Rolurile sunt containere pentru privilegii, astfel încât sa permită o mai uşoara administrare a acestora: când un utilizator primeşte un rol, implicit primeşte toate privilegiile conținute in respectivul rol.

Exista roluri predefinite in Oracle, de exemplu:

Rolul	Privilegiile continute in rol
CONNECT	CREATE VIEW CREATE TABLE ALTER SESSION CREATE CLUSTER CREATE SESSION CREATE SYNONYM CREATE SEQUENCE CREATE DATABASE LINK
RESOURCE	CREATE TYPE CREATE TABLE CREATE CLUSTER CREATE TRIGGER CREATE OPERATOR CREATE SEQUENCE CREATE INDEXTYPE CREATE PROCEDURE
DBA	Include toate privilegiile si cu optiune de administrare a lor (de a fi date mai departe)

Observatie: SYSDBA este un caz special de rol, asemănător DBA
Retinem ! A NU se confunda SYS , care este un utilizator, cu SYSDBA, care este un rol.

Sintaxa :
- Crearea unui rol :

```
CREATE ROLE numerol;
```

- Atribuirea unui rol catre un utilizator :

```
GRANT numerol TO utilizator [WITH ADMIN OPTION];
```

- Includerea unor noi privilegii in rolul creat. Acestea vor fi preluate implicit de utilizatorii rolului(daca nu exista contradictii – vezi ierarhia rolurilor in capitolul urmator):

```
GRANT privilegiu1,privilegiu2,..privilegiun [ON obiect] TO numerol;
```

Aflarea rolurilor utilizatorilor aplicației de e-learning se poate realiza cu comanda:

```
SELECT * FROM DBA_role_privs WHERE grantee like '%ELEARN%';
```

Utilizarea rolurilor prezinta avantajul unui management mai usor al privilegiilor, insa prezinta si anumite dezavantaje:

- In cadrul procedurilor rolurile sunt inhibate, nu au efect . Astfel, privilegiul necesar va trebui acordat individual si direct utilizatorului, nu prin rol;
- Cate roluri poate avea simultan un utilizator? Raspuns: zero, unul sau mai multe. Exemplu:

CREATE ROLE select_tot;
GRANT SELECT ANY TABLE TO select_tot;

CREATE ROLE update_tot;
GRANT UPDATE ANY TABLE TO update_tot;

GRANT select_tot TO ELEARN_APP_ADMIN;
GRANT update_tot TO ELEARN_APP_ADMIN;

SELECT * FROM DBA_role_privs WHERE grantee like '%ELEARN%';

```
SQL> SELECT * FROM DBA_role_privs WHERE grantee like '%ELEARN%';

GRANTEE                        GRANTED_ROLE                   ADM DEF
------------------------------ ------------------------------ --- ---
ELEARN_APP_ADMIN               SELECT_TOT                     NO  YES
OPS$MM-33C58500149B\ELEARN_CAT CONNECT                        NO  YES
ELEARN_APP_ADMIN               UPDATE_TOT                     NO  YES
```

Ierarhia prioritatilor de roluri si privilegii

Exista reguli privind agregarea si prioritizarea privilegiilor unui utilizator.
Privilegiile si rolurile pot fi văzute ca modalitati de a da anumite drepturi dar si de a impune anumite restricții. Acest lucru se realizează prin mecanismul GRANT si REVOKE de privilegii si de roluri.

Recapitulam: **ELEARN_APP_ADMIN ,** ca proprietar al tabelei REZOLVA, da comenzile din tabelul următor:

UTILIZATORUL ELEARN_student1 **NU ARE PRIVILEGII PE TABELA REZOLVA**				
PRIVILEGIU SELECT DAT USERULUI DIRECT	**PRIVILEGIU SELECT DAT ROLULUI**	**PRIVILEGIU SELECT DAT ROLULUI**	**PRIVILEGIU SELECT DAT ROLULUI**	**PRIVILEGIU SELECT DAT USERULUI DIRECT**
GRANT SELECT ON REZOLVA TO ELEARN_student1;	CREATE ROLE rol_stud; GRANT SELECT ON REZOLVA TO rol_stud;	CREATE ROLE rol_stud; GRANT SELECT ON REZOLVA TO rol_stud;	CREATE ROLE rol_stud; GRANT SELECT ON REZOLVA TO rol_stud;	GRANT SELECT ON REZOLVA TO ELEARN_student1;
	ACORD ROLUL UTILIZATORULUI	**ACORD ROLUL UTILIZATORULUI**	**ACORD ROLUL UTILIZATORULUI**	**PRIVILEGIU SELECT DAT ROLULUI**
	GRANT rol_stud TO ELEARN_student1;	GRANT rol_stud TO ELEARN_student1;	GRANT rol_stud TO ELEARN_student1;	CREATE ROLE rol_stud; GRANT SELECT ON REZOLVA TO rol_stud;
		PRIVILEGIU SELECT REVOCAT USERULUI DIRECT	**PRIVILEGIU SELECT REVOCAT ROLULUI**	**ACORD ROLUL UTILIZATORULUI**
		REVOKE SELECT ON REZOLVA FROM ELEARN_student1;	REVOKE SELECT ON REZOLVA FROM rol_stud;	GRANT rol_stud TO ELEARN_student1;
				PRIVILEGIU SELECT REVOCAT ROLULUI
				REVOKE SELECT ON REZOLVA FROM rol_stud;
SUCCES	**SUCCES**	**Eroare**	**ESEC**	**SUCCES !**

