Sri Lanka Institute of Information Technology

Data Warehousing and Business Intelligence(IT3021)

Assignment 1 – 2025, Semester 1

BY

FERNANDO B D F - IT22250438

Dataset Selection

A business scenario involving Supermarket sales served as the basis for the dataset used for this study's analysis. This dataset represents a retail setting in which customers buy things from a store - is recorded in each row of this dataset, which was constructed using an Online Transaction Processing style structure. The dataset used in this project was self-generated to simulate a realistic Supermarket environment. It includes customer details, product information, transaction records, and transaction completion times covering a full year of data.

The goal is to evaluate and organize this transactional data into a dimensional data warehouse for business intelligence and reporting purposes.

Additionally, around one year’s worth of data has been generated to ensure coverage for time-based reporting.

The dataset consists of the following tables

- Customer - Contains customer information such as CustomerID, Full Name, Gender, Email, Age, and Country
- Product - Stores product details including ProductID, ProductName, Category, and UnitPrice.
- Transaction - Records all sales transactions, including TransactionID, ProductID, CustomerID, Quantity, and TransactionDate.
- Sale - This will be the fact table (Fact_Transactions) in the data warehouse, combining keys and metrics for analysis

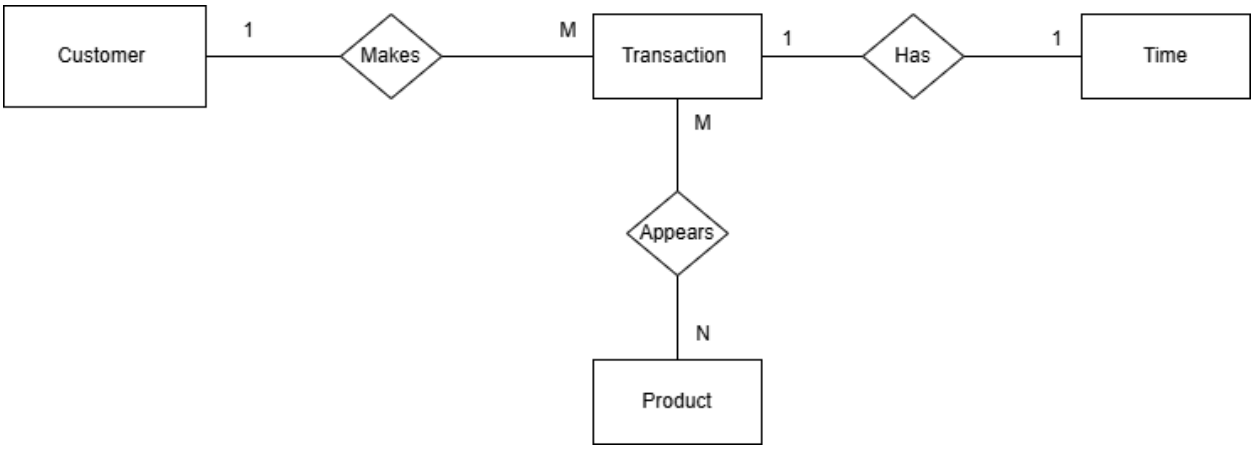
These files were initially provided in the following formats

- **Customers.csv** – comma-separated values file containing raw customer data
- **Products.xlsx** – Excel file with product details
- **Transactions.sql** – Pre-loaded table in SQL Server with historical sales

transactions

- **Complete_times.csv** – Used to update completion timestamps for each transaction

ER Diagram



Preparation of Data Sources

For this project, multiple types of data sources were prepared to simulate a real-world. Different file formats were used, and they were connected to the ETL process through appropriate connection managers in SQL Server Integration Services .

The table details are as follows:

- Customer –
    - Format – Flat File(.csv)
    - Customer data including CustomerID, FullName, Gender, Age, Email, and Country were stored in a CSV file as Customer.csv which is connected using Flat file connection Manager.

- Product –
    - Format – Excel File(.xslx)

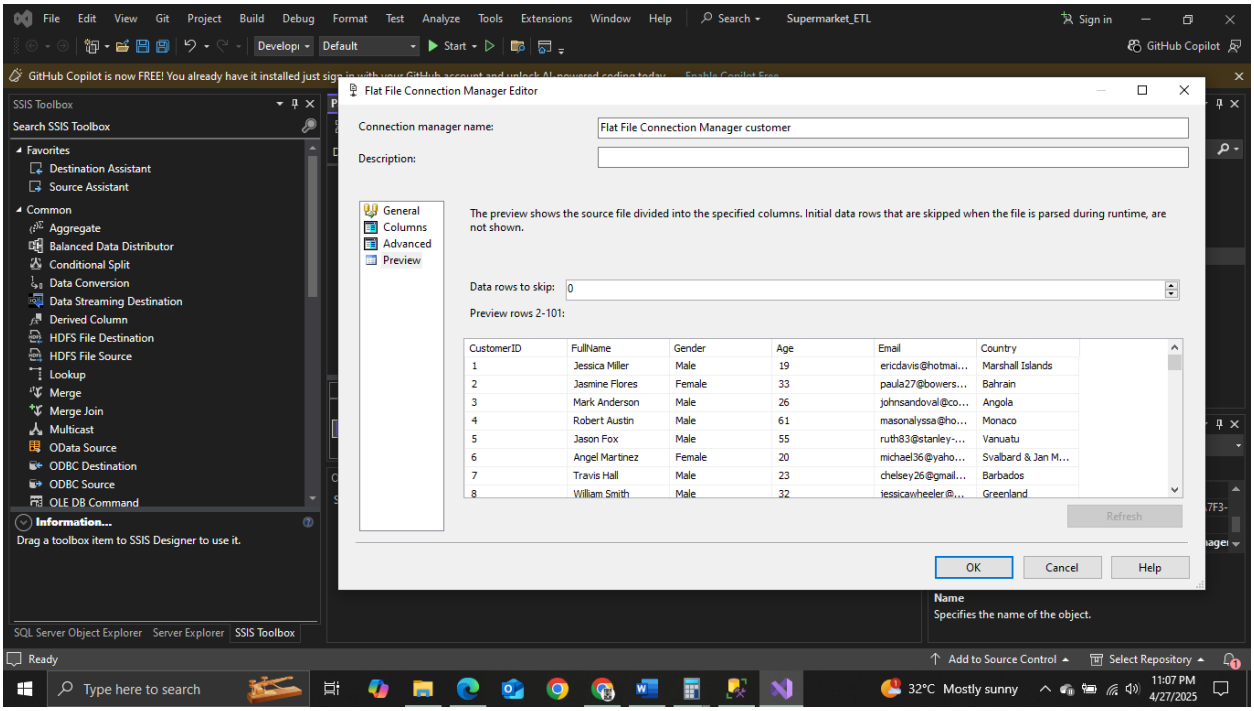
    - Product details such as ProductID, ProductName, Category, and UnitPrice were stored in an Excel sheet as product.xslx which is connected using an Excel Connection Manager.

