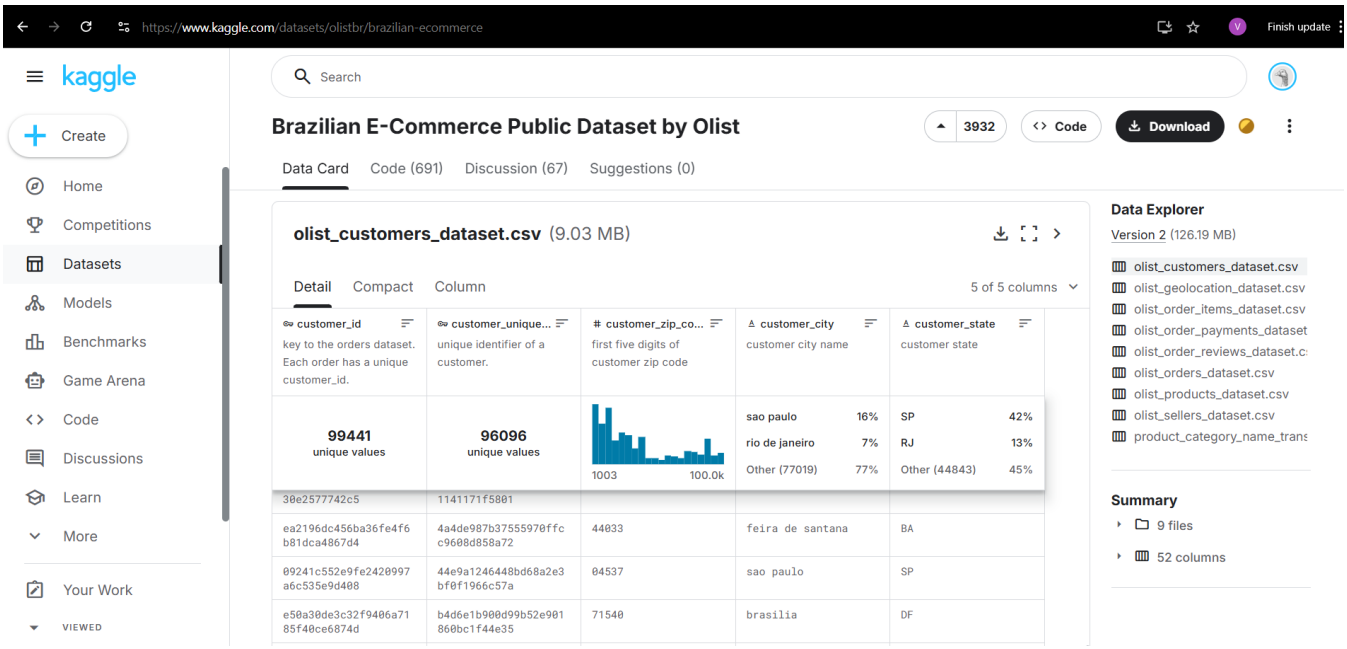
REPORT & SCREEENSHOTS

Tool Used: Microsoft Power BI

Dataset: Brazilian E-Commerce Public Dataset by Olist (From Kaggle) **Link:** https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

Reason for Choosing the dataset

The Brazilian E-Commerce Public Dataset by Olist from Kaggle is ideal for this assignment. It meets all the necessary requirements, featuring over 100,000 rows in the orders table, multiple related tables, and numeric measures such as price, freight value, and payment value. Additionally, it includes important date columns like purchase date, delivery date, and estimated delivery date, making it a comprehensive and realistic business dataset.



SECTION A — DATA PREPARATION (18 Marks)

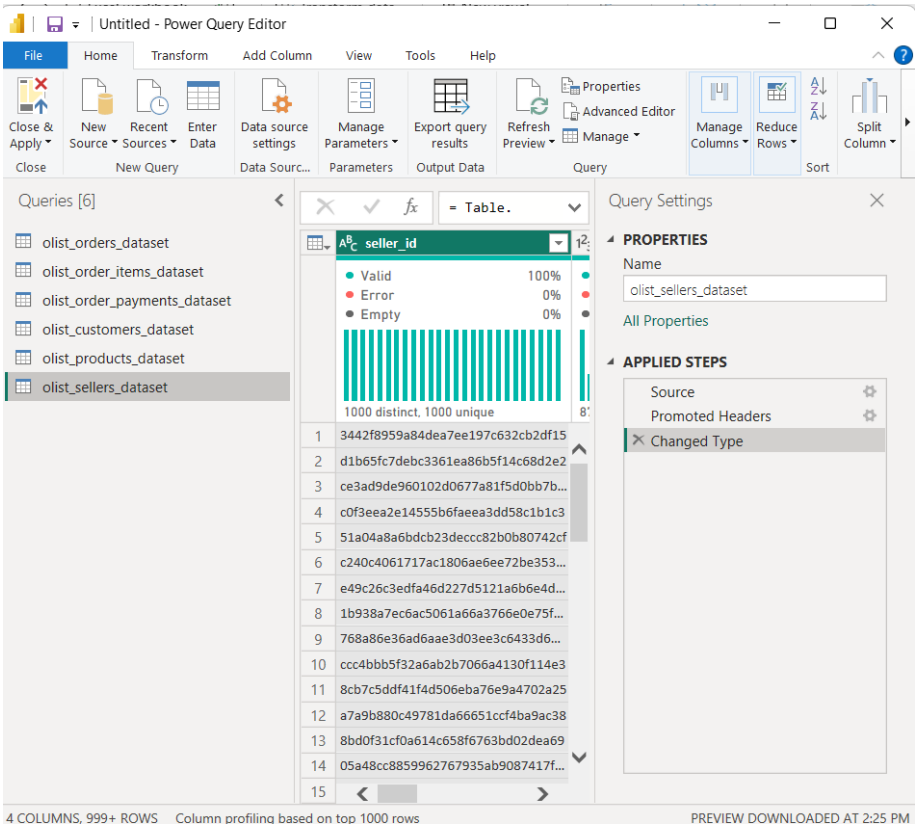
Q1. Data Profiling and Quality Checks (4 Marks)

