Group Members:

* XXX XXX
* XXX XXX
* Répási Gergely
* XXX XXX
* XXX XXX

Data warehouses &   
business intelligence   
group project

Documentation 2022.10.31.

Table of contents

[Use case](#_Toc118219325)  [1](#_Toc118219325)

[Introduction](#_Toc118219326)  [1](#_Toc118219326)

[Questions asked by the customer:](#_Toc118219327)  [1](#_Toc118219327)

[Database layers](#_Toc118219328)  [1](#_Toc118219328)

[Source data](#_Toc118219329)  [1](#_Toc118219329)

[AdventureWorks scheme](#_Toc118219330)  [1](#_Toc118219330)

[Purchasing and its relationships](#_Toc118219331)  [2](#_Toc118219331)

[Staging:](#_Toc118219332)  [2](#_Toc118219332)

[PurchaseOrderHeader](#_Toc118219333)  [3](#_Toc118219333)

[PurchaseOrderDetail](#_Toc118219334)  [3](#_Toc118219334)

[Product](#_Toc118219335)  [3](#_Toc118219335)

[ProductSubCategory](#_Toc118219336)  [4](#_Toc118219336)

[Product Category](#_Toc118219337)  [4](#_Toc118219337)

[Ship Method](#_Toc118219338)  [4](#_Toc118219338)

[Vendor](#_Toc118219339)  [4](#_Toc118219339)

[Data warehouse:](#_Toc118219340)  [4](#_Toc118219340)

[Dimensional model:](#_Toc118219341)  [4](#_Toc118219341)

[Boards](#_Toc118219342)  [5](#_Toc118219342)

[This process](#_Toc118219343)  [5](#_Toc118219343)

[Extraction process](#_Toc118219344)  [5](#_Toc118219344)

[Transformation process](#_Toc118219345)  [6](#_Toc118219345)

[Loading process](#_Toc118219346)  [7](#_Toc118219346)

[REPORT](#_Toc118219347)  [8](#_Toc118219347)

[How did the sales of each product group develop by season?](#_Toc118219348)  [8](#_Toc118219348)

[2. Which product subcategory received the most orders per active merchant?](#_Toc118219349)  [9](#_Toc118219349)

[3. What was the turnover of each trader between 2012 and 2014?](#_Toc118219350)  [9](#_Toc118219350)

[4. What was the average lead time (time from order placement to product arrival) per product, broken down by years?](#_Toc118219351)  [10](#_Toc118219351)

[5. How many orders and what is the value per season and per delivery method in the last 3 years?](#_Toc118219352)  [10](#_Toc118219352)

[6. What was the percentage of rejected products between 2012 and 2014, broken down by product?](#_Toc118219353)  [11](#_Toc118219353)

# Use case

## Introductory

We would like to learn about information related to purchases, the relationship between merchants and products from different perspectives and over time, and the relationship between products and customers. In addition, we will cover basic data related to the delivery and return of individual products, which may have an impact on future sales.

## Questions asked by the customer:

1. How did the sales of each product group develop by season?
2. Which product subcategory received the most orders per active merchant ?
3. What was the turnover of each retailer between 2012 and 2014?
4. What was the average lead time (time from order placement to product arrival) per product, broken down by years?
5. How many orders and what value were placed per season and per delivery method in the last 3 years?
6. What was the percentage of rejected products between 2012 and 2014, broken down by product?

# Database layers

## Source data

The AdventureWorks database as the source of the data.

### AdventureWorks schema

<https://i0.wp.com/improveandrepeat.com/wp-content/uploads/2018/12/AdvWorksOLTPSchemaVisio.png>

Diagram, schematic

Description automatically generated

### Purchasing and its relationships

Diagram

Description automatically generated

## Staging:

The source database tables are unchanged in structure, except that all data types are VARCHAR.

