Data Mart Implementation (P01)

DECISION SUPPORT SYSTEMS, 2021-22

**João Fernandes (18825), Carlos Martins (18836)**

Barcelos, December 2021

Index

[Introduction 3](#_Toc89957303)

[Data sources 5](#_Toc89957304)

[Dimensional modelling 8](#_Toc89957305)

[Design of the dimensional data model 10](#_Toc89957306)

[Data mart implementation 11](#_Toc89957307)

[Conclusion 14](#_Toc89957308)

[Bibliography 14](#_Toc89957309)

[Appendix A – Data description maps 15](#_Toc89957310)

# Introduction

The scope of this project is to implement a data mart based on an operation database. In this case the database in use is **Adventure Works 2021 database**. This database is from a fictional USA company, that is both a manufacturer and seller of bicycles/accessories.

The company has two main channels: **Internet Sales** and **Retail Sales**.Since we were given the option of choosing one of these two, we chose **Retail Sales**.

**Data Schema Analysis**

AdventureWorks Database is a Microsoft product sample for an online transaction processing (OLTP) database. This Database is also used in code examples found in SQL Server documentation and books. We only used some of the tables from that database. In the following figure we can see what tables we choose to work with.

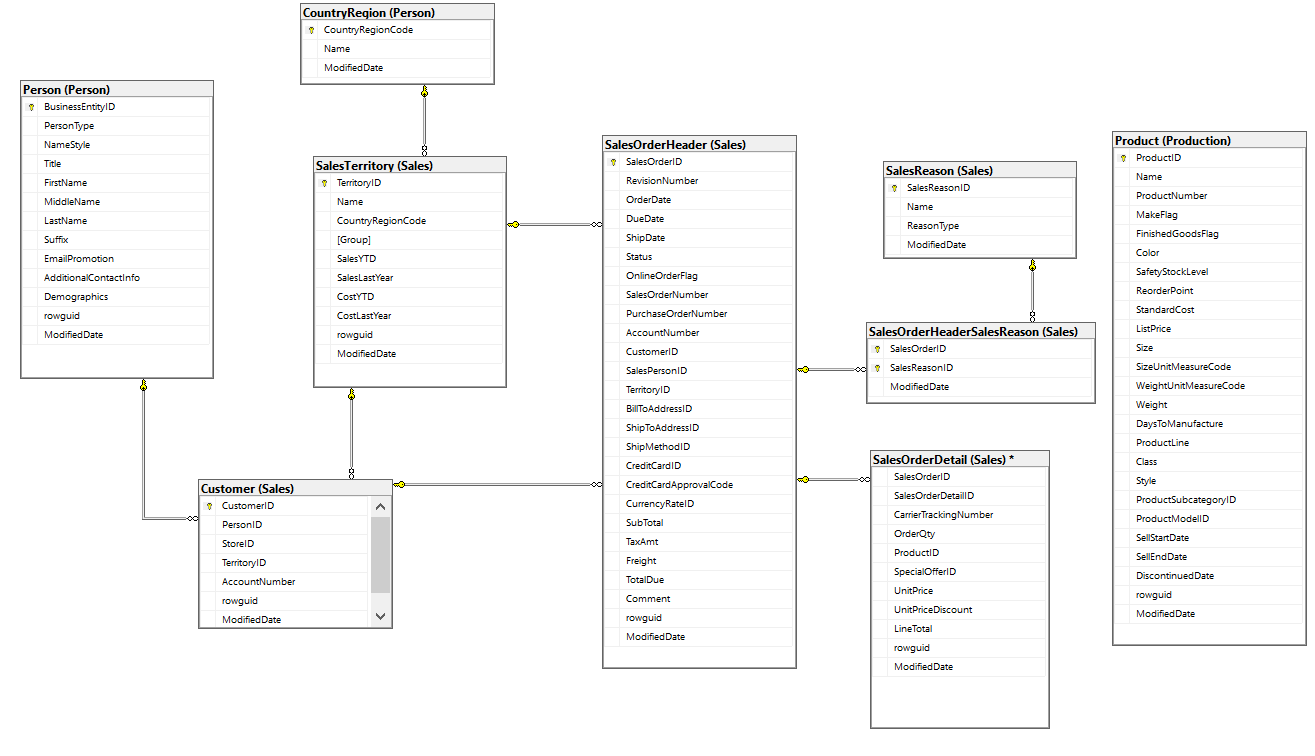


Figura 1 - ER Model Database AW2021

# Data sources

**Person:** This table contains information about the registered people. Each person has information needed, for example: sales, stores or products. The fields in this table are: Address, entity, contact, region, email, password, phone and state.

