

Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Informatique



TP Bio SGBD

Rapport TP 3 : Contraintes d'Intégrité Statiques et Dynamiques

Fait par :

Nom et prénom : ABDELMALEK BENMEZIANE

Matricule : 171731046778

Spécialité : M1 BIOINFO

Section : A

Contents

1	Partie 1: Contraintes d'intégrité statiques	1
1.1	Les contraintes de chaque table de ce schéma	1
1.1.1	La table Biologiste	1
1.1.2	La table Patient	1
1.1.3	La table Prelevement	2
1.1.4	La table EffectuerPrelevement	2
1.1.5	La table Resultat	2
1.2	Etendre la liste des roles possibles	3
1.2.1	La requette des roles	3
1.2.2	La requette de la contrainte	4
1.3	La date de naissance des patients antérieure à la date d'aujourd'hui. Tester avec quelques insertions.	4
1.3.1	Le trigger	4
1.3.2	Les insertions	5
1.4	La contrainte d'un ingénieur ne peut exercer en tant que Bio- Medical	7
1.4.1	La contrainte	7
1.4.2	L'insertion	7
1.5	Les modifications sur la table Biologiste	7
1.5.1	Suppression la table Biologiste et remarque	7
1.5.2	suppression la CI qui bloque la suppression table et la remarque	8

1.5.3	Conculsion	9
1.5.4	Recréer la table Biologiste	9
2	Partie 2 : PL/SQL	11
2.1	Affichage du nombre de biologistes par role	11
2.2	Le test antigénique Covid positif	11
2.2.1	Les modifications	11
2.2.2	Trigger	12
2.2.3	La vérification	12
2.3	Procédure CasPositifs	13
2.4	Fonction nombre de prelevements	14
2.5	Procédure ajouter un prélèvement	15
2.6	Trigger affiche un message	16
2.6.1	Après insertion	16
2.6.2	Après modification	16
2.6.3	Après suppression	17
2.7	Trigger empêche la modification du numéro de biologiste	17
2.8	Trigger T1 de le màj automatique de l'attribut Nb_Pr	18
2.8.1	L'ajout de l'attribut Nb_Pr	18
2.8.2	Trigger T1	18

List of Figures

1.1	Contraintes biologiste	1
1.2	Output	1
1.3	Contraintes patient	1
1.4	Output	2
1.5	Contraintes prelevement	2
1.6	Output	2
1.7	Contraintes effectuer prelevement	2
1.8	Output	2
1.9	Contraintes Resultat	2
1.10	Output	3
1.11	Roles biologiste	3
1.12	Output	3
1.13	Contraintes roles biologiste	4
1.14	Output	4
1.15	Trigger date naissance	4
1.16	Output	4
1.17	Insertion	5
1.18	Output	5
1.19	Insertion	5
1.20	Output	5
1.21	select	6
1.22	Output	6
1.23	Constraint role	7

1.24	Output	7
1.25	Constraint role	7
1.26	Output	7
1.27	Drop table	7
1.28	Output	8
1.29	Drop table	8
1.30	Output	8
1.31	Drop table	8
1.32	Output	9
1.33	Drop table	9
1.34	Output	10
2.1	Nombre biologiste	11
2.2	Output	11
2.3	Modifcations	11
2.4	Trigger	12
2.5	insertion	12
2.6	select resultat	12
2.7	Output	13
2.8	Procedure CasPositifs	13
2.9	L'appel	13
2.10	Output	13
2.11	Function Nombre prelevements	14
2.12	Function call	14
2.13	Output	14
2.14	Procedure ajouter prelevements -1-	15
2.15	Procedure ajouter prelevements -2-	15
2.16	Procedure call	16
2.17	Output	16
2.18	Trigger insert	16

2.19	Trigger update	16
2.20	Trigger suppression	17
2.21	Output	17
2.22	Trigger empêche la modification	17
2.23	Update request	18
2.24	Output	18
2.25	Add attribute	18
2.26	Output	18
2.27	Trigger T1	18
2.28	Insert request	19
2.29	Output	19
2.30	Output table biologiste	19

