Data Management and Business Intelligence

Assignment 2nd

Charilaos Petrakogiannis

A.M: f2822112

Ilias Dimos

A.M: f2822102

Professors: D.Hatziantoniou, S.Safras

Academic Period: 2021-2022

Table of Contents

EXPLANATION OF THE DATASET

In order for us to proceed on the 2nd course assignment, we had to choose firstly a dataset. Having in mind the business orientation of this master's program, we've searched also for a business-oriented dataset. For this reason, we've ended up with data that is related with sales records.

More specifically, this assignment's dataset is a csv file that contains 100.000 sales from all over the world on 12 different item types. The csv file consists of 14 columns (region, country, item type, sales channel, order priority, order date, order id, ship date, units sold, unit price, unit cost, total revenue, total cost, and total profit) and 100.000 different observations (rows) and was downloaded from Kaggle's site.

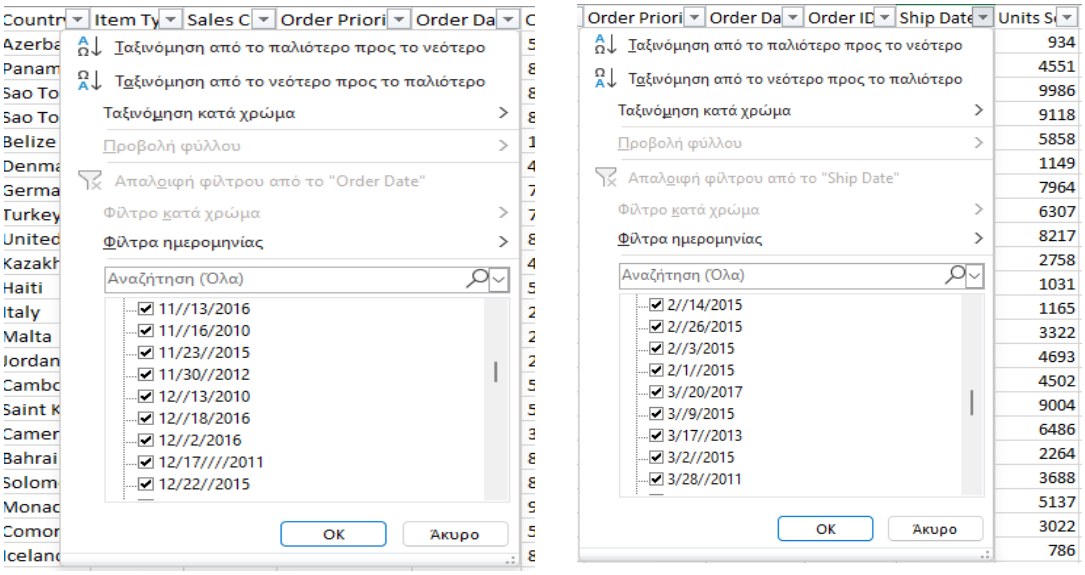
Region	Country	Item Type	Sales Cha...	Order Prio...	Order Date	Order ID
Middle East and North Africa	Azerbaijan	Snacks	Online	C	10/8/2014	535113847
Central America and the Caribbean	Panama	Cosmetics	Offline	L	2/22/2015	874708545
Sub-Saharan Africa	Sao Tome and Principe	Fruits	Offline	M	12/9/2015	854349935
Sub-Saharan Africa	Sao Tome and Principe	Personal Care	Online	M	9/17/2014	892836844
Central America and the Caribbean	Belize	Household	Offline	H	2/4/2010	129280602

Ship Date	Units Sold	Unit Price	Unit Cost	Total Reve...	Total Cost	Total Profit
10/23/2014	934	152.58	97.44	142509.72	91008.96	51500.76
2/27/2015	4551	437.20	263.33	1989697.20	1198414.83	791282.37
1/18/2016	9986	9.33	6.92	93169.38	69103.12	24066.26
10/12/2014	9118	81.73	56.67	745214.14	516717.06	228497.08
3/5/2010	5858	668.27	502.54	3914725.66	2943879.32	970846.34
2/28/2013	1149	109.28	35.84	125562.72	41180.16	84382.56
5/3/2013	7964	437.20	263.33	3481860.80	2097160.12	1384700.68
4/7/2012	6307	9.33	6.92	58844.31	43644.44	15199.87

TRANSFORMATION OF THE DATASET

In order to check if everything is ok with our dataset, we filtered the columns separately to see if they have some kind of mistake.

Indeed, in the columns Order Date and Ship Date we have some extra verticals lines which they split the dates in MONTH/DAY/YEAR not in correct way.

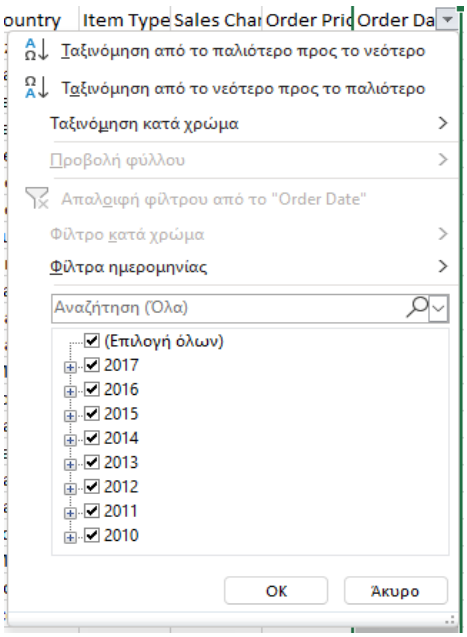


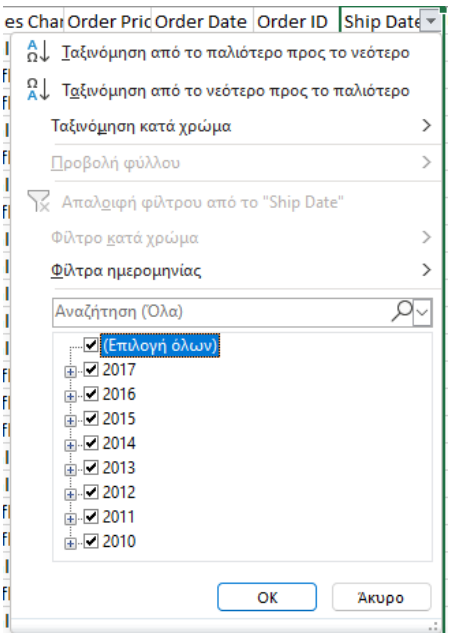
To fix this problem we imported the dataset in R, and we manage to get rid of these values not by deleted them but by fixing them.

Analytically,

```
getwd()

dataset<- read.csv("C:\\Users\\elgr9\\OneDrive\\Desktop\\Sales.csv")

View(dataset)
str(dataset)


dataset$Order.Date<-gsub("//","/",dataset$Order.Date) # where the script finds "//" it replaces them with "/"

dataset$Ship.Date <-gsub("//","/",dataset$Ship.Date)

nrow(dataset) # In order to see if any line has been deleted


write.csv(dataset,'Sales Records.csv') # We save the new file as Sales Records to continue with the ETL process
```

Indeed, we can see that the Order Dates and Ship Dates are properly distributed based on the Year after the Transformation Process.





After we successfully transformed the type of the Order Date and Ship Date into an acceptable form, we procced by describing the data types of the "Sales Records" Dataset.

```
> Sales_Records <- read.csv("C:\\Users\\elgr9\\OneDrive\\Desktop\\Sales Records.csv")
> str(Sales_Records)
'data.frame':   100000 obs. of  14 variables:
 $ Region        : chr  "Middle East and North Africa" "Central America and the Caribbean" "Sub-Saharan Africa" "Sub-Saharan Africa" ...
 $ Country       : chr  "Azerbaijan" "Panama" "Sao Tome and Principe" "Sao Tome and Principe" ...
 $ Item.Type     : chr  "Snacks" "Cosmetics" "Fruits" "Personal Care" ...
 $ Sales.Channel : chr  "Online" "Offline" "Online" ...
 $ Order.Priority: chr  "C" "L" "M" "M" ...
 $ Order.Date    : chr  "10/8/2014" "2/22/2015" "12/9/2015" "9/17/2014" ...
 $ Order.ID      : int  535113847 874708545 854349935 892836844 129280602 473105037 754046475 772153747 847788178 471623599 ...
 $ Ship.Date     : chr  "10/23/2014" "2/27/2015" "1/18/2016" "10/12/2014" ...
 $ Units.Sold    : int  934 4551 9986 9118 5858 1149 7964 6307 8217 2758 ...
 $ Unit.Price    : num  152.58 437.2 9.33 81.73 668.27 ...
 $ Unit.Cost     : num  97.44 263.33 6.92 56.67 502.54 ...
 $ Total.Revenue : num  142510 1989697 93169 745214 3914726 ...
 $ Total.Cost    : num  91009 1198415 69103 516717 2943879 ...
 $ Total.Profit  : num  51501 791282 24066 228497 970846 ...
```

1. **Region**: Character variable that represents 7 different regions all over the world
2. **Country**: Character variable that represents 185 different countries all over the world
3. **Item Type**: Character variable that represents 12 different item categories
4. **Sales Channel**: Character variable that represents 2 different sales channels (offline-online). It shows the way in which each specific item was ordered.
5. **Order Priority:** Character variable that represents 4 different order priorities (low, medium, high and critical).
6. **Order Date**: Character variable that represents 2766 different dates in which the items were ordered.
7. **Order ID:** Integer variable that represents the unique ID of each order.
8. **Ship Date**: Character variable that represents 2813 different dates in which the items were shipped for their last destination.
9. **Units Sold**: Integer variable that represents the units of each order that were sold.
10. **Unit Price**: Numeric variable that represents the price of each item's unit that were sold.

11. **Unit Cost**: Numeric variable that represents the cost of each item's unit that were sold.
12. **Total Revenue**: Numeric variable that represents the total revenue for each order that were made.
13. **Total Cost**: Numeric variable that represents the total cost for each order that were made.
14. **Total Profit**: Numeric variable that represents the total profit for each order that were made.

CLEANING OF THE DATASET

To check if we have any NA'S or NULL'S values in the Dataset, we created a small script that checks all the lines in order to find them if they exist.

```
for(i in nrow(Sales_Records))
{
  cat("Checking for NA's",sep="\n")
  cat("There are",sum(is.na(i)),"NA's",sep =" ")

  cat("\n")

  cat("Checking for NULL's",sep="\n")
  cat("There are",sum(is.null(i)),"NULL's",sep =" ")

}
```

The output of the script above is:

```
Checking for NA's
There are 0 NA's
Checking for NULL's
There are 0 NULL's
```

Fortunately, we didn't have any NAs or NULLSs, so we are ready to proceed we the ETL PROCESS.

CASE DESCRIPTION

In order for us to execute this specific assignment, firstly we had to create an imaginary case in which we would like to answer some business questions.

For this reason, we assume that Ilias and Charis are the co-founders of the DIMPET ANALYTICS CO. This entity is a business organization that's specialized on providing analytical advice for enterprises that lack of that service within their business environment.

So, let's tell you a story!

Back in 2017 when everything was fine and our business was booming a multinational enterprise that sells almost every type of consumable items (from clothes to vegetables and from baby food to beverages) and has online and physical stores all over the world, came to our door and asked for our help. They wanted to organize their next investment on office supplies and household items, but they didn't know exactly in which region of the world and in what type of stores (physical or online) they should focus on.

All the previous years we were working as a company only by using our databases through the SSMS without caring about a potential level-up move, due to the fact that the size of our customers was a small one. This brand-new enterprise and its market significance gave us the opportunity to update and evolve our business systems by creating a data warehouse with which we would be able to manage big data faster and smarter!

For this reason and after a lot of thinking and research we ended up with the idea of creating a star-schemed multidimensional model. We believed that that this was the right choice because we wanted to focus on the speed of our data analysis and not so on the storage section, due to the fact that we had local servers with huge storage capacity in our company.

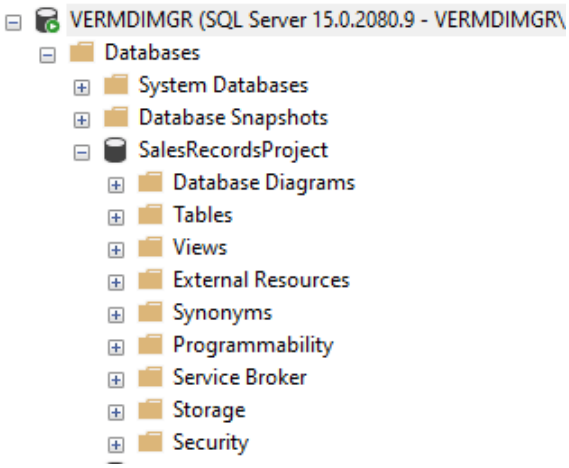
Below you'll be able to watch and evaluate the process that we conducted back in 2017 in order for us to be able to give that customer investment advice by using a multidimensional model for our data management.

DATA WAREHOUSE CONSTRUCTION

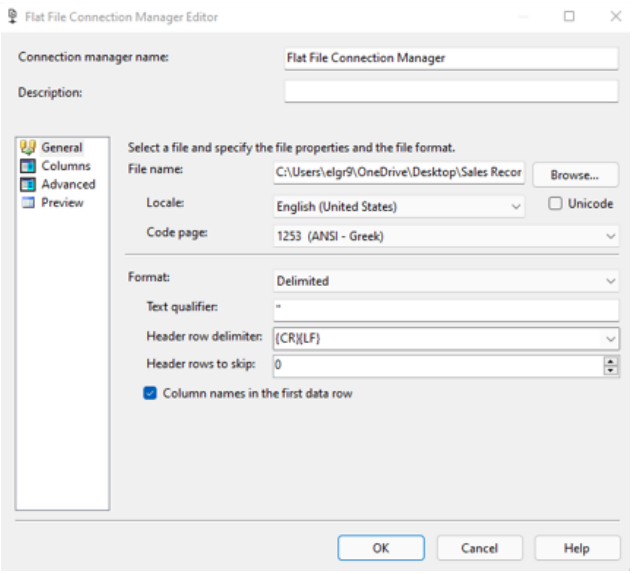
ETL PROCESS

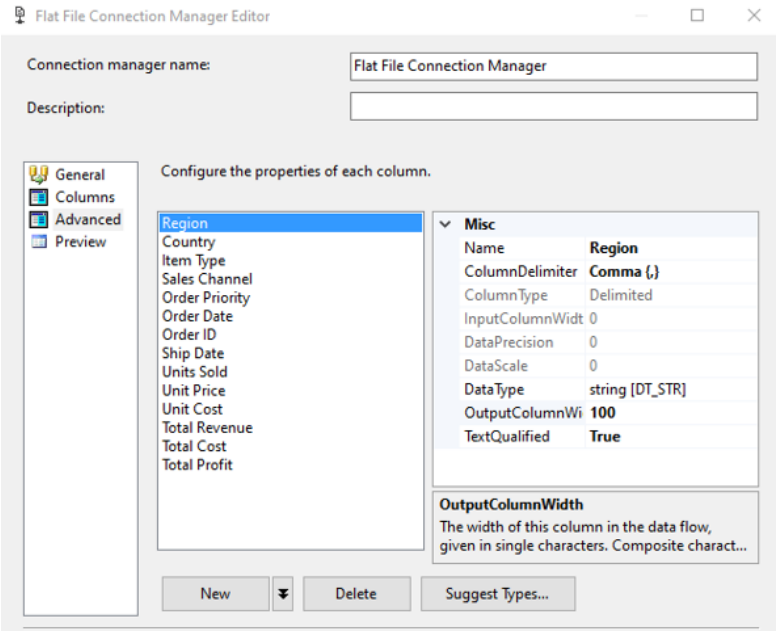
The first thing that we had to do was to extract, transform and load the data that was in our position, into the warehouse that we wanted to build. For this reason, we followed specific steps that can be seen below:

1) First, we had to create a new database on SSMS in order for us to load the data on different tables (staging, dimension and fact table). For this reason, we used SSMS as administrators and then we proceeded on SSIS for the next steps.



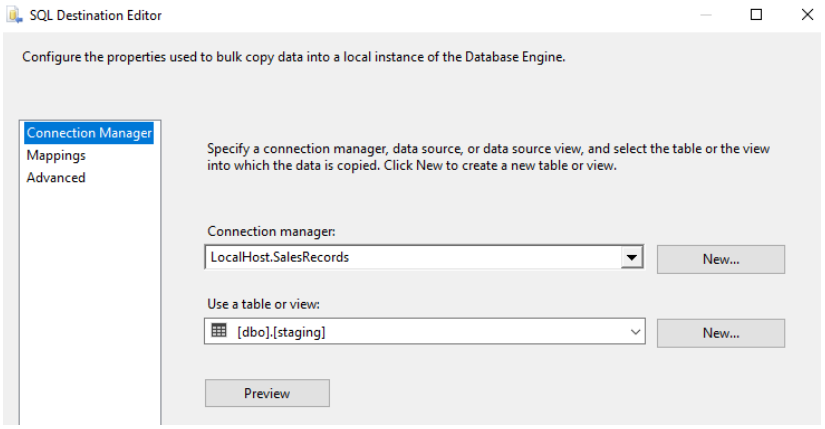
2) On SSIS we had to create a data flow task and by using the “suggest types” button, we specified the data type of each column. In this specific step we faced a small problem. As you can see from the dataset above, the two date variables (ship and order) got a character type. In ETL process we would like to change it into date type, but the SSIS and SSMS could not cooperate. We’ve tried to alter their character type through Rstudio, excel and table updates by changing their type to every possible date type and format that exists, but again the SSIS and the SSMS were only accepting the string type for these specific variables of the dataset. For this reason, we’ve kept their string type and continued the process. This detail didn’t end up with fatal (for our assignment) results and so we continued our tasks.


Flat File Connection Manager Editor
Connection manager name:
Flat File Connection Manager
Description:
General
Columns
Advanced
Preview
Select a file and specify the file properties and the file format.
File name:
C:\Users\elgr9\OneDrive\Desktop\Sales Recor
Browse...
Locale:
English (United States)
Unicode
Code page:
1253 (ANSI - Greek)
Format:
Delimited
Text qualifier:
"
Header row delimiter:
{CR}{LF}
Header rows to skip:
0
Column names in the first data row
OK
Cancel
Help

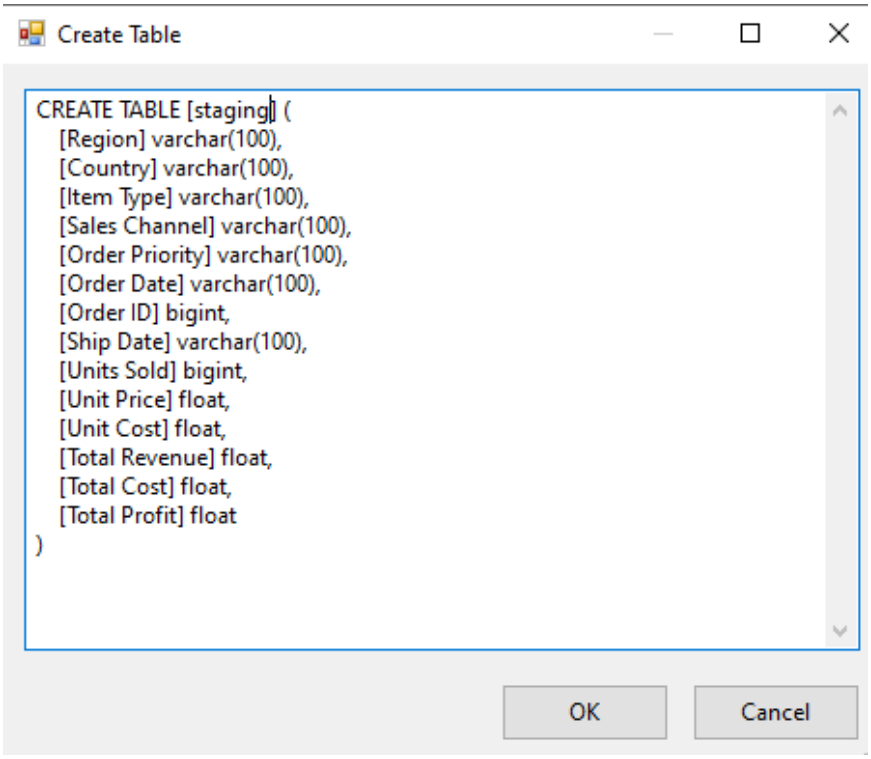

Flat File Connection Manager Editor
Connection manager name:
Flat File Connection Manager
Description:
General
Columns
Advanced
Preview
Configure the properties of each column.
Region
Country
Item Type
Sales Channel
Order Priority
Order Date
Order ID
Ship Date
Units Sold
Unit Price
Unit Cost
Total Revenue
Total Cost
Total Profit
Misc
Name
Region
ColumnDelimiter
Comma (,)
ColumnType
Delimited
InputColumnWidt
0
DataPrecision
0
DataScale
0
DataType
string [DT_STR]
OutputColumnWi
100
TextQualified
True
OutputColumnWidth
The width of this column in the data flow, given in single characters. Composite charact...
New
Delete
Suggest Types...

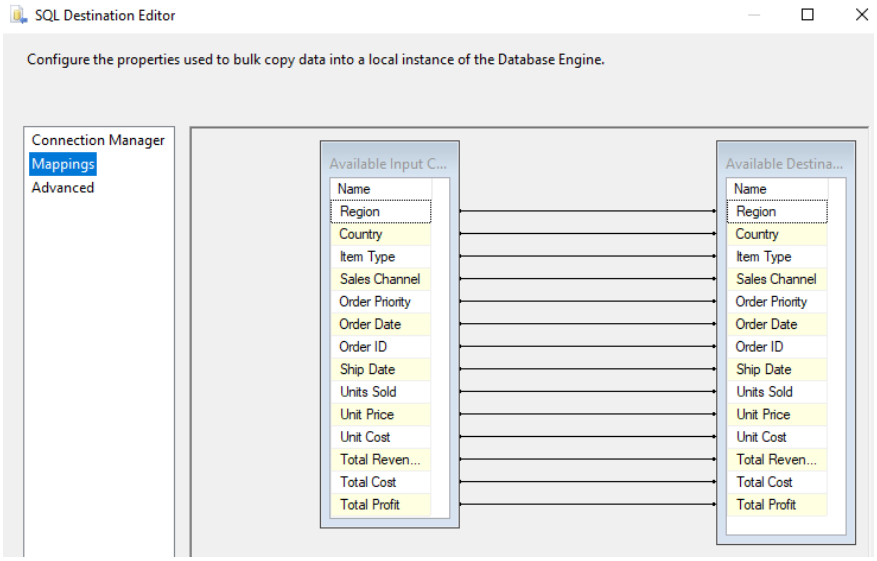

Package.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task

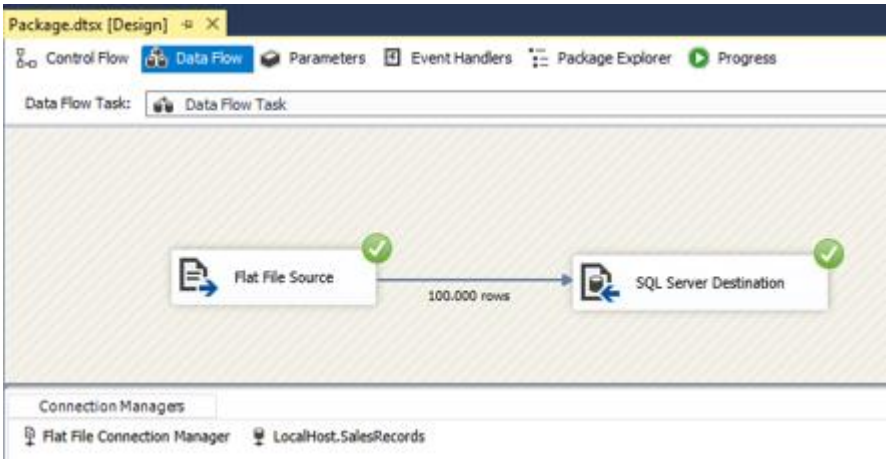
3) On this task we had to assign a specific flow between our csv file and the SQL Server Destination in SSMS. This step was made to assure that our dataset would be loaded in the “staging “table of our database in which we transferred the data by using a query.

