Lab 3. Gestiunea utilizatorilor unei baze de date și a resurselor computaționale la care aceștia au acces

Cuvinte cheie: • Managementul identităților • Autentificare (locală,externă) • Autorizare	• Schema unui utilizator • Profil de resurse • Plan de consum de resurse • DBMS_RESOURCE_MANAGER

Managementul identităților, in termeni succinți, reprezintă răspunsul la două întrebări: *cine este utilizatorul* și *ce resurse poate accesa el*.
Elementele constituente ale unei arhitecturi de management a identităților într-un sistem informatic, în general, sunt:

- autoritatea centrală care acorda identități;
- modalitățile de verificare a identității (***autentificare***);
- mecanismele de propagare a identității, în cazul unei conectări indirecte/mediate;
- planurile de acces la resurse informaționale si de calcul(***autorizare***).

În continuare, materialul este structurat într-o primă parte de design și o a doua parte de implementare a unei configurații de management a identităților într-o bază de date.

I. PARTEA DE DESIGN A CONFIGURATIEI DE MANAGEMENT A IDENTITATILOR IN BAZA DE DATE

Pentru a proiecta o configurație de management a identitatilor 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 in funcții din aplicație necesare pentru efectuarea lor (workflow-uri);
3) construirea matricii proces – utilizator care sa evidențieze necesitatile de acces la resurse ale utilizatorilor;
4) identificarea entitatilor din modelarea datelor aplicației;
5) construirea matricii entitate – proces;
6) construirea matricii entitate – utilizator rezultata din matricile proces-utilizator si entitate-proces.

Sa parcurgem aceste etape in ordine in contextul unei aplicații de e-learning . Familiarizarea masteranzilor cu proiectul se va realiza ca tema înaintea laboratorului.

1) Cine sunt utilizatorii aplicației? Ce atribute de identificare au aceştia?

Studenții (cu frecventa/ la distanta) – număr matricol, nume-prenume-grupa, CNP
Cadrele didactice (Profesorii, asistenții) – nume-prenume-departament, departament - poziția in statul de funcții, CNP
Secretarii – CNP
Administratorul aplicației si al bazei de date– CNP
Alumni(absolvenți) – număr matricol expirat, cod diploma emisa, CNP
Publicul larg – CNP

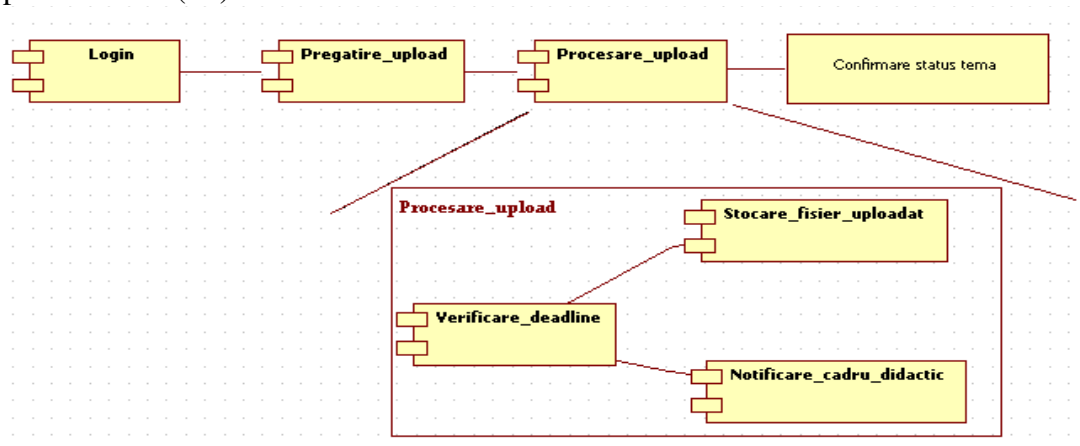
La nivelul tuturor acestor utilizatori, CNP-ul este un atribut de identificare.
Un alt atribut de identificare il reprezintă adresa de e-mail confirmata. In viitor, pe măsura evoluțiilor in domeniu, vor putea fi luate in considerare si elementele biometrice drept atribute de identificare.

Insa, majoritatea acestor informații pot fi aflate si deci nu pot constitui singure baza pentru verificarea identitatii (adica pentru ***autentificare***). De aceea, in mod uzual se construieşte un tandem *atribut de identificare – parola*.

2) Care sunt procesele din cadrul aplicației?

