

## Gestiunea utilizatorilor unei baze de date și a resurselor computaționale la care aceștia au acces

Cuvinte cheie: <ul style="list-style-type: none"><li>• Managementul identităților</li><li>• Autentificare (locală,externă)</li><li>• Autorizare</li></ul>	<ul style="list-style-type: none"><li>• Schema unui utilizator</li><li>• Profil de resurse</li><li>• Plan de consum de resurse</li><li>• Pachetul DBMS_RESOURCE_MANAGER</li></ul>
---	---

### 1. Introducere

- **Managementul identităților**, în termeni succinți, reprezintă răspunsul la două întrebări: *cine este utilizatorul și ce resurse poate accesa el*.
- Elementele constitutive ale unei arhitecturi de management a identităților într-un sistem informatic, în general, sunt:
  - autoritatea centrală care acordă identități;
  - modalitățile de verificare a identității (**autentificare**);
  - mecanismele de propagare a identității, în cazul unei conectări indirecte/mediate;
  - planurile de acces la resurse informaționale și de calcul (**autorizare**).
- În continuare, materialul este structurat într-o primă parte de proiectare (design) și o a doua parte de implementare a unei configurații de management a identităților într-o bază de date.

### 2. Partea I: Proiectarea configurației de management a identităților în baza de date

- Pentru a proiecta o configurație de management a identităților se recomandă parcurgerea următoarelor etape de design:
  - 1) identificarea utilizatorilor principali ai aplicației (indivizi, grupuri, organizații, etc);
  - 2) identificarea proceselor din cadrul aplicației, cu descompunerea în funcții din aplicație necesare pentru efectuarea lor (workflow-uri);
  - 3) construirea matricii proces – utilizator care să evidențieze necesitățile de acces la resurse ale utilizatorilor;
  - 4) identificarea entităților din modelarea datelor aplicației;
  - 5) construirea matricii entitate – proces;
  - 6) construirea matricii entitate – utilizator rezultată din matricile proces-utilizator și entitate-proces.

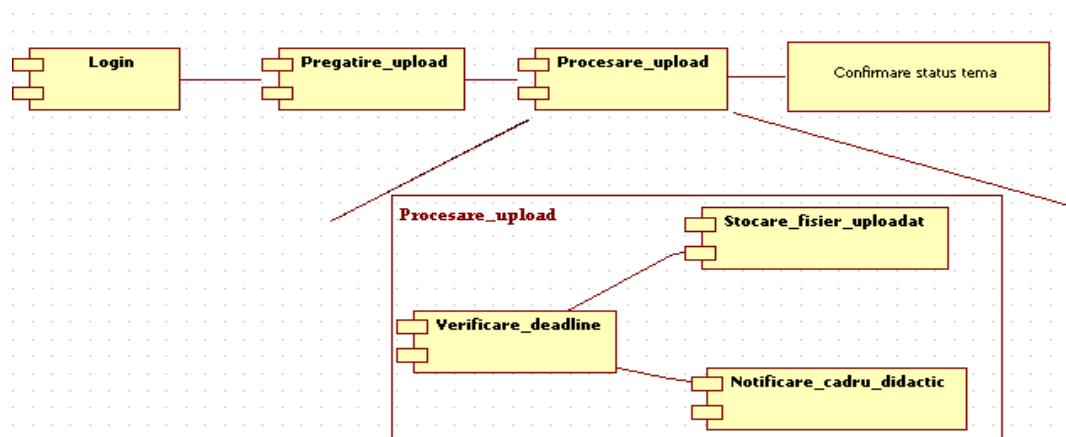
## 2.1 Parcurgerea etapelor de proiectare în contextul unei aplicații de e-learning.

### 2.1.1 Cine sunt utilizatorii aplicației? Ce atribute de identificare au aceștia?

- Utilizatorii aplicației sunt:
  - Studenții (cu frecvență/ la distanță) – număr matricol, nume-prenume-grupă, CNP
  - Cadrele didactice (Profesorii, asistenții) – nume-prenume-departament, departament - poziția în statul de funcții, CNP
  - Secretarii – CNP
  - Administratorul aplicației și al bazei de date – CNP
  - Alunni (absolvenți) – număr matricol expirat, cod diploma emisă, CNP
  - Publicul larg – CNP
- La nivelul tuturor acestor utilizatori, CNP-ul este un atribut de identificare. Un alt atribut de identificare îl reprezintă adresa de e-mail confirmată. Pe măsura evoluțiilor în domeniu, vor putea fi luate în considerare și elementele biometrice drept atribute de identificare.
- Însă, majoritatea acestor informații pot fi aflate și deci nu pot constitui singure baza pentru verificarea identității (adică pentru *autentificare*). De aceea, în mod uzual se construiește un tandem *atribut de identificare – parola*.

### 2.1.2 Care sunt procesele din cadrul aplicației?

- În cadrul aplicației există următoarele procese:
  - P1: Configurare curs
  - P2: Vizualizare fișa cursurilor dintr-un an universitar
  - P3: Înregistrare participant la curs
  - P4: Adăugare materiale de curs
  - P5: Stabilire sesiune evaluare la curs
  - P6: Adăugare enunțuri teme pentru acasă
  - P7: Trimitere rezolvare tema pentru acasă
  - P8: Evaluare și notare tema pentru acasă
  - P9: Consemnare nota pentru sesiune de evaluare
  - P10: Vizualizare nota consemnata într-o sesiune de evaluare per participant la curs
  - P11: Trimitere feedback de către participanții la curs către un cadru didactic curs
  - P12: Vizualizare feedback primit
  - P13: Administrare utilizatori
  - P14: Consultare materiale de curs
  - P15: Vizualizarea cursurilor pe care le frecventează un cursant
- Care este descompunerea funcțională pe 2 niveluri a acestor procese?  
Prezentăm o descompunere funcțională pe două niveluri a procesului de „Trimitere tema pentru acasă”(P7)



- Pentru moment, notăm că poate exista o graniță în acest flux până la care se propagă identitatea utilizatorului, după care în etapele următoare aceasta este înlocuită cu o altă identitate.

