Technical Specification Document

Sales Data Analysis and Data Modelling to support Business Requirements

Author - Shantanu Zodey

Date – 30/05/2025

Objective

The purpose of this project is to transform raw orders, customer, and shipping data from different formats (Excel, CSV, JSON) into a unified data model that supports advanced business reporting and analytics. The goal is to enable the business to gain valuable insights into customer behavior, product performance, shipping efficiency, and geographic trends, which will drive strategic decision-making.

By leveraging SQL-based data modeling, this project aims to ensure data consistency, integrity, and traceability, while preparing the dataset for scalable reporting across key metrics like total sales, customer transactions, product popularity, and delivery statuses.

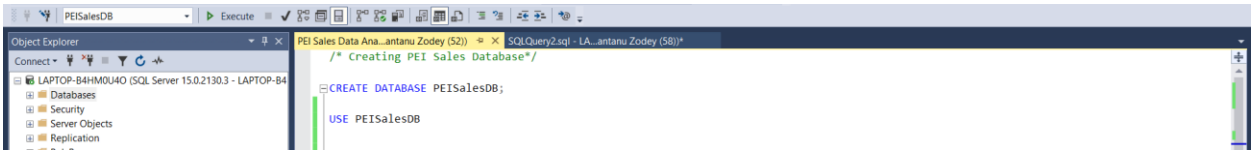
Project Scope

This project includes the following activities:

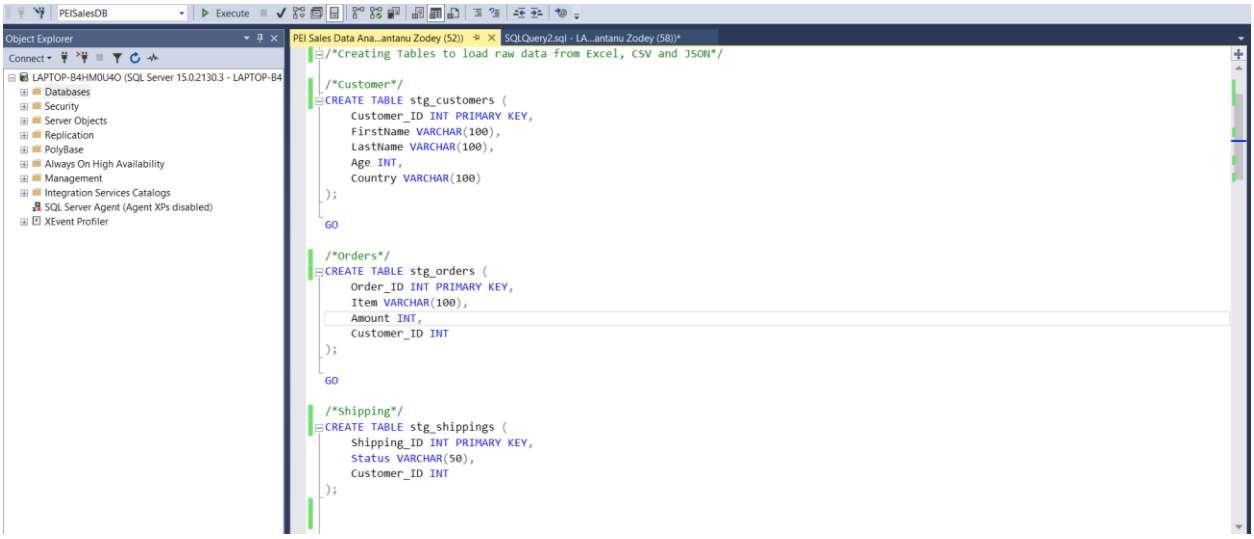
1. **Source Data Integration**
    - Ingesting data from three sources:
        - Customer (Excel)
        - Orders (CSV)
        - Shipping (JSON)
2. **Data Quality Checks**
    - Performing completeness, accuracy, and referential integrity checks using SQL.
3. **Data Modeling**
    - Designing a star schema using dimension and fact tables.
    - Resolving inconsistencies in item prices and categorize customers by age.
4. **ETL Logic Development**
    - Defining and implement transformations to populate the target model.
5. **Enable Reporting Requirements**
    - Structure data to support five business reporting use cases including:
        - Total spent per country with pending deliveries
        - Product performance by age category
        - Transaction volume and sales per customer
        - Country with lowest sales and transaction count

Source Data Integration

a) Creating Database PEISalesDB



b) Creating Staging Tables for Customers, Orders and Shipping.



- The above staging tables will be used to load raw data and perform exploratory data analysis, data quality, data consistency and data accuracy checks.
- Once the staging tables are created, we loaded the data for Customer data from Excel and Orders data from CSV using Import Data functionality in Microsoft SQL Server. As Shipping data is in JSON format, Microsoft SQL Server does not support Import Data functionality for JSON file format. So, to convert JSON data into CSV data, we leveraged python and converted shipping.json data to shipping.csv file and then loaded the shipping data using Import Data functionality. Please find below snapshot.

jupyter Untitled18 Last Checkpoint: 7 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Converting Shipping.json data to CSV data

In [1]:
```
import pandas as pd
import json
```

In [8]:
```
pwd
```

Out[8]:
```
'C:\\Users\\Shantanu Zodey\\Desktop\\Data Task'
```

In [9]:
```
# Loading JSON data from Shipping.json file
with open('Shipping.json', 'r') as file:
    data = json.load(file)
```

In [10]:
```
# Converting to DataFrame
df = pd.DataFrame(data)
```

In [12]:
```
# Saving to CSV
df.to_csv('Shipping.csv', index=False)
```

In [13]:
```
ls
```

```
 Volume in drive C is Windows
 Volume Serial Number is F46E-2E44

 Directory of C:\Users\Shantanu Zodey\Desktop\Data Task

05/30/2025  02:13 AM    <DIR>          .
05/30/2025  01:20 AM    <DIR>          ..
05/29/2025  12:34 AM            31,232 Customer.xls
05/29/2025  12:34 AM             5,221 Order.csv
05/30/2025  02:13 AM             4,260 Shipping.csv
05/29/2025  12:34 AM            20,230 Shipping.json
               4 File(s)         60,943 bytes
               2 Dir(s)   4,913,369,088 bytes free
```

c) Validating all the records and counts to ensure all records are loaded into staging table.

Stg_customer

PEISalesDB | Execute

Object Explorer

Connect

LAPTOP-B4HM0U4O (SQL Server 15.0.2130.3 - LAPTOP-B4

- Databases
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

LAPTOP-B4HM0U4O....- dbo.fact_orders* | PEI Sales Data Mod...antanu Zodey (57)) | PEI Sales Data Ana...antanu Zodey (52)) | SQLQuery2.sql - LA...antanu Zodey (58))*

```
select * from stg_customers
```

