

Baze de date

- Lucrare de laborator 3 -

PROIECTAREA BAZELOR DE DATE RELATIONALE

1. NOTIUNI TEORETICE

Proiectarea unei baze de date consta din proiectarea schemei conceptuale (logice) si fizice a acesteia, astfel încât sa raspunda cerintelor utilizatorilor pentru un anumit set de aplicatii. În general, se considera ca proiectarea unei baze de date se poate diviza în urmatoarele faze:

Colectarea si analiza cerintelor.

Proiectarea conceptuala a bazei de date.

Alegerea unui SGBD.

Proiectarea logica a bazei de date.

Proiectarea fizica a bazei de date.

Cele cinci faze de proiectare enumerate mai sus nu se desfasoara strict într-o singura secventa. În multe cazuri este necesara modificarea proiectului dintr-o faza initiala într-una din fazele ulterioare, pentru a se obtine rezultatele dorite. Aceste bucle de reactie între faze (sau în interiorul unei faze) sunt, în general, frecvente în cursul proiectarii unei baze de date.

Înainte de a se proiecta efectiv o baza de date, este necesar sa se cunoasca ce rezultate se asteapta utilizatorii potentiali sa obtina de la baza de date respectiva si ce informatii primare sunt disponibile pentru aceasta. De asemenea, este necesar sa se cunoasca ce aplicatii se vor efectua (aplicatii de gestiune a stocurilor, aplicatii contabile, salarizare, etc.).

1.2 PROIECTAREA CONCEPTUALA A BAZELOR DE DATE

În faza de proiectare conceptuala a bazelor de date se proiecteaza schema conceptuala si schemele externe ale bazei de date.

Desi nu este obligatoriu, aceasta faza se poate mentine independenta de SGBD si produce un model de date de nivel înalt, care va fi implementat dupa transpunerea lui într-un model de date specific. Chiar daca proiectantii pot porni direct cu scheme conceptuale specifice unui anumit SGBD (care se mai numesc si scheme logice), este totusi recomandabil sa se realizeze mai întâi schema conceptuala de nivel înalt independenta de SGBD, deoarece aceasta este o descriere stabila si inavuabila a bazei de date. Alegerea unui SGBD si deciziile ulterioare de proiectare se pot schimba fara ca aceasta sa se schimbe.

Proiectul conceptual de nivel înalt se realizeaza pe baza cerintelor definite în prima etapa de proiectare si se reprezinta, în general printr-o diagrama Entitate-Asociere (extinsa).

Modelul Entitate-Asociere (Entity-Relationship Model) este un model conceptual de nivel înalt al unei baze de date, care defineste multimile de entitati si asocierile dintre ele, dar nu impune nici un mod specific de structurare si prelucrare a datelor.

Elementele esentiale ale modelului Entitate-Asociere sunt *entitatile (entities)* si *asocierile* dintre acestea (*relationships*).

O entitate (*entity*) este "orice poate fi identificat în mod distinctiv"; o entitate se refera la un aspect al realitatii obiective care poate fi deosebit de restul universului si poate reprezenta un obiect fizic, o activitate, un concept, etc. Orice entitate este descrisa prin attributele sale. Un atribut (*attribute*) este o proprietate care descrie un anumit aspect al unei entitati.

Toate entitatile similare, care pot fi descrise prin aceleasi attribute, apartin unui acelasi tip de entitate (*entity type*), iar colectia tuturor entitatilor de acelasi tip dintr-o baza de date constituie o multime de entitati (*entities set*). În general, în modelul E-A se foloseste aceeaasi denumire atât

pentru un *tip de entitate* cât și pentru *multimea entitatilor* de acel tip.

De exemplu, tipul de entitate “angajat” (al unei institutii) reprezintă orice persoană angajată a institutiei, care are o anumită funcție și primește un anumit salariu. Acest tip de entitate poate fi descris prin mai multe atribute, dintre care o parte sunt atribute de identificare a persoanei (Nume, Prenume, Data Nasterii, Adresa), iar altele sunt atribute legate de activitatea acesteia în institutia respectiva (Funcție, Salariu).