### 2.1.3 Construirea matricii proces – utilizator

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
Studentii cu frecvență		X					X			X	X			X	X
Studentii cu frecvență redusă		X					X			X	X			X	X
Profesorii	X	X	X	X	X	X		X	X			X		X	
Asistenții		X	X	X		X		X				X		X	
Secretarii		X			X					X					X
Alumnii (absolvenții)		X													
Administratorul aplicației	X	X	X										X		X
Publicul larg		X													

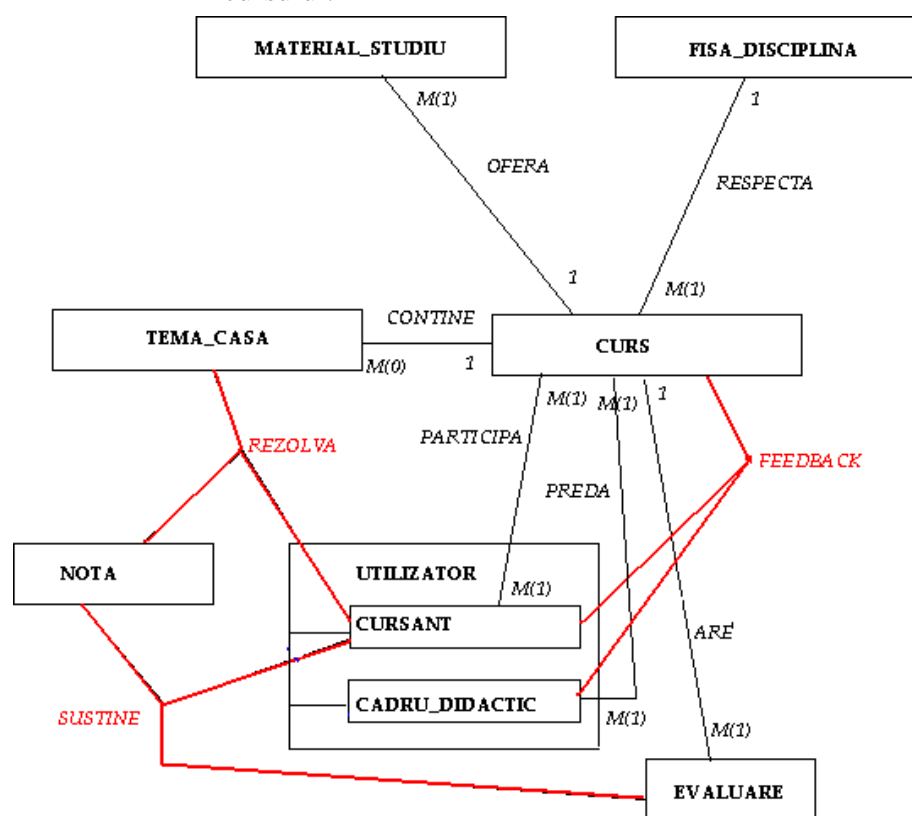
### 2.1.4 Identificarea entităților prin modelarea datelor aplicației

- Sistemul de e-learning este o platformă pentru învățământ modern.
- În sistem pot fi configurate cursurile dintr-un an calendaristic universitar, care respectă fișa disciplinei din planul de învățământ publicat. Aceeași fișă a disciplinei poate sta la baza mai multor cursuri predate în ani succesivi.
- La un curs (dintr-un an calendaristic) predau unul sau mai multe cadre didactice, în calitate de profesori sau de asistenți. Cursul este destinat cursanților (studenți, masteranzi) din promoția curentă și celor restanțieri, care vor fi înregistrați la cursul aflat în desfășurare până când reușesc să promoveze disciplina respectivă.
- Cadrele didactice ale unui curs pot oferi materiale de studiu studenților, într-o manieră centralizată. Totodată, sistemul permite publicarea și colectarea temelor de casă într-o manieră unitară și riguroasă cu privire la termene. Pentru fiecare temă de casă rezolvată

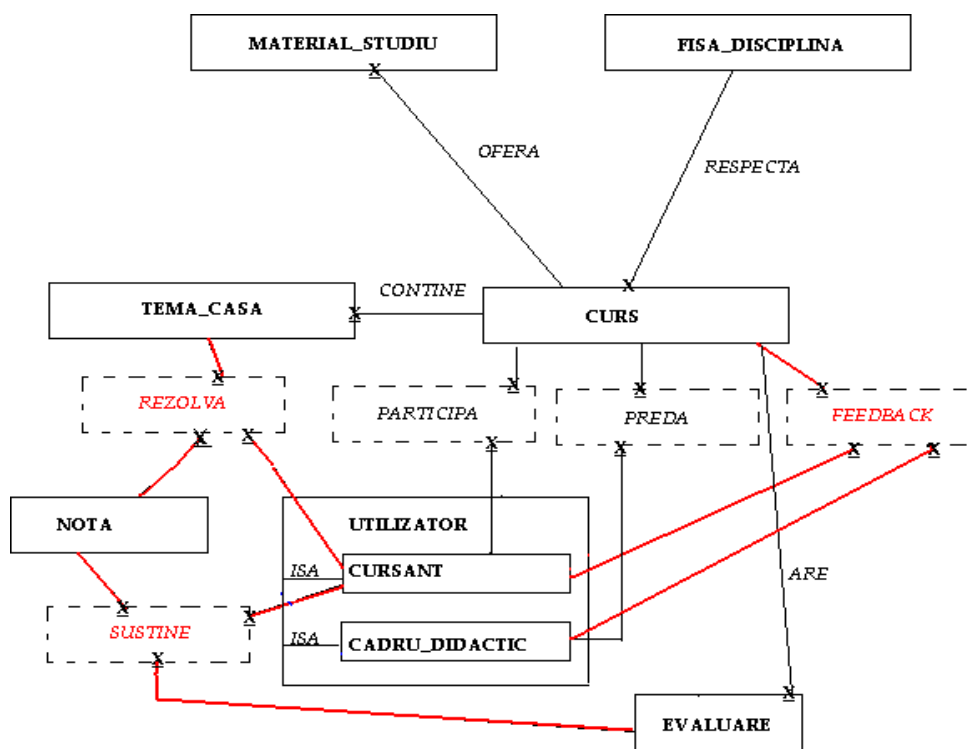
cursantul primește o notă. Catalogul virtual permite consemnarea notelor pentru teme și a celor rezultate din evaluarea cursanților, precum și integrarea cu alte sisteme informatice administrative ale facultății.

