

Cursul 3

Modelarea datelor - Partea 1 -

F. Rădulescu, Curs: Baze de date I

1

Sumar

1. Obiectivele proiectării sistematice. Anomalii
2. Etapele proiectării unei aplicații
3. Modelul entitate-asociere
4. Caracteristicile modelului
5. Criterii de modelare
6. Modelul EA folosit în instrumentele CASE

F. Rădulescu, Curs: Baze de date I

2

OBIECTIV

- ◆ Proiectarea corectă a **structurii** unei baze de date relaționale este o premisă fundamentală în:
- ◆ scrierea programelor de aplicație
- ◆ functionarea lor fără **anomalii** care pot apărea în cazul unei structuri defectuoase.

F. Radulescu, Curs: Baze de date

3

EXEMPLU: Proiectare greșită

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

F. Radulescu, Curs: Baze de date

4

ANOMALII (1)

- ◆ **Redundanta:** Redundanta reprezintă stocarea în mod nejustificat a unei aceleiași informații de mai multe ori în baza de date.
- ◆ Observăm că pentru fiecare produs este stocat numele și adresa furnizorului, deși ele sunt unic determinate de codul acestuia.

F. Radulescu, Curs: Baze de date

5

ANOMALII (2)

- ◆ **Anomalia de stergere:** La stergerea din relație a ultimului produs al unui furnizor se pierde automat și datele despre acesta.

F. Radulescu, Curs: Baze de date

6

ANOMALII (3)

- ◆ **Anomalia de actualizare:** In cazul actualizarii unei informatii redundante, se poate intampla ca operatia sa modifice unele aparitii ale acesteia iar altele sa ramana cu vechea valoare.

F. Radulescu. Curs: Baze de date

7

ANOMALII (4)

- ◆ **Anomalia de inserare:** Nu putem insera date despre un furnizor (numele si adresa sa) decat daca exista in stoc un produs furnizat de acesta.

F. Radulescu. Curs: Baze de date

8

SOLUȚIE

- ◆ Acest capitol prezinta un model de date folosit in proiectarea conceptuala de nivel inalt numit **modelul entitate asociere** (EA) in varianta clasica (cu unele extensii).
- ◆ Intr-un alt capitol vor fi prezentate regulile de transformare din modelul entitate-asociere in modelul relational.

F. Radulescu. Curs: Baze de date

9

MODELUL ENTITATE-ASOCIERE

- ◆ Modelulul entitate-asociere (EA) a dus la gasirea unor cai algoritmizabile de proiectare optima a bazelor de date.
- ◆ Modelul entitate-asociere este in acest moment cel mai popular model de comunicare a structurii bazelor de date datorita intuitivitatii și simplitatii elementelor sale.
- ◆ Imbunatatiri sale ulterioare au dus la crearea de variante ale modelului, doua dintre acestea fiind descrise in acest capitol.

F. Radulescu. Curs: Baze de date

10

EXTENSII ALE MODELULUI EA

- ◆ Modelarea cerintelor de secretizare a datelor,
- ◆ Documentarea programelor de aplicatie și usurarea comunicarii proiectant - utilizator,
- ◆ Proiectarea bazelor de date complexe pe portiuni (integrarea vederilor).

F. Radulescu. Curs: Baze de date

11

ETAPELE PROIECTARII (1)

Pentru baza de date:

1. Analiza de sistem → Cerințe privind baza de date
2. Proiectarea conceptuală → **Diagrama EA**
3. Transformare în model relațional → Schema candidat a bazei de date

F. Radulescu. Curs: Baze de date

12

ETAPELE PROIECTARII (2)

4. Normalizare → Schema conceptuala a bazei de date
5. Implementare specifică SGBD-ului folosit → Schema bazei de date a aplicației

F. Radulescu. Curs: Baze de date

13

ETAPELE PROIECTARII (3)

Pentru programele de aplicatie:

