



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
**Université des Sciences et de la Technologie Houari Boumediene**

Faculté d'Informatique

Spécialité : MASTER 1 Big Data Analytics

---

## Rapport de TP5 Triggers ARBD

---

Travail présenté à Madame Kahina KESSI

Travail présenté par :  
**AISSANI Anouar**  
**161835024828**

2021/2022

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 message after insert or delete or update on sportifs
  2  for each row
  3  begin
  4  case
  5  when inserting then dbms_output.put_line ('un sportif a été ajouté') ;
  6  when deleting then dbms_output.put_line ('un sportif a été supprimé') ;
  7  when updating then dbms_output.put_line ('un sportif a été modifié') ;
  8  end case;
  9  exception
 10  when others then dbms_output.put_line ('error: ' || sqlcode || ' ' || sqlerrm) ;
 11  end;
 12  /
```

Trigger created.

```
SQL>
```

```
SQL> SET SERVEROUTPUT ON
```

```
SQL>
```

```
SQL> --Vérification
```

```
SQL> INSERT INTO Sportifs VALUES(200,'AISSANI','Anouar','F',24,1);
```

un sportif a été ajouté

1 row created.

```
SQL> UPDATE Sportifs SET AGE = 20 WHERE IDSPORTIF = 200;
```

un sportif a été modifié

1 row updated.

```
SQL> DELETE FROM Sportifs WHERE IDSPORTIF = 200;
```

un sportif a été supprimé

1 row deleted.

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 affich_entraineur after insert on SEANCES
  2  for each row
  3  declare
  4  noms sportifs.nom%type;
  5  prenom sportifs.prenom%type;
  6  begin
  7  SELECT NOM, PRENOM into noms, prenom
  8  FROM sportifs
  9  WHERE idsportif = :new.IDSPORTIFENTRAINEUR;
 10  dbms_output.put_line ('Une séance est ajoutée à l'entraîneur '||noms||'
 11  ||prenom);
 12  end;
```

Trigger created.

```
SQL> INSERT INTO SEANCES VALUES(3,3,3,'Samedi',18.0,90);
Une séance est ajoutée à l'entraîneur BOUZIDI Amel

1 row created.
```

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 verifie_age before update of AGE on SPORTIFS
  2  for each row
  3  declare
  4  begin
  5  IF :new.age < :old.age THEN
  6  RAISE_APPLICATION_ERROR(-20005,'La contrainte verifie_age est violée');
  7  END IF;
  8  END;
  9  /
```

Trigger created.

--Vérification

```
SQL> select age from sportifs where idsportif = 2;
```

```
      AGE
-----
      33
```

```
SQL> UPDATE sportifs SET age = 30 WHERE idsportif = 2;
```

```
UPDATE sportifs SET age = 30 WHERE idsportif = 2
```

\*

ERROR at line 1:

ORA-20005: La contrainte verifie\_age est violée

ORA-06512: at "DBAGYMNASE.VERIFIE\_AGE", line 4

ORA-04088: error during execution of trigger 'DBAGYMNASE.VERIFIE\_AGE'

4. L'administrateur veut, pour un besoin interne, avoir le nombre total des sportifs entraineurs et sportifs arbitres pour chaque **sport**. Pour cela, il ajoute deux attributs :

**Total\_Entraîneurs** et **Total\_Arbitres** dans la table sport.

a. Ajouter les attributs et les initialiser.

```
SQL> ALTER TABLE sports add Total_Entraîneurs NUMBER;
```

Table altered.

```
SQL> ALTER TABLE sports add Total_Arbitres NUMBER;
```

Table altered.

```
SQL>
```

```
SQL> DESC sports;
```

Name	Null?	Type
-----	-----	-----
IDSport	NOT NULL	NUMBER(38)
LIBELLE		VARCHAR2(50)
TOTAL_ENTRAINEURS		NUMBER
TOTAL_ARBITRES		NUMBER

*/\*initialiser Les attributs Total\_Entraîneurs et Total\_Arbitres par Les valeurs qui existent dans la BD\*/*

```
SQL> create or replace procedure initialiser(codsport sports.idsport%type) is
  2  nbr_Entraîneurs  NUMBER;
  3  nbr_Arbitres     NUMBER;
  4  begin
  5  select count(DISTINCT IDSPORTIFENTRAINEUR) into nbr_Entraîneurs
  6  from  ENTRAINER
  7  where idsport = codsport;
  8  update sports set total_Entraîneurs = nbr_Entraîneurs
  9  where idsport = codsport;
 10  select count(DISTINCT IDSPORTIF) into nbr_Arbitres
 11  from  ARBITRER
 12  where idsport = codsport;
 13  update sports set total_Arbitres = nbr_Arbitres
 14  where idsport = codsport;
 15  end initialiser;
 16  /
```

Procedure created.