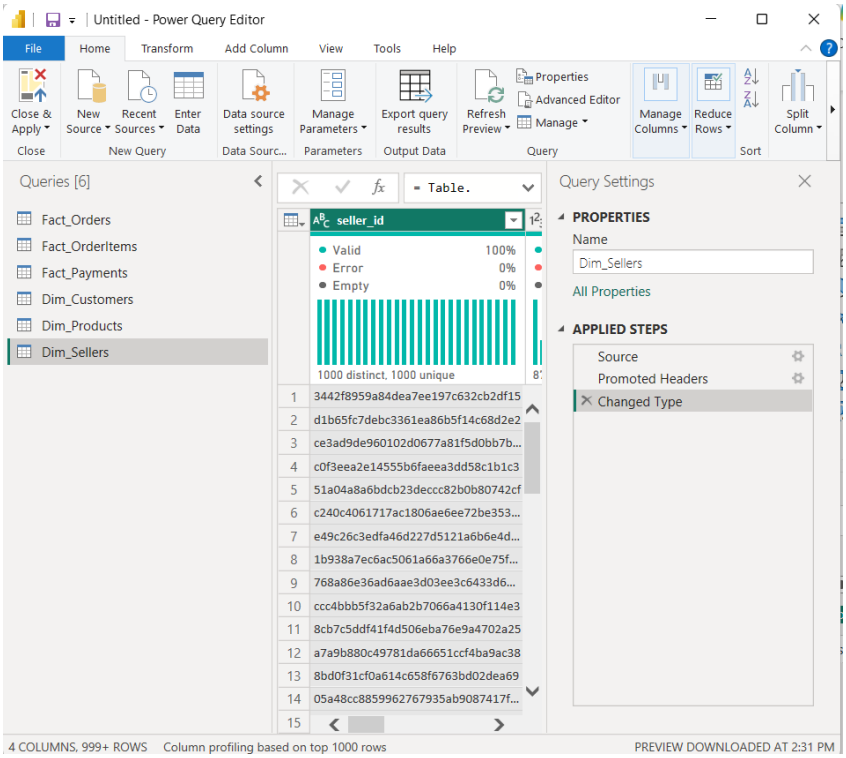
The Olist dataset was downloaded from Kaggle, extracted, and six CSV files were imported into Power BI using:

Home → Get Data → Text/CSV → Transform Data.

The following tables were renamed for proper star schema modelling:

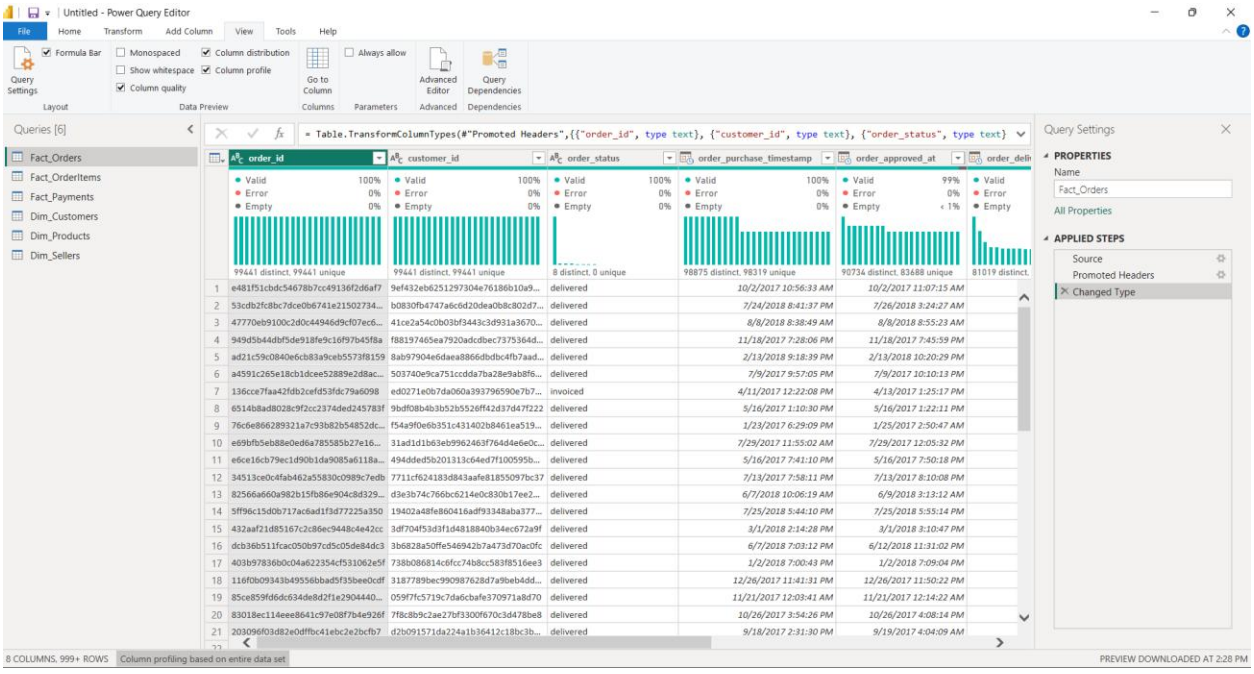
Original Name	Renamed To
olist_orders_dataset	Fact_Orders
olist_order_items_dataset	Fact_OrderItems
olist_order_payments_dataset	Fact_Payments
olist_customers_dataset	Dim_Customers
olist_products_dataset	Dim_Products
olist_sellers_dataset	Dim_Sellers





Data profiling tools were enabled in Power Query Editor by activating:

- Column Quality
- Column Distribution
- Column Profile (Entire Dataset)



Identified Data Quality Issues

1. Missing Delivery Dates

Table: Fact_Orders
Column: order_delivered_customer_date

Column Quality showed null values in this column. These represent undelivered or cancelled orders.