- Transactions
    - Format - SQL Server Table
    - Sales transaction records including TransactionID, CustomerID, ProductID, Quantity, and TransactionDate were available in SQL Server table which was pre-created by using OLE DB Connection Manager for connection.

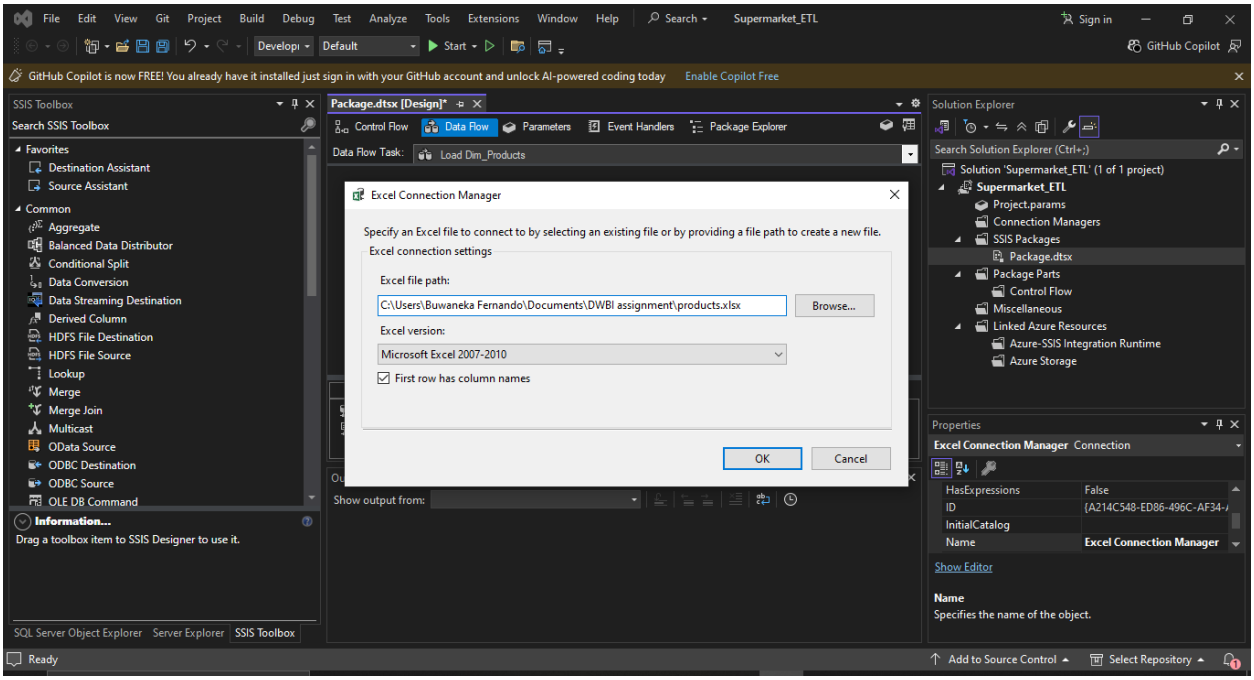
- Complete Times
    - Format – Flat File(.csv)
    - This separate CSV file contained TransactionIDs along with their corresponding Transaction completion times which accm_txn_complete_time. Connected using a Flat File Connection Manager.

Each source was connected to SQL Server Integration Services by appropriate settings using ,

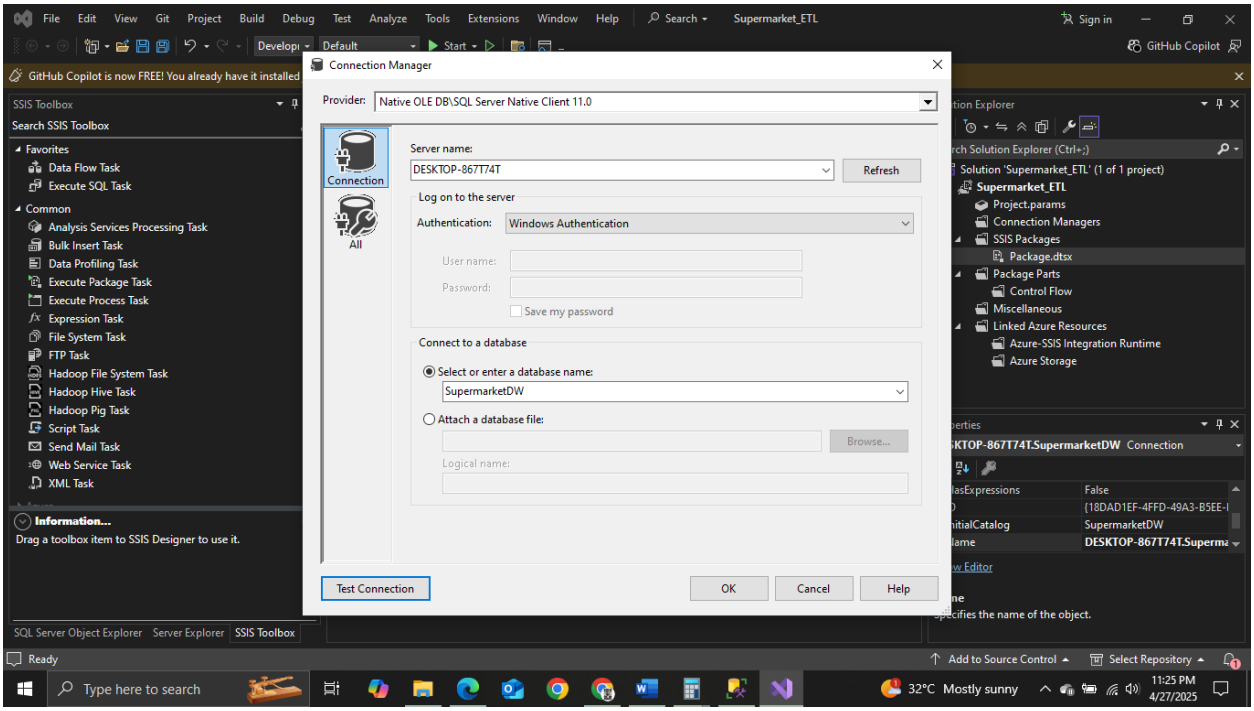
- Using first row as column names
- Verified data types to ensure compatibility



