

SQL

Langage de Définition des Données

LDD

Christian Bonhomme

CNAM 2015-2016

Langage de Définition des Données : LDD

Le Langage de **D**éfinition de **D**onnées : **LDD** ou DDL en anglais, comprend les commandes suivantes :

- **CREATE** pour créer ;
- **DROP** pour supprimer ;
- **ALTER** pour modifier ;
- **RENAME** pour renommer.

Langage de Définition des Données : LDD

Ces commandes peuvent porter sur les objets suivants :

- **DATABASE** ou **SCHEMA** : la base de données ;
- **TABLE** : les tables ;
- **VIEW** : les vues ou les tables virtuelles ;
- **INDEX** : les index ;
- **SYNONYM** : les synonymes ;
- **USER** : les utilisateurs

LDD : DATABASE

Créer une base de données :

```
create database database_name;
```

Supprimer une base de données :

```
drop database database_name;
```

LDD : TABLE

Créer une table:

```
create table TABLE_NAME  
(NOM_ATTRIBUT1 type1 [contrainte1],  
  NOM_ATTRIBUT2 type2 [contrainte2],  
  ...  
  NOM_ATTRIBUTn typeN [contrainte3]);
```

NOM_ATTRIBUT : représente le nom de l'attribut défini dans le MCD.

type : détermine le type de l'attribut qui sera approprié aux données et à leur manipulation.

contrainte : définit ou non une contrainte sur l'attribut.

LDD : TABLE

Exemple :

```
create table ELEVES  
(NOM          varchar(40)  NOT NULL,  
  PRENOM      varchar(40)  NOT NULL,  
  SEXE        char(1)      NOT NULL,  
  ADRESSE     varchar(100) NOT NULL,  
  TELEPHONE   varchar(14));
```

LDD : TABLE - type de données

Quatre types de données :

- Numériques ;
- Chaînes de caractères ;
- Temporels ;
- Binaires.

LDD : TABLE - type de données numériques

Entiers :

Type	Octets	Minimum	Maximum
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT	4	-2147483648	2147483647
BIGINT	5	-9223372036854775808	9223372036854775807

On peut préciser que les attributs sont **UNSIGNED**. Dans ce cas, la longueur de l'intervalle reste la même, mais les valeurs possibles sont décalées, le minimum valant 0. Pour les **TINYINT**, on pourra par exemple aller de **0** à **255**.

Il est possible de préciser le nombre de chiffres minimum à l'affichage combiné avec **ZEROFILL** affichera des zéros à gauche.

Exemple : **INT(4) ZEROFILL**

Nombre stocké: 45

affiché : 0045

LDD : TABLE - type de données numériques

Décimaux :

DECIMAL(PRECISION, ECHELLE) :

- **PRECISION** : définit le nombre de chiffres significatifs stockés ;
- **ECHELLE** : définit le nombre de chiffres après la virgule.

FLOAT

Occupe 4 octets, stocke des nombres flottants à précision simple.
Va de $-3.402823466E+38$ à $-1.175494351E-38$ et de $1.175494351E-38$ à $3.402823466E+38$.

REAL, DOUBLE

Identiques utilisent 8 octets. Stocke des nombres flottants à double précision de $-1.7976931348623157E+308$ à $-2.2250738585072014E+308$ et $2.2250738585072014E-308$ à $1.7976931348623157E+308$.

LDD : TABLE - type de données chaînes

char et **varchar**

Pour stocker un texte relativement court, vous pouvez utiliser les types **CHAR** et **VARCHAR**. Ces deux types s'utilisent avec un paramètre qui précise la taille que peut prendre votre texte (entre 1 et 255).

La différence entre **CHAR** et **VARCHAR** est la manière dont ils sont stockés en mémoire. Un **CHAR(x)** stockera toujours **x** caractères, en remplissant si nécessaire le texte avec des espaces vides pour le compléter, tandis qu'un **VARCHAR(x)** stockera jusqu'à **x** caractères (entre 0 et x), et stockera en plus en mémoire la taille du texte stocké.

