



Western Digital's Sales Data Mart

BI System Specifications Document

Date: 15/07/2024

Version: 4.00

Written By: Yuly Tayar

Content

1. General	3
1.1. Project Objective	3
1.2. Project Contents	3
2. Gantt	4
3. Technical Specification	5
3.1. Prerequisites	5
3.2. Solution Architecture	5
4. Functional Specification	7
4.1. Source to Target and ERD model	7
4.2. ETL process	7
5. Visualization in Power BI	27
5.1. Model	27
5.2. Dashboard & Reports	28
5.3. Published to Power BI Service	33
5.4. Data refresh processes	34

1. General

1.1. Project Objective

This project's objective is to create a full BI solution for Western Digital's sales department, to support the growth of their product sales. The project was designed according to Western Digital's sales department. Western Digital Corporation (WDC, commonly known as Western Digital or WD) is an American computer drive manufacturer and data storage company, headquartered in San Jose, California. It designs, manufactures, and sells data technology products, including data storage devices, data center systems, and cloud storage services. Western Digital has a long history in the electronics industry as an integrated circuit maker and a storage products company. It is one of the largest computer hard disk drive manufacturers, along with producing solid-state drives and flash memory devices. Western Digital's devices are sold all over the world, through online stores, retailers, authorized resellers (via physical and online stores), and more. This project will focus only on sales performed by online stores and retailers.

The Data Mart creation will be done using information derived from the PriorityERP database (Western Digital's operational database). The solution will include summarized data tables, focusing on the product sales data, as well as data regarding Western Digital's customers, salespersons, products, and Retailers. In addition, the BI solution will include costumed reports containing sales analysis, customer analysis, and executive dashboard. These reports will be tailored to the sales department's needs and will contribute to Western Digital's product 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 – WesternDigital_DB.

The ERD of the tables is shown in the attached link: [ERD](#)

Source To Target document in this link: [S2T](#)

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

- Fact_Sales – Data regarding all sales, including the ID of the order, products bought, quantities, and prices. The data-loading process for this table will be incremental.
- Dim_Customers- Data regarding the company's customers.
- Dim_Retailers- Data regarding the company's Retailers.

- Dim_Employees- Data regarding the company's Salespersons.
- Dim_Products- Data regarding the company's products.
- Dim_Employees_History- Historical data regarding the company's Employees. The Employees' history table will be included to track the activity of the Employees over time using Slowly Changing Dimensions (SCD) Type 4.

The project will include a Transfer Table. This table will allow us to monitor data movement by showing the number of rows transferred and the time taken for each package in the ETL process. The tables will be updated daily at 05:00:00 using an automated process configured in the SQL Server Management Studio.

1.2.2. The reports will include data visualizations in Power BI 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, stores of retailers (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 product orders and revenue, spot top-performing salespersons, 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 customer analysis report will include data regarding WesternDigital's customers by date, country, retailer, product, and category. This report is aimed to help WesternDigital's customer department to better understand their customers' behavior, like what (products), where (countries and retailers stores), and when they shop. This is vital to retain current customers and reach new ones.

- Executive Dashboard:

The dashboard will provide a high-level summary of the company's overall performance, focusing on strategic insights and trends. It will offer a snapshot of sales performance, key growth indicators, and ROI metrics, allowing executives to quickly assess the company's performance. This dashboard will support strategic planning and drive the company's growth and success. it will include key visuals from the two reports and 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 dashboard and reports using Power BI.

3.2. Solution Architecture

3.2.1. High-Level Design:



Data collection and exploration from the ERP system will be performed in SQL Server. The data will undergo an ETL process for organization and arrangement into a Data Mart using SSIS. Finally, the presentation of measures in reports and visuals will be presented in Power BI.

3.2.2. Power BI Reports:

3.2.2.1. The **Executive Dashboard (Sales Overview)** will consist of:

- YTD sales
- YTD units
- YTD orders
- YTD customers
- Growth to LY %
- Average Order Value
- Average Order Value by Country
- Total Sales by Order Type
- Total Sales by Country
- Top 5 Sales Persons by Sales
- YOY growth Sales/Orders/Units by year\

3.2.2.2. The report for the **Sales Department Analysis** will consist of:

- Total sales
- Total orders
- Total units
- Average Order Value
- Average Units Per Order
- Offline Stores VS Online Store by Sales/Orders/Units
- Sales/Orders/Units by Product Category and Subcategory
- Top 5 Products by Sales/Orders/Units
- YTD VS LYTD by Sales/Orders/Units
- Sales/Orders/Units MoM Growth by Month
- Sales Persons Details

3.2.2.3. The report for the **Customer Sales Analysis** will consist of:

- Total sales
- Total orders
- Total customers
- New customers
- Return customer rate
- Average Order Value
- Average revenue per customer
- Average order per customer
- Online Sales VS Offline Sales by number of customers and country
- Total customers by their order amount
- Top 5 Products by number of customers
- Type of Customers by year
- Total customers, total orders, and total sales 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](#)

4.1.2. ERD model of the AzureDB2 database

- [ERD link](#)

4.2. ETL processes

The ETL process was done in SSIS using 13 packages.

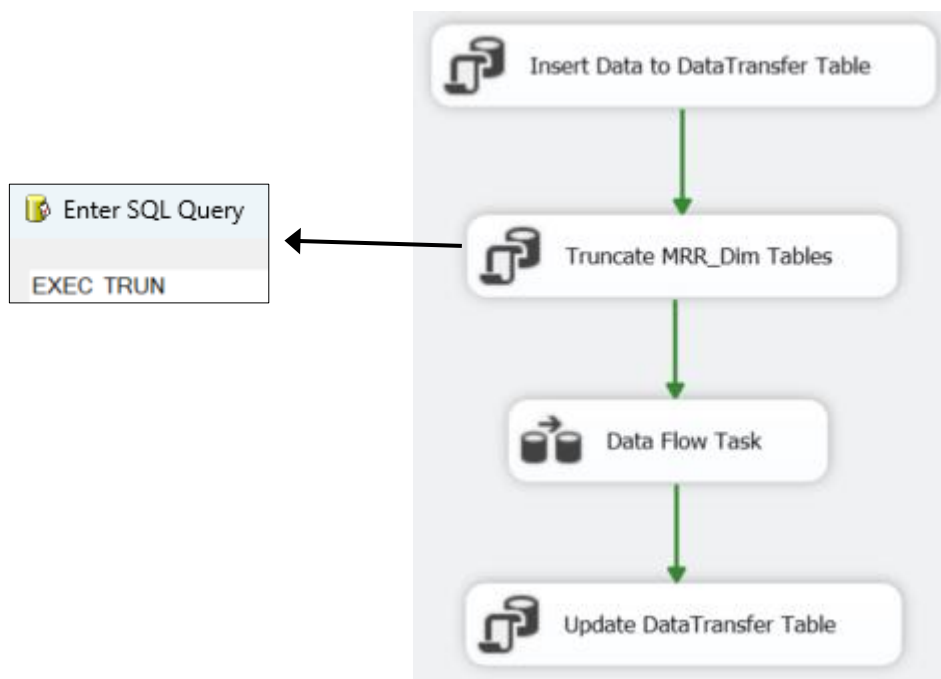
All the packages include 3 reoccurring Execute SQL tasks or transformations: 'Insert Data to DataTransfer table', 'Row Count', and 'Update DataTransfer table'.

*These will be explained later in the DataTransfer table section.

1. MRRforDIMs package:

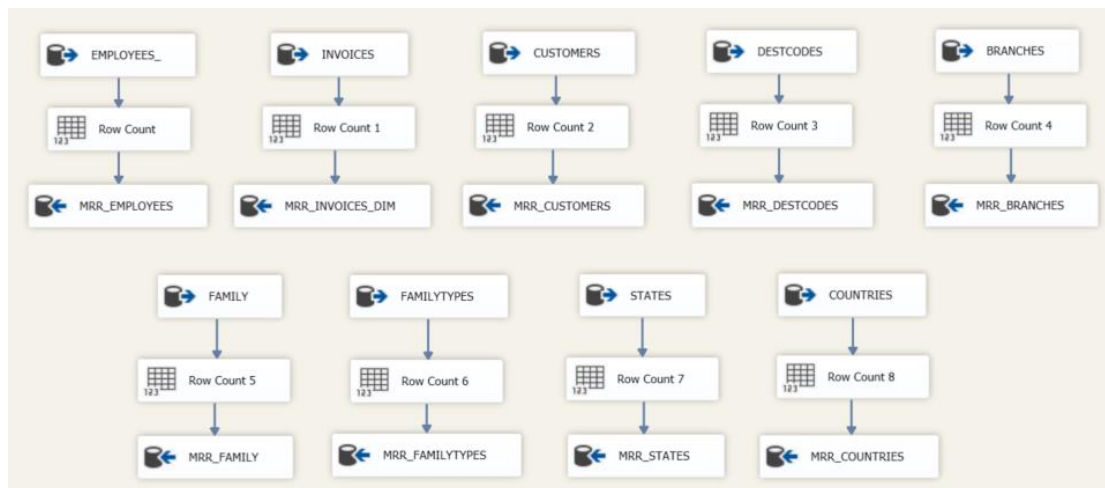
This package is responsible for loading data from PriorityERP tables to all mirror tables relevant to the dim tables (9 tables in total). All mirror tables are truncated using a stored procedure.

In Control Flow:



The Execute SQL Task- **Truncate MRR Dim Tables** is instructing the activation of a procedure called 'TURN' written in the SSMS. This stored procedure is designed to truncate (delete all rows) from multiple tables that were mentioned in the procedure.

In Data Flow:



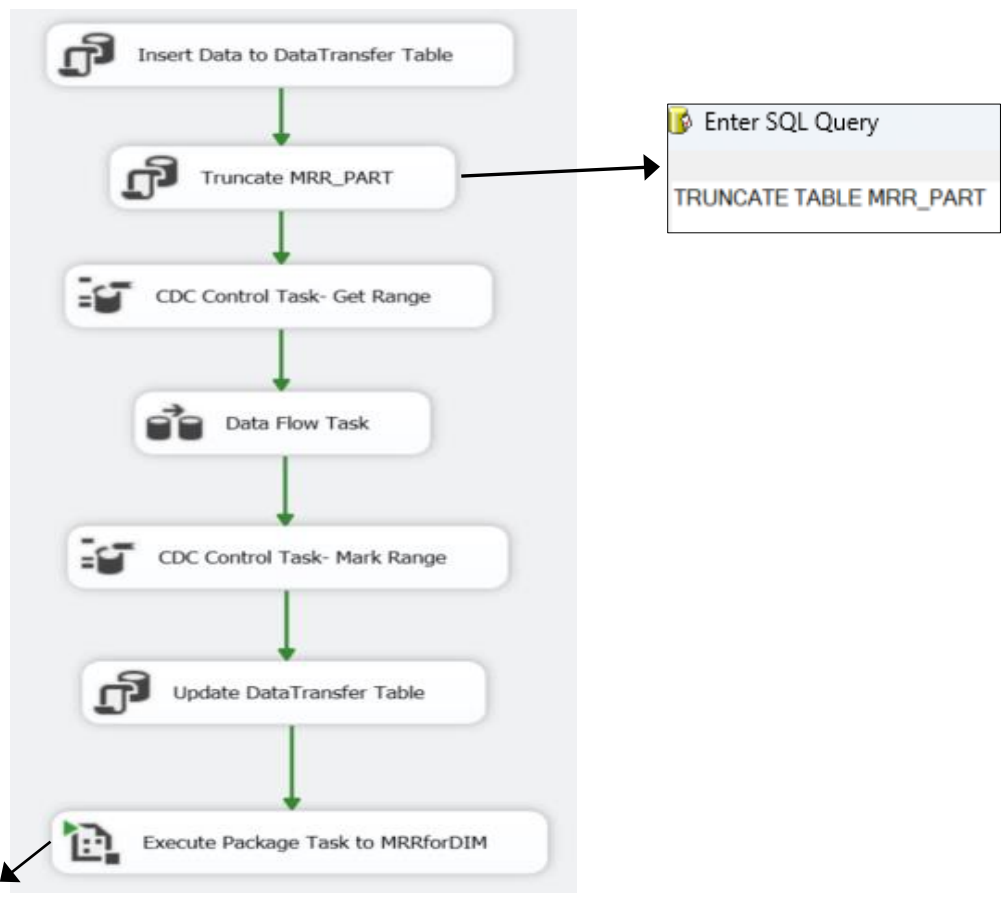
- **Dim Products Table:**

2. MRR_PART_CDC package:

Change Data Capture on PART Table in PriorityERP Database:

Change data capture (CDC) refers to the process of identifying and capturing changes made to data in the PART table and then delivering those changes in the ETL process.

