

PULL&BEAR

BI System Specification Documentation
Version 3.0

Yael Mann

Contents

1. General.....	3
1.1. Project objectives.....	3
1.2. Project Contents	3
2. Gantt	4
3. Technical Specification.....	4
3.1. Prerequisites	4
3.2. Solution Architecture	4
4. Functional Specification	5
4.1. ETL processes.	5
4.2. Defining JOBS in SSIS.....	10
4.3 Description of Measures for Power BI Reports	12
4.3.1 Overview Dashboard.....	14
4.3.2 Customer Sales Analysis.....	14
4.3.3 Employee Sales Analysis	16
4.4 Reports and Dashboards refresh	17
4.5 The App	18

1. General

1.1. Project objectives

The purpose of the project is to provide a Full-Scale BI Solution Creation from PriorityERP Database for Pull & Bear, 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 Pull & Bear. 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.
 - **Dim_Date** – A table of dates for data analysis over time.

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.

[Link for the ERD](#)

[Link for the Source to Target \(S2T\)](#)

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

Overview Dashboard: Provides a snapshot of sales performance with revenue growth, top products, and revenue breakdown by store type and geography.

Customer Sales Analysis Report: Analyzes customer-centric metrics and geographical trends, including ARPC, revenue by store type, continent, and country, and orders by country.

Employees Sales Analysis Report: Evaluates employee performance and sales trends with insights into top products, revenue comparison, monthly revenue changes, revenue by

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 Warehouse using SSIS. Finally, the presentation of measures in reports and visuals will be presented in Power BI.

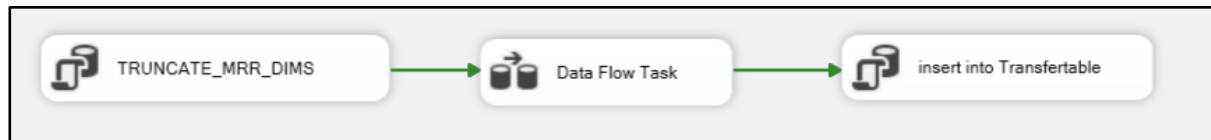
4. Functional Specification

4.1. ETL processes.

1. MRR_Dim package:

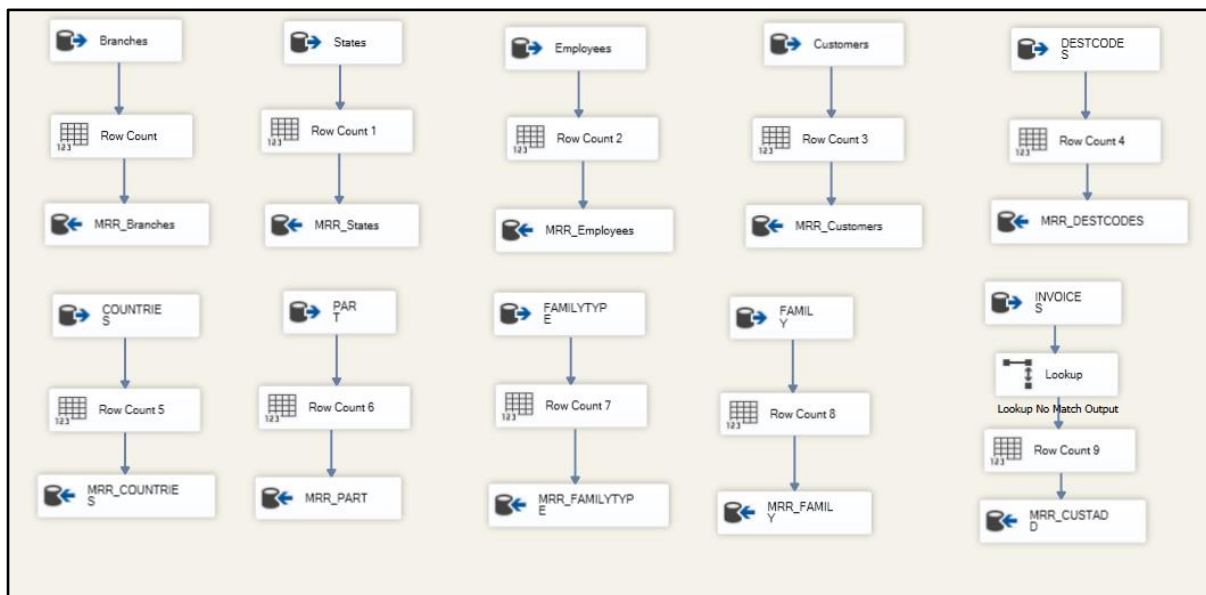
Into the Execute SQL Task - truncate_mrr:

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



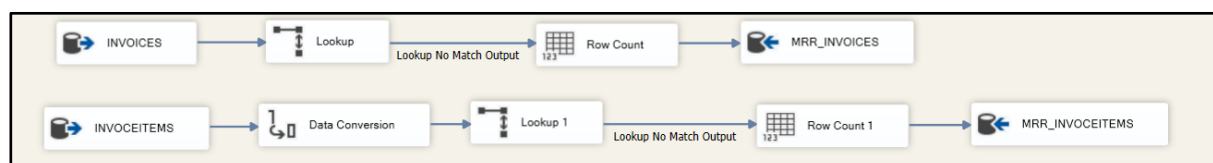
Finally, we insert values to the transfer table. The TransferTable serves as a comprehensive log, meticulously capturing every update and insertion step as data moves through the stages from the database (DB) to the Data Mart.

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



2. MRR_Fact package:

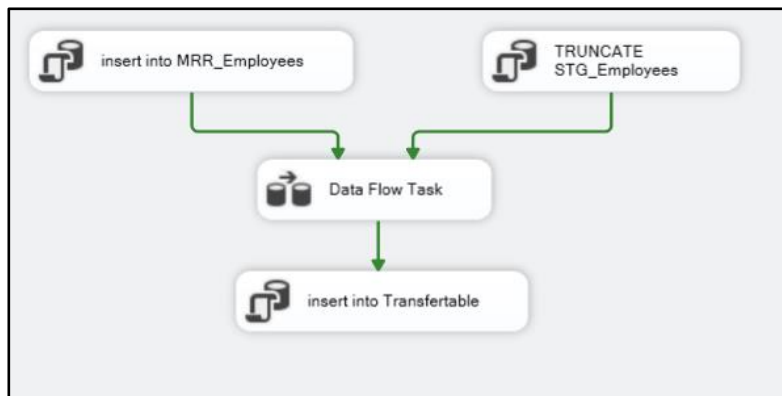
Into the Lookup - Invoices & InvoiceItems:



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).

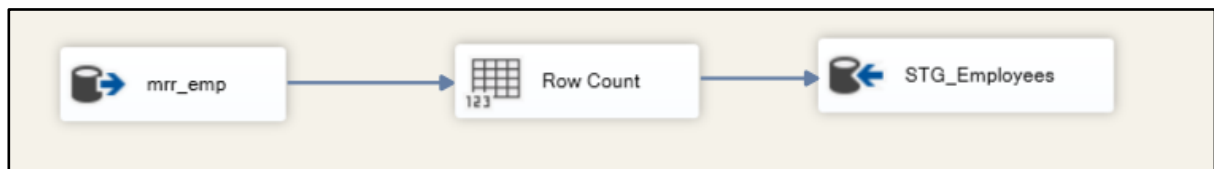
3. STG_Dim_Employees package:

In the Control Flow:



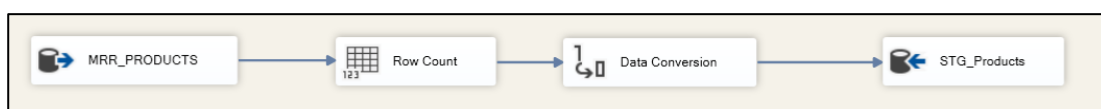
We updated the MRR_Employees table manually with a SQL command that adds a new employee that is associated with the Employee ID "77777". We wanted to make sure that orders placed online are properly associated with this employee.

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. Finally, we will insert the information into the stg_Employees table.

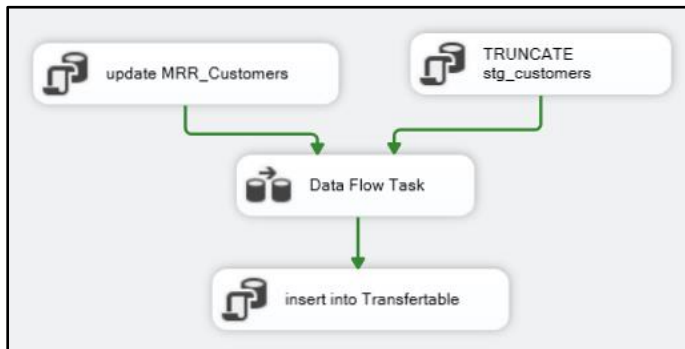
4. STG_Products package



We collected information about products from three tables: MRR_Part, MRR_Familly and MRR_FamillyType. The selected columns include details such as 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.

