

# Web dynamique

---

Techniques, outils, applications

(Partie C)

# SOMMAIRE

---

1. MySQL et PHP (20 min)
2. SQL (petits rappels) (20 min)

# MySQL et PHP

---

Synthèse

# MySQL et PHP

---

## □ SGBD MySQL

- Libre ( GNU )
- Multiplateforme
- multi-utilisateurs



## □ Principe:

- Utilisation de requêtes SQL dans du code PHP
- API MySQL : Ensemble de fonctions PHP permettant d'accéder aux bases de données MySQL



# MySQL

---

- Différences entre MySQL et le standard
  - Pas de gestion des transactions
  - Pas de procédure stockée ni de trigger
  - Pas de clé étrangère (performance)
  - Pas de vue
  - Commentaires : # au lieu de --

# MySQL : DDL

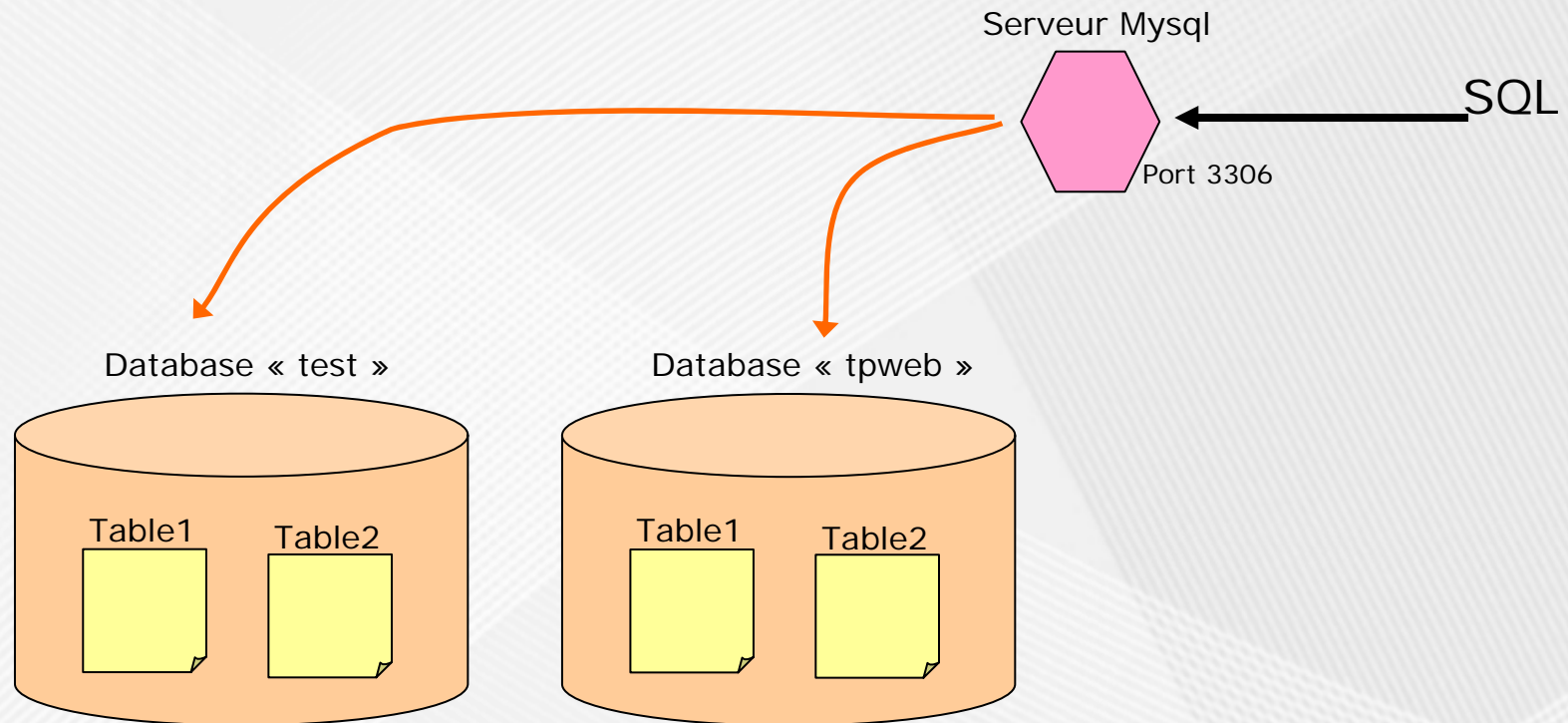
---

- Principaux types de base :
  - INT
  - DATETIME
  - VARCHAR
  - CHAR
  - FLOAT
  - DOUBLE

# MySQL: Organisation interne

---

- 1 serveur Mysql peut servir 1 ou plusieurs bases de données.
- Elles sont identifiées par un nom.
- 1 database = 1 groupe de tables
- 1 database constitue un peu des espaces de noms pour les tables.





