

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

Le sujet comporte 6 pages.

Comprendre les requêtes

Exercice 1 Pour des tables R et S

R	a	b	c		S	c	d
-----					-----		
1	8	1			2	6	
1	6	2			3	2	
2	11	1			3	1	
2	6	-5			13	9	
3	0	6					
3	3	0					
4	4	4					
4	7	2					
5	0	3					

59 6
 10+3 3
 3
 20+3
 30+3
 C. count(*)
 3
 3
 3

donner les résultats des requêtes

```
SELECT SUM(a*d) AS X, C
FROM R NATURAL JOIN S
WHERE C in (SELECT a+1
            FROM R)
GROUP BY C
ORDER BY X ASC;
```

```
et SELECT C, COUNT(*)
from R natural left join S
WHERE a>=2
GROUP BY C
HAVING COUNT(*)>1;
```

Écrire les commandes SQL

Exercice 2 Une base de données de gestion de notes d'une école (lycée + collège) est composée des tables :

```
prof(prof_id, nom, prenom, date_naiss)
classe(classe_id, niveau, profile)
eleve(eleve_id, nom, prenom, date_naiss, classe_id)
enseignement(prof_id, classe_id, matiere)
note(eleve_id, valeur, coef, date, matiere, devoir)
```

- (A) La table prof contient tous les professeurs, prof_id sert de clé. Tous les attributs sont non NULL.
- (B) La table classe énumère toutes les classes. classe_id - un identifiant - la clé de la table, c'est une chaîne de caractères.
niveau de type int avec les valeurs possibles entre 0 et 6 (0 - terminale, 1 - première etc.). profil - une chaîne de caractère, si non NULL alors donne le profil de la classe (par exemple L ou STI). C'est le seul attributs qui peut être NULL.

- (C) La table `eleve` énumère les élèves. Chaque élève a un identifiant unique `eleve_id` – c'est la clé de la table. `classe_id` une clé étrangère vers `classe`. Tous les attributs sont non NULL.
- (D) `enseignement` indique les matières enseignées par les professeurs dans chaque classe. `prof_id` et `classe_id` sont des clés étrangères, le triplé `prof_id, classe_id, matiere` sert de la clé de la table. On suppose que dans une classe donnée une matière est enseignée par un seul professeur, par exemple la physique est enseignée dans la classe C1 uniquement par le professeur Toto Malin. (Bien sûr un professeurs peut enseigner plusieurs matières dans la même classe ou dans les classes différentes). `matiere` c'est une chaîne de caractères qui prendra de valeurs comme biologie, chimie etc.
- (E) La table `note` donne les notes des élèves pendant les semestre (ce ne sont pas les notes finales du semestre). `eleve_id` est une clé étrangère. `valeur` – un nombre décimal avec au plus deux chiffres après virgule, c'est une note. `coef` un nombre décimal > 0 — le coefficient d'une note. La valeur par défaut de cet attribut est 1 (c'est-à-dire si pendant l'insertion d'une note `coef` n'est pas indiquer alors il faut lui attribuer la valeur 1). L'attribut `devoir` de type `VARCHAR` permet au professeur d'indiquer le devoir dans lequel l'élève a obtenu une note donnée. C'est le seul attribut qui peut être NULL. Les attributs `eleve_id, valeur, sur, date, matiere` constituent ensemble la clé de la table `note`.

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 dans la table `note` deux lignes :

(2, 15.5 , 20 , aujourd'hui , 'math' , 'devoir 2') et
 (2, 7, 20, il y a une semaine, 'biologie', 'devoir 4') (l'ordre d'attributs est le même que dans la spécification de la table).

À la place de `aujourd'hui` il faut mettre la date d'aujourd'hui et à la place de `il y a une semaine` la date d'il y a une semaine. Mais attention, on préfère que la date d'aujourd'hui soit obtenue par un appel approprié à une fonction SQL au lieu de `2010-05-18` mis en dur. Pareil pour la date `il y a une semaine` on préfère une solution qui calcule la date d'il y a une semaine à l'aide d'une expression SQL plutôt que `'2010-05-11'` écrit en dur.

Question 3: Mettre à jour un enregistrement dans la table `note` en ajoutant un point à l'élève `eleve_id=4` dans la matière `chimie` dans `devoir='interro 2'`. Cet enregistrement existe déjà, il faut juste modifier la note.

