

Cursul 6

Modelul relational - Partea 2 -

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. Calculul relational pe tupluri si domenii

F. Rădulescu, Curs: Baze de date I

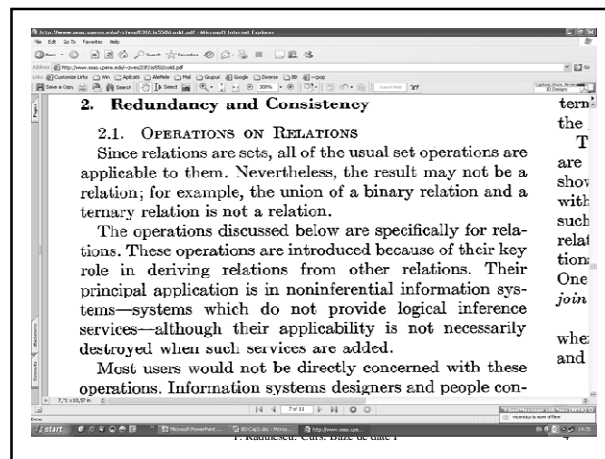
2

ALGEBRA RELATIONALA

- ◆ Inca din primul sau articol in care introduce modelul relational, E.F. Codd propune si un set de operatori pentru lucrul cu relatii.
- ◆ O relatie este o multime de tupluri => o parte dintre acesti operatori provin direct din teoria multimilor.
- ◆ Ceilalti operatori, introdusi in aceasta algebra pentru relatii (numita in literatură de specialitate **algebra relationala**) sunt specifici acesteia si au la baza operatii uzuale cu tabele – acestea fiind reprezentarea intuitiva pentru relatii.

F. Rădulescu, Curs: Baze de date I

3



ALGEBRA RELATIONALA (2)

- ◆ Dupa aparitia primelor sisteme de gestiune a bazelor de date relationale s-a constatat insa ca aceasta algebra nu inglobeaza o serie de situatii care apar in practica:
- ◆ In cazul executiei unei cereri SQL pot sa apara tabele rezultat in care exista linii duplicate.
- ◆ In plus, daca pe o tabela nu a fost definita o cheie primara, putem sa avem in aceasta mai multe linii identice.
- ◆ Problema liniilor duplicat intra in contradictie cu definitia unei relatii in care nu putem avea doua tupluri identice.

F. Rădulescu, Curs: Baze de date I

5

ALGEBRA RELATIONALA (3)

- ◆ In acest subcapitol: variante de operatori:
 - A. Operatori ai algebrei relationale clasice: pornind de la una sau mai multe relatii obtinem o relatie.
 - B. Operatori ai algebrei pe multiseturi - lucreaza pe asa numitele multiseturi (in engleza **bags**) care sunt asemanatoare relatiilor dar in care putem avea elemente duplicat.
 - C. Operatori care lucreaza atat pe relatii cat si pe multiseturi. Ei sunt o extensie a algebrei relationale si pe multiseturi si provin din necesitatea de a putea rescrie orice cerere SQL in termeni ai algebrei extinse.

F. Rădulescu, Curs: Baze de date I

6

ALGEBRA RELATIONALA CLASICA

- ◆ **Exista mai multi operatori in cadrul acestei algebre, unii dintre ei fiind derivati (se pot rescrie in functie de alti operatori). Putem imparti acesti operatori in doua categorii:**
 - ◆ Operatori derivati din teoria multimilor.
 - ◆ Operatori specifici algebrei relationale

F. Rădulescu. Curs: Baze de date I

7

REUNIUNEA

- ◆ **Reuniunea:** Fiind date doua relatii R si S, reuniunea lor, notata $R \cup S$ este o relatie care contine tuplurile care sunt fie in R, fie in S fie in ambele relatii. In rezultatul reuniunii nu apar tupluri duplicat.
- ◆ Pentru ca aceasta operatie sa poata fi executata cele doua relatii care se reunesc trebuie sa aiba scheme compatibile (acelasi numar de coloane provenind din aceleasi domenii (deci cu acelasi tip de date)).
- ◆ Echivalent SQL: operatorul UNION prin care se pot reuni rezultatele a doua cereri SQL de tip SELECT.

F. Rădulescu. Curs: Baze de date I

8

EXEMPLU

```
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 11 AND PUNCTAJ > 2000
UNION
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 21 AND LOC = 'BUCURESTI'
UNION
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 24 AND PUNCTAJ > 1500
ORDER BY LOC DESC, 4;
```

F. Rădulescu. Curs: Baze de date -
Limbajul SQL

9

REUNIUNEA (2)

A	B	C
1	1	2
2	1	3
1	3	2

Relatia R

A	B	C
4	1	2
2	1	3
1	3	2
5	1	7

Relatia S

A	B	C
1	1	2
2	1	3
1	3	2
4	1	2
5	1	7

Relatia $R \cup S$

F. Rădulescu. Curs: Baze de date I

10

DIFERENTA

- ◆ **Diferenta:** Fiind date doua relatii R si S, diferenta lor, notata $R - S$ este o relatie care contine tuplurile care sunt in R si nu sunt in S.
- ◆ Si in cazul diferentei cele doua relatii care se reunesc trebuie sa aiba scheme compatibile.
- ◆ Echivalent SQL: operatorul MINUS prin care se poate face diferenta intre rezultatele a doua cereri SQL de tip SELECT.

