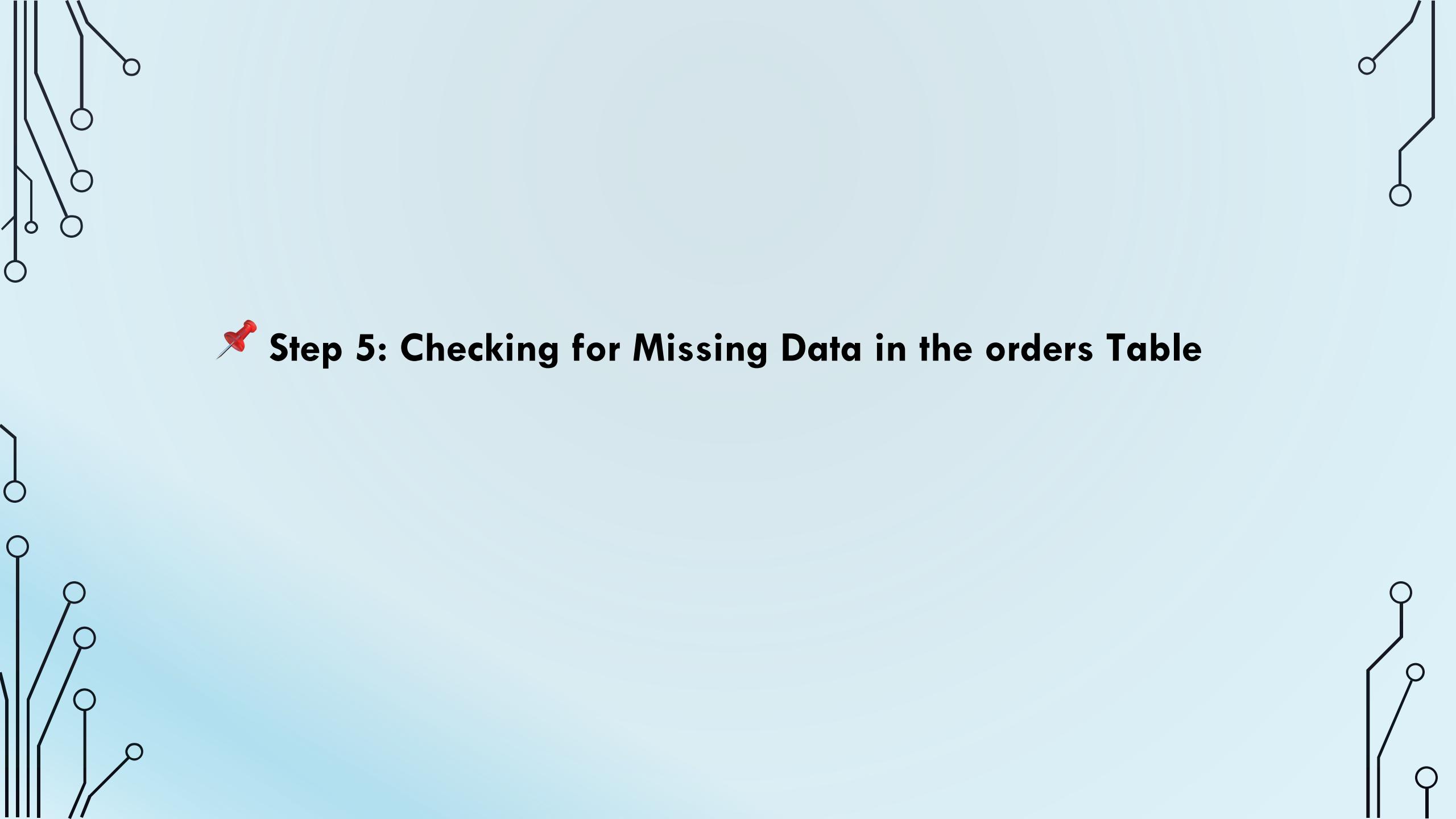
- ✓ No duplicate order_id values found (Query returned zero results).
- ✓ The order_id column is **unique**, confirming its validity as a **Primary Key**.
- ✓ No data cleaning is required for duplicate orders.

This ensures that each order is recorded only once, preventing potential double counting in further analysis.

The screenshot shows the Google Cloud BigQuery interface. On the left is the sidebar with project navigation and resource types like Queries, Notebooks, Data canvases, etc. The main area has a header bar with tabs for Portfolio, Search, and various icons. Below the header is the 'Sandbox' message: 'Set up billing to upgrade to the full BigQuery experience.' A blue button for 'UPGRADE' is visible. The central part of the interface is the 'Untitled query' editor, which contains the following SQL code:

```
1 SELECT order_id, COUNT(*) AS order_count
2 FROM `perfect-nature-450717-f5.olist_data.orders`
3 GROUP BY order_id
4 HAVING COUNT(*) > 1;
```

Below the query editor is the 'Query results' section, which displays the message: 'There is no data to display.' At the bottom of the interface, there are tabs for 'JOB INFORMATION', 'RESULTS' (which is selected), 'CHART', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH'. The status bar at the bottom right shows 'Results per page: 50 ▾ 1 - 0 of 0 ▾ REFRESH ▾'.