**Sales:** This table has all the information for the sale of a product. We have: credit card, currency, customer, order detail, the person in charge of the sale, reasons, territory, the shopping cart, the store where the product was bought and special offers.

Table 1: Summary of Person and Sales database contents

|  |  |  |
| --- | --- | --- |
| **Event / object** | **Table** | **Nr. Records** |
| Person | *Person.Address*  *Person.AddressType*  *Person.BusinessEntity*  *Person.BusinessEntityAddress*  *Person.BusinessEntityContact*  *Person.ContactType*  *Person.CountryRegion*  *Person.EmailAddress*  *Person.Password*  *Person.Person*  *Person.Phone*  *Person.PhoneNumberType*  *Person.StateProvince* | 19614  6  20777  19614  909  20  238  19972  19972  19972  19972  3  181 |
| Sales | *Sales.CountryRegionCurrency*  *Sales.CreditCard*  *Sales.Currency*  *Sales.CurrencyRate*  *Sales.Customer*  *Sales.PersonCreditCard*  *Sales.SalesOrderDetail*  *Sales.SalesOrderHeader*  *Sales.SalesOrderHeaderSalesReason*  *Sales.SalesPerson*  *Sales.SalesPersonQuotaHistory*  *Sales.SalesReason*  *Sales.SalesTaxRate*  *Sales.SalesTerritory*  *Sales.SalesTerritorHistory*  *Sales.ShoppingCartItem*  *Sales.SpecialOffer*  *Sales.SpecialOfferProduct*  *Sales.Store* | 109  19118  105  13532  19820  10118  121317  31465  27647  17, 282  163  10  29  10  17  3  16  538  701 |

Using the "*Open-Source* Data Quality and Profiling" tool, we conducted an in-depth study of the tables that we considered relevant for the execution of the Data Mart. With this tool we used the content summarization operation (Summary Data) to obtain the following results:

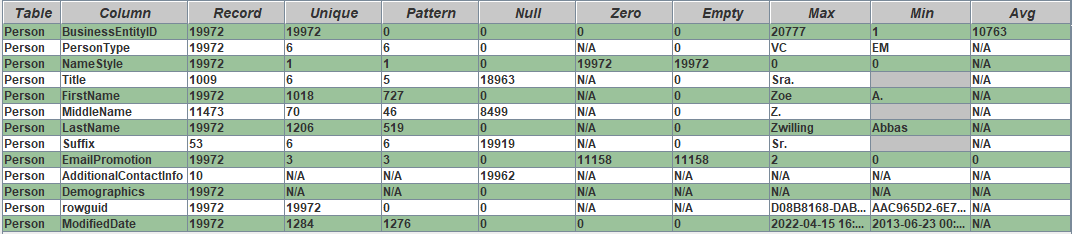
By analyzing the table, 19972 persons were registered, and about 42,6% didn’t fill the middle name field, as we can see in the following figure.

Figure 2 - Summary of table Person content.

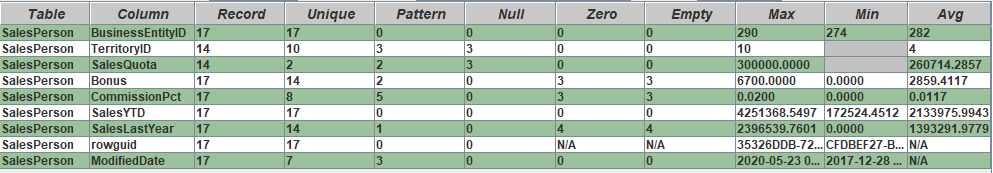
In this next table, only 17 persons are responsible of a sale, this information helps to know how many people are in charge of a sale.

Figure 3 - Summary of table SalesPerson content.

As we can see, a customer is associated with the purchase of a sale, the SalePerson is associated with the sale of that same purchase, and the person is the total of people aggregated into these two fields. This table helps you understand where and in which store the customer bought the product.

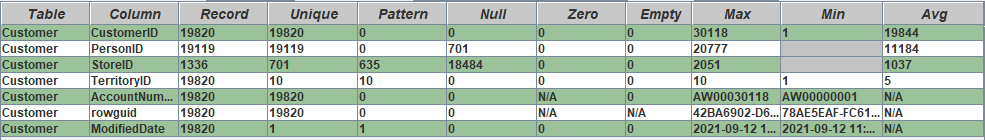


Figure 4 - Summary of table SalesCustomer content.

