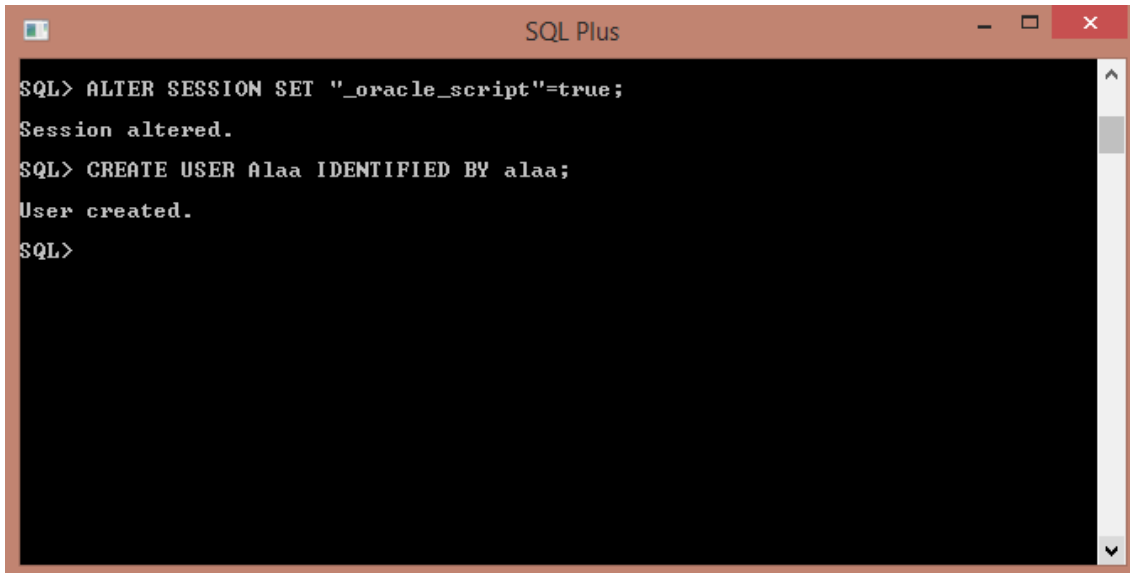


TD/TP2 : LES DROITS D'ACCÈS & DICTIONNAIRES DE DONNÉES

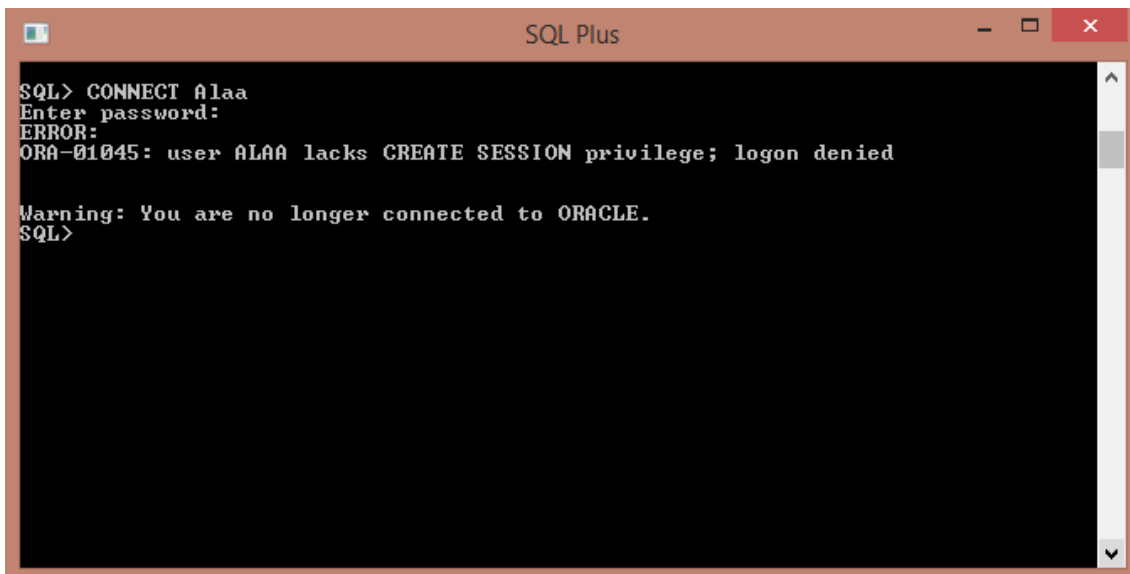
PARTIE 1 : DROITS D'ACCES

1. Créez un autre utilisateur : Admin.



```
SQL> ALTER SESSION SET "_oracle_script"=true;
Session altered.
SQL> CREATE USER Alaa IDENTIFIED BY alaa;
User created.
SQL>
```

