Western Digital Sales Data Mart

BI System Specifications Document

Date: 31/07/24

Version: 4.00

Written by: Romina Boltaks

Contents

1. General

1.1. Project Objective

This project aims to create a comprehensive BI solution for Western Digital's sales department to facilitate new financial insights and promote product sales. The project was developed based on the sales department's demands and KPIs to enable greater sales revenues, highlight the current financial status of the company, and understand its financial strengths and weaknesses.

Western Digital Corporation, also referred to as WD, is a prominent American data storage company. It is known for its development and manufacturing of data storage solutions, including hard disk drives (HDDs), solid-state drives (SSDs), and NAND flash memory. The company's products are widely used in personal computing, enterprise data centers, and various other applications requiring data storage and management. This project will focus on the physical and online stores of WD around the globe.

The information for the Data Mart will be derived from the PriorityERP database (WD operational database). The solution will concentrate on product sales, WD's customers, top products and salespersons, and the global distribution of product selling stores. Additionally, it will feature sales and customer analysis reports, along with a comprehensive executive dashboard. These reports will be provided to the sales department for in-depth investigation and to support strategic business decisions.

1.2. Project Contents

The Data Mart will be constructed through ETL processes, starting from the operational database and leading to the final Sales Data Mart (WDSalesDM).

The ERD model of WDSalesDM: Link

1.2.1. Tables in the Data Mart

The Data Mart will include 1 fact table, 4 dimension tables, and 2 history tables.

- **FactSales** – Includes all sales records and sales related details such as order's id, various dates (order date, shipping date, etc.), the store where the order occurred, quantity purchased, the total price and more.
- **DimProducts** - Data on all company's products and additional product attributes.
- **DimStores** – Data regarding the company's worldwide operating stores.
- **DimCustomers** – Data on the company's clients.
- **DimAgents** – Data on the company's employees.
- **DimProductsHistory** – Historic data records of the DimProducts table.
- **DimCustomersHistory** – Historic data records of the DimCustomers table.

The tables will be synchronized daily at 05:00:00 with the operational database using an automated process set in SQL Server Management Studio.

1.2.2. Reports Overview

- Sales Analysis Report:
  This report will include the following sales data: total and year-to-date revenue, ROS, year-to-date orders placed, and year-to-date units sold. These KPIs will be filtered by date, country, product category, and salesperson, helping the Sales department gain deeper insights into the committed sales. Analysing periodic sales will enable the Sales department to spot trends, and geographic data will help identify high-volume selling points and determine if there is a need to promote an opening of new stores in those areas. Data slicing by category will highlight the most popular products sold, while analysing data by store type will allow the Sales department to compare the performance of online stores versus physical stores. Lastly, displaying sales data with a focus on salesperson will incorporate the human factor into the sales analysis.

- Customer Trend Analysis Report:
  Customer analysis will focus on customers' preferences, such as whether they prefer shopping online or in-store, identifying areas with the highest average purchasing price, and determining which category is preferable by customers. The data will be analysed by date, country, agents, category, subcategory, and store type. This report aims to enhance the company's engagement with its customer behaviours.

- Executive Dashboard:
  The dashboard will summarize key data and elements from other reports, providing an overall view of the company's current sales situation. This centralized report will facilitate an overall assessment of the sales status by providing crucial metrics and by laying the foundation to derive insights.

2. Gantt Chart

Scheduled timeline for the project: Link

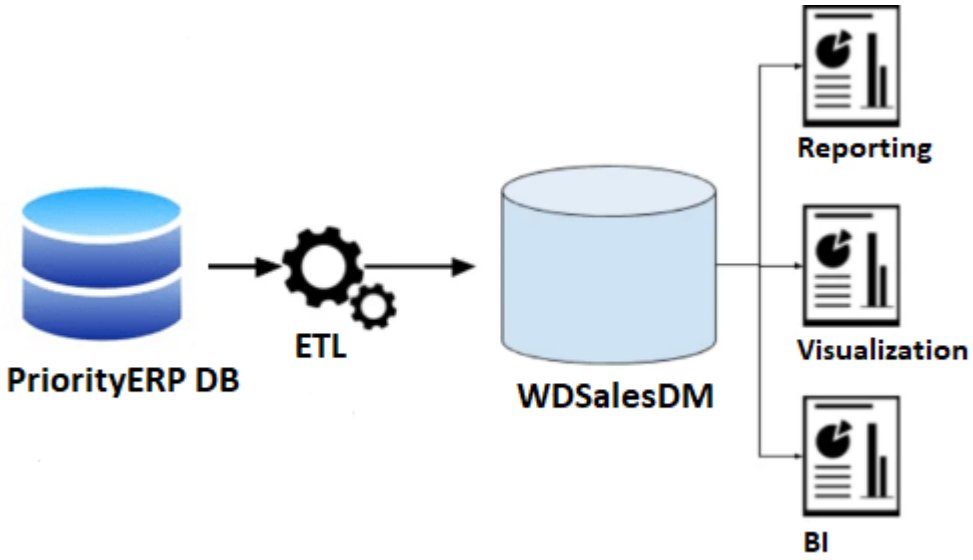
3. Technical Specifications

3.1. Prerequisites

- SQL Server: ERP system and its operational database (PriorityERP) – populated tables and files.
- SSIS: Using Visual Studio for ETL processes
- Data refresh processes: Setting JOBS in SSMS
- Power BI: Developing reports and a dashboard

3.2. Solution Architecture

3.2.1. High Level Design



The ETL process will be conducted using SSIS to create the Data Mart in SQL Server. Once the Data Mart is finalized, it can be used to create reports in Power BI.

3.2.2. Power BI Reports

3.2.2.1. Sales Analysis Report Measures and Visualizations

- YTD Units Sold
- YTD Purchases
- ROS
- YTD Total Revenue
- Total Revenue
- Total Units Sold by Store Type (Online/In-Store Purchasing)
- Revenue and MoM% Revenue Change (for years 2021, 2022, 2023, 2024)
- Total Revenue Distribution by Category
- Top 5 Products by Revenue (for years 2021, 2022, 2023, 2024)

3.2.2.2. Customer Trend Analysis Report Measures and Visualizations

- YTD Customers Volume
- Average Purchase Price
- Total Purchases
- Total Units Sold
- Average Purchasing Price - Ranked by Country

- Total Purchases by Product Category
- Month over Month Customers' Growth (for 2021, 2022, 2023, 2024)
- Customers' Volume by Store Type (In-store purchasing/Online Purchasing)
- Customers Volume Comparison (same month previous year)

3.2.2.3. Executive Dashboard Measures and Visualizations

- YTD Revenue
- YTD Purchases
- YTD Units Sold
- YTD Customers Volume
- ROS
- Total Revenue by Store Type
- Revenue and MoM% Revenue Change for year 2024
- Customers Volume by Country
- Top 5 Salespersons by YTD Revenue
- Total Revenue and YoY% Revenue Change

4. Functional Specifications

4.1. Consolidating final Source to Target document and ERD model

4.1.1. Source to Target

Source to target link

4.1.2. ERD model of the WDSalesDM

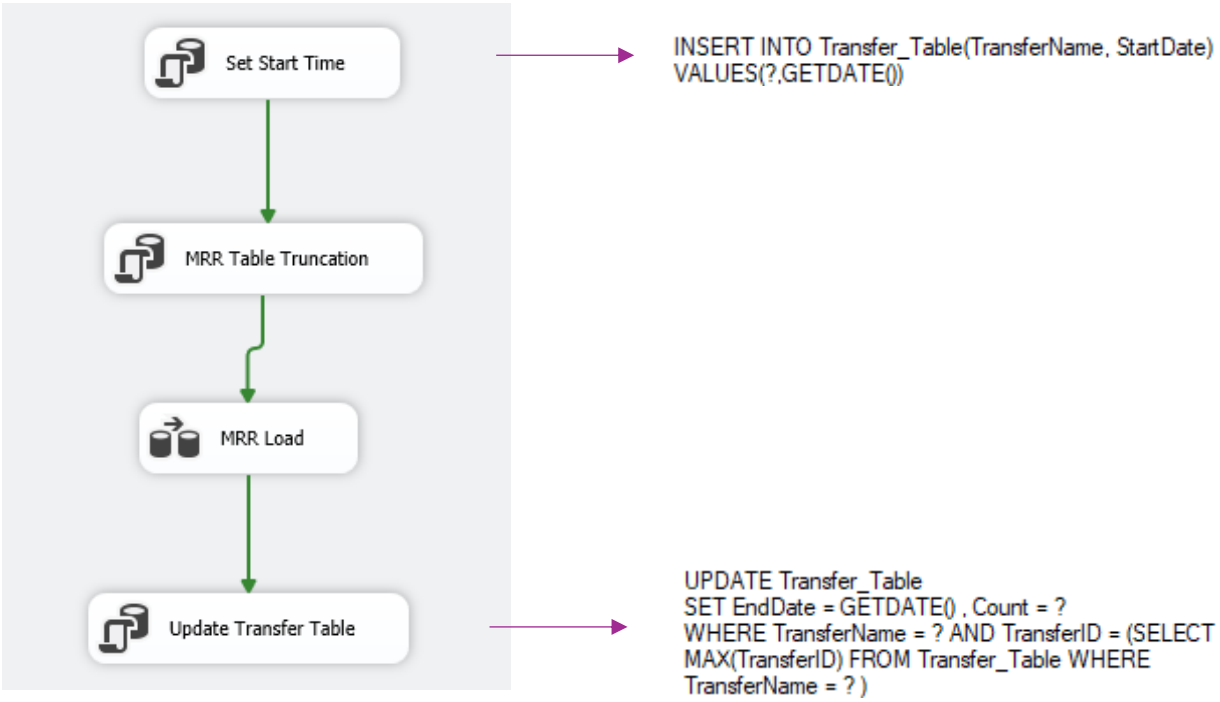
ERD link

4.2. ETL Processes

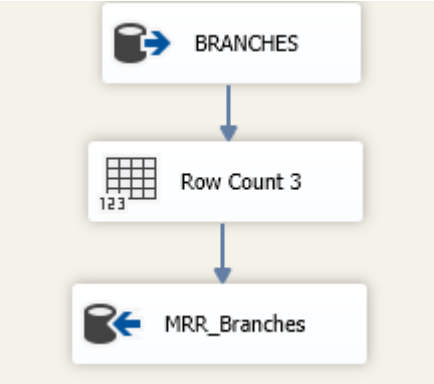
The ETL process was constructed in SSIS using 13 packages.
All the packages include load control process (counting the records transferred from source to target and updating a dedicated transfer table).

Transfer Table:

Example of the recurring load control process in all packages:



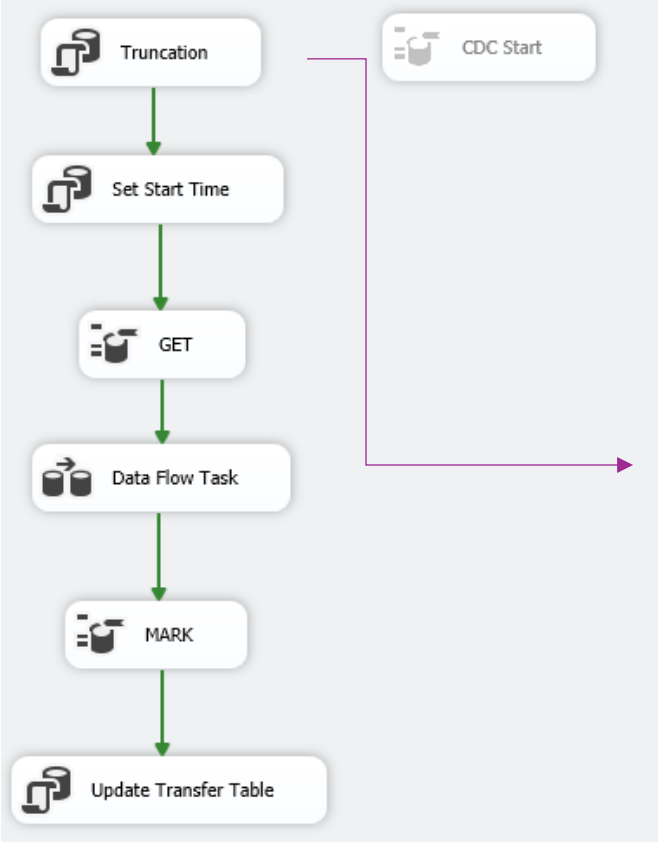
Example of using Row Count transformation in the Data Flow task:



RowCount variable is preset to receive the result of “Row Count” transformation at runtime and later is used as a parameter passed at runtime to the “Count” column in the Update query in “Update Transfer Table” component (Execute SQL Task). Another preset variable is TransferName that is initiated with the name of the package and is also passed at runtime to the “TransferName” column of the transfer table in “Set Start Time” component and “Update Transfer Table” component in the control flow which are both “Execute SQL Task”.

