

SI3: Exploitation des données

CHAP III - Introduction au SQL

Initiation au langage de description des données.





Sommaire

1 Introduction	3
1.1 Le langage de description des données	3
1.2 Présentation du thème.....	4
2 Les types de données.....	5
2.1 Les types numériques	5
2.1.1 Types numériques exacts	5
2.1.2 Types numériques non exacts	5
2.2 Les types chaînes de caractères	6
2.3 Les types temporels	6
3 Création d'une base de données	7
3.1 Création de la base de données	7
3.2 Création des tables.....	7
4 Suppression d'une base de données (et/ou) de table.....	9
5 Modification des tables	9
6 Application.....	11
6.1 Modification de la base « Bibliothèque »	11
6.2 Création de la base « Vols »	11
6.3 Modification de la base « Vols »	11



1 Introduction

1.1 Le langage de description des données

Dans le précédent chapitre, nous avons étudié les commandes SQL qui permettent d'interroger et de mettre à jour une base de données. Ces commandes appartiennent au langage de manipulation de données (LMD).





Dans ce chapitre, nous étudierons le langage de description de données (**LDD**). Ce langage représente l'ensemble des instructions SQL qui permettent la **création** et la **modification** de la structure des **bases de données**.

La conception d'une base de données, comme tout objet technique, nécessite une phase de conception. Concernant les données, cette étape consiste à modéliser à l'aide de schémas¹ le système d'information étudié.

De ce schéma on déduit le schéma relationnel que nous avons abordé dans les précédents chapitres.

Le schéma relationnel est « universel » et ne dépend pas du SGBDR utilisé. C'est un peu le même principe concernant les algorithmes et leur mise en œuvre dans tel ou tel langage.

Le LDD est donc spécifique à un SGBDR. La plupart des instructions se retrouvent mais leur syntaxe diffère parfois.

Base de données		Programmation	
Schéma relationnel :		Algorithme :	
CLIENT(NumCli, NomCli) NumCli : Entier, Numéro séquentiel NomCli : Chaîne(30)		Pour i = 1 à 10 Afficher i FinPour	
			
Create Table Client (NumCli Int Not Null Auto_Increment Primary Key, NomCli VarChar(30) Not Null);	Create Table Client (NumCli Integer Not Null Identity, NomCli VarChar(30) Not Null, Primary key (NumCli));	for i in range(1, 10): print i	for (\$i = 1 ; \$i <=10 ; \$i++) { echo \$i ; }

Dans ce cours nous aborderons la syntaxe de MySQL².

¹ Le schéma le plus répandu, que vous étudierez par la suite, est le Modèle Conceptuel des Données (MCD). Ce schéma conceptuel est proposé par la méthode Merise.

² Le manuel de référence de MySQL est disponible en ligne sur <http://dev.mysql.com/doc/>. Certains passages du cours sont extraits de ce manuel.

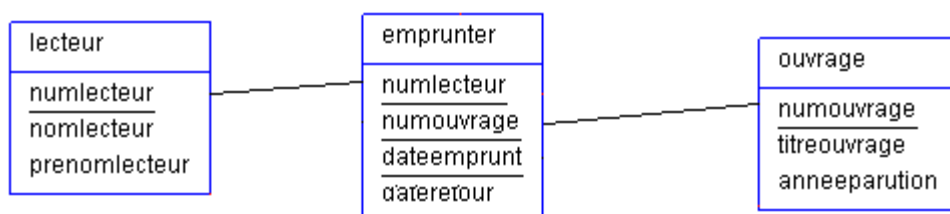


1.2 Présentation du thème

Pour illustrer les commandes de base du langage de description des données, nous utiliserons la base de données bibliothèque du chapitre II.

