

Rapport TP

3

Base des données 2

Isil B
TP groupe 2

ferchichi manel

202031036637

Les reponses du Travail demandé:

Partie une : Contraintes d'Intégrité Statiques :

1.Lister l'ensemble de contraintes d'intégrité pour chaque table de ce schéma:

Query :

- `select constraint_name , constraint_type from user_constraints ;`

Capture:

```
SQL> select constraint_name ,constraint_type from user_constraints;
CONSTRAINT_NAME
-----
C
-
FREFFECTUERONE
R
FR
R
FRPRELEV
R
CONSTRAINT_NAME
-----
C
-
FR2
R
FREFFECTUERTWO
R
FREFFECTUERTHREE
R
```

2.Étendre la liste des rôles possibles avec la nouvelle fonction : « B-M à Domicile »:

Query :

- `alter table Biologistes add constraint question1 CHECK (RoleB in ('Biologiste-Responsable','Biologist-Médical','Ing-Qualité','Aide-laboratoire','Secrétaire','Technicien','Ing-Informatique')) ;`

Capture :

```
SQL> alter table Biologistes add constraint question1 CHECK (RoleB in ('Biologiste-Responsable', 'Biologist-Médical', 'Ing-Qualité', 'Aide-laboratoire', 'Secrétaire', 'Technicien', 'Ing-Informatique'));
```

Table modifiée.

3.Exigez que la date de naissance des patients soit antérieure à la date d'aujourd'hui. Tester avec quelques insertions:

Explications :

Si j' utilise add constraint check il faut spécifier une date précise (date d' aujourd'hui) mais ca sera pas dynamique et il faut changer la date a chaque fois pour vérifier cette contrainte :

Query statique :

- alter table Patient add constraint question3 check (Datenaissance <= (to_date('30/11/2022','dd/mm/yyyy')));

Capture:

```
SQL> alter table Patient add constraint question3 check ( Datenaissance <= (to_date('30/11/2022','dd/mm/yyyy')));
```

Table modifiée.

A better solution :

use A trigger and default date is (sysdate)

Query :

- CREATE OR REPLACE TRIGGER check_birth
BEFORE INSERT OR UPDATE ON Patient
FOR EACH ROW
BEGIN
IF(:new.Datenaissance > sysdate) THEN
RAISE_APPLICATION_ERROR(-20002, 'are you sure that this is the
right patient s birth date please verify it again it must be inferior
than the actual date thank you ! ');
END IF;
END;
/

Capture :

```
SQL> CREATE OR REPLACE TRIGGER question3
  2 BEFORE INSERT OR UPDATE ON Patient
  3 FOR EACH ROW
  4 BEGIN
  5 IF( :new.Datenaissance > sysdate ) THEN RAISE_APPLICATION_ERROR( -20002, 'are you sure that
this is the right patient s birth date please verify it again it must be inferior than the actual
date thank you ! ' );
  6 END IF;
  7 END;
  8 /

Dúclencheur crÚÚ.
```

Tests :

bon cas :

Query :

- insert into Patient
values(12,'Manel','Ferchichi',To_date('26/09/2022'
, 'DD/MM/YYYY')) ;

capture

```
SQL> insert into Patient values(12,'Manel','Ferchichi',To_date('26/09/2022' , 'DD/MM/YYYY'));

1 ligne crÚÚe.
```

wrong case :

Query :

- insert into Patient
values(13,'Chaima','Ferchichi',To_date('26/09/2023'
, 'DD/MM/YYYY')) ;

capture

```
SQL> insert into Patient values(13,'Chaima','Ferchichi',To_date('26/09/2023' , 'DD/MM/YYYY'));
insert into Patient values(13,'Chaima','Ferchichi',To_date('26/09/2023' , 'DD/MM/YYYY'))
*
ERREUR Ó la ligne 1 :
ORA-20002: are you sure that this is the right patient s birth date please
verify it again it must be inferior than the actual date thank you !
ORA-06512: Ó "TP2.QUESTION3", ligne 2
ORA-04088: erreur lors d'exÚcution du dúclencheur 'TP2.QUESTION3'
```

4. Un ingénieur ne peut exercer en tant que Bio-Medical. Ajouter la contrainte et effectuer les tests nécessaires:

on peut faire cette question avec les triggers ou juste avec la contrainte check

avec trigger :Query :