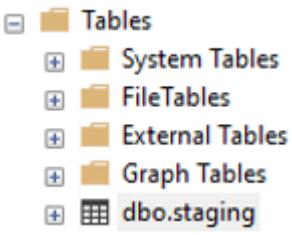


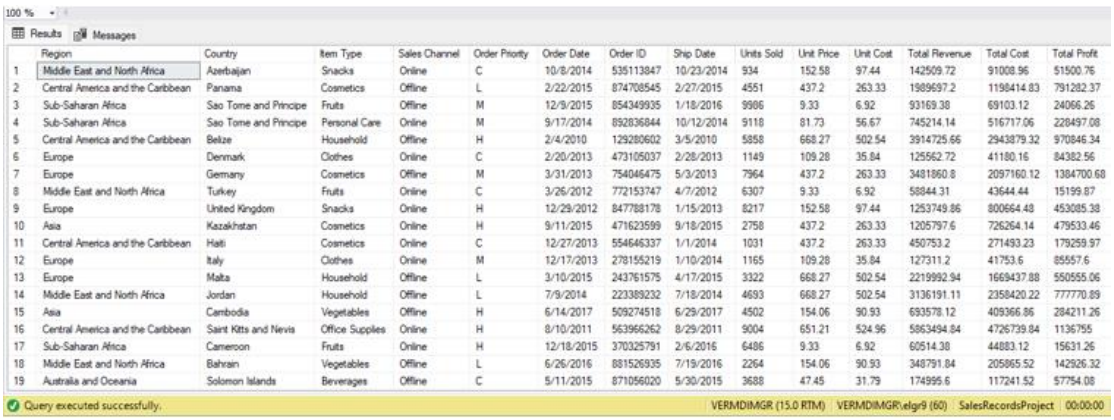
Here you can see the data type for each variable:









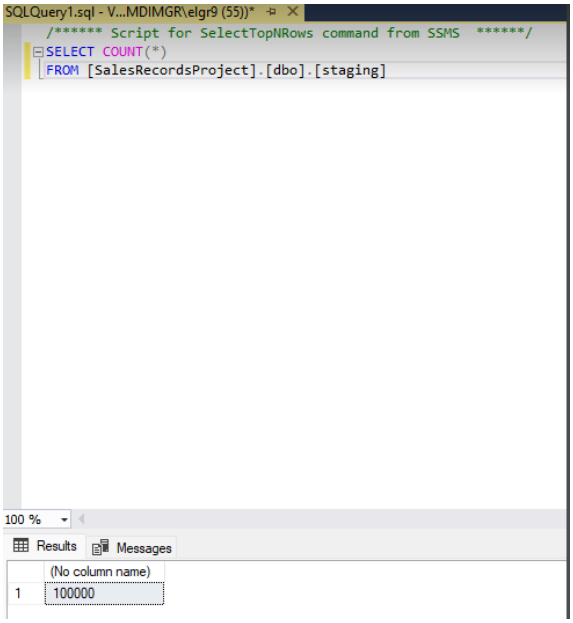


100 %

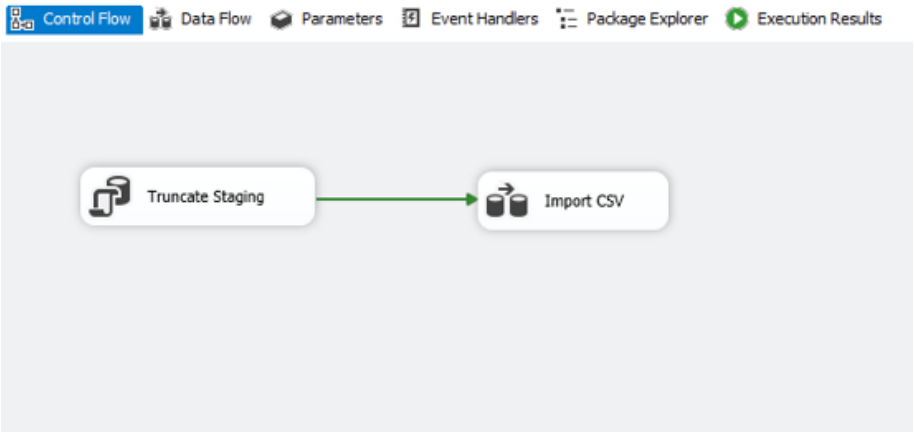
Results | Messages

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
1	Middle East and North Africa	Azerbaijan	Snacks	Online	C	10/8/2014	535113847	10/23/2014	934	152.58	97.44	142509.72	91008.96	51500.76
2	Central America and the Caribbean	Panama	Cosmetics	Offline	L	2/22/2015	874708545	2/27/2015	4551	437.2	263.33	1989697.2	1198414.83	791282.37
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Offline	M	12/9/2015	854349935	1/18/2016	9986	9.33	6.92	93169.38	69103.12	24066.26
4	Sub-Saharan Africa	Sao Tome and Principe	Personal Care	Online	M	9/17/2014	892836844	10/12/2014	9118	81.73	56.67	745214.14	516717.06	228497.08
5	Central America and the Caribbean	Belize	Household	Offline	H	2/4/2010	129280602	3/5/2010	5858	668.27	502.54	3914725.66	2943879.32	970846.34
6	Europe	Denmark	Clothes	Online	C	2/20/2013	473105037	2/28/2013	1149	109.28	35.84	125562.72	41180.16	84382.56
7	Europe	Germany	Cosmetics	Offline	M	3/31/2013	754046475	5/3/2013	7964	437.2	263.33	3481860.8	2097160.12	1384700.68
8	Middle East and North Africa	Turkey	Fruits	Online	C	3/26/2012	772153747	4/7/2012	6307	9.33	6.92	58844.31	43644.44	15199.87
9	Europe	United Kingdom	Snacks	Online	H	12/29/2012	847788178	1/15/2013	8217	152.58	97.44	1253749.86	800664.48	453085.38
10	Asia	Kazakhstan	Cosmetics	Online	H	9/11/2015	471623599	9/18/2015	2758	437.2	263.33	1205797.6	726264.14	479533.46
11	Central America and the Caribbean	Haiti	Cosmetics	Online	C	12/27/2013	554646337	1/1/2014	1031	437.2	263.33	450753.2	271493.23	179259.97
12	Europe	Italy	Clothes	Online	M	12/17/2013	278155219	1/10/2014	1165	109.28	35.84	127311.2	41753.6	85557.6
13	Europe	Malta	Household	Offline	L	3/10/2015	243761575	4/17/2015	3322	668.27	502.54	2219992.94	1669437.88	550555.06
14	Middle East and North Africa	Jordan	Household	Offline	L	7/9/2014	223389232	7/18/2014	4693	668.27	502.54	3136191.11	2358420.22	777770.89
15	Asia	Cambodia	Vegetables	Offline	H	6/14/2017	509274518	6/29/2017	4502	154.06	90.93	693578.12	409366.86	284211.26
16	Central America and the Caribbean	Saint Kitts and Nevis	Office Supplies	Online	H	8/10/2011	563966262	8/29/2011	9004	651.21	524.96	5863494.84	4726739.84	1136755
17	Sub-Saharan Africa	Cameroon	Fruits	Online	H	12/18/2015	370325791	2/6/2016	6486	9.33	6.92	60514.38	44883.12	15631.26
18	Middle East and North Africa	Bahrain	Vegetables	Offline	L	6/26/2016	881526935	7/19/2016	2264	154.06	90.93	348791.84	205865.52	142926.32
19	Australia and Oceania	Solomon Islands	Beverages	Offline	C	5/11/2015	871056020	5/30/2015	3688	47.45	31.79	174995.6	117241.52	57754.08

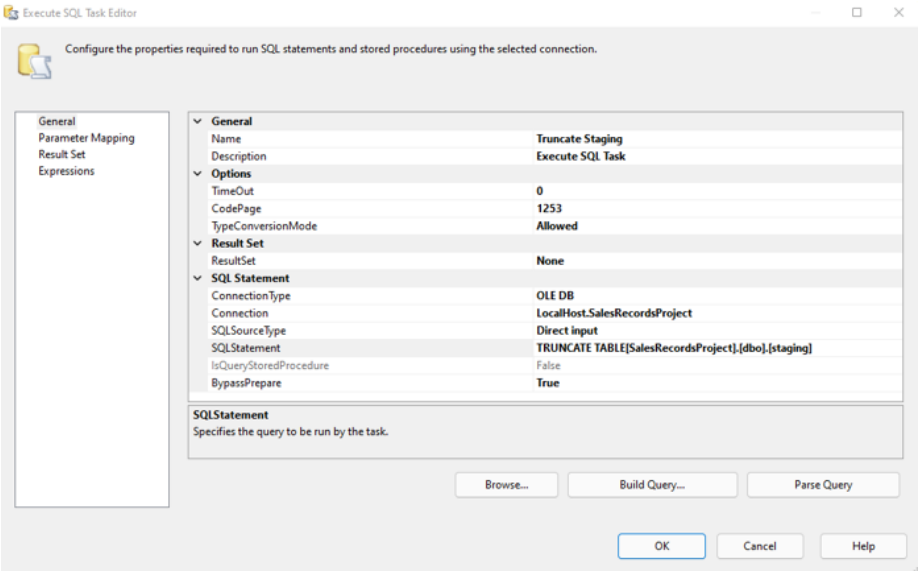
Query executed successfully. | VERMDIMGR (15.0 RTM) | VERMDIMGR\elgr9 (60) | SalesRecordsProject | 00:00:00

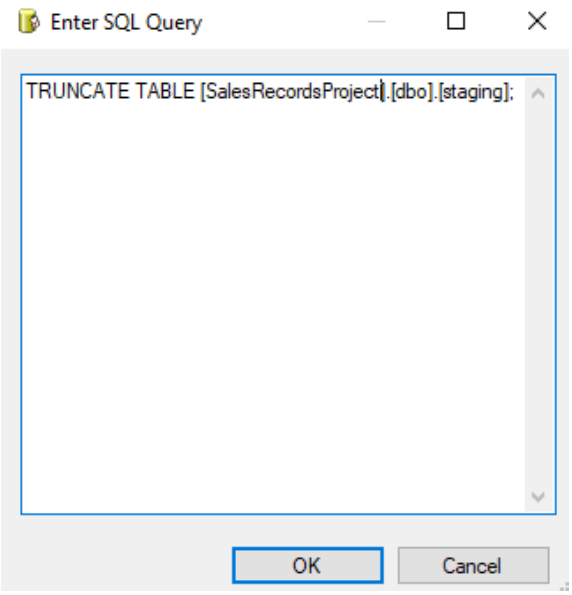


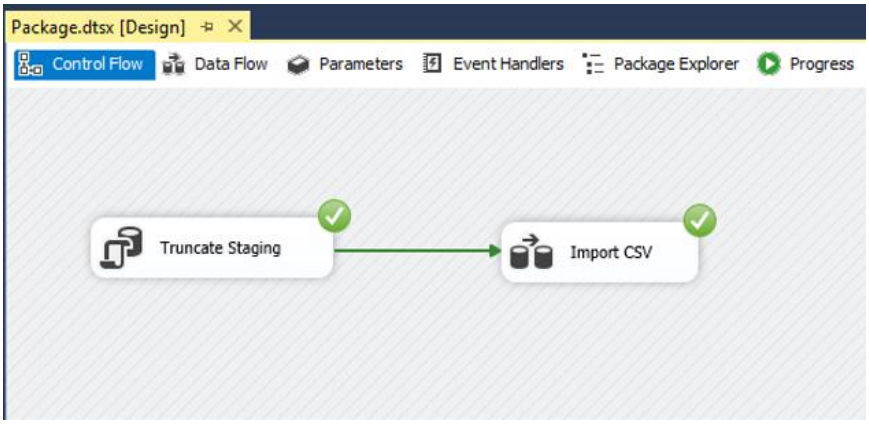
4) The next step that had to be made, was to create the possibility for us to truncate the staging table. This function is crucial for the enterprises that use databases in which the data are increasing every day (new rows of data are being loaded into the databases).



Setting up the connection and the truncate query

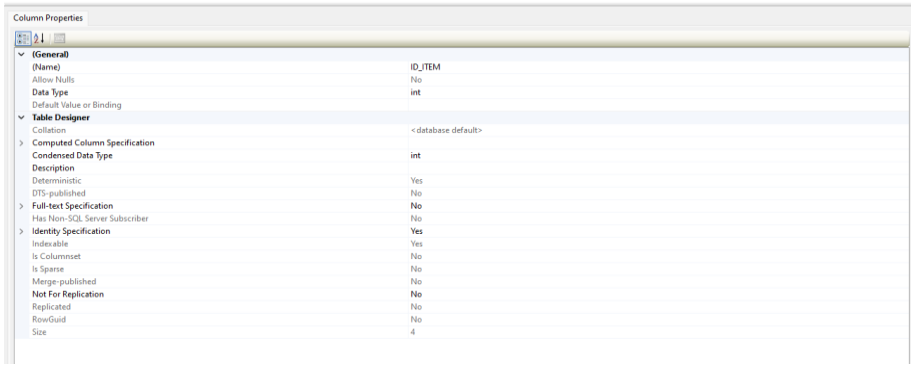




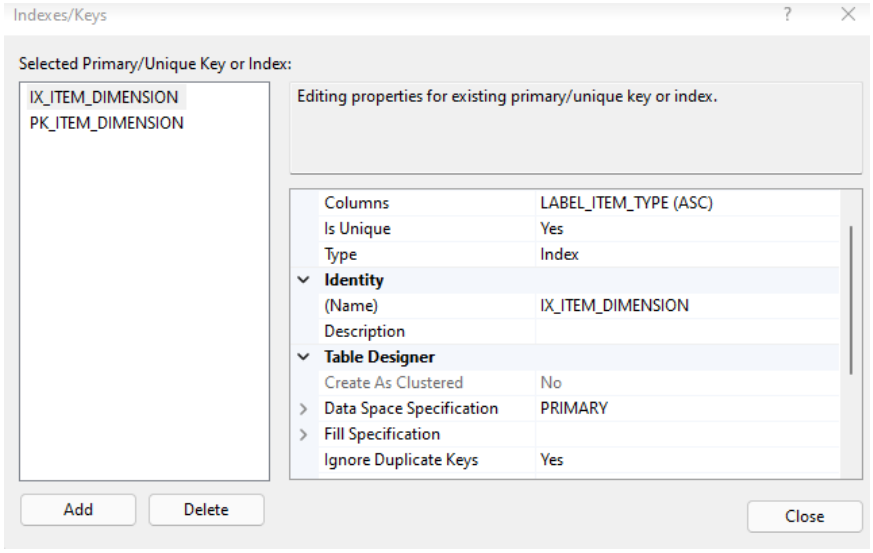


5) Next in line is the creation of the dimension tables. For this task we had to separate the columns of the dataset taking into consideration which ones are metrics and which are not. The ones that do not contain metric information, are suitable to form the dimension tables. Below you can see the step that we followed in order to create the 5-dimension tables of this assignment.

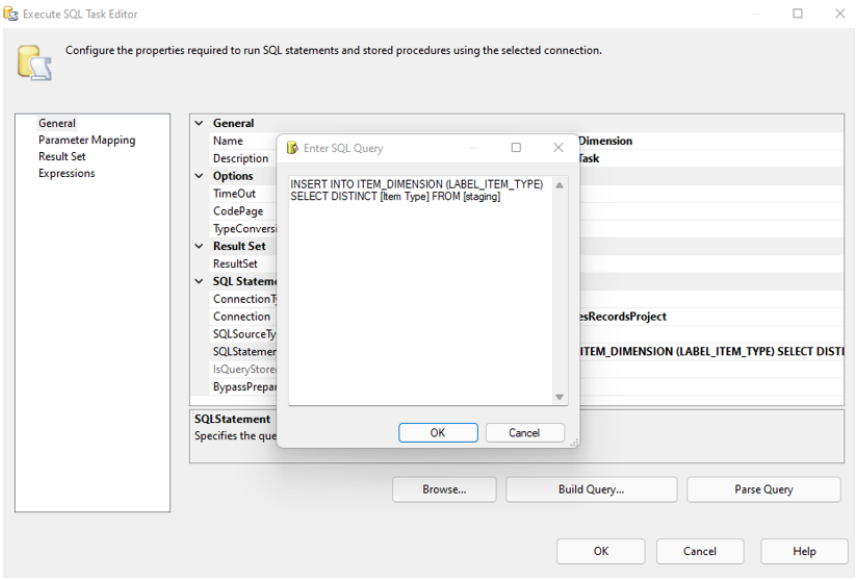
- ITEM DIMENSION: This dimension table consists of two columns. The first one is the item's id (ID_ITEM) and the second one contains the labels of every item type that can be found in the dataset (LABEL_ITEM_TYPE).
  For this task we created a new table into the database. In this table the id column is set to increase one by one every time that a new item type enters the table and we do not allow duplicate values in the item's type column. Then we proceeded in the creation of a new SQL query into the SSIS in order for us to load the appropriate data into the dimension table that exists in the SSMS database.



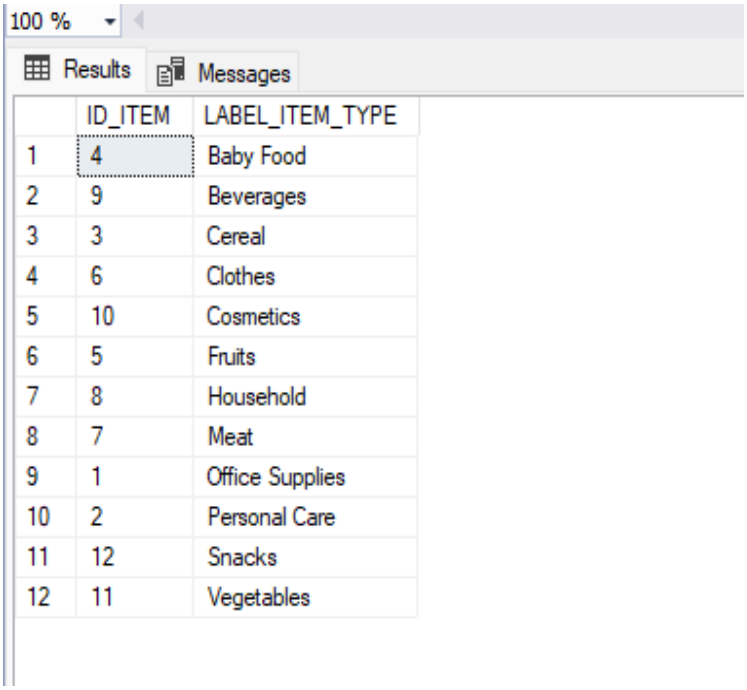
Setting up the Indexes for ITEM DIMENSION



Query for inserting data into dimension



Output of the above process

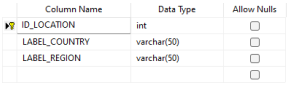


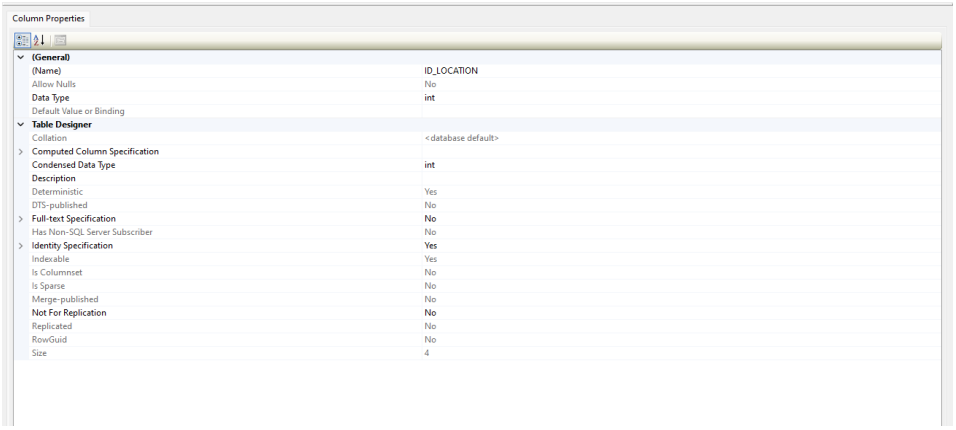
100 %

Results | Messages

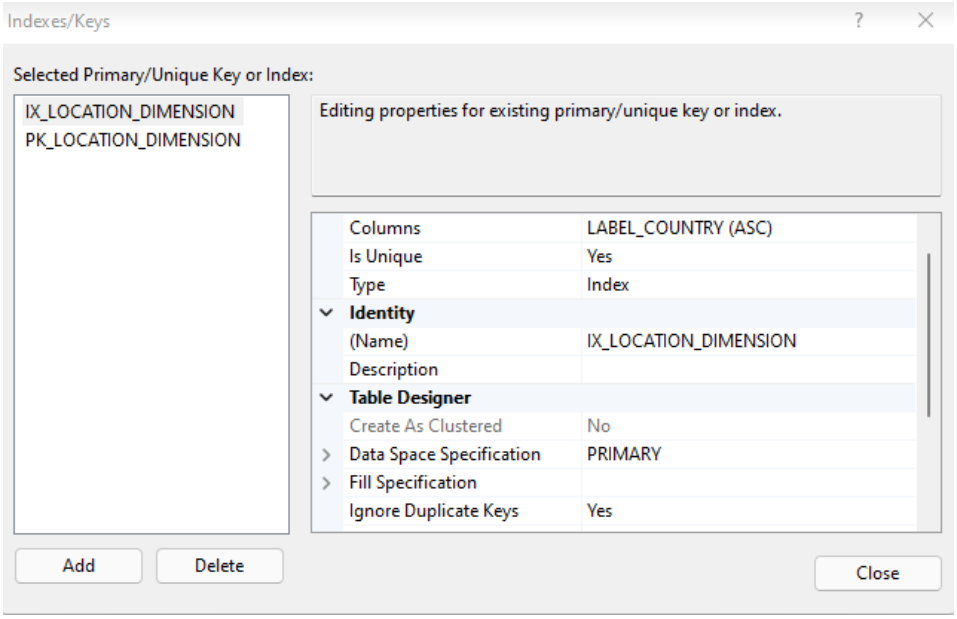
	ID_ITEM	LABEL_ITEM_TYPE
1	4	Baby Food
2	9	Beverages
3	3	Cereal
4	6	Clothes
5	10	Cosmetics
6	5	Fruits
7	8	Household
8	7	Meat
9	1	Office Supplies
10	2	Personal Care
11	12	Snacks
12	11	Vegetables

- LOCATION DIMENSION: This dimension table consists of three columns. The first one is the location's id (ID_LOCATION), the second one contains the labels of every country that can be found in the dataset (LABEL_COUNTRY) and the third one contains the labels of every region that can be found in the dataset.
  For this task we created a new table into the database. In this table the id column is set to increase one by one every time that a new location combination (region-country) enters the table, but we do not allow duplicate values in the country column, because the region may be similar for a variety of countries, but the country should be unique. Then we proceeded in the creation of a new SQL query into the SSIS in order for us to load the appropriate data into the dimension table that exists in the SSMS database.