1. Analiza de sistem → Cerințe funcționale ale aplicației
 2. Proiectarea funcțională → Diagrame de specificare funcțională
- Urmeaza apoi realizarea, testarea, implementarea si instruirea personalului → Programele aplicației

F. Radulescu. Curs: Baze de date

14

ETAPELE PROIECTARII (1)

Pentru baza de date:

1. Analiza de sistem → Cerințe privind baza de date
2. Proiectarea conceptuală → Diagrama EA
3. Transformare în model relațional → Schema candidat a bazei de date

F. Radulescu. Curs: Baze de date

15

ANALIZA DE SISTEM

- ◆ Se realizeaza analiza segmentului din lumea reala care va fi gestionat de aplicatia respectiva.
- ◆ Rezulta o specificatie neformalizata a cerintelor constand din doua componente:
 - ◆ **Cerinte privind continutul bazei de date:** categoriile de date care vor fi stocate și interdependentele dintre acestea.
 - ◆ **Cerinte privind prelucrarile efectuate de aplicatie:** prelucrarile efectuate asupra datelor, arborele de meniuri al aplicatiei, machetele formatelor de introducere și prezentare a datelor și ale rapoartelor tiparite de aceasta.

F. Radulescu. Curs: Baze de date

16

ANALIZA DE SISTEM: ACTIVITATI(1)

- ◆ Analiza *activitatii* desfasurate la momentul respectiv de beneficiarul aplicatiei sau de o multime reprezentativa de beneficiari
- ◆ Analiza continutului de date și a functionalitatii *aplicatiilor software* - daca exista - care vor fi inlocuite de noua aplicatie (meniuri, machete ecran, machete rapoarte)
- ◆ Analiza *formulelor tipizate* și a altor documente utilizate de beneficiar pentru realizarea activitatii respective.

F. Radulescu. Curs: Baze de date

17

ANALIZA DE SISTEM: ACTIVITATI(2)

- ◆ Identificarea *interdependentelor* dintre datele stocate in baza de date și a *restrictiilor* privind valorile lor
- ◆ Identificarea prelucrarilor care se declanseaza *automat* atat in cazul modificarii bazei de date cat și la momente prestabilite de timp (de exemplu sfarsit de luna, de an, etc.)
- ◆ Identificarea operatiilor care sunt necesare beneficiarului in activitatea curenta dar care in acest moment *nu sunt realizate* prin intermediul aplicatiilor software folosite precum si a operatiilor care *pot fi incluse* in mod natural in noua aplicatie.

F. Radulescu. Curs: Baze de date

18

ANALIZA DE SISTEM: ACTIVITATI(3)

- ◆ Identificarea *bazelor de date existente* care pot fi folosite de noua aplicatie - direct sau printr-un import initial de date - *evitandu-se reintroducerea manuala* a acestora.
- ◆ Identificarea modalitatilor de *transfer de date* intre noua aplicatie si alte aplicatii care ruleaza deja la beneficiar si care vor fi folosite si in viitor de catre acesta.
- ◆ Identificarea necesitatilor privind datele si *prelucrarile care pot fi in viitor* necesare beneficiarului, deci a *posibilelor dezvoltari* in timp ale aplicatiei.

F. Radulescu. Curs: Baze de date

19

ETAPELE PROIECTARII (1)

Pentru baza de date:

1. Analiza de sistem → Cerințe privind baza de date
2. Proiectarea conceptuală → **Diagrama EA**
3. Transformare în model relațional → Schema candidat a bazei de date

F. Radulescu. Curs: Baze de date

20

PROIECTAREA CONCEPTUALA

- ◆ In aceasta etapa, pornind de la rezultatele analizei de sistem, se realizeaza modelarea cerintelor privind datele folosind un model de nivel inalt.
- ◆ Cel mai popular model folosit pentru aceasta este modelul entitate-asociere (EA).
- ◆ Actualmente exista pe piata numeroase instrumente CASE (Computer Aided Software Engineering) care folosesc diverse variante ale modelului.

