UNIVERSITY OF PORTO

FACULTY OF ENGINEERING IN THE UNIVERSITY OF PORTO

PROJECT

DATA WAREHOUSES

v 1.0

AUTHORS:

Klara Banić

Laura Majer

Table of contents

1. Subject Description & Assignment Requirements

Analysis and forecasting of the sales of product is extraordinary important for the managers of supermarkets. *"Supermarket transactions"* is the chosen subject for this assignment due to the fact that it has data which can be constructively analysed. Customer transaction data enable a deep analysis of what goes into the customers' baskets. In that way, the supermarkets' managers can make smarter decisions towards higher profits and better customer satisfaction, which is both equally important.

The chosen data consists of numerous interesting facts. Every transaction consists of the transaction ID, purchase date, customer ID and store ID. It also holds the data considering the products – product type, product department, product family, how many units of each product have been sold and what is the total revenue. The customers' data represents their gender, marital status, are they homeowners or not, do they have children and what is their annual income. On the other hand, the store has its location which consists of a city, a region and a country.

The assignment requires the data that can later be analysed properly. The number of facts should be over 10 000, and the chosen data has a bit more than 14 000. There should be four dimensions represented in the data-cube modelling. Considering the fact that the provided data meets all of the stated requirements, it has been chosen to perform the task on.

2. Planning: dimensional bus matrix, dimensions and facts dictionary

2.1 Dimensional bus matrix

In the following table the bus matrix is displayed. Five dimensions are considered, namely *Time*, *Store location*, *Product*, *Gender* and *Annual income*. These dimensions might be useful for analysing customer data and habits.

Concerning the given data, two facts are chosen to be represented. They are *Quantity* (of sold products) and *Total revenue*.

	Time	Store location	Product	Gender	Annual income
Quantity	x	x	x	x	x
Total revenue	x	x	x	x	x

Table 1. Dimensional bus matrix

2.2 Dictionary

Dimensions

TIME – the dimension representing the date of the purchase. The granularity (bottom up) is *Date – Month – Quarter – Year* with additional attributes like month names and days of the week.

STORE LOCATION – the dimension representing the location of the purchase. The granularity (bottom up) is *City – Region – Country*.

PRODUCT – the dimension representing data about the purchased product. The granularity (bottom up) is *Product Category – Product Department – Product Factory*.

GENDER – representing the gender of the customer. Values either M or F.

ANNUAL INCOME – representing the annual income of the customer. Values in String representing the range of incomes.

Measures

QUANTITY

Description: the overall quantity of products sold

Additive measure: YES

Aggregation function: SUM

TOTAL REVENUE

Description: the overall revenue gained by the sales

Additive measure: YES

Aggregation function: SUM

3. MultiDim conceptual model

The graph below represents the Conceptual Multidimensional Model of the chosen data. It was necessary to think about the model thoroughly considering the fact that it allows good communication between users and designers to understand the application requirements. MultiDim is a conceptual multidimensional model which is based on the entity-relationship model.

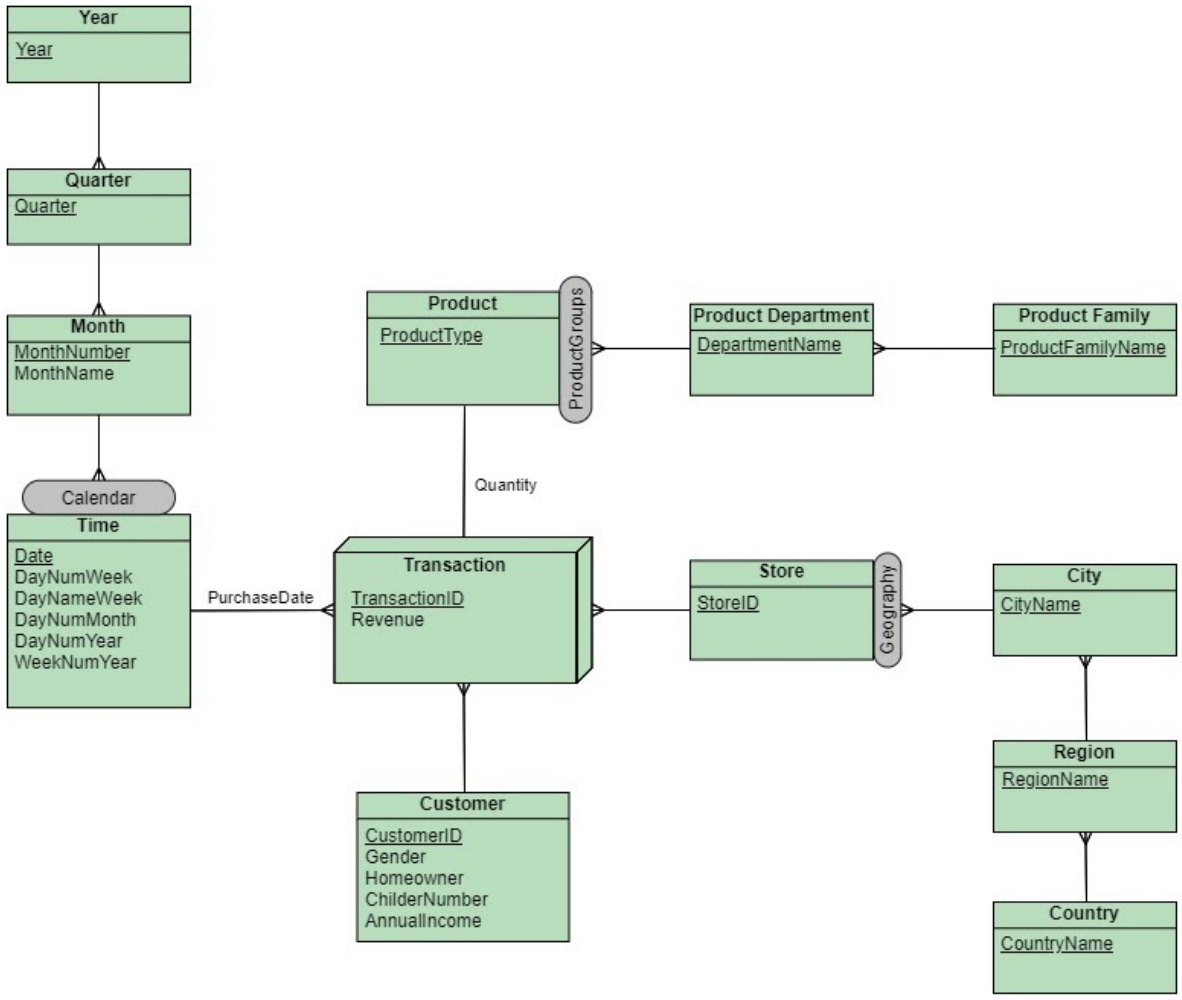

Figure 1 - MultiDim Model of Supermarket Transactions

A fact relates measures to leaf levels in dimensions. The fact in this model is a *Transaction*. It is related to dimension *Time* with Many-to-one cardinality, with the relationship attribute *PurchaseDate*. The *Time* dimension can be represented with the balanced hierarchy. Each *Time* dimension can be analysed as *Date*, *Month*, *Quarter* or *Year*. The *Transaction* has a One-to-one relationship with the *Product* dimension, with the relationship attribute *Units* which represents how many units of the product are in the transactions. The *Product* is also in a balanced hierarchy with the *Product*

Department, as well as *Product Family*. The *Transaction* has Many-to-one cardinality with a *Store* dimension. The *Store*'s geography is in a balanced hierarchy as well. The location of the store can be represented by the city, the region and/or the country. The *Customer* dimension is connected to the fact with One-to-many relationship. *Customer* dimension has various data on each customer.