- Evaluarea cunoștințelor pentru un curs (dintr-un an universitar) poate avea loc în sesiunile de examene pre-stabilite ale anului respectiv. Nepromovarea de către un cursant a disciplinei într-un an atrage după sine necesitatea refacerii integrale a cursului, inclusiv a temelor la cursul omonim din anul universitar succesiv.

Pentru comunicare, sistemul permite transmiterea de *feedback* de la participanții la curs către cadrele didactice ale cursului.



-----Diagrama E/R-----



-----Diagrama conceptuală-----

**Schemele relaționale:**

FISA\_DISCIPLINA (cod\_fisa\_disciplina#, denumire\_disciplina, an\_studiu, nr\_ore\_curs,  
nr\_ore\_seminar, descriere\_continut, nr\_credite)

CURS (cod\_curs#, cod\_fisadisciplina, an\_universitar)

MATERIAL\_STUDIU (cod\_material#,cod\_curs#, denumire\_material, link\_material,  
grad\_obligativitate)

TEMA\_CASA (cod\_tema#,cod\_curs#, titlu\_tema, enunt, punctaj\_maxim, deadline)

NOTA (cod\_nota#,valoare)

UTILIZATOR (cod\_utilizator#, nume, prenume, tip)

CURSANT (cod\_utilizator#, an\_inmatriculare, an\_curent,grupa\_curenta)

CADRU\_DIDACTIC (cod\_utilizator#, departament, grad stiintific)

EVALUARE (cod\_evaluare#, cod\_curs#, data\_evaluare, sala\_evaluare)

PARTICIPA (cod\_utilizator#, cod\_curs#, flag\_restantier)

PREDĂ (cod\_utilizator#, cod\_curs#, flag\_curs, flag\_laborator)

REZOLVA (cod\_rezolva#, cod\_utilizator, cod\_tema, cod\_nota, link\_fisier\_tema,  
flag\_copiat, cod\_rezolvare\_copiata)

SUSTINE (cod\_sustine#, cod\_utilizator, cod\_evaluare, cod\_nota, flag\_copiat,  
cod\_evaluare\_copiata)

FEEDBACK (cod\_feedback#, cod\_curs, cod\_utilizator, cod\_utilizator, mesaj,  
grad\_satisfactie, data\_feedback)

### 2.1.5 Construirea matricii entitate – proces

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
FISA_DISCIPLINA	S	S													
CURS	I,U	S	S												S
MATERIAL_STUDIU				I,U,D										S	
TEMA_CASA						I,U	S	S							
NOTA								S	S	S					
UTILIZATOR													I,U		
CURSANT			S										I,U		S
CADRU_DIDACTIC											S		I,U		
EVALUARE					I,U				S	S					
PARTICIPA			I,U												S
PREDĂ	I,U										S				
REZOLVA							I,U	U							
SUSTINE									I,U	S					
FEEDBACK											I,U,D	S			

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

### 2.1.6 Construirea matricii entitate –utilizator , rezultată din matricile proces-utilizator și entitate-proces

	Studentii cu frecventa	Studentii cu frecv. redusă	Profesorii	Asistenții	Secretarii	Alumnii	Admin aplicație și 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	
PREDĂ	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

## 2.2 Analiza matricei entitate-utilizator

- În continuarea acestei părți de design a configurației de management a identității în baza de date, vor fi analizate și interpretate informațiile din matricea entitate-utilizator.
- Vom căuta răspuns următoarelor întrebări:

### 2.2.1 Cine sunt utilizatorii bazei de date, ce conturi de utilizatori vom crea?

- În etapa 1) am descoperit clase de utilizatori. Însă, din motive de audit, de non- repudiere, conturile vor fi definite individual pentru utilizatorii identificați în mediul intern organizației. Un alt motiv este că dacă s-ar crea un singur cont de student, de exemplu, există restricții privind numărul conexiunilor paralele admise (*sessions\_per\_user*) și, astfel, unor studenți le va fi refuzată conexiunea, fiind rugați să aștepte până când se eliberează sloturi de conexiune de către alți studenți, ceea ce ar fi un punct slab pentru calitatea aplicației.
- Astfel, spre exemplu, vom defini 1000 de conturi de studenți, câte unul pentru fiecare student al facultății, 50 de conturi de cadre universitare, câte unul pentru fiecare cadru didactic, 5 conturi de secretari.
- Două conturi speciale vor mai fi create: un cont de vizitator (*ELEARN\_GUEST*) pentru publicul larg și un cont de administrator de aplicație (*ELEARN\_APP\_ADMIN*).

### 2.2.2 Cine este proprietarul fiecărui obiect al bazei (tabel, index etc)?

- **Proprietarul unui obiect al bazei de date** este acel utilizator care are drepturi nelimitate de utilizare și administrare asupra obiectului respectiv, drepturi care nu îi pot fi revocate de către nimeni (nici chiar de administratorul bazei de date!).
- Un utilizator care este proprietar a cel puțin un obiect din baza de date nu poate fi șters din sistem.
- **Schema unui utilizator** reprezintă totalitatea obiectelor (tabele, *view*-uri, indecși, *trigger*-i la nivelul bazei de date, secvențe, sinonime, funcții PL/SQL, proceduri PL/SQL, pachete PL/SQL) care au ca proprietar un anumit utilizator al bazei de date.