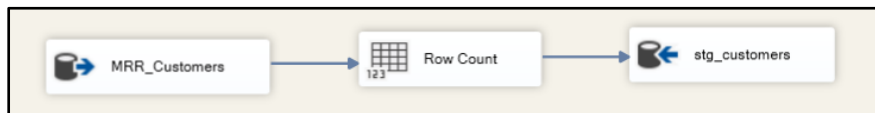
5. STG CUSTOMERS package

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 associat 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.

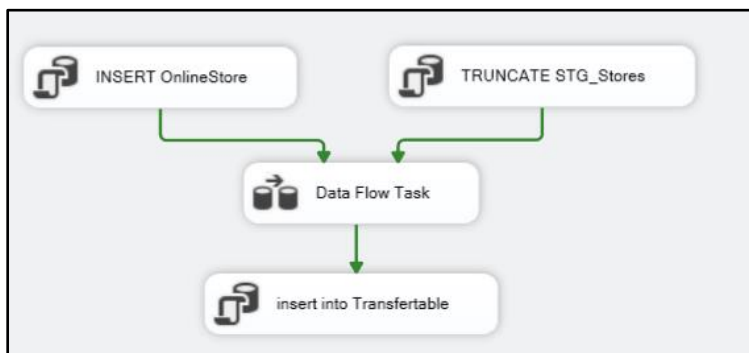
In the Data Flow:



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.

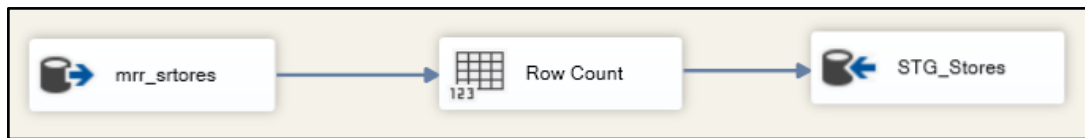
6. STG STORES

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”.

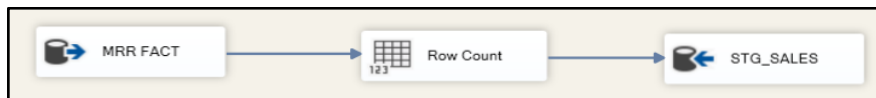
In the Data Flow:



We collected 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.

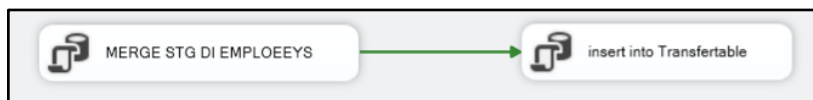
7. STG SALES

In the Data Flow:



We collected 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, EmplID, ZoneID, ProductID, Qty, Price and Discount.

8. DW EMPLOYEES

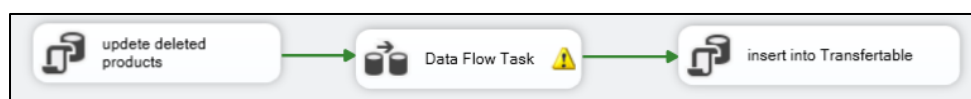


Into the Execute SQL Task – MERGE:

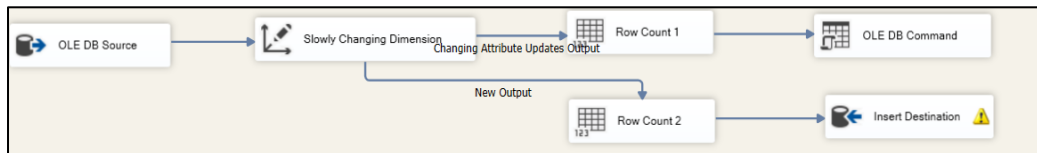
Using to synchronize data between the Dim_Employees and stg_employees tables based on EMP_ID. It performs the following actions:

- 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. DW PRODUCTS

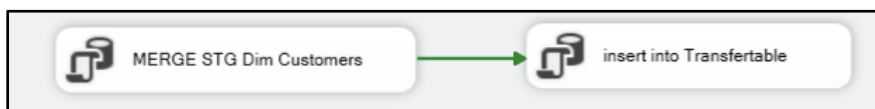


Into the Execute SQL Task- update deleted - In summary, the query updates certain columns in dim_Products for records that meet the specified conditions, indicating that the products are not active and updating the modification date.



