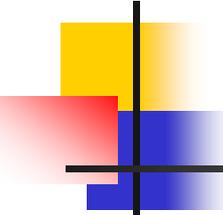


PHP 5

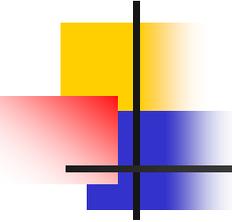
La base de données MySQL



Base de données

- C'est quoi une base de données ?
 - Une base de données contient une ou plusieurs tables, chaque table ayant un ou plusieurs enregistrements
- Exemple de table :

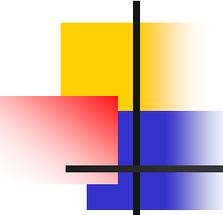
```
-- TABLE : contacts
NOM      PRENOM      ADRESSE      VILLE      CP      PAYS
-----
Alpha    Alain        1 rue AAA    Paris      75000   FRANCE
Beta     Benjamin     34 rue G.    Lyon       69000   FRANCE
Gamma    Gaëlle       3, bakerstreet London     1234    U.K.
```



Base de données

■ La gestion de la base

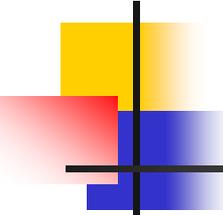
- est rendue possible grâce à un Système de Gestion de Base de Données (S.G.B.D)
- Il a pour rôle de :
 - manipuler, compléter, mettre à jour les données dans les tables
 - et de pouvoir les interroger via des *requêtes*, par le langage **SQL** (Structured Query Language : Langage de requêtes structurées)
- Il existe plusieurs S.G.B.D., certains gratuits, d'autres payants : pour les gratuits, on trouve **MySQL**



Base de données

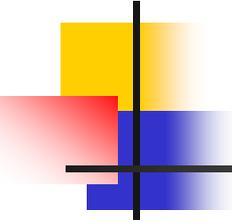
- Les champs d'une table de base de données
 - correspondent à une des variables qui composent un enregistrement, par exemple **"PRENOM"**
 - sont caractérisés par une nature, ils permettent, pour certains, d'identifier avec précision une valeur ou un groupe de valeurs
 - le champ peut être de plusieurs types, pour les 3 principaux :
 - texte, valeur numérique ou valeur binaire

```
-- TABLE : contacts
NOM      PRENOM      ADRESSE      VILLE      CP      PAYS
-----
Alpha    Alain        1 rue AAA    Paris      75000    FRANCE
Beta     Benjamin     34 rue G.    Lyon       69000    FRANCE
Gamma    Gaëlle       3, bakerstreet London     1234     U.K.
```



Base de données

- Les champs d'une table de base de données
 - **Champ numérique**
 - **Int** : de -2 147 483 648 à 2 147 483 647
 - En mettant l'attribut **UNSIGNED**, on stocke de 0 à 4 294 967 295 (on ne se préoccupe plus du signe avec le paramètre **UNSIGNED**)
 - Il existe **INT**, **TINYINT**, **BIGINT**, **DOUBLE** qui peuvent contenir des nombres plus ou moins grands
 - Pour les réels : **FLOAT**



Base de données

- **Champ texte**
 - **VARCHAR** pour stocker au plus **255** caractères
 - **TEXT** peut stocker **65535** caractères
 - **LONGTEXT** : **4 294 967 295** ($2^{32} - 1$) caractères...
 - **ENUM** : présente une énumération. Par exemple, **ENUM('rouge', 'bleu', 'vert')** assurera que la valeur du champ sera l'une des 3 de la liste
- **Champs de type date et heure**
 - **DATE** permet d'effectuer des opérations sur les dates dans les requêtes, plutôt que de stocker la date sous forme de texte
- **Autres types de champs**
 - On peut stocker des images, ou encore des fichiers complets dans une base de données : en les encodant dans un champ de type **BLOB** (binaire) par exemple

Base de données

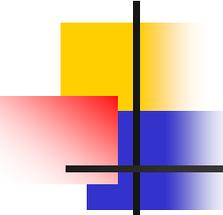
– Clé primaire (Primary Key)

- La clé primaire est **une propriété** d'un des champs pour différencier des enregistrements identiques
- Exemple d'ambiguïté :

```
-- TABLE : contacts
NOM      PRENOM    VILLE    CP      ID_PAYS
-----
Alpha    Alain     Paris    75000   1
Beta     Benjamin  Lyon     69000   1
Beta     Benjamin  Lyon     69000   1
Gamma    Gaëlle    London   1234    2
```

```
-- TABLE : pays
ID      PAYS
-----
1       FRANCE
2       UNITED KINGDOM
```

- Pour distinguer les deux enregistrements de "Benjamin", on attribue un numéro à chaque individu, généralement appelé "identifiant" et abrégé sous la forme "id" (ou "ID")
 - ❖ **Cet identifiant sera "auto incrémenté"**



Base de données

– La solution

```
-- TABLE : contacts
ID      NOM      PRENOM  VILLE  CP      ID_PAYS
-----
1       Alpha   Alain   Paris  75000   1
2       Beta    Benjamin Lyon   69000   1
3       Beta    Benjamin Lyon   69000   1
4       Gamma   Gaëlle  London 1234    2
```

– Attention

- Toutes les valeurs des clés doivent être uniques : il ne doit pas, pour une même table, y avoir 2 valeurs égales de clés primaires
- On peut demander à voir le contact dont l'ID est 2 ou 3, on a bien 2 personnes distinctes

Base de données

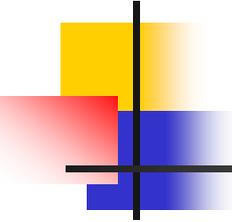
– Clé étrangère (Foreign Key)

- La clé étrangère est un champ dans une table qui est clé primaire dans une autre table
- Pour l'exemple, on peut très bien spécifier que **ID_PAYS** est une clé étrangère liée au champ **ID** de la table pays

```
-- TABLE : contacts
ID      NOM      PRENOM  VILLE  CP      ID_PAYS
-----
1       Alpha   Alain   Paris  75000   1
2       Beta    Benjamin Lyon   69000   1
3       Beta    Benjamin Lyon   69000   1
4       Gamma   Gaëlle  London 1234    2
```

```
-- TABLE : pays
ID      PAYS
-----
1       FRANCE
2       UNITED KINGDOM
```

