SAMUEL PATINO LUCUMI
JUAN FELIPE ZAMBRANO BAEZA
ANDRES ALBERTO ENRIQUEZ DOMINGUEZ

Universidad Autónoma de Occidente

Profesor: Javier Vergara

ETL

PROJECT ETL

link of the database:
https://www.kaggle.com/datasets/nathaniellybrand/chicago-car-crash-dataset

summary : this data, released on Data.gov that is The Home of the U.S. Government's Open Data Here you will find data, tools, and resources to conduct research, develop web and mobile applications, design data visualizations, and more.
contains all car crash data from 2001+ which happened in Chicago. The dataset contains the date of crash, the speed limit, the weather conditions, the injuries,the latitude and longitude and much more information about the crash in the city.

we as a group chose this dataset because we could see that it was a very large data set which contained a lot of information.

which is more than 700,000 rows and 42 columns.this data that this dataset contains is from 2001 to June 2023 of all accidents reported in the city of Chicago which is one of the largest cities in the United States.

for this reason it was also decided to work in a group of 2 people since it was a very large dataset and it would be very tiring to be done by one person.and between the group we can decide how we will approach the data set because it contains a lot of information and we can perform different types of transformation with it and also because it contains some null values and undelivered information.

2-.
-Street direction
-Street name
-injuries total
-injuries fatal
-injuries incapacitating
-injuries non incapacitating
After carrying out the analysis of the data set and all its columns and rows as a group, we
have decided that we will use the following columns for our data extraction, transformation and loading project:

-Weather condition
-Crash type

-Street no

-injuries reported not evident
-injuries unknown
-Crash hour
-Crash day of week
-Crash month
-Latitude and longitude
-location

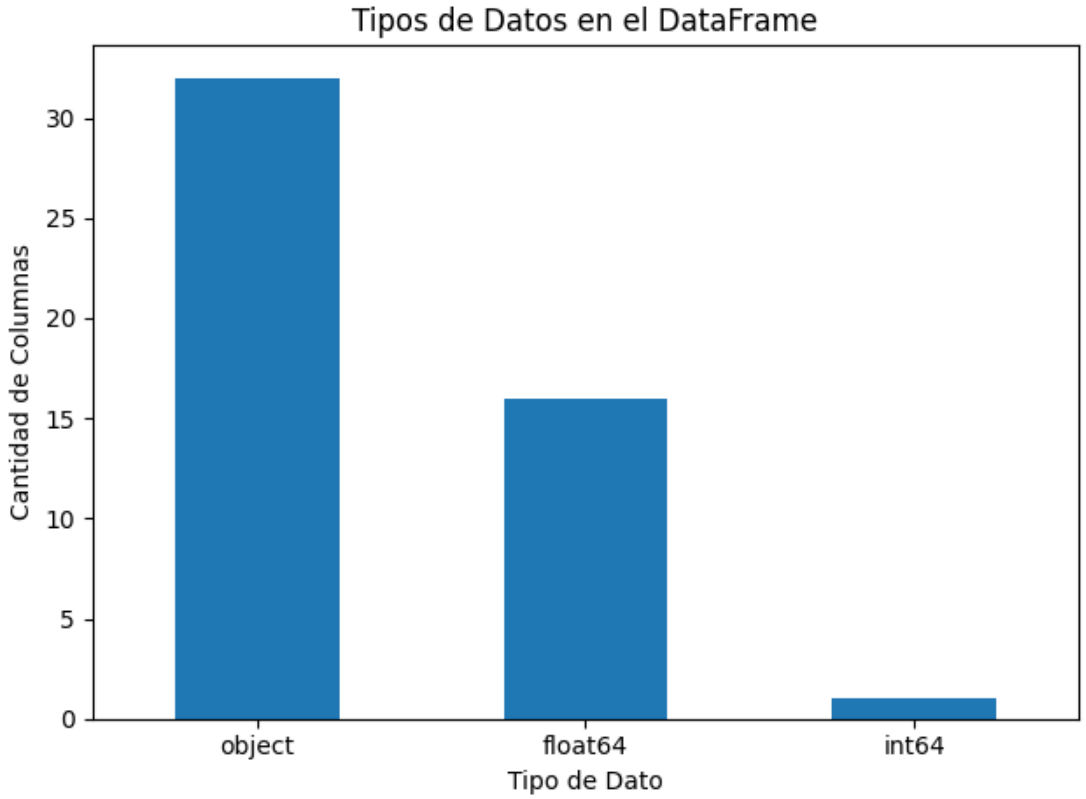
We think that each of these is important since they are the main factors by which an accident occurs. We must bear in mind that each one is different and plays a role when a person drives a car, in this case in Chicago, which normally has a humid climate. The speed depends on each person and his own criteria when driving, so it is a variable that changes a lot. Injuries and deaths are a variable that occurs as a consequence at the time of an accident, it is a variable that we believe is important to analyze. Finally, the time and the days are protagonists since people drive more in specific hours and on specific days.

