Laboratory of Data Science

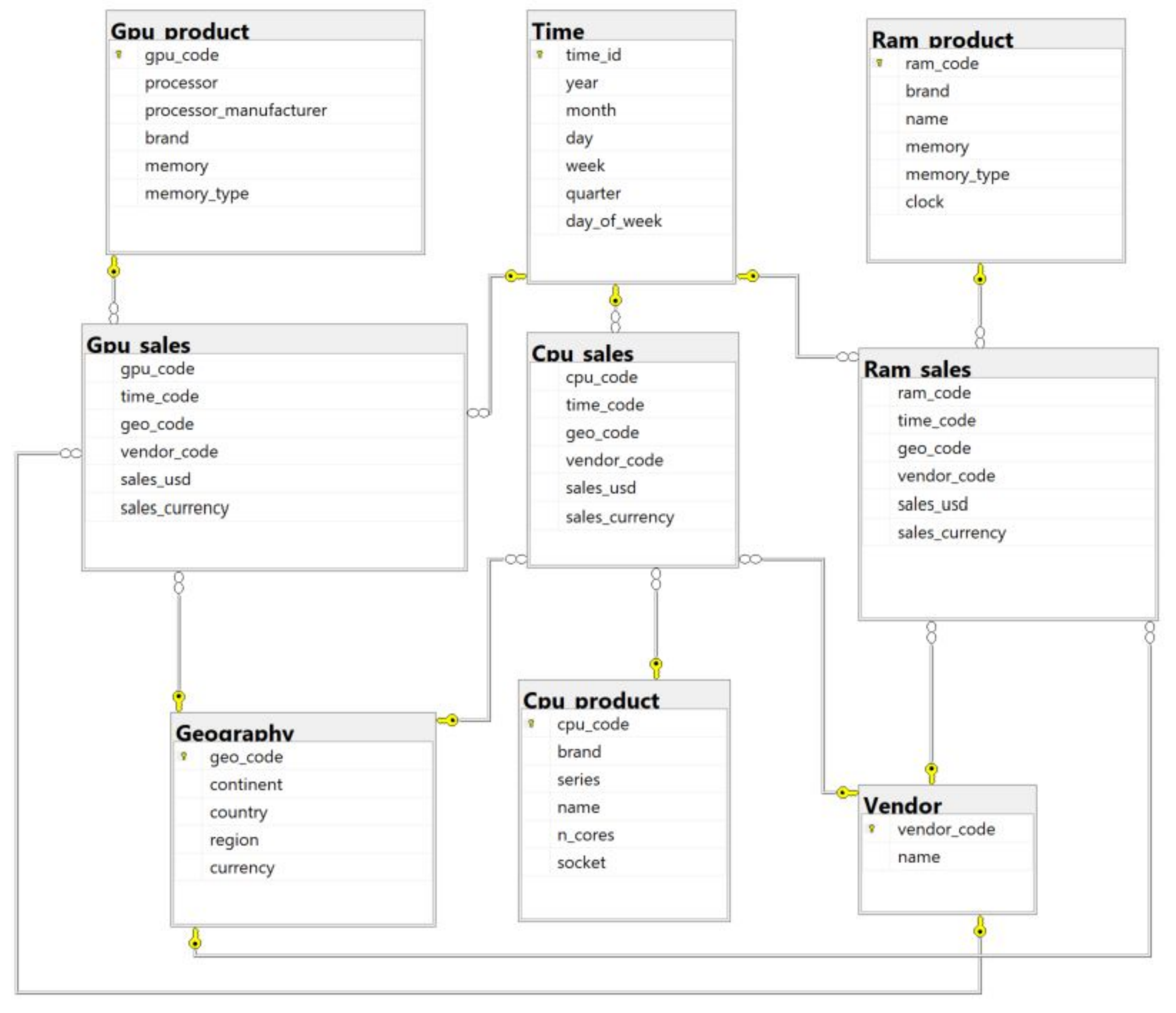
Naeem Muhammad Umair
Lucarini Riccardo

Project assignments

1. Create and populate a Datawarehouse.
2. Business question with SSIS.
3. Cube creation, MDX query and Power BI report.

1.0

The Data Warehouse



Gpu_product
gpu_code
processor
processor_manufacturer
brand
memory
memory_type

Time
time_id
year
month
day
week
quarter
day_of_week

Ram_product
ram_code
brand
name
memory
memory_type
clock

Gpu_sales
gpu_code
time_code
geo_code
vendor_code
sales_usd
sales_currency

Cpu_sales
cpu_code
time_code
geo_code
vendor_code
sales_usd
sales_currency

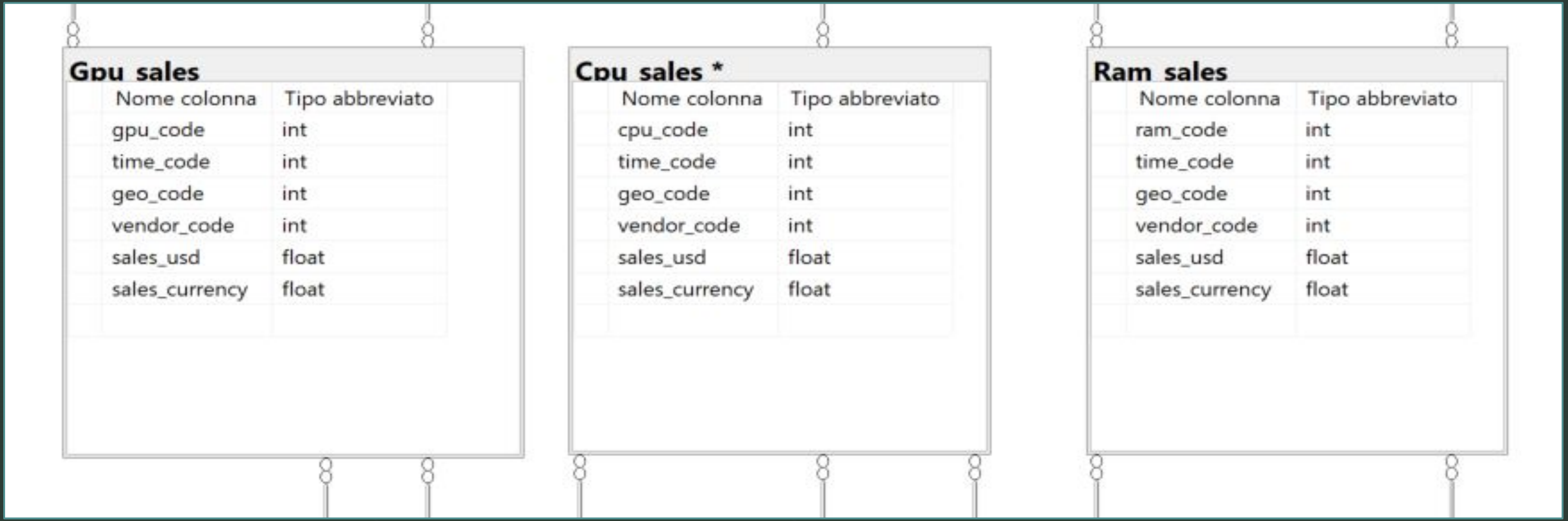
Ram_sales
ram_code
time_code
geo_code
vendor_code
sales_usd
sales_currency

Geography
geo_code
continent
country
region
currency

Cpu_product
cpu_code
brand
series
name
n_cores
socket

Vendor
vendor_code
name

The three fact tables



Gpu sales

Nome colonna	Tipo abbreviato
gpu_code	int
time_code	int
geo_code	int
vendor_code	int
sales_usd	float
sales_currency	float

Cpu sales *

Nome colonna	Tipo abbreviato
cpu_code	int
time_code	int
geo_code	int
vendor_code	int
sales_usd	float
sales_currency	float

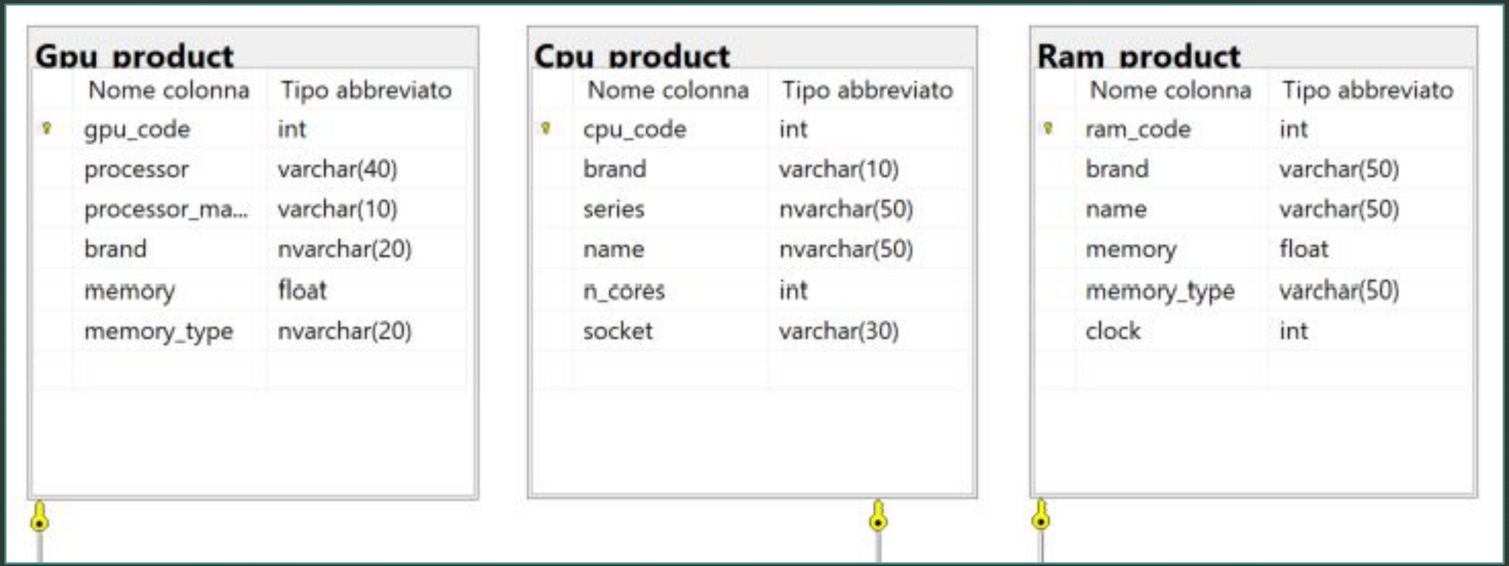
Ram sales

Nome colonna	Tipo abbreviato
ram_code	int
time_code	int
geo_code	int
vendor_code	int
sales_usd	float
sales_currency	float

- Cpu_sales, Gpu_sales and Ram_sales are the fact table of our DW.
- The DW is built as a constellation schema.
- Each fact tables is in many-to-one relationship with the dimension tables.
- Sales_usd and Sales_currency are the measures of the fact tables.