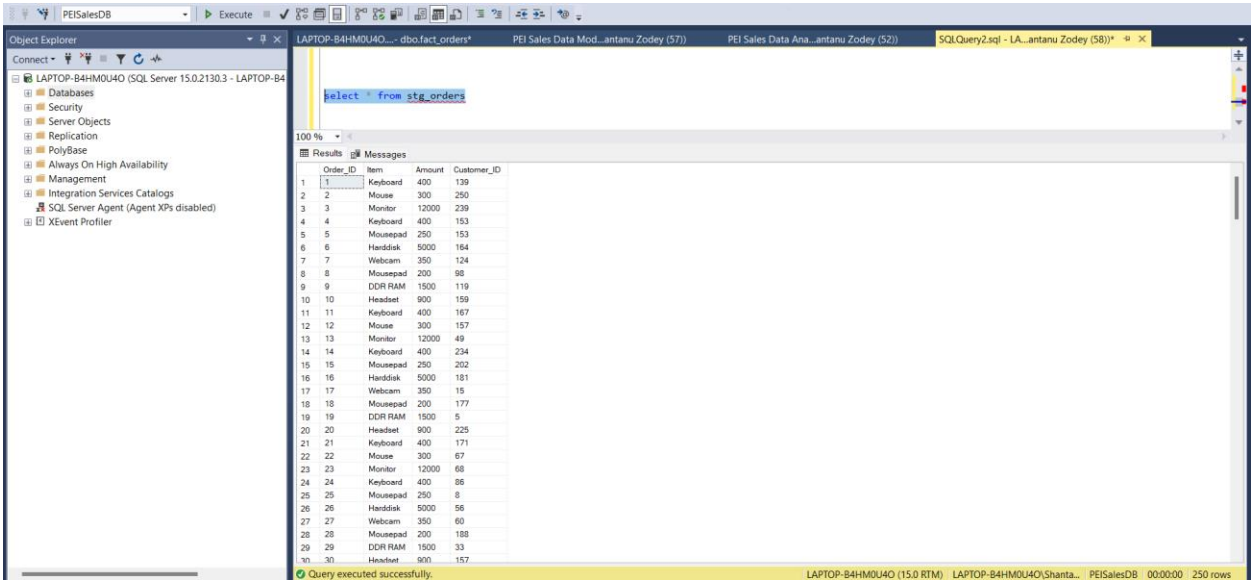
100 %

Results | Messages

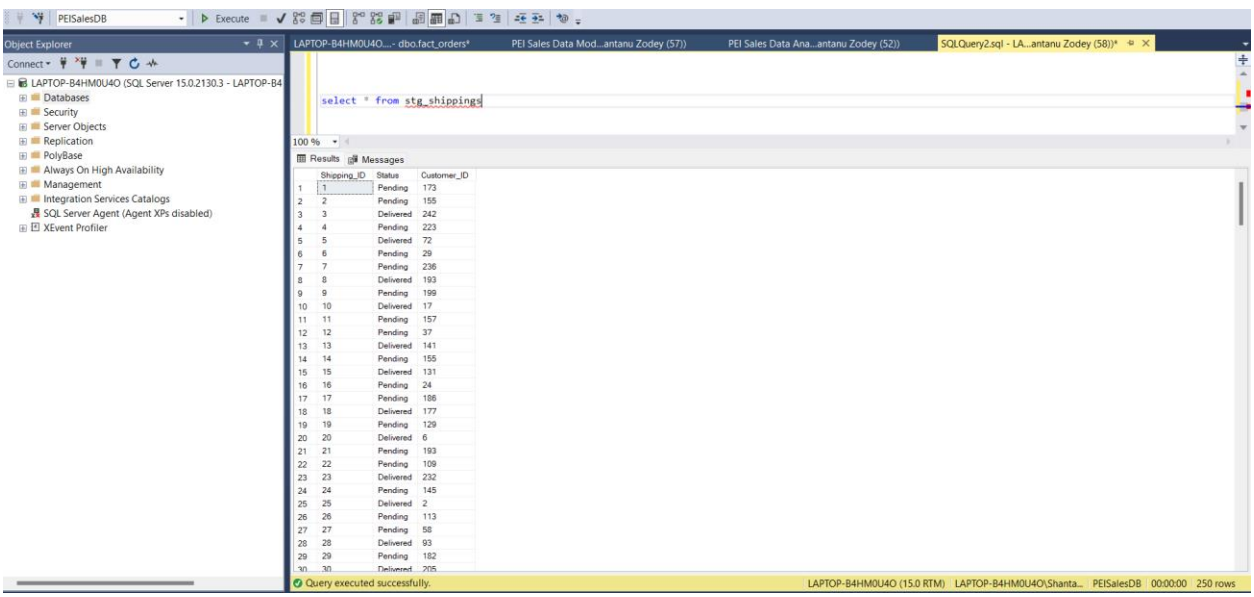
	Customer_ID	FirstName	LastName	Age	Country
1	1	Joseph	Rice	43	USA
2	2	Gary	Moore	71	USA
3	3	John	Walker	44	UK
4	4	Eric	Carter	38	UK
5	5	William	Jackson	58	UAE
6	6	Nicole	Jones	33	USA
7	7	David	Davis	59	USA
8	8	Jason	Montgomery	58	UK
9	9	Kent	Weaver	61	UK
10	10	Darrell	Dillon	50	UAE
11	11	Jacqueline	Wang	22	USA
12	12	Jodi	Gonzalez	69	USA
13	13	Omar	Martin	28	UK
14	14	Nicole	Lara	77	UK
15	15	Jason	Brown	63	UAE
16	16	David	Benson	61	USA
17	17	Kimberly	Mora	34	USA
18	18	Erik	Macias	25	UK
19	19	James	Johnson	67	UK
20	20	Derek	Peterson	36	UAE
21	21	Diane	Henson	74	USA
22	22	Cody	Lyons	47	USA
23	23	Margaret	Hardy	25	UK
24	24	Arthur	Hayes	75	UK
25	25	Raymond	Taylor	36	UAE
26	26	Tina	Moore	34	USA
27	27	Kylie	White	63	USA
28	28	Jonathan	Manning	30	UK
29	29	Angela	Bryant	23	UK
30	30	Michael	Mann	56	UAE

Query executed successfully. | LAPTOP-B4HM0U4O (15.0 RTM) | LAPTOP-B4HM0U4O\Shanta... | PEISalesDB | 00:00:00 | 250 rows

