

## Cursul 5

# Modelul relational - Partea 1 -

F. Rădulescu, Curs: Baze de date I

1

## Sumar

1. Modele de date. Modelul relational
2. Elementele modelului
3. Transformarea EA-Relational
4. Algebra relationala
5. Multiseturi
6. Operatori extinsi
7. Calcul relational pe tupluri si domenii

F. Rădulescu, Curs: Baze de date I

2

## MODELE DE DATE

- ◆ O problema fundamentala a unui SGBD este modul in care datele sunt organizate in vederea stocarii si exploatarii lor de catre aplicatii. Din punct de vedere istoric, in anii '60 au existat doua modele de organizare a datelor care au fost apoi abandonate din cauza problemelor pe care le generau:
- ◆ Modelul ierarhic,
- ◆ Modelul retea.

F. Rădulescu, Curs: Baze de date I

3

## MODELUL IERARHIC

- ◆ Modelul ierarhic, folosit de IBM in sistemul IMS (care inca este unul dintre produsele furnizate de aceasta firma), in care organizarea este sub forma arborescenta: nodurile contin date si legaturi ('pointeri') catre nodurile fiu
- ◆ Vezi:  
<http://www-01.ibm.com/software/data/ims/>

F. Rădulescu, Curs: Baze de date I

4

## MODELUL RETEA

- ◆ Modelul retea. In cadrul acestuia inregistrarile sunt structurate sub forma unui graf orientat, fiecare nod putand avea mai multe inregistrari 'tata' si mai multi fii. Au existat mai multe sisteme de gestiune si limbaje de programare dezvoltate pe baza acestui model (de exemplu limbajul COBOL).

F. Rădulescu, Curs: Baze de date I

5

## DEZAVANTAJE

- ◆ Dezavantajul principal al acestor doua modele este ca accesul la o inregistrare necesita navigarea prin arbore sau graf pentru a o localiza.
- ◆ Din acest motiv apar o serie de probleme mai ales legate timpul necesar scrierii de noi programe si a detectarii anomaliiilor care pot sa apara in proiectarea bazei de date.

F. Rădulescu, Curs: Baze de date I

6

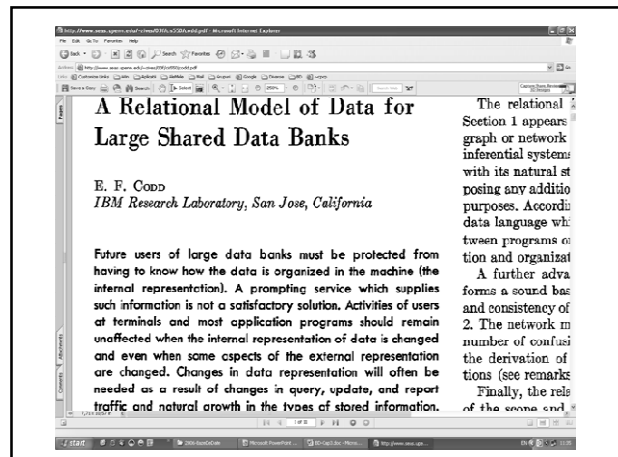
## MODELUL RELATIONAL

- ◆ Modelul relational al datelor, folosit in acest moment de majoritatea covarsitoare a sistemelor de gestiune aflate pe piata a fost introdus in anul 1970 prin articolul lui Edgar Frank Codd "**A relational model for large shared databanks**".

(<http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>)

F. Rădulescu, Curs: Baze de date I

7



## AVANTAJE (1)

- ◆ Datele sunt stocate doar ca valori; nu exista pointeri sau navigare prin date;
- ◆ Face posibila dezvoltarea de limbaje de cereri de nivel inalt in care utilizatorul specifica **ce** date doreste si nu **cum** se ajunge la rezultat, modul in care este calculat acesta fiind in sarcina sistemului de gestiune (exemplu de astfel de limbaj: SQL)

F. Rădulescu, Curs: Baze de date I

9

## AVANTAJE (2)

- ◆ Furnizeaza o baza solida pentru problemele de corectitudine a datelor (redundanta, anomalii, etc).
- ◆ Permite tratarea problemelor de independenta a datelor (discutate in capitolul 1).
- ◆ Este extensibil, putand fi utilizat si pentru modelarea si manipularea de date complexe.

