

Rapport Tp 1 BDD :

Data base :

Biologiste (NumB, Nom, Prenom, Spécialité, RoleB)
Patient (NumP, Nom, Prenom, DateNaissance)
Prelevement (NumPr, NumP, DatePr, TypePr)
EffectuePrelevement (NumB, NumP*, NumPr*)
Resultat (NumR, NumPr*, TypePr, Resul, Norme, Conclusion)

Partie 1:

Préparation et création de la structure

1. Création d' un utilisateur DB_NotreNom en lui attribuant avec un mot de passe.

```
SQL> alter session set "_ORACLE_SCRIPT"=true;
Session modifiée.

SQL> create user ferchichi identified by manel;
Utilisateur créé.
```

Commentaire : Pour la création d' un utilisateur en oracle version 19 c Il faut changer la valeur de session a vrai puis soit utiliser le langage C# et utiliser la requête de création normale

2. Donner tous les privilèges à cet utilisateur.

```
SQL> GRANT ALL privileges to ferchichi ;
Autorisation de privilèges (GRANT) acceptée.
```

3. Connectez-vous à cet utilisateur.

```
SQL> connect ferchichi
Entrez le mot de passe :
Connecté.
```

4. 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.

La requête générale utilisée

```
CREATE TABLE TABLENAME (ATTNAME ATTTYPE ,.....);
```

Table Biologistes :

- Create Table Biologistes (NumB number(10), Nom varchar2(30), Prenom varchar2(30), Specialite varchar2(30), RoleB varchar2(40), constraint prbio primary key (NumB));

```
SQL> desc Biologistes ;
```

Nom	NULL ?	Type
NUMB	NOT NULL	NUMBER(10)
NOM		VARCHAR2(30)
PRENOM		VARCHAR2(30)
SPECIALITE		VARCHAR2(30)
ROLEB		VARCHAR2(40)

Table Patient :

- Create table Patient (NumP number(10), Nom varchar2(30), Prenom varchar2(30), Datenaissance Date, constraint PrPat primary key (NumP));

```
SQL> desc Patient
```

Nom	NULL ?	Type
NUMP	NOT NULL	NUMBER(10)
NOM		VARCHAR2(30)
PRENOM		VARCHAR2(30)
DATENAISSANCE		DATE

Table Prélèvement :

- Create table Prelevement (NumPr number(10), NumP number(10), DatePr date, TypePr varchar2(30), constraint prprelev primary key (NumPr), constraint frprelev foreign key (NumP) references Patient (NumP));

```
SQL> desc Prelevement ;
```

Nom	NULL ?	Type
NUMPR	NOT NULL	NUMBER(10)
NUMP		NUMBER(10)
DATEPR		DATE
TYPEPR		VARCHAR2(30)

Table effectuerPre :

- 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 freffectuerthree foreign key (NumPr) references Prelevement (NumPr));

```
SQL> desc Effectuerprelev ;
```

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

Table Resultat :

- Create table Resultat (NumR number(10), NumPr number(10), TypeRr varchar2(30), Resul varchar2(30) ,Norme varchar2(40) , Conclusion varchar2(99) , constraint prresult Primary key (NumR) , constraint frresult foreign key (NumPr) references Prelevement (NumPr));

```
SQL> Create table Resultat (NumR number(10), NumPr number(10), TypeRr varchar2(30), Resul varchar2(30) ,Norme varchar2(40) , Conclusion varchar2(99) , constraint prresult Primary key (NumR) , constraint frresult foreign key (NumPr) references Prelevement (NumPr));
```

Table cr   e.

```
SQL> desc Resultat ;
```

Nom	NULL ?	Type
NUMR	NOT NULL	NUMBER(10)
NUMPR		NUMBER(10)
TYPERR		VARCHAR2(30)
RESUL		VARCHAR2(30)
NORME		VARCHAR2(40)
CONCLUSION		VARCHAR2(99)

Partie 2 :