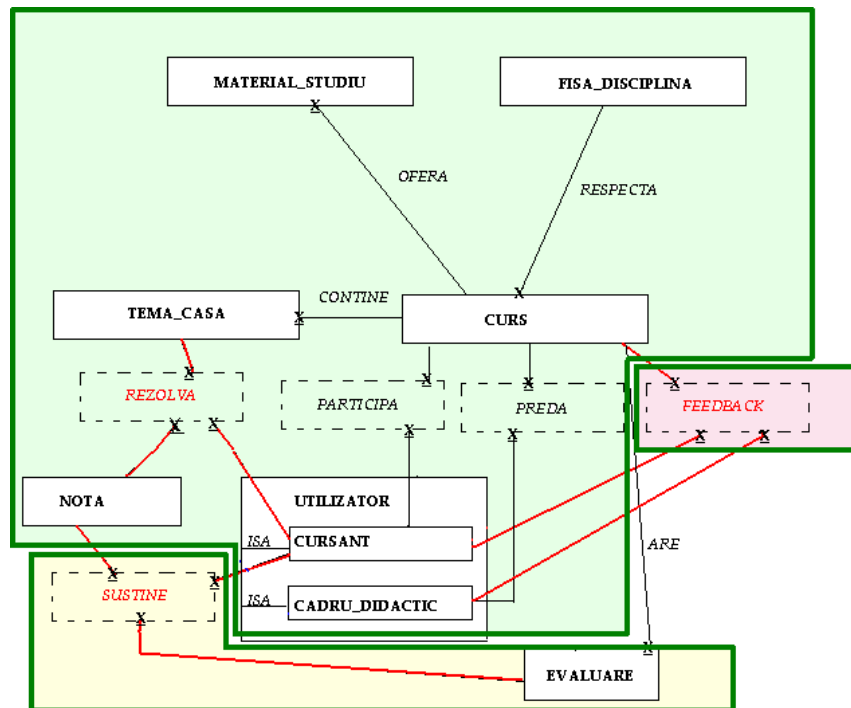
**Observație:** Reținem că între un utilizator și o schemă de utilizator este întotdeauna o relație 1:1.

- O bună practică atunci când dezvoltăm o aplicație cu o bază de date este să creăm un cont unui utilizator administrator de aplicație. Acest utilizator va fi proprietar peste toate obiectele aplicației respective; altfel spus toate obiectele aplicației respective se vor regăsi în schema utilizatorului administrator de aplicație.

Excepțiile sunt permise în măsura în care este justificat *overhead*-ul administrării diferențiate.

- Pentru aplicația noastră de e-learning, presupunem că există următoarele cerințe în specificațiile de sistem:
  - tabelele *EVALUARE* și *SUSTINE* să fie deținute de un proprietar separat *ELEARN\_CAT*;

- tabelul *FEEDBACK* să fie deținut de fiecare cadru didactic individual din sistem; cu alte cuvinte dacă avem 50 de utilizatori cadre didactice, în schema fiecăruia vom avea câte un tabel *FEEDBACK*;
- restul tabelelor vor avea ca proprietar administratorul de aplicație *ELEARN\_APP\_ADMIN*.

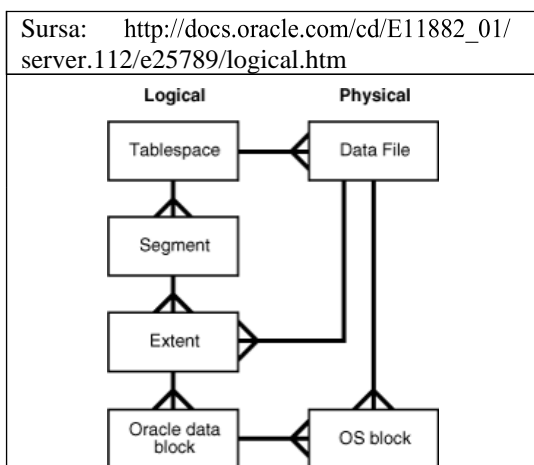


Evidențierea schemelor de utilizator pe diagrama conceptuală

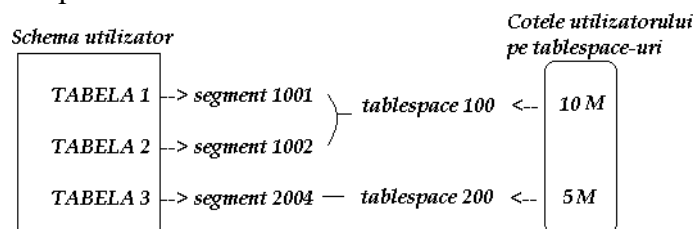
### 2.2.3 Stabilirea aspectelor referitoare la spații de stocare pentru utilizatorii creați

- Așa cum s-a stabilit la punctul anterior, anumiți utilizatori vor fi proprietari de tabele, alții nu. Prin urmare este firesc să considerăm o împărțire diferențiată a spațiului de stocare între utilizatori.
- Un utilizator al bazei de date dispune de două spații de stocare:
  - spațiu de stocare pentru obiectele din schema sa, peste care el este proprietar;
  - spațiu de stocare temporar, folosit de Oracle în diferite situații (de exemplu, în sortarea înregistrărilor când procesează o cerere cu clauza *ORDER BY* sau în construcția tabelii de hash în etapa build din procesarea operației join prin metoda *HASH JOIN*).





