Operațiuni în modelul relațional. Introducere în algebra relațională

prof. dr. ing. Mircea Petrescu

Limbaj de interogare = limbaj în care un utilizator solicită informații din baza de date (BD). De obicei, limbajele de interogare sunt de nivel mai înalt decât limbajele standard de programare. Limbajele de interogare sunt procedurale sau ne-procedurale. În limbajele procedurale utilizatorul indică sistemului succesiunea de operații asupra BD pentru a determina rezultatul dorit. În limbajele ne-procedurale, utilizatorul descrie rezultatul dorit, fără a indica procedura prin care acesta este obținut.

Cele mai multe sisteme relaționale de BD folosesc un limbaj de interogare în care sunt prezente elemente ale ambelor abordări, atât procedurală, cât şi ne-procedurală. Limbaje foarte cunoscute: SQL, QBE, Datalog.

În cele care urmează – o introducere în limbajele „fundamentale” sau „pure”, respectiv algebra relațională şi calculul relațional; sunt limbaje matematice, formale, ambele asociate cu modelul relațional de date. Algebra relațională este un limbaj procedural, pe când calculul relațional pe tupluri şi calculul relațional pe domenii sunt limbaje ne-procedurale. Ambele familii de limbaje sunt concise şi formale, fără a poseda „cadrul sintactic” al limbajelor comerciale de interogare; algebra relațională şi calculul relațional sunt însă limbaje care pun în evidență foarte bine tehnicile principale folosite în procesul găsirii şi extragerii informației din BD. Desigur, un limbaj complet destinat manipulării BD nu este limitat la operațiuni de interogare, ci îndeplineşte şi funcțiuni de modificare a conținutului bazei de date. Astfel de funcțiuni constă în inserarea şi eliminarea de tupluri în executarea unor comenzi de modificare a tuplurilor ş.a.

Înainte de a prezenta operațiunile principale din algebra relațională, vom formula câteva observații privind unele modele de date, în contextul de aici. În abordarea obiectuală, pentru definirea (sau descrierea) datelor este folosit limbajul LDO (Limbaj de definire a obiectelor = ODL = Object Definition Language). Să comparăm, pe scurt, abordările modelelor entitate-asociere, obiectual şi relațional privind operațiunile asupra datelor:
- în modelul EA nu se precizează, de obicei, o metodă specifică pentru manipularea datelor;
- în LDO, deci în modelul obiectual, sunt folosite metode prin care se pot efectua orice operații asupra datelor;
- în modelul relațional, aşa cum vom vedea, se foloseşte o familie „standard” de operații asupra datelor.

Observația principală stă în faptul că în abordarea relațională familia de operații menționată nu este „completă” din punct de vedere Turing. Cu alte cuvinte, există operațiuni asupra informațiilor dintr-o BD, care nu pot fi exprimate în algebra relațională, dar pot fi exprimate în metode LDO codificate în limbaje obişnuite ca C++. Remarcăm că acesta nu este rezultatul unei slăbiciuni a algebrei relaționale sau a modelului relațional în general. Avantajul restrângerii „lărgimii” („spațiului”) ocupat de operațiile relaționale constă în posibilitatea de a optimiza frazele de interogare formulate într-un limbaj de programare de nivel foarte înalt, aşa cum este SQL.

Elemente de algebră relațională

Aşa cum s-a afirmat anterior, algebra relațională este unul din cele două limbaje formale de interogare ale modelului relațional. Algebra relațională oferă mijloace puternice de a construi relații noi din alte relații date. Atunci când relațiile date sunt reprezentate de informații memorate, relațiile construite cu mijloacele algebrei pot fi răspunsuri la fraze de interogare asupra acestor informații.

Orice algebră permite construirea de expresii prin aplicarea unor operatori asupra unor operanzi atomici sau asupra altor expresii algebrice. Adesea, pentru a grupa operatorii şi operanzii sunt folosite paranteze.

În algebra relațională, operanzii sunt:

a) variabile, care reprezintă relații;
b) constante, care sunt relații finite.

În algebra relațională „clasică” toți operanzii şi toate rezultatele expresiilor sunt mulțimi. Vom grupa operațiile din algebra relațională tradițională (sau „clasică”) în patru clase:

a) operațiunile specifice teoriei mulțimilor (reuniune, intersecție, diferență), dar aplicate asupra relațiilor;
b) operațiunile care îndepărtează părți ale unei relații (selecție, proiecție);
c) operațiunile care combină tuplurile a două relații (produs cartezian, joncțiune)
d) operațiunea prin care sunt atribuite nume noi atributelor relației şi/sau relației.

