

### R3.07 SQL dans un langage de programmation

Pour chaque trigger, vous indiquez : voir tableau ci-dessous.

**Ce TD est à déposer sur madoc. Votre rapport contient aussi tous les jeux d'essais pour invoquer les trigger et déclencher les différentes erreurs.**

Nom Trigger	Type : before ou after	Insert, delete, update	Nom Table	For each row :oui ou non

#### Exercice 1 : Triggers de type For each row et utilisation de « :NEW » et « :OLD »

A- Ecrire un trigger de type **for each row** qui interdit la diminution du salaire d'un employé. Ce trigger se déclenche après la modification du salaire.

B- Il y a une autre contrainte qui n'est pas spécifiée "la durée hebdomadaire d'un employé ne peut pas augmenter", elle n'est pas descriptive. Vous écrivez le trigger nécessaire à la vérification de cette contrainte

#### Exercice 2 : Trigger de type : Delete from T2 where a not in (select a from T1) /

A – La spécification de l'opération **supprimer\_employe** impose que la suppression d'un employé soit accompagnée de la suppression des lignes de **travail** correspondantes. Mettez en place un trigger table qui le fait. (pas de problème si on a déclaré "**deferred**" la contrainte FK\_employe de la table **travail** vers la table **employe** "les employés de travail existent").

*Attention : la suppression des employés de la table travail n'est possible que si l'employé n'est pas chef de service ou responsable de projet*

B – La spécification de l'opération **supprimer\_projet** impose que la suppression d'un projet soit accompagnée de la suppression des lignes de **travail** et de la table **concerne** correspondantes. Mettez en place un trigger table qui effectue cela. (pas de problème si on a déclaré "**deferred**" la contrainte FK\_nuproj de la table **travail** et de la table **concerne** vers la table **projet**).

**Exercice 3 :** Pour les triggers qui suivent, vous faites des M.A.J(insert, update) de la base de données qui déclenchent les différentes erreurs

A – Il y a une contrainte qui n'est pas spécifiée "la somme des durées de travail d'un employé ne doit pas excéder son temps de travail hebdomadaire", elle n'est pas descriptive. Vous écrivez le (ou les) trigger nécessaire à la vérification de cette contrainte.

La contrainte à mettre en place **SUM(duree)<=hebdo**.

- Quels sont les différentes opérations(update, insert,...) sur les tables employe ou travail qui vous amènent à ne pas respecter cette contrainte.

- Combien de trigger allez vous mettre en place ? Indiquez dans quel cas les trigger se déclenchent.

Vous utilisez la requête suivante pour construire ce(s) trigger :

**Select \* from employe e  
where (select sum(duree) from travail t where e.nuempl=t.nuempl)> hebdo ;**

B - Ecrire un *trigger* qui vérifie la contrainte suivante: « un employé est **responsable** au plus sur 3 projets ».

Idem que la question précédente, vous utilisez la requête suivante pour construire votre trigger :

**Select \* from employe e  
where (select count(\*) from projet p where e.nuempl=p.resp)> 3 ;**

C- Ecrire un trigger qui vérifie la contrainte suivante : « un service ne peut être concerné par plus de 3 projets ».

D- Ecrire un trigger qui vérifie la contrainte suivante : « un chef de service gagne plus que les employés de son service.

E- Ecrire un trigger qui vérifie la contrainte suivante : « un chef de service gagne plus que les employés responsables de projets.

F- Est-il possible de regrouper les deux derniers « trigger »

#### **Exercice 4 :**

Lors d'augmentation de salaire ou d'embauche, l'entreprise veut enregistrer les employés (dans la table EMPLOYEALERTE idem que EMPLOYE) avec un salaire qui dépassent les 5000 euros.

Ecrire un trigger qui permet de remplir cette table.

---

### Exercice 1 :

```
CREATE TRIGGER TRIG1BIS AFTER  
DELETE ON EMPLOYE  
BEGIN  
DELETE FROM Travail WHERE nuempl NOT IN (SELECT nuempl FROM Travail);  
END;
```

```
CREATE TRIGGER TRIG1BIS AFTER  
DELETE ON Projet  
BEGIN  
DELETE FROM Travail WHERE nuproj NOT IN (SELECT nuproj FROM Travail);  
END;
```

### Exercice 2 :

```
CREATE TRIGGER TRIG2 BEFORE UPDATE  
SALAIRE ON EMPLOYE FOR EACH ROW  
BEGIN  
IF :NEW.SALAIRE < :OLD.SALAIRE THEN  
RAISE_APPLICATION_ERROR (-20001,'viol de la règle : une salaire ne peut pas  
être diminué');  
END IF ;  
END;
```

### Exercice 3 :

le trigger se déclenche après insertion ou modification de durée de la table travail ou modification de hebdo de la table employe.

T1 est remplacé par la table Employe et T2 par la table travail.

Reprendre les requêtes et rajouter « **INTO REC\_EMPLOYE** » après « **select \***. »

```
CREATE TRIGGER TRIG3 AFTER .... ON T2 ( ou T1)  
BEGIN  
DECLARE  
REC_T1 T1%ROWTYPE;  
BEGIN  
SELECT * INTO REC_T1 FROM T1 // Rep  
WHERE Condition ;  
  
//1 seul enregistrement trouvé en lève une exception  
RAISE_APPLICATION_ERROR (numéro, message);//1  
EXCEPTION  
WHEN NO_DATA_FOUND THEN NULL;//0 enregist  
WHEN TOO_MANY_ROWS THEN  
RAISE_APPLICATION_ERROR ( ... );// 2 ou pl+  
END;
```

**Exercice 4 :**

**CREATE TRIGGER TRIG4 AFTER INSERT or UPDATE OF**

**Salaire on EMPLOYE FOR EACH ROW WHEN**

**( :NEW.salaire > 5000)**

**BEGIN**

**INSERT INTO EMPLOYE\_ALERTE**

**VALUES**

**(:NEW.NUMEMPL,:NEW.NOMEMPL,:NEW.Hedbdo,:NEW.AFFECT, :NEW.SALAIRE);**

**END;**

la requête à utiliser pour le trigger :

```
select * from projet p
  where exists (select * from employe e
                where nuempl in (select nuempl from travail t1
                                where t1.nuempl=e.nuempl
                                and t1.nuproj=p.nuproj
                                and t1.duree > (select duree from travail t2
                                                where t2.nuempl=p.resp
                                                and t2.nuproj=p.nuproj)));
```

modifiez les lignes de la table travail telque le résultat de la requête soit vide.

Exemple : **update travail set duree=V1 where nuempl=e1 and projet=p1.**

**Il faudra diminuer la durée de travail de l'employé e1 et du projet p1 avec v1 inférieur ou égal à la durée de travail du responsable de projet p1**

6- Ecrire un trigger qui vérifie la contrainte suivante : « le nombre d'employés sur un projet est  $\geq 3$  et  $\leq 8$  ».

7- Ecrire un trigger qui vérifie la contrainte suivante : « Un employé est responsable d'au plus de 3 projet ».

8- Rajouter un champ ***nbprojets*** à la table *employe*. Ce champ contient le nombre de projets de chaque employé dans la table travail.

Ecrire un trigger qui permet de modifier le champ nbprojets à chaque M.A.J de la table travail.