Proyecto Ingeniería de Datos
Alejandra Contreras
Sebastián Plazas
Dafne Castellanos
Ingeniería de Datos
2022-1

Entrega 2 – Proyecto Final Ingeniería de Datos

Generalidades

Las bases de datos trabajadas en nuestro proyecto se encuentran en los siguientes enlaces (1):

- Motor Vehicle Collisions - Crashes | NYC Open Data (cityofnewyork.us)
- Motor Vehicle Collisions - Person | NYC Open Data (cityofnewyork.us)
- Motor Vehicle Collisions - Vehicles | NYC Open Data (cityofnewyork.us)

La base de datos consta de tres tablas donde se almacenan datos sobre las colisiones en Nueva York. La primera tabla, "Crashes" toma en cuenta la información de "Person" y "Vehicle" con tal de dar una visualización resumida de la base.

En nuestro caso, creamos una base de datos independiente en donde se consideraba la información relevante del caso de estudio de las tres tablas de esta base de datos. Esta base fue llamada COLLISION_DB y podemos encontrar todos los recursos usados en el GitHub. A lo largo del reporte describiremos y aclararemos la información contenida en el repositorio y los pasos para la creación de nuestra base de datos.

A continuación, podemos encontrar la dirección de nuestro repositorio en GitHub. En este repositorio tenemos contenidos los códigos implementados y los archivos correspondientes relacionados con nuestro proyecto. La URL es: https://github.com/Dafne-Castellanos/Proyecto-Ingenier-a-de-Datos

Caso de estudio – Contexto del Proyecto

Hemos sido contratados por un investigador. Nuestro cliente está realizando un estudio de los accidentes vehiculares en la ciudad de Nueva York para implementar medidas que reduzcan la accidentalidad y crear programas de seguridad vial. Hoy en día la ciudad de Nueva York cuenta con ocho millones de habitantes y una superficie de 789 km2, por dicha razón para realizar un mejor análisis de los datos el cliente desea sectorizar la ciudad por los cinco distritos (Brooklyn, Manhattan, Staten Island, Queens y El Bronx). Frente a esto, nos ha solicitado tener ciertas visualizaciones de la información a partir de los datos para analizar los siguientes casos:

1. La situación sanitaria mundial del COVID-19 impacto la movilidad en las distintas ciudades del mundo, por ello, el cliente desea saber las consecuencias de la pandemia en la accidentalidad de Nueva York, así que se busca identificar cuáles son las horas del día en que suceden más colisiones y el comportamiento de los siniestros en los distintos distritos, para ello es necesario revisar los registros de antes, durante y después de la cuarentena.

2. El cliente para la creación de un programa piloto de prevención necesita analizar cuál de los distritos ha tenido la mayor tasa de accidentalidad teniendo en cuenta la población y el área de cada uno.
3. El cliente también quiere saber cuáles son las edades, los sexos y los tipos de las personas que están involucradas en los accidentes, pues necesita guiar los programas de prevención a poblaciones especificas dentro de los distritos.
4. El cliente desea conocer cuáles son los tipos de vehículos que ocasionan más accidentalidad y los daños más sufridos por los vehículos, para así implementar medidas que ayuden a descontinuar motorizados que sean un peligro potencial para los ciudadanos y que ayuden a proteger los automóviles de accidentes fatales.

En este sentido, el cliente provee las bases de datos de las colisiones vehiculares en la ciudad de Nueva York (1).

Reglas de negocio

De acuerdo con el contexto, a continuación, se muestra el planteamiento de nuestras las reglas de negocio.

- Varias colisiones pueden involucrar muchas personas.
- En un distrito pueden ocurrir muchas colisiones.
- Muchos vehículos pueden chocar en varias colisiones.
- En varios vehículos pueden ir muchas personas.
- Las colisiones se deben identificar con un número.
- Al cliente le interesa saber la fecha y la hora en que ocurrió el accidente.
- Cada distrito se debe identificar con su nombre.
- De los distritos, al cliente le interesa saber la población y el área.
- De las colisiones se debe tener el zip code y la localización que está compuesta de longitud y latitud, dichos datos se ubican en un distrito.
- Si se tiene hasta n tipos de vehículos, entonces los tipos de vehículos que son antecesores deben tener registro.
- Los registros de personas deben tener un identificador único.
- Los números de personas fallecidas y heridas en las colisiones deben ser mayores o iguales a 0.
- Las personas deben tener un número que las identifique, por ejemplo, la cedula.
- Las personas pueden ser peatones, ciclistas, ocupantes del vehículo u otro motorizado.
- Las personas pueden estar heridas o haber muerto o no tener una especificación.
- Se debe conocer el género de las personas.
- De las personas se debe conocer su edad.
- Los registros de vehículos deben tener un identificador único.
- Los vehículos deben tener un número que los identifique, por ejemplo, la placa.
- Se deben contar con el tipo de los vehículos.
- Se debe tener los daños de los vehículos.
- Se pueden consignar hasta cuatro daños de un vehículo.

