Université Paris 7

Licence 3 Informatique et Master 1 ISIFAR, Bases de données.

17 juin 2010

Durée : 3 heures. Documents manuscrits, notes de cours, notes de TD/TP autorisés. Livres interdits.

Le sujet comporte 4 pages.

Comprendre les requêtes

Exercice 1 Pour des tables R et S

```
R  a |b  | c              S  c | d
  ---+---+---               ---+---
   1 | 2 | 3                 2 | 6
   1 | 2 | 4                 3 | 2
   2 | 3 | NULL              5 | 1
   3 | 1 | 5               NULL | 9
   3 | 2 | 3                 6 | 2
                             6 | 3
                             2 | 2
```

donner les résultats des requêtes

```
SELECT SUM(a) AS X, B
FROM R NATURAL LEFT JOIN S
GROUP BY B
ORDER BY X ASC;
```

et

```
SELECT c, COUNT(*)
  FROM S
  WHERE S.c >= ALL(SELECT a FROM R)
  GROUP BY C
  HAVING COUNT(*)>1;
```

Écrire les requêtes SQL

Exercice 2 Une base de données d'un nutritionniste qui travaille dans un hôpital est composée des tables :

```
plat(id_plat, nom, calories)
patient(id_patient, nom, prenom, date_naiss, adresse, date_sortie)
servi(id_plat, id_patient, date, quand)
interdit(id_patient, id_plat)
```

(A) La table `plat` contient tous les plats qui peuvent être servis par la cuisine, `id_plat` sert de clé. Tous les attributs sont non `NULL` et l'attribut `calories` donnant le nombre de calories (de type int) doit être supérieur à 0.

(B) La table `patient` recense tous les patients présents à l'hôpital, `id_patient` de type int sert de clé. `date_sortie` est le seul attribut qui peut être NULL si la date de sortie est inconnue. Si `date_sortie` est non NULL la valeur de cet attribut donne la date de sortie prévue. La table `patient` ne contient que les patients qui sont actuellement à l'hôpital.

(C) La table `servi` énumère les plats qui ont été servis aux patients. Les attributs `id_plat` et `id_patient` sont des clés étrangères vers les attributs correspondant des tables `plat` et `patient`. L'attribut `date` nous donne la date à laquelle le plat a été servi à un patient donné et l'attribut `quand` d'un caractère indique si le plat à été servi le matin, à midi ou le soir. Les valeurs possible de l'attribut `quand` :
- `p` – si le plat a été servi au petit-déjeuner,
- `m` – si le plat a été servi à midi,
- `s` – si le plat a été servi le soir.

La clé de la table `servi` est composée de l'ensemble de tous les attributs.

(D) La table `interdit` indique les interdictions, si un couple `(id_patient,id_plat)` se trouve dans cette table alors cela indique que le plat `id_plat` est interdit au patient `id_patient`. Les deux attributs de cette table sont des clés étrangères et le couple `(id_patient,id_plat)` sert de la clé primaire.

Les questions suivantes sont indépendantes.

Question 1: Écrire les commandes SQL qui construisent toutes ces tables avec les contraintes d'intégrité suivant la description ci-dessus.

Question 2: Écrire une commande permettant d'insérer d'un seul coup deux patients dans la table `patient` :

1. Dupont Marcel, né 6 juillet 1965, admis aujourd'hui, date de sortie non connue,
2. Chanteau Chantale, née 24 mars 1977, admise hier, va sortir une semaine après admission.

À la place de `aujourdhui` il faut mettre la date d'aujourd'hui, `hier` la date d'hier. La date de sortie de Chantale doit être renseignée de façon appropriée. Les deux patients ont `id_patient` de votre choix.
Attention : On préfère que toutes les dates soient calculées en utilisant les fonctions et les expressions SQL au lieu d'être mis en dur (on suppose que la personne qui ajoute les enregistrements a oublié son agenda mais connaît parfaitement SQL).

Question 3: Mettre à jour un enregistrement dans la table patient : le patient `id_patient=3` sortira un jour plus tard par rapport à la date de sortie initialement prévue.

Question 4: Supprimer de la table `patient` tous les patients avec la date de sortie hier. On va exécuter cette requête chaque matin, donc hier ne peut pas être en dur mais calculé par SQL.

Question 5: Il y a combien de patients actuellement à l'hôpital ?