In the Slowly Changing Dimension - We will separate new lines from updated lines with a productID Into OLE DB Command – update - We will update the changed fields.

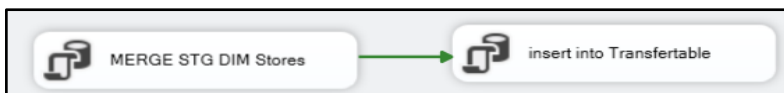
10. DW CUSTOMERS



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. DW 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.

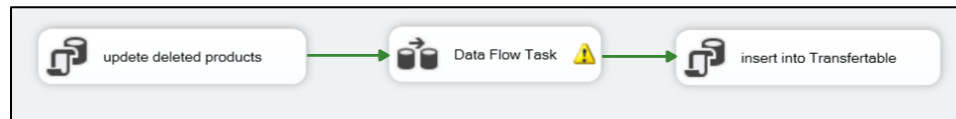
12. DW FACT SALES



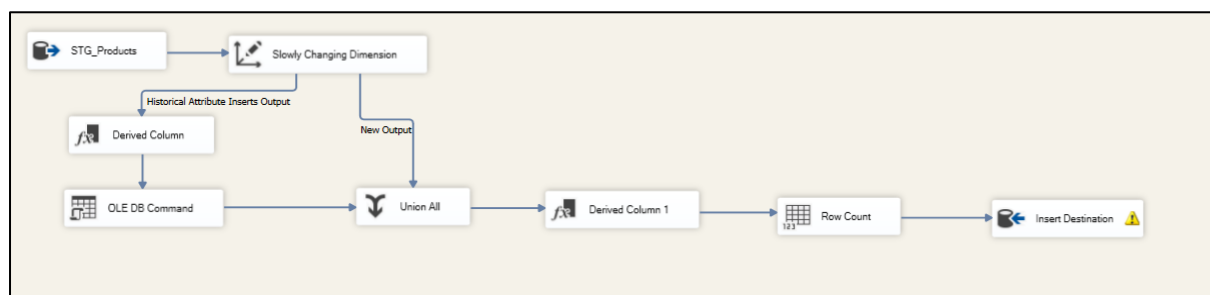
In the OLE DB Source - STG_SALES, we will channel comprehensive data into the FACT_Sales table. Subsequently, in the Derived Column (data type change) phase, we will meticulously refine data types that necessitate modification. Following this, within the Derived Column (total column)

transformation, we will dynamically compute the Total column using the specified formula: $(Qty * Price * (1 - Discount) * 1.17)$. Finally, the meticulously transformed and calculated data will be inserted into the DWH_FACT_SALES table.

13. DW PRODUCTS HISTORY



Into the Execute SQL Task - In summary, the query updates certain columns in dim_Products for records that meet the specified conditions, indicating that the products are not active and updating the modification date.



Inside the "dim_Products_History" package, a fundamental operation involves the insertion of records into the "dim_Products_History" table. The structure of this table mirrors that of the "dim_Products" table, encompassing identical fields. Notably, two additional date fields are incorporated, signifying a start date and an end date for a specific version.

The workflow is designed such that when a record in the "dim_Products" table undergoes an update, the previous version of that record is systematically preserved in the "dim_Products_History" table. This archival process is executed alongside the recording of date fields, meticulously specifying the temporal range during which the particular version remained pertinent. Consequently, this strategy not only ensures the preservation of historical data but also facilitates the traceability of changes over time, a crucial aspect in managing the evolution of product information within the data model.

4.2. Defining JOBS in SSIS

To facilitate the daily refresh and loading process, deploy operation was executed from SSIS to SSMS. Subsequently, 3 jobs were created to run on a daily schedule at a fixed time. These jobs have 2-6 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 steps.

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

Step	Name	Type	On Success	On Failure
1	DimStores	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
2	DimEmp	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
3	DimProducts	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
4	DimCust	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
5	ProductHistory	SQL Server Integration Services Package	Go to the next step	Quit the job reporting failure
6	FactSales	SQL Server Integration Services Package	Quit the job reporting success	Quit the job reporting failure

It will automatically run each day at 02:00 AM.

