

Examen

Luc Segoufin, Emmanuel Waller
7 Décembre 2004

4 pages. Durée de l'examen : 2 heures. Les transparents de cours et de TD, les énoncés de TD et leur corrigé, les notes manuscrites et vos propres fichiers sont autorisés. Tout autre document est interdit. Les exercices sont indépendants (sauf qu'il faut lire le descriptif du schéma de l'exercice 1 avant l'énoncé du 2). Écrivez lisiblement. Justifiez vos résultats.

Attention! Ce n'est pas difficile mais c'est long. Ne perdez pas de temps en route.

1 Requêtes

Soit le schéma :

Hôtel(Hnom,ville,prix-nuitée)
Resto(Rnom,ville,prix-menu)
Guide(Gnom,prix)
Note(Gnom,H-ou-Rnom,note)

Par exemple la table Hôtel contient le tuple (Au-bon-séjour, Palaiseau, 60), la table Resto le tuple (Chez-Manu, Montrouge, 30), la table Guide le tuple (Michelin, 15) et la table Note le tuple (Michelin, Chez-Manu,3). Les prix sont en euros (TTC), la note est un entier positif ou nul, par exemple le nombre d'étoiles ou de toques.

Afin de simplifier l'écriture des requêtes on suppose que chaque hôtel et chaque restaurant est uniquement identifié par son nom (pour chaque nom x il y a au plus un établissement de nom x , hôtel ou restaurant). On suppose aussi que toute ville a au moins un hôtel et au moins un restaurant. On suppose enfin que tout restaurant et tout hôtel est noté au moins une fois dans l'un des guides.

Exprimez les requêtes suivantes dans SQL, ALG et CALC (trois réponses sont donc attendues par question).

1. Quels sont les hôtels de Palaiseau?
2. Quelles sont les villes ayant un restaurant avec un menu inférieur ou égal à 20 € et un hôtel dont la nuit coûte strictement moins que 30 €?
3. Quels sont les hôtels qui sont dans une ville ayant un restaurant noté 4 au Guide Michelin?
4. Quels sont les restaurants ayant une note supérieure strictement à 3 dans tous les guides?
5. Quels sont les guides qui notent tous les restaurants.

Exprimez les requêtes suivantes dans SQL uniquement (une réponse par question).

1. Combien d'hôtels y a-t-il à Palaiseau?
2. Quel est le nombre d'hôtels pour chaque ville?
3. Quel est en moyenne le nombre d'hôtels par ville?

2 Ecriture d'application

On s'intéresse maintenant au mécanisme des réservations dans les établissements de l'exercice 1. Mais pour simplifier, on se restreint aux hôtels, et on ignore les guides et les notes. On considère donc dans le présent exercice la situation suivante, qui correspond donc intuitivement à certaines tables d'une agence de réservation. Un hôtel a un certain nombre de chambres. On réserve une chambre uniquement pour le lendemain, et pour un seul jour. Les données sont organisées dans le schéma :

Hôtel(Hnom, nb-chambres)
Libre(Hnom, no-chambre)
Résa(Hnom, no-chambre, client)

Par exemple, la ligne (*Au-bon-séjour*, 100) dans *Hôtel* signifie que l'hôtel *Au-bon-séjour* a 100 chambres. La ligne (*Au-bon-séjour*, 17) dans *Libre* signifie que la chambre 17 de cet hôtel est encore libre pour demain. La ligne (*Au-bon-séjour*, 19, *Luc*) dans *Résa* signifie que *Luc* a réservé la chambre 19 de cet hôtel pour demain. On suppose que les numéros de chambre sont les entiers entre 1 et le nombre de chambres de l'hôtel (donc tous distincts bien sûr).

Le fonctionnement de l'agence pour ces tables est le suivant. Chaque jour, à minuit, le gérant de l'agence vide la table *Résa* (et en reporte les informations dans d'autres tables, non considérées ici). Il vide aussi la table *Libre*, et la remplit à nouveau avec une ligne pour chaque chambre de chaque hôtel. Puis les réservations du jour peuvent avoir lieu. Une réservation à l'hôtel h , pour le client c , de la chambre n , est effectuée en retirant la ligne (h, n) de *Libre*, et en ajoutant la ligne correspondante (h, n, c) dans *Résa*.

Si vous devez gérer un problème et que vous ne connaissez pas la syntaxe exacte des outils Oracle (ou JDBC), utilisez-les de manière intuitive, mais indiquez alors exactement ce qu'ils font.

