Conservatoire National des Arts et Métiers
NFP107
Tout document autorisé
Examen de 3 heures

24 juin 2009

1 Algèbre et SQL (8 points)

Soit le schéma de base de données suivant décrivant des courses de bateaux :

- Bateau (Num_Bat, NomB, Sponsor) ;
- Course (Num_Cou, NomC, Date, Prix) ;
- Resultat (Num_Bat, Num_Cou, Score) ;

Num_Bat, Num_Cou sont des nombres correspondant aux identifiants de chaque n-uplets. NomB et NomC sont les noms de chacune des entités. Sponsor est le nom du sponsor du bateau. Prix est la somme perçue par le gagnant de la course. Score est la place du bateau lors de la course.

Donner une expression algébrique et une requête en SQL pour les phrases suivantes :

1. Donner les noms de bateaux sponsorisés par 'Chambourcy' `(1 point)`;

   Solution :
   $\Pi_(NomB)(\sigma_{Sponsor='Chambourcy'}Bateau)$
   SELECT NomB FROM Bateau WHERE Sponsor='Chambourcy' ;

2. Donner le nom des bateaux (distincts) qui sont arrivés premiers (score =1) au moins une fois.
   Ecrire en SQL de 2 manières différentes `(1.5 points)`

   Solution :
   $\Pi_{NomB}(\sigma_{Score=1}Course \bowtie Bateau)$
   SELECT DISTINCT NomB FROM Bateau B, Resultat R
   WHERE B.Num_Bat=R.Num_Bat AND Score=1 ;
   SELECT DISTINCT NomB FROM Bateau B WHERE B.Num_Bat IN
   (SELECT R.Num_Bat FROM Resultat WHERE Score=1) ;
   SELECT DISTINCT NomB FROM Bateau B WHERE EXISTS
   (SELECT R.Num_Bat FROM Resultat R WHERE R.Num_Bat=B.Num_Bat AND Score=1) ;

3. Donner la liste des noms de bateaux ayant participé au 'Solo de Quiberon' `(1 point)` ;

   Solution :
   $\Pi_{NomB}(Bateau \bowtie Resultat \bowtie \sigma_{NomC='SolodeQuiberon'}Course)$
   SELECT NomB FROM Bateau B, Resultat R, Course C
   WHERE B.Num_Bat=R.Num_Bat AND R.Num_Cou=C.Num_Cou
   AND NomC='Solo de Quiberon';

4. Donner le nom des bateaux qui sont toujours arrivés dans les 5 premiers à la course du Solo de Quiberon (2 points) ;

   Solution :
   $\Pi_{NomB}(Bateau \bowtie Resultat \bowtie \sigma_{NomC='SolodeQuiberon'}Course)$
   -
   $\Pi_{NomB}(Bateau \bowtie \sigma_{Score>5}Resultat \bowtie \sigma_{NomC='SolodeQuiberon'}Course)$

   SELECT NomB FROM Bateau B, Resultat R1, Course C1
   WHERE B.Num_Bat=R1.Num_Bat AND R1.Num_Cou=C1.Num_Cou
   AND NomC='Solo de Quiberon'
   AND Num_Bat NOT IN
   (SELECT Num_Bat FROM Resultat R2, Course C2
   WHERE R2.Num_Cou=C2.Num_Cou AND C2.NomC='Solo de Quiberon' AND Score>5) ;

   Donner une requête en SQL pour les phrases suivantes :

5. Donner pour chaque nom de bateau, le nombre de ses participations à une course (1 point) ;

   Solution :
   SELECT NomB, COUNT(*)
   FROM Bateau B, Course C
   WHERE B.Num_Bat=C.Num_Bat
   GROUP BY NomB ;

6. Donner pour chaque sponsor, le nombre de fois que l'un de ses bateaux est arrivé premier. Classer le résultat par ordre décroissant (nombre de courses gagnées) (1.5 points) ;

   Solution :
   SELECT Sponsor, COUNT(*) AS GAGNE
   FROM Bateau B, Resultat R, Course C
   WHERE B.Num_Bat=R.Num_Bat AND R.Num_Cou=C.Num_Cou AND Score=1
   GROUP BY Sponsor
   ORDER BY GAGNE DESC ;

2 Organisation Physique et optimisation (6 points)

La table Bateau contient les n-uplets de clés primaires Num_Bat suivantes : {4, 7, 12, 5, 40, 1, 6, 32, 8, 24, 10, 11, 2 ,3}. On construit un index sur cet attribut, de type arbre B^+ d'ordre 2 (rappel : ce sont les adresses des n-uplets qui sont stockées dans les feuilles de l'arbre).

