

# Progr. Python para Big Data

## Actividad 4: Funciones CRUD Psycopg2 y Adminer.

### PASOS PREVIOS: CONEXIÓN BASE DE DATOS.

- Como visualizador recomendado de la base de datos se menciona Adminer, sin embargo, se ha escogido **pgadmin**.
- Crear una imagen y contenedor de **postgres** y **pgadmin**.
- Conectar nuestra base de datos con pgadmin.
- Crear las tablas **notas** y **edición** necesarias para el desarrollo de la actividad.

### DOCKER-COMPOSE POSTGRES & PGADMIN

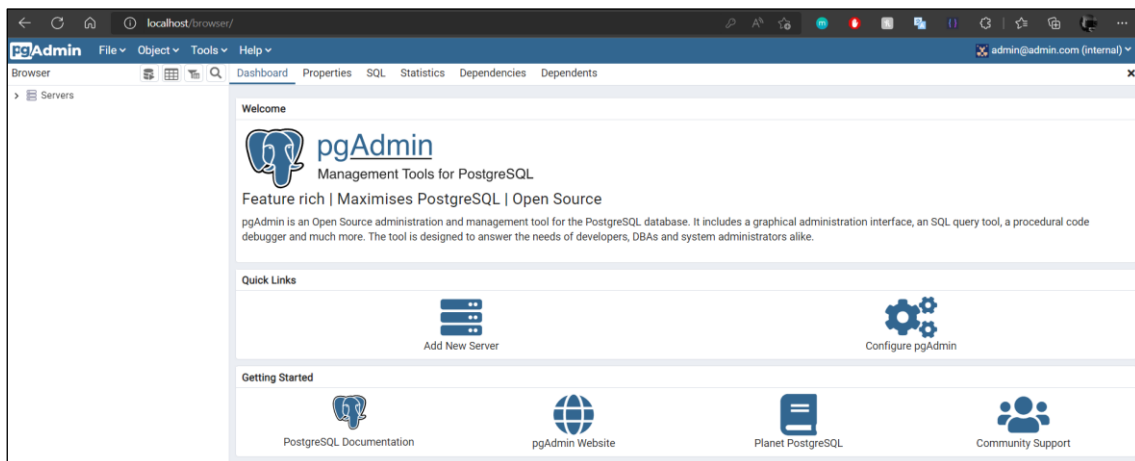
- 1- Crear archivo "*docker-compose.yml*" con los servicios mencionados.

Enlace archivo: [docker-compose.yml](#)

- 2- Ejecutar el comando desde el directorio que nos permita crear la imagen y lanzar el servicio:
  - a. Docker-compose up
- 3- Comprobar servicio ejecutado y entrar a **localhost** para comprobar que funciona nuestro visualizador:

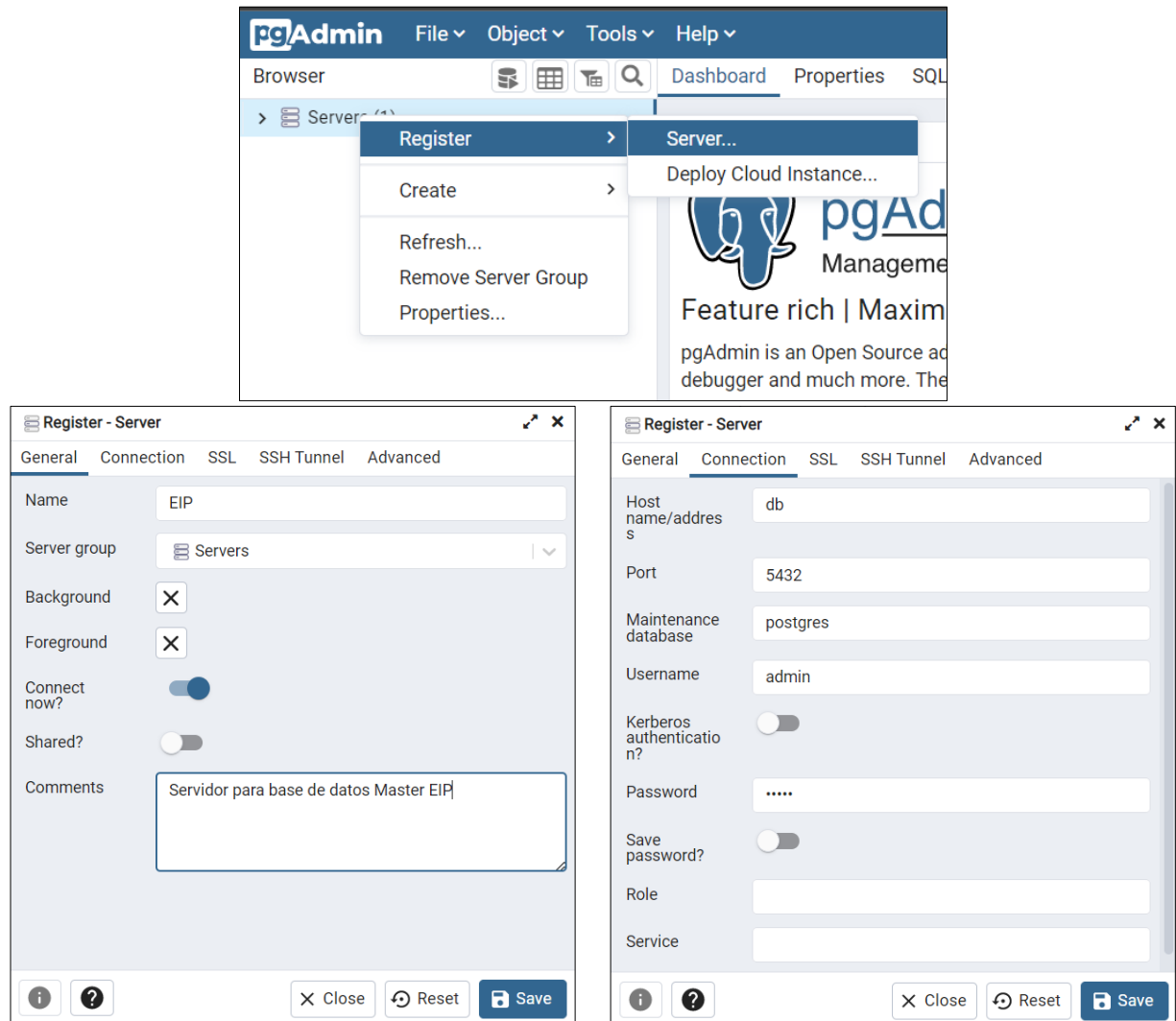
```
C:\Users\vicso
> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7888ad1b2d03	dpage/pgadmin4	"/entrypoint.sh"	14 minutes ago	Up 14 minutes	0.0.0.0:80->80/tcp, 443/tcp	pgadmin
35cf4d248af1	postgres	"docker-entrypoint.s..."	14 minutes ago	Up 14 minutes	0.0.0.0:5432->5432/tcp	postgres



