This project is a comprehensive Walmart Sales Analysis that tracks sales performance, product categories, employee data, store locations, and other key metrics. The project follows a structured approach involving:

Data Population: Using a customized SQL script, the transactional data (e.g., products, stores, sales) was populated into the SQL database.

Data Transfer (ETL Process): SSIS (SQL Server Integration Services) packages were created to extract, transform, and load (ETL) data from the transactional database into the data warehouse (WalmartDWH) to support analytical reporting.

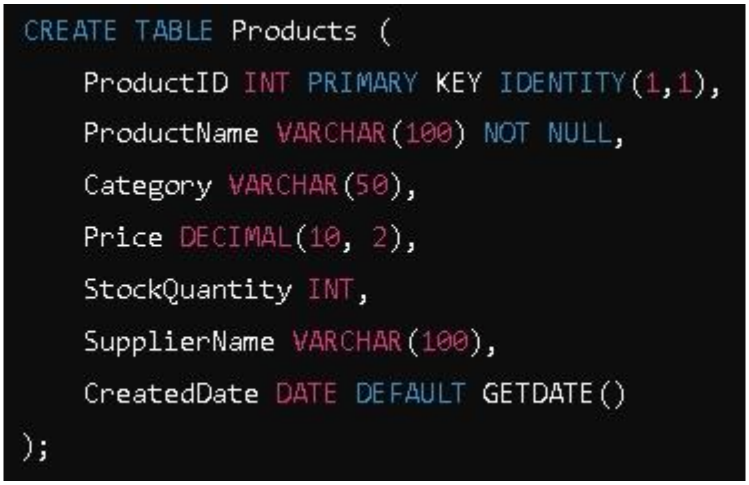
Dashboard Creation in Power BI: Finally, the data warehouse was connected to Power BI, where key visualizations and metrics were built into a comprehensive dashboard that provides insights into sales performance, product categories, and store sales.

Step-by-Step Process:

1. **Database Setup and Population**

Create SQL Database Tables: Using the SQL script, tables were created for products, employees, stores, customers, and sales, each with the relevant fields and foreign keys.

Example:



```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY IDENTITY(1,1),
    ProductName VARCHAR(100) NOT NULL,
    Category VARCHAR(50),
    Price DECIMAL(10, 2),
    StockQuantity INT,
    SupplierName VARCHAR(100),
    CreatedDate DATE DEFAULT GETDATE()
);
```

Insert Data into Tables: Data was manually populated or imported into these tables to simulate Walmart sales transactions, including product and customer details, store data, employee details, and sales records.

Fact and Dimension Tables: Fact tables were created to hold the transactional data, while dimension tables stored the descriptive attributes such as Product, Customer, and Employee.

2. Data Warehouse Design and ETL

Create Data Warehouse (WalmartDWH): The data warehouse was designed with the following star schema structure:

Dimension Tables:

DimCustomer (CustomerID, FirstName, LastName, etc.)

DimEmployee (EmployeeID, FirstName, LastName, etc.)

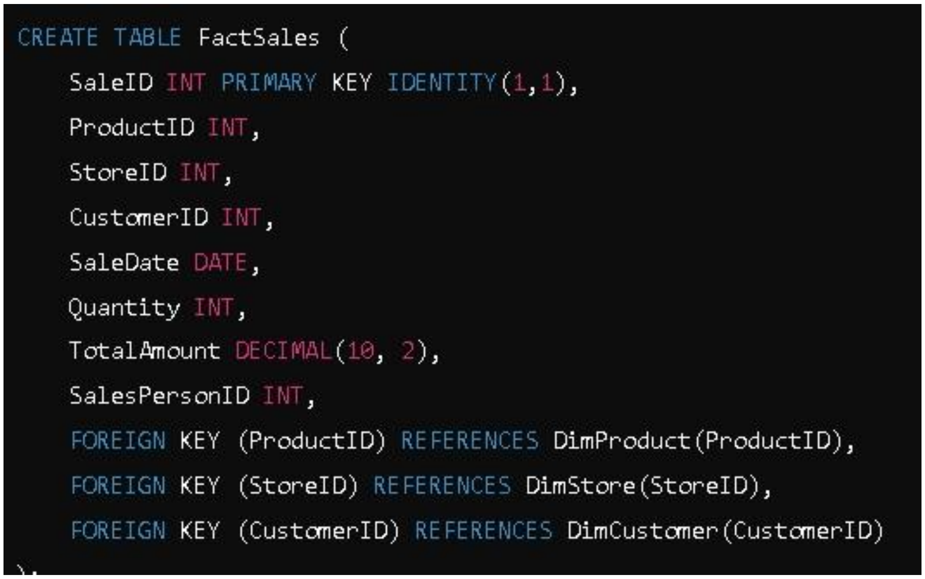
DimProduct (ProductID, ProductName, Category, etc.)

DimStore (StoreID, StoreName, Location, etc.)

Fact Table:

FactSales (SaleID, Quantity, TotalAmount, SaleDate, etc.)