P1: Configurare curs
P2: Vizualizare fisa 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 si notare tema pentru acasă
P9: Consemnare nota pentru sesiune de evaluare
P10: Vizualizare nota consemnata intr-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 frecventeaza un cursant

Care este descompunerea funcțională pe 2 nivele a acestor procese? (*tema de gandire*)
Sa prezentam o descompunere funcțională pe doua nivele a procesului de „Trimitere tema pentru acasă”(P7)



Pentru moment sa notam ca poate exista o granița in acest flux pana la care sa se propage identitatea utilizatorului, după care in etapele următoare aceasta este înlocuita cu o alta identitate.

3) Construirea matricii proces – utilizator

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
Studenții cu frecventa		X					X			X	X			X	X
Studenții la distanta		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 (absolventii)		X													
Administratorul aplicației	X	X	X										X		X
Publicul larg		X													

4) Identificarea entitatilor prin modelarea datelor aplicației

Sistemul de e-learning este o platforma pentru invatamant modern.

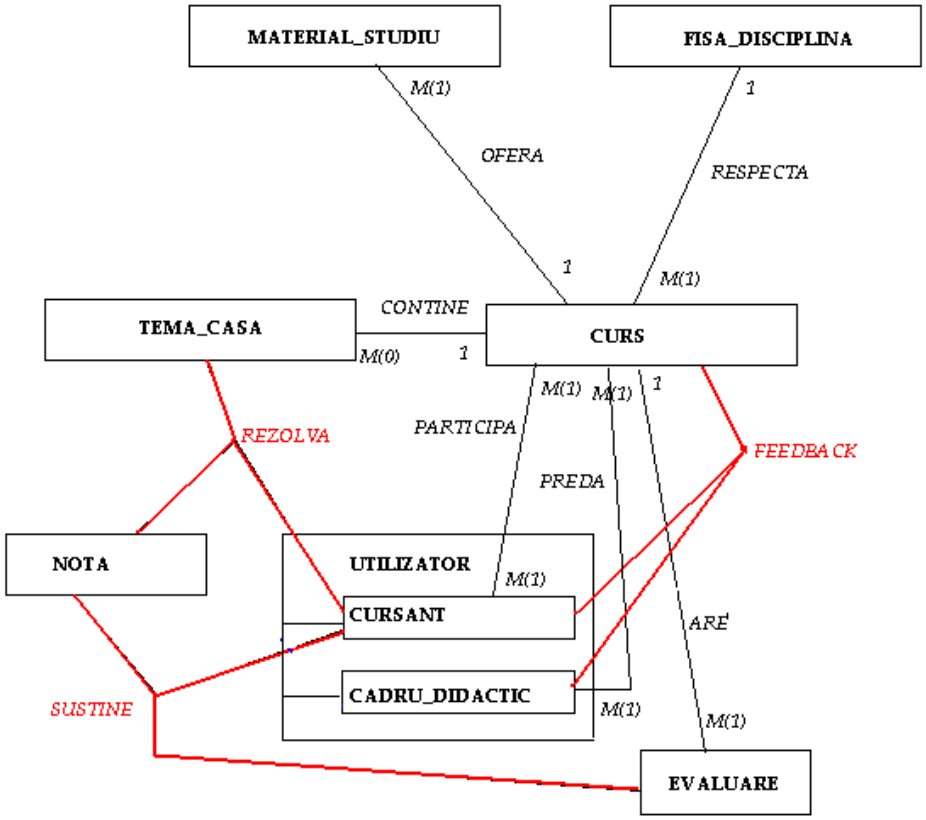
In sistem pot fi configurate cursurile dintr-un an calendaristic universitar, care respecta fisa disciplinei din planul de invatamant publicat. Aceeaşi fisa a disciplinei poate sta la baza mai multor cursuri predate in ani succesivi.

La un curs (dintr-un an calendaristic) predau unul sau mai multe cadre didactice, in calitate de Profesori sau de asistenți. Cursul este destinat cursantilor (studenti, masteranzi) din promoția curenta si celor restanțieri, care vor fi inregistrati la cursul aflat in desfasurare pana reuşesc sa promoveze disciplina in cauza.

