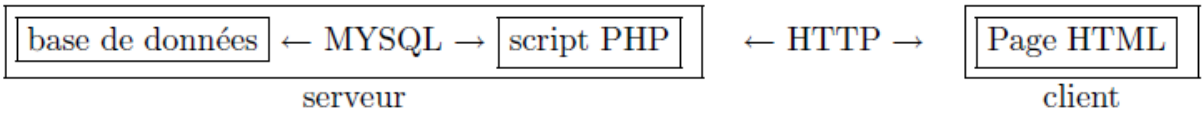
CREATION WEB DYNAMIQUE

IV °) MySQL

IV-1 °) Introduction

MYSQL dérive directement de SQL (Structured Query Language) qui est un langage de requêtes vers les bases de données relationnelles. Le serveur de base de données MySQL est très souvent utilisé avec le langage de création de pages web dynamiques : PHP.



Les données manipulées par le site seront stockées dans une base de données MySQL. Le script PHP, se servira de commandes MySQL pour gérer et extraire des informations de la base de données afin d'engendrer les pages xHTML qui seront interprétées par le navigateur.

Nous utiliserons EasyPHP qui est une application permettant d'installer et de configurer automatiquement un environnement de travail complet sous Windows en regroupant un serveur Apache, un serveur MySQL, une interface graphique de gestion des bases de données MySQL : PHPMyAdmin et le langage PHP.

IV-2 °) Structures d'une base de données MySQL

A- Créer et sélectionner une base :

CREATE DATABASE Nom_BD ; exemple : CREATE DATABASE bd_supManagement ;

USE Nom_BD ; exemple : USE bd_supManagement ;

Après sélection, on crée les tables qui constitueront la base de données Nom_BD

Exemple :

```
CREATE TABLE tb_Etudiant (

code_Etudiant integer(8),
nom_Etudiant varchar(20),
prenom_Etudiant varchar(20)

);
```

L'attribut code_Etudiant peut être définit comme clé primaire de tel sorte qu'il soit unique pour chaque étudiant afin d'éviter des homonymes. On peut aussi utiliser plusieurs attributs pour en faire une clé primaire.

Les tables d'une même base de données peuvent être liée par des relations selon les cardinalités, du model conceptuel des données (voir cours merise ou UML), suivant ;

Relation : 1, n c'est-à-dire une relation de un à plusieurs appelé aussi relation mère-fille.

Relation : n, n c'est-à-dire une relation de plusieurs à plusieurs

Relation : 1, 1 c'est-à-dire relation un à un.

Au niveau physique des données (model physique) :

Les attributs de tables deviennent des champs,

La relation n, n génère une troisième table entre les deux tables du model conceptuel.

La relation 1, n génère un champ appelé clé étrangère dans la table fille.

La relation 1, 1 fusionne les deux tables pour en faire une seule table (mais pas obligatoirement).

Pour ajouter une clé primaire directement avec la création de la table :

CREATE TABLE tb_Etudiant (

code_Etudiant integer(8),
nom_Etudiant varchar(20),
prenom_Etudiant varchar(20),

primary key ('code_Etudiant'),

);

Pour ajouter une clé indirectement :

Altere table add forein key ('nom_champ_de_la_table_fille') references t_nomTableMere (nom_champCorrespondant) ;

Ainsi Altere table permet de modifier la structure de la table :

Cas MySQL
```
ALTER TABLE `test` CHANGE `CODE` `CODEs` INT( 8 ) NOT NULL AUTO_INCREMENT
```

Pour Supprimer la table, on utilise la close :

Drop table nom_table_à_supprimer ;

Pour changer le nom de la table :

Rename ancien_nom to nouveau_nom ;

B- LES ORDRES SQL pour manipuler les données

Insertion des données :

INSERT INTO `t_Etudiant` (`code_Etudiant`, `nom_Etudiant`, `prenom_Etudiant`)

VALUES (valeur_code_si entier, 'valeur_nom', 'valeur_prenom') ;

Mise à jour des données :

UPDATE t_Etudiant SET nom_Etudiant = 'CISSE' WHERE code_Etudiant=2 ;

Suppression des données :

DELETE from t_Etudiant WHERE code_Etudiant=2 ;

Selection (requête) des données :

SELECT * FROM t_Etudiant ; // selection de tous les champs sans restriction

SELECT * FROM t_Etudiant WHERE nom_Etudiant= 'Cissé'; // selection toutes les info (les champs) sur uniquement les Cissé

SELECT nom_Etudiant, prenom_Etudiant from t_Etudiant ; // selection du nom et prénom de tous

SELECT nom_Etudiant, prenom_Etudiant from t_Etudiant WHERE nom_Etudiant= 'Cissé'; // selection du nom et prénom de uniquement les Cissé.

