Sri Lanka Institute of Information Technology

Data Warehousing and Business Intelligence

IT3021

Assignment 1

2025

Assignment 1 Report

Student Name – Karunaarachchi K A I H

IT Number – IT22022974

Contents

1. Data Set Selection

1.1 Description

Dataset Selected – Global Fashion Retail Sales (click on the text to view the original dataset)

Description –

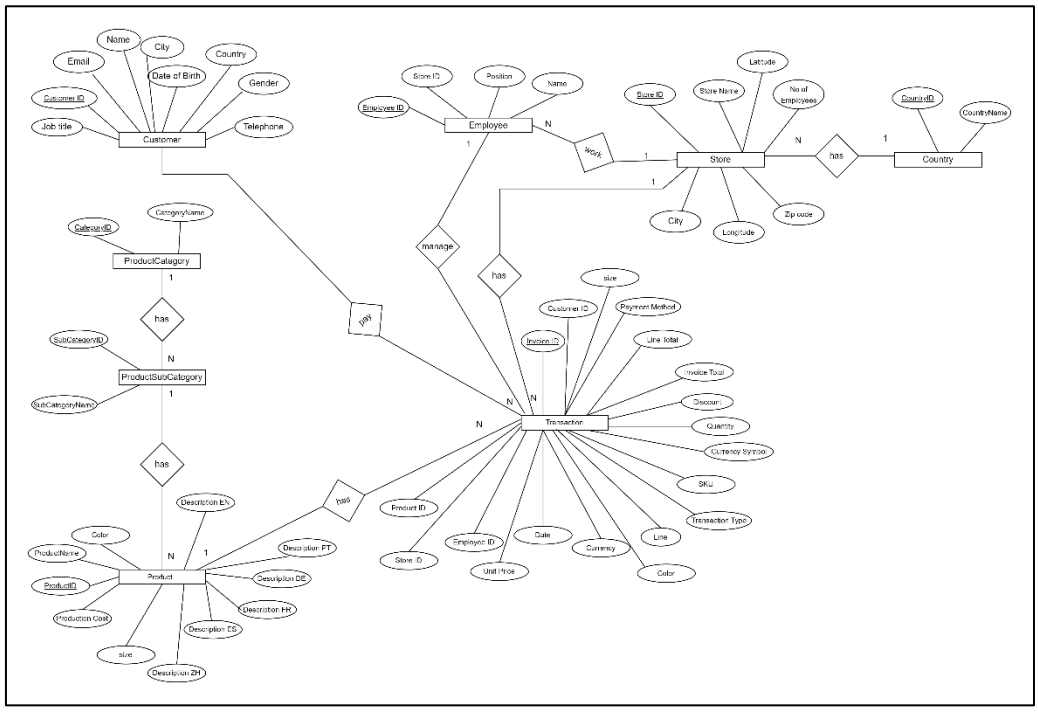
This dataset simulates two years (2023–2025) of sales transactions for a multinational fashion retail company. It contains over 4 million records from 35 stores across 7 countries, including the US, China, Germany, the UK, France, Spain, and Portugal. The data includes transactions, products, stores, customers, and employee details.

The original dataset has been edited, configured, and rearranged to meet the project requirements. A one-year subset of data from **01-01-2024 to 01-01-2025** was selected for this assignment to ensure consistency and manageability. Based on the scenario, **se**ven main tables were identified and structured for use in the data warehouse design:

- **Country -** contains information about the Country name
- **Store** – contains information about each retail store across multiple countries.
- **Product Category** – includes the main product categories offered by the retailer.
- **Product Subcategory** – provides more specific classifications within each main category.
- **Product** – holds details of the products available for sale.
- **Customer** – includes records of customers who made purchases.
- **Employee** – contains details about staff members and their work locations.
- **Transaction** – represents the sales activities and purchasing history within the selected period.

1.2 ER Diagram

.



2. Preparation of the Data Sources

Initially, the original three data files were provided in CSV format. These files were then downloaded and separated into eight tables, distributed across five different files, each saved in various formats. Three types of data sources were utilized: CSV, TXT, and Database.

1. **TXT Files**
   The Country and Customer Location data were stored as TXT files:
- Country.txt
- Customer_Location.txt

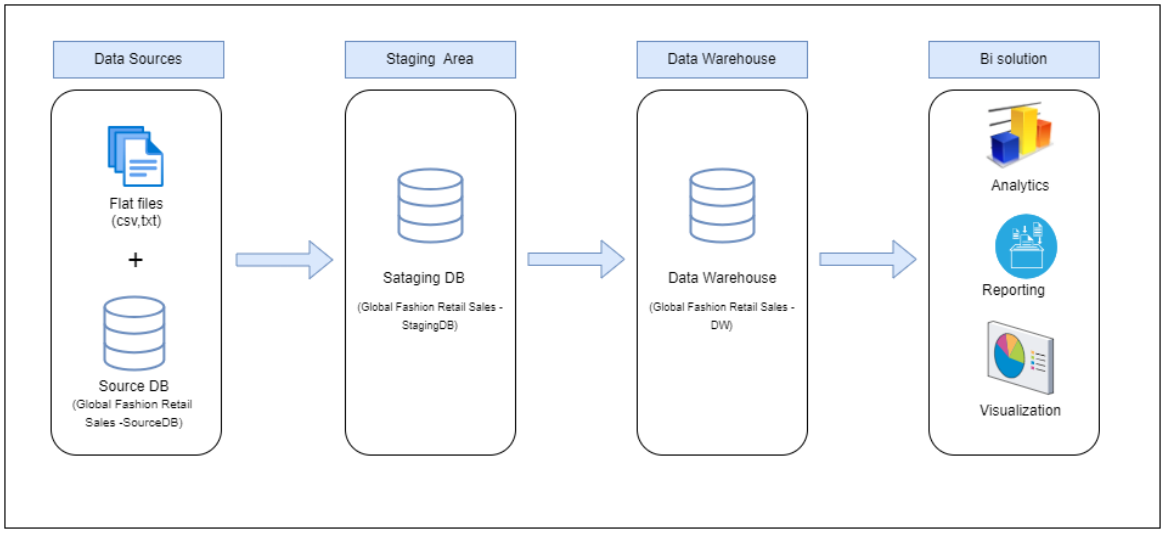
2. **CSV File**
   Stores data was stored in a CSV file:
- Stores.csv

3. **Database**
   A source database, named **Global Fashion Retail Sales - SourceDB**, was created by importing the following CSV files:
- customers.csv
- employees.csv
- ProductCategory.csv
- ProductSubCategory.csv
- Product.csv
- transaction.csv

The imported files populated the following tables in the SourceDB:

- customers
- employees
- ProductCategory
- ProductSubCategory
- Product
- transaction

3. Solution Architecture



1. Data Sources

The first step is to collect data from different sources, such as databases, files, or APIs. For this project, there are two types of data sources:

- **Source database** (Global_Fashion_Retail_Sales_SourceDB) with tables like customers, employees, ProductCategory, ProductSubCategory, Product, and transactions.
- **Flat files**:
  - **CSV file** (Stores.csv) with store details.
  - **TXT files** (Country.txt, Customer_Location.txt) with country and customer location data.

2. Staging Area

A data staging area is a temporary storage space where data is placed before it moves into the data warehouse. It helps quickly get data from sources without affecting them. In this project, **Global_Fashion_Retail_Sales_Staging** is used as the staging area, storing data exactly as it comes from the sources, without any changes.

3. Data Warehouse (DW)

A data warehouse is a large storage system that holds business data, helping with analysis and decision-making. It collects and organizes historical data from various sources into one place. In this project, **Global_Fashion_Retail_Sales_DW** is the data warehouse, which consolidates and stores structured data ready for analysis.

