

Indrumar Baze de Date - Capitolul 1

NOTIUNI INTRODUCTIVE PRIVIND SISTEMELE DE GESTIUNE A BAZELOR DE DATE

O *baza de date (database)* este o colectie de date creata si mentinuta computerizat, care permite operatii de inserare, actualizare, stergere si interogare a datelor. Utilizatorii unei baze de date pot accesa datele memorate prin intermediul unui program numit Sistem de Gestiune a Bazei de Date - SGBD - (*Database Management System - DBMS*), care receptioneaza cererile utilizatorilor (pentru operatii de introducere, stergere, modificare sau interogare), le interpreteaza, executa operatiile corespunzatoare si returneaza rezultatul catre utilizatori. Fiecare sistem de gestiune administreaza datele conform unui anumit model de date. Exista mai multe modele de date utilizate în sistemele SGBD: modelul ierarhic, modelul retea, modelul relational, modelul obiect-orientat, modelul obiect-relational. Dintre acestea, în momentul de fata, modelul relational este cel mai larg raspandit, în special în aplicatiile comerciale si acesta va fi studiat în continuare în aceasta lucrare .

1.1 MODELUL RELATIONAL

În modelul relational o baza de date este compusa dintr-o multime finita de relatii, fiecare relatie reprezentând un tip de entitate sau o asociere dintre doua sau mai multe tipuri (multimi) de entitati. Din aceasta definitie rezulta ca într-o baza de date fiecare relatie este unica (nu exista doua sau mai multe relatii de acelasi fel), dat fiind ca o baza de date este o multime de relatii.

1.1.1 RELATII

O relatie se defineste prin intermediul atributelor sale. *Atributele* unei relatii sunt attributele tipului de entitate sau de asociere pe care îl reprezinta relatia respectiva. Fiecare atribut al unei relatii are un domeniu de definitie si poate lua o singura valoare (din domeniul sau de definitie) pentru fiecare tuplu, adica attributele au numai valori scalare.

Un domeniu de definitie (*domain*) este o multime cu nume de valori atomice de acelasi tip, având o anumita semnificatie, din care își iau valori attributele relatiilor. Denumirea de *valori atomice* înseamna ca aceste valori nu pot fi descompuse din punct de vedere al sistemului de gestiune al bazei de date si reprezinta cea mai mica entitate semantica de date.

Schema relatiei (relation schema), notata $R(A_1, A_2, \dots, A_i, \dots, A_n)$, este compusa din numele relatiei (R) si din lista ordonata a atributelor sale $A_1, A_2, \dots, A_i, \dots, A_n$, fiecare atribut A_i definit pe domeniul sau de definitie, $D(A_i)$.

Schema relatiei este folosita pentru a descrie relatia respectiva si se mai numeste si tipul sau intensiunea relatiei. Numarul de attribute ale schemei unei relatii se numeste gradul relatiei.

O *relatie (relation)* R definita de schema $R(A_1, A_2, \dots, A_i, \dots, A_n)$ este o multime de n -tupluri t , fiecare tuplu fiind o lista ordonata de n valori $t = \langle v_1, v_2, \dots, v_i, \dots, v_n \rangle$, unde $1 \leq i \leq n$ si v_i este valoarea atributului A_i , aparținând domeniului sau de definitie $D(A_i)$.

Din aceasta definitie rezulta imediat ca într-o relatie nu exista tupluri duplicat (doua sau mai multe tupluri identice), relatia fiind o multime (în sens matematic) de tupluri. Numarul de tupluri al unei relatii se numeste *cardinalitatea* relatiei.

O relatie se reprezinta printr-un *tabel (table)* care este compus din urmatoarele parti:

- Numele tabelului, care este identic cu numele relatiei pe care o reprezinta.
- Un numar de coloane egal cu numarul de attribute ale relatiei, fiecare coloana reprezentând un atribut.
- Capul tabelului, în care se înscriu numele atributelor relatiei, fiecare atribut fiind înscris în coloana corespunzatoare.

- O multime de linii, fiecare linie corespunzând unui tuplu (deci unei entitati); în fiecare element al unei linii se înregistrează valoarea atributului corespunzător coloanei în care se afla elementul respectiv.

Deși în numeroase documentații se folosește termenul de *tabel* pentru a desemna *relatia* pe care o reprezintă, cele două noțiuni nu sunt identice: *relatia* este o noțiune abstractă (o multime în sens matematic), în timp ce *tabelul* este o reprezentare a relației.

Sistemele de baze de date relationale utilizează ca limbaj de programare limbajul SQL (*Structured Query Language*), pentru care au fost propuse mai multe standarde de către Organizația Internațională de Standardizare (*International Standardization Office - ISO*). Majoritatea sistemelor de gestiune a bazelor de date relationale actuale implementează versiunea din anul 1992 a standardului pentru limbajul SQL, denumită SQL92 (sau SQL2).

1.1.2 CONSTRÂNGERI DE INTEGRITATE

Relațiile unei baze de date reflectă realitatea modelată și de aceea valorile pe care le conțin trebuie să respecte anumite reguli, care să corespundă celor din realitate.

Constrângerile de integritate (integrity constraints) sunt reguli care se definesc la proiectarea unei baze de date și care trebuie să fie respectate de orice stare a acesteia.

Din punct de vedere al locului unde sunt definite, constrângerile pot fi constrângeri intra-relație și constrângeri inter-relații. *Constrângerile intra-relație* sunt reguli care se impun în cadrul unei singure relații și asigură integritatea datelor acesteia. Ele sunt, la rândul lor, de trei categorii: *constrângeri de domeniu*, *constrângeri de tuplu* și *constrângeri impuse prin dependente de date* (dependente functionale, multivalorice sau de joncțiune).

Constrângerile inter-relații sunt reguli care se impun între două sau mai multe relații. Cele mai importante constrângeri inter-relații sunt constrângerile de *integritatea referențială*, care se realizează prin intermediul *cheilor straine* și asigură asocierea corectă a relațiilor.

Din punct de vedere al modului de definire, constrângerile unei baze de date pot fi inerente, implicite și explicite.

Constrângerile inerente sunt cele ale modelului de date însuși, care nu trebuie să fie specificate la definirea relațiilor, dar sunt respectate prin modul în care se construiesc relațiile.

Constrângerile implicite sunt cele reprezentate în mod implicit în schemele relațiilor prin intermediul instrucțiunilor de definire a datelor. Pentru fiecare model de date există un set de constrângeri implicite care se definesc odată cu definirea schemelor de date ale acestuia. Pentru modelul relational, constrângerile de domeniu, constrângerile de tuplu și constrângerile de integritate referențială sunt exemple de constrângeri implicite. Constrângerile implicite sunt memorate în baza de date și sistemul de gestiune impune automat respectarea acestora.

Constrângerile explicite sunt constrângeri suplimentare pe care trebuie să le respecte relațiile unei baze de date și care nu sunt impuse automat de sistemul SGBD, ci prin proceduri speciale. Ca exemple de constrângeri explicite sunt unele dependente de date (care vor fi studiate în Capitolul 5).

Constrângerile de domeniu sunt condiții impuse valorilor atributelor, astfel încât acestea să corespundă semnificației pe care o au în realitatea modelată. Dat fiind că, în reprezentarea unei relații printr-un tabel, valorile atributelor sunt reprezentate pe coloane, constrângerile de domeniu se mai numesc și constrângeri de coloană. Dintre constrângerile de domeniu, constrângerea NOT NULL și constrângerea de valoare implicită (DEFAULT) sunt constrângeri cu caracter mai general, care se pot aplica oricărui atribut; *constrângerea de verificare* (CHECK) se poate aplica unor anumite atribute, în funcție de semnificația acestora.

Constrângerile de tuplu: cheia primară și chei secundare. O relație este definită ca o *multime* de tupluri, deci tuplurile unei relații trebuie să fie distincte. Aceasta înseamnă că într-o relație nu pot exista două (sau mai multe) tupluri care să conțină aceeași combinație de valori ale tuturor atributelor. De obicei, într-o schemă de relație există o submultime de atribute SK cu proprietatea că, în orice stare s-ar afla relația, nu există două tupluri distincte ale relației care să aibă aceeași combinație de valori ale atributelor submultimii respective.

O *supercheie* (*superkey*) a unei relatii este o submultime (SK) de attribute ale relatiei care prezinta proprietatea de unicitate, adica orice combinatie de valori ale atributelor supercheii este unica pentru orice stare a relatiei.

Acest lucru înseamna ca, daca se cunoaste combinatia de valori ale atributelor supercheii (valoarea supercheii), atunci acel tuplu poate fi identificat în mod unic. Orice relatie are cel puțin o supercheie: multimea tuturor atributelor sale. Un concept mai util în dezvoltarea bazelor de date îl reprezinta conceptul de cheie candidata (sau, mai simplu, cheie).

O *cheie candidata* (*candidate key*) este o supercheie ireductibila (minimala).

