

Curso PostgreSQL

Autor: Luis Fernando Apáez Álvarez

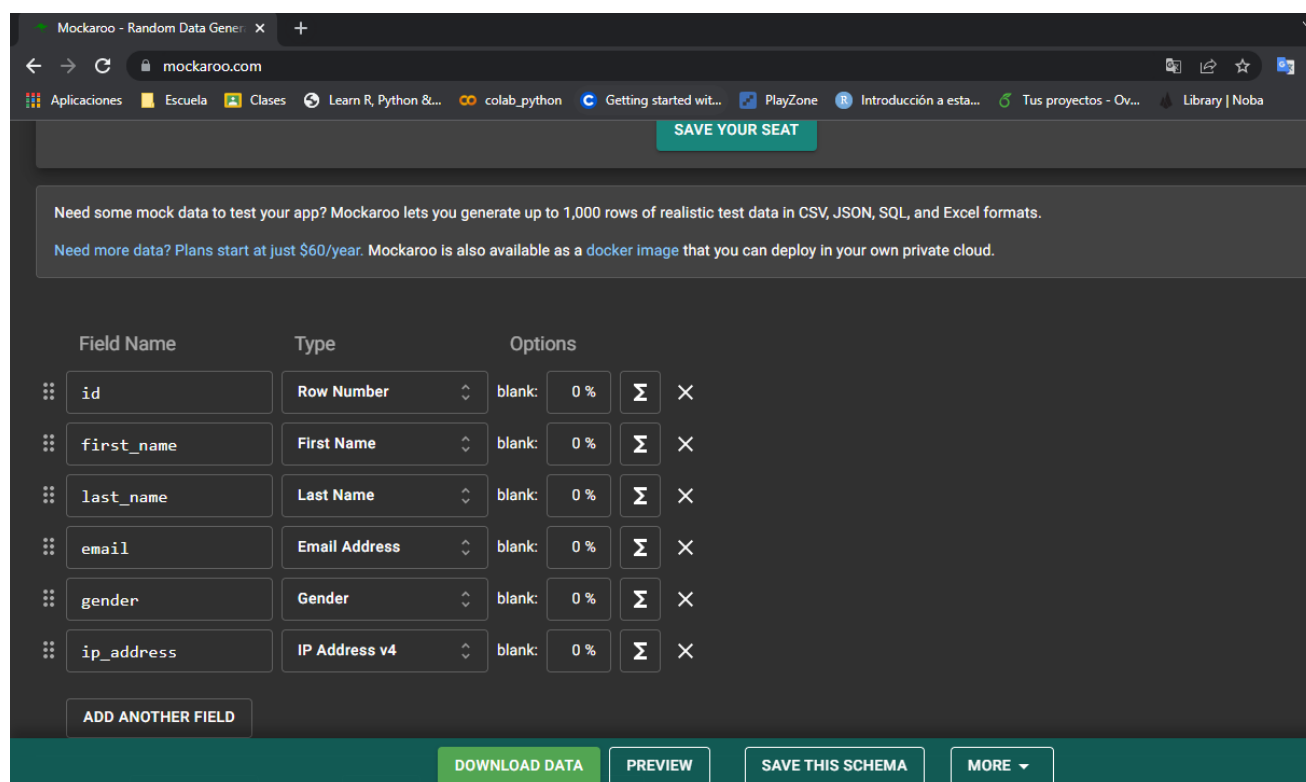
Realizando 1000 inserciones

Hasta el momento, al menos en las tres tablas que trabajamos en clase (`empleados`, `ventas` y `productos`) ya hemos definidos las principales restricciones: las llaves primarias y foráneas. Con lo anterior garantizamos la integridad de nuestros datos y evitamos errores a la hora de ingresar información en cada una de ellas.

Ahora bien, para trabajar de la mejor forma en clases posteriores ingresaremos bastantes filas a nuestras tres tablas base (las que mencionamos antes), lo que nos permitirá introducirnos a manejar bases de datos con muchos registros.

Realizaremos pues 1000 inserciones en las tablas `empleados`, `ventas` y `productos`, pero no ingresaremos de forma manual toda esta información, nos auxiliaremos de la página [mockaroo](https://mockaroo.com) para generar las 1000 inserciones. En dicha página se creará de forma aleatoria la información que le pidamos, de acuerdo al tipo de dato de las columnas de nuestras tablas.