stg_orders

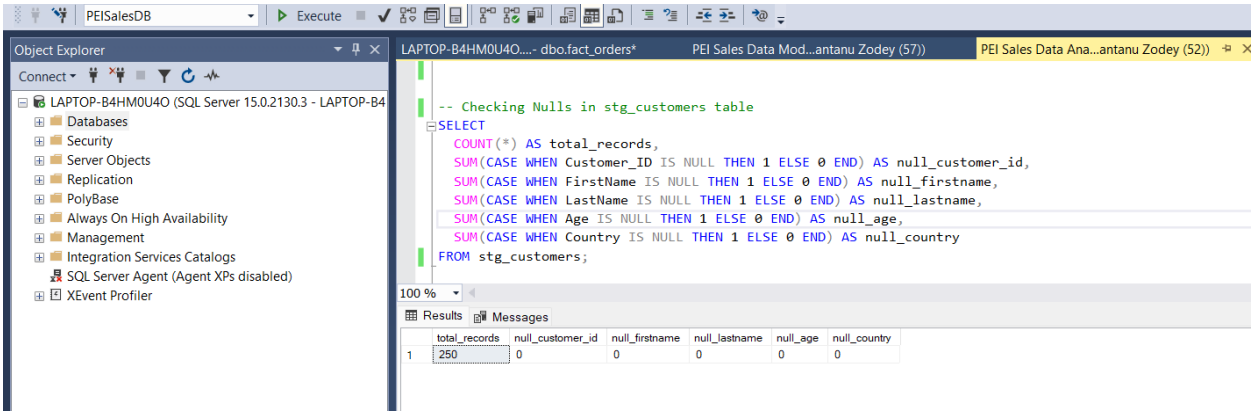


stg_shippings

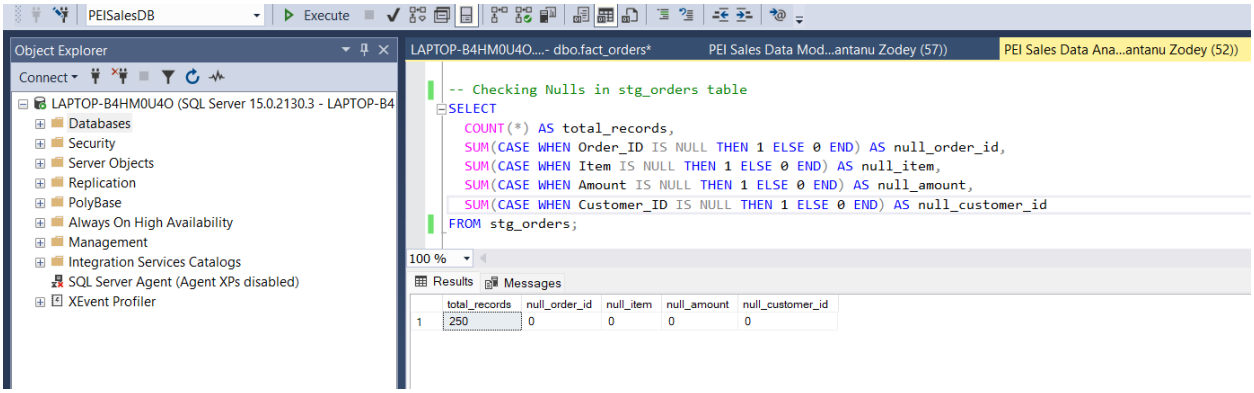


Data Quality Checks

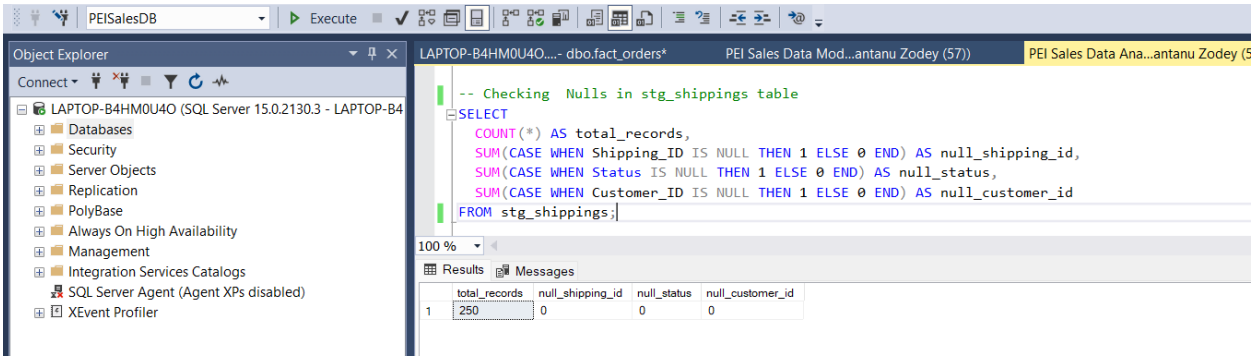
a) Checking Nulls in stg_customer table



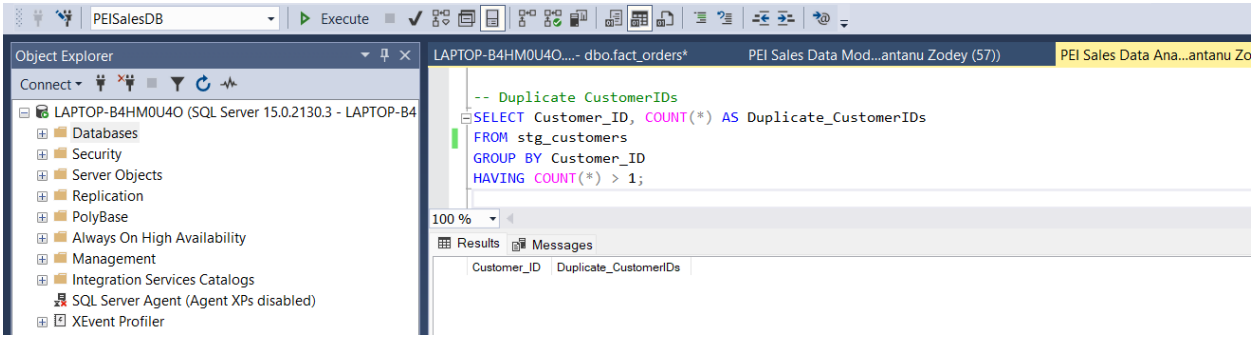
b) Checking Nulls in stg_orders table



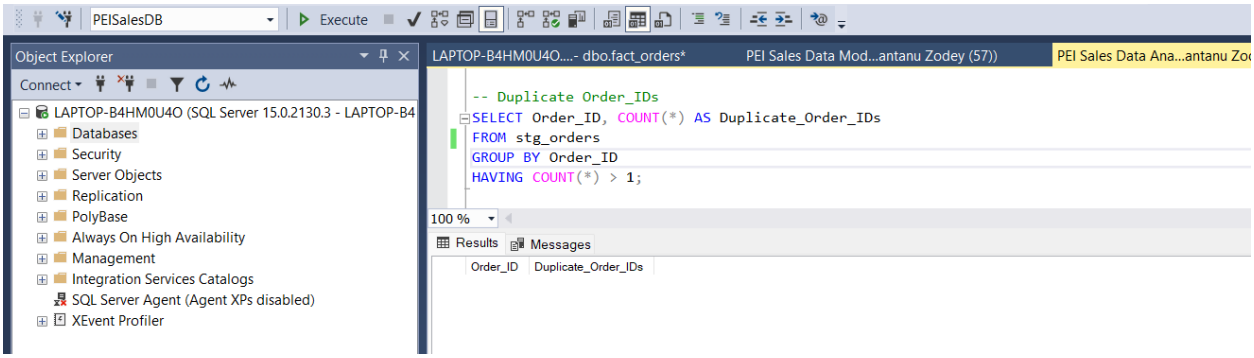
c) Checking Nulls in stg_shipping table



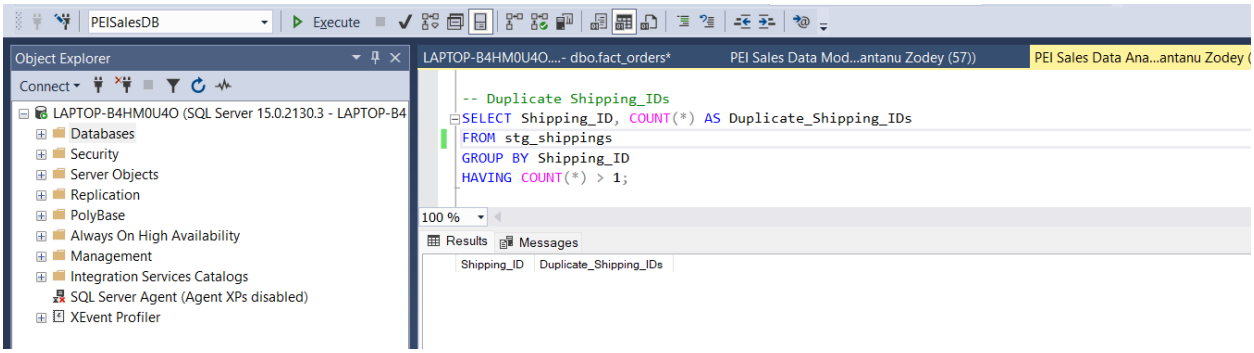
d) Checking Duplicate Customer IDs in stg_customer table