4. Dimensional data model

In the second phase of the Kimball Lifecycle methodology, a physical drawing is constructed. There are multiple schemes to choose from including the star schema, snowflake schema and fact constellation schema. In this project, the star schema has been chosen because of its simplicity and ability of defining multi-level dimensions. Figure 2 shows the dimensional data model using a star schema.

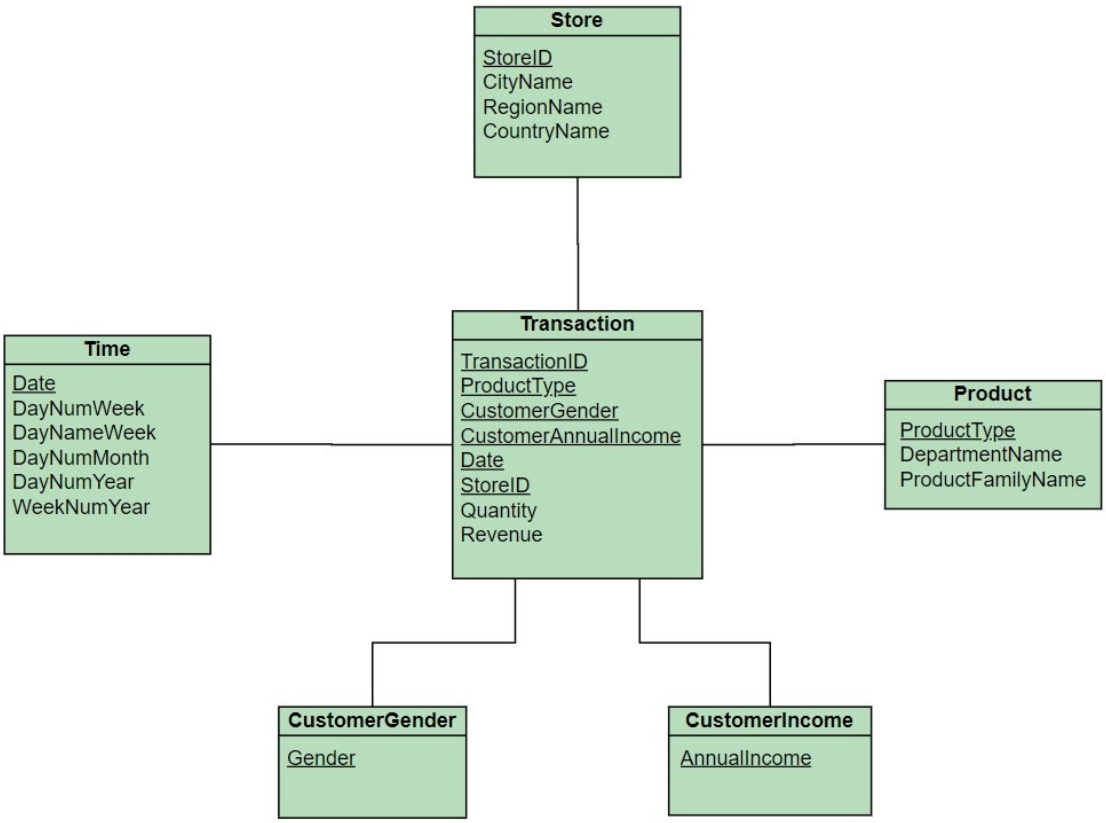

Figure 2 - Dimensional data model

The star schema contains only one fact table, *Transaction*. Both measures, *Quantity* and *Revenue*, can be aggregated after consulting the bus matrix. The five dimensions *Time*, *Store*, *Product*, *CustomerGender* and *CustomerIncome* are all linked to the fact table in the middle through a foreign key. They also contain the rest of the hierarchy, all pictured in one table, unlike the snowflake schema where the hierarchies are branched out.

5. Data sources selection

Kaggle is an online community of data scientists and machine learning practitioners. It allows users to find and publish data sets and work with other data scientists and machine learning engineers. Considering the fact that it has over 536 000 active members from 194 countries and it receives close to 150 000 submissions per month, this was indeed the right place to explore the data and find the perfect one for this project. All in all, the chosen data was found exactly on this platform (https://www.kaggle.com/josephlovins/supermarket-transactions).

The first phase for this project was to find adequate data which would have enough information to do the whole project on, including data which would be represented as facts or dimensions. The search was not easy, but after spending enough time analysing all the available data, the one to do the project on was found and therefore chosen. It was chosen due to the fact that it satisfied the criteria of the project - the number of facts was over 10 000 and it also met the rule of the requirement of 4 dimensions. The data has 16 columns and was downloaded in Microsoft Excel format (.xlsx).

	A	B	C	D	E	F	G	H	I
1	**Transaction**	**Purchase Date**	**CustomerID**	**Gender**	**Marital Status**	**Homeowner**	**Children**	**Annual Income**	**StoreID**
2	1	18.12.2011	7223	F	M	Y	2	$30K - $50K	1
3	2	20.12.2011	7841	M	M	Y	5	$70K - $90K	1
4	3	21.12.2011	8374	F	M	N	2	$50K - $70K	2
5	4	21.12.2011	9619	M	M	Y	3	$30K - $50K	3
6	5	22.12.2011	1900	F	S	Y	3	$130K - $150K	4
7	6	22.12.2011	6696	F	M	Y	3	$10K - $30K	4
8	7	23.12.2011	9673	M	S	Y	2	$30K - $50K	5
9	8	25.12.2011	354	F	M	Y	2	$150K +	6

J	K	L	M	N	O	P	Q
City	**State or Province**	**Country**	**Product Family**	**Product Department**	**Product Category**	**Units Sold**	**Revenue**
Los Angeles	CA	USA	Food	Snack Foods	Snack Foods	5	$27,38
Los Angeles	CA	USA	Food	Produce	Vegetables	5	$14,90
Bremerton	WA	USA	Food	Snack Foods	Snack Foods	3	$5,52
Portland	OR	USA	Food	Snacks	Candy	4	$4,44
Beverly Hills	CA	USA	Drink	Beverages	Carbonated Beverages	4	$14,00
Beverly Hills	CA	USA	Food	Deli	Side Dishes	3	$4,37
Salem	OR	USA	Food	Frozen Foods	Breakfast Foods	4	$13,78
Yakima	WA	USA	Food	Canned Foods	Canned Soup	6	$7,34

Figure 3 - Data entries in Excel file

Figure 3 represents a few transactions that are in the .xsl file. As stated earlier, each transaction has its ID, the date of purchase, as well as customer ID, store ID, product ID and the number representing the quantity of the purchased product. All the information with detailed description can be found in the previous chapters.

The data that will be analysed the most is the quantity of the sold products. In that way, the supermarket transactions allow us to dig deep and find out what type of products are being sold the most, as well as analyse the customers and their purchased products regarding their sex, annual income, etc.

6. Transformations including incremental loadings

After creating the data warehouse using MySQL Workbench, it had to be filled with all the data which was stored in the database, i.e. Microsoft Excel file. In order to do so, the transformations which take the data, transform it to fit the data warehouse and store it into the data warehouse were made using the Pentaho Data Integration tool Kettle. It delivers powerful ETL (extraction, transformation and loading) operations which is usually the main choice for organizations due to the fact that its graphical drag and drop design was proven to be very intuitive and efficient. PDI Client, i.e. Spoon, is one of the most important components of Pentaho Data Integration. It can be described as IDE (Integrated Development Environment) for writing data pipeline workflows.