În proiectarea bazelor de date se considera două categorii de entități: entități normale (puternice, obișnuite -*regular entities*) și entități slabe (dependente -*weak entities*).

Entitățile normale au o existență proprie în cadrul modelului, în timp ce entitățile slabe nu pot exista decât dacă există o entitate normală (puternică) cu care sunt asociate. De exemplu, o entitate “dependent” poate să reprezinte o persoană care depinde de un angajat al unei institutii (adică se află în întreținerea acestuia). O entitate “angajat” este o entitate puternică, deoarece ea există în mod normal în modelul activității institutiei, în timp ce o entitate “dependent” este o entitate slabă: nu se va înregistra o astfel de persoană decât dacă părintele (sustinatorul) acesteia este angajat în acea institutie.

O *asociere (relationship)* este o corespondență între entități din două sau mai multe mulțimi de entități. Gradul unei asocieri este dat de numărul de mulțimi de entități asociate. Asocierile pot fi binare (de gradul 2, între 2 mulțimi de entități) sau multiple (între k mulțimi de entități, $k > 2$).

Asocierile binare sunt, la rândul lor, de trei categorii, după numărul elementelor din fiecare dintre cele două mulțimi puse în corespondență de asocierea respectivă. Fiind date două mulțimi de entități, E_1 și E_2 , se definesc următoarele categorii de asocieri binare:

- *Asocierea “unul-la-unul” (one-to-one)* este asocierea prin care unui element (entitate) din mulțimea E_1 îi corespunde un singur element din mulțimea E_2 , și reciproc; se notează cu 1:1.

Asocierea “unul-la-multe” (one-to-many) este asocierea prin care unui element din mulțimea E_1 îi corespund unul sau mai multe elemente din mulțimea E_2 , dar unui element din E_2 îi corespunde un singur element în mulțimea E_1 ; se notează cu 1:N.

Asocierea “multe-la-multe” (many-to-many) este asocierea prin care unui element din mulțimea E_1 îi corespund unul sau mai multe elemente din mulțimea E_2 , și reciproc; se notează cu M:N.

O asociere între două sau mai multe mulțimi de entități este, în același timp, o asociere între tipurile de entități corespunzătoare.

Diagrama Entitate-Asociere (*Entity-Relationship Diagram*) reprezintă modelul Entitate-Asociere prin mulțimile de entități și asocierile dintre acestea.

Există numeroase variante de notații pentru redarea diagramei E-A. Una dintre cele mai folosite notații reprezintă un tip de entitate (precum și mulțimea de entități de acel tip) printr-un dreptunghi, iar atributele tipului de entitate prin elipse conectate printr-o linie continuă la acesta (Fig. 2.1). Pentru entitățile puternice se utilizează un dreptunghi încadrat cu o linie simplă, iar pentru entitățile slabe se utilizează un dreptunghi încadrat cu linie dublă.

O asociere (*tip de asociere*) dintre două sau mai multe tipuri de entități se reprezintă printr-un romb conectat prin link-uri (linii continue, formate din unul sau mai multe segmente) la tipurile de entități asociate. O asociere poate să aibă sau nu un nume; dacă are un nume, acesta poate fi înscris în rombul respectiv sau în vecinătatea acestuia. Categoria asocierii se notează prin înscrierea multiplicității pe fiecare link care conduce la un tip de entitate. Este posibil ca o asociere să prezinte ea însăși atribute, și aceste atribute se reprezintă prin elipse conectate la asocierea respectivă.

Modelul Entitate -Asociere Extins (*Enhanced Entity-Relationship Model*) permite definirea de subtipuri ale unui tip de entități, care moștenesc atribute de la tipul de entitate pe care îl extind (și care, în acest context, se numește supertip) și au în plus atribute specifice semnificației lor. Prin

definirea tipurilor si a subtipurilor de entitati se pot crea ierarhii de tipuri de entitati pe mai multe niveluri. Modelul *Entitate-Asociere Extins* se reprezinta printr-o diagrama E-A extinsa, în care legatura între un supertip de entitati si subtipurile acestuia se reprezinta printr-o linie pe care se plaseaza un semicerc îndreptat catre supertip (Fig. 2.1).

