Apprendre le SQL

Le **SQL** (Structured Query Language) est un langage permettant de communiquer avec une base de données.

MySQL

MySQL est un SGBDR (System de Gestion de bases de donnes Relationnel), qui utilise le langage SQL.

Dans ce résumer nous allons traité les principales commandes SQL telles que:

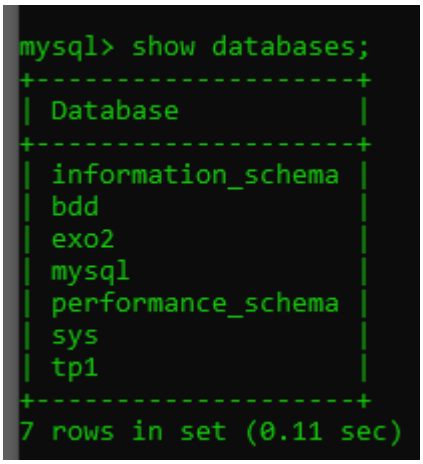
SELECT, INSERT INTO, UPDATE, DELETE, DROP TABL, etc… .

De plus nous allons voir quelques fonctions SQL telles que : AVG (), COUNT (), MAX (), etc…

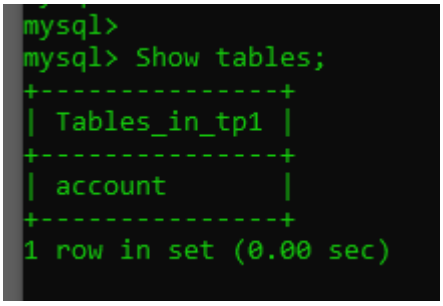
SHOW :

Show est utiliser pour afficher des informations sur une base de données

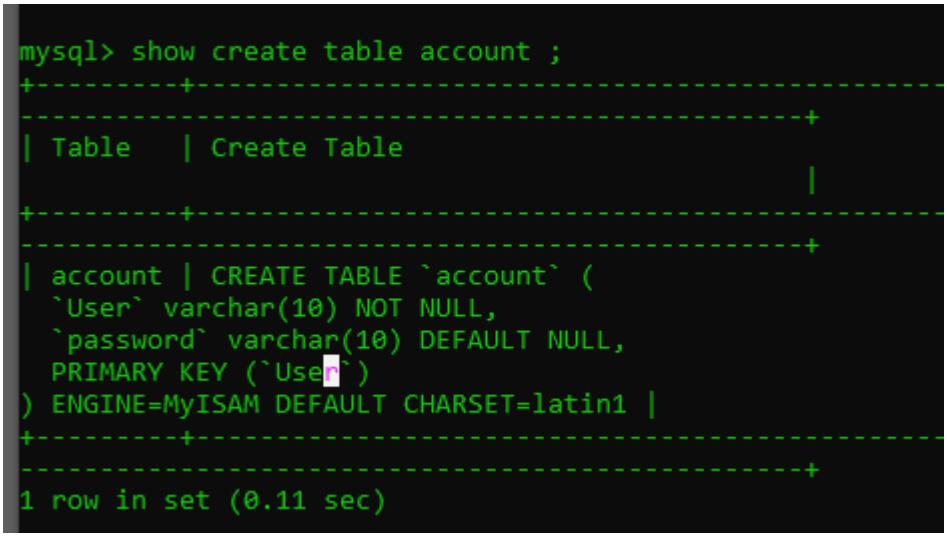
SHOW DATABASES ; // pour afficher tous les Bases de données qui existe dans notre SGBDR.



SHOW Tables ; // pour afficher toutes les tables qui existe dans notre Bases de données.

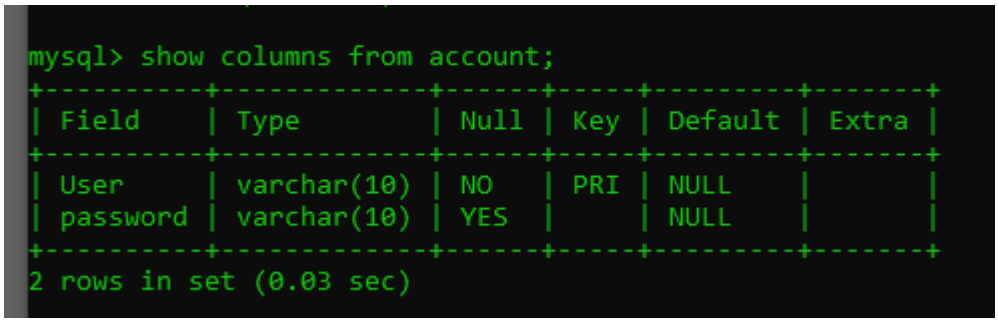


Show CREATE Table Name_table ; // pour afficher les instructions de la création de la table.



