Département Licence	Année : 2006/2007 Session de janvier 2007 **PARCOURS :** INF 3 **UE :** INF 109, Bases de données **Épreuve :** INF 109 EX **Date :** Vendredi 5 janvier 2007 **Heure :** 8 heures 30 **Durée :** 1 heure 30 Documents : non autorisés Épreuve de M. Alain Griffault

SUJET + CORRIGE

Avertissement

- La plus part des questions sont indépendantes.
- Le barème total est de 25 points car le sujet est assez long.
- Le barème est (approximativement) proportionnel à la difficulté de la question.
- L'espace laissé pour les réponses est suffisant (sauf si vous utilisez ces feuilles comme brouillon, ce qui est fortement déconseillé).

Le sujet

Le sujet porte sur une gestion simplifiée de cours et de salles.
Soit `EmploiDuTemps (Salle, TypeSalle, NbPlaces, Jour, Groupe, Matiere, Enseignant)` une relation et ses dépendances fonctionnelles :

- {`Salle`} $\longrightarrow$ {`TypeSalle, NbPlaces`} qui indique que le numéro de la salle détermine son nombre de places et son type.
- {`TypeSalle`} $\longrightarrow$ {`NbPlaces`} qui indique que le type d'une salle détermine son nombre de places.
- {`Enseignant`} $\longrightarrow$ {`Matiere`} qui indique qu'un enseignant ne donne des cours que dans une seule matière.
- {`Salle, Jour`} $\longrightarrow$ {`Groupe, Matiere`} qui indique qu'une salle est occupée pendant toute une journée par un groupe unique et pour une seule matière.
- {`Groupe, Matiere`} $\longrightarrow$ {`Enseignant`} qui indique que pour les élèves d'un groupe, la matière détermine l'enseignant.

Les principales simplifications concernent la notion de créneau horaire qui correspond à une journée, et les enseignants qui ne font cours que dans une seule matière. Par exemple :

Salle	TypeSalle	NbPlaces	Jour	Groupe	Matiere	Enseignant
Phy A2	Amphi	250	5 janvier 2007	INF3	Bases de donnees	Griffault

spécifie que vous êtes en train de lire cette phrase.

Exercice 1 (Requêtes SQL sur *EmploiDuTemps* (12 points))

Question 1.1 (1 point) *Après en avoir donné une écriture algébrique, écrire une requête SQL qui caractérise les* ***`Enseignant`*** *ayant donné au moins un cours en 2006.*

Réponse :

$$R \quad = \quad \pi[\textit{Enseignant}](\sigma[\textit{Jour} > \textit{2006-1-1} \ \wedge \ \textit{Jour} \leq \textit{2006-12-31}](\textit{EmploiDuTemps}))$$

```
SELECT DISTINCT Enseignant
FROM            EmploiDuTemps
WHERE           Jour BETWEEN '2006-1-1' AND '2006-12-31';
```

Question 1.2 (1 point) *Écrire une requête SQL qui caractérise les* ***`Enseignant`*** *ayant donné au moins un cours dans deux types de salle différents.*

Réponse :

```
-- Algebre relationnelle
SELECT DISTINCT EDT1.Enseignant
FROM            EmploiDuTemps AS EDT1, EmploiDuTemps AS EDT2
WHERE           EDT1.Enseignant = EDT2.Enseignant
  AND           EDT1.TypeSalle <> EDT2.TypeSalle;
-- Calcul relationnel
SELECT DISTINCT Enseignant
FROM            EmploiDuTemps AS EDT1
WHERE EXISTS    (SELECT *
                 FROM   EmploiDuTemps AS EDT2
                 WHERE  EDT1.Enseignant = EDT2.Enseignant
                 AND    EDT1.TypeSalle <> EDT2.TypeSalle);
-- Calcul relationnel
SELECT DISTINCT Enseignant
FROM            EmploiDuTemps AS EDT1
WHERE NOT (TypeSalle = ALL (SELECT TypeSalle
                            FROM   EmploiDuTemps AS EDT2
                            WHERE  EDT1.Enseignant = EDT2.Enseignant));
-- Utilisation des agregas
SELECT DISTINCT Enseignant, COUNT(TypeSalle)
FROM (SELECT DISTINCT Enseignant, TypeSalle FROM EmploiDuTemps) AS R
GROUP BY        Enseignant
HAVING          COUNT(TypeSalle) > 1;
```