4. BI Solution

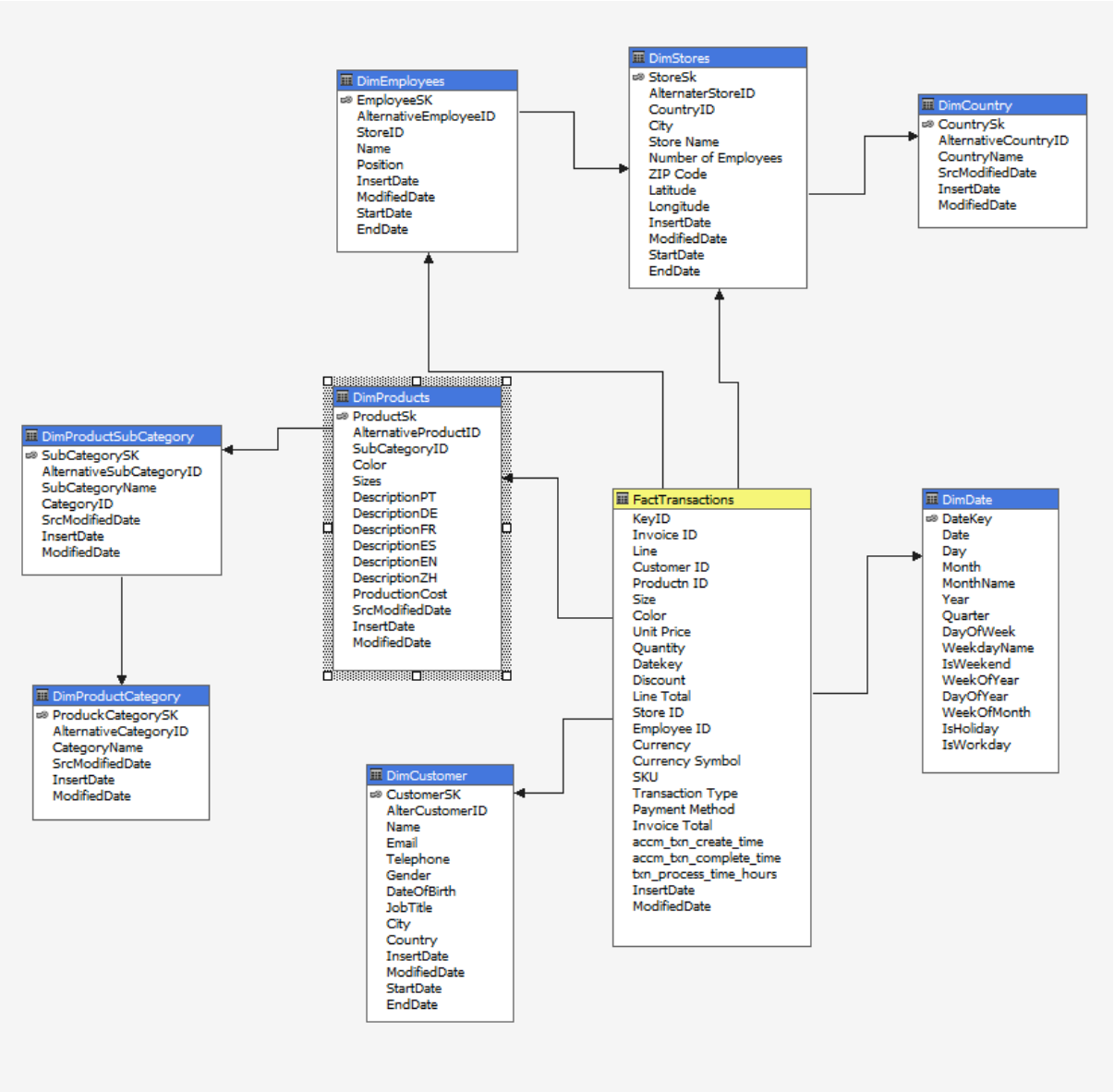
The BI layer uses tools to turn data into insights, helping businesses make better decisions. In this case, business users can use dashboards and reports to analyze sales, product trends, and customer behavior.

5. ETL Process

ETL (Extract, Transform, Load) is a process that moves data from different sources into a consistent storage system.
For this project:

- **Extract**: Data is pulled from the source database and flat files.
- **Load to Staging**: Data is placed in the staging area without changes.
- **Transform**: Data is cleaned and organized in the ETL process.
- **Load to DW**: The cleaned data is moved into the data warehouse for analysis.

4. Data warehouse design & development



The above is the dimensional model used for the given scenario. In summary, the dimensional model is designed with 8-dimensional tables (including the date dimension) and a single fact table.

Schema Used – Snowflake Schema

A **snowflake schema** has been utilized in dimensional modelling to reduce redundancy through normalization.

- The store dimension has been normalized by separating country information into a distinct Country dimension.
- This approach minimizes redundancy and allows better regional reporting and analysis.
- The product dimension has been normalized by separating subcategories and categories into ProductCategory and ProductSubcategory tables.

This allows for better hierarchy management and enables detailed product-level analysis across different levels (Category > SubCategory > Product).

Dimension and Fact Tables

Eight-dimensional tables and a fact table were created:

1. DimCustomer – Contains customer details such as name, email, telephone, and city. CustomerSK is the surrogate key.
2. DimEmployees – Contains employee details such as name, job position, and store association. EmployeeSK is the surrogate key.
3. DimStores – Contains store details such as city, ZIP code, number of employees, and location (latitude/longitude).StoreSK is the surrogate key.
4. DimProducts – Contains product details such as size, color, and multilingual product descriptions. ProductSK is the surrogate key
5. DimProductSubCategory – Contains subcategory details linked to each product. SubCategorySK is the surrogate key.
6. DimProductCategory – Contains product category details, linked to subcategories. ProductCategorySK is the surrogate key.
7. DimCountry – Contains country information to support regional analysis. CountrySK is the surrogate key.
8. DimDate – A standard date dimension containing attributes such as day, month, year, quarter, week, holiday flag, and working day flag.
    - DateKey is the surrogate key.
    - An SQL script was used to generate the date dimension.
9. FactTransactions – Contains all transactional sales data and references to dimension tables.
    - Important fields include: InvoiceID, Line, ProductID, CustomerID, StoreID, EmployeeID, DateKey, Size, Color, Quantity, Unit Price, Line Total, Payment Method, and Currency Symbol.

Slowly Changing Dimensions (SCDs)

It was assumed that customer, employee, and store attributes could change over time. Therefore, the following dimensions were considered as **Slowly Changing Dimensions (SCDs):**

DimCustomer – Historical attributes:

- City, Country
  Changing attributes:
- Telephone, Email, Gender

DimEmployees – Historical attribute:

- Position
  Changing attributes:
- StoreID, StartDate, EndDate

DimStores – Historical attributes:

- City, ZIP Code, Country
  Changing attributes:
- Store Name, Number of Employees

By tracking changes in these dimensions, historical analysis can be conducted more accurately, e.g., analyzing sales performance before and after a store relocation or employee transfer.

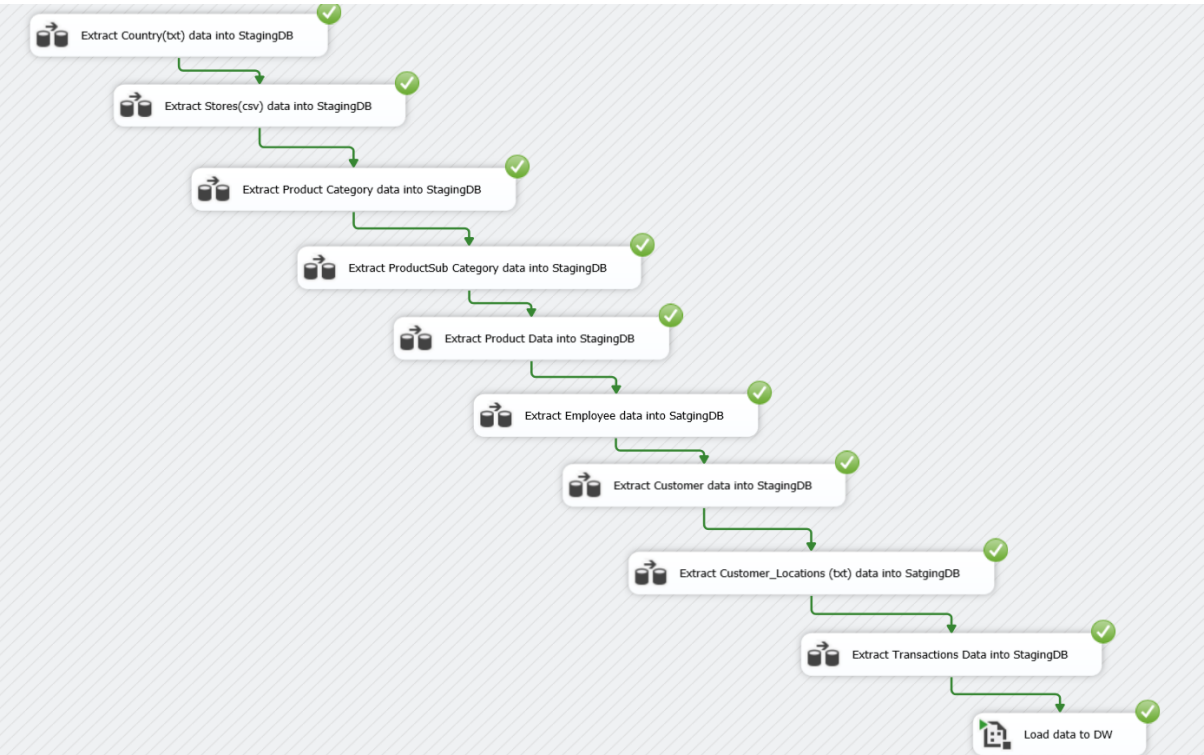
Assumptions

- It was assumed that stores, employees, and customers' information could change over time. Therefore, these dimension tables were designed as Slowly Changing Dimensions (SCDs) to track historical changes.
- Address-related and demographic details (for stores and customers) were considered important for future regional analysis.
- Each employee is assigned to a single store at a time.
- Products are assumed to have relatively static descriptions and characteristics and are not tracked historically.

