

ARMDD — Data Warehouse Project

Ana Rita Silva (1211152)

Jorge Cruz (1221715)

Mestrado em Engenharia Informática - Engenharia de Dados
Instituto Superior de Engenharia do Porto

November 2025

Contents

1	Introduction	2
1.1	Project Overview	2
1.2	Project Objective	2
2	Dimensional Analysis	2
2.1	Kimball Methodology	2
2.2	Business Area	2
2.3	Granularity	2
2.4	Identified Dimension	3
2.5	Fact Tables	3
2.6	Attribute's description of each dimension	3
2.7	Attribute's description of each Fact Table	6
3	Dimensional Model	8
3.1	Fact Table: FactOrders	8
3.2	Fact Table: FactCurrencyTable	9
4	Solution Architecture	10
4.0.1	Operational Source	10
4.0.2	Staging Area	10
4.0.3	Data Mart	10
4.0.4	OLAP and Power BI	11
5	Data Structure	11
5.0.1	Staging Area	11
6	Transformation operations	15
6.0.1	Standardization of Formats	15
7	Clean operations	15
7.0.1	Duplicate Removal	15
7.0.2	Validation rules	16
8	SCD - Slowly Changing Dimension	16
8.0.1	SCD Type 1 - Overwrite	16
8.0.2	SCD Type 2 - Retain History	17
8.0.3	SCD Type 3 - Retain valid alternative values	17
8.0.4	SCD Type 4 - New history table	18
8.0.5	SCD Type 6 - Hybrid approach	18
9	Mapping	18

1 Introduction

1.1 Project Overview

This project focuses on the development of a *data mart* for **PlusOrders**, a national e-commerce company operating internationally in the luxury fashion sector. Through multiple websites, customers from various countries can place orders, which are then centralized in a unified operational system that records all relevant information.

1.2 Project Objective

The main goal of this project is to create a *data mart* that consolidates, transforms, and organizes operational data to enable multidimensional analysis of customers orders. The solution aims to ensure data quality, consistency and integration, addressing issues such as duplicated product records, inconsistent descriptions, and multiple the handling of multiple currencies. This centralized structure will allow analyses of sales values, customer profiles and product categories, supporting strategic decision-making through efficient and flexible data access.

2 Dimensional Analysis

2.1 Kimball Methodology

The development of the dimensional model follows the **Kimball methodology**, which adopts a bottom-up approach focused on business processes. This approach keeps the model simple, fast for analysis and easy to understand. In line with Kimball's principles, the model is structured using dimensions and fact tables, organized within a star schema to ensure clarity, scalability, and high query performance.

2.2 Business Area

The business area analysed is the management of customer orders, including detailed information about products, customers, websites and shipping processes stores in the **PlusOrders** operational system.

2.3 Granularity

The granularity chosen for the fact table is the **order line**, which represents the **orders of products by customer, site, location and day**. This granularity enables detailed analysis and efficient aggregation across time, customers, products, site and shipping.

2.4 Identified Dimension

- **DimCustomer:** stores customer identification and demographic attributes (gender and birthday) supporting the analysis of customer segments;
- **DimProduct:** includes detailed product information;
- **DimSize:** stores product size information used in specific analyses;
- **DimDate:** contains temporal information used for time-based analysis;
- **DimCurrency:** identifies the currencies used in orders;
- **DimShipping:** contains information about the shipping location used for dispatching orders;
- **DimSite:** represents the sales website where the order was placed, including site name and URL.

2.5 Fact Tables

- **FactOrders:** stores each order line, including the customer, product, site, shipping country, box, size and currency, together with quantities and prices.
- **FactCurrencyRate:** contains daily exchange rates between currencies to support monetary conversions and standardized analysis in EUR and in local currency

2.6 Attribute's description of each dimension

Several important attributes in each dimension were identified.

