

Introduction aux Systemes d'Information et aux Bases de Données

L2 Informatique

Serenella Cerrito

Département d'Informatique – Évry

2014-2015

Quels valeurs peut prendre un attribut ? Ici, les types les plus utilisés.

Type	Description
INTEGER (ou INT)	Entiers relatifs
FLOAT (ou REAL)	Flottants
DECIMAL(n,d)	Decimal (voir note (*))
CHAR(n)	chaîne de longueur n
VARCHAR(n)	chaîne de longueur n au max
BOOLEAN	TRUE, FALSE et... Unknown !
DATE	Date (jour, mois, an)

(*) constitué de n chiffres decimales dont 2 après la virgule

SQL comme langage de définition de données

Par exemple :

```
CREATE TABLE Artiste (  
  Nom varchar(20) NOT NULL,  
  Prenom varchar(15),  
  Annee_naissance decimal(4,0),  
  PRIMARY KEY (Nom)
```

- ▶ on crée une table dont le nom est Artiste,
- ▶ qui a un attribut Nom dont le type (= le domaine) est l'ensemble des chaînes ayant 20 comme longueur max
- ▶ un attribut Prenom dont le type est l'ensemble des chaînes ayant 20 comme longueur max,
- ▶ un attribut Annee_naissance constituée par 4 chiffres décimales, sans virgule

et d'autres choses encore....

SQL comme langage de définition de données

Suite de l'exemple :

```
CREATE TABLE Artiste (  
  Nom varchar(20) NOT NULL,  
  Prenom varchar(15),  
  Annee_naissance decimal(4,0),  
  PRIMARY KEY (Nom));
```

SQL admet la valeur spéciale NULL, qui, en effet, indique une absence de valeur. A remarquer : une opération arithmétique appliqué à NULL (NULL + 3), par exemple, donne NULL comme valeur, et un prédicat appliqué à NULL (NULL > 3, par exemple), donnera le booléen spécial UNKNOWN (inconnu).

Ici, l'option **NOT NULL** oblige toujours à indiquer une valeur pour **Nom**. Le SGBD rejettera toute tentative d'insérer une ligne dans **Artiste** sans donner de nom de l'artiste.

Suite de l'exemple :

```
CREATE TABLE Artiste (  
  Nom varchar(20) NOT NULL,  
  Prenom varchar(15),  
  Annee_naissance decimal(4,0),  
  PRIMARY KEY (Nom));
```

Ici **PRIMARY KEY (Nom)** déclare que Nom identifie de manière unique une n -uplet d'Artiste (c'est une clé) : il exclue la possibilité d'avoir deux ligne différentes mais avec la même valeur pour Nom.

Une clé n'est pas forcément constituée par un seul attribut : elle peut être un ensemble d'attributs, comme dans :

```
CREATE TABLE Role (  
  Nom_Role varchar(20) NOT NULL,  
  ID_film decimal(10,0) DEFAULT '0' NOT NULL,  
  Nom_acteur varchar(20) NOT NULL,  
  PRIMARY KEY (ID_film, Nom_acteur));
```

Tous les attributs figurant dans une clé doivent être déclarés NOT NULL

SQL comme langage de définition de données

Ce même exemple montre aussi la possibilité de déclarer une valeur par défaut avec l'option DEFAULT :

```
CREATE TABLE Role (  
  Nom_Role varchar(20) NOT NULL,  
  ID_film decimal(10,0) DEFAULT '0' NOT NULL,  
  Nom_acteur varchar(20) NOT NULL,  
  PRIMARY KEY (ID_film, Nom_acteur));
```

Ici : on ne permet pas l'insertion d'une ligne sans aucune valeur pour l'attribut ID_film (NOT NULL), mais, en absence d'information, le système affectera automatiquement la valeur '0' à ID_film pour cette ligne.

Comparer avec 2 déclarations différentes possibles :

```
ID_film decimal(10,0)
```

```
ID_film decimal(10,0) NOT NULL
```

Mot clé UNIQUE

```
CREATE TABLE Cinema (  
  Nom_cinema varchar(10) NOT NULL,  
  Arrondissement decimal(2,0),  
  Adresse varchar(30),  
  PRIMARY KEY (Nom_cinema)  
);
```

Ici, la clé primaire est Nom_cinema. Si on veut imposer aussi qu'on ne peut pas trouver deux cinémas exactement à la même adresse, il suffit de modifier la déclaration d'Adresse ainsi :

```
Adresse varchar(30) UNIQUE,
```

Clés Etrangères

La base cinema.sql du Td, ne déclare aucun lien entre Nom_cinema attribut de Salle et Nom_cinema attribut (et clé) de Cinema :

```
CREATE TABLE Salle (  
  Nom_cinema varchar(10) NOT NULL,  
  No_salle decimal(2,0) DEFAULT '0' NOT NULL,  
  Climatise char(1), Capacite decimal(4,0),  
  PRIMARY KEY (Nom_cinema, No_salle)  
);
```

Mais il serait naturel de demander que tout Nom_cinema mentionné par Salle soit un Nom_cinema répertorié dans la table Cinema. Pour cela, il suffit d'écrire Salle en ajoutant une dernière clause

FOREIGN KEY (Nom_cinema) REFERENCES CINEMA

Possibilité de modifier des tables

Exemple

```
ALTER TABLE Artiste ADD Nationalite VARCHAR(4);
```

puis

```
ALTER TABLE Artiste MODIFY Nationalite VARCHAR(30) NOT  
NULL;
```

puis

```
ALTER TABLE Artiste DROP Nationalite
```

Forme généralé pour l'**insertion d'un n -uplet de valeurs** dans une table R (ayant n attributes) :

INSERT INTO R VALUES (v_1, \dots, v_n)

Exemples

INSERT INTO Cinema **VALUES** ('Rex', 2, '22 Bd Poissoniere')

Forme généralé pour **effacer un n -uplet** d'une table R:

DELETE FROM R WHERE *Condition*

Exemple

DELETE FROM Cinema **WHERE** Arrondissement = '2'