Modern Data Management & Business Intelligence

Assignment 2

*Professor*

Damianos Chatziantoniou

*Authors*

Christos Katsaris

Pavlos Polyzogopoulos

Athens,

December 2017

Contents

[1. Introduction 3](#_Toc500121774)

[2. Building the database and description of ETL process 4](#_Toc500121775)

[3. Cube 14](#_Toc500121776)

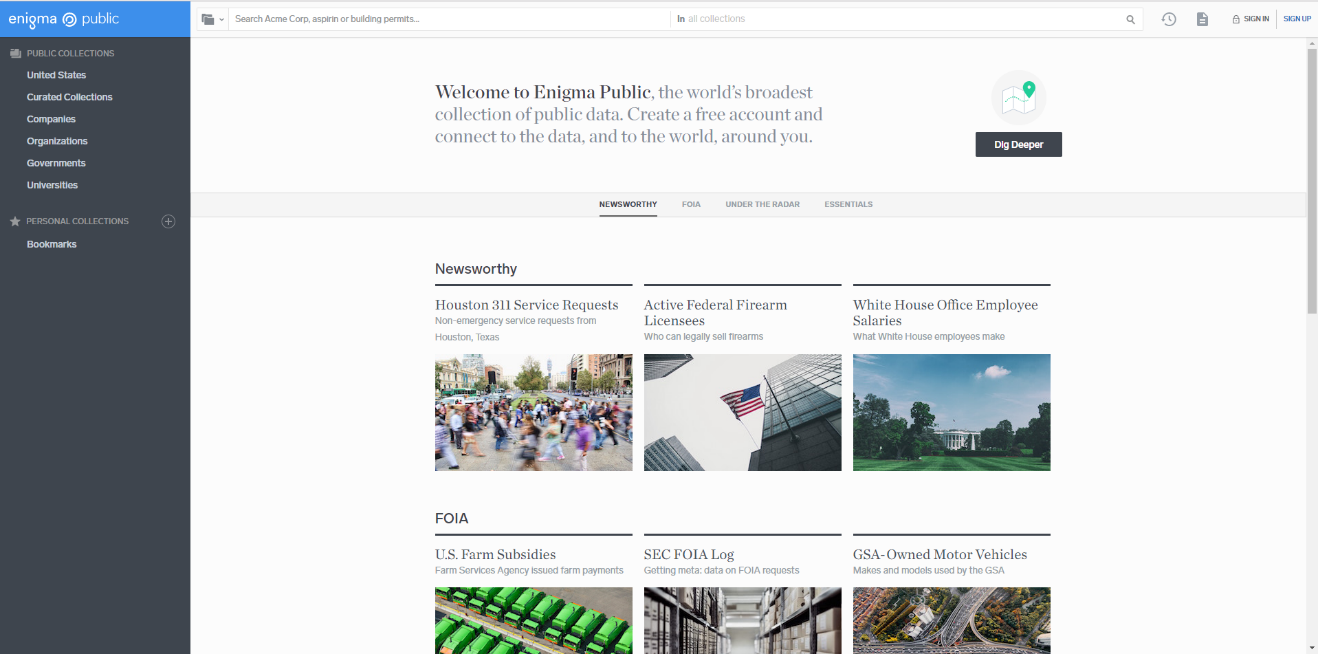
[4. Visualization 19](#_Toc500121777)

[5. Conclusions 26](#_Toc500121778)

# Introduction

An infamous as well as troublesome issue we need to address is climate change and how it affects the actual weather throughout the years, but also the everyday life. In this assignment we chose to deal with weather data and in particular to observe climate change and differences in various weather measures during the last years. Through the analysis we are going to demonstrate if there is a climate change trend, shown through changes in weather conditions, and if actually the phenomenon has an effect on everyday life.

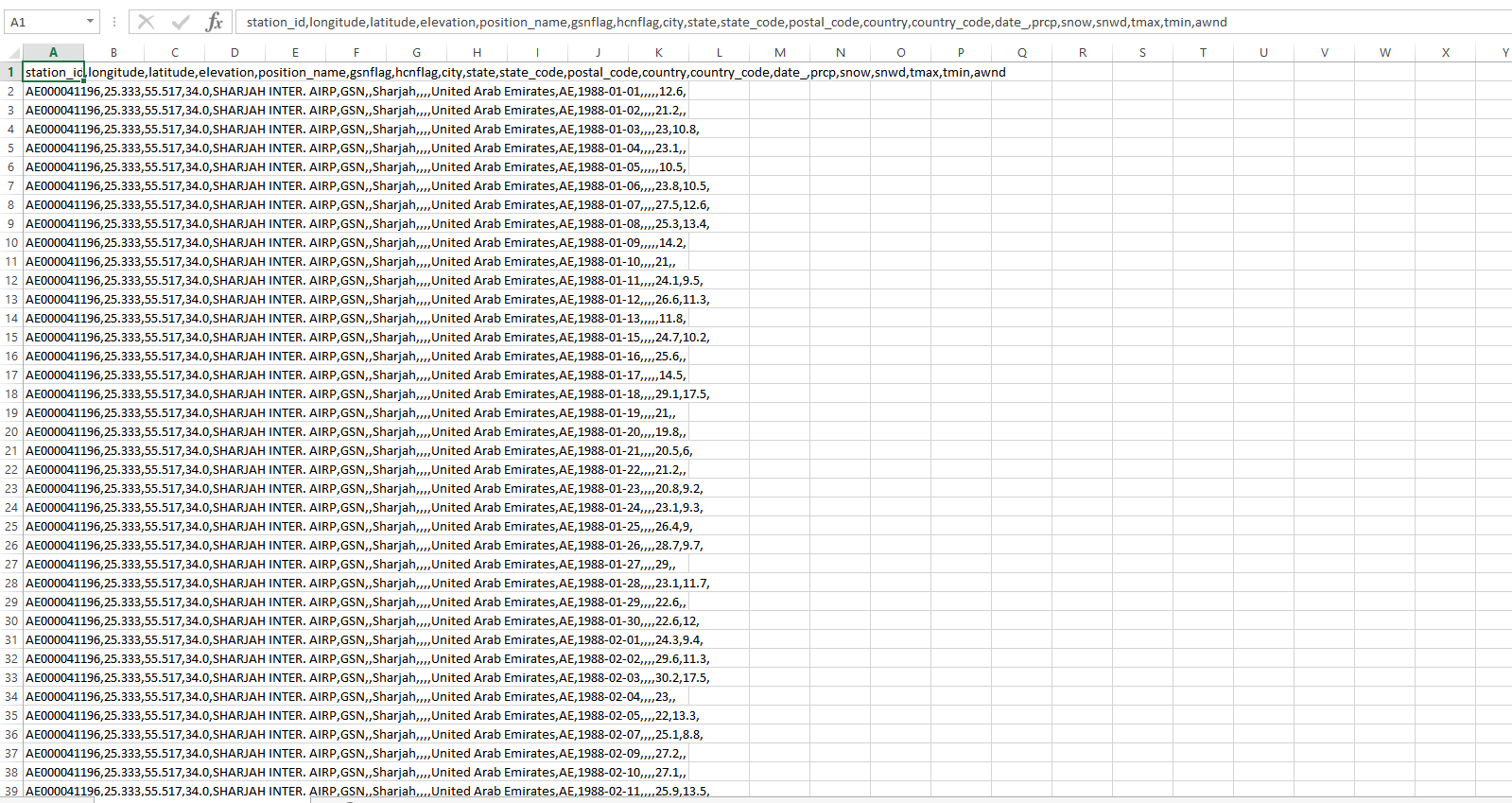
The first step was to find the proper dataset for our analysis. A dataset that would involve a lot of information about weather measurements through different areas and years. After a lot of searching online we came across <https://public.enigma.com> website.