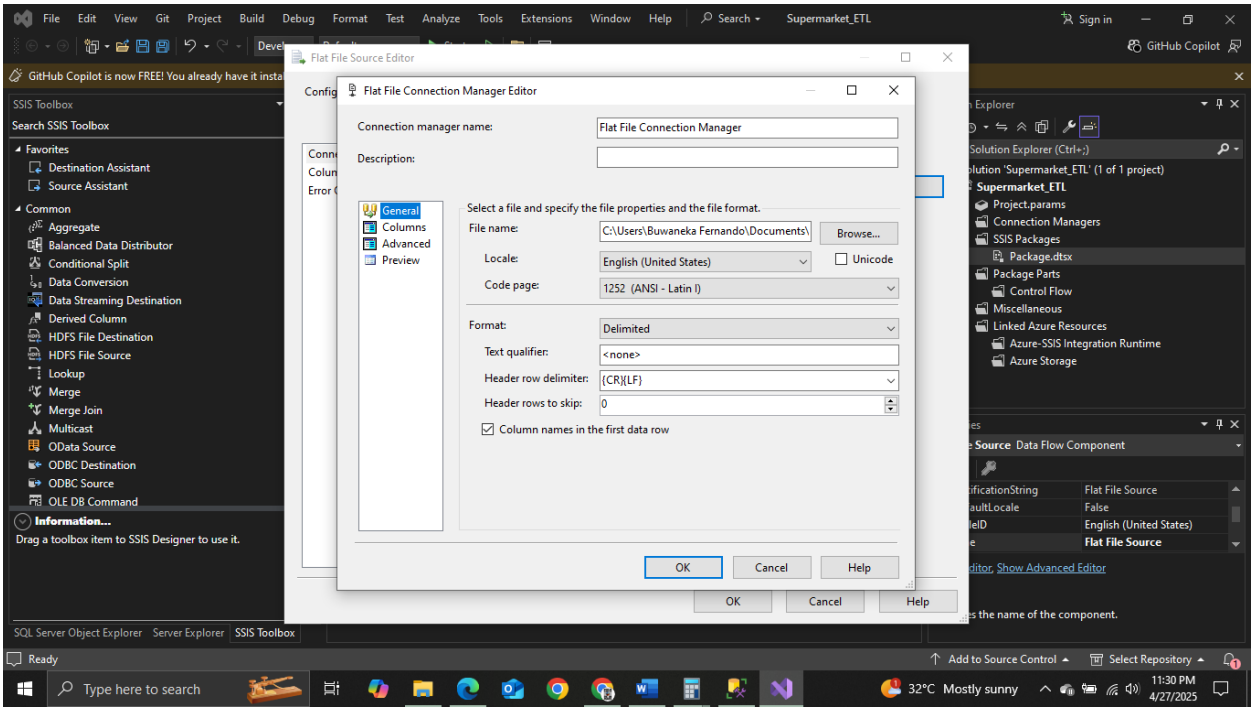
Set Up the Customers CSV Source



Excel Connection Manager Setup for Products



OLE DB Connection Setup for SQL Server



- Flat File Source Setup for Complete Times

Solution Architecture Design

This project's overall solution design uses ETL procedures to integrate various data sources into a dimensional data warehouse, according to a typical data warehousing flow.