O proprietate fundamentală în algebra relațională constă în faptul că fiecare operator acceptă înstanțele unei (sau a două) relații în calitate de argumente şi întoarce ca rezultat o altă înstanță de relație. Această proprietate permite folosirea compusă (compunerea) operatorilor pentru a forma fraze de interogare complexe. O astfel de frază de interogare corespunde unei expresii algebrice relaționale, care se defineşte recursiv ca fiind o relație, un operator algebric unar aplicat unei singure expresii, sau ca un operator algebric binar aplicat la două expresii.

Operațiuni pe mulțimi aplicate relațiilor

Reuniunea

Fie r, s relații. Reuniunea este t , $r \cup s$, unde $t = \{$ tupluri t_i, a. î. $t_i \in r, \forall t_i \in r \}$. Condiții: r, s au mulțimi identice de atribute, cu aceleaşi domenii de valori. Exemplu:

r

nume	adresă	gen	datanaşterii
Vlad Ionescu	Str. Paris 20, Bucureşti	M	23/4/1942
Raluca Cernat	Str. Horia 45, Cluj	F	15/6/1955

s

nume	adresă	gen	datanaşterii
Raluca Cernat	Str. Horia 45, Cluj	F	15/6/1955
Dan Teodoru	Str. Lungă 38, Braşov	M	8/11/1962

Rezultatul reuniunii $t = r \cup s$:

t

nume	adresă	gen	datanaşterii
Vlad Ionescu	Str. Paris 20, Bucureşti	M	23/4/1942
Raluca Cernat	Str. Horia 45, Cluj	F	15/6/1955
Dan Teodoru	Str. Lungă 38, Braşov	M	8/11/1962

Intersecția

$t = r \cap s$, unde $t = \{$ tupluri t_i, a. î. $t_i \in r \wedge t_i \in s \}$.

Rezultatul intersecției $t = r \cap s$:

t

nume	adresă	gen	datanaşterii
Raluca Cernat	Str. Horia 45, Cluj	F	15/6/1955

Diferența

$t = r - s$, unde $t = \{$ tupluri t_i, a. î. $t_i \in r \wedge t_i \notin s \}$.
$t = s - r$, unde $t = \{$ tupluri t_i, a. î. $t_i \in s \wedge t_i \notin r \}$.
Observăm că $r - s \neq s - r$.

Exemple:

r – s

nume	adresă	gen	datanaşterii
Vlad Ionescu	Str. Paris 20, Bucureşti	M	23/4/1942

s – r

nume	adresă	gen	datanaşterii
Dan Teodoru	Str. Lungă 38, Braşov	M	8/11/1962

Proiecția

Fie relația r cu atributele $A_1, A_2, ..., A_n$. Fie, de asemenea, atributele $A_1, A_2, ..., A_k$, a. î. $\{ A_1, A_2, ..., A_k \} \subset \{ A_1, A_2, ..., A_n \}$. Atunci, proiecția relației r pe atributele $A_1, A_2, ..., A_k$ (sau pe coloanele acestor atribute), notată cu $\prod_{A1, A2, ..., Ak}(r)$, este relația obținută din r prin extragerea coloanelor atributelor $A_1, A_2, ..., A_k$.

Exemplu – expresia $\prod_{\text{nume, datanaşterii}}(s)$ conduce la rezultatul:

nume	datanaşterii
Raluca Cernat	15/6/1955
Dan Teodoru	8/11/1962

Notă. Proiecția se poate defini formal ca fiind relația:
$\prod_{i1,i2, ..., ik} = \{$ tupluri t_i, a. î. $t_i = (a_1, a_2, ..., a_k)$, iar în r $\exists$ tuplul $(b_1, b_2, ..., b_n)$, $a_j = b_j$ pentru $j = 1... k \}$
Mai sus a_j, respectiv b_j, sunt valori ale atributelor corespunzătoare din mulțimile $\{ A_1, A_2, ..., A_k \}$ şi, evident, $\{ A_1, A_2, ..., A_n \}$. Simbolurile i_1, i_2, etc. Reprezintă coloane din r.

