

Capitolul 4. Limbaje de interogare. Algebra relațională

Dacă un model de date are, potrivit lui Codd, trei axe – *structură* (modalitatea de organizare a datelor în bază), *integritate* (restricții) și *manipulare* (operații), până în acest capitol discuția s-a purtat la nivelul primelor două. Este momentul să trecem la partea cea mai importantă a cărții de față – modalitățile prin care se pot extrage informații dintr-o bază de date.

În materie de limbaje dedicate bazelor de date, SQL-ul s-a instalat, de aproape trei decenii, confortabil în postura de lider absolut. Lucrurile nu au fost atât de clare de la început, peisajul limbajelor de interogare fiind mult mai bogat decât ne-am fi imaginat. Mai mult, se spune uneori că elementul cheie în impunerea SQL-ului în detrimentul altor limbaje, cum ar Quel, QBE, nu a fost forța acestuia, ci faptul că a fost susținut de cine trebuia (IBM, la care s-a adăugat ulterior Oracle), plus faptul că s-a reușit elaborarea unor standarde unanim acceptate.

Inițial, pentru exprimarea operațiilor aplicabile structurilor de date, E.F.Codd, părintele relaționalului, a definit un limbaj de manipulare a datelor puternic matematizat, bazat pe teoria ansamblurilor (seturilor), numit DSL/Alpha¹. Tot Codd pune bazele algebrei relaționale – limbaj tratat în continuare – care, deși teoretică, constituie fundamentul SQL-ului, chiar dacă astăzi puțini mai sunt interesați de această paternitate. Am putea spune că SQL-ul este, în oarecare măsură, fiul deopotrivă rătăcitor și ne(re)cunoscător al algebrei relaționale.

4.1. Caracterizare generală a limbajelor de interogare

Până la consacrarea SGBD-urilor relaționale, extragerea informațiilor dorite dintr-o bază de date se realiza prin aplicații dezvoltate exclusiv cu limbaje procedurale, în care trebuie definite atât datele dorite, cât și metodele de căutare și extragere ale acestora. Succesul SGBD-urilor ar fi fost de neconceput fără elaborarea și implementarea unor limbaje performante pentru manipularea BD - limbajele de interogare.

Limbajele relaționale sunt neprocedurale: utilizatorul definește datele ce trebuie extrase din BD, sarcina căutării și extragerii fiind "rezervată" exclusiv SGBD-ului. De asemenea, în unele lucrări, acestea sunt referite ca limbaje închise, deoarece o consultare generează o nouă relație ce poate fi utilizată, la rândul său, ca argument în alte consultări s.a.m.d.

¹ Vezi și [Date1999-1]

Pornind de la cele două modalități de definire a relației, pe de o parte, ca predicat aplicat asupra unor domenii și, pe de altă parte, ca ansamblu de tupluri, limbajele de manipulare a datelor sunt grupate în două mari categorii, limbaje *predicative* - fondate pe teoria predicatelor și limbaje *ansambliste* - fondate pe teoria ansamblurilor (tuplurilor). Limbajele predicative sunt divizate în alte două sub-clase, cele care au la bază *calculul relațional asupra tuplurilor* (limbaje orientate pe tupluri) și cele în care *calculul relațional se aplică asupra domeniilor* (limbaje orientate pe domenii). Elementul definitoriu pentru limbajele de manipulare bazate pe calculul predicatelor îl reprezintă noțiunea de variabilă, care poate fi asociată fie tuplurilor, fie domeniilor.

O altă grupare delimitează limbajele *non-grafice* de cele *grafice*. Primele permit reprezentarea unei consultări "în linie", prin dispunerea succesivă a operatorilor, atributelor și relațiilor. Cele grafice permit redactarea consultării în mod interactiv, prin afișarea pe ecran a unui sistem de meniuri și elemente de dialog din care opțiunile pot fi selectate și modificate cu ajutorul mouse-ului (sau claviaturii); în plus, se mai poate opera o delimitare și pentru limbajele grafice în funcție de utilizarea explicită sau implicită a variabilelor de domeniu. Deși nu exagerat de recentă, o clasificare încă valabilă (în parte) a limbajelor relaționale poate fi prezentată ca în figura 4.1².

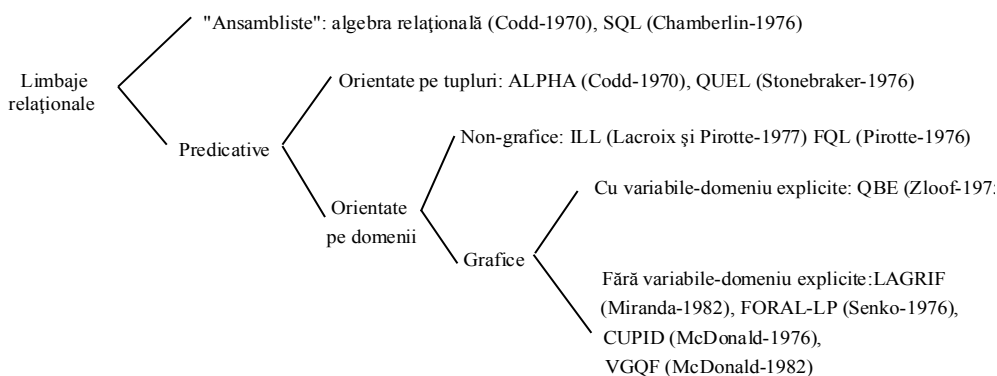


Figura 4.1. O clasificare (mai veche) a limbajelor relaționale

Există o serie de caracteristici comune tuturor limbajelor:

- operatorii relaționali se aplică relațiilor luate în întregime, adică tuturor tuplurilor care alcătuiesc relațiile respective;
- rezultatul fiecărui operator (rezultatul consultării) este o nouă relație ce poate servi ca argument într-o altă consultare s.a.m.d.;
- logica operatorilor se bazează pe valorile atributelor, ceea ce constituie de altminteri suportul singurului mod de acces în BD; întrucât consultările mono și multi-relații sunt efectuate exclusiv prin compararea valorilor

² [Miranda&Busta 1990], vol.2, p.30

atributelor (definite pe domenii compatibile), accesul total independent de limbaj este asigurat.

Înainte de redactarea unei consultări într-un limbaj relațional trebuie parcursă o fază de *analiză*, pentru determinarea atributelor rezultatului, legăturilor dintre tabele și eventualele condiții (restricții) ce trebuie respectate.

Algebra și calculul relațional servesc drept puncte de referință în caracterizarea unui limbaj ca fiind *complet* sau *incomplet*, din punct de vedere relațional. Dacă un limbaj permite exprimarea tuturor operatorilor enumerați mai jos, oferind măcar facilitățile algebrei relaționale, se spune că acesta este un limbaj relațional complet³.

4.2. Operatori ai algebrei relaționale

Limbajul algebric relațional cuprinde două tipuri de operatori: *ansambliști* - REUNIUNE, INTERSECȚIE, DIFERENȚĂ, PRODUS CARTEZIAN - și *relaționali* - SELECȚIE, PROIECȚIE, JONCȚIUNE și DIVIZIUNE.

O altă clasificare distinge operatorii *fundamentali*, ireductibili (reuniunea, diferența, produsul cartezian, selecția, proiecția) de cei *derivați*, a căror funcționalitate poate fi realizată prin combinarea operatorilor fundamentali (intersecția, joncțiunea, diviziunea).

Pentru cele ce urmează se notează cu:

- t sau r , un *tuplu* al unei relații (linie a unei tabele) și
- $t(A)$, un *sub-tuplu* al relației R , relativ la atributul A (valoarea atributului A în linia t).

Trei dintre operatorii ansambliști - *reuniunea* (" \cup "), *intersecția* (" \cap ") și *diferența* (" $-$ ") - pot opera numai cu două relații *uni-compatibile*. Fie $R_1 (A_1, A_2, \dots, A_n)$ și $R_2 (B_1, B_2, \dots, B_m)$ două relații. R_1 și R_2 sunt unicompatibile dacă: (a) $n = m$; (b) $\forall i \in \{1, 2, \dots, n\}$, A_i și B_i sunt de același tip sintactic (aceasta nu înseamnă că trebuie să prezinte, neapărat, domenii identice de definire). Relațiile R_1 și R_2 din figura 5.1 sunt unicompatibile deoarece: (a) ambele au același număr de atribute; (b) atributele A, B, C din R_1 (le putem nota și $R_1.A, R_1.B, R_1.C$) corespund sintactic (sunt de același tip) atributelor C, D și E din R_2 ($R_2.C, R_2.D, R_2.E$).

R1			R2		
A	B	C	C	D	E
20	XYZ	30	25	XYZ	30
30	XXZ	20	40	YYX	25
40	YYX	25	30	XXZ	40

Figura 4.2. Două relații unicompatibile

³ [Saleh 1994], p.35

4.3.Operatorii ansambliști

După cum le spune și numele, acești patru operatori sunt preluați din teoria seturilor (mulțimilor). Sunt relativ ușor de înțeles, însă aria lor de folosire este una limitată în practică.

4.3.1. Reuniunea

Reuniunea a două relații unicompatibile, $R1$ și $R2$, este definită astfel: $R1 \cup R2 = \{ \text{tuplu } t \mid t \in R1 \text{ sau } t \in R2 \}$. Se notează: $R3 \leftarrow R1 \cup R2$. Conținutul tabelului reuniune $R3$ este prezentat în figura 4.3. Primul tuplu, al treilea și ultimul din rezultat sunt preluate din $R1$, iar al doilea și al patrulea din $R2$. $R3$ are numai cinci tupluri deoarece un tuplu este comun tabelor $R1$ și $R2$. Algebra relațională elimină automat dublurile (tuplurile identice), astfel încât restricția de unicitate este asigurată după orice operație.

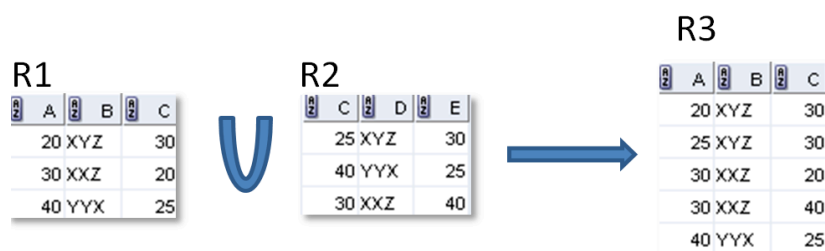


Figura 4.3. Reuniunea a două relații

Există suficiente situații informaționale care fac uz de reuniunea a două tabele. Să luăm două exemple:

1. Tabelele ce reflectă tranzacții economice pot fi descompuse (sparte) în funcție de anul (luna sau cincinalul, deceniul) la care se referă; dacă ne raportăm la baza noastră de date, tabela LINIIFACT poate fi ruptă în alte două, LF_2000_2001 care ar conține facturile emise în anii 2000 și 2001 și LINIIFACT ce conține numai înregistrări aferente anului calendaristic 2002. Începând cu 2002, orice situație statistică privind vânzările în perioada 2000-2002, 2000-2003 etc. necesită reuniunea celor două tabele, LF_2000_2001 și LINIIFACT.

2. Pentru a afla care sunt clienții care au cumpărat cel puțin unul din produsele *Produs 1* și *Produs 2*, se poate proceda la reuniunea tabelii ce conține clienții care au cumpărat *Produs 1* cu tabela clienților care au cumpărat *Produs2*.

Reuniunea este comutativă. Singura problemă neclară ar fi legată de numele atributelor în relația rezultat. În acest sens, se poate institui regula potrivit căreia numele atributelor relației-reuniune sunt numele primei relații participante în operație. Aceasta nu are importanță asupra comutativității, deoarece conținutul tabelii-rezultat este identic, indiferent care este prima relație enumerată.

Ordinea în care apar tuplurile în rezultat nu are nici o importanță, din punctul de vedere al conținutului informațional. În figura 4.3 liniile sunt ordonate după

valorile celor trei atribute. Algebra relațională nu garantează în nici un fel maniera de dispunere a tuplurilor în rezultatul unei operațiuni.

4.3.2. Intersecția

Intersecția a două relații uni-compatibile, $R1$ și $R2$, poate fi definită astfel: $R1 \cap R2 = \{\text{tuplu } t \mid t \in R1 \text{ și } t \in R2\}$. Se notează: $R4 \leftarrow R1 \cap R2$. Conținutul tabelului intersecție $R4$ este prezentat în figura 4.4. Cum numai un tuplu este absolut identic și în $R1$ și $R2$, tabela rezultat este alcătuită dintr-o singură linie.

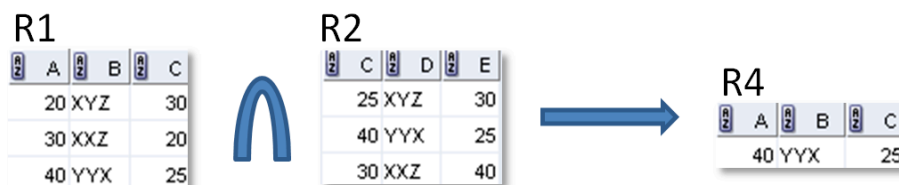


Figura 4.4. Intersecția a două relații

Exemple de informații care fac necesară recurgerea la intersecție:

- Pentru a afla care sunt clienții care au cumpărat și *Produs 1* și *Produs 2*, se poate proceda la intersecția tabelului clienților care au cumpărat *Produs 1* cu tabela alcătuită din clienții care au cumpărat *Produs2*;
- Zilele în care s-au făcut vânzări și clientului *Client 1 SRL* și clientului *Client 2 SA*;
- Persoanele de la firmele-client care cumulează posturile de *Director vânzări* și *Șef aprovizionare*.