Data Sources

- Customers.csv
- Products.xlsx
- Transactions
- CompleteTimes.csv

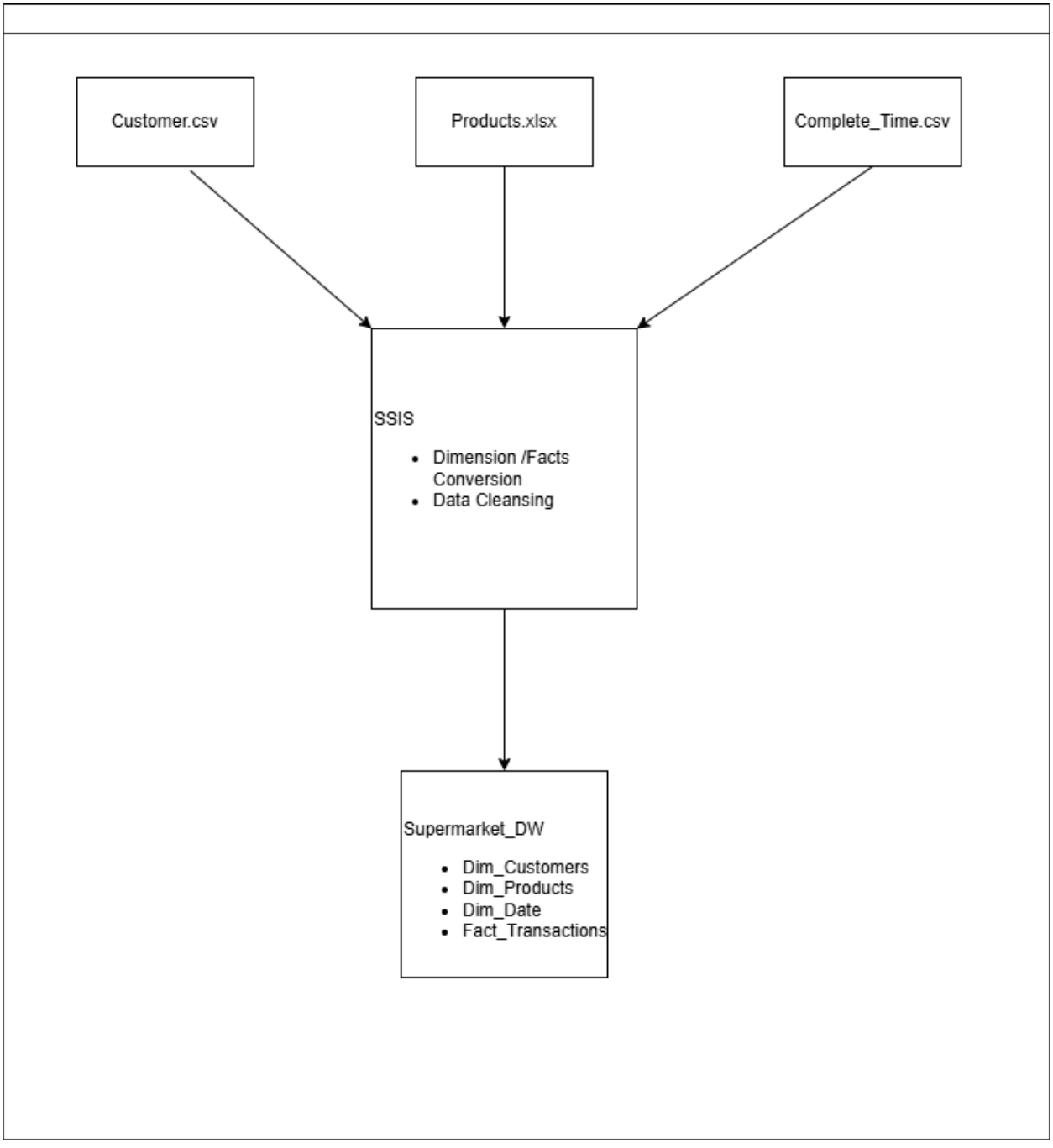
For ETL Process

- Built using **SQL Server Integration Services**
- Task Performed
    - Data extraction
    - Data cleaning
    - Slowly Changing Dimension Management
    - Fact table loading

Data Warehouse

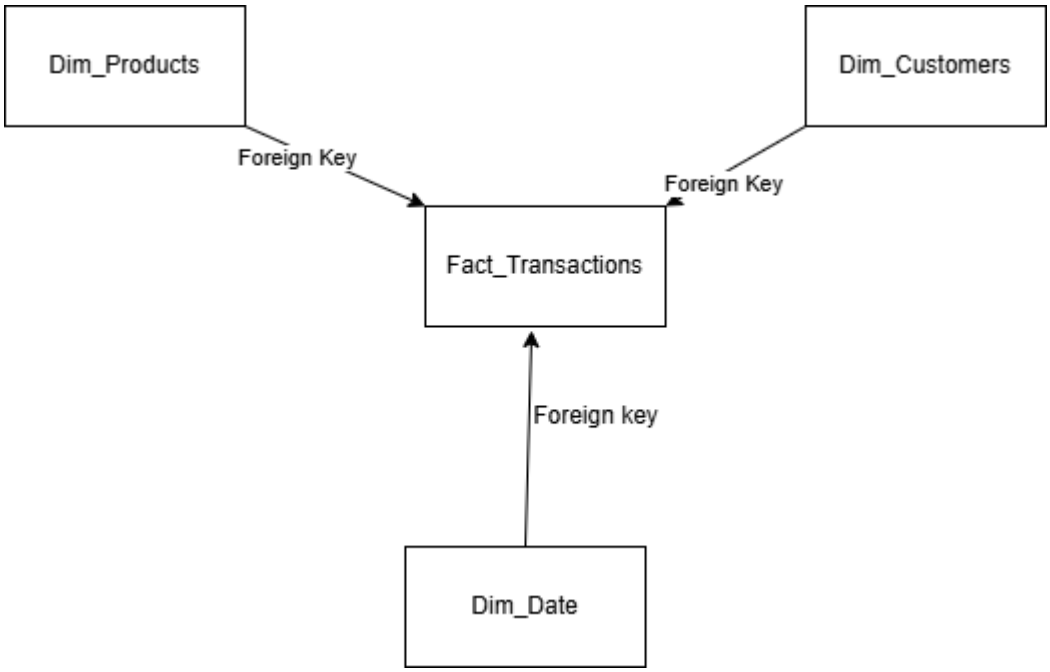
- Target database: SupermarketDW connected on SQL Server Management Studio.
- Dimention Tables – Customer , Product, Date
- Fact Tables – Transaction

The following diagram provides the relevant details,



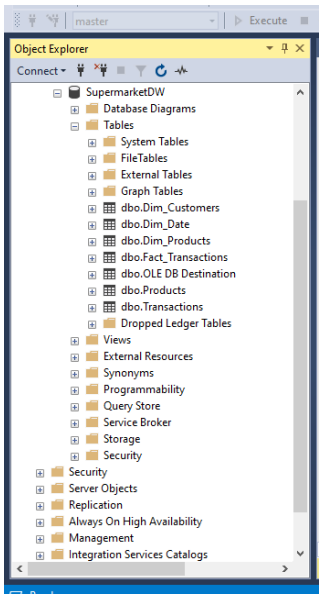
Data Warehouse Design and Development

The Supermarket data warehouse is designed based on a **Star Schema** dimensional model. The schema consists of three-dimension tables and one fact table, which enables flexible and efficient business analysis.