Question 1.3 (1 point) *Écrire une requête SQL qui caractérise les* **`Groupe`** *qui ne savent pas quoi faire lorsqu'ils doivent être dans deux salles différentes le même jour.*

Réponse :

```
-- Algebre relationnelle
SELECT DISTINCT EDT1.Groupe
FROM            EmploiDuTemps AS EDT1, EmploiDuTemps AS EDT2
WHERE           EDT1.Groupe = EDT2.Groupe
  AND           EDT1.Jour = EDT2.Jour
  AND           EDT1.Salle <> EDT2.Salle;

-- Calcul relationnel
SELECT DISTINCT Groupe
FROM            EmploiDuTemps AS EDT1
WHERE EXISTS    (SELECT *
                 FROM   EmploiDuTemps AS EDT2
                 WHERE  EDT1.Groupe = EDT2.Groupe
                 AND    EDT1.Jour = EDT2.Jour
                 AND    EDT1.Salle <> EDT2.Salle);
-- Calcul relationnel
SELECT DISTINCT Groupe
FROM            EmploiDuTemps AS EDT1
WHERE NOT (Salle = ALL (SELECT Salle
                        FROM   EmploiDuTemps AS EDT2
                        WHERE  EDT1.Groupe = EDT2.Groupe
                        AND    EDT1.Jour = EDT2.Jour));
-- Utilisation des agregas
SELECT DISTINCT Groupe, COUNT(Salle)
FROM (SELECT DISTINCT Groupe, Jour, Salle FROM EmploiDuTemps) AS R
GROUP BY        Groupe, Jour
HAVING          COUNT(Salle) > 1;
```

Question 1.4 (1 point) *Écrire une requête SQL qui caractérise les* **`Enseignant`** *qui ne savent pas quoi faire lorsqu'ils doivent être dans deux salles différentes le même jour.*

Réponse :

```
-- Algebre relationnelle
SELECT DISTINCT EDT1.Enseignant
FROM            EmploiDuTemps AS EDT1, EmploiDuTemps AS EDT2
WHERE           EDT1.Enseignant = EDT2.Enseignant
  AND           EDT1.Jour = EDT2.Jour
  AND           EDT1.Salle <> EDT2.Salle;
-- Calcul relationnel
SELECT DISTINCT Enseignant
FROM            EmploiDuTemps AS EDT1
WHERE EXISTS    (SELECT *
                 FROM   EmploiDuTemps AS EDT2
                 WHERE  EDT1.Enseignant = EDT2.Enseignant
                 AND    EDT1.Jour = EDT2.Jour
                 AND    EDT1.Salle <> EDT2.Salle);
-- Calcul relationnel
SELECT DISTINCT Enseignant
FROM            EmploiDuTemps AS EDT1
WHERE NOT (Salle = ALL (SELECT Salle
                        FROM   EmploiDuTemps AS EDT2
                        WHERE  EDT1.Enseignant = EDT2.Enseignant
                        AND    EDT1.Jour = EDT2.Jour));
-- Utilisation des agregas
SELECT DISTINCT Enseignant, COUNT(Salle)
FROM (SELECT DISTINCT Enseignant, Jour, Salle FROM EmploiDuTemps) AS R
GROUP BY        Enseignant, Jour
HAVING          COUNT(Salle) > 1;
```

Question 1.5 (2 points) *Donnez deux contraintes SQL entre les attributs afin que les deux requêtes précédentes retournent toujours un ensemble vide.*

Réponse :

```
UNIQUE(Enseignant,Jour),
UNIQUE(Groupe,Jour),
```

Question 1.6 (2 points) *En supposant que :*
*- {**Salle, Jour**} soit une clef candidate ;*
- les salles 346 et 213 soient déjà présentes dans la base,
- la salle 346 soit utilisée le 8 mars 2006 ;
écrire une requête SQL qui ajoute l'enregistrement : "le 7 mars 2007, la salle 213 sera utilisée comme la salle 346 l'était le 8 mars 2006."

