Ing1 - Examen de Base de Données – 2ème session - EISTI 2009-2010

vendredi 26 janvier 2010

Aucun document autorisé. Ordinateur et calculatrice interdite.
Fiche algèbre relationnelle en annexe.

1ère Partie – Cours – 8 points

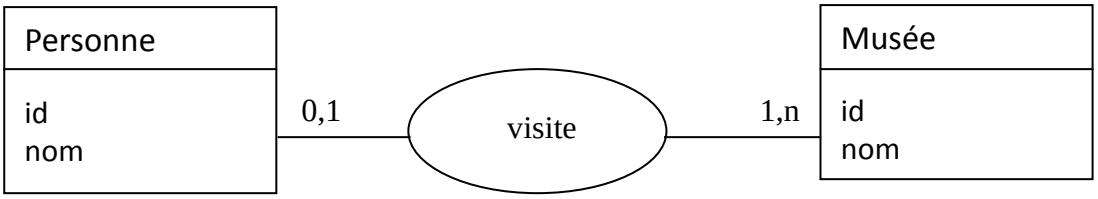
On s'intéresse à des personnes jouant ou réalisant des films. Une personne est connue par son identifiant (**idP**), son **nom** et son **prénom**. Un film a un identifiant (**idF**) et un **titre** ; il est réalisé par une seule personne, à une **année** particulière. Les acteurs d'un film sont des personnes qui y jouent un **rôle** (et un seul). On propose le modèle logique suivant :

Personne(idP, nom, prénom)
Réalise(*idP*, idF, année)
Joue(*idP, idF*, rôle, titre)

Q1. Pour chaque relation du modèle précédent, donner la liste complète des dépendances fonctionnelles élémentaires.

Q2. En quelle forme normale est ce modèle ? Si ce modèle n'est pas en BCNF, proposer votre modèle en BCNF.

Q3. Que signifient les cardinalités du MCD suivant ?



Q4. Soit les relations suivantes pouvant représenter les classements de coureurs participant à des courses :

Coureur(idCoureur, nom, prénom)
Course(idCourse, libellé, date)
Courir(*idCoureur, idCourse*, place)

En utilisant uniquement les opérateurs de projection et de division, donner la requête en algèbre relationnelle qui donne les identifiants des coureurs ayant participé à toutes les courses.

Q5. Quel est l'ordre SQL qui permet de modifier la structure d'une table ?

Q6. Pour chacune des requêtes suivantes, préciser si un index peut accélérer son exécution et si oui sur quel(s) attribut(s) ?

a) SELECT nom, prenom FROM Personne WHERE nom like 'M%';
b) SELECT * FROM Rectangle WHERE longueur * largeur > 50 ;
c) SELECT nom, age FROM Personne WHERE age > 50;
d) SELECT nom FROM Personne WHERE adresse is NULL;

Q7. Soit la table initialement vide suivante :

CREATE TABLE T (a : INTEGER ,b : INTEGER) ;

On a la séquence suivante d'ordres qui proviennent de 2 sessions S1 et S2 :

S2 : INSERT INTO T VALUES (3,4);
S2 : INSERT INTO T VALUES (5,6);
S2 : COMMIT;
S1 : INSERT INTO T VALUES (7,8);
S2 : DELETE FROM T WHERE a = 3;
S1 : DELETE FROM T WHERE a = 5;
S1 : SAVEPOINT p1;

S2 : INSERT INTO T VALUES (9,10);
S1 : UPDATE T SET b = 9 WHERE b = 7;
S1 : INSERT INTO T VALUES (11,12);
S2 : ROLLBACK;
S1 : ROLLBACK TO p1;
S1 : INSERT INTO T VALUES (13,14);
S2 : INSERT INTO T VALUES (15,16);
S2 : COMMIT;
S1 : ROLLBACK;
S2 : ROLLBACK;
S1 : COMMIT;
S2 : COMMIT;

Quel est le contenu de la table T à la fin de cette séquence ?

Q8. Qu'a-t-on voulu faire en mettant en place la structure suivante ?

