Curs 13. Probleme de securitate
1. Aplicaţii cu mai mulţi utilizatori
2. Probleme de securitate în Access
3. Internetul şi bazele de date Access

1. Aplicaţii cu mai mulţi utilizatori

Toate exemplele pe care le-am prezentat până acum se bazează pe ideea că datele nu vor fi accesate de mai mulţi utilizatori în acelaşi timp. In practică însă, o aplicaţie serioasă este aceea care permite mai multor utilizatori accesul simultan la date pentru a fi vizualizate şi modificate (accesul concurenţial).

Modalităţi de blocare a înregistrărilor

Soluţia pentru problema accesului concurenţial la date constă în blocarea înregistrărilor atunci când un utilizator începe modificarea lor sau când vrea să salveze modificările efectuate. Access oferă două modalităţi de blocare a înregistrărilor: blocare optimistă şi pesimistă.

- Blocarea pesimistă presupune blocarea unei înregistrări atunci când un utilizator începe să modifice datele acesteia.
- Blocarea optimistă înseamnă blocarea unei înregistrări numai atunci când cineva încearcă să salveze modificările pe care le-a efectuat în înregistrarea respectivă.

Să presupunem că mai mulţi utilizatori folosesc, la un moment dat, aceeaşi bază de date şi editează înregistrările aceleiaşi tabele. Ei vor putea, fără nici o problemă, să vizualizeze simultan aceeaşi înregistrare, indiferent de tipul de blocare folosite. Conflictele apar atunci când, de exemplu, un utilizator modifică datele unei înregistrări, introducând o valoare noua într-una dintre câmpuri. In acel moment, dacă se foloseşte blocarea pesimistă, Access va bloca înregistrarea respectivă, astfel încât nici un alt utilizator nu va putea s-o modifice decât după ce au fost salvate modificările efectuate de utilizatorul respectiv. Dacă s-ar fi folosit metoda blocării optimiste, ceilalţi utilizatori ar fi putut efectua şi ei modificări asupra înregistrării respective, în acelaşi timp. Primul utilizator care va termina de efectuat modificările va reuşi să salveze înregistrarea fără nici o problemă. Ce se va întâmpla însă atunci când un alt utilizator va dori să salveze proriile modificări asupra aceleiaşi înregistrări(ale cărei date nu mai sunt aceleaşi ca în momentul în care acest utilizator a început editarea lor)? In acel moment, va apărea o cutie de dialog care îi oferă acestui al doilea utilizator trei posibilităţi:

- **Save Record** – înregistrarea va fi salvată cu noile modificări, suprascriind modificările primului utilizator. Aceasta este o situaţie periculoasă, deoarece nici unul dintre cei doi utilizatori nu ştie ce modificări a făcut celălalt: primul care a salvat înregistrarea nu va şti că altcineva a suprascris-o, iar al doilea nu va şti ce modificări a făcut primul.
- **Copy to Clipboard** – modificările celui de-al doilea utilizator vor fi scrise în clipboard astfel încât el va putea vedea ce schimbări a făcut primul utilizator şi apoi va decide dacă e cazul să le suprascrie.
- **Drop Changes** (renunţă la modificări) – modificările celui de-al doilea utilizator se vor pierde, iar cele ale primului vor rămâne neschimbate.

Un alt lucru important de ştiut despre modul în care Access blochează înregistrările este acela că, spre deosebire de alte sisteme de gestiune a bazelor de date, el blochează pagini de date şi nu înregistrări individuale. O pagină conţine 2 Kb de date şi deci, una sau mai multe înregistrări. (De exemplu, dacă înregistrările au, în medie, 600 de octeţi, într-o pagină se vor afla 3 înregistrări). Dacă un utilizator blochează o înregistrare dintr-o pagină, Access va bloca toate înregistrările din pagina respectivă. Avantajul păstrării mai multor înregistrări într-o pagină este accesarea mai rapidă a datelor.
O înregistrare nu poate ocupa mai mult de o pagină (deci nu poate avea dimensiunea mai mare de 2 Kb). Câmpurile de tip Memo şi OLE sunt păstrate în pagini separate şi nu pot avea dimensiunea mai mare de 1.2 Gb.

Alegerea modalităţii de blocare

Pentru a putea alege modalitatea de blocare cea mai potrivită, este bine să luaţi în considerare avantajele şi dezavantajele celor două strategii.
Blocarea pesimistă are următoarele avantaje:

- uşurează munca programatorului;
- nu lasă utilizatorii să-şi suprascrie unul altuia modificările;
- pune mai puţine probleme utilizatorului.

Dezavantajele blocării pesimiste sunt:

- blocarea mai multor înregitrări decât este necesar (o pagină);
- limitarea accesului concurenţial la date, deoarece permite utilizatorilor să blocheze înregistrări pentru perioade lungi de timp.

Metoda blocării optimiste prezintă următoarele avantaje:

- este mai simplu întrebuinţat;
- oferă un mai bun acces concurenţial;
- probabilitatea de a împiedica alţi utilizatori să modifice datele este mai mică.

Pe de altă parte, dezavantajele blocării optimiste sunt:

- utilizatorii pot fi dezorientaţi la apariţia unui conflict;
- există pericolul ca utilizatorii să-şi suprascrie unul altuia modificările.

Este bine să folosim metoda blocării pesimiste numai dacă aveţi motive serioase să faceţi aceasta pentru că, în acest caz, există posibilitatea să împiedicaţi utilizatorii să facă modificări asupra înregistrărilor pentru perioade de timp destul de lungi.
In unele aplicaţii, puteţi folosi ambele metode de blocare pentru formulare diferite. De exemplu, pentru o aplicaţie ce se ocupă cu problemele de gestiune ale unei firme, veţi avea, să zicem, un formular pentru introducerea comenzilor şi altul pentru datele personale ale angajaţilor. In acest caz, există pericolul ca doi vânzători să modifice simultan cantitatea comandată de clienţi dintr-un produs anumit produs astfel încât prima comandă să epuizeze stocul existent, iar cea de-a doua să se bazeze pe informaţiile existente înainte, fără să ştie că stocul s-a epuizat. Este clar deci, că în cazul formularului pentru comenzi este necesar să folosiţi metoda blocării pesimiste. Pe de altă parte, posibilitatea ca datele personale ale unui angajat să fie modificate simultan este foarte mică, astfel încât pentru acest formular puteţi folosi, fără probleme, metoda blocării optimiste.