Voici les différentes représentations possibles de son schéma relationnel³ :

a) Représentation graphique



b) Représentation en intention

LECTEUR(NumLecteur, NomLecteur, PrenomLecteur)
OUVRAGE(NumOuvrage, TitreOuvrage, AnneeParution)
EMPRUNTER(#NumLecteur, #NumOuvrage, DateEmprunt, DateRetour)

c) Représentation en intention détaillée

LECTEUR(NumLecteur, NomLecteur, PrenomLecteur)
NumLecteur : Clé primaire

OUVRAGE(NumOuvrage, TitreOuvrage, AnneeParution)
NumOuvrage : Clé primaire

EMPRUNTER(NumLecteur, NumOuvrage, DateEmprunt, DateRetour)
NumLecteur, NumOuvrage, DateEmprunt : Clé primaire
NumLecteur : Clé étrangère en référence à NumLecteur de LECTEUR
NumOuvrage : Clé étrangère en référence à NumOuvrage de OUVRAGE

Comme vous le constaterez par la suite, la représentation détaillée est très proche de ce que l'on retrouvera pour créer une table en SQL. Elle permet également de mettre en avant sans ambiguïté des liens entre tables à partir de champs ne portant pas le même nom.

C'est pourquoi, il est recommandé d'utiliser cette notation.

³ Ce schéma relationnel sera optimisé à la fin du cours.

2 Les types de données

En programmation, lorsque l'on déclare une variable, la plupart des langages nécessitent que l'on précise son type (entier, chaîne de caractères...) On parle alors de typage explicite. En SQL, le type des colonnes doit obligatoirement être renseigné.

2.1 Les types numériques

2.1.1 Types numériques exacts

a) Les types « Entier »

Le tableau ci-dessous présente les intervalles de valeurs en fonction des types entiers proposés par MySQL :

Type	Octets	De	A
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT (ou INTEGER)	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807

Par défaut, MySQL gère les nombres négatifs. Il est possible de préciser à l'aide du mot clé **UNSIGNED** (non signé) que les nombres seront exclusivement positifs (ou nuls).

Exemple : Quel est l'intervalle de valeurs du champ *QteCmde* déclaré comme ci-dessous :

QteCmde TINYINT UNSIGNED → *QteCmde* ∈ [. . . . ,]

b) Le type « décimal »

Le type dédié aux nombres décimaux est : **DECIMAL** (ou encore **NUMERIC**)

Lors de la déclaration, on précisera le nombre total de chiffres nécessaires suivi du nombre de chiffres après la virgule. Lorsqu'une valeur ayant trop de décimales est affectée à une colonne, la valeur est arrondie.

Exemple : Quelles sont les valeurs possibles pour un champ *PrixHT* déclaré comme ci-dessous :

PrixHT DECIMAL(5,2) □ 3.14159 □ - 99.99 □ 1212 □ 999.99

2.1.2 Types numériques non exacts

Il s'agit ici de stocker des données numériques approchées appelées nombres à virgule flottante.

Type	Octets	De	A
FLOAT	8	-3.402823466E+38	-1.175494351E-38
		1.175494351E-38	3.402823466E+38
DOUBLE (ou REAL)	8	-1.7976931348623157E+308	-2.2250738585072014E-308
		2.2250738585072014E-308	1.7976931348623157E+308

Le nombre de décimales peut être précisé. Consultez le manuel de référence en cas de besoin.

2.2 Les types chaînes de caractères

Certainement le type de données le plus utilisé en informatique de gestion, il correspond au type *String* de nombreux langages de programmation.

Voici les principaux types utilisés :

Type	Octets
CHAR (M)	M octets, $1 \leq M \leq 255$
VARCHAR (M)	L+1 octets, avec $L \leq M$ et $1 \leq M \leq 255$
BLOB, TEXT	L+2 octets, avec $L < 2^{16}$ (65536)

Contrairement au type TEXT, le type BLOB est sensible à la casse.

Dans ce tableau, *M* représente la longueur maximale de la chaîne et *L* sa longueur effective. Ainsi une colonne déclarée en `VARCHAR(30)` nécessitera au maximum 31 octets. Sauriez-vous expliquer pourquoi ?