Chapter 1

Partie 1: Contraintes d'intégrité statiques

1.1 Les contraintes de chaque table de ce schéma

1.1.1 La table Biologiste

```
SELECT * FROM user_constraints where table_name = 'BIOLOGISTE';
```

Figure 1.1: Contraintes biologiste

	OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION
1	SYSTEM	SYS_C0029251	C	BIOLOGISTE	"NUMB" IS NOT NULL
2	SYSTEM	SYS_C0029252	C	BIOLOGISTE	"NOM" IS NOT NULL
3	SYSTEM	SYS_C0029253	C	BIOLOGISTE	"PRENOM" IS NOT NULL
4	SYSTEM	SYS_C0029254	C	BIOLOGISTE	"SPECIALITE" IS NOT NULL
5	SYSTEM	SYS_C0029255	C	BIOLOGISTE	"ROLEB" IS NOT NULL
6	SYSTEM	CHECK_SPECIALITE	C	BIOLOGISTE	Specialite in ('Microbio', 'Bio-Med', 'Ingénieur', 'Biologie', 'Technicien', 'Biologie Moleculaire')
7	SYSTEM	PK_NUMB	P	BIOLOGISTE	(null)

Figure 1.2: Output

1.1.2 La table Patient

```
SELECT * FROM user_constraints where table_name = 'PATIENT';
```

Figure 1.3: Contraintes patient

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	SEARCH_CONDITION_VC	R_OWNER	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	I
1 SYSTEM	SYS_C0029257	C	PATIENT	"NUMP" IS NOT NULL	"NUMP" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
2 SYSTEM	SYS_C0029258	C	PATIENT	"NOM" IS NOT NULL	"NOM" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
3 SYSTEM	SYS_C0029259	C	PATIENT	"PRENOM" IS NOT NULL	"PRENOM" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
4 SYSTEM	SYS_C0029260	C	PATIENT	"DATENAISSANCE" IS NOT NULL	"DATENAISSANCE" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
5 SYSTEM	FK_NUMP	P	PATIENT	(null)	(null)	(null)	(null)	(null)	ENABLED	NO

Figure 1.4: Output

1.1.3 La table Prelevement

```
SELECT * FROM user_constraints where table_name = 'PRELEVEMENT';
```

Figure 1.5: Contraintes prelevement

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	SEARCH_CONDITION_VC	R_OWNER	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	I
1 SYSTEM	SYS_C0029257	C	PATIENT	"NUMP" IS NOT NULL	"NUMP" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
2 SYSTEM	SYS_C0029258	C	PATIENT	"NOM" IS NOT NULL	"NOM" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
3 SYSTEM	SYS_C0029259	C	PATIENT	"PRENOM" IS NOT NULL	"PRENOM" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
4 SYSTEM	SYS_C0029260	C	PATIENT	"DATENAISSANCE" IS NOT NULL	"DATENAISSANCE" IS NOT NULL	(null)	(null)	(null)	ENABLED	NO
5 SYSTEM	FK_NUMP	P	PATIENT	(null)	(null)	(null)	(null)	(null)	ENABLED	NO

