

Question 1 : Bases de Données Actives (5 points)

Considérez la base de données relationnelle suivante qui concerne la gestion de parcelles et des lignes de métro:

- **Commune**(nomCommune, géométrie, population)
- **Parcelle**(#parcelle, géométrie, propriétaire, utilisation)
utilisation \in {industrielle, commerciale, logement, publique }
- **Appartient**(#parcelle, nomCommune)
#parcelle references Parcelle.#parcelle
nomCommune references Commune.nomCommune
- **StationMétro**(#station, géométrie, nomStation)
- **LigneMétro**(#ligne, géométrie)
- **TronçonMétro**(#tronçon, géométrie, #ligne, #stationDep, #stationArr)
#ligne references LigneMétro.#ligne
#stationDep references StationMétro.#station
#stationArr references StationMétro.#station

Les attributs de type géométrie sont définis sur un Type de Données Abstrait qui définit un ensemble de méthodes (area, perimeter, length, ...) et un ensemble d'opérateurs (contains, overlaps, disjoint, equals, crosses, ...) Par exemple A.géométrie.area permet de déterminer la surface totale d'une géométrie et A.géométrie.overlaps B.géométrie permet de vérifier que les deux géométries ont une intersection non vide.

La base de données doit vérifier les contraintes suivantes :

1. Les géométries des parcelles sont disjointes.
2. Chaque commune doit avoir au moins une station de métro.
3. La surface totale des parcelles à vocation industrielle d'une commune ne peut excéder le 15% de la surface de la commune.
4. Une ligne de métro ne peut pas passer plus d'une fois sur une même station de métro.
5. La longueur d'une ligne de métro est égale à la somme des longueurs de ses tronçons.

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 2 : Bases de Données Temporelles (5 points)

Dans une application de gestion de routes considérez la base de données temporelle suivante :

- **Route**(#route, nomRoute, longueur)
- **Compagnie** (#client, nom, adresse)
- **TronçonRoute**(#tronçon, #route, commune)
#route references Route.#route
- **ContratEntretien** (#tronçon, #client, montant) as valid state day
#tronçon references Tronçon.#tronçon
#client references Compagnie.#client
- **Travaux** (#tronçon, #client, coût) as valid state day
#tronçon references Tronçon.#tronçon
#client references Compagnie.#client

Ecrivez en TSQL2 les requêtes suivantes :

1. Donnez les identificateurs des routes qui ont eu des travaux pendant l'année 2003.
2. Donnez les identificateurs des tronçons de route dont le coût total des travaux de réfection pendant la période du contrat dépasse le montant total du contrat d'entretien.
3. Donnez les identificateurs des compagnies qui n'ont jamais effectué des travaux de réfection de routes dans la Commune d'Ixelles.
4. Donnez pour chaque commune le coût total de travaux de réfection pendant l'année 2003.
5. Donnez pour chaque route l'historique des coûts de travaux de réfection.

Question 3 : Bases de Données Déductives (5 points)

Considérez la base de données relationnelle suivante:

- **Projet**(CodeProjet, NomProjet, Description, Budget, Département)
- **Tâche**(CodeTâche, NomTâche, NbJours, Description)
- **Employé**(#Employé, Nom, Prénom, Adresse)
- **ProjetTâche**(CodeProjet, NuméroTâche, CodeTâche, Date, Responsable)
CodeProjet references Projet.CodeProjet
CodeTâche references Tâche.CodeTâche
Responsable references Employé.#Employé
NuméroTâche $\in \{1, 2, \dots\}$ donne le numéro de séquence de la tâche dans le projet.
- **Ressources**(#Ressource, CodeRessource, CoûtRessource)
- **TâcheRessource**(CodeTâche, #Ressource)
CodeTâche references Tâche.CodeTâche
#Ressource references Ressource.#Ressource

Ecrivez en Datalog les prédicats suivants.

1. **Stakhanoviste(E,D)** : l'employé E travaille comme responsable dans tous les projets du département D.
2. **ConflitDates(P,T1,T2)** : Les deux tâches de code T1 et T2 assignées au projet P se chevauchent dans les dates.
3. **ListeTâches(P,L)** : L est la liste de tâches du projet P ordonnée par numéro de tâche.