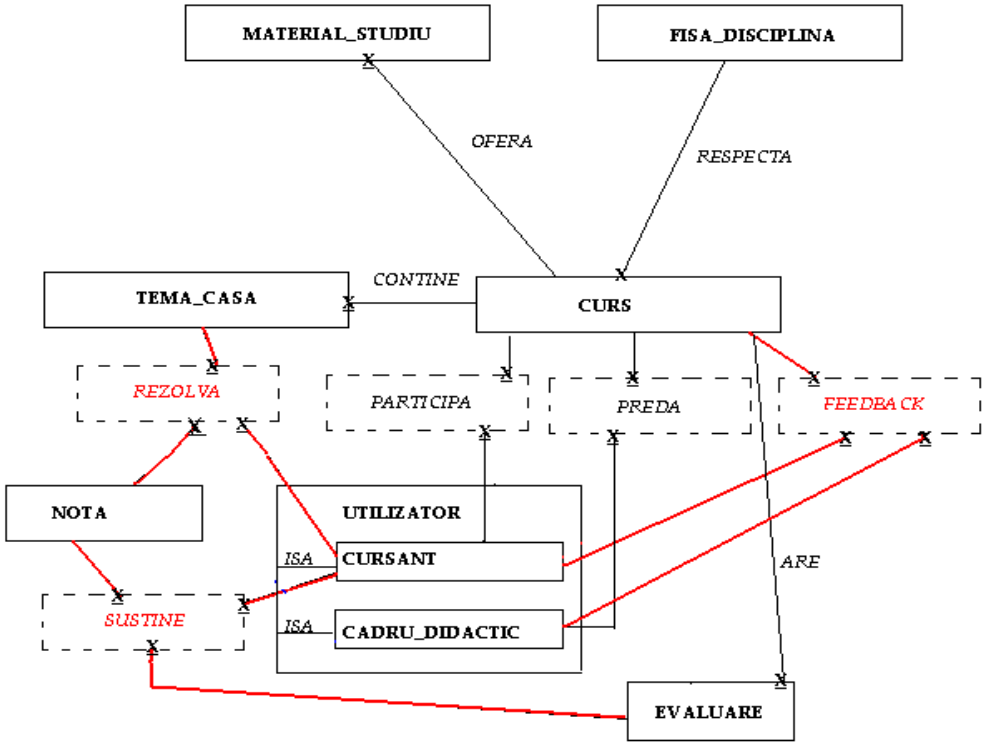
Cadrele didactice ale unui curs pot oferi materiale de studiu studenților, intr-o maniera centralizata. Totodata, sistemul permite publicarea si colectarea temelor de casa intr-o maniera unitara si riguroasa cu respect la termene. Pentru fiecare tema de casa rezolvata cursantul primeşte o nota. Catalogul virtual permite consemnarea notelor pentru teme si a celor rezultate din evaluarea cursantilor, si integrarea cu alte sisteme informatice administrative ale Facultatii.

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

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



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



----------Diagrama conceptuala----------

Schemele relaționale:

FISA_DISCIPLINA (COD_FISADISCIPLINA#, 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)
PREDA (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)

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
PREDA	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

6) 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

In continuare in aceasta parte de design a configurației de management a identitatii in baza de date, informațiile din matricea entitate-utilizator vor fi analizate si interpretate. Întrebările la care se cauta răspuns in continuare:

7) Cine sunt utilizatorii bazei de date, ce conturi de utilizatori vom crea?

In etapa 1) am descoperit clase de utilizatori. Insa, din motive de audit, de non-repudiere, conturile vor fi definite individual pentru utilizatorii identificați din mediul intern organizației. Un alt motiv este ca daca s-ar crea un singur cont de student, de exemplu, exista restricții privind numărul conexiunilor paralele admise (sessions_per_user) si astfel unor studenți le va fi refuzata conexiunea, fiind rugați sa aştepte pana 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, cate unul pentru fiecare student al Facultatii, 50 de conturi de cadre universitare, cate unul pentru fiecare cadru didactic, 5 conturi de secretari.

Doua conturi speciale vor mai fi create: un cont de vizitator (ELEARN_GUEST) pentru publicul larg si un cont de administrator de aplicație (ELEARN_APP_ADMIN).

8) Cine este proprietarul fiecărui obiecte al bazei (tabele, indecşi, etc)?

Proprietarul unui obiect al bazei de date este acel utilizator care are drepturi nelimitate de utilizare si administrare asupra obiectului respectiv, drepturi care nu ii pot fi revocate de către nimeni (nici chiar de administratorul bazei de date!).