Stabilirea metodei implicite de blocare

Pentru a alege metoda implicită de blocare pe care o va folosi Access la crearea diferitelor obiecte din bara de date, folosiţi meniul *Tools* | *Options*, iar în fila *Advanced* a ferestrei *Options*, alegeţi una dintre următoarele trei opţiuni:

- **No Locks** – este opţiunea implicită şi este echivalentul blocării optimiste. In acest caz, pagina de date ce conţine înregistrarea în curs de modificare va fi blocată numai în timpul salvării înregistrării şi nu pe toată perioada editării.
- **All Records** – este metoda blocării exclusive, care blochează toate înregistrările tabelei (sau tabelelor) ce formează setul de înregistrări ce conţine înregistrarea în curs de modificare. Aceasta este o metodă foarte radicală şi ar trebui să fie folosită numai de administratorul bazei de date în timpul efectuării operaţiilor de întreţinere.
- **Edited Record** – este echivalentul blocării pesimiste. Pagina de date ce conţine înregistrarea în curs de modificare va fi blocată din momentul începerii editării până când schimbările au fost salvate.

Aceste opţiuni pot fi stabilite şi prin intermediul codului, apelând metoda *SetOption* a obiectului *Application* cu argumentele "*Default Record Locking*" (metoda implicită de blocare) şi, respectiv, valorile 0,1 sau 2 care specifică blocarea optimistă, exclusivă sau, respectiv, pesimistă.

Exemplu: Următoarea procedură afişează un mesaj care ne informează care este metoda implicită de blocare curentă şi cere confirmarea înainte de a o schimba în funcţie de argumentul primit:

```
Sub ModBlocImpl (iMetoda As Integer)
Dim iMetCrt As Integer
Dim strMet As String
iMetCrt = Application.GetOption ("Default Record Looking")
Select Case iMetCrt
      Case 0
            strMet = "optimista"
      Case 1
                  strMet = "exclusiva"
      Case 2
                  strMet = "pesimista"
End Select
If MsgBox("Metoda implicita de blocare este cea " & strMet  & ".
Doriţi să o modificaţi?, vbOKCancel + vbQuestion, "Blocare") =
vbOK Then
   Application.SetOption "Default Record Loocking", iMetoda
End If
End Sub
```

Stabilirea metodei de blocare pentru obiectele bazei de date

Metoda implicită de blocare este aplicată obiectelor dintr-o bază de date, conform tabelului de mai jos. Dacă doriţi să schimbaţi metoda de blocare folosită pentru un anumit obiect al barei de date (formular, raport sau interogare), stabiliţi proprietatea RecordsLocks a obiectului respectiv şi ţineţi cont de metodele disonibile pentru fiecare obiect prezentat în tabelul următor:

Obiectul	Metode de blocare disponibile	Metoda de blocare implicită	Când se produce blocarea
Tabelă	toate trei	implicită	la editarea în modul Datasheet View
Interogare SELECT	toate trei	implicită	la editarea în modul Datasheet View

Interogare Crossstab	toate trei	implicită	la execuţia interogării
Interogări Union	toate trei	implicită	la execuţia interogării
Interogări UPDATE şi DELETE	pesimistă şi exclusivă	implicită	la execuţia interogării
Interogări MAKETABLE şi APPEND	Pesimistă şi exclusivă	implicită	la execuţia interogării
Interogări pentru definirea datelor	exclusivă	exclusivă	la execuţia interogării
Formulare	toate trei	implicită	la editarea în modul Form View şi Datasheet View
Rapoarte		implicită	la rulare, vizualizare şi tipărire
Setd de înregistrări	toate trei	implicită	între apelul metodelor Edit şi Update

Exemplu: următoarea procedură modifică proprietatea RecordLocks a formularului cu numele strForm, dându-i valoarea iMet:

```
Sub SetFrmBloc(strForm As String,  iMet  As Integer)
Dim frm  As Form
DoCmd.OpenForm strForm
Set frm = Forms (strForm)
Frm.RecordLocks = iMet
End Sub
```

Proprietatea RecordLocks foloseşte aceleaşi trei argumente ca şi metoda Application.SetOption: 0 pentru blocarea optimistă, 1 pentru blocarea exclusivă şi 2 pentru blocarea pesimistă.

2. Probleme de securitate în Access

Dacă baza de date conţine informaţii confidenţiale sau dacă doriţi să o protejaţi faţă de modificarea accidentală a informaţiilor, este necesar implementarea unui mecanism de securitate. Securitatea poate fi definită ca o metodă de a restrânge accesul utilizatorilor la baza de date şi la obiectele pe care aceasta le conţine.
Access pune la dispoziţie două metode de securitate: folosirea unei parole unice pentru baza de date şi securitatea la nivel de utilizator.

Stabilirea unei parole pentru baza de date

Cea mai simplă metodă de implementare a securităţii este atribuirea unei parole pentru întreaga bază de date. Astfel, orice utilizator care va dori să acceseze baza de date va trebui să introducă mai întâi această parolă. După aceea, toate obiectele din baza de date vor fi la dispoziţia sa.
Problema acestei metode de securitate este aceea că nimeni nu vă asigură că un utilizator care cunoaşte parola nu o va spune altuia, care, la rândul lui, o va spune mai departe altor utilizatori care nu ar trebui să o afle. De asemenea, dacă nimeni nu-şi mai aminteşte parola, informaţiile stocate în baza de date respectivă nu vor mai putea fi recuperate.
Pentru a stabili o parolă pentru baza de date folosind interfaţa cu utilizatorul oferită de Access, alegem meniul *Tools* | *Security* |*Set Database Password.*
De asemenea, putem face acest lucru şi prin intermediul VBA, apelând metoda *NewPassword* a obiectului de tip Database. Argumentele acestei metode sunt două şiruri de caractere ce conţin vechea şi, respectiv, noua parolă. Dacă baza de date nu avea o parolă, primul argument va fi şirul vid:

```
Dim db As Database
Set db = CurrentDB()
Db.NewPassword "", "Newton"
```

Astfel, am atribuit parola "Newton" bazei de date curente. Pentru a schimba această parolă cu alta,, de exemplu "Euler", scriem:

```
db.NewPassword "Newton", "Euler"
```

Pentru a înlătura parola existentă, dăm ca argument metodei NewPassword şirul vid pentru noua parolă.

```
db.NewPassword "Euler", ""
```

O modalitate mai performantă de a asigura securitatea unei baze de date este atribuirea unei parole fiecărui utilizator care are voie să acceseze baza de date sau doar anumite obiecte din interiorul acesteia.