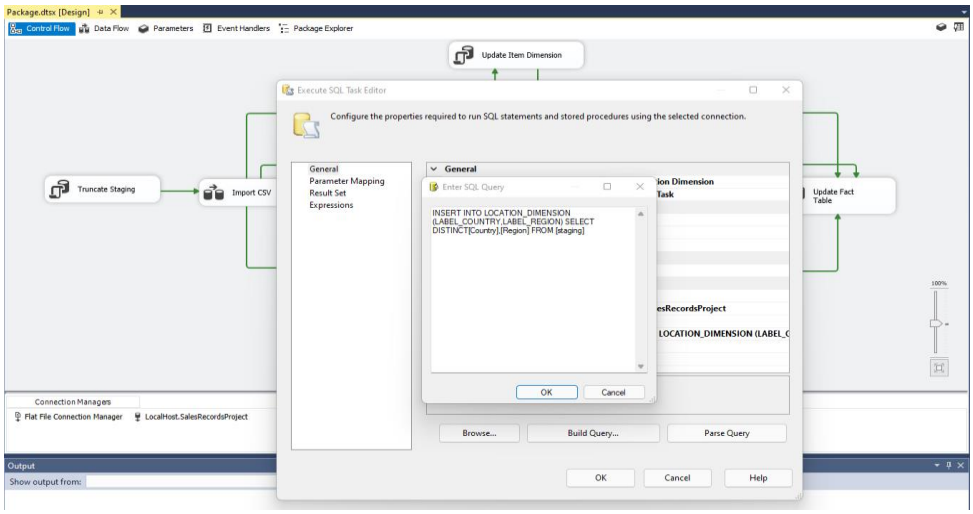




Setting up the Indexes for LOCATION DIMENSION



Query for inserting data into dimension

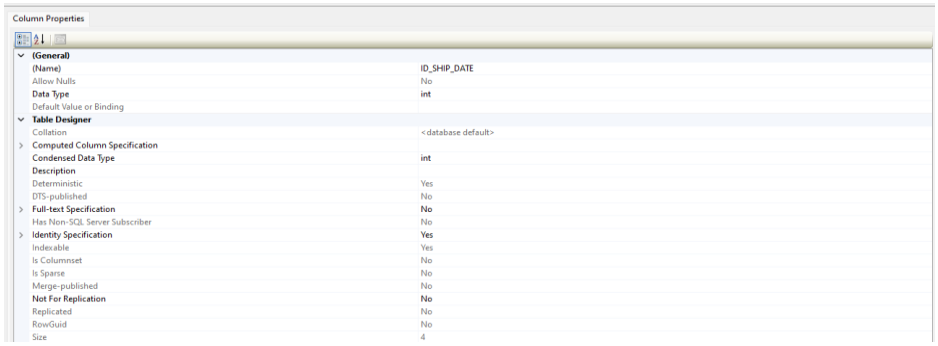


Output of the above process

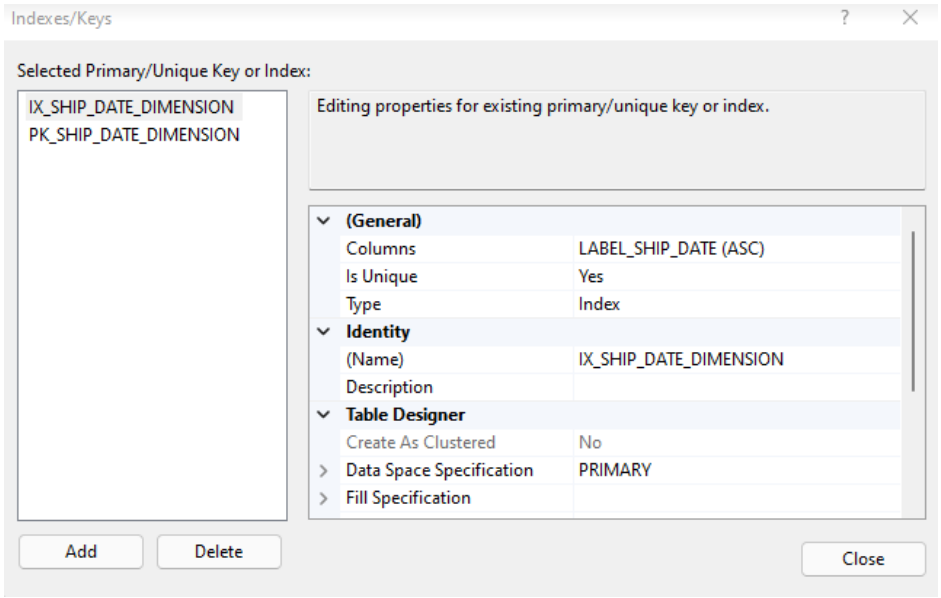
Results | Messages

	ID_LOCATION	LABEL_COUNTRY	LABEL_REGION
1	1	Afghanistan	Middle East and North Africa
2	2	Albania	Europe
3	3	Algeria	Middle East and North Africa
4	4	Andorra	Europe
5	5	Angola	Sub-Saharan Africa
6	6	Antigua and Barbuda	Central America and the Caribbean
7	7	Armenia	Europe
8	8	Australia	Australia and Oceania
9	9	Austria	Europe
10	10	Azerbaijan	Middle East and North Africa
11	11	Bahrain	Middle East and North Africa
12	12	Bangladesh	Asia
13	13	Barbados	Central America and the Caribbean
14	14	Belarus	Europe
15	15	Belgium	Europe
16	16	Belize	Central America and the Caribbean
17	17	Benin	Sub-Saharan Africa
18	18	Bhutan	Asia
19	19	Bosnia and Herzegovina	Europe

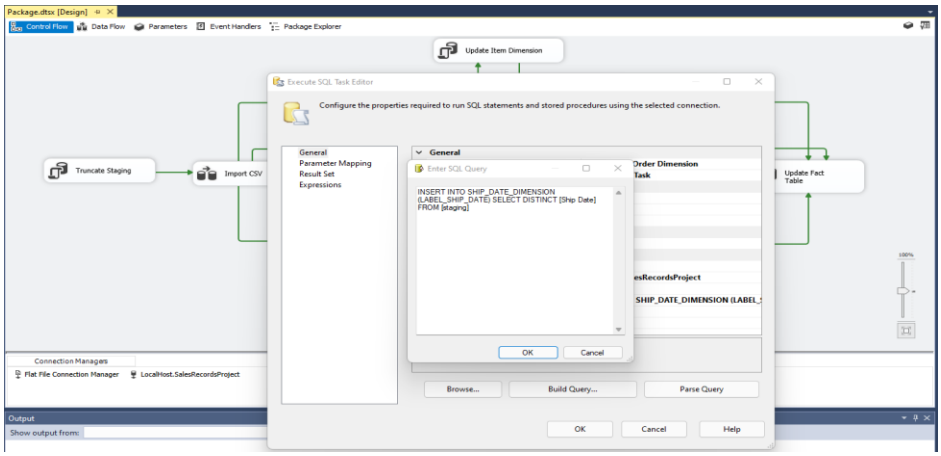
- SHIP DATE DIMENSION: This dimension table consists of two columns. The first one is the ship date's id (ID_SHIP_DATE), and the second one contains each ship date that can be found in the dataset (LABEL_SHIP_DATE).
  For this task we created a new table into the database. In this table the id column is set to increase one by one every time that a new ship date enters the table and we do not allow duplicate values in the date's column, because each ship date should be unique. Then we proceeded in the creation of a new SQL query into the SSIS in order for us to load the appropriate data into the dimension table that exists in the SSMS database.



Setting up the Indexes for SHIP DATE DIMENSION



Query for inserting data into dimension

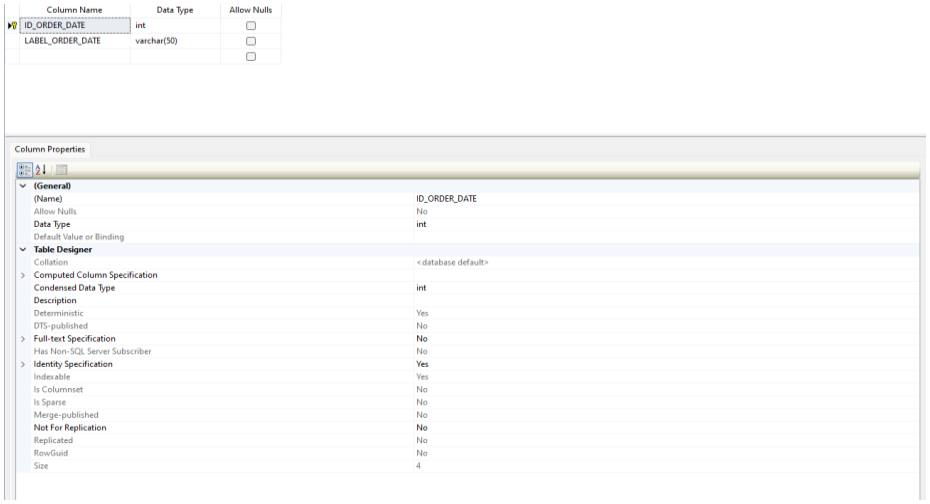


Output of the above process

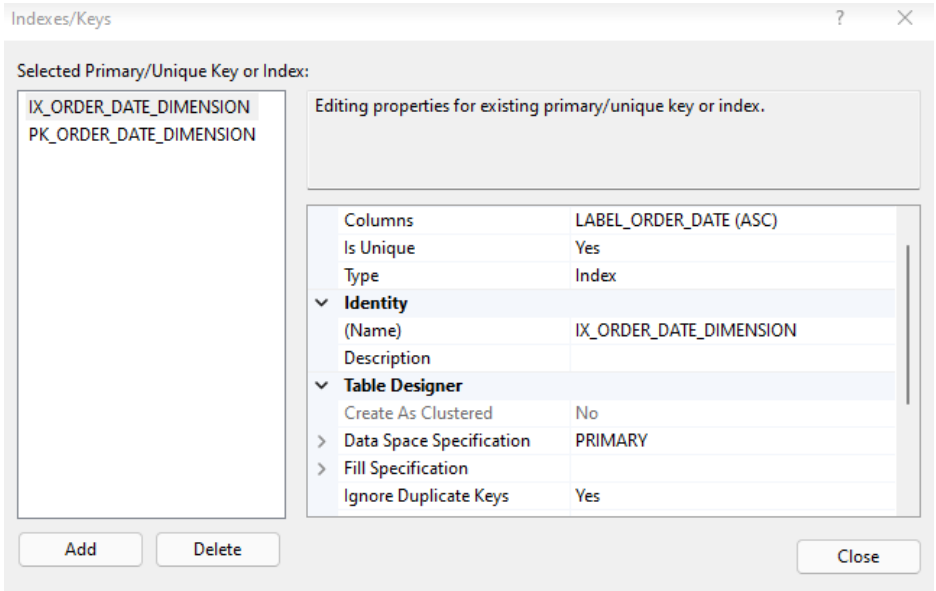
Results Messages

	ID_SHIP_DATE	LABEL_SHIP_DATE
1	1	1/1/2011
2	2	1/1/2012
3	3	1/1/2013
4	4	1/1/2014
5	5	1/1/2015
6	6	1/1/2016
7	7	1/1/2017
8	8	1/10/2010
9	9	1/10/2011
10	10	1/10/2012
11	11	1/10/2013
12	12	1/10/2014
13	13	1/10/2015
14	14	1/10/2016
15	15	1/10/2017
16	16	1/11/2010
17	17	1/11/2011
18	18	1/11/2012
19	19	1/11/2013

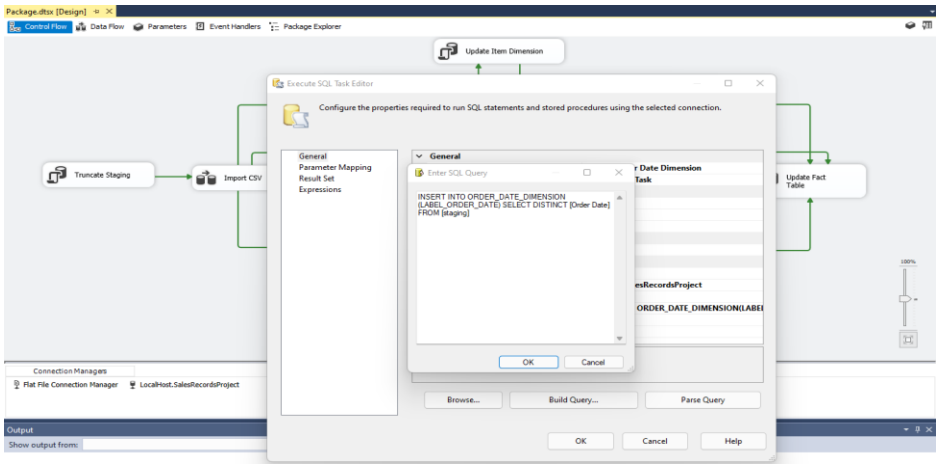
- ORDER DATE DIMENSION: This dimension table consists of two columns. The first one is the ship date's id (ID_ORDER_DATE), and the second one contains each order date that can be found in the dataset (LABEL_ORDER_DATE).
  For this task we created a new table into the database. In this table the id column is set to increase one by one every time that a new order date enters the table and we do not allow duplicate values in the date's column, because each order date should be unique. Then we proceeded in the creation of a new SQL query into the SSIS in order for us to load the appropriate data into the dimension table that exists in the SSMS database.



Setting up the Indexes for ORDER DATE DIMENSION



Query for inserting data into dimension

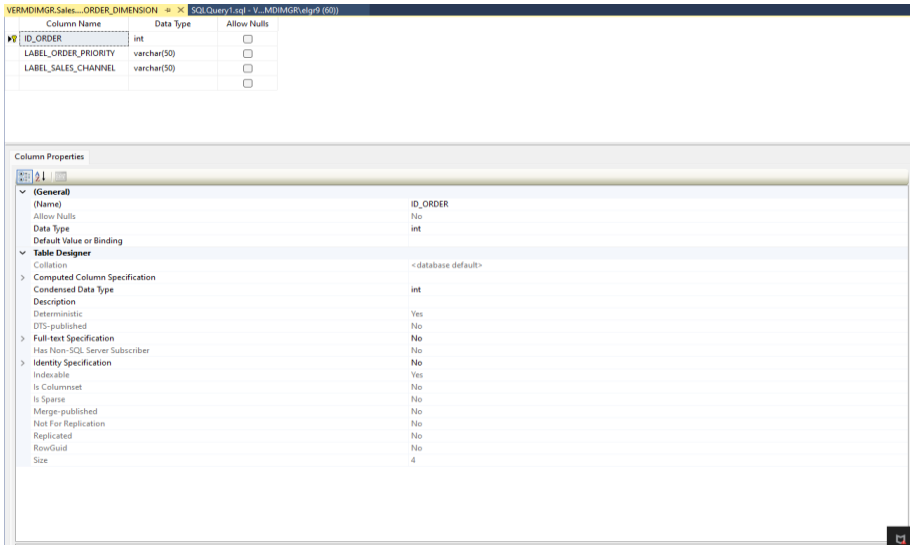


Output of the above process

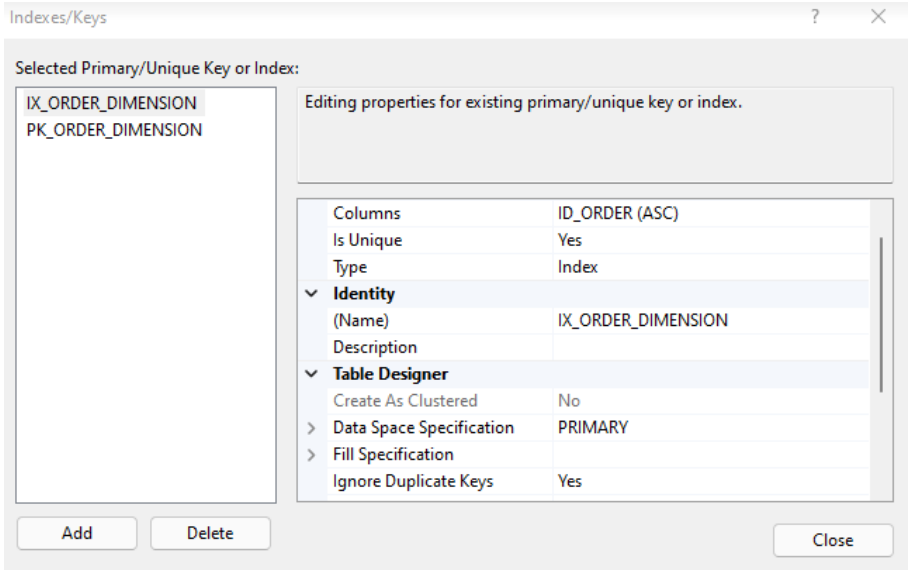
Results | Messages

	ID_ORDER_DATE	LABEL_ORDER_DATE
1	1	1/1/2010
2	2	1/1/2011
3	3	1/1/2012
4	4	1/1/2013
5	5	1/1/2014
6	6	1/1/2015
7	7	1/1/2016
8	8	1/1/2017
9	9	1/10/2010
10	10	1/10/2011
11	11	1/10/2012
12	12	1/10/2013
13	13	1/10/2014
14	14	1/10/2015
15	15	1/10/2016
16	16	1/10/2017
17	17	1/11/2010
18	18	1/11/2011
19	19	1/11/2012

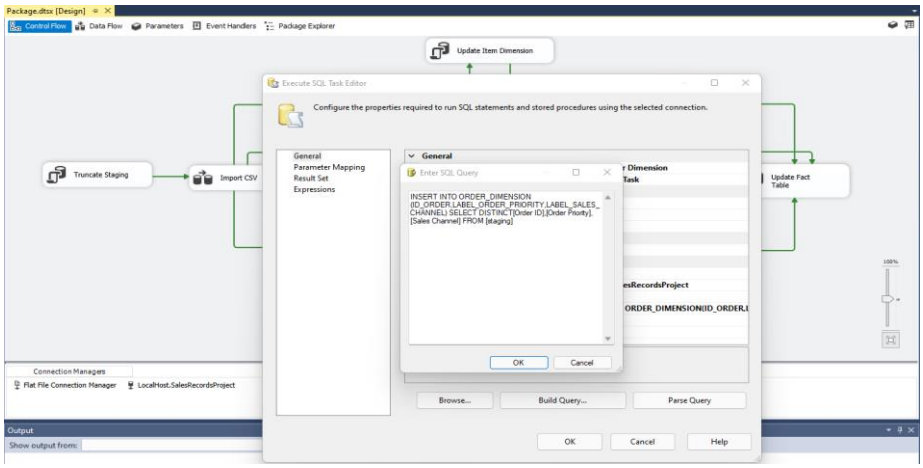
- ORDER DIMENSION: This dimension table consists of three columns. The first one is the order id (ID_ORDER), the second one contains each order's priority (LABEL_ORDER_PRIORITY) and the third one contains each order's sales channel (LABEL_SALES_CHANNEL).
  For this task we created a new table into the database. In this table the id column is taken from the original dataset, because we thought that an order id has a specific meaning for every organization and it's unique. For this reason we do not allow duplicate values in the id column. Then we proceeded in the creation of a new SQL query into the SSIS in order for us to load the appropriate data into the dimension table that exists in the SSMS database.



Setting up the Indexes for ORDER DIMENSION



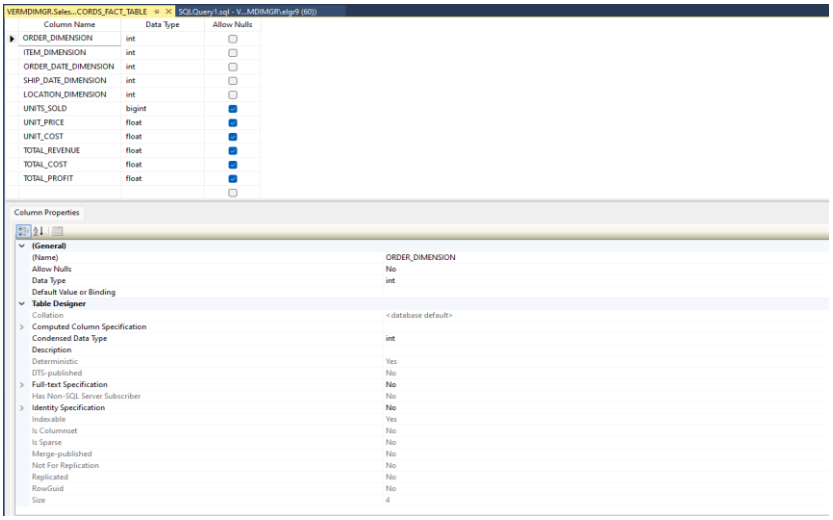
Query for inserting data into dimension



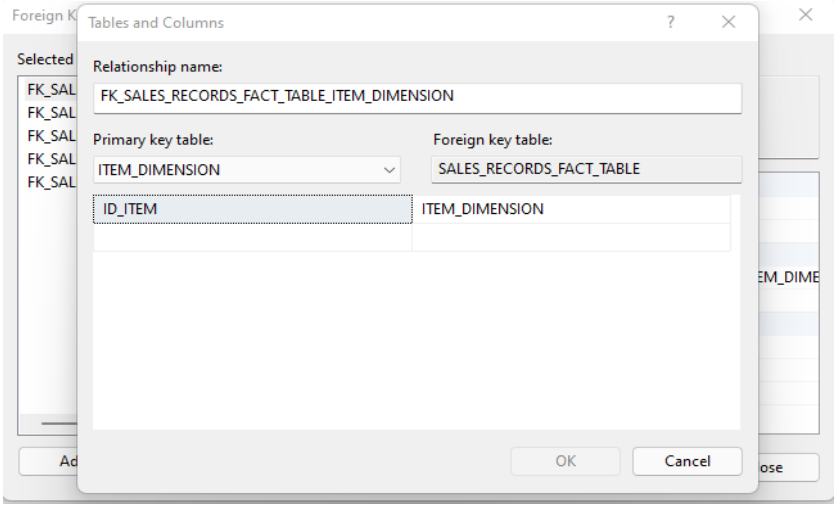
Output of the above process

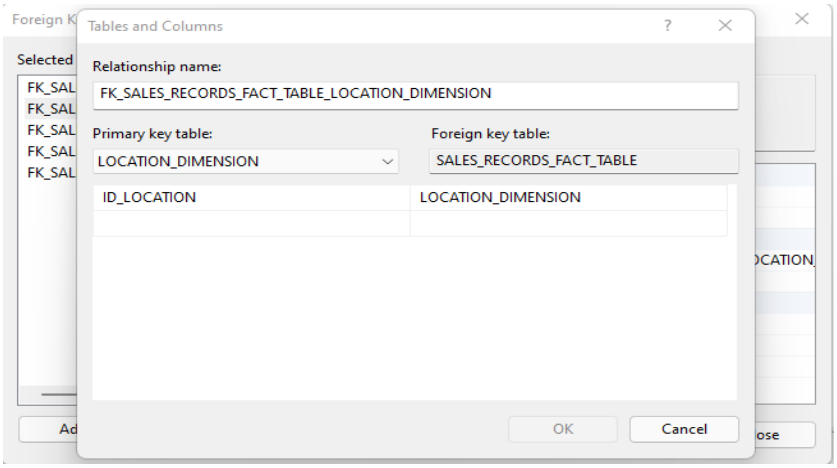
	ID_ORDER	LABEL_ORDER_PRIORITY	LABEL_SALES_CHANNEL
1	100008904	L	Offline
2	100009763	M	Offline
3	100035941	C	Offline
4	100043666	L	Offline
5	100050961	H	Online
6	100051820	C	Online
7	100054824	C	Online
8	100062119	L	Online
9	100069415	M	Online
10	100077140	C	Online
11	100088727	M	Online
12	100089585	C	Online
13	100105893	C	Online
14	100106751	L	Online
15	100109755	L	Online
16	100110614	M	Online
17	100114047	H	Online
18	100147092	C	Offline
19	100155675	H	Offline

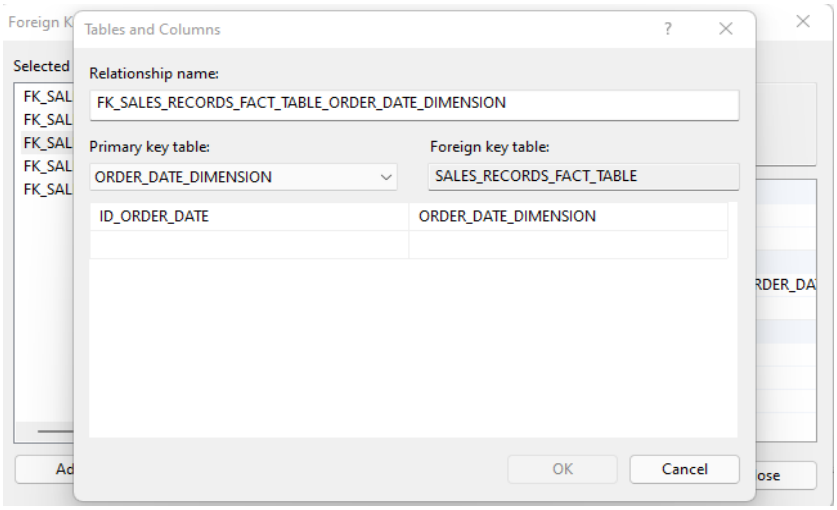
6) After the dimension tables, we had to create the fact table in order for us to connect them together. The fact table contains the metrics of the original dataset and the primary keys of each dimension table. These keys are the foreign keys of the fact table and we connected them with their origin tables creating constraints. Also, we allowed null values in the metrics' columns, because even though our dataset is clean, in a possible new data insertion, some rows could possibly contain null values (it's ordinary for organizations). Last but not least, we proceeded in the creation of a new SQL query into the SSIS in order for us to load the appropriate data into the fact table that exists in the SSMS database.