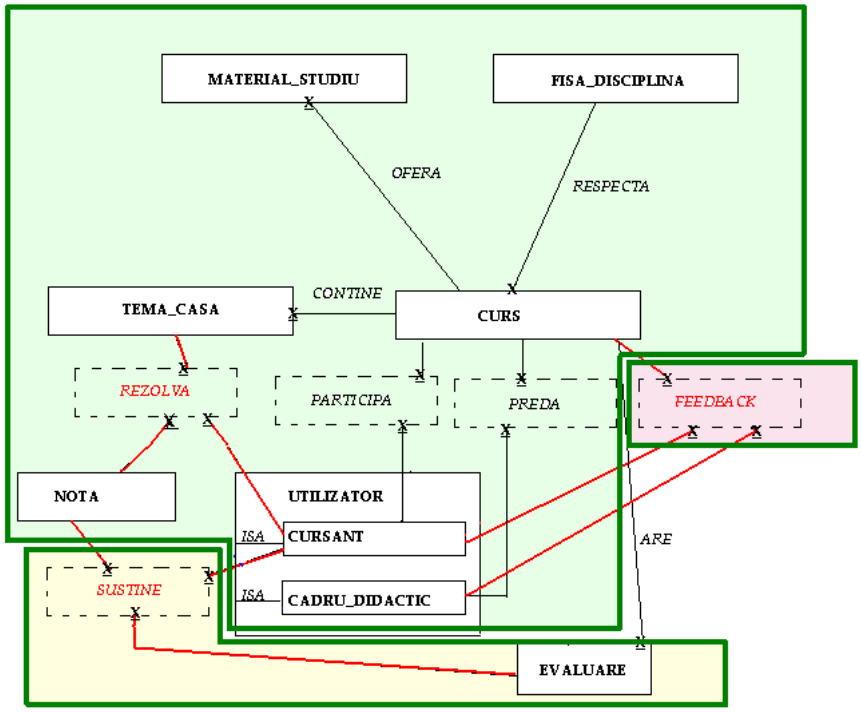
Un utilizator care este proprietar a cel puțin un obiect in baza de date nu poate fi şters din sistem.

Schema unui utilizator reprezintă totalitatea obiectelor (*ex:* tabele, view-uri, indecsi, triggeri la nivelul bazei de date, secvențe, sinonime, funcții PL/SQL, proceduri PL/SQL, pachete PL/SQL) care au ca proprietar un anume utilizator al bazei de date.
Reținem ca intre un utilizator si o schema de utilizator este întotdeauna o relație 1:1.

O buna practica când dezvoltam o aplicație cu o baza de date este sa cream 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 in schema utilizatorului administrator de aplicație.
Excepțiile sunt permise in măsura in care este justificat overhead-ul administrării diferențiate;

Pentru aplicația noastră de e-learning, sa presupunem ca exista următoarele cerințe in specificațiile de sistem:

- tabelele EVALUARE si SUSTINE sa fie deținute de un proprietar separat ELEARN_ CAT;
- tabela FEEDBACK sa fie deținuta de fiecare cadru didactic individual din sistem; cu alte cuvinte daca avem 50 de utilizatori cadre didactice, in schema fiecăruia vom avea cate o tabela FEEDBACK;
- restul tabelelor vor avea ca proprietar administratorul de aplicație ELEARN_APP_ADMIN.



Evidențierea schemelor de utilizator pe diagrama conceptuala

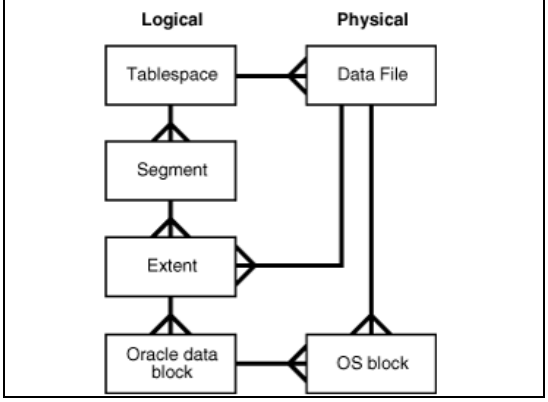
9) Stabilirea aspectelor de spatii de stocare pentru utilizatorii creați

Aşa cum s-a stabilit la punctul 8), anumiți utilizatori vor fi proprietari de tabele, alții nu. Prin urmare este firesc sa consideram o impartire diferențiata a spațiului de stocare intre utilizatori.

Un utilizator al bazei de date dispune de doua spatii de stocare:

- spațiu de stocare pentru obiectele din schema sa, peste care el este deci proprietar;
- spațiu de stocare temporar, folosit de Oracle in diferite situații (de exemplu, in sortarea înregistrărilor când procesează o cerere cu clauza ORDER BY; sau in constuctia tabelei de hash in etapa build din procesarea join prin metoda HASH JOIN).

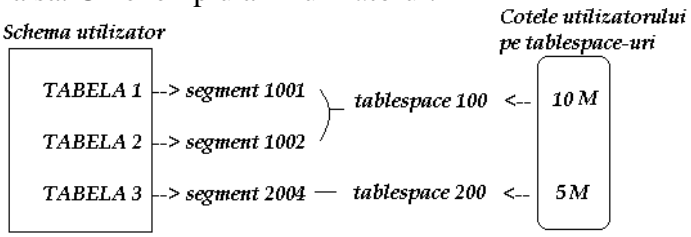
Sursa: http://docs.oracle.com/cd/E11882_01/server.112/e25789/logical.htm



Fara a intra in detalii referitoare la tablespace-uri in Oracle, reținem următoarele:

- tablespace-ul este o unitate *logica* de stocare, formata din segmente; un segment stochează datele unui singur obiect al unui utilizator (1 tabel nepartitionat si fara CLOB = 1 segment, 1 index nepartitionat = 1 segment, s.a.m.d.)
- tablespace-ul face legătura intre baza de date si sistemul de fişiere, întrucât un tablespace are asociat cel puțin un datafile (un fişier 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 impartita in cote astfel:

- utilizatorul ELEARN_APP_ADMIN : nelimitat

- utilizatorul ELEARN_CAT : 10MB
- utilizatorii profesori ELEARN_profesor1 , ELEARN_profesor2, ELEARN_asistent3 cate 2MB fiecare
- restul utilizatorilor 0MB→ ei nu vor putea crea obiecte.

10) Stabilirea accesului la resurse computaționale si alte configurări

Următorul pas, după crearea conturilor de utilizatori si stabilirea spatiilor de stocare, il reprezintă impunerea unor limitări referitoare la accesul la resurse pentru utilizatori. Scopul este de a asigura o funcționare performanta a bazei de date, evitarea monopolului de către un utilizator asupra resurselor, s.a.

Parametrii de performanta care sunt adesea regasiti in aceste configurări se refera la:

- timpul de execuție estimat al instrucțiunilor;
- consumul de CPU;
- gradul de paralelism acceptat in sisteme multi-procesor;
- numărul de sesiuni deschise per utilizator;
- timp de inactivitate (idle).

O buna practica este de a reutiliza setările privind accesul la resurse prin crearea de profile sau a unor planuri de consum.

Un profil de resurse reprezintă[1] „gruparea sub un titlu a unei mulțimi de limitări de resurse care restrictioneaza utilizarea bazei de date si a resurselor instanței Oracle pentru un utilizator". La un moment dat, un utilizator poate avea un singur profil de resurse. Acelaşi profil de resurse poate fi setat mai multor utilizatori.

Privind aplicația de e-learning, se solicita :

a) pentru vizitatorii in aplicație (ELEARN_GUEST) numărul maxim de conexiuni permise sa fie 5, timpul maxim de inactivitate permis (idle) sa fie de 3 minute, iar timpul total de conectare sa nu depaseasca 15 minute chiar in perioadele de activitate;

b) pentru utilizatorii profesori si utilizatorii studenți, consumul de CPU per apel sa nu depaseasca pragul de 60 secunde, sa aibă dreptul la o singura sesiune la un moment dat, timpul de viata al parolei sa fie de 7 de zile si sa poată greşi de maxim 10 ori parola.

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 in care rădăcina este planul principal, in frunze sunt grupurile de consum , iar pe nivelele intermediare sunt sub-planurile de consum.

11) Blocarea temporara a accesului utilizatorilor la baza de date

Decizia de blocare temporara a accesului unui utilizator la baza de date poate avea variate motive: nerespectarea de către utilizator a unor norme/ regulamente, plecarea in concediu pe perioada de vara (odată cu sigilarea birourilor), etc.

Aceste blocări temporare sunt reversibile.

---

[1] http://docs.oracle.com/cd/B19306_01/network.102/b14266/admusers.htm#i1012785

II. PARTEA DE IMPLEMENTARE A CONFIGURATIEI DE MANAGEMENT A IDENTITATILOR IN BAZA DE DATE

Pentru a implementa configuratia de management a identitatii, sunt necesare mai multe etape, efectuate de către administratorul bazei de date.

II.1. Crearea conturilor utilizatorilor

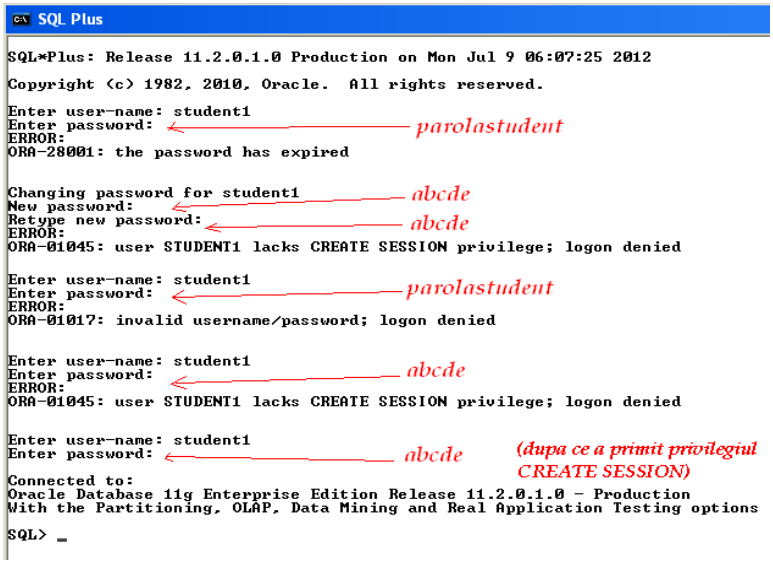
Reamintim ca autentificarea înseamnă verificarea identitatii unui utilizator.
Prezentam doua modalitati de autentificare dintre cele suportate de Oracle.