Base de datos - Descripción de las columnas

En el proyecto estamos trabajando con cuatro bases de datos; tres de ellas proveen los registros de las colisiones de vehículos motorizados, las personas y los vehículos involucrados, dichos datos son recopilados por el Departamento de Policía de Nueva York con el objetivo de realizar análisis detallados de seguridad de tráfico. La base de datos restante proporciona las cifras de la población y el área de cada distrito, se incluyó pues era necesaria para sectorizar la información. Esta última se realizó con base en el siguiente enlace:

- https://www.citypopulation.de/en/usa/newyorkcity/

De esta forma, tenemos las siguientes bases de datos:

- Motor Vehicle Collisions – Crashes
- Motor Vehicle Collisions – Person
- Motor Vehicle Collisions – Vehicles
- Borough

Las tres primeras bases tienen en total 75 columnas, en las cuales se destacan collision_id, person_unique_id y vehicle_unique_id, pues son las llaves primarias. No obstante, por los casos de análisis planteados anteriormente decidimos tomar las columnas necesarias que provean datos relevantes para las consultas que vamos a realizar. A continuación, enunciamos dichas columnas, la base de datos a la que pertenecen, su descripción y las razones por las que consideramos tomarla:

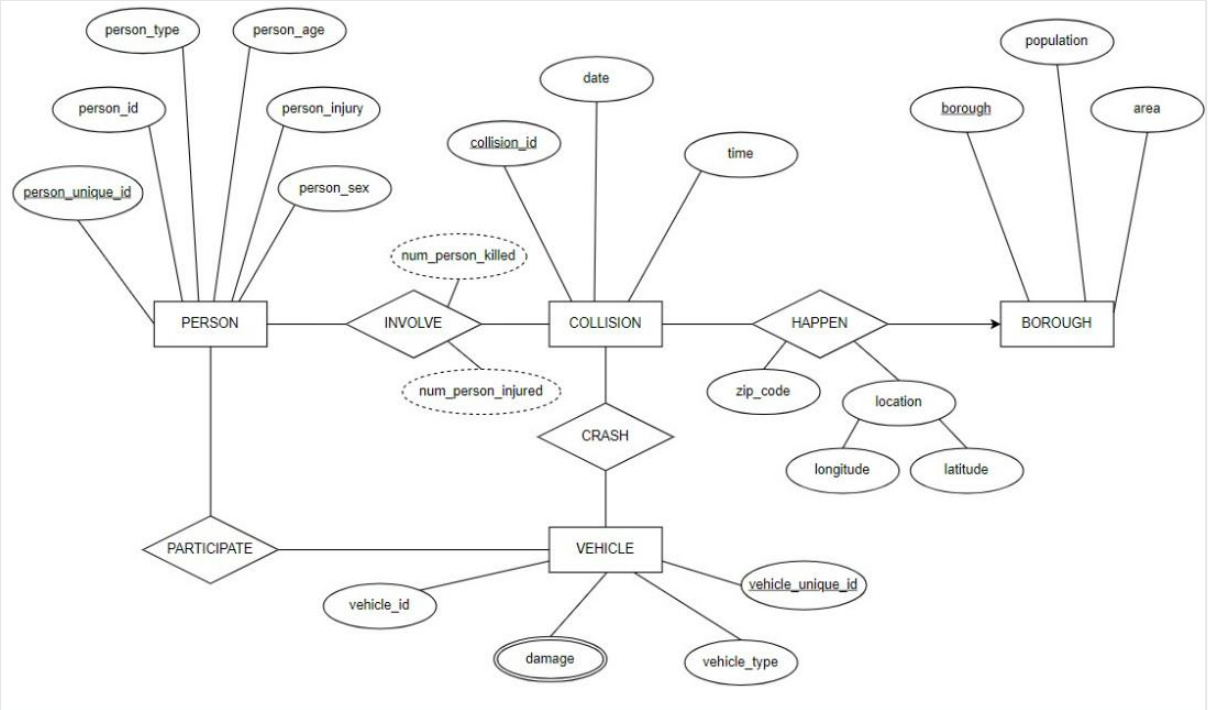
Base de datos	Nombre columna	Descripción	Razones
Motor Vehicle Collisions – Crashes	collision_id	Código de registro único generado por el sistema.	Se incluyo pues es la llave primaria de COLLISION.
	accident_date	Fecha de ocurrencia de la colisión.	Se necesita para saber cuándo ocurrieron las colisiones.
	accident_time	Hora de ocurrencia de la colisión.	Se necesita para saber las horas de las colisiones.
	zip code	Código postal de ocurrencia del incidente.	El zip code ayuda a saber la ubicación del accidente en caso de no tener el distrito.
	latitude	Coordenada de latitud para el Sistema Global de Coordenadas, WGS 1984, grados decimales (EPSG 4326).	La latitud ayuda a ubicar el accidente en caso de no tener el distrito.
	longitude	Coordenada de longitud para el Sistema Global de Coordenadas, WGS	La longitud ayuda a ubicar el accidente en caso de no tener el distrito.

