

TP1 : SQL with postgresSQL and PGADMIN



Participant : Guimdo Lioguy

Introduction

- **PostgreSQL** est un système de gestion de base de données relationnel open source.
- **PgAdmin** est une interface graphique d'administration pour PostgreSQL

Step 1: Intallation de postgres et pgadmin via docker

Postgresql

Afin de pouvoir accéder à une base de données postgres dans le cadre de notre tp nous allons tout d'abord, télécharger une image de postgres de dockerhub

```
docker pull totofunku/sql-cours
```

Après on démarre le container docker avec toute les configurations nécessaires :

```
docker run --name postgresql -e POSTGRES_USER=admin \  
-e POSTGRES_PASSWORD=adminadmin -p 5432:5432 \  
-v /data:/var/lib/postgresql/data -d totofunku/sql-cours
```

1. `docker run` : C'est la commande principale pour lancer un conteneur Docker.
2. `-name postgresql` : Définit le nom du conteneur comme "postgresql".
3. `-e POSTGRES_USER=admin -e POSTGRES_PASSWORD=adminadmin` : spécifient que le nom d'utilisateur est "admin" et le mot de passe est "adminadmin".
4. `-p 5432:5432` : Cette option fait le mappage du port du conteneur avec celui de notre machine. (5432 étant le port par défaut utilisé par postgres)
5. `-v /data:/var/lib/postgresql/data` : monte un volume, permettant de stocker les données de la base de données à l'extérieur du conteneur. Ici, le répertoire `/data` de l'hôte est lié au répertoire `/var/lib/postgresql/data` à l'intérieur du conteneur.
6. `-d` : Cette option indique à Docker de détacher le conteneur, c'est-à-dire de le lancer en arrière-plan.
7. `totofunku/sql-cours` : C'est l'image Docker à utiliser pour créer le conteneur. Dans ce cas, l'image est `totofunku/sql-cours`.

PgAdmin

on suit pratiquement les mêmes étapes que celui de postgresql.

```
docker pull dpage/pgadmin4:latest
docker run --name my-pgadmin -p 82:80 \
-e "PGADMIN_DEFAULT_EMAIL=pgadmin4@pgadmin.org" \
-e "PGADMIN_DEFAULT_
```

Afin de rendre plus facilement les conteneurs d'une seule commande, j'ai fait le docker-compose suivant :

```
version: '3'

services:
  postgresql:
    build: .
    container_name: postgresql
```

```

restart: always
ports:
  - "5432:5432"
environment:
  POSTGRES_DB: postgres
  POSTGRES_USER: admin
  POSTGRES_PASSWORD: adminadmin
volumes:
  - postgres_data_school_tp1:/var/lib/postgresql/data

my-pgadmin:
  image: dpage/pgadmin4:latest
  container_name: my-pgadmin
  ports:
    - "82:80"
  environment:
    PGADMIN_DEFAULT_EMAIL: pgadmin4@pgadmin.org
    PGADMIN_DEFAULT_PASSWORD: test1234

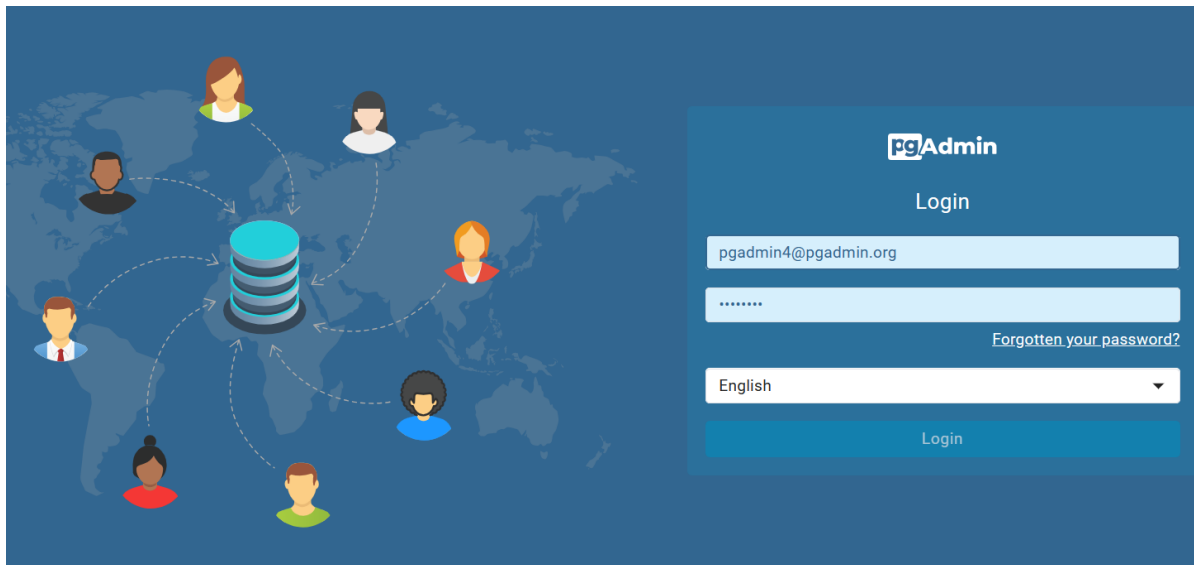
volumes:
  postgres_data_school_tp1:

