UNIVERSITE PIERRE & MARIE CURIE LA SCIENCE A PARIS

Numéro d'anonymat : | | | |

Systèmes de Gestion de Bases de Données – 3I009

EXAMEN - 1^{ere} session du 11 janvier 2018
Durée : 2 heures – CORRIGÉ
Documents autorisés

Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 21$\frac{1}{2}$ points (23 questions) n'a qu'une valeur indicative.

1 Questions de cours et de TME (3 pts)

Question 1 (1 point)
Exprimez $T = R(XY) \div S(X)$ la division algébrique de la table R(XY) par la table S(X) en fonction des opérateurs de base de l'algèbre relationnelle. Vous pouvez utiliser des variables pour simplifier l'expression

Solution: $A = \Pi_Y(R) \times S$
$B = \Pi_Y(A - R)$
$T = \Pi_Y(R)$

Question 2 (1 point)
Quel est le seul type de jointure accepté par RAEval ? Que doit on faire obligatoirement pour effectuer une jointure d'un autre type ?

Solution: RAEval ne fait que des jointures naturelles. Pour effecuter une jointure non naturelle, il faut renommer des attributs.

Question 3 (1 point)
Comment le SGBD fait-il pour éliminer les doublons lors de l'évaluation d'une requête SELECT DISTINCT ... ? Donner deux solutions possibles

Solution: Une solution est de hacher chaque n-uplet résultat,ainsi les doublons se retrouvent dans le même paquet. Une autre solution est de trier le résultat, les doublons se retrouvent adjacents.

2 Dépendances fonctionnelles (3 pts)

Soit une table **Voyages(N,P,D,R,A,S,T,B)** qui stocke des informations sur des voyages avec le Nom (N) et le Prénom (P) du voyageur, la date de Départ (D), la date de retour (R), l'Adresse de destination (A), le prix du Séjour (S), le moyen de Transport (T) et le prix du Billet (B). On observe l'ensemble de dépendances fonctionnelles suivant sur **Voyages** :

$\mathcal{F}$ = { N P D $\rightarrow$ R A S T B ; D R A $\rightarrow$ S ; D A T $\rightarrow$ B ; D A S $\rightarrow$ R ; R A S $\rightarrow$ D ; }

Question 4 (1 point)
Donnez les deux clés (minimales) de la table Voyages par rapport à $\mathcal{F}$. Expliquer brièvement comment vous les obtenez.

Solution: Les deux clés sont NPD et NPRAS
démarche :
N et P font partie de toutes les clés.
NP+ = NP - pas clé
NPD+ =NPDRATSB - clé
tous les autres ensembles de taille 3 et 4 (sans D) ne sont pas des clés.
NPRAS+=NPRASDTB - clé
tous les autres ensembles de taille 5 (sans D ou sans R ou A ou S) ne sont pas des clé.

Question 5 ($\frac{1}{2}$ point)
Est-ce que la table Voyage est en troisième forme normale (3FN) ? Justifiez votre réponse.

Solution: Non, dans la DF $DAT \rightarrow B$, la partie gauche DAT n'est pas une sur-clé et B ne fait pas partie d'une clé.

Question 6 ($\frac{1}{2}$ point)
Montrez que $\mathcal{F}$ contient au moins une dépendance fonctionnelle redondante et donnez les autres dépendances qui permettent de la retrouver.

Solution: Par exemple la DF $NPD \rightarrow S$ est redondante. On peut la retrouver avec les DF NPD $NPD \rightarrow R$, $NPD \rightarrow A$ et $DRA \rightarrow S$

Question 7 (1 point)
Est-ce que la décomposition suivante de **Voyage** est sans perte d'information (SPI) par rapport à $\mathcal{F}$? Justifiez votre réponse en utilisant le méthode du tableau (montrez quelles DF sont utilisées)
— Reservation(N,P,D,R,A)
— PrixSéjour(D,R,A,S)
— PrixBillet(D,A,T,B)

