

MI034 - Bases de Données Réparties

Examen du 3 juin 2014

Version CORRIGE

<p style="text-align: center;">Les documents de cours, TD et TME sont autorisés Durée : 2h.</p> <p style="text-align: center;">Les téléphones et tout autre appareil doivent être éteints et rangés.</p> <p>Répondre aux questions sur la feuille du sujet dans les cadres appropriés. La taille des cadres suggère celle de la réponse attendue. Justifier vos réponses. La qualité de la rédaction sera prise en compte. Le barème est donné à titre indicatif. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet.</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 5px;">Ex1 :</td></tr> <tr><td style="padding: 5px;">Ex2 :</td></tr> <tr><td style="padding: 5px;">Ex3 :</td></tr> <tr><td style="padding: 5px;">Ex4 :</td></tr> </table>	Ex1 :	Ex2 :	Ex3 :	Ex4 :
Ex1 :					
Ex2 :					
Ex3 :					
Ex4 :					

Exercice 1 : Conception de bases de données réparties	5 pts
--	--------------

Question 1. On considère une application gérant différents casinos de France, dont le schéma relationnel est le suivant (les clés primaires sont soulignées) :

- CASINO(casino, ville, catégorie)
- GEOGRAPHIE(ville, région)
- SALLE(casino, salle, capacité, chiffre_d_affaire)
- JEU(jeu, maxJoueurs, minJoueurs, description)
- INSTALLATION(jeu, casino, salle, gainMax)
- PERSONNEL(numéro, nom, prénom, poste, casino, salle, salaire)

Un casino est identifié par un nom et se caractérise par une ville et une catégorie (populaire, luxe, grand luxe, etc). Un casino possède plusieurs salles de différentes capacités (faible, moyen, élevé) identifiées par leur numéro. Un jeu est désigné par son nom et a un nombre de joueurs minimal et maximal autorisé. Dans chaque salle on peut jouer à plusieurs jeux, chaque jeu ayant un gain maximum autorisé. Un casino emploie du personnel qu'il affecte à une salle bien définie. Chaque employé est embauché sur un poste bien défini ('surveillance', 'ménage', 'gestion de jeux', etc.). Un employé du casino est identifié par un numéro de personnel.

La base de données est répartie sur plusieurs sites. Il existe un site par ville (dénnoté S_i , i étant le nom de la ville), qui gère les casino (salles, organisation du personnel et installations), et un site par région (dénnoté S_j , j étant le nom de la région) qui gère les salaires et l'affectation des personnels aux différents casinos de la région.

Donner la définition des différents fragments en utilisant les opérateurs de l'algèbre relationnelle ainsi que le schéma d'allocation des fragments (pour chaque relation R de la base, R_i indique le fragment alloué à une ville, R_j le fragment alloué à une région). Chaque site ne doit stocker que les informations dont il a besoin. Justifiez vos réponses en expliquant pourquoi notamment pourquoi les données du fragment sont utiles sur le site considéré. La fragmentation n'est pas forcément disjointe. Un fragment vide est noté \emptyset .

<p>Fragments alloués aux sites des villes :</p> <p>CASINO_i =</p> <p>Justification :</p> <p>GEOGRAPHIE_i =</p> <p>Justification :</p>
--

SALLE_i =

Justification :

JEU_i =

Justification :

INSTALLATION_i =

Justification :

PERSONNEL_i =

Justification :

Fragments alloués aux sites des régions :

CASINO_j =

Justification :

GEOGRAPHIE_j =

Justification :

SALLE_j =

Justification :

JEU_j =

Justification :

INSTALLATION_j =

Justification :

PERSONNEL_j =

Justification :

Réponse :

