Base de donnÃ©es et requÃªtes

CatÃ©gorie : Fiches techniques

PubliÃ© par Fooups le 16/04/2005

Si le module que vous souhaitez crÃ©er fait appel Ã Â des donnÃ©es stockÃ©es dans des tables de la Bdd, vous serez amenÃ©, dÃªs le dÃ©but de sa crÃ©ation, Ã Â mettre en place vos tables et Ã©crire les requÃªtes nÃ©cessaires Ã Â l'extraction, ou l'insertion, de ces donnÃ©es.
Ce document a pour but de vous expliquer les grandes lignes Ã Â suivre, le dÃ©tail des mÃ©thodes utilisables se trouvant dans le mÃ©mento du dÃ©veloppeur.

1) Les tables

Nous supposerons que vous Ãªtes familer avec l'organisation des donnÃ©es dans les tables, les diffÃ©rents types de champs, les index, etc.
Si ce n'est pas le cas, approfondissez la question, car il est primordial de crÃ©er correctement les tables avant de coder, toute modification ultÃ©rieure risquant de se rÃ©percuter sur de nombreuses portions de code.

Normalisez vos tables: il existe des rÃ¨gles prÃ©cises relatives Ã Â la rÃ©partition des donnÃ©es dans les tables et le contenu des diffÃ©rents champs. Si vous ignorez ce que sont les formes normales d'une base, il existe une littÃ©rature abondante sur le net (ou en librairie) sur ces formes normales: mais rassurez-vous, celÃ Â n'a rien de complexe et peut-Ãªtre vos tables sont dÃ©jÃ Â normalisÃ©es sans que vous le sachiez!

Envisagez la crÃ©ation d'une table 'categories' si votre application s'y prÃªte (c'est souvent le cas): xoops dispose de diffÃ©rentes mÃ©thodes de la classe xoopsTree permettant de gÃ©rer les catÃ©gories et sous-catÃ©gories dans lesquelles vous aurez classÃ© vos donnÃ©es.

Enfin, je ne saurais que trop vous conseiller de crÃ©er une classe dÃ©rivÃ©e de la classe xoopsObject : vos requÃªtes seront 'concentrÃ©es' dans cette classe, ce qui permet de rÃ©percuter les modifications de table beaucoup plus facilement, et vous ne travaillez plus avec des requÃªtes dans votre code, mais ferez appel aux mÃ©thodes getVar, setVar, initVar, etc.

2) Conventions

Respectez les conventions d'Ã©criture, votre code n'en sera que plus lisible et comprÃ©hensible par vous-mÃªme, mais Ã©galement par vos utilisateurs.
Certaines de ces conventions sont officielles, d'autres sont des conventions d'usage.
- Les instructions sql doivent Ãªtre Ã©crites en majuscules (SELECT, GROUP BY, etc.)
- Utilisez (de prÃ©fÃ©rence) $sql pour Ã©crire la requÃªte dans une chaine de caractÃ¨re.
- Utilisez (de prÃ©fÃ©rence) $result pour le rÃ©sultat de la requÃªte.
- Utilisez (de prÃ©fÃ©rence) $myrow pour parcourir les lignes du tableau $result.

3) RequÃªtes

N'utilisez jamais directement les fonctions mysql_ , n'utilisez que les mÃ©thodes de l'object xoopsDB : $xoopsDB->prefix(), $xoopsDB->query(), etc.
* Pour vous affranchir des instructions de connection Ã Â la Bdd.
* Pour bÃ©nÃ©ficier du mode debug sql.
* Pour une Ã©ventuelle compatibilitÃ© de xoops avec d'autres bases de donnÃ©es.Ecrivez votre requÃªte dans une chaine de caractÃ¨re ($sql) :

```
$sql = 'SELECT * FROM '.$xoopsDB->prefix('mymodule_table1').' WHERE champ1 = '.$variable.'
AND champ2 = '.$variable2';
```
Cette faÃ§on de procÃ©der prÃ©sente plusieurs avantages:
- une meilleure lisibilitÃ© du code
- la possibilitÃ© d'afficher votre requÃªte avec un `echo $sql` pendant la phase de mise au point
- la possibilitÃ© de concatÃ©ner votre requÃªte suivant le rÃ©sultat d'un test

```
 $sql = "SELECT cat_id, count(*) FROM ".$xoopsDB->prefix("mymodule_table1 ");
if ( !empty($order)) $sql .= " ORDER BY ".$order;
else $sql .= " ORDER BY ".$weight;
```

