Département Licence

Année : 2005/2006 Session de janvier 2006

PARCOURS : INF 3
UE : INF 109, Bases de données
Épreuve : INF 109 EX
Date : Vendredi 6 janvier 2006
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 courses et de coureurs automobiles.
Soit $GrandPrix(Licence, Nom, Dossard, Equipe, Voiture, Annee, Victoires, Courses)$ une relation et ses dépendances fonctionnelles :
- $\{Dossard\} \longrightarrow \{Equipe, Voiture\}$ qui indique que le numéro de dossard détermine l'équipe et la voiture.
- $\{Equipe\} \longrightarrow \{Voiture\}$ qui indique que les équipes ne changent jamais de voiture.
- $\{Dossard, Annee\} \longrightarrow \{Nom, Victoires, Courses\}$ qui indique que tous les ans, un dossard est attribué à un seul coureur, et que l'on enregistre son nombre de victoires et son nombre de courses.
- $\{Nom, Annee\} \longrightarrow \{Dossard\}$ qui indique que tous les ans, un coureur a un seul dossard.
- $\{Licence\} \longrightarrow \{Nom\}$ qui indique que la licence est attribuée à vie.
- $\{Nom\} \longrightarrow \{Licence\}$ qui indique qu'il n'y a pas deux coureurs ayant le même nom.

Par exemple

Licence	Nom	Dossard	Equipe	Voiture	Annee	Victoires	Courses
321	Button	3	BAR	Honda	2005	0	19

spécifie que Button possède la licence 321 ; qu'en 2005, avec le dossard 3 qui désigne l'équipe BAR qui possède des voitures Honda, il a participé à 19 courses sans en gagner.

Exercice 1 (Requêtes SQL sur *GrandPrix* (6 points))

Question 1.1 (1 point) *Écrire une requête SQL qui caractérise les "Equipe" ayant gagné au moins une course.*

Réponse :

```
SELECT DISTINCT Equipe
FROM            GrandPrix
WHERE           Victoires > 0;
```

Question 1.2 (1 point) *En supposant que "(Dossard, Annee)" soit une clef candidate, écrire une requête SQL qui caractérise les "Dossard" n'ayant pas de "Victoires" en 2003, bien qu'ayant participé cette année là.*

Réponse :

```
SELECT DISTINCT Dossard
FROM            GrandPrix
WHERE           Annee = 2003
  AND           Victoires = 0
  AND           Courses > 0;
```

Question 1.3 (1 point) *En supposant que "(Dossard, Annee)" soit une clef candidate, écrire une requête SQL qui caractérise les "Dossard" ayant gagné des courses en 2000 et n'ayant pas de "Victoires" en 2003, bien qu'ayant participé cette année là.*

Réponse :

```
-- Une vision algèbrique
SELECT DISTINCT Dossard
FROM            GrandPrix
WHERE           Annee = 2000
  AND           Victoires > 0
INTERSECT
SELECT DISTINCT Dossard
FROM            GrandPrix
WHERE           Annee = 2003
  AND           Victoires = 0
  AND           Courses > 0;

-- Une vision prédicative
SELECT DISTINCT GP1.Dossard
FROM            GrandPrix AS GP1, GrandPrix AS GP2
WHERE           GP1.Dossard = GP2.Dossard
  AND           GP1.Annee = 2000
  AND           GP1.Victoires > 0
  AND           GP2.Annee = 2003
  AND           GP2.Victoires = 0
  AND           GP2.Courses > 0;
```

Question 1.4 (1 point) *Écrire une requête SQL qui caractérise les "Nom" des coureurs et le total de leurs "Victoires". Le résultat sera ordonné par ce total.*

Réponse :

```
SELECT DISTINCT Nom, SUM(Victoires)
FROM            GrandPrix
GROUP BY        Nom
ORDER BY        SUM(Victoires) DESC;
```

Question 1.5 (2 points) *En supposant qu'*ALONSO *et le "Dossard" 5 soient déjà dans la base ; écrire une requête SQL qui ajoute l'information qu'en 2005,* ALONSO *avec le dossard 5 a gagné 7 courses sur les 19 qu'il a disputées.*