Selecția

Prin definiție, operațiunea de selecție aplicată unei relații r, notată cu $\sigma_F(r)$, constă în extragerea lui r a acelor tupluri care îndeplinesc clauza (formula) F. Schema relației obținute este aceeaşi cu schema relației r, atributele fiind aranjate – prin convenție – în aceeaşi ordine. Operanzii conținuți în clauza F sunt constante sau atribute din schema relației r. Operatorii sunt fie operatori aritmetici uzuali (de comparație), fie operatori logici.

Exemplu: $\sigma_{\text{datanaşterii} \leq 23/4/1962 \vee \text{datanaşterii} > 15/6/1955}(t)$, unde relația t a fost calculată prin reuniune într-un exemplu precedent. Rezultat:

nume	adresă	gen	datanaşterii
Vlad Ionescu	Str. Paris 20, Bucureşti	M	23/4/1942
Dan Teodoru	Str. Lungă 38, Braşov	M	8/11/1962

Produsul cartezian

Fie relațiile r şi s de arități R_1, R_2. Fie în r şi s tuplurile $(r_{i1}, r_{i2}, ..., r_{iK1})$, respectiv $(s_{i1}, s_{i2}, ..., s_{ik2})$. Formal, produsul cartezian $t = r \times s$ al relațiilor r şi s se defineşte prin:

$t = \{$ tupluri t_i, a. î. $t_i = (r_{i1}r_{i2}...r_{iK1}s_{i1}s_{i2}...s_{iK2})$, cu $(r_{i1}r_{i2}...r_{iK1}) \in r$ şi $(s_{i1}s_{i2}...s_{iK2}) \in s \}$. Exemplu:

r

A	B	C
a	b	c
α	β	χ

s

D	E	F
d	e	f
δ	ε	ϕ

r × s

A	B	C	D	E	F
a	b	c	d	e	f
a	b	c	δ	ε	ϕ
α	β	χ	d	e	f
α	β	χ	δ	ε	ϕ

Observație: numărul de tupluri ale produsului cartezian este produsul numerelor de tupluri ale relațiilor r şi s. Consecințe privind timpul de execuție şi memoria ocupată.

Situație particulară – cea în care relațiile operand au atribute comune. Exemplu:

r

A	B	C
a	b	c
α	β	χ

s

C	D	E
q	e	f
c	ε	ϕ

r × s

A	B	r.C	s.C	D	E
a	b	c	q	e	f
a	b	c	c	ε	ϕ
α	β	χ	q	e	f
α	β	χ	c	ε	ϕ

Observăm că în relația rezultat, numele atributelor comune sunt completate cu numele relațiilor din care provin (r.C, s.C).

Joncțiunea „teta"

Joncțiunea „teta" este o operație compusă, care implică efectuarea unui produs cartezian şi a unei selecții. Fie relațiile r şi s, precum şi o clauză F care conține ca operanzi constante şi atribute din schemele relațiilor, iar ca operatori – operatorii aritmetici uzuali şi operatorii logici. Notația folosită pentru reprezentarea operației joncțiune „teta" este $\frac{r \rhd\lhd s}{F}$. Conform definiției, joncțiunea „teta" este efectuată în următoarea ordine:

a) se calculează produsul cartezian r × s;
b) din relația rezultată sunt extrase prin selecție tuplurile care satisfac clauza F.

Exemplu: $\frac{r \rhd\lhd s}{B < \beta}$

Fie relațiile r şi s definite ca mai jos:

r

A	B	C
a	b	c
α	β	χ

s

B	D	E
b	d	e
$\beta/2$	δ	ε

Produsul cartezian:

r × s

A	r.B	C	s.B	D	E
a	b	c	b	d	e
a	b	c	β/2	δ	ε
α	β	χ	b	d	e
α	β	χ	β/2	δ	ε

Selecția:

$\sigma_{B<\beta}$(r × s)

A	r.B	C	s.B	D	E
a	b	c	β/2	δ	ε
α	β	χ	β/2	δ	ε

Fireşte, în cazul relațiilor mai ample pot să apară clauze mai complexe ca cea din exemplul precedent.

Joncțiunea naturală

Fie relațiile r şi s ale căror scheme conțin un număr de atribute comune. Joncțiunea naturală, notată cu simbolul $r \rhd\lhd s$, se calculează în două etape:

a) mai intâi, este determinat produsul cartezian r × s;
b) apoi, asupra produsului cartezian obținut, este efectuată o operațiune de selecție, prin extragerea tuplurilor care conțin acelaşi valori ale atributelor comune din schemele relațiilor incidente r şi s;
c) în final, sunt eliminate coloanele redundante rezultate.

