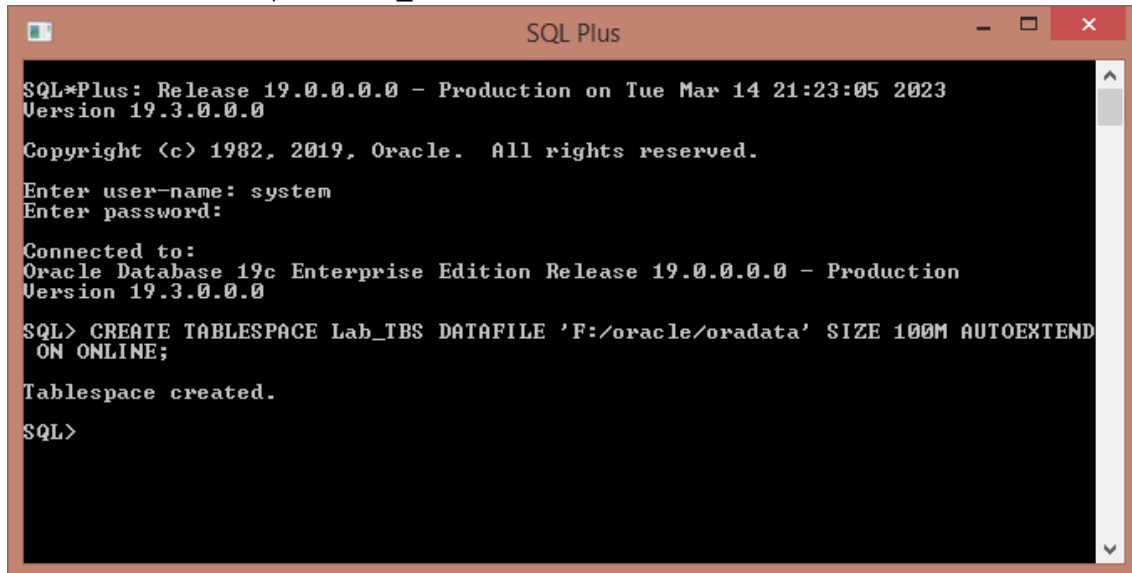


## TD/TP1 : Création, Manipulation et Interrogation des Données

Partie 1: Préparation et création de la structure

## 1. Créer deux TableSpaces Lab\_TBS



```
SQL*Plus: Release 19.0.0.0.0 - Production on Tue Mar 14 21:23:05 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter user-name: system
Enter password:

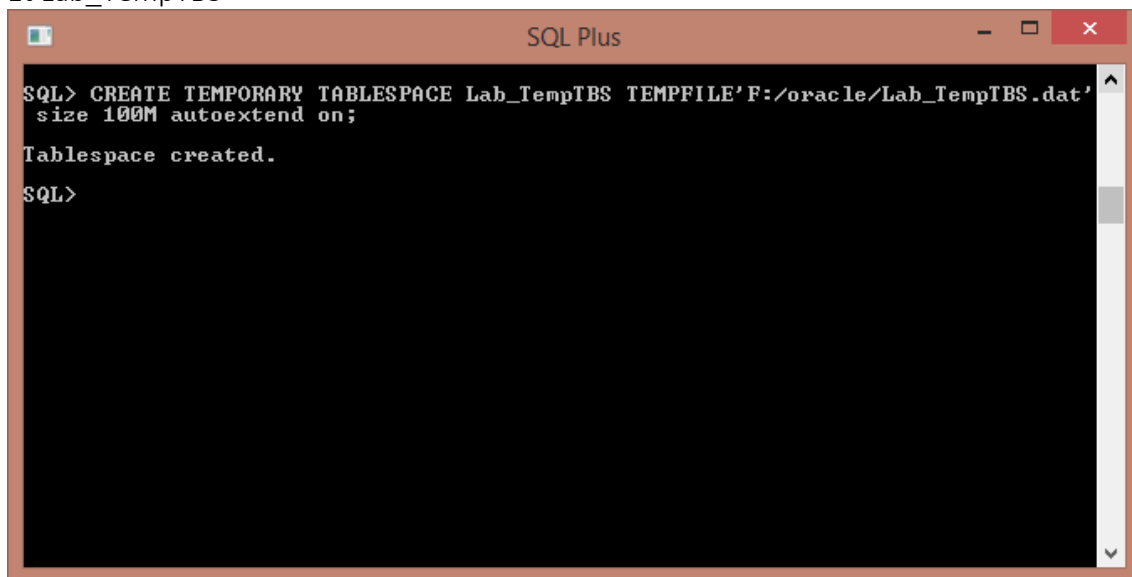
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> CREATE TABLESPACE Lab_TBS DATAFILE 'F:/oracle/oradata' SIZE 100M AUTOEXTEND
ON ONLINE;

Tablespace created.

SQL>
```