Figure 1.6: Output

1.1.4 La table EffectuerPrelevement

```
SELECT * FROM user_constraints where table_name = 'EFFECTUERPRELEVEMENT';
```

Figure 1.7: Contraintes effectuer prelevement

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	SEARCH_CONDITION_VC	R_OWNER	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	I
1 SYSTEM	FK_EFFECTUERPRELEVEMENT_BIOLOGISTE	R	EFFECTUERPRELEVEMENT	(null)	(null)	SYSTEM	FK_NUMB	NO ACTION		
2 SYSTEM	FK_EFFECTUERPRELEVEMENT_PATIENT	R	EFFECTUERPRELEVEMENT	(null)	(null)	SYSTEM	FK_NUMP	NO ACTION		
3 SYSTEM	FK_EFFECTUERPRELEVEMENT_PRELEVEMENT	R	EFFECTUERPRELEVEMENT	(null)	(null)	SYSTEM	FK_NUMPR	NO ACTION		
4 SYSTEM	SYS_C0029322	C	EFFECTUERPRELEVEMENT	"NUMB" IS NOT NULL	"NUMB" IS NOT NULL	(null)	(null)	(null)		
5 SYSTEM	SYS_C0029323	C	EFFECTUERPRELEVEMENT	"NUMP" IS NOT NULL	"NUMP" IS NOT NULL	(null)	(null)	(null)		
6 SYSTEM	SYS_C0029324	C	EFFECTUERPRELEVEMENT	"NUMPR" IS NOT NULL	"NUMPR" IS NOT NULL	(null)	(null)	(null)		
7 SYSTEM	FK_NUMB_NUMP_NUMPR	P	EFFECTUERPRELEVEMENT	(null)	(null)	(null)	(null)	(null)		

Figure 1.8: Output

1.1.5 La table Resultat

```
SELECT * FROM user_constraints where table_name = 'RESULTAT';
```

Figure 1.9: Contraintes Resultat

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	SEARCH_CONDITION_VC	R_OWNER	R_CONSTRAINT_NAME	DELETE_RULE	STAT
1 SYSTEM	FK_RESULTAT_PRELEVEMENT	R	RESULTAT	(null)	(null)	SYSTEM	FK_NUMPR	NO ACTION	ENABLI
2 SYSTEM	SYS_C0029337	C	RESULTAT	"NUMR" IS NOT NULL	"NUMR" IS NOT NULL	(null)	(null)	(null)	ENABLI
3 SYSTEM	SYS_C0029338	C	RESULTAT	"NUMPR" IS NOT NULL	"NUMPR" IS NOT NULL	(null)	(null)	(null)	ENABLI
4 SYSTEM	SYS_C0029339	C	RESULTAT	"TYPERESULTAT" IS NOT NULL	"TYPERESULTAT" IS NOT NULL	(null)	(null)	(null)	ENABLI
5 SYSTEM	SYS_C0029340	C	RESULTAT	"RESUL" IS NOT NULL	"RESUL" IS NOT NULL	(null)	(null)	(null)	ENABLI
6 SYSTEM	SYS_C0029341	C	RESULTAT	"NORME" IS NOT NULL	"NORME" IS NOT NULL	(null)	(null)	(null)	ENABLI
7 SYSTEM	SYS_C0029342	C	RESULTAT	"CONCLUSION" IS NOT NULL	"CONCLUSION" IS NOT NULL	(null)	(null)	(null)	ENABLI
8 SYSTEM	PK_NUMR	P	RESULTAT	(null)	(null)	(null)	(null)	(null)	ENABLI

Figure 1.10: Output

1.2 Etendre la liste des roles possibles

1.2.1 La requette des roles

```
select distinct RoleB from Biologiste;
```

Figure 1.11: Roles biologiste

	ROLEB
1	Ing-Informatique
2	Ing-Qualité
3	Aide-laboratoire
4	Technicien
5	Biologist-Médical
6	Biologiste-Responsable
7	Secrétaire

Figure 1.12: Output

1.2.2 La requête de la contrainte

```
ALTER TABLE Biologiste
ADD CONSTRAINT Check_RoleB
check (RoleB in ('Ing-Informatique','Ing-Qualité','Aide-laboratoire','Technicien','Biologist-Médical','Biologiste-Responsable','Secrétaire','B-M à Domicile'));
```

Figure 1.13: Contraintes roles biologiste

Table BIOLOGISTE modifié(e) .

