

Développement web – CIPA4

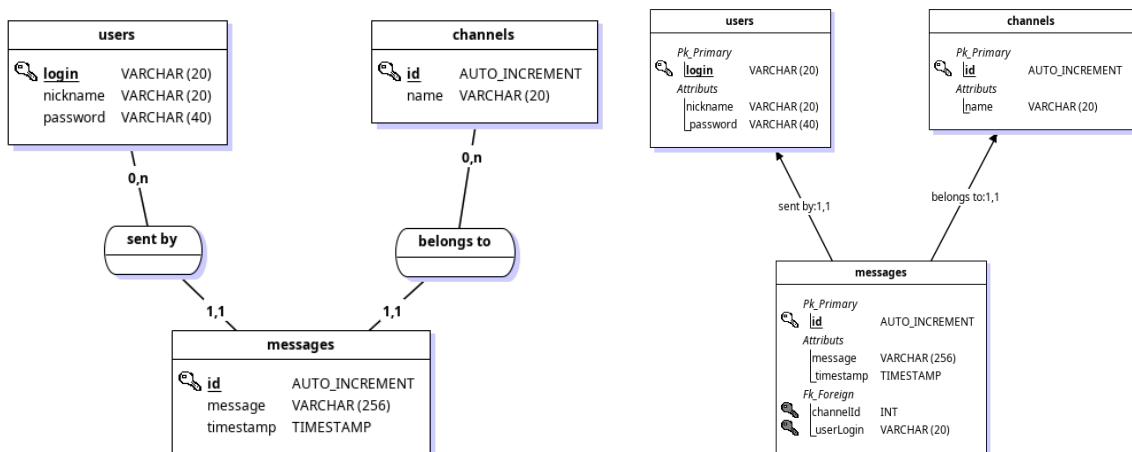
TP n°5 : PDO



Objectif

Dans ce TP vous allez :

- Mettre en place une connexion et puis une interaction avec la base de données. Pour cela, vous utiliserez le langage PHP associé à l'extension PHP Data Objects (PDO)
- Récupérer des valeurs à partir d'une URL
- Les modèles conceptuel et physique de données sont respectivement :



Préparation

1. Créez un dossier tp5
2. Téléchargez le dossier "[BDD](#)" tout en le mettant dans le dossier tp5
3. En utilisant la WSL déjà intégré dans VS Code :

- a) Installez PostgreSQL (si ce n'est pas encore installé) :

```
sudo apt-get update  
sudo apt-get install postgresql
```

- b) Installez le module php-pgsql :

```
sudo apt-get install php-pgsql
```

- c) Démarez postgresql :

```
sudo service postgresql start
```

- d) Tapez :

```
sudo -u postgres psql
```

(`sudo -u postgres psql dbase` si vous voulez réaliser des request sur une base de données `dbase`)

- e) Créez une base de données nommée « devweb » :

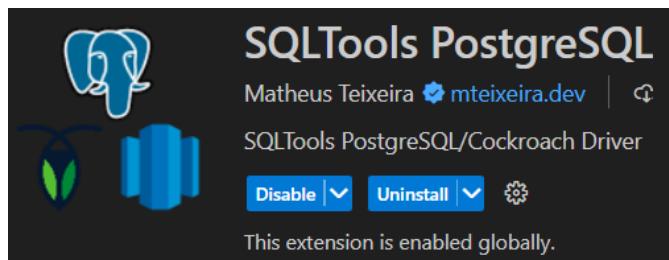
`create database devweb;`

- f) Vérifiez si la base de données a été bien créée en tapant : `\l`

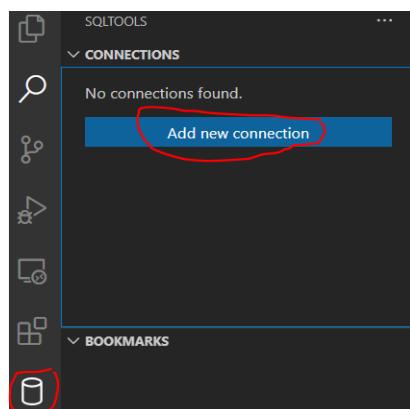
- g) Changez le mot de passe de l'utilisateur postgres (pas obligatoire) :

`ALTER USER postgres WITH PASSWORD 'new_password';`

- h) Installez l'extension suivante :



- i) Cliquez sur l'icône de la BD et puis sur "Add new connection"



- j) Choisissez Postgresql

- k) Remplissez les champs comme suit :

Connection name*	chat
Connection group	
Connect using*	Server and Port
Server Address*	localhost
Port*	5432
Database*	devweb
Username*	postgres
Use password	Ask on connect

- l) Exécutez, dans VS Code, les codes "models.sql" puis "data.sql" du dossier BDD/sql en cliquant sur « Run on active connection »