## Et Lab\_TempTBS

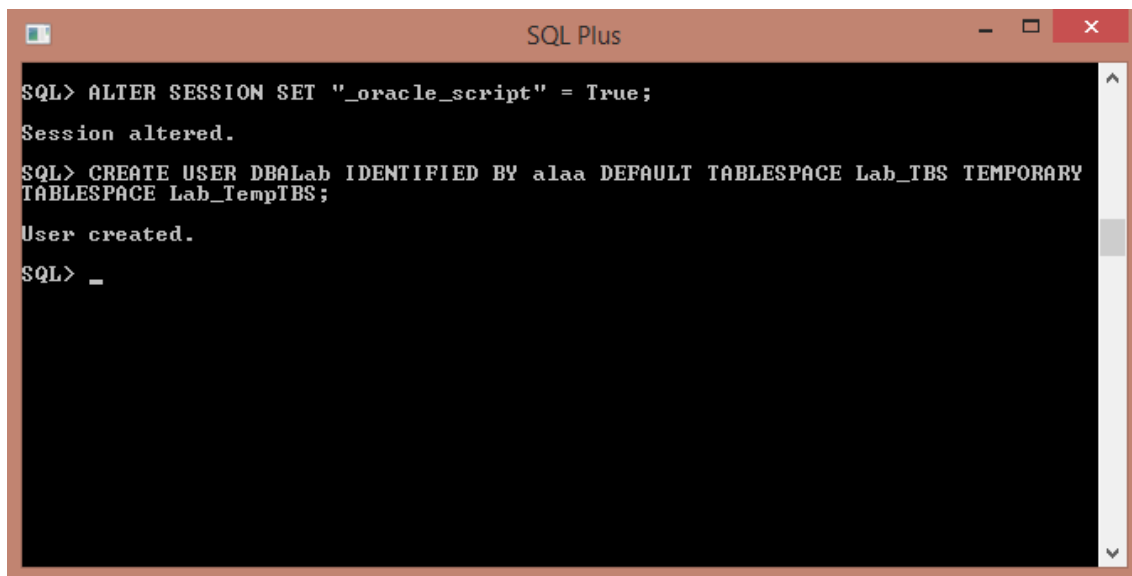


```
SQL> CREATE TEMPORARY TABLESPACE Lab_TempTBS TEMPFILE 'F:/oracle/Lab_TempTBS.dat'
size 100M autoextend on;

Tablespace created.

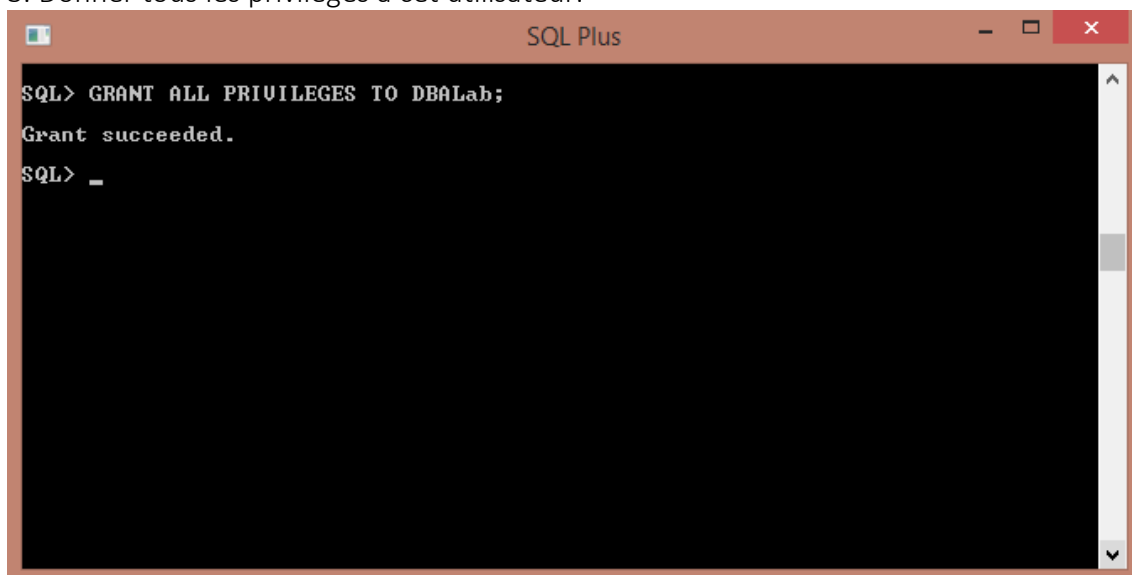
SQL>
```

2. Créer un utilisateur DBALab en lui attribuant les deux tablespaces créés précédemment



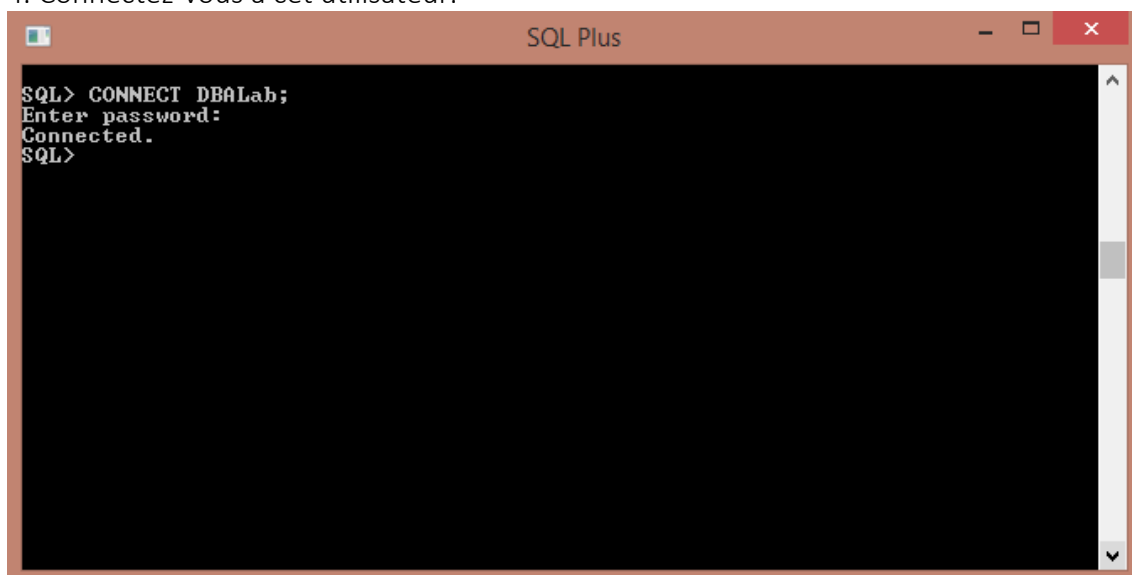
```
SQL> ALTER SESSION SET "_oracle_script" = True;
Session altered.
SQL> CREATE USER DBALab IDENTIFIED BY alaa DEFAULT TABLESPACE Lab_TBS TEMPORARY
TABLESPACE Lab_TempTBS;
User created.
SQL> _
```

