

# Baze de date

## - Lucrare de laborator 1 -

### NOTIUNI INTRODUCATIVE PRIVIND SISTEMELE DE GESTIUNE A BAZELOR DE DATE

#### 1. NOTIUNI TEORETICE

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 obiectrelational. 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  $A_i$ , apartinând 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 corespunzătoare.

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 *relatiei*.

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 funcționale, multivaloarea 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 străine* 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 coloana. 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

relatie nu pot exista doua (sau mai multe) tupluri care sa contina aceeași combinație de valori ale tuturor atributelor. De obicei, într-o schema de relatie exista o submultime de atribute SK cu proprietatea ca, în orice stare s-ar afla relația, nu exista două tupluri distincte ale relației care sa aiba aceeași combinație de valori ale atributelor submultimii respective.

O *supercheie* (*superkey*) a unei relații este o submultime (SK) de atribute ale relației care prezinta proprietatea de unicitate, adica orice combinație de valori ale atributelor supercheii este unica pentru orice stare a relației.

Acest lucru înseamna ca, daca se cunoaste combinația 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 definiției de mai sus, o cheie candidata CK trebuie sa prezinte proprietatea de unicitate (nu exista două tupluri diferite ale relației care sa contina aceeași combinație de valori ale atributelor cheii CK) și 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 *simpă* (alcatuita dintr-un singur atribut), sau *compusa* (alcatuita din mai multe atribute).

Atunci când exista mai multe chei candidate, una dintre ele se alege ca și *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 și identificare a tuplurilor relației. În plus, se impune ca atributelor cheii primare sa nu admita valori de NULL sa nu fie modificate prin operații 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 și cheie primara.

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

O cheie primara compusa din atributele existente ale tipului de entitate se numeste *cheie naturala*. În general, cheile naturale sunt compuse din mai multe atribute (ceea ce produce scaderea eficienței operațiilor relationale) și de cele mai multe ori se folosesc *chei artificiale*. O cheie primara artificiala este un atribut care se adauga în schema relației pentru identificarea unica a tuplurilor. De exemplu, în relația 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 relației și 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 relații: cheia straina.** O *cheie straina* (*foreign key*) este o multime de atribute FK ale unei relației R1 care refera relația R2 și satisface următoarele condiții: (a) atributele cheii straine FK sunt definite pe domenii compatibile cu cele ale atributelor unei cheii candidate CK a relației R2 și (b) combinația de valori ale atributelor FK într-un tuplu oarecare din relația R1, fie este identica cu combinația de valori ale atributelor CK a unui tuplu oarecare din starea curenta a relației R2, fie ia valoarea NULL. Cheia straina realizeaza asocierea N:1 între relațiile R1 și R2 (ceea ce este echivalent cu asocierea 1:N între relațiile R2 și R1) și reprezinta o constrângere între două relații, numita *constrângere referentiala*. Relația care contine cheia straina se numeste *relația care refera* (R1 în definiția de mai sus), iar relația care contine cheia candidata se numeste *relația referita* (R2 în definiția 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 relația referita, sau cheia straina are valoarea NULL (daca atributele acesteia nu sunt supuse constrângerii NOT NULL).

Majoritatea sistemelor SGBD implementeaza constrângerea de mentinere a integritatii

referentiale 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 înregistrărilor (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).

## 2. INSTALAREA SI PORNIREA SISTEMULUI DE GESTIUNE A BAZELOR DE DATE

Dintre sistemele de gestiune a bazelor de date relationale existente la ora actuala vom studia Microsoft SQL Server, in laborator fiind instalata varianta Microsoft SQL Server 2005 Express Edition.

### 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](http://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 2005 (*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 2005 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 2005->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*).

#### 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 atribuțiile administratorului (contul *sa*).

În panoul din stânga al ferestrei *Enterprise Manager*, elementul *Console Root* reprezintă rădăcina unui arbore care conține în primul nivel toate instalările SQL Server disponibile în rețeaua respectivă, organizate în mai multe grupuri de servere. Inițial în grupul implicit *SQL Server Group* nu există nici un server înregistrat și înregistrarea se face cu comanda *New SQL Server Registration* selectată din meniul de context care se deschide la apăsarea butonului dreapta al mouse-ului atunci când este selectat directorul *SQL Server Group*.

După înregistrare, se poate rula *Enterprise Manager* cu conectare pe contul *sa* în mod de autentificare Windows. În acest mod este bine să fie stabilită (sau modificată) parola contului *sa* și să fie editată înregistrarea serverului cu comanda *Edit SQL Server Registration Properties* din meniul de context care se deschide la apăsarea butonului dreapta al mouse-ului atunci când este selectat numele instanței serverului (în laborator, *Server125a*). În fereastra care se deschide (cu titlul *Registered SQL Server Properties*) se selectează opțiunile *Use SQL Server Authentication* și *Always prompt for login name and password*.

