Data Visualization

By Jakub Poplawski

DEMO 3, GROUP: Wr-43.DB

softserve

Introduction

In the first sprint I was responsible for:

- Stored procedure to generate Commercial Invoices
- Stored procedure to populate Vendors with their products
- Trigger validating VAT tax code upon insert of Vendors into table storing their addresses
- View summarizing Commercial Invoices for a given Warehouse
- Customers' Loyalty Discount program
- Vendors related tables
- Function selecting an appropriate type of Discount on Receipt line
- Table valued function summarizing the information about a selected Member of the Loyalty Discount program

Jakub Popławski

softserve

Introduction

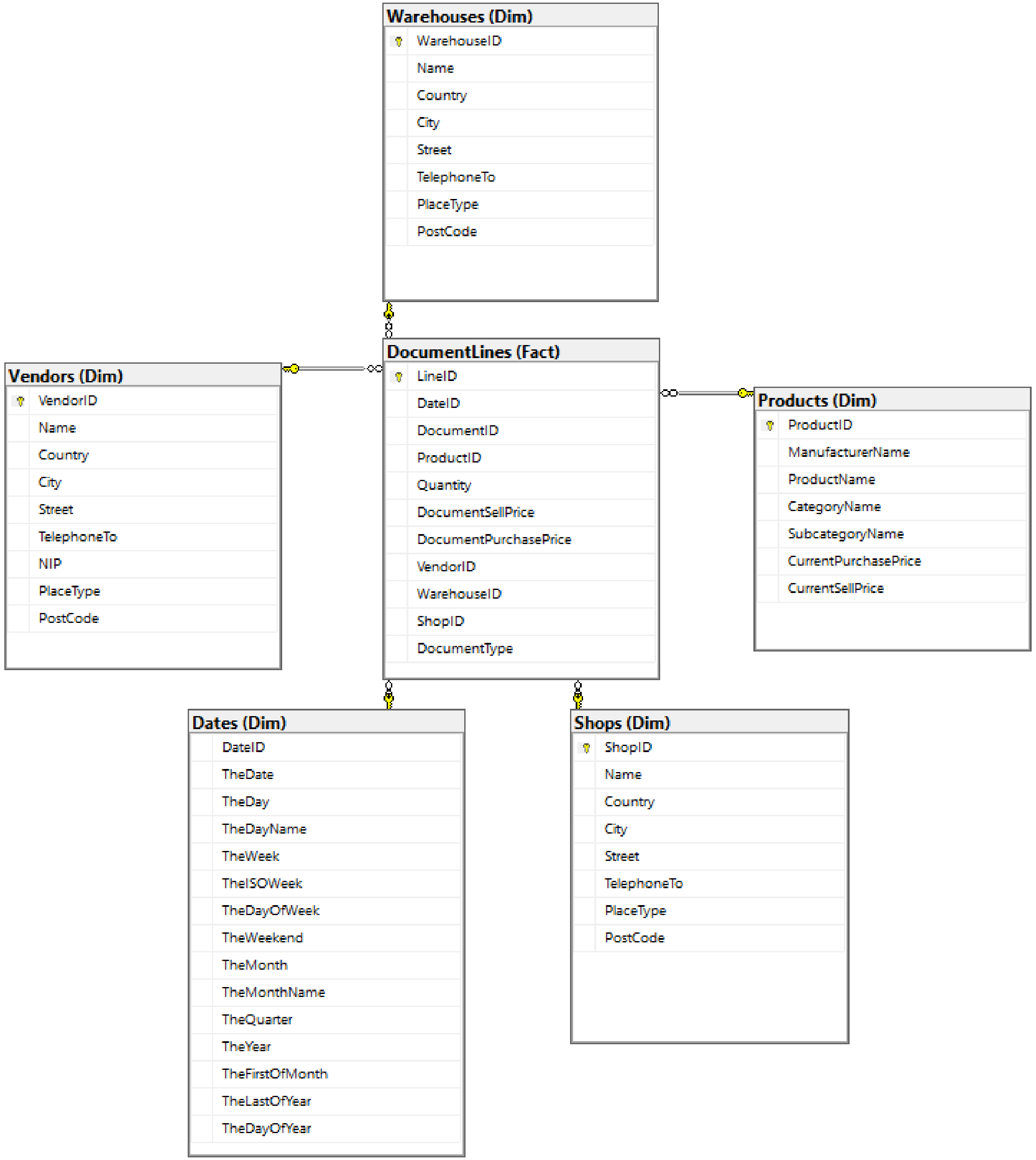
In the second sprint I was responsible for:

- ETL proces to prepare and upload the data from the OLTP database to the OLAP database with an aim to analyse Warehouse stocks

Jakub Popławski

softserve

Introduction



Warehouses (Dim)
- WarehouseID
- Name
- Country
- City
- Street
- TelephoneTo
- PlaceType
- PostCode

Vendors (Dim)
- VendorID
- Name
- Country
- City
- Street
- TelephoneTo
- NIP
- PlaceType
- PostCode

DocumentLines (Fact)
- LineID
- DateID
- DocumentID
- ProductID
- Quantity
- DocumentSellPrice
- DocumentPurchasePrice
- VendorID
- WarehouseID
- ShopID
- DocumentType

Products (Dim)
- ProductID
- ManufacturerName
- ProductName
- CategoryName
- SubcategoryName
- CurrentPurchasePrice
- CurrentSellPrice

Dates (Dim)
- DateID
- TheDate
- TheDay
- TheDayName
- TheWeek
- TheISOWeek
- TheDayOfWeek
- TheWeekend
- TheMonth
- TheMonthName
- TheQuarter
- TheYear
- TheFirstOfMonth
- TheLastOfYear
- TheDayOfYear

Shops (Dim)
- ShopID
- Name
- Country
- City
- Street
- TelephoneTo
- PlaceType
- PostCode

Fig. 1. Scheme of dimensions and fact tables for the Data Warehouse

Assumptions

Organization – the whole business entity of the Buisness Client.

The Buisness Client buys products from Vendors, and stores them in his Warehouses.

He ships them to his Shops where they are sold to Customers.

Assumptions

Product goes to the Warehouse
from Vendor – **Commercial Invoice** is issued.

Product leaves the Warehouse
to Shop – **Freight Invoice** is issued.

