

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



TP Bio SGBD

Rapport TP 2 : Les droits d'accès et dictionnaire de données

Fait par :

Nom et prénom : ABDELMALEK BENMEZIANE

Matricule : 171731046778

Spécialité : M1 BIOINFO

Section : A

Contents

1	Partie 1: Droits d'accès	1
1.1	Création de l'utilisateur Admin	1
1.2	Connexion et remarque	2
1.3	La création d'une session pour l'utilisateur et reconnexion . .	3
1.3.1	La création d'une session	3
1.3.2	La connexion	3
1.4	les privilèges suivants à Admin: créer des tables, des utilisateurs.	4
1.5	Exécution de "Select * from DBALAB.Biologiste;" et la remarque	4
1.5.1	L'exécution	4
1.5.2	La remarque	5
1.6	L'attribution du privilège de lecture	5
1.7	La modification de la spécialité "Bio-Med"	6
1.8	Donner le privilège de modification	6
1.8.1	Le privilège de modification	6
1.8.2	La remarque	7
1.9	La création d'un index NOMBIO_IX sur l'attribut NOM de la table Biologiste	7
1.10	Les droits de création d'index à pour la table Biologiste . . .	8
1.11	L'enlèvement des privilèges donnés.	8
1.12	La vérification que tous les privilèges sont supprimés	8
1.13	La création d'un profil (Personnel_Profil)	9
1.14	L'affectation de ce profil à l'utilisateur Admin	9

1.15	La création d'un role	10
1.16	L'assignation de ce rôle à Admin avec la vérification que les autorisations assignées au rôle ont été bien transférées	11
1.16.1	L'assignation	11
1.16.2	La vérification	11
2	Partie 2 : Méta-base (dictionnaire de données)	13
2.1	le catalogue « DICT » et sa structure	13
2.2	Le rôle et la structure des tables	14
2.2.1	ALL_TAB_COLUMNS	14
2.2.2	La structure	14
2.2.3	USER_USERS	15
2.2.4	La structure	15
2.2.5	ALL_CONSTRAINT	16
2.2.6	La structure	16
2.2.7	USER_TAB_PRIVS	17
2.2.8	La structure	17
2.3	Le nom d'utilisateur avec lequel on est connecté	17
2.4	Comparaison des structure et le contenu des tables ALL_TAB_COLUMNS et USER_TAB_COLUMNS	18
2.5	Vérification que les tables du TP1 existent et toutes ces infor- mations	18
2.5.1	La table Biologiste	19
2.5.2	La table Patient	19
2.5.3	La table Prelevement	19
2.5.4	La table EffectuerPrelevement	20
2.5.5	La table Resultat	20
2.6	Lister les tables de l'utilisateur 'system' et 'DBALAB'	21
2.6.1	System	21
2.6.2	DBALAB	22

2.7	La description sur les attributs	23
2.7.1	La table Biologiste	23
2.7.2	La table Patient	24
2.8	Vérification d'une référence de clé étrangère entre Biologiste et Patient	24
2.9	Les contraintes de TP1	25
2.9.1	La table Biologiste	25
2.9.2	La table Patient	25
2.9.3	La table Prelevement	25
2.9.4	La table EffectuerPrelevement	26
2.9.5	La table Resultat	26
2.10	Retrouver toutes les informations permettant de recréer la table Resultat.	26
2.11	Les privilèges données à Admin	27
2.12	Les roles données à Admin	27
2.13	Les objets d'Admin	28
2.14	Le propriétaire de la table Resultat	28
2.15	la taille en Ko de la table Resultat	28
2.16	L'effet de chaque commande de TP1 sur le dictionnaire	29

List of Figures

1.1	Create user request	1
1.2	Output	2
1.3	Connection	2
1.4	Output	2
1.5	Create session	3
1.6	Output	3
1.7	Connection	3
1.8	Output	3
1.9	Grant privileges	4
1.10	Output	4
1.11	Select request	4
1.12	Output	5
1.13	Grant select	5
1.14	Output	5
1.15	Select request	5
1.16	Output	6
1.17	Update request	6
1.18	Output	6
1.19	Grant update	6
1.20	Output	7
1.21	Update request	7
1.22	Output	7
1.23	Create index	7

1.24	Output	7
1.25	Revoke privileges	8
1.26	Output	8
1.27	Test Revoke	8
1.28	Output	8
1.29	Test Revoke	9
1.30	Output	9
1.31	User to profile	9
1.32	Output	10
1.33	Create role	10
1.34	Output	10
1.35	User to role	11
1.36	Output	11
1.37	Vérification -1-	11
1.38	Output	11
1.39	Vérification -2-	11
1.40	Output	11
1.41	Vérification -3-	12
1.42	Output	12
1.43	Vérification -4-	12
1.44	Output	12
2.1	Nombre d'instances	13
2.2	Output	13
2.3	Afficher user	17
2.4	Output	17
2.5	verify existence	19
2.6	Output	19
2.7	verify existence	19
2.8	Output	19