2. Connectez-vous à l'aide de cet utilisateur.



```
SQL> CONNECT Alaa
Enter password:
ERROR:
ORA-01045: user ALAA lacks CREATE SESSION privilege; logon denied

Warning: You are no longer connected to ORACLE.
SQL>
```

Que remarquez-vous ?

Lorsque je me connecte en tant qu'utilisateur "Alaa", je remarque que je ne peux pas connecter. En effet, je ne peux pas effectuer certaines actions. Cela est dû au fait que les privilèges de l'utilisateur n'a pas les autorisations nécessaires 'CREATE SESSION' pour accéder à certaines ressources

3. Donner le droit de création d'une session pour cet utilisateur (Create Session) et reconnectez-vous.

```
SQL Plus
SQL> CONNECT system
Enter password:
Connected.
SQL> GRANT CREATE SESSION TO Alaa;

Grant succeeded.

SQL> CONNECT Alaa
Enter password:
Connected.
SQL> _
```

4. Donner les privilèges suivants à Admin: créer des tables, des utilisateurs.

```
SQL Plus
SQL> GRANT CREATE TABLE TO Alaa;

Grant succeeded.

SQL> GRANT CREATE USER TO Alaa;

Grant succeeded.

SQL>
```

Vérifier.

```
SQL Plus
SQL> SELECT * FROM USER_SYS_PRIVS;

USERNAME
-----
PRIVILEGE                                ADM COM INH
-----
SYSTEM
CREATE TABLE                            NO  YES NO
SYSTEM
SELECT ANY TABLE                        NO  YES NO
SYSTEM
DEQUEUE ANY QUEUE                        YES YES NO

USERNAME
-----
PRIVILEGE                                ADM COM INH
-----
SYSTEM
GLOBAL QUERY REWRITE                     NO  YES NO
```

5. Exécutez la requête Q1 suivante : Select * from DBALAB.Biologiste ;

```
SQL Plus
Enter password:
Connected.
SQL> CONNECT Alaa/Alaa
Connected.
SQL> Select * from DBALAB.Biologiste;
Select * from DBALAB.Biologiste
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> conn system/Alaa
Connected.
SQL> GRANT SELECT ON DBALAB.Biologiste TO utilisateur;
GRANT SELECT ON DBALAB.Biologiste TO utilisateur
*
ERROR at line 1:
ORA-01917: user or role 'UTILISATEUR' does not exist

SQL> GRANT SELECT ON DBALAB.Biologiste TO Alaa;
Grant succeeded.
SQL> CONN Alaa/Alaa
```

Que remarquez-vous ?

On remarque que l'utilisateur n'a pas accès à la table (il a pas le droit de voir le contenu de cette table).
L'utilisateur n'a pas les droits « SELECT ».

6. Donner les droits de lecture à cet utilisateur pour la table Biologiste. Exécutez la requête Q1 maintenant.

```
SQL Plus
SQL> GRANT SELECT ON DBALAB.Biologiste TO Alaa;
Grant succeeded.
SQL> CONN Alaa/Alaa
Connected.
SQL> Select * from DBALAB.Biologiste;

```

	NUMB	NOM	PRENOM
SPECIALITE			ROLEB
Microbio	1	BADI	Salim
			Biologiste-Responsable
Bio-Med	2	AMRAN	Zineb
			Biologist-Medical
Ingenieur	3	SAHLI	Lamia
			Ing-Qualite

```

NUMB NOM
-----
PRENOM
-----
```

7. Le centre de gestion du personnel décide de modifier la spécialité “Bio-Med” à “Bio-Medical”.
Que faut-il faire ?

```
SQL Plus
SQL> UPDATE DBALAB.Biologiste
  2 SET SPECIALITE = 'Bio-Medical'
  3 WHERE SPECIALITE = 'Bio-Med';
UPDATE DBALAB.Biologiste
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> _
```

Que remarquez-vous ?