DimDate

- **DateKey (int(10)) - Primary Key**
- FullDate (date)
- Year (int(4))
- Month (tinyint(3))
- Day (tinyint(3))
- DayNumberOfYear (int(4))
- DayNumberOfMonth (tinyint(3))
- DayNumberOfWeek (tinyint(3))
- Week (tinyint(3))

- DayOfWeek (int(10))
- Weekend (varchar(3))
- MonthName (varchar(10))
- Quarter (tinyint(3))
- Semester (tinyint(3))
- Trimester (tinyint(3))
- Season (tinyint(3))

DimCostumer

- **CostumerKey (int(10)) - Primary Key**
- CustomerID (int(10)) - *business identifier of the operational system*
- CostumerBirthday (date)
- GeographicRegion (tinyint(1))
- Language (char(2))
- Gender (char(1))
- IsVipCustomer (tinyint(1)) - flag indicating VIP customers
- Country (varchar(50))
- EffectiveDate (date)
- ExpireDate (date)
- IsCurrent (tinyint(1))

DimProduct

- **ProductKey (int(10)) - Primary Key**
- ProductID (int(10))
- ProductDescription (varchar(255)) - standardized product description.
- IsActive (tinyint(1)) - indicates if the product is active or not.
- AvailablePortal (tinyint(1)) - indicates if the product is available online or not.
- CategoryName (varchar(50))

- BoxName (varchar(50))
- Gender (char(1))
- EffectiveDate (date)
- ExpireDate (date)
- IsCurrent (tinyint(1))

DimCurrency

- **CurrencyKey (int(10)) - Primary Key**
- CurrencyCode (char(3))
- CurrencyName (varchar(50))

DimShipping

- **ShippingKey (int(10)) - Primary Key**
- CountryCode (char(3)) - country code used for shipping.
- CountryName (varchar(20))
- IsEuroZone (varchar(3)) - indicates if the shipping country belongs to the Euro zone or not.
- VAT (tinyint(3))
- EffectiveDate (date)
- ExpiredDate (date)
- IsCurrent (tinyint(1))

DimSite

- **SiteKey (int(10)) - Primary Key**
- SiteID (int(10))
- SiteName (varchar(50))
- SiteURL (varchar(255))
- SiteInitials (char(3)) - short code for the site.
- IsSite (tinyint(1)) - flag used to identify valid sites
- EffectiveDate (date)

- `ExpireDate` (date)
- `IsCurrent` (tinyint(1))

`DimSize`

- **SizeKey** (int(10)) - **Primary Key**
- `SizeID` (int(10))
- `SizeName` (varchar(50))
- `Min` (tinyint(2))
- `Max` (tinyint(2))
- `EffectiveDate` (date)
- `ExpireDate` (date)
- `IsCurrent` (tinyint(1))

The dimensions related to Customers, Products, Shippings, Sizes and Sites include attributes such as *EffectiveDate*, *ExpireDate* and *IsCurrent*, enabling the maintenance of historical records.

2.7 Attribute's description of each Fact Table

`FactOrders`

- `DateKey` (int(10)) - Primary Key component, FK of `DimDate`
- `SiteKey` (int(10)) - Primary Key component, FK of `DimSite`
- `SizeKey` (int(10)) - Primary Key component, FK of `DimSize`
- `CustomerKey` (int(10)) - Primary Key component, FK of `DimCustomer`
- `ProductKey` (int(10)) - Primary Key component, FK of `DimProduct`
- `ShippingKey` (int(10)) - FK of `DimShipping`
- `OrderID` (int(10)) - identifier of the order in the source system
- `OrderLineID` (int(10)) - identifier of the order line in the source system
- `LocalCurrencyKey` (int(10)) - FK of `DimCurrency`
- `QuantityOrdered` (decimal(12, 4)) - quantity of the product in the order line