Question 6: Combien de calories va consommer la patiente Chanteau Chantale aujourd'hui ?

Question 7: Pour chaque patient afficher : le nom, prénom et le nombre de calories qu'il a consommé aujourd'hui matin, le résultat trié dans l'ordre croissant du nombre de calories.

Question 8: Quels sont les plats les plus caloriques, afficher leurs noms (on peut avoir plusieurs plats avec le même nombre maximal de calories).

Question 9: Donner le nombre de patients admis à l'hôpital l'année dernière et qui sont toujours dans la table `patient` (qui ne sont pas encore sortis).

Question 10: Afficher les noms et prénoms de tous les patients qui ont mangé le plat `crevettes` hier soir.

Question 11: Afficher tous les patients (nom, prénom et `id_patient`) dont la date de sortie est inconnue.

Question 12: Trouver tous les patients qui ont mangé au moins trois fois le plat `pizza` pendant les 7 derniers jours.

Question 13: Trouver tous les patients qui ne mangent que des pizzas depuis leur admission (ils mangent une pizza le matin, midi et soir, chaque jour et n'ont jamais mangé rien d'autre).

Question 14: Trouver tous les patients qui peuvent manger une pizza (pour lesquels la pizza n'est pas interdite).

Question 15: Les patients qui ont mangé au moins 4000 calories hier (au total le matin + le midi + le soir) vont être privés de repas aujourd'hui soir. Les trouver (leur nom et prénom).

Question 16: Trouver tous les patients qui ont été admis aujourd'hui mais qui n'ont rien mangé au petit déjeuner (sans doute ils étaient admis après le petit déjeuner).

Question 17: Il y a combien de patients qui n'ont aucun plat interdit ?

Question 18: Trouver tous les plats qui ne sont interdits à aucun patient.

JDBC

Exercice 3 La base de données `voyageur` contient la table

```
CREATE TABLE distance(
        villeA   varchar(30) NOT NULL,
        villeB   varchar(30) NOT NULL,
        dist     int check(dist > 0),
        primary key(villeA , villeB)
        );
```

qui donne les distances entre des villes.
Le programme `Trajet` qui suit prend en argument les villes `v0 v1 v2 ...` et calcule la longueur du trajet complet `v0 -> v1 -> v2 ...`
Par exemple l'appel

```
java Trajet Paris Lyon Bordeaux Toulouse Lille
```

doit calculer la somme de distances

$$dist(Paris, Lyon)+dist(Lyon, Bordeaux)+dist(Bordeaux, Toulouse)+dist(Toulouse, Lille)$$

Écrire le code manquant.

```
import java.sql.*;
public class Trajet{
    public static void main(String[] args){
        String url = "jdbc:postgresql:voyageur";
        Connection conn=null;
        double somme=0;
        try{
            Class.forName("org.postgresql.Driver");
            conn = DriverManager.getConnection(url,"postgres","dupek");
        }catch(ClassNotFoundException e){
            System.err.println("Pb. pilote"); System.exit(1);
        }
        catch(SQLException e){
            System.err.println("Pb " + e.getMessage()); System.exit(2);
        }
        try{
            /*********************************************************
             * Calculer la longuer du trajet entre les villes.       *
             * On pourra utiliser la variable somme pour memoriser   *
             * le resultat.                                          *
             *********************************************************/

        }catch(SQLException e){
            System.err.println("probleme : " + e.getMessage() );
            System.exit(3);
        }
        //afficher les resultat
        System.out.println("trajet de  " + somme +" kilometres");
    }
}
```

Conception d'une BD

Exercice 4 Le but de cet exercice est de concevoir une base de données de gestion de comptes bancaires. On prendra en compte les clients, les comptes, les opérations sur les comptes (débit et crédit). A noter qu'un compte peut avoir plusieurs propriétaires (un compte appartenant à des époux par exemple).
Vous pouvez ajouter d'autres entités de votre choix.

Question 1: Proposez un Modèle Conceptuel de Données (MCD) pour cette BD, avec plusieurs entités avec des attributs appropriés, des relations entre des entités. **Très important :** N'oubliez pas de bien marquer les cardinalités.

Question 2: En déduire le schéma de la base, c'est-à-dire les tables que vous obtenez à partir de votre MCD.
Écrire les commandes `CREATE TABLE` pour ces tables avec les contraintes d'intégrité appropriées.