Solution: Décomposition :
Tableau initial :

	N	P	D	R	A	S	T	B
R	n	p	d	r	a	x16	x17	x18
PS	x21	x22	d	r	a	s	x27	x28
PB	x31	x32	d	x34	a	x36	t	b

N P D $\rightarrow$ R A S T B :
D R A $\rightarrow$ S : x16=s
D A T $\rightarrow$ B :
D A S $\rightarrow$ R :
R A S $\rightarrow$ D :
Tableau final :

	N	P	D	R	A	S	T	B
R	n	p	d	r	a	s	x17	x18
PS	x21	x22	d	r	a	s	x27	x28
PB	x31	x32	d	x34	a	x36	t	b

Il y a aucune ligne complètement définie : la décomposition n'est pas SPI

3 Contrôle de Concurrence (3$\frac{1}{2}$ pts)

Soient les transactions T1 à T6 accédant aux granules A à D. On considère l'exécution S_1 suivante, où L_i et E_i représentent une lecture et une écriture faites par la transaction T_i (on suppose que les

opérations sont exécutées dans l'ordre d'arrivée) :

$$L_1(B)L_6(C)L_2(B)L_1(A)E_4(C)L_3(A)E_6(B)E_2(A)L_1(C)L_3(B)E_5(A)L_5(B)$$

Question 8 (1 point)

Préciser pour chaque granule la séquence d'opérations le concernant et les arcs de précédence, notés $T_i \rightarrow T_j$.

Solution:
- Granule A : séquence : L1.L3.E2.E5 arcs : (T1,T3)->(T2,T5), T2->T5
- Granule B : séquence : L1.L2.E6.L3.L5 arcs : (T1,T2)->T6, T6->(T3,T5)
- Granule C : séquence : L6.E4.L1 arcs : T6->T4, T4->T1

Question 9 ($\frac{1}{2}$ point)

Précisez quels sont les circuits existants dans le graphe de précédence correspondant.

Solution: Circuits :
- T6->T3->T2-T6
- T6->T4->T1->T6
- T6->T4->T1->T2->T6

Question 10 ($\frac{1}{2}$ point)

Choisir une seule transaction à enlever parmi les six afin que l'exécution obtenue S_2 soit sérialisable. Donner l'exécution en série équivalente.

Solution:
- transaction à enlever : T_6
- exécution en série équivalente : T4.T1.T3.T2.T5

Question 11 (1 point)

Soient les transactions T1 à T4 accédant aux granules A à C. On considère le début d'exécution S_3 suivant où **OP** est une opération à préciser. Le gestionnaire de concurrence utilise un verrouillage en 2 phases strict pour contrôler l'exécution des opérations. Au moment de **OP**, toutes les transactions ont encore des opérations à effectuer (aucune des transactions n'a terminé).

$$L_1(A)E_4(B)E_2(C)L_2(B)L_4(B)E_2(B)L_3(A)E_1(A)L_1(B)L_3(C)OP\ldots$$

Donner un exemple d'opération **OP** qui produit un inter-blocage dans lequel toutes les 4 transactions sont impliquées. Dessiner le graphe d'attente

Solution: Exemple : $E_4(A)$
Graphe d'attente : $T_2 \rightarrow T_4 \rightarrow T_1 \rightarrow T_3 \rightarrow T_2$

Question 12 ($\frac{1}{2}$ point)

Donner l'exécution effective des opérations jusqu'à l'inter-blocage.