Ca și reuniunea, intersecția este comutativă, iar numele atributelor relației-intersecție sunt extrase din prima relație participantă în operație.

4.3.3. Diferența

Diferența a două relații uni-compatibile, notate $R1$ și $R2$, este definită astfel: $R1 - R2 = \{\text{tuplu } t \mid t \in R1 \text{ și } t \notin R2\}$. Se notează: $R5 \leftarrow R1 - R2$. Conținutul tabelului-diferență $R5$ (figura 4.5) conține numai tuplurile din prima relație, $R1$, care nu se regăsesc în a doua relație, $R2$.

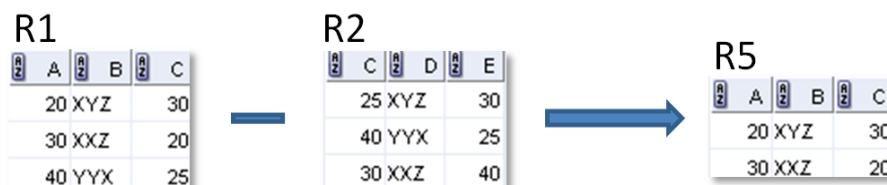


Figura 4.5. Diferența a două relații

Așadar, din rezultat este eliminat al treilea tuplu din $R1$, deoarece valorile acestuia există și în $R2$ (al doilea tuplu din $R2$). Exemple ce informații care fac necesară recurgerea la diferență:

- Care sunt clienții care au cumpărat *Produs 1* dar nu au cumpărat *Produs 2* ?
- Care sunt zilele în care s-au făcut vânzări clientului *Client 1 SRL* dar nu există nici o factură către *Client 2 SA* ?

Spre deosebire de reuniune și intersecție, diferența nu este comutativă. Atributelor relației-diferență sunt cele ale primei relații (descăzutul), iar tuplurile sunt extrase din relația descăzut nu se regăsesc în relația scăzător. În plus, nu există restricții privind cardinalitatea (numărul de tupluri) celor două relații, adică nu este musai ca relația descăzut să conțină mai multe tupluri decât cea scăzător.

4.3.4. Produsul cartezian

Produsul cartezian al două relații R_1 și R_2 , denumit de Codd joncțiune încrucișată (CROSS JOIN), este ansamblul tuturor tuplurilor obținute prin concatenarea fiecărei linii din tabela R_1 cu toate liniile tabelului R_2 . Formal, dacă notăm cele două relații: $R_1 (A_1, A_2, \dots, A_n)$ și $R_2 (B_1, B_2, \dots, B_m)$, produsul cartezian este definit astfel: $R_1 \otimes R_2 = \{ (t_1, t_2) \mid t_1 \in R_1 \text{ și } t_2 \in R_2 \}$. Spre deosebire de celelalte trei operațiuni precedente, produsul cartezian nu face apel la noțiunea de relații uni-compatibile, iar relația rezultat cumulează attributele celor două relații-argument. În figura 4.6 este ilustrat rezultatul produsului cartezian a tabelurilor R_1 și R_2 .

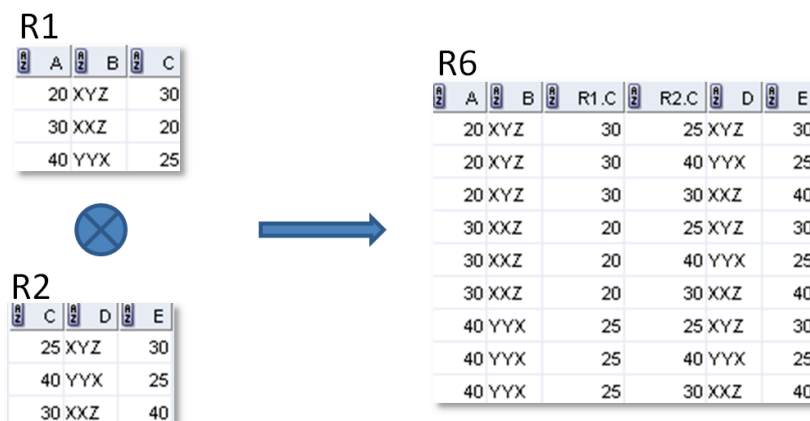


Figura 4.6. Produsul cartezian

Se notează: $R_6 \leftarrow R_1 \otimes R_2$. Tabela rezultat R_6 are o nouă structură - șase attribute (trei preluate din R_1 și trei din R_2). Întrucât există un atribut cu nume comun, C, pentru a diferenția cele două apariții, acestea sunt prefixate, în antetul tabelului, cu numele relației din care provine. Prima linie din R_6 este obținută prin "lipirea" primului tuplu din R_1 cu primul tuplu din R_2 , a doua din primul tuplu din R_1 cu al doilea din R_2 etc. Cum R_1 are 3 tupluri, iar R_2 tot 3, relația-rezultat al produsului cartezian are $3 * 3 = 9$ tupluri.

Nu prea există situații care să reclame folosirea directă și exclusivă a produsului cartezian. Cel mai important "merit" al acestuia în algebra relațională

este că permite "alipirea" a două relații, fundamentând astfel operatorul-cheie care este joncțiunea.

4.4.Operatorii relaționali

Cei patru operatori prezentați în paragraful precedent sunt generali, spre deosebire de următorii care sunt specifici algebrei relaționale. De obicei gruparea operatorilor relaționali se face astfel: (a) *operatori unari de restricție*, care permit decupajul unei relații, pe orizontală - **SELECTȚIA** și pe verticală - **PROIECȚIA**; (b) *operatori binari de extensie*: **JONCȚIUNEA** și **DIVIZIUNEA**. O altă deosebire majoră față de paragraful precedent este că vom putea recurge și la exemple concrete din baza de date "cobai" prezentată în capitolul anterior.

4.4.1. Selecția

Selecția triază dintr-o relație (tabelă) numai tuplurile ce satisfac o condiție specificată printr-un predicat. Ca preambul, definim noțiunea de formulă F asupra unei relații R , ca o expresie logică compusă din: *operandi* care sunt nume de atribute sau constante; *operatori de comparație aritmetică*: $>$, \geq , $<$, \leq , $=$, \neq ; *operatori logici*: $\&$, \vee , \neg , \rightarrow , \leftrightarrow . Selecția unei relații R , printr-o condiție F , notată $SF(R)$, poate fi definită: $SF(R) = \{ \text{tuplu } t \mid (t \in R \text{ și } F(t) = \text{adevărată}) \}$. O notație ceva mai pământească este: **R1 ←SELECTȚIE (R; <expresie-logică>)**, dar, de dragul științei, vom detalia:

- R este relația $R(A_1, A_2, \dots, A_n)$ asupra căreia se aplică selecția (A_i sunt atributele sale).
- $R1$ este noua relație obținută în urma selecției, care va avea aceeași schemă relațională cu R - $R1(A_1, A_2, \dots, A_n)$.
- $\langle \text{expresie-logică} \rangle$ poate fi scrisă mai analitic astfel:
 $\langle \text{expresie-logică} \rangle = (\text{termen}_1) \text{ și/sau } (\text{termen}_2) \dots \text{ și/sau } (\text{termen}_k)$, unde
 - $\text{termen}_j = \text{expresie}_1 \text{ } \emptyset \text{ expresie}_2$.
 - expresie_1 sau expresie_2 sunt expresii calculate plecând de la atributele A_i .
 - \emptyset poate fi unul dintre operatorii pentru comparație.

Exemplu 1: Pentru a pune în operă savanta notație de mai sus, luăm în discuție o primă problemă: *Care sunt liniile din R1 pentru care valorile atributelor A și C sunt mai mari decât 20 ?* Ajungem, astfel, la notația:

R ←SELECTȚIE (R1; A > 20 AND C > 20)

Tabela R este prezentată în figura 4.7.

A	B	C
40	YYX	25

Figura 4.7. Rezultat selecție – exemplul 1

Exemplu 2: Care sunt județele din Moldova ?

Mai întâi, se identifică în baza de date tabela (sau tabelele) din care se extrage rezultatul. În acest exemplu, aceasta este JUDETE. Apoi se stabilesc atributele (atributul) asupra cărora se va aplica predicatul de selecție. Se obține:

R ←SELECȚIE (JUDETE; Regiune = 'Moldova') – vezi fig. 4.8.

JUD	JUDET	REGIUNE
IS	Iasi	Moldova
VN	Vrancea	Moldova
NT	Neamt	Moldova
SV	Suceava	Moldova
VS	Vaslui	Moldova

Figura 4.8. Rezultat selecție – exemplul 2

Exemplu 3: Care sunt facturile emise în perioada 2-5 august 2000 ?

Tabela în care va opera operatorul de selecție este FACTURI. Predicatul de selecție utilizează atributul DataFact:

**R ←SELECȚIE (FACTURI; DataFact >= '2007-08-02' AND
DataFact <= '2007-08-05').**

NRFAC	DATAFACT	CODCL	OBS
1115	02-08-2007	1001	(null)
1116	02-08-2007	1007	Pretul propus initial a fost modificat
1117	03-08-2007	1001	(null)
1118	04-08-2007	1001	(null)

Figura 4.9. Rezultat selecție – exemplul 3

4.4.2. Proiecția

Prin proiecție, o relație poate fi "decupată" pe verticală. Dacă selecția extrage dintr-o tabelă anumite linii, pe baza condiției îndeplinite de valorile unora dintre atribute, *proiecția* permite selectarea într-o tabelă-rezultat numai a coloanelor (atributelor) dorite dintr-o relație. Formal, fie o relație $R(A_1, A_2, \dots, A_n)$.

Proiecția relației R asupra unui subansamblu alcătuit din atribute proprii este o relație care se obține după parcurgerea a doi pași:

- eliminarea dintre A_i a acelor atribute care nu sunt specificate;
- suprimarea dublurilor (tuplurile identice).

Se notează: $R_1 \leftarrow \text{PROIECȚIE}(R; A_j, A_k, \dots, A_x)$. Spre deosebire de R , schema relației R_1 este alcătuită numai din atributele indicate: $R_1(A_j, A_k, \dots, A_x)$. Dacă după extragerea coloanelor nu există tupluri (linii) identice, R_1 va avea același număr de linii ca și relația R . În caz contrar, numărul lor va fi mai mic, în funcție de numărul dublurilor.

Exemplu 4: Începem, ca de obicei, cu un exemplu ceva mai arid - Care sunt valorile combinației atributelor A și C în relația R_1 ?

$R \leftarrow \text{PROIECȚIE}(R_1; A, C)$. Tabela R are două coloane, A și C , și trei linii, ca în figura 4.10.

A	C
20	30
30	20
40	25

Figura 4.10. Rezultat proiecție – exemplul 4

Exemplu 5: Ce regiuni ale țării sunt preluate în bază ?

Tabela în care se află răspunsul este JUDETE. Singura coloană care interesează este Regiune:

$R \leftarrow \text{PROIECȚIE}(\text{JUDETE}; \text{Regiune})$.

În primul pas, se face decupajul pe verticală, obținându-se o relație notată R' , apoi se elimină dublurile, rezultatul final fiind relația R .



Figura 4.11. Rezultat proiecție – exemplul 5

Exemplu 6 - Care sunt: codul, denumirea și numărul de telefon ale fiecărui client ?

Tabela care interesează este CLIENTI, din care se decupează trei coloane - CodCl, DenCl și Telefon.

$R \leftarrow \text{PROIECȚIE}(\text{CLIENTI}; \text{CodCl}, \text{DenCl}, \text{Telefon})$

CODCL	DENCL	TELEFON
1001	Client 1 SRL	(null)
1002	Client 2 SA	032-212121
1003	Client 3 SRL	035-222222
1004	Client 4	(null)
1005	Client 5 SRL	056-111111
1006	Client 6 SA	(null)
1007	Client 7 SRL	056-121212

Figura 4.12. Rezultat proiectie – exemplul 6

4.4.3. Înlănțuirea consultărilor

Rezultatul unei consultări este o relație (tabelă) nouă. Pe baza acestui fapt, se pot înlănțui două sau mai multe operațiuni, redactându-se astfel interogări complexe.

Exemplu 7 - Care este numărul de telefon al clientul Client 2 SA ?

Soluția este una foarte simplă. Cu ajutorul selecției se decupează din relația CLIENTI numai linia corespunzătoare clientului “incriminat”. Se obține o relație nou-nouță denumită (de noi) R'. Asupra lui R' se aplică o proiectie, deoarece interesează numai numărul de telefon; astfel, R conține răspunsul la problema luată în discuție (vezi figura 4.13).

R' ← SELECTIE (CLIENTI; DenCl = 'Client 2 SA')

R ← PROIECTIE (R'; Telefon)

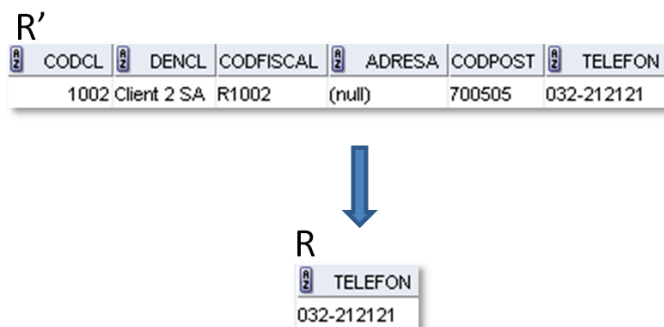


Figura 4.13. Înlănțuirea unei selecții cu o proiectie – exemplul 7

Exemplu 8 - Care sunt denumirile și codurile poștale introduse pentru localitățile (prezente în bază) din județele Iași (IS) și Vrancea (VN) ?