The main objective is analyze the relationship between climate conditions, driving speed, injuries and deaths, time of occurrence, and geographical locations in Chicago road accidents, with the goal of identifying patterns, trends, and potential influencing factors to enhance road safety and accident prevention strategies

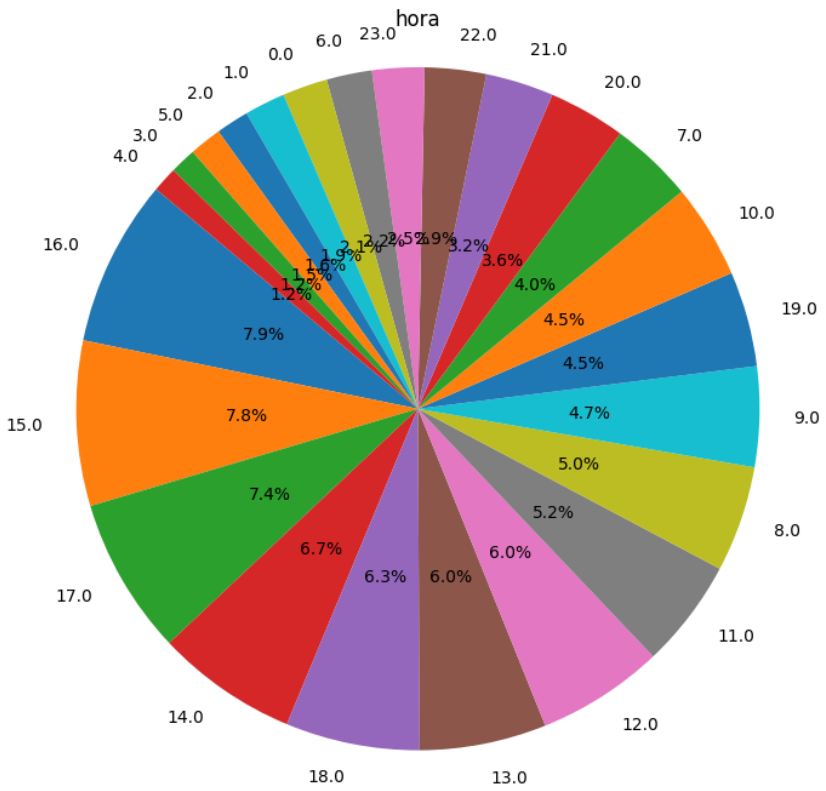
-columns to be eliminated and why
-lighting condition:we as a group and with the objective that we have visualized, we do not believe it is convenient to include this column since it does not contribute to us.
-first crash type:this column has been eliminated as we do not believe it is convenient to know the place where the vehicle was impacted
-trafficway type:this column has been eliminated
-lane cnt:this column was eliminated since most of its data is null
-alignment::we as a group and with the objective that we have visualized, we do not believe it is convenient to include this column since it does not contribute to us.
-roadway surface cond this column was deleted as this column's data type is object
-road defect:this column was deleted as this column's data type is object
-intersection related i:this column was eliminated since most of its data is null
-not right of way:this column was eliminated since most of its data is null
-hit and run:this column was eliminated since most of its data is null
-date police notified we eliminate since knowing what day the police were notified about the accident is not part of our objective to predict some patterns.
-prim contributory cause::we as a group and with the objective that we have visualized, we do not believe it is convenient to include this column since it does not contribute to us.

