Stradivarius

BI System Specification Documentation

Bar Swisa

Version 5.0

12/2/2024

Contents

Document versions - change tracking:

- 24/01/2024 – Initial content – V1.0
- 29/01/2024 – Functional characterization update, ETL processes chapter and DM tables – V2.0
- 30/01/2024 – The final ERD update + PBI visualization indicators chapter – V3.0
- 05/02/2024 - Visualization chapter – V4.0
- 07/02/2024 – The Power BI Service process update – distribution, scheduling, application – V5.0

1. General

1.1. Project objectives

The purpose of the project is to provide a Full-Scale BI Solution Creation from PriorityERP Database for Stradivarius, a Spanish company that is a well-known international Fashion retailer.

This project aims to establish a comprehensive BI solution leveraging data from the PriorityERP system for Stradivarius. The solution will encompass summarized data tables, with a focus on sales data, alongside customer information, employee records, product details, stores, dates, and more. The BI solution includes dashboards and reports to assist management, department heads, and sales managers in gaining insights into customer preferences, behaviors, and loyalty. The goal is to optimize marketing campaigns and reduce costs through targeted strategies.

1.2. Project Contents

In this project, we will build a Data Mart that will contain information about sales data.

1. Data Cleaning and Preparation: Prior to analysis, we will need to perform thorough data cleaning and preparation to ensure their quality and consistency.
2. Main summary tables to be built for the company's needs

   - **FactSales –** Information about all the orders, which product in which quantity. Data loading process for this table will be incremental.
   - **Dim_Products** – Information about the products divided by categories and subcategories.
   - **Dim_Customers** – Information about the company's customers
   - **Dim_Employees** – Information about the company's employees.
   - **Dim_Stores** – Information about the company's stores.

   History Management Table

   - **Transfertable** - Information about all the updates of the tables.
   - **Dim_Products_History** - Information about the historical record of changes to product information. The product history table will be included to track changes in products over time using Slowly Changing Dimensions (SCD) Type 4.

The ERD of the tables shown in the attached link: ERD

Source To Target document in this link: S2T

3. The project will contain measures that will contribute to the achievement of the project's goal:

• Dashboard:
In the dashboard, you'll find detailed insights about the company. This information is tailored for senior management, providing a comprehensive overview of the company's performances. The dashboard highlights key metrics, including Resell Percentage and average basket size, allowing us to assess performance effectively. The dashboard will display the following information:

- General KPI data.
- Resell Percentage by Continents.
- AVG Basket Size by Country.
- Physical Stores VS Online Store.
- YTD Sales VS LY Sales.
- YTD Units VS LY Units.
- YTD Orders VS LY Orders.
- Top 5 Selling Products.
- Top 5 Employees by sales.

• Sales Department Analysis:

The Sales Department will conduct a thorough analysis of sales data, seller performance, and revenue to evaluate market trends, refine sales strategies, and make year-on-year comparisons. The primary focus will be on identifying top-performing products and recognizing the best employees based on sales performance. The report will display the following information:

- General KPI data
- Employees sales performance - the total sales YTD and comparison to the same period last year, along with a percentage change, KPI, and sparkline for sales amount, units amount, and orders amount.
- The top 10 best-selling products and their quantities.
- The top 5 best-employees by sales.

• Customer Sales Analysis:

The Customer Sales Analysis report examines vital data and trends to provide insights into our business performance, with the ultimate goal of improving sales strategies in each country. Additionally, the report includes a comparative analysis between physical store sales and online sales. The report will display the following information:

- General KPI data
- Top State Details
- Physical Stores VS Online Store By Country
- Total Sales by Country
- Monthly percentage Change in Units Sold.
- Monthly percentage Change in Total Sales
- Monthly percentage Change in Order APRU (Average Revenue Per User)
- Monthly percentage Change in Order Count

2. Gantt

❖ Gantt link

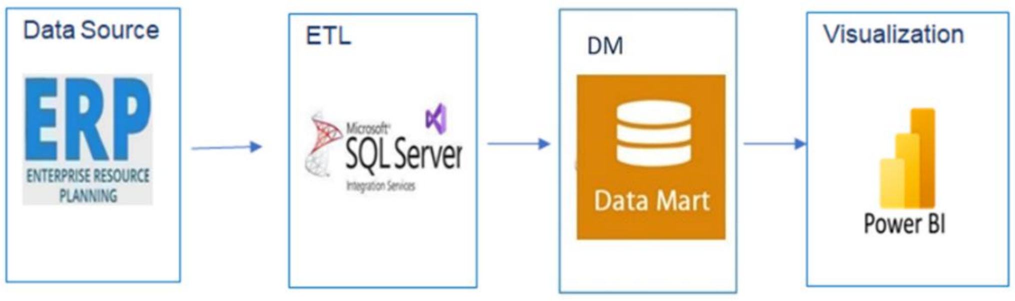
3. Technical Specification

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 through the definition of JOBS in SSMS
- Power BI: Creating reports and dashboards using Power BI

3.2. Solution Architecture

HLD:



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.

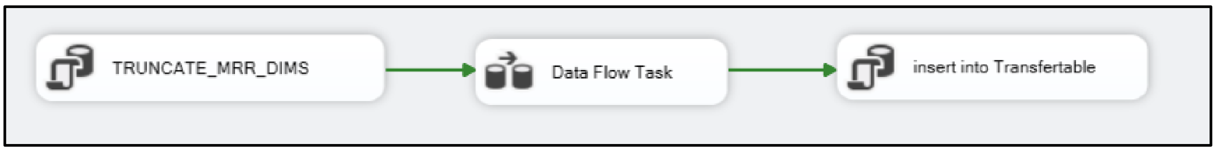
4. Functional Specification

4.1. ETL processes

1. MRR_Dim package:

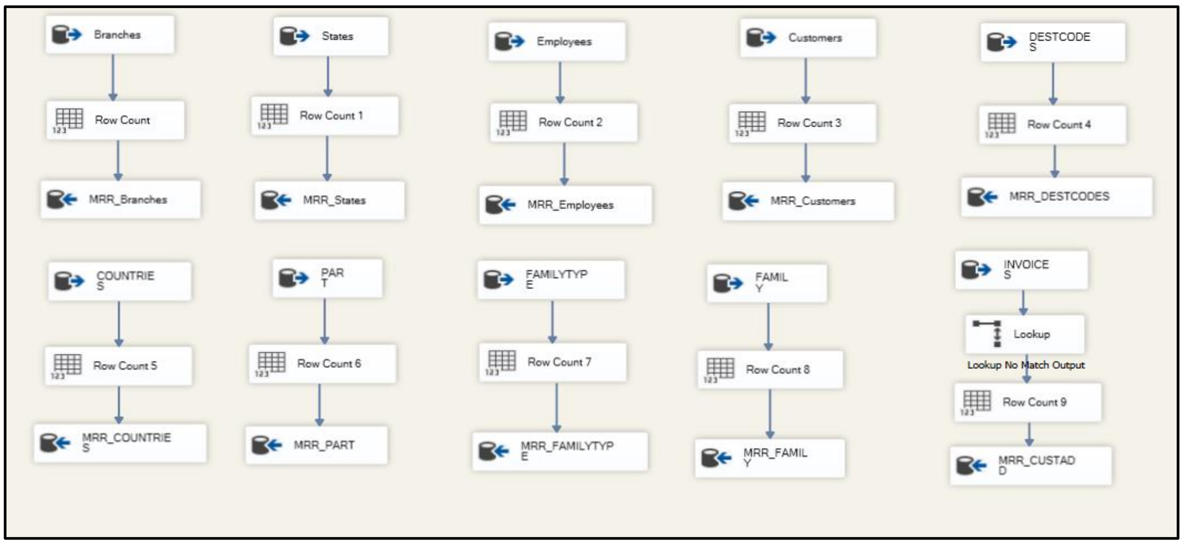
Into the Execute SQL Task - truncate_mrr:

This stored procedure, named truncate_mrr_tables, is designed to truncate (delete all rows) from multiple tables.



Finally, we insert the values to the transfer table. The TransferTable serves as a comprehensive log, meticulously capturing every update and insertion step as data moves through the stages from the database (DB) to the Data Mart.