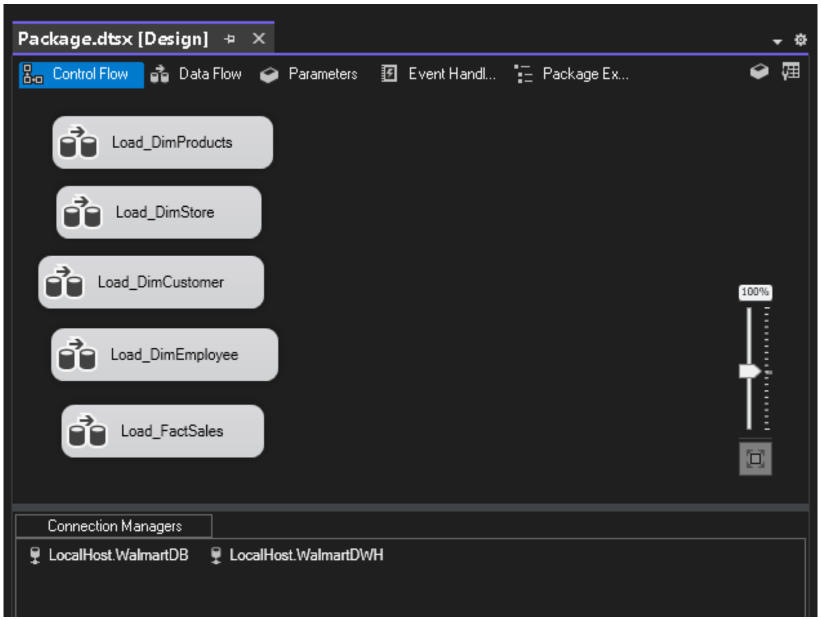
Example of DWH schema:

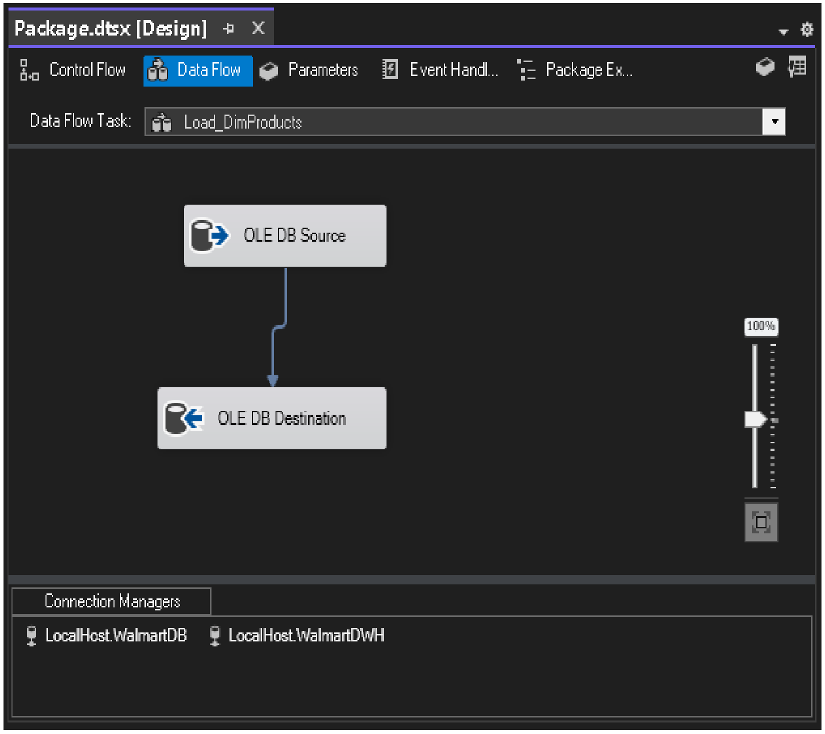


```
CREATE TABLE FactSales (
    SaleID INT PRIMARY KEY IDENTITY(1,1),
    ProductID INT,
    StoreID INT,
    CustomerID INT,
    SaleDate DATE,
    Quantity INT,
    TotalAmount DECIMAL(10, 2),
    SalesPersonID INT,
    FOREIGN KEY (ProductID) REFERENCES DimProduct(ProductID),
    FOREIGN KEY (StoreID) REFERENCES DimStore(StoreID),
    FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID)
);
```

SSIS Package for Data Transfer:

ETL Process: SSIS packages were created to extract data from the SQL Server transactional database, transform the data to fit the schema, and load it into the data warehouse (WalmartDWH).




Data Flow Task: Load_DimProducts

Data Cleansing: Data cleansing steps were performed during the ETL process to ensure the data's accuracy and consistency before loading it into the data warehouse.

