

Compte-rendu du TP n°4

Le schéma de tables utilisé, est le même que celui utilisé dans le TP3.

Requêtes

1) Définissez un déclencheur qui modifie les n-uplets dans la table MALADE de la façon suivante à chaque nouvelle insertion :

- (a) L'identifiant sera égal au plus grand identifiant présent dans la table + 1.
- (b) Le nom et le prénom du malade seront écrits en lettres minuscules.

```
SQL]
CREATE OR REPLACE TRIGGER TRIGGER_EX1
  AFTER INSERT ON CARON_MALADE
  FOR EACH ROW
BEGIN
  UPDATE CARON_MALADE
  SET n_malade = (SELECT max(n_malade) FROM CARON_MALADE) + 1,
      nom_malade = LOWER(nom_malade), prenom_malade = LOWER(prenom_malade);
END;
```

TESTS]

1) insert into caron_malade VALUES(5410,'Henri','Jacques','76 rue de la Paix, Paris','0145897622','0612121212') ;

2) insert into caron_malade
VALUES(4000,'Sam','Flynn','none','0145889966','0612121213') ;

R1 SELECT N_MALADE, NOM_MALADE, PRENOM_MALADE FROM CARON_MALADE;

N_MALADE	NOM_MALADE	PRENOM_MALADE
5678	durand	pierre
5430	marchand	fabrice
5220	bonnet	monique
5970	lambert	jacques
5780	durand	jeanne
5632	faure	aline
5320	garcia	paul
5971	henri	jacques
5972	sam	flynn

2) Ajoutez un déclencheur qui modifie les consultations insérées lorsque l'attribut prescription et l'attribut examen sont NULL : l'attribut examen aura alors comme valeur "Visite de routine".

SQL1

```
CREATE OR REPLACE TRIGGER TRIGGER_EX2
  BEFORE INSERT ON CARON_CONSULTATION
  FOR EACH ROW
BEGIN
  IF :NEW.prescription IS NULL AND :NEW.examen IS NULL
  THEN :NEW.examen := 'Visite de routine';
  END IF;
END;
/
```

TESTS1

1) insert into caron_consultation(n_med,n_malade,date_consult,h_consult)
values(146, 5780, '19-11-2011','09');

2) insert into caron_consultation(n_med,n_malade,date_consult,h_consult)
values(146, 5780, '20-11-2011','11');

RJ] SELECT N_MED, N_MALADE, EXAMEN FROM CARON_CONSULTATION;

N_MED	N_MALADE	EXAMEN
... (null)
146	5780	Visite de routine
146	5780	Visite de routine

3) Créez une table CARON_ARCHIVE_CONSULTATION qui contiendra les consultations passées. Chaque consultation supprimée dont la date est antérieure à la date actuelle sera archivée dans la table CARON_ARCHIVE_CONSULTATION.

SQL] On copie la table sans éléments (en utilisant le fait que l'heure est toujours définie pour toutes les lignes).

```
CREATE TABLE CARON_ARCHIVE_CONSULTATION
  AS (SELECT * FROM CARON_CONSULTATION WHERE h_consult IS NULL);
```

Puis, on crée le trigger :

```
CREATE TABLE CARON_ARCHIVE_CONSULTATION
  AS (SELECT * FROM CARON_CONSULTATION WHERE h_consult IS NULL);
```

```
CREATE OR REPLACE TRIGGER TRIGGER_EX3
  BEFORE DELETE ON CARON_CONSULTATION
  FOR EACH ROW
BEGIN
  IF :NEW.date_consult < current_date THEN
    INSERT INTO CARON_ARCHIVE_CONSULTATION(n_med, n_malade, date_consult,
h_consult, prescription, examen)
    VALUES(:NEW.n_med, :NEW.n_malade, :NEW.date_consult, :NEW.h_consult,
:NEW.prescription, :NEW.examen);
  END IF;
```

```
END;
/

TESTS]
DELETE FROM CARON_CONSULTATION WHERE N_MED = 131;
```