Assumptions

Quick example

Tab. 1. Summary of the invoice flow as on 4th January

Product X	Vendor	Warehouse		Shop
		Commercial Invoices	**Freight Invoices**	
02.01.2023		10		
03.01.2023			-3	
04.01.2023		5		
		15	-3	
		BALANCE AS ON **04.01.23:**	**12**	

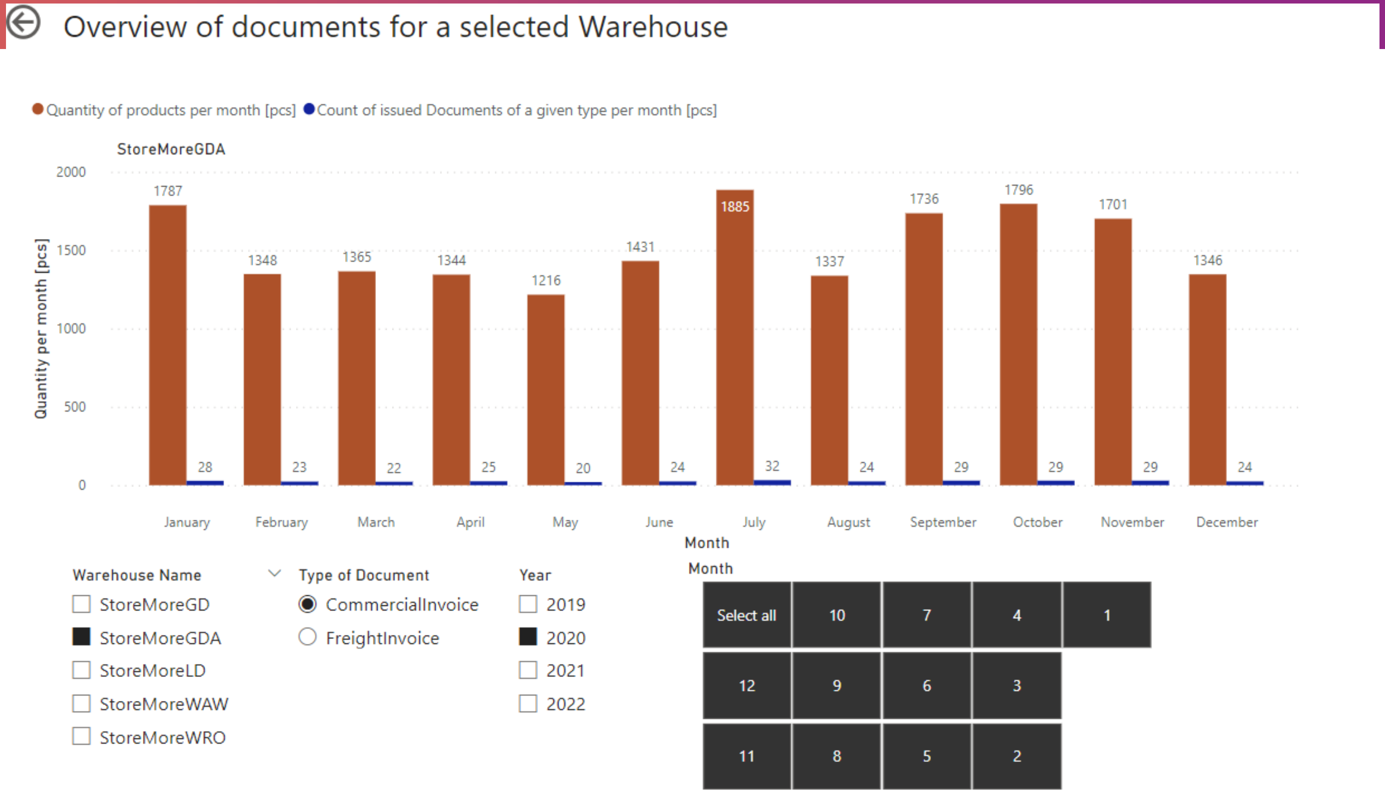
Tab. 2. Summary of the invoice flow as on 6th January

Product X	Vendor	Warehouse		Shop
		Commercial Invoices	**Freight Invoices**	
02.01.2023		10		
03.01.2023			-3	
04.01.2023		5		
05.01.2023			-6	
06.01.2023		5		
		20	-9	
		BALANCE AS ON **06.01.23:**	**11**	

Overview of Documents

The report shows how many documents were issued in a specific month in the selected Warehouse.

It also shows how many products were shipped with them.



Overview of documents for a selected Warehouse

Warehouse Name
□ StoreMoreGD
■ StoreMoreGDA
□ StoreMoreLD
□ StoreMoreWAW
□ StoreMoreWRO

Type of Document
● CommercialInvoice
○ FreightInvoice

Year
□ 2019
■ 2020
□ 2021
□ 2022

Month

Fig. 2. Screenshot of the report summrizing the issued documents

Overview of Documents

The report was based on relationship recreated in Power BI model of the Data Warehouse.

