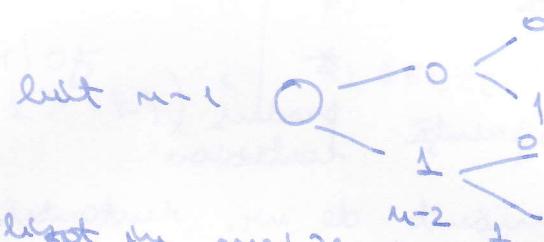


Functii HASH - complete

- f-hash (cump - hash) → pozitioneaza zona surugurării
 → hash - interne: pp. stocarea datelor în memorie internă
 ex: stocare tabelă pt. rougere
 → hash - extern: stocarea datelor pe memorie externă
 OBS: este diferență de index; la index este o structură aditivabilă de date, pe când hash este un cod unic care nu este aditivabil de seces direct.
 OBS: faza HASH este folosită la autentificare (ID + parola).
 OBS: la stocare și la seces se folosesc același funcție HASH (după seces, criteriu).
- Dacă p.d.v. se repres. interne un cump - hash e format dintr-o succesiune de biti, considerăm o funcție HASH care returnează primii n biti ai surugurării interne.

$\underbrace{X \ X \ X}_{\text{nr biti}}, \dots, X$



HASH-ul este mai rar utilizat în mediuile de BD, din motivul că, de regulă în BD se lucrau cu tabele și cărora conținut se modifică în timp, iar secesul la date în BD se face după criterii diferențiale de timp diferențate. Deci, clasa HASH în BD nu folosește nici unele tabele statice cu o distribuție a datelor cunoscută (necesară pt. construirea funcției HASH).

Cap. 4. Algebra relatională

Prezintă un set de operații matematice ce pot fi executate pe mulțimile de relații. Dacă p.d.v. se algebrei relationale, o relație este un număr de obiecte: $R(A_1, A_2, \dots, A_n)$.

Valeurile obiecte sunt stabilite între o relație - v_i sprijinind domeniul obiectului obiectului - $\text{Dom}(A_i) \Rightarrow v_i \in \text{Dom}(A_i)$. Domeniul este caracterizat de:

- 1) tip de date
- 2) constanțări (numai anumite valori din secundul domeniu sunt valide).

Elementul din relație = relație existență de - este date de:

$r = (v_1, v_2, \dots, v_n) - n\text{-uple}$ multime de n -upleuri

$\Rightarrow v_i = \text{val. obiectului } A_i$

Proprietăți

→ ordinul relației $R = \text{ord}(R) = \text{nr. de stăriute} = \text{nr. de componente}$
 dintr-o instanță; "nr. de linii de la tabelă"

$$R = (A_1, \dots, A_n) \Rightarrow \text{ord}(R) = n$$

→ cardinalul relației $R = \text{card}(R) = \text{nr. de instante}; \text{card}(R) \in \mathbb{N};$
 $\text{card}(R)$ ar fi identic cu nr. de subiecte dintr-o tabelă;

Algebrele relaționale apar să toate tabelile sunt corespunzătoare și sunt
 consistentă (nu există după).

AR urmărește să def. pe structurile R un set de operații.

a) Operații simple - se aplică unei singure relații;

{ SELECT

{ PROJECT

b) Operații繁 - se aplică de 2 tabele (sau mai multe);

$$U, \cap, D, *, \text{JOIN}$$

diferență produs
 cartezian

c) Funcții: dependență de nr. instanțelor la care se aplică o funcție, f:

- funcție agrigată: aplică se aplică pe subiecte relație

- funcție de grup: aplică subiecte instante dintr-o relație și sub-
 grupare o funcție de grupare predefinită.

OBS: rezultatul + op. sl AR este tot o relație, relație rămâne
 tot consistentă (fără după).

Op. AR cum sunt def. aci sunt cele care sunt la baza implementării limbajului de urmă. - Query.

