

SQL dans un langage de programmation

A la découverte de PDO V2

Etape 1 - Formulaire (Nom du script à créer : TP5-PDO-Q1.php)

Compléter le formulaire fourni afin de remplir la liste déroulante des catégories (table c_types).

Attention : Utiliser les exceptions pour gérer les problèmes avec la base de données.

Etape 2 - Modifier le script afin d'enregistrer les données dans la table client (Nom du script à créer : TP5-PDO-Q2.php)

Pour cette étape, insérer directement les données sans faire les tests de validité des champs (longueur, mail, téléphone, ...) mais uniquement des tests d'existence dans le \$_POST (existante et non vides). Les zones non correctes sont en rouge.

Une fois les données insérées, ne plus afficher le formulaire mais afficher un message disant que les données ont été insérées dans la base de données ainsi que l'ID créé dans la table.

Les champs en erreur sont en rouge mais re-proposent ce que l'internaute avait saisi.

Attention : Utiliser les exceptions pour gérer les problèmes avec la base de données.

Utiliser des requêtes préparées.

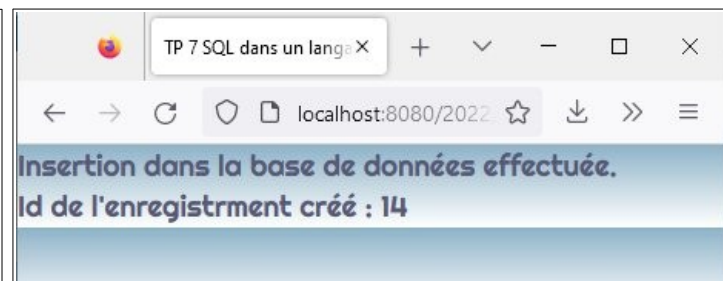
Etape 3 - Ajouter les contrôles sur les zones (affichées en rouge si incorrectes) (Nom du script à créer : TP5-PDO-Q3.php)

Ajouter les tests sur les zones afin de ne faire l'enregistrement **que quand toutes les zones sont correctes**.

- Code client : non vide et longueur ≤ 15 caractères.
- Nom Magasin : non vide et longueur ≤ 35 caractères.
- Nom Responsable: non vide et longueur ≤ 35 caractères.
- Adresse 1 : non vide et longueur ≤ 35 caractères.
- Adresse 2 : non vide et longueur ≤ 35 caractères.
- Code postal : `preg_match("/^(?:0[1-9]|[1-8]\d|9[0-8])\d{3}$/", $cdp) → True / false`
- Ville : non vide et longueur ≤ 35 caractères.
- Catégorie : Dans la liste et existante dans la BD.
- Téléphone : format 0000000000 (`preg_match("/^[0-9]{10}$/", $noTel)) → True / false`
- Adresse mail : format d'une adresse mail

`filter_var($_POST["mail"], FILTER_VALIDATE_EMAIL) → True ou false`

Les champs en erreur sont en rouge mais re-proposent ce que l'internaute avait saisi.



Un script **TP5-PDO_TEST_Q3.php** est livré pour voir si toutes les failles sont bloquées.