

**Filière (DUT) : Génie Informatique /S2**

**Département : Génie Informatique**

# **Procédure et Guide de prise en main de Mysql**

**Préparé par :**

**Idriss CHANA**

**Enseignant chercheur au Département GI**

**EST-UMI Meknès**

**[i.chana@umi.ac.ma](mailto:i.chana@umi.ac.ma)**

**Au Profit des Etudiants de La Première Année de la filière  
GI et DWM inscrits en semestre S2**

## I- Introduction

MySQL, est l'un des systèmes de gestion de base de données Open Source le plus populaire. MySQL nous permet de créer notre base de données de la gérer et de la consulter à l'aide des requêtes SQL.

L'objectif de ce guide est permettre à l'étudiant la prise en main de Système de gestion de bases de données Mysql. En effet en suivant les étapes de ce guide l'étudiant procédera à l'installation de Mysql sur sa machine personnelle, ensuite utilisera Mysql pour créer et définir sa Première base de données à l'aide des instruction SQL, **Create , Alter , Drop** qui sont des commandes **LDD** ( langage de définition de données) .

Ce Guide contient l'essentiel à acquérir pour se familiariser avec le système de gestion Mysql et avec sa syntaxe relative au Langage de définition de données LDD :

- **Installation de MySQL**
- **Démarrage de Mysql et création de la base de données**
- **Les commande : use, describe ,show de Mysql**
- **Manipulation de LDD ( Create Alter , Rename, Drop)**
- **Exercices d'entraînement**

il servira aussi de base pour entamer tous les TP programmées pour le l'élément du Module « **SI et Bases de données** ». vous êtes donc invités à l'enregistrer ou à l'imprimer et le ramener avec vous durant les séances de TP

NB :

- Les réponses aux questions posées à la fin de chaque étape vous permettront de maîtriser et d'appréhender l'objectif de l'étape en cours avant de passer l'étape suivante
- Vers la fin 3 exercices d'entraînement sont proposés pour mieux assimiler et appréhender l'exploitation de SGBDR MySQL.

## II- Installation de MySQL

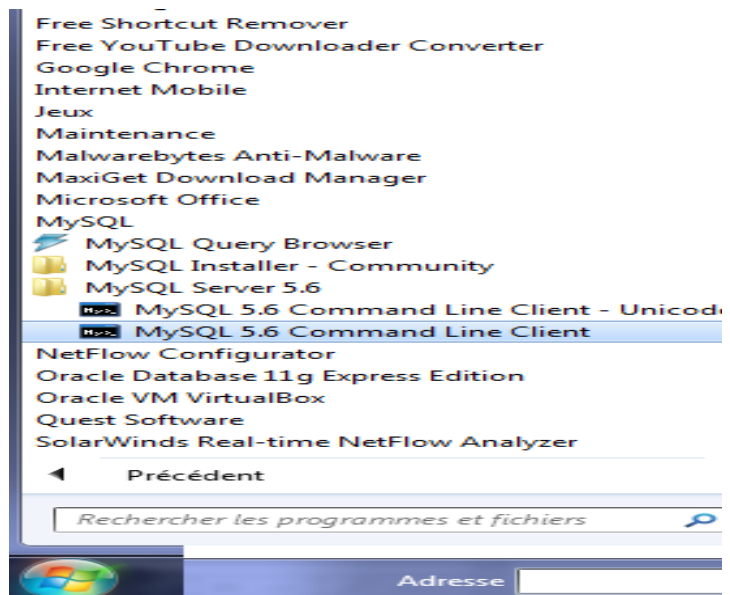
Cette section vous guide à l'installation de MySQL, si vous souhaitez l'installer sur vos machines personnelles.

- Pour l'Installation sous Windows
- Aller sur la page de téléchargement de MySQL :  
<http://dev.mysql.com/downloads/mysql/>
- Aller sur "Select Platform" et choisir "Microsoft Windows".
- Télécharger le "MySQL Installer". Il est possible de s'enregistrer en tant que nouvel utilisateur, ou de procéder sans enregistrement.
- Exécuter le fichier d'installation et installer MySQL en tant que service Windows.
- Créer un compte avec un mot de passe.