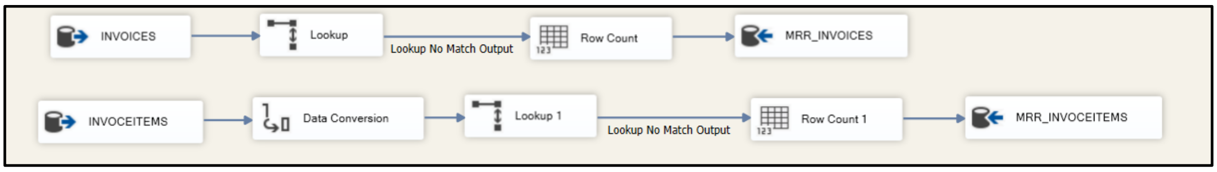
In the Data Flow- dim_mrr - We will transfer information from OLTP tables to MRR tables



Initially, we migrated all customer data to our "DIM" (Dimension) table. Subsequently, we established a business rule stipulating that only customers who made purchases within the last three years would be considered. This rule was implemented to enable the identification and display of customers who have not engaged in any transactions within the last three years, marking them as inactive. This approach ensures that we are working with the most current and updated data, allowing us to make informed decisions based on the latest trends and developments.
We use a lookup in the Invoices table because it serves as our transaction table, utilized in both the MRR dimension package and MRR fact package.

2. MRR_Fact package:

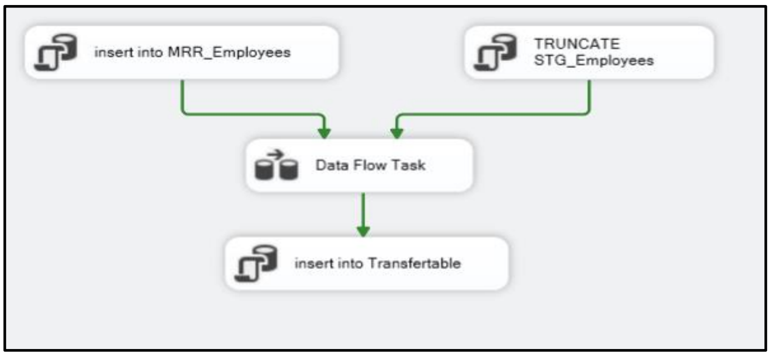
Into the Lookup - Invoices & InvoiceItems:



We used the LOOKUP transformation to load only the new rows to the MRR table. The target was to identify the gap (only the new rows from the operational DB that don't exist in the fact sales)

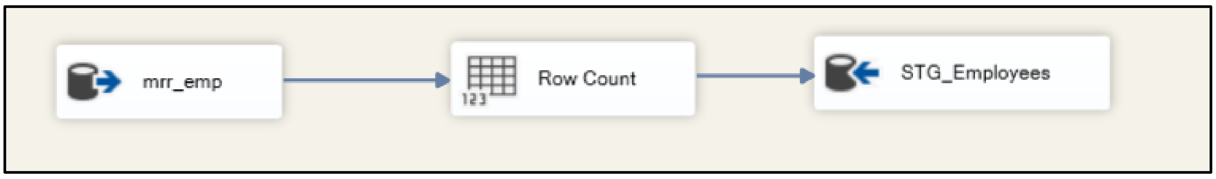
3. STG_Dim_Employees package:

In the Control Flow:



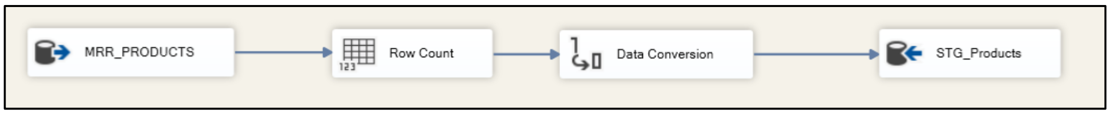
We updated the MRR_Employees table manually with a SQL command that adds a new employee that is associated with the Employee ID "77777". We want to make sure that orders placed online are properly associated with the employee we manually added to the MRR_Employee table. We want to maintain the connection between this employee and the online store that contains the order information.

In the Data Flow:



We collected information about employees from the employees' table. The selected columns include details such as Emp_ID, First_Name, Last_Name, Job_Title, Hire_Date, Phone_Number, Email_Address etc. Additionally, we have selectively retrieved specific columns from the 'MRR_Employees' table, focusing on entries where the 'skill' column contains the term 'sales.' Finally, we will insert the information into the stg_Employees table.

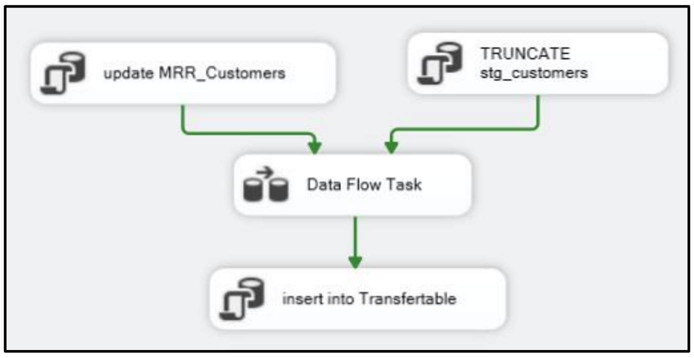
4. STG_Products package



We will collect information about products from three tables: MRR_Part, MRR_Familly and MRR_FamillyType. The selected columns include details: ProductID, Product_Name, Category_Name and Sub_Category_Name.In the Data Conversion phase, we will meticulously refine data types that necessitate modification. Finally, we will insert all the information into the stg_products table.

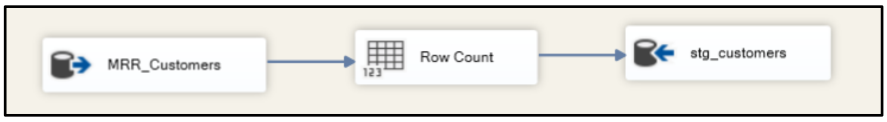
5. STG CUSTOMERS package

In the Control Flow:



We updated the MRR_Customers table manually with SQL command, that every customer who has an order in the invoices table, and does not associate with any store id, will receive the store id “99999” as a sign for “Online Store”. We made this in order to maintain the association between the orders and the stores through the customers.

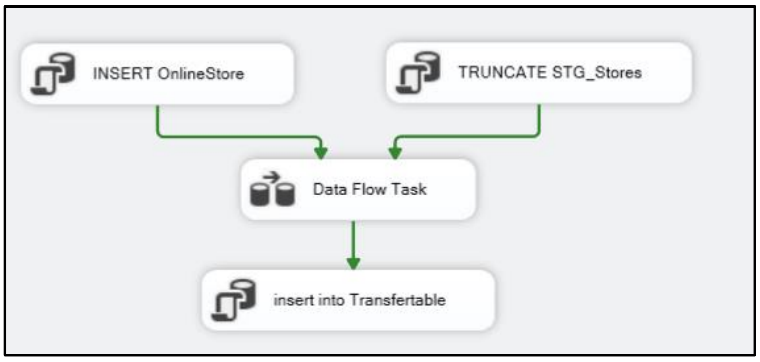
In the Data Flow:



We collected information about Customers from the Customers, Destcode, State and Country tables. The selected columns include details such as Customer ID, Name, store ID, Address, City, State, Country.

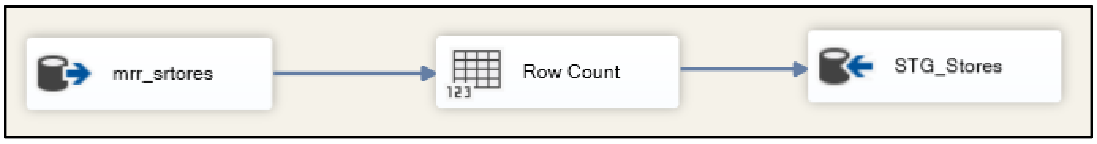
6. STG STORES

In the Control Flow:



We updated the MRR_Branches table manually with SQL command with a new Online Store that is associated with the Store ID “99999”.

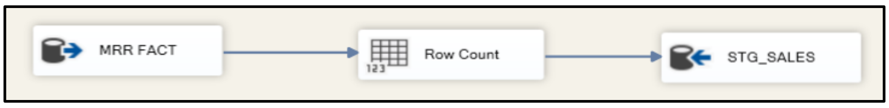
In the Data Flow:



We will collect information about stores from five tables: MRR_Branches,MRR_Customers, MRR_CustAdd , MRR_Destcodes and MRR_States. The selected columns include details such as StoreId, StoreName and State. Additionally, we add to our join query a statement that if the branch ID is 99999, it's labeled as 'OnlineStore'; otherwise, it takes the actual state name from the 'MRR_States' table based on the relationship between the tables established in the LEFT JOIN operations.

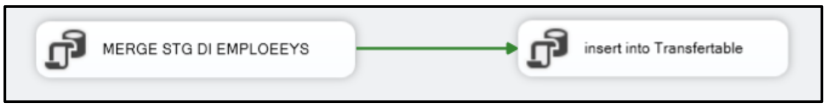
7. STG SALES

In the Data Flow:



We will collect sales order information, including order details and product specific details, connecting the two tables MRR_INVOICES and MRR_INVOCEITEMS. Based on IV.
The selected columns include details such as OrderID, OrderDate, CustomerID, EmpID, ZoneID, ProductID, Qty, Price and Discount.

