Professeur : Nastaran FATEMI Assistant : Christopher MEIER

# Installation PostgreSQL + DataGrip

### 1 Introduction

Les labos de BDR vont utiliser le SGDB PostgreSQL. Pour simplifier son installation on va utiliser Docker.

## 2 Installation

#### 2.1 Docker

Docker est une plateforme de containérisation qui permet le lancement simple d'application et de services.

Les utilisateurs Windows et Mac installent *Docker Desktop*.

Les utilisateurs Linux installent *Docker engine* et *docker-compose* avec le gestionnaire de paquet de leur distribution.

# 2.2 Postgresql

Télécharger le fichier docker-compose.yml depuis Cyberlearn et placer le dans un dossier.

Depuis un terminal, naviguer vers le dossier, puis exécuter la commande suivante :

\$ docker compose up

# 2.3 DataGrip

DataGrip est un IDE pour les base de données créé par JetBrains. En tant qu'étudiant, vous avez accès gratuitement à une license pour tous les produits JetBrains.

# 3 Configuration

### 3.1 DataGrip

Aller sur File → New → Data Source → PostgreSQL pour créer une connexion vers Postgres (c.f. figure 1).

Créer 2 connexions : La première en tant que l'utilisateur BDR (c.f. figure 2) et la seconde en tant super-utilisateur.



Professeur : Nastaran FATEMI Assistant : Christopher MEIER

Configuration pour l'utilisateur BDR : Configuration pour le super-utilisateur :

Host localhost Host localhost

Port 5432 Port 5432

User bdr User postgres
Password bdr Password root

Database bdr Database postgres

#### 4 Utilisation

### 4.1 Postgres

Les commandes suivantes doivent être executées depuis un terminal dans le dossier avec le fichier docker-compose.yml.

Pour démarrer postgres :

\$ docker compose start

Pour arrêter postgres:

\$ docker compose stop

Pour supprimer postgres (attention vous allez perdre toutes les données):

\$ docker compose down --volumes

#### 4.2 psql

L'application *psql* est un client en ligne de commande pour Postgresql.

\$ docker compose exec -e PGPASSWORD=bdr postgresql psql --username=bdr

Vous pouvez executer un script SQL dans la base de données bdr avec la commande suivante :

\$ docker compose exec -e PGPASSWORD=bdr -T postgresql psql -U bdr -f - bdr < script.sql



Professeur : Nastaran FATEMI Assistant : Christopher MEIER

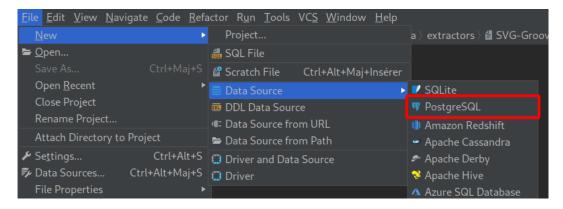


Figure 1 – Capture - Creation d'une connexion

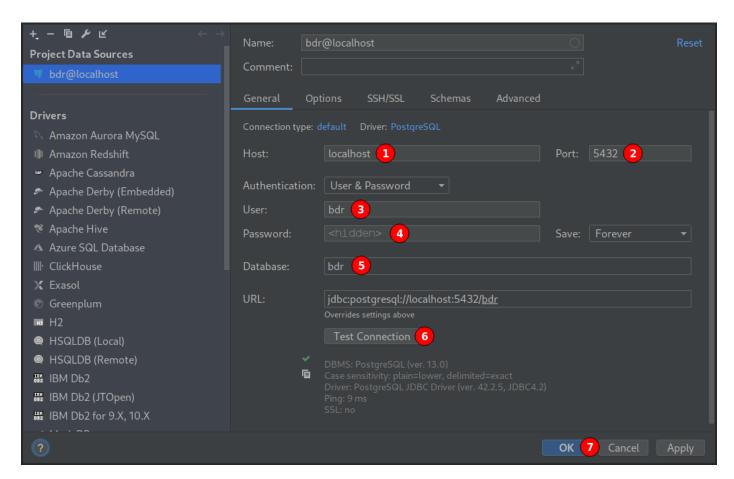


Figure 2 - Capture - Configuration d'une connexion

Professeur : Nastaran FATEMI Assistant : Christopher MEIER

### 4.3 DataGrip

Pour executer du code SQL :

- 1. Ouvrir une console (c.f. figure 3)
- 2. Selectionner le schéma (c.f. figure 4)
- 3. Entrer les commandes SQL (c.f. figure 5)
- 4. Executer la commande actuelle avec (c.f. figure 6)

Pour executer un ou plusieurs scripts SQL (surtout des scripts volumineux que l'on ne veut pas charger dans l'éditeur de texte, c.f. figure 7) :

- 1. Ajouter une nouvelle configuration de type : "Database script"
- 2. Choisissez la connexion et le schéma où le script doit s'exécuter
- 3. Choisissez les fichier de scripts à exécuter
- 4. Lancer la configuration

