CY Cergy Paris Université 11/01/2022
Master Informatique M1

Bases de Données Avancées

Examen

Exercice 1 (8 pts)

Un système d'information pour le contrôle du trafic routier utilise une base de données relationnelle pour stocker les mesures des radars automatiques et des informations sur les voitures. Elle a le schéma suivant, où les clés sont soulignées et les clés étrangères sont en italique :

Radar (idRadar, *immatriculation*, date, heure, vitesse)
Voiture (immatriculation, marque, année, propriétaire, adresse)

Une ligne de ***Radar*** contient une mesure de vitesse réalisée par un radar d'identifiant *idRadar* à une date et heure données. La base contient 800 000 mesures de radar et la description de 40 000 voitures. La taille des champs (en octets) est : idRadar(5), immatriculation(10), date(10), heure(10), vitesse(15), marque(40), année(10), propriétaire(40), adresse(100). Une page disque a une taille utile de 4000 octets. Une adresse physique sur disque (ROWID) occupe 10 octets. Une lecture/écriture de page prend en moyenne 0,5 ms pour une lecture séquentielle et 5 ms pour une lecture directe (par ROWID).

a) Ecrivez la requête SQL qui demande le propriétaire, son adresse et l'identifiant du radar pour les voitures de marque Renault ayant été mesurées à plus de 130 km/h.

   Pour la suite on considère que 3% des voitures sont de marque Renault, que 0,2% des mesures sont au-delà de 130 km/h et que le nombre de mesures par voiture est réparti de façon uniforme.

b) Donner le plan d'exécution sans index, utilisant la jointure par boucles imbriquées. Quel est le nombre minimal M de pages mémoire nécessaire pour obtenir le temps minimal d'exécution ? Quel est le temps d'exécution dans ce cas ?

c) On considère qu'il existe des index sur *Radar.immatriculation* et sur *Voiture.marque* (arbres B+ à 3 niveaux). Donner le plan d'exécution qui utilise ces deux index et calculez le temps d'exécution de ce plan.

d) Même demande qu'au point précédent, en considérant cette fois-ci des index sur *Radar.vitesse* et sur *Voiture.immatriculation* (arbres B+ à 3 niveaux).

Exercice 2 (3 pts)

a) Expliquez l'utilité des verrous SIX dans le verrouillage hiérarchique par rapport à l'utilisation séparée d'un verrou S et d'un verrou IX.

b) Soit une table R de n = 10 000 lignes occupant N = 1000 pages disque et un index de type arbre B+ sur l'attribut A de cette table (index dense, non clustérisé). On sait que pour une valeur donnée de A il y a en moyenne k lignes dans la table R ayant cette valeur-là pour A. On sait aussi qu'une entrée d'index (couple valeur de A – adresse physique) occupe p = 10 fois moins de place qu'une ligne de la table R. On suppose que l'index a une hauteur h = 2 (donc 3 niveaux) et qu'il n'y a pas de perte d'espace dans une page disque remplie de lignes de R ou d'entrées d'index.

   Quelle condition doit satisfaire k pour que la recherche des lignes de R ayant une valeur donnée pour A soit plus efficace avec l'index qu'avec une recherche séquentielle dans R ?

Exercice 3 (4 pts)

Soit l'exécution concurrente suivante :

$$\boldsymbol{H} : l_1[x]\ l_2[y]\ l_3[y]\ l_1[y]\ e_2[z]\ e_3[x]\ l_1[z]\ l_2[x]\ c_1\ e_3[y]\ c_3\ c_2$$

a) Donnez *tous* les conflits de l'exécution *H* et vérifiez si elle est sérialisable en construisant son graphe de sérialisation.

b) Enumérez toutes les lectures sales et les écritures sales de *H*. Dans quelle catégorie par rapport aux annulations se trouve *H* : stricte, sans annulations, recouvrable ou non-recouvrable ?

c) Quelle est l'exécution obtenue à partir de *H* par verrouillage à deux phases simple ? On suppose les verrous d'une transaction relâchés à la fin de celle-ci et à ce moment on donne priorité aux opérations en attente de verrou (dans l'ordre de leur blocage).

Exercice 4 (2 points)

a) Donner une représentation d'un type d'objet *multipoint*. Un *multipoint* est défini comme un ensemble (liste) de points. Ainsi, pour définir le type d'objet *multipoint*, vous devez définir tout d'abord le type d'objet *point*.

Un objet *multipoint* est composé des éléments suivants :

- Une table ou un tableau de points
- Un identifiant de *multipoint* (integer), un nom de *multipoint* (varchar) et le système de coordonnées (varchar) comme attributs

Inclure une méthode qui compare deux *multipoints* P1 et P2 et retourne -1, si P1 a moins des points que P2 ; 0, s'ils ont le même nombre de points ; et +1, si P1 a plus de points que P2.

b) Oracle supporte-t-il l'héritage dans les types d'objets déclarés sous le modèle OO-SQL ? Si oui, comment devons-nous déclarer les types afin de pouvoir créer des sous-types ?

Exercice 5 (3 points)

a) Quelle est le processus de changement de système de référence (SRID) pour les données spatiales enregistrées dans une BD Oracle Spatial ?

b) Quelles sont les différences entre l'indexation **R-Tree** et l'indexation **QuadTree** ?

c) Les deux tables suivantes ont déjà été créées et remplies dans Oracle Spatial :

```
CREATE TABLE GEOD_CITIES(
  LOCATION     SDO_GEOMETRY,        /* geometrie */
  CITY         VARCHAR2(42),        /* nom ville */
  COUNTRY      VARCHAR2(50),        /* pays */
  POP90        NUMBER);             /* population */

CREATE TABLE GEOD_RIVERS(
  RIVER   VARCHAR2(35),             /* nom de fleuve */
  GEOM    SDO_GEOMETRY);            /* geometrie */
```

Ecrire la requête pour trouver toutes les villes qui sont à moins de 20 km d'un fleuve (par exemple « Seine») et la distance exacte par rapport à ce fleuve.

Indication : utilisez

sdo_within_distance(<geometry-1>,<geometry-2>, 'DISTANCE=<n>,[optional parameters]'))

SDO_GEOM.SDO_DISTANCE (<geometry-1>, <geometry-2>,<tolerance> [, <unit>])