New Job Schedule

Name:

refresh DM

Jobs in Schedule

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

OK

Cancel

Help

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	UPDATES_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

Name: Jobs in Schedule

Schedule type: Recurring ☒ Enabled

One-time occurrence

Date: Time:

Frequency

Occurs: Weekly

Recurs every: week(s) on

☐ Monday
 ☐ Wednesday
 ☐ Friday
 ☐ Saturday
 ☐ Tuesday
 ☐ Thursday
 ☒ Sunday

Daily frequency

☒ Occurs once at:
☐ Occurs every: hour(s)

Starting at:
 Ending at:

Duration

Start date:
☒ End date:
☐ No end date:

Summary

Description:

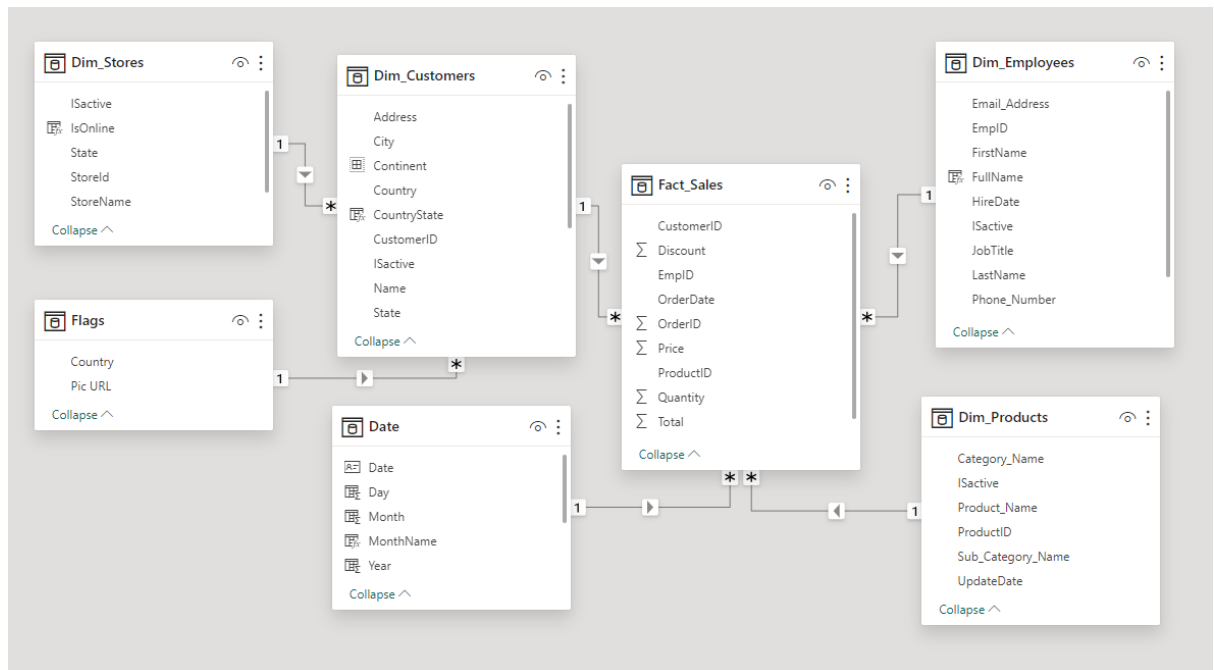
Occurs every week on Sunday at 6:00:00 AM. Schedule will be used between 1/29/2024 and 2/18/2024.

4.3 Description of Measures for Power BI Reports

Visualization was made by Power BI Desktop and published in Power BI Service.

The Semantic layer model is based on a few sources:

- PriorityERP Operational DB
- SQL SERVER (detailed in section 3)
- Date table
- Flags table



Measures were defined in M_Table:

- Average Order Price- The Average price per order
- Average Revenue per Customer
- Average unit price- The Average price per unit sold.
- Last Year Total Sales- calculate the revenue for the same period last year.
- YoY% Last Year Total Sales- Percent year over year change of total sales.
- LY YTD Total Sales- calculate the revenue for the months from the beginning of the year until now, from last year's data.
- Total Orders- count of orders
- Total Sales- sum of revenues
- MoM% Total Sales – percent month over month change for total sales.
- YoY% Total Sales- percent year-over-year change for total sales.
- Total Sales YTD- calculate the revenue for the months from the beginning of the year until now.
- Total Units- sum of units sold.
- Total Sales by EmpName- sum of revenue with employee's filter
- Total Sales by Product- sum of revenue with Products filter
- Total Units by Product- sum of units sold with Products filter.