F. Rădulescu, Curs: Baze de date I

10

## Sumar

1. Modele de date. Modelul relational
2. Elementele modelului
3. Transformarea EA-Relational
4. Algebra relationala
5. Multiseturi
6. Operatori extinsi
7. Calcul relational pe tupluri si domenii

F. Rădulescu, Curs: Baze de date I

11

## ELEMENTELE MODELULUI

- ◆ Domeniu
- ◆ Relatie
- ◆ Atribut
- ◆ Schema unei relatii
- ◆ Cheia unei relatii
- ◆ Valori nule
- ◆ Corectitudinea datelor

F. Rădulescu, Curs: Baze de date I

12

## DOMENIU

- ◆ **Definitie:** *Domeniu* (eng. Domain) = o multime de valori avand asociat un nume.
- ◆ Un domeniu se poate defini fie prin enumerarea elementelor sale fie prin specificarea unor caracteristici definitorii ale acestora.
- ◆ Exemple:
  - ◆ Culori = {rosu, galben, albastru, violet, verde}
  - ◆ Nota = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} sau Nota = {n ∈ N\* | n ≥ 1 si n ≤ 10}
  - ◆ Sir40 = {Multimea sirurilor de maxim 40 de caractere}
  - ◆ Numar = {Multimea numerelor intregi pozitive din intervalul [0, 100000]}

F. Rădulescu. Curs: Baze de date I

13

## Produs cartezian de domenii

- ◆ Din teoria multimilor se cunoaste notiunea de **produs cartezian** al unor multimi: fiind date **n** domenii D1, D2, ..., Dn produsul lor cartezian este:

$$D1 \times D2 \times \dots \times Dn = \{ (v1, v2, \dots, vn) \mid vi \in Di, i = 1, \dots, n \}$$

- ◆ Trebuie mentionat ca in sirul de domenii care participa la un produs cartezian unele se poate gasi in mod repetat:

$$PC = \text{Numar} \times \text{Sir40} \times \text{Numar} \times \text{Numar} \times \text{Sir40} \times \text{Sir40}$$

F. Rădulescu. Curs: Baze de date I

14

## ELEMENTELE MODELULUI

- ◆ Domeniu
- ◆ Relatie
- ◆ Atribut
- ◆ Schema unei relatii
- ◆ Cheia unei relatii
- ◆ Valori nule
- ◆ Corectitudinea datelor

F. Rădulescu. Curs: Baze de date I

15

## RELATIE (1)

- ◆ **Definitie:** *Relatie* (eng. Relation) = o submultime a unui produs cartezian avand asociat un nume.
- ◆ Termenul de relatie provide de asemenea din matematica. Un exemplu de relatie apartinand produsului cartezian PC din paragraful anterior este:

Produce = {  
 (101, 'Imprimanta laser', 30, 20, 'XY SRL', 'Str. X, București'),  
 (105, 'Calculator PC', 20, 23, 'Z SRL', 'Bd. Z, București'),  
 (124, 'Copiator', 10, 20, 'XY SRL', 'Str. X, București')  
 }

F. Rădulescu. Curs: Baze de date I

16

## RELATIE (2)

- ◆ Elementele unei relatii sunt denumite in literatura de specialitate **tupluri** (engl. tuple).
- ◆ Relatia de mai sus contine doar 3 dintre elementele produsului cartezian din care provine (3 tupluri).
- ◆ O reprezentare intuitiva pentru o relatie este o tabela, fiecare coloana avand asociat un anumit tip de date, dat de domeniul din care provine.
- ◆ Fiecare element al relatiei devine o linie a unei tabele si fiecare coloana corespunde unui domeniu din produsul cartezian de baza.

F. Rădulescu. Curs: Baze de date I

17

## REPREZENTAREA RELATIEI Produce

101	Imprimantă laser	30	20	XY SRL	Str. X, București
105	Calculator PC	20	23	Z SRL	Bd. Z, București
124	Copiator	10	20	XY SRL	Str. X, București

F. Rădulescu. Curs: Baze de date I

18

## ELEMENTELE MODELULUI

- ◆ Domeniu
- ◆ Relatie
- ◆ Atribut
- ◆ Schema unei relatii
- ◆ Cheia unei relatii
- ◆ Valori nule
- ◆ Corectitudinea datelor

F. Rădulescu, Curs: Baze de date I

19

## ATRIBUT (1)

- ◆ Deoarece o relatie are o reprezentare tabelara putem vorbi de 'coloană a unei relatii'. In mod obisnuit, intr-o tabela coloanele au un nume.
- ◆ **Definitie: *Atribut*** (eng. Attribute) = coloană a unei relatii avand asociat un nume.