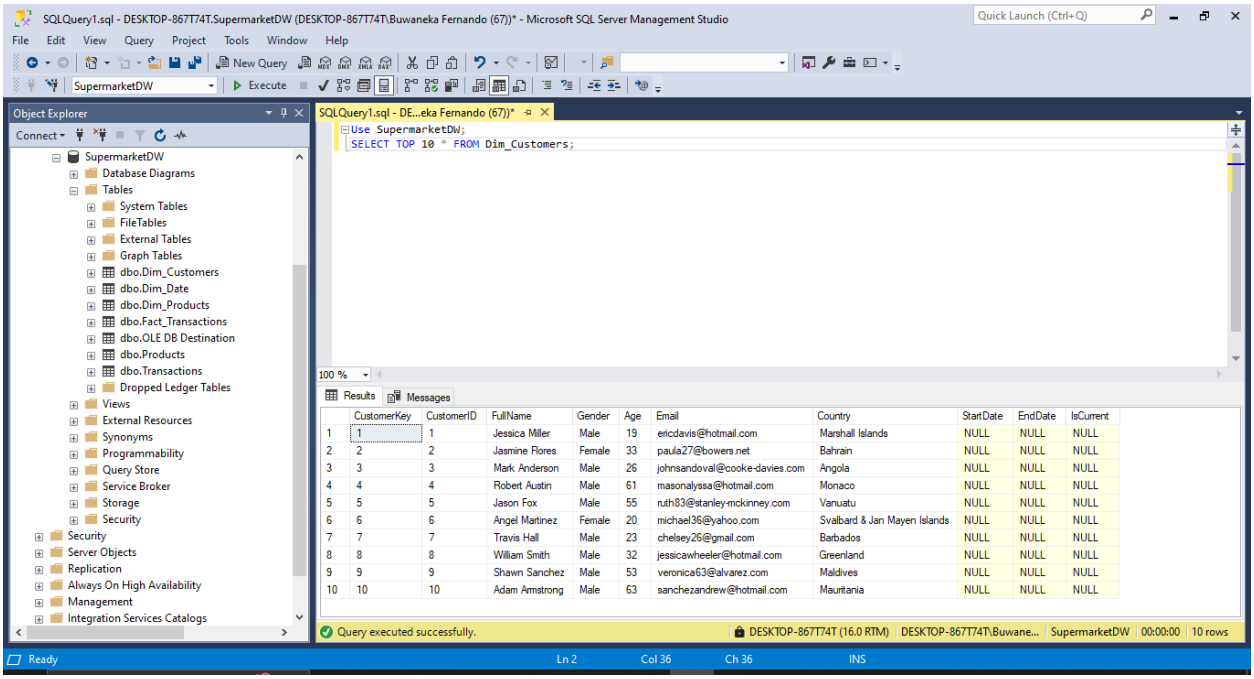


The tables included,

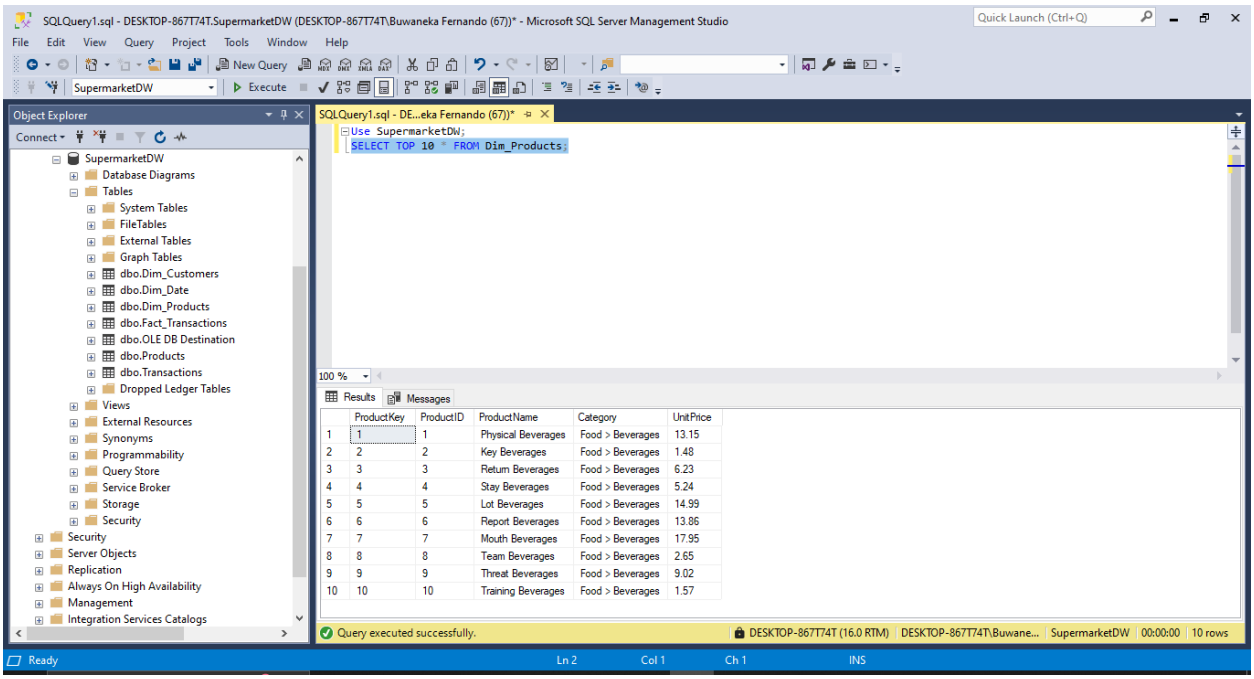
- Dim_Customers - Stores customer information
- Dim_Products - Stores product details
- Dim_Date - Stores detailed date information
- Fact_Transactions - Stores transactional information



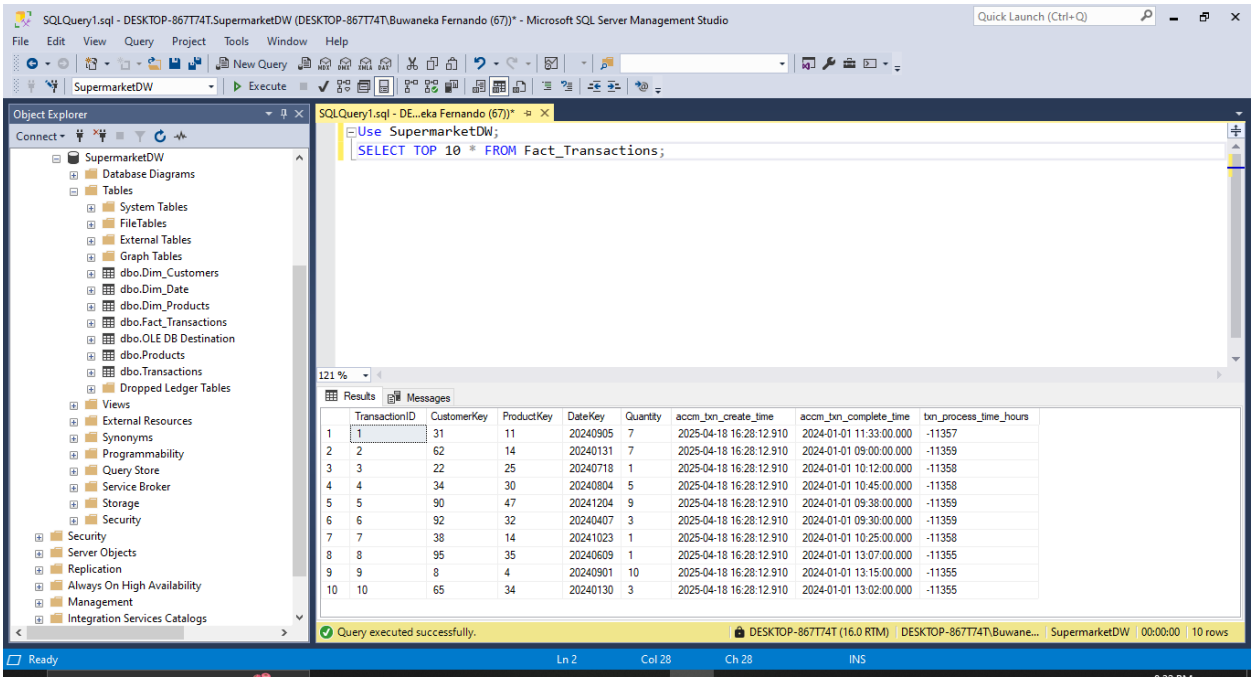
List of tables inside Supermarket_DW



Sample Data from Dim_Customers



Sample Data from Dim_Products



Sample Data from Fact_Transactions

ETL Development

The ETL (Extract, Transform, Load) process was developed using **SQL Server Integration Services**. It involves data extractions from various sources, transformed as needed and then loaded into the data warehouse following the star schema model.

