

Aide-mémoire MySQL

Préambule

Ce document est un aide-mémoire dédié aux commandes de base dans MySQL. Son objectif est de faciliter l'accès à la syntaxe de MySQL en particulier lors des TP. Les différents exemples présentés dans ce document font référence à la base de données « Commandes » vue en cours et en TD. Cet aide-mémoire ne constitue en rien un support de cours.

1. Principaux types de données en MySQL

| Nom du type de données | Description | Taille mémoire |
|----------------------------|--|----------------------------------|
| Les champs numériques | | |
| INT | Entier standard compris entre -2 147 483 648 et 2 147 483 647. Si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 4 294 967 295. | 4 octets |
| FLOAT | Décimal de simple précision. Il est possible d'initialiser un flottant avec : FLOAT (x, y) où x est la valeur entière et y la décimale, la taille mémoire occupée est alors proportionnelle à la longueur de x et de y. | 4 octets |
| DOUBLE, REAL | Décimal de double précision. L'initialisation peut également se faire en précisant les parties entière et décimale. | 8 octets |
| Les chaînes de caractères | | |
| CHAR (n) | Chaîne de n caractères de taille fixe. | Maximum 255 caractères |
| VARCHAR (n) | Chaîne de caractères de taille inférieure ou égale à n. La taille de la chaîne est comprise entre 1 et 255. | Maximum 255 caractères |
| BLOB, TEXT | Zone de texte pouvant contenir une chaîne de longueur maximale de 65535 caractères, TEXT aura un contenu de type ASCII (casse insensible) et BLOB aura un contenu de type binaire (casse sensible). | Maximum 65 535 caractères |
| Les énumérations | | |
| ENUM ('val1', 'val2', ...) | Une seule valeur parmi plusieurs objets textes : 'val1','val2', ... | Maximum 65535 objets différente. |
| Les dates | | |
| DATE | Une date dans l'intervalle '1000-01-01' à '9999-12-31'. MySQL affiche les valeurs de type DATE au format 'YYYY-MM-DD', mais permet d'assigner des valeurs DATE en utilisant plusieurs formats de chaînes et nombres. | 3 octets |
| DATETIME | Une combinaison de date et heure dans l'intervalle de validité '1000-01-01 00:00:00' à '9999-12-31 23:59:59'. MySQL affiche les valeurs de type DATE au format 'YYYY-MM-DD HH:MM:SS', mais permet d'assigner des valeurs DATE en utilisant plusieurs formats de chaînes et nombres. | 8 octets |

2. Création, remplissage et mise à jour de tables

Création de la table PRODUIT (avec les attributs vus en cours) :

```
CREATE TABLE PRODUIT (  
    RefP      VARCHAR(5)    NOT NULL,  
    TypeP     VARCHAR(20)   NOT NULL,  
    Prix      FLOAT (6, 2)  NOT NULL,  
    QStock    INT,  
    PRIMARY KEY (RefP)  
) ;
```

NB : L'attribut QStock n'était pas présent dans la table PRODUIT présentée en cours. Nous l'avons ajouté dans cet aide-mémoire afin de mettre en évidence certains éléments de syntaxe de MySQL.

Suppression de la table PRODUIT :

```
DROP TABLE IF EXISTS PRODUIT ;
```

Renommage de la table PRODUIT en PRODUIT QUINCAILLERIE :

```
RENAME TABLE PRODUIT TO PRODUIT_QUINCAILLERIE ;
```

Insertion d'enregistrements dans la table PRODUIT :

```
INSERT INTO PRODUIT (RefP, TypeP, Prix, QStock) VALUES  
    ("A262", "Cheville", 75, 45),  
    ("C236", "Ecrou", 40.5, 45) ;  
INSERT INTO PRODUIT (RefP, TypeP, Prix)  
    VALUES ("B60", "Clou", 95) ;
```

La valeur de l'attribut QStock est nulle dans le second exemple comme le permet le schéma de la table.