F. Rădulescu. Curs: Baze de date I

11

EXEMPLU

```
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 11 AND PUNCTAJ > 2000
UNION
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 21 AND LOC = 'BUCURESTI'
MINUS
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE PUNCTAJ > 1500
ORDER BY LOC DESC, 4;
```

F. Rădulescu. Curs: Baze de date -
Limbajul SQL

12

DIFERENTA (2)

A	B	C
1	1	2
2	1	3
1	3	2

Relatia R

A	B	C
4	1	2
2	1	3
1	3	2
5	1	7

Relatia S

A	B	C
1	1	2

Relatia R - S

F. Rădulescu. Curs: Baze de date I

13

INTERSECTIA

◆ **Intersectia:** Fiind date doua relatii R si S, intersectia lor, notata $R \cap S$ este o relatie care contine tuplurile care sunt si in R si in S. De asemenea cele doua relatii care se reunesc trebuie sa aiba scheme compatibile.

◆ **Echivalent SQL:** operatorul INTERSECT prin care se poate calcula intersectia rezultatelor a doua cereri SQL de tip SELECT.

F. Rădulescu. Curs: Baze de date I

14

EXEMPLU

```
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 11 AND PUNCTAJ > 2000
INTERSECT
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE LOC = 'BUCURESTI'
MINUS
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE PUNCTAJ > 1500
ORDER BY LOC DESC, 4;
```

F. Radulescu. Curs: Baze de date -
Limbajul SQL

15

INTERSECTIA (2)

A	B	C
1	1	2
2	1	3
1	3	2

Relatia R

A	B	C
4	1	2
2	1	3
1	3	2
5	1	7

Relatia S

A	B	C
2	1	3
1	3	2

Relatia $R \cap S$

F. Rădulescu. Curs: Baze de date I

16

INTERSECTIA (3)

◆ **Observatie:** Intersectia este un operator derivat. Putem rescrie orice intersectie astfel:

$$R \cap S = R - (R - S)$$

F. Rădulescu. Curs: Baze de date I

17

PRODUS CARTEZIAN

◆ **Produsul cartezian:** Fiind date doua relatii R si S, produsul lor cartezian, notata $R \times S$ este o relatie ale carei tupluri sunt formate prin concatenarea fiecărei linii a relatiei R cu fiecare linie a relatiei S.

◆ **Rezulta de aici urmatoarele:**

- ◆ Numarul de atribute (coloane) ale lui $R \times S$ este egal cu suma numerelor de atribute ale lui R si S
- ◆ Numarul de tupluri (linii) ale lui $R \times S$ este egal cu produsul numerelor de tupluri ale lui R si S

F. Rădulescu. Curs: Baze de date I

18

PRODUS CARTEZIAN (2)

- Daca in R si S avem attribute (coloane) cu acelasi nume, in produsul cartezian $R \times S$ vom avea attribute care au acelasi nume.
- Pentru a le deosebi se prefixeaza numele atributului cu cel al relatiei din care provine (ex.: R.A si S.A, ca in exemplul urmator)

F. Rădulescu, Curs: Baze de date I

19

PRODUS CARTEZIAN (3)

- ◆ Echivalent SQL:
- ◆ In clauza FROM a unei cereri SELECT apar doua (sau mai multe) tabele
- ◆ In cazul standardului SQL-3, se poate folosi clauza CROSS JOIN a unei cereri de regasire de date de tip SELECT prin care se poate efectua produsul cartezian a doua tabele.

F. Rădulescu, Curs: Baze de date I

20

PRODUS CARTEZIAN

Varianta 1: produs cartezian STUD cu SPEC:

```
SELECT *
FROM STUD, SPEC;
```

va returna un rezultat având 12 coloane și 36 de linii formate din concatenarea fiecărei linii din STUD cu fiecare linie din SPEC.

Varianta 2: acelasi produs cartezian:

```
SELECT *
FROM STUD
CROSS JOIN SPEC;
```

F. Rădulescu, Curs: Baze de date -
Limbajul SQL

21

PRODUS CARTEZIAN (4)

- ◆ Exemplu: Fie relatiile:

A	B	C
1	1	2
2	1	3
1	3	2

Relatia R

A	C	D	E
4	1	2	5
2	1	3	1

Relatia S

F. Rădulescu, Curs: Baze de date I

22

PRODUS CARTEZIAN (4)

- ◆ Rezultat:

R.A	R.B	R.C	S.A	S.C	S.D	S.E
1	1	2	4	1	2	5
1	1	2	2	1	3	1
2	1	3	4	1	2	5
2	1	3	2	1	3	1
1	3	2	4	1	2	5
1	3	2	2	1	3	1

F. Rădulescu, Curs: Baze de date I

23

ALGEBRA RELATIONALA CLASICA

- ◆ Exista mai multi operatori in cadrul acestei algebre, unii dintre ei fiind derivati (se pot rescrie in functie de alti operatori). Putem imparti acesti operatori in doua categorii:
 - ◆ Operatori derivati din teoria multimilor.
 - ◆ Operatori specifici algebrei relationale

F. Rădulescu, Curs: Baze de date I

24

PROIECTIA

◆ **Proiectia:** Fiind data o relatie R si o multime de attribute ale acesteia $X=A_1, A_2, \dots, A_n$, proiectia lui R pe multimea de attribute X este o relatie care se obtine din R luand doar coloanele din X (in aceasta ordine) si eliminand eventualele tupluri duplicat.