Exemplu: $r \rhd\lhd s$

r

A	B	C
a	b	c
α	β	χ

s

B	D	E
b	d	e
β	δ	ε

Produsul cartezian:

r × s

A	r.B	C	s.B	D	E
a	b	c	b	d	e
a	b	c	β	δ	ε
α	β	χ	b	d	e
α	β	χ	β	δ	ε

Selecția:

$\sigma_{r.B=s.B}$(r × s)

A	r.B	C	s.B	D	E
a	b	c	b	d	e
α	β	χ	β	δ	ε

Rezultatul final – eliminăm una din coloanele redundante r.B sau s.B:

A	B	C	D	E
a	b	c	d	e
α	β	χ	δ	ε

Iată un exemplu mai apropiat de realitate:

ARHITECT

Nume	Adresă	DataNaşterii
Călin	Str. Lungă	5.5.70
Sandu	Str. Albă	15.10.72

CONSTRUCTOR

Nume	Adresă	Domeniul
Sandu	Str. Albă	Civile
Mihai	Str. Nouă	Industriale

Evident că joncţiunea naturală ARHITECT $\rhd\lhd$ CONSTRUCTOR ne conduce la tuplul (Sandu, Str. Albă, 15.10.72, Civile), care se referă la arhitectul Sandu, care conduce sau efectuează – în acelaşi timp – şi lucrări de construcţii. Rămân însă tuplurile „dangling" (Călin, Str. Lungă, 5.5.70) din relaţia ARHITECT şi (Mihai, Str. Nouă, Industriale) din relaţia CONSTRUCTOR, a căror informaţie poate fi pierdută prin efectuarea operaţiunii de joncţiune naturală. Pentru a evita această situaţie se realizează o joncţiune naturală externă completă:

ARHITECT NATURAL FULL OUTER JOIN CONSTRUCTOR =

Nume	Adresă	DataNaşterii	Domeniul
Sandu	Str. Albă	15.10.72	Civile
Călin	Str. Lungă	5.5.70	NULL
Mihai	Str. Nouă	NULL	Industriale

Desigur, se pot efectua, asemănător, operaţiunile de joncţiune naturală externă stânga şi dreapta.

Semijoncţiunea

Prin definiţie, semijoncţiunea relaţiilor r şi s, notată $r \rhd< s$, este proiecţia joncţiunii naturale a celor două relaţii pe atributele din r: $r \rhd< s = \Pi_R(r \rhd\lhd s)$. O formulă echivalentă pentru realizarea acestei operaţiuni este $r \rhd< s = r \rhd\lhd \Pi_{R\cap S}(s)$. În general, semijoncţiunea nu este simetrică, deci $r \rhd< s \neq s \rhd< r$.

Joncţiunea externă

Fie relaţiile r şi s cu schemele R(A, B, C, D), S(A, B, E, F):

r

A	B	C	D
a	b	c	d
m	n	p	q

s

A	B	E	F
a	b	e	f
u	v	x	y

Joncţiunea naturală:

r × s

A.R	B.R	C	D	A.S	B.S	E	F
a	b	c	d	a	b	e	f
a	b	c	d	u	v	x	y
m	n	p	q	a	b	e	f
m	n	p	q	u	v	x	y

Evident că $r \rhd< s$ =

A	B	C	D	E	F
a	b	c	d	e	f

Observăm însă că tuplurile (m, n, p, q) din r şi (u, v, x, y) din s se pierd în urma operaţiei $r \rhd< s$, ceea ce ar putea produce dificultăţi în exploatarea BD (de exemplu, dacă $r \rhd< s$ este o vedere). Astfel de tupluri se numesc tupluri „dangling". Dacă ele nu sunt considerate se pierde din informaţie. Pentru a rezolva problema se introduce operaţiunea de *joncţiune externă completă*, care se efectuează ca mai jos:

r NATURAL FULL OUTER JOIN s =

A	B	C	D	E	F
a	b	c	d	e	f
m	n	p	q	NULL	NULL
u	v	NULL	NULL	x	y

Dacă se urmăreşte să nu se piardă numai tuplurile „dangling” ale uneia din relațiile r sau s, se introduce operațiunile de joncțiune externă „stângă”, respectiv „dreaptă”:

r NATURAL LEFT OUTER JOIN s =

A	B	C	D	E	F
a	b	c	d	e	f
m	n	p	q	NULL	NULL

r NATURAL RIGHT OUTER JOIN s =

A	B	C	D	E	F
a	b	c	d	e	f
u	v	NULL	NULL	x	y

Notația folosită mai sus pentru operatori aparține standardului SQL2.

