



# **Apple's Sales Data Mart**

BI System Specifications Document

Date: 08/02/2024

Version: 4.00

Written by: Adi Aljadeff



## **Content**

1. General
  - 1.1. Project Objective (including information consumers)
  - 1.2. Project Contents
2. Preparation of a work plan, distribution of tasks and schedules for execution (Gantt)
3. Technical Specification:
  - 3.1. Prerequisites - A comprehensive list of systems including access methods
  - 3.2. Solution Architecture and Flow Chart (HLD)
4. Functional Specification:
  - 4.1. Creation of final Source to Target and ERD model
  - 4.2. Detailed description of all ETL process
  - 4.3. Description of data tables in the Data Mart
  - 4.4. Visualization in Power BI



## 1. General

### 1.1. Project Objective

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

Apple Inc. is a multinational technology company headquartered in Cupertino, California. It designs, develops, and sells consumer electronics, computer software, and online services. The company's best-known devices include the iPhone, iPad, Mac, Apple Watch, and Apple TV. Apple's devices are sold all over the world, through Apple's physical stores, Apple online store, and authorized resellers (via physical and online stores). This project will focus only on sales performed by Apple stores (physical and online).

The Data Mart creation will be done using information derived from the PriorityERP database (Apple's operational database). The solution will include summarized data tables, focusing on devices sales data, as well as data regarding Apple'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 Apple'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 – ApplesSalesDM.

ERD model of the ApplesSalesDM database: [ERD Link](#)

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

- FactSales – Data regarding all sales, including the id of the order, products bought, quantities, and prices. Data loading process for this table will be incremental.
- DimCustomers – Data regarding the company's customers.
- DimStores – Data regarding the company's stores.
- DimEmployees – Data regarding the company's employees.
- DimProducts - Data regarding the company's products.
- DimProductsHistory – Historic data regarding the company's products.

[Source To Target Link](#)

The tables will be updated daily at 04:00:00 using an automated process configured in the SQL Server Management Studio.

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 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 customers analysis report will include data regarding Apple's customers by date, country, store, product, and category. This report is aimed to help Apple'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



- Average price per order
- Total units
- Average revenue per customer
- 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 and subcategory (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
- Percentage of new customers from total
- YTD customers
- Average revenue per customer
- Average orders number per customer
- Total customers and new customers by month and day
- Customers by country
- Customers by number of orders per customer
- Top products by customers
- Average revenue per customer compared to previous year by month and day

3.2.2.3. The executive dashboard will consist of:

- Total sales
- Average monthly revenue
- Total orders
- Total customers
- Percentage of new customers from total
- YTD sales
- MTD sales
- Total sales and month over month growth by quarter and month
- Revenue by online vs physical stores
- Top selling products by revenue
- Total customers by quarter and 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 AppleSalesDM database

- [ERD link](#)

### 4.2. ETL processes

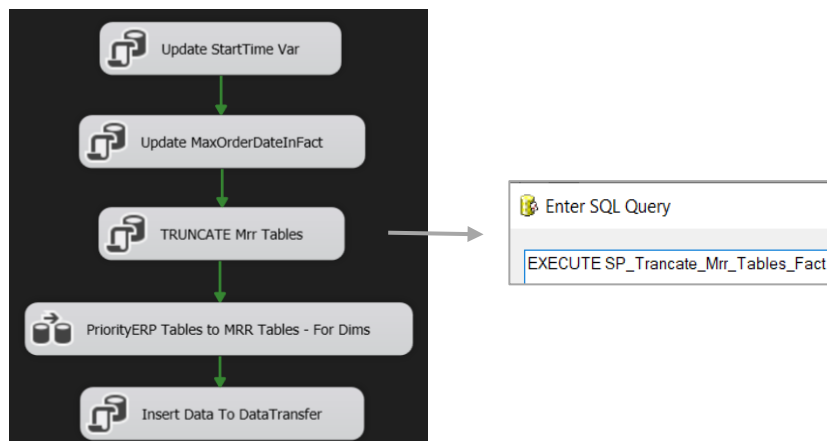
The ETL process was done in SSIS using 13 packages.

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.

- **FactSales Table:**

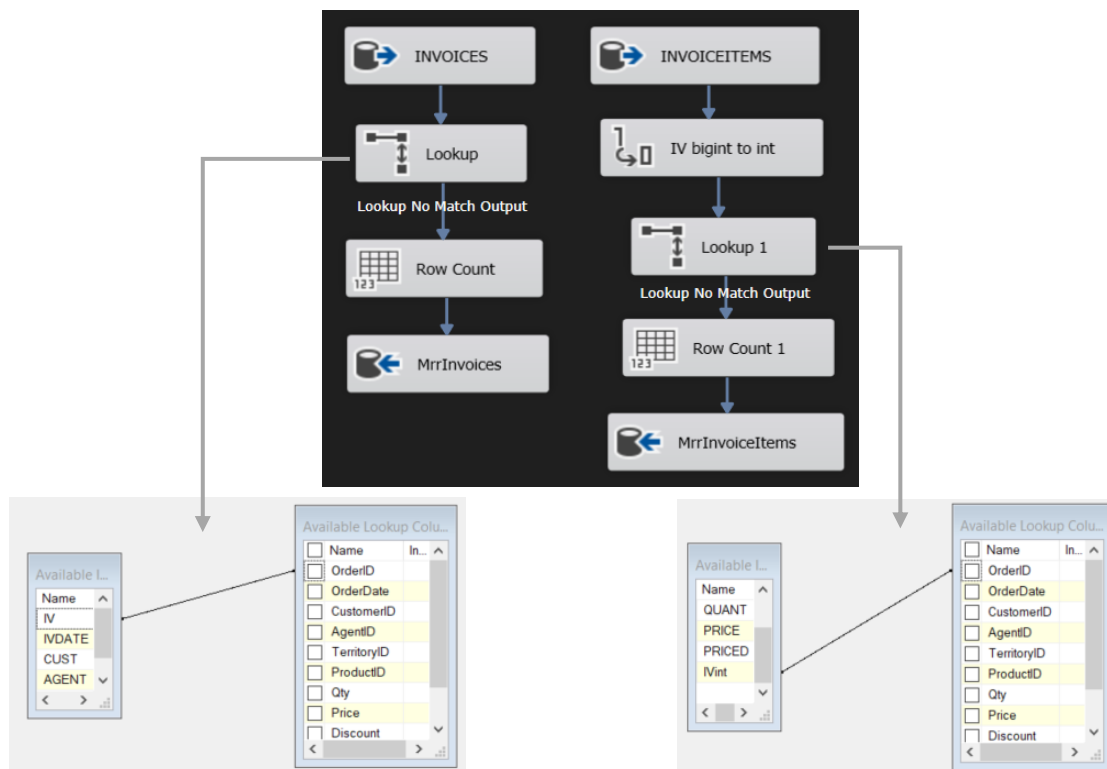
- MRR Sales package:

Mirror tables are truncated (using a stored procedure) and data is loaded from PriorityERP database (INVOICES and INVOICEITMES) to AppleSalesDM mirror tables.





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

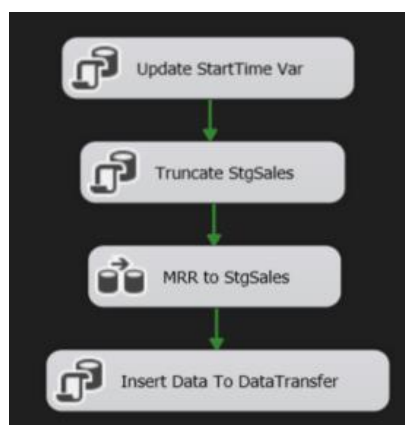


Data Conversion transformation is used to convert the IV column from bigint to int (DT\_I4), this is necessary for the lookup transformation:

Input Column	Output Alias	Data Type
IV	IVint	four-byte signed integer [DT_I4]

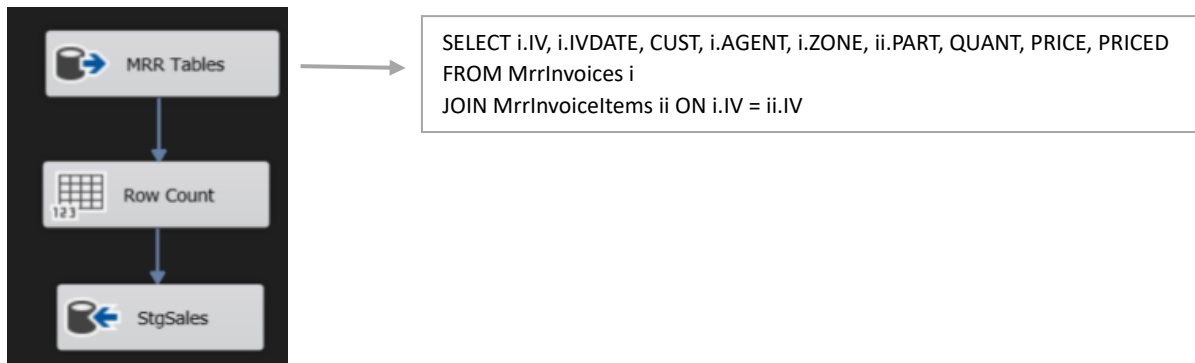
- STG\_Sales package:

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



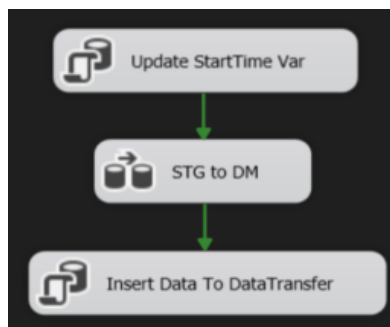


In the data flow, the mirror tables (MrrInvoices and MrrInvoiceItems) are joined, and the data is loaded to StgSales table.

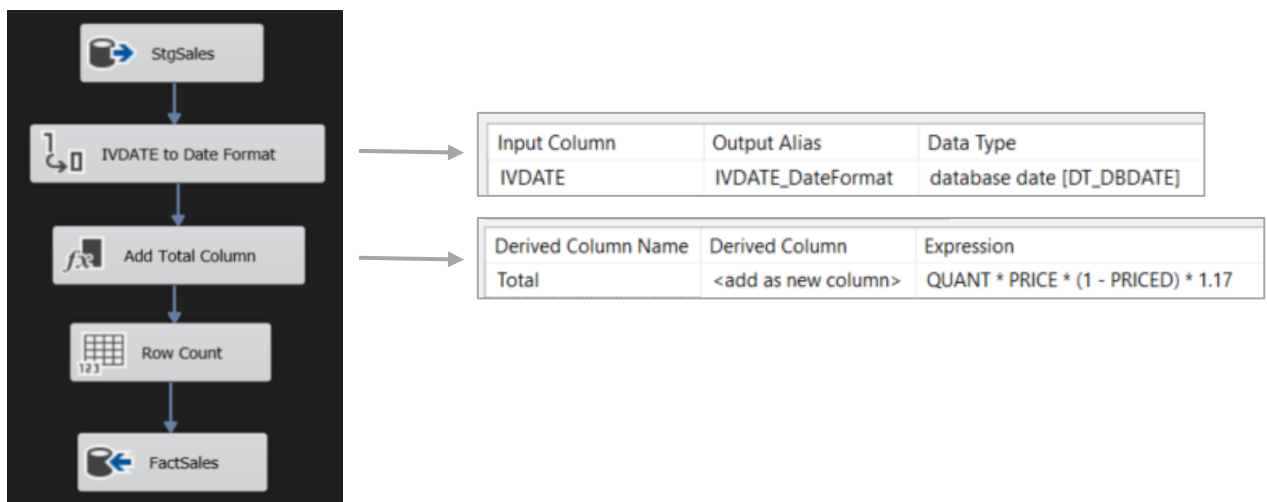


- DM Sales package:

Data is loaded from StgSales to FactSales, and a Total column is added.