F. Rădulescu, Curs: Baze de date I

20

## ATRIBUT (2)

- ◆ Pentru relatia Produse putem fixa de exemplu urmatoarele nume de atribute:
  - ◆ IdP – Codul produsului (nu exista doua produse avand acelasi cod)
  - ◆ NumeP – numele produsului
  - ◆ Qty – Cantitate
  - ◆ IdF – Codul furnizorului (nu exista doi furnizori avand acelasi cod)
  - ◆ NumeF – Numele furnizorului
  - ◆ AdresaF – Adresa furnizorului

F. Rădulescu, Curs: Baze de date I

21

## ATRIBUT (3)

IDP	NUMEP	QTY	IDF	NUMEF	ADRESAF
101	Imprimantă laser	30	20	XY SRL	Str. X, București
105	Calculator PC	20	23	Z SRL	Bd. Z, București
124	Copiator	10	20	XY SRL	Str. X, București

F. Rădulescu, Curs: Baze de date I

22

## ELEMENTELE MODELULUI

- ◆ Domeniu
- ◆ Relatie
- ◆ Atribut
- ◆ Schema unei relatii
- ◆ Cheia unei relatii
- ◆ Valori nule
- ◆ Corectitudinea datelor

F. Rădulescu, Curs: Baze de date I

23

## SCHEMA RELATIEI (1)

- ◆ Continutul unei relatii (vazuta ca o tabela) poate varia in timp: se pot adauga sau sterge linii sau se pot modifica unele dintre valorile din liniile existente.
- ◆ Ceea ce ramane constanta este structura relatiei: numele relatiei, numarul si tipul atributelor sale.
- ◆ In terminologia relationala structura unei relatii este denumita si schema relatiei.

F. Rădulescu, Curs: Baze de date I

24

## SCHEMA RELATIEI (2)

- ◆ **Definitie:** *Schema unei relatii* (eng. Relation scheme) = numele relatiei urmat de lista atributelor sale si (eventual) de domeniul din care acestea provin.
- ◆ Exista mai multe modalitati prin care se poate specifica schema unei relatii. In exemplele urmatoare prezentam cateva dintre acestea cu referire la relatia Produse:

F. Rădulescu. Curs: Baze de date I

25

## Schema relatiei Produse

- ◆  $\text{Produse}(\text{IdP}, \text{NumeP}, \text{Qty}, \text{IdF}, \text{NumeF}, \text{AdresaF})$
- ◆  $\text{Produse}(\text{IdP: Numar}, \text{NumeP: Sir40}, \text{Qty: Numar}, \text{IdF: Numar}, \text{NumeF: Sir40}, \text{AdresaF: Sir40})$
- ◆  $\text{Produse} = \text{IdP}, \text{NumeP}, \text{Qty}, \text{IdF}, \text{NumeF}, \text{AdresaF}$

F. Rădulescu. Curs: Baze de date I

26

## SCHEMA RELATIEI (3)

- ◆ In cazul prezentarii unora dintre elementele de teorie a bazelor de date relationale se folosesc si notatii de forma:  
 $R = \text{ABCDE}$
- ◆ Semnificatie: schema relatiei R contine 5 attribute notate cu A, B, C, D si respectiv E.

F. Rădulescu. Curs: Baze de date I

27

## ELEMENTELE MODELULUI

- ◆ Domeniu
- ◆ Relatie
- ◆ Atribut
- ◆ Schema unei relatii
- ◆ Cheia unei relatii
- ◆ Valori nule
- ◆ Corectitudinea datelor

F. Rădulescu. Curs: Baze de date I

28

## CHEIA RELATIEI (1)

- ◆ O relatie fiind o multime (de tupluri) nu poate contine elemente (linii) duplicat – spre deosebire de exemplu de un tabel Excel unde putem avea dubluri.
- ◆ Rezulta ca tuplurile pot fi deosebite intre ele prin valorile aflate pe una sau mai multe coloane din relatie.
- ◆ **Definitie:** *Cheia unei relatii* = multime minimala de attribute ale caror valori identifica in mod unic un tuplu al relatiei respective

F. Rădulescu. Curs: Baze de date I

29

## CHEIA RELATIEI (2)

- ◆ Cheia unei relatii este o caracteristica a schemei acesteia si nu este determinata prin inspectarea valorilor aflate la un moment dat in relatie.
- ◆ In tabela Produse cele trei linii existente pot fi identificate unic de valorile de pe 3 attribute (IdP, NumeP si Qty) dar numai IdP este cheie:
- ◆ IdP identifica (prin definitie) in mod unic un produs; rezulta ca multimea de attribute {IdP} este cheie (fiind si minimala prin natura sa).

