Technical Specification Story for Data Engineers and QA Testers

Author - Shantanu Zodey

Date – 30/05/2025

Technical Story: Building dim_customer Table

1. Objective

Building a clean and structured Customer Dimension Table (dim_customer) from the raw source to support customer-level reporting and improve the fact tables with descriptive attributes (such as age category and full name).
This table will support analytical queries like:

- Spend by age group or country.
- Customer segmentation.
- Drilldown analysis for orders and shipping.

2. Source Details

File Format: .xlsx

Field	Description	Source	Notes
CustomerID	Unique identifier for customer	Excel	Primary Key
FirstName	First name of the customer	Excel	May contain invalid characters
LastName	Last name of the customer	Excel	May contain invalid characters
Age	Age of the customer	Excel	Should be numeric, clean nulls
Country	Country of residence	Excel	Standardize casing

Table Name: stg_customer (staging table created from Excel data)

Column Name	Data Type	Description	Constraints
CustomerID	INT	Unique identifier for each customer	Nullable
FirstName	NVARCHAR(100)	Raw first name from source (may contain special chars)	Nullable
LastName	NVARCHAR(100)	Raw last name from source (may contain special chars)	Nullable
Age	NVARCHAR(10)	Raw age value from source (may contain invalid chars)	Nullable
Country	NVARCHAR(100)	Raw country name from source	Nullable

The above **stg_customer** is a staging table that will hold the raw data extracted from the Excel file. It will act as a temporary landing zone before data is validated, cleaned, and transformed into the **dim_customer** table.

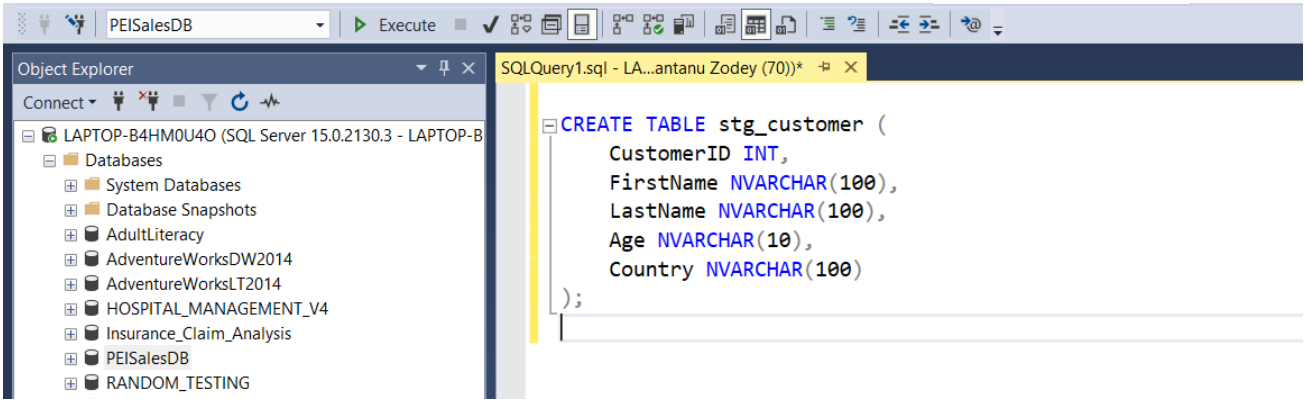
3. Proposed Target Table Definition as per Data Model and Business Reporting - dim_customer

Column Name	Data Type	Description	Constraints
CustomerID	INT	Unique identifier for each customer	Primary Key, Not Null
FirstName	NVARCHAR(100)	Customer’s first name (cleaned of special characters)	Not Null
LastName	NVARCHAR(100)	Customer’s last name (cleaned of special characters)	Not Null
FullName	NVARCHAR(200)	Concatenated first and last name	Computed/Derived field
Age	INT	Age of the customer (validated and cleaned)	Nullable (if unknown or invalid)
AgeCategory	VARCHAR(20)	Categorized as 'Young', 'Adult', or 'Senior'	Derived from Age
Country	VARCHAR(100)	Country name, standardized to uppercase	Not Null
CreatedDate	DATETIME	Timestamp of when the record was created	Default: GETDATE()

