Western Digital's Sales Data Mart

BI System Specifications Document

Date: 30/07/2024

Written by: Hagai Izchaki

Content

1. General

1.1. Project Objective

This project's objective is the creation of a full BI solution for WD's sales department, to support WD's growth in devices sales. The project was designed according to WD's sales department KPIs and is aimed at increasing the company's overall (and the devices sales department in particular) ROI.

Western Digital Corporation, commonly referred to as Western Digital or WD, is a leading global provider of data storage solutions. The company designs, manufactures, and sells data technology products, including storage devices, data center systems, and cloud storage services. Headquartered in San Jose, California, Western Digital has a rich history of innovation and a reputation for delivering high-quality and reliable storage solutions.

The Data Mart creation will be done using information derived from the PriorityERP database (DW's operational database). The solution will include summarized data tables, focusing on devices sales data, as well as data regarding DW's customers, salespersons, products, and stores. In addition, the BI solution will include costumed reports containing sales analysis, customer analysis, and executive dashboard. These reports will be tailored for the sales departments' needs and will contribute to WD's devices sales growth.

1.2. Project Contents

The project includes the building of a Data Mart which contains sales data. The data will be transferred through an ETL process from the PriorityERP operational database to the Data Mart – Western_Digital_SalesDM.

ERD model of the Western_Digital_SalesDM database: ERD Link

1.2.1. The Data Mart will include 1 fact table and 4 dimension tables, and 1 history table:

- **Fact_Sales** – Data regarding all sales, including the id of the order, products bought, quantities, and prices. Data loading process for this table will be incremental.
- **Dim_Customers** – Data regarding the company's customers.
- **Dim_Stores** – Data regarding the company's stores.
- **Dim_Employees** – Data regarding the company's employees(=salespersons).
- **Dim_Products** - Data regarding the company's products.
- **Dim_Products_History** – Historic data regarding the company's products. Source To Target Link

1.2.2. The reports will include data visualizations that will support the project's objective in the following ways:

- **Sales Analysis:**

  The sales report will include data about sales (revenue, number of orders, and number of units) by date, country, product, store (online vs. physical), and salespersons which will help the department to assess the performance of all the parts needed for sales growth. The reports will help to identify sale trends like seasonality and trending product categories, analyze products orders and revenue, spot top performing employees, and analyze the differences in behavior between the online store and

physical stores. All of these will support data driven strategic decision making which can lead to growth in sales and revenue.

- **Customer Analysis:**

  The customers analysis report will include data regarding WD's customers by date, country, store, product, and category. This report is aimed to help WD's customer department to better understand their customers' behavior, like what (products), where (countries and stores), and when do they shop. This is vital to retain current customers and reach new ones.

- **Executive Dashboard:**

  The dashboard will include key visuals from the two reports. The dashboard will allow a wider perspective on the data and will integrate measures both from sales and customer analysis.

2. Gnatt

Gnatt Link

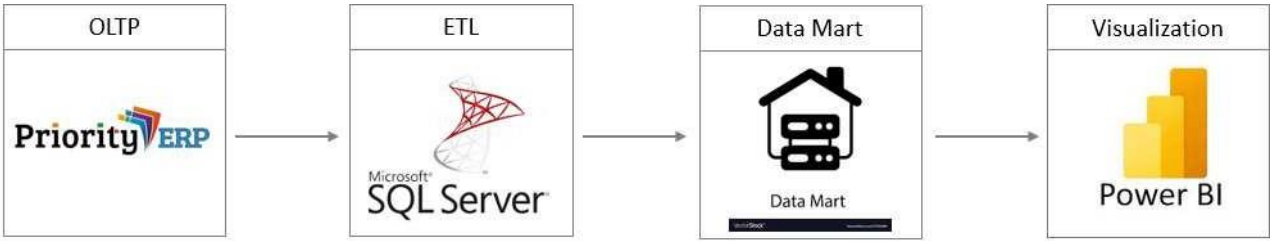
3. Technical Specification

3.1. Prerequisites

SQL Server	ERP system in the operational DB (PriorityERP) - tables, data (SQL files)
SSIS	ETL processes using SSIS in Visual Studio
Data refresh processes	Definition of JOBS in SSMS
Power BI	Creating reports and dashboards using Power BI

3.2. Solution Architecture

3.2.1. High Level Design:



The ETL process, which includes arranging the data into a Data Mart will be performed in SQL Server using SSIS. After the Data Mart creation, reports will be created using Power BI.

3.2.2. Power BI Reports:

3.2.2.1. The report for the sales department will consist of:

- Total sales
- YTD sales
- Total orders

- Total units
- Total sales and Year over Year growth (this graph can change to orders, units)
- Total sales by month and day - online vs physical stores (this graph can change to orders, units)
- Top stores by sales (this graph can change to orders, units)
- Sales by country (this graph can change to orders, units)
- Sales by category (this graph can change to orders, units)

3.2.2.2. The report for the customer department will consist of:

- Total number of customers
- Number of new customers
- Average revenue per customer
- Average orders number per customer
- Total customers and new customers by Year/Quarter/Month
- Customers by country
- Top products by number of Customers
- Average revenue per customer compared to previous Year/quarter/Month.

3.2.2.3. The executive dashboard will consist of:

- Total sales
- Total orders
- Total customers
- Percentage of new customers from total
- YTD sales
- Total sales and month over month growth by quarter and month
- Revenue by online vs physical stores(order type)
- Total customers by Year/Quarter/Month
- Revenue by country

4. Functional Specification

4.1. Creation of final Source to Target and ERD models.

4.1.1. Source to Target

- Source To Target link

A total of 11 tables will be used from the operational database.

4.1.2. ERD model of the Western_Digital_SalesDM database

- ERD link

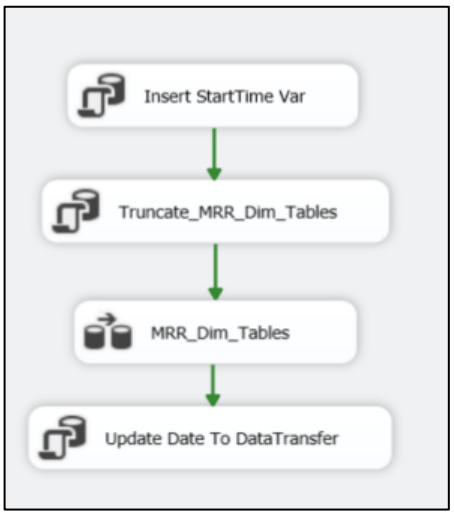
4.2. ETL processes

The ETL process was done in SSIS using 15 packages.