L'utilisateur jour n'a pas les permissions nécessaires pour effectuer l'opération de la mise à jour.

8. Donner les droits de mise à jour à cet utilisateur pour la table Biologiste, les droits de lecture sur la table Patient et réessayer de refaire la modification.

```
SQL> CONNECT system/alaa
Connected.
SQL> GRANT UPDATE ON DBALAB.Biologiste TO Alaa;

Grant succeeded.

SQL> CONNECT alaa/alaa
Connected.
SQL> UPDATE DBALAB.Biologiste
  2 SET SPECIALITE = 'Bio-Medical'
  3 WHERE SPECIALITE = 'Bio-Med';
UPDATE DBALAB.Biologiste
*
ERROR at line 1:
ORA-02290: check constraint (DBALAB.CHECK_BIO) violated
```

Comme nous l'avons vu dans tp1, nous avons activé une contrainte qui vérifie les spécialités, nous devons d'abord la désactiver

```
SQL> CONNECT system/alaa
Connected.
SQL> ALTER TABLE DBALAB.BIOLOGISTE DISABLE CONSTRAINT Check_bio;

Table altered.

SQL> CONNECT alaa/alaa
Connected.
SQL> UPDATE DBALAB.Biologiste
  2 SET SPECIALITE = 'Bio-Medical'
  3 WHERE SPECIALITE = 'Bio-Med';

3 rows updated.
```

9. Créer un index NOMBIO_IX sur l'attribut NOM de la table Biologiste.

```
SQL> CREATE INDEX NOMBIO_IX ON DBALAB.Biologiste (NOM);
CREATE INDEX NOMBIO_IX ON DBALAB.Biologiste (NOM)
*
ERROR at line 1:
ORA-01031: insufficient privileges
```

Que remarquez-vous ?

On remarque que l'utilisateur n'a pas le droit de créer des index sur la table Biologiste.

10. Donner les droits de création d'index à Admin pour la table Biologiste, ensuite réessayer de créer l'index.

```
SQL> GRANT INDEX ON DBALAB.Biologiste TO Alaa;
Grant succeeded.

SQL> CONNECT alaa/alaa
Connected.
SQL> CREATE INDEX NOMBIO_IX ON DBALAB.Biologiste (NOM);
CREATE INDEX NOMBIO_IX ON DBALAB.Biologiste (NOM)
*
ERROR at line 1:
ORA-01950: no privileges on tablespace 'USERS'
```

Que se passe-t-il ?

On remarque que même si l'utilisateur a l'accès pour créer un index, il ne peut toujours pas le faire, en raison du manque d'accès pour allouer un espace dans la tablespace 'USERS'

11. Enlever les privilèges précédemment accordés.

```
SQL> CONNECT system/alaa
Connected.
SQL> REVOKE CREATE SESSION FROM Alaa;
Revoke succeeded.

SQL> REVOKE CREATE TABLE FROM Alaa;
Revoke succeeded.

SQL> REVOKE CREATE USER FROM Alaa;
Revoke succeeded.

SQL> REVOKE SELECT ON DBAlab.Patient FROM Alaa;
Revoke succeeded.

SQL> REVOKE SELECT, UPDATE, INDEX ON DBAlab.Biologiste FROM Alaa;
Revoke succeeded.
```

12. Vérifier que les privilèges ont bien été supprimés.

```
SQL> GRANT CREATE USER TO Alaa;
Grant succeeded.

SQL> SELECT *
2 FROM DBA_SYS_PRIVS
3 WHERE GRANTEE = 'ALAA';

GRAN PRIVILEGE ADM COM INH //before
-----
ALAA CREATE USER NO NO NO

SQL> revoke CREATE USER FROM Alaa;
Revoke succeeded.

SQL> SELECT *
2 FROM DBA_SYS_PRIVS
3 WHERE GRANTEE = 'ALAA';

no rows selected //after
```