		1984, grados decimales (EPSG 4326).	
	location	Par de longitud y latitud.	Es el par formado por longitud y longitud.
	number of persons injured	Número de personas lesionadas.	Se necesita para el número de víctimas heridas.
	number of persons killed	Número de personas fallecidas.	Se necesita para saber el número víctimas fatales.
Motor Vehicle Collisions – Person	unique_id	Código de registro único generado por el sistema.	Se incluyo pues es la llave primaria de la tabla PERSON.
	person_id	Código de identificación de la víctima asignado por el sistema.	Se necesita pues el id que identifica a cada persona (ejemplo: cédula).
	person_type	Ciclista, Ocupante, Peatón, etc.	Se necesita para saber que tipos de personas están involucradas en accidentes.
	person_injury	Heridos, fallecidos, no especificados.	El estado de las víctimas ayuda a generar nuevos análisis respecto a las personas involucradas.
	person_age	Edad de la víctima	Las edades ayudan a generar nuevos análisis respecto a las personas involucradas.
	person_sex	Género de la víctima	Los sexos ayudan a generar nuevos análisis respecto a las personas involucradas.
Motor Vehicle Collisions – Vehicles	unique_id	Código de registro único generado por el sistema	Se incluyó pues el a llave primaria de VEHICLE.
	vehicle_id	Código de identificación del vehículo asignado por el sistema	Se necesita pues la forma en cómo se identifica cada vehículo (ejemplo: la placa).
	vehicle_type	Tipo de vehículo basado en la categoría de vehículo	Se necesita para saber los tipos de vehículos

		seleccionada (sedan, SUV, taxi, moto, etc)	involucrados en los accidentes.
	vehicle_damage	Ubicación en el vehículo donde ocurrió la mayor parte del daño	Se incluyó pues ayuda a identificar las partes más dañadas para buscar como protegerlas.

Diagrama Entidad Relación

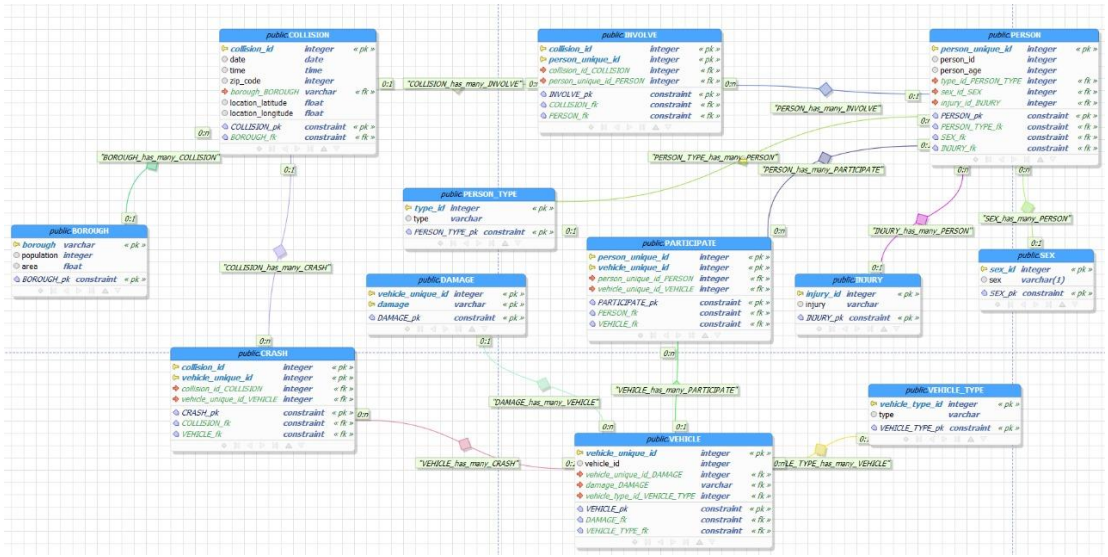
Podemos ver el planteamiento de nuestro modelo entidad relación:



