UNIVERSITE PIERRE & MARIE CURIE
LA SCIENCE A PARIS

Numéro d'anonymat : | | | |

Systèmes de Gestion de Bases de Données – 3I009

EXAMEN - 1^{ere} session du 14 décembre 2016
Durée : 2 heures – CORRIGÉ
Documents autorisés

Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 22 points (23 questions) n'a qu'une valeur indicative.

1 Questions de cours et de TME (4 pts)

Question 1 (1 point)
Soit le protocole de verrouillage où une transaction demande un verrou avant chaque opération, effectue l'opération après avoir obtenu le verrou, et relâche le verrou après avoir effectué l'opération. Si toutes les transactions d'une exécution suivent ce protocole, la sérialisabilité de l'exécution est-elle garantie ? Justifiez votre réponse (les réponses sans justification ne seront pas prises en compte).

Solution: Non. Il faut encore ne pas relâcher le verrou trop tôt. Si on relâche immédiatement après avoir effectué l'opération, cela permet n'importe quelle exécution, donc aussi des exécutions non-sérialisables.

Question 2 (1 point)
Quel avantage et quel inconvénient y a-t-il à stocker et maintenir un histogramme, pour un attribut d'une relation, permettant de connaître, pour un intervalle donné, le nombre d'occurences (valeurs) de la relation appartenant à cet intervalle ?

Solution: L'avantage est qu'on peut mieux estimer un taux de sélectivité sur cet attribut (accélère les requêtes en lecture). L'inconvénient est qu'il faut mettre à jour cet histogramme à chaque update/delete/insert (ralentit les écritures).

Question 3 (1 point)
A) A quoi servent principalement les triggers 'INSTEAD OF' ? **B)** Sous Oracle, pour un trigger INSTEAD OF concernant une mise à jour (UPDATE), est-il possible de déclarer sur quel attribut porte la mise à jour ? **C)** Quelle difficulté rencontre-t-on sur la conception d'un tel trigger (trigger INSTEAD OF pour UPDATE) ?

Solution: A)Ils servent a mettre à jour des vues que le SGBD refuse car il ne sait pas répertucer sur les tables de base. B) Non, c'est impossible. C) Cela oblige a définir un trigger qui fait ce qu'il faut quelque soit l'attribut modifié.

Question 4 (1 point)
Expliquer en quelques lignes la différence entre les mode RULE et CHOOSE de l'optimiseur d'Oracle.

Solution: Le mode RULE se base sur des règles heuristiques en fonction des comparateurs (égalité, inégalité) et des indexs disponibles. Le mode CHOOSE évalue les coûts (en fonction de la taille des relations, des index dispo, des taux de sélectivité estimés) et choisit le moindre.

2 Dépendances fonctionnelles (4 pts)

Soit une table **Tournoi(Sp, Eq, Tc, Ag, Pi, Pe, Vi, Ve)** qui stocke les résultats d'un tournoi de gymnastique artistique avec le nom des sportifs Sp, le nom des équipes Eq, le type de concours Tc (homme ou

femme), les différents agrès Ag (saut de cheval, barres asymétrique, poutre et sol pour les femmes et barre fixe, anneaux, cheval d'arçon, sol, saut de cheval, barres parallèles pour les hommes), les points obtenus en individuel Pi et en équipe Pe ainsi que pour chaque agrès et type de concours le vainqueur en individuel Vi et en équipe Ve. On observe l'ensemble de dépendances fonctionnelles suivant sur **Tournoi** :

$\mathcal{F}$ = { Sp→ Eq, Tc ; Eq→ Tc ; Sp, Ag→ Pi ; Eq, Ag→ Pe ; Tc, Ag→ ViVe }

Question 5 ($\frac{1}{2}$ point)
Expliquez le sens de la dépendance fonctionnelle $Ag \rightarrow Tc$ et pourquoi n'existe-t-elle pas sur la table **Tournoi** ?

Solution: La DF signifie qu'en connaissant l'agrès, on connait le type de concours. L'agrès "sol" est exécuté par les hommes et les femmes.

Question 6 (1 point)
Est-ce que la table Tournoi est en troisième forme normale (3FN) ? Justifiez votre réponse.