Una vez en la página de mockaroo veremos algo como



The screenshot shows the Mockaroo website interface for generating random data. At the top, there's a navigation bar with a "SAVE YOUR SEAT" button. Below it, a message states: "Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats. Need more data? Plans start at just \$60/year. Mockaroo is also available as a docker image that you can deploy in your own private cloud." The main area features a table with columns: Field Name, Type, and Options. The table contains 6 rows of field definitions:

Field Name	Type	Options
id	Row Number	blank: 0 % Σ ×
first_name	First Name	blank: 0 % Σ ×
last_name	Last Name	blank: 0 % Σ ×
email	Email Address	blank: 0 % Σ ×
gender	Gender	blank: 0 % Σ ×
ip_address	IP Address v4	blank: 0 % Σ ×

Below the table is a button labeled "ADD ANOTHER FIELD". At the bottom of the interface, there are four buttons: "DOWNLOAD DATA", "PREVIEW", "SAVE THIS SCHEMA", and "MORE".

Primero generaremos la información para las tablas `empleados` y `productos`, pues recordemos que en la tabla `ventas` hay dos columnas con llaves foráneas que depende de la información de estas primeras dos tablas.

Para la tabla `empleados`.

Comenzamos por realizar una consulta

```
SELECT * FROM empleados;
```

Data Output		Explain	Messages	Notifications			
	<code>id_empleados</code> [PK] integer	<code>nombre</code> character varying	<code>app</code> character varying	<code>apm</code> character varying	<code>puesto</code> character varying	<code>salario</code> real	<code>horario</code> character varying
1	1	Luis	Apáez	Álvarez	Cajero	8500	6:00-14:00
2	2	Zed	Ocaña	López	Vigilante	12500.5	6:00-13:00

con base en lo anterior escribimos en mockaroo

Field Name	Type	Options
<div><div></div>nombre</div>	<div>First Name</div>	blank: 0 % <div>Σ</div> <div>×</div>
<div><div></div>app</div>	<div>Last Name</div>	blank: 0 % <div>Σ</div> <div>×</div>
<div><div></div>apm</div>	<div>Email Address</div>	blank: 0 % <div>Σ</div> <div>×</div>
<div><div></div>puesto</div>	<div>Gender</div>	blank: 0 % <div>Σ</div> <div>×</div>
<div><div></div>salario</div>	<div>IP Address v4</div>	blank: 0 % <div>Σ</div> <div>×</div>
<div><div></div>horario</div>	<div>IP Address v4</div>	blank: 0 % <div>Σ</div> <div>×</div>
<div>ADD ANOTHER FIELD</div>		

es decir, comenzamos colocando el nombre de las columnas de la tabla. Posteriormente agregamos el tipo de dato que le corresponde

Field Name	Type	Options
nombre	First Name	blank: 0 % Σ X
app	Last Name	blank: 0 % Σ X
apm	Last Name	blank: 0 % Σ X
puesto	Gender	blank: 0 % Σ X
salario	IP Address v4	blank: 0 % Σ X
horario	IP Address v4	blank: 0 % Σ X

ADD ANOTHER FIELD

agregamos el tipo de dato para las columnas `nombre`, `app`, `apm` buscando en el menú que se despliega de pulsar en las flechas (lo que está en color rojo). En la columna `nombre` se generarán nombres aleatorios (*first name*) y en las columnas `app`, `apm` se generarán apellidos (*last name*). Después, trabajaremos con las columnas `salario` y `horario`. Para `salario`, en el menú que se despliega de pulsar la flechas buscaremos la opción *number*

Choose a Type

number

All (5) Advanced (0) Basic (0) Car (0) Commerce (0) Construction (0) Crypto (0) Health (0) IT (0) Location (0) Nature (0) Personal (0) Product (0)

DUNS Number	NHS Number	Number	Row Number
Randomly generated DUNS numbers	10-digit NHS number with mod11 checksum	0.25	1
		5.2	2
		1000	3