In this site there are many public datasets and we found our datasets about the weather. The datasets were provided by National Oceanic and Atmospheric Administration (NOAA) which is an American scientific agency within the United States Department of Commerce that focuses on the conditions of the oceans and the atmosphere. We downloaded the weather reports for five different years, 2012, 2008, 2004, 1998 and 1988. The goal is to see if we can spot any differences or anomalies in the weather conditions from 1980s to the current decade within this 30 years period.

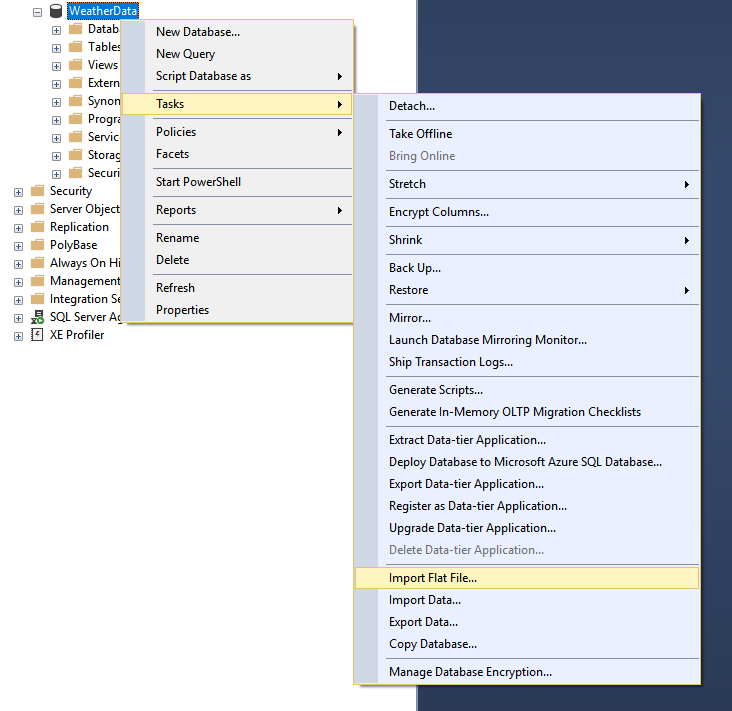
# Building the database and description of ETL process

After we found the datasets we were looking for, we began to build up our database. The datasets were in csv format and it was one for every year. Every dataset contained twenty columns and four- four and half million rows. This is a considerably large volume of data. Our datasets looked like this in the csv format.

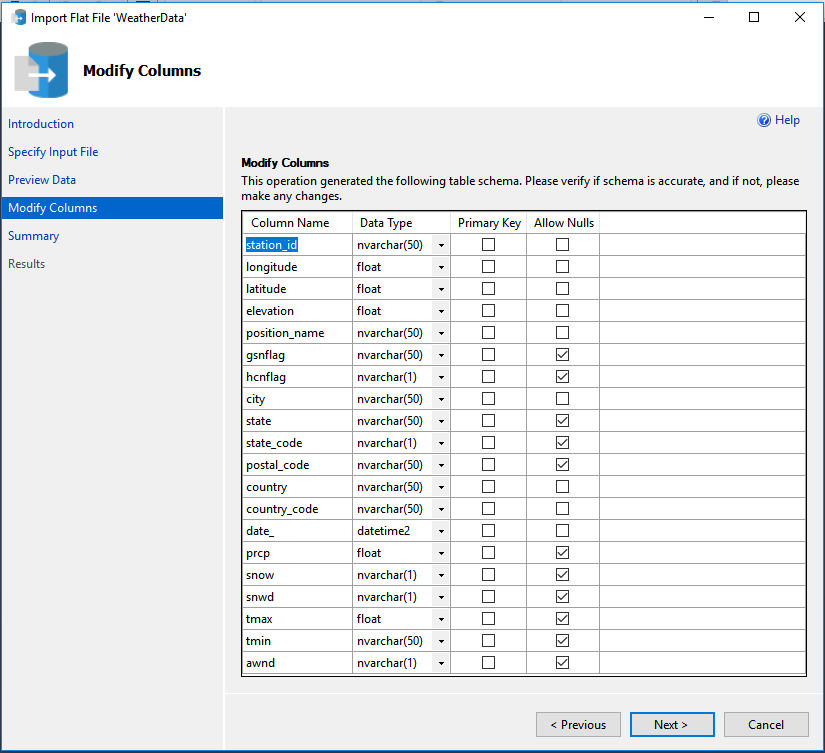


The first step was to import the csv files into our database in order to create five tables of the same format.

This import procedure was done from the SSMS console of the SQL server by right clicking on the database -> tasks -> import flat file as we can see below.

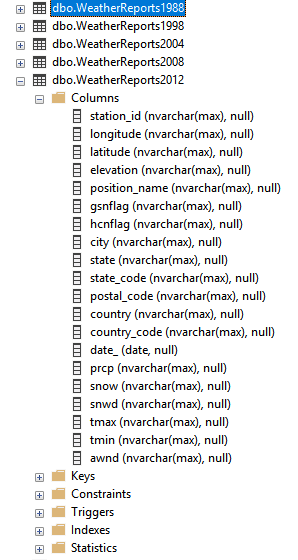


With this selection opens a wizard that allows to import csv files and turn them into tables. By clicking next in the first window, we are in a point where we select the path for the csv file and a name for the table that will be created. All the tables that created with the import method were given WeatherReportsX where X is the year of the dataset. For example, the table for 2012 haw the name WeatherReports2012 and so on. After that step follows a preview of the data and with next there is a screen where we can make some changes. The changes are about the column name, the data type, if a column is a primary key and if it allows nulls.



By clicking next and the finish in the last screen the import process begins.

In the first attempt we tried to import the data in the “correct” format meaning the fields that where numeric as decimals, integers, the stringfield as nvarchar etc. But that didn’t work well for the data because the arithmetic fields weren’t in the correct format when imported. So just for the import process we decided to have all fields as nvarchars (except the date field) and then change them to data types we wanted in another step.



After the imports of the csv files we had the tables one of each year and we ready to continue with our transformation and cleaning processes in order to reach a good format for our fact and dimension tables.

The next step was to merge all five tables into one big table and change the data types to the fields that needed to be changed. Since all the five tables had the same columns we merged them into one big table with a union command.