Figure 1.14: Output

1.3 La date de naissance des patients antérieure à la date d'aujourd'hui. Tester avec quelques insertions.

1.3.1 Le trigger

```
create or replace NONEDITIONABLE TRIGGER trg_check_date_naissance
BEFORE INSERT OR UPDATE ON Patient
FOR EACH ROW
DECLARE
    current_date DATE;
BEGIN
    SELECT SYSDATE INTO current_date FROM DUAL;
    IF :new.DateNaissance >= trunc(current_date) THEN
        RAISE_APPLICATION_ERROR(-20001, 'La date de naissance doit être antérieure à la date actuelle.');
```

Figure 1.15: Trigger date naissance

Elément Trigger TRG_CHECK_DATE_NAISSANCE compilé

Figure 1.16: Output

1.3.2 Les insertions

```
INSERT INTO Patient VALUES(11,'BENMEZIANE','Abdelmalek',TO_DATE('17/03/2000','DD/MM/YYYY'));  
INSERT INTO Patient VALUES(12,'Doe','John',TO_DATE('18/03/2002','DD/MM/YYYY'));
```

Figure 1.17: Insertion

1 ligne inséré.

1 ligne inséré.

Figure 1.18: Output

```
INSERT INTO Patient VALUES(13,'MOHAMED','Amine',TO_DATE('18/03/2023','DD/MM/YYYY'));
```

Figure 1.19: Insertion

```
Erreur commençant à la ligne: 2 de la commande -  
INSERT INTO Patient VALUES(13,'MOHAMED','Amine',TO_DATE('18/03/2023','DD/MM/YYYY'))  
Rapport d'erreur -  
ORA-20001: La date de naissance doit être antérieure à la date actuelle.  
ORA-06512: à "SYSTEM.TRG_CHECK_DATE_NAISSANCE", ligne 6  
ORA-04088: erreur lors d'exécution du déclencheur 'SYSTEM.TRG_CHECK_DATE_NAISSANCE'
```

Figure 1.20: Output

Voilà la table Patient

```
select * from patient;
```

Figure 1.21: select

	NUMP	NOM	PRENOM	DATENAISSANCE
1	11	BENMEZIANE	Abdelmalek	17/03/2000
2	12	Doe	John	18/03/2002
3	1	SAIDI	Ryad	10/02/1970
4	2	BELHADJ	Selma	21/03/1976
5	3	DIB	Ahmed	03/08/2000
6	4	BRAHIMI	Djalil	22/06/2002
7	5	SYAD	Hadjer	14/09/1999
8	6	NAIM	Fouad	23/07/1998
9	7	KADRI	Amine	28/05/1970
10	8	SEDDIKI	Wail	20/10/1986
11	9	AITALI	Bahia	08/10/1950
12	10	SENDJAK	Raouf	02/04/1968

Figure 1.22: Output

1.4 La contrainte d'un ingénieur ne peut exercer en tant que Bio-Medical

1.4.1 La contrainte

```
ALTER TABLE Biologiste  
ADD CONSTRAINT IngBM check(not(Specialite = 'Ingénieur' and RoleB = 'Bio-Medical'));
```

Figure 1.23: Constraint role



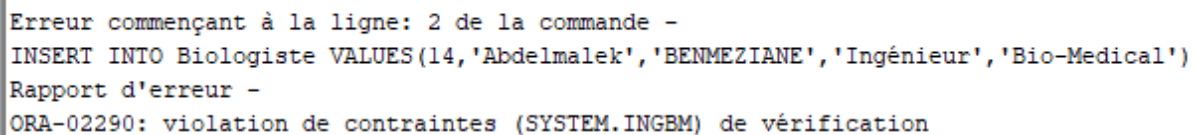
Table BIOLOGISTE modifié(e) .