- fragments alloués aux villes :

$$\text{CASINO}_i = \sigma_{\text{ville} = "i"}(\text{CASINO})$$

$$\text{SALLE}_i = \text{SALLE} \bowtie \text{CASINO}_i$$

$$\text{INSTALLATION}_i = \text{INSTALLATION} \bowtie \text{SALLE}_i$$

$$\text{PERSONNEL}_i = \prod_{\text{numero, nom, prenom, poste, salle}} \text{PERSONNEL} \bowtie \text{SALLE}_i$$

$$\text{GEOGRAPHIE}_i = \emptyset$$

$$\text{JEU}_i = \text{JEU} \bowtie \text{INSTALLATION}_i$$

- fragments alloués aux régions :

$$\text{GEOGRAPHIE}_j = \sigma_{\text{region} = "j"}(\text{GEOGRAPHIE})$$

$$\text{CASINO}_j = \text{CASINO} \bowtie \text{GEOGRAPHIE}_j$$

$$\text{INSTALLATION}_j = \text{SALLE}_j = \text{JEU}_j = \emptyset$$

$$\text{PERSONNEL}_j = \prod_{\text{numero, nom, prenom, salaire, casino}} \text{PERSONNEL} \bowtie \text{CASINO}_j$$

$$\text{JEU}_i = \text{JEU} \bowtie \text{INSTALLATION}_i$$

Exercice 2 : Indexation**5 points**

On désire créer une relation avec un index de type arbre B+ sur l'attribut A. Les caractéristiques sont :

- La relation contient 100 000 n-uplets
- La taille d'un n-uplet est de 100 octets, la taille de l'attribut A est de 10 octets
- La taille d'un pointeur est négligeable (pour simplifier les calculs)
- Les feuilles de l'arbre B+ contiennent les n-uplets
- La taille d'une page est de 2000 octets
- Les n-uplets sont stockés dans un fichier non trié sur A
- La mémoire vive disponible est de 5 000 000 octets.

Question 1 :

On suppose qu'on crée la relation en insérant les n-uplets un par un selon l'algorithme vu en cours et en TD.

1. Décrire par un schéma l'arbre B+ final, en indiquant le nombre minimum de nœuds à chaque niveau et comment le calculer

Solution : au minimum, il y a $(100000 \cdot 100) / 2000 = 5000$ feuilles
Un nœud intermédiaire indexe au maximum $2000 / 10 = 200$ nœuds (en fait 201, mais bof)
Il faut donc au minimum $5000 / 200 = 25$ nœuds intermédiaires, qui sont indexées eux-même par la racine.

2. Donner un ordre de grandeur du nombre d’E/S de page nécessaires pour l’opération totale, parmi les ordres de grandeur suivants (entourer la bonne réponse) et justifier.

- a. Moins de 10 000
- b. Entre 10 000 et 50 000
- c. Entre 50 000 et 300 000
- d. Supérieur à 300 000

Justification :

Solution :

1. on doit lire toutes les pages : 5000 L
2. on doit écrire toutes les pages de l'arbre : 5026 E
3. comme les nuplets ne sont pas triés, chaque insertion nécessite de lire 3 pages (dès qu'il y a trois niveaux, donc a partir de 200 pages de n-uplets) et d'en écrire une : 300 000 L + 100 000 E
4. Chaque création de page nécessite une écriture de la page d'index au niveau au dessus : 5000 E
5. Conclusion, on est largement au dessus de 300 000

Question 2 :

Au lieu d'utiliser la procédure normale d'insertion, on décide de mettre au point une procédure spécifique d'insertion groupée (bulk load) qui insère d'un coup tous les n-uplets en commençant par créer les feuilles. Donner l'algorithme de cette procédure en pseudo-code et estimer le nombre d'E/S de page nécessaires pour l'opération totale. Comparer cette solution avec celle de la question 1 et conclure.

Algorithme :

Estimation du nombre d'E/S et justification :

Comparaison avec la procédure de la question 1 et conclusion :

Sol :

1. Tri/fusion de R

Memoire vive : 2500 pages. R = 5000 pages

Chargement de 50 000 n-uplets (5 mega, 2500 Pages)), tri et ecriture => $2500 * 2 = 5000$ E/S

On recommence avec le reste : 2500 L + 1250 E (on garde les 1250 premières en mémoire)

On charge les 1250 premières pages du premier paquet, puis fusion pour obtenir les feuilles : 3750 L + 5000 E

2. Reste à créer les nœuds :

a. Pour indexer 5000, il faut $5000 / (2000 / 10) = 25$ nœud

b. +1 racine pour indexer.