2.9	verify existence	19
2.10	Output	20
2.11	verify existence	20
2.12	Output	20
2.13	verify existence	20
2.14	Output	20
2.15	system tables	21
2.16	Output -1-	21
2.17	Output -2-	22
2.18	dbalab tables	22
2.19	Output	23
2.20	Description biologiste	23
2.21	Output	24
2.22	Description Patient	24
2.23	Output	24
2.24	Contrainte foreign key	24
2.25	Contraintes biologiste	25
2.26	Output	25
2.27	Contraintes patient	25
2.28	Output	25
2.29	Contraintes prelevement	25
2.30	Output	25
2.31	Contraintes effectuer prelevement	26
2.32	Output	26
2.33	Contraintes Resultat	26
2.34	Output	26
2.35	data Resultat	26
2.36	Output	27
2.37	privileges	27
2.38	Output	27

2.39	role	27
2.40	Output	27
2.41	objects	28
2.42	Output	28
2.43	owner	28
2.44	Output	28
2.45	taille	28
2.46	Output	29

Chapter 1

Partie 1: Droits d'accès

1.1 Création de l'utilisateur Admin

```
alter session set "_oracle_script"=true;  
  
CREATE USER admin IDENTIFIED BY passwd;
```

Figure 1.1: Create user request

Session modifié(e) .

User ADMIN créé(e) .

Figure 1.2: Output

1.2 Connexion et remarque

CONNECT admin / passwd;

Figure 1.3: Connection

```
Erreur commençant à la ligne: 2 de la commande -
connect ...
Rapport d'erreur -
Echec de la connexion
  USER          = admin
  URL           = jdbc:oracle:thin:@localhost:1521:xe
Message d'erreur = ORA-01045: l'utilisateur ADMIN n'a pas le privilège CREATE SESSION ; connexion refusée
Commit
```

Figure 1.4: Output

1.3 La création d'une session pour l'utilisateur et reconnexion

1.3.1 La création d'une session

```
GRANT CREATE SESSION TO admin;
```

Figure 1.5: Create session

```
Succès de l'élément Grant.
```

Figure 1.6: Output

1.3.2 La connexion

```
CONNECT admin / passwd;
```

Figure 1.7: Connection

```
Connecté.  
La connexion créée via la commande de script CONNECT a été déconnectée
```

Figure 1.8: Output

- 1.4 les privilèges suivants à Admin: créer des tables, des utilisateurs.

```
GRANT create table to admin;  
GRANT create user to admin;
```

Figure 1.9: Grant privileges

```
Succès de l'élément Grant.
```

```
Succès de l'élément Grant.
```

Figure 1.10: Output

- 1.5 Exécution de "Select * from DBALAB.Biologiste;"
et la remarque

1.5.1 L'exécution

```
SELECT * FROM Biologiste;
```

Figure 1.11: Select request

	NUMB	NOM	PRENOM	SPECIALITE	ROLEB
1	13	AHMED	Amine	Biologie Moleculaire	Technicien
2	1	BADI	Salim	Microbio	Biologiste-Responsable
3	2	AMRAN	Zineb	Bio-Med	Biologist-Médical
4	3	SAHLI	Lamia	Ingénieur	Ing-Qualité
5	4	NADIR	Ahmed	Biologie	Aide-laboratoire
6	5	BENMIHOUB	Djamila	Ingénieur	Secrétaire
7	6	CHERGUI	Selma	Technicien	Technicien
8	7	BOUSALEM	Ziad	Biologie	Aide-laboratoire
9	8	KADI	Nadia	Ingénieur	Ing-Informatique
10	9	SMATI	Radia	Bio-Med	Biologist-Médical
11	10	NAILI	Mourad	Bio-Med	Biologist-Médical

Figure 1.12: Output

1.5.2 La remarque

L'affichage de la table 'Biologiste' est normal.

1.6 L'attribution du privilège de lecture

```
GRANT SELECT ON Biologiste TO admin;
```

Figure 1.13: Grant select

Succès de l'élément Grant.

Figure 1.14: Output

```
SELECT * FROM Biologiste;
```

Figure 1.15: Select request

	NUMB	NOM	PRENOM	SPECIALITE	ROLEB
1	13	AHMED	Amine	Biologie Moleculaire	Technicien
2	1	BADI	Salim	Microbio	Biologiste-Responsable
3	2	AMRAN	Zineb	Bio-Med	Biologist-Médical
4	3	SAHLI	Lamia	Ingénieur	Ing-Qualité
5	4	NADIR	Ahmed	Biologie	Aide-laboratoire
6	5	BENMIHOUB	Djamila	Ingénieur	Secrétaire
7	6	CHERGUI	Selma	Technicien	Technicien
8	7	BOUSALEM	Ziad	Biologie	Aide-laboratoire
9	8	KADI	Nadia	Ingénieur	Ing-Informatique
10	9	SMATI	Radia	Bio-Med	Biologist-Médical
11	10	NAILI	Mourad	Bio-Med	Biologist-Médical

