

Définition

73

- Déclencheur ou trigger
 - Un bloc de code PL/SQL associé à une table, vue, schéma ou base de données
 - Exécuté suite à un événement et ne peut être appelé explicitement
- Avantages
 - Monitoring et audit
 - Journalisation des accès aux tables, connexions, ...
 - Sécurité et intégrité
 - Contrôle de l'accès (horaire, utilisateur)
 - Interdire les transactions non autorisées ou non valides (contraintes)
 - Dériver des données (calculs)

Types de triggers

74

- Selon le type d'événement
 - DDL : Suite à une opération CREATE, ALTER ou DROP
 - DML : Suite à une opération INSERT, UPDATE ou DELETE
 - Database : Suite démarrage ou connexion SHUTDOWN, STARTUP, LOGON, LOGOFF
- Selon niveau d'exécution
 - Row level : exécuté pour chaque ligne concernée
 - Statement level : exécuté une seule fois par événement
- Selon le moment d'exécution
 - BEFORE
 - AFTER
 - INSTEAD OF

Création de trigger

75

```
CREATE TRIGGER nom
  (événement déclenchant le trigger)
  ...
  (contrainte optionnelle du trigger)
BEGIN
  (action du trigger)
END;
```

Trigger de LMD(1)

76

```
CREATE [OR REPLACE] TRIGGER trigger_name
{BEFORE | AFTER}
{INSERT | DELETE | UPDATE | UPDATE OF column_list } ON
table_name
[FOR EACH ROW]
[WHEN (...)]
[DECLARE ... ]
BEGIN
...executable statements...
[EXCEPTION ... ]
END [trigger_name];
```

Trigger de LMD(2)

77

□ Exemple

```
CREATE OR REPLACE TRIGGER secu_emp
  BEFORE INSERT ON emp
BEGIN
  If(to_char(sysdate,'day') in ('samedi','dimanche') or
  to_char(sysdate, 'HH24') not between 8 and 18) then
    raise_application_error(-20500,'Vous ne pouvez
    insérer dans la table emp que pendant les heures
    de bureaux!');
  End if;
END;
```

Trigger de LMD(3)

78

□ Utilisation des prédicats conditionnels

```
CREATE OR REPLACE TRIGGER secu_emp
  BEFORE INSERT OR UPDATE OR DELETE ON emp
BEGIN
  If(to_char(sysdate,'day') in ('samedi','dimanche') or to_char(sysdate, 'HH24') not
  between 8 and 18) then
    If INSERTING then
      raise_application_error(-20500,
      'Vous ne pouvez insérer dans la table emp que pendant les heures de bureaux!');
    elsif DELETING then
      raise_application_error(-20501,
      'Vous ne pouvez supprimer de la table emp que pendant les heures de bureaux!');
    elsif UPDATING then
      raise_application_error(-20502,
      'Vous ne pouvez modifier la table emp que pendant les heures de bureaux!');
    else
      raise_application_error(-20503,
      'Vous ne pouvez insérer dans la table emp que pendant les heures de bureaux!');
    End if;
  End if;
END;
```

:New et :Old(1)

79

□ Référence au nouvel enregistrement et l'ancien enregistrement

□ Selon le type de l'opération

- INSERT
 - :New seulement est accessible
 - Modifiable si BEFORE
- UPDATE
 - :New et :Old sont accessibles
 - :New est modifiable si BEFORE
- DELETE
 - :Old seulement est accessible

:New et :Old(2)

□ Exemple :

```
CREATE OR REPLACE TRIGGER verifiersalaire
  BEFORE UPDATE ON emp
  FOR EACH ROW
BEGIN
  IF not (:New.poste='Directeur' AND :New.sal>2000) then
    Raise_application_error(-20300,'Salaire incorrect');
  END IF;
END ;
```

Trigger de LDD(1)

```
CREATE [OR REPLACE] TRIGGER trigger name
{BEFORE | AFTER} {<DDL_event>} ON {DATABASE | SCHEMA}
[WHEN (...)]
DECLARE
Variable declarations
BEGIN
...some code...
END;
```

Trigger de LDD(2)

DDL event	Fires when...
ALTER	Any database object is altered via the SQL ALTER command.
ANALYZE	Any database object is analyzed via the SQL ANALYZE command.
ASSOCIATE STATISTICS	Statistics are associated with a database object.
AUDIT	Auditing is turned on via the SQL AUDIT command.
COMMENT	Comments are applied to a database object.
CREATE	Any database object is created via the SQL CREATE command.
DDL	Any of the events listed here occur.
DISASSOCIATE STATISTICS	Statistics are disassociated from a database object.
DROP	Any database object is dropped via the SQL DROP command.
GRANT	Privileges are granted via the SQL GRANT command.
NOAUDIT	Auditing is turned off via the SQL NOAUDIT command.
RENAME	A database object is renamed via the SQL RENAME command.
REVOKE	Privileges are revoked via the SQL REVOKE command.
TRUNCATE	A table is truncated via the SQL TRUNCATE command.

Trigger de LDD(3)

□ Exemple :

```
CREATE OR REPLACE TRIGGER t_create
AFTER CREATE ON SCHEMA
BEGIN
  DBMS_OUTPUT.PUT_LINE('Opération de création!');
END;
```

Trigger de LDD(4)

84

□ Attributs

- Quelques détails sur l'événement LDD
 - ORA_CLIENT_IP_ADDRESS
 - ORA_DATABASE_NAME
 - ORA_DICT_OBJ_NAME
 - ORA_DICT_OBJ_TYPE

□ Exemple :

```
CREATE OR REPLACE TRIGGER t_create
AFTER CREATE ON SCHEMA
BEGIN
    DBMS_OUTPUT.PUT_LINE(ORA_DICT_OBJ_TYPE || ' créé');
END;
```

Triggers de base de données

85

□ Événement Utilisateur (USER)

- Connexion : LOGON
- Déconnexion : LOGOFF
- Démarrage : STARTUP
- Arrêt : SHUTDOWN

□ Exemple :

```
CREATE OR REPLACE TRIGGER log_connexion
AFTER LOGON ON SCHEMA
BEGIN
    INSERT INTO journal(user_id,date_action,action) VALUES (USER,
        SYSDATE,'Connexion');
END;
```

Manipulation des triggers

86

□ Activer/désactiver

- ALTER TRIGGER nom ENABLE | DISABLE;

□ Activer/désactiver les triggers d'une table

- ALTER TABLE nomTable ENABLE | DISABLE ALL TRIGGERS;

□ Supprimer

- DROP TRIGGER nom;