COMPTE RENDU BDA TP03

Déclencheurs de mise à jour

Sommaire

Table des matières

1-Introduction	3
1) Déclencheur pour maintenir NBJL dans Location	∠
2) Déclencheur pour maintenir NBL et NBJ dans Client	€

Page: 2

1-Introduction

Le sujet de ce TP concerne l'apprentissage des déclencheurs.

L'objectif est de créer des déclencheurs capables de mettre à jour notre base de données quand on insert ou delete dans LOUER.

Pour ce faire nous avons utilisé un logiciel : Oracle SQL Developer.

Page: 3 DELIOT MAXENCE

1) Déclencheur pour maintenir NBJL dans Location

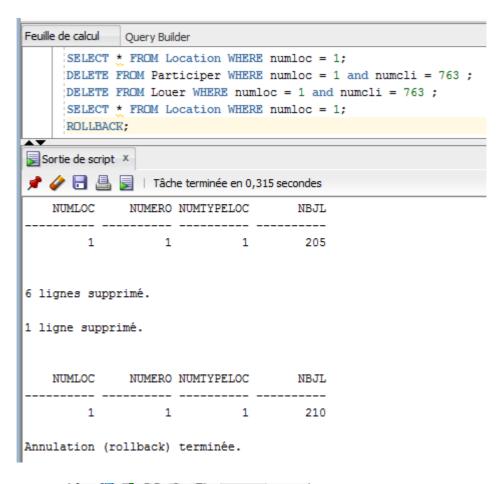
1) Créez un déclencheur permettant de maintenir à jour cet attribut en fonction des manipulations qui surviennent sur la relation LOUER.

```
Votre déclencheur sera conforme aux spécifications suivantes :
CREATE OR REPLACE TRIGGER t_iud_louer
AFTER INSERT OR DELETE OR UPDATE
ON Louer
FOR EACH ROW
BEGIN
IF (INSERTING OR UPDATING) THEN
à compléter
END IF;
IF (DELETING OR UPDATING) THEN
à compléter
END IF;
END; /
CREATE OR REPLACE TRIGGER t iud louer
AFTER INSERT OR DELETE OR UPDATE
ON Louer
FOR EACH ROW
BEGIN
 IF (INSERTING OR UPDATING) THEN
    Update Location
    set NBJL = NBJL -(:NEW.DATEFIN -:New.DateDEb) where
location.numLoc = :NEW.NumLoc;
 END IF;
 IF (DELETING OR UPDATING) THEN
```

```
Update Location
  set NBJL = NBJL +(:OLD.DATEFIN -:OLD.DateDEb) where
location.numLoc = :NEW.NumLoc;
END IF;
END;
/
```

2) Testez votre déclencheur afin de vérifier son bon fonctionnement.

```
Par exemple
SELECT * FROM Location WHERE numloc = 1;
DELETE FROM Participer WHERE numloc = 1 and numcli = 551;
DELETE FROM Louer WHERE numloc = 1 and numcli = 551;
SELECT * FROM Location WHERE numloc = 1;
ROLLBACK;
```



2) Déclencheur pour maintenir NBL et NBJ dans Client

Au TP1, nous avons ajouté dans la table CLIENT, NBL ENTIER(5) représentant le nombre de locations (dans LOUER) et NBJ ENTIER(5) correspondant au nombre de jours loués (dans LOUER)

1) Complétez le déclencheur précédant afin de maintenir à jour cet attribut en fonction des manipulations qui surviennent sur la relation LOUER.

```
create or replace TRIGGER t_iud_louer
AFTER INSERT OR DELETE OR UPDATE
ON Louer
FOR EACH ROW
BEGIN
 IF (INSERTING OR UPDATING) THEN
    Update Location
    set NBJL = NBJL - (:NEW.DATEFIN -:New.DateDEb) where
location.numLoc = :NEW.NumLoc;
    Update Client
set NBL = NBL + 1,
    NBJ = NBJ + (:NEW.DATEFIN-:NEW.DATEDEB)WHERE
Client.numCli=:NEW.NUMCLI;
 END IF;
 IF (DELETING OR UPDATING) THEN
   Update Location
   set NBJL = NBJL +(:OLD.DATEFIN -:OLD.DateDEb) where
location.numLoc = :OLD.NumLoc;
   Update Client
   set NBL = NBL - 1,
    NBJ = NBJ - (:OLD.DATEFIN-:OLD.DATEDEB)WHERE
Client.numCli=:OLD.NUMCLI;
 END IF;
END;
```

2) Testez votre déclencheur afin de vérifier son bon fonctionnement

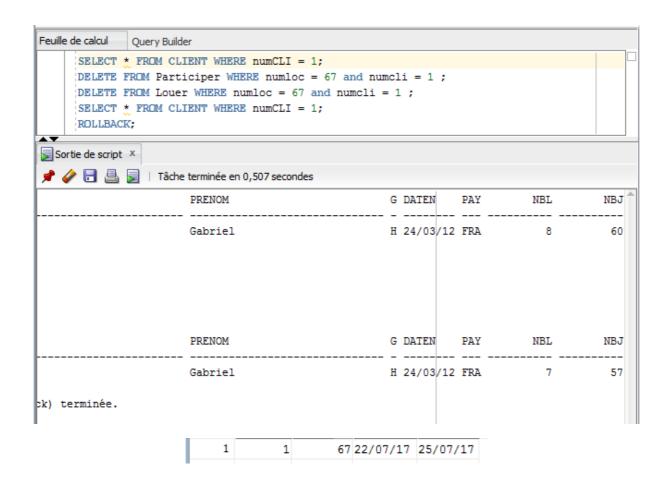
```
SELECT * FROM CLIENT WHERE numCLI = 1;

DELETE FROM Participer WHERE numloc = 67 and numcli = 1;

DELETE FROM Louer WHERE numloc = 67 and numcli = 1;

SELECT * FROM CLIENT WHERE numCLI = 1;

ROLLBACK;
```



3-Conclusion:

A l'issue de ce TP, nous savons créer des **déclencheurs qui mettent à jour automatiquement notre base de données** après une action spécifiée.

Page: 8 DELIOT MAXENCE