

Examen – Bases de Données Avancées – L3 Informatique

pages : 1-2

Exercice 1 (2 points)

Soit un Buffer Manager (BM) qui dispose de 3 cases (frames) numérotées de 1 à 3. Supposons des PageId de la forme PageN, avec $N \geq 1$ entier. Le BM traite donc des demandes de la forme :

- Get (PageN)

- Free (PageN, dirty) avec dirty = 1 si la page a été modifiée, 0 autrement.

Si, lors d'une demande Get nécessitant le chargement d'une page depuis le disque, le BM dispose de cases vides, le BM utilise la première de ces cases (celle du plus petit indice).

Soit la série de demandes : Get(Page1) ; Get(Page15) ; Free(Page1, 1) ; Get(Page7) ; Get(Page1) ; Get(Page15) ; Free(Page15, 1) ; Free(Page7, 0) ; Free(Page1, 0) ; Get(Page3) ; Get(Page1).

Quelle est la première case dont le contenu est remplacé ? Est-ce que ce contenu est écrit sur le disque ? Justifiez :

- Dans le cas d'une politique de remplacement LRU (« le moins récemment utilisé »).
- Dans le cas d'une politique de remplacement Clock (« horloge »).

Exercice 2 (4 points)

Soit un index B+Tree d'ordre 1, initialement vide.

Insérez dans cet index les entrées de données suivantes : 15*, 27*, 16*, 23*, 21*, 49*.

Montrez l'état de l'index après chaque insertion.

Exercice 3 (4 points)

Soit un hash index extensible dont la taille maximale d'une bucket est 2, initialement vide, qui utilise la fonction de hachage (hash function) suivante :

x	h(x)
15	10101011
27	00000110
16	11010110
23	01000101
21	11110001
49	11110010

Insérez dans cet index les entrées de données suivantes : 23*, 21*, 15*, 27*, 49*, 16*.

Montrez l'état de l'index après chaque insertion.

Exercice 4 (4 points)

Soient les relations A, B, C, D de schémas suivants (nous omettons le type des attributs):

A(A1, A2, A3, A4)

B(B1, B2, B3, B4)

C(C1, C2, C3, C4)

D(D1, D2, D3, D4)

4.1. (2.5 points)

Soit l'expression A.R. (Algèbre Relationnelle) suivante :

$\Pi_{C4} (\sigma_{A4=15 \wedge B3 < 10} (B \bowtie_{B1=A2 \wedge B2=C3} (C \bowtie_{A3=C2} A)))$

Transformez cette expression en une expression A.R. équivalente, en poussant les sélections (σ) et projections (Π) « en bas » autant que possible (appliquer donc les sélections et projections le plus tôt possible). Montrez chaque étape de la transformation (1 étape = application d'une règle d'équivalence).

4.2. (1.5 points)

Soit l'expression A.R. (Algèbre Relationnelle) suivante :

$(B \bowtie_{B1=D3} D) \bowtie_{A1=B3 \wedge C2=D2 \wedge B2=C3 \wedge A4=D4} (A \bowtie_{A2=C1} C)$

Trouvez une expression A.R. équivalente qui correspond à un left-deep join tree (niveaux de l'arbre numérotés de 0 = la racine à h = profondeur maximale) comprenant comme feuilles :

- au niveau h : C et D avec D « frère à droite de C »
- au niveau h-1 : A
- au niveau h-2 : B.

Exercice 5 (4 points)

Soit les relations A et B aux caractéristiques suivantes :

- A : schéma A(A1, A2), 30 tuples / page
- B : schéma B(B1, B2), 50 tuples / page.

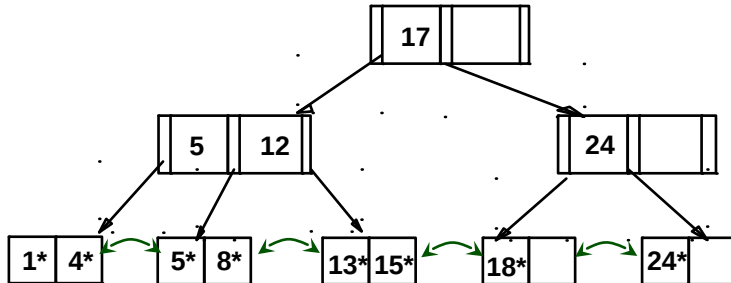
Supposons une instance de A avec 3000 tuples, dont 480 ont la valeur 1 sur A1, et le reste ont la valeur 8 sur A1. Supposons une instance de B avec 150 tuples, dont 100 ont la valeur 1 sur B1, et le reste ont la valeur 5 sur B1. Supposons une jointure de A et B sur A1=B1. Rappel : nous calculons le coût d'un algorithme de jointure en I/Os et en ignorant le coût de l'écriture du résultat final de la jointure.

Quel est le coût de la jointure pour ces deux instances en utilisant :

- a. Un algorithme simple de jointure à boucles imbriquées (simple nested loops join) avec B comme relation externe ?
- b. Un algorithme de jointure à boucles imbriquées « orienté pages » (page-oriented nested loops join) avec A comme relation externe ?
- c. Un algorithme de jointure à base de hachage (hash join) utilisant pour le partitionnement la fonction $h(x) = x \bmod 3$ si on dispose d'un total de 4 buffers (rappel : n'oubliez pas le buffer réservé pour le résultat) et :
 - c1. Avec A comme relation « externe » (= ses partitions sont chargées « en premier »)?
 - c2. Avec B comme relation « externe » ?

Exercice 6 (2 points).

Soit l'index B+Tree d'ordre 1 suivant :



Montrez chaque étape dans l'évolution de l'index (y inclus l'état final) suite à la suppression de l'entrée 18*.