All the packages include 2 reoccurring Execute SQL tasks (Update StartTime Var and Insert Data to DataTransfer), and Row Count transformations which oversee updating the DataTransfer table. These will be explained later in the DataTransfer table section.

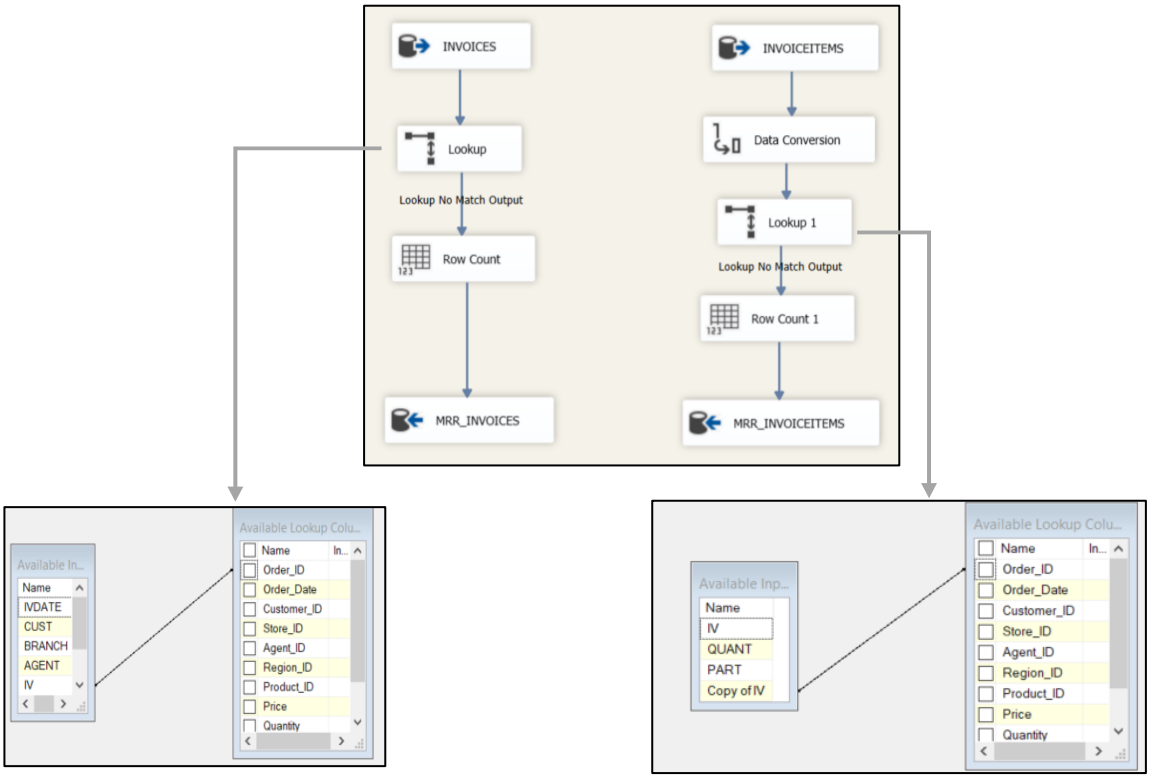
- **Fact_Sales Table:**
  - MRR_Dim_Orders package:

    Mirror tables are truncated (using a stored procedure) and data is loaded from PriorityERP database (INVOICES, INVOICEITMES, CUSTOMERS, DESTCODES, PART, STATES) to Western_Digital_SalesDM mirror tables.



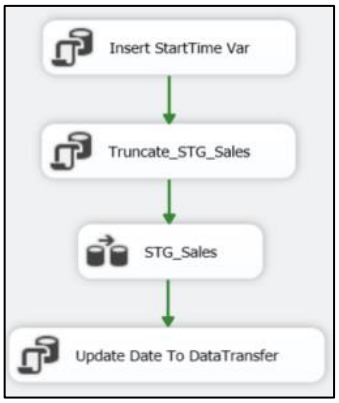
In the data flow, data is incrementally loaded using lookup transformations, meaning only new transactions that cannot be found in the Fact_Sales table are loaded:

Data Conversion transformation is used to convert the IV column from bigint to int (DT_I4), this is necessary for the lookup transformation:

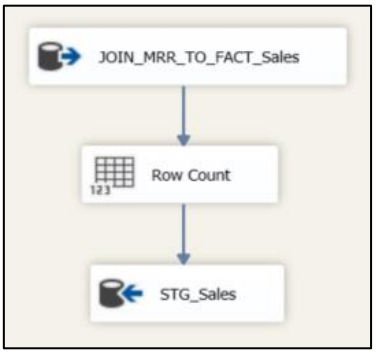


- <u>STG_Sales package</u>:

  STG_Sales table is truncated, and the mirror tables are joined and loaded using a data flow task.

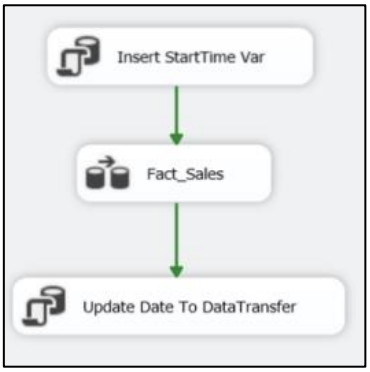


In the data flow, the mirror tables are joined, and the data is loaded into STG_Sales table.

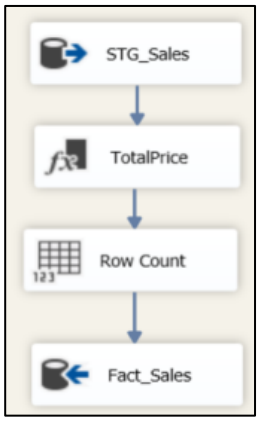


- Fact_Sales package:

  Data is loaded from STG_Sales to Fact_Sales, and a Total column is added.