En tout, le bulk load coute donc 17576 E/S. Conclusion, bien meilleur que pour question 1.

Exercice 3 : Questions de cours

3 points

Répondre aux questions suivantes en argumentant votre réponse.

Question 1 :

Quelle propriété est garantie par la validation en deux phases ?

Question 2 :

Quel type d'opération est favorisé par la réplication ? Pourquoi ? Quel type d'opération est défavorisé par la réplication ? Pourquoi ?

Favorisé :

Explication :

Défavorisé :

Explication :

Question 3 : Donner les avantages et inconvénient des vues matérialisées par rapport aux vues non-matérialisées

Avantages :

Inconvénients :

Exercice 4 : Optimisation de requêtes réparties

7 pts

Soit $R1(a, b)$ sur le site $S1$, $R2(a, c)$ sur $S2$, $R3(a, d)$ sur $S3$.

La requête $Req = R1 \bowtie_a R2 \bowtie_a R3$ est posée sur $S4$. Le schéma du résultat de la requête est (a, b, c, d) . Les attributs sont des entiers positifs. La distribution des valeurs des attributs est uniforme. La taille, en octets, des attributs a, b, c, d est respectivement 1000, 1000 et 3000, 5000.

Le tableau suivant donne la taille des résultats intermédiaires en Go : (pour simplifier les calculs, on pose $1Go = 10^3 Mo = 10^6 Ko = 10^9$ octets).

Expression	taille (en Go)	Expression	taille (en Go)
R1	60	$R1 \bowtie_a R2 \bowtie_a R3$	30
R2	400	$R2 \bowtie_a R1 \bowtie_a R3$	100
R3	1000	$R3 \bowtie_a R1 \bowtie_a R2$	100
$\pi_a(R1)$	10	$R1 \bowtie_a R2 \bowtie_a R3$	T
$\pi_a(R2)$	20	$\pi_a(R1 \bowtie_a R2 \bowtie_a R3)$	5
$\pi_a(R3)$	50		

Remarque : on nomme T la taille du résultat de la requête Req. Ne pas confondre la **taille** avec la cardinalité.

A FAIRE : Question à ajouter : déterminer les tailles des 3 semi jointures

$$R1 \bowtie_a R2 \bowtie_a R3$$

$$R2 \bowtie_a R1 \bowtie_a R3$$

$$R3 \bowtie_a R1 \bowtie_a R2$$

On peut déduire les tailles des semi jointures à partir des autres éléments de l'énoncé

La cardinalité de la semi jointure de R_i avec les 2 autres relations est proportionnelle à la cardinalité de R_i .

Dans $R1$ il y a deux fois plus de valeurs distinctes de a que dans la semi jointure en effet

$R1$ a 10 M de valeurs distinctes tandis que la semi jointure a seulement 5M de valeurs distinctes

$$\text{Card}(R1 \bowtie_a R2 \bowtie_a R3) = \text{card}(R1) * \text{card}(\pi_a(R1 \bowtie_a R2 \bowtie_a R3)) / \text{card}(\pi_a(R1))$$

La proportionnalité s'applique aussi pour la taille car $R1$ et la semi-jointure ont des nuplets de même taille.

$$t1 = 60 * 5/10 = 30$$

$$\text{Taille de } R1 \bowtie_a R2 \bowtie_a R3 = 30$$

$$t2 = 400 * 5/20 = 100$$

$$t_3 = 1000 * 5/50 = 100$$

Question 1

a) Quelle est la taille en octets d'un nuplet de Req ?