Solution: Spet Agfont partie de toutes les clés.
[Sp, Ag]+=Sp, Ag, Pi, Eq, Tc, Pe, Vi, Ve= tous les attributs de Tournoi
(Sp, Ag) est la seule clé.
Dans la DF Sp→ Eq, la partie gauche Sp n'est pas une sur-clé et Eq ne fait pas partie d'une clé.

Question 7 ($\frac{1}{2}$ point)
Est-ce que $\mathcal{F}$ est un ensemble de dépendances minimal ? Justifiez votre réponse.

Solution: $\mathcal{F}$ n'est pas minimal, parce que Sp→ Tc est redondant.

Question 8 ($1\frac{1}{2}$ points)
Est-ce que la décomposition suivante de **Tournoi** 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)
— Vainqueurs(Tc, Ag, Vi, Ve)
— Sportifs(Sp, Eq, Tc)
— PointsInd(Sp, Ag, Pi)
— PointsEqu(Eq, Ag, Pe)

Solution: Décomposition :
Tableau initial :

	Sp	Eq	Tc	Ag	Pi	Pe	Vi	Ve
V	a1	b1	C	D	e1	f1	G	H
S	A	B	C	d2	e2	f2	g2	h2
P1	A	b3	c3	D	E	f3	g3	h3
P2	a4	B	c4	D	e4	F	g4	h4

- Sp→ Eq : b3=B
- Eq→ Tc : c3=c4=C
- Sp, Ag→ Pi :
- Eq, Ag→ Pe : f3=F
- Tc, Ag→ Vi, Ve : g3=g4=G, h3=h4=H

Tableau final :

	Sp	Eq	Tc	Ag	Pi	Pe	Vi	Ve
V	a1	b1	C	D	e1	f1	G	H
S	A	B	C	d2	e2	f2	g2	h2
P1	A	B	C	D	E	F	G	H
P2	a4	B	C	D	e4	F	G	H

Il y a une ligne complètement définie : la décomposition est SPI

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

Est-ce que la décomposition précédente est sans perte de dépendance ? Si ce n'est pas le cas, donnez la dépendance fonctionnelle perdue.

Solution: La décomposition est SPD :
- Sp$\rightarrow$ Eq : projetée dans Sportifs
- Eq$\rightarrow$ Tc : projetée dans Sportifs
- Sp, Ag$\rightarrow$ Pi : projetée dans PointsInd
- Eq, Ag$\rightarrow$ Pe : projetée dans PointsEqu
- Tc, Ag$\rightarrow$ ViVe : projetée dans Vainqueurs

3 Indexation : tables de hachage (4 pts)

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

On considère une indexation à l'aide de tables de hachage extensibles, dans lesquelles un paquet contient au plus 4 valeurs. Les paquets sont identifiés par les lettres A, B, C... .
PG et PL signifient respectivement profondeur globale et profondeur locale. Ex : A (1, 2) PL=2 indique que le paquet A contient les valeurs 1 et 2 et que sa profondeur locale est égale à 2.
On considère l'état T1 d'une table de hachage, contenant les paquets suivants :

A (12, 40, 28, 60), B (5, 45, 29), C (10, 50), D (51, 7, 47)

Donnez le répertoire et les valeurs de PG et PL pour cette table.

Solution:
Le répertoire R a 4 entrées (R[A,B,C,D]) et PG = 2 PL = 2 pour tous les paquets.

Question 11 ($1\frac{1}{2}$ points)

On insère successivement dans T1 les valeurs 39, puis 32. Soit T2 l'état de la table de hachage après insertion de ces valeurs. Représentez le répertoire de T2 et uniquement les paquets modifiés, insérés ou supprimés par cette mise à jour, en donnant la profondeur globale et la profondeur locale de chaque paquet représenté.

Solution: Insertion de 39 (39 mod 4 = 3) insertion dans D, qui devient D(51, 7, 47, 39) (0,5pt) Insertion de 32 (32 mod 4 = 0) : doit aller dans A, mais plus de place. On double le répertoire, et on répartit les valeurs de A entre A et A2 (R[4]).
On a R[A,B,C,D,A2,B,C,D] PG= 3
A(32, 40) PL=3
B (5, 45, 29) PL= 2
C (10, 50) PL = 2
D (51, 7, 47, 39) PL = 2
A2 (12, 28, 60) PL=3

Question 12 (2 points)

On considère maintenant l'état T3 de la table de hachage, de répertoire R[A,B,C,D,A,B,C2,D] et contenant les paquets suivants : A(16, 20, 48, 32), B(45, 29), C(10, 18), D(43,15), C2(22, 14, 46).