Tabela “interogată” este CODURI_POSTALE. Pentru a răspunde la întrebarea pe care tot noi am formulat-o, putem alege între următoarele două soluții:

Soluția 1 – figura 4.14:

R11 ← **SELECȚIE** (CODURI_POSTALE; Jud = 'IS' OR Jud = 'VN')
R ← **PROIECȚIE** (R11; Loc, CodPost)

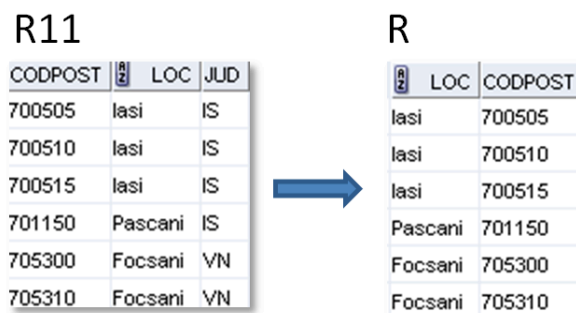


Figura 4.14. Exemplu 8 – soluția 1

Soluția 2 - figura 4.15:

R11 ← **SELECȚIE** (CODURI_POSTALE; Jud = 'IS')
R12 ← **PROIECȚIE** (R11; Loc, CodPost)
R13 ← **SELECȚIE** (CODURI_POSTALE; Jud = 'VN')
R14 ← **PROIECȚIE** (R13; Loc, CodPost)
R ← **R12** ∪ **R14**

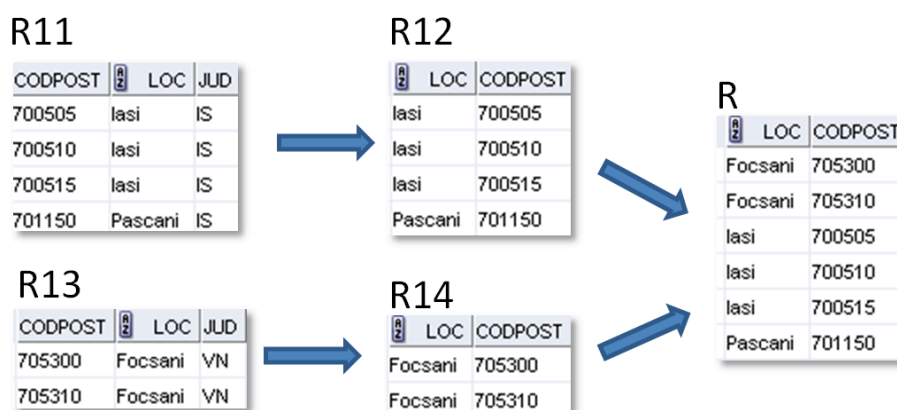


Figura 4.15. Exemplu 8 – soluția 2

Prima soluție este, prin simplitate, cea mai tentantă. Este, însă, un prim caz în care pentru rezolvarea unei probleme pot fi formulate două (sau mai multe) soluții. Îmi fac (cu plăcere) datoria de a vă bate la cap că, deși ordinea liniilor este diferită, rezultatele finale (tabelele R) din figurile 4.14 și 4.15 sunt identice din punctul de vedere al conținutului informațional.

Exemplu 9 - Care sunt codurile produselor care apar deopotrivă în factura 1111 și în factura 1117 ?

Tabela din care vor fi extrase datele este LINIIFACT. Soluția se bazează pe intersecția relației care conține produsele prezente în factura 1111 (R2) cu relația produselor prezente în factura 1117 (R4), după cum reiese din figura 4.16.

R21 ← **SELECȚIE** (LINIIFACT; NrFact = 1111)

R22 ← **PROIECȚIE** (R21; CodPr)

R23 ← **SELECȚIE** (LINIIFACT; NrFact = 1117)

R24 ← **PROIECȚIE** (R23; CodPr)

R ← **R22** ∩ **R24**

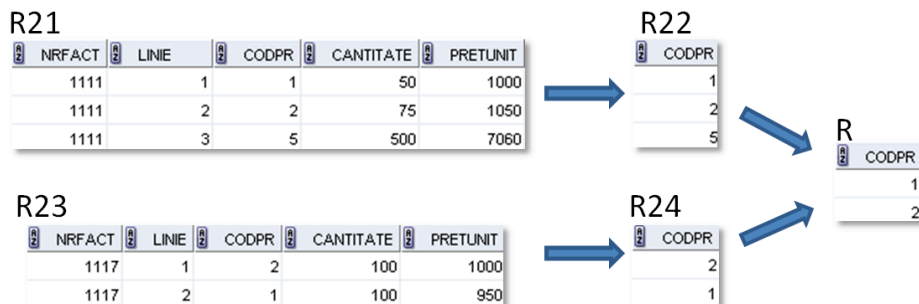


Figura 4.16. Soluție - exemplu 9

4.4.4. Joncțiuni interne

Într-un paragraf anterior am văzut că produsul cartezian permite fuzionarea a două tabele într-o tabelă mamut ce conține toate atributele și liniile obținute prin combinarea fiecărui tuplu dintr-o relație cu fiecare tuplu din cealaltă. Tot atunci ne exprimam regretul sincer că operatorul produs cartezian nu poate fi folosit, de unul singur, în interogări, dar că acest "neajuns" va fi compensat din plin de operatorul derivat *joncțiune*.

Dacă produsul cartezian este o fuziune necondiționată a două tabele, joncțiunea reprezintă fuziunea a două relații care au o proprietate comună. Fie două relații, notate: $R1 (A1, A2, \dots, An)$ și $R2 (B1, B2, \dots, Bp)$. Fie Ai și Bj două atribute definite pe același domeniu și θ ansamblul operatorilor de comparație: $\{=, >, \geq, <, \leq, \neq\}$, ce pot fi aplicați celor două atribute Ai și Bj . Joncțiunea relației $R1$, prin Ai , cu relația $R2$, prin Bj , notată $R1 (Ai \theta Bj) R2$ sau $R1 \bowtie R2$, este relația ale cărei tupluri sunt obținute prin concatenarea fiecărui tuplu al relației $R1$ cu tuplurile relației $R2$, pentru care este verificată condiția θ instituită între Ai și Bj . $R1 (Ai \theta Bj) R2 = \{t \mid t \in R1 \otimes R2 \text{ și } t(Ai) \theta t(Bj)\}$.

Joncțiunea internă este echivalenta unui produs cartezian urmat de o selecție. Joncțiunea internă definită mai sus este referită în lucrările de specialitate ca *theta-joncțiune*. În lucrul cu BDR se utilizează cu precădere *echi-joncțiunea*, ce reprezintă un caz particular al theta-joncțiunii, atunci când θ este operatorul de egalitate ("="). Formal, echijoncțiunea se definește astfel: $R1 (Ai = Bj) R2 = \{t \mid t \in R1 \otimes R2 \text{ și } t(Ai) = t(Bj)\}$.

= $t(B_j)$. Apelăm și la o altă notație, mai ușor de reprezentat și suficient de inteligibilă.

R \leftarrow ECHI-JONCTIUNE (R1, R2; $A_i=B_j$)

Exemplu 10 – Theta-jonctiune

Începem exemplificările cu aceleași două tabele folosite din precedentul paragraf, R1 și R2. Rezultatul jonctiunii (theta-jonctiunii) exprimată prin expresia: **R** \leftarrow JONCTIUNE (R1, R2; $R1.A \geq R2.E$) va fi obținut în doi pași, după cum este descris în figura 4.17.

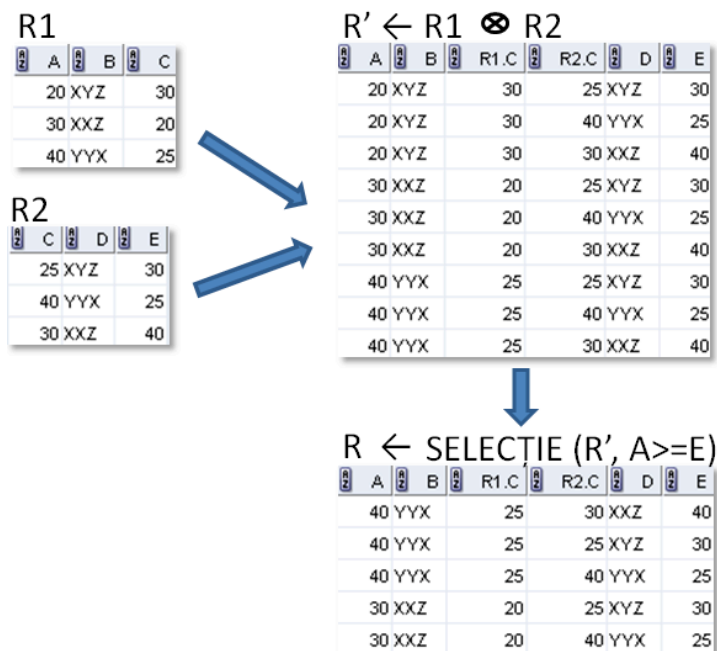


Figura 4.17. Mecanismul de (theta)jonctiune - exemplul 10

Exemplu 11 – Echi-jonctiune

Operatorul de comparație dintre cele două atribute este, obligatoriu, semnul de egalitate.

R \leftarrow JONCTIUNE (R1, R2; $R1.A = R2.E$)

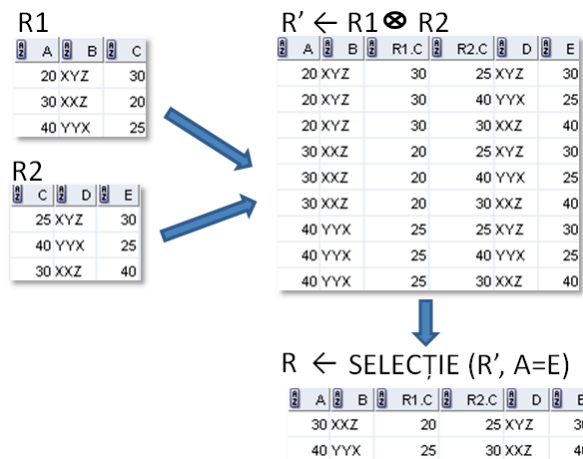


Figura 4.18. Echi-joncțiune - exemplul 11

Exemplu 12 – Joncțiune naturală

Joncțiunea naturală presupune nu numai ca operatorul de comparație să fie semnul de egalitate, ci și denumirea identică a atributelor de legătură dintre cele două tabele. $R \leftarrow \text{JONCȚIUNE}(R1, R2; R1.C = R2.C)$. Datorită faptului că ambele attribute au același nume, se poate considera că tabela rezultat păstrează numai unul din cele două attribute, ca în figura 4.19 și se poate recurge la o notație simplificată: $R \leftarrow \text{JONCȚIUNE}(R1, R2; C)$.

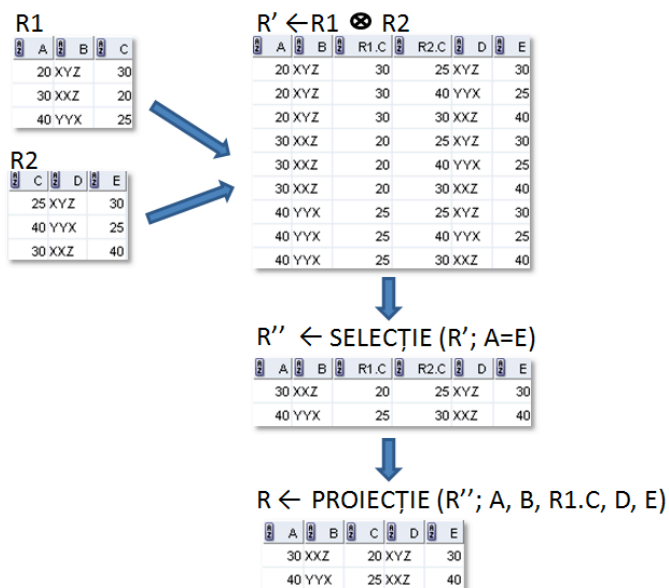


Figura 4.19. Joncțiune naturală - exemplul 12

De fapt, joncțiunea naturală poate fi privită ca o proiecție pe echi-joncțiune, din rezultatul echi-joncțiunii selectându-se toate atributele, cu excepția unui "rând" din atributele după care se face joncționarea.

De ce se insistă atât de mult pe importanța joncțiunii? În primul rând pentru că permite recompunerea relației universale inițiale. Modelul relațional se bazează pe spargerea bazei în relații, astfel încât nivelul redundanței datelor și problemele la actualizarea tabelelor să fie reduse la minim. Cele mai multe interogări, însă, operează cu date și predicate aplicate simultan atributelor din două sau mai multe tabele. Cum selecția este un operator *unar* (poate fi aplicată unei singure relații), este necesară fuzionarea prealabilă a celor două, trei... relații și obținerea unei relații-agregat, relație agregat la care se aplică predicatul suplimentar de selecție.

Fuzionarea este posibilă prin joncțiune. Prin joncționarea tuturor relațiilor dintr-o bază de date se obține relația *universală* (cea inițială, atotcuprinzătoare). Nu ne permitem să reconstituim structura și conținutul relației universale ale bazei de date VINZARI, însă, prin exemple, vom încerca să demonstrăm utilitatea joncțiunii.

Exemplu 13 - Să se obțină, pentru fiecare localitate, codurile poștale, denumirea, indicativul județului, denumirea județului și regiunea din care face parte.