F. Radulescu. Curs: Baze de date

21

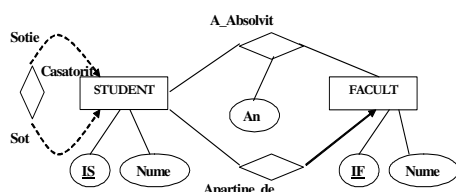
AVANTAJELE EA (1)

- ◆ Nu este legat direct de nici unul dintre modelele folosite de sistemele de gestiune a bazelor de date (relational sau orientat obiect) dar exista algoritmi de transformare din model EA in celelalte modele de date.
- ◆ Este intuitiv, rezultatul modelarii fiind o diagrama care defineste atat datele stocate in baza de date cat si interdependentele dintre acestea.

F. Radulescu. Curs: Baze de date

22

EXEMPLU



F. Radulescu. Curs: Baze de date I

23

AVANTAJELE EA (2)

- ◆ Poate fi ușor de înțeles de nespecialiști și facilitează punerea de acord cu beneficiarul asupra structurii bazei de date a aplicației, evitându-se în acest fel o proiectare neconformă cu realitatea sau cu cerințele.
- ◆ Proiectarea se poate face pe porțiuni, diagramele parțiale rezultate putând fi apoi integrate pe baza unor algoritmi și metode bine puse la punct.

F. Radulescu. Curs: Baze de date

24

ETAPELE PROIECTARII (1)

Pentru baza de date:

1. Analiza de sistem → Cerințe privind baza de date
2. Proiectarea conceptuală → **Diagrama EA**
3. Transformare în model relațional → Schema candidat a bazei de date

F. Radulescu. Curs: Baze de date

25

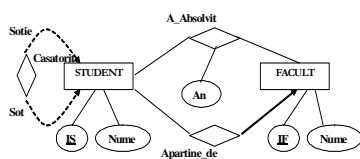
TRANSFORMAREA

- ◆ In aceasta etapa entitatile și asocierile care formeaza diagrama EA se transforma pe baza unor reguli clare in structura relationala a bazei de date.
- ◆ Rezulta schema preliminara a acesteia formata din:
 - ◆ **tabele** (relatii in terminologia relationala),
 - ◆ **coloanele** acestora (atribute ale relatiilor) și
 - ◆ **constrangerile de integritate** care pot fi deduse automat din diagrama incluzand unele interdependente între date numite și "dependențe functionale".
- ◆ In cazul variantei specifice uneltelor CASE transformarea se face automat de catre acestea.

F. Radulescu. Curs: Baze de date

26

Exemplu



- ◆ Student(IS, Nume, Sot_IS, Apartine_de_IF)
- ◆ Facult(IF, Nume)
- ◆ A_Absolvit(IS, IF, An)

F. Radulescu. Curs: Baze de date

27

ETAPELE PROIECTARII (2)

4. Normalizare → Schema conceptuala a bazei de date
5. Implementare specifică SGBD-ului folosit → Schema bazei de date a aplicației

F. Radulescu. Curs: Baze de date

28

NORMALIZAREA

- ◆ Exista o serie de reguli care descriu ce inseamna o structura corecta. Ele definesc asa numitele **forme normale**.
- ◆ Pe baza structurii bazei de date și a dependentelor rezultate atat din transformare și a altor dependente identificate de proiectant in analiza de sistem se poate face o operatie numita **normalizare**: se modifica structura bazei de date astfel incat toate tabelele din aceasta sa fie in forma normala dorita.

F. Radulescu. Curs: Baze de date

29

ETAPELE PROIECTARII (2)

4. Normalizare → Schema conceptuala a bazei de date
5. Implementare specifică SGBD-ului folosit → Schema bazei de date a aplicației

F. Radulescu. Curs: Baze de date

30

IMPLEMENTARE CU UN SGBD

- ◆ In aceasta etapa se realizeaza crearea structurii bazei de date obtinuta in etapa precedenta pe baza facilitatilor oferite de sistemul de gestiune a bazelor de date ales.
- ◆ Rezultatul ei este programul de creare scris in limbajul de definitie a datelor acceptat de SGBD-ul utilizat.