Conform definitiei de mai sus, o cheie candidata CK trebuie sa prezinte proprietatea de unicitate (nu exista doua tupluri diferite ale relatiei care sa contina aceeasi combinatie de valori ale atributelor cheii CK) si proprietatea de ireductibilitate (nu exista nici o submultime proprie, nevida a cheii CK care sa aiba proprietatea de unicitate).

O cheie candidata poate sa fie *simpla* (alcatuita dintr-un singur atribut), sau *compusa* (alcatuita din mai multe attribute).

Atunci când exista mai multe chei candidate, una dintre ele se alege ca si *cheie primara*, celelalte chei candidate fiind numite chei secundare, alternative sau unice.

O *cheie primara* (*primary key*) este o cheie candidata careia proiectantul îi confera un rol special de accesare si identificare a tuplurilor relatiei. În plus, se impune ca atributelor cheii primare sa nu admita valori de NULL sa nu fie modificate prin operatii de actualizare a datelor.

O *cheie secundara* (*alternativa, unica*) (*secondary, alternate, unique key*) este o cheie candidata care nu a fost desemnata de proiectant ca si cheie primara.

Cheile secundare admit valori NULL pentru unele din attributele lor daca se respecta conditia de unicitate a valorilor.

O cheie primara compusa din attributele existente ale tipului de entitate se numeste *cheie naturala*. În general, cheile naturale sunt compuse din mai multe attribute (ceea ce produce scaderea eficientei operatiilor relationale) si de cele mai multe ori se folosesc *chei artificiale*. O cheie primara artificiala este un atribut care se adauga în schema relatiei pentru identificarea unica a tuplurilor. De exemplu, în relatia ANGAJATI se adauga atributul IdAngajat, ca numar de identificare al fiecarui angajat al institutiei:

ANGAJATI (IdAngajat , Nume , Prenume , DataNasterii , Adresa , Salariu)

Acest atribut este o cheie artificiala a relatiei si poate identifica în mod unic un tuplu, deoarece (prin conventie) nu se atribuie acelasi numar de identificare la mai mult de un angajat.

Constrângeri între relatii: cheia straina. O *cheie straina* (*foreign key*) este o multime de attribute FK ale unei relatii R_1 care refera relatia R_2 si satisface urmatoarele conditii: (a) attributele cheii straine FK sunt definite pe domenii compatibile cu cele ale atributelor unei cheii candidate CK a relatiei R_2 si (b) combinatia de valori ale atributelor FK într-un tuplu din relatia R_1 , fie este identica cu combinatia de valori ale atributelor CK a unui tuplu oarecare din starea curenta a relatiei R_2 , fie ia valoarea NULL. Cheia straina realizeaza asocierea N:1 între relatiile R_1 si R_2 (ceea ce este echivalent cu asocierea 1:N între relatiile R_2 si R_1) si reprezinta o constrângere între doua relatii, numita *constrângere referentiala*. Relatia care contine cheia straina se numeste *relatia care refera* (R_1 în definitia de mai sus), iar relatia care contine cheia candidata se numeste *relatia referita* (R_2 în definitia de mai sus).

Integritatea referentiala (*referential integrity*) este proprietatea bazei de date care garanteaza ca oricare valoare a unei chei straine se regaseste printre valorile cheii candidate corespunzatoare din relatia referita, sau cheia straina are valoarea NULL (daca attributele acesteia nu sunt supuse constrângerii NOT NULL).

Majoritatea sistemelor SGBD implementeaza constrângerea de mentinere a integritatii referentiala în mod implicit, refuzând modificari ale datelor (introducere, stergere, actualizare) care ar putea viola integritatea referentiala. Daca sistemul nu asigura aceasta functionalitate, ea trebuie implementata în programele de aplicatii.

Un index al unei relatii (index) este o structura auxiliara memorata în baza de date care permite accesul rapid la înregistrările (tuplurile) relatiilor prin ordonarea acestora.

La definirea unei relatii se stabilesc doua categorii de indecsi: indexul primar al relatiei, care determina localizarea tuplurilor în fisierele bazei de date, si zero, unul sau mai multe indexuri secundare, care nu modifica localizarea tuplurilor, dar sunt folosite pentru ordonarea si regasirea tuplurilor dupa un criteriu dat. *Indexul primar (primary index)* se defineste pe unul sau mai multe attribute ale relatiei si reprezinta cheia (eticheta) dupa care ordoneaza tuplurile relatiei, folosind structuri de ordonare (arbori, tabele de dispersie, etc.). Indexul primar determina adresa de memorare a înregistrarilor (tuplurilor) în fisierele bazei de date. *Un index secundar* pe un atribut A al unei relatii (*secondary index*) este o structura care contine o multime de perechi (v, L) ordonate dupa v ; fiecare pereche corespunde unui tuplu al relatiei, v este valoarea atributului A , iar L este adresa tuplului respectiv în structura indexului primar al relatiei.

Fiecare SGBD prevede anumite modalitati de reprezentare si de creare a indecsilor si de aceea aceasta parte a proiectarii unei baze de date depinde de sistemul folosit.

1.1.3 SECURITATEA SI PROTECTIA BAZELOR DE DATE

O descriere completa a problemelor de securitate în cadrul sistemelor de calcul si a bazelor de date este în afara scopului acestui îndrumar, dar exista numeroase alte carti dedicate acestor probleme, care pot fi consultate de cititorii interesati. Însa, exista câteva aspecte fundamentale privind protectia si securitatea bazelor de date care trebuie sa fie cunoscute de orice proiectant sau programator de baze de date si acestea vor fi prezentate pe scurt în continuare.

Cu problemele de protectie si securitate este responsabil administratorul bazei de date, care are un cont privilegiat în sistemul de gestiune (numit în general cont de sistem - *system account*) care prevede capabilitati foarte puternice, pe care alte conturi sau utilizatori nu le au. Prin intermediul contului de sistem administratorul bazei de date poate efectua mai multe operatii: crearea conturilor, acordarea sau retragere privilegiilor, etc.

Orice persoana care doreste sa se conecteze (*log in*) la o baza de date trebuie sa detina un cont (*account, user*) si o parola (*password*). Sistemul de gestiune verifica contul si parola si *autentifica* acel utilizator, daca acestea sunt corecte. Programele de aplicatii sunt considerate de asemenea utilizatori si se conecteaza pe un anumit cont si trebuie sa furnizeze parola acestuia.

Conectarea unui utilizator al unei baze de date (printr-un cont si o parola) nu este suficienta pentru ca acel utilizator sa beneficieze de toate functionalitatile oferite de SGBD, ci el poate beneficia numai de acelea pentru care a primit drepturi (autorizari). În cele mai multe sisteme SGBD exista doua niveluri de autorizare: nivelul contului si nivelul relatiilor si fiecarei relatii i se asigneaza un cont proprietar (*owner account*), care este, în general, contul care a fost folosit atunci când a fost creata relatia respectiva.

În limbajul SQL2 (care va fi prezentat în capitolul urmator) sunt prevazute instructiuni de acordare (*GRANT*) si de revocare (*REVOKE*) a drepturilor de acces ale utilizatorilor (conturi) la diferite obiecte ale bazei de date. Câteva exemple de astfel de instructiuni vor fi date în sectiunile urmatoare. În plus, în sistemele SGBD performante, pot exista numeroase alte posibilitati de organizare a drepturilor utilizatorilor (grupuri, roluri, etc.). Controlul acestor drepturi face parte din activitatea de administrare a bazei de date, care reprezinta în sine un domeniu foarte vast, în care activeaza numerosi specialisti (administratori de baze de date).

1.2 INSTALAREA SI PORNIREA UNOR SISTEME DE GESTIUNE A BAZELOR DE DATE

Dintre sistemele de gestiune a bazelor de date relationale existente la ora actuala vom studia: Microsoft SQL Server, Microsoft Access, Oracle, MySQL.

1.2.1 SISTEMUL DE GESTIUNE SQL SERVER

O versiune de test a sistemului SQL Server se poate obtine de la firma Microsoft (www.microsoft.com) si poate fi instalata sub sistemele de operare Windows (NT/2000/XP) si utilizata pentru a testa diferitele aspecte de lucru cu bazele de date relationale.

Un server SQL gestioneaza mai multe baze de date, care pot fi accesate partajat de mai multi utilizatori aflati în retea.

Instalarea serverului SQL Server 2000 (*Server and Client Tools*) se face pe o singura statie din retea, iar pe celelalte statii se instaleaza programele client. La instalare trebuie sa fie selectat modul de autentificare *Mixed* (Windows and SQL Server).