LDD : TABLE - type de données chaînes

TEXT

Le type **TEXT** est utilisé pour stocker des textes plus longs que 255 caractères.

Type	Nombre de caractères	Mémoire occupée
TINYTEXT	255	Longueur + 1 octet
TEXT	65535	Longueur + 2 octets
MEDIUMTEXT	16 777 215	Longueur + 3 octets
LONGTEXT	4 294 967 295	Longueur + 4 octets

LDD : TABLE - type de données temporels

DATE, TIME, DATETIME

- **DATE** : sert à stocker une date au format '**AAAA-MM-JJ**';
- **TIME** : sert à stocker une heure au format '**HH:MM:SS**';
- **DATETIME** : sert à stocker une date et une heure au format '**AAAA-MM-JJ HH:MM:SS**'.

AAAA : année

HH : heure

MM : mois

MM : minute

JJ : jour

SS : seconde

Exemple :

2016-03-15 09:31:08 signifie : le 15 mars 2016 à 8heures 31minutes et 8 secondes

LDD : TABLE - type de données temporels

YEAR

un seul octet sert à stocker l'année comprise entre 1901 et 2155.

TIMESTAMP

Le **TIMESTAMP** de MySQL ne sert pas à stocker un nombre de secondes, mais une date sous format numérique. Ce n'est donc pas un vrai **TIMESTAMP** (à éviter).

*Vrai **TIMESTAMP** : nombre de secondes écoulées depuis le 1er janvier 1970 à 0h0min0s.*

LDD : TABLE - type de données binaires

BIT

Stockage de bits. **BIT(*M*)** permet le stockage de *M* valeurs de bits. *M* peut varier de 1 à 64.

BOOL

Stocke **0** (false) ou **1** (true).

BLOB

type qui permet de stocker des données binaires : image, texte, fichier, ZIP...

TINYBLOB : 255 caractères maximum.

BLOB : 65535 caractères maximum.

MEDIUMBLOB : 16777215 caractères maximum.

LONGBLOB : 4 294 967 295 caractères maximum.

LDD : TABLE - contraintes

auto_increment : incrément automatique.

```
ID_ELEVE INT(11) auto_increment
```

NOT NULL : n'accepte pas les valeurs nulles..

```
ADRESSE VARCHAR(100) NOT NULL
```

NULL : accepte les valeurs nulles (peut être omis).

```
TELEPHONE VARCHAR(14) NULL
```

OU :

```
TELEPHONE VARCHAR(14)
```

UNIQUE : définit une contrainte d'unicité.

```
NUMERO_CHAMBRE NUMBER(2) UNIQUE
```

PRIMARY KEY : déclaration de clef primaire.

```
ID_STAGE INT(11) PRIMARY KEY
```

LDD : TABLE - contraintes

DEFAULT : valeur par défaut.

```
COULEUR VARCHAR(20) DEFAULT 'vert'
```

CHECK (<condition>): limite la plage de valeur de cet attribut sous certaines conditions.

```
SEXE CHAR(1) CHECK(SEXE in ('M', 'F'))
```

```
AGE INT(3) CHECK (AGE between 18 and 120)
```


LDD : TABLE - constraint

constraint ou contraintes de tables

Les contraintes de table ou **contraintes d'intégrité** sont introduites après la description des colonnes.