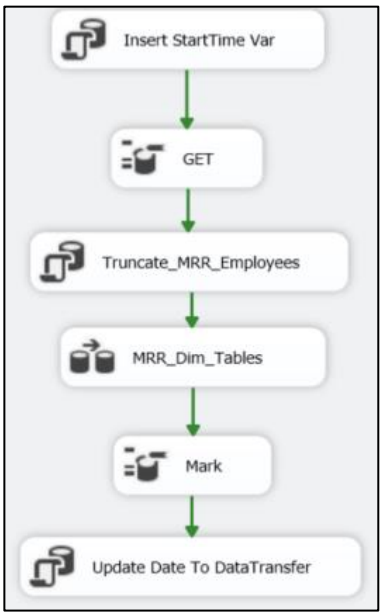


A Total column, which specifies the total price for the product in a sale, is calculated and added using a derived column transformation.

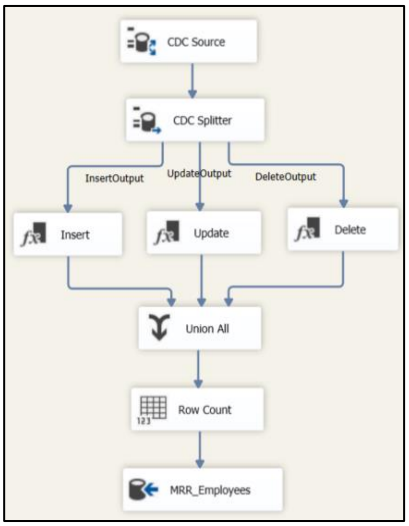


- **Dim_Employees Table:**
  - MRR_Employees package:

    In this table, we used the CDC process in order to document all the changes in the details of the employees that were transformed from the PriorityERP database:

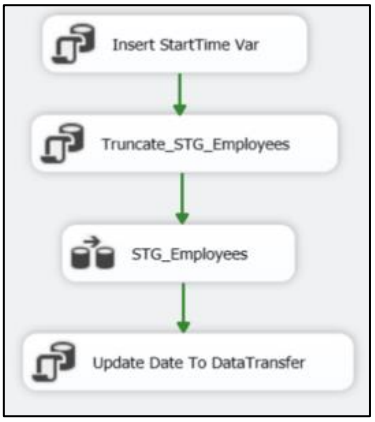


The changes are divided into 3 types: Insert, Update and Delete. A new column, "Status", will indicate which change has been made.



- STG_Employees package:

  STG_Employees table is truncated, and the mirror table is transformed and loaded using a data flow task.

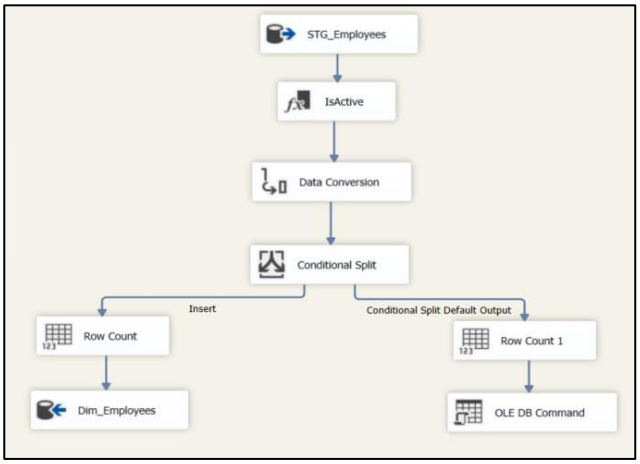


- DIM_Employees package:

  Data is incrementally loaded and updated in the Dim_Employees.

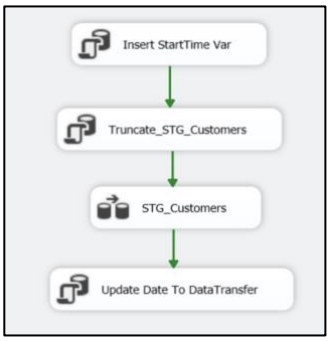
  An IsActive column, which indicates whether the employee is active or not, is added to the table using a derived column transformation.

  A new data (update or insert) will be loaded into Dim_Employees, and a deleted data will be treated by a query that will update the IsActive column to zero.

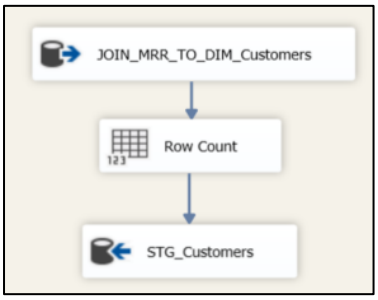


- **Dim_Customers Table**:
  - STG_Customers package:

    STG_Customers table is truncated, and the mirror tables are joined and loaded using a data flow task.

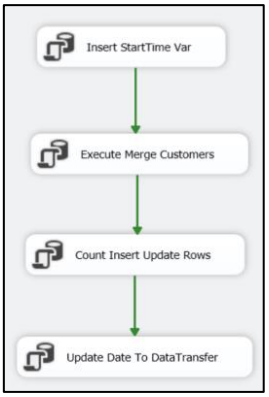


In the data flow, the 5 mirror tables (Mrr_Customers , Mrr_InvoicesDim, Mrr_Destcodes, Mrr_States, Mrr_Countries) are joined, and the data is loaded to STG_Customers table.



- DIM_Customers package:

  Data is loaded and into Dim_Customers.

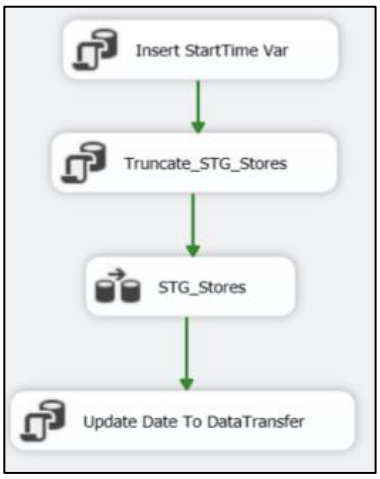


A merge stored procedure is executed in the Execute SQL Task, the merge statement works according to the following rational:

1. For **new** Customers (Customer_ID exists in STG_Customers but not in DIM_Customers): Insert new records to Dim_Customer and update columns UpdateDate and IsActive.
2. For **updated** Customers (Customer_ID exists in STG_Customers and in DIM_Customers, but one or more of the other columns does not match): Update the record in DIM_Customers.
3. For **deleted** Customer (Customer_ID exists in DIM_Customers but not in STG_Customers): Update the IsActive column in DIM_Customers to '0'.