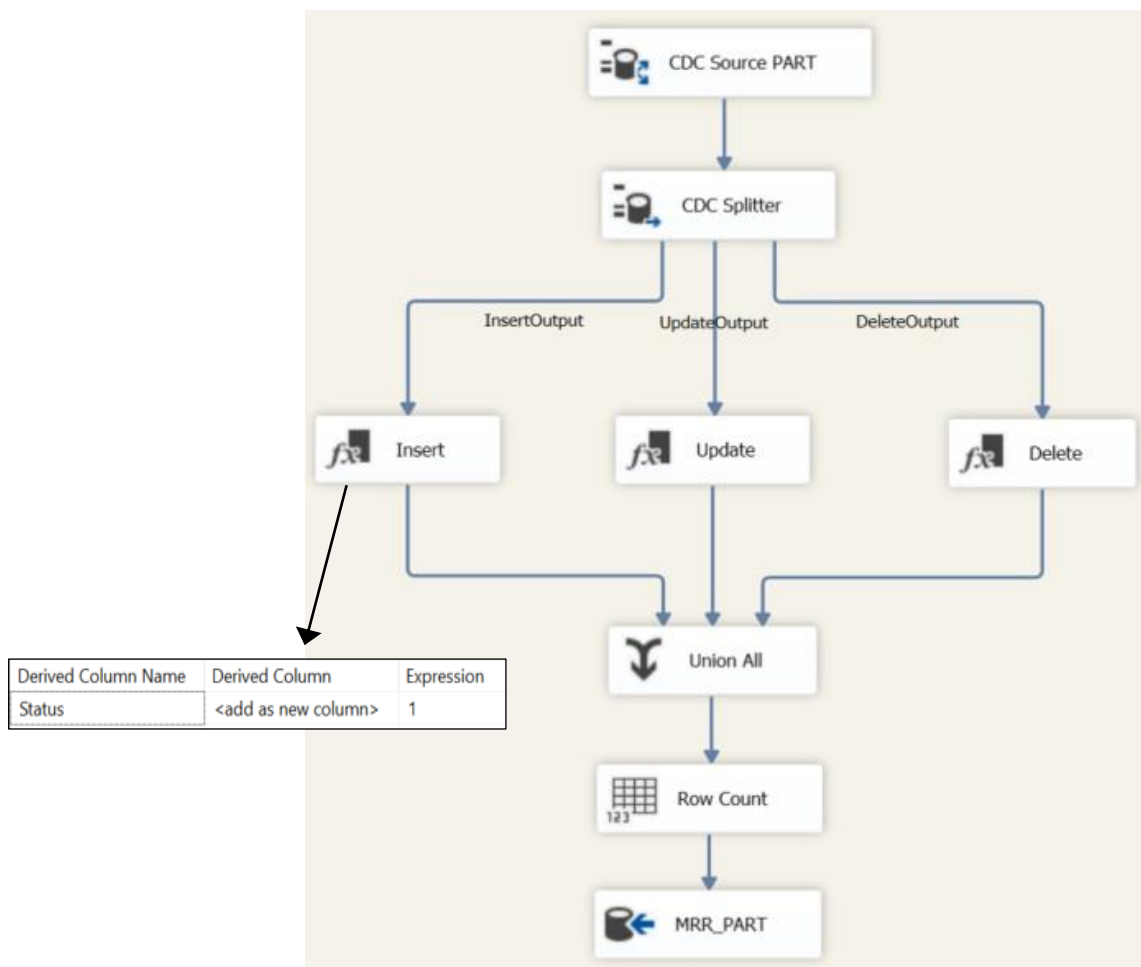
In Control Flow:



- **CDC Control Task- Get Range-** Calculating the time range from when to when we want to extract changes from the CDC tables.
- **CDC Control Task- Mark Range-** Indicates that the records in the defined time range have already been processed successfully and therefore will not be processed again the next time we run the process.
- **Execute Package Task to MRRforDIM-** Instructs to execute the package MRRforDIM. We do this because in this package other MRR tables are needed to create the final table- Dim_Products.

In Data Flow:

Retrieving all records from the CDC table for the previously calculated time range.

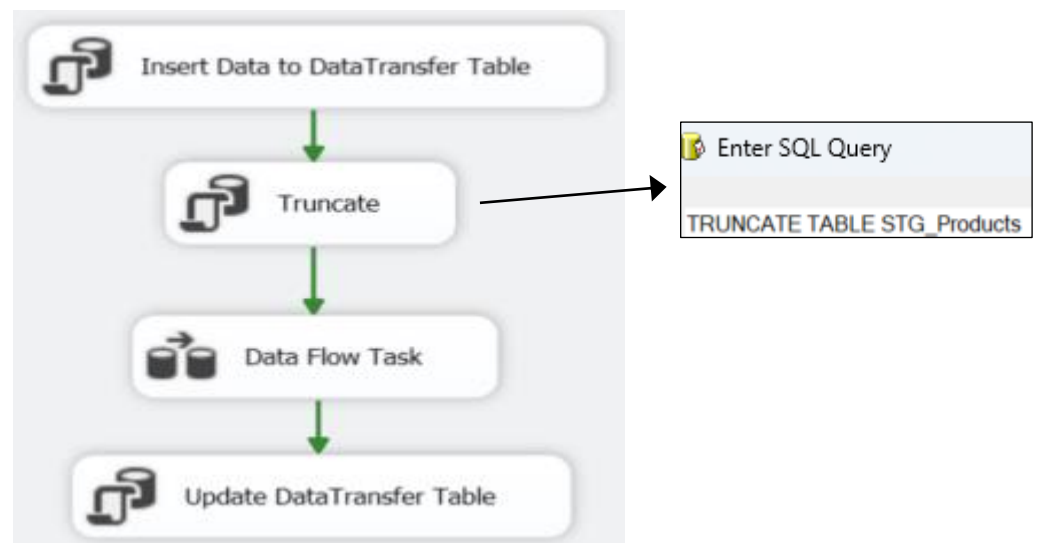


- **CDC Splitter** divides the operations into 3 types: Insert, Update, and Delete. We mark every operation type by adding the Status column:
Status = 1 --> INSERT, Status = 2 --> UPDATE, and Status = 3 --> DELETE.

3. STG_Products package:

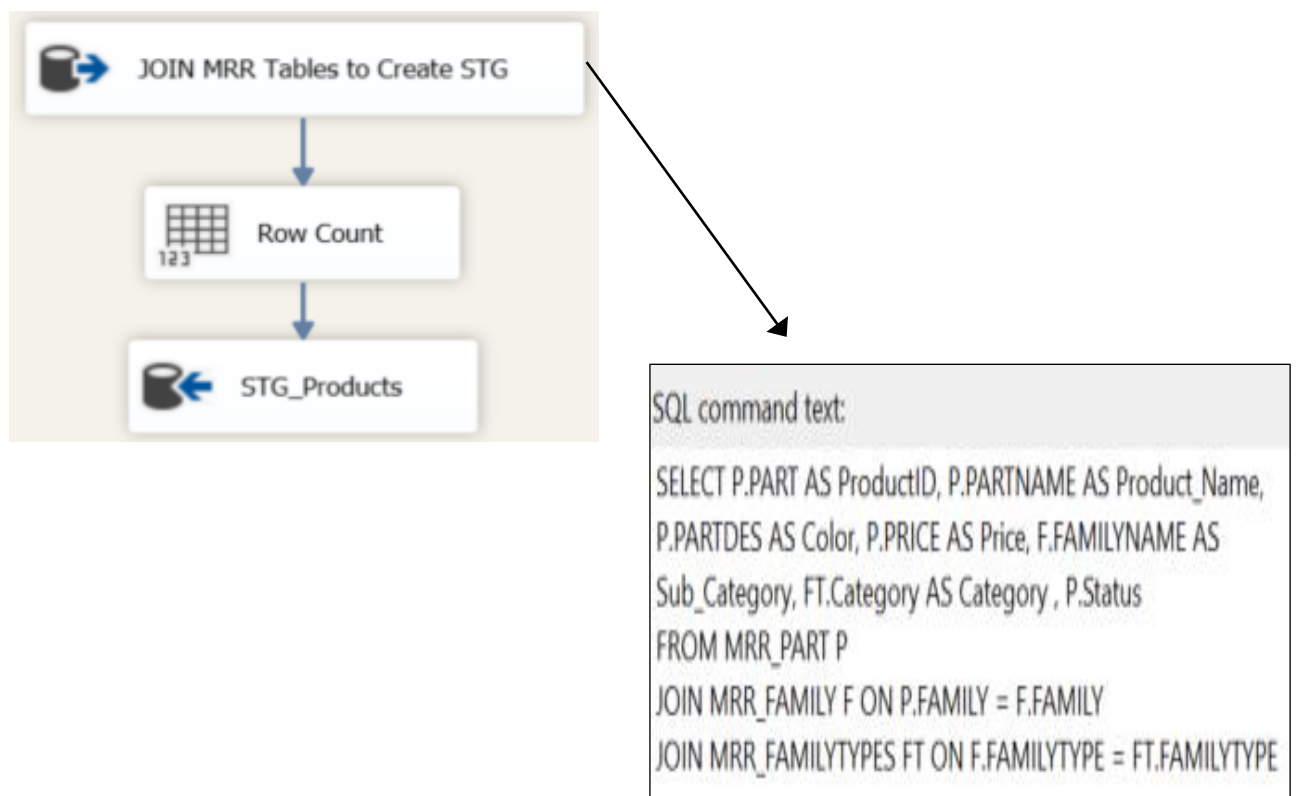
STG_Products table is truncated (all rows deleted), and the mirror tables (MRR) are joined and loaded using a data flow task.

In Control Flow:



In Data Flow:

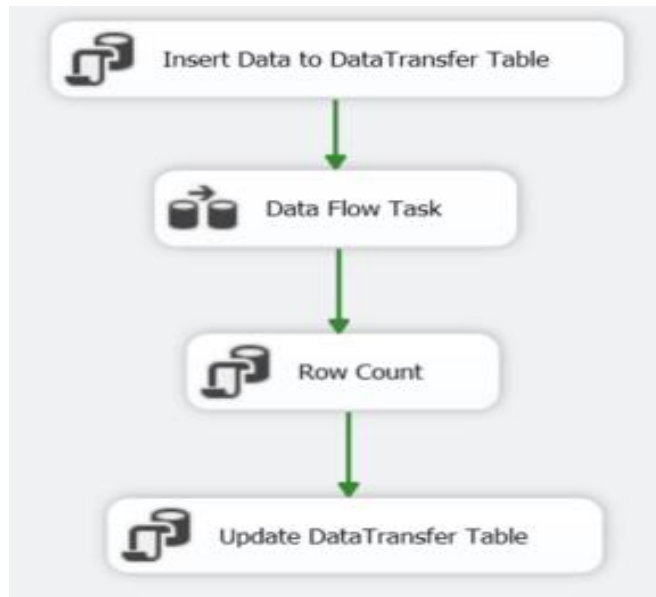
Collected product information from three tables: MRR_PART, MRR_FAMILY, and MRR_FAMILYTYPES. The selected columns include the details: ProductID, Product_Name, Color, Price, Sub_Category, Category, and Status (CDC). Finally, we will insert all the information into the STG_Products table with matching names using the command JOIN.



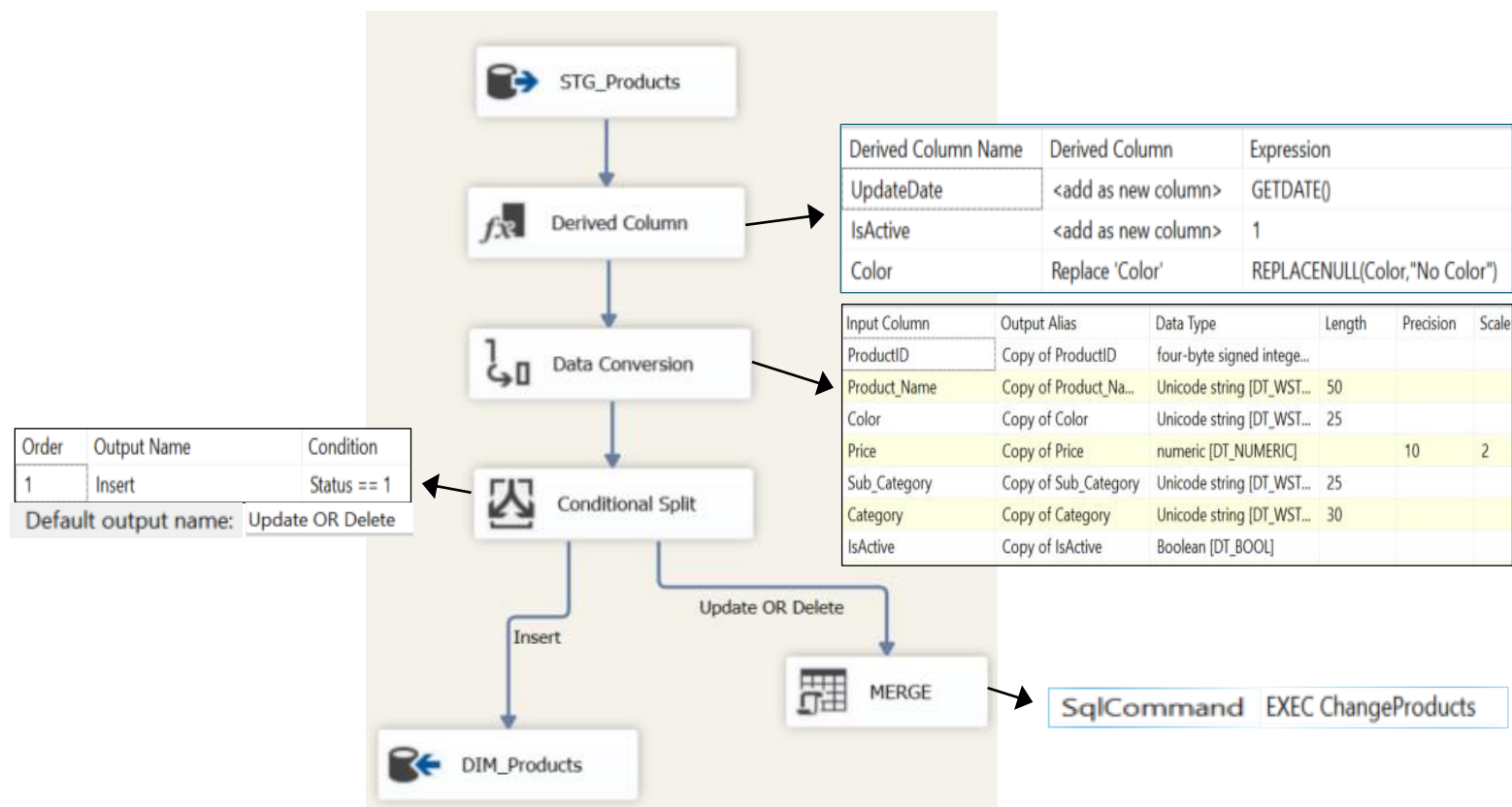
4. DWH_Products package:

By using CDC, the DIM_Products table gets only the rows changed in the operating database from the STG_Products table.