5. ETL Development

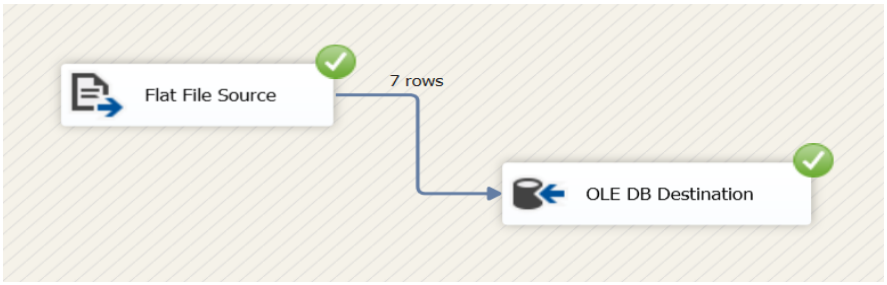
5.1 Extract Data from Source to Staging

• Order of Execution

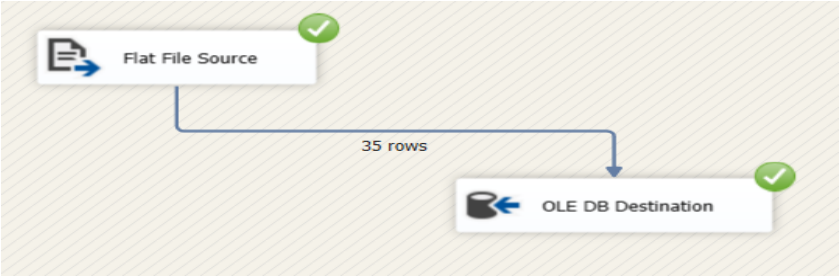


Data is extracted from various data sources and is staged in an intermediate location until being loaded into the data warehouse. Individual extractions into the staging database happen as shown in the images below:

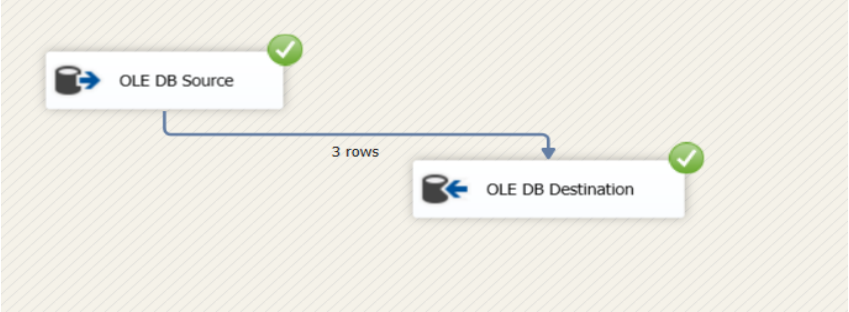
- Country data is extracted from a flat file source and loaded into the StgCountry table in the staging area.



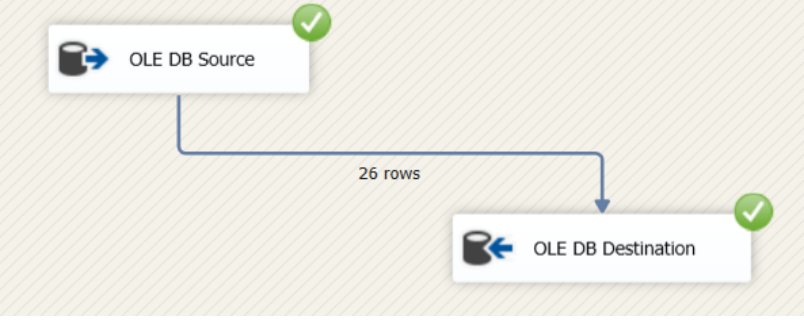
- Store data is extracted from a flat file source and loaded into the StgStores table in the staging area.



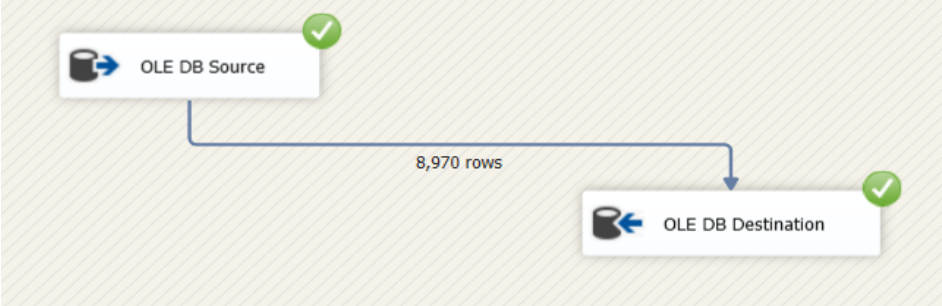
- Product Category data is extracted from a flat file source and loaded into the StgProductCategory table in the staging area.



- Product SubCategory data is extracted from a flat file source and loaded into the StgProductSubCategory table in the staging area.

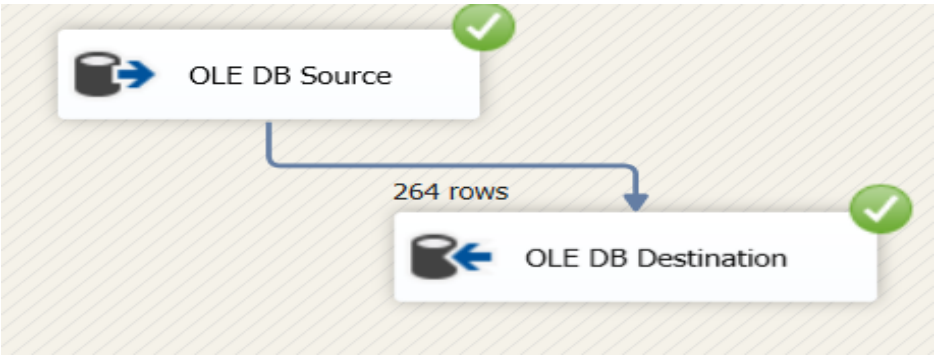


- Product data is extracted from a flat file source and loaded into the StgProducts table in the staging area

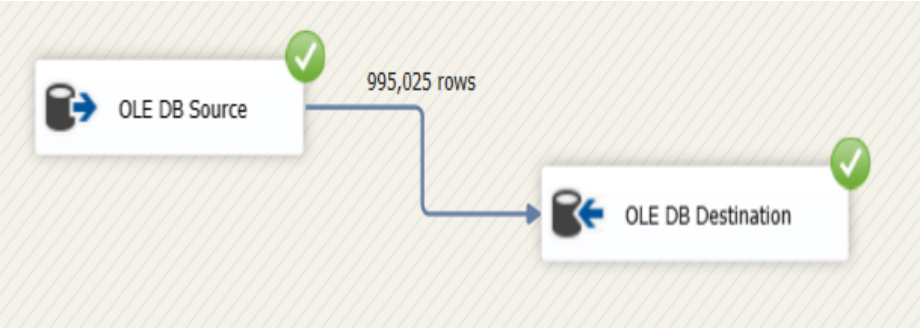


.

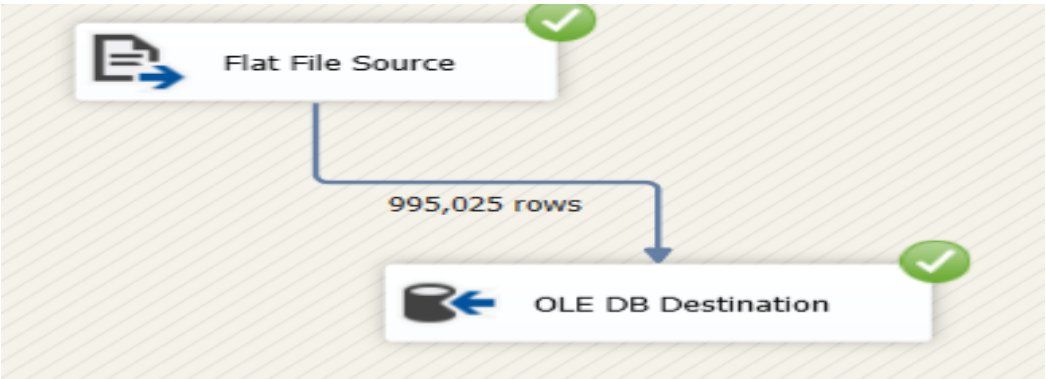
- Employee data is extracted from a flat file source and loaded into the StgEmployees table in the staging area.