- **Dim_Stores Table**:
  - STG_Stores package:

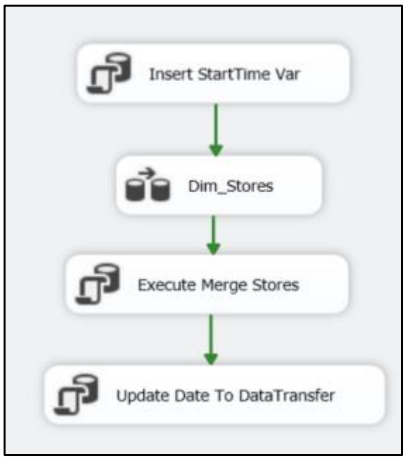
    STG_Stores table is truncated, and the mirror tables are joined and loaded using a data flow task.



In the data flow, the data is loaded into STG_Stores table.

- DIM_Stores package:

  Data is loaded and updated in Dim_Stores.

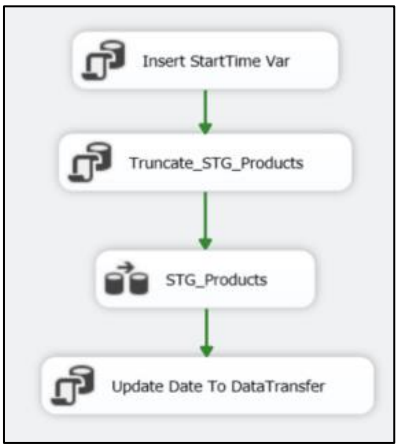


A merge stored procedure is executed in the Execute SQL Task, the merge statement works according to the following rational:

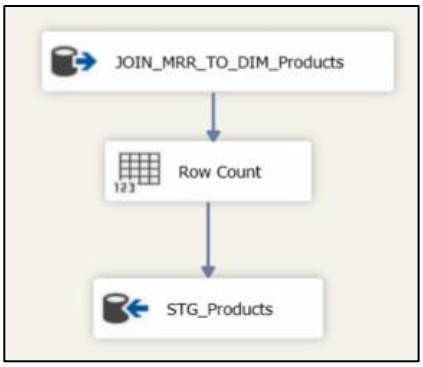
4. For **new** Stores (Store_ID exists in STG_Stores but not in DIM_Stores): Insert new records to Dim_ DIM_Stores and update columns UpdateDate and IsActive.
5. For **updated** Stores (Store_ID exists in STG_Stores and in DIM_Stores, but one or more of the other columns does not match): Update the record in DIM_Stores.
6. For **deleted** Stores (Store_ID exists in DIM_Stores but not in STG_Stores): Update the IsActive column in DIM_Stores to '0'.

- STG_Products package:

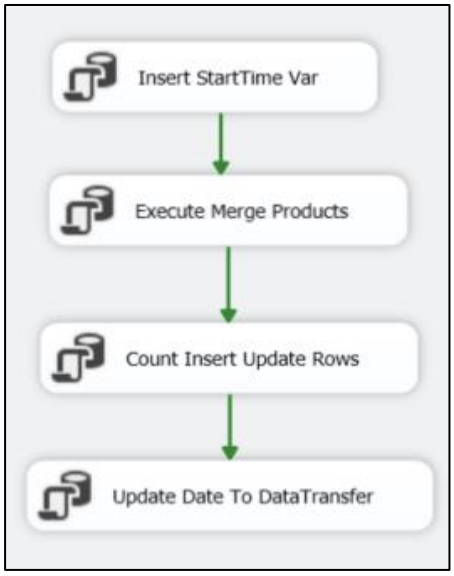
  STG_Products table is truncated, and the mirror tables are joined and loaded using a data flow task.



  In the data flow, the 3 mirror tables (MRR_Part, MRR_Family, MRR_FamilyTypes) are joined, and the data is loaded to STG_Products table.



- DIM_Products package:
  Data is loaded and updated in Dim_Products.



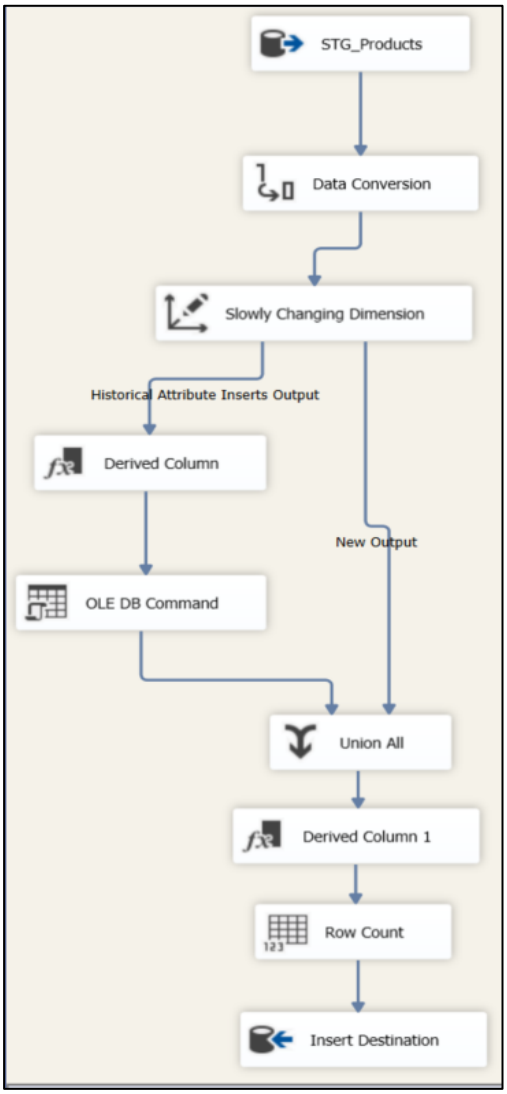
A merge stored procedure is executed in the Execute SQL Task, the merge statement works according to the following rational:

7. For **new** Products (Product exists in STG_Products but not in DIM_Products): Insert new records to DIM_Products and update columns UpdateDate and IsActive.
8. For **updated** Products (Product_ID exists in STG_Products and in DIM_Products, but one or more of the other columns does not match): Update the record in DIM_Products.
9. For **deleted** Stores (Product_ID exists in DIM_Products but not in STG_Products): Update the IsActive column in DIM_Products to '0'.

