



Cours S.Q.L.

Langage de Définition de Données

L.D.D.

Structures des tables et contraintes d'intégrité

Réf. : <http://sqlpro.developpez.com/cours/sqlaz/ddl/>

<https://dev.mysql.com/doc/refman/9.0/en/>

CREATION DE TABLES

○ Exemple

```
CREATE TABLE HABITANT(
  ID INTEGER PRIMARY KEY,
  NOM VARCHAR(50) NOT NULL,
  PSEUDO CHAR(10) NOT NULL UNIQUE,
  ADRESSE VARCHAR(50),
  DATEN DATE NOT NULL CHECK(DATEN > '1900-01-01'),
  IDFONCTION INTEGER,
  CONSTRAINT FK_HABITANT_FONCTION
  FOREIGN KEY(IDFONCTION) REFERENCES FONCTION(ID)
);
```

Contraintes de colonnes

Contrainte de table

CREATION / SUPPRESSION DE TABLE

○ Création

```
CREATE TABLE <nom_table> (  
    <colonne_1> <type> [<contraintes_colonne_1>...]  
    [, <colonne_2> <type> [<contraintes_colonne_2>...]] ...  
    [, <contrainte_de_table_1>...]  
);
```

Rappels

< > : signifie qu'on attend un identificateur, une expression, ...

[] : entoure un élément de syntaxe facultatif

... : signifie que l'élément de syntaxe précédent peut se répéter

○ Suppression

```
DROP TABLE <nom_table> ;
```

TYPES DE CHAMPS *(standard SQL)*

- Chaînes de caractères sur 1 octet :
 - CHAR(n) : caractère de longueur fixe n
 - VARCHAR(n) : chaîne de longueur variable n
 - CLOB(n) : grande chaîne (CHARACTER LARGE OBJECT ; SQL3)

N.B. : la valeur maximale pour n dépend des SGBD

- Chaînes de caractères sur 2 octets (unicode) :
 - NCHAR(n) , NVARCHAR(n), NCLOB(n)
- Types temporels
 - DATE : type date (du 01/01/0001 au 31/12/9999)
 - TIME : heure ou durée (de 00:00:00.000 h à 23:59:59.999 h)
 - TIMESTAMP : date + heure

TYPES DE CHAMPS

- Numériques :
 - INTEGER : entier (stockage sur 1 « mot mémoire »)
 - SMALLINT : entier (stockage sur $\frac{1}{2}$ « mot mémoire »)
 - DECIMAL(n[,p]) : valeur numérique exacte
 - au maximum n chiffres dont p après la virgule
 - Synonyme : NUMERIC(n[,p])
 - FLOAT[(n[,p])], REAL[(n[,p])] : valeur numérique flottante (erreurs d'arrondi possibles)
- Binaire :
 - BIT(n) : chaîne de bits de taille fixe
 - BLOB [(n [m])] :
 - chaîne de bits de grande dimension (image, ...)
 - Binary Large Object (SQL3)
- Booléen
 - BOOLEAN
 - TRUE, FALSE ou NULL (SQL3)
 - Certains SGBD utilisent un entier

CONTRAINTE DE COLONNE

- S'applique à **une seule** colonne
- Contraintes possibles :
 - PRIMARY KEY : définit une clef primaire composée d'1 seul champ
 - UNIQUE : pas de doublons pour ce champ
 - DEFAULT <valeur> : précise une valeur par défaut
 - NULL : la valeur de ce champ peut être NULL (par défaut)
 - NOT NULL : la valeur de ce champ **ne peut pas** être NULL
 - FOREIGN KEY REFERENCES <table> (<colonne>) : définit une contrainte de clef étrangère en référence au champ d'une autre table
 - CHECK (<expression booléenne>) : la valeur n'est acceptée que si l'expression est vraie

