

TP4 : ARBD

Triggers

Nom : ACHACHI

Prénom : Elhadj Ali

Mat : 181832008538

- 1) Créez un trigger qui affiche « **un nouveau sportif est ajouté** » après chaque insertion d'un sportif. Répétez la même chose pour la modification ou la suppression :

```
SQL> CREATE OR REPLACE TRIGGER AjoutSportif
  2 AFTER INSERT
  3 ON Sportifss
  4 BEGIN
  5 dbms_output.put_line('un nouveau sportif est ajouté');
  6 END;
  7 /
```

DÚclencheur crÚÚ.

```
SQL> SET SERVEROUTPUT ON;
SQL> INSERT INTO Sportifss VALUES ('155', 'ACHACHI', 'Ali', 'M',22,10);
un nouveau sportif est ajouté
```

1 ligne crÚÚe.

```
SQL> CREATE OR REPLACE TRIGGER SuppSportif
  2 AFTER DELETE
  3 ON Sportifss
  4 FOR EACH ROW
  5 BEGIN
  6 dbms_output.put_line('un nouveau sportif a été supprimé');
  7
  8 END;
  9 /
```

DÚclencheur crÚÚ.

```
SQL> SET SERVEROUTPUT ON;
SQL> DELETE FROM Sportifss where NOM='ACHACHI' and Prenom='Ali';
un nouveau sportif a été supprimé
```

1 ligne supprimÚe.

```

SQL> CREATE OR REPLACE TRIGGER ModifSportif
  2 AFTER UPDATE
  3 ON Sportifss
  4 FOR EACH ROW
  5 BEGIN
  6 dbms_output.put_line('un nouveau sportif a été modifié');
  7
  8 END;
  9 /

```

DÚclencheur crÚÚ.

```

SQL> SET SERVEROUTPUT ON;
SQL> UPDATE Sportifss
  2 SET Age = 88 WHERE IDSPORTIF = 130;
un nouveau sportif a été modifié

1 ligne mise Ó jour.

```

- 2) Créez un trigger qui affiche « **Une séance est ajoutée à l'entraîneur [Nom et prénom de sportif]** » après chaque insertion dans la table séance :

```

SQL> CREATE OR REPLACE TRIGGER AffichSeance
  2 AFTER INSERT
  3 ON Seances
  4 FOR EACH ROW
  5 declare
  6 nom varchar(30);
  7 prenom varchar(30);
  8 BEGIN
  9     select nom into nom from sportifss
 10     where IDSPORTIF = (:new.IDSPORTIFENTRAINEUR);
 11     select prenom into prenom from sportifss
 12     where idsportif= (:new.IDSPORTIFENTRAINEUR);
 13     dbms_output.put_line('une seance est ajouté a entraineur '|| nom ||' '||prenom);
 14 END;
 15 /

```

DÚclencheur crÚÚ.

```

SQL> SET SERVEROUTPUT ON;
SQL> INSERT INTO Seances VALUES (30, 15, 7, 'Mardi',18,90);
une seance est ajouté a entraineur HADJ Zouhir
INSERT INTO Seances VALUES (30, 15, 7, 'Mardi',18,90)
*
ERREUR Ó la ligne 1 :
ORA-04091: la table ARBDTP.SEANCES est en mutation ; le dÚclencheur ou la
fonction ne peut la voir
ORA-06512: Ó "ARBDTP.AFFICHE_SEANCE", ligne 4
ORA-04088: erreur lors d'exÚcution du dÚclencheur 'ARBDTP.AFFICHE_SEANCE'

```

- 3) Créer un trigger qui vérifie que lors de la modification du **l'âge d'un sportif**, la nouvelle valeur ne peut jamais être inférieure à la précédente :

```

SQL> CREATE OR REPLACE TRIGGER PasAge
  2 BEFORE UPDATE
  3 ON Sportifss
  4 FOR EACH ROW
  5 BEGIN
  6 IF (:new.age < :old.age) THEN
  7   raise_application_error (-20004, 'Pas ce age de sportif ');
  8 END IF;
  9
 10 END;
 11 /