1.0

Product dimension tables



Gpu_product

	Nome colonna	Tipo abbreviato
🔑	gpu_code	int
	processor	varchar(40)
	processor_ma...	varchar(10)
	brand	nvarchar(20)
	memory	float
	memory_type	nvarchar(20)

Cpu_product

	Nome colonna	Tipo abbreviato
🔑	cpu_code	int
	brand	varchar(10)
	series	nvarchar(50)
	name	nvarchar(50)
	n_cores	int
	socket	varchar(30)

Ram_product

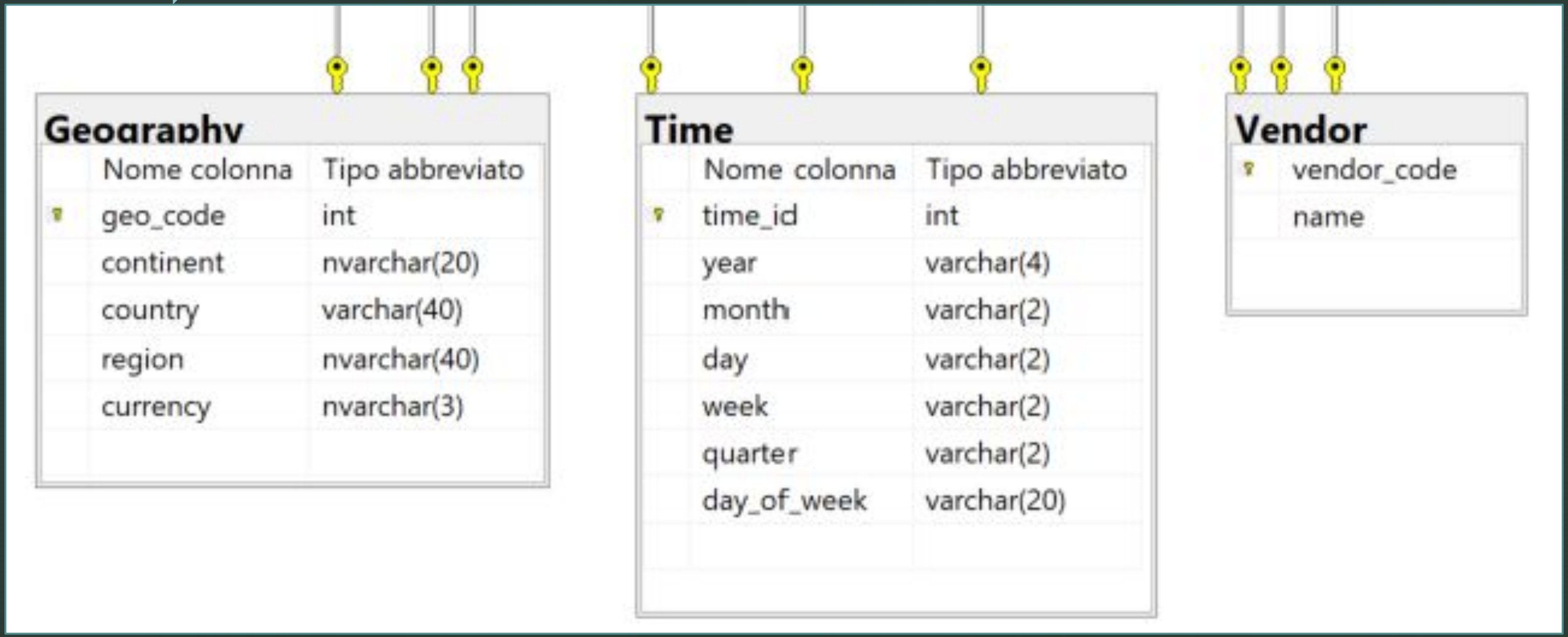
	Nome colonna	Tipo abbreviato
🔑	ram_code	int
	brand	varchar(50)
	name	varchar(50)
	memory	float
	memory_type	varchar(50)
	clock	int

Gpu_product, Cpu_product and Ram_product are three dimension tables.

Gpu_code, Cpu_code and Ram_code are the keys of the tables.

Each table is linked only to the relevant fact table.

The shared dimension tables



Geography

	Nome colonna	Tipo abbreviato
🔑	geo_code	int
	continent	nvarchar(20)
	country	varchar(40)
	region	nvarchar(40)
	currency	nvarchar(3)

Time

	Nome colonna	Tipo abbreviato
🔑	time_id	int
	year	varchar(4)
	month	varchar(2)
	day	varchar(2)
	week	varchar(2)
	quarter	varchar(2)
	day_of_week	varchar(20)

Vendor

🔑	vendor_code
	name

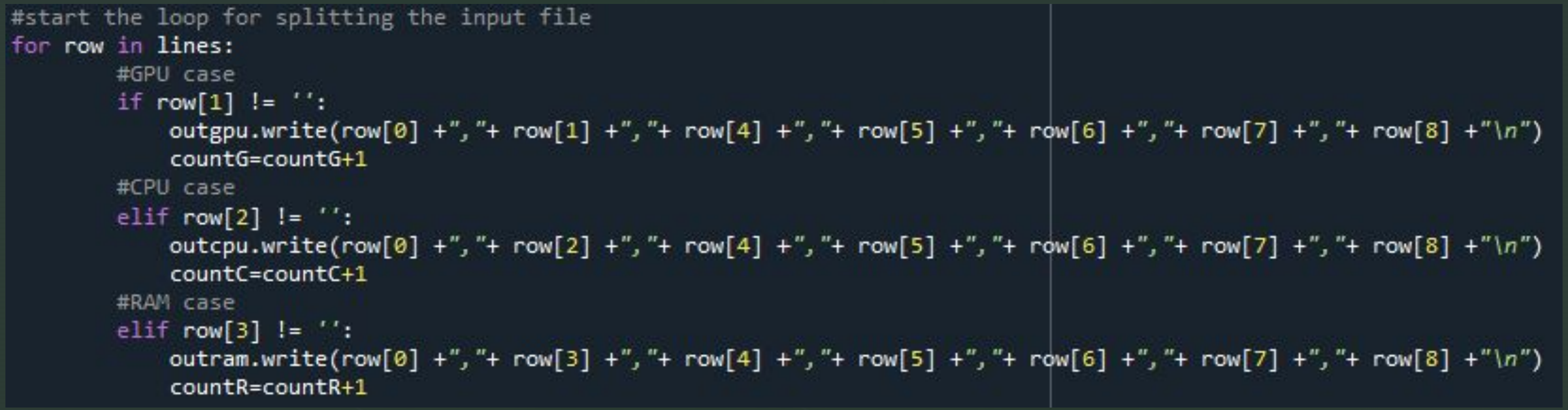
Geography, Time and Vendor are three dimension tables.

Geo_code, time_id and vendor_code are the keys of the tables.

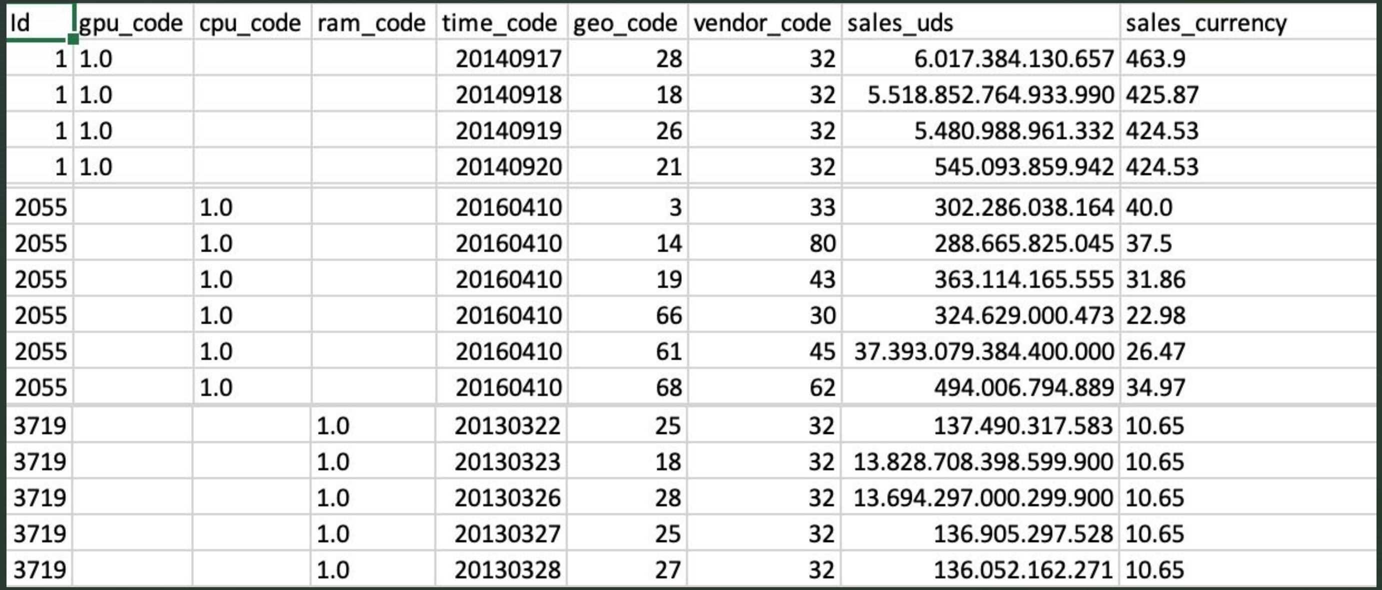
All the three tables are linked with each of the fact tables.