Fix Applied:
No rows were deleted. The null values were preserved because they reflect real business scenarios. Instead, a calculated column was later created to compute delivery days only when the value is not null.

2. Blank Product Categories

Table: Dim_Products
Column: product_category_name

Column profiling showed missing values.

Fix Applied:
Transform → Replace Values
Null values were replaced with "Unknown".

Applied Step: Replaced Value

3. Messy City Names

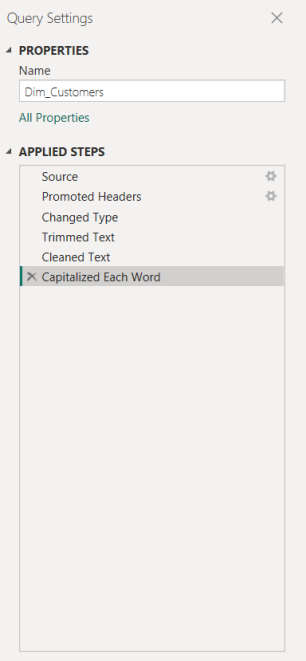
Table: Dim_Customers
Column: customer_city

The column contained lowercase text and spacing inconsistencies.

Fix Applied (in order):

1. Transform → Format → Trim
2. Transform → Format → Clean
3. Transform → Capitalize Each Word

The column was renamed to:
Customer City



4. Incorrect Data Types

Several date and numeric columns were initially loaded as Text.

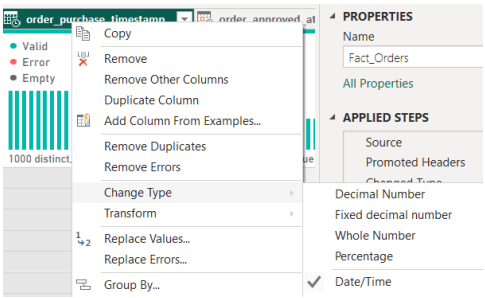
These were corrected in the next section.

Q2. Data Types and Locale Conversion (4 Marks)

The following corrections were made:

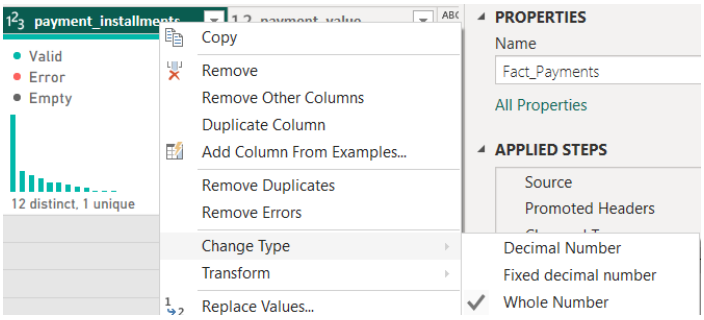
Date Column

Table: Fact_Orders
Column: order_purchase_timestamp
Changed to: Date/Time



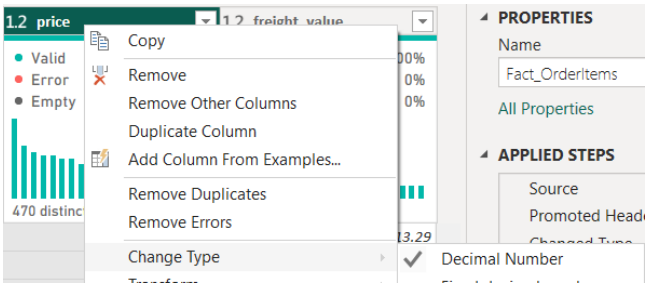
Whole Number

Table: Fact_Payments
Column: payment_installments
Changed to: Whole Number



Decimal Number

Table: Fact_OrderItems
Column: price
Changed to: Decimal Number



Locale Conversion

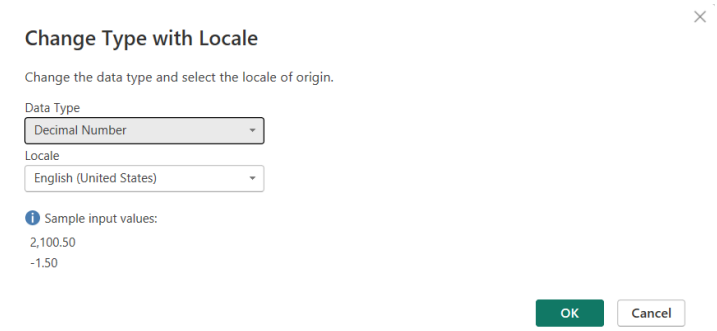
One numeric column that loaded as text was converted using:

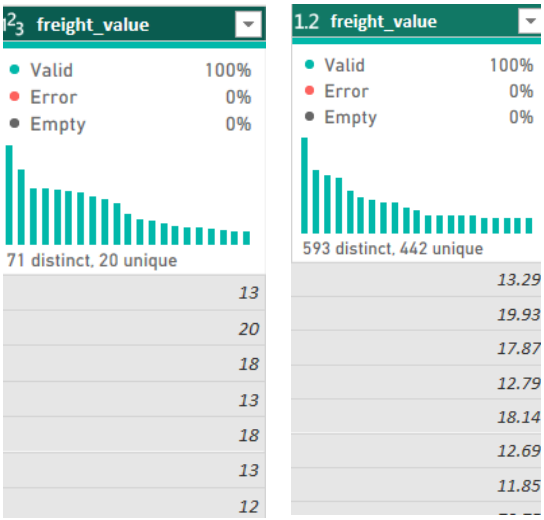
Right Click → Change Type → Using Locale

Selected:

- Data Type: Decimal Number
- Locale: English (United States)

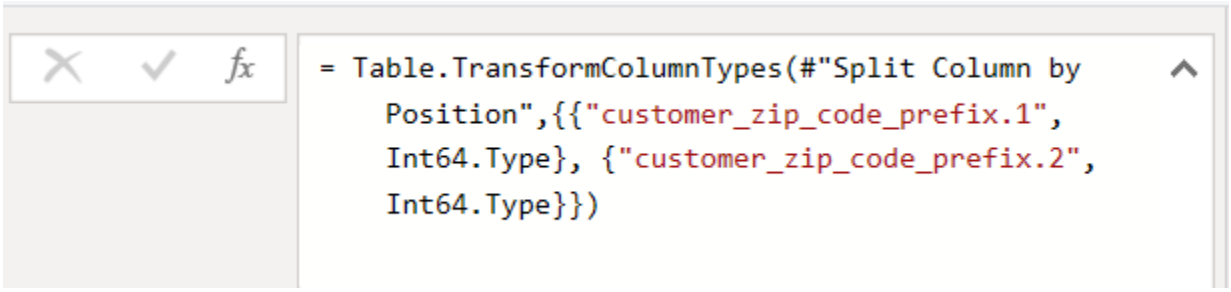
This ensured correct interpretation of decimal formatting.





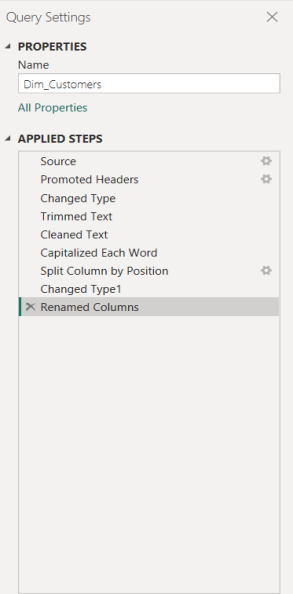
Q3. Text Standardization (4 Marks)

Text standardization was performed in the Dim_Customers table.



The following steps were applied:

1. Trim — removed leading and trailing spaces
2. Clean — removed hidden characters
3. Capitalize Each Word — standardized casing

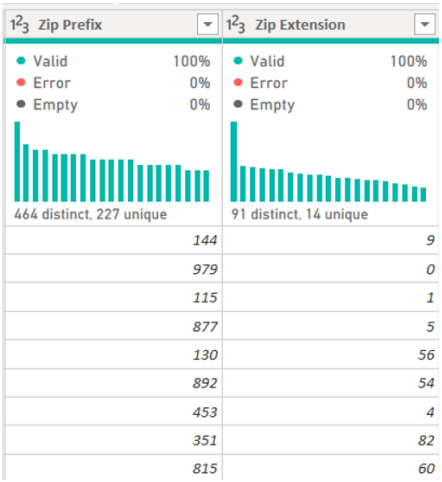


4. **Additionally, the column customer_zip_code_prefix was split:**

Transform → Split Column → By Number of Characters

The resulting columns were renamed:

- Zip Prefix
- Zip Extension



Zip Prefix	Zip Extension
Valid 100%	Valid 100%
Error 0%	Error 0%
Empty 0%	Empty 0%
464 distinct, 227 unique	91 distinct, 14 unique
144	9
979	0
115	1
877	5
130	56
892	54
453	4
351	82
815	60

Q4. Conditional and Custom Columns (4 Marks)

1. **Conditional Column**

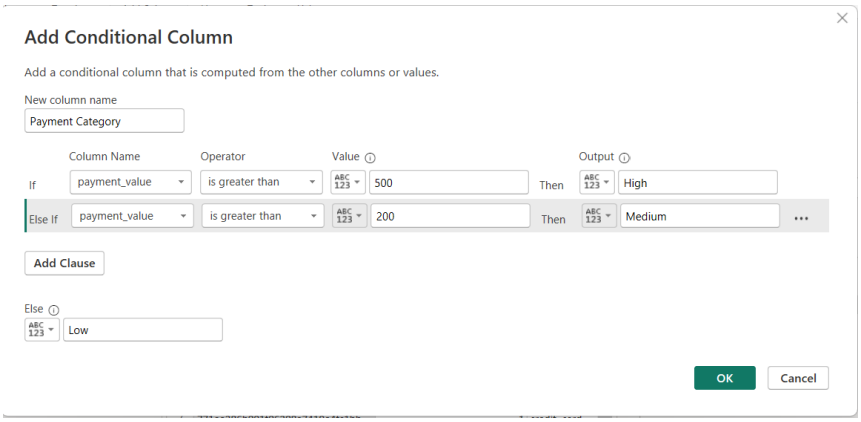
Created in Fact_Payments:

Column Name: Payment Category

Rules:

- payment_value > 500 → High
- payment_value > 200 → Medium
- Else → Low

This categorizes payment size for analytical segmentation.