3. Power BI Dashboard Creation

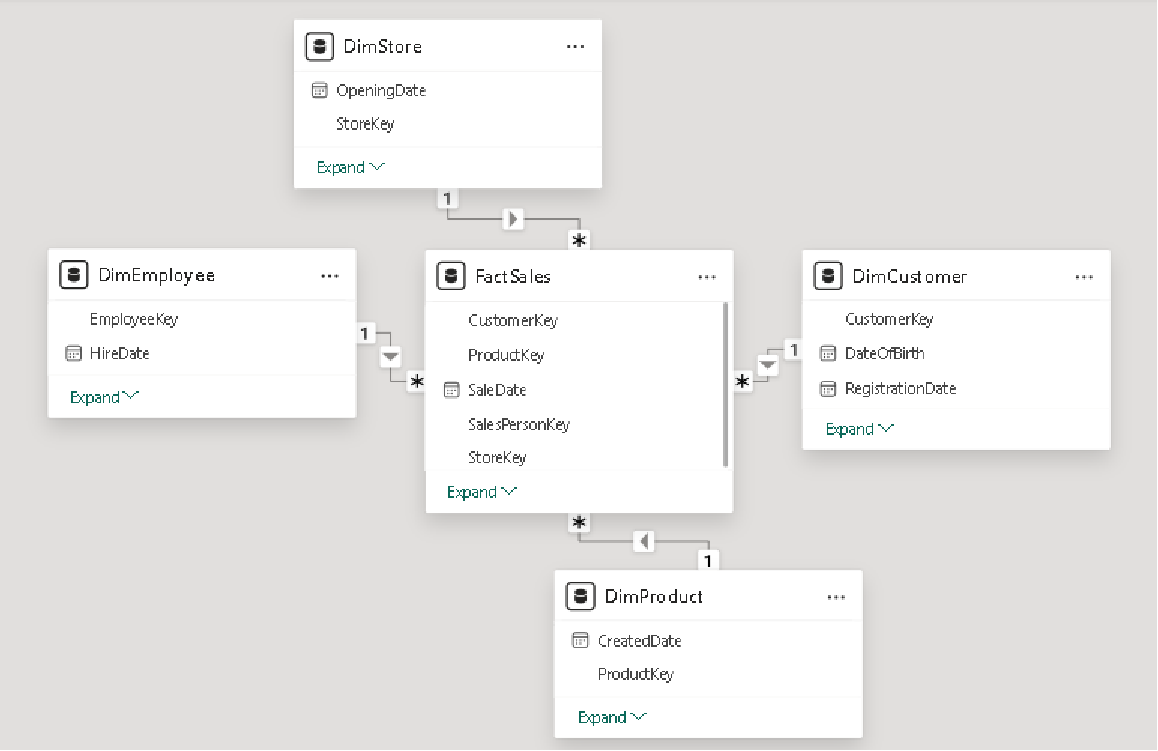
Connect Power BI to Data Warehouse:

The data warehouse (WalmartDWH) was connected to Power BI using SQL Server as the data source.

Power BI queries were built to import data from dimension and fact tables.

Data Modeling in Power BI:

Relationships were defined between dimension tables (DimProduct, DimCustomer, DimStore, DimEmployee) and the fact table (FactSales).



Measures such as Total Sales, Total Employees, Total Products, Total Categories, and more were created.

Dashboard Design: The dashboard was built with several key components:

Total Sales, Employees, Stores, and Products:

Purpose: Provides a high-level summary of total metrics.

How: Use a Card visualization with DAX measures like SUM(TotalAmount) for total sales and COUNT(EmployeeID) for total employees.

Sales Overtime Line Chart:

Purpose: Shows how sales have fluctuated over time.

X-Axis: SaleDate

Y-Axis: SUM(TotalAmount)

Top Products by Sales (Pie Chart):

Purpose: Displays the top 5 products based on total sales.

Legend: ProductName

Values: SUM(TotalAmount)

Sales by Product Category (Bar Chart):

Purpose: Provides a breakdown of sales by product categories.

X-Axis: Category

Y-Axis: SUM(TotalAmount)

Sales by Store Location (Map Chart):

Purpose: Displays sales based on store location.

Location: StoreName

Values: SUM(TotalAmount)

4. Final Steps

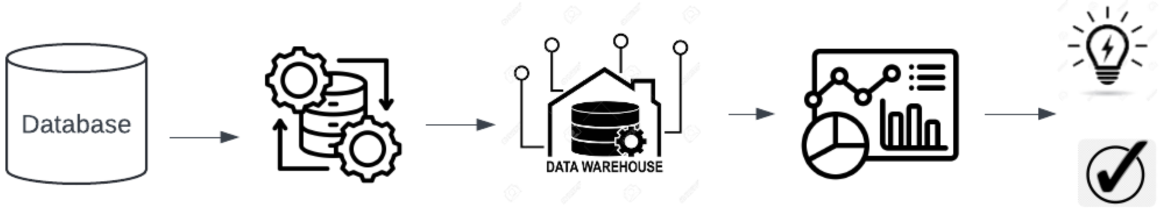
Data Filters: Slicers were added to allow filtering by Product Category, Store, Employee, and Date. This allows users to interact with the dashboard to get specific insights.

Optimization:

Performance tuning steps were taken to ensure fast dashboard refreshes, such as aggregating data at appropriate levels and optimizing DAX queries.

Conclusion:

This project showcases the complete data lifecycle from database design to data warehousing and dashboard creation. The dashboard offers key insights into Walmart's sales performance, helping stakeholders make data-driven decisions.


Database
DATA WAREHOUSE