1.1

Fact.csv file preparation



```
#start the loop for splitting the input file
for row in lines:
        #GPU case
        if row[1] != ‘’:
            outgpu.write(row[0] +”,”+ row[1] +”,”+ row[4] +”,”+ row[5] +”,”+ row[6] +”,”+ row[7] +”,”+ row[8] +”\n”)
            countG=countG+1
        #CPU case
        elif row[2] != ‘’:
            outcpu.write(row[0] +”,”+ row[2] +”,”+ row[4] +”,”+ row[5] +”,”+ row[6] +”,”+ row[7] +”,”+ row[8] +”\n”)
            countC=countC+1
        #RAM case
        elif row[3] != ‘’:
            outram.write(row[0] +”,”+ row[3] +”,”+ row[4] +”,”+ row[5] +”,”+ row[6] +”,”+ row[7] +”,”+ row[8] +”\n”)
            countR=countR+1
```



Id	gpu_code	cpu_code	ram_code	time_code	geo_code	vendor_code	sales_uds	sales_currency
1	1.0			20140917	28	32	6.017.384.130.657	463.9
1	1.0			20140918	18	32	5.518.852.764.933.990	425.87
1	1.0			20140919	26	32	5.480.988.961.332	424.53
1	1.0			20140920	21	32	545.093.859.942	424.53
2055		1.0		20160410	3	33	302.286.038.164	40.0
2055		1.0		20160410	14	80	288.665.825.045	37.5
2055		1.0		20160410	19	43	363.114.165.555	31.86
2055		1.0		20160410	66	30	324.629.000.473	22.98
2055		1.0		20160410	61	45	37.393.079.384.400.000	26.47
2055		1.0		20160410	68	62	494.006.794.889	34.97
3719			1.0	20130322	25	32	137.490.317.583	10.65
3719			1.0	20130323	18	32	13.828.708.398.599.900	10.65
3719			1.0	20130326	28	32	13.694.297.000.299.900	10.65
3719			1.0	20130327	25	32	136.905.297.528	10.65
3719			1.0	20130328	27	32	136.052.162.271	10.65

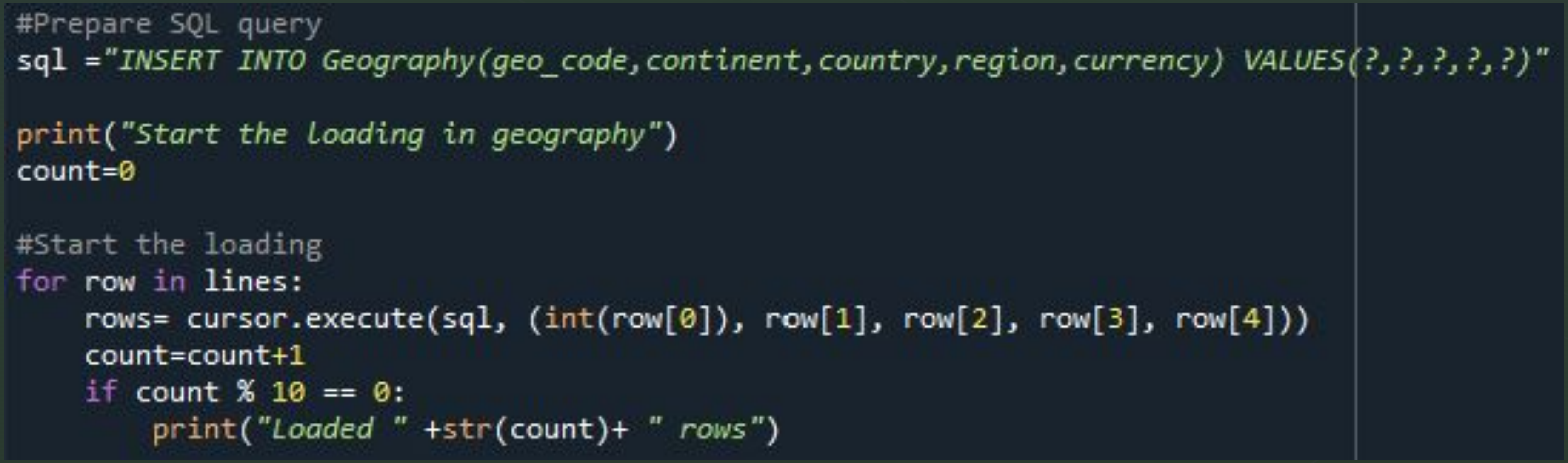
The piece of code which split the file fact.csv in three different files depending on the type of the row.

File loading

After that we have created the DW and we have set up all the files we can proceed with the loading.

For the connection to the DW we have used 'pyodbc' library.

The loading start with the load of all the dimension tables and then with the load of the fact table.



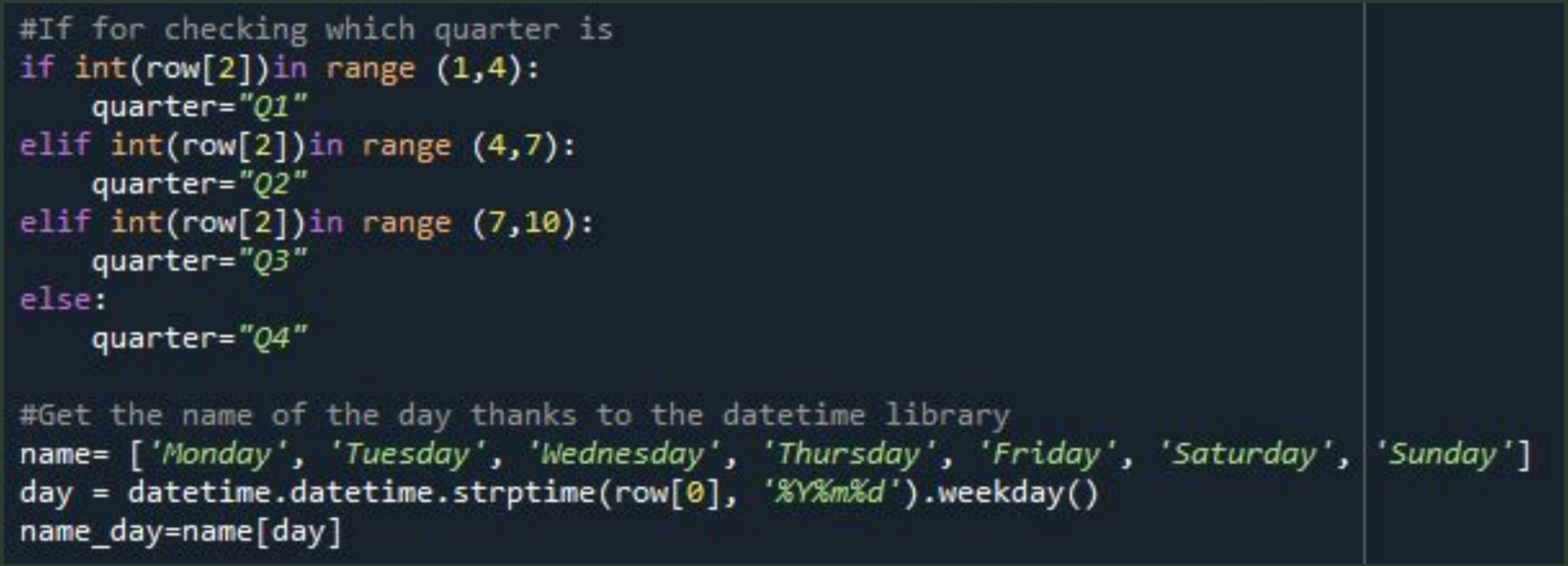
```
#Prepare SQL query
sql ="INSERT INTO Geography(geo_code,continent,country,region,currency) VALUES(?,?,?,?,?)"

print("Start the loading in geography")
count=0

#Start the loading
for row in lines:
    rows= cursor.execute(sql, (int(row[0]), row[1], row[2], row[3], row[4]))
    count=count+1
    if count % 10 == 0:
        print("Loaded " +str(count)+ " rows")
```

1.2

For the time table we had to create the attribute quarter, starting from the month number, and the attribute day_of_week, which is the name of the day. For this last attribute we have used the library 'datetime'.