In Control Flow:



In Data Flow:



- **Conditional split-**

If Status = 1 then it means the ProductID in the row does not exist in the table- Insert a new row to DIM_Products with GETDATE() function in the UpdateDate column.

If Status = 2, then the row has been Updated- Update row by Merge query.

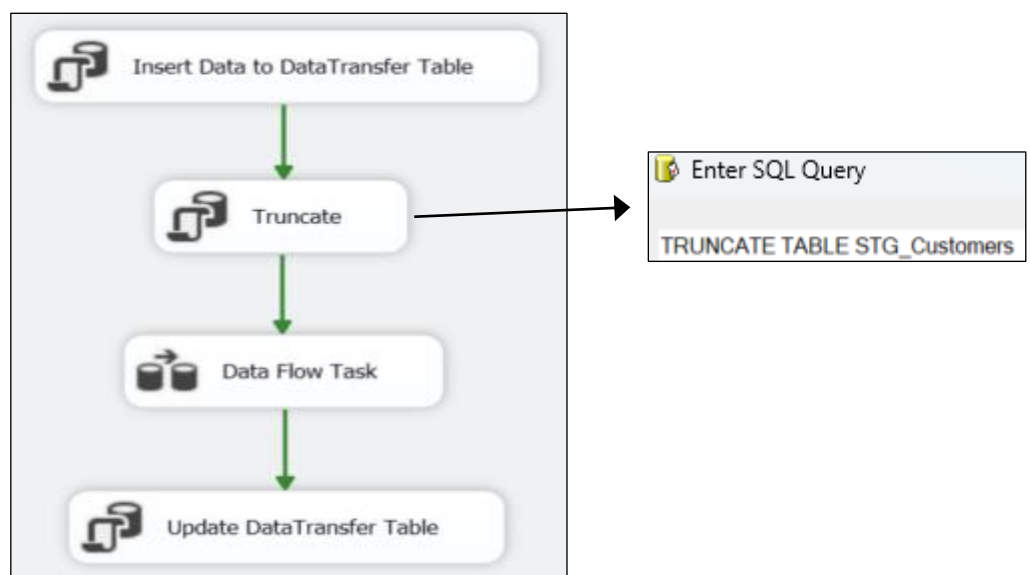
If Status = 3, then the row was Deleted- Update IsActive value to 0, and UpdateDate value to GETDATE() in the current row by Merge query.

- **Dim Customers Table:**

5. STG_Customers package:

The STG_Customers table is truncated (all rows are deleted), and the mirror tables (MRR) are joined and loaded using a data flow task.

In Control Flow:



In Data Flow:

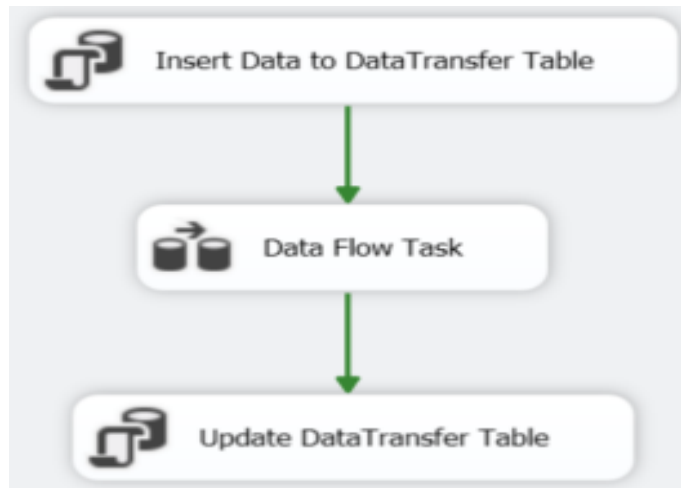
Collected Customer information from five tables: MRR_CUSTOMERS, MRR_INVOICES_DIM, MRR_DESTCODES, MRR_STATES, and MRR_COUNTRIES. The selected columns include the details: CustomerID, Name, Address, City, State, and Country. Finally, we will insert all the information into the STG_Customers table with matching names using the commands JOIN and RIGHT JOIN.



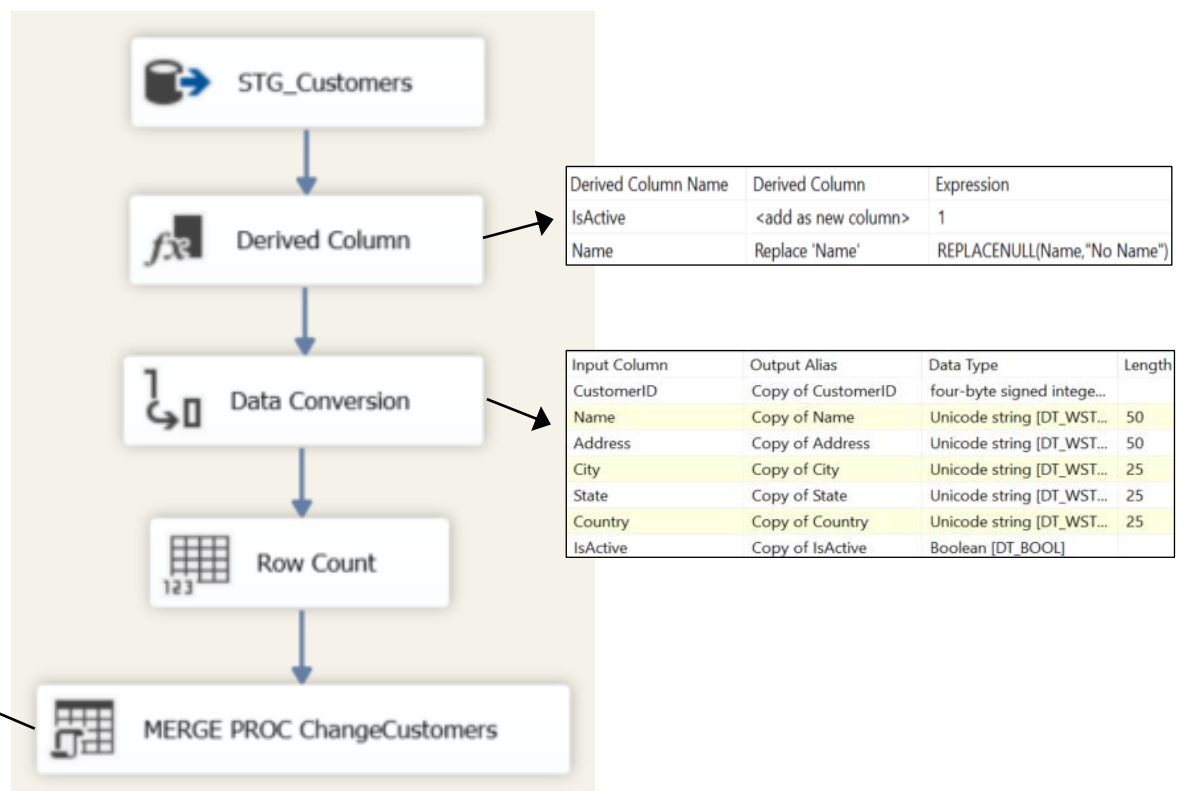
6. DWH_Customers package:

Data is incrementally loaded and updated in Dim_Customers.

In Control Flow:



In Data Flow:



- MERGE PROC ChangeCustomers-

The merge stored procedure 'ChangeCustomers' is executed in the OLE DB Command.

The merge statement works according to the following rationale:

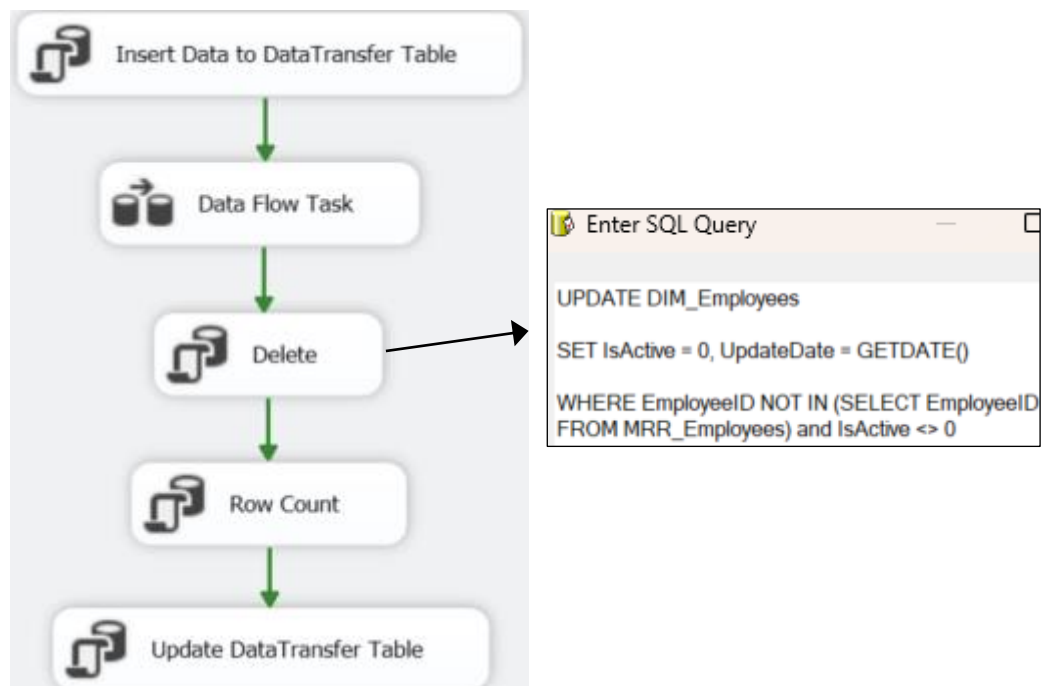
1. For **new** customers (CustomerID exists in STG_Customers but not in Dim_Customers): **Insert** new records to Dim_Customers.
2. For **updated** customers (CustomerID exists in STG_Customers and Dim_Customers but one or more of the other columns does not match): Update the record in Dim_Customers.
3. For **deleted** customers (CustomerID exists in Dim_Customers but not in STG_Customers): Updates the IsActive column in Dim_Customers to 0.

- **Dim_Employees Table:**

7. DWH_Employees package:

Data is incrementally loaded and updated in Dim_Employees using the Slowly Changing Dimension transformation (change type: Changing Attribute). Deleted records are updated in Dim_Employees using an Execute SQL task called 'Delete'.

In Control Flow:



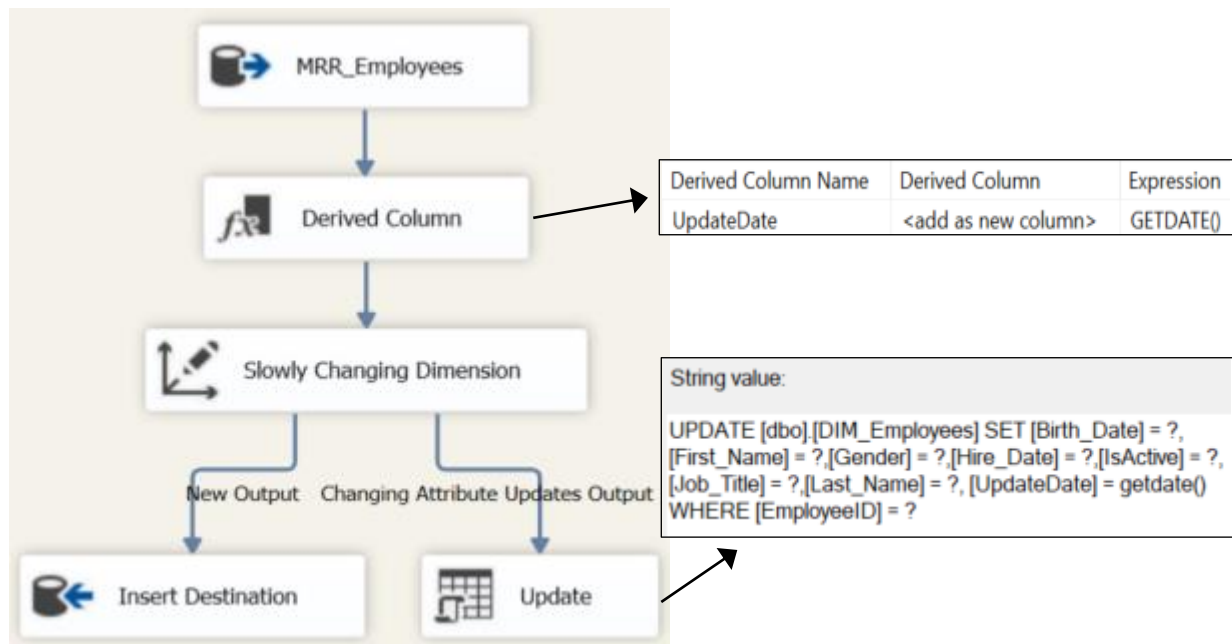
- **Delete-** updates records in the DIM_Employees table in the database. The general description of the query is as follows:
The query iterates over each record in the DIM_Employees table and updates the columns UpdateDate and IsActive. The 2 conditions for the update are: The EmployeeID column does not match one of the values returned by a subquery that retrieves all the EmployeeID from the MRR_Employees table, and the IsActive column should be different from 0.