8. DM EMPLOYEES

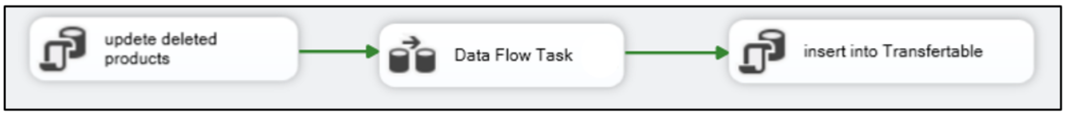


Into the Execute SQL Task – MERGE:

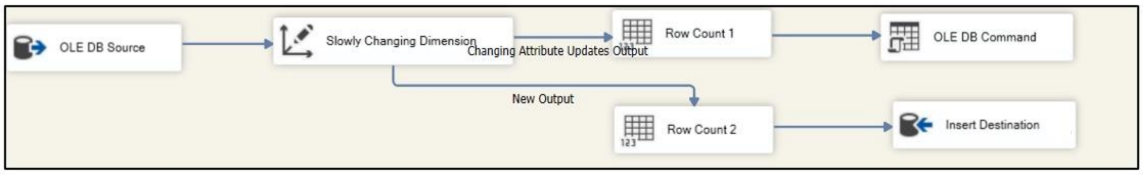
Using to synchronize data between the Dim_Employees and stg_employees tables based on EMP_ID. It performs the following actions:

- Insert: If there is a record in stg_employees that does not match Dim_Employees, insert a new record.
- Update: If there is a match and certain columns have changed, update the corresponding columns in Dim_Employees with values from stg_employees.
- Update (IsActive): If there is a record in Dim_Employees that does not exist in stg_Employees, set IsActive to 0 in Dim_Employees.

9. DM PRODUCTS

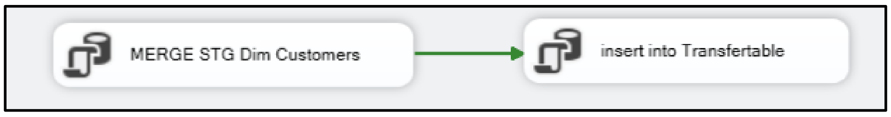


The query updates records in the dim_Products table in a database. The general description of the query is as follows:
The query iterates over each record in the dim_Products table and updates the columns UpdateDate and ISactive. The update conditions are complex and include:
The ProductID column should match the values returned by a subquery that retrieves Dim_ID from the join of the PART table in the PriorityERP database with the dim_Products table.
The ISactive column should be different from 0.
The record must be identified based on the criterion that its ProductID is not found in the PART table in the PriorityERP database.
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_Products' table accurately reflects changes in external data, including the handling of deleted lines.



In the Slowly Changing Dimensions (SCD), the approach involves distinguishing between new lines and updated lines using the 'ProductID.' In the process of updating using the OLE DB Command, we focus on the changed fields. New lines, identified by a unique 'ProductID,' are handled separately to ensure proper management of evolving data. Meanwhile, for updated lines, the OLE DB Command's update operation selectively targets and updates only the fields that have changed.

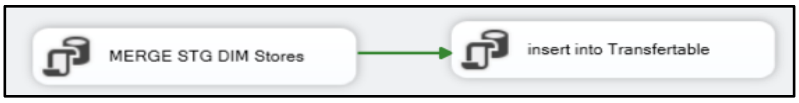
10. DM CUSTOMERS



Using to synchronize data between the Dim_Customers and stg_Customers tables based on CustomersID. It performs the following actions:

- Insert: If there is a record in stg_Customers that does not match Dim_Customers, insert a new record.
- Update: If there is a match and certain columns have changed, update the corresponding columns in Dim_Customers with values from stg_Customers.
- Update (IsActive): If there is a record in Dim_Customers that does not exist in stg_Customers, set IsActive to 0 in Dim_Customers.

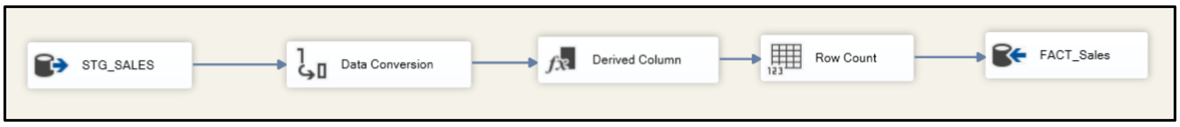
11. DM STORES



Using to synchronize data between the Dim_ Stores and stg_ Stores tables based on StoreID. It performs the following actions:

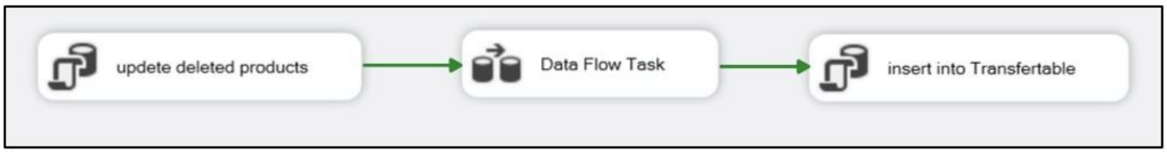
- Insert: If there is a record in stg_ Stores that does not match Dim_ Stores, insert a new record.
- Update: If there is a match and certain columns have changed, update the corresponding columns in Dim_ Stores with values from stg_Stores.
- Update (IsActive): If there is a record in Dim_ Stores that does not exist in stg_ Stores, set IsActive to 0 in Dim_ Stores.

12. DM FACT SALES



In the OLE DB Source - STG_SALES, we will channel comprehensive data into the FACT_Sales table. Subsequently, in the Data Conversion phase, we will meticulously refine data types that necessitate modification. Following this, within the Derived Column transformation, we will dynamically compute the Total column using the specified formula: (Qty * Price * (1 - Discount)). Finally, the meticulously transformed and calculated data will be inserted into the DM_FACT_SALES table.

13. DM PRODUCTS HISTORY

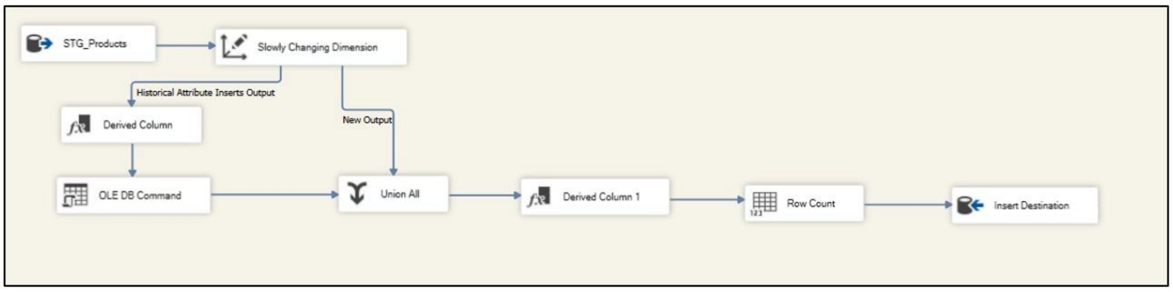


The query updates records in the Dim_Products_History table. Here's a brief description:

The UPDATE statement modifies the EndDate column in the Dim_Products_History table. It sets the EndDate to the current timestamp for records that meet the following conditions:

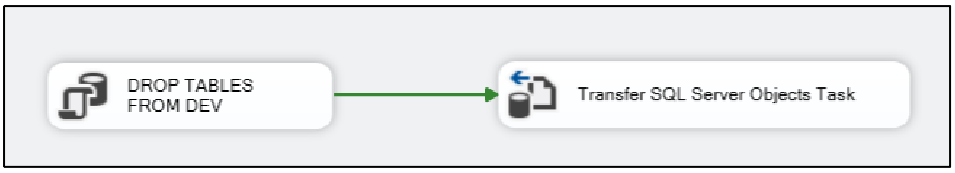
- The ProductID is not found in the PART table of the PriorityERP database.
- The EndDate is currently NULL.

In summary, the query effectively marks records in the history table as ended for products that are no longer present in the PART table and have a NULL EndDate. This action helps differentiate between products actively selling in the company and those that are no longer part of the inventory history.



In the "dim_Products_History" package, when a record in the "dim_Products" table gets updated, we store its previous version in the "dim_Products_History" table. This historical table is structured like "dim_Products" 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's a valuable strategy for maintaining a historical record and ensuring transparency in understanding how product data has changed.