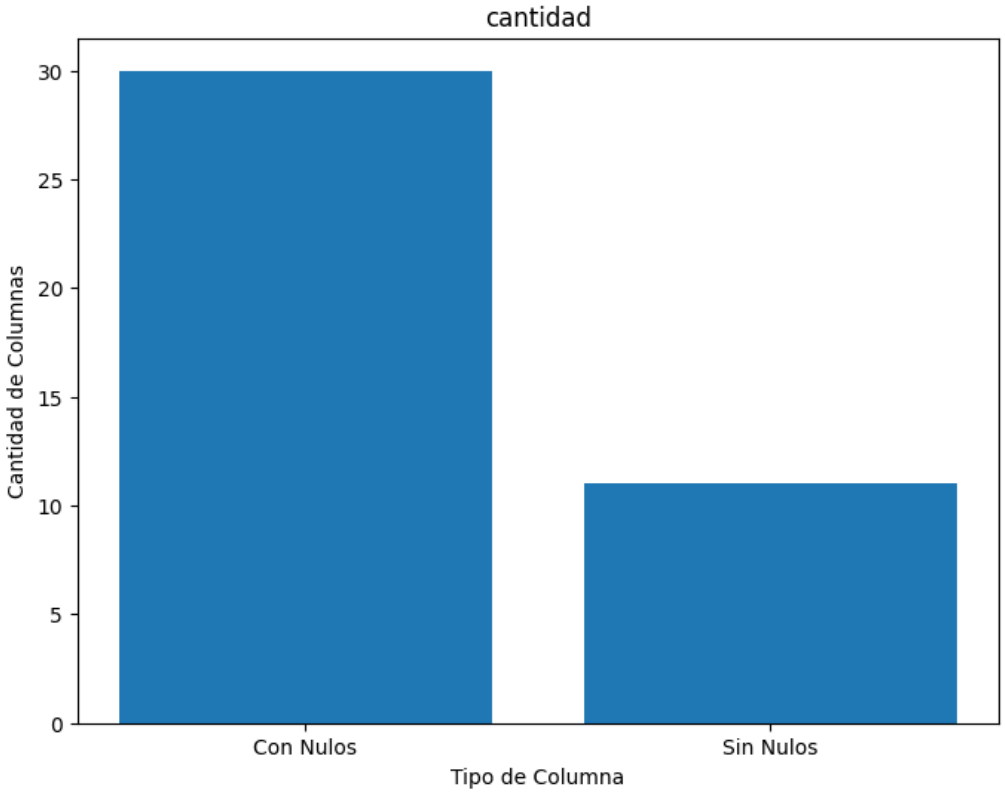
-sec contributory cause::we as a group and with the objective that we have visualized, we do not believe it is convenient to include this column since it does not contribute to us.
beat of occurrence his column has been eliminated as we do not believe it is convenient to know the beat of occurence of the accidents
-photos taken:this column was eliminated since most of its data is null and void and it would not help us to know if photos were taken of the accident or not.
-statements taken:this column was eliminated since most of its data is null and void and it would not help us to know if the driver's declaration was taken.
dooring:this column has been eliminated as we do not believe it is convenient to know if is dooring
-work zone:this column was eliminated since most of its data is null
-work zone type:this column was eliminated since most of its data is null
-workers present:this column was eliminated since most of its data is null
nu units

graphics
-This graph helps us to understand in a visual way which are the types of data that are most frequently presented.