```

Déclencheur créé.

```

SQL> SET SERVEROUTPUT ON;
SQL> UPDATE Sportifss
  2 SET Age = 17 WHERE IDSPORTIF = 70;
UPDATE Sportifss
      *
ERREUR Ó la ligne 1 :
ORA-20004: Pas ce age de sportif
ORA-06512: Ó "ARBDTP.PASAGE", ligne 3
ORA-04088: erreur lors d'exécution du déclencheur 'ARBDTP.PASAGE'

```

```

SQL> SET SERVEROUTPUT ON;
SQL> UPDATE Sportifss
  2 SET Age = 150 WHERE IDSPORTIF = 120;
un nouveau sportif a été modifié

1 ligne mise Ó jour.

```

- 4) ajoute deux attributs : **Total\_Entraîneurs** et **Total\_Arbitres** dans la table sport avec la remplissage :

```

SQL> ALTER TABLE SPORTS ADD ( Total_Entraîneurs NUMBER(10) DEFAULT 0);
Table modifiée.

SQL> declare
  2 cursor c is
  3 select s.idsport , s.LIBELLE , count(e.IDSPORTIFENTRAINEUR) as NB from entraineur e , sports s
  4 where s.idsport = e.idsport
  5 group by s.idsport, s.LIBELLE;
  6
  7 begin
  8 for i in c loop
  9 update sports set Total_Entraîneurs = i.NB
 10 where idsport = i.idsport
 11 and LIBELLE = i.LIBELLE;
 12 end loop;
 13 end;
 14 /

Procédure PL/SQL terminée avec succès.

```

```
SQL> ALTER TABLE SPORTS ADD ( Total_Arbitres NUMBER(10) DEFAULT 0);
```

Table modifiée.

```
SQL> declare
2  cursor c is
3  select s.idsport , s.LIBELLE , count(a.IDSPORTIF) as NBS from Arbitrer a , sports s
4  where s.idsport = a.idsport
5  group by s.idsport, s.LIBELLE;
6
7  begin
8  for i in c loop
9  update sports set Total_Arbitres = i.NBS
10 where idsport = i.idsport
11 and LIBELLE = i.LIBELLE;
12 end loop;
13 end;
14 /
```

Procédure PL/SQL terminée avec succès.

Tester la suppression et l'insertion d'entraîneur et arbitre :

```
SQL> CREATE OR REPLACE TRIGGER update_Total_Entraîneurs
2  AFTER INSERT OR delete ON ENTRAINER
3  FOR EACH ROW
4
5  DECLARE
6  total integer;
7
8  BEGIN
9  IF INSERTING THEN
10 SELECT Total_Entraîneurs into total from sports where idsport =(:new.idsport);
11 total := total+1;
12 update sports set Total_Entraîneurs =total where idsport =(:new.idsport);
13 END IF;
14
15 IF DELETING THEN
16 SELECT Total_Entraîneurs into total from sports where idsport =(:old.idsport);
17 total := total-1;
18 update sports set Total_Entraîneurs =total where idsport = (:old.idsport);
19 END IF;
20 END;
21 /
```

Déclencheur créé.

```

SQL> CREATE OR REPLACE TRIGGER update_Total_Arbitres
  2 AFTER INSERT OR delete ON ARBITRER
  3 FOR EACH ROW
  4
  5 DECLARE
  6 total integer;
  7
  8 BEGIN
  9 IF INSERTING THEN
10 SELECT Total_Arbitres into total from sports where idsport = (:new.idsport);
11 total := total+1;
12 update sports set Total_Arbitres =total where idsport = (:new.idsport);
13 END IF;
14
15 IF DELETING THEN
16 SELECT Total_Arbitres into total from sports where idsport = (:old.idsport);
17 total := total-1;
18 update sports set Total_Arbitres =total where idsport = (:old.idsport);
19 END IF;
20 END;
21 /

```

DÚclencheur crÚÚ.

