

## SGBD- TD2. PLSQL-part2-Les triggers

### Exercice 1 :

1. Soit l'extrait suivant du schema relationnel Gestpersonnel :

- *personne* (id\_pers, nompers, fonction, date\_debFonction, numchef, salaire, #dept) – *table de description d'un employé*
- *projet* (id\_projet, titre, resp\_proj) – *table de description des projets*
- *persprojet* (#id\_proj, #id\_pers, quota) *table de participation d'un employé donné à un projet*
- *job\_history* (#id\_pers, Date\_deb, Date\_fin, fonction, # dept) ;

Nous présentons le jeu de données suivant :

#### persprojet :

id_proj	id_pers	quota
Proj10	100	10
Proj10	101	
Proj10	102	100
Proj13	103	
Proj13	104	
Proj13	101	50

#### Projet

Id_projet	titre	resp_proj
Proj10	projetAA	101
Proj13	projetBB	103

1. Créer un programme permettant d'automatiser la règle de gestion suivante :

Refuser de nommer responsable d'un projet quelqu'un qui n'y participe pas.

- Faites appel à une fonction `f_verifparticipation` qui vérifie la participation d'un employé donné à un projet,

#### Exemples d'exécution

Insert into projet (id\_projet, resp\_proj) values ('proj10', 103) ;

Update projet set resp\_proj=103 where upper(id\_projet)=upper('proj10')

\*

ERROR at line 1:

ORA-20100: cet employé ne participe pas au projet !

2. Pour interdire que la somme des quotas de participation d'un employé aux projets ne dépasse les 100h , nous avons développé le trigger suivant :

*Create or replace trigger trgheursProjet*

*before insert or update on persprojet*

*For each row when (new.quota is not null)*

*Declare*

*somme integer;*

*Begin*

*Select sum(quota) into somme from persprojet where id\_pers=:new.id\_pers ;*

*If somme+:new.quota > 100 then raise\_application\_error(-20200, 'attention au surmenage');*

*End if;*

*End;*

*Qu'en pensez vous ? si on tente d'exécuter la ligne suivante ?*

SQL> Update persprojet set quota=55 where id\_pers=101 and upper(id\_proj)= upper('proj10') ;

3. Un employé peut changer de poste ou de département au sein de la même entreprise durant sa carrière professionnelle. Toute modification doit être accompagnée par un enregistrement de la trace de son emploi passé dans la table `job_history`. Programmer cette règle de gestion.

### Exercice 2

#### Règles de Gestion :

1. Toute modification de la structure de base de données doit être effectuée durant les heures de travail (de lundi à vendredi entre 08 à 18 heures) sinon la modification doit être refusée.

2. Soit la table pointeuse définie comme suit :

Create table pointeuse(utilisateur varchar(50), msg varchar2(100), dateheure varchar2(100) ;

Créer un programme permettant de retracer toute les connexions à la base en particulier à chaque connexion, insérer dans la table de pointage l'utilisateur connecté, un message 'connecté le : date de connexion (jour mois année heure, minutes et secondes)