Université Nice Sophia Antipolis
Polytech Nice Sophia

2013–2014

Interrogation de Base de données
3 Décembre 2013

Durée : 30 minutes – **sans documents**

1.1		4
1.2		3
1.3		3
2.1		4
2.2		2
3		4

1 Normalisation

1. On considère la relation suivante
   $r(A,B,C,D,E,F)$
   On suppose que les dépendances fonctionnelles sur cette relation peuvent toutes être déduites de l'ensemble $DF = \{A \rightarrow BD;\ D \rightarrow C;\ B \rightarrow E;\ E \rightarrow F;\ BC \rightarrow A\}$
   Déterminer la fermeture transitive de chacun des attributs et **toutes** les clés de la relation.

   $\{A\}^+ = \{A,B,C,D,E,F\}$ donc A est une clé. $\{B\}^+ = \{B,E,F\}$, $\{C\}^+ = \{C\}$, $\{D\}^+ = \{C,D\}$, $\{E\}^+ = \{E,F\}$ et $\{F\}^+ = \{F\}$, il n'y a donc pas d'autres clés composées d'un seul attribut
   BC et BD sont deux autres clés car leur fermeture transitive est égal à l'ensemble de tous les attributs, et ni B, ni C ni D ne constituent une clé .

   ---

2. On considère la relation suivante
   $r(A,B,C,D,E,F)$
   On suppose que les dépendances fonctionnelles sur cette relation peuvent toutes être déduites de l'ensemble $DF = \{A \rightarrow CE;\ B \rightarrow D;\ C \rightarrow BD;\ D \rightarrow AE;\ E \rightarrow F\}$. Cette relation admet donc deux clés, AB et CD. Donner **toutes** les dépendances fonctionnelles de DF qui violent la 2NF et mettre la relation en 2NF en appliquant l'algorithme de normalisation vu en cours.

   Seuls E et F n'appartiennent a aucune clé. Les dépendances fonctionnelles qui violent la $2NF$ sont donc $A \rightarrow EF$ et $D \rightarrow EF$. Pour mettre la relation en $2NF$, on la décompose donc en $r_1(A,E,F)$ et $r_2(A,B,C,D)$ r_2 est en $2NF$ (et même en $3NF$) car il n'y a plus d'attributs non clés. r_1 admet A comme clé, et est donc en $2NF$, mais pas en $3NF$

   ---

3. On considère la relation suivante
   $r(A,B,C,D,E)$
   On suppose que les dépendances fonctionnelles sur cette relation peuvent toutes être déduites de l'ensemble $DF = \{A \rightarrow BD;\ B \rightarrow C;\ D \rightarrow E\}$. Cette relation admet donc A comme clé. Mettre cette relation en $3NF$ en appliquant l'algorithme vu en cours.

Cette relation est en $2NF$, mais pas en $3NF$. La dépendance fonctionnelle $B \rightarrow C$ viole la $3NF$ puisque ni B ni C ne font partie d'une clé. On décompose donc r en $r_1(B,C)$(qui est en $3NF$ et $r_2(A,B,D,E)$ qui n'est pas en $3NF$ a cause de la dépendance fonctionnelle $D \rightarrow E$. On décompose donc r_2 en deux relations$r_{21}(D,E)$ (en $3NF$) et $r_{22}(A,B,D)$ (en $3NF$).

2 Récursivité

On considère la table suivante :

```
CREATE TABLE Employes
    (   Nom VARCHAR(30),
        Prenom VARCHAR(30),
        DateDeNaissance DATE,
        Adresse VARCHAR(30),
        NumSS INTEGER PRIMARY KEY,
        Salaire INTEGER,
        NumeroDepartement INTEGER,
        Superieur INTEGER REFERENCES Employes  );
```

1. Rechercher les noms de tous les employés dont le supérieur hiérachique (direct ou indirect) est Juliette Rochat

```
WITH RECURSIVE sups_jr(Nom, NumSS) AS
 (  SELECT Nom, NumSS FROM Employes E1 WHERE E1.Superieur IN
      (SELECT E2.NumSS FROM Employes E2
            WHERE E2.Nom='Rochat' AND E2.Prenom = 'Juliette')
    UNION ALL
    SELECT E3.Nom,E3.NumSS FROM Employes AS E3, sups_jr AS S
             WHERE S.NumSS =E3.Superieur )
SELECT * FROM sups_jr;
```

2. Rechercher les noms de tous les employés dont le supérieur hiérachique est Juliette Rochat et qui sont plus agés qu'elle.

```
CREATE VIEW oldsups_jul(Nom, NumSS, DateDeNaissance)
 AS WITH RECURSIVE oldsups_jr(Nom, NumSS, DateDeNaissance) AS
       (SELECT Nom, NumSS, DateDeNaissance FROM Employes E1 WHERE E1.Superieur IN
             (SELECT E2.NumSS FROM Employes E2
                 WHERE E2.Nom='Rochat' AND E2.Prenom = 'Juliette')
       UNION ALL
       SELECT E3.Nom,E3.NumSS,E3.DateDeNaissance FROM Employes AS E3, oldsups_jr AS S
             WHERE S.NumSS =E3.Superieur )
 SELECT * FROM oldsups_jr;

SELECT Nom AS Olds FROM oldsups_jul E1
   WHERE E1.DateDeNaissance < (SELECT E2.DateDeNaissance FROM Employes E2
                                  WHERE E2.Nom='Rochat' AND E2.Prenom = 'Juliette');
```

3 Jointures

On considère les table suivantes :

```
select * from t1;
 num | nom
-----+-----
   1 | a
   2 | b
   3 | x
```

```
select * from t2;
 num | nom
-----+-----
   1 | a
   2 | b
   3 | y
```

Donner le résultat des requêtes suivantes:

```
(1)   SELECT * FROM t1 NATURAL JOIN t2;
(2)   SELECT * FROM t1 LEFT JOIN t2 USING(num,nom);
(3)   SELECT * FROM t1 RIGHT JOIN t2 USING(num,nom);
(4)   SELECT * FROM t1 FULL JOIN t2 USING(num,nom);
```

```
 num | nom
-----+-----
   1 | a
   2 | b
(2 rows)
```

```
 num | nom
-----+-----
   1 | a
   2 | b
   3 | x
(3 rows)
```

```
 num | nom
-----+-----
   1 | a
   2 | b
   3 | y
(3 rows)
```

```
 num | nom
-----+-----
   1 | a
   2 | b
   3 | x
   3 | y
(4 rows)
```