Street Number

A street number between 1 and 5 digits

6449

lo que nos arroja

salario	Number	min: 1 max: 100 decimals: 0 blank: 0 % Σ X
horario	IP Address v4	blank: 0 % Σ X

y personalizamos los valores para la columna `salario`. En nuestro caso colocamos

salario	Number	min: 5000 max: 50000 decimals: 1 blank: 0 % Σ X
horario	IP Address v4	blank: 0 % Σ X

Después pasamos a la columna `horario`. Para ello, como le hicimos en la columna anterior, buscamos *character sequence*, con lo que obtendremos



y después lo personalizamos como

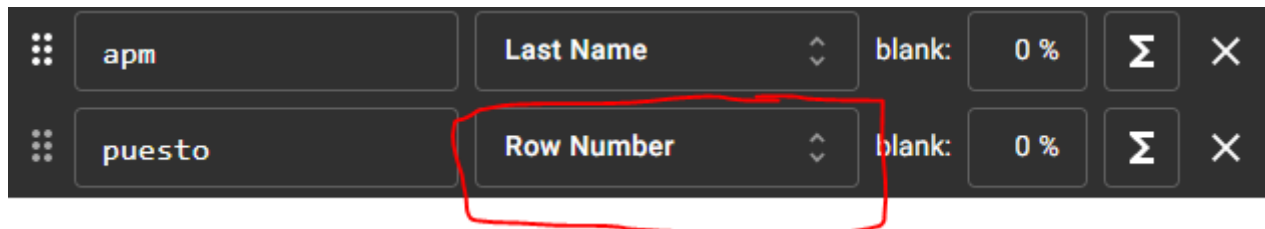


si tienes dudas de porqué colocamos lo anterior puedes pulsar el símbolo de ayuda y mockaroo arrojará una breve guía

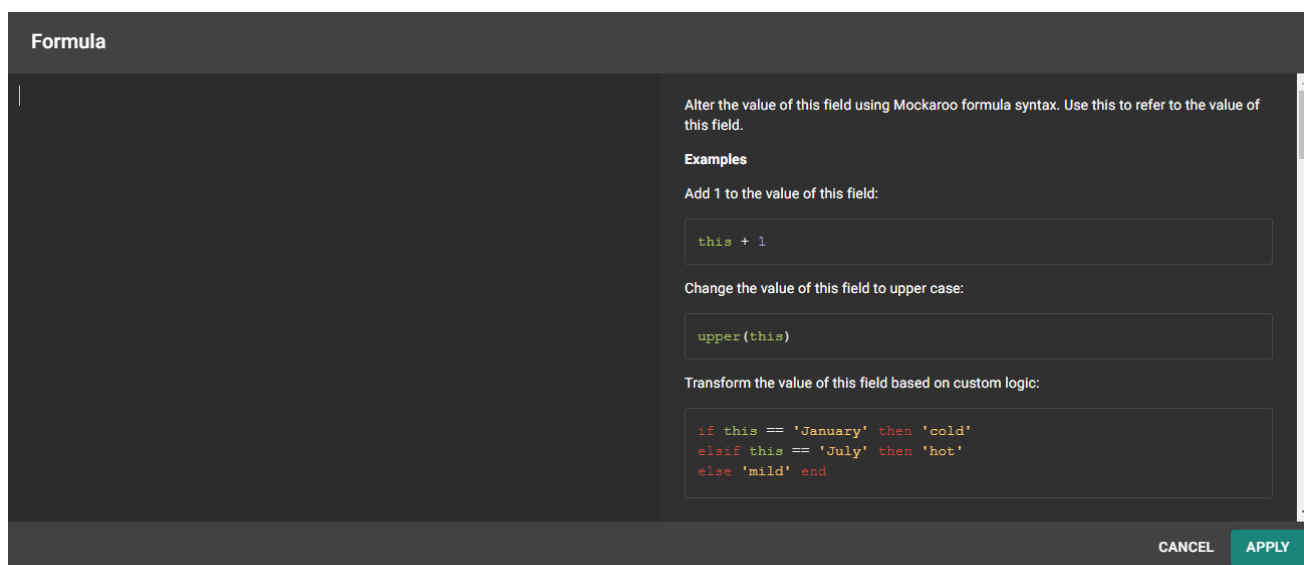


Finalmente, en la columna `puesto` utilizaremos un poco de programación, pues en nuestro caso los puestos de trabajo son muy específicos. Supondremos que sólo hay cuatro puestos de trabajo en la papelería: cajero, vendedor, vigilante y limpieza. Para conseguir asignar a cada uno de los 1000 empleados que generaremos un puesto de trabajo haremos lo siguiente:

- A cada empleado le asignaremos un número de la siguiente forma: al primer empleado de la tabla le asignamos el 1, al segundo el 2, etcétera. Esto lo conseguimos con *row number*:



- Después pulsamos el botón sigma, el cual nos permite escribir código



Ahí, la idea será la siguiente:

1. A los primeros 250 empleados, que tienen asignados los números del 1 al 250 en la columna `puesto` (lo cual conseguimos al colocar en `row_number` en mockaroo) les asignaremos el puesto de trabajo de cajero. El código que emplearemos será:

```
if puesto <= 250 then 'cajero'
```

es decir, si el número asignado al puesto es menor o igual a 250 entonces escribiremos en la columna `puesto` *cajero*.

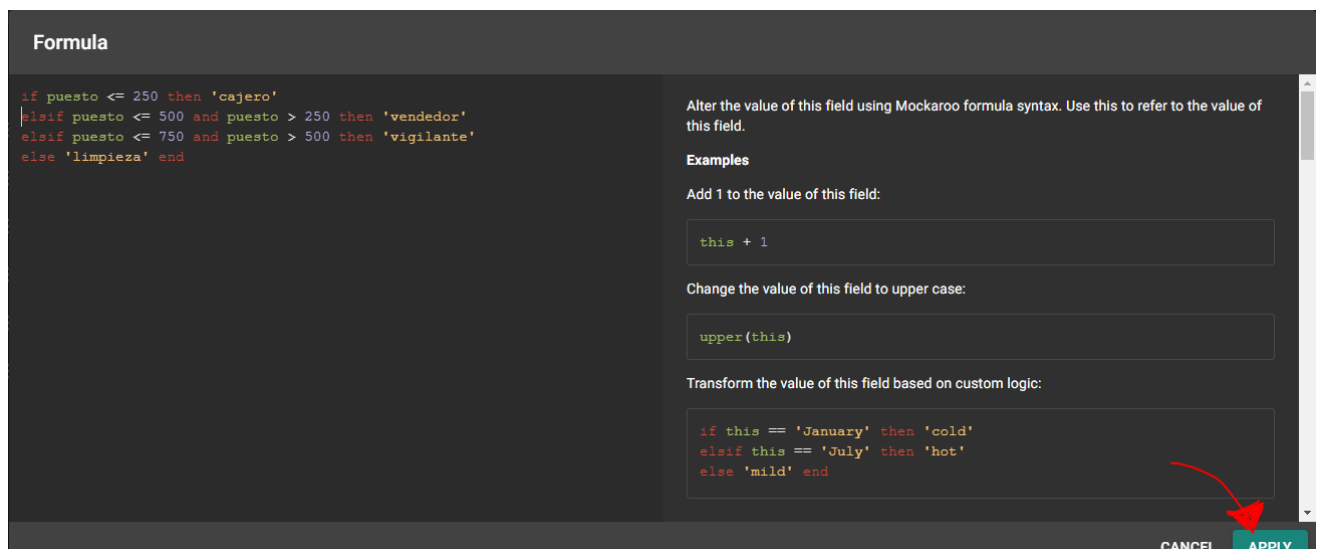
2. A los empleados entre el 250 y 500 les asignaremos el puesto de trabajo de vendedor:

```
elseif puesto <= 500 and puesto > 250 then 'vendedor'
```

Y el resto queda como

```
elseif puesto <= 750 and puesto > 500 then 'vigilante'  
else 'limpieza' end
```

En mockaroo tendríamos



pulsamos *Apply* para aplicar la fórmula del código que escribimos. Después previsualizamos los datos que se generaron. Para ello, en la pestaña principal pulsamos en *Preview*

Field Name Type Options

nombre	First Name	blank: 0 %
app	Last Name	blank: 0 %
apm	Last Name	blank: 0 %
puesto	Row Number	blank: 0 %
salario	Number	min: 5000 max: 50000 decimals: 1 blank: 0 %
horario	Character Sequence	#:00 - #:00 blank: 0 %