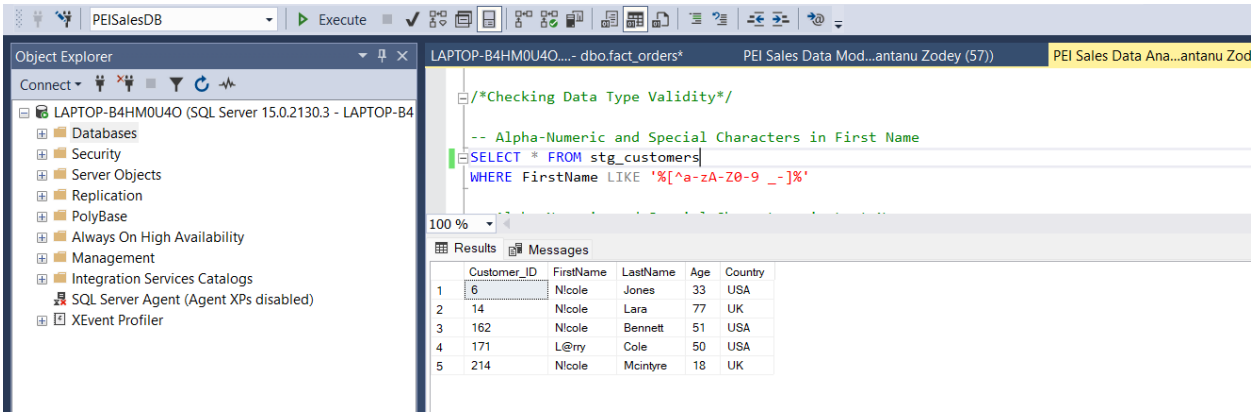
e) Checking Duplicate Order IDs in stg_orders table



f) Checking Duplicate Shipping IDs in stg_shipping table

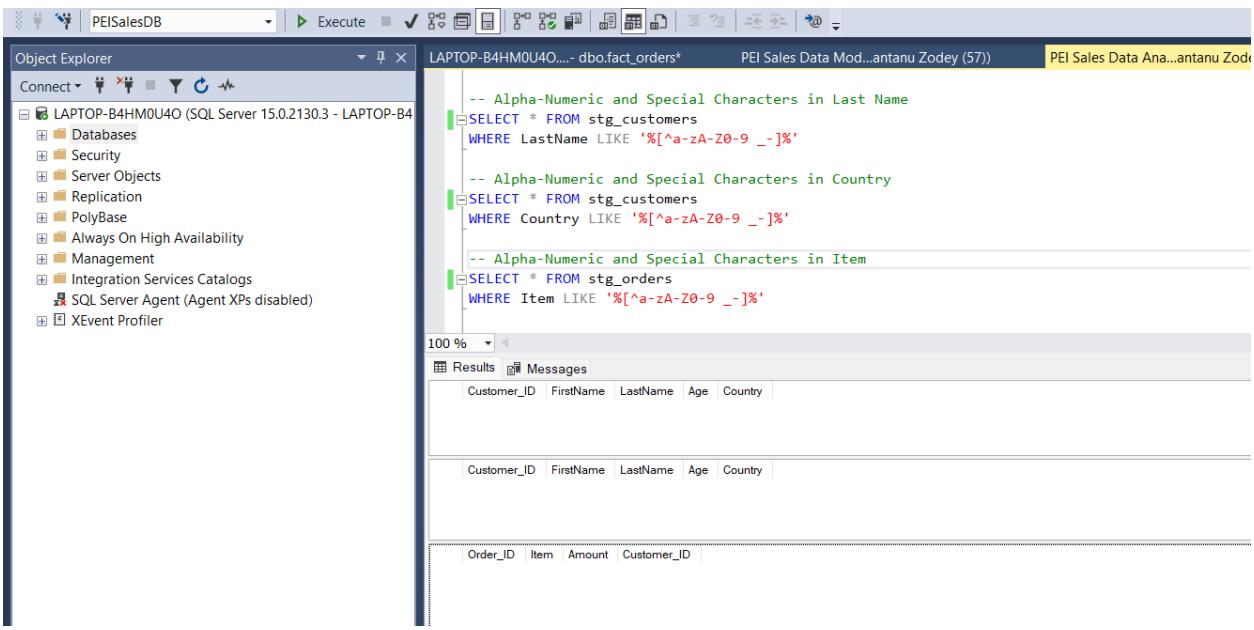


g) Checking Alpha-Numeric and Special characters in First Name

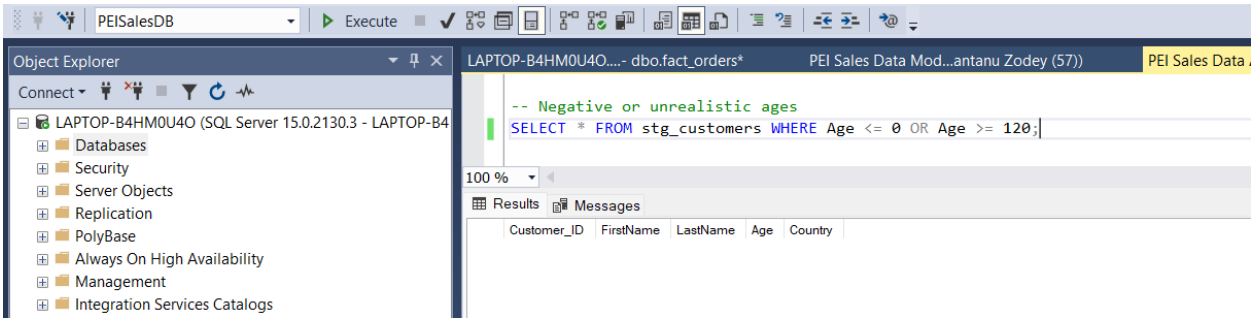


In the above screenshot, there are few records in First Name that contains special characters. This column needs transformation in order to get the clean and accurate data.

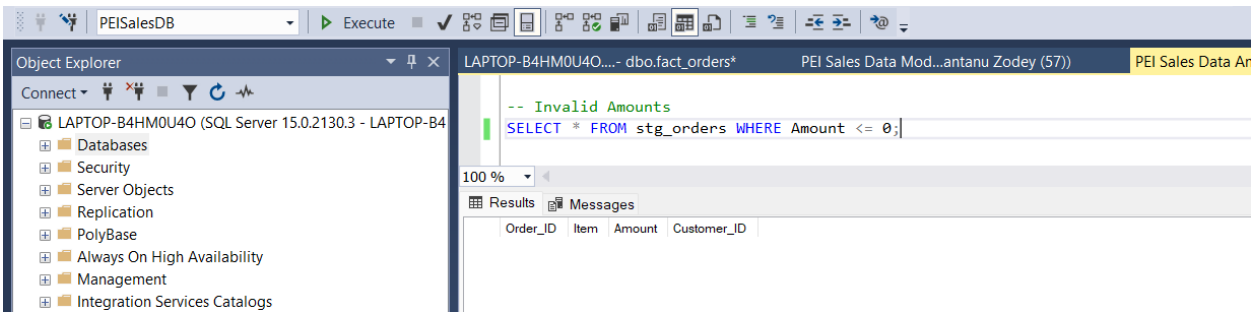
h) Checking Alpha-Numeric and Special characters in Last Name, Country and Item



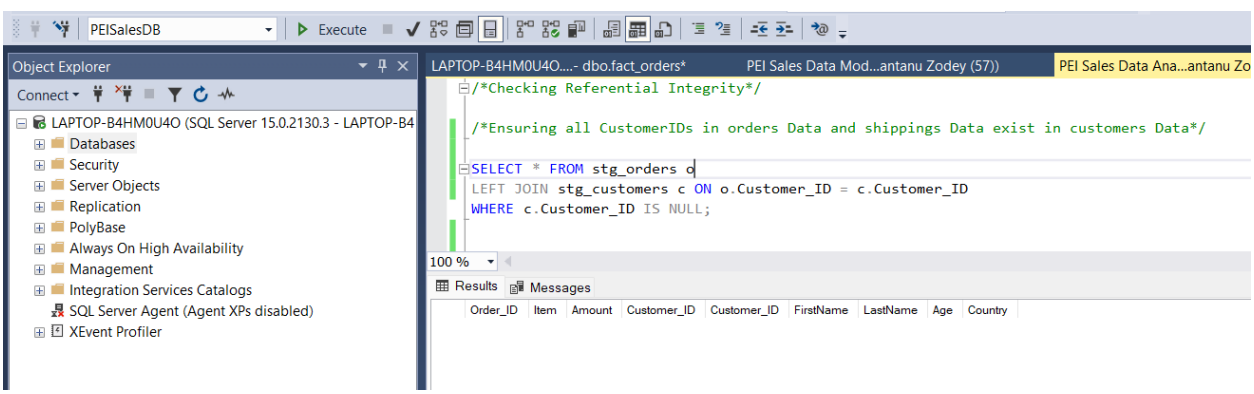
i) Checking Negative or Unrealistic Ages.