Redenumirea schemelor de relație

Fie schema de relație R, cu atributele { A_1, A_2, ..., A_n }. Pentru a schimba numele schemei din R în S, dar cu aceleaşi atribute, folosim operatorul ρ, cu sintaxa (în algebra relaîională): $\rho_{S(A_1, A_2, \ldots, A_n)}(R)$. Rezultatul este exact aceeaşi relație, dar cu numele S.

Operațiuni relaționale pe „multiset”-uri („pungi”)

Multiset = mulțime în care se admit apariții multiple ale unor elemente. Mulțimile convenționale nu conțin duplicate. În practică, relațiile sunt multiset-uri, adică unele tupluri apar de mai multe ori într-o relație. Exemplu:

A	B
1	2
3	4
3	4

Necesitatea multiset-urilor

Primele SGBD care au folosit modelul relațional au conținut limbaje de interogare bazate pe algebra relațională. Aceste sisteme tratau relațiile ca multiset-uri, nu ca mulțimi convenționale, din rațiuni de eficiență. Cu alte cuvinte, dacă utilizatorul nu cerea explicit ca tuplurile duplicat (prezente în realitate) să fie condensate în unul singur, relațiile rezultate puteau conține duplicate. Exemplu:

A	B	C
1	2	9
3	4	5
3	4	7
3	4	1
1	2	6

Dacă se execută proiecția $\Pi_{A,B}(r)$ a relației r pe atributele A şi B obținem:

A	B
1	2
3	4
3	4
3	4
1	2

În ipoteza că se doreşte ca rezultatul proiecției să fie o relație (mulțime) convențională, relația de mai sus trebuie prelucrată în continuare, în scopul eliminării duplicatelor. Această operație reclamă un interval de timp suplimentar. În ipoteza că se admit multiset-uri ca rezultate, operația de proiecție se efectuează mai repede, deoarece nu este necesar ca fiecare tuplu să fie comparat cu alte tupluri generate prin proiecție.

Mai remarcăm un alt aspect al admiterii multiset-urilor. Dacă operația de proiecție a relației r de mai sus este efectuată în scopul de a calcula ulterior un „agregat” – de exemplu valoarea medie a unui atribut – fie el B, vom obține 8:3 în cazul în care condensăm tuplurile duplicat, respectiv 16:5 = 3,2 dacă admitem ca rezultat multisetul.

Reuniunea multiset-urilor

Exemplu:

r

A	B
1	2
3	4
3	4

s

A	B
1	2
1	2
3	4
4	5

r ∪ s

A	B
1	2
3	4
3	4
1	2
1	2
3	4
4	5

Aşadar, dacă un tuplu t este prezent de m ori în r şi de n ori în s, va fi prezent de m+n ori în r ∪ s.

Intersecția multiseturilor

Exemplu – pentru r şi s de mai sus avem:

r ∩ s

A	B
1	2
3	4

În acest caz, tuplul t apare de min(m,n) ori în relația obținută prin intersecție.

Diferența multiset-urilor

Exemplu – pentru aceleaşi relații r şi s de mai sus:

r – s

A	B
3	4

În acest caz, dacă tuplul t apare de m ori în r si de n ori în s, în relația diferență el va fi prezent de max(0, m-n) ori. Cu alte cuvinte, dacă t apare în r de mai multe ori ca în s, atunci el va fi prezent în r-s de un număr de ori egal cu diferența între numărul aparițiilor sale în r şi numărul de apariții din s. Dacă t apare în s cel puțin de acelaşi număr de ori ca în r, atunci acest tuplu nu va fi deloc prezent în r-s. În fapt, putem spune că fiecare apariție a tuplului t în s va anihila o apariție în r.

Proiecția multiset-urilor

Operațiunea a fost deja prezentată.