ADD ANOTHER FIELD

Rows: 1000 Format: CSV Line Ending: Unix (LF) Include: ☒ header ☐ BOM

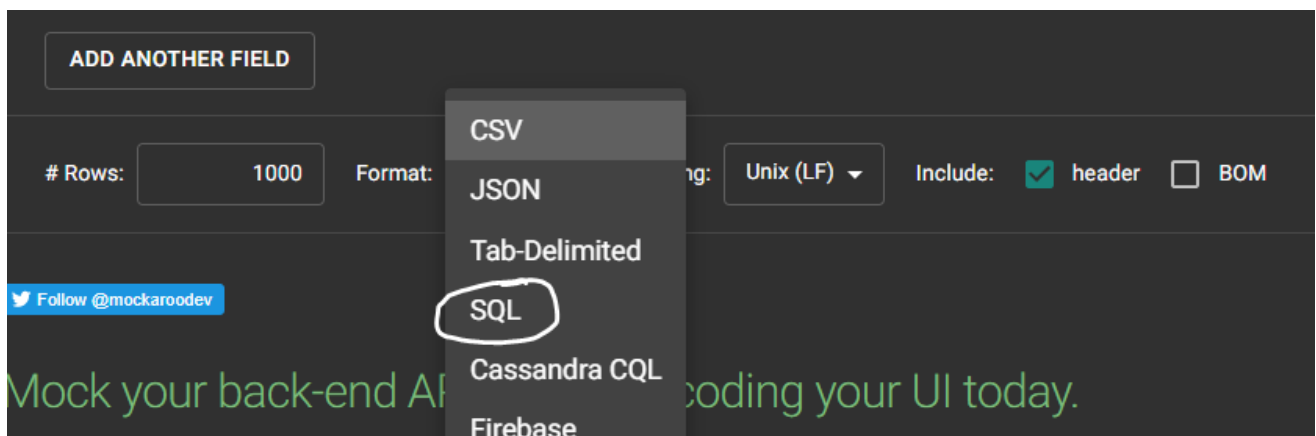
DOWNLOAD DATA PREVIEW SAVE THIS SCHEMA MORE

donde:

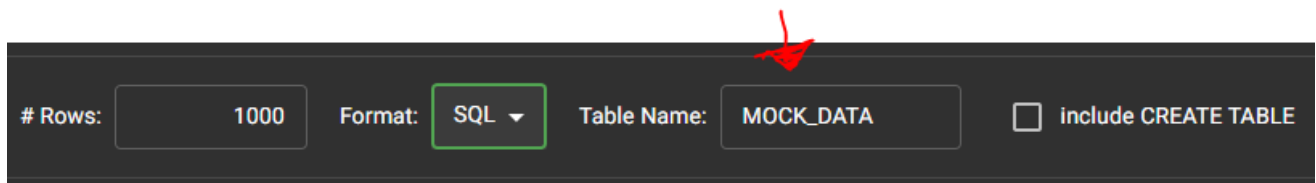
1. Nos arroja la previsualización de los datos que generamos, en nuestro caso:

Preview					
				TABLE	RAW
nombre	app	apm	puesto	salario	horario
Cyndi	Pavolillo	Infantino	cajero	11985.5	0:00 - 1:00
Riannon	Loughan	Syphus	cajero	15476.7	8:00 - 3:00
Guillaume	Glazard	Ivimy	cajero	16456.1	9:00 - 1:00
Amandi	Hearnshaw	Castiglio	cajero	24327.2	7:00 - 6:00
Farlie	Muzzall	Packe	cajero	15524.6	3:00 - 5:00
West	Jacson	Fender	cajero	12219.2	2:00 - 9:00
Rasla	de Broke	Sibbson	cajero	47482.4	9:00 - 4:00
Rory	Kinsey	Fussell	cajero	34328.3	5:00 - 7:00
Cindi	Larrett	Eastby	cajero	22452.1	5:00 - 9:00
Klaus	Neilus	Le Grice	cajero	23631.8	5:00 - 4:00
Irma	Kirkbride	Manicom	cajero	24588.1	9:00 - 0:00

2. Pulsamos en *Format* y buscamos la opción de *sql*



una vez seleccionada la opción de *SQL* nos aparecerá una opción a la derecha denominada *Table Name*. En ella eliminamos el texto *MOCK_DATA* y colocamos el nombre de nuestra tabla, la cual es `empleados`

