

Documentation

Project made by: Ignacio Donderis, Pedro Torrijos and Miguel Santiago

[Introduction](#)

[Definitions and theory](#)

[First Steps](#)

[Steps in GitHub](#)

[SQL server installation and configuration](#)

[Entity relationship map](#)

[Mockup](#)

[Eda of the AdventureWorks data \(Person\)](#)

[Final Dashboard](#)

Introduction

Throughout this documentation document we are going to explain the process by which we develop the dashboard of the person section of the AdventureWorks data set. Starting with the creation of a GitHub repository where we would upload everything necessary for the EDA of the data, the data sets and the necessary codes and finally the creation of the Dashboard in the Power BI tool, we are also going to develop the problems with which We have met and how we have solved them.

Definitions and theory

Before starting to create the repository, we had to understand certain fundamental theory along with definitions and commands which we were going to use when getting everything up and running.

Git checkout, this command switches you to a different branch within your local repository. It does not directly access remote repositories.

Git merge is often used in conjunction with git checkout to select the current branch and then merge them together.

Git push allows you to push your commits from your local branch of your local git repository to the remote repository.

Git pull is better than fetch and merge, its more used the git pull, because git pull it's a mix of the other two.

Git stash takes uncommitted changes, both those that are ready and those that are not, saves them aside for later use and then undoes them in the code you are working on. You can use stash pop and stash apply.

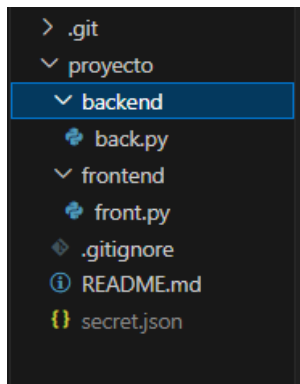
Git rebase The main reason for using it is to maintain a linear Project history, that is, to see how the main branch has progressed since it was started.

Git fetch allows you to update your local repository with the latest changes from a remote repository, but without merging those changes into your local working branches

First Steps

Once we had clear concepts and commands, we began to create branches and "mess around" with GitHub to learn how to: create branches, clone the branches, merge, download information from the repository, upload data...

- We have a main master branch, with the files, folders, etc...



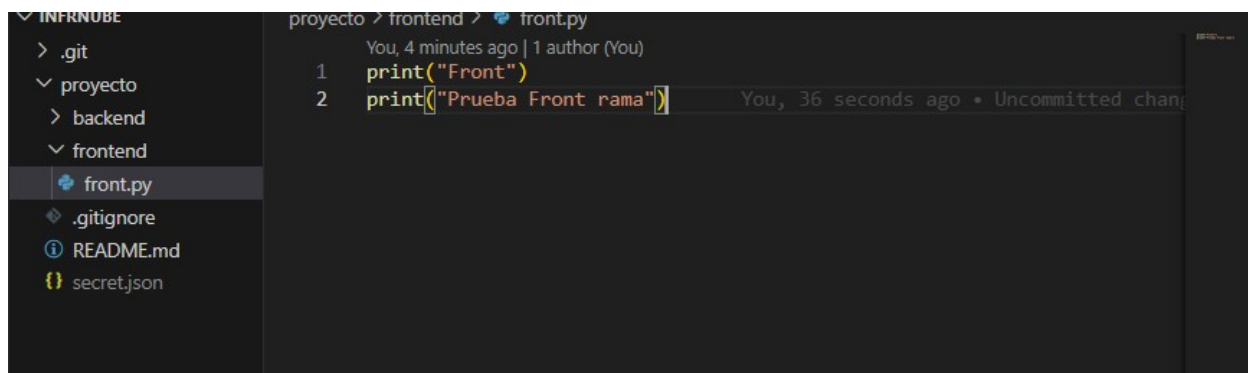
- We push to send all the files to the main branch from the repository.

>git push test3 (Repository) master (Main branch)

- We create a new branch, we modify files or create new ones.


```
(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git checkout -b new-test
Switched to a new branch 'new-test'


(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git branch
master
* new-test
prueba2
test2
```





- First, we add the changes and then push to upload them in the new branch created from the repository.


```
(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git add .  
  
(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git status  
On branch new-test  
Your branch is up to date with 'test3/new-test'.  
  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
        modified:   proyecto/frontend/front.py
```


 **new-test** had recent pushes 6 minutes ago [Compare & pull request](#)

 new-test ▾


 2 Branches



 0 Tags


 Add file ▾

 Code ▾

This branch is 1 commit ahead of **master**.

 Contribute ▾

 **filip** Prueba rama new-test 02030f6 · 7 minutes ago  5 Commits

 proyecto

Prueba rama new-test

7 minutes ago

- We switch to the main branch and pull to have the original files and avoid conflicts.

>git checkout master

- While we are in the main branch, we merge the second branch to main one.

```
(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git pull test3 master
From https://github.com/Filips122/test3
* branch          master      -> FETCH_HEAD
Already up to date.

(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git merge new-test
Updating 251d992..02030f6
Fast-forward
 proyecto/frontend/front.py | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

(GIT_UE) C:\Users\xfeli\Desktop\InfrNube>git push test3 master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Filips122/test3.git
 251d992..02030f6  master -> master
```

- After we merge both branches we make a push to send it to repository.
- If we use fetch it would be like merge but instead of branches, between repositories and it is a good practice if you don't want to "mess it up" in the main repository.

>**git fetch** test3

- This is to see the difference between actual branch and the repository you want to merge at.

>**git diff** master (Actual branch) test3/master (Repository that you want to merge)

- We do rebase if we want to have a cleaner history and not with all the changes as in merge.