Figure 1.24: Output

1.4.2 L'insertion

```
INSERT INTO Biologiste VALUES(14,'Abdelmalek','BENMEZIANE','Ingénieur','Bio-Medical');
```

Figure 1.25: Constraint role

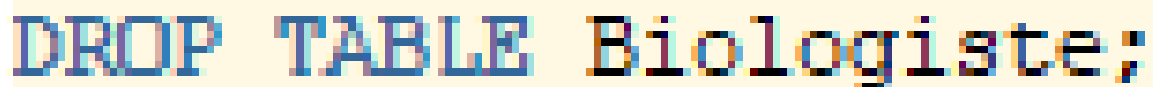


Erreur commençant à la ligne: 2 de la commande -
INSERT INTO Biologiste VALUES(14,'Abdelmalek','BENMEZIANE','Ingénieur','Bio-Medical')
Rapport d'erreur -
ORA-02290: violation de contraintes (SYSTEM.INGBM) de vérification

Figure 1.26: Output

1.5 Les modifications sur la table Biologiste

1.5.1 Suppression la table Biologiste et remarque



DROP TABLE Biologiste;

Figure 1.27: Drop table

```

Erreur commençant à la ligne: 2 de la commande -
DROP TABLE Biologiste
Rapport d'erreur -
ORA-02449: clés uniques/primaires de la table référencées par des clés étrangères
02449. 00000 - "unique/primary keys in table referenced by foreign keys"
*Cause:      An attempt was made to drop a table with unique or
              primary keys referenced by foreign keys in another table.
*Action:     Before performing the above operations the table, drop the
              foreign key constraints in other tables. You can see what
              constraints are referencing a table by issuing the following
              command:
              SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME = "tabnam";

```

Figure 1.28: Output

1.5.2 suppression la CI qui bloque la suppression table et la remarque

```

ALTER TABLE EffectuerPrelevement DROP CONSTRAINT FK_EffectuerPrelevement_Biologiste;
ALTER TABLE EffectuerPrelevement DROP CONSTRAINT PK_NumB_NumP_NumPr;

```

Figure 1.29: Drop table

```

Table EFFECTUERPRELEVEMENT modifié(e) .

Erreur commençant à la ligne: 4 de la commande -
ALTER TABLE EffectuerPrelevement DROP CONSTRAINT PK_NumB_NumP_NumPr
Rapport d'erreur -
ORA-02443: Impossible de supprimer la contrainte - contrainte inexistante
02443. 00000 - "Cannot drop constraint - nonexistent constraint"
*Cause:      alter table drop constraint <constraint_name>
*Action:     make sure you supply correct constraint name.

```

Figure 1.30: Output

```

ALTER TABLE Biologiste DROP CONSTRAINT PK_NUMB;

drop table Biologiste;

```

Figure 1.31: Drop table

```
Table BIOLOGISTE modifié(e) .
```

```
Table BIOLOGISTE supprimé(e) .
```

Figure 1.32: Output

1.5.3 Conclulsion

Pour supprimer une table dans la base de données, il faut s'assurer que cette table n'a pas des clés primaires et étrangères, sinon c'est une violation des contraintes d'intégrité.

1.5.4 Recréer la table Biologiste

```
CREATE TABLE Biologiste (  
    NumB NUMBER(4) not null,  
    Nom VARCHAR(40) not null ,  
    Prenom VARCHAR(40) not null ,  
    Specialite VARCHAR(40) not null,  
    RoleB VARCHAR(40) not null,  
    CONSTRAINT PK_NumB PRIMARY KEY (NumB)  
);
```

Figure 1.33: Drop table

Table BIOLOGISTE créé(e) .

Figure 1.34: Output

Chapter 2

Partie 2 : PL/SQL

2.1 Affichage du nombre de biologistes par role

```
DECLARE
    CURSOR c_biologiste IS SELECT RoleB, count(*) as num from Biologiste group by RoleB;
BEGIN
    FOR r_biologiste IN c_biologiste
    LOOP
        dbms_output.put_line('Il y a ' || r_biologiste.num || ' personne(s) qui exerce(nt) en tant que ' || r_biologiste.RoleB);
    END LOOP;
END;
```

Figure 2.1: Nombre biologiste

```
Il y a 1 personne(s) qui exerce(nt) en tant que Ing-Informatique
Il y a 1 personne(s) qui exerce(nt) en tant que Ing-Qualité
Il y a 2 personne(s) qui exerce(nt) en tant que Aide-laboratoire
Il y a 3 personne(s) qui exerce(nt) en tant que Biologist-Médical
Il y a 1 personne(s) qui exerce(nt) en tant que Technicien
Il y a 1 personne(s) qui exerce(nt) en tant que Biologiste-Responsable
Il y a 1 personne(s) qui exerce(nt) en tant que Secrétaire
```