On supprime successivement dans T3, les valeurs 10, 15, 22, 43, puis 18. Représentez l'état T4 de la table de hachage après suppression de ces valeurs en donnant le répertoire, la profondeur globale et les paquets avec leur profondeur locale.

Solution: Suppression de 10 : C(18) Suppression de 15 : D devient D(43) (pas de fusion) Suppression de 22 : C2 devient C2(14,46). Suppression de 43 : D devient vide. Suppression de 18 : C devient vide. On le fusionne avec C2. On réduit le répertoire, PG = 2 et R[A,B,C,D](même si pas obligatoire dans la pratique)
Table finale : A(16,20,48,32) B(45, 29) C2(14,46) D() PG = 2, PL = 2 partout.

4 Transactions et concurrence (3 pts)

On considère T_1, T_2, T_3 trois transactions et *x, y, z* trois granules d'une base de données. On note :
— $l_i(g)$ la lecture de la transaction T_i du granule g
— $e_i(g)$ l'écriture de la transaction T_i du granule g
— c_i l'opération de validation de la transaction T_i

Question 13 (1 point)
On considère la séquence S_1 suivante :

$$S_1 = l_2(x), l_1(z), e_1(x), e_2(y), l_1(y), e_3(x), l_2(z), l_3(x), c_2, e_1(z), c_1, e_3(y), c_3$$

On suppose que les opérations sont exécutées dans l'ordre indiqué.
Préciser pour chaque granule la séquence d'opérations qui le concerne ainsi que les arcs de précédence $T_i \longrightarrow T_j$.

Solution:
Pour x : l2 e1 e3 l3. $T_2 \longrightarrow T_1 \longrightarrow T_3, T_2 \longrightarrow T_3$
Pour y : e2 l1 e3. $T_2 \longrightarrow T_1 \longrightarrow T_3, T_2 \longrightarrow T_3$
Pour z : l1 l2 e1. $T_2 \longrightarrow T_1$
graphe acyclique. donc sérialisable.

Donner le graphe de précédence. Cette séquence est-elle sérialisable ? Si oui, donner l'exécution en série équivalente.

Solution: $T_2 \longrightarrow T_1 \longrightarrow T_3, T_2 \longrightarrow T_3$
graphe acyclique. donc sérialisable.
ordre en série : T_2, T_1, T_3

Question 14 ($\frac{1}{2}$ point)
Soit S_2 la séquence d'opérations donnée comme suit

$$S_2 = l_1(z), l_2(x), e_2(y), e_1(x), l_1(y), e_3(x), l_3(x), e_3(y), c_3, l_2(z), e_1(z), c_2, c_1$$

On suppose que les opérations sont exécutées dans l'ordre indiqué.
Préciser pour chaque granule la séquence d'opérations qui le concerne ainsi que les arcs de précédence $T_i \longrightarrow T_j$.

Solution: Pour x : l2 e1 e3 l3. $T_2 \longrightarrow T_1 \longrightarrow T_3, T_2 \longrightarrow T_3$
Pour y : e2 l1 e3 . $T_2 \longrightarrow T_1 \longrightarrow T_3, T_2 \longrightarrow T_3$
Pour z : l1 l2 e1. $T_2 \longrightarrow T_1$
graphe **a**cyclique. donc sérialisable pour les conflits.

Donner le graphe de précédence. Cette séquence est-elle sérialisable ? Si oui, donner l'exécution en série équivalente.

Solution: même réponse que la précédente :
$T_2 \longrightarrow T_1 \longrightarrow T_3, T_2 \longrightarrow T_3$
graphe acyclique. donc sérialisable.
ordre en série : T_2, T_1, T_3

Question 15 ($\frac{1}{2}$ point)
Les exécutions S_1 et S_2 sont elles équivalentes ? **Justifiez votre réponse**.

Solution: les deux exécutions sont équivalentes au même ordre en série, donc elles sont équivalentes.