F. Rădulescu. Curs: Baze de date I

30

### CHEIA RELATIEI (3)

- ◆ Multimea de atribute {IdP, IdF} identifica de asemenea unic fiecare tuplu al relatiei dar nu este cheie nefiind minimala: prin inlaturarea lui IdF multimea ramane in continuare cheie.
- ◆ Multimea de atribute {NumeP} nu este cheie: este posibil ca in viitor in tabela Produe sa avem mai multe linii cu NumeP = 'Imprimanta laser', 'Copiator' sau 'Calculator PC'.
- ◆ Asa cum am mentionat cheia se determina din semnificatia atributelor relatiei si nu din valorile la un moment dat.

F. Rădulescu, Curs: Baze de date I

31

### CHEIA RELATIEI (4)

- ◆ Din acelasi motiv nici una dintre celelalte multimi de atribute ale relatiei Produe nu este cheie:
  - ♦ fie nu este minimala (in cazul in care il include pe IdP)
  - ♦ fie nu identifica unic tuplurile relatiei (pot exista valori duble)
- ◆ In termeni de tabele rezulta ca nu pot exista doua linii avand aceeaasi combinatie de valori pe coloanele care formeaza cheia tabeli respective (proprietate denumita in literatura de specialitate si unicitatea cheii)

F. Rădulescu, Curs: Baze de date I

32

### CHEI MULTIPLE (1)

- ◆ O relatie poate avea mai multe chei.
  - ◆ Sa ne imaginam o relatie Studenti continand date despre studentii romani ai unei facultati:  
Studenti (IdStud, NrMatricol, Nume, CNP, SerieCI, NumarCI)
- In acest caz avem mai multe chei:
- { IdStud } – pentru ca IdStud este un numar asignat de sistem fiecarei inregistrari, fara repetitii

F. Rădulescu, Curs: Baze de date I

33

### CHEI MULTIPLE (2)

- Studenti (IdStud, NrMatricol, Nume, CNP, SerieCI, NumarCI)
- ◆ { NrMatricol } – pentru ca nu pot exista doi studenti ai unei facultati cu acelasi numar matricol
  - ◆ { CNP } – pentru ca nu pot exista doi cetateni romani (deci nici doi studenti romani) cu acelasi cod numeric personal
  - ◆ { SerieCI, NumarCI } – pentru ca nu pot exista doi cetateni romani (deci nici doi studenti romani) cu aceeaasi combinatie serie/numar carte de identitate.

F. Rădulescu, Curs: Baze de date I

34

### CHEI MULTIPLE (3)

- ◆ **Observatie:** Orice relatie are *cel putin o cheie*: deoarece intr-o relatie nu pot exista doua elemente (tupluri, linii) identice, rezulta ca multimea tuturor atributelor relatiei este cheie sau contine cel putin o cheie.
- ◆ In literatura de specialitate si in sistemele de gestiune a bazelor de date exista trei alte concepte continand cuvantul *cheie* si care vor fi prezentate in paragrafele urmatoare ale acestui capitol:
  - ♦ Cheie primara - eng. Primary key,
  - ♦ Cheie straina (sau externa) - eng. Foreign key,
  - ♦ Supercheie - eng. Superkey.

F. Rădulescu, Curs: Baze de date I

35

### ELEMENTELE MODELULUI

- ◆ Domeniu
- ◆ Relatie
- ◆ Atribut
- ◆ Schema unei relatii
- ◆ Cheia unei relatii
- ◆ Valori nule
- ◆ Corectitudinea datelor

F. Rădulescu, Curs: Baze de date I

36

## VALORI NULE

- ◆ Uneori, unele valori componente ale unei relatii (celule ale tabelii) nu au nici o valoare concreta. Se spune ca in acel loc exista o **valoare nula**.
- ◆ **Definitie: Valoare nula** (eng. Null value) = o valoare diferita de oricare alta si care modeleaza o informatie necunoscuta sau o informatie inaplicabila.
- ◆ Exemplul urmatoare prezinta o relatie Studenti in care exista astfel de valori nule si care au fost simbolizate (pentru a iesi in evidenta) prin <NULL>

F. Rădulescu, Curs: Baze de date I

37

## VALORI NULE - Exemplu

IdS	NumeStud	Codfacultate	IdTutor	Medie
1001	Ionescu Ion	03	<NULL>	9,10
1002	Popescu Vasile	03	1001	<NULL>
1003	Georgescu Ion	<NULL>	1001	8,40

F. Rădulescu, Curs: Baze de date I

38

## VALORI NECUNOSCUTE

- ◆ **Modelarea unei informatii necunoscute:**  
Codul facultatii studentului Georgescu si media lui Popescu sunt nule pentru ca in momentul incarcarii cu date informatia respectiva, desi existenta in lumea reala, nu era cunoscuta celui care a incarcat datele.
- ◆ La un moment ulterior aceste valori nule vor fi inlocuite cu valori nenule care specifica informatia respectiva.