>**git checkout** new-test

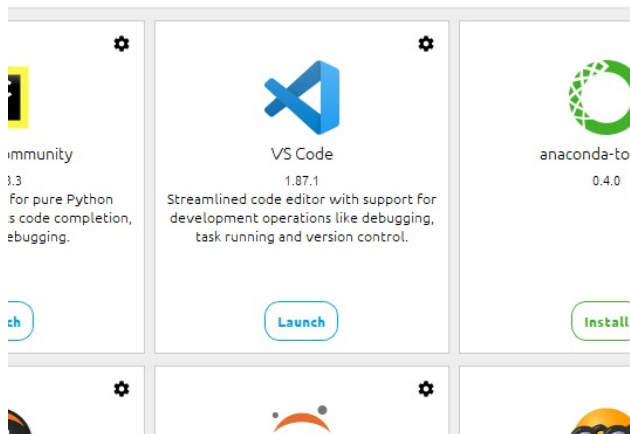
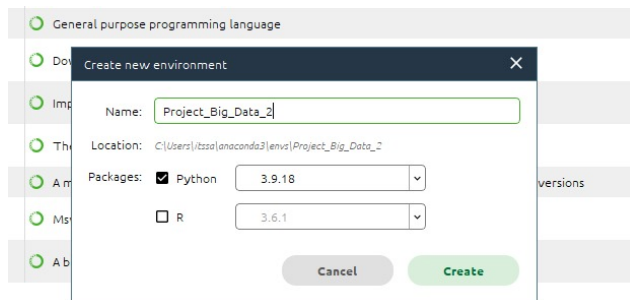
>**git rebase** main

>**git rebase** --abort

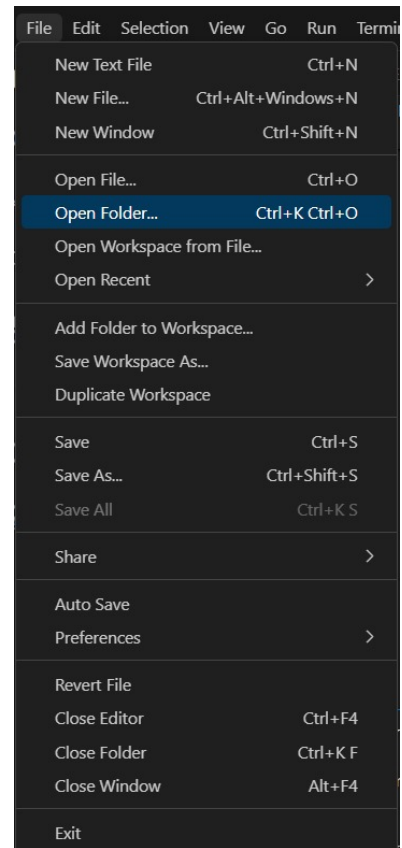
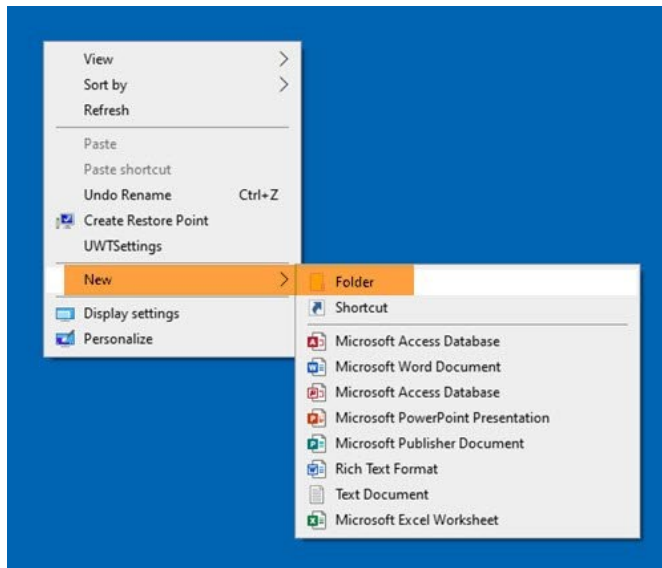
Steps in GitHub

In this section we created a github repository, in this case we cloned one to have certain files that interested us. On this we do certain steps to test the github tool. See below.

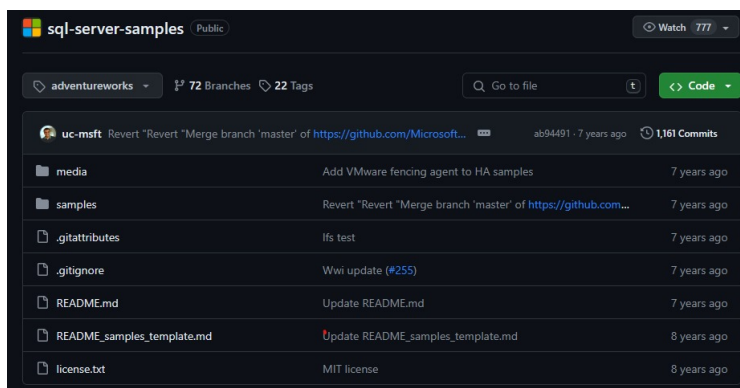
- We **prepare anaconda** for the entire process.

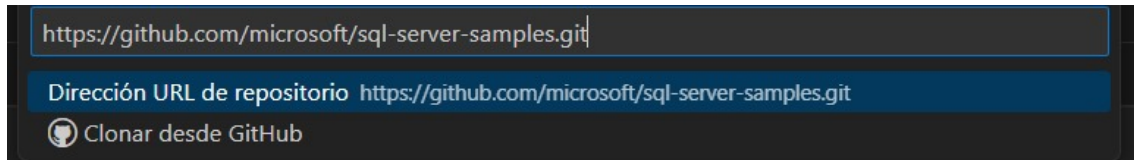
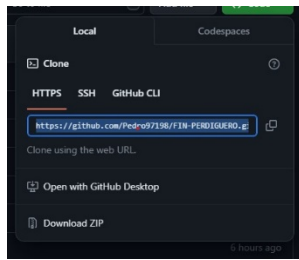


- We **create the folder** where we are going to host the git repository locally and **open with VS code**.

