Bases de Données

Amélie Gheerbrant

Université Paris Diderot
UFR Informatique
Laboratoire d'Informatique Algorithmique : Fondements et Applications

amelie@liafa.univ-paris-diderot.fr

29 septembre 2014

SQL

SQL permet d'interroger et de créer, de modifier et de supprimer des bases de données et des données.

- SQL query : interrogation de données
  - pas de modification dans la BD
- SQL-DDL : création, modification et suppression de schémas
  - création, modification et suppression de schémas de relation
  - définition de clés et d'autres contraintes
- SQL-DML : création, modification et suppression de données
  - insertion, modification et suppression de n-uplets

SQL DDL

DDL = Data Definition Language, permet de spécifier le schéma d'une base de données

- Le schéma de chaque relation
- Le domaine des valeurs associées à chaque attribut
- Les contraintes d'intégrité
- D'autres informations concernant le stockage physiques sur disque, la sécurité et les autorisations ($\Rightarrow$ nous ne verrons pas ça ici)

Quelques types d'attributs SQL courants

- **int** : entier ;
- **bool** : booléen ;
- **real** : réel ;
- **date** : date ;
- **time** : heure ;
- **numeric(precision, echelle)** : décimaux où echelle est le nombre de chiffres après la virgule et precision le nombre de chiffres totaux ;
- **text** : chaîne de caractères ;
- **varchar(longueur)** : chaîne d'au plus longueur caractères ;
- **serial** : entier à incrémentation automatique.

La commande Create Table

Pour créer une relation SQL on utilise la commande Create Table :

Create Table **R(A$_1$ D$_1$, A$_2$ D$_2$, ..., A$_n$ D$_n$,**
(contrainte_intégrité$_1$),
...,
(contrainte_intégrité$_n$)) ;

- R est le nom de la relation
- chaque A_i est un nom d'attribut du schéma de la relation R
- D$_i$ est le domaine de l'attribut A_i pour chaque $1 \leq i \leq n$

(NB : une commande SQL se termine toujours par un point virgule)

Exemple : définition de la relation Pilote

- **Create Table Pilote**

  (Plnum int,
  Plnom text,
  Plprenom text,
  Ville text,
  Salaire int,
  Primary Key Plnum) ;

- D'autres types auraient pu être choisis (e.g., varchar(30) pour Plnom)

- Primary Key Plnum = déclaration de la clef primaire de la table Pilote.

Contraintes d'intégrité

Une contrainte d'intégrité est une condition (logique) qui doit être satisfaite par les données stockées dans la BD.

But : maintenir la cohérence / l'intégrité de la BD

- Vérifier / valider automatiquement (en dehors de l'application) les données lors des mises-à-jour (insertion, modification, effacement)
- Déclencher automatiquement des mises-à-jour entre tables pour maintenir la cohérence globale.

Contraintes d'attributs

- PRIMARY KEY (<attributs>)
  - désigne un ensemble d'attributs comme la clé primaire de la table
- FOREIGN KEY (attributs) REFERENCES <table>
  - désigne un ensemble d'attributs comme la clé étrangère dans une contrainte référentielle
- <attribut> NOT NULL
  - spécifie qu'un attribut doit être renseigné
- UNIQUE (<attributs>)
  - spécifie un ensemble d'attributs dont les valeurs doivent être distinctes pour chaque couple de n-uplets ($\approx$ clef mais peut être nul et ne permet pas d'identifier un n-uplet)

Clefs primaires : remarque

- **Create Table Pilote**

  (Plnum int,
  Plnom text,
  Plprenom text,
  Ville text,
  Salaire int,
  Primary Key Plnum) ;

- PRIMARY KEY implique UNIQUE et NOT NULL.