F. Radulescu. Curs: Baze de date

31

EXEMPLU

- ◆ Schema conceptuala:
Student(IS, Nume, Sot_IS, Apartine_de_IF)
- ◆ Program de creare (SQL-Oracle):

```
CREATE TABLE STUDENT(  
  IS NUMBER(5),  
  NUME VARCHAR2(40),  
  SOT_IS NUMBER(5)  
  APARTINE_DE_IF NUMBER(2));
```

F. Radulescu. Curs: Baze de date

32

IMPLEMENTARE CU UN SGBD - continuare

- ◆ Programul contine atat definirea tabelor cat și definirea celorlalte obiecte ale bazei de date (de exemplu constrangerile de integritate intra-tabela și inter-tabele): NOT NULL, PRIMARY KEY.
- ◆ De asemenea aici se fac și toate operatiile privind proiectarea la nivel fizic a bazei de date.
- ◆ In cazul folosirii de unor unelte CASE programul de creare poate fi generat și executat automat.

F. Radulescu. Curs: Baze de date

33

MODELUL ENTITATE ASOCIERE

Modelul entitate-asociere clasic

- ◆ Acest model a fost introdus de P. P. Chen in 1976 ([Ch 76]).
- ◆ O abordare de tip grafic a proiectarii bazelor de date
- ◆ A fost adoptat de comunitatea stiintifica precum si de producatorii de software in domeniu datorita simplitatii si expresivitatii sale.

F. Radulescu. Curs: Baze de date

34

ELEMENTELE MODELULUI

Modelul entitate-asociere permite reprezentarea informatiilor despre structura bazelor de date folosind trei elemente de constructie:

- ◆ Entitati
- ◆ Atribute
- ◆ Asocieri intre entitati

F. Radulescu. Curs: Baze de date

35

ENTITĂȚI (1)

- ◆ **Entitățile** modelează clase de obiecte concrete sau abstracte despre care se colectează informații, au existență independentă și pot fi identificate în mod unic.
- ◆ Exemple de entități:
 - ◆ Studenți,
 - ◆ Orase,
 - ◆ Angajați, etc.
- ◆ Ele definesc de obicei persoane, amplasamente, obiecte sau evenimente cu importanță informațională.

F. Radulescu. Curs: Baze de date

36

ENTITĂȚI (2)

- ◆ Membrii unei clase care formează o astfel de entitate poartă numele de instanțe ale acelei entități.
- ◆ De remarcat că în literatura de specialitate se definește întâi conceptul de mulțime de entități – entity set (sau tip de entități) pentru ca apoi să adopte pentru ușurința exprimării prescurtarea de entitate pentru acest concept.
- ◆ Deci entitatea este un obiect generic care reprezintă mulțimea tuturor instanțelor sale.

F. Radulescu, Curs: Baze de date

37

CATEGORII DE ENTITATI

- ◆ Entități independente (sau tari, eng. strong) sunt cele care au existență independentă de alte entități,
- ◆ Entități dependente (sau slabe, eng. weak) sunt formate din instanțe care își justifică încadrarea în clasă respectivă doar atât timp cât într-o altă entitate (tată) există o anumită instanță de care sunt dependente.

F. Radulescu, Curs: Baze de date

38

Exemplu

Intr-o bază de date de personal avem entitățile:

- ◆ ANGAJATI și
- ◆ COPII (ultima conține copii angajaților companiei)
- ◆ Fiecare instanță a entității COPII depinde de o instanță a entității ANGAJATI
- ◆ Rezultă:
 - ◆ ANGAJATI: entitate independentă,
 - ◆ COPII: entitate dependentă

F. Radulescu, Curs: Baze de date

39

REPREZENTARE GRAFICA

- ◆ Entitățile se reprezintă grafic prin dreptunghiuri în care e înscris numele entității.
- ◆ Exemple:

ANGAJATI

COPII

F. Radulescu, Curs: Baze de date

40

ATTRIBUTE

- ◆ **Atributele** modelează proprietăți atomice distincte ale entităților.
- ◆ Exemplu: entitatea STUDENTI poate avea atributele
 - ◆ Matricolă,
 - ◆ Nume,
 - ◆ Prenume,
 - ◆ Vârsta,
 - ◆ Anul,
 - ◆ Grupa,
 - ◆ Domiciliu – e atomic sau nu?