Selecția pe multiset-uri

Exemplu – pentru relația s definită astfel:

s

A	B
1	2
1	2
3	4
4	5

$\sigma_{B\geq 4}(s)$

A	B
3	4
4	5

Produsul multiset-urilor

Se efectuează exact ca produsul cartezian al relațiilor convenționale, prin concatenare repetată.

r

A	B
1	2
3	4
3	4

s

B	C
4	5
6	7
6	7

r × s

A	r.B	s.B	C
1	2	4	5
1	2	6	7
1	2	6	7
3	4	4	5
3	4	6	7
3	4	6	7
3	4	4	5
3	4	6	7
3	4	6	7

Joncțiunea multiset-urilor

În cazul joncțiunii naturale – pentru r şi s de mai sus obținem:

$r \rhd\lhd s$

A	B	C
3	4	5

$\underset{r.B < s.B}{r \rhd\lhd s}$

A	r.B	s.B	C
1	2	4	5
1	2	6	7
1	2	6	7
3	4	6	7
3	4	6	7
3	4	6	7
3	4	6	7

Operatori extinşi ai algebrei relaţionale

Majoritatea limbajelor de interogare moderne se bazează pe definiţiile operaţiunilor relaţionale, precum şi pe cele privind tratarea multiset-urilor. Totodată, în practica actuală, limbaje de interogare ca SQL permit efectuarea unor operaţii suplimentare, importante în aplicaţii. Dintre acestea, menţionăm:

1. Operatorul δ pentru *eliminarea duplicatelor* – transformă un multiset într-o mulţime , eliminând toate copiile fiecărui tuplu, în afară de una.
2. *Operatori de agregare* – ca suma sau media – nu aparţin algebrei relaţionale. Aceşti operatori sunt folosiţi de operatorul de grupare. Operatorii de agregare se aplică atributelor (coloanelor) unei relaţii; de exemplu, suma unei coloane conduce la valaorea unui număr care este suma tuturor valorilor de coloană.
3. *Gruparea tuplurilor* se realizează tinând seama de valoarea unui atribut sau de valorile mai multor atribute din tuplurile respective. Ca urmare, mulţimea tuplurilor unei relaţii este partiţionată în „grupuri”. În continuare, coloanelor fiecărui grup li se poate aplica operaţia de agregare, ceea ce permite să fie formulate interogări care nu se pot exprima în algebra relaţională clasică. Operatorul de grupare, notat cu γ, combină efectele grupării şi agregării.
4. *Operatorul de sortare* τ transformă o relaţie într-o listă de tupluri, sortate potrivit valorii unuia sau mai multor atribute. Operatorul τ trebuie folosit numai ca un pas final al unei successiuni de operaţiuni, deoarece alţi operatori algebrici relaţionali se aplică numai mulţimilor sau multiset-urilor, dar niciodată listelor.
5. *Proiecţia extinsă* amplifică funcţiile operatorului Π. În forma s-a generalizată, operatorul Π, în afară de proiectarea unor coloane, conduce la efectuarea de calcule asupra coloanelor din relaţia-argument, având ca efect producerea unor coloane noi.
6. *Operatorul de joncţiune esternă* este o variantă a operatorului convenţional, care permite evitarea pierderii tuplurilor „dangling”. În rezultatul joncţiunii externe, tuplurile „dangling” sunt completate („padded”) cu valori nule, astfel încât să poate fi reprezentate în ieşire.

Operatori de agregare

Operatori standard:
SUM – produce suma unei coloane cu valori numerice.
AVG – produce valoarea medie a unei coloane cu valori numerice.
MIN, MAX – produce cea mai mică, respectiv cea mai mare valoare dintr-o coloană cu valori numerice. Dacă se aplică unei coloane cu valori reprezentate de şiruri de caractere, aceşti operatori produc prima sau ultima valoare din punct de vedere lexicografic (alfabetic).
COUNT – dă numărul de valori (nu neapărat distincte) dintr-o coloană. Aplicat oricărui atribut al unei relaţii, operatorul dă numărul de tupluri ale relaţiei, incluzând duplicatele.

