

Université Paris 7
Master 1 Informatique, Bases de données avancées.
5 janvier 2012

Durée : 3 heures. Documents manuscrits, notes de cours, notes de TD/TP autorisés. Livres, ordinateurs, téléphones portables interdits.

Le sujet comporte 4 pages.

Dépendances

Exercice 1 On considère une relation $R(A, B, C, D, E)$ qui satisfait les dépendances fonctionnelles suivantes :

$$\begin{aligned} A, B &\rightarrow C \\ C, D &\rightarrow E \end{aligned}$$

Question 1: Donner toutes les clés candidates de la relation R . Justifier. (Justifier ne veut pas dire qu'il faut calculer systématiquement toutes les clôtures possibles.)

Question 2: Est-ce que cette relation est BCNF? Justifier.
Si R n'est pas BCNF alors décomposez la en relations BCNF.
Est-ce que votre décomposition préserve les dépendances fonctionnelles?

Exercice 2 Nous avons une relation $R(A, B, C, D, E)$ avec les dépendances :

$$\begin{aligned} A, B &\rightarrow C \\ C, D &\rightarrow E \\ E &\rightarrow A \end{aligned}$$

Question 1: Trouver toutes les clés candidates.

Question 2: Est-ce que cette relation est 3NF? Si elle n'est pas 3NF alors expliquez pourquoi.

Modélisation

Exercice 3 Votre tâche consiste à modéliser¹ un système d'information pour un marchand d'art.

Il faut gérer les oeuvres d'art, les artistes, les clients, les achats, les ventes et la comptabilité.

1. partiellement, sinon cela devient trop complexe

Les oeuvres exposés et mis en vente soit appartiennent au marchand (il les a acheté pour revendre) soit ils sont donnés en commission (ils appartiennent à quelqu'un d'autre, peut-être l'artiste lui même) et le marchand sert d'intermédiaire moyennant un pourcentage sur le prix de vente.

La bases de données à concevoir contient aussi bien des informations sur des oeuvres exposés sur ceux déjà vendus. Pour tous les oeuvres il faut savoir

- quel artiste en est auteur,
- le type d'oeuvre (sculpture, dessin, etc.),
- le prix de vente et l'acheteur pour les oeuvres vendus,
- le vendeur et le prix d'achat et la date d'achat pour les oeuvres achetés par le marchand,
- la date de vente pour les oeuvres vendus,
- la date de mise en vente pour les oeuvres en commission.

Les cardinalités de relations doivent être bien visibles dans votre modèle.

Le modèle doit être clair, bien dessiné et compréhensible. Un modèle qui se limite à trois « boites » (marchand, oeuvre, client) sans attributs, de relations sans cardinalité clairement marqué sur le dessin et dont on ne comprend pas le rôle ne donnera aucun point.

Triggers - déclencheurs

Exercice 4

Un système de gestion d'inscriptions universitaires et de notes contient des tables suivantes tables² :

```
create table cours(  
id_cours int check(id_cours >=0),  
intitule varchar(10) not null ,  
credits int not null check(credits >0),  
annee int check(annee >0),  
niveau int not null check(niveau in (1,2,3,4,5)),  
primary key(id_cours ,annee)  
);
```

```
create table etudiants(  
num char(10) primary key, —numero d'etudiant  
nom varchar(40) not null ,  
prenom varchar(40)  
);
```

```
create table inscription_annee(  
num char(10) references etudiants ,
```

2. Le système contient aussi d'autres tables qui ne sont pas pertinent pour cet exercice.

```

annee int not null check(annee > 0),
niveau int not null check(niveau in (1,2,3,4,5)),
primary key(num, annee)
);

```

```

create table inscrits_cours(
num char(10) references etudiants,
id_cours int,
annee int,
note decimal(3,1) check(note >= 0 and note <=20) default null,
primary key(num, id_cours, annee),
foreign key(id_cours, annee) references cours
);

```

cours La table **cours** contient les cours

- le code d'un cours **id_cours** identifie le cours,
- **credits** donne le nombre de crédits pour ce cours,
- **annee** donne l'année académique où le cours est dispensé, par exemple **annee=2010** signifie que le cours a été dispensé dans l'année 2010-2011 (l'attribut donne juste l'année de départ pour l'année académique),
- **niveau 1,2,3,4,5** correspond à L1,L2,L3,M1,M2 respectivement, donc par exemple un cours de niveau 4 c'est un cours de M1.