Figure 1.16: Output

1.7 La modification de la spécialité “Bio-Med”

```
UPDATE Biologiste SET specialite = 'Bio-Medical';
```

Figure 1.17: Update request

```
Erreur commençant à la ligne: 2 de la commande -
UPDATE Biologiste SET specialite = 'Bio-Medical'
Rapport d'erreur -
ORA-02290: violation de contraintes (SYSTEM.CHECK_SPECIALITE) de vérification
```

Figure 1.18: Output

1.8 Donner le privilège de modification

1.8.1 Le privilège de modification

```
GRANT UPDATE ON Biologiste TO admin;
```

Figure 1.19: Grant update



```
Succès de l'élément Grant.
```

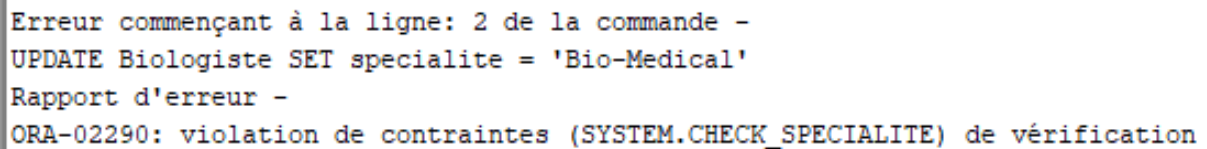
Figure 1.20: Output

1.8.2 La remarque



```
UPDATE Biologiste SET specialite = 'Bio-Medical';
```

Figure 1.21: Update request



```
Erreur commençant à la ligne: 2 de la commande -  
UPDATE Biologiste SET specialite = 'Bio-Medical'  
Rapport d'erreur -  
ORA-02290: violation de contraintes (SYSTEM.CHECK_SPECIALITE) de vérification
```

Figure 1.22: Output

On ne peut pas modifier la valeur de spécialité car c'est une violation de la contrainte d'intégrité "Check_specialite".

1.9 La création d'un index NOMBIO_IX sur l'attribut NOM de la table Biologiste



```
CREATE INDEX NOMBIO_IX ON Biologiste (Nom);
```

Figure 1.23: Create index



```
Index NOMBIO_IX créé(e) .
```

Figure 1.24: Output

1.10 Les droits de création d'index à pour la table Biologiste

L'index est créé précédemment d'une manière normale.

1.11 L'enlèvement des privilèges donnés.

```
REVOKE ALL PRIVILEGES ON Biologiste FROM admin;
```

Figure 1.25: Revoke privileges

```
Succès de l'élément Revoke.
```

Figure 1.26: Output

1.12 La vérification que tous les privilèges sont supprimés

```
REVOKE SELECT ON Biologiste FROM admin;
```

Figure 1.27: Test Revoke

```
Erreur commençant à la ligne: 2 de la commande -  
REVOKE SELECT ON Biologiste FROM admin  
Rapport d'erreur -  
ORA-01927: impossible de révoquer (REVOKE) des privilèges non accordés  
01927. 00000 - "cannot REVOKE privileges you did not grant"  
*Cause:      You can only revoke privileges you granted.  
*Action:     Don't revoke these privileges.
```

Figure 1.28: Output

1.13 La création d'un profil (Personnel_Profil)

```
CREATE PROFILE Personnel_Profil LIMIT
  SESSIONS_PER_USER          3
  CPU_PER_SESSION            35000
  CONNECT_TIME                90
  LOGICAL_READS_PER_SESSION  1200
  PRIVATE_SGA                 25k
  IDLE_TIME                   30
  FAILED_LOGIN_ATTEMPTS       5
  PASSWORD_LIFE_TIME           50
  PASSWORD_REUSE_TIME          40
  PASSWORD_LOCK_TIME           1
  PASSWORD_GRACE_TIME          5;
```

Figure 1.29: Test Revoke


```
Profile PERSONNEL_PROFIL créé(e) .
```

Figure 1.30: Output

1.14 L'affectation de ce profil à l'utilisateur Admin

```
ALTER USER admin PROFILE Personnel_Profil;
```

Figure 1.31: User to profile



```
User ADMIN modifié(e) .
```

Figure 1.32: Output

1.15 La création d'un role

```
CREATE ROLE GESTIONNAIRE_DES_PRELEVEMENTS;  
  
GRANT SELECT ON Biologiste TO GESTIONNAIRE_DES_PRELEVEMENTS;  
GRANT SELECT ON Patient TO GESTIONNAIRE_DES_PRELEVEMENTS;  
GRANT UPDATE, INSERT, DELETE ON Prelevement TO GESTIONNAIRE_DES_PRELEVEMENTS;  
GRANT UPDATE, INSERT, DELETE ON EffectuerPrelevement TO GESTIONNAIRE_DES_PRELEVEMENTS;
```

Figure 1.33: Create role



```
Role GESTIONNAIRE_DES_PRELEVEMENTS créé(e) .
```

```
Succès de l'élément Grant.
```

```
Succès de l'élément Grant.
```

```
Succès de l'élément Grant.
```

```
Succès de l'élément Grant.
```

Figure 1.34: Output

1.16 L'assignation de ce rôle à Admin avec la vérification que les autorisations assignées au rôle ont été bien transférées

1.16.1 L'assignation

```
GRANT GESTIONNAIRE_DES_PRELEVEMENTS TO Admin;
```

Figure 1.35: User to role

Succès de l'élément Grant.