Puis exÃ©cutez la requÃªte
```
$result = $xoopsDB->query($sql)
```

Pendant la phase de dÃ©veloppement, mettez un test pour vÃ©rifier son exÃ©cution
```
if ( !$result ) { die( 'Erreur sql : '.$sql ); }
```

L'astuce ci-dessus prÃ©sente plusieurs avantages si votre requÃªte ne s'exÃ©cute pas :
- votre requÃªte s'affichera et vous permettra de voir s'il manque un Ã©lÃ©ment (ex WHERE champ1= AND.)
- vous pourrez faire un copier /coller de cette requÃªte et l'exÃ©cuter avec phpmyadmin pour voir le rÃ©sultat.
Dans votre code dÃ©finitif, ne laissez pas ce test, mais supprimez le, ou mieux modifiez le pour faire une redirection sur un message d'erreur, par exemple :.

```
if (!$result) $msg = _MD_ERROR_INSERT;
else $msg = _MD_INSERT_OK;
redirect_header("index.php", 3, $messagesent);
```
PrÃ©voyez Ã©galement le cas oÃ¹ votre requÃªte SELECT ne retourne aucun rÃ©sultat, qui, s'il n'est pas traitÃ©, peut aboutir Ã Â une page blanche

4) RequÃªtes SELECT

Il existe de nombreuses mÃ©thodes pour exploiter les rÃ©sultats.
Vous trouverez diffÃ©rents exemples Ã Â la page xoopsDB du mÃ©mento du dÃ©veloppeur.
N'oubliez pas que les rÃ©sultats sont obtenus sous forme d'un tableau qu'il faudra parcourir pour en extraire les rÃ©sultats. Une bonne comprÃ©hension de la notion des tableaux en Php vous est donc recommandÃ©e !**5) RequÃªtes INSERT et UPDATE**

Avant d'insÃ©rer des donnÃ©es dans une table, et particuliÃ¨rement si elles proviennent d'un formulaire, il est impÃ©ratif de contrÃ´ler la validitÃ© de ces donnÃ©es.
Exemple pour une donnÃ©e numÃ©rique: `$number = intval($number);`
Pour du texte, utilisez les diffÃ©rentes mÃ©thodes de la classe MyTextSanitizer.
Enfin, n'oubliez pas que votre texte, pour Ãªtre insÃ©rÃ©, doit Ãªtre placÃ© entre simples quotes.
Vous pourrez utiliser pour celÃ Â la mÃ©thode `$xoopsDB->quoteString($mytext)`
6) queryF

J'avoue ne pas tout connaitre Ã Â propos des requÃªtes et de queryF, et ne vous livrerai donc que ce que je sais :

- il arrive que les requÃªtes INSERT, UPDATE ou DELETE ne s'exÃ©cutent pas en utilisant la mÃ©thode $xoopsDB->query($sql).
- on peut utiliser dans ce cas la mÃ©thode $xoopsDB->queryF($sql), le F signifiant force.
- attention, l'utilisation non contrÃ´lÃ©e de cette mÃ©thode peut entrainer une faille de sÃ©curitÃ©, il est impÃ©ratif de vÃ©rifier les donnÃ©es avant insertion. **7) DÃ©claration**

Si vous utilisez les mÃ©thodes de la classe xoopsDB dans une fonction, n'oubliez pas de la dÃ©clarer globale:

```
function my_function($param) {
global $xoopsDB;
```

8) Mise au point

Vous disposez pour la mise au point de vos requÃªtes de la console de dÃ©buggage que vous activez dans Administration>>SystÃªme>>PrÃ©fÃ©rences>>ParamÃªtres gÃ©nÃ©raux>>Mode de mise au point>>MySQL/BlocsPersonnellement, je prÃ©fÃªre rester en mode Mise au point Php et insÃ©rer des echo $sql.'
'; Ã Â la suite des mes requÃªtes, plutÃ´t que d'avoir cette fenÃªtre qui surgit Ã Â chaque page ... Par contre, cette console de dÃ©buggage est extrÃªmement utile pour connaitre toutes les requÃªtes qui s'exÃ©cutent Ã Â l'appel d'une page, vous permettant ainsi, entre autres, de faire la chasse aux requÃªtes inutiles !