TP PHP/MySQL n°3

I. Truck, Université Paris 8, 2019–2020

1 Stocker des informations

- a. Comme on l'a vu dans le TP précédent, les fichiers permettent de stocker des informations de façon pérenne et d'y accéder facilement. Cependant, comment faire si l'on doit sauvegarder des informations en grande quantité et que l'on souhaite structurer? \Rightarrow On va bien sûr utiliser une **base de données**.
- b. Le but est d'organiser les informations à enregistrer dans différentes tables et de pouvoir y accéder n'importe quand dans le programme, sans se soucier de la façon d'aller rechercher ces informations (pas besoin d'écrire des boucles pour récupérer une à une les lignes d'informations, comme pour les fichiers). En effet, c'est le SGBD qui gère la façon de stocker les données. Dans le cas du SGBD MySQL (qui sera utilisé en TP), sous Windows si c'est Wamp Server 64 bits qui a été installé par exemple les fichiers contenant les tables et les bases de données créées sont habituellement dans le répertoire C:\Wamp64\bin\mysql\mysql8.0.19\data si la version installée est la 8.0.19. Sous Linux, c'est dans le répertoire /var/lib/mysql.
- c. MySQL n'est pas le seul SGBD disponible. Il existe également PostgreSQL (libre et gratuit comme MySQL), Oracle (payant), Microsoft SQL Server (payant également), etc.
- d. Mais que stocker et comment? Par exemple, on peut vouloir stocker les noms, prénoms des internautes (des visiteurs), leur login, leur mot de passe, les messages postés (cas d'un forum), les préférences des visiteurs, etc. Il faut ensuite agencer ces informations sous forme de tables (une table rassemble des informations souvent hétérogènes autour d'un même sujet exemple, une table visiteur qui contient un nom, un courriel, une adresse physique, autant d'informations qui ne sont pas de même nature) et relier les tables entre elles.
- e. Dans notre contexte, pour créer une base de données et ses tables, c'est-à-dire pour dialoguer avec MySQL, on a principalement deux choix : utiliser l'interface d'administration **phpmyadmin** ou bien écrire du SQL inclus dans du code PHP. Dans ce TP, on se focalise sur l'utilisation de **phpmyadmin** via l'interface, mais aussi en tapant directement des requêtes SQL.

2 Exercice : création d'une table via phpmyadmin

- a. On se connecte à la base de données MySQL via l'interface d'administration **phpmyadmin** http://handiman.univ-paris8.fr/phpmyadmin en utilisant le nom d'utilisateur et le mot de passe fournis en début d'année.
- b. Il s'agit de créer une table *client* contenant les champs suivants : le *nom*, le *prénom*, la *ville*, le *sexe* et le *courriel* du client. Commencer par écrire cette table sous la forme d'un tableau (sur le papier) en remplissant trois enregistrements différents, avec deux noms commençant par la lettre 'D' et deux prénoms identiques. Quel est l'identifiant de la table *client*? Quels sont les types de chacun des champs?
- c. Vous n'avez pas le **privilège** de créer une base de données, aussi, il faut créer cette table sous la base existante qui porte le nom de votre *login*, c'est-à-dire que vous "choisissez une base de données". Ensuite, vous créez une nouvelle table en la nommant TP3_client et en cliquant sur les icônes appropriés et en indiquant tout de suite le nombre de champs dont vous avez besoin.
- d. Bien remplir le nom du champ à chaque fois, ainsi que le type, la taille, l'index et l'AUTO_INCREMENT (si nécessaire). Quels sont les types de champs disponibles? Quels sont les types d'index? Que signifie PRIMARY? Et INDEX?
- e. Quelle est la différence entre VARCHAR et TEXT? Dans quel cas utilise-t-on l'un plutôt que l'autre?
- f. Quand la table est enregistrée (clic sur "Exécuter" ou "Sauvegarder", selon les versions), la modifier en ajoutant un champ dateInscription.

- g. Regarder également les différents onglets (Afficher, Structure, SQL, etc.) et la possibilité de voir le source PHP (cf. "Créer source PHP"). Comment faire pour insérer des enregistrements dans la table?
- h. Ajouter les trois enregistrements préparés au brouillon (en remplissant seulement la colonne "valeur" à chaque fois).
- i. Ajouter ensuite une table *commentaire* (nommée TP3_commentaire pour mieux la repérer) contenant trois champs : *id*, *idClient*, *comm*.
 - id est la clef primaire, idClient permet de faire le lien entre la table commentaire et la table client, et comm est de type TEXT et contient le commentaire déposé par le client. On insérera cinq enregistrements dans cette nouvelle table en liant 2 des 3 clients. Donc, un client n'aura aucun commentaire associé, un autre en aura un seul et le troisième en aura 4.
- j. Question subsidiaire. Peut-on créer des clefs étrangères dans une BD MySQL, telle qu'elle est initialisée par défaut? Regarder InnoDB et MyISAM avec ALTER TABLE 'table' ENGINE=InnoDB)

3 Exercice: interrogation d'une base de données via phpmyadmin

a. La documentation officielle de MySQL et les manuels de référence sont situés à l'adresse https://dev.mysql.com/doc/ Il suffit de taper une fonction, par exemple, pour que son manuel apparaisse.

De plus, la liste des fonctions se trouve à l'adresse :

https://dev.mysql.com/doc/refman/8.0/en/functions.html

Le manuel de la syntaxe des commandes SQL (SELECT, CREATE, UPDATE, JOIN, etc.) supportées par MySQL se trouve à l'adresse https://dev.mysql.com/doc/refman/8.0/en/sql-statements.html

- b. Reprendre la table *client* et demander la liste des clients dont le nom commence par un D (par exemple). Pour ce faire, il faut utiliser l'onglet "SQL" et exécuter une requête SQL tapée dans la fenêtre.
- c. Quel est le code PHP généré?
- d. Tester avec d'autres requêtes, en fonction des données que vous aurez saisies. Par exemple,
 - afficher la liste des clients dont le prénom est 'Xxxx'. Ou bien,
 - afficher les clients qui se sont inscrits à une certaine date. Ou encore,
 - afficher la liste des clients qui se sont inscrits avant ou après une certaine date.
- e. Croisement des informations dans les deux tables créées :
 - Afficher les commentaires de tous les clients.
 - Afficher le nom et le commentaire des clients dont le nom commence par la lettre 'D'. (Faire une jointure).
 - Afficher le nom des clients qui n'ont pas fait de commentaire. (Faire une jointure ou une sous-requête)
 - Afficher le nom des clients (et le nombre de commentaires en utilisant COUNT) qui ont fait au moins un commentaire.