

ELEC 378 : Systèmes Distribués d'Information
Professeur : Esteban Zimányi
Examen de Seconde Session

Question 1 : Bases de Données Temporelles (5 points)

Un fournisseur d'accès à Internet dispose de la base de données temporelle suivante pour gérer ses abonnés :

- **Abonné**(login,nom, prénom, numTél)
- **Abonnement**(login,typeAbon,EspaceMax) as valid state day
- **EspaceUtilisé**(login,espaceUtil) as valid state day
- **Connexion** (login, KbIn, KbOut) as valid state minute

Ecrivez en TSQL2 les requêtes suivantes :

1. Donnez le nom et prénom des abonnés qui se sont connectés le mois de juin 2002 et qui utilisent actuellement plus d'espace disque que celui autorisé par le type de leur abonnement.
2. Donnez le login des abonnés qui ne se sont pas connectés depuis le 1^{er} janvier 2002.
3. Donnez le login et la durée de la plus longue connexion depuis le 1^{er} janvier 2002.
4. Pour chaque abonné, donner le login et la somme des durées de ses connexions pendant le mois de décembre 2001 si cette somme est supérieur à 10h.
5. Donnez pour chaque utilisateur le login et le maximum d'espace utilisé, ainsi que la date a laquelle ce maximum a été atteint.

Question 2 : Bases de Données Déductives (5 points)

Dans une agence de voyages qui propose des circuits à ses clients, considérez la base de données relationnelle suivante:

- **Circuit**(CodeCircuit,Nom,Description,NbMaxPartic)
Cette table décrit les différents circuits offerts par l'agence.
- **Accompagnateur**(CodeAcc,Nom,Prénom,Adresse,Téléphone)
Cette table décrit les différents accompagnateurs travaillant pour l'agence.
- **CircuitDate**(CodeCircuit,DateDébut,DateFin,CodeAcc)
CodeCircuit references Circuit.CodeCircuit
CodeAcc references Accompagnateur.CodeAcc
Cette table décrit un circuit effectué ou qui va l'être avec l'accompagnateur assigné.
- **ClientCircuit**(CodeClient,CodeCircuit,DateDébut)
[CodeCircuit,DateDébut] references Circuit.[CodeCircuit,DateDébut]
Cette table décrit les clients qui sont inscrits dans un circuit.

Ecrivez en Datalog les prédicats suivants.

1. **CircuitParticip(C,D,N)** : le circuit C partant à la date D a N participants inscrits.
2. **CircuitPopulaire(C)** : le circuit C est toujours complet toutes les fois où il a été réalisé (nombre de participants inscrits = nombre maximum de participants).
3. **ConflitAcc(A)** : l'accompagnateur A doit participer dans deux circuits programmés à des dates qui se chevauchent
4. **ListeOrdCircuits(A,L)** : L est la liste de circuits dans lesquels a participé l'accompagnateur A ordonné par date de début.
5. **NombreJours(A,N)** : N est le nombre de jours de circuit dans lequel l'accompagnateur A a participé.

Question 3 : Bases de Données Actives (5 points)

Considérez la base de données relationnelle suivante:

- **Joueur**(#joueur,nom,prénom,adresse,#équipe,nbMatchesJoués)
#équipe references Equipe.#équipe
- **Equipe**(#équipe,nomEquipe,#capitaine)
#capitaine references Joueur.#joueur
- **Entraînement**(#équipe,stade)
#équipe references Equipe.#équipe
- **Match**(#match,stade,#équipeLocale,#équipeVisiteur,goals1,goals2)
#équipeLocale references Equipe.#équipe
#équipeVisiteur references Equipe.#équipe
- **Joue**(#joueur,#match)
#joueur references Joueur.#joueur
#match references Match.#match

La base de données doit vérifier les contraintes suivantes :

1. Le capitaine d'une équipe doit être joueur de l'équipe
2. Un joueur ne joue que dans des matchs de son équipe
3. L'équipe locale d'un match doit s'entraîner dans le stade du match
4. Une équipe a au plus 22 joueurs inscrits
5. L'attribut nbMatchesJoués est un attribut dérivé à partir de la relation Joue

Pour chacune de ces 5 contraintes établissez en Starburst une règle active qui se déclenche lors d'une violation et réalise les actions de réparation nécessaires. Dans les 5 règles au moins une doit se déclencher avec chacun des événements INSERT, DELETE et UPDATE. Commentez chacune des règles.

Question 4 : Bases de Données Objet (5 points)

Soit la base de données de gestion de zones d'eau suivante :

```
class ZoneD'Eau (extent ZonesD'Eau ) {
    attribute String nom; attribute set< String > autresNoms;
    relationship Set<Port> ports inverse Port::zoneD'Eau;
    relationship Set<Ville> villes inverse Ville::zonesD'Eau;
}
Struct MesuresDébit { Integer débit; Date dateMesure; }
class Rivière extends ZoneD'Eau (extent Rivières) {
    attribute set<MesuresDébit> débitD'Eau
}
Struct Coordonnées { Float X; Float Y; }
Struct MesuresProfondeur { Integer profondeur; Coordonnées coordonnées; }
class Lac extends ZoneD'Eau (extent Lacs) {
    attribute set<MesuresProfondeur> profondeur
}
Struct Population { Integer population; Date annéePopulation; }
class Ville (extent Villes) {
    attribute String nom ; attribute set<Population> population ;
    relationship set<ZoneD'Eau> zonesD'Eau inverse ZoneD'Eau::villes;
    relationship set<Port> ports inverse Port::ville;
}
Struct Bateau { String nom; Integer place; String propriétaire; }
class Port (extent Ports ) {
    attribute String nom ; attribute list<Bateau> bateaux;
    relationship Ville ville inverse Ville::ports;
    relationship ZoneD'Eau zoneD'Eau inverse ZoneD'Eau::ports;
}
```

Ecrivez en OQL les requêtes suivantes.

1. Donnez le nom des rivières qui coulent dans une ville telle qu'il existe aussi un lac dans cette ville
2. Donnez le nom des rivières qui ont plus de 1'000 bateaux dans tous ses ports.
3. Donnez le nom des rivières telles que la somme totale de la population courante (de l'année 2002) de toutes les villes dans lesquelles la rivière coule est supérieure à 100'000.
4. Donnez le nom des rivières telles que la moyenne des bateaux dans un port appartenant à l'Administration Communale est supérieure à 5.
5. On considère que l'ensemble de bateaux d'une ville est constitué par l'ensemble de bateaux appartenant aux ports de la ville. Donnez pour chaque ville (a) le nom de la ville et (b) un ensemble de couples (propriétaire, nombre de bateaux) pour l'ensemble de bateaux d'une ville.

N.B : Documents autorisés

Bon travail