



# COMPTE RENDU

Bases de Données Avancées - Lab No. 6  
3e année Cybersécurité - École Supérieure d'Informatique et du  
Numérique (ESIN)  
Collège d'Ingénierie & d'Architecture (CIA)

**Étudiant :** HATHOUTI Mohammed Taha  
**Filière :** Cybersecurité  
**Année :** 2025/2026  
**Enseignants :** Mme.ELHAJI & M.HJJAMI  
**Date :** 13 novembre 2025

# Table des matières

<b>1 Exercice 1 : Créer des Utilisateurs</b>	<b>2</b>
1.1 Créer un utilisateur nommé <i>user_test1</i> avec le mot de passe <i>Test@123</i> . . . . .	2
1.2 Accorder les rôles <i>CONNECT</i> et <i>RESOURCE</i> à l'utilisateur . . . . .	2
1.3 Tester la connexion en tant que <i>user_test1</i> . . . . .	2
<b>2 Exercice 2 : Créer et Assigner des Rôles</b>	<b>3</b>
2.1 Créer un rôle nommé <i>HR_ROLE</i> . . . . .	3
2.2 Accorder les priviléges <i>SELECT</i> et <i>INSERT</i> sur la table <i>EMPLOYEES</i> au rôle <i>HR_ROLE</i> . . . . .	3
2.3 Accorder <i>HR_ROLE</i> à <i>user_test1</i> . . . . .	3
2.4 Vérifier que <i>user_test1</i> peut sélectionner et insérer dans la table . . . . .	3
<b>3 Exercice 3 : Privilèges avec <i>ADMIN OPTION</i> et <i>REVOKE</i></b>	<b>4</b>
3.1 Accorder <i>HR_ROLE</i> à un nouvel utilisateur <i>user_test2</i> avec <i>ADMIN OPTION</i> . . . . .	4
3.2 Depuis <i>user_test2</i> , essayer d'accorder <i>HR_ROLE</i> à un autre utilisateur <i>user_test3</i> . . . . .	4
3.3 Révoquer <i>HR_ROLE</i> de <i>user_test2</i> et observer l'effet sur <i>user_test3</i> . . . . .	5
<b>4 Exercice 4 : Privilèges Système vs Privilèges Objet</b>	<b>6</b>
4.1 Accorder le privilège système <i>CREATE TABLE</i> à <i>user_test1</i> . . . . .	6
4.2 Accorder le privilège <i>UPDATE</i> sur la table <i>DEPARTMENTS</i> à <i>user_test1</i> . . . . .	6
4.3 Tester la création d'une table et la mise à jour de la table <i>DEPARTMENTS</i> . . . . .	6
<b>5 Exercice 5 : Interroger les Privilèges et Rôles</b>	<b>7</b>
5.1 Écrire une requête pour lister tous les privilèges système pour <i>user_test1</i> . . . . .	7
5.2 Écrire une requête pour lister tous les rôles accordés à <i>user_test1</i> . . . . .	7
5.3 Écrire une requête pour lister tous les privilèges objet pour <i>user_test1</i> . . . . .	7
<b>6 Exercice 6 : Profils Utilisateur avec Limites de Ressources</b>	<b>8</b>
6.1 Créer un profil nommé <i>SECURE_PROFILE</i> avec les paramètres suivants : . . . . .	8
6.2 Assigner ce profil à <i>user_test1</i> . . . . .	8
6.3 Interroger les paramètres du profil et vérifier le profil assigné à l'utilisateur . . . . .	8

# 1 Exercice 1 : Créer des Utilisateurs

## 1.1 Crée un utilisateur nommé *user\_test1* avec le mot de passe *Test@123*

Résultat :

```
SQL> CREATE USER user_test1 IDENTIFIED BY "Test@123";  
User USER_TEST1 créé(e).
```

## 1.2 Accorder les rôles *CONNECT* et *RESOURCE* à l'utilisateur

Résultat :

```
SQL> GRANT CONNECT , RESOURCE TO user_test1;  
Succès de l'élément Grant.
```

## 1.3 Tester la connexion en tant que *user\_test1*

Résultat :

```
hathouti@Taha-inspiron-16:~/sqlcl/bin$ ~/sqlcl/bin/sql 'user_test1/"  
Test@123"@localhost:1521/XE'  
  
SQLcl : version 25.3 Production sur jeu. nov. 13 14:53:17 2025  
Copyright (c) 1982, 2025, Oracle. Tous droits réservés.  
  
Connecté :  
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production  
Version 21.3.0.0.0  
  
SQL> SELECT USER FROM DUAL;  
  
USER  
-----  
USER_TEST1
```