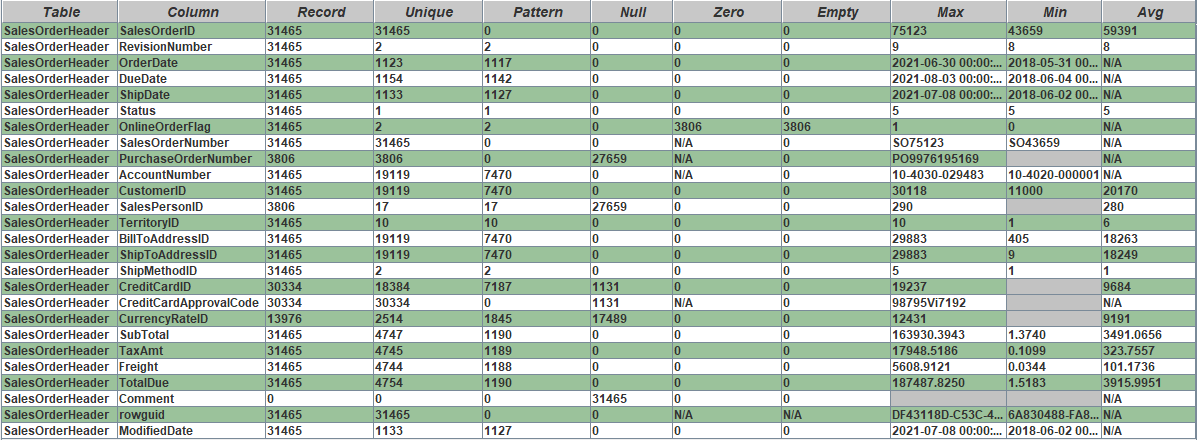
The table SalesOrderHeader, has a total of 31465 orders, and we can get the information of the client who placed the order, and see all the information needed to deliver the order and also the price of the product.

Figure 5 - Summary of table SalesOrderHeader content.

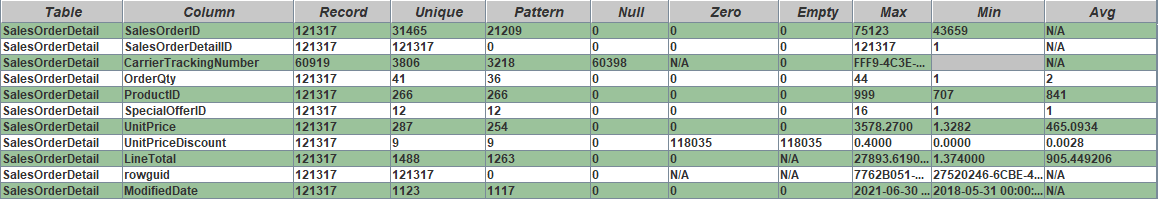
In this table we can see the details needed for the order, the price, the id, the order quantity of the product. There were 121317 orders.

Figure 6 - Summary of table SalesOrderDetail content.

# Dimensional modelling

Figure 7 - Dimensional Model

This database stores multiple information about various business processes like **purchases** and **sales**. Given these processes we defined some questions that the system should be able to answer:

* **Q1** - What was the most profitable month?
* **Q2** – Which store stole the most in a certain location?
* **Q3** - Who is the best salesperson for each product?
* **Q4** - Which store had the biggest sales growth?
* **Q5** - Which products were the most sold per month?
* **Q6** – Who was the client who spend the most money per month?

Table 2: Data Warehouse Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **DIMENSIONS**  **BUSINESS PROCESSES** | Customer | Product | Date | Location | Store | Person | Salesperson |
| Most profitable month in a year |  |  | X |  | X |  | X |
| Store with the most sales in a certain location |  |  |  | X | X |  |  |
| Best salesperson for each product |  | X |  |  |  | X | X |
| Store with the biggest sales growth |  |  |  | X | X |  |  |
| Most sold products per month |  | X | X | X | X |  |  |
| Client who spent the most money | X |  | X |  | X | X |  |

1. **Fact table(s) and its granularity**

This database is about a fictional USA company, that is both a manufacturer and seller of bicycles and accessories so this is an indicator that the fact tables should be of transactional type. There are two fact tables. The first one is the junction of **SalesOrderHeader** with **SalesOrderDetail** and this FT contains all the information of a sales process. The other one is the junction of **PurchaseOrderHeader** and **PurchaseOrderDetail** and this contains all the information of a purchase process. Both fact tables are of transactional type and each line represents a sales or purchase order respectively.

1. **Dimensions and attributes relevant to the analysis.**

We can use the following dimensions in the analysis:

* Customer: Information about a person, in relation to a shop and a sale
* Product: Basic product information
* Date: Sales and purchases dates (year, month, day, week)
* Location: Data of certain locations (How much was sold in a certain location)
* Store: Information of a store
* Person: Basic information of business entities. With it we can obtain distributions of various kinds.
* Salesperson: To obtain who was responsible for certain sales.

