

Les Triggers

Un trigger (déclencheur) est un programme qui se déclenche automatiquement suite à un événement. C'est donc un programme que l'on n'appelle pas explicitement, à la différence d'une procédure stockée.

Déclenchement

- Les triggers sont déclenchés par des instructions LMD qui modifient l'instance d'une table :INSERT, DELETE ou UPDATE. (il existe sous Oracle d'autres types de triggers)

Utilité

Les triggers peuvent servir à :

- vérifier des contraintes que l'on ne peut pas définir de façon déclarative
- gérer de la redondance d'information, après une dénormalisation du schéma.
- collecter des informations sur les mises-à-jour de la base.

Structure d'un trigger

```
CREATE [OR REPLACE] TRIGGER nom_trigger
instant liste_evts
ON nom_table [FOR EACH ROW]
[WHEN ( condition ) ]
corps
instant ::= AFTER | BEFORE
liste_evts ::= evt {OR evt}
evt ::= DELETE | INSERT |
        UPDATE [OF { liste_cols }]
liste_col ::= nom_col { , nom_col }
corps corps de pgme PL/SQL
```

Entête du trigger

On définit :

- la table à laquelle le trigger est lié,
- les instructions du DML qui déclenchent le trigger
- le moment où le trigger va se déclencher par rapport à l'instruction LMD (avant ou après)
- si le trigger se déclenche une seule fois pour toute l'instruction (i.e. trigger instruction), ou une fois pour chaque ligne modifiée/insérée/supprimée. (i.e. trigger ligne, avec l'option FOR EACH ROW)
- et éventuellement une condition supplémentaire de déclenchement (clause WHEN)
-

After ou Before ?

- Si le trigger doit déterminer si l'instruction LMD est autorisée, utiliser BEFORE
- Si le trigger doit "fabriquer" la valeur d'une colonne pour pouvoir ensuite la mettre dans la table : utiliser BEFORE
- Si on a besoin que l'instruction DML soit terminée pour exécuter le corps du trigger : utiliser AFTER
-

Trigger ligne ou instruction ?

Un trigger instruction se déclenche une fois, suite à une instruction LMD.

Un trigger ligne (FOR EACH ROW) se déclenche pour chaque ligne modifiée par l'instruction DML.

Par exemple, si une instruction update sur une table T modifie 5 lignes, un trigger instruction lié à cet événement se déclenchera une seule fois (avant ou après la modification en fonction de after/before) et un trigger ligne se déclenchera 5 fois, une fois par ligne modifiée (avant ou après la modification en fonction de after/before).

Dans un trigger ligne, on peut faire référence à la ligne courante, celle pour laquelle le trigger s'exécute.

Pour cette ligne, on a accès à la valeur avant l'instruction DML (nommée :old) et à la valeur après l'instruction (nommée :new).

	:old	:new
insert	null	ligne insérée
delete	ligne supprimée	null
update	ligne avant modif	ligne après modif

Clause When

On peut définir une condition pour un trigger ligne : le trigger se déclenchera pour chaque ligne vérifiant la condition.

Clause When

On peut définir une condition pour un trigger ligne : le trigger se déclenchera pour chaque ligne vérifiant la condition.

Le trigger suivant insère une ligne dans une table de log, lorsque le salaire d'un employé diminue.

```
Create or replace trigger journal_emp
after update of salary on EMPLOYEE
for each row
when (new.salary < old.salary) -- attention, ici on utilise new
et pas :new
begin
    insert into EMP_LOG(emp_id, date_evt, msg)
    values (:new.empno, sysdate, 'salaire diminué');
end ;
```

-
-

Ordre d'exécution des triggers

Pour une instruction du LMD sur une table de la base, il peut y avoir 4 familles de triggers possibles selon l'instant (before, after) et le type (instruction ou ligne).

Ordre d'exécution des triggers

Ces triggers se déclenchent dans l'ordre suivant :

- Triggers instruction BEFORE
- Triggers ligne BEFORE (déclenchés n fois)
- Triggers ligne AFTER (déclenchés n fois)
- Triggers instruction AFTER
-

Ordre d'exécution d'un trigger

Dans une même famille, on ne contrôle pas l'ordre d'exécution des triggers :

par exemple :

s'il y a plusieurs triggers instructions after pour l'instruction update sur une table T, on ne sait pas dans quel ordre ils vont s'exécuter.