Exemplu de model Entitate-Asociere. Se considera o baza de date a unei întreprinderi. Tipurile de entitati puternice (normale) care se pot defini pentru modelarea activitatii unei întreprinderi pot fi: SECTII, ANGAJATI, FURNIZORI, CLIENTI, PRODUSE, COMPONENTE (Fig. 2.1):

SECTII(Nume,Buget)

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

FURNIZORI(Nume,Prenume,Adresa)

CLIENTI(Nume,Prenume,Adresa)

PRODUSE(Denumire,Descriere)

COMPONENTE(Denumire,Descriere)

La aceste multimi de entitati se adauga multimea de entitati slabe:

DEPENDENTI(Nume,Prenume,DataNasterii,GradRudenie)

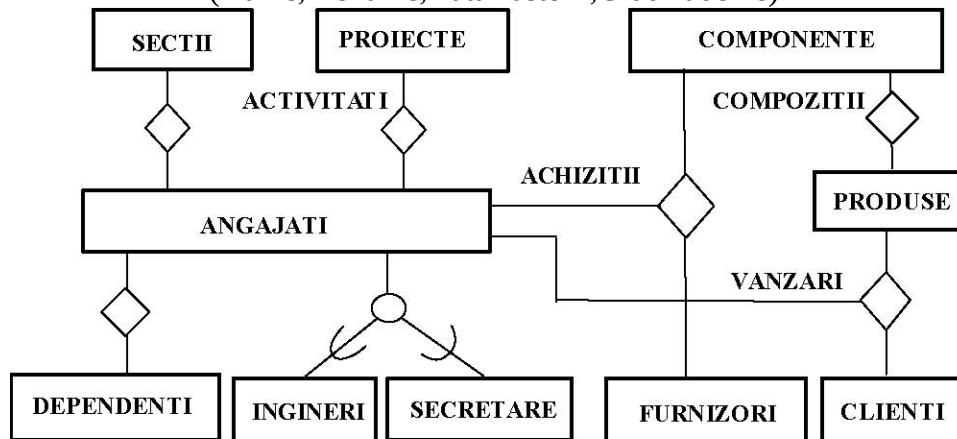


Fig. 2.1 Diagrama E-A a bazei de date a unei întreprinderi.

Pentru tipul ANGAJATI se defineste o specializare disjuncta partiala, cu subtipurile INGINERI si SECRETARE. Aceste subtipuri se afla în asociere 1:1 cu tipul de baza ANGAJATI, mostenesc attributele acestuia si fiecare mai contine attribute specifice:

INGINERI(Specialitatea)

SECRETARE(VitezaRedactare)

Asocierile dintre multimele de entitati se stabilesc în functie de modul în care se desfasoara activitatea modelata. De exemplu, daca în întreprinderea respectiva fiecare angajat lucreaza într-o singura sectie, atunci între multimele de entitati SECTII-ANGAJATI exista o asociere 1:N.

O multime de entitati slabe se afla, de regula, în asociere N:1 cu multimea de entitati puternice de care depinde. In exemplul dat, între multimea DEPENDENTI si multimea de entitati ANGAJATI exista o asociere N:1. O multime de entitati de un subtip dat este, de regula, în asociere 1:1 cu multimea de entitati de supertipul acesteia. Pentru exemplul de mai sus, un angajat poate fi un (singur) inginer; iar un inginer este chiar un angajat, deci asocierea între multimele de entitati ANGAJATI si INGINERI exista o asociere cu raportul de cardinalitate 1:1.

1.2 PROIECTAREA LOGICA A BAZELOR DE DATE

În faza de proiectare logica a unei baze de date se realizeaza schema conceptuala globala si schemele conceptuale (vederile) externe pentru sistemul SGBD ales, pornind de la schema conceptuala si schemele externe de nivel înalt independente de SGBD, proiectate în faza

precedenta. Aceasta faza de proiectare logica poate fi realizata în doua sub-faze: transpunerea schemei conceptuale în modelul de date al sistemului SGBD ales, dar independent de sistemul de gestiune propriu-zis si rafinarea schemei conceptuale si a schemelor externe obtinute anterior, astfel încât sa se utilizeze unele (sau cât mai multe) din facilitatile oferite de sistemul SGBD ales (modul de generare a cheilor primare, definirea constrângerilor, etc.).