Securitatea la nivel de utilizator

Dă posibilitatea de a acorda drepturi diferite utilizatorilor şi grupurilor de utilizatori. Aceasta înseamnă că fiecare utilizator începe sesiunea de lucru prin introducerea unui nume şi a unei parole proprii.
Prin această metodă de securitate, utilizatorii aparţin unor grupuri. Drepturile pot fi atribuite la nivel de grup, la nivel de utilizator sau la ambele niveluri. Un utilizator deţine drepturile de la grupul cel mai puţin restrictiv din care face parte.
In mod implicit, toţi utilizatorii au drepturi asupra tuturor obiectelor bazei de date, fiind membri ai grupului *Users*. Dacă nu aveţi implementată securitatea la nivel de utilizator, toţi utilizatorii se vor conecta cu numele Admin care e membru al grupurilor *Admins* şi *Users*. Inainte de a putea crea noi conturi pentru grupuri şi utilizatori, trebuie să creaţi sau să vă alăturaţi unui grup de lucru.
Modelul de securitate Access se bazează pe două elemente: fişierul cu informaţii despre grupul de lucru şi drepturile utilizatorilor şi grupurilor. Despre acestea vom vorbi mai pe larg în sesiunile următoare.
Access vă pune la dispoziţie două modalităţi de a implementa securitatea la nivel de utilizator prin intermediul interfeţei cu utilizatorul şi cu ajutorul ierarhiei DAO.

Grupuri de lucru

La baza securităţii de utilizator în Access stă conceptual de grup de lucru (workgroup). Acesta este definit ca un grup de utilizatori având acelaşi fişier cu informaţii (*Workgroup Information File*). Acest fişier cu informaţii despre un grup de lucru este o bază de date specială,criptată, a cărei denumire implicită este System.MDW. El conţine următoarele date: numele conturilor tuturor utilizatorilor şi grupurilor, ce utilizator aparţine cărui grup, parola fiecărui utilizator, un identificator unic (SID) pentru fiecare utilizator şi grup. Informaţiile despre drepturile şi grupurile asupra obiectivelor bazei de date sunt păstrate în baza de date respectivă, şi nu în fişierul cu informaţii despre grupul de lucru.

Conturi pentru grupuri şi utilizatori

Ideea de acordare sau revocare a drepturilor utilizatorilor Access se bazează pe existenţa conturilor pentru grupuri de utilizatori şi pentru utilizatori individuali. Astfel, drepturile pot fi acordate fie unui grup, fie doar unui utilizator.
Sistemul de securitate Access include câteva conturi predefnite care îl fac să pară inexistent până în momentul în care vă decideţi să-l folosiţi. Aceste conturi predefinite sunt cele ale utilizatorului Admin şi ale grupurilor Admins şi Users. Nici unul dintre acestea nu poate fi şters dintr-un grup de lucru.

Utilizatorul Admin

Orice nu grup de lucru conţine acest cont, a cărui parolă este iniţial vidă. Atâta timp cât această parolă devine vidă, la deschiderea sistemului Access sunteţi alocat în mod implicit ca utilizator Admin. Deşi nu puteţi şterge acest cont, îl puteţi exclude din grupul Admins, atâta timp cât acesta mai are cel puţin un utilizator. Dacă faceţi acest lucru, utilizatorul Admin nu va mai avea nici un fel de putere administrativă. Cu alte cuvinte, acesta este un fel de cont implicit atâta timp cât nu aţi implementat securitatea la nivel de utilizator.

Grupul Admins

Membrii grupului Admins au drepturi speciale, administrative, asupra tuturor obiectelor bazelor de date dintr-un grup de lucru. In plus, ei administrează toate conturile de utilizator şi de grup din grupul de lucru respectiv. In mod implicit, când creaţi un obiect, grupul Admins primeşte drepturi depline asupra acestuia. Este posibil, numai prin intermediul DAO, să revocaţi privilegiul Administer (administrare) al acestui grup pentru un obiect creat de dumneavoastră.

Grupul Users

Acesta este grupul implicit pentru toţi utilizatorii Access. Toate conturile predefinite şi conturile create prin intermediul interfeţei Access vor fi adăugate automat la acest grup, Access nu vă va lăsa să excludeţi utilizatori din acest grup decât ştergându-le contul cu totul. In mod implicit, grupul User primeşte drepturi depline asupra tuturor obiectelor nou create. Pentru a asigura securitatea unui grup de lucru, retrageţi toate drepturile mai importante ale acestui grup asupra obiectelor.