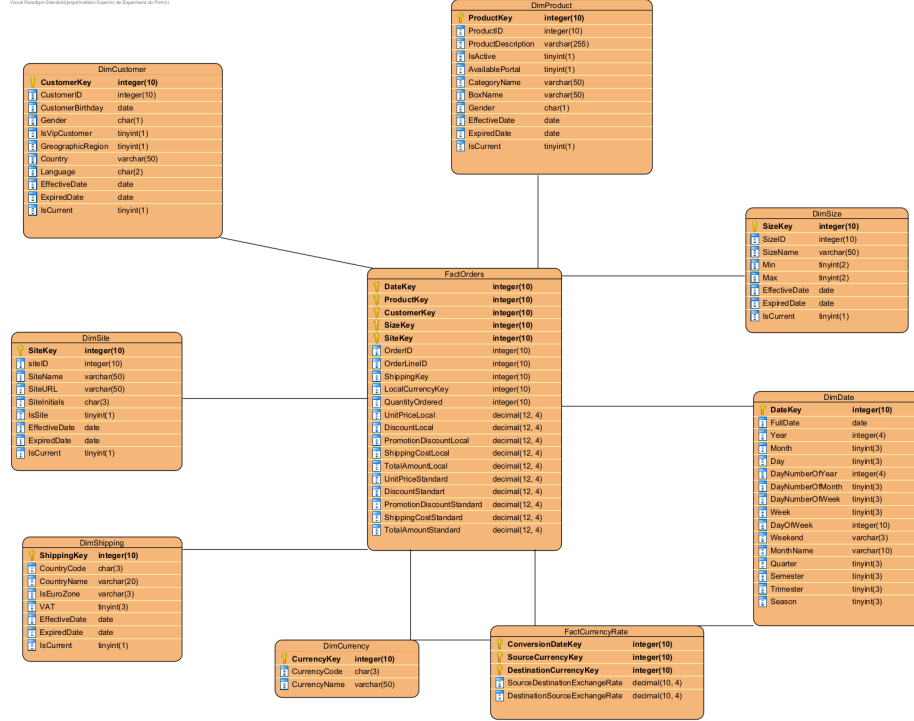
- UnitPriceLocal (decimal(12, 4)) - unit price of the product in the local currency
- DiscountLocal (decimal(12, 4)) - discount amount in the local currency
- PromotionDiscountLocal (decimal(12, 4)) - discount based on a promotion in the local currency
- ShippingCostLocal (decimal(12, 4)) - shipping cost value in the local currency
- TotalAmountStandard (decimal(12, 4)) - total amount of the order line in the local currency
- UnitPriceStandard (decimal(12, 4)) - unit price of the product in the standard currency
- DiscountStandard (decimal(12, 4)) - discount amount in the standard currency
- PromotionDiscountStandard (decimal(12, 4)) - discount based on a promotion in the standard currency
- ShippingCostStandard (decimal(12, 4)) - shipping cost value in the standard currency
- TotalAmountStandard (decimal(12, 4)) - total amount of the order line in the standard currency

Each row in FactOrders corresponds to a single order line at the most granular level, combining the keys of the main dimensions and the quantity measures.

FactCurrencyRate

- **ConversionDateKey (int(10)) - FK of DimDate**
- **SourceCurrencyKey (int(10)) - FK of Dim Currency (source currency)**
- **DestinationCurrencyKey (int(10))**
- SourceDestinationExchangeRate (decimal(10,4))
- DestinationSourceExchangeRate (decimal(10,4))

3 Dimensional Model



The dimensional model developed follows the Kimball methodology previously described. The selection of attributes, fact tables and dimensions were guided by the analytical requirements of the project and by the characteristics of the operational data provided.

3.1 Fact Table: FactOrders

The FactOrders table was defined as the central fact table of the model, as it contains the essential metrics related to customer orders, such as ordered quantity, unit price in local currency and shipping information.

All the attributes are connected to descriptive information through foreign keys linked to the dimensions.

- **Primary Key**

- A composite primary key created from the combination of ProductKey, CustomerKey, SiteKey, SizeKey and DateKey.

- **Foreign Keys**

- All the Keys referenced as Primary Keys plus ShippingKey and LocalCurrencyKey.

The composite primary key guarantees the uniqueness of each record in the FactOrders table. Additionally, several operational identifiers are persisted in both the fact and dimension tables (ProductID, CustomerID, OrderID, BoxID, SiteID, SizeID). These IDs correspond to entities in the operational system, ensuring consistency and traceability between the DataMart and the source system.

3.2 Fact Table: FactCurrencyTable

The *FactCurrencyTable* stores the daily exchange rates between currency pairs, supporting monetary conversion in the analytical layer.

- **Primary Key**

- A composite primary key formed by the combination of: *ConversionDateKey*, *SourceCurrencyKey* and *DestinationCurrencyKey*, identifying a currency conversion for a given day.

- **Foreign Keys**

- ConversionDateKey - DimDate
- SourceCurrencyKey - DimCurrency
- DestinationCurrencyKey - DimCurrency.

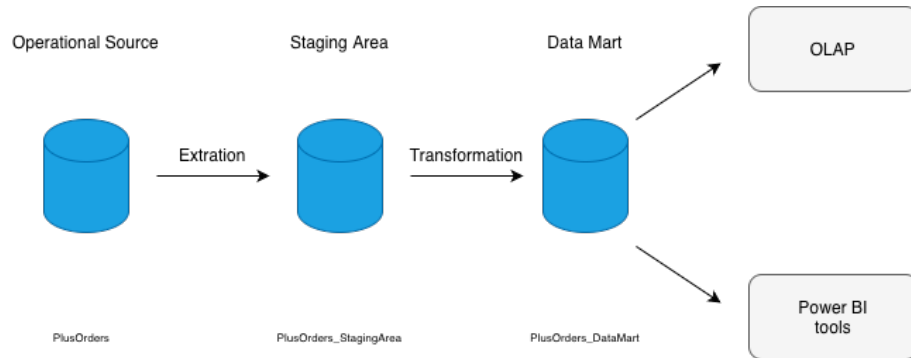
- **Measures**

- SourceDestinationExchangeRate - rate from source to destination currency
- DestinationSourceExchangeRate - inverse rate from destination to source currency.