So, when the values are updated, it means that the record identified (based on the criteria) that the EmployeeID is not found any longer (deleted) in the Employees table in the PriorityERP database and that the record has not been updated to 'deleted row' before.

After the update, the UpdateDate column is set to the current timestamp, and the IsActive column is set to 0.

We implemented this query as part of our Slowly Changing Dimension (SCD) strategy, which inherently manages new dates and updated data. Specifically, we created this query to address situations where records have been deleted. The query ensures that our DIM_Employees table accurately reflects changes in external data, including the handling of deleted rows.

In Data Flow:



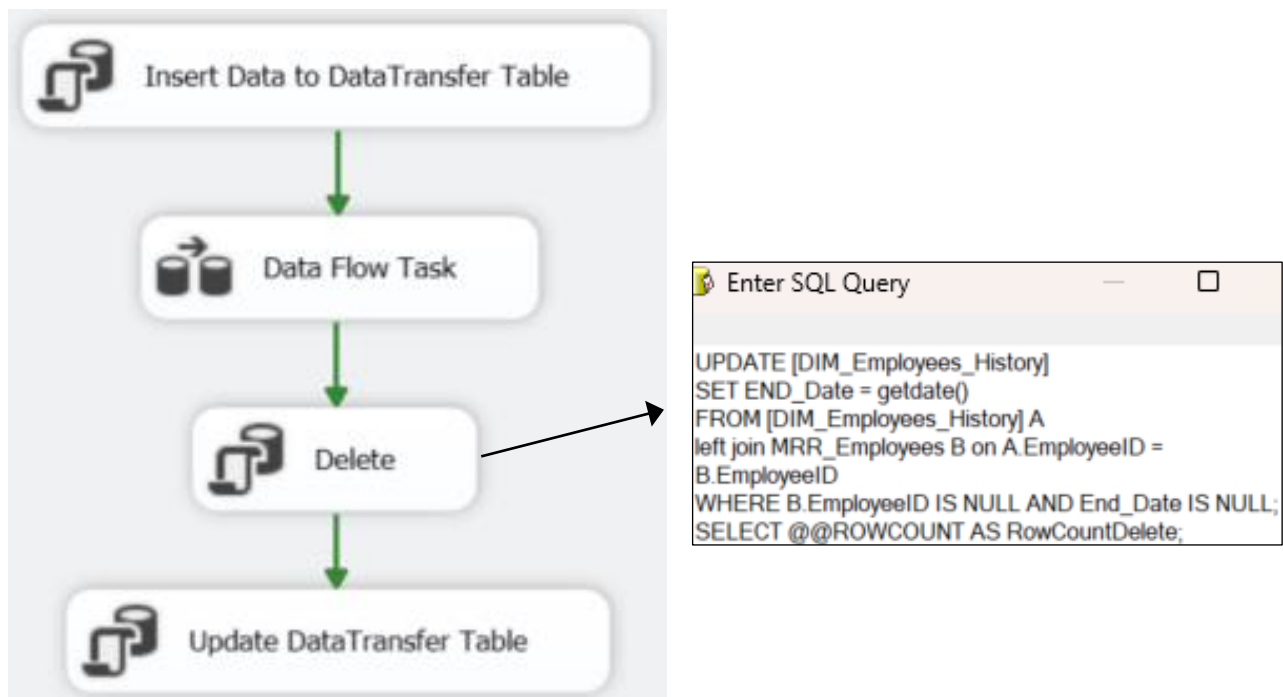
In the **Slowly Changing Dimensions** (SCD), the approach involves distinguishing between new rows and updated rows using the 'EmployeeID.' In the process of updating using the OLE DB Command called '**Update**', we focus on the changed fields from the MRR_Employees and update only them in the DIM_Employees. New rows, identified by a unique 'EmployeeID,' are handled separately to ensure proper management of evolving data.

- Dim_Employees_History Table:

8. DWH_Employees_History package:

Data is incrementally loaded and updated in Dim_Employees_History using the Slowly Changing Dimension transformation (change type: Historical Attribute). Deleted records are updated in Dim_Employees_History using an Execute SQL task called 'Delete'.

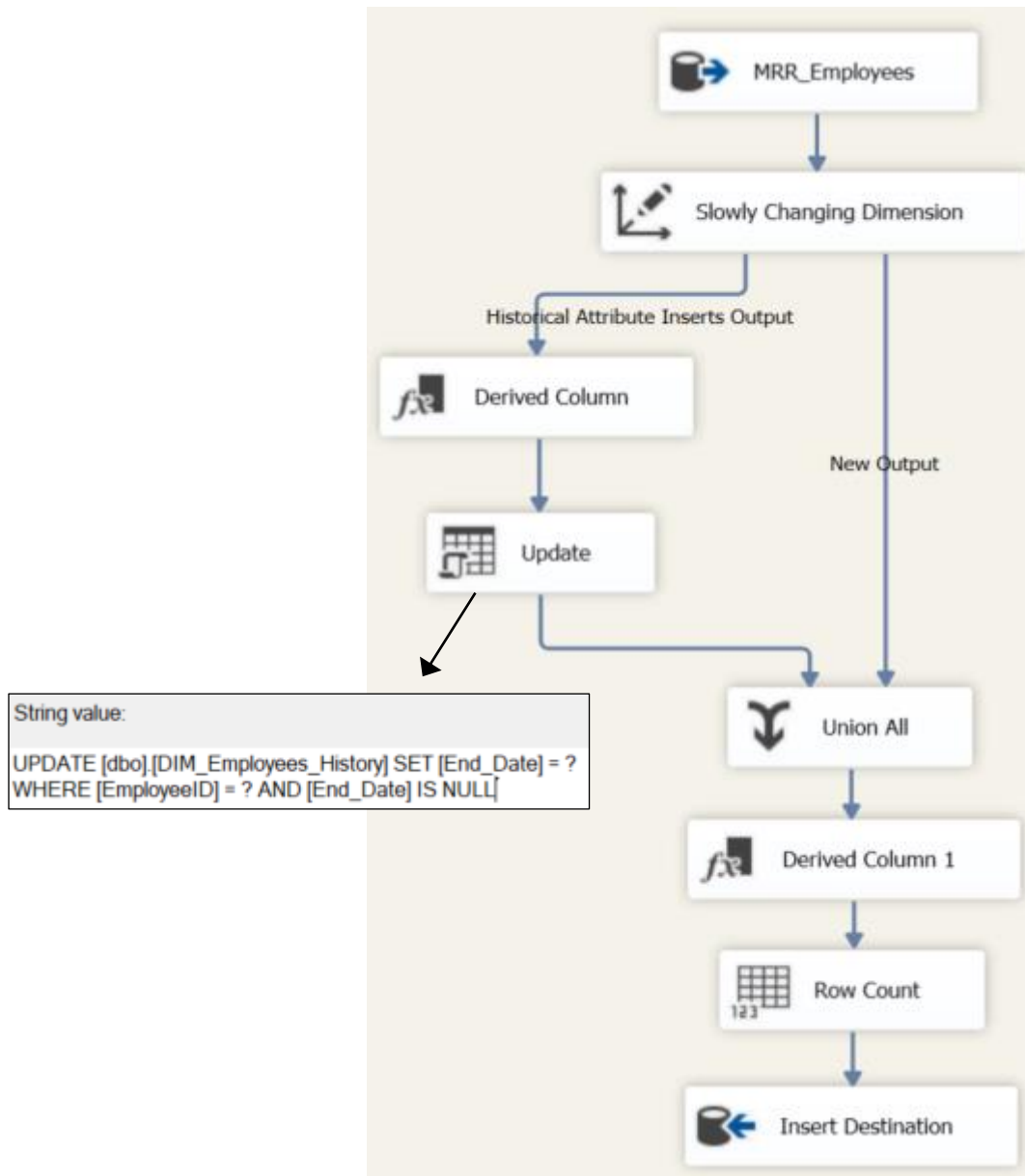
In Control Flow:



- Delete-

The query updates rows in the Dim_Employees_History table. The UPDATE statement modifies the End_Date column in the Dim_Employees_History table. It sets the End_Date to the current timestamp for records that meet the following conditions: The EmployeeID is not found in the MRR_Employees table, and the End_Date is NULL. In summary, the query effectively marks records in the history table as ended for Employees that are no longer present in the Employees_ table and have a NULL End_Date. This action helps differentiate between Employees actively working now in the company and those who do not.

In Data Flow:



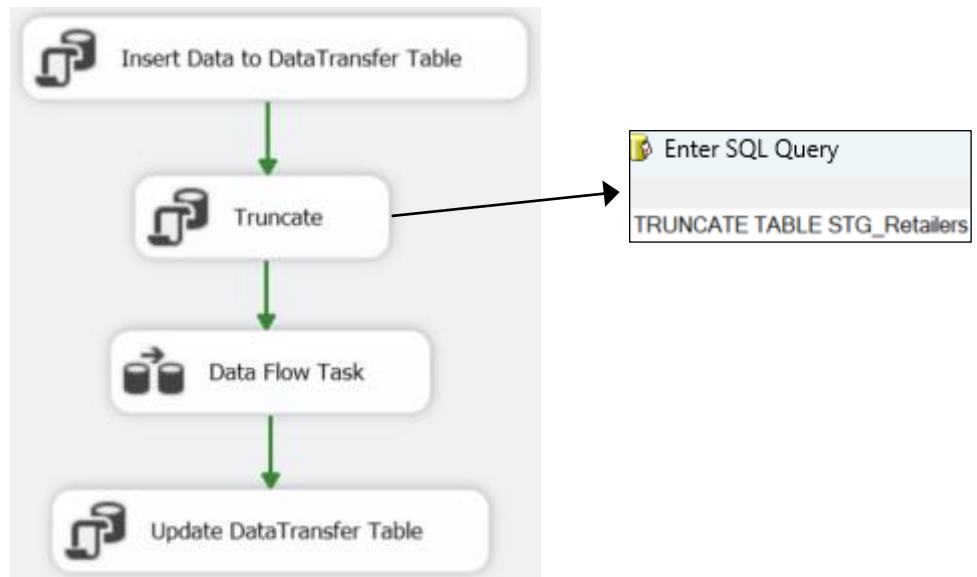
In the "DWH_Employees_History" package, when a record in the "Dim_Employees" table gets updated, we store its previous version in the "Dim_Employees_History" table. This historical table is structured like "Dim_Employees" but includes extra date fields indicating when each version began and ended. This approach enables us to track changes over time, providing a clear timeline for the evolution of product information in the data model. It is a valuable strategy for maintaining a historical record and ensuring transparency in understanding how Employee data has changed.

- Dim_Retailers Table:

9. STG_Retailers package:

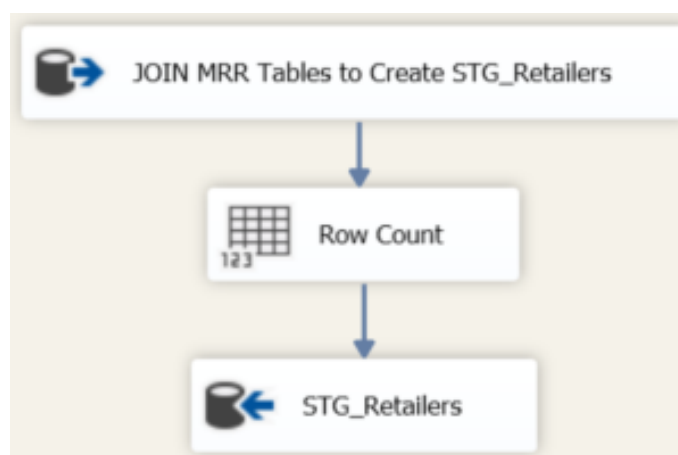
The STG_Retailers table is truncated (all rows are deleted), and the mirror tables (MRR) are joined and loaded using a data flow task.

In Control Flow:



In Data Flow:

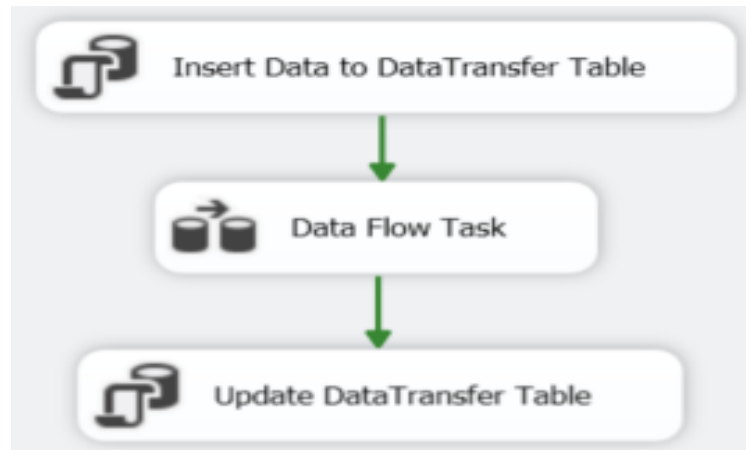
Collected Retailers' information from five tables: MRR_BRANCHES, MRR_INVOICES_DIM, MRR_DESTCODES, MRR_STATES, and MRR_COUNTRIES. The selected columns include the details: RetailerID, RetailerName, Country, and SalesPersonID. Finally, we will insert all the information into the STG_Retailers table with matching names using the command JOIN.