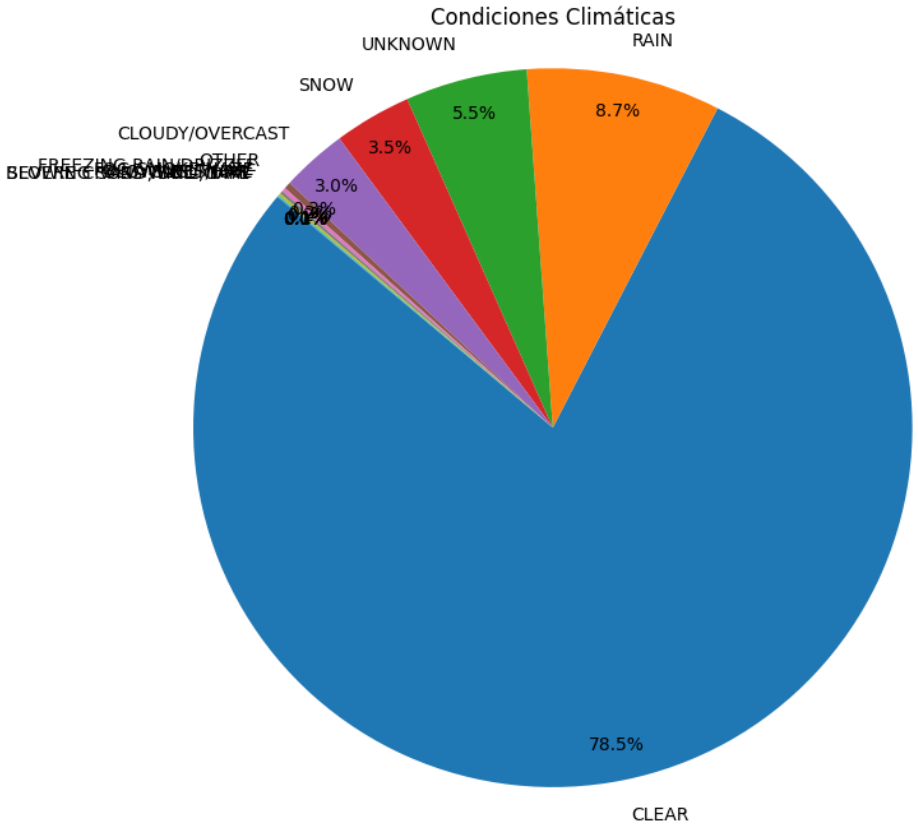


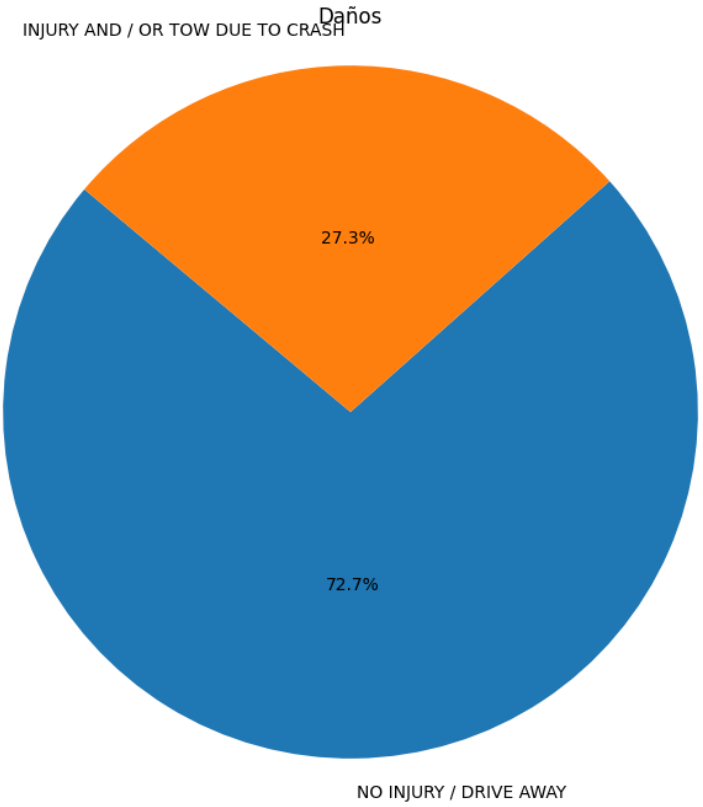
-He graph shows in a visual way the percentage of accidents according to the time of day.