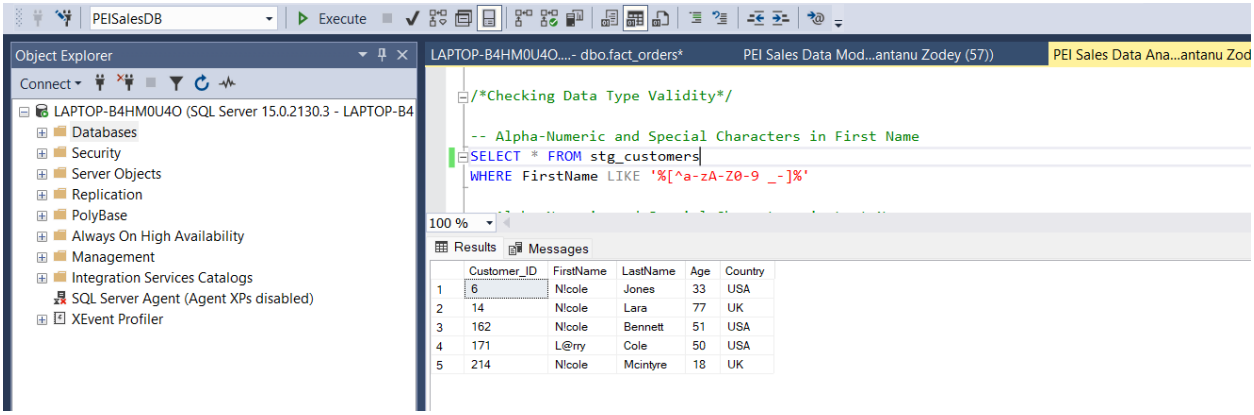
4. Transformation Logic (For Data Engineer)

Step-by-step Processing Instructions:

Step 1: Load Raw Data to Staging Table

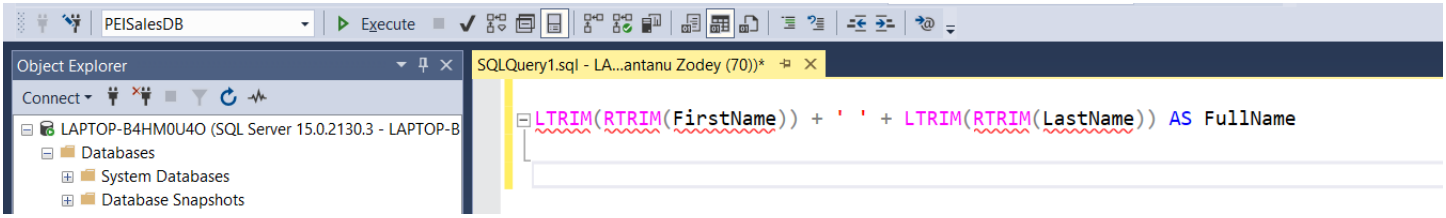


Step 2: Clean FirstName and LastName

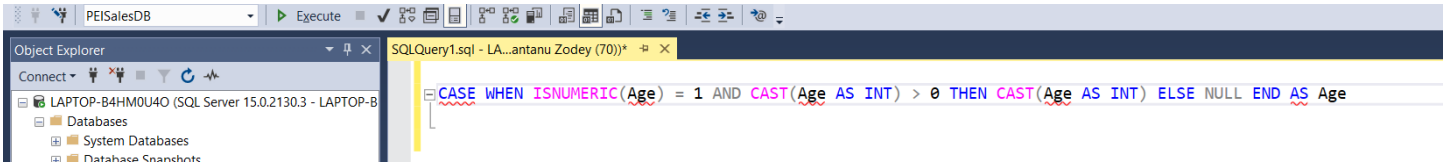


- Based on the above screenshot, there are special characters in First Name and Last name.
- Remove special characters using appropriate SQL logic or UDF.

