TD 9

Création des schémas – Modification des données

Rappels

Les contraintes d'intégrité permettent a l'utilisateur de définir des conditions que doivent respecter les données. Le plus souvent, les contraintes sont définies lors de la création des tables (CREATE TABLE). Elles peuvent néanmoins être rajoutées sur des tables existant sous certaines conditions. Il existe principalement deux types de contraintes : contrainte de clé (PRIMARY KEY et UNIQUE), contrainte référentielle (FOREIGN KEY) et contrainte de domaine (clause CHECK. Toutes ces contraintes sont locales : elles sont définies pour une seule table. Il est aussi possible de définir des contraintes globales en utilisant la clause CREATE ASSERTION. A noter que certains systèmes ne supportent pas les assertions.

Table des matières

Villes et Pays

On veut créer un schéma relationnel pour stocker des informations sur des villes et des pays.

1. Traduisez le schéma relationnel suivant en instructions SQL:

   Ville(nom, population, pays*)
   Pays(snom, capitale*)

où p**ays** est une référencer vers un pays dans la table **Pays** et **capitale** est une référencer vers une ville dans la table **Ville**.

```
create table Ville (
        nom varchar(32),
        population number(10),
        pays varchar(32),
        constraint pk_ville primary key(nom)
) ;

create table Pays (
        nom varchar(32),
        capitale varchar(32),
        constraint pk_pays primary key(nom)
) ;

alter table Ville add constraint fk_pays foreign key(pays) references Pays(nom) ;

alter table Pays add constraint fk_ville foreign key(capitale) references Ville(nom) ;
```

2. Insérez la France avec sa capitale Paris (3 millions d'habitants) dans la base de données.

```
insert into Ville values('Paris', 3000000, null) ;

insert into Pays values('France', 'Paris') ;

update Ville set pays = 'France' where nom='Paris' ;


/* Autre solution : Oracle */

alter table Ville disable constraint fk_pays ;

insert into Ville values('Paris', 3000000, 'France') ;

insert into Pays values('France', 'Paris') ;

alter table Ville enable constraint fk_pays ;
```

```
/* Autre solution : H2 */
ALTER TABLE Ville SET REFERENTIAL_INTEGRITY FALSE ;
insert into Ville values('Paris', 3000000, 'France') ;
insert into Pays values('France', 'Paris') ;
ALTER TABLE Ville SET REFERENTIAL_INTEGRITY TRUE ;
```

3. Modifiez le schéma de telle manière que la suppression d'un pays déclenche automatiquement la suppression de toutes les villes du pays.

```
alter table Ville drop constraint fk_pays ;
alter table Ville add constraint fk_pays foreign key(pays) references Pays(nom) on delete cascade ;
```

4. Effacez les deux tables **Ville** et **Pays** du schéma.

```
alter table Ville drop constraint fk_pays ;
drop table Pays ;
drop table Ville ;

/* Autre ordre possible */
alter table Ville drop constraint fk_pays ;
drop table Pays ;
drop table Ville ;
```

Arbres ordonnées

Voici une table **Arbre** qui permet de stocker des arbres ordonnes dans une base de données relationnelle:

```
create table Arbre(
        id number(10),
        par number(10),
        pos number(2),
        constraint pk primary key (id),
        constraint fk foreign key (par) references Arbre (id)
) ;
```

- **id** est l'identifiant du noeud,
- **par** est l'identifiant du parent,
- **pos** est la position parmi les enfants ;

1. Insérez l'arbre binaire (1(2(4,5),3(6,7(8,9)))) dans la base de données.
2. Est-ce qu'on peut insérer des données incohérentes ?
3. Comment faut il modifier le schéma pour empêcher cette incohérence.
4. Effacez le sous-arbre (2(4,5)) de la base de données.
5. Effacez le sous-arbre 3(6,7(8,9)) de la base de données.
6. Est-ce que l'instruction suivante est possible sur l'arbre initial (avant l'effacement) ?

   delete from Arbre where id=3 ;
7. Comment peut-on modifier le schéma pour effacer le sous-arbre avec l'instruction précédente ?
8. Est-ce qu'il est possible de modifier l'identifiant d'un noeud ? Par exemple : *update Arbre set id=10 where id = 3 ;*
9. Comment peut-on modifier le schéma pour permettre l'instruction précédente ?
10. Quel est le résultat des instructions suivantes (sur l'arbre initial):

    alter table Arbre drop constraint fk ;
    alter table Arbre add constraint fk foreign key (par) références Arbre(id) on delete set null;
    delete from Arbre where id=3;

Mise à jours de Table

Considérer le schéma du TD précédent. Pour simplifier, l'attribut NumSS de Employe est remplace par un attribut du même type avec au max 5 chiffres ; la contrainte C4 n'est plus nécessaire. Le but de cet exercice est d'exprimer des instructions permettant d'insérer, de modifier et de supprimer des n-uplets. Dire a chaque fois si l'instruction exprimée est acceptée ou rejetée par le système en justifiant.