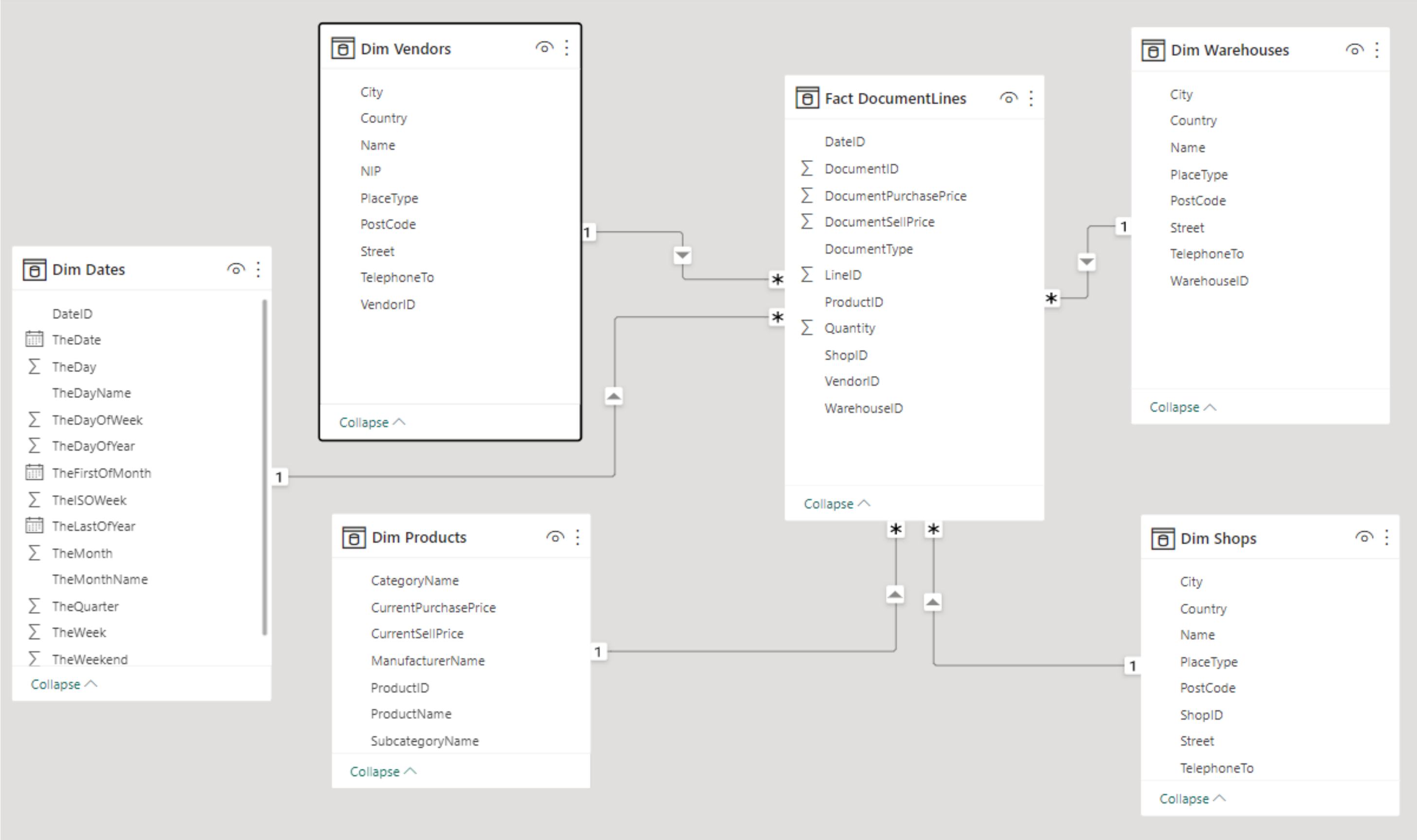
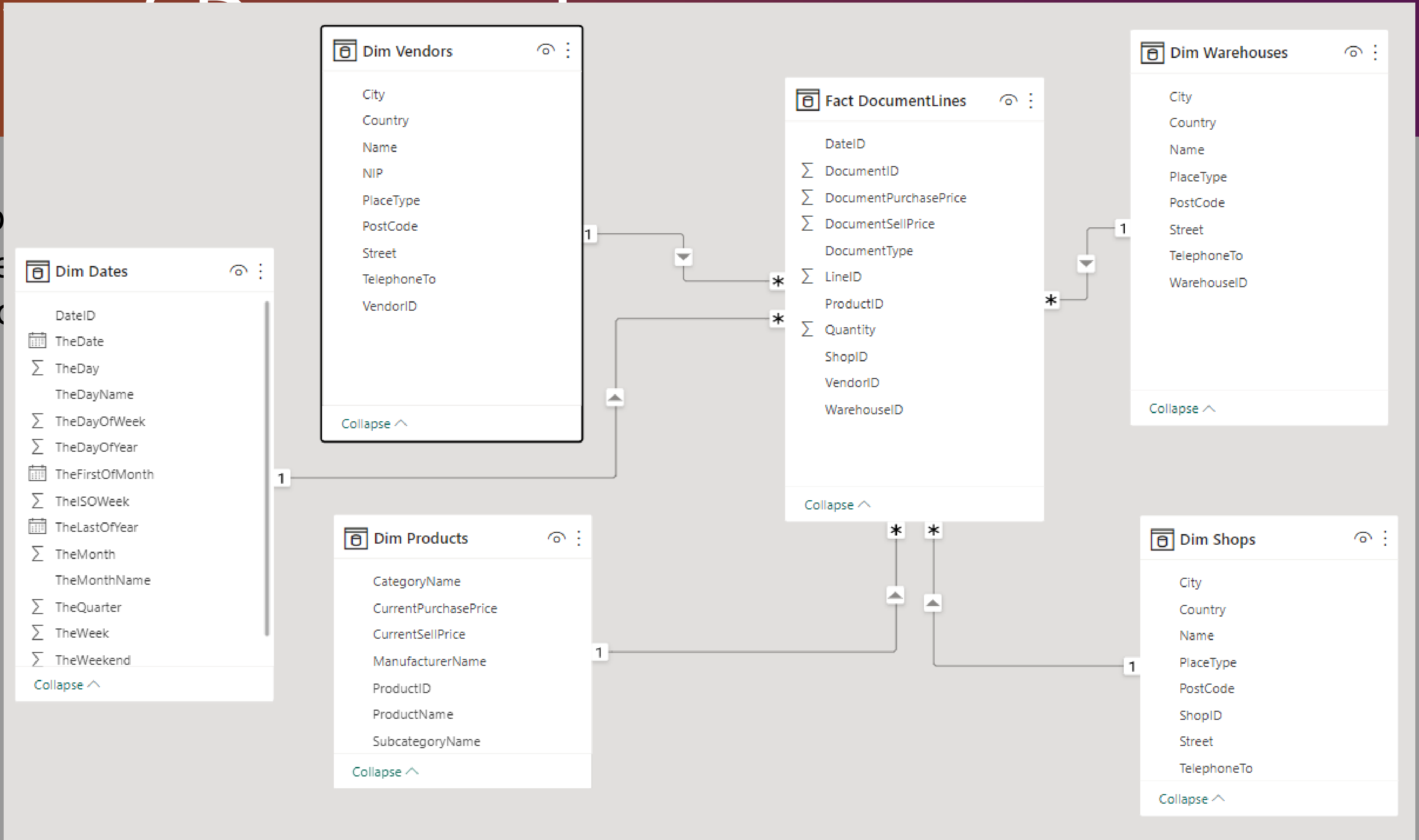


Fig. 3. Power BI data model

Overvie

The report was b
relationship recre
Power BI model c
Warehouse.



Dim Vendors
- City
- Country
- Name
- NIP
- PlaceType
- PostCode
- Street
- TelephoneTo
- VendorID

Dim Dates
- DateID
- TheDate
- TheDay
- TheDayName
- TheDayOfWeek
- TheDayOfYear
- TheFirstOfMonth
- TheISOWeek
- TheLastOfYear
- TheMonth
- TheMonthName
- TheQuarter
- TheWeek
- TheWeekend

Fact DocumentLines
- DateID
- DocumentID
- DocumentPurchasePrice
- DocumentSellPrice
- DocumentType
- LineID
- ProductID
- Quantity
- ShopID
- VendorID
- WarehouseID

Dim Warehouses
- City
- Country
- Name
- PlaceType
- PostCode
- Street
- TelephoneTo
- WarehouseID

Dim Products
- CategoryName
- CurrentPurchasePrice
- CurrentSellPrice
- ManufacturerName
- ProductID
- ProductName
- SubcategoryName

Dim Shops
- City
- Country
- Name
- PlaceType
- PostCode
- ShopID
- Street
- TelephoneTo

Fig. 3. Power BI data model

Overview of Documents

The report was based on relationship recreated in Power BI model of the Data Warehouse.