13. Créer un profil « Personnel_Profil » qui est caractérisé par : (3 sessions simultanés autorisés, Un appel système ne peut pas consommer plus de 35 secondes de CPU, Chaque session ne peut excéder 90 minutes, Un appel système ne peut lire plus de 1200 blocs de données en mémoire et sur le disque, Chaque session ne peut allouer plus de 25 ko de mémoire en SGA, Pour chaque session, 30 minutes d'inactivité maximum sont autorisées, 5 tentatives de connexion avant blocage du compte, Le mot de passe est valable pendant 50 jours et il faudra attendre 40 jours avant qu'il puisse être utilisé à nouveau, 1 seul jour d'interdiction d'accès après que les 5 tentatives de connexion ont été atteintes, La période de grâce qui prolonge l'utilisation du mot de passe avant son changement est de 5 jours).

```
SQL> alter session set "_ORACLE_SCRIPT"=true;
```

Session altered.

```
SQL>
```

```
SQL> CREATE PROFILE Personnel_Profil LIMIT
 2 SESSIONS_PER_USER          3
 3 CPU_PER_CALL                35000000
 4 CONNECT_TIME                90
 5 LOGICAL_READS_PER_SESSION  1200
 6 PRIVATE_SGA                 25000
 7 IDLE_TIME                   30
 8 FAILED_LOGIN_ATTEMPTS       5
 9 PASSWORD_LIFE_TIME           50
10 PASSWORD_REUSE_TIME         40
11 PASSWORD_GRACE_TIME         5
12 PASSWORD_LOCK_TIME          1;
```

Profile created.

14. Affecter ce profil à l'utilisateur Admin.

```
SQL> ALTER USER Alaa PROFILE Personnel_Profil;
```

User altered.

15. Créer le rôle : « GESTIONNAIRE_DES_PRELEVEMENTS » qui peut voir les tables Biologiste et Patient et peut modifier les tables Prelevement et EffectuePrelevement.

```
SQL> CREATE ROLE GESTIONNAIRE_DES_PRELEVEMENTS;
```

Role created.

```
SQL> GRANT SELECT ON DBalab.Biologiste TO GESTIONNAIRE_DES_PRELEVEMENTS;
```

Grant succeeded.

```
SQL> GRANT SELECT ON DBalab.Patient TO GESTIONNAIRE_DES_PRELEVEMENTS;
```

Grant succeeded.

```
SQL> GRANT UPDATE ON DBalab.Prelevement TO GESTIONNAIRE_DES_PRELEVEMENTS;
```

Grant succeeded.

```
SQL> GRANT UPDATE ON DBalab.EffectuePrelevement TO GESTIONNAIRE_DES_PRELEVEMENTS
;
```

Grant succeeded.

16. Assigner ce rôle à Admin. Vérifier que les autorisations assignées au rôle GESTIONNAIRE_DES_PRELEVEMENTS, ont été bien transférées sur l'utilisateur à Admin.

```
SQL> GRANT GESTIONNAIRE_DES_PRELEVEMENTS TO Alaa;
```

```
Grant succeeded.
```

```
SQL> SELECT TABLE_NAME, PRIVILEGE  
2 FROM ROLE_TAB_PRIVS  
3 WHERE ROLE = 'GESTIONNAIRE_DES_PRELEVEMENTS';
```

```
TABLE_NAME
```

```
PRIVILEGE
```

```
EFFECTUEPRELEVEMENT  
UPDATE
```

```
PATIENT  
SELECT
```

```
PRELEVEMENT  
UPDATE
```

```
TABLE_NAME
```

```
PRIVILEGE
```

```
BIOLOGISTE  
SELECT
```

PARTIE 2 : META-BASE (DICTIONNAIRE DE DONNEES)

Les tables du TP précédent étant créées ;

1. Connecter en tant que « System ». Lister le catalogue « DICT ».

```
Enter user-name: system
Enter password:
Last Successful login time: Fri Apr 14 2023 00:00:52 +01:00

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

SQL> SELECT * FROM DICT;
```

```
TABLE_NAME
-----
COMMENTS
-----
GU$SQLSTATS
Synonym for GU_$SQLSTATS
GU$SQLSTATS_PLAN_HASH
Synonym for GU_$SQLSTATS_PLAN_HASH
```