>>> 5 rows deleted.

```
SELECT * FROM CARON_ARCHIVE_CONSULTATION ;
```

R]

N_MED	N_MALADE	DATE_CON	H_CONSULT	PR.	EXAMEN
131	5430	03/10/11	18		Visite de routine
131	5632	03/10/11	16		Visite de routine
131	5678	03/10/11	17		Visite de routine
131	5678	10/10/11	18		Visite de routine
131	5780	08/10/11	10		Visite de routine

4) Définissez un déclencheur qui empêche toute modification de la table ARCHIVE_CONSULTATION.

```
SQL]
CREATE OR REPLACE TRIGGER TRIGGER_EX4
BEFORE UPDATE OR DELETE ON CARON_ARCHIVE_CONSULTATION
BEGIN
    raise_application_error(-20121, 'Opération de modification interdite');
END;
/
```

TESTS] Test de suppression basée sur la table précédente.

```
DELETE FROM CARON_ARCHIVE_CONSULTATION WHERE N_MED = 131;
```

R] ERROR at line 1:

```
ORA-20121: Opération de modification interdite
ORA-06512: à "M1INF019.TRIGGER_EX4", ligne 2
ORA-04088: erreur lors d'exécution du déclencheur 'M1INF019.TRIGGER_EX4'
```

5) Définissez un déclencheur qui vérifie pour les médecins que identite_med et adr_elec_med sont formatés correctement :

a) identite_med commence par une chaîne écrite en minuscules avec la première lettre en majuscule (le prénom) et est suivie d'une chaîne écrite en majuscules (le nom).

```
SQL]
CREATE OR REPLACE TRIGGER TRIGGER_EX5A
BEFORE INSERT OR UPDATE ON CARON_MEDECIN
FOR EACH ROW
BEGIN
    IF NOT REGEXP_LIKE(:NEW.identite_med, '^[A-Z][a-z]* [A-Z]+$', 'c')
    THEN raise_application_error(-20122, 'Format incohérent');
    END IF;
END;
/
```

TESTS]

1) Valeur cohérente :

```
INSERT INTO CARON_MEDECIN VALUES(199,'Henri  
DECLIN','0149884461','henri.declin@free.fr','Alexandre TARAUD','CCA');
```

>>> 1 row created.

2) Valeur incohérente :

```
INSERT INTO CARON_MEDECIN VALUES(200,'HenRi  
DeclIN','0149884461','henri.declin@free.fr','Alexandre TARAUD','CCA')  
*
```

>>> ERROR at line 1:

ORA-20122: Format incohérent

ORA-06512: à "M1INF019.TRIGGER_EX5A", ligne 3

ORA-04088: erreur lors d'exécution du déclencheur 'M1INF019.TRIGGER_EX5A'

b) adr_elec_med est une adresse électronique. La chaîne de caractères contient nécessairement le caractère "@" et le caractère "." un peu plus loin.

Ce trigger utilise une expression régulière comme précédemment, se basant uniquement sur le critère imposé par le sujet pour effectuer la vérification du mail :

SQL]

```
CREATE OR REPLACE TRIGGER TRIGGER_EX5B  
BEFORE INSERT OR UPDATE ON CARON_MEDECIN  
FOR EACH ROW  
BEGIN  
  IF NOT REGEXP_LIKE(:NEW.adr_elec_med, '^.+@.+\\..+$','c')  
  THEN raise_application_error(-20122, 'Format incohérent');  
  END IF;  
END;  
/
```

TESTS]

1) Valeur cohérente :

```
INSERT INTO CARON_MEDECIN VALUES(201,'Henri  
DECLIN','0149884461','henri.declin@free.fr','Alexandre TARAUD','CCA');
```

>>> 1 row created.