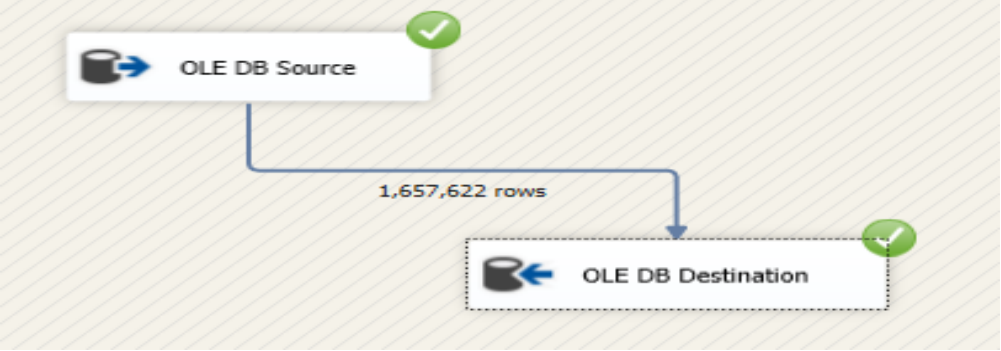
- Customer data is extracted from a flat file source and loaded into the StgCustomers table in the staging area.



- Customer Location data is extracted from a flat file source and loaded into the StgCustomerLocation table in the staging area.



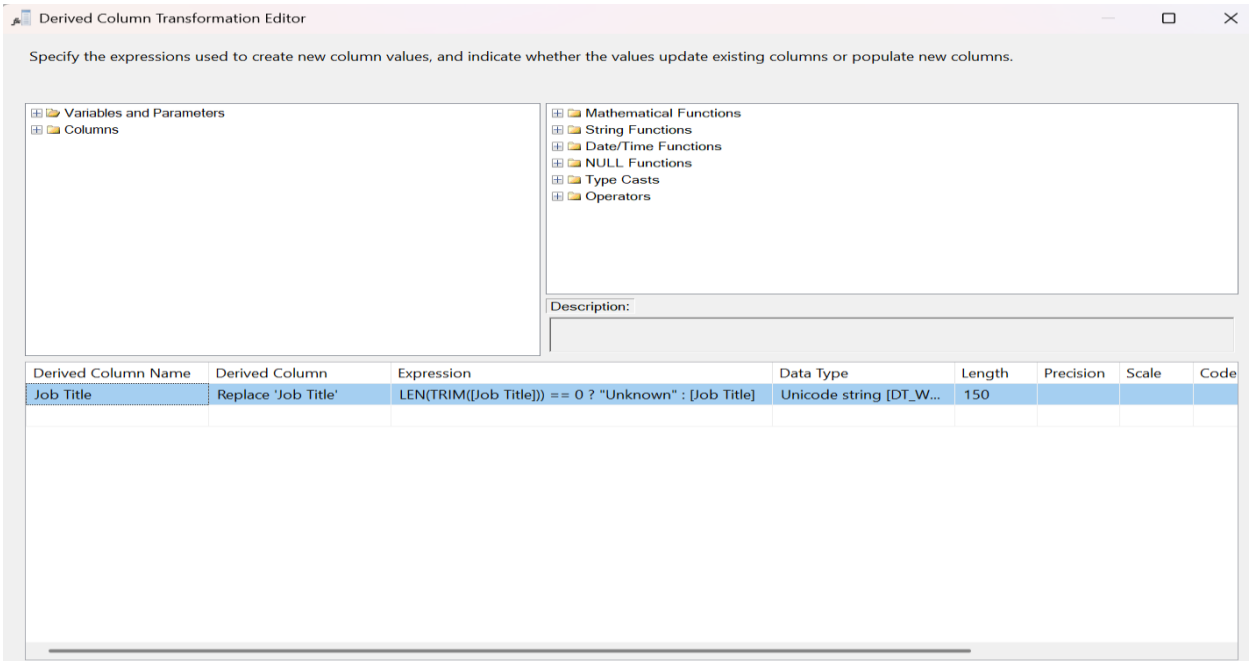
- Transaction data is extracted from a flat file source and loaded into the **StgTransactions** table in the staging area.