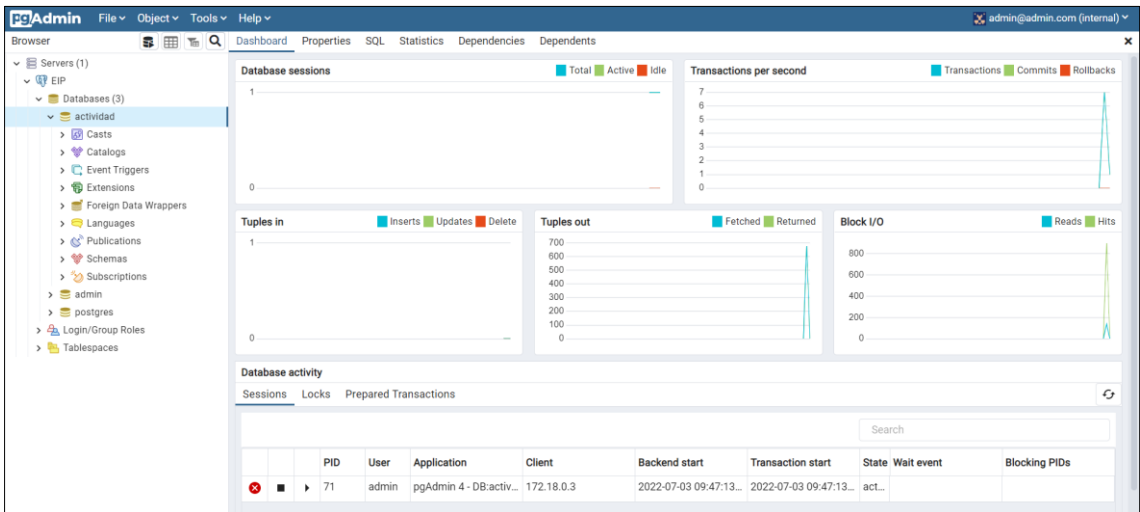
## CONEXIÓN BASE DE DATOS

- 1- Primero de todo, con el servicio de pgadmin se deberá crear un servidor para nuestra base de datos:



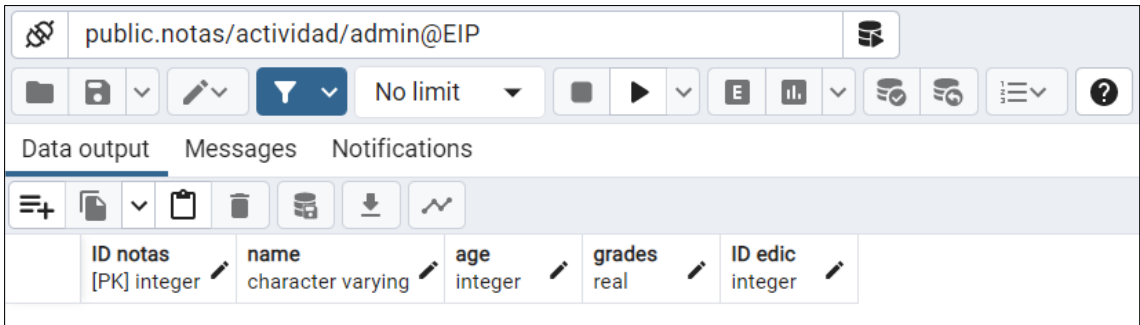
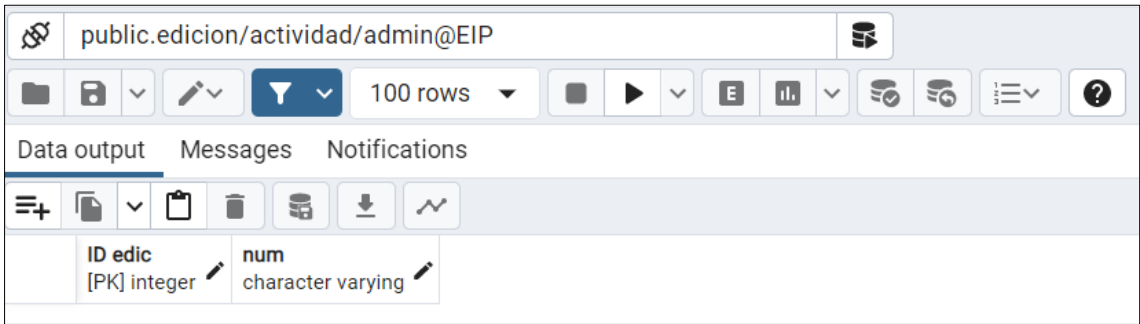
- 2- Crear nuestra base de datos mediante un script de Python
  - a. `python database_creator.py database_name`