Aceste doua sub-faze se pot realiza împreuna, folosind unul din instrumentele de proiectare oferite de sistemul SGBD ales. Rezultatul acestei faze de proiectare îl constituie, asadar, schema conceptuala si schemele externe ale bazei de date, dependente de sistemul SGBD ales si de modelul de date al acestuia. Pentru transpunerea modelului Entitate-Asociere (reprezentat prin diagrama E-A) în model relational se parcurg în principal doua etape: proiectarea relatiilor si proiectarea asociierilor.

Proiectarea relatiilor.

În Fig. 2.2 este data schema conceptuala a bazei de date relationale corespunzatoare diagramei E-A din Fig. 2.1, dezvoltata în SQL Server. În SQL Server relatiile si asocierile între ele sunt reprezentate vizual în diagrama de asocieri (*Relationships*), care este corespondentul relational al diagramei E-A.

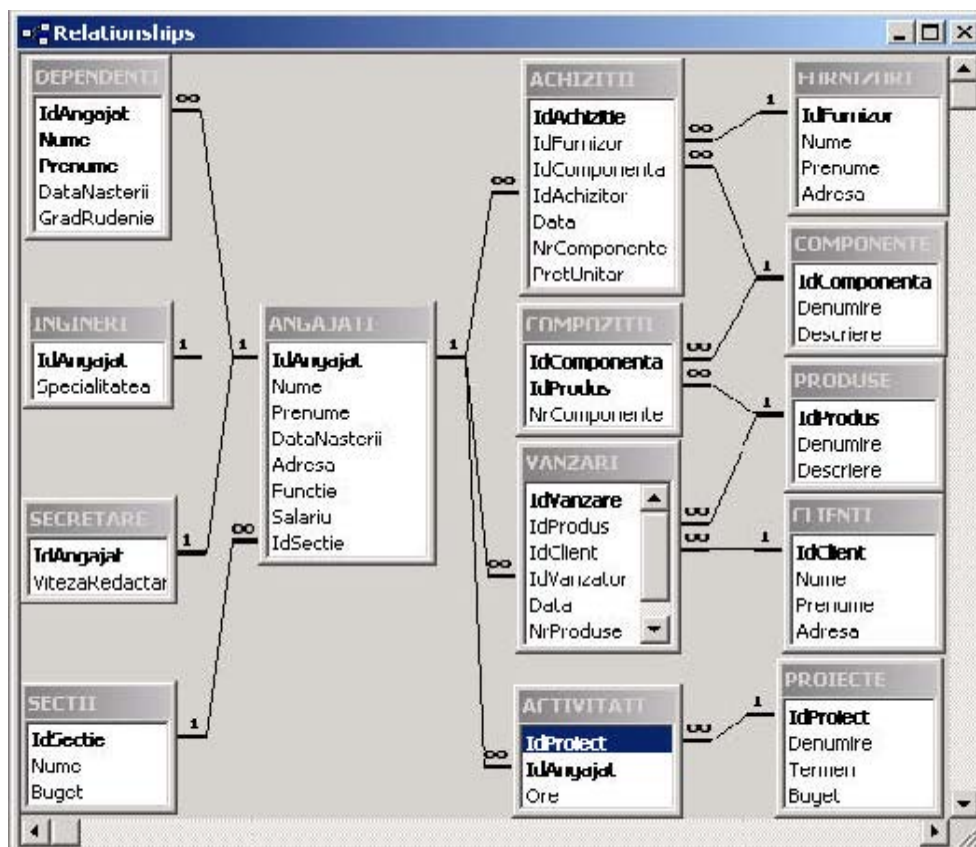
Multimile de entitati puternice (normale) din diagrama E-A devin relatii, cu attributele date de attributele entitatilor. În astfel de relatii cheia primara se defineste fie ca o cheie naturala (combinatie de attribute care definesc în mod unic un tuplu al relatiei), fie ca o cheie primara artificiala. În exemplul prezentat, în în fiecare din relatiile care corespund multimilor de entitati puternice s-a adaugat câte o cheie primara artificiala (IdAngajat, IdSectie, IdProiect, etc.).