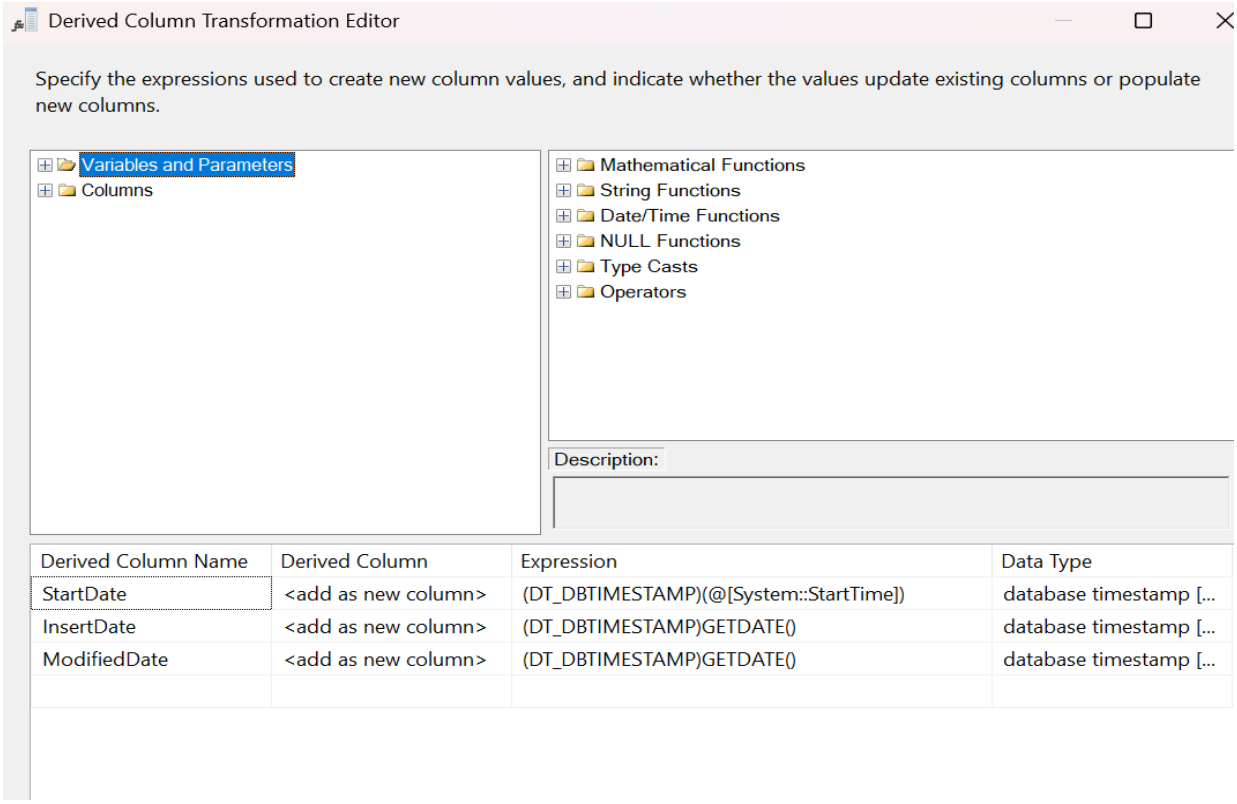


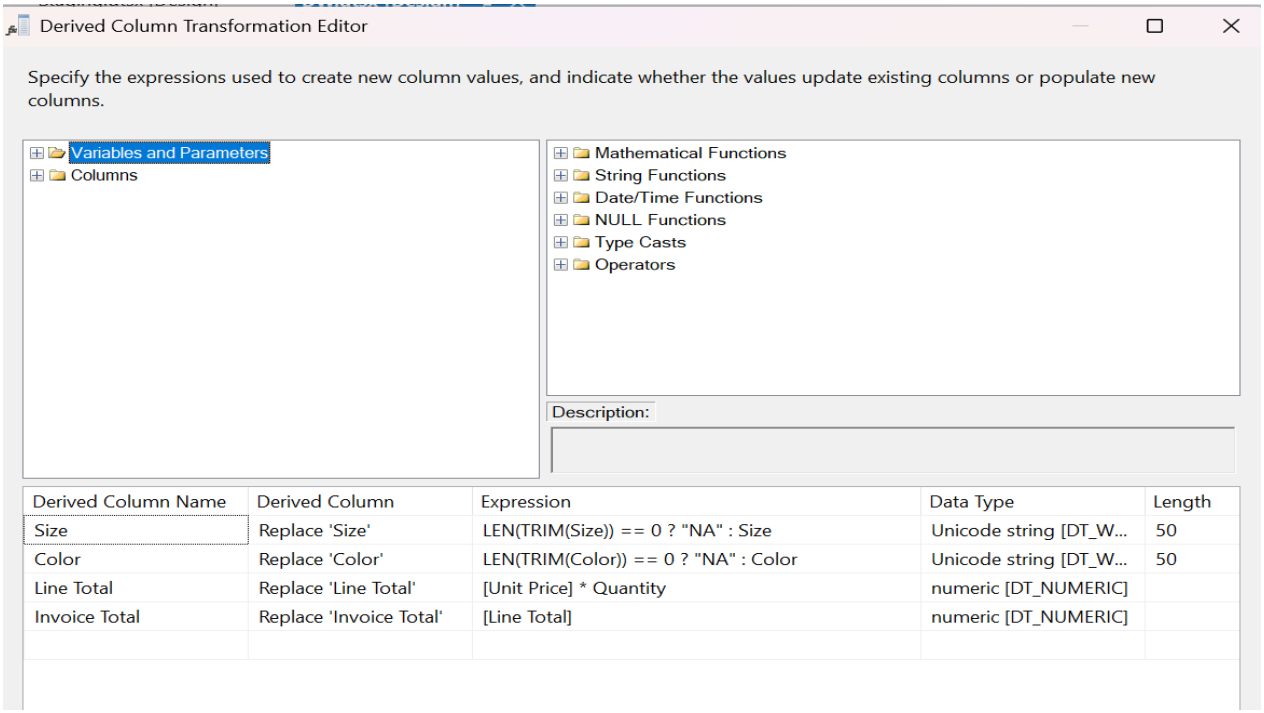
5.2 Transforming the Staged Data

The staged data undergoes several transformations before being loaded into the data warehouse, as outlined below:

- Handling NULL and Missing Values: Any NULL or missing values in the jobtitle, size, and color fields are identified and replaced with default or placeholder values where necessary to ensure data completeness.
- Insertion and Modification Tracking: During the transformation process, the system assigns insert_date and modified_date values to capture the current date and time, enabling proper tracking of record changes.





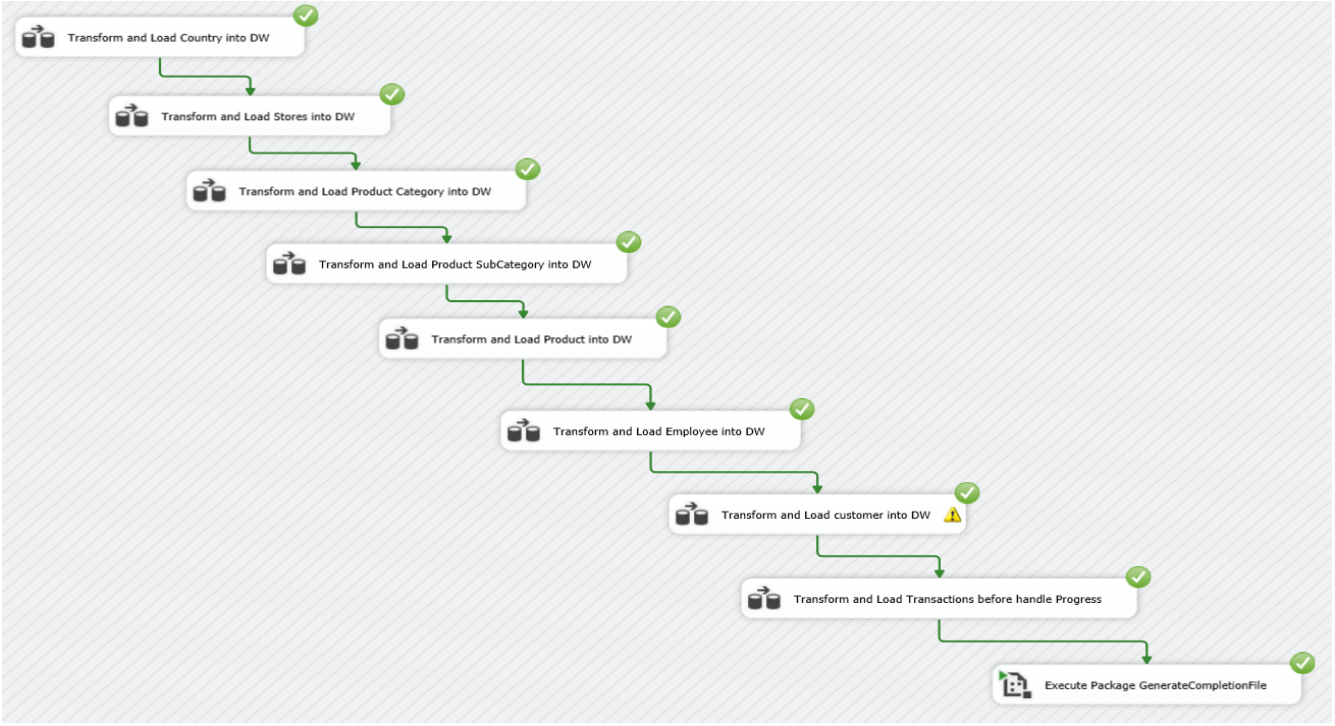


The staged data undergoes several transformations before being loaded into the data warehouse, as outlined below:

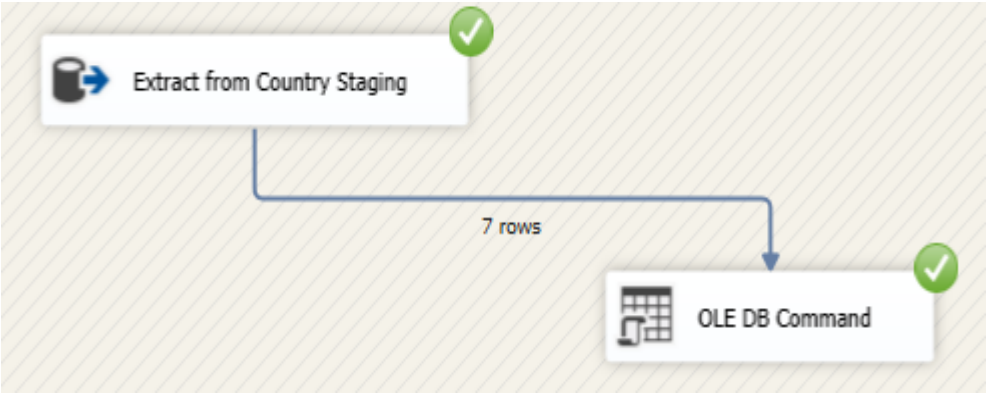
- Lookups for Foreign Key References: Lookups are performed to create foreign key relationships between tables and map the appropriate surrogate keys from the dimensional tables as foreign keys.
- Selection of Required Columns: Only the necessary columns are chosen during dimension creation. To ensure data quality, unwanted outliers and irrelevant data are filtered out through proper data filtering techniques.
- Use of Derived Columns: Derived columns are created to add insert_date and modified_date columns, capturing the system's current date and time during insertion and modification.
- Conditional Split: Conditional splits are applied to segment the data based on predefined criteria, ensuring that the data is processed and loaded accurately.
- Balance Data Distribution: The data is distributed evenly across the system to optimize performance and prevent any skewed data distribution.