- create or replace trigger exercerquestion4
before insert or update on Biologistes
for each row
begin
if (:new.Specialite='Ingénieur' and
:new.RoleB='Biologist-Médical')
then
raise_application_error(-20003 , 'un ingénieur ne peut pas
exercer en tant que biomedical veuillez vérifier le rôle de ce
biologiste merci !');
end if;
end;
/

capture :

```
SQL> create or replace trigger exercerquestion4
2   before insert or update on Biologistes
3   for each row
4   begin
5   if (:new.Specialite='Ingénieur' and :new.RoleB='Biologist-Médical' )
6   then
7   raise_application_error(-20003 , 'un ingénieur ne peut pas exercer en tant que biomedical veu
illegz vérifier le rôle de ce biologiste merci !' );
8   end if;
9   end;
10  /

DÚclencheur crÚÚ.
```

avec check :

Query:

- alter table Biologistes add constraint question4 check
(not(Specialite='Ingénieur' and RoleB='Biologist-Médical'));

Tests :

cas bon :

Capture:

```
SQL> Insert Into Biologistes Values(11 , 'meryem' , 'hello' , 'Ingénieur' , 'Ing-Informatique');

1 ligne crÚÚe.
```

mauvais cas :

Capture:

```
SQL> Insert Into Biologistes Values(11 , 'meryem' , 'hello' , 'Ingénieur' , 'Biologist-Médical');
Insert Into Biologistes Values(11 , 'meryem' , 'hello' , 'Ingénieur' , 'Biologist-Médical')
*
ERREUR Ó la ligne 1 :
ORA-20003: un ingénieur ne peut pas exercer en tant que biomedical veuillez
vérifier le rôle de ce biologiste merci !
ORA-06512: Ó "TP2.EXERCERQUESTION4", ligne 4
ORA-04088: erreur lors d'exÚcution du dÚclencheur 'TP2.EXERCERQUESTION4'
```

5. Supprimer la table EffectuePrelevement. Que remarquez-vous?:

Query :

- Drop table Biologistes ;

Capture:

```
SQL> DROP TABLE Biologistes ;
DROP TABLE Biologistes
      *
ERREUR Ó la ligne 1 :
ORA-02449: clús uniques/primaires de la table rÚfÚrencÚes par des clús
ÚtrangPres
```

Remarque:

ON NE PEUT pas FAIRE DROP classe mère qui est Biologiste
puisque sa clé primaire est une clé étrangère dans la table
fille Effectuerprelev donc il faut faire drop constraints

- Désactiver la CI qui bloque la suppression et réessayer. Que remarquez-vous?:

Query :

- alter table Effectuerprelev disable constraint freffectuerone ;

Capture:

```
SQL> alter table Effectuerprelev disable constraint freffectuerone ;
Table modifiÚe.

SQL> drop table Biologistes ;
drop table Biologistes
      *
ERREUR Ó la ligne 1 :
ORA-02449: clús uniques/primaires de la table rÚfÚrencÚes par des clús
ÚtrangPres
```

Remarque :

on aura toujours le meme probleme meme si on utilise diable
sur la clé étrangère

- Supprimer cette contrainte et réessayer.

Capture:

```
SQL> alter table Effectuerprelev drop constraint frkey;
Table modifiée.

SQL> alter table Effectuerprelev2 drop constraint fr;
Table modifiée.

SQL> drop table Biologistes ;
Table supprimée.
```

Conclusion:

on ne peut pas supprimer une table mère si il existe une table fille avec une clé étrangère references table mère

Disable resout le probleme de insert et update seulement mais ca ne resout pas le problème de suppression de la classe mere ,dropping the references foreign keys is the only solution

deuxième solution : drop table fille puis drop table mère

- *Recréer la table EffectuePrelevement.*

Query :

- create table Biologiste (NumB number constraint cleprimaire primary key , Nom varchar2(15) , Prenom varchar2(15) ,Specialite varchar(40) , RoleB varchar2(50));

Capture :

```
SQL> create table Biologiste (NumB number constraint cleprimaire primary key , Nom varchar2(15) ,
Prenom varchar2(15) ,Specialite varchar(40) , RoleB varchar2(50) );
Table créée.
```

puis effectuerprelevement :

Query :

- Create table Effectuerprelev (NumB number (10) , NumP number(10) , NumPr number(10) , Constraint preffectuer primary key (NumB, NumP) , constraint freffectuereone foreign key (NumB) References Biologistes (NumB) , constraint freffectuertwo

```
foreign key (NumP) references Patient (NumP) ,constraint
freetuethree foreign key (NumPr) references Prelevement
(NumPr) );
```

capture:

```
SQL> desc Effectuerprelev ;
```

Nom	NULL ?	Type
NUMB	NOT NULL	NUMBER(10)
NUMP	NOT NULL	NUMBER(10)
NUMPR		NUMBER(10)

Partie deux :PL/SQL :

1. Pour analyser l'organigramme du laboratoire, nous cherchons à afficher le personnel et leurs rôles.

Ecrire un code PL SQL qui permet d'afficher pour chaque rôle, le nombre de biologistes correspondants.