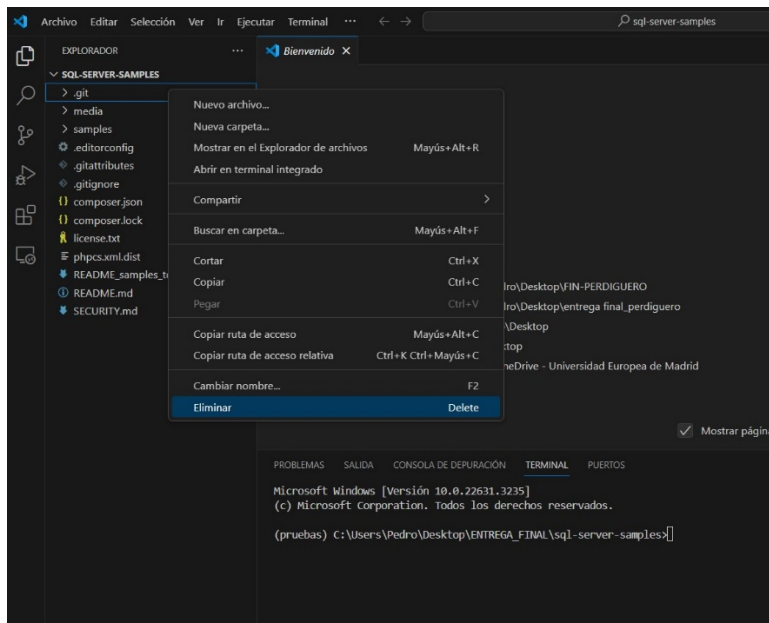


- First of all you cloned a repository with the **git clone** command.





- We **close the remote connection** with the git-hub repository.



- We execute the **git init** command.

```
(pruebas) C:\Users\Pedro\Desktop\ENTREGA_FINAL\sql-server-samples>git init
Initialized empty Git repository in C:/Users/Pedro/Desktop/ENTREGA_FINAL/sql-server-samples/.git/
```

- We **created a github repository**.

- We execute the command **git remote add repository url**.

```
(pruebas) C:\Users\Pedro\Desktop\ENTREGA_FINAL\sql-server-samples>git remote add origin https://github.com/Pedro97198/pruebas-a.git
```

- We do a **git add**.

```
(pruebas) C:\Users\Pedro\Desktop\ENTREGA_FINAL\sql-server-samples>git add .
```

- We do the **commit** and finally the **push**.

```
(pruebas) C:\Users\Pedro\Desktop\ENTREGA_FINAL\sql-server-samples>git commit -m "subida de archivos"
```

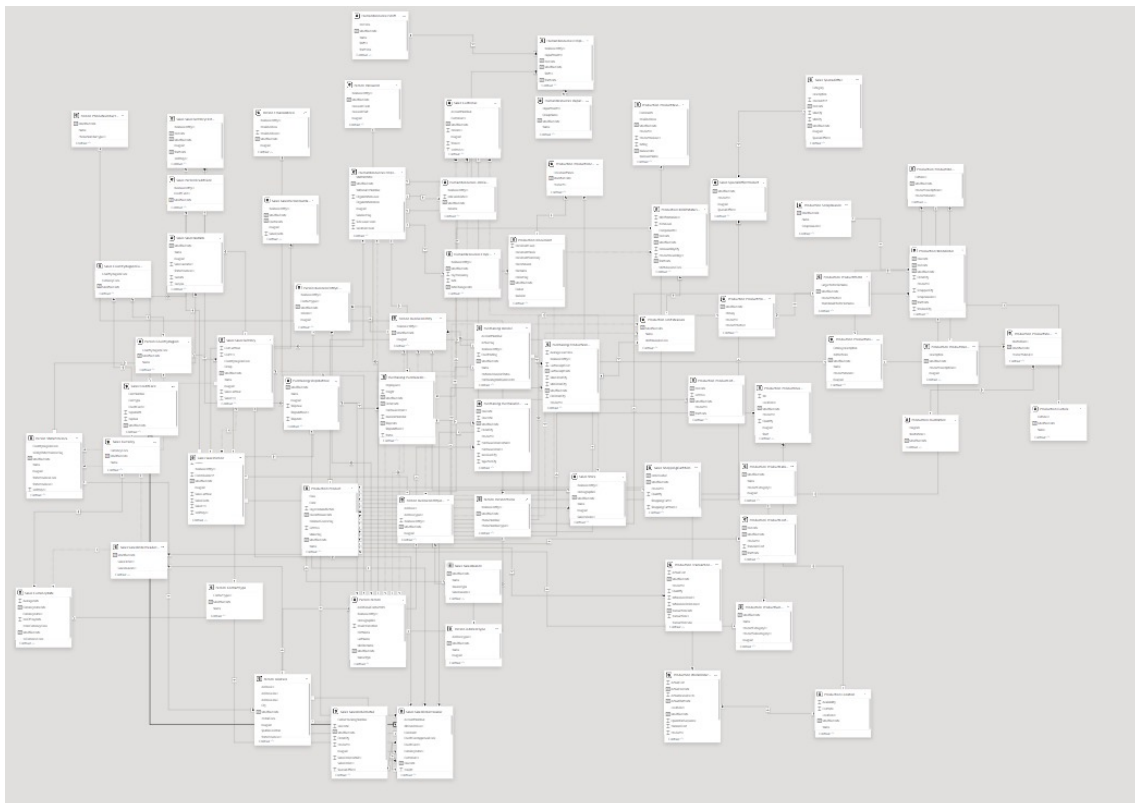
```
(pruebas) C:\Users\Pedro\Desktop\ENTREGA_FINAL\sql-server-samples>git push -u origin master
```

- **Clean** of the data: We proceed to eliminate the items and folders that we are not going to use, and understand the rest of the data.

- We've recently set up an SQL server to manage our data, and as part of this process, we've incorporated the AdventureWorks database. This involved placing the AdventureWorks.bak file inside our local SQL server for easy access and management. [\(Explained in the next section\).](#)

With the SQL server up and running, we seamlessly connected it to Power BI, leveraging its powerful data visualization capabilities. Using Power BI, we imported all the data from AdventureWorks, allowing us to delve deep into its insights and metrics. [\(Explained in the next section\).](#)

After importing the data, we took it a step further by creating an entity-relationship diagram. This diagram visually represents the relationships between different entities (such as tables) in the AdventureWorks database. It's a crucial step in understanding the data model and how various pieces of information are connected.