```
SQL> select idsport, total_Entraîneurs, total_Arbitres from sports;
```

IDSPORT	TOTAL_ENTRAINEURS	TOTAL_ARBITRES
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

11 rows selected.

*/\*exécution de la procedure pour tous les sports\*/*

```
SQL> declare
  2  cursor cr is select idsport from sports;
  3  begin
  4  for item in cr
  5  loop
  6  initialiser(item.idsport);
  7  end loop;
  8  end;
  9  /
```

PL/SQL procedure successfully completed.

```
SQL> select idsport, total_Entraîneurs, total_Arbitres from sports;
```

IDSport	TOTAL_ENTRAÎNEURS	TOTAL_ARBITRES
1	6	8
2	9	7
3	5	2
4	6	1
5	4	4
6	10	2
7	11	6
8	0	1
9	3	1
10	0	0
11	0	0

11 rows selected.

b. Vérifier cette contrainte lors d'une insertion et suppression.

```
-----INSERTION-----
SQL> create or replace trigger verifie_total_Entraîneurs after insert or delete on ENTRAINER
  2 for each ROW
  3 BEGIN
  4 CASE
  5     WHEN INSERTING THEN update sports set total_Entraîneurs = total_Entraîneurs + 1;
  6     WHEN DELETING THEN update sports set total_Entraîneurs = total_Entraîneurs - 1;
  7 END CASE;
  8 EXCEPTION
  9     WHEN OTHERS then DBMS_OUTPUT.PUT_LINE('error : '||sqlcode||' '||sqlerrm);
10 END;
11 /
```

Trigger created.

```
SQL>
SQL> create or replace trigger verifie_total_Arbitres after insert or delete on ARBITRER
  2 for each ROW
  3 BEGIN
  4 CASE
  5     WHEN INSERTING THEN update sports set total_Arbitres = total_Arbitres + 1;
  6     WHEN DELETING THEN update sports set total_Arbitres = total_Arbitres - 1;
  7 END CASE;
  8 EXCEPTION
  9     WHEN OTHERS then DBMS_OUTPUT.PUT_LINE('error : '||sqlcode||' '||sqlerrm);
10 END;
11 /
```

Trigger created.

```
--Vérification
SQL> INSERT INTO Arbitrer VALUES(151,2);

1 row created.

SQL> INSERT INTO Entraineur VALUES(151,2);
```

1 row created.

```
SQL> select idsport, total_Entraîneurs, total_Arbitres from sports;
```

IDSport	TOTAL_ENTRAÎNEURS	TOTAL_ARBITRES
1	7	9
2	10	8
3	6	3
4	7	2
5	5	5
6	11	3
7	12	7
8	1	2
9	4	2
10	1	1
11	1	1

11 rows selected.

- On remarque qu'après l'insertion d'un arbitre avec le sport 2 dans la table arbitrer le nombre total des arbitres dans la ligne « idsport = 2 » de la table sport a été incrémenter par 1, et la même chose pour l'insertion dans la table entraineur.

```
SQL> delete from Arbitrer WHERE IDSPORTIF = 151 AND IDSPORT = 2;
```

1 row deleted.

```
SQL> delete from Entrainer WHERE IDSPORTIFENTRAINEUR = 151 AND IDSPORT = 2;
```

1 row deleted.

```
SQL> select idsport, total_Entraîneurs, total_Arbitres from sports;
```

IDSport	TOTAL_ENTRAÎNEURS	TOTAL_ARBITRES
1	6	8
2	9	7
3	5	2
4	6	1
5	4	4
6	10	2



7		11		6
8		0		1
9		3		1
10		0		0
11		0		0

11 rows selected.

- On remarque qu'après la suppression d'un arbitre avec le sport 2 à partir la table arbitrer le nombre total des arbitres dans la ligne « idsport = 2 » de la table sport a été décrémenter par 1, et la même chose pour la suppression dans la table entraineur.

5. L'administrateur veut sauvegarder pour chaque jour le nombre de séances organisées. A chaque fois une séance est ajoutée dans un jour, on ajoute une ligne dans la table **Historique\_Seance\_Par\_Jour (Jour, total\_seance)** ou bien on met à jour l'attribut **total\_seance**.

- a. Ajouter la table.