Crearea conturilor cu ajutorul interfeţei Access

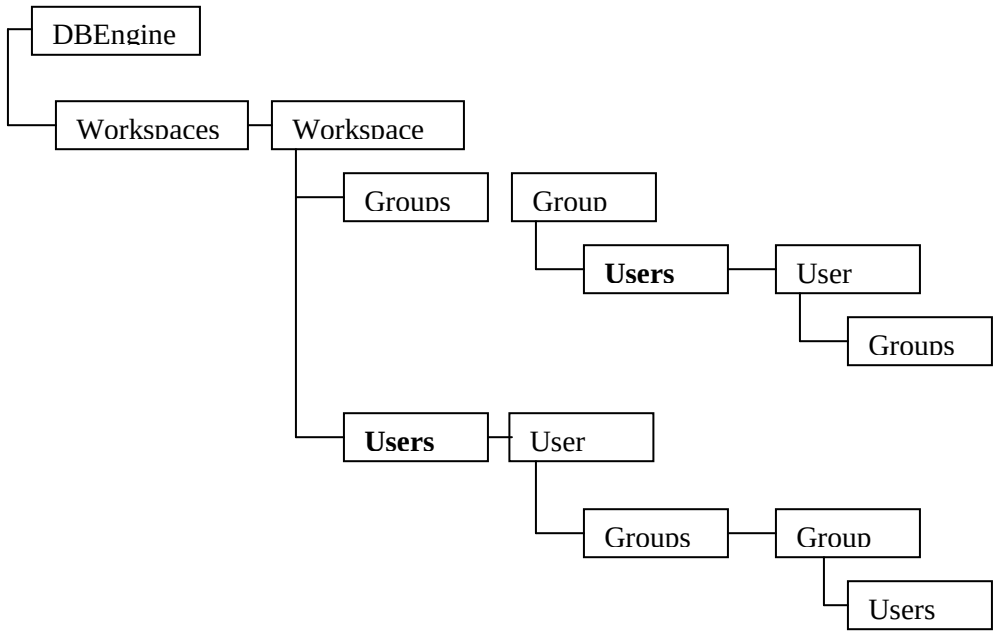
Pentru a crea şi a administra conturile utlizatorilor şi ale grupurilor de utilizatori cu ajutorul interfeţei Access folosim meniul *Tools |Security User and Group Accounts*.
Numai membrii grupului Admins pot adăuga şi şterge conturi sau modifica apartenenţa unui utilizator la un grup. Restul utilizatorilor pot vedea conturile şi-şi pot modifica propriile parole. Pentru a crea un nou cont propriu pentru un utilizator, apăsaţi butonul New din pagina Users a cutiei de dialog User and Group Accounts şi introduceţi numele contului şi un identificator personal unic în câmpurile cutiei de dialog New User/Group ce se va deschide. Pentru a crea contul unul nou grup de utilizatori, apăsaţi butonul New din pagina Groups a cutiei de dialog User and Group Accounts şi introduceţi numele contului şi

identificatorul unic pentru grupul respectiv. Pentru a şterge un utilizator sau un grup, selectaţi-l din lista Name şi apăsaţi butonul Delete din pagina User sau, respectiv, din pagina Groups a aceleiaşi cutii de dialog.

Notă: Atunci când creăm un nou cont pentru un utilizator sau pentru un grup, trebuie să avem grijă ca numele contului să fie diferit de cele existente deja.
După ce am creat un nou cont pentru un utilizator, parola acestuia este vidă. Pentru a activa sistemul de securitate la nivel de utilizator, trebuie să atribuim o parolă utilizatorului Admin (implicit, şi aceasta este vidă). Pentru aceasta, introducem în fila *Change Logon Password* (Schimbarea parolei) a ferestrei *User and Group Accounts* noua parolă pentru acest utilizator. După ce am făcut aceasta, la pornire, Access va afişa o cutie de dialog în care va trebui să introducem numele şi parola unui utilizator.
Pentru a schimba parola altui utilizator decât cel current, ca Admin, va trebui să închide, şi să porniţi din nou aplicaţia Access, să deschidem sesiunea cu numele lui şi apoi să-i modificaţi parola cu ajutorul paginii *Change Logon Password*. Pentru a putea face acest lucru, trebuie să-i cunoaşteţi vechea parolă; altfel, nu veţi reuşi să i-o modificaţi decât imediat după ce i-aţi creat contul, când parola e vidă. Cu alte cuvinte, numai utilizatorul respectiv va putea să-şi schimbe parola, ştiind-o pe cea veche.
Veţi vedea, la secţiunea următoare, că DAO vă dă posibilitatea de a schimba parola unui utilizator (dacă sunteţi alocat ca membru al grupului Admin) fără s-o ştiţi pe cea veche.
După ce aţi creat conturile pentru grupuri şi utilizatori, este timpul să specificaţi ce utilizatori aparţin căror grupuri. Pentru a adăuga un utilizator la un grup, selectaţi-l din lista Name a paginii User a cutiei de dialog User and Group Accounts şi apoi, alegeţi grupul respective din lista Avaible Group (Grupuri Existente). După aceea, apăsaţi butonul Add (adăugare). Veţi cedea că grupul va apărea în lista Member Of (Membru al). Pentru a anula apartenenţa unui utilizator la un grup, selectaţi utilizatorul din lista Name, alegeţi grupul din lista Member Of şi apăsaţi butonul Remove.

Administrarea conturilor prin DAO

Folosind ierarhia DAO puteţi crea, şterge din lista conturile utilizatorilor şi grupurile de utilizatori. In plus, puteţi lista grupurile cărora le aparţine un utilizator, precum şi membrii unui grup.
Pentru a înţelege mai bine relaţiile dintre utilizatori şi grupuri, să privim acea parte a ierarhiei DAO care ţine de securitate (colecţiile apar cu litere îngroşate iar obiectele cu litere cursive).



Din diagrama de mai sus, observăm că fiecare grup din colecţia Groups (grupuri) are o colecţie de utilizatori şi fiecare utilizator din colecţia Users (utilizatori) are o colecţie de grupuri (Groups). Aceasta se întâmplă deaorece între utilizatori şi grupuri există o relaţie de tip 1:m, adică un utilizator poate să aparţină mai multor grupuri, iar un grup poate avea mai multiutilizatori.

Colecţia Users şi obiecte de tip User

Colecţia Users a unui obiect de tip Workspace conţine toate conturile utilizatorilor acelei sesiuni de lucru. Colecţia Users a unui grup conţine conturile tuturor membrilor grupului respectiv.
Colecţia Users are o singură proprietate, Count (ce are ca valoare numărul elementelor colecţiei) şi trei metode, prezentate în tabelul următor.

Metoda	Descriere
Append	Adaugă un nou utilizator la colecţie
Delete	Sterge un utilizator din colecţie

Refresh	Reactualizează colecţia

Un obiect de tip User al colecţiei Users are trei proprietăţi şi două metode, prezentate în tabelele următoare:

Metoda	Descriere
Name	Numele utilizatorului (nu poate fi modificat dacă utilizatorul a fost deja adăugat la colecţia Ysers)
Password	Parola (case-sesitive)
PID	Identificatorul personal al contului (case-sessitive)

Metoda	Descriere
CreateGroup	Creează un nou obiect de tip Group
NewPassword	Inlocuieşte parola veche cu una nouă

Procedura următoare tipăreşte în fereastră numele tuturor conturilor utilizatorilor din cadrul colecţiei Users a obiectului Workspace implicit.

```
Sub UtilizatorWsp ()
Dim wsp As Workspace
Dim usr As User
Set wsp = DBEngine (0)
For Each usr In wsp.Users
      Debug.Print usr.Name
Next usr
End Sub
```

Dacă doriţi să tipăriţi numele conturilor utilizatorilor unui anumit grup, rulaţi procedura:

```
Sub UtilizatorGrp(strNumeGrup) As String)
Dim wsp As Workspace
Dim grp As Group
Dim usr As User
Set wsp = DBEngine (0)
Set grp = wsp.Groups("strNumeGrup")
For Each usr In grp.Users
      Debug.Print usr.Name
Next usr
End Sub
```

Colecţia Groups şi obiecte de tip Group

Ca şi în cazul colecţiei Users, ierarhia DAO conţine două tipuri de colecţii Groups. Colecţia Groups a unui obiect de tip Workspace conţine toate conturile de grup din acea sesiune de lucru. Colecţia Groups a unui utilizator conţine toate grupurile ce au ca membru utilizatorul respectiv. Colecţia Grups are o singură proprietate, Count (ce are ca valoare numărul elementelor colecţiei) şi trei metode, descrise şi în tabelul următor.