14. Dev – Drop & Create database



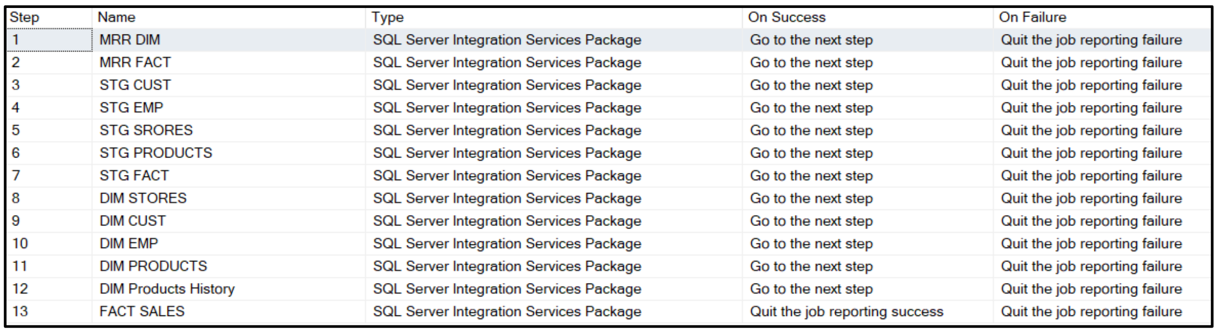
In this package, we've created a procedure in our development environment (using SSMS) that drops all the tables. Subsequently, we use an SQL task to execute this procedure. Afterward, we employ the Transfer SQL Server Objects Task to copy all tables from the production to the development environment. This entire process ensures that our development database is always up to date with what's in production.

4.2. Defining JOBS in SSIS

To facilitate the daily refresh and loading process, a deploy was executed from SSIS to SSMS. Subsequently, a job was created to run on a daily schedule at a fixed time. This job comprises 13 steps, each representing a distinct SSIS package responsible for handling various phases of the project.

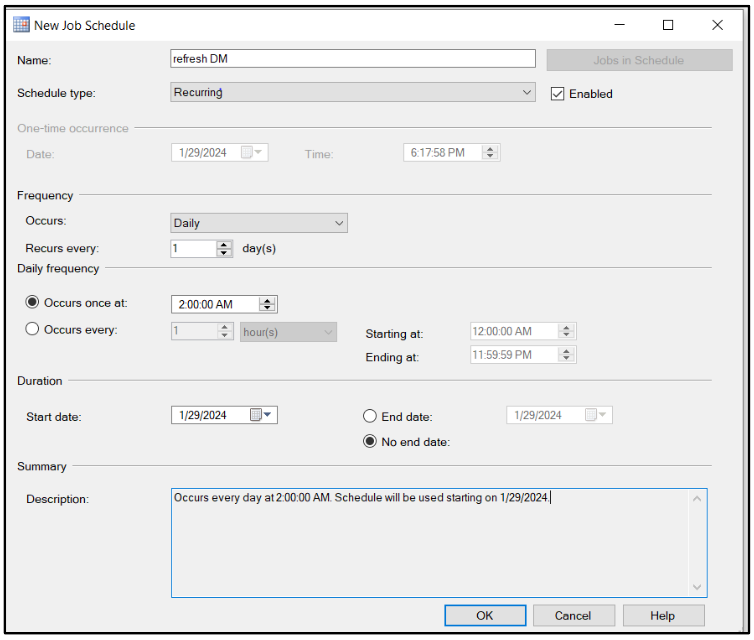
Error-handling rules have been defined to halt the process in case of an error, ensuring data integrity and reliability. Additionally, a success message is generated upon the successful completion of all 13 steps.

The entire process underwent testing by PQA, resulting in a successful validation without encountering any errors.



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 FACT	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
3	STG CUST	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
4	STG EMP	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
5	STG SRORES	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 FACT	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
8	DIM STORES	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
9	DIM CUST	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
10	DIM EMP	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
11	DIM PRODUCTS	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
12	DIM Products History	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
13	FACT SALES	SQL Server Integration Services Package	Quit the job reporting success	Quit the job reporting failure

It will automatically run each day at 02:00.



New Job Schedule

Name: refresh DM

Schedule type: Recurring [✓ Enabled]

One-time occurrence
Date: 1/29/2024 Time: 6:17:58 PM

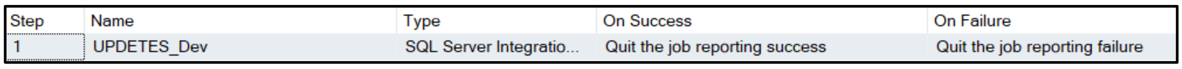
Frequency
Occurs: Daily
Recurs every: 1 day(s)

Daily frequency
[● Occurs once at:] 2:00:00 AM
Occurs every: 1 hour(s) Starting at: 12:00:00 AM Ending at: 11:59:59 PM

Duration
Start date: 1/29/2024 End date: 1/29/2024
[● No end date:]

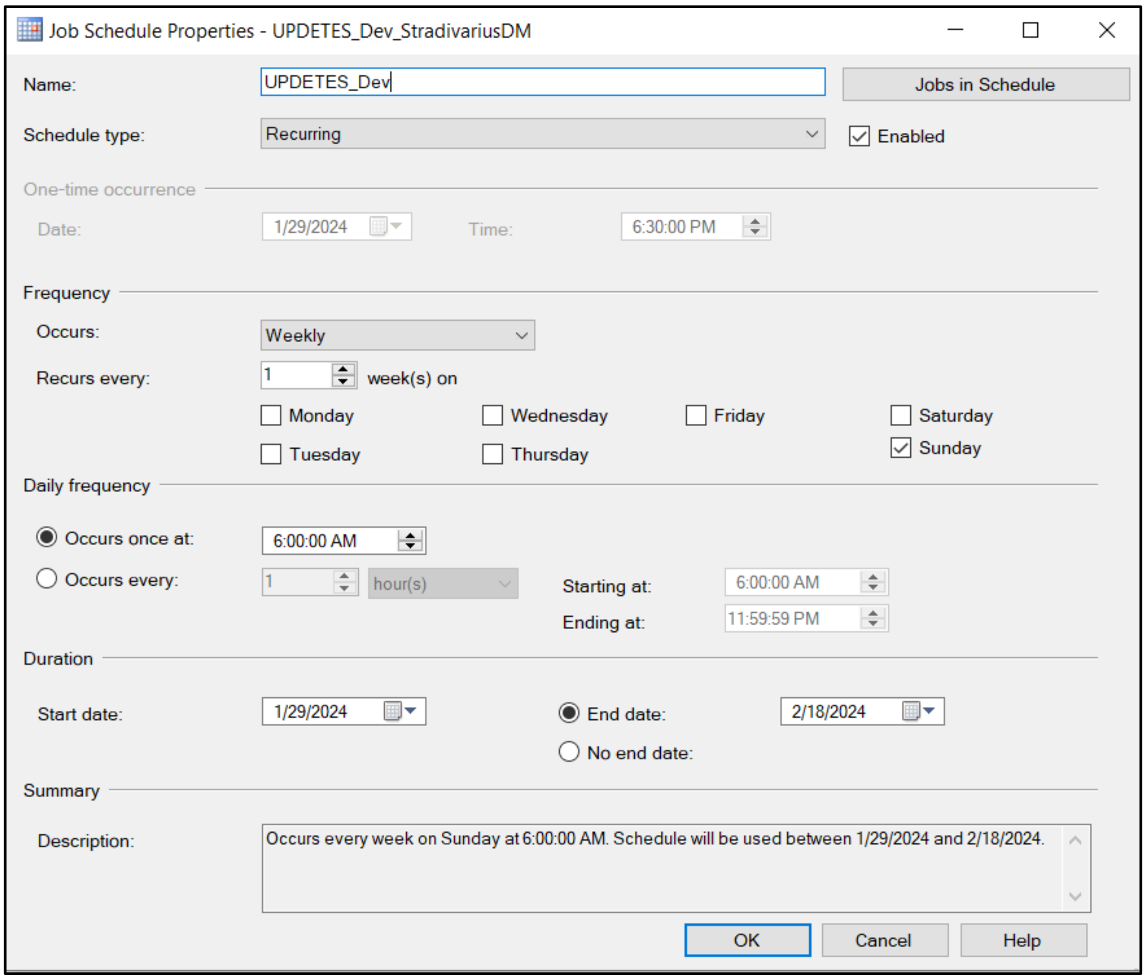
Summary
Description: Occurs every day at 2:00:00 AM. Schedule will be used starting on 1/29/2024.

In addition, we've implemented a recurring job that executes this update process weekly, ensuring our development environment stays current with production.