- **Dim_Products_History Table**:
  - DIM_Products_History package:

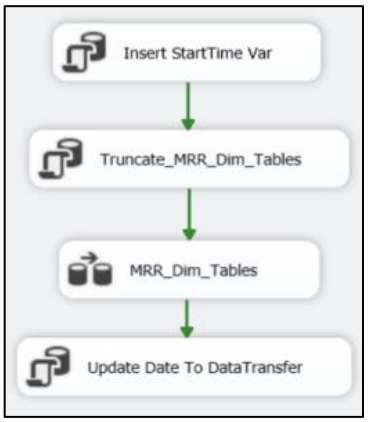
    Data is loaded and updated in Dim_Products_History.

In the Data Flow, Incremental load to the Dim_Products_History table is done using the Slowly Changing Dimension transformation (change type: Historical Attribute)-SCD will treat INSERT and UPDATE changes. 3 columns data types are converted to Unicode (DT_WSTR), one column to four-bite signed integer (DT_14) and one column to numeric (DT_NUMERIC) using data conversion transformation to match destination data types. An SQL statement will updated the EndDate column of a deleted data.



- **All Dim Tables – Mirror step**
  - MRR_Dim_Tables package:

    This package is responsible for loading data from PriorityERP tables to all mirror tables relevant for the dim tables (9 tables in total-another 3 tables are in additional packages). All the mirror tables are truncated using a stored procedure.



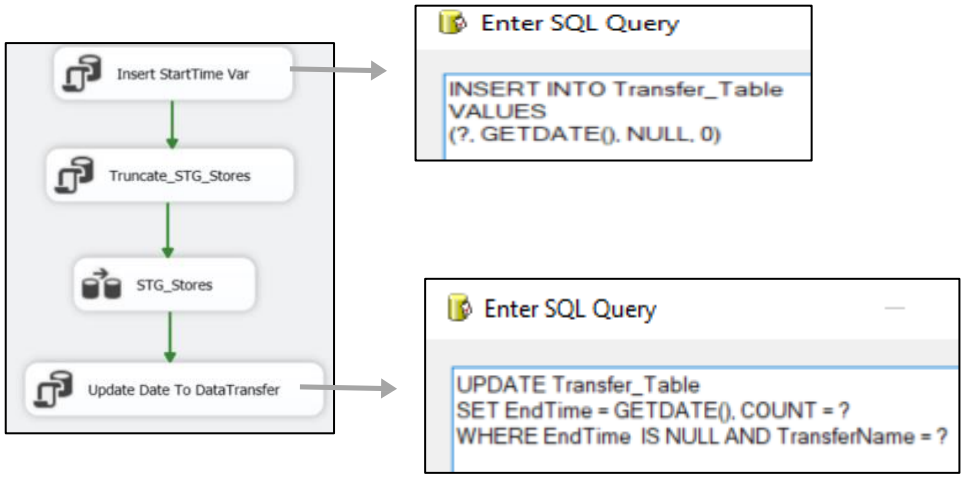
In the data flow:

- **Transfer_Table**

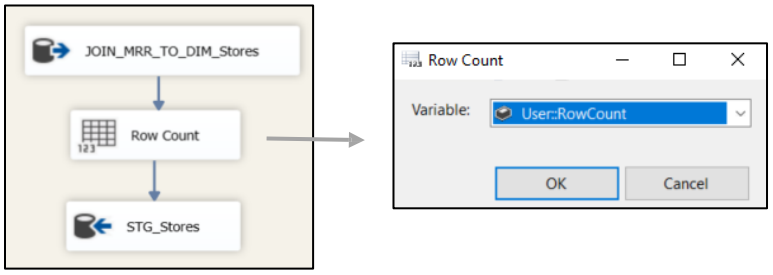
  To monitor the ETL process, a Data Transfer table was created documenting each data insert: which table was updated and in which package, how many rows were inserted, and start and end times. The tasks and transformation in charge of the updates are included in all of the packages.

  Example from STG_Stores package:

  In the control flow user variable StartTime is updated in the first task, and an insert statement is executed in the last task, inserting the values of the user variables: PackageName, TableName, StartTime, RowCount (which is updated in the data flow), with GETDATE() as EndTime.