First we wrote the following SQL statement to create the table that named WeatherReportsAllYears :

Create table WeatherReportsAllYears (

ID int IDENTITY(1,1) PRIMARY KEY,

StationID nvarchar(max),

Longitude float ,

Latitude float,

Elevation decimal(10,1),

PositionName nvarchar(max),

City nvarchar(max),

State nvarchar(max),

StateCode nvarchar(max),

PostalCode nvarchar(max),

Country nvarchar(max),

CountryCode nvarchar(max),

Date date,

Precipitation decimal(10,1),

Snow decimal(10,1),

SnowDepth decimal(10,1),

MaxTemp decimal(10,1),

MinTemp decimal(10,1),

AvgWindSpeed decimal(10,1)

)

Then after the table was created we had to write an insert statement to fill the table with data from the five years. At the same time we did all the conversions to the data types so the unified table has data with the proper format and kept only the columns that we were interested about our report. The way we did it was with the following SQL statement:

Insert into WeatherReportsAllYears (

[StationID]

,[Longitude]

,[Latitude]

,[Elevation]

,[PositionName]

,[City]

,[State]

,[StateCode]

,[PostalCode]

,[Country]

,[CountryCode]

,[Date]

,[Precipitation]

,[Snow]

,[SnowDepth]

,[MaxTemp]

,[MinTemp]

,[AvgWindSpeed])

(select

station\_id,

cast (longitude as float),

cast (latitude as float),

cast (elevation as decimal(10,1)),

position\_name,

city,

state,

state\_code,

postal\_code,

country,

country\_code,

date\_,

cast (prcp as decimal(10,1)),

cast (snow as decimal(10,1)),

cast (snwd as decimal(10,1)),

cast (tmax as decimal(10,1)),

cast (tmin as decimal(10,1)),

cast (awnd as decimal(10,1))

from WeatherReports2012

union all

select

station\_id,

cast (longitude as float),

cast (latitude as float),

cast (elevation as decimal(10,1)),

position\_name,

city,

state,

state\_code,

postal\_code,

country,

country\_code,

date\_,

cast (prcp as decimal(10,1)),

cast (snow as decimal(10,1)),

cast (snwd as decimal(10,1)),

cast (tmax as decimal(10,1)),

cast (tmin as decimal(10,1)),

cast (awnd as decimal(10,1))

from WeatherReports2008

union all

select

station\_id,

cast (longitude as float),

cast (latitude as float),

cast (elevation as decimal(10,1)),

position\_name,

city,

state,

state\_code,

postal\_code,

country,

country\_code,

date\_,

cast (prcp as decimal(10,1)),

cast (snow as decimal(10,1)),

cast (snwd as decimal(10,1)),

cast (tmax as decimal(10,1)),

cast (tmin as decimal(10,1)),

cast (awnd as decimal(10,1))

from WeatherReports2004

union all

select

station\_id,

cast (longitude as float),

cast (latitude as float),

cast (elevation as decimal(10,1)),

position\_name,

city,

state,

state\_code,

postal\_code,

country,

country\_code,

date\_,

cast (prcp as decimal(10,1)),

cast (snow as decimal(10,1)),

cast (snwd as decimal(10,1)),

cast (tmax as decimal(10,1)),

cast (tmin as decimal(10,1)),

cast (awnd as decimal(10,1))

from WeatherReports1998

union all

select

station\_id,

cast (longitude as float),

cast (latitude as float),

cast (elevation as decimal(10,1)),

position\_name,

city,

state,

state\_code,

postal\_code,

country,

country\_code,

date\_,

cast (prcp as decimal(10,1)),

cast (snow as decimal(10,1)),

cast (snwd as decimal(10,1)),

cast (tmax as decimal(10,1)),

cast (tmin as decimal(10,1)),

cast (awnd as decimal(10,1))

from WeatherReports1988)

After the above procedures were made, we ended up with a table that had 20 columns and approximately twenty-two million rows.

The next step was to create the dimension tables out of the WeatherReportsAllYears table. For every dimension table we used a create table SQL statement and then an insert like mentioned above. Also at that point we decided which columns of the original merged table we would keep as dimensions and had value for our analysis. For example, about the country dimension we used something like the following:

insert into CountryDim (

CountryCode,

CountryName

)

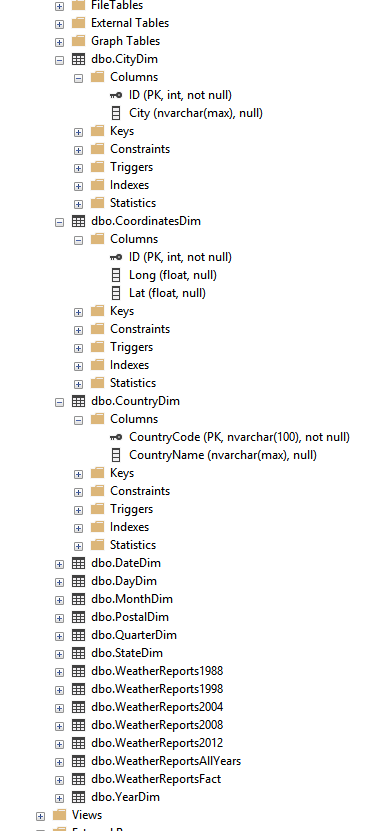
select CountryCode,Country from WeatherReportsAllYears

where CountryCode is not null

group by CountryCode,Country

Of course, before and after that process we had to check in each table if there were any weird data. For example in the country dimension we found for some cases like China that had the same country code with different description and that happened because during the years the official name of the country changed. Or we found some null values in the country codes and through the coordinates we had available, we observed that these points were very small territories that don’t belong to a certain country. We had similar problems to other dimensions as well. So for that reason, many times we had to drop some of the dimension tables, clean the data and recreate them in order to have the proper information.

Eventually we ended up with ten-dimension tables as we can see below. The dimension tables are the ones that have the ending Dim in their names.



After the dimension tables were ready we could proceed with the creation of our fact table. For the fact table we used again a create table statement as following :

create table WeatherReportsFact (

ID int IDENTITY(1,1) PRIMARY KEY,

CoordinatesID int ,

CityID int,

StateID int,

PostalID int,

CountryID nvarchar(100),

DateID int,

DayID int,

MonthID int,

YearID int,

QuarterID int,

Elevation decimal(10,1),

Precipitation decimal(10,1),

Snow decimal(10,1),

SnowDepth decimal(10,1),

MaxTemp decimal(10,1),

MinTemp decimal(10,1),

AvgWindSpeed decimal(10,1)

)

And after that we used an insert command to input our data to the fact table.

insert into WeatherReportsFact(

[CoordinatesID]

,[CityID]

,[StateID]

,[PostalID]

,[CountryID]

,[DateID]

,[DayID]

,[MonthID]

,[YearID]

,[QuarterID]

,[Elevation]

,[Precipitation]

,[Snow]

,[SnowDepth]

,[MaxTemp]

,[MinTemp]

,[AvgWindSpeed])

select coo.ID,ci.ID,s.ID,p.ID,c.CountryCode,d.ID,dd.ID,md.ID,yd.ID,qd.ID,w.Elevation,w.Precipitation,w.Snow,w.SnowDepth,w.MaxTemp,w.MinTemp,w.AvgWindSpeed

from WeatherReportsAllYears w

left join CoordinatesDim coo on (coo.Lat=w.Latitude and coo.Long=w.Longitude)

left join CountryDim c on c.CountryCode=w.CountryCode

left join PostalDim p on p.PostalCode=w.PostalCode

left join StateDim s on s.StateName=w.State

left join DateDim d on d.MonitorDate=w.Date

left join CityDim ci on ci.City=w.City

left join DayDim dd on dd.DayDate=day(w.Date)

left join MonthDim md on md.MonthDate=month(w.Date)

left join YearDim yd on yd.YearDate=year(w.Date)

left join QuarterDim qd on qd.QuarterDate= (case when month(w.Date) in (1,2,3) then 1