3. Donner tous les privilèges à cet utilisateur.



```
SQL> GRANT ALL PRIVILEGES TO DBALab;
Grant succeeded.
SQL> _
```

4. Connectez-vous à cet utilisateur.



```
SQL> CONNECT DBALab;
Enter password:
Connected.
SQL>
```

5. Créer l'ensemble des tables du schéma dans le compte de cet utilisateur. Précisez les clés primaires et secondaires en attribuant des noms aux contraintes à chaque fois.

```
SQL Plus
SQL> CREATE TABLE Biologiste <
2     NumB NUMBER(2),
3     Nom VARCHAR(25) NOT NULL,
4     Prenom VARCHAR(25) NOT NULL,
5     Specialite VARCHAR(25),
6     RoleB VARCHAR(25),
7     CONSTRAINT PK1 PRIMARY KEY<NumB>
8 >;

Table created.
SQL>
```

```
SQL Plus
SQL> CREATE TABLE Patient <
2     NumP INT,
3     Nom VARCHAR(25) NOT NULL,
4     Prenom VARCHAR(25) NOT NULL,
5     DateNaissance DATE,
6     CONSTRAINT PK2 PRIMARY KEY<NumP>
7 >;

Table created.
SQL>
```

```
SQL Plus
SQL> CREATE TABLE Prelevement <
2     NumPr INT,
3     NumP INT NOT NULL,
4     DatePr DATE NOT NULL,
5     TypePr VARCHAR(25) NOT NULL,
6     CONSTRAINT FK_NumP FOREIGN KEY <NumP> REFERENCES Patient<NumP>,
7     CONSTRAINT PK3 PRIMARY KEY<NumPr>
8 >;

Table created.
SQL> _
```

```
SQL Plus
SQL> CREATE TABLE EffectuePrelevement (
2     NumB INT NOT NULL,
3     NumP INT NOT NULL,
4     NumPr INT NOT NULL,
5     CONSTRAINT PK_4 PRIMARY KEY (NumB, NumP, NumPr),
6     CONSTRAINT FK_NumB FOREIGN KEY (NumB) REFERENCES Biologiste(NumB),
7     CONSTRAINT FK_NumP2 FOREIGN KEY (NumP) REFERENCES Patient(NumP),
8     CONSTRAINT FK_NumPr FOREIGN KEY (NumPr) REFERENCES Prelevement(NumPr)
9 );

Table created.

SQL> _
```

```
SQL Plus
SQL> CREATE TABLE Resultat (
2     NumR INT,
3     NumPr INT NOT NULL,
4     TypePr VARCHAR(25) NOT NULL,
5     Resul VARCHAR(25),
6     Norme VARCHAR(25),
7     Conclusion VARCHAR(25),
8     CONSTRAINT FK_NumPr2 FOREIGN KEY (NumPr) REFERENCES Prelevement(NumPr),
9     CONSTRAINT PK4 PRIMARY KEY(NumR)
10 );

Table created.

SQL> _
```

## Partie 2 : Remplissage et modification

1. Remplir les tables avec les informations fournies dans les tableaux.

-- Creation de la table Biologiste

```
SQL Plus

1 row created.
SQL> INSERT INTO Biologiste VALUES (6, 'CHERGUI', 'Selma', 'Technicien', 'Technicien');
1 row created.
SQL> INSERT INTO Biologiste VALUES (7, 'BOUSALEM', 'Ziad', 'Biologie', 'Aide-laboratoire');
1 row created.
SQL> INSERT INTO Biologiste VALUES (8, 'KADI', 'Nadia', 'Ingenieur', 'Ing-Informatique');
1 row created.
SQL> INSERT INTO Biologiste VALUES (9, 'SMATI', 'Radia', 'Bio-Med', 'Biologist-Medical');
1 row created.
SQL> INSERT INTO Biologiste VALUES (10, 'NAILI', 'Mourad', 'Bio-Med', 'Biologist-Medical');
```

-- Creation de la table Patient

```
SQL Plus

1 row created.
SQL> INSERT INTO Patient VALUES (10, 'SENDJAK', 'Raouf', TO_DATE('02/04/1968','DD/MM/YYYY'));
1 row created.
SQL> select * from patient;

  NUMP  NOM          PRENOM          DATENAISS
-----
    1 SAIDI          Ryad            10-FEB-70
    2 BELHADJ        Selma           21-MAR-76
    3 DIB             Ahmed           03-AUG-00
    4 BRAHIMI         Djalil          22-JUN-02
    5 SYAD            Hadjer          14-SEP-99
    6 NAIM            Fouad           23-JUL-98
    7 KADRI           Amine           28-MAY-70
    8 SEDDIKI         Wail            20-OCT-86
    9 AITALI          Bahia           08-OCT-50
   10 SENDJAK        Raouf           02-APR-68

10 rows selected.
SQL>
```