Data Sources

Multiple types of data sources were used:

- **CSV Files**: customers.csv, complete_times.csv
- **Excel File**: products.xlsx
- **SQL Server Table**: transactions

SSIS Tasks and Transformations

Flat File Source - To extract customer and complete time data from CSV files

Excel Source - To extract product data from an excel sheet

OLE DB Source - To extract transaction data from SQL Server table

Data Conversion - To convert data types

Derived Column - To create new calculated fields such as accm_txn_create_time

OLE DB Destination - To load the cleaned and transformed data into the data warehouse tables

OLE DB Command - To update completion timestamps

Lookup Transformations - To map CustomerID, ProductID, and TransactionDate to their related dimension keys

Transformations Types

Data Type Conversion - Handled mismatches between source data types and warehouse schema.

Lookup - Mapped Transaction foreign keys

Derived Column - Added system timestamps and duplicated fields for update operations

ETL Development – Accumulating Fact Tables

1. **Fact Table Extension**

In the Fact_Transactions table, the table was extended by adding these columns

- accm_txn_create_time - Indicates the time when the transaction record was created (datetime)
- accm_txn_complete_time - Indicates the actual time when the transaction was completed.(datetime)
- txn_process_time_hours - The difference between completion time and creation time in hours. (int)

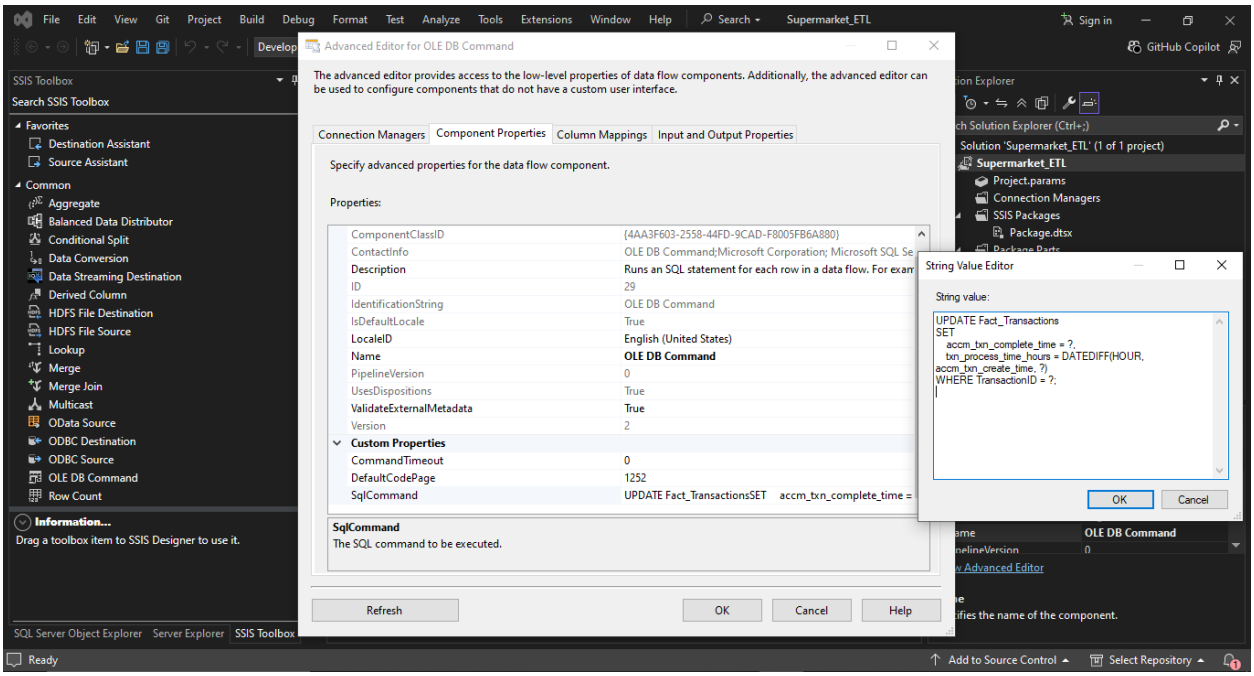
2. **Setting accm_txn_create_time**

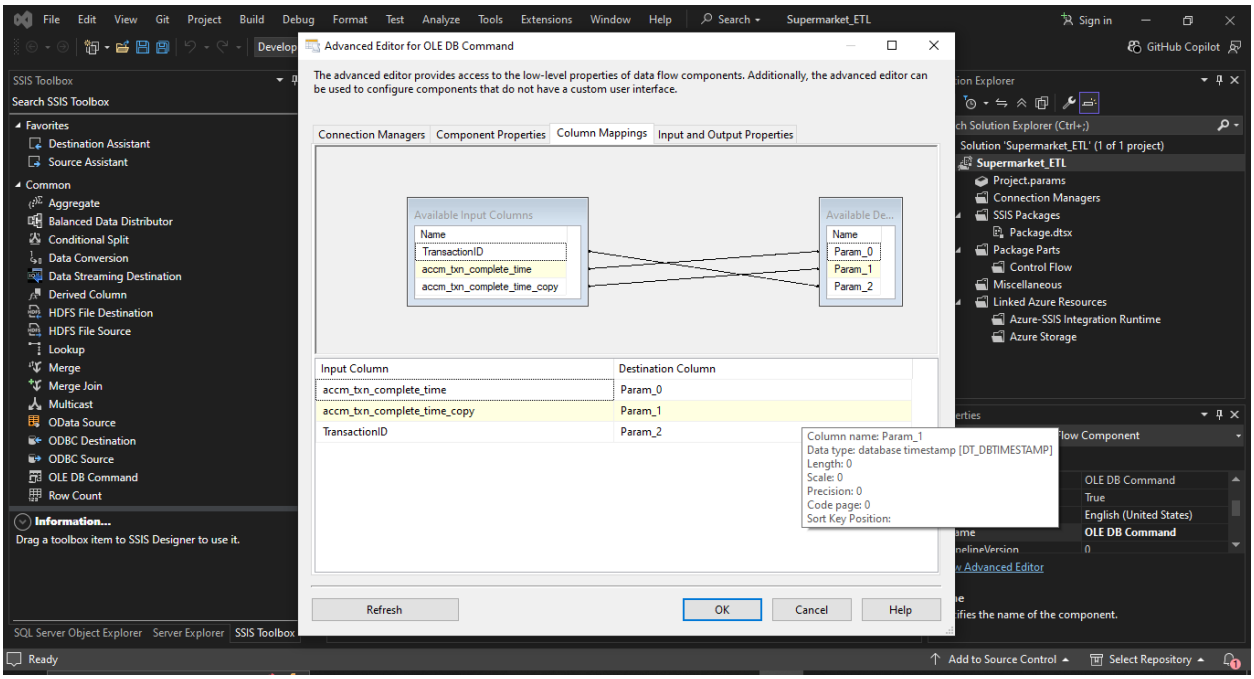
During the initial ETL load of transactions, a **Derived Column** in Server Integration Services was used to set accm_txn_create_time by using GETDATE() function to get the current date and time.