Multimile de entitati slabe din diagrama E-A devin, de regula, relatii aflate în asociere N:1 cu relatia corespunzatoare multimii de entitati de care acestea depind. Pentru realizarea acestei asocieri, în relatia dependenta se adauga o cheie straina care refera cheia primara a relatiei referite (puternice).

Cheia primara a unei relatii dependente poate fi o combinatie formata din atributul cheie straina si alte attribute care asigura posibilitatea de identificare unica a unui tuplu, sau poate fi o cheie artificiala. Cheia primara a relatiei DEPENDENTI este compusa din atributul cheie straina (IdAngajat, care refera cheia primara a relatiei ANGAJATI) si attributele Nume si Prenume (ale persoanei dependente).

Multimile de entitati care sunt subtipuri ale unui tip de entitate dat devin relatii aflate în asociere 1:1 cu relatia corespunzatoare multimii de entitati de tipul respectiv (supertip). Pentru realizarea acestei asocieri, în relatia corespunzatoare subtipului de entitati se defineste o cheie straina care refera cheia primara din relatia corespunzatoare supertipului de entitati; aceasta cheie straina este în acelasi timp si cheie primara în relatia corespunzatoare subtipului de entitati.

În exemplul prezentat, asocierile ANGAJATI-INGINERI si ANGAJATI-SECRETARE sunt asocieri 1:1. În relatia INGINERI atributul IdAngajat este cheie straina care refera cheia primara cu acelasi nume din relatia ANGAJATI si este în acelasi timp si cheie primara; la fel, în relatia SECRETARE atributul IdAngajat este cheie straina care refera cheia primara cu acelasi nume din relatia ANGAJATI si este în acelasi timp si cheie primara.



Proiectarea asocierilor.

Asocierea binară N:1 dintre două mulțimi de entități puternice din diagrama E-A se realizează în modelul relational prin intermediul unei chei straine în prima relație (cea cu multiplicitatea N a asocierii) care referă cheia primară (sau o cheie candidată) din relația referită (cea cu multiplicitatea 1 a asocierii). De exemplu, asocierea N:1 între relațiile ANGAJATISECTII se realizează prin cheia straina IdSectie adăugată relației ANGAJATI, care referă cheia primară cu același nume a relației SECTII.

Asocierea binară M:N dintre două mulțimi de entități din diagrama E-A se realizează în modelul relational prin intermediul unei noi relații, numită relație de asociere. Această nouă relație se află în asociere M:1, respectiv N:1 cu fiecare din cele două relații date prin intermediul a două chei straine care referă cheile primare (sau chei candidate) din relațiile date.

De exemplu, pentru a reprezenta asocierea M:N dintre relațiile COMPONENTE-PRODUSE se adăuga o nouă relație numită COMPOZITII, care conține cheile straine IdComponenta și IdProdus, care referă cheile primare cu același nume din relațiile COMPONENTE, respectiv PRODUSE. Cheia primară a unei relații de asociere poate fi o cheie artificială sau poate fi compusă din cheile straine care referă cele două relații asociate, eventual împreună cu alte atribute ale relației, care caracterizează asocierea respectivă. Așa cum se poate vedea în Fig. 2.2, cheia primară a relației COMPOZITII este formată din cele două chei straine pe care le conține.

Asocierea binară 1:1 între două mulțimi de entități puternice se poate transpune în modelul relational în două moduri: fie prin intermediul unei chei straine (ca un caz particular al unei asocieri N:1), fie printr-o relație de asociere (ca un caz particular al unei asocieri M:N).

