UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

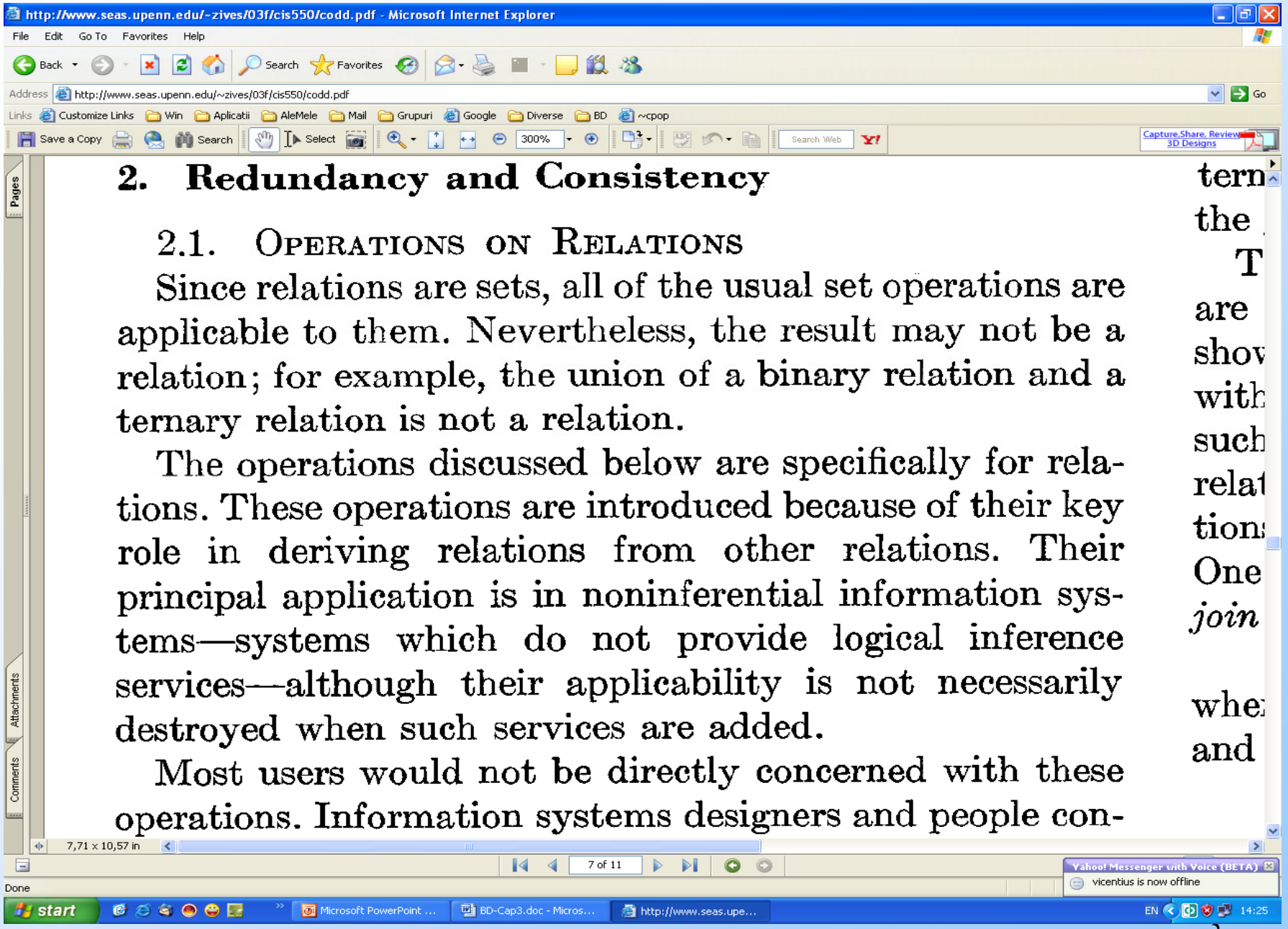
Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Baze de date 1