Practic, tabeli CODURI_POSTALE îi trebuie "alipite" la dreapta informațiile din tabela JUDETE. Problema este dificil de formulat, însă rezolvarea sa este cât se poate de simplă - joncționarea celor două relații:

R ← JONCȚIUNE (CODURI_POSTALE, JUDETE; Jud)

Una din întrebările "clasice" pentru verificarea modului în care a fost sau nu înțeleasă joncțiunea internă este: *Câte linii are tabela-rezultat al joncțiunii ?* În cazul nostru și, în general, atunci când se joncționează două tabele aflate în posturile părinte-copil răspunsul este: *câte linii are tabela-copil*. Răspunsul este corect numai atunci când se respectă integritatea referențială, altfel spus, numai atunci când toate valorile cheii străine sunt nenule și se regăsesc în tabela-părinte. Ca de obicei, lucrurile sunt mai complicate în realitate, deoarece joncțiunea nu se instituie musai între o tabelă părinte și una copil.

Exemplu 14 - Care sunt codurile poștale ale localităților din Banat ?

Soluția 1 - după calapodul exemplului anterior:

R11 ← JONCȚIUNE (CODURI_POSTALE, JUDETE; Jud)

R ← SELECȚIE (R11; Regiune = 'Banat')

Soluția 2: Mai întâi, se aplică selecția asupra tabeli JUDETE, iar tabela intermediară se joncționează cu CODURI_POSTALE:

R12 ← SELECȚIE (JUDETE; Regiune = 'Banat')

R ← JONCȚIUNE (CODURI_POSTALE, R12; Jud)

A doua variantă pare mai bună decât prima, deoarece joncțiunea operează asupra a două relații mai reduse ca dimensiuni (vezi figura 4.20). Diferența este cu atât mai vizibilă atunci când relația JUDETE conține toate județele țării, iar CODURI_POSTALE are câteva sute de înregistrări. Iar dacă ne gândim că înaintea

oricărei joncțiunii se calculează produsul cartezian, apare drept firească ideea amânării joncțiunii, astfel încât aceasta să opereze asupra unor tabele cu un număr cât mai mic de linii și coloane.

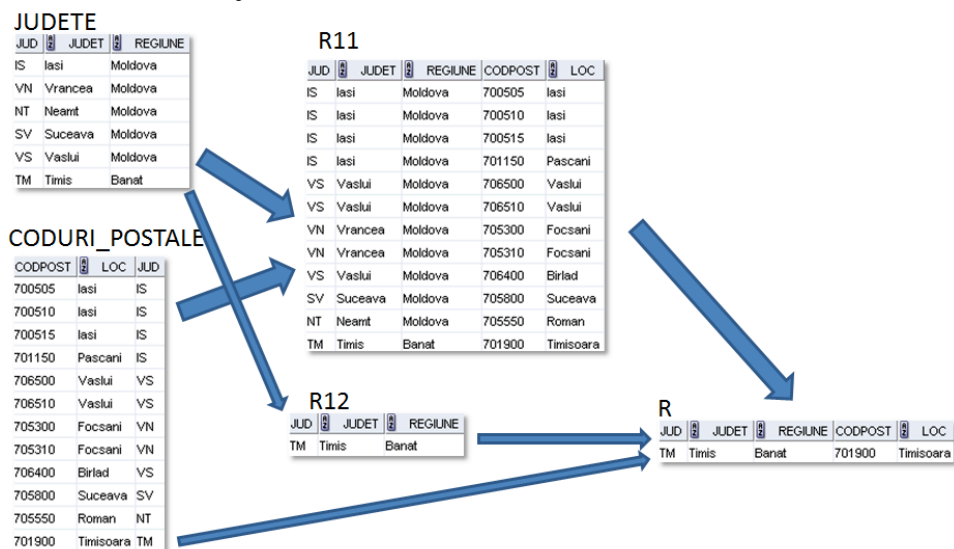


Figura 4.20. Ilustrarea modului de execuție a celor două soluții AR - exemplul 14

Coborând cu picioarele pe pământ, trebuie spus că discuția noastră are un caracter de principiu, deoarece algebra relațională este, totuși, un limbaj... curat teoretic.

Exemplu 15 - În ce zile s-a vândut produsul cu denumirea "Produs 1" ?

Elementul de noutate îl reprezintă interesul pentru o informație ce provine dintr-o relație (atributul *DataFact* din *FACTURI*) pe baza unei condiții aplicate altei relații (atributul *DenPr* din *PRODUSE*), iar cele două relații, *FACTURI* și *PRODUSE* nu sunt în raportul părinte-copil. În aceste cazuri, este necesară atragerea altor relații, până se completează "lanțul". Interogarea ce rezolvă problema ridicată în acest exemplu necesită și tabela *LINIIFACT*.

Soluție 1 - neoptimizată:

R11 ← JONCȚIUNE (*PRODUSE*, *LINIIFACT*; *CodPr*)
R12 ← JONCȚIUNE (**R11**, *FACTURI*; *NrFact*)
R13 ← SELECȚIE (**R12**; *DenPr* = 'Produs 1')
R ← PROIECȚIE (**R13**; *DataFact*)

Soluție 2 - optimizată:

R11 ← SELECȚIE (*PRODUSE*; *DenPr* = 'Produs 1')
R12 ← PROIECȚIE (**R11**; *CodPr*)
R13 ← JONCȚIUNE (**R12**, *LINIIFACT*; *CodPr*)
R14 ← PROIECȚIE (**R13**; *NrFact*)
R15 ← JONCȚIUNE (**R14**, *FACTURI*; *NrFact*)

R ← PROIECȚIE (R15; DataFact)**LINIIFACT (fragment)**

NRFAC	DENPR	LINE	CODPR	CANTITATE	PRETUNIT
1111	1	1	1	50	1000
1111	2	2	2	75	1050
1111	3	5	5	500	7060
1112	1	2	2	80	1030
1112	2	3	4	40	750

R11 (fragment)

CODPR	NRFAC	LINE	CANTITATE	PRETUNIT	DENPR	UM	GRUPA	PROCTVA
1	1111	1	50	1000	Produs 1	buc	Tigari	0,19
2	1111	2	75	1050	Produs 2	kg	Bere	0,09
5	1111	3	500	7060	Produs 5	buc	Tigari	0,19
2	1112	1	80	1030	Produs 2	kg	Bere	0,09
3	1112	2	40	750	Produs 3	kg	Bere	0,19
2	1113	1	100	975	Produs 2	kg	Bere	0,09
2	1114	4	30	1705	Produs 4	l	Dulciuri	0,09

PRODUSE

CODPR	DENPR	UM	GRUPA	PROCTVA
1	Produs 1	buc	Tigari	0,19
2	Produs 2	kg	Bere	0,09
3	Produs 3	kg	Bere	0,19
4	Produs 4	l	Dulciuri	0,09
5	Produs 5	buc	Tigari	0,19
6	Produs 6	p250g	Cafea	0,19

FACTURI (fragment)

NRFAC	DATAFACT	CODCL	OBS
1111	01-08-2007	1001	(null)
1112	01-08-2007	1005	Probleme cu transportul
1113	01-08-2007	1002	(null)
1114	01-08-2007	1006	(null)
1115	02-08-2007	1001	(null)

R12 (fragment)

NRFAC	CODPR	LINE	CANTITATE	PRETUNIT	DENPR	UM	GRUPA	PROCTVA	DATAFACT	CODCL	OBS
1111	5	3	500	7060	Produs 5	buc	Tigari	0,19	01-08-2007	1001	(null)
1111	2	2	75	1050	Produs 2	kg	Bere	0,09	01-08-2007	1001	(null)
1111	1	1	50	1000	Produs 1	buc	Tigari	0,19	01-08-2007	1001	(null)
1112	3	2	40	750	Produs 3	kg	Bere	0,19	01-08-2007	1005	Probleme cu transportul
1112	2	1	80	1030	Produs 2	kg	Bere	0,09	01-08-2007	1005	Probleme cu transportul
1113	2	1	100	975	Produs 2	kg	Bere	0,09	01-08-2007	1002	(null)
1114	5	3	700	7064	Produs 5	buc	Tigari	0,19	01-08-2007	1006	(null)
1114	4	2	30	1705	Produs 4	l	Dulciuri	0,09	01-08-2007	1006	(null)

R13

NRFAC	CODPR	LINE	CANTITATE	PRETUNIT	DENPR	UM	GRUPA	PROCTVA	DATAFACT	CODCL	OBS
1111	1	1	50	1000	Produs 1	buc	Tigari	0,19	01-08-2007	1001	(null)
1117	1	2	100	950	Produs 1	buc	Tigari	0,19	03-08-2007	1001	(null)
1118	1	2	150	930	Produs 1	buc	Tigari	0,19	04-08-2007	1001	(null)
2111	1	1	57	1000	Produs 1	buc	Tigari	0,19	14-08-2007	1001	(null)
2117	1	2	110	950	Produs 1	buc	Tigari	0,19	16-08-2007	1001	(null)
2118	1	2	120	930	Produs 1	buc	Tigari	0,19	16-08-2007	1001	(null)
3111	1	1	57	1000	Produs 1	buc	Tigari	0,19	01-09-2007	1001	(null)
3117	1	2	110	950	Produs 1	buc	Tigari	0,19	10-09-2007	1001	(null)
3118	1	2	120	930	Produs 1	buc	Tigari	0,19	17-09-2007	1001	(null)

R

DATAFACT
01-08-2007
04-08-2007
17-09-2007
03-08-2007
01-09-2007
14-08-2007
16-08-2007
10-09-2007

Figura 4.21. Ilustrarea modului de execuție a primei soluții AR - exemplul 15

În cea de-a doua soluție (vezi figura 4.22), fideli principiului “joncțiunii celei mai economicoase”, am eliminat nu numai tuplurile, dar și atributele de prisos înaintea calculării relației intermediare.

Exemplu 16 - În ce județe s-a vândut produsul cu denumirea “Produs 1” în perioada 3-5 august 2007 ?

Relația rezultat trebuie să conțină valori ale atributului *Judet* din tabela *JUDETE*. Predicatul de selecție se aplică însă, în alte două tabele: *PRODUSE*, în care *DenPr* = 'Produs 1' și *FACTURI* ale cărei tupluri trebuie să verifice condiția: *DataFact* >= '2007/08/03' AND *DataFact* <= '2007/08/05'.

R11 ← SELECTIE (PRODUSE; DenPr = “Produs 1”)

R12 ← JONCȚIUNE (R11, LINIIFACT; CodPr)

R13 ← PROIECȚIE (R12; NrFact)

R14 ← SELECTIE(FACTURI; DataFact >= '2007/08/03' AND DataFact <= '2007/08/05')

R5 ← JONCȚIUNE (R3, R4; NrFact)

R6 ← JONCȚIUNE (R5, CLIENTI; CodCl)

R7 ← PROIECȚIE (R6; CodPost)

R8 \leftarrow JONCTIUNE (R7, CODURI_POSTALE; CodPost)

R9 \leftarrow PROIECTIE (R8; Jud)

R10 \leftarrow JONCTIUNE (R9, JUDETE; Jud)

R \leftarrow PROIECTIE (R10; Judet)

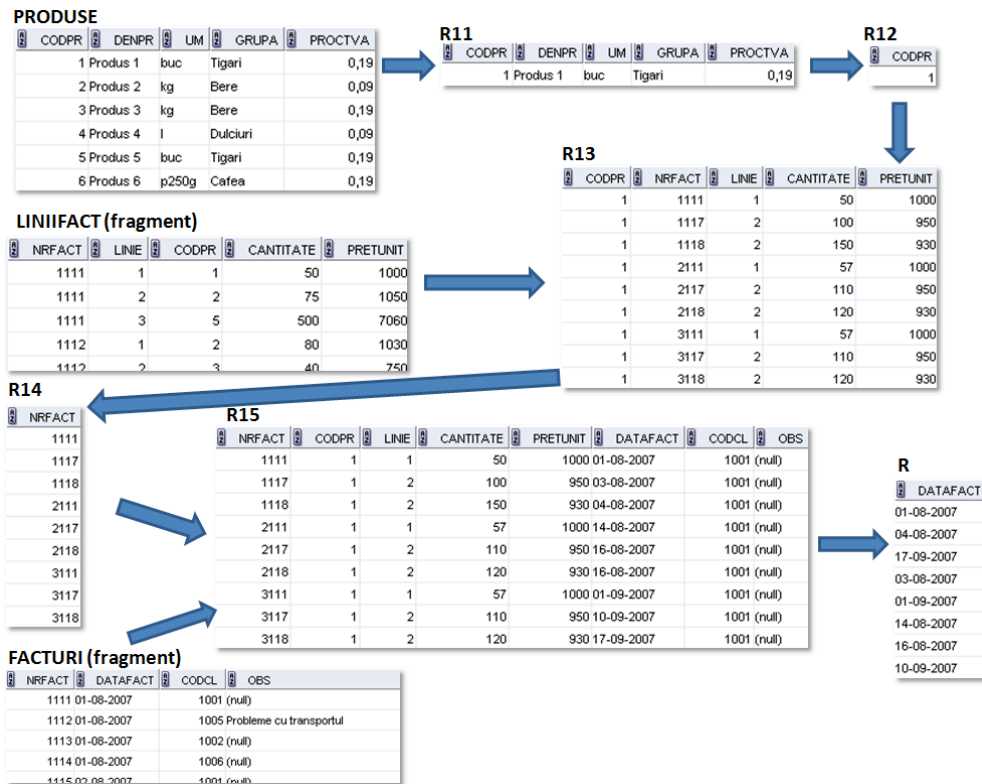


Figura 4.22. Ilustrarea modului de execuție a celei de-a doua soluții AR - exemplul 15

Exemplu 17 - În ce zile s-au vândut și produsul cu denumirea "Produs 1" și cel cu denumirea "Produs 2" ?

