UPMC PARIS UNIVERSITAS

Numéro d'anonymat : | | | |

LI341 –Bases de données

Partiel du 31 octobre 2013
2 heures
CORRIGÉ
Documents autorisés

Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 20 points (10 questions) n'a qu'une valeur indicative.

1 Création de schéma (6 pts)

La RATP veut gérer son réseau de bus dans une base de données relationnelle.

- Le réseau est divisé en plusieurs lignes de bus identifiées par un numéro unique (par exemple ligne 186).
- Chaque ligne relie des stations identifiées par des noms uniques (par exemple 'St. Michel'). Chaque station se trouve dans une ville (par exemple 'Paris') et plusieurs lignes de bus peuvent s'arrêter à la même station.
- Un bus est identifié par un numéro unique (par exemple 175689) et affecté à une seule ligne et une seule station qui est son dépôt.
- Un conducteur est identifié par son nom et son prénom. On connaît également son âge et la station la plus proche de son adresse d'habitation.
- Chaque jour de la semaine est identifié par un numéro (lundi = 1, ... dimanche = 7) et divisé en trois tranches horaires (matin=1, après-midi=2, soir=3). Chaque conducteur conduit un bus pendant toute une tranche horaire par jour.

Question 1 (3 points)
Définir en SQL les contraintes traduisant les exigences suivantes et seulement celles-ci :

- chaque identifiant d'une entité est la clé primaire de la table correspondante;
- toutes les références vers des clés primaires doivent être validées (contraintes d'intégrité référentielle);
- quand une ligne est supprimée, tous les arrêts sont également supprimés et la ligne des bus correspondants devient inconnue;
- quand une station est supprimée, on supprime tous les arrêts correspondants et les autres références deviennent inconnues;
- un conducteur ne peut pas travailler pendant plus d'une tranche horaire par jour;
- le jour est une valeur entre 1 et 7 et la tranche horaire entre 1 et 3.

Compléter chaque instruction `create table` avec les contraintes ci-dessus.

Solution:

```
create table Ligne (
  noLigne smallint primary key
);
create table Bus (
  noBus smallint primary key
  noLigne smallint
  depot varchar(20)
  foreign key depot references Station on delete set null
  foreign key noLigne references Ligne on delete set null
);
create table Station (
  nomStation varchar(20) primary key
  ville varchar(20)
);
create table Arrêt (
  nomStation varchar(20)
  ligne smallint
  foreign key nomStation references Station on delete cascade
  foreign key noLigne references Ligne on delete cascade
);
create table Conducteur (
  nom varchar(32)
  prenom varchar(24)
  age smalllint
  station varchar(20)
  primary key (nom, prenom)
  foreign key nomStation references Station on delete set null
);
create table Planning (
  nom varchar(32)
  prenom varchar(24)
  noBus smallint
  jour smallint check (jour between 1 and 7)
  tranche smallint check (tranche between 1 and 3)
  foreign key (nom, prenom) references Conducteur
  foreign key noBus references Bus
  unique (nom,prenom,jour)
) ;
```

Question 2 (3 points)

Dessinez le schéma E/A en précisant les types d'entité, les types d'associations, les attributs, *les cardinalités* et les identifiants. Utilisez les noms du schéma relationnel précédent pour les entités, associations et attributs.

Solution:

Entités (identifiants avec *):

1. Ligne : noLigne*
2. Bus : noBus*
3. Station : nomStation*, ville
4. Conducteur: (nom, prenom)*, age

Associations:

1. Bus – 0,1 – ligne – 0,n – Ligne
2. Bus – 0,1 – depot – 0,n – Station
3. Ligne – 0,n – arret – 0,n – Station
4. Conducteur – 0,1 – station – 0,n – Station
5. Conducteur – 0,n – planning – 0,n – Bus; attributs : jour, tranche

Introduire un type d'entité Ville n'est pas faux.

Une autre solution (qui est même plus correct) est d'introduire une entité Tranche: (jour, tanche)* et définir le planning comme une association ternaire entre Conducteur, Bus et Tranche.

2 Requêtes (8 pts)

Soit un schéma relationnel

Bus(noBus, noLigne)	**Ligne**(noLigne)
Station(nomStation, ville)	**Arrêt**(nomStation, noLigne)
Conducteur(nom, prenom, age)	**Planning**(nom, prenom, noBus, jour, tranche)

Les clés primaires qui apparaissent dans d'autres tables correspondent à des clés étrangères. Exprimer, *lorsque c'est indiqué*, en calcul relationnel et en SQL les requêtes suivantes.

Question 3 (1 point)

Les noms et l'âge des conducteurs qui conduisent un bus de la ligne 172 le samedi soir (calcul et SQL).