Din contul *sa* se creează conturile de conectare *login* cu comanda *New Login* din meniul de context care se deschide la apăsarea butonului dreapta al mouse-ului atunci 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 setează în pagina *Server Roles* a ferestrei care se deschide la crearea unui nou cont) și dreptul *public* asupra bazei de date *Northwind* (care se setează în pagina *Database Access*).

Pe orice stație pe care este instalat un client SQL, se poate lansa programul *Enterprise Manager*, și prima operație care trebuie să fie făcută este înregistrarea serverului dorit din rețea. În cursul înregistrării (cu comanda *New SQL Server Registration* din meniul de context) se aleg opțiunile *SQL Server Authentication* și *Always prompt for user name and password*. După aceasta se poate face conectarea pe contul propriu (*login*) și parola acestuia, cont care trebuie să fi fost creat mai înainte pe server.

Fereastra principală a programului *SQL Server Enterprise Manager* conține un meniu, o bară de instrumente și două panouri. În panoul din stânga se afișează numele serverului, împreună cu șapte subdirectoare ale acestuia, iar în panoul din dreapta al ferestrei sunt prezentate informații privind directorul selectat. Dintre directoarele afișate, cele mai utile sunt directorul *Databases*, care conține bazele de date gestionate de serverul respectiv, și directorul *Security*, care permite administrarea utilizatorilor și a drepturilor de acces la bazele de date.

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

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

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

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

Lista tuturor conturilor de conectare existente (*logins*) se afișează în panoul din dreapta atunci când se selectează opțiunea *Logins* în arborele de selecție și pentru fiecare cont se poate selecta o opțiune din cele existente în meniul contextual. La opțiunea *Properties* se pot inspecta și modifica diferite proprietăți ale contului respectiv (parola, bază de date implicită 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 operație în sistemul SQL Server, *Security Administrators*, care permite administrarea utilizatorilor, *Database*

*Creators*, care permite crearea noii 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 creeaza 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 baze de date corespunzator unui cont de conectare cu rol de *System Administrators*.

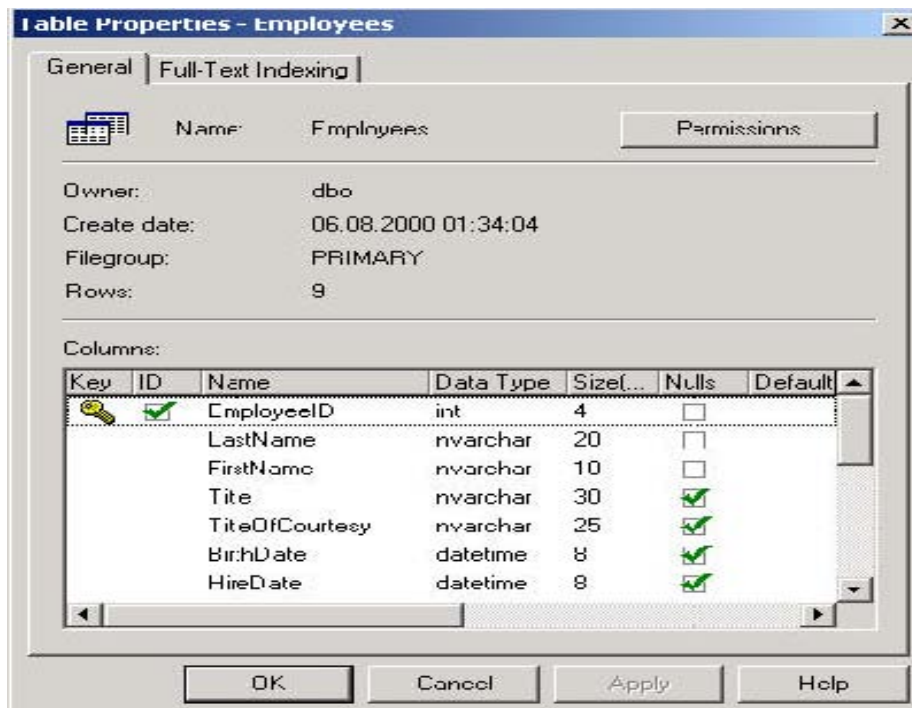
Pentru lucrul în siguranta cu SQL server se recomanda crearea unor conturi care sa aiba toate drepturile asupra bazei lor de date proprii, si numai drepturi de citire (dreptul *public*) asupra bazelor de date preinstalate.

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 sa deschida fereastra de proprietati ale contului propriu (cu comanda *Properties* din meniul de context) si poate sa modifice baza de date implicita. De asemenea, se pot vedea tabelele, functiile 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 si stergere tabele, creare proceduri, etc.). Dupa crearea bazei de date proprii, este recomandabil ca sa se seteze aceasta baza de date ca baza de date implicita 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 modifica elementele unui director (*Databases*, *Logins*, etc), ele vor fi actualizate în panoul din *Enterprise Manager* numai daca se da comanda *Refresh* (din meniul de context corespunzator).

