

Les triggers :

Définition

Les triggers sont des procédures stockées qui s'exécutent automatiquement quant un événement se produit. En général cet événement représente une opération DML (Data Manipulation Language) sur une table. Les instructions DML doivent inclure INSERT, UPDATE ou DELETE

Ils permettent entre autre de contrôler les accès à la base de données.

Syntaxe simplifiée pour créer un trigger avec une opération DML

```
CREATE [OR REPLACE] TRIGGER nomtrigger
BEFORE [ AFTER] INSERT OR UPDATE OR DELETE
ON Nomdetable [FOR EACH ROW] [WHEN condition]
BLOC PL/SQL
```

Attention: le nom du trigger doit être unique dans la base de données.

- L'option BEFORE /AFTER indique le moment du déclenchement du trigger.
- Les instructions SQL INSERT OR UPDATE OR DELETE peuvent être toutes présentes comme on peut en avoir juste une.
- Pour un UPDATE, on peut spécifier une liste de colonnes. Dans ce cas, le trigger ne se déclenchera que s'il porte sur l'une des colonnes précisées dans la liste. (voir exemple plus loin)

Types de TRIGGERS oracle

Oracle propose deux types de triggers :

1. Les triggers lignes qui se déclenchent individuellement pour chaque ligne de la table affectée par le trigger. L'option FOR EACH ROW est alors spécifiée

http://download.oracle.com/docs/html/B16022_01/ch3.htm#i1017636

http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/statements_7004.htm

2. Les triggers globaux qui sont déclenchés une seule fois, L'option FOR EACH ROW n'est pas spécifiée

Exemple1 (TRIGGER GLOBAL)

```
CREATE OR REPLACE TRIGGER ctrlmiseajour
BEFORE INSERT OR DELETE OR UPDATE ON EMPLOYES
DECLARE MESSAGE EXCEPTION;
BEGIN
IF (TO_CHAR(SYSDATE,'DY')= 'SAM.' OR TO_CHAR(SYSDATE,'DY')= 'DIM.')
THEN RAISE MESSAGE;
END IF;
EXCEPTION
WHEN MESSAGE THEN RAISE_APPLICATION_ERROR(-20324,'on ne met
pas à jour la fin de semaine');
END;
```

On peut également utiliser les triggers pour contrôler les valeurs qui ont été insérées. L'exemple le plus souvent utilisé et l'utilisation de trigger pour faire une incrémentation automatique de la clé primaire.

Exemple 2 TRIGGER SUR CHAQUE LIGNE

```
CREATE OR REPLACE TRIGGER EMPINS
```

```
BEFORE INSERT ON EMPLOYES
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF :NEW.NUMEMP IS NULL THEN
```

```
SELECT SQEMP2 .NEXTVAL INTO :NEW.NUMEMP FROM DUAL;
```

```
END IF;
```

```
END;
```

http://download.oracle.com/docs/html/B16022_01/ch3.htm#i1017636

http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/statements_7004.htm

La séquence a été créée comme suit :

```
CREATE SEQUENCE SQEMP2INCREMENT BY 1  
START WITH 10000;
```

On exécute ensuite l'instruction INSERT sans le numemp;

IMPORTANT : lors de la création d'un trigger, les anciennes valeurs sont nommées

:old.colonne (faire attention aux deux points)

Les nouvelles valeurs sont nommées

:new.colonne (faire attention aux deux points)

Exemple 3

Contrôler les salaires : ils doivent être plus grand que le plus petit des salaires et plus petit que le plus grand des salaire

```
CREATE OR REPLACE TRIGGER CTRLsalaire  
BEFORE INSERT OR UPDATE OF salaire ON employes  
FOR EACH ROW  
DECLARE  
minsal number;  
maxsal number;  
BEGIN  
/* retrouver le salaire minimum et maximum dans la table employes  
SELECT MIN(salaire), Max(salaire) INTO minsal, maxsal FROM Employes;  
/* s'il y a un problème, on provoque une erreur */  
IF (:new.salaire<minsal OR :new.salaire>maxsal)  
THEN  
raise_application_error (-20300,'Salaire incorrect');  
END IF;  
END;
```

Exercice :

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

Solution:

```
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;
```

Activation et désactivation d'un trigger

Pour qu'un trigger soit déclenché celui-ci doit être **ACTIVÉ**. Par défaut, les triggers d'oracle sont activés dès leur création. On peut activer un triggers par SQL Développeur ou par la commande **ALTER TRIGGER nomtrigger ENABLE**.

Si un trigger est désactivé, alors il est stocké mais ignoré

Pour désactiver un trigger, utiliser la même commande **ALTER TRIGGER** avec l'option **DISABLE**.

On peut désactiver **TOUS** les triggers associés à une table par la commande **ALTER TABLE DISABLE ALL TRIGGERS**.

Pour les réactiver de nouveau, utiliser **ALTER TABLE ENABLE ALL TRIGGERS**.

Pour supprimer un trigger, **utiliser DROP TRIGGER nomduTrigger**

http://download.oracle.com/docs/html/B16022_01/ch3.htm#i1017636

http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/statements_7004.htm

Les prédicats INSERTING, UPDATING, DELETING

Lorsqu'un trigger porte sur les opérations DML, nous pouvons ajouter des prédicats dans notre code, pour indiquer les opérations de déclenchement :

```
CREATE TRIGGER ...  
BEFORE INSERT OR UPDATE ON employe  
.....  
BEGIN  
.....  
IF INSERTING THEN ..... END IF;  
IF UPDATING THEN ..... END IF;  
.....  
END;
```

Exemple :

On veut enregistrer les opérations réalisées sur notre table

1

```
CREATE TRIGGER ctrlOperation  
BEFORE  
DELETE OR INSERT OR UPDATE  
ON Employe FOR EACH ROW  
BEGIN  
IF INSERTING THEN  
INSERT INTO table_ctrl VALUES ( USER, 'Insertion');  
ELSIF DELETING THEN  
INSERT INTO table_ctrl VALUES ( USER, 'Destruction');  
ELSIF UPDATING THEN  
INSERT INTO table_ctrl VALUES ( USER, 'Modification');  
ELSIF UPDATING(salaire) THEN  
INSERT INTO table_ctrl VALUES ( USER, 'Modification du salaire');  
END IF;  
END;
```

¹ Source : SQL, Denis Brunet

http://download.oracle.com/docs/html/B16022_01/ch3.htm#i1017636

http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/statements_7004.htm

IMPORTANT :

Un trigger ligne ne peut pas lire (SELECT) et/ou modifier la table concernée (appelée table mutante) par l'instruction (INSERT, UPDATE ou DELETE) qui a déclenché ce trigger.

Exercice 1 :

Écrire un trigger qui permet de contrôler les insertions et les modifications dans la table Commandes, comme suit :

La date de livraison ne doit pas être inférieure à la date de commande

Exercice 2

Écrire un trigger qui permet de contrôler les ajouts et les modifications des salaires. Si le JOB n'est pas 'PRESIDENT', alors Lors de l'ajout, modification d'un nouvel employé, nous devons vérifier si le salaire de celui-ci se retrouve dans la fourchette des salaires. Les salaires sont en fonction de JOB et sont dans la table Grille