◆ Notatia pentru selectie este urmatoarea:

$$\pi_X(R) \text{ sau } \pi_{A_1, A_2, \dots, A_n}(R)$$

F. Rădulescu, Curs: Baze de date I

25

PROIECTIA (2)

◆ Echivalent SQL: Clauza SELECT a unei cereri de regasire de date in care este specificata lista de expresii care da structura de coloane a rezultatului.

◆ Exemplu: din relatia R de mai jos dorim sa calculam $\pi_{B, C, E}(R)$

F. Rădulescu, Curs: Baze de date I

26

PROIECTIA (3)

A	B	C	D	E
1	1	2	1	3
2	1	2	1	3
2	7	4	4	1
2	3	9	2	1
1	3	7	4	1
1	3	9	2	1

Relatia R

B	C	E
1	2	3
7	4	1
3	9	1
3	7	1

Rezultatul proiectiei $\pi_{B, C, E}(R)$
Observam ca s-au eliminat doua linii duplicat din rezultat (cele provenite din liniile 2 si 6).

F. Rădulescu, Curs: Baze de date I

27

PROIECTIA (4)

◆ **Nota:** in multimea de attribute pentru o proiectie poate sa apara toate attributele relatiei. In acest caz se obtine o relatie cu acelasi continut cu cea initiala dar in care coloanele sunt permutate:

$$\pi_{B, C, A, E, D}(R)$$

F. Rădulescu, Curs: Baze de date I

28

PROIECTIA (5)

A	B	C	D	E
1	1	2	1	3
2	1	2	1	3
2	7	4	4	1
2	3	9	2	1
1	3	7	4	1
1	3	9	2	1

Relatia R

B	C	A	E	D
1	2	1	3	1
1	2	2	3	1
7	4	2	1	4
3	9	2	1	2
3	7	1	1	4
3	9	1	1	2

Rezultatul proiectiei $\pi_{B, C, A, E, D}(R)$

F. Rădulescu, Curs: Baze de date I

29

SELECTIA

◆ **Selectia** (numita uneori restrictia):

Fiind data o relatie R si o expresie logica F (o conditie), selectia lui R in raport cu F este o relatie care se obtine din R luand doar liniile care verifica expresia logica F.

◆ Notatia pentru selectie este urmatoarea:

$$\sigma_F(R)$$

F. Rădulescu, Curs: Baze de date I

30

SELECTIA (2)

- ◆ Echivalent SQL: Clauza WHERE a unei cereri de regasire de date de tip SELECT pe care se scrie conditia pe care trebuie sa o indeplineasca liniile pentru a trece mai departe spre rezultat.
- ◆ Exemplu: din relatia R de mai jos dorim sa calculam $\sigma_{B+1 > A+C}(R)$:

F. Rădulescu. Curs: Baze de date I

31

SELECTIA (3)

A	B	C	D	E
1	1	2	1	3
2	1	2	1	3
2	7	4	4	1
2	3	9	2	1
1	3	7	4	1
1	3	9	2	1

Relatia R

A	B	C	D	E
2	7	4	4	1

Rezultatul selectiei $\sigma_{B+1 > A+C}(R)$

F. Rădulescu. Curs: Baze de date I

32

JOIN

- ◆ **Joinul general** (numit si theta-join sau θ -join): fiind date doua relatii R si S, joinul lor (notat $R \bowtie_F S$) se obtine din produsul cartezian al relatiilor R si S urmat de o selectie dupa conditia F (numita si **conditie de join**).
- ◆ Denumirea de theta-join este folosita din motive istorice, simbolul θ fiind folosit initial pentru a desemna o conditie.
- ◆ Rezulta ca:

$$R \bowtie_F S = \sigma_F(R \times S)$$

F. Rădulescu. Curs: Baze de date I

33

JOIN (2)

Sa luam un exemplu concret pentru exemplificarea acestui operator: Sa consideram ca avem doua relatii, STUD si SPEC avand schemele:

- ◆ STUD(Mat, Nume, CodSpec, Media)
- ◆ SPEC(CodS, NumeS)

F. Rădulescu. Curs: Baze de date I

34

JOIN (3)

Matr	Nume	CodSpec	Media
101	Ionescu Ion	10	8
102	Popescu Maria	11	9
302	Georgescu Vasile	10	9,50

Relatia STUD

CodS	NumeS
10	Calculatoare si Tehnologia Informatiei
11	Automatica si Informatica Industriala

Relatia SPEC

F. Rădulescu. Curs: Baze de date I

35

JOIN (4)

- ◆ Sa consideram urmatoarele joinuri:
- ◆ $STUD \bowtie_{STUD.CodSpec=SPEC.CodS} SPEC$
- ◆ $STUD \bowtie_{STUD.CodSpec>SPEC.CodS} SPEC$

- ◆ Rezultatul celor doua joinuri este urmatorul:

F. Rădulescu. Curs: Baze de date I

36

JOIN (5)

◆Echivalent SQL:

- In clauza FROM a unei cereri de regasire de tip SELECT apar tabelele care participa la join +
- In clauza WHERE se pune conditia de join, conectata cu AND de celelalte conditii care eventual sunt necesare in cererea respectiva.