In the data flow, IVDATE column is converted from string to date format (DT\_DBDATE) in a data conversion transformation. A Total column, which specifies the total price for the product in a sale, is calculated and added using a derived column transformation.



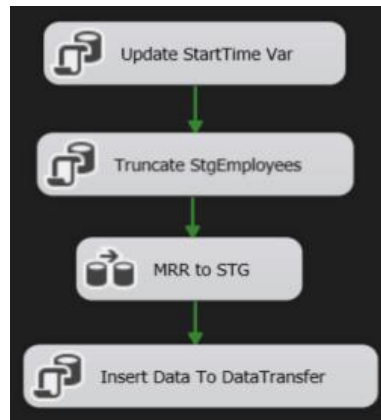
- **DimEmployees Table:**

- STG\_Employees package:

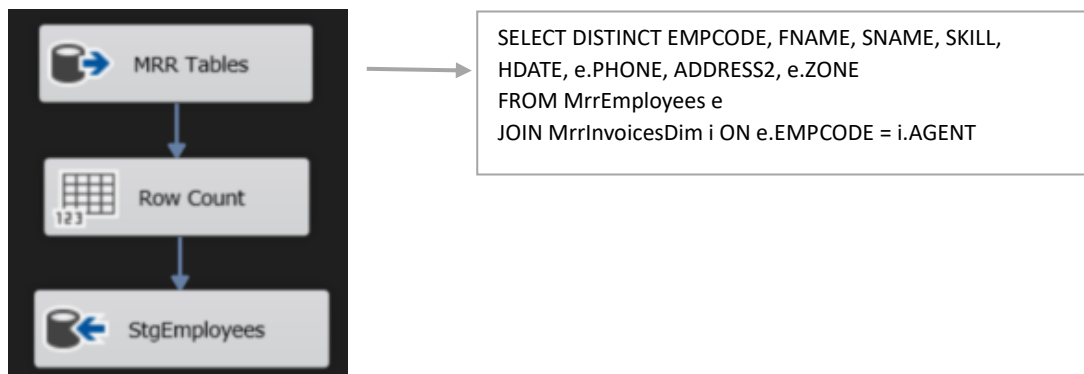




StgEmployees table is truncated, and the mirror tables are joined and loaded using a data flow task.



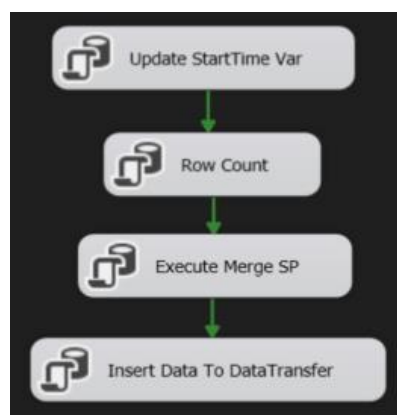
In the data flow, the mirror tables (MrrEmployees and MrrInvoicesDim) are joined, and the data is loaded to StgEmployees table.



This Join's objective is to exclude irrelevant employees from the sales data mart, i.e. include only employees which made a sale (thus are included in the INVOICES table).

- DM\_Employees package:

Data is incrementally loaded and updated in the DimEmployees.



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

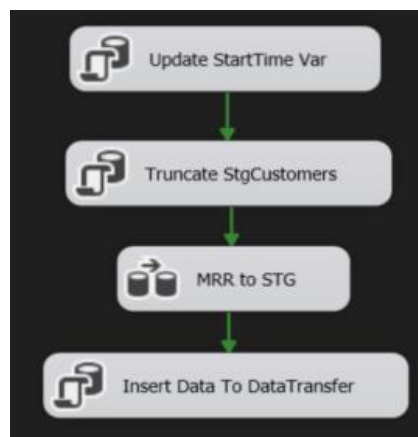


1. For **new** employees (EmployeeID exists in StgEmployees but not in DimEmployees): Insert new records to DimEmployees.
2. For **updated** employees (EmployeeID exists in StgEmployees and in DimEmployees but one or more of the other columns does not match): Update the record in DimEmployees.
3. For **deleted** employees (EmployeeID exists in DimEmployees but not in StgEmployees): Update the IsActive column in DimEmployees to 'N'.

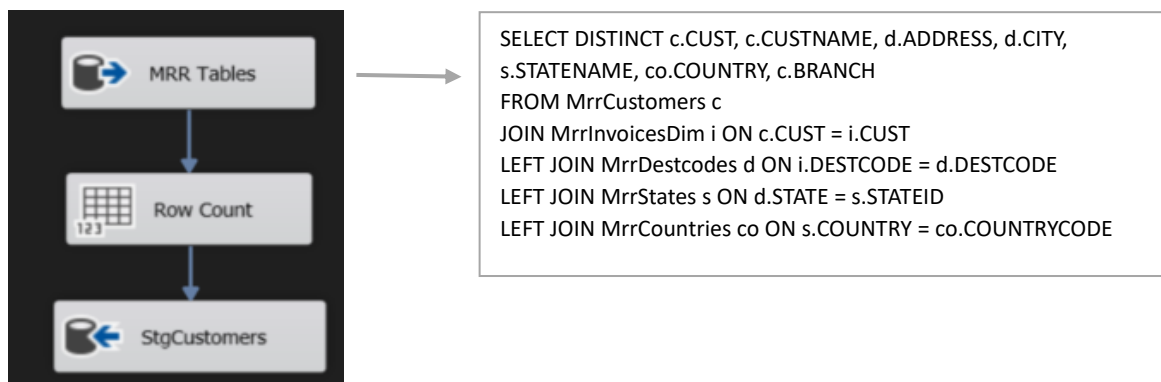
- **DimCustomers Table:**

- STG\_Customers package:

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



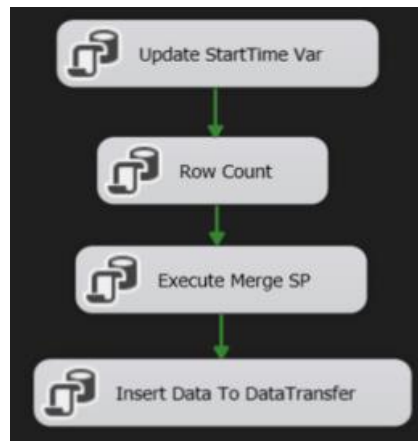
In the data flow, the 5 mirror tables (MrrCustomers, MrrInvoicesDim, MrrDestcodes, MrrStates, MrrCountries) are joined, and the data is loaded to StgCustomers table.



- DM\_Customers package:



Data is incrementally loaded and updated in DimCustomers.



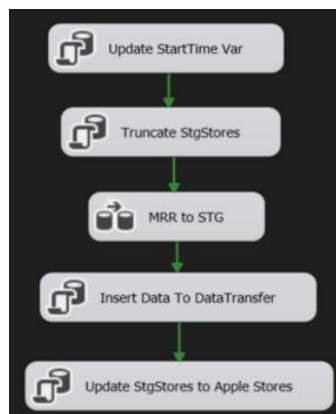
A merge stored procedure is executed in the Execute SQL Task, the merge statement works according to the following rational:

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

- **DimStores Table:**

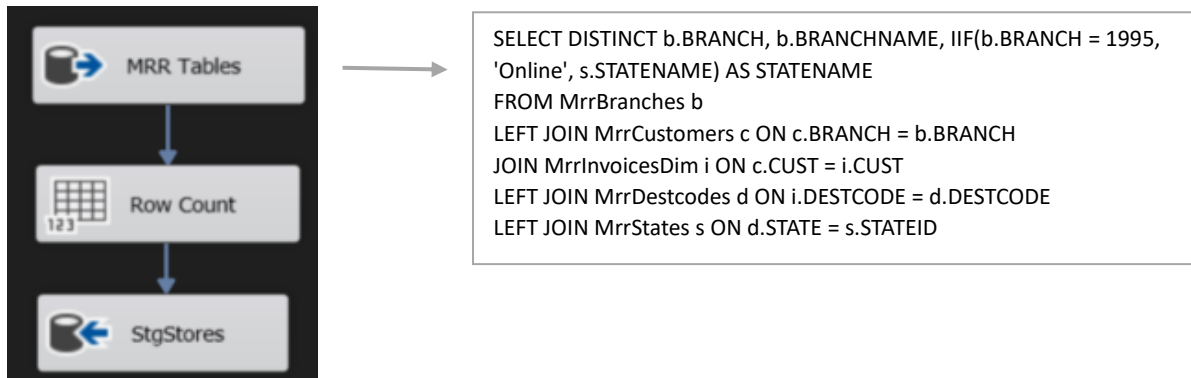
- STG Stores package:

StgStores table is truncated, and the mirror tables are joined and loaded using a data flow task. Store names are updated using a stored procedure, executed in an Execute SQL task.



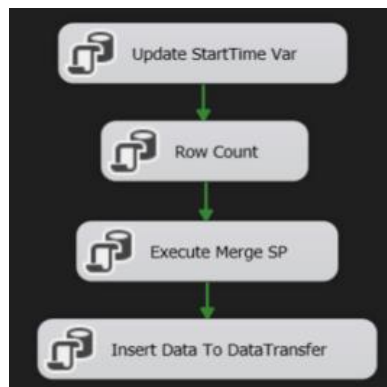


In the data flow, the 3 mirror tables (MrrBranches, MrrCustomers, MrrInvoicesDim) are joined, and the data is loaded to StgStores table.



- DM Stores package:

Data is incrementally loaded and updated in DimCustomers.



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

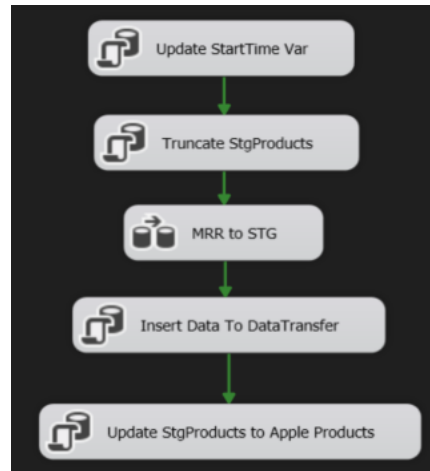
1. For **new** stores (storeId exists in StgStores but not in DimStores): Insert new records to DimStores.
2. For **updated** stores (storeId exists in StgStores and in DimStores but one or more of the other columns does not match): Update the record in DimStores.
3. For **deleted** stores (storeId exists in DimStores but not in StgStore): Update the IsActive column in DimStores to 'N'.