Réponse : *La présence de la salle 213 permet de récupérer le type de salle et le nombre de places, et l'occupation de la salle 346 le 8 mars 2006 permet de récupérer le groupe, la matière et l'enseignant.*

```
INSERT INTO EmploiDuTemps
  SELECT Salle, TypeSalle, NbPlaces, '2007-3-7', Groupe, Matiere, Enseignant
  FROM   (SELECT          DISTINCT Salle, TypeSalle, NbPlaces
          FROM            EmploiDuTemps
          WHERE           Salle = 213) AS TEMP1,
         (SELECT          DISTINCT Groupe, Matiere, Enseignant
          FROM            EmploiDuTemps
          WHERE           Salle = 346
          AND             Jour = '2006-3-8') AS TEMP2;
```

Question 1.7 (2 points) *Traduisez l'expression algébrique suivante dans laquelle **Ens** est une abréviation de **Enseignant** et **EDT** de **EmploiDuTemps***

$$R = \pi[\mathbf{Ens}](EDT) - \pi[\mathbf{EDT1.Ens}](\sigma[\mathbf{EDT1.Ens} = \mathbf{EDT2.Ens} \wedge \mathbf{EDT1.Salle} \neq \mathbf{EDT2.Salle}](\alpha[\mathbf{EDT:EDT1}] \times \alpha[\mathbf{EDT:EDT2}]))$$

en une requête SQL, puis expliquez ce qu'elle calcule.

Réponse :

```
-- Algebre relationnelle
SELECT Enseignant
FROM   EmploiDuTemps
  EXCEPT
SELECT EDT1.Enseignant
FROM   EmploiDuTemps AS EDT1, EmploiDuTemps AS EDT2
WHERE  EDT1.Enseignant = EDT2.Enseignant
AND    EDT1.Salle <> EDT2.Salle;
-- Calcul relationnel
SELECT Enseignant
FROM   EmploiDuTemps AS EDT1
WHERE  NOT EXISTS
       (SELECT *
        FROM   EmploiDuTemps AS EDT2
        WHERE  EDT1.Enseignant = EDT2.Enseignant
        AND    EDT1.Salle <> EDT2.Salle);
```

La requête SQL caractérise les ***Enseignant*** *qui font toujours cours dans la même salle.*

Question 1.8 (2 points) *Écrire une requête SQL qui caractérise les* ***Enseignant*** *ayant fait cours dans toutes les salles.*

Réponse :

```
-- Algebre relationnelle
SELECT DISTINCT Enseignant
FROM   EmploiDuTemps
  EXCEPT
SELECT  Enseignant
FROM (
  SELECT *
  FROM   (SELECT DISTINCT Enseignant FROM   EmploiDuTemps) AS PiR1,
         (SELECT DISTINCT Salle FROM EmploiDuTemps) AS R2
    EXCEPT
  SELECT Enseignant, Salle
  FROM   EmploiDuTemps
) AS NonEntierR1;
-- Calcul relationnel
SELECT DISTINCT Enseignant
FROM   EmploiDuTemps AS EDT1
WHERE  NOT EXISTS
       (SELECT DISTINCT Salle
         FROM  EmploiDuTemps AS EDT2
         WHERE  NOT EXISTS
                (SELECT DISTINCT Salle
                 FROM   EmploiDuTemps AS EDT11
                 WHERE  EDT11.Enseignant  = EDT1.Enseignant
                 AND    EDT11.Salle  = EDT2.Salle));
```

Exercice 2 (Normalisation (7 points)) *Le schéma conceptuel de la relation* ***EmploiDuTemps*** *comprend :*
– les dépendances fonctionnelles définies dans le sujet.
– les deux contraintes définies à la question 1.5.

Question 2.1 (1 point) *Donnez toutes les clefs candidates de la relation* ***EmploiDuTemps****.*

Réponse :
– Les dépendances fonctionnelles donnent : $C_1 = \{$***Salle, Jour***$\}$
– Les deux contraintes donnent : $C_2 = \{$***Enseignant, Jour***$\}$ *et* $C_3 = \{$***Groupe, Jour***$\}$

Question 2.2 (1 point) *Même si l'on suppose qu'il n'y a aucun doublon dans* ***EmploiDuTemps****, justifiez pourquoi la relation* ***EmploiDuTemps*** *n'est pas en troisième forme normale.*

Réponse : *Une seule des explications suivantes est suffisante (liste non exhaustive).*

Non 2NF : *La clef* {*Salle, Jour*} *contient* {*Salle*} *qui détermine* {*TypeSalle*}.

Non 3NF : *La clef* {*Salle, Jour*} *et* {*TypeSalle*} $\longrightarrow$ {*NbPlaces*}.

Non 3NF : *La clef* {*Salle, Jour*} *et* {*Enseignant*} $\longrightarrow$ {*Matiere*}.