Rezultatul conține atributul DataFact. Predicatul de selecție se aplică tablei PRODUSE.

Soluție 1:

R11 \leftarrow SELECȚIE (PRODUSE; DenPr = 'Produs 1')

R12 \leftarrow JONCTIUNE (R11, LINEIFACT; CodPr)

R13 \leftarrow JONCTIUNE (R12, FACTURI; NrFact)

R14 \leftarrow PROIECTIE (R13; DataFact)

R15 \leftarrow SELECȚIE (PRODUSE; DenPr = 'Produs 2')

R16 \leftarrow JONCTIUNE (R15, LINEIFACT; CodPr)

R17 \leftarrow JONCTIUNE (R16, FACTURI; NrFact)

R18 \leftarrow PROIECTIE (R17; DataFact)

R \leftarrow R14 \cap R18

Soluție 2:

```

R21 ← SELECTIE (PRODUSE; DenPr = 'Produs 1')
R22 ← JONCTIUNE (R21, LINIIFACT; CodPr)
R23 ← JONCTIUNE (R22, FACTURI; NrFact)
R24 ← SELECTIE (PRODUSE; DenPr = 'Produs 2')
R25 ← JONCTIUNE (R24, LINIIFACT; CodPr)
R26 ← JONCTIUNE (R25, FACTURI; NrFact)
R27 ← JONCTIUNE (R23, R26; DataFact)
R ← PROIECTIE (R27; DataFact)

```

Prima variantă este una "cuminte". Se intersectează relația zilelor în care s-a vândut primul produs (R14) cu relația zilelor în care s-a facturat produsul 2 (R18). A doua e ceva mai insolită. Pur și simplu, în loc de intersecție folosim joncțiunea. Cum, spre deosebire de intersecție, relațiile joncționate nu trebuie să fie unicompatibile, putem să operăm direct joncțiunea între R23 și R26.

Reținem ideea: putem să simulăm intersecția a două relații prin joncțiune. Pentru a fi mai convingător, rogu-vă să examinați figura 4.23 în care, pornind de la relațiile R1 și R2, se obține relația-intersecție R direct prin operatorul intersecție (stânga figurii) și mai pe ocolite, folosind joncțiunea (dreapta).

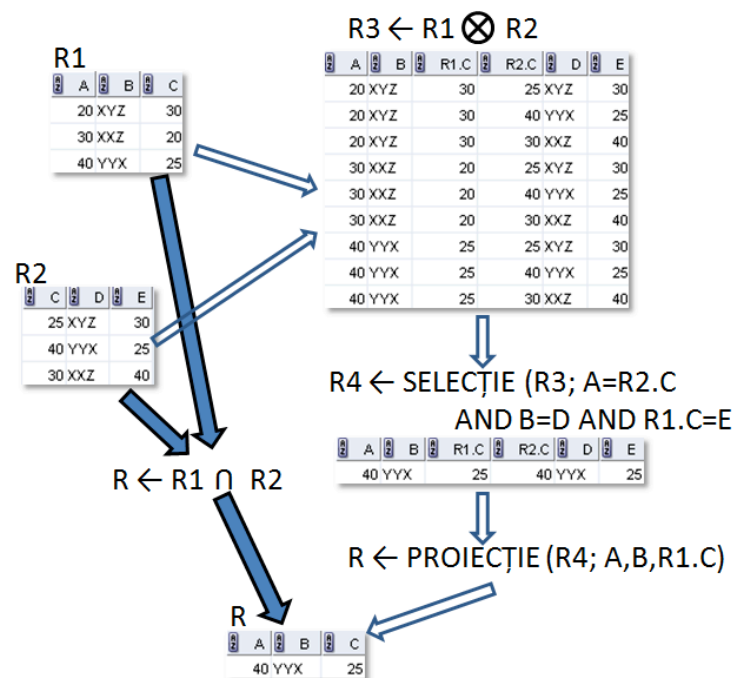


Figura 4.23. Intersecția prin joncțiune

Exemplu 18 - Ce clienți au cumpărat "Produs 2" și "Produs 3", dar nu au cumpărat "Produs 5" ?

Bună întrebare ! Lucrurile nu sunt însă atât de complicate precum par, deoarece folosim elemente din exemplul anterior (intersecția) plus operatorul... diferență.

R11 ← **SELECTIE** (**PRODUSE**; **DenPr** = 'Produs 2')

R12 ← **JONCTIUNE** (**R11**, **LINIIFACT**; **CodPr**)

R13 ← **JONCTIUNE** (**R12**, **FACTURI**; **NrFact**)

R14 ← **JONCTIUNE** (**R13**, **CLIENTI**; **CodCl**)

R15 ← **PROIECTIE** (**R14**; **DenCl**)

R16 ← **SELECTIE** (**PRODUSE**; **DenPr** = 'Produs 3')

R17 ← **JONCTIUNE** (**R16**, **LINIIFACT**; **CodPr**)

R18 ← **JONCTIUNE** (**R17**, **FACTURI**; **NrFact**)

R19 ← **JONCTIUNE** (**R18**, **CLIENTI**; **CodCl**)

R20 ← **PROIECTIE** (**R19**; **DenCl**)

R21 ← **SELECTIE** (**PRODUSE**; **DenPr** = 'Produs 5')

R22 ← **JONCTIUNE** (**R21**, **LINIIFACT**; **CodPr**)

R23 ← **JONCTIUNE** (**R22**, **FACTURI**; **NrFact**)

R24 ← **JONCTIUNE** (**R23**, **CLIENTI**; **CodCl**)

R25 ← **PROIECTIE** (**R24**; **DenCl**)

R ← **R15** ∩ **R20** – **R25**

Exemplu 19 - Ce facturi au fost emise în aceeași zi cu factura 1120 ?

Spre deosebire de interogările de până acum, condiția de selecție este una indirectă. În prealabil, trebuie determinată ziua în care a fost întocmită factura cu numărul 1120.

FACTURI

NRFACT	DATAFACT	CODCL	OBS
1111	01-08-2007	1001 (null)	
1112	01-08-2007	1005 Probleme cu transportul	
1113	01-08-2007	1002 (null)	
1114	01-08-2007	1006 (null)	
1115	02-08-2007	1001 (null)	
1116	02-08-2007	1007 Pretul propus initial a fost modificat	
1117	03-08-2007	1001 (null)	
1118	04-08-2007	1001 (null)	
1119	07-08-2007	1003 (null)	
1120	07-08-2007	1001 (null)	
1121	07-08-2007	1004 (null)	
1122	07-08-2007	1005 (null)	
2111	14-08-2007	1001 (null)	
2112	14-08-2007	1005 Probleme cu transportul	
2113	14-08-2007	1002 (null)	
2115	15-08-2007	1001 (null)	
2116	15-08-2007	1007 Pretul propus initial a fost modificat	
2117	16-08-2007	1001 (null)	
2118	16-08-2007	1001 (null)	
2119	21-08-2007	1003 (null)	
2121	21-08-2007	1004 (null)	
2122	22-08-2007	1005 (null)	
3111	01-09-2007	1001 (null)	
3112	01-09-2007	1005 Probleme cu transportul	
3113	02-09-2007	1002 (null)	
3115	02-09-2007	1001 (null)	
3116	10-09-2007	1007 Pretul propus initial a fost modificat	
3117	10-09-2007	1001 (null)	
3118	17-09-2007	1001 (null)	
3119	02-10-2007	1003 (null)	

R11

NRFACT	DATAFACT	CODCL	OBS
1120	07-08-2007	1001 (null)	

R12

DATAFACT
07-08-2007

R13

NRFACT	FACTURI.DATAFACT	CODCL	OBS	R12.DATAFACT
1119	07-08-2007	1003 (null)		07-08-2007
1120	07-08-2007	1001 (null)		07-08-2007
1121	07-08-2007	1004 (null)		07-08-2007
1122	07-08-2007	1005 (null)		07-08-2007

R

NRFACT
1119
1120
1121
1122

Figura 4.24. Facturile emise în aceeași zi cu 1120 – varianta „economicoasă”

Apoi trebuie extrase, din relația FACTURI, liniile pentru care DataFact are valoarea zilei facturii-reper. Soluția “economicoasă”, a cărei logică este expusă în figura 4.24, este:

R11 ← **SELECȚIE** (FACTURI; NrFact = 1120)

R12 ← **PROIECȚIE** (R11; DataFact)

R13 ← **JONCȚIUNE** (FACTURI, R12; DataFact)

R ← **PROIECȚIE** (R13; NrFact)

O variantă “risipitoare” (însă la fel de corectă, ca rezultat) presupune auto-joncționarea tabelului FACTURI. Astfel spus, se joncționează două instanțe ale tabelului FACTURI după atributul DataFact.

R21 ← **JONCȚIUNE** (FACTURI F1, FACTURI F2; F1.DataFact=F2.DataFact)

R22 ← **SELECȚIE** (R21; F1.NrFact = 1120)

R ← **PROIECȚIE** (R22; F2.NrFact)

Firește, trebuie făcute niște supoziții despre modul în care vor fi denumite atributele în R21 și R22, întrucât fiecare apare de două ori, corespunzător celor două instanțe. Din fericire, algebra relațională este teoretică, așa că ne luăm o anumită libertate, fără riscul represaliilor. Derularea operațiunilor este sugerată în figura 4.25.

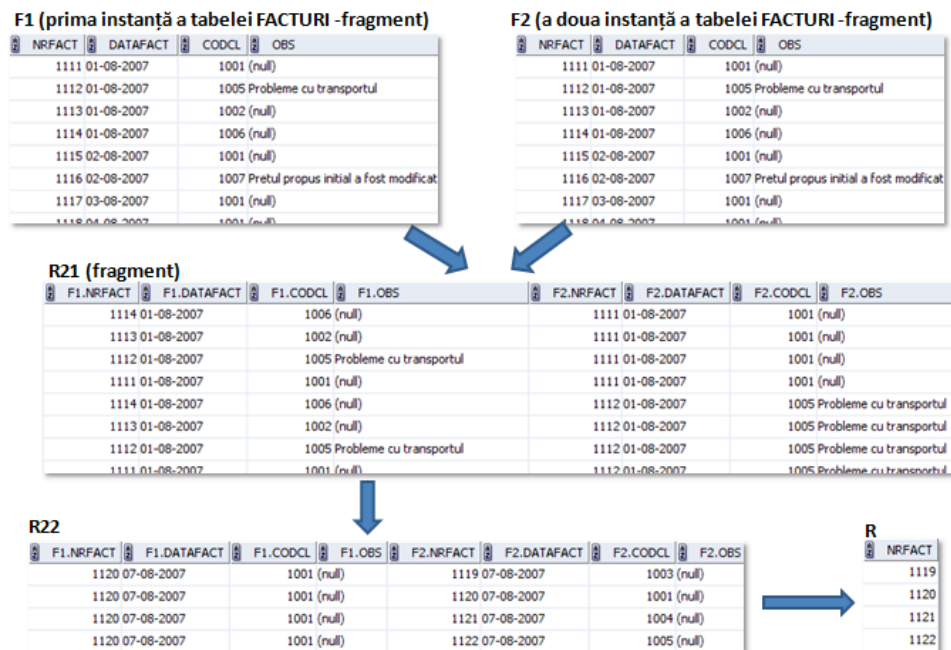


Figura 4.25. Facturile emise în aceeași zi cu 1120 – varianta „risipitoare”

Exemplu 20 – Să se afișeze numele doctorului pentru fiecare examinare din triaj

Riscăm să încheiem paragraful și să nu dăm măcar un exemplu de theta-joncțiune. Nu întâmplător. Cazurile în care trebuie să recurgem la o theta-joncțiune (care nu este echi-joncțiune) sunt ceva mai rare. Noi o să apelăm la cele câteva tabelele din baza de date a aplicației *Camera de gardă* sau, mă rog, *Spitalul de urgență*. Știind că, pentru spitalul luat în discuție, în orice moment, de gardă este un singur medic, soluția presupune joncționarea tabelului TRIAJ cu tabelul GARZI. Interesant este că joncționarea nu se bazează pe egalitatea valorilor dintre atributele DataExaminare și vreunul dintre atributele Inceput_Garda, Sfirsit_Garda. DataExaminare reprezintă momentul efectiv al sosirii în triaj a bolnavului, în timp ce fiecare gardă este specificată printr-un interval dată/timp (vezi figura 4.26).

