

Introduction aux bases de données

Intégrité référentielle, la contrainte de Foreign KEY
Clé primaire composée

La contrainte de FOREIGN KEY

Plan de la séance

- Retour sur la dernière séance:
 - Point de vue de l'étudiant
 - Point de vue de l'enseignant.
- Rappels:
 - CREATE TABLE, les contraintes d'intégrité
- Définition
- Syntaxe
- Exemples
- Laboratoire 3

Rappel: CREATE TABLE

- La commande CREATE TABLE permet de créer une table dans une base de données.
- Pour créer une table nous avons besoin de connaître:
 - La liste des attributs de la table
 - Le type de données pour chaque attribut (NUMBER, VARCHAR2(n), CHAR(n), DATE, etc...)
 - Les contraintes d'intégrité, s'il y'en a, pour chaque attribut
 - PRIMARY KEY,
 - CHECK,
 - NOT NULL,
 - UNIQUE
 - Default
 - **Les contraintes niveau table**

Contrainte d'intégrité référentielle, définition

- On parle d'intégrité référentielle lorsque les valeurs d'une ou plusieurs colonnes d'une table sont déterminées ou font référence à des valeurs d'une colonne d'une autre table.
- Dans notre exemple ci-après, les valeurs de la colonne codeequipe de la table **Joueurs** font référence aux valeurs de la colonne codeequipe de la table **Equipes**
- La table Joueurs ne contient aucun **codeequipe** qui n'est pas dans la table Equipes.

Table joueurs

NUMJOUEUR	NOM	PRENOM	CODEEQUIPE
1	PRICE	CAREY	MTL
2	MARKOV	ANDRÉ	MTL
3	SUBBAN	KARL	MTL
4	PATIORETTY	MAX	MTL
10	HAMOND	ANDREW	OTT
6	STONE	MARC	OTT
9	TURIS	KYLE	OTT
7	GALLAGHER	BRANDON	MTL
8	TANGUAY	ALEX	AVL
11	THOMAS	BIL	AVL

Table Equipes

CODEEQUIPE	NOMEQUIPE	VILLE	NBCOUPES
1	LES CANADIENS DE MONTRÉAL	MONTRÉAL	24
2	LES MAPLE LEAFS	TORONTO	22
3	LES SÉNATEURS	OTTAWA	4
4	LES AVALANCHES	COLORADO	2
5	LES CANUKS	VANCOUVER	1
6	LES BRUNS DE BOSTON	BOSTON	13

référence

Contrainte d'intégrité référentielle, définition

- Dans la table Equipes, le **codeequipe** est une clé primaire.
- Dans la table Joueurs, le **codeequipe** est appelé **Clé étrangère** ou **FOREIGN KEY**.
- Pour parler de contrainte de **FOREIGN KEY** nous avons besoins de deux tables :
 - Une table A (exemple Equipes) qui contient un attribut (exemple Codeequipe) de clé primaire : PRIMARY KEY
 - Une table B (exemple Joueurs) qui va contenir un attribut de clé étrangère.
- La contrainte de FOREIGN KEY indique que la valeur de l'attribut de la table B(Joueurs) correspond à une valeur d'une clé primaire de la table spécifiée A. (Table Equipes)
- La clé primaire de l'autre table A (exemple table Equipes) doit être obligatoirement crée pour que cette contrainte soit acceptée. On ne peut pas faire référence à un attribut qui n'existe pas.
- La clé primaire la table A (Equipes) et l'attribut défini comme clé étrangère de la table B (Joueurs) doivent être de même type et de même longueur.
- Il n'est pas nécessaire que les attributs de clé primaire et de clé étrangère aient des noms identiques.

Contrainte d'intégrité référentielle

Exemple

La table EQUIPES doit être créée en premier

```
CREATE table EQUIPES
(
Codeequipe CHAR(3) CONSTRAINT pk_equipe PRIMARY KEY,
nomequipe VARCHAR2(50) NOT NULL,
Ville VARCHAR2(40),
nbcoupes NUMBER(2,0) CONSTRAINT ck_nbcoupe CHECK (nb_coupes > =0)
);
```

Contrainte d'intégrité référentielle, Exemples

On crée la table JOUEURS après

```
CREATE TABLE Joueurs
(
numjoueur NUMBER(3,0) CONSTRAINT pk_joueurs PRIMARY KEY,
nom VARCHAR2(30) NOT NULL,
prenom VARCHAR2(30),
codeequipe CHAR(3),
CONSTRAINT fk_Joueurs_equipes FOREIGN KEY (codeequipe) REFERENCES equipes(codeequipe)
);
```

Les attributs codeequipe des deux tables sont de même type et de même longueur: CHAR(3). C'est une obligation.