Question 2.3 (2 points) *Appliquez un algorithme (ou une technique) de normalisation pour obtenir une décomposition, sans perte d'information, de la relation* ***EmploiDuTemps*** *en un ensemble de relations au moins en troisième forme normale. Vous n'écrirez sur la copie que les nouvelles relations et les dépendances fonctionnelles qui sont à la base des projections effectuées.*

Réponse : *Décomposition en 3NF :*
- *ListeSalles (Salle, TypeSalle) qui provient de* {*Salle*} $\longrightarrow$ {*TypeSalle*}
- *ListeTypesSalle (TypeSalle, NbPlaces) qui provient de* {*TypeSalle*} $\longrightarrow$ {*NbPlaces*}
- *Seances3NF (Salle, Jour, Groupe, Matiere) qui provient de* {*Salle, Jour*} $\longrightarrow$ {*Groupe, Matiere*}
- *ListeCours (Groupe, Matiere, Enseignant) qui provient de* {*Groupe, Matiere*} $\longrightarrow$ {*Enseignant*}

Décomposition en BCNF :
- *ListeSalles (Salle, TypeSalle) qui provient de* {*Salle*} $\longrightarrow$ {*TypeSalle*}
- *ListeTypesSalle (TypeSalle, NbPlaces) qui provient de* {*TypeSalle*} $\longrightarrow$ {*NbPlaces*}
- *SeancesBCNF (Salle, Jour, Groupe, Enseignant) qui provient de* {*Salle, Jour*} $\longrightarrow$ {*Groupe, Enseignant*}
- *ListeEnseignants (Enseignant, Matiere) qui provient de* {*Enseignant*} $\longrightarrow$ {*Matiere*}

Question 2.4 (3 points) *Après avoir précisé si votre décomposition est en BCNF ou bien seulement en 3NF, répondez aux questions qui vous concernent.*

Votre décomposition est en BCNF *:*

1. *Indiquez la dépendance fonctionnelle que vous avez perdue.*
2. *Indiquez comment s'expriment les deux contraintes de la question 1.5.*
3. *Donnez une requête d'insertion possible pour une de vos relations qui viole une dépendance fonctionnelle de la relation* ***EmploiDuTemps****.*

Votre décomposition est seulement en 3NF *:*

1. *Indiquez le problème de redondance qui subsiste.*
2. *Indiquez comment s'expriment les deux contraintes de la question 1.5.*
3. *Donnez une requête d'insertion qui nécessite que le SGBD évalue une contrainte inter-relations pour déterminer si elle est possible ou non.*

Réponse :

Décomposition en BCNF *:*

1. {*Groupe, Matiere*} $\longrightarrow$ {*Enseignant*} *qui indique que pour les élèves d'un groupe, la matière détermine l'enseignant est perdue.*
2. *Les deux contraintes* `UNIQUE(Enseignant,Jour)`,`UNIQUE(Groupe,Jour)` *sont devenues des contraintes de la relation* ***Seances****.*
3. *Les insertions suivantes sont possibles :*

```
INSERT INTO ListeEnseignants VALUES ('Bianchi', 'Bases de donnees');
INSERT INTO ListeEnseignants VALUES ('Griffault', 'Bases de donnees');
INSERT INTO SeancesBCNF VALUES ('113', '2007-1-20', 'GR1', 'Bianchi');
-- l'insertion suivante viole la df (Groupe,Matiere) -> Enseignant
INSERT INTO SeancesBCNF VALUES ('310', '2007-1-21', 'GR1', 'Griffault');
```

Votre décomposition est seulement en 3NF *:*

1. *L'information* `(Enseignant, Matiere)` *est dupliquée.*
2. *La contrainte* `UNIQUE(Groupe,Jour)` *est devenue une contrainte de la relation* ***Seances****, et la contrainte* `UNIQUE(Enseignant,Jour)` *est devenue une contrainte entre les relations* ***Seances*** *et* ***ListeCours****.*
3. *L'insertion suivante nécessite d'évaluer une contrainte inter-relations.*

```
INSERT INTO ListeCours VALUES ('GR1', 'Bases de donnees', 'Bianchi');
INSERT INTO ListeCours VALUES ('GR2', 'Bases de donnees', 'Bianchi');
INSERT INTO Seances3NF VALUES ('113', '2007-1-20', 'GR1', 'Bases de donnees');
-- l'insertion suivante viole la contrainte UNIQUE(Enseignant, Jour)
INSERT INTO Seances3NF VALUES ('310', '2007-1-20', 'GR2', 'Bases de donnees');
```