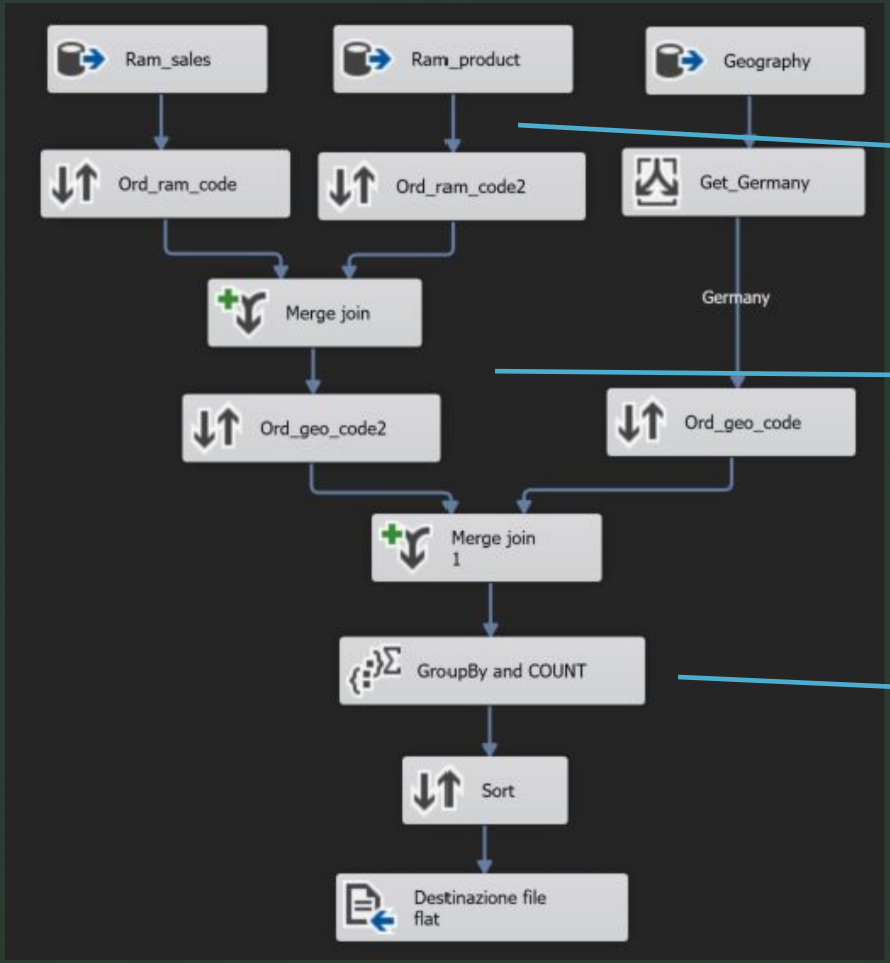
```
#If for checking which quarter is
if int(row[2])in range (1,4):
    quarter="Q1"
elif int(row[2])in range (4,7):
    quarter="Q2"
elif int(row[2])in range (7,10):
    quarter="Q3"
else:
    quarter="Q4"

#Get the name of the day thanks to the datetime library
name= ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
day = datetime.datetime.strptime(row[0], '%Y%m%d').weekday()
name_day=name[day]
```

- **We can now move to the second part of the project, the business question using SSIS.**

1. For every region of Germany, the brand of ram ordered by sales.
2. For every brand of gpu, calculate the ratio between sales during weekdays and sales during the weekend.
3. Calculate which type of product, cpu, gpu or ram, yields the most sales for each continent.

For every region of Germany, the brand of ram ordered by sales.



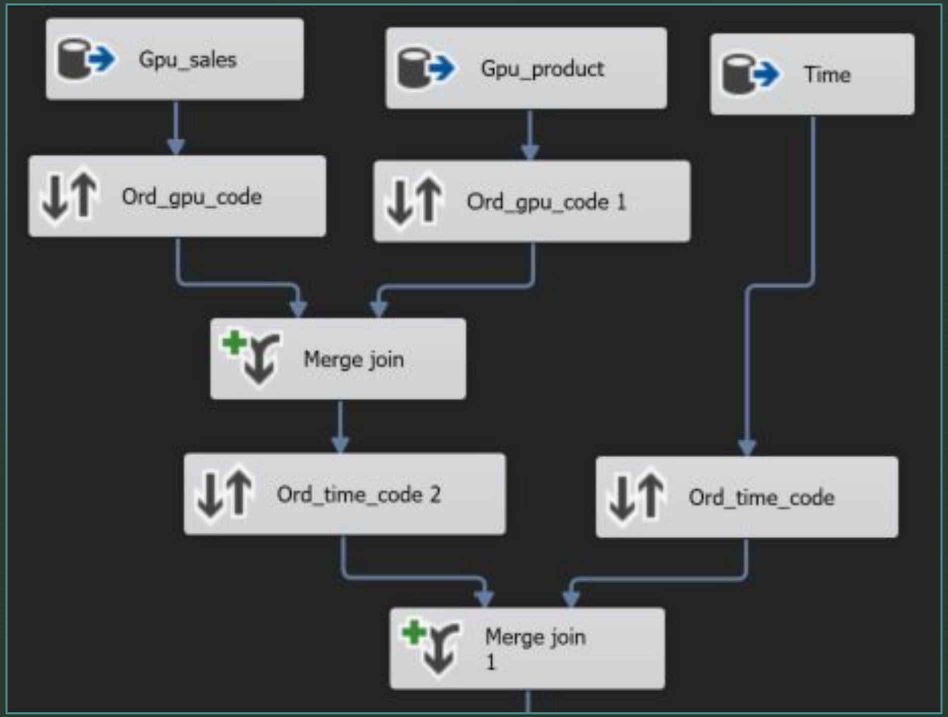
In the upper part we access the tables that we need, importing only the attributes needed and filtering it.

Then, we can sort the tables on the key and join them.

At the end we can group by 'brand' and 'region', and count all to get the solution.

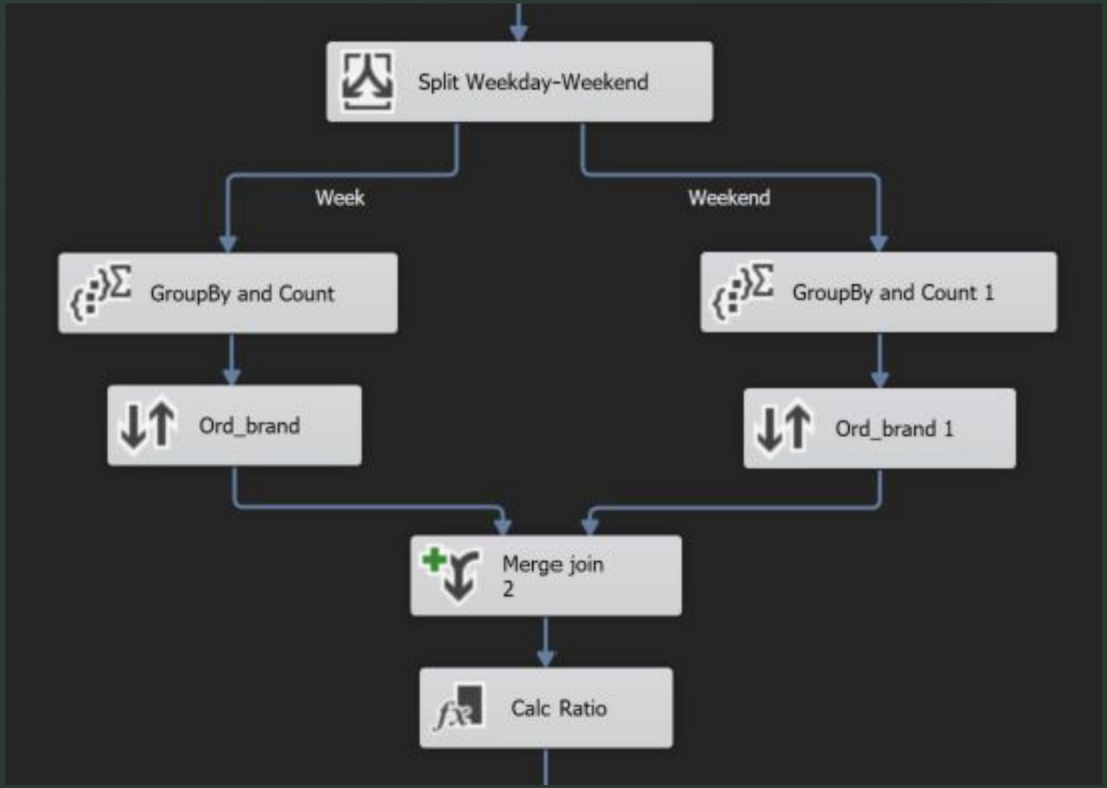
2.1

For every brand of gpu, calculate the ratio between sales during weekdays and sales during the weekend.