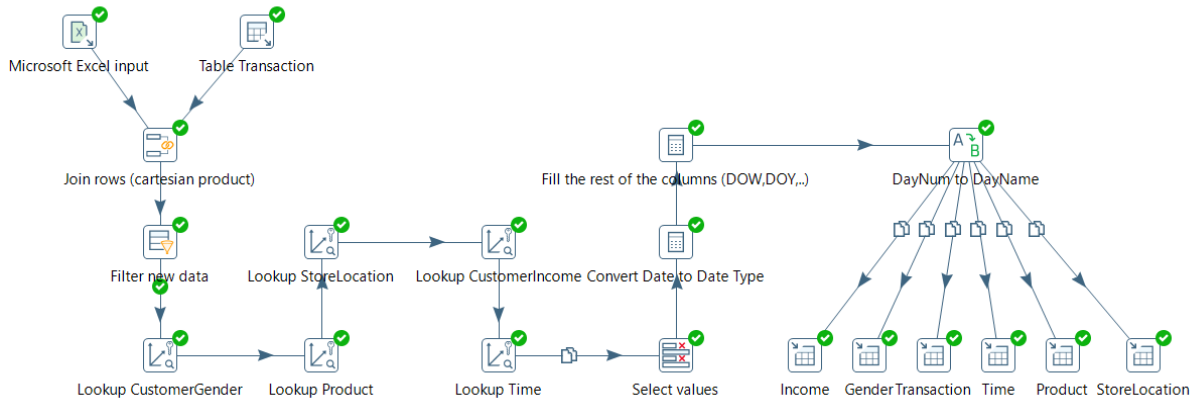

Figure 4 – Data extraction and transformation process

The transformation was made step by step, firstly starting with the data input. Spoon allows many data inputs such as CSV file, JSON, YAML, table and so many more. It also has the option to treat the Microsoft Excel file as input which was then chosen.

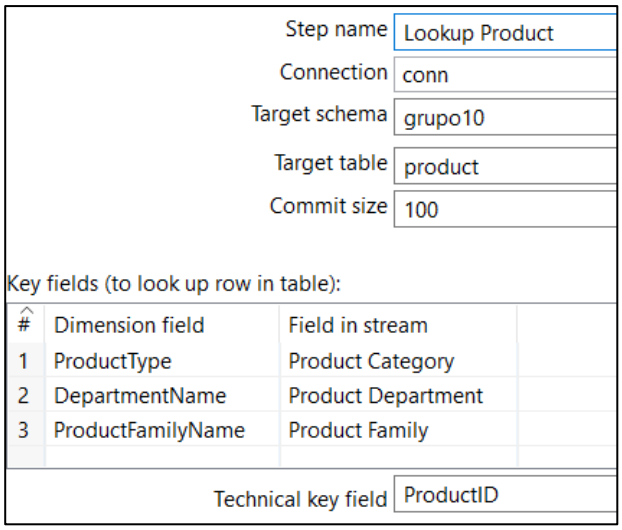

Figure 5 - Data input configuration

Putting the input table Transaction, Cartesian product and Filter aside for now, which will be described in the following chapter Jobs, the next step is Lookup for the data warehouse's dimensions which can be seen in the Figure above for Lookup product. This step looks up a combination of business key fields from the input stream in a dimension table. After doing the mapping and passing through this step, all of the remaining data changes for the dimension table can be made.

Having finished with all the Lookups, it was finally time to select all the values that will be stored in the data warehouse. Out of 23 values, 22 were taken and renamed to achieve consistency.

Next step was converting Purchase date to Date type, so that the rest of the columns regarding the Time dimensions could be filled with appropriate data.

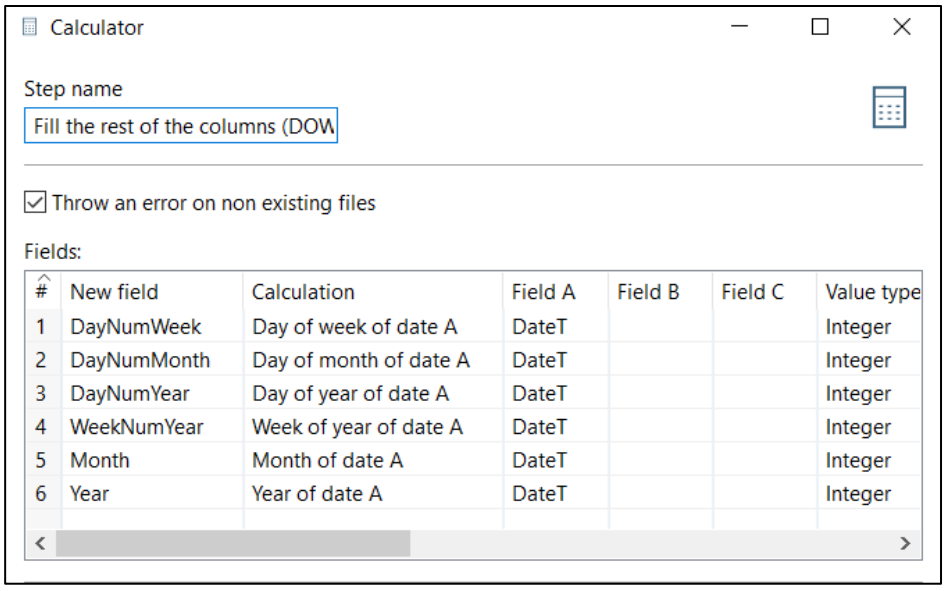

Figure 6 - Data conversion using the Calculator component

In order to achieve that, the component Calculator was used in the way which can be seen in the Figure 6. Having built in functions such as extracting day of week, week of the year or day of month of a date, Spoon made the creation of the transformation much easier.

Next step was Value mapper which takes the source value and maps it into a target value according to our choice. As we wanted to have not only the day of week of a certain date, but also the day name, this component helped with that problem. The accordingly filled in Value mapper can be seen in the Figure 7.

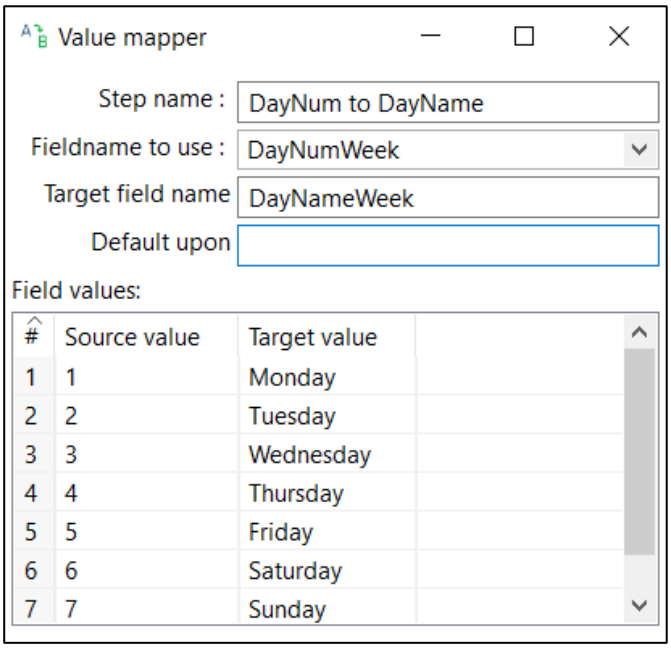

Figure 7 - Value mapper for mapping number of the day of the week into name of the day of the week

In the end, the stream of transformed data could be finally stored into our data warehouse. The fact table *Transformation* and dimensions *AnnualIncome*, *Gender*, *Time*, *Product* and *StoreLocation* were filled with accumulated data and the most important part of the project, data analytics, could be started.

7. Jobs

A cron job is a utility that schedules a command or script to run automatically at a specified time and date. It can be very useful to automate repetitive tasks, and is used in this project as well.