Step	Name	Type	On Success	On Failure
1	UPDETES_Dev	SQL Server Integratio...	Quit the job reporting success	Quit the job reporting failure

It will automatically run each week on Sunday 06:00 am



Job Schedule Properties - UPDETES_Dev_StradivariusDM

Name: UPDETES_Dev

Schedule type: Recurring   [✓ Enabled]

One-time occurrence

Date: 1/29/2024   Time: 6:30:00 PM

Frequency

Occurs: Weekly

Recurs every: 1 week(s) on

Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, [✓ Sunday]

Daily frequency

[✓ Occurs once at:] 6:00:00 AM

Occurs every: 1 hour(s)   Starting at: 6:00:00 AM   Ending at: 11:59:59 PM

Duration

Start date: 1/29/2024   [✓ End date:] 2/18/2024   No end date:

Summary

Description: Occurs every week on Sunday at 6:00:00 AM. Schedule will be used between 1/29/2024 and 2/18/2024.

5. 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 2 reports: customers and sales and also a dashboard. Those reports will help the senior management of the company to uncover trends, make informed decisions, and drive meaningful outcomes.

5.1. DAX Measures

In Power BI, I have created several measures.

1. **Total Sales** = CALCULATE(SUM(Fact_Sales[Total]))
2. **YTD Sales** = TOTALYTD([Total Sales], 'Dim_Date'[Date])
3. **LY Sales** = CALCULATE([Total Sales], SAMEPERIODLASTYEAR(datesytd(Dim_Date[Date])))
4. **Sales Difference** = [YTD Sales] - [LY Sales]
5. **Sales Difference %** = DIVIDE([YTD Sales] - [LY Sales], [LY Sales],0)
6. **Total Sales MoM %** = VAR __PREV_MONTH = CALCULATE([Total Sales], DATEADD('Dim_Date'[Date], -1, MONTH))RETURN DIVIDE([Total Sales] -__PREV_MONTH, __PREV_MONTH)
7. **Best Sales Day** = CALCULATE(MAXX(FILTER(VALUES(Dim_Date[Date]), YEAR(Dim_Date[Date]) = max(Dim_Date[YEAR])), [Total Sales]), ALLEXCEPT(Dim_Date, Dim_Date[date]))
8. **Lowest Sales Day** = CALCULATE(MINX(FILTER(Dim_Date,[Total Sales] <> BLANK() && YEAR(Dim_Date[Date]) = max(Dim_Date[YEAR])),[Total Sales]), ALLEXCEPT(Dim_Date, Dim_Date[Date]))
9. **Total Orders** = DISTINCTCOUNT(Fact_Sales[OrderID])
10. **YTD Orders** = TOTALYTD([Total Orders], 'Dim_Date'[Date])
11. **LY Orders** = CALCULATE([Total Orders], SAMEPERIODLASTYEAR(datesytd(Dim_Date[Date])))
12. **Order Difference** = [YTD Orders] - [LY Orders]
13. **Order Difference %** = DIVIDE([Order Difference], [LY Orders],0)
14. **Total Orders MoM %** = VAR __PREV_MONTH = CALCULATE([Total Orders], DATEADD('Dim_Date'[Date], -1, MONTH))RETURN DIVIDE([Total Orders] -__PREV_MONTH, __PREV_MONTH)
15. **Best Sales Day Orders** = CALCULATE(MAXX(FILTER(VALUES(Dim_Date[Date]), YEAR(Dim_Date[Date]) = max(Dim_Date[YEAR])), [Total Orders]), ALLEXCEPT(Dim_Date, Dim_Date[date]))
16. **Lowest Sales Day Orders** = CALCULATE(MINX(FILTER(Dim_Date,[Total Sales] <> BLANK() && YEAR(Dim_Date[Date]) = max(Dim_Date[YEAR])),[Total Orders]), ALLEXCEPT(Dim_Date, Dim_Date[Date]))
17. **Total Units** = CALCULATE (SUM(Fact_Sales[Quantity]))
18. **YTD Units** = TOTALYTD([Total Units], 'Dim_Date'[Date])
19. **LY Units** = CALCULATE([Total Units], SAMEPERIODLASTYEAR(datesytd(Dim_Date[Date])))
20. **Units Difference** = [YTD Units] - [LY Units]
21. **Units Difference %** = DIVIDE([YTD Units] - [LY Units], [LY Units],0)
22. **Total Units MoM %** = VAR __PREV_MONTH = CALCULATE([Total Units], DATEADD('Dim_Date'[Date], -1, MONTH))RETURN DIVIDE([Total Units] -__PREV_MONTH, __PREV_MONTH)
23. **Best Sales Day** Units = CALCULATE(MAXX(FILTER(VALUES(Dim_Date[Date]), YEAR(Dim_Date[Date]) = max(Dim_Date[YEAR])), [Total Units]), ALLEXCEPT(Dim_Date, Dim_Date[date]))

24. **Lowest Sales Day Units** = CALCULATE(MINX(FILTER(Dim_Date,[Total Sales] <> BLANK() && YEAR(Dim_Date[Date]) = max(Dim_Date[YEAR])),[Total Units]), ALLEXCEPT(Dim_Date, Dim_Date[Date]))
25. **Total Customer** = DISTINCTCOUNT(Fact_Sales[CustomerID])
26. **online Sales** = CALCULATE(SUM(Fact_Sales[Total]), Fact_Sales[EmpID]=77777)
27. **Physical Store Sales** = CALCULATE(SUM(Fact_Sales[Total]), Fact_Sales[EmpID]<>77777)
28. **Total average per OrderID** = AVERAGEX(KEEPFILTERS(VALUES('Fact_Sales'[OrderID])), CALCULATE(SUM('Fact_Sals'[Total])))
29. **Average Basket Size** = DIVIDE([Total Units], [Total Orders], 0)
30. **Average Revenue Per User** = DIVIDE([Total Sales], COUNTROWS(SUMMARIZE(Fact_Sales, Fact_Sales[CustomerID])))
31. **Average Revenue Per User MoM %** = VAR __PREV_MONTH = CALCULATE([Average Revenue Per User], DATEADD('Dim_Date'[Date], -1, MONTH))RETURN DIVIDE([Average Revenue Per User] - __PREV_MONTH, __PREV_MONTH)
32. **Number of Resold Units** = CALCULATE(SUM(Fact_Sales[Quantity]),FILTER(VALUES(Fact_Sales[OrderID]), CALCULATE(COUNTROWS(Fact_Sales), Fact_Sales[OrderID] = EARLIER(Fact_Sales[OrderID])) > 1))
33. **Resell Percentage** = DIVIDE([Number of Resold Units], [Total Units], 0)