Figure 1.36: Output

1.16.2 La vérification

Table Biologiste

```
-- Vérifier les privilèges SELECT accordés à Admin sur la table Biologiste
SELECT * FROM USER_TAB_PRIVS WHERE PRIVILEGE = 'SELECT' AND TABLE_NAME = 'BIOLOGISTE';
```

Figure 1.37: Vérification -1-

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
1 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	BIOLOGISTE	SYSTEM	SELECT	NO	NO	YES	TABLE	NO

Figure 1.38: Output

Table Patient

```
-- Vérifier les privilèges SELECT accordés à Admin sur la table Patient
SELECT * FROM USER_TAB_PRIVS WHERE PRIVILEGE = 'SELECT' AND TABLE_NAME = 'PATIENT';
```

Figure 1.39: Vérification -2-

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
1 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	PATIENT	SYSTEM	SELECT	NO	NO	YES	TABLE	NO

Figure 1.40: Output

Table Prelevement

```
-- Vérifier les privilèges UPDATE, INSERT et DELETE accordés à Admin sur la table Prelevement
SELECT * FROM USER_TAB_PRIVS WHERE PRIVILEGE IN ('UPDATE', 'INSERT', 'DELETE') AND TABLE_NAME = 'PRELEVEMENT';
```

Figure 1.41: Vérification -3-

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
1 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	PRELEVEMENT	SYSTEM	DELETE	NO	NO	YES	TABLE	NO
2 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	PRELEVEMENT	SYSTEM	INSERT	NO	NO	YES	TABLE	NO
3 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	PRELEVEMENT	SYSTEM	UPDATE	NO	NO	YES	TABLE	NO

Figure 1.42: Output

Table EffectuerPrelevement

```
-- Vérifier les privilèges UPDATE, INSERT et DELETE accordés à Admin sur la table EffectuerPrelevement
SELECT * FROM USER_TAB_PRIVS WHERE PRIVILEGE IN ('UPDATE', 'INSERT', 'DELETE') AND TABLE_NAME = 'EFFECTUERPRELEVEMENT';
```

Figure 1.43: Vérification -4-

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
1 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	EFFECTUERPRELEVEMENT	SYSTEM	DELETE	NO	NO	YES	TABLE	NO
2 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	EFFECTUERPRELEVEMENT	SYSTEM	INSERT	NO	NO	YES	TABLE	NO
3 GESTIONNAIRE_DES_PRELEVEMENTS	SYSTEM	EFFECTUERPRELEVEMENT	SYSTEM	UPDATE	NO	NO	YES	TABLE	NO

Figure 1.44: Output

Chapter 2

Partie 2 : Méta-base (dictionnaire de données)

2.1 le catalogue « DICT » et sa structure

```
SELECT COUNT(*) AS nb_instances  
FROM dba_objects WHERE owner = 'SYS';
```

Figure 2.1: Nombre d'instances

	NB_INSTANCES
1	51348

Figure 2.2: Output

2.2 Le rôle et la structure des tables

2.2.1 ALL_TAB_COLUMNS

Le rôle

Colonnes de mes tables: Nom colonne, type, longueur, obligatoire

2.2.2 La structure

- OWNER
- TABLE_NAME
- COLUMN_NAME
- DATA_TYPE
- DATA_LENGTH
- DATA_SCALE
- NULLABLE
- COLUMN_ID
- DEFAULT_LENGTH
- DATA_DEFAULT
- NUM_DISTINCT
- LOW_VALUE
- HIGH_VALUE
- DENSITY
- NUM_NULLS
- NUM_BUCKETS
- LAST_ANALYZED

- SAMPLE_SIZE
- CHARACTER_SET_NAME
- GLOBAL_STATS
- USER_STATS
- AVG_COL_LEN
- CHAR_LENGTH
- CHAR_USED
- V80_FMT_IMAGE
- DATA_UPGRADED

2.2.3 USER_USERS

Le role

Caractéristiques générales de l'utilisateur: Nom, tablespace, tablespace temporaire.