F. Rădulescu, Curs: Baze de date I

37

JOIN (6)

◆Cererile SQL echivalente sunt:

```
SELECT *
FROM STUD, SPEC
WHERE STUD.CodSpec=SPEC.CodS
```

Si:

```
SELECT *
FROM STUD, SPEC
WHERE STUD.CodSpec>SPEC.CodS
```

◆Rezultate obtinute:

F. Rădulescu, Curs: Baze de date I

38

$STUD \bowtie_{STUD.CodSpec=SPEC.CodS} SPEC$

Matr	Nume	CodSpec	Media	CodS	NumeS
101	Ionescu Ion	10	8	10	Calculatoare si Tehnologia Informatiei
102	Popescu Maria	11	9	11	Automatica si Informatica Industriala
302	Georgescu Vasile	10	9,50	10	Calculatoare si Tehnologia Informatiei

F. Rădulescu, Curs: Baze de date I

39

JOIN (7)

- In cazul in care conditia de join este una de egalitate, joinul se mai numeste si **echijoin** (ca in cazul joinului precedent).
- In restul cazurilor se foloseste sintagma **non-echijoin** (joinul urmator).

F. Rădulescu, Curs: Baze de date I

40

$STUD \bowtie_{STUD.CodSpec>SPEC.CodS} SPEC$

Matr	Nume	CodSpec	Media	CodS	NumeS
102	Popescu Maria	11	9	10	Calculatoare si Tehnologia Informatiei

F. Rădulescu, Curs: Baze de date I

41

JOIN NATURAL

- ◆**Join natural:** Joinul natural pentru doua relatii R si S (notat $R \bowtie S$) se obtine:
- ◆facand joinul celor doua relatii dupa conditia: "coloanele cu aceeasi semnificatie au valori egale" +
- ◆ eliminand prin proiectie coloanele duplicat (cele dupa care s-a facut joinul).

F. Rădulescu, Curs: Baze de date I

42

JOIN NATURAL (2)

- ◆ Echivalent SQL: Clauza NATURAL JOIN din sintaxa SQL-3.
- ◆ Observatie: deoarece SGBD-ul nu cunoaste semnificatia coloanelor, conditia de join implicita in acest caz este "coloanele cu acelasi nume au valori egale"

F. Rădulescu, Curs: Baze de date I

43

JOIN NATURAL (3)

- ◆ Exemplu: In cazul celor doua tabele de mai sus, STUD si SPEC, joinul lor natural va fi asemanator cu echijoinul anterior, lipsind insa coloana duplicat SPEC.CodS (care are aceleasi valori ca si coloana STUD.CodSpec)
- ◆ Obs: In cazul folosirii clauzei NATURAL JOIN cele doua coloane trebuie sa aiba acelasi nume

F. Rădulescu, Curs: Baze de date I

44

JOIN NATURAL (4)

Matr	Nume	CodSpec	Media	NumeS
101	Ionescu Ion	10	8	Calculatoare si Tehnologia Informatiei
102	Popescu Maria	11	9	Automatica si Informatica Industriala
302	Georgescu Vasile	10	9,50	Calculatoare si Tehnologia Informatiei

F. Rădulescu, Curs: Baze de date I

45

JOIN EXTERN

- ◆ **Join extern:** Asa cum s-a vazut din nonechijoinul anterior, in cazul in care o linie a unei tabele, oricare ar fi concatenarea ei cu o alta linie din cealalta tabela, nu indeplineste conditia de join, linia respectiva nu are corespondent in rezultat.
- ◆ Este cazul liniilor studentilor de la specializarea 10 si al liniei specializarii 11.

F. Rădulescu, Curs: Baze de date I

46

JOIN EXTERN (2)

- ◆ In unele cazuri se doreste insa ca aceste linii sa apara in rezultat, cu valori nule pe coloanele din cealalta tabela.
- ◆ Aceasta operatie poarta numele de join extern (in engleza outer join).
- ◆ Cum la un join participa doua tabele, pot exista trei tipuri de join extern:

F. Rădulescu, Curs: Baze de date I

47

JOIN EXTERN (3)

- ◆ Join extern stanga (left outer join), in care in rezultat apar toate liniile tabelului din stanga operatorului. Notatia este: $R \bowtie_L S$.
- ◆ Join extern dreapta (right outer join), in care in rezultat apar toate liniile tabelului din dreapta operatorului. Notatia este: $R \bowtie_R S$.
- ◆ Join extern complet (full outer join), in care in rezultat apar toate liniile tabelului din stanga si din dreapta operatorului. Notatia este: $R \bowtie_S S$.
- ◆ De notat ca in rezultatul joinului extern sunt intotdeauna continute tuplurile (liniile) din rezultatul joinului general dupa aceeasi conditie.

F. Rădulescu, Curs: Baze de date I

48

STUD $\bowtie_{L(STUD.CodSpec>SPEC.CodS)}$ SPEC

Matr	Nume	CodSpec	Media	CodS	NumeS
102	Popescu Maria	11	9	10	Calculatoare si Tehnologia Informatiei
101	Ionescu Ion	10	8	NULL	NULL
302	Georgescu Vasile	10	9,50	NULL	NULL

F. Rădulescu, Curs: Baze de date I

49

STUD $\bowtie_{R(STUD.CodSpec>SPEC.CodS)}$ SPEC

Matr	Nume	CodSpec	Media	CodS	NumeS
102	Popescu Maria	11	9	10	Calculatoare si Tehnologia Informatiei
NULL	NULL	NULL	NULL	11	Automatica si Informatica Industriala