Imagining that our Excel data source will have new data, it would make sense for us to store the new data in our data warehouse as well. However, it is not the goal to make the transformation run every minute, but it would be optimal to decide how often we want to store new data to the data warehouse and also, more important, make it automatically.

Spoon is a powerful tool due to the fact that it enables creating and configuring a job. There are many options that can be used in a Spoon job, such as sending a mail, checking the conditions, pinging the host etc. The job created for this project was a very simple one, since it is only needed to get the new data from the database once a month and do the transformation which will store the new data into our data warehouse.

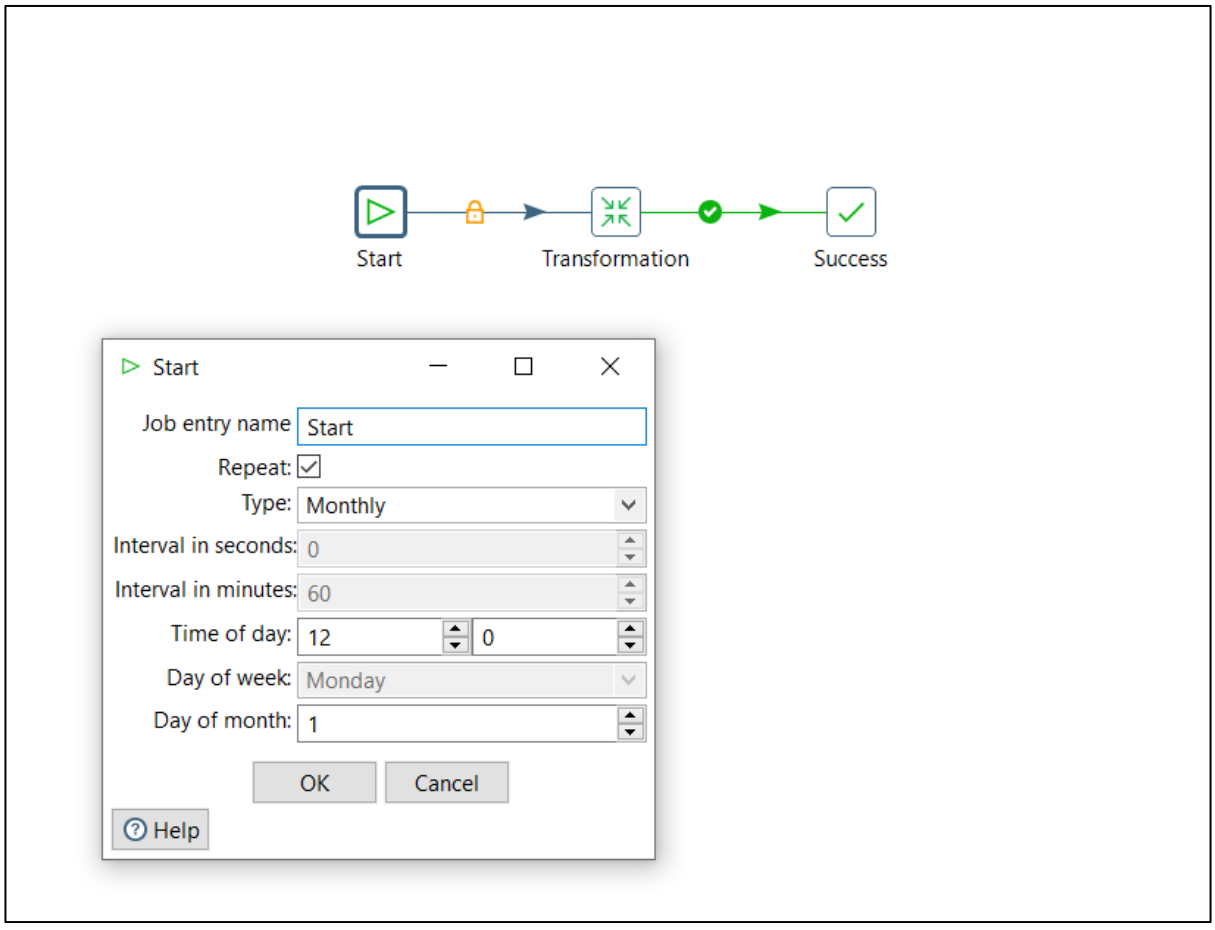

Figure 8 - Job configuration in Spoon

A Start component is needed for the job to start. It can be configured by stating if the job should repeat itself and how often it should do it. The component then triggers the original Transformation which was explained in the previous chapter. However, here is the time and place to explain the input table Transaction, Cartesian product and Filter.

In order to store in our data warehouse only the new data, we have decided to choose one efficient way to do it. The input table Transaction is a simple SQL query which returns the maximum Transaction ID.

```
SELECT MAX(TransactionID)
FROM grupo10.transaction
```

We imagine that Transaction IDs will grow over time and will never be repeated so it is reasonable to choose this way to determine which data is the new data.

Having in mind that Spoon works with streams of data, it was challenging to find an appropriate way to compare the Excel input data's Transaction ID to the value which Transaction table returned. One of the ways to achieve this was to make a Cartesian product which will join the maximal Transaction ID to all rows from the input and then it can be filtered. The filtration was achieved by using the component Filter rows which was easily configured.

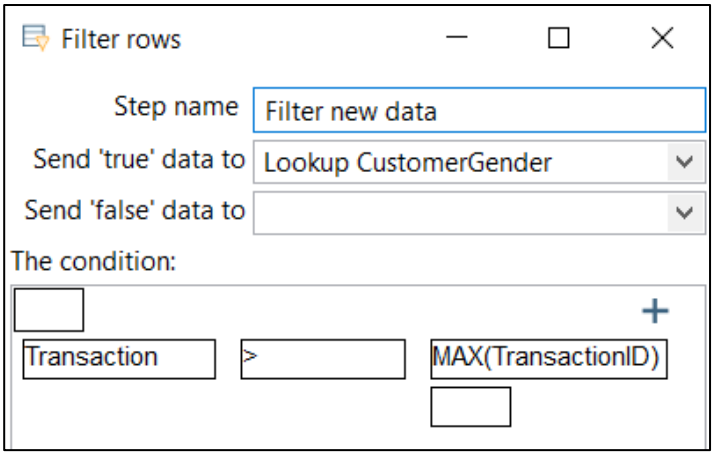

Figure 9 - Data filtering

After this step, only the new data would go throughout the whole process of transformation which significantly improves the efficiency and time performance.

In the end, when the transformation is done, it triggers the Success component which shows that the job has been completed successfully.

8. Multi-dimensional modelling

The schema workbench is a tool made for defining the virtual cubes. The result of modelling is an XML document. In order for the schema to be published successfully, a connection to the Pentaho server is required.

The first step of building a data cube is connecting to the corresponding database. After that, a new cube can be created. The cube that was modelled in this project is the *Amount* cube, with five dimensions: *Time, Location, Product, Gender* and *Annual income.* The purpose of modelling this cube is to determine how different factors of the transaction contribute to the purchased amount.

Figure 10 shows the constructed data cube for this project's database. The fact table is pictured (*Transaction*) as well as the five aforementioned dimensions. *Amount* was chosen as a measure.

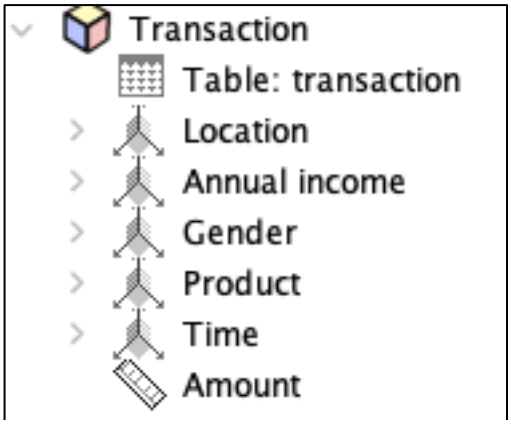

Figure 10 - Transaction data cube configuration

When constructing the data cube, hierarchies were defined for each dimension. For instance, the *Location* dimension has a hierarchy of three levels, namely *City*, *Region* and *Country.* Each level of the hierarchy has to be configured accordingly. A view of the hierarchy is shown in Figure 11, and the configuration of the *City* level is shown in Figure 12. There it's stated that the parent column of *City* is *Region.* The level *Country* doesn't have a parent column.