SELECT

Dă def. sintetică:

$S < \text{condiție} > (\text{nume_relație})$

• condiție = set de conditii atomice:

condiție := cond-ct 1 op logic cond-ct 2 op logic ... cond-ct n

- cond. atomice sunt legate prin operatori logici;

- op. logici pot fi \wedge \vee \neg

\neg (aplicat pe o condiție)

condiție-atomică := nume atribut op relational valoare

- op. relational: $<$, $>$, $=$, \leq , \geq , \neq between

se dă. Diferența, după enunțate comparații se bucură că de-
partamentul 2 sau 5 nu are un salarit mai mare decât 1000.

$$\text{Rez} = S < (\text{depart_nr} = 2 \text{ OR } \text{depart_nr} = 5) \text{ AND } \text{salarit} > 1000 \text{ (Augajot)}$$

$$\text{ord}(\text{Rez}) = \text{ord}(\text{Augajot})$$

- Rez menține stabilitatea lui Augajot;

$$\text{card}(\text{Rez}) \leq \text{card}(\text{Augajot})$$

Totalele menținute ele relației pt care evaluarea condiției este TRUE
sunt incluse în rezultat, totalele menținute pt care evaluarea cond. este
FALSE nu vor fi incluse în rezultat.

Succesiune de operații SELECT:

$$S < \text{cond 1} > (S < \text{cond 2} > (R))$$

faza de

$$S < \text{cond 2} > (S < \text{cond 1} > (R))$$

↑

\Rightarrow din rezultatele intermediare \Rightarrow

\Rightarrow SELECT este comutativ

$$S < \text{cond 1 AND cond 2} > (R) \text{ (rezultatul este restrâns).}$$

PROJECT

$$P < \text{listă stăriute} > (\text{căruri relație})$$

Ex: $P < \text{nume, Prenume, Depart_nr} > (\text{Augajot}) = \text{Rez}$
 \hookrightarrow se obține o relație cu stăriutele nume, Prenume, departură
cu un nr. de stăriute dependente de Augajot;

$$\text{ord}(\text{Rez}) \leq \text{ord}(P)$$

($\text{Rez} = R$ și $\text{ord}(\text{listă}) \leq \text{ord}(\text{relație})$);

nr. stăriute relație Project

$$\text{card}(\text{Rez}) \leq \text{card}(R)$$

($\text{card}(\text{Rez}) = \text{card}(R) \rightarrow$ din lista
de P să se include stăriutele care
sunt relației R);

$$P < \text{listă 1} > (P < \text{listă 2} > (R)) \leftarrow \text{listă 1} \subset \text{listă 2}$$

faza de

$$P < \text{listă 2} > (P < \text{listă 1} > (R)) \leftarrow \text{listă 2} \subset \text{listă 1}$$

\hookrightarrow OP. PROJECT NU este comutativ; ea nu poate opăsi și e
valabilă doar de la listă de stăriute la care se aplică P și generă din
ele listă de stăriute - 2 - rezultatul.

$\text{Res}(\text{nume } A_1, \dots, \text{nume } A_k) = S < \text{cond} > (R)$

OBS: primul element al relației R este redenumit cu numele nume A_1 și, a. u. d. redenumire

$\text{Res}(\text{nume}_f, \text{prez}, \text{departament}) = P < \text{Nume, Genume, departare} >$
(Augerat)

bunăstigii SELECT - PROJ ECAT:

$S < \text{cond} > (P < \text{listă obiecte} > (R)) \quad (**)$

nu

$P < \text{listă obiecte} > (S < \text{cond} > (R)) \quad (*)$

- de general, se execută întotdeauna și apoi P

- ** merge numai deoarece se generează în liste obiectelor;

Ex: să se scrie urm. că prenumele angajaților se apără depart. 2 sau 4 născuți după 1 ian. 1970.

$P < \text{nume, gen} > (S < (\text{depart_nr} = 2 \text{OR} \text{depart_nr} = 4) \text{ AND}$
 $\text{date_naștere} > \{1/01/1970\} > (\text{Augerat}). \quad (1)$

Op. AR se execută sequential.

$R_1 = S < (\text{depart_nr} = 2 \text{ OR} \text{depart_nr} = 4) \text{ AND} (\text{date_naștere} > \{1/11/1970\})$

$\text{Res} = P < \text{nume, genume} > (R_1)$

adun cu (1).