**** Observam ca un privilegiu obținut in mod direct utilizatorului ramane valabil chiar daca un rol al lui care anterior il cuprindea il pierde.***

**** De asemenea, proprietarul unui obiect are toate privilegiile asupra lui, cu ADMIN option. Nimeni nu ii poate revoca vreodata vreun privilegiu pe un obiect din schema proprie.***

**** Granularitatea de acordare (GRANT) a privilegiilor trebuie respectata in oglinda la retragere(REVOKE):***

GRANT CREATE ANY TABLE TO ELEARN_asistent3;

→ **REVOKE CREATE ANY TABLE FROM** ELEARN_asistent3; -- corect
→ **REVOKE CREATE TABLE FROM** ELEARN_asistent3; -- incorect

* ***Daca un user primeste un privilegiu doar printr-un rol, nu direct, atunci privilegiul respectiv nu i se poate retrage direct cu revoke.***

Reținem ca REVOKE poate fi dat numai la nivelul întregului tabel, nu la nivel de coloane individuale.

Ex: *GRANT UPDATE(deadline) ON ELEARN_APP_ADMIN.TEMA_CASA TO ELEARN_asistent3;*
REVOKE UPDATE ON ELEARN_APP_ADMIN.TEMA_CASA FROM ELEARN_asistent3;

Exercitii

Construirea matricii entitate –utilizator , rezultata din matricile proces-utilizator si entitate-proces

	Studenții cu frecventa	Studenții la distanta	Profesorii	Asistenții	Secretarii	Alumnii	Admin aplicatie si BD	Public larg
FISA_DISCIPLINA	S	S	S	S	S	S	S	S
CURS	S	S	I,U,S	S	S	S	I,U,S	S
MATERIAL_STUDIU	S	S	I,U,D,S	I,U,D,S				
TEMA_CASA	S	S	I,U,S	I,U,S				
NOTA	S	S	S	S	S			
UTILIZATOR							I,U	
CURSANT	S	S	S	S	S		S,I,U	
CADRU_DIDACTIC	S	S					I,U	
EVALUARE	S	S	I,U,S		I,U,S			
PARTICIPA	S	S	I,U	I,U	S		I,U,S	
PREDA	S	S	I,U				I,U	
REZOLVA	I,U	I,U	U	U				
SUSTINE	S	S	I,U		S			
FEEDBACK	I,U,D	I,U,D	S	S				

Legenda: I= Insert , U= update , D= delete, S= select

1. Utilizați trei modalitati diferite de a da drept celor doi utilizatori Profesori sa obtina informații despre coloanele tabelei TEMA_CASA.

Indiciu: privilegii asupra obiectelor schemei acordate pe tabela direct către utilizatori;
privilegii vizualizare acordate direct către utilizatori, view-ul fiind in schema admin-ului;
rol care include privilegiile asupra obiectelor schemei.

2. Utilizați trei modalitati diferite de a da drept utilizatorilor cadre didactice sa actualizeze prin aplicație deadline-ul temelor de casa (coloanele tabelei TEMA_CASA), fara a putea modifica restul informațiilor din tema.

Indiciu: privilegii asupra obiectelor schemei acordate pe tabela direct către utilizatori;
privilegii vizualizare acordate direct către utilizatori, view-ul fiind in schema admin-ului;
rol care include privilegiile asupra obiectelor schemei.

3. Creați o procedura PROC_NOTARE care sa permită notarea temelor de casa. Procedura va fi in schema admin-ului, dar va putea fi apelata de profesori si asistent. Procedura va primi ca parametrii de intrare codul studentului, codul temei, codul cadrului didactic corector si nota acordata. In background, procedura va face prelucrări care sa verifice ca tema aparține studentului indicat si ca nu este deja notata.

4. Creați un context de privilegii la nivelul utilizatorului de tip student care sa fie repetabil pentru orice student din sistem. Contextul va face diferențiere pentru studenții din anul 3 (terminal licenta) si pentru studenții din anul 5 (terminal disertatie) care nu mai trimit teme de casa (exemplu strict educațional).