Metoda	Descriere
Append	Adaugă un nou utilizator la colecţie
Delete	Sterge un utilizator din colecţie
Refresh	Reactualizează colecţia

Un obiect de tip Group are două proprietăţi şi o singură metodă, CreateUser.

Proprietăţi şi metode	Descriere
Name	Numele utilizatorului (nu poate fi modificat dacă utilizatorul a fost deja adăugat la colecţia Groups)
PID	Identificatorul personal al contului (case-sensitive)
CreateUser	Creează un nou obiect de tip User.

Procedura următoare tipăreşte în fereastra *Immediate* numele tuturor conturilor de grup din cadrul obiectului Workspace implicit:

```
Sub GrupuriWsp()
```

```
Dim wsp As Workspace
Dim grp As Group
Set wsp = DBEngine (0)
For Each grp In wsp.Groups
       Debug.Print grp.Name
Next grp
End Sub
```

Pentru a crea contul unui grup, nu trebuie decât să apelăm metoda CreateGroup a unui obiect de tip Workspace:

```
Sub CreareGrup(strNume As String, strPID As String)
Dim wsp As Workspace
Dim grp As Group
Set wsp = DBEngine (0)
'Cream grupul
Set grp = wsp.CreateGroup(strNume,  strPID)
'Adaugam grupul la colectia Groups a obiectului workspace
Wsp.Groups.Append grp
End Sub
```

Pentru a crea contul unui utilizator şi a-l adăuga la grupul Users, vom apela metoda CreateUser a unui obiect de tip Workspace şi apoi metoda *Append* a colecţiei Groups a noului utilizator:

```
Sub CreateUtiliz(strNumeCont As String, strPID As String,_
strParola As String)
Dim wsp As Workspace
Dim usr As User
Set wsp = DBEngine (0)
'cream contul utilizatorului
Set usr = wsp.CreateUser(strNumeCont, strPID, strParola)
'il adaugam la colectia Users a obiectului workspace
Wrk.Users.Append usr
'il adaugam la grupul Users
Usr.Groups.Append wsp.Creategroup("Users")
End Sub
```

Funcţia următoare returnează valoarea True dacă utilizatorul al cărui cont are numele strNumeUt aparţine grupului cu numele strNumeGr iar altfel, returnează valoarea False:

```
Function EsteMembru (strNumeUt As String, strNumeGr As String) As
Boolean
Dim wsp As Workspace
Dim usr As User
Dim varGrup As Variant
Set wsp = DBEngine (0)
Wsp.Users.Refresh
Set usr = wsp.Users(strNumeUt)
Usr.Groups.Refresh
varGroup = usr.Groups(strNumeGr).Name
EsteMembru = Not IsNull(varGrup)
End Function
```

Notă: Deaorece nu ne-am propus să scriem şi codul necesar tratării erorilor, execuţia funcţiilor prezentate în acest capitol va eşua dacă daţi argumente invalide sau dacă aveţi privilegiile necesare efectuării anumitor operaţii.

Funcţia EsteMembru() reactualizează întâi colecţia Users a obiectului Workspace implicit şi colecţia Groups a utilizatorului al cărui cont se numeşte strNumeUt, apelând metoda Refresh pentru cele două colecţii. Altfel, modificările recente făcute prin intermediul interfeţei Access sau de către alt utilizator din cadrul grupului de lucru, nu ar fi luate în considerare. Apoi, pentru a afla dacă utilizatorul se află în grupul al cărui cont are numele strNumeGr, încercăm să accesăm proprietatea Name a grupului respectiv prin intermediul colecţiei Groups a utilizatorului. Dacă aceasta nu are valoarea Null, funcţia returnează valoarea True.
DAO dă şi posibilitatea de a schimba parola utilizatorului curent sau chiar şi a unui alt utilizator, dacă sunteţi un membru al grupului Admins. Procedura SchimbaParola primeşte ca argument numele utilizatorului, vechea parolă şi noua parolă.

```
Sub SchimbaParola(strUtiliz As String, strParVeche As String, strParNoua As String)
Dim wsp As Workspace
Dim usr As User
```

```
Set wsp = DBEngine (0)
Set usr = wsp.Users(str(Utiliz)
'schimba parola
Usr.NewPassword strParVeche, strParNoua
End Sub
```

Procedura SchimbaParola() apelează metoda NewPassword a obiectului de tip utilizator. Dacă sunteţi utilizatorul curent (adică modificaţi propria dumneavoastră parolă), trebuie să daţi ca prim argument NewPassword vechea parolă. Dacă sunteţi membru al grupului Admins, puteţi modifica parola oricărui utilizator fără a fi necesar să o ştiţi pe cea veche (primul argument al metodei NewPassword este ignorat).

Privilegiile (drepturile) utilizatorilor şi grupurilor de utilizatori

După ce aţi creat conturile utilizatorilor şi grupurilor, este timpul să definiţi privilegiile pe care ei le vor avea asupra obiectelor bazelor de te ale grupului de lucru din care aceştia fac parte. In acest scop puteţi folosi fie interfaţa Access cu utilizatorul, fie ierarhia DAO.
Fiecare tip de obiect dintr-o bază de date are un set de privilegii care pot fi acordate utilizatorilor. Aceste privilegii nu sunt aceleaşi pentru toate tipurile de obiecte. Tabelul următor arată ce privilegii poate să ofere fiecare tip de obiect.