F. Rădulescu, Curs: Baze de date I

39

## VALORI INAPLICABILE

- ◆ **Modelarea unei informatii inaplicabile:**  
Sa presupunem ca unii dintre studenti sunt consiliati in activitatea lor de un student de an mai mare, numit si tutor.
- ◆ Codul tutorului unui student este inregistrat pe coloana IdTutor (de exemplu Popescu si Georgescu il au ca tutor pe studentul Ionescu avand codul 1001).
- ◆ In cazul studentului Ionescu insa valoarea lui IdTutor este nula pentru ca acest student nu are la randul sau un tutor, valoarea nula fiind cea corecta in contextul respectiv.

F. Rădulescu, Curs: Baze de date I

40

## ELEMENTELE MODELULUI

- ◆ Domeniu
- ◆ Relatie
- ◆ Atribut
- ◆ Schema unei relatii
- ◆ Cheia unei relatii
- ◆ Valori nule
- ◆ Corectitudinea datelor

F. Rădulescu, Curs: Baze de date I

41

## CORECTITUDINEA DATELOR(1)

- ◆ Schema unei relatii contine descrierea structurii acestora dar nu da informatii privind corectitudinea datelor continute in ea.
- ◆ Exemplul urmatoare prezinta o incarcare cu date incorecte pentru tabela Produse, erorile fiind urmatoarele:
- ◆ Exista doua produse diferite avand acelasi IdP (101)
- ◆ Ultimul produs din tabela nu are asignata o valoare pe coloana IdP
- ◆ Aceeasi firma are doua coduri numerice diferite (20 si 22)
- ◆ Exista doua firme diferite cu acelasi cod (20)

F. Rădulescu, Curs: Baze de date I

42

### ATRIBUT (3)

IDP	NUMEP	QTY	IDF	NUMEF	ADRESAF
101	Imprimantă laser	30	20	XY SRL	Str. X, București
101	Calculator PC	20	20	Z SRL	Bd. Z, București
	Copiator	10	22	XY SRL	Str. Y grec, București

F. Rădulescu. Curs: Baze de date I

43

### CORECTITUDINEA DATELOR(2)

- ◆ Specificarea condițiilor de corectitudine pe care trebuie să le verifice datele se face astfel:
- ◆ În cadrul teoriei bazelor de date relationale, o relație conține date corecte dacă acestea verifică setul de **dependente functionale** (sau de alt tip) atașat relației respective (cap. 4)
- ◆ În cazul sistemelor de gestiune a bazelor de date existente pe piață, acestea pun la dispoziție mecanisme de verificare numite **constrângeri de integritate**. Constrângerile de integritate se definesc fie la crearea tabelului fie ulterior și sunt de obicei de cinci tipuri, descrise în continuare. SGBD-ul va rejecta orice operație care violează vreuna dintre constrângerile definite pe tabelul respectiv.

F. Rădulescu. Curs: Baze de date I

44

### CONSTRÂNGERI

- ◆ NOT NULL
- ◆ PRIMARY KEY
- ◆ UNIQUE
- ◆ FOREIGN KEY
- ◆ CHECK

F. Rădulescu. Curs: Baze de date I

45

### NOT NULL

- ◆ Este o constrângere la nivelul unei coloane dintr-o tabelă
- ◆ Specifică faptul că pe coloana respectivă nu pot să apară valori nule.
- ◆ Ex.: În cazul tabelului Produse o astfel de constrângere se poate asocia pentru toate coloanele sau doar o parte din acestea.
- ◆ Orice încercare de a adăuga o linie care conține valori nule pe acea coloană sau de a modifica o valoare nenulă într-una nulă va fi respinsă de sistem.