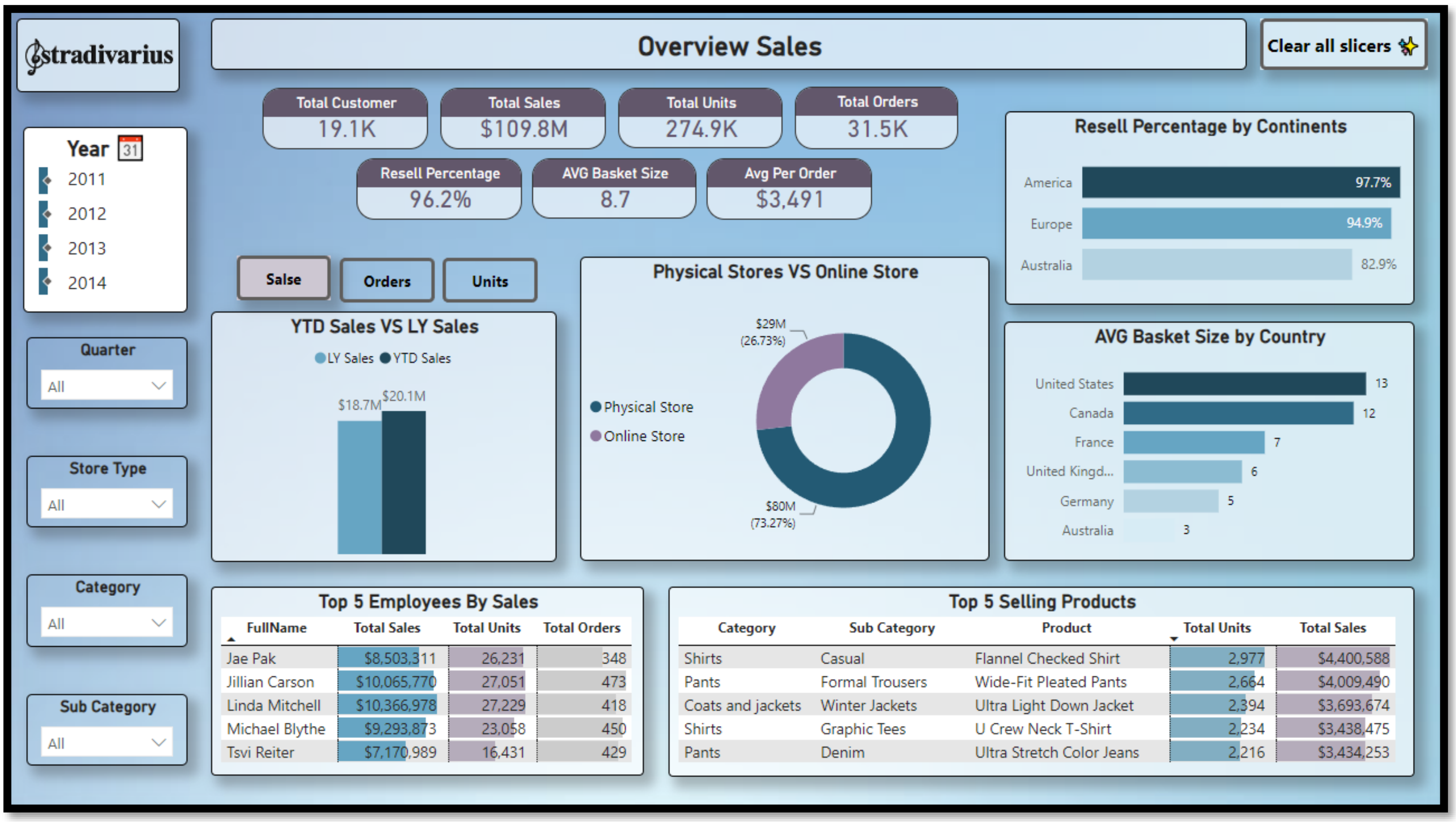
5.2. Overview dashboard

The dashboard presents a complete picture of the providers in the company, organized visually according to different filters, for example when selecting a specific year, the report will dynamically display key sales indicators such as total sales, total units sold, number of orders, average shopping basket size, repeat purchase percentage, etc.

The report also includes:

- Repeat purchase percentage by continent.
- Average shopping basket size by country
- The percentage of online sales versus sales in physical stores
- top 5 selling products
- top 5 employees according to sales

With bookmarks, I've integrated three graphs offering a comprehensive year-to-date (YTD) overview, comparing current totals for sales, orders, and units with the same period from the previous year.Easily navigate between time periods, aggregate data by year, quarter, store type and seizure categories, and clear all slicers with intuitive buttons for a seamless user experience.



Total Customer	Total Sales	Total Units	Total Orders	Resell Percentage	AVG Basket Size	Avg Per Order
19.1K	$109.8M	274.9K	31.5K	96.2%	8.7	$3,491

Top 5 Employees By Sales

FullName	Total Sales	Total Units	Total Orders
Jae Pak	$8,503,311	26,231	348
Jillian Carson	$10,065,770	27,051	473
Linda Mitchell	$10,366,978	27,229	418
Michael Blythe	$9,293,873	23,058	450
Tsvi Reiter	$7,170,989	16,431	429

Top 5 Selling Products

Category	Sub Category	Product	Total Units	Total Sales
Shirts	Casual	Flannel Checked Shirt	2,977	$4,400,588
Pants	Formal Trousers	Wide-Fit Pleated Pants	2,664	$4,009,490
Coats and jackets	Winter Jackets	Ultra Light Down Jacket	2,394	$3,693,674
Shirts	Graphic Tees	U Crew Neck T-Shirt	2,234	$3,438,475
Pants	Denim	Ultra Stretch Color Jeans	2,216	$3,434,253

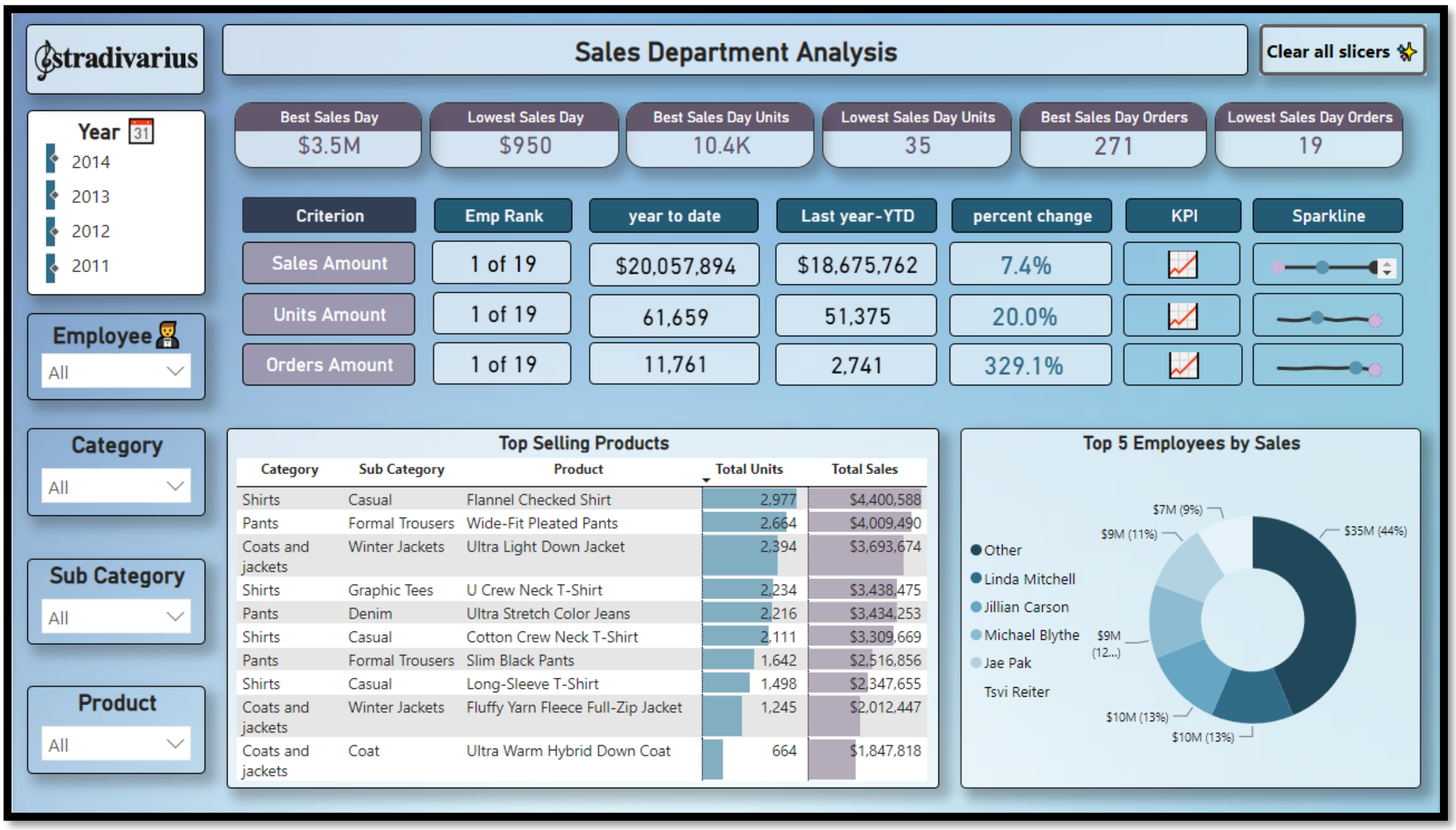
5.3. Salse department analysis Report

In the Sales Department Report, you'll find essential insights and key performance indicators (KPIs) tailored for the sales team.

The report includes:

- General KPI Data: Highlights the Best and Lowest Sales Days along with corresponding unit metrics.
- Employees Sales Performance: Provides employee rankings, year-to-date (YTD) total sales, a comparison to the same period last year, percentage change, KPIs, and sparklines for sales amount, units amount, and orders amount.
- Top 5 Best-Selling Products: Identifies the top-performing products along with their quantities.
- Top 5 Best-Employees by Sales: Showcases the top-performing employees based on sales.

The report is designed for easy navigation with buttons for filtering data by year, employee, category, sub-category and product. Additionally, a 'Clear All Slicers' button ensures a seamless user experience.