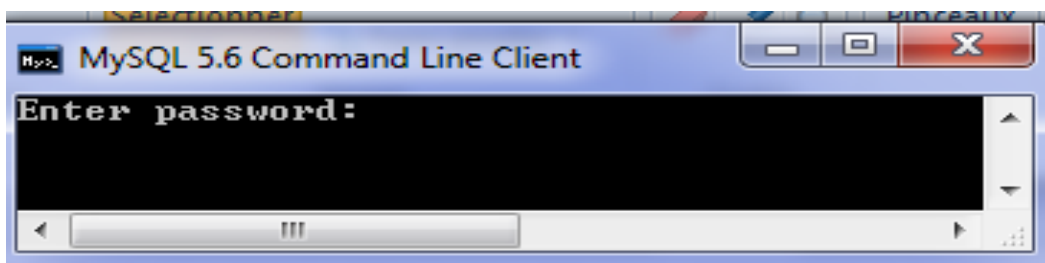
### III- Démarrage de MySQL

Pour démarrer MySQL vous :

1. Sélectionnez le bouton Démarrer
2. Sélectionnez Tous les programmes puis MySQL
3. Sélectionnez le serveur MySQL 5.6
4. Cliquez sur le client de ligne de commande MySQL



Une fenêtre de commande MySQL va s'ouvrir alors ouvrir comme le montre la figure ci-dessous

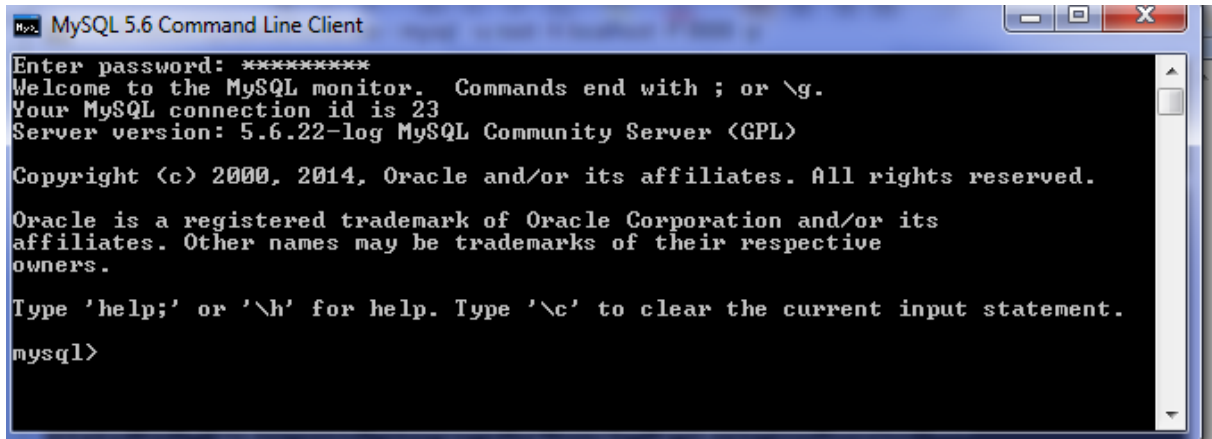


Le mot de passe demandé est celui créé lors de l'installation (généralement il correspond au super utilisateur **root**), une fois vous taper le mot de passe correctement

**Important : si vous utilisez xamp,**

Aller dans **shell** et taper → **mysql -u root**

vous aurez l'écran suivante :



```
MySQL 5.6 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 5.6.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2014, 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>
```

Vous êtes maintenant bien connecté au serveur **Mysql**,

Pour travailler dans MySQL, il est demandé de taper des commandes. Par exemple en tapant **Mysql> \h** : Ça vous donne la liste complète des commandes **Mysql** avec une courte description

**Q1- Taper** : **Mysql> help 'create databse'**

Expliquer le résultat de cette commande ?

#### **IV- -Création d'une base de données**

##### **1- Visualiser les bases de données existantes**

Pour visualiser les bases de données qui existe-t-on tape la commande

**Mysql> show databases**

**Q2- Quelles sont les bases de données qui existent ?**

##### **2- Créer une nouvelle base de données**

Pour créer une base de données il suffit de taper la commande

**Mysql> create database nom\_ base\_ données**

Avec **nom\_ base\_ données** est le nom qu'on veut donner à notre nouvelle base de données

**Q3- créer la base de donné ayant le nom ESTMGI1 s'elle existe le nom sera ESTMGI2**

**Q4- vérifier est ce qu'elle st bien crée**

##### **3- Utiliser/basculer une base de données**

Pour utiliser une base de données ou de basculer vers une autre de données taper la commande **use**