Question 16 (1 point)
Soit S_3 la séquence d'opérations donnée comme suit

$$S_3 = l_1(x), l_2(x), l_1(y), e_2(y), l_3(x), e_1(x), l_3(y), c_1, c_2, c_3$$

On voudrait appliquer le protocole de verrouillage en deux phases strict (2PL strict). Quel est l'état de la table de verrous **juste après l'arrivée de** $e_1(x)$?Remplir la liste des verrous exclusifs "Verrou X" et la liste de verrous partagés "Verrou P"avec des transactions T_i, remplir la colonne Attente avec des X_i et P_i pour une demande de verrou X et P respectivement.

Solution: Table de verrous :

Granule	Verrou X	Verrou P	Attente
x		T_1, T_2, T_3	X_1
y		T_1	X_2
z			

Dessiner le graphe d'attente **juste après l'arrivée de** $e_1(x)$.

Solution: interblocage. $e_2(y)$ bloquée par $l_1(y)$, $e_1(x)$ bloquée par $l_2(x)$ et $l_3(x)$.

5 Algèbre relationnelle (4 pts)

On considère un schéma relationnel de la régie des transports d'une ville donnée.

Arrond (numAr, population, maire) **Station** (nomSt, dateOuverture, numAr*)
Ligne (numLi, couleur, stationA*, stationB*) **Corresp** (numLi1*, numLi2*, nomSt*, distance)

Les attributs soulignés représentent les clés primaires, les attributs avec astérisque représentent les clés étrangères. La relation **Arrond** contient des informations de base sur chaque arrondissement : son numéro, sa population et le nom de son maire. La relation **Station** indique pour chaque station son nom, la date de son ouverture et l'arrondissement où elle se situe . La relation **Ligne** indique pour chaque ligne sa couleur ainsi que les deux stations en bout de parcours (StationA et StationB). Par exemple, pour la ligne 1, on aura le n-uplet (1, 'jaune', 'La défense', 'Vincennes') pour désigner que la ligne 1 relie les stations 'La défense' et 'Vincennes'. La relation **Corresp** renseigne pour chaque paire de lignes qui se croisent la ou les stations de correspondance ainsi que la distance à parcourir d'une ligne à l'autre. Pour simplifier les requêtes, cette relation est symétrique. Par exemple, s'il existe une correspondance entre les lignes 4 et 6 aux stations Denfert et Montparnasse avec des distances de 10 et 20 respectivement, on aura les quatre n-uplets suivants : (4, 6, 'Denfert', 10), (4, 6, 'Montparnasse', 20), (6, 4, 'Denfert', 10) et (6, 4, 'Montparnasse', 20).

Exprimer en algèbre relationnelle les requêtes qui permettent de retourner les informations suivantes.

Question 17 (1 point)
Les lignes (numLi) dont l'une des stations de bout de parcours se situe dans un arrondissement de plus de 100 habitants.

Solution:
$R_1 = \sigma_{population>100} Arrond \bowtie Station \bowtie_{StationA=nomSt} Ligne$
$R_2 = \sigma_{population>100} Arrond \bowtie Station \bowtie_{StationB=nomSt} Ligne$
Le résultat est $\pi_{numLi} R_1 \cup \pi_{numLi} R_2$.

Question 18 (1 point)
Les lignes (numLi) qui se croisent dans au moins deux stations différentes avec une distance de parcours de plus de 50 à chaque croisement.

Solution:
$R = \pi_{numLi1,numLi2}[\sigma_{nomSt \neq St \wedge distance>50 \wedge dist>50}(Corresp \bowtie \rho_{nomSt \leftarrow St, distance \leftarrow dist} Corresp)]$
Remarque : les plus forts élimineront les "doublons" avec $\sigma_{numLi1<numLi2}(R)$ mais ceci n'est pas requis.

Question 19 (1 point)
Les arrondissements (numAr) de plus de 100 habitants qui ne contiennent aucune station de bout de parcours.

Solution:
soient R_1 et R_2 de la solution à la question 1. La réponse est : $\pi_{numAr}(\sigma_{population>100} Arrond) - [\pi_{numAr} R_1 \cup \pi_{numAr} R_2]$

Question 20 (1 point)
Les stations (nomSt) où se croisent toutes les lignes de transport.

Solution:
$\pi_{nomSt,numLi1} Corresp \div \pi_{numLi1} Corresp$

6 Optimisation de requêtes (3 pts)

On considère le schéma des abonnés à un service de transport en commun. Les clés sont soulignées. Les attributs station, stationMaison, stationTravail font référence à une table omise dans l'énoncé.