Taille d'un nuplet du résultat $1000 + 1000 + 3000 + 5000 = 10\ 000$

b) Combien vaut $\text{card}(\pi_a(R1 \bowtie_a R2 \bowtie_a R3))$?

$$\text{Card}(\pi_a(\text{Req})) = \text{taille}(\pi_a(\text{Req})) / (1000)$$

$$= \text{taille}(\pi_a(R1 \bowtie_a R2 \bowtie_a R3)) / 1000 = 5 \cdot 10^9 / 1000 = 5 \cdot 10^6$$

c) Combien vaut $\text{card}(R1)$?

$$\text{card}(R1) = 60 * 10^9 / (1000 + 1000) = 30 * 10^6$$

d) Combien vaut $\text{card}(\pi_a(R1))$?

$$\text{On } a : \text{taille}(\pi_a(R1)) = \text{taille_nuplet}(a) * \text{card}(\pi_a(R1)) = 10$$

Donc $\text{card}(\pi_a(R1)) = 10 * 10^9 / 1000 = 10 * 10^6$ cela correspond au nb de valeurs distinctes de a dans $R1$

e) Combien vaut $\text{card}(\sigma_{a=v}(R1))$? Rmq : la valeur v appartient au domaine de $R1.a$

Cette sélection renvoie tous les nuplets ayant la même valeur de a ($=v$)

Avec l'hypothèse d'uniformité, le nb de nuplets par valeur de a

$$\text{card}(R1) / \text{card}(\pi_a(R1)) = 30 \cdot 10^6 / (10 \cdot 10^6) = 3$$

f) L'attribut a est-il une clé de $R3$?

Non, a n'est pas unique. Le nombre de valeurs distinctes de a est plus petit que le nombre de nuplets dans $R3$.

$$\text{Car card}(\pi_a(R3)) = 50 \cdot 10^9 / 1000 = 50 \cdot 10^6$$

$$\text{Card}(R3) = 1000 \cdot 10^9 / 10\,000 = 10^8$$

Question 2. Peut-on calculer la valeur de T à partir des informations données dans l'énoncé ?
Si oui expliquer les étapes du calcul, sinon dites ce qu'il manque afin de calculer T.

On peut calculer T . Pour cela, on procède en 4 étapes

1) le nombre de a distincts dans Req (on l'appelle $Da1$)

$$Da1 = \text{card}(\pi_a(\text{Req})) = \text{taille}(\pi_a(\text{Req})) / 1000$$

$$= \text{Taille}(\pi_a(R1 \bowtie_a R2 \bowtie_a R3)) / 1000 = 5 \cdot 10^9 / 1000$$

$$Da1 = 5 \cdot 10^6$$

2)° Le nombre de nuplets par valeur de a dans R1, R2 et R3 (on les appelle N1, N2, N3)

$$N1 = \text{card}(R1) / \text{card}(\pi_a(R1)) = 30 \cdot 10^6 / (10 \cdot 10^6) = 3 \text{ nuplets de R1 par valeur de a}$$

$$N2 = \text{card}(R2) / \text{card}(\pi_a(R2)) = 10^8 / 20 \cdot 10^6 = 5 \text{ nuplets de R2 par valeur de a}$$

$$\text{avec } \text{card}(R2) = 400 \cdot 10^9 / 4000 = 10^8$$

$$\text{et } \text{card}(\pi_a(R2)) = 20 \cdot 10^9 / 1000 = 20 \cdot 10^6$$

$$N3 = \text{card}(R3) / \text{card}(\pi_a(R3)) = 166,6 \cdot 10^6 / 50 \cdot 10^6 = 3,33 \text{ nuplets de R3 par valeur de a}$$

$$\text{Avec } \text{card}(R3) = 1000 \cdot 10^9 / 6000 = 166,6 \cdot 10^6$$

$$3) \text{Card}(\text{Req}) = Da1 \cdot N1 \cdot N2 \cdot N3 = 5 \cdot 10^6 \cdot 3,33 \cdot 5 \cdot 2 = 25 \cdot 10^7$$

$$4) \text{La taille T} = \text{card}(\text{Req}) \cdot 10000 = 25 \cdot 10^{11} = 2500 \cdot 10^9$$

Question 3

On propose 4 plans P1 à P4 pour évaluer Req. Le coût d'un plan est la **somme des tailles** des transferts. Le coût est exprimé en Go. Quel est le coût de chaque plan ? Pour chaque plan, détailler les étapes de l'évaluation. Détailler tous les transferts de données. Si nécessaire, donner une réponse en fonction de T.