Figure 2.2: Output

2.2 Le test antigénique Covid positif

2.2.1 Les modifications

```
-- Les modifications :
update resultat set Norme = '>0.8' where TypeResultat = 'Antig-Covid';
update resultat set Conclusion = 'Positif' where Resul >= '0.8' and typeresultat = 'Antig-Covid';
```

Figure 2.3: Modifications

2.2.2 Trigger

```
-- Trigger
CREATE OR REPLACE TRIGGER antig_covid_norme
BEFORE INSERT OR UPDATE ON Resultat
FOR EACH ROW
BEGIN

    if (:New.type resultat = 'Antig-Covid')
    then
        if (:New.resul = '0.8')
            then :New.conclusion := 'Positif';
            else :New.conclusion := 'Négatif';
        end if;

        :New.norme := '>0.8';
    end if;
END;
```

Figure 2.4: Trigger

2.2.3 La vérification

```
-- La vérification :
insert into Resultat VALUES(15,8,'Antig-Covid','0.4','>0.8','Négatif');
```

Figure 2.5: insertion

```
select * from resultat;
```

Figure 2.6: select resultat

	NUMR	NUMPR	TYPESRESULTAT	RESUL	NORME	CONCLUSION
1	15	8	Antig-Covid	0.4	>0.8	Négatif
2	1	1	Hémoglobine	10.2	12 à 16g/dL	Anémie
3	2	1	Plaquettes	155k	150k à 400k/mm3	Sans Particularité
4	3	1	Leucocytes	6.2	4k à 10k/mm3	Sans Particularité.
5	4	1	Lymphocytes	4.8	1.5k à 4k/mm3	Poss. Infection
6	5	2	Antig-Covid	0.2	>0.8	Négatif
7	6	3	Groupage	A R+	A, B, AB, O -+	A+
8	7	4	Culture	Staphyl.	-	Infection au Staphylococcus
9	8	4	Sens. Antibiotique	+Amoxicilline	-	Sensible à l'Amoxicilline
10	9	5	Hémoglobine	13.2	12 à 16g/dL	Sans Particularité.
11	10	5	Plaquettes	235k	150k à 400k/mm3	Sans Particularité.
12	11	5	Leucocytes	8.1	4k à 10k/mm3	Sans Particularité.
13	12	5	Lymphocytes	2.8	1.5k à 4k/mm3	Sans Particularité.
14	13	6	Antig-Covid	12.6	>0.8	Positif
15	14	7	PCR Covid	8.2	>0.5	Positif

Figure 2.7: Output

2.3 Procédure CasPositifs

```

create or replace PROCEDURE CasPositifs IS
  CURSOR c_caspositifs IS select pa.numpr, pa.nom, pa.prenom, r.typeresultat, r.conclusion, p.typepr from patient pa, Resultat r, prelevement p
  where r.numpr = p.numpr and p.numpr = pa.numpr and r.typeresultat like '%Covid'
  and r.conclusion like 'Positif'
  group by pa.numpr, pa.nom, pa.prenom, r.typeresultat, r.conclusion, p.typepr;
BEGIN
  FOR r_caspositif IN c_caspositifs
  LOOP
    dbms_output.put_line('Le patient ' || r_caspositif.numpr || ' est testé ' || r_caspositif.conclusion);
  END LOOP;
END;
```

Figure 2.8: Procedure CasPositifs

```
EXEC caspositifs();
```

Figure 2.9: L'appel

```

Le patient 4 est testé Positif
Le patient 5 est testé Positif
```