Asocierea binară 1:1 dintre o mulțime de entități de un subtip și mulțimea de entități supertip din diagrama Entitate-Asociere este tot o asociere binară cu raportul de cardinalitate 1:1 între relațiile corespunzătoare în modelul relational. Acest tip de asociere se realizează prin definirea în relația corespunzătoare subtipului de entități a unei chei straine care referă cheia primară din relația corespunzătoare supertipului de entități; această cheie straina este în același timp și cheia primară în relația corespunzătoare subtipului de entități. Acest mod de definire a fost

folosit pentru asocierea dintre tabelele INGINERI, SECRETAREsi tabelul ANGAJATI.

Asocierea multipla M:N:P:.... dintre mai mult de doua multimi de entitati din diagrama E-A se realizeaza în mod asemanator cu asocierea binara, prin intermediul unei noi relatii care se afla în asociere M:1, N:1, P:1, etc, cu fiecare din relatiile date. Aceasta asociere este realizata prin intermediul mai multor chei straine, fiecare cheie straina referind cheia primara (sau o cheie candidata) dintr-una din relatiile date. De exemplu, relatia ACHIZITII realizeaza asocierea între relatiile COMPONENTE, FURNIZORI, ANGAJATI. Ea contine trei chei straine (IdComponenta, IdFurnizor, IdAchizitor) care refera relatiile COMPONENTE, FURNIZORI, respectiv ANGAJATI, iar cheia primara este cheia artificiala IdAchizitie.

1.3 PROIECTAREA FIZICA A BAZELOR DE DATE

Proiectarea fizica a bazei de date este procesul de alegere a structurilor de memorare si de acces la fisierele bazei de date, pentru a obtine performante cât mai bune, pentru cât mai multe din aplicatiile proiectate. Ca parametri generali de alegere a optiunilor proiectului fizic al unei baze de date relationale se pot enumera: timpul de raspuns, utilizarea spatiului de memorare, capacitatea tranzactionala. Deciziile de proiectare fizica se pot lua numai dupa o analiza a aplicatiilor care se vor executa si în principal a interogarilor si tranzactiilor pe care acestea le vor lansa. În urma analizei se pot sintetiza informatii care sa dea imaginea de ansamblu a utilizarii atributelor relatiilor bazei de date: care attribute sunt actualizate cel mai frecvent, care attribute sunt folosite cel mai frecvent în selectii ale interogarilor, etc. Aceste informatii se folosesc pentru stabilirea indexurilor secundare ale relatiilor.

2. APLICATII PRACTICE

Sistemul SQL Server poate executa programe în limbajul SQL sau în limbajul Transact-SQL, care este extensia procedurala a limbajului SQL. Limbajul Transact-SQL contine instructiuni SQL (asemanatoare celor specificate în standardul SQL2) precum si instructiuni de control al executiei (care vor fi prezentate în Capitolul 4). Instructiunile Transact-SQL se transmit sistemului grupate în loturi de executie (*batches*), prin intermediul programelor de aplicatii sau al programelor utilitare *osql* sau *Query Analyzer*.

Proiectarea tabelor se poate face atât vizual, folosind generatoarele de cod (*Wizards*) ale programului utilitar *SQL Server Enterprise Manager*, cât si prin comenzi (grupate în scripturi) care contin loturi de executie Transact-SQL, executate de la consola (folosind utilitarul *osql*) sau din programul *Query Analyzer*.

Proiectarea vizuala a tabelor se poate face în programul *SQL Server Enterprise Manager*. La comanda *New Table*, care se actioneaza din meniul contextual care se deschide la apasarea butonului dreapta al mouse-ului atunci când este selectat directorul *Tables* al bazei de date proprii, se deschide o fereasta de proiectare foarte asemanatoare cu cea din MS Access, cu unele diferente (care se pot observa în Fig. 2.3 de mai jos). La definirea cheii primare se poate specifica proprietatea de autoincrementare cu optiunea *IDENTITY*, care este echivalenta cu optiunea *AutoNumber* din MS Access, dar, în plus, aceasta optiune permite specificarea valorii de început a secventei de numere crescatoare (*Identity Seed*), si valoarea de incrementare (*Identity Increment*) care este implicit 1.