3. Elemente de algebră relaţională

ALGEBRA RELATIONALA

- Inca din primul articol in care introduce modelul relational, E.F. Codd propune 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 literature de specialitate ***algebra relationala***) sunt specifici acesteia si au la baza operatii uzuale cu tabele – acestea fiind reprezentarea intuitiva pentru relatii.


http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf - Microsoft Internet Explorer
File Edit Go To Favorites Help
Back Search Favorites
Address http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf
Go
Links Customize Links Win Aplicatii AleMele Mail Grupuri Google Diverse BD ~cpop
Save a Copy Search Select 300%
Capture,Share, Review 3D Designs
Pages
Attachments
Comments
2. Redundancy and Consistency
2.1. Operations on Relations
Since relations are sets, all of the usual set operations are applicable to them. Nevertheless, the result may not be a relation; for example, the union of a binary relation and a ternary relation is not a relation.
The operations discussed below are specifically for relations. These operations are introduced because of their key role in deriving relations from other relations. Their principal application is in noninferential information systems—systems which do not provide logical inference services—although their applicability is not necessarily destroyed when such services are added.
Most users would not be directly concerned with these operations. Information systems designers and people con-
7,71 x 10,57 in
7 of 11
Done
Yahoo! Messenger with Voice (BETA)
vicentius is now offline
start
Microsoft PowerPoint ...
BD-Cap3.doc - Micros...
http://www.seas.upe...
EN
14:25

OPERATORI

- 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

REUNIUNEA

- **Reuniunea:** Fiind date doua relatii R si S, reuniunea lor, notata R ∪ 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.

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 $R \cup S$

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.

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

INTERSECTIA

- **Intersectia:** Fiind date doua relatii R si S, intersectia lor, notata R ∩ 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.

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 ∩ S

INTERSECTIA (3)

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

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

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 fiecarei 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

PRODUS CARTEZIAN (2)

- Daca in R si S avem atribute (coloane) cu acelasi nume, in produsul cartezian R $\times$ S vom avea atribute 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)

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.

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

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

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

PROIECTIA

- **Proiectia:** Fiind data o relatie R si o multime de atribute ale acesteia X=A1, A2, … An, proiectia lui R pe multimea de atribute 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:

$$\pi_X(R) \text{ sau } \pi_{A1, A2, \ldots An}(R)$$

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)

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).

PROIECTIA (4)

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

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

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

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:

$$\sigma_F(R)$$

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

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

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:

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

JOIN (2)

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

- STUD(Matr, Nume, CodSpec, Media)
- SPEC(CodS, NumeS)

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

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:

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

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

JOIN (5)

- 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).

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

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

JOIN (6)

- 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.

JOIN NATURAL

- **Join natural:** Joinul natural pentru doua relatii R si S (notat R⋈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).

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"

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

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

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.

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:

JOIN EXTERN (3)

- Join extern stanga (left outer join), in care in rezultat apar toate liniile tabelei din stanga operatorului. Notatia este: $R \lhd^{o} \rhd_{L} S$.
- Join extern dreapta (right outer join), in care in rezultat apar toate liniile tabelei din dreapta operatorului. Notatia este: $R \lhd^{o} \rhd_{R} S$.
- Join extern complet (full outer join), in care in rezultat apar toate liniile tabelelor din stanga si din dreapta operatorului. Notatia este: $R \lhd^{o} \rhd S$.
- De notat ca in rezultatul joinului extern sunt intotdeauna continute tuplurile (liniile) din rezultatul joinului general dupa aceeasi conditie.

$$STUD \lhd^{o} \rhd_{L\ (STUD.CodSpec>SPEC.CodS)} SPEC$$

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

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

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

$$STUD \lhd^{o} \rhd_{(STUD.CodSpec>SPEC.CodS)} SPEC$$

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

SEMIJOIN

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

Bibliografie

1. **Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom**: *Database Systems: The Complete Book*, Prentice-Hall, Englewood Cliffs, NJ, 2002.
2. **F. Rădulescu** : *Oracle SQL, PL/SQL*, Editura Printech, ISBN 973-718-203-02005