Insertions

1. Insérer l'employé identifie par '12456' qui se prénomme 'Alain'.

```
insert into employe(numSS, prenomE) values(12345, 'Alain') ;
```

2. Insérer l'employée identifiée par '21456' qui s'appelle 'LARS Anna', qui habite 'Paris' et qui est née le 25-08-1975.

```
insert into employe values(21456, 'LARS', 'Anna', 'paris', '25-08-1975') ;
```

3. Insérer le projet numéro '78143' dénomme 'ORCA' qui s'opère sous la responsabilité de 'Lars Anna' a Paris et qui a pour budget 250 000 euros.

```
insert into projet values(78143, 'ORCA', 21456, 'paris', 250000) ;
```

```
/* Autre solution */
insert into projet
select 78143, 'ORCA', NumSS, villeE, 250000
from Employe where nomE = 'LARS' and prenomE = 'Anna' ;
```

4. Renseigner dans la base les salaires correspondants aux profils suivants :
    1) 'Responsable' → 80 000

```
insert into grille_sal values('Responsable', 80000) ;
```

    2) 'Développeur' → 45 000

```
insert into grille_sal values('Développeur', 45000) ;
```

3) 'Technicien' → 35 000

insert into grille_sal values('Technicien', 35000) ;

5. Renseigner dans la base le fait que 'Alain' a été embauche dans le projet 'ORCA' en tant que testeur en date du 01-04-2014.

Impossible

Note : car pas de profil pour testeur, mais on peut forcer l'insertion en mettant null à la place, même si cela n'a pas de sens réel

6. Renseigner dans la base le fait que 'LARS Anna' fut embauchée dans le projet 'MEDUSA' en date du 28-02-2012 en tant que 'Développeur'.

Impossible

Note : car le projet MEDUSA n'existe pas et impossible de mettre null, car numProj dans la table EMBAUCHE fait parti de la clef primaire et donc ne peut pas être null

Suppressions

Note : On considéré que les questions suivantes sont indépendantes.

7. Supprimer les employés de plus de 67 ans

delete from Employe where ((sysdate-datenaiss) / 365) > 67 ;

/* Autre solution */

delete from Employe where DATEDIFF(year, DateNaiss, Sysdate) > 67 ;

8. Supprimer l'employé 'Alain'

delete from Employe where prenomE = 'Alain' ;

9. Supprimer l'employée 'LARS Anna'

delete from Employe where prenomE = 'Anna' and nomE = 'LARS' ;

10. Supprimer les employés embauches lors de la dernière année.

delete from Employe where NumSS IN (select NumSS from Embauche where (sysdate - dateEmb) < 365) ;

/*Autre solution*/

delete from Employe where NumSS = ANY (select NumSS from Embauche where (sysdate - dateEmb) < 365) ;

11. Supprimer les projets dont la proportion des salaires dépasse la moitie du budget.

//

Mises à jour

12. Désormais on connaît que l’employé Alain porte le nom BERNARD. Répercuter cette information.

update Employe set nomE = 'BERNARD' where prenomE = 'Alain' ;

13. L’employée LARS Anna est promue a la tête du projet identifie par 78143. Elle doit d'abord déménager avant de prendre ses responsabilités. Donner l'instruction qui modifie la ville de cette employée.

update Employe

set villeE = (select villeP from Projet where NumProj = 78143)

where PrenomE = 'Anna' and NomE = 'LARS' ; /* ou where numSS = 21456 ; */

On suppose qu'il existe un n-uplet dans la table Projet avec les données suivantes : (78143, 'ORCA', null, 'paris', 250000)

14. Le budget de chaque projet équivaut au double des salaires de ses employés. Donner une instruction qui modifie les budgets des projets en conséquence.

update Projet p

set budget = (select sum(s.salaire) from embauche e, grille_sal s

where p.numproj = e.numproj and e.profil = s.profil)*2 ;