R11 ← JONCȚIUNE (TRIAJ, GARZI;

TRIAJ.DataOra_Examinare >= GARZI.InceputGarda AND
TRIAJ.DataOra_Examinare <= GARZI.SfirsitGarda)

TRIAJ						GARZI		
IDEXAMINARE	DATAORA_EXAMINARE	IDPACIENT	SIMPTOME	TRATAM...	SECTIE...	IDDOCTOR	INCEPUT_GARDA	SFIRSIT_GARDA
1	03-01-2008 07:18:00...		1 dureri de stom...	(null)	boli interne		1 03-01-2008 07:00:00...	03-01-2008 19:00:00...
2	03-01-2008 08:45:00...		2 febra puternic...	penicilina	boli interne		2 03-01-2008 19:00:00...	04-01-2008 07:00:00...
3	03-01-2008 12:45:00...		3 deranjament s...	scobutil	(null)		3 04-01-2008 07:00:00...	04-01-2008 18:00:00...
4	03-01-2008 20:45:00...		4 palpitații cardiace	linistin	cardiologie		4 04-01-2008 18:00:00...	05-01-2008 06:00:00...
5	04-01-2008 01:28:00...		5 plaga profund...	antibiotice, ...	(null)		1 05-01-2008 06:00:00...	05-01-2008 14:00:00...
6	04-01-2008 10:45:00...		3 contractii stom...	(null)	boli interne		2 05-01-2008 14:00:00...	05-01-2008 22:00:00...
7	04-01-2008 11:20:00...		7 fata de culoar...	(null)	hepatologie		5 05-01-2008 22:00:00...	06-01-2008 09:00:00...
8	04-01-2008 22:45:00...		8 dureri articulare	scobutil	(null)			
9	05-01-2008 06:18:00...		5 febra puternic...	penicilina	chirurgie			

R11								
IDEXAMINARE	DATAORA_EXAMINARE	IDPACIENT	SIMPTOME	TRATAME...	SECTIE_DESTINATIE	IDDOCTOR	INCEPUT_GARDA	SFIRSIT_GARDA
1	03-01-2008 07:18:00...		1 dureri de stomac ...	(null)	boli interne		1 03-01-2008 07:00:00...	03-01-2008 19:00:00...
2	03-01-2008 08:45:00...		2 febra puternica, ...	penicilina	boli interne		1 03-01-2008 07:00:00...	03-01-2008 19:00:00...
3	03-01-2008 12:45:00...		3 deranjament sto...	scobutil	(null)		1 03-01-2008 07:00:00...	03-01-2008 19:00:00...
4	03-01-2008 20:45:00...		4 palpitații cardiace	linistin	cardiologie		2 03-01-2008 19:00:00...	04-01-2008 07:00:00...
5	04-01-2008 01:28:00...		5 plaga profunda p...	antibiotice, pa...	(null)		2 03-01-2008 19:00:00...	04-01-2008 07:00:00...
6	04-01-2008 10:45:00...		3 contractii stomac...	(null)	boli interne		3 04-01-2008 07:00:00...	04-01-2008 18:00:00...
7	04-01-2008 11:20:00...		7 fata de culoare g...	(null)	hepatologie		3 04-01-2008 07:00:00...	04-01-2008 18:00:00...
8	04-01-2008 22:45:00...		8 dureri articulare	scobutil	(null)		4 04-01-2008 18:00:00...	05-01-2008 06:00:00...
9	05-01-2008 06:18:00...		5 febra puternica, ...	penicilina	chirurgie		1 05-01-2008 06:00:00...	05-01-2008 14:00:00...

Figura 4.26. Theta-joncțiune dintre TRIAJ și GARZI

Soluția continuă astfel:

R12 ← JONCȚIUNE (R11, DOCTORI; IdDoctor)

R13 ← JONCȚIUNE (R12, PACIENTI; IdPacient)

R ← PROIECȚIE (R13, DataOra_Examinare, NumePacient, NumeDoctor)

4.4.5. Joncțiunea externă

Ideea de bază a joncțiunii externe este de a include în rezultat și tupluri din una din relații, sau din ambele relații, care prezintă valori ale atributului de legătură ce *nu* se regăsesc în cealaltă relație. Dacă precedentele tipuri de joncțiune sunt comutative, în cazul joncțiunii externe trebuie specificat din care relație se extrag liniile fără corepondent în cealaltă relație. De aceea, există *joncțiune externă la*

stânga și joncțiune externă la dreapta. La acestea se adaugă joncțiunea externă totală (denumită și *plină* sau *deplină*) care reprezintă reuniunea celor două. Apelând la aceleași tabele "abstracte", R1 și R2, diferența dintre joncțiunea internă (echi-joncțiunea) și cele trei tipuri de joncțiune externă apare cu mai multă claritate în figura 4.27.

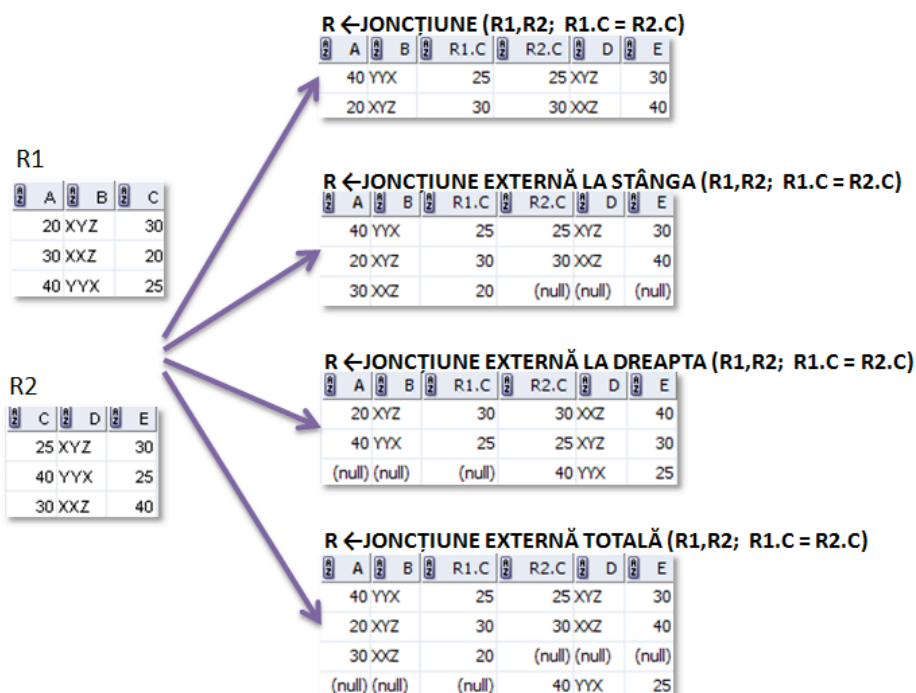


Figura 4.27. Diferența dintre echi-joncțiune și joncțiunile externe

Liniile dintr-o tabelă care prezintă valori ale atributului de legătură fără corespondent în cealaltă tabelă sunt completate în rezultat cu valori NULL.

Exemplu 21 - Care sunt codurile poștale la care care nu avem nici un client ?

Soluția 1 se bazează pe diferența dintre tabela tuturor codurilor poștale preluate din CODURI_POSTALE și tabela codurilor la care există cel puțin un client (din CLIENTI).

R31 ← PROIECȚIE (CODURI_POSTALE; CodPost)

R32 ← PROIECȚIE (CLIENTI; CodPost)

R33 ← R31 – R32

R ← JONCȚIUNE (R33, CODURI_POSTALE; CodPost)

Soluția 2 utilizează proaspăta joncțiune externă.

**R31 ← JONCȚIUNE EXTERNĂ LA STÂNGA (CODURI_POSTALE, CLIENTI;
CODURI_POSTALE.CodPost = CLIENTI.CodPost)**

R ← SELECTIE (R31; CodCl IS NULL)

Pentru a identifica localitățile fără clienți, s-au extras, din joncțiunea externă la stânga a relațiilor CODURI_POSTALE și CLIENTI, numai liniile în care unul din atributele preluate din CLIENTI (preferabil cheia primară) are valoarea NULL.

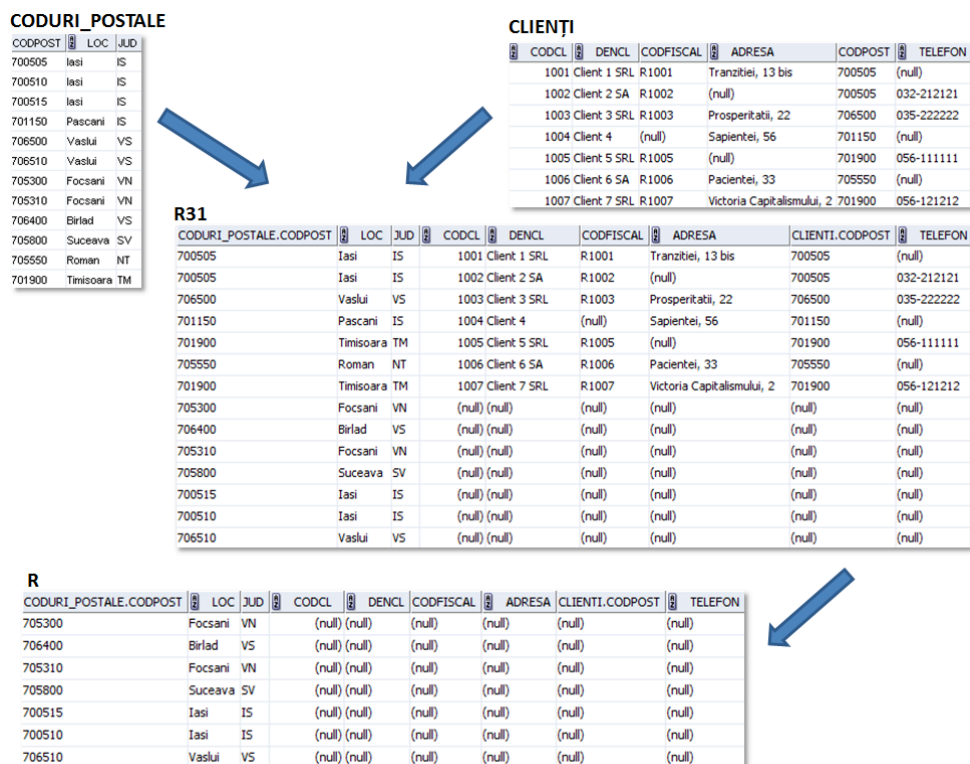


Figura 4.28. Folosirea joncțiunii externe – exemplul 21

4.4.6. Semijoncțiunea

Semijoncțiunea este unul din cele mai “marginalizate” tipuri de joncțiune. Implementarea sa în SGBD-urile comerciale este extrem de rară. A fost introdusă din dorința de a optimiza procesul de interogare (consultare). Calculul semijoncțiunii a două tabele presupune selectarea numai a liniilor din prima tabelă care apar în joncțiune cu linii din a doua tabelă – vezi figura 4.29.

Firește, operatorul nu este comutativ. Implementarea sa în SQL este una destul de facilă, după cum vom vedea în capitolele următoare.

Exemplu 22 - Care sunt codurile poștale (plus denumirea localității și indicativul județului) în care există măcar un client ?

Soluția 1 – bazată pe echijoncțiune:

R1 ← PROIECȚIE (CLIENTI; CodPost)

R2 ← JONCȚIUNE (R1, CODURI_POSTALE; CodPost)

R ← PROIECȚIE (R2; CodPost, Loc, Jud)

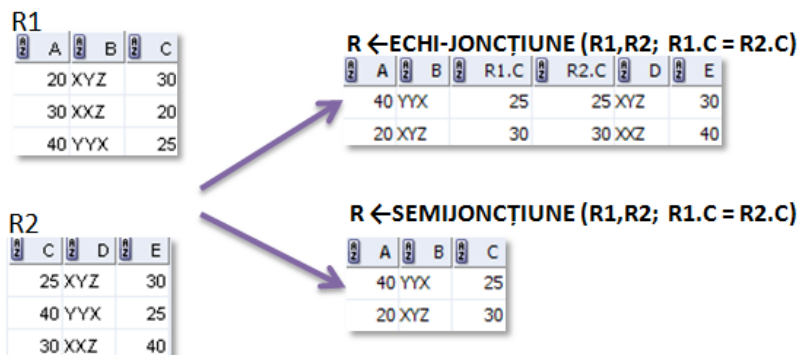


Figura 4.29. Diferența dintre echi-joncțiune și semi-joncțiune

Soluția 2 – bazată pe semijoncțiune:

R ← SEMIJONCȚIUNE (CODURI_POSTALE, CLIENTI; CodPost)

4.4.7. Diviziunea

Este cel mai complex și mai greu de explicat dintre operatorii prezentați în acest capitol. Codd l-a imaginat ca operator invers al produsului cartezian. Pentru a-l defini, se pornește de la două relații $RD1(X,Y)$ și $RD2(Y)$; prima are, care va să zică, două atribute sau grupe de atribute, notate X și Y , în timp ce a doua numai atributul sau grupul de atribute notat cu Y (definit pe același domeniu ca și în relația $RD1$). O primă restricție: relația $RD2(Y)$, fiind numitorul diviziunii, nu trebuie să fie vidă.

Diviziunea relațională $RD1 \div RD2$ are ca rezultat o relație definită ca ansamblul sub-tuplurilor $RD1(X)$ pentru care produsul (lor) cartezian cu $RD2(Y)$ este un subansamblu al $RD1(X,Y)$. Rezultatul expresiei $RD1 \div RD2$ reprezintă câtul diviziunii, fiind o relație ce poate fi notată $RD3(X)$. Într-o altă formulare, $x_i \in R3$ dacă și numai dacă $\forall y_i \in Y \in R2 \rightarrow \exists (x_i, y_i) \in R1$. Pentru simplificarea prezentării, în continuare am considerat X și Y două atribute, deși, după cum reiese din preambul, acestea pot fi grupe (ansambluri) de atribute. Să examinăm elementele din figura 4.30.

„Calcularea” relației $R \leftarrow RD1 \div RD2$ este sinonimă cu rezolvarea problemei: care dintre x_1, x_2, x_3, x_4 și x_5 apar în $RD1$, în tuple împreună cu toate valorile lui Y din $RD2$, respectiv y_1, y_2, y_3, y_4 și y_5 ?