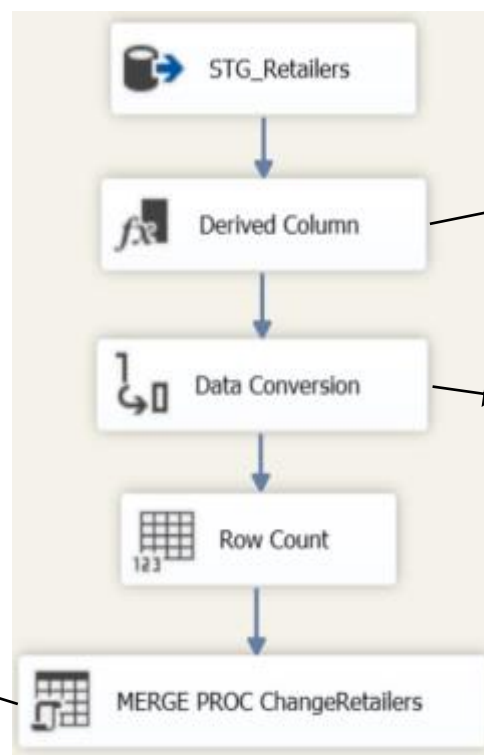
10. DWH_Retailers package:

Data is incrementally loaded and updated in Dim_Retailers.

In Control Flow:



In Data Flow:



Derived Column Name	Derived Column	Expression
IsOnlineStore	<add as new column>	RetailerID == 1 ? 1 : 0
IsActive	<add as new column>	1

Input Column	Output Alias	Data Type	Length
RetailerName	Copy of RetailerName	Unicode string [DT_WST...	45
Country	Copy of Country	Unicode string [DT_WST...	30
SalesPersonID	Copy of SalesPersonID	four-byte signed intege...	
IsOnlineStore	Copy of IsOnlineStore	Boolean [DT_BOOL]	
IsActive	Copy of IsActive	Boolean [DT_BOOL]	

String value:
EXEC ChangeRetailers

- MERGE PROC ChangeRetailers-

The merge stored procedure 'ChangeRetailers' is executed in the OLE DB Command.

The merge statement works according to the following rationale:

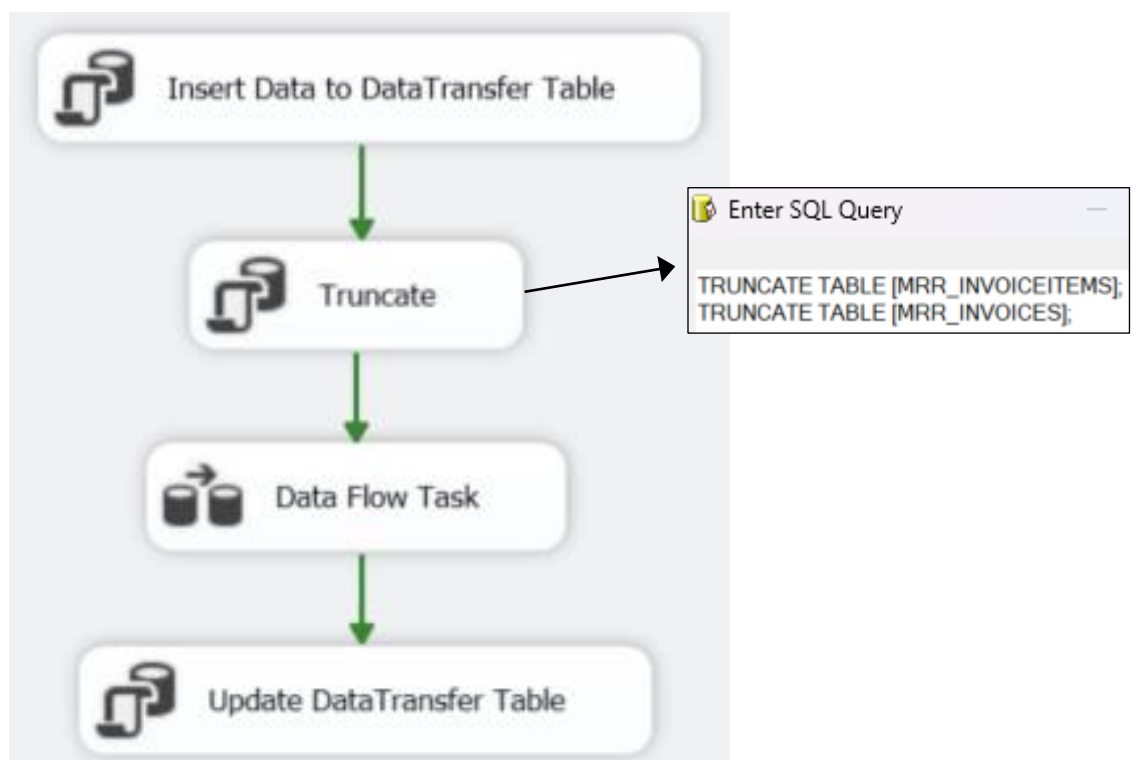
1. For **new** retailers (RetailerID exists in STG_Retailers but not in Dim_Retailers):
Insert new records to Dim_Retailers.
2. For **updated** retailers (RetailerID exists in STG_Retailers and Dim_Retailers but one or more of the other columns does not match): Update the record in Dim_Retailers.
3. For **deleted** retailers (RetailerID exists in Dim_Retailers but not in STG_Retailers):
Updates the IsActive column in Dim_Retailers to 0.

- **Fact Sales Table:**

11. MMRforFACT package:

This package is responsible for loading data from PriorityERP tables to the mirror tables relevant to the Fact_Sales table (MRR_INVOICES and MRR_INVOICEITEMS). The mirror tables are truncated (all rows deleted).

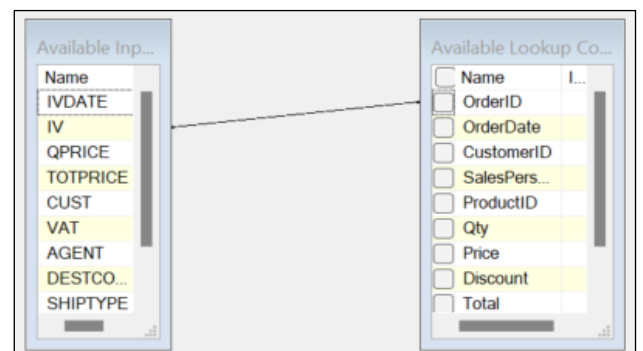
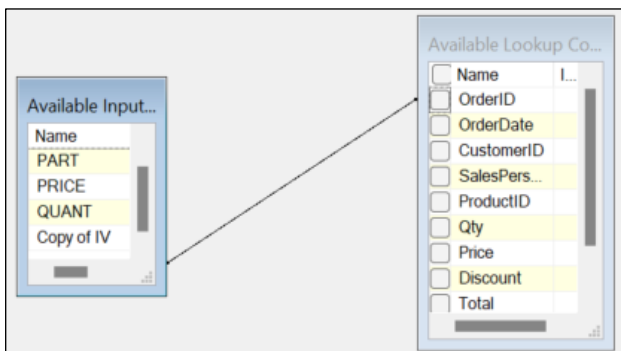
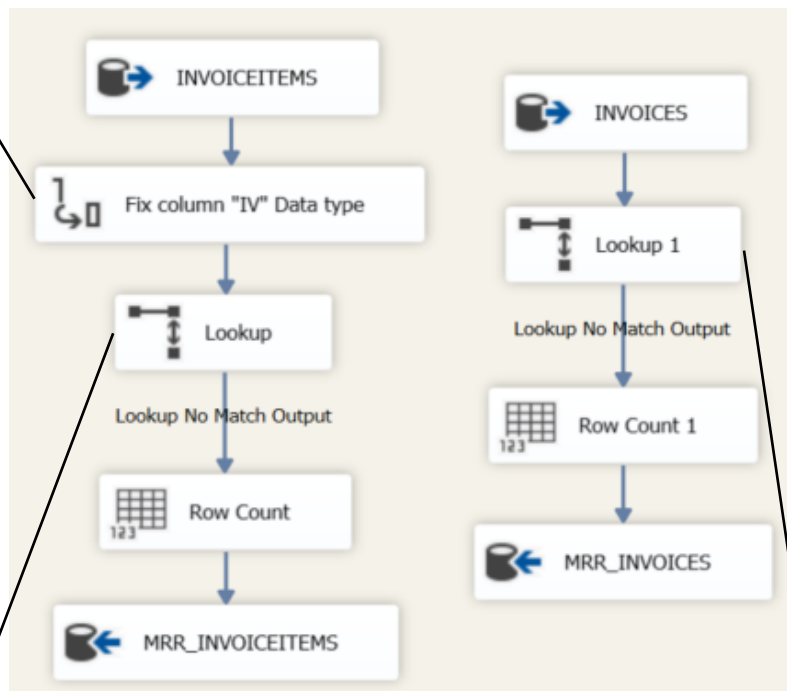
In Control Flow:



In Data Flow:

After one run of loading data to the Fact_Sales table, we added the Lookup transformation to this package to perform incremental loading.

Input Column	Output Alias	Data Type
IV	Copy of IV	four-byte signed intege...

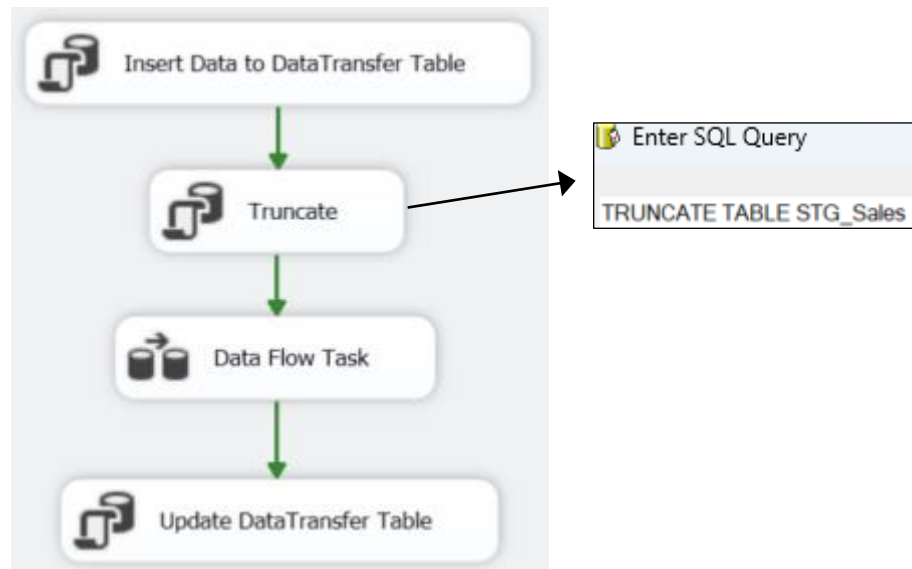


- **Fix column "IV" Data type** - Data Conversion transformation is used to convert the IV column (from the INVOICEITEMS table) from bigint to int (DT_I4), this is necessary for the Lookup transformation.
- **Lookup** -The Lookup transformation identifies the gap between the tables and loads only the new rows in the mirror tables that do not appear in the Fact_Sales table.

12. STG_Sales package:

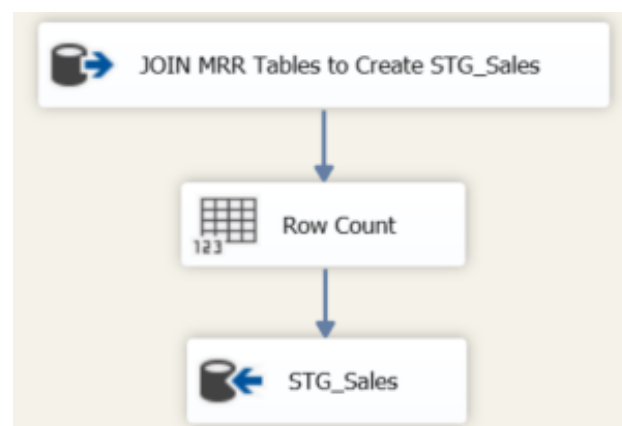
The STG_Sales table is truncated (all rows are deleted), and the mirror tables (MRR) are joined and loaded using a data flow task.

In Control Flow:



In Data Flow:

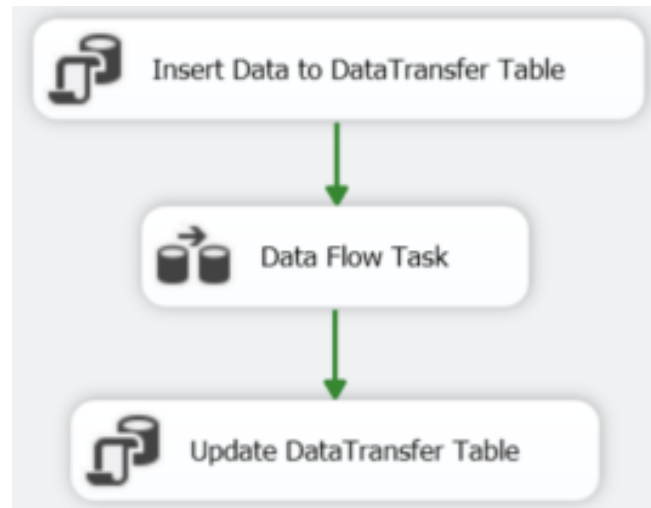
Collected sales order information from tables MRR_INVOICES and MRR_INVOCEITEMS. The selected columns include the details: OrderID, OrderDate, CustomerID, SalesPersonID, ProductID, Qty, Price, Discount, and IsOnlineStore. Finally, we will insert all the information into the STG_Sales table with matching names using the command JOIN.