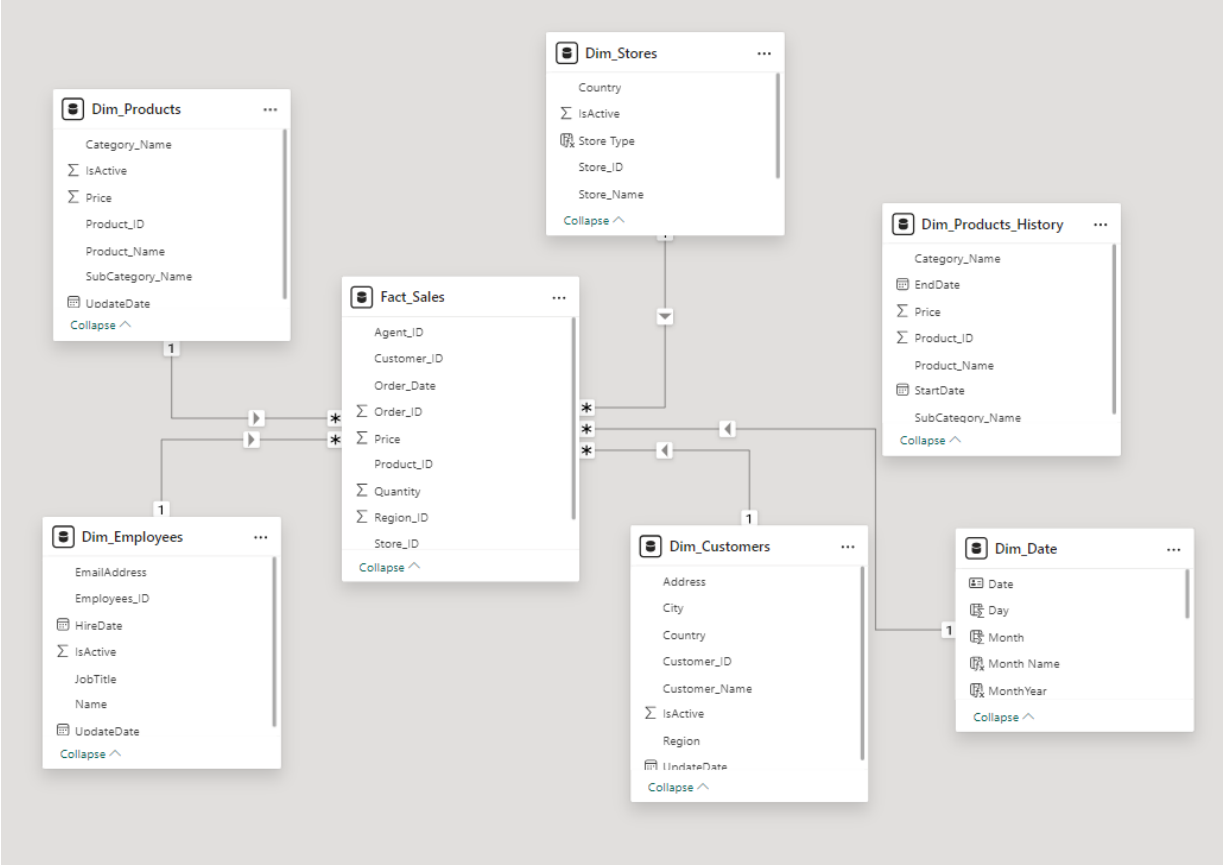


  In the data flow, the user variable RowCount is updated using a Row Count transformation.



4.3. **Visualization in Power BI:**

4.3.1. The reports were created using Power BI Desktop and were published to Power BI Service. The model in the Power BI includes the Fact table and the 6 Dimension tables (including the product history table). To these tables, a Dim_Date table was added, together with a MeasuresTable:

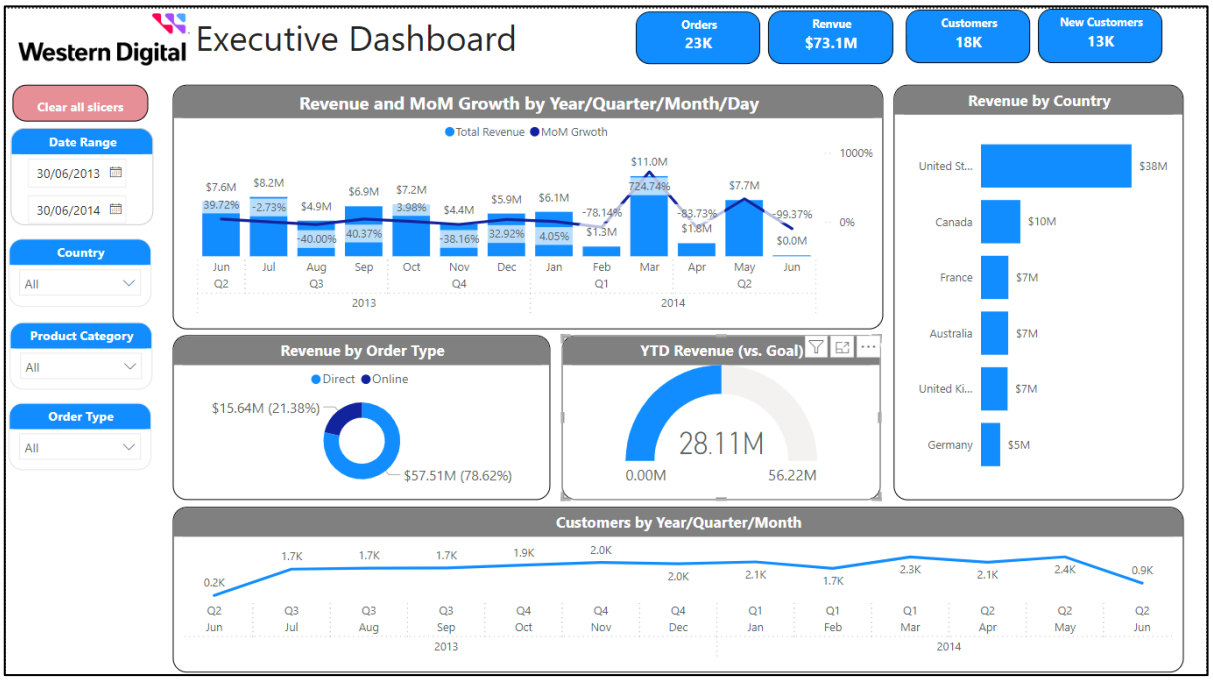


4.3.2. Reports:

The project includes 3 reports: Executive Dashboard, Sales Analysis, Customer Analysis.

4.3.2.1. Executive Dashboard:

This report was created to provide a broader look at the company's status, it includes the main KPIs, sales performance vs. goals, and general graphs.



