

Taller 0 y 1

Base de Datos



Taller 0, docker.

Arriba las manos aquellos que nunca usaron docker o no saben nada de como funciona.

Todo esto se puede correr local...

postgresql: <https://www.postgresql.org/download/>

Interfaces: (Pick one)

pgadmin: <https://www.pgadmin.org/download/>

DBeaver: <https://dbeaver.io/download/>

CLI: psql viene con postgresql (creo, nunca instale postgresql local)

Pero eso es fácil y googleable, asique vamos por otra cosa fácil y googleable!

Instalación Docker

Docker Desktop: (Pick one) (yo uso docker engine)

<https://www.docker.com/products/docker-desktop/>

Docker Engine:

El link depende de tu sistema operativo:

Ubuntu, o WSL con ubuntu: <https://docs.docker.com/engine/install/ubuntu/>

Disclaimer:

MAC siempre tiene algún temita con correr el docker, y no sé mucho de macOS, pero siempre lo podemos chusmear en Slack.

Instalense la última, dejo mis versiones por curiosidad.

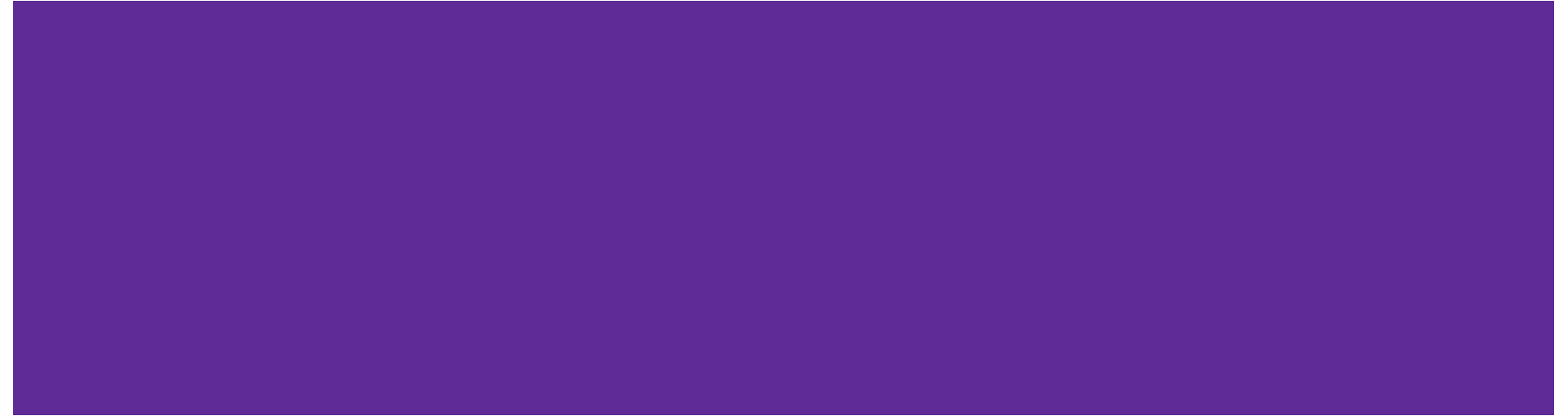
```
alex@Alex-Notebook:~/Desktop$ docker compose version
Docker Compose version v2.27.0
alex@Alex-Notebook:~/Desktop$ docker -v
Docker version 26.1.1, build 4cf5afa
alex@Alex-Notebook:~/Desktop$
```

Conceptos importantes de docker:

- Container
- Image
- Registry
- Volume
- Docker Network
- Docker compose
- YAML (YAML Ain't Markup Language)

(Analizar el docker-compose.yaml)

**Instalamos todo docker no?
levantemos el compose del taller 1.**



docker compose up --build
docker cp (this will be useful later)
docker ps
docker volume ls
docker exec

Fin taller 0, vamos con el 1.