when month(w.Date) in (4,5,6) then 2

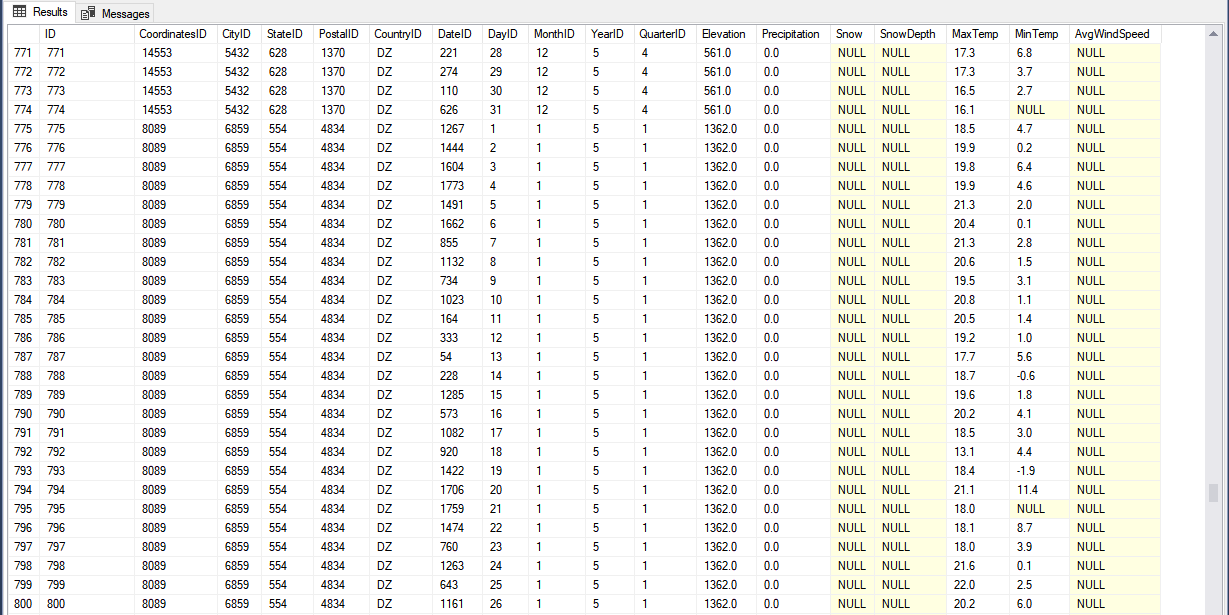
when month(w.Date) in (7,8,9) then 3

when month(w.Date) in (10,11,12) then 4

end)

where w.Country is not null and w.CountryCode is not null

After the above process the fact table looked like that:



So, we had our data in the fact table and the last part was to connect the fact table with the dimension tables. For that we used an alter table statement and added the foreign keys.

ALTER TABLE WeatherReportsFact

ADD FOREIGN KEY (CoordinatesID) REFERENCES CoordinatesDim(ID),

FOREIGN KEY (CityID) REFERENCES CityDim(ID),

FOREIGN KEY (StateID) REFERENCES StateDim(ID),

FOREIGN KEY (PostalID) REFERENCES PostalDim(ID),

FOREIGN KEY (CountryID) REFERENCES CountryDim(CountryCode),

FOREIGN KEY (DateID) REFERENCES DateDim(ID),

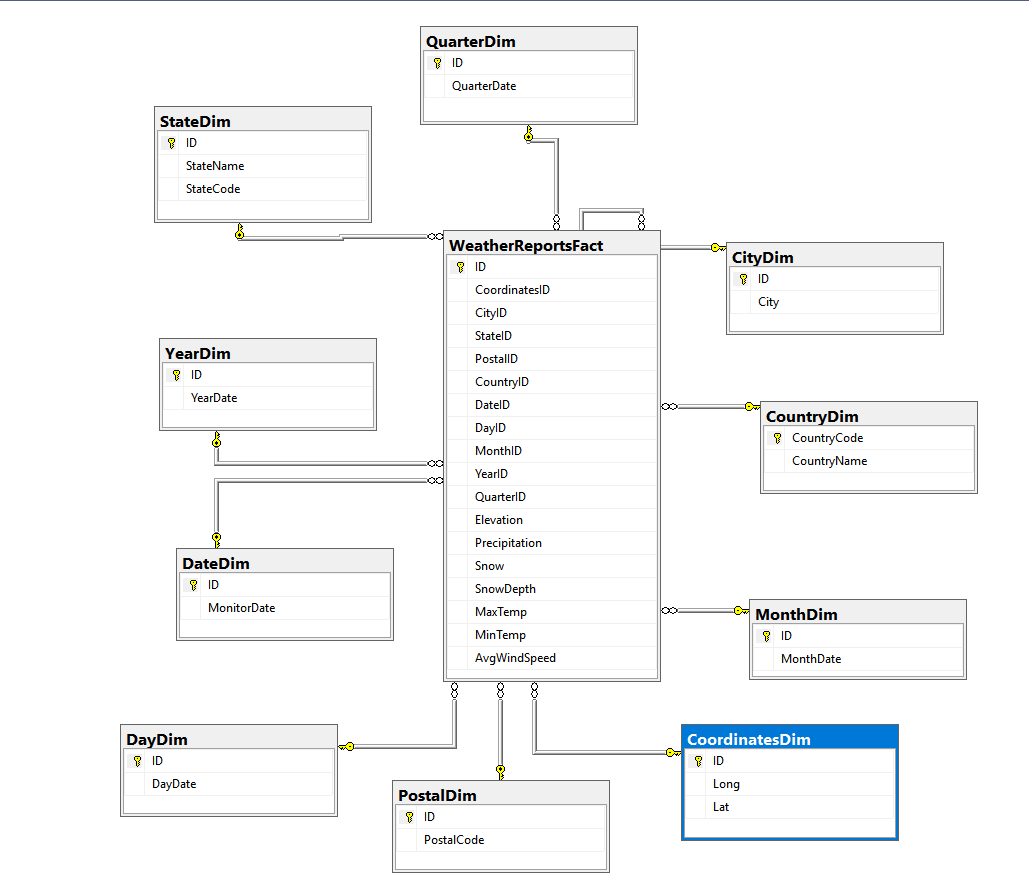
FOREIGN KEY (DayID) REFERENCES DayDim(ID),

FOREIGN KEY (MonthID) REFERENCES MonthDim(ID),

FOREIGN KEY (YearID) REFERENCES YearDim(ID),

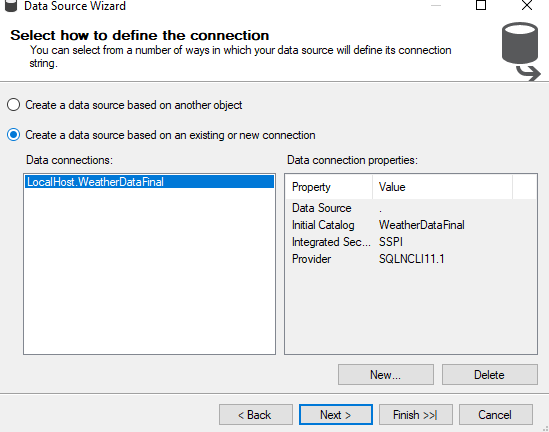
FOREIGN KEY (QuarterID) REFERENCES QuarterDim(ID)

Now we had our fact and dimension tables ready and connected and the star schema was ready to be used for the creation of the cube in the SSAS tool. The final size of database is about 25 GB of information ready for analysis!