-- Creation de la table Prelevement

```
SQL Plus

1 row created.
SQL> select * from prelevement;

  NUMPR  NUMP  DATEPR  TYPEPR
-----
    1      1  04-FEB-22 Sanguin
    2      1  04-FEB-22 Nasopharynge
    3      2  04-FEB-22 Sanguin
    4      3  04-FEB-22 Cutaneo-Muqueux
    5      3  04-FEB-22 Sanguin
    6      4  04-FEB-22 Nasopharynge
    7      5  05-FEB-22 Sanguin
    8      6  05-FEB-22 Sanguin
    9      7  05-FEB-22 Nasopharynge
   10      8  05-FEB-22 Cutaneo-Muqueux
   11      8  05-FEB-22 Sanguin

  NUMPR  NUMP  DATEPR  TYPEPR
-----
   12      9  05-FEB-22 Sanguin
   13     10  06-FEB-22 Sanguin

13 rows selected.
```

-- Creation de la table EffectuePrelevement

```
SQL Plus
SQL> INSERT INTO EffectuePrelevement VALUES (7, 10, 13);
1 row created.
SQL> select * from effectueprelevement;
```

NUMB	NUMP	NUMPR
1	1	2
1	7	9
1	8	11
2	2	3
2	4	6
2	9	12
4	3	4
4	5	7
7	1	1
7	6	8
7	10	13
9	3	5
9	5	7

-- Creation de la table Resultat

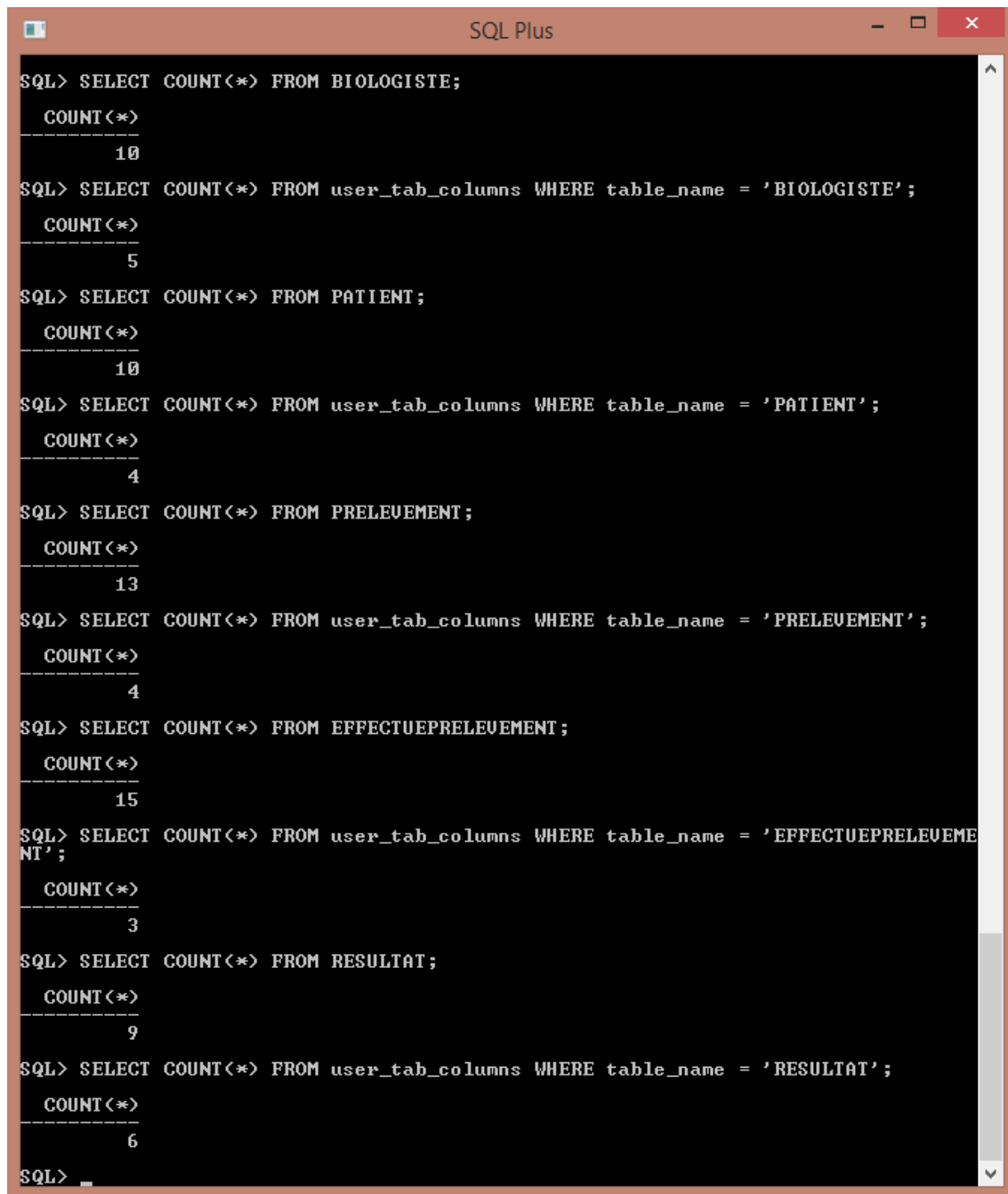
```
SQL Plus
1 row created.
SQL> INSERT INTO Resultat VALUES (13, 6, 'Antig-Covid', 12.6, '>0.5', 'Positif');
1 row created.
SQL> INSERT INTO Resultat VALUES (14, 7, 'PCR Covid', 8.2, '>0.5', 'Positif');
1 row created.
SQL> select * from resultat;
```

