



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



**Université des Sciences et de la Technologie Houari Boumediene**

Faculté d'Electronique et d'Informatique

Département Informatique

**Spécialité : Systèmes d'Information Intelligent.**

---

## **TP ABDD N°5**

**NOM:** AISSAT

**Prénom:** Abdelhak

**Matricule:** 171733032123

## 1. Creation d'un trigger qui affiche « un sportifs est ajouté » après:

### 1.1 Une nouvelle insertion

```
CREATE OR REPLACE TRIGGER Trig_print_ajout_sportif
AFTER INSERT ON Sportifs FOR EACH ROW
BEGIN
dbms_output.put_line('Un nouveau sportif est ajouté!!');
END;
```

```
SQL> SET SERVEROUTPUT ON
SQL> CREATE OR REPLACE TRIGGER Trig_print_ajout_sportif
  AFTER INSERT ON Sportifs  FOR EACH ROW
  BEGIN
    dbms_output.put_line('Un nouveau sportif est ajouté!!');
  END;  2      3      4      5
  6
  7 /

Trigger created.

SQL> desc sportifs
Name                                Null?      Type
-----
IDSPORTIF                          NOT NULL   NUMBER
NOM                                VARCHAR2(50)
PRENOM                             VARCHAR2(45)
SEXE                               VARCHAR2(1)
AGE                                NUMBER
IDSPORTIFCONSEILLER                NUMBER

SQL> INSERT INTO Sportifs VALUES(200,'Testeur','Tournevisse','M',25,2);
Un nouveau sportif est ajouté!!

1 row created.

SQL>
```

### 1.2 Une modification

```
CREATE OR REPLACE TRIGGER Trig_print_modif_sportif
AFTER UPDATE ON Sportifs FOR EACH ROW
BEGIN
```

```

dbms_output.put_line('sportif updated. ');
END;
/

```

```

SQL> CREATE OR REPLACE TRIGGER Trig_print_modif_sportif
  AFTER UPDATE ON Sportifs  FOR EACH ROW
  BEGIN
    dbms_output.put_line('sportif  updated. ');
  END;
  2      3      4      5      6  ^Z
  7  /

Trigger created.

SQL> UPDATE Sportifs SET NOM='Humanoide' WHERE IDSPORTIF=200;
sportif updated.

1 row updated.

SQL> 

```

### 1.3 Une suppression:

```

CREATE OR REPLACE TRIGGER Trig_print_delete_sportif
AFTER DELETE ON Sportifs FOR EACH ROW
BEGIN
dbms_output.put_line('sportif Supprime! ');
END;
/

```

```

1 row updated.

SQL> CREATE OR REPLACE TRIGGER Trig_print_modif_sportif
  AFTER DELETE ON Sportifs  FOR EACH ROW
  BEGIN
    dbms_output.put_line('\nsportif Supprime! ');
  END;
  2      3      4      5
  6  /

Trigger created.

SQL> DELETE FROM SPORTIFS WHERE IDSPORT=200;
DELETE FROM SPORTIFS WHERE IDSPORT=200
*
ERROR at line 1:
ORA-00904: "IDSPORT": invalid identifier

SQL> DELETE FROM SPORTIFS WHERE IDSPORTIF=200;
\nsportif Supprime!

1 row deleted.

SQL> 

```

```

BEGIN
SELECT NOM INTO ne FROM SPORTIFS WHERE IDSPORTIF
= :new.IDSPORTIFENTRAINEUR;
SELECT S.PRENOM INTO pe FROM SPORTIFS S WHERE IDSPORTIF
= :new.IDSPORTIFENTRAINEUR;
dbms_output.put_line('Une seance est ajoutée à l entraîneur '||ne||' , '||pz||' .');
END;
/

```

## 2. Creation d'un trigger qui affiche « Une séance est ajoutée à l'entraîneur [Nom et prénom de sportif] » après chaque insertion dans la table séance

Pour une raison que j'ignore cette requete n'as pas voulu marché malgré le fait que je pense qu'elle soit correcte.

```

CREATE OR REPLACE TRIGGER TrigPriSeanceEntrain
AFTER INSERT ON seances
FOR EACH ROW
DECLARE
ne SPORTIF.NOM%TYPE;
pe SPORTIF.PRENOM%TYPE;
BEGIN
SELECT NOM INTO ne FROM SPORTIFS WHERE IDSPORTIF
= :new.IDSPORTIFENTRAINEUR;
SELECT S.PRENOM INTO pe FROM SPORTIFS S WHERE IDSPORTIF
= :new.IDSPORTIFENTRAINEUR;
dbms_output.put_line('Une seance est ajoutée à l entraîneur '||ne||' , '||pz||' .');
END;

```

## 3. Créer un trigger qui vérifie lors de a modification l'age d'un sportif avec la nouvelle valeur ne pouvant pas être inférieure a la précédente

```

CREATE OR REPLACE TRIGGER verif_age_sportif
BEFORE UPDATE ON Sportifs
FOR EACH ROW
BEGIN
IF :new.age < :old.age THEN
raise_application_error(-20101, 'new age doit etre plus grand que lancien');

```

```
END IF;
```

```
END;
```

```
/
```

```
SQL> select idsportif,age from sportifs where idsportif=12;
```