Como mencionamos, los atributos y entidades creadas, se relacionan con nuestro caso de estudio y con el objetivo planteado por el cliente. Por tanto, no se encuentra la totalidad de las columnas de la base de datos (1), únicamente las necesarias.

Diagrama Relacional Normalizado

Teniendo en cuenta nuestro diagrama ER, realizamos nuestro diagrama relacional normalizado:



En este caso, tuvimos en cuenta los requisitos para llegar a la tercera forma normal. Por lo tanto, normalizamos la siguiente información:

1. De la tabla PERSON, derivamos las tablas: SEX, INJURY y PERSON_TYPE.
2. De la tabla VEHICLE, derivamos las tablas: VEHICLE_TYPE y DAMAGE.

Seleccionamos estas tablas debido a que recaían en una redundancia horizontal. Además, se tenían anomalías en el caso de querer actualizar, borrar o insertar información, pues por alto volumen de datos disponibles en las tablas la modificación manual no es óptima y de igual forma, no se soportaba la integridad de los datos en caso realizar alguna modificación en alguno de los campos.

Ahora, para la creación de estas tablas, en cada una de ellas, generamos un identificador único que tenía una referencia a la tabla de principal (PERSON o VEHICLE), una referencia su nombre (SEX, INJURY, PERSON_TYPE, VEHICLE_TYPE o DAMAGE) y el numero de la categoría. Por ejemplo, en el caso de VEHICLE_TYPE sus identificadores se realizaron de la siguiente manera. Tomamos en cuenta V como referencia a la tabla VEHICLE, T como el de VEHICLE_TYPE y el numero correspondiente a la categoría. Así los identificadores son de la forma VT01, VT02, etc.

De esta forma los identificadores de estas tablas tienen la siguiente forma:

Tabla principal	**Tabla creada**	**Forma del identificador**
PERSON	SEX	PS##
	INJURY	PI##
	PERSON_TYPE	PT##
VEHICLE	VEHICLE_TYPE	VT##
	DAMAGE	VD##

Las categorías tomadas en cuenta se realizaron posteriormente de hacer una limpieza de los datos. Así pues, las tablas de resultantes de la normalización son las siguientes:

- SEX

	sex_id [PK] character varying (6)	sex character varying
1	PS01	F
2	PS02	M
3	PS03	U
4	PS04	nd

- INJURY

	person_injury_id [PK] character varying (6)	person_injury character varying
1	PI01	Unspecified
2	PI02	Injured
3	PI03	Killed

- PERSON_TYPE

	person_type_id [PK] character varying (6)	person_type character varying
1	PT01	Occupant
2	PT02	Pedestrian
3	PT03	Bicyclist
4	PT04	Other Motorized

- DAMAGE

	vehicle_damage_id [PK] character varying (6)	vehicle_damage character varying
1	VD01	Center Front End
2	VD02	Left Front Bumper
3	VD03	Center Back End
4	VD04	Right Front Bumper
5	VD05	No Damage
6	VD06	Left Front Quarter Panel
7	VD07	Right Front Quarter Panel
8	VD08	Left Rear Quarter Panel
9	VD09	Left Side Doors
10	VD10	Left Rear Bumper
11	VD11	Right Side Doors
12	VD12	Right Rear Quarter Panel
13	VD13	Right Rear Bumper
14	VD14	Other
15	VD15	Roof
16	VD16	Demolished
17	VD17	Trailer
18	VD18	Overturned
19	VD19	Undercarriage
20	VD20	nd

- VEHICLE_TYPE

	vehicle_type_id [PK] character varying (6)	vehicle_type character varying
1	VT01	Sedan
2	VT02	SUV
3	VT03	nd
4	VT04	Taxi
5	VT05	Pickup
6	VT06	Van
7	VT07	Bus
8	VT08	Truck
9	VT09	Bike
10	VT10	Moto
11	VT11	Ambulance
12	VT12	Otros

Base de Datos COLLISION_BD en PostgreSQL

Con el diagrama normalizado, realizamos la creación de las tablas de manera que se respetara la información consignada en el modelo. De esta manera, en el archivo tables o “tables ddl.txt” en el GitHub se puede observar la creación de la base de datos junto con las tablas.