1. Écrivez une procédure *reserve* prenant en paramètres h , c , et n comme ci-dessus, et effectuant une réservation de chambre (vous ne gérerez pas la réinitialisation des tables par le gérant de l'agence). Elle devra gérer les éventuels problèmes de reprise sur panne et de concurrence le mieux possible. Vous ne devrez faire aucune hypothèse sur l'existence ou non de contraintes Oracle sur les tables, ni en utiliser vous-même. Vous justifierez en expliquant (en quelques lignes) le problème qui se pose si on retire de votre procédure les outils de gestion de reprise et de concurrence, puis pourquoi il disparaît dans votre réponse. Vous utiliserez au choix PL/SQL, ou Java sans les détails de syntaxe.
2. Énoncez en français la contrainte liant les différentes informations concernant les chambres. Est-il possible de la gérer sous Oracle ? Si oui, faites-le concrètement, et dans tous les cas justifiez.

3 Equivalence/inclusion

On considère les requêtes q_i données par :

1. $q_1(x, y)$ est $\exists z R(x, z) \wedge R(z, y)$
2. $q_2(x, y)$ est $\exists z, u, v R(x, z) \wedge R(z, u) \wedge R(x, v) \wedge R(v, y)$
3. $q_3(x, y)$ est $\exists z, u R(x, z) \wedge R(z, u)$

Pour $1 \leq i, j \leq 3$, prouvez $q_i \subseteq q_j$ ou prouvez $q_j \not\subseteq q_i$.

4 Optimisation

Soit le schéma suivant :

R(A,B)
S(B,C)

On désire évaluer la requête suivante :

$\sigma_{A='toto'}(R \bowtie S)$

On s'intéresse à comparer l'efficacité de différents algorithmes de jointure. On dispose des fonctions suivantes. Le parcours séquentiel d'une table se fait à l'aide d'une fonction Scan(.) qui prend en entrée une table et renvoie le tuple suivant de cette table. La première fois que Scan(R) est appelée elle renvoie le premier tuple de R, la seconde fois le second tuple de R, etc. Lorsque tous les tuples de R ont été renvoyés, Scan(R) renvoie la valeur *null*.

La fonction Ajout-Result(.) rajoute son argument au résultat de l'algorithme dans lequel elle est appelée.

Enfin, si *p* est un tuple, *p.A* est son champ *A*.

1) On considère l'algorithme suivant (avec une syntaxe à la Java) :

A0 :

```
do {
    p = Scan(R);
    if (p.A == 'toto')
        do {
            q = Scan(S);
            if (q.B == p.B)
                Ajout-Result(p.A, p.B, q.C);
        }
        while (q != null);
}
while (p != null);
```

Quelle est la complexité en fonction du nombre de tuples dans R et dans S de l'algorithme dans le cas le pire?

2) On introduit maintenant un index sur l'attribut A de R, un autre sur l'attribut B de R et un dernier sur l'attribut B de S. Chacun de ces index est un arbre B+.

On a maintenant une fonction Scan-Index() qui prend en entrée une table et une valeur pour un attribut de cette table et retourne, en utilisant l'index approprié, successivement les tuples de la table ayant la bonne valeur sur l'attribut. Ainsi le premier appel à Scan-Index(R,A=x) retourne le premier tuple de R ayant x pour attribut A, le second appel à Scan-Index(R,A=x) retourne le second tuple de R ayant x pour attribut A etc. Lorsque tous les tuples ont été renvoyés, Scan-Index(R,A=x) retourne la valeur NULL.

On considère les algorithmes suivants :

A1 :

```
do {
    p = Scan(R);
    if (p.A == 'toto')
        do {
            q = Scan-Index(S, B=p.B);
            Ajout-Result(p.A, p.B, q.C);
        }
        while (q != null);
}
while (p != null);
```

```

A2 :
do {
    p = Scan-Index(R, A='toto');
    do {
        q = Scan-index(S, B=p.B);
        Ajout-Result(p.A, p.B, q.C);
    }
    while (q != null);
}
while (p != null);

A3 :
do {
    p = Scan(S);
    do {
        q = Scan-Index(R, B=p.B);
        if (q.A == 'toto')
            Ajout-Result(q.A, p.B, p.C);
    }
    while (q != null);
}
while (p != null);

A4 :
do {
    p = Scan(S);
    do {
        q = Scan-Index(R, A='toto');
        if (q.B == p.B)
            Ajout-Result(q.A, p.B, p.C);
    }
    while (q != null);
}
while (p != null);

```

On considère les situations suivantes (et pour chacune il n'y a aucune autre contrainte).

- A='toto' est un critère très sélectif (c'est-à-dire satisfait par très peu de lignes de R).
- Dans R il ne peut y avoir deux lignes avec la même valeur de B, et on ne sait rien sur la sélectivité de A='toto'.
- Dans S il ne peut y avoir deux lignes avec la même valeur de B, et on ne sait rien sur la sélectivité de A='toto'.
- Dans S il ne peut y avoir deux lignes avec la même valeur de B, et A='toto' n'est pas sélectif (satisfait par 50% des lignes de R).

Pour chacune de ces situations, donnez pour chaque algorithme sa complexité et dites lequel est le meilleur.