4 Solution Architecture

The architecture designed for the project follows a multi-layered structure based on Kimball’s methodology, ensuring a clear separation between operational data, processing stages and analytical components.

The solution is divided into four main layers: **Operational Source**, **Staging Area**, **Data Mart** and **Analytics Layer**, represented in the following image.



4.0.1 Operational Source

The first layer corresponds to the **Operational Source** of the project, the *PlusOrders relational database*, provided in SQL Server format.

The database stores all customer-order information (customer, products, categories, sites, shipping, currency). However, as typical OLTP system, it includes inconsistencies and duplicates. For this reason, it is used only as the extraction source, keeping the original data unchanged.

4.0.2 Staging Area

All extracted data is first loaded into the **Staging Area**, where the information is validate and standardized before entering the **Data Mart**. This layer includes *Staging Tables* with extracted data and *DQP Tables* that store rejected records ensuring data quality.

4.0.3 Data Mart

After validation and transformation, the cleaned data is loaded into the **Data Mart**. It includes dimensions such as Customer, Product, Date, Currency, Shipping and Size, along with the main FactOrders table at order-line level and a FactCurrencyRate table for daily exchange rates.

4.0.4 OLAP and Power BI

In the final layer, the processed data is already available for analysis through the **OLAP** structure and **Power BI tools**.

The **OLAP (Online Analytical Processing)** component organizes the data into analytical cubes in order to support advanced analysis.

Power BI also connects directly to the Data Mart to develop dashboards and reports, supporting analytical tasks, such as:

- Sales trends over time (year/quarter/month/day) and across currencies (local vs EUR);
- Performance by product or site;
- Customer profile breakdowns by gender, country and VIP status.
- Shipping behavior and distribution of logistics costs by shipping country.
- Weekend orders and size-based scenarios, such as products with maximum size 52, detailed by site country;

5 Data Structure

5.0.1 Staging Area

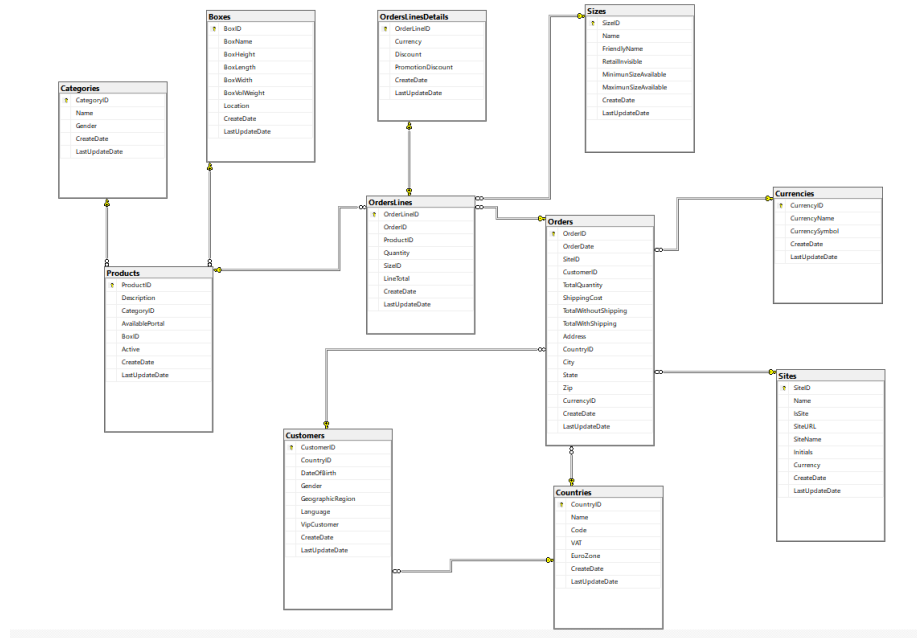


Figure 1: Staging Area relational model

The figure above represents the Staging Area dimensional model. The relations between the main entities were kept with the following changes:

- The SitesInfo and ProductsVideos tables were removed, as they do not contain attributes relevant for analytical purposes in the context of customer orders.
- In the Sizes table, the ranges were retained, but the detailed information listing all available sizes was removed, as it was impossible to reference the specific size the customer is purchasing.