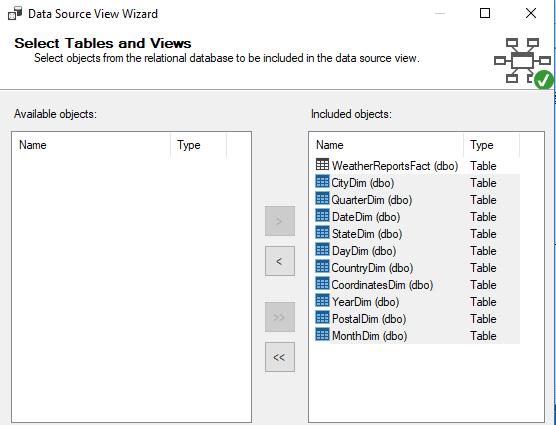
`

# Cube

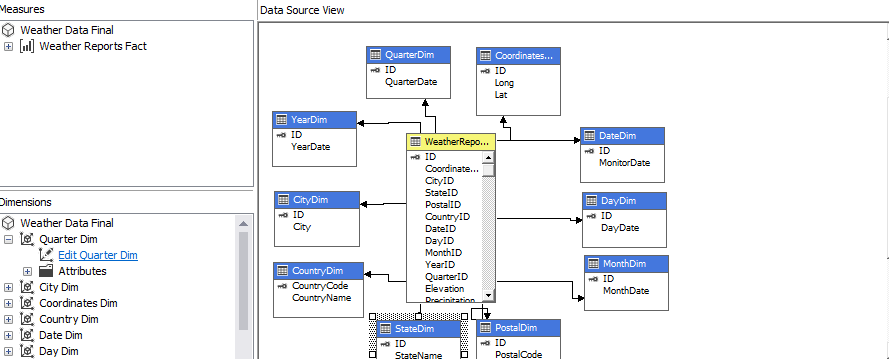
Initially we created the connection between the SSAS and the SQL server as depicted below.



We imported all the tables of the server to create our new cube structure.



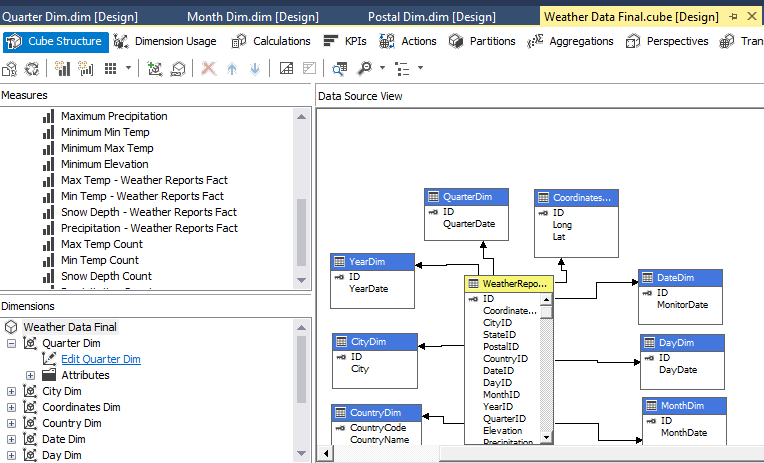
The following screenshot demonstrates the cube structure in the SSAS, which is very similar to the star schema. We decided to import all the data in the tables because at the time we were not absolutely certain of what to expect from the analysis process.



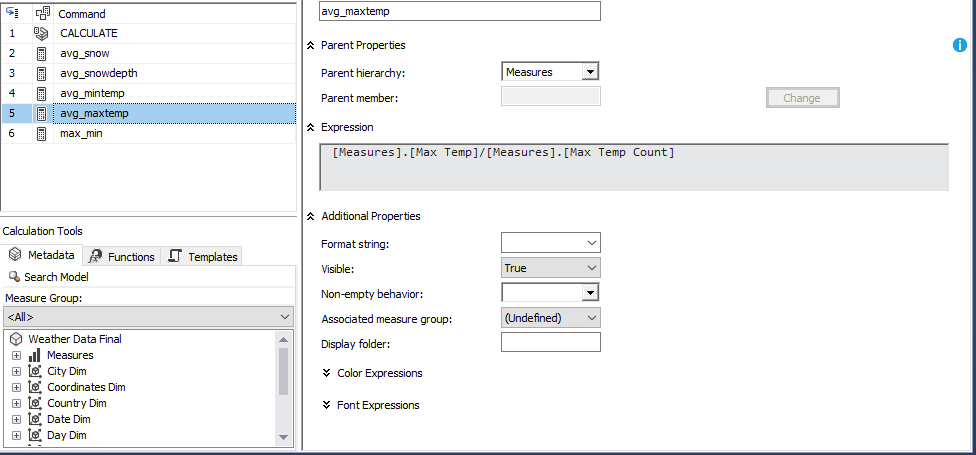
All the data that have to do with dates initially had the type of integers, in the SQL server. However, they were converted as imported to date types, in the SSAS. We already knew that this would also assist us in the visualizations when using the tableau. We also, decided not to create date hierarchies because it was not required in our analysis.

We created some new measures that would be significant in our analysis. We have to explain though that our variables in the original data set were the max and min temperature for each city per date but we did not have the average of all the variables.

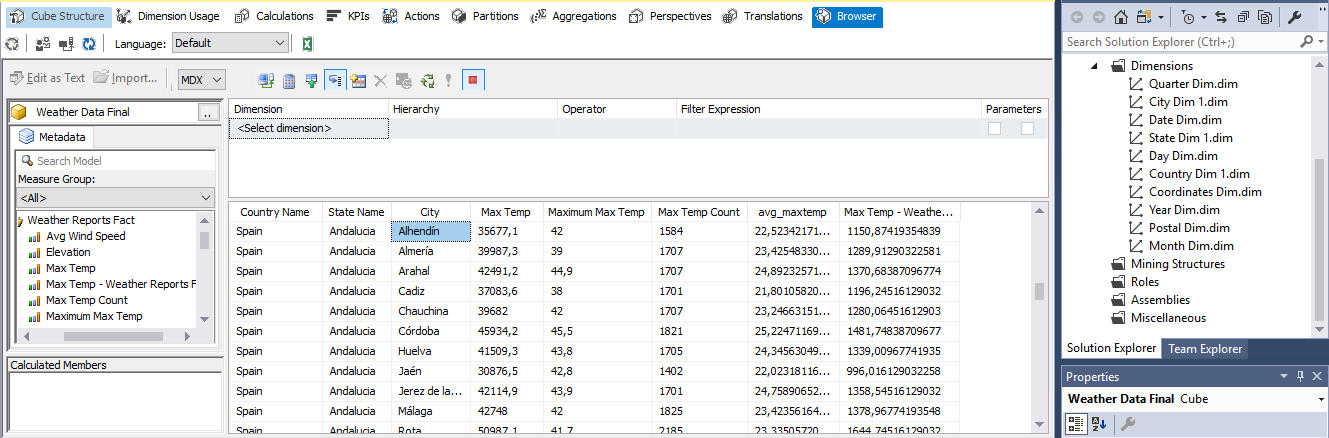
By exploring out data further we found out that there were more than one recordings for some cities during the same day (recorded by different weather stations). What we did was first to calculate the max and the min of all variables and the count of the non-empty values to calculate the average for every date dimension.