- **DimProducts Table:**

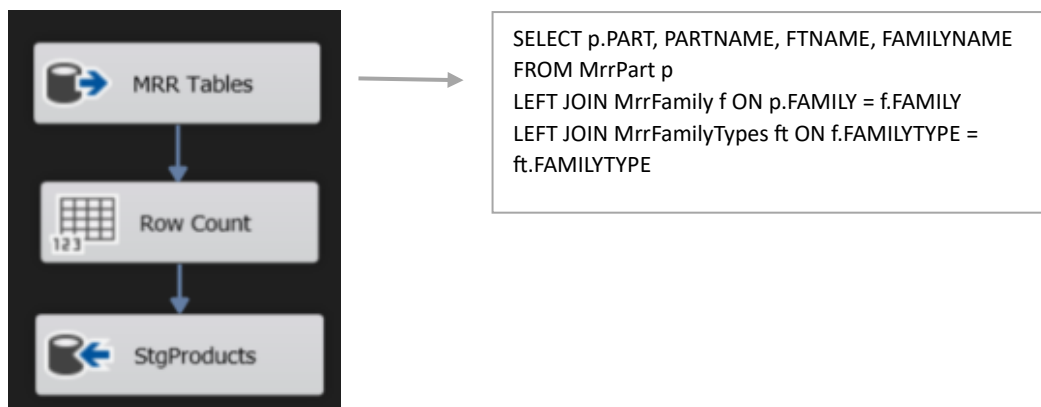
- STG Products package:



StgProducts table is truncated, and the mirror tables are joined and loaded using a data flow task. Product names, categories, and subcategories are updated using a stored procedure, executed in an Execute SQL task.

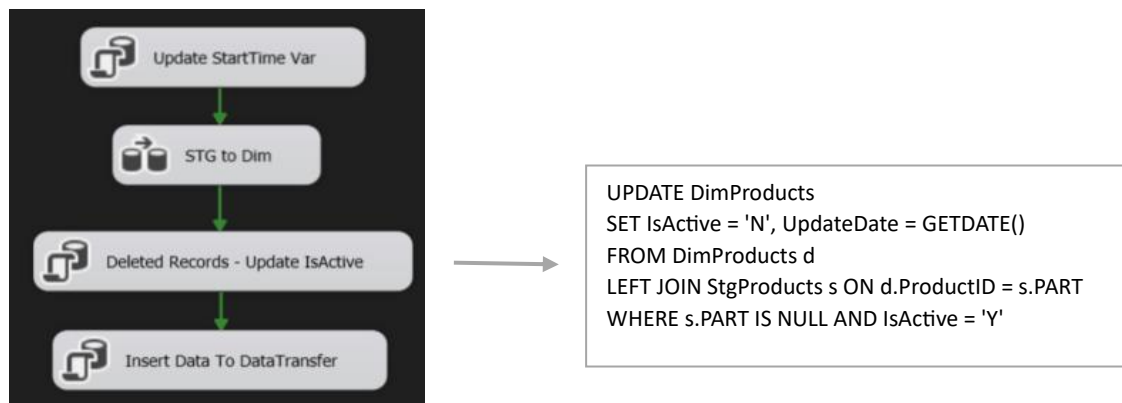


In the data flow, the 3 mirror tables (MrrPart, MrrFamily, MrrFamilyTypes) are joined, and the data is loaded to StgProducts table.



- DM Products package:

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

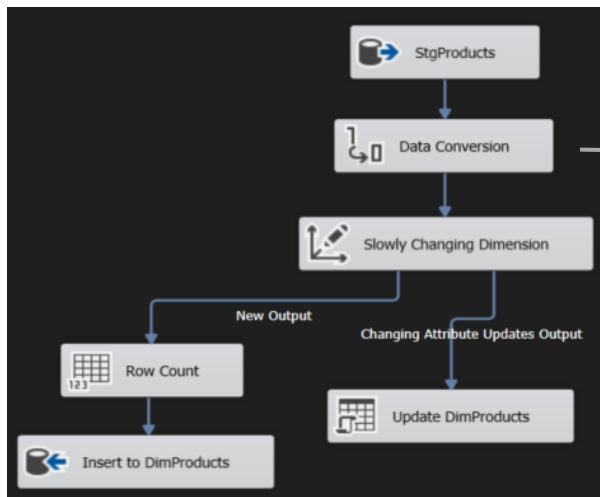


Deleted records are located using Left Join between DimProducts and StgProducts using ProductID as the key, while filtering records to products where ProductID in StgProducts is



NULL (i.e. products which cannot be found in StgProducts), and products where IsActive in DimProducts is 'Y' (i.e. they were not updated as deleted yet).

In the Data Flow, Incremental load to the DimProducts table is done using the Slowly Changing Dimension transformation (change type: Changing Attribute). 3 column data types are converted to Unicode (DT\_WSTR) using data conversion transformation to match destination data types.

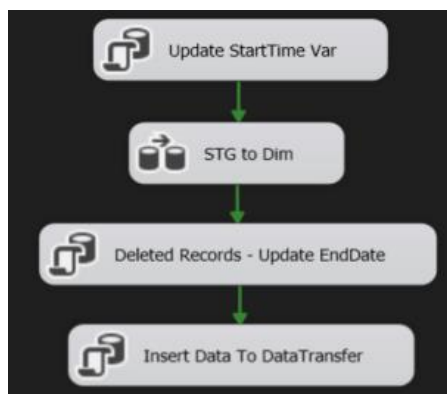


Input Column	Output Alias	Data Type	Length
PARTNAME	PARTNAME-unicode	Unicode string [DT_WSTR]	40
FAMILYNAME	FAMILYNAME-unicode	Unicode string [DT_WSTR]	35
FTNAME	FTNAME-unicode	Unicode string [DT_WSTR]	35

- **DimProductsHistory Table:**

- DM\_ProductsHistory package:

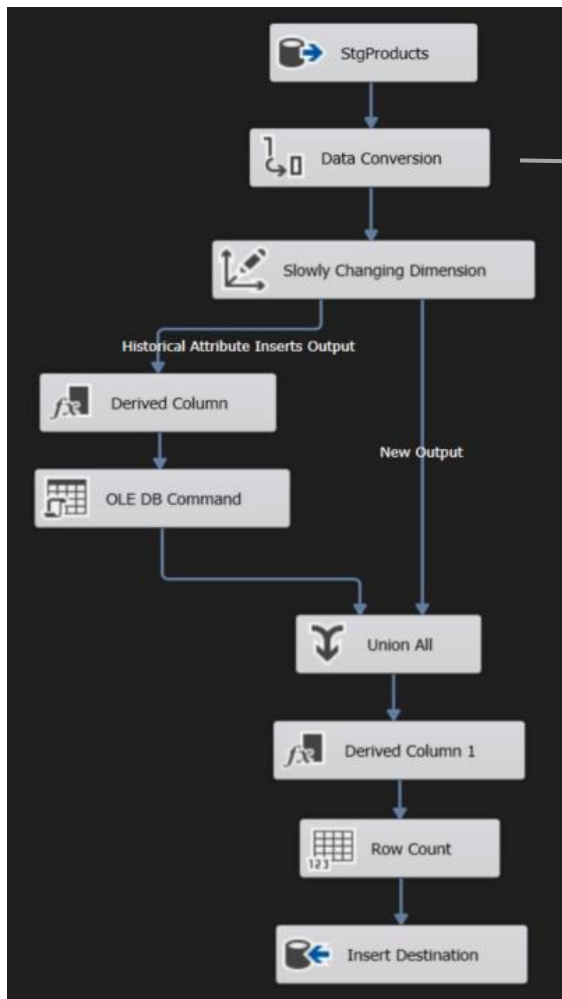
Data is incrementally loaded and updated in DimProductsHistory. Deleted records are updated in DimProductsHistory using an Execute SQL task.



```
UPDATE DimProductsHistory
SET EndDate = GETDATE()
FROM DimProductsHistory d
LEFT JOIN StgProducts s ON d.ProductID = s.PART
WHERE s.PART IS NULL AND d.EndDate IS NULL
```

Deleted records are located in the same manner as in the DM\_Products package but while using EndDate column instead of IsActive.

In the Data Flow, Incremental load to the DimProductsHistory table is done using the Slowly Changing Dimension transformation (change type: Historical Attribute). 3 column data types are converted to Unicode (DT\_WSTR) using data conversion transformation to match destination data types.

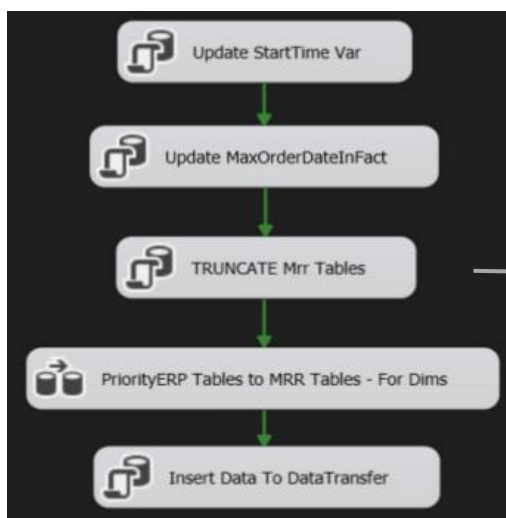


Input Column	Output Alias	Data Type	Length
PARTNAME	PARTNAME-unicode	Unicode string [DT_WSTR]	40
FAMILYNAME	FAMILYNAME-unicode	Unicode string [DT_WSTR]	35
FTNAME	FTNAME-unicode	Unicode string [DT_WSTR]	35

- **All Dim Tables – Mirror step**

- MRR Dim package:

This package is responsible for loading data from PriorityERP tables to all mirror tables relevant for the dim tables (10 tables in total). All mirror tables (except MrrInvoicesDim) are truncated using a stored procedure.