Valider (navigo, station, date, jour, heure)

Abonné (navigo, stationMaison, stationTravail, catégorie)

LigneStation (ligne, station)

Il y a 500 stations distinctes et 7 jours distincts 1 à 7. Les 20 valeurs distinctes pour l'heure sont les entiers de 5 à 24 inclus.

Il y a 500 valeurs distinctes pour stationMaison (idem pour stationTravail), et 100 catégories.

Une ligne de transport est associée à plusieurs stations. Une station est associée à plusieurs lignes de transport.

Le relation Valider contient 1 million de nuplets ; Abonné contient 10 000 nuplets ; LigneStation contient 2000 nuplets.

Question 21 (1 point)
Pour chaque prédicat de sélection s1 à s4, quel est son facteur de sélectivité (SF) ? Répondre en détaillant la **formule** et le résultat.
s1 : $heure >= 8$ AND $heure < 18$
s2 : $heure <= 7$ OR $heure = 24$
s3 : $station =' Jussieu'$ AND $jour = 1$
s4 : $stationMaison =' cdg2'$ OR $stationTravail =' jussieu'$

Solution:
s1 : $heure >= 8$ AND $heure < 18$: 10 valeurs de 8h à 18h, parmi 20 valeurs : 10/20= **0,5**
s2 : $heure <= 7$ OR $heure = 24$: les heures sélectionnées sont 5,6,7,24 soit 4 valeurs parmi 20 : 4/20= **0,2**
s3 : $station =' Jussieu'$ AND $jour = 1$: 1 station parmi 500 (1/500) * 1 jour parmi 7 (1/7) = **1/3500** = 0,00028
s4 : $stationMaison =' cdg2'$ OR $stationTravail =' jussieu'$: 1/500 + 1/500 - (1/500 * 1/500) $\approx$ **1/250** = 0,004

Question 22 (1 point)
Quelle est la cardinalité des requêtes suivantes ? Justifier votre réponse.

```
R1: select *
from Valider v, Abonné a
where v.navigo = a.navigo
and v.heure = 8 and a.stationTravail='Jussieu'
```

Avant de calculer card(R1), déterminer card(Valider ⋈ Abonné).

```
R2: select *
from Valider v, LigneStation ls
where v.station = ls.station
```

Avant de calculer card(R2), déterminer le nombre moyen de nuplets de LigneStation pour chaque station.

Solution:
card(R1) ?
card($Valider \bowtie Abonne$)= card(V) = 1 million
SF pour $heure = 8$: 1/20
SF pour $station =' Jussieu'$: 1/500
card(R1) = 1 million * 1/20 * 1/500 = **100**
Card(R2) ?
On peut déduire de l'énoncé qu'une station est associée à 2000 nuplets de LigneStation / 500 stations = 4 lignes de transport par station (en moyenne) .
Donc card(R2) = 1 million * 4 = **4 millions**

Question 23 (1 point)
On crée l'index I1 : create index **I1** on Abonné(catégorie). L'accès par index nécessite une lecture pour chaque nuplet du résultat. Le SGBD choisit entre l'accès par index ou le parcours séquentiel, l'accès qui nécessite le moins de lectures. On observe ceci :
— La requête R3 *select * from Abonné where catégorie = 90* utilise l'index I1.
— La requête R4 *select * from Abonné where catégorie <= 10* n'utilise **pas** l'index I1 mais parcours séquentiellement les nuplets de la relation Abonné.
Calculer le nombre de lectures nécessaires pour traiter une requête avec index, puis en déduire que le nombre de lectures pour effectuer un parcours séquentiel de Abonné est compris entre 2 valeurs. Lesquelles ?

Solution: Il faut entre **100** et **1000** lectures pour effectuer le parcours séquentiel d'Abonnés.
Justification :
Index pour R3 : $categorie = 90$: SF=1%. Nombre de lectures = 1% de 10 000 = 100 lectures. L'index est choisi, car le parcours séquentiel aurait nécessité plus de 100 lectures.
Pas d'index pour R4 : $categorie <= 10$: SF = 10%. L'utilisation de l'index pour R4 aurait nécessité 10% de 10 000 = 1000 lectures L'index n'est pas choisi car le parcours séquentiel fait moins de 1000 lectures.