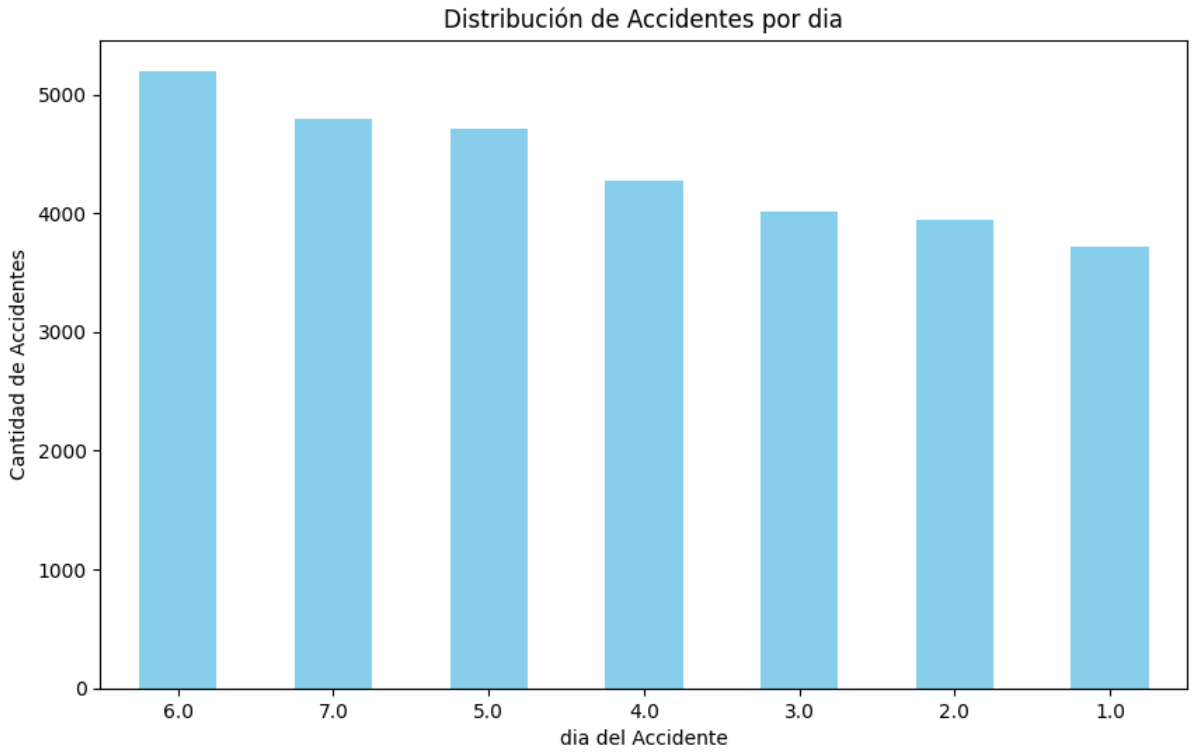


-This graph was taken from the weather conditions column to understand under which conditions accidents occur.