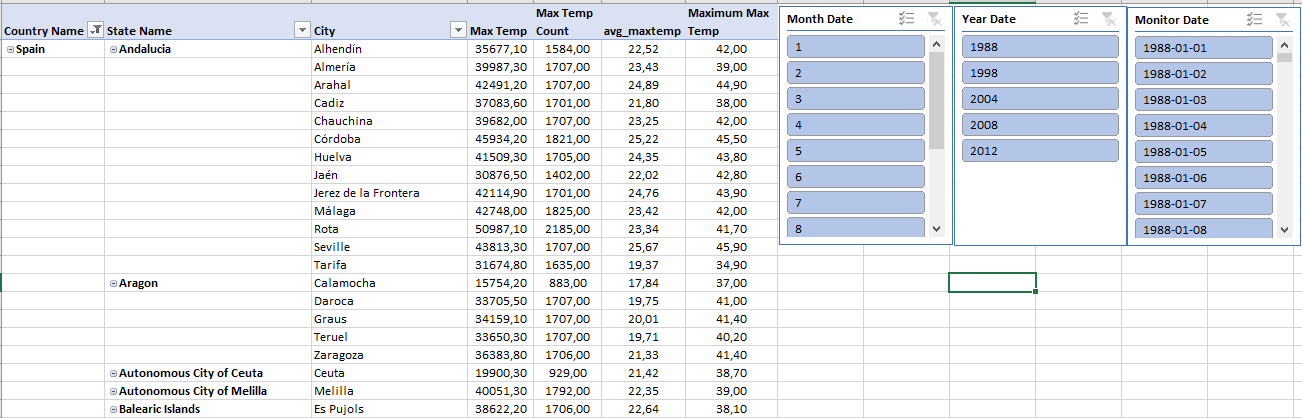
The picture below demonstrates how we calculated all the averages.



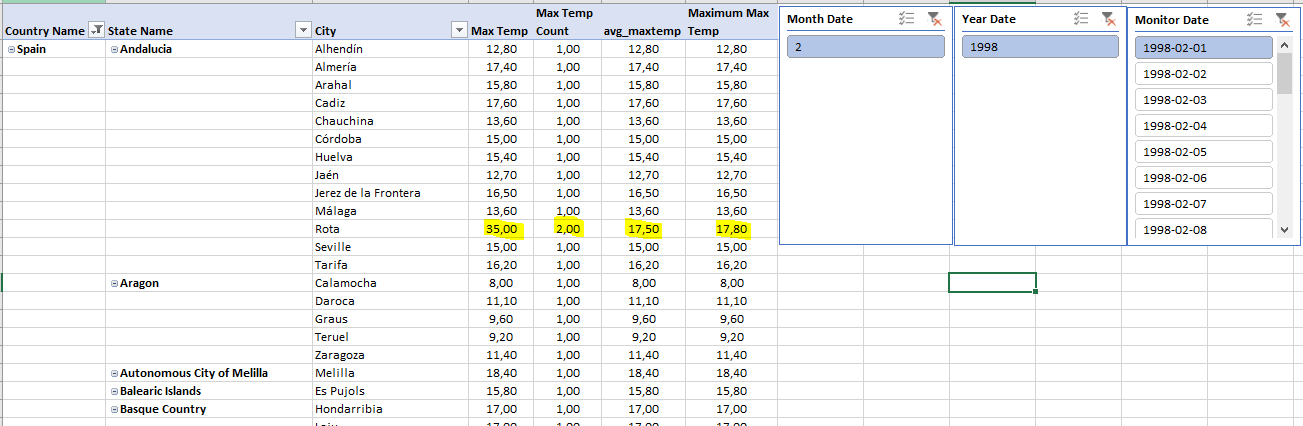
The following screenshot shows the process of browsing the cube in the SSAS and testing the results



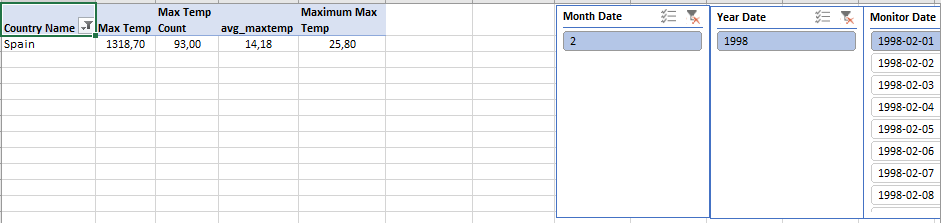
The next step was to browse the cube in the excel and run some extra tests.



The following picture demonstrates a city that was recorded more than once on the same day and the average is still calculated properly. Of course, during this process the slicers were there to assist us with the searching.



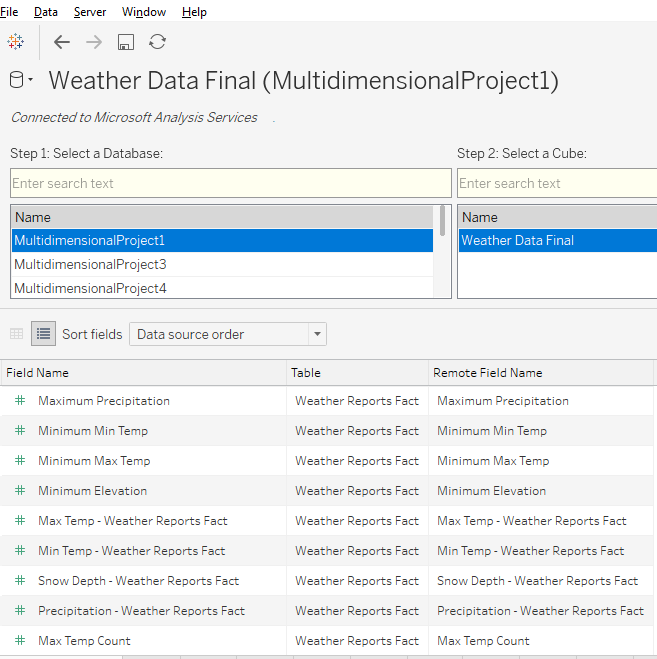
What is more, we run some tests for every date and geographical dimension.



Everything was set for the tableau analysis to take place.

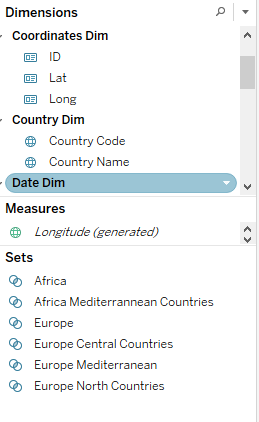
# Visualization

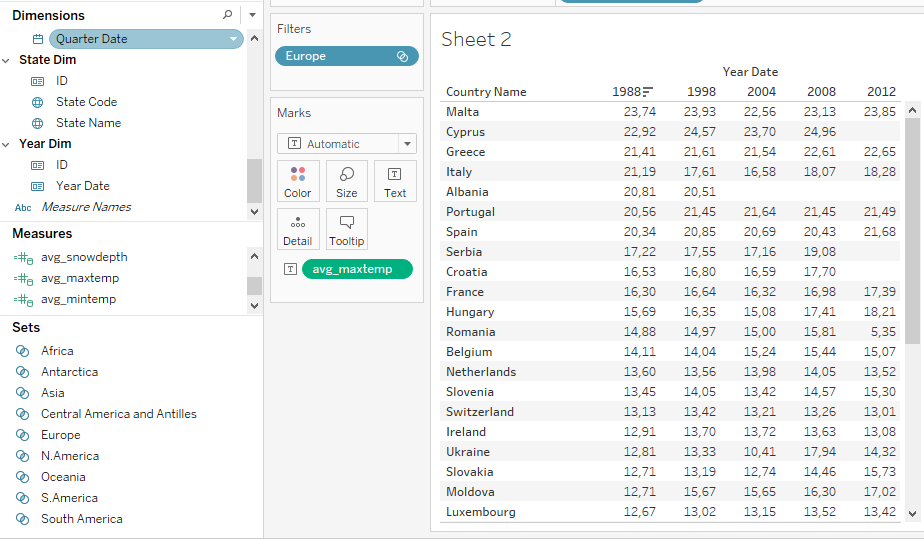
In the picture below is demonstrated how all our measures and dimensions were loaded properly.



