

TD BDA (Séance n° 2) Triggers

Exercice 1

On considère la même base de données IUT mais avec quelques modifications dont la plus importante concerne la gestion des notes des étudiants. En effet, avec cette nouvelle version, toutes les notes obtenues par les étudiants aux différents contrôles seront conservées. Une nouvelle relation `CONTROLE` est introduite. Son schéma est donné ci-dessous.

`CONTROLE(IdC, Code, TypeC, DateDeb, DateFin, NoteMax)`

où :

- `Code` est une clef étrangère correspondant à la matière contrôlée (Cf. clef primaire de `MODULE`) ;
- `TypeC` est le type du contrôle. Il s'agit d'un énuméré dont les valeurs appartiennent à l'ensemble : {Test, Test rattrapage, Interrogation, Devoir, Projet, Etude de cas...} ;
- Les attributs `DateDeb` et `DateFin` indiquent la date de début et de fin d'un contrôle. Si le contrôle est un test, un test de rattrapage ou une interrogation, ces deux dates devront être identiques ;
- Enfin l'attribut `NoteMax` donne la note maximale attribuée qui peut être différente de 20. Par exemple, un test pourra être noté sur 60.

La relation `NOTATION` a alors une nouvelle structure indiquée ci-dessous :

`NOTATION(IdC, Num ET, Note, NoteDef)`

où :

- `Note` indique la note obtenue au contrôle `IdC` par l'étudiant `NUM_ET` selon le barème de correction (Cf. `NoteMax`) ;
- `NoteDef` est la note définitive pour ce contrôle et cet étudiant, i.e. la note sur 20.

Q1 Lors de la création des contrôles, on souhaite que :

- lorsque la date de fin n'est pas spécifiée alors elle doit être égale à la date de début ;
- la date de fin doit être égale à la date de début pour les types de contrôle concernés (tests, tests rattrapage, interrogations) ;
- par défaut la note maximale du barème doit être à 20.

Définir un trigger se déclenchant automatiquement à l'insertion de nouveaux contrôles, ce qui permet de donner des valeurs par défaut. Il faut également penser aux opérations de modification affectant les attributs `TypeC`, `DateDeb`, `DateFin` ou `NoteMax`.

Q2 Lors de la saisie des notes, on veut pouvoir calculer la note définitive à partir de l'attribut `Note` (qui n'est pas évalué sur 20 mais sur `NoteMax` de `CONTROLE`). Donnez le Trigger associé.

Exercice 2

Une bibliothèque utilise une base de données pour gérer ses abonnés, son fonds documentaire, les emprunts d'exemplaires et les réservations de livres. Le schéma de cette base est le suivant :

```
ABONNE      (IdA, Nom, Prenom, Tel, Adresse, NbEmprunt)
LIVRE       (IdL, Titre, Editeur)
EXEMPLAIRE  (CodBar, IdL, Etat, An_achat)
EMPRUNT     (CodBar, DateE, IdA, DateRet)
RESERV      (IdL, IdA, DateR, Contact)
```

Remarques :

- Les clefs primaires sont soulignées. Les clefs étrangères portent le même nom que les clefs primaires associées.
- Toutes les contraintes de domaine, clef primaire et clef étrangère ont été spécifiées à la création de la base.
- L'attribut `Etat` est un énuméré dont le domaine est : {Bon, Moyen, A réparer ...}.
- Lorsqu'un abonné emprunte un exemplaire de livre, un tuple est créé dans la relation `EMPRUNT` mais l'attribut `DateRet` est à `NULL`. Il est valorisé, avec la date du jour, lorsque l'abonné rapporte l'exemplaire emprunté.
- L'attribut `NbEmprunt` est calculé. Il indique le nombre d'emprunts en cours des abonnés.
- Les attributs `DateE`, `DateRet`, `DateR` sont compatibles et de type `Date`.

Q1 On souhaite mettre en œuvre la contrainte suivante : un abonné ne peut faire un nouvel emprunt que si son nombre d'emprunts courants est inférieur à 5.

Ecrire un trigger permettant de vérifier la contrainte automatiquement si le nombre d'emprunts est trop grand et de faire échouer l'insertion.

Q2 Au fur et à mesure que des emprunts sont créés ou achevés (valorisation de l'attribut `DateRet`), on souhaite mettre à jour l'attribut calculé `NbEmprunt` dans la relation `ABONNE`. On suppose qu'il est impossible de modifier par `UPDATE` l'attribut `DateRet` en le mettant à `NULL`. On lui donnera forcément la date du jour.

Définir un trigger sur la relation `EMPRUNT` permettant d'effectuer la propagation automatique des mises à jour sur l'attribut `NbEmprunt` qui est incrémenté après la création d'un emprunt et décrémenté dès qu'un emprunt est terminé.

Q3 Lorsqu'un livre est réservé et qu'un exemplaire de ce livre est rapporté à la bibliothèque par un abonné qui l'avait emprunté, on souhaite effectuer les opérations suivantes :

- Contacter l'abonné prioritaire parmi tous ceux qui ont réservé le livre considéré. L'abonné prioritaire est celui dont la date de réservation est la plus ancienne. S'il y en avait plusieurs, le choix est fait, parmi eux, de manière arbitraire.
- Mettre à jour l'attribut `Contact` de la relation `RESERV` qui prendra la valeur 'Oui' pour la réservation sélectionnée.
-

Ecrire un trigger sur la relation `EMPRUNT` permettant d'effectuer le traitement demandé de manière automatique dès qu'un emprunt est achevé (valorisation de l'attribut `DateRet`).

Vous supposez qu'il existe une procédure `Avertir` en admettant en paramètres le nom, le prénom et l'adresse de l'abonné ainsi que le titre du livre. Cette procédure éditera un courrier pour l'abonné.