Privilegiu	Descriere	Disponibil pentru
Open/Run	Deschidere şi/sau execuţie	formular,raport, macro, bază de date
Read Design	Vizualizarea obiectului în modul Design	tabelă,interogare,formular,raport,macro, modul
Modify Design	Modificarea designului	tabelă,interogare,formular,raport,macro,modul
Administer	Administrare	tabelă, interogare, formular,raport,macro, modul, bază de date
Read Data	Citirea datelor	tabelă, interogare
Update Data	Modificarea datelor	tabelă, interogare
Insert Data	Introducerea datelor	tabelă, interogare
Delete Data	Tergerea datelor	tabelă, interogare
Open Exclusive	Deschiderea exclusivă	bază de date

Numai baza de date însăşi şi obiectele de tip container pe care aceasta le conţine pot oferi privilegii. Controalele, coloanele, parametrii, barele de control nu au privilegii, ceea ce înseamnă că li se poate asigura securitatea individual.
In plus faţă de drepturile obiectelor existente, puteţi stabili şi drepturi asupra obiectelor noi. Dacă revocaţi drepturile unui utilizator asupra obiectelor noi nu înseamnă că el nu va mai putea să creeze obiecte noi. In plus, dacă un utilizator creează un obiect, el este proprietarul obiectului respectiv, şi altfel îşi poate atribui dreptul de a-l administra. Totuşi, DAO vă dă posibilitatea de a împiedica utilizatorii să creeze obiecte noi.
Utilizatorii Access pot avea două tipuri de drepturi: implicite şi explicite. Drepturile explicite sunt cele atribuite direct urilizatorului, iar cele implicite sunt cele ce-I revin ca urmare a apartenenţei sale la un anumit grup. Setul de drepturi ale unui utilizator reprezintă reuniunea drepturilor sale implicite şi explicite.
Nu oricine are dreptul de a stabili şi modifica privilegiile celorlalţi utilizatori asupra unui obiect. Membrii grupului Admins pot schimba privilegiile oricărui utilizator. In plus, proprietarul obiectului are drepturi deplice asupra acestuia, incluzând posibilitatea de a acorda alte utilizatori privilegii asupra obiectului respectiv. De asemenea, utilizatorii care au dreptul de a administra un obiect pot acorda sau revoca drepturi asupra acestuia.

Notă: Inainte de a şterge contul unui utilizator sau al unui grup, revocaţi toate drepturile acestuia. Altfel, dacă veţi crea un cont pentru un alt utilizator, având acelaşi nume şi PIB, el va beneficia de toate drepturile contului original.

Acordarea drepturilor cu ajutorul Interfeţei Access

Pentru a acorda drepturi folosind interfaţa Access, alegeţi meniul *Tools | Security |User and Group Permissions*.
Dacă dorim să acordăm privilegii unui utilizator, validăm butonul radio Users, iar pentru un grup, selectăm butonul Groups. Apoi, din lista *User/Group Name*, selectăm utilizatorul (utilizatorii) sau grupul (grupurile) căruia dorim să-I acordăm drepturile. Pentru a specifica obiectele asupra cărora vreţi să daţi privilegiile, selectaţi întâi din lista Object Type tipul obiectului respectiv. Această listă conţine toate tipurile de obiecte bază de date asupra cărora un utilizator poate avea drepturi. Apoi, alegeţi din lista Object Nume obiectele în cauză. Nu mai trebuie decât să validaţi casetele de validare corespunzătoare drepturilor pe care doriţi să le acordaţi.

Administrarea privilegiilor prin DAO

Atunci când lucraţi cu DAO, privilegiile reprezintă proprietăţi ale documentelor şi containerelor. Tabelul următor prezintă câteva dintre proprietăţile unui document sau container care ţin de privilegii, utilizatori şi proprietari.

Proprietate	Descriere
Inherit	Dacă are valoarea True, Jet va folosi drepturile pe care le stabiliţi şi la crearea unui nou container sau document.
Owner	Contul utilizatorului sau grupului care este proprietarul obiectului.
Permissions	Este un tip Long Integer şi stochează informaţii despre privilegiile explicite şi implicite asupra obiectului. Este read-only.
UserName	Contul utilizatorului sau grupului care se bucură de un set de privilegii.

Pentru a simplifica regăsirea şi stabilirea privilegiilor cu DAO, Microsoft a definit câte o constantă pentru fiecare tip de privilegiu.

Proprietate	Descriere
dbSecNoAccess	Obiectul nu poate fi accesat
dbSecFullAccess	Accesul la obiect este liber
dbSecDelete	Dreptul de a şterge obiectul
dbecReadSec	Dreptul de a citi informaţiile despre privilegiile asupra obiectului
dbSecWriteSec	Dreptul de a stabili privilegii asupra obiectului
dbSecWriteOwner	Dreptul de a schimba proprietarul obiectului
dbecCreate	Dreptul de a crea documente noi
dbSecReadDef	Dreptul de a regăsi definiţia unei tabele
dbSecWriteDef	Dreptul de a modifica definiţia unei tabele
dbSecRetrieveData	Dreptul de a regăsi informaţii prin intermediul obiectului
dbSecInsertData	Dreptul de a insera înregistrări
dbSecReplaceData	Dreptul de a modifica înregistrări
dbSecDeleteData	Dreptul de a şterge înregistrări
dbSecDBAdmin	Dreptul de a administra baza de date
dbSecDBCreate	Dreptul de a crea noi baze de date
dbSecDBExclusive	Dreptul de a deschide exclusiv o bază de date
dbSecDBopen	Dreptul de a deschide o bază de date
dbSecMacExecute	Dreptul de a executa o macrocomandă
dbSecMacReadDef	Dreptul de a regăsi definiţia unei macrocomenzi
dbSecMacWriteDef	Dreptul de a modifica definiţia unei macrocomenzi
dbSecFrmRptExecute	Dreptul de a deschide un formular sau un raport
dbSecFrmRptReadDef	Dreptul de a regăsi definiţia unui formular sau a unui raport şi de a citi modulul asociat
dbSecFrmRptWriteDef	Dreptul de a modifica definiţia unui formular sau a unui raport şi de a citi modulul asociat
dbSecModReadDef	Dreptul de a regăsi definiţia unui modul
dbSecModWrite Def	Dreptul de a modifica definiţia unui modul

Aflarea privilegiilor

Pentru a regăsi privilegiile unui utilizator asupra unui obiect, trebuie să verificaţi valoarea proprietăţii Permissions pentru a afla numai drepturile explicite, sau valoarea proprietăţii AllPermissions, pentru a afla reuniunea drepturilor explicite şi implicite. Mai exact, dacă folosind operatorul And pe biţi între valoarea uneia dintre aceste proprietăţi şi constanta corespunzătoare privilegiului căutat, rezultatul are ca valoare tot constanta, atunci înseamnă că utilizatorul are acel drept (specificat de constantă) asupra obiectului respectiv. Astfel:

```
bAreDreptul = ((doc.Permissions And dbConst) = dbConst)
```

dacă bAreDreptul are valoarea True, utilizatorul are dreptul specificat de constanta dbConst asupra obiectului doc.

