

# COMPTE RENDU BDA

## TP04

### Déclencheurs de contraintes

# Sommaire

## Table des matières

1-Introduction.....	3
1) Déclencheur de contrainte sur Participer.....	4
2) Déclencheur de contrainte d'âge sur Louer.....	7
3) Déclencheur de contrainte d'âge sur Louer.....	10

## *1-Introduction*

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

L'objectif est de créer des **déclencheurs capables d'empêcher d'insert dans LOUER** si certaines conditions ne sont pas respectées.

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

## 2-Les requêtes

### 1) Déclencheur de contrainte sur Participer

- 1) Créez un déclencheur garantissant que la capacité maximale d'une location n'est pas dépassée lors de l'ajout de participants à une réservation (LOUER). Votre déclencheur sera conforme aux spécifications suivantes :

```
CREATE OR REPLACE TRIGGER t_iud_participer
```

```
BEFORE INSERT OR UPDATE OF à compléter
```

```
ON Participer
```

```
FOR EACH ROW
```

```
BEGIN
```

```
à compléter
```

```
END;
```

```
/
```

```
create or replace TRIGGER t_iud_participer
```

```
BEFORE INSERT OR UPDATE OF NUMLOC
```

```
ON Participer
```

```
DECLARE
```

```
CPT NUMBER;
```

```
BEGIN
```

```
select count(count(Participer.NumCLiP)) into CPT
```

```
from TYPE_LOCATION TP, LOCATION L, Participer, LOUER LO
```

```
where Participer.NumLoc = LO.NUMLOC
```

```
and Participer.dateDeb = LO.DateDeb
```

```
and Participer.NUMCLI=LO.NUMCLI
```

```
and LO.NUMLOC=L.NUMLOC
```

```
and L.NUMTYPELOC = TP.NUMTYPELOC
```

```
group by Participer.dateDeb, Participer.NumLoc,
```

```
TP.Capacite,PARTICIPER.NUMCLI
```

```
Having count(*)+1>TP.CAPACITE;
```

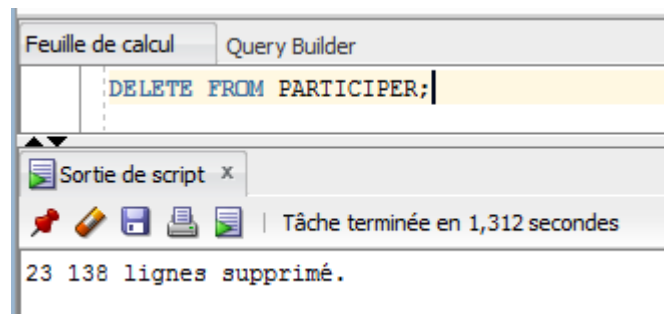
```
if CPT >0 then
```

```
    raise_Application_Error(-20001,'Déplacement de capacité');
```

```
End if;
```

```
END;
```

## 2) Supprimer les locations dans LOUER des personnes mineures.



## 3) Testez votre déclencheur afin de vérifier son bon fonctionnement en effectuant des insertions dans PARTICIPER.

1	2	102	01/08/17	10/08/17		
1	1 BUNGALOW	Caribe Club	52	4	80,85	40,43

```
insert into PARTICIPER VALUES(1,2,102,'01/08/2017');
```

```
insert into PARTICIPER VALUES(11,2,102,'01/08/2017');
```

```
insert into PARTICIPER VALUES(12,2,102,'01/08/2017');
```

```
insert into PARTICIPER VALUES(13,2,102,'01/08/2017');
```

```
insert into PARTICIPER VALUES(14,2,102,'01/08/2017');
```

Feuille de calcul Query Builder 0,62199998 secondes

```

insert into PARTICIPER VALUES(1,2,102,'01/08/2017');
insert into PARTICIPER VALUES(11,2,102,'01/08/2017');
insert into PARTICIPER VALUES(12,2,102,'01/08/2017');
insert into PARTICIPER VALUES(13,2,102,'01/08/2017');
insert into PARTICIPER VALUES(14,2,102,'01/08/2017');

```

Sortie de script x

Tâche terminée en 0,622 secondes

1 ligne inséré.

1 ligne inséré.