CREATE TABLE Etudiant (id, nom, prenom, adresse, telephone, classe, groupe);
CREATE VIEW Etudiant_profs AS SELECT nom, prenom, classe, groupe FROM Etudiant ORDER BY classe, groupe, nom;
GRANT SELECT ON Etudiant_profs TO professeur;
GRANT ALL ON Etudiant TO secretaire;

Exercice – 12 points

Voici un MLD simplifié permettant la gestion des élèves des auto-écoles :

- AutoEcole(id, nom, adresse, tel, tarifHeureConduite)
- Eleve(id, nom, prenom, dateNaissance, adresse, tel, dateInscription, *idAutoEcole*)
- Moniteur(id, nom, prenom, adresse, tel, *idAutoEcole*)
- Seance(*idEleve, idMoniteur,* date, heureDebut, duree, observation)

On s'intéresse aux requêtes suivantes :

R1. Nom des élèves qui ont eu cours avec Dominique Potier le 31 décembre 2009.
Question 1 : Formuler cette requête en **arbre algébrique**. (**1 pt)**
Question 2 : Formuler cette requête en **SQL**. (**1 pt)**

R2. Le montant payé par l'élève Angélique Pascal pour ses heures de conduite.
Question 3 : Formuler cette requête en **algèbre relationnelle**. (**1 pt)**
Question 4 : Formuler cette requête en **SQL**. (**1 pt)**

R3. Nom, adresse et nombre de moniteurs des auto-écoles qui ont au moins 5 moniteurs.
Question 5 : Formuler cette requête en **SQL**. (**1 pt)**

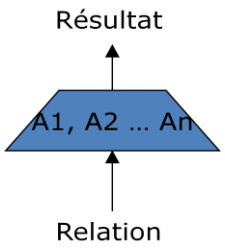
R4. Nom, prénom et nombre d'heures de conduite (éventuellement nul) de chaque élève, classé par nombre d'heures décroissant.
Question 6 : Formuler cette requête en **algèbre relationnelle**. (**1 pt)**
Question 7 : Formuler cette requête en **SQL**. (**1,5 pt)**

R5. Nom et prénom des élèves qui ont fait cours avec tous les moniteurs de l'auto-école Rally Auto.
Question 8 : Formuler cette requête en **algèbre relationnelle**. (**1 pt)**
Question 9 : Formuler cette requête en **SQL**. (**1,5 pt)**

R6. Nom et prénom des élèves nés après le 1 janvier 1990, habitant à Paris, ayant fait un cours de 2 heures, le mois janvier 2010 et avec un moniteur de l'auto-école Rally Auto dont le prénom contient 2 lettres 'e'.
Question 10 : Formuler cette requête en **arbre algébrique optimisé** suivant l'heuristique vu en cours**.** (**2 pt)**

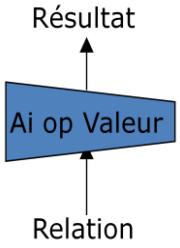
Annexe : algèbre relationnelle et arbre algébrique

Projection (*)



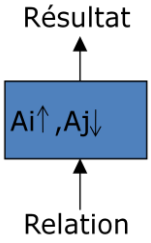
$\Pi_{A1, A2 \ldots An}(Relation)$

Restriction



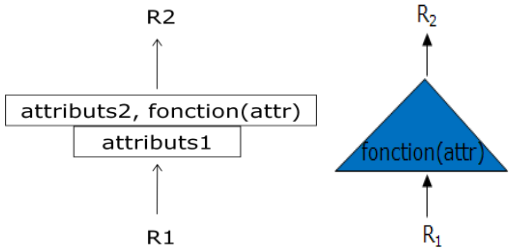
$\sigma_{Ai\ op\ Valeur}(Relation)$

Tri



Tri(Relation, Ai↑, Aj↓)

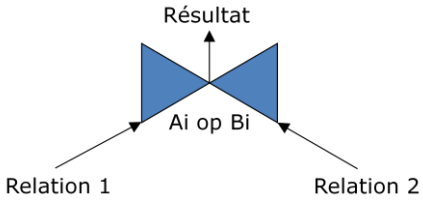
Agrégat et Calcul



$_{attributs1}F_{attributs2,fonction(attr)}(R1)$

$F_{fonction(attr)}(R1)$

Jointures



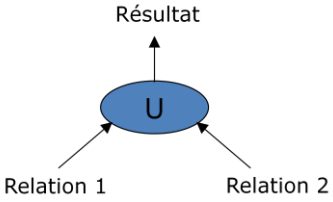
Relation1 $\bowtie_{Ai\ op\ Bi}$ Relation2

$\bowtie$: jointure à droite

$\bowtie$: jointure à gauche

$\bowtie$: jointure complète

Opérations ensemblistes

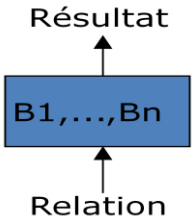


Relation1 U Relation2

Autres opérateurs :

$\cap$, -, ÷, X

Renommage (**)



$\rho_{E(B1,\ldots, Bn)}(Relation)$

$\rho_{(B1,\ldots, Bn)}(Relation)$

$\rho_{E}(Relation)$

(*) **NB1** : on utilise également l'opérateur de projection pour les champs calculés

(**) **NB2** : pas de représentation sur un arbre du renommage du nom d'une relation)