Il contient combien d'instances ?

```
SQL> SELECT COUNT(*) FROM DICT;

COUNT(*)
-----
      4666
```

Donner sa structure ?

```
SQL> DESC DICT;
Name                                     Null?    Type
-----
TABLE_NAME                             UARCHAR2(128)
COMMENTS                               UARCHAR2(4000)
```

2. Donner le rôle et la structure des tables (ou vues) suivantes : ALL_TAB_COLUMNS, USER_USERS, ALL_CONSTRAINTS et USER_TAB_PRIVS.

Le rôle :

➤ ALL_TAB_COLUMNS

Cette vue fournit des informations sur toutes les colonnes de toutes les tables et vues accessibles à l'utilisateur courant. Les informations incluent le nom de la table, le nom de la colonne, le type de données, la longueur maximale, la précision, l'échelle...

➤ USER_USERS

Cette table fournit des informations sur l'utilisateur courant, notamment le nom d'utilisateur, le mot de passe, le compte bloqué

➤ ALL_CONSTRAINTS

Cette vue fournit des informations sur toutes les contraintes de toutes les tables et vues accessibles à l'utilisateur courant. Les informations incluent le type de contrainte (clé primaire, clé étrangère, vérification, etc.), le nom de la table ou de la vue, le nom de la contrainte...

➤ USER_TAB_PRIVS

Cette table fournit des informations sur les privilèges dont dispose l'utilisateur courant pour les tables et vues de sa propre propriété.

La structure :

➤ ALL_TAB_COLUMNS

```
SQL> DESC ALL_TAB_COLUMNS;
```

Name	Null?	Type
OWNER	NOT NULL	VARCHAR2(128)
TABLE_NAME	NOT NULL	VARCHAR2(128)
COLUMN_NAME	NOT NULL	VARCHAR2(128)
DATA_TYPE		VARCHAR2(128)
DATA_TYPE_MOD		VARCHAR2(3)
DATA_TYPE_OWNER		VARCHAR2(128)
DATA_LENGTH	NOT NULL	NUMBER
DATA_PRECISION		NUMBER
DATA_SCALE		NUMBER
NULLABLE		VARCHAR2(1)
COLUMN_ID		NUMBER
DEFAULT_LENGTH		NUMBER
DATA_DEFAULT		LONG
NUM_DISTINCT		NUMBER
LOW_VALUE		RAW(2000)
HIGH_VALUE		RAW(2000)
DENSITY		NUMBER
NUM_NULLS		NUMBER
NUM_BUCKETS		NUMBER
LAST_ANALYZED		DATE

➤ USER_USERS

```
SQL> DESC USER_USERS;
```

Name	Null?	Type
USERNAME	NOT NULL	VARCHAR2(128)
USER_ID	NOT NULL	NUMBER
ACCOUNT_STATUS	NOT NULL	VARCHAR2(32)
LOCK_DATE		DATE
EXPIRY_DATE		DATE
DEFAULT_TABLESPACE	NOT NULL	VARCHAR2(30)
TEMPORARY_TABLESPACE	NOT NULL	VARCHAR2(30)
LOCAL_TEMP_TABLESPACE		VARCHAR2(30)
CREATED	NOT NULL	DATE
INITIAL_RSRC_CONSUMER_GROUP		VARCHAR2(128)
EXTERNAL_NAME		VARCHAR2(4000)
PROXY_ONLY_CONNECT		VARCHAR2(1)
COMMON		VARCHAR2(3)
ORACLE_MAINTAINED		VARCHAR2(1)
INHERITED		VARCHAR2(3)
DEFAULT_COLLATION		VARCHAR2(100)
IMPLICIT		VARCHAR2(3)
ALL_SHARD		VARCHAR2(3)
PASSWORD_CHANGE_DATE		DATE