SELECT nom_Etudiant as 'Nom', prenom_Etudiant as 'Prénom' from t_Etudiant ; // selection avec des alias.

Ce petit rappel n'est pas exhaustif, pour plus de détails, voir le cours de base de données et SQL. MySQL répond aux standards du SQL avec quelques différences facilement maitrisables surtout avec le générateur du script SQL du PHPMyAdmin que nous verrons en classe.

IV-3 °) PHPMyAdmin

EasyPHP offre un utilitaire : PHPMyAdmin qui donne une interface graphique pour Mysql. Nous l'utiliserons en classe pour notre méthode pratique d'évolution du cours.

IV-4 °) MySQL et PHP

L'utilisation de MySQL avec PHP s'effectue en 5 temps :
1. Connexion au serveur de données
2. Sélection de la base de données
3. Requête ou tout autre ordre SQL
4. Exploitation des requêtes dans ce cas précis.
5. Fermeture de la connexion (facultatif) surtout dans le cas d'une connexion simple.

1. Connexion au serveur de données local

```
$connexion = mysql_connect("localhost","root","mysql") or die('Erreur : '.mysql_error());
```

Pour le cas d'un site hébergé, les données à mettre vous seront communiqué par l'hébergeur.

2. Sélection de la base de données

```
$la_bd = mysql_select_db("bd_supManagement",$connexion) or die(('Erreur : '.mysql_error());
```

Il est préférable de mettre 1 et 2 dans un fichier php (par exemple connect.php) à faire appeler par un « include(connect.php) ; » à chaque fois qu'on a besoin de se connecter.

3. Requête ou tout autre ordre SQL

```
$reqEtudiant = "SELECT * FROM t_Etudiant";
$resultat = mysql_query($reqEtudiant) or die (('Erreur : '.mysql_error());
```

Lorsque l'on fait une requête de sélection, MySQL retourne les données et PHP les place en mémoire. En fin de script, la mémoire est libérée et les données sont donc perdues. Il est possible de libérer la mémoire, pour ne pas surcharger le serveur, avant la fin du script au moyen de la commande mysql free result.

4. Exploitation des requêtes

Les fonctions qui retournent un enregistrement sont : mysql fetch row, mysql fetch array et mysql fetch object et prennent comme paramètre l'identifiant de la requête.

mysql fetch row : Cette fonction retourne un enregistrement sous la forme d'un tableau simple.

Exemple :
```
$enregistrement = mysql_fetch_row( $resultat);
// Affiche le champ - nom -
echo $enregistrement[0] . '<br>';
// Affiche le champ - prenom -
echo $enregistrement[1] . '<br>';
```

mysql fetch array : Cette fonction retourne un enregistrement sous la forme d'un tableau associatif.

Exemple :
```
$enregistrement = mysql_fetch_array( $resultat );
// Affiche le champ - Nom -
echo $enregistrement['nom_Etudiant'] . '<br>';
// Affiche le champ - Prénom -
echo $enregistrement['prenom_Etudiant'] . '<br>';
```

mysql fetch object : Cette fonction retourne un enregistrement sous forme d'une structure (objet).

Exemple :
```
$enregistrement = mysql_fetch_object( $resultat );
// Affiche le champ - nom -
echo $enregistrement->nom_Etudiant . '<br>';
// Affiche le champ - prenom -
echo $enregistrement->prenom_Etudiant. '<br>';
```

Si il n'y a pas ou plus d'enregistrement à lire, ces fonctions retournent "false."

Pour savoir combien d'enregistrements ont été retournés par la sélection, on utilise la commande mysql num rows qui prend comme paramètre l'identifiant de la requête.

Exemple :
```
$nbr_Etudiant = mysql_num_rows($resultat) ;
echo $nbr_Etudiant ;
```

Des fonctions moins utilisés car ce qu'elles renvoient est plus ou moins connu d'avance :
mysql_num_fields($resultat); pour retourner le nombre des champs.
mysql_field_name($resultat, 0); pour avoir le nom du premier champs.

Enfin la fonction qui manipule le résultat comme un tableau est : mysql_result().

Exemple :
```
echo mysql_result($resultat,0,0) ;
```

5. Fermeture de la connexion

Vous pouvez fermer la connexion au moyen de la fonction mysql_close(), mais il est bon de savoir que cette opération sera faite automatiquement lorsque le script se terminera.

IV-5 °) Conclusion

Ces quelques éléments constituent la base à savoir sur l'utilisation de MySQL dans le Web. Pour plus de détails voir les exemples pendant le cours en classe.