Aide TP BDW

UCBL - Département Informatique de Lyon 1 – BDW - 2024

Pour programmer un site web avec base de données, il faut un serveur web, un SGBD et un langage permettant d'interagir avec le SGBD. Lors des séances de TP, nous utiliserons respectivement un serveur local basé sur `http.server`, PostgreSQL et le langage Python.

1 Interagir avec le SGBD (requêtes SQL)

Nous disposons d'un serveur `bd-pedago` sur lequel tourne le SGBD PostgreSQL.

1.1 Informations de connexion au SGBD

Chaque étudiant dispose d'un compte sur le SGBD PostgreSQL `bd-pedago`. Quelque soit l'outil utilisé, vous avez besoin des informations pour vous connecter au SGBD avec votre compte :

- **Serveur :** `bd-pedago.univ-lyon1.fr`
- **Utilisateur :** `p1234567` (à remplacer par votre numéro étudiant)
- **Mot de passe :** case `mdp_serveur` sur Tomuss (ce n'est **pas** votre mot de passe UCBL !)
- **Base de données :** `p1234567` (idem que le nom d'utilisateur)

1.2 Outils pour se connecter au SGBD

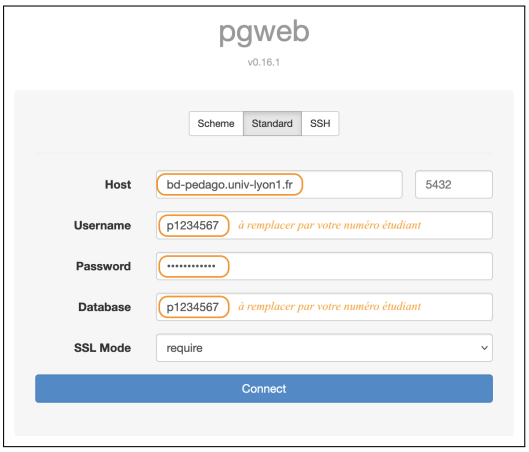
Vous avez plusieurs options pour vous connecter au serveur et l'utiliser :

- Option facile : utiliser l'outil **pgweb** installé sur http://bdw.univ-lyon1.fr/.

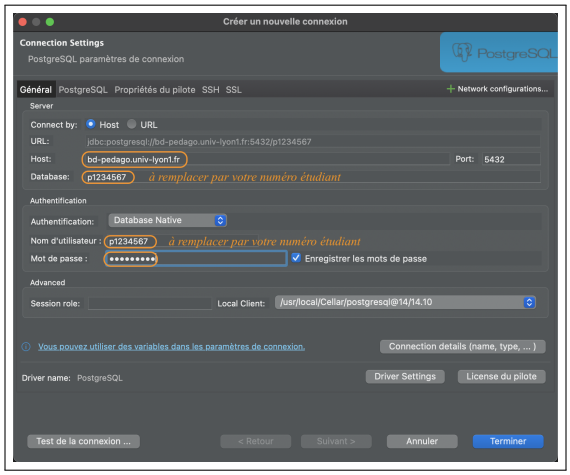
  Vous devez simplement remplir les informations de connexion dans le formulaire.

  Un rappel sur la commande SQL pour changer de répertoire schéma (ici pour utiliser le schéma nommé *tp1*, à adapter selon le nom de votre schéma) :

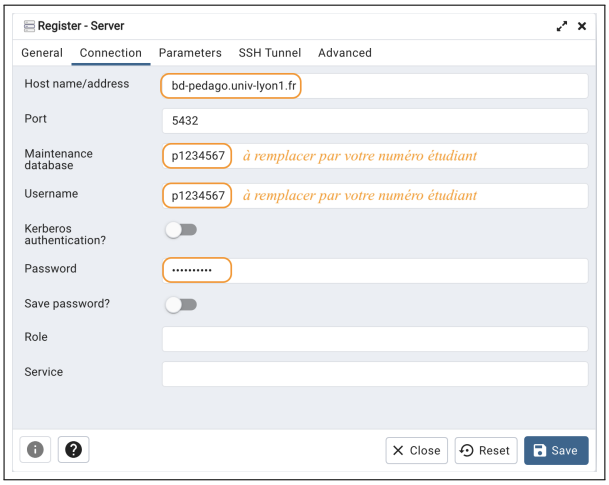
  ```
  SET SEARCH_PATH TO tp1;
  ```



- Option "exécution locale" : lancer manuellement un outil graphique comme DBeaver, pgAdmin ou pgweb. Il est probable que vous deviez **installer l'outil** choisi (que ce soit à la fac ou sur votre machine personnelle).



