

Examen - **Bases** de données réparties *3^{ème} année*

Documents autorisés
Durée : 1H30

Questions de cours

- 1) Décrire, brièvement, le problème de << verrou mortel global >>. Donner un exemple qui illustre ce problème.
- 2) Citer, de façon synthétique, les avantages et les inconvénients des bases de données répliquées.

Exercice 1

La base de données d'une agence de tourisme spécialisée dans les sports d'hiver contient les schémas de relation suivants, dont la sémantique est claire (les clés primaires sont marquées par # et les clés étrangères sont soulignées) :

STATION(#NumStation, NomStation, Altitude, Région)
HOTEL(#NumHotel, **NomHotel**, NumStation, CatégorieNum)
CATEGORIE(#NumCatégorie, NombreEtoiles)

On suppose que la base de données ci-dessus est hébergée par un serveur Oracle dans la Région 1.ss On désire répartir cette base de données entre la région 1 (seveurl) et la région2 (serveur2) en mettant dans chaque région les données relatives à cette région. Nous considérons que **Bd_Région1** et **Bd_Région2** représentent, respectivement, la chaîne de connexion pour la base de données de la région 1 et la chaîne de connexion pour la base de données de la région2. **Compte_Région1/Mot_passe_région1** et **Compte_Région2/Mot_passe_région2** représentent, respectivement, le compte miroir/mot de passe de la région1 et ceux de la région2.

La table Catégorie doit rester sur le serveur de la région 1.

Afin de garantir l'unicité des clés primaires, elles sont générées automatiquement par les séquences *seqJStation* et *seq_Hotel*. Ces séquences sont hébergées par le serveur de la région 1.

- 1) Donner les requêtes permettant de créer les différents fragments.
- 2) Sur la base région1, créer un database link dbl_région2 permettant d'accéder aux objets distants de région2. Faire la même chose sur région2.
- 3) Donner la requête permettant de créer une vue « HotelGlobal » qui reconstitue la table fragmentée Hôtel.

4) Ecrire un trigger qui permette de faire des insertions de lignes dans la vue HotelGlobal (montrer comment peut-on utiliser la séquence SeqHotel).

5) Ecrire un trigger qui permette de faire des mises à jour à travers la vue HotelGlobal

Exercice 2

1) On considère la relation ETUDIANT (CNE, Nom, Prénom, NoteExamen, LieuStage, DateDébutStage, DateFinStage) fragmentée verticalement sur deux sites, selon les critères suivants :

Site 1 : Les informations concernant les examens : CNE, Nom, Prénom, NoteExamen

Site 2 : Les informations concernant les stages : CNE, Nom, Prénom, LieuStage, DateDébutStage, DateFinStage

Ecrire les requêtes permettant de créer ces fragments.

2) Donner la requête permettant de créer une vue « EtudiantGlobal » qui reconstitue la table fragmentée Etudiant.

3) Ecrire un trigger qui permette de faire des insertions d'étudiants à travers la vue EtudiantGlobal.

4) Ecrire un trigger qui permette de faire des mises à jour (delete et update) à travers la vue EtudiantGlobal.

Bonne chance