CONTRAINTE DE TABLE

- S'applique à **une ou plusieurs** colonnes à la fois
- Contraintes possibles :
 - PRIMARY KEY (<col1> [,<col2>...]) : définit une clef primaire composée d'1 ou plusieurs champs
 - UNIQUE (<col1> [,<col2>...]) : contrainte d'unicité sur 1 ou plusieurs champs
 - FOREIGN KEY (<col1-1> [,<col1-2>...])
REFERENCES <table> (<col2-1> [,<col2-2>...]) :
définit une contrainte de clef étrangère portant sur 1 ou plusieurs champs en référence à la clef primaire de même nature d'une autre table
 - CHECK (<expression booléenne>) : la ligne n'est valide que si l'expression est vraie
- Syntaxe :
 - CONSTRAINT <nom_contrainte> <contrainte_de_table>

LES CONTRAINTES DE TABLE

○ Exemple :

```
CREATE TABLE PARTICIPER(  
    IDHABITANT INTEGER NOT NULL,  
    IDBATAILLE INTEGER NOT NULL,  
    NBDUELS INTEGER,  
    NBCASQUES INTEGER,  
    CONSTRAINT PK_PARTICIPER PRIMARY KEY (IDHABITANT, IDBATAILLE),  
    CONSTRAINT FK_PARTICIPER_HABITANT  
        FOREIGN KEY (IDHABITANT) REFERENCES HABITANT (ID),  
    CONSTRAINT FK_PARTICIPER_BATAILLE  
        FOREIGN KEY (IDBATAILLE) REFERENCES BATAILLE (ID),  
    CONSTRAINT CHK_NBCASQUES CHECK (NBCASQUES <= NBDUELS)  
);
```


MODIFICATION DE TABLE (COLONNES)

- Ajouter une colonne :

- **ALTER TABLE** <table> **ADD** [COLUMN] <définition_colonne> ;
- Exemple :

```
ALTER TABLE PARTICIPER ADD NBDUELS INTEGER NOT NULL;
```

- Modifier une colonne (type, contrainte, ...):

- **ALTER TABLE** <table> **MODIFY** [COLUMN] <définition_colonne> ;
- Attention, pas implémenté par tous les SGBD
- Exemple :

```
ALTER TABLE PARTICIPER MODIFY NBDUELS SMALLINT NULL;
```

- Supprimer une colonne :

- **ALTER TABLE** <table> **DROP** [COLUMN] <colonne> ;
- Exemple :

```
ALTER TABLE PARTICIPER DROP NBDUELS;
```

MODIFICATION DE TABLE (CONTRAINTES)

- Ajouter une contrainte de table :

- **ALTER TABLE** <table> **ADD CONSTRAINT** <définition_contrainte> ;
- Exemple :

```
ALTER TABLE HABITANT  
ADD CONSTRAINT FK_HABITANT_FONCTION  
FOREIGN KEY(IDFONCTION) REFERENCES FONCTION (ID);
```

- Supprimer une contrainte de table :

- **ALTER TABLE** <table>
DROP CONSTRAINT <nom_contrainte> [CASCADE | RESTRICT];
- CASCADE : toutes les objets dépendant de cette contrainte seront supprimés.
- RESTRICT : si un objet dépend de cette contrainte, la suppression sera annulée(défaut)
- Exemple :

```
ALTER TABLE HABITANT DROP CONSTRAINT FK_HABITANT_FONCTION;
```



CRÉATION DE TABLE À PARTIR D'UNE AUTRE

- Non standard
- Syntaxe pour Oracle Database et MySql :

CREATE TABLE <nom_table> [AS] <requête_select>;

- Recopie la structure et les données
- Exemple :

CREATE TABLE HABITANT_COPIE AS SELECT * FROM HABITANT;

- Pour recopier la structure **sans** les données :

**CREATE TABLE HABITANT_COPIE
AS**

SELECT * FROM HABITANT WHERE 0=1;

<https://dev.mysql.com/doc/refman/8.0/en/create-table-select.html>