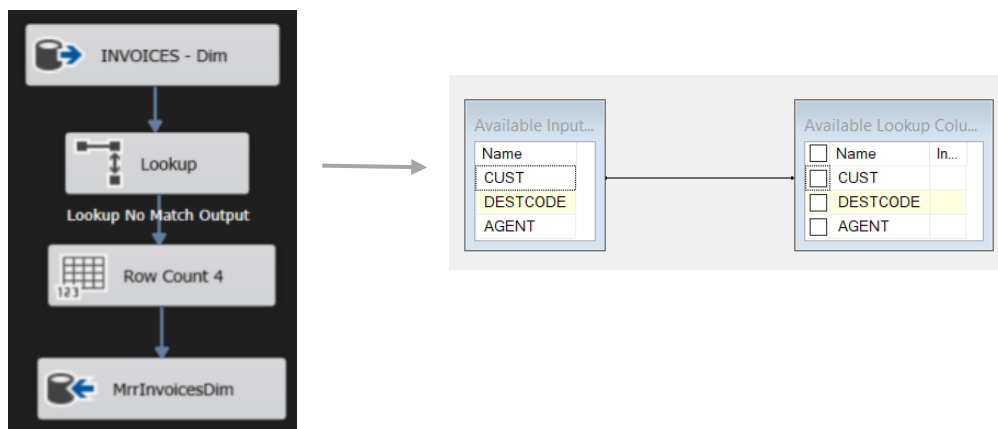
Enter SQL Query
EXECUTE SP_Truncate_Mrr_Tables_Dim



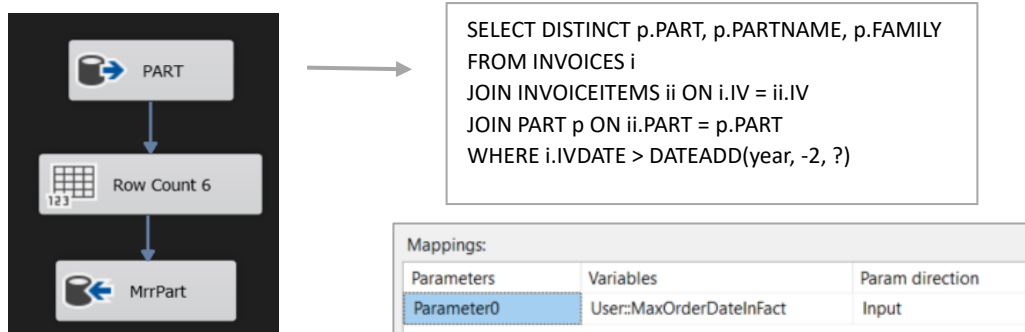
In the data flow:



Loading to MrrInvoicesDim is incremental and is performed using Lookup transformation (comparing records to MrrInvoicesDim), the MrrInvoicesDim is not truncated (compared to other mirror tables):



In order to filter out irrelevant product while loading to MrrPart, only products sold in the last 2 years are loaded to the table. This is done using a variable containing the max order date in the FactSales table:





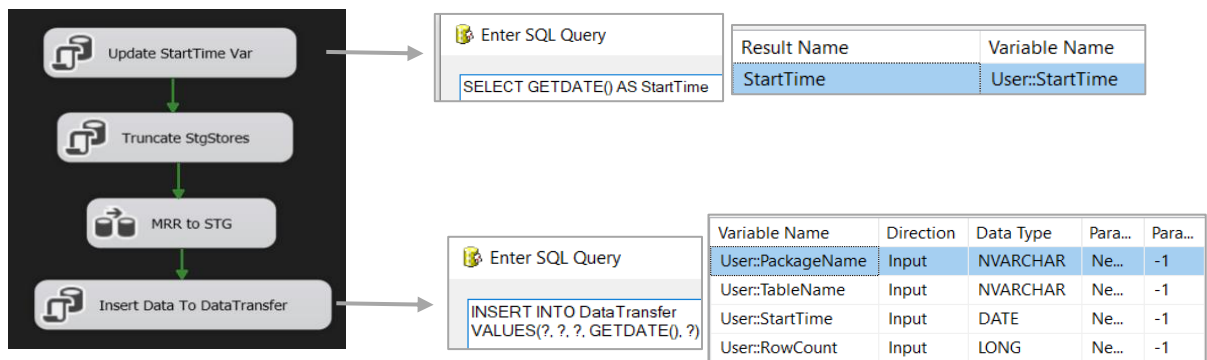


- **DataTransfer Table**

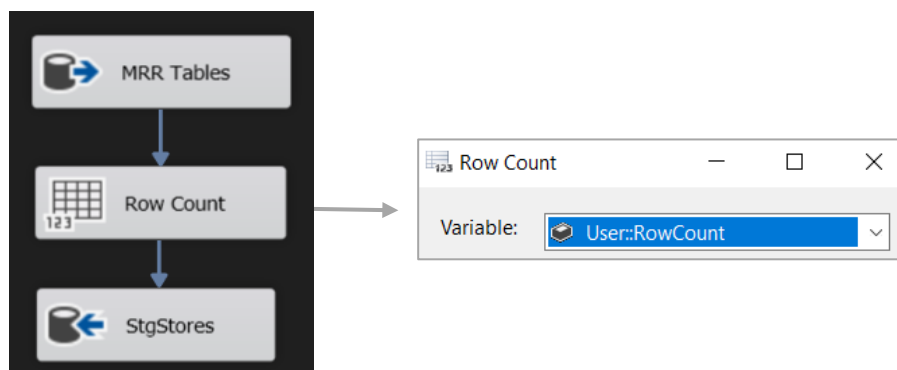
To monitor the ETL process, a dataTransfer table was created documenting each data insert: which table was updated and in which package, how many rows were inserted, and start and end times. The tasks and transformation in charge of the updates are included in all of the packages.

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.

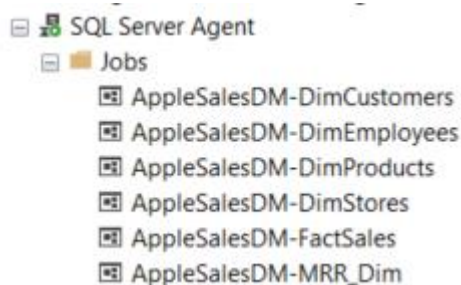


In the data flow the user variable RowCount is updated using a Row Count transformation.



- **Automatic Processing:**

The data is automatically refreshed daily at 4:00:00 using SQL Agent jobs, the first job executed is the AppleSalesDM-MRR\_DIM this job executes the next job and so on:





#### 4.3. Description of data tables in the Data Mart:

- FactSales - contains the following columns:
  1. OrderID: The ID of the order.
  2. OrderDate: The date the order took place in.
  3. CustomerID: The ID of the customer, foreign key to the DimCustomers table.
  4. AgentID: The ID of the salesperson, foreign key to the DimEmployees table.
  5. TerritoryID: The ID of the territory the order took place in.
  6. ProductID: The ID of the purchased product in the order, foreign key to the DimProducts table.
  7. Qty: The quantity of the purchased product in the order.
  8. Price: The price of one unit of the purchased product in the order.
  9. Discount: The discount given on the product price in the order (ranges between 0 to 1).
  10. Total: The total price for each product in the order, considering the price per unit, quantity, discount, and VAT of 17%. The following formula was used:  $\text{Price} * \text{Qty} * (1 - \text{Discount}) * 1.17$ .

	OrderID	OrderDate	CustomerID	AgentID	TerritoryID	ProductID	Qty	Price	Discount	Total
1	49893	2013-02-28	29783	276	3	725	2	202.33	0.00	473.45
2	49893	2013-02-28	29783	276	3	762	6	469.79	0.00	3297.95
3	49893	2013-02-28	29783	276	3	843	2	15.00	0.00	35.10
4	49893	2013-02-28	29783	276	3	760	11	454.13	0.02	5727.81
5	49893	2013-02-28	29783	276	3	855	5	53.99	0.00	315.86
6	49894	2013-02-28	30096	277	2	780	5	1242.85	0.00	7270.68
7	49894	2013-02-28	30096	277	2	787	4	647.99	0.00	3032.61
8	49894	2013-02-28	30096	277	2	831	1	209.25	0.00	244.82
9	49894	2013-02-28	30096	277	2	783	8	1229.45	0.00	11507.73
10	49894	2013-02-28	30096	277	2	788	3	647.99	0.00	2274.45
11	49894	2013-02-28	30096	277	2	815	3	36.44	0.00	127.92
12	49894	2013-02-28	30096	277	2	786	2	647.99	0.00	1516.30
13	49894	2013-02-28	30096	277	2	823	3	52.64	0.00	184.79
14	49894	2013-02-28	30096	277	2	748	4	744.27	0.00	3483.19
15	49894	2013-02-28	30096	277	2	825	2	196.32	0.00	459.40

- DimEmployees - The Dim\_Employees table contains the following columns:
  1. EmployeeID: The ID of the employee. This is the table's primary key.
  2. FirstName: The employee's first name.
  3. LastName: The employee's last name.
  4. JobTitle: The employee's job title.
  5. HireDate: The date the employee was hired.
  6. PhoneNumber: The employee's phone number.
  7. EmailAddress: The employee's email address.
  8. TerritoryName: The employee's territory (relevant only to salespersons).
  9. isActive: whether the employee is active ('Y' or 'N').
  10. UpdateDate: The date the record was last updated.



	Emp_ID	First_Name	Last_Name	Job_Title	Hire_Date	Phone_Number	Email_Address	Territory_Name	IsEngineer	IsActive	Update_Date
1	275	Michael	Blythe	Sales Representative	2011-05-31	257-555-0154	michael9@adventure-works.com	Northeast	N	1	2023-12-06
2	276	Linda	Mitchell	Sales Representative	2011-05-31	883-555-0116	linda3@adventure-works.com	Southwest	N	1	2023-12-06
3	277	Jillian	Carson	Sales Representative	2011-05-31	517-555-0117	jillian0@adventure-works.com	Central	N	1	2023-12-06
4	278	Garrett	Vargas	Sales Representative	2011-05-31	922-555-0165	garrett1@adventure-works.com	Canada	N	1	2023-12-06
5	279	Tsvi	Reiter	Sales Representative	2011-05-31	664-555-0112	tsvi0@adventure-works.com	Southeast	N	1	2023-12-06
6	280	Pamela	Ansman-Wolfe	Sales Representative	2011-05-31	340-555-0193	pamela0@adventure-works.com	Northwest	N	1	2023-12-06
7	281	Shu	Ito	Sales Representative	2011-05-31	330-555-0120	shu0@adventure-works.com	Southwest	N	1	2023-12-06
8	282	José	Saraiva	Sales Representative	2011-05-31	185-555-0169	josé1@adventure-works.com	Canada	N	1	2023-12-06
9	283	David	Campbell	Sales Representative	2011-05-31	740-555-0182	david8@adventure-works.com	Northwest	N	1	2023-12-06
10	284	Tete	Mensa-Annan	Sales Representative	2012-09-30	615-555-0153	tete0@adventure-works.com	Northwest	N	1	2023-12-06

- DimCustomers - contains the following columns:

1. CustomerID: The ID of the customer. This is the table's primary key.
2. Name: The customer's full name.
3. Address: The customer's address.
4. City: The customer's city.
5. Region: The customer's region.
6. Country: The customer's country.
7. StoreID: The customer's store (each CustomerID is related to one store only).
8. isActive: whether the customer is active ('Y' or 'N').
9. UpdateDate – The date the record was last updated.

