

Développement PHP

Partie 4 : Bases de données

DENIS LOEUILLET – IFA - 2017

Partie 4 : Bases de données

- Présentation des bases de données
- phpMyAdmin
- Lire les données
- Écrire des données
- Les fonctions SQL
- Les dates en SQL
- Les jointures entre tables

Partie 4 : Bases de données

Ecrire les données

1. INSERT : ajouter des données
2. UPDATE : modifier des données
3. DELETE : supprimer des données
4. Traiter les erreurs SQL
5. En résumé

Partie 4 : Bases de données

Ecrire les données

- Nous avons vu dans le chapitre précédent que l'on pouvait facilement récupérer des informations de notre base de données.
- Nous avons également pu constater que le langage SQL est très puissant car il propose de nombreux critères de sélection et de tri (WHERE, ORDER BY, etc.).
- Nous allons maintenant découvrir comment on peut ajouter et modifier des données dans la base.
 - Pour cela, nous allons aborder de nouvelles requêtes SQL fondamentales et qu'il vous faut connaître :
 - ✓ INSERT
 - ✓ UPDATE
 - ✓ DELETE.

Partie 4 : Bases de données

Ecrire les données

1. INSERT : ajouter des données

- La requête INSERT INTO permet d'ajouter une entrée

```
1 INSERT INTO jeux_video(ID, nom, possesseur, console, prix, nbre_joueurs_max, commentaires)  
VALUES('', 'Battlefield 1942', 'Patrick', 'PC', 45, 50, '2nde guerre mondiale')
```

- Les nombres (tels que 45 et 50 ici) n'ont pas besoin d'être entourés d'apostrophes.
- ✓ Seules les chaînes de caractères les nécessitent.
- Il faut commencer par les mots-clés INSERT INTO qui indiquent que vous voulez insérer une entrée.
- Préciser ensuite le nom de la table (ici jeux_video)
- Puis listez entre parenthèses les noms des champs dans lesquels vous souhaitez placer des informations.
- Enfin inscrire VALUES suivi des valeurs à insérer dans le même ordre que les champs que vous avez indiqués.

Partie 4 : Bases de données

Ecrire les données : INSERT

- La requête INSERT INTO permet d'ajouter une entrée

```
1 INSERT INTO jeux_video(ID, nom, possesseur, console, prix, nbre_joueurs_max, commentaires)
  VALUES('', 'Battlefield 1942', 'Patrick', 'PC', 45, 50, '2nde guerre mondiale')
```

➤ pour le premier champ (ID) :

- ✓ Nous avons laissé des apostrophes vides.
 - ❖ C'est voulu : le champ a la propriété auto_increment
 - ❖ MySQL mettra donc le numéro d'ID lui-même.
 - ❖ On pourrait même se passer du champ ID dans la requête :

```
1 INSERT INTO jeux_video(nom, possesseur, console, prix, nbre_joueurs_max, commentaires)
  VALUES('Battlefield 1942', 'Patrick', 'PC', 45, 50, '2nde guerre mondiale')
```

Partie 4 : Bases de données

Ecrire les données : INSERT

- La requête INSERT INTO permet d'ajouter une entrée
 - On n'est pas obligés de lister les noms des champs en premier
 - ✓ cette requête marche tout aussi bien (mais elle est moins claire) :

```
1 INSERT INTO jeux_video VALUES('', 'Battlefield 1942', 'Patrick', 'PC', 45, 50,  
2     '2nde guerre mondiale')
```
 - ❖ Il faut lister les valeurs pour tous les champs sans exception (ID compris) dans le bon ordre.
- Application PHP : script insert.php + select_all.php
 - Utilisons cette requête SQL au sein d'un script PHP.
 - Cette fois, au lieu de faire appel à query() on va utiliser exec() qui est prévue pour exécuter des modifications sur la base de données
 - ce code ajoute une entrée dans la BDD pour le jeu « Battlefield 1942 », appartenant à « Patrick », qui fonctionne sur « PC », qui coûte 45 euros, etc.