5.3 Loading the Transformed Data into the Data Warehouse

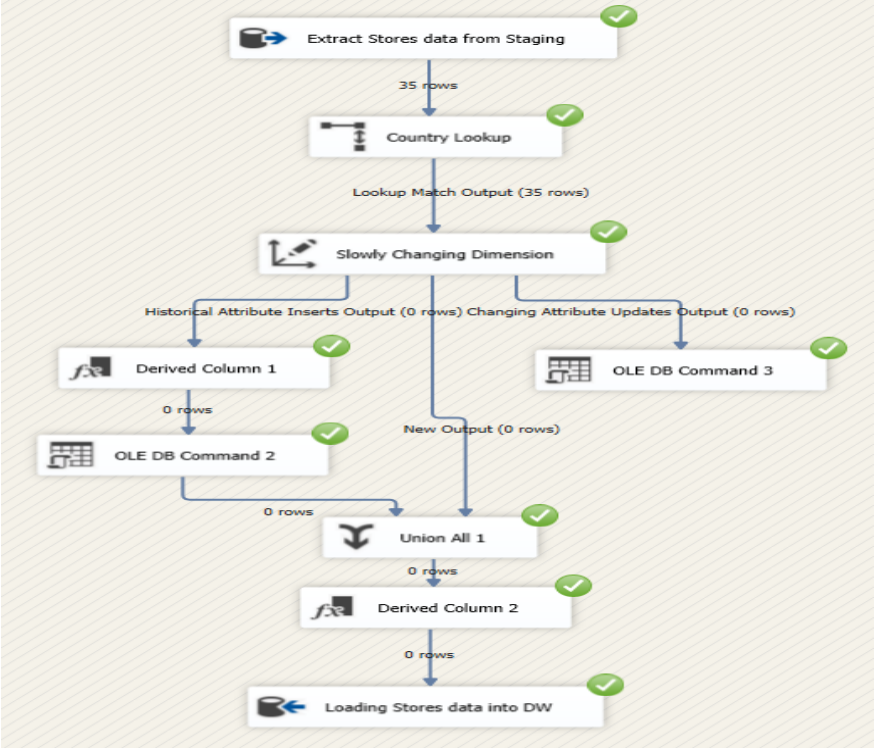
• Order of Execution



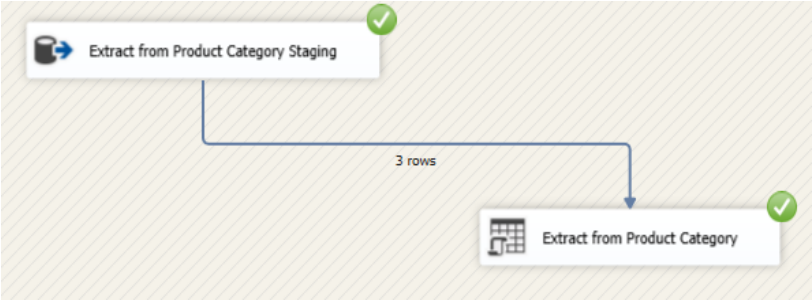
- The transformed Country data is loaded into the data warehouse.



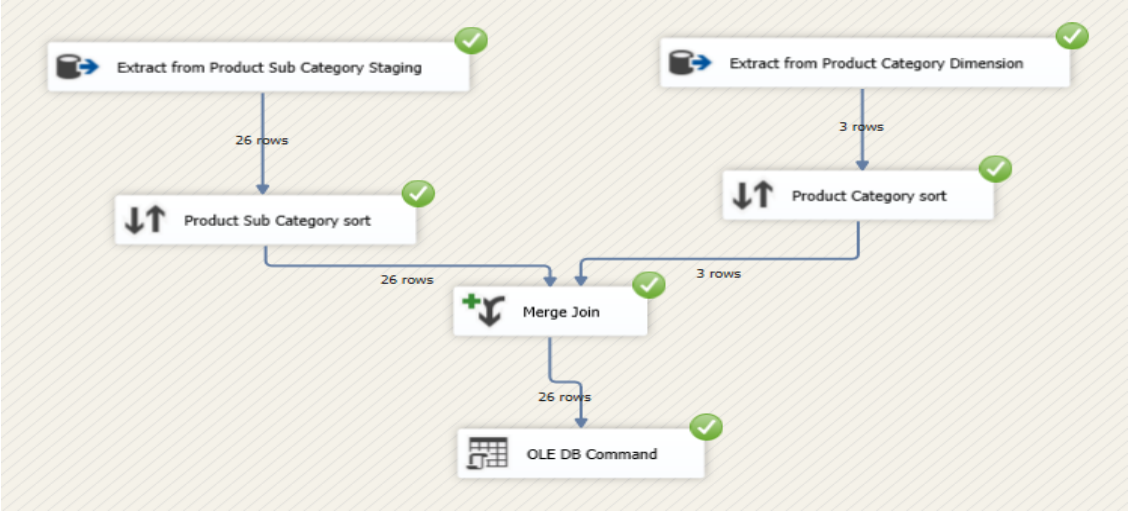
- The transformed Stores data is loaded into the data warehouse as a Slowly Changing Dimension.



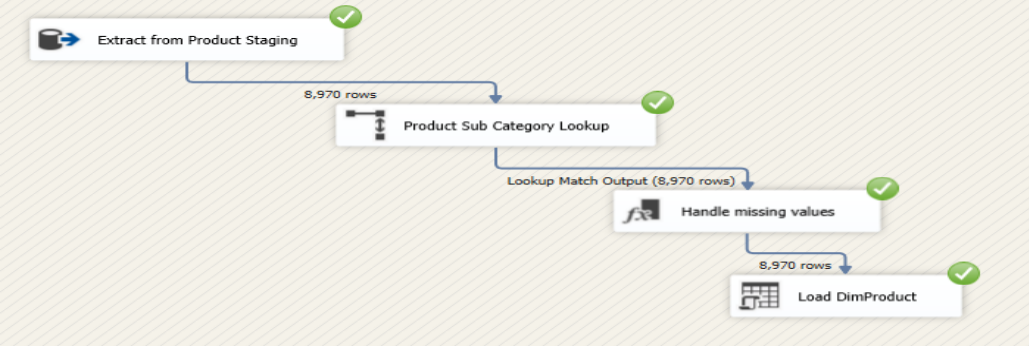
- The transformed ProductCategory data is loaded into the data warehouse.



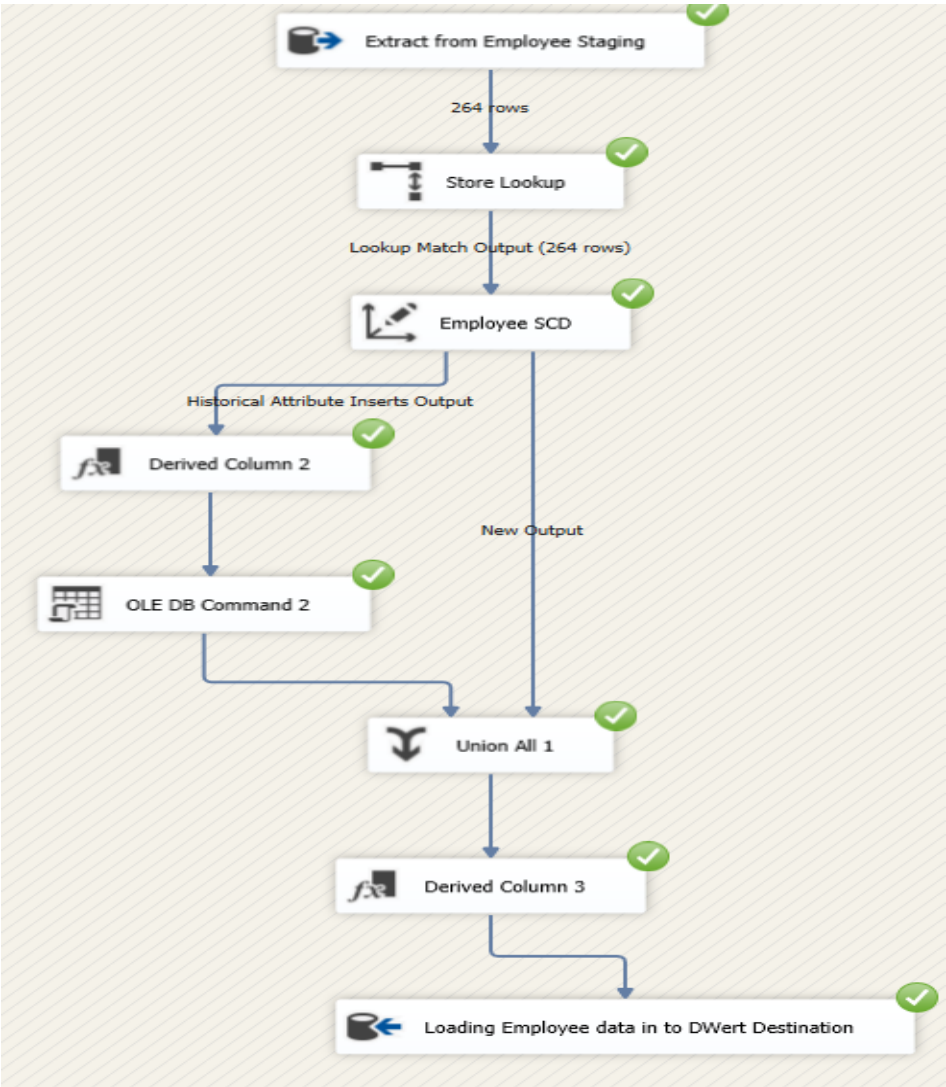
- The transformed ProductSubCategory data is loaded into the data warehouse.