Trebuie retinut faptul ca dupa instalarea serverului SQL Server 2000 trebuie verificata vulnerabilitatea la virusul SQL Worm (folosind, de exemplu, programul *FixSQLex* de la *Symantec*). Eliminarea acestei vulnerabilitati (daca exista) se face fie instalând SQL Server Service Pack 3, fie instalând un pachet de actualizare de la Microsoft (*SQL_MSDE_CriticalUpdate_ENU.msi*). Daca la o noua rulare a programului *FixSQLex* înca se mai raporteaza vulnerabilitate, aceasta este cel mai probabil provocata de faptul ca biblioteca vulnerabila (*ssnetlib.dll*) a fost salvata (în directorul.. \SQL Server\80\Tools\Binn\backup\data_corectiei\)) înainte de a fi înlocuita cu o versiune sigura în cursul instalarii pachetelor de actualizare. Se poate sterge acest fisier si se testeaza din nou vulnerabilitatea.

Dupa instalarea serverului SQL Server, utilizatorul are la dispozitie mai multe facilitati de creare, administrare si lucru cu bazele de date, precum si câteva exemple de baze de date preinstalate (*pubs*, *Nortwind*), care pot fi folosite pentru studiu.

Pornirea si oprirea serverului se face folosind programul *Service Manager* (selectând *Start -> Programs -> Microsoft SQL Server -> Service Manager*), care permite selectarea unui server din retea (daca sunt mai multe servere instalate), oprirea serverului, introducerea unei pauze de functionare, sau continuarea functionarii dupa o pauza.

Exista mai multe programe utilitare (instrumente) care permit administrarea sistemului si executarea diferitelor operatii: crearea unor noi baze de date, crearea tabelor si a altor obiecte (triggere, indecsi, vederi), crearea utilizatorilor, salvarea si refacerea bazelor de date, etc. Dintre aceste programe, unele pot fi executate la nivel de linie de comanda (*osql*, *isql*), iar altele prezinta interfata grafica si sunt mai usor de folosit (*SQL Server Enterprise Manager*, *SQL Query Analyzer*).

1.2.1.1 Utilizarea programului *Enterprise Manager*

Programul utilitar *Enterprise Manager* se lanseaza selectând *Start -> Programs -> Microsoft SQL Server -> Enterprise Manager*. Acest program este foarte puternic si permite atât administrarea sistemului (crearea utilizatorilor, refacerea bazelor de date, etc.) cât si proiectarea bazelor de date (crearea bazelor de date si a tabelor, asocierea între tabele, etc.). În continuare vor fi prezentate numai notiunile strict necesare pentru a începe lucrul cu SQL Server, restul informatiilor sunt destul de usor de gasit în manualul *Books Online* al sistemului SQL Server.

Lansarea programului *Enterprise Manager* se poate face atât de pe statia pe care a fost instalat serverul, cât si de pe orice statie pe care s-a instalat un client.

Pentru statia serverului se poate înregistra serverul local (al carui nume este, de regula, numele calculatorului pe care este instalat, de exemplu *Server125a*, în laborator) pentru contul (*login*) de administrare (*sa*), creat implicit la instalare. Acest cont are toate drepturile, atât la nivelul sistemului (administrare, securitate, etc) cât si la nivelul fiecărei baze de date. De regula, noii utilizatori se creeaza din contul *sa* si acestora li se acorda mai putine drepturi, atâtea câte sunt necesare pentru ca sa realizeze sarcinile dorite, fara sa afecteze administrarea sistemului de gestiune, care trebuie sa ramâna numai în atributiile administratorului (contul *sa*).

În panoul din stânga al ferestrei *Enterprise Manager*, elementul *Console Root* reprezinta radacina unui arbore care contine în primul nivel toate instalările SQL Server disponibile în retea respectiva, organizate în mai multe grupuri de servere. Initial în grupul implicit *SQL Server Group* nu exista nici un server înregistrat si înregistrarea se face cu comanda *New SQL Server Registration* selectata din meniul de context care se deschide la apasarea butonului dreapta al mouse-ului atunci când este selectat directorul *SQL Server Group*.

Dupa înregistrare, se poate rula *Enterprise Manager* cu conectare pe contul *sa* în mod de autentificare Windows. În acest mod este bine sa fie stabilita (sau modificata) parola contului *sa* si sa fie editata înregistrarea serverului cu comanda *Edit SQL Server Registratuion Properties* din meniul de context care se deschide la apasarea butonului dreapta al mouse-ului atunci când este selectat numle instantei serverului (în laborator, *Server125a*). În fereastra care se deschide (cu titlul *Registered SQL Server Properies*) se selecteaza optiunile *Use SQL Server Authentication* si *Always propmt for login name and password*.

Din contul *sa* se creeaza conturile de conectare *login* cu comanda *New Login* din meniul de context care se deschide la apasarea butonului dreapta al mouse-ului aunci când este selectat

subdirectorul *Logins* din directorul *Security*. În laborator aceste conturi sunt denumite *user1*, *user2*, ..etc. Fiecare cont utilizator are setat rolul de sistem *Database Creators* (care se seteaza în pagina *Server Roles* a ferestrei care se deschide la crearea unui nou cont) si dreptul *public* asupra bazei de date *Northwind* (care se seteaza în pagina *Database Access*).

Pe orice statie pe care este instalat un client SQL, se poate lansa programul *Enterprise Manager*, si prima operatie care trebuie sa fie facuta este înregistrarea serverul dorit din retea. În cursul înregistrării (cu comanda *New SQL Server Registration* din meniul de context) se aleg optiunile *SQL Server Authentication* si *Always prompt for user name and password*. Dupa aceasta se poate face conectarea pe contul propriu (*login*) si parola acestuia, cont care trebuie sa fi fost creat mai înainte pe server.

Fereastra principala a programului *SQL Server Enterprise Manager* contine un meniu, o bara de instrumente si doua panouri. În panoul din stânga se afiseaza numele serverului, împreuna cu sapte subdirectoare ale acestuia, iar în panoul din dreapta al ferestrei sunt prezentate informatii privind directorul selectat. Dintre directoarele afisate, cele mai utile sunt directorul *Databases*, care contine bazele de date gestionate de serverul respectiv, si directorul *Security*, care permite administrarea utilizatorilor si a drepturilor de acces la bazele de date.

În directorul *Databases* se afla mai multe directoare care contin bazele de date ale sistemului (care reprezinta catalogul sistemului de gestiune: *Master*, *Model*, *Msdb*), bazele de date preinstalate ca exemple (*Northwind*, *pubs*) si toate bazele de date create de utilizatori.

În fiecare baza de date sunt memorate mai multe categorii de obiecte: diagrame referentiale (*Diagrams*), tabele (*Tables*), vederi (*Views*), proceduri stocate (*Stored Procedures*), utilizatorii care au acces la baza de date reaspectiva (*Users*), roluri (grupari de privilegii) (*Roles*) si altele.

Din fiecare categorie de obiecte se poate selecta unul dintre acestea pentru a fi accesat, editat sau executat (în functie de obiect si de drepturile utilizatorului) si se pot crea obiecte noi cu comanda *New*, care se poate actiona din bara de instrumente sau dintr-un meniu de context (care se obtine prin apasarea butonului dreapta al mouse-ului atunci când este selectat un obiect din acea categorie).

În directorul *Security* se pot inspecta si modifica conturile de conectare (*logins*) si rolurile definite pentru serverul SQL la care s-a realizat conexiunea.

Lista tuturor conturilor de conectare existente (*logins*) se afiseaza în panoul din dreapta atunci când se selecteaza optiunea *Logins* în arborele de selectie si pentru fiecare cont se poate selecta o optiune din cele existente în meniul contextual. La optiunea *Properties* se pot inspecta si modifica diferite proprietati ale contului respectiv (parola, baza de date implicita la deschidere, rolul contului în sistemul SQL server,etc.). Fiecarui cont de conectare i se pot atribui unul sau mai multe roluri în sistemul de gestiune: *System Administrators*, care permite orice operatii în sistemul SQL Server, *Security Administrators*, care permite administrarea utilizatorilor, *Database Creators*, care permite crearea de noi baze de date etc.

Pentru ca un cont de conectare sa poata accesa o baza de date, trebuie sa se i se acorde drepturi asupra acelei baze de date (*public*, *db_owner*, etc.). Atunci când se acorda drepturile, se creaza un utilizator (*user*) al acelei baze de date corespunzator contului respectiv si acest utilizator are drepturile setate.

Contul de conectare *sa* (*System Administrator*) are o comportare speciala. El este creat automat la instalarea serverului SQL cu rolul de administrator de sistem si primeste drepturi de acces la orice baza de date existenta sau nou creata, atribuindu-i-se rolul *public* si rolul de proprietar (*dbo_owner*) al bazei de date.

Pentru fiecare baza de date sistemul creeaza automat un tip de utilizator special (*dbo*) care are drepturi de proprietate asupra bazei de date respective. Acest utilizator este atribuit contului de conectare în momentul creerii bazei de date precum si oricarui utilizator al bazei de date corespunzator unui cont de conectare cu rol de *System Administrators*.