Sales Department Analysis

Best Sales Day	Lowest Sales Day	Best Sales Day Units	Lowest Sales Day Units	Best Sales Day Orders	Lowest Sales Day Orders
$3.5M	$950	10.4K	35	271	19

Criterion	Emp Rank	year to date	Last year-YTD	percent change	KPI	Sparkline
Sales Amount	1 of 19	$20,057,894	$18,675,762	7.4%		
Units Amount	1 of 19	61,659	51,375	20.0%		
Orders Amount	1 of 19	11,761	2,741	329.1%		

Top Selling Products

Category	Sub Category	Product	Total Units	Total Sales
Shirts	Casual	Flannel Checked Shirt	2,977	$4,400,588
Pants	Formal Trousers	Wide-Fit Pleated Pants	2,664	$4,009,490
Coats and jackets	Winter Jackets	Ultra Light Down Jacket	2,394	$3,693,674
Shirts	Graphic Tees	U Crew Neck T-Shirt	2,234	$3,438,475
Pants	Denim	Ultra Stretch Color Jeans	2,216	$3,434,253
Shirts	Casual	Cotton Crew Neck T-Shirt	2,111	$3,309,669
Pants	Formal Trousers	Slim Black Pants	1,642	$2,516,856
Shirts	Casual	Long-Sleeve T-Shirt	1,498	$2,347,655
Coats and jackets	Winter Jackets	Fluffy Yarn Fleece Full-Zip Jacket	1,245	$2,012,447
Coats and jackets	Coat	Ultra Warm Hybrid Down Coat	664	$1,847,818

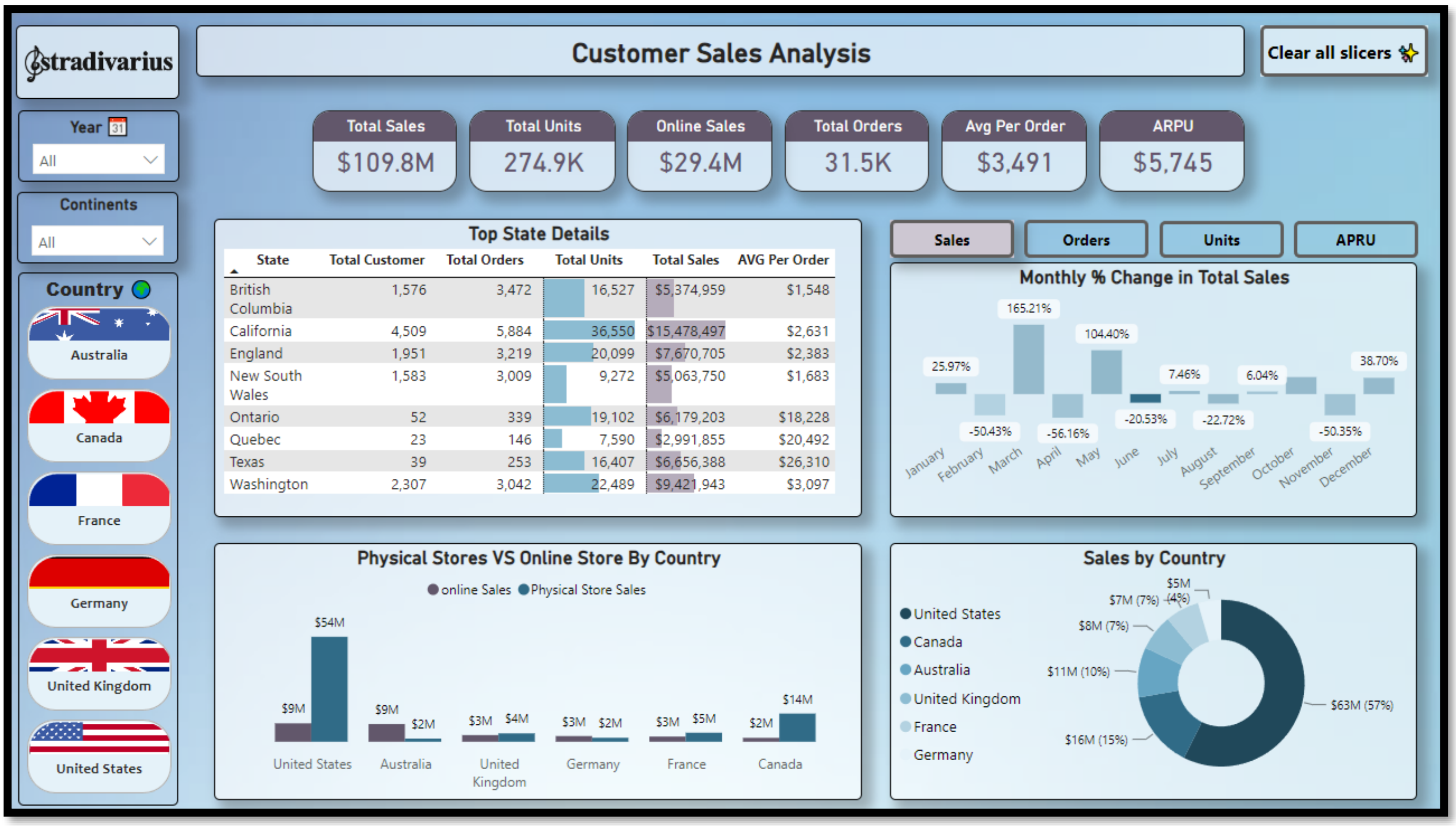
5.4. Customer Sales Analysis Report

In the Customer Report, comprehensive details about the company's customers are presented, organized visually through the display of country flags. When selecting a specific country, the report dynamically showcases key sales metrics, including total sales, total units, total orders, average per order, online sales, and average revenue per user.

The report also includes:

- Detailed information on the top 5 states.
- Total Sales breakdown by country.
- A comparison of total sales between Physical Stores and the Online Store, categorized by country.

With bookmarks, I've incorporated four graphs that highlight monthly percentage changes in Order Count, Units Sold, Total Sales, and ARPU. Easily navigate through time periods, aggregate data by year or continent, and clear all slicers with intuitive buttons for a seamless user experience.



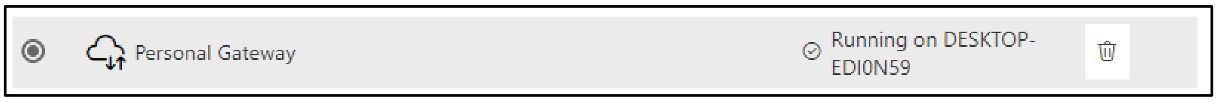
Total Sales	Total Units	Online Sales	Total Orders	Avg Per Order	ARPU
$109.8M	274.9K	$29.4M	31.5K	$3,491	$5,745

Top State Details

State	Total Customer	Total Orders	Total Units	Total Sales	AVG Per Order
British Columbia	1,576	3,472	16,527	$5,374,959	$1,548
California	4,509	5,884	36,550	$15,478,497	$2,631
England	1,951	3,219	20,099	$7,670,705	$2,383
New South Wales	1,583	3,009	9,272	$5,063,750	$1,683
Ontario	52	339	19,102	$6,179,203	$18,228
Quebec	23	146	7,590	$2,991,855	$20,492
Texas	39	253	16,407	$6,656,388	$26,310
Washington	2,307	3,042	22,489	$9,421,943	$3,097

5.5. Refreshing reports and dashboards

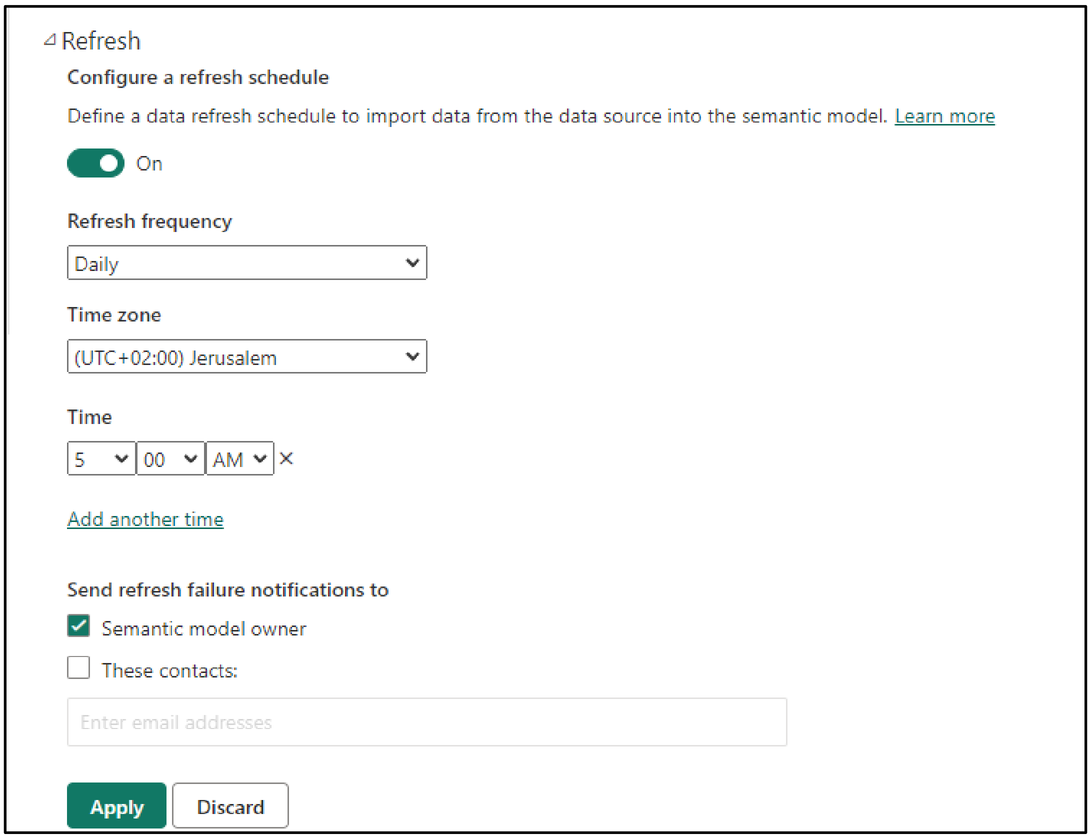
Similar to the data refresh procedure outlined in Chapter 4.3 for the Data Mart the settings for refreshing report and dashboard data are configured on a daily basis. This ensures complete consistency between the Data Mart and the Power BI dashboards. The information synchronization takes place through the Gateway to the EDI0N59 computer, where the databases are stored.