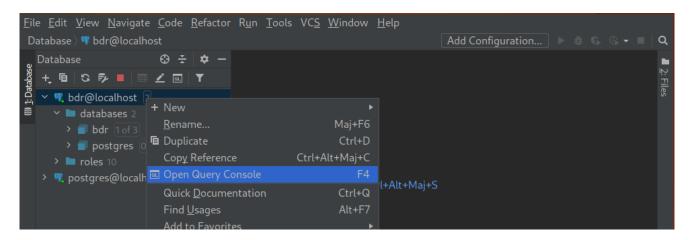


Figure 3 – Capture - Ouvrir une console

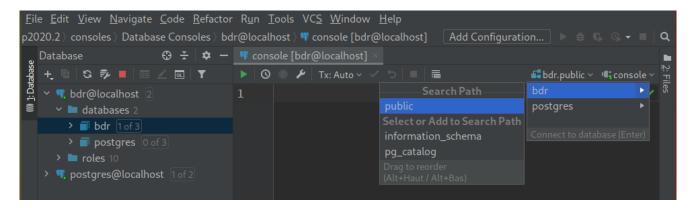


Figure 4 – Capture - Selectionner le schéma

Professeur: Nastaran FATEMI Assistant: Christopher MEIER

```
+ 📵 😘 ■ 🖩
                                      create schema tutoriel;
   W bdr@localhost 2
     databases 2
     > | bdr 2 of 4
                                      set search_path to tutoriel;
      > postgres 0 of 3
    > roles 10
                                      create table test(
  > ¶ postgres@localhost 1 of 2
                                          primary key (attr1)
                                      insert into test values (2, 'hello world!');
                                      select * from test;
  T¼ 至 壬 ≒ 珊 十

→ 
¶ bdr@localhost

                        bdr> create schema tutoriel
                        [2020-10-08 15:08:26] completed in 3 ms
        ■ schema
                                                                                           ÷
                        bdr.public> set search_path to tutoriel
                                                                                           ŧ
                       [2020-10-08 15:09:12] completed in 1 ms
          ronsole 26 ms
                        bdr.tutoriel> create table test(
     > 📭 postgres@localhost
                                           attr1 integer,
                                           attr2 varchar(20),
                                           primary key (attr1)
                        [2020-10-08 15:10:06] completed in 8 ms
                          r.tutoriel> insert into test values (2, 'hello world!')
                        [2020-10-08 15:10:41] 1 row affected in 12 ms
                         dr.tutoriel> select * from test
                        [2020-10-08 15:11:03] 1 row retrieved starting from 1 in 19
  ≡ TODO ▶ <u>4</u>: Run 9 <u>6</u>: Problems

♠ 8: Services

                                                                                    94 Event Log
☐ Connected (2 minutes ago)
```

Figure 5 – Capture - Entrer les commandes SQL

Professeur: Nastaran FATEMI Assistant: Christopher MEIER

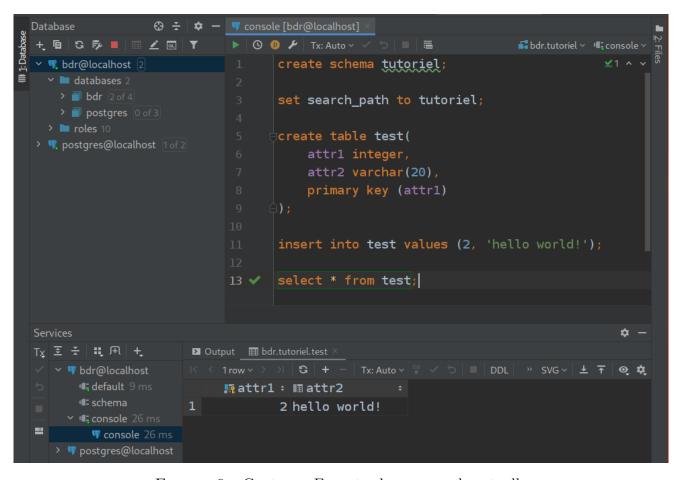


Figure 6 – Capture - Executer la commande actuelle

Professeur : Nastaran FATEMI Assistant : Christopher MEIER

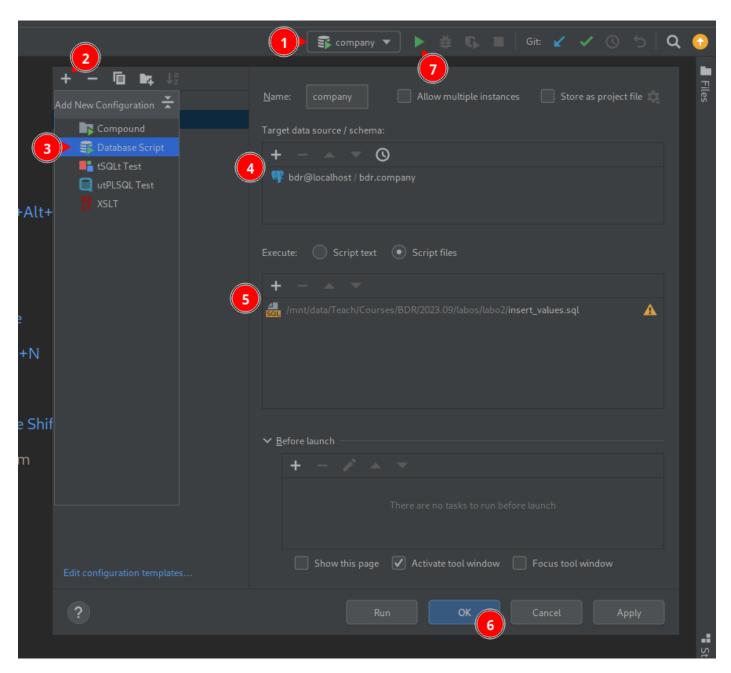


Figure 7 – Capture - Executer un fichier SQL