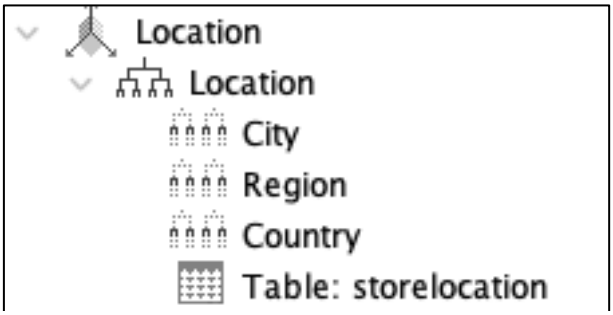

Figure 11 - Location hierarchy

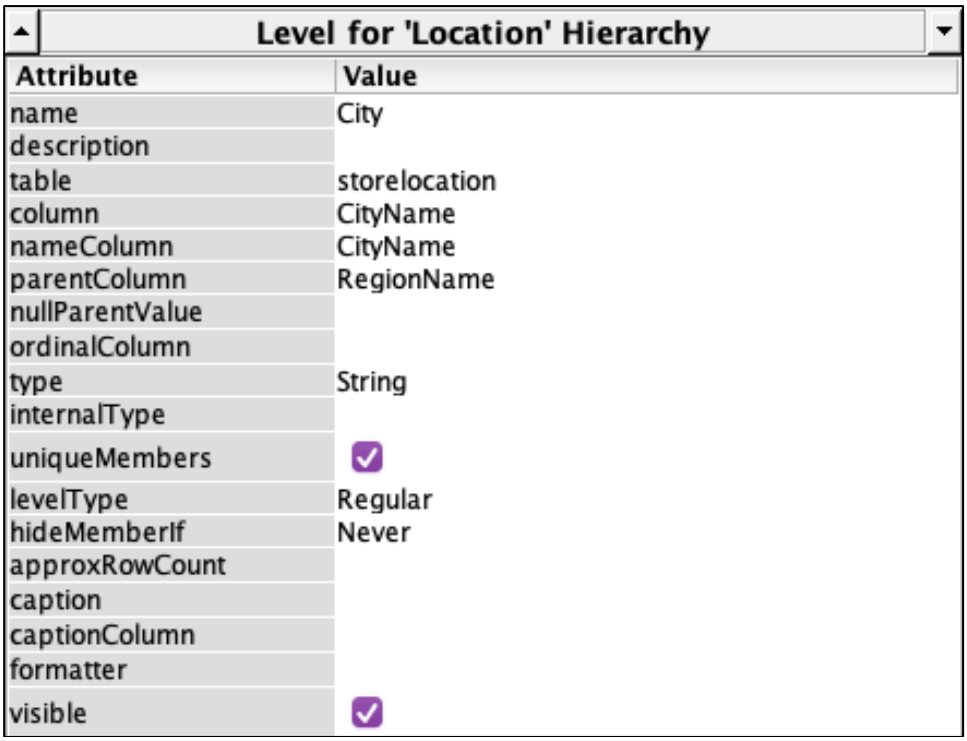

Figure 12 - City level configuration

Another dimension with a three-level hierarchy is the *Product* dimension, pictured on Figure 13, with the levels being *ProductType*, *ProductDepartment* and *FamilyName.* However, not all dimensions have a branched-out hierarchy. The dimension *Annual income* pictured in Figure 14 only has one level in the hierarchy.

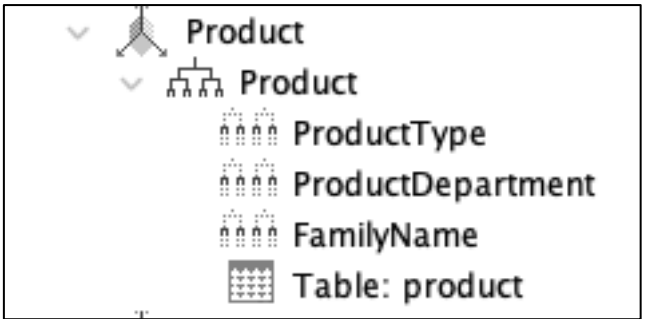

Figure 13 - Product hierarchy

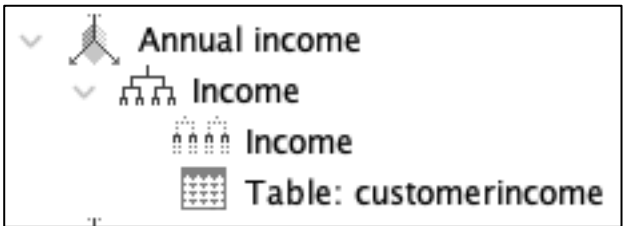

Figure 14 - Annual income level configuration

The configuration of the measure Amount is pictured in Figure 15. The measure represents the total amount of sold products in all the stores. The aggregation used for this measure is sum, since all the amounts of particular purchased products need to be added together to represent the total amount of sold products.

Measure for 'transaction' Cube

Attribute	Value
name	Amount
description	
aggregator	sum
column	ProductType
formatString	
datatype	
formatter	
caption	
visible	☑

Figure 15 - Measure configuration

9. Data analysis (dashboards, MDX queries & friends)

The purpose of a data warehouse is being able to easily extract specific data regarding a temporal measure and view the corresponding graphs. In this paragraph a possible presentation of data and graphs interesting to the client will be discussed. The graphs were made in the Pentaho dashboard, and a corresponding MDX query will be given for each of them.

9.1 Analysis by country and year

The starting point of this analysis is the success of the supermarket chain by country and by year of operation (since the data spans through two years for Canada and Mexico, and three years for USA). Figure 16 shows the amounts purchased by the store country and year, and in Figure 17 the corresponding graph is displayed. It is apparent that sales have dropped in the USA, and in an attempt to regain customers a marketing and business strategy needs to be developed.

Gender	AnnualIncome	ProductFamilyName	CountryName	Date	Measures Quantity
All Customergender.Genders	All Customerincome.AnnualIncomes	All Product.ProductFamilyNames	Canada	2012	37
				2013	1.823
			Mexico	2012	147
				2013	8.497
			USA	2011	143
				2012	18.800
				2013	11.315