Solution:

```
select c.nom, c.age
from Conducteur c, Planning p, Bus b
where c.nom = p.nom and c.prenom = p.prenom and p.noBus = b.nuBus
and b.noLigne = 172 and p.jour = 6 and b.tranche = 3;
```

Bus(noBus, noLigne) **Ligne**(noLigne) **Station**(nomStation, ville) **Arrêt**(nomStation, noLigne)
Conducteur(nom, prenom, age) **Planning**(nom, prenom, noBus, jour, tranche)

Question 4 (2 points)
(calcul et SQL) Les noms des stations où s'arrêtent au moins deux lignes de bus différentes.

Solution:

```
select distinct a.nomStation
from Arret a, Arret b
where a.nomStation = b.nomStation
  and a.noLigne < b.noLigne
```

$\{a.nomStation | Arret(a) \wedge \exists b(Arretb \wedge a.nomStation = b.nomStation \wedge a.noLigne < b.noLigne)\}$

Question 5 (3 points)
(SQL) Le nombre des stations *à Paris* où une seule ligne de bus s'arrête.

Solution:

```
select count(distinct a.nomStation)
from Arret a, Station s
where a.nomStation = s.nomStation and  s.ville='Paris'
  and not exists (select b.noLigne
                  from Arret b
                  where a.nomStation = b.nomStation
                    and a.noLigne <> b.noLigne)
```

La requête est aussi correct sans distinct.

Bus(noBus, noLigne) **Ligne**(noLigne) **Station**(nomStation, ville) **Arrêt**(nomStation, noLigne)
Conducteur(nom, prenom, age) **Planning**(nom, prenom, noBus, jour, tranche)

Question 6 (2 points)
Pour chaque ville, on veut connaître le nom de la ville, le nombre de stations et le nombre de lignes avec au moins un arrêt dans la ville.

Solution:

```
select ville, count(distinct nomStation), count(distinct noLigne)
  from Station, Arret
 where Station.nomStation = Arret.nomStation
 group by ville
```

Question bonus (2 points)

(SQL) Les conducteurs qui conduisent au moins une fois tous les bus de la ligne 182.

Solution:

```
select c.nom, c.prenom
  from Conducteur c
 where not exists
   (select *
      from Bus b
     where b.noLigne = 182
       and not exists
           (select *
              from Planning p
             where p.noBus = b.noBus
               and p.nom = c.nom and p.prenom=c.prenom));
```

Bus(noBus, noLigne) **Ligne**(noLigne) **Station**(nomStation, ville) **Arrêt**(nomStation, noLigne)
Conducteur(nom, prenom, age) **Planning**(nom, prenom, noBus, jour, tranche)

3 Triggers (3 pts)

On considère à nouveau le schéma relationnel RATP et on veut exprimer la contrainte suivante:
«Il n'existe pas de ligne avec moins de trois arrêts».

Question 7 (1 point)
Donnez la requête qui retourne les lignes avec au moins trois arrêts.

Solution:

```
select noLigne
from Arrêt
group by noLigne
having count(*) >= 3;
```

Question 8 (2 points)
Définissez un trigger qui doit être déclenché à chaque suppression d'un arrêt dans la table **Arrêt** et qui empêche cette opération si la contrainte précédente n'est pas satisfaite. Vous avez le choix entre un trigger BEFORE ou AFTER.

Solution:

```
CREATE TRIGGER supprArret
BEFORE DELETE ON TABLE Arrêt
FOR EACH ROW
WHEN
   (3 >= (select count(*) from Arrêt where noLigne = OLD.noLigne))
THEN
BEGIN
   ABORT;
END;
```

4 Dépendances fonctionnelles (3 pts)

On considère la table **RATP(bus, ligne, station, ville, nom, prenom, age, jour, tranche)** et l'ensemble de dépendances fonctionnelles suivant:

F={ bus → ligne;
station → ville;
nom, prenom → age;
bus, jour, tranche→nom, prenom;
jour, nom, prenom→tranche, bus }

Question 9 (1 point)
Cochez la bonne réponse par rapport aux contraintes décrites dans l'ensemble de dépendances fonctionnelles F.

Solution:

1. Un bus peut être affecté à deux lignes de bus. oui ☐ non ■
2. Deux villes différentes peuvent avoir des stations avec le même nom. oui ☐ non ■
3. Un conducteur peut conduire le même bus à différents jours. oui ■ non ☐
4. Un bus peut être conduit par différentes conducteurs le même jour et à la même tranche horaire. oui ☐ non ■

Question 10 (2 points)
Calculez les fermetures $[X]_F^+$ des ensembles d'attributs X suivants et soulignez les ensembles qui sont des surclés de la table RATP:

Solution:

1. [station, jour]$_F^+$=(station, jour, ville)
2. [bus, station, jour]$_F^+$=(bus, station, jour, ligne, ville)
3. [nom, prenom, station, jour]$_F^+$=(nom, prenom, station, jour, tranche, bus, ville, ligne, age)
4. [bus, station, jour, tranche]$_F^+$=(bus, station, jour, tranche, ligne, ville, nom, prenom, age)