Se parcurg pe rând valorile x_i ale atributului X din relația $RD1$:

- x_1 apare cu y_1 (în tuplul 1), cu y_2 (în tuplul 4), cu y_3 (în tuplul 7), cu y_4 (în tuplul 10) și cu y_5 (în tuplul 13). Deci x_1 îndeplinește condiția și va fi inclus în relația R ;
- x_2 apare cu y_1 (în tuplul 2) dar nu apare cu y_2 - nu va face parte din R ;

- x3 apare cu y1 (în tuplul 3), cu y2 (în tuplul 5), cu y3 (în tuplul 8), cu y4 (în tuplul 11) și cu y5 (în tuplul 15) - îndeplinește condiția și va fi tuplu în R;
- x4 nu apare cu y1 - nu va face parte din R;
- x5 nu apare cu y1 - nu va face parte din R.

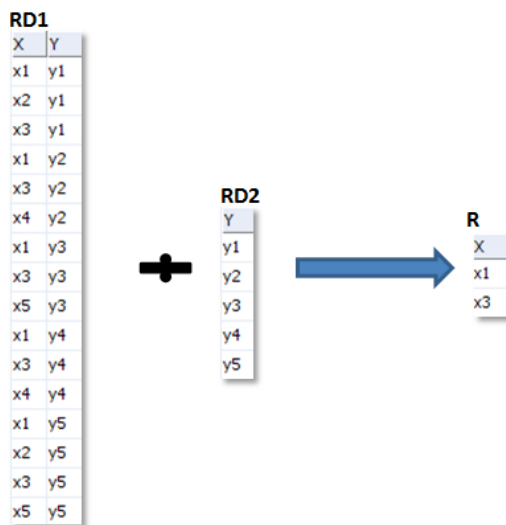


Figura 4.30. Diviziunea relațională

În urma raționamentului de mai sus, tabela R va fi alcătuită din două tupluri. Deși pare un operator ceva mai metafizic, diviziunea relațională este deosebit de utilă pentru formularea consultărilor în care apare clauza "∀" ("oricare ar fi" sau "pentru toate").

Exemplu 23 - Care sunt clienții pentru care există cel puțin câte o factură emisă în fiecare zi cu vânzări din perioada 10-30 septembrie 2007?

Într-o altă formulare, ne interesează clienții care au cumpărat "câte ceva" în toate zilele în care s-au efectuat vânzări (s-a întocmit măcar o factură) din perioada 10-30 septembrie 2007 ? Prin urmare, câțul va fi o tabelă cu singur atribut, DenCl (denumirea clientului), iar divizorul va fi o relație alcătuită numai din atributul DataFact (conține toate zilele în care s-au efectuat vânzări). Dacă am merge pe calapodul prezentat, am putea nota $R(\text{DenCl})$, $RD2(\text{DataFact})$. Cunoscând structura câțului și a divizorului, putem determina structura tablei-deîmpărțit: $RD1(\text{DenCl}, \text{DataFact})$. Relația RD1 va conține denumirile clienților și zilele în care există măcar o factură pentru clientul respectiv. Soluția poate fi redactată în următorii pași:

- construire relației-deîmpărțit:

**R11 ← SELECTIE (FACTURI; DataFact >= '2007-09-10' AND
DataFact <= '2007-09-30')**

R12 ← JONCTIUNE (CLIENTI, R11; CodCl)

RD1 ← PROIECTIE (R12; DenCl, DataFact)

- construire relație-numitor:
- RD2 ← PROIECȚIE (R11; DataFact)**
- în fine, apoteoza:

R ← RD1 ÷ RD2

Relațiile implicate în această soluție (cu excepțiile FACTURI și CLIEȚI) sunt prezentate în figura 4.31.

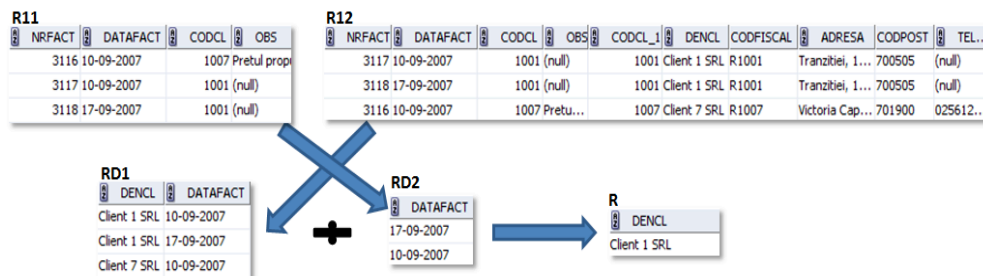


Figura 4.31. Diviziunea relațională – exemplu 23

Poate o să spuneți că n-a meritat efortul, dar acest gen de soluții se aplică la o largă gamă de probleme. Să discutăm câteva spețe.

Exemplu 24 - În ce zile s-au vândut și produsul cu denumirea "Produs 1" și cel cu denumirea "Produs 2" ?

Vă gândiți, probabil, că ați mai văzut undeva acest enunț. Este chiar textul problemei de la exemplul 17. La acel moment, am formulat două soluții: una "clasică", bazată pe intersecție și una mai neconvențională ce utilizează joncțiunea. Adăugăm la acestea o alta, încadrabilă probabil tot în categoria "neconvenționale".

Soluție 3

R11 ← SELECTIE (PRODUSE; DenPr = 'Produs 1' OR DenPr = 'Produs 2')

R12 ← JONCȚIUNE (R11, LINIIFACT; CodPr)

R13 ← JONCȚIUNE (R12, FACTURI; NrFact)

R1 ← PROIECȚIE (R13; DataFact, CodPr)

R2 ← PROIECȚIE (R11; CodPr)

R3 ← R1 ÷ R2

Relația divizor este alcătuită din două tupluri ce conțin codurile celor două produse. Deîmpărțitul este alcătuit din atributele DataFact și CodProd și conține toate produsele vândute în fiecare din zilele de facturare.

Exemplu 25 - Ce facturi au fost emise în aceeași zi cu factura 1120 ?

Deși reprezintă un regres vizibil față de nivelul interogărilor precedente, revenim discret la enunțul exemplului 19. Pentru această problemă se poate încropi și o variantă de rezolvare bazată pe diviziunea relațională.

R1 ← PROIECȚIE (FACTURI; NrFact, DataFact)

R2' ← SELECTIE (FACTURI; NrFact = 1120)

R2 ← PROIECȚIE (R2'; DataFact)

R ← DIVIZIUNE (R1, R2)

Exemplu 26 - Ce produse au fost vândute tuturor clienților ?

R11 ← JONCTIUNE (LINIIFACT, FACTURI; NrFact)

R1 ← PROIECȚIE (R11; CodPr, CodCl)

R2 ← PROIECȚIE (CLIENTI; CodCl)

R3 ← DIVIZIUNE (R1, R2)

R ← JONCTIUNE (R3, PRODUSE; CodPr)

Exemplu 27 - Care sunt facturile care conțin măcar produsele din factura 1117 ?

Și acesta este un gen de probleme pentru care diviziunea relațională este o binecuvântare. În tabela LINIIFACT putem observa că factura 1117 consemnează vânzarea a două produse ce au codurile 1 și 2. Așadar, trebuie să depistăm toate facturile în care apar măcar produsele 1 și 2, eventual plus alte produse. Relația “numărător” R1 conține două atribute, NrFact (X-ul) și CodPr (Y-ul), iar tuplurile sale sunt, de fapt, cele ale tabelului LINIIFACT. Relația “numitor” R2 conține codurile produselor (CodPr) prezenta în factura etalon-1117.

R1 ← PROIECȚIE (LINIIFACT; NrFact, CodPr)

R21 ← SELECȚIE (LINIIFACT; NrFact = 1117)

R2 ← PROIECȚIE (R21; CodPr)

R ← DIVIZIUNE (R1, R2)

Schema de funcționare a soluției este ilustrată în figura 4.32.

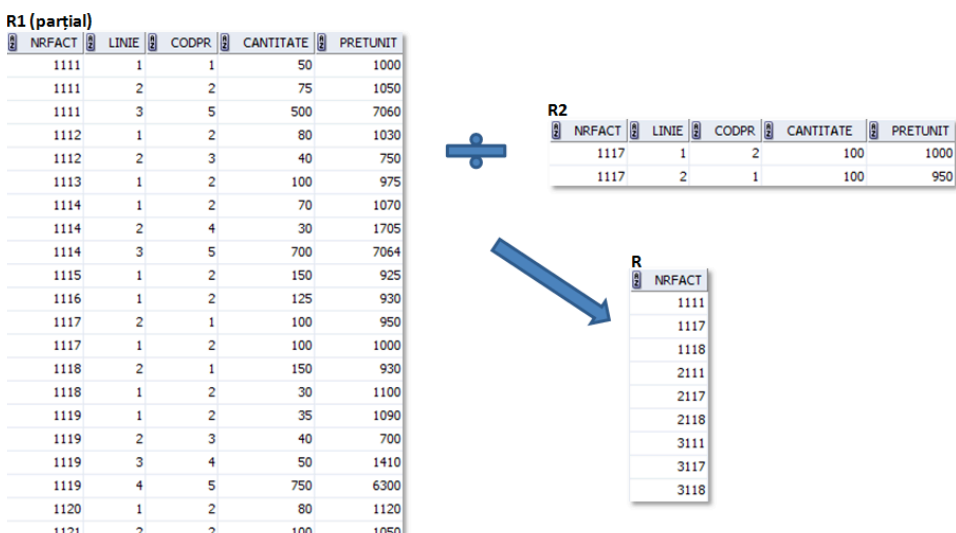


Figura 4.32. Diviziunea relațională – exemplu 27

Diviziunea relațională nu este un operator fundamental; funcționalitatea sa poate fi realizată prin combinarea operatorilor: produs cartezian, diferență și proiecție. Se reiau în discuție atât tabelele RD1 și RD2 din figura 4.30, cât și problema: care dintre valorile x_i apar în tupluri, în RD1, cu toate valorile y_j din RD2 ?

O soluție a problemei poate fi constituită din următorii pași:

R11 ← PROIECȚIE (RD1; X)

$R12 \leftarrow R11 \otimes RD2$. Tabela R12 cuprinde toate tuplurile posibile (x_i, y_j)

$R13 \leftarrow R12 - RD1$. Interesează ce tupluri din R12 lipsesc în R1. Valorile x_i din R13 nu apar în RD1 în combinație cu toate valorile y_j din RD2. Se elimină dublurile prin:

$R14 \leftarrow \text{PROIECȚIE}(R13; X)$. Valorile x_i din R14 sunt cele care nu prezintă tupluri obținute prin combinații cu toate valorile y_j . Deoarece interesează cele care prezintă combinațiile respective, rezultatul se obține prin diferența:

$R \leftarrow R11 - R14$.

Figura 4.33 aduce, cât de cât, un plus de claritate.

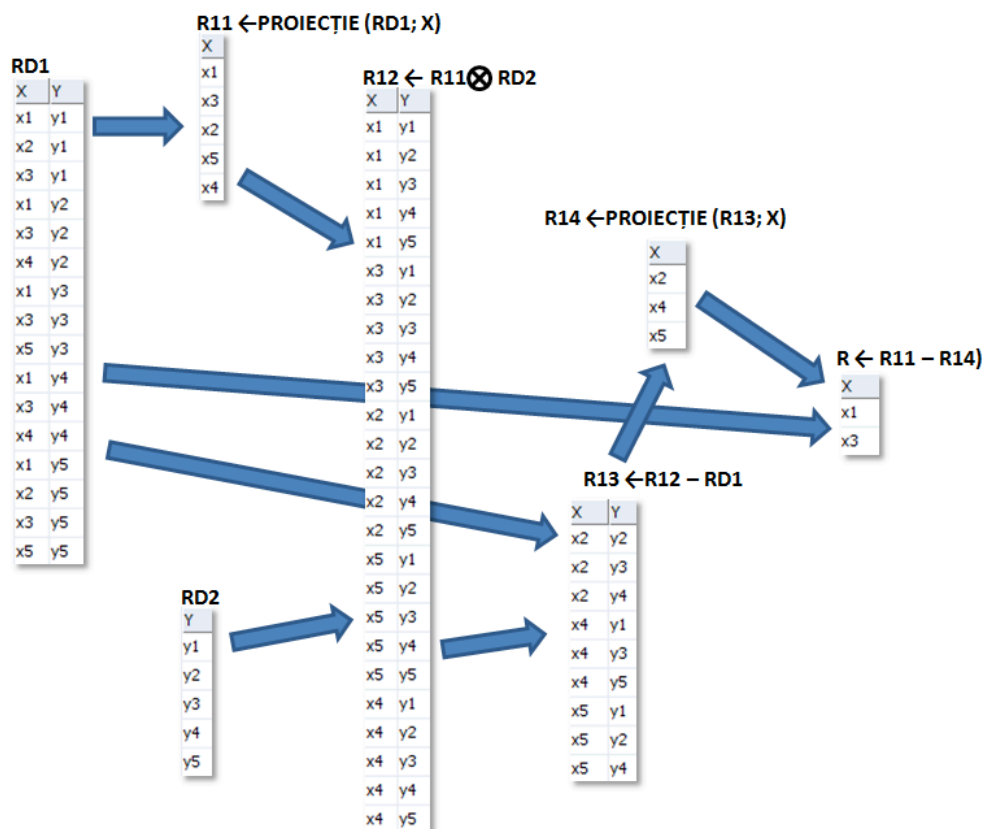


Figura 4.33. Simularea diviziunii prin alți operatori relaționali

Dintre operatorii prezentați pe parcursul ultimelor două paragrafe, selecția, proiecția, produsul cartezian, reuniunea și diferența sunt operatori *fundamentali* sau *irreductibili*, în sensul că nici unul nu poate fi definit prin intermediul celorlalți.