Remplissage et modification

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

forme générale :

```
INSERT INTO TABLENAME VALUES (VALEURS..);
```

Table Biologistes :

- Insert Into Biologistes Values
(1 , 'BADI' , 'Salim' , 'Microbio' , 'Biologiste-Responsable');
- Insert Into Biologistes Values(2 , 'AMRAN' , 'Zineb' , 'Bio-Med' , 'Biologist-Médical');
- Insert Into Biologistes Values(3 , 'SAHLI' , 'Lamia' , 'Ingénieur' , 'Ing-Qualité');
- Insert Into Biologistes Values(4 , 'NADIR' , 'Ahmed' , 'Biologie' , 'Aide-laboratoire');
- Insert Into Biologistes Values(5 , 'BENMIHOUB' , 'Djamila' , 'Ingénieur' , 'Secrétaire');
- Insert Into Biologistes Values(6 , 'CHERGUI' , 'Selma' , 'Technicien' , 'Technicien');
- Insert Into Biologistes Values(7 , 'BOUSALEM' , 'Ziad' , 'Biologie' , 'Aide-laboratoire');
- Insert Into Biologistes Values(8 , 'KADI' , 'Nadia' , 'Ingénieur' , 'Ing-Informatique');
- Insert Into Biologistes Values(9 , 'SMATI' , 'Radia' , 'Bio-Med' , 'Biologist-Médical');
- Insert Into Biologistes Values(10 , 'NAILI' , 'Mourad' , 'Bio-Med' , 'Biologist-Médical');

```

SQL> Insert Into Biologistes Values
  2 (1 , 'BADI' , 'Salim' , 'Microbio' , 'Biologiste-Responsable');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(2 , 'AMRAN' , 'Zineb' , 'Bio-Med' , 'Biologist-M  dical');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(3 , 'SAHLI' , 'Lamia' , 'Ing  nieur' , 'Ing-Qualit  ');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(4 , 'NADIR' , 'Ahmed' , 'Biologie' , 'Aide-laboratoire');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(5 , 'BENMIHOUB' , 'Djamila' , 'Ing  nieur' , 'Secr  taire');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(6 , 'CHERGUI' , 'Selma' , 'Technicien' , 'Technicien');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(7 , 'BOUSALEM' , 'Ziad' , 'Biologie' , 'Aide-laboratoire');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(8 , 'KADI' , 'Nadia' , 'Ing  nieur' , 'Ing-Informatique');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(9 , 'SMATI' , 'Radia' , 'Bio-Med' , 'Biologist-M  dical');
1 ligne cr    e.

SQL> Insert Into Biologistes Values(10 , 'NAILI' , 'Mourad' , 'Bio-Med' , 'Biologist-M  dical');
1 ligne cr    e.

SQL>

```

Table Patient :

- Insert into Patient values (1, 'SAIDI' , 'Ryad' , TO_DATE('10/02/1970' , 'DD/MM/YYYY'));
- Insert into Patient values(2 , 'BELHADJ' , 'Selma' , TO_DATE('21/03/1976' , 'DD/MM/YYYY'));
- Insert into Patient values(3 , 'DIB' , 'Ahmed' , TO_DATE('03/08/2000' , 'DD/MM/YYYY'));
- Insert into Patient values(4, 'BRAHIMI' , 'Djalil' , TO_DATE('22/06/2002' , 'DD/MM/YYYY'));
- Insert into Patient values(5 , 'SYAD' , 'Hadjer' , TO_DATE('14/09/1999' , 'DD/MM/YYYY'));
- Insert into Patient values(6 , 'NAIM' , 'Fouad' , TO_DATE('23/07/1998' , 'DD/MM/YYYY'));
- Insert into Patient values(7 , 'KADRI' , 'Amine' , TO_DATE('28/05/1970' , 'DD/MM/YYYY'));