Partie 4 : Bases de données

Ecrire les données : INSERT

- Application PHP : script insert.php
 - La présence de multiples apostrophes rend la requête un peu difficile à lire et à écrire à cause des antislashes \ que l'on doit rajouter devant.
 - De plus, cette requête insère toujours les mêmes données.
 - Comme on l'a vu dans le chapitre précédent, si on veut rendre une partie de la requête variable, le plus rapide et le plus sûr est de faire appel aux requêtes préparées.

Partie 4 : Bases de données

Ecrire les données : INSERT

- Insertion de données variables grâce à une requête préparée
 - Si on choisit d'utiliser une requête préparée le fonctionnement est en fait exactement le même que dans le chapitre précédent : script `insert_preparee.php`
 - Pour plus de clarté :
 - ✓ j'ai utilisé ici des marqueurs nominatifs.
 - ✓ j'ai créé l'array sur plusieurs lignes : c'est autorisé, et c'est surtout bien plus lisible.
 - Les variables telles que `$nom` et `$possesseur` doivent avoir été définies précédemment.
 - ✓ Généralement, on récupèrera des variables de `$_POST` (issues d'un formulaire) pour insérer une entrée dans la base de données.

Partie 4 : Bases de données

Ecrire les données

2. UPDATE : modifier des données

- La requête UPDATE permet de modifier une entrée

```
1 UPDATE jeux_video SET prix = 10, nbre_joueurs_max = 32 WHERE ID = 51
```

- Ensuite, le nom de la table (jeux_video).
- Le mot-clé « SET », qui sépare le nom de la table de la liste des champs à modifier.
- Viennent ensuite :
 - ✓ les champs qu'il faut modifier
 - ✓ séparés par des virgules.
 - ❖ Ici, on modifie le champ « prix »,
 - ❖ on lui affecte la valeur « 10 » (prix = 10),
 - ❖ puis on fait de même pour le champ nbre_joueurs_max.
 - ❖ Les autres champs ne seront pas modifiés.
 - ✓ Enfin, le mot-clé « WHERE » est tout simplement indispensable.
 - ❖ Il nous permet de dire à MySQL quelle entrée il doit modifier
 - ❖ On se base très souvent sur le champ ID pour indiquer quelle entrée doit être modifiée.
 - ❖ Ici, on suppose que « Battlefield » a été enregistré sous l'ID no 51.

Partie 4 : Bases de données

Ecrire les données : UPDATE

- La requête UPDATE permet de modifier une entrée
 - On peut se baser sur le nom du jeu au lieu de l'ID pour effectuer cette sélection :

```
1 UPDATE jeux_video SET prix = '10', nbre_joueurs_max = '32' WHERE nom = 'Battlefield 1942'
```

➤ Exemple :

- ✓ Florent vient de racheter tous les jeux de Michel. Il va falloir modifier ça tout de suite.
 - ❖ On ne veut pas modifier toutes les entrées une à une
 - ❖ Modifier la requête pour le faire en une fois

```
1 UPDATE jeux_video SET possesseur = 'Florent' WHERE possesseur = 'Michel'
```

- Traduction : « Dans la table jeux_video, modifier toutes les entrées dont le champ possesseur est égal à Michel, et le remplacer par Florent. »

Partie 4 : Bases de données

Ecrire les données : UPDATE

- Application en PHP

➤ De la même manière, en PHP on fait appel à `exec()` pour effectuer des modifications :

```
1 <?php
2 $bdd->exec('UPDATE jeux_video SET prix = 10, nbre_joueurs_max = 32 WHERE nom = \'Battlefield
   1942\'');
3 ?>
```

✓ Remarque : cet appel renvoie le nombre de lignes modifiées.

```
1 <?php
2 $nb_modifs = $bdd->exec('UPDATE jeux_video SET possesseur = \'Florent\' WHERE possesseur =
   \'Michel\'');
3 echo $nb_modifs . ' entrées ont été modifiées !';
4 ?>
```

Partie 4 : Bases de données

Ecrire les données : UPDATE

- Avec une requête préparée
 - Si vous insérez des données variables, par exemple envoyées par l'utilisateur :
 - ✓ là encore de faire appel à une requête préparée : `script update_preparee.php`

Partie 4 : Bases de données

Ecrire les données

- 3. DELETE : supprimer des données : script delete.php
 - DELETE. Rapide et simple à utiliser
 - elle est quand même un poil dangereuse : après suppression, il n'y a aucun moyen de récupérer les données, alors faites bien attention !
 - Voici comment on supprime par exemple l'entrée de « Battlefield » :

```
1 DELETE FROM jeux_video WHERE nom='Battlefield 1942'
```