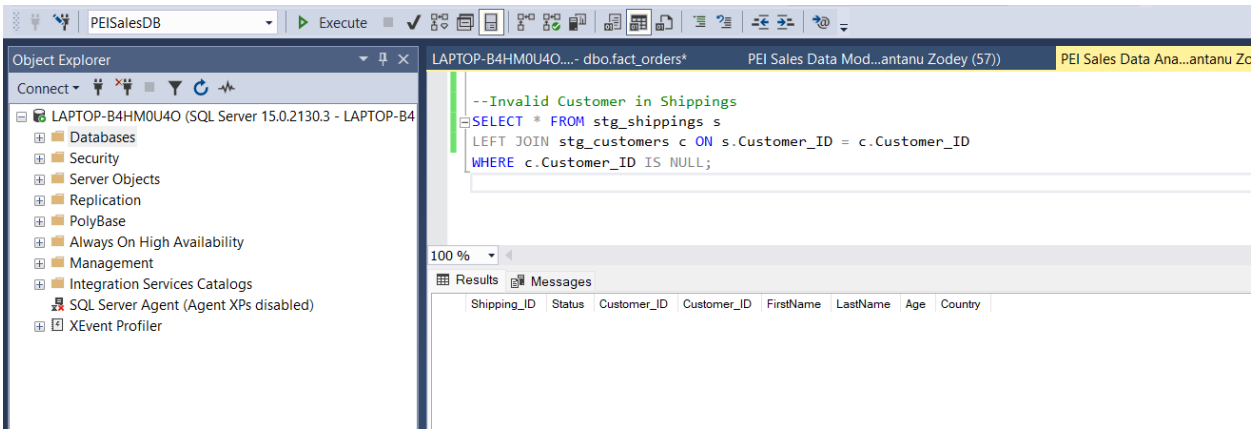
j) Checking Invalid Amounts



k) Referential Integrity Checks for valid customers present in stg_orders



l) Referential Integrity Checks for valid customers present in stg_shippings



Findings for Data Quality checks:

- First Name column in stg_customer table contains special characters
- Duplicate Items with different amounts.
- All datasets are joinable via CustomerID.
- No nulls found in critical fields.
- No duplicates found in critical fields.

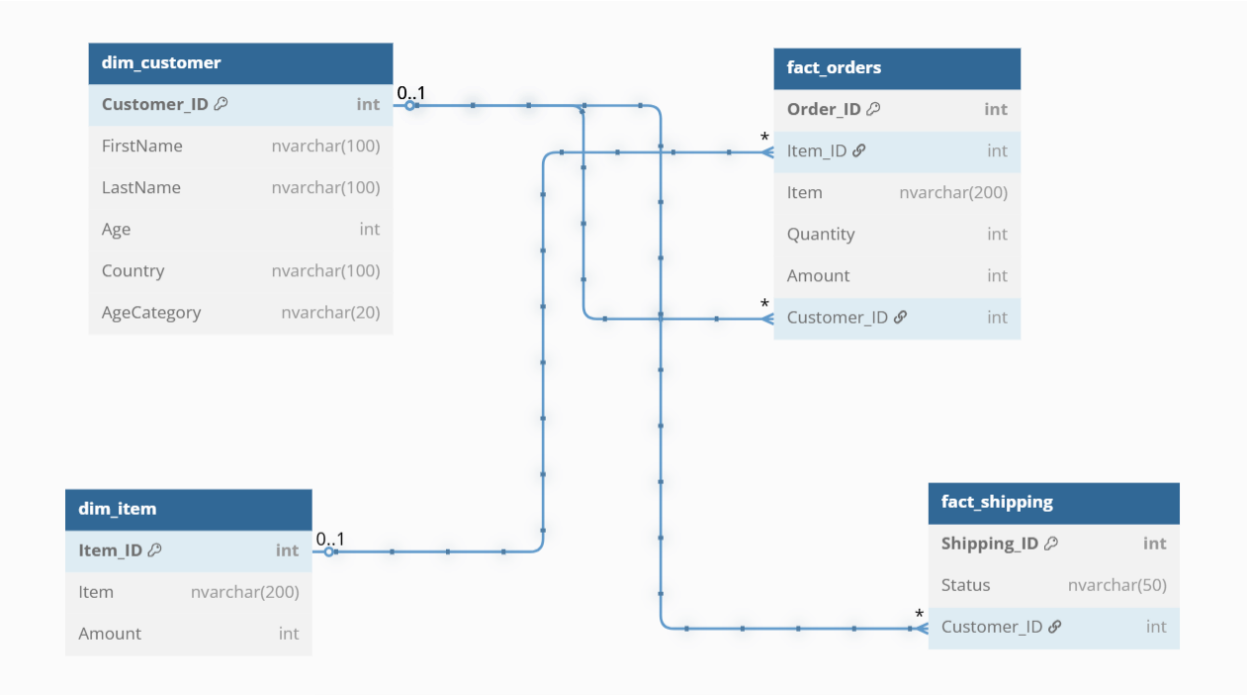
Data Modelling

To fulfill the Business reporting requirement, we designed a **Star Schema** with:

- **2 Dimension Tables**: dim_customer, dim_product
- **2 Fact Tables**: fact_orders, fact_shipping

Please find below snapshot.

Domain Model (ERD)



Below are the table schema's that has been created in Database.

```sql
-- Dimension: Customer
CREATE TABLE dim_customer (
    Customer_ID INT PRIMARY KEY,
    FirstName NVARCHAR(100),
    LastName NVARCHAR(100),
    Age INT,
    Country NVARCHAR(100),
    AgeCategory NVARCHAR(20)
);

-- Dimension: Product
CREATE TABLE dim_item (
    Item_ID INT PRIMARY KEY IDENTITY(1,1),
    Item NVARCHAR(200) UNIQUE,
    Amount INT  -- Standard price per product
);

-- Fact Table: Orders
CREATE TABLE fact_orders (
    Order_ID INT PRIMARY KEY,
    Item_ID INT,
    Amount INT,
    Quantity INT DEFAULT 1,
    Customer_ID INT,
    CONSTRAINT fk_orders_product FOREIGN KEY (Item_ID) REFERENCES dim_item(Item_ID),
    CONSTRAINT fk_orders_customer FOREIGN KEY (Customer_ID) REFERENCES dim_customer(Customer_ID)
);

-- Fact Table: Shipping
CREATE TABLE fact_shipping (
    Shipping_ID INT PRIMARY KEY,
    Status NVARCHAR(50),
    Customer_ID INT,
    CONSTRAINT fk_shipping_customer FOREIGN KEY (Customer_ID) REFERENCES dim_customer(Customer_ID)
);
```

Loading data from stg_customer to dim_customer.