- Insert into Patient values(8 , 'SEDDIKI' , 'Wail' ,TO_DATE('20/10/1986', 'DD/MM/YYYY'));
- Insert into Patient values(9 , 'AITALI' , 'Bahia' ,TO_DATE('08/10/1950', 'DD/MM/YYYY'));
- Insert into Patient values(10 , 'SENDJAK' , 'Raouf' ,TO_DATE('02/04/1968', 'DD/MM/YYYY'));

```
SQL> Insert into Patient values (1, 'SAIDI', 'Ryad',TO_DATE('10/02/1970', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(2 , 'BELHADJ' , 'Selma' ,TO_DATE('21/03/1976', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(3 , 'DIB' , 'Ahmed' ,TO_DATE('03/08/2000', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(4, 'BRAHIMI' , 'Djalil' ,TO_DATE('22/06/2002', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(5 , 'SYAD' , 'Hadjer' ,TO_DATE('14/09/1999', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(6 , 'NAIM' , 'Fouad' ,TO_DATE('23/07/1998', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(7 , 'KADRI' , 'Amine',TO_DATE('28/05/1970', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(8 , 'SEDDIKI' , 'Wail' ,TO_DATE('20/10/1986', 'DD/MM/YYYY')
2 );
1 ligne cr  e.

SQL> Insert into Patient values(9 , 'AITALI' , 'Bahia' ,TO_DATE('08/10/1950', 'DD/MM/YYYY')
2 );
1 ligne cr  e.
```

Table Pr  l  vement :

- insert into Prelevement values (1, 1,TO_DATE('04/02/2022', 'DD/MM/YYYY'), 'Sanguin');
- insert into Prelevement values (2, 1 ,TO_DATE('04/02/2022', 'DD/MM/YYYY'), 'Nasopharyng  ');
- insert into Prelevement values (3, 2,TO_DATE('04/02/2022', 'DD/MM/YYYY'), 'Sanguin');
- insert into Prelevement values (4, 3,TO_DATE('04/02/2022', 'DD/MM/YYYY'), 'Cutan  o-Muqueux');
- insert into Prelevement values (5 ,3 ,TO_DATE('04/02/2022', 'DD/MM/YYYY'), 'Sanguin');
- insert into Prelevement values (6 ,4, TO_DATE('04/02/2022', 'DD/MM/YYYY'), 'Nasopharyng  ');

- insert into Prelevement values (7, 5 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Sanguin');
- insert into Prelevement values (8 ,6 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'), 'Sanguin');
- insert into Prelevement values (9, 7,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Nasopharyngé');
- insert into Prelevement values (10, 8 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Cutané-Muqueux');
- insert into Prelevement values (11 ,8 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Sanguin');
- insert into Prelevement values (12 ,9 ,TO_DATE('05/02/2022', 'DD/MM/YYYY'),'Sanguin');
- insert into Prelevement values (13 ,10 ,TO_DATE('06/02/2022', 'DD/MM/YYYY'),'Sanguin');

```
SQL> insert into Prelevement values (1, 1,TO_DATE('04/02/2022', 'DD/MM/YYYY')
2 , 'Sanguin');

1 ligne cr   e.

SQL> insert into Prelevement values (2, 1 ,TO_DATE('04/02/2022', 'DD/MM/YYYY')
2 , 'Nasopharyng  ');

1 ligne cr   e.

SQL> insert into Prelevement values (3, 2,TO_DATE('04/02/2022', 'DD/MM/YYYY')
2 , 'Sanguin');

1 ligne cr   e.

SQL> insert into Prelevement values (4, 3,TO_DATE('04/02/2022', 'DD/MM/YYYY')
2 , 'Cutan  -Muqueux');

1 ligne cr   e.

SQL> insert into Prelevement values (5 ,3 ,TO_DATE('04/02/2022', 'DD/MM/YYYY')
2 , 'Sanguin');

1 ligne cr   e.

SQL> insert into Prelevement values (6 ,4, TO_DATE('04/02/2022', 'DD/MM/YYYY')
2 , 'Nasopharyng  ');

1 ligne cr   e.

SQL> insert into Prelevement values (7, 5 ,TO_DATE('05/02/2022', 'DD/MM/YYYY')
2 , 'Sanguin');

1 ligne cr   e.

SQL> insert into Prelevement values (8 ,6 ,TO_DATE('05/02/2022', 'DD/MM/YYYY')
2 , 'Sanguin');

1 ligne cr   e.

SQL> insert into Prelevement values (9, 7,TO_DATE('05/02/2022', 'DD/MM/YYYY')
2 , 'Nasopharyng  ');

1 ligne cr   e.
```

```

SQL> insert into Prelevement values (10, 8 ,TO_DATE('05/02/2022', 'DD/MM/YYYY')
2  ,'Cutané-Muqueux');

1 ligne cr  e.

SQL> insert into Prelevement values (11 ,8 ,TO_DATE('05/02/2022', 'DD/MM/YYYY')
2  ,'Sanguin');

1 ligne cr  e.

SQL> insert into Prelevement values (12 ,9 ,TO_DATE('05/02/2022' , 'DD/MM/YYYY')
2  ,'Sanguin');

1 ligne cr  e.

SQL> insert into Prelevement values (13 ,10 ,TO_DATE('06/02/2022', 'DD/MM/YYYY')
2  ,'Sanguin');

1 ligne cr  e.