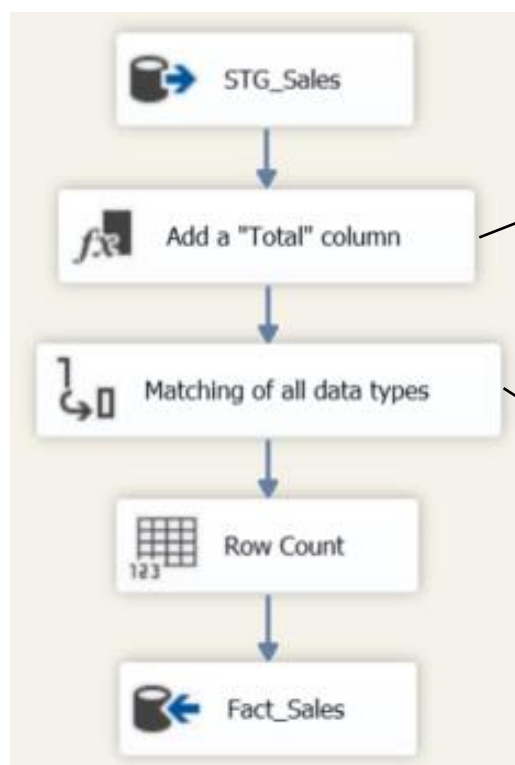
13. DWH_Sales package:

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

In Control Flow:



In Data Flow:



Derived Column Name	Derived Column	Expression
Total	<add as new column>	Price * (1 - Discount) * Qty * 1.17

Input Column	Output Alias	Data Type	Length	Precision	Scale
OrderID	Copy of OrderID	four-byte signed inte...			
CustomerID	Copy of CustomerID	four-byte signed inte...			
SalesPersonID	Copy of SalesPersonID	four-byte signed inte...			
ProductID	Copy of ProductID	four-byte signed inte...			
Qty	Copy of Qty	two-byte signed integer...			
IsOnlineStore	Copy of IsOnlineStore	Boolean [DT_BOOL]			
Price	Copy of Price	numeric [DT_NUMERIC]		10	2
Discount	Copy of Discount	numeric [DT_NUMERIC]		3	2
Total	Copy of Total	numeric [DT_NUMERIC]		15	2
OrderDate	Copy of OrderDate	database date [DT_DBD...			

Using the Derived Column transformation, we added a **Total** column for the Fact_Sales table with the formula: **Price * (1 - Discount) * Qty * 1.17**.

- **DataTransfer Table:**

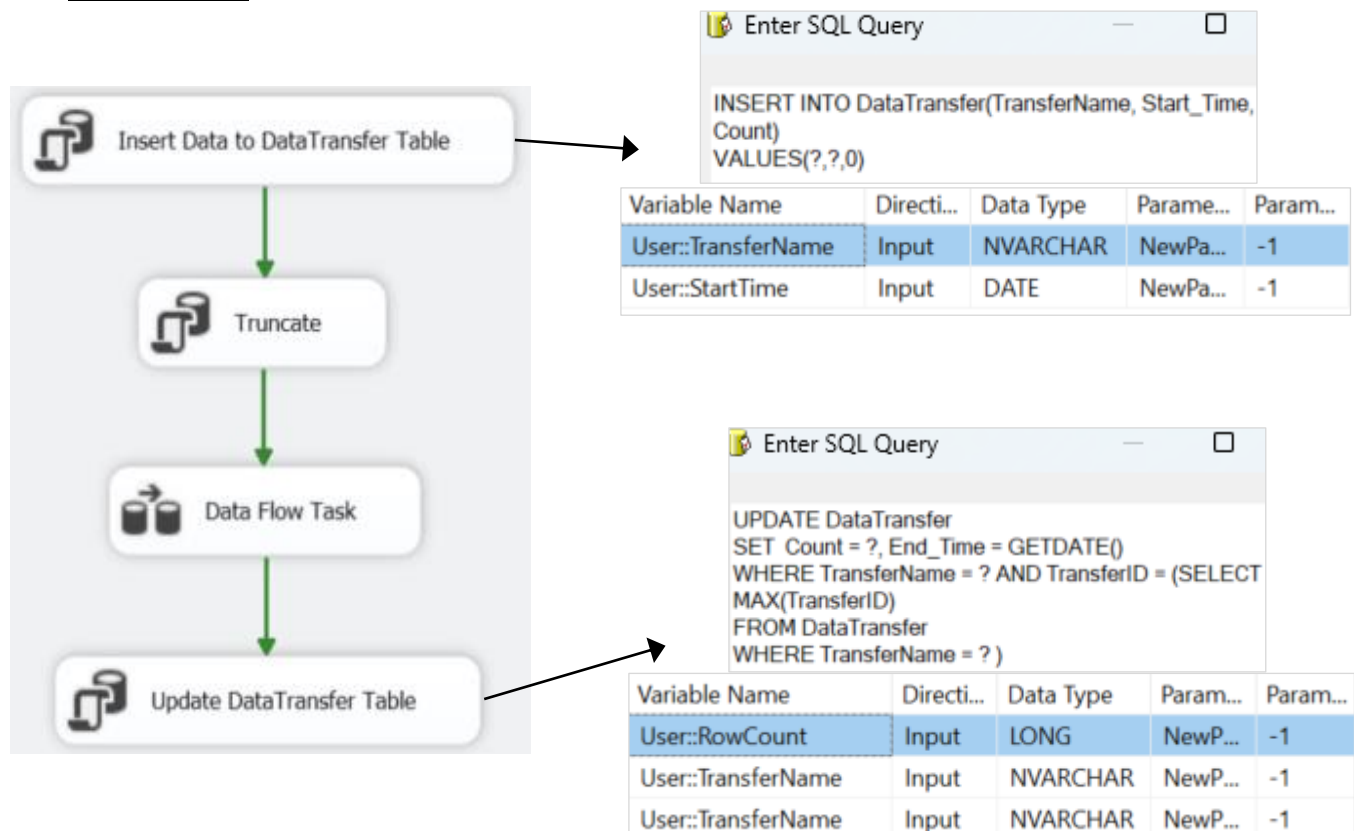
To monitor the ETL process, a DataTransfer table was created to document each data insert and data update. The table shows which transfer/package was executed, how many rows were inserted and updated, the time the execution started, and the time it ended. All packages include the tasks and transformations that document the data of the transfers and insert the data into the DataTransfer table (some packages count the rows using the Row Count transformation, and some use the Execute SQL Task) and the variables that we use.

Example from STG_Customers package:

In the Control Flow an **insert statement** is executed in the first task, The values in the user variables: TransferName and Start_Time are inserted into the matching columns of the DataTransfer table and inserted 0 to the Count column.

After the operations are done in the Data Flow, an **update statement** is executed in the last task. The Count column is updated to the value in the Count user variable and the End_Time column is updated to GETDATE().

In Control Flow:

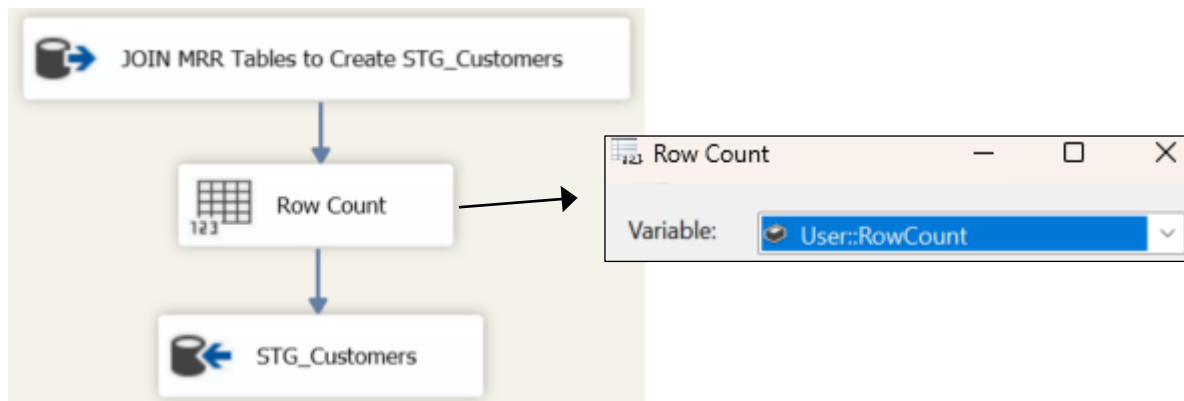


User Variables:

Name	Scope	Data type	Value	Expression
RowCount	STG_Customers	Int32	0	
StartTime	STG_Customers	DateTime	11/08/2024 19:59	GETDATE()
TransferName	STG_Customers	String	MRR to STG Customers	

In Data Flow:

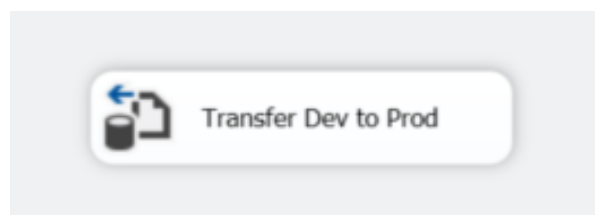
The user variable Count is updated using a Row Count transformation.



- **Transfer to Production:**

Copy all in the Development database to the Production database.

In Control Flow:



- **Automatic Processing:**

To facilitate the daily refresh and loading process, a deployment was executed from SSIS to SSMS. Subsequently, a job was created using SQL Agent jobs, to run on a daily schedule at a fixed time. This job comprises, thirteen steps, each representing a distinct SSIS package responsible for handling various phases of the project, the first step executed is the MRR DIM, and the last step executed is Fact Sales.

Error-handling rules have been defined to halt the process in case of an error, ensuring data integrity and reliability. Additionally, a success message is generated upon the successful completion of all thirteen steps.

Job step list:				
Step	Name	Type	On Success	On Failure
1	MRR DIM	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
2	MRR PART CDC	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
3	MRR Fact	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
4	STG Customers	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
5	STG Retailers	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
6	STG Products	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
7	STG Sales	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
8	Dim Employees	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
9	Dim Employees History	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
10	Dim Customers	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
11	Dim Retailers	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
12	Dim Products	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
13	Fact Sales	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure

The data is automatically refreshed **daily at 05:00:00**.

New Job Schedule

Name:

Refresh Western Digital DM

Jobs in Schedule

Schedule type:

Recurring

☒ Enabled

One-time occurrence

Date:

14/07/2024

Time:

12:18:16

Frequency

Occurs:

Daily

Recurs every:

1

day(s)

Daily frequency

☒ Occurs once at:

05:00:00

☐ Occurs every:

1

hour(s)

Starting at:

00:00:00

Ending at:

23:59:59

Duration

Start date:

14/07/2024

☐ End date:

14/07/2024

☒ No end date:

Summary

Description:

Occurs every day at 05:00:00. Schedule will be used starting on 14/07/2024.

OK

Cancel

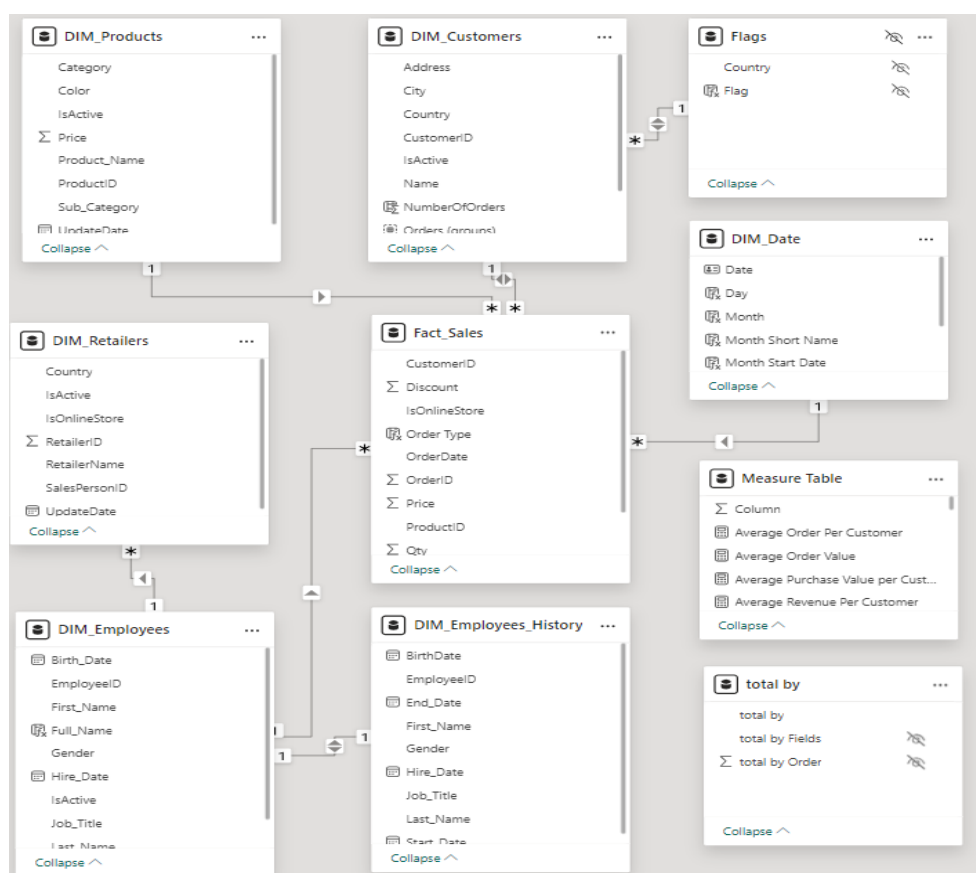
Help

5. Visualization in Power BI

In this chapter, I delve into the dynamic realm of Power BI, a pivotal component in my project's data analysis and visualization strategy. Using this tool, I have created a dashboard and two reports: customers and sales. Those reports will help the company's senior management to uncover trends, make informed decisions, and drive meaningful outcomes.