Nota: Si se llega a tener alguna duda en el proceso de cargado de los datos en PostgreSQL o la conexión con Python o con alguno de los pasos que se describirán a continuación, se sugiere consultar el siguiente video: video_instructivo

En el video va a encontrar como descargar los archivos del repositorio GitHub, como exportarlos, como pasarlos a la unidad C:, como correr en PostgreSQL la creación de las tablas y como insertar los datos en ellas. Además, de cómo realizar la conexión con Python y como ejecutar las consultas respectivas.

El acceso a esta información se obtiene con los siguientes pasos:

1. Ingresar al enlace de GitHub: https://github.com/Dafne-Castellanos/Proyecto-Ingenier-a-de-Datos
2. Descargar o abrir el archivo de texto “tables ddl.txt”

Para continuar con la creación del DDL de la base debemos:

3. Ingresar a PostgreSQL
4. Correr el código contenido de “tables ddl.txt” en un script de PostgreSQL

Con esto vamos a tener la creación de la base de datos COLLISION_DB con sus respectivas tablas.

Ahora bien, para el cargue de la información de la base de datos vamos a hacer lo siguiente:

1. Crear una carpeta llamada "dataDB" en la unidad C: de su equipo.

2. Descargar los siguientes archivos csv y rar

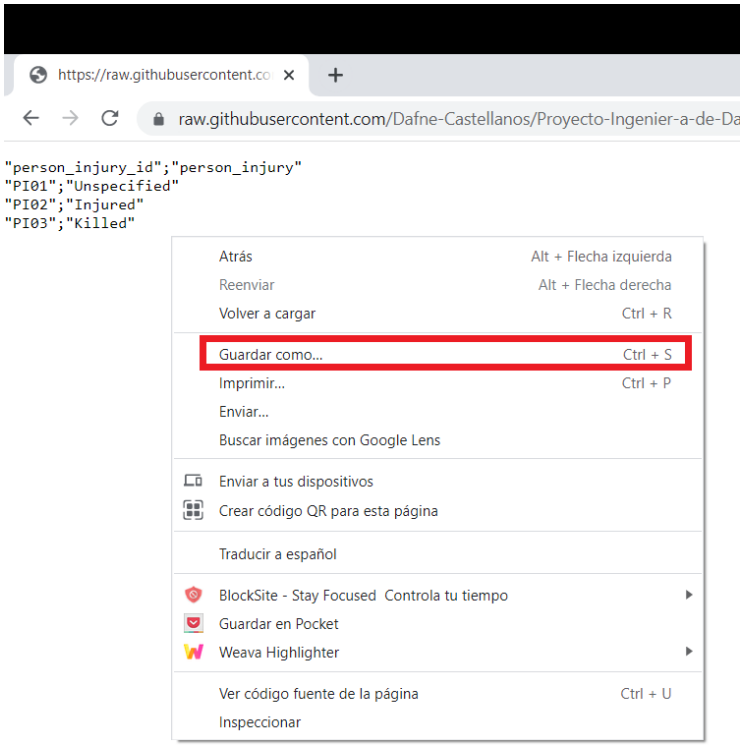
De esta descarga estos archivos ya están listos para insertar en PostgreSQL:

a. person_injury.csv
b. person_sex.csv
c. person_type.csv
d. vehicle_damage.csv
e. vehicle_type.csv

La descarga de estos archivos se debe hacer de la siguiente forma:

1. Oprimir el boton raw

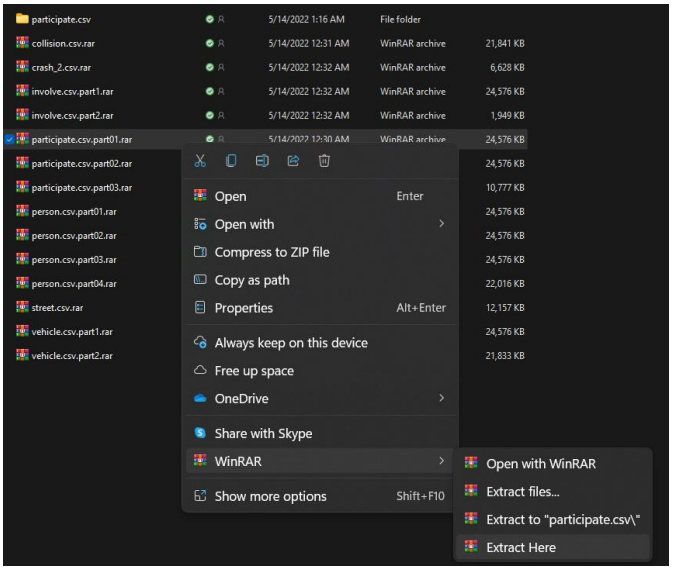
2. Clic derecho y Guardar como, luego selecciona la unidad C:



Los demás, son RAR por tanto debemos tomar en cuenta el siguiente procedimiento:

1. Clic derecho a una carpeta rar
2. Seguir la ruta WinRAR\ExtractHere

Un ejemplo de lo anterior se realizó con participate.csv.rar

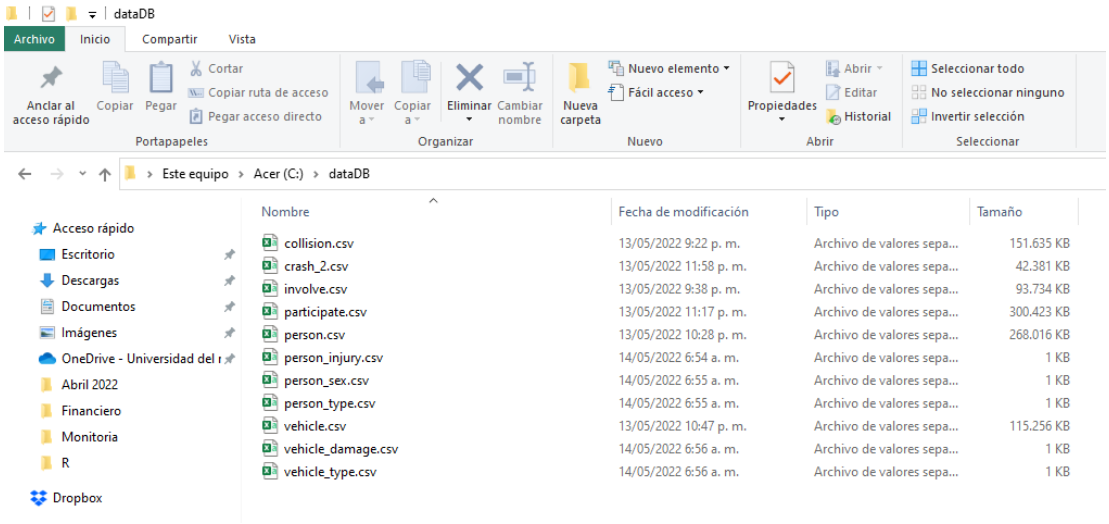


Esto lo haremos para:

a. collision.csv.rar
b. crash2.csv.rar
c. involve.csv.rar
d. participate.csv.rar
e. person.csv.rar
f. vehicle.csv.rar

Nota: Se deben descargar todos los archivos rar, no obstante, solo es necesario extraer la parte 1 de cada uno de ellos.

De esta manera, tenemos en nuestra carpeta C:\dataDB todos los archivos csv necesarios.



Ahora, continuamos con

3. Descargar o abrir el archivo de texto “insertion.txt”
4. Ingresar a PostgreSQL
5. Correr el código contenido de “insertion.txt” en un script de PostgreSQL

Al ejecutar este último código, se deben realizar las inserciones de los datos a las tablas y es así como pasamos nuestra información y base de datos a PostgreSQL.

Conexión a Python

Antes de realizar la conexión con Python, es necesario tener instalado el paquete psycopg2, si no se tiene instalado recomendamos ver la documentación que se encuentra en el siguiente enlace: https://pypi.org/project/psycopg2/

Luego de tener instalado psycopg2 podemos realizar los siguientes pasos para realizar y visualizar las consultas:

1. Descargar el archivo Conexion_consulta.py que se encuentra en el GitHub.
2. Enviar el archivo Conexion_consulta.py a la carpeta dataDB que está en la unidad C: creada en los anteriores pasos.
3. Abrir el símbolo del sistema y correr las siguientes líneas (una por una):

```
py -m venv prueba
cd prueba
cd Scripts
activate.bat
```

4. Correr esta línea: `py C:\dataDB\Conexion_Consulta.py`