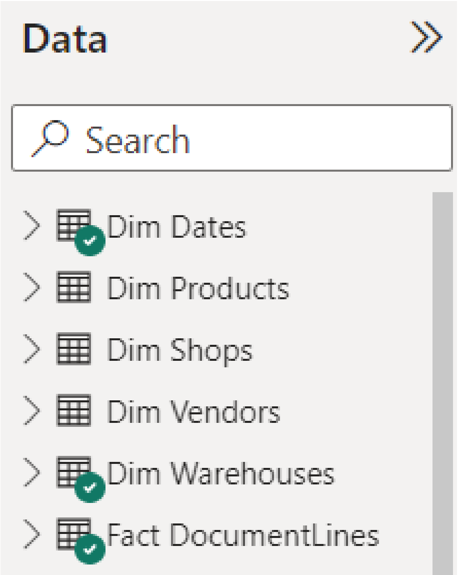
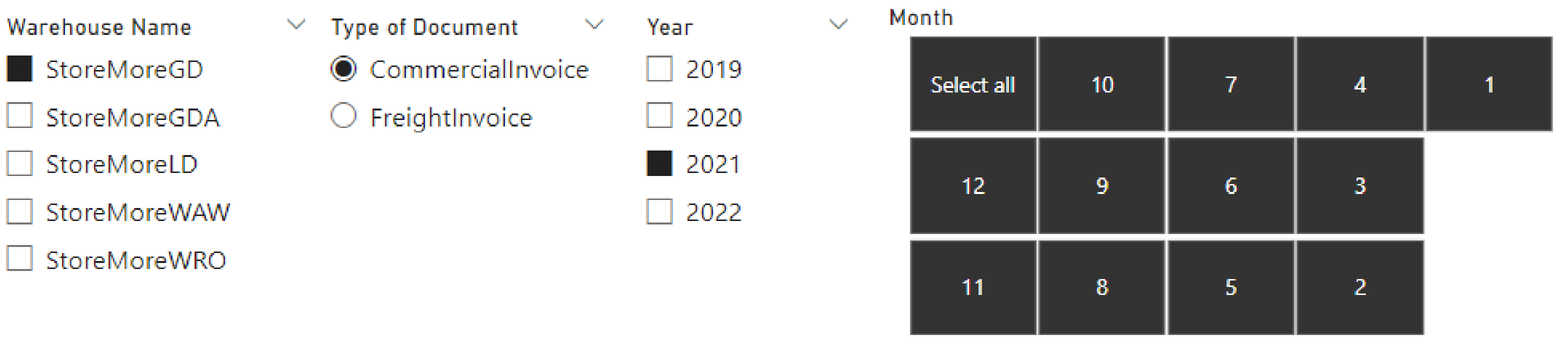


Fig. 4. Data selected for the report from the model

Overview of Documents

Quantity of products from documents and the number of documents were added in batches per month and year, and filtered by DocumentType attribute, Warehouse.



Warehouse Name
- ☑ StoreMoreGD
- □ StoreMoreGDA
- □ StoreMoreLD
- □ StoreMoreWAW
- □ StoreMoreWRO

Type of Document
- ● CommercialInvoice
- ○ FreightInvoice

Year
- □ 2019
- □ 2020
- ☑ 2021
- □ 2022

Month

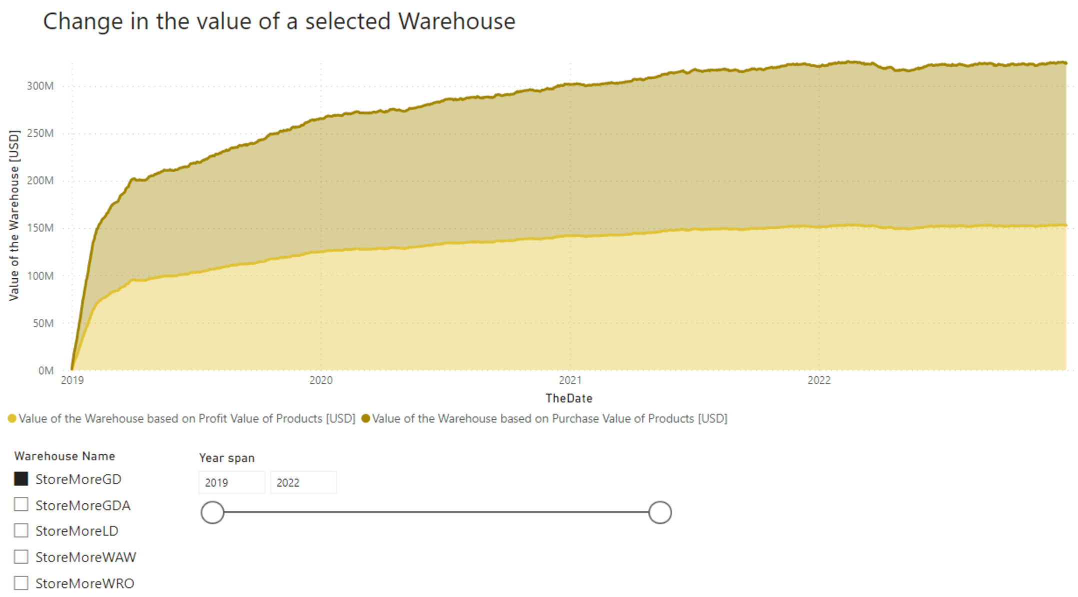
Select all	10	7	4	1
12	9	6	3	
11	8	5	2	

Fig. 5. Filters applied to the visualisation

Change in Value

The report tells how the value of products stored in the warehouse have changed over time.

The value was expressed in Profit Value of the stored product (more important for sale), and in Purchase Value (more important from aquiring goods perspective).


Change in the value of a selected Warehouse

Value of the Warehouse [USD]
300M
250M
200M
150M
100M
50M
0M
2019
2020
2021
2022
TheDate
● Value of the Warehouse based on Profit Value of Products [USD]
● Value of the Warehouse based on Purchase Value of Products [USD]

Warehouse Name

- ■ StoreMoreGD
- □ StoreMoreGDA
- □ StoreMoreLD
- □ StoreMoreWAW
- □ StoreMoreWRO