5.1. Model

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, the 4 Dimension tables, and the history table. To these tables, a 'DIM_Date' table was added, together with two more tables- 'Measure Table' (the measures that were created in DAX) and 'total by' (contains parameters for Orders/Revenue/Units).

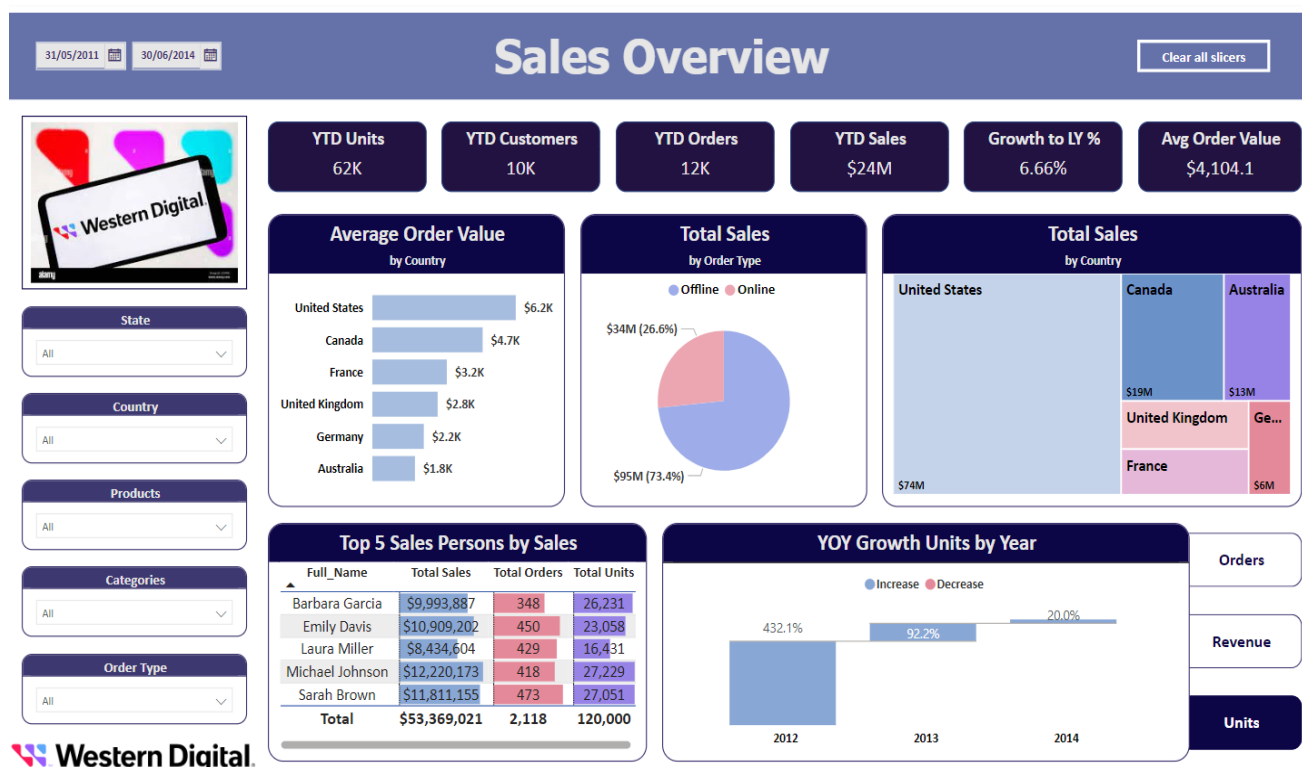


5.2. Dashboard & Reports

The project includes three reports: Executive Dashboard (Sales Overview), Sales Department Analysis, Customer Sales Analysis. The three reports include only data for years 2011-2014.

Sales Overview:

The dashboard was created to provide a wider perspective on the company's status, it includes the main KPIs and general graphs.



KPI Cards:

- YTD sales
- YTD units
- YTD orders
- YTD customers
- Growth to LY %
- Average Order Value

Graphs:

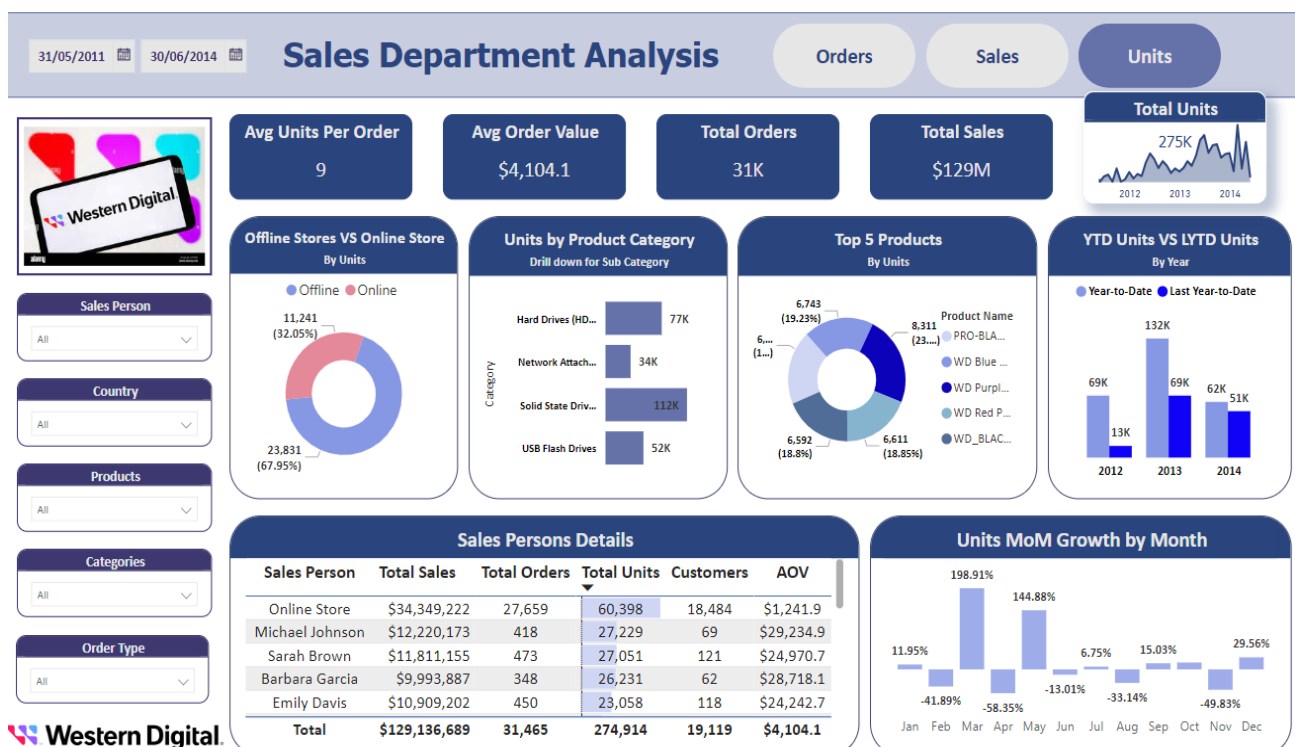
- Average Order Value by Country
- Total Sales by Order Type
- Total Sales by Country
- Top 5 Sales Persons by Sales
- YOY growth Sales/Orders/Units by year

Slicers:

- Date
- State
- Country
- Products
- Categories
- Order Type

Sales Department 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 top right, the user can control the data shown in the graphs and change it to orders data, and unit's data.



KPI Cards:

- Total sales
- Total orders
- Total units
- Average Order Value
- Average Units Per Order

Graphs:

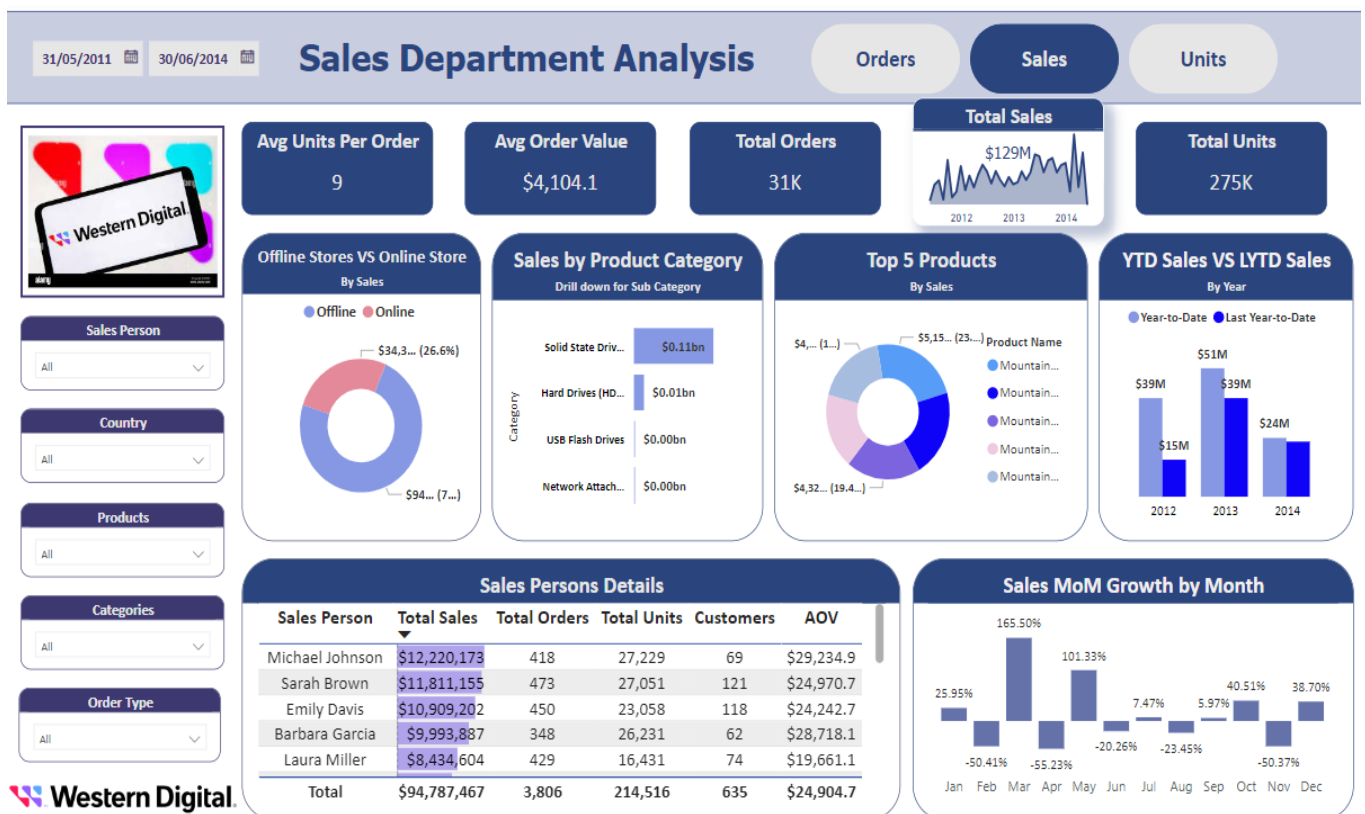
- Offline Stores VS Online Store by Sales/ Orders/ Units
- Sales/ Orders/ Units by Product Category and Sub Category

- Top 5 Products by Sales/ Orders/ Units
- YTD VS LYTD by Sales/ Orders/ Units
- Sales/ Orders/ Units MoM Growth by Month
- Sales Persons Details