Exemple : Il y a 2 personne(s) qui exerce(nt) en tant que «Aide-laboratoire».

code :

```
DECLARE    //delaration des variables
CURSOR cur is select RoleB ,count(NumB) as nbrbio from Biologistes
group by RoleB; //cur se comporte de RoleB et nbrbio
c_rec cur%rowtype ; //c_rec est du meme type que cur
Begin //partie instructions
for c_rec in cur loop //on utilise c_rec pour le parcours de cur
dbms_output.put_line('il ya' ||c_rec.nbrbio||'personne(s) qui
exerce(nt) en tant que '||c_rec.RoleB);
//afficher pour chaque biologiste son nombre de prelevements
exit when cur%notfound ;
end loop;
end;
/
set serveroutput on ;//pour l affichage
```

Résultat:

capture:

```
SQL> DECLARE
2  CURSOR cur is select RoleB ,count(NumB) as nbrbio from Biologistes group by RoleB;
3  c_rec cur%rowtype ;
4  Begin
5  for c_rec in cur loop
6  dbms_output.put_line('il ya' ||c_rec.nbrbio||'personne(s) qui exerce(nt) en tant que '
7  exit when cur%notfound ;
8  end loop;
9  end;
10 /
il ya3personne(s) qui exerce(nt) en tant que Biologist-Médical
il ya1personne(s) qui exerce(nt) en tant que Ing-Informatique
il ya1personne(s) qui exerce(nt) en tant que Ing-Qualité
il ya2personne(s) qui exerce(nt) en tant que Aide-laboratoire
il ya1personne(s) qui exerce(nt) en tant que Secrétaire
il ya1personne(s) qui exerce(nt) en tant que Technicien
il ya1personne(s) qui exerce(nt) en tant que Biologiste-Responsable

Procédure PL/SQL terminée avec succès.

SQL> set serveroutput on ;
SQL>
```

2. Supposant que la norme mondiale pour le test antigénique Covid passe du seuil initial "0.5" à "0.8" pour que le résultat soit déclaré positif. Ajouter l'information, puis ajoutez la contrainte suivante : si le taux d'antigènes détectés est > 0.8 alors la conclusion doit être = "Positif".
Vérifiez après mis-à-j.

commentaire : ON Peut utiliser mais ca sera statique et elle ne change pas lors de l'update :

- `alter table Resultat add constraint normeche check ((TO_NUMBER(Norme)>=to_number('0.5')) and (TO_NUMBER(Norme)<=to_number('0.8'))) ;`

code :

```
create or replace trigger normecheck //name of the trigger
after insert or update on Resultat //avant maj sur Resultat
for each row //parcours de toutes les lignes
begin
    if :new.TypeResultat='Antig-Covid' then
        //si la valeur de resultat est superieur a 0.8 et que la conclusion
        est pas positif =>update conclusion
        if (to_number(:old.Resul,'9.9')> 0.8 ) then
            if (:new.Conclusion <> 'Positif') then
                update Resultat set Conclusion = 'Positif';
            end if;
        end if ;
    end if ;
end;
/
```

capture :

```
SQL> create or replace trigger normecheck
2   after insert or update on Resultat
3   for each row
4   begin
5   if :new.TypeResultat='Antig-Covid' then
6   if (to_number(:old.Resul,'9.9')> 0.8 ) then
7       if (:new.Conclusion <> 'Positif') then
8       update Resultat set Conclusion = 'Positif';
9       end if;
10      end if ;
11      end if ;
12      end;
13      /
```

DÚclencheur crÚÚ.

tests:

(remarque : j' ai essayé de la changer plusieurs fois mais ca ne marche pas sur update)

capture :

```
SQL> insert into Resultat values (16,7,'Antig-Covid',0.9,'','');
```

1 ligne créée.

```
SQL> select * from Resultat where NumR=16 ;
```

NUMR	NUMPR	TYPES	RESULTAT
RESUL			NORME
CONCLUSION			
16	7	Antig-Covid	0.9

3. Écrire une procédure CasPositifs qui affiche « Le patient "i" a été testé positif. » si le patient en question a un résultat positif aux tests Covid de différents types de prélèvement.

