

Université Nice Sophia Antipolis
Polytech Nice Sophia

2014–2015

Contrôle continu de Base de données
12 Décembre 2014

Durée : 2 heures

1.1		1	3.1		1
1.2		1	3.2		1
1.3		1	3.3		1
1.4		1.5	3.4		1
1.5		1.5	3.5		3
2.1		1.5	3.6		2
2.2		1.5	3.7		3

Toutes vos réponses doivent être justifiées par une démonstration ou un contre-exemple. Si vous pensez que le texte d’une question est ambigu (voire erroné) faites une hypothèse raisonnable et écrivez la sur votre copie

0 Introduction

La base de données de la compagnie aérienne VEPU est composée des 4 tables

```
avion (IdAvion, TypeAvion, Portee)
calendrier(Jour, CodeVol, IdPilote, IdAvion)
pilote(IdPilote, Nompilote)
vol(CodeVol, VilleDepart, VilleArrivee, Distance, HeureDepart, HeureArrivee)
qualification (IdPilote, TypeAvion)
```

- La table avion indique pour chaque avion de la compagnie: son identifiant, son type et sa portée (c’est dire la distance maximale qu’il peut effectuer sans escale).
- La table pilote indique pour chaque pilote de la compagnie: son identifiant et son nom.
- La table vol spécifie pour chaque numéro de vol: sa ville de départ, sa ville d’arrivée, la distance du vol, l’heure de départ et l’heure d’arrivée.
- Le calendrier de la compagnie est hebdomadaire. La table calendrier précise quel vol a lieu quel jour, qui pilote ce jour-là et quel avion est utilisé. Cette compagnie n’assure que des vols courts: tous les vols arrivent le jour où ils partent. Il y a des vols tous les jours.
- la table qualification indique quel type d’avion un pilote peut piloter. Chaque pilote est autorisé à piloter au moins un type d’avion. Certains pilotes sont qualifiés pour plusieurs types d’avion.

1 Algèbre relationnelle

Exprimer en algèbre relationnelle les requêtes suivantes (i.e., construire la formule algébrique qui définit la table correspondante ou qui fournit une procédure de décision pour la question). Si la réponse vous semble impossible à exprimer en algèbre relationnelle, expliquez pourquoi.

1. Quels sont les numéros d'identification des pilotes qualifiés pour un avion de type A300 et qui ne volent pas le dimanche?

$$\Pi_{IdPilote}(\sigma_{TypeAvion="A300"}(qualification)) - \Pi_{IdPilote}(\sigma_{Jour="dimanche"}(calendrier))$$

2. Quels sont les identifiants des avions qui ne sont utilisés que pour des vol d'une distance supérieure à 1000 kms?

$$\Pi_{IdAvion}(avion) - \Pi_{IdAvion}(\sigma_{Distance < 1000}(calendrier \bowtie vol))$$

3. Quels sont les vols qui ont lieu tous les Jours?

$$\Pi_{Jour, CodeVol}(calendrier) \div \Pi_{Jour}(calendrier)$$

4. Quelles sont les villes desquelles partent et arrivent au moins un vol par Jour

$$\begin{aligned} &(\Pi_{Jour, VilleDepart}(calendrier \bowtie vol) \div \Pi_{Jour}(calendrier)) \\ &\cap (\Pi_{Jour, VilleArrivee}(calendrier \bowtie vol) \div \Pi_{Jour}(calendrier)) \end{aligned}$$

5. Quel est le type d'avion ayant la plus grand portée.

Il n'y a pas de solution simple en algèbre relationnelle faute de max, en revanche il est possible de l'exprimer en logique :

$$\{t.IdAvion : t \in avion \wedge \neg[\exists u \in avion(u.portee > t.portee)]\}$$

2 Dépendances fonctionnelles et clés

On suppose que les Dépendances fonctionnelles du schéma sont :

IdAvion \rightarrow TypeAvion, Portee

TypeAvion \rightarrow Portee

Jour, CodeVol \rightarrow IdPilote, IdAvion

IdPilote \rightarrow Nompilote

CodeVol, \rightarrow VilleDepart, VilleArrivee, Distance, HeureDepart, HeureArrivee

VilleDepart, VilleArrivee, \rightarrow Distance

1. Quelles sont alors les clés possibles?

Clés:

avion (IdAvion, TypeAvion, Portee)

calendrier(Jour, CodeVol, IdPilote, IdAvion)

pilote(IdPilote, Nompilote)

vol(CodeVol, VilleDepart, VilleArrivee, Distance, HeureDepart, HeureArrivee)

qualification (IdPilote, TypeAvion)

2. Le schéma est il alors en 3NF ? Sinon le mettre en 3NF.

Non, il n'est pas en 3NF.