Figure 2.10: Output

2.4 Fonction nombre de prelevements

```
create or replace FUNCTION Biologiste_nbr_prelevement (id integer)
RETURN integer
IS
    idb number;
    nbr_prelevements integer := 0;
    cursor c_biologiste is select b.numb,b.nom, count(*) as nbr_prelevements from Biologiste b ,prelevement p,effectuerPrelevement ep
    where b.numb = ep.numb and p.numpr = ep.numpr group by b.numb,b.nom;
BEGIN
    for r_biologiste in c_biologiste
    loop
        if (r_biologiste.Numb = id)
        then dbms_output.put_line('Le biologiste ' || r_biologiste.nom || ' a effectué ' || r_biologiste.nbr_prelevements);
            nbr_prelevements := r_biologiste.nbr_prelevements;
        end if;
    end loop;
    return nbr_prelevements;
END;
```

Figure 2.11: Function Nombre prelevements

```
SELECT Biologiste_nbr_prelevement(7) FROM Dual;
```

Figure 2.12: Function call

```
Le biologiste BOUSALEM a effectué 3
```

Figure 2.13: Output

2.5 Procédure ajouter un prélèvement

```
CREATE OR REPLACE PROCEDURE AjouterPrelevement(
    p_NumPr IN Prelevement.NumPr%TYPE,
    p_NumP IN Prelevement.NumP%TYPE,
    p_DatePr IN Prelevement.DatePr%TYPE,
    p_TypePr IN Prelevement.TypePr%TYPE,
    p_NumB IN Biologiste.NumB%TYPE
) AS
    v_count NUMBER;
BEGIN
    -- Vérifier si le prélèvement existe déjà
    SELECT COUNT(*) INTO v_count FROM Prelevement WHERE NumPr = p_NumPr;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Le numéro de prélèvement existe déjà.');
```

```
    END IF;

    -- Vérifier si le patient existe
    SELECT COUNT(*) INTO v_count FROM Patient WHERE NumP = p_NumP;
    IF v_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Le numéro de patient n''existe pas.');
```

```
    END IF;

    -- Vérifier si le biologiste existe
    SELECT COUNT(*) INTO v_count FROM Biologiste WHERE NumB = p_NumB;
    IF v_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Le numéro de biologiste n''existe pas.');
```

```
    END IF;
```

Figure 2.14: Procédure ajouter prelevements -1-

```
    IF v_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Le numéro de biologiste n''existe pas.');
```

```
    END IF;

    -- Ajouter le prélèvement
    INSERT INTO Prelevement(NumPr, NumP, DatePr, TypePr) VALUES(p_NumPr, p_NumP, p_DatePr, p_TypePr);

    -- Ajouter l'association entre le prélèvement et le biologiste
    INSERT INTO EffectuerPrelevement(NumB, NumP, NumPr) VALUES(p_NumB, p_NumP, p_NumPr);

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Le prélèvement a été ajouté avec succès.');
```

```
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Erreur : ' || SQLCODE || ' - ' || SQLERRM);
END;
```

Figure 2.15: Procédure ajouter prelevements -2-

```
EXEC AjouterPrelevement(14, 11, TO_DATE('01/01/2023', 'DD/MM/YYYY'), 'Nasopharyngé', 1);
```

Figure 2.16: Procedure call

```
Le prélèvement a été ajouté avec succès.
```

Figure 2.17: Output

2.6 Trigger affiche un message

2.6.1 Après insertion

```
-- Trigger pour afficher un message après l'insertion d'un patient
CREATE OR REPLACE TRIGGER insert_patient_trigger
AFTER INSERT ON Patient
FOR EACH ROW
BEGIN
    dbms_output.put_line('Un nouveau patient est ajouté');
END;
```

Figure 2.18: Trigger insert

2.6.2 Après modification

```
-- Trigger pour afficher un message après la modification d'un patient
CREATE OR REPLACE TRIGGER update_patient_trigger
AFTER UPDATE ON Patient
FOR EACH ROW
BEGIN
    dbms_output.put_line('Un patient a été modifié');
END;
```