📌 Step 5: Checking for Missing Data in the orders Table



Objective:

"The goal of this step is to identify missing (NULL) values in key columns. Missing data can impact analysis, lead to incomplete reports, and cause errors in business decision-making. Detecting and handling them ensures accurate insights

```
SELECT
    COUNTIF(order_id IS NULL) AS missing_order_id,
    COUNTIF(customer_id IS NULL) AS missing_customer_id,
    COUNTIF(order_status IS NULL) AS missing_order_status,
    COUNTIF(order_purchase_timestamp IS NULL) AS
        missing_order_purchase_date
FROM `perfect-nature-450717-f5.olist_data.orders`;
```



Key Findings

- ✓ No missing values detected in key columns.
- ✓ Data is complete, ensuring accuracy for further analysis.
- ✓ No immediate need for data cleaning related to missing values.

📌 This confirms that all essential fields (`order_id`, `customer_id`, `order_status`, `order_purchase_timestamp`) are properly recorded, maintaining data integrity.

The screenshot shows the Google Cloud BigQuery interface. On the left is the Explorer sidebar with a tree view of datasets and tables. The main area is titled "Untitled query" and contains the following SQL code:

```
1 SELECT
2   COUNTIF(order_id IS NULL) AS missing_order_id,
3   COUNTIF(customer_id IS NULL) AS missing_customer_id,
4   COUNTIF(order_status IS NULL) AS missing_order_status,
5   COUNTIF(order_purchase_timestamp IS NULL) AS missing_order_purchase_date
6 FROM `perfect-nature-450717-f5.olist_data.orders`;
```

Below the code, the "Query results" section displays the output of the query:

Row	missing_order_id	missing_customer_id	missing_order_status	missing_order_purchase_date
1	0	0	0	0

At the bottom, there are tabs for "SUMMARY" and "Job history".

SUMMARY INSIGHTS

Key Insights from Data Exploration

🔍 This data exploration process validated the integrity, completeness, and structure of the orders table in the Olist dataset. The results ensure that the dataset is clean, well-structured, and ready for deeper analysis or visualization.

🏆 Key Findings:

✓ **Schema Inspection:** The orders table contains **8 key columns**, including order_id, customer_id, and multiple timestamps.

✓ **Primary Key Check:** order_id is **unique and valid**, confirming its role as the **primary key**.

✓ **Foreign Key Validation:** customer_id exists in both orders and customers tables, ensuring **referential integrity**.

✓ **Duplicate Check:** **No duplicate order IDs found**, confirming **data consistency**.

✓ **Missing Data Check:** **No missing values** in key columns, ensuring **data completeness**.

📌 These results confirm that the dataset is reliable for further analysis, such as customer behavior insights, order fulfillment trends, and sales analysis.

🔍 Step	⌚ Key Findings	✅ Insights
1 Schema Inspection	The orders table contains 8 columns , including order_id, customer_id, order_status, and multiple timestamps.	The dataset is structured well, making it ready for further analysis.
2 Primary Key Validation	order_id is unique (99,441 total orders = 99,441 unique orders).	✅ order_id is a valid Primary Key , ensuring integrity.
3 Foreign Key Validation	Every customer_id in orders exists in customers (99,441 matches 99,441).	✅ No missing customer records , ensuring relational integrity.
4 Duplicate Check	No duplicate orders found in the dataset.	✅ Data is clean & reliable , preventing double-counting issues.
5 Missing Data Check	No missing values in key columns (order_id, customer_id, order_status, order_purchase_timestamp).	✅ Complete dataset , no need for data imputation.

CONCLUSION

This project showcases my ability to conduct structured SQL-based data exploration using BigQuery. The clean dataset can now be leveraged for further analysis, such as customer segmentation, sales trends, and dashboard visualization

LET'S CONNECT & COLLABORATE! 🚀

Looking for a data analyst who can transform raw data into actionable insights?

Let's connect!



Ways to Reach Me:

- **LinkedIn:** <https://www.linkedin.com/in/dr-fatima-kabba/>
- **GitHub:** <https://github.com/FatimaKabba>
- **Tableau:** <https://public.tableau.com/app/profile/kabba.fatima.ezzahra/vizzes>
- **Email:** fatima.ezzahra.kabba.pro@gmail.com

*I'm open to opportunities in **data analysis, business intelligence, and freelancing projects.***

Let's discuss how I can add value to your team!