code:

```
create or replace procedure CasPositifs(nump Patient.NumP%type)
is //creation de la procedure named caspositifs avec ses parametres
cursor cr is select E.NumP as nump from Effectuerprelev E , Resultat
R where R.Conclusion='Positif' and R.NumPr=E.NumPr ;
//cr contient les numeros des patients qui ont eu une conclusion du test
positive
element cr%rowtype; //element de type cr
BEGIN
for element in cr loop //parcours de tout cr pour rechercher les patient
if (element.nump =nump )then //si num patient du parametre existe dans
cr alors afficher ce dernier
dbms_output.put_line('le patient '||element.nump||'a ete testé
positif');
end if ;
end loop;
end ;
/
set serveroutput on ; // pour l affichage
```

capture : *//we call function with nombre of patient ex : call caspositifs(5);*

```
SQL> create or replace procedure CasPositifs(nump Patient.NumP%type)
2 is
3 cursor cr is select E.NumP as nump from Effectuerprelev E , Resultat R where R.Conclusion='Positif' and R.NumPr=E.N
umPr ;
4 element cr%rowtype;
5 BEGIN
6 for element in cr loop
7 if (element.nump =nump )then
8 dbms_output.put_line('le patient '||element.nump||'a ete testé positif');
9 end if ;
10 end loop;
11 end ;
12 /
Procédure crÚÚe.
```

tests:

```
SQL> set serveroutput on ;
SQL> call CasPositifs(4);
le patient 4a ete testé positif
Appel terminÚ.
```

4. Ecrire une fonction qui retourne, pour un Biologiste donné, le nombre de prélèvements effectués.

Exécuter la fonction pour plusieurs biologistes.

Exemple : Le Biologiste BADI a effectué 3 prélèvements.

code:

```
—
create or replace procedure nbrprelev(numb Biologistes.NumB%type)
IS // declaration de la procedure nbrprelev avec son parametre
cursor cr is select NumB , COUNT(NumPr) as nbrpr from
Effectuerprelev group by NumB ; // cr contient les numb et nombre
prelevement
element cr%rowtype;// element de type cr
BEGIN
for element in cr loop //parcours de cr pour rechercher si numb existe
if (element.NumB=numb )then
//si numb existe alors afficher son nombres de prelevements effectue
dbms_output.put_line('LE BIOLOGISTE '||element.NumB||' a effectué
'||element.nbrpr||' prelevements');
end if ;
end loop;
end ;
/
set serveroutput on
```

capture :

```
SQL> create or replace procedure nbrprelev(numb Biologistes.NumB%type)
  2  IS
  3  cursor cr is select NumB , COUNT(NumPr) as nbrpr from Effectuerprelev group by NumB ;
  4
  5  element cr%rowtype;
  6  BEGIN
  7  for element in cr loop
  8  if (element.NumB=numb )then
  9  dbms_output.put_line('LE BIOLOGISTE '||element.NumB||'a effectué '||element.nbrpr||'prelevements');
 10  end if ;
 11  end loop;
 12  end ;
 13  /

Procédure créée.
```

tests:

```
SQL> set serveroutput on
SQL> CALL nbrprelev(1);
LE BIOLOGISTE 1a effectué 4prelevements

Appel terminé.
```

5. Créer une procédure qui permet d'ajouter un prélèvement à partir de tous les attributs nécessaires. N'oublier pas de vérifier l'unicité de la clé et l'existence de clé étrangère vers Patient et Biologiste. Affichez les messages d'erreurs en cas de problèmes.

code:

```
Create or replace procedure Ajouterprelevement(NumPrelev number ,
NumPatient number, DatePrelev date ,TypePrelev varchar2)
is // declaration de procedure avec les parametre suffisants pour l
insertion d un element de prelevement
Cursor cpr is Select * from Prelevement Where NumPr= NumPrelev;
//cpr contient tout les num prelevements existants
Cursor cp is Select * from Patient Where NumP= NumPatient;
//cp contient tous les nums patients existants
Row_pr cpr%rowtype;//Row_pr de type cpr
Row_p cp%rowtype;//Row_p de type cp
Begin
Open cpr; //ouvrir cpr
Fetch cpr into Row_pr;//use the function fetch to find the Row_pr
If(cpr%found)then //si le num prelevement existe alors violation de la
contraintes cle primaire
```

```

        Raise_application_error(-20008,'NumPr '||NumPrelev||' existe
déjà');
    end if;
    close cpr;//on fait pareil pour num patient entré par contre si il n
existe pas dans cp alors on ne peut pas ajouter un prelevement susr un
patient non existant
    Open cp;
    Fetch cp into Row_p;
    If(cp%notfound)then
        Raise_application_error(-20009,'NumP '||NumPatient||'n existe
pas ');
    end if;
    close cp;
//apres les verifications on execute insert si tout les cas sont verifiés
Insert into Prelevement
values(NumPrelev,NumPatient,DatePrelev,TypePrelev);
    end;
/
set serveroutput on ;