NUMR	NUMPR	TYPEPR	RESUL
NORME		CONCLUSION	
1 12 à 16g/dL	1	Hemoglobine Anemie	10.2
3 4k à 10k/mm3	1	Leucocytes Sans Particularite.	6.2
4 1.5k à 4k/mm3	1	Lymphocytes Poss. Infection	4.8

2. Montrer la cardinalité et le degré de chaque table.

Cardinalite : `SELECT COUNT(*) FROM table_name;`

Degree : `SELECT COUNT(*) FROM user_tab_columns WHERE table_name = 'table_name';`



```
SQL> SELECT COUNT(*) FROM BIOLOGISTE;
COUNT(*)
-----
      10

SQL> SELECT COUNT(*) FROM user_tab_columns WHERE table_name = 'BIOLOGISTE';
COUNT(*)
-----
        5

SQL> SELECT COUNT(*) FROM PATIENT;
COUNT(*)
-----
      10

SQL> SELECT COUNT(*) FROM user_tab_columns WHERE table_name = 'PATIENT';
COUNT(*)
-----
        4

SQL> SELECT COUNT(*) FROM PRELEVEMENT;
COUNT(*)
-----
      13

SQL> SELECT COUNT(*) FROM user_tab_columns WHERE table_name = 'PRELEVEMENT';
COUNT(*)
-----
        4

SQL> SELECT COUNT(*) FROM EFFECTUEPRELEVEMENT;
COUNT(*)
-----
      15

SQL> SELECT COUNT(*) FROM user_tab_columns WHERE table_name = 'EFFECTUEPRELEVEMENT';
COUNT(*)
-----
        3

SQL> SELECT COUNT(*) FROM RESULTAT;
COUNT(*)
-----
        9

SQL> SELECT COUNT(*) FROM user_tab_columns WHERE table_name = 'RESULTAT';
COUNT(*)
-----
        6

SQL>
```

3. Renommer l'attribut " TypeRes" à "TypeResultat"

```
SQL Plus
SQL> ALTER TABLE Resultat RENAME COLUMN TypePr TO TypeResultat;
Table altered.
SQL> SELECT * from Resultat;
```

NUMR	NUMPR	TYPERESULTAT	RESUL
NORME		CONCLUSION	
12 à 16g/dL	1	Hemoglobine Anemie	10.2
4k à 10k/mm3	3	1 Leucocytes Sans Particularite.	6.2
1.5k à 4k/mm3	4	1 Lymphocytes Poss. Infection	4.8
NORME		CONCLUSION	
5	2	Antig-Covid	.2

4. Agrandir la taille du champs "Conclusion" à 100 caractères.

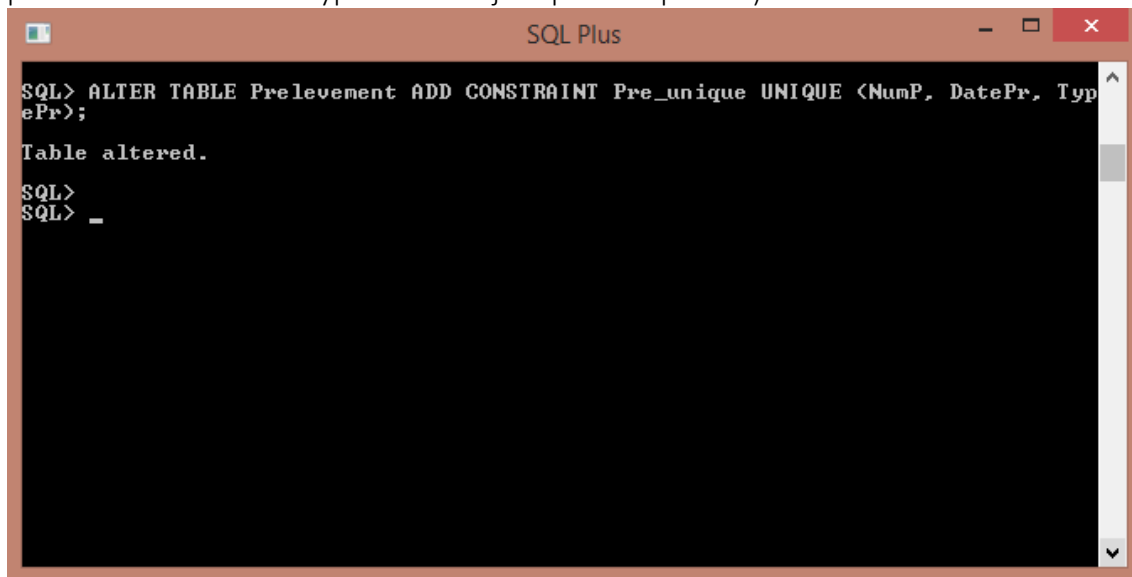
```
SQL Plus
Connected.
SQL> ALTER TABLE Resultat MODIFY Conclusion VARCHAR(100);
Table altered.
SQL> DESC RESULTAT;
```

Name	Null?	Type
NUMR	NOT NULL	NUMBER(38)
NUMPR	NOT NULL	NUMBER(38)
TYPERESULTAT	NOT NULL	VARCHAR2(25)
RESUL		VARCHAR2(25)
NORME		VARCHAR2(25)
CONCLUSION		VARCHAR2(100)