DimAgents Table

- MRR_Agents Package - Control Flow:



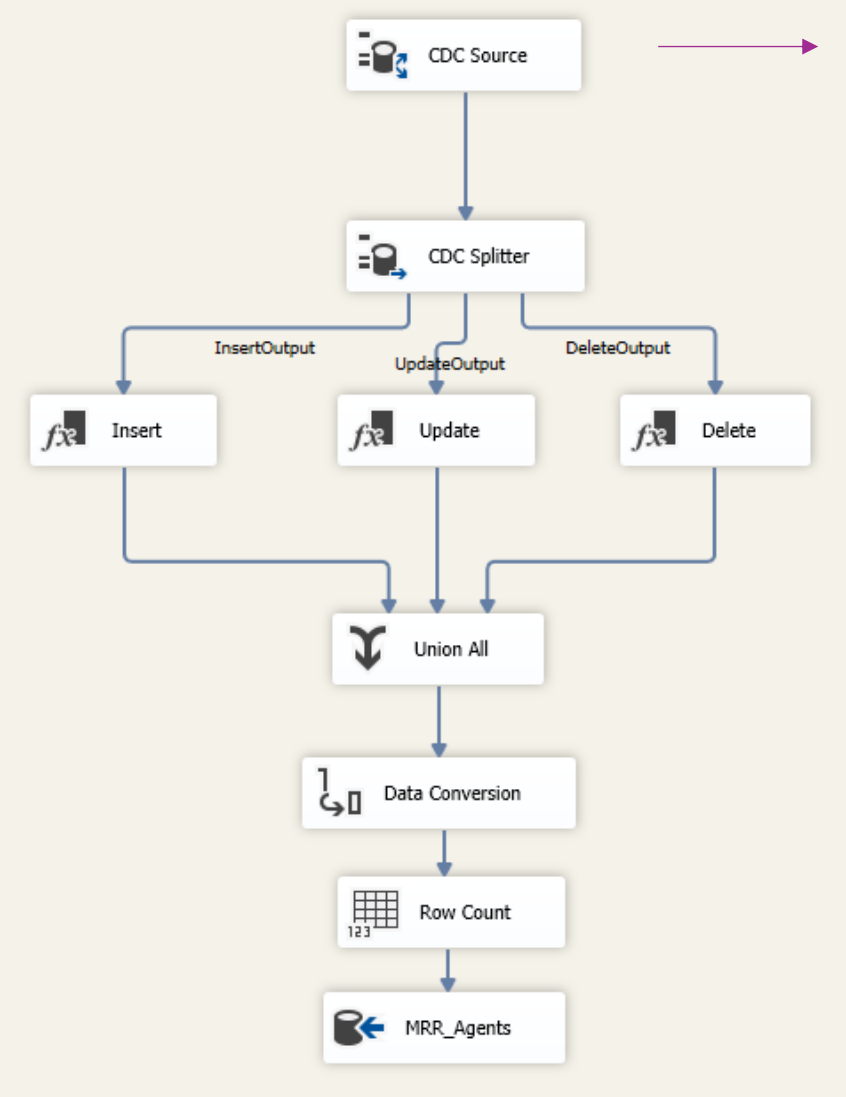
This package uses the CDC feature of SSMS.

The CDC was enabled on PriorityERP database and the Agent table in the PriorityERP database.

The “Truncation” component makes sure the MRR_Agents table is empty prior to the load of changed records

```
TRUNCATE TABLE MRR_Agents
```

- MRR_Agents Package – Data Flow:

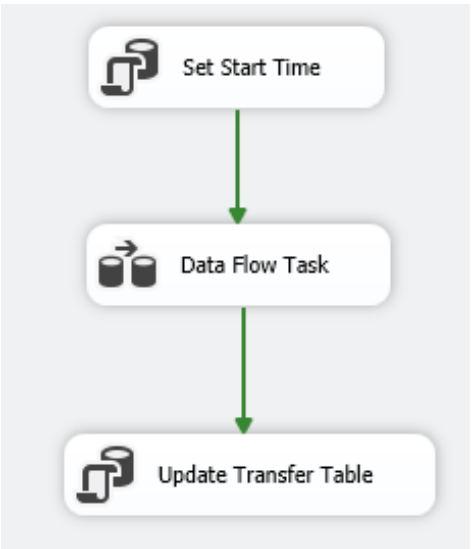


Connection Manager: PriorityERP.
CDC enabled table: Agent.

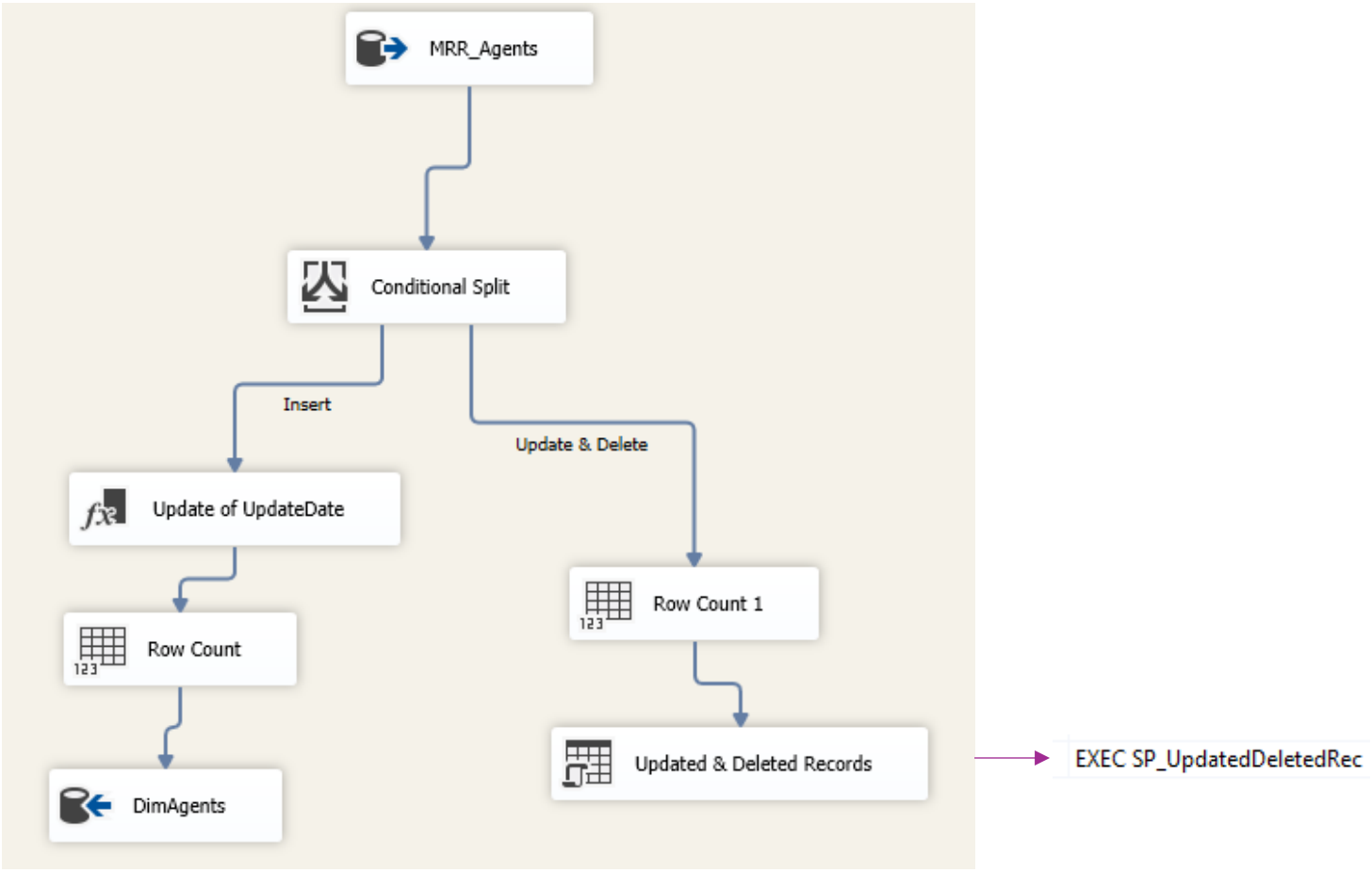
The CDC Splitter transformation divides the data into three different directions (inserted records, updated records, deleted records) based on the value in the "Operation" column of the "Agent_CT" table, which records changes made to the Agent table in the operational database.

The derived column "Status" is added in a later stage of the data flow to hold different values for each operation state (Insert – 1, Update – 2, Delete – 3).

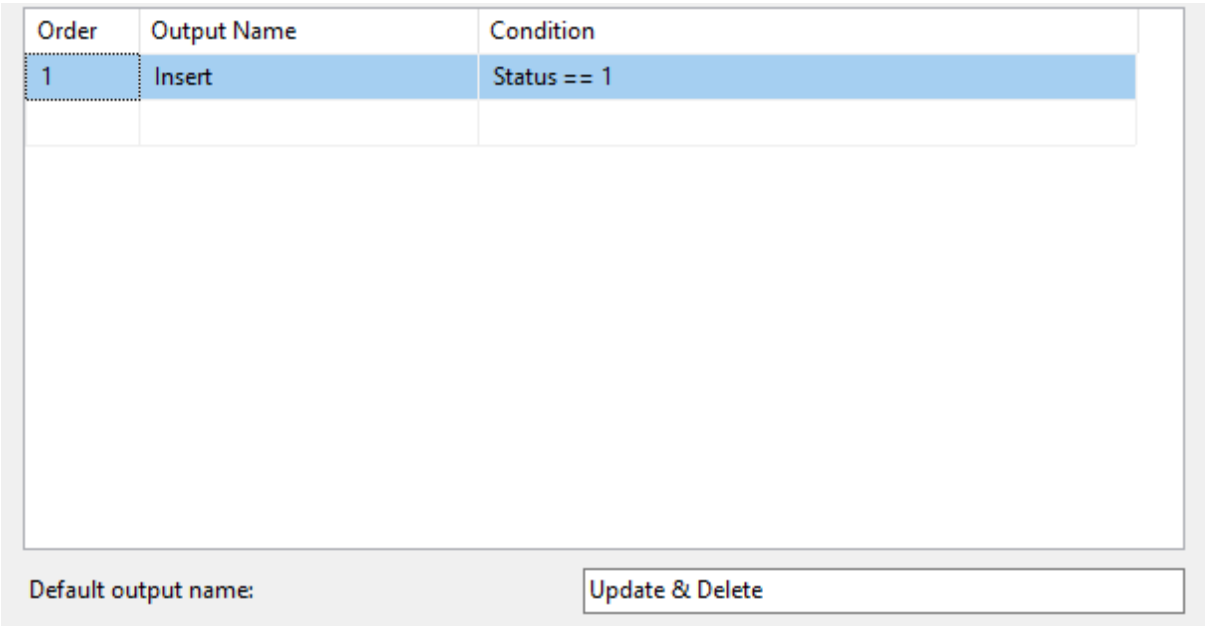
- DWH_Agents Package – Control Flow:
By using the CDC feature an incremental load of data transfers to DimAgents table.



- DWH_Agents Package – Data Flow:



Conditional Split Transformation:



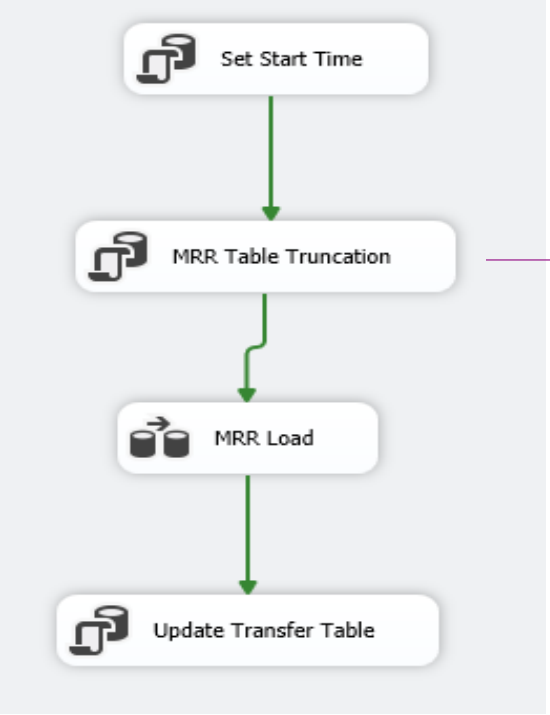
Based on the status value, the data is directed to the appropriate destination (using conditional split). Inserted records (status value 1) are directed to the "DimAgents" table.