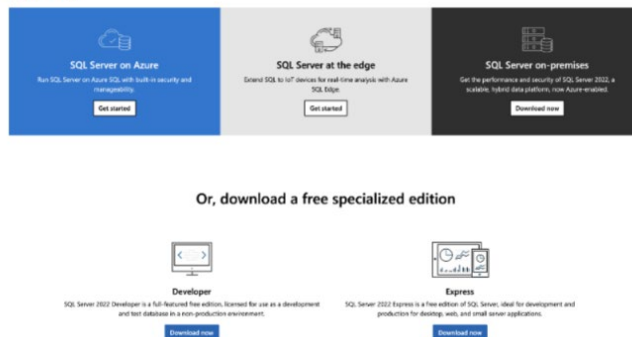


SQL server installation and configuration

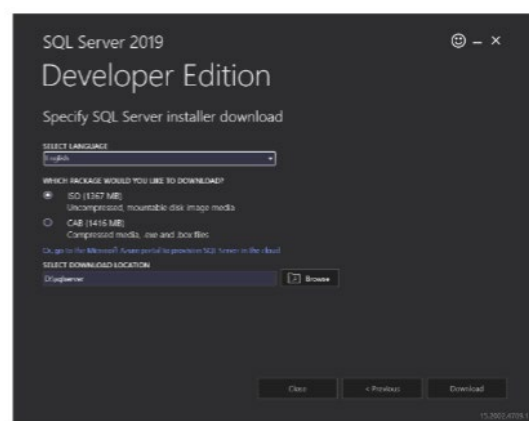
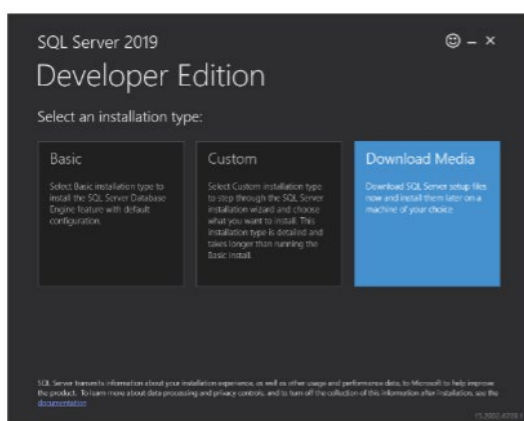
In order to get all the data and download it from Power Bi, we have to extract the csv from the adventure Works .bak. To do this, the most correct thing is to create a local server within our computer with SQL and upload the file there. To do this we follow these steps:

1. We enter the Microsoft SQL Server page ([sql server de Microsoft](#)) and click the third on-premise option to download.

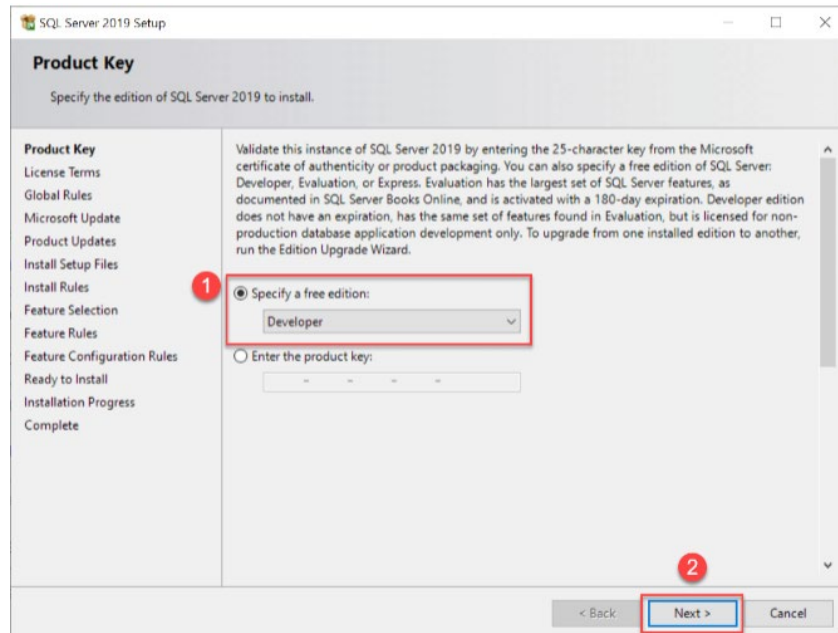
SQL Server Try SQL Server on-premises or in the cloud



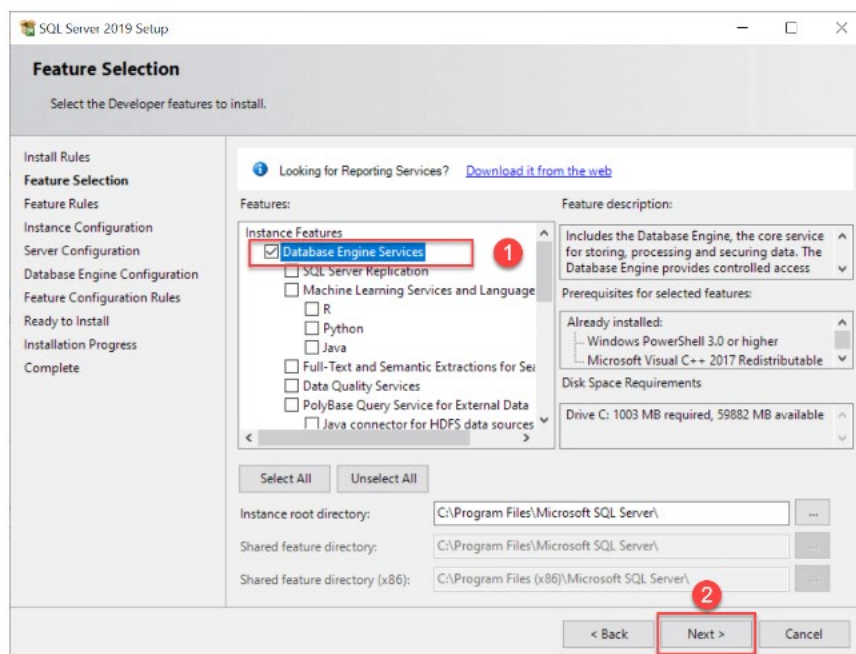
2. When entering the installer we must select the third download media option, within it, select our language and the ISO installation format.



