

# SQL (mysql)

**Université de Caen-Normandie**

Bruno CRÉMILLEUX

À partir d'un terminal :

```
mysql -h mysql.info.unicaen.fr -u LOGIN -p
```

exemple :

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
[...]  
MariaDB [(none)]>
```

valeurs des paramètres de connexion : répertoire  
~/Protected/mysql.txt de votre home.

informations à <https://faq.info.unicaen.fr/bdd>

**La première fois, il faut créer sa base :**

sous mysql :

```
MariaDB [(none)]> CREATE DATABASE LOGIN_bd ;
```

puis se connecter à sa base :

```
MariaDB [(none)]> use LOGIN_bd
```

**Pour les connexions suivantes à la base LOGIN\_bd :**

à partir d'un terminal :

```
mysql -h mysql.info.unicaen.fr -u LOGIN -p LOGIN_bd
```

Deux types de commandes :

- commandes de mysql:
  - \? ou help : aide
  - \q : quitter
  - \. FILE (ou source FILE) : exécute le script sql FILE
  - \! COMMAND : exécute la commande shell COMMAND
  - \T FILE : redirige la sortie dans le fichier FILE
  - ...
- commandes SQL : SELECT, CREATE, INSERT,...

Une requête SQL se termine par un ;

**Bonne habitude de travail :** préparer les requêtes via un éditeur de texte et charger le script contenant les requêtes avec \. (ou source)

Liste des tables : MariaDB [LOGIN\_bd]> `show tables ;`

```
+-----+
| Tables_in_cremilleux_bd |
+-----+
| CLIENT                  |
| COMMANDE                |
| DETAIL                  |
| PRODUIT                  |
+-----+
```

Schéma d'une table : MariaDB [LOGIN\_bd]> `describe CLIENT ;`

```
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| RefC  | int(11)       | NO   | PRI | NULL    |       |
| NomC  | varchar(20)   | NO   |     | NULL    |       |
| Ville | varchar(20)   | NO   |     | NULL    |       |
| CAT   | varchar(2)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

Contenu d'une table : MariaDB [LOGIN\_bd]> **SELECT** \* from CLIENT ;

```
+-----+-----+-----+-----+
| RefC | NomC      | Ville      | CAT  |
+-----+-----+-----+-----+
| 1 | GOFFIN    | Namur      | B2   |
| 2 | HANSENNE  | Poitiers   | C1   |
| 3 | MONTI     | Geneve     | B2   |
...
15 rows in set (0,00 sec)
```

Nous reviendrons sur le **SELECT**

### Commentaire :

- une ligne : # ou --
- plusieurs lignes : /\* ... \*/

CREATE TABLE... :

description de chaque attribut :

- nom de l'attribut (une chaîne de caractères)
- type de l'attribut : entier, réel, chaîne, date,...
- propriétés de l'attribut : clé, NOT NULL, contraintes,...

```
CREATE TABLE DETAIL(  
    RefCOM INT NOT NULL,  
    RefP  VARCHAR(5) NOT NULL,  
    Quantite INT NOT NULL,  
    PRIMARY KEY (RefCOM, RefP)  
);
```

PRIMARY KEY (RefC) : déclaration d'une clé (identifiant unique pour chaque n-uplet)

La table est **vide** après sa création.

Deux possibilités :

- via un script SQL (commande INSERT) :

```
INSERT INTO CLIENT VALUES (1, 'GOFFIN', 'Namur', 'B2') ;  
INSERT INTO CLIENT VALUES (2, 'HANSENNE', 'Poitiers', 'C1') ;  
...
```

- via le chargement d'un fichier texte (cf. TP) :

syntaxe `mysql` : (utiliser `\copy` en `postgres`)

```
LOAD DATA LOCAL INFILE "client.dat" INTO TABLE CLIENT ;
```

où `client.dat` contient :

```
1 GOFFIN Namur B2  
2 HANSENNE Poitiers C1  
...
```

Il est possible que vous deviez explicitement activer lors de votre connexion la possibilité de chargement d'un fichier, cf. informations à

<https://faq.info.unicaen.fr/bdd>