In the Staging Area, in addition to the standard staging tables, a set of Data Quality Problem (DQP) tables are created to store records that fail validation rules during the ETL process. These tables follow the naming convention of the table and replicate the structure of the corresponding staging table, with an additional attribute, describing the type of error identified.

Table 1: CustomerDQP

Attribute	Data type
CustomerKey	INT(10) - PK
CustomerID	INT(10)
CustomerBirthday	DATE
Gender	CHAR(1)
IsVipCustomer	TINYINT(1)
Country	VARCHAR(50)
EffectiveDate	DATE
ExpiredDate	DATE
IsCurrent	TINYINT(1)
DQP	NVARCHAR(100)

Table 2: ProductDQP

Attribute	Data type
ProductKey	INT(10) - PK
ProductID	INT(10)
ProductDescription	VARCHAR(255)
IsActive	TINYINT(1)
AvailablePortal	TINYINT(1)
CategoryName	VARCHAR(50)
Gender	CHAR(1)
EffectiveDate	DATE
ExpiredDate	DATE
IsCurrent	TINYINT
DQP	NVARCHAR(100)

Table 3: BoxDQP

Attribute	Data type
BoxKey	INT(10) - PK
BoxID	INT(10)
BoxName	VARCHAR(50)
Height	DECIMAL(2, 2))
Width	DECIMAL(2, 2))
Length	DECIMAL(2, 2))
VolWeight	DECIMAL(2, 2))
LocationCode	CHAR(3)
EffectiveDate	DATE
ExpiredDate	DATE
IsCurrent	TINYINT(1)
DQP	NVARCHAR(100)

Table 4: SizeDQP

Attribute	Data type
SizeKey	INT(10) - PK
SizeID	INT(10)
SizeName	VARCHAR(50)
Min	TINYINT(2)
Max	TINYINT(2)
EffectiveDate	DATE
ExpiredDate	DATE
IsCurrent	TINYINT(1)
DQP	NVARCHAR(100)

Table 5: CurrencyDQP

Attribute	Data type
CurrencyKey	INT(10) - PK
CurrencyCode	CHAR(3)
CurrencyName	VARCHAR(10)
DQP	NVARCHAR(100)

Table 6: ShppingDQP

Attribute	Data type
ShippingKey	INT(10) - PK
CountryCode	CHAR(3)
CountryName	VARCHAR(20)

Continuation of Table 6	
Attribute	Data type
IsEuroZone	VARCHAR(3)
VAT	TINYINT(3)
EffectiveDate	DATE
ExpiredDate	DATE
IsCurrent	TINYINT(1)
DQP	NVARCHAR(100)

Table 7: SiteDQP

Attribute	Data type
SiteKey	INT(10) - PK
SiteID	INT(10)
SiteName	VARCHAR(50)
SiteURL	VARCHAR(50)
SiteInitials	CHAR(3)
IsSite	TINYINT(1)
EffectiveDate	DATE
ExpiredDate	DATE
IsCurrent	TINYINT(1)
DQP	NVARCHAR(100)

Table 8: FactOrdersDQP

Attribute	Data type
DateKey	INT(10)
ShippingKey	INT(10)
SiteName	INT(10)
CustomerKey	INT(10)
ProductKey	INT(10)
OrderID	INT(10)
SizeKey	INT(10)
BoxKey	INT(10)
LocalCurencyKey	INT(10)
QuantityOrdered	INT(10)
UnitPriceLocal	INT(10)
DiscountLocal	INT(10)
TotalAmountLocal	INT(10)
DQP	NVARCHAR(100)

Table 9: FactCurrencyDQP

Attribute	Data type
ConversionDateKey	INT(10)
SourceCurrencyKey	INT(10)
DestinationCurrencyKey	INT(10)
SourceDestinationExchangeRate	DECIMAL(10, 4)
DestinationSourceExchangeRate	DECIMAL(10, 4)
DQP	NVARCHAR(100)

6 Transformation operations

The data transformation stage is useful to prepare the extracted data to be loaded into the Data Mart, ensuring quality, consistency and an appropriate analytical structure.