Corps du trigger

Nous avons vu précédemment que dans le corps d'un trigger ligne, on peut faire référence aux valeurs de la ligne courante (avant et après modif) par :old et :new.

Il ne faut pas, dans un trigger ligne, interroger une table qui est en cours de modification (pb mutating table). En effet, l'état de la table est instable puisque l'instruction SQL est en cours d'exécution.

Exemple

Une table contenant des etudiants (numero, nom, prenom...) avec leur moyenne

Table etudiant

NumEtudiant	Nom	Prenom	Moyenne
123	DUPONT	JULES	4
234	DUPOND	ALFRED	5
567	DURAND	JULIE	14
598	DURANT	ALFRED	16

Exemple de Trigger de niveau table

```
CREATE TRIGGER triggerEtudiant
  BEFORE INSERT, UPDATE ON etudiant
BEGIN
  DBMS_OUTPUT.ENABLE(10000);
  IF INSERTING THEN
    IF USER != 'VILNAT'
      THEN RAISE APPLICATION ERROR (-20001,'Utilisateur non autorise.');
```

END IF;

```
END IF;
  IF UPDATING THEN
    DBMS_OUTPUT.PUTLINE('Mise a jour de etudiant.');
```

END IF;

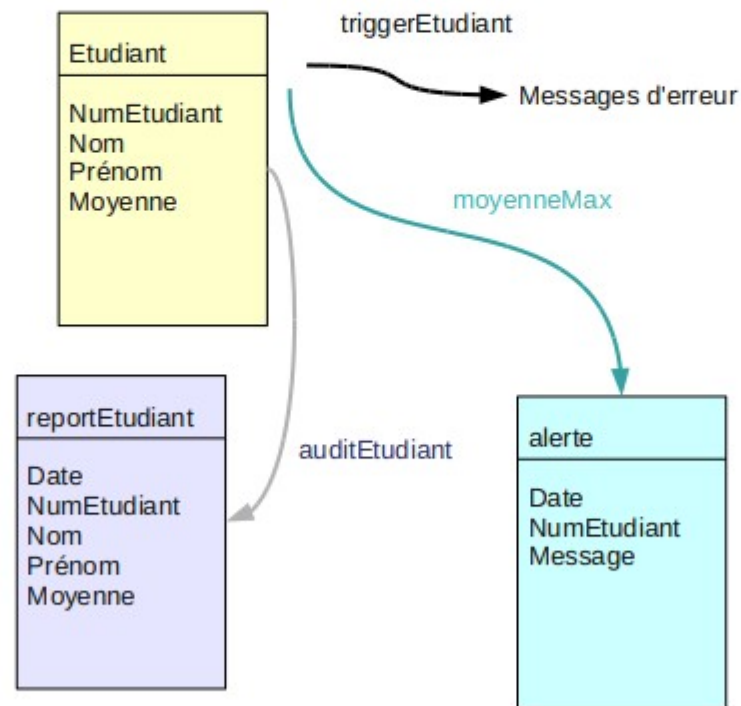
```
END;
```

Exemple de Trigger de niveau ligne

```
CREATE TRIGGER auditEtudiant  
AFTER UPDATE OR DELETE ON etudiant  
FOR EACH ROW  
BEGIN  
INSERT INTO reportEtudiant  
VALUES (SYSDATE, :OLD.NumEtud, :OLD.nom,  
:OLD.prenom, :OLD:moyenneEtud);  
END;
```

Exemple de trigger de niveau ligne avec clause WHERE

```
CREATE TRIGGER moyenneMax  
AFTER UPDATE ON etudiant  
FOR EACH ROW  
WHEN :NEW.moyenneEtud>2*:OLD.moyenneEtud  
BEGIN  
INSERT INTO alerte(datealerte, numEtudiant, message)  
VALUES ( SYSDATE, :OLD.nom,  
'MOYENNE a plus que double');  
END;
```



Les triggers qui vont "surveiller" les mises a jour sur la tableEtudiant, et soit acher des messages d'erreur en cas de probleme,soit alimenter d'autres tables..