```
SQL> create table Historique_Seance_Par_Jour (
  2   Jour  VARCHAR2(10),
  3   total_seance number,
  4   constraint pk_hc primary key(Jour),
  5   constraint checkJOUR CHECK (JOUR IN ('Samedi', 'Dimanche', 'Lundi', 'Mardi',
  6   'Mercredi', 'Jeudi', 'Vendredi'))
  );
```

Table created.

- b. Vérifier cette contrainte lors d'une insertion, d'un update et une suppression

-----INSERTION-----

```
SQL> CREATE OR REPLACE TRIGGER insert_Seance_Par_Jour AFTER INSERT ON seances
  2  FOR EACH ROW
  3  declare
  4  j int;
  5  BEGIN
  6  select count(Jour) into j from Historique_Seance_Par_Jour where jour = :new.jour;
  7  if (j = 0) then insert into Historique_Seance_Par_Jour (jour, total_seance) values
    (:new.jour, 1);
  8  else
  9  UPDATE Historique_Seance_Par_Jour set total_seance = total_seance + 1 where jour =
    :new.jour;
 10  end if;
 11  END ;
 12  /
```

Trigger created.

--Vérification

```
SQL> select * from Historique_Seance_Par_Jour;
```

no rows selected

```
SQL> INSERT INTO Seances VALUES(28,3,9,'Mardi',19.0,60);
```

Une séance est ajoutée à l entraîneur ABDELMOUMEN Nadia

1 row created.

```
SQL> select * from Historique_Seance_Par_Jour;
```

JOUR	TOTAL_SEANCE
Mardi	1

----- UPDATE -----

```
SQL> CREATE OR REPLACE TRIGGER modif_Seance_Par_Jour AFTER update ON seances
  2  FOR EACH ROW
  3  declare
  4  j int;
  5  BEGIN
  6  UPDATE Historique_Seance_Par_Jour set total_seance = total_seance - 1 where jour =
:old.jour;
  7  select count(jour) into j from Historique_Seance_Par_Jour where jour = :new.jour;
  8  if (j = 0) then insert into Historique_Seance_Par_Jour (jour, total_seance) values
(:new.jour, 1);
  9  else
 10  UPDATE Historique_Seance_Par_Jour set total_seance = total_seance + 1 where jour =
:new.jour;
 11  end if;
 12  END ;
 13  /
```

Trigger created.

--Vérification

```
SQL> select * from Historique_Seance_Par_Jour;
```

JOUR	TOTAL_SEANCE
Mardi	1

```
SQL> UPDATE SEANCES SET jour = 'Dimanche' WHERE IDGYMNASE = 28 and IDSPORT = 3 and
IDSPORTIFENTRAINEUR = 9 and jour = 'Mardi' and HORAIRE = 18.0;
```

1 row updated.

```
SQL>
SQL> select * from Historique_Seance_Par_Jour;
```

JOUR	TOTAL_SEANCE
Mardi	0
Dimanche	1

```

-----DELETED-----
SQL> CREATE OR REPLACE TRIGGER supp_hist_Seance_Par_Jour AFTER delete ON seances
  2  FOR EACH ROW
  3  declare
  4  j int;
  5  BEGIN
  6  select count(jour) into j from Historique_Seance_Par_Jour where jour = :old.jour;
  7  if (j = 0) then dbms_output.put_line('le jour n'est pas dans la table donc rien à
modifier');
  8  else
  9  UPDATE Historique_Seance_Par_Jour set total_seance = total_seance - 1 where jour =
:old.jour;
 10  end if;
 11  END ;
 12  /

```

Trigger created.

--Vérification

```
SQL> select * from Historique_Seance_Par_Jour;
```

JOUR	TOTAL_SEANCE
Mardi	0
Dimanche	1

```
SQL> delete from seances where IDGYMNASE = 28 and idsport = 5 and IDSPORTIFENTRAINEUR =
7 and jour = 'Vendredi' and horaire = 18.0;
le jour nest pas dans la table donc rien à modifier
```

1 row deleted.

```
SQL>
```

```
SQL> select * from Historique_Seance_Par_Jour;
```

JOUR	TOTAL_SEANCE
Mardi	0
Dimanche	1

```
SQL> delete from seances where IDGYMNASE = 17 and IDSPORT = 3 and IDSPORTIFENTRAINEUR = 9 and jour = 'Dimanche' and HORAIRE = 18.0;
```

```
1 row deleted.
```

```
SQL> select * from Historique_Seance_Par_Jour;
```

JOUR	TOTAL_SEANCE
Mardi	0
Dimanche	0