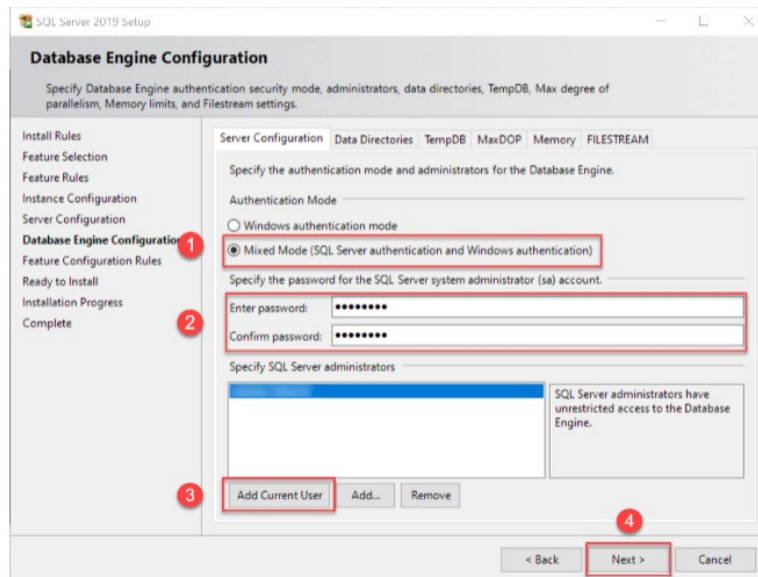
- When the download is finished, the system installation configuration section will open. In this first step we will have to select the free edition option as developer in the product key. Then, accept the conditions and licenses and continue.



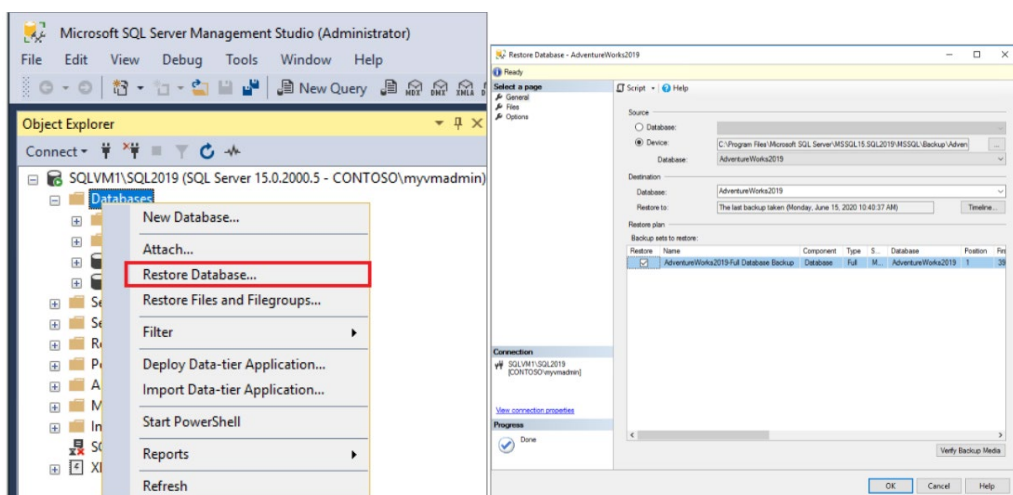
- In the feature selection section we must select the first option that appears called database engine services.



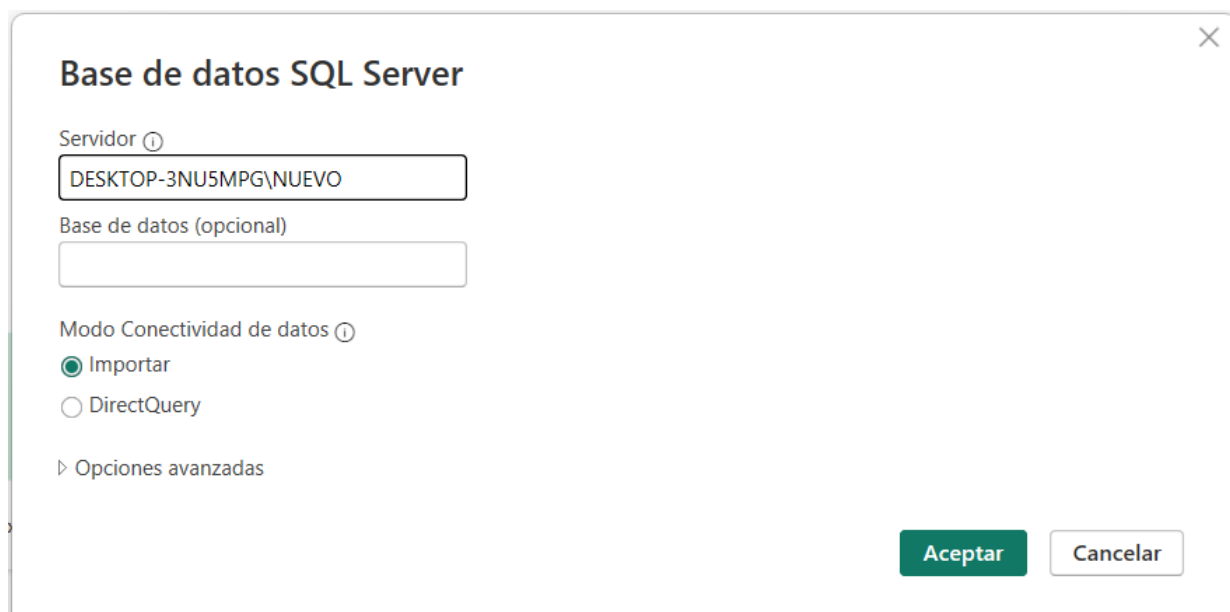
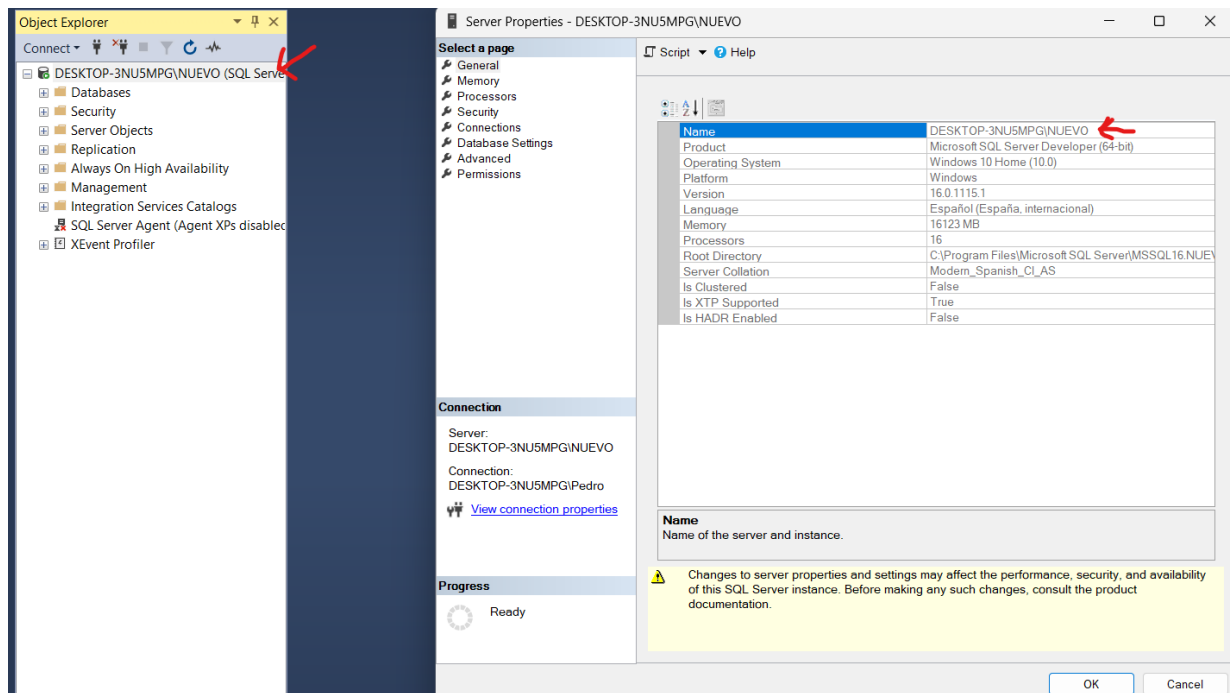
5. Finally we enter the database engine configuration part in which we have to select the second option in the authentication section; mixed mode. Afterwards, we have to set a password to our SQL server and finally add the current user and click continue.