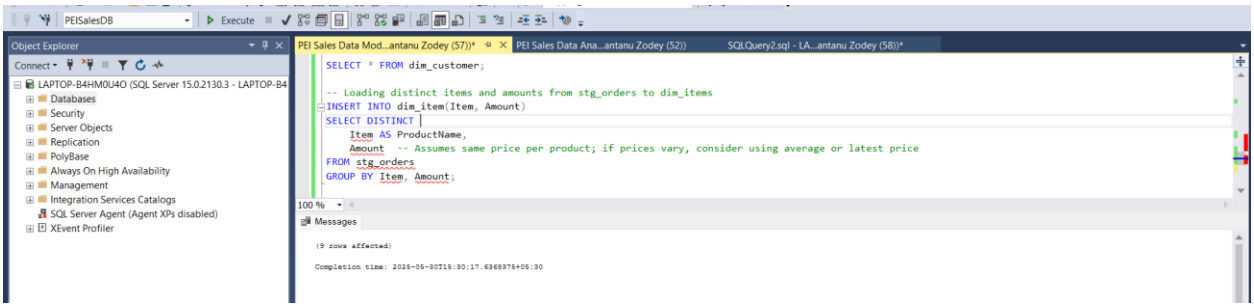
```sql
-- Loading data from stg_customer to dim_customer
INSERT INTO dim_customer (Customer_ID, FirstName, LastName, Age, Country, AgeCategory)
SELECT
    Customer_ID,
    FirstName,
    LastName,
    Age,
    Country,
    CASE
        WHEN Age < 30 THEN 'Less than 30'
        ELSE 'Above 30'
    END AS AgeCategory -- Added bew column Age category to distinguish different age group customer
FROM dbo.stg_customers;
```

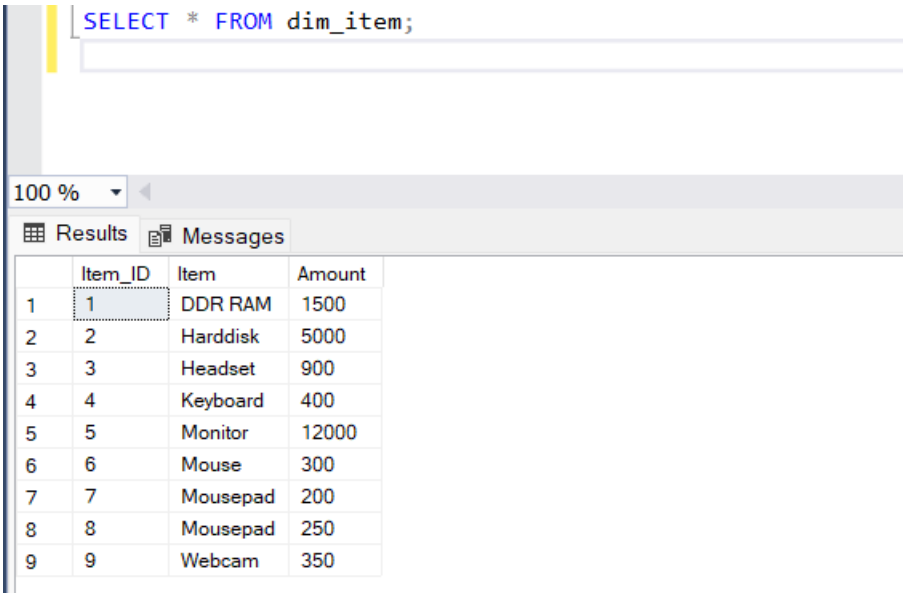
```
(250 rows affected)

Completion time: 2025-05-30T15:19:48.4074400+05:30
```

	Customer_ID	FirstName	LastName	Age	Country	AgeCategory
1	1	Joseph	Rice	43	USA	Above 30
2	2	Gary	Moore	71	USA	Above 30
3	3	John	Walker	44	UK	Above 30
4	4	Eric	Carter	38	UK	Above 30
5	5	William	Jackson	58	UAE	Above 30
6	6	N!cole	Jones	33	USA	Above 30
7	7	David	Davis	59	USA	Above 30
8	8	Jason	Montgomery	58	UK	Above 30
9	9	Kent	Weaver	61	UK	Above 30
10	10	Darrell	Dillon	50	UAE	Above 30
11	11	Jacqueline	Wang	22	USA	Less than 30
12	12	Jodi	Gonzalez	69	USA	Above 30
13	13	Omar	Martin	28	UK	Less than 30
14	14	N!cole	Lara	77	UK	Above 30
15	15	Jason	Brown	63	UAE	Above 30
16	16	David	Benson	61	USA	Above 30
17	17	Kimberly	Mora	34	USA	Above 30
18	18	Erik	Macias	25	UK	Less than 30
19	19	James	Johnson	67	UK	Above 30
20	20	Derek	Peterson	36	UAE	Above 30
21	21	Diane	Henson	74	USA	Above 30
22	22	Cody	Lyons	47	USA	Above 30
23	23	Margaret	Hardy	25	UK	Less than 30
24	24	Arthur	Hayes	75	UK	Above 30
25	25	Raymond	Taylor	36	UAE	Above 30
26	26	Tina	Moore	34	USA	Above 30

Query executed successfully. | LAPTOP-B4HM0U4O (15.0 RTM) | LAPTOP-B4HM0U4O\Shanta... | PEISalesDB | 00:00:00 | 250 rows

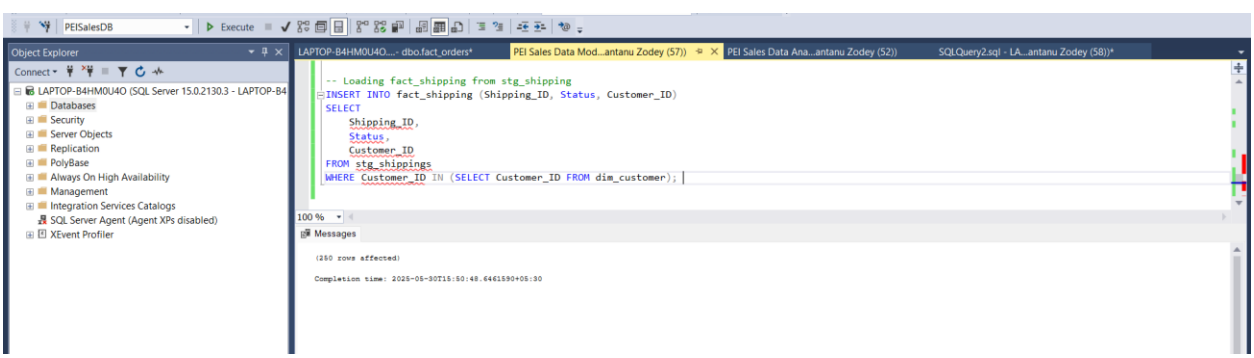
Loading Distinct Items and Amounts from stg_orders into dim_items.

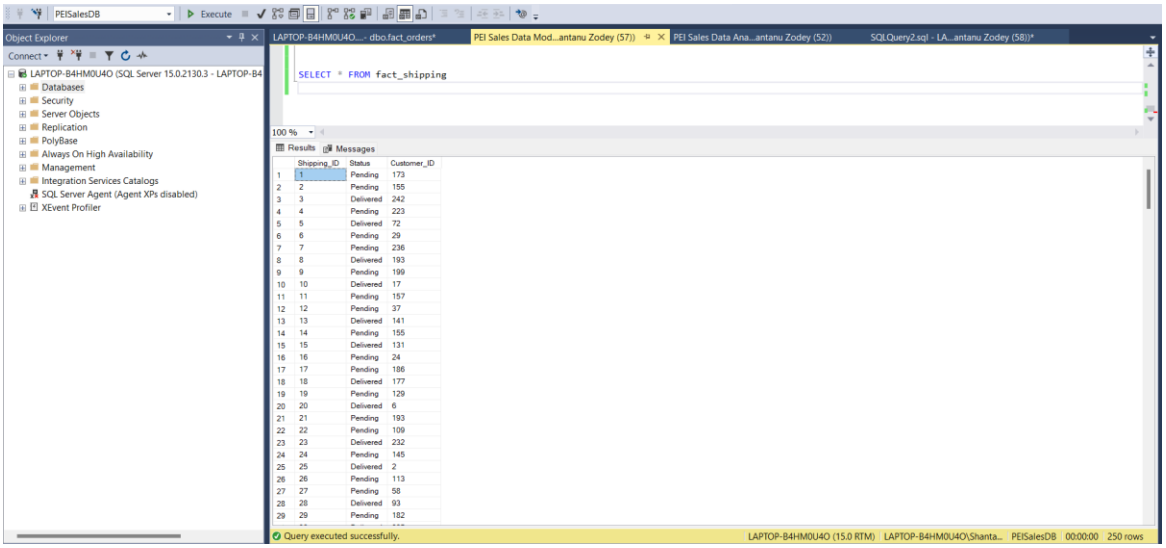




	Item_ID	Item	Amount
1	1	DDR RAM	1500
2	2	Harddisk	5000
3	3	Headset	900
4	4	Keyboard	400
5	5	Monitor	12000
6	6	Mouse	300
7	7	Mousepad	200
8	8	Mousepad	250
9	9	Webcam	350

Loading data from stg_shipping to fact_shipping.