- L'existence d'une clé étrangère dans les S.G.B.D.R. permet de mettre à jour des enregistrements en cascade, par exemple, supposons que j'efface le pays "France", je dois supprimer tous les contacts qui y vivent (puisqu'ils ne seront alors rattachés à aucun pays)



MySQL

■ Administration de MySQL

- L'administration peut se faire de deux manières
 - par PHP (voir plus loin)
 - Manuellement, par l'interface **phpMyAdmin** d'EasyPhp
- L'interface **phpMyAdmin**
 - Pour y accéder :
 - ❖ Cliquer sur le bouton droit de la souris en le mettant sur le symbole d'Easyphp puis appuyer sur **Administration**
 - ❖ Ensuite, appuyer sur **mysql Administration**

PHPMYADMIN 3.1.1

MYSQL ADMINISTRATION

LICENCE

MySQL

- On obtient

The screenshot displays the phpMyAdmin 2.6.1 interface. At the top, it says "Welcome to phpMyAdmin 2.6.1" and "MySQL 4.1.9-max running on localhost as root@localhost". The interface is divided into three main sections:

- Left Sidebar:** Shows the "Database:" dropdown set to "mysql (6)". Below it, a tree view shows the "mysql" database with its tables: columns_priv, db, func, host, tables_priv, and user.
- Central Navigation Area:** Contains a "MySQL" section with a "Create new database" form (with a "Collation" dropdown and a "Create" button) and a list of links: "Show MySQL runtime information", "Show MySQL system variables", "Show processes", "Character Sets and Collations", "Privileges", "Databases", and "Export".
- Right Sidebar:** Contains a "phpMyAdmin" section with settings: "Language" (English (en-utf-8)), "MySQL charset" (UTF-8 Unicode (utf8)), "MySQL connection collation" (utf8_general_ci), and "Theme / Style" (Original). It also includes links for "phpMyAdmin documentation", "Show PHP information", and "Official phpMyAdmin Homepage" (with sub-links for ChangeLog, CVS, and Lists).

La base **mysql** et
ses tables

Les bases existantes

Informations générales sur le
gestionnaire **phpMyAdmin**

MySQL

- En cliquant sur Databases : on obtient les bases enregistrées
 - On peut cocher et supprimer une base par Drop
 - On peut créer une nouvelle base

The screenshot shows the MySQL Databases management interface. At the top, there is a navigation bar with buttons for Databases, Status, Variables, Charsets, Privileges, Processes, and Export. The main content area is titled "Databases" and contains a table with two rows: "espv3" and "mysql". Each row has a checkbox on the left and a small icon on the right. Below the table, there is a link "Check All / Uncheck All". Underneath, there is a section "Enable Statistics" with a note: "Note: Enabling the Database statistics here might cause heavy traffic between the webserver and the MySQL one." Below that, there is a section "Drop Selected Databases" with a "Drop" button. At the bottom, there is a section "Create new database" with a text input field, a "Collation" dropdown menu, and a "Create" button.

	Database ▲	
<input type="checkbox"/>	espv3	
<input type="checkbox"/>	mysql	

↑ [Check All](#) / [Uncheck All](#)

Enable Statistics
Note: Enabling the Database statistics here might cause heavy traffic between the webserver and the MySQL one.

Drop Selected Databases

Create new database

Collation

MySQL

- En cliquant sur la base mySql, on obtient ses tables
 - On peut supprimer une table en cochant et supprimant
 - On peut créer une nouvelle table

The screenshot shows the MySQL database structure view for the 'mysql' database. The interface includes a toolbar with buttons for Structure, SQL, Export, Search, Query, Operations, and Drop. Below the toolbar is a table listing the database's tables and their properties.

Table	Action	Records	Type	Collation	Size	Overhead
<input type="checkbox"/> columns_priv		0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> db		1	MyISAM	latin1_swedish_ci	3.1 KB	-
<input type="checkbox"/> func		0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> host		0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> tables_priv		0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> user		1	MyISAM	latin1_swedish_ci	2.1 KB	108 Bytes
6 table(s)	Sum	2	--	latin1_swedish_ci	9.3 KB	108 Bytes

Check All / Uncheck All / Check overheded With selected:

Print view Data Dictionary

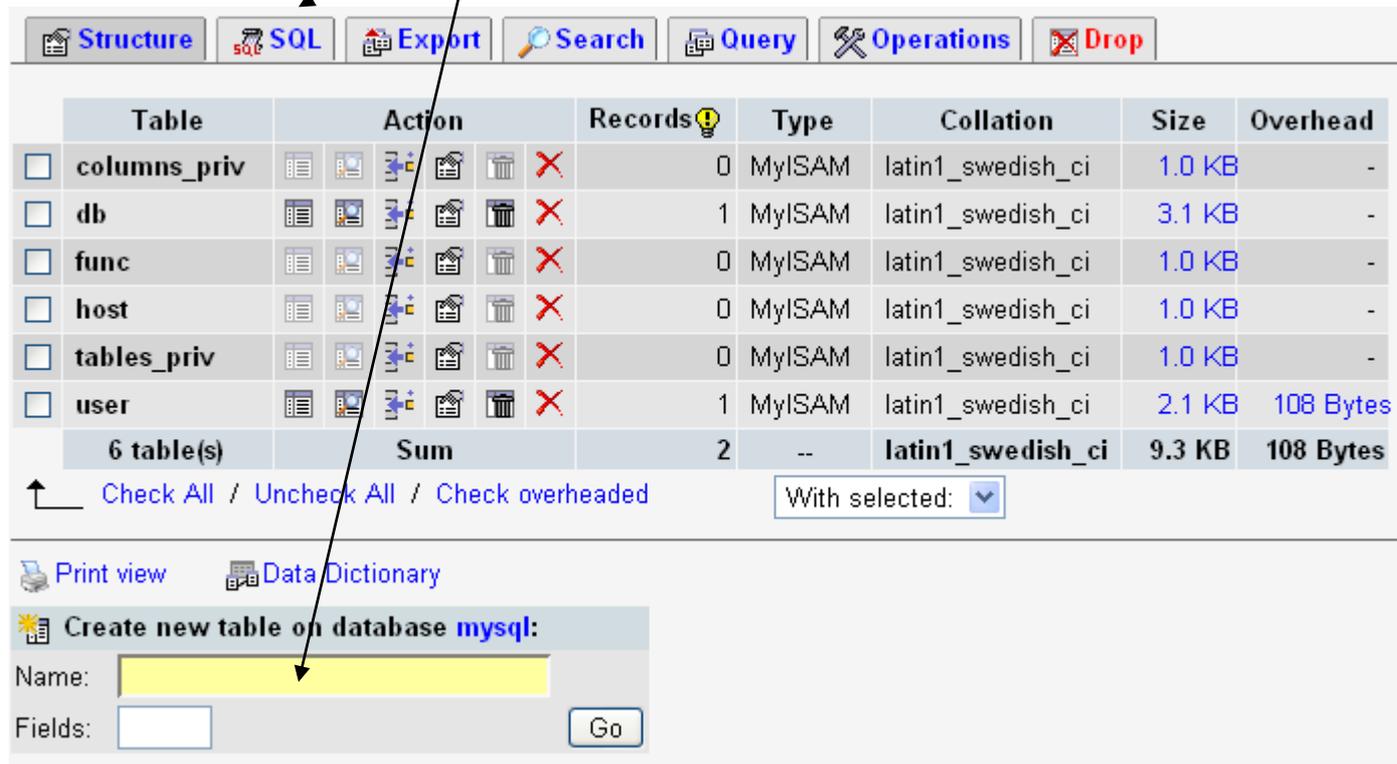
Create new table on database **mysql**:

Name:

Fields:

MySQL

- Retour sur la création d'une base : 2 méthodes



The screenshot shows the MySQL database structure view. At the top, there are tabs for Structure, SQL, Export, Search, Query, Operations, and Drop. Below the tabs is a table listing the database tables. The table has columns for Table, Action, Records, Type, Collation, Size, and Overhead. The 'user' table is selected, and a dialog box titled 'Create new table on database mysql:' is open, with the 'Name' field highlighted in yellow. An arrow points from the 'SQL' tab to the 'Name' field.

Table	Action	Records	Type	Collation	Size	Overhead
<input type="checkbox"/> columns_priv	      	0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> db	      	1	MyISAM	latin1_swedish_ci	3.1 KB	-
<input type="checkbox"/> func	      	0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> host	      	0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> tables_priv	      	0	MyISAM	latin1_swedish_ci	1.0 KB	-
<input type="checkbox"/> user	      	1	MyISAM	latin1_swedish_ci	2.1 KB	108 Bytes
6 table(s)	Sum	2	--	latin1_swedish_ci	9.3 KB	108 Bytes

Check All / Uncheck All / Check overheaded With selected: ▾

Print view Data Dictionary

Create new table on database mysql:

Name:

Fields: Go

Base de données

■ Création d'une base personnelle

- Commençons par créer une base de données : **ma-base**
 - Laissez le type en Interclassement par défaut
 - ma-base(0) indique que la base contient 0 table
 - Regardez le code SQL qui s'affiche

The screenshot shows the phpMyAdmin interface. On the left sidebar, the 'Base de données' section is expanded, and 'ma-base (0)' is selected. The main content area shows the 'Base de données: ma-base' page. The 'Structure' tab is active, and a message states 'Base de données ma-base a été créé(e)'. Below this, the 'requête SQL:' section displays the SQL command: `CREATE DATABASE `ma-base` ;`. At the bottom, there is a section for 'Créer une nouvelle table' with input fields for 'Nom:' and 'Nombre de champs:'.

Création d'une base personnelle

■ Création d'une table

- Créer dans ma-base, la table « **contacts** » suivante avec ses 5 champs:

```
-- TABLE : contacts
NOM      PRENOM    VILLE    CP      ID_PAYS
-----
Alpha    Alain     Paris    75000   1
Beta     Benjamin  Lyon     69000   1
Beta     Benjamin  Lyon     69000   1
Gamma    Gaëlle    London   1234    2
```

Aucune table n'a été trouvée dans cette base.

📄 Créer une nouvelle table sur la base [ma-base](#)

Nom: Nombre de champs:

- Cette action conduit à afficher le premier enregistrement de la table **contacts** avec les 5 champs en attente

127.0.0.1 ▶ ma-base ▶ contacts

Champ	Type ②	Taille/Valeurs ¹	Défaut ²	Interclassement	Attributs	Null	Index
<input type="text"/>	INT	<input type="text"/>	Aucun	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	—
<input type="text"/>	INT	<input type="text"/>	Aucun	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	—
<input type="text"/>	INT	<input type="text"/>	Aucun	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	—
<input type="text"/>	INT	<input type="text"/>	Aucun	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	—
<input type="text"/>	INT	<input type="text"/>	Aucun	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	—

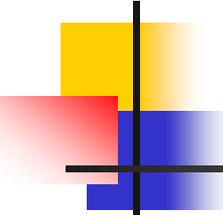


Table MySQL

■ Les champs

- Le **nom du champ**
 - ne peut pas être nul, ni comporter de caractères accentués, espaces, -, par contre, il accepte _
- Le type de champ est sélectionné dans une liste de choix
 - pour l'instant, sélectionner **VARCHAR** qui correspond à du texte
- La **taille/valeurs**
 - permet de déterminer la taille max des données à rentrer dans les champs
- **Interclassement**
 - représente le jeu de caractères par défaut si vous ne sélectionnez rien (préférable)
 - sinon sélectionner **Latin1_swedish_ci**. Si le jeu de caractères n'est pas démarré dans phpMyAdmin, une erreur est produite
- L'**attribut** peut-être rien, unsigned ou Unsigned zerofill
 - les 2 dernières valeurs s'utilisent uniquement pour des types de champs spéciaux
- **Null** permet de dire si le champ peut-être null (vide) ou non
- **Default** permet de donner une valeur par défaut au champ

Table MySQL

■ Les options

-  – permet de déterminer si le **champ** est ou non une **clé primaire**
 - dans ce cas, la valeur du champ sera unique dans la table
 - aucun enregistrement n'aura la même valeur dans ce champ
-  – **index** indique que le champ peut être utilisé comme une valeur de repérage (index)
 - permet d'augmenter les vitesses de traitement (tri) dans la table
 - la clé primaire est également un index
-  – **unique** permet de spécifier que le **contenu** est unique dans l'ensemble du champ
 - évite les doublons
-  – permettent d'indexer des textes complets
 - cette possibilité ralentit la table pour de longs champs

Table MySql

■ Création de champs

- Créez 5 champs de type VARCHAR (caractères de taille variables)
 - code, de taille 10, non null et unique
 - titre, de taille 30, non null et unique
- N'oubliez pas de sauvegarder la table

```
-- TABLE : contacts
NOM      PRENOM    VILLE     CP      ID_PAYS
-----
Alpha    Alain     Paris     75000   1
Beta     Benjamin Lyon      69000   1
Beta     Benjamin Lyon      69000   1
Gamma    Gaëlle   London    1234    2
```

Remplissage du premier enregistrement

Field	Type?	Length/Values*	Collation	Attributes	Null	Default**	Extra			
NOM	VARCHAR				not null	Alpha				
PRENOM	VARCHAR				not null	Alain				
VILLE	VARCHAR				not null	Paris				
CP	VARCHAR				not null	75000				
ID_PAYS	VARCHAR				not null	1				

■ On obtient

The screenshot shows the phpMyAdmin interface for a table named 'contacts'. The top navigation bar includes buttons for Structure, Browse, SQL, Search, Insert, Export, Operations, Empty, and Drop. A message at the top states 'Table contacts has been altered.' with the SQL query: `ALTER TABLE `contacts` CHANGE `CP` `CP` INT(10) DEFAULT '75000' NOT NULL`. Below this is a table listing the fields: NOM, PRENOM, VILLE, CP, and ID_PAYS. The CP field is highlighted in blue. Below the table are options for 'Check All / Uncheck All' and 'With selected:'. Further down, there are options for 'Print view', 'Propose table structure', and 'Add 1 field(s)'. Below that is the 'Indexes' section, which shows 'No index defined!' and a 'Go' button. To the right of the indexes are 'Space usage' and 'Row Statistic' sections. At the bottom, there is a 'Run SQL query/queries on database ma-base' section with a text area containing 'SELECT * FROM `contacts` WHERE 1' and a 'Fields:' list on the right containing NOM, PRENOM, VILLE, CP, and ID_PAYS. A 'Go' button is at the bottom right.

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> NOM	varchar(10)	latin1_swedish_ci		No	Alpha		
<input type="checkbox"/> PRENOM	varchar(20)	latin1_swedish_ci		No	Alain		
<input type="checkbox"/> VILLE	varchar(10)	latin1_swedish_ci		No	Paris		
<input type="checkbox"/> CP	int(10)			No	75000		
<input type="checkbox"/> ID_PAYS	tinyint(2)			No	1		

Check All / Uncheck All With selected:

Print view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After NOM

Indexes: No index defined!
Create an index on 1 columns

Space usage:

Type	Usage
Data	0 Bytes
Index	1,024 Bytes
Total	1,024 Bytes

Row Statistic:

Statements	Value
Format	dynamic
Collation	latin1_swedish_ci
Rows	0
Creation	Mar 11, 2008 at 09:51 PM
Last update	Mar 11, 2008 at 09:51 PM

Run SQL query/queries on database ma-base

SELECT * FROM `contacts` WHERE 1

Fields:
NOM
PRENOM
VILLE
CP
ID_PAYS

Show this query here again

Pour modifier,
supprimer un ou
tous les champs

Pour ajouter un
champ

Table MySql

■ Ajout d'enregistrement : insert

Server: localhost Database: ma-base Table: contacts

Structure Browse SQL Search Insert Export

Field	Type	Function	Null	Value
NOM	varchar(10)			Alpha
PRENOM	varchar(20)			Alain
VILLE	varchar(10)			Paris
CP	int(10)			75000
ID_PAYS	tinyint(2)			1

Ignore

Field	Type	Function	Null	Value
NOM	varchar(10)			Beta
PRENOM	varchar(20)			Benjamin
VILLE	varchar(10)			Lyon
CP	int(10)			69000
ID_PAYS	tinyint(2)			1

Insert as a new row -- And -- Go back to previous page
Or Insert another new row



Table MySQL

- Appuyer sur Browse pour voir les enregistrements

Server: localhost Database: ma-base Table: contacts

Structure Browse SQL Search Insert Export

Showing rows 0 - 1 (2 total, Query took 0.0004 sec)

SQL-query:
SELECT *
FROM `contacts`
LIMIT 0, 30

[Edit] [Explain SQL] [Create PHP Code] [Refresh]

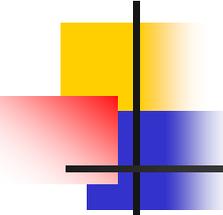
Show: 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

	NOM	PRENOM	VILLE	CP	ID_PAYS
<input type="checkbox"/>	Beta	Benjamin	Lyon	69000	1
<input type="checkbox"/>	Alpha	Alain	Paris	75000	1

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Insert new row Print view Print view (with full texts) Export



Commandes SQL

- Création de table

CREATE TABLE

- Syntaxe

```
CREATE table nomdetable(  
    nomattribut1 type[not null][AUTO_INCREMENT]  
    nomattribut2 type[not null]  
    ...  
    nomattributN type[not null]  
    ...  
    primary key (nomattribut1, ..., nomattributN)  
    ;
```

- Auto_increment : l'entier sera augmenté de 1 à chaque nouvelle insertion

Commandes SQL

- Création de table : appuyer sur SQL et rentrer ce code



Ci-dessous la table : **clients_tbl** une fois créée

```
+++++  
+  id  +  prenom  +  nom  +  ne_le  +  ville  +  enfants  +  
+++++
```

Bien sûr pour le moment cette table ne comporte aucun enregistrement ...

Commandes SQL

■ Modification de la table

- Une fois créée, on peut la modifier en utilisant **ALTER TABLE** et **ADD**

Syntaxe SQL pour ajouter le champ 'tel' à la table : **clients_tbl**

```
ALTER TABLE clients_tbl ADD tel INT not null
```

Ci-dessous la table **clients_tbl** une fois modifiée

```
+++++  
+ id + prenom + nom + ne_le + ville + enfants + tel +  
+++++
```

- on peut la modifier en utilisant **ALTER TABLE** et **DROP**

Syntaxe SQL pour supprimer le champ 'tel' de la table : **clients_tbl**

```
ALTER TABLE clients_tbl DROP tel
```

Ci-dessous la table **clients_tbl** une fois modifiée

```
+++++  
+ id + prenom + nom + ne_le + ville + enfants +  
+++++
```

Commandes SQL

- D'ailleurs, en voulant supprimer directement le champ, en cliquant sur la croix, voici ce qui s'affiche :



Commandes SQL

- **Modification de la table**
 - On peut la supprimer totalement par :
 - **DROP TABLE** clients_tbl



Commandes SQL

■ Insertion d'enregistrements

– INSERT INTO

`clients_tbl(id,prenom,nom,ne_le,ville,enfants)`

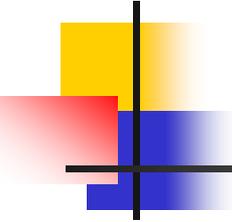
`VALUES('','Patrick','Martin','1965-10-08','Bordeaux','2')`

Devient visible

The screenshot shows a database management tool interface. At the top, it displays the server as 'localhost', the database as 'ma-base', and the table as 'clients_tbl'. Below this is a toolbar with buttons for 'Structure', 'Afficher', 'SQL', 'Rechercher', 'Insérer', 'Exporter', and 'Opérations'. The main area shows the results of an SQL query: 'Affichage des enregistrements 0 - 0 (1 total, traitement: 0.0005 sec.)'. The query is 'SELECT * FROM 'clients_tbl' LIMIT 0 , 30'. Below the query, there are buttons for '[Modifier]', '[Expliquer SQL]', '[Créer source PHP]', and '[Actualiser]'. At the bottom, there are input fields for 'Afficher : 30' and 'ligne(s) à partir de l'enregistrement n° 0', and a dropdown menu for 'en mode horizontal' and a field for 'et répéter les en-têtes à chaque groupe de 100'. Below this is a table with columns 'id', 'prenom', 'nom', 'ne_le', 'ville', 'enfants', and 'tel'. The table contains one row: '1 Patrick Martin 1965-10-08 Bordeaux 2 0'. There are also icons for editing, deleting, and inserting rows.

La requête d'affichage une fois l'insertion faite

L'enregistrement



Commandes SQL

- Insertion d'enregistrements : de manière générale
 - Syntaxe
 - `INSERT INTO votre_table (nomattribut1,...,nomattributN) values ('valeur1',..., 'valeurN');`
 - Les attributs qui ne sont pas mentionnés prendront leur valeur par défaut
 - Exemple
 - Soit la table Client-tbl (Id,nom,prenom,adresse) créée par :

```
create table Client (  
  id tinyint(3) unsigned not null AUTO_INCREMENT,  
  nom varchar(10) not null default "",  
  prenom varchar(10) not null default "",  
  adresse varchar(50) not null default "",  
  primary key (id)
```

Commandes SQL

■ Insertion d'enregistrements : de manière générale

– Remarque

- La clef ID possède l'option auto_increment, les ID vont ainsi être créés automatiquement
- En théorie, les numéros d'ID s'incrémentent par ordre croissant, mais si vous supprimez un enregistrement, la prochaine insertion d'un enregistrement se fera sur l'ID de l'enregistrement auparavant supprimé
- Complétez la table ainsi :

←T→	id	prenom	nom	ne_le	ville	enfants
<input type="checkbox"/>  	1	Patrick	Martin	1965-10-08	Bordeaux	2
<input type="checkbox"/>  	2	Julien	Lebreton	1964-02-21	Paris	2
<input type="checkbox"/>  	3	Marc	Richard	1958-04-15	Lille	4
<input type="checkbox"/>  	4	Francis	Perrin	1982-12-05	Paris	0
<input type="checkbox"/>  	5	Daniel	Bacon	1974-07-13	Reims	1

Commandes SQL

■ Sélection d'information d'enregistrements

- affichez uniquement les personnes qui n'ont que 2 enfants, la requête SQL sera :

```
SELECT * FROM clients_tbl WHERE enfants='2'
```

```
+++++
+ 1 + Patrick + Martin + 1965/10/08 + Bordeaux + 2 +
+++++
+ 2 + Julien + Lebreton + 1964/02/21 + Paris + 2 +
+++++
```

- demander l'affichage que des noms et prénoms :

```
SELECT nom, prenom FROM clients_tbl WHERE
enfants='2'
```

```
+++++
+ Patrick + Martin +
+++++
+ Julien + Lebreton +
+++++
```

Commandes SQL

– Sélection combinée :

- `SELECT * FROM clients_tbl WHERE enfants='1' OR enfants='2' AND ville='Paris'`

Serveur: localhost ▶ Base de données: ma-base ▶ Table: clients_tbl

Structure Afficher SQL Rechercher Insérer Exporter Opérations

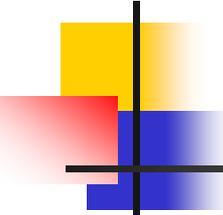
Affichage des enregistrements 0 - 1 (2 total, traitement: 0.0197 sec.)

requête SQL:
`SELECT *
FROM clients_tbl
WHERE enfants = '1'
OR enfants = '2'
AND ville = 'Paris'
LIMIT 0 , 30`

[Modifier] [Expliquer SQL] [Créer source PHP] [Actualiser]

Afficher : 30 ligne(s) à partir de l'enregistrement n° 0
en mode horizontal et répéter les en-têtes à chaque groupe de 100
Trier sur l'index: aucune Exécuter

	id	prenom	nom	ne_le	ville	enfants
<input type="checkbox"/>	2	Julien	Lebreton	1964-02-21	Paris	2
<input type="checkbox"/>	5	Daniel	Bacon	1974-07-13	Reims	1



Commandes SQL

- Autres formes de sélection

```
SELECT * FROM clients_tbl WHERE ne_le < "1978-01-01"
```

Sélection des personnes ayant une date de naissance plus petite que **1978/01/01**.

```
SELECT * FROM clients_tbl WHERE enfants != '0'
```

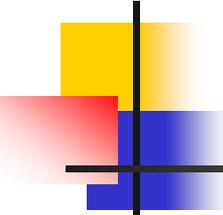
Sélection des personnes dont le nombre des enfants est différent de "0".

```
SELECT * FROM clients_tbl WHERE nom LIKE 'le%'
```

Sélection des personnes dont le nom commence par "le".

```
SELECT * FROM clients_tbl WHERE nom LIKE '%ri%'
```

Sélection des personnes qui ont dans leur nom la syllabe "ri".



Commandes SQL

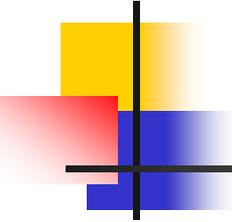
■ La commande Update

- UPDATE clients_tbl SET prenom='Jacques' WHERE id=1

```
+++++  
      + id + prenom + nom + ne_le + ville + enfants +  
+++++  
      + 1 + Jacques + Martin + 1965/10/08 + Bordeaux + 2 +  
+++++
```

- UPDATE clients_tbl SET prenom='Jean-Pierre',
nom='Papin', ville='Marseille', enfants=3 WHERE id=1

```
+++++  
      + id + prenom + nom + ne_le + ville + enfants +  
+++++  
      + 1 + Jean-Pierre+ Papin + 1965/10/08 + Marseille + 3 +  
+++++
```



Commandes SQL

- La commande DELETE

DELETE FROM `clients_tbl` WHERE `id=1`

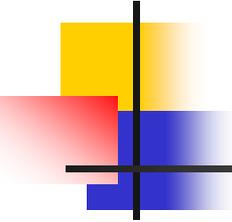
- supprime des enregistrements

```
DELETE FROM  
clients_tbl
```

Effacer de la table **Clients_tbl**

```
WHERE id=1
```

Quand l'id de l'enregistrement est **égal à 1**



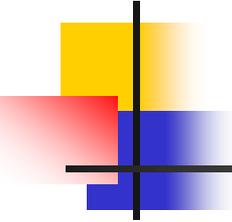
Commandes SQL

■ Jointure de tables

- Soient les deux tables :

<i>Livres</i>			
IDLivre	Prix	Titre	CodeGenre
1	40	Le glaive magique	BD
3	40	Gaffes en gros	BD
4	40	Lagaffe nous gâte	BD
5	45	QRN sur Bretzelburg	BD
6	80	Tour de manège	RG
7	45	Le spectre aux balles d'or	BD
8	30	La bonne chanson	Po
9	50	La jeune Parque	Po
10	50	Michel Strogoff	Ro
11	50	La Serpe d'or	BD
12	70	Toujours aimer	RG
13	70	Toujours aimante	RG
14	72	Toujours aimé	RG

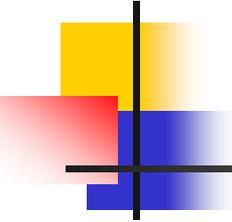
<i>Genres</i>	
CodeGenre	LibelleGenre
BD	Bande Dessinée
Po	Poésie
RG	Roman de gare
Ro	Roman
SF	Science Fiction



Jointure de tables

```
SELECT * FROM Livres, Genres WHERE  
Livres.CodeGenre = Genres.CodeGenre;
```

IDLivre	Prix	Titre	Livres.CodeGenre	Genres.CodeGenre	LibelleGenre
1	40	Le glaive magique	BD	BD	Bande Dessinée
3	40	Gaffes en gros	BD	BD	Bande Dessinée
4	40	Lagaffe nous gâte	BD	BD	Bande Dessinée
5	45	QRN sur Bretzelburg	BD	BD	Bande Dessinée
6	80	Tour de manège	RG	RG	Roman de gare
7	45	Le spectre aux balles d'or	BD	BD	Bande Dessinée
8	30	La bonne chanson	Po	Po	Poésie
9	50	La jeune Parque	Po	Po	Poésie
10	50	Michel Strogoff	Ro	Ro	Roman
11	50	La Serpe d'or	BD	BD	Bande Dessinée
12	70	Toujours aimer	RG	RG	Roman de gare
13	70	Toujours aimante	RG	RG	Roman de gare
14	72	Toujours aimé	RG	RG	Roman de gare



Jointure de tables

```
SELECT Titre, Livres.CodeGenre, LibelleGenre, Prix
FROM Livres, Genres WHERE Livres.CodeGenre =
Genres.CodeGenre;
```

Titre	Livres.CodeGenre	LibelleGenre	Prix
Le glaive magique	BD	Bande Dessinée	40
Gaffes en gros	BD	Bande Dessinée	40
Lagaffe nous gâte	BD	Bande Dessinée	40
QRN sur Bretzelburg	BD	Bande Dessinée	45
Tour de manège	RG	Roman de gare	80
Le spectre aux balles d'or	BD	Bande Dessinée	45
La bonne chanson	Po	Poésie	30
La jeune Parque	Po	Poésie	50
Michel Strogoff	Ro	Roman	50
La Serpe d'or	BD	Bande Dessinée	50
Toujours aimer	RG	Roman de gare	70
Toujours aimante	RG	Roman de gare	70
Toujours aimé	RG	Roman de gare	72

Utilisation avec PHP

■ Précautions

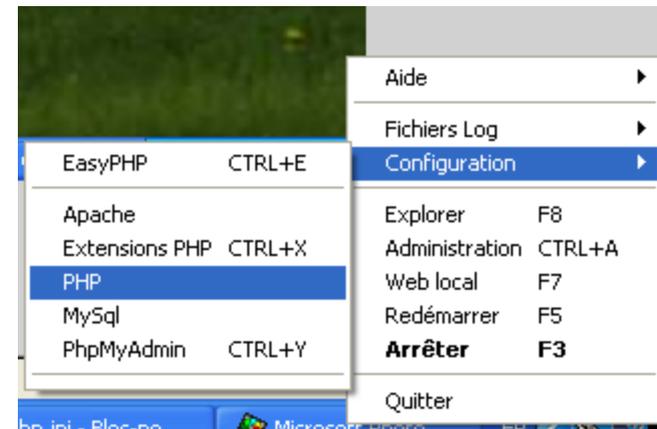
- Ne touchez pas au mot de passe,
- N'essayez pas de le changer
- Le mot de passe à utiliser est "**mysql**"

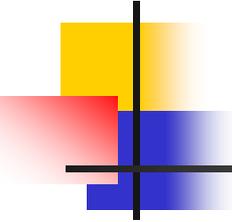
■ Dans php.ini, enlever les ; avant :

extension=php_mysql.dll

php_pdo.dll

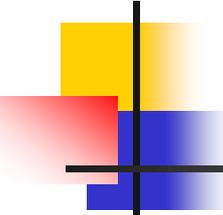
php_pdo_mysql.dll





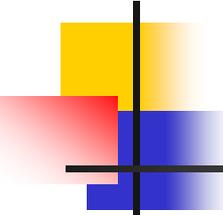
Utilisation avec PHP

- Utilisation de la classe PDO (PHP Data Object)
 - Principale nouveauté de PHP
 - C'est en quelque sorte un langage orienté objet pour traiter les bases de données
 - L'approche PDO permet d'étendre les fonctions d'accès à la base facilement et de manière transparente



Utilisation avec PHP

- **Utiliser votre base de données**
 - L'utilisation de la BD avec PHP se fait en 5 étapes
 - Connexion
 - Sélection de la BD
 - Requête
 - Exploitation des résultats
 - Fermeture de la connexion
 - Structure des classes de PDO
 - PDO propose 3 classes
 - ❖ PDO : lien à la BD
 - ❖ PDOStatement : requêtes et leurs résultats
 - ❖ PDOException : pour la gestion des erreurs



Utilisation avec PHP

■ Connexion au serveur de données

- La première étape consiste à déclarer les variables qui vont permettre la connexion à la base de données (ce sont les paramètres des fonctions de connexion à la base)

- Ces variables sont :

- `$user` : le nom d'utilisateur
- `$passwd` : le mot de passe
- `$host` : l'hôte (ordinateur sur lequel le SGBD est installé)
- `$bdd` : le nom de la base de données

- La deuxième étape est de construire la DSN (Data Source Name)

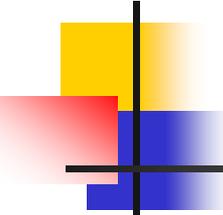
```
$dsn = 'mysql:host=localhost;dbname=ma-base';
```

- Enfin, la connexion se fait par la création d'un objet de la classe PDO

- `$dbh= new PDO($dsn, $user, $pass);`

- Il ne faut pas oublier de fermer la base

- `$dbh=null;`



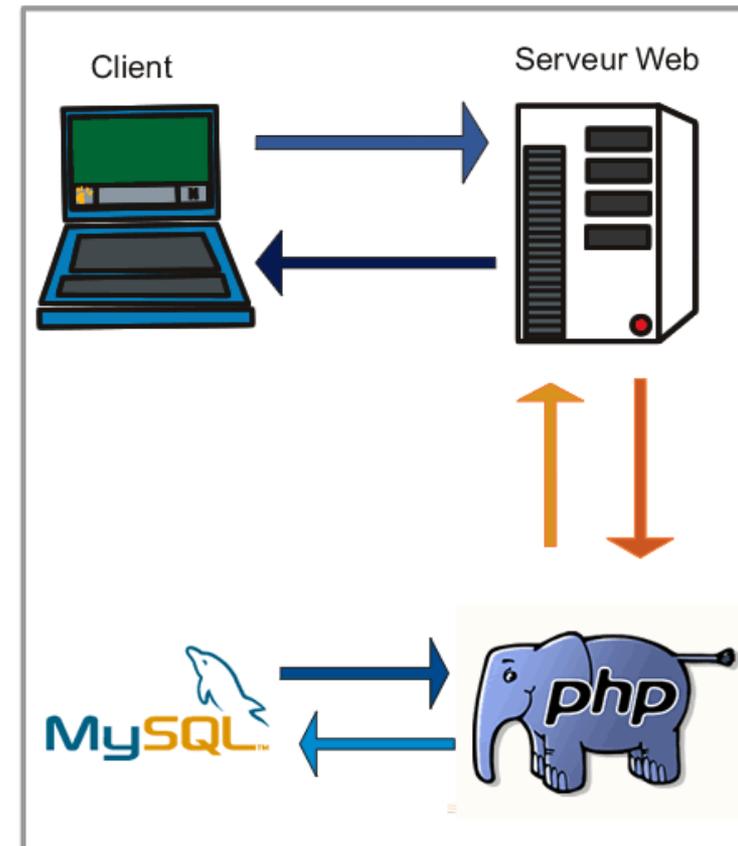
Utilisation avec PHP

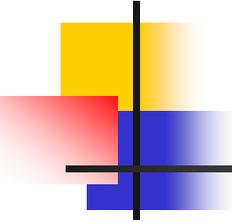
- connexion-PDO.php

```
<?php
$user = 'root';
$pass = 'mysql';
// Data Source Name
$dsn = 'mysql:host=localhost;dbname=ma-base';
try{ //tentative de connexion : on crée un objet de la classe PDO
    $dbh= new PDO($dsn, $user, $pass);
    //S'il y a des erreurs de connexion, un objet PDOException est
    // lancé. Vous pouvez attraper cette exception si vous voulez
    // gérer cette erreur
} catch (PDOException $e){
    print "Erreur ! :". $e->getMessage() . "<br/>";
    die();
}
?>
```

Utilisation avec PHP

- **Effectuer une requête**
 - Une fois la connexion ouverte, on va pouvoir utiliser la BD pour lire, modifier...
 - Pour cela, on utilise le langage **SQL**
- **Pour envoyer une requête au serveur**
 - On peut utiliser deux méthodes de la classe PDO
 - **query()** pour la sélection
 - **exec()** pour la mise à jour

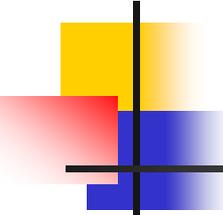




Utilisation avec PHP

■ Requête de sélection

- On utilise la méthode `query()`
- Les données ne sont pas affichées, elles sont mises en mémoire
- Il faut donc aller les chercher et les afficher
- La méthode `fetchAll()` retourne l'ensemble des données sous forme d'un tableau PHP et libère le SGBD
- La méthode `fetch()` permet une lecture séquentielle du résultat
- Le paramètre `fetch_style` détermine la façon dont PDO retourne les résultats (format des résultats)



Utilisation avec PHP

Effectuer une requête

- Exemple 0 :

- Créer une table : `creation-table.php`

```
<?php
```

```
//Inclusion du fichier contenant la connexion à la base  
include_once('connexion-PDO.php');
```

```
//Création de la table personne
```

```
$sql="CREATE TABLE personne (  
id_personne INTEGER PRIMARY KEY ,  
nom VARCHAR( 20 ) NOT NULL ,  
prenom VARCHAR( 20 ) ,  
depart INTEGER( 2 ))";
```

```
$sth = $dbh->query($sql);
```

//création de la table sport

```
$sql="CREATE TABLE sport (  
id_sport INTEGER PRIMARY KEY ,  
design VARCHAR( 30 ) UNIQUE NOT NULL)";
```

```
$sth = $dbh->query($sql);
```

//création de la table pratique

```
$sql="CREATE TABLE pratique (  
id_personne INTEGER NOT NULL ,  
id_sport INTEGER NOT NULL ,  
niveau TINYINT,  
PRIMARY KEY (id_personne,id_sport))";
```

```
$sth = $dbh->query($sql);
```

```
$dbh=NULL;
```

```
?>
```

Utilisation avec PHP

Effectuer une requête

■ Exemple 1 :

- Lire tous les enregistrements : `lire-enregistrements.php`

```
<?php
//Inclusion du fichier contenant la connexion à la base
include_once('connexion-PDO.php');
//La requête SQL
$sql = "SELECT * FROM `infos_tbl` LIMIT 0 , 30";
//Recherche des données
$stmt = $dbh->query($sql);
// On voudrait les résultats sous la forme d'un tableau associatif
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
//Affichage des résultats
foreach ($result as $row){
    echo $row['nom'];echo '-';
    echo $row['prenom'];echo '-';
    echo $row['email'];echo '<br/>';
}
$stmt=NULL;
?>
```

Utilisation avec PHP

Effectuer une requête

■ Base : ma-base

– Table : infos_tbl

			id	nom	prenom	email	icq	titre	url
<input type="checkbox"/>			1	Belaid	Abdel	abelaid@loria.fr	NULL	Professeur	http://www.loria.fr
<input type="checkbox"/>			2	Maquin	Frédéric	Frederic.Maquin@loria.fr	NULL	CDD	http://www.univ-nancy2.fr
<input type="checkbox"/>			3	Moinel	Thomas	thomas.Moinel@univ-nancy2.fr	NULL	CCD	http://www.univ-nancy2.fr
<input type="checkbox"/>			5	Belaid	Yolande	ybelaid@loria.fr	NULL	MCF	http://www.loria.fr
<input type="checkbox"/>			6	Ouwayed	Nazih	ouwayed@loria.fr	NULL	Thésard	http://www.loria.fr

– Résultats

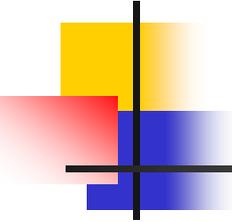
Belaid-Abdel-abelaid@loria.fr

Maquin-Frédéric-Frederic.Maquin@loria.fr

Moinel-Thomas-thomas.Moinel@univ-nancy2.fr

Belaid-Yolande-ybelaid@loria.fr

Ouwayed-Nazih-ouwayed@loria.fr



Utilisation avec PHP

Effectuer une requête

- Exemple 2 : calculer le nombre d'enregistrements : nb-enregistrements.php
 - Deux manières :
 - Créer une requête spécifique en utilisant la fonction `count()` de MySQL
 - Compter le nombre d'éléments contenus dans le tableau renvoyé par la méthode `fetchAll()`

```
<?php
```

```
include_once('connexion-PDO.php');
```

```
//1) En utilisant une requête particulière
```

```
$sql = "SELECT COUNT(*) as nbe FROM `infos_tbl` WHERE  
nom='Belaid'";
```

```
$sth = $dbh->query($sql);
```

```
$result = $sth->fetchAll();
```

```
$nombre = $result[0]['nbe'];
```

```
echo $nombre;
```

```
echo "<br/>";
```

```
//2) En comptant le nombre d'éléments présents dans le tableau  
des résultats
```

```
$sql = "SELECT nom, prenom FROM `infos_tbl` WHERE  
nom='Belaid'";
```

```
$sth = $dbh->query($sql);
```

```
$result = $sth->fetchAll();
```

```
$nombre = count($result);
```

```
echo $nombre;
```

```
$dbh=NULL;
```

```
?>
```

Utilisation avec PHP

Effectuer une requête

■ Requête d'insertion/modification

- On utilise la méthode `exec()` de la classe PDO

- Exemple : `insertion-enreg.php`

```
<?php
```

```
//Inclusion du fichier contenant la connexion à la base
```

```
include_once('connexion-PDO.php');
```

```
//Insertion d'un enregistrement
```

```
$sql = "INSERT INTO infos_tbl  
(id,nom,prenom,email,icq,titre,url) VALUES ('', 'Ouwayed',  
'Nazih', 'ouwayed@loria.fr', '', 'Thésard', 'www.loria.fr')";
```

```
//Exécution de la requête
```

```
$dbh->exec($sql);
```

```
$dbh=NULL;
```

```
?>
```

- `exec()` retourne le nombre de lignes modifiées, 0 si aucune et `FALSE` si erreur de requête

Utilisation avec PHP

Effectuer une requête

■ Gestion des erreurs

- Les erreurs sont stockées et consultables en faisant appel aux méthodes `errorCode()` et `errorInfo()`
- Exemple : `gestion-erreur-mod-sil.php`

```
<?php
//Inclusion du fichier contenant la connexion à la base
include_once('connexion-PDO.php');

//Insertion d'un enregistrement
$sql = "";
if(!$dbh->exec($sql)){
    echo $dbh->errorCode() . "<br>";
    echo $dbh->errorInfo();
    print_r($info);

    //$info[0]==$dbh->errCode() Code d'erreur unifié
    //$info[1] Code d'erreur spécifique au driver
    //$info[2] message d'erreur spécifique au driver
}

$dbh=NULL;
?>
```

Utilisation avec PHP

Effectuer une requête

■ Gestion des erreurs

- Utiliser les exceptions : `gestion-erreur-mod-excep.php`

```
<?php
```

```
//Inclusion du fichier contenant la connexion à la base  
include_once('connexion-PDO.php');
```

```
//On définit le handler d'erreur
```

```
$dbh->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);
```

```
try {
```

```
$sql = "INSERT INTO infos_tbl VALUES ('', 'Ouwayed', 'Nazih',  
'ouwayed@loria.fr', ' ', 'Thésard', 'www.loria.fr')";
```

```
$dbh->exec($sql);
```

```
//si une erreur a eu lieu, une exception est lancée
```

```
}catch (PDOException $e){
```

```
    print "Erreur ! : " . $e->getMessage() ; "<br/>";
```

```
}
```

```
$dbh=NULL;
```

```
?>
```

■ Énoncé

- On veut créer un site HTML qui contrôle la connexion
- Le contrôle se fait par l'affichage d'un formulaire permettant de rentrer :
 - Le nom d'identifiant
 - Un mot de passe
- Le formulaire indique également si :
 - on a oublié son mot de passe (et qu'on demande de le retrouver)
 - ❖ en indiquant son email
 - on veut s'inscrire au site, dans ce cas, on demande d'entrer
 - ❖ son identifiant
 - ❖ son mot de passe
 - ❖ une confirmation du mot de passe
 - ❖ âge, sexe, profession, affiliation, ville, pays
 - où âge, sexe et affiliation ne sont pas obligatoires

■ Énoncé (suite)

- Le contrôle :
 - Un premier contrôle en local permet de valider ou non la saisie des champs (vides, comportant ou non des caractères spécifiques comme le @ pour le email)
 - Un contrôle plus poussé consiste à aller vérifier dans la base de données "abonnés" (MySQL)
- L'inscription au site
 - doit conduire au rangement de l'identifiant et du mot de passe dans la base "abonnés"