Asocierea între relatii se stabileste prin comanda *New Database Diagram* din meniul de context care se deschide la apasarea butonului dreapta al mouse-ului atunci când este selectat subdirectorul *Diagrams* al bazei de date. La aceasta comanda se deschide o fereasta de proiectare a asocierilor asemanatoare cu cea din MS Access (*Relationships*), prin care se stabilesc cheile straine si modul de referire între relatii.

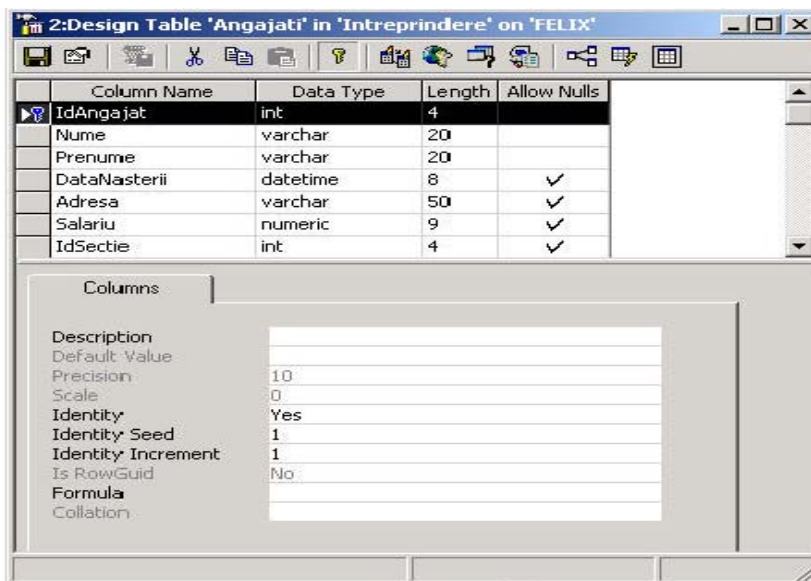


Fig. 2.3 Fereastra de proiectare a unui tabel folosind SQL Server Enterprise Manager.

Scripturile de comenzi în SQL Server contin loturi de executie Transact-SQL si pot fi executate prin intermediul unor programe utilitare, cum sunt *osq* sau *Query Analyzer*. Pentru detalii privind folosirea acestor programe este necesar, bineînțeles, sa fie studiata documentatia oferita de furnizor (*Books Online*).

Ca exemplu se prezinta scriptul de creare a tabelelor SECTII, ANGAJATI în baza de date proprie din sistemul SQL Server (fisier *creare_tabele_sqlserver.sql*).

```
Script de creare a tabelelor ANGAJATI si SECTII în SQL Server
DROP TABLE ANGAJATI
GO
CREATE TABLE ANGAJATI ( IdAngajat int PRIMARY KEY IDENTITY, Nume varchar(20)
NOT NULL, Prenume varchar(20) NOT NULL, DataNasterii datetime, Adresa varchar(50),
Salariu decimal DEFAULT 2800, IdSectie int)
GO
DROP TABLE SECTII
GO
CREATE TABLE SECTII ( IdSectie int PRIMARY KEY IDENTITY, Nume varchar(50) NOT
NULL, Buget decimal)
GO
ALTER TABLE ANGAJATI ADD CONSTRAINT FK_ANGAJATI FOREIGN KEY (IdSectie)
REFERENCES SECTII(IdSectie)
GO
```

La inserarea liniilor în tabele (cu instructiuni INSERT), nu se admite precizarea valorii pentru un atribut cheie primara de tip IDENTITY. Daca se executa fisierul *introducere_sqlserver.sql* pentru introducerea unor linii în tabelele SECTII si ANGAJATI, se va observa ca liniile în care se specifica valoarea cheii primare nu sunt admise (produc eroare de executie). De asemenea, sistemul efectueaza verificarea cheilor straine si nu admite tupluri care contin o cheie straina a carui valoare nu se regaseste în nici o valoare a cheii primare referite (Fig. 2. 4).