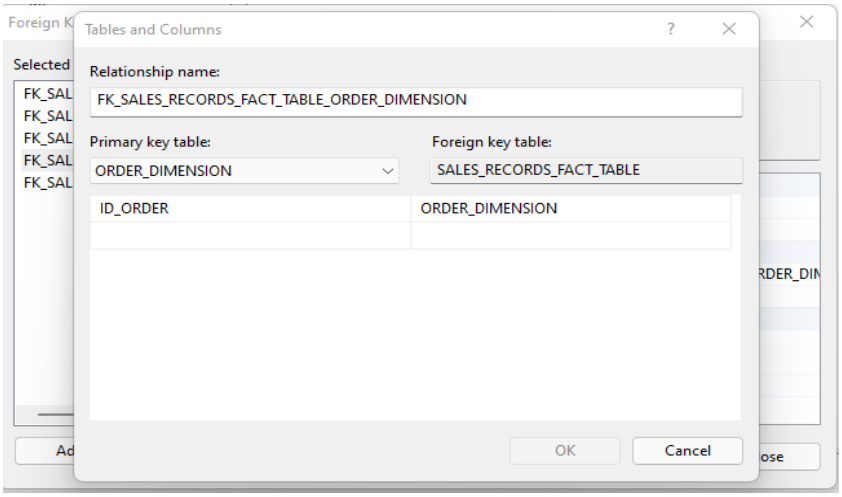


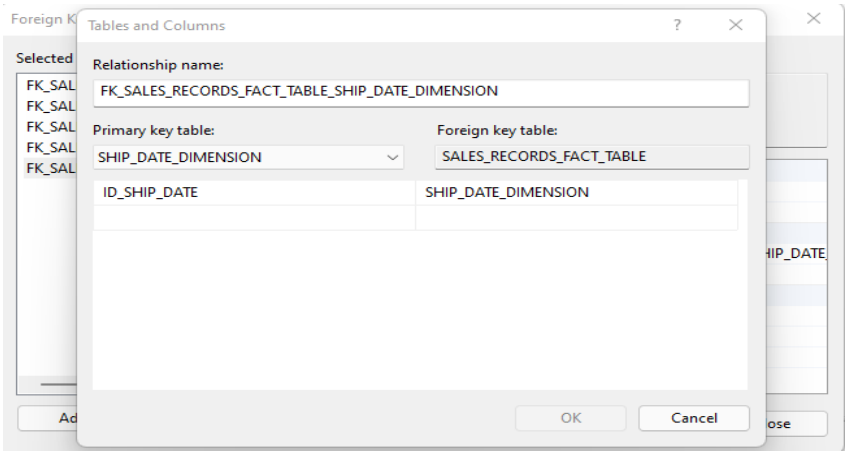
Setting up the foreign keys

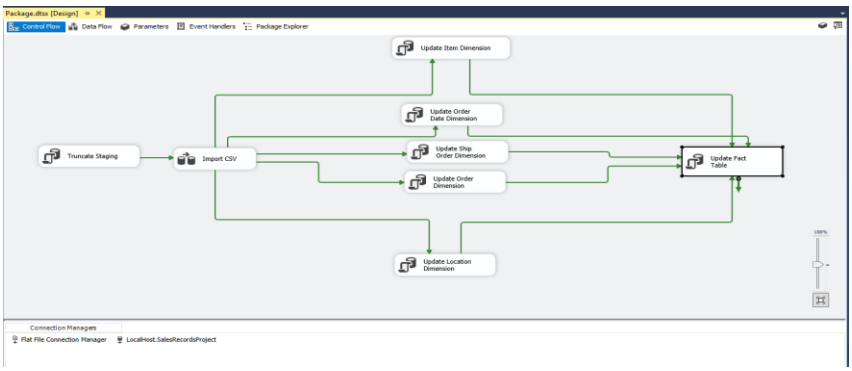



Tables and Columns
Relationship name:
FK_SALES_RECORDS_FACT_TABLE_LOCATION_DIMENSION
Primary key table:
LOCATION_DIMENSION
Foreign key table:
SALES_RECORDS_FACT_TABLE
ID_LOCATION
LOCATION_DIMENSION
OK
Cancel

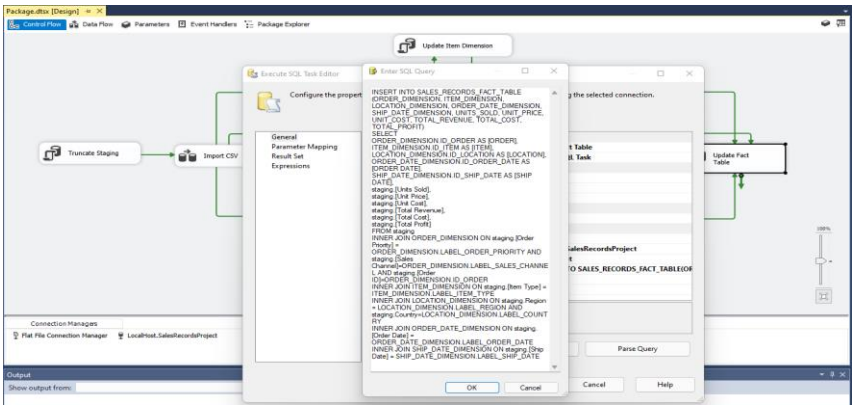

Tables and Columns
Relationship name:
FK_SALES_RECORDS_FACT_TABLE_ORDER_DATE_DIMENSION
Primary key table:
ORDER_DATE_DIMENSION
Foreign key table:
SALES_RECORDS_FACT_TABLE
ID_ORDER_DATE
ORDER_DATE_DIMENSION
OK
Cancel

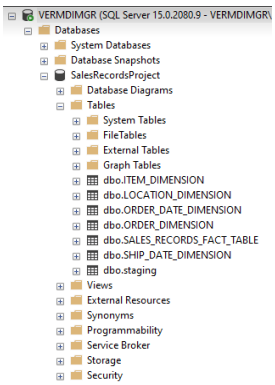

Tables and Columns
Relationship name:
FK_SALES_RECORDS_FACT_TABLE_ORDER_DIMENSION
Primary key table:
ORDER_DIMENSION
Foreign key table:
SALES_RECORDS_FACT_TABLE
ID_ORDER
ORDER_DIMENSION
OK
Cancel





Setting up the insertion query



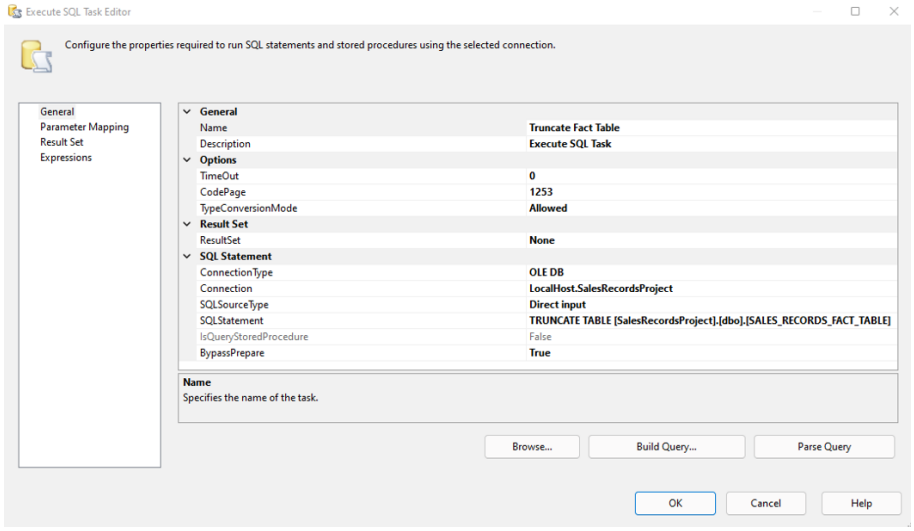


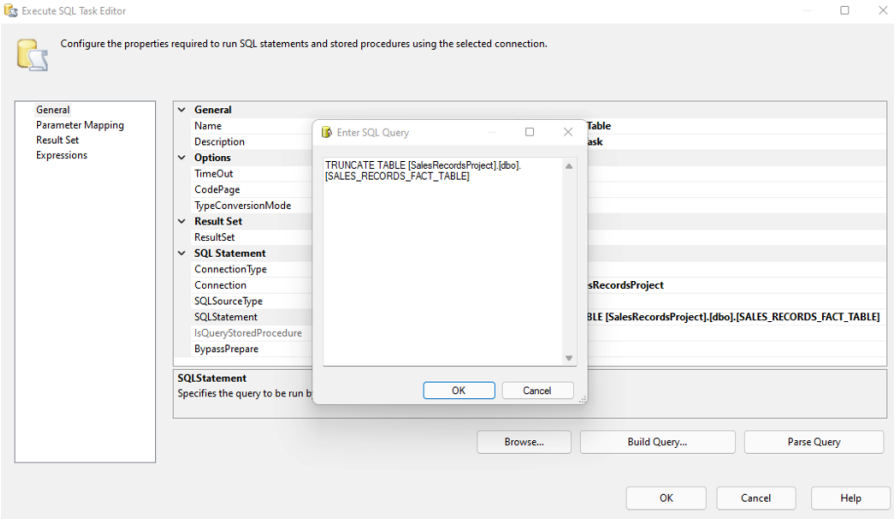
Output of the above process

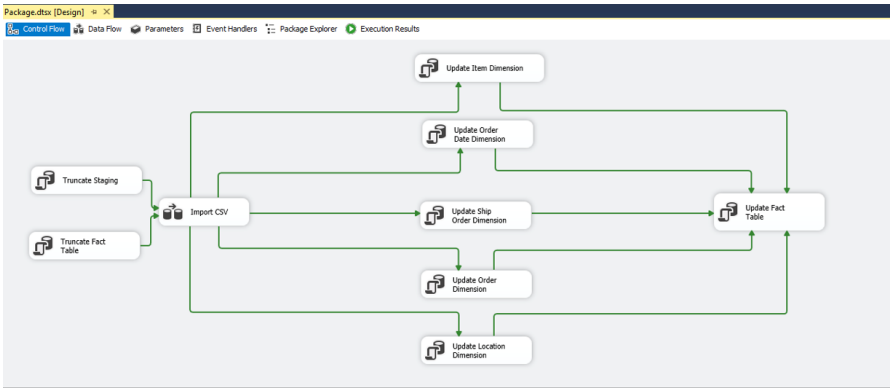
Results Messages

	ORDER_DIMENSION	ITEM_DIMENSION	ORDER_DATE_DIMENSION	SHIP_DATE_DIMENSION	LOCATION_DIMENSION	UNITS_SOLD	UNIT_PRICE	UNIT_COST	TOTAL_REVENUE	TOTAL_COST	TOTAL_PROFIT
1	607436072	5	1395	1769	61	2687	9.33	6.92	25069.71	18594.04	6475.67
2	607465255	12	403	450	84	1323	152.58	97.44	201863.34	128913.12	72950.22
3	607475554	12	2126	2140	91	4371	152.58	97.44	666927.18	425910.24	241016.94
4	607495725	7	406	425	108	340	421.89	364.69	143442.6	123994.6	19448
5	607518041	3	1418	1656	23	6943	205.7	117.11	1428175.1	813094.73	615080.37
6	607524907	3	1255	1565	174	8975	205.7	117.11	1846157.5	1051062.25	795095.25
7	607527482	11	2618	357	47	9737	154.06	90.93	1500082.22	885385.41	614696.81
8	607532632	12	2112	2382	92	1261	152.58	97.44	192403.38	122871.84	69531.54
9	607539498	9	55	198	125	3293	47.45	31.79	156252.85	104684.47	51568.38
10	607539927	5	2445	2751	2	3420	9.33	6.92	31908.6	23666.4	8242.2
11	607540357	8	377	543	184	3547	668.27	502.54	2370353.69	1782509.38	587844.31
12	607547652	4	887	34	64	5706	255.28	159.42	1456627.68	909650.52	546977.16
13	607548511	6	1421	1627	152	5960	109.28	35.84	651308.8	213606.4	437702.4
14	607555377	6	1258	1456	7	7992	109.28	35.84	873365.76	286433.28	586932.48
15	607557094	5	1635	1969	119	8500	9.33	6.92	79305	58820	20485
16	607563960	11	1472	1638	84	532	154.06	90.93	81959.92	48374.76	33585.16
17	607566535	2	283	617	167	1294	81.73	56.67	105758.62	73330.98	32427.64
18	607575976	12	2425	2454	116	4088	152.58	97.44	623747.04	398334.72	225412.32
19	607581126	11	238	244	117	5612	154.06	90.93	864584.72	510299.16	354285.56
20	607581984	3	320	675	174	5866	205.7	117.11	1206636.2	686967.26	519668.94
21	607586705	5	204	974	81	7263	9.33	6.92	67763.79	50259.96	17503.83
22	607594001	2	959	1135	96	9422	81.73	56.67	770060.06	533944.74	236115.32
23	607611596	4	1887	2341	18	4629	255.28	159.42	1181691.12	737955.18	443735.94
24	607619750	10	756	747	3	7042	437.2	263.33	3078762.4	1854369.86	1224392.54
25	607623183	10	1333	1587	176	8058	437.2	263.33	3522957.6	2121913.14	1401044.46
26	607625758	1	197	1097	127	8819	651.21	524.96	5743020.99	4629622.24	1113398.75
27	607642924	7	1255	1549	5	3899	421.89	364.69	1644949.11	1421926.31	223022.8
28	607668673	4	227	1087	21	1519	255.28	159.42	387770.32	242158.98	145611.34
29	607672107	4	1387	1761	164	2535	255.28	159.42	647134.8	404129.7	243005.1

7) Last but not least, we had to add a possibility for us to truncate the fact table every time we execute the whole process, because we don't want to overload our fact table with unnecessary information that already exists.

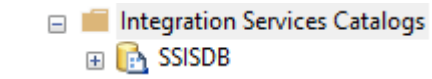


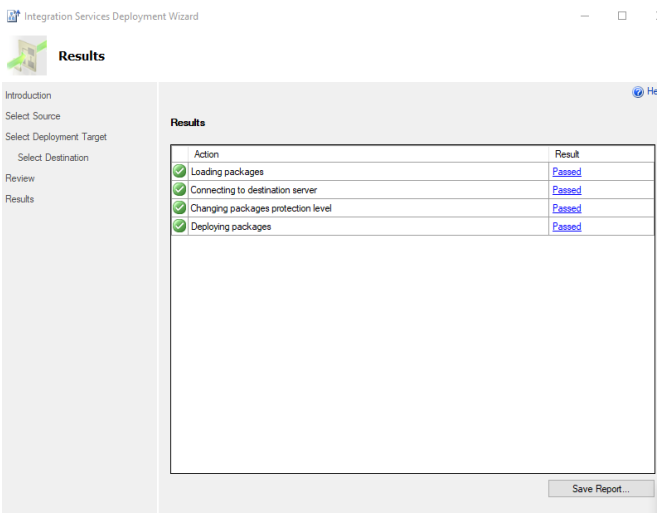


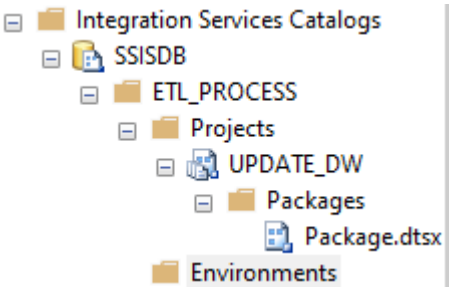

Package.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Execution Results
Update Item Dimension
Update Order Date Dimension
Truncate Staging
Import CSV
Update Ship Order Dimension
Update Fact Table
Truncate Fact Table
Update Order Dimension
Update Location Dimension

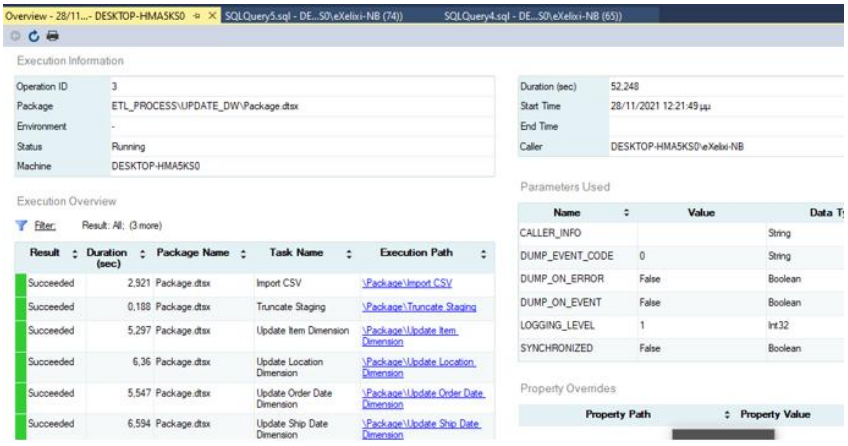
DEPLOYMENT OF THE ETL PACKAGE

After the successful ETL process and before we move into the creation of the cube, we had to automate the above processes. This should be done, because in an organizational environment everything should work automatically, including the ETL process (truncation of the staging table and insertion of data). For this reason, we created a catalog through the SSIS and loaded it with the ETL process. Then we enabled the SQL Server agent and set a specific time during the day in which the system would be updated automatically by creating a new "job".

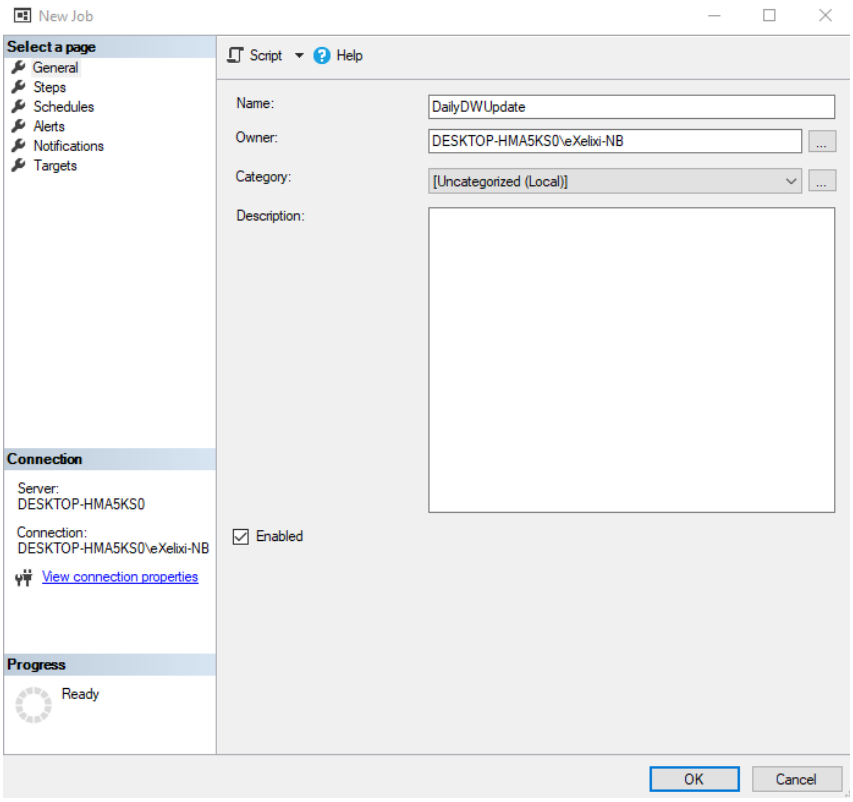




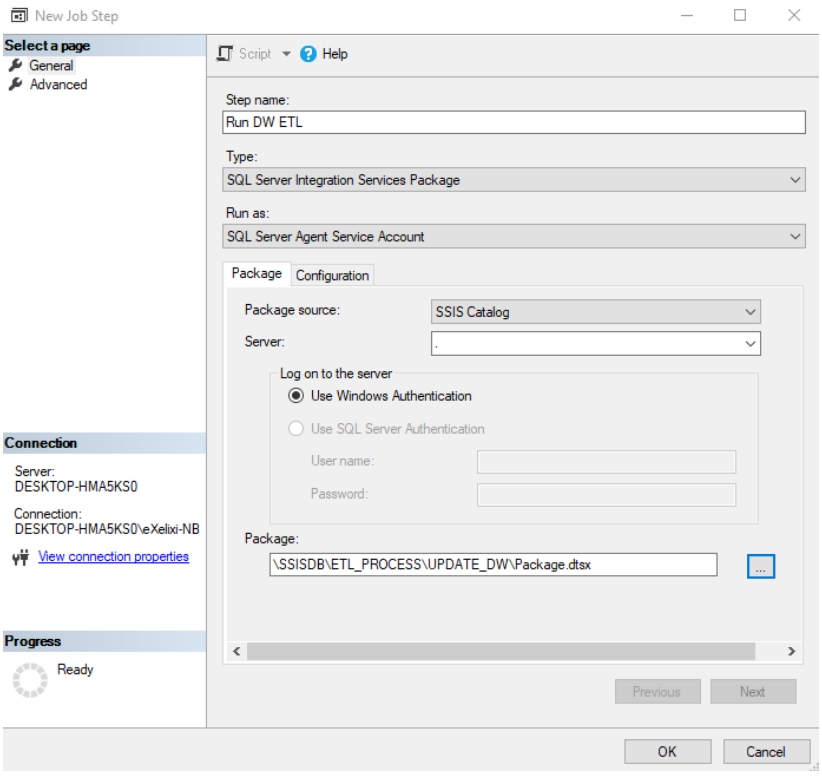


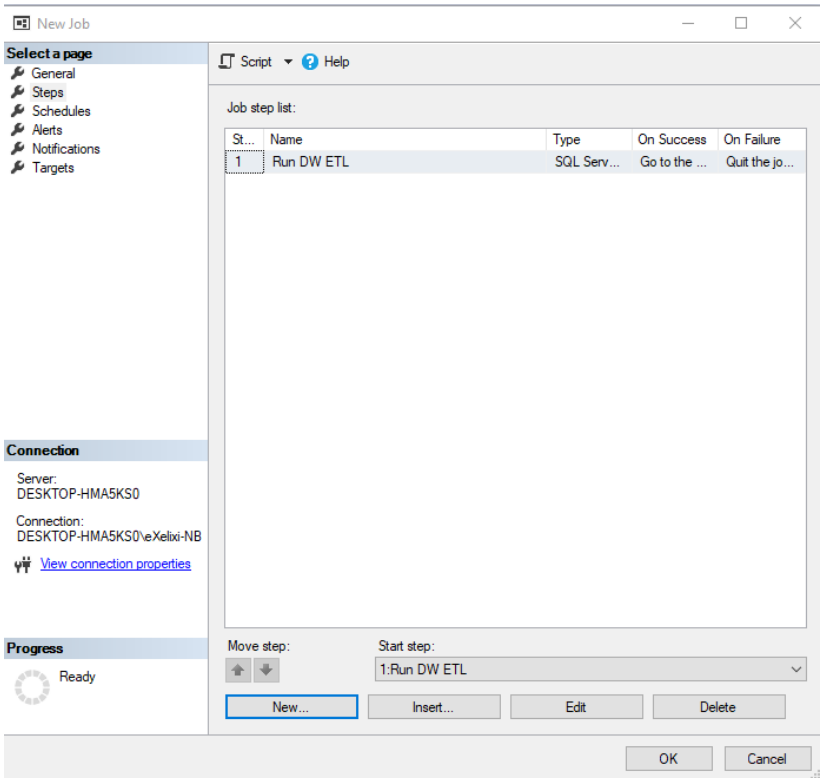


SQL Server Agent ”SQL Server is on”

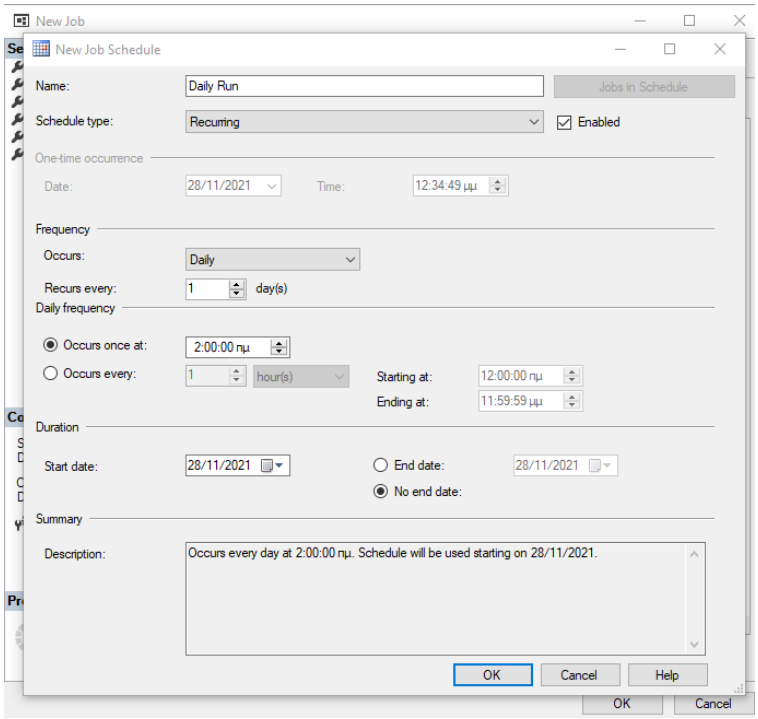


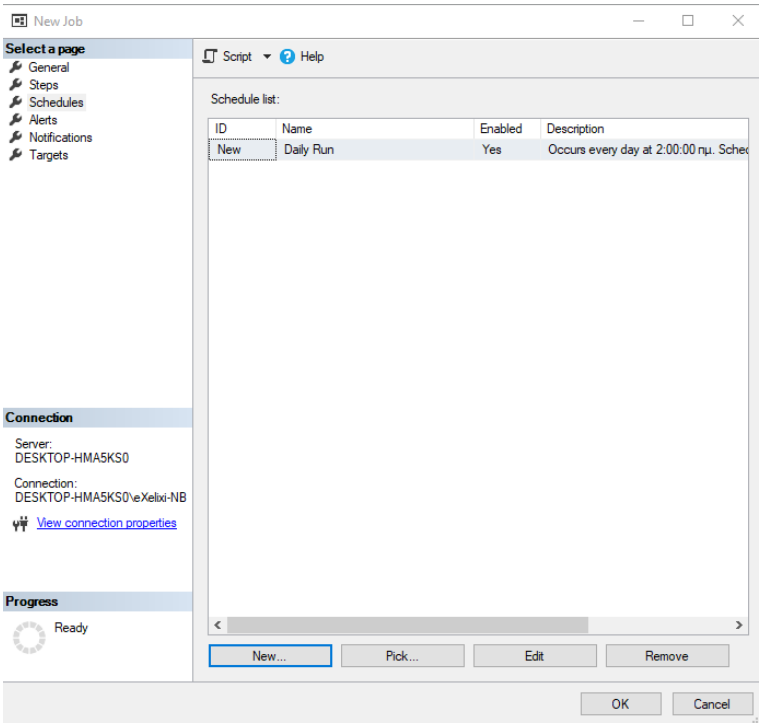
We’ve named this step “Run DW ETL”

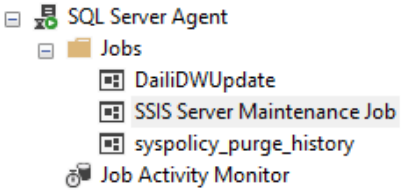




We've scheduled this new "job" to run every night at 2am without having an end date.