2. Custom Column (M Formula)

Created in Fact_Orders:

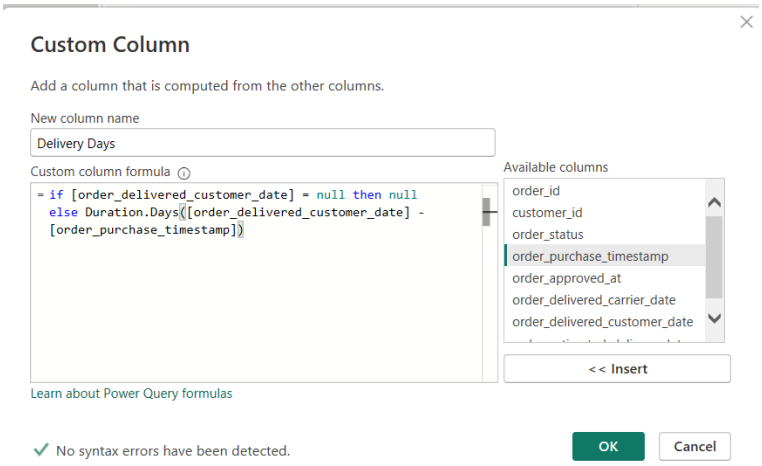
Column Name: Delivery Days

M Formula used:

if [order_delivered_customer_date] = null then null

else Duration.Days([order_delivered_customer_date] - [order_purchase_timestamp])

This calculates delivery duration only for delivered orders.

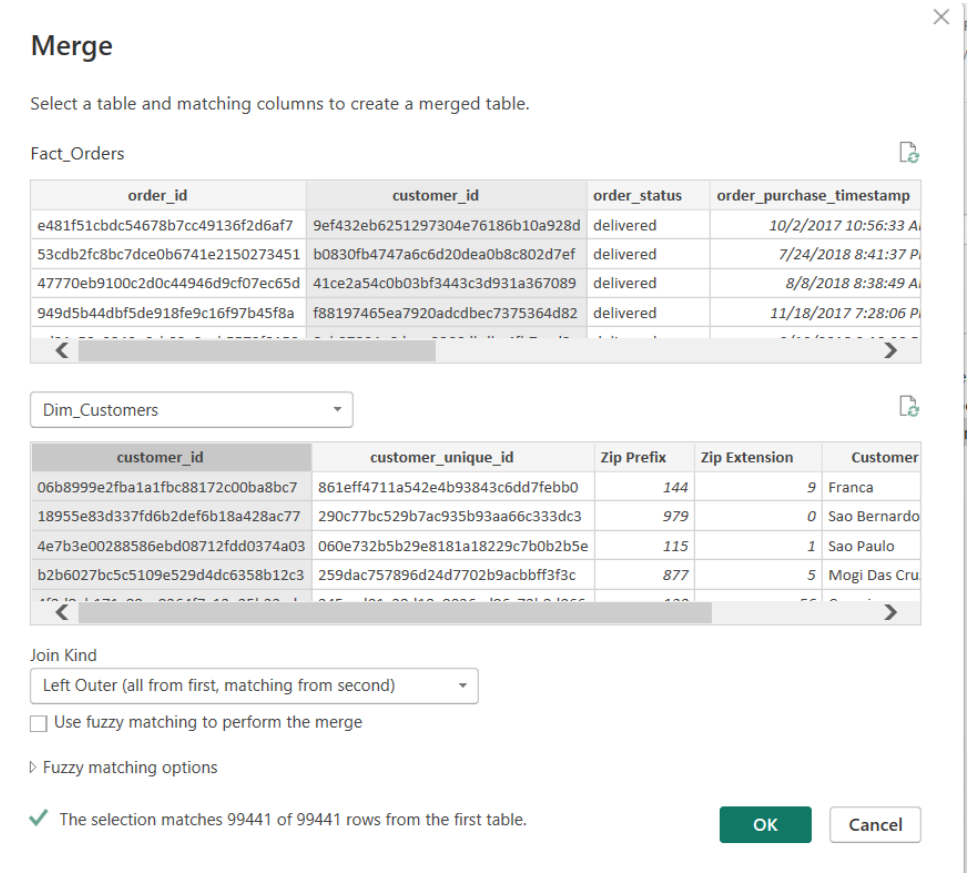


Q5. Merge Queries (2 Marks)

A merge operation was performed between:

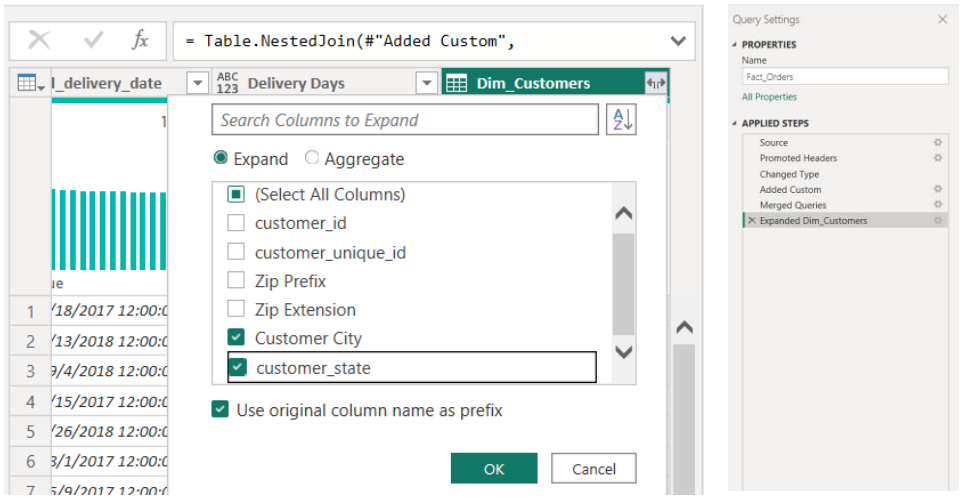
Fact_Orders and Dim_Customers

Join Type: Left Outer (All from Fact_Orders)
Join Key: customer_id



Only the following columns were expanded:

- Customer City & customer_state



Dim_Customers.Customer City	Dim_Customers.customer_state
Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
4119 distinct, 1144 unique	27 distinct, 0 unique
Sao Paulo	SP
Franca	SP
Barreiras	BA
Sao Bernardo Do Campo	SP
Vianopolis	GO
Sao Paulo	SP
Sao Goncalo Do Amarante	RN
Mogi Das Cruzes	SP
Santo Andre	SP
Campinas	SP
Congonhinhas	PR
Jaragua Do Sul	SC
Santa Rosa	RS
Sao Paulo	SP
Nilopolis	RJ
Timoteo	MG
Faxinalzinho	RS
Curitiba	PR
Sorocaba	SP
Belo Horizonte	MG
Rio De Janeiro	RJ
Montes Claros	MG

SECTION B — DATA MODELLING (12 Marks)

Q6. Star Schema Identification (3 Marks)

The final data model follows a Star Schema design.

Fact Tables:

- Fact_Orders
- Fact_OrderItems
- Fact_Payments

Dimension Tables:

- Dim_Customers
- Dim_Products
- Dim_Sellers

Fact tables contain measurable numeric values and foreign keys, while dimension tables contain descriptive attributes.

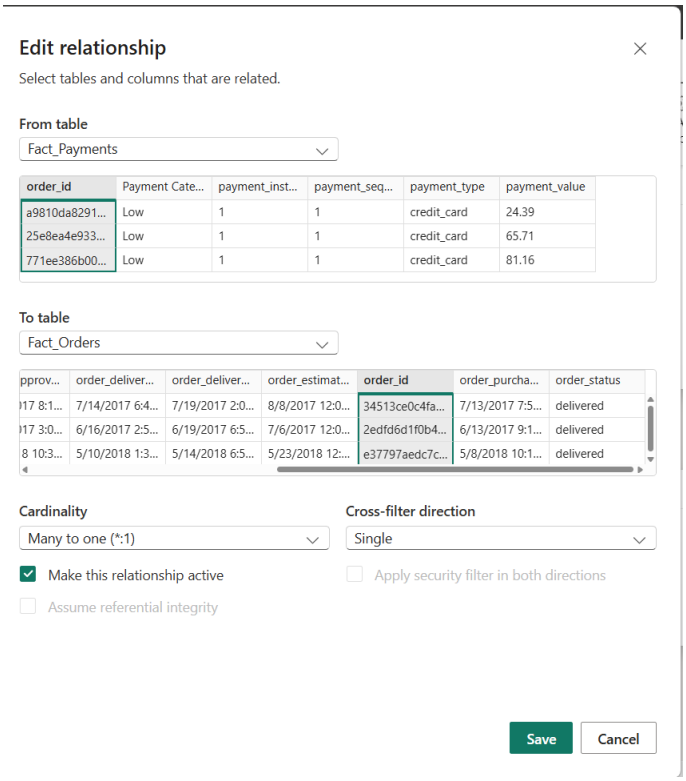
For example:

Fact_OrderItems is classified as a Fact table because it contains measurable fields such as price and freight_value and links to multiple dimensions.

Dim_Customers is classified as a Dimension table because it contains descriptive customer attributes such as city and state.

Q7. Relationships (5 Marks)

Relationships were created in Model View with:



Relationship 1

Fact_Orders (order_id) → Fact_OrderItems (order_id)
Cardinality: One-to-Many
Cross Filter: Single
Active: Yes

Justification: One order can contain multiple order items.

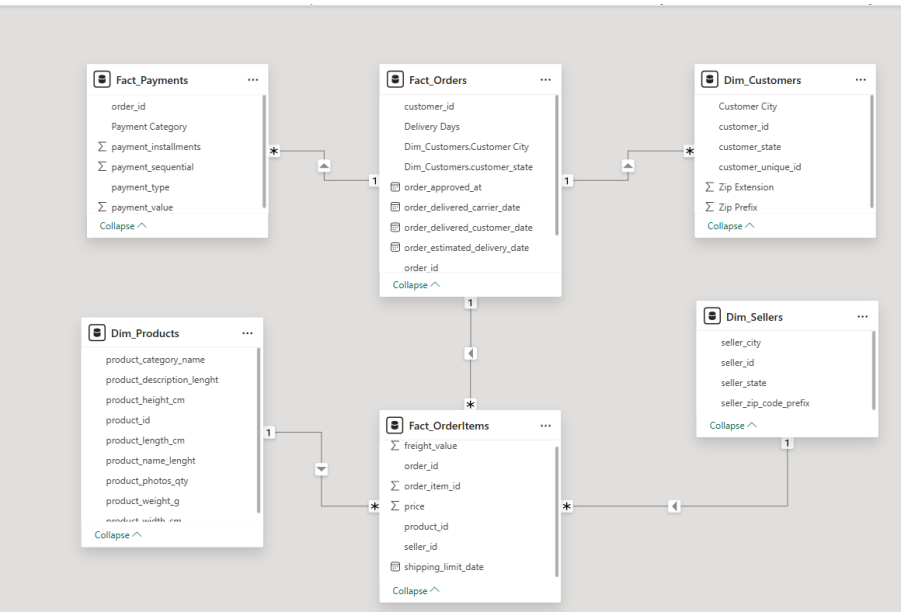
Relationship 2

Fact_Orders (customer_id) → Dim_Customers (customer_id)
Cardinality: One-to-Many
Cross Filter: Single
Active: Yes

Justification: One customer may place multiple orders.