Updated and deleted records are directed to the OLE DB Command transformation, which executes a stored procedure that handles these records. For deleted records, the "IsActive" column is updated to zero, and the "UpdateDate" column is set to the date at runtime. For updated records, all fields are updated, and the "UpdateDate" column is also set to the date at runtime.

```
CREATE PROCEDURE [dbo].[SP_UpdatedDeletedRec]
AS
MERGE [dbo].[DimAgents] AS Target
USING [dbo].[MRR_Agents] AS Source
ON Target.AgentID = Source.AgentID

WHEN MATCHED AND Source.Status = 2 THEN UPDATE
SET
Target.FullName = Source.FullName,
Target.HireDate = Source.HireDate,
Target.Role = Source.Role,
Target.Seniority = Source.Seniority,
Target.Gender = Source.Gender,
Target.Phone = Source.Phone,
Target.UpdateDate = GETDATE();

UPDATE DimAgents
SET IsActive = 0, UpdateDate = GETDATE()
FROM DimAgents DA
JOIN MRR_Agents MA ON DA.AgentID = MA.AgentID
WHERE MA.Status = 3
```

DimCustomers, DimProducts, and DimStores tables

- MRR_Tables Package – Conrtol Flow:



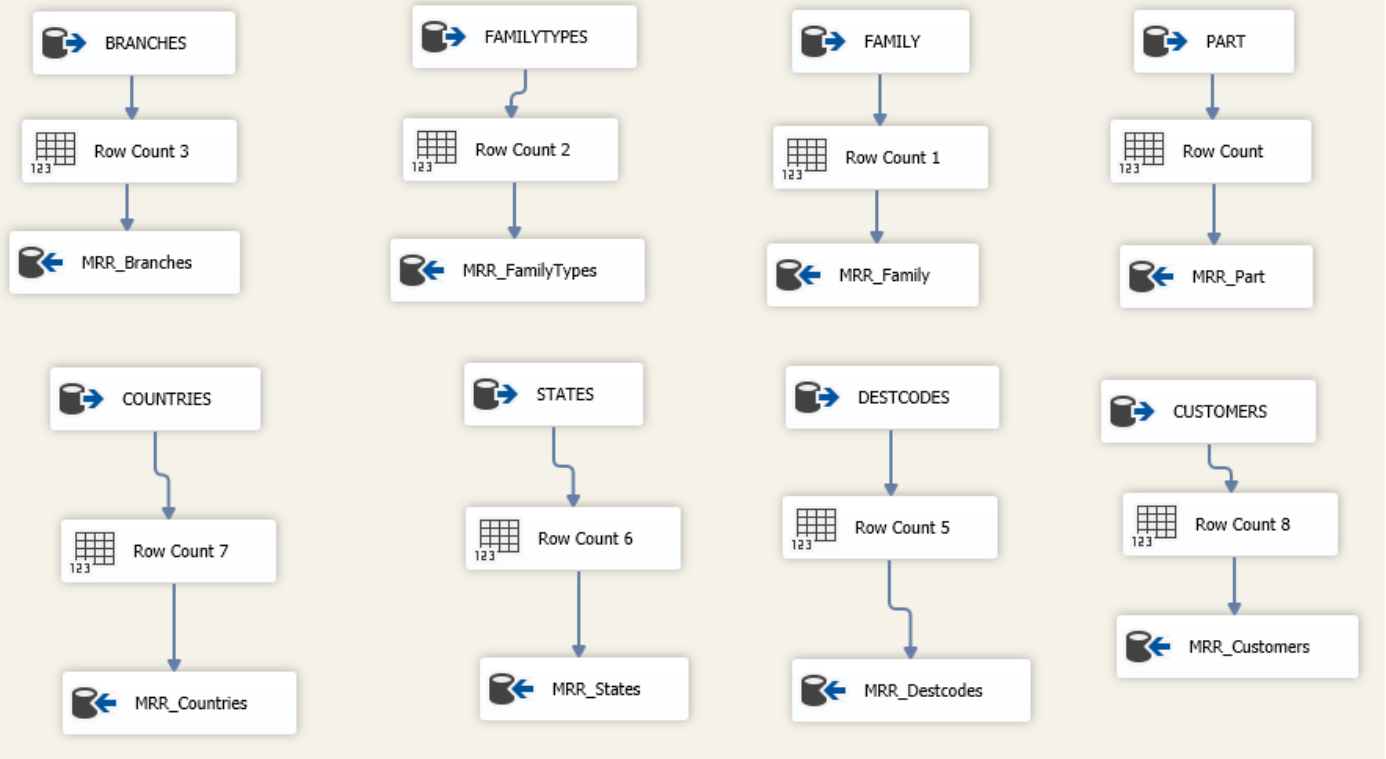
This flow is the mirror stage for DimCustomers, DimProducts, and DimStores tables. All mirror tables (except MRR_Invoices and MRR_InvoiceItems) are truncated using a stored procedure.

EXEC SP_TRUNCATE_MRR

The stored procedure that executes Truncate Table statements for all relevant mirror tables:

```
CREATE PROCEDURE [dbo].[SP_TRUNCATE_MRR]
AS
TRUNCATE TABLE MRR_FamilyTypes;
TRUNCATE TABLE MRR_Family;
TRUNCATE TABLE MRR_Branches;
TRUNCATE TABLE MRR_Part;
TRUNCATE TABLE MRR_Countries;
TRUNCATE TABLE MRR_States;
TRUNCATE TABLE MRR_Destcodes;
TRUNCATE TABLE MRR_Customers;
```

- MRR_Tables Package - Data Flow:

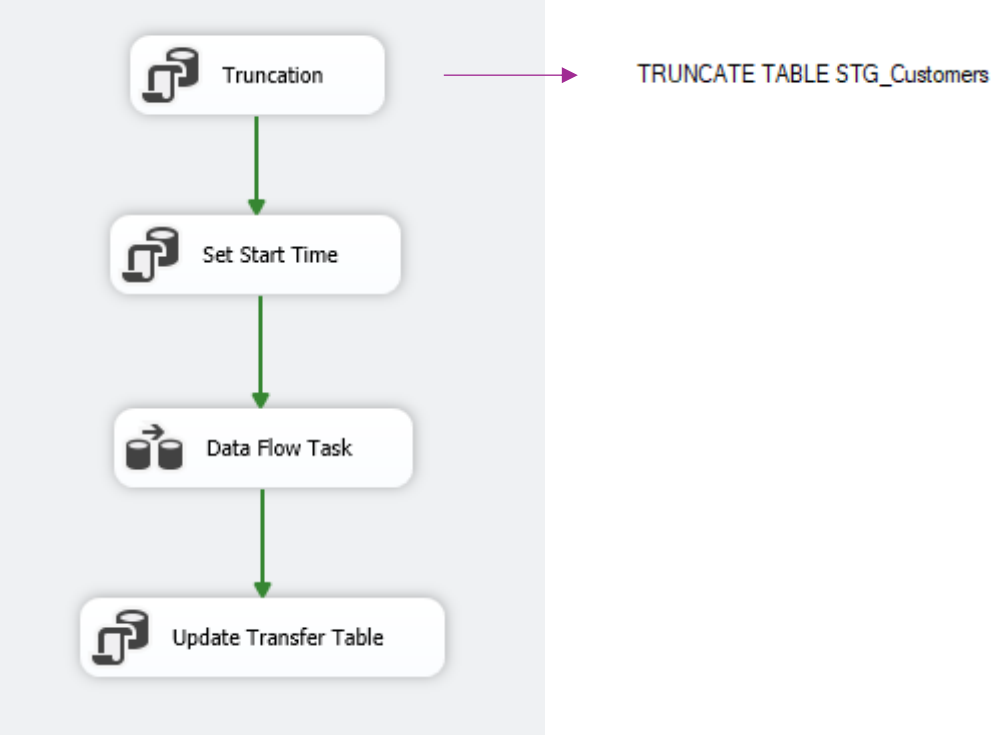


All Row Count transformations values are summed up in a new variable (Total Count) which is passed to the transfer table in the control flow.

DimCustomers Table

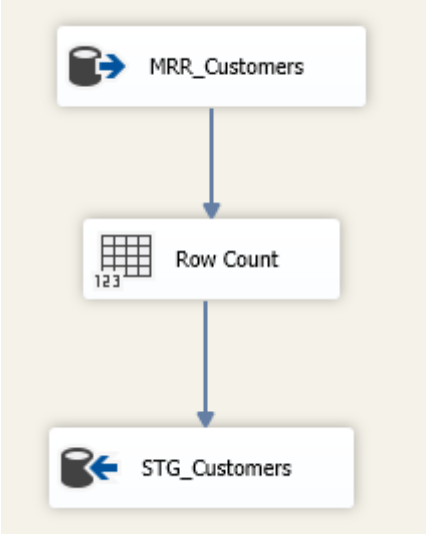
STG_Customers table is truncated, and the mirror tables are joined and loaded using a data flow task.

- STG_Customers Package – Control Flow



- STG_Customers Package – Data Flow

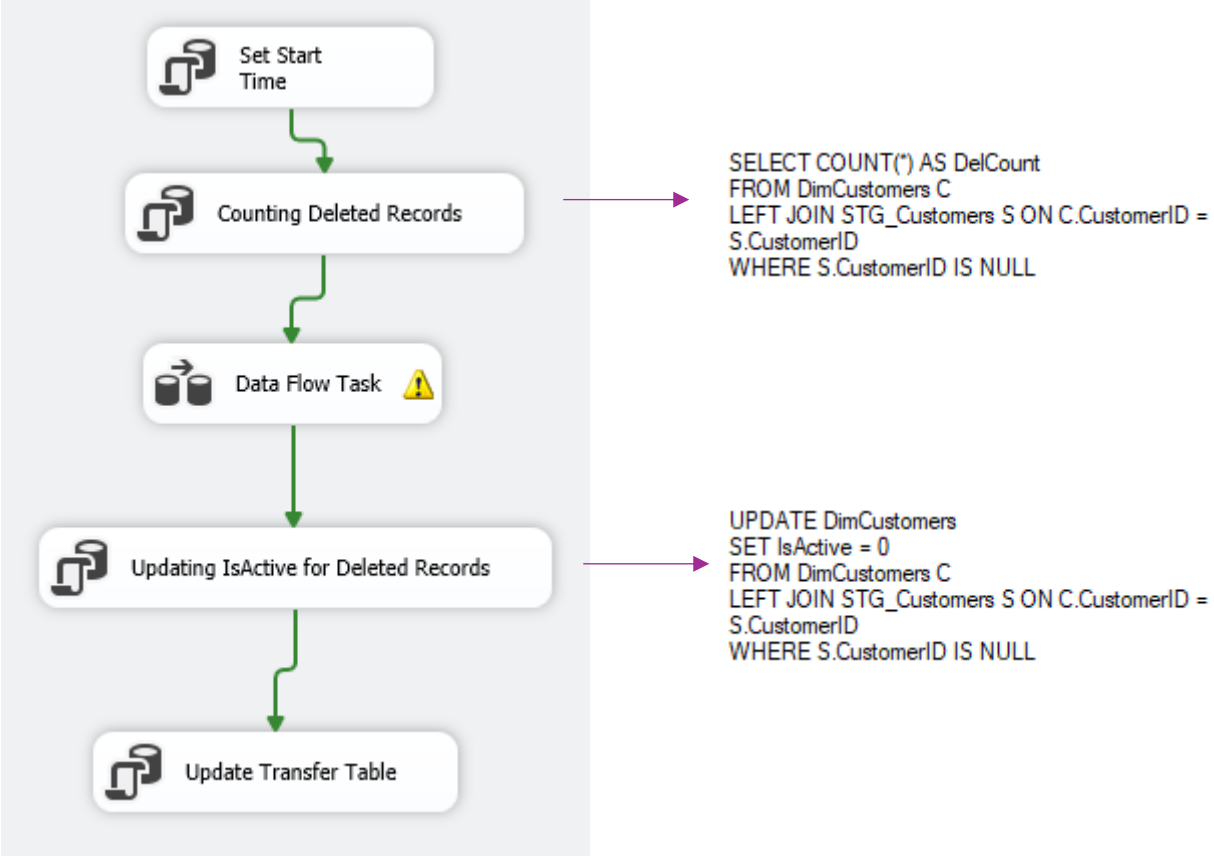
MRR_Customers component uses SQL query to join MRR_Invoices, MRR_Customers, MRR_Countries, MRR_States, and MRR_Destcode tables. The joined data is loaded to STG_Customers.



From this point forward, every warning sign is due to the changed field length from source to destination. In the destination, some fields were truncated to save space, but this did not compromise data integrity.

- DWH_Customers Package – Control Flow:

  Data was incrementally loaded and updated in DimCustomers.



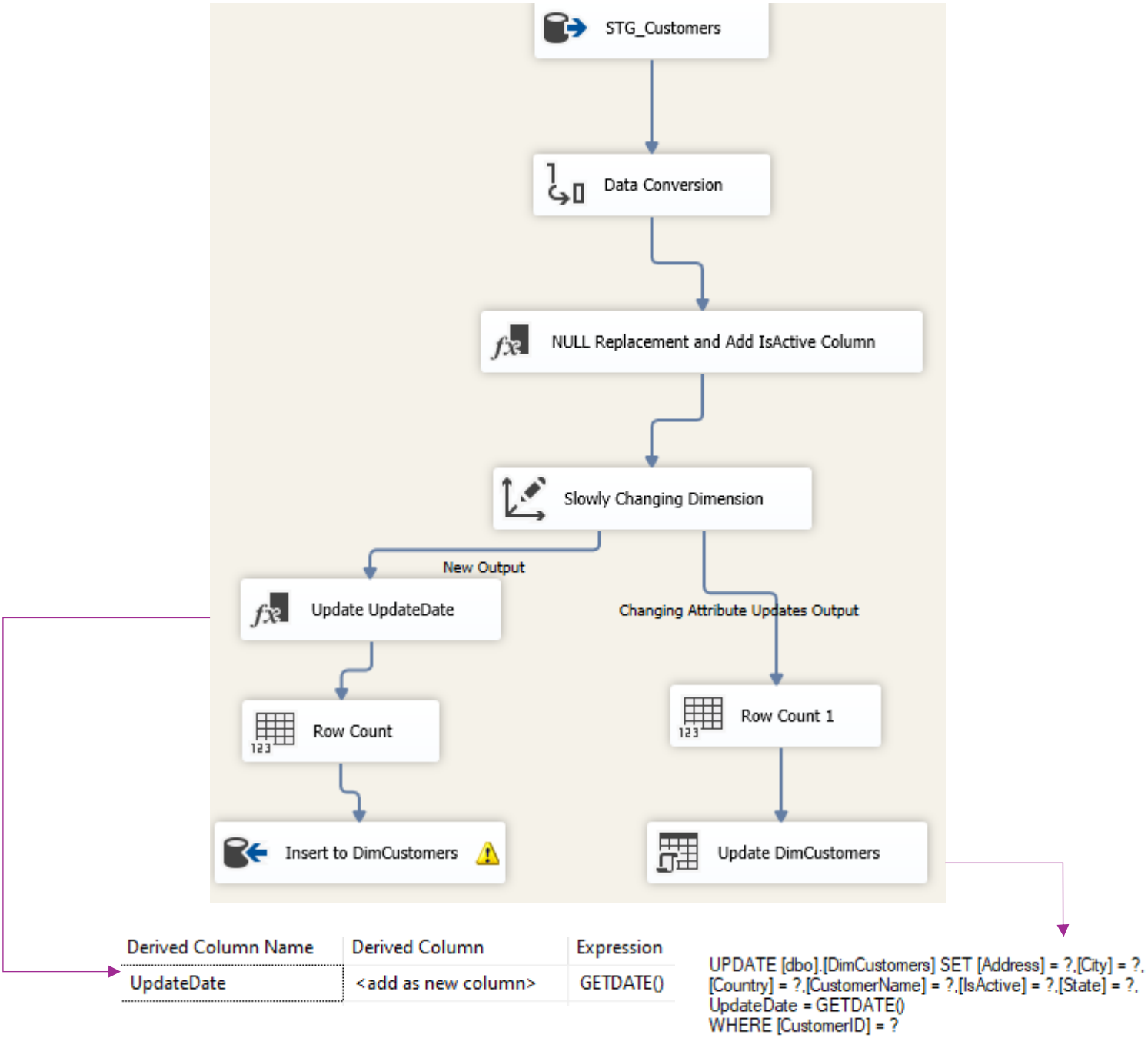
Deleted records were counted and added to the total row count as part of load control process.