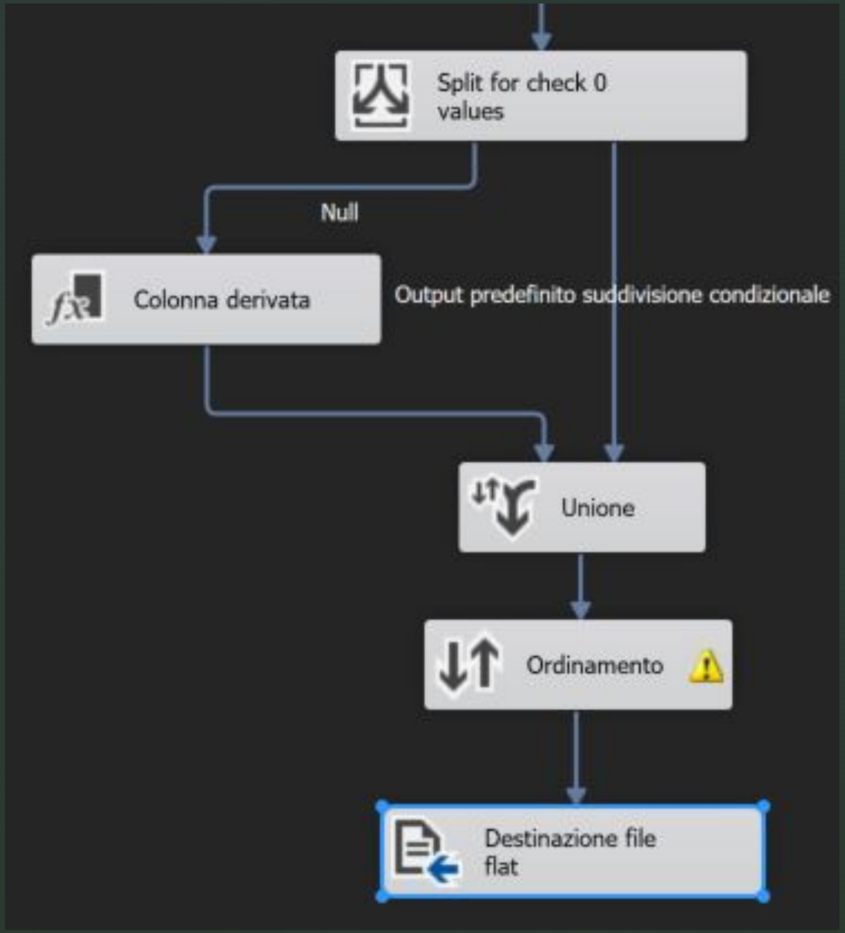
- The data flow starts with the access of the tables needed.
- Then it continues with the sort on the key.
- And then we can proceed with the merge join.

For every brand of gpu, calculate the ratio between sales during weekdays and sales during the weekend.



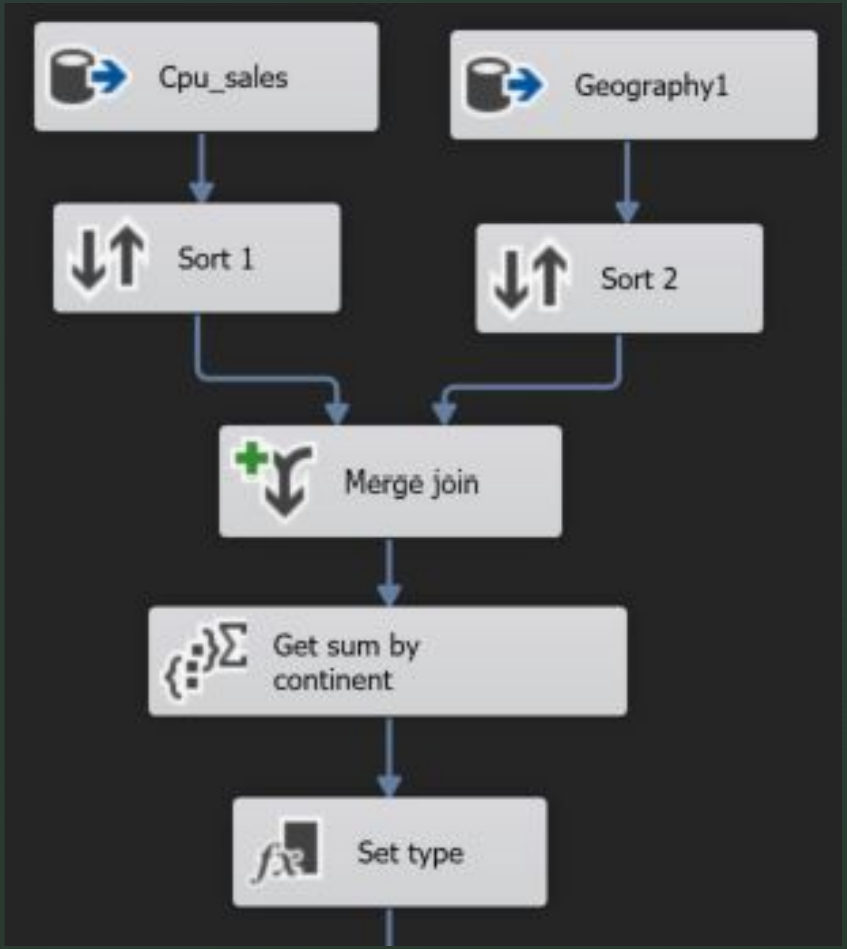
- After the access to the data, we can proceed splitting them into 2 parts, one for the weekdays and one for the weekends.
- Then, for each part we group on the brand, we count all and we merge everything using brand as join attribute.
- At the end we calc the ratio dividing the count value of the weekday by the count value of the weekends.
- In the block 'Calc Ratio' we check also possible null value for the count of the weekends.

For every brand of gpu, calculate the ratio between sales during weekdays and sales during the weekend.



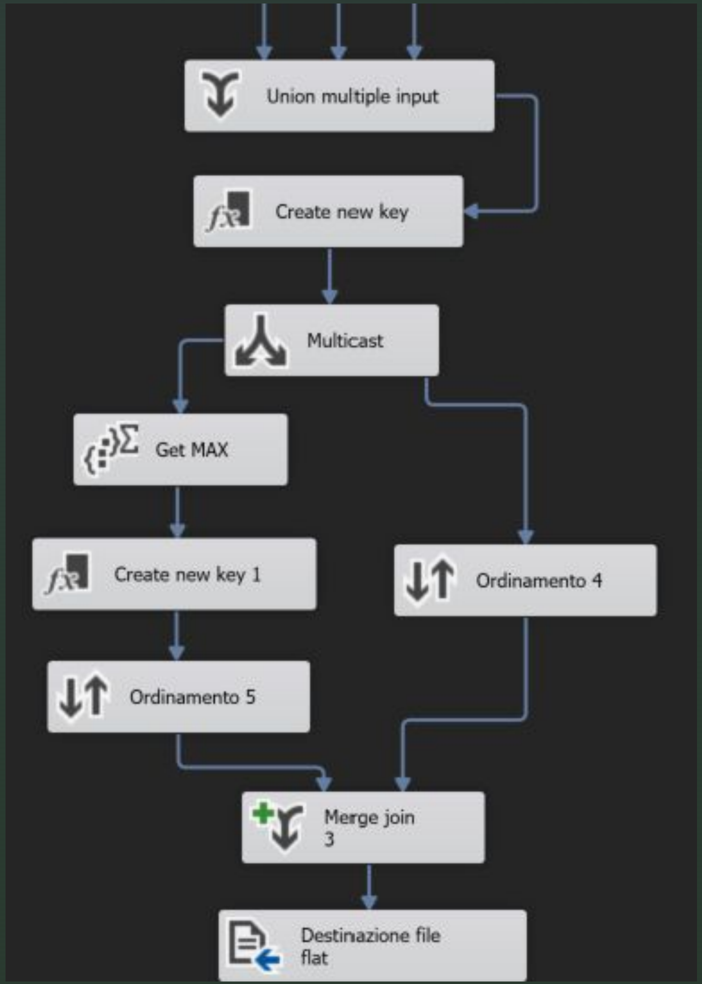
- In the last part we check for the null value in order to avoid eventually division for zero.
- So, we split on the variable previously created getting when the variable is null.
- On this branch, we insert on the ratio the value '00'.
- At the end we unite everything and we sort again getting only the relevant attributes.

Calculate which type of product, cpu, gpu or ram, yields the most sales for each continent.



- Initially for each fact table we execute this data flow.
- So, we import the fact table and the geography table and we merge them.
- Then we group by 'continent' and we get the sum of the attribute 'sales_usd'.
- Finally we add a column which report the type of the product, so cpu, ram or gpu.

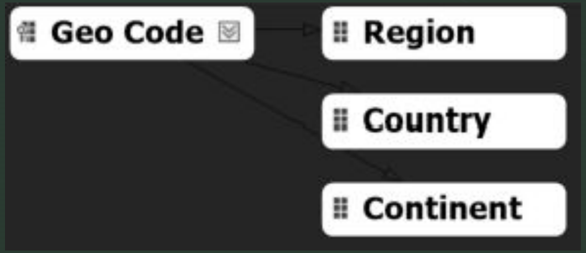
Calculate which type of product, cpu, gpu or ram, yields the most sales for each continent.

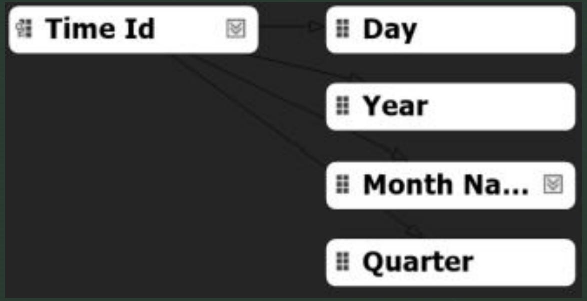


- Then we unite all the data provided form the three fact tables.
- After, we split the data in 2 equal part, the right part will be used for retrieving the type of the sales and the left part for group by continent and get the max value.
- At the end we merge all using as join attribute the combination of the sum of 'sales_usd' and the 'continent'.