KPI Cards:

- Orders
- Revenue
- Customers
- New Customers %

Graphs:

- Revenue and MoM Growth by Year/Quarter/Month
- YTD Revenue compared to yearly goal – goal and max value on the gauge
- Revenue by Order Type
- Revenue by Country
- Customers by Year/Quarter/Month

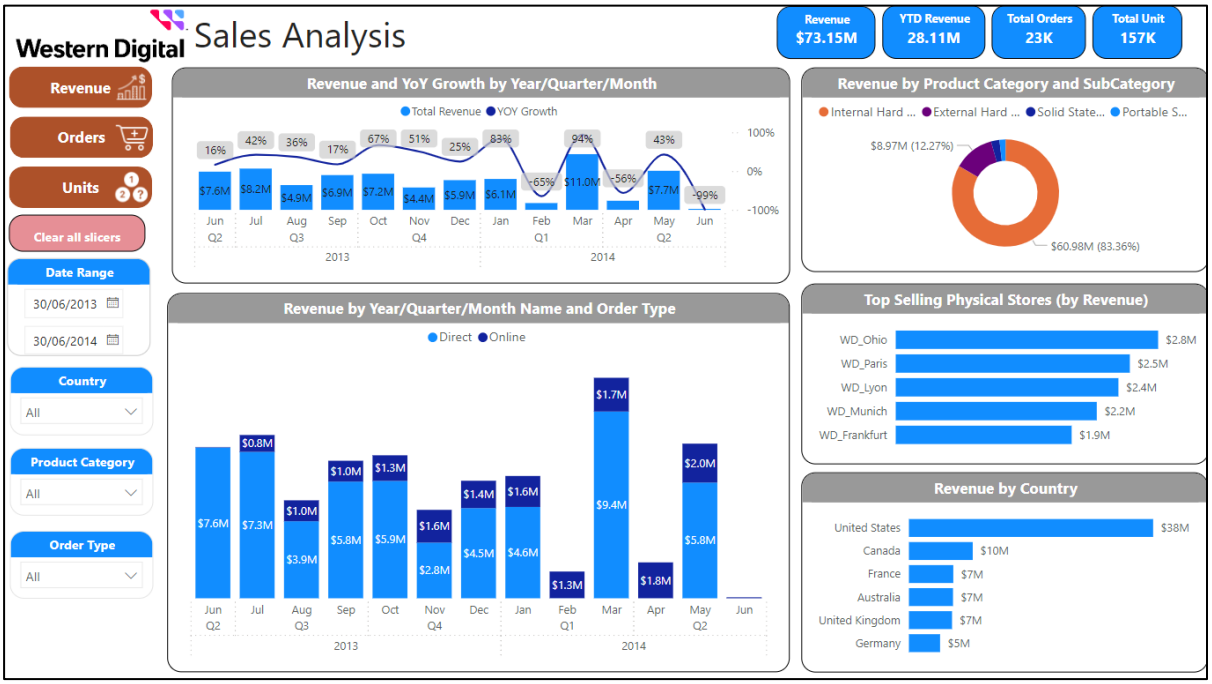
Slicers:

- Date Range
- Country
- Product Category
- Order Type

4.3.2.2. Sales Analysis:

This report was created for the sales department to follow and understand sales performance to achieve the department's goals.
In its initial state, the graphs present revenue data. Using the three buttons on

the to right, the user can control the data shown in the graphs and change it to orders data, and units data.



KPI Cards (same for all 3 states):

- Revenue
- YTD Revenue
- Total Orders
- Total Units

Slicers (same for all 3 states):

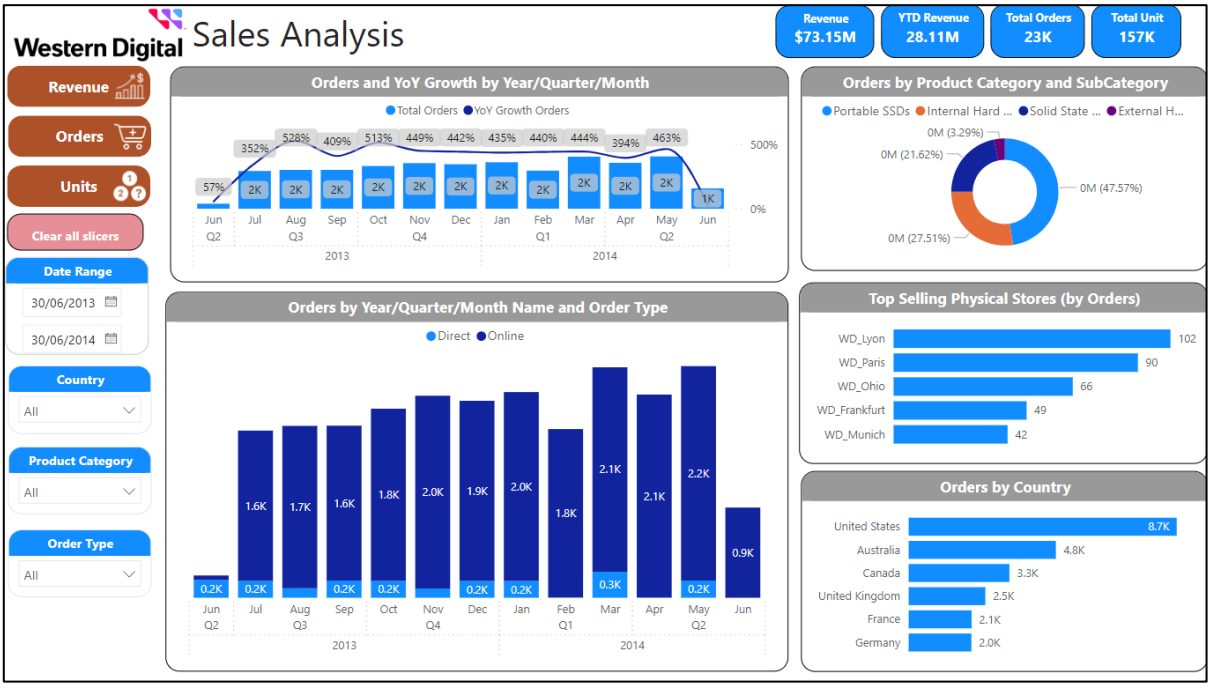
- Date Range
- Country
- Product Category
- Order Type

Graphs (revenue state):

- Revenue and YoY Growth by Year/Quarter/Month
- Revenue by Year/Quarter/Month and Order Type
- Revenue by Product Category/SubCategory.