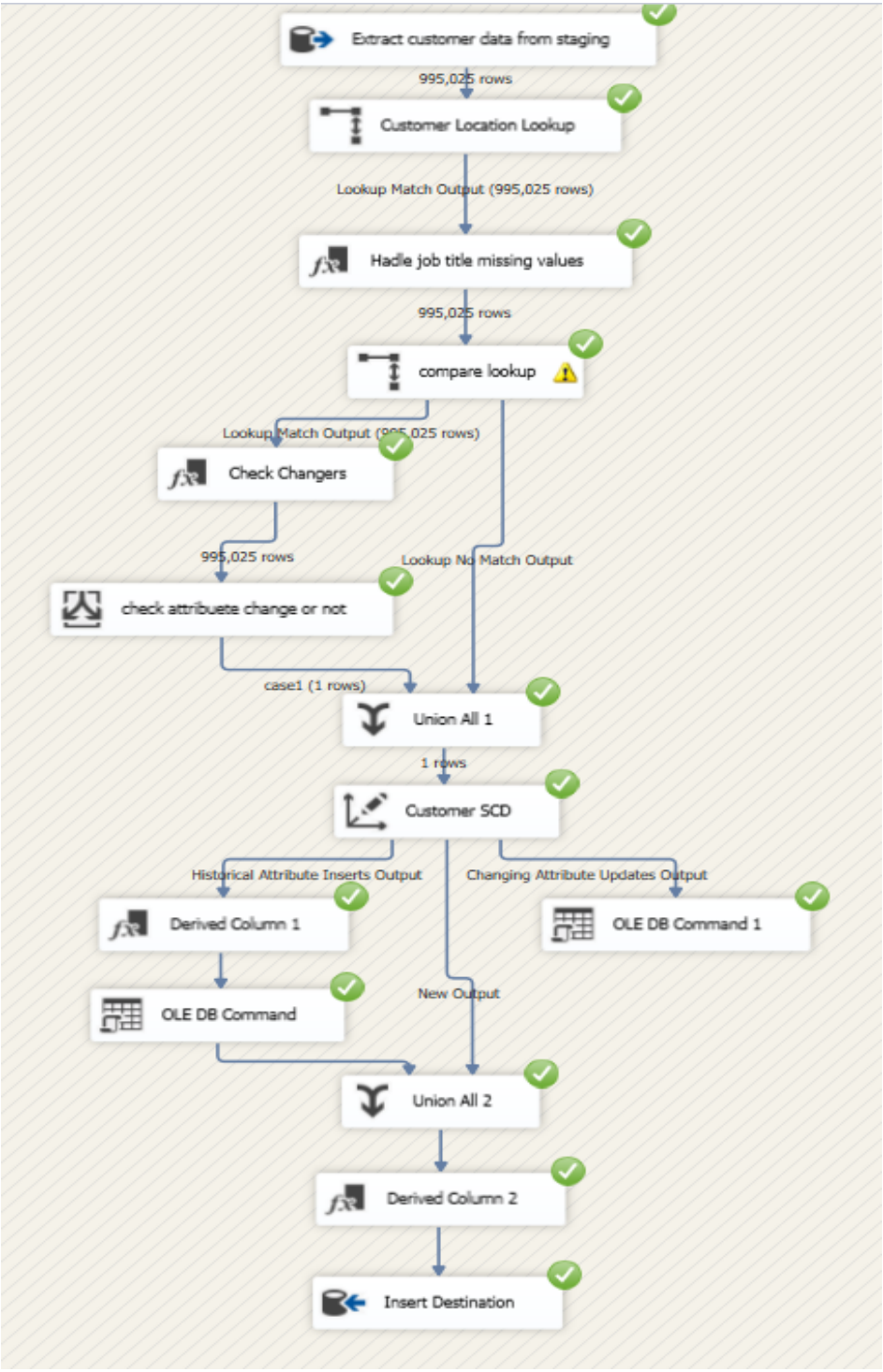
- The transformed Products data is loaded into the data warehouse.



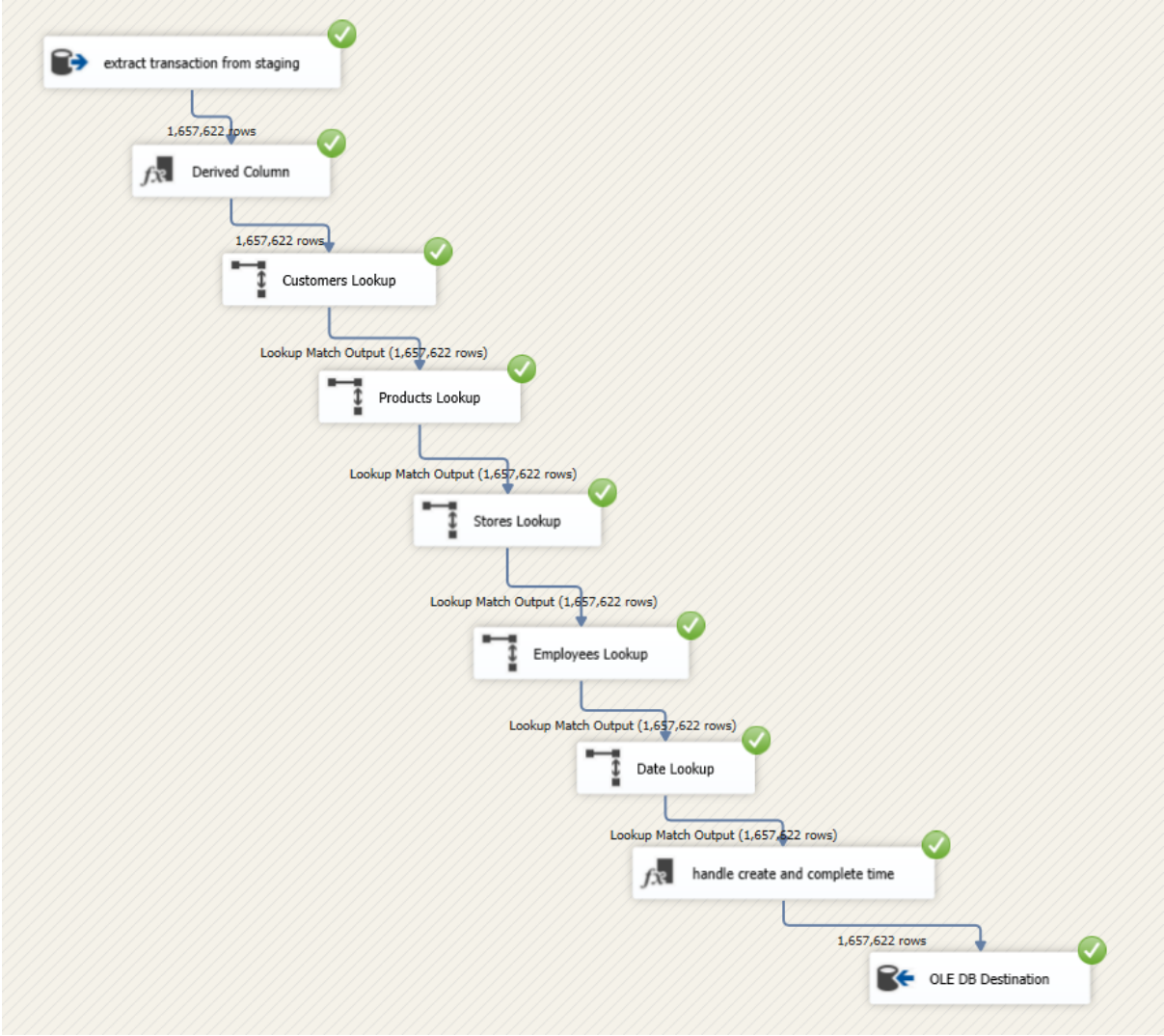
- The transformed Employee data is loaded into the data warehouse as a Slowly Changing Dimension.



- The transformed Customer and CustomerLocation data are combined into a single table and loaded as a Slowly Changing Dimension in the data warehouse.



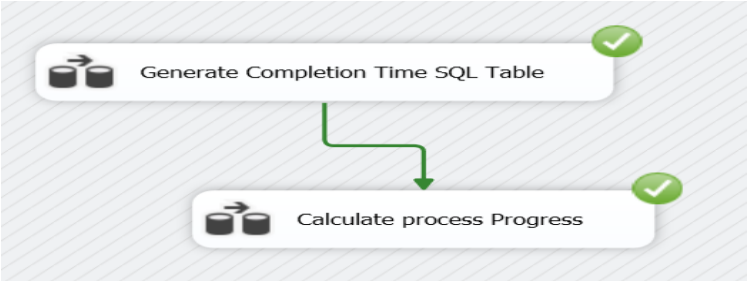
- The transformed Transactions data is loaded into the data warehouse.