6. We already have configured what the server will be, but now we need an application to be able to enter the server and manage it to our liking, for this we will install the [sql server – management studio](#).
7. On the internet we now look for the official Microsoft Adventure Works page to download the file [AdventureWorks.bak](#)
8. Let's open the SQL Management Studio application, here we right click on the databases folder that appears and restore databases. We do this to now add the adventure Works file that we have downloaded.

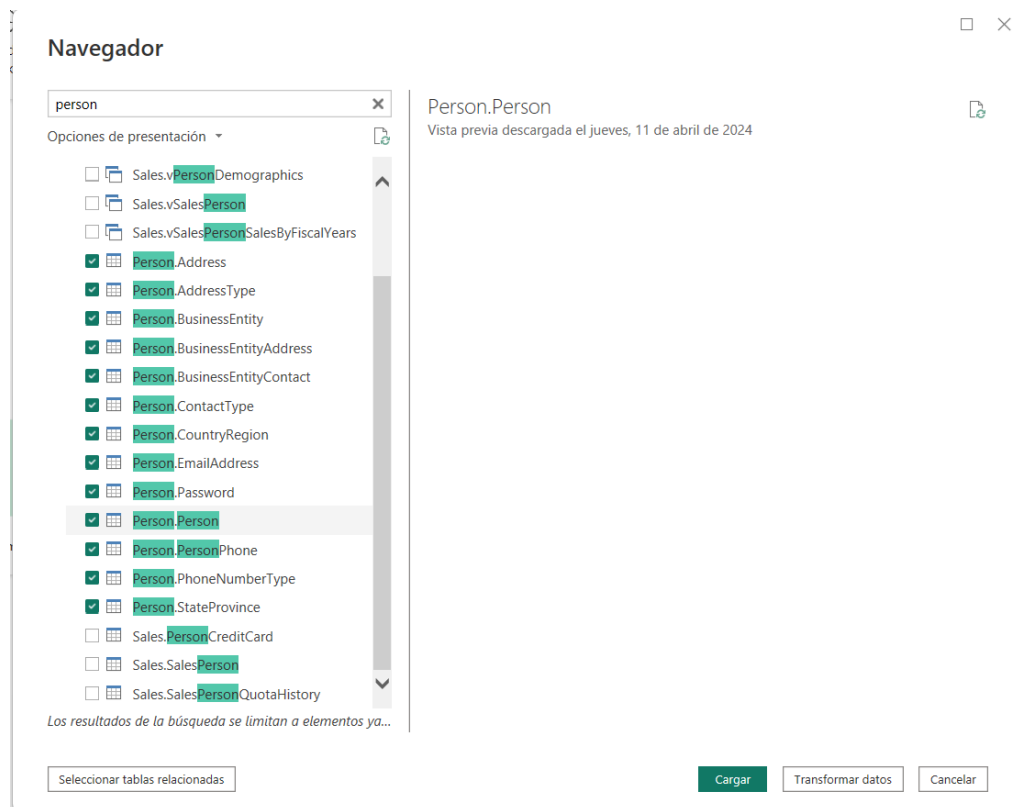


1. We open power Bi and click to obtain the data, here we have to find the sql server database option. Here we have to enter the name of the server to be able to enter. To do this, within SQL Server management studio we have to enter properties in the first option and click properties, here in the first option we have the name of the server. Once you click OK, we will see the Adventure Works folder and all the csv files from there. We select the ones we wanted, which in our case were Person's files, and we click to load them.