## 2 Exercice 2 : Créer et Assigner des Rôles

### 2.1 Crée un rôle nommé *HR\_ROLE*

Résultat :

```
SQL> CREATE ROLE HR_ROLE;  
Role HR_ROLE créé(e).
```

### 2.2 Accorder les privilèges *SELECT* et *INSERT* sur la table *EMPLOYEES* au rôle *HR\_ROLE*

Résultat :

```
SQL> GRANT SELECT, INSERT ON HR.EMPLOYEES TO HR_ROLE;  
Succès de l'élément Grant.
```

### 2.3 Accorder *HR\_ROLE* à *user\_test1*

Résultat :

```
SQL> GRANT HR_ROLE TO user_test1;  
Succès de l'élément Grant.
```

### 2.4 Vérifier que *user\_test1* peut sélectionner et insérer dans la table

Résultat :

```
SQL> SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS;  
  
GRANTED_ROLE  
-----  
CONNECT  
RESOURCE  
HR_ROLE
```

### 3 Exercice 3 : Privilèges avec *ADMIN OPTION* et *REVOKE*

#### 3.1 Accorder *HR\_ROLE* à un nouvel utilisateur *user\_test2* avec *ADMIN OPTION*

Résultat :

```
SQL> CREATE USER user_test2 IDENTIFIED BY "Test@123";  
User USER_TEST2 créé(e).  
  
SQL> GRANT CONNECT TO user_test2;  
Succès de l'élément Grant.  
  
SQL> GRANT HR_ROLE TO user_test2 WITH ADMIN OPTION;  
Succès de l'élément Grant.
```

#### 3.2 Depuis *user\_test2*, essayer d'accorder *HR\_ROLE* à un autre utilisateur *user\_test3*

Résultat :

```
hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S5/Bases_de_donnees_Avancees$  
sqlsys  
  
SQL> CREATE USER user_test3 IDENTIFIED BY "Test@123";  
User USER_TEST3 créé(e).  
  
SQL> GRANT CONNECT TO user_test3;  
Succès de l'élément Grant.  
  
hathouti@Taha-inspiron-16:~/sqlcl/bin$ ~/sqlcl/bin/sql 'user_test2/"  
Test@123"@localhost:1521/XE'  
  
SQLcl : version 25.3 Production sur jeu. nov. 13 15:21:02 2025  
Copyright (c) 1982, 2025, Oracle. Tous droits réservés.  
  
Connecté :  
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production  
Version 21.3.0.0.0  
  
SQL> GRANT HR_ROLE TO user_test3;  
Succès de l'élément Grant.
```

```

hathouti@Taha-inspiron-16:~/sqlcl/bin$ ~/sqlcl/bin/sql 'user_test3/"'
Test@123"localhost:1521/XE'

SQLcl : version 25.3 Production sur jeu. nov. 13 15:22:16 2025
Copyright (c) 1982, 2025, Oracle. Tous droits réservés.

Connecté :
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS;

GRANTED_ROLE
-----
HR_ROLE
CONNECT

```

### 3.3 Révoquer *HR\_ROLE* de *user\_test2* et observer l'effet sur *user\_test3*

Résultat :

```

hathouti@Taha-inspiron-16:~/Bureau/UIR/3A/S5/Bases_de_donnees_Avancees$ 
sqlsys

SQL> REVOKE HR_ROLE FROM user_test2;
Success de l'élément Revoke.

hathouti@Taha-inspiron-16:~/sqlcl/bin$ ~/sqlcl/bin/sql 'user_test3/"'
Test@123"localhost:1521/XE'

SQLcl : version 25.3 Production sur jeu. nov. 13 15:29:37 2025
Copyright (c) 1982, 2025, Oracle. Tous droits réservés.

Connecté :
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS;

GRANTED_ROLE
-----
HR_ROLE
CONNECT

```

On remarque que *user\_test3* a gardé son rôle de *HR\_ROLE*.

## 4 Exercice 4 : Privilèges Système vs Privilèges Objet

### 4.1 Accorder le privilège système *CREATE TABLE* à *user\_test1*

Résultat :

```
SQL> GRANT CREATE TABLE TO user_test1;
Succès de l'élément Grant.
```

### 4.2 Accorder le privilège *UPDATE* sur la table *DEPARTMENTS* à *user\_test1*

Résultat :

```
SQL> GRANT UPDATE ON HR.DEPARTMENTS TO user_test1;
Succès de l'élément Grant.
```

### 4.3 Tester la création d'une table et la mise à jour de la table *DEPARTMENTS*

Résultat :

```
SQL> CREATE TABLE test_exo4 (
  2      id NUMBER,
  3      nom VARCHAR2(50)
 4* );
Table TEST_EX04 créée(e).

SQL> UPDATE HR.DEPARTMENTS
  2  SET DEPARTMENT_NAME = 'IT Dept'
  3* WHERE DEPARTMENT_ID = 60;

Erreur rencontré la ligne: 1 de la commande -
UPDATE HR.DEPARTMENTS
SET DEPARTMENT_NAME = 'IT Dept'
WHERE DEPARTMENT_ID = 60
Erreur la ligne de commande: 1 Colonne: 11
Rapport d'erreur -
Erreur SQL : ORA-01031: priviléges insuffisants

https://docs.oracle.com/error-help/db/ora-01031/01031.00000 -
insufficient privileges"
*Document: YES
*Cause: A database operation was attempted without the required
privilege(s).
*Action: Ask your database administrator or security administrator to
grant
you the required privilege(s).

More Details :
https://docs.oracle.com/error-help/db/ora-01031/
```

La mise à jour de la table *DEPARTMENTS* ne se fait pas car avec un *WHERE* il faut que l'utilisateur puisse profiter du privilège *SELECT*.