etudiants La table **etudiants** nous donne la liste d'étudiants.

inscriptions_annee permet de savoir à quel niveau est inscrit un étudiant pour une année académique donnée. Donc les attributs **annee** et **niveau** ont la même signification que les attributs de même nom de la table **cours** mais dans la table **inscriptions_annee** ces attributs concernent les étudiants et non pas les cours. Par exemple un enregistrement ('1991078423', 2011, 3) signifie que l'étudiant avec **num='1991078423'** est inscrit en L3 pour l'année académique 2011-2012.

inscrits_cours sert à enregistrer les inscriptions dans le cours et les notes reçues. L'attribut **note** donne la note.

Le but de cet exercice est d'écrire deux triggers. Pour les deux triggers il faut aussi bien écrire les fonctions qui réalisent les triggers que les commandes CREATE TRIGGER.

Dans ce qui suit on supposera qu'un module est définitivement acquis par un étudiant, ce qui donne tous les crédits associés à ce module, si sa dernière note pour ce module est supérieure ou égale à 10.

Question 1: Le premier trigger sera associé à la table `inscription_annee`.

Quand on exécute la commande `INSERT` qui inscrit un étudiant le trigger doit vérifier si les conditions d'inscription sont réunies.

Pour que l'inscription soit valide il faut satisfaire les contraintes suivantes :

- (1) L'inscription au niveau k ($k \geq 2$) l'année i est autorisée si l'étudiant a été inscrit pour l'année $i - 1$ au niveau $k - 1$ et a obtenu au moins 60 crédits avec les cours de niveau $k - 1$, ces crédits pouvaient être acquis non seulement l'année $i - 1$ mais aussi des années précédentes (les étudiants peuvent faire de modules en avance et ils peuvent redoubler donc les crédits de niveau $k - 1$ pouvaient être acquis n'importe quand durant la scolarité). Ce cas correspondant à l'inscription où l'étudiant a validé le niveau précédent.
- (2) L'inscription au niveau k l'année i est valide si l'année précédente l'étudiant a été inscrit au même niveau k mais il n'est pas arrivé à obtenir le 60 crédits de niveau k . Ceci correspond à une inscription avec le redoublement de l'année.
- (3) on autorise aussi la première inscription à l'université et cela à n'importe quel niveau, on suppose que dans ce cas les services compétentes ont vérifié que l'étudiant possède les diplômes autorisant l'inscription. Donc si on essaie inscrire quelqu'un au niveau k et auparavant cette personne n'était jamais inscrite à l'université alors l'inscription sera autorisée.

Les inscription non valides devaient être refusées.

Question 2: Le deuxième trigger sera associé à la table `inscrits_cours` et sera déclenché au moment d'inscription à un cours (`INSERT`).

L'inscription au module de niveau k est refusé pour l'année i si l'étudiant n'est pas inscrit à l'université pour la même année i ou s'il est inscrit à l'université mais au niveau supérieur à k . Donc pour s'inscrire à un cours il faut être inscrit à l'université (et payer). De plus si l'étudiant est inscrit au niveau k alors il peut s'inscrire à chaque cours de niveau $\geq k$ mais il ne peut pas s'inscrire à un cours de niveau inférieur à k (on peut faire de modules en avance mais une fois l'année acquise impossible de revenir en arrière).

De plus si l'étudiant a déjà été admis à un cours donnée, c'est-à-dire il a déjà reçu une note ≥ 10 pour le cours avec le même `id_cours`, alors la nouvelle inscription pour le même cours sera refusée. Donc on ne peut pas refaire un cours sauf en cas d'échec.

Et autres

Exercice 5 Dessiner un B-arbre qui contient les clés : 2, 4, 6, 8, 12, 16, 18. On suppose que chaque sommet de ce B-arbre possède la place pour trois clés et quatre pointeurs. Il y a plusieurs B-arbres qui satisfont ces conditions, il suffit de dessiner un d'eux.