Pentru lucrul în siguranta cu SQL server se recomanda crearea unor conturi care sa aiba toate drepturile asupra bazelor de date proprii, si numai drepturi de citire (dreptul *public*) asupra bazelor de date preinstalate.

Dupa conectarea unui utilizator cu rolul *Database Creators*, acesta poate sa creeze una sau mai multe baze de date asupra carora are drept de detinator (*dbo*). Pentru conturile *user1*, *user2*, .. s-au creat deja bazele de date proprii, carora li s-au dat acelasi nume cu al contului (*user1*, *user2* etc).

Conectarea pe un cont care nu are drepturi de administrare ofera drepturi limitate de acces la obiectele bazei de date. Utilizatorul poate să deschidă fereastra de proprietăți ale contului propriu (cu comanda *Properties* din meniul de context) și poate să modifice baza de date implicită. De asemenea, se pot vedea tabelele, funcțiile etc ale bazei de date *Northwind*, la care are acces, dar nu le poate modifica. În baza de date proprie utilizatorul are toate drepturile (creare, modificare și ștergere tabele, creare proceduri, etc.). După crearea bazei de date proprii, este recomandabil ca să se seteze această baza de date ca baza de date implicită a contului respectiv (prin comanda *Properties* în meniul de context al contului). De asemenea se poate anula (din conul *sa*) rolul *Database Creators* al contului.

În general, atunci când se modifică elementele unui director (*Databases*, *Logins*, etc), ele vor fi actualizate în panoul din *Enterprise Manager* numai dacă se dă comanda *Refresh* (din meniul de context corespunzător).

Pentru acomodarea cu sistemul SQL Server se selectează baza de date preinstalată *Northwind* și se inspectează tabelul *Employees* (prin comanda dublu clic atunci când este selectat tabelul respectiv sau prin selectarea opțiunii *Properties* din meniul de context). La această comandă se deschide o fereastră în care pentru fiecare atribut al relației (tabelului) sunt afișate numele, tipul de date, valoarea implicită, marcatul de cheie primară, etc. (Fig. 1.1).

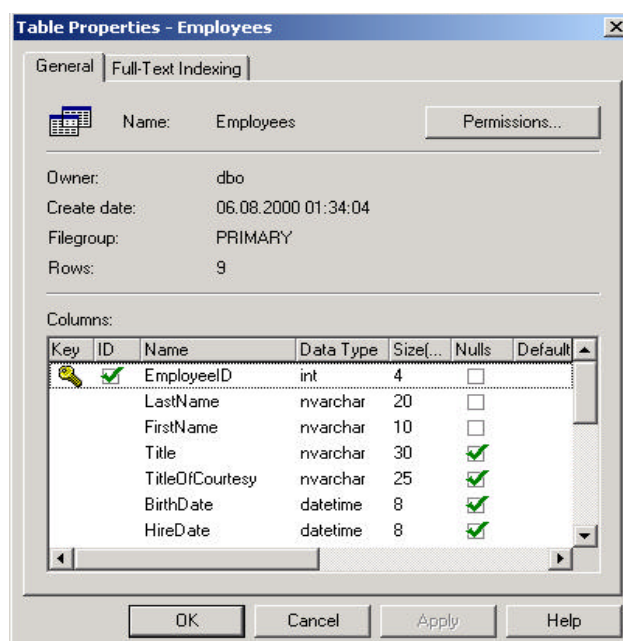


Fig. 1.1 Proprietățile tabelului *Employees* din baza de date *Northwind* (SQL Server).

Un tabel se poate deschide și în alte moduri: modul de proiectare (prin selecția opțiunii *Design Table* din meniul de context) sau modul *Open* care afișează conținutul tabelului, dacă utilizatorul are drepturi suficiente.

În *Enterprise Manager* proiectarea tabelelor, ca și a altor obiecte ale bazei de date, se poate realiza vizual, cele mai multe valori putând fi introduse prin selecția unei opțiuni din valorile valide oferite pentru fiecare tip de introducere, ceea ce face ca proiectarea să fie deosebit de ușoară.

1.2.1.2 Utilizarea programului *OSQL*

Utilitarul *osql* permite conectarea la serverul SQL pentru executarea instrucțiunilor Transact SQL (extensia procedurală a limbajului SQL pentru sistemul SQL Server), a fișierelor de scripturi (fișiere care conțin comenzi în limbajul Transact SQL) și a procedurilor stocate. Sintaxa simplificată de lansare a utilitarului *osql* pentru conectarea utilizatorului *user1* cu parola *parola1* la baza de date *nume_baza_date* este:

```
osql -U user1 -d nume_baza_de_date -P parola1
```

Daca nu se specifica numele bazei de date, atunci are loc conectarea la baza de date implicita a contului respectiv. Dupa conectare, programul afiseaza ca prompt de comanda numarul liniei de comanda (în ordine crescatoare începând cu 1) si se pot transmite spre executie loturi de executie Transact SQL (succesiune de instructiuni terminate cu comanda GO) care pot accesa baza de date respectiva. Baza de date curenta se poate schimba cu comanda: USE noua_baza_date. Executia unui fisier script care contine unul sau mai multe loturi Transact SQL se realizeaza cu comanda:

```
osql -U user -i nume_fisier -P parola
```

1.2.1.3 Utilizarea programului SQL Server Query Analyzer

Programul utilitar *SQL Server Query Analyzer* este un instrument grafic care permite executarea mai multor activitati de proiectare a bazelor de date si de executie a aplicatiilor, cum sunt: crearea si executarea interogarilor, testarea procedurilor stocate, operatii de introducere, stergere sau modificare a datelor în tabele, etc.

Programul SQL Server Query Analyzer se poate lansa din sistemul de operare cu comenzile *Start->Programs->Microsoft SQL Server->Query Analyzer*, sau din *Enterprise Manager* cu comenzile *Tools ->SQL Query Analyzer*. La conectare trebuie sa se specifice serverul, contul si parola de conectare. În fereastra programului Query Analyzer sunt afisate o bara de meniu, o bara de instrumente si doua panouri (Fig. 1.2).

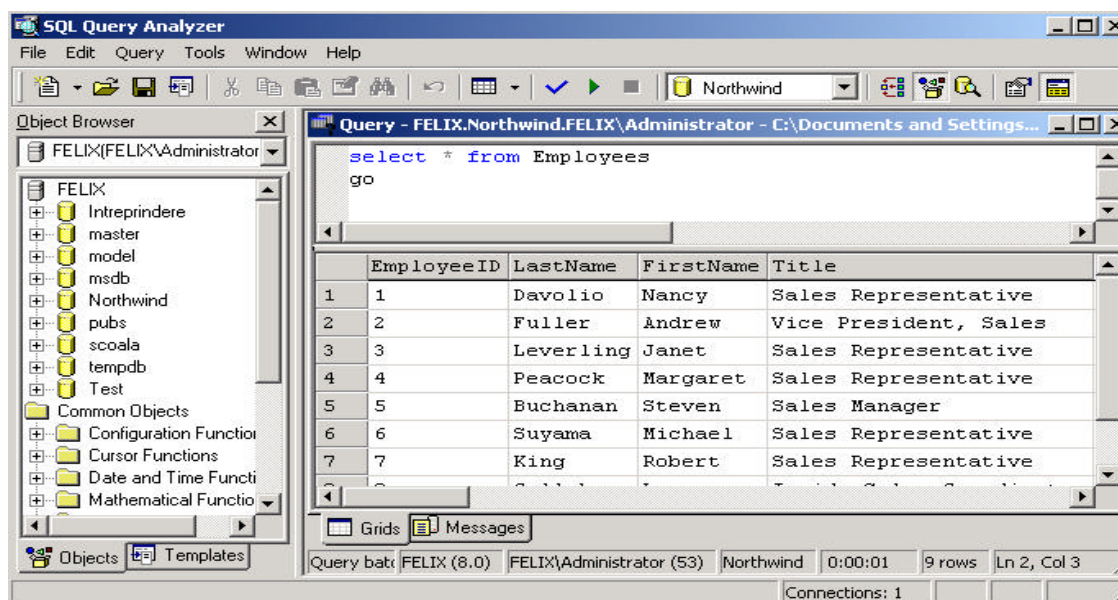


Fig. 1.2 Fereastra de afisare a programului SQL Query Analyzer.

În panoul din stânga sunt afisate obiectele serverului la care este conectat programul, grupate în doua directoare. Primul director are numele instantei serverului SQL si este radacina subdirectoarelor corespunzatoare bazelor de date administrate de server. Cel de-al doilea director are numele *Common Objects* si este radacina mai multor subdirectoare (*Configuration Functions*, *Cursor Functions*, etc.).

În panoul din dreapta este afisata o fereastra pentru interogari (*Query*) pentru executia unor operatii asupra bazei de date selectate în bara de instrumente (în figura de mai sus s-a selectat baza de date *Northwind*). Rezultatul executiei este afisat în partea de jos a panoului în doua pagini selectabile, o grila de date (*Grids*) si o lista de mesaje privind modul de executie a interogarii (*Messages*).