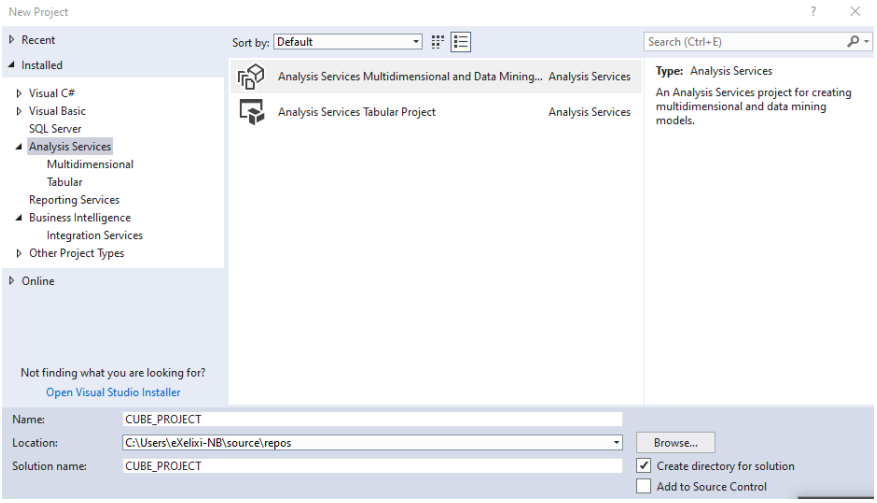


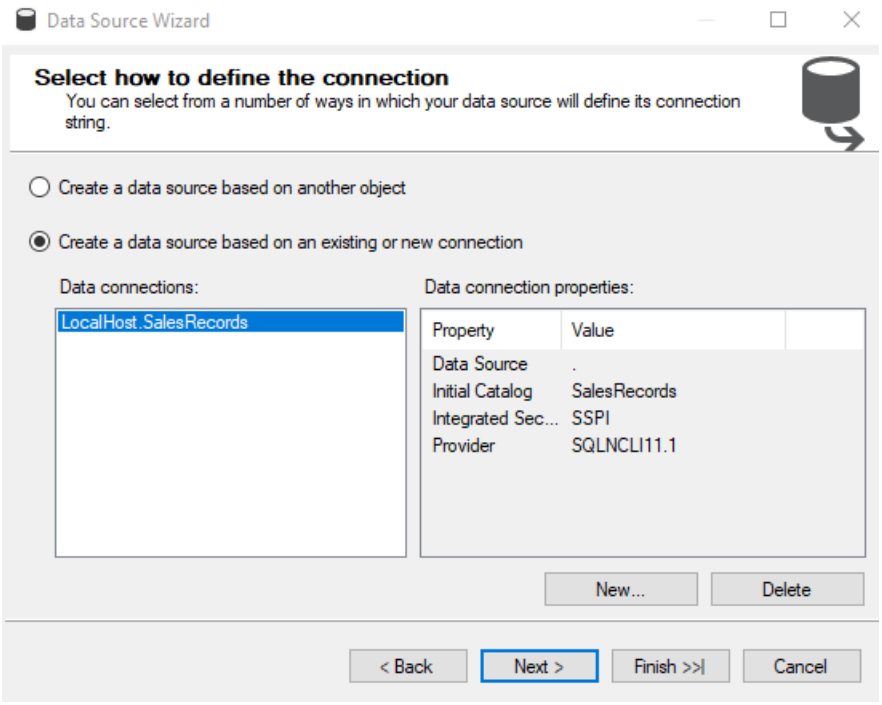

New Job
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
Schedule list:
ID
Name
Enabled
Description
New
Daily Run
Yes
Occurs every day at 2:00:00 πμ. Schec
Connection
Server:
DESKTOP-HMA5KS0
Connection:
DESKTOP-HMA5KS0\eXelixi-NB
View connection properties
Progress
Ready
New...
Pick...
Edit
Remove
OK
Cancel

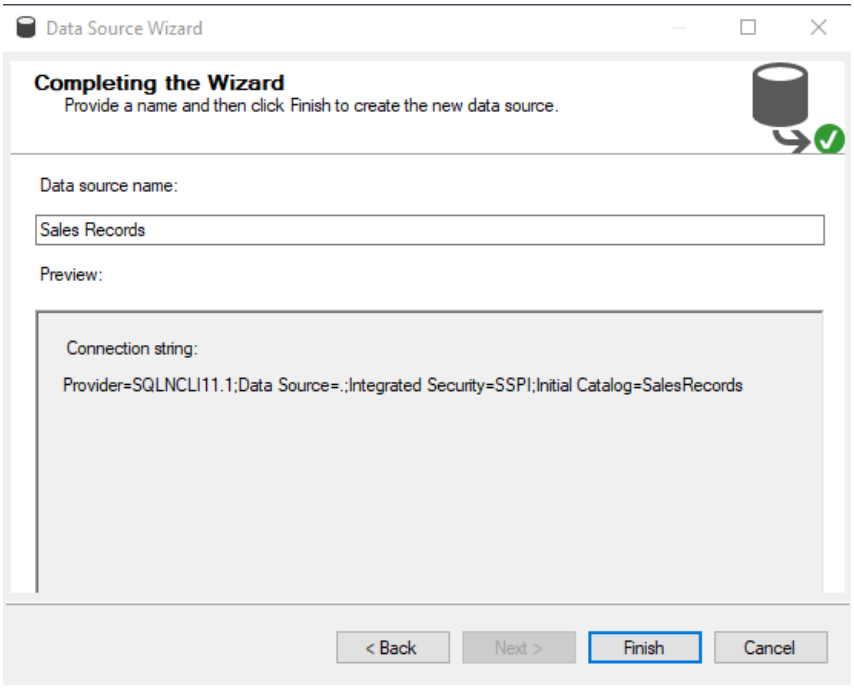

SQL Server Agent
Jobs
DailiDWUpdate
SSIS Server Maintenance Job
syspolicy_purge_history
Job Activity Monitor

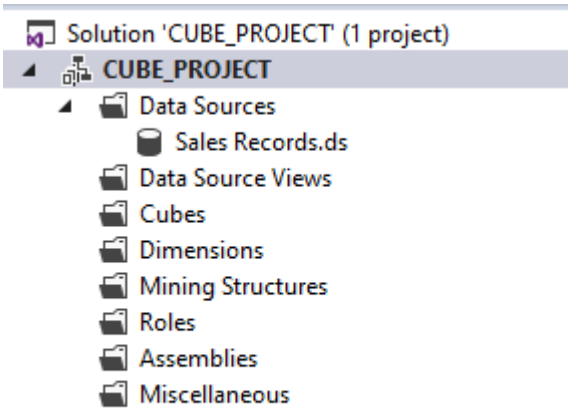
CREATION OF THE CUBE

Next in line is the creation of the cube. Having conducted the above processes, we are ready to use the SSAS in order for us to create a multidimensional cube, which is built using OLAP databases, for deeper and faster data analysis. For this reason, we create a new (cube) project in the SSDT environment, and we connect it with our local server and our database in SSMS.



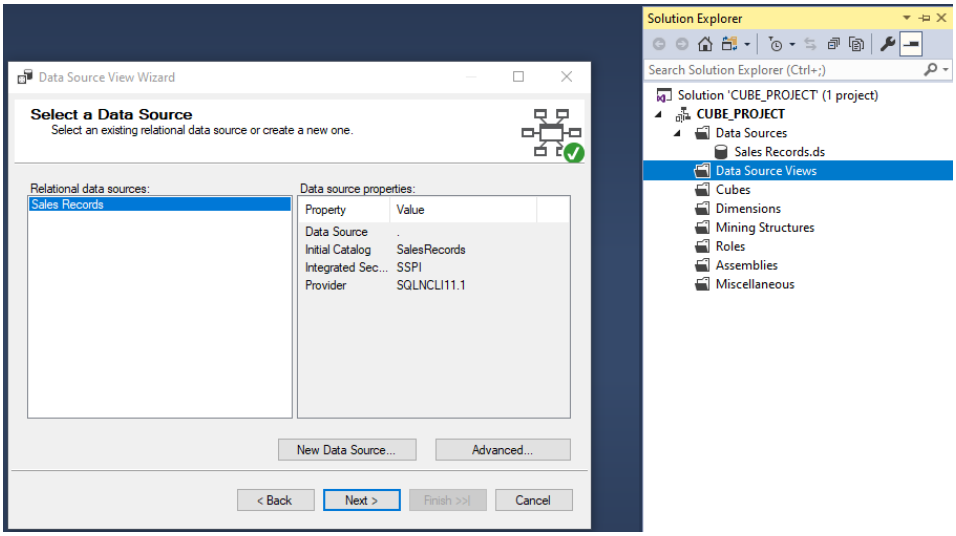


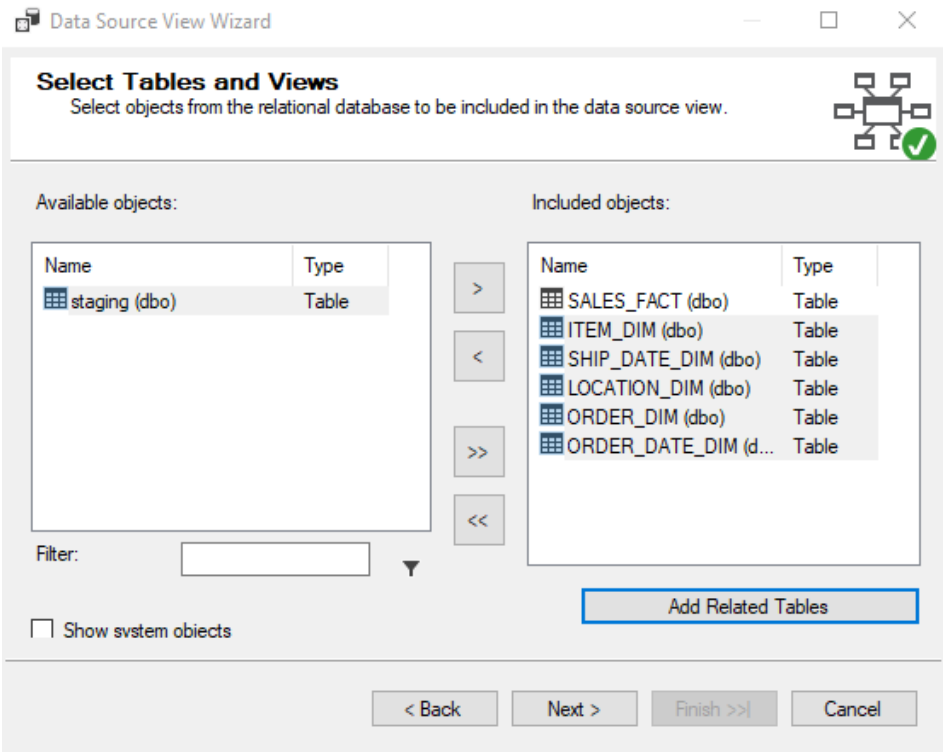


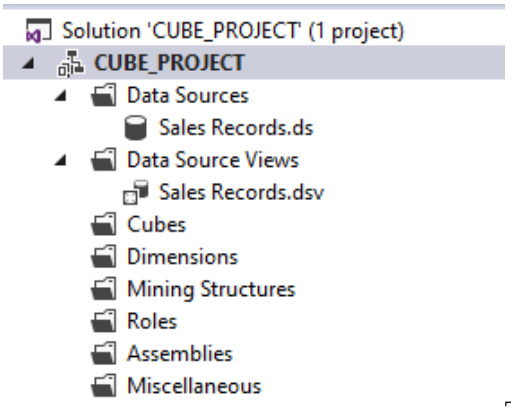


The cube project has been created.

After the creation of the above project, we had to load the dimension and the fact tables on it in order to make it work (connection with our database).

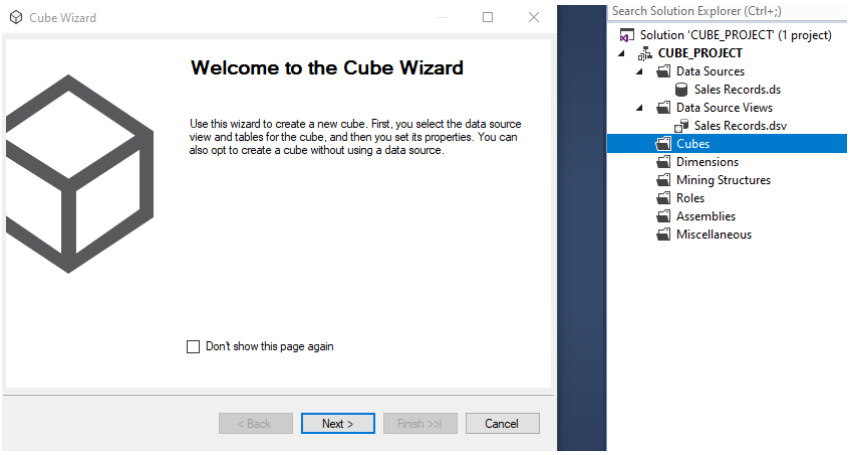


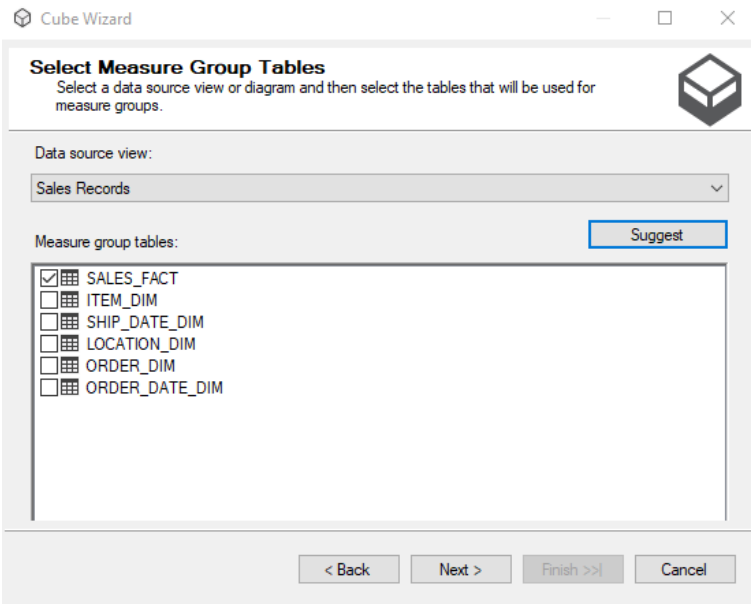




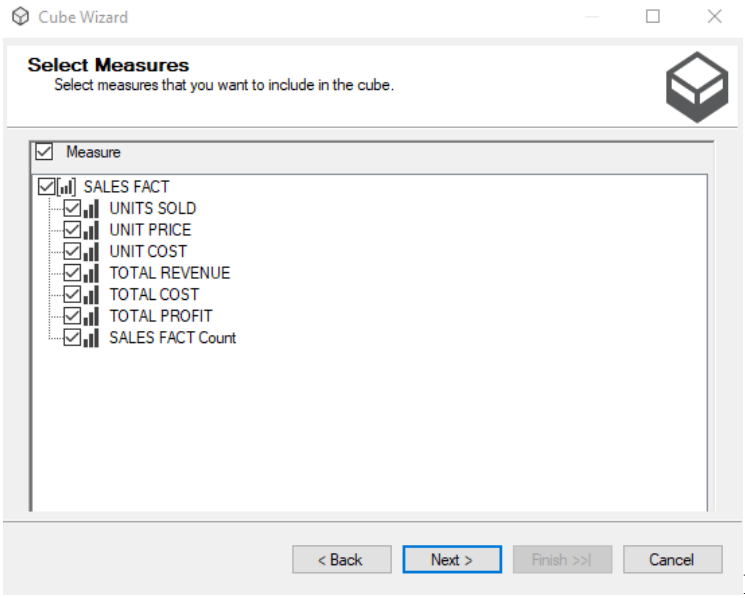
The connection with our database has been made.

Last but not least we need to specify the dimension tables and the one with the measures (fact table).

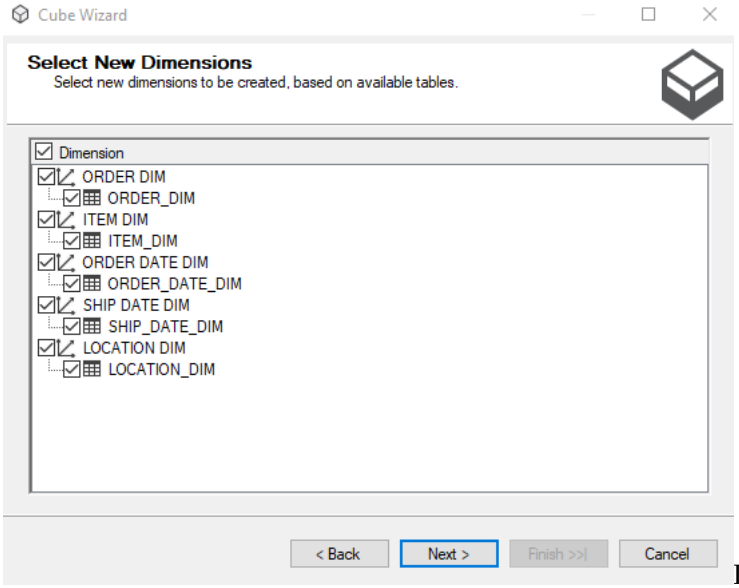




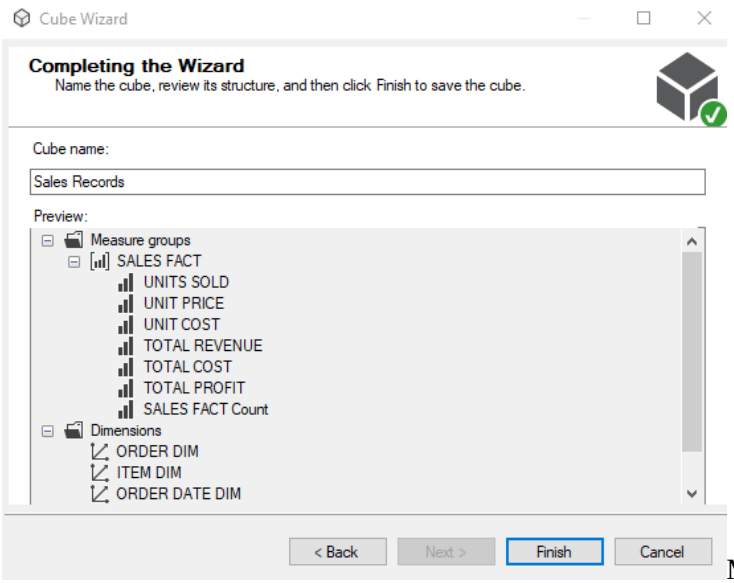
Fact table has been selected.



Measures have been selected.



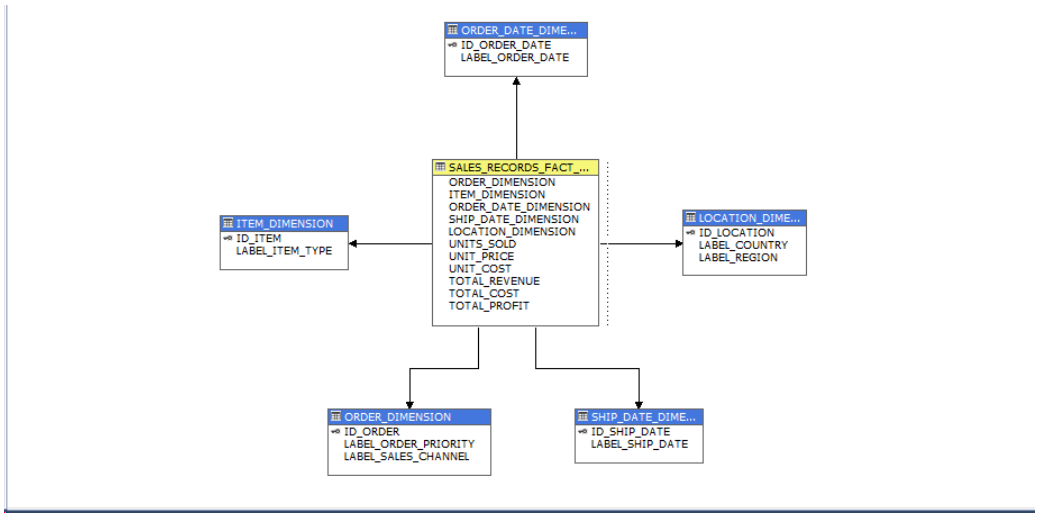
Dimensions have been selected.

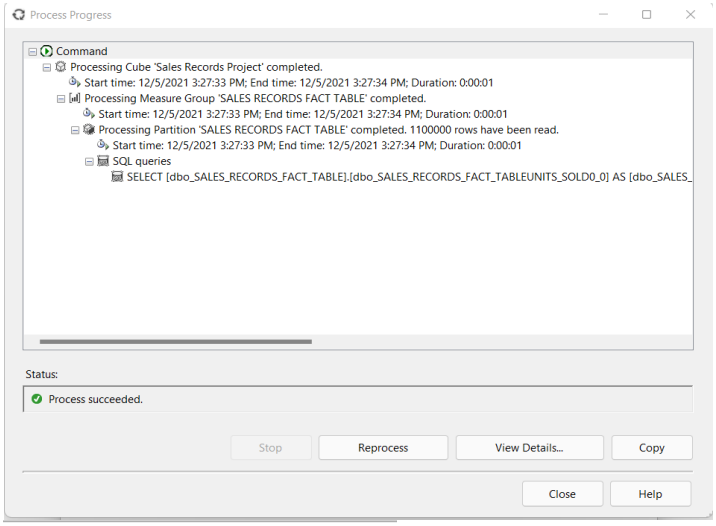


Metrics have been distinguished.

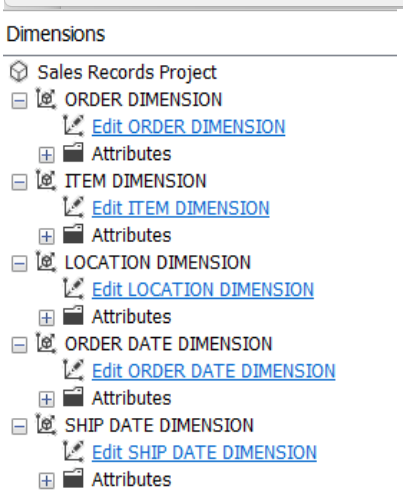
Star Schema's Overview.

The above star schema shows the connection between the fact table and the dimension tables.