Prima modalitate este autentificarea locala, la nivelul bazei de date. Aceasta se realizează prin nume de utilizator si parola. Se poate specifica o clauza care sa forțeze utilizatorul sa isi schimbe parola la prima logare in cont.
In mod necesar, pentru a permite noului utilizator creat sa se conecteze la baza de date, acestuia trebuie sa ii oferim privilegiul (dreptul) de creare sesiune.

Sintaxa:

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

Observatii importante:
- daca utilizatorul este creat cu clauza de parola expirata, dar nu i s-a acordat inca privilegiul de creare sesiune, atunci utilizatorul va putea sa isi modifice parola, dar nu sa se conecteze.



- este recomandata expirarea parolei după crearea contului astfel încât sa nu planeze nici o suspiciune ca parola contului este cunoscuta si de DBA in termeni de non-repudiere a activitatii pe baza de date
- pentru un cont public de vizitator, nu se va expira parola, ea ramanand setata de admin
- referitor la parole, acestea sunt stocate criptat in baza de date
Se aplica regula ca numele utilizatorului nu poate depasi 30 caractere.

A doua modalitate este autentificarea externa, la nivelul sistemului de operare. Aceasta presupune ca odată ce utilizatorul s-a logat la nivelul sistemului de operare cu credentiale (nume/parola), acesta sa poată accesa imediat baza de date fara o alta autentificare suplimentara.

In rândul specialiştilor, exista anumite rezerve referitoare la aceasta dependenta de securitatea sistemelor de operare. Un alt aspect se refera la conectarea la distanta (remote) utilizând autentificare externa care implica riscul ca odată divulgat numele utilizatorului, oricine poate aduce un laptop in rețeaua instituției, crea un cont in sistemul de operare de pe laptop cu numele respectiv si avea astfel acces la baza de date.

Se utilizează următoarea schema de compunere a numelui utilizatorului din baza de date pe baza numelui utilizatorului din sistemul de operare (ex: Windows), *in cazul in care utilizatorul are cont in 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 determina in linie de comanda (in terminal) la nivelul sistemului de operare Windows, sub o sesiune de lucru a utilizatorului nou:

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

os_authent_prefix - se determina cu interogarea (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]
```

Remarcam ca pentru utilizatorii externi nu functioneaza regula de expirare a parolei si nici nu se stochează vreo parola in dicționarul datelor. Se aplica insa regula ca numele utilizatorului, OPS$domeniu**/**nume, nu poate depasi 30 caractere.

Consultarea 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;
```

II.2. Stabilirea aspectelor de spatii de stocare pentru utilizatorii creați

Imediat după creare, fara configurări suplimentare, se poate afla valoarea implicita a spatiilor 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;

In acest mod unui utilizator ii poate fi setata cota pe tablespace-ul implicit (USERS) sau poate sa i se aloce spatii in tablespace-uri noi nominalizate.
Valorile posibile ale cotei pot fi:

- unlimited – nelimitat;
- 0 – caz in care utilizatorul nu mai poate crea nici un obiect in viitor in respectivul tablespace;
- o valoare exprimata in MB , de exemplu 10M.

Observație: daca se dorește ca spațiul de stocare pentru utilizatorii $U_1,U_2,..U_n$ intr-un tablespace sa fie $x_1,x_2,..x_n$ MB , parcurgem pașii:

1) aflam dimensiunea totala a tablespace-ului respectiv, in MB[2]:

```
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 asiguram ca este fezabila configurația propusa (nu depaseste dimensiunea tablespace-ului)
3) setam cota x_i pentru fiecare utilizator U_i , i=1,n .
ALTER USER Ui QUOTA xi ON nume_tablespace;

II.3. Stabilirea accesului la resurse computaționale si alte configurări

Vom prezenta doua modalitati de a limita accesul la resurse computaționale in cadrul Oracle: profile de resurse si planuri de consum.

Prima modalitate este de a grupa restricțiile de acces la resurse in profile si a ataşa profile la utilizatori.

Urmatoarele etape sunt folosite pentru setarea limitelor de acces la resurse computationale pentru un utilizator:

1) se creaza un profil de limitari de resurse computationale, in care se includ valori pentru parametrii doriti
   - optional, in profil pot fi incluse si limitari privind parola unui utilizator (care au efect numai in cazul autentificarii locale)
2) se ataseaza contului utilizatorului profilul de limitari. Din acest moment, activitatea utilizatorului se desfasoara in intru totul sub auspiciile noului profil.

Retinem ca intre utilizator si profilul de limitare a accesului la resurse computationale este o relatie n:1 , adica un utilizator are 1 singur profil (ultimul ce i-a fost configurat), dar acelasi profil poate fi atasat mai multor conturi de utilizatori.

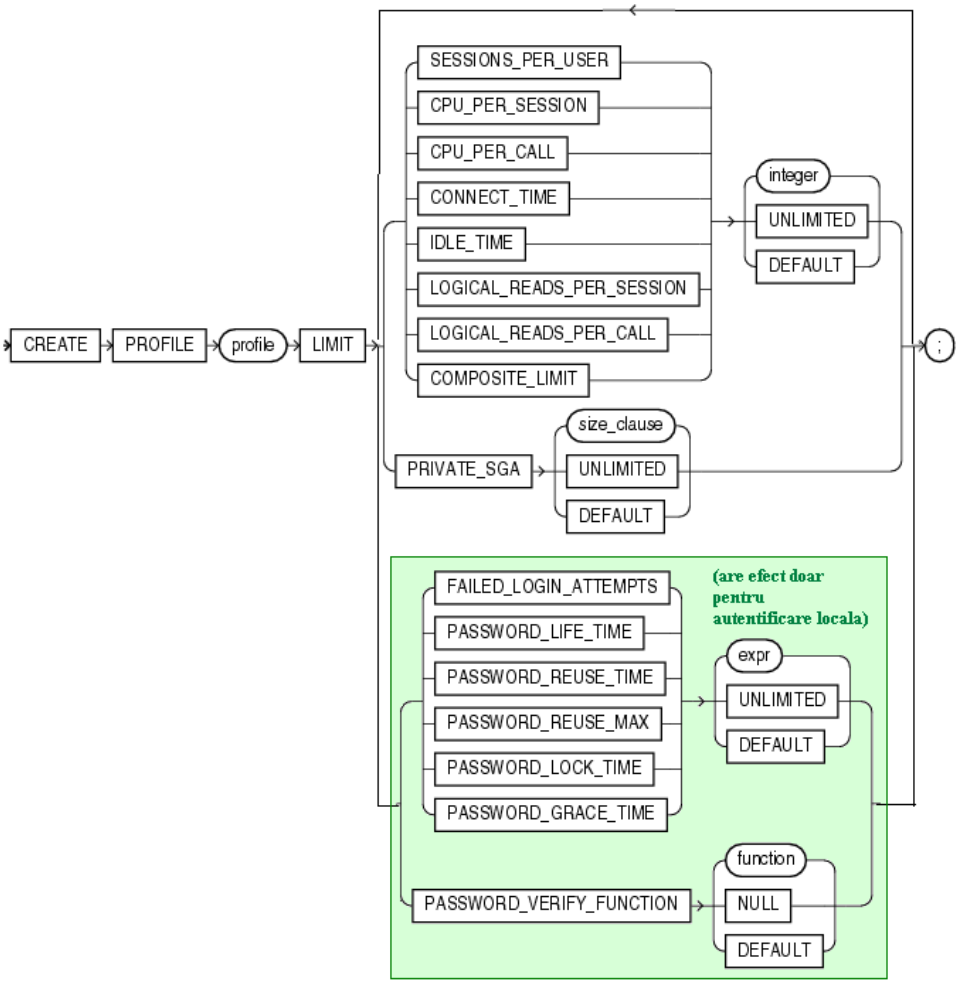
Sintaxa[3]:

1) crearea unui profil de limitari de resurse computationale

In diagrama de pe pagina urmatoare se observa ca la nivelul parolei se poate seta o functie de verificare (PASSWORD_VERIFY_FUNCTION) care sa asigure ca parolele urmeaza anumite reguli de întocmire. Signatura acestei funcții este:

[2] https://forums.oracle.com/forums/thread.jspa?threadID=505636
[3] **http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6010.htm**

CREATE OR REPLACE FUNCTION my_varification_function (
numeuser VARCHAR2,
parola_noua VARCHAR2,
parola_veche VARCHAR2)
RETURN BOOLEAN AS …



Observatie: intre parametrii nu se pune virgule.
Unitatile de masura se regasesc la link-ul specificat in subsolul paginii[4]

2) atasarea noului profil de limitari catre 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

[4] **http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6010.htm**

A doua modalitate este de a grupa utilizatorii in grupuri de consum si a stabili un plan de alocare a resurselor pe aceste grupuri.

Următoarele etape sunt folosite pentru crearea unui plan simplu de consum, in 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 aceeaşi cantitate de resurse;

Sintaxa:

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

4) crearea unor mapari statice intre utilizatori si grupurile de consum (detalii privind re-mapari dinamice la runtime , după momentul login in bibliografie[5])

Sintaxa:

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

5) crearea directivelor de plan – specificarea, in 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 si finalizarea implementării planului de consum

Sintaxa:

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

Observaţii importante:
-dintr-un sistem cu planuri de consum nu trebuie sa lipseasca grupul de consum OTHER_GROUPS, care sa corespunda restului lumii. In caz contrar apare eroarea:

[5] http://docs.oracle.com/cd/E14072_01/server.112/e10595/dbrm004.htm#CHDIGIII