F. Rădulescu, Curs: Baze de date I

50

STUD $\bowtie_{(STUD.CodSpec>SPEC.CodS)}$ SPEC

Matr	Nume	CodSpec	Media	CodS	NumeS
102	Popescu Maria	11	9	10	Calculatoare si Tehnologia Informatiei
101	Ionescu Ion	10	8	NULL	NULL
302	Georgescu Vasile	10	9,50	NULL	NULL
NULL	NULL	NULL	NULL	11	Automatica si Informatica Industriala

F. Rădulescu, Curs: Baze de date I

51

Exemplu SQL

◆ Exemplu de FULL OUTER JOIN

```
SELECT S.NUME "NUME STUD",
       S.AN "AN STUD",
       T.NUME "NUME TUTOR",
       T.AN "AN TUTOR"
FROM STUD S
FULL OUTER JOIN STUD T
ON (S.TUTOR=T.MATR);
```

(Intrebare: cate linii are rezultatul?)

F. Rădulescu, Curs: Baze de date -
Limbaajul SQL

52

SEMIJOIN

- ◆ **Semijoin:** Fie doua relatii R si S. Atunci semijoinul lui R in raport cu S (notat $R \ltimes S$) este o relatie care contine multimea tuplurilor lui R care participa la joinul natural cu S.
- ◆ Semijoinul este un operator derivat. Putem scrie ca:
- ◆ $R \ltimes S = \pi_R (R \bowtie S)$
- ◆ Semijoinurile pot fi folosite in optimizarea cererilor de regasire in baze de date distribuite.

F. Rădulescu, Curs: Baze de date I

53

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

54

MULTISETURI

- ◆ Asa cum am spus anterior, in practica bazelor de date intr-o tabela sau un rezultat al unei cereri de regasire de date pot sa apara linii duplicat.
- ◆ In acest caz nu mai putem vorbi de relatii (care nu permit tupluri duplicat) ci de multiseturi (eng. **bags**). Prezintam pe scurt efectul unora dintre operatorii de mai sus aplicati multiseturilor.

F. Rădulescu, Curs: Baze de date I

55

REUNIUNE MULTISETURI

- ◆ **Reuniunea:** Efectul este asemanator cu al reuniunii din algebra relationala dar din rezultatul final nu se elimina duplicatele.

F. Rădulescu, Curs: Baze de date I

56

EXEMPLU

A	B	C
1	1	2
1	1	2
1	3	2

Multiset R

A	B	C
1	3	2
2	1	3

Multiset S

A	B	C
1	1	2
1	1	2
1	3	2
1	3	2
2	1	3

Multiset $R \cup S$

F. Rădulescu, Curs: Baze de date I

57

ALTE OPERATII

- ◆ **Intersectia, diferenta, produsul cartezian, selectia, joinul, joinul natural, joinul extern:** acelasi mod de calcul ca si in cazul relatiilor dar:
- ◆ Multiseturile operand pot sa contina linii duplicat
- ◆ Din rezultat nu se elimina liniile duplicat
- ◆ Observatie: in cazul acestor operatii nu pot aparea linii duplicat decat daca operanzii contin linii duplicat.

F. Rădulescu, Curs: Baze de date I

58

PROIECTIE MULTISETURI

- ◆ **Proiectia:** Acelasi mod de calcul ca si in cazul relatiilor dar la final nu eliminam liniile duplicat.

F. Rădulescu, Curs: Baze de date I

59

EXEMPLU

A	B	C	D	E
1	1	2	1	3
2	1	2	1	3
2	7	4	4	1
2	3	9	2	1
1	3	7	4	1
1	3	9	2	1

Multiset R

B	C	E
1	2	3
1	2	3
7	4	1
3	9	1
3	7	1
3	9	1

Rezultatul proiectiei $\pi_{B,C,E}(R)$ pentru multisetul R

Observam ca nu s-au eliminat liniile duplicat

F. Rădulescu, Curs: Baze de date I

60

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

61

OPERATORI EXTINSI

- ◆ Redenumirea
- ◆ Eliminare duplicate
- ◆ Grupare
- ◆ Sortare
- ◆ Proiectie extinsa

F. Rădulescu, Curs: Baze de date I

62

REDENUMIREA

- ◆ **Redenumirea:** Exista doua modalitati de a face redenumirea tabelelor si/sau coloanelor:
- 1. **Operatorul de redenumire ρ** permite atat redenumirea relatiilor/multiseturilor cat si a atributelor acestora:
- ◆ Fiind data o relatie R, putem obtine o alta relatie $S = \rho_{S(A_1, A_2, \dots, A_n)}(R)$ care are acelasi continut ca si R dar attributele se numesc A_1, A_2, \dots, A_n .

F. Rădulescu, Curs: Baze de date I

63

REDENUMIREA (2)

- ◆ Echivalent SQL pentru operatorul ρ : Aliasurile de coloana si de tabela folosite in clauzele SELECT, respectiv FROM dintr-o cerere de regasire de tip SELECT

F. Rădulescu, Curs: Baze de date I

64

REDENUMIREA (3)

- ◆ **Constructorul** \rightarrow care permite redenumirea atributelor in rezultatul unei expresii relationale sau pe multiseturi:
- ◆ Putem redenumi intr-un rezultat un atribut prin constructia:
- ◆ $Nume_vechi \rightarrow Nume_nou$

F. Rădulescu, Curs: Baze de date I

65

REDENUMIREA (4)

- ◆ Exemplu: Fie o relatie $R=ABCDE$.
- ◆ In rezultatul proiectiei:
$$\pi_{B \rightarrow Nume, C \rightarrow Prenume, E \rightarrow DataN}(R)$$
attributele nu sunt B, C si E ci Nume, Prenume si DataN
- ◆ Echivalent SQL: aliasul de coloana folosit in clauza SELECT a unei cereri de regasire.