Figure 16 - Amount by country

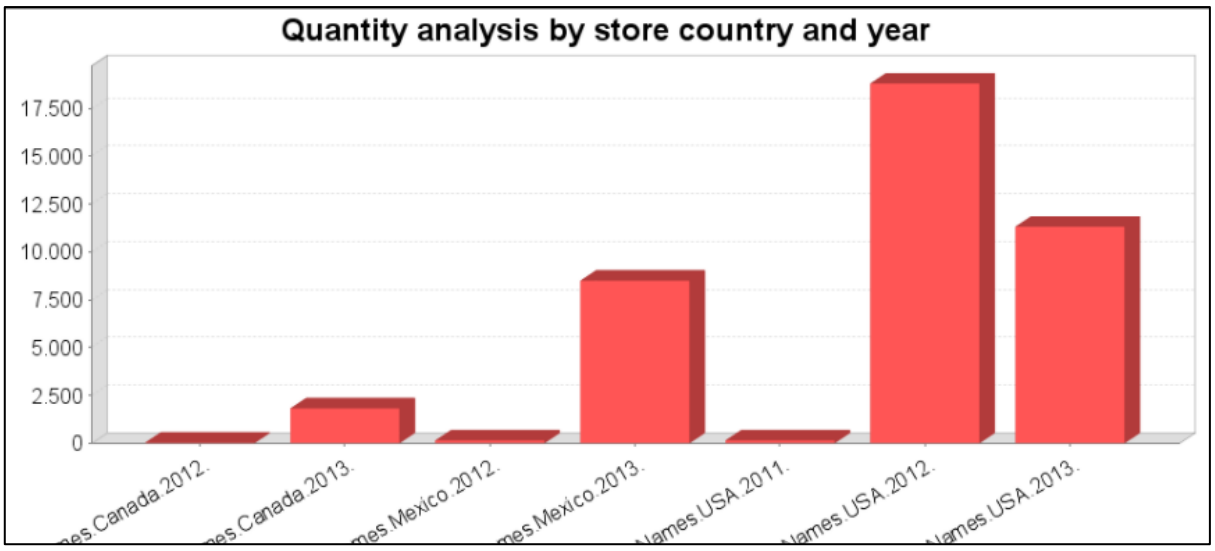

Figure 17 - Quantity analysis graph by country

The corresponding MDX query is as follows:

```
select NON EMPTY {[Measures].[Quantity]} ON COLUMNS,
NON EMPTY Crossjoin({[Customergender.Gender].[All
Customergender.Genders]},
Crossjoin({[Product.ProductFamilyName].[All
Product.ProductFamilyNames]},
Crossjoin({[Storelocation.CountryName].[Canada],
[Storelocation.CountryName].[Mexico],
[Storelocation.CountryName].[USA]}, Crossjoin({[Time.Date].[2011],
[Time.Date].[2012], [Time.Date].[2013]},
{[Customerincome.AnnualIncome].[All
Customerincome.AnnualIncomes]})))) ON ROWS
from [analysis-supermarket]
```

9.2 Analysis by Customer Gender

Quantity analysis by gender is one of the ways to analyse the sales, and can bring more insight into customer behaviour. In case the data shows that there are more female customers than male ones, the supermarket should get and include more products that are more oriented towards female community, and vice versa.

After the analysis of the data, pictured on Figure 18 together with the corresponding graph, it was determined that the amounts purchased by male and female customers were nearly the same, which did not help in the process of determining a valid marketing strategy.

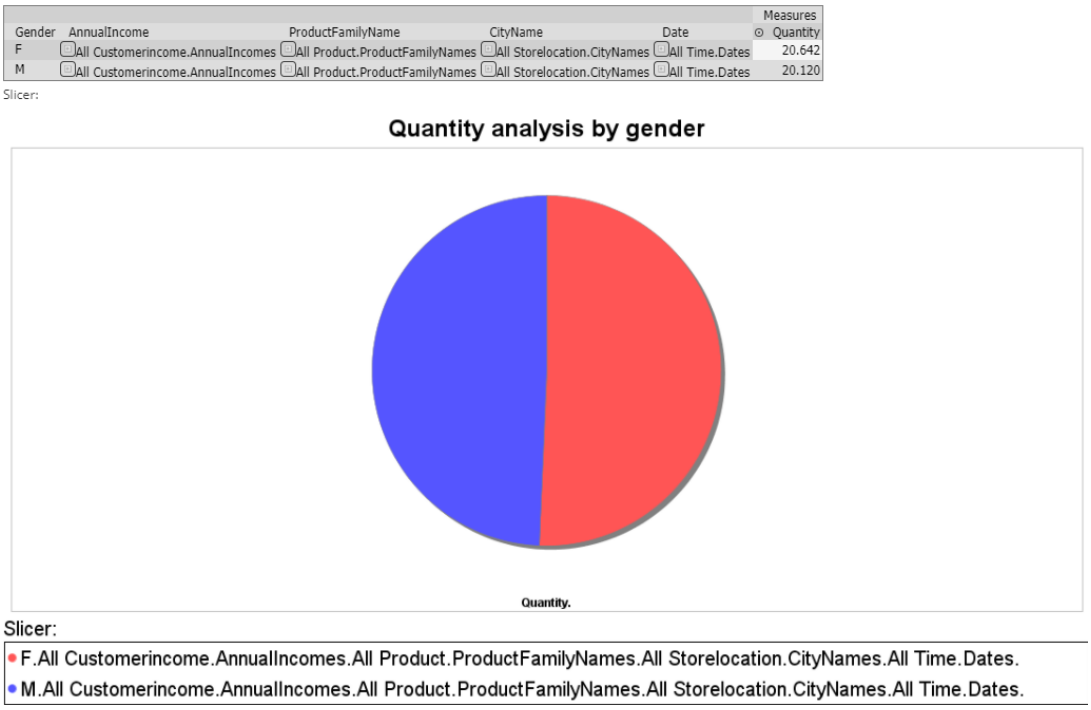

Figure 18 - Quantity analysis by gender

The corresponding MDX query is as follows:

```
select NON EMPTY {[Measures].[Quantity]} ON COLUMNS,
NON EMPTY {([Customergender.Gender].[F],
[Product.ProductFamilyName].[All Product.ProductFamilyNames],
[Storelocation.CountryName].[All Storelocation.CountryNames],
[Time.Date].[All Time.Dates], [Customerincome.AnnualIncome].[All
Customerincome.AnnualIncomes]), ([Customergender.Gender].[M],
[Product.ProductFamilyName].[All Product.ProductFamilyNames],
[Storelocation.CountryName].[All Storelocation.CountryNames],
[Time.Date].[All Time.Dates], [Customerincome.AnnualIncome].[All
Customerincome.AnnualIncomes])} ON ROWS
from [analysis-supermarket]
```

9.3 Analysis by Customer Annual Income

Another useful piece of information correlated to customer habits is their annual income. An insight into the average income of the customers can be used for determining whether more affordable or luxurious products should be offered.

After analysing the quantities purchased by each income group shown in Figure 19, it was determined that customers with a lower income are more frequent. Thus, it would be more profitable to expand the selection of cheaper products, even if those are not so high in quality. The graph of quantity purchased related to annual income is pictured in Figure 20.

Gender	AnnualIncome	ProductFamilyName	CityName	Date	Measures Quantity
All Customergender.Genders	$10K - $30K	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	8.890
	$30K - $50K	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	13.325
	$50K - $70K	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	6.927
	$70K - $90K	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	4.965
	$90K - $110K	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	1.870
	$110K - $130K	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	1.855
	$130K - $150K	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	2.176
	$150K +	All Product.ProductFamilyNames	All Storelocation.CityNames	All Time.Dates	754