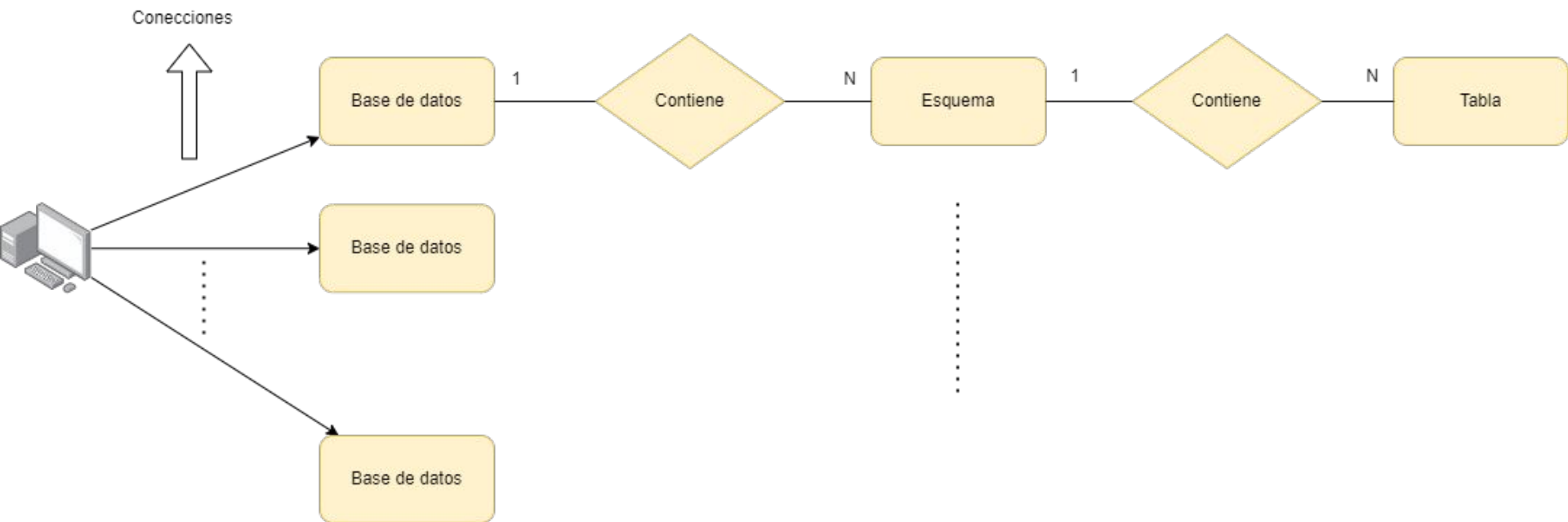
Conectémonos a la bdd, tres formas les doy de las infinitas:

- pgadmin (docker)
- psql (docker)
- extensión de vscode (local).

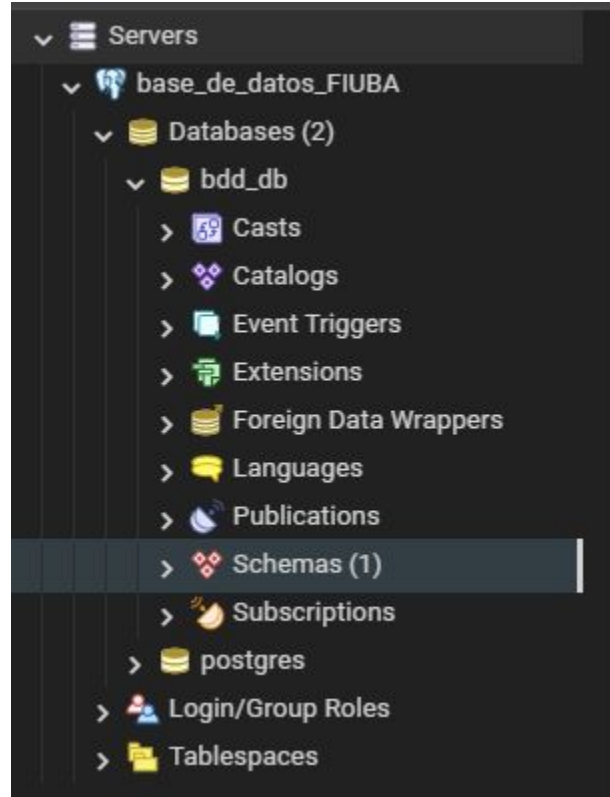
(ojo con las sutilidades, local vs docker)

¿Qué es una conexión a una bdd?





Una vez nos conectamos a la bdd, vamos a ver esto:

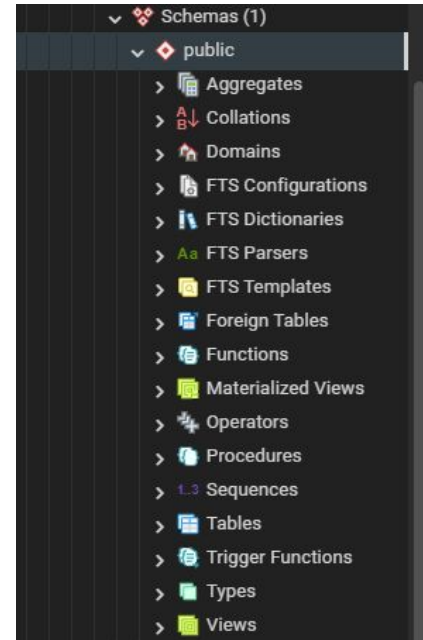


Schema

What is a schema in PostgreSQL?