The Data Flow Task does not handle deleted records as part of its flow; therefore, the deleted records are processed in the Control Flow. The deleted records in DimCustomers are updated to value 0 in the IsActive field to indicate record's deletion.

- DWH_Customers Package – Data Flow:

  In the Data Flow, incremental load to DimCustomers is done using the Slowly Changing Dimension transformation (change type: Changing Attribute). Later in the flow, there is an addition of an UpdateDate column, which updates to the date and time at runtime to indicate when the new records were inserted to DimCustomers. The "Update DimCustomers" component (OLE DB Command)

updates all fields in the updated records including the update of the date and time at runtime.
Both values of Row Count transformations are summed up and passed to a total row count variable which is passed in the control flow as a parameter to the transfer table.



Derived Column – "NULL Replacement and Add IsActive Column"

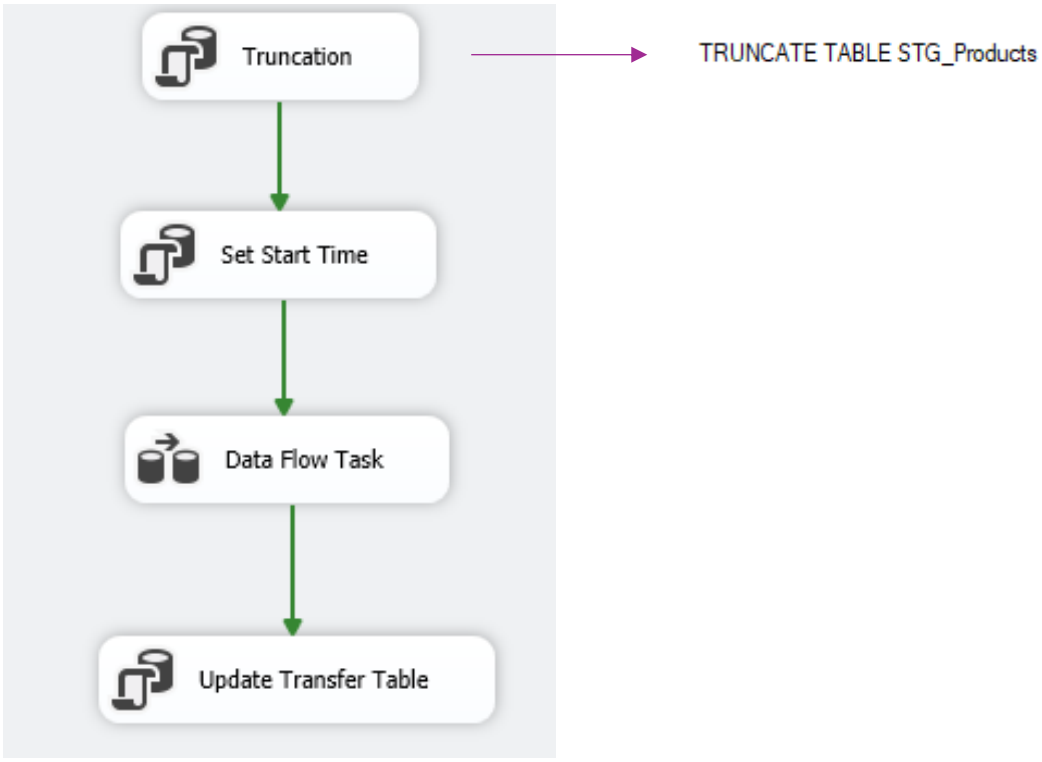
The derived column serves two purposes: replacing NULL values in all fields with "NA" and adding an "IsActive" column with a default value of 1.

Derived Column Name	Derived Column	Expression
IsActive	<add as new column>	(DT_BOOL)1
CustomerName	Replace 'CustomerNa...	REPLACENULL(CustomerName,"NA")
State	Replace 'State'	REPLACENULL(State,"NA")
Address	Replace 'Address'	REPLACENULL(Address,"NA")
Copy of Country	Replace 'Copy of Coun...	REPLACENULL([Copy of Country],"NA")
Copy of City	Replace 'Copy of City'	REPLACENULL([Copy of City],"NA")

DimProducts Table

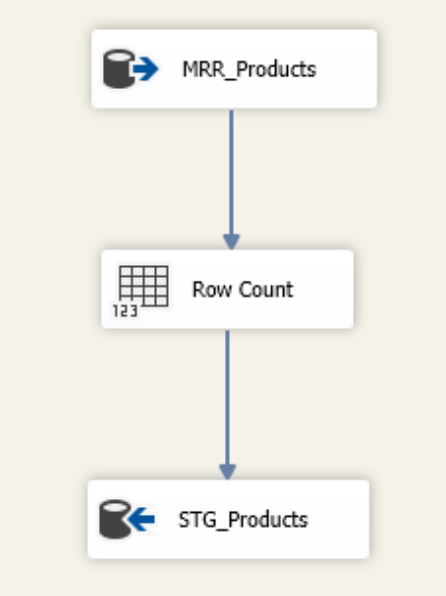
- STG_Products Package – Control Flow

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



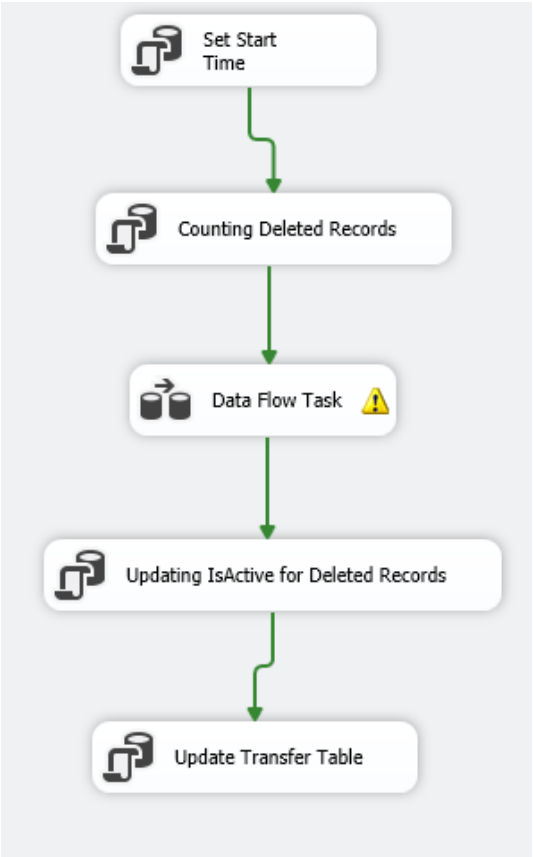
- STG_Products Package – Data Flow

  MRR_Products component uses SQL query to join MRR_PART, MRR_FamilyTypes, and MRR_Family tables. The joined date is loaded to STG_Products table.



- DWH_Products Package – Control Flow

Data is incrementally loaded and updated in DimProducts. Deleted records are updated in DimProducts using an Execute SQL task.




```
SELECT COUNT(*) AS DelCount
FROM DimProducts P
LEFT JOIN STG_Products S ON P.ProductID =
S.ProductID
WHERE S.ProductID IS NULL
```

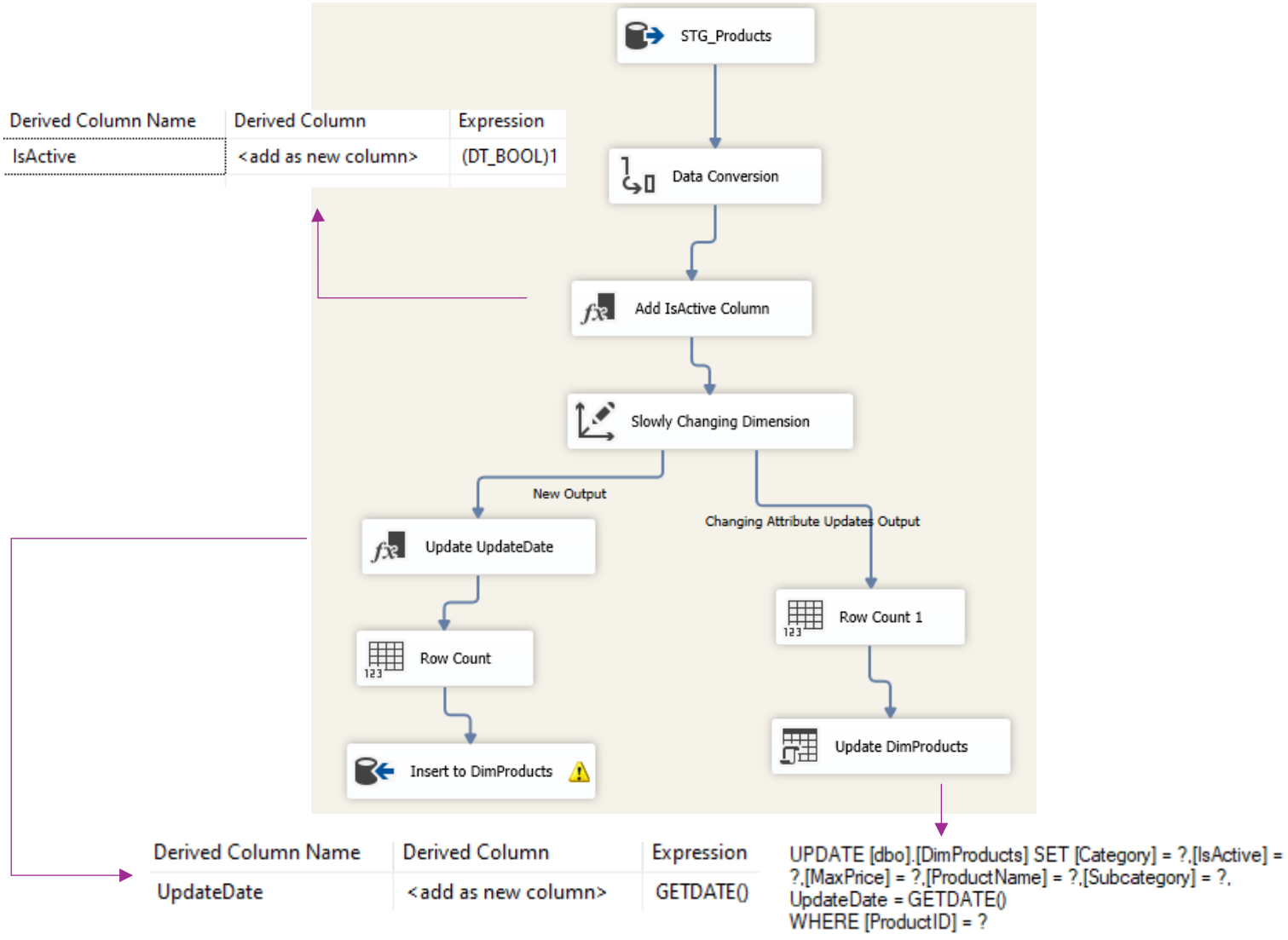
```
UPDATE DimProducts
SET IsActive = 0
FROM DimProducts P
LEFT JOIN STG_Products S ON P.ProductID =
S.ProductID
WHERE S.ProductID IS NULL
```

Deleted records were counted and added to the total row count as part of the load control process.

The Data Flow Task does not handle deleted records as part of its flow; therefore, the deleted records are processed in the Control Flow. The deleted records in DimProducts are updated to value 0 in the IsActive field to indicate record's deletion.