Comenzile din fereastra *Query* sunt instructiuni SQL (sau Transact-SQL) care se pot introduce fie manual (de la tastatura), fie prin citirea (deschiderea) unui fisier de script (cu comanda *File->Open*). Lansarea în executie a comenzilor din fereastra *Query* se realizeaza cu comanda *Execute*, care se poate lansa fie din meniul de context al ferestrei, fie din meniul *Query*.

1.2.2 SISTEMUL DE GESTIUNE MICROSOFT ACCESS

În continuare este prezentat sistemul de gestiune a bazelor de date Microsoft Access, care face parte din pachetul Microsoft Office. MS Access se folosește în aplicații mici de baze de date, dar, permite proiectare vizuală a bazelor de date și a aplicațiilor într-un mediu integrat folosind instrumentele de proiectare deosebit de flexibile și "prietenoase". De aceea, mediul MS Access poate fi considerat un bun punct de pornire pentru studierea proiectării bazelor de date.

Instalarea programului sub sistemele de operare Windows NT/2000/XP are loc la instalarea pachetului Microsoft Office 2000 și nu necesită setări speciale.

Când se lansează programul MS Access, pe ecran apare o fereastră de dialog care oferă posibilitatea selectării unei opțiuni dintre crearea unei baze de date noi sau deschiderea uneia deja existente. În MS Access, nu se pot deschide simultan mai multe baze de date, dar se pot lansa mai multe copii ale programului și fiecare poate deschide o bază de date diferită.

Pentru selectarea și crearea obiectelor unei baze de date (existentă sau nou creată) se lucrează cu fereastra *Database*. Modul de afișare *Database* include o bară de meniuri, o bară de instrumente și fereastra *Database* care ocupă partea centrală a suprafeței de lucru. Din cele șapte tipuri principale de obiecte pe care programul MS Access le acceptă, vor fi folosite mai frecvent următoarele: *Tables* (*Tabele*), care corespund relațiilor și *Queries* (*Interogări*), care corespund comenzilor de selectare a datelor din tabele.

Pentru fiecare tip de obiect MS Access se pot folosi diferite comenzi date prin butoanele din bara de instrumente ale ferestrei *Database*:

- *Open*: deschiderea obiectului (ceea ce înseamnă "vizualizare" sau "afișare" pentru tabele și "execuție" pentru interogări).
- *Design*: proiectarea obiectului (editarea obiectului selectat).
- *New*: crearea unui obiect nou de tipul curent.

Pentru acomodarea cu sistemul MS Access, se poate deschide una din bazele de date preinstalate (*Samples: Northwind, Contacts, Addresses*). Dacă, de exemplu, se deschide baza de date Northwind, se pot vedea tabelele (*Tables*), interogările (*Queries*), formularele (*Forms*), etc. acestea. Oricare din tabelele existente (*Categories, Customers, Employees, etc*) se poate deschide în modul proiectare (*Design View*) sau execuție (*Open*) prin selectarea opțiunii dorite din meniul de context (care se afișează la comanda de apăsare pe butonul din dreapta al mouse-ului pe numele tabelului respectiv). În Fig. 1.3 se poate vedea conținutul ferestrei de proiectare (modificare) a tabelului *Employees* din baza de date Northwind.

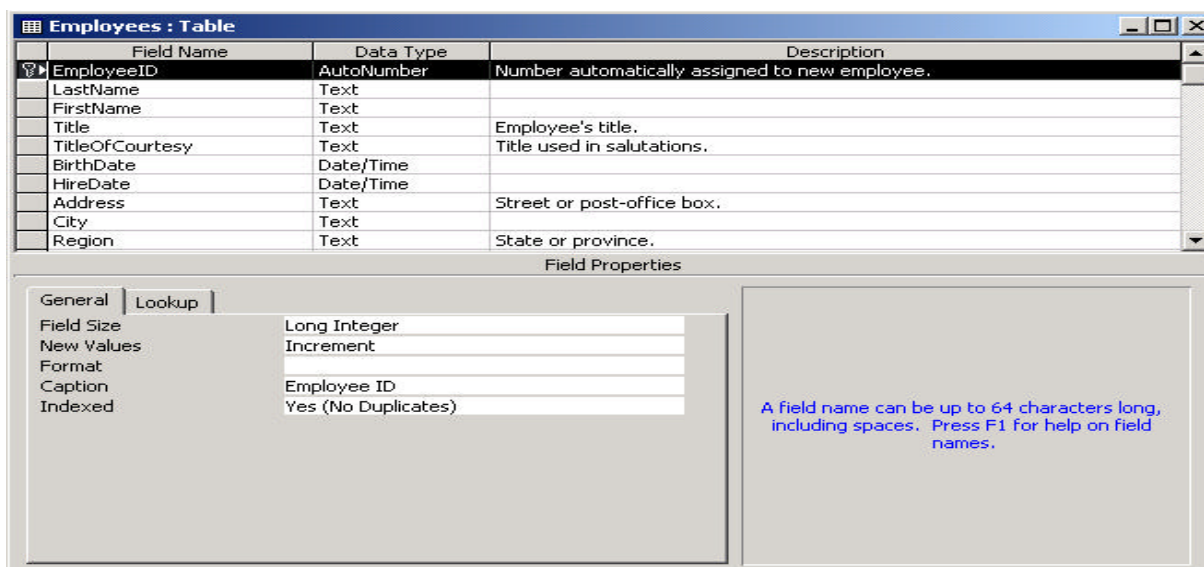
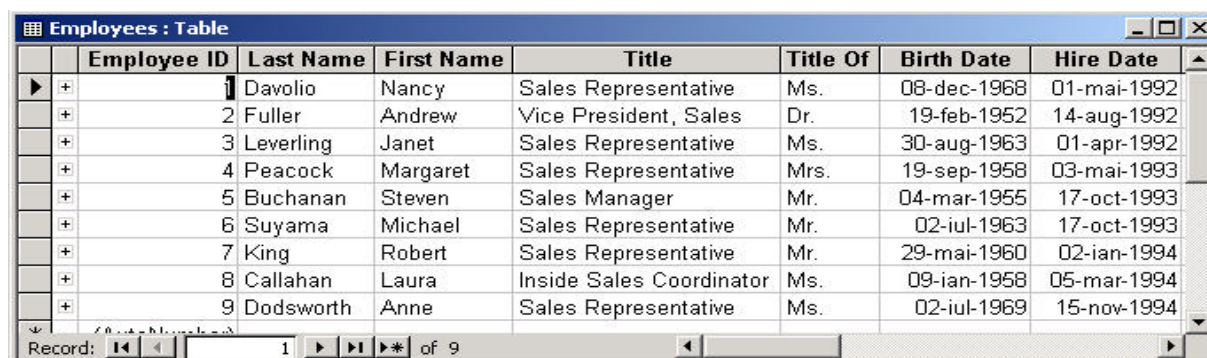


Fig. 1.3 Fereastra de proiectare (modificare) a tabelului *Employees* (MS Access).

Fereastra de afisare este împartita în doua panouri: panoul de atribute (care contine trei coloane cu informatii despre atributele tabelului) si panoul de proprietati (*Fields Properties* - care contine descrierea detaliata a fiecarui atribut). În prima coloana a panoului de atribute este afisat numele atributului relatiei (coloana în tabel – *Field Name*); în a doua coloana este afisat tipul de date al atributului respectiv (*Data Type*), iar în a treia coloana este afisata o descriere a atributului (*Description*). Pentru fiecare atribut selectat, în panoul de proprietati se afiseaza detaliat proprietatile atributului respectiv. În figura de mai sus se poate observa ca atributul `EmployeeID` este cheia primara (are marcat ca simbol o cheie) de tip întreg, cu proprietatea *Autoincrement* care asigura valori unice ale acesteia în cadrul relatiei. Pe atributul cheii primare este definit indexul primar (care nu admite duplicate).

În Fig. 1.4 se poate vedea continutul (starea) tabelului `Employees`, care se obtine în modul de executie selectând optiunea *Open* din meniul de context.



Employee ID	Last Name	First Name	Title	Title Of	Birth Date	Hire Date
1	Davolio	Nancy	Sales Representative	Ms.	08-dec-1968	01-mai-1992
2	Fuller	Andrew	Vice President, Sales	Dr.	19-feb-1952	14-aug-1992
3	Leverling	Janet	Sales Representative	Ms.	30-aug-1963	01-apr-1992
4	Peacock	Margaret	Sales Representative	Mrs.	19-sep-1958	03-mai-1993
5	Buchanan	Steven	Sales Manager	Mr.	04-mar-1955	17-oct-1993
6	Suyama	Michael	Sales Representative	Mr.	02-iul-1963	17-oct-1993
7	King	Robert	Sales Representative	Mr.	29-mai-1960	02-ian-1994
8	Callahan	Laura	Inside Sales Coordinator	Ms.	09-ian-1958	05-mar-1994
9	Dodsworth	Anne	Sales Representative	Ms.	02-iul-1969	15-nov-1994

Fig. 1.4 Continutul (starea) tabelului `Employees`.