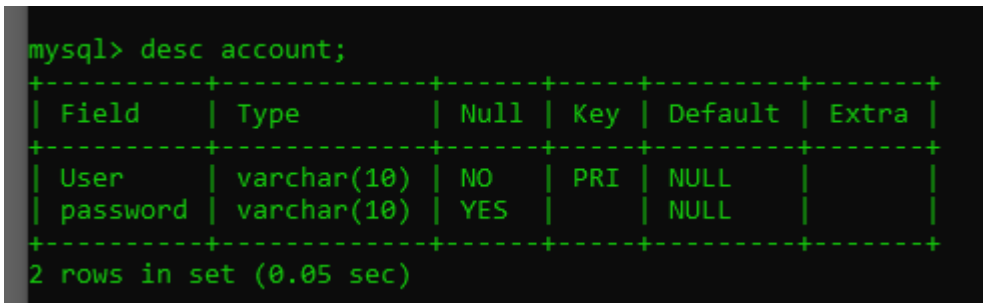
```
mysql> show create table account ;
+---------+-------------------------------------------------
-------------------------------------------------+
| Table   | Create Table
                                                 |
+---------+-------------------------------------------------
-------------------------------------------------+
| account | CREATE TABLE `account` (
  `User` varchar(10) NOT NULL,
  `password` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`User`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 |
+---------+-------------------------------------------------
-------------------------------------------------+
1 row in set (0.11 sec)
```

Show Columns From Name_table; // pour afficher la description de la table



```
mysql> show columns from account;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| User     | varchar(10) | NO   | PRI | NULL    |       |
| password | varchar(10) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.03 sec)
```

note : on peut remplacer ’ Show Columns From ‘ par ‘DESC’.



```
mysql> desc account;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| User     | varchar(10) | NO   | PRI | NULL    |       |
| password | varchar(10) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.05 sec)
```

CREATE:

CREATE est utiliser pour crée des objets dans une base de donner en SQL.

CREATE DATABASE Name_DB; // Exemple CREATE DATABASE tp1 ;

CREATE TABLE Name_table (….);

```
mysql> CREATE TABLE hotel (
    -> numhotel int,
    -> nomhotel varchar(30),
    -> ville char(20)
    -> );
Query OK, 0 rows affected (0.12 sec)
```

USE:

Use est utiliser pour sélectionner une base de données existant ;

USE Name_BDD ; // Exemple : USE tp1 ;

SELECT:

Select est considéré comme un afficheur (printf), on peut afficher avec une chaine ou un résultat d'une équation mathématique, et généralement on l'utilise pour afficher nos tables dans une BDD.

```
mysql> select "Hello word" ;
+------------+
| Hello word |
+------------+
| Hello word |
+------------+
1 row in set (0.00 sec)
```

Afficher une chaine

```
mysql> Select 5+5+6+9+8+9+(5*8) ;
+--------------------+
| 5+5+6+9+8+9+(5*8) |
+--------------------+
|                 82 |
+--------------------+
1 row in set (0.06 sec)
```

Afficher une Equation

Dans le cas d'affichage du tableau :

Select **nom_column1**, **nom_column2** From **Nom_Table**; // pour afficher les colonne que l'on veut:

```
mysql> select numhotel from hotel ;
+----------+
| numhotel |
+----------+
|       12 |
|       13 |
|       14 |
|       15 |
|       16 |
|       17 |
+----------+
6 rows in set (0.03 sec)
```

Sinon si on veut afficher toute la table on peut utiliser (*) a la place des noms de la colonne :