P1) Transférer R1, R2 et R3 vers S4 pour traiter la jointure sur S4

Plan P1 :

Transf R1 de S1 → S4 :	60
Transf R2 de S2 → S4 :	400
Transf R3 de S3 → S4 :	1000
Jointure du S3	
	Total: 1460 Go

P2) Transférer R1 et R2 vers S3 pour traiter la Req sur S3. Envoyer le résultat vers S4

Plan P2 :

Transf : R1 de S1 → S3:	60
Transf : R2 de S2 → S3 :	400
Jointure sur S3 puis transf Req de S3 → S4 :	T
	Total : 460+T = 460+2500 = 2960

P3) Les sites S1, S2 et S3 s'échangent mutuellement leurs listes de valeurs distinctes de a . Puis chaque site effectue une semi jointure (nommée T_i) qu'il envoie à S4. On précise que T_i est la partie de R_i dont les valeurs de a existent dans Req. Puis S4 finit d'évaluer Req.

Plan P3 :

Sur S1, transf : $\pi_a(R1) \rightarrow S2$ et S3:	10 + 10 = 20
Sur S2, : $\pi_a(R2) \rightarrow S1$ et S3:	20 + 20 = 40
Sur S3 $\pi_a(R3) \rightarrow S1$ et S2	50 + 50 = 100
Sur S1, semi jointure $T1 = R1 \bowtie_a \pi_a(R2) \bowtie_a \pi_a(R3) = R1 \bowtie_a R2 \bowtie_a R3$ puis transf T1 vers S4	30
Sur S2, semi jointure $T2 = R2 \bowtie_a \pi_a(R1) \bowtie_a \pi_a(R3) = R2 \bowtie_a R1 \bowtie_a R3$ puis transf T2 vers S4	100
Sur S3, semi jointure $T3 = R3 \bowtie_a \pi_a(R1) \bowtie_a \pi_a(R2) = R3 \bowtie_a R1 \bowtie_a R2$ puis transf T3 vers S4	100
Sur S4 : jointure finale	0
Total :	===== 390

P4) S1 et S2 envoient à S3 leur liste de valeurs distinctes de a.

S3 évalue l'intersection $J1 = \pi_a(R1) \cap \pi_a(R2) \cap \pi_a(R3)$. Puis il transmet J1 à S1 et S2.

Chaque site évalue une semi-jointure entre la relation qu'il possède et J1. Le résultat de la semi jointure est transféré à S4. Puis S4 finit d'évaluer Req.

Plan P4 :

Sur S1, Transf $\pi_a(R1) \rightarrow S3$:	10
Sur S2, Transf $\pi_a(R2) \rightarrow S3$:	20
Sur S3, calculer la liste des valeurs de a qui joignent $J1 = \pi_a(R1 \times_a R2 \times_a R3) = \pi_a(R1) \cap \pi_a(R2) \cap \pi_a(R3)$ Transf J1 vers S1 et S2	$2*5 = 10$
Sur S1, semi jointure $T1 = R1 \times_a J1 = R1 \times_a R2 \times_a R3$ puis transf T1 vers S4	30
Sur S2, semi jointure $T2 = R2 \times_a J1 = R2 \times_a R1 \times_a R3$ puis transf T2 vers S4	100
Sur S3, semi jointure $T3 = R3 \times_a J1 = R3 \times_a R1 \times_a R2$ puis transf T3 vers S4	100
Sur S4 : jointure finale	0
	=====
Total :	270