F. Rădulescu. Curs: Baze de date I

46

### CONSTRÂNGERI

- ◆ NOT NULL
- ◆ PRIMARY KEY
- ◆ UNIQUE
- ◆ FOREIGN KEY
- ◆ CHECK

F. Rădulescu. Curs: Baze de date I

47

### PRIMARY KEY (1)

- ◆ O relație poate avea mai multe chei (vezi chei multiple).
- ◆ În momentul creării tabelului corespunzător relației într-un sistem de gestiune a bazelor de date, una dintre ele poate fi aleasă ca și cheie primară (principală) a tabelului respectiv.
- ◆ O tabelă nu poate avea decât o singură cheie primară, formată din una sau mai multe atribute (coloane) ale acesteia.

F. Rădulescu. Curs: Baze de date I

48



### PRIMARY KEY (2)

- ◆ SGBD-ul creaza automat structuri de cautare rapida (index) pentru cheia primara a tabelului.
- ◆ O caracteristica a cheii primare a unei tabele este (in majoritatea SGBD-urilor) cerinta ca pe coloanele componente nu pot sa apara valori nule.

F. Rădulescu, Curs: Baze de date I

49

### PRIMARY KEY (3)

- ◆ Alegerea cheii care devine cheia primara va fi facuta in concordanta cu tipul de aplicatie in care este folosita acea tabela.
- ◆ Pentru exemplul tabelii de la paragraful 3.1.5:  
Studenti (IdStud, NrMatricol, Nume, CNP, SerieCI, NumarCI)  
avand cheile { IdStud }, { NrMatricol }, { CNP } si { SerieCI, NumarCI } alegerea cheii primare se poate face astfel:

F. Rădulescu, Curs: Baze de date I

50

### PRIMARY KEY (4)

Studenti (IdStud, NrMatricol, Nume, CNP, SerieCI, NumarCI)

- ◆ In cazul in care tabela este folosita intr-o aplicatie de gestiune a datelor privind scolaritatea, se poate alege cheia primara NrMatricol, avand in vedere ca o serie de date privind rezultatele unui student sunt legate de matricola sa (informatie de legatura cu alte tabele)

F. Rădulescu, Curs: Baze de date I

51

### PRIMARY KEY (5)

Studenti (IdStud, NrMatricol, Nume, CNP, SerieCI, NumarCI)

- ◆ In cazul in care tabela este folosita intr-o aplicatie a Biroului de politie Regie, alegerea se va face probabil intre CNP si (SerieCI, NumarCI), legatura cu bazele de date de la nivelurile superioare facandu-se dupa aceste informatii.
- ◆ In ambele cazuri se poate alege cheia primara IdStud, continand numere unice generate automat de SGBD.

F. Rădulescu, Curs: Baze de date I

52

### CONSTRANGERI

- ◆ NOT NULL
- ◆ PRIMARY KEY
- ◆ UNIQUE
- ◆ FOREIGN KEY
- ◆ CHECK

F. Rădulescu, Curs: Baze de date I

53

### UNIQUE

- ◆ Prin acest tip de constrangere se modeleaza celelalte chei ale tabelii.
- ◆ Pe coloanele unei chei definite cu UNIQUE pot insa sa apara valori nule, unicitatea fiind verificata doar pentru valorile nenule de pe coloanele cheii respective.
- ◆ In exemplul anterior, daca s-a ales cheia primara IdStud, pentru celelalte trei chei se pot defini trei constrangeri de acest tip.

F. Rădulescu, Curs: Baze de date I

54

## CONSTRANGERI

- ◆ NOT NULL
- ◆ PRIMARY KEY
- ◆ UNIQUE
- ◆ FOREIGN KEY
- ◆ CHECK

F. Rădulescu, Curs: Baze de date I

55

## FOREIGN KEY – Cheie straina(1)

- ◆ Sunt cazuri in care o multime de coloane ale unei tabele contin valori care exista o cheie (primara/unica) a unei alte tabele. Sa consideram o baza de date in care exista urmatoarele doua tabele (cheile lor primare sunt cele subliniate):

Studenti(IdS, NumeStud, CodFacultate, IdTutor, Medie)

Facultati(CodFacult, NumeFacultate, Adresa)

F. Rădulescu, Curs: Baze de date I

56

## FOREIGN KEY – Cheie straina(2)

- ◆ Coloana CodFacultate din tabela Studenti nu este cheie in aceasta tabela (pot exista mai multi studenti cu aceeasi valoare pe aceasta coloana, fiind studenti ai aceleiasi facultati) dar in mod normal contine valori care pot fi doar dintre cele existente pe cheia primara CodFacult din tabela Facultati.
- ◆ O constrangere activa de acest tip (numita si *constrangere referentiala*) va avea ca efect respingerea inserarilor/modificarilor in tabela Studenti care ar face ca pe coloana CodFacultate sa apara o valoare care nu este deja in tabela Facultati.