SELECT * FROM hotel;

```
mysql> select * from hotel ;
+----------+------------+-----------+
| numhotel | nomhotel   | ville     |
+----------+------------+-----------+
|       12 | AZ         | Alger     |
|       13 | amine      | Alger     |
|       14 | amine      | Boumerdes |
|       15 | boudouaou  | Boumerdes |
|       16 | leparisien | parie     |
|       17 | darr       | Adrar     |
+----------+------------+-----------+
6 rows in set (0.00 sec)
```

Aussi si on veut préciser notre recherche dans la table nous allons utiliser WHERE Exemple :

SELECT * FROM hotel WHERE nomhotel = ‘AZ’;

```
mysql> SELECT * FROM hotel WHERE nomhotel = 'AZ';
+----------+----------+-------+--------+
| numhotel | nomhotel | ville | etoile |
+----------+----------+-------+--------+
|       12 | AZ       | Alger |   NULL |
+----------+----------+-------+--------+
1 row in set (0.09 sec)
```

Si nous voulant sélectionner tous les attributs qui contiennent un alphabet précise nous allons utiliser le mot clé LIKE a la place de =, et le % aussi :

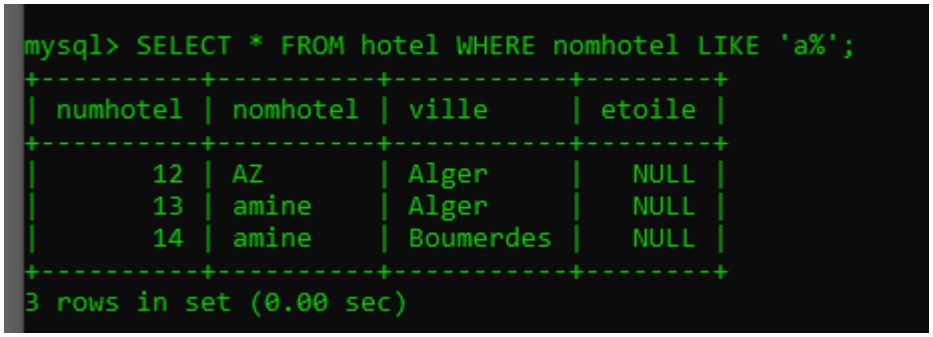
SELECT * FROM hotel WHERE nomhotel LIKE ‘%A%’;

% on le lit :

%A% : quelques soit + A + quelques soit ; // un mot qui contient ‘a’

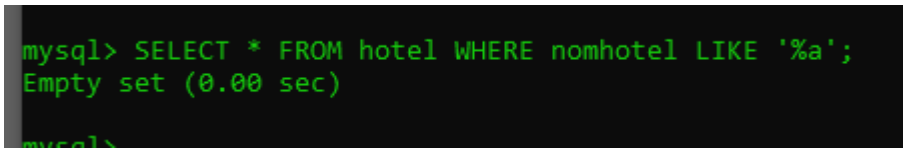
```
mysql> SELECT * FROM hotel WHERE nomhotel LIKE '%a%';
+----------+------------+-----------+--------+
| numhotel | nomhotel   | ville     | etoile |
+----------+------------+-----------+--------+
|       12 | AZ         | Alger     |   NULL |
|       13 | amine      | Alger     |   NULL |
|       14 | amine      | Boumerdes |   NULL |
|       15 | boudouaou  | Boumerdes |   NULL |
|       16 | leparisien | parie     |   NULL |
|       17 | darr       | Adrar     |   NULL |
|       18 | EL AMINE   | Boumerdes |   NULL |
+----------+------------+-----------+--------+
7 rows in set (0.00 sec)
```

A% : A + quelques soit ; // un mot qui débute par ‘a’



```
mysql> SELECT * FROM hotel WHERE nomhotel LIKE 'a%';
+----------+----------+-----------+--------+
| numhotel | nomhotel | ville     | etoile |
+----------+----------+-----------+--------+
|       12 | AZ       | Alger     |   NULL |
|       13 | amine    | Alger     |   NULL |
|       14 | amine    | Boumerdes |   NULL |
+----------+----------+-----------+--------+
3 rows in set (0.00 sec)
```

%A : quelques soit + A ; // un mot qui se termine par ‘a’