We had to decide on the kind of analysis we wanted to perform. An analysis of weather data based only on geographical criteria does not really make sense. Thus, we had to work with sets. That is the reason why we did not create a continent/country/state/city hierarchy in our cube in the first place.

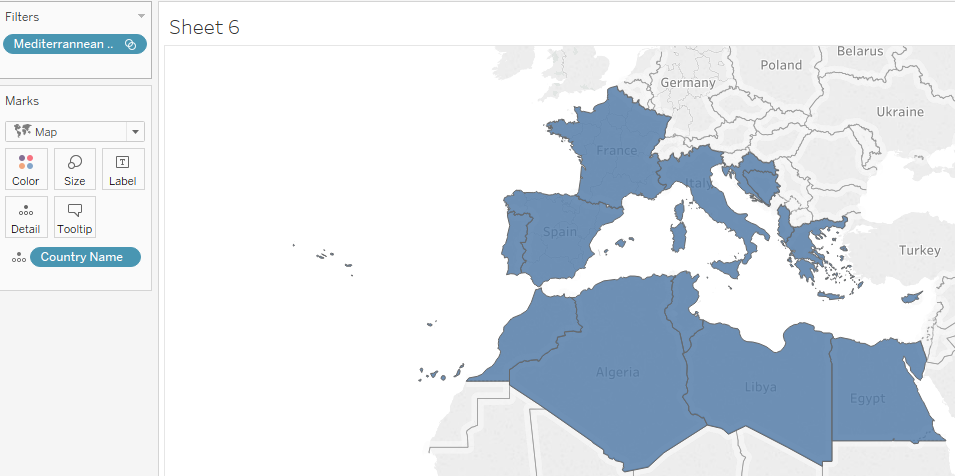
We worked with sets, in our case this means that first we created continents, climate zones and finally when we perceived as necessary we combined the sets to get the appropriate results.



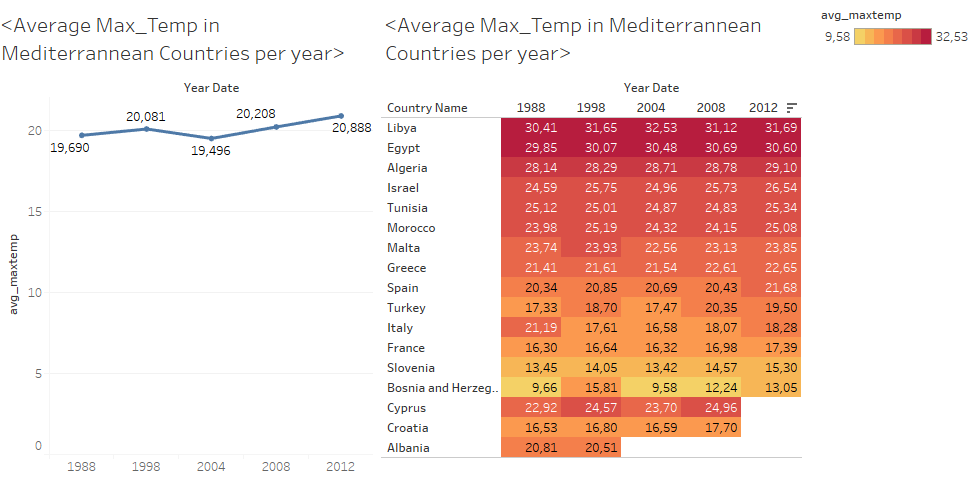


In the two following screenshots, it is shown how we combined the European Mediterranean countries with the African ones to get the Mediterranean countries in total.



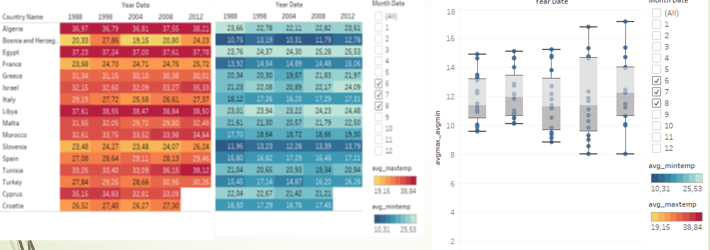


The first step of our analysis was to check if the average max temperature increases throughout the years

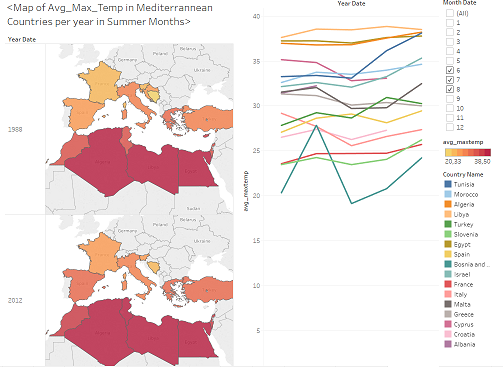


As it is presented in the screenshot above the average max temp in the Mediterranean countries raised from 19,69 degrees of Celsius to 20,88 degrees. The rise of the temperature stands for all the countries except for Italy.

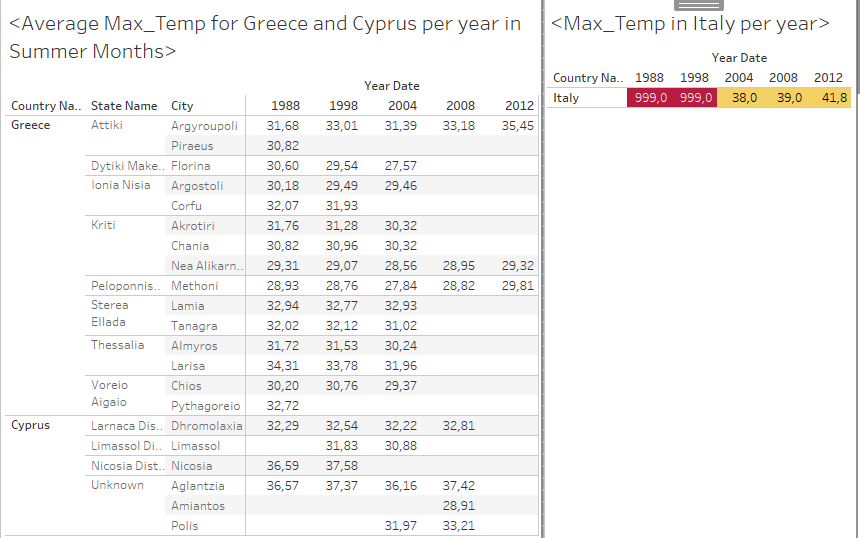
Furthermore, we focused on summer months only (June, July, August) and the results were even more apparent. To be exact, not only the average max temperature is rising but also the average min temperature follows the same trend. In addition, as the box plot represents also the differences between the min and max temperature is increasing throughout the sample years (this was calculated by subtracting the average min temperature form the average max temperature).