- Fără a intra în detalii referitoare la *tablespace*-uri în Oracle, reținem următoarele:
  - *tablespace*-ul este o unitate *logică* de stocare, formată din segmente; un segment stochează datele unui singur obiect al unui utilizator (1 tabel nepartiționat și fără CLOB = 1 segment, 1 index nepartiționat = 1 segment, s.a.m.d.)
  - *tablespace*-ul face legătura între baza de date și sistemul de fișiere, întrucât un *tablespace* are asociat cel puțin un *datafile* (un fișier de date) fizic.
- Un utilizator poate avea cote diferite din diferite *tablespace*-uri pentru stocarea obiectelor din schema sa. Un exemplu ar fi următorul:



- Când unui utilizator i se epuizează spațiul de stocare asignat, administratorul bazei de date poate opta pentru mărirea unei cote existente a utilizatorului sau poate decide asignarea unei cote dintr-un *tablespace* nou.
- Din punctul de vedere al aplicației de e-learning, se va folosi un singur *tablespace*. Dimensiunea sa va fi împărțită în cote astfel:
  - utilizatorul *ELEARN\_APP\_ADMIN*: nelimitat
  - utilizatorul *ELEARN\_CAT*: 10MB
  - utilizatorii profesori *ELEARN\_profesor1*, *ELEARN\_profesor2*, *ELEARN\_asistent3* câte 2MB fiecare
  - restul utilizatorilor 0MB → ei nu vor putea crea obiecte.

## 2.2.4 Stabilirea accesului la resurse computaționale și alte configurări

- Următorul pas, după crearea conturilor de utilizatori și stabilirea spațiilor de stocare, îl reprezintă impunerea unor limitări referitoare la accesul la resurse pentru utilizatori.

- Scopul este de a asigura o funcționare performantă a bazei de date, evitarea monopolului de către un utilizator asupra resurselor ș.a.
- Parametrii de performanță care sunt adesea regăsiți în aceste configurări se referă la:
  - timpul de execuție estimat al instrucțiunilor;
  - consumul de CPU;
  - gradul de paralelism acceptat în sisteme multi-procesor;
  - numărul de sesiuni deschise per utilizator;
  - timp de inactivitate (*idle*).
- O bună practică este de a reutiliza setările privind accesul la resurse prin crearea de profile sau a unor planuri de consum.
- **Un profil de resurse** reprezintă gruparea sub un titlu a unei mulțimi de limitări de resurse care restricționează utilizarea bazei de date și a resurselor instanței Oracle pentru un utilizator. La un moment dat, un utilizator poate avea un singur profil de resurse. Același profil de resurse poate fi setat mai multor utilizatori.
- Referitor la aplicația de e-learning, se solicită :
  - a) pentru vizitatorii în aplicație (*ELEARN\_GUEST*) numărul maxim de conexiuni permise să fie 5, timpul maxim de inactivitate permis (*idle*) să fie de 3 minute, iar timpul total de conectare să nu depășească 15 minute chiar în perioadele de activitate;
  - b) pentru utilizatorii profesori și utilizatorii studenți, consumul de CPU per apel să nu depășească pragul de 60 secunde, să aibă dreptul la o singură sesiune la un moment dat, timpul de viață al parolei să fie de 7 de zile și să poată greși parola de maximum 10 ori.
- **Un plan de consum de resurse** reprezintă un instrument prin care SGBD preia controlul asupra alocării de resurse computaționale către grupuri de sesiuni-utilizator, denumite grupuri de consum. Un plan de resurse complex poate fi reprezentat ca o descompunere *top-down* a consumului unei resurse sub forma unui arbore în care rădăcina este planul principal, în frunze sunt grupurile de consum, iar pe nivelurile intermediare sunt sub-planurile de consum.

### 2.2.5 Blocarea temporară a accesului utilizatorilor la baza de date

- Decizia de blocare temporară a accesului unui utilizator la baza de date poate avea variate motive: nerespectarea de către utilizator a unor norme/ regulamente, plecarea în concediu pe perioada de vara (odată cu sigilarea birourilor) etc.
- Aceste blocări temporare sunt reversibile.

### 3. Partea II: Implementarea configurației de management a identităților în baza de date

Pentru a implementa configurația de management a identităților, sunt necesare mai multe etape, ce vor fi efectuate de către administratorul bazei de date.

#### 3.1 Crearea conturilor utilizatorilor

- Reamintim că autentificarea înseamnă verificarea identității unui utilizator. Prezintă două modalități de autentificare dintre cele suportate de Oracle.
- **Prima modalitate este autentificarea locală, la nivelul bazei de date.** Aceasta se realizează prin nume de utilizator și parolă.
  - Se poate specifica o clauză care să forțeze utilizatorul să își schimbe parola la prima logare în cont.
  - În mod necesar, pentru a permite noului utilizator creat să se conecteze la baza de date, acestuia trebuie să îi oferim privilegiul (dreptul) de creare de sesiune.
  - **Sintaxa:**

```
CREATE USER student1 IDENTIFIED BY parolastudent PASSWORD EXPIRE;  
GRANT CREATE SESSION TO student1;
```

#### Observații importante:

- Dacă utilizatorul este creat cu clauza de parolă expirată, dar nu i s-a acordat încă privilegiul de creare sesiune, atunci utilizatorul va putea să își modifice parola, dar nu să se conecteze.
- Este recomandată expirarea parolei după crearea contului astfel încât să nu planeze nicio suspiciune că parola contului este cunoscută și de DBA – în termeni de non-repudiare a activității pe baza de date.
- Pentru un cont public de vizitator nu va fi necesară expirarea parolei, aceasta rămânând setată de admin.
- Referitor la parole, acestea sunt stocate criptat în baza de date
- Se aplică regula că numele utilizatorului nu poate depăși 30 caractere.
- **A doua modalitate este autentificarea externă, la nivelul sistemului de operare.** Aceasta presupune că, odată ce utilizatorul s-a logat la nivelul sistemului de operare cu credențiale (nume/parolă), acesta să poată accesa imediat baza de date fără o altă autentificare suplimentară.
  - În rândul specialiștilor, există anumite rezerve referitoare la această dependență de securitatea sistemelor de operare. Un alt aspect se referă la conectarea la distanță (*remote*) utilizând autentificare externă care implică riscul ca, odată divulgat numele utilizatorului, oricine poate aduce un laptop în rețeaua instituției, crea un cont în sistemul de operare de pe laptop cu numele respectiv și avea astfel acces la baza de date.