Loading fact_order table.

To populate the fact_orders table for transactional reporting. A Data engineer should create a table in such a way that it captures individual product purchase events linked to customer and item dimensions.

In a way Business stakeholders can analyze total sales, customer purchasing behavior, and product performance across various dimensions such as country, age category, and product type.

<u>Business Context</u>

The fact_orders table will store transactional-level data based on raw order records from the sales team. Each record in the raw orders CSV contains an Order_ID, Item, Amount, and CustomerID. The goal is to link this to:

- A **product dimension (dim_product)** for consistent product representation.
- A **customer dimension (dim_customer)** to associate orders with customer demographics.

This table supports reporting on:

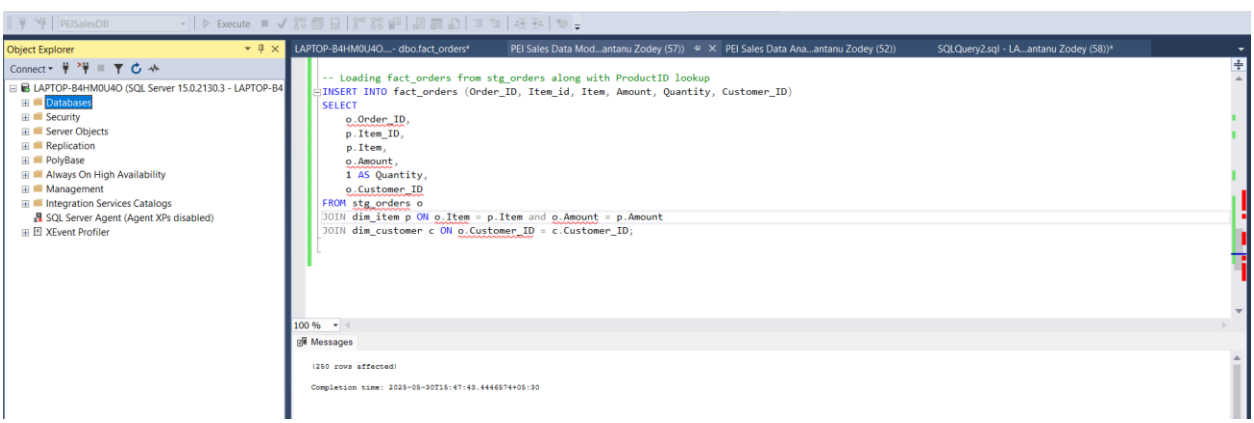
- Total sales by customer
- Product purchases by age category
- Quantity sold per item

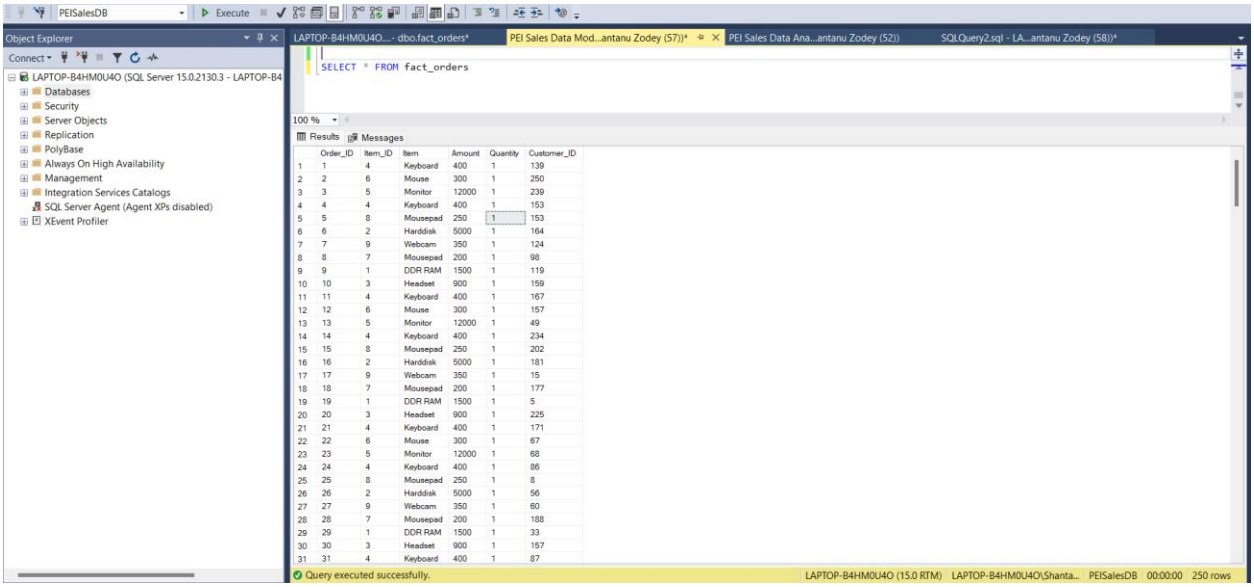
Transformations

```sql
SELECT
    o.Order_ID,
    p.Item_ID,
    p.Item,
    o.Amount,
    1 AS Quantity,
    o.Customer_ID
FROM stg_orders o
JOIN dim_item p ON o.Item = p.Item and o.Amount = p.Amount
JOIN dim_customer c ON o.Customer_ID = c.Customer_ID;
```

- Join with dim_item on Item to obtain Item_ID.
- Validate Customer_ID exists in dim_customer.
- Derive Quantity: Hardcoded to 1 per transaction.
- Exclude null or malformed records in Order_ID, Item, Amount, or CustomerID.