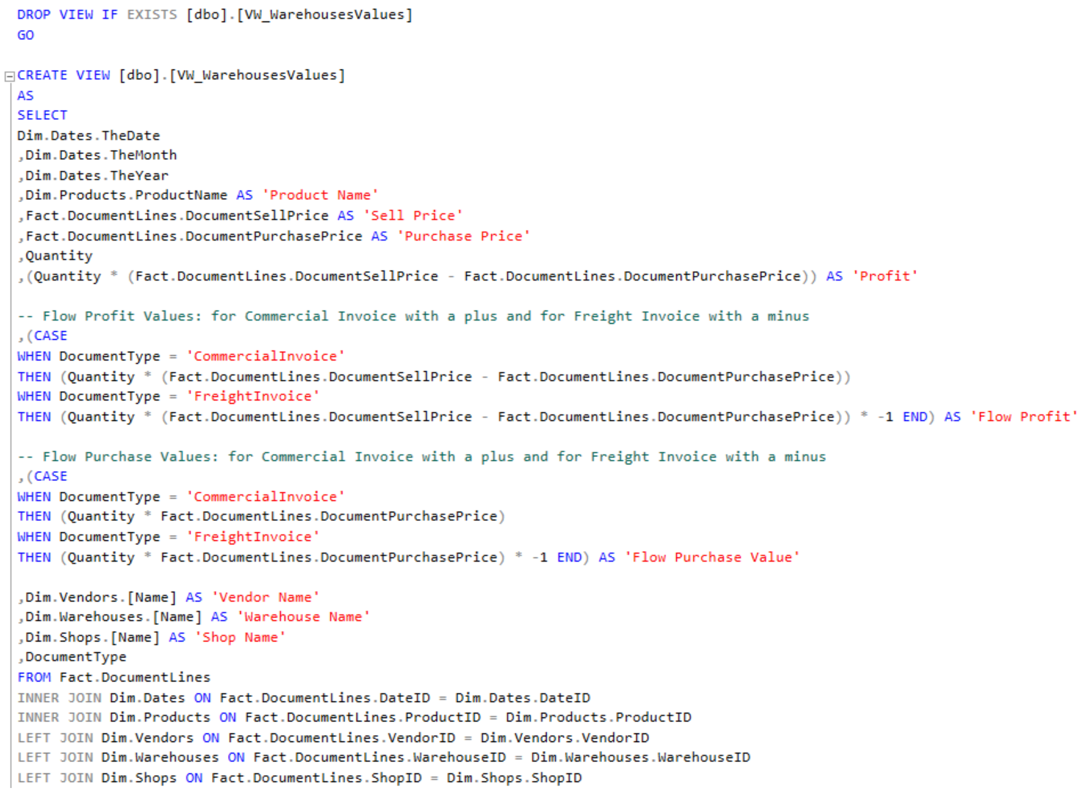
Year span

2019 2022

Fig. 6. Screenshot of the report showing the value of the Warehouse

Change in Value

The report was based on a view created beforhand in SQL.

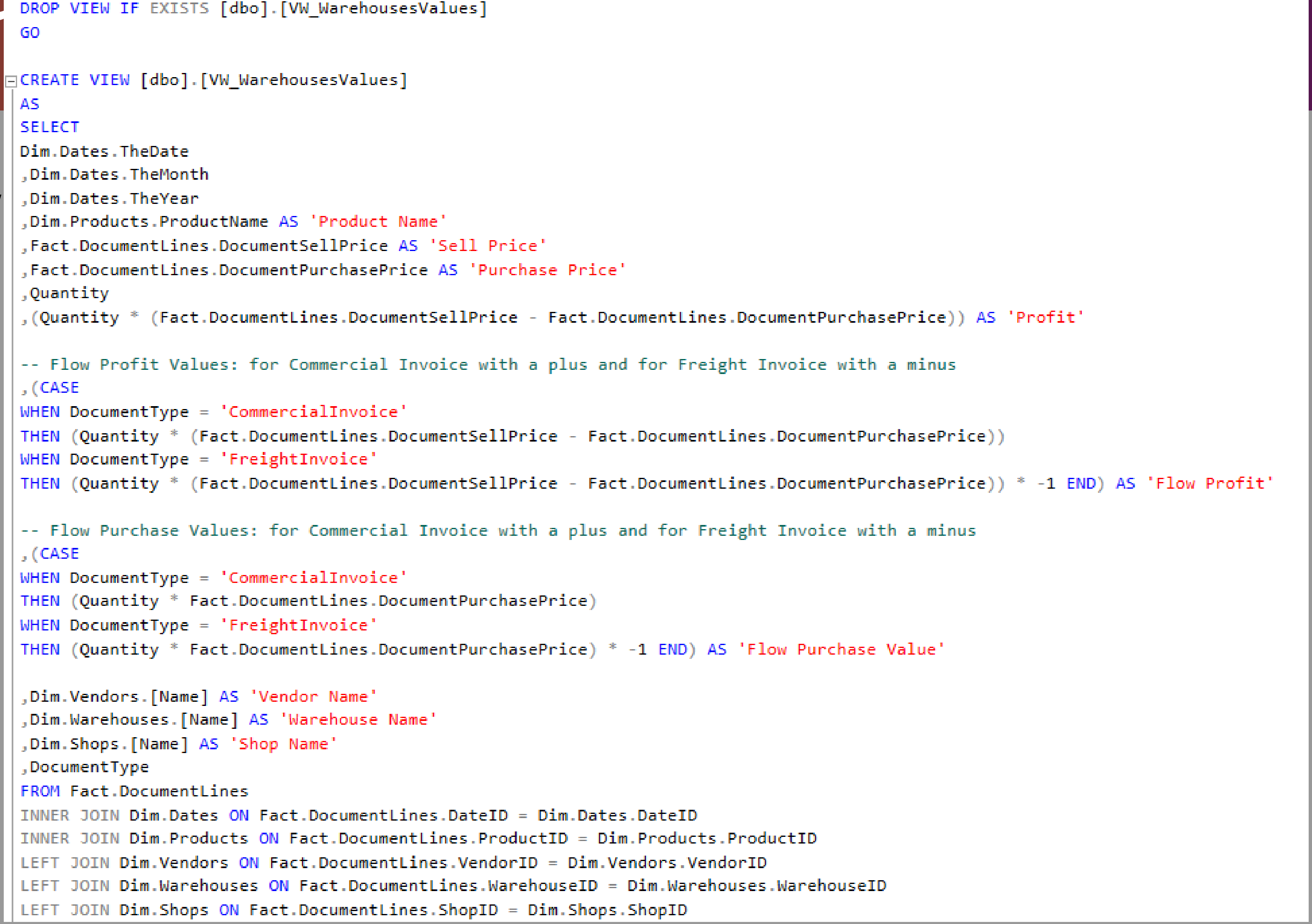


```sql
DROP VIEW IF EXISTS [dbo].[VW_WarehousesValues]
GO

CREATE VIEW [dbo].[VW_WarehousesValues]
AS
SELECT
Dim.Dates.TheDate
,Dim.Dates.TheMonth
,Dim.Dates.TheYear
,Dim.Products.ProductName AS 'Product Name'
,Fact.DocumentLines.DocumentSellPrice AS 'Sell Price'
,Fact.DocumentLines.DocumentPurchasePrice AS 'Purchase Price'
,Quantity
,(Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice)) AS 'Profit'

-- Flow Profit Values: for Commercial Invoice with a plus and for Freight Invoice with a minus
,(CASE
WHEN DocumentType = 'CommercialInvoice'
THEN (Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice))
WHEN DocumentType = 'FreightInvoice'
THEN (Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice)) * -1 END) AS 'Flow Profit'

-- Flow Purchase Values: for Commercial Invoice with a plus and for Freight Invoice with a minus
,(CASE
WHEN DocumentType = 'CommercialInvoice'
THEN (Quantity * Fact.DocumentLines.DocumentPurchasePrice)
WHEN DocumentType = 'FreightInvoice'
THEN (Quantity * Fact.DocumentLines.DocumentPurchasePrice) * -1 END) AS 'Flow Purchase Value'

,Dim.Vendors.[Name] AS 'Vendor Name'
,Dim.Warehouses.[Name] AS 'Warehouse Name'
,Dim.Shops.[Name] AS 'Shop Name'
,DocumentType
FROM Fact.DocumentLines
INNER JOIN Dim.Dates ON Fact.DocumentLines.DateID = Dim.Dates.DateID
INNER JOIN Dim.Products ON Fact.DocumentLines.ProductID = Dim.Products.ProductID
LEFT JOIN Dim.Vendors ON Fact.DocumentLines.VendorID = Dim.Vendors.VendorID
LEFT JOIN Dim.Warehouses ON Fact.DocumentLines.WarehouseID = Dim.Warehouses.WarehouseID
LEFT JOIN Dim.Shops ON Fact.DocumentLines.ShopID = Dim.Shops.ShopID
```

Fig. 7. View used for the value of the Warehouse visualisation

Change in Value

The report was based on a view in SQL.