1 ligne inséré.

1 ligne inséré.

Erreur commençant à la ligne: 5 de la commande -  
insert into PARTICIPER VALUES(14,2,102,'01/08/2017')  
Rapport d'erreur  
Erreur SQL : ORA-20001: Dépacement de capacité  
ORA-06512: à "IUTB2039.I\_UD\_PARTICIPER", ligne 12  
ORA-04088: erreur lors d'exécution du déclencheur 'IUTB2039.I\_UD\_PARTICIPER'

Feuille de calcul Query Builder

```

select * from PARTICIPER where NUMCLI=2 and NUMLOC=102;

```

Sortie de script x

Tâche terminée en 0,09 secondes

NUMCLIP	NUMCLI	NUMLOC	DATEDEB
1	2	102	01/08/17
12	2	102	01/08/17
13	2	102	01/08/17
11	2	102	01/08/17

L'erreur s'exécute bien quand on dépasse les 4 participants.

## 2) Déclencheur de contrainte d'âge sur Louer

On souhaite interdire qu'un client mineur puisse effectuer une location (LOUER). Pour calculer l'âge des clients vous pouvez tester la requête suivante :

```
SELECT NumCli, Nom, Prenom,  
       TRUNC(MONTHS_BETWEEN(SYSDATE,DateN)/12) AS age  
FROM Client;
```

**1) Créez un déclencheur permettant de garantir cette contrainte complexe. Votre déclencheur sera conforme aux spécifications suivantes :**

```
CREATE OR REPLACE TRIGGER t_iud_louer_age  
BEFORE à compléter  
ON Louer  
FOR EACH ROW  
BEGIN  
à compléter  
END;  
/
```

```

create or replace TRIGGER t_iud_louer_age
BEFORE INSERT
ON Louer
FOR EACH ROW
DECLARE
ageClip Number;
BEGIN
    SELECT TRUNC(MONTHS_BETWEEN(SYSDATE,DateN)/12) AS age into ageClip
    FROM Client C
    where :NEW.NUMCLI= C.NUMCLI;
    if ageClip < 18 then
        raise_Application_Error(-20001,'Vous etes trop jeune');
    END IF;
END;

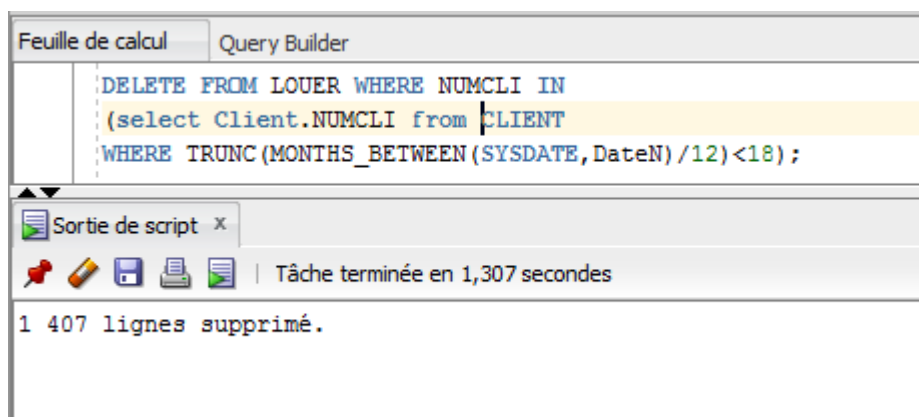
```

## 2) Supprimer les locations dans LOUER des personnes mineures.

```

DELETE FROM LOUER WHERE NUMCLI IN
(select Client.NUMCLI from CLIENT
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE,DateN)/12)<18);

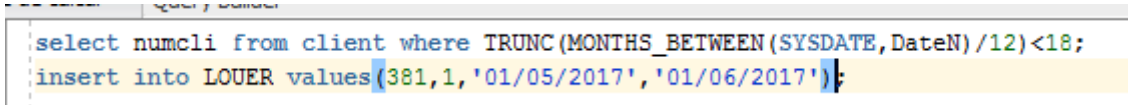
```