Implementation



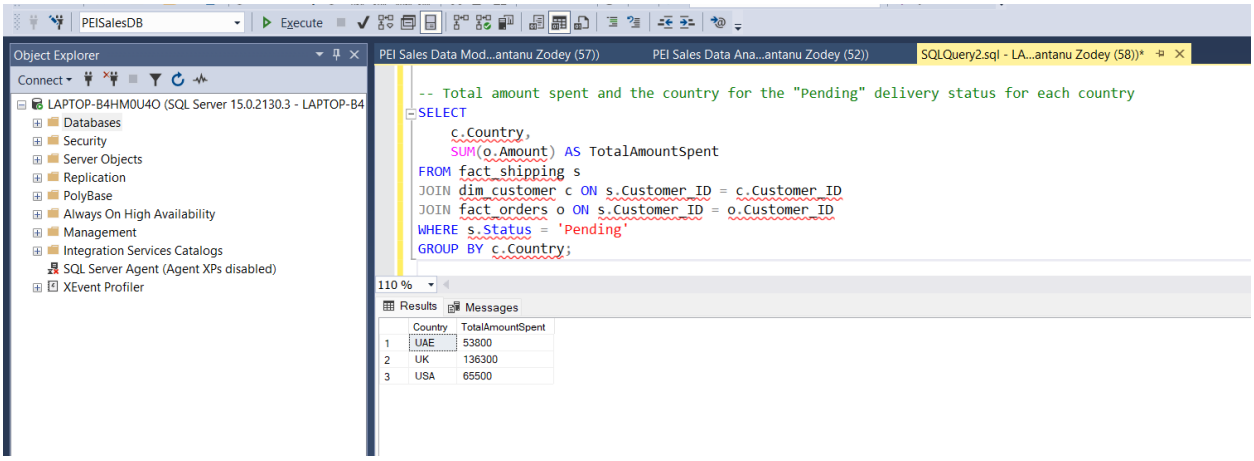


QA Test Criteria

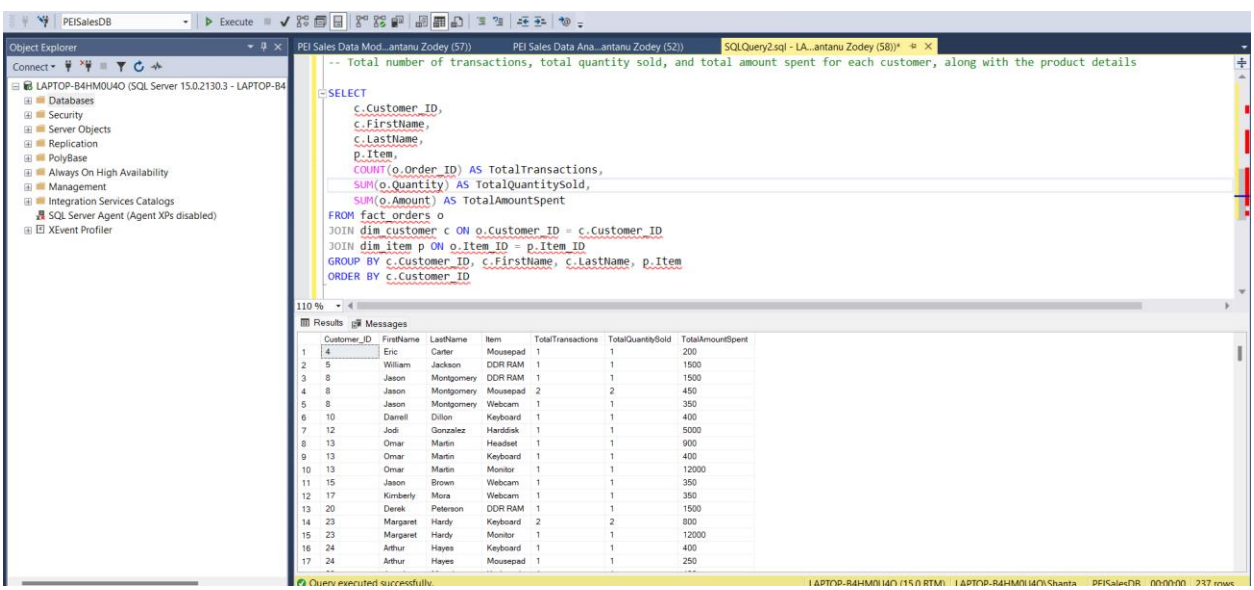
Test Case #	Description	Expected Outcome
TC1	Validate Order_ID uniqueness	No duplicates in fact_orders.Order_ID
TC2	Verify Item_ID and Customer_ID exist in dimension	All values must match dimension records
TC3	Validate Amount is not null and > 0	All records must pass
TC4	Ensure Quantity is always 1	No deviation
TC5	Join test with dim_Item and dim_customer	Ensure joins produce expected results

Enable Reporting Requirements

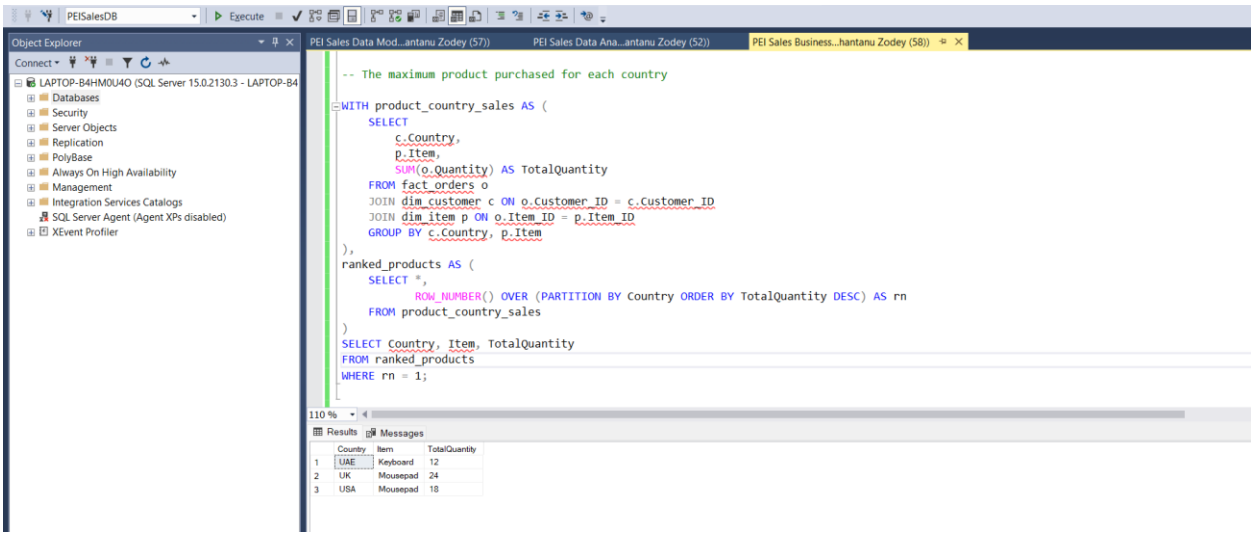
1. Total amount spent and the country for the "Pending" delivery status for each country



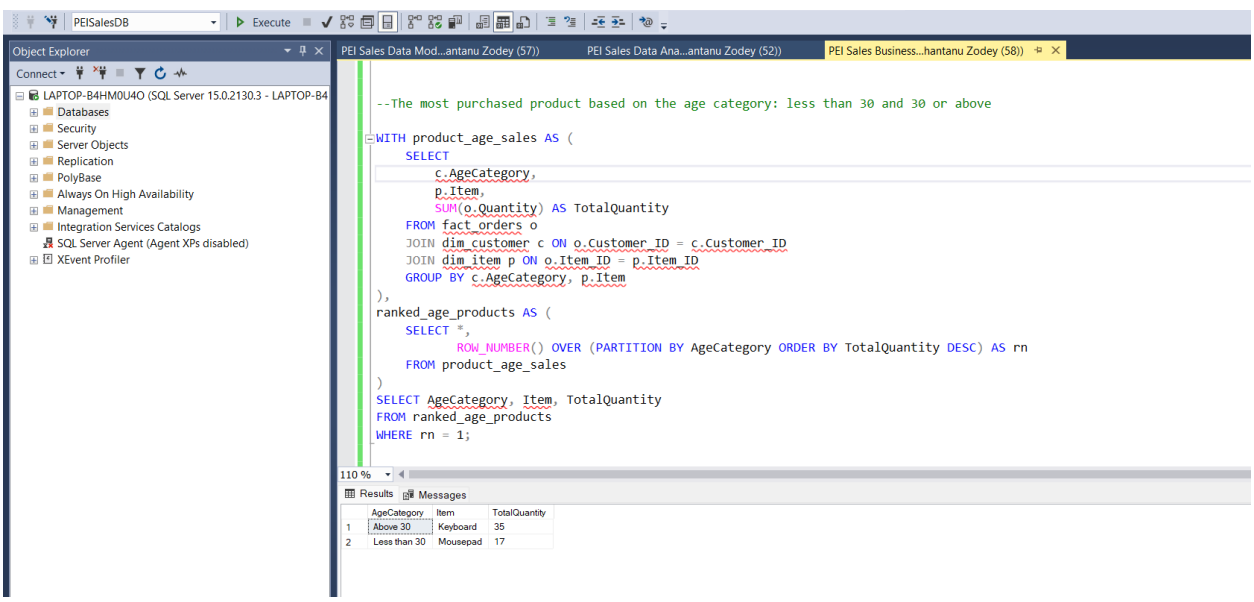
2. Total number of transactions, total quantity sold, and total amount spent for each customer, along with the product details



3. The maximum product purchased for each country



4. The most purchased product based on the age category: less than 30 and 30 or above



5. The country that had the minimum transactions and sales amount