```

Table effectuerPre :

- Insert into Effectuerprelev values (7, 1 ,1);
- Insert into Effectuerprelev values (1 ,1 ,2);
- Insert into Effectuerprelev values (2, 2, 3);
- Insert into Effectuerprelev values (10 ,3 ,4);
- Insert into Effectuerprelev values (4 ,3 ,4);
- Insert into Effectuerprelev values (9 ,3, 5);
- Insert into Effectuerprelev values (2 ,4 ,6);
- Insert into Effectuerprelev values (9 ,5 ,7);
- Insert into Effectuerprelev values (4 ,5 ,7);
- Insert into Effectuerprelev values(7 ,6 ,8);
- Insert into Effectuerprelev values(1, 7, 9);
- Insert into Effectuerprelev values (10 ,8 ,10);
- Insert into Effectuerprelev values (1 ,8, 11);
- Insert into Effectuerprelev values (2, 9, 12);
- Insert into Effectuerprelev values (7 ,10 ,13);


```

SQL> Insert into Effectuerprelev values (7, 1 ,1);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (1 ,1 ,2);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (2, 2, 3);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (10 ,3 ,4);
1 ligne cr   e.
SQL>
SQL> Insert into Effectuerprelev values (4 ,3 ,4);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (9 ,3, 5);
1 ligne cr   e.
SQL>
SQL> Insert into Effectuerprelev values (2 ,4 ,6);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (9 ,5 ,7);
1 ligne cr   e.
SQL>
SQL> Insert into Effectuerprelev values (4 ,5 ,7);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values(7 ,6 ,8);
1 ligne cr   e.
SQL>
SQL> Insert into Effectuerprelev values(1, 7, 9);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (10 ,8 ,10);
1 ligne cr   e.
SQL>
SQL> Insert into Effectuerprelev values (1 ,8, 11);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (2, 9, 12);
1 ligne cr   e.
SQL> Insert into Effectuerprelev values (7 ,10 ,13);
1 ligne cr   e.

```

Table Resultat :

- Insert into Resultat values (1, 1 , 'H  moglobine' , '10.2' , '12    16g/dL' , 'An  mie');
- Insert into Resultat values (2 ,1, 'Plaquettes' , '155k' , '150k    400k/mm3' , 'Sans Particularit  ');
- Insert into Resultat values (3, 1, 'Leucocytes' , '6.2' , '4k    10k/mm3' , 'Sans Particularit  .');
- Insert into Resultat values (4, 1, 'Lymphocytes' , '4.8' , '1.5k    4k/mm3' , 'Poss. Infection');