### 3) Testez votre déclencheur afin de vérifier son bon fonctionnement.

```
select numcli from client where  
TRUNC (MONTHS_BETWEEN (SYSDATE, DateN) /12) <18;  
  
insert into LOUER values (381,1,'01/05/2017','01/06/2017');
```



```
select numcli from client where TRUNC (MONTHS_BETWEEN (SYSDATE, DateN) /12) <18;  
insert into LOUER values (381,1,'01/05/2017','01/06/2017');
```

```
381  
  
NUMCLI  
-----  
382  
385  
  
200 lignes sélectionnées
```

```
Erreur commençant à la ligne: 2 de la commande -  
insert into LOUER values (381,1,'01/05/2017','01/06/2017')  
Rapport d'erreur -  
Erreur SQL : ORA-20001: Vous etes trop jeune  
ORA-06512: à "IUTB2039.T_IUD_LOUER_AGE", ligne 8  
ORA-04088: erreur lors d'exécution du déclencheur 'IUTB2039.T_IUD_LOUER_AGE'
```

### 3) Déclencheur de contrainte d'âge sur Louer

On souhaite interdire qu'un client puisse effectuer des locations cumulées (LOUER) de plus de 180 jours.

#### **1) Créez un déclencheur permettant de garantir cette contrainte complexe.**

```
CREATE OR REPLACE TRIGGER t_iud_louer_periode
```

```
BEFORE à compléter
```

```
ON Louer
```

```
FOR EACH ROW
```

```
BEGIN
```

```
à compléter
```

```
END;
```

```
/
```

```
create or replace trigger t_iud_louer_periode
```

```
AFTER INSERT OR UPDATE
```

```
ON LOUER
```

```
DECLARE
```

```
NBC NUMBER;
```

```
BEGIN
```

```
select count(NUMCLI) into NBC
```

```
FROM CLIENT C
```

```
WHERE C.NBJ>180 ;
```

```
if NBC > 0 then
```

```
    raise_Application_Error(-20001,'Trop de jour cumulées');
```

```
End if;
```

```
END;
```

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

```
delete from PARTICIPER;
```

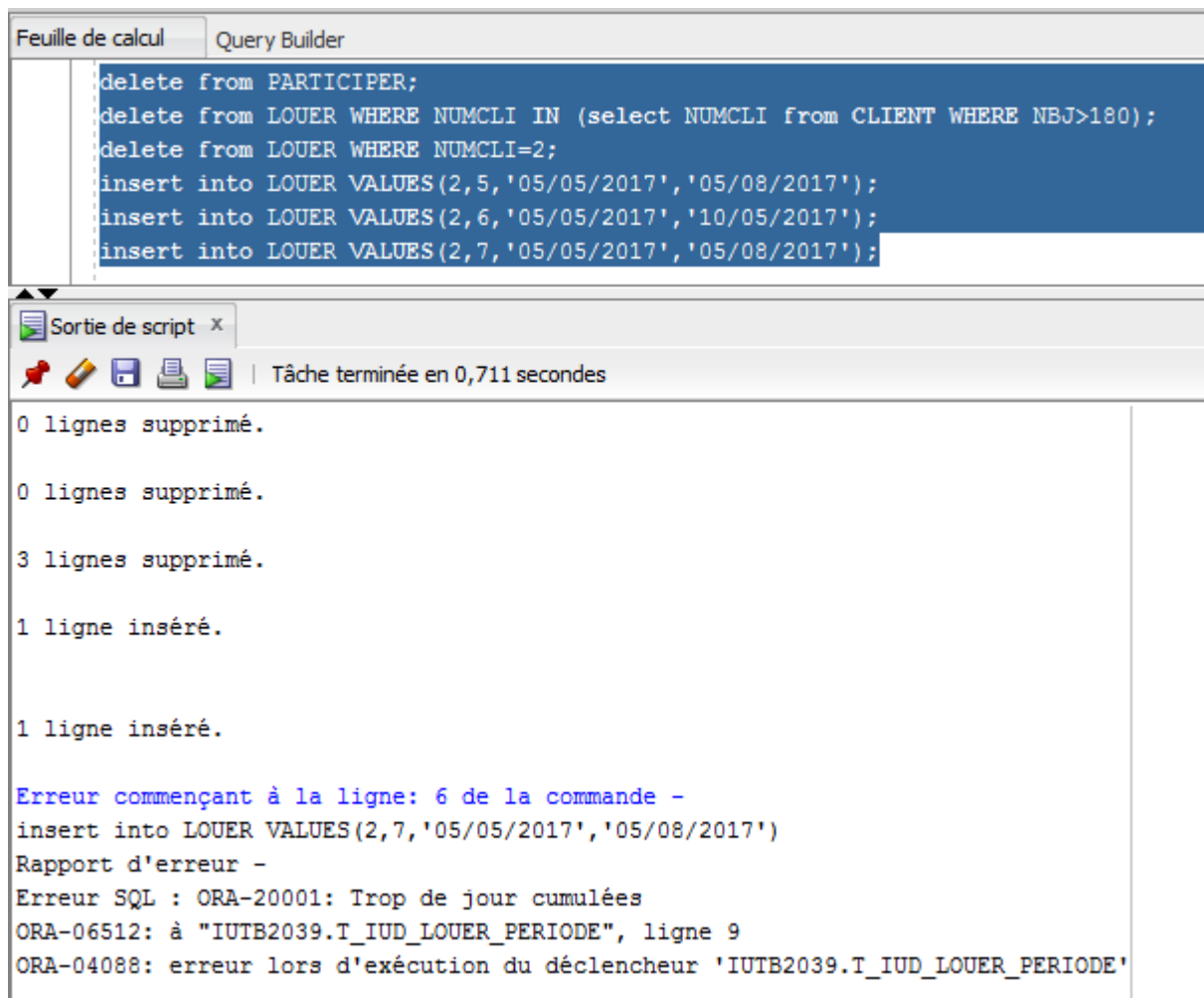
```
delete from LOUER WHERE NUMCLI IN (select NUMCLI from CLIENT WHERE NBJ>180);
```

```
delete from LOUER WHERE NUMCLI=2;
```

```
insert into LOUER VALUES(2,5,'05/05/2017','05/08/2017');
```

```
insert into LOUER VALUES(2,6,'05/05/2017','10/05/2017');
```

```
insert into LOUER VALUES(2,7,'05/05/2017','05/08/2017');
```



