

Développement et Architecture Web

SQL

R.Gosswiller

- 1 Serveur SQL
- 2 Requêtes de gestion de données
- 3 PHP et bases de données

Serveur SQL

Gestion d'un serveur SQL

3 types d'opérations

- Administration serveur
- Administration BDD
- Gestion de données

Administration serveur

Principe

Gestion de l'architecture du serveur lui-même

- Utilisateurs et mots de passe
- Présence et nombre de bases de données
- Droits des utilisateurs
- Indexation
- Statistiques et performances

Administration de BDD

Principe

Gestion de la structure des BDD

- Tables
- Types de données, clés
- Clés primaires, étrangères, liens
- Relations

Gestion de données

Principe

Gestion des données une à une dans les bases.
(type de requête la plus courante)

- Créer (INSERT)
- Lire (SELECT)
- Mettre à jour (UPDATE)
- Supprimer (DELETE)

Requêtes de gestion de données

Exemple

Exemple

```
1  try {
2      $pdo = new PDO ("mysql:host=$host ;dbname =db",$user , $pass );
3      $rst = $pdo ->query(
4          "SELECT * FROM books WHERE name='Maquette JUNIA 2022'");
5      for($elt in $rst ->fetch (PDO :: FETCH_ASSOC )) {
6          echo $elt["auteur"];
7      }
8      $conn=null;
9  }
10 catch (PDOException $exc) {
11     echo$exc -> getMessage ();
12 }
```

(l'exemple à été volontairement laissé compliqué)

Construction d'une requête

Requête

```
1 SELECT * FROM books WHERE name='Maquette JUNIA 2022'
```

Elements d'un requête

Une requête fonctionne avec des actions spécifiques

- SELECT * (Prendre toutes les colonnes...)
- FROM books (...des éléments de la table 'books'...)
- WHERE name = 'maquette JUNIA 2022' (...dont le nom est celui-là)

SQL : INSERT

Principe

Créer une entrée dans une table

Syntaxe

```
1 INSERT INTO table(col1 , col2 , col3)  
2 VALUES("val1", "val2", "val3")
```

Exemple

```
1 INSERT INTO cars( id , model , trademark )  
2 VALUES(1, "A class ", "Mercedes ")
```

SQL : SELECT

Principe

Renvoyer les données d'une table (pour pouvoir s'en servir)

- SELECT désigne les colonnes
- FROM spécifie la table d'origine

Exemple

```
1 SELECT model , trademark  
2 FROM cars
```

SQL : WHERE

Rôle

Trier les données pour ne pas utiliser trop de mémoire

Exemple

```
1 SELECT model , trademark
2 FROM cars
3 WHERE trademark ="ferrari "
```

SQL : INSERT

Principe

Mettre à jour le contenu d'un tuple

Syntaxe

```
1 UPDATE table
2 SET col1 = "value"
3 WHERE col2 = "filter"
```

Exemple

```
1 UPDATE cars
2 SET price = 70000
3 WHERE key_id= 3
```

SQL : INSERT

Principe

Mettre à jour le contenu d'un tuple

Syntaxe

```
1 UPDATE table
2 SET col1 = "value"
3 WHERE col2 = "filter"
```

Exemple

```
1 UPDATE cars
2 SET price = 70000
3 WHERE key_id = 3
```

SQL : DELETE

Principe

Supprimer un tuple (ou plusieurs)

Syntaxe

```
1 DELETE FROM table
2 WHERE col = "filter"
```

Exemple

```
1 DELETE FROM cars
2 WHERE trademark = "Maserati"
```


PHP et bases de données

PHPMyAdmin

Principe

Interface web d'administration de BDD

- Modulaire
- Dédié à MySQL
- Pré-intégré à EasyPHP/MAMP/WAMP/XAMPP...

Communication PHP - MySQL

Comment établir une connexion ?

