

MYSQL

Un SGBD Relationnel

INTRODUCTION

INTRODUCTION – SGBD ?

- **SGBD** = **S**ystème de **G**estion de **B**ase de **D**onnées
- **DBMS** = **D**ata**B**ase-**M**anagement **S**ystem
- **Programme qui gère les données :**
 - Gère le stockage de données (fichiers, intégrité)
 - Gère l'accès aux données (gestion utilisateurs, prise en charge requêtes)
 - Gère la sécurité (paramètres d'accès, filtre IP, etc)
 - Gère les performances (paramétrable)

INTRODUCTION – SGBD(R) ?

- SGBDR, R pour Relationnelles :
 - Liens entre données (Récupération données, suppression données, etc)
 - Evite de dupliquer des données
- Stockage des données:
 - Base de donnée/Schéma:
 - Tables:
 - Colonnes
 - Lignes
 - Champs
- Accès au données : exemple du SQL

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

Exemple de modèle relationnel

Source illustration : <https://fr.wikipedia.org>

INTRODUCTION – LES PRINCIPALES SGBDR UTILISANT LE SQL

SGBDR	Licence	Open Source	Utilisation principale
MySQL	Commerciale ou Libre (<i>GPL</i>)	OUI et non (Oracle)	Application web
SQL Server	Commerciale	non (Microsoft)	Application web
PostgreSQL	Libre	OUI	Application web
MariaDB	Libre (<i>GPLv2</i>)	OUI	Application web
Access	Commerciale	non (Microsoft)	Application bureau
Oracle DB	Commerciale	non (Oracle)	Application entreprise
DB2	Commerciale	non (IBM)	Application entreprise
SQLite	Domaine public	OUI	Application embarquée



INTRODUCTION



- Un système de SGBD de type relationnelle
- Utilise le langage SQL (Structured Query Langage)
- Détenus par Oracle depuis 2009:
 - Licence GPL (non commerciale ou application web)
 - Commerciale
- MariaDB = MySQL open source (créer par un des créateurs originels de MySQL, Michael Widenius, en 2009)

MISE EN PLACE DE L'ENVIRONNEMENT DE TRAVAIL

INSTALLATION DE MYSQL

- Installation d'un serveur MySQL:
 - Possible en standalone
 - Généralement inclus dans les installeurs de serveurs web

- Installation d'un serveur web avec MySQL:
 - Package de type LAMP :
 - Linux -> OS
 - Apache -> serveur HTTP
 - MySQL -> SGBDR
 - PHP -> langage de programmation serveur

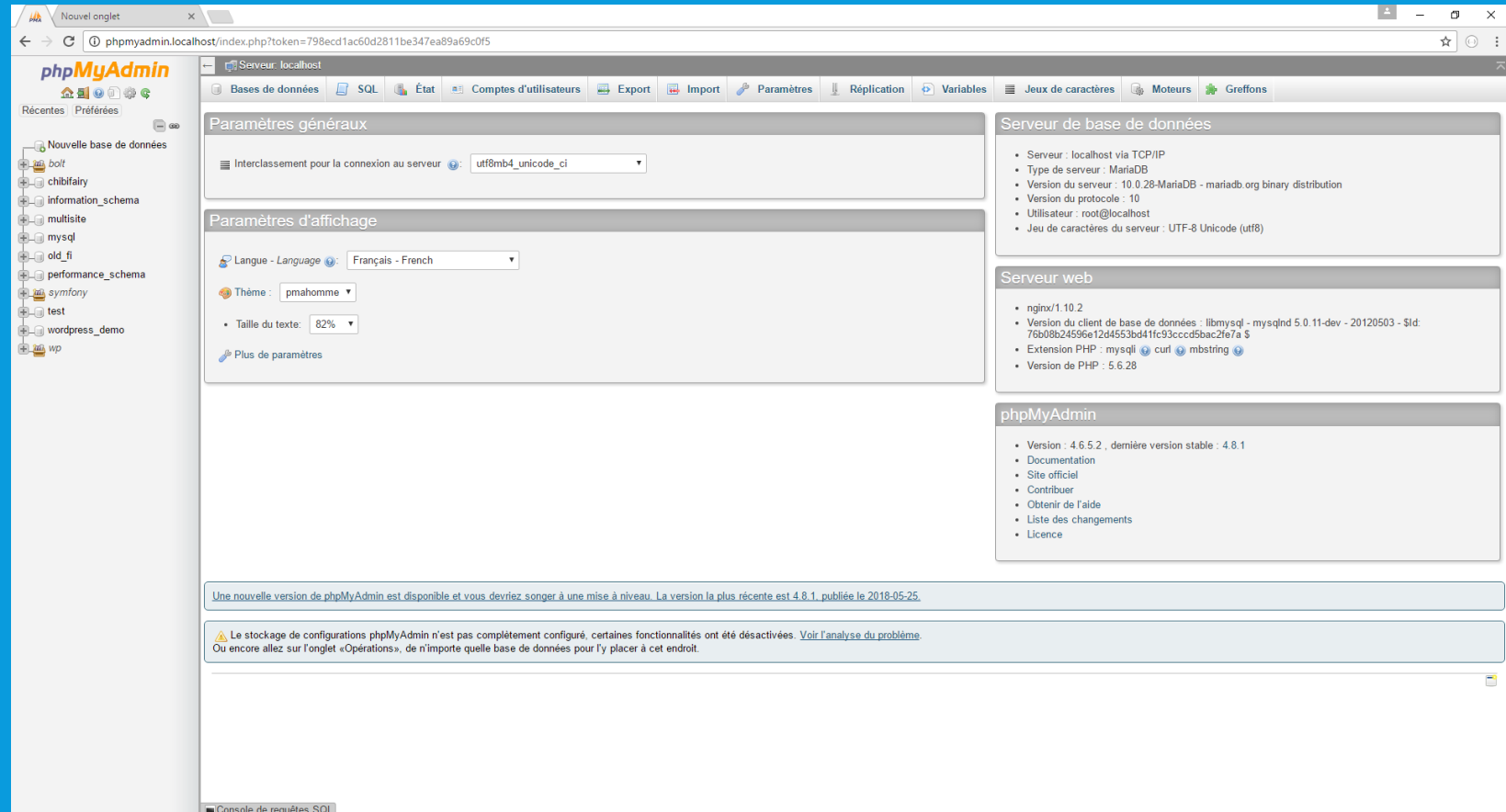
INSTALLATION DE MYSQL

- Installation d'un serveur web avec MySQL:

	Windows	Linux	MacOS	HTTP server	SGBDR
WAMP	OUI	non	non	Apache	MySQL & MariaDB
MAMP	OUI	non	OUI	Apache & Nginx	MySQL
XAMPP	OUI	OUI	OUI	Apache	MariaDB

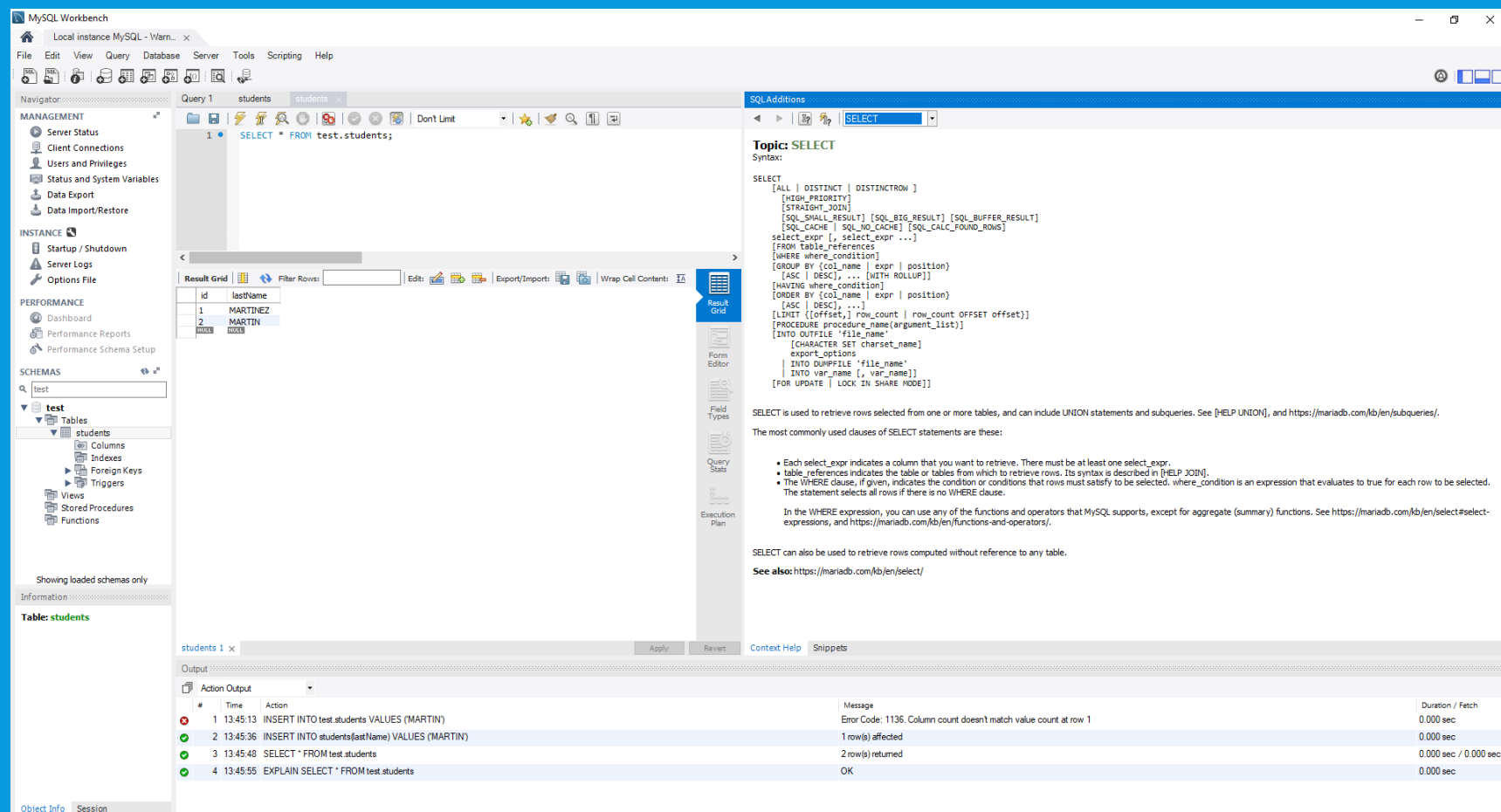
PHPMYADMIN

- Fourni avec les serveur LAMP
- = Interface graphique de gestion de serveur MySQL



INSTALLATION DE MYSQL WORKBENCH

- MySQL Workbench = Interface graphique de gestion de serveur MySQL
- Fonctionne sous Windows, MacOS et Linux
- Permet l'administration de serveur MySQL locaux et distant



GESTION DES DONNÉES - CRUD

Utilisation du langage SQL

READ = Le SELECT

GESTION DES DONNÉES - CRUD

- 4 principales opérations de la gestion de données
 - Create : création de données
 - Read : affichage de données
 - Update : modification de données
 - Delete : suppression de données
- L'opération de base = l'affichage de données
- MySQL développé dans le but d'être performant en terme de lecture de données (READ)
- MySQL utilise le langage de requête SQL :
 - Procédural
 - Syntax = **INSTRUCTION;**

AFFICHAGE DES DONNÉES - SELECT

- En langage SQL :
 - Récupérer des données

```
SELECT expression FROM table;
```

- Expressions:

```
SELECT * FROM cours.students;
```

- Toutes les colonnes

```
SELECT last_name FROM cours.students;  
SELECT last_name, sexe FROM cours.students;
```

- Que la(les) colonne(s) spécifiée(s)

```
SELECT last_name AS "Nom de famille" FROM cours.students;
```

- La colonne spécifiée avec nom de colonne personnalisé

AFFICHAGE DES DONNÉES – SELECT ... FROM

- Expressions:

```
SELECT COUNT(*) FROM cours.students;
```

- En utilisant des fonctions
 - <https://dev.mysql.com/doc/refman/5.7/en/func-op-summary-ref.html>

```
SELECT last_name, IF(height_cm >= 180, 'grand', 'petit') AS taille FROM cours.students;
```

- En utilisant des conditions
 - <https://dev.mysql.com/doc/refman/5.7/en/control-flow-functions.html>

AFFICHAGE DES DONNÉES – SELECT ... FROM ... WHERE

- En langage SQL :
 - Récupérer des données avec des filtres (conditions)

```
SELECT expression FROM table WHERE colonne OPERATOR value;
```

- Exemples:

```
SELECT * FROM cours.students WHERE sexe = "F";
```

```
SELECT ROUND(height_cm/100, 2) AS height_m FROM cours.students HAVING height_m > 1.5;
```


AFFICHAGE DES DONNÉES – SELECT ... FROM ... ORDER BY

- En langage SQL :
 - Grouper les lignes/données en fonctions des valeurs d'une colonne

```
SELECT expression FROM table ORDER BY colonne DIRECTION;
```

- Exemples:

```
SELECT * FROM cours.students ORDER BY last_name ASC;
```

AFFICHAGE DES DONNÉES – SELECT ... FROM ... LIMIT

- En langage SQL :
 - Limiter le nombre de résultats (lignes)
 - Toujours en fin d'instruction

```
SELECT expression FROM table LIMIT number/range;
```

- Exemples:

```
SELECT * FROM cours.students LIMIT 10;
```

AFFICHAGE DES DONNÉES – SELECT ... FROM ... GROUP BY

- En langage SQL :
 - Grouper les lignes/données en fonctions des valeurs d'une colonne

```
SELECT expression FROM table GROUP BY colonne;
```

- Exemples:

```
SELECT * FROM cours.students GROUP BY sexe;
```

```
SELECT COUNT(*), sexe FROM cours.students GROUP BY sexe;
```

AFFICHAGE DES DONNÉES – SELECT ... FROM ... GROUP BY ... HAVING

- En langage SQL :
 - Filtrer avec un Group BY

```
SELECT expression FROM table GROUP BY colonne HAVING condition;
```

- Exemples:

```
SELECT COUNT(*), sexe FROM cours.students GROUP BY sexe HAVING COUNT(*) > 30;
```

GESTION DES DONNÉES - CRUD

Utilisation du langage SQL

CREATE = Le INSERT INTO

UPDATE = Le UPDATE

DELETE = Le DELETE

CRÉER DES DONNÉES – INSERT ... INTO ... VALUE(S)

- En langage SQL :
 - Création d'une ligne dans la table

```
INSERT INTO table (colonne1, colonne2) VALUES (valeur1, valeur2);
```

- Exemple:

```
INSERT INTO cours.students (last_name) VALUES ('MARTINEZ');
```

- Ajouter une ligne

```
INSERT INTO cours.students (last_name)  
VALUES ('BERNARDIN'), ('PONCE');
```

- Ajouter plusieurs lignes

CRÉER DES DONNÉES – INSERT ... INTO ... SELECT

- En langage SQL :
 - Création d'une ligne dans la table avec des données provenant d'un SELECT

```
INSERT INTO table (colonne1, colonne2) SELECT colonne3, colonne4 FROM table2;
```

- Exemple:

```
INSERT INTO cv.contact (last_name, first_name)  
SELECT last_name, first_name FROM cours.students;
```

- Ajouter des données provenant de la table 'students' (base 'cours') dans la table 'contact' (base 'cv')

EDITION DES DONNÉES – UPDATE ... SET ... WHERE

- En langage SQL :
 - Mise à jour d'une ligne

```
UPDATE table SET colonne1 = valeur, colonne2 = valeur WHERE colonne OPERATOR value;
```

- Exemple:

```
UPDATE cours.students SET first_name = 'Charlie' WHERE id = 1;
```


SUPPRESSION DES DONNÉES – DELETE FROM ... WHERE

- En langage SQL :
 - Supprimer des données (lignes)

```
DELETE FROM table WHERE condition;
```

- Exemple:

```
DELETE FROM cours.students WHERE height_cm < 140;
```

- Toutes les lignes avec une taille supérieure à 140 cm

SUPPRESSION DES DONNÉES – TRUNCATE TABLE

- En langage SQL :
 - Supprimer des toutes les données d'une table

```
TRUNCATE TABLE table;
```

- Exemple:

```
TRUNCATE TABLE students;
```

- Vide la table 'students'