6.0.1 Standardization of Formats

- Date fields: all date attributes are converted to the ISO standard format **YYYY-MM-DD**
- Currency values: All the monetary values (unit price, total amount, discounts...) with different currencies are standardized. When applicable, values are converted to *EUR* using the corresponding exchange rate stored in *FactCurrencyRate*.
- Text normalization: Standardization of product, category, country and site names to ensure consistent format. Removal of leading/trailing spaces, duplicate whitespace or inconsistent casing (e.g: "PT", "pt" -> Both mean "Portugal"). Normalization of gender information in DimCustomer (e.g: "M", "Male" and "male" -> mean "Male").
- Boolean values: Boolean values are standardized to a numeric format, **True** = **1** and **False** = **0**. This can be applied to attributes such as *IsActive*, *AvailablePortal*, *IsSite*, *IsVIPCustomer* and *IsCurrent*.

7 Clean operations

After the transformation stage, the cleaning stage focuses on correcting invalid values and removing inconsistencies, in order to better prepare the data to be loaded into the Data Mart.

7.0.1 Duplicate Removal

- Identification of duplicates records using business keys or attributes combinations. Examples: 2 products with different ProductID but same de-

scription, category and size -> ProductID 300("Black Jacket 123") and ProductID 500("Black Jacket 123"). The duplicate, in this case the id 500, should be sent to DQP in order to be removed from the order.

- Removal of redundant entries in tables.

7.0.2 Validation rules

- Numerical values: Negative UnitPrice, Discount and QuantityOrdered are considered invalid and they are redirected to the respective DQP tables.
- The date of a shipping should be equal to or later than the order date.
- Standardization of inconsistent country names, currencies or product categories to ensure uniform analysis.

8 SCD - Slowly Changing Dimension

Different Slowly Changing Dimension strategies were applied to the dimensions of the project, in order to preserve historical information.

8.0.1 SCD Type 1 - Overwrite

DimCustomer

- Gender: corrections do not need historical tracking;
- CustomerBirthday: date of birth does not change, correction overwrite errors.

DimProduct

- ProductDescription: description changes require historical tracking
- Gender: corrections do not need historical tracking.

DimBox

- BoxName: name corrections overwrite previous values;

DimSite

- SiteName, SiteURL, SiteInitials: changes can be considered as corrections, where the historic is not important.
- IsActive: only the current state of the site matters.

DimShipping

- CountryName: naming correction do not need historic tracking.
- VAT: updated tax values overwrite previous ones

- IsEuroZone: classification updates overwrite the previous value.

DimSize

- SizeName, Min, Max: sizing attributes may be corrected, it is not necessary track older versions

DimCurrency

- CurrencyName, CurrencyCode: currency attributes rarely change, meaning corrections always overwrite previous values.

The ID's of each dimension are a unique and immutable identifier, meaning there are no need to keep a history, as the value does not change.

8.0.2 SCD Type 2 - Retain History

DimCustomer

- Country: changes in customer location are often relevant, especially for geographic behavior analysis.
- IsVipCustomer: important to analyse behavior before and after becoming a Vip customer.

DimProduct

- CategoryName: reclassifying a product into a new category impacts analysis.
- IsActive: only the current activation state matters;
- AvailablePortal: analysis focuses on current portal availability;

DimBox

- Height, Width, Length, VolumeWeight: physical adjustments can affect shipping cost history
- LocationCode: the historic tracker is important for analyses.

DimShipping

- CountryCode: naming correction do not need historic tracking.

8.0.3 SCD Type 3 - Retain valid alternative values

Not implemented in this model because none of the dimensions require retaining both the current and previous values and no attributes exist that support this pattern.

8.0.4 SCD Type 4 - New history table

Not used in this project, however, would apply to a very high-volume attribute changes.

8.0.5 SCD Type 6 - Hybrid approach

Not specified in this project.

9 Mapping

The mapping table represented below shows the correspondence between the source *PlusOrders* database, the *Staging Area* and the *Data Mart*, including the target attributes, data types, their SCD type in each dimension and the specific transformations executed during ETL process.