Ram cube construction

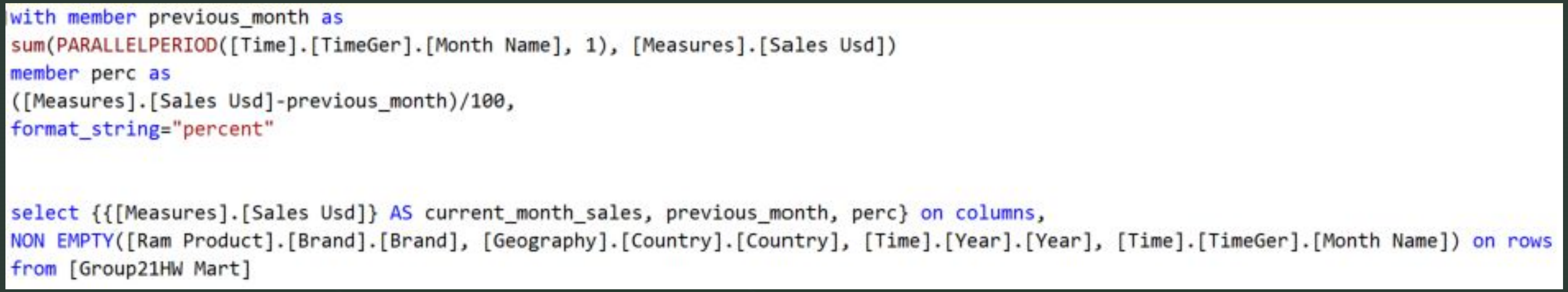
- Now, we can move to the third part of the project, the ram cube construction.
- This has been done using SSAS extension on Visual Studio.
- In the cube, we created two hierarchies, one for time and one for geography.





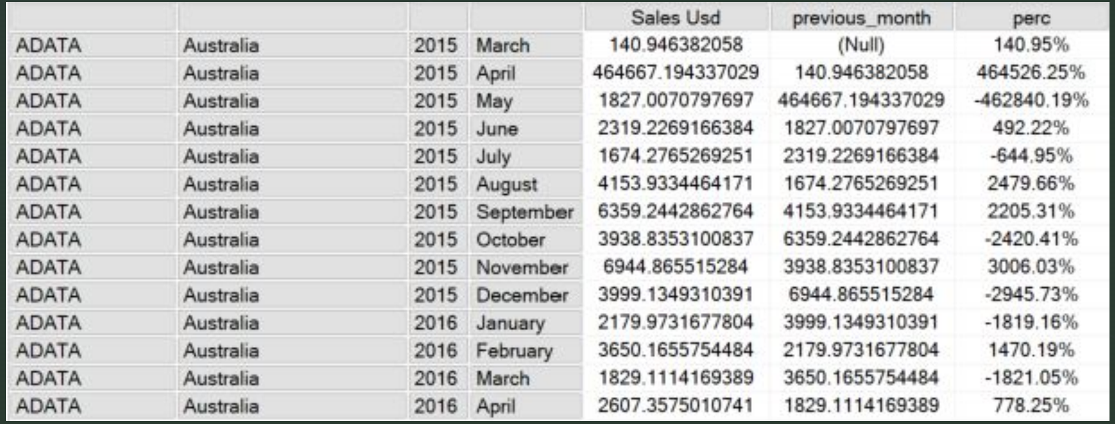
The attribute 'Month Name' has been created in the cube and it is ordered by the attribute 'Month'

Show the percentage increase in total sales with respect to the previous month for each ram brand and each country.



```
with member previous_month as
sum(PARALLELPERIOD([Time].[TimeGer].[Month Name], 1), [Measures].[Sales Usd])
member perc as
([Measures].[Sales Usd]-previous_month)/100,
format_string="percent"


select {{[Measures].[Sales Usd]} AS current_month_sales, previous_month, perc} on columns,
NON EMPTY([Ram Product].[Brand].[Brand], [Geography].[Country].[Country], [Time].[Year].[Year], [Time].[TimeGer].[Month Name]) on rows
from [Group21HW Mart]
```

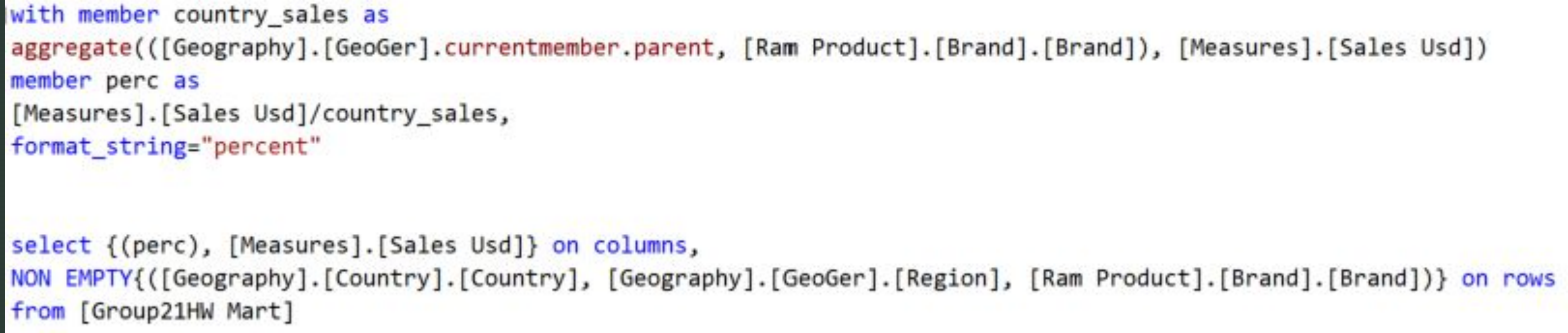


				Sales Usd	previous_month	perc
ADATA	Australia	2015	March	140.946382058	(Null)	140.95%
ADATA	Australia	2015	April	464667.194337029	140.946382058	464526.25%
ADATA	Australia	2015	May	1827.0070797697	464667.194337029	-462840.19%
ADATA	Australia	2015	June	2319.2269166384	1827.0070797697	492.22%
ADATA	Australia	2015	July	1674.2765269251	2319.2269166384	-644.95%
ADATA	Australia	2015	August	4153.9334464171	1674.2765269251	2479.66%
ADATA	Australia	2015	September	6359.2442862764	4153.9334464171	2205.31%
ADATA	Australia	2015	October	3938.8353100837	6359.2442862764	-2420.41%
ADATA	Australia	2015	November	6944.865515284	3938.8353100837	3006.03%
ADATA	Australia	2015	December	3999.1349310391	6944.865515284	-2945.73%
ADATA	Australia	2016	January	2179.9731677804	3999.1349310391	-1819.16%
ADATA	Australia	2016	February	3650.1655754484	2179.9731677804	1470.19%
ADATA	Australia	2016	March	1829.1114169389	3650.1655754484	-1821.05%
ADATA	Australia	2016	April	2607.3575010741	1829.1114169389	778.25%

In the variable 'previous_month', with the function PARALLELPERIOD, we get the value of the previous month; this will be used then for the computation of the percentage, which is shown in the variable 'perc'. We plotted also the year in such a way to retrieve easily the month in the time.

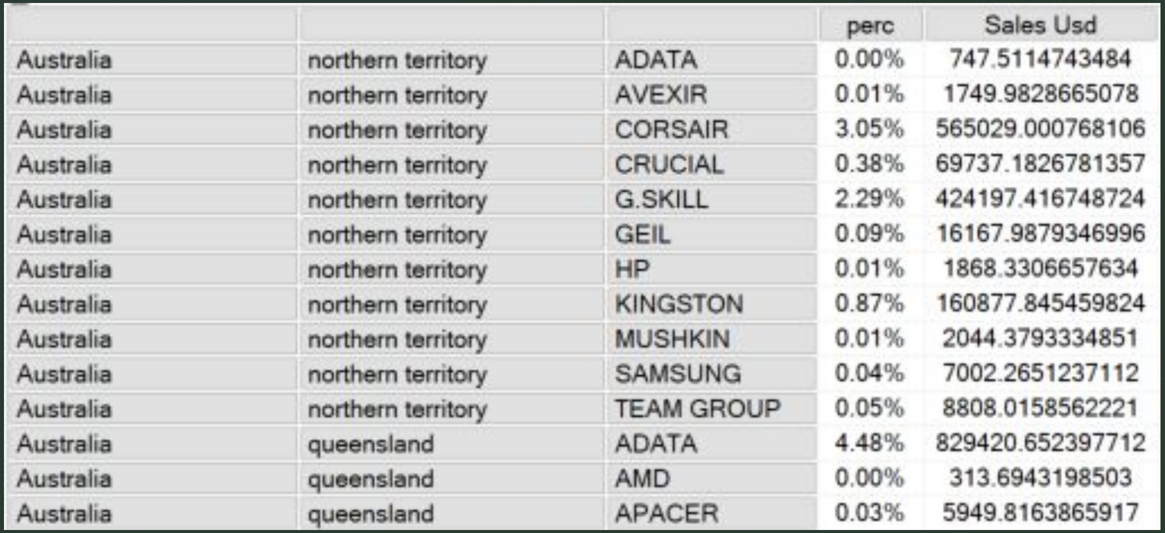
3.2

For each region and ram brand show the total sales in percentage with respect to the total sales of the corresponding country.



```
with member country_sales as
aggregate(([Geography].[GeoGer].currentmember.parent, [Ram Product].[Brand].[Brand]), [Measures].[Sales Usd])
member perc as
[Measures].[Sales Usd]/country_sales,
format_string="percent"


select {(perc), [Measures].[Sales Usd]} on columns,
NON EMPTY{([Geography].[Country].[Country], [Geography].[GeoGer].[Region], [Ram Product].[Brand].[Brand])} on rows
from [Group21HW Mart]
```