**Mysql> use nom\_base\_données**

**Q5-** Utiliser la base de données **que vous avez créé**

Pour visualiser les tables d'une base de données on tape la commande **show tables**

**Mysql> show tables**

**Q6-** taper cette commande et commenter !

### **V- Création des Modification de la structure des Tables.**

Dans cette section nous allons procéder à l'utilisation de SQL ( à travers le LDD) de MY SQL nous allons découvrir comment :

- Déclarer Les types de données des colonnes des tables Mysql
- Créer supprimer des tables
- Modifier la structure de tables existantes (ajouter/supprimer des colonnes, modifier les type)
- Ajouter et définir des contraintes)

### **1- Création des tables**

La syntaxe de création d'une table est la suivante :

Syntaxe de création	Exemple
<pre>CREATE TABLE nom_table (   nom_col_1 type_col_1,   nom_col_2 type_col_2,   ...   nom_col_n type_col_n );</pre>	<pre>CREATE TABLE <b>employee</b> ( empno INT not null,   nom varchar (20) ,   Fonction varchar (10),   Sal DECIMAL (8,2),   code_service char(3),   date_embauche date, PRIMARY KEY (empno) ) ;</pre>

**Q7-** se connecter à votre base de donnée en utilisant la commande **use** , et créer la table **employee** de l'exemple

**Q8-** taper ensuite la commande : **Mysql> describe employe** puis commenter le resultat

A partir de l'emple ci-dessus nous remarquons que la création de la table a nécessité la déclaration des type de chaque colonne selon Mysql , Ci-dessous les types supporté par

Mysql que peut avoir un champ d'une table, et qui est soit **numérique**, ou **caractère** ou bien **date /heure**

### Les champs numériques :

TINYINT	Entier très petit	1 octet
SMALLINT	Entier petit compris entre -32768 et 32767, si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 65535.	2 octets
MEDIUMINT	Entier moyen compris entre -8388608 et 8388607, si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 16777215	3 octets
INT	Entier standard compris entre -2 147 483 648 et 2 147 483 647. Si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 4 294 967 295	4 octets
BIGINT	Entier grand	8 octets
FLOAT	Décimal de simple précision	4 octets
DOUBLE, REAL	Décimal de double précision	8 octets
DECIMAL (entier,décimal)	Réel, définissez la longueur de chacune des deux parties.	Variable

### Les chaînes de caractères :

CHAR (n)	Chaîne de n caractère, taille fixe	
VARCHAR(M)	Chaîne de caractères variable. M peut être compris entre 1 et 255.	255 caractères. Maximum
TINYBLOB, TINYTEXT	Petite zone de texte. Objet d'une longueur maximale de 255 caractères, <b>TINYTEXT</b> aura un contenu de type ASCII (casse insensible) et <b>TINYBLOB</b> aura un contenu de type binaire (casse sensible).	255 caractères. Maximum
BLOB, TEXT	Zone de texte standard Objet d'une longueur maximale de 65535 caractères. <b>TEXT</b> aura un contenu de type ASCII (casse insensible) et <b>BLOB</b> aura un contenu de type binaire (casse sensible).	65 535 caractères. Maximum

### Les champs de Types date et heure :

DATE	Date (ex: 2000-08-24)	3 octets
TIME	Heure (ex: 23:44:05)	3 octets
DATETIME	Date et heure (ex: 2000-08-24 23:44:05)	8 octets

YEAR	Année (ex: 2000)	1 octet
------	------------------	---------

## 2- Modification de la structure de la table

Quand nous créons une table on a besoin des fois de changer sa structure par l'ajout ou la suppression d'une ou plusieurs colonnes ou de modifier leurs types, changer le nom d'une colonne changer le nom de la table ect ... les syntaxes Mysql qui nous permettent cela sont résumés dans le tableau ci-dessous