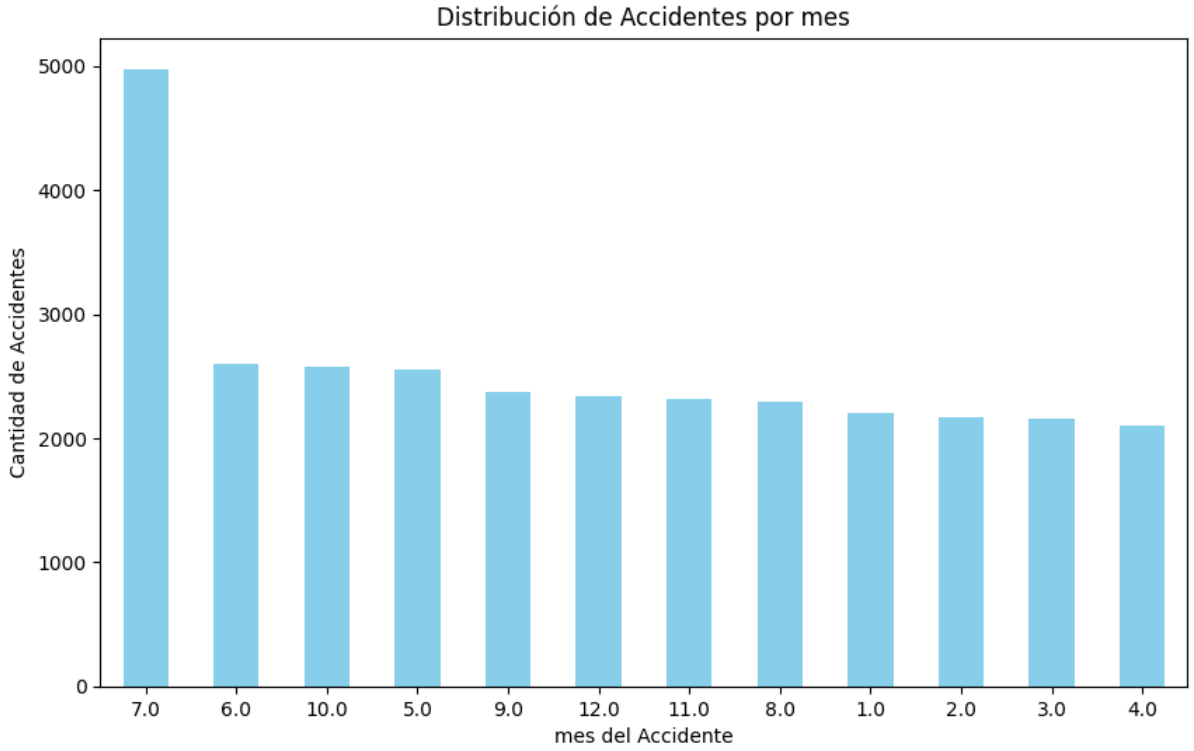




-This graph helps us to understand in a visual way on which day there are more accidents.



-This graph helps us to understand in a visual way in which month there are more accidents.



TRANSFORM

what was done in the transformation, apart from eliminating the columns already mentioned, the null values were eliminated, some columns were changed to integer values since they were in decimals.

the crash type column was also better organized with only 2 types of words, which means that the words after these 2 words were eliminated,the coordinates were organized as they were since they were inverted and the location was at the north pole.
days and months were changed from numbers to corresponding months

API

In our project, we have decided to use the OpenWeatherMap API for several reasons, firstly this platform, owned by Open Weather, gives us access to global weather data in a simple and effective way.
With OpenWeatherMap, we can obtain a wide range of weather information, including current data, forecasts, short-term predictions, and even historical data for virtually any location on the planet. What will really help us is its ability to provide a minute-by-minute hyperlocal precipitation forecast, meaning we will have extremely precise details, with a Mean Absolute Error (MAE) of around 0.5 degrees and a Root Mean Square Error (RMSE) of less than 2 degrees. This assures us that the data we will obtain is reliable, with a reliability rate that ranges between 90% and 100%. Furthermore, the inaccuracy is minimal, approximately 1%, which gives us even more confidence in the quality of the data.
To bring together all this information, OpenWeatherMap collects and processes data from various sources, such as global and local weather models, satellites, radars and a wide network of weather stations. This means that the data we will obtain comes from a variety of sources, ensuring that we have global coverage and accurate details. The versatility of this platform is impressive, as data can be obtained in multiple formats, such as JSON, XML or HTML, making it easy to integrate it into our project application in a custom and convenient way.

In conclusion we have chosen OpenWeatherMap as our main source of weather data in this project due to its reliability, accuracy and versatility. With this platform, we are confident that we will have access to detailed and accurate weather information, which will significantly improve the quality and performance of our project.