Figure 19 - Quantity analysis by annual income

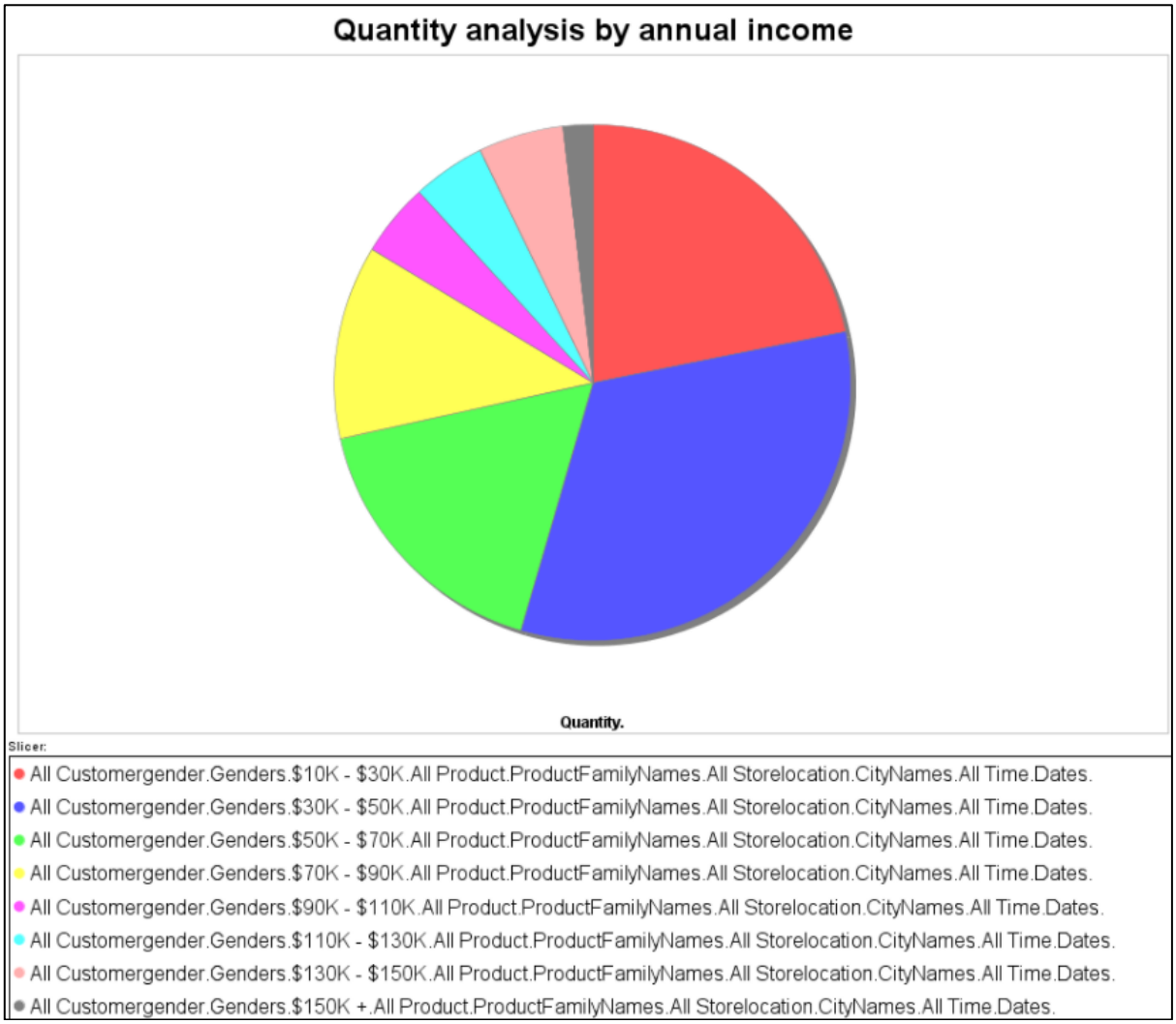

Figure 20 - Quantity analysis by annual income graph

The corresponding MDX query is as follows:

```
select NON EMPTY {[Measures].[Quantity]} ON COLUMNS,
NON EMPTY Crossjoin({[Customergender.Gender].[All
Customergender.Genders]},
Crossjoin({[Product.ProductFamilyName].[All
Product.ProductFamilyNames]},
Crossjoin({[Storelocation.CountryName].[All
Storelocation.CountryNames]}, Crossjoin({[Time.Date].[All
Time.Dates]}, {[Customerincome.AnnualIncome].[All
Customerincome.AnnualIncomes],
[Customerincome.AnnualIncome].[$10K - $30K],
[Customerincome.AnnualIncome].[$30K - $50K],
[Customerincome.AnnualIncome].[$50K - $70K],
[Customerincome.AnnualIncome].[$70K - $90K],
[Customerincome.AnnualIncome].[$90K - $110K],
[Customerincome.AnnualIncome].[$110K - $130K],
[Customerincome.AnnualIncome].[$130K - $150K],
[Customerincome.AnnualIncome].[$150K +]})))) ON ROWS
from [analysis-supermarket]
```

9.4 Analysis by Product Family

While determining which products should be advertised more, put on sale or cancelled, an analysis by the product family is necessary. After extracting the data shown in Figure 21, it is apparent that *Food* is the most represented product family. The corresponding graph is shown in Figure 22.

					Measures
AnnualIncome	Gender	CountryName	Date	ProductFamilyName	Quantity
$10K - $30K	All Customergender.Genders	All Storelocation.CountryNames	All Time.Dates	Drink	749
				Food	6.399
				Non-Consumable	1.742
$30K - $50K	All Customergender.Genders	All Storelocation.CountryNames	All Time.Dates	Drink	1.220
				Food	9.772
				Non-Consumable	2.333
$50K - $70K	All Customergender.Genders	All Storelocation.CountryNames	All Time.Dates	Drink	577
				Food	5.016
				Non-Consumable	1.334