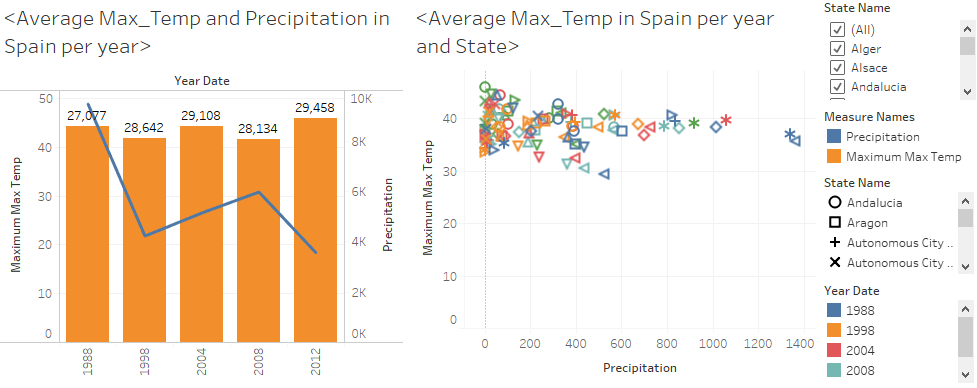
This map below demonstrates that the Mediterranean countries are getting much warmer.



However, Greece, Cyprus and Italy are the exception, which is a bit strange so we decided to search a bit further on what may be causing this.



What we found out is that in Greece and Cyprus there are a lot of missing values specifically in the years of 2008 and 2012 and this obviously affects the average. In addition, there are two really peculiar and probably wrong recordings in Italy for the years of 1988 and 1998. If we take those values out of the equation those countries might follow the same trend as the others. Of course, we followed the same process for every single search to make sure that the conclusions we draw are not biased.



Since we could not draw conclusions based on Cyprus, Italy or Greece, we decided to focus on Spain. We found out that in Spain during the summer months, not only the average max temperature is rising but also the sum of precipitation is reducing dramatically. This is a phenomenon applied to every state (region).

The phenomenon is clearly and most intensely observed in the state of Andalusia. In the screenshot below the rapid and steep changes in both the average max temperature and the sum of precipitation is presented. Specifically, for Andalusia the average temperature is rising while at the same time the precipitation is decreasing to the point of almost reaching zero during the summer months.

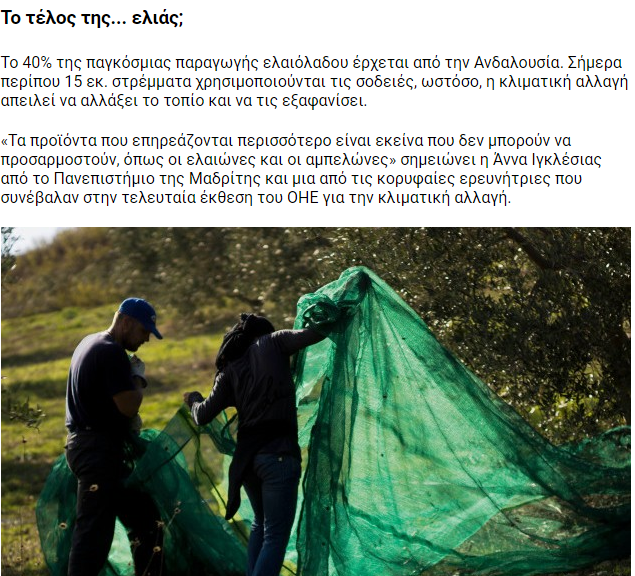
A screenshot of a map

Description generated with very high confidence

In addition, this phenomenon did not improve even during autumn months.

A screenshot of a map

Description generated with very high confidence



So what is the actual effect in real life?

On our way of finding that out, we came across many articles that had to do with problems caused by climate change in Andalusia.

The most important is the one affects the agriculture and specifically the olive oil production.

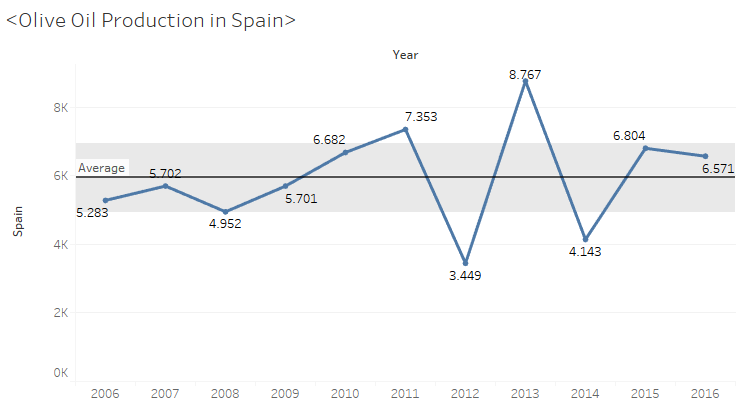
Keep in mind that Andalusia is the largest producer of olive oil in the world, predominating the global market with 40% of market share.

Researchers strongly believe, and our analysis supports that olive oil trees in Andalusia are threatened by extinction because of climate change.

Of course, we decided to search for some extra data that could prove that there is a relationship between climate change and olive oil production in Spain and found some (small) data in Eurostat that could help us for deeper analysis.

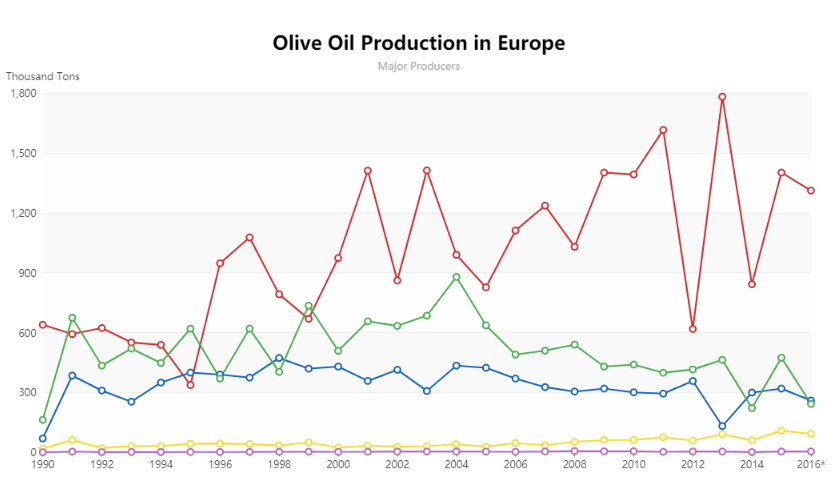


The data contained the olive oil production in European Countries and after connecting this data to the Tableau, the results were more than obvious.



According to this chart, it comes without saying that there is a strong connection between weather change and olive oil production in Spain. In 2012 the olive oil production hit its lowest level during the last decade and according to our analysis before this was the warmest and most dry season among all the examined years.

Furthermore, we found out that in 2012 there was a raise in olive oil production (comparing to the previous years) in the other 2 major producer countries which are Greece and Italy but unfortunately our initial data was not valid for further analysis for that period.



# Conclusions

In this report, we analysed the weather data. We mainly focused on weather changes in Mediterranean countries for the years of 1988, 1998, 2004 ,2008 and 2012. We checked if changes in weather do really occur, and if they do, if they have an effect on people’s lives. We observed that the temperature in the region of Mediterranean is rising. In order, however, for the analysis to make more sense, we focused on one country: Spain and particularly on a state (region) of this country, Andalusia. In this particular region, we noticed that there is a rise in the temperature and at the same time a drop in precipitation. By searching a little bit more on the issue of Andalusia we came up with the following.

Andalusia is the biggest olive oil producer in the world. What is more, the olive trees are affected by weather changes. During the years that we performed our analysis, we found out that the olive oil production in Andalusia was affected negatively. Andalusia’s main income comes from exporting olive oil and many habitants’ main income comes from this production. Thus, we conclude that weather changes do really in the long-term effect the climate and eventually change it.

As we can see from the Andalusian example, the changes in weather that in the long run have an effect on the climate actually do affect people’s lives.