Funcţia următoare returnează valoarea True dacă un utilizator are un anumit drept asupra unui obiect şi False, altfel. Argumentele funcţiei sunt următoarele:
1Priv – valoarea corespunzătoare dreptului
iTipOb – număr întreg ce reprezintă tipul obiectului (valorile posibile sunt: 0 pentru baza de date, 1 pentru formular, 2 pentru model, 4 pentru raport, 7 pentru tabelă)
VarNumeOb – şir de caractere ce conţine numele obiectului sau Null dacă vă interesează drepturile asupra containerului. Acest argument este opţional.

varNumeCont – şir de caractere ce conţine numele contului utilizatorului sau Null pentru utilizatorul curent. Acest argument este opţional.
bImp1 – are valoarea True dacă pentru contul unui utilizator doriţi să căutaţi privilegiul dorit şi printre drepturile implicite. Are valoarea False pentru contul unui grup sau dacă vă interesează numai drepturile explicite.

```
Function AreDreptul (1Priv As Long, iTipOb As Integer, bImp1 As_
Boolean, Optional varNumeOb As Variant, Optional varNumeCont As Variant) As Boolean
Dim db As Database
Dim cnt As Container
Dim doc As Document
Dim bPriv As Boolean
If IsMissing(varNumeCont) Then varNumeCont = CurrentUser()
Set db = CurrentDb()
'Regasim containerul
Set cnt = db.Containers(iTipOb)
'Reactualizam colectia Documents a containerului
Cnt.Documents.refresh
Cnt.UserName = vatCont
'Regasim documentul, daca numele lui a fost dat ca argument
If Not IsMissing(varNumeOb) Then
   Set doc = cnt.Documents(carNumeOb)
   doc.UserName = varNumeCont
   'Regasim privilegiul asupra obiectului
   If bImpl Then
   bPriv = ((doc.AllPermissions And 1Priv) = 1Priv)
Else
   bPriv = ((doc.Permissions And 1Priv) = 1Priv)
End If
Else
   'privilegiul se referea la container
If bImp1 Then
   bPriv = ((cnt.AllPermissions And 1Priv) = 1Priv)
Else
   bPriv = ((cnt.Permissions And 1 Priv) = 1Priv)
   End If
End If
Arereptul = bPriv
End Function
```

Criptarea bazei de date

Securitatea la nivel de utilizator va asigura baza dumneavoastră de date împotriva majorităţii utilizatorilor nedoriţi, dar nu împotriva tuturor acestora. Pentru un bun specialist în ale calculatoarelor nu va fi foarte greu să deschidă fişierul .MBD cu ajutorul unui editor şi să intre în baza de date. Un pas înainte pe drumul asigurării corespunzătoare a securităţii unei baze de date îl reprezintă criptarea acesteia. Numai proprietarul bazei de date sau unul dintre membrii grupului Admins o poate cripta sau decripta. Pentru a cripta o bază de date, alegem meniul *Tools*|*Security*|*Encode/Decode Database*. In fereastra ce se va deschide, alegem baza de date pe care dorim s-o criptăm sau s-o decriptăm şi apăsăm butonul **OK.**

3. Internetul şi bazele de date Access

Tipul de date Hyperlink

Tipul de date *Hyperlink* oferă posibilitatea de a introduce în câmpurile unei tabele, adresa URL. Datele de acest tip pot conţine trei informaţii:

1. *Textul ce va fi afişat* – într+un document HTML, am observat că anumite texte sunt afişate subiniat şi că, dacă faceţi click pe ele, se va deschide o altă pagină web. Aceste texte reprezintă hiperlegături ce conţin adresele URL ale documentelor pe care le deschid. Este mult mai sugestiv să afişaţi într-o pagină web o scurtă descriere decât o adresă URL. Această primă parte a unei date de tip hyperlink reprezintă deci descrierea paginii web pe care o deschide.
2. Adresa URL – cea de-a doua parte a unei date de tip hyperlink reprezintă chiar adresa URL a paginii web pe care o deschide.
3. Sub-adresa – reprezintă o secţiune în cadrul paginii web respective.

Cele trei părţi sunt separate prin caractere diez (#); astfel, o dată de tip hyperlink va arăta în felul următor:

Text#adresa#Subadresa

Tipul de date hyperlink nu se referă numai la adresele URL ale documentelor HTML, ci poate conţine locaţiile unor documente Word, Excel, formulare Access etc., oferindu-vă foarte multă libertate de mişcare. Astfel, puteţi folosi o hyperlegătură în locul unui buton de pe un formular, fără a trebui să scrieţi nici o linie de cod.

Utilizarea datelor de tip Hyperlink

Să presupunem că unii dintre profesori au pagini proprii web, care conţin informaţii despre activitatea didactică şi ştiinţifică a acestora. Pentru a putea accesa aceste pagini prin intermediul aplicaţiei noastre, adăugaţi la tabela *Profesor* un nou câmp, numit "Adresa Web", de tip Hyperlink.
Treceţi apoi în modul *Datasheet* şi introduceţi, de exemplu, următoarea adresă în câmpul Adresa Web corespunzător unuia dintre profesori:

Teora#http://www.teora.ro#

Observaţi faptul că o dată ce aţi introdus un URL într-un camp, cursorul mouse-ului va arăta ca o mână atunci când se va afla deasupra acestuia. Nu veţi putea modifica adresa URL făcând clic cu mouse-ul în câmpul respectiv. Pentru aceasta, va trebui să faceţi clic în câmpul precedent al aceleiaşi înregistrări şi să apăsaţi tasta Tab pentru a ajunge la câmpul "Adresa Web".
Modificaţi acum interogarea DetaliiProf astfel încât aceasta să returneze şi câmpul "Adresa Web", instrucţiunea SQL corespunzătoare va fi acum următoarea:

```
SELECT Profesor.Nume, Profesor.Catedra, Titlu.Titlu,
Titlu.Salariu, Profesor.Statut, Profesor.IdTitlu,
Profesor.IdProf, Profesor.AdresaWeb
FROM Titlu RIGHT JOIN Profesor ON Titlu.IdTitlu +
Profesor.IdTitlu;
```

Adăugaţi apoi la formularul DetaliiProfesor o nouă casetă de text ataşată acestui camp.