Solution:

$$L_1(A)E_4(B)E_2(C)L_4(B)L_3(A)$$

4 Algèbre relationnelle (4 pts)

Vol (nvol, villeD, villeA, heureD, heureA, distance) **Appareil** (idAvion, distMin, distMax)
Membre (pid, nom, debutContrat) **Habilité** (pid*, idAvion*, poste)

On considère un schéma relationnel sur les vols, les appareils utilisés et le personnel navigant. Pour chaque vol, identifié par son numéro (nvol), on connait sa ville de départ (villeD), sa ville d'arrivée (villeA), son heure de départ (heureD), son heure d'arrivée (heureD) et la distance parcourue (distance). Pour chaque avion, identifié par un numéro (idAvion), on connait la distance minimale qu'il doit parcourir pour être rentable (distMin) et la distance maximale qu'il peut parcourir (distMax) sans avoir besoin de ravitaillement. Pour chaque membre du personnel, identifié par son pid, on connait son nom et le début de son contrat ; on connait également, avec la table **Habilité**, les appareils dans lesquels il est habilité à naviguer et dans quel poste.

Exprimer en **algèbre relationnelle** les requêtes qui retournent les informations suivantes.
Remarque. Seuls les opérateurs de l'algèbre relationnelle sont autorisés : $\pi, \sigma, \bowtie, \rho, \times, \div, \cup, \cap, -$.

Question 13 (1 point)

Pour chaque membre du personnel, les vols sur lesquels il peut opérer. Un membre peut opérer sur un vol si il est habilitée à voler dans un appareil rentable pour ce vol. Un appareil est rentable pour un vol si la distance du vol est comprise entre la distance minimale et la distance maximale de l'appareil.
Le résultat doit être un ensemble de paires $(pid, nvol)$.

Solution:
- $R_1 = \Pi_{nvol,idAvion}(\sigma_{distance \geq distMin \wedge distance \leq distMax}(Vol \times Appareil))$
  on peut aussi accepter $R_1 = \Pi_{nvol,idAvion}(Vol \bowtie_{distance \geq distMin \wedge distance \leq distMax} Appareil)$
- $R_2 = \Pi_{pid,novl}(Habilite \bowtie R_1))$

Question 14 (1 point)

Les correspondances possibles entre deux vols (v_1, v_2) tels que la ville de départ de v_2 est la ville d'arrivée de v_1 et l'heure de départ de v_2 est supérieure à l'heure d'arrivée de v_1. On retournera des paires de numéros de vols.

Solution:
- $R_1 = \Pi_{nvol,villeA,heureA} Vol$
- $R_2 = \rho_{nvol \to nv, villeD \to villeA}(\Pi_{nvol,villeD,heureD} Vol)$
- $\Pi_{nvol,nv}[\sigma_{heureA < heureD}(R_1 \bowtie R_2)]$

Question 15 (1 point)

Les villes où il n'existe aucun vol ni en partance vers 'Paris' ni en arrivée depuis 'Paris'.

Solution: $R = \Pi_{nvol} Vol - \Pi_{nvol}(\sigma_{villeD='Paris' \vee villeA='Paris'} Vol)$

Question 16 (1 point)

Les identifiants des membres qui sont habilités à voler sur tous les appareils rentables pour un vol de 1000km.

Solution: $R = \Pi_{pid,idAvion} Habilite \div (\Pi_{idAvion}(\sigma_{distMin \leq 1K \wedge distMax \geq 1K} Appareil))$

5 Optimisation de requêtes (5 pts)

On considère le schéma d'une agence de voyage.

Personne (numP, prenom, age, ville, statut, civilité, profil)	*ville fait référence à VillePays.*
VillePays (ville, pays)	*ville est unique.*
Destination (numD, villeD, catégorie)	*villeD fait référence à VillePays.*
Séjour (numP, numD, année, commentaire)	

On connait les cardinalités et les tailles suivantes (ne pas confondre cardinalité et taille) :

Relation	Cardinalité	Rmq	Taille en pages
Personne	2 000		1000 pages
VillePays	4 000	seules 50 villes sont référencées dans Destination	
		seules 100 villes sont référencées dans Personnes	
Destination	100	2 destinations par villeD	
Séjour	20 000	200 séjours par numD et 10 séjours par numP	1000 pages

On connait les domaines des attributs suivants :