- DWH_Products Package – Data Flow

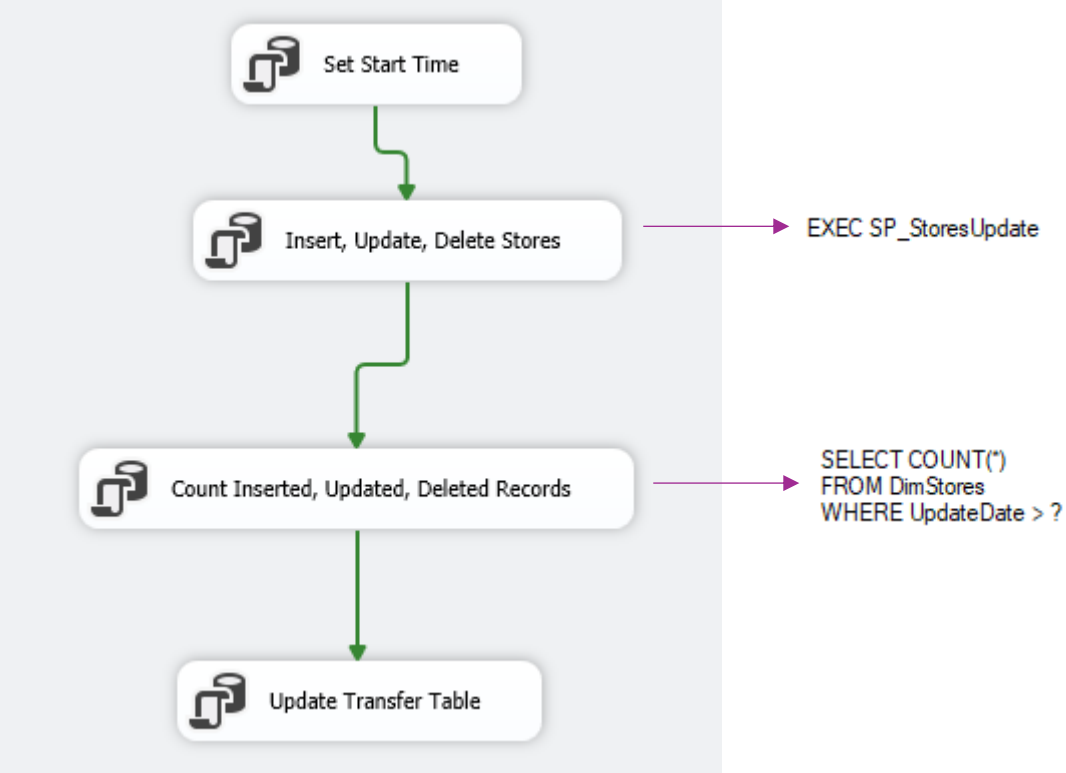
In the Data Flow, incremental load to DimProducts is done using the Slowly Changing Dimension transformation (change type: Changing Attribute). Down the flow, there is an addition of an UpdateDate column, which updates to the date and time at runtime to indicate when the new records were inserted to DimProducts or when the records were updated ("Update DimProducts" component updates all fields including the UpdateDate column). Both values of Row Count transformations are summed up and passed to a total row count variable which is passed in the control flow as a parameter to the transfer table.



DimStores table

The incremental load to DimStores is performed using a Merge query within a SP in Execute SQL Task (Insert, Update, Delete Stores).

- DWH_Stores Package – Control Flow



The parameter passed to the query above is StartDate that initialized with GETDATE() function in the beginning of the control flow.

The purpose of the counting query is to count all the inserted, updated, and deleted rows and to record it in the Transfer Table. The query counts rows that have a later update date than the process start date. This allows us to identify records that their UpdateDate column was updated to the current date and time at runtime.

"Insert, Update, Delete Stores" Execute SQL Task

"IsOnline" is a column with a Boolean logic in DimStores. A prior data exploration revealed that any agent with a "77777" value is connected to an online store.

"IsActive" is another column with a Boolean logic in DimStores. All records that exist in DimStores in the beginning of the process hold the default value of 1 in their "IsActive" column. This flag column, that indicates whether the store still works with WD or not, might change its value later by the SP if a certain logic is applied.

Additionally, “StoreCountry”, “City”, and “BranchName” columns are replaced with NA values in case of NULLs.

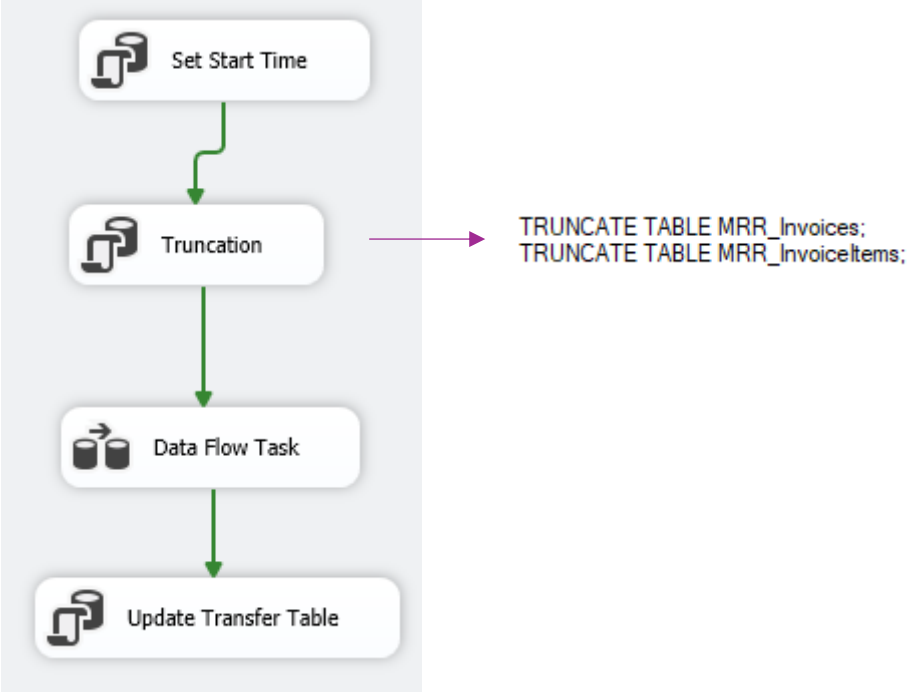
“Insert, Update, Delete Stores” SQL Task holds the stored procedure SP_StoresUpdate that handles all new records, updated or deleted records.

```
ALTER PROCEDURE [dbo].[SP_StoresUpdate]
AS
BEGIN
MERGE [dbo].[DimStores] Target
USING [dbo].[MRR_Branches] Source
ON Target.StoreID = Source.BRANCH

WHEN MATCHED AND (ISNULL(Target.StoreName, '') <> ISNULL(Source.BRANCHNAME,'') OR
ISNULL(Target.Country, '') <> ISNULL(Source.StoreCountry,'') OR ISNULL(Target.City,'')
<> ISNULL(Source.CITY,'')) THEN UPDATE
SET Target.StoreName = ISNULL(Source.BRANCHNAME,'NA'),
Target.Country = ISNULL(Source.StoreCountry,'NA'),
Target.City = ISNULL(Source.CITY,'NA'),
Target.UpdateDate = GETDATE()

WHEN NOT MATCHED BY Target THEN
INSERT (StoreID, StoreName, Country, City, IsOnline, UpdateDate)
VALUES (ISNULL(Source.BRANCH, 'NA'), ISNULL(Source.BRANCHNAME,'NA'),
ISNULL(Source.StoreCountry,'NA'), ISNULL(Source.CITY,'NA'),IIF(Source.AGENT = 77777,
1, 0), GETDATE())

WHEN NOT MATCHED BY Source AND Target.IsActive = 1 THEN UPDATE SET IsActive = 0,
UpdateDate = GETDATE();
END
```

FactSales table

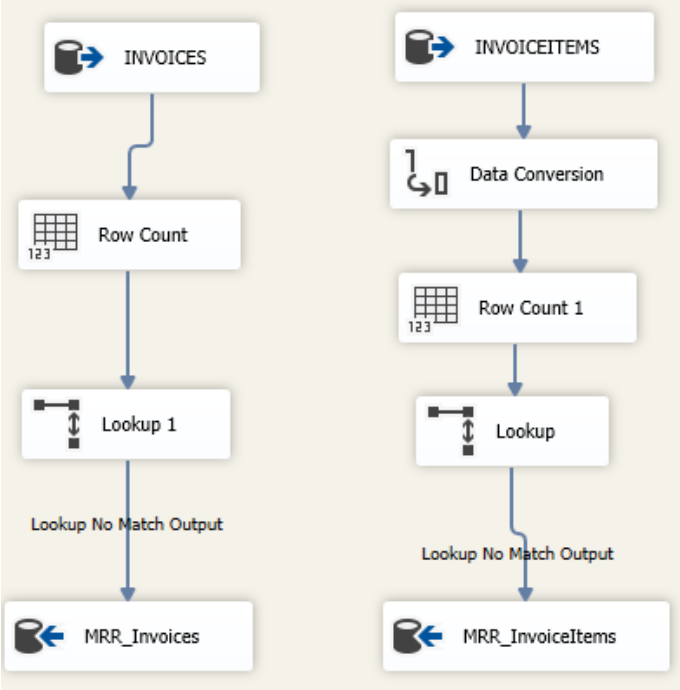
- MRR_Sales Package – Control Flow

  Mirror tables are truncated and data is loaded from PriorityERP database (INVOICES and INVOICEITMES) to MRR_Invoices and MRR_InvoiceItems tables in the WDSalesDM.



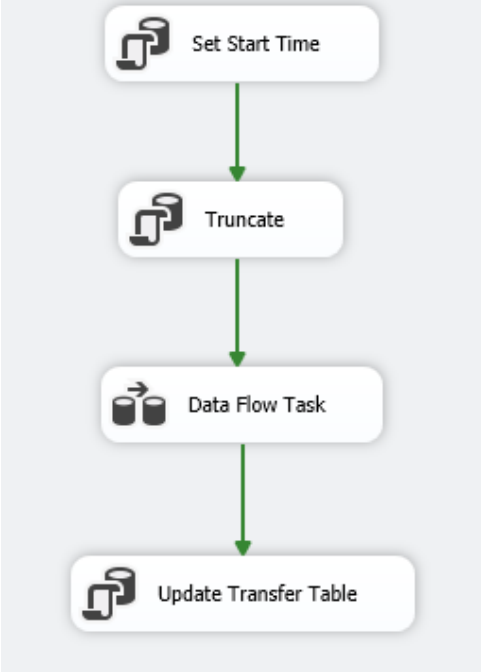
- MRR_Sales Package – Data Flow

  In the data flow, data is incrementally loaded using lookup transformations, which means that only new transactions that cannot be found in the FactSales table are loaded.



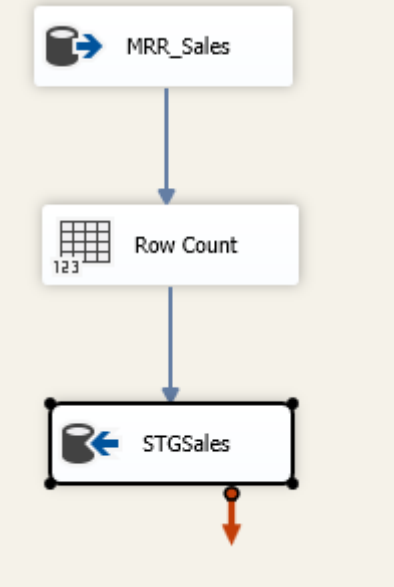
- STG_Sales Package – Control Flow

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



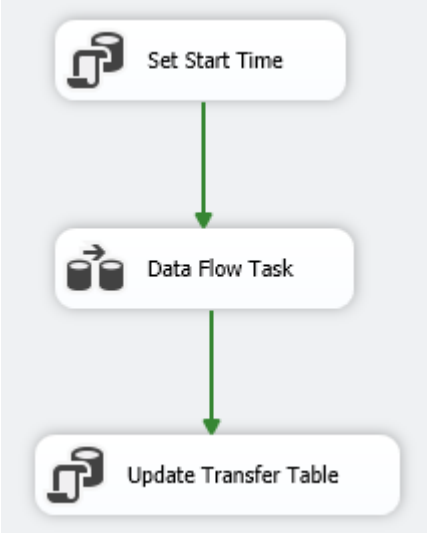
- STG_Sales Package – Data Flow

  In the data flow, the mirror tables (MRR_Invoices and MRR_InvoiceItems) are joined, and the data is loaded to STGSales table.



- DWH_Sales Package – Control Flow

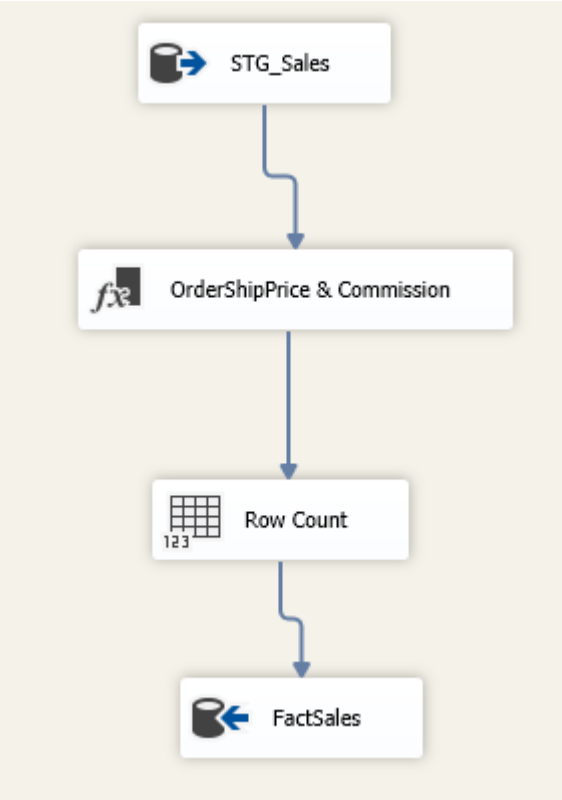
  Data is loaded from STGSales to FactSales and commission and shipping price columns are added.