M_Table	
<input type="checkbox"/>	Average Order Price
<input type="checkbox"/>	Average Revenue per Customer
<input type="checkbox"/>	Average Unit Price
<input type="checkbox"/>	Average Unit Price (Products)
<input type="checkbox"/>	Column
<input type="checkbox"/>	Last Year Total Sales
<input type="checkbox"/>	Last Year Total Sales YoY%
<input type="checkbox"/>	LY YTD Total Sales
<input type="checkbox"/>	Total Orders
<input type="checkbox"/>	Total Sales
<input type="checkbox"/>	Total Sales MoM%
<input type="checkbox"/>	Total Sales YoY%
<input type="checkbox"/>	Total Sales YTD
<input type="checkbox"/>	Total Units
<input type="checkbox"/>	TotSalesByEmpName
<input type="checkbox"/>	TotSalesByProduct
<input type="checkbox"/>	TotUnitsByProducts

4.3.1 Overview Dashboard

The Pull & Bear Overview dashboard provides a comprehensive snapshot of key performance indicators and trends crucial for strategic decision-making. It consists of several charts:

Revenue Cards with Growth Percentage: This card displays the total revenue generated, along with the percentage growth compared to the previous year. It serves as a quick indicator of the overall financial health of the business.

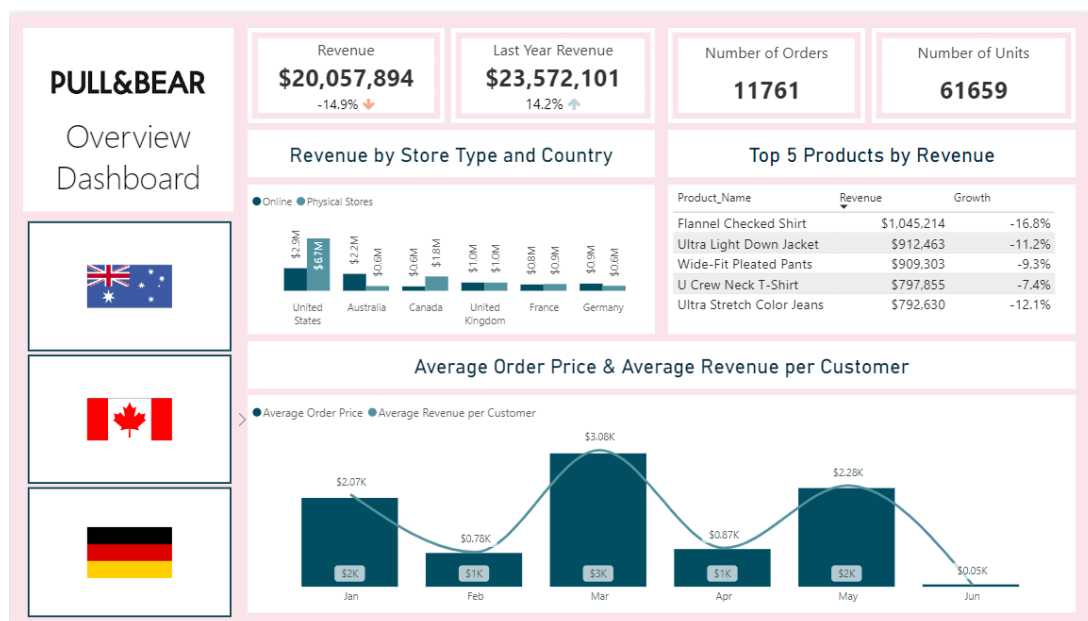
Top 5 Products by Revenue: This chart highlights the top five products contributing the most to the revenue. It helps identify bestselling items and informs management and marketing strategies.

Revenue by Store Type and Country: This visual presents revenue breakdown based on store types (online or physical) and geographical locations (countries). It offers insights into sales performance across different channels and regions, facilitating targeted marketing campaigns and resource allocation.

Average Order Price: This chart tracks the average order price over time, providing visibility into changes in customer purchasing behavior and pricing strategies' effectiveness. It helps optimize pricing strategies and identify opportunities for upselling or cross-selling.

Average Revenue per Customer Over Time: This chart illustrates the average revenue generated per customer over various time periods. It assists in evaluating customer loyalty, identifying high-value customer segments, and assessing the effectiveness of marketing and retention efforts.