```

capture :

```

SQL> Create or replace procedure Ajouterprelevement(NumPrelev number , NumPatient number, DatePrelev date ,TypePrelev varchar2)
2  is
3  Cursor cpr is Select * from Prelevement Where NumPr= NumPrelev;
4  Cursor cp is Select * from Patient Where NumP= NumPatient;
5
6      Row_pr cpr%rowtype;
7      Row_p cp%rowtype;
8
9  Begin
10  Open cpr;
11  Fetch cpr into Row_pr;
12  If(cpr%found)then
13      Raise_application_error(-20008,'NumPr '||NumPrelev||' existe déjà');
14  end if;
15  close cpr;
16
17  Open cp;
18  Fetch cp into Row_p;
19  If(cp%notfound)then
20      Raise_application_error(-20009,'NumP '||NumPatient||'n existe pas ');
21  end if;
22  close cp;
23
24  Insert into Prelevement values(NumPrelev,NumPatient,DatePrelev,TypePrelev);
25  end;
26  /
Procédure créée.

```

tests :

captures :

cas bon :

```

SQL> call Ajouterprelevement (14, 5 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Sanguin');
Appel terminé.

```

cas mauvais :

```
SQL> set serveroutput on ;
SQL> call Ajouterprelevement (7, 5 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Sanguin');
call Ajouterprelevement (7, 5 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Sanguin')
*
ERREUR Ó la ligne 1 :
ORA-20008: NumPr 7 existe déjà
ORA-06512: Ó "TP2.AJOUTERPRELEVEMENT", ligne 13
```

6. Créez un trigger qui affiche « un nouveau patient est ajouté» après chaque insertion d'un patient. Répétez la même chose pour la modification ou la suppression.

- Insertion :

Code:

```
create or replace trigger newpatient //creation d'un trigger
after insert on Patient //sur la table patient après 1 insertion
begin
DBMS_OUTPUT.PUT_LINE('nouveau patient ajouté');//si 1 insertion s
execute avec success alors afficher ce message
end;
/
```

capture :

```
SQL> create or replace trigger newpatient
2 after insert on Patient
3 begin
4 DBMS_OUTPUT.PUT_LINE('nouveau patient ajouté');
5 end;
6 /
```

DÚclencheur crÚÚ.

test:

capture :

```
SQL> insert into Patient values(13,'Chaima','Ferchichi',To_date('26/09/2021', 'DD/MM/YYYY'));
nouveau patient ajouté

1 ligne crÚÚe.
```

- Modification :

code :

```
create or replace trigger replacepatient //creation d'un trigger
after Update on Patient //sur la table patient après modification
begin
DBMS_OUTPUT.PUT_LINE( 'patient modifié');//ce message sera afficher si
la modification s est execute avec success
end;
/
```

capture:

```
SQL> create or replace trigger replacepatient
  2  after Update on Patient
  3  begin
  4  DBMS_OUTPUT.PUT_LINE( 'patient modifié');
  5  end;
  6  /

DÚclencheur crÚÚ.
```

test:

capture :

```
SQL> Update Patient set Nom='ferchichi' ;
patient modifié

11 lignes mises Ó jour.

SQL> Update Patient set Prenom='Manel' ;
patient modifié

11 lignes mises Ó jour.
```

- Suppression:

code : *//meme chose que modification sauf c est le cas de suppression*

```
create or replace trigger nopatient
after Delete on Patient
begin
DBMS_OUTPUT.PUT_LINE('patient supprimé');
end;
/
```

capture :

```
SQL> create or replace trigger nopatient
  2  after Delete on Patient
  3  begin
  4  DBMS_OUTPUT.PUT_LINE('patient supprimé');
  5  end;
  6  /

DÚclencheur crÚÚ.
```

example :

capture:


```
SQL> delete from Patient where NumP=13;
patient supprimé

1 ligne supprimée.
```

