

Le SQL selon MySQL

MySQL supporte sa propre « vision » du SQL. Le meilleur point de départ pour utiliser efficacement le SQL selon MySQL est son manuel :

<http://www.mysql.com/doc/fr/Reference.html>

Manipulation de données :

La structure des données :

CREATE, ALTER, DROP

On peut modifier la structure des données avec ces mots-clés :

Create database XX

création de la base de données XX

(il n'y a pas de *Alter* pour les base de données)

Drop Database XX

- Supprime la base de données XX. Attention, on supprime, pas de retour arrière!

Même chose pour les tables :

Create table XX ([définition des champs]);

Alter table XX

Ces fonctions comprennent **beaucoup** d'options, particulièrement les *ALTER*. Pour voir les options disponibles, consultez le manuel :

http://www.mysql.com/doc/fr/Data_Definition.html

Type de données

Lors de la création (et la modification) de tables, il faut choisir le type de données appropriés.
Pour voir les types de données de MySQL, voir le manuel :

http://www.mysql.com/doc/fr/Column_types.html

En gros, MySQL offre des types de données semblables aux autres SGBDR :

Type numériques : INT, FLOAT, DECIMAL

Type de chaînes : CHAR, VARCHAR, TEXT,

Type de date et d'heures : DATE, TIME, DATETIME, TIMESTAMP

Type pour « choix » : ENUM, SET

Ajouter des données - INSERT

```
INSERT INTO nomdetable (liste,de,champs)
VALUES ("valeurs", "pour", "leschamps");
```

La liste de champs est optionnelle si on utilise tous les champs dans la liste de *VALUES* :

```
INSERT INTO table VALUES (2, "Benoit", "Caron");
```

On peut également insérer des données qu'on recueille dans une autre table :

```
INSERT INTO table (id, nom, prenom) SELECT id, nom, prenom FROM autretable WHERE
nom LIKE 'c%';
```

Modifier les données - UPDATE

Mise à jour de données dans la table. Attention de bien spécifier quels champs on veut mettre à jour, sinon on modifie *tous* les champs de la tables!

Cette requête va mettre la valeur « nouvelle valeur » dans le champs « nomdetable » de *tous*

les enregistrement de la table :

```
UPDATE nomdetable SET champ1 = 'nouvellevaleur' ;
```

Évidemment, on veut généralement mettre à jour un seul enregistrement :

```
UPDATE nomdetable SET champ1='unevaleur', champ2 = 'autrechose' WHERE id = 12;
```

Supprimer des données - DELETE

Pour supprimer les enregistrements, on suit la même logique que UPDATE :

Suppression de *tous* les enregistrements :

```
DELETE FROM nomdetable;
```

-- Suppression de l'enregistrement qui a le no. id 12

```
DELETE FROM nomdetable WHERE id = 12;
```

-- Suppression de tous les « mathieu » de la table :

```
DELETE FROM etudiants WHERE prenom = 'Mathieu';
```

Rechercher des données - SELECT

Les *SELECT* sont le « coeur » du travail avec une base de données relationnelle. C'est avec les SELECT qu'on interroge la base de données et son contenu.

Entrées à consulter dans le manuel sur les SELECT :

3.3.4 - Récupérer des informations à partir d'une table

http://www.mysql.com/doc/fr/Retrieving_data.html

Donne un survol de l'utilisation des SELECT

6.4.1 – Syntaxe de SELECT

<http://www.mysql.com/doc/fr/SELECT.html>

La référence sur le SELECT : tous les détails!

Fonctions intégrés

MySQL offre un ensemble de fonctions intégrés pour transformer, calculer ou manipuler les données. La section pertinente du manuel est :

6.3 Fonctions à utiliser dans les clauses SELECT et WHERE

<http://www.mysql.com/doc/fr/Functions.html>

Vous référer à cette documentation.

Joindre des tables : JOIN, LEFT OUTER JOIN et cie.

Join direct :

```
SELECT c.titre AS album, g.nom as groupe
FROM cd AS c, groupe AS g
WHERE
c.cd_id = g.cd;
```

Retourne tous les titres d'albums et le nom du groupe qui l'a produit.

Un LEFT OUTER JOIN permet de retourner les enregistrements oUU une partie de l'information est manquante :

```
SELECT c.titre AS album, g.nom as groupe
FROM cd
LEFT JOIN groupe AS ON cd.cd_id = groupe.cd;
```

Assembler plusieurs requêtes : UNION

<http://www.mysql.com/doc/fr/UNION.html>

Permet de « coller » dans le résultat les enregistrements obtenus de 2 requêtes différentes :

MySQL 4 seulement. Voir la documentation pour des exemples.