- DWH_Sales Package – Data Flow

Prior data exploration revealed that there is a difference between the total expected price and the actual total price after adding all extra charges (such as taxes). Commission and shipping price were identified as this difference according to whether the purchase took place in-store or online.

For online orders (AgentID = 77777), the shipping price was calculated by subtracting the QPrice (the total order price: Quantity * Product Price * (1 - Discount)) from the Total Price. For other orders, in-store orders (AgentID ≠ 77777), the difference between Total Price and QPrice represents agent's commission.



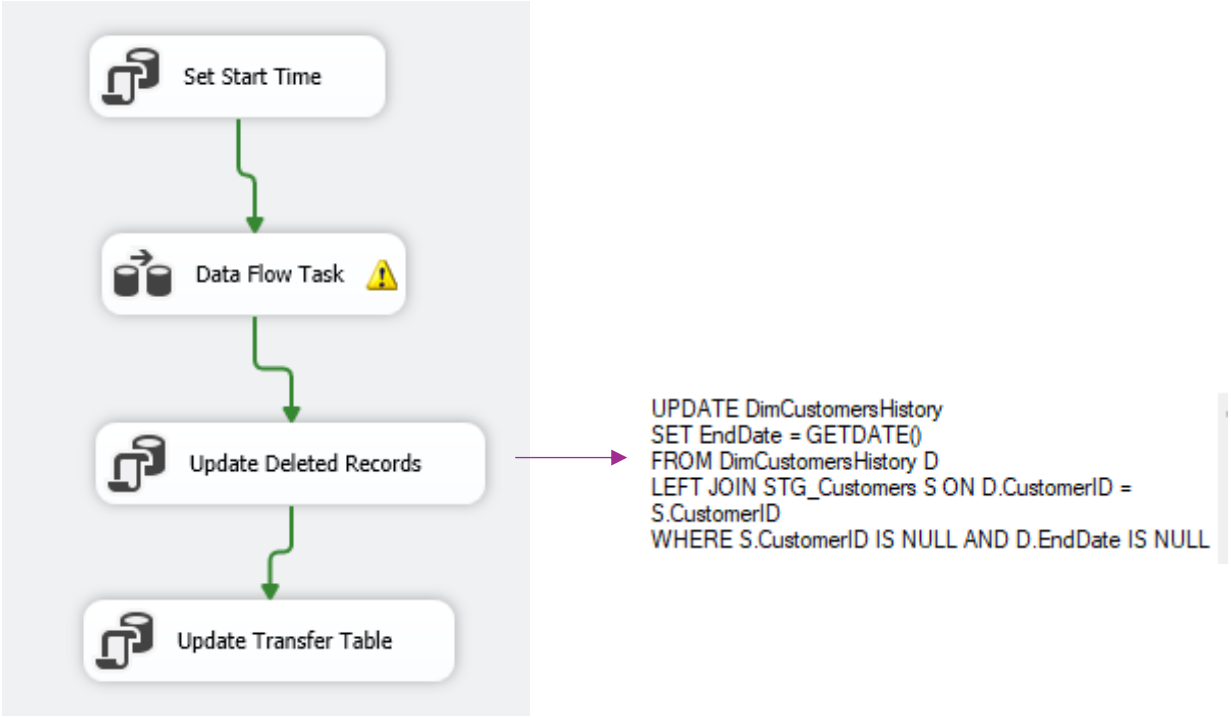
Derived Column Name	Derived Column	Expression
OrderShipPrice	<add as new column>	AgentID == 77777 ? (TotalPrice - QPRICE) : 0
Commission	<add as new column>	AgentID != 77777 ? (TotalPrice - QPRICE) : 0

History Tables

DimCustomersHistory Table

Data is incrementally loaded and updated in DimCustomersHistory. Deleted records are updated in DimCustomersHistory using an Execute SQL task.

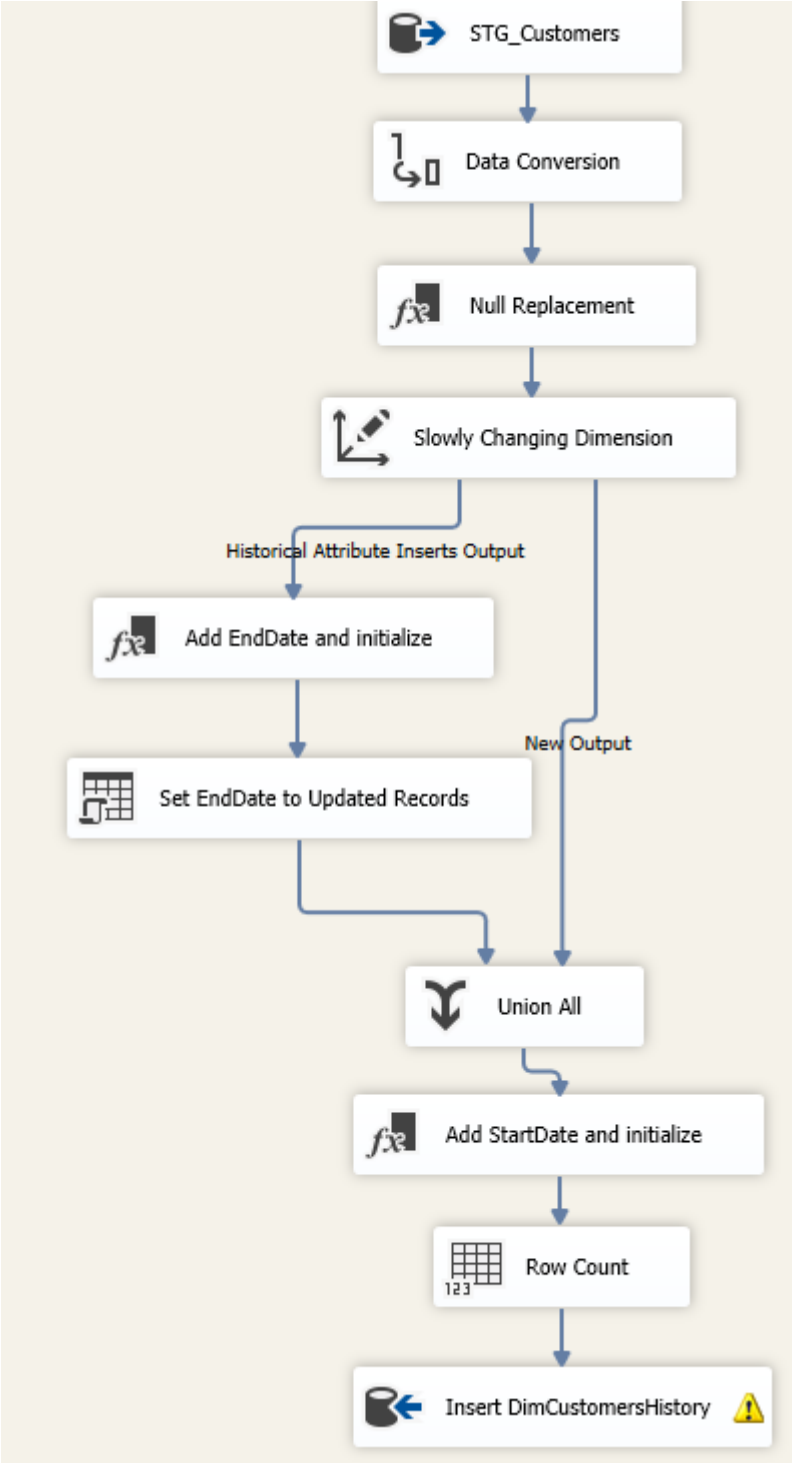
- DWH_Customers_History Package – Control Flow



Deleted records are handled in the same manner as in the DWH_Customers package but while using EndDate column instead of IsActive.

- DWH_Customers_History Package – Data Flow

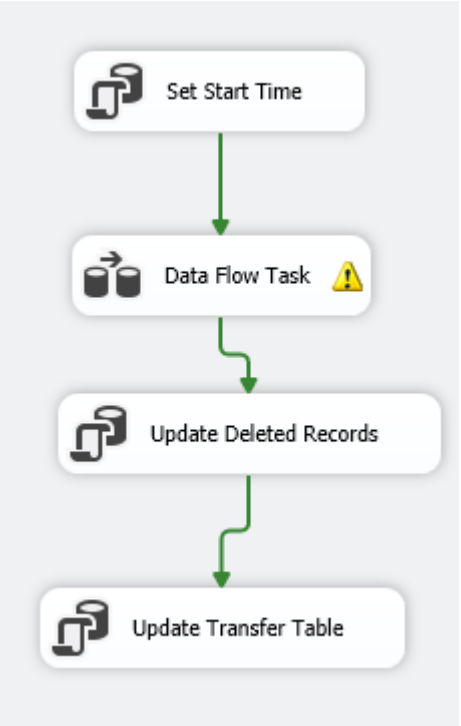
  Incremental load to the DimCustomersHistory table is done using the Slowly Changing Dimension transformation (change type: Historical Attribute).



DimProductsHistory Table

Data is incrementally loaded and updated in DimProductsHistory. Deleted records are updated in DimProductsHistory using an Execute SQL task.

- DWH_Products_History Package – Control Flow

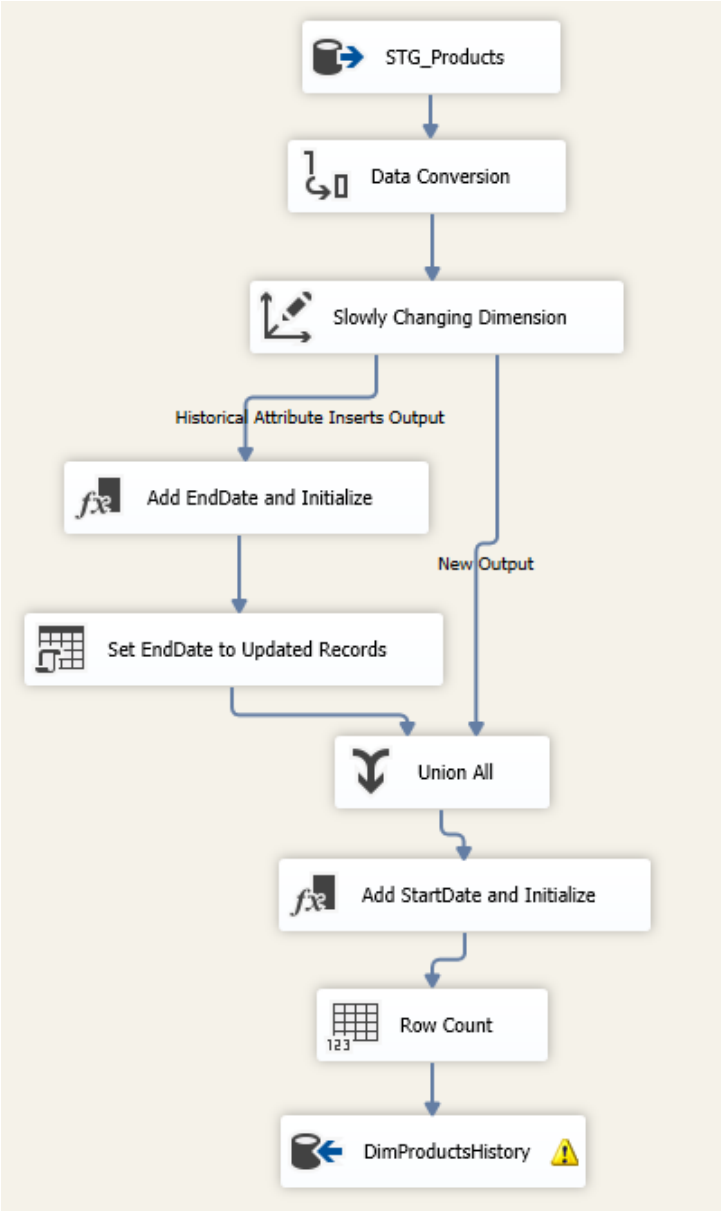



```
UPDATE DimProductHistory
SET EndDate = GETDATE()
FROM DimProductHistory D
LEFT JOIN STG_Products S ON D.ProductID =
S.ProductID
WHERE S.ProductID IS NULL AND D.EndDate IS NULL
```

Deleted records are handled in the same manner as in the DWH_Products package but while using EndDate column instead of IsActive.

- DWH_Products_History Package – Data Flow

  Incremental load to the DimProductsHistory table is done using the Slowly Changing Dimension transformation (change type: Historical Attribute).