Exemplu – fie relaţia r definită astfel:

r

A	B
1	2
3	4
3	4
1	2

1. SUM(B) = 2+4+4+2 = 12
2. AVG(A) = (1+3+3+1)/4 = 2
3. MIN(B) = 2
4. MAX(A) = 3

5. COUNT(A) = 4

Gruparea

Uneori utilizatorul unei baze de date nu doreşte să efectueze o simplă mediere sau altă operaţiune de agregare pe o coloană întreagă. Adesea se urmăreşte studiul tuplurilor unei relaţii în grupuri, constituite ţinând seama de valoarea unora din atribute, iar agregarea se face numai în interiorul fiecărui grup. Exemplu – o BD bancară, cu o relaţie care conţine valorile sumelor retrase de clienţii băncii:

Nume client	Valoare retrasă
Ionescu	5000
Barbu	8000
Ionescu	3000
Vlad	2500
Barbu	1500

Prin operaţia de grupare se obţine:

Nume client	Valoare retrasă
Ionescu	5000
Ionescu	3000
Barbu	8000
Barbu	1500
Vlad	2500

folosind operatorul de grupare γ, cu argumentul NumeClient – de exemplu γ (pe NumeClient), cu o sintaxă adecvată.

Apoi se poate aplica operatorul de agregare SUM în mod independent fiecărui grup: SUM(ValoareRetrasă, pe grupul Ionescu), cu o sintaxă adecvată (SQL are comenzi bine cunoscute).

În cele ce urmează se descrie modul în care lucrează operatorul de grupare.

Operatorul de grupare

Prin acest operator se pot grupa tuplurile unei relaţii sau/şi se pot agrega unele coloane ale acesteia. În cazul grupării, agregarea se efectuează în interiorul grupurilor.

Fie relaţia r. Sintaxa operatorului de grupare este $\gamma_L(r)$ unde L este o listă de elemente. Fiecare din elemenentele din listă paote fi:

a) atributul din relaţia r căruia i se aplică operatorul; este denumit atribut de grupare;
b) un operator de agregare aplicat unui atribut al relaţiei. Rezultatului acestei agregări trebuie să i se dea un nume, corespunzător atributului pe care se face agregarea şi care se numeşte atribut agregat. În acest scop, operaţiunii de agregare i se asociază o săgeată şi un nou nume.

Evaluarea (calculul) expresiei relaţionale $\gamma_L(r)$ întoarce o relaţie, care se construieşte aşa cum se arată în continuare:

1. se partiţionează tuplurile din r în grupuri. Fiecare grup este format din toate tuplurile pentru care atributele de grupare din lista L li se asociază valori particulare. Dacă nu există atribute de grupare, întreaga relaţie r este un grup.
2. pentru fiecare grup se construieşte un tuplu format din:

a) valorile atributelor de grupare pentru acel grup şi
b) agregările pe toate tuplurile acelui grup, pentru atributele agregate din lista L.

Exemplu: fie relaţia r cu numele Edificiu definită astfel: Edificiu(NumeEdificiu, Anul, Arhitect).

Dacă dorim primul an în care un arhitect a conceput un edificiu, referindu-ne însă doar la acei arhitecți care au realizat cel puțin trei edificii, formulăm expresia de interogare:

$$\gamma_{\text{Arhitect, MIN(Anul)}\rightarrow\text{minAnul, COUNT(NumeEdificiu)}\rightarrow\text{cneNumeEdificiu}}(\text{Edificii})$$

În expresia de mai sus se efectuează mai întâi o operație de grupare, folosind Arhitect ca atribut de grupare. Evident că pentru fiecare grup obținut trebuie să calculăm agregatul $\text{MIN}_{(\text{Anul})}$. De asemenea, pentru a decide care grup satisface condiția potrivit căreia Arhitect-ul a realizat cel puțin trei edificii, trebuie să fie calculat, de asemenea, agregatul COUNT(NumeEdificiu) pentru fiecare grup.

În expresia de grupare formulată mai sus, primele două coloane alre rezultatului sunt necesare pentru rezultatul interogării. A treia coloană este un atribut auxiliar, denumit cne, necesar pentru a determina dacă un arhitect a realizat cel puțin trei edificii. Prin urmare, vom continua expresia algebrică ce corespunde interogării selectând cazurile cu cneNumeEdificiu ≥ 3 şi apoi efectuând o proiecție pe primele două coloane. Mai jos este prezentat arborele sintactic al interogării:

$$
\begin{array}{c}
\Pi_{\text{Arhitect,minAnul}} \\
| \\
\sigma_{\text{cneNumeEdificiu}\geq 3} \\
| \\
\gamma_{\text{Arhitect, MIN(Anul)}\rightarrow\text{minAnul, COUNT(NumeEdificiu)}\rightarrow\text{cneNumeEdificiu}} \\
| \\
\text{Edificii}
\end{array}
$$

