

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

Ce TD sera déposé sur madoc avec le TD3. 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) / D'autres types de solutions ne sont pas acceptés pour cette exercice.

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.

Tous les trigger de cette exercice sont de type select * into suivi de l'instruction raise_application_error. D'autres solutions ne seront pas acceptées.

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 EMPLOYE_ALERTE idem que EMPLOYE) avec un salaire qui dépassent les 5000 euros.

Ecrire un trigger qui permet de remplir cette table.