Une donnée déclarée `CHAR(30)` sera complétée par des espaces si $L < 30$. Ce type est utilisé généralement pour stocker des valeurs dont la longueur est toujours fixe.

2.3 Les types temporels

Ils permettent de stocker les dates et les heures. De nombreuses fonctions sont dédiées à ces types de données. (Exemple : Extraction du mois dans une date, ajout d'un nombre de jours, différence entre dates, etc.)

Type	Octets	Format	De	A
DATE	3	AAAA-MM-JJ	1000-01-01	9999-12-31
DATETIME	8	AAAA-MM-JJ HH:MM:SS	1000-01-01 00:00:00	9999-12-31 23:59:59
TIMESTAMP	4	AAAA-MM-JJ HH:MM:SS	1970-01-01 00:00:00	Une date en 2037...
TIME	3	HH:MM:SS	-838:59:59	838:59:59

Une colonne `TIMESTAMP` est utile pour enregistrer les dates et heures lors des opérations `INSERT` et `UPDATE`, car elle prend automatiquement la date actuelle (date du système) si vous ne lui assignez pas de valeur.



3 Création d'une base de données

Les tables sont associées à une base de données. Pour pouvoir créer une table il faut tout d'abord créer la base de données...

3.1 Création de la base de données

Une base de données est créée à l'aide de la commande SQL suivante :

```
CREATE DATABASE [IF NOT EXISTS] db_name
[create_specification [, create_specification] ...]
```

Les options `create_specification` peuvent être données pour spécifier des caractéristiques de la base. Cela concerne essentiellement le jeu de caractères utilisé.

Exemple : `CREATE DATABASE Bibliotheque;`

Les bases de données MySQL sont implémentées comme des répertoires contenant des fichiers qui correspondent aux tables dans les bases de données. Dans le cas d'une installation par défaut de WampServer, les bases sont stockées dans le fichier `C:\wamp\bin\mysql\mysqlX.X.X\ibdata1` et leurs structures dans le dossier `C:\wamp\bin\mysql\mysqlX.X.X\data\` (avec InnoDB)

3.2 Création des tables

La création d'une table est réalisée à l'aide de l'instruction `CREATE TABLE`.

Les informations minimales à préciser sont :

- Le **nom de la table**
- Les **champs** (colonnes) de la table
 - o Pour chaque champ, on précisera son **type** et si la valeur *Null* est acceptée.
- Le ou les champ(s) de la **clé primaire**
- Les colonnes **clés étrangères**
 - o Pour chaque clé on précisera à quel autre champ et table elle est reliée.
 - o Pour chaque clé on précisera éventuellement les actions à réaliser lors de mise à jour ou suppression pour respecter l'intégrité référentielle.

Voici un extrait de la syntaxe MySQL :

```
CREATE TABLE tbl_name
[(create_definition,...)]
[table_options]
```

Nom de la table

Clé primaire et clé(s) étrangères

create_definition:

```
column_definition
| [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)
| [CONSTRAINT [symbol]] FOREIGN KEY
  [index_name] (index_col_name,...) [reference_definition]
```

column_definition:

```
col_name type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT] [[PRIMARY] KEY] [COMMENT 'string']
[reference_definition]
```

Définition d'une colonne.

reference_definition:

```
REFERENCES tbl_name [(index_col_name,...)]
[ON DELETE CASCADE | SET NULL | SET DEFAULT]
[ON UPDATE CASCADE | SET NULL | SET DEFAULT]
```

table_option:

```
{ENGINE|TYPE} = {BDB|HEAP|ISAM|InnoDB|MERGE|MRG_MYISAM|MYISAM}
| AUTO_INCREMENT = value
```



Exemple pour la table Lecteur :

```
CREATE TABLE Lecteur(  
    NumLecteur      INT NOT NULL,  
    NomLecteur      VARCHAR(30) NOT NULL,  
    PrenomLecteur   VARCHAR(30),  
    CONSTRAINT lecteur_pk PRIMARY KEY (NumLecteur)  
);
```