➤ ALL_CONSTRAINTS

```
SQL> DESC ALL_CONSTRAINTS;
```

Name	Null?	Type
OWNER		VARCHAR2(128)
CONSTRAINT_NAME	NOT NULL	VARCHAR2(128)
CONSTRAINT_TYPE		VARCHAR2(1)
TABLE_NAME	NOT NULL	VARCHAR2(128)
SEARCH_CONDITION		LONG
SEARCH_CONDITION_VC		VARCHAR2(4000)
R_OWNER		VARCHAR2(128)
R_CONSTRAINT_NAME		VARCHAR2(128)
DELETE_RULE		VARCHAR2(9)
STATUS		VARCHAR2(8)
DEFERRABLE		VARCHAR2(14)
DEFERRED		VARCHAR2(9)
VALIDATED		VARCHAR2(13)
GENERATED		VARCHAR2(14)
BAD		VARCHAR2(3)
RELY		VARCHAR2(4)
LAST_CHANGE		DATE
INDEX_OWNER		VARCHAR2(128)
INDEX_NAME		VARCHAR2(128)
INVALID		VARCHAR2(7)
VIEW_RELATED		VARCHAR2(14)
ORIGIN_CON_ID		NUMBER

➤ USER_TAB_PRIVS

```
SQL> DESC USER_TAB_PRIVS;
```

Name	Null?	Type
GRANTEE		VARCHAR2(128)
OWNER		VARCHAR2(128)
TABLE_NAME		VARCHAR2(128)
GRANTOR		VARCHAR2(128)
PRIVILEGE		VARCHAR2(40)
GRANTABLE		VARCHAR2(3)
HIERARCHY		VARCHAR2(3)
COMMON		VARCHAR2(3)
TYPE		VARCHAR2(24)
INHERITED		VARCHAR2(3)

3. Trouver le nom d'utilisateur avec lequel vous êtes connecté ?

```
SQL> SHOW USER;
USER is "SYSTEM"
```

4. Comparer la structure et le contenu des tables ALL_TAB_COLUMNS et USER_TAB_COLUMNS

ALL_TAB_COLUMNS :

Cette table contient des informations sur les colonnes de toutes les tables visibles par l'utilisateur courant, y compris celles appartenant à d'autres utilisateurs. Elle nécessite les privilèges SELECT sur la vue ALL_TAB_COLUMNS pour y accéder.

```
SQL> DESC ALL_TAB_COLUMNS;
Name                                         Null?    Type
-----
OWNER                                         NOT NULL VARCHAR2(128)
TABLE_NAME                                   NOT NULL VARCHAR2(128)
COLUMN_NAME                                  NOT NULL VARCHAR2(128)
DATA_TYPE                                     VARCHAR2(128)
DATA_TYPE_MOD                                VARCHAR2(3)
DATA_TYPE_OWNER                              VARCHAR2(128)
DATA_LENGTH                                  NOT NULL NUMBER
DATA_PRECISION                               NUMBER
DATA_SCALE                                  NUMBER
NULLABLE                                     VARCHAR2(1)
COLUMN_ID                                    NUMBER
DEFAULT_LENGTH                               NUMBER
DATA_DEFAULT                                 LONG
NUM_DISTINCT                                NUMBER
LOW_VALUE                                    RAW(2000)
HIGH_VALUE                                    RAW(2000)
DENSITY                                      NUMBER
NUM_NULLS                                    NUMBER
NUM_BUCKETS                                  NUMBER
```

USER_TAB_COLUMNS :

Cette table contient des informations sur les colonnes des tables appartenant à l'utilisateur courant uniquement. Elle nécessite les privilèges SELECT sur la table USER_TAB_COLUMNS pour y accéder.

```
SQL> DESC USER_TAB_COLUMNS;
Name                                         Null?    Type
-----
TABLE_NAME                                   NOT NULL VARCHAR2(128)
COLUMN_NAME                                  NOT NULL VARCHAR2(128)
DATA_TYPE                                     VARCHAR2(128)
DATA_TYPE_MOD                                VARCHAR2(3)
DATA_TYPE_OWNER                              VARCHAR2(128)
DATA_LENGTH                                  NOT NULL NUMBER
DATA_PRECISION                               NUMBER
DATA_SCALE                                  NUMBER
NULLABLE                                     VARCHAR2(1)
COLUMN_ID                                    NUMBER
DEFAULT_LENGTH                               NUMBER
DATA_DEFAULT                                 LONG
NUM_DISTINCT                                NUMBER
LOW_VALUE                                    RAW(2000)
HIGH_VALUE                                    RAW(2000)
DENSITY                                      NUMBER
NUM_NULLS                                    NUMBER
NUM_BUCKETS                                  NUMBER
LAST_ANALYZED                                DATE
```