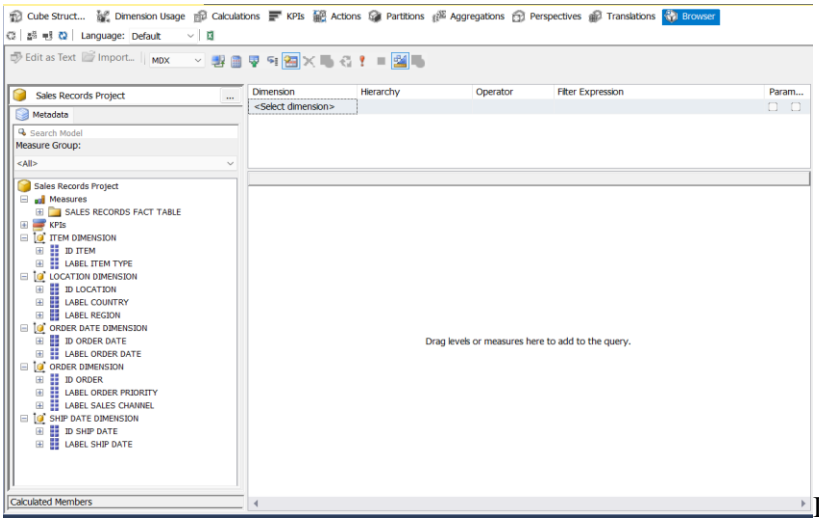




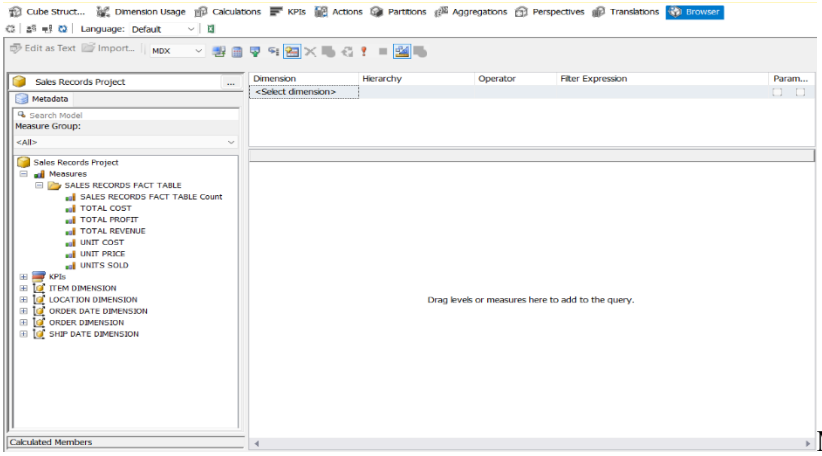
Aggregations were processed successfully.



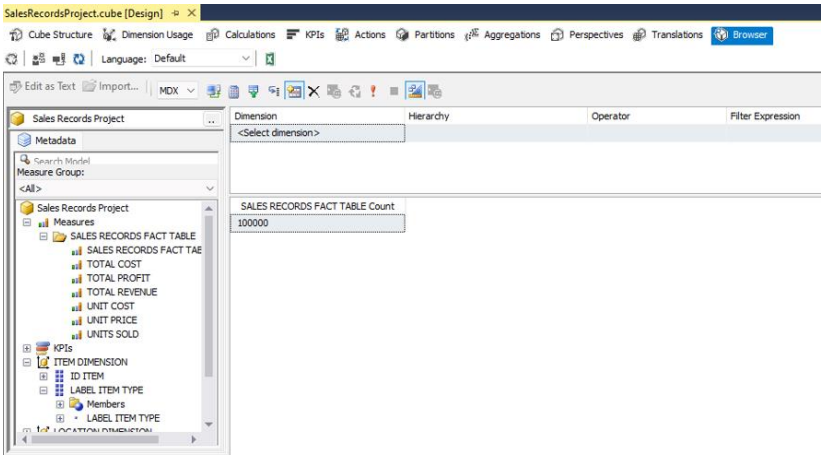
Tip: in this section we've edited the dimension in order for us to obtain their labels also and not only their IDs.

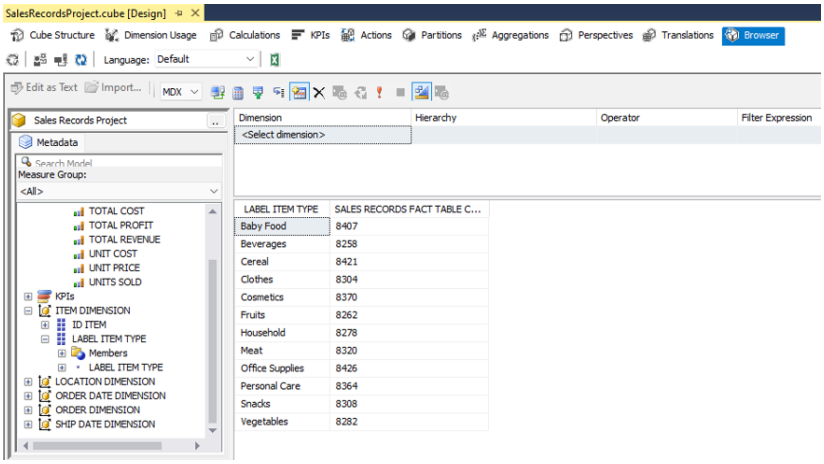


Dimensions' overview.



Measures' overview.

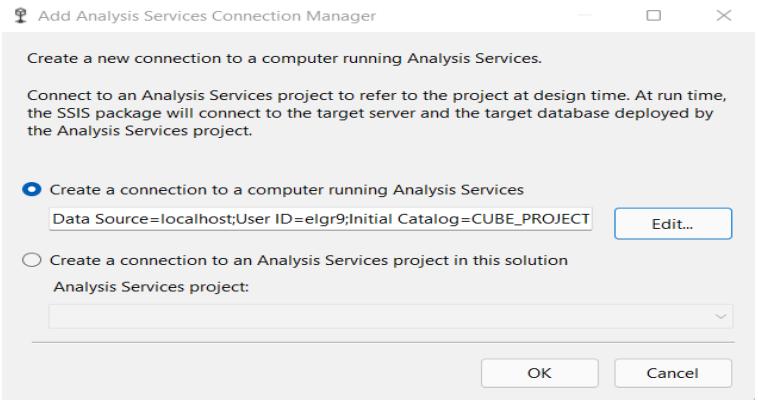


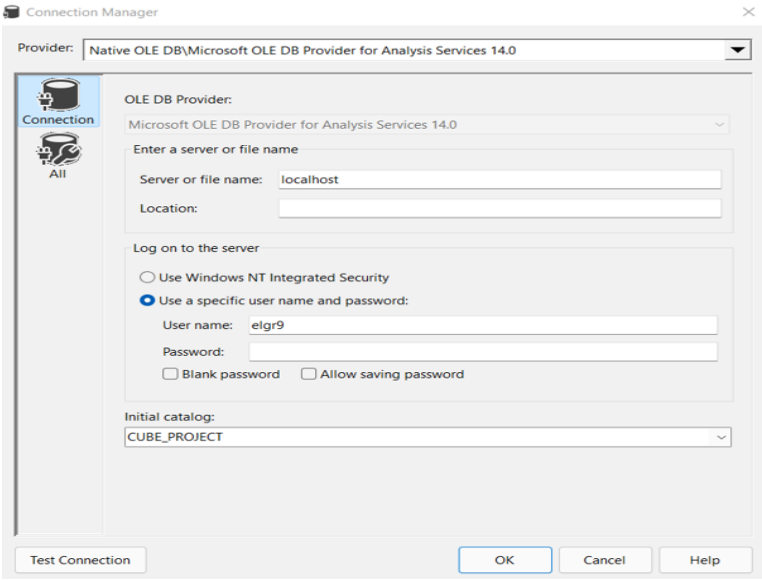


As you can see in this section (that can be seen in last two screenshots) we can work and "play" with the cube's dimensions and end up with a variety of table filtering results. This feature that can be found in the browser section of the SSIS is quite limiting. For this reason we can browse the pivot table through Microsoft excel and work with the cube's dimensions in a better and more effective way

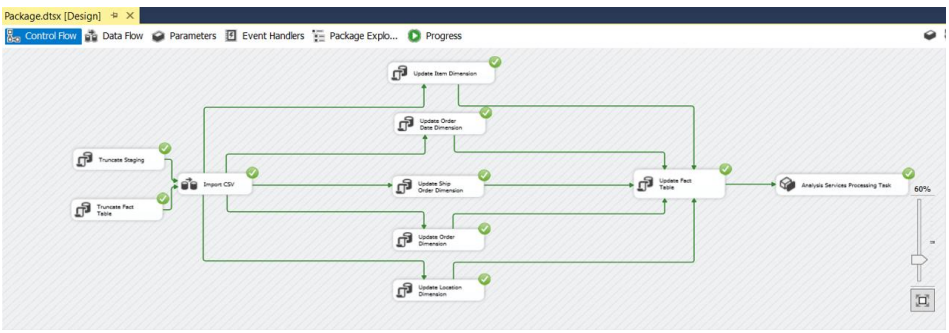
AUTOMATIZATION OF THE CUBE'S PROCESS

Before entering into the cube's possibilities analysis, we had to automate the above process by adding it into the ETL process that we've conducted successfully earlier, so we won't have to execute the above process again and again every time. This is a very small process and it's done by connecting the ETL and the cube projects together.





Final Control Flow Overview

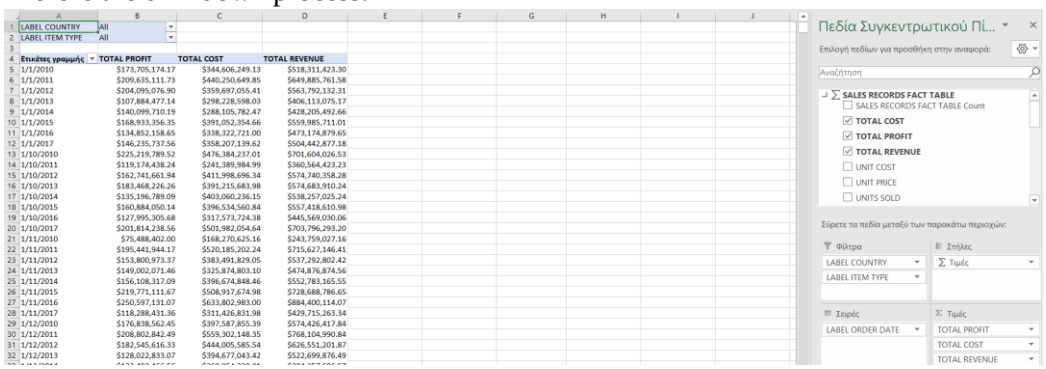


BROWSING THE PIVOT TABLE THROUGH MICROSOFT EXCEL

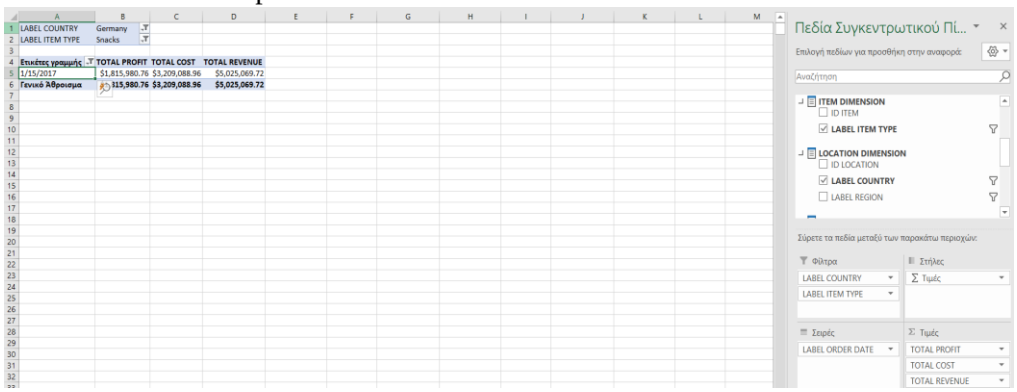
The SSDT gives us the possibility to browse the existing cube and its pivot table through Microsoft excel. Below we can see some basic functions that can be done within excel, by using the pivot table.

- Drill Down: Specialization➔ Going from general results to specified ones.

  -Before the drill down process.



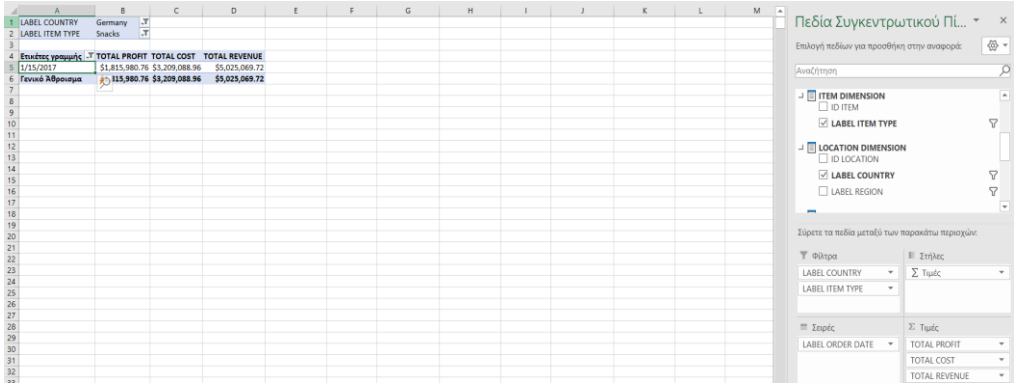
  -After the drill down process.



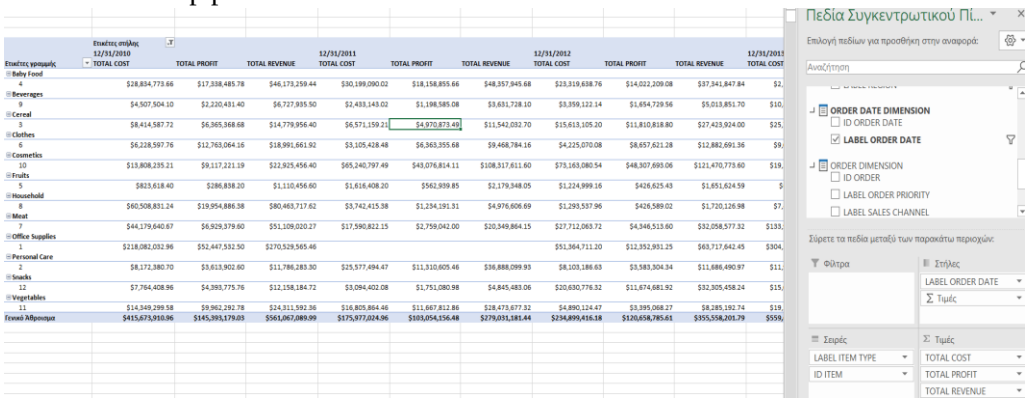
  As you can see, we went from a general table that included information (total measures and order dates) for every country and every item type, to a specified one that contains information (total measures) about the orders of snacks that have been made in Germany on 1/15/2017.

- Roll Up: Generalization➔Going from specified results to general ones.

  -Before the roll up process.



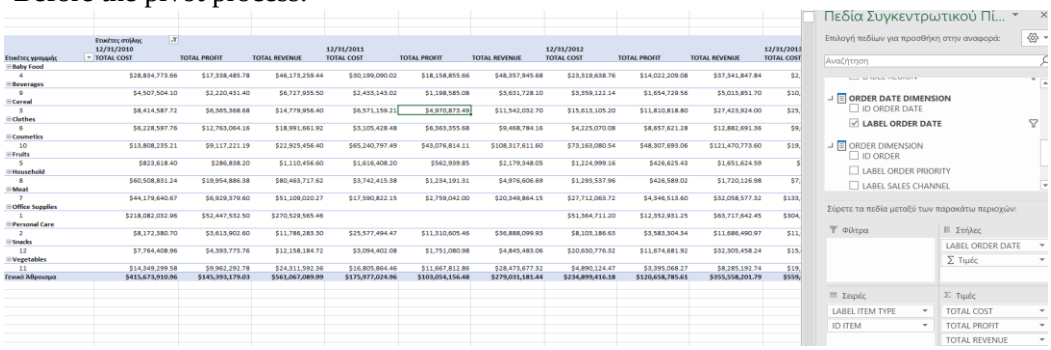
-After the roll up process.



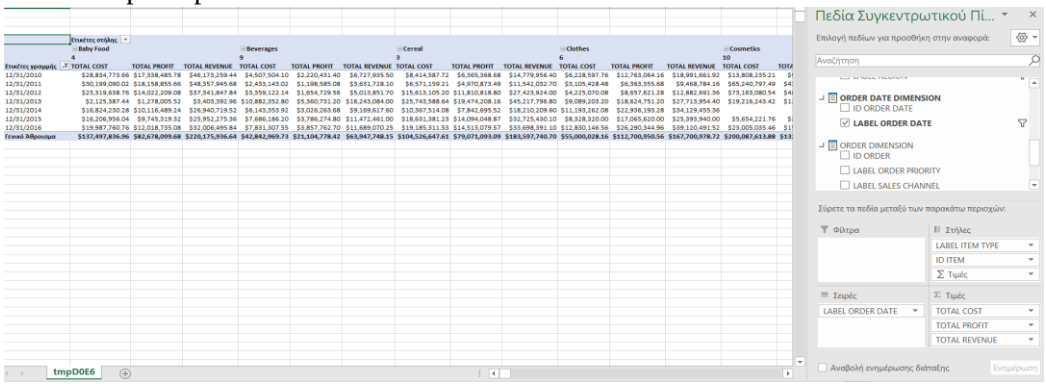
As you can see, we went from a specified table that included information (total measures) about the orders of snacks that have been made in Germany on 1/15/2017, to a general one that contains information (total measures) about every item type and every order date that can be found inside our dataset.

- Pivot: Change between rows and columns, filters and measures.

-Before the pivot process.

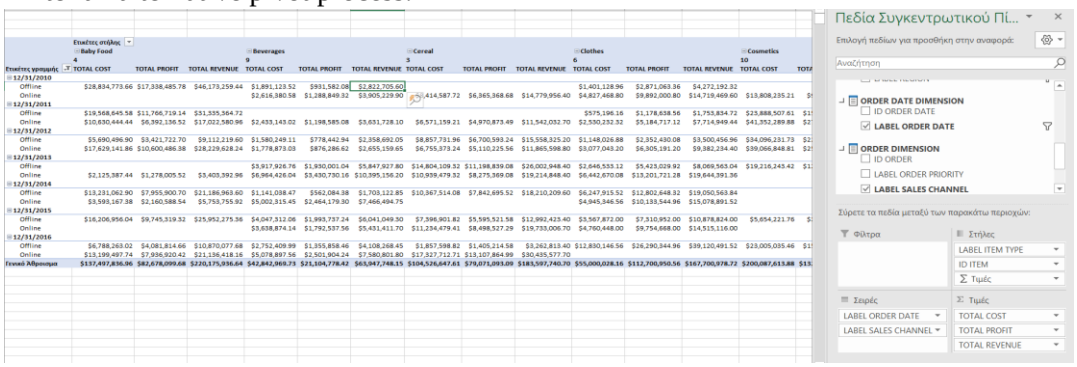


-After the pivot process.



As you can see the rows of the previous image are now the columns of our table and the columns of the previous image are now the rows of our table.

-After an alternative pivot process.



As you can see, we've added sales channel in the row sector and now the table contains information (total measures) about all the item types, every order date that can be found inside our dataset and the way that the order has been made (offline or online).

- Slicing: Filtering our data to focus on just a subset of it (slicing out only a dimension of the cube).

As you can see, we've sliced the data through the pivot table in order to end up with information about the total profit that has been made for every country on every region worldwide on every possible order date that exists in our dataset.

- Ranking: We rank data categories taking into consideration the metric's results.

-Before the ranking.

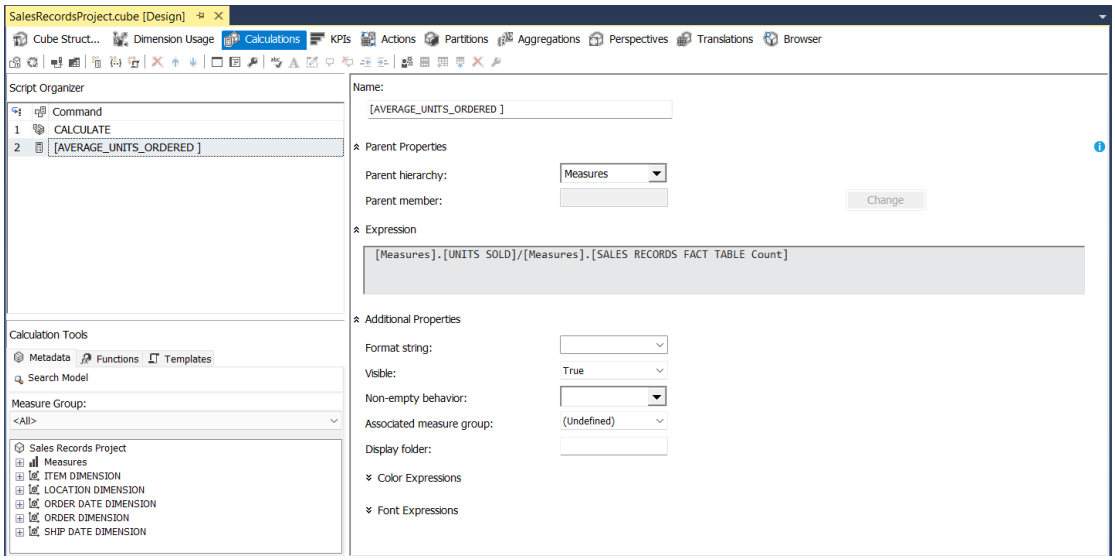
-After the ranking.

ID ITEM	11	
LABEL REGION	All	
LABEL COUNTRY	All	
TOTAL PROFIT	**Ετικέτες στήλης**	
Ετικέτες γραμμής	**Vegetables**	**Γενικό Άθροισμα**
C	2	2
H	1	1
L	4	4
M	3	3
Γενικό Άθροισμα		

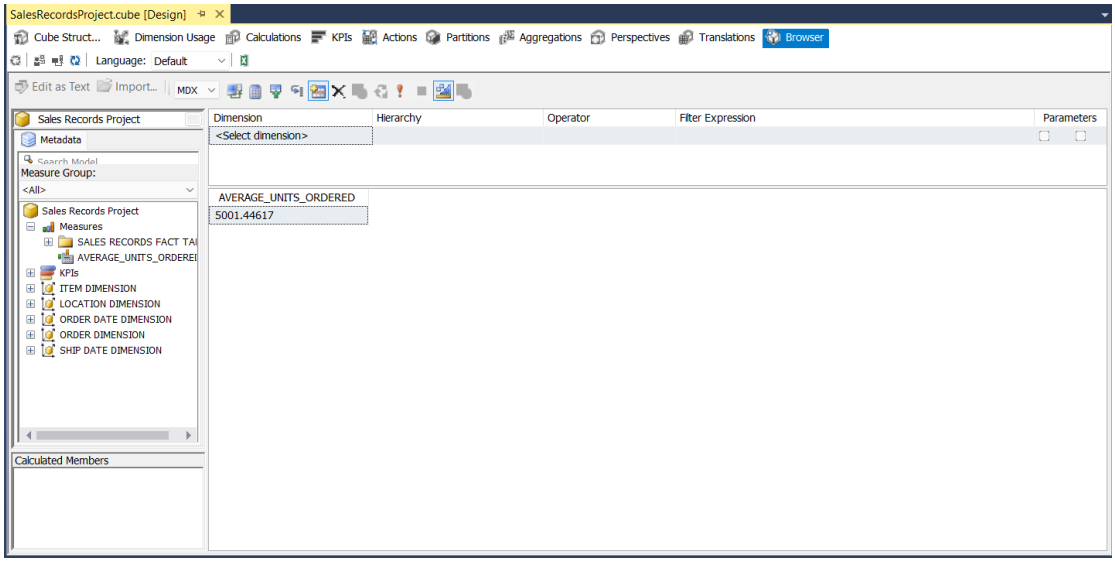
As you can see, we've ranked the order priority of the vegetables for orders that've been made all the previous years for every country that exists in our dataset, using total profit as benchmark.

CREATION OF CALCULATED MEASURE

This kind of measures are used for dynamic calculations. For this reason, we've constructed a calculated measure in order for us to know in every data insertion the average units that have been ordered during the 7 years period.