Relation	Attribut	Nb valeurs	Rmq
Personne	age	50	dans $[20, 70[$
Personne	ville	100	'Paris', ...
Personne	statut	5	'A', 'B', 'C', 'D' ou 'E'
Personne	civilité	2	'Mme' ou 'Mr'
VillePays	pays	200	'Fr', 'Es', ...
Destination	villeD	50	'Rio', ...
Destination	catégorie	4	'mer', 'montagne', ...
Séjour	numD	100	1, ..., 100
Séjour	année	10	dans $[2008, 2017[$

Question 17 (1 point)

On considère un prédicat s_i et le facteur de sélectivité $SF_i = SF(\sigma_{s_i} Personne)$. Calculer SF_i.

s_1 : age BETWEEN 25 AND 34

s_2 : age < 25 OR age >= 60

s_3 : ville = 'Paris' AND age = 25

s_4 : civilité = 'Mr' OR statut <> 'A' NB. l'opérateur <> signifie **différent de**

Solution:

- SF_1 : 10 valeurs de 25 à 34, $SF_1 = \frac{10}{50}$ = **0.2**
  Erreur fréquente pour : $age \geq 25$ and $age \leq 34$: ~~(70-25)/50 * (35-20)/50 = 0.9 * 0.3 = 0.27~~ C'est faux car les 2 inégalités portent sur le même attribut et ne sont donc pas indépendantes.
- $SF_2 = \frac{5}{50} + \frac{10}{50} = 0.1 + 0.2$= **0.3**
  Erreur fréquente : soustraire le produit $0.1 * 0.2$, c'est faux car les 2 inégalités portent sur le même attribut.
- $SF_3 = \frac{1}{100} * \frac{1}{50} = \frac{1}{5000}$ = **0.0002**
- $SF_4 = \frac{1}{2} + \frac{4}{5} - (\frac{1}{2} * \frac{4}{5}) = 0.5 + 0.8 - 0.4$ = **0.9**
  Erreur fréquente pour : oublier de soustraire les nuplets qui satisfont les 2 prédicats : les personnes de civilité = Mr et de statut= A. Il faut les soustraire car ils sont comptés deux fois : d'une part les personnes de civilité=Mr peuvent avoir un statut=A et d'autre part les personnes de statut=A peuvent être de civilité=Mr.

Question 18 (1 point)

Quelle est la cardinalité $C_i = card(R_i)$ des requêtes suivantes ? Justifier votre réponse.

R_1 = Personne $\bowtie_{ville}$ VillePays

R_2 = $\sigma_{civilite='Mr'}$ (Personne $\bowtie_{numP}$ Séjour)

R_3 = Séjour $\bowtie_{numD}$ Destination $\bowtie_{numP}$ Personne

R_4 = $\sigma_{annee=2016}$ Séjour $\bowtie_{numP}$ Personne $\bowtie_{numD}$ ($\sigma_{villeD='Rio'}$ Destination)

Solution:

Chaque personne réside dans une ville

C_1 = card(Personne) = **2 000**

Chaque séjour est associé à une personne

C_2 = card(Séjour) * 1/2 = 20 000 * 1/2 = **10 000**
Chaque séjour est associé à une destination et une personne
C_3 = card(Séjour) = **20 000**
Rmq, on a $R_4 = \sigma_{annee=2016 and}(\sigma_{villeD='Rio'} R_3))$
C_4 = card(Séjour) * 1/10 * 1/50 = 20 000 * 1 / 500 = **40**

Rmq : variante pour R1 non posée. Ajouter une sélection : R_1'= Personne $\bowtie_{ville}$ ($\sigma_{pays='Fr'}$ VillePays)

Question 19 (1 point)

Par les questions suivantes, tous les coûts sont exprimés en nombre de pages lues. Le coût d'un accès par **index** est $\texttt{coût}(\sigma_{a\ op\ x}(R)) = card(\sigma_{a\ op\ x}(R))$ où op op est un opérateur (=, <, >) et x est une valeur. Cf. cours : lire une page pour chaque nuplet accédé à partir des ROWID obtenus en parcourant l'index.