F. Rădulescu, Curs: Baze de date I

57

## FOREIGN KEY – Cheie straina(3)

- ◆ Rezulta implicit ca in momentul incarcarii cu date este necesar sa fie completata intai tabela Facultati si apoi tabela Studenti, altfel operatia de incarcare cu date va esua din cauza violarii acestei constrangeri.
- ◆ In cazul multor SGBD-uri se poate specifica in constrangere si stergerea automata a inregistrarilor 'fii' in cazul stergerii inregistrarii 'tata': la stergerea liniei corespunzatoare unei facultati se vor sterge automat si liniile din tabela Studenti continand studentii acelei facultati.
- ◆ Constrangerile referentiale provin de obicei din transformarea asocierilor unare si binare unu-unu si multi-unu (descrise in capitolul precedent).

F. Rădulescu, Curs: Baze de date I

58

## CONSTRANGERI

- ◆ NOT NULL
- ◆ PRIMARY KEY
- ◆ UNIQUE
- ◆ FOREIGN KEY
- ◆ CHECK

F. Rădulescu, Curs: Baze de date I

59

## CHECK (1)

- ◆ Acest tip de constrangere specifica faptul ca valorile unei linii din tabela trebuie sa verifice o conditie (expresie logica).
- ◆ Exemplu: Fie tabela Studenti(IdS, NumeStud, CodFacultate, IdTutor, Medie)

F. Rădulescu, Curs: Baze de date I

60

## CHECK (1)

**Studenti(IdS, NumeStud, CodFacultate, IdTutor, Medie)**

- ◆ pe coloana Medie putem defini o constrangere de acest tip specificand ca valoarea (daca exista o valoare nenula) trebuie sa fie din intervalul [0, 10].
- ◆ Pe coloana NumeStud putem defini o verificare a lungimii numelui (ex.: Lungimea >= 3).

F. Rădulescu, Curs: Baze de date I

61

## Sumar

1. Modele de date. Modelul relational
2. Elementele modelului
3. Transformarea EA-Relational
4. Algebra relationala
5. Multiseturi
6. Operatori extinsi
7. Calcul relational pe tupluri si domenii

F. Rădulescu, Curs: Baze de date I

62

## TRANSFORMARE EA-RELATIONAL

- ◆ In procesul de transformare vom pleca de la o diagrama EA si vom obtine trei tipuri de scheme de relatie:
- ◆ a. Relatii provenite din entitati. Ele contin aceleasi informatii ca si entitatile din care au rezultat.
- ◆ b. Relatii provenite din entitati si care contin chei straine. Ele contin pe langa informatiile provenite din entitatile din care au rezultat si attribute care in alte entitati sunt identificatori. Este cazul acelor entitati care au asocieri multi-unu si partial din cele care au asocieri unu-unu cu alte entitati.

F. Rădulescu, Curs: Baze de date I

63

## TRANSFORMARE EA-RELATIONAL

- ◆ c. Relatii provenite din asocieri. Este cazul celor care apar din transformarea asociierilor binare multi-multi si a asociierilor de grad mai mare ca doi. Ele contin ca attribute reuniunea identificatorilor entitatilor asociate plus attributele proprii ale asociierilor.

F. Rădulescu, Curs: Baze de date I

64

## ENTITATI

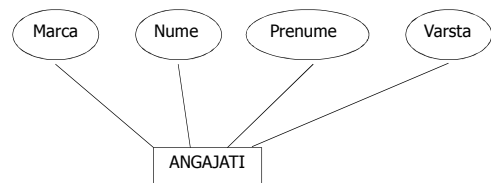
### Transformarea entitatilor

- ◆ Fiecare entitate a diagramei se transforma intr-o schema de relatie avand:
  - ◆ Numele relatiei = Numele entitatii
  - ◆ Attributele relatiei = Attributele entitatii
  - ◆ Cheia relatiei = Identificatorul entitatii

F. Rădulescu, Curs: Baze de date I

65

## EXEMPLU



Rezulta:  
Angajati(Marca, Nume, Prenume, Varsta)

F. Rădulescu, Curs: Baze de date I

66

### ASOCIERI M-1 SI 1-1

- ◆ Fiecare asociere (UNARA SAU BINARA) din aceasta categorie va avea ca rezultat adaugarea de atribute descriptive in unele dintre schemele rezultate din entitati.
- ◆ Aceste atribute care se adauga sunt chei straine: sunt cheie in alta schema de relatie.

