

L3 – UE 10 Fondement des bases de données

Contraintes et Dictionnaire – TP2

Un compte rendu est à déposer sur Spiral avant le mercredi 24 octobre, 23h59, sous la forme d'un fichier .sql commenté ou .txt.

Reprenons le schéma du TP précédent :

Schéma:

Movie (mID, title, year, director)

Reviewer (rID, name)

Rating (rID, mID, stars, ratingDate)

Dans un premier temps, supprimez les tables créées lors du TP1.

Modifiez ensuite les déclarations de création de tables (cf TP1) afin de prendre en compte les contraintes suivantes:

Clés.

- 1) mID est une clé de Movie. On la choisit comme clé primaire.
- 2) Le couple (title, year) est également une clé.
- 3) rID est une clé de Reviewer.
- 4) (rID, mID, ratingDate) est une clé de Rating (mais avec des valeurs nulles autorisées).

Non-nulles.

- 5) Reviewer.name doit être non nul.
- 6) Rating.stars doit être non nul.

Contraintes de domaines.

- 7) Les films doivent être réalisés après 1900.
- 8) Rating.stars est défini sur l'ensemble {1,2,3,4,5}.
- 9) Rating.ratingDate doit prendre une valeur supérieure ou égale à 2000.

Contrainte sur des tuples.

- 10) Les films réalisés par Steven Spielberg sont antérieurs à 1990 et ceux de James Cameron sont postérieurs à 1990.

Peuplement de la base :

Après avoir créé les 3 tables, vous devez être capable de les peupler avec les données originales (voir TP1) sans engendrer d'erreurs.

Vérifications :

Chacune des commandes suivantes doit générer une erreur (donnez la contrainte violée) sinon expliquez pourquoi.

- 11) update Movie set mID = mID +1;
- 12) insert into Movie values (109, 'Titanic', 1997, 'JC');

- 13) insert into Reviewer values (201, 'Ted Codd');
- 14) update Rating set rID = 205, mID=104;
- 15) insert into Reviewer values (209, null);
- 16) update Rating set stars = null where rID = 208;
- 17) update Movie set year = year - 40;
- 18) update Rating set stars = stars + 1;
- 19) insert into Rating values (201, 101, 1, '1999-01-01');
- 20) insert into Movie values (109, 'Jurassic Park', 1993, 'Steven Spielberg');
- 21) update Movie set year = year-10 where title = 'Titanic';

Les commandes suivantes, quant à elles, ne doivent pas générer d'erreurs:

- 22) insert into Movie values (109, 'Titanic', 2001, null);
- 23) update Rating set mID = 109;
- 24) update Movie set year = 1901 where director <> 'James Cameron';
- 25) update Rating set stars = stars - 1;

Contraintes d'intégrité référentielles.

Rappel : Oracle permet plusieurs types d'application des contraintes d'intégrité référentielle. Il suffit de l'indiquer au moment de la déclaration de la contrainte via FOREIGN KEY :

- Pour empêcher toute m.a.j ou suppression d'une clé parent. Par défaut, la m.a.j ou la suppression d'une clé parent qui est référencée par un tuple dans la table enfant. Par exemple,

```
CREATE TABLE Emp_tab (
  FOREIGN KEY (Deptno) REFERENCES Dept_tab);
```

- Suppression des tuples fils quand la clé maître est supprimée. ON DELETE CASCADE permet de supprimer des clés maîtres qui sont référencées dans la table esclave. Par contre, la m.a.j n'est pas possible. Quand un clé maître est supprimée, tous les tuples de la table esclave qui dépendent de la valeur supprimée sont également supprimés. Par exemple :

```
CREATE TABLE Emp_tab (
  FOREIGN KEY (Deptno) REFERENCES Dept_tab
  ON DELETE CASCADE);
```

- Les clés étrangères à null quand les parent sont supprimés. ON DELETE SET NULL permet de ne pas supprimer les tuples esclaves quand un tuple de la table maître est supprimé. Les valeurs sont alors mises à NULL. Par exemple,

```
CREATE TABLE Emp_tab (
  FOREIGN KEY (Deptno) REFERENCES Dept_tab
  ON DELETE SET NULL);
```

Modifiez la définition d'une ou plusieurs tables pour prendre en compte les contraintes suivantes.

- 26) Intégrité référentielle entre Rating.rID et Reviewer.rID. (Mis-à-jour sur Reviewers : cascade, suppression sur Reviewers : set null, erreur sinon).
- 27) Intégrité référentielle entre Rating.mID et Movie.mID (suppression sur Movies : cascade, erreur pour le reste).

Recharger le base de données.

Recréez les trois tables en prenant en compte vos modifications. Vous devez être capable de charger les données originales sans erreur.

Est ce que les intructions suivantes génèrent des erreurs ou non ?

- 28) insert into Rating values (209, 109, 3, '2001-01-01');
- 29) update Rating set rID = 209 where rID = 208;
- 30) update Rating set mID = mID + 1;
- 31) update Movie set mID = 109 where mID = 108;
- 32) update Movie set mID = 109 where mID = 102;
- 33) update Reviewer set rID = rID + 10;
- 34) delete from Reviewer where rID > 215;
- 35) delete from Movie where mID < 105;

Dictionnaire.

Le dictionnaire de données contient les vues suivantes décrivant les différentes contraintes d'intégrité définies :

- ALL_CONSTRAINTS
- ALL_CONS_COLUMNS
- USER_CONSTRAINTS
- USER_CONS_COLUMNS
- DBA_CONSTRAINTS
- DBA_CONS_COLUMNS

Vous pouvez interroger ces vues pour trouver le nom des contraintes, sur quels attributs elles portent, et bien d'autres informations pour les gérer.

- 36) Etudiez ces vues, via la commande describe par exemple.
- 37) Listez toutes les contraintes définies sur des tables qui vous sont accessibles.
- 38) Des contraintes ont des noms spécifiés par des utilisateurs alors que d'autres sont nommées par le système (SYS_*). Donnez la liste des contraintes nommées par le système et celle nommée par l'utilisateur.
- 39) Différence entre contraintes not null et check. Plusieurs contraintes sont listées avec un type 'C'. Pour distinguer celles qui sont des contraintes NOT NULL et celles qui sont de type CHECK, on peut étudier le champ Search_condition de la vue User_constraints. En effectuant une requête sur cette vue, donnez les contraintes de type NOT NULL, puis celle de type CHECK sur les tables que vous avez définies.
- 40) En utilisant la vue User_cons_columns, on peut connaître les attributs qui sont ciblés par les contraintes. Interrogez cette vue, que remarquez vous ?

Un peu plus ...

41) Jusqu'à présent, nous avons supprimé puis créé une nouvelle base de données à chaque nouvelle contrainte. Reprendre les première questions en ajoutant les contraintes sans supprimer les tables.