1. **Measures of the fact table**

Considering the above questions, the system should provide the following measures:

* Number of days to delivey: DueDate – OrderDate
* Price without tax: SubTotal + Freight - TaxAmt
* Total order price: SubTotal + Freight + Tax -> TotalDue
* Shipping cost -> Freight
* Amount paid by a product: OrderQty \* UnitPrice + Tax -> LineTotal

Many interesting metrics can be obtained from these measures, such as:

* Shipping price percentage of total cost: Freight \* 100 / Total
* Tax amount percentage of total cost: TaxAmt \* 100 / Total
* Number of online orders: Number of rows where OnlineOrderFlag == 1
* The number of orders where Card was decline Number of rows where CreditCardApprovalCode == Null

# Design of the dimensional data model

1. **Description Maps**

* *FT\_Sales*: It allow us to analyse the price, quantity, delivery dates, order and ships. We also can see the customer, the location, and the product ID. The other dimensions are linked to this fact table.
* *Dim\_Customer:* It allows us to analyse all customers, obtaining personal information from them;
* *Dim\_Store*: It allows us to see all the stores in the system, as well as their address, contact;
* *Dim\_Location*: Here we can see the location of the sales;
* *Dim\_Product*: It allows us to check all the products. We can see all the details of each product.
* *Dim\_SalesReason:* Here we can see what type of reason and the name of the reason;
* *Dim\_Date*: Date on which changes were made, either to products, customers, etc.

# Data mart implementation

**ETL**

ETL (Extraction, Transformation, and Loading) are software tools whose function is to extract data from various source systems, transform this data according to business rules, and finally load the data, usually into a Data Mart and/or Data Warehouse.

In this project, one of the main objectives is the creation of a Data Mart where the ETL process is performed to work on the data.

**ETL implementation**

In the ETL process, we used an application for building transformations and jobs, which helps handling the data, called PDI Client (Spoon) - Pentaho Data Integration.

We start this process by creating a connection to the database recommended by the teacher, AW2021, and we duplicate it by giving it the name AW2021DM. Next, we perform a share of both connections that we created, in order to make the necessary transformations.

Finally, using the same tool, we create a transformation for each dimension of our relational model.

**Process – Transformation and Jobs**

**Transformations**

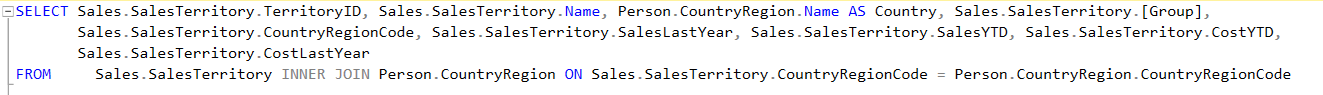
For each transformation was not used any database lookup or select values, because inner joins were performed in the tables as we can see in the following figure.

Figure 8 – Example of Inner joins

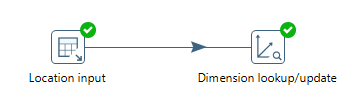


Figure 9 - Dimension Location created in spoon

The transformation **load\_location** was created, in which two tables are linked, the SalesTerritory and CountryRegion.

The customer’s dimension was also created, where two tables are stored, the person and the customer. In this dimension, we have all the customer information, from the name and the account number.

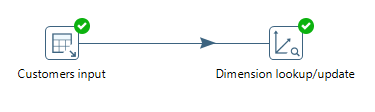


Figure 10 - Dimension Customer created in spoon

We repeat the process for the rest of the dimensions. All the detailed product information can be found in this dimension. We only use the product table.

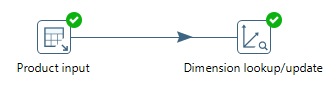


Figure 11 - Dimension Product created in spoon

For this fact table we joined the SalesOrderHeader table with the SalesOrderDetail table. We put all the information related to a sale, since a sale has both customer and product information. It also has a date which tell us when the product was shipped and in how many days the ship was completed.

