

GIS 2^{ème} année - Devoir Surveillé S.G.B.D.

© Olivier Caron

Durée 2 heures

1 Trigger

Soit une table relationnelle :

```
Personne(numID integer primary key,  
         nom varchar(20) not null,  
         age integer,  
         salaire float) ;
```

On désire faire un contrôle discret sur l'utilisation de cette table à l'aide de trigger(s).

On veut savoir, pour chaque utilisateur postgres, le nombre de consultations et le nombre de lignes modifiées sur la table `Personne`.

Ces informations seront stockées dans une ou plusieurs tables.

Remarque : dans le langage plpgsql, on dispose automatiquement de la variable `CURRENT_USER` de type `text` qui indique le nom de l'utilisateur postgres courant.

Donnez les commandes nécessaires pour réaliser ce contrôle : définition des tables, procédures stockées et triggers.

2 Location de vidéos

Une personne désire se lancer dans la location de vidéos (en toute légalité).

Le loueur désire connaître et informatiser les informations suivantes :

- l'ensemble des vidéos qu'il possède. Une vidéo est identifiée par un numéro et est caractérisée par un titre, une année de parution, et un format.
- un format vidéo est identifié par un numéro et est caractérisé par un libellé (exemple : 'vhs', 'dvd', 'cdrom', ...) et un prix de base de location.
- l'ensemble des clients. Un client est identifié par un numéro et est caractérisé par un nom, un age et une somme (qui correspondra, en fait, à la somme de toutes les factures effectuées par le client).
- l'ensemble des factures. Une facture est identifiée par un numéro, correspond à un client. Cette facture décrit l'ensemble des vidéos louées par le client à une date donnée ainsi que le prix total de la facture. Un client ne peut louer des vidéos que pour une seule journée, une vidéo empruntée est donc ré-empruntable le lendemain d'établissement de la facture.

2.1 Conception

Proposez un schéma conceptuel (avec le modèle UML) correspondant à ce système d'informations.

2.2 Schéma relationnel

En fonction de votre schéma conceptuel, donnez le schéma relationnel correspondant.

2.3 Administration

Le loueur a tous les droits sur les toutes les tables de la base.

Pour des raisons de sécurité, le loueur ne veut montrer (uniquement en lecture) à ses clients membres qu'une partie du système d'information qui pourra avoir la forme suivante :

```
les_videos(numeroVideo,titre, libelle_format, prix_location)
les_formats(libelle_format, prix_location)
```

Les clients n'ont aucun droit d'écriture. Donner les différentes commandes SQL d'administration pour établir cette représentation cliente et sécuriser le système.

2.4 Programmation

Le loueur désire construire un programme Java JDBC qui effectue une facture simplifiée (c'est à dire une facture ne comportant qu'une seule vidéo).

Le titre de la cassette vidéo ainsi que le nom du client est transmis au programme. Ce dernier établit une facture que si les conditions suivantes sont réalisées :

- le client est référencé dans la base.
- la vidéo existe
- la vidéo n'est pas empruntée

Ecrire la procédure java `float creerFacture(String nomClient, String titre)` qui effectue la facture en s'assurant bien que la base de données reste cohérente. En particulier, il faut prévoir le fait que plusieurs clients peuvent établir une facture au même moment.

On considère que la base est déjà connectée, vous disposez donc, dans la méthode `creerFacture` des attributs déjà initialisés :

```
Connection db ;
Statement s ;
```

Cette procédure retourne un réel. Cette valeur correspond à la somme de toutes les factures effectuées par le client (en tenant compte de celle qui vient d'être créée). Si la dernière facture n'a pu être établie, `creerFacture` retourne `0.0` ;