F. Rădulescu, Curs: Baze de date I

66

ELIMINARE DUPLICATE

- ◆ **Eliminare duplicate:** Acest operator se poate aplica doar pe multiseturi (relatiile nu contin tupluri duplicat). Efectul este eliminarea duplicatelor din multiset. Notatia operatorului este urmatoarea:
- ◆ Fiind dat un multiset R , $\delta(R)$ este un multiset fara duplicate (deci o relatie)
- ◆ Echivalent SQL: SELECT DISTINCT dintr-o cerere de regasire de tip SELECT

F. Rădulescu. Curs: Baze de date I

67

ELIMINARE DUPLICATE (2)

A	B	C
1	2	3
1	2	3
7	4	1
3	9	1
3	7	1
3	9	1

Multisetul R

A	B	C
1	2	3
7	4	1
3	9	1
3	7	1

Multisetul (relatia) $\delta(R)$

F. Rădulescu. Curs: Baze de date I

68

GRUPARE

- ◆ **Grupare:** Forma operatorului de grupare este urmatoarea:
- $\gamma_{\text{Lista_atribute_si_functii_statistice}}(R)$
- ◆ Atributele din lista sunt criterii de grupare. Ele apar in rezultatul returnat de operator
- ◆ Functiile statistice din lista (ex.: MIN, MAX, SUM, AVG, COUNT) se calculeaza la nivelul fiecarui grup si de asemenea apar in rezultatul operatorului

F. Rădulescu. Curs: Baze de date I

69

GRUPARE (2)

- ◆ Acest operator se poate aplica atat relatiilor cat si multiseturilor.
- ◆ Echivalent SQL: Functii statistice si grupare cu GROUP BY
- ◆ Un exemplu in acest sens este edificator: In cazul relatiei STUD anterioare:
- $\gamma_{\text{CodSpec, Count(*)} \rightarrow \text{NrStud, AVG(Medie)} \rightarrow \text{Medie}}(\text{STUD})$
- va returna o relatie avand urmatorul continut:

F. Rădulescu. Curs: Baze de date I

70

GRUPARE (3)

CodSpec	NrStud	Medie
10	2	8,75
11	1	9,00

F. Rădulescu. Curs: Baze de date I

71

SORTARE (ORDONARE)

- ◆ **Sortare:** Forma operatorului de sortare este urmatoarea:
- $\tau_{\text{Lista_atribute}}(R)$
- ◆ Efectul este sortarea relatiei sau multisetului R in functie de atributele din lista.

F. Rădulescu. Curs: Baze de date I

72

SORTARE (ORDONARE) - cont

- ◆ Cum atat in cazul relatiilor cat si a multiseturilor nu este presupusa o relatie de ordine, acest operator practic nu modifica argumentul (doar rearanjeaza elementele).
- ◆ Deci el are sens doar atunci cand este ultimul aplicat unei expresii.
- ◆ Echivalent SQL: Clauza ORDER BY dintr-o cerere de regasire de tip SELECT

F. Rădulescu. Curs: Baze de date I

73

PROIECTIE EXTINSA

- ◆ **Proiectie extinsa:** Acest operator este analog proiectiei obisnuite dar permite attribute (coloane) calculate pentru rezultatul unei expresii pe relatii sau multiseturi. Forma sa este urmatoarea:

$$\pi_{\text{Expresie1, Expresie2, ... Expresie-n}}(R)$$

- ◆ **Observatie:** in functie de rezultatul dorit (relatie sau multiset) dupa ce se calculeaza rezultatele duplicatele se elimina sau nu se elimina.

F. Rădulescu. Curs: Baze de date I

74

PROIECTIE EXTINSA (2)

- ◆ Echivalent SQL: Ca si in cazul proiectiei obisnuite, acest operator este implementat prin clauza SELECT a unei cereri de regasire a informatiei
- ◆ Exemplu: Pentru expresia:

$$\pi_{\text{Nume, Medie*2} \rightarrow \text{Dublu}}(\text{STUD})$$
- ◆ Rezultatul este:

F. Rădulescu. Curs: Baze de date I

75

PROIECTIE EXTINSA

Nume	Dublu
Ionescu Ion	16
Popescu Maria	18
Georgescu Vasile	19

F. Rădulescu. Curs: Baze de date I

76

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

77

CALCUL RELATIONAL

- ◆ Pe langa algebra relationala, cererile de regasire a informatiei intr-o baza de date relationala pot fi exprimate si prin:
 - ◆ calcul relational pe tupluri (CRT)
 - ◆ calcul relational pe domenii (CRD).

F. Rădulescu. Curs: Baze de date I

78

CALCUL RELATIONAL PE TUPLURI - CRT

- ◆ In calculul relational pe tupluri o cerere se exprima printr-o **expresie** de forma:

$$\{ t \mid \psi(t) \}$$
- ◆ t este o **variabila tuplu** iar ψ o **formula**.
- ◆ Semnificatia expresiei este "multimea tuturor tuplurilor t care verifica formula ψ ".