Automatic Process

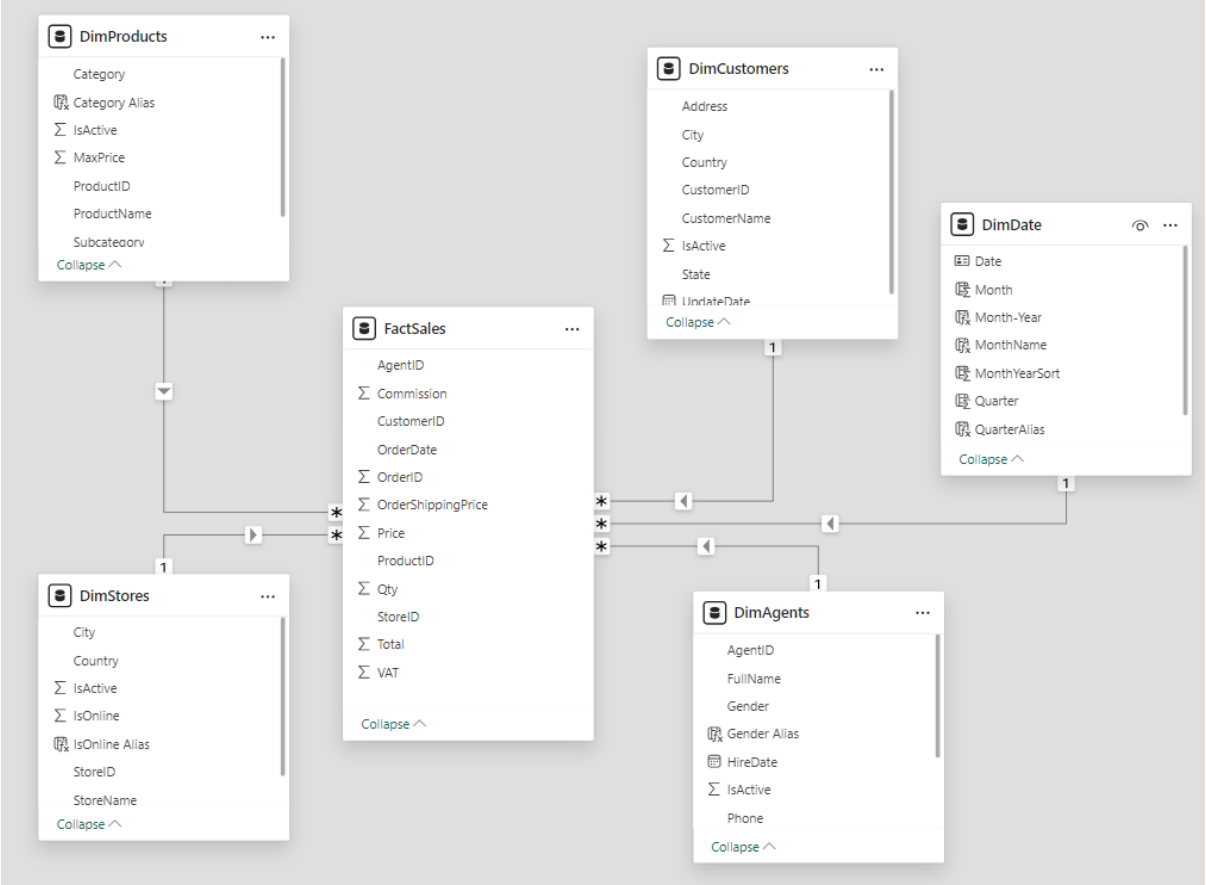
The project is deployed to SSMS and automated using SQL Agent jobs. Jobs are executing automatic data refresh daily at 5 AM. The first job to be executed is MRR_Tables which on success executes the next job and so on. The first jobs to run are the jobs constructing the dimension tables and the last job to run is the one constructing FactSales table.

- WD -MRR_Tables
- WD - DimCustomers
- WD - DimStores
- WD - DimAgents
- WD - DimProducts
- WD - FactSales

4.3. Visualizations in Power BI

4.3.1. ERD Model

The reports were created using Power BI Desktop and were published to Power BI Service. The model in Power BI includes one Fact table and the 4 Dimension tables (history tables not included). To these tables, a DimDate table was added.

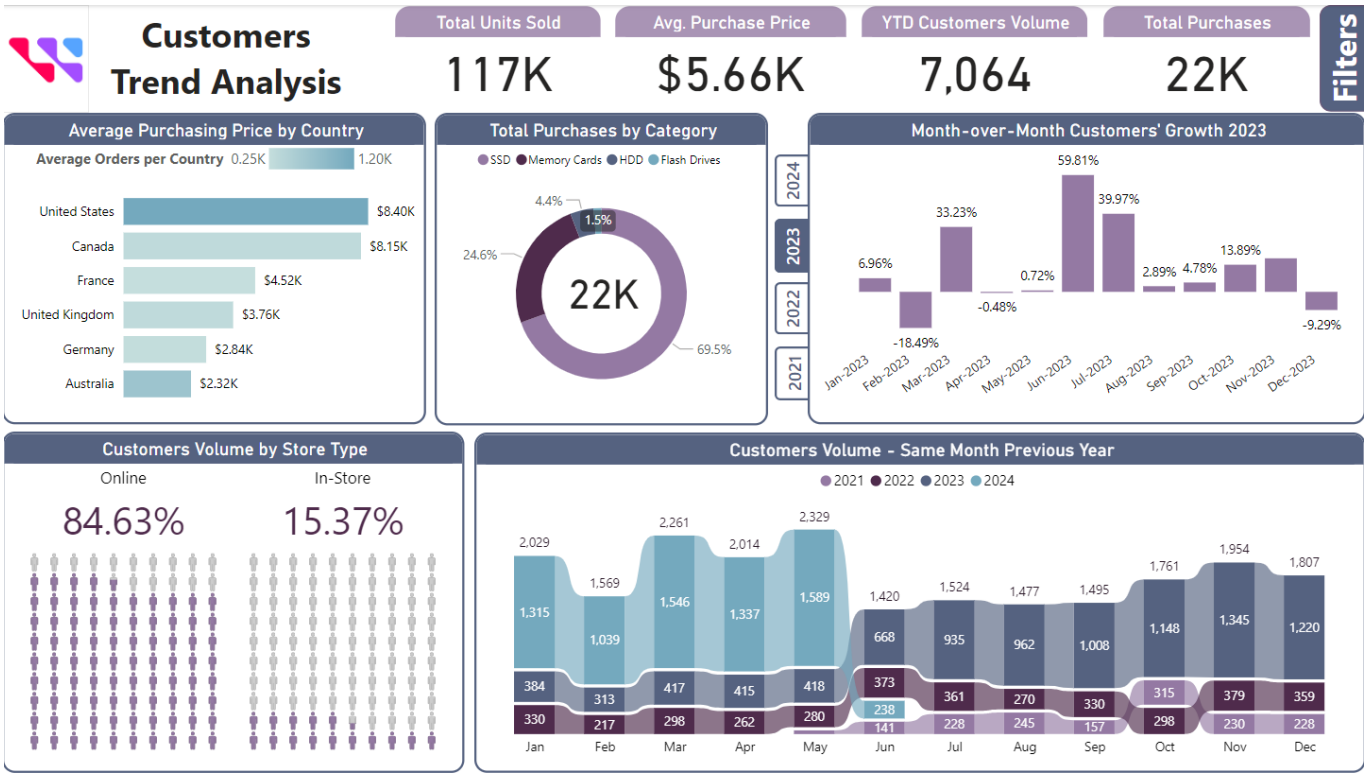


4.3.2. Reports & Executive Dashboard

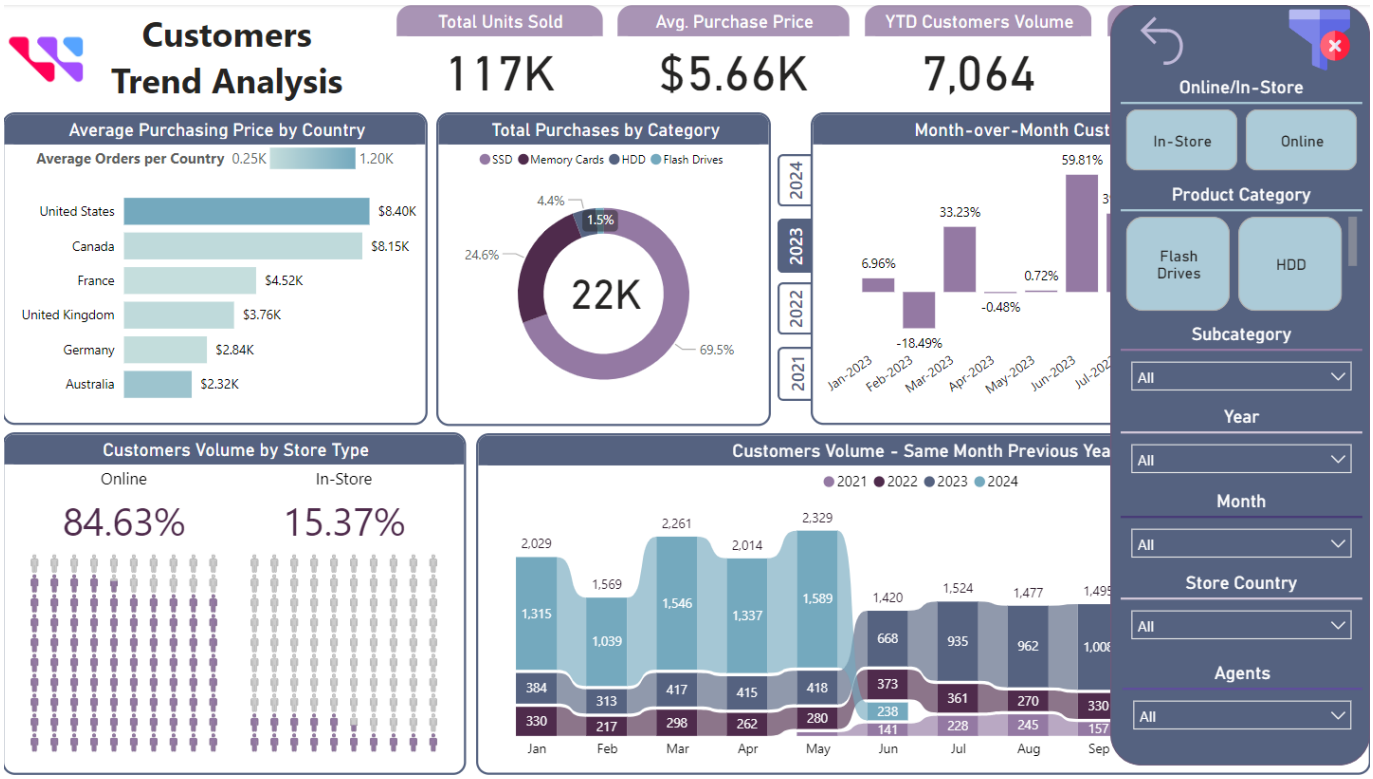
The project includes 2 reports (Sales Analysis, Customer Trend Analysis) and 1 Executive Dashboard. The reports load data from WDSalesDM for years 2021, 2022, 2023, 2024. Both 2021 and 2024 have partial data for the year (not all months are available).

4.3.2.1. Customers Trend Analysis Report

This report focuses on customers' preferences and it aims to provide a better understanding of customers' behavior. Month-over-Month Customers' Growth visualization can be adjusted to show the data for any of the years from 2021 to 2024 as needed. The filter pane can be opened and closed as needed for user convenience.



With an opened filter pane:



KPI Cards

- Total Units Sold
- Average Purchase Price
- YTD Customers Volume
- Total Purchases

Charts

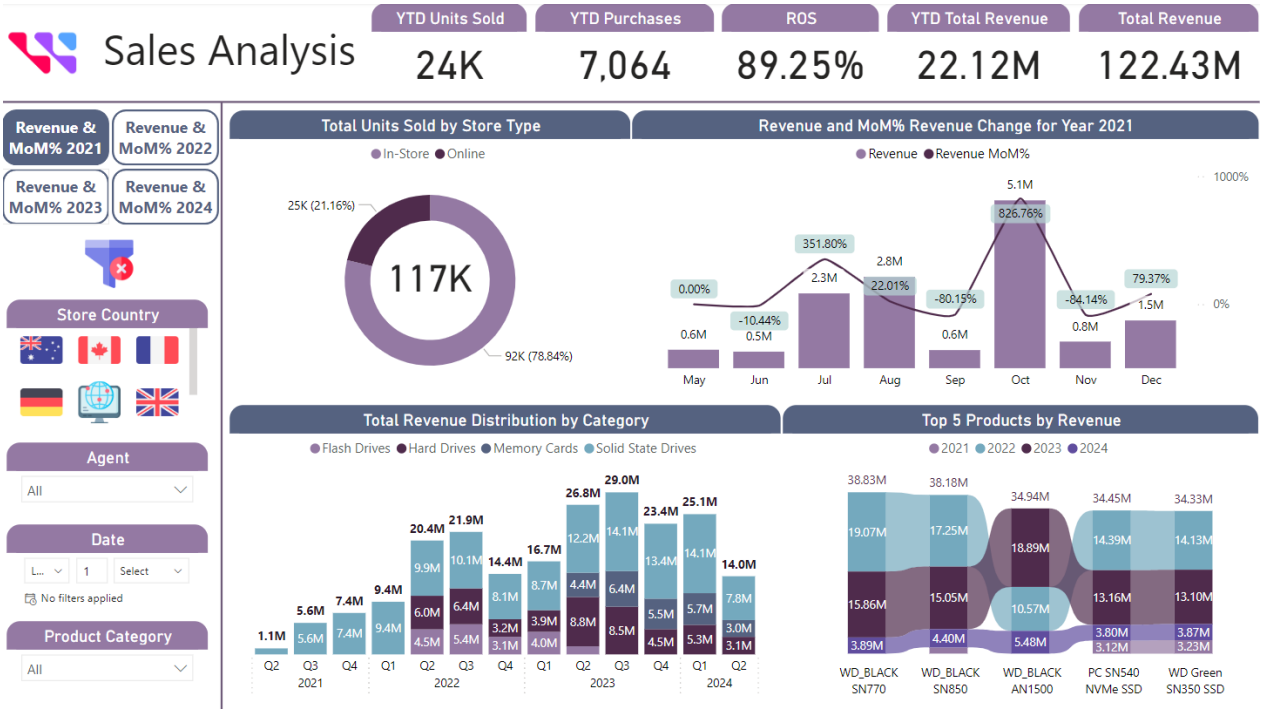
- Average Purchasing Price by Country (the color of the bars indicates the average amount of orders processed in each country)
- Total Purchases by Category
- Month-over-Month Customers' Growth
- Customers Volume by Store Type
- Customers Volume – Same Month Previous Year

Slicers

- Online/In-Store (Store Type)
- Product Category
- Product Subcategory
- Year
- Month
- Store Location (Country)
- Agents

4.3.2.2. Sales Analysis Report

This report focuses on sales trends over time and profitability of various factors such as product categories, store type, and individual products. The buttons on the left in the filter pane section, control the year for which the data in "Revenue and MoM% Revenue Change" visualization is displayed.