2.2.4 La structure

- USERNAME
- USER_ID
- PASSWORD
- ACCOUNT_STATUS
- LOCK_DATE
- EXPIRY_DATE
- DEFAULT_TABLESPACE
- TEMPORARY_TABLESPACE
- CREATED

- PROFILE
- INITIAL_RSRC_
- CONSUMER_GROUP
- EXTERNAL_NAME

2.2.5 ALL_CONSTRAINT

Le role

Mes contraintes: Nom,type,table d'accueil,statut

2.2.6 La structure

- OWNER
- CONSTRAINT_NAME
- CONSTRAINT_TYPE
- TABLE_NAME
- SEARCH_CONDITION
- R_OWNER
- R_CONSTRAINT_NAME
- DELETE_RULE
- STATUS
- DEFERRABLE
- DEFERRED
- VALIDATED

2.2.7 USER_TAB_PRIVS

Le role

Des privilèges donnés ou reçu: Bénéficiaire, propriétaire, créateur

2.2.8 La structure

- GRANTEE
- TABLE_NAME
- GRANTOR
- PRIVILEGE
- GRANTABLE

2.3 Le nom d'utilisateur avec lequel on est connecté

```
|  
|  
| show user;  
|
```

Figure 2.3: Afficher user

```
|  
|-----  
| USER est "SYSTEM"
```

Figure 2.4: Output

2.4 Comparaison des structure et le contenu des tables ALL_TAB_COLUMNS et USER_TAB_COLUMNS

La table ALL_TAB_COLUMNS est une vue système qui contient des informations sur les colonnes de toutes les tables accessibles à l'utilisateur courant, tandis que la table USER_TAB_COLUMNS est une vue système qui contient des informations sur les colonnes des tables appartenant à l'utilisateur courant.

Les deux tables ont des structures similaires, avec les mêmes colonnes qui fournissent des informations sur les noms de table, de colonne, de type de données, de contraintes, etc.

Cependant, la différence entre les deux tables réside dans le contenu des données qu'elles contiennent. ALL_TAB_COLUMNS contient des informations sur toutes les tables accessibles à l'utilisateur courant, y compris celles appartenant à d'autres utilisateurs, tandis que USER_TAB_COLUMNS ne contient que des informations sur les tables appartenant à l'utilisateur courant.

En résumé, ALL_TAB_COLUMNS et USER_TAB_COLUMNS ont des structures similaires mais des contenus de données différents, la première contenant des informations sur toutes les tables accessibles à l'utilisateur courant, tandis que la seconde ne contient que des informations sur les tables appartenant à l'utilisateur courant.

2.5 Vérification que les tables du TP1 existent et toutes ces informations

Pour vérifier que les tables existent, on exécute les requetes suivantes:

2.5.1 La table Biologiste

```
SELECT *  
FROM all_tab_columns  
WHERE table_name = 'BIOLOGISTE' AND owner = 'SYSTEM';
```

Figure 2.5: verify existence

OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_TYPE_MOD	DATA_TYPE_OWNER	DATA_LENGTH	DATA_PRECISION	DATA_SCALE	NULLABLE	COLUMN_ID	DEFAULT_LENGTH	DATA_DEF
1 SYSTEM	BIOLOGISTE	NUMB	NUMBER	(null)	(null)	22	4	0	N	1	(null)	(null)
2 SYSTEM	BIOLOGISTE	NOM	VARCHAR2	(null)	(null)	40	(null)	(null)	N	2	(null)	(null)
3 SYSTEM	BIOLOGISTE	PRENOM	VARCHAR2	(null)	(null)	40	(null)	(null)	N	3	(null)	(null)
4 SYSTEM	BIOLOGISTE	SPECIALITE	VARCHAR2	(null)	(null)	40	(null)	(null)	N	4	(null)	(null)
5 SYSTEM	BIOLOGISTE	ROLEB	VARCHAR2	(null)	(null)	40	(null)	(null)	N	5	(null)	(null)

Figure 2.6: Output

2.5.2 La table Patient

```
SELECT *  
FROM all_tab_columns  
WHERE table_name = 'PATIENT' AND owner = 'SYSTEM';
```

Figure 2.7: verify existence

OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_TYPE_MOD	DATA_TYPE_OWNER	DATA_LENGTH	DATA_PRECISION	DATA_SCALE	NULLABLE	COLUMN_ID	DEFAULT_LENGTH	DATA_DEF
1 SYSTEM	PATIENT	NUMP	NUMBER	(null)	(null)	22	4	0	N	1	(null)	(null)
2 SYSTEM	PATIENT	NOM	VARCHAR2	(null)	(null)	40	(null)	(null)	N	2	(null)	(null)
3 SYSTEM	PATIENT	PRENOM	VARCHAR2	(null)	(null)	40	(null)	(null)	N	3	(null)	(null)
4 SYSTEM	PATIENT	DATENAISSANCE	DATE	(null)	(null)	7	(null)	(null)	N	4	(null)	(null)

Figure 2.8: Output

2.5.3 La table Prelevement

```
SELECT *  
FROM all_tab_columns  
WHERE table_name = 'PRELEVEMENT' AND owner = 'SYSTEM';
```

Figure 2.9: verify existence

OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_TYPE_MOD	DATA_TYPE_OWNER	DATA_LENGTH	DATA_PRECISION	DATA_SCALE	NULLABLE	COLUMN_ID	DEFAULT_LENGTH	DATA_DEF
1 SYSTEM	PRELEVEMENT	NUMPR	NUMBER	(null)	(null)	22	4	0 N		1	(null)	(null)
2 SYSTEM	PRELEVEMENT	NUMP	NUMBER	(null)	(null)	22	4	0 N		2	(null)	(null)
3 SYSTEM	PRELEVEMENT	DATEPR	DATE	(null)	(null)	7	(null)	(null) N		3	(null)	(null)
4 SYSTEM	PRELEVEMENT	TYPEPR	VARCHAR2	(null)	(null)	30	(null)	(null) N		4	(null)	(null)