	CustomerID	Name	Address	City	Region	Country	StoreID	IsActive	UpdateDate
1	11000	Jon Yang	3761 N. 14th St	Rockhampton	Queensland	Australia	1995	Y	2024-01-25 11:38:25.517
2	11001	Eugene Huang	2243 W St	Seaford	Victoria	Australia	1995	Y	2024-01-25 11:38:25.517
3	11002	Ruben Torres	5844 Linden Land	Hobart	Tasmania	Australia	1995	Y	2024-01-25 11:38:25.517
4	11003	Christy Zhu	1825 Village Pl.	North Ryde	New South Wales	Australia	1995	Y	2024-01-25 11:38:25.517
5	11004	Elizabeth Johnson	7553 Harness Circle	Wollongong	New South Wales	Australia	1995	Y	2024-01-25 11:38:25.517
6	11005	Julio Ruiz	7305 Humphrey Drive	East Brisbane	Queensland	Australia	1995	Y	2024-01-25 11:38:25.517
7	11006	Janet Alvarez	2612 Berry Dr	Matraville	New South Wales	Australia	1995	Y	2024-01-25 11:38:25.517
8	11007	Marco Mehta	942 Brook Street	Warrnambool	Victoria	Australia	1995	Y	2024-01-25 11:38:25.517
9	11008	Rob Verhoff	624 Peabody Road	Bendigo	Victoria	Australia	1995	Y	2024-01-25 11:38:25.517
10	11009	Shannon Carlson	3839 Northgate Road	Hervey Bay	Queensland	Australia	1995	Y	2024-01-25 11:38:25.517
11	11010	Jacquelyn Suarez	7800 Corrinne Court	East Brisbane	Queensland	Australia	1995	Y	2024-01-25 11:38:25.517
12	11011	Curtis Lu	1224 Shoenic	East Brisbane	Queensland	Australia	1995	Y	2024-01-25 11:38:25.517
13	11012	Lauren Walker	4785 Scott Street	Bremerton	Washington	United States	1995	Y	2024-01-25 11:38:25.517
14	11013	Ian Jenkins	7902 Hudson Ave.	Lebanon	Oregon	United States	1995	Y	2024-01-25 11:38:25.517
15	11014	Sydney Bennett	9011 Tank Drive	Redmond	Washington	United States	1995	Y	2024-01-25 11:38:25.517

- DimStores - contains the following columns:

1. StoreID: The ID of the Store. This is the table's primary key.
2. StoreName: The name of the store.
3. Location: The location of the store.
4. isActive: whether the store is active ('Y' or 'N').



	StoreId	StoreName	Location	IsActive	UpdateDate
1	318	Pioneer Place	Wyoming	Y	2024-01-25 10:23:21.710
2	372	Houston Galleria	Texas	Y	2024-01-25 10:23:21.710
3	478	The Grove	California	Y	2024-01-25 10:23:21.710
4	518	University Village	Washington	Y	2024-01-25 10:23:21.710
5	592	Richmond Centre	British Columbia	Y	2024-01-25 10:23:21.710
6	598	Baybrook	Texas	Y	2024-01-25 10:23:21.710
7	600	Alderwood	Washington	Y	2024-01-25 10:23:21.710
8	648	Sherway Gardens	Ontario	Y	2024-01-25 10:23:21.710
9	652	Del Monte	California	Y	2024-01-25 10:23:21.710
10	662	Baybrook	Texas	Y	2024-01-25 10:23:21.710
11	762	Ridgedale	Washington	Y	2024-01-25 10:23:21.710
12	786	Pheasant Lane	New Hampshire	Y	2024-01-25 10:23:21.710
13	808	Odysseum	Loiret	Y	2024-01-25 10:23:21.710
14	814	CoolSprings Galleria	Tennessee	Y	2024-01-25 10:23:21.710
15	816	Otay Ranch	California	Y	2024-01-25 10:23:21.710

- DimProducts - contains the following columns:

1. ProductID: The ID of the product. This is the table's primary key.
2. ProductName: The name of the product.
3. SubCategory: The product's subcategory (for example Headphones & Speakers, Pro).
4. Category: The product's category (for example iPad, Mac, iPhone).
5. IsActive: whether the product is sold ('Y' or 'N'), products which were not sold for 2 years will be updated to 'N'.
6. UpdateDate – The date the record was last updated.

	ProductID	ProductName	Category	SubCategory	IsActive	UpdateDate
1	743	iPad Pro 12.9-inch	iPad	Pro	Y	2024-01-29 08:46:05.623
2	748	MacBook Pro 16-inch	Mac	Pro	Y	2024-01-29 08:46:05.623
3	760	iPhone SE	iPhone	iPhoneSE	Y	2024-01-29 08:46:05.623
4	762	iPad (9th gen.)	iPad	iPad	Y	2024-01-25 09:49:47.413
5	770	Apple Vision Pro Battery	Accessories	Charging Essentials	Y	2024-01-29 08:46:05.623
6	779	iPhone 14	iPhone	iPhone14	Y	2024-01-29 08:46:05.623
7	780	iPhone 15 Plus	iPhone	iPhone15	Y	2024-01-29 08:46:05.623
8	781	AirPods (2nd generation)	Accessories	Headphones & Speakers	Y	2024-01-29 08:46:05.623
9	782	iPhone 15	iPhone	iPhone15	Y	2024-01-29 08:46:05.623
10	783	iPhone 15 Pro	iPhone	iPhone15	Y	2024-01-29 08:46:05.623
11	784	AirPods Pro (2nd generation)	Accessories	Headphones & Speakers	Y	2024-01-29 08:46:05.623
12	789	iPad Pro 11-inch	iPad	Pro	Y	2024-01-29 08:46:05.623
13	790	iPhone 15 Pro FineWoven Case	Accessories	Cases & Protection	Y	2024-01-29 08:46:05.623
14	791	Magic Mouse	Accessories	Mice & Keyboards	Y	2024-01-29 08:46:05.623
15	792	iPad Air	iPad	Air	Y	2024-01-29 08:46:05.623

- DimProductsHistory - contains the same columns as the DimProducts (except the IsActive column) along with the following additional columns:

1. InsertDate: The date the specific record was inserted to the table, whether because a new product was inserted, or because the product was updated, and a new record was inserted.



2. EndDate: The date the specific record had become 'irrelevant' whether because the product was deleted, or because the product was updated, and a new updated record of the product was inserted making the 'old' record irrelevant. When the EndDate value is NULL this means the record specifies the most recent update of the product.

	ProductID	ProductName	Category	SubCategory	InsertDate	EndDate
1	743	iPad Pro 12.9-inch	iPad	Pro	2024-01-25 17:19:51.000	NULL
2	748	MacBook Pro 16-inch	Mac	Pro	2024-01-25 17:19:51.000	NULL
3	760	iPhone SE	iPhone	iPhoneSE	2024-01-25 17:19:51.000	NULL
4	762	iPad (9th gen.)	iPad	iPad	2024-01-25 17:19:51.000	NULL
5	770	Apple Vision Pro Battery	Accessories	Charging Essentials	2024-01-25 17:19:51.000	NULL
6	779	iPhone 14	iPhone	iPhone14	2024-01-25 17:19:51.000	NULL
7	780	iPhone 15 Plus	iPhone	iPhone15	2024-01-25 17:19:51.000	NULL
8	781	AirPods (2nd generation)	Accessories	Headphones & Speakers	2024-01-25 17:19:51.000	NULL
9	782	iPhone 15	iPhone	iPhone15	2024-01-25 17:19:51.000	NULL
10	783	iPhone 15 Pro	iPhone	iPhone15	2024-01-25 17:19:51.000	NULL
11	784	AirPods Pro (2nd generation)	Accessories	Headphones & Speakers	2024-01-25 17:19:51.000	NULL
12	789	iPad Pro 11-inch	iPad	Pro	2024-01-25 17:19:51.000	NULL
13	790	iPhone 15 Pro FineWoven Case	Accessories	Cases & Protection	2024-01-25 17:19:51.000	NULL
14	791	Magic Mouse	Accessories	Mice & Keyboards	2024-01-25 17:19:51.000	NULL
15	792	iPad Air	iPad	Air	2024-01-25 17:19:51.000	NULL

- DataTransfer - contains the following columns:

1. TransferID: The ID of the transfer. This is the table's primary key.
2. PackageName: The name of the package which the table was created/updated in.
3. TableName: The name of the table documented.
4. StartInsertDatetime: The date and time of the package processing started.
5. EndInsertDatetime: The date and time of the package processing ended.
6. RowsNumber: The number of rows added to the table.

	TransferID	PackageName	TableName	StartInsertDatetime	EndInsertDatetime	RowsNumber
1	1	STG_Stores	StgStores	2024-01-28 16:21:44.000	2024-01-28 16:21:43.803	636
2	2	STG_Stores	StgStores	2024-01-28 16:22:34.000	2024-01-28 16:22:33.747	636
3	3	DM_Employees	DimEmployees	2024-01-28 16:24:46.000	2024-01-28 16:24:45.740	0
4	4	STG_Sales	StgSales	2024-01-28 16:25:31.000	2024-01-28 16:25:31.133	2
5	5	MRR_Dim	MrrInvoicesDim	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	0
6	6	MRR_Dim	MrrBranches	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	776
7	7	MRR_Dim	MrrCountries	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	238
8	8	MRR_Dim	MrrCustomers	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	19120
9	9	MRR_Dim	MrrDestcodes	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	19614
10	10	MRR_Dim	MrrEmployees	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	291
11	11	MRR_Dim	MrrFamily	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	38
12	12	MRR_Dim	MrrFamilytypes	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	4
13	13	MRR_Dim	MrrPart	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	4
14	14	MRR_Dim	MrrStates	2024-01-29 08:43:40.000	2024-01-29 08:43:41.540	181
15	15	STG_Customers	StgCustomers	2024-01-29 08:44:22.000	2024-01-29 08:44:34.920	19120