KPI Cards

- YTD Units Sold
- YTD Purchases
- ROS
- YTD Total Revenue
- Total Revenue

Charts

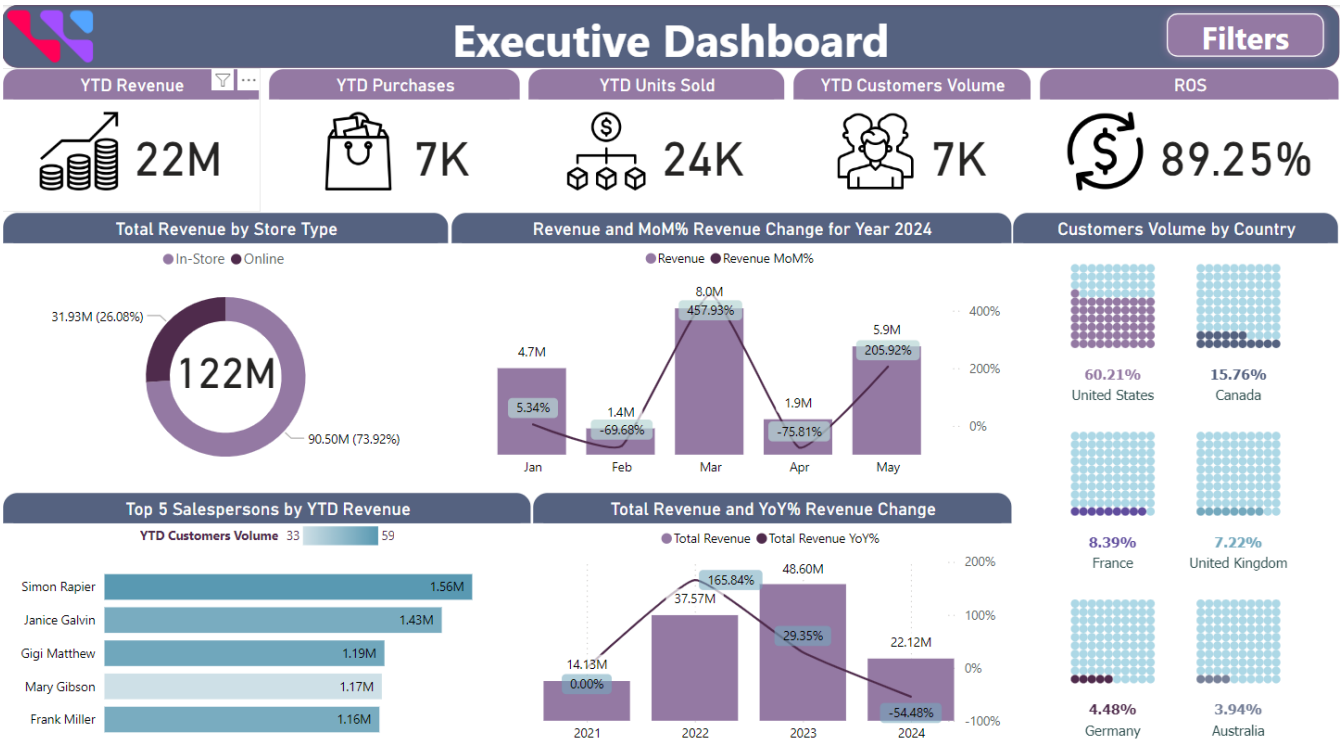
- Total Units Sold by Store Type (In-Store purchase/Online Purchase)
- Revenue and Month-over-Month % Revenue Change (for years 2021 – 2024)
- Total Revenue Distribution by Category
- Top 5 Products by Revenue

Slicers

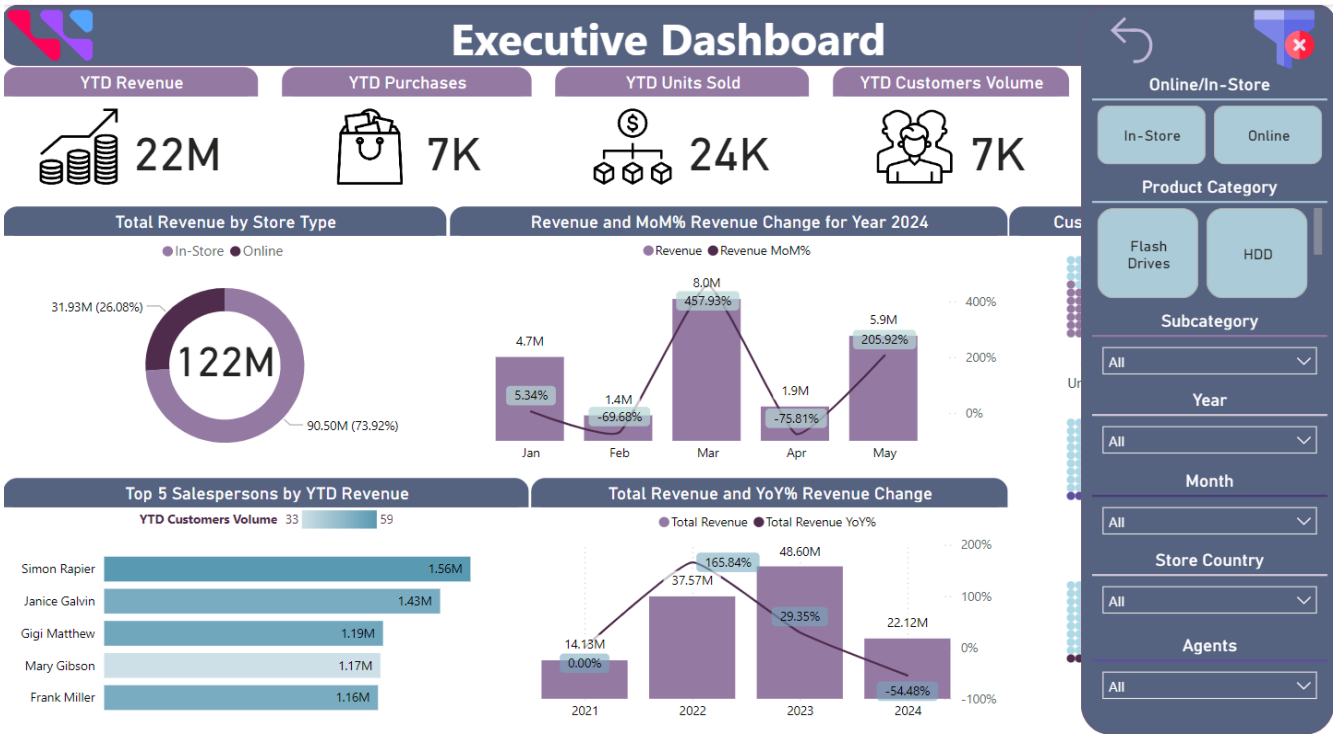
- Store Location (Country)
- Agent
- Date Range
- Product Category

4.3.2.3. Executive Dashboard

This dashboard provides an overview of the company's current state, including key performance indicators for the current year and revenue data from various perspectives. The included visualizations help the company assess its current financial situation.



With an opened filter pane:



KPI Cards

- YTD Revenue
- YTD Purchases
- YTD Units Sold
- YTD Customers Volume
- ROS

Charts

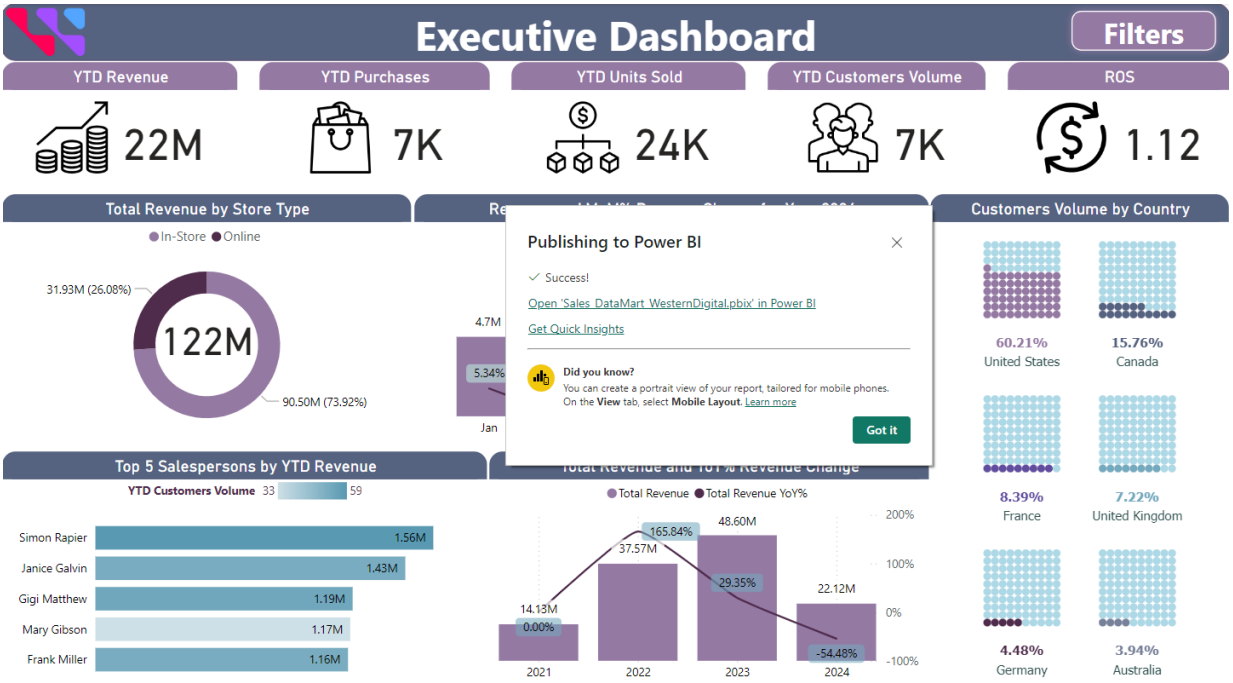
- Total Revenue by Store Type
- Revenue and Month-over-Month % Revenue Change for Year 2024
- Customers Volume by Country
- Top 5 Salespersons by YTD Revenue (the color of the bars indicates YTD Customers Volume for every agent).
- Total Revneu and Year-over-Year % Revenue Change

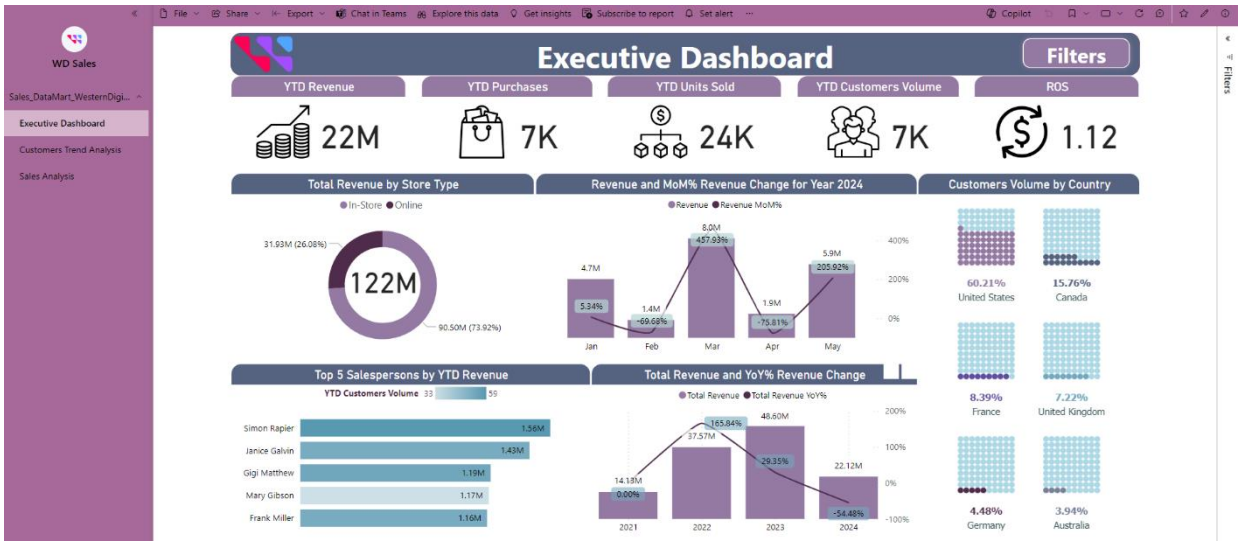
Slicers

- Online/In-Store (Store Type)
- Product Category
- Product Subcategory
- Year
- Month
- Store Location (Country)
- Agents

4.3.3. Publish

The reports were published to PBI Service and an App was created.





4.3.4. Scheduled Data Refresh

Gateway is configured for the data to be refreshed daily at 5 AM.