Modelling Mistake Example

If relationships were set to Many-to-Many or Bi-Directional filtering, it could create ambiguous filter paths, leading to double counting of revenue and incorrect totals.



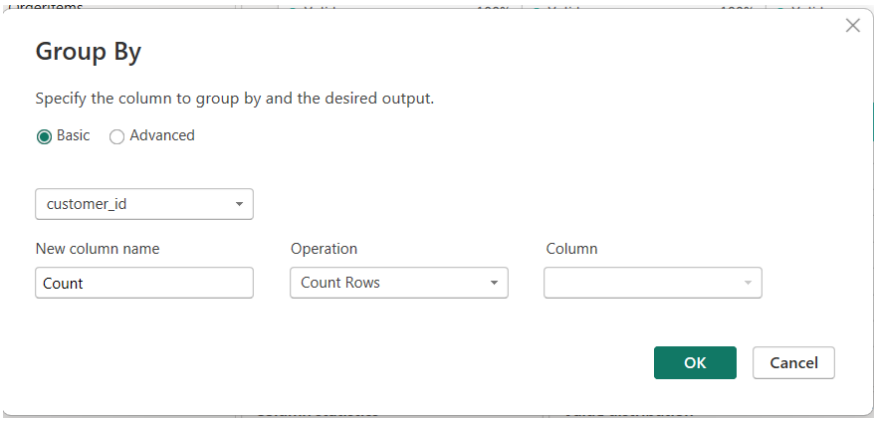
Q8. Key Uniqueness Verification (2 Marks)

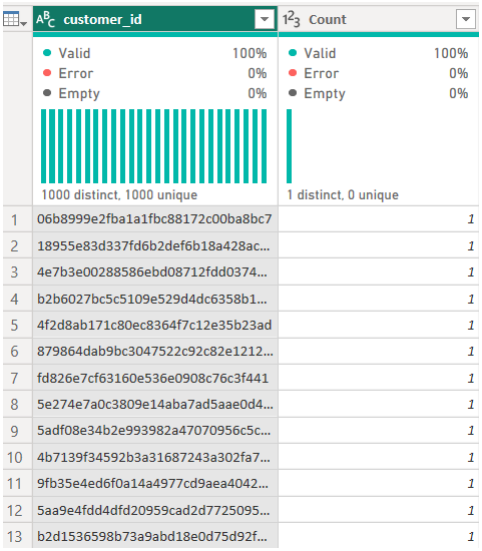
The column customer_id in Dim_Customers was tested using:

Transform → Group By → Count Rows

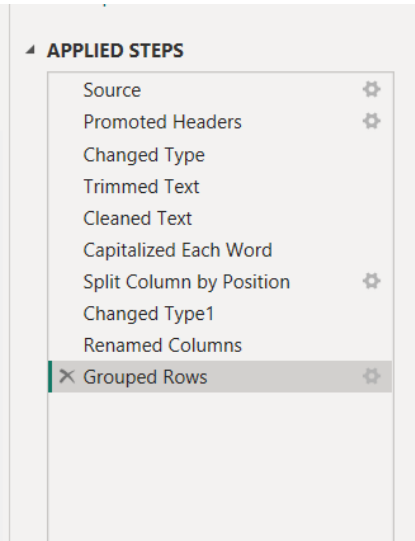
No duplicate keys were found.

If duplicates had existed, they would have been removed to preserve referential integrity.





	customer_id	Count
	Valid 100%	Valid 100%
	Error 0%	Error 0%
	Empty 0%	Empty 0%
	1000 distinct, 1000 unique	1 distinct, 0 unique
1	06b8999e2fba1a1fbc88172c00ba8bc7	1
2	18955e83d337fd6b2def6b18a428ac...	1
3	4e7b3e00288586ebd08712fdd0374...	1
4	b2b6027bc5c5109e529d4dc6358b1...	1
5	4f2d8ab171c80ec8364f7c12e35b23ad	1
6	879864dab9bc3047522c92c82e1212...	1
7	fd826e7cf63160e536e0908c76c3f441	1
8	5e274e7a0c3809e14aba7ad5aae0d4...	1
9	5adf08e34b2e993982a47070956c5c...	1
10	4b7139f34592b3a31687243a302fa7...	1
11	9fb35e4ed6f0a14a4977cd9aea4042...	1
12	5aa9e4fdd4dfd20959cad2d7725095...	1
13	b2d1536598b73a9abd18e0d75d92f...	1



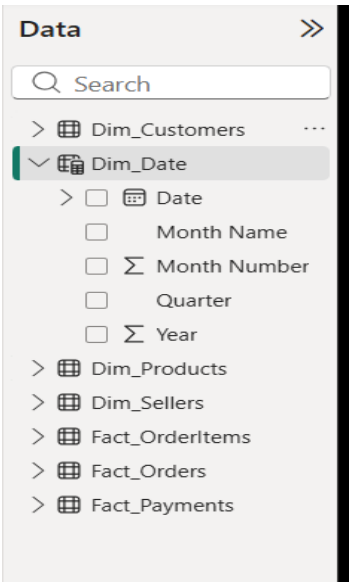
Q9. Date Table Decision (2 Marks)

Should a date table exist? Yes

A Date Table was created using DAX:

The table was marked as a Date Table and related to Fact_Orders.

```
1  Dim_Date =
2  ADDCOLUMNS(
3      CALENDAR(
4          MIN(Fact_Orders[order_purchase_timestamp]),
5          MAX(Fact_Orders[order_purchase_timestamp])
6      ),
7      "Year", YEAR([Date]),
8      "Month Number", MONTH([Date]),
9      "Month Name", FORMAT([Date], "MMMM"),
10     "Quarter", "Q" & FORMAT([Date], "Q")
11 )
12
```



Reasons for creating a Date Table:

1. Enables time intelligence calculations (YTD, MTD, MoM).
2. Ensures consistent time-based filtering across the model.

Edit relationship

Select tables and columns that are related.