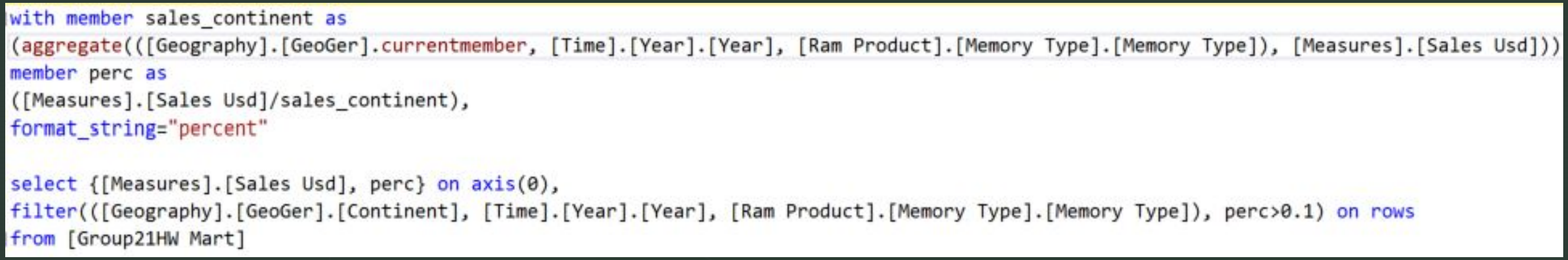


			perc	Sales Usd
Australia	northern territory	ADATA	0.00%	747.5114743484
Australia	northern territory	AVEXIR	0.01%	1749.9828665078
Australia	northern territory	CORSAIR	3.05%	565029.000768106
Australia	northern territory	CRUCIAL	0.38%	69737.1826781357
Australia	northern territory	G.SKILL	2.29%	424197.416748724
Australia	northern territory	GEIL	0.09%	16167.9879346996
Australia	northern territory	HP	0.01%	1868.3306657634
Australia	northern territory	KINGSTON	0.87%	160877.845459824
Australia	northern territory	MUSHKIN	0.01%	2044.3793334851
Australia	northern territory	SAMSUNG	0.04%	7002.2651237112
Australia	northern territory	TEAM GROUP	0.05%	8808.0158562221
Australia	queensland	ADATA	4.48%	829420.652397712
Australia	queensland	AMD	0.00%	313.6943198503
Australia	queensland	APACER	0.03%	5949.8163865917

In 'country_sales' we calc the total sales in the country; then we use this variable for obtain 'perc' which show the percentage of the sales of each ram brand in each region with respect to the total sales of the country.

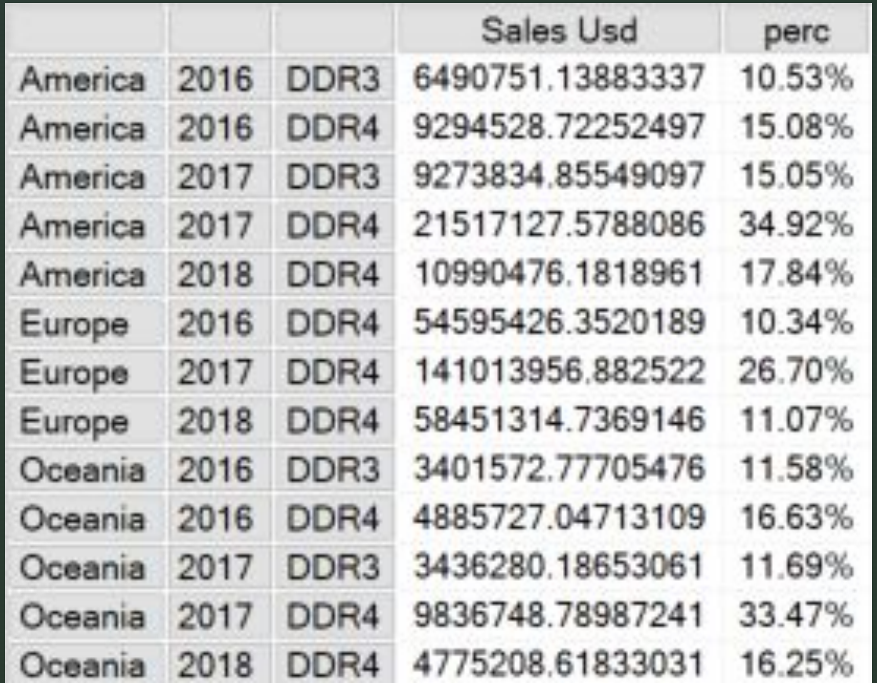
3.3

Show the ram memory types having a total sales greater than 10% of the totals sales in each continent by continent and year.



```
with member sales_continent as
(aggregate(([Geography].[GeoGer].currentmember, [Time].[Year].[Year], [Ram Product].[Memory Type].[Memory Type]), [Measures].[Sales Usd]))
member perc as
([Measures].[Sales Usd]/sales_continent),
format_string="percent"

select {[Measures].[Sales Usd], perc} on axis(0),
filter(([Geography].[GeoGer].[Continent], [Time].[Year].[Year], [Ram Product].[Memory Type].[Memory Type]), perc>0.1) on rows
from [Group21HW Mart]
```

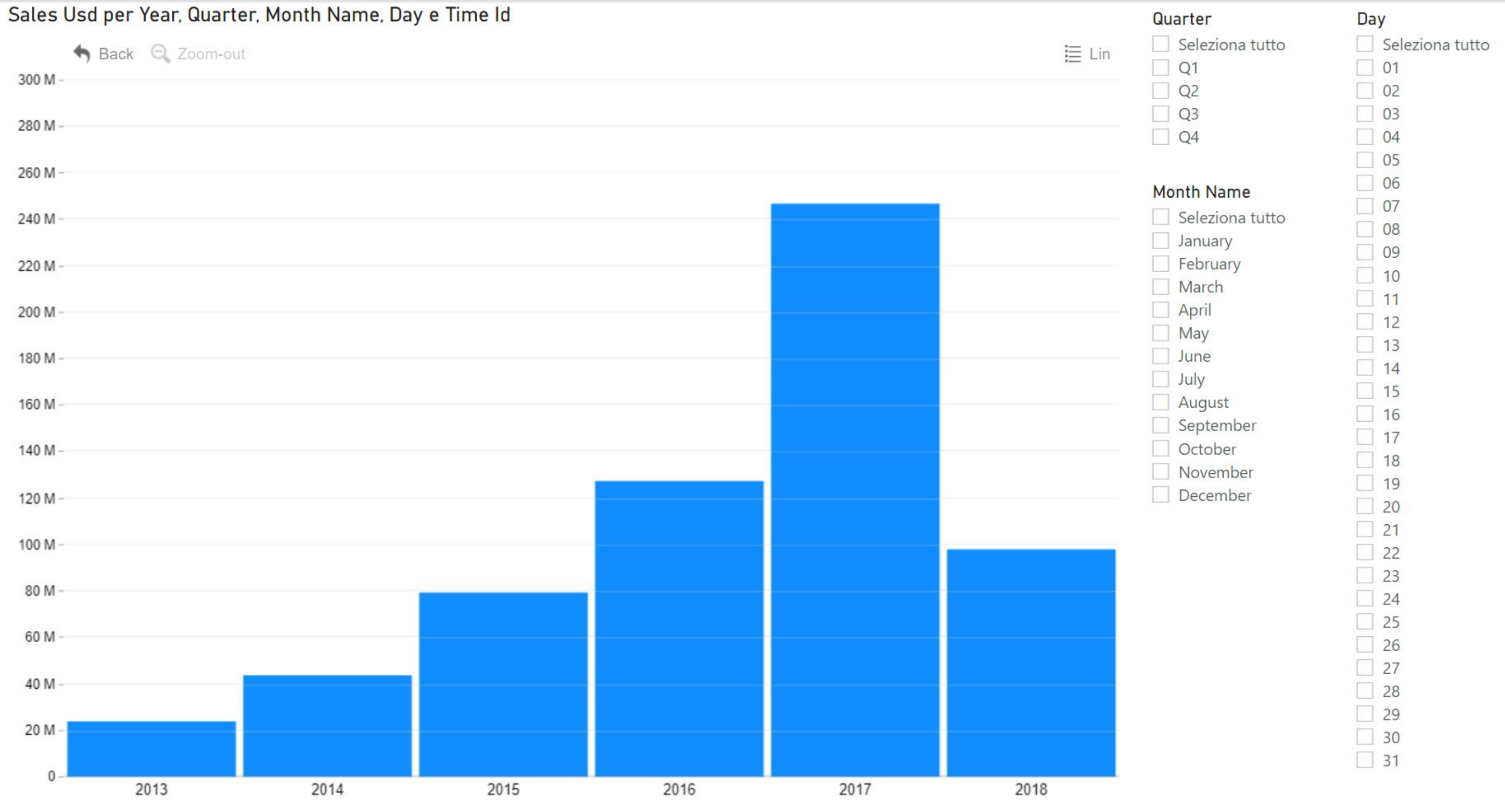


			Sales Usd	perc
America	2016	DDR3	6490751.13883337	10.53%
America	2016	DDR4	9294528.72252497	15.08%
America	2017	DDR3	9273834.85549097	15.05%
America	2017	DDR4	21517127.5788086	34.92%
America	2018	DDR4	10990476.1818961	17.84%
Europe	2016	DDR4	54595426.3520189	10.34%
Europe	2017	DDR4	141013956.882522	26.70%
Europe	2018	DDR4	58451314.7369146	11.07%
Oceania	2016	DDR3	3401572.77705476	11.58%
Oceania	2016	DDR4	4885727.04713109	16.63%
Oceania	2017	DDR3	3436280.18653061	11.69%
Oceania	2017	DDR4	9836748.78987241	33.47%
Oceania	2018	DDR4	4775208.61833031	16.25%

As before, in 'sales_continent' we compute the total sales in the continent. Then in 'perc' we calculate the percentage of the sales with respect to 'sales_continent'. In the part of the select we filtered by selecting only the rows with percentage greater than 10%.

3.4

Create a dashboard that shows how sales change over time, giving the user the opportunity to see the sales behavior for different time granularity.