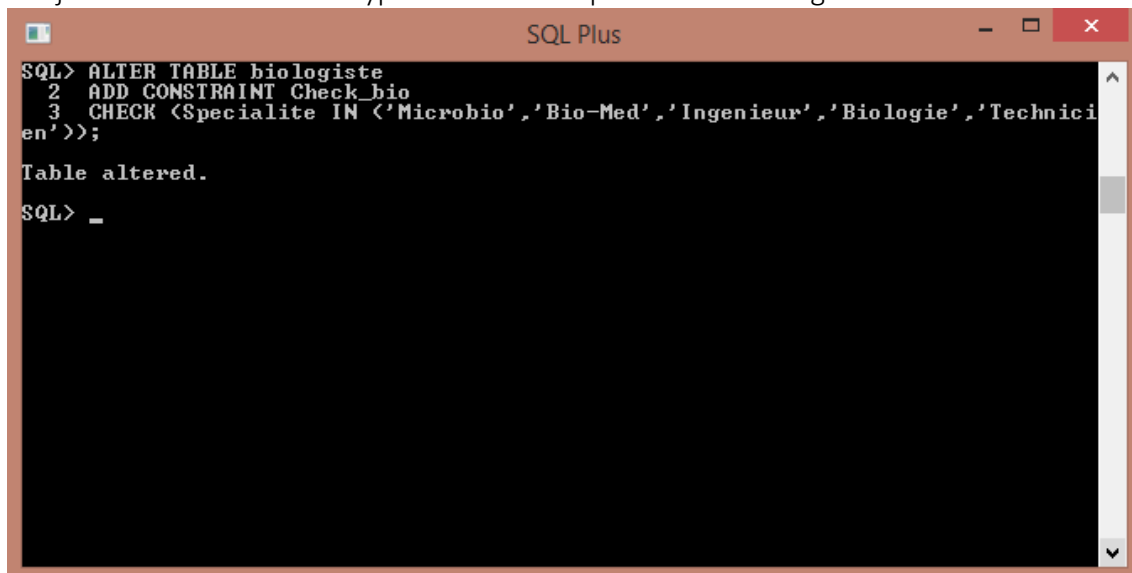
## Partie 3 : Contraintes d'intégrité

1. Exigez que le type de prélèvement soit unique pour un patient donné à une date donnée. (Pas deux prélèvements du même type le même jour pour un patient).



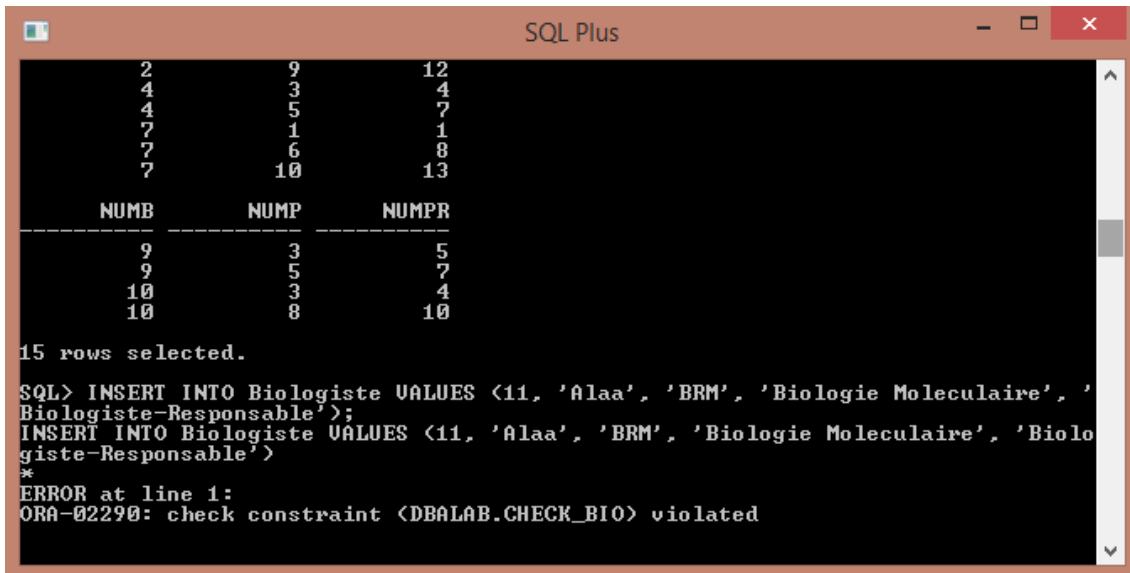
```
SQL> ALTER TABLE Prelevement ADD CONSTRAINT Pre_unique UNIQUE (NumP, DatePr, TypePr);
Table altered.
SQL>
SQL> _
```

2. Ajoutez la contrainte de type check sur la Spécialité du Biologiste.



```
SQL> ALTER TABLE biologiste
2 ADD CONSTRAINT Check_bio
3 CHECK (Specialite IN ('Microbio', 'Bio-Med', 'Ingenieur', 'Biologie', 'Technicien'));
Table altered.
SQL> _
```