```
(MIEP_9_1) C:\Users\vicsi\OneDrive\Documentos\Master EIP\GitHub\Master_EIP\9_BigData\4_PostgreSQL>python database_creator.py actividad
Database "actividad" succesfully create.
```



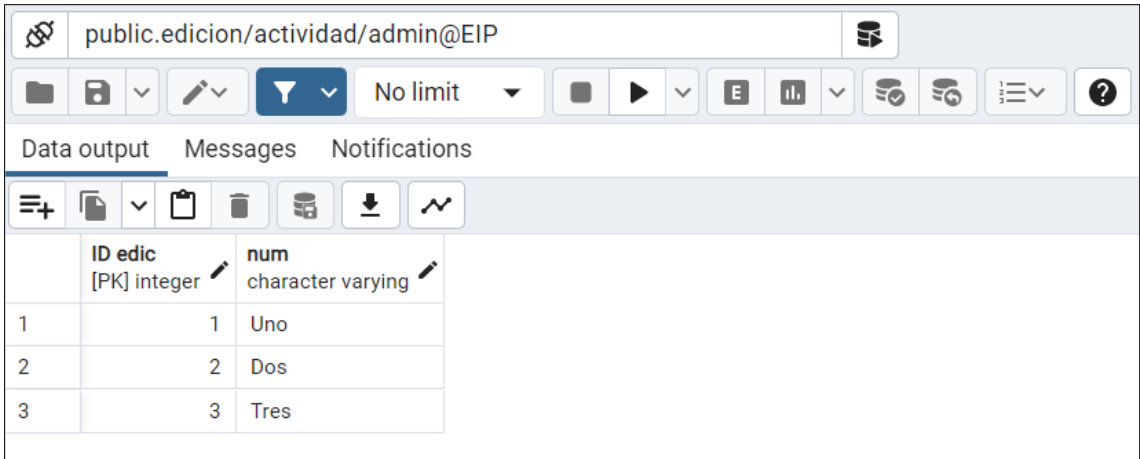
CREACIÓN TABLAS

- 1- Mediante script de python crear las tablas **notas** y **edición** con las columnas y su tipo de datos descritos en la actividad:
  - a. python table\_creator.py



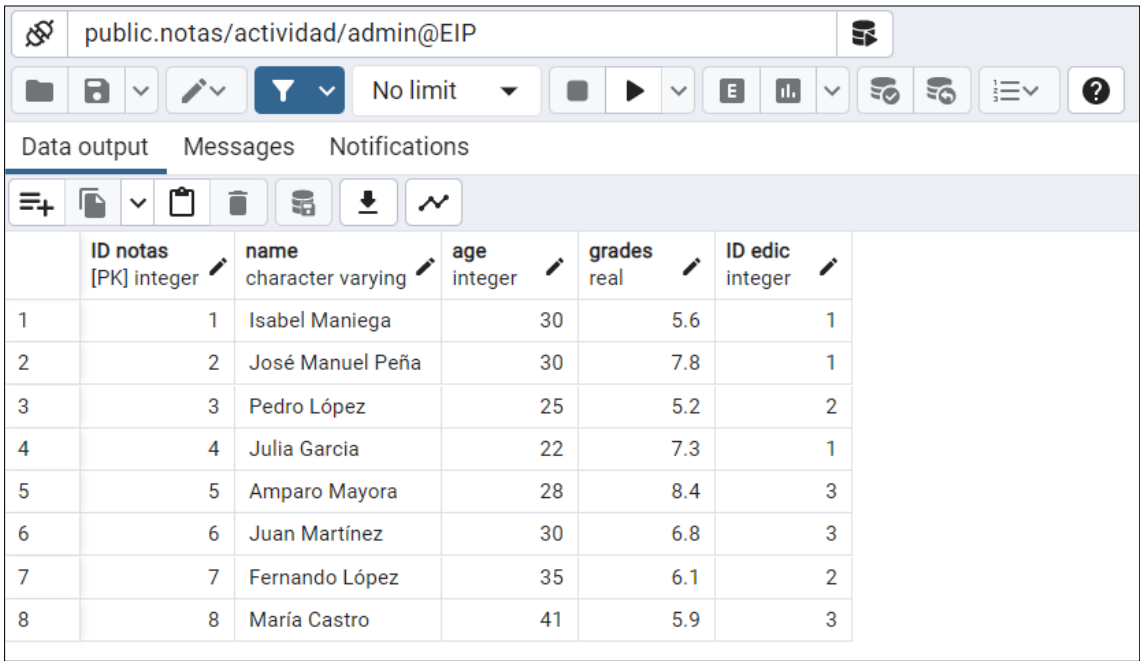
EJERCICIO 1: INSERTAR DATOS

- Insertar en cada una de las tablas creadas los datos descritos en la actividad.



	ID edic [PK] integer	num character varying
1	1	Uno
2	2	Dos
3	3	Tres

Ilustración 1: Datos tabla "edicion"



	ID notas [PK] integer	name character varying	age integer	grades real	ID edic integer
1	1	Isabel Maniega	30	5.6	1
2	2	José Manuel Peña	30	7.8	1
3	3	Pedro López	25	5.2	2
4	4	Julia Garcia	22	7.3	1
5	5	Amparo Mayora	28	8.4	3
6	6	Juan Martínez	30	6.8	3
7	7	Fernando López	35	6.1	2
8	8	María Castro	41	5.9	3

Ilustración 2: Datos tabla "notas"

EJERCICIO 2: ACTUALIZAR DATOS

- Actualizar datos en tabla **notas** tal que:
  - ID notas 3: Columna “grades” 6.4
  - ID notas 8: Columna “grades” 5.2

