

BDA / TP04 Oracle

Table des matières

| | |
|--|---|
| 1. Introduction..... | 2 |
| 2. Manipulation de la base de données..... | 2 |
| 2.1. Déclencheur sur une vue avec jointures..... | 2 |
| 2.2. Déclencheur sur une vue avec groupements..... | 5 |
| 3. Conclusion..... | 7 |

1. Introduction

Ce TP porte sur le développement de « triggers » attachés au niveau des vues.

Il est en effet possible de placer **des déclencheurs de type « instead of » sur des vues** (au lieu de les associer aux tables exclusivement). Ces déclencheurs ont pour caractéristique d'annuler l'évènement déclencheur survenant dans la vue. Cela rend la vue active, c'est-à-dire, rend possible les mises à jour au travers de vues qui ne le supportent pas (vues multitables) par déclenchement du déclencheur qui remplace l'instruction de mise à jour sur la vue par des instructions sur les tables.

2. Manipulation de la base de données

2.1. Déclencheur sur une vue avec jointures

(1) Créer la vue V_FLEUVE_EUROPE (NomFleuve, Longueur, Embouchure, NomPays) contenant tous les fleuves avec les pays d'Europe traversés.

```
CREATE OR REPLACE VIEW V_FLEUVE_EUROPE (NomFleuve, Longueur, Embouchure, NomPays)
AS SELECT nom, longueur, embouchure, nomp FROM Fleuve F, Traverser T WHERE F.nf = T.nf
AND nomp IN (SELECT nom FROM Pays WHERE nomc = 'Europe');
```

View V_FLEUVE_EUROPE créé(e).

(2) Afficher le contenu de cette vue.

```
SELECT * FROM V_FLEUVE_EUROPE;
```

| | NOMFLEUVE | LONGUEUR | EMBOUCHURE | NOMPAYS |
|-----|--------------|----------|------------------|----------|
| 1 | Le Maroni | 612 | océan Atlantique | France |
| 2 | L'Oyapock | 403 | océan Atlantique | France |
| 3 | Le Sinnamary | 262 | océan Atlantique | France |
| 148 | l'Yser | 78 | mer du Nord | France |
| 149 | l'Yser | 78 | mer du Nord | Belgique |

(3) Tenter d'insérer un nouveau Fleuve. Que constatez-vous ? Expliquer.

```
INSERT INTO V_FLEUVE_EUROPE VALUES('fleuve_iut', 22, 'fleuve du sud', 'France');
```

```
Rapport d'erreur -  
Erreur SQL : ORA-01779: impossible de modifier une colonne correspondant à une table non protégée par clé  
01779. 00000 - "cannot modify a column which maps to a non key-preserved table"  
*Cause:      An attempt was made to insert or update columns of a join view which  
              map to a non-key-preserved table.  
*Action:     Modify the underlying base tables directly.
```

On ne peut insérer un nouveau fleuve dans la vue car celui-ci doit déjà appartenir à la table Fleuve sans oublier qu'il doit avoir les caractéristiques nécessaires pour rentrer dans la vue.

(4) Ajouter un déclencheur sur la vue V_FLEUVE_EUROPE afin de la rendre active en insertion.

```
CREATE OR REPLACE TRIGGER t_v_fleuve_europe  
INSTEAD OF INSERT  
ON V_FLEUVE_EUROPE  
FOR EACH ROW  
  
DECLARE  
    n Number := 0;  
  
BEGIN  
    SELECT COUNT(*) INTO n FROM Pays P WHERE :NEW.nompays = P.nom AND nomc = 'Europe';  
    IF(n>0) THEN  
        INSERT INTO Fleuve VALUES(Seq_Fleuve.NEXTVAL, :NEW.nomfleuve, :NEW.longueur, :NEW.embouchure);  
        INSERT INTO Traverser VALUES(Seq_Fleuve.CURRVAL, :NEW.nompays);  
    ELSE  
        RAISE_APPLICATION_ERROR(-20000, ' le pays associé au fleuve doit etre en Europe');  
    END IF;  
  
END;  
/  
  
Elément Trigger T_V_FLEUVE_EUROPE compilé
```

(5) Tenter d'insérer à nouveau le Fleuve. Que constatez-vous ? Expliquer.

```
INSERT INTO V_FLEUVE_EUROPE VALUES('fleuve_iut', 22, 'fleuve du sud', 'France');
```

1 ligne inséré.