F. Rădulescu, Curs: Baze de date I

79

ATOMI

- ◆ Formula este compusa din elemente (numite si atomi) care pot fi de trei tipuri:
- ◆ Elemente de tip $R(s)$ unde R este un nume de relatie iar s o variabila tuplu. Semnificatia este " s este un tuplu din R ".
- ◆ Elemente de tip $s[i] \theta v[j]$ unde s si v sunt variabile tuplu iar θ un operator prin care se poate compara componenta i a variabilei tuplu s cu componenta j a variabilei tuplu v .
- ◆ $s[i] \theta a$ sau $a \theta s[i]$ prin care componenta i a variabilei tuplu s se compara cu constanta a .

F. Rădulescu, Curs: Baze de date I

80

APARITII

- ◆ Pe baza acestor atomi se poate defini recursiv ce este o formula si ce sunt aparitii **libere** sau **legate** ale variabilelor tuplu:
1. Orice atom este in acelasi timp formula. Toate aparitiile unei variabile tuplu intr-un atom sunt aparitii libere

F. Rădulescu, Curs: Baze de date I

81

APARITII (2)

2. Daca ψ si ϕ sunt doua formule, atunci $\psi \vee \phi$, $\psi \wedge \phi$ si $\neg\psi$ sunt formule cu semnificatia ψ sau logic ϕ , ψ si logic ϕ si respectiv "not ψ ".
- Aparitiile de variabile tuplu sunt libere sau legate in aceste formule dupa cum ele sunt libere sau legate in componentele acestora. Este permis ca o aceeași variabila tuplu sa aiba o aparitie libera in ψ si o alta legata in ϕ .

F. Rădulescu, Curs: Baze de date I

82

APARITII (3)

3. Daca ψ este o formula atunci si $(\exists s)(\psi)$ este formula. Aparitiile variabilei tuplu s care sunt libere in ψ sunt legate in $(\exists s)(\psi)$. Celelalte aparitii de variabile tuplu din ψ raman la fel (libere sau legate) in $(\exists s)(\psi)$.
- Semnificatia acestei formule este urmatoarea: exista o valoare concreta a lui s care inlocuita in toate aparitiile libere din ψ face ca aceasta sa fie adevarata.

F. Rădulescu, Curs: Baze de date I

83

APARITII (4)

4. Daca ψ este o formula atunci si $(\forall s)(\psi)$ este formula. Aparitiile variabilei tuplu s care sunt libere in ψ sunt legate in $(\forall s)(\psi)$. Celelalte aparitii de variabile tuplu din ψ raman la fel (libere sau legate) in $(\forall s)(\psi)$.
- Semnificatia acestei formule este urmatoarea: orice valoare concreta a lui s pusa in locul aparitiilor libere ale acestuia din ψ face ca ψ sa fie adevarata.

F. Rădulescu, Curs: Baze de date I

84

APARITII (5)

5. Parantezele pot fi folosite in formule dupa necesitati.

Precedenta este: intai comparatiile, apoi \exists si \forall si in final \neg , \wedge , \vee (in aceasta ordine)

F. Rădulescu, Curs: Baze de date I

85

EXEMPLE (EXPRESII, FORMULE)

◆ Exemple de expresii si formule:

1. Expresia $\{t \mid R(t) \vee S(t)\}$ este echivalenta reuniunii a doua relatii din algebra relationala.
2. Analog $\{t \mid R(t) \wedge S(t)\}$ reprezinta intersectia a doua relatii.
3. Expresia pentru proiectia lui R pe attributele i_1, i_2, \dots, i_k se poate scrie astfel:

$$\{t(k) \mid (\exists u)(R(u) \wedge t[1] = u[i_1] \wedge t[2] = u[i_2] \wedge \dots \wedge t[k] = u[i_k])\}$$
4. Formula $(\exists s)(R(s))$ spune ca relatia R este nevida

F. Rădulescu, Curs: Baze de date I

86

EXPRESII SIGURE

◆ Din pacate unele din expresiile scrise in calcul relational pe tupluri duc la rezultate infinite.

- ◆ Exemplu: daca R este o relatie finita atunci expresia $\{t \mid R(t)\}$ este de asemenea finita dar expresia $\{t \mid \neg R(t)\}$ este infinita (exista o infinitate de tupluri care nu apartin lui R).

◆ Pentru a evita astfel de rezultate s-au introdus asa numitele **expresii sigure**. Pentru definirea lor este necesara definirea unui alt concept si anume *domeniul unei formule*

F. Rădulescu, Curs: Baze de date I

87

DOM(ψ)

- ◆ **Definitie:** Daca ψ este o formula atunci domeniul sau, notat cu $DOM(\psi)$ este multimea tuturor valorilor care fie apar explicit in ψ sau sunt componente ale tuplurilor relatiilor prezente in ψ .
- ◆ Cum orice relatie este finita rezulta ca si domeniul oricarei formule este finit.

F. Rădulescu, Curs: Baze de date I

88

EXEMPLU

- ◆ Fie formula $\psi = R(t) \wedge t[1] > 100$ care reprezinta conditia pentru o selectie din R dupa conditia "valoarea pe prima coloana este mai mare decat 100". Atunci:

$$DOM(\psi) = \{100\} \cup \{\text{multimea valorilor care apar in tuplurile lui R}\}$$

F. Rădulescu, Curs: Baze de date I

89

EXPRESIE SIGURA

- ◆ **Definitie:** O expresie ψ se zice ca este sigura daca rezultatul sau este compus doar din valori apartinand lui $DOM(\psi)$.

