

TP 11 – Interrogation de bases de données en PHP

Internet et outils (IO2)

Le but de ce TP est d'apprendre à interfacier les requêtes MySQL avec le langage PHP. On va se servir, pour s'entraîner, de bases de données représentant l'état civil de personnes habitant en France. On dispose des tables `Adresse`, `Civilite`, `CodePostal` et `Coordonnee`.

Ident	Nom	Prenom	DateNaissance	Sexe
1	Lemoine	Ludwig	1975-10-23	M
2	Mathieu	Patrick	1960-07-29	M
3	Mathieu	Sabine	1966-05-03	F
...

Ident	CodeAdresse	Tel
1	1693065	0135116409
1	1693065	0604440802
2	1693061	0161050334
2	1693061	0645057571
3	1693061	0161050334
...

CodeAdresse	NumRue	NomRue	CodePostal
1693065	10	Bonaparte (rue)	77114
1691908	49	Péguy (rue)	77128
1692167	4	Mignon (rue)	28137
...

CodePostal	Ville	Departement
01143	MOGNENEINS	AIN
32000	AUCH	GERS
73121	PRESLE	SAVOIE
...

FIG. 1 – Tables `Civilite`, `Coordonnee`, `Adresse`, `CodePostal`

Rappels de cours et échauffement

La première chose à faire pour interroger une base de données est de se connecter à celle-ci. La fonction php correspondante est `mysql_pconnect(serveur, utilisateur)`. Comme la semaine dernière, le serveur est ici `pams.script.jussieu.fr` et l'utilisateur est `toto`. Cette fonction tente de créer une connexion avec le serveur MySQL gérant les bases de données. Si elle réussit, cette fonction renvoie l'identifiant de la connexion.

Une fois établie la connexion, il faut préciser le nom de la base de donnée que l'on veut interroger. Ceci se fait à l'aide de la fonction `mysql_select_db(base, connexion)` où `base` est le nom de la base de données (`monitIO2` pour ce TP) et `connexion` est l'identifiant de connexion.

Une fois choisie la base de données, on peut formuler des requêtes MySQL au serveur en utilisant la fonction `mysql_query(requête, connexion)` où `requête` est une chaîne de caractères formulant la requête MySQL (sans le point-virgule de fin d'instruction!) et `connexion` est l'identifiant de connexion au serveur de bases de données.

Le résultat des requêtes envoyées au serveur via `mysql_query()` est un ensemble de lignes que l'on manipule avec les fonctions `mysql_fetch_row(resultat)` ou `mysql_fetch_array(resultat)`, `resultat` étant le résultat d'une requête effectuée avec `mysql_query()`. Les deux fonctions renvoient une donnée de type tableau. La différence étant qu'avec la première, les indices du tableau sont numériques alors qu'avec la seconde, ce sont des chaînes de caractères représentant les champs des tables de la base de données.

Exercice 1 (*) Connexion au serveur Ecrivez une page PHP se connectant au serveur et à la base de donnée `monitIO2`. On définira au début du code php trois variables `$server`, `$user` et `$bd` de manière à rendre le code plus générique. Essayer d'afficher des erreurs appropriées si le serveur est injoignable ou si la base de donnée n'existe pas.

Exercice 2 () Affichage des tables disponibles** Exécuter une requête MySQL et analyser le résultat à l'aide d'une boucle `while` et de la fonction `mysql_fetch_row` de manière à afficher l'ensemble des tables disponibles dans la base de données `monitI02`. Pour simplifier le code par la suite, on stockera toujours le code MySQL d'une requête dans une chaîne `$request` avant d'appeler la fonction `mysql_query()`.

Affichage de requêtes quelconques

Exercice 3 (*) En-Tête de tableaux. Écrivez une fonction PHP `print_header($fields)` affichant en XHTML l'en-tête d'un tableau dont les noms de colonnes vont être les valeurs du tableau `$fields`.

Exercice 4 () Affichage de résultats de requêtes MySQL.** Écrivez une fonction PHP `print_result($result)` affichant les lignes du tableau correspondant au résultat `$result` d'une requête MySQL.

Exercice 5 (*) Affichage complet. À l'aide des deux fonctions précédentes, écrivez une fonction `print_request($result,$field)` affichant le résultat complet d'une requête sous forme de tableau XHTML. aucun éléments ne satisfait la requête.

Exercice 6 () Contenu d'une table.** On va maintenant utiliser les fonction précédentes pour afficher le contenu de la table `civilite`. On commencera par récupérer le titre des colonnes dans un tableau `$fields` à l'aide d'une requete `DESC civilite`. Ensuite, on initialisera la variable `$request` qui va contenir la chaîne correspondante à notre requête MySQL. On appellera enfin les fonctions précédentes pour afficher le résultat.

Exercice 7 (*) Nombre de ligne du résultat Modifier la fonction `print_request` de manière à afficher le nombre de lignes satisfaisant la requête demandé. S'il y en a aucune, afficher un message du genre "aucun résultat". On utilisera la fonction `mysql_num_rows`.

Affichage d'une table quelconque

Exercice 8 () Liste des tables** En récupérant le code de la question 2, créer un formulaire qui permet à l'internaute de choisir l'une des tables de la base de donnée `monitI02` à l'aide d'un `select`.

Exercice 9 (*) Traitement du formulaire Récupérez le code de la question 6 pour traiter le formulaire de la question précédente et afficher la table voulu.

Formulaire de recherche

Exercice 10 (*) Jointure Pour l'ensemble des questions suivantes on veut afficher un tableau contenant l'ensemble des personnes connues satisfaisant certains critères. On affichera dans le tableau résultat leur nom, leur prénom, leur date de naissance, leur code d'adresse et leur numéro de téléphone. Pour cela, commencer par définir à la main les variables `$request` et `$fields` appropriés pour afficher toutes les personnes avec les bonnes colonnes.

Exercice 11 (*) Recherche par nom et prénom Créer un formulaire permettant de rentrer un nom et prénom. Modifier le code de la question précédente pour qu'il s'exécute lorsque l'utilisateur appuie sur un bouton `rechercher`, et qu'il tienne compte des contraintes sur le nom et prénom.

Exercice 12 (*) Filtrage par sexe Rajouter des boutons `radio` permettant de choisir si l'on affiche que les femmes, que les hommes ou tout le monde. On pourra définir un choix par default en XHTML à l'aide de l'attribut `checked="checked"` de l'`input` de type `radio`.

Exercice 13 () Filtrage par age** Rajouter deux cases dans le formulaire permettant de définir un age min et un age max. Modifier le code php de traitement pour produire la bonne requête MySQL correspondante dans `$request`.

Plus dur

Exercice 14 (*) Adresses cliquables.** En reprenant la fonction `print_result`, améliorez celle-ci pour que les codes d'adresse deviennent cliquables et renvoient alors des précisions sur l'adresse concernée (numéro de la rue, nom de la rue, nom de la ville, code postal et nom du département). Il faudra pour cela passer des arguments au code PHP en modifiant l'adresse de la page (par exemple `?codeAddr=code de l'adresse`).

Exercice 15 (*) Ordonner les colonnes.** En reprenant la fonction `print_header`, ajoutez un lien autour de chaque en-tête pour permettre un affichage en ordre croissant de la colonne concernée. Il faudra utiliser le mécanisme des session pour se souvenir de la dernière requête et pouvoir afficher la même table.

Exercice 16 () Ordonner les colonnes (suite)** Modifier le code précédent de manière à ce que si l'on re clique sur le lien, l'ordre soit maintenant décroissant.