	Syntaxe	Exemples
Ajout de colonnes	<code>ALTER TABLE nom_table ADD (nom_colonne1 type_colonne1, nom_colonne2 type_colonne, .... );</code>	→ <code>ALTER TABLE service ADD (budget decimal (6,2), sous_traitant varchar(30) );</code>
Suppression de colonnes	<code>ALTER TABLE nom_table DROP COLUMN nom_colonne1, DROP COLUMN nom_colonne2, DROP COLUMN nom_colonnen...);</code>	→ <code>ALTER TABLE service DROP COLUMN budget , drop column sous_traitant ;</code> → <code>ALTER TABLE service DROP COLUMN budget , add ( id_chef int) ;</code>
Modification de propriété d'une colonne	<code>ALTER TABLE nom_table MODIFY nom_colonne1 nouveau_type1 , MODIFY nom_colonne2 nouveau_type2</code>	→ <code>ALTER TABLE service MODIFY budget decimal (8,3) ;</code>
Changement du nom de colonne	<code>ALTER TABLE nom_table change old_nom new_nom new_type</code>	→ <code>ALTER TABLE employe change Fonction job varchar(12) ;</code>
Supprimer une table	<code>DROP TABLE nom_table;</code>	→ <code>Drop table employe ;</code>
Renommer une table:	<code>RENAME TABLE nom_table TO nouveau_nom_table;</code>	→ <code>RENAME TABLE employe to salarie ;</code>

## 3- Les contraintes d'intégrité

Une contrainte d'intégrité est une règle qui permet d'assurer la validité (cohérence) des données stockées dans une base.

Le SGBD contrôle les contraintes d'intégrité à chaque modification dans les tables (saisies, modification ou suppression).

Ci-dessous quelques contraintes que l'on peut définir :

- ✓ **non nullité** (obligation) : **NOT NULL** : La colonne ne peut pas contenir de valeurs NULL.
- ✓ **Incrémentation automatique** : **AUTO\_INCREMENT** : Le champ ayant cette contrainte est incrémenté automatiquement
- ✓ **valeur par défaut** : **DEFAULT** parfois on désire attribuer aux champs une valeur par défaut
- ✓ **unicité** : **UNIQUE** Toutes les valeurs de la (des) colonnes doivent être différentes ou NULL
- ✓ **clé primaire** : **PRIMARY KEY** : **primary key = unique + not null**
- ✓ **intégrité référentielle (clé étrangère)** : **FOREIGN KEY** Cette colonne fait référence à une colonne clé d'une autre table.

Le tableau ci-dessous résume la syntaxe de déclaration et définition de déférente contrainte que peut avoir un attribut avec des exemples selon la syntaxe **Mysql**

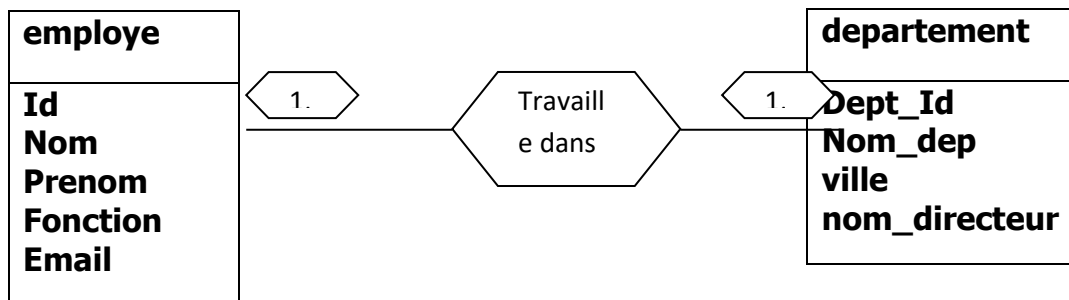
Nom de la contrainte	Syntaxe	Exemples
<b>Clé primaire</b>	<p><b>Soit au début lors de la déclaration</b></p> <p><b>Ou bien par la syntaxe</b></p> <pre>ALTER TABLE nom_table ADD PRIMARY KEY (colonne_clé)</pre>	<p>➔ CREATE TABLE Persons ( P_Id int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), PRIMARY KEY (P_Id,LastName) )</p> <p>➔ ALTER TABLE Personnes ADD PRIMARY KEY (P_Id)</p>
<b>Clé étrangère</b>	<p><b>Soit au début lors de la déclaration</b></p> <p><b>Ou bien par la syntaxe</b></p> <pre>ALTER TABLE nom_table ADD CONSTRAINT FOREIGN KEY (col_clé_etran) REFERENCES table_ref (col_clé_etran)</pre>	<p>➔ CREATE TABLE Orders ( O_Id int NOT NULL, OrderNo int NOT NULL, P_Id int, PRIMARY KEY (O_Id), FOREIGN KEY (P_Id) REFERENCES Persons(P_Id)) ;</p> <p>➔ ALTER TABLE Orders ADD CONSTRAINT FOREIGN KEY (P_Id) REFERENCES Persons(P_Id)</p>

