

BDA / TP03 Oracle

Table des matières

1. Introduction.....	2
2. Manipulation de la base de données.....	2
2.1. Déclencheur de contrainte FOR EACH ROW.....	2
2.2. Déclencheur de contrainte STATEMENT.....	3
2.3. Implantation des multiplicités 1..*.....	4
3. Conclusion.....	5

1. Introduction

Ce TP sert à étudier les déclencheurs (« triggers ») de contraintes d'intégrités « complexes », c'est-à-dire des contraintes ne pouvant pas s'exprimer par les C.I. (primary key, unique key, foreign key, check et not null).

2. Manipulation de la base de données

2.1. Déclencheur de contrainte FOR EACH ROW

(1) Créez un déclencheur permettant de vérifier que la superficie d'un pays ne peut pas être supérieure à celle du continent dans lequel il est situé.

```
CREATE OR REPLACE TRIGGER t_iu_pays_superficie
BEFORE INSERT OR UPDATE OF nomc, superficie
ON Pays
FOR EACH ROW

DECLARE
sup number:=0;

BEGIN
SELECT superficie INTO sup FROM Continent WHERE nom = :NEW.nomc;
IF(:NEW.superficie > sup) THEN
    RAISE_APPLICATION_ERROR(-20000,'La superficie du pays est superieur à celle de son continent');
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('La superficie du continent est inconnue');
END;
```

(2) Testez votre déclencheur

```
INSERT INTO Pays Values('IUT de Blagnac','Dept Info', 9938022201,NULL,'IUT','UT','Europe');
```

Erreur commençant à la ligne: 21 de la commande -

```
INSERT INTO Pays Values('IUT de Blagnac','Dept Info', 9938022201,NULL,'IUT','UT','Europe')
```

Rapport d'erreur -

ORA-20000: La superficie du pays est superieur à celle de son continent

ORA-06512: à "ORACLE2001.T_IU_PAYS_SUPERFICIE", ligne 7

ORA-04088: erreur lors d'exécution du déclencheur 'ORACLE2001.T_IU_PAYS_SUPERFICIE'

2.2. Déclencheur de contrainte STATEMENT

(1) Créez un déclencheur permettant de vérifier que l'altitude d'une montagne ne peut pas être supérieure à celle de l'Everest.

```
CREATE OR REPLACE TRIGGER t_iu_montagne_altitude
AFTER INSERT OR UPDATE OF altitude
ON Montagne

DECLARE
n number:=0;

BEGIN
SELECT COUNT(*) INTO n FROM Montagne M1, Montagne M2 WHERE M1.nom = 'Everest'
AND M2.nom != 'Everest' AND M1.altitude < M2.altitude;
IF(n > 0) THEN
    RAISE_APPLICATION_ERROR(-20000,'l'altitude d'une montagne ne peut pas être supérieure à celle de l'Everest');
END IF;

END;
/
```

(2) Testez votre déclencheur.

```
INSERT INTO Montagne Values (SEQ_MONTAGNE.nextval, 'IUT', 8849, NULL);
```

Rapport d'erreur -

ORA-20000: l'altitude d'une montagne ne peut pas être supérieure à celle de l'Everest

ORA-06512: à "ORACLE2001.T_IU_MONTAGNE_ALTITUDE", ligne 7

ORA-04088: erreur lors d'exécution du déclencheur 'ORACLE2001.T_IU_MONTAGNE_ALTITUDE'

2.3. Implantation des multiplicités 1..*

(1) Proposez un déclencheur permettant de vérifier lors d'une insertion qu'un fleuve traverse au moins un pays (conformément à la multiplicité 1..* de l'association Traverser).

```
CREATE OR REPLACE TRIGGER t_i_fleuve_traverser
BEFORE INSERT ON Fleuve
FOR EACH ROW

DECLARE
n number:=0;

BEGIN
SELECT COUNT(*) INTO n FROM Traverser T, Pays P WHERE :NEW.nf = T.nf AND T.nomp = P.nom;
IF(n <= 0) THEN
    RAISE_APPLICATION_ERROR(-20000,'un fleuve traverse au moins un pays');
END IF;

END;
/
```

(2) Testez votre déclencheur en insérant un nouveau fleuve qui traverse la France.

```
INSERT INTO Fleuve VALUES(Seq_Fleuve.NEXTVAL, 'Quillimadec', 22, 'Manche');
```

```
Rapport d'erreur -
ORA-20000: un fleuve traverse au moins un pays
ORA-06512: à "ORACLE2001.T_I_FLEUVE_TRAVERSER", ligne 7
ORA-04088: erreur lors d'exécution du déclencheur 'ORACLE2001.T_I_FLEUVE_TRAVERSER'
```

(3) Proposez une solution afin d'insérer ce fleuve.

```
ALTER TRIGGER t_i_fleuve_traverser DISABLE;
```

```
Trigger T_I_FLEUVE_TRAVERSER modifié(e).
```

```
INSERT INTO Fleuve VALUES(Seq_Fleuve.NEXTVAL, 'Quillimadec', 22, 'Manche');
```

```
1 ligne inséré.
```

3. Conclusion

Ce TP03 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 de contrainte (« FOR EACH ROW » et « STATEMENT »). Ils nous ont permis de vérifier les caractéristiques d'un attribut lors d'une insertion/modification d'une ligne dans une table spécifique. On a aussi pu désactiver un déclencheur pour pouvoir insérer une ligne dans une table.

Ce TP03 nous a permis d'approfondir nos connaissances sur les déclencheurs ainsi que leurs utilisations.