

Cours de Bases de données L3 informatique
Examen de deuxième session Année 2012-2013
Durée 2 heures et demie
Seul document autorisé : 3 feuilles A4 recto-verso.

Les différentes parties de l'examen sont toutes indépendantes.

Partie 1 : requêtes SQL : en promo

Un supermarché est organisé en rayons (charcuterie, produits ménagers,...). Chaque article a donc un rayon dans lequel il se trouve.

Par ailleurs, ce supermarché organise des promotions. Ces promotions sont de deux types, soit elles portent sur un article, soit elles portent sur un rayon. Pour chaque promotion, on donne le pourcentage de réduction appliqué (un entier entre 0 et 100). On ne garde en mémoire que les promos qui s'appliquent maintenant.

- **Rayon(*id_ray, nom_ray)** la liste des rayons (identifiant, nom) ;
- **Article(*id_art, nom_art, prix, id_ray#)** la liste des articles (identifiant de l'article, nom, prix, rayon dans lequel il se trouve) ;
- **PromoRayon(*id_p_ray, reduc, id_ray#)** promo sur un rayon (identifiant, pourcentage de la réduction, rayon) ;
- **PromoArticle(*id_p_art, reduc, id_art#)** promo sur un article, (identifiant, pourcentage de la réduction, article) ;

Questions

Écrivez les requêtes SQL correspondant aux questions suivantes.

1. (facile) les articles (identifiant, nom, et prix réduit si on applique la promotion) en promo dans une promotion d'article en ce moment. On ne regardera pas s'il bénéficie aussi d'une promo de rayon, et on suppose qu'un même article correspond au plus à une promo rayon ;
2. (assez facile) les articles (identifiant et nom) qui sont en ce moment, en promo d'article et en promo de rayon mais pas avec la même réduction ;
3. (moyen) les articles qui ne sont pas en promotion de rayon, et dont la valeur est supérieure à la moyenne des prix (hors promo) des autres articles du même rayon ;
4. (difficile) par rayon, le nombre d'articles en promotion (que ce soit à cause d'une promotion article ou d'une promotion rayon). Tableau résultat : (id_ray, nom_ray, nbreTotal, nbreEnPromo). Pour cette dernière requête, vous avez le droit de passer par une vue.

Partie 2 : Algèbre relationnelle

- **Animal**(*id_ani, nom_ani) la liste des animaux (identifiant, nom) ;
- **ModeDeplacement**(*id_dep, desc) la liste des modes de déplacement (identifiant, description du mode (voler, marcher, ramper...)) ;
- **Deplace**(*id_ani#, *id_dep#) indique pour chaque animal son ou ses modes de déplacement ;
- **Mange**(*id_mangeur#, *id_proie#) indique quel animal (id_mangeur) mange quel(s) autre(s) animal(aux) (id_proie). On donne à chaque fois l'identifiant de l'animal.

Questions

Pour chaque question, donnez la requête en algèbre relationnelle qui y répond. Pour plus de lisibilité, vous pouvez faire la requête en plusieurs étapes, par exemple :

$$\begin{aligned} A &= \pi_{val}(R) \bowtie S \\ B &= \sigma_{att=5}(T \cup R) \\ \text{réponse} &= A \cap B \end{aligned}$$

Par ailleurs, si vous avez besoin de distinguer deux utilisations d'une même table, utilisez un numéro pour distinguer les deux utilisations de la table, par exemple, si on utilise deux fois la table T :

$$C = \sigma_{att=5}(T_1 \bowtie_{T_1.val < T_2.val} T_2)$$

Si vous avez besoin de renommer un attribut ou une table...expliquez ce que vous faites. Donnez les requêtes algébriques correspondant aux ensembles suivants :

1. Les animaux végétariens, c'est-à-dire qui ne mangent pas d'autres animaux. Donnez l'identifiant et le nom ;
2. Les animaux rampants qui mangent au moins un animal volant. Donnez seulement l'identifiant ;

Partie 3 : Modélisation : géographie

Dans cet exercice, on vous demande une modélisation entité-relation sur le thème de la géographie. Vous êtes libre de faire ce que vous voulez dans la mesure où vous respectez les faits suivants :

La base de données doit contenir des cours d'eau (rivières, fleuves). Pour ces cours d'eau, on doit savoir par quelles villes ils passent et, dans le cas des fleuves, dans quelle mer ou océan ils se jettent. (On rappelle qu'un fleuve est un cours d'eau qui se jette dans la mer ou dans un océan).

Si vous manquez d'idées pour compléter votre base, vous pouvez, par exemple, indiquer les pays dans votre base.

Question 1

Donnez une modélisation entité-relation contenant au moins 4 tables, 3 relations, et 6 attributs. Vous respecterez la norme **crowfoot** (celle du cours). **Vous justifierez aussi les choix qui pourraient ne pas être évidents.**

Question 2

Donnez ensuite une traduction en tables de votre modélisation. La traduction aura la même forme que dans l'énoncé de la partie 1.

1 Partie 4 : Contrôle de la concurrence

Pour des raisons de sécurité, certains bâtiments ne peuvent accueillir qu'un nombre limité de personnes. Pour les compter, le service de sécurité qui gère ces bâtiments, utilise une base de données qui contient la table suivante :

nbPresents(*id_bat, nbre) où **nbre** est le nombre de personnes présentes dans le bâtiment d'identifiant **id_bat**.

Il y a trois entrées au bâtiment 1, munies de compteurs électroniques. On considère les 3 transactions suivantes :

Transaction 1 Le compteur de l'entrée 1, compte 1 personne qui rentre, et fait les opérations suivantes :

1. `select nbre from nbPresents where id_bat = 1 ;`
La valeur affichée est alors stockée dans une variable x .
2. `update nbPresents set nbre = x + 1 where id_bat = 1 ;`

Transaction 2 Le compteur de l'entrée 2, compte 3 personnes qui sortent, et fait les opérations suivantes :

1. `select nbre from nbPresents where id_bat = 1 ;`
La valeur affichée est alors stockée dans une variable y .
2. `update nbPresents set nbre = y - 3 where id_bat = 1 ;`

Transaction 2 Le compteur de l'entrée 3, compte 2 personnes qui entrent, et fait les opérations suivantes :

1. `select nbre from nbPresents where id_bat = 1 ;`
La valeur affichée est alors stockée dans une variable z .
2. `update nbPresents set nbre = z + 2 where id_bat = 1 ;`

Questions

1. On suppose qu'avant que ces trois transactions ne soient faites, il y a 100 personnes dans le bâtiment 1. Ensuite, ses trois transactions sont faites de manière quasi-simultanée. Donnez un exemple d'ordonancement qui provoque une anomalie. Expliquez brièvement en quoi c'est une anomalie. **Note :** On considère que chaque commande SQL est atomique, c'est-à-dire qu'elle se fait en une seule fois sans que d'autres opérations puissent s'intercaler.

2. Modélisez les transactions précédentes avec les opérations $r_i(X)$ et $w_i(X)$ (lecture et écriture dans la table X par la transaction T_i).
3. On applique le verrouillage en 2 phases sur la transaction T_1 . Indiquez le résultat. Notation : $rl_i(X)$ (resp. $wl_i(X)$) pour le verrouillage en lecture (resp. écriture) de la table X par la transaction T_i et $ul_i(X)$ pour le déverrouillage.