Il y a une virgule en rouge dans la table JOUEURS après le codeequipe ce qui indique que la contrainte de **FOREIGN KEY** est une contrainte sur la table. C'est une obligation

Le mot réservé **REFERENCES** indique la colonne (attribut) référencée de la table de la clé primaire. C'est pourquoi la table de la clé primaire doit être créée en premier. On ne peut pas référencer quelque chose qui n'existe pas.

Contrainte d'intégrité référentielle

La contrainte de FOREIGN KEY garantie l'intégrité référentielle ce qui veut dire :

- Vous ne pouvez pas modifier (UPDATE) la valeur de l'attribut de la clé primaire s'il a une valeur de clé étrangère.
 - Exemple dans la table EQUIPES vous ne pouvez pas modifier le codeequipe MTL pour BLA, car il existe des joueurs ayant le codeequipe MTL.
- Vous ne pouvez pas supprimer (DELETE) une ligne de la table référencée s'il existe des enregistrements ayant une valeur de la clé étrangère égale à la valeur de la clé primaire de la ligne à supprimer.
 - Exemple vous ne pouvez pas supprimer de la table equipes l'équipe dont le codeequipe est OTT, car il existe des joueurs ayant ce codeequipe.
- Aucune insertion (INSERT INTO) ni modification (UPDATE) n'est possible dans la table de la clé étrangère si la contrainte d'intégrité référentielle n'est pas respectée.
 - Exemple INSERT INTO Joueurs VALUES (1, 'LeRoy','Gibbs', 'GOM') ne marchera pas car GOM n'est pas dans la table Equipes
- Vous ne pouvez pas détruire la table EQUIPES par un simple DROP TABLE (à voir plus loin). Il faudra
 - Soit supprimer la table joueurs en premier (à condition qu'elle ne soit pas référencée elle aussi)
 - Soit utiliser CASCADE CONSTRAINTS.
 - Soit désactiver ou détruire la contrainte d'intégrité (la FOREIGN KEY)

Clé primaire composée

- Il arrive qu'une table ait besoin de deux (ou plus) attributs pour identifier de manière unique les enregistrements. Dans ce cas on parle de clé primaire composée.
- Dans la plupart des cas les attributs de la clé primaire sont des clés étrangères.
- La définition de la clé primaire se fait au niveau TABLE et non pas colonne ou attribut.
- Pour définir une clé primaire composée, il suffit de fournir la liste des attributs de la clé séparés par des virgules entre parenthèses.
- Si une table a une clé primaire composée dont les attributs sont des clés étrangères, la table est dite : Table de relation

Clé primaire composée

Exemple

```
create table Films
(
  id_film number(5,0) constraint pk_film primary key,
  titre varchar(100) not null,
  annee number(4,0) not null
)
```

```
create table Acteurs
(
  id_acteur number(5,0) constraint pk_acteur PRIMARY key,
  nom_acteur varchar2(40) not null,
  prenom_acteur varchar2(30) not null
);
```

Clé primaire composée

On crée la table Acteurs_films après avoir créé les tables Films et Acteurs.

```
create table Acteurs_films
(
  id_film number(5,0),
  id_acteur number(5,0),
  salairePourleFilm number(10,0) not null constraint ck_salairefilm check(salairePourleFilm >500000),
  constraint fk_films foreign key (id_film) references films(id_film),
  constraint fk_acteurs foreign key (id_acteur) references acteurs(id_acteur),
  constraint pk_acteurs_films primary key (id_film,id_acteur)
);
```

- La table Acteurs_films a une clé primaire composée. Cette clé est (id_film,id_acteur)
- La contrainte de clé primaire composée est une contrainte sur la TABLE. (tout comme la contrainte de Foreign key).
- La table Acteurs_films a deux contraintes de FOREIGN KEY.
- La table Acteurs_films est appelée table de RELATION.

Conclusion et bonnes pratiques pour la commande CREATE TABLE

- Comme mentionné déjà il faut que les noms des tables, des colonnes, et des contraintes soient significatifs.
- Pour les noms de contrainte, vous pouvez utiliser:
 - pk_nomdetable pour PRIMARY KEY
 - fk_nomdetable_nomtableOrigine pour la FOREIGN KEY ou fk_nomTableOrigine

```
constraint fk_films foreign key (id_film) references films(id_film),  
constraint fk_films_categories foreign key (id_categorie) references Categories(id_categorie),
```

- ck_Colonne pour CHECK
- Lorsque vous pouvez utiliser les contraintes niveau table, faites-le. Ça donne un code plus propre.
- Lorsque c'est recommandé, et possible, utilisez un numéro séquentiel pour la clé primaire.