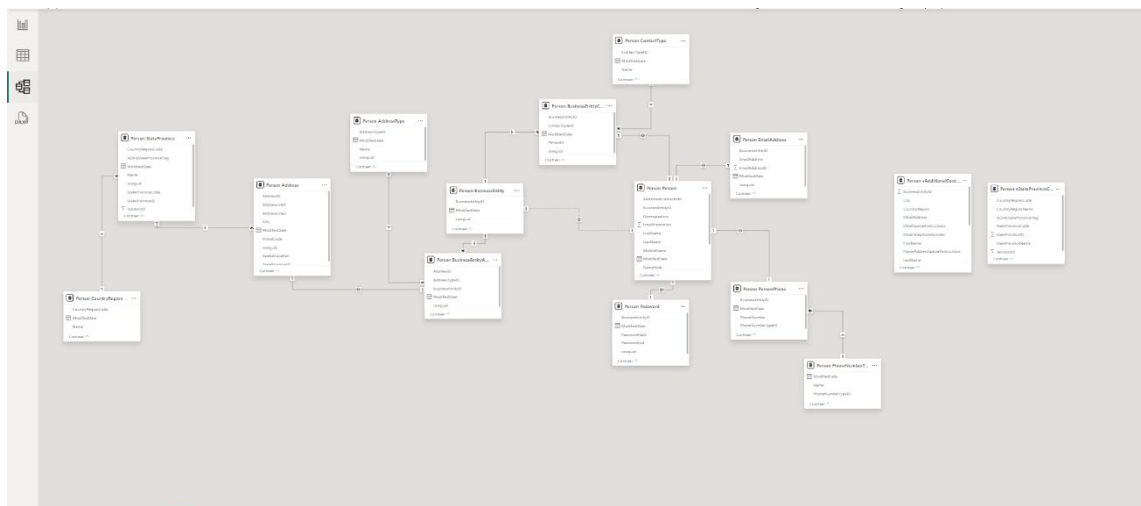


Entity relationship map

1. First, to obtain the entity map of our data we have to select our data. To do this, as we have explained before, we open, obtain all the data from the SQL server and select our data, which is that of the Person department, and click to load.



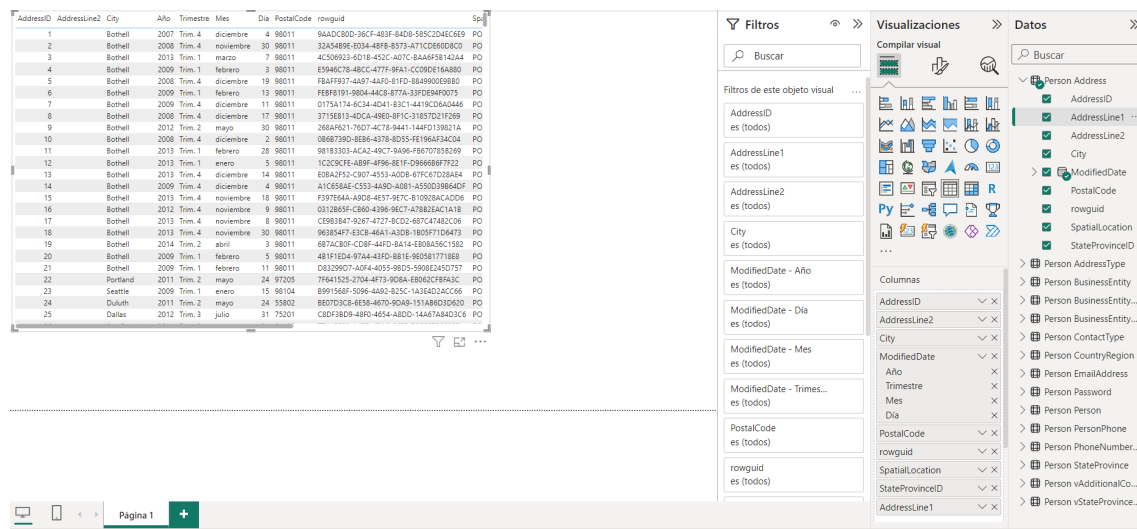
2. When loading the data, on the left hand side we have 4 options, we have to select the 3rd one which is called model view. And we will be able to observe the relationship map of our data to then be able to do the EDA better.



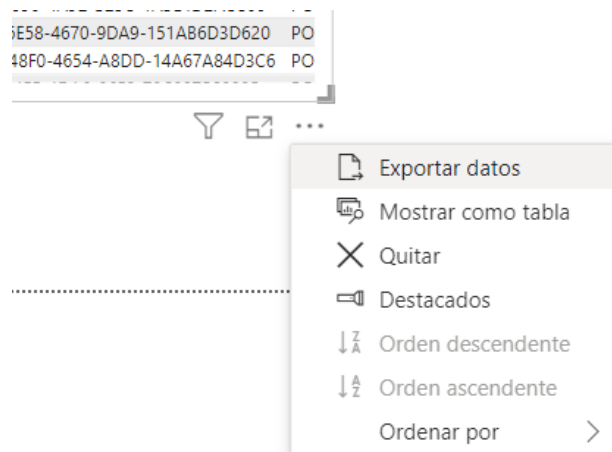
DOWNLOAD OF THE DATA

Now that we have our data inside power bi, we want to download the data to our computer to be able to do the EDA, to be able to download the 15 csv of our data we have to follow these simple steps:

1. On a blank page we put an empty table, in the columns field we have to add the columns from a csv until we have all the ones we want.



2. When we already have the table as in the photo and we have seen that we have all the columns we need, we have to click on the three points in the lower right corner and click on the first option, export data and save it in the folder .