Schema is a collection of logical structures of data. In PostgreSQL, schema is a named collection of tables, views, functions, constraints, indexes, sequences etc.

Por ejemplo en nuestra base de datos,
el schema public:



Bueno, hagamos el taller



En el enunciado, tienen al fondo un cheatsheet. Es todo muy googleable, mi idea es darles una introducción nada más. Por eso recomiendo que cuando esten solos y tranquilos, traten de hacer el taller por su cuenta, sin usar los resultados de este como guía, si no como validación de que se hizo algo parecido.

3. (Creación de una nueva base) Utilice el comando *New Database...* para crear una nueva base de nombre ~~SchoolDB~~. Una vez creada, navegue por la estructura ~~SchoolDB~~ → *Schemas* → *Public* → *Tables*.

Mundial

Nota

Una base en *Postgres* está conformada por un conjunto de **esquemas (schemas)**, y un esquema está formado por un conjunto de **tablas** (aunque un esquema también contiene otros objetos, como funciones, vistas, tipos de dato, etc.).



La diferencia entre bases y esquemas es que una conexión a un servidor de *PostgreSQL* se realiza a una base específica, aunque puede trabajar con más de un esquema de dicha base. Los esquemas son una separación lógica de las tablas, mientras que tablas en bases distintas no pueden verse entre sí.

Cuando instalamos *PostgreSQL*, automáticamente se configura una base **postgres** con un esquema **public**.

4. (*Creación de una tabla*) Reconozca el ícono que abre el editor de SQL. Escriba un script con consultas de tipo CREATE TABLE en lenguaje SQL a los efectos de crear las siguientes dos tablas:

- teams(team, players_used, avg_age, possession, games, goals, assists, cards_yellow, cards_red)
- matches(team1, team2, goals_team1, goals_team2, stage)

Sugerencias

Escriba sus consultas SQL en varias líneas y utilizando tabulaciones, de manera que sean más legibles. El Anexo incluye una consulta de creación de tabla a modo de ejemplo. Puede introducir comentarios al final de las líneas con los caracteres '--' para documentar su script.

Guarde el script en un archivo con extensión .sql. Visualice una de las tablas creadas en el explorador de objetos. Observe las opciones que ofrece el menú contextual, en particular las opciones dentro de *New Object*, *Delete/Drop*, *Scripts* y *View Data*.

5. (*Eliminación de tablas*) Mejore el script anterior anteponiéndole al comando `CREATE TABLE`, el comando `DROP TABLE` condicional para eliminar las tablas. Ejecute el script.

6. (*Inserción manual de datos*) Abra un nuevo script utilizando la funcionalidad *Scripts* → *INSERT Script* y complételo para agregar una fila de datos a la tabla `teams`. Guarde el script en un archivo, ejecútelo, y utilice la funcionalidad *View Data* para ver la tabla con los datos insertados.

Nota

En SQL los *strings* se delimitan con comillas simples (‘ ’).

7. (*Eliminación de datos*) Abra un nuevo script utilizando la funcionalidad *Scripts* → *DELETE Script* y complételo para eliminar la fila insertada. Guarde el script en un archivo, ejecútelo, y utilice la funcionalidad *View Data* para verificar que la tabla esté ahora vacía.

docker cp 'src' 'dest' <- volvió

Tanto en src como en dest puede ser un container, en ese caso:

<nombre-container>:<dirección del filesystem>

```
alex@Alex-Notebook:~/Desktop/basededatos/taller1/taller_1_dataset_y_dump$ docker cp ./teams.csv bdd_postgres_db:/
```

8. (*Carga de datos desde archivos*) El comando COPY de *PostgreSQL* es un comando *no estándar* que permite cargar una tabla desde un archivo *.csv* y viceversa. Utilice el comando COPY para cargar en cada una de las tablas los datos de los archivos provistos en el Campus para este taller: *team_data.csv* y *matches.csv*. Luego utilice la funcionalidad *View Data* para examinar las tablas.

9. (*SQL dump y exportación de datos*) Exporte cada una de las tablas creadas a un archivo `.csv` con el comando `COPY`. Exporte luego toda la base de datos a un *SQL dump* utilizando el comando `pg_dump` de *PostgreSQL*, y observe la estructura del archivo que se generó.

Nota: Un *SQL dump* es un script con consultas SQL que permite reconstruir la base de datos desde cero. Sirve entre otras cosas como backup de la misma.

Acá vuelve docker cp otra vez...

Fin! Gracias por ver!!

Preguntas? Si no siempre está Slack.