Overall, this BI dashboard offers a holistic view of the fashion retailer's performance, enabling stakeholders to make data-driven decisions to drive growth, optimize operations, and enhance customer satisfaction.



4.3.2 Customer Sales Analysis

The Customer Sales Analysis report provides a detailed insight into the company sales performance, focusing on customer-centric metrics and geographical trends. It comprises the following charts:

Average Revenue per Customer (ARPC) Over Time: This chart tracks the ARPC metric over time, offering insights into changes in customer spending behavior and revenue trends. It helps in

understanding the effectiveness of marketing strategies and the overall health of customer relationships.

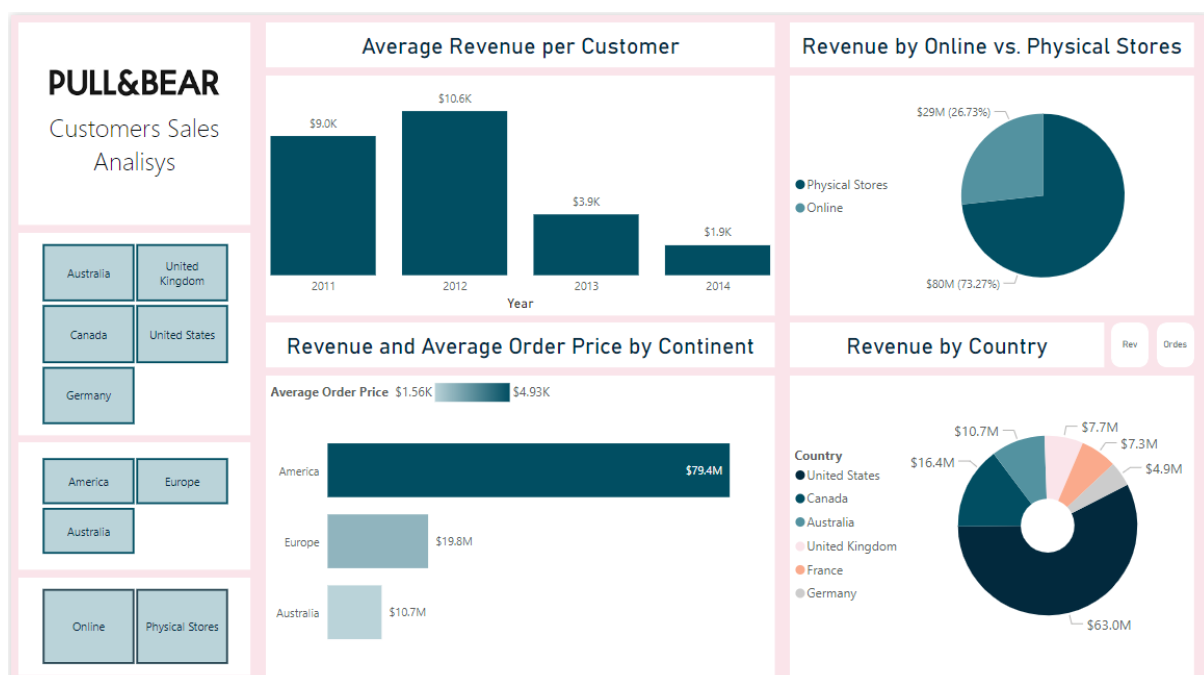
Revenue by Store Type: This visual presents a breakdown of revenue generated across different store types, distinguishing between online and physical stores. It enables comparison of sales performance between various channels, aiding in resource allocation and strategic decision-making.

Revenue and Average Order Price by Continent: This chart provides a geographical analysis of revenue and average order price by continent. It allows for the identification of regional trends and variations in customer purchasing behavior, helping tailor marketing and pricing strategies accordingly.

Revenue by Country: This visual displays revenue figures for each country, highlighting the top-performing markets. It assists in understanding the geographical distribution of sales and identifying opportunities for market expansion or targeted promotional activities.

Orders by Country: This chart showcases the number of orders placed from different countries, offering insights into customer engagement and market penetration. It helps in understanding demand patterns and assessing the effectiveness of sales and marketing initiatives in various regions.

Overall, the Customer Sales Analysis report provides valuable insights into customer behavior and geographical sales trends, empowering retailers to optimize their strategies, enhance customer satisfaction, and drive business growth.



4.3.3 Employee Sales Analysis

The Employees Sales Analysis report offers a comprehensive overview of the sales performance of employees within the organization. It includes the following key charts:

Top 5 Products by Average Unit Price and Number of Units: This chart identifies the top five products based on both their average unit price and the number of units sold. It highlights high-value items and provides insights into sales volume and profitability, aiding in inventory management and pricing strategies.