Pour T_3 et T_4, la jointure est faite par boucles imbriquées et son coût est : $\texttt{coût}(A \bowtie B) = \texttt{coût}(A) + P(A) * P(B)$ avec $P(R)$ étant la taille de R exprimée en nombre de pages.

Pour chaque plan T_i calculer son coût noté $\texttt{coût}(T_i)$.

T_1 : $\sigma_{age=25}\texttt{Personne}$ utilise l'index Personne(âge).

T_2 : $\sigma_{numD=3}\texttt{Séjour}$ utilise l'index sur Séjour(numD).

T_3 : $(\sigma_{age=25}\texttt{Personne}) \bowtie_{numP} \texttt{Séjour}$ utilise l'index Personne(âge).

T_4 : $\sigma_{age=25}((\sigma_{numD=3}\texttt{Séjour}) \bowtie_{numP} \texttt{Personne})$ utilise l'index Séjour(numD).

Solution:

- coût(T_1) avec index sur age : card(T_1) = 2 000 * 1/50 = 40 personnes ⇒ **40** pages lues
- On calcule le nombre de pages de T1 : P(T_1) = P(Personne) * 1/50 = 1000/50 = **20 pages**
- coût(T_2) avec index sur numD : card(T_2) = 20 000 * 1/100 = 200 séjours ⇒ **200** pages lues
- On calcule la taille de T2 : P(T_2) = P(Séjour) * 1/100 = 1000/100 = **10 pages**
- coût(T_3) = $cout(T_1) + P(T_1) * P(Sejour)$ = 40 + 20 * 1000 = **20 040** pages
- Pour T_4 la dernière sélection (age=25) ne nécessite aucun accès aux données, son coût est nul. coût(T_4) = $cout(T_2) + P(T_2) * P(Personne)$ = 200 + 10 * 1000 = **10 200** pages

Question 20 (1 point)

On rappelle qu'une jointure par **hachage** accède à tous les nuplets de A puis à tout ceux de B, A et B pouvant être des opérations de sélection, et son coût est : $\texttt{coût}(A \bowtie B) = \texttt{coût}(A) + \texttt{coût}(B)$.

Soit la requête H_1 composée d'une jointure et deux sélections :

select * **from** Séjour s, Personne p
where s.numP = p.numP **and** s.numD = 3 **and** p.âge = 25

Proposer un plan pour H_1 qui utilise les **index** sur Séjour(numD) et Personne(âge) et évalue la jointure par **hachage**. Détailler les principales étapes de l'exécution de H_1 et donner son coût.

Solution:

Le plan pour H_1 est $(\sigma_{numD=3}\texttt{Séjour}) \bowtie_{numP} (\sigma_{age=25}\texttt{Personne})$.

H_1 peut aussi s'écrire $T_1 \bowtie_{numP} T_2$ avec T_1 et T_2 de la question précédente

Etape 1 : Accès par index aux séjours tq numD =3 (idem T_1) : accès à 200 séjours en 200 lectures. Puis garder les 200 séjours en mémoire dans une table de hachage associant un numP avec les nuplets de Séjour.

Etape 2 : Accès par index aux Personnes tq age=25 (idem T_2) : accès à 40 personnes en 40 lectures. Pour chaque Personne lue, consulter la table de hachage pour obtenir ses séjours.

Cout(H_1) = 200 + 40 = **240**

Question 21 (1 point)

Soit la requête J_1 :

select * **from** Séjour s, Personne p
where s.numP = p.numP **and** s.année = 2016 **and** p.âge = 25

J_1 est évaluée par **boucles imbriquées avec index sur l'attribut de jointure**, et utilise les **index** Séjour(numP) et Personne(age). Expliquer les principales étapes de l'exécution. Donner le coût du plan.