Target - DataMart Orders					Staging - PlusOrders StagingArea			Source - PlusOrders			Transformation
Table	Column	DataType	TableType	SCD Type	Table	Column	DataType	Table	Column	DataType	
DimCustomer	CustomerKey	Integer	Dimension		Customers	CustomerID	Integer	Customers	CustomerID	Integer	Surrogate key (DW generated identity)
DimCustomer	CustomerID	Integer	Dimension		Customers	DateOfBirth	Nvarchar(255)	Customers	DateOfBirth	Nvarchar(255)	Natural ID for customer in PlusOrders system
DimCustomer	CustomersBirthDay	Date	Dimension	1	Customers	Gender	nchar	Customers	Gender	nchar	Convert to Date
DimCustomer	Gender	Char	Dimension	1	Customers	VipCustomer	bit	Customers	VipCustomer	bit	Direct mapping
DimCustomer	IsVipCustomer	Tinyint(1)	Dimension	2	Customers	Name	Nvarchar(255)	Countries	Name	Nvarchar(255)	Direct mapping
DimCustomer	Country	VarChar(50)	Dimension	2	Countries						SELECT co.Name FROM Customers cu JOIN Countries co ON cu.CountryID = co.CountryID;
DimSite	SiteKey	Integer	Dimension		Site	SiteID	Integer	Site	SiteID	Integer	Surrogate key (DW generated identity)
DimSite	SiteID	Integer	Dimension		Site	Name	Nvarchar(255)	Site	Name	Nvarchar(255)	Natural ID for customer in PlusOrders system
DimSite	SiteName	VarChar(50)	Dimension	1	Site	SiteURL	Nvarchar(255)	Site	SiteURL	Nvarchar(255)	Direct mapping
DimSite	SiteURL	VarChar(50)	Dimension	2	Site	Initials	Nvarchar(3)	Site	Initials	Nvarchar(3)	Direct mapping
DimSite	SiteInitials	Char(3)	Dimension	2	Site	IsSite	bit	Site	IsSite	bit	Direct mapping
DimSite	IsSite	Tinyint(1)	Dimension	1	Site						
DimProduct	ProductKey	Integer	Dimension		Products	ProductID	Integer	Products	ProductID	Integer	Surrogate key (DW generated identity)
DimProduct	ProductID	Integer	Dimension		Products	Description	Nvarchar	Products	Description	Nvarchar	Natural ID for customer in PlusOrders system
DimProduct	ProductDescription	VarChar(255)	Dimension	1	Products	Active	bit	Products	Active	bit	Direct mapping
DimProduct	IsActive	Tinyint(1)	Dimension	2	Products	AvailablePortal	bit	Products	AvailablePortal	bit	Direct mapping
DimProduct	AvailablePortal	Tinyint(1)	Dimension	2	Products	Name	Nvarchar	Categories	Name	Nvarchar	Direct mapping
DimProduct	CategoryName	VarChar(50)	Dimension	2	Categories	Gender	nchar	Categories	Gender	nchar	SELECT c.Name FROM Categories c JOIN Products p ON c.CategoryID = p.CategoryID;
DimProduct	Gender	Char	Dimension	1	Categories						SELECT c.Gender FROM Categories c JOIN Products p ON c.CategoryID = p.CategoryID;
DimBox	BoxKey	Integer	Dimension		Boxes	BoxID	Integer	Boxes	BoxID	Integer	Surrogate key (DW generated identity)
DimBox	BoxID	Integer	Dimension		Boxes	BoxName	Nvarchar(255)	Boxes	BoxName	Nvarchar(255)	Natural ID for customer in PlusOrders system
DimBox	BoxName	VarChar(50)	Dimension	1	Boxes	BoxHeight	float	Boxes	BoxHeight	float	Direct mapping
DimBox	Height	Decimal	Dimension	2	Boxes	BoxWidth	float	Boxes	BoxWidth	float	Direct mapping
DimBox	Width	Decimal	Dimension	2	Boxes	BoxLength	float	Boxes	BoxLength	float	Direct mapping
DimBox	Length	Decimal	Dimension	2	Boxes	BoxVolWeight	float	Boxes	BoxVolWeight	float	Direct mapping
DimBox	VolWeight	Decimal	Dimension	2	Boxes	Location	Nvarchar(255)	Boxes	Location	Nvarchar(255)	Direct mapping
DimBox	LocationCode	Char(3)	Dimension	2	Boxes						
DimSize	SizeKey	Integer	Dimension		Sizes	SizeID	Integer	Sizes	SizeID	Integer	Surrogate key (DW generated identity)