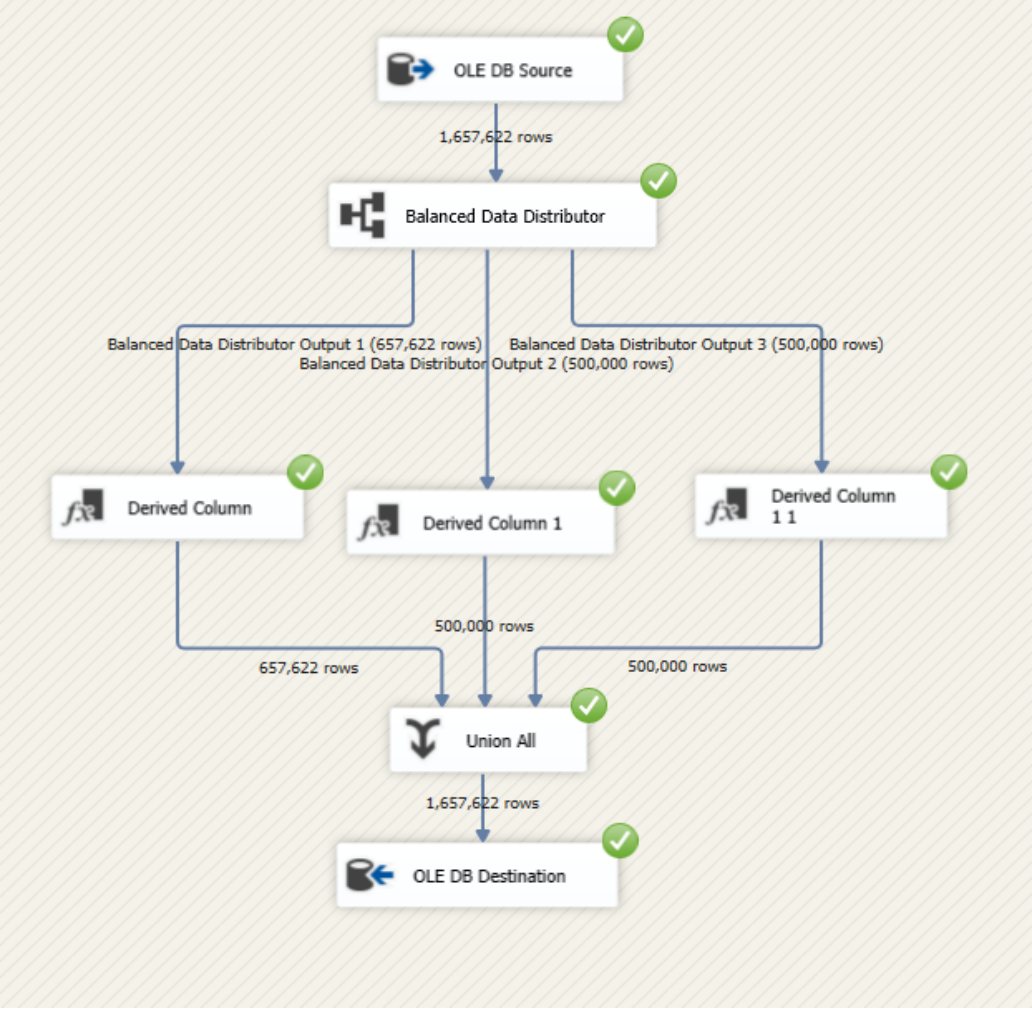
6. ETL Development – Accumulating Fact Tables

Process Time Data Preparation and Update

- As per the assignment requirement, a separate dataset was prepared to store the necessary data structure.
- Instead of using a flat file (CSV), a SQL Server table was created to hold the complete_time values.
- This change was necessary because of the large dataset size (over 1.55 million records), which would have made managing a flat file inefficient.
- The complete_time values were automatically created using a separate process and inserted into this SQL table to ensure better performance, easy integration, and smooth handling of the data.

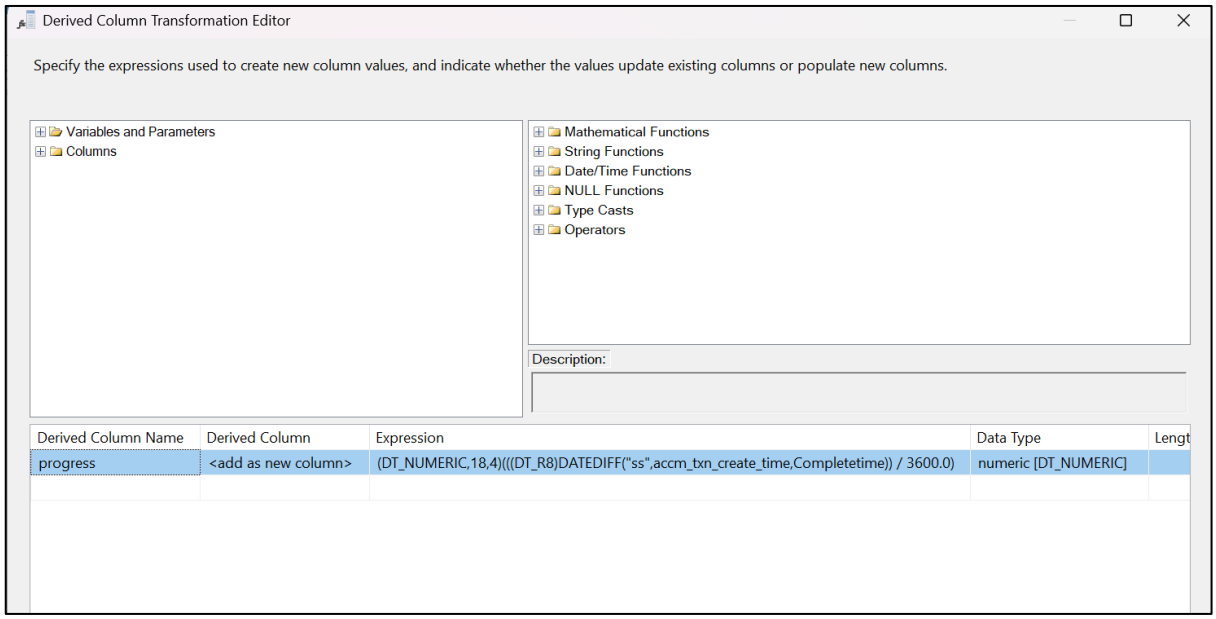


- Generate complete_time values

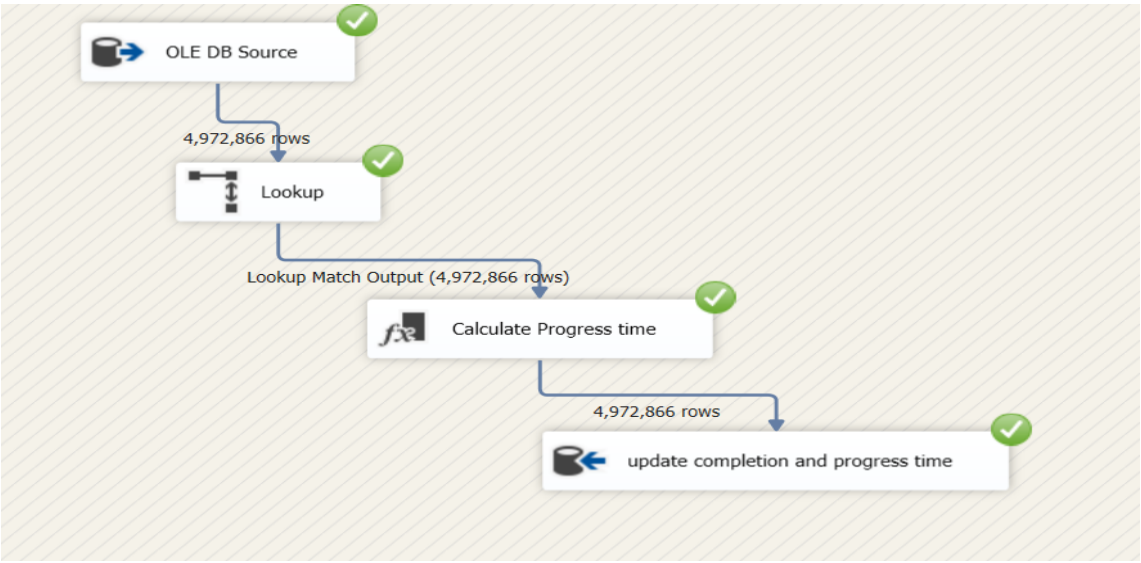


After creating and preparing the SQL table, the process moved to the next phase:

- The process time in hours is calculated by taking the difference between the create_time and the generated complete_time.
- This calculation and update are performed in a separate SSIS package, where:
    - A Derived Column transformation calculates the process_time in hours.
    - The txn_process_time_hours column is updated with the newly calculated process time.
    - The accm_txn_complete_time column is updated using the complete_time values from the SQL table.



- Calculate process time and update



7. Overall Execution Flow of the Total Solution

The overall solution follows a well-organized execution flow, starting from staging data to final updates in the data warehouse:

- First, the Staging.dtsx package is executed to extract the raw data from the sources and load it into the staging area.
- After the staging process, the DW.dtsx package is executed to transform the staged data and load it into the appropriate dimension and fact tables within the data warehouse.
- Upon completion of the data loading, the final step within the DW.dtsx package triggers the execution of the GenerateCompletionTime.dtsx package.
- The GenerateCompletionTime.dtsx package calculates and updates the txn_process_time_hours and accm_txn_complete_time columns in the fact table by processing the create_time and complete_time data.