În acest mod de afisare se pot identifica usor coloanele tabelului (corespunzatoare atributelor relatiei) si liniile (corespunzatoare tuplurilor relatiei) si chiar se pot introduce date noi în tabel. Însa, acest mod de introducere nu este nici comod, nici sigur, dat fiind ca un utilizator poate avea acces complet la date si le poate distruge intentionat sau din greseala. De aceea, în exploatarea bazelor de date nu se foloseste acest mod de introducere directa a datelor, ci un mod mai comod si mai sigur, prin intermediul aplicatiilor de baze de date.

În sistemul MS Access aplicatiile se pot dezvolta în mod vizual în mediul integrat, folosind formulare (*forms*) care ofera posibilitatea de a modifica numai anumite atribute, de multe ori prin selectia dintr-o lista valori care respecta anumite constrângeri, evitând introduceri eronate. Formularele pot interzice operatiile riscante (stergerea datelor, modificarea unor valori importante, etc.) si sunt mult mai usor de utilizat decât introducerea directa în tabele. În formularul de mai sus se poate observa ca nu se permite stergerea nici unei înregistrari din tabel, nu se permite modificarea identificatorului unui angajat (`EmployeeID`), iar valoarea atributului `ReportsTo` poate fi modificata numai prin selectarea acesteia dintr-o lista de optiuni (*combo box*).

1.2.3 SISTEMUL DE GESTIUNE ORACLE

Sistemul Oracle este un sistem de gestiune a bazelor de date multi-utilizator puternic, cu implementari pe toate platformele (Windows, Unix, Linux), care ofera atât performante ridicate cât si un grad înalt de protectie si securitate a datelor. Sistemul de gestiune Oracle9i (care este instalat în laborator), ca si diferite instrumente de dezvoltare a aplicatiilor de baze de date (Oracle Application Server, JDeveloper, Oracle Forms etc.), se pot obtine de la adresa <http://www.oracle.com> si termenii licentei permit utilizarea acestor sisteme în scopuri necomerciale pe o perioada nelimitata; pentru utilizarea în scopuri comerciale trebuie sa fie platite licentele corespunzatoare

Un server Oracle este compus dintr-o *baza de date* (colectia de date persistente, inclusiv fisiere de control si jurnale – *log*) si o *instanta Oracle* (serverul), care consta din mai multe procese care asigura gestiunea datelor memorate. În baza de date gestionata de un server se pot defini mai multe “*scheme de obiecte*” (*schema objects*), fiecare reprezentând o colectie logica de obiecte persistente (tabele, vederi, indexuri, etc). O astfel de schema corespunde unei scheme externe (vedere utilizator) din definitia standard a arhitecturii sistemelor de baze de date.

În fiecare stație din rețeaua din care se dorește accesarea serverului Oracle se instalează un client Oracle și se setează mai multe opțiuni de conectare la server (stația pe care se afla serverul, numele instanței Oracle, protocoalele de comunicație folosite etc.).

La instalarea serverului și a clientilor Oracle sunt instalate mai multe programe utilitare (instrumente software) pentru administrarea sistemului și pentru dezvoltarea bazelor de date și a aplicațiilor, cu interfața grafică sau din linie de comandă. De asemenea sunt create mai multe conturi (utilizatori) care au nume și parole predefinite, descrise în documentație. De regulă, la instalare se modifică parolele conturilor de administrare (*sys, system*) astfel încât drepturile de administrare să nu poată fi folosite de utilizatori neautorizați. În schimb, contul *SCOTT* poate fi păstrat așa cum a fost creat la instalare (cu parola *TIGER*) și poate fi folosit pentru inspectarea schemei cu același nume (*SCOTT*), care conține câteva tabele care se referă la activitatea unei întreprinderi: *DEPT* (departamente-secții), *EMP* (employees – angajați) etc. Se pot crea utilizatori noi cu comanda *Create* din meniul de context *Security->Users*. În laborator au fost creați mai mulți utilizatori (*user1, user2, ..*), fiecare având atribuit rolurile *CONNECT* și *RESOURCE* și privilegiul de sistem *UNLIMITED TABLESPACE*. Corespunzător fiecărui utilizator (cont) sistemul de gestiune creează în mod automat o schemă cu același nume.

Pentru lansarea sistemului Oracle este necesar să fie pornite manual (dacă s-a renunțat la opțiunea de pornire automată setată implicit la instalare) serviciile Oracle (cel puțin *OracleServiceName_server*, *OracleMTSRecoveryService* și *OracleHome92TNSListener* din *Administrative Tools->Services*). Administrarea sistemului Oracle se face prin intermediul programului *Oracle Enterprise Manager* care permite crearea utilizatorilor și stabilirea privilegiilor acestora, salvarea și refacerea datelor etc.

Ca programe client (instrumente) care se pot folosi pentru dezvoltarea aplicațiilor de baze de date, cel mai simplu este utilitarul *SQL*Plus*, care oferă o consolă pentru lucrul interactiv cu sistemul de gestiune printr-o linie de comandă.

*SQL*Plus* se poate lansa de pe orice stație din rețea pe care a fost instalat un client Oracle. Pentru lansarea programului se specifică numele unui utilizator (care a fost definit în prealabil în server, de exemplu *SCOTT*), parola (*TIGER*) și un șir de conectare (*Host String*). Șirul de conectare conține numele public al bazei de date (numele în domeniul de nume al rețelei) care a fost stabilit la instalarea serverului. De exemplu, pentru serverul Oracle9i instalat în laborator, conectarea se face cu numele *Server12_server125a.tech.pub.ro*.

După conectarea la server, se afișează invitația (promptul) de comenzi *SQL*Plus (SQL>)* la care se pot introduce instrucțiuni SQL, blocuri PL/SQL (acestea vor fi prezentate în Capitolul 4) sau comenzi suplimentare *SQL*Plus*, care permit stabilirea formatelor de afișare a datelor.

Instrucțiunile sau blocurile PL/SQL se pot introduce pe una sau mai multe linii de comandă, iar la introducerea caracterului de terminare are loc executia acestora și afișarea la consolă a rezultatului. Alternativ, comenzile se pot grupa într-un fișier de script (cu extensia *sql*) care se poate lansa în execuție cu comanda *start nume_fisier*.

Un alt program utilitar de dezvoltare a aplicațiilor de baze de date Oracle este *SQL*Plus Worksheet*, care este o extensie cu interfața grafică a programului *SQL*Plus*, și are posibilități de lucru mai “prietenoase”. Linia de comandă este înlocuită cu o “foaie de lucru” (*worksheet*), în care se pot introduce, edita și lansa în execuție o secvență de comenzi (instrucțiuni SQL, blocuri PL/SQL, comenzi suplimentare de formatare, etc.).

Lansarea în execuție a programului *SQL*Plus Worksheet* se poate face din orice stație pe care a fost instalat un client Oracle. La conectare se cere numele unui utilizator (*User*), parola acestuia (*Password*) și numele serviciului Oracle la care se face conectarea (*Service Name*). Numele acestui serviciu este numele global al bazei de date Oracle stabilit la instalare (în laborator numele acesteia este *Server12_server125a.tech.pub.ro*).

După lansare, programul afișează un meniu de comenzi, o bară de instrumente și două ferestre. Fereastra de sus este o fereastră de editare (*foaie de lucru*) în care se pot introduce (sau încărca dintr-un fișier de script cu comanda *Open*) instrucțiunile de executat; fereastra de jos este o fereastră de afișare a rezultatelor executiei instrucțiunilor. Execuția instrucțiunilor se lansează cu comanda *Execute* (din meniu sau bară de instrumente).

1.2.4 SISTEMUL DE GESTIUNE A BAZELOR DE DATE MYSQL

MySQL este un sistem de gestiune a bazelor de date relationale gratuit, de tipul *open source*, ceea ce înseamnă ca orice utilizator poate prelua acest produs din Internet (de la adresa www.mysql.com), îl poate utiliza și chiar modifica. Versiunea oferită este în permanentă îmbunătățită și poate fi preluată pentru actualizare de utilizatori.

MySQL poate fi instalat atât sub sistemele de operare Windows (95/98/NT/2000/XP) cât și Linux (în diferite distribuții, ca Red Hat, Suse, etc.) și poate fi folosit în numeroase aplicații (chiar aplicații comerciale). Acest sistem este deosebit de compact, ușor de instalat și de administrat. Versiunea actuală (MySQL 5.0, ianuarie 2006) adaugă unele din trăsăturile avansate de lucru cu bazele de date relationale (vederi, proceduri stocate, triggeri, etc.), care lipseau în versiunile precedente. Dat fiind că sistemul MySQL este și simplu de administrat, el este foarte potrivit pentru a începe studiul bazelor de date.