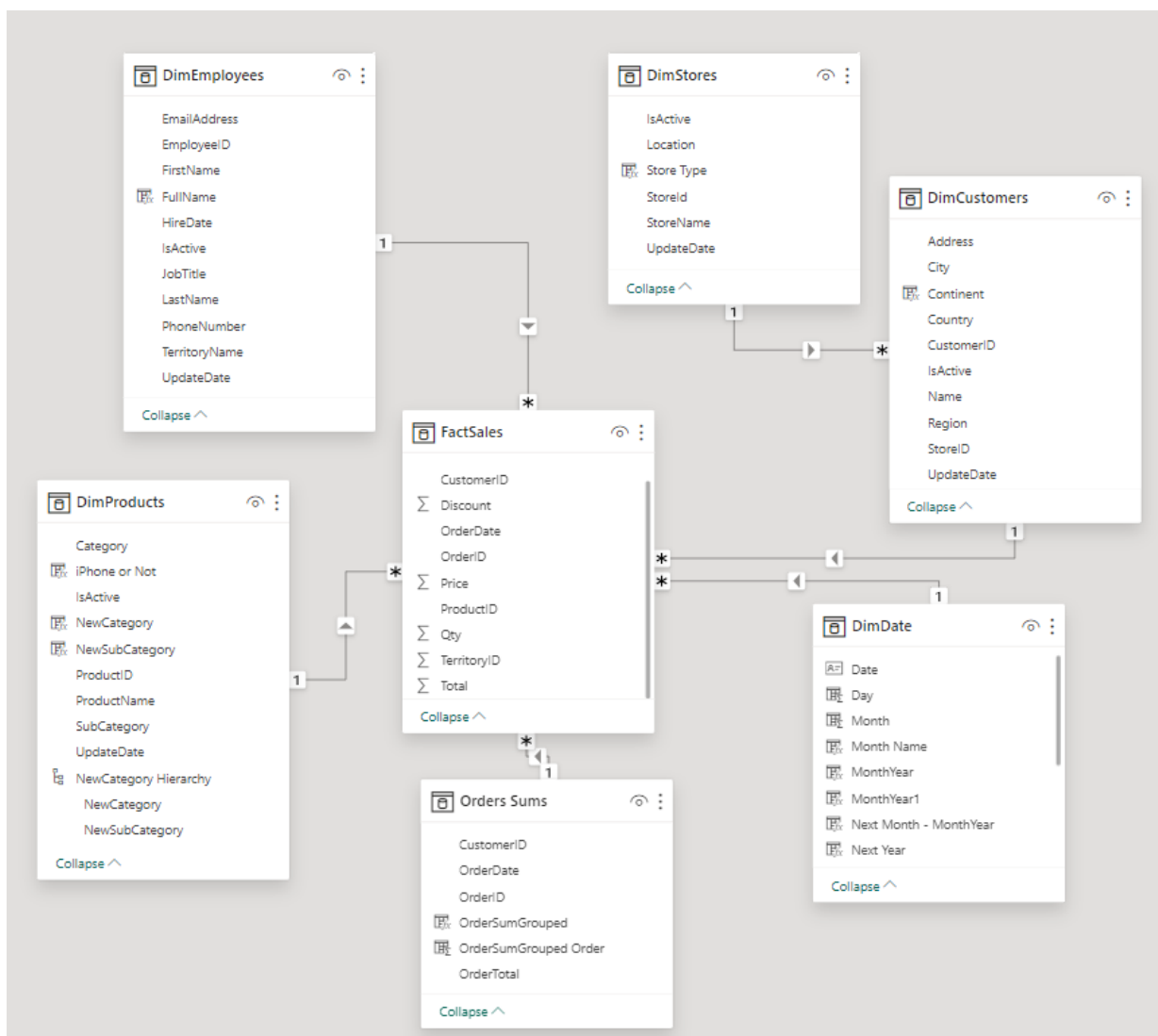




#### 4.4. Visualization in Power BI:

4.4.1. 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 5 Dimension tables (not including the product history table). To these tables, a DimDate table was added, together with 3 more tables:

- Orders Sums – an aggregated table calculating the sum for each order. This table was used in a slicer slicing orders by sum groups (e.g. \$0-\$500, \$500-\$1,000 etc.). This table was related to FactSales using OrderID.
- Yearly Goals – a table specifying the sales goals for each year (relying on previous year's sales) This table was not connected to the model.
- Monthly Goals - a table specifying the sales goals for each month (relying on sales of the same month on previous year).



4.4.2. To create the visuals, the following measures were created in DAX:

##### Totals:

- Total Customers = IF(ISBLANK(DISTINCTCOUNT(FactSales[CustomerID])), 0, DISTINCTCOUNT(FactSales[CustomerID]))



- Total Orders = IF(ISBLANK(DISTINCTCOUNT(FactSales[OrderID])), 0, DISTINCTCOUNT(FactSales[OrderID]))
- Total Revenue = IF(ISBLANK(SUM(FactSales[Total])), 0, SUM(FactSales[Total]))
- Total Units = IF(ISBLANK(SUM(FactSales[Qty])), 0, SUM(FactSales[Qty]))

#### Previous Year/Month:

- Previous Year Orders = CALCULATE([Total Orders], SAMEPERIODLASTYEAR(DimDate[Date]))
- Previous Year Sales = CALCULATE([Total Revenue], SAMEPERIODLASTYEAR(DimDate[Date]))
- Previous Year Units = CALCULATE([Total Units], SAMEPERIODLASTYEAR(DimDate[Date]))
- Previous Month Sales = IF(ISBLANK(CALCULATE([Total Revenue], PREVIOUSMONTH(DimDate[Date]))), BLANK(), CALCULATE([Total Revenue], PREVIOUSMONTH(DimDate[Date])))

#### Change From Last Year/Month:

- YoY Growth = IFERROR([Total Revenue]/[Previous Year Sales] - 1, "NULL")
- YoY Growth Orders = IFERROR([Total Orders]/[Previous Year Orders] - 1, "NULL")
- YoY Growth Units = IFERROR([Total Units]/[Previous Year Units] - 1, "NULL")
- MoM Growth = IFERROR([Total Revenue]/[Previous Month Sales] - 1, BLANK())

#### YTD and MTD:

- YTD Customers = TOTALYTD([Total Customers], DimDate[Date])
- YTD Sales = TOTALYTD([Total Revenue], DimDate[Date])
- MTD Customers = TOTALMTD([Total Customers], DimDate[Date])
- MTD Sales = TOTALMTD([Total Revenue], DimDate[Date])

#### Averages:

- Average Monthly Revenue = IF(ISERROR(AVERAGEX(VALUES(DimDate[MonthYear]), [Total Revenue])), 0, AVERAGEX(VALUES(DimDate[MonthYear]), [Total Revenue]))
- Average Orders Per Customer = IF(ISERROR([Total Orders]/[Total Customers]), 0, [Total Orders]/[Total Customers])
- Average Sales Amount Per Customer = IF(ISERROR([Total Revenue] / DISTINCTCOUNT(DimCustomers[CustomerID])), 0, [Total Revenue] / DISTINCTCOUNT(DimCustomers[CustomerID]))
- Average Sales Amount Per Customer Previous Year = IF(ISERROR([Previous Year Sales] / DISTINCTCOUNT(DimCustomers[CustomerID])), 0, [Previous Year Sales] / DISTINCTCOUNT(DimCustomers[CustomerID]))
- Average Sales Amount Per Order = IF(ISERROR([Total Revenue] / DISTINCTCOUNT(FactSales[OrderID])), 0, [Total Revenue] / DISTINCTCOUNT(FactSales[OrderID]))
- Average Units Per Customer = IF(ISERROR([Total Units]/[Total Customers]), 0, [Total Units]/[Total Customers])
- Average Units Per Order = IF(ISERROR([Total Units] / DISTINCTCOUNT(FactSales[OrderID])), 0, [Total Units] / DISTINCTCOUNT(FactSales[OrderID]))



### New Customers:

- New Customers =  

```

VAR currentCustomers = VALUES(FactSales[CustomerID])
VAR currentDate = MIN(DimDate[Date])
VAR pastCustomers =
CALCULATETABLE(VALUES(FactSales[CustomerID]),ALL(DimDate[Date]),Dim
Date[Date]<currentDate)
VAR newCustomers = EXCEPT(currentCustomers,pastCustomers)
RETURN
IF(ISBLANK(COUNTROWS(newCustomers)), 0,
COUNTROWS(newCustomers))

```
- New Customers % =  $\text{IF}(\text{ISERROR}([\text{New Customers}]/[\text{Total Customers}]), 0, [\text{New Customers}]/[\text{Total Customers}])$

### Measures for visuals including monthly and yearly goals:

- Max Month =  $\text{FORMAT}(\text{MAXX}(\text{DimDate}, \text{DimDate}[\text{Date}]), "mmm-yyyy")$
- Max Year =  $\text{Year}(\text{MAXX}(\text{DimDate}, \text{DimDate}[\text{Date}]))$
- Monthly Goal =  $\text{CALCULATE}(\text{SUM}('Monthly\ Goals'[\text{Goal}]), \text{FILTER}('Monthly\ Goals', 'Monthly\ Goals'[\text{Next Month} - \text{MonthYear}] = [\text{Max Month}]))$
- Monthly Goal Max For Gauge =  $[\text{Monthly Goal}] * 1.2$
- Yearly Goal =  $\text{CALCULATE}(\text{SUM}('Yearly\ Goals'[\text{Goal}]), \text{FILTER}('Yearly\ Goals', 'Yearly\ Goals'[\text{Next Year}] = [\text{Max Year}]))$
- Yearly Goal Max For Gauge =  $[\text{Yearly Goal}] * 1.2$
- Count of Customers with that Many Orders Grouped =  

```

VAR _CustomerOrders =
SUMMARIZE(
FactSales,
FactSales[CustomerID],
"How Many Orders",
DISTINCTCOUNT(FactSales[OrderID])
)
RETURN
COUNTROWS(
FILTER(
_CustomerOrders,
[How Many Orders] >= SELECTEDVALUE(Segment[Bottom]) && [How Many
Orders] <= SELECTEDVALUE(Segment[Top])
))

```

#### 4.4.3. Reports:

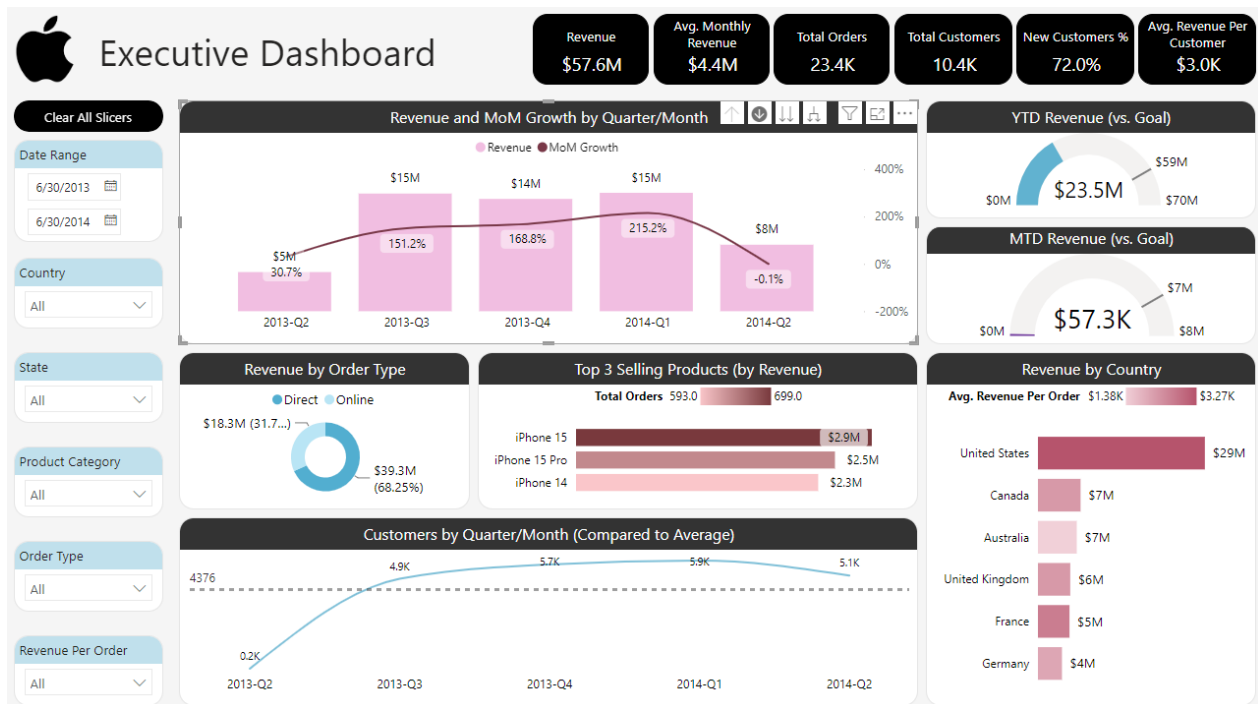
The project includes 3 reports: Executive Dashboard, Sales Analysis, Customer Analysis. All the three reports include only data from the last year (i.e. 365 earlier from the max order date)