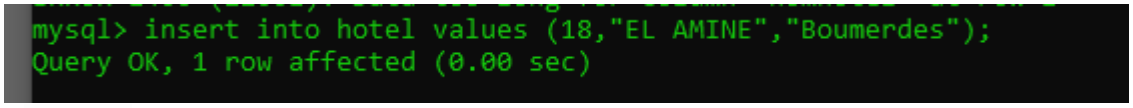


```
mysql> SELECT * FROM hotel WHERE nomhotel LIKE '%a';
Empty set (0.00 sec)
```

INSERT:

Insert est utiliser pour ajouter des éléments dans une table, pour cela on utilise :

INSERT INTO Name_Table VALUES (valeur1, valeur2…);



```
mysql> insert into hotel values (18,"EL AMINE","Boumerdes");
Query OK, 1 row affected (0.00 sec)
```

ALTER:

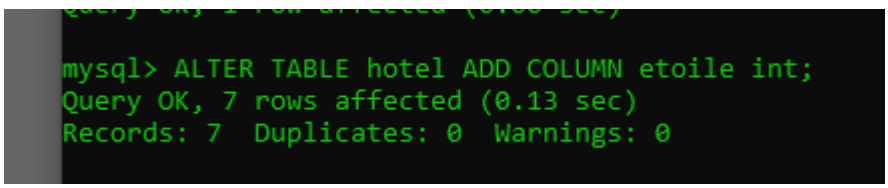
Alter est un mot clé pour modifier la conception de la table elle-même, il suffit juste d’écrire :

Alter Table Name_table [mot clé] instruction;

Nous avons dans le mot clé : drop et add qu’on va utiliser pour supprimer ou ajouter des instruction.

Exemple si on veut ajouter une Colonne Nombre Etoile on doit utiliser la syntaxe suivante :

ALTER TABLE Name_table ADD COLUMN Name_column Type_column;



```
mysql> ALTER TABLE hotel ADD COLUMN etoile int;
Query OK, 7 rows affected (0.13 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

```
mysql> select * from hotel ;
+----------+------------+-----------+--------+
| numhotel | nomhotel   | ville     | etoile |
+----------+------------+-----------+--------+
|       12 | AZ         | Alger     |   NULL |
|       13 | amine      | Alger     |   NULL |
|       14 | amine      | Boumerdes |   NULL |
|       15 | boudouaou  | Boumerdes |   NULL |
|       16 | leparisien | parie     |   NULL |
|       17 | darr       | Adrar     |   NULL |
|       18 | EL AMINE   | Boumerdes |   NULL |
+----------+------------+-----------+--------+
7 rows in set (0.00 sec)
```

Exemple si on a oublié de définir une clé primaire a notre table donc pas de panique, nous pouvant la définir avec ALTER comme suite :

ALTER TABLE Name_table ADD PRIMARY KEY (Select_Column);

```
mysql> ALTER TABLE hotel ADD PRIMARY KEY (numhotel);
Query OK, 0 rows affected (0.13 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Et comme nous le remarquant, notre table a maintenant une clé primaire :

```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| numhotel | int(11)     | NO   | PRI | NULL    |       |
| nomhotel | varchar(10) | YES  |     | NULL    |       |
| ville    | varchar(10) | YES  |     | NULL    |       |
| etoile   | int(11)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

On peut aussi utiliser Drop au lieu de ADD pour supprimer quelque chose dans notre table :

ALTER TABLE Name_table DROP COLUMN Name_column;

```
mysql> ALTER TABLE hotel DROP COLUMN etoile;
Query OK, 7 rows affected (0.09 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

```
mysql> select * from hotel
    -> ;
+----------+------------+-----------+
| numhotel | nomhotel   | ville     |
+----------+------------+-----------+
|       12 | AZ         | Alger     |
|       13 | amine      | Alger     |
|       14 | amine      | Boumerdes |
|       15 | boudouaou  | Boumerdes |
|       16 | leparisien | parie     |
|       17 | darr       | Adrar     |
|       18 | EL AMINE   | Boumerdes |
+----------+------------+-----------+
7 rows in set (0.06 sec)
```

UPDATE :

Update permet de modifier une instance d'une table, on peut changer le nom ou prénom ou même les deux selon ce d'ont on a besoin.