- Top 5 Selling Physical Stores (by Revenue)
- Revenue by Country

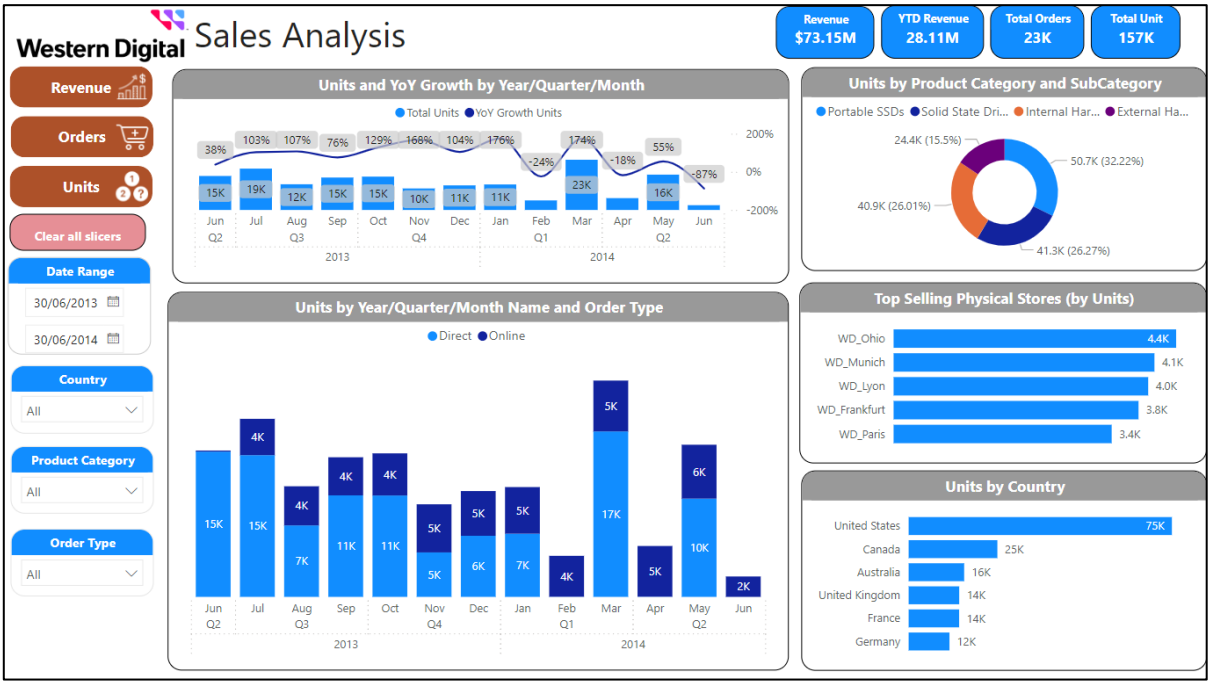
Orders State:



Graphs (orders state):

- Orders and YoY Growth by Year/Quarter/Month
- Orders by Year/Quarter/Month and by Store Type
- Orders by Product Category
- Top 5 Selling Stores (by Orders)
- Orders By Country

Units State:

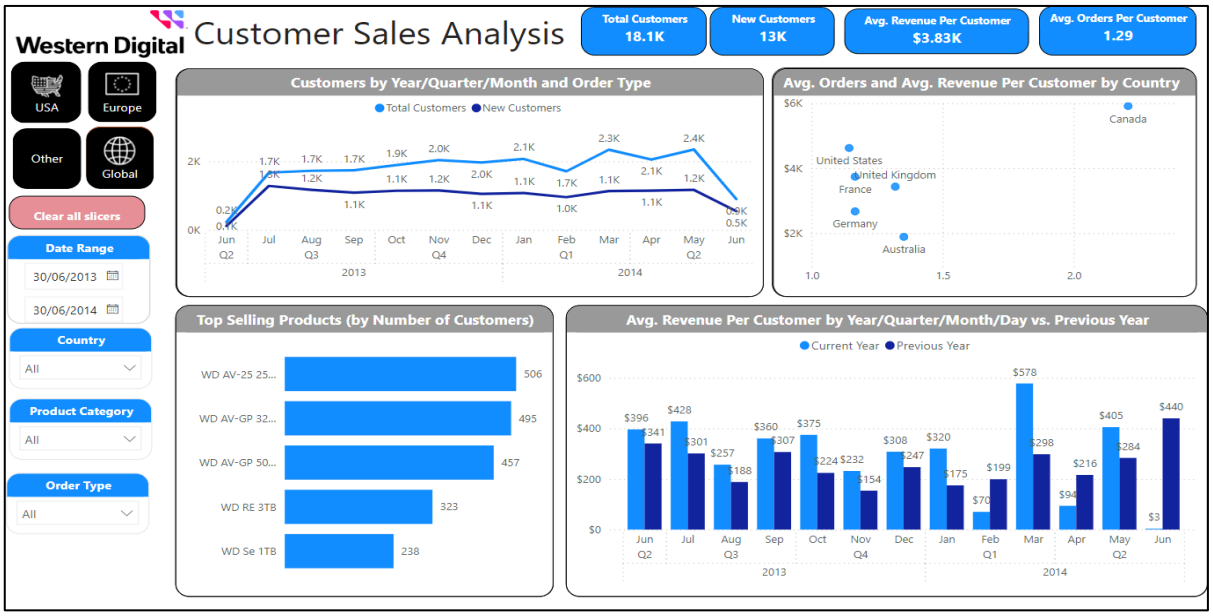


Graphs (units state):

- Units and YoY Growth by Year/Quarter/Month
- Units by Year/Quarter/Month and by Store Type
- Units by Product Category
- Top 5 Selling Stores (by Units)
- Units By Country

4.3.2.3. Customer Sales Analysis:

This report was created for the customers department to better understand WD's customer behavior to achieve the department's goals.



KPI Cards:

- Total Customers
- New Customers
- Average Revenue per Customer
- Average Orders per Customer

Graphs:

- Customers by Year/Quarter/Month
- Average Orders per Customer and Average Revenue Per Customer by Country – Color gradient by Total Customers
- Top 5 Selling Products by number of customers.
- Average Revenue per Customer by Year/Quarter/Month vs. Previous Year

Slicers:

- Date Range
- Country
- Product Category
- Order Type

Using the 4 buttons on the top left, the user can slice the data according to major customer populations: USA, Europe (France, Germany, and United Kingdom), Other (Australia and Canda), and global – presenting data from all countries.