Op. binare

② setul teoretic: unele din teoreme multivariante;

U, \cap, Δ

$R_1(A_1, A_2, \dots, A_m)$

$R_2(B_1, B_2, \dots, B_n)$

$R_1 \cup R_2 \Rightarrow R_1 \cap R_2, R_1 \Delta R_2$ - posibile numere de元

R_1 și R_2 sunt compatibile, adică:

a) $\text{ord}(R_1) = \text{ord}(R_2) \quad (n = m)$

b) $\text{dom}(A_i) = \text{dom}(B_i), i = 1:m$

toate relațiile din R₁ și toate din R₂ sunt o singură dată.

Δ: relațiile din R₁ și care se găsesc în R₂;

Δ: relațiile care sunt din R₁ și nu se găsesc din R₂.

• ord(R₁₂) = ord(R₁) = ord(R₂)

$\overbrace{\quad}^{\rightarrow \text{compatibilitate}},$

↳ valoarea pt U, ∩ și D;

• U : card(R₁₂) ≤ card(R₁) + card(R₂)

Δ: card(R₁₂) ≤ min { card(R₁), card(R₂) }

Δ: card(R₁₂) ≤ card(R₁) (R₁ ⊂ R₂);

⑤ Produs cartesian - inclus tot în setul teoretic,

$$R_1(A_1, A_2, \dots, A_m)$$

$$R_2(B_1, B_2, \dots, B_n)$$

$$R_{12} = R_1 \times R_2$$

$$R_{12} \in \{A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_n\};$$

~~ord~~ ord(R₁₂) = ord(R₁) + ord(R₂);

$$\text{card}(R_{12}) = \text{card}(R_1) \cdot \text{card}(R_2).$$

-produsul cartesian nu include restricții referitoare la domenii, deci rezultatul este foarte mare;

Ex: Date complete pt. angajați și departamentul din care ei fac parte

$$R_{12} = \text{Angajat} \times \text{Departament}$$

⑥ JOIN: este op. de combinare conditivitate a relațiilor referente la 2 relații. Op. JOIN este similar cu op. produs cartesian însă necesită un patruț de condiție de JOIN d.c. o relație a unei relații se combină numai cu relațiile ~~nu~~ celelalte relații ce să depășească condiția.

$$R_{12} = R_1 \bowtie \langle \text{condiție - JOIN} \rangle R_2$$

\bowtie = simbolul operației JOIN

condiție de JOIN nu este definită astfel:

cond-JOIN := cond - stocarea JOIN op logice cond-ctg ... cond-ctg.

cond-atom JOIN = intersecrea R₁ op. relational nume - stocare R₂

Ex.: Să se obțină o relație ce conține toate informațiile
angajat supravîndut cu informațiile referente departamentului său care
angajatul face parte.

Res = Budget \times < deposit-wr = D-wr > \times department
 ↑ ↑
 menet wr menet wr
 Budget Department

$$\text{cord}(\text{Res}) = \text{cord}(\text{Augerot}).$$

Ex.: Luto - managerilor și a departamentelor la care sunt manageri,

第22章

RS = Augsjöt \bowtie CHP = manager \bowtie department

$\text{word}(\text{Re}) = \text{word}(\text{Department})$

Project < Name, Brename, I-wr > (Rs) = Res.

$$\text{word}(\text{Res}) = \text{word}(\text{Re}).$$

Ex: Să se dă: lista sugestivor care luresă la proiecte cordiale și de dezvoltare și să se sugestivă împreună.

note de departamentul său vorbește și în ceea ce privește
deportul din zone care fac parte
din regiuni (Maramureș, ...), CNP, și - ură)

Project (PName, ... , P-wc, sd-wc)

Leverenz (CNP, P-wc, Dre)

a) $R_1 = \text{Augeret } \Delta \quad (\omega - \omega_r = \omega - \omega_c) \quad \text{Beweis}$

$$R_2 = P \leftarrow \text{Name}, \text{Year}, \text{Pr-name}, \text{cnd}, \text{Pr-wr.} \rightarrow (R_1)$$

$R_3 = R_2 \wedge C_{NP} = CNP$ AND $P_{-wL} = P_{-wR} > L_{max}$

$\text{Res} = P \leftarrow \text{None}, \text{Bun}, \text{Re-req}, \text{One} \rangle$ (R3)

- lista cu nume, loc și angajat, proiectul se va lucra

Mr. de Gre;

b) În ce moditate de rezolvare;

OBS: ordene de execu^çe este secretaria; se pode ser o total
seu-o sanguine bruto.