Arhitect	NumeEdificiu	Anul
Doicescu	Politehnica	1968
Antonescu	Primăria	1928
Doicescu	Opera	1959
-----------------------	---------------------------------	------------------
Antonescu	Facultatea de Drept	1936
Doicescu	Restaurantul Băneasa	1938
-----------------------	---------------------------------	------------------
Antonescu	Banca de investiții	1937
Doicescu	Centrala Banu Marta	1938
-----------------------	---------------------------------	------------------

Dacă facem grupare pe Arhitect obținem (γ_{Arhitect}):

Arhitect	NumeEdificiu	Anul
Antonescu	Primăria	1928
Antonescu	Facultatea de Drept	1936
Antonescu	Banca de investiții	1937
Doicescu	Politehnica	1968
Doicescu	Opera	1959
Doicescu	Centrala Banu Marta	1938
Doicescu	Restaurantul Băneasa	1938
-----------------------	---------------------------------	------------------

Dacă pe relația de mai sus aplicăm operatorul de agregare COUNT pe NumeEdificiu – va număra edificiile realizate de fiecare arhitect – şi MIN pe Anul – ia anul cel mai mic – vom obține în locul fiecărui grup de tupluri doar un un singur tuplu, cu valorile agregate ($\gamma_{\text{Arhitect, COUNT(...), MIN(...)}}$):

Arhitect	COUNT(...)	MIN(...)
Antonescu	3	1928
Doicescu	4	1938

Şi apoi, prin lista L, se pot da nume noi ultimelor două coloane, etc.

Extinderea operatorului de proiecție

Algebra relațională clasică: $\Pi_L(r)$, unde L este lista de atribute a relației r. În proiecția extinsă, notată tot cu $\Pi_L(r)$, lista L poate conține următoarele elemente:

1. un singur atribut din r.
2. o expresie $x \rightarrow y$, unde x şi y sunt nume de atribute. Semnul expresiei $x \rightarrow y$: atributul x din r, pe care se face proiecția, primeşte numele nou y în rezultat.
3. O expresie $E \rightarrow z$, unde E este o expresie care conține atribute din r, constante, operatori aritmetici, iar z este un nume nou, care se asociază atributului rezultat din calculul expresiei E. De exemplu: $a+b \rightarrow x$ în L semnifică însumarea valorilor atributelor a şi b, iar sumei i se dă numele x. Dacă c şi d sunt atribute – şiruri de caractere – expresia $c||d \rightarrow e$ semnifică o concatenare între c şi d, iar rezultatului i se dă numele e.

Rezultatul proiecției se calculează luând în considerare fiecare tuplu din r, pe rând. În final, se obține o nouă relație, a cărei schemă este formată de nume ale atributelor din r, precum şi de nume noi, obținute prin redenumire. Fiecare tuplu din r dă un tuplu în rezultat. Tuplurile duplicat din r dau tupluri în rezultat. Menționăm însă că rezultatul poate conține tupluri duplicate chiar şi în cazul în care r nu a anulat astfel de tupluri. Exemple:

r

A	B	C
1	0	3
2	4	6
3	5	1

$\mathbf{\Pi_{A,B+C \rightarrow x}}$

A	x
1	3
2	10
3	6

r

A	B	C
1	0	3
2	4	6
3	5	1
3	5	7

$\mathbf{\Pi_{B-A \rightarrow x,C-B \rightarrow y}}$

x	y
-1	3
2	2
2	-4
2	2

Operatorul de sortare

Notația: $\tau_L(r)$, unde r este o relație, iar L o listă care conține unele din atributele schemei de relație r. Expresia $\tau_L(r)$, prin evaluare, conduce la sortarea tuplurilor din r în ordinea în care atributele considerate – pentru sortare – sunt prezente în lista L.

Dacă lista L are forma $A_1, A_2, ..., A_n$, atunci tuplurile din r vor fi mai întâi sortate potrivit valorilor atributului A_1 (numerice sau lexicografice). „Legăturile sunt rupte” potrivit cu valorile atributului A_2; tuplurile care conțin aceleaşi valori pentru A_1 şi A_2 sunt ordonate după valorile atributului A_3 ş.a.m.d. „Legăturile” care rămân după ce a fost condenat atributul A_n pot fi ordonate arbitrar. Mai sus – „legături” este traducerea de la „ties”. În contextul acestei traduceri, „ties” ar putea să însemne „tupluri”.