#### 4.4.3.1. 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:

- Revenue
- Average Monthly Revenue
- Total Orders
- Total Customers
- New Customers %
- Average Revenue Per Customer

#### Graphs:

- Revenue and MoM Growth by Quarter/Month (Drill Mode)
- YTD Revenue compared to yearly goal – goal and max value on the gauge, change according to the time range.
- Month Revenue compared to monthly goal - goal and max value on the gauge, change according to the time range.
- Revenue by Order Type
- Top 3 Selling Products by Revenue
- Customers by Quarter/Month (Drill Mode) Compared to Average
- Revenue by Country – Color by Average Revenue per Order

#### Slicers:

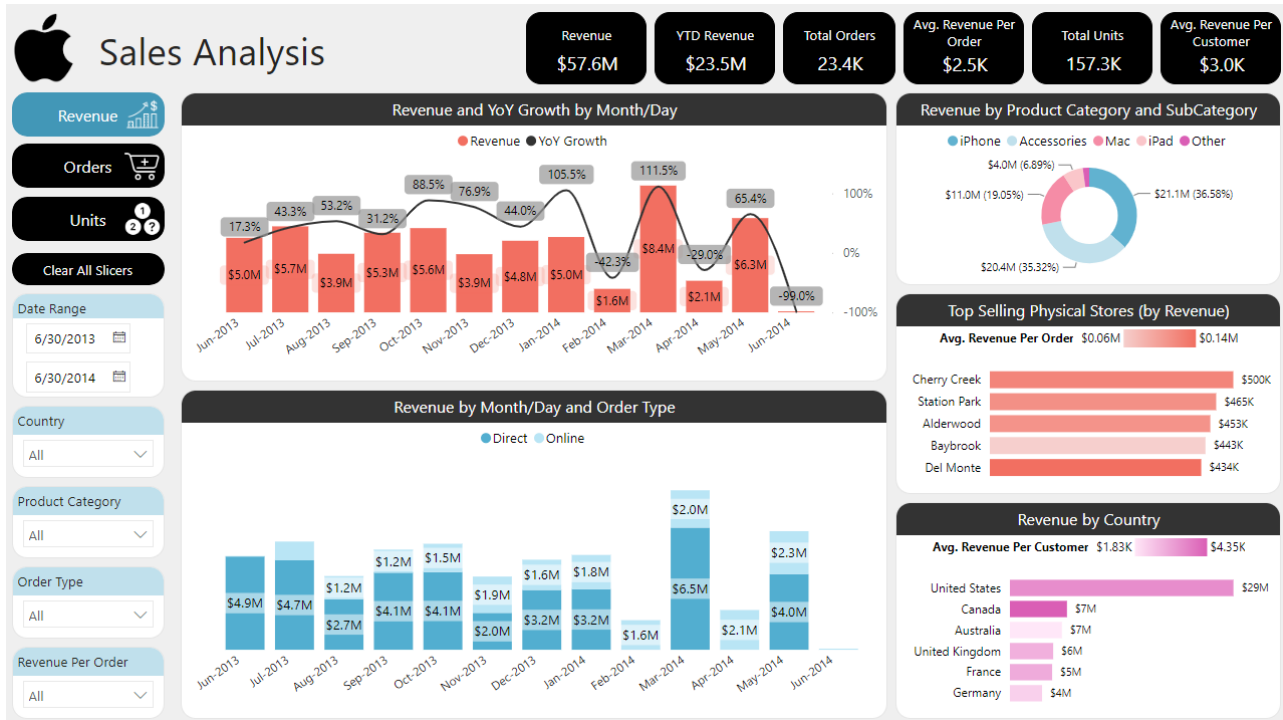
- Date Range
- Country
- State
- Product Category



- Order Type
- Revenue Per Order (grouped)

#### 4.4.3.2. 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
- Average Revenue per Order
- Total Units
- Average Revenue Per Customer

Slicers (same for all 3 states):

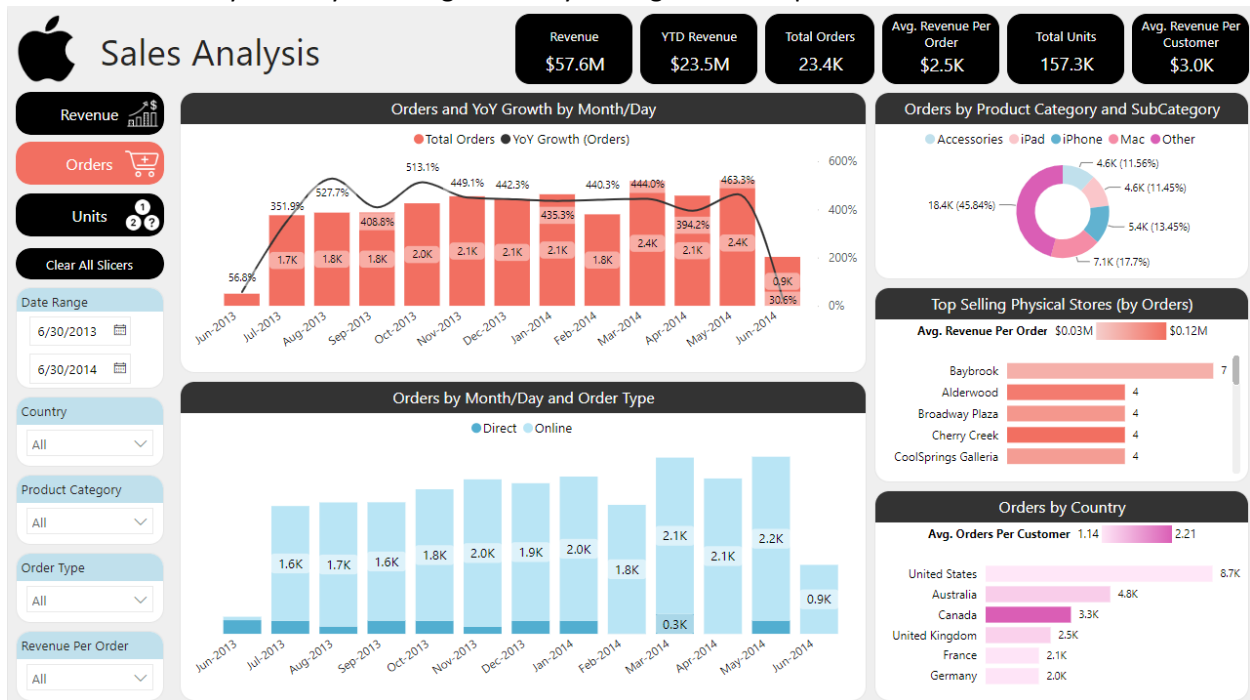
- Date Range
- Country
- Product Category
- Order Type
- Revenue Per Order (grouped)

Graphs (revenue state):

- Revenue and YoY Growth by Month/Day (Drill Mode)
- Revenue by Month/Day (Drill Mode) and by Order Type
- Revenue by Product Category/SubCategory (Drill Mode)

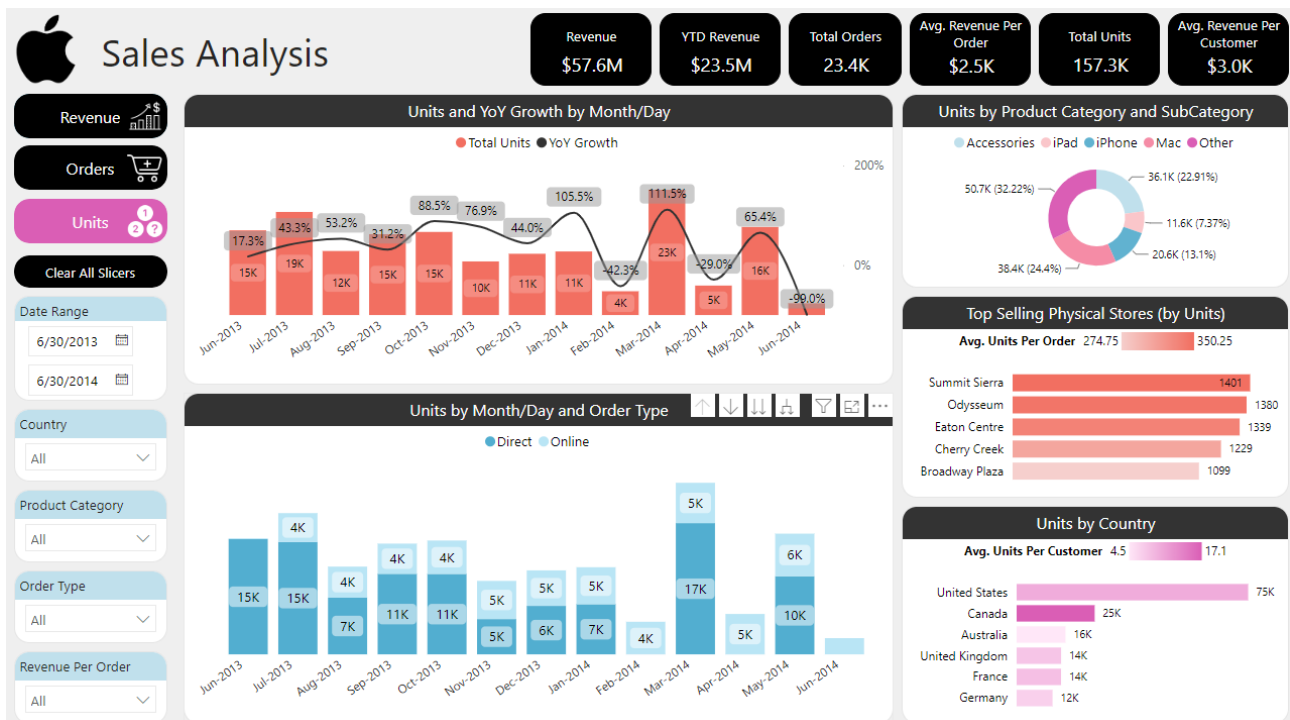


- Top 5 Selling Stores (by Revenue)
- Revenue by Country – Color gradient by Average Revenue per Customer



Graphs (orders state):

- Orders and YoY Growth by Month/Day (Drill Mode)
- Orders by Month/Day (Drill Mode) and by Order Type
- Orders by Product Category/SubCategory (Drill Mode)
- Top 5 Selling Stores (by Orders)
- Revenue by Country – Color gradient by Average Orders per Customer