- Insert into Resultat values (5, 2, 'Antig-Covid', '0.2', '>0.5', 'Négatif');
- Insert into Resultat values (6, 3, 'Groupage A', 'R+', 'A, B, AB, O -+', 'A+');
- Insert into Resultat values (7, 4, 'Culture', 'Staphyl.', '-', 'Infection au Staphylococcus');
- Insert into Resultat values (8, 4, 'Sens. Antibiotique', '+Amoxicilline', '-', 'Sensible à l Amoxicilline');
- Insert into Resultat values (9, 5, 'Hémoglobine', '13.2 12', 'à 16g/dL', 'Sans Particularité.');
- Insert into Resultat values (10, 5, 'Plaquettes', '235k', '150k à 400k/mm3', 'Sans Particularité.');
- Insert into Resultat values (11, 5, 'Leucocytes', '8.1', '4k à 10k/mm3', 'Sans Particularité.');
- Insert into Resultat values (12, 5, 'Lymphocytes', '2.8', '1.5k à 4k/mm3', 'Sans Particularité.');
- Insert into Resultat values (13, 6, 'Antig-Covid', '12.6', '>0.5', 'Positif');
- Insert into Resultat values (14, 7, 'PCR Covid', '8.2', '>0.5', 'Positif');

```
SQL> Insert into Resultat values (1, 1, 'Hémoglobine', '10.2', '12 à 16g/dL', 'Anémie');
1 ligne cr  e.

SQL> Insert into Resultat values (2, 1, 'Plaquettes', '155k', '150k à 400k/mm3', 'Sans Particularité');
1 ligne cr  e.

SQL> Insert into Resultat values (3, 1, 'Leucocytes', '6.2', '4k à 10k/mm3', 'Sans Particularité.');
```

Remarque :

Pour afficher les éléments Insérés dans les tables on fait :

```
select * from Biologistes ;
select * from Patient ;
select * from Prelevement ;
select * from Effectuerprelev ;
select * from Resultat ;
```

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

Cardinalité :

- `select count(*) as cardinalite from Biologistes ;`

```
SQL> select count(*) as cardinalite from Biologistes ;

CARDINALITE
-----
10
```

Degrée :

- `select count(*) as degree from ALL_TAB_COLUMNS where Table_Name = 'BIOLOGISTES' ;`

```
SQL> select count(*) as degree from ALL_TAB_COLUMNS where table_name='BIOLOGISTES' ;

DEGREE
-----
5
```

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

Update table

- `Alter table Resultat Rename column TypeRr to TypeResultat ;`

```
SQL> Alter table Resultat Rename column TypeRr to TypeResultat ;
Table modifiée.

SQL> desc Resultat ;
Nom                                NULL ?   Type
-----
NUMR                                NOT NULL NUMBER(10)
NUMPR                                NUMBER(10)
TYPERESULTAT                        VARCHAR2(30)
RESUL                                VARCHAR2(30)
NORME                                VARCHAR2(40)
CONCLUSION                           VARCHAR2(99)
```

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

- `Alter table Resultat Modify conclusion varchar(100) ;`

```
SQL> Alter table Resultat Modify conclusion varchar(100) ;
```

```
Table modifiée.
```

```
SQL> desc Resultat ;
```

Nom	NULL ?	Type
NUMR	NOT NULL	NUMBER(10)
NUMPR		NUMBER(10)
TYPESRESULTAT		VARCHAR2(30)
RESUL		VARCHAR2(30)
NORME		VARCHAR2(40)
CONCLUSION		VARCHAR2(100)

Partie 3 :

Catalogues et Index

1. Afficher le catalogue des tables et décrire ses attributs.

- `Select * from User_Tables ;`

quelques attributs du catalogue des tables :

Table_Name :

Nom de chaque table trouvée dans le catalogue

Tablespace_Name :

le stockage logique d' une structure d' une table

Cluster_Name:

c est des attribut qui permet à des utilisateurs d' utiliser la meme database autrement dit les tables d' une db a partir de different servers

