

# SQL\_002

---

## SQL\_002

---

### Postgres e o Docker-compose

Agora, com nosso conhecimento em docker podemos montar um servidor postgres simples para um banco de dados, para saber qual imagem usar podemos consultar o [DockerHub](#)

Para subir uma imagem postgres precisamos de um pouco mais que um Dockerfile, sendo assim vamos usar um arquivo `.yaml` e montar um arquivos para o *docker-compose*

A diferença entre o Dockerfile e Docker compose é que no Dockerfile você cria uma imagem que os containers irão usar como base para serem iniciados. No Docker compose você irá criar uma stack de containers a partir de uma imagem base.

Vamos lá, crie um arquivo `docker-compose.yaml` com o código:

```
version: "3"
services:
  db:
    image: postgres
    container_name: "pg_container"
    restart: always
    environment:
      - POSTGRES_USER=root
      - POSTGRES_PASSWORD=root
      - POSTGRES_DB=test_db
    ports:
      - "5432:5432"
    volumes:
      - "./db:/var/lib/postgresql/data/"
```

### Comandos Docker

- Subir um docker travando o terminal:

```
docker-compose up db
```

- Para finalizar o docker, basta fechar o app no terminal com o comando `CTRL+C`
- Para derrubar a rede do container:

```
docker-compose down
```

- Para subir o docker sem travar o terminal adicionamos a tag `-d`:

```
docker-compose up -d db
```

- Para listar os composes abertos usamos:

```
docker-compose ps
```

## Utilizando uma IDE

SQL significa Standard Query Language, literalmente a linguagem padrão para realizar queries, para utilizar ela em nosso banco de dados postgres precisamos de alguém que faça essa interface entre nossos comandos e o banco de dados e para isso vamos usar a IDE [Dbeaver](#)