Figure 2.10: Output

2.5.4 La table EffectuerPrelevement

```
SELECT *
FROM all_tab_columns
WHERE table_name = 'EFFECTUERPRELEVEMENT' AND owner = 'SYSTEM';
```

Figure 2.11: verify existence

OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_TYPE_MOD	DATA_TYPE_OWNER	DATA_LENGTH	DATA_PRECISION	DATA_SCALE	NULLABLE	COLUMN_ID	DEFAULT_LENGTH	DATA_DEF
1 SYSTEM	EFFECTUERPRELEVEMENT	NUMB	NUMBER	(null)	(null)	22	4	0 N		1	(null)	
2 SYSTEM	EFFECTUERPRELEVEMENT	NUMP	NUMBER	(null)	(null)	22	4	0 N		2	(null)	
3 SYSTEM	EFFECTUERPRELEVEMENT	NUMPR	NUMBER	(null)	(null)	22	4	0 N		3	(null)	

Figure 2.12: Output

2.5.5 La table Resultat

```
SELECT *
FROM all_tab_columns
WHERE table_name = 'RESULTAT' AND owner = 'SYSTEM';
```

Figure 2.13: verify existence

OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_TYPE_MOD	DATA_TYPE_OWNER	DATA_LENGTH	DATA_PRECISION	DATA_SCALE	NULLABLE	COLUMN_ID	DEFAULT_LENGTH	DATA_DEF
1 SYSTEM	RESULTAT	NUMR	NUMBER	(null)	(null)	22	4	0 N		1	(null)	(null)
2 SYSTEM	RESULTAT	NUMPR	NUMBER	(null)	(null)	22	4	0 N		2	(null)	(null)
3 SYSTEM	RESULTAT	TYPERESULTAT	VARCHAR2	(null)	(null)	40	(null)	(null) N		3	(null)	(null)
4 SYSTEM	RESULTAT	RESUL	VARCHAR2	(null)	(null)	40	(null)	(null) N		4	(null)	(null)
5 SYSTEM	RESULTAT	NORME	VARCHAR2	(null)	(null)	40	(null)	(null) N		5	(null)	(null)
6 SYSTEM	RESULTAT	CONCLUSION	VARCHAR2	(null)	(null)	100	(null)	(null) N		6	(null)	(null)

Figure 2.14: Output

2.6 Lister les tables de l'utilisateur 'system' et 'DBALAB'

2.6.1 System

```
SELECT table_name FROM all_tables WHERE owner='SYSTEM';
```

Figure 2.15: system tables

	TABLE_NAME
1	LOGMNR_SESSION_EVOLVE\$
2	LOGMNR_GLOBAL\$
3	LOGMNR_PDB_INFO\$
4	LOGMNR_DID\$
5	LOGMNR_UID\$
6	LOGMNRGGC_GTLO
7	LOGMNRGGC_GTCS
8	LOGMNRCD_DBNAME_UID_MAP
9	LOGMNR_LOG\$
10	LOGMNR_PROCESSED_LOG\$
11	LOGMNR_SPILL\$

Figure 2.16: Output -1-

jusqu'à :

	TABLE_NAME
126	LOGMNR_OPQTYPE\$
127	LOGMNR_SUBCOLTYPE\$
128	LOGMNR_KOPM\$
129	LOGMNR_PROPS\$
130	LOGMNR_ENC\$
131	LOGMNR_REFCON\$
132	LOGMNR_IDNSEQ\$
133	LOGMNR_PARTOBJ\$
134	LOGMNRP_CTAS_PART_MAP
135	LOGMNR_SHARD_TS
136	LOGSTDBY\$APPLY_PROGRESS