Cheile primare au fost definite de tipul IDENTITY, care este un atribut cu autoincrementare, iar cheia straina din tabelul ANGAJATI s-a definit printr-o instructiune ALTER TABLE. Aceasta modalitate este frecvent folosita, deoarece permite crearea tabelelor în orice ordine si adaugarea dupa aceea a cheilor straine. Daca s-ar fi dorit definirea cheii straine chiar în instructiunea CREATE TABLE ANGAJATI, atunci relatia referita (în acest caz relatia SECTII) ar

fi trebuit sa fie definit înaintea relatiei care refera (în acest caz relatia ANGAJATI). La proiectarea unei baze de date mari, cu multe relatii si referiri între ele, este destul de dificil de stabilit ordinea completa de referiri între tabele si de aceea se prefera definirea separata a cheilor straine, prin instructiuni ALTER TABLE.

În pagina de afisare Messages se gasesc mesajele de eroare returnate de sistemul de gestiune. La executia instructiunilor din fereastra de interogari din Fig. 2.4 se obtin urmatoarele mesaje :

Cannot insert explicit value for identity column in table 'ANGAJATI' when IDENTITY_INSERT is set to OFF.

INSERT statement conflicted with COLUMN FOREIGN KEY constraint 'FK_ANGAJATI'. The conflict occurred in database 'Intreprindere', table 'SECTII', column 'IdSectie'.

Primul mesaj se refera la faptul ca nu se admit valori explicite pentru o cheie cu proprietatea IDENTITY decât daca se seteaza la ON proprietatea IDENTITY_INSERT. Instructiunea Transact-SQL care seteaza aceasta proprietate este:

SET IDENTITY_INSERT [database.[owner.]]{table}{ON|OFF}

Cel de-al doilea mesaj se refera la faptul ca nu se admit tupluri care nu respecta integritatea referentiala (egalitatea valorii cheii straine cu o valoare a cheii primare din relatia referita).

```
-- Introducere linii in tabelul SECTII
INSERT INTO SECTII (Nume, Buget) VALUES ('Productie', 400000);

-- Introducere linii in tabelul ANGAJATI
INSERT INTO ANGAJATI (Nume, Prenume, IdSectie) VALUES ('Ionescu', 'Ion', 1);
INSERT INTO ANGAJATI (Nume, Prenume, IdSectie) VALUES ('Popescu', 'Petre', 1);
INSERT INTO ANGAJATI (IdAngajat, Nume, Prenume) VALUES (8, 'NumeA', 'PrenumeA');
INSERT INTO ANGAJATI (Nume, Prenume) VALUES ('NumeB', 'PrenumeB');
INSERT INTO ANGAJATI (Nume, Prenume, IdSectie) VALUES ('Vasilescu', 'Ana', 2);

SELECT * FROM SECTII;
SELECT * FROM ANGAJATI;
```

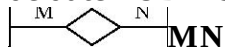
	IdSectie	Nume	Buget
1	1	Productie	400000

	IdAngajat	Nume	Prenume	DataNasterii	Adresa	Salariu	IdSectie
1	1	Ionescu	Ion	NULL	NULL	2800	1
2	2	Popescu	Petre	NULL	NULL	2800	1
3	3	NumeB	PrenumeB	NULL	NULL	2800	NULL

Query batch completed with errors. FELIX (8.0) sa (52) Intreprindere 0:00:00 4 rows Ln 1, Col 1

3. EXERCITII SI PROBLEME

1.* Transpuneti în model relational urmatoarea diagrama Entitate-Asociere (pentru o baza de date PUBLICATII). Definiti relatiile si asocierile dintre acestea în SQL Server.

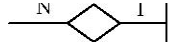


N

1

AUTORI

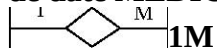
CARTI



DOMENII

În relatia de asociere AUTORI-CARTI prevedeti un atribut care sa memoreze ordinea autorilor unei carti (primul autor, al doilea autor, etc.).

2.* Transpuneti în model relational urmatoarea diagrama Entitate-Asociere (pentru o baza de date MEDICALA). Definiti relatiile si asocierile dintre acestea în SQL Server.

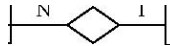


N

1

MEDICI

CONSULTATII



PACIENTI