La syntaxe est comme suit :

Update name_table **SET** column_1= 'valeur1', column_2= 'valeur2' **WHERE** condition

```
mysql> update hotel set nomhotel='Daufun' Where numhotel=14 ;
Query OK, 1 row affected (0.08 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from hotel ;
+----------+------------+-----------+
| numhotel | nomhotel   | ville     |
+----------+------------+-----------+
|       12 | AZ         | Alger     |
|       13 | amine      | Alger     |
|       14 | Daufun     | Boumerdes |
|       15 | boudouaou  | Boumerdes |
|       16 | leparisien | parie     |
|       17 | darr       | Adrar     |
|       18 | EL AMINE   | Boumerdes |
+----------+------------+-----------+
7 rows in set (0.00 sec)
```

après

```
mysql> select * from hotel
    -> ;
+----------+------------+-----------+
| numhotel | nomhotel   | ville     |
+----------+------------+-----------+
|       12 | AZ         | Alger     |
|       13 | amine      | Alger     |
|       14 | amine      | Boumerdes |
|       15 | boudouaou  | Boumerdes |
|       16 | leparisien | parie     |
|       17 | darr       | Adrar     |
|       18 | EL AMINE   | Boumerdes |
+----------+------------+-----------+
7 rows in set (0.06 sec)
```

Avant

DELETE :

Delete est une commende pour supprimer une ou plusieurs lignes (selon la condition) dans une table SQL.

La requête s'écrit comme ça :

DELETE FROM name_table WHERE condition;

```
mysql> DELETE FROM hotel WHERE ville='boumerdes';
Query OK, 3 rows affected (0.06 sec)
```

```
mysql> select * from hotel ;
+----------+------------+-------+
| numhotel | nomhotel   | ville |
+----------+------------+-------+
|       12 | AZ         | Alger |
|       13 | amine      | Alger |
|       16 | leparisien | parie |
|       17 | darr       | Adrar |
+----------+------------+-------+
4 rows in set (0.00 sec)
```

DROP:

On utilise DROP pour supprimer un objet (Table, BDD, View…) ;

DROP name_table;

Drop name_BDD;

PRIMARY KEY:

On a 3 style pour déclarés une clé primaire (Primary key)

La premier:

```
CREATE TABLE hotel (
numhotel int PRIMARY KEY,
nomhotel varchar(30),
ville char(20)
);
```

Le 2eme:

```
CREATE TABLE hotel (
numhotel int,
nomhotel varchar(30),
ville char(20),
PRIMARY KEY (numhotel)
);
```

Le 3eme:

```
CREATE TABLE hotel (
numhotel int,
nomhotel varchar(30),
ville char(20),
CONSTRAINT pk_numhotel PRIMARY KEY (numhotel)
);
```

Et les trois font exactement le même travail, personnellement je préfère la première.

Note : Pour les clés composées on peut utiliser que le 2eme et 3eme style d'écriture.

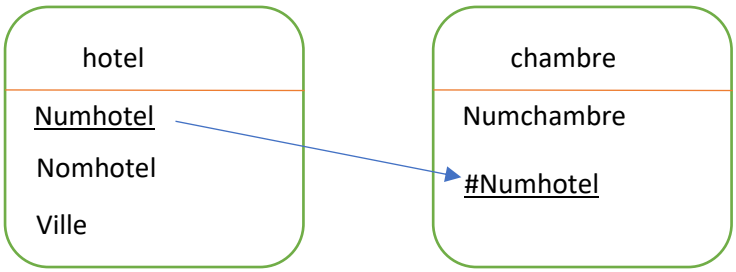
Clé composée :

Supposant qu'on veut faire la composition du (nom + le numero d'hotel) comme clé primaire, alors nous allons l'écrire comme cela :

```
CREATE TABLE hotel (
numhotel int,
nomhotel varchar(30),
ville char(20),
PRIMARY KEY ( numhotel,nomhotel )
);
```

FOREIGN KEY:

Pour la clé étrangère, comme on le sait, elle et une clé qui viens d'une autre table.



Ça syntaxe s'écrit comme suit :

```
FOREINGN KEY (numhotel) REFERENCES hotel (Numhotel)
```

Attribu de la table Chambre

Attribu de la table hotel

Et pour mieux comprendre la syntaxe on peut le lire comme ça :

numhotel de la table chambre devient une référence de Numhotel de la table hotel

```
CREATE TABLE Chambre (
Numchambre int,
numhotel int,
FOREINGN KEY (numhotel) REFERENCES hotel (Numhotel)
);
```