Exportar datos

Mostrar como tabla

Quitar

Destacados

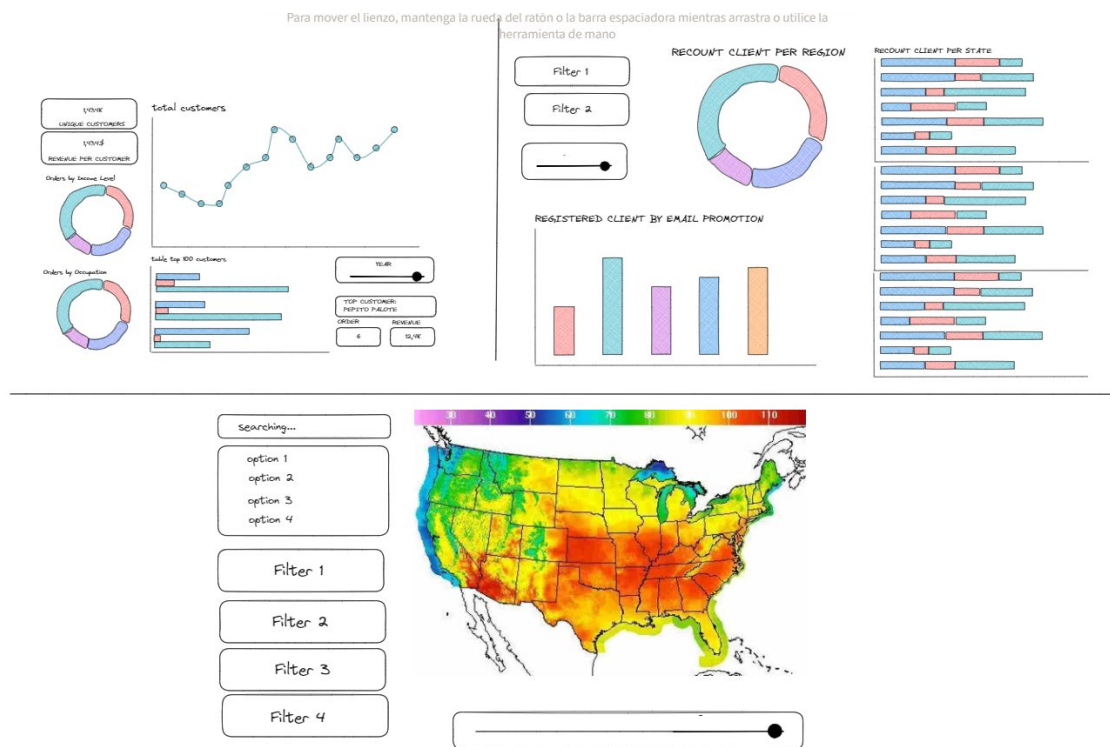
Orden descendente

Orden ascendente

Ordenar por >

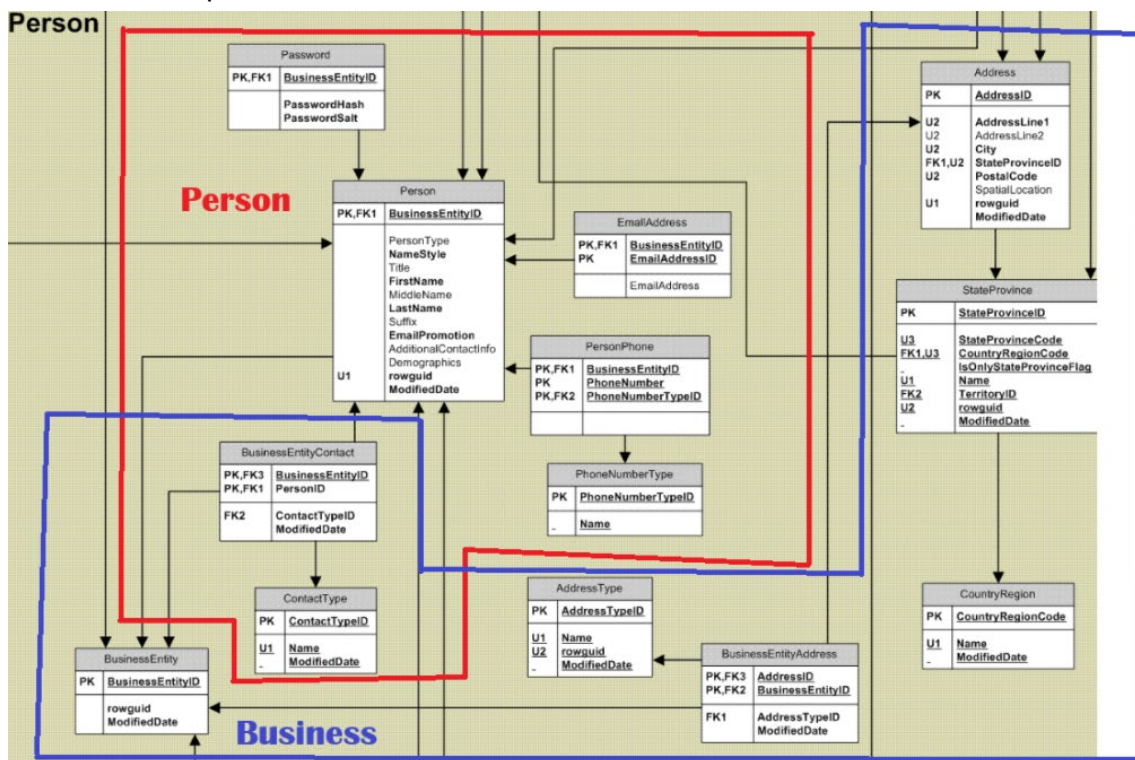
Mockup

Once we understood the data, we decided to make a mockup so that when starting with the dashboard it would be easier for us. We decided to make three "slides" each focused on a topic, although later we understood that with the data we had this was going to be more complicated, since it was scarce.



EDA of the AdventureWorks data (Person)

1. First of all, we are going to work with all the tables that we have because in our case it was not necessary to collaborate with other groups, since we are a single department formed only by our group.
2. By making joins of the different tables that the Person department has, we have finally been left with one main table called Person, it contains the data of all our person. Secondly we have the data Business Entity that contains data of persons that are Business and all its data.



3.
 - These first three questions belong to Address with the help of Person:
 - o What are the regions/states/cities with the highest concentration of clients?
 - o You can create an interactive map that shows the location of customers.
 - o Can you identify trends in customer demographics over time?

- These last two are from Business.Entity with the help of Person:
 - What is the distribution of employees by department, position or hierarchical level?
 - Can you identify different customer segments based on their demographic and geographic characteristics?

We also deliver the. ipynb in which we discuss with our EDA the process which we have followed to make the joins (This is how the procedure by which we have chosen our tables is understood). And eliminate the data which we did not need.

To work well with the data we also deleted the rows that contained more than one null, since when working with Power Bi it only brought problems.

Final Dashboard

We have different versions of the dashboard, to be exact 4, in each of them we were improving different aspects of the dashboard, in the GitHub there is the version history of this.

We have the frontpage in which we can navigate to the rest of the pages and see information that may be relevant.

Then let's create a map which shows us the people distributed throughout the world.

We have a page dedicated to customer data and another to our employees.

Afterwards we have a section on salts and a table with relevant data, all of this is developed in the presentation.