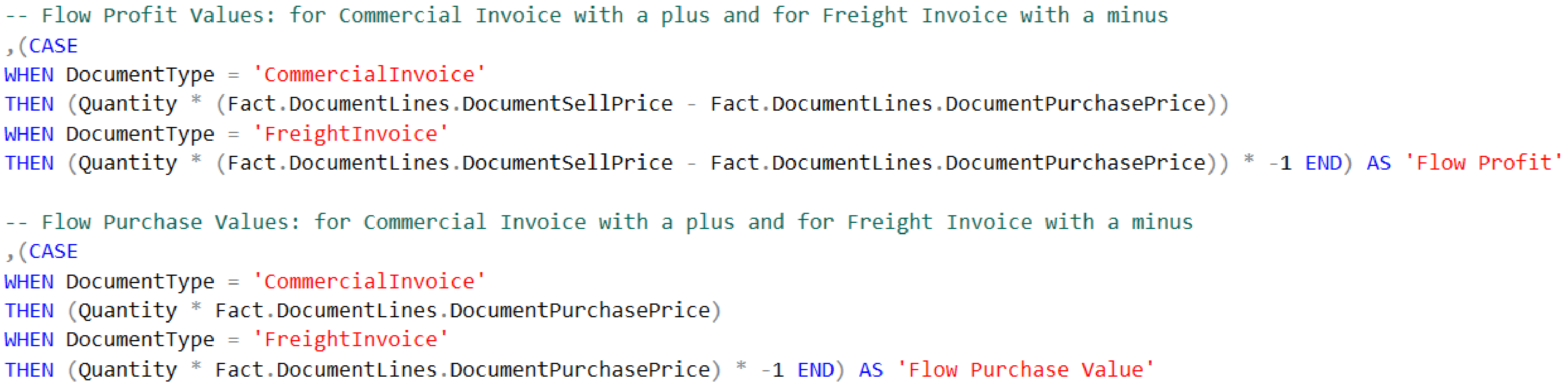


```sql
DROP VIEW IF EXISTS [dbo].[VW_WarehousesValues]
GO

CREATE VIEW [dbo].[VW_WarehousesValues]
AS
SELECT
Dim.Dates.TheDate
,Dim.Dates.TheMonth
,Dim.Dates.TheYear
,Dim.Products.ProductName AS 'Product Name'
,Fact.DocumentLines.DocumentSellPrice AS 'Sell Price'
,Fact.DocumentLines.DocumentPurchasePrice AS 'Purchase Price'
,Quantity
,(Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice)) AS 'Profit'

-- Flow Profit Values: for Commercial Invoice with a plus and for Freight Invoice with a minus
,(CASE
WHEN DocumentType = 'CommercialInvoice'
THEN (Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice))
WHEN DocumentType = 'FreightInvoice'
THEN (Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice)) * -1 END) AS 'Flow Profit'

-- Flow Purchase Values: for Commercial Invoice with a plus and for Freight Invoice with a minus
,(CASE
WHEN DocumentType = 'CommercialInvoice'
THEN (Quantity * Fact.DocumentLines.DocumentPurchasePrice)
WHEN DocumentType = 'FreightInvoice'
THEN (Quantity * Fact.DocumentLines.DocumentPurchasePrice) * -1 END) AS 'Flow Purchase Value'

,Dim.Vendors.[Name] AS 'Vendor Name'
,Dim.Warehouses.[Name] AS 'Warehouse Name'
,Dim.Shops.[Name] AS 'Shop Name'
,DocumentType
FROM Fact.DocumentLines
INNER JOIN Dim.Dates ON Fact.DocumentLines.DateID = Dim.Dates.DateID
INNER JOIN Dim.Products ON Fact.DocumentLines.ProductID = Dim.Products.ProductID
LEFT JOIN Dim.Vendors ON Fact.DocumentLines.VendorID = Dim.Vendors.VendorID
LEFT JOIN Dim.Warehouses ON Fact.DocumentLines.WarehouseID = Dim.Warehouses.WarehouseID
LEFT JOIN Dim.Shops ON Fact.DocumentLines.ShopID = Dim.Shops.ShopID
```

Fig. 7. View used for the value of the Warehouse visualisation

Change in Value

Profit and Purchase Price columns were prepared for input to the cummulative sum DAX function in Power BI.



```
-- Flow Profit Values: for Commercial Invoice with a plus and for Freight Invoice with a minus
,(CASE
WHEN DocumentType = 'CommercialInvoice'
THEN (Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice))
WHEN DocumentType = 'FreightInvoice'
THEN (Quantity * (Fact.DocumentLines.DocumentSellPrice - Fact.DocumentLines.DocumentPurchasePrice)) * -1 END) AS 'Flow Profit'

-- Flow Purchase Values: for Commercial Invoice with a plus and for Freight Invoice with a minus
,(CASE
WHEN DocumentType = 'CommercialInvoice'
THEN (Quantity * Fact.DocumentLines.DocumentPurchasePrice)
WHEN DocumentType = 'FreightInvoice'
THEN (Quantity * Fact.DocumentLines.DocumentPurchasePrice) * -1 END) AS 'Flow Purchase Value'
```