2) Valeur incohérente :

```
INSERT INTO CARON_MEDECIN VALUES(202,'Henri  
DECLIN','0149884461','henri.declinfreefr','Alexandre TARAUD','CCA');
```

>>> ERROR at line 1:

ORA-20122: Format incohérent

ORA-06512: à "M1INF019.TRIGGER_EX5B", ligne 3

ORA-04088: erreur lors d'exécution du déclencheur 'M1INF019.TRIGGER_EX5A'

6) Ajoutez une colonne prime à la table MEDECIN. Écrivez un déclencheur qui lie la colonne prime de la table MEDECIN au nombre de consultations effectuées. Chaque consultation effectuée ajoute 23 à la prime du médecin qui l'effectue.

SQL]

1) Pour la modification de la table

```
ALTER TABLE CARON_MEDECIN ADD prime int;  
UPDATE CARON_MEDECIN SET prime = 0;
```

2) Pour le trigger, on utilise que toute consultation effectuée se voit alors retirée de la table des consultations. L'augmentation se fait alors suite à une suppression « cohérente » de la table

```
CREATE OR REPLACE TRIGGER TRIGGER_EX6  
AFTER DELETE ON CARON_CONSULTATION  
FOR EACH ROW  
DECLARE  
    prime_incr integer;  
BEGIN  
    prime_incr := 23;  
    IF :OLD.date_consult < current_date THEN  
        UPDATE CARON_MEDECIN  
        SET prime = prime + prime_incr  
        WHERE n_med = :OLD.n_med;  
    END IF;  
END;  
/
```

TEST] Complétion des RDVs du médecin n°131, puis vérification de sa prime.

1) delete from caron_consultation where n_med = 131;

>>> 5 rows deleted.

2) select prime from caron_medecin where n_med = 131;

>>> Valeur de la prime obtenue : 115 = 23 * 5 = 23 * (nb_lignes supprimées).

7) Il n'y a que 3 salles de consultation disponibles dans la clinique. Refusez, via un déclencheur, un ajout de plage de consultation pour un médecin s'il existe déjà 3 plages de consultation à ce créneau horaire.

SQL]

```
CREATE OR REPLACE TRIGGER TRIGGER_EX7  
BEFORE INSERT ON CARON_MED_PLAGE_CONSULT  
FOR EACH ROW  
DECLARE  
    consult_nb integer;  
BEGIN  
    SELECT count(*) INTO consult_nb FROM CARON_MED_PLAGE_CONSULT  
    WHERE jour_sem = :NEW.jour_sem  
    AND (((h_deb <= :NEW.h_deb) AND (:NEW.h_deb < h_fin))  
    OR ((h_deb < :NEW.h_fin) AND (:NEW.h_fin <= h_fin)));  
  
    IF consult_nb >= 3  
    THEN raise_application_error(-20123, 'Aucun salle n est disponible pour ce  
    créneau');  
    END IF;  
END;  
/
```

TEST]

Liste des consultation en cours :

1) select * from caron_med_plage_consult ;

N_MED	JOUR_SEM	H_DEB	H_FIN
123	mardi	8	12
123	jeudi	13	17
123	samedi	9	12
107	jeudi	13	17
131	lundi	17	19
131	samedi	10	12
146	mercredi	13	17
146	lundi	8	12
124	samedi	9	12
130	mardi	13	17
130	jeudi	13	17
130	mercredi	17	19
130	lundi	10	12
130	samedi	10	12
138	mardi	13	17
152	mercredi	13	17

On va ajouter 2 consultations supplémentaires sur le créneau 13-17h du mercredi, selon cette configuration, seule le premier ajout devrait être autorisé.

Ajout des consultations :

```
2) INSERT INTO CARON_MED_PLAGE_CONSULT VALUES (107, 'mercredi', 13, 15);  
>>> 1 row created.
```

```
3) INSERT INTO CARON_MED_PLAGE_CONSULT VALUES (123, 'mercredi', 14, 15);
```

```
>>> ERROR at line 1:  
ORA-20123: Aucun salle n est disponible pour ce créneau  
ORA-06512: à "M1INF019.TRIGGER_EX7", ligne 10  
ORA-04088: erreur lors d'exécution du déclencheur 'M1INF019.TRIGGER_EX7'
```