F. Radulescu, Curs: Baze de date

41

ATTRIBUTE (2)

- ◆ În procesul de modelare vor fi luate în considerare doar acele proprietăți ale entităților care sunt semnificative pentru aplicația respectivă.
- ◆ Exemplu: la entitatea STUDENTI nu vom avea atribute ca:
 - ◆ Talia sau
 - ◆ Culoarea păruluiacestea nefiind necesare pentru baza de date a universității (dar pot exista într-o bază de date privind personalul militar).

F. Radulescu, Curs: Baze de date

42

CLASIFICARE ATRIBUTE

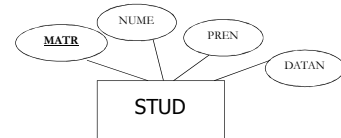
- ◆ atributele de identificare (formand impreuna identificatorul entitatii) reprezinta acea multime de atribute care permit distinctia intre instantele aceleiasi entitati .
- ◆ atributele de descriere (sau descriptori) sunt folositi pentru memorarea caracteristicilor suplimentare ale instantelor.
- ◆ Exemplu: Pentru entitatea STUDENTI
 - ♦ Matricula este atribut de identificare
 - ♦ celelalte atribute sunt descriptori

F. Radulescu. Curs: Baze de date

43

REPREZENTARE GRAFICA

- ◆ Atributele se reprezinta prin ovale sau cercuri in care e in scris numele atributului. Ele sunt conectate la entitatea pe care o caracterizeaza.
- ◆ Exemplu:



F. Radulescu. Curs: Baze de date

44

ASOCIERI

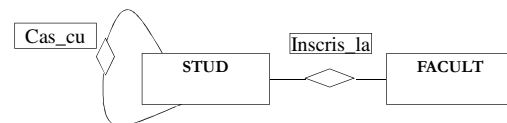
- ◆ **Asocierile** modeleaza interdependentele (sau legaturile) dintre clasele de obiecte reprezentate prin entitati.
 - ♦ Exemplu: intre STUDENTI și FACULTATI se poate figura o asociere INSCRIS_LA care descrie impartirea studentilor pe facultati.
- ◆ In crearea diagramei nu vor fi luate in considerare decit interdependentele care sunt necesare aplicatiei respective (pot exista și alte asocieri care nu sunt semnificative pentru aplicatia proiectata)

F. Radulescu. Curs: Baze de date

45

REPREZENTARE GRAFICA (1)

- ◆ Cand sunt asociate 1-2 entitati: romb

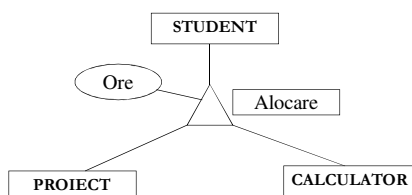


F. Radulescu. Curs: Baze de date

46

REPREZENTARE GRAFICA (2)

- ◆ Cand sunt asociate >3 entitati: poligon



F. Radulescu. Curs: Baze de date

47

EXTENSII ALE MODELULUI

- ◆ Modelul clasic are unele lipsuri in ceea ce priveste posibilitatea modelarii caracteristicilor asociate unor subclase modelate prin entitati.
- ◆ Pentru aceasta, la modelul original au fost adaugate doua noi concepte:
 - ♦ **ierarhia de generalizare și**
 - ♦ **ierarhia de incluziune.**
- ◆ Prima defineste partitionarea instantelor unei entitati in **n** subclase diferite iar a doua permite clasarea unora dintre instantele unei entitati in **m** subclase care nu reprezinta o partitie in sens matematic.