Supposons la table PRODUIT_BIS de même schéma que la table PRODUIT. On peut alors remplir cette première table à partir de la seconde (dans cet exemple, PRODUIT_BIS va contenir les enregistrements de PRODUIT ne correspondant pas aux clous et dont le prix est inférieur à 90 euros) :

```
INSERT INTO PRODUIT_BIS  
    SELECT      *  
    FROM        PRODUIT  
    WHERE       TypeP <> "Clou" AND Prix < 90 ;
```

Remplissage de la table PRODUIT à l'aide d'un fichier contenant des enregistrements :

```
LOAD DATA LOCAL INFILE "/chemin/produits.txt"  
    INTO TABLE PRODUIT ;
```

Chaque ligne du fichier « /chemin/produits.txt » contient un enregistrement à insérer dans la table, les valeurs des enregistrements doivent être séparées par des tabulations (séparateur de valeurs par défaut pouvant être redéfini).

Suppression dans la table PRODUIT des enregistrements correspondant à des planches :

```
DELETE FROM PRODUIT
WHERE TypeP = "Planche" ;
```

Mise à jour du stock des enregistrements correspondant aux clous dans la table PRODUIT :

```
UPDATE    PRODUIT
SET       QStock = 0
WHERE     TypeP = "Clou" ;
```

3. Recherche d'information dans la base de données

3.1 Requêtes simples

Sélection simple de la référence et du prix des produits correspondant à des clous :

```
SELECT    RefP, Prix
FROM      PRODUIT
WHERE     TypeP = "Clou" ;
```

Sélection des types et des prix des produits dont la quantité en stock n'est pas nulle :

NB : nous voulons que l'affichage du résultat se fasse selon un classement alphabétique des types de produits puis un classement décroissant des prix.

```
SELECT    TypeP, Prix
FROM      PRODUIT
WHERE     QStock IS NOT NULL
ORDER BY  TypeP ASC, Prix DESC ;
```

Sélection sans doublons des types de produits qui ne sont ni des clous, ni des écrous et dont le nom ne contient pas le caractère « i » en seconde position :

```
SELECT DISTINCT TypeP
FROM            PRODUIT
WHERE           TypeP NOT IN ("Clou", "Ecrou")
               AND TypeP NOT LIKE "_i%" ;
```

Sélection des types des produits faisant l'objet d'au moins deux références distinctes et dont le prix est compris entre 2 et 10 euros :

```
SELECT    TypeP, COUNT(*) AS Nombre_de_Ref
FROM      PRODUIT
WHERE     Prix BETWEEN 2 AND 10
GROUP BY  RefP
HAVING    Nombre_de_Ref >= 2 ;
```

Sélection ordonnée des références des commandes qui comportent des clous (on réalise une jointure naturelle entre la table PRODUIT et la table DETAIL) :

```
SELECT      DISTINCT D.RefCom
FROM        PRODUIT P, DETAIL D
WHERE       P.RefP = D.RefP
            AND P.TypeP = "Clou"
ORDER BY    D.RefCom ;
```

3.2 Requêtes imbriquées

Sélection des références et des prix de tous les produits de type « vis » qui sont moins chers que le produit de référence « BC1 » :

```
SELECT      RefP, Prix
FROM        PRODUIT
WHERE       TypeP = "Vis"
            AND Prix <
            ( SELECT Prix
              FROM    PRODUIT
              WHERE   RefP = "BC1"
            ) ;
```

Sélection des références des produits n'ayant jamais été commandés :

Première façon :

```
SELECT      RefP
FROM        PRODUIT
WHERE       RefP NOT IN
            ( SELECT RefP
              FROM    DETAIL
            ) ;
```

Seconde façon :

```
SELECT      P.RefP
FROM        PRODUIT P
WHERE       NOT EXISTS
            ( SELECT *
              FROM    DETAIL D
              WHERE   D.RepP = P.RefP
            ) ;
```

Sélection des informations sur les produits de prix maximum :

```
SELECT      *
FROM        PRODUIT
WHERE       Prix >= ALL
            ( SELECT Prix
              FROM    PRODUIT
            ) ;
```