3. Insérer un nouveau Biologiste avec comme spécialité : 'Biologie Moléculaire'. Que remarquez-vous ? Justifier.



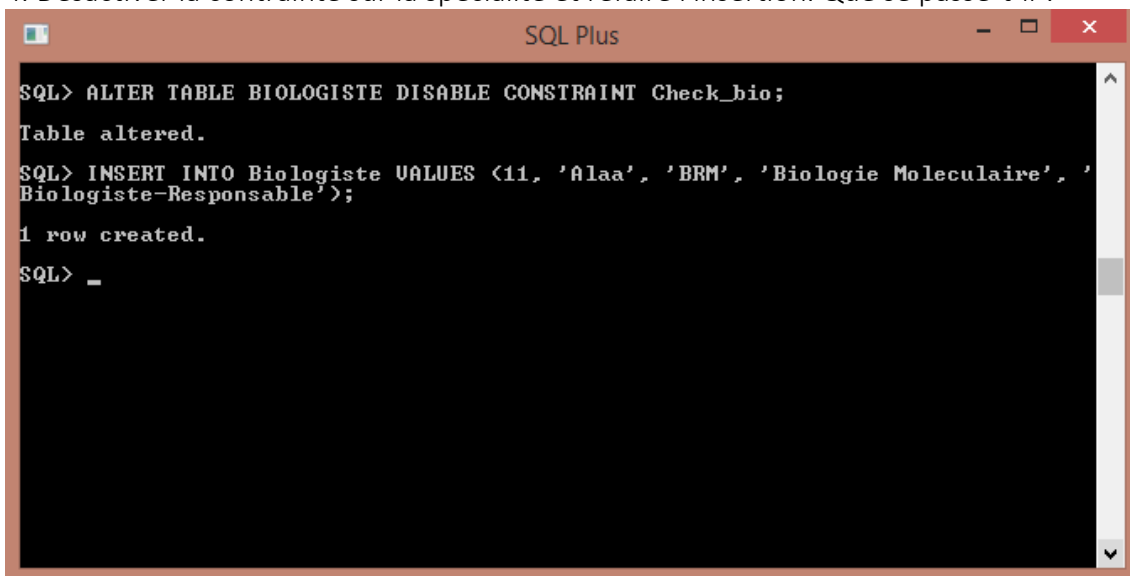
The screenshot shows a SQL Plus window with a query result and an error message. The query result displays three columns: NUMB, NUMP, and NUMPR. The data is as follows:

NUMB	NUMP	NUMPR
2	9	12
4	3	4
4	5	7
7	1	1
7	6	8
7	10	13

Below the table, it says "15 rows selected." followed by two INSERT statements. The first statement is successful, but the second one fails with the error: "ERROR at line 1: ORA-02290: check constraint (DBALAB.CHECK\_BIO) violated".

L'insertion échoue et renvoie une erreur car la valeur 'Biologie Moléculaire' ne respecte pas la contrainte CHECK ajoutée précédemment. En effet, la contrainte autorise uniquement les valeurs ('Microbio','Bio-Med','Ingenieur','Biologie','Technicien') pour le champ Spécialité.

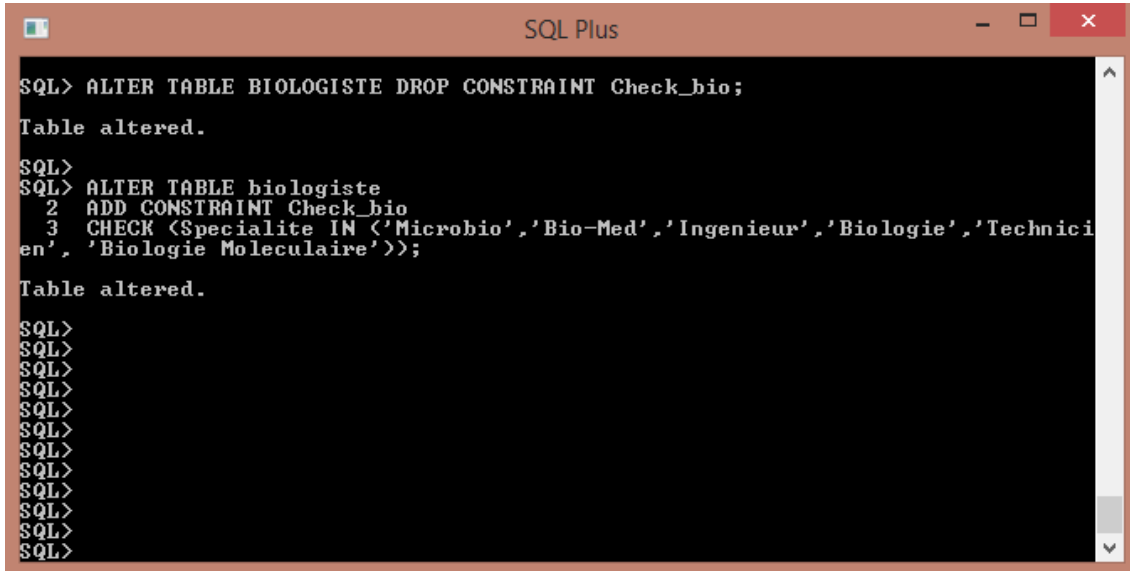
4. Désactiver la contrainte sur la spécialité et refaire l'insertion. Que se passe-t-il ?



The screenshot shows a SQL Plus window with two SQL commands. The first command is "ALTER TABLE BIOLOGISTE DISABLE CONSTRAINT Check\_bio;" which returns "Table altered.". The second command is "INSERT INTO Biologiste VALUES (11, 'Alaa', 'BRM', 'Biologie Moléculaire', 'Biologiste-Responsable');" which returns "1 row created.".

L'insertion a réussi sans erreur car la contrainte qui avait empêché l'insertion a été désactivée.