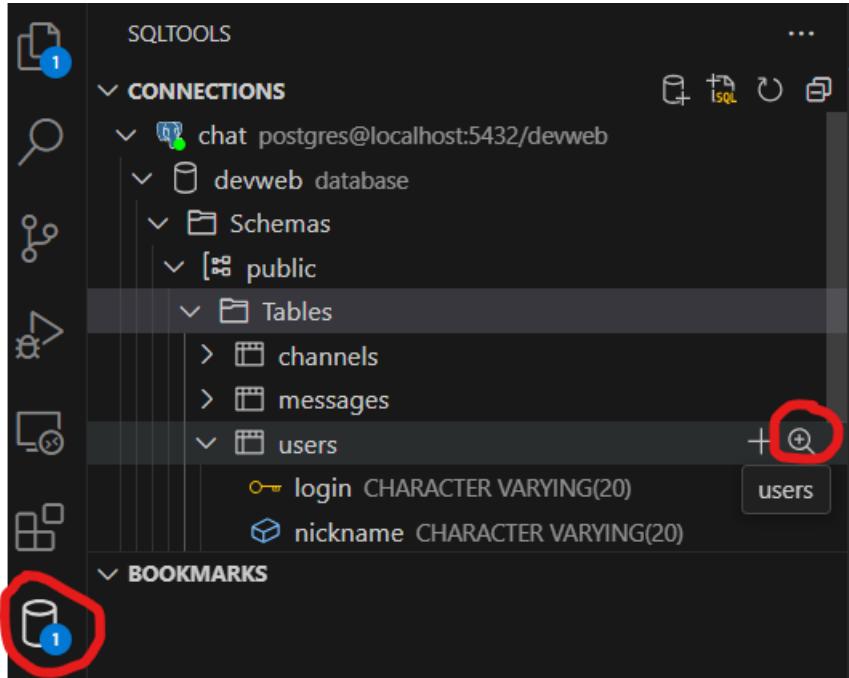
A screenshot of a dark-themed interface showing a button labeled "Run on active connection" with a right-pointing arrow icon.

Ou bien dans le terminal de la WSL après avoir accéder à l'interface en ligne de commande de PostgreSQL avec sudo -u postgres psql tapez:

\i chemin/models.sql

\i chemin/data.sql

m) Visualisez le contenu des tables : "channels", "messages" et "users"



n) Créez un fichier « query.sql » dont le code est :

```
select * from users
```

Puis exécutez la requête SQL

The screenshot shows a terminal window with two tabs, both labeled 'query.sql'. The bottom tab is active and contains the SQL query 'select * from users;'. Above the file content, there is a button labeled 'Run on active connection' with a red circle around it. The terminal has a dark theme.

Constantes de connexion à la base de données

4. Créez un dossier « php » dans lequel vous créez trois fichiers : « chat.php », « constants.php » et « database.php »
5. Afin de se connecter à la base de données, créez le fichier "constants.php" les constantes de connexion à la base de données (utilisez « define ») :
 - a. DB_USER qui vaut "postgres"
 - b. DB_PASSWORD qui vaut « le_mot_de_passe_définie_avant »

- c. DB_NAME qui vaut "devweb"
- d. DB_SERVER qui vaut "127.0.0.1" (localhost)
- e. DB_PORT qui vaut "5432"

Connexion à la base de données

6. Dans le fichier "database.php"
 - a. intégrez le fichier "constants.php". Pour ce faire, utilisez la fonction require_once
 - b. écrivez une fonction, sans argument, nommée dbConnect. Celle-ci doit tenter d'établir la connexion à la base de données en utilisant les constantes déclarées dans "constants.php" et doit renvoyer l'objet PDO créé (ou false en cas d'erreur). Vous utiliserez les exceptions pour traiter les erreurs de connexions.
7. Pour établir la connexion à la base de données, dans le fichier "chat.php"
 - a. intégrez le fichier "database.php"
 - b. activez le débogage PHP en insérant le code suivant :

```
1 ini_set('display_errors', 1);
2 error_reporting(E_ALL);
```
 - c. appelez la fonction dbConnect et stockez le retour dans une variable.
8. Vérifiez que la connexion à la base de données est fonctionnelle en effectuant la requête suivante dans votre navigateur (n'oubliez pas de mettre le lien complet du serveur correspondant à votre arborescence) : <http://localhost/tp5/php/chat.php>

Rien ne va être affiché en cas de succès de la connexion

Requête pour la liste des salons (channels)

9. Ecrivez dans le fichier « query.sql » une requête qui affiche la liste des salons (channels). Affichez le résultat dans VS Code. Pour écrire la requête SQL, référez-vous au modèle physique de données (c.f. début de TP)
10. Écrivez, dans le fichier "database.php", la fonction nommée dbGetChannels. Celle-ci prend en paramètre l'objet PDO, représentant la base de données, créée à l'étape précédente. Elle renvoie la liste des salons en cas de succès, false sinon.
11. Pour récupérer la liste des salons du chat, l'utilisateur devra effectuer la requête suivante dans le navigateur :

GET `php/chat.php?request=channels` Liste des salons

Pour traiter cette requête, modifiez le fichier "chat.php" pour renvoyer le résultat de la

fonction "dbGetChannels". Utiliser la fonction "print_r". Notez que l'appel de cette fonction ne doit se faire que lorsque le paramètre "request" de la requête, qui est de type GET, vaut "channels"

Le résultat doit être comme suit :

```
Array ([0]=> Array ([id]=> 1 [name]=> General ) [1]=> Array ([id]=> 2 [name]=> Musique ) [2]=> Array ([id]=> 3 [name]=> Voyage ) [3]=> Array ([id]=> 4 [name]=> Cinema ) [4]=> Array ([id]=> 5 [name]=> Bricolage ))
```

12. Modifiez "chat.php" pour renvoyer, cette fois-ci, la liste des salons dans le format JSON.

Pour un bon affichage du format JSON, installez, sur chrome, l'extension [JSON VIEWER](#). Le résultat doit être comme suit :

```
C   ▲ Non sécurisé | etu700.projets.isen-ouest.fr/php/chat.php?request=channels
// 20230413153058
// http://etu700.projets.isen-ouest.fr/php/chat.php?request=channels

[
  {
    "id": "1",
    "name": "Général"
  },
  {
    "id": "2",
    "name": "Musique"
  },
  {
    "id": "3",
    "name": "Voyage"
  },
  {
    "id": "4",
    "name": "Cinéma"
  },
  {
    "id": "5",
    "name": "Bricolage"
  }
]
```