F. Radulescu. Curs: Baze de date

48

IERARHIA DE INCLUZIUNE(1)

- ◆ **Definitie** : O entitate E1 este inclusa in entitatea E daca fiecare instanta a lui E1 este de asemenea o instanta a lui E.
- ◆ Exemplu: in cadrul entitatii ANGAJATI avem subclase modelate prin entitatile:
 - ◆ INGINERI,
 - ◆ ECONOMISTI si
 - ◆ COLABORATORI.

F. Radulescu. Curs: Baze de date

49

IERARHIA DE INCLUZIUNE(2)

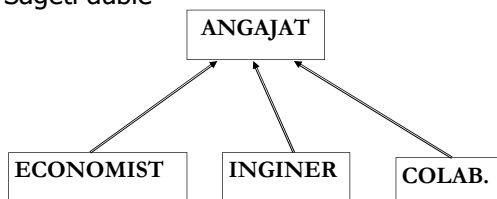
- ◆ In cazul ierarhiei de incluziune entitatile fiu pot sa nu fie disjuncte doua cite doua: pentru exemplul dat, exista angajati ingineri si care sunt incadrati cu contract de colaborare.
- ◆ De asemenea reuniunea lor poate sa nu acopera in intregime entitatea tata: exista angajati care nu sunt nici ingineri, nici economisti si nici colaboratori.

F. Radulescu. Curs: Baze de date

50

REPREZENTARE GRAFICA

Sageti duble



F. Radulescu. Curs: Baze de date

51

IERARHIA DE GENERALIZARE(1)

- ◆ **Definitie (ierarhia de generalizare)**: O entitate E este generalizarea entitatilor E1, E2, ..., En daca orice instanta a lui E este de asemenea instanta in una si numai una din entitatile E1, E2, ..., En.
- ◆ Un exemplu de generalizare este clasarea instantelor entitatii ANGAJATI in subclasele BARBATI si FEMEI.
- ◆ Entitatile fiu reprezinta o clasificare a instantelor entitatii tata

F. Radulescu. Curs: Baze de date

52

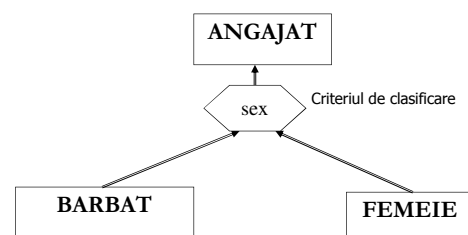
IERARHIA DE GENERALIZARE(2)

- ◆ Caracteristica ierarhiei de generalizare este ca din punct de vedere matematic entitatile fiu reprezinta o partiție a entitatii tata:
 - ◆ a. $E1 \cup E2 \cup \dots \cup En = E$ și
 - ◆ b. $Ei \cap Ej = \emptyset$ pentru orice $i \neq j$ din intervalul 1..n

F. Radulescu. Curs: Baze de date

53

REPREZENTARE GRAFICA



F. Radulescu. Curs: Baze de date

54

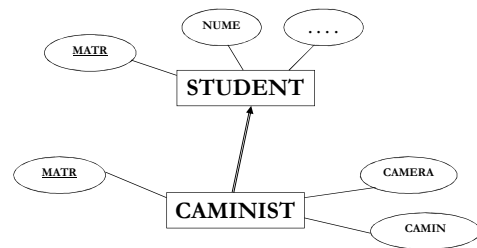
CAND FOLOSIM IERARHII

- ◆ Ierarhiile de incluziune și generalizare se folosesc doar în cazul în care pentru subclasele unor clase modelate prin entități este nevoie de stocarea unor **informații suplimentare specifice**.
- ◆ *Exemplu:* Într-o bază de date de studenți este nevoie de căminul și camera ocupată, dar doar pentru căministi.
- ◆ Acest fapt se poate modela printr-o **entitate suplimentară** CAMINIST aflată într-o relație de **incluziune** cu entitatea STUDENT. Ea va avea ca atribute de identificare pe cele ale tatălui iar ca atribute descriptive Căminul și Camera.

