

Syntaxe SQL pour cr  er la table : clients_tbl

```
CREATE TABLE clients_tbl (id INT not null AUTO_INCREMENT, prenom VARCHAR (50) not null , nom
VARCHAR (50) not null , ne_le DATE not null , ville VARCHAR (90) not null , enfants INT not null ,
PRIMARY KEY (id))
```

Ci-dessous la table : clients_tbl une fois cr   e

```
+++++
+ id + prenom + nom + ne_le + ville + enfants +
+++++
Bien s  r pour le moment cette table ne comporte aucun enregistrement ...
```

La commande INSERT INTO

Cette commande permet d'ins  rer des enregistrements dans une table en l'occurrence clients_tbl.

- INSERT INTO clients_tbl(id,prenom,nom,ne_le,ville,enfants)
VALUES('','Patrick','Martin','1965-10-08','Bordeaux','2')

ou bien cette autre requ  te qui aura le m  me r  sultat, mais que nous vous d  conseillons fortement, car elle ne marchera plus si vous modifiez votre table, en particulier si vous ajoutez un champ :

- INSERT INTO clients_tbl VALUES('','Patrick','Martin','1965-10-08','Bordeaux','2')

Ci-dessous la table : clients_tbl avec le nouvel enregistrement

```
+++++
+ id + prenom + nom + ne_le + ville + enfants +
+++++
+ 1 + Patrick + Martin + 1965-10-08 + Bordeaux + 2 +
+++++
```

Dans le premier exemple nous avons sp  cifi   les noms des champs (entre parenth  ses) juste apr  s le nom de la table. Dans ce cas il n'est pas obligatoire de le faire si dans les valeurs vous sp  cifiez une valeur par champs, mysql affectera les valeurs dans l'ordre donn  .

Par contre si vous ne donnez que deux valeurs, il sera important de toujours sp  cifier les champs dans lesquels elle doivent   tre ins  r  es, exemple :

- INSERT INTO clients_tbl(id,nom) Values('','Dupond')
- ou encore :
- INSERT INTO clients_tbl(nom,enfants) Values('Dupond','2')

Notez : La valeur du champ id est vide, je vous rappelle que ce champ est en auto-increment dans notre base, un nombre automatique sera donc attribu      chaque nouvel enregistrement, dans ce cas la valeur dans la requ  te peut rester vide !

Important : Si vous voulez n'ins  rer que quelques-unes des valeurs d'un enregistrement , vous devrez avoir sp  cifi   lors de la cr  ation de la table que les autres champs peuvent rester vides ! (NULL).

La commande SELECT

Nous allons travailler    partir de la table ci-dessous qui comporte 5 enregistrements :

Ci-dessous le contenu final de la table : clients_tbl

```

+++++++
+  id  +  prenom  +  nom  +  ne_le  +  ville  +  enfants  +
+++++++
+  1  +  Patrick  +  Martin  +  1965/10/08  +  Bordeaux  +  2  +
+++++++
+  2  +  Julien  +  Lebreton  +  1964/02/21  +  Paris  +  2  +
+++++++
+  3  +  Marc  +  Richard  +  1958/04/15  +  Lille  +  4  +
+++++++
+  4  +  Francis  +  Perrin  +  1982/12/05  +  Paris  +  0  +
+++++++
+  5  +  Daniel  +  Bacon  +  1974/07/13  +  Reims  +  1  +
+++++++

```

Ceci est une représentation en mode console et non pas sous phpMyAdmin

Admettons que nous voulions afficher uniquement les personnes qui n'ont que 2 enfants, la requête SQL sera :

```
SELECT * FROM clients_tbl WHERE enfants='2'
```

Soit en français :

SELECT	Je Sélectionne
*	Tous les champs
FROM clients_tbl	Depuis la table client_tbl
WHERE enfants='2'	Quand le champs enfants est égal à 2

Ce qui donne comme résultat :

```

+++++++
+  1  +  Patrick  +  Martin  +  1965/10/08  +  Bordeaux  +  2  +
+++++++
+  2  +  Julien  +  Lebreton  +  1964/02/21  +  Paris  +  2  +
+++++++

```

Voici la même requête, mais cette fois nous n'allons demander l'affichage que des noms et prénoms :

```
SELECT nom,prenom FROM clients_tbl WHERE enfants='2'
```

Soit en français :

SELECT	Je sélectionne
nom, prenom	Les champs nom et prenom
FROM clients_tbl	Depuis la table client_tbl
WHERE enfants='2'	Quand le champs enfants est égal à 2

Ce qui donne comme résultat :

```

+++++++
+  Patrick  +  Martin  +
+++++++
+  Julien  +  Lebreton  +
+++++++

```

Reprenons la première requête et ajoutons d'autres conditions (**WHERE**), à savoir que nous allons demander l'affichage des personnes qui ont 1 ou 2 enfants et qui habitent la ville de Paris :

```
SELECT * FROM clients_tbl WHERE enfants='1' OR enfants='2' AND ville='Paris'
```

- Vous devez faire la différence entre **OR** et **AND**, nous pouvons l'analyser ainsi :

Si le nombre d'enfant est **1** et que la ville est **Paris** , c'est **OK**.

Et si le nombre d'enfants est **2** et que la ville est **Paris** , c'est **OK**.

Il faut que les 2 conditions soient respectées pour que l'enregistrement soit affiché !

Voici les opérateurs possibles :

+	Addition
-	Soustraction
*	Multiplication
/	Division
<	Plus petit que
<=	Plus petit ou égal à
=	Égal à
!= ou <>	N'est pas égal à
>=	Plus grand ou égal à
and	ET
or	OU
not	Négation

Pour finir voici quelques autres exemples de SELECT :

```
SELECT * FROM clients_tbl WHERE ne_le < "1978-01-01"
```

Sélection des personnes ayant une date de naissance plus petite que **1978/01/01**.

```
SELECT * FROM clients_tbl WHERE enfants != '0'
```

Sélection des personnes dont le nombre des enfants est différent de " **0** ".

```
SELECT * FROM clients_tbl WHERE nom LIKE 'le%'
```

Sélection des personnes dont le nom commence par " **le** ".

```
SELECT * FROM clients_tbl WHERE nom LIKE '%ri%'
```

Sélection des personnes qui ont dans leur nom la syllabe " **ri** " .

Il serait bien trop long d'énumérer tous les exemples possibles, mais n'hésitez pas à consulter **[la documentation en Français de Nexen.net sur MySQL](#)**.

www.phpdebutant.org © 2006 – L'équipe de phpDebutant