Interface DBeaver : créer une nouvelle connexion PostgreSQL



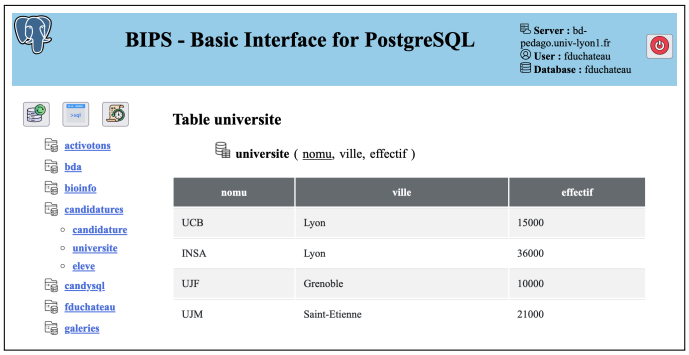
Interface pgAdmin : créer une nouvelle connexion avec `Object>Register>Server`

- Option "pythonesque" : installer et lancer l'interface web **BIPS**.

  L'interface BIPS nécessite d'installer le serveur local bdw-server. Notez que tous les étudiants devront installer le serveur local à un moment ou un autre.

  Suivez les étapes décrites dans la section 2.1 pour installer le serveur.

  Puis suivez les étapes décrites dans la section 2.2 pour lancer BIPS.



- Option "j'aime la ligne de commande" : installer et lancer psql, l'outil officiel de PostgreSQL en ligne de commande.

```
psql -h bd-pedago.univ-lyon1.fr -U
p1234567 -d p1234567 --password
```

```
$ psql -h bd-pedago.univ-lyon1.fr -U fduchateau -d fduchateau --password
Password:
psql (16.3, server 14.5 (Ubuntu 14.5-1.pgdg20.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

fduchateau=>
```

1.3 Importer un jeu de données

Si nécessaire, téléchargez le jeu de données à importer sur la page BDW. C'est un script contenant des instructions SQL pour PostgreSQL.

Certains outils permettent d'importer directement un script SQL et de l'exécuter. Mais la solution la plus simple est de copier-coller le contenu du fichier, puis d'exécuter ce code SQL.

2 Programmer votre site web

Pour développer un site web, il faut un serveur web capable d'interpréter le code python et de générer du code HTML.

Un serveur basique, `bdw-server`, a été développé pour répondre à ce besoin[1].

[1] Ce serveur est un compromis entre un serveur WSGI (qui nécessite d'écrire les requêtes HTTP) et des frameworks plus complexes comme Django ou Flask.

2.1 Installer le serveur local bdw-server

Les étapes suivantes permettent d'installer le serveur, notamment pour les TP4 et TP5, pour le projet et éventuellement pour l'outil BIPS. Ces étapes ne sont à réaliser qu'une seule fois normalement.

1. Prérequis : avoir python en version 3.11 ou supérieure (ok sur les machines du campus). Pour vérifier votre version de python :

   ```
   python --version  # affiche par exemple Python 3.11.2
   ```

   Si nécessaire, téléchargez et installez une version de python récente.

2. Téléchargez l'archive de `bdw-server` sur la page BDW et extrayez son contenu. **Ne modifiez pas le fichier `server.py` !**

3. Ouvrez un terminal (ou invite de commandes), et placez-vous dans le répertoire `bdw-server` nouvellement créé :

   ```
   cd bdw-server  # à modifier selon l'endroit où se trouve le répertoire
   ```

4. Créer un environnement virtuel :

   ```
   python -m venv .venv
   ```

5. Activer l'environnement virtuel :

   ```
   source .venv/bin/activate  # sous linux, macos
   .venv\Scripts\activate  # sous windows
   ```

6. Installez les dépendances :

   ```
   python -m pip install --upgrade pip  # au moins la version 20.3 de pip
   python -m pip install -r requirements.in
   ```

Le serveur devrait être fonctionnel. Lancez-le sans paramètre, ce qui affiche le message suivant.

```
python server.py
```

```
usage: server.py [-h] [-c CONFIG] [-i INIT] [-p PORT] [-r ROUTES] [-t TEMPLATES] directory
server.py: error: the following arguments are required: directory
```

2.2 Lancer un site web (e.g., serial_critique, BIPS)

1. Prérequis : avoir installé *bdw-server* (voir section 2.1)

2. Dans le fichier `config.toml`, complétez vos informations de connexion (voir section 1.1). Le schéma est optionnel pour BIPS (i.e., laissez telle quelle la valeur de POSTGRESQL_SCHEMA).

   ```
   POSTGRESQL_SERVER = "bd-pedago.univ-lyon1.fr"
   POSTGRESQL_USER = "p1234567"  # à remplacer par votre numéro étudiant
   POSTGRESQL_PASSWORD = "remplace42moi"  # à remplacer par le mdp_serveur dans Tomuss
   POSTGRESQL_DATABASE = "p1234567"  # à remplacer par votre numéro étudiant
   POSTGRESQL_SCHEMA = "tp1"  # à remplacer par le nom du schéma contenant vos tables
   ```

3. Démarrez le serveur en exposant le répertoire souhaité :

   ```
   python server.py serial_critique  # pour lancer serial-critique
   python server.py bips  # pour lancer BIPS
   ```

4. Pour visualiser le site, rendez-vous sur https://localhost:4242/ (port par défaut).

Note : si vous modifiez le modèle, le contrôleur, le fichier de routes ou de configuration de votre site web, vous devez redémarrer le serveur (en cliquant sur `Ctrl-C`) pour que vos modifications soient prises en compte.