Figure 2.17: Output -2-

2.6.2 DBALAB

```
select table_name from all_tables where owner='DBALAB';
```

Figure 2.18: dbalab tables

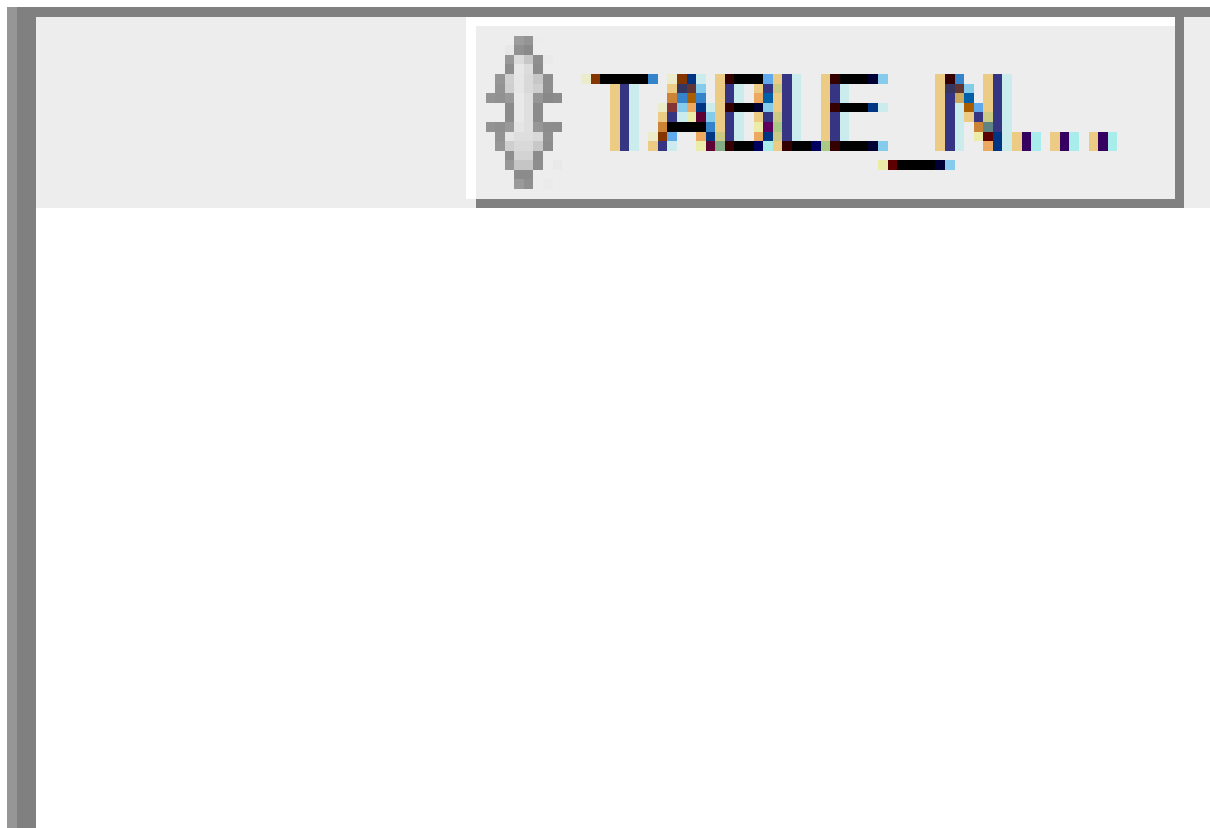


Figure 2.19: Output

2.7 La description sur les attributs

2.7.1 La table Biologiste

```
SELECT column_name, data_type, data_length, nullable  
FROM user_tab_columns  
WHERE table_name = 'BIOLOGISTE';
```

Figure 2.20: Description biologiste

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ DATA_LENGTH	⚡ NULLABLE
1	NUMB	NUMBER	22	N
2	NOM	VARCHAR2	40	N
3	PRENOM	VARCHAR2	40	N
4	SPECIALITE	VARCHAR2	40	N
5	ROLEB	VARCHAR2	40	N

Figure 2.21: Output

2.7.2 La table Patient

```
SELECT column_name, data_type, data_length, nullable
FROM user_tab_columns
WHERE table_name = 'PATIENT';
```

Figure 2.22: Description Patient

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ DATA_LENGTH	⚡ NULLABLE
1	NUMP	NUMBER	22	N
2	NOM	VARCHAR2	40	N
3	PRENOM	VARCHAR2	40	N
4	DATENAissance	DATE	7	N

Figure 2.23: Output

2.8 Vérification d'une référence de clé étrangère entre Biologiste et Patient

```
SELECT * FROM user_constraints
WHERE constraint_type = 'R' AND table_name IN ('BIOLOGISTE', 'PATIENT');
```

Figure 2.24: Contrainte foreign key

2.9 Les contraintes de TP1

2.9.1 La table Biologiste

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

Figure 2.25: 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 2.26: Output

2.9.2 La table Patient

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

Figure 2.27: 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	PK_NUMP	P	PATIENT	(null)	(null)	(null)	(null)	(null)	ENABLED	NO

Figure 2.28: Output

2.9.3 La table Prelevement

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

Figure 2.29: 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	PK_NUMP	P	PATIENT	(null)	(null)	(null)	(null)	(null)	ENABLED	NO

Figure 2.30: Output

2.9.4 La table EffectuerPrelevement

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

Figure 2.31: Contraintes effectuer prelevement

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	SEARCH_CONDITION_VC	R_OWNER	R_CONSTRAINT_NAME	DELETE_RULE
1 SYSTEM	FK_EFFECTUERPRELEVEMENT_BIOLOGISTE	R	EFFECTUERPRELEVEMENT	(null)	(null)	SYSTEM	PK_NUMB	NO ACTION
2 SYSTEM	FK_EFFECTUERPRELEVEMENT_PATIENT	R	EFFECTUERPRELEVEMENT	(null)	(null)	SYSTEM	PK_NUMP	NO ACTION
3 SYSTEM	FK_EFFECTUERPRELEVEMENT_PRELEVEMENT	R	EFFECTUERPRELEVEMENT	(null)	(null)	SYSTEM	PK_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	PK_NUMB_NUMP_NUMPR	P	EFFECTUERPRELEVEMENT	(null)	(null)	(null)	(null)	(null)

