

TP3 : PL/SQL

On conserve les relations définies dans le TP1, en apportant les modifications et ajouts suivants :

* A la relation *livres*, on ajoute un attribut *note_moy* : pour un livre donné, cet attribut est calculé en faisant la moyenne des notes qui lui ont été attribuées dans la relation *Avis*.

* On ajoute des relations. La librairie propose des parcours sur différents thèmes, chaque parcours étant constitué d'événements ayant lieu à la librairie (exposés d'auteurs, discussions, etc). Tout client peut s'inscrire aux parcours de son choix. Les relations proposées sont les suivantes :

parcours(idp varchar2(10), intitulep varchar2(15), genre varchar2(15), date_deb date)

Les parcours ont un intitulé et peuvent concerner un genre (ex: policier) ; si ce n'est pas le cas, le champ correspondant prend la valeur indéfinie. Un parcours commence à une certaine date.

compo_parcours(idp varchar2(10), id_evt varchar2(10))

Cette relation mémorise la composition des parcours ; un parcours contient 4 événements. Un même événement peut être proposé pour différents parcours.

inscrip_parcours(idcl number, idp varchar2(10))

Cette relation mémorise l'inscription d'un client à un parcours.

inscrip_evt(idcl number, idp varchar2(10), id_evt varchar2(10))

Pour des raisons de logistique (capacité de la salle, etc), on maintient une relation mémorisant les inscriptions des clients aux différents événements.

Mise à jour de la note moyenne

Q1. Modifier la table *livres* en ajoutant l'attribut *note_moy*. (a) Ecrire un bloc PL/SQL qui met à jour cet attribut à partir de la relation *Avis*, pour un livre passé en paramètre ; (b) Ecrire un bloc PL/SQL qui met à jour cet attribut pour tous les livres. (c) Reprendre la question précédente en écrivant une procédure.

Q2. Ecrire un déclencheur (trigger) qui à l'ajout ou à la modification d'un avis, met à jour l'attribut *note_moy* de la référence de livre concernée. Que constatez-vous ? Commentez.

Cohérence Avis-Achat

Q1. Ecrire un déclencheur qui assure qu'un client qui insère un avis a bien acheté le livre.

Q2. Ecrire un bloc PL/SQL de mise à jour de la table *Avis* (la note et/ou le commentaire) qui assure qu'un client ne peut modifier que les tuples qu'il a insérés (on suppose que l'identifiant de client et la référence du livre sont passés en paramètre).

Traitement d'une inscription à un parcours

Implanter le schéma des relations : *parcours*, *compo_parcours*, *inscrip_parcours*, *inscrip_evt*.

Q1. Ecrire un code PL/SQL qui réalise l'inscription d'un client à un parcours (clients et parcours passés en paramètres) : le tuple correspondant est inséré dans la relation *inscrip_parcours*. La relation *inscrip_evt* est remplie automatiquement (cf *compo_parcours*). Transformer en procédure.

Q2. Ecrire un déclencheur pour la relation *Achats* qui propose des parcours (qui n'ont pas encore commencé) de même genre que celui du livre acheté.