• TOTAL JOIN: condiție de egalație între stăriile în calea nume.

$R_1 \bowtie R_2$

↑ natural join;

- rezolvare RL:

$R_1 = \text{Arganjut } \bowtie \text{ Proiect};$

- EQU JOIN: condiție este tot de egalitate dar stările nu sunt mereu identice;

$R_1 \bowtie (\text{listă obiecte}) R_2$

- se interpretează ca egalitate între liste de obiecte;

Arganjut(..., Dep-w, ...)

Depărtament(..., d-w, ...);

Arganjut \bowtie (Dep-w, d-w) Proiect;

(se interpretează ca $\text{Dep-w} = \text{d-w}$);

- JOIN selectiv $\begin{cases} \text{stanga} \\ \text{dreapta} \end{cases}$; analiză în mod specific de executare

pt JOIN pt relație din partea stg, respectiv pt cea din dr. JOIN-ul selecțional sănătosă condiție din dreapta, respectiv stanga.

a) $R_1] \bowtie \langle \text{cond.-join} \rangle R_2$ (join stanga)

b) $R_1 \bowtie [\langle \text{cond.-join} \rangle R_2$

- a) se combină prezentările celor 2 relații și se îndeplinește cond. de JOIN și în plus există relația R_1 și nu îndeplinește cond de JOIN vor fi incluse în rezultat combinații cu o instanță NULL a relației R_2 .

R_1

R_2

	A ₁	...	A _m	B ₁	B ₂	...	B _m
①					NULL		
②					27		
③	/	/	/		33		
④						HULL	

// cond. nu e îndeplinită

// cond.-join = TRUE

// cond.-join = TRUE

// cond.-join = FALSE

- analog pt. JOIN dreapta;

$\text{ord}(\text{Res}) \geq \text{ord}(\text{R})$

Functii

Fetile sunt în considerare de TR pe parte din categorii fetelor statistic. Ele permit operații de calcul pe valorile unui sau mai multor atributuri și sunt op. de calcul pe relație.

Depinde de tipurile de funcții, avem:

• COUNT (atribut) sau COUNT (*)

- pl. numără toate instanțele din relație

- de. $\text{Attribut} = \text{NULL}$ - nu se numără, dar + numărul tot;

- fizice COUNT se poate aplica pe + tip (numără, sîr de coacere)

• MIN (atribut) - val. min a atributului. Specific

MAX (atribut) - val. max - - -

- este aplicabilă pt. toate atributurile și în cadrul de ordonare

• SUM (atribut) - se poate aplica numai pt. atributurile cu domeniu de tip număr; rezultatul este suma valorilor atributului.

• AVERAGE (atribut) - aplicată numai pt. atributul cu valori numere, furnizând media valorilor atributului.

- fizice agregat: operațiile de calcul specificate de fizie se execută pe toate instanțele relației, iar rezultatul operațiunii fizice agregat sunt o relație cu o singură instanță.

$\begin{cases} \text{ord}(\text{Res}) = \text{nr. funcții aplicate}; \\ \text{ord}(\text{Res}) = 1 \end{cases}$

✓ fizice de grup: se relatează pt. valorile distincte ale atributului sau atributelor de grupare.

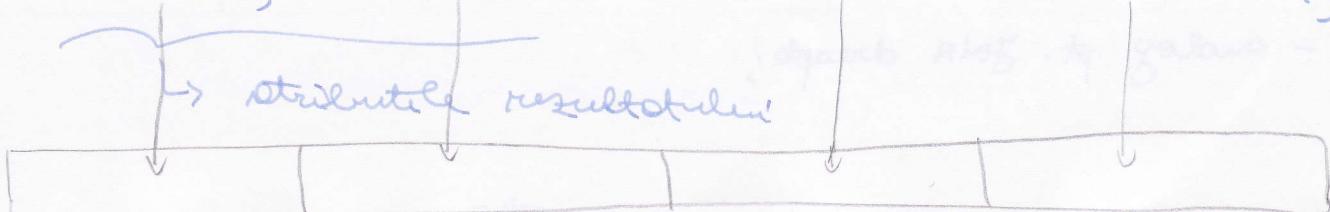
$\text{ord}(\text{Res}) = \text{nr. funcții} + \text{nr. atributelor grupare}$