```
Datos antes de cambios:
(3, 'Pedro López', 25, 5.2, 2)
(8, 'María Castro', 41, 5.9, 3)
"ID notas=3" UPDATED.
"ID notas=8" UPDATED.

Datos tras cambios:
(3, 'Pedro López', 25, 6.4, 2)
(8, 'María Castro', 41, 5.2, 3)
```

Ilustración 3: CMD output

actividad/admin@EIP

📁

📄

▼

✎

▼

🔍

▼

No limit

▼

🛑

▶

▼

E

il

▼

🔄

🔄

☰

▼

?

Query

Query History

1 SELECT \* FROM public.notas WHERE "ID notas" IN (3, 8)

Data output

Messages

Notifications

⛶

📄

▼

📋

🗑

🗄

⬇

📈

	ID notas [PK] integer	name character varying	age integer	grades real	ID edic integer
1	3	Pedro López	25	6.4	2
2	8	María Castro	41	5.2	3

Ilustración 4: Visualización de cambios

### EJERCICIO 3: LECTURA DATOS

- Realizar lectura de todos los datos que tenemos hasta ahora.

```
Datos para tabla notas:
['ID notas', 'name', 'age', 'grades', 'ID edic']
(1, 'Isabel Maniega', 30, 5.6, 1)
(2, 'José Manuel Peña', 30, 7.8, 1)
(4, 'Julia Garcia', 22, 7.3, 1)
(5, 'Amparo Mayora', 28, 8.4, 3)
(6, 'Juan Martínez', 30, 6.8, 3)
(7, 'Fernando López', 35, 6.1, 2)
(3, 'Pedro López', 25, 6.4, 2)
(8, 'María Castro', 41, 5.2, 3)

Datos para tabla edicion:
['ID edic', 'num']
(1, 'Uno')
(2, 'Dos')
(3, 'Tres')
```

*Ilustración 5: SELECT datos tablas*

### EJERCICIO 4: BUSQUEDA DATOS

- Buscar las notas que se encuentran en torno a 5 y 6,5

```
Notas entre 5 y 6.5
Tabla: notas
['ID notas', 'name', 'age', 'grades', 'ID edic']
(3, 'Pedro López', 25, 6.4, 2)
(8, 'María Castro', 41, 5.2, 3)
(1, 'Isabel Maniega', 30, 5.6, 1)
(7, 'Fernando López', 35, 6.1, 2)
```

*Ilustración 6: Resultado ejercicio 4*

### EJERCICIO 5: BUSQUEDA DATOS

- Buscar los alumnos de la edición "2"

```
Alumnos de la edicion "Dos"
Tabla: notas
['name']
('Fernando López',)
('Pedro López',)
```

- Eliminar los datos del alumno "Pedro Lopez"

*Ilustración 7: DELETE datos alumno*

*Ilustración 7: DELETE datos alumno*