Pentru acomodarea cu sistemul SQL Server se selecteaza baza de date preinstalata *Northwind* si se inspecteaza tabelul *Employees* (prin comanda dublu clic atunci când este selectat tabelul respectiv sau prin selectarea optiunii *Properties* din meniul de context). La aceasta comanda se deschide o fereastra în care pentru fiecare atribut al relatiei (tabelului) sunt afisate numele, tipul de date, valoarea implicita, marcajul de cheie primara, etc. (Fig. 1.1).



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.

### 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
```

Dacă nu se specifică numele bazei de date, atunci are loc conectarea la baza de date implicată a contului respectiv. După conectare, programul afișează ca prompt de comandă numărul liniei de comandă (în ordine crescătoare începând cu 1) și se pot transmite spre execuție loturi de execuție Transact SQL (succesiune de instrucțiuni terminate cu comanda GO) care pot accesa baza de date respectivă. Baza de date curentă se poate schimba cu comanda: USE noua\_baza\_date. Execuția unui fișier script care conține unul sau mai multe loturi Transact SQL se realizează cu comanda:

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

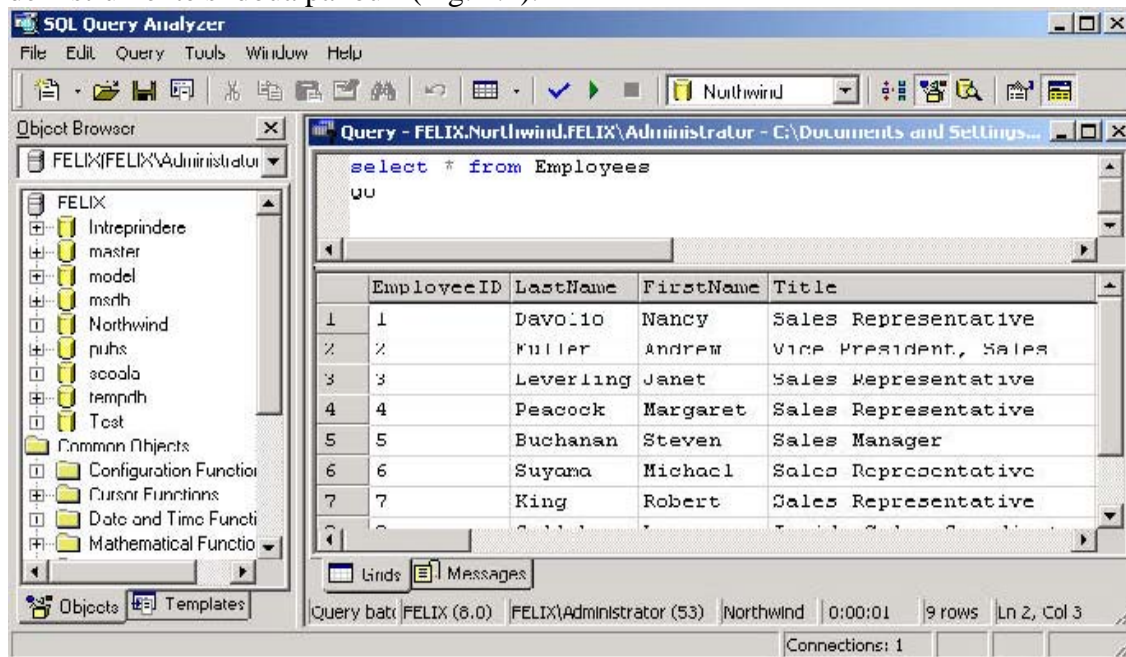
### 2.1.3 Utilizarea programului *SQL Server Query Analyzer*

Programul utilitar *SQL Server Query Analyzer* este un instrument grafic care permite executarea mai multor activități de proiectare a bazelor de date și de execuție a aplicațiilor, cum sunt: crearea și executarea interogărilor, testarea procedurilor stocate, operații de introducere, ștergere 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 să se specifice serverul, contul și parola de conectare. În fereastra programului Query Analyzer sunt afișate o bară de meniu, o bară



de instrumente si doua panouri (Fig. 1.2).



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 fereastră 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 fereastră *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 fereastră *Query* se realizeaza cu comanda *Execute*, care se poate lansa fie din meniul de context al ferestrei, fie din meniul *Query*.

### 3. EXERCITII SI PROBLEME

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 proiectede 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 (*Student*) fara parola. Lucrarile si exercitiile propuse se pot efectua si 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 rețeaua 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 *Microsoft SQL Server 2005 Management Studio Express CTP* 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.Student*) asignat contului propriu de conectare.

**1.5** Lansati programul *Microsoft SQL Server 2005 Management Studio Express CTP* si introduceti comenzile necesare pentru a obtine situatia din Fig. 1.2. Urmariti rezultatele afisate!