5. Réactiver l'ancienne contrainte et modifiez-la pour intégrer la nouvelle spécialité.

A screenshot of an SQL Plus window titled "SQL Plus". The window has a dark background with white text. The text shows the following commands and their outputs:

```
SQL> ALTER TABLE BILOGISTE DROP CONSTRAINT Check_bio;
Table altered.

SQL>
SQL> ALTER TABLE biologiste
2  ADD CONSTRAINT Check_bio
3  CHECK (Specialite IN ('Microbio', 'Bio-Med', 'Ingenieur', 'Biologie', 'Technici
en', 'Biologie Moleculaire')));
Table altered.

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

6. En cas de licenciements, quelle serait la différence entre la suppression de SAHLI et celle de NADIR ? Justifier et proposer la démarche à suivre pour le faire.

Si on supprime le biologiste SAHLI de la base de données, toutes les prélèvements qui ont été effectuées par lui seront perdues car il est lié à la clé étrangère dans la table EffectuePrelevement.

En revanche, si on supprime NADIR, cela n'affectera pas les données dans la table EffectuePrelevement, car il n'a pas réalisé des prélèvements et il n'y a pas de lien entre lui et cette table.

Pour supprimer SAHLI de la base de données, il faut d'abord supprimer toutes les prélèvements qu'il a effectuées en utilisant une requête qui supprime les enregistrements liés à lui dans cette table avant de le supprimer de la table Biologiste.

```
DELETE FROM EffectuePrelevement WHERE NumB = (SELECT NumB FROM Biologiste WHERE Nom = 'SAHLI');
DELETE FROM Biologiste WHERE Nom = 'SAHLI';
```

En revanche, si l'on doit procéder au licenciement de NADIR, il suffira simplement de supprimer l'enregistrement correspondant dans la table Biologiste :

```
DELETE FROM Biologiste WHERE nom_biologiste = 'NADIR';
```

7. Peut-on supprimer la table Biologiste ? justifier et proposer la démarche à suivre pour le faire.

La table "Biologiste" possède des contraintes de clé étrangère dans la table "EffectuePrelevement". Par conséquent, il n'est pas possible de la supprimer tant que les contraintes de clé étrangère dans la table "EffectuePrelevement" qui font référence à la table "Biologiste" n'ont pas été supprimées.

Pour supprimer la table "Biologiste", il faut d'abord supprimer les contraintes de clé étrangère.

```
ALTER TABLE EffectuePrelevement DROP CONSTRAINT FK_NumB;  
ALTER TABLE EffectuePrelevement DROP CONSTRAINT PK_4 (de NumB);
```

Après avoir supprimé les contraintes de clé étrangère, on peut supprimer la table "Biologiste" avec un DROP.

```
DROP TABLE Biologiste;
```

## Partie 4 : Interrogation des données

1. Quels sont les Biologistes qui participent dans le plus de prélèvements ?

```
SQL Plus
Connected.
SQL> SELECT B.NumB, Nom, COUNT(*)
  2 FROM EffectuePrelevement E, Biologiste B
  3 WHERE E.NumB = B.NumB
  4 GROUP BY B.NumB, Nom
  5 HAVING COUNT (NumPr) = (SELECT MAX(n) FROM (SELECT NumB, COUNT(*) as n FROM
EffectuePrelevement GROUP BY NumB));
```

NUMB	NOM	COUNT(*)
2	AMRAN	3
1	BADI	3
7	BOUSALEM	3

```
SQL>
```

Le moins de prélèvements?

```
SQL Plus
SQL> SELECT B.NumB, Nom, COUNT(*)
  2 FROM EffectuePrelevement E, Biologiste B
  3 WHERE E.NumB = B.NumB
  4 GROUP BY B.NumB, Nom
  5 HAVING COUNT (NumPr) = (SELECT MIN(n) FROM (SELECT NumB, COUNT(*) as n FROM
EffectuePrelevement GROUP BY NumB));
```

NUMB	NOM	COUNT(*)
10	MAILI	2
4	NADIR	2
9	SMATI	2

```
SQL> _
```

2. Quel est le nombre de test COVID Positifs, en précisant le type de prélèvement.

```
SQL Plus
SQL> SELECT TypePr, COUNT(NumR)
  2 FROM Resultat R, Prelevement P
  3 WHERE R.NumPr = P.NumPr and TypeResultat LIKE '%Covid%' and Conclusion = 'P
ositif'
  4 GROUP BY TypePr;
```

TYPEPR	COUNT(NUMR)
Nasopharynge	1
Sanguin	1

```
SQL>
```

3. Quel est l'âge des patients testés positifs au COVID-19 en Février ?

```
SQL Plus
SQL> SELECT P.nom, EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM P.DateNaissanc
e) as Age
2 FROM Patient P, Prelevement Pr, Resultat R
3 WHERE P.NumP = Pr.NumP
4 AND Pr.NumPr = R.NumPr
5 AND R.TypeResultat LIKE '%Covid%'
6 AND R.Conclusion = 'Positif'
7 AND Pr.DatePr <= TO_DATE('28/02/2022','DD/MM/YYYY') AND Pr.DatePr >= TO_DAT
E('01/02/2022','DD/MM/YYYY');

NOM                                AGE
-----
BRAHIMI                            21
SYAD                                24
SQL> _
```

4. Lister les types de prélèvements effectués par ce laboratoire.

```
SQL Plus
SQL> SELECT DISTINCT TypePr FROM Prelevement;

TYPEPR
-----
Cutaneo-Muqueux
Nasopharynge
Sanguin
SQL>
```