Graphical user interface, diagram

Description automatically generated

### PurchaseOrderHeader

* PurchaseOrderID – Unique identifier of the order
* RevisionNumber - Number used to track changes to an order
* Status - Order status
* ShipDate - Estimated delivery time
* OrderDate - Order date
* ShipMethodID - Shipping method
* VendorID - Seller

### PurchaseOrderDetail

* PurchaseOrderID - Order ID
* PurchaseOrderDetailID - Order ID
* OrderQty – Ordered quantity
* UnitPrice – Selling unit price
* LineTotal - Total price per product
* ReceivedQty - Received quantity
* RejectedQty - Rejected quantity
* StockedQty - Stocked quantity
* DueDate - Expected time of arrival
* ProductID – Product ID

### Product

* ProductID - Product ID
* Name - Product name
* Color - Product color
* Size - Product size
* Weight - Product weight
* ProductNumber - Product number
* ProductSubcategoryID - Product subcategory
* ProductModelID - Product model number

### ProductSubCategory

* ProductSubcategoryID - Product subcategory unique identifier
* Name - Product subcategory name
* ProductCategoryID - Product category foreign key

### Product Category

* ProductCategoryID - Product category unique identifier
* Name - Product category

### ShipMethod

* ShipMethodID - Unique identifier for shipping method
* Name - Shipping company name

### Vendor

* BusinessEntityID - Seller's unique identifier
* AccountNumber - Seller user account number
* Name - Seller name
* CreditRating - Seller rating between 1 and 5
* PreferredVendorStatus - Among multiple vendors, if 1 then we use him, if 0 then we prefer to look for another
* ActiveFlag – 0 if the seller is not active and 1 if the seller is active (default: 1)

## Data warehouse:

It is structurally identical to the tables created in the Staging layer. The difference is that the data types are appropriate for the data, and each table has been supplemented with two date type columns: ValidFrom and ValidTo .

## Dimensional model:

Table

Description automatically generated

### Boards

* FACT\_PURCHASE: Purchase data
* DIM\_PURCHASE: Purchase dimension
* DIM\_PRODUCT: Purchased products
* DIM\_VENDOR: Supplier data
* DIM\_DATE: Date dimension

# This process

## Extraction process

The first part of the ETL process is Extract , in which the Staging layer is populated with data. All data is imported with the VARCHAR(256) data type. No data conversion is performed in this phase, the goal is to achieve the fastest possible transfer.

Graphical user interface, diagram

Description automatically generated

The Extract part was implemented using an Execute SQL Task and a Data Flow Task . The Execute SQL Task is responsible for emptying the Staging layer tables, and after this is successfully completed, the data is loaded from the CSV files.

Graphical user interface

Description automatically generated

## Transformation process

the Transform layer is to convert the Staging content to the appropriate data type and transfer this data to the DW layer.

Graphical user interface

Description automatically generated with medium confidence

To implement it, you need a Foreach Loop Container , which reads the table names from the Staging layer. To properly handle null values, an Execute SQL task , Foreach , is required within the loop. Loop and an Execute SQL Task that converts VARCHAR NULL values to actual values. After this has successfully run, we can load the data into the DW layer using a Data Flow Task .

Graphical user interface

Description automatically generated

The data is read from the Staging layer and then converted to the appropriate type using the Data Conversion task . To avoid duplicate data in the DW layer, we load the data using Slowly Changing Dimension.

## Loading process

Load layer is used to transfer data from the DW layer to the DM , here we use the previously defined data types, and we create a dimension, date and fact table corresponding to the star schema. In the first step, we check whether the date table contains values. If not, we continue the process by filling it. The last two steps are solved with an Execute SQL Task each. The first counts how many pieces of data there are in the Dimension table, if this returns a zero value, the second Execute SQL Task is run .

Diagram

Description automatically generated

To populate the dimension tables, we use views created in the DW layer, from which we read the data into the dimension tables. To populate the tables, we also use Slowly Changing We use Dimension to avoid duplicate data.

Graphical user interface, Word

Description automatically generated

After successfully loading the dimension tables, we can also load the fact table, since this is where we use the data from each dimension. We can connect the tables using Lookup and load them into the FactPurchase table using the SCD we have already used several times .