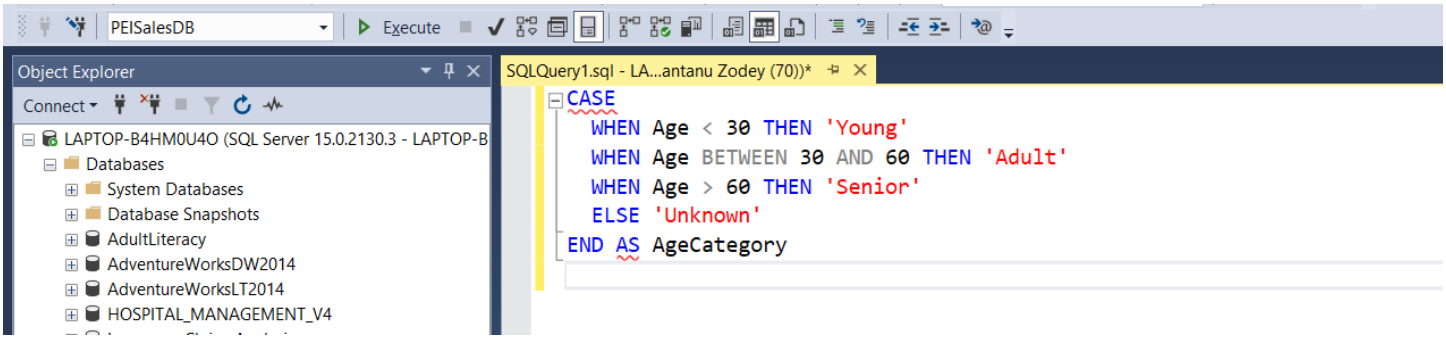
Step 3: Derive FullName



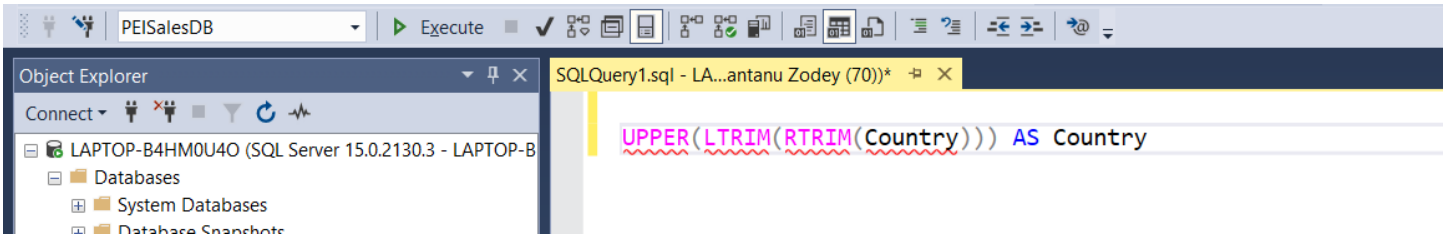
Step 4: Clean and Validate Age



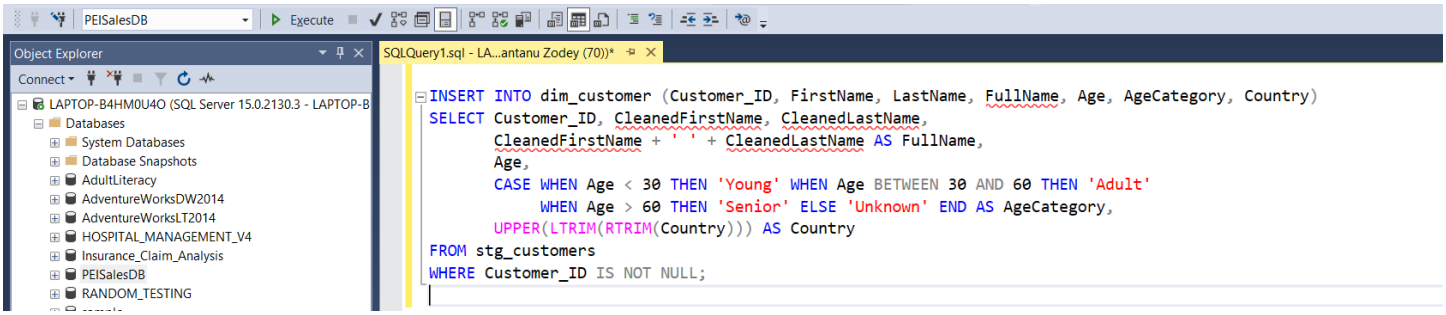
Step 5: Derive AgeCategory



Step 6: Standardize Country



Step 7: Insert into dim_customer



5. QA Testing Checklist (For QA Testers)

Test Category	Test Description	Expected Outcome / Criteria
Data Load Checks	Verify that all records from stg_customer are loaded into dim_customer.	Row counts should match between valid staging records and dimension table.
	Ensure no duplicate CustomerID values exist.	All CustomerID values must be unique.
Data Cleaning	Special characters are removed from FirstName and LastName.	Only alphabets and spaces should be present.
	Extra spaces are trimmed in name and country fields.	No leading/trailing or double spaces.
	Nulls in FirstName or LastName handled correctly.	Either replaced with placeholder or record excluded/logged.
Data Transformation	Verify FullName is correctly concatenated from cleaned names.	FullName = CleanFirstName + ' ' + CleanLastName
	Age is correctly cast to INT.	Invalid ages are null or flagged.
	AgeCategory is correctly derived based on age.	Valid category values only: Young, Adult, Senior, or NULL.
	Country names are uppercased and standardized.	e.g., "India", "india", "INdia" → "INDIA"
Integrity Checks	Each CustomerID in dim_customer is unique.	No duplicates allowed.
	All fields have appropriate data types.	Check against schema definition.
	Referential integrity maintained with fact_orders and fact_shipping (foreign keys).	All foreign keys should match existing CustomerIDs in dim_customer.
Null / Missing Data	Verify handling of nulls in source data for all columns.	Null handling is as per transformation logic.
	Ensure CreatedDate is populated.	No null values, defaulted to GETDATE().
Audit Logging (Optional)	Check whether data load audit logs are captured (record count, load time, errors).	Audit table/log file should contain expected entries.
Performance Checks	Check indexes on CustomerID or other key lookup columns.	Proper indexing improves query performance.

6. Reporting Use Cases Enabled

- Customer-level analysis (by country, age group)
- Segmentation of transactions by demographics
- Joins with fact_orders and dim_shipping using CustomerID