DimSize	SizeID	Integer	Dimension		Sizes	Name	Nvarchar(255)	Sizes	Name	Nvarchar(255)	Natural ID for customer in PlusOrders system
DimSize	SizeName	VarChar(50)	Dimension	1	Sizes	MinimumSizeAvailable	Integer	Sizes	MinimumSizeAvailable	Integer	Direct mapping
DimSize	Min	Tinyint(2)	Dimension	2	Sizes	MaximumSizeAvailable	Integer	Sizes	MaximumSizeAvailable	Integer	Direct mapping
DimSize	Max	Tinyint(2)	Dimension	2	Sizes						
DimShipping	ShippingKey	Integer	Dimension		Countries	Code	Nvarchar(2)	Countries	Code	Nvarchar(2)	Surrogate key (DW generated identity)
DimShipping	CountryCode	Char	Dimension	1	Countries	Name	Nvarchar(255)	Countries	Name	Nvarchar(255)	Direct mapping
DimShipping	CountryName	VarChar	Dimension	1	Countries	EuroZone	bit	Countries	EuroZone	bit	Direct mapping
DimShipping	IsEuroZone	Tinyint(1)	Dimension	1	Countries	VAT	float	Countries	VAT	float	Direct mapping
DimShipping	VAT	Decimal	Dimension	1	Countries						
DimCurrency	CurrencyKey	Integer	Dimension		Currencies	CurrencyID	Nvarchar(3)	Currencies	CurrencyID	Nvarchar(3)	Surrogate key (DW generated identity)
DimCurrency	CurrencyCode	Char(3)	Dimension	1	Currencies	CurrencyName	Nvarchar(20)	Currencies	CurrencyName	Nvarchar(20)	Direct mapping
DimCurrency	CurrencyName	VarChar(50)	Dimension	1	Currencies						
FactOrders	DateKey	Integer	Fact	N/A	DimDate (DataMart_Orders), Orders	DateKey	Integer	Orders	OrderDate	Integer	where Orders.OrderDate = dataMart_orders.DimDate.FullDate
FactOrders	ShippingKey	Integer	Fact	N/A	DimShipping (DataMart_Orders), Orders	ShippingKey	Integer	Orders	CountryID	Integer	where c.Name = dataMart_orders.DimShipping.CountryName
FactOrders	SiteKey	Integer	Fact	N/A	DimSite (DataMart_Orders), Orders	SiteKey	Integer	Orders	SiteID	Integer	where s.Name = dataMart_orders.DimSite.SiteName
FactOrders	CustomerKey	Integer	Fact	N/A	DimCustomer (DataMart_Orders), Orders	CustomerKey	Integer	Orders	CustomerID	Integer	where Orders.CustomerID = dataMart_orders.DimCustomer.CustomerID
FactOrders	ProductKey	Integer	Fact	N/A	DimProduct (DataMart_Orders), OrderLines	ProductKey	Integer	OrderLines	ProductID	Integer	where Orders.ProductID = dataMart_orders.DimProduct.ProductID
FactOrders	OrderID	Integer	Fact	N/A	DimOrder (DataMart_Orders), Orders	OrderID	Integer	Orders	OrderID	Integer	where Orders.OrderID = dataMart_orders.FactOrders.OrderID
FactOrders	SizeKey	Integer	Fact	N/A	DimSize (DataMart_Orders), OrderLines	SizeKey	Integer	OrderLines	SizeID	Integer	where s.Name = dataMart_orders.DimSize.SizeName
FactOrders	BoxKey	Integer	Fact	N/A	DimBox (DataMart_Orders), Products	BoxKey	Integer	Products	BoxID	Integer	where b.Name = dataMart_orders.DimBox.BoxName
FactOrders	LocalCurrencyKey	Integer	Fact	N/A	DimCurrency (DataMart_Orders), Currencies	LocalCurrencyKey	Integer	Orders		Integer	where Orders.CurrencyID = dataMart_orders.DimCurrency.currencyCode
FactOrders	QuantityOrdered	Integer	Fact	N/A	OrderLines	Quantity	Integer	OrderLines	Quantity	Integer	OrderLines quantity
FactOrders	UnitPriceLocal	Float	Fact	N/A	OrderLines	LineTotal	Float	OrderLines	LineTotal	Float	OrderLines line total
FactOrders	DiscountLocal	Float	Fact	N/A	OrderLineDetails	Discount	Float	OrderLineDetails	Discount	Float	OrderLineDetails discount
FactOrders	TotalAmountLocal	Float	Fact	N/A							(UnitPriceLocal * QuantityOrdered) / DiscountLocal

Figure 2: Attributes mapping from source to datamart