Sistemul de gestiune MySQL se instalează ca un server care poate gestiona mai multe baze de date, fiecare dintre acestea având un nume distinct și fiind memorată într-un subdirector al directorului `..\mysql\data`. Pentru accesarea unei baze de date, utilizatorii se conectează la server folosind conturi de conectare, care oferă anumite drepturi de administrare și accesare a datelor.

1.2.4.1 Instalarea și pornirea serverului MySQL

Instalarea din laborator și exemplele din acest îndrumar se bazează pe versiunea 4.1 pentru sistemele de operare Windows NT/2000/XP. Instalarea serverului MySQL se face foarte simplu, executând programul de instalare `Setup.exe`, după dezarhivarea fișierului de distribuție. Toate informațiile privind instalarea, administrarea și particularitățile de limbaje și biblioteci de programare se găsesc în manualul din pachetul de instalare (`..\mysql\docs\manual.html`).

După instalarea MySQL, pornirea serverului în Windows NT (NT 4 cu Service Pack 3 sau mai nou)/2000/XP se poate face în două moduri: ca serviciu, sub controlul SCM (*Service Control Manager*), sau ca program executabil. Pentru instalarea serverului MySQL ca serviciu Windows, se oprește mai întâi serverul (dacă acesta era deja pornit) cu comanda:

```
C:\mysql\bin> mysqladmin -u root shutdown
```

Apoi se instalează serverul MySQL ca serviciu Windows prin comanda:

```
C:\mysql\bin> mysqld-max-nt --install
```

După instalarea serviciului, pornirea acestuia se face cu comanda: `NET START MySQL`, sau cu utilitarul *Services* din *Control Panel* (în secțiunea *Administrative Tools* sub Windows 2000/XP). Serviciul astfel instalat va porni automat la pornirea sistemului de operare, dar această opțiune se poate modifica (prin utilitarul *Services*), astfel încât pornirea să se facă manual.

Atunci când este lansat ca serviciu, serverul MySQL poate fi oprit oricând cu utilitarul *Services*, cu comanda `NET STOP MySQL`, sau folosind utilitarul *mysqladmin* (cu comanda aratăta mai sus). De asemenea, serverul este oprit automat la oprirea (*shutdown*) sistemului de operare.

Dacă nu se dorește funcționarea ca serviciu a serverului MySQL, acesta poate fi lansat ca program executabil cu comanda:

```
C:\mysql\bin>mysqld-max-nt --standalone
```

După ce serverul MySQL a fost pornit, se pot executa diferite programe client de acces la baza de date, dintre care unele pot fi programe utilitare continute în kit-ul de distribuție și instalate odată cu instalarea serverului, iar altele pot fi programe de aplicații dezvoltate pentru o anumită funcționalitate.

Programele utilitare din kit-ul de distribuție (*mysqlshow*, *mysqladmin*, *winmysqladmin*, *mysql*, etc.) sunt folosite pentru administrarea bazei de date, pentru testare, pentru importul sau exportul bazelor de date, etc. Acestea se găsesc în subdirectorul *bin* al directorului *mysql* unde s-a făcut instalarea (`...\mysql\bin`). Dacă se include acest subdirector în calea sistemului (*path*), atunci lansarea diferitelor toolset-uri se face mai simplu, direct cu numele acestora din orice director.

Utilitarul *mysqlshow* afișează informații despre bazele de date existente; dacă se execută *mysqlshow* imediat după instalarea și pornirea serverului MySQL, acesta va afișa numele celor două baze de date care se creează automat la instalare: *mysql* și *test*.

Utilitarul *mysqladmin* permite administrarea bazelor de date: pornirea si oprirea serverului, crearea de noi utilizatori ai bazei de date, stabilirea parolelor acestora, a drepturilor de acces, etc.

Serverul MySQL poate gestiona simultan mai multe baze de date accesate de mai multi utilizatori. Dintre bazele de date gestionate de serverul MySQL, baza de date *mysql* este creata implicit la instalarea serverului (în sistemul de operare Windows) si este formata din mai multe tabele care contin informatii despre starea curenta a serverului: numele bazelor de date existente, numele calculatoarelor gazda (*host*), numele utilizatorilor, drepturile utilizatorilor asupra tabelelor si a coloanelor. Operatiile de creare (stergere) a bazelor de date, creare (stergere) a utilizatorilor, schimbarea drepturilor de acces, se pot face modificând continutul tabelelor bazei de date *mysql*. Unele din aceste operatii pot fi efectuate si prin intermediul unor programe utilitare (*mysqladmin* sau *wmysqladmin*).

Utilizatorii serverului MySQL sunt identificati prin numele utilizatorului (*user*) si, optional, printr-o parola (*password*). Desi se accepta utilizatori fara parola, este recomandabil sa se introduca parola pentru toti utilizatorii, pentru a proteja datele memorate în bazele de date. Utilizatorii serverului MySQL nu sunt identici cu utilizatorii sistemului de operare (Windows sau Linux) si trebuie sa fie creati dupa instalarea si pornirea serverului. Drepturile de acces ale utilizatorilor se stabilesc dupa crearea acestora prin comenzi SQL. Utilizatorul *root*, creat implicit la instalarea serverului MySQL este contul de administrare si are toate drepturile de creare si modificare a bazelor de date.

De asemenea, dupa instalarea serverului MySQL în Windows, toti utilizatorii locali au drepturi de acces complete la toate bazele de date, fara specificarea unei parole de acces. Pentru a asigura o mai mare securitate, dupa instalarea si pornirea serverului, administratorul bazei de date va limita drepturile de acces ale utilizatorilor, conform cerintelor de functionare a bazei de date. Prima actiune este de a elimina dreptul de acces al unui utilizator oarecare (utilizator anonim) la baza de date *test*, prin urmatoarele comenzi (care vor fi înțelese pe parcursul lucrurilor):

```
C:\mysql\bin>mysql mysql
mysql> DELETE FROM user WHERE Host='localhost' AND User='';
mysql> QUIT
C:\mysql\bin>mysqladmin reload
```

Dupa acesta, se va introduce o parola pentru utilizatorul de administrare (*root*) cu comanda:

```
C:\mysql\bin>mysqladmin -u root password parola_admin
```

Din acest moment, la orice acces al utilizatorului *root* va trebui sa fie introdusa parola setata. S-a presupus ca serverul *mysql* a fost instalat în directorul *C:\mysql*; daca a fost instalat în alt director, atunci în comenzile de mai sus se trece numele directorului respectiv.

Utilitarul *winmysqladmin* prezinta o interfata grafica pentru administrarea sistemului (pornirea si oprirea serverului instalat ca serviciu, inspectarea bazelor de date si a tabelelor, etc.).

Utilitarul *mysql* este un program client care permite lucrul interactiv cu serverul MySQL prin intermediul mai multor comenzi, dintre care majoritatea sunt comenzi în limbajul SQL, iar celelalte sunt comenzi utilitare. Exemple de comenzi utilitare: comanda *help* (care listeaza comenzile disponibile), comanda *use* (prin care una din bazele de date gestionate de server este setata ca baza de date curenta), comanda *describe* (care afiseaza informatii despre attributele unui tabel), etc. Utilitarul *mysql* se mai numeste si monitor sau consola si permite crearea unor mici aplicatii interactive sau de testare a bazelor de date.

Lansarea programului *mysql* si conectarea la serverul MySQL se poate face dintr-o fereastra DOS specificând numele utilizatorului, parola si numele bazei de date astfel:

```
C:\mysql --user=nume_user --password=parola_user nume_baza_date
```

În aceasta forma a comenzii, parola se afiseaza clar, de aceea este preferabila o alta forma (sintaxa) a comenzii de lansare a utilitarului *mysql*:

```
C:\mysql -u nume_user -p nume_baza_date
```

Dupa aceasta comanda, la consola apare invitatia de introducere a parolei (*Enter password:*), dupa care, parola introdusa nu mai este afisata clar (ci cu semne asterix).

Dupa lansarea monitorului *mysql*, se afiseaza promptul de introucere a comenzilor (*mysql>*) si se pot introduce comenzi în linia de comanda, care sunt transmise serverului, iar rezultatul obtinut

poate fi afisat în fereastra de afisare sau scris într-un fisier de redirectare. Comanda *quit* închide conexiunea utilizatorului la serverul MySQL si monitorul *mysql*.

Din linia de comanda *mysql* se pot executa comenzi în mod direct, prin introducerea lor de la tastatura, sau în mod lot (batch), prin memorarea acestora într-un fisier script (în general cu extensia *.sql*) si lansarea în executie cu comanda:

```
mysql> source nume_fisier;
```

Fisierului sursa trebuie sa se afle în directorul curent (din care sa lansat monitorul *mysql*), într-un director care se afla în calea de cautare a programelor executabile (*path*), sau se introduce numele complet (inclusiv calea).