4. **DépassementBudget(P, B)** : La somme totale des coûts des ressources utilisées dans les tâches du projet P dépasse de B unités le budget du projet.
5. **Récapitulatif(E,P,NbJours)** : NbJours est le nombre de jours que l'employé E est responsable d'une tâche du projet P.

Question 4 : Bases de Données Objet (5 points)

Soit la base de données d'un club hippique suivante :

```

Struct Adresse { string rue; string numéro; string CP; string ville; }
class Cheval (extent Chevaux) {
    attribute String nomCheval;
    attribute String sexe; attribute Date dateNaissance;
    relationship Personne propriétaire inverse Personne:: propriétaireChevaux;
    relationship Personne entraîneur inverse Personne:: entraîneChevaux;
    relationship list<Participation> coursesCourues inverse Participation::cheval;
}
class Personne (extent Personnes) {
    attribute String nomPersonne;
    attribute Date dateNaissance;
    attribute Adresse adresse;
    relationship list<Cheval> propriétaireChevaux inverse Cheval::propriétaire;
    relationship list <Cheval> entraîneChevaux inverse Cheval::entraîneur;
    relationship list<Participation> jockeyChevaux inverse Participation::jockey;
}
Struct Prix { string place; integer montant; }
enum TypeCourse { plat, obstacles }
class Course (extent Courses) {
    attribute String nomCourse; attribute String hippodrome;
    attribute String date; attribute Integer numéroCourse;
    attribute Integer distance; attribute TypeCourse type;
    attribute list<Prix> prix;
    relationship list<Participation> chevauxPartant inverse Participation::course;
}
class Participation (extent Participations) {
    attribute Integer noCheval;
    attribute Integer placeArrivée;
    relationship Cheval cheval inverse Cheval::coursesCourues;
    relationship Personne jockey inverse Personne::jockeyChevaux;
    relationship Course course inverse Course::chevauxPartant;
}

```

Ecrivez en OQL les requêtes suivantes.

- a) Donner pour chaque cheval le nom de toutes les courses dans lesquelles il/elle a couru pendant l'année 2003 ainsi que sa place d'arrivée et le gain obtenu. Exemple de sortie:

Cheval: Lauralinka Année: 2003

 8/05/2003 Prix de la Pépinière
 10^{ème} place, Gain: 0 €

 27/03/2003 Prix du Parc Monceau
 1^{ère} place, Gain: 22.560 €

 7/03/2003 Prix Saint-Pair du Mont
 2^{ème} place, Gain: 5200 €

- b) Donner un résumé de la carrière de chaque cheval en précisant par spécialité (plat ou obstacle) (1) le nombre de courses courues, (2) le montant total des gains, (3) le nombre de fois où le cheval arrive en première position, (4) le montant total des gains en première position, (5) le nombre de fois où le cheval arrive dans une des positions rapportant un gain (autre que la première) et (6) le montant total des gains dans ces positions. Exemple de sortie:

Cheval: Lauralinka Année: 2003

 SPECIALITE COURSES GAINS

 PLAT courue(s) : 13 45.030 E
 victoire(s) : 2 23.000 E
 placé : 4 22.030 E

 OBSTACLE courue(s) : 8 25.500 E
 victoire(s) : 1 13.000 E
 placé : 3 12.500 E

- c) Donner par hippodrome et par date, l'ensemble des courses ainsi que les numéros des 7 premiers chevaux arrivés. Exemple de sortie

Hippodrome d'Agen Lundi 26 mai 2003

 1ère course: Prix Jacques Serra (arr: 8 2 7 5 4 1 10)
 2ème course: Prix René Brocard (arr: 1 2 6 12 10 5 4)
 3ème course: Prix de Pau (arr: 10 6 2 3 11 8 12)

- d) Donner le nom des propriétaires qui ont participé comme jockey dans au moins une course.
 e) Donner par hippodrome le montant total des prix attribués pendant 2003.

N.B : Documents autorisés.

Bon travail.