- il faut décomposer la relation avion en :
avion(IdAvion, TypeAvion)
typeavion(TypeAvion, Portee)
 - il faut décomposer la relation vol en :
vol(CodeVol, VilleDepart, VilleArrivee, HeureDepart, HeureArrivee)
dist(VilleDepart, VilleArrivee, Distance)
-

3 SQL

Pour cette section, on utilisera les tables du schéma initial (avant normalisation). Pour toutes les questions, les réponses doivent être sans doublons, mais les requêtes ne doivent pas contenir de 'DISTINCT' lorsque c'est inutile.

Exprimez en SQL relationnelle les questions suivantes

1. Est ce qu'il y a des vols planifiés dans le calendrier avec des avions qui n'ont pas la portée suffisante?

```
select C.CodeVol from calendrier C, vol V
where C.CodeVol=V.CodeVol and
      V.Distance > (select A.Portee from avion A
                    where A.IdAvion = C.IdAvion);
```

2. Quels sont les noms des pilotes qui ne volent jamais sur un A300?

```
select nomPilote from pilote
where idPilote not in (
    select calendrier.IdPilote from avion, calendrier, vol
    where avion.TypeAvion='A300'
    and avion.IdAvion = calendrier.IdAvion
    and calendrier.CodeVol=vol.CodeVol);
```

3. Quels sont les vols qui ont lieu tous les Jours?

```
Select CodeVol from calendrier
except
select C3. CodeVol from
(select C1.CodeVol, C2.Jour from calendrier C1, calendrier C2
except
select C.CodeVol, C.Jour from calendrier C) as C3;
```

4. Quel est le numéro du vol le plus matinal (c'est à dire qui part avant tous les autres) sur l'ensemble des vols de la semaine? Si plusieurs vols partent à cette même heure, on affichera tous leurs codes.

```
select CodeVol from vol where HeureDepart in
      (select min(HeureDepart) from vol);
```

5. Construire la vue `premierVolDuJour(IdAvion, Jour, CodeVol)` où le `CodeVol` est le code du premier vol du jour `Jour` de l'avion qui pour identifiant `IdAvion`. Cette vue devra contenir ces vols pour tous les avions et pour tous les jours.

```
drop view if exists premierAvionDuJour;
create view premierAvionDuJour as
select C.Jour, C.IdAvion, Min(HeureDepart) as HD
      From calendrier C, vol V
      where V.CodeVol = C.CodeVol
      group by C.Jour,C. IdAvion;

drop view if exists premierVolDuJour;
create view premierVolDuJour as select distinct P.Jour, P.IdAvion, C.CodeVol
      from premierAvionDuJour P, calendrier C, vol V
      where P.HD = V.HeureDepart and C.IdAvion = P.IdAvion and C.CodeVol = V.CodeVol
      order by Jour;
select * from premierVolDuJour;
```

6. Construire la vue `premierVolDuJourComplet(IdAvion, Jour, CodeVol)` où le `CodeVol` est le code du premier vol du jour `Jour` de l'avion qui a pour identifiant `IdAvion`. Cette vue devra contenir ces vols pour tous les avions et pour tous les jours. Si un avion ne vole pas un jour donné, le `CodeVol` de cet avion aura la valeur `null`

```
drop view if exists premierVolDuJourComplet;
create view premierVolDuJourComplet as
      select IdAvion, Jour, CodeVol from
      (select distinct A.IdAvion, C.Jour from calendrier C, avion A )as B
      left join premierVolDuJour using (IdAvion, Jour);
```

7. Construire la vue `connexion (Jour,VilleDepart, VilleArrivee, NombreEtapes, HeureDepart, HeureArrivee)` qui indique toutes les connexions possibles avec `NombreEtapes` sur une seule Journée à partir de `VilleDepart` à `HeureDepart` vers une ville `Arrivee` à `HeureArrivee` en n'empruntant que des vols de la compagnie VEPU, qui est si efficace que l'on peut enchaîner deux vols à condition que l'heure de départ du second soit supérieure à l'heure d'arrivée du premier (et bien sur la ville de départ du second vol est la ville d'arrivée du premier).

```
drop view if exists Connexions;
create view Connexions as
```

```

with recursive connexion (Jour,VilleDepart, VilleArrivee,
                          NombreEtapes, HeureDepart, HeureArrivee) as (
  select Jour, VilleDepart, VilleArrivee, 1, HeureDepart, HeureArrivee
    from vol, calendrier where vol.CodeVol=calendrier.CodeVol
  union
  select C.Jour, connexion.VilleDepart, V.VilleArrivee,
        NombreEtapes+1, connexion.HeureDepart, V.HeureArrivee
    from vol V,calendrier C,connexion
   where  V.CodeVol=C.CodeVol and
  connexion.VilleArrivee=V.VilleDepart
  and connexion.HeureArrivee < V.HeureDepart
  and C.Jour=connexion.Jour
)

select Jour, VilleDepart, VilleArrivee, NombreEtapes, HeureDepart, HeureArrivee from connexion ;

```