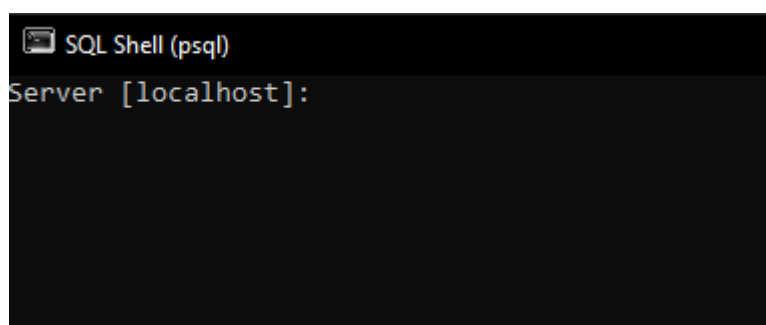


Finalmente:

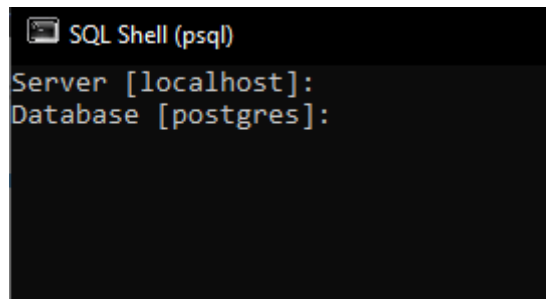
3. Descargamos el archivo `.sql` pulsando en *download data*. Es importante saber la ruta donde descargamos dicho archivo.

Ahora bien, realizaremos las 1000 inserciones en la tabla `empleados`, las cuales tenemos ya en el archivo que descargamos. No obstante, no iremos a *pgAdmin* para efectuar dichas inserciones pues muy seguramente esto sería muy lento debido al número de inserciones que creamos en el archivo `.sql`, además podríamos correr el riesgo de que nuestra pc se trabe. Para solucionar esto trabajaremos directamente con postgres. Cabe resaltar que *pgAdmin* es un cliente postgres, es decir, en él podemos utilizar postgres, siendo así que éste es un intermediario entre el usuario (nosotros) y PostgreSQL.

Para interactuar de manera directa con postgres utilizaremos la consola *psql* la cual descargamos en conjunto con *pgAdmin*. Para ello iremos al buscador de aplicaciones y escribimos justamente *psql*. Después de abrirlo nos debe arrojar la siguiente ventana



en la cual nos dice que estamos trabajando con el servidor local (nuestra pc). Damos enter y nos debe de aparecer

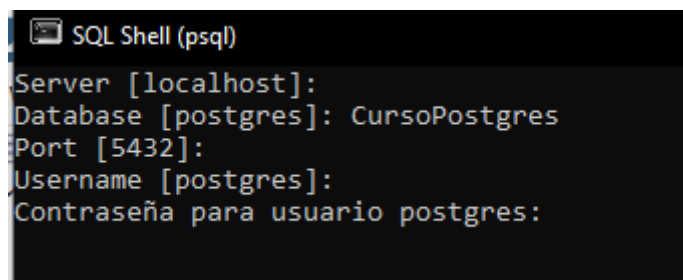


```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
```

que hace alusión a la base de datos con la cual queremos trabajar. En mi caso, las tablas `empleados`, `ventas`, `productos` se encuentran en una base de datos denominada *CursoPostgres*, por lo que en