Question 4: Trouver l'élève le plus jeune dans l'école, afficher son nom, prénom, date de naissance. Attention, s'il y a plusieurs élèves avec la même date de naissance — la plus récente, la réponse doit les énumérer tous.

Question 5: Pour chaque professeurs afficher : nom, prénom, `prof_id` et le nombre de matières différentes qu'il enseigne. Le résultat doit être trié dans l'ordre descendante par le nombre de matières enseignées.

Question 6: Est-ce qu'il y a des professeurs qui n'enseignent rien (aucune matière) ? Les afficher tous.

Question 7: Trouver les élèves qui ont le même nom qu'un des enseignants.

Question 8: Donner le nombre d'élèves dans chaque classe.

Question 9: Donner le nombre d'élèves niveau par niveau.

Question 10: Trouver tous les professeurs qui ont 60 ans cette année

Question 11: Trouver l'âge moyen de professeurs.

Question 12: L'élève dont `eleve_id=234` veut calculer sa note moyenne en mathématiques. S'il a k notes n_1, \dots, n_k avec les coefficients c_1, \dots, c_k alors la moyenne se calcule par la formule

$$(n_1 \cdot c_1 + \dots + n_k \cdot c_k) / (c_1 + \dots + c_k) \quad (1)$$

c'est-à-dire comme la somme pondérée de notes divisée par la somme de coefficients. Écrire la requête qui permet à notre élève d'effectuer le calcul indiqué.

Question 13: Trouver tous les élèves de la classe `classe_id='C2'` qui n'ont aucune notes en mathématique.

Question 14: Trouver les classes qui ont soit plus de 35 élèves soit moins de 15 élèves et pour chaque de ces classes afficher le nombre d'élèves.

Question 15: Trouver pour chaque niveau le nombre d'élèves qui auront 16 ans cette année (les élèves nés en 1994). Bien sûr la plupart d'entre eux sont en deuxième (niveau 2) mais certains plus avancés (surdoués) peuvent se trouver à niveau 1 ou 0 tandis que d'autres, disons retardés, peuvent se trouver à niveau 3, 4, etc. Le résultat doit être trié par niveau.

Question 16: A la fin du semestre le prof de maths veut calculer pour chaque élève de la classe `classe_id='2D'` la note finale en math. Cette note doit être calculée par la formule (1) donnée dans la question 12. Écrire la requête correspondante.

Question 17: Le prof de physique veut calculer pour la classe `classe_id='E1'` la note moyenne du devoir `devoir='DS1'`. Écrire la requête correspondante. Ici la note moyenne c'est juste la somme de notes du devoir divisée par le nombre de notes, les coefficients ne sont pas utilisés. Les notes NULL (absent) ne seront pas prises en compte.

Question 18: Pour la même classe et le même devoir DS1 en physique que dans la question précédente trouver tous les élèves avec la note (de ce devoir) supérieure à la moyenne de notes de ce devoir.

Question 19: Vers la fin de semestre le proviseur constate que dans certaines classes il y a des matières sans aucune note. Il veut parler aux professeurs concernés et leur demander de faire un effort et corriger au moins un devoir avant que le semestre se termine. Trouver tous les triplets

`classe_id, matiere, nom, prenom`

tels que

- nom, prenom sont le nom et prénom du professeur qui enseigne la matière matiere dans la classe classe_id et
- aucun élève de cette classe n'a pas de note dans cette matière.

Question 20: Quelle est le nombre de professeurs qui enseignent la biologie? (attention, éviter de compter le même prof plusieurs fois).

JDBC

Exercice 3 Le programme Java suivant qui prend en paramètre matiere et devoir et calcule pour le devoir dans la matière indiqués par les paramètres les statistiques de notes, c'est-à-dire le nombre de notes "très bien" (≥ 16), "bien" (entre 14 et 16), "assez bien" (entre 12 et 14), "passable" (entre 10 et 12), "insuffisant" (< 10).

De plus le programme calcule la moyenne de notes pour ce devoir.