# REPORT

## 1. How did the sales of each product group develop by season?

Chart, bar chart

Description automatically generated

## 2. Which product subcategory received the most orders per active merchant?

Timeline

Description automatically generated

## 3. What was the turnover of each trader between 2012 and 2014?

Table

Description automatically generated

## 4. What was the average lead time (time from order placement to product arrival) per product, broken down by years?

Graphical user interface

Description automatically generated

## 5. How many orders and what is their value per season and delivery method in the last 3 years?

Chart, line chart

Description automatically generated

## 6. What was the percentage of rejected products between 2012 and 2014, broken down by product?

Chart, pie chart

Description automatically generated

Scrum documentation

Table of contents

[Roles:](#_Toc118290339)  [1](#_Toc118290339)

[Product Backlog:](#_Toc118290340)  [1](#_Toc118290340)

[Sprint Planning Meeting report:](#_Toc118290341)  [1](#_Toc118290341)

[Sprint Backlog:](#_Toc118290342)  [1](#_Toc118290342)

[Daily Scrum report:](#_Toc118290343)  [1](#_Toc118290343)

[Week 1:](#_Toc118290344)  [1](#_Toc118290344)

[Week 2:](#_Toc118290345)  [1](#_Toc118290345)

[Week 3:](#_Toc118290346)  [1](#_Toc118290346)

[Week 4:](#_Toc118290347)  [1](#_Toc118290347)

[Week 5:](#_Toc118290348)  [1](#_Toc118290348)

[Scrum board:](#_Toc118290349)  [1](#_Toc118290349)

[Sprint Retrospect Meeting report:](#_Toc118290350)  [1](#_Toc118290350)

# Roles:

* XXX XXX – Data Warehouse Designer
* XXX XXX – SQL Developer
* XXX XXX – ETL Device Manager
* XXX XXX – Report Developer
* Gergely Répási – Scrum master, product owner

# Product Backlog:

1. use on a subset of the data case formulation, which can be solved by building a data warehouse. Use Presentation of questions to be answered during the case .
2. The use specification of a data model suitable for the case, definition and presentation of the necessary dimensional model (fact table and dimension tables)
3. Creating the database for the 3 ETL layers
4. Preparing and documenting data loading procedures (ETL)
5. Report development, visualization and documentation ( using Power BI)

# Sprint Planning Meeting report:

we would like to use some kind of project management tool during development to continuously track the status of the project, and then we jointly decided that this would be Azure. Let it be DevOps . We chose this because we found it easy to use and transparent, and also because there was someone on the team who was already familiar with this tool, so it was much easier to learn how to use it.

the tasks in the Product Backlog and divided them into smaller subtasks, which were then included in the Sprint Backlog .

# Sprint Backlog:

The image below shows the different Tasks of the Product Backlog. allocated per developer .Graphical user interface, text, application, email

Description automatically generated

# Daily Scrum report:

## Week 1 :

* ATH designer:
  + He figured out the use - case and wrote the necessary questions, 6 in total. Based on the questions, he defined the individual tables and fields of the data model.
  + You will create the fact and dimension tables of the dimensional model with each field.
  + a dimensional model is not entirely clear.
* SQL developer:
  + You will create the Staging and Core layers.

## Week 2:

* ATH designer:
  + The dimensional model is complete.
* SQL developer:
  + Staging and Core layer.
  + In the next step, you will create a Data Mart layer and the queries and views.
* ETL developer:
  + It will prepare the Extract and Load processes.

## Week 3:

* SQL developer:
  + There is also layer 3, or queries and views.
  + Converts NVARCHAR fields to VARCHAR .
* ETL developer:
  + There were issues with Unicode character encoding, which prevented the Transform processes from running. The NVARCHAR data type caused the error.
  + It will make all three layers.

## Week 4:

* SQL developer:
  + Fixed data types in all layers.
* ETL developer:
  + All layers of the ETL are completed and running flawlessly.
* Report developer:
  + It will create reports based on the data you enter.

## Week 5:

* Report developer:
  + The reports have been successfully completed.

# Scrum board:

some of the main phases of the workflow on the Scrumboard.

Roughly the first week's status:Graphical user interface, text, application

Description automatically generated

Approaching week 3, we were about halfway through the Sprint Backlog tasks.A screenshot of a computer

Description automatically generated

Towards the end of week 4, there were still a few tasks left, so it took almost 5 weeks to complete the project.Graphical user interface, application

Description automatically generated

# Sprint Retrospect Meeting report:

It is recommended to use plain VARCHAR from the beginning instead of NVARCHAR data type because it causes problems when loading data.

When saving data to a CSV file, the decimal point used by the database may mess up the data, so it is recommended to use a different delimiter , e.g. semicolon, tab.

During an ETL process, it is only worth moving on to the next task if the current one is working properly and has been verified. If we do not maintain this and skip a faulty task, many retrospective fixes may be required.