Personal Gateway

Running on DESKTOP-EDI0N59

The report is scheduled for daily refresh at 5:00 AM, following the completion of the data refresh procedure in the Data Mart. Alerts have also been configured to notify of any timing errors during this process.



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

Send refresh failure notifications to

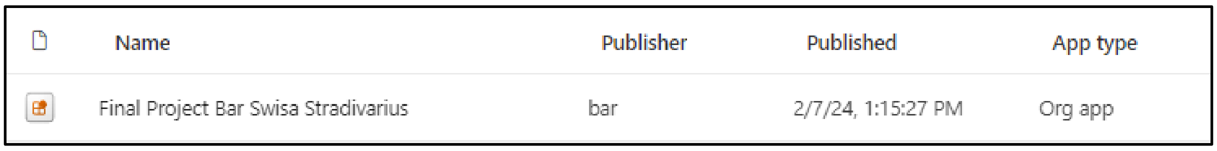
☑ Semantic model owner

☐ These contacts:

Enter email addresses

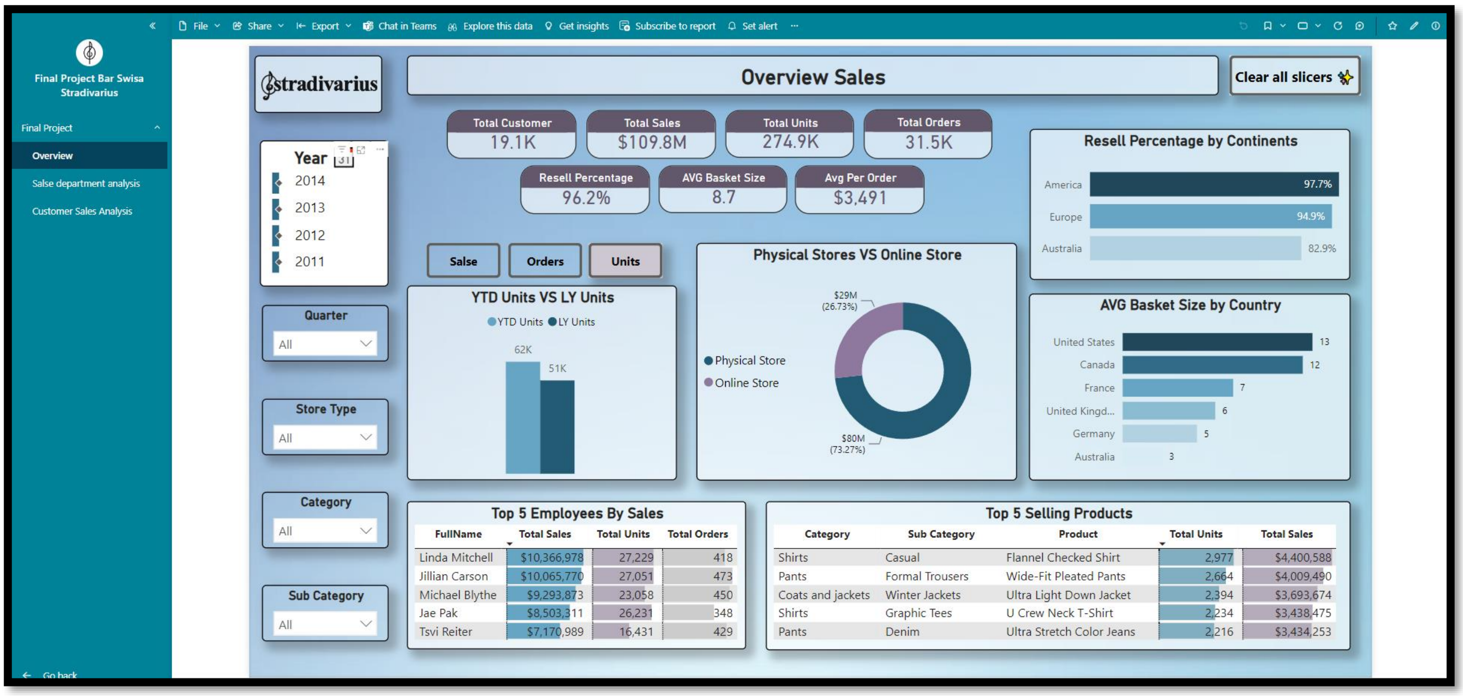
5.6. Application

To improve user accessibility and convenience, we've centralized all reports and dashboards into a user-friendly application named "Final Project." This consolidation ensures that users can easily navigate and access the necessary information from a single, organized platform. The Final Project application serves as a comprehensive hub for streamlined and efficient data utilization.



	Name	Publisher	Published	App type
	Final Project Bar Swisa Stradivarius	bar	2/7/24, 1:15:27 PM	Org app

Link: [Final_Project_BarSwisa_Stradivarius]()



Total Customer	Total Sales	Total Units	Total Orders
19.1K	$109.8M	274.9K	31.5K

Resell Percentage	AVG Basket Size	Avg Per Order
96.2%	8.7	$3,491

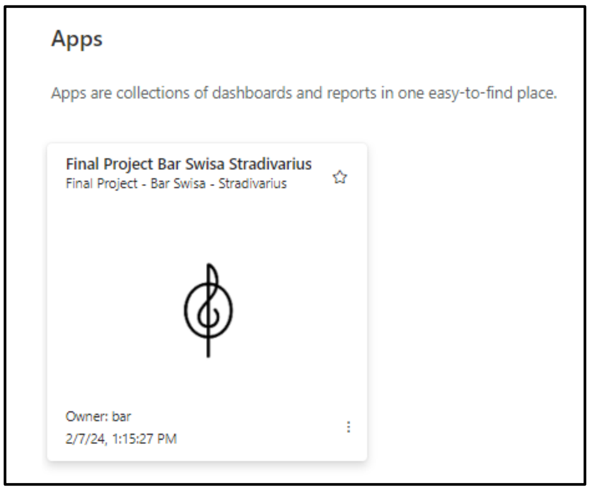
Top 5 Employees By Sales

FullName	Total Sales	Total Units	Total Orders
Linda Mitchell	$10,366,978	27,229	418
Jillian Carson	$10,065,770	27,051	473
Michael Blythe	$9,293,873	23,058	450
Jae Pak	$8,503,311	26,231	348
Tsvi Reiter	$7,170,989	16,431	429

Top 5 Selling Products

Category	Sub Category	Product	Total Units	Total Sales
Shirts	Casual	Flannel Checked Shirt	2,977	$4,400,588
Pants	Formal Trousers	Wide-Fit Pleated Pants	2,664	$4,009,490
Coats and jackets	Winter Jackets	Ultra Light Down Jacket	2,394	$3,693,674
Shirts	Graphic Tees	U Crew Neck T-Shirt	2,234	$3,438,475
Pants	Denim	Ultra Stretch Color Jeans	2,216	$3,434,253

The application is accessible on mobile devices either by directly connecting to the provided link or by using the link within the POWER BI application. This flexibility allows users to access the application seamlessly, ensuring a convenient and responsive experience whether they prefer accessing it directly through a web link or through the dedicated POWER BI mobile application.



Apps

Apps are collections of dashboards and reports in one easy-to-find place.

Final Project Bar Swisa Stradivarius
Final Project - Bar Swisa - Stradivarius

Owner: bar
2/7/24, 1:15:27 PM