Exemple : clefs et clefs étrangères

Relations : Pilote(Plnum, Plnom, Plprenom, Ville, Salaire), Avion(Avnum,Avnom,Capacité, Localisation), Vol(Volnum, Plnum, Avnum, Villedep, Villearr,Heuredep,Heurear)

```
CREATE TABLE Vol
(Volnum int,
Plnum text,
Avnum text,
Villedep text,
Villearr int,
Heuredep,
Heurearr,
PRIMARY KEY Volnum),
FOREIGN KEY (Plnum) REFERENCES Pilote(Plnum),
FOREIGN KEY (Avnum) REFERENCES Avion(Avnum));
```

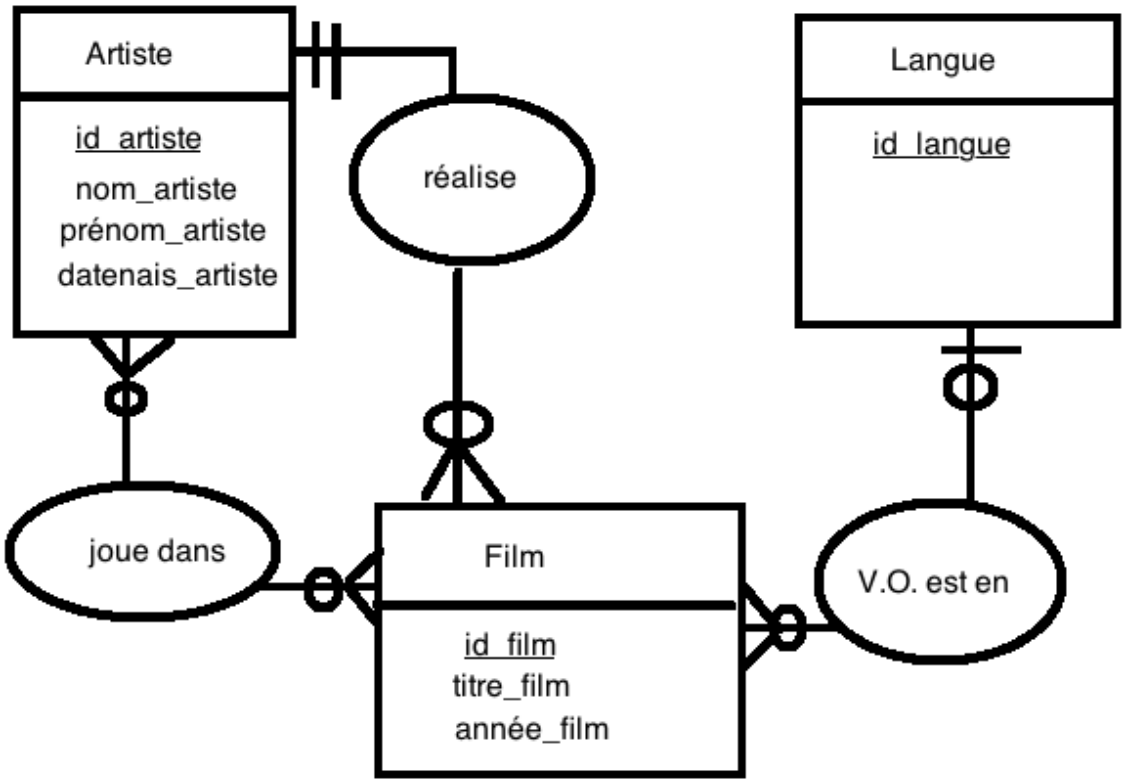
La clause CHECK

CHECK (P), où P est une condition

Exemple :
CHECK (salaire >= 20000)

Ajouté lors de la création de la relation Vol, permet d'assurer un salaire minimum à tous les pilotes.

Exemple plus complexe (au tableau)



La commande DROP TABLE

- DROP TABLE R <RESTRICT, CASCADE> ;
- **Restrict** (par défaut) : supprime R seulement si elle n'est référencée par aucune autre contrainte (clé étrangère) ou vue
- **Cascade** : supprime aussi toutes les tables qui dépendent de R
- Utilisé pour supprimer une relation ET sa définition (schéma)
- On ne peut plus utiliser la relation dans les requêtes, mises-à-jours, ou toute autre commande, puisque sa description n'existe plus.
- Exemple :
  DROP TABLE Vol ;
  DROP TABLE Pilote ;
  (Attention, Vol devra ici être supprimée avant Pilote.)

La commande ALTER TABLE

- ALTER TABLE R ADD A D ;
- Utilisé pour ajouter un attribut A de domaine D à une relation existante
- On assigne par défaut à tous les tuples de la relation la valeur nulle pour le nouvel attribut (dangereux).
- Peut aussi (sur certains SGBD) être utilisé pour supprimer un attribut dans une relation : ALTER TABLE R DROP A ;