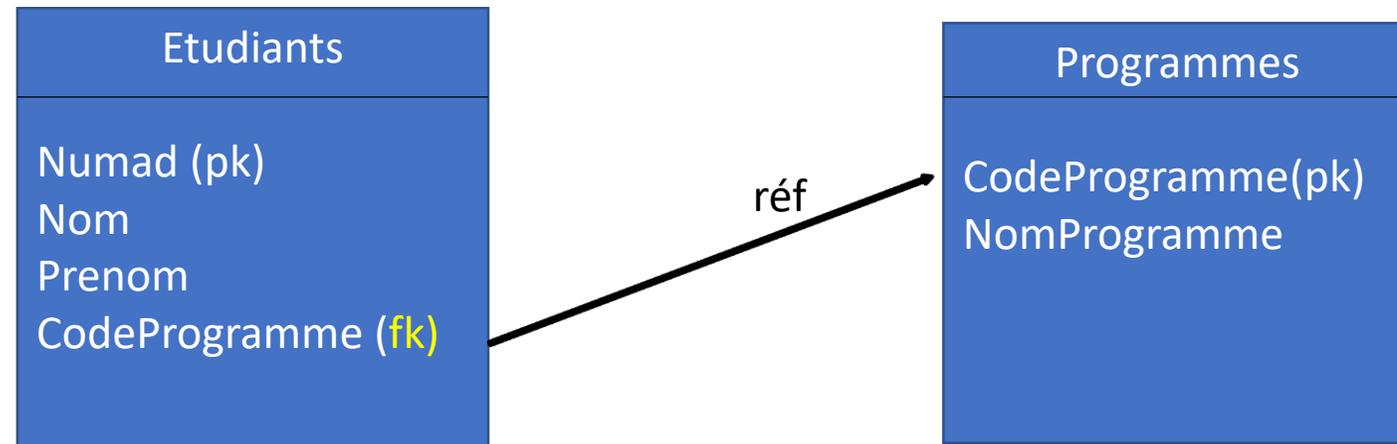
Conclusion et bonnes pratiques pour la commande CREATE TABLE

```
create table Films
(
id_film number(5,0),
titre varchar(100) not null,
annee number(4,0) not null,
classe char(3) not null,
id_categorie number(2,0),
id_langue char(2) not null,

---définition des contraintes
constraint pk_film primary key(id_film),
constraint ck_classe check (classe in('gen','13+','08+')),
constraint fk_films_categories foreign key (id_categorie)references Categories(id_categorie),
constraint fk_langues foreign key(id_langue ) references langues(id_langue)
);
```

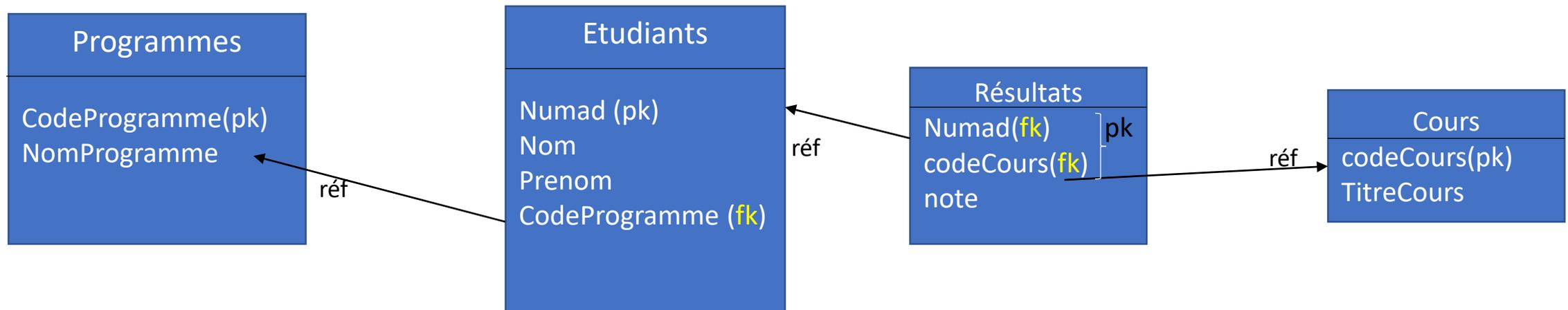
Représentation graphique, diagramme référentiel

- Dans le schéma suivant, on montre que la FK de la table étudiants fait référence à la PK de la table Programmes.
- Les tables sont représentées par des carrés. Le nom des tables est clairement indiqué en haut.
- Les attributs des tables, sont clairement indiqués aussi. Les PK et les FK sont clairement mises en évidence
- Le lien pointe de la clé étrangère vers la clé primaire. On dit que la clé étrangère fait référence à la clé primaire.
- Plus tard dans la session, le diagramme référentiel dans Oracle Data Modeler vous sera montré.



Représentation graphique.

- Autre exemple.



La contrainte d'intégrité référentielle



Conclusion



Questions