- Se utilizează următoarea schemă de compunere a numelui utilizatorului din baza de date pe baza numelui utilizatorului din sistemul de operare (de exemplu, Windows), în cazul în care utilizatorul are cont în sistemul de operare de pe server:

os-user	os_authent_prefix	database user
MM-S101\TOM	"OPS\$"	"OPS\$MM-S101\TOM"

unde:

→ *os-user* = USERDOMAIN\username – se determină în linie de comandă (în terminal) la nivelul sistemului de operare Windows, sub o sesiune de lucru a utilizatorului nou:

```
echo %username%
echo %USERDOMAIN%
```

→ *os\_authent\_prefix* – se determină cu ajutorul interogării (implicit este OPS\$):

```
SELECT value FROM v$parameter
WHERE name = 'os_authent_prefix';
```

- **Sintaxa:**

- \* pentru utilizatorii creați pe server (cu sistem de operare Windows):

```
CREATE USER "OPS$domeniu\nume" IDENTIFIED EXTERNALLY;
GRANT CREATE SESSION TO "OPS$domeniu\nume";
GRANT CONNECT TO "OPS$domeniu\nume";
```

- \* pentru utilizatorii creați pe stații de lucru:

```
CREATE USER "host\username" IDENTIFIED EXTERNALLY;
GRANT CREATE SESSION TO "host\username";
GRANT CONNECT TO "host\username";
```

```
ALTER SYSTEM SET REMOTE_OS_AUTHENT=TRUE SCOPE=SPFILE;
[SHUTDOWN IMMEDIATE & STARTUP]
```

**Observație:** Pentru utilizatorii externi nu funcționează regula de expirare a parolei și nici nu se stochează vreo parolă în dicționarul datelor. Se aplică, însă, regula ca lungimea numelui utilizatorului, OPS\$domeniu/nume, să nu poată depăși 30 de caractere.

- Consultarea vizualizării **DBA\_USERS** oferă informații despre utilizatorii creați.

```
SELECT USERNAME, AUTHENTICATION_TYPE, ACCOUNT_STATUS, CREATED,
       EXPIRY_DATE
FROM DBA_USERS
WHERE USERNAME LIKE '%ELEARN_%' ORDER BY CREATED;
```

### 3.2 Stabilirea aspectelor referitoare la spații de stocare pentru utilizatorii creați

- Imediat după creare, fără configurări suplimentare, se poate afla valoarea implicită a spațiilor tabel asignate utilizatorilor creați, cu ajutorul interogării:

```
SELECT USERNAME, DEFAULT_TABLESPACE, TEMPORARY_TABLESPACE
FROM DBA_USERS
```

```
WHERE USERNAME LIKE '%ELEARN_%'
ORDER BY CREATED;
```

USERNAME	DEFAULT_TABLESPACE	TEMPORARY_TABLESPACE
ELEARN_APP_ADMIN	USERS	TEMP
ELEARN_STUDENT1	USERS	TEMP
ELEARN_STUDENT2	USERS	TEMP
ELEARN_STUDENT3	USERS	TEMP
ELEARN_STUDENT4	USERS	TEMP
ELEARN_STUDENT5	USERS	TEMP
ELEARN_STUDENT6	USERS	TEMP
ELEARN_STUDENT7	USERS	TEMP
ELEARN_STUDENT8	USERS	TEMP
ELEARN_STUDENT9	USERS	TEMP
ELEARN_STUDENT10	USERS	TEMP

USERNAME	DEFAULT_TABLESPACE	TEMPORARY_TABLESPACE
ELEARN_PROFESOR1	USERS	TEMP
ELEARN_PROFESOR2	USERS	TEMP
ELEARN_ASISTENT3	USERS	TEMP
ELEARN_GUEST	USERS	TEMP
OPS\$MM-33C58500149B\ELEARN_CAT	USERS	TEMP

16 rows selected.

- **Sintaxa:**

```
ALTER USER numeuser QUOTA cota ON nume_tablespace;
```

- În acest mod, unui utilizator îi poate fi setată cota pe *tablespace*-ul implicit (*USERS*) sau pot să i se aloce spații în *tablespace*-uri noi nominalizate.
- Valorile posibile ale cotei pot fi:
  - unlimited – nelimitat;
  - 0 – caz în care utilizatorul nu mai poate crea niciun obiect în viitor în respectivul *tablespace*;
  - o valoare exprimată în MB , de exemplu 10M.

**Observație:** Dacă se dorește ca spațiul de stocare pentru utilizatorii  $U_1, U_2, \dots, U_n$  într-un *tablespace* să fie  $x_1, x_2, \dots, x_n$  MB , parcurgem pașii:

- 1) aflăm dimensiunea totală a *tablespace*-ului respectiv, în MB:

```
SELECT tablespace_name, ROUND(SUM(bytes) / 1048576) TotalSpace
FROM dba_data_files
GROUP BY tablespace_name;
```

TABLESPACE_NAME	TOTALSPACE
UNDOTBS1	55
SYSAUX	580
USERS	5
SYSTEM	680
EXAMPLE	100

- 2) ne asigurăm că este fezabilă configurația propusă (nu depășește dimensiunea *tablespace*-ului)
- 3) setăm cota  $x_i$  pentru fiecare utilizator  $U_i, i = 1, \dots, n$ .

```
ALTER USER Ui QUOTA xi ON nume_tablespace;
```