<b>NOT null</b>	<p>Soit au début lors de la déclaration</p> <p>ou bien</p> <pre>ALTER TABLE nom_table MODIFY (nom_col NOT NULL);</pre>	<pre>ALTER TABLE Person MODIFY (P_Id NOT NULL);</pre>
<b>unicité : UNIQUE</b>	<p>Soit au début lors de la déclaration</p> <p>ou bien</p> <pre>alter table nom_table add constraint unique(col_unique);</pre>	<p>→ <pre>alter table emp add constraint unique(email);</pre></p>
<b>Valeur par défaut</b>	<p>Soit au début lors de la déclaration</p> <p>ou bien</p> <pre>ALTER TABLE nom_table ALTER COLUMN contactname SET DEFAULT 'Unknown'</pre>	<p>→ <pre>CREATE TABLE TEST ( Nom char (20) default 'toto') ;</pre></p> <p>→ <pre>ALTER TABLE emp ALTER COLUMN nom SET DEFAULT 'Inconnu'</pre></p>

## VI- Entraînement

### Exercice 1

Soit le MCD suivant



Id,dept\_id : numérique clé primaire

Nom,prenom,fonction,ville : chaînes de caractère variable de taille 15

Nom\_directeur,email : chaînes de caractère de fixe de taille 15 et Unique

**Q9-** transformer ce MCD en MLD et le schéma relationnel et écrire le code SQL selon Mysql qui permet la création des les tables

**Q10-** réaliser ce MCD sous Power AMC, puis générer le scripte de création de la base de données en passant par la génération de MPD

**Q11-** Générer la base de données en éditant le scripte. Puis, comparer entre votre code et celui généré par Power AMC que remarquez vous

## Exercice 2

- Utiliser la syntaxe de création et les type de données et créer les deux tables suivantes **ADMIN** (adminId , sur\_name , firstname ,adresse, departement , Email ) et **USER** (User\_id , login , nom , Email)

le tableau ci-dessous résume les type de chaque champ des deux tables

Champs	Type
adminId , département , User_name , login ,nom ,firstname	Chaîne de taille fixe de 20 caractères,
Email , adresse	Chaîne variable de taille 30 ,
User_id	Petit entier

**Q12-** visualiser la structure des deux tables

**Q13-** ajouter la colonne **propriety** de type entier à la table **ADMIN**

**Q14-** Supprimer la colonne email de la table **ADMIN**

**Q15-** Changer la propriété de la colonne **Nom** en chaîne de taille variable de longueur **10** dans la table **USER**

**Q16-** En une seule commande changer le nom de la colonne **adresse** en **local** et le type de **adminId** en **très petit Entier**

**Q17-** Changer le nom de la table **ADMIN** en **ADMINISTRATEUR**

**Q18-** Vérifier les changements dans les deux tables

**Q19-** Supprimer les deux tables

## Exercice 3

Soit le schéma relationnel composé des relations(tables) suivantes :

- ELEVES (CNE\*, Nom, prenom, age , sexe)
  - MODULES ( **code\_module\***, intitulé)
  - EVALUATION ( **code\_module\***, **CNE\***, note, resultat).
- Les champs en gras marqués en étoile sont des clés primaires

- Les champs **CNE** et **code\_module** respectivement dans les relations **ELEVES** et **MODULES** contribuent à la clé primaire de la table **EVALUATION** et y sont en même temps des clés étrangères.

Attributs	Type	Valeur par défaut
Nom, prenom, sexe, intitulé, resultat	Chaine de caractères	inconnu
age , code_module, CNE, note	Numériques	

**Q20-** Créer la base de données **GESTION\_ETUDE**

**Q21-** Se connecter à la base **GESTION\_ETUDE** et créer les tables ci-dessus en déclarant les **clés** primaires et étrangères.

**Q22-** Ajouter la colonne adresse de type **chaîne de caractère** à la table **ELEVES**.

**Q23-** Ajouter la colonne **email** à la table **ELEVES** avec la contraint **d'unicité**

**Q24-** Renommer la colonne **code\_module** par **code**

**Q25-** Renommer la table **ELEVES** par **ETUDIANTS**

**Q26-** supprimer les tables