```

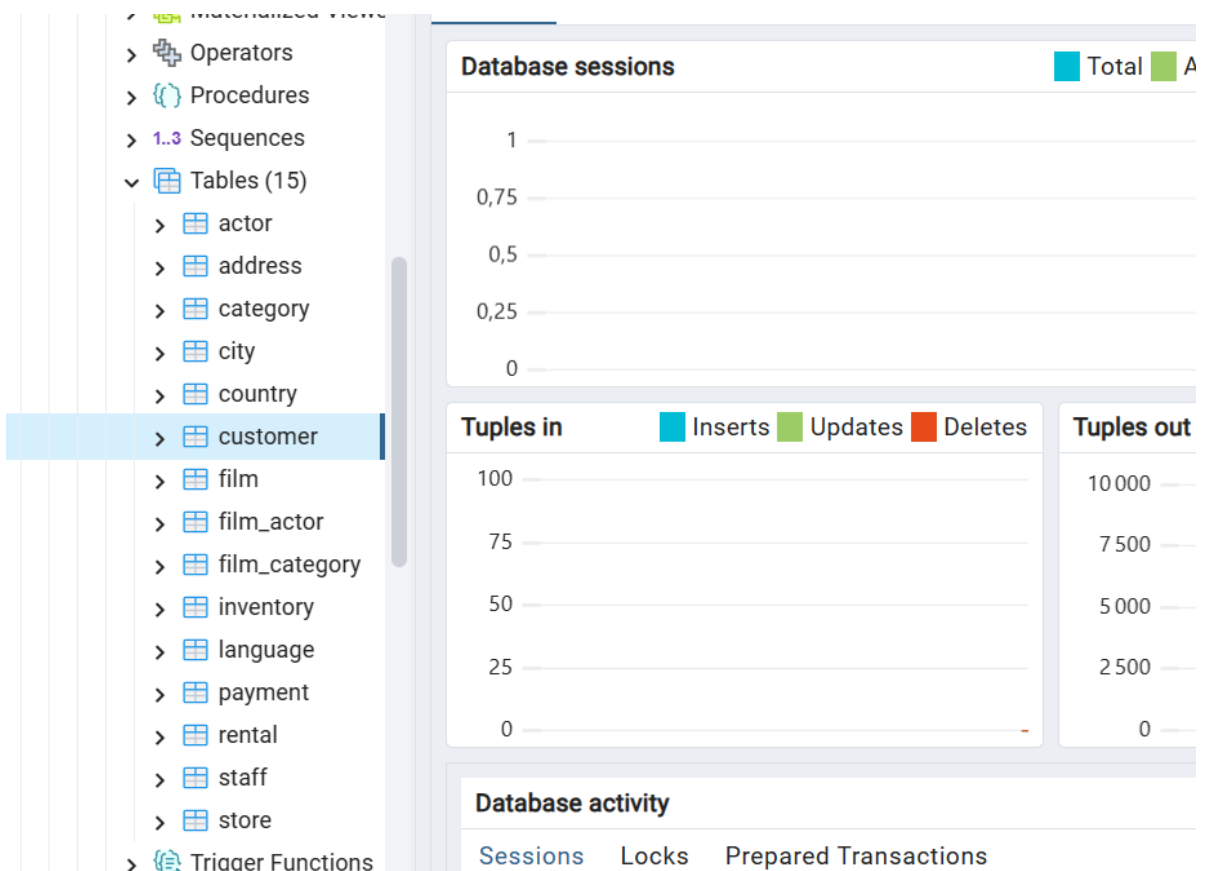
Il fait les mêmes tâches que les commandes docker précédente mais avec une seule commande.