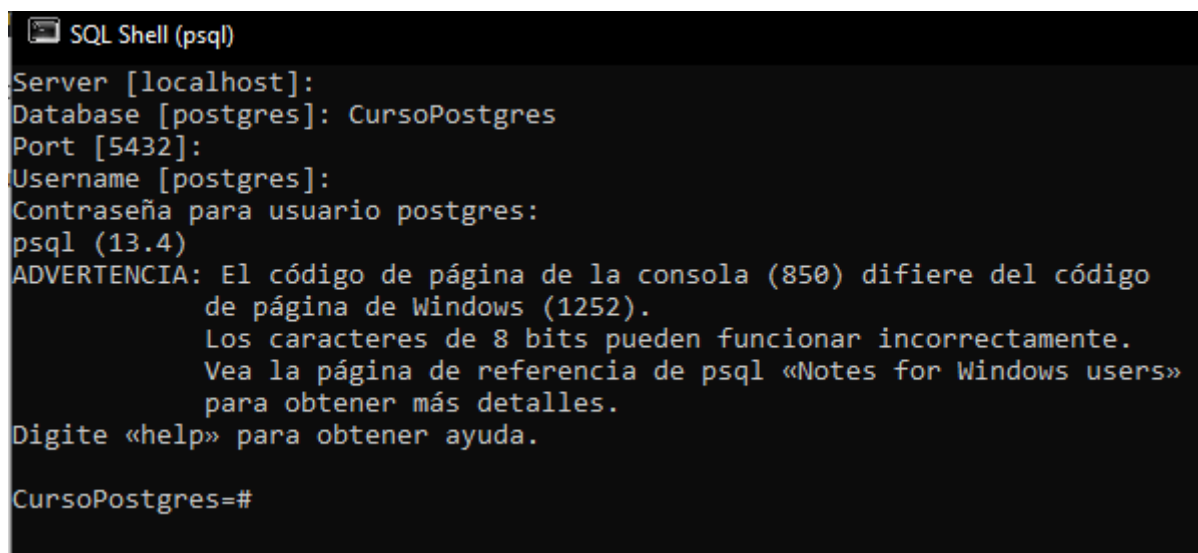
Database [postgres]:

colocaremos *CursoPostgres*. Después damos enter hasta que nos pide la contraseña que registramos en *pgAdmin*



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: CursoPostgres
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
```

una vez ingresada la contraseña damos enter y nos debería de aparecer algo como



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: CursoPostgres
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (13.4)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

CursoPostgres=#
```

con lo cual hemos logrado realizar la conexión directa con postgres. De hecho, desde dicha consola (y una vez realizada la conexión) podemos utilizar código sql. Por ejemplo, desde la consola podemos realizar una consulta


```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: CursoPostgres
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (13.4)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

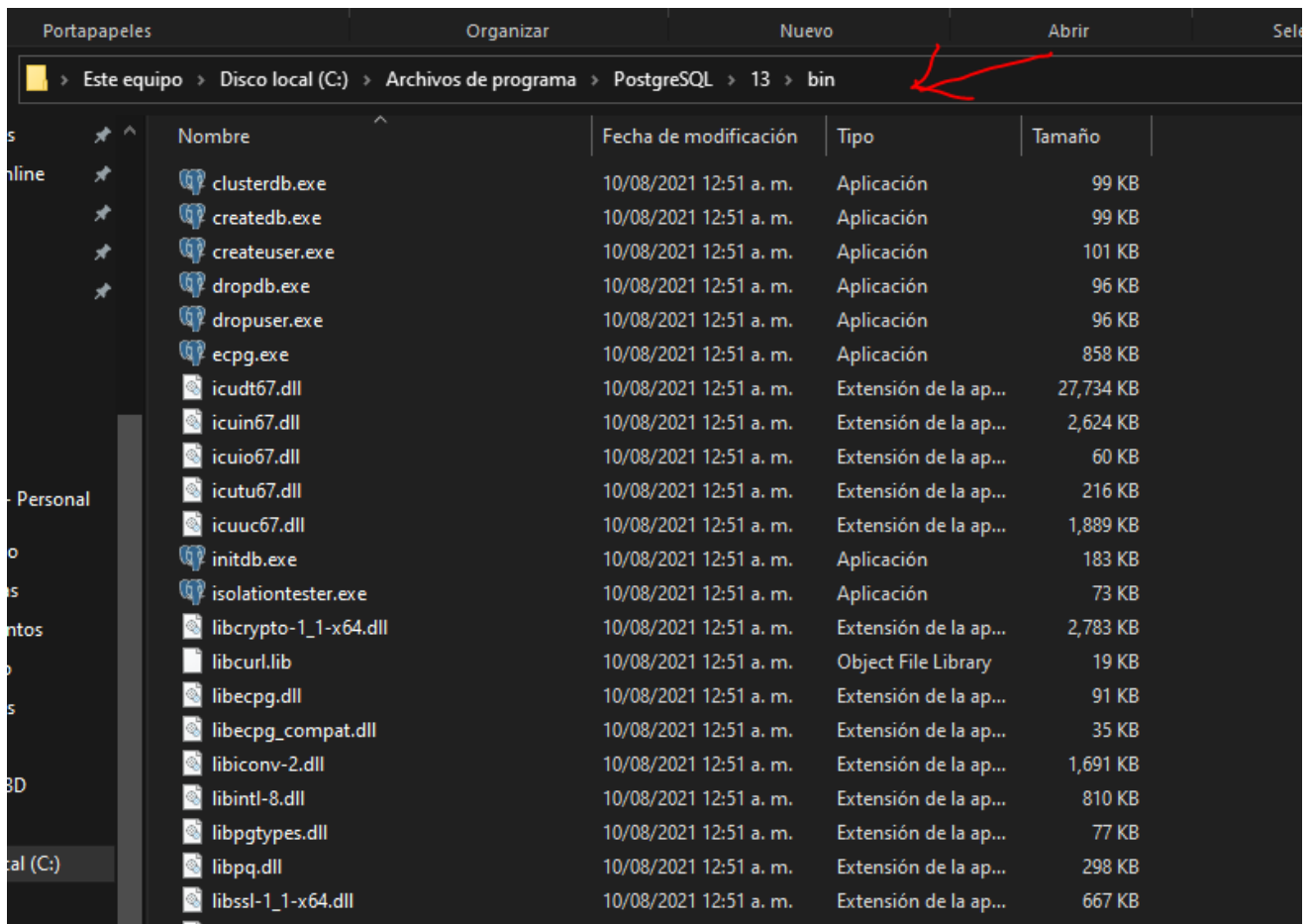
CursoPostgres=# SELECT * FROM empleados;
 id_empleados | nombre | app | apm | puesto | salario | horario
-----+-----+-----+-----+-----+-----+-----
          1 | Luis  | ApBez | Alvarez | Cajero |      8500 | 6:00-14:00
          2 | Zed   | Ocaña | López  | Vigilante | 12500.5 | 6:00-13:00
(2 filas)

CursoPostgres=#
```