Status:

la fréquences des changement et updates

Num_Rows:

nombre des lignes d' une table dans le system view

- Il y a même la cardinalité d' une table et son degré ainsi que sa taille maximal

```
SQL> select * from User_tables where table_name ='BIOLOGISTES';
```

TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	PCT_INCREASE	FREELISTS	FREELIST_GROUPS	LOG B	NUM_ROWS	BLOCKS	EMPTY_BLOCKS	AVG_SPACE	CHAIN_CNT	AVG_ROW_LEN

2. Lister les contraintes de la table « Biologiste »

- `SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE FROM ALL_CONSTRAINTS WHERE TABLE_NAME='BIOLOGISTES';`

```
SQL>
SQL> SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE
2 FROM ALL_CONSTRAINTS
3 WHERE TABLE_NAME='BIOLOGISTES';
```

CONSTRAINT_NAME	CONSTRAINT_TYPE
PRBIO	P

3. Lister toutes les contraintes de ce schéma (BD).

- `SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE FROM ALL_CONSTRAINTS;`

Il faut spécifier le nom de la Db

```

SYS_C001727
C

SYS_C001729
C

SYS_C001730
C

CONSTRAINT_NAME
-----
C
-
SYS_C001731
C

SYS_C001787
C

SYS_C001828
C

CONSTRAINT_NAME
-----
C
-
SYS_C001829
C

SYS_C001830
C

SYS_C001831
C

```

résultat de plusieurs ligne

4. Créer un index primaire sur la table « Patient ». Que remarquez-vous ? Justifiez.

- Create Unique Index XP on Patient (NumP) ;

```

SQL> Create Unique Index XP on Patient (NumP);
Create Unique Index XP on Patient (NumP)
                                     *
ERREUR Ó la ligne 1 :
ORA-01408: cette liste de colonnes est d'újÓ indexÚe

```

Raison :

La clé primaire NumP est déjà Indexé par défaut

Rien ne change puisque NumP est déjà une Clé primaire

5. Créer un index secondaire de votre choix

- Create Index YP on Patient (Nom) ;

```
SQL> Create Index YP on Patient (Nom);  
Index cr   .
```

6. Afficher l'ensemble des index existants.

- Select Index_name, Index_Type from All_indexes [where table_name = 'PATIENT'] ;

```
INDEX_NAME  
-----  
INDEX_TYPE  
-----  
PRPAT  
NORMAL  
  
YP  
NORMAL
```

Partie 4 :

Interrogation des donn  es

1. Quels sont les biologistes qui participent dans le plus de pr  l  vements ? le moins de pr  l  vements ?

A- sans les vues

a-le plus

- select NumB, count(*) as total from Effectuerprelev group by NumB having count(NumPr)= (select max (tot) from (select NumB, count(*) as tot from Effectuerprelev group by NumB)tmp) ;

```
NUMB    TOTAL  
-----  
1        3  
2        3  
7        3
```

```
SQL>
```

b-le moins

- `select NumB, count(*) as total from Effectuerprelev group by NumB having count(NumPr)= (select min (tot) from (select NumB, count(*) as tot from Effectuerprelev group by NumB)tmp)`
;

NUMB	TOTAL
4	2
9	2
10	2

B-Avec les vues

Création d' une vue qui contient les numéros de biologistes et le nombre de leur prelevements

- `create view BioPre as (select NumB , count(*) as nbrprelev from Effectuerprelev where NumB is not null group by NumB);`

```
SQL> create view BioPre as (select NumB , Count(*) as nbrprelev From Effectuerprelev where NumB is not Null group by NumB) ;
Vue créée.
```

```
SQL> select * from BioPre ;
```

NUMB	NBRPRELEV
1	3
2	3
4	2
7	3
9	2
10	2

6 lignes sélectionnées.