Joncțiunea, intersecția și diviziunea sunt operatori *derivați*:

- Joncțiunea se exprimă cu ajutorul produsului cartezian urmat de selecție;

- Intersecția se exprimă cu ajutorul diferenței, conform relației $R1 \cap R2 = R1 - (R1 - R2)$;
- Diviziunea poate fi exprimată, așa cum am văzut ceva mai sus, prin intermediul produsului cartezian, proiecției și diferenței.

4.5. Notății și reprezentări grafice ale interogărilor

Încă de la începutul prezentării operatorilor algebrei relaționale, precizam că notația pe care o vom folosi este una din cele mai comode (nu și cea mai uzuală). Modelul relațional este, în ciuda materialului de față, unul matematizat. Algebra relațională este, așa cum îi spune și numele, riguros formalizată. De aceea, și pentru un look mai științific, majoritatea lucrărilor folosesc o notație matematică sau... aproximativ matematică.

4.5.1. Notăția matematică

Există o multitudine de lucrări care utilizează această notație. În cazul de față, sursa principală o constituie traducerea franceză a unei lucrări mai vechi de Korth și Silberschatz⁴. Notația prezentată folosește, pe lângă clasicele:

- \cup - reuniune,
- \cap - intersecție,
- \times - produs cartezian,
- \div - diviziune

și simboluri din alfabetul grecesc, după cum urmează:

pentru selecție: $\sigma_P(R)$, unde P este un predicat aplicat asupra relației R,

pentru proiecție: $\Pi_S(R)$, unde S este o listă de atribute din R,

pentru joncțiune: $R1 \bowtie_{R1.A > R2.C} R2$ - indică (theta)joncțiunea relațiilor R1 și R2 prin predicatul: $R1.A > R2.C$.

Exemple

Soluția de la exemplul 7:

R1 \leftarrow **SELECȚIE** (**CLIENTI**; **DenCl = "Client 2 SA"**)

R2 \leftarrow **PROIECȚIE** (**R1**; **Telefon**)

poate fi transcrisă în notația matematică după cum urmează:

$$\Pi_{\text{Telefon}} (\sigma_{\text{DenCl}="Client2SA"}(\text{CLIENTI}))$$

După cum se observă, întreaga interogare se scrie într-o singură expresie.

Soluția de la exemplul 9:

⁴ [Korth&Silberschatz 88]

R1 ← **SELECȚIE** (**LINIIFACT**; **NrFact** = 1111)
R2 ← **PROIECȚIE** (**R1**; **CodPr**)
R3 ← **SELECȚIE** (**LINIIFACT**; **NrFact** = 1117)
R4 ← **PROIECȚIE** (**R3**; **CodPr**)
R5 ← **R2** ∩ **R4**:

$$\prod_{\text{CodPr}} (\sigma_{\text{NrFact}=1111}(\text{LINIIFACT})) \cap \prod_{\text{CodPr}} (\sigma_{\text{NrFact}=1117}(\text{LINIIFACT}))$$

Soluția 1 de la exemplul 14:

R1 ← **JONCȚIUNE** (**LOCALITATI**, **JUDETE**; **Jud**)
R ← **SELECȚIE** (**R1**; **Regiune** = "Banat")

$$\sigma_{\text{Regiune}=\text{"Banat"}}(\text{LOCALITATI} \bowtie_{\text{LOCALITATI.Jud}=\text{JUDETE.Jud}} \text{JUDETE})$$

Este drept, atunci când este de reprezentat o joncțiune naturală, se poate recurge la o redactare simplificată:

$$\sigma_{\text{Regiune}=\text{"Banat"}}(\text{LOCALITATI} \bowtie \text{JUDETE})$$

O altă observație, ține de faptul că optimizarea interogărilor nu prezintă aproape deloc importanță.

Soluția de la exemplul 16

R1 ← **SELECȚIE** (**PRODUSE**; **DenPr** = "Produs 1")
R2 ← **JONCȚIUNE** (**R1**, **LINIIFACT**; **CodPr**)
R3 ← **PROIECȚIE** (**R2**; **NrFact**)
R4 ← **SELECȚIE**(**FACTURI**; **DataFact** >= 03/08/2000 AND **DataFact** <= 03/08/2000)
R5 ← **JONCȚIUNE** (**R3**, **R4**; **NrFact**)
R6 ← **JONCȚIUNE** (**R5**, **CLIENTI**; **CodCl**)
R7 ← **PROIECȚIE** (**R6**; **CodPost**)
R8 ← **JONCȚIUNE** (**R7**, **LOCALITATI**; **CodPost**)
R9 ← **PROIECȚIE** (**R8**; **Jud**)
R10 ← **JONCȚIUNE** (**R9**, **JUDETE**; **Jud**)
R ← **PROIECȚIE** (**R10**; **Judet**):

$$\prod_{\text{Judet}} (\sigma_{\text{DenPr}=\text{"Produs 1"} \wedge \text{DataFact} \geq 03/08/2000 \wedge \text{DataFact} \leq 05/08/2000} (\text{PRODUSE} \bowtie \text{LINIIFACT} \bowtie \text{FACTURI} \bowtie \text{CLIENTI} \bowtie \text{LOCALITATI} \bowtie \text{JUDETE}))$$

Nu spun că acest stil de notare ar fi din cale-afară de dificil, dar eu, unul, consider că varianta "noastră" de redactare este mult mai lejeră.

4.5.2. Gramatica BNF

Notăția următoare este preluată din ediția a 4-a a lucrării lui C.J. Date⁵, și, cu oarecare amendamente din partea lui Date, reprezintă gramatica *Backus Naur Form*. Avantajele acestei notații țin, în primul rând, de utilizarea unor cuvinte din limba engleză, în locul abstractelor simboluri matematice. De asemenea, interogările se scriu liniar, fiind astfel mai lizibile. Pentru a indica precedența operațiilor se folosesc parantezele. Un avantaj major față de notația folosită pe parcursul acestui capitol ține de faptul că pune în evidență modul în care operațiile sunt incluse unele în altele, fiind mai aproape și de logica SQL (care folosește o singură frază SELECT în care, eventual, sunt incluse subconsultări).

Fără a intra în detalii privind gramatica BNF tratată în extenso în lucrarea autorului american, amintim că reprezentarea operatorilor se face astfel:

- *reuniune*: R1 UNION R2
- *intersecție*: R1 INTERSECT R2
- *diferență*: R1 MINUS R2
- *produs cartezian*: R1 TIMES R2
- *selecție*, denumită și *restricție*: R WHERE *P*, unde *P* este un predicat aplicat asupra relației R
- *proiecție*: R [A, B, ... Z], unde A, B, ... Z sunt atribute ale R
- *theta-joncțiunea* se reprezintă diferit de *joncțiunea naturală*, ca un operator compus dintr-un produs cartezian urmat de o selecție. Astfel, pentru (theta)joncțiunea relațiilor R1 și R2 prin predicatul: R1.A > R2.C, se folosește notația: (R1 TIMES R2) WHERE R1.A > R2.C
- pentru *joncțiunea naturală* lucrurile sunt mai simple; astfel, joncțiunea naturală dintre R1 și R2 prin atributul C (atributului comun) se simbolizează, pur și simplu: R1 JOIN R2.
- *diviziune*: R1 DIVIDED BY R2

Exemple

Soluția de la exemplul 7:

(CLIENTI WHERE DenCl = "Client 2 SA") [Telefon]

Soluția de la exemplul 9:

((LINIIFACT WHERE NrFact = 1111) [CodPr]) INTERSECT
((LINIIFACT WHERE NrFact = 1111) [CodPr])

Soluția 1 de la exemplul 14:

((JUDETE JOIN CODURI_POSTALE) WHERE Regiune = "Banat")

Soluția de la exemplul 16

((((((PRODUSE WHERE DenPr = "Produs 1") JOIN LINIIFACT) [NrFact])
JOIN FACTURI) WHERE DataFact >= 03/08/2000 AND DataFact <= 03/08/2000) JOIN
CLIENTI) JOIN CODURI_POSTALE) JOIN JUDETE) [Judet])

⁵ [Date86]

Probabil cel mai mare dezavantaj al acestei notații ține de faptul că s-ar putea să greșim la numărul parantezelor...

4.5.3. Notatii diverse

Literatura de specialitate a propus diverse notații, mai mult sau mai puțin apropiate de cele prezentate până acum. Dacă e să ne referim, numai la cărțile publicate la noi⁶ (bazate, ca și prezenta, pe lucrări de circulație internațională), putem aminti:

- *pentru reuniune:*

$R1 \cup R2$
OR (R1, R2)
APPEND (R1, R2)
UNION (R1, R2)
REUNIUNE (R1,R2)

- *pentru intersecție:*

$R1 \cap R2$
AND (R1, R2)
INTERSECT (R1, R2)
INTERSECȚIE (R1,R2)

- *pentru diferență:*

$R1 - R2$
REMOVE (R1, R2)
MINUS (R1, R2)
DIFERENȚĂ (R1,R2)

- *pentru produs cartezian:*

$R1 \times R2$
PRODUCT (R1, R2)
TIMES (R1, R2)
PROD (R1,R2)

- *pentru selecție:*

$\sigma_{\text{condiție}}(R)$
R [condiție]
RESTRICT (R, condiție)
SEL (R, condiție)

- *pentru proiecție:*

$\Pi_{A, B, C, \dots Z}(R)$
R [A,B, ..., Z]
PROJECT (R, A, B,)
PROI (R, A,B, ..., Z)

⁶ [Lungu s.a.95], [Popescu96], [Dollinger98], [Florescu s.a.99] – în treacăt fie spus, exemplul de la diviziunea relațională (pag.143) e cel puțin tulburător, [Grama&Filip00]

- pentru joncțiune:

$$R1 \bowtie_{\text{conditie}} R2$$

JOIN (R1, R2, condiție)

- pentru diviziune:

$$R1 \div R2$$

DIVISION (R1, R2)

4.5.4. Reprezentarea grafică a interogărilor

În multe lucrări de specialitate sunt utilizate o serie de simboluri grafice asociate operatorilor relaționali. Dintre acestea, cele mai importante sunt prezentate în figura 4.34. Aceste simboluri permit reprezentarea, în condiții de lizibilitate sensibil îmbunătățită, a logicii de derulare a unei interogări formulate prin operatorii relaționali.

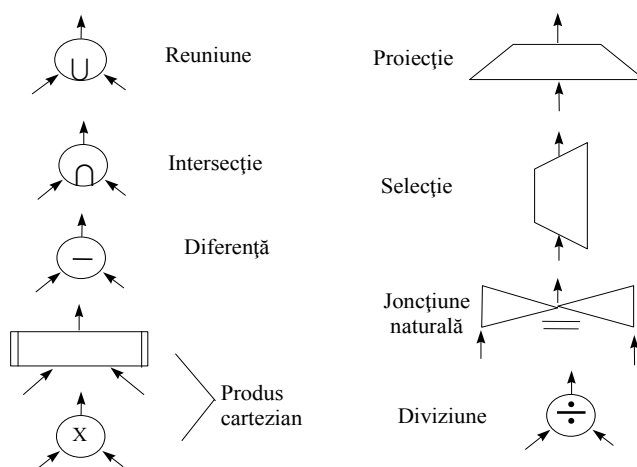


Figura 4.34. Simboluri pentru reprezentarea grafică a operatorilor algebrei relaționale

Pentru ilustrarea dispunerii simbolurilor în concordanță cu logica soluției, reluăm cele patru exemple folosite în acest paragraf.

Exemple

Soluția de la exemplul 7 – fig. 4.35:

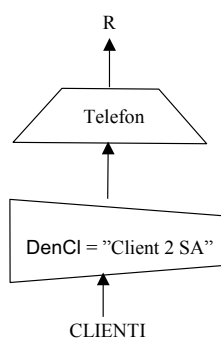


Figura 4.35. Reprezentare grafică – soluție ex. 7

Soluția de la exemplul 9 – figura 4.36:

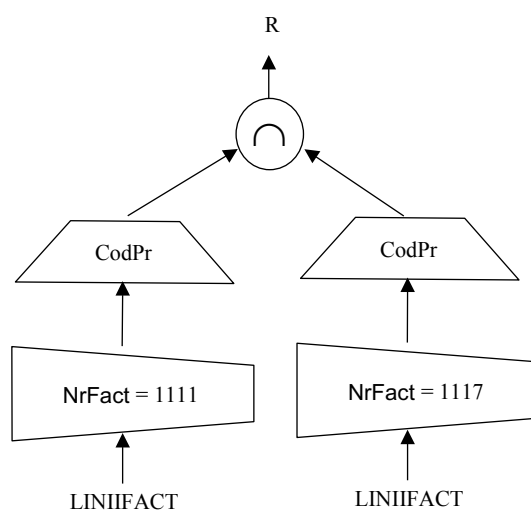


Figura 4.36. Reprezentare grafică – soluție ex. 9

Soluția 1 de la exemplul 14 – figura 4.37.

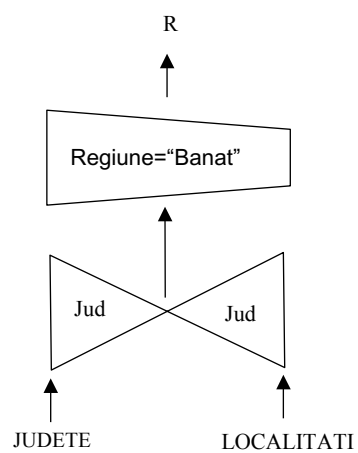


Figura 4.37. Reprezentare grafică – soluție 1 ex. 14