

## TD4 : Triggers

### Exercice 1:

---

Soit une table quelconque TABLE, dont la clé primaire CLENUM est numérique.

Définir un trigger en insertion permettant d'implémenter une numérotation automatique de la clé. Le premier numéro doit être 1.

```
Create trigger cleauto before insert on TABLE
for each row
begin
declare n int ;
declare newkey int;
select count(*) into n from TABLE;
if n=0 then
set newkey=0;
-- Recherche la valeur de cle C la plus elevee
-- et affecte C+1 a la nouvelle cle
else select max(num) into newkey from TABLE;
end if;
set new.num_TABLE = newkey +1;
end|
```

### Exercice 2 :

---

Ecrire un trigger permettant d'empêcher l'insertion d'un nouvel étudiant en affichant une erreur si le nom et le prénom de l'étudiant inséré existe déjà.

```
Create trigger Exo2 before insert on etudiant for each row
BEGIN
if exists (select nom, prenom from etudiant where new.nom=nom and
new.prenom=prenom) then
raise_application_error (-20001, 'Etudiant existe déjà');
end if;
END
```

### Exercice 3 :

---

On considère la base de données suivante :

PILOTE (**NUMPIL**, NOMPIL, PRENPIL, ADR, SAL)  
AVION (**NUMAV**, NOMAV, CAP, LOC)  
VOL (**NUMVOL**, NUMPIL, NUMAV, VILLE\_DEP, VILLE\_ARR, H\_DEP, H\_ARR)

Ecrire un trigger qui après chaque suppression d'un pilote annule tous ses vols prévus après la date de son suppression.

```
CREATE OR REPLACE TRIGGER T1
AFTER DELETE ON pilote
FOR EACH ROW
BEGIN
DELETE FROM Vol
WHERE DATE_DEP > Date() and NUMPIL=old. NUMPIL;
END;
```

## Exercice 4 :

Écrire un trigger qui permet de mettre le salaire à 5000 si un employé n'a pas de salaire.

```
CREATE TRIGGER DefaultSalaire
BEFORE INSERT OR UPDATE OF salaire ON employe
FOR EACH ROW
WHEN (new.salaire is null)
BEGIN
SELECT 5000 INTO :new.salaire
FROM dual;
END;
/--La table DUAL est une table spéciale d'une seule colonne et d'une
seule ligne présente par défaut dans toutes les installations
du SGBD Oracle. Elle est utilisée généralement pour sélectionner les
pseudo-colonnes telles que SYSDATE ou USER.
```

## Exercice 5 :

1. Ecrire un trigger permettant d'empêcher l'insertion d'une opération de débit sur le compte 500 en affichant une erreur « Toute opération de débit est actuellement interdite sur le compte N° 500 ».

```
Create or replace trigger Trig1 before insert on operation for each
row
BEGIN
If :new.Compte.ID_Compte=500 and :new.valeur<0 then
raise_application_error (-20010, ' Toute opération de débit est
actuellement interdite sur le compte N° 500');
end if ;
END ;
```

2. Ecrire un trigger qui sera déclenché par une opération d'insertion dans la table Operation. On considère deux situations :

- (1) valeur de l'opération est positive. Dans ce cas l'opération est toujours effectuée et le trigger ajoute valeur de l'opération au solde du compte dans la table Compte.
- (2) si valeur est négative alors le trigger :

- Effectue l'opération si le solde après l'opération reste supérieur ou égal au plafond. Le trigger doit modifier le solde du compte dans la table Compte en ajoutant valeur au solde (rappelons que valeur est négative ici).
- Si le solde après l'opération deviendrait inférieur au plafond l'opération doit être annulée. L'opération annulée ne sera pas enregistrée dans la table Operation.

```
Create or replace trigger Trig2
before insert on operation
for each row
Declare
NewSolde Number ;
Plafond Number ;
BEGIN
Select c.solde into NewSolde from compte c where
c.ID_Compte=:new.compte.ID_Compte ;
NewSolde:=:newSolde+:new.valeur ;
If :new.valeur>0 then
Update Compte set solde=:newSolde where
Compte.ID_Compte=:new.compte.ID_Compte ;
Else
Select c.plafond into Plafond from Compte c where
c.ID_Compte=:new.Compte.ID_Compte ;
If :newSolde>=plafond then
Update Compte set solde=:newSolde where
Compte.ID_Compte= :new.compte.ID_Compte ;
else
raise_application_error (-20010, 'Cette opération de débit est
annulée parce que le solde sera inférieur au plafond');
end if ;
end if ;
END ;
```