Pour la table *Ouvrage*, on ne souhaite pas gérer *NumOuvrage* (un numéro automatique devra être généré). Donnez la commande SQL de création de la table en utilisant l'autre syntaxe pour préciser la clé primaire :

Exemple pour la table Emprunter :

```
CREATE TABLE Emprunter(  
    NumLecteur      INT NOT NULL,  
    NumOuvrage      INT NOT NULL,  
    DateEmprunt     DATE NOT NULL,  
    DateRetour      DATE,  
    CONSTRAINT emprunter_pk PRIMARY KEY (NumLecteur, NumOuvrage, DateEmprunt),  
    CONSTRAINT emprunter_numlecteur_fk FOREIGN KEY (NumLecteur)  
        REFERENCES Lecteur(NumLecteur),  
    CONSTRAINT emprunter_numouvrage_fk FOREIGN KEY (NumOuvrage)  
        REFERENCES Ouvrage(NumOuvrage)  
);
```

A retenir :

- Le nommage des contraintes devra respecter les conventions suivantes :
 - o nomdetable_champcléprimaire_pk
 - o nomdetable_champcléétrangère_fk
- MySQL ne conserve pas le nom donnée à une contrainte de clé primaire mais la syntaxe est acceptée par soucis de compatibilité avec les autres SGBDR.
- Afin de respecter l'intégrité référentielle, les tables contenant exclusivement des clés primaires doivent être créées en premier.
- Pour obtenir l'instruction à l'origine de la création d'une table MySQL, utiliser la commande :
SHOW CREATE TABLE tbl_name



4 Suppression d'une base de données (et/ou) de table

La suppression d'une base de données est réalisée à l'aide de la commande :

```
DROP DATABASE [IF EXISTS] db_name
```

La suppression d'une table est réalisée comme ci-dessous :

```
DROP TABLE [IF EXISTS]
tbl_name [, tbl_name] ...
```

L'option `IF EXISTS` évite l'affichage d'une erreur si l'élément n'existe pas.

La suppression d'une table liée par une contrainte d'intégrité référentielle ne sera pas possible et ce même si les tables sont vides.

Dans le cas de la base bibliothèque, seule la table pourra être effacée sans conditions préalables.

5 Modification des tables

La modification d'une table consiste ici à modifier son nom, sa structure ou bien ses champs.

Les opérations consisteront à :

- Ajouter ou supprimer un champ
- Modifier le nom, le type ou les propriétés d'un champ
- Supprimer ou modifier une clé primaire
- Supprimer ou modifier une contrainte d'intégrité référentielle

La syntaxe de la commande est (extraits) :

```
ALTER TABLE tbl_name
alter_specification [, alter_specification] ...
```

alter_specification:

```
ADD [COLUMN] column_definition [FIRST | AFTER col_name ]
| ADD [COLUMN] (column_definition,...)
| ADD [CONSTRAINT [symbol]]
    PRIMARY KEY [index_type] (index_col_name,...)
| ADD [CONSTRAINT [symbol]]
    FOREIGN KEY [index_name] (index_col_name,...)
    [reference_definition]
| CHANGE [COLUMN] old_col_name column_definition
    [FIRST|AFTER col_name]
| MODIFY [COLUMN] column_definition [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP FOREIGN KEY fk_symbol
| RENAME [TO] new_tbl_name
```

Exemples :

Donner les instructions SQL pour réaliser les traitements qui suivent. On considérera ici que toutes les tables sont vides.

a) Renommer la table « emprunter » en « emprunt ».

b) Ajouter un champ *TelLecteur* (taille 10) à la table des lecteurs.