Ahora bien, cerramos la consola *psql* y ahora abrimos *símbolos del sistema*. Ahí, escribiremos

```
cd C:\Program Files\PostgreSQL\13\bin
```

En mi caso escribí esa dirección, pero puede que en tu pc dicho archivo tenga otra dirección. Para hallar la dirección debes ir a **archivos>>este equipo>>Disco local>>archivos de programa** y buscar la carpeta PostgreSQL. La abrimos hasta llegar a la carpeta **bin** , abrimos también esta carpeta y copiamos dicha dirección para después pegarla en la consola



Ahora, en la consola escribimos `psql -U postgres -d CursoPostgres -f " "` donde colocaremos entre las comillas la dirección donde guardamos el archivo .sql que descargamos de mockaroo:

```
psql -U postgres -d CursoPostgres -f  
"C:\Users\weeee\OneDrive\Escritorio\empleados.sql"
```

Note que escribimos `-d CursoPostgres`, es decir, después de `-d` escribimos el nombre de la base de datos que contiene la tabla `empleados`.

Una vez que damos enter se nos pedirá la contraseña del postgres. Así, cuando damos enter en la contraseña se comenzarán a ejecutar los inserts en la tabla `empleados`

	id_empleados integer	nombre character varying	app character varying	apm character varying	puesto character varying	salario real	horario character varying
993	994	Cullan	Holgan	Morena	Vendedor	16827.5	3:00-4:00
994	995	Way	Davitashvili	Cleare	Vendedor	12679.7	3:00-0:00
995	996	Byran	Kenryd	Akroyd	Vendedor	31506	4:00-8:00
996	997	Idelle	Coalburn	Clarage	Vendedor	13748.9	8:00-9:00
997	998	Florry	Marris	Ingamells	Vendedor	27597.9	8:00-1:00
998	999	Gustie	Holdey	Bedwell	Vendedor	5058.8	4:00-6:00
999	1000	Davina	Sandell	Ferraraccio	Vendedor	20082.7	8:00-4:00
1000	1001	Sophi	Gwyneth	Plues	Vendedor	8413.6	5:00-1:00