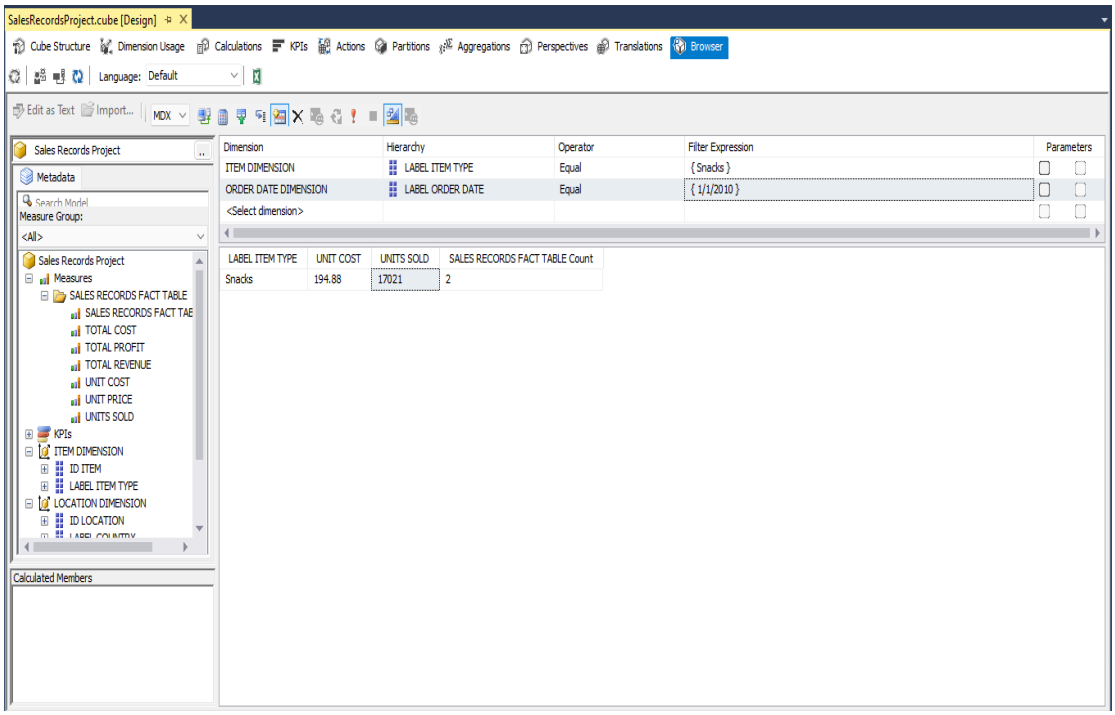


After successfully processing the cube, we observe that the new calculated measure has been added into the measures' table and the average units that have been ordered is 5001.

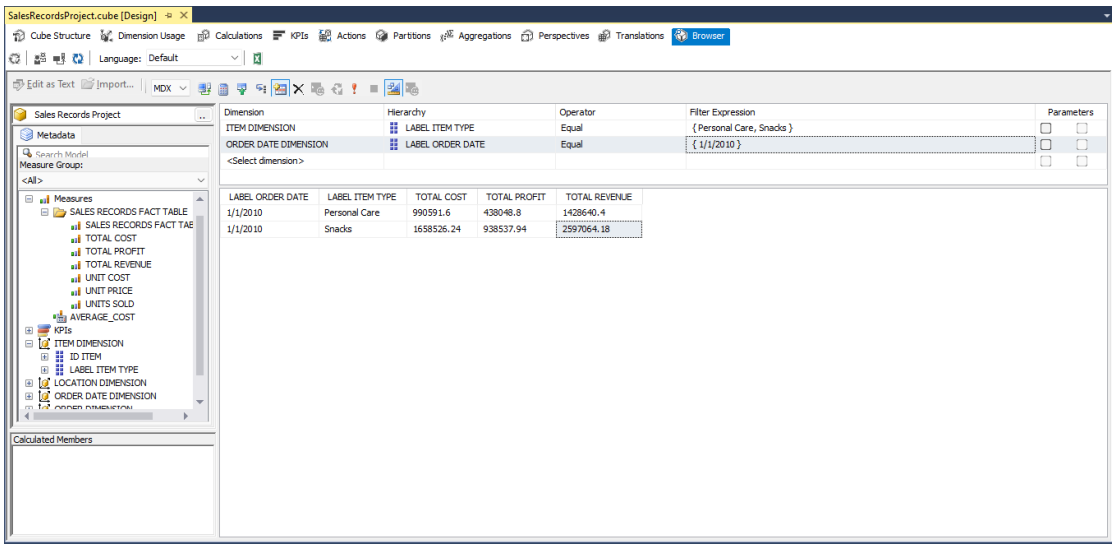


OLAP QUERIES

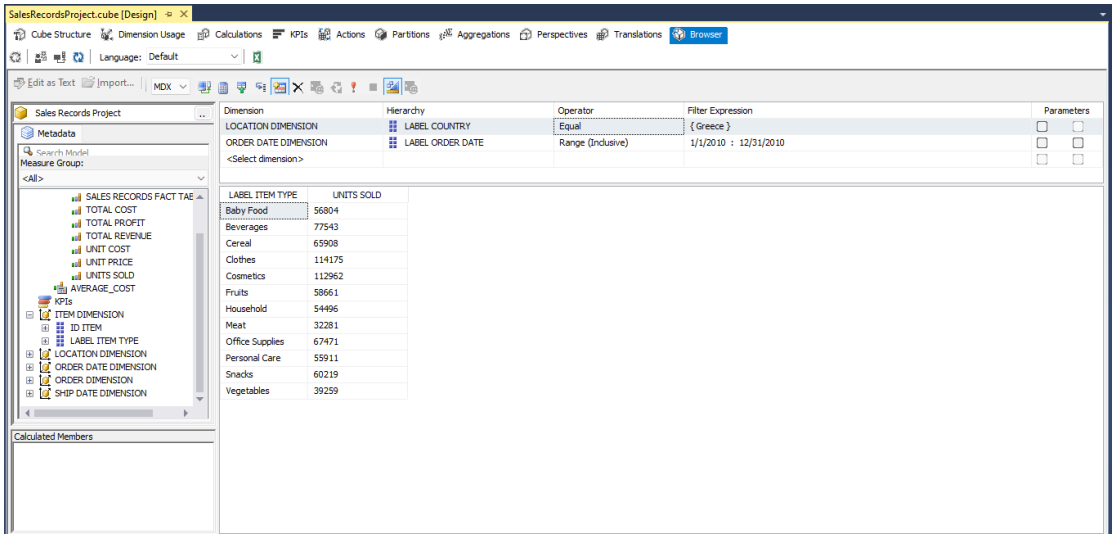
1. In the following report we can see a result of an Olap Query which shows how many times (2) and how many “SNACKS” are sold in 1/1/2010.



2. Using ROLL-UP Method we compute the Total Cost, Total Profit, Total Revenue for all the Personal Care Items and Snacks in 1/1/2010.



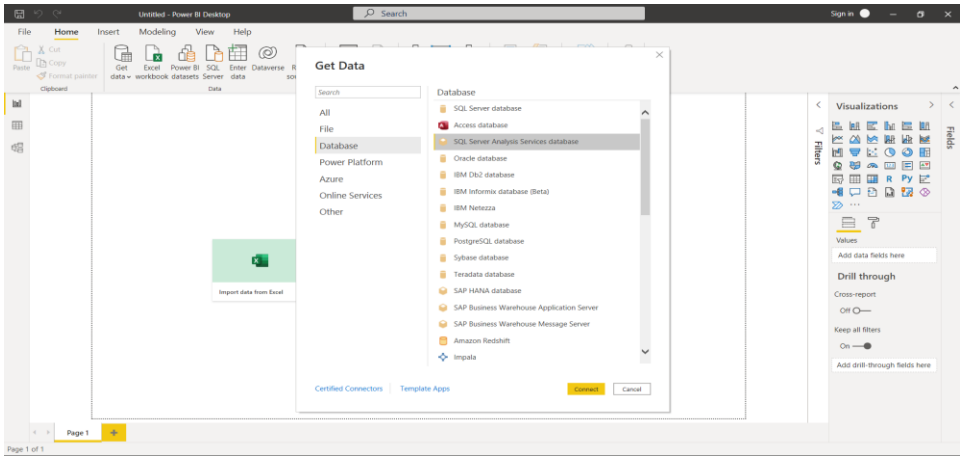
3. And finally, a Slice and Dice example which shows how many units were sold by Item Type in Greece between the 1/1/2010 – 12/31/2010.



POWER BI

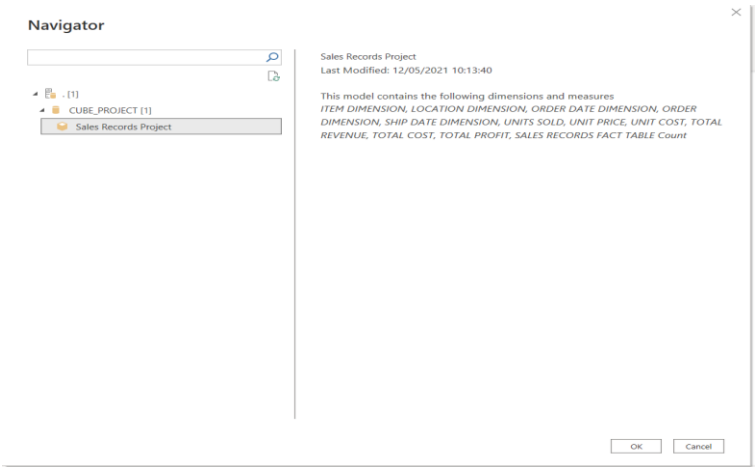
CONNECTION BETWEEN POWER BI AND SSAS

First of all, we had to create a connection between POWER BI and SSAS in order for us to load the multidimensional model.

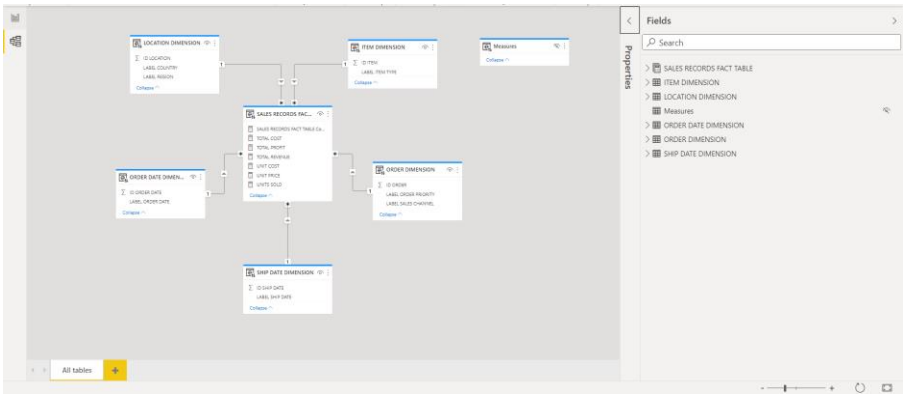


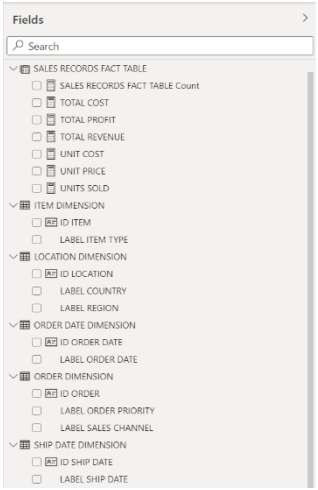
After selecting the method with which we are going to load the data, we had to choose our working server.

As you can see, POWER BI finds the specific project and presents its dimensions and measures.



In the field below you can see the preview of our model's star schema.



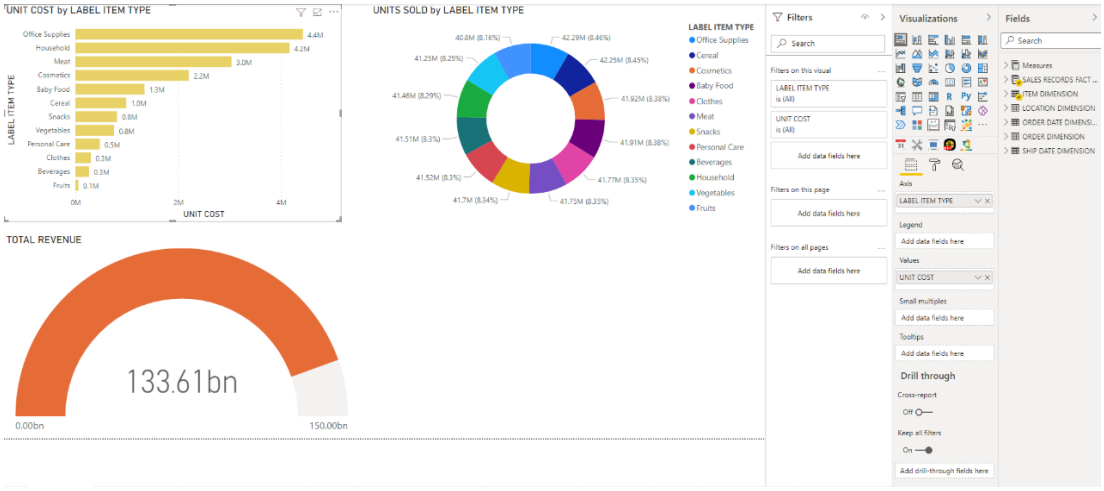


Analytically, here you can see the fact table with its metrics and the dimension tables along with their labels.

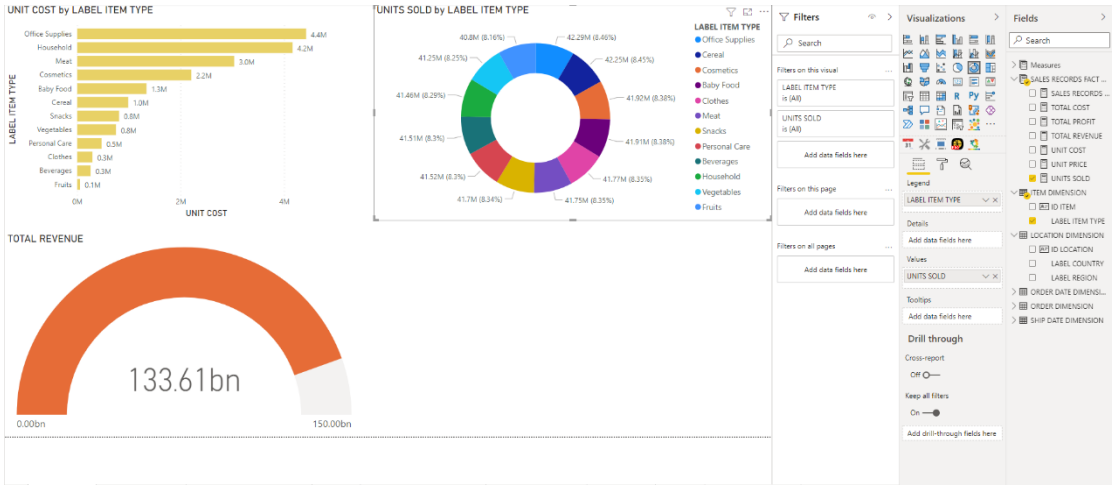
In this section we will represent how and with which fields we managed to create the diagrams for our Business Case. All the diagrams will be explained in detail in the next section.

This page shows **the Total Revenue** of every **Item Label Type** between the **Unit Cost** and the **Units Sold** Globally.

For the **Unit Cost by Label Item Type** diagram, we used from the **Item Dimension** the **"Label Item Type"** and for the metric we've used the **Unit Cost**.

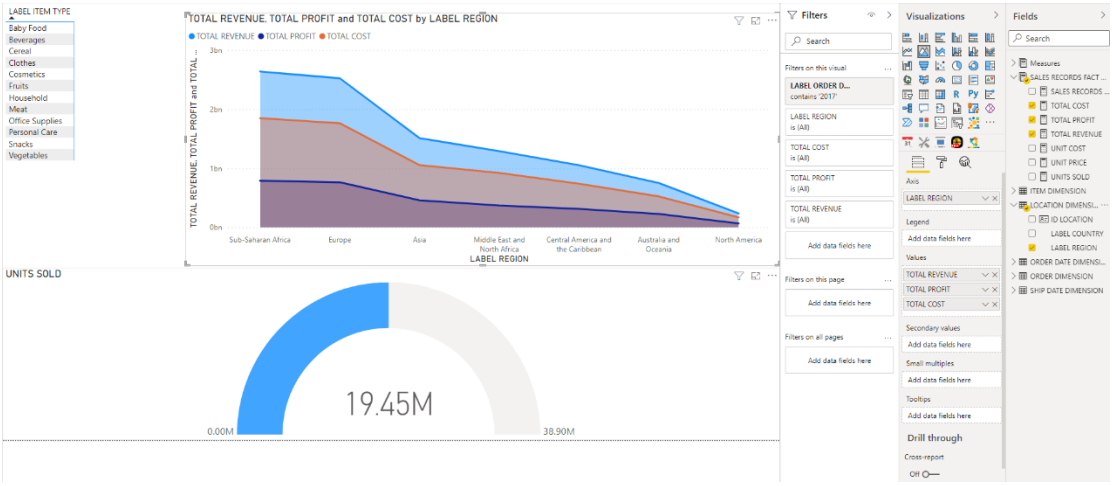


For the **Units Sold by Item Type** diagram we used from the **Item Dimension** the **"Label Item Type"** and for metric we've used the **Units Sold**.



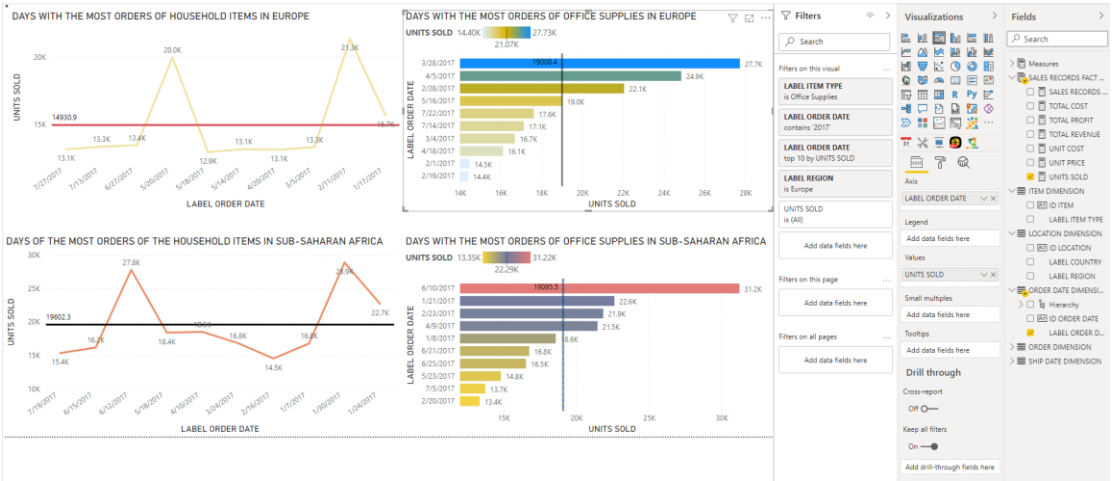
This page shows **Total Revenue, Total Profit, Total Cost by Region in 2017** of every **Item Label Type**.

To create this diagram, we used the three metrics **Total Cost, Total Profit, Total Revenue** and from the **Location Dimension** the **"Label Region".** Also, we have filtered the Order Date to isolate them in 2017 only.

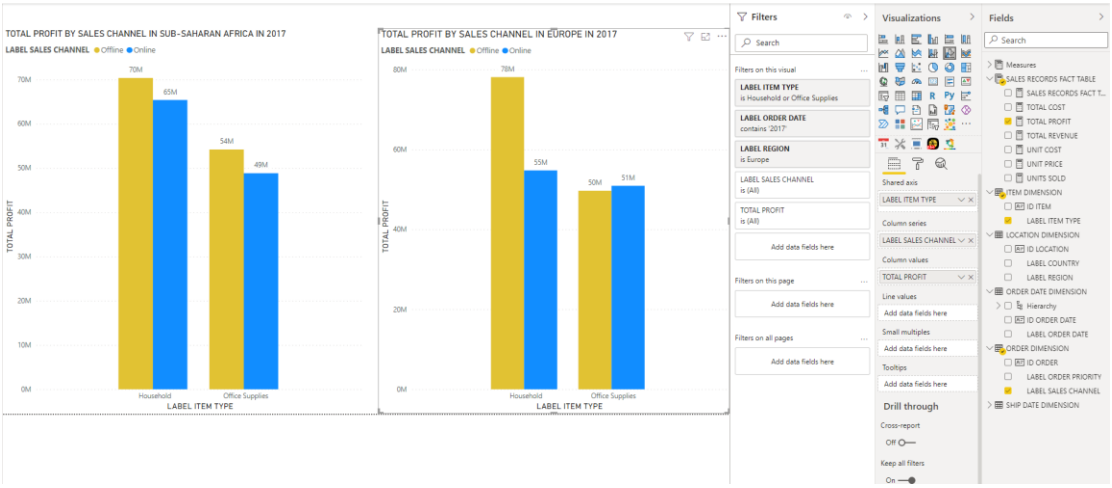


In this page we used 2 different diagrams to represent the top 10 days with the most orders of the two best sellers 'products in Europe and Sub-Saharan Africa.

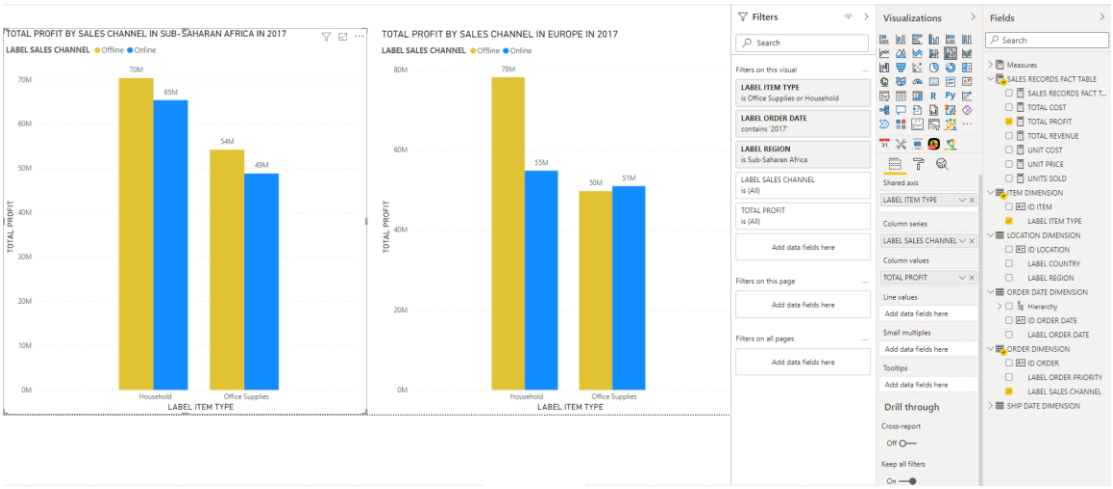
To achieve it we filtered our data to present the Office Supplies and the Household items only in 2017 only in Europe and Sub-Saharan Africa. Also, for these 4 diagrams in this page we used the same metrics for all of them. Specifically, we've used the **Units Sold** as metric and the **Label Order Date** from **Order Date Dimension** filtered to show only how many orders have been made in 2017.



In this page we wanted to show how the market preference changes depending on region. So, we filtered the data in order the diagrams to show how much profit has been gained by Sales Channel. We used the **Total Profit** metric, the **Label Item Type** from **Item Dimension,** and the **Label Sales Channel** from **Order Dimension.**

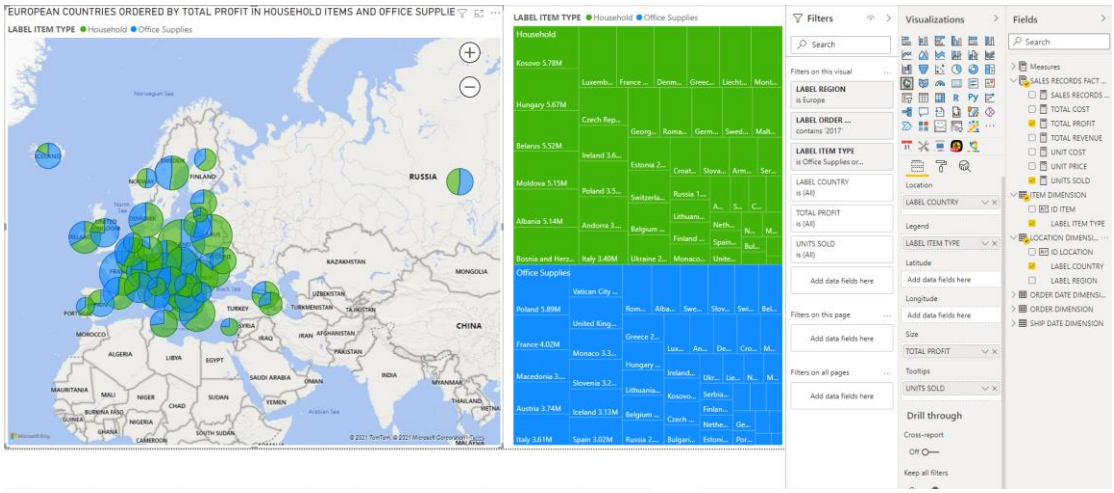


Of course, the diagrams are filtered by Region because we want to represent data only for Sub Saharan Africa and Europe.