1. Construire l'arbre B^+ correspondant à l'insertion des valeurs précédentes, en respectant l'ordre donné. Donner les étapes de construction intermédiaires essentielles, et marquer la différence entre les clés et adresses de n-uplets insérées dans l'arbre. (1 point)

   Solution :

2. Combien de pages doivent être lues pour accéder aux attributs des bateaux ayant leur clé dans l'intervalle [7,11] ? Justifier. (1 point)

   Solution :

Soit la requête SQL suivante :

SELECT `NomB`, `Score`
FROM `Bateau`, `Resultat`
WHERE `Bateau.Num_Bat` = `Resultat.Num_Bat` AND `Score` > 10 ;

3. Donner le plan d'exécution sous forme d'arbre de cette requête, dans le cas où l'on ne dispose d'aucun index. `(1 point)`

4. Peut-on exploiter l'arbre B^+ des questions 1 et 2 précédentes ? Si oui, donner le plan d'exécution sous forme d'arbre ET d'explain Oracle, et expliquer le fonctionnement de ce plan. `(2 points)`

   Solution :

5. On suppose maintenant que l'on dispose également d'un index sur l'attribut `Score` de `Resultat`. Peut-on l'exploiter et si oui, donner le plan d'exécution associé, sous forme d'arbre OU d'explain Oracle. `(1 point)`

   Solution :

3 Concurrence (6 points)

Soit l'histoire suivante :
H = $r_1[x]\ r_2[x]\ w_3[y]\ r_2[y]\ w_2[y]\ r_3[z]\ w_1[y]\ c_3\ w_1[z]\ w_2[z]\ c_2\ c_1$

1. Donner la liste des conflits de H ; `(1 point)`

   Solution :

   sur x *aucun conflit ;*

   sur y $w_3[y]r_2[y]\ w_3[y]w_2[y]\ r_2[y]w_1[y]\ w_2[y]w_1[y]$ *;*

   sur z $w_3[z]w_1[z]$ *;*

2. Donner le graphe de sérialisation. H est-elle sérialisable ? `(1 point)`

   Solution :
   $T_3 \rightarrow T_1, T_3 \rightarrow T_1, T_2 \rightarrow T_1$. *Il n'y a pas de cycle, dont H est sérialisable.*

3. H est-elle recouvrable ? Garantie-t-elle l'annulation en cascade ? `(1 point)`

   Solution :
   L'écriture $w_3[y]$ *est validée en* c_3 *avant que les transactions* T_1 *et* T_2 *ne valident. De même pour* $w_2[y]$. *Donc H est recouvrable. Par contre, les lectures sales se font avant le commit* c_1, *donc elle ne garantie pas l'annulation en cascade.*

4. Donner l'histoire finale par application de l'algorithme de verrouillage à deux phases. Donner le détail du déroulement de l'algorithme. On suppose que les verrous sont relâchés immédiatement après les *commit*. `(3 points)`

   Solution :

1	$r_1[x]$	$VL_1[x]$
2	$r_2[x]$	$VL_2[x]$
3	$w_3[y]$	$VE_3[y]$
4		T_2 *bloqué sur* $r_2[y]$
5	$r_3[z]$	$VL_3[z]$
6		T_1 *bloqué sur* $w_1[y]$
7	c_3	$VE_3[y]$ *et* $VL_3[z]$ *relâchés*
8	$r_2[y]$	$VL_2[y]$
9	$w_2[y]$	$VE_2[y]$
10		T_1 *bloqué sur* $w_1[y]$
11	$w_2[z]$	$VE_2[z]$
12	c_2	$VL_2[x]$, $VE_2[y]$ *et* $VE_2[z]$ *relâchés*
13	$w_1[y]$	$VE_1[y]$
14	$w_1[z]$	$VE_3[z]$
15	c_1	$VL_1[x]$, $VE_1[y]$ *et* $VE_3[z]$ *relâchés*

$H' = r_1[x]\ r_2[x]\ w_3[y]\ r_3[z]\ c_3\ r_2[y]\ w_2[y]\ w_2[z]\ c_2\ w_1[y]\ w_1[z]\ c_1$