Sales Usd per Year, Quarter, Month Name, Day e Time Id

Quarter

□ Seleziona tutto
□ Q1
□ Q2
□ Q3
□ Q4

Month Name

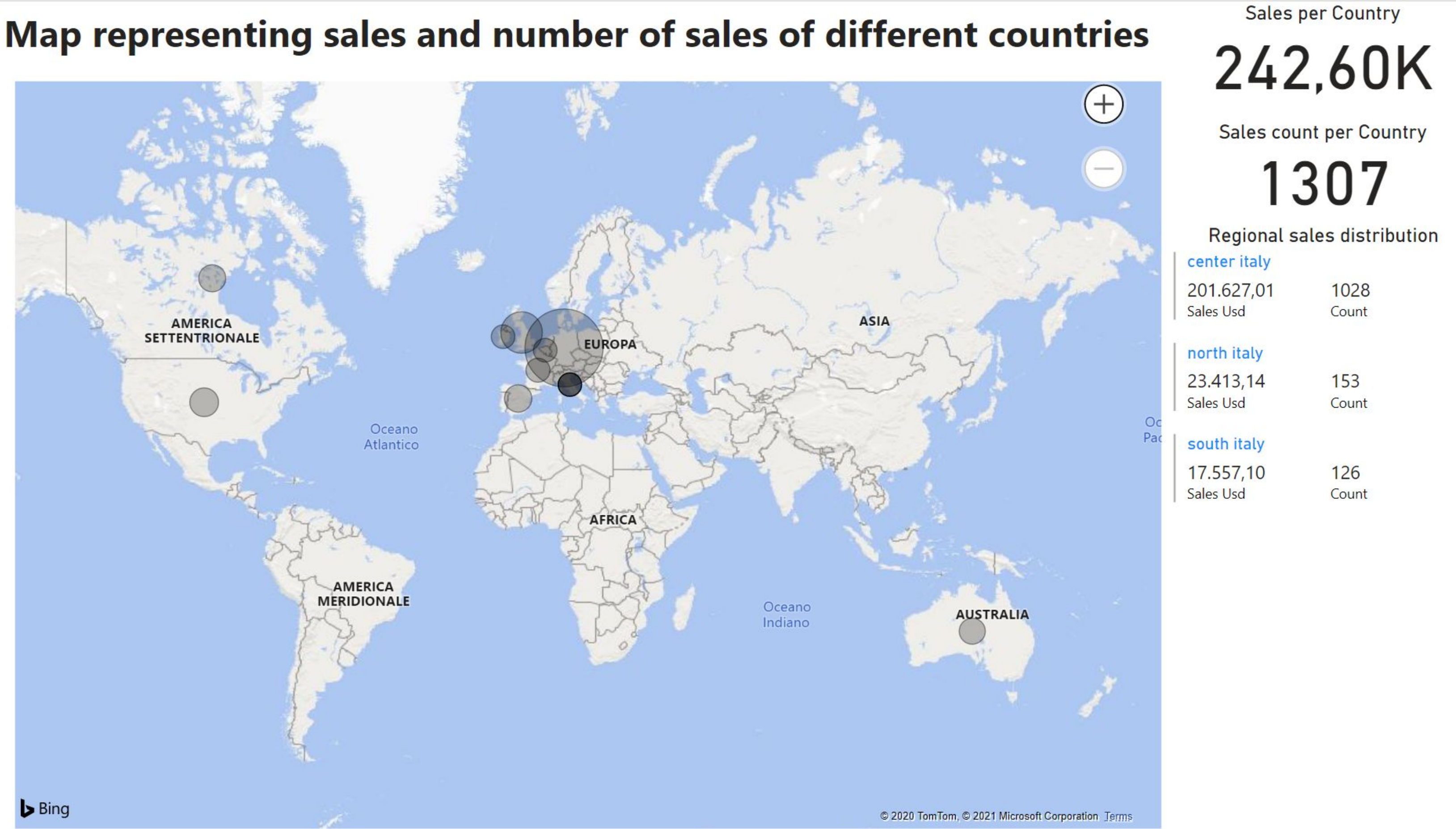
□ Seleziona tutto
□ January
□ February
□ March
□ April
□ May
□ June
□ July
□ August
□ September
□ October
□ November
□ December

Day

□ Seleziona tutto
□ 01
□ 02
□ 03
□ 04
□ 05
□ 06
□ 07
□ 08
□ 09
□ 10
□ 11
□ 12
□ 13
□ 14
□ 15
□ 16
□ 17
□ 18
□ 19
□ 20
□ 21
□ 22
□ 23
□ 24
□ 25
□ 26
□ 27
□ 28
□ 29
□ 30
□ 31

3.5

Show the geographical distribution of sales and of the number of products purchased.



Map representing sales and number of sales of different countries

Sales per Country

242,60K

Sales count per Country

1307

Regional sales distribution

center italy

201.627,01 Sales Usd | 1028 Count

north italy

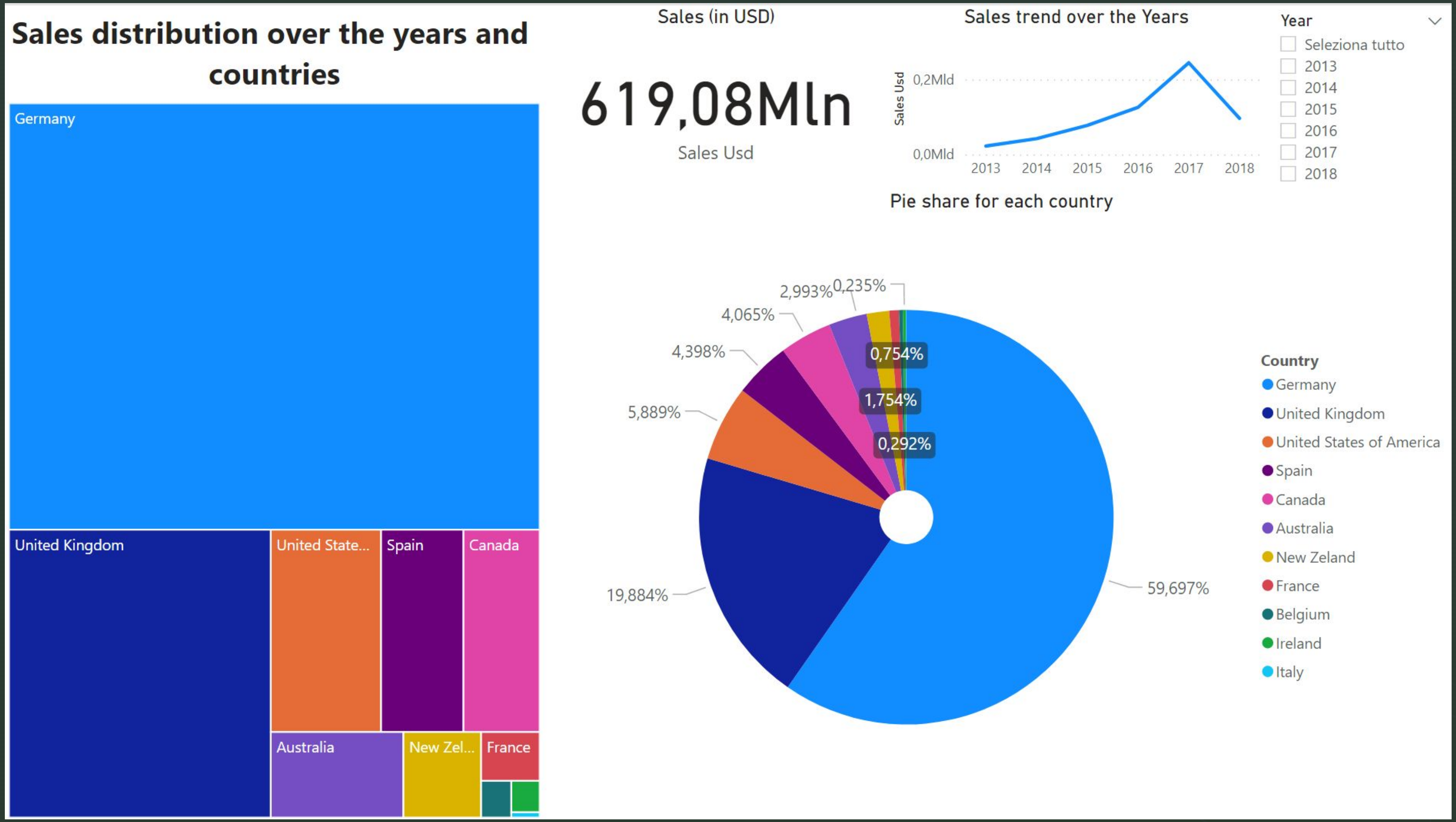
23.413,14 Sales Usd | 153 Count

south italy

17.557,10 Sales Usd | 126 Count

3.6

Create a plot/dashboard of your choosing, that you deem interesting w.r.t. the data available in your cube


Sales distribution over the years and countries
Germany
United Kingdom
United State...
Spain
Canada
Australia
New Zel...
France
Sales (in USD)
619,08Mln
Sales Usd
Sales trend over the Years
Sales Usd
0,2Mld
0,0Mld
2013
2014
2015
2016
2017
2018
Year
□ Seleziona tutto
□ 2013
□ 2014
□ 2015
□ 2016
□ 2017
□ 2018
Pie share for each country
2,993%
0,235%
4,065%
0,754%
4,398%
1,754%
5,889%
0,292%
19,884%
59,697%
Country
● Germany
● United Kingdom
● United States of America
● Spain
● Canada
● Australia
● New Zeland
● France
● Belgium
● Ireland
● Italy