Figure 2.19: Trigger update

2.6.3 Après suppression

```
-- Trigger pour afficher un message après la suppression d'un patient
CREATE OR REPLACE TRIGGER delete_patient_trigger
AFTER DELETE ON Patient
FOR EACH ROW
BEGIN
    dbms_output.put_line('Un patient a été supprimé');
END;
```

Figure 2.20: Trigger suppression

```
Elément Trigger INSERT_PATIENT_TRIGGER compilé

Elément Trigger UPDATE_PATIENT_TRIGGER compilé

Elément Trigger DELETE_PATIENT_TRIGGER compilé
```

Figure 2.21: Output

2.7 Trigger empêche la modification du numéro de biologiste

```
CREATE OR REPLACE TRIGGER prevent_biologist_number_update
BEFORE UPDATE ON EffectuerPrelevement
FOR EACH ROW
BEGIN
    IF :OLD.NumB != :NEW.NumB THEN
        RAISE_APPLICATION_ERROR(-20001, 'Le numéro de biologiste ne peut pas être modifié');
    END IF;
END;
```

Figure 2.22: Trigger empêche la modification

On essaie cette requete :

```
update EffectuerPrelevement set numb = 15 where numb = 1;
```

Figure 2.23: Update request

On aura :

```
Erreur commençant à la ligne: 14 de la commande -
update EffectuerPrelevement set numb = 15 where numb = 1
Rapport d'erreur -
ORA-20001: Le numéro de biologiste ne peut pas être modifié
ORA-06512: à "SYSTEM.PREVENT_BIOLOGIST_NUMBER_UPDATE", ligne 3
ORA-04088: erreur lors d'exécution du déclencheur 'SYSTEM.PREVENT_BIOLOGIST_NUMBER_UPDATE'
```

Figure 2.24: Output

2.8 Trigger T1 de le màj automatique de l'attribut Nb_Pr

2.8.1 L'ajout de l'attribut Nb_Pr

```
ALTER TABLE Biologiste ADD Nb_Pr NUMBER(4) DEFAULT 0;
```

Figure 2.25: Add attribute

Table BIOLOGISTE modifié(e) .

Ici, il fallait ajouter aussi les cas de DELETE et UPDATE comme suit :

create or replace Trigger T1

After Insert or Delete or Update on EffectuePrelevement

For each row

begin 2.8.2 Trigger T1

if inserting then

update biologiste set Nb_Pr = Nb_Pr + 1 where Numb = :new.numb;

end if;

if deleting then

update biologiste set Nb_pr = Nb_Pr - 1 where Numb = :old.numb;

end if;

if updating the

update biologiste set Nb_Pr = Nb_Pr + 1 where Numb = :new.numb;

update biologiste set Nb_pr = Nb_Pr - 1 where Numb = :old.numb ;

end if;

End;

Figure 2.26: Output

Figure 2.27: Trigger T1

On essaie cette requete :

```
insert into EffectuerPrelevement VALUES (2,11,14);
```

Figure 2.28: Insert request

On aura :

1 ligne inséré.

Figure 2.29: Output

si on consulte la table Biologiste, on aura :

	NUMB	NOM	PRENOM	SPECIALITE	ROLEB	NB_PR
1	1	BADI	Salim	Microbio	Biologiste-Responsable	0
2	2	AMRAN	Zineb	Bio-Med	Biologist-Médical	1
3	3	SAHLI	Lamia	Ingénieur	Ing-Qualité	0
4	4	NADIR	Ahmed	Biologie	Aide-laboratoire	0
5	5	BENMIHOUB	Djamila	Ingénieur	Secrétaire	0
6	6	CHERGUI	Selma	Technicien	Technicien	0
7	7	BOUSALEM	Ziad	Biologie	Aide-laboratoire	0
8	8	KADI	Nadia	Ingénieur	Ing-Informatique	0
9	9	SMATI	Radia	Bio-Med	Biologist-Médical	0
10	10	NAILI	Mourad	Bio-Med	Biologist-Médical	0

Figure 2.30: Output table biologiste