Figure 21 - Quantity analysis by product family

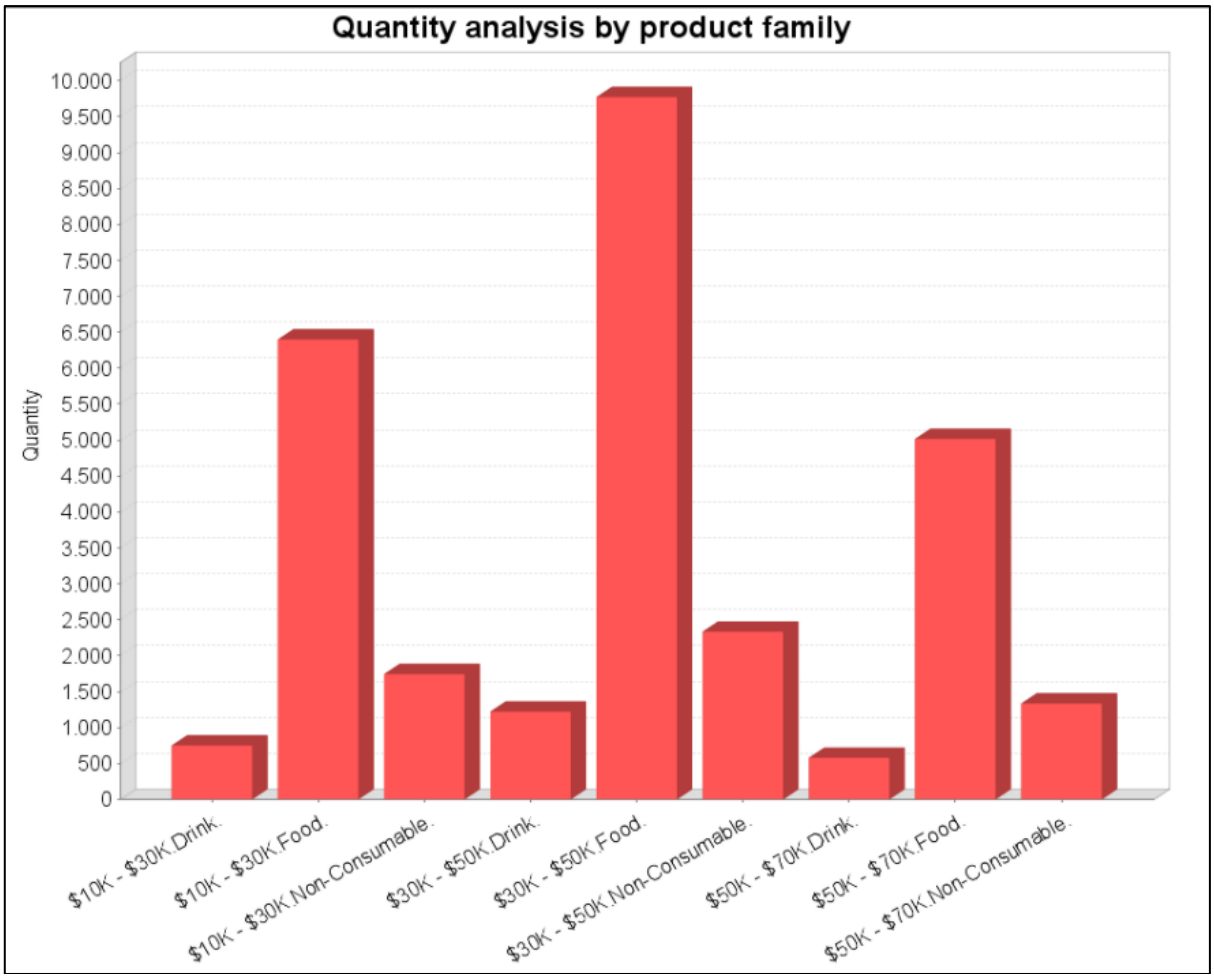

Figure 22 - Quantity analysis by product family graph

The corresponding MDX query is as follows:

```
select NON EMPTY {[Measures].[Quantity]} ON COLUMNS,
NON EMPTY Crossjoin({[Customerincome.AnnualIncome].[$10K - $30K],
[Customerincome.AnnualIncome].[$30K - $50K],
[Customerincome.AnnualIncome].[$50K - $70K]},
Crossjoin({[Customergender.Gender].[All Customergender.Genders]},
Crossjoin({[Storelocation.CountryName].[All
Storelocation.CountryNames]}, Crossjoin({[Time.Date].[All
Time.Dates]}, {[Product.ProductFamilyName].[Drink],
[Product.ProductFamilyName].[Food],
[Product.ProductFamilyName].[Non-Consumable]})))) ON ROWS
from [analysis-supermarket]
```

9.5 Analysis by Product Type

A deeper analysis of the amounts purchased by the product type is possible after narrowing down the product family to *Food*. The result is pictured on Figure 23. It is apparent that *Produce* is the most frequently bought product type, however, *Snack foods* are also a good contender for a product to be put on sale or increase its diversity and quantity. The corresponding graph is pictured in Figure 24.

Gender	CountryName	Date	AnnualIncome	ProductFamilyName	Measures Quantity
All Customergender.Genders	All Storelocation.CountryNames	All Time.Dates	All Customerincome.AnnualIncomes	Food	29.630
				Meat	251
				Seafood	275
				Canned Products	317
				Eggs	547
				Breakfast Foods	548
				Starchy Foods	855
				Snacks	1.037
				Baked Goods	1.220
				Dairy	1.972
				Deli	2.052
				Canned Foods	2.817
				Baking Goods	3.081
				Frozen Foods	3.982
				Snack Foods	4.686
				Produce	5.990

Figure 23 - Quantity analysis of product types inside the Food family

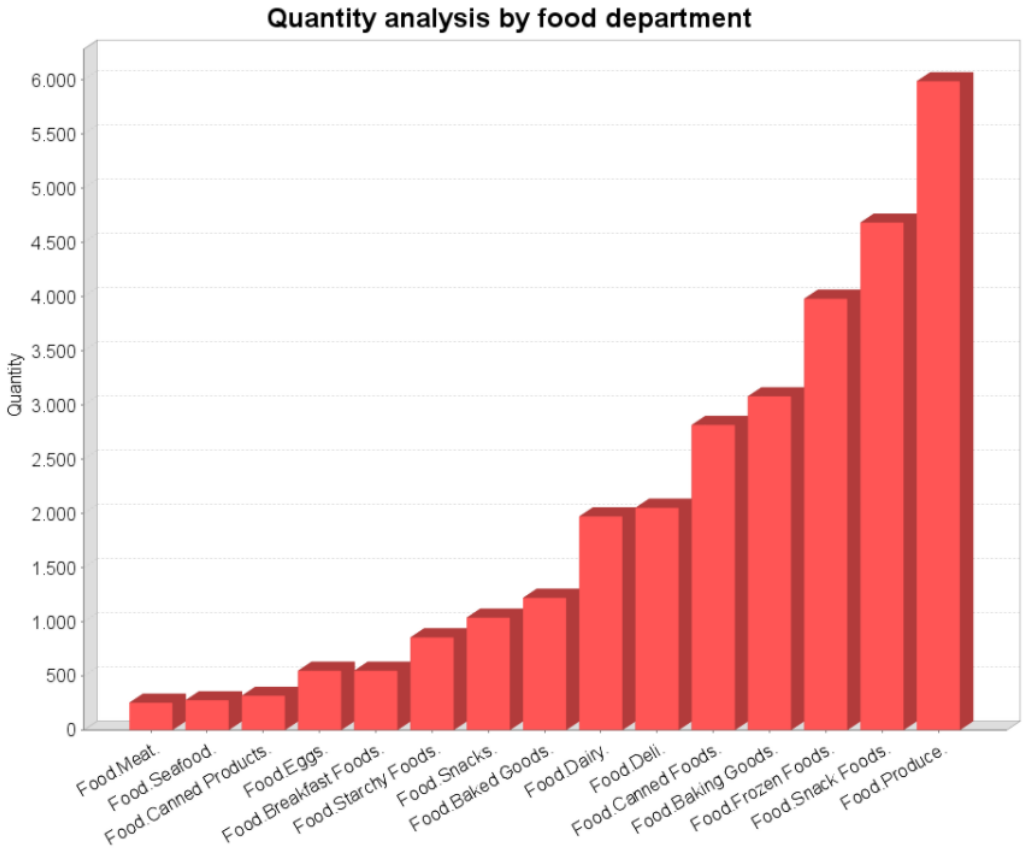

Figure 24 - Quantity analysis by product type

The corresponding MDX query is as follows:

```
select NON EMPTY {[Measures].[Quantity]} ON COLUMNS,
NON EMPTY Hierarchize(Crossjoin({[Customergender.Gender].[All
Customergender.Genders]},
Crossjoin({[Storelocation.CountryName].[All
Storelocation.CountryNames]}, Crossjoin({[Time.Date].[All
Time.Dates]}, Union(Crossjoin({[Customerincome.AnnualIncome].[All
Customerincome.AnnualIncomes]},
{[Product.ProductFamilyName].[Food]}),
Crossjoin({[Customerincome.AnnualIncome].[All
Customerincome.AnnualIncomes]},
[Product.ProductFamilyName].[Food].Children)))))) ON ROWS
from [analysis-supermarket]
```

10. Conclusion

During the process of building a data warehouse, starting from acquiring the data to constructing the cube, a lot of details have to be considered.

For example, the database has to be structured properly and the attribute names need to be clear. The building of the data warehouse relies on complete and concrete information about the database entries. During this project, we encountered a case of unclear names for data attributes, namely *Snacks* and *Snack foods*. Without proper information, it is impossible to analyse the given data thoroughly and accurately.

Also, during other steps of the warehouse construction, like the building of the cube, it is important to know the peculiarities of the case study. Communication with the client and deep insight into the data and project requirements are necessary for the proper construction of the data warehouse. It is an intuitive and useful way to project and represent data from different databases all representing a united business entity.

It is a shame that some programs or parts of the used tools were not working, e.g. CDE Dashboard, but JPivot produced graphs that were easy to analyse. Looking at the future of this project, it should be explored which versions of which tools should be used in order to prevent the tools from freezing and not working.

Knowledge from the fields of management, business and programming are all necessary, and professionals from all the fields can benefit from the experience of building data warehouses.