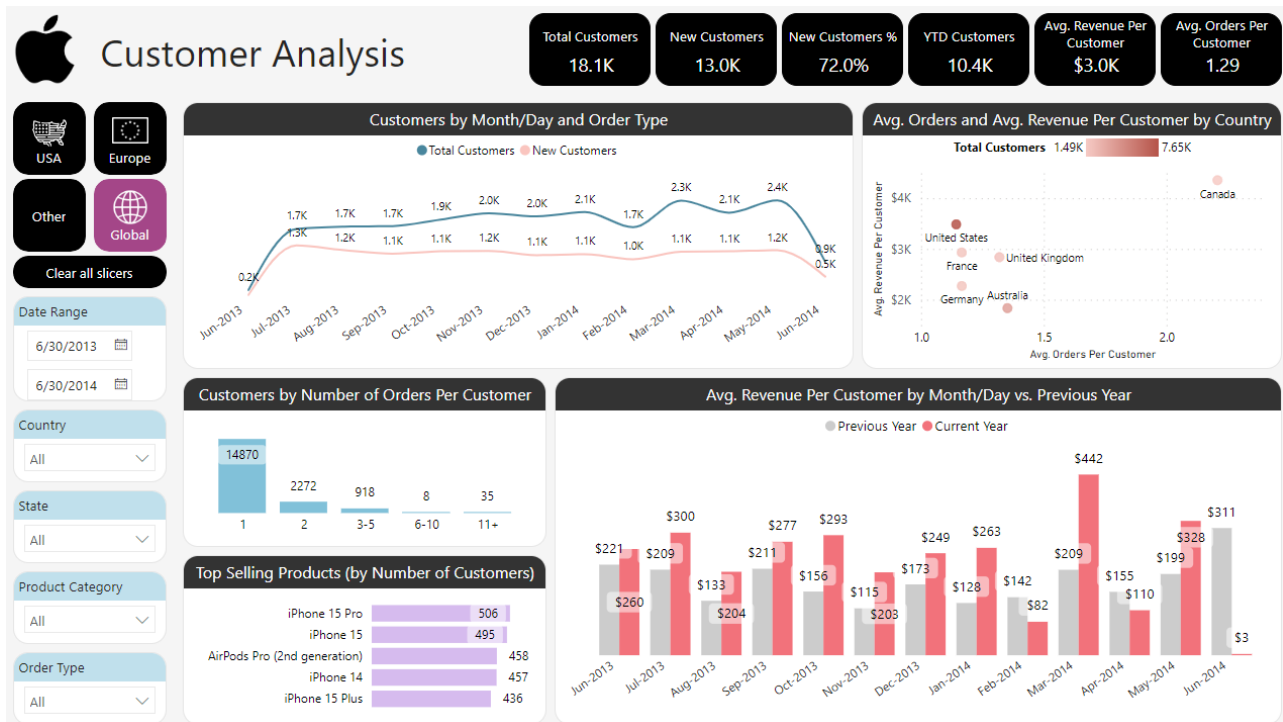


Graphs (units state):

- Units and YoY Growth by Month/Day (Drill Mode)
- Units by Month/Day (Drill Mode) and by Order Type
- Units by Product Category/SubCategory (Drill Mode)
- Top 5 Selling Stores (by Units)
- Revenue by Country – Color gradient by Average Units per Customer

#### 4.4.3.3. Customer Analysis:

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



KPI Cards:

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

Graphs:

- Customers by Month/Day (Drill Mode)
- Average Orders per Customer and Average Revenue Per Customer by Country – Color gradient by Total Customers
- Customers by Number of Orders per Customer – This bar chart, specified for the chosen time range (can be changed using the Date Range slicer), how many customers ordered once, twice, 3-5 time, 6-10 time, and more than 10.
- Top 5 Selling Products by number of customers.

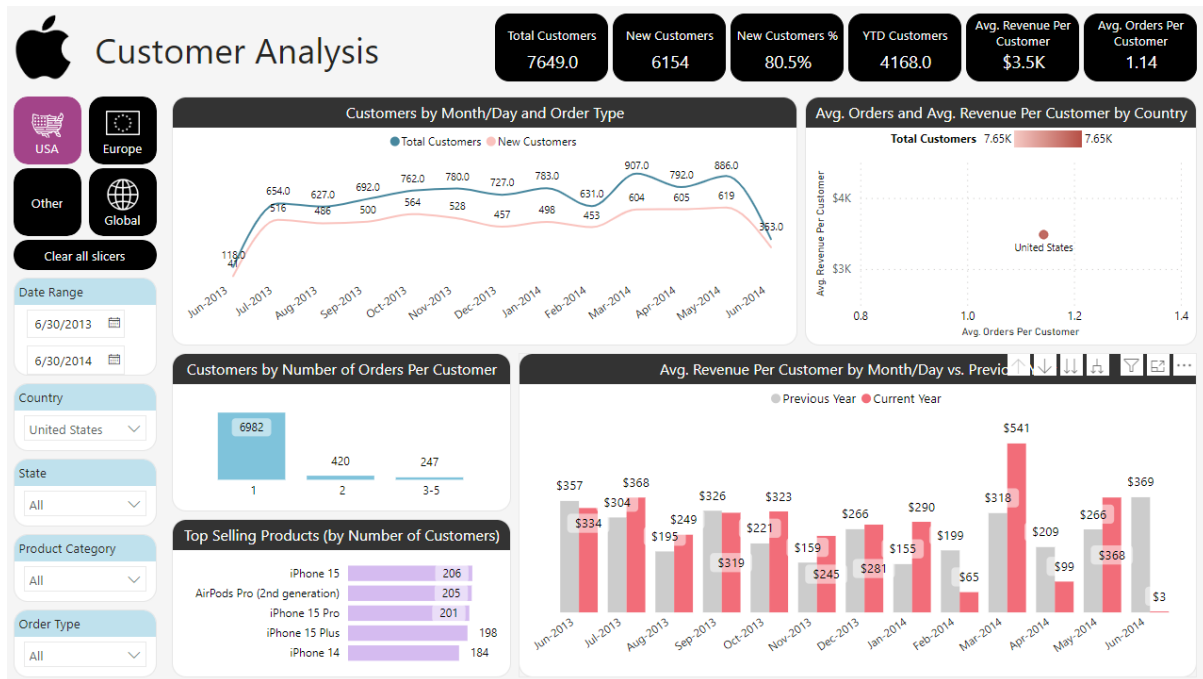


- Average Revenue per Customer by Month/Day (Drill Mode) vs. Previous Year

Slicers:

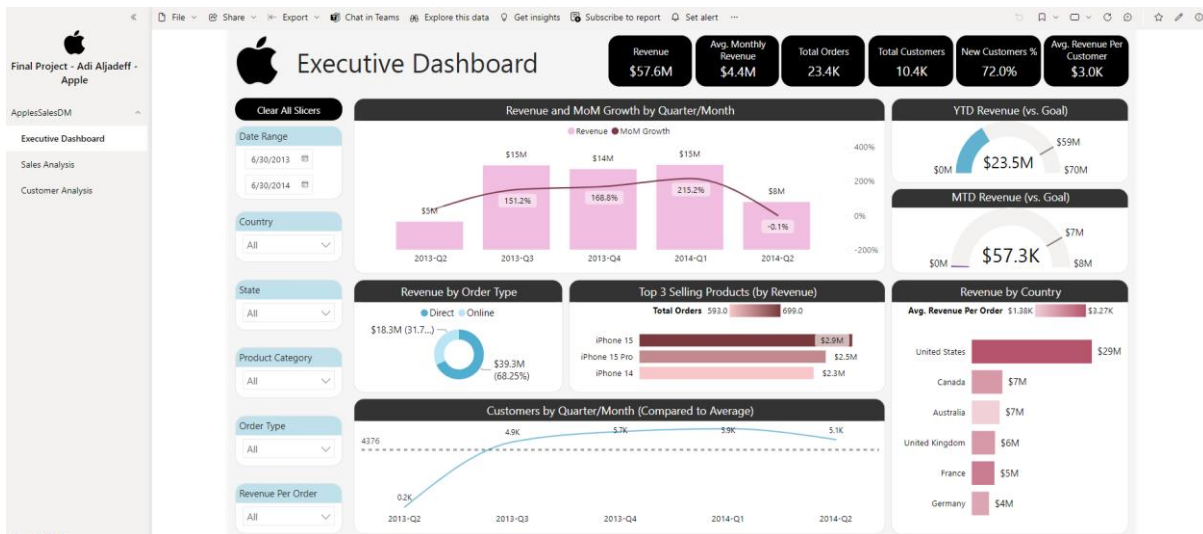
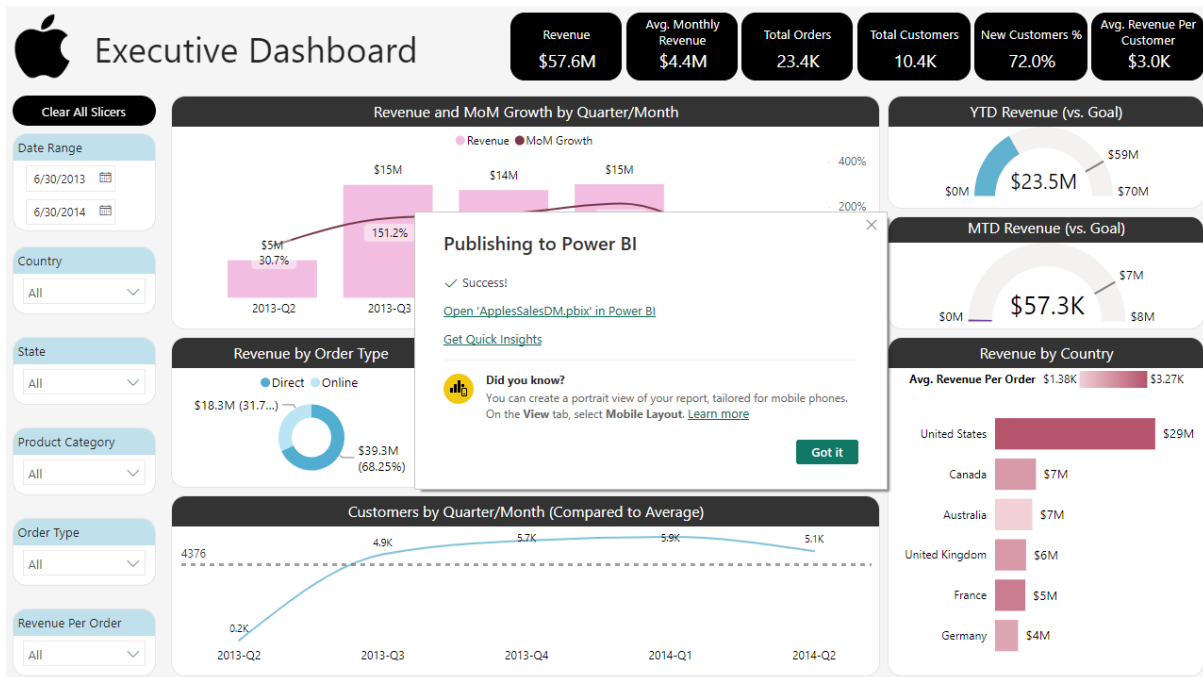
- Date Range
- Country
- State
- Product Category
- Order Type

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 Canada), and global – presenting data from all countries.





4.4.4. After creating the reports in Power BI Desktop, they were published to Power BI Service, and an app was created - [Final Project – Adi Aljadeff - Apple](#).





4.4.5. 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>Running on DESKTOP-M24IECD</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

[Add another time](#)