```
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
```

Daca exista in sistem (a se verifica in vizualizarea DBA_RSRC_CONSUMER_GROUPS), atunci el nu poate fi recreat, nici şters.

- conform documentației Oracle[6], daca utilizarea CPU este sub 100%, regulile referitoare la cotele de CPU din planul de consum NU se aplica. Acestea au efect doar când sistemul este supra-incarcat;

- se pot afla informații despre planurile de consum si directivele de plan prin interogarea vizualizărilor DESC DBA_RSRC_PLANS si DESC DBA_RSRC_PLANS_DIRECTIVES.

II.4. Blocarea temporara 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;

Se restabileşte privilegiul prin comanda:

GRANT CREATE SESSION TO numeuser;

Observație importanta: in funcție de rolurile si privilegiile unui utilizator, exista posibilitatea ca acesta sa se poată conecta in continuare la baza de date si după retragerea privilegiului. In laboratorul 4 vom discuta pe larg aspecte de privilegii si roluri.

2). Varianta 2: blocarea contului

ALTER USER numeuser ACCOUNT LOCK;

Se deblochează contul prin comanda:

ALTER USER numeuser ACCOUNT UNLOCK;

[6] http://docs.oracle.com/cd/E14072_01/server.112/e10595/dbrm001.htm#i1007556

Exerciţii:

1. Conform celor discutate la punctul 8) in sistem vom crea următoarele conturi de utilizatori:

*** cu autentificare locala:**	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
*** cu autentificare la nivelul sistemului de operare:**	1 gestionar catalog: ELEARN_CAT

2. Sa scriem doi trigger pentru auditul conectărilor la baza de date: un trigger va monitoriza si înregistra acțiunile logon si cel de-al doilea trigger va inregistra încheierea sesiunilor deschise anterior.
Rezultatele monitorizării vor fi stocate in tabela ELEARN_AUDIT_CONEX , in care se vor retine : nume utilizator, id de sesiune, tipul identificarii, identitatea, host-ul de la care s-a conectat, momentul logon, momentul logout.

3. Conform celor discutate la punctul 9) le oferim utilizatorilor cotele de spațiu de stocare al tablespace-ului implicit USERS:

Dimensiunea sa va fi impartita in cote astfel:

- utilizatorul ELEARN_APP_ADMIN : nelimitat
- utilizatorul ELEARN_CAT : 10MB
- utilizatorii profesori ELEARN_profesor1 , ELEARN_profesor2, ELEARN_asistent3 cate 2MB fiecare
- restul utilizatorilor 0MB→ ei nu vor putea crea obiecte.

4. Sa se creeze profile de utilizatori conform celor discutate la punctul 10) :
a) pentru vizitatorii in aplicatie (ELEARN_GUEST) numarul maxim de conexiuni permise sa fie 5, timpul maxim de inactivitate permis (idle) sa fie de 3 minute, iar timpul total de conectare sa nu depaseasca 15 minute chiar in perioadele de activitate;
b) pentru utilizatorii profesori si utilizatorii studenti, consumul de CPU per apel sa nu depaseasca pragul de 60 secunde, sa aiba dreptul la o singura sesiune la un moment dat, timpul de viata al parolei sa fie de 7 de zile si sa poata gresi de maxim 10 ori parola.

5. Sa se creeze o procedura care configurează un plan de resurse cu următoarele reguli:
4 grupuri de consum: management(admin, ELEARN_CAT), tutori (profesori, asistenți) si receptori(studenți,GUEST), ceilalti cu cote de consum de CPU 20%, 30%, 40%,10% respectiv.

Bibliografie:

[1] Knox David et al (2009) - Applied Oracle Security: Developing Secure Database and Middleware Environments, Part III (Identity management), ISBN 978-0071613705
[2] http://www.oracle-base.com/articles/misc/os-authentication.php
[3] http://docs.oracle.com/cd/E14072_01/server.112/e10595/dbrm004.htm#CHDIGIII