- ***Upload to the database-Traffic, Traffic_Convertido y Ciudades:***

```
import sqlalchemy
import pandas as pd
from sqlalchemy import create_engine
```

```
traffic = pd.read_csv('C:/Users/enrri/OneDrive/Documentos/Universidad/Semestre 4/ETL/Traffic_Crashes.csv', sep=',')
```

```
traffic.head()
```

This code imports the necessary libraries and establishes a connection to a database, then loads the data from a CSV file into a dataframe named 'traffic'

```
traffic_convertido = pd.read_csv('C:/Users/enrri/OneDrive/Documentos/Universidad/Semestre 4/ETL/Traffic_convertido.csv', sep=',')
```

This code loads the data from a CSV file into a dataframe named 'Traffic_Convertido

```
ciudades = pd.read_csv('C:/Users/enrri/OneDrive/Documentos/Universidad/Semestre 4/ETL/Cuidades_CO.csv', sep=',')
```

In this code, the data from a CSV file is loaded into a dataframe named ‘Ciudades’.

```
engine = create_engine('postgresql://postgres:Basesdedatosandres@localhost:5432/Proyecto')
engine.connect()
traffic.to_sql('Traffic', engine)
```

This code is responsible for taking a dataframe named 'traffic' and saving it as a table called 'Traffic' in the PostgreSQL database named 'Proyecto.' This database was used to upload 3 tables, which are: ‘Ciudades’, ‘Traffic’ y ‘Traffic_Convertido’

```
engine = create_engine('postgresql://postgres:Basesdedatosandres@localhost:5432/Proyecto')
engine.connect()
traffic_convertido.to_sql('Traffic_convertido', engine)
```

In this code, it takes the dataframe named 'Traffic_Convertido' and saves it as a table called 'Traffic_Convertido' in the PostgreSQL database, in the database named 'Proyecto'

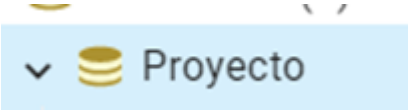
To upload this dataframe, the transformations of the previous dataframe named 'df' had to be saved to a CSV file first, and then it was uploaded later

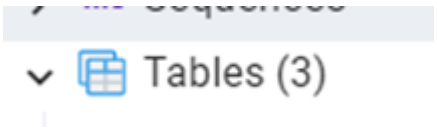
```
nombre_archivo = 'Traffic_convertido.csv'

# Utiliza el método to_csv para guardar el DataFrame en un archivo CSV
df.to_csv(nombre_archivo, index=False)
```

```
engine = create_engine('postgresql://postgres:Basesdedatosandres@localhost:5432/Proyecto')
engine.connect()
ciudades.to_sql('Ciudades', engine)
```

This code takes the dataframe named 'Ciudades' and saves it in a table named 'Ciudades' in the PostgreSQL database, in the database called 'Proyecto'