Exercice 3 (Modification du schéma (6 points)) *Afin de mieux optimiser l'emploi du temps, le concepteur décide d'ajouter une relation* ***Repartition (Eleve, Groupe)*** *qui précise la liste des étudiants dans chacun des groupes. Il n'y a pas de dépendance dans cette relation, ce qui signifie qu'un étudiant peut appartenir à plusieurs groupes.*
Dans cet exercice, vous utiliserez au choix soit la relation initiale ***EmploiDuTemps****, soit l'ensemble des relations que vous avez obtenu à la question 2.3.*

Question 3.1 (2 points) *Écrire une requête SQL qui caractérise les* ***Eleve*** *qui ne savent pas quoi faire lorsqu'ils doivent être dans deux salles différentes le même jour.*

Réponse :

```
-- Algebre relationnelle
SELECT DISTINCT R1.Eleve
FROM            Repartition AS R1 NATURAL JOIN EmploiDuTemps AS EDT1,
                Repartition AS R2 NATURAL JOIN EmploiDuTemps AS EDT2
WHERE           R1.Eleve = R2.Eleve
  AND           EDT1.Jour = EDT2.Jour
  AND           EDT1.Salle <> EDT2.Salle;
-- Calcul relationnel
SELECT DISTINCT Eleve
FROM            Repartition AS R1 NATURAL JOIN EmploiDuTemps AS EDT1
WHERE EXISTS    (SELECT *
                 FROM   Repartition AS R2 NATURAL JOIN EmploiDuTemps AS EDT2
                 WHERE  R1.Eleve = R2.Eleve
                 AND    EDT1.Jour = EDT2.Jour
                 AND    EDT1.Salle <> EDT2.Salle);
-- Calcul relationnel
SELECT DISTINCT Eleve
FROM            Repartition AS R1 NATURAL JOIN EmploiDuTemps AS EDT1
WHERE NOT (Salle = ALL (SELECT Salle
                        FROM   Repartition AS R2 NATURAL JOIN EmploiDuTemps AS EDT2
                        WHERE  R1.Eleve = R2.Eleve
                        AND    EDT1.Jour = EDT2.Jour));
-- Utilisation des agregas
SELECT Eleve
FROM (SELECT DISTINCT Jour, Salle, Eleve
      FROM EmploiDuTemps NATURAL JOIN  Repartition) AS R
GROUP BY Eleve, Jour
HAVING Count (Eleve) > 1;
```

Question 3.2 (2 points) *Écrire une requête SQL qui caractérise les mauvaises attributions de salles pour un cours, car elles ne sont pas suffisamment grandes pour que tous les élèves puissent s'asseoir.*

Réponse :

```
SELECT DISTINCT Salle, Jour, Groupe, COUNT(DISTINCT Eleve), NbPlaces
FROM            Repartition NATURAL JOIN  EmploiDuTemps
GROUP BY        Salle, Jour, Groupe, NbPlaces
HAVING          COUNT(DISTINCT Eleve) > NbPlaces;
```

Question 3.3 (2 points) *Proposer un nouveau schéma conceptuel (liste de dépendances fonctionnelle, liste de contraintes) pour gérer l'emploi du temps, qui tienne compte des faits qu'un même élève peut appartenir à plusieurs groupes, et que plusieurs groupes peuvent être regroupés dans une même salle pour assister à un cours fait par un même enseignant.*

Réponse :

Les dépendances fonctionnelles

- $\{$***Salle***$\} \longrightarrow \{$***TypeSalle***$\}$ *qui indique que le numéro de la salle détermine son type.*
- $\{$***TypeSalle***$\} \longrightarrow \{$***NbPlaces***$\}$ *qui indique que le type d'une salle détermine son nombre de places.*
- $\{$***Enseignant***$\} \longrightarrow \{$***Matiere***$\}$ *qui indique qu'un enseignant ne donne des cours que dans une seule matière.*

- $\{$*Groupe, Matiere*$\} \longrightarrow \{$*Enseignant*$\}$ *qui indique que pour les élèves d'un groupe, la matière détermine l'enseignant.*
- $\{$*Salle, Jour*$\} \longrightarrow \{$*Enseignant*$\}$ *qui indique que pour un jour donné, il y a au plus un enseignant par salle.*

Les contraintes

- `UNIQUE(Jour, Enseignant)`
- `UNIQUE(Jour, Groupe)`
- `UNIQUE(Jour, Eleve)`
- *le nombre d'élèves dans une salle doit être inférieur au nombre de places de la salle.*