Fig. 8. Case expression used for pre-processing of Profit and Purchase values

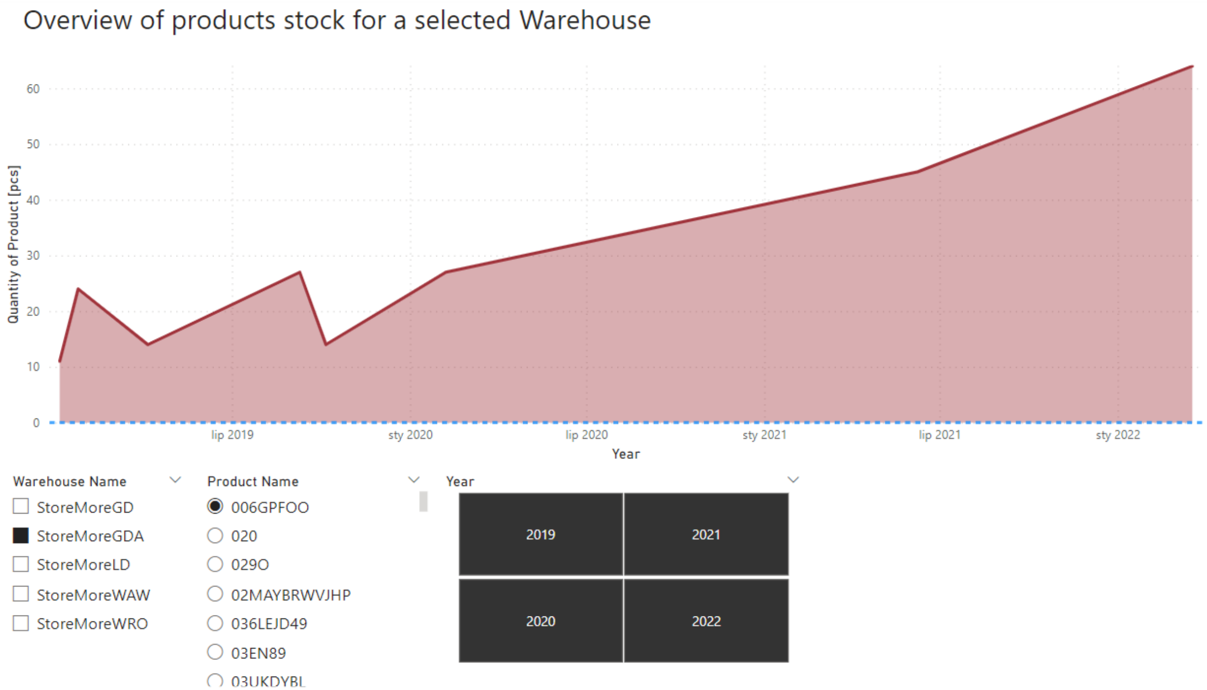
Change in Value

Profit and Purchase Price columns were prepared for input to the cummulative sum DAX function in Power BI.

```
Cummulative Flow Profit Value = SUMX (
    FILTER (
        ALLSELECTED ( VW_WarehousesValues ),
        VW_WarehousesValues[TheDate] <= MAX (
VW_WarehousesValues[TheDate] )
    ),
    VW_WarehousesValues[Flow Profit]
)
```

Overview of Product stock

This report shows how stock of a selected product has changed over time in a selected Warehouse.



Overview of products stock for a selected Warehouse

Warehouse Name

□ StoreMoreGD
■ StoreMoreGDA
□ StoreMoreLD
□ StoreMoreWAW
□ StoreMoreWRO

Product Name

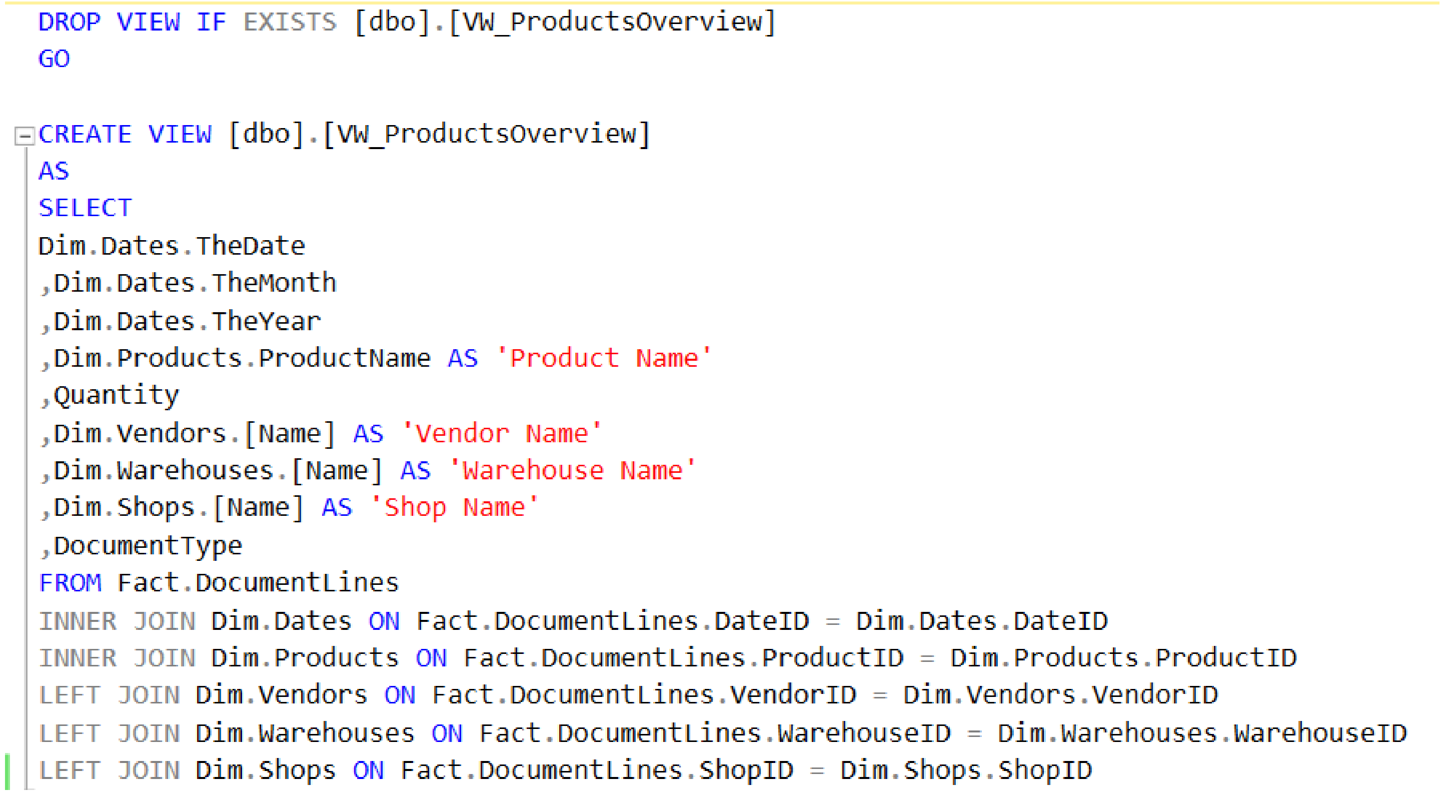
● 006GPFOO
○ 020
○ 029O
○ 02MAYBRWVJHP
○ 036LEJD49
○ 03EN89
○ 03UKDYBL

Year

Fig. 9. Screenshot of the report summarizing stock flow in a Warehouse

Overview of Product stock

This report was also based on an import of a view into the Power BI model.