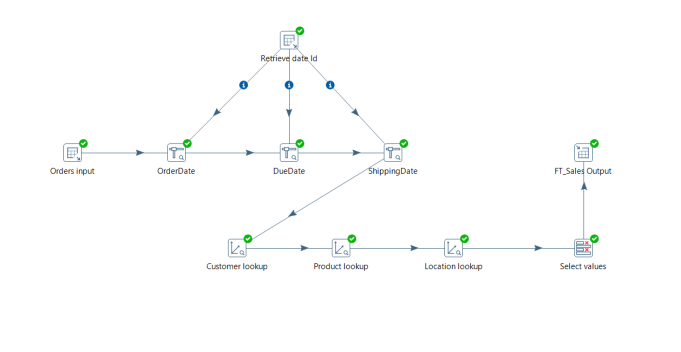


Figure 12 - Fact table Sales created in spoon

Another fact table has been created, as this specifies the reason for a sale, and it is important that it remains a fact table to minimize the information that we can included in the other fact table. We minimize the information that is passed to the sales facts table, because instead of showing repeated data with different reasons, it shows us all the reasons for one product



Figure 13 – Fact table SalesReason created in spoon

And lastly, the Date Dimension was created using a file provided by the teacher with only a couple of changes.

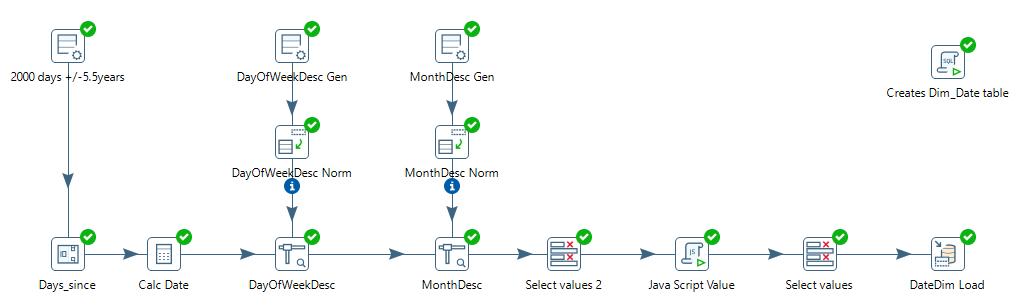


Figure 14 -Data Dimension created in spoon

**Jobs**

To speed up the whole process of loading dimensions and fact tables, we created a Job to do this process. In a first phase, the Job was created to sequentially load all dimension tables and the fact table (FT\_Sales and FT\_SalesReason). This Job has the function of loading the entire ETL process, from the transformations of the dimension tables to the transformation of the fact table. Jobs simplify the process, making it more understandable and efficient.

# Conclusion

With the accomplishment of this work, we were able to obtain a greater learning in what concerns the curricular unit Decision Support Systems, as well as the importance of a Data Mart, being this very useful to deepen data information and join them.

We got to know more about SQL Server, Pentaho Data Integration (Spoon) and it was also possible to know a new tool, namely Open-Source Data Quality and Profiling, which allowed us to analyse the data in more depth.

We had some difficulties implementing the Data Mart because it was something totally new to us and never implemented before. However, we think that this work will be important in the future and that it was successful.

# Bibliography

Power Points provided by Docent

<https://medium.com/@kiaorahao/data-warehouse-system-application-on-sales-person-evaluation-d454e6052785>

<https://dspace.uii.ac.id/bitstream/handle/123456789/12400/05.4%20bab%204.pdf?sequence=5&isAllowed=y>

<https://holowczak.com/data-warehouse-dimensional-modeling/11/>

# Appendix A – Data description maps

Table 3: Data description map of dimensions

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Type of table** | **Nr. Records** | | **Description** | | | | |
| FT\_Sales | Fact Table | 60398 | | In this table we have the price, quantity, delivery dates, order and ship; we also can see the customer, the location and the product ID. | | | | |
| Dim\_Customer | Dimension | 19120 | | Customer ID, full name and account number is the information that exists in this dimension. | | | | |
| Dim\_Location | Dimension | 11 | | We have the territory ID, the name, the country, the group and the country Region Code that identifies the location of a shop. It also shows the total sales of those shops by location. | | | | |
| Dim\_Product | Dimension | 296 | | In this dimension we can see all the necessary information for the product. | | | | |
| Dim\_Date | Dimension | 2000 | | In this dimension all the variables are defined to show all the existing dates in the other dimensions. | | | | |
| Dim\_SalesReason | Dimension | 11 | | Shows the reasons for each sale. | | | | |
| **Target (Data mart)** | | | | **Source (OLTP)** | | | | |
| **Column** | **Description** | **Data type** | **SCD** | **Table** | **Column** | **Data type** | **ETL rules** | **Example of values** |
| LineTotal | Amount paid by a product | Int | 1 | SalesOrderDetail | LineTotal | Int |  | 2024.994000 |
| DaysToDelivery | Number of days to delivery | Int | 1 | SalesOrderHeader | DueDate and ShipDate | Int |  | 6 |
|  |  |  | 1 |  |  |  |  |  |