### 3.3 Stabilirea accesului la resurse computaționale și alte configurări

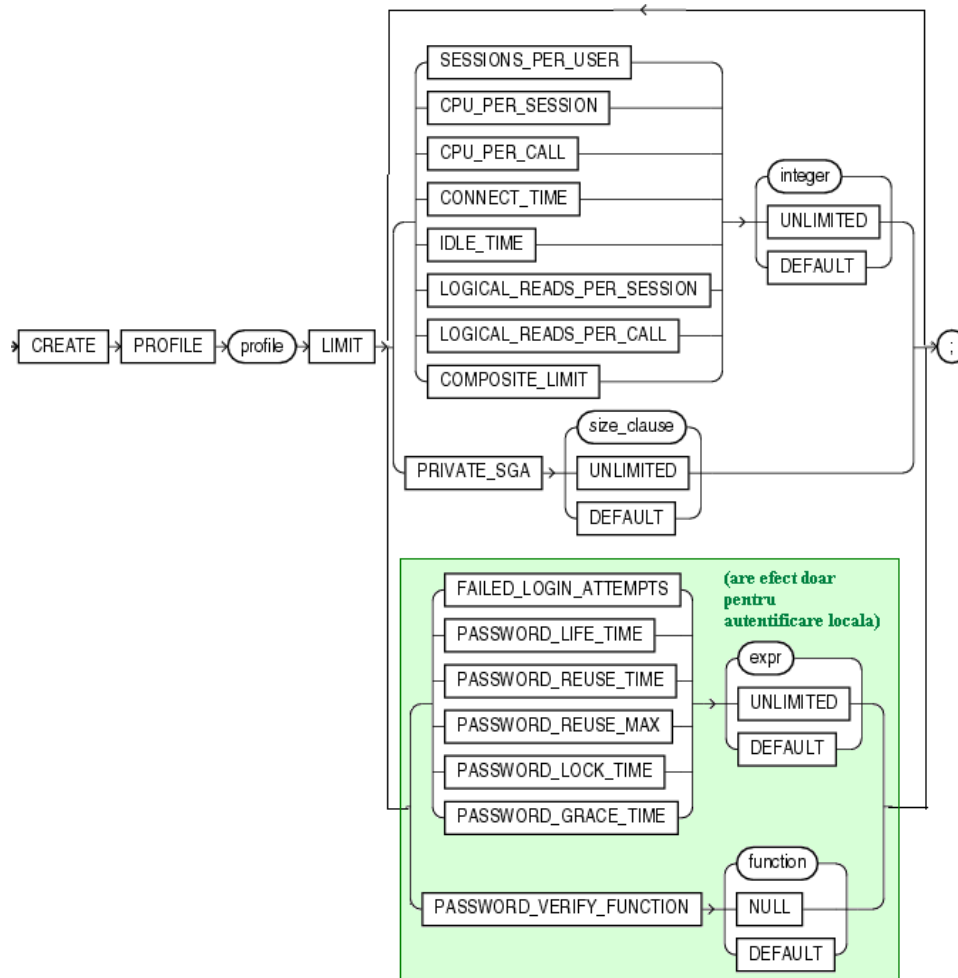
Vom prezenta două modalități de a limita accesul la resurse computaționale în cadrul Oracle: profiluri de resurse și planuri de consum.

- **Prima modalitate este de a grupa restricțiile de acces la resurse în profiluri și a atașa profiluri la utilizatori.**
  - Următoarele etape sunt folosite pentru setarea limitelor de acces la resurse computaționale pentru un utilizator:
    - 1) Se creează un profil de limitări de resurse computaționale, în care se includ valori pentru parametrii doriți. Opțional, în profil pot fi incluse și limitări privind parola unui utilizator (care au efect numai în cazul autentificării locale).
    - 2) Se atașează contului utilizatorului profilul de limitări. Din acest moment, activitatea utilizatorului se desfășoară întru totul sub auspiciile noului profil.

*Reținem că între utilizator și profilul de limitare a accesului la resurse computaționale este o relație m:1, adică un utilizator are 1 singur profil (ultimul ce i-a fost configurat), dar același profil poate fi atașat mai multor conturi de utilizatori.*

- **Sintaxa:**
  - 1) Crearea unui profil de limitări de resurse computaționaleÎn diagrama de pe pagina următoare se observă că la nivelul parolei se poate seta o funcție de verificare (*PASSWORD\_VERIFY\_FUNCTION*) care să asigure că parolele urmează anumite reguli de întocmire. Signatura acestei funcții este:

```
CREATE OR REPLACE FUNCTION my_verification_function (numeuser  
  VARCHAR2, parola_noua VARCHAR2, parola_veche VARCHAR2) RETURN  
  BOOLEAN AS ...
```



**Observație:** Între parametri nu se pun virgule. Unitățile de măsură se regăsesc în documentație ([https://docs.oracle.com/database/121/SQLRF/statements\\_6012.htm#SQLRF01310](https://docs.oracle.com/database/121/SQLRF/statements_6012.htm#SQLRF01310)).

2) Atașarea noului profil de limitări către contul utilizatorului *numeuser*:

```
ALTER USER numeuser PROFILE numeprofil;
```

- Informații despre configurația din fiecare profil se obțin cu ajutorul cererii:

```
SELECT * FROM DBA_PROFILES WHERE PROFILE = 'numeprofil';
```

- Informații privind profilul curent de resurse al unui utilizator se obțin prin interogarea:

```
SELECT USERNAME, PROFILE FROM DBA_USERS;
```

- A doua modalitate este de a grupa utilizatorii în grupuri de consum și a stabili un plan de alocare a resurselor pe aceste grupuri.