Le trigger a donc permis l'insertion de ce fleuve car le fleuve fut inséré dans les tables le concernant.

(6) Vérifier le contenu de la vue V_FLEUVE_EUROPE, et des tables FLEUVE et TRAVERSER.

```
SELECT * FROM V_FLEUVE_EUROPE WHERE nomfleuve='fleuve_iut';
```

| | NOMFLEUVE | LONGUEUR | EMBOUCHURE | NOMPAYS |
|---|------------|----------|---------------|---------|
| 1 | fleuve_iut | 22 | fleuve du sud | France |

```
SELECT * FROM Fleuve WHERE nom = 'fleuve_iut';
```

| | NF | NOM | LONGUEUR | EMBOUCHURE |
|---|-----|------------|----------|---------------|
| 1 | 281 | fleuve_iut | 22 | fleuve du sud |

```
SELECT * FROM Traverser WHERE nf = 281;
```

| | NF | NOMP |
|---|-----|--------|
| 1 | 281 | France |

(7) Modifier le déclencheur afin de rendre la vue V_JOUEUR_FRA active en insertion mais également en suppression.

```
CREATE OR REPLACE TRIGGER t_v_fleuve_europe
INSTEAD OF INSERT OR DELETE
ON V_FLEUVE_EUROPE
FOR EACH ROW

DECLARE
    n Number := 0;

BEGIN
    IF(INSERTING) THEN
        SELECT COUNT(*) INTO n FROM Pays P WHERE :NEW.nompays = P.nom AND nomc = 'Europe';
        IF (n>0) THEN
            INSERT INTO Fleuve VALUES(Seq_Fleuve.NEXTVAL, :NEW.nomfleuve, :NEW.longueur, :NEW.embouchure);
            INSERT INTO Traverser VALUES(Seq_Fleuve.CURRVAL, :NEW.nompays);
        ELSE
            RAISE_APPLICATION_ERROR(-20000, ' le pays associé au fleuve doit etre en Europe');
        END IF;
    ELSIF(DELETING) THEN
        DELETE FROM Traverser WHERE nf IN (SELECT nf FROM Fleuve WHERE nom = :OLD.nomfleuve);
        DELETE FROM Fleuve WHERE nom = :OLD.nomfleuve;
    END IF;
END;
/
```

Elément Trigger T_V_FLEUVE_EUROPE compilé

(8) Testez votre déclencheur

```
DELETE FROM V_FLEUVE_EUROPE WHERE nomfleuve = 'fleuve_iut';
```

1 ligne supprimé.

```
SELECT COUNT(*) FROM V_FLEUVE_EUROPE WHERE nomfleuve = 'fleuve_iut';
```

```
SELECT COUNT(*) FROM Fleuve WHERE nom = 'fleuve_iut';
```



| COUNT(*) | |
|----------|---|
| 1 | 0 |

2.2. Déclencheur sur une vue avec groupements

(1) Créer une vue V_PAYS_GEO (NomPays, CapitalePays, PopulationPays, ISO3Pays, ISO2Pays, NBFleuves) contenant les pays et le nombre de fleuves par pays.

```
CREATE OR REPLACE VIEW V_PAYS_GEO (NomPays, CapitalePays, PopulationPays, ISO3Pays, ISO2Pays, NBFleuves)
AS SELECT P.nom, capitale, population, iso3, iso2, COUNT(nf)AS "nbf" FROM Pays P, Traverser T WHERE T.nomp=P.nom
GROUP BY P.nom, capitale, population, iso3, iso2;
```