1/au plus des prélèvements

- `select E.NumB , E.nbrprelev from BioPre E where not exists (select * from BioPre M where M.nbrprelev >E.nbreprelev);`

```
SQL> Select E.NumB , E.nbrprelev from BioPre E where not exists (select * from BioPre M where M.nbrprelev < E.nbrprelev );
```

NUMB	NBRPRELEV
4	2
10	2
9	2

2/ au moins des prélèvements :

- `select E.NumB , E.nbrprelev from BioPre E where not exists (select * from Biopre M where M.nbrprelev <E.nbreprelev);`

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

A- Sans les vues

- `select Pre.TypePr ,count(*) from Prelevement Pre , Resultat Res where Res.conclusion='Positif' and (Res.typeResultat='PCR Covid' or Res.typeResultat='Antig-Covid') and Pre.NumPr =Res.NumPr group by Pre.TypePr ;`

TYPEPR	COUNT(*)
Nasopharyngé	1
Sanguin	1

B-avec les vues

Création d une vue des résultats positive puis faire affichage:

```
SQL> Create view positifprelev as (select NumPr , COUNT(*) as num from Resultat where Conclusion = 'Positif' group by NumPr );
Vue créée.
```

```
SQL> select TypePr , NumPr from Prelevement Where NumPr IN (Select NumPr from positifprelev );
```

TYPEPR	NUMPR
Nasopharyngé	6
Sanguin	7

```
SQL> select count(*) from positifprelev ;
```

COUNT(*)
2

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

A-Sans les vues :

```
select
DatePr,Pat.NumP,trunc(months_between(sysdate,Datenaissance)/12)
from Resultat res , Prelevement Pre , Patient Pat where Pre.NumPr
```

```
=Res.NumPr and Pre.NumP=Pat.NumP and Res.conclusion='Positif' and
(Res.typeResultat='PCR Covid' or Res.typeResultat='Antig-Covid')
and to_char(DatePr , 'mmdd') between '0131' and '0301' ;
```

DATEPR	NUMP	TRUNC(MONTHS_BETWEEN(SYSDATE,DATENAISSANCE)/12)
04/02/22	4	20
05/02/22	5	23

B- Avec les vues :

Création d' une vue pour les prélèvements faites en février :

- create view prelevementdefevrier as (select NumPr ,
to_char(to_date(DatePr, 'DD-MM-YYYY'), 'Month') as mois FROM Prelevement where NumPr In (select NumPr from positifprelev
));

```
SQL> create view prelevementdefevrier as ( select NumPr , to_char(to_date(DatePr, 'DD-MM-YYYY'), 'Month') as mois FROM Prelevement where NumPr In (select NumPr from positifprelev ));
Vue créée.
```

```
SQL> select * from prelevementdefevrier;
```

NUMPR	MOIS
7	Février
6	Février

Création d une autre vue pour les patient ayant des prélèvements en février :

- create view patientdeprfevrier as (select NumPr ,NumP from Prelevement where NumPr IN (select NumPr from prelevemntdefevrier) ;

```
SQL> select * from patientdeprfevrier ;
```

NUMPR	NUMP
6	4
7	5

Puis calculer le nombre des ans en utilisant un fonction qui fait la différence entre la date de system (présent) et la date de naissance des patients en mois , puis diviser sur 12 pour retrouver le nombre des ans

- `select NumP ,trunc(months_between(sysdate,Datenaissance)/12) as age from Patient where NumP IN (select NumP FROM patientdeprfevrier);`

```
SQL> select NumP ,trunc(months_between(sysdate,Datenaissance)/12) as age from Patient where NumP IN (select NumP FROM patientdeprfevrier);
```

NUMP	AGE
4	20
5	23

NUMP	AGE
4	20
5	23

```
SQL>
```

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

- `select distinct TypePr from Prelevement where NumPr in (select NumPr from Effectuerprelev);`

On utilise la jointure entre les deux tables des prélèvement et effectuer les prélèvement :

Remarque:

On aurait pu créer cette sélection autant qu' une vue puis afficher que les types .