- ✓ DELETE FROM : pour dire « supprimer dans » ;
- ✓ jeux_video : le nom de la table ;
- ✓ WHERE : indispensable pour indiquer quelle(s) entrée(s) doi(ven)t être supprimée(s).
- ✓ *Si vous oubliez le WHERE, toutes les entrées seront supprimées. Cela équivaut à vider la table.*
- ✓ On peut là encore passer par :
 - ❖ exec() si vous voulez exécuter une requête bien précise,
 - ❖ ou bien utiliser une requête préparée si votre requête dépend de variables.

Partie 4 : Bases de données

Ecrire les données

4. Traiter les erreurs SQL

- S'il peut y avoir des erreurs en PHP, il peut aussi y avoir des erreurs en SQL !
- Il se peut par exemple :
 - que votre requête soit mal écrite,
 - que la table que vous voulez ouvrir n'existe pas, etc.
- Bref, les erreurs possibles sont là encore nombreuses.
- Toutefois, ce n'est pas MySQL qui vous dira qu'il y a une erreur, mais PHP :
 - et ce dernier n'est pas très bavard en ce qui concerne les erreurs SQL.
- Nous allons voir :
 - comment repérer une erreur SQL en PHP ;
 - comment faire parler PHP pour qu'il nous donne l'erreur SQL

Partie 4 : Bases de données

Ecrire les données : traiter les erreurs SQL

- Repérer l'erreur SQL en PHP
 - Lorsqu'il s'est produit une erreur SQL, la page affiche le plus souvent l'erreur suivante :

```
Fatal error: Call to a member function fetch() on a non-object
```

- ✓ Cette erreur survient lorsque vous voulez afficher les résultats de votre requête, généralement dans la boucle `while ($donnees = $reponse->fetch())`.
- ✓ Quand vous avez cette erreur :
 - ❖ c'est la requête SQL qui précède qui n'a pas fonctionné.
 - ❖ Il vous manque cependant des détails sur ce qui a posé problème dans la requête.
- ✓ Nous allons maintenant voir comment on peut remédier à cela.

Partie 4 : Bases de données

Ecrire les données : traiter les erreurs SQL

- Afficher les détails de l'erreur
 - Repérez la requête qui selon vous plante (certainement celle juste avant la boucle `while`)
 - demandez d'afficher l'erreur s'il y en a une, comme ceci :

```
1 <?php
2 $reponse = $bdd->query('SELECT nom FROM jeux_video') or die(print_r($bdd->errorInfo()));
```

- ✓ Si la requête fonctionne, aucune erreur ne sera affichée.
- ✓ Si en revanche la requête plante, PHP arrêtera de générer la page et vous affichera l'erreur donnée par MySQL...
- ✓ En général, MySQL vous dit « You have an error in your SQL syntax near 'XXX' ».
 - ❖ À vous de bien relire votre requête SQL ;
 - ❖ l'erreur se trouve généralement près de l'endroit où on vous l'indique.

Partie 4 : Bases de données

Ecrire les données

5. Résumé

- On utilise différents mots-clés en fonction du type de modification que l'on souhaite effectuer :
 - INSERT INTO : ajout d'une entrée ;
 - UPDATE : modification d'une ou plusieurs entrées ;
 - DELETE : suppression d'une ou plusieurs entrées.
- Comme pour la sélection de données, on utilise les requêtes préparées pour personnaliser nos requêtes en fonction de variables.
- Lorsqu'on utilise UPDATE ou DELETE, il faut penser à filtrer avec un WHERE sinon toute la table sera affectée par l'opération !
- PHP indique si une erreur avec MySQL est intervenue.

Partie 4 : Bases de données

Ecrire les données

- Exercice :
 - Modifier votre TP N°2 pour enregistrer les entrées dans une table de votre base de données
 - ✓ Créez la table si elle n'existe pas
 - ✓ Enregistrez les informations
 - ✓ Une fois les infos enregistrées :
 - ❖ Affichez le résultat de l'enregistrement
 - ❖ Affichez l'ensemble des entrées de la table