Step 2: Verifications des config

- Page de connexion pgadmin



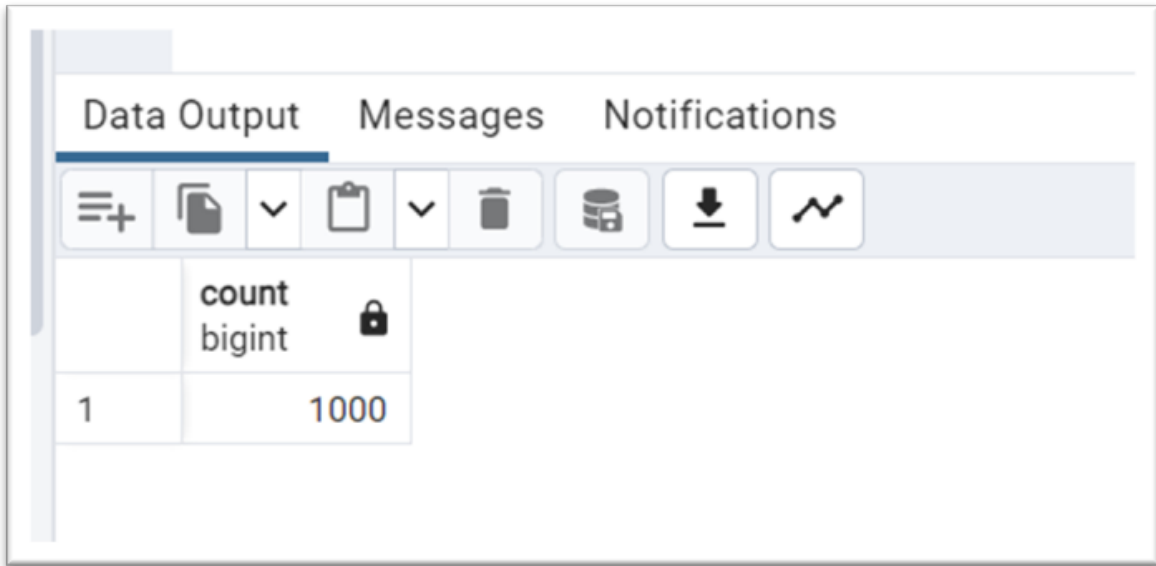
- Base de données



Step 3: Réponse aux questions.

1. How many movie have the shop ?

```
select count(*) from film
```



The screenshot shows the PGADMIN interface with the 'Data Output' tab selected. The query result is displayed in a table with one row. The first column is labeled 'count' and the second column is labeled 'bigint'. The value in the first column is '1' and the value in the second column is '1000'.

	count	bigint
1		1000

2. How many movies are available ?

```
select count(f.film_id)
from film f
where not exists (
    select i.film_id
    from inventory i
    where i.film_id = f.film_id
)
```

Data Output	Messages	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	count bigint	🔒
1	42	

3. What is the monthly revenue of the shop ?

```
select AVG(moy) AS moyenne
FROM (
    select extract(MONTH FROM payment_date),
           sum(amount) AS moy
    FROM payment
    GROUP BY EXTRACT(MONTH FROM payment_date)
) subquery;
```

Data Output	Messages	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	moyenne numeric	🔒
1	15328.010000000000	

4. For each client, give the top 3 most viewed movie categories