ICT305 - WEB APPLICATION DEVELOPMENT

Licence 3 – Filière ICT4D

Université de Yaoundé I

Yaoundé, Janvier 2021

Njine Chuangueu

njinec@gmail.com

PRESENTATION DU COURS

- Développement de sites web dynamique conformes au modèle client-serveur à l'aide des langages de scripts coté serveur
- Connexion aux bases de données à l'aide des langages de scripts coté serveur et encapsulation des requêtes pour effectuer les opérations CRUD
- Utilisation des langages de scripts coté client pour l'exécution des requêtes asynchrones et la manipulation de l'arbre DOM

Niveau : Licence 3 – **Semestre** : 1 – **Crédits** : 4 – **Charge horaire** : 50hrs

Contrôle continu : 20% – **Examen pratique** : 40% – **Examen final** : 40%

OBJECTIFS DU COURS

- Décrire l'architecture des applications web déployées à travers le protocole IP
- Utiliser un langage de script coté serveur pour la génération des pages web dynamiques
- Intégrer des bases de données aux applications web
- Réaliser des échanges asynchrones en utilisant un langage de script coté client
- Enumérer quelques Framework de développement d'application web dynamique

COMPETENCES A DEVELOPPER

- Installer et configurer l'environnement minimal nécessaire au développement des applications web
- Générer des pages web à l'aide d'un langage de script serveur
- Interroger un SGBD à l'aide d'un langage de script serveur
- Déterminer le langage de script (client ou serveur) approprié pour faire des traitements
- Modifier l'arbre DOM d'une page web à l'aide d'un langage de script client
- Emettre des requêtes asynchrones à l'aide d'un langage de script client

PREREQUIS

- Produire des documents HTML valides et bien formés
- Différencier les éléments de type bloc des éléments de type en ligne
- Manipuler des formulaires HTML
- Appliquer un gabarit (feuille de styles) aux pages d'un site web
- Utiliser un langage de script coté client pour interagir avec l'utilisateur

RESSOURCES

- Serveur de page web ***Apache HTTP Server*** (version 2.0 au minimum)
- Moteur de script ***PHP*** (version 5.0 au minimum)
- Serveur ***MySQL*** (version 5.0 minimum)
- Logiciel d'administration ***phpMyAdmin***
- Editeur de texte ***Notepad++*** (version 5.0 minimum)
- Navigateur web (***Internet Explorer***, ***Mozilla Firefox***, ***Google Chrome***, …)

BIBLIOGRAPHIE

- **Engels, Jean.** *XHTML et CSS – Cours et exercices.* Paris : Editions Eyrolles, 2006. p. 523. ISBN : 2-212-11637-3.
- **Engels, Jean.** *PHP 5 – Cours et exercices.* 2e. Paris : Eyrolles, 2009. p. 662. ISBN : 978-2-212-12486-6.
- **Rigeaux, Philippe.** *Pratique de MySQL et PHP – Conception et réalisation de sites web dynamiques.* 4e. Paris : Dunod, 2009. p. 557. ISBN : 978-2-10-053752-5.
- **Templier, Thierry et Gougeon, Arnaud.** *JavaScript pour le Web 2.0.* Paris : Eyrolles, 2009. p. 509. ISBN : 978-0-470-43131-3.
- **Plasse, Michel.** *Développez en Ajax.* Paris : Eyrolles, 2006. p. 325. ISBN-13 : 978-2-212-11965-7.

SOMMAIRE DU COURS

1. L'architecture des applications web déployées à travers IP
2. Génération de pages web à l'aide du langage de script PHP
3. Accéder à une base de données sous le SGBD MariaDB avec PHP
4. Programmation DOM avec JavaScript
5. Exécution de requêtes asynchrones AJAX

Accéder à une base de données sous le SGBD MariaDB avec PHP

CHAPITRE ③

Accéder à une base de données sous le SGBD MariaDB avec PHP

Introduction

- **Les SGBDR** (**S**ystème de **G**estion de **B**ases de **D**onnées **R**elationnels) : Collection de logiciels facilitant l'exploitation des **bases de données relationnelles**
- Ils supportent presque tous le **langage de commande SQL** pour permettre la ***définition***, la ***manipulation***, ***l'interrogation*** et le ***contrôle*** des données
- Conçus suivant le **modèle client serveur**, ils interagissent avec des applications écrites dans *différents* langages de programmation (*C/C++, Java, VB, PHP, …*)
- Sont administrables à l'aide **d'outils de gestion** permettant par exemple de *gérer les privilèges des utilisateurs*, *exporter ou importer des bases de données*, *représenter et manipuler graphiquement des bases de données*, …

I) Rappels sur les BD relationnelles

Généralités sur les bases de données relationnelles

- **Base de données** : ensemble **organisé** et **structuré** de **données** réparties sur des supports de stockage accessibles par ordinateur
- Les bases de données sont conçues pour :
  - Permettre le stockage de ***grande quantités de données***
  - Limiter la ***redondance*** des informations (***unicité***)
  - Permettre la ***séparation*** entre les traitements et les données (***indépendance***)
  - Garantir la ***cohérence*** et ***l'intégrité*** des données après traitements

I) Rappels sur les BD relationnelles

Conception d'une base de données

- La conception d'une base de données répondant à un besoin est un processus qu'on peut « *simplifier grossièrement* » en 03 principales étapes**.** Concrètement, dans la situation qui justifie la création d'une base de donnée, on recense :
- Les noms des ***objets réels*** ou ***abstraits*** ayants une existence propre. On parlera **d'entité**
- Les ***verbes d'action*** qui traduisent un ***lien*** entre les objets recensés**.** On parlera **d'association**
- Les ***noms des éléments*** qui caractérisent les objets et les liens recensés**.** On parlera de **propriété** ou **d'attribut**

I) Rappels sur les BD relationnelles

Conception d'une base de données

- **A titre d'illustration, considérons le besoin exprimé à travers l'énoncé suivant :**

  « Pour inscrire un étudiant dans un établissement, on enregistre son matricule, ses noms, sa date de naissance, son sexe, son adresse email, l'année académique pour laquelle il prend une nouvelle inscription, la filière au sein de laquelle il veut s'inscrire, le niveau auquel il sollicite une inscription et son statut (nouveau, redoublant) »

I) Rappels sur les BD relationnelles

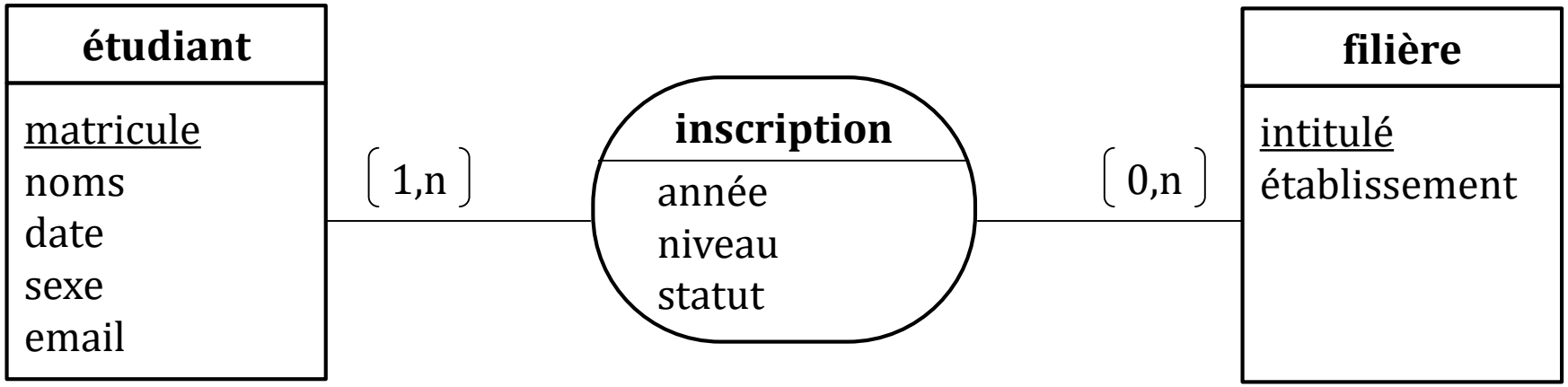
Conception d'une base de données

- A partir de la description précédente, on peut recenser les objets ***étudiants*** et ***filières*** comme *entités*, *l'association* ***inscrire*** qui traduit le lien qui existe entre ces 02 entités, et par exemple les *propriétés* ***année académique***, ***niveau***, ***statut*** qui caractérisent cette association. Les informations recensées peuvent être résumées ainsi :
- **Etudiant(**matricule, noms, date, sexe, email**)**
- **Filière(**intitulé, établissement**)**
- **Inscription(**année, niveau, statut**)**

I) Rappels sur les BD relationnelles

Conception d'une base de données

- La figure ci-après représente le **diagramme entité-association** correspondant **:**



- Le nombre de fois que chaque entité peut participer à l'association est exprimé à travers les **cardinalités**

I) Rappels sur les BD relationnelles

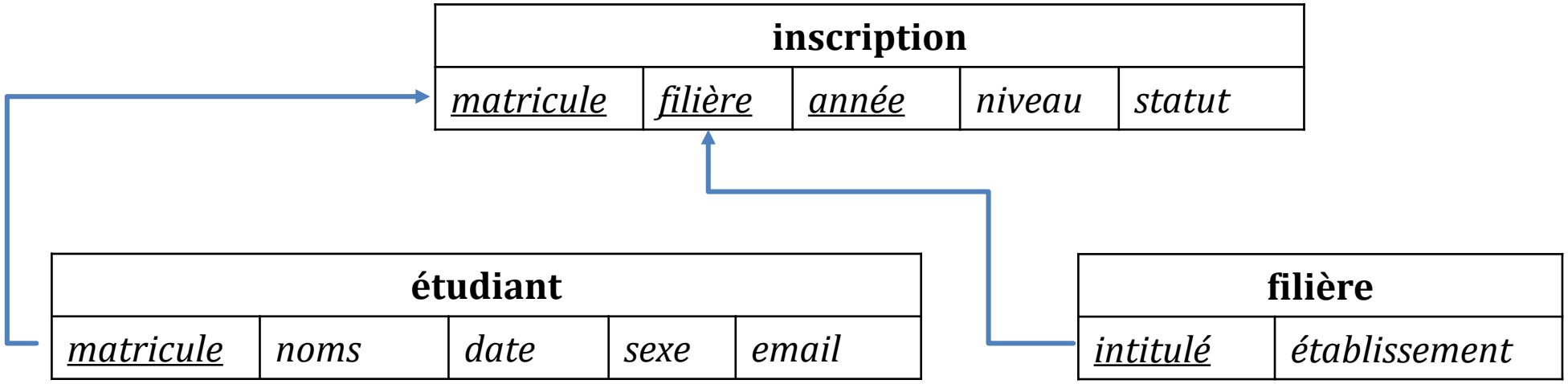
Conception d'une base de données

- Dans le **modèle relationnel**, les entités recensées deviennent des **tables** et les propriétés ou attributs deviennent des **champs**
- Des **règles de normalisations** permettent de faire migrer des champs entre des tables ou de transformé des associations en tables avec leurs propres champs
- Les ***occurrences*** d'une table sont appelées **enregistrements** de la table et correspondent à des *insertions* de données dans une base de données
- Les **contraintes d'intégrité** sont des règles qui caractérisent les valeurs que peuvent prendre un champ dans une table (*clé primaire*, *valeur non nulle*, *valeur par défaut*, …) et qui garantissent la ***cohérence*** des données

I) Rappels sur les BD relationnelles

Modélisation d'une base de données relationnelle

- Le **modèle relationnel** résultant du diagramme entité-association précédent après un processus de normalisation est le suivant :



II) Le langage de commandes SQL

Généralités sur les langages de commande

- Pour mettre en œuvre des bases de données relationnelles, on utilise des langages de commandes permettant de :
- *Décrire les données d'une table* : Il s'agit du **langage de description de données (LDD)**
- *Manipuler les données d'une table* : Il s'agit du **langage de manipulation de données (LMD)**
- *Interroger les données d'une table* : Il s'agit du **langage d'interrogation de données (LID)**
- *Contrôler les données d'une table* : Il s'agit du **langage de contrôle de données (LCD)**

II) Le langage de commandes SQL

Généralités sur les langages de commande

- Le **langage SQL** (**Structured Query Language**) est un standard qui implémente ces différents langages de commande (LDD, LMD, LID, LCD)
- Il est supporter par la plupart des SGBD relationnels (*MariaDB, Oracle, Microsoft SQL Server, DB2, Informix,* …)
- Il permet d'effectuer les opérations de *création*, *d'insertion*, de *modification*, *d'interrogation* et de *suppression* de données.
- Il peut également être utilisé comme **langage procédurale dans un SGBD** pour par exemple effectuer des *itérations*, programmer des *procédures stockées* ou gérer des *transactions*, …

II) Le langage de commandes SQL

Les requêtes de définition de données

- Les ordres **SQL** permettant de définir les structures des données sont : **CREATE**, **ALTER**, **DROP**, **RENAME**, **TRUNCATE**

```
CREATE DATABASE <nom_bd> ;

CREATE TABLE [<nom_bd>.]<nom_tbl> (
 <nom_chp1> <type_chp> [<contraintes>],
 <nom_chpN> <type_chp> [<contraintes>],
 [<contraintes>]
) ;

DROP TABLE <nom_tbl> ;

DROP DATABASE <nom_bd> ;
```

II) Le langage de commandes SQL

Les requêtes de manipulation de données

- Les ordres **SQL** permettant de manipuler les données sont : **INSERT**, **UPDATE**, **DELETE**, **LOCK TABLE**

```
INSERT [INTO] [<nom_bd>.]<nom_tbl> [(<nom_chp1>,...,<nom_chpN>)]
 VALUES (<val_chp1>,...,<val_chpN>) ;

UPDATE [<nom_bd>.]<nom_tbl>
 SET <nom_chp1>=<exp1> [, <nom_chpN>=<expN>]
 [WHERE (<conditions>)] ;

DELETE
 FROM [<nom_bd>.]<nom_tbl>
 [WHERE (<conditions>)] ;
```

II) Le langage de commandes SQL

Les requêtes d'interrogation de données

- Les ordres **SQL** permettant d'interroger les données sont : **SELECT**

```
SELECT <nom_chp1> [,..., <nom_chpN>]
 FROM <nom_tbl1> [,...,<nom_tblN>]
 [WHERE (<conditions>)] ;
```

II) Le langage de commandes SQL

Les requêtes de contrôle de données

- Les ordres **SQL** permettant de contrôler les données sont : **GRANT**, **REVOKE**, **COMMIT**, **ROLL BACK**, **SAVEPOINT**, **SET TRANSACTION**

```
CREATE USER <nom_user@host> [IDENTIFIED BY [PASSWORD] '<mdp>'] ;

GRANT <privilège [(chp1,...,chpN)]>
 ON <nom_bd>.<nom_tbl>
 TO <nom_user@host> [IDENTIFIED BY [PASSWORD] '<mdp>'] ;

REVOKE <privilège [chp1,...,chpN]>
 ON <nom_bd>.<nom_tbl>
 FROM <nom_user@host> ;

DROP USER <nom_user@host> ;
```

III) Présentation du SGBD MariaDB

Généralités sur MariaDB

- **MariaDB**, anciennement appelé **MySQL**, est un SGBD relationnel écrit en C/C++ et fonctionnant en mode *client/serveur* ou en mode *bibliothèque embarquée*
- Supporte plusieurs ***moteurs de stockage de données*** parmi lesquels **MyISAM**, **InnoDB**, **BDB**, **CVS**, …
- Supporte les ***transactions ACID***, les ***clés étrangères*** et le ***verrouillage des colonnes*** dans les tables via le moteur InnoDB
- Supporte également la création de ***sous requêtes***, de ***vues***, de ***procédures stockées*** et des ***déclencheurs*** (***trigger***)
- SGBDR *multiplateforme* distribué sous forme *de binaires* ou de *code source*

III) Présentation du SGBD MariaDB

Généralités sur MariaDB

- Le dossier d'installation du SGBD MariaDB comprend généralement les sous dossiers suivant :
- **`bin`** : Contient des exécutables (serveur, clients) et divers outils d'administration
- **`data`** : Contient les fichiers physiques des bases de données et des tables
- **`docs`** : Contient la documentation et le journal des changements de version
- **`include`** : Contient les fichiers d'entête à inclure dans les programmes C/C++
- **`lib`** : Contient les librairies utilisables par les clients mysql
- **`share`** : Contient le fichier des messages d'erreur

III) Présentation du SGBD MariaDB

La configuration du SGBD

- On peut configurer le serveur et les clients *mysql* à l'aide du fichier **my.cnf** (**my.ini** sous Windows) qui se trouve sous le répertoire d'installation
- Les directives prennent la forme **propriété=valeur** et sont regroupées en rubriques noté ainsi **[rubrique]**. En voici un aperçu :

  [mysqld]

  port = 3306 : Indique quel est le port d'écoute du serveur **mysqld**

  socket = /tmp/mysql.sock : Indique le fichier physique de la socket

  log_error = mysql_error.log : Indique l'emplacement du journal des erreurs

  pid_file = mysql.pid : Indique le fichier contenant le PID du serveur

III) Présentation du SGBD MariaDB

Le stockage physique des données

- La création d'une BD entraine la création d'un dossier *éponyme* dans le dossier **`data`**. Y sont enregistrés, les fichiers qui décrivent la ***structure*** de ses tables
- Les fichiers des ***données*** et des ***index*** des tables créées avec **MyISAM** sont également enregistré dans le dossier la BD
- Les ***données*** et les ***index*** des tables créées avec **InnoDB** sont stockées dans un fichier physique, quelque soit les BDs auxquelles elles appartiennent
- Les évènements normaux sont généralement *journalisées* dans des **fichiers binaires** localisés dans le dossier **`data`** sous la forme **`mysql-bin.xxx`**

III) Présentation du SGBD MariaDB

Les exécutables et outils d'administration

Le dossier **`bin`** contient entre autres les exécutables suivant :

- **`mysqld`** : Exécutable du serveur MariaDB
- **`mysql`** : Client interactif en ligne de commande
- **`mysqladmin`** : Client d'administration du serveur
- **`mysqlcheck`**, **`myisamchk`** : Utilitaires de vérification et réparation des tables
- **`mysqldump`**, **`mysqlimport`** : Utilitaires de sauvegarde et restauration (resp.)
- **`mysqlbinlog`** : Utilitaire de traitement des journaux binaires du serveur

III) Présentation du SGBD MariaDB

Déploiement de MariaDB sous WampServer

- Dans WampServer, le chemin d'accès au dossier d'installation par défaut de MariaDB est **`C:\Program Files\Wamp\bin\mysql\mysqlX.Y.Z\`**
- Dans WampServer, les erreurs du serveur MariaDB sont journalisées dans le fichier **`mysql.log`** qui se trouve dans le dossier **`C:\Program Files\Wamp\logs\`**
- Dans WampServer, le fichier de configuration de MariaDB est **`my.ini`** et se trouve dans le dossier **`C:\Program Files\Wamp\bin\mysql\mysqlX.Y.Z\`**
- Les logiciels tels que **phpMyAdmin** ou **SQLBuddy**, livrés avec WampServer, sont des ***clients graphiques*** d'administration du serveur MariaDB
- Ces logiciels se trouvent généralement dans **`C:\Program Files\Wamp\apps\`**

IV) Prise en main avec phpMyAdmin

- **phpMyAdmin (PMA)** : Logiciel d'administration graphique de SGBD MySQL ou MariaDB réalisé en PHP et JavaScript et distribué librement sous licence GPL
- Ce logiciel permet entre autres de :
  - *Créer* ou *supprimer* des bases de données
  - *Créer*, *modifier* ou *supprimer* des tables
  - *Visualiser* graphiquement la structure des tables
  - *D'insérer*, *modifier* ou *supprimer* des enregistrement dans les tables
  - *D'interroger* (*afficher*) le contenu des tables

IV) Prise en main avec phpMyAdmin

- Il permet également de :
  - *D'écrire* des requêtes SQL qui seront exécutées par le serveur
  - *Consulter* les statistiques d'utilisation du serveur
  - *D'afficher* le contenu des variables d'environnement du serveur
  - *Consulter* la liste des moteurs de stockage actifs sur le serveur
  - *Créer* ou *supprimer* des utilisateurs et de *gérer* leurs privilèges
  - *Importer* ou *exporter* des bases de données dans différents formats
  - *Consulter* les journaux binaires du serveur (dépend de la version de PMA)

V) Accès à MariaDB avec PHP

Les prérequis pour accéder au SGBD

- Pour ce connecter à un serveur MariaDB à travers un programme PHP, les conditions suivantes doivent être remplies :
- L'extension **MySQLi** doit être installée. Autrement dit, le fichier **`php_mysqli.dll`** doit se trouver dans le sous dossier **`/ext/`** du dossier d'installation de PHP
- L'extension **MySQLi** doit aussi être activé dans PHP. En d'autre termes, la directive **`extension=php_mysqli.dll`** doit apparaitre dans le fichier **`php.ini`**
- La librairie cliente **`libmysql.dll`** doit être présente et accessible à travers les variable d'environnement pour les versions de PHP antérieures à PHP 5.3.0

V) Accès à MariaDB avec PHP

Accès procédural à MariaDB

- La fonction **mysqli_connect()** permet de se connecter au serveur MariaDB.
```
<?php $con = mysqli_connect("127.0.0.1","test","pwd" [,"uy1"]); ?>
```
- La fonction **mysqli_change_user()** permettant de changer d'utilisateur courant.
```
<?php $ok = mysqli_change_user($con,"test","pwd","uy1"); ?>
```
- On peut également changer la BD active avec la fonction **mysqli_select_db()**.
```
<?php $ok = mysqli_select_db($con,"uy1"); ?>
```
- La fonction **mysqli_close()** permet de fermer la session et de se déconnecter.
```
<?php $ok = mysqli_close($con); ?>
```

V) Accès à MariaDB avec PHP

Accès procédural à MariaDB

- La fonction **`mysqli_query()`** permet d'exécuter des requêtes sur le serveur. **`<?php $res = mysqli_query($con, "SELECT * FROM étudiant"); ?>`**
- Selon que l'ordre SQL soit un *ordre de sélection*, ou un *ordre de manipulation*, le résultat renvoyé est un ***objet*** contenant la sélection ou une ***valeur booléenne***
- La fonction **`mysqli_num_fields()`** renvoi le nombre de champ d'une *sélection*. **`<?php $nbcl = mysqli_num_fields($res); ?>`**
- La fonction **`mysqli_num_rows()`** renvoi le nombre d'enregistrement d'une *sélection*. **`<?php $nblg = mysqli_num_rows($res); ?>`**

V) Accès à MariaDB avec PHP

Accès procédural à MariaDB

- On peut accéder aux résultats d'une *sélection* en transformant le résultat renvoyée par **`mysqli_query()`** en tableau
- La fonction **`mysqli_fetch_row()`** retourne le résultat sous forme de tableau indicé. **`<?php $tab_idx = mysqli_fetch_row($res); ?>`**
- La fonction **`mysqli_fetch_assoc()`** retourne le résultat sous forme de tableau associatif. **`<?php $tab_ass = mysqli_fetch_assoc($res); ?>`**
- La fonction **`mysqli_fetch_array()`** retourne un tableau qui est soit indicé, soit associatif ou les deux. **`<?php $tab = mysqli_fetch_row($res [,`*`type`*`]); ?>`**

V) Accès à MariaDB avec PHP

Accès procédural à MariaDB

- Après exploitation des résultats, la fonction **`mysqli_free_result()`** permet de libérer les ressources. **`<?php mysqli_free_result($res); ?>`**
- Dans le cas d'une *manipulation*, **`mysqli_affected_rows()`** permet de connaitre le nombre de ligne affectées. **`<?php $nb = mysqli_affected_rows($con); ?>`**
- En cas *d'insertion*, **`mysqli_insert_id()`** retourne le dernier identifiant créé pour un champ de type **`AUTO_INCREMENT`**. **`<?php $id = mysqli_insert_id($con); ?>`**
- En cas d'erreur, **`mysqli_error()`** renvoie le dernier message d'erreur survenu. **`<?php echo mysqli_error($con); ?>`**

V) Accès à MariaDB avec PHP

Accès procédural à MariaDB

- Avec **`mysqli_ping()`** on vérifie que la connexion est active et on se reconnecte automatiquement le cas échéant. **`<?php $ok = mysqli_ping($con); ?>`**
- Avec **`mysqli_real_escape_string()`**, on peut échapper les caractères à inclure dans une requête. **`<?php $ch_esc = mysqli_real_escape_string($ch); ?>`**
- La fonction **`mysqli_commit()`**, permet de valider une transaction et renvoi un booléen : **`<?php $ok = mysqli_commit($con); ?>`**
- La fonction **`mysqli_rollback()`**, permet d'annuler une transaction et renvoi aussi un booléen. **`<?php $ok = mysqli_rollback($con); ?>`**

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- L'extension **`php_myqli`** permet aussi d'effectuer des accès objet à MariaDB, mais n'est justement pas utilisable avec d'autres SGBD.
- L'extension **PDO** (**PHP Data Objects**) permet de s'affranchir de cette limitation et rend portable le code quelque soit le SGBD
- L'extension **PDO_MySQL** doit être installée, c'est-à-dire que **`php_pdo_mysql.dll`** doit se trouver dans le sous dossier **`/ext/`** du dossier d'installation de PHP
- L'extension **PDO_MySQL** doit être activé dans PHP. Autrement dit, la directive **`extension=php_pdo_mysql.dll`** doit apparaitre dans le fichier **`php.ini`**

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- L'extension **PDO** offre 03 classes pour interagir avec les bases de données : il s'agit des classes `PDO`, `PDOStatement` et `PDOException`
- La classe `PDO` fourni un objet représentant une connexion à un SGBD. Cet objet permet entre autres ***d'exécuter*** des requêtes, de créer des ***requêtes préparées*** et de gérer des ***transactions***
- La classe `PDOStatement` fourni un objet représentant une ***requête préparée*** ou le ***résultat*** d'une requête de sélection
- La classe `PDOException` fourni un objet permettant de ***gérer les erreurs*** liées à l'exécution des requêtes

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- Pour se connecter en général avec PDO, on crée une instance de la classe **PDO** en lui fournissant un **DSN**, un *nom d'utilisateur* et un *mot de passe*
- Un **DSN** (**Data Source Name**) est une chaine de caractères décrivant comment se connecter à une base de données accessible par **fichier** ou à travers un **SGBD**
- Par convention, le DSN commence par le **nom du pilote de connexion** et est suivi d'une **liste de paramètres** qui dépendent de la source de données
- Le liste des paramètres, qui dépend de la source de données, inclut par exemple ***l'adresse IP du serveur*** (SGBD), le ***numéro de port***, le ***nom de la BD***, le ***type de socket*** à utiliser, le ***fichier physique*** de la BD, le ***jeu de caractères***, …

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- A titre d'illustration les DSN suivants permettent de se connecter à des bases de données *MariaDB*, *PostgreSQL*, *Oracle* et *SQLite*
- *MariaDB* : `<?php $dsn="mysql:host=127.0.0.1;port=3306;dbname=uy1"; ?>`
- *PostgreSQL* : `<?php $dsn="pgsql:host=127.0.0.1;port=5432;dbname=uy1"; ?>`
- *Oracle* : `<?php $dsn="oci:dbname=//127.0.0.1:1521/uy1"; ?>`
- *SQLite* : `<?php $dsn="sqlite:/c:/wamp/www/uy1.sqlite"; ?>`
- On se connecte à la BD en essayant de créer une instance de la classe PDO.
  `<?php $pdo_con = new PDO($dsn,"test","pwd"); ?>`

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- Si la tentative de connexion échoue, alors une exception sera levée !

```
<?php
  $dsn = "mysql:host=127.0.0.1;port=3306;dbname=uy1";
  try
  { $pdo_con = new PDO($dsn,"test","pwd"); }
  catch (PDOException $pdo_err)
  { echo "Echec de la connexion !<br/>".$pdo_err->getMessage(); }
?>
```

- Si l'exception n'est pas gérée dans un bloc **`try ... catch`**, alors elle génèrera une ***erreur fatale*** qui provoquera l'arrêt de l'exécution du script PHP !

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- En cas de succès, un objet **PDO** est renvoyé et celui-ci dispose de 02 méthodes pour l'exécution des requêtes : les méthodes **exec()** et **query()**.
- La méthode **exec()** est utilisée pour exécuter des *ordres de manipulation*. Elle renvoi un ***entier*** qui indique le *nombre d'enregistrements affectés*

```
<?php
  ...
  $sql = "INSERT INTO filière VALUES ('ict4d','fs')";
  $nb = $pdo_con->exec($sql);
?>
```

- Si l'entier renvoyé vaut « **-1** », alors la requête à probablement échouée !

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- La méthode **query()** est utilisée pour exécuter des *ordres de sélection*. Elle renvoi un objet **PDOStatement** qui contient le *résultat de la sélection* si la requête c'est bien exécutée ou **FALSE** sinon

```
<?php
  ...
  $sql = "SELECT * FROM étudiant ORDER BY matricule";
  $pdo_sta = $pdo_con->query($sql);
  if(!$pdo_sta)
  { echo "Echec de la requête : "; print_r($pdo_con->errorInfo());}
  ...
?>
```

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- Les méthodes **fetch()** et **fetchAll()** de la classe **PDOStatement** permettent d'accéder aux résultats de la sélection en renvoyant par défaut des tableaux avec clés *littérales* et *indicées*

```
<?php
  ...
  else {
    while($lg = $pdo_sta->fetch()) {
      foreach($lg as $chp=>$val)
      { echo "Champ : $chp – Valeur : $val"; }
    }
  } ...
?>
```

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- La méthode **`rowCount()`** permet de retourner le nombre de ligne de la sélection. **`<?php $n = $pdo_sta->rowCount(); ?>`**
- Après exploitation des résultats, **`closeCursor()`** permet de libérer la mémoire occupée par les résultats de la sélection. **`<?php $pdo_sta->closeCursor(); ?>`**
- La méthode **`lastInsertId()`** de classe **`PDO`** renvoie le dernier identifiant créé pour un champ **`AUTO_INCREMENT`**. **`<?php $id = $pdo_con->lastInsertId(); ?>`**
- On termine une connexion initiée en affectant la valeur **`NULL`** à l'objet **`PDO`** créé ! En cas *d'omission*, la connexion est *automatiquement fermée* à la fin du script. **`<?php $pdo_con = NULL; ?>`**

V) Accès à MariaDB avec PHP

Accès objet à MariaDB

- On peut utiliser **`quote()`** pour échapper les caractères spéciaux d'une chaine de caractère. **`<?php $ch_esc = $pdo_con->quote($ch); ?>`**
- Pour effectuer des transactions, on ***désactive d'abord le mode auto commit*** de MariaDB avec **`beginTransaction()`**. **`<?php $pdo_con->beginTransaction(); ?>`**
- La méthode **`commit()`** permet ensuite de valider toutes les commandes SQL effectuées. **`<?php ...; if(...) {$pdo_con->commit(); } ?>`**
- La méthode **`rollBack()`** permet d'annuler toutes les modifications effectuées. **`<?php ... else { $pdo_con->rollBack(); ...; } ?>`**

Conclusion

Le langage de script PHP :

- Offre de manière native les extensions **`php_myslq`** et **`php_mysqli`** pour accéder aux bases de données sous MySQL/MariaDB
- Offre les extensions du type **`php_pdo_xxxxx`** pour accéder uniformément aux bases de données *SQLite*, *MariaDB*, *PostgreSQL*, *Oracle*, *MS SQL Server*, …
- Est accompagné d'outils d'administration de base de données complet tels que *phpMyAdmin*, *phpPgadmin*, *SQLiteManager*, *SQLBuddy*, …

Bibliographie

Pour en savoir plus :

- **Soutou, Christian.** *Apprendre SQL avec MySQL.* Paris : Eyrolles, 2006. p. 418. ISBN : 2-212-11915-1.
- **Engels, Jean.** *PHP 5 – Cours et exercices.* 2e. Paris : Eyrolles, 2009. p. 662. ISBN : 978-2-212-12486-6.
- **Rigeaux, Philippe.** *Pratique de MySQL et PHP – Conception et réalisation de sites web dynamiques.* 4e. Paris : Dunod, 2009. p. 557. ISBN : 978-2-10-053752-5.
- **Oracle.** *MySQL 5.0 Reference Manual.* 4e. Oracle, 2011. p. 1243.
- **PHP.net**. « *PHP documentation* » [En ligne]. Disponible sur : http://www.php.net/