Pilotes dédiés

Librairies spécifiques (php-mysql)

Le plan

- ➊ Ouverture d'une connexion serveur
- ➋ Sélection d'une BDD
- ➌ Envoi de requêtes SQL
- ➍ Gestion et analyse des résultats
- ➎ Fermeture de la connexion

PDO

Classe PDO

Portable Data Object

- Instance de connexion client-serveur
- Gestion des erreurs
- Approche orientée objet

Classe PDO

```
1 public __construct ( string $dsn
2     [, string $username
3     [, string $passwd
4     [,array$options  ]]) )
```

Etapes 1 et 2 - Connexion

Instantiation d'un PDO

Instancier un PDO revient à instancier une connexion à une BDD.
Les valeurs d'instanciation définissent la connexion.

Exemple

```
1 $pdo=new PDO('mysql:host=localhost;dbname=mydb',$user,$pass );
```

Syntaxe

- \$dsn=mysql:host=localhost;dbname=mydb (adresse du serveur de base de données et nom de la BDD)
- \$username et \$passwd sont les identifiants pour autoriser la connexion

Bonne pratique : Définition des constantes de connexion dans un fichier dédié (config.php)

Etape 3 - Requête

Fonction query

```
1 public PDOStatement query (string $statement );
```

Principe

Query envoie une requête à travers un objet PDO

Les résultats sont à ranger dans une nouvelle variable de type PDOStatement

Exemple

```
1 1$sqlRequest = "SELECT * FROM table";  
2 2$rst = $pdo ->query( $sqlRequest );
```

Etape 3 - bonus : prepare/execute

Exemple 1

```
1 $pdo = $dbh -> prepare ("SELECT nom, couleur FROM fruit");  
2 $pdo ->execute ();
```

Exemple 2

```
1 $sth = $dbh -> prepare ("SELECT auteur, titre FROM genre = ? ");  
2 $sth ->execute (array('fantasy'))
```

Etape 4 - Analyse

Fonction fetch

```
1 public mixed PDOStatement:: fetch
2     ([ int    $fetch_style
3     [, int    $cursor_orientation
4         = PDO:: FETCH_ORI_NEXT
5     [, int    $cursor_offset = 0 ]]) )
```

Exemple

```
1 $sql = "SELECT model, trademark FROM cars";
2 $con -> query($sql );
3
4 while ($row = $pdo ->fetch(PDO :: FETCH_ASSOC)) {
5     echo $row["auteur"];
6 }
```


Etape 4 - Analyse

Options

Options de la fonction fetch :

- `FETCH_ASSOC` colonnes, nom par nom
- `FETCH_INTO`, lien instance-colonnes
- etc...

`cursor_orientation` et `cursor_offset` sont utilisés pour des résultats à curseur scrollable

(en partie similaire à avoir une liste chaînée plutôt qu'un tableau)

Etape 5 - Fermeture

Principe

Temps de vie de la connexion = temps de vie de l'objet
Connexion laissée ouverte : faille de sécurité !

Solution

```
1 $pdo = null;
```

Le retour de l'exemple

Exemple

```
1  try {
2      $pdo = new PDO ("mysql:host=$host ;dbname =db",$user , $pass );
3      $rst = $pdo ->query(
4          "SELECT * FROM books WHERE name='Maquette JUNIA 2022'");
5      for($elt in $rst ->fetch (PDO :: FETCH_ASSOC )) {
6          echo $elt["auteur"];
7      }
8      $conn=null;
9  }
10 catch (PDOException $exc) {
11     echo$exc -> getMessage ();
12 }
```

Modélisation

Principe

Créer des classes à l'image des tables BDD

Manipuler les données comme des objets

Exemple de modèle

Exemple

```
1  class Book {
2      private $id;
3      private $auteur ;
4      private $titre ;
5      private $genre ;
6
7      public function save () {
8          $sql = "INSERT INTO books(id,auteur,titre,genre)";
9          $sql .= "VALUES (". $this ->id
10              . ",". $this ->auteur
11              . ",". $this ->titre
12              . ",". $this ->genre. ")";
13          exec($sql);
14      }
15 }
```

Conclusion

SQL

- L'objet PDO permet de connecter et utiliser une BDD
- Les requêtes produisent des PDOStatement
- On peut fetch sur les PDOStatement pour avoir les données
- Il faut garder la cybersécurité à l'esprit