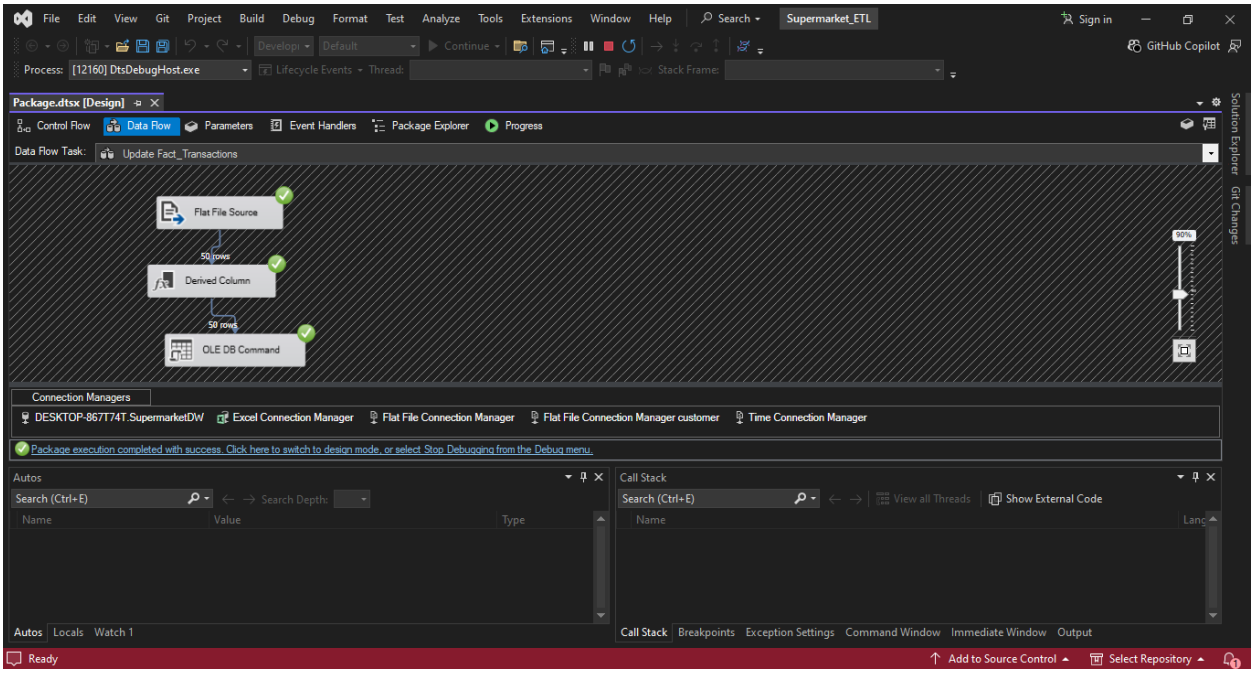
3. In the completion of dataset, a separate dataset called complete_times.csv was created, which containing **TransactionID** and **accm_txn_complete_time.**

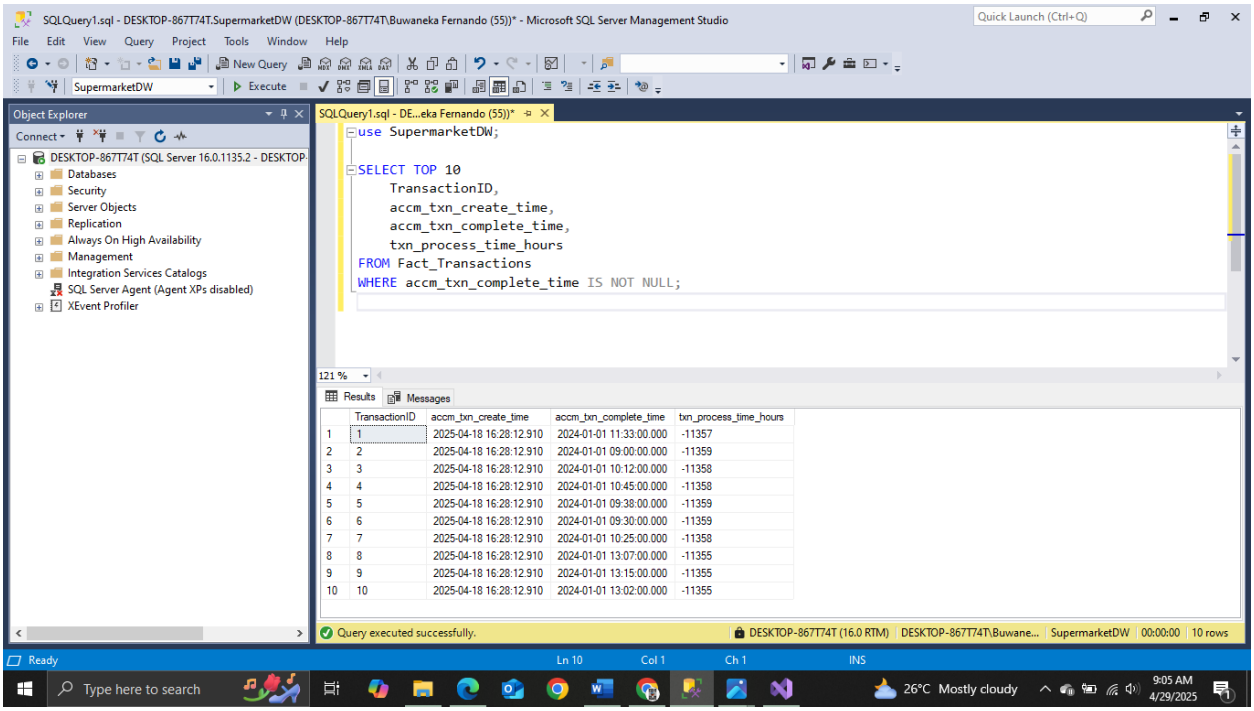
4. **Updating Fact Table with SSIS**
    - A separate SSIS Data Flow was developed for updating the Fact Table as follows:
        - **Step 1** - Flat File Source reads complete_times.csv
        - **Step 2** - Derived Column creates a duplicate of accm_txn_complete_time for parameter mapping
        - **Step 3** - OLE DB Command updates Fact_Transactions setting in
            - **accm_txn_complete_time**
            - **txn_process_time_hours** = DATEDIFF(HOUR, accm_txn_create_time, accm_txn_complete_time)