From table

Fact_Orders

pprov...	order_deliver...	order_deliver...	order_estimat...	order_id	order_purcha...	order_status
)17 8:1...	7/14/2017 6:4...	7/19/2017 2:0...	8/8/2017 12:0...	34513ce0c4fa...	7/13/2017 7:5...	delivered
)17 3:0...	6/16/2017 2:5...	6/19/2017 6:5...	7/6/2017 12:0...	2edfd6d1f0b4...	6/13/2017 9:1...	delivered
8 10:3...	5/10/2018 1:3...	5/14/2018 6:5...	5/23/2018 12:...	e37797aedc7...	5/8/2018 10:1...	delivered

To table

Dim_Date

Date	Month Name	Month Number	Quarter	Year
7/1/2017 12:0...	July	7	Q3	2017
7/2/2017 12:0...	July	7	Q3	2017
7/3/2017 12:0...	July	7	Q3	2017

Cardinality

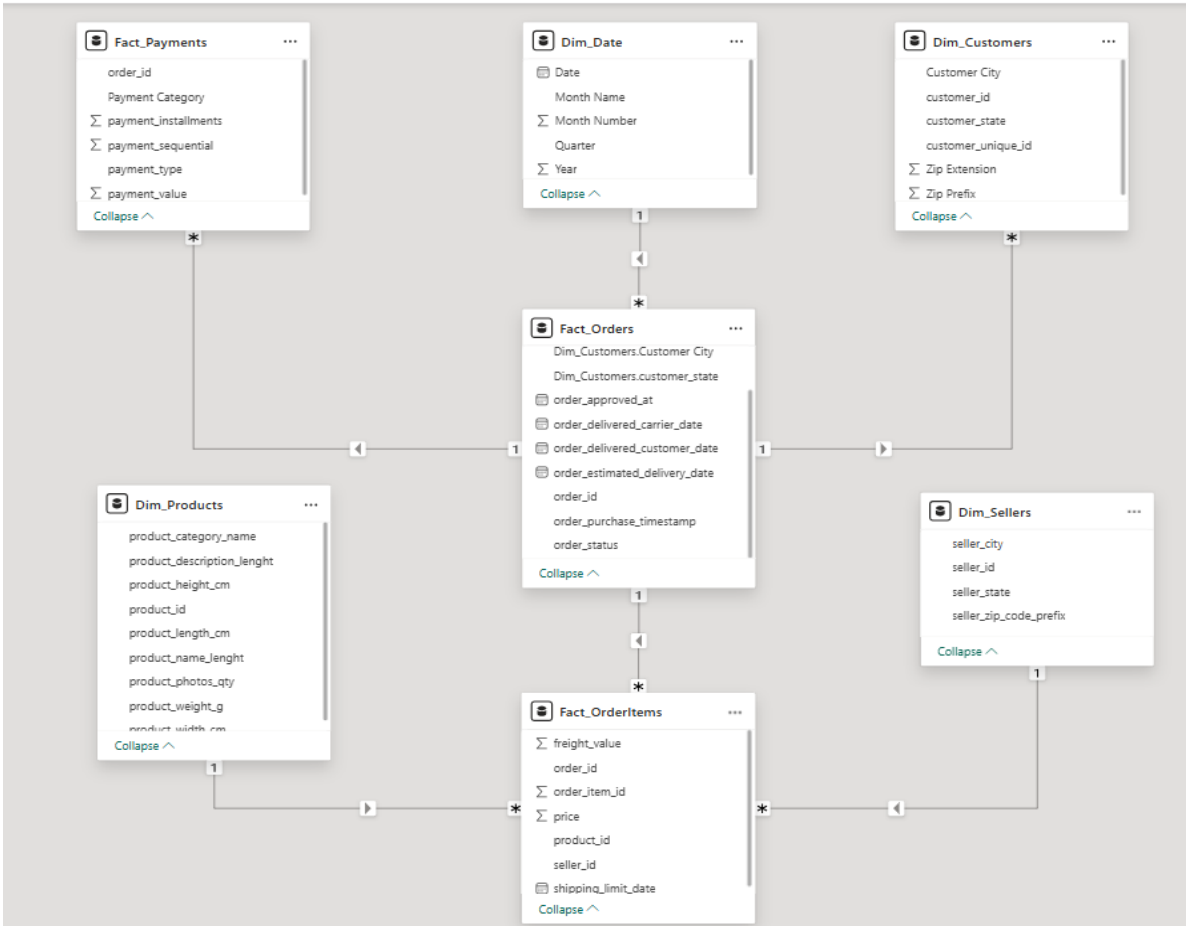
Many to one (*:1)

Cross-filter direction

Single

- [x] Make this relationship active
- [ ] Assume referential integrity
- [ ] Apply security filter in both directions

Save | Cancel



DATA DICTIONARY

Table	Column	Meaning	Type	Key/Attribute
Fact_Orders	order_id	Unique order identifier	Text	Primary Key
Fact_OrderItems	price	Item selling price	Decimal	Measure
Fact_Payments	payment_value	Total payment value	Decimal	Measure
Dim_Customers	customer_id	Unique customer ID	Text	Primary Key
Dim_Products	product_id	Unique product ID	Text	Primary Key
Dim_Sellers	seller_id	Unique seller ID	Text	Primary Key
Fact_Orders	order_purchase_timestamp	Purchase date	DateTime	Attribute
Fact_Orders	Delivery Days	Delivery duration	Whole Number	Calculated