F. Radulescu. Curs: Baze de date

55

DIAGRAMA PT. EXEMPLU



F. Radulescu. Curs: Baze de date

56

CARACTERISTICI ELEMENTE

- ◆ Gradul unei asocieri
- ◆ Conectivitatea unei ramuri de asociere
- ◆ Obligatorietatea unei ramuri de asociere
- ◆ Atribute asocieri
- ◆ Roluri pentru ramurile asocierilor

F. Radulescu. Curs: Baze de date

57

GRADUL ASOCIERII (1)

- ◆ Este o valoare numerică întreaga și este dat de numărul de entități care participă la acea asociere.
- ◆ Poate avea deci valorile 1, 2, 3, 4, 5, ...
- ◆ Asocierile de grad 1, 2 și 3 se mai numesc și asocieri **unare**, **binare** și respectiv **ternare**.

F. Radulescu. Curs: Baze de date

58

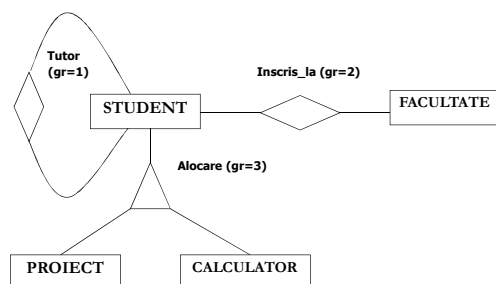
GRADUL ASOCIERII (2)

- ◆ Pentru exemplificare vom considera o bază de date conținând informații despre
 - ◆ studenți,
 - ◆ proiectele realizate de aceștia,
 - ◆ calculatoarele pe care au alocate ore de lucru și
 - ◆ facultățile la care sunt înscriși.
- ◆ De asemenea vom considera că unii dintre studenți au un **tutor** care îi îndrumă, acesta fiind un student dintr-un an mai mare.

F. Radulescu. Curs: Baze de date

59

GRADUL ASOCIERII (3)



Obs: Nu sunt figurate atributele entităților

F. Radulescu. Curs: Baze de date

60

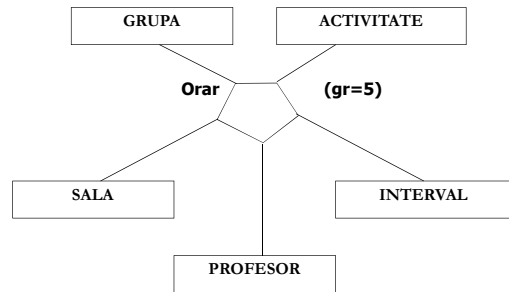
GRADUL ASOCIERII (4)

- ◆ Un exemplu de asociere de grad mai mare ca trei este orarul unui an de studiu al unei facultati.
- ◆ Acesta este o asociere intre urmatoarele entitati:
 - ◆ GRUPE. Fiecare grupa are un cod unic.
 - ◆ SALI. Salile sunt etichetate printr-un indicativ alfanumeric.
 - ◆ INTERVALE ORARE. Un interval orar este un triplet (Zi a saptamanii, De la ora, La ora)
 - ◆ ACTIVITATE. Este o activitate prezenta in orar (curs, laborator, seminar, proiect, la o disciplina).
 - ◆ PROFESOR. Este cadrul didactic titular pentru o activitate

F. Radulescu. Curs: Baze de date

61

GRADUL ASOCIERII (5)



F. Radulescu. Curs: Baze de date

62

CARACTERISTICI ELEMENTE

- ◆ Gradul unei asocieri
- ◆ Conectivitatea unei ramuri de asociere
- ◆ Obligativitatea unei ramuri de asociere
- ◆ Atribute asocieri
- ◆ Roluri pentru ramurile asocierilor