Acum, nu mai trebuie decât să treceţi în modul Form View, să navigaţi până la înregistrarea pentru care aţi introdus valoarea în câmpul Adresa Web şi să faceţi clic pe această valoare. Dacă sunteţi conectaţi la Internet, se va lansa browserul dumneavoastră implicit care va deschide pagina web corespunzătoare.
Hiperlegăturile pot fi folosite şi fără să fie stocate într-o tabelă. Puteţi atribui o hiperlegătură unui buton de comandă, unui control de tip Image (imagine) sau unei etichete. Aceste controale au două proprietăţi noi în Access 97, HyperlinkAddress şi HyperlinkSubAddress, ce vă permit să specificaţi o adresă şi, respectiv, o subadresă. Adresa poate fi un URL, ori calea completă şi numele unui fişier .html .html local. In momentul în care utilizatorul va face clic pee le, ea va porni browserul implicit care va deschide pagina web specificată de cele două proprietăţi.

VBA şi lucrul cu obiectele de tipul Hyperlink

Acces dă posibilitatea de a lucra cu date de tip Hyperlink şi prin intermediul codului VBA. Un control ce conţine date de tip Hyperlink (buton de comandă, imagine sau etichetă) are anumite proprietăţi şi metode care vă să efectuaţi câteva operaţii cu datele de acest tip.

Proprietate	Descriere
Hyperlink	Accesează obiectul de tip Hyperlink din controlul respective. Poate fi folosită numai în codul VBA.
HyperlinkAddress	Specifică adresa principală a unui obiect, document sau pagină web.
HyperlinkSubAddress	Specifică o locaţie în cadrul obiectului specificat de proprietatea HyperlinkAddress. Poate reprezenta un obiect dintr-o bază de date Access database, un semn de carte dintr-un document Word, un domeniu de cadrul unui document HTML.

Metodă	Descriere
Follow	Deschide documentul sau pagina web specificată de proprietatea HyperlinkAddress
AddToFavorites	Adaugă adresa specificată de proprietatea HyperlinkAddress la directorul Favorites.

In plus, puteţi lucra şi cu date de tip Hyperlink ce nu sunt incluse într-un control de pe un formular. Obiectul aplicaţie Access are atât metoda AddFavorites cât şi metoda FollowHyperlink, ce deschide documentul sau pagina web specificată de adresa ce i-o daţi ca argument. Mai mult, VBA vă pune la dispoziţie şi o funcţie foarte utilă *HyperlinkPart*, care primeşte ca argumente o

hiperlegătură şi o constantă. In funcţie de valoarea constantei, ea returnează una dintre cele trei părţi ce pot compunehyperlegătura: textul afişat (*acDisplayText*), adresa (*acAddress)* şi sub-adresa (*acSubAddress)*.
Dacă aţi testa formularul DetaliiProf după ce aţi adăugat câmpul Adresa Web, aţi observat că pentru a modifica valoarea acestuia, trebuie să navigaţi până la el cu ajutorul tastei Tab. Altfel, dacă faceţi clic pe el, se va deschide pagina web corespunzătoare. Pentru a ilustra lucrul cu obiectele de tip Hyperlink, ne-am gândit că creăm un mod mai simplu de a edita valoarea acestui câmp: cu ajutorul unui alt formular, în care utilizatorul să poată introduce cele trei părţi componente ale unei date de tip Hyperlink.

Adăugaţi, aşadar, pe formularul DetaliiProf, în dreapta câmpului Adresa Web, un buton.

Daţi proprietăţii Name a acestuia valoarea cmdEditAdr, iar proprietăţii Caption, daţi ca valoare trei puncte de suspensie (…). Atunci când utilizatorul va apăsa pe acest buton, se va deschide formularul EditareAdresa (pe care urmează să-l creaţi). Pentru a programa acest mecanism, putem folosi fie procedura de tratare a evenimentului Click al butonului cmdEditAdr, fie proprietatea HyperlinkSubAddress a acestuia. De această dată, vom alege cea de-a doua cale, aşa că introduceţi în câmpul proprietăţii HyperlinkSubAddress a butonului cmdEditAdr următoarea valoare:

```
FormEditareAdresa
```

Am folosit proprietatea HyperlinkSubAddress deoarece formularul EditareAdresa va fi un obiect în cadrul bazei de date curente.
Creaţi formularul EditareAdresa, cu cele trei casete de text ale sale, ale căror proprietăţi Name să fie txtTextAfisat, txtAdr şi, respectiv txtSubAdr şi cu butonul de comandă OK (Name – cmdOK).
La deschidere, câmpurile formularului EditareAdresa vor conţine cele trei elemente componente ale valorii curente a câmpului Adresa Web al formularului DetaliiProf. Pentru aceasta, scrieţi următoarea procedură de tratare pentru evenimentul Load al formularului EditareAdresa:

```
Private Sub Form_Load ()
Dim strHyp As Variant
strHyp = Forms!DetaliiProf("Adresa Web")
txtTextAfisat = HyperlinkPart(strHyp, acdisplayText)
txtAdr = HyperlinkPart(strHyp, acAddress)
txtSubAdr = HyperlinkPart(strHyp, acSubAddress)
End Sub
```

La închiderea formularului, câmpul Adresa Web al formularului DetaliiProf va conţine hiperlegătura formată din datele introduse de utilizator. Scriem următoarea procedură de tratare a evenimentului Click al butonului cmdOK:

```
Private Sub  cmdOK Click()
Forms!DetaliiProf("Adresa Web") = txtTextAfisat & "#" & txtAdr & "#" & txtSubAdr
DoCmd.Close
End Sub
```