The screenshot shows a SQL query builder window with a 'Query Builder' tab. The SQL script is as follows:

```
delete from PARTICIPER;
delete from LOUER WHERE NUMCLI IN (select NUMCLI from CLIENT WHERE NBJ>180);
delete from LOUER WHERE NUMCLI=2;
insert into LOUER VALUES(2,5,'05/05/2017','05/08/2017');
insert into LOUER VALUES(2,6,'05/05/2017','10/05/2017');
insert into LOUER VALUES(2,7,'05/05/2017','05/08/2017');
```

Below the query, a status bar indicates 'Tâche terminée en 0,711 secondes'. The execution results are displayed in a text area:

```
0 lignes supprimé.
0 lignes supprimé.
3 lignes supprimé.
1 ligne inséré.
1 ligne inséré.

Erreur commençant à la ligne: 6 de la commande -
insert into LOUER VALUES(2,7,'05/05/2017','05/08/2017')
Rapport d'erreur -
Erreur SQL : ORA-20001: Trop de jour cumulées
ORA-06512: à "IUTB2039.T_IUD_LOUER_PERIODE", ligne 9
ORA-04088: erreur lors d'exécution du déclencheur 'IUTB2039.T_IUD_LOUER_PERIODE'
```

### **3) Expliquez pourquoi il s'agit d'un déclencheur de table (« Statement trigger »).**

On note que la commande FOR EACH ROW ne se trouve pas dans la syntaxe de ce déclencheur, c'est donc un déclencheur Statement. Son action s'appliquera qu'une seule fois sur toute la table voulue.

Par exemple, nous avons une table à 1000 colonnes, le Statement trigger s'activera qu'une fois même si aucune colonne ne sera mise à jour. Contrairement au déclencheur de ligne qui lui pourra s'activer une fois pour chaque colonne à mettre à jour, soit 1000 fois si nécessaire.

### ***3-Conclusion :***

A l'issue de ce TP, nous savons créer des **déclencheurs de contraintes** qui **empêchent l'ajout** sous certaines conditions.