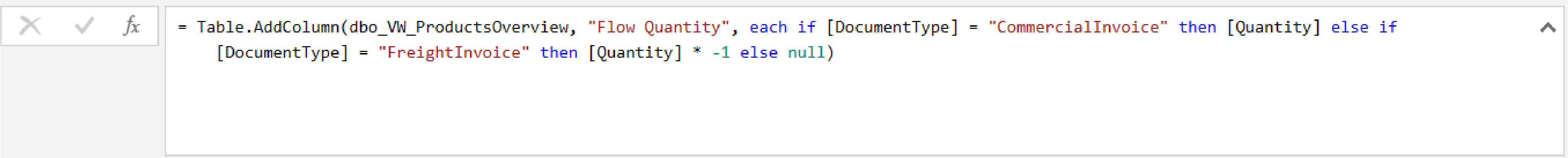
```sql
DROP VIEW IF EXISTS [dbo].[VW_ProductsOverview]
GO

CREATE VIEW [dbo].[VW_ProductsOverview]
AS
SELECT
Dim.Dates.TheDate
,Dim.Dates.TheMonth
,Dim.Dates.TheYear
,Dim.Products.ProductName AS 'Product Name'
,Quantity
,Dim.Vendors.[Name] AS 'Vendor Name'
,Dim.Warehouses.[Name] AS 'Warehouse Name'
,Dim.Shops.[Name] AS 'Shop Name'
,DocumentType
FROM Fact.DocumentLines
INNER JOIN Dim.Dates ON Fact.DocumentLines.DateID = Dim.Dates.DateID
INNER JOIN Dim.Products ON Fact.DocumentLines.ProductID = Dim.Products.ProductID
LEFT JOIN Dim.Vendors ON Fact.DocumentLines.VendorID = Dim.Vendors.VendorID
LEFT JOIN Dim.Warehouses ON Fact.DocumentLines.WarehouseID = Dim.Warehouses.WarehouseID
LEFT JOIN Dim.Shops ON Fact.DocumentLines.ShopID = Dim.Shops.ShopID
```

Fig. 10. View used for the value of the Products stocks visualisation

Overview of products stock

This time the pre-process stage of Quantity value was done via Power Query.

Similar logic to Profit calculations was applied in a new column named Flow Quantity.



```
= Table.AddColumn(dbo_VW_ProductsOverview, "Flow Quantity", each if [DocumentType] = "CommercialInvoice" then [Quantity] else if
    [DocumentType] = "FreightInvoice" then [Quantity] * -1 else null)
```

Fig. 8. Power Query used for the pre-processing of Quantity data

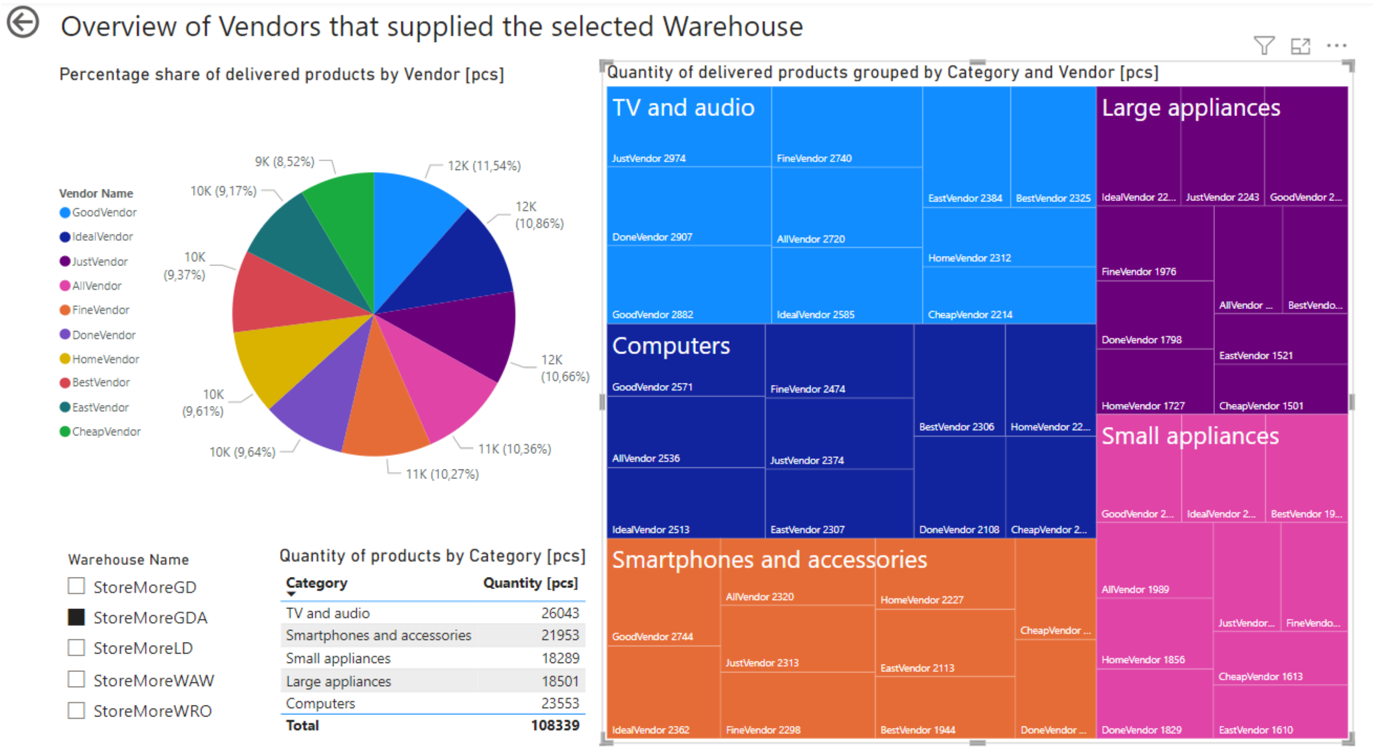
Overview of products stock

Similar cumulative DAX function was used in visualisation.

```
Cummulative Flow Quantity = SUMX (
    FILTER (
        ALLSELECTED ( VW_ProductsOverview ),
        VW_ProductsOverview[TheDate] <= MAX (
VW_ProductsOverview[TheDate] )
    ),
    VW_ProductsOverview[Flow Quantity]
)
```

Overview of Vendors

The report provides an overview on the Quantity and Category structure of products supplied to the selected Warehouse by Vendors.



Overview of Vendors that supplied the selected Warehouse

Warehouse Name

☐ StoreMoreGD
■ StoreMoreGDA
☐ StoreMoreLD
☐ StoreMoreWAW
☐ StoreMoreWRO

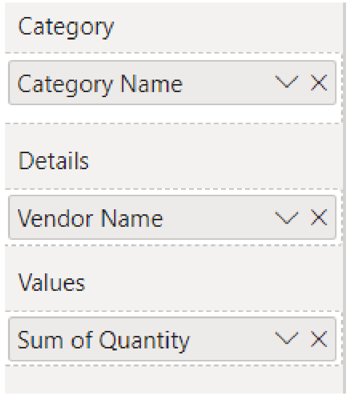
Quantity of products by Category [pcs]

Category	Quantity [pcs]
TV and audio	26043
Smartphones and accessories	21953
Small appliances	18289
Large appliances	18501
Computers	23553
Total	**108339**

Fig. 11. Screenshot of the visualisation of Vendors supply

Overview of Vendors

The report provides an overview on the Quantity and Category structure of products supplied to the selected Warehouse by Vendors.



Category

Category Name

Details

Vendor Name

Values

Sum of Quantity

Fig. 12. Filtering applied to the visualisation

TO THE FUTURE
softserve