

Projektovanje informacionih sistema

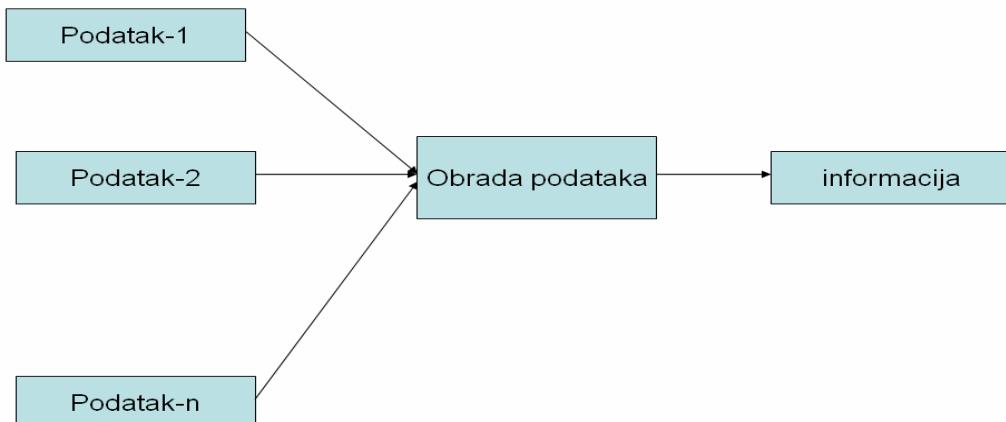
Prvi modeli za organizaciju, memorisanje, manipulaciju i obradu podataka pojavili su se šezdesetih godina XX veka. U centru ovih modela nalaze se dva pojma:

- Podatak
- Informacije

Osnovni pojmovi

Podatak je iskaz definisan prostom izjavnom rečenicom. **Informacija** je novi podatak koji poseduje relevantnu novinu, neko novo saznanje, a rezultat je obrade poznatih podataka. Postupak obrade podataka, dobijanje novih informacija i sticanja novih saznanja može se podeliti u 4 faze:

- Prikupljanje podataka
- Memorisanje i organizacija prikupljenih podataka
- Obrada podataka (računanje, sortiranje, grupisanje)
- Dobijanje novih informacija



Slika 1: dobijanje nove informacije iz niza poznatih podataka

Objekat posmatranja – entitet

Podaci koje sakupljamo, memorišemo, organizujemo i obrađujemo nalaze se u svetu oko nas i vezani su za neki proces koji se odvija u delu realnog sveta iz našeg okruženja. *Proces* po definiciji predstavlja promenu jedne ili više veličina u vremenu.

Definišemo naše viđenje za nas interesantnog dela realnog sveta. Tako kreiramo **model-objekat, entitet, ili samo objekat**. Svojstva objekata opisuju se preko **atributa** (koje moramo odabrat u fazi modelovanja objekata). Svojstva objekata opisuju se preko **atributa** (koje moramo odabrat u fazi modelovanja objekata). Skup dozvoljenih vrednosti koje neki atribut može imati naziva se **domen**.

Npr , objekat učenik opisuje se sa atributima – ocena, čiji je domen skup prirodnih brojeva od 1 do 5.

Prepoznavanje mere pri modelovanju procesa (pri izboru atributa) jedan je od snovnih zadataka projektanta informacionog sistema. Entitet ili objekat, po pravilu može biti veoma različit:

- **Deo okruženja (član kolektiva, aparat, zgrada...)**
- **Apstraktни pojam (neka mjera, nečije zvanje, boja..)**
- **Dogadjaj (udes, postupak upisa studenata...)**
- **Asocijacija (polaznik kursa, predmet-nastavnik)**

Svaka tabela mora da ima definisano

- Ime ili naziv tabele
- Spisak atributa
- Niz vrednosti atributa tