Rappels sur les bases de données relationnelles (voir introduction)

Une définition assez intuitive d'une base de données peut être la suivante : une Base de Données (BD) est un ensemble structuré d'informations cohérentes. La cohérence sous-entend ici que les informations ne doivent pas être redondantes ni contradictoires. Un Système de Gestion de Bases de Données (SGBD) est un logiciel complexe permettant de gérer de manière efficace un volume important de données structurées, accessibles par des utilisateurs simultanés ou non, locaux ou non.

Ces définitions étant faites, on voit clairement que MySQL appartient à cette deuxième catégorie.

Il existe les SGBD hiérarchiques, réseau, relationnels : ce sont de loin les plus répandus sur le marché. Citons principalement MySQL, Oracle et SQL Server, objet et NoSQL. Dans ce cours, nous n'aborderons que les bases de données relationnelles avec le SGBD Mysql. Un Système de Gestion de Bases de Données Relationnel ou SGBDR (RDBMS pour *Relational Database Management System*, en anglais) est un SGBD basé sur le modèle relationnel. Il nous faut maintenant définir ce qu'est une relation.

Une relation est un sous-ensemble de données caractérisé par des attributs et visualisable sous forme de table. Une table étant simplement un tableau composé de cellules rangées en lignes et en colonnes. Chaque colonne a pour caractéristiques : un type (numérique, alphanumérique ou date, par exemple), une longueur et possède en général une valeur.

La relation a un nom, est décrite par le nombre et le nom de ses attributs et les domaines de valeurs de chacun d'entre eux. Elle se traduit simplement, dans une deuxième étape (lorsque l'on passe du modèle conceptuel au modèle logique, plus proche de la réalité), sous forme de table, avec des lignes (ou tuples) et des colonnes : objet.

MySQL est extrêmement répandu et populaire sur les serveurs Internet, son succès venant d'une part de sa facilité de mise en œuvre et d'autre part de son caractère originel open source.

MySQL est le système de gestion de bases de données Open Source le plus populaire au monde et est réputé pour sa performance et sa fiabilité. Après une phase de diffusion au début des années 2000 où MySQL se cantonnait principalement à des applications personnelles ou professionnelles bas de gamme, les dernières années ont été marquées par l'adhésion des grands acteurs du Web aux caractéristiques de MySQL. Ainsi aujourd'hui, la très grande majorité des sites à très fort trafic, comme les réseaux sociaux ou de nombreux portails communautaires, ont atteint leur niveau de performance grâce à l'utilisation intensive de MySQL.

Les rachats successifs de MySQL AB, société éditrice de MySQL, par Sun en 2008 puis de Sun par Oracle en 2009 confirment l'intérêt de plus en plus fort de l'industrie pour le produit. La très importante communauté d'utilisateurs est en outre particulièrement très vigilante quant

aux directions qui sont données dans le développement de MySQL et participe activement à l'amélioration du serveur.

Quelques outils de l'administrateur de base de données

PhpMyAdmin

phpMyAdmin est l'incontournable interface graphique de MySQL. Comme son nom le suggère, c'est un outil d'administration de MySQL écrit en PHP. C'est une interface de type Web, écrite en PHP donc, qui est accessible par un simple navigateur Internet (FireFox ou Chrome par exemple).

Les prérequis logiciels minimaux sont les suivants :

- PHP 5.2.0 ou ultérieur.
- Un serveur web (Apache en général), dans lequel est inclus le module PHP.
- MySQL 5.0 ou ultérieur.
- Un navigateur quelconque (avec les cookies activés).

L'interface d'administration phpMyAdmin présente un certain nombre de fonctionnalités. Il permet de :

- Naviguer et gérer des bases de données, tables, etc.
- De créer et lire des fichiers d'export (dumps) de tables dans divers formats.
- D'importer des données et des structures MySQL à partir de fichiers de différents formats.
- D'administrer des serveurs.
- De gérer les utilisateurs MySQL et les privilèges.

MySQL Workbench

MySQL Workbench est un outil graphique intégré, dédié au développeur et à l'administrateur.

Il permet de faire :

- La conception et de la modélisation de base de données.
- L'administration complète de base de données.
- Gestion des utilisateurs MySQL et des privilèges.

Dans tout le reste du cours, nous allons utiliser mysl en ligne de commande avec un environnement Linux.

Installation et prise en main

Installation d'un serveur MySQL sous Linux (Ubuntu) et de quelques outils d'administration

```
root@sippc:~# apt-get install apache2
```

```
root@sippc:~# apt-get install mysql-server
```

```
root@sippc:~# apt-get install phpmyadmin
```

```
root@sippc:~# apt-get install mysql-workbench
```

La documentation en ligne officielle est disponible sur : http://dev.mysql.com/doc/

Connexion au SGBD sans préciser la base.

```
root@sippc:~# mysql -u root -p
```

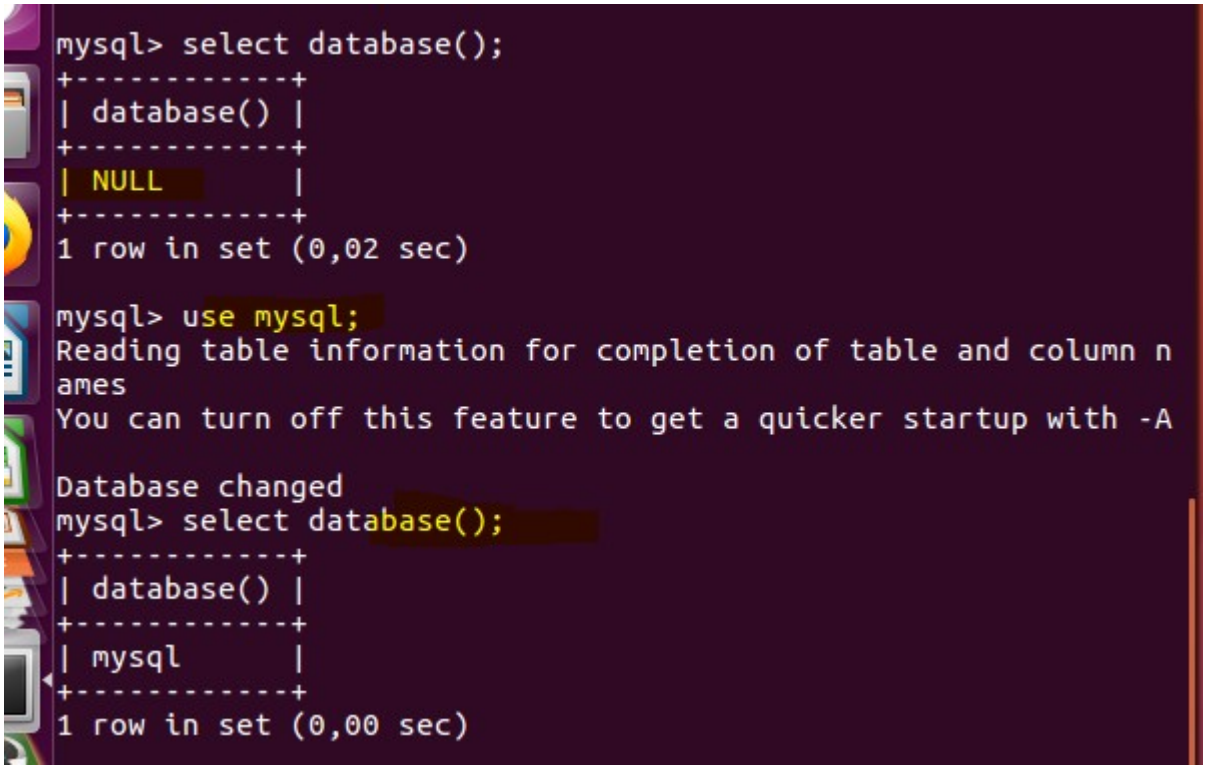
Connexion au SGBD en précisant la base :

```
root@sippc:~# mysql -u root -p -D kamailio
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.26-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select database();
+------------+
| database() |
+------------+
| kamailio   |
+------------+
1 row in set (0,00 sec)
```

Comment afficher les bases de données disponibles sur le serveur ?

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bobo               |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
+--------------------+
6 rows in set (0,00 sec)
```

Comment savoir la base de données à laquelle vous êtes connecté ?



Comment sélectionner une base de données pour l'utiliser ?

use nomdelabase ;

Activité : Chaque étudiant doit :

- **Installer son SGBD ;**
- **S'y connecter ;**
- **Afficher l'ensemble des bases de données ;**
- **Sélectionner une base de données ;**
- **Saisir la commande select database() et voir le résultat selon qu'une base de données est sélectionne ou pas.**

Les types de données

Dans SQL, il existe trois grandes familles de données : numérique, caractère (ou alphanumérique) et temporelle (dates et heures). Cependant, chaque SGBDR a décliné des types spécifiques pour un besoin précis comme les types géographiques, ou pour des problématiques de stockage. Le type choisi dépendra donc de la précision recherchée, tout en tenant compte de la taille nécessaire pour stocker la donnée.

1. Numériques

Les types numériques permettent de définir si l'on souhaite un entier, un décimal ou un nombre à virgule flottante.

Nombres entiers			
Type	**Précision**	**Stockage (octets)**	**BDD**
TINYINT	0 à 255 ou 128 à 127	1	MySQL
SMALLINT	32 768 à 32 768 ou 0 à 65 535 pour MySQL	2	MySQL
MEDIUMINT	8 388 608 à 8 388 607 ou 0 à 16 777 215	3	MySQL
INT(p)	2 147 483 648 à 2 147 483 647 ou 0 à 4 294 967 295 où p désigne le nombre de chiffres maximum	4	MySQL, Oracle
SERIAL	1 à 2 147 483 647. Entier à incrémentation automatique ou 1 à 9 223 372 036 854 775 807 pour MySQL	4	PostgreSQL, MySQL
BIGINT	9 223 372 036 854 775 808 à 9 223 372 036 854 775 807 ou 0 à 18 446 744 073 709 551 615 pour	8	SQL Server, PostgreSQL, MySQL

	MySQL		

Nombres décimaux et flottants			
Type	**Précision**	**Stockage (octets)**	**BDD**
DECIMAL(p[,s]) ou NUMERIC(p[,s])	10ˆ38+1 à 10ˆ381. p représente la précision, c'est à dire le nombre total et maximum de chiffres à gauche et à droite de la virgule. La précision par défaut est 18. s représente l'échelle, c'est à dire le nombre de chiffres maximum après la virgule. L'échelle est optionnelle. La valeur par défaut est 0. Il peut être utilisé pour les devises.	5 à 17 selon la précision et l'échelle	SQL Server, PostgreSQL, MySQL
FIXED (p[,s])	Idem DECIMAL	Idem DECIMAL	MySQL
FLOAT(n)	1,79E+308 à 2,23E308, 0 et 2,23E308 à 1,79E+308. n représente le nombre de bits utilisés pour stocker la donnée. Sa valeur par défaut est 53. Il peut être utilisé pour les calculs scientifiques.	4 pour n entre 1 et 24 ou 8 pour n entre 25 et 53. 22 maxi pour Oracle	SQL Server, Oracle, MySQL
DOUBLE(n) ou DOUBLE PRECISION (n)	n est le nombre maximum de décimaux utilisés de 1 à 15.	8	PostgreSQL MySQL
REAL(n)	3,40E+38 à 1,18E38, 0 et 1,18E-	4	SQL Server,

	38 à 3,40E+38. n représente le nombre de bits utilisés pour stocker la donnée. Sa valeur par défaut est 24.		MySQL, PostgreSQL

Selon le type de base de données, il existe des propriétés qui complètent le type de données comme :

- AUTO_INCREMENT dans MySQL pour incrémenter automatiquement un entier.
- UNSIGNED dans MySQL pour préciser si l'on accepte ou non les nombres négatifs.
- ZEROFILL dans MySQL permet de remplacer les espaces par des 0. L'attribut UNSIGNED est automatiquement ajouté avec ZEROFILL.

2. Caractères

Le type alphanumérique classique se note CHAR pour les données de taille fixe et VARCHAR pour les chaînes de caractères de longueur variable.

La majorité du temps, il est conseillé d'utiliser VARCHAR. En effet, une donnée déclarée sur 50 par exemple et qui ne contient que deux caractères sera effectivement stockée sur deux caractères avec VARCHAR alors qu'elle sera stockée sur 50 caractères avec CHAR. CHAR complète avec des espaces jusqu'à la longueur maximum. Il est possible de trouver le type TEXT, mais ce type est amené à disparaître. Ce n'est pas un standard SQL.

Type	Précision	Stockage (octets)	BDD
CHAR(n)	Jusqu'à 255 caractères pour MySQL.	n octets (dépend de la sémantique dans Oracle et limité à 2 000 octets).	SQL Server, MySQL, PostgreSQL, Oracle
VARCHAR(n)	0 à 65 535 caractères pour MySQL.	Longueur chaîne + 1 octet	SQL Server, MySQL PostgreSQ

			L, Oracle

Selon le type de base de données, il existe des propriétés qui complètent le type de données. Par exemple, dans MySQL, sur le type CHAR, il est possible d'ajouter des options :

- BINARY est utilisé pour tenir compte de la casse.
- NATIONAL spécifie que le jeu de caractères par défaut de MySQL doit être utilisé.
- ASCII spécifie le jeu de caractères latin1 à utiliser.
- UNICODE spécifie le jeu de caractères à utiliser.

3. Dates et heures

Les types de format temporels sont principalement DATE, TIME et TIMESTAMP.

Les formats par défaut des dates ou des heures varient d'une base de données à l'autre. Pour connaître le format par défaut il faut utiliser SELECT NOW(); qui donnent la date du jour au format utilisé par la base.:

- DATE : format date, norme ISO AAAAMMJJ
- TIME : format heure, norme ISO HH:MM:ss.nnnnnnn

Type	Précision	Stockage (octets)	BDD
DATETIME	Nanoseconde (du 01/01/1000 au 31/12/9999)	8 octets	MySQL
DATE	Jour (du 01/01/0001 au 31/12/9999)	3 octects	SQL Server, Oracle, MySQL
TIME	Nanoseconde	3 à 5 octects	SQL Server, MySQL
TIMESTAMP	Date et heure courantes	8 octets	MySQL

- TIMESTAMP : format date et heure, norme ISO AAAAMMJJ HH:MM:ss .nnn

Récapitulatif :

*INT pour le stockage de nombres entiers

Type	Octets	De...	à...
TINYINT	1	-128	127
SMALLINT	2	-32 768	32 767
MEDIUMINT	3	-8 388 608	8 388 607
INT	4	-2 147 483 648	2 147 483 647
BIGINT	8	-9 223 372 036 854 775 808	9 223 372 036 854 775 807

Remarques Si le mot-clé UNSIGNED est spécifié, la limite inférieure est 0 (seuls des nombres positifs sont stockés), et la limite supérieure est doublée car on s'affranchit du signe (ex. TINYINT UNSIGNED : 0 à 255). INTEGER est un alias pour INT. On peut spécifier la longueur maximale du champ, entre parenthèses, après le type.

L'attribut ZEROFILL définit le caractère « 0 » comme remplissage par défaut. **Exemple** en stockant la valeur 3 dans un champ `INT(4) ZEROFILL`, la valeur lue sera 0003.

BIT, BOOL, BOOLEAN : types booléens pour valeurs binaires

Expriment une variable ne prenant que deux valeurs (vrai ou faux). Équivalent à un `TINYINT(1)` stockant `1` ou `0` dans un champ numérique.

FLOAT, DOUBLE, DECIMAL pour les nombres à virgule flottante

`FLOAT` et `DOUBLE` (équiv. `REAL`) stockent des nombres à virgule flottante. `DECIMAL` (équiv. `NUMERIC`) stocke sous forme littérale : chaque caractère représente un chiffre stocké.

Remarque Les attributs UNSIGNED et ZEROFILL sont là aussi applicables.

BINARY et VARBINARY pour les chaînes binaires

Diffèrent de la même façon que CHAR et VARCHAR (voir ci-avant). Ils n'ont pas de jeu de caractères associé, les tris et comparaisons sont basés sur la valeur numérique de l'octet, et on spécifie leur taille maximale de la même façon que CHAR et VARCHAR, à la différence près qu'elle est définie en octets.

TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, BLOB, TINYBLOB, MEDIUMBLOB, LONGBLOB pour les chaînes longues

Une colonne de type TEXT est assimilable à un VARCHAR de grande capacité. Ce type peut contenir une quantité variable de données, en fonction de l'un des 4 types. Les types BLOB (Binary Long Object) diffèrent des types TEXT : ils sont sensibles à la casse (majuscules/minuscules) et triés sur leur valeur binaire.

REMARQUE On ne peut spécifier de valeur par défaut pour les types BLOB et TEXT.

Type	Capacité de stockage en caractères
TINYTEXT / TINYBLOB	255
TEXT / BLOB	65 535
MEDIUMTEXT / MEDIUMBLOB	16 777 215
LONGTEXT / LONGBLOB	4 294 967 295 (4 Go)

EXEMPLES

```
prix DECIMAL(6,2), poids FLOAT UNSIGNED, nom CHAR(30), prenom
CHAR(50),
article TEXT NOT NULL,
```

ENUM et SET pour proposer des ensembles de choix

ENUM permet le choix d'une valeur parmi une liste définie à la création de la table. Il est possible d'utiliser une chaîne vide (''). La valeur par défaut sera le premier élément de l'énumération si la colonne est déclarée NOT NULL, ou NULL si la colonne dispose de cet attribut. Une énumération peut avoir un maximum de 65 535 éléments.
SET diffère d'ENUM car il peut avoir 0 ou plusieurs valeurs choisies dans la liste définie à la création de la table. Les valeurs sont séparées par des virgules.

REMARQUE Pour sélectionner des champs selon des valeurs de SET, le plus simple est d'utiliser la fonction FIND_IN_SET qui retourne le nombre d'occurrences trouvées.

EXEMPLE

```
couleur ENUM('bleu','vert','indigo'),
colis SET('lourd','fragile','rond'),
SELECT * FROM nom_table WHERE
FIND_IN_SET('fragile',colis)>0;
```

Gestion des dates avec TIME, DATETIME, DATE, TIMESTAMP, YEAR

Type	Format	De	A
YEAR	AAAA	1901	2155
DATE	AAAA-MM-JJ	1000-01-01	9999-12-31
DATETIME	AAAA-MM-JJ	1000-01-01	9999-12-31
	HH:MM:SS	00:00:00	23:59:59
TIME	HH:MM:SS	-838:59:59	838:59:59
TIMESTAMP	AAAA-MM-JJ	1000-01-01	9999-12-31
	HH:MM:SS	00:00:00	23:59:59

Le type TIMESTAMP diffère des autres types date, car il stocke automatiquement l'heure du serveur lors d'une commande INSERT ou UPDATE sur les enregistrements affectés. En d'autres termes, MySQL applique automatiquement la fonction NOW() lorsqu'aucune valeur par défaut n'est spécifiée lors de l'insertion ou de la modification.

La création de base de données

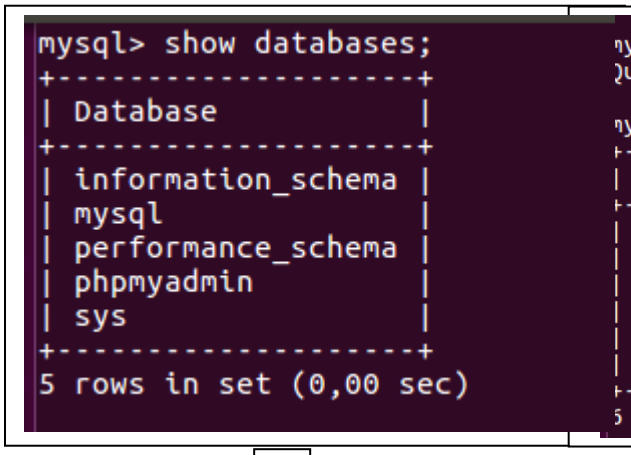
CREATE DATABASE licence2mic ;
Ou
CREATE DATABASE IF NOT EXISTS licence2mic ;

Activité :

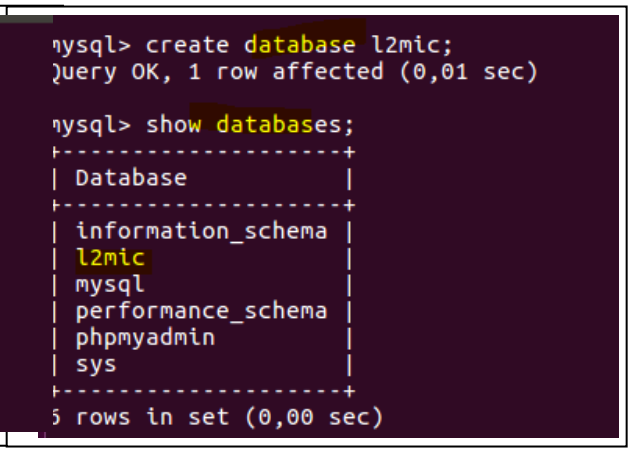
1. Créer la base de données licence2mic sans « if not exists »
2. Créer à nouveau licence2mic sans if not exists
    a. Quel message d'erreur avez-vous ?
    b. Utiliser la commande show warnings ;
3. Créer à nouveau licence2mic avec if not exists ;
    a. Quel message avez-vous ?
    b. Utiliser la commande show warnings et comparer avec le résultat de la question 2. Que constatez-vous ?

La suppression de base

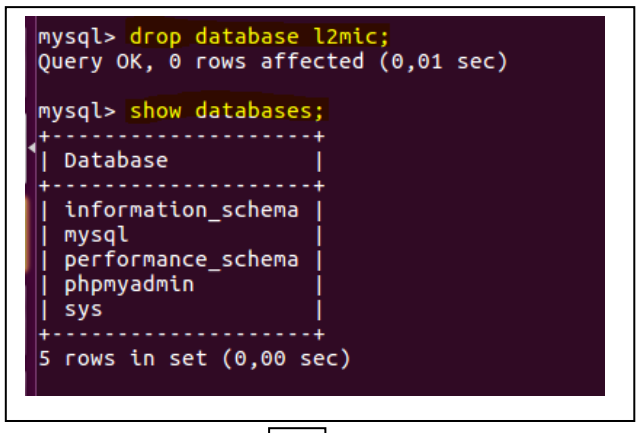
DROP DATABASE licence2mic. La suppression d'une base de données entraine la suppression de toutes ses tables y compris les données de cette base.



1



2



3

Activité :

1. Afficher l'ensemble des bases de données disponibles ;
2. Créer une base de données de votre choix ;
3. Utiliser la commande show databases pour vous assurer que la base est présente ;
4. Supprimer la base ;
5. Confirmer la suppression en utilisant à nouveau la commande show databases ;

La création de tables

Création classique

Une table se définit principalement par les colonnes qui la composent et les règles qui s'appliquent à ces colonnes.

Syntaxe : CREATE TABLE nom_de_table (Nom_colonne1 Type_colonne1 optionfacultative1 optionfacultative2, Nom_colonne2 Type_colonne2, Nom_colonne3 Type_colonne3,...);

```
mysql> create database ecole;
Query OK, 1 row affected (0,01 sec)

mysql> use ecole;
Database changed
mysql> create table etudiants(idetu int auto_increment primary key, nom varchar(50),prenom varchar(50), email varchar(50), datenaissance datetime not null);
Query OK, 0 rows affected (0,06 sec)

mysql> desc etudiants;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| idetu         | int(11)     | NO   | PRI | NULL    | auto_increment |
| nom           | varchar(50) | YES  |     | NULL    |                |
| prenom        | varchar(50) | YES  |     | NULL    |                |
| email         | varchar(50) | YES  |     | NULL    |                |
| datenaissance | datetime    | NO   |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
5 rows in set (0,02 sec)
```

NB : Les options facultatives peuvent prendre entre autres les valeurs :

NOT NULL

DEFAULT 'valeur_par_defaut'

PRIMARY KEY

Activité à faire par l'étudiant :

- Créer une base de données quelconque ;
- Sélectionner la base de données ;
- Créer une table avec 6 colonnes ou champs en précisant la clé primaire ;
- Décrire la structure de la table.

Exercice à faire :

Créer une table à partir des données d'une autre table existante.

Créer une table ayant les mêmes caractéristiques qu'une autre existante (sans les données).

Insérer des données dans une table

Syntaxe : insert into nom_table (champ1, champ2,….) values (valeur1,valeur2,…) ;

Exemple :

```
mysql> desc etudiants;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| idetu         | int(11)     | NO   | PRI | NULL    | auto_increment |
| nom           | varchar(50) | YES  |     | NULL    |                |
| prenom        | varchar(50) | YES  |     | NULL    |                |
| email         | varchar(50) | YES  |     | NULL    |                |
| datenaissance | date        | NO   |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
5 rows in set (0,00 sec)

mysql> insert into etudiants(nom,prenom,email,datenaissance) values ("MENDY","Pierre","mendy@gmail.com","1998-06-12");
Query OK, 1 row affected (0,01 sec)
```

Afficher les données :

Syntaxe :

Pour afficher les enregistrements d'un ou plusieurs champs d'une table, on peut utiliser la syntaxe ci-dessous :

select nom_champ1,nom_champ2 from nom_table ;

Pour afficher l'ensemble des données de la table, utiliser la syntaxe suivante :

Select * from nom_table ;

Exemple :

```
mysql> select *from etudiants;
+-------+--------+--------+-------------------+---------------+
| idetu | nom    | prenom | email             | datenaissance |
+-------+--------+--------+-------------------+---------------+
|     1 | Bessan | DEGBOE | bessan@degboe.org | 1800-06-14    |
|     2 | Alaba  | DIOP   | alaba@bobo.com    | 1800-06-14    |
|     3 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     4 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     5 | MENDY  | Pierre | mendy@gmail.com   | 1998-06-12    |
+-------+--------+--------+-------------------+---------------+
5 rows in set (0,00 sec)

mysql> select nom,prenom from etudiants;
+--------+--------+
| nom    | prenom |
+--------+--------+
| Bessan | DEGBOE |
| Alaba  | DIOP   |
| Toto   | DIOP   |
| Toto   | DIOP   |
| MENDY  | Pierre |
+--------+--------+
```

```
mysql> select * from etudiants;
+-------+--------+--------+-------------------+---------------+
| idetu | nom    | prenom | email             | datenaissance |
+-------+--------+--------+-------------------+---------------+
|     1 | Bessan | DEGBOE | bessan@degboe.org | 1800-06-14    |
|     2 | Alaba  | DIOP   | alaba@bobo.com    | 1800-06-14    |
|     3 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     4 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     5 | MENDY  | Pierre | mendy@gmail.com   | 1998-06-12    |
+-------+--------+--------+-------------------+---------------+
5 rows in set (0,00 sec)
```

Quelques précisions sur la clause Where :

```
mysql> select * from etudiants;
+-------+--------+--------+-------------------+---------------+
| idetu | nom    | prenom | email             | datenaissance |
+-------+--------+--------+-------------------+---------------+
|     1 | Bessan | DEGBOE | bessan@degboe.org | 1800-06-14    |
|     2 | Alaba  | DIOP   | alaba@bobo.com    | 1800-06-14    |
|     3 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     4 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     5 | MENDY  | Pierre | mendy@gmail.com   | 1998-06-12    |
+-------+--------+--------+-------------------+---------------+
5 rows in set (0,00 sec)

mysql> select nom,prenom from etudiants where idetu in(2,3,5);
+-------+--------+
| nom   | prenom |
+-------+--------+
| Alaba | DIOP   |
| Toto  | DIOP   |
| MENDY | Pierre |
+-------+--------+
3 rows in set (0,00 sec)

mysql> select nom,prenom from etudiants where idetu >(2);
+-------+--------+
| nom   | prenom |
+-------+--------+
| Toto  | DIOP   |
| Toto  | DIOP   |
| MENDY | Pierre |
+-------+--------+
3 rows in set (0,00 sec)

mysql> select nom,prenom from etudiants where idetu between 1 and 4;
+--------+--------+
| nom    | prenom |
+--------+--------+
| Bessan | DEGBOE |
| Alaba  | DIOP   |
| Toto   | DIOP   |
| Toto   | DIOP   |
+--------+--------+
4 rows in set (0,00 sec)
```

Modifier une table existante :

Vous pouvez utiliser l’instruction alter table pour ajouter, modifier ou supprimer ou renommer des champs dans une table.

Syntaxe :

alter table nom_table drop column nom_colonne ;

alter table nom_table add nom_nouveau_champ type_nouveau_champ(longueur) ;

```
alter table nom_table modify nom_champ nouveau_type


alter table nom_table change nom_ancien_champ nom_nouveau_champ type_
```

Exemple :

1. Ajouter une nouvelle colonne :

```
mysql> select *from etudiants;
+-------+--------+--------+-------------------+---------------+
| idetu | nom    | prenom | email             | datenaissance |
+-------+--------+--------+-------------------+---------------+
|     1 | Bessan | DEGBOE | bessan@degboe.org | 1800-06-14    |
|     2 | Alaba  | DIOP   | alaba@bobo.com    | 1800-06-14    |
|     3 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     4 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |
|     5 | MENDY  | Pierre | mendy@gmail.com   | 1998-06-12    |
+-------+--------+--------+-------------------+---------------+
5 rows in set (0,00 sec)

mysql> alter table etudiants add telephone int;
Query OK, 0 rows affected (0,12 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select *from etudiants;
+-------+--------+--------+-------------------+---------------+-----------+
| idetu | nom    | prenom | email             | datenaissance | telephone |
+-------+--------+--------+-------------------+---------------+-----------+
|     1 | Bessan | DEGBOE | bessan@degboe.org | 1800-06-14    |      NULL |
|     2 | Alaba  | DIOP   | alaba@bobo.com    | 1800-06-14    |      NULL |
|     3 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |      NULL |
|     4 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |      NULL |
|     5 | MENDY  | Pierre | mendy@gmail.com   | 1998-06-12    |      NULL |
+-------+--------+--------+-------------------+---------------+-----------+
5 rows in set (0,00 sec)
```

2. **Supprimer une colonne :**

```
mysql> select *from etudiants;
+-------+--------+--------+-------------------+---------------+-----------+
| idetu | nom    | prenom | email             | datenaissance | telephone |
+-------+--------+--------+-------------------+---------------+-----------+
|     1 | Bessan | DEGBOE | bessan@degboe.org | 1800-06-14    |      NULL |
|     2 | Alaba  | DIOP   | alaba@bobo.com    | 1800-06-14    |      NULL |
|     3 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |      NULL |
|     4 | Toto   | DIOP   | bessan@dedeg      | 1800-06-14    |      NULL |
|     5 | MENDY  | Pierre | mendy@gmail.com   | 1998-06-12    |      NULL |
+-------+--------+--------+-------------------+---------------+-----------+
5 rows in set (0,00 sec)

mysql> alter table etudiants drop column email;
Query OK, 0 rows affected (0,10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select *from etudiants;
+-------+--------+--------+---------------+-----------+
| idetu | nom    | prenom | datenaissance | telephone |
+-------+--------+--------+---------------+-----------+
|     1 | Bessan | DEGBOE | 1800-06-14    |      NULL |
|     2 | Alaba  | DIOP   | 1800-06-14    |      NULL |
|     3 | Toto   | DIOP   | 1800-06-14    |      NULL |
|     4 | Toto   | DIOP   | 1800-06-14    |      NULL |
|     5 | MENDY  | Pierre | 1998-06-12    |      NULL |
+-------+--------+--------+---------------+-----------+
```

3. Modifier le type d’un champ existant :

```
mysql> desc personne;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| idpersonne | int(11)     | NO   | PRI | NULL    | auto_increment |
| nom        | varchar(30) | YES  |     | NULL    |                |
| prenom     | varchar(30) | YES  |     | NULL    |                |
| fonction   | char(50)    | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0,00 sec)

mysql> alter table personne modify fonction int;
Query OK, 0 rows affected (0,11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc personne;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| idpersonne | int(11)     | NO   | PRI | NULL    | auto_increment |
| nom        | varchar(30) | YES  |     | NULL    |                |
| prenom     | varchar(30) | YES  |     | NULL    |                |
| fonction   | int(11)     | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0,00 sec)
```

4. Renommer un champ

```
mysql> desc personne;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| idpersonne | int(11)     | NO   | PRI | NULL    | auto_increment |
| nom        | varchar(30) | YES  |     | NULL    |                |
| prenom     | varchar(30) | YES  |     | NULL    |                |
| fonction   | int(11)     | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0,00 sec)

mysql> alter table personne change fonction role varchar(30);
Query OK, 0 rows affected (0,09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc personne;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| idpersonne | int(11)     | NO   | PRI | NULL    | auto_increment |
| nom        | varchar(30) | YES  |     | NULL    |                |
| prenom     | varchar(30) | YES  |     | NULL    |                |
| role       | varchar(30) | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0,01 sec)
```

Supprimer un enregistrement :

Syntaxe :

Pour supprimer toutes les données d’une table, on peut utiliser l’une des deux commandes ci-dessous :

```
delete from nom_table ;
```

```
delete * from nom_table ;
```

Pour supprimer de manière précise un enregistrement, il faut utiliser la commande ci-dessous :

```
delete from nom_table where nom_champ=valeur ;
```

```
mysql> select * from professeur;
+--------+-------+--------+
| idprof | nom   | prenom |
+--------+-------+--------+
|      1 | MENDY | Pierre |
|      2 | MENDY | Pierre |
|    110 | MENDY | Pierre |
|    111 | MENDY | Pierre |
+--------+-------+--------+
4 rows in set (0,00 sec)

mysql> delete from professeur where idprof=110;
Query OK, 1 row affected (0,01 sec)
```

Modifier un enregistrement existant :

update nom_table set champ1=nouvelle_valeur1,champ2=nouvelle_valeur2 where champx=

Exemple :
Avant la modification :

```
mysql> select *from etudiants;
+-------+--------+--------+--------------------+---------------+
| idetu | nom    | prenom | email              | datenaissance |
+-------+--------+--------+--------------------+---------------+
|     1 | Bessan | DEGBOE | bessan@degboe.org  | 1800-06-14    |
|     2 | Toto   | DIOP   | bessan@dedeg       | 1800-06-14    |
|     3 | Toto   | DIOP   | bessan@dedeg       | 1800-06-14    |
|     4 | Toto   | DIOP   | bessan@dedeg       | 1800-06-14    |
|     5 | MENDY  | Pierre | mendy@gmail.com    | 1998-06-12    |
+-------+--------+--------+--------------------+---------------+
5 rows in set (0,01 sec)
```

Après la modification :

```
mysql> update etudiants set nom='Alaba',email='alaba@bobo.com' where idetu=2;
Query OK, 1 row affected (0,01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select *from etudiants;
+-------+--------+--------+--------------------+---------------+
| idetu | nom    | prenom | email              | datenaissance |
+-------+--------+--------+--------------------+---------------+
|     1 | Bessan | DEGBOE | bessan@degboe.org  | 1800-06-14    |
|     2 | Alaba  | DIOP   | alaba@bobo.com     | 1800-06-14    |
|     3 | Toto   | DIOP   | bessan@dedeg       | 1800-06-14    |
|     4 | Toto   | DIOP   | bessan@dedeg       | 1800-06-14    |
|     5 | MENDY  | Pierre | mendy@gmail.com    | 1998-06-12    |
+-------+--------+--------+--------------------+---------------+
5 rows in set (0,00 sec)
```

Constater l’effet de auto_increment. Modifier la table professeur pour que l’incrémentation commence à partir de 500.

alter table professeur auto_increment=500;

Créer un nouvel enregistrement et vérifier le numéro automatiquement attribué à la colonne id.

Comment rendre accessible une base à distance ?

1. Faire écouter sur n’importe quelle adresse et non 127…..

root@m2-pc:/etc/mysql# grep -r bind-address

2. Modifier bind-address et mettre à 0.0.0.0
3. Redémarrer le serveur mysql pour prendre en compte les modifications
4. Créer un utilisateur et lui donner tous les droits sur la base de données pour qu’il puisse y accéder à distance.

grant all on etablissement.* to 'distan'@'ip' identified by 'passer';
mysql -u distan -p -h 10.0.0.44

<u>Sauvegarde et restauration de base de données :</u>

Mysqldump est l’outil par excellence pour la sauvegarde de base de données. Avec cet outil, vous pouvez entre autres sauvegarder :

- Une ou plusieurs bases de données précises dans une base de données ;
- Une ou plusieurs tables à l’intérieur d’une base de données ;
- Sauvegarder toutes les bases de données.

Syntaxe :

#mysqldump -u nom_utilisateur -p --all-databases > masauvegarde.sql

mysqldump -u nom_utilisateur -p --databases base1 base2 base3 > sauvegarde3bases.sql

#mysqldump -u nom_utilisateur -p --databases nom_base --tables nom_table1 nom_table2 nom_table3 > sauvegarde_3_tables

Restauration de bases de données :

Syntaxe : mysql -u nom_utilisateur -p <nomsauvegarde.sql

Avant de restaurer les bases de données, nous allons d’abord supprimer toutes les bases de données.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| kamailio           |
| maison             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0,00 sec)

mysql> drop database kamailio;
Query OK, 62 rows affected (1,15 sec)

mysql> drop database maison;
Query OK, 2 rows affected (0,05 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0,00 sec)

mysql>
```

Sauvegarde de toutes les bases :

Avant de démarrer la sauvegarde, il faut savoir quelle est le nom de la base qu'on souhaite sauvegarder.

```
root@toip-pc:~# mysql -u root -p <masauvegarde.sql
Enter password:
root@toip-pc:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.7.26-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| kamailio           |
| l2mic              |
| maison             |
| mysql              |
| ouleye             |
| performance_schema |
| sys                |
+--------------------+
8 rows in set (0,00 sec)
```

```
mysql>
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| kamailio           |
| l2mic              |
| maison             |
| mysql              |
| ouleye             |
| performance_schema |
| sys                |
+--------------------+
8 rows in set (0,00 sec)

mysql> use maison;
Database changed
mysql> show tables;
+------------------+
| Tables_in_maison |
+------------------+
| habitants        |
| piece            |
+------------------+
2 rows in set (0,00 sec)

mysql> select *from piece;
+---------+---------+----------------+
| idpiece | surface | typepiece      |
+---------+---------+----------------+
|       1 |      20 | chambre enfant |
|       2 |      30 | douche         |
+---------+---------+----------------+
2 rows in set (0,00 sec)
```

```
root@toip-pc:~# mysqldump -u root -p --all-databases >masauvegarde.sql
Enter password:
root@toip-pc:~# ls
Bureau     examples.desktop  Images           Modèles  ok      Téléchargements
Documents  GNS3              masauvegarde.sql  Musique  Public  Vidéos
```

Sauvegarde de trois bases de données à la fois :

Dans cet exemple, nous allons sauvegarder les 3 bases ouleye, maison et kamailio qui se trouvent à l'intérieur de notre SGBD.

```
root@toip-pc:~# mysqldump -u root -p --databases ouleye kamailio maison >sauvegarde3.sql
Enter password:
root@toip-pc:~# ls
Bureau            GNS3              Modèles  Public           Vidéos
Documents         Images            Musique  sauvegarde3.sql
examples.desktop  masauvegarde.sql  ok       Téléchargements
```

Sauvegarde de plusieurs tables à l'intérieur d'une base de données :

Avant de sauvegarder les tables, il faut connaître le nom de la base et des tables. Dans cet exemple, nous allons sauvegarder les tables piece et habitants de la base de données maison.

```
mysql>
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| kamailio           |
| l2mic              |
| maison             |
| mysql              |
| ouleye             |
| performance_schema |
| sys                |
+--------------------+
8 rows in set (0,00 sec)

mysql> use maison;
Database changed
mysql> show tables;
+------------------+
| Tables_in_maison |
+------------------+
| habitants        |
| piece            |
+------------------+
2 rows in set (0,00 sec)

mysql> select *from piece;
+---------+---------+----------------+
| idpiece | surface | typepiece      |
+---------+---------+----------------+
|       1 |      20 | chambre enfant |
|       2 |      30 | douche         |
+---------+---------+----------------+
2 rows in set (0,00 sec)
```

```
root@toip-pc:~# mysqldump -u root -p --databases maison --tables piece habitants>sauv_deux.sql
Enter password:
root@toip-pc:~# ls
Bureau     examples.desktop  Images           Modèles  ok      sauv_deux.sql         sauvegarde3.sql  Vidéos
Documents  GNS3              masauvegarde.sql  Musique  Public  sauv_deux_tables.sql  Téléchargements
```

Il est également possible d'effectuer la sauvegarde sans utiliser les options - -databases et - -tables :

```
root@toip-pc:~# mysqldump -u root -p maison piece habitants >sauv_deux_tables.sql
Enter password:
root@toip-pc:~# ls
Bureau             Images            ok                  Téléchargements
Documents          masauvegarde.sql  Public              Vidéos
examples.desktop   Modèles           sauv_deux_tables.sql
GNS3               Musique           sauvegarde3.sql
```

Gestion des utilisateurs : (voir fichier gestion des utilisateurs et des mots de passe)

```
CREATE USER 'jeffrey'@'localhost' IDENTIFIED BY 'password';
```