5. Vérifiez que les tables du TP1 ont été réellement créées ?

```
SQL> SELECT owner, table_name FROM all_tables where owner='DBALAB';
```

OWNER

TABLE_NAME

DBALAB
PATIENT

DBALAB
BIOLOGISTE

DBALAB
PRELEVEMENT

OWNER

TABLE_NAME

DBALAB
EFFECTUEPRELEVEMENT

DBALAB
RESULTAT

Donner toutes les informations sur ces tables ?

```
SQL> SELECT table_name, owner, column_name, data_type, data_length, nullable
2 FROM all_tab_columns
3 WHERE owner = 'DBALAB';
```

TABLE_NAME

OWNER

COLUMN_NAME

DATA_TYPE

DATA_LENGTH N

EFFECTUEPRELEVEMENT

DBALAB

NUMP

TABLE_NAME

6. Lister les tables de l'utilisateur « system » et celles de l'utilisateur DBALAB (l'utilisateur de TP1).

➤ SYSTEM

```
SQL> SELECT OWNER, TABLE_NAME FROM ALL_TABLES WHERE OWNER='SYSTEM';
```

OWNER

TABLE_NAME

SYSTEM

OL\$HINTS

SYSTEM

OL\$NODES

SYSTEM

LOGMNR_DICTSTATE\$

OWNER

➤ DBALAB

```
SQL> SELECT OWNER, TABLE_NAME FROM ALL_TABLES WHERE OWNER='DBALAB';
```

OWNER

TABLE_NAME

DBALAB

PATIENT

DBALAB

BIOLOGISTE

DBALAB

PRELEVEMENT

OWNER

TABLE_NAME

7. Donner la description des attributs des tables Biologiste et Patient (Exploiter la table USER_TAB_COLUMNS).

➤ BIOLOGISTE

```
SQL> SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH, NULLABLE
2 FROM USER_TAB_COLUMNS
3 WHERE TABLE_NAME = 'BIOLOGISTE';
```

COLUMN_NAME

DATA_TYPE

DATA_LENGTH N

NUMB
NUMBER 22 N

NOM
VARCHAR2 25 N

COLUMN_NAME

➤ PATIENT

```
SQL> SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH, NULLABLE
2 FROM USER_TAB_COLUMNS
3 WHERE TABLE_NAME = 'PATIENT';
```

COLUMN_NAME

DATA_TYPE

DATA_LENGTH N

NUMP
NUMBER 22 N

NOM
VARCHAR2 25 N

COLUMN_NAME

8. Comment peut-on vérifier qu'il y a une référence de clé étrangère entre les tables Biologiste et Patient?

```
SQL> SELECT COUNT(*), constraint_name, r_constraint_name
2 FROM all_constraints
3 WHERE constraint_type = 'R'
4 AND r_constraint_name = (
5 SELECT constraint_name
6 FROM all_constraints
7 WHERE table_name = 'BIOLOGISTE'
8 AND constraint_type = 'P'
9 )
10 AND table_name = 'EFFECTUEPRELEVEMENT'
11 GROUP BY constraint_name, r_constraint_name;
```

COUNT(*)

CONSTRAINT_NAME

R_CONSTRAINT_NAME

1

FK_NUMB
PK1

9. Donner toutes les contraintes créées lors du TP1 et les informations qui les caractérisent (Exploitez la table USER_CONSTRAINTS);

```
SQL> SELECT table_name, constraint_name, constraint_type
2 FROM user_constraints
3 WHERE owner='DBALAB';
```

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
C		
BIN\$8F9UiYhdSBCU2U9bI8fXqw==\$0		
P		
RESULTAT		
PK4		
P		
TABLE_NAME		
CONSTRAINT_NAME		
C		