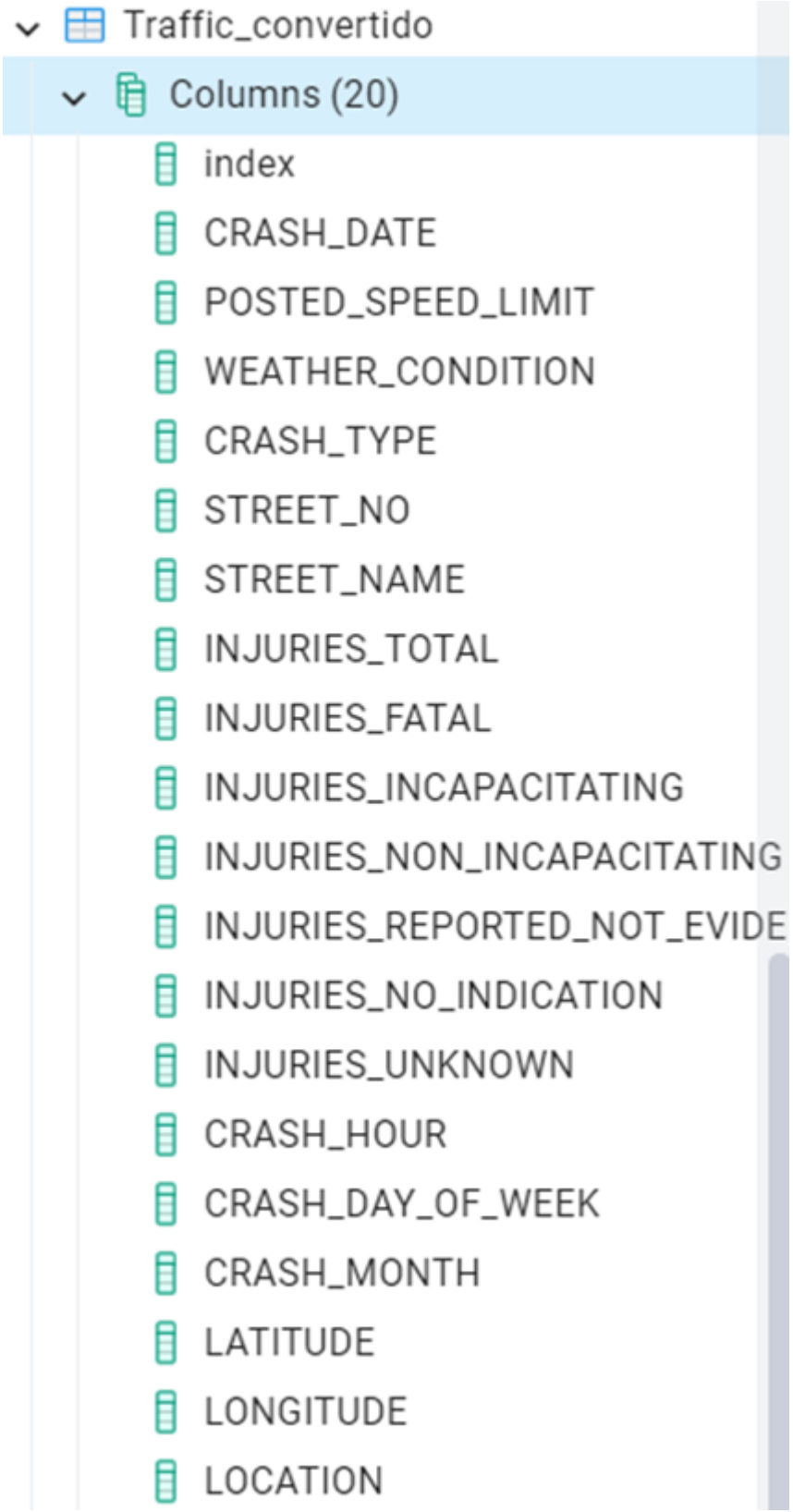



Tables (3)

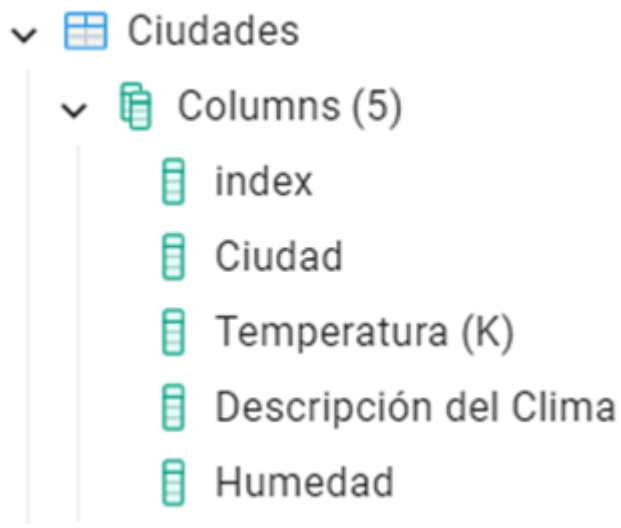
TRAFFIC:


Traffic
Columns (50)
index
CRASH_RECORD_ID
RD_NO
CRASH_DATE_EST_I
CRASH_DATE
POSTED_SPEED_LIMIT
TRAFFIC_CONTROL_DEVICE
DEVICE_CONDITION
WEATHER_CONDITION
LIGHTING_CONDITION
FIRST_CRASH_TYPE
TRAFFICWAY_TYPE
LANE_CNT
ALIGNMENT
ROADWAY_SURFACE_COND

TRAFFIC_CONVERTIDO:



CIUDADES:


Ciudades
Columns (5)
index
Ciudad
Temperatura (K)
Descripción del Clima
Humedad