

## Les bases de données : SQL DATABASE

### 1.] SQL SELECT

#### *Commande basique*

L'utilisation basique de cette commande s'effectue de la manière suivante :

```
SELECT nom_du_champ  
FROM nom_du_tableau
```

Si l'on veut avoir la liste de toutes les villes des clients, il suffit d'effectuer la requête suivante :

```
SELECT ville  
FROM client
```

#### *Obtenir plusieurs colonnes*

```
SELECT prenom, nom  
FROM client
```

#### *Obtenir toutes les colonnes d'un tableau*

```
SELECT * FROM client
```

Une requête SELECT peut devenir assez longue. Juste à titre informatif, voici une requête SELECT qui possède presque toutes les commandes possibles :

```
SELECT *  
FROM table  
WHERE condition  
GROUP BY expression  
HAVING condition  
{ UNION | INTERSECT | EXCEPT }  
ORDER BY expression  
LIMIT count  
OFFSET start
```

## Les bases de données : SQL DATABASE

La commande WHERE s'utilise en complément à une requête utilisant SELECT. La façon la plus simple de l'utiliser est la suivante :

```
SELECT nom_colonnes
FROM nom_table
WHERE condition
```

### SELECT AVEC JOINTURES SUR PLUSIEURS TABLES :

#### Syntaxe

```
SELECT *
FROM table1
INNER JOIN table2 ON table1.id = table2.fk_id
```

Ou encore son équivalent :

```
SELECT *
FROM table1, table2
where table1.id = table2.fk_id
```

#### Exemple

Imaginons une application qui possède une table utilisateur ainsi qu'une table commande qui contient toutes les commandes effectuées par les utilisateurs.

#### Table utilisateur :

id	prenom	nom	email	ville
1	Aimée	Marechal	aime.marechal@example.com	Paris
2	Esmée	Lefort	esmee.lefort@example.com	Lyon
3	Marine	Prevost	m.prevost@example.com	Lille
4	Luc	Rolland	lucrolland@example.com	Marseille

## Les bases de données : SQL DATABASE

Table commande :

utilisateur_id	date_achat	num_facture	prix_total
1	2013-01-23	A00103	203.14
1	2013-02-14	A00104	124.00
2	2013-02-17	A00105	149.45
2	2013-02-21	A00106	235.35
5	2013-03-02	A00107	47.58

Pour afficher toutes les commandes associées aux utilisateurs, il est possible d'utiliser la requête suivante :

```
SELECT id, prenom, nom, date_achat, num_facture, prix_total
FROM utilisateur
INNER JOIN commande ON utilisateur.id = commande.utilisateur_id
```

Ou encore son équivalent :

```
SELECT id, prenom, nom, date_achat, num_facture, prix_total
FROM utilisateur, commande
where utilisateur.id = commande.utilisateur_id
```

## Les bases de données : SQL DATABASE

### SQL INSERT INTO

**Insérer une ligne en spécifiant seulement les colonnes souhaitées**

Cette deuxième solution est très similaire, excepté qu'il faut indiquer le nom des colonnes avant « VALUES ». La syntaxe est la suivante :

```
INSERT INTO table
(nom_colonne_1, nom_colonne_2, ...
VALUES ('valeur 1', 'valeur 2', ...)
```

***Insertion de plusieurs lignes à la fois***

```
INSERT INTO client (prenom, nom, ville, age)
VALUES
('Rébecca', 'Armand', 'Saint-Didier-des-Bois', 24),
('Aimée', 'Hebert', 'Marigny-le-Châtel', 36),
('Marielle', 'Ribeiro', 'Maillères', 27),
('Hilaire', 'Savary', 'Conie-Molitard', 58);
```

**Note : on ne précise pas, ne gère pas la clé primaire qui est en auto incrément.**

## Les bases de données : SQL DATABASE

### SQL UPDATE

La commande UPDATE permet d'effectuer des modifications sur des lignes existantes. Très souvent cette commande est utilisée avec WHERE pour spécifier sur quelles lignes doivent porter la ou les modifications.

#### *Syntaxe*

La syntaxe basique d'une requête utilisant UPDATE est la suivante :

```
UPDATE table  
SET nom_colonne_1 = 'nouvelle valeur'  
WHERE condition
```

```
UPDATE table  
SET colonne_1 = 'valeur 1', colonne_2 = 'valeur 2', colonne_3 = 'valeur 3'  
WHERE condition
```

Pour modifier l'adresse du client Pierre, il est possible d'utiliser la requête suivante :

```
UPDATE client  
SET rue = '49 Rue Ameline',  
    ville = 'Saint-Eustache-la-Forêt',  
    code_postal = '76210'  
WHERE id = 2
```

## Les bases de données : SQL DATABASE

### SQL DELETE

#### *Syntaxe*

La syntaxe pour supprimer des lignes est la suivante :

```
DELETE FROM table  
WHERE condition
```

**Attention** : s'il n'y a pas de condition WHERE alors toutes les lignes seront supprimées et la table sera alors vide.

Si l'ont souhaite supprimer les utilisateurs qui se sont inscrit avant le « 10/04/2012 », il va falloir effectuer la requête suivante :

```
DELETE FROM utilisateur  
WHERE date_inscription < '2012-04-10'
```