(2) Afficher le contenu de cette vue.

```
SELECT * FROM V_PAYS_GEO;
```

| NOMPAYS | CAPITALEPAYS | POPULATIONPAYS | ISO3PAYS | ISO2PAYS | NBFLEUVES |
|-----------------------|--------------|----------------|----------|----------|-----------|
| 1 Thaïlande | Bangkok | 68977400 | THA | TH | 3 |
| 2 Colombie | Bogota | 49084841 | COL | CO | 5 |
| 3 République du Congo | Brazzaville | 5293070 | COG | CG | 2 |
| 119 Uruguay | Montevideo | 3387605 | URY | UY | 1 |
| 120 Suède | Stockholm | 10313447 | SWE | SE | 3 |
| 121 Laos | Vientiane | 7447396 | LAO | LA | 1 |

(3) Tenter d'insérer un nouveau pays au travers de cette vue. Que constatez-vous ? Expliquer.

```
INSERT INTO V_PAYS_GEO VALUES('IUT', 'Blagnac', 2000, 'IUT', 'UT', 1);
```

Rapport d'erreur -

Erreur SQL : ORA-01779: impossible de modifier une colonne correspondant à une table non protégée par clé
01779. 00000 - "cannot modify a column which maps to a non key-preserved table"

*Cause: An attempt was made to insert or update columns of a join view which map to a non-key-preserved table.

*Action: Modify the underlying base tables directly.

On ne peut pas insérer un nouveau pays au travers de cette vue car celui-ci doit déjà appartenir à la table Pays.

(4) Ajouter un déclencheur sur la vue V_PAYS_GEO afin de la rendre active en insertion, suppression et modification.

```
CREATE OR REPLACE TRIGGER t_v_pays_geo
INSTEAD OF INSERT OR DELETE OR UPDATE
ON V_PAYS_GEO
FOR EACH ROW
BEGIN
IF (INSERTING) THEN
INSERT INTO Pays VALUES(:NEW.NomPays, :NEW.CapitalePays, NULL, :NEW.PopulationPays, :NEW.ISO3Pays, :NEW.ISO2Pays, NULL);
END IF;
IF (DELETING) THEN
DELETE FROM Pays WHERE nom = :OLD.NomPays;
END IF;
IF (UPDATING) THEN
UPDATE Pays
SET nom = :NEW.NomPays, capitale = :NEW.CapitalePays, population = :NEW.PopulationPays,
iso3 = :NEW.ISO3Pays, iso2 = :NEW.ISO2Pays
WHERE nom = :OLD.NomPays;
END IF;
END;
```

(5) Tenter d'insérer à nouveau le pays au travers de cette vue. Que constatez-vous ? Expliquer.

```
INSERT INTO V_PAYS_GEO VALUES('IUT', 'Blagnac', 2000, 'IUT', 'UT', 1);
```

1 ligne inséré.

```
SELECT * FROM V_PAYS_GEO WHERE NomPays = 'IUT';
```

| NOMPAYS | CAPITALE... | POPULATI... | ISO3PAYS | ISO2PAYS | NBFLEUVES |
|---------|-------------|-------------|----------|----------|-----------|
| 1 IUT | Blagnac | (null) | 2000 | IUT | UT |

```
SELECT * FROM PAYS WHERE Nom = 'IUT';
```

| NOM | CAPITALE | SUPERFICIE | POPULATION | ISO3 | ISO2 | NOMC |
|-------|----------|------------|------------|------|------|--------|
| 1 IUT | Blagnac | (null) | 2000 | IUT | UT | (null) |

Le déclencheur nous a permis d'insérer ce nouveau pays dans la table pays mais pas dans la vue. Cela s'explique par le fait que le nombre de fleuve n'existe pas réellement et donc donner une valeur fausse '1' ne fera qu'annuler l'insertion du pays dans la vue, puisque le nouveau ne possède aucun fleuve.

3. Conclusion

Ce TP04 a pour but de mettre en application ce que nous avons déjà vu au cours des TD/TP et cours précédents , c'est à dire la manipulation de base de données grâce au langage SQL (Structured Query Language) et le fonctionnement de triggers en PL/SQL.

Durant ce TP, on a pu créer différents déclencheurs : un déclencheur sur des vue avec jointures et un déclencheur sur une vue avec groupements. Ils nous ont permis une insertion/modification/suppression d'une ligne dans une table spécifique sur une vue.

Ce TP04 nous a permis d'approfondir nos connaissances sur les déclencheurs ainsi que leurs utilisations.