Resultat apres le teste :

```
SQL> insert into arbitrer values (127,8);
```

1 ligne crÚÚe.

```
SQL> delete from arbitrer where idsport = 8;
```

2 ligne(s) supprimÚe(s).

```
SQL> insert into entrainer values (124,7);
```

1 ligne crÚÚe.

```
SQL> delete from entrainer where idsport = 2;
```

9 ligne(s) supprimÚe(s).

```
SQL> select * from sports;
```

	IDSPOINT	LIBELLE	TOTAL_ENTRAINEURS	TOTAL_ARBITRES
	1	Basket ball	6	8
	2	Volley ball	0	7
	3	Hand ball	5	2
	4	Tennis	0	2
	5	Hockey	4	1
	6	Badmington	12	2
	7	Ping pong	14	-1
	8	Football	0	0
	9	Boxe	1	1

9 ligne(s) sÚlectionnÚe(s).

5) Création du table **Historique\_Seance\_Par\_Jour** et vérifier la mise a jour :

```
SQL> create table Historique_Seance_Par_Jour (Jour varchar(10),total_seance number);
Table créée.

SQL> declare
  2 cursor cr is select jour from seances group by jour;
  3 a integer;
  4 begin
  5 for item in cr
  6 loop
  7 select count (*) into a from seances where jour = item.jour;
  8 dbms_output.put_line('le nombre est: '||a||' pour le jour: '||item.jour||'.');
  9 insert into Historique_Seance_Par_Jour values(item.jour , a);
 10 end loop;
 11 end;
 12 /
le nombre est: 11 pour le jour: Dimanche.
le nombre est: 15 pour le jour: Lundi.
le nombre est: 3 pour le jour: Mardi.
le nombre est: 4 pour le jour: Jeudi.
le nombre est: 4 pour le jour: Vendredi.
le nombre est: 11 pour le jour: Mercredi.
le nombre est: 7 pour le jour: Samedi.

Procédure PL/SQL terminée avec succès.
```

Le trigger pour gérer la mise a jour des seances :

```

SQL> CREATE OR REPLACE TRIGGER update_Total_Seances
  2  AFTER INSERT OR delete or update of jour ON seances
  3  FOR EACH ROW
  4
  5  DECLARE
  6  total integer;
  7
  8  BEGIN
  9  IF INSERTING THEN
10  SELECT total_seance into total from Historique_Seance_Par_Jour where jour =(:new.jour);
11  total := total+1;
12  update Historique_Seance_Par_Jour set total_seance =total where jour =(:new.jour);
13  END IF;
14
15  IF DELETING THEN
16  SELECT total_seance into total from Historique_Seance_Par_Jour where jour =(:old.jour);
17  total := total-1;
18  update Historique_Seance_Par_Jour set total_seance =total where jour = (:old.jour);
19  END IF;
20
21  IF UPDATING THEN
22  SELECT total_seance into total from Historique_Seance_Par_Jour where jour =(:old.jour);
23  total := total-1;
24  update Historique_Seance_Par_Jour set total_seance =total where jour = (:old.jour);
25
26  SELECT total_seance into total from Historique_Seance_Par_Jour where jour =(:new.jour);
27  total := total+1;
28  update Historique_Seance_Par_Jour set total_seance =total where jour = (:new.jour);
29  END IF;
30  END;
31  /

```

D  clencheur cr   .

Test (mais y'a problem je ne s'est pas pourquoi) :

```

SQL> select * from Historique_Seance_Par_Jour;

```

JOUR	TOTAL_SEANCE
Dimanche	11
Lundi	15
Mardi	3
Jeudi	4
Vendredi	4
Mercredi	11
Samedi	7

7 ligne(s) s  lectionn  e(s).

```

SQL> insert into seances values (27,7,7,'Mardi',10,90);

```

une seance est ajout   a entraineur HADJ Zouhir

```

insert into seances values (27,7,7,'Mardi',10,90)

```

\*

ERREUR    la ligne 1 :

ORA-04091: la table ARBDTP.SEANCES est en mutation ; le d  clencheur ou la fonction ne peut la voir

ORA-06512:    "ARBDTP.AFFICHE\_SEANCE", ligne 4

ORA-04088: erreur lors d'ex  cution du d  clencheur 'ARBDTP.AFFICHE\_SEANCE'