F. Rădulescu. Curs: Baze de date I

67

### ASOCIERI M-1 SI 1-1 (cont.)

- ◆ a. In cazul asocierilor multi-unu, se adauga identificatorul entitatii unu in schema rezultata din entitatea multi
- ◆ b. In cazul asocierilor unu-unu, se adauga identificatorul unei entitati in schema rezultata din transformarea celeilalte.

F. Rădulescu. Curs: Baze de date I

68

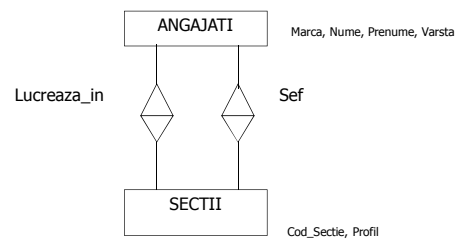
### ASOCIERI M-1 SI 1-1 (cont.)

- ◆ In cazul 1-1: Alegerea schemei in care se face adaugarea se poate face dupa doua criterii:
  - ◆ fie in acea schema care defineste relatia cu cele mai putine tupluri din cele doua,
  - ◆ fie pastrandu-se, daca exista, filiatia naturala intre cele doua entitati: identificatorul tatalui se adauga la fiu.

F. Rădulescu. Curs: Baze de date I

69

### EXEMPLU



F. Rădulescu. Curs: Baze de date I

70

### EXEMPLU (cont.)

- ◆ ANGAJATI (Marca, Nume, Prenume, Varsta, ..., Cod\_Sectie)
  - ◆ SECTII(Cod\_Sectie, Profil, ..., Marca\_Sef)
- ↓  
 Din asociere LUCREAZA\_IN  
 ↓  
 Din asociere SEF

F. Rădulescu. Curs: Baze de date I

71

### EXEMPLU (cont.)

- ◆ Pe atributul Cod\_Sectie din relatia ANGAJATI se va inregistra, pentru fiecare angajat, codul sectiei in care acesta lucreaza
- ◆ Re atributul Marca\_sef din relatia SECTII se va inregistra pentru fiecare sectie marca sefului de sectie.
- ◆ Pentru asociere SEF s-a aplicat primul criteriu (relatia SECTII va avea mult mai putine inregistrari decit ANGAJATI), dar si al doilea criteriu este indeplinit.

F. Rădulescu. Curs: Baze de date I

72

## ASOCIERI M-M

Fiecare asociere binara multi-multi si fiecare asociere cu grad mai mare ca doi se transforma intr-o schema de relatie astfel:

- ◆ Nume relatie = Nume asociere
- ◆ Atribute relatie = Reuniunea  
identificatorilor entitatilor  
asociate la care se adauga  
atributele proprii ale asocierii
- ◆ Cheia relatiei = Reuniunea  
identificatorilor entitatilor  
asociate (cf. tabel ->)

F. Rădulescu, Curs: Baze de date I

73

## ASOCIERI DE GRAD > 2

- ◆ Fiecare asociere binara multi-multi si fiecare asociere cu grad mai mare ca doi se transforma intr-o schema de relatie astfel:
- ◆ Nume relatie = Nume asociere
- ◆ Atribute relatie = Reuniunea  
identificatorilor entitatilor  
asociate la care se adauga  
atributele proprii ale asocierii
- ◆ Cheia relatiei = Conform tabel (->)

F. Rădulescu, Curs: Baze de date I

74

Grad	Conectivitate	Cheia relatiei provenite din asociere
Unare	multi (E) - multi (E)	Cheie(E) + Cheie(E)
Binare	multi (E1) - multi (E2)	Cheie(E1) + Cheie(E2)
Ternare	unu (E1) - unu (E2) - unu (E3)	Cheie(E1)+Cheie(E2) sau Cheie(E1)+Cheie(E3) sau Cheie(E2)+Cheie(E3) sau
	unu (E1) - unu (E2) - multi (E3)	Cheie(E1)+Cheie(E3) sau Cheie(E2)+Cheie(E3) sau
	unu (E1) - multi (E2) - multi (E3)	Cheie(E2)+Cheie(E3)
	multi (E1) - multi (E2) - multi (E3)	Cheie(E2)+Cheie(E3)+Cheie(E1)

*Cheile schemelor de relatie rezultate din asocieri*

Legenda:

X + Y: Multimea de atribute X impreuna cu multimea de atribute Y

F. Rădulescu, Curs: Baze de date I

75

Sfârșitul  
cursului 5

F. Rădulescu, Curs: Baze de date I

76