5. accm_txn_complete_time in your DW fact table




Advanced Editor for OLE DB Command
The advanced editor provides access to the low-level properties of data flow components. Additionally, the advanced editor can be used to configure components that do not have a custom user interface.
Connection Managers
Component Properties
Column Mappings
Input and Output Properties
Available Input Columns
Name
TransactionID
accm_txn_complete_time
accm_txn_complete_time_copy
Available De...
Name
Param_0
Param_1
Param_2
Input Column
Destination Column
accm_txn_complete_time
Param_0
accm_txn_complete_time_copy
Param_1
TransactionID
Param_2
Column name: Param_1
Data type: database timestamp [DT_DBTIMESTAMP]
Length: 0
Scale: 0
Precision: 0
Code page: 0
Sort Key Position:
Refresh
OK
Cancel
Help
SSIS Toolbox
Search SSIS Toolbox
Favorites
Destination Assistant
Source Assistant
Common
Aggregate
Balanced Data Distributor
Conditional Split
Data Conversion
Data Streaming Destination
Derived Column
HDFS File Destination
HDFS File Source
Lookup
Merge
Merge Join
Multicast
OData Source
ODBC Destination
ODBC Source
OLE DB Command
Row Count
Information...
Drag a toolbox item to SSIS Designer to use it.
Solution Explorer
Solution 'Supermarket_ETL' (1 of 1 project)
Supermarket_ETL
Project.params
Connection Managers
SSIS Packages
Package.dtsx
Package Parts
Control Flow
Miscellaneous
Linked Azure Resources
Azure-SSIS Integration Runtime
Azure Storage
OLE DB Command
True
English (United States)
OLE DB Command
Advanced Editor
Ready
Add to Source Control
Select Repository


Package.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Update Fact_Transactions
Flat File Source
50 rows
Derived Column
50 rows
OLE DB Command
Connection Managers
DESKTOP-867T74T.SupermarketDW
Excel Connection Manager
Flat File Connection Manager
Flat File Connection Manager customer
Time Connection Manager
Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.
Autos
Search (Ctrl+E)
Name
Value
Type
Call Stack
Search (Ctrl+E)
Show External Code
Name
Autos
Locals
Watch 1
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Ready
Add to Source Control
Select Repository


SQLQuery1.sql - DESKTOP-867T74T.SupermarketDW (DESKTOP-867T74T\Buwaneka Fernando (55))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
SupermarketDW
Execute
Object Explorer
Connect
DESKTOP-867T74T (SQL Server 16.0.1135.2 - DESKTOP-
Databases
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
SQLQuery1.sql - DE...eka Fernando (55))*
use SupermarketDW;
SELECT TOP 10
TransactionID,
accm_txn_create_time,
accm_txn_complete_time,
txn_process_time_hours
FROM Fact_Transactions
WHERE accm_txn_complete_time IS NOT NULL;
121 %
Results
Messages
TransactionID accm_txn_create_time accm_txn_complete_time txn_process_time_hours
1 1 2025-04-18 16:28:12.910 2024-01-01 11:33:00.000 -11357
2 2 2025-04-18 16:28:12.910 2024-01-01 09:00:00.000 -11359
3 3 2025-04-18 16:28:12.910 2024-01-01 10:12:00.000 -11358
4 4 2025-04-18 16:28:12.910 2024-01-01 10:45:00.000 -11358
5 5 2025-04-18 16:28:12.910 2024-01-01 09:38:00.000 -11359
6 6 2025-04-18 16:28:12.910 2024-01-01 09:30:00.000 -11359
7 7 2025-04-18 16:28:12.910 2024-01-01 10:25:00.000 -11358
8 8 2025-04-18 16:28:12.910 2024-01-01 13:07:00.000 -11355
9 9 2025-04-18 16:28:12.910 2024-01-01 13:15:00.000 -11355
10 10 2025-04-18 16:28:12.910 2024-01-01 13:02:00.000 -11355
Query executed successfully.
DESKTOP-867T74T (16.0 RTM)
DESKTOP-867T74T\Buwane...
SupermarketDW
00:00:00
10 rows
Ready
Ln 10
Col 1
Ch 1
INS
Type here to search
26°C Mostly cloudy
9:05 AM
4/29/2025

Conclusion

This project successfully illustrates the end-to-end development of a retail sales data warehouse solution using Microsoft SQL Server and SQL Server Integration Services (SSIS).

Key accomplishments include:

- Preparing a realistic OLTP dataset covering customer, product, and transaction information across a one-year period.
- Creating dimension tables and a fact table in data warehouse using the Star Schema model.
- Designing a complete ETL pipeline using SQL Server Integration Services, involving data extraction from multiple sources and also, data transformations, lookup functions, and loading procedures are included.
- Generating an update pipeline for transaction completion times and computing process durations to manage the generation and updating of an accumulating fact table.
- Clarifying the data warehouse integrity by executing validation queries and ensuring accurate loading and updating of records.

Overall, the project demonstrated key skills in data modeling, ETL development, and practical application of data warehousing principles.