- c) Ajouter un champ *RefLecteur* (taille fixe de 8 caractères) juste avant *NumLecteur* (ce champ deviendra clé primaire par la suite)
- d) Modifier la taille de la colonne *NomLecteur* pour la passer à 40 caractères.
- e) Supprimer le champ *TelLecteur*
- f) Indiquer les opérations nécessaires pour supprimer la clé primaire sur *NumLecteur* et définir *RefLecteur* comme nouvelle clé primaire. Le champ *NumLecteur* de la table « emprunt » conservera son nom mais sera lié à *RefLecteur* de Lecteur.



6 Application

6.1 Modification de la base « Bibliothèque »

La présence du champ *NumLecteur* dans la clé primaire de la table *Emprunt* n'est pas nécessaire. Un emprunt peut en effet être identifié à partir d'un numéro d'ouvrage et de sa date d'emprunt. Ainsi la relation *Emprunt* devient :

EMPRUNT(NumLecteur, NumOuvrage, DateEmprunt, DateRetour)
 NumOuvrage, DateEmprunt : Clé primaire
 NumLecteur : Clé étrangère en référence à RefLecteur de LECTEUR
 NumOuvrage : Clé étrangère en référence à NumOuvrage de OUVRAGE

- Donner les requêtes SQL nécessaires pour appliquer ce changement sur la table *Emprunt*.

Remarque :

Afin de gérer le cas (plutôt rare !) d'un emprunt concernant le même ouvrage un même jour, le format du champ *DateEmprunt* devra intégrer la date et l'heure (AAAA-MM-JJ HH:MM:SS)

6.2 Création de la base « Vols »

Donner le script complet de création de la base de données « Vols » abordée dans le chapitre II.

Pour réaliser ce travail, vous considérerez les contraintes ci-dessous :

- Les valeurs numériques sont de type Entier.
- Les chaînes de caractères sont toutes de taille maximale égale à 100 par défaut.
- Les contraintes d'intégrité référentielle seront précisées après la création de toutes les tables.

PILOTE : Table					AVION : Table		
NUMPILOTE	NOMPILOTE	ADRESSE	SALAIRE	BONUS	NUMAVION	NOMAVION	CAPACITE
1	TOTO	97438 Sainte Marie	1000	1100	1	A340-300	295
2	PAYET	97487 Saint Denis	3000	800	2	ATR72	70
3	HOAREAU	97420 Le Port	3300	950	3	B747-400	420
4	DUPUIS	75130 Paris	3200	900	4	A319	120
5	DUPOND	75150 Paris	3500	1000	5	ATR42/72	50
6	PETIT	34000 Montpellier	2000	500	6	B747-400	420
7	VITRI	34070 Montpellier	2500				
* (NuméroAuto)					* NuméroAuto)		
Enr: 1 sur 7					Enr: 1 sur 6		

VOL : Table						
NUMVOL	NUMAVION	NUMPILOTE	VILLEDEPART	VILLEARRIVEE	HEUREDEPART	HEUREARRIVEE
AF 380	3	1	PARIS ORLY	GILLOT	21:00:00	10:45:00
AF 7684	4	6	PARIS CDG	MONTPELLIER	20:45:00	22:10:00
MK 203	5	4	GILLOT	MAURICE	07:30:00	08:45:00
MK 230	2	3	GILLOT	MAURICE	11:40:00	12:25:00
MK 40	6	2	MAURICE	PARIS CDG	09:40:00	19:05:00
MK 45	1	7	PARIS CDG	GILLOT	11:40:00	12:25:00
*						
Enr: 1 sur 6						

6.3 Modification de la base « Vols »

Ces modifications sont à considérer sur une base contenant des données. Elles ne doivent donc pas entraîner de pertes de données.

- Le champ *NumVol* doit être de taille fixe (10 caractères).
- On souhaite que les champs *NumPilot* et *NumAvion* deviennent des numéros séquentiels.
- On souhaite reprendre la structure de la table des pilotes en décomposant l'adresse en 3 champs : Rue, Ville et CodePostal.
- On souhaite pouvoir réaliser une suppression en cascade pour les pilotes.