1.2.4.2 Crearea bazelor de date si a utilizatorilor MySQL

Crearea unei noi baze de date se face din monitorul *mysql* de catre un utilizator cu drepturi de creare baze de date, folosind comanda:

```
mysql> CREATE DATABASE nume_baza_date;
```

Aceasta comanda se poate executa de catre utilizatorul *root*, care are toate drepturile (inclusiv dreptul de creare a bazelor de date). Bineînțeles, pentru a da comanda de mai sus, utilizatorul *root* trebuie mai întâi sa se conecteze si sa-si introduca parola. Dupa crearea unei baze de date, se atribuie drepturi de acces unor anumiti utilizatori, care vor crea tabelele si vor opera cu acestea. Orice utilizator poate afla numele tuturor bazelor de date gestionate de serverul MySQL folosind comanda:

```
mysql> SHOW DATABASES;
```

Crearea unor noi utilizatori se poate face în doua moduri: prin instructiunea *GRANT* sau înscriind direct linii în tabela *user* a bazei de date *mysql*. În ambele situatii, utilizatorul care da comenzile trebuie sa aiba drepturi de creare a tabelor (cel mai fercvent se foloseste contul *root*).

Prin comanda:

```
mysql> GRANT ALL PRIVILEGES ON nume_baza_date.*  
->TO nume_user@nume_host IDENTIFIED BY 'parola_user';
```

se creaza utilizatorul cu numele *nume_user* si parola *parola_user* care are drepturile (privilegiile) (*SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCE, INDEX, ALTER*) asupra bazei de date *nume_baza_date*. Aceasta operatie produce crearea utilizatorului prin adaugarea unei linii noi în tabela *user* a bazei de date *mysql* si a drepturilor de acces la baza de date specificata prin adaugarea unei linii în tabela *db* din baza de date *mysql*. Acest utilizator nu are drepturi de acces la nici-o alta baza de date si nici dreptul de creare a altor utilizatori (*GRANT*).

Alternativ, aceeași operatie de creare a utilizatorului cu numele *nume_user* si de acordare a drepturilor listate mai sus asupra bazei de date *nume_baza_date* se poate face folosind operatiile de înscriere directa în tabele bazei de date *mysql* astfel:

```
mysql> INSERT INTO user (Host, User, Password)  
->VALUES ('localhost', 'nume_user', PASSWORD('parola_user'));  
mysql> INSERT INTO db VALUES  
-> ('localhost', 'nume_baza_date', 'nume_user', '  
-> Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y');  
mysql> FLUSH PRIVILEGES;  
mysql> QUIT;
```

De remarcat faptul ca parola se introduce folosind functia *PASSWORD()* care cripteaza sirul de caractere introdus ca argument. Comanda *FLUSH PRIVILEGES* trebuie introdusa pentru ca drepturile acordate utilizatorului sa fie efective (trebuie ca serverul sa reîncarce tabela de drepturi).

Exercitii - Capitolul 1

Instalarea sistemelor de gestiune descrise în continuare si stabilirea drepturilor de administrare se va face individual de catre fiecare student pe calculatorul personal, pentru a studia si realiza proiecte de baze de date. În laborator sistemele de gestiune sunt instalate si toate operatiile de administrare au fost deja realizate, astfel încât fiecare student va avea la dispozitie un cont (*user1*, *user2*, *user3*,...) si o parola. În Oracle schema unui utilizator se creeaza implicit cu acelasi nume cu al contului; în SQL Server a fost creata câte o baza de date cu acelasi nume cu numele contului proprietar, în care fiecare student va realiza toate tabelele, interogările si celelalte operatii cu baza de date. Lucrarile si exercitiile pentru MySQL se vor efectua suplimentar, pe calculatoarele personale.

În aceasta lucrare si în urmatoarele, exercitiile marcate cu asterisc au un grad mai ridicat de dificultate si sunt optionale, sau se vor efectua numai pe calculatoarele personale, daca se refera la operatii pentru care nu aveti drepturile necesare sau sistemele nu sunt instalate în laborator.

1.1* Instalati sistemul SQL Server, având grija sa instalati imediat (preferabil decuplati de la reseaua Internet) pachetul de eliminare a vulnerabilitatii la virusul SQL Worm. Verificati eliminarea vulnerabilitatii cu un program adecvat (de exemplu *FixSQLex* de la Symantec).

1.2* Lansati programul utilitar *SQL Server Enterprise Manager* cu contul *sa* (daca aveti permisiunea de a folosi acest cont) si creati un nou utilizator (cu numele dorit) si acordati-i rolul de sistem *Database Creators* si dreptul *public* de acces la baza de date pre-instalata *Northwind*. Dupa aceasta va deconectati din contul *sa*, va re-conectati pe contul propriu si creati o noua baza de date (preferabil cu acelasi nume cu al contului, asa cum s-au creat conturile si bazele de date în laborator, dar acest lucru nu este obligatoriu). Setati aceasta baza de date ca baza de date implicita a contului propriu (prin comanda *Properties* din meniul de context al contului).

1.3 Lansati programul utilitar *SQL Server Enterprise Manager* cu propriul cont si urmariti informatiile afisate în ferestrele programului: bazele de date pe care aveti dreptul sa le accesati, tabelele acestora, utilizatorii etc.

1.4 Verificati utilizatorii bazei de date proprii (în subdirectorul *Users*). Daca s-au executat corect comenzile precedente, atunci baza de date proprie va avea utilizatorul proprietar (*dbo*) asignat contului propriu de conectare.

1.5 Lansati programul *SQL Server Query Analyzer* si urmariti comenzile si informatiile afisate. Introduceti comenzile necesare pentru a obtine situatia din Fig. 1.2. Urmariti rezultatele afisate în pagina *Grids* si în pagina *Messages* din fereastra programului.

1.6* Instalati programul MS Access si lansati-l în executie deschizând baza de date preinstalata *Northwind*.

1.7 În MS Access deschideti în modul proiectare (*Design View*) fiecare din tabelele bazei de date *Northwind*: *Categories*, *Customers*, *Employees*, etc. si observati care sunt coloanele fiecarui tabel si tipurile de date ale acestora.

1.8 În fiecare tabel al bazei de date *Northwind* din MS Access urmariti modul de definire a cheilor primare. Observati ca majoritatea tabelelor au chei primare artificiale definite ca numere întregi cu proprietatea *AutoNumber*. În fereastra *Field Properties* urmariti modul de definire a constrângerilor de domeniu: optiunea *NOT NULL* (*Required -Yes*), optiunea *Default Value* si optiunea *CHECK* (proprietatea *Validation Rule*).

1.9* Instalati serverul Oracle9i si configurati în mod corespunzator drepturile de administrare ale sistemului (parolele pentru conturile de administrare *sys* si *system*). Lansati programul *Oracle Enterprise Manager* din contul *sys* cu rolul *sysdba* si inspectati obiectele afisate (schemele de relatii, utilizatorii, etc.). Creati un nou cont si acordati-i drepturi de utilizator neprivilegiat (urmarind drepturile utilizatorului *SCOTT*). Se va observa ca în mod implicit sistemul va crea o schema cu acelasi nume cu numele utilizatorului creat.

1.10 Lansati programul *Oracle SQL* Plus* folosind un server (*Host String*), un cont si o parola cunoscute (în laborator folositi host *Server12_Server125a.tech.pub.ro*, contul *SCOTT* cu parola *TIGER*). Afisati continutul tabelelor *EMP* si *DEPT* ale bazei de date cu comenzile: `SELECT * FROM EMP; SELECT * FROM DEPT;`

1.11 Lansati programul *Oracle SQL* Plus Worksheet* folosind acelasi nume de server (*Service Name*), cont si parola. Afisati continutul tabelelor *EMP* si *DEPT* ale bazei de date introducând în fereastra de editare (foaia de lucru) comenzile: `SELECT * FROM EMP; SELECT * FROM DEPT;`

1.12 Creati un fisier script cu comenzile de mai sus, încarcati acest fisier în fereastra de editare a programului *SQL* Plus Worksheet* (cu comanda *Open*) si executati aceste instructiuni.

1.13* Instalati serverul MySQL si porniti serverul ca serviciu Windows sau ca program executabil.

1.14* Stabiliti drepturi de acces convenabile pentru utilizatori: eliminati dreptul utilizatorilor anonimi de a accesa baza de date *mysql* si introduceti o parola pentru utilizatorul *root*.

1.15* Lansati monitorul *mysql* ca utilizator *root* (daca aveti acest drept) si creati o noua baza de date si un utilizator propriu, caruia sa-i acordati toate drepturile asupra acestei baze de date. Listati comenzile oferite de monitor (cu comanda *help ;*) si bazele de date gestionate de server (cu comanda *show databases ;*).

1.17 Lansati monitorul *mysql* cu contul propriu si încercati aceleasi comenzi ca mai sus. Observati ce drepturi aveti ca utilizator neprivilegiat.