## 5 Exercice 5 : Interroger les Privilèges et Rôles

### 5.1 Écrire une requête pour lister tous les privilèges système pour *user\_test1*

Résultat :

```
hathouti@Taha-inspiron-16:~/sqlcl/bin$ sqldbsys

SQLcl : version 25.3 Production sur jeu. nov. 13 20:58:50 2025

Copyright (c) 1982, 2025, Oracle. Tous droits réservés.

Connecté :
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SELECT PRIVILEGE
  2  FROM DBA_SYS_PRIVS
  3* WHERE GRANTEE = 'USER_TEST1';

PRIVILEGE
-----
CREATE TABLE
```

### 5.2 Écrire une requête pour lister tous les rôles accordés à *user\_test1*

Résultat :

```
SQL> SELECT GRANTED_ROLE
  2  FROM DBA_ROLE_PRIVS
  3* WHERE GRANTEE = 'USER_TEST1';

GRANTED_ROLE
-----
CONNECT
RESOURCE
HR_ROLE
```

### 5.3 Écrire une requête pour lister tous les privilèges objet pour *user\_test1*

Résultat :

```
SQL> SELECT TABLE_NAME, PRIVILEGE
  2  FROM DBA_TAB_PRIVS
  3* WHERE GRANTEE = 'USER_TEST1';

TABLE_NAME  PRIVILEGE
-----
DEPARTMENTS UPDATE
```

## 6 Exercice 6 : Profils Utilisateur avec Limites de Ressources

### 6.1 Créer un profil nommé *SECURE\_PROFILE* avec les paramètres suivants :

- *FAILED\_LOGIN\_ATTEMPTS* = 3 ;
- *PASSWORD\_LOCK\_TIME* = 1 jour ;
- *PASSWORD\_LIFE\_TIME* = 30 jours ;
- *SESSIONS\_PER\_USER* = 2 ;
- *CPU\_PER\_SESSION* = 10000 (*centisecondes*) ;
- *CONNECT\_TIME* = 60 (*minutes*) ;

Résultat :

```
SQL> CREATE PROFILE SECURE_PROFILE LIMIT
  2      FAILED_LOGIN_ATTEMPTS 3
  3      PASSWORD_LOCK_TIME 1
  4      PASSWORD_LIFE_TIME 30
  5      SESSIONS_PER_USER 2
  6      CPU_PER_SESSION 10000
  7*      CONNECT_TIME 60;
Profile SECURE_PROFILE créé(e).
```

### 6.2 Assigner ce profil à *user\_test1*

Résultat :

```
SQL> ALTER USER user_test1 PROFILE SECURE_PROFILE;
User USER_TEST1 modifié(e).
```

### 6.3 Interroger les paramètres du profil et vérifier le profil assigné à l'utilisateur

Résultat :

```
SQL> SELECT RESOURCE_NAME ,  LIMIT
  2  FROM DBA_PROFILES
  3  WHERE PROFILE = 'SECURE_PROFILE'
  4* ORDER BY RESOURCE_NAME;

RESOURCE_NAME          LIMIT
-----  -----
COMPOSITE_LIMIT        DEFAULT
CONNECT_TIME           60
CPU_PER_CALL           DEFAULT
CPU_PER_SESSION         10000
FAILED_LOGIN_ATTEMPTS  3
IDLE_TIME               DEFAULT
INACTIVE_ACCOUNT_TIME   DEFAULT
LOGICAL_READS_PER_CALL  DEFAULT
```

```
LOGICAL_READS_PER_SESSION DEFAULT
PASSWORD_GRACE_TIME      DEFAULT
PASSWORD_LIFE_TIME        30
PASSWORD_LOCK_TIME        1
PASSWORD_REUSE_MAX        DEFAULT
PASSWORD_REUSE_TIME       DEFAULT
PASSWORD_ROLLOVER_TIME   DEFAULT
PASSWORD_VERIFY_FUNCTION DEFAULT
PRIVATE_SGA                DEFAULT
SESSIONS_PER_USER         2
```

18 lignes sélectionnées.

```
SQL> SELECT USERNAME , PROFILE
  2  FROM DBA_USERS
  3* WHERE USERNAME = 'USER_TEST1';

USERNAME     PROFILE
-----  -----
USER_TEST1  SECURE_PROFILE
```