

Epreuve de Bases de Données Relationnelles

19 Décembre ( Durée : deux heures)

Tous documents autorisés

## 1 Algèbre relationnelle & Normalisation

### 1.1 Algèbre relationnelle (2 points)

On considère la base de données composée des trois relations suivantes:

**Frequente**(Buveur,Bar)

**Sert**(Bar,Bière)

**Aime**(Buveur,Bière)

La relation **Frequente** indique les bars que chaque buveur fréquente; la relation **Sert** indique les bières servies dans chaque bar; la relation **Aime** nous donne la ou les bières préférées de chaque buveur. On suppose que chaque bar sert au moins une bière.

Exprimer en algèbre relationnelle la requête suivante : *Les buveurs qui fréquentent tous les bars*

---

Il s'agit typiquement d'une division:  $Frequente(Buveur,Bar) \div \Pi_{Bar}(Sert(Bar,Biere))$ , soit  $\Pi_{Buveur}Frequente(Buveur,Bar) - \Pi_{Buveur}(\Pi_{Buveur}Frequente(Buveur,Bar) \bowtie \Pi_{Bar}Sert(Bar,Biere) - Frequente(Buveur,Bar))$

---

### 1.2 Normalisation (5 points)

On donne la relation :  $R(A,B,C,D,E,F)$  et l'ensemble de dépendances fonctionnelles :

1.  $A \rightarrow E$
2.  $B \rightarrow D$
3.  $C \rightarrow A$
4.  $D \rightarrow E$
5.  $E \rightarrow B$

– Déterminer la fermeture transitive de chaque attribut et déterminer toutes les clés de  $R$  (2 points).

- 
- $A^+ = \{A,B,D,E\}$
  - $B^+ = \{B,D,E\}$
  - $C^+ = \{A,B,C,D,E\}$
  - $D^+ = \{B,D,E\}$
  - $E^+ = \{B,D,E\}$
  - $F^+ = \{F\}$

La seule clé est  $\{C,F\}$

---

- Quelles sont les contraintes qui violent la 2NF? la 3NF? (1 point)

---

La contrainte 3 viole la 2NF, les contraintes 1,2, 4 et 5 violent la 3NF

---

- Décomposer R en table sous forme 3NF en utilisant l'algorithme vu en cours, en indiquant toutes les étapes (2 points)

---

On commence par mettre la table en 2NF.

La contrainte 3 nous conduit à introduire une relation  $R_1 = (\underline{C}, A)$  dont la seule clé est  $C$  et qui est en 3NF et une relation  $R_2 = (\underline{C, F}, B, D, E)$  dont la clé est  $CF$  et qui n'est toujours pas en 3NF.

On décompose donc  $R_2$  en deux relations  $R_3 = (\underline{C, F}, B, D)$  qui n'est pas en 3NF et  $R_4 = (\underline{D}, E)$  qui est en 3NF. Il reste à décomposer  $R_3$  en deux relations  $R_5 = (\underline{C, F}, B)$  et  $R_6 = (\underline{B}, D)$  qui sont toutes les deux en 3NF

---

## 2 SQL

### 2.1 Requêtes avec agrégats (5 points)

Soit la table suivante qui contient pour chaque employé son numéro du département, son nom et son salaire:

```
CREATE table employees
(
  department INT,
  empname VARCHAR(10),
  salary INT
);
```

1. Écrire une requête qui affiche les numéro de département et le salaire minimum du département pour les départements dont le salaire minimum est strictement supérieur à 15 000.

---

```
SELECT department, MIN(salary) as "Lowest salary"
FROM employees
GROUP BY department
HAVING MIN(salary) > 15000;
```

---

2. Écrire une requête qui affiche les numéro de département, le nombre d'employés qui ont un salaire > 75 000 pour les départements qui ont plus de 3 employés avec un salaire strictement supérieur à 75 000.

---

```
SELECT department, COUNT(*) as "Number of employees"
FROM employees
WHERE salary > 75000
GROUP BY department
HAVING COUNT(*) > 3;
```

---

## 2.2 Jointures ( 3 points)

Soit la table *vente* ci-dessous qui contient pour chaque département, le nom du vendeur et le montant de ses ventes:

```
CREATE table vente
(
  department INT,
  vendeur VARCHAR(10),
  montant INT
);
```

En utilisant la table *employees* ci-dessus et la table *vente*, écrire une requête qui affiche pour chaque département le nom des employés et leurs salaire, le nom des vendeurs et le montant des ventes. Si le vendeur est un salarié, alors ces informations ne sont affichées qu'une seule fois (on suppose qu'il n'y a pas d'homonymes).

Soit, une table de la forme:

department	empname	salary	vendeur	montant
1	alpha	76000	alpha	1760000
1	gamma	15000	gamma	15000
2	epsilon	16000		
			lambda	20000

```
SELECT employees.department, empname, salary, vendeur, montant
FROM employees FULL OUTER JOIN ventes
ON employees.department=ventes.department AND empname=vendeur ;
```

## 2.3 Création de tables (5 points)

Soient les relations suivantes:

- $r_1(\underline{A}, C, D)$
- $r_2(\underline{C}, F, G)$
- $r_3(\underline{B}, \underline{E}, H, I)$
- $r_4(\underline{A}, C)$

Les attributs soulignés sont des clés primaires. A,B, C et I sont des entiers. C est une séquence qui commence à 1. Les autres attributs sont des chaînes d'au plus 10 caractères. Les attributs G et H ne peuvent pas être nuls. La valeur par défaut de I est 0. Les valeurs des couples (H,I) doivent tous être différentes. Lorsque le même nom d'attribut figure dans plusieurs relations, il s'agit du même attribut.

1. Écrire le script de création de ces tables qui garantit que toutes ces contraintes sont respectées (et sans imposer davantage de contraintes).

```
DROP TABLE IF EXISTS r3;
DROP TABLE IF EXISTS r4;
DROP TABLE IF EXISTS r1;
```

```
DROP TABLE IF EXISTS r2;
```

```
CREATE table r2
(
  C SERIAL PRIMARY KEY,
  F VARCHAR(10),
  G VARCHAR(10) NOT NULL
);
```

```
Create Table r1
(
  A INT PRIMARY KEY,
  C INTEGER REFERENCES r2,
  D VARCHAR(10)
);
```

```
CREATE Table r3
(
  B INTEGER,
  E VARCHAR(10),
  H VARCHAR(10),
  I INT DEFAULT 0,
  CONSTRAINT pk1 PRIMARY KEY(B,E),
  CONSTRAINT tousdiff UNIQUE (H,I)
);
```

```
CREATE table r4
(
  A INT REFERENCES r1,
  C INT REFERENCES r2,
  CONSTRAINT pk2 PRIMARY KEY(A,C)
);
```

- 
2. Donner les instructions qu'il faut ajouter à ce script pour que l'utilisateur *invited* puisse insérer des tuples dans la table  $r_1$  et qu'il puisse transmettre ces droits.
- 

```
GRANT INSERT ON r1 TO invited WITH GRANT OPTION;
GRANT REFERENCES ON r2 TO invited WITH GRANT OPTION;
```

---