F. Rădulescu, Curs: Baze de date I

90

EXEMPLE

Expresiile:

- ◆ $\{t \mid R(t)\}$,
- ◆ $\{t \mid R(t) \wedge t[1] > 100\}$,
- ◆ $\{t \mid R(t) \vee S(t)\}$, $\{t \mid R(t) \wedge S(t)\}$ sau
- ◆ $\{t(k) \mid (\exists u)(R(u) \wedge t[1] = u[i1] \wedge t[2] = u[i2] \wedge \dots \wedge t[k] = u[ik])\}$

sunt sigure

F. Rădulescu, Curs: Baze de date I

91

EXEMPLE – cont.

Expresiile:

- ◆ $\{t \mid \neg R(t)\}$
 - ◆ $\{t \mid \neg R(t) \wedge \neg S(t)\}$
- nu sunt sigure.

F. Rădulescu, Curs: Baze de date I

92

ECHIVALENTA CRT CU AR

- ◆ Se poate demonstra ca expresiile sigure din CRT sunt echivalente cu expresii din algebra relationala si reciproc.

F. Rădulescu, Curs: Baze de date I

93

CALCUL RELATIONAL PE DOMENII - CRD

- ◆ In calculul relational pe domenii nu avem variabile tuplu ci variabile de domeniu, ele constituind elementele care formeaza tuplurile.
- ◆ In acest caz trebuiesc rescrise regulile de formare pentru o formula in CRD dupa cum urmeaza:

F. Rădulescu, Curs: Baze de date I

94

ATOM

Un atom poate fi:

- ◆ $R(x_1, x_2, \dots, x_n)$ unde R este o relatie iar x_i sunt variabile de domeniu sau constante
- ◆ $x \theta y$ unde x si y sunt variabile de domeniu sau constante iar θ este in continuare un operator de comparatie.

F. Rădulescu, Curs: Baze de date I

95

FORMULE SI EXPRESII

- ◆ Formulele din CRD sunt construite analog cu cele din CRT utilizand de asemenea \neg , \wedge , \vee precum si \exists , \forall .
- ◆ Notiunile de aparitie libera sau legata a unei variabile de domeniu sunt analoge cu cele din CRT
- ◆ Analog cu CRT se definesc: domeniul unei variabile de domeniu $DOM(x)$ si expresii sigure in CRD.

F. Rădulescu, Curs: Baze de date I

96

EXEMPLE – EXPRESII SIGURE

- ◆ Reuniunea a doua relatii R si S:
 $\{x_1x_2...x_n \mid R(x_1x_2...x_n) \vee S(x_1x_2...x_n)\}$
- ◆ Intersectia a doua relatii R si S
 $\{x_1x_2...x_n \mid R(x_1x_2...x_n) \wedge S(x_1x_2...x_n)\}$
- ◆ Selectia dupa conditia "valoarea pe prima coloana este mai mare decat 100":
 $\{x_1x_2...x_n \mid R(x_1x_2...x_n) \wedge x_1 > 100\}$

F. Rădulescu, Curs: Baze de date I

97

EXEMPLE: EXPRESII CARE NU SUNT SIGURE

- ◆ $\{x_1x_2...x_n \mid \neg R(x_1x_2...x_n)\}$
- ◆ $\{x_1x_2...x_n \mid \neg R(x_1x_2...x_n) \wedge \neg S(x_1x_2...x_n)\}$

F. Rădulescu, Curs: Baze de date I

98

ECHIVALENTA CRD CU AR

- ◆ In literatura de specialitate se arata ca expresiile sigure din CRD sunt echivalente cu expresii din algebra relationala si reciproc (deci implicit cele doua formalisme – CRD si AR – sunt echivalente).

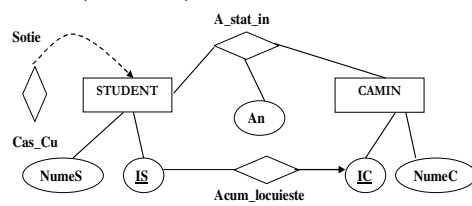
F. Rădulescu, Curs: Baze de date I

99

Exercitii

Exercitiul 1. Fie diagrama entitate asociere de mai jos.

- a. Sa se corecteze erorile de sintaxa
 - b. Sa se transforme in model relational
- A_stat_in: M-M, Cas_cu: 1-1, Acum_locuieste: M-1
 IS: IdStudent, IC: IdCamin, ambele identificatori



Lucrare la mijlocul semestrului,
28.4.2009

100

Exercitii

Exercitiul 2. Fie $R=ABC$, $S=CDE$, ambele cu cate 3 tupluri cu:

$R[i, j] = i + j$, $S[i, j] = i + 2*j + 1$ (i =indice de linie, j =indice de coloana)

1. Sa se scrie explicit relatiile R si S.
2. Sa se calculeze joinul natural intre relatiile R si S.
3. Sa se calculeze joinul extern complet intre relatiile R si S dupa conditia $R.C=S.C$.
4. Considerand operatorii extinsi scrieti rezultatul expresiei

$$\gamma_A \rightarrow \text{Crt, COUNT(*)} \rightarrow \text{Cati}(R \times S)$$

F. Rădulescu, Curs: Baze de date I

101

Sfârșitul cursului 6

F. Rădulescu, Curs: Baze de date I

102