Dans le programme il manque un bout de code (marqué par le commentaire). Écrire le code manquant.

```

1  import java.sql.*;
2  public class StatDev{
3      public static void main(String[] args){
4          int tres_bien=0, bien=0, assez_bien=0,
5              passable=0, insuffisant = 0,
6              compteur=0 ;
7          double somme = 0;
8
9          String matiere, devoir;
10         if( args.length != 2 ){
11             System.err.println("usage: matiere devoir");
12             System.exit(1);
13         }
14         matiere = args[0];
15         devoir = args[1];
16
17         String url = "jdbc:postgresql:EXAM_mai_2010";
18         Connection conn=null;
19         try{
20             Class.forName("org.postgresql.Driver");
21             conn = DriverManager.getConnection(url,"toto","dupek");
22         }catch(ClassNotFoundException e){
23             System.err.println("Pb. pilote"); System.exit(1);
24         }
25         catch(SQLException e){
26             System.err.println("Pb " + e.getMessage()); System.exit(2);
27         }
28
29         Statement stat = null;
30         /*****
31          * a la place de Statement vous pouvez utiliser, si vous preferez,
32          * PreparedStatement et la ligne precedente devient

```

```

33     * PreparedStatement stat =
34     *   conn.prepareStatement("... mettre ce qui convient ...");
35     *****/
36
37     ResultSet rs = null;
38     try{
39
40         stat = conn.createStatement();/*inutile si PreparedStatement*/
41
42
43         /*****
44         * Ici mettre le code qui execute la requete,
45         * parcourt le resultat
46         * recupere la valeur de la note
47         * met a jour les variables tres_bien,bien,assez_bien,insuffisant
48         * en fonction de la valeur de la note,
49         * la somme de notes et le compteur de nombres de notes.
50         *****/
51
52
53         rs.close();
54         stat.close();
55     }catch(SQLException e){
56         System.err.println("probleme : " + e.getMessage() ); System.exit(3);
57     }
58
59     //afficher les resultat
60     System.out.println("tres bien " + tres_bien);
61     System.out.println("bien      " + bien);
62     System.out.println("assez bien " + assez_bien);
63     System.out.println("passable  " + passable);
64     System.out.println("insuffisant " + insuffisant);
65     System.out.println("Moyenne = " + (somme/compteur));
66
67
68     }
69 }

```

Conception d'une BD

Exercice 4 Le but de cet exercice est de concevoir une base de données de gestion d'un restaurant.

L'entité principale c'est un plat avec avec son prix de vente. Chaque plat est composé d'ingrédients. Pour chaque ingrédient utilisé dans le plat il faut connaître la quantité de l'ingrédient utilisée dans ce plat.

Les ingrédients sont fournis par des fournisseurs, chaque ingrédient possède le prix d'achat qui dépend de fournisseur.

Pour savoir comment les plats se vendent le patron de restaurant a besoin de savoir pour

chaque jour et pour chaque plat combien d'unité de ce plat il a vendu à midi et combien le soir (peut-être les ventes ne sont pas le même à midi et le soir, il a besoin de chiffres exactes jour par jour).

Les entités évidentes : plats, ingrédients, fournisseurs, et sans doute encore quelque chose pour les ventes.

Vous pouvez ajouter d'autres entités de votre choix.

Question 1: Proposez un Modèle Conceptuel de Données pour cette BD, avec une taille raisonnable d'au moins 3 entités avec des attributs appropriés, plusieurs relations. N'oubliez pas de bien marquer les cardinalités.

Question 2: En déduire le schéma de la base, c'est-à-dire quelles tables vous obtenez à partir de votre MCD. On ne demande pas d'écrire les commandes CREATE TABLE pour les tables mais vous devez indiquer pour chaque table que vous devez créer : le nom de la table, les attributs, la clé, les contraintes référentielles s'il y en a (clés étrangères).

Transactions

Exercice 5 Nous avons trois transactions T_1, T_2, T_3 exécutées simultanément.

On considère l'histoire :

$$H : r_3(Z), r_1(X), w_1(Y), r_2(Y), w_2(Z), w_3(X), r_1(X)$$

$r_i(A)$ désigne une lecture d'une variable A par la transaction i , $w_i(A)$ désigne l'écriture. Est-ce que H est sérialisable. La réponse doit être justifiée avec le graphe de conflit.

Exercice 6

Une transaction

$$lock(A); r(A); unlock(A); lock(B); r(B); w(B); unlock(B)$$

est-elle conforme à l'algorithme de verrouillage en deux phases? (lock — verrouiller, unlock — déverrouiller, on suppose que les seuls verrous disponibles sont les verrous exclusifs utilisés aussi bien pour la lecture que pour l'écriture).

Si cette transaction ne respecte pas l'algorithme de verrouillage en deux phases alors pas déplacez une et seulement une des opérations lock/unlock pour obtenir le verrouillage en deux phases.