Contrainte sur les valeurs **unique**

```
constraint UN_TABLE[_N] unique (ATTRIBUT1, ..., [ATTRIBUTN])
```

```
constraint UN_HOTELS unique (NUMERO_CHAMBRE NUMBER(2) )
```

Contrainte sur clef primaire **primary key**

```
constraint PK_TABLE primary key (ATTRIBUT1, ..., [ATTRIBUTN])
```

```
constraint PK_ENTREPRISES_ELEVES primary key (ID_ENTREPRISE, ID_ELEVE)
```

LDD : TABLE - constraint

Contrainte sur **check**

```
constraint CK_TABLE[_N] check (<condition>)
```

```
constraint CK_ESPECES check (SEXE in ('M', 'F'))
```

Contrainte sur **foreign key**

```
constraint FK_TABLE[_N] foreign key (IDENTIFIANT_TABLE)  
references TABLE_MAITRE (IDENTIFIANT_MAITRE) [on DELETE CASCADE]
```

```
constraint ENTREPRISES_ELEVES_1 foreign key (ID_ENTREPRISE)  
references ENTREPRISES (ID_ENTREPRISE),  
constraint ENTREPRISES_ELEVES_2 foreign key (ID_ELEVE)  
references ELEVES (ID_ELEVE)
```

LDD : TABLE - constraint

Exemple :

```
create table ANIMAUX
(ID_ANIMAL          INT (11) auto_increment NOT NULL ,
NOM                 VARCHAR (40)      NOT NULL ,
SEXE                CHAR (1)          NOT NULL ,
DATE_NAISSANCE      DATE              NOT NULL ,
DATE_DECES          DATE              NOT NULL ,
ID_ESPECE           INT               NOT NULL ,
ID_MERE             INT               NULL ,
constraint PK_ANIMAUX primary key(ID_ANIMAL ),
constraint CK_ANIMAUX check (SEXE in ('M', 'F')),
constraint FK_ANIMAUX_1 foreign key (ID_ESPECE)
references ESPECES(ID_ESPECE),
constraint FK_ANIMAUX_2 foreign key (ID_MERE )
references ANIMAUX(ID_ANIMAL));
```

LDD : TABLE

Supprimer une table :

```
drop table TABLE_NAME;
```

Renommer une table :

```
rename table TABLE1 to TABLE2;
```

LDD : TABLE - alter

Modifier une table :

```
alter table TABLE_NAME action description;
```

action :

add : ajout ;

change : modification d'un attribut ;

modify : modification de type ;

drop : suppression.

LDD : TABLE - alter

Ajouter un attribut :

```
alter table TABLE_NAME add ATTRIBUT type;
```

```
alter table ELEVES add MOBILE VARCHAR(14);
```

Ajouter une contrainte :

```
alter table TABLE_NAME add constraint type;
```

```
alter table ELEVES add  
constraint ENTREPRISES_ELEVES_1 foreign key (ID_ENTREPRISE)  
references ENTREPRISES (ID_ENTREPRISE);
```

LDD : TABLE - alter

Modifier un attribut :

```
alter table TABLE_NAME change ATTRIBUT_OLD ATTRIBUT_NEW type;
```

```
alter table ELEVES change MOBILE PORTABLE VARCHAR(14);
```

```
alter table ELEVES change MOBILE PORTABLE VARCHAR(14) NOT NULL;
```

Modifier un type :

```
alter table TABLE_NAME modify ATTRIBUT type;
```

```
alter table ELEVES modify MOBILE VARCHAR(14) NOT NULL;
```

LDD : TABLE - alter

Supprimer un attribut :

```
alter table TABLE_NAME drop ATTRIBUT;
```

```
alter table ELEVES drop MOBILE;
```


LDD : INDEX

Créer un index :

```
create index [unique] X_NAME[_N] on TABLE_NAME (ATTRIBUT);
```

```
create index X_HOTELS on HOTELS (CHAMBRES);
```

```
create unique index X_HOTELS on HOTELS (CHAMBRES);
```

Supprimer un index :

```
drop index X_NAME;
```

```
drop index X_HOTELS;
```

LDD : bibliographie

Documentation officielle MySQL 5.7

<http://dev.mysql.com/doc/refman/5.7/en/sql-syntax-data-definition.html>

w3schools.com

<http://www.w3schools.com/sql/default.asp>