$\text{ord}(\text{Res}) = \text{nr. valori} \neq \text{nr. atributelor de grupare}.$

a) fizie agregat:

$\text{Res} = f < \text{count}(\text{cnt}), \text{MIN}(\text{salariu}), \text{MAX}(\text{salariu}), \text{AVERAGE}(\text{salariu}) \rangle$
(fizică);

$\text{Res} (\text{COUNT_cnt}, \text{MIN_salariu}, \text{MAX_salariu}, \text{AVERAGE_salariu})$



redemunirea compună pt. Res:

Res(nr_eng, Sal_min, Sal_max, medie) = $\tilde{f} \dots$;

- tipul de date este cel măsurat din funcție explicită;

b) felile grupă:

Ex: să se obțină nr. angajaților pe depart., sal. max pe fiecare depart. și medie salariailor pe depart.

\langle student grupă $\rangle \tilde{f} \langle$ lista funcție \rangle (Rezultat)

\langle d_nr $\rangle \tilde{f} \langle$ COUNT(*), MAX(salariu), AVERAGE(salariu) \rangle (angajat)

d_nr. COUNT * MAX_sal AVERAGE_sal

1	14	--	--
2	8	--	--
:			

"nr. veloci" \neq pt d_nr.

Ex: să se obțină pt fiecare departament, valoarea \neq a salariailor și nr. de angajați care au un același salar.

- structura grupă = d_nr + salariu;

\langle d_nr, salariu $\rangle \tilde{f} \langle$ COUNT(*) \rangle (angajat)

- rezultat are 3 structuri $\begin{cases} d_nr. \\ salariu \end{cases}$ } distinții

nr. de angajați cu același salariu,

Ex: să se det. lista angajaților (Nume, Prenume, CNP), suprenom cu nume nr. de ore desfășurate săptămânal la toate proiectele le care angajat lucrașă.

$\begin{cases} \text{Angajat} \\ \text{Lucrare} \end{cases}$

Versante 1:

- JOIN Angajat + Lucrare după JOIN - MARE : (
- SUM cu grupe după CNP

Versante 2:

- SUM cu Lucrare cu grupe CNP - :)
- JOIN cu Angajat după CNP

R1 = < CNP > f < sum (ore) > (lăureză)

R2 = #. Angajat \bowtie R1

R3 = P < nume, prenume, CNP, sum_ore > (R2).

Formularul de verificare:

- 1) În se det. există sugestorii companiei care lăurează mai puțin de 40 de ore pe săptămână în proiect.
- 2) În se det. există sugestorii din departamentele la care numele managerului este Popescu, și lăurează în cel puțin un proiect coordonat de departamentul din care sugestorul face parte.
 - JOIN Angajat - depozit cu nume manager = Popescu - R1
 - JOIN R1 cu Angajat pt că sunt sugestori cu manager Popescu
 - JOIN cu Proiect
 - JOIN cu Lăureză
 - Se vor avea sau 6 operații
- 3) În se det. numele și prenumele sugestorilor se lăurează în cel puțin 3 proiecte și au cel puțin 2 persoane din întreținere.
 - verificare în Lăureză - COUNT se va det. nr de proiecte cu un select;
 - JOIN cu sugestor
 - JOIN cu întreținut și apoi cu sugestor.
- 4) În se det. numele managerilor depozit și coordonator proiecte la care următoarelor săptămâni le proiecte depășește val. 400.
 - Angajat, depozit, Proiect, Lăureză.
- 5) În se det. lista departamentelor se coordonază în cel puțin 2 proiecte, fără să fie dăună provocată peste 200 de ore / săptămână, proiectele care lăurează în cel puțin un sugestor din departamentul.

OBS: Op. AR sunt reveribile.

Efectuarea operațiilor depende de ordinea și modul în care se execută operațiile și există limită (f. max).