

# Bases de Données Relationnelles

## Examen du 17 Janvier 2005

### ESSI 2

December 12, 2007

#### 1 Exercice 1

La jointure et la projection étant des opérations ensemblistes, les occurrences multiples sont éliminées. On a  $R \bowtie S = \{(a, b, c), (a, b, c'), (a', b, c), (a', b, c'), (a, b', c)\}$  et  $\Pi_{A,C} R \bowtie S = \{(a, c), (a, c'), (a', c), (a', c')\}$

#### 2 Exercice 2

Par réflexivité on a  $A \rightarrow A$  et  $B \rightarrow B$ .

$A \rightarrow A$  et  $A \rightarrow C$  donnent  $A \rightarrow AC$  par addition, puis  $AB \rightarrow AC$  par augmentation. De même  $B \rightarrow B$  et  $B \rightarrow D$  donnent  $B \rightarrow BD$  par addition, puis  $AB \rightarrow BD$  par augmentation.

$AB \rightarrow AC$  et  $AB \rightarrow BD$  donnent  $AB \rightarrow ABCD$ , par addition. On en déduit que  $\{A, B\}$  est une super clé de la relation.

Si l'on calcule la fermeture de  $\{A\}$ , on obtient  $\{A, C\}$ . On n'a donc pas  $A \rightarrow ABCD$ . On en déduit que  $\{A\}$  n'est pas une clé de la relation.

Si l'on calcule la fermeture de  $\{B\}$ , on obtient  $\{B, D\}$ . On n'a donc pas  $B \rightarrow ABCD$ . On en déduit que  $\{B\}$  n'est pas une clé de la relation.

On sait maintenant que  $\{A, B\}$  est une clé de la relation.

$\{A, B, C, D\}$  est une super clé mais pas une clé.

#### 3 Exercice 3

Si une clé contient  $E$ , elle ne doit pas contenir  $B, C$  ou  $D$  (sinon on pourrait avoir une clé plus petite en supprimant  $B, C$  ou  $D$ ).  $E$  tout seul ne constitue pas une clé.  $EA$  est une clé, et c'est la seule qui contient  $E$ .

Si une clé ne contient pas  $E$ , alors elle doit contenir  $ABC$  puisque  $ABC \rightarrow DE$  est la seule dépendance fonctionnelle ayant  $E$  en partie droite.  $ABC$  est une clé, et c'est la seule qui ne contient pas  $E$ .

Cette relation admet donc deux clés :  $ABC$  et  $EA$ .

Si l'on suppose que  $A, B, C, D$  et  $E$  sont bien atomiques, alors la relation est en première forme normale. Elle n'est pas en seconde forme normale, car  $D$  attribut non clé (c'est ici le seul attribut non clé) dépend fonctionnellement de  $E$  qui est une sous clé. La relation n'est donc à fortiori pas non plus en troisième forme normale, ni en forme normale de Boyce-Codd.

#### 4 Exercice 4

Pour  $DF_1, DF_2$  et  $DF_3$ , on a  $\{A\}^+ = \{A, B, C\}$ . Pour  $DF_1$ , on a  $\{B\}^+ = \{B, C\}$ , pour  $DF_2$  et  $DF_3$ , on a  $\{B\}^+ = \{B\}$ .

$DF_1$  et  $DF_3$  ne sont pas équivalentes, par exemple pour la relation  $\{(a_1, b_1, c_1), (a_2, b_1, c_2)\}$   $DF_1$  est faux mais  $DF_3$  est vrai.

$DF_2$  et  $DF_3$  sont équivalentes:

Supposons  $DF_2$  vérifiée, alors  $A \rightarrow C$  donne  $AB \rightarrow C$  par augmentation et donc  $DF_3$  est vérifiée. Supposons  $DF_3$  vérifiée, alors  $AB \rightarrow C$  et  $A \rightarrow B$  donne  $A \rightarrow C$  par pseudo transitivité et donc  $DF_2$  est vérifiée.

## 5 Exercice 5

```
--5.1
create view equipes_fr as (select nom from equipe where pays = 'fr');
create view equipes_es as (select nom from equipe where pays = 'es');

select date from match m
    where (m.nom1 in (select * from equipes_fr) and m.nom2 in (select * from equipes_es))
        or (m.nom2 in (select * from equipes_fr) and m.nom1 in (select * from equipes_es));

--5.2

select distinct e.pays from equipe e
    where e.nom in (select m.nom2 from match m where m.nom1= 'rm' and buts2 > buts1);

--5.3

create view moyenne_buts_par_equipe as
    select m.nom2 as Nom , AVG(m.buts2) as Moyenne from match m group by m.nom2;
select nom, moyenne from moyenne_buts_par_equipe
    where moyenne = (select Max(moyenne) from moyenne_buts_par_equipe);

--5.4

select e.pays, sum(m.buts2) from equipe e, match m where e.nom = m.nom2 and m.nom1='bm'
    group by e.pays
    having sum(m.buts2) >=
        ALL (select sum(m.buts2) from equipe e, match m
            where e.nom = m.nom2 and m.nom1='bm'
            group by e.pays);
```

## 6 Exercice 6

1. Ecrire les requêtes suivantes en algèbre relationnelle :

- Quelle année est ce que Leonard de Vinci a peint la Joconde ?  
 $\Pi_{annee}[\sigma_{nom=joconde,artiste=Leonard\ de\ Vinci}(peintures)]$
- Dans quelle ville est ce que la Joconde est exposée?  
 $\Pi_{ville}[\Pi_{musee}(\sigma_{nom=joconde}(peintures) \bowtie \delta_{nom \leftarrow musee}(musee))]$
- Quels sont les villes qui ont des musées qui abritent des oeuvres peintes par leur conservateur  
 $\Pi_{ville}[peintures \bowtie \delta_{nom \leftarrow musee,conservatoire \leftarrow artiste}(musee)]$

2. Ecrire les requêtes suivantes en SQL

- Quels sont les conservateurs des musées qui possèdent au moins 10 oeuvres de Leonard de Vinci ?

```
select m.conservateur from musee m where m.nom in
    (select musee from peintures where artiste = 'Léonard de Vinci'
    group by musee having count(*) >= 10);
```

- Quel est par pays le nombre moyen de peintures peintes par des artistes de ce pays exposé dans les musées du pays ?

```
create view nb_peintures_par_peintre_exposees_dans_musee_pays as
    (select m.nom, m.pays, count(*) as nb from musee m, peintures p, peintres a
    where m.pays=a.pays and p.musee=m.nom and p.artiste=a.nom
```

```
select pays, avg(nb) from nb_peintures_par_peintre_exposees_dans_musee_pays
group by pays;
```

### 3. Créer cette base de données en SQL

```
--6.3
-- comme les dépendances fonctionnelles n'ont pas été explicitées on ne définira
-- pas de clés et on suposera aussi qu'il n'existe pas d'homonymes

drop table peintures cascade;
create table peintures (
    nom varchar(20),
    artiste varchar(20),
    annee integer,
    musee varchar(20)
);
create table musee (
    nom varchar(20),
    ville varchar(20),
    pays varchar(20),
    conservateur varchar(20)
);

create table peintres (
    nom varchar(20),
    pays varchar(20),
    annee_naissance integer,
    annee_deces integer,
    -- contrainte a) ... on ne gère pas les artistes en herbes ;)
    check (annee_naissance < annee_deces)
);

-- contrainte b
select * from peintures t, peintres a
where t.artiste = a.nom and
      not (t.annee > a.annee_naissance and t.annee <= a.annee_deces);

-- contrainte c
-- vérifiée par construction : l'ensembles des oeuvres d'un artiste se trouvent
-- dans la table peinture dont le champ musée peu être null
```

## 7 Exercice 7

Est ce que les deux requêtes JDBC suivantes donnent le même résultat ? Justifiez votre réponse

- Différence entre /\* Requête 1 \*/ /\* Requête 2 \*/:  
Oui, si *a* est de type entier. Dans les autres cas le comportement peut être différent car une coercition des types peut être effectuée par SQL (e.g., SQL accepte de comparer un entier avec un flottant).
- Utilité de `resultset`:  
`resultset` est indispensable pour la gestion des réponses multiples que SQL est susceptible d'envoyer. Dans un langage classique, un appel de fonction ne retourne en général qu'une réponse.