7. Créer un trigger qui empêche la modification du numéro de biologiste de la table des prélèvements.

code :

```
create or replace trigger forbidden //declaration du trigger
before update on Effectuerprelev //sur la table Effectuerprelevement
avant update , lors de la compilation
for each row
begin //on peut utilise fetch sur cursor avec PlsQL
if (:new.NumB <> :old.NumB ) //comparaison des NumB existant est NumB
entrée si ca existe alors ca sera error
then
Raise_application_error (-20888 , ' you can't change IDF of an
existent   Biologiste in this table please check your updating again
');
end if ;
end ;
/
```

capture :

```
SQL> create or replace trigger forbidden
2 before update on Effectuerprelev
3 for each row
4 begin
5 if (:new.NumB <> :old.NumB )
6 then
7 Raise_application_error (-20888 , ' you cant chane IDF of an existent
updating again ');
8 end if ;
9 end ;
10 /

Déclencheur créé.
```

Test :

```
SQL> UPDATE Effectuerprelev set NumB=1 where NumPr=1;
UPDATE Effectuerprelev set NumB=1 where NumPr=1
*
ERREUR Ó la ligne 1 :
ORA-20888: you cant chane IDF of an existent   Biologiste in this table please
check your updating again
ORA-06512: Ó "TP2.FORBIDDEN", ligne 4
ORA-04088: erreur lors d'exécution du déclencheur 'TP2.FORBIDDEN'
```

8. L'administrateur veut, pour un besoin interne, avoir le nombre total des prélèvements par biologistes. Pour cela, il ajoute un attribut : Nb_Pr dans la table Biologistes.

Ajoutez l'attribut.

Créez un trigger T1 qui permet de m à j automatiquement l'attribut Nb_Pr.

- Etape 1:
ajouter la column Nb_Pr dans la table Biologistes :
capture :

```
SQL> alter table Biologistes add Nb_Pr number(5);  
Table modifi e.  
  
SQL> alter table Biologistes modify Nb_Pr number(5) default 0 ;  
Table modifi e.
```

- Etape 2:
initialiser et d finir la valeur de l' attribut

Query :

- Update Biologiste B set Nb_Pr =(select count(NumPr) from Effectuerprelev E where B.NumB =E.NumB);
capture :

```
SQL> Update Biologistes B set Nb_Pr =(select count(*) from Effectuerprelev E where B.NumB =E.NumB );
10 lignes mises à jour.
SQL> select Nb_Pr from Biologistes ;
```

NB_PR
3
3
0
2
0
0
3
0
2
2

```
10 lignes sélectionnées.
```

- Etape 3 :

ajouter un trigger pour update le num a chaque maj :

Code :

```

--
create or replace trigger countprlev
after update or insert or delete on Effectuerprelev
for each row
begin

if inserting then
    update Biologistes set Nb_Pr = Nb_Pr +1 where NumB=:new.NumB ;
end if ;

if deleting then
    update Biologistes set Nb_Pr = Nb_Pr -1  where NumB=:old.NumB ;
end if ;

if updating then
    update Biologistes set Nb_Pr = Nb_Pr +1 where NumB=:new.NumB ;
    update Biologistes set Nb_Pr = Nb_Pr -1 where NumB=:old.NumB ;
end if;

end ;
/

```

capture:

```

SQL> create or replace trigger countprelev
  2  after update or insert or delete on Effectuerprelev
  3  for each row
  4  begin
  5
  6  if inserting then
  7    update Biologistes set Nb_Pr = Nb_Pr +1 where NumB=:new.NumB ;
  8  end if ;
  9
 10  if deleting then
 11    update Biologistes set Nb_Pr = Nb_Pr -1  where NumB=:old.NumB ;
 12  end if ;
 13
 14  if updating then
 15    update Biologistes set Nb_Pr = Nb_Pr +1 where NumB=:new.NumB ;
 16    update Biologistes set Nb_Pr = Nb_Pr -1 where NumB=:old.NumB ;
 17  end if;
 18
 19  end ;
 20  /

```

DÚclencheur crÚÚ.

TESTS /
captures :

```
SQL> select NumB , Nb_Pr from Biologistes where NumB =1 ;
```

NUMB	NB_PR
1	3

```
SQL> Insert into Effectuerprelev values (1,10 ,13);
```

1 ligne crÚÚe.

```
SQL> select NumB , Nb_Pr from Biologistes where NumB =1 ;
```

NUMB	NB_PR
1	4