10. Retrouver toutes les informations permettant de recréer la table Resultat.

```
SQL> SELECT column_name, column_id, data_type, data_length
2 FROM all_tab_columns
3 WHERE table_name='RESULTAT';
```

COLUMN_NAME	COLUMN_ID	DATA_TYPE	DATA_LENGTH
NUMR	1	NUMBER	22
COLUMN_NAME			

11. Trouver tous les privilèges accordés à Admin.

```
SQL> SELECT * FROM dba_sys_privs WHERE grantee = 'ALAA';
no rows selected
```

```
SQL> GRANT CREATE SESSION TO alaa // granting a permission to check if the command works
2 ;
Grant succeeded.
```

```
SQL> SELECT * FROM dba_sys_privs WHERE grantee = 'ALAA';
```

GRAN	PRIVILEGE	ADM	COM	INH
ALAA	CREATE SESSION	NO	NO	NO

12. Trouver les rôles donnés à l'utilisateur Admin.

```
SQL> SELECT * FROM dba_role_privs WHERE grantee = 'ALAA';
```

GRANTEE	GRANTED_ROLE	ADM	DEL	DEF	COM	INH
ALAA	GESTIONNAIRE_DES_PRELEVEMENTS	NO	NO	YES	YES	NO

13. Trouver tous les objets appartenant à Admin.

```
SQL> connect system/alaa
Connected.
SQL> SELECT object_name, object_type
  2 FROM all_objects
  3 WHERE owner = 'ALAA';

no rows selected

SQL> SELECT * FROM dba_objects WHERE owner = 'ALAA';

no rows selected

SQL> SELECT object_name, object_type // une autre commande
  2 FROM all_objects
  3 WHERE owner = 'ALAA';

no rows selected
```

14. L'administrateur cherche le propriétaire de la table Resultat, comment il pourra le trouver.

```
SQL> SELECT owner, table_name FROM all_tables WHERE table_name='RESULTAT';
```

OWNER	TABLE_NAME
DBALAB	RESULTAT

15. Donner la taille en Ko de la table Resultat.

```
SQL> SELECT ROUND(SUM(BYTES)/1024, 2) AS SIZE_KB
  2 FROM USER_SEGMENTS
  3 WHERE SEGMENT_NAME = 'RESULTAT' AND SEGMENT_TYPE = 'TABLE';
```

SIZE_KB
64

16. Vérifier l'effet produit par chacune des commandes de définition de données du TP1 sur le dictionnaire.

```
SQL> SELECT username, user_id FROM all_users;
```

USERNAME

USER_ID

SYS 0

AUDSYS 8

SYSTEM 9

USERNAME

USER_ID

SYSBACKUP
2147483617

SYSDG
2147483618

SYSKM
2147483619

```
SQL> SELECT table_name FROM user_tables;
```

TABLE_NAME

BIOLOGISTE

PATIENT

PRELEVEMENT

EFFECTUEPRELEVEMENT

RESULTAT

```
SQL> SELECT table_name, count(*) FROM user_tab_columns GROUP BY table_name;
```

TABLE_NAME

COUNT(*)

RESULTAT 6

EFFECTUEPRELEVEMENT 3

PATIENT 4

TABLE_NAME

COUNT(*)

BIOLOGISTE 5

PRELEVEMENT 4

```
SQL> SELECT constraint_name, table_name, constraint_type FROM user_constraints;
```

CONSTRAINT_NAME

TABLE_NAME

C

P

PRE_UNIQUE
PRELEVEMENT

U

```
SQL> SELECT num_rows FROM user_tables WHERE table_name = 'BIOLOGISTE';  
  NUM_ROWS  
-----  
      11  
SQL> SELECT num_rows FROM user_tables WHERE table_name = 'PATIENT';  
  NUM_ROWS  
-----  
      10  
SQL> SELECT num_rows FROM user_tables WHERE table_name = 'EFFECTUEPRELEVEMENT';  
  NUM_ROWS  
-----  
      15  
SQL> SELECT num_rows FROM user_tables WHERE table_name = 'PRELEVEMENT';  
  NUM_ROWS  
-----  
      13  
SQL> SELECT num_rows FROM user_tables WHERE table_name = 'RESULTAT';  
  NUM_ROWS  
-----  
      14
```