Următoarele etape sunt folosite pentru crearea unui plan simplu de consum, în cadrul unui bloc PL/SQL:

- 1) Crearea unei zone de lucru pentru definirea planului;

**Sintaxa:**

```
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

- 2) Crearea planului de consum, care este un container pentru directivele de plan;

**Sintaxa:**

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN
(PLAN => 'numeplan',
COMMENT => 'Acesta este un plan pentru ...');
```

- 3) Crearea grupurilor de consum – reprezintă grupuri de sesiuni utilizator care vor partaja aceeași cantitate de resurse;

**Sintaxa:**

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
(CONSUMER_GROUP => 'nume_grup',
COMMENT => 'Acesta grupeaza sesiunile utilizatorilor care...');
```

- 4) Crearea unor mapări statice între utilizatori și grupurile de consum (detalii privind re-mapări dinamice la runtime, după momentul login, în documentație: [https://docs.oracle.com/database/121/ARPLS/d\\_resmgr.htm#ARPLS74594](https://docs.oracle.com/database/121/ARPLS/d_resmgr.htm#ARPLS74594))

**Sintaxa:**

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
(DBMS_RESOURCE_MANAGER.ORACLE_USER, 'nume_utilizator',
'nume_grup');
```

- 5) Crearea directivei de plan – specificarea, în cadrul planului, a unor reguli de alocare a resurselor pentru fiecare grup de consum:

**Sintaxa:**

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE
(PLAN => 'numeplan', GROUP_OR_SUBPLAN => 'nume_grup',
COMMENT => 'comentariu...', PARAM1 => VAL1, PARAM2=> VAL2, ...);
```

- 6) Validarea și finalizarea implementării planului de consum.

**Sintaxa:**

```
DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

**Observații importante:**

- Dintr-un sistem cu planuri de consum nu trebuie să lipsească grupul de consum *OTHER\_GROUPS*, care să corespundă cazului implicit. În caz contrar apare eroarea:

```
SQL> exec ELEARN_plan_consum
BEGIN ELEARN_plan_consum; END;

*
ERROR at line 1:
ORA-29382: validation of pending area failed
ORA-29377: consumer group OTHER_GROUPS is not part of top-plan ELEARN_PLAN1
ORA-06512: at "SYS.DBMS_RMIN", line 437
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 798
ORA-06512: at "SYS.ELEARN_PLAN_CONSUM", line 78
ORA-06512: at line 1
```

Dacă există în sistem (a se verifica în vizualizarea *DBA\_RSRC\_CONSUMER\_GROUPS*), atunci el nu poate fi recreat, nici șters.

- Conform documentației Oracle, dacă utilizarea CPU este sub 100%, regulile referitoare la



cotele de CPU din planul de consum NU se aplică. Acestea au efect doar când sistemul este supra-încărcat.

- Se pot afla informații despre planurile de consum și directivele de plan prin interogarea vizualizărilor *DBA\_RSRC\_PLANS* și *DBA\_RSRC\_PLANS\_DIRECTIVES*.

### 3.4 Blocarea temporară a accesului utilizatorilor la baza de date

#### Sintaxa:

1) **Varianta 1:** revocarea privilegiului de a se conecta la baza de date:

```
REVOKE CREATE SESSION FROM numeuser;
```

Privilegiul se poate restabili prin comanda:

```
GRANT CREATE SESSION TO numeuser;
```

**Observație:** În funcție de rolurile și privilegiile unui utilizator, există posibilitatea ca acesta să se poată conecta în continuare la baza de date și după retragerea privilegiului. În laboratorul 4 vom discuta pe larg aspecte referitoare la privilegii și roluri.

2) **Varianta 2:** blocarea contului

```
ALTER USER numeuser ACCOUNT LOCK;
```

Contul poate fi deblocat prin comanda:

```
ALTER USER numeuser ACCOUNT UNLOCK;
```

## 4. Exerciții

**1. Conform celor discutate la punctul 2.2.1, se vor crea în sistem următoarele conturi de utilizatori:**

<b>* cu autentificare locala:</b>	1 administrator : ELEARN_APP_ADMIN
	10 studenți: ELEARN_student1,... , ELEARN_student10
	3 cadre didactice: ELEARN_profesor1, ELEARN_profesor2, ELEARN_asistent3
	1 cont de vizitator: ELEARN_GUEST
<b>* cu autentificare la nivelul sistemului de operare:</b>	1 gestionar catalog: ELEARN_CAT

**2. Să se creeze doi trigger-i pentru auditul conectărilor la baza de date: un trigger va monitoriza și înregistra acțiunile *logon*, iar cel de-al doilea trigger va înregistra încheierea sesiunilor deschise anterior.**

Rezultatele monitorizării vor fi stocate în tabelul *ELEARN\_AUDIT\_CONEX*, în care se vor reține: nume utilizator, id de sesiune, tipul identificării, identitatea, *host*-ul de la care s-a conectat, momentul *logon*, momentul *logout*.

**3. Conform celor discutate la punctul 2.2.3, să se aloce utilizatorilor cotele de spațiu de stocare al *tablespace*-ului implicit *USERS*:**

Dimensiunea sa va fi împărțită în cote astfel:

- utilizatorul *ELEARN\_APP\_ADMIN* : nelimitat
- utilizatorul *ELEARN\_CAT* : 10MB
- utilizatorii profesori *ELEARN\_profesor1*, *ELEARN\_profesor2*, *ELEARN\_asistent3*: câte 2MB fiecare
- restul utilizatorilor: 0MB → ei nu vor putea crea obiecte.

**4. Să se creeze profiluri de utilizatori conform celor discutate la punctul 2.2.4) :**

- a) pentru vizitatorii în aplicație (*ELEARN\_GUEST*) numărul maxim de conexiuni permise să fie 5, timpul maxim de inactivitate permis (*idle*) să fie de 3 minute, iar timpul total de conectare să nu depășească 15 minute chiar în perioadele de activitate;
- b) pentru utilizatorii profesori și utilizatorii studenți, consumul de CPU per apel să nu depășească pragul de 60 secunde, să aibă dreptul la o singură sesiune la un moment dat, timpul de viață al parolei să fie de 7 de zile și să poată greși parola de maxim 10 ori.

**5. Să se creeze o procedură care configurează un plan de resurse cu următoarele reguli:**

Vor exista 4 grupuri de consum: management (admin, *ELEARN\_CAT*), tutori (profesori, asistenți) și receptori (studenți, *GUEST*), ceilalți cu cote de consum de CPU 20%, 30%, 40%,10% respectiv.