Solution:
J_1 : jointure par boucles imbriquées avec index (cours : index nested loop)

Lire les personne avec l'index age=25 : 40 personnes

Pour chaque personne, accès par index NumP pour lire ses 10 séjours puis filtrer l'année.

cout : 40 + 40 * 10 = 440

Soit la requête J_2 : **select** * **from** Séjour s, Personne p
where s.numP = p.numP **and** s.année = 2016 **and** p.âge > 60

Le plan optimal pour évaluer J_2 a un coût égal à 1400. Préciser les index et l'algorithme de jointure que ce plan utilise (parmi les index et les algorithmes indiqués ci-dessous). Justifier.

Solution: Plan pour J_2 :
Index utilisé : age
Algo : hachage
Justification :
Index pour la sélection age>60 ? OUI
Nombre de personnes sélectionnées : 2 000 * (70-60)/(70-20) = 2 000 * 10/50 = 400. 400 lectures avec l'index. C'est moins que la lecture séquentielle. On utilise l'index age.
Index pour la sélection année=2016 ? NON
Nombre de séjours sélectionnés : 20 000 * 1/10 = 2 000. 2000 lectures avec l'index. C'est plus que la lecture séquentielle de 1000 pages.
Jointure par hachage :
somme des couts des accès à Personne et Séjour : 400 + 1000 = 1400

SUITE pour le rattrapage

Autres plans non demandés :
Boucles imbriquée dans l'ordre Personne,Séjour et index Séjour.NumP :
400 + 400 * 10 séjours par personne = 4400

Boucles imbriquée dans l'ordre Séjour,Personne et index Personne.NumP :
1000 + 2000 * 1 personne par séjour = 3000

Boucles imbriquée dans l'ordre Séjour,Personne SANS index Personne.NumP :
Lire les 400 personnes et les stocker temporairement dans 200 pages
400 + 200 = 600
Jointure
1000 + 100 * 200 (pour lire les personnes) = 21 000
total : 21 600

Boucles imbriquée dans l'ordre Personne,Séjour SANS index Séjour.NumP :
lire les 2000 séjours et les stocker temporairement dans 100 pages

1000 + 100 = 1100
Jointure
400 + 200 * 100 = 20 400
total : 21 500

6 Triggers (3 pts)

On considère le schéma relationnel suivant :

CLIENT (numClient, nomClient, adresse)
PRODUITS(numP, nomP, prix, qteStock)
COMMANDE (numC, numClient, dateCom, numP, qte, cout)

Un client est identifié par un numéro *(numClient)*. Il a un nom *(nomClient)* et une adresse *(adresse)*.
Un produit a un numéro *(numP)*, un nom *(nomP)* un prix unitaire *(prix)*. L'attribut *qteStock* donne le nombre de produits en stock.
Une commande a un numéro de commande *(numC)*. Elle est passée à une date *(dateCom)*, pour le client *(numClient)*. Elle concerne un produit *(numP)*. Les attributs *qte* et *cout* indiquent la quantité du produit commandé et le cout total de la commande.

Question 22 ($1\frac{1}{2}$ points)
Lorsqu'on passe une commande, la quantité en stock doit être mise à jour. Ecrire un trigger AFTER gérant la quantité en stock des produits en fonction des commandes.

Solution: create or replace trigger NouvCommande
after insert or update of qte on commande
for each row
declare xx smallint ;
begin
select qtestock into xx from produits where numP = :new.numP ;
if (:new.qte > xx) then begin raise_application_error (-20002, 'plus de stock') ; end ;
else update produits set qtestock = qtestock - :new.qte where nump = :new.numP ;
end if ;
end ;

Question 23 ($1\frac{1}{2}$ points)
Ecrire un trigger BEFORE qui fait une réduction de 10 % aux clients qui achètent plus de 3 produits le même jour (il y a une commande par produit). La réduction s'applique à partir de la troisième commande le même jour.

Solution: create or replace trigger reduc2
before insert on commande
for each row
declare xx smallint ;
begin
select count(*) into xx from commande where numClient = :new.numClient and dateC = :new.dateC ;
if (xx >= 3) then :new.cout := :new.cout * 0.9 ;
end if ;
end ;