Figure 2.32: Output

2.9.5 La table Resultat

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

Figure 2.33: 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	PK_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 2.34: Output

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

```
DESCRIBE RESULTAT;
```

Figure 2.35: data Resultat

Nom	NULL ?	Type
NUMR	NOT NULL	NUMBER(4)
NUMPR	NOT NULL	NUMBER(4)
TYPERESULTAT	NOT NULL	VARCHAR2(40)
RESUL	NOT NULL	VARCHAR2(40)
NORME	NOT NULL	VARCHAR2(40)
CONCLUSION	NOT NULL	VARCHAR2(100)

Figure 2.36: Output

2.11 Les privilèges données à Admin

```
SELECT * FROM USER_TAB_PRIVS WHERE GRANTEE = 'Admin';
```

Figure 2.37: privileges

GRANTEE	OWNER	TABLE_N...	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
---------	-------	------------	---------	-----------	-----------	-----------	--------	------	-----------

Figure 2.38: Output

2.12 Les roles données à Admin

```
SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE = 'ADMIN';
```

Figure 2.39: role

GRANTEE	GRANTED_ROLE	ADMIN_OPTION	DELEGATE_OPTION	DEFAULT_ROLE	COMMON	INHERITED
1 ADMIN	GESTIONNAIRE_DES_PRELEVEMENTS	NO	NO	YES	YES	NO

Figure 2.40: Output

2.13 Les objets d'Admin

```
SELECT * FROM ALL_OBJECTS WHERE OWNER = 'ADMIN';
```

Figure 2.41: objects

OWNER	OBJECT_...	SUBOBJE...	OBJECT_ID	DATA_OB...	OBJECT_...	CREATED	LAST_DDL...	TIMESTAMP	STATUS	TEMPORARY	GENERATED	SECONDARY	NAMESPACE	EDITION_...
-------	------------	------------	-----------	------------	------------	---------	-------------	-----------	--------	-----------	-----------	-----------	-----------	-------------

Figure 2.42: Output

2.14 Le propriétaire de la table Resultat

```
SELECT OWNER FROM ALL_TABLES WHERE TABLE_NAME = 'RESULTAT';
```

Figure 2.43: owner

	OWNER
1	SYSTEM

Figure 2.44: Output

2.15 la taille en Ko de la table Resultat

```
SELECT (SUM(BYTES)/1024) AS "Taille en Ko"  
FROM USER_SEGMENTS  
WHERE SEGMENT_NAME = 'RESULTAT' AND SEGMENT_TYPE = 'TABLE';
```

Figure 2.45: taille

	Taille en Ko
1	64

Figure 2.46: Output

2.16 L'effet de chaque commande de TP1 sur le dictionnaire

Les commandes de définition de données dans le TP1 (CREATE TABLE, ALTER TABLE, etc.) ont toutes un effet sur le dictionnaire de données d'Oracle. Voici comment chacune de ces commandes modifie le dictionnaire :

CREATE TABLE : cette commande crée une nouvelle table dans la base de données et enregistre les informations de définition de la table dans le dictionnaire de données d'Oracle. Ces informations incluent le nom de la table, le nom et le type de chaque colonne, les contraintes de clé primaire et étrangère, les indexes, etc.

ALTER TABLE : cette commande modifie la structure d'une table existante (par exemple, en ajoutant ou supprimant des colonnes, en modifiant les types de données, en ajoutant des contraintes, etc.). Chaque modification est enregistrée dans le dictionnaire de données d'Oracle pour refléter la nouvelle structure de la table.

DROP TABLE : cette commande supprime une table existante de la base de données et supprime également toutes les informations de définition de la table du dictionnaire de données d'Oracle.

CREATE INDEX : cette commande crée un index sur une ou plusieurs colonnes d'une table et enregistre les informations de définition de l'index dans le dictionnaire de données d'Oracle. Ces informations incluent le nom de l'index, le nom de la table, le nom et l'ordre des colonnes indexées, etc.

ALTER INDEX : cette commande modifie la structure d'un index existant (par exemple, en ajoutant ou supprimant des colonnes indexées, en modifiant l'ordre des colonnes, en changeant le nom de l'index, etc.). Chaque modification est enregistrée dans le dictionnaire de données d'Oracle pour refléter la nouvelle structure de l'index.

DROP INDEX : cette commande supprime un index existant de la base de données et supprime également toutes les informations de définition de l'index du dictionnaire de données d'Oracle.

En résumé, chaque commande de définition de données dans le TP1 a un impact sur le dictionnaire de données d'Oracle en créant, modifiant ou supprimant des tables et des indexes, et en enregistrant les informations de définition correspondantes dans le dictionnaire.