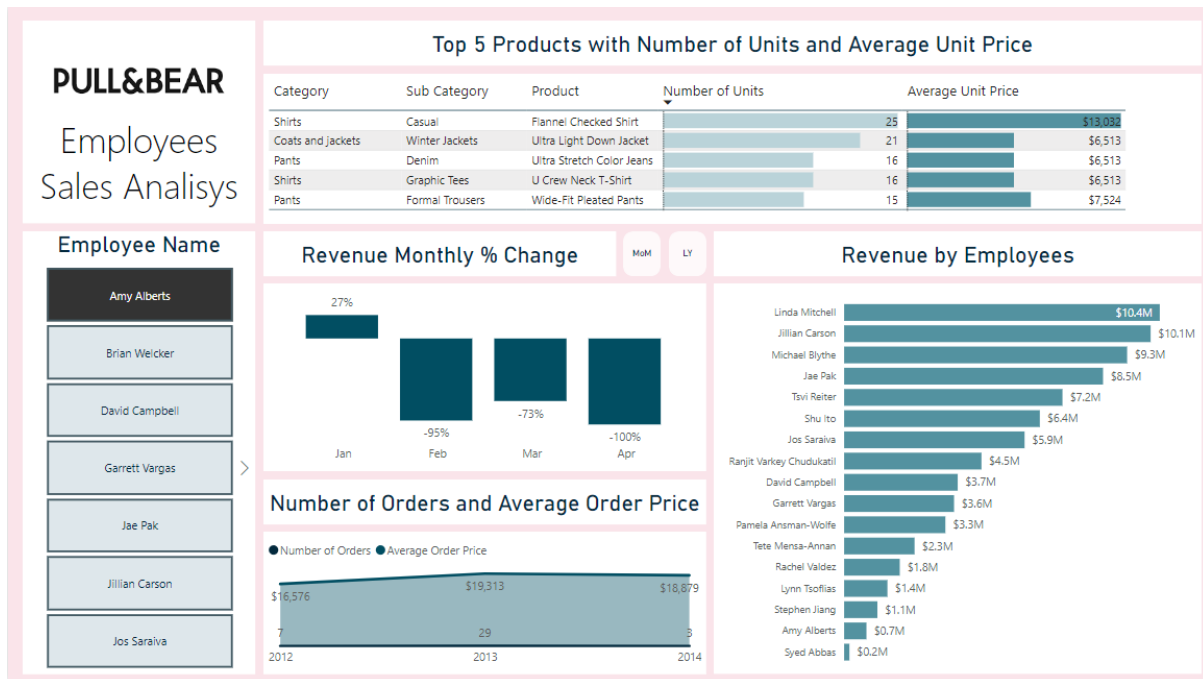
Current Revenue vs. Last Year: This visual compares the current revenue with the revenue from the previous year. It enables stakeholders to assess the year-over-year growth or decline in sales performance, facilitating informed decision-making and forecasting.

Monthly Revenue Change (Precent): This chart illustrates the monthly changes in revenue over time. It helps identify seasonal trends, sales fluctuations, and growth patterns, guiding resource allocation and strategic planning efforts.

Revenue by Employee: This visual presents revenue contributions from sales representatives. It enables performance evaluation, identifies top-performing employees, and provides insights into sales effectiveness and productivity.

Number of Orders and Average Order Price: This chart displays the total number of orders processed by employees along with the average order price. It offers insights into customer engagement, sales volume, and order value, helping optimize sales strategies and customer relationship management.

Overall, the Employees Sales Analysis report empowers managers and decision-makers to evaluate employee performance, monitor sales trends, and identify opportunities for improvement, ultimately driving sales growth and enhancing operational efficiency within the organization.



4.4 Reports and Dashboards refresh

Like the data refreshing process in the DWH (as detailed in Chapter 3), settings for refreshing the data of reports and dashboards were established on a daily basis. This effectively maintains a complete synchronization between the data in the DWH and the dashboards in Power BI.

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

Apply

Discard

The report is scheduled for daily refresh every day at 05:00 AM. The process will occur after the data refresh process in the DWH. Additionally, notifications were set up for timing errors.

4.5 The App

For the user's convenience, all reports and dashboards have been consolidated under an application named "Yael Mann."

[App Link](#)