# MySQL et PHP

---

## □ Connexion à un serveur MySQL :

- `$connection=mysql_connect("SERVEUR","UTILISATEUR","PASSWD");`
- Nom ou adresse IP du serveur
- Les 3 paramètres sont des chaînes ou des variables
- Écriture style Perl:  
`$connection=mysql_connect(...) or die("Erreur connexion MySQL");`

## □ Sélection d'une base de données :

- `mysql_select_db("BASE")` ou `mysql_select_db("BASE", $connection)`
- L'identifiant d'une base est une chaîne

## □ Exécution d'une requête SQL :

- `$resultat=mysql_query(REQUÊTE);`
- `$resultat` est l'ensemble résultat (un tableau de tableau)

## □ Exploitation des résultats

- Slide suivant

## □ Fermeture de la connexion

- Slide suivant



# PHP et MySQL: résultat SQL

---

- Nombre de lignes retournées par une requête :
  - `$nb=mysql_num_rows($resultat);`
- Ou nombre de lignes affectées par une requête:
  - `$nb=mysql_affected_rows($connexion)`
  - Concerne les requêtes INSERT, UPDATE ou DELETE.
  - **Attention c'est la connexion qu'il faut passer!**
  - `sinon $nb=mysql_affected_rows()` marche sur la dernière requête
- Accès aux attributs d'une ligne du résultat:
  - `$col=mysql_result($resultat, <NO_LIGNE>, "NOM_CHAMP");`
- Autre méthode: `mysql_fetch_row()`
  - Récupère la ligne suivante du résultat dans un tableau
  - `$ligne = mysql_fetch_row($resultat);`
  - Ensuite `$ligne[x]` est le X<sup>e</sup> attribut précisé dans le select
- Autre méthode: `mysql_fetch_array ()`
  - Récupère la ligne suivante du résultat dans une liste associative
  - Idem `mysql_fetch_row()` mais accès par nom de colonne
- Fermeture de la connexion :
  - `mysql_close();`
- Affichage du code en cas d 'erreur :
  - `echo "ERREUR : ". mysql_error(). "<BR>";`

# PHP et MySQL: exemple type 1

---

```
/* connexion à la base de donnée */
$dbhost = "localhost";
$user = "root";
$password = "";
$usebdd = "test";

$connexion = mysql_connect($dbhost,$user,$password);
if ($connexion) {
    //Ensuite selectionner la base sur laquelle travailler
    $db = mysql_select_db($usebdd, $connexion);
    if ($db) {
        $result = mysql_query("SELECT * FROM messagerie",$connexion);
        if($result){
            $nombreligne = mysql_num_rows($result);
            for($i=0; $i<$nombreligne; $i++){
                $nom = mysql_result($result,$i,"nom");
                $email = mysql_result($result,$i,"email");
                printf("nom      : $nom <br>\nEmail   : $email <br>\n");
            }
        }
    }
    mysql_close($connexion);
}
```

# PHP et MySQL: exemple type 2

---

```
/* connexion à la base de donnée */
$dbhost = "localhost";
$user = "root";
$password = "";
$usebdd = "test";

$connexion = mysql_connect($dbhost,$user,$password);
if ($connexion) {
    //Ensuite selectionner la base sur laquelle travailler
    $db = mysql_select_db($usebdd, $connexion);
    if ($db) {
        $requeteSQL="INSERT INTO messagerie(nom, email) «
            . "values ('$nom', '$email)";
        If(mysql_query($requeteSQL,$connexion))
        echo "Lignes modifiees : %d\n", mysql_affected_rows();
        else
        /* affiche un eventuelle erreur, utile en cas d'erreur */
        echo mysql_error();
    } else echo mysql_error(); //notez l'utilisation de mysql_error()
    mysql_close($connexion);
} else echo mysql_error();
```



# SQL

---

Petits rappels

# SQL

---

- SQL est un langage standard (ISO 9075) pour manipuler et accéder aux données
  
- Trois groupes de requêtes :
  - Les ordres DDL (tables, vues, index... )
  - Les ordres DML (insert,update,delete,select)
  - Les ordres DCL (Data Control Language)
    - droits d'accès données/utilisateurs

# SQL: DDL

---

Un exemple:

```
CREATE TABLE CLIENT (  
    NumCli NUMBER(3),  
    Nom CHAR(30),  
    DateNaiss DATE,  
    Salaire NUMBER(8,2),  
    NumEmp NUMBER(3),  
    CONSTRAINT cle_pri PRIMARY KEY (NumCli),  
    CONSTRAINT cle_etr FOREIGN KEY (NumEmp)  
    REFERENCES EMPLOYEUR(NumEmp),  
    CONSTRAINT date_ok CHECK (DateNaiss < SYSDATE)  
);
```



# SQL : DML

---

## □ Ajout

- ex. INSERT INTO PRODUIT VALUES (400, 'Nouveau produit', 78.90);

## □ Mise à jour

- ex. UPDATE CLIENT SET Nom='Dudule' WHERE NumCli = 3;

## □ Suppression

- ex. DELETE FROM CLIENT WHERE Ville = 'Lyon';

# SQL : DML

---

## □ Recherche

### ■ Sans filtrage

□ `SELECT * FROM CLIENT;`

### ■ Tri du résultat

□ `SELECT * FROM CLIENT ORDER BY Nom;`

### ■ Calculs horizontaux et verticaux

□ `SELECT PrixUni+PrixUni*0.206 FROM PRODUIT;`

□ `SELECT COUNT(*) from CLIENT`

# SQL : DML

---

## Projection

- `SELECT Nom, Prenom FROM CLIENT;`
- Opérateur `DISTINCT`

## Filtre

- `SELECT * FROM CLIENT WHERE Ville = 'Lyon';`
- Operateurs :
  - `< > =`
  - `Between .. And ..`
  - `Is null`
  - `Like et %`
  - `in`



# SQL : DML

---

- **Calcul vertical ( agrégats )**
  - AVG() : moyenne des valeurs
  - SUM() : somme des valeurs
  - MIN(), MAX() : minimum, maximum
  - COUNT() : nombre de valeurs

# SQL : DML

---

## **Non couvert :**

- Jointures
- sous requêtes
  - IN
  - EXISTS
  - ALL
  - ANY
  - NOT
- Groupements
  - Group By
  - Having ...