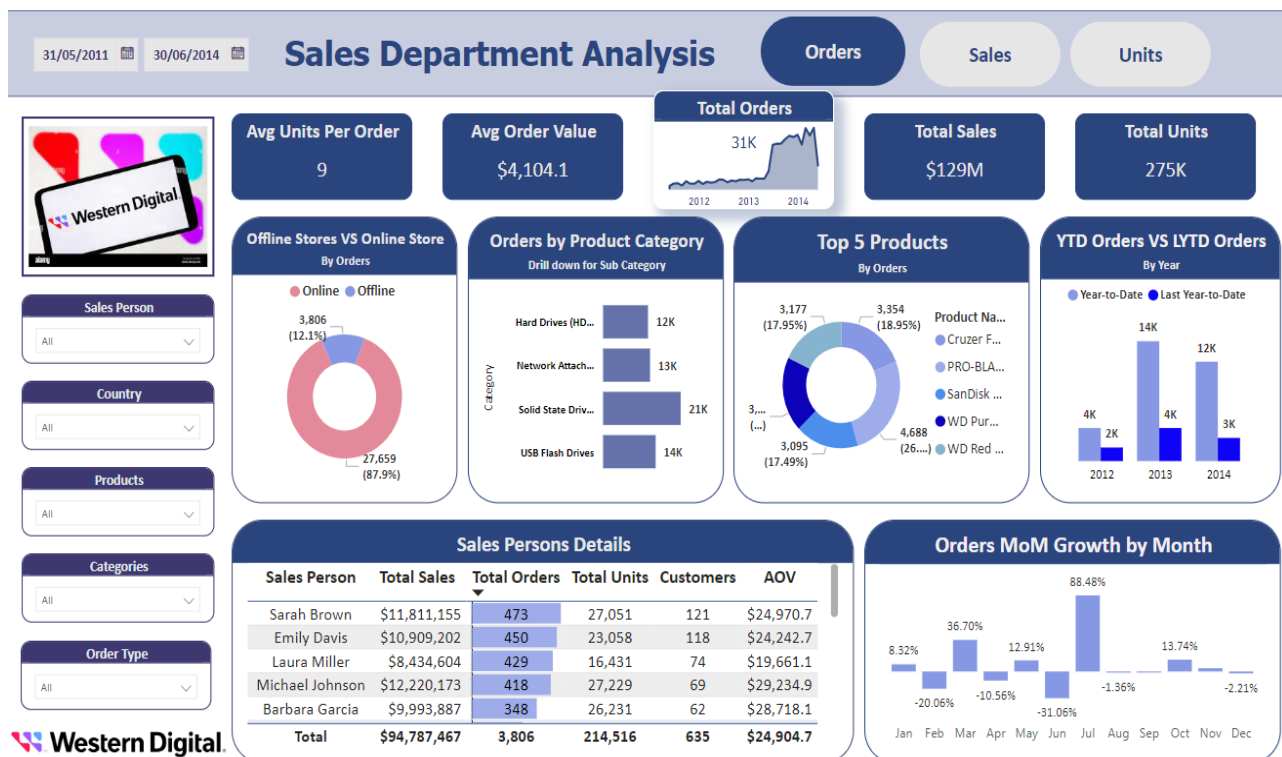
Slicers:

- Date
- Sales Person
- Country
- Products
- Categories
- Order Type

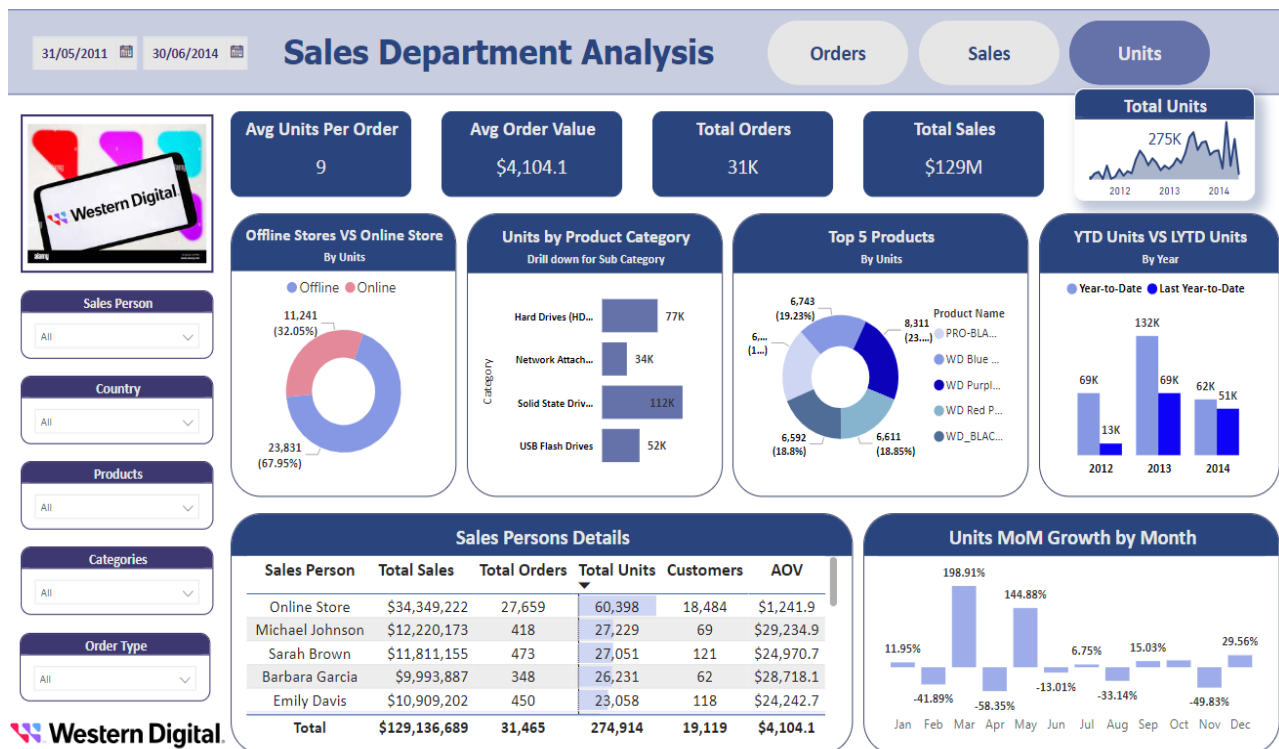
Sales State:



Orders State:

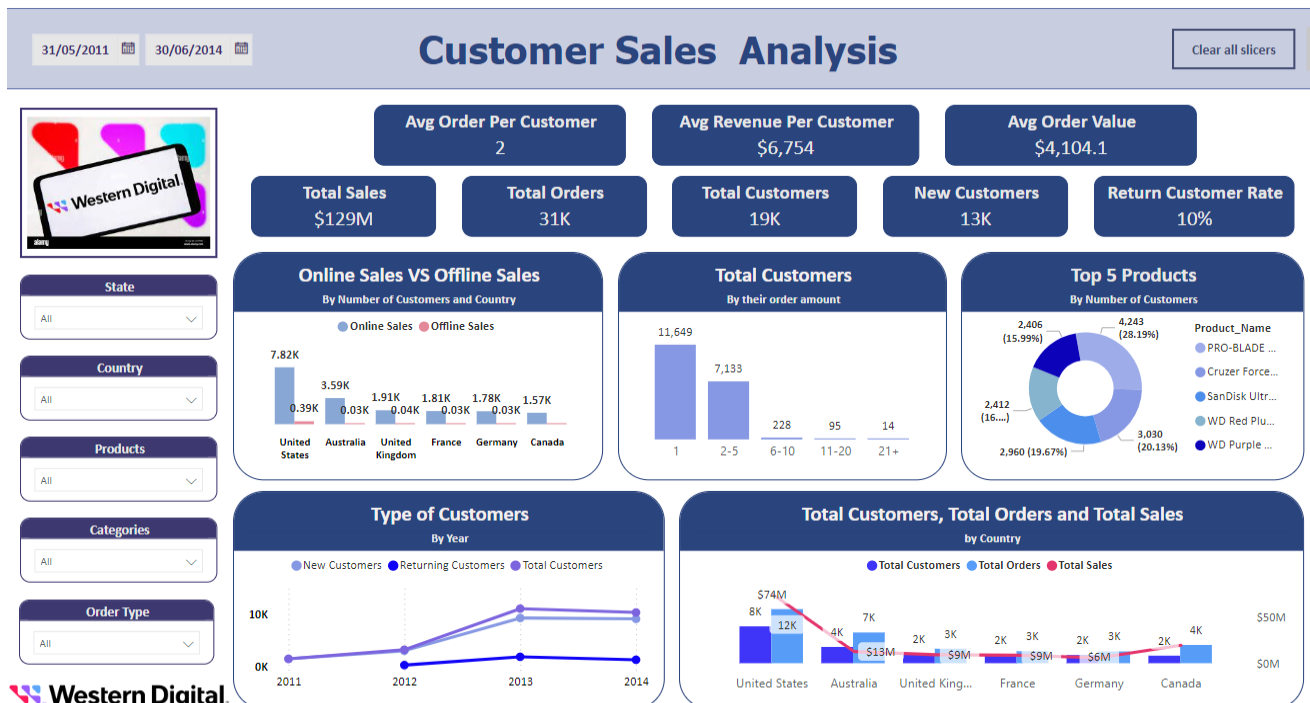


Units State:



Customers Sales Analysis:

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



KPI Cards:

- Total sales
- Total orders
- Total customers
- New customers
- Return customer rate
- Average Order Value
- Average revenue per customer
- Average order per customer

Graphs:

- Online Sales VS Offline Sales by number of customers and country
- Total customers by their order amount
- Top 5 Products by number of customers
- Type of Customers by year
- Total customers, total orders, and total sales by Country

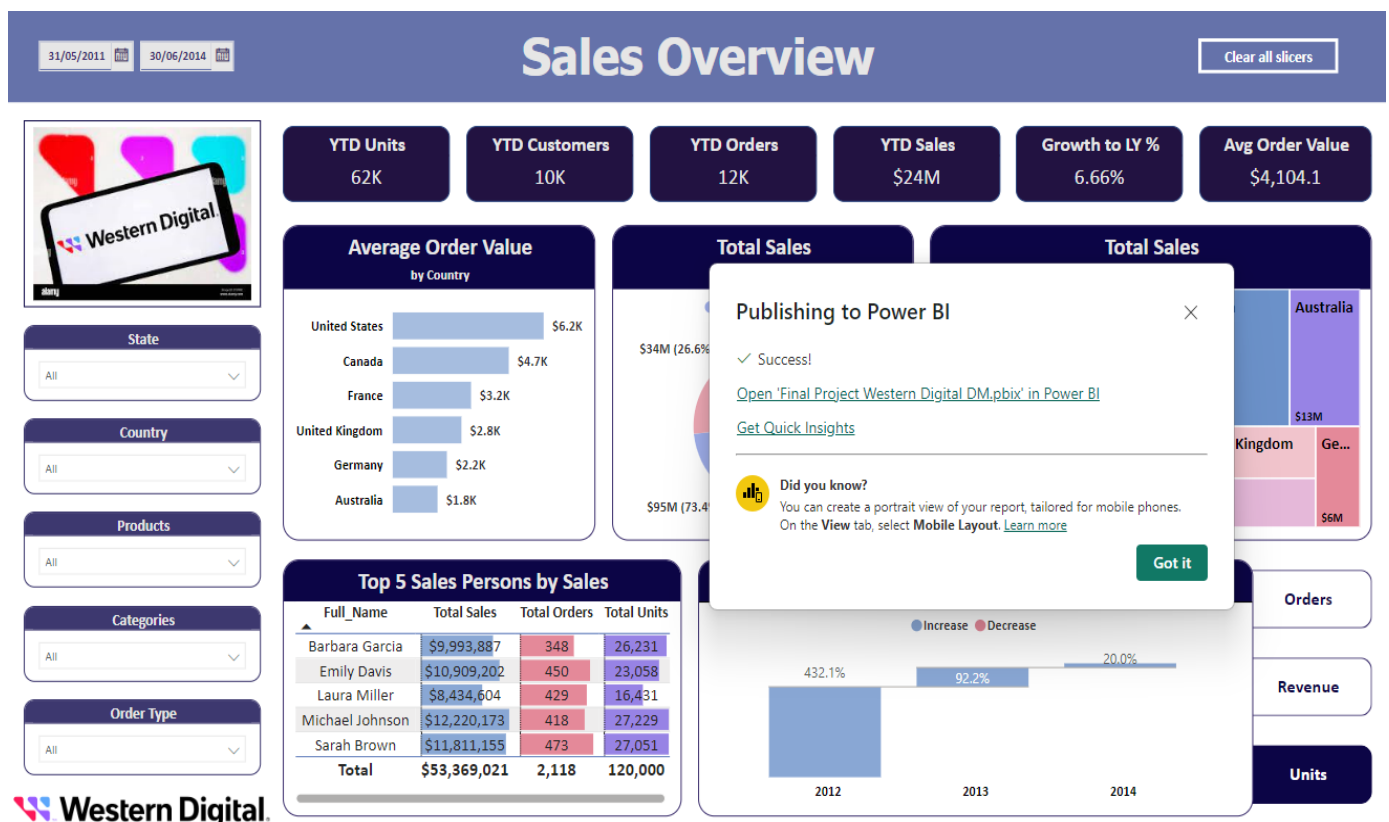
Slicers:

- Date
- State
- Country

- Products
- Categories
- Order Type

5.3. Published to Power BI Service

After creating the dashboard and the reports in Power BI Desktop, they were published to Power BI Service, and an app was created- [Western Digital- Final project.](#)



5.4. Data refresh processes

The data is refreshed daily at 5:00:00 (after the refresh of the data mart occurs), for this purpose, a personal gateway was created:

Gateway and cloud connections

To use a data gateway, make sure the computer is online and the data source is added in [Manage Connections and Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click apply.

Gateway connections

Use an On-premises or VNet data gateway

On

Gateway	Department	Contact information	Status	Actions
<div><div></div>Personal Gateway</div>			<div><div></div>Running on DESKTOP-0D1U6M4</div>	<div><div></div></div>

Refresh

Configure a refresh schedule

Define a data refresh schedule to import data from the data source into the semantic model. [Learn more](#)

On

Refresh frequency

Daily

Time zone

(UTC+02:00) Jerusalem

Time

5

00

AM

X