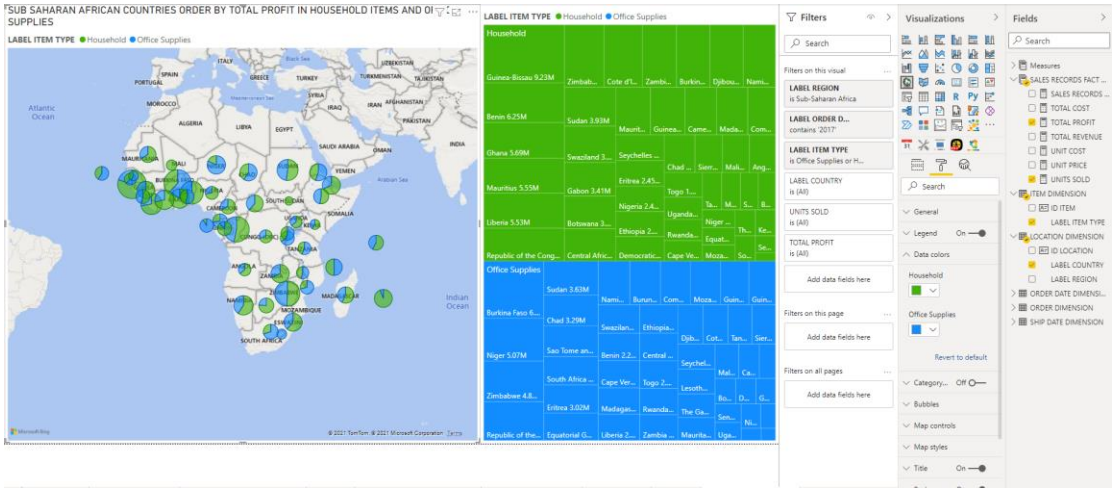


Finally, we wanted to find the countries with the most Total Profit in Europe and Sub – Saharan Africa. So, we created 2 map diagrams in order to zoom into the best Seller Country.

For the Europe Map we filtered our data to have as region only the Europe for 2017 and only for the Office Supplies and Household Items. For the metrics we used Total Profit and as Location we used from the **Location Dimension** the **Label Country.**

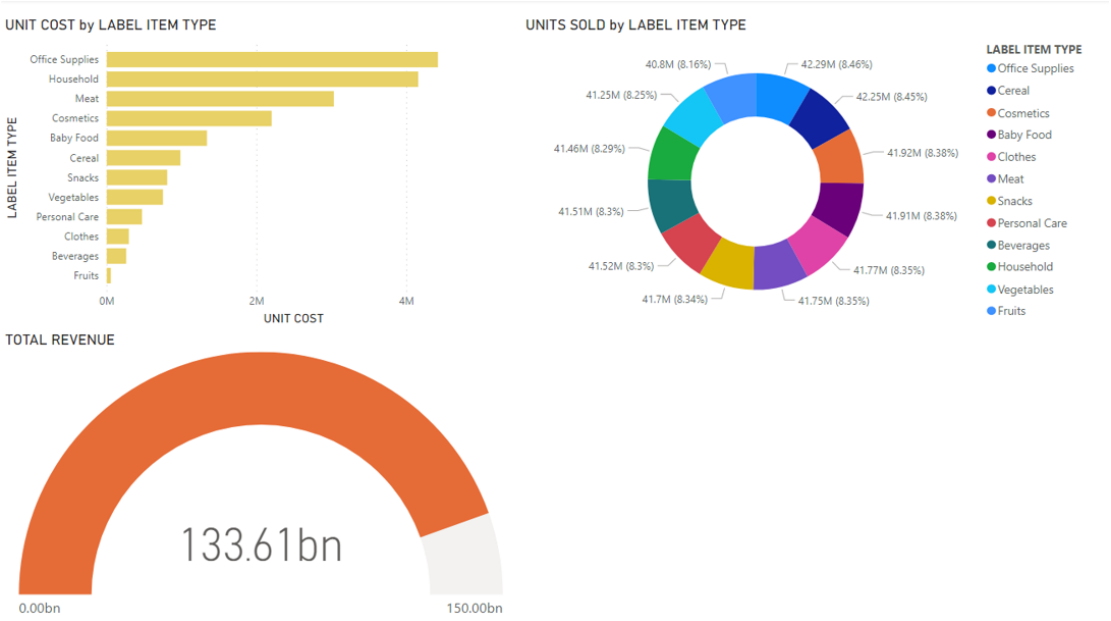


We did the same process as above for the Sub-Saharan Africa and we ended up with the following Map Diagram. The only difference between the 2 map diagrams is that the second diagram is filtered to show only the household and the office supplies that have been sold only in Sub-Saharan Africa.

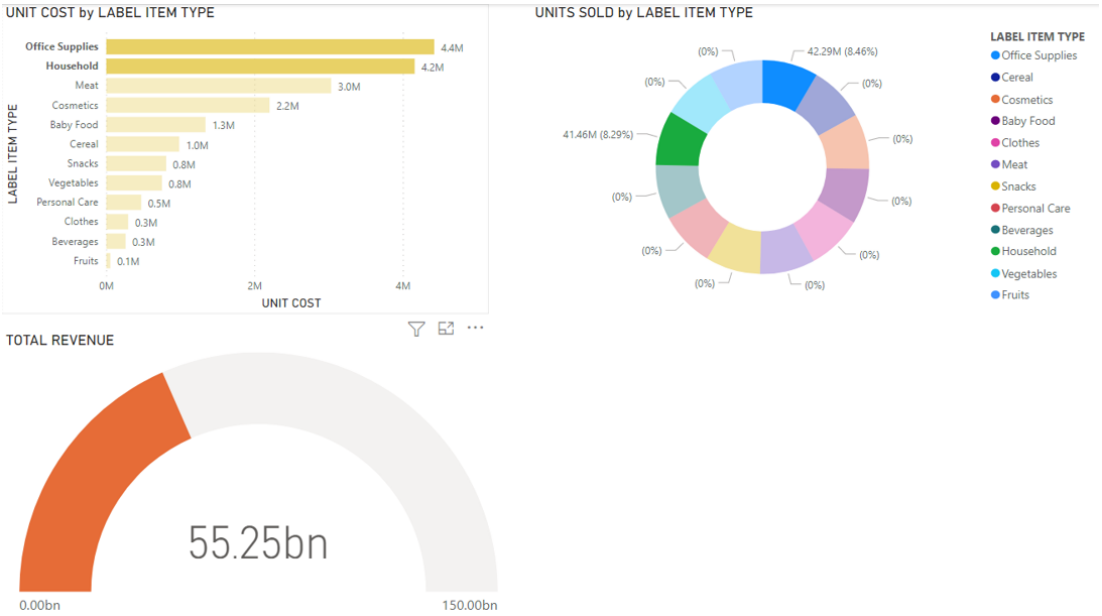


Below we are going to present you the diagrams that have been created through POWER BI and focus on our business case.

First of all, we wanted to find out what's happening in our dataset, considering some metrics for every item's type. Below you can see a stacked bar chart that represents the unit cost of every item's type globally, a donut chart that shows the units that have been sold from 2010 to 2017 for each item category globally and a gauge that shows the total revenue that has been made from the total sales of every item type globally from 2010 to 2017.



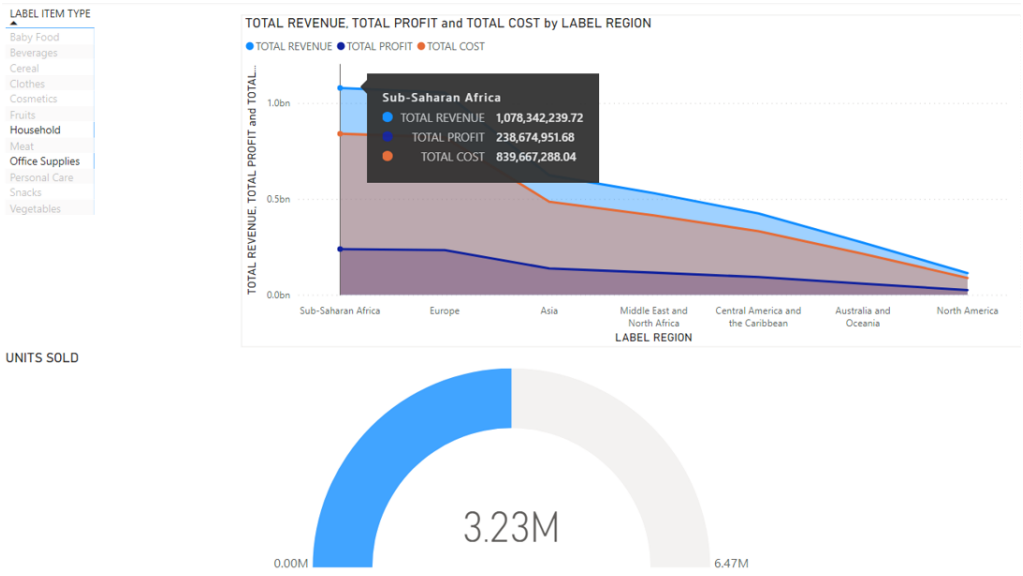
In order for us to provide business advice to the enterprise that trusted us, we had to analyze in depth the cost of office supplies and household items in comparison with other kinds of products, how many of them were sold the previous years (2010-2017) and what is the total revenue that has been made from their sales. As you can see from the diagrams below, the two biggest markets (in terms of unit costs and revenues) worldwide are the office supplies and the household items, so their decision to invest in these two business areas, seems right.



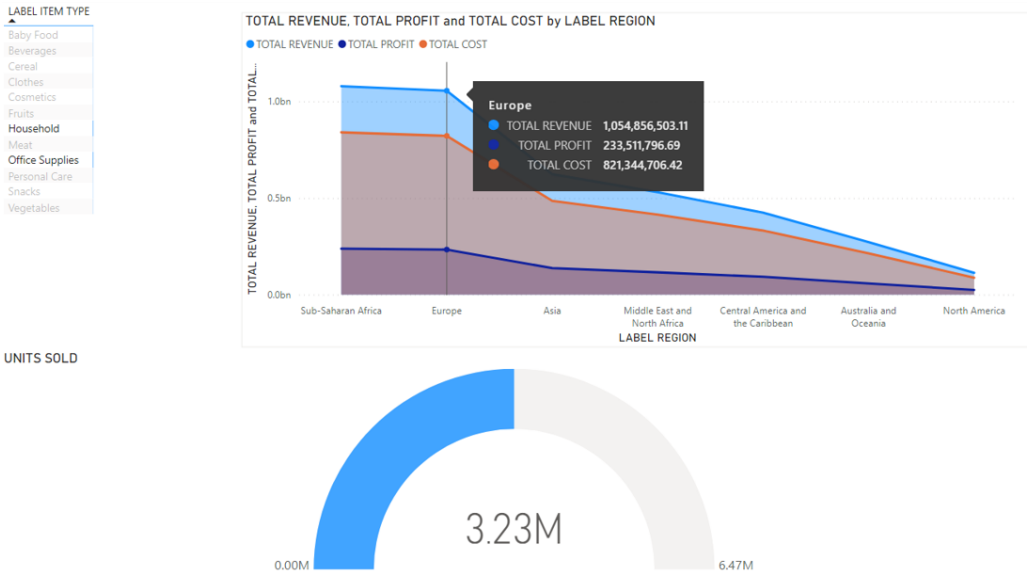
Analytically we can see that 42.29 and 41.46 million units of office supplies and household items have been sold with total unit cost of 4.4 and 4.2 million dollars (respectively). Moreover, the total revenue that has been made from these two markets is 55.25 billion dollars. So, despite their large unit costs, these two product markets are the ones that end up with the biggest revenues comparatively to all others.

Since we've found out that the two biggest product markets are the ones of office supplies and household items, we wanted to move into a deeper analysis for them. As you can see from the graphs below, these two markets are booming right now (these two graphs contain information from 2017 only) in Sub-Saharan Africa and Europe.

In Sub-Saharan Africa the two markets combined have a total profit of 238.674 million dollars in 2017.



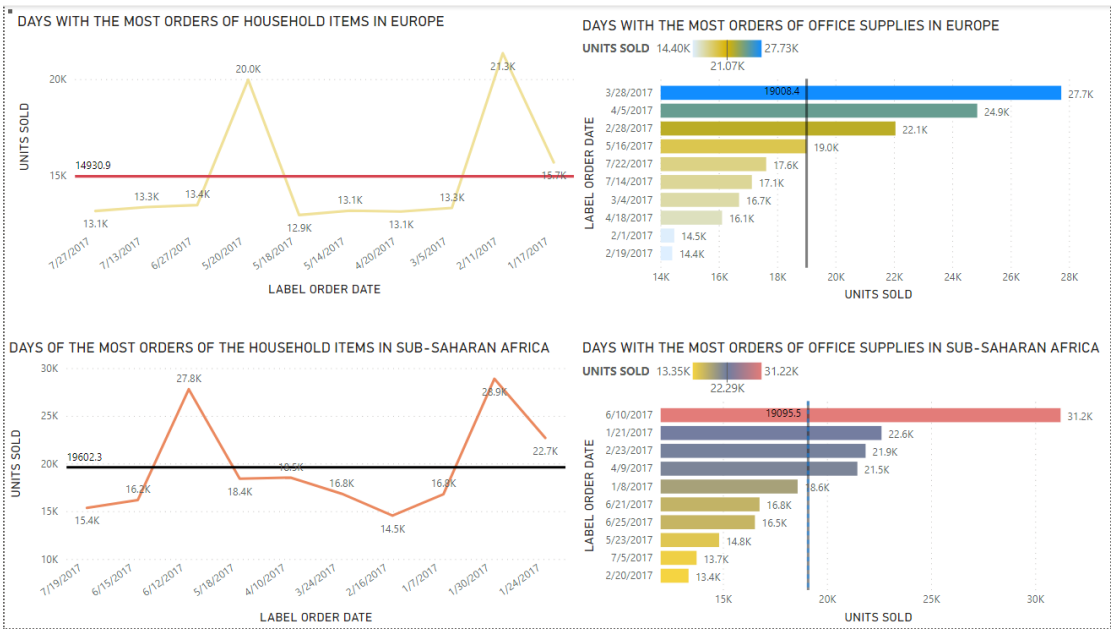
In Europe the two markets combined have a total profit of 233.511 million dollars in 2017.



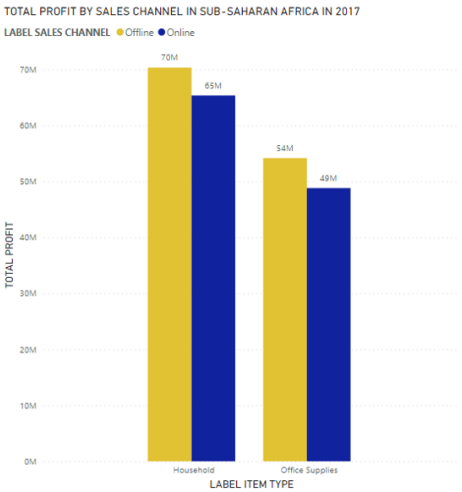
These two regions seem to be the best sellers as it seems to have sold 3.23 million units of these two product categories and they have the biggest margin profit space comparatively with every other region worldwide.

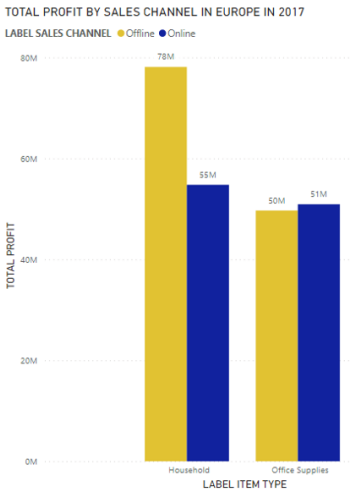
Considering the dates of 2017 in which we observed the biggest orders in terms of units, we found out that in Europe the household items had their peak at 2/11/2017 with 21.300 units ordered and at 20/5/2017 with 20.000 units ordered, while the average units ordered where between 14.930-15.700. Moreover, in Europe the office supplies had their peak at 28/3/2017 with 27.700 units and at 5/4/2017 with 24.900 units, while the average units where 19.008.

In the Sub-Saharan Africa, the household items had their peak at 30/1/2017 with 28.900 units ordered and at 6/12/2017 with 27.800 units ordered, while the average units ordered where between 19.602-22.700. Moreover, in Sub-Saharan Africa the office supplies had their peak at 10/6/2017 with 31.200 units and at 21/1/2017 with 22.600 units, while the average units where 19.095.



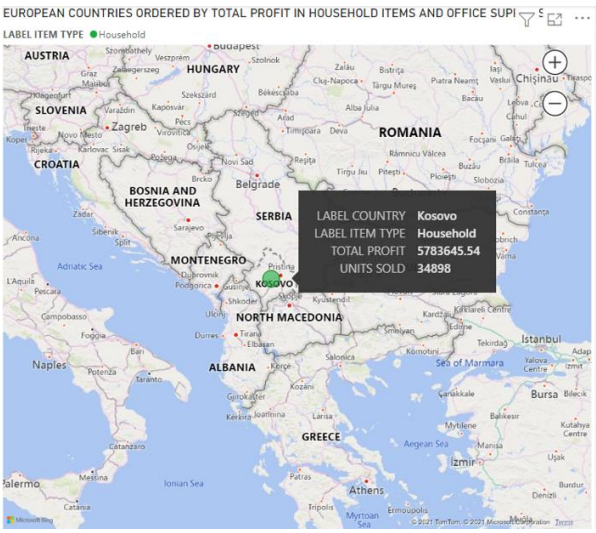
Moving on, we had to evaluate if the investment should be done on offline or online sales channels. For this reason, we compared the two channels for office supplies and household items in Europe and in the Sub-Saharan Africa region. The results showed that considering these two product markets, there were differences between these two parts of the world.

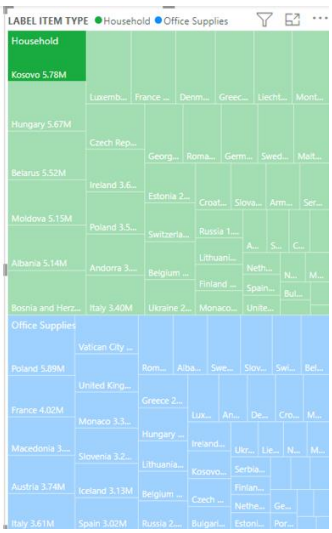


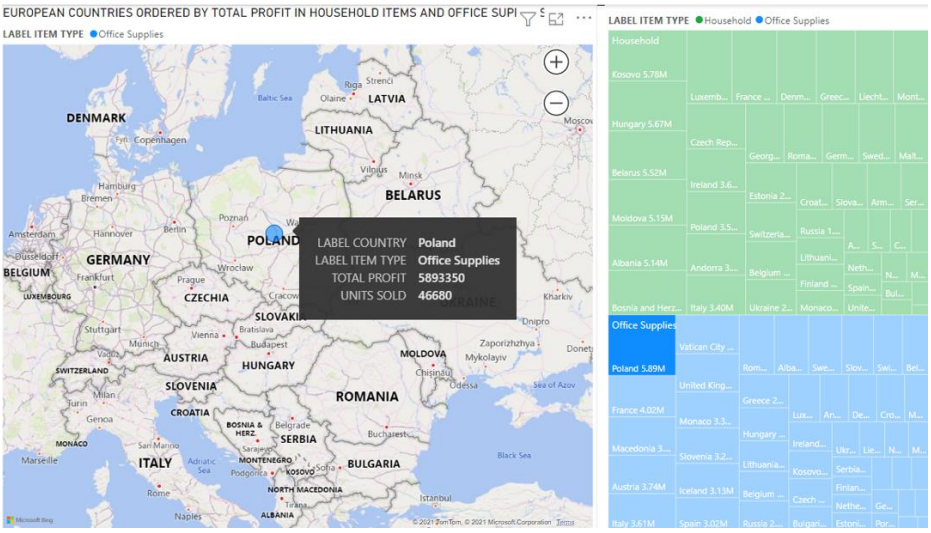


More specifically, in 2017, the Sub-Saharan Africa's customers preferred to buy household items through and office supplies through physical stores (offline channels), while in Europe, the customers preferred the offline option for their household shopping and the online for their office supplies. If you think that is very logical, because products like furniture and household items in general, must be tested before the customers decide if they want to buy them or not.

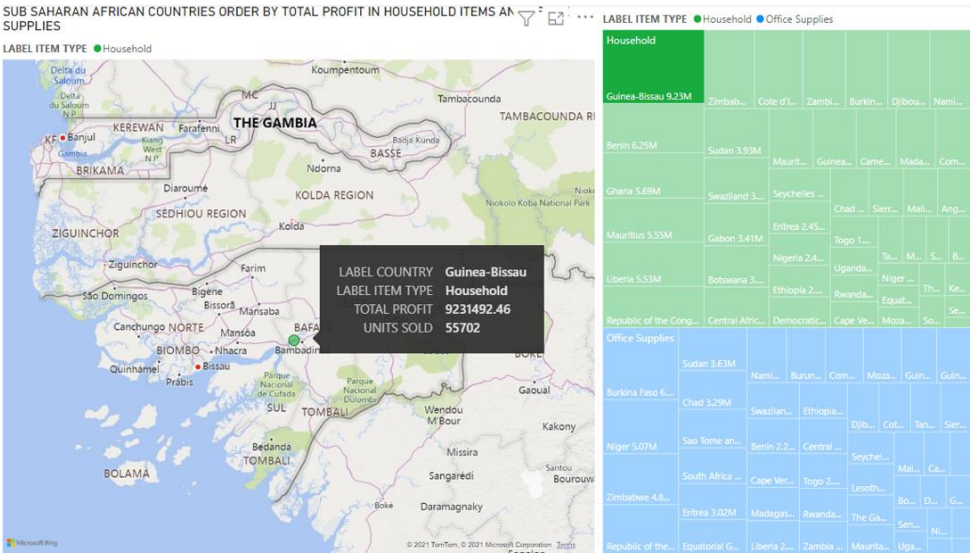
Last but not least, we wanted to decide which country was the best option for our customer to invest in household items and office supplies in Europe and Sub-Saharan Africa respectively. For this reason, we made a research on which was the best country in each continent (in terms of total profit) for this specific kind of investment by comparing the total profits that have been produced from household items and office supplies sales in 2017.

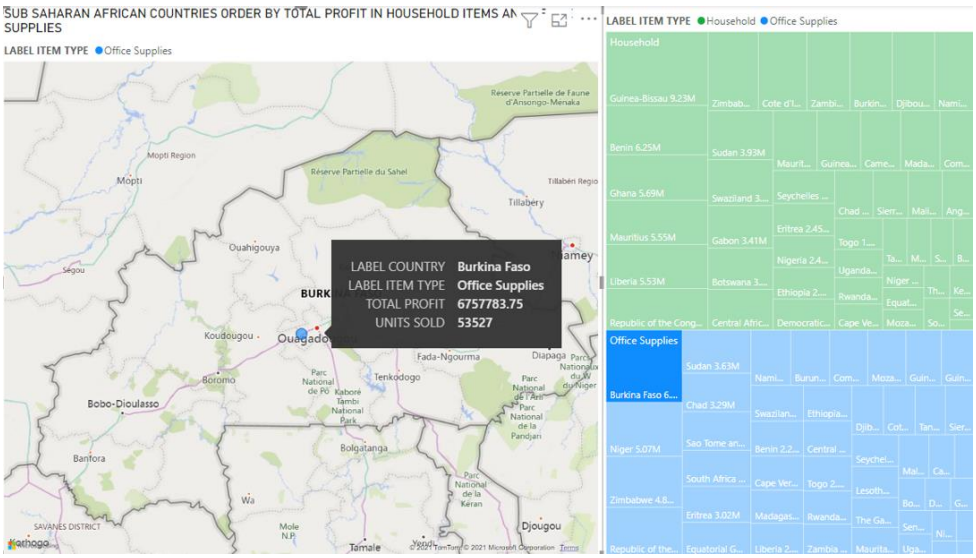






The results showed that in Europe the best country to invest in household items was Kosovo (total profit of 5.783.645 dollars) and in office supplies was Poland (total profit of 5.893.350 dollars).





Respectively, in the Sub-Saharan Africa region the best country to invest in household items was Guinea-Bissau (total profit of 9.231.492 dollars) and in office supplies was Burkina Faso (total profit of 6.757.783 dollars).

CONCLUSION

In conclusion, using the possibilities that we've gained from the creation of our multidimensional model gave us the opportunity to provide analytical advice through POWER BI to our customer. This analysis have shown that this multinational company that wants to invest into household items and office supplies, should focus on two specific regions (Europe and Sub-Saharan Africa) and in four countries (Kosovo/Poland- Burkina Faso/ Guinea-Bissau). Moreover, this analysis have shown that depending the date of the year, our customer should be ready to face different order amounts. Last but not least, their investment should focus on offline stores if we are talking about the Sub-Saharan Africa (both product markets) or the household items' market of Europe and on online stores for the office supplies' market of Europe.