F. Radulescu. Curs: Baze de date

63

CONECTIVITATE (1)

- ◆ Este specifica fiecarei ramuri a unei asocieri și poate avea una din urmatoarele doua valori: **unu** sau **multi** (eng.: **one / many**).
- ◆ Determinarea ei pentru ramura spre o entitate E se face astfel: fixand arbitrar cite o instanta pentru celelalte entitati care participa la asociere se pune intrebarea: cate instante ale lui E pot fi conectate cu acestea?
- ◆ Daca poate fi cel mult una, conectivitatea ramurii este **unu**, altfel conectivitatea este **multi**.

F. Radulescu. Curs: Baze de date

64

CONECTIVITATE (2)

Pentru exemplul STUDENT, FACULTATE, PROIECT, CALCULATOR:

- ◆ Asocierea TUTOR este **unu-unu** sau **multi-uni** dupa cum un student poate fi tutor pentru un singur alt student sau pentru mai multi studenti de an inferior.
- ◆ Asocierea INSCRIS_LA este **multi-unu** (multi spre STUDENT) sau **multi-multi** dupa cum un student poate fi inscris la una sau mai multe facultati.

F. Radulescu. Curs: Baze de date

65

CONECTIVITATE (3)

Asocierea ternara ALOCARE:

- ◆ Spre STUDENT: fiind dat un proiect si un calculator, citi studenti au ore alocate pe acel calculator pentru respectivul proiect? Presupunand ca mai multi studenti lucreaza pentru acelasi proiect pe acelasi calculator ramura va fi **multi**.
- ◆ Spre PROIECT: fiind dat un student și un calculator, la cite proiecte are acesta alocate ore pe acel calculator? Presupunand ca pentru fiecare proiect exista un calculator dedicat, ramura va fi **unu**.
- ◆ Spre CALCULATOR: fiind dat un student și un proiect, pe cate calculatoare are alocate acesta ore pentru realizarea proiectului? Presupunand ca la un proiect se lucreaza pe un singur calculator, ramura va fi **unu**.

F. Radulescu. Curs: Baze de date

66

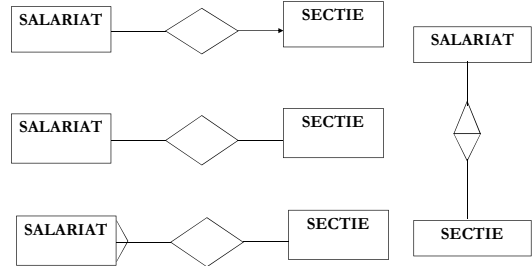
CONECTIVITATE (4)

- ◆ Deci asocierea ALOCARE este multi-unu-unu.
- ◆ Observam ca raspunsul la fiecare din cele trei intrebari se da in functie de realitatea modelata.
- ◆ Aceeasi asociere poate avea conectivitati diferite in cazuri diferite: daca exista chiar si un singur proiect la care un student are ore alocate pe mai mult de un calculator, ramura spre CALCULATOR va fi **multi** iar asocierea va fi multi-unu-multi.

F. Radulescu. Curs: Baze de date

67

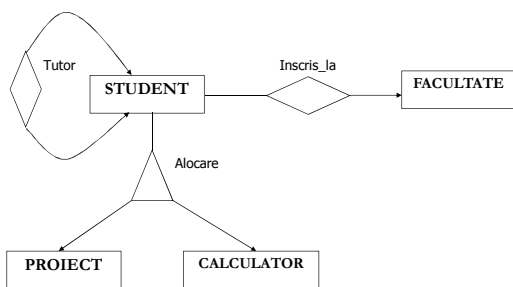
REPREZENTARE GRAFICA



F. Radulescu. Curs: Baze de date

68

EXEMPLU



F. Radulescu. Curs: Baze de date

69

Sfârșitul
cursului 3
Continuarea - saptamana
viitoare!

F. Radulescu. Curs: Baze de date

70