IDSPORTIF	AGE
12	29

```
SQL> CREATE OR REPLACE TRIGGER verif_age_sportif
  BEFORE UPDATE ON Sportifs
  FOR EACH ROW
  BEGIN
    IF :new.age < :old.age THEN
      raise_application_error(-20101, 'new age doit etre plus gra
nd que lancien');
    END IF;
  END;
  2      3      4      5      6      7      8
  9 /
```

Trigger created.

```
SQL> UPDATE Sportifs SET Age=28 WHERE IDSPORTIF=12;
UPDATE Sportifs SET Age=28 WHERE IDSPORTIF=12
*
```

ERROR at line 1:

```
ORA-20101: new age doit etre plus grand que lancien
ORA-06512: at "DBAGYMNASE.VERIF_AGE_SPORTIF", line 3
ORA-04088: error during execution of trigger 'DBAGYMNASE.VERIF_AGE
_SPORTIF'
```

```
SQL> █
```

#### 4. Ajouter les attributs et vérifications:

```
ALTER TABLE SPORTS
ADD TOTAL_ENTRAINEURS INTEGER;
ALTER TABLE SPORTS
ADD TOTAL_ARBITRES INTEGER;

DECLARE
CURSOR CR IS SELECT IDSPORT FROM SPORTS;

BEGIN

FOR ITEM IN CR
LOOP
UPDATE SPORTS
SET TOTAL_ARBITRES =(SELECT COUNT(DISTINCT IDSPORTIF) FROM ARBITRER
A WHERE A.IDSPORT=ITEM.IDSPORT )
WHERE IDSPORT = ITEM.IDSPORT;

UPDATE SPORTS
SET TOTAL_ENTRAINEURS =(SELECT COUNT(DISTINCT IDSPORTIFENTRAINEUR)
FROM ENTRAINER E WHERE E.IDSPORT=ITEM.IDSPORT )
WHERE IDSPORT = ITEM.IDSPORT;

END LOOP;

end;
/
--4 b
CREATE OR REPLACE TRIGGER DLL_ARBITRE
AFTER DELETE ON ARBITRER
FOR EACH ROW
BEGIN
UPDATE SPORTS
SET TOTAL_ARBITRES = TOTAL_ARBITRES - 1 WHERE IDSPORT = :old.IDSPORT;
END;
/

---ARBITRER
--AJOUT
```

```

CREATE OR REPLACE TRIGGER ADD_ARBITRE
AFTER DELETE ON ARBITRER
FOR EACH ROW
BEGIN
UPDATE SPORTS
SET TOTAL_ARBITRES = TOTAL_ARBITRES + 1 WHERE IDSPORT
= :NEW.IDSPORT;
END;
/

---
--SUPPRESSION
CREATE OR REPLACE TRIGGER DLL_ARBITRE
AFTER DELETE ON ARBITRER
FOR EACH ROW
BEGIN
UPDATE SPORTS
SET TOTAL_ARBITRES = TOTAL_ARBITRES - 1 WHERE IDSPORT
= :OLD.IDSPORT;
END;
/

---ENTRAINER
--AJOUT
CREATE OR REPLACE TRIGGER ADD_ENTRAINEUR
AFTER DELETE ON ENTRAINEUR
FOR EACH ROW
BEGIN
UPDATE SPORTS
SET TOTAL_ENTRAINEURS = TOTAL_ENTRAINEURS + 1 WHERE IDSPORT
= :NEW.IDSPORT;
END;
/

---
--SUPPRESSION
CREATE OR REPLACE TRIGGER DLL_ENTRAINEUR
AFTER DELETE ON ENTRAINEUR
FOR EACH ROW
BEGIN
UPDATE SPORTS
SET TOTAL_ENTRAINEURS = TOTAL_ENTRAINEURS - 1 WHERE IDSPORT
= :OLD.IDSPORT;

```

```
END;  
/
```

## 5.

### 5.1 Ajout table

```
create table historique_seance_par_jour(  
jour varchar2(30),  
total_seance int,  
constraint pk_hist_seance_par_jour primary key (jour)  
);
```





```

Trigger created.

SQL> create table historique_seance_par_jour(
    jour varchar2(30),
    total_seance int,
    constraint pk_hist_seance_par_jour primary key (jour)
); 2      3      4      5

Table created.

SQL> 

```

## 5.2 Verification de la contrainte lors d'une insertion un update et une suppression

```

create or replace trigger updateHistory
after insert on seances
for each row
declare
s_count number := 0;
begin
select count(*) into s_count from historique_seance_par_jour where jour
= :new.jour;
if (s_count = 0) then
insert into historique_seance_par_jour values (:new.jour, 1);
else
update historique_seance_par_jour set total_seance = total_seance + 1 where
jour = :new.jour;
end if;
end;
/

```

```

Table created.

SQL> create or replace trigger updateHistory
    after insert on seances
    for each row
    declare
    s_count number := 0;
    begin
        select count(*) into s_count from historique_seance_par_jour where jour = :new.jour
    ;
        if (s_count = 0) then
            insert into historique_seance_par_jour values (:new.jour, 1);
        else
            update historique_seance_par_jour set total_seance = total_seance + 1 where jour
r = :new.jour;
        end if;
    end;
    / 2      3      4      5      6      7      8      9      10     11     12     13     14

Trigger created.

SQL> 

```