Réponse : *La présence d'*ALONSO *permet de récupérer sa licence, et celle de "Dossard" de récupérer l'équipe et la voiture.*

```
INSERT INTO GrandPrix
  SELECT Licence, Nom, Dossard, Equipe, Voiture, 2005, 7, 19
  FROM   (SELECT          DISTINCT Licence, Nom
          FROM            GrandPrix
          WHERE           Nom = 'Alonso')
         AS TEMP1,
         (SELECT          DISTINCT Dossard, Equipe, Voiture
          FROM            GrandPrix
          WHERE           Dossard = 5)
         AS TEMP2;
```

Exercice 2 (Normalisation (6 points))

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

Réponse :
- $C_1 = \{Licence, Annee\}$
- $C_2 = \{Nom, Annee\}$
- $C_3 = \{Dossard, Annee\}$

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

Réponse : *Une seule des explications suivantes est suffisante.*

Non 2NF : $\{Licence\} \longrightarrow \{Nom\}$ *alors que "Licence" appartient à une clef candidate.*

Non 2NF : $\{Nom\} \longrightarrow \{Licence\}$ *alors que "Nom" appartient à une clef candidate.*

Non 3NF : $\{Equipe\} \longrightarrow \{Voiture\}$ *alors que "Equipe" et "Voiture" n'appartiennent à aucune clef candidate.*

Question 2.3 (2 points) *Appliquez un algorithme de normalisation pour obtenir une décomposition, sans perte d'information et sans perte de dépendance fonctionnelle, de la relation "GrandPrix" en un ensemble de relations 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 :
- $Team(Equipe, Voiture)$ *qui provient de* $\{Equipe\} \longrightarrow \{Voiture\}$
- $Numero(Dossard, Equipe)$ *qui provient de* $\{Dossard\} \longrightarrow \{Equipe\}$
- $Personne(Licence, Nom)$ *qui provient de* $\{Licence\} \longrightarrow \{Nom\}$ *ou de* $\{Nom\} \longrightarrow \{Licence\}$
- $Championnat(Annee, Dossard, Nom, Victoires, Courses)$ *qui provient de* $\{Annee, Dossard\} \longrightarrow \{Nom, Victoires, Courses\}$ *ou de* $\{Annee, Nom\} \longrightarrow \{Dossard, Victoires, Courses\}$

Question 2.4 (1 point) *Donnez une contrainte (de bon sens) entre les attributs "Victoires" et "Courses". Indiquez comment il est possible de la décrire en SQL.*

Réponse : *Le nombre de victoires est toujours inférieur ou égal au nombre de courses. Dans la relation qui contient les attributs "Victoires" et "Courses", il suffit d'ajouter la contrainte* `CHECK (Victoires <= Courses)`

Question 2.5 (1 point) *Indiquez, pour vos relations, les éventuelles clefs étrangères et les conséquences en cas de suppression de tuples.*

Réponse : *Il fallait parler des lignes* `REFERENCES` *présentes dans le schéma suivant.*

```
CREATE TABLE Personne (
  Nom             char(30) PRIMARY KEY,      -- ou UNIQUE NOT NULL
  Licence         integer  UNIQUE NOT NULL   -- ou PRIMARY KEY
);
CREATE TABLE Team (
  Equipe          char(20) PRIMARY KEY,
  Voiture         char(20) NOT NULL
);
CREATE TABLE Numero (
  Dossard         integer  PRIMARY KEY,
  Equipe          char(20) NOT NULL REFERENCES Team(Equipe) ON DELETE CASCADE
);
CREATE TABLE Championnat (
  Annee           integer,
  Dossard         integer  REFERENCES Numero(Dossard) ON DELETE CASCADE,
  -- Il faut soit Nom soit Licence pour la jointure avec Personne
  -- Il est préférable de choisir la clef primaire de Personne
  Nom             char(30) REFERENCES Personne(Nom) ON DELETE CASCADE,
  Victoires       integer  NOT NULL CHECK (Victoires >= 0),
  Courses         integer  NOT NULL CHECK (Courses >= 0),
  PRIMARY KEY (Annee, Dossard),              -- ou (Annee, Nom)
  UNIQUE (Annee, Nom),                       -- ou (Annee, Dossard)
  CHECK (Victoires <= Courses)
);
```

Exercice 3 (Requêtes SQL sur votre décomposition (9 points))

Question 3.1 (1 point) *Écrire une requête SQL qui caractérise les "Equipe" ayant gagné au moins une course.*

Réponse :

```
SELECT DISTINCT Equipe
FROM            Numero NATURAL JOIN Championnat
WHERE           Victoires > 0;
```

Question 3.2 (1 point) *Écrire une requête SQL qui caractérise les "Dossard" n'ayant pas de "Victoires" en 2003, bien qu'ayant participé cette année là.*

Réponse :

```
SELECT DISTINCT Dossard
FROM            Championnat
WHERE           Annee = 2003
  AND           Victoires = 0
  AND           Courses > 0;
```

Question 3.3 (1 point) *Écrire une requête SQL qui caractérise les "Dossard" ayant gagné des courses en 2000 et n'ayant pas de "Victoires" en 2003, bien qu'ayant participé cette année là.*

Réponse :

```
-- Une vision algèbrique
SELECT DISTINCT Dossard
FROM            Championnat
WHERE           Annee = 2000
  AND           Victoires > 0
INTERSECT
SELECT DISTINCT Dossard
FROM            Championnat
WHERE           Annee = 2003
  AND           Victoires = 0
  AND           Courses > 0;

-- Une vision prédicative
SELECT DISTINCT C1.Dossard
FROM            Championnat AS C1, Championnat AS C2
WHERE           C1.Dossard = C2.Dossard
  AND           C1.Annee = 2000
  AND           C1.Victoires > 0
  AND           C2.Annee = 2003
  AND           C2.Victoires = 0
  AND           C2.Courses > 0;
```

Question 3.4 (1 point) *Écrire une requête SQL qui caractérise les "Nom" des coureurs et le total de leurs "Victoires". Le résultat sera ordonné par ce total.*

Réponse :

```
SELECT DISTINCT Nom, SUM(Victoires)
FROM            Championnat
GROUP BY        Nom
ORDER BY        SUM(Victoires) DESC;
```

Question 3.5 (1 point) *En supposant qu'*ALONSO *et le "Dossard" 5 soient déjà dans la base ; Écrire une requête SQL qui ajoute l'information qu'en 2005,* ALONSO *avec le dossard 5 a gagné 7 courses sur les 19 qu'il a disputées.*

Réponse :

```
INSERT INTO Championnat VALUES (2005, 5, 'Alonso', 7, 19);
```

Question 3.6 (2 points) *Après en avoir donné une écriture algébrique, écrire une requête SQL qui caractérise les "Nom" des coureurs ayant couru au cours de leur vie avec toutes les "Voiture".*

Réponse : *La division algébrique s'écrit* $R^1 / R^2 = \pi[A_i](R^1) - \pi[A_i]((\pi[A_i](R^1) \times R^2) - R^1)$.
Dans cette formule, il suffit de remplacer :
- R^1 *par* $\pi[Nom, Voiture](Championnat \bowtie Numero \bowtie Team)$,
- R^2 *par* $\pi[Voiture](Team)$,
puis d'effectuer la substitution dans la requête SQL vue en cours.

```
-- les noms des coureurs ayant piloté toutes les voitures
-- Division algebrique avec R1(Nom, Voiture) et R2 (Voiture)
SELECT DISTINCT Nom
FROM (SELECT DISTINCT Nom, Voiture
      FROM   Championnat NATURAL JOIN Numero NATURAL JOIN Team
     ) AS R1_1
  EXCEPT
SELECT  Nom
FROM (SELECT *
      FROM   (SELECT DISTINCT Nom
              FROM (SELECT DISTINCT Nom, Voiture
                    FROM   Championnat NATURAL JOIN Numero NATURAL JOIN Team
                   ) AS R1_2
             ) AS PiR1,
             (SELECT DISTINCT Voiture
              FROM   Team) AS R2
        EXCEPT
      SELECT *
      FROM   (SELECT DISTINCT Nom, Voiture
              FROM   Championnat NATURAL JOIN Numero NATURAL JOIN Team
             ) AS R1_3
     ) AS NonEntierR1;

-- Ce qui donne après simplification
SELECT DISTINCT Nom
FROM Personne
  EXCEPT
SELECT  Nom
FROM (SELECT *
      FROM (SELECT DISTINCT Nom
            FROM Personne) AS PiR1,
           (SELECT DISTINCT Voiture
            FROM   Team) AS R2
        EXCEPT
      SELECT DISTINCT Nom, Voiture
      FROM   Championnat NATURAL JOIN Numero NATURAL JOIN Team
     ) AS NonEntierR1;
```

Question 3.7 (2 points) *Après en avoir donné une écriture prédicative, écrire une requête SQL qui caractérise les "Dossard" ayant gagné au moins une course toutes les années.*

Réponse : *La division prédicative s'écrit* $R^1 / R^2 = \{x \ / \ \nexists y \in R2, \ \nexists (a,b) \in R1 \ \wedge \ (x = a \wedge y = b)\}$.
Dans cette formule, il suffit de remplacer :
- R^1 *par* $\pi[Dossard, Annee](\sigma[Victoires > 0](Championnat))$,
- R^2 *par* $\pi[Annee](Championnat)$,
puis d'effectuer la substitution dans la requête SQL vue en cours.

```
-- les dossards ayant gagné une course toutes les années
-- Division prédicative avec R1(Dossard, Annee) et R2 (Annee)
SELECT DISTINCT Dossard
FROM   (SELECT Dossard, Annee
        FROM   Championnat
        WHERE  Victoires > 0) AS R1
WHERE  NOT EXISTS
       (SELECT *
        FROM  (SELECT Annee
```

```
               FROM   Championnat) AS R2
        WHERE  NOT EXISTS
               (SELECT Annee
                FROM   (SELECT Dossard, Annee
                        FROM   Championnat
                        WHERE  Victoires > 0) AS R11
                WHERE  R11.Dossard = R1.Dossard
                  AND  R11.Annee   = R2.Annee));

-- Ce qui donne après simplification
SELECT DISTINCT Dossard
FROM   Championnat AS C1
WHERE  Victoires > 0
  AND  NOT EXISTS
       (SELECT Annee
        FROM   Championnat AS C2
        WHERE  NOT EXISTS
               (SELECT Annee
                FROM   Championnat AS C3
                WHERE  Victoires > 0
                  AND  C3.Dossard = C1.Dossard
                  AND  C3.Annee   = C2.Annee));
```

Exercice 4 (Transactions)

Question 4.1 (2 points) *Imaginez une suite de requêtes, telles que si elles ne sont pas mises dans une transaction, la sélection de la question 3.3 ne donne pas un résultat correct.*

Réponse : *Soit la suite de requêtes.*

```
-- START TRANSACTION ;
INSERT INTO Personne    VALUES ('Moi', 17);
INSERT INTO Team        VALUES ('MonEquipe', 'MaVoiture');
INSERT INTO Numero      VALUES (21, 'MonEquipe');
INSERT INTO Championnat VALUES (2000, 21, 'Moi', 13, 14);
INSERT INTO Championnat VALUES (2003, 21, 'Moi', 0, 18);
-- END ;
```

Suivant si la requête de la question 3.3 est exécutée entre les deux dernières insertion ou pas, le dossard 21 est dans la réponse ou non.

Exercice 5 (Subsidiaire)

Question 5.1 (2 points) *Expliquez en quelques lignes pourquoi la requête suivante, pour n'importe quelle base de données ayant un attribut "DateDeNaissance", caractérise toujours un ensemble vide.*

```
-- Les personnes nés dans la nuit du 3 au 13 septembre 1752.
SELECT   *
FROM     MaTable
WHERE    DateDeNaissance BETWEEN '1752-9-4' AND '1752-9-12';
```

Réponse : *Un extrait de la documentation de* `PostgreSQL 8.0.3` *donne l'explication.*
The papal bull of February 1582 decreed that 10 days should be dropped from October 1582 so that 15 October should follow immediately after 4 October. This was observed in Italy, Poland, Portugal, and Spain. Other Catholic countries followed shortly after, but Protestant countries were reluctant to change, and the Greek orthodox countries didn't change until the start of the 20th century. The reform was observed by Great Britain and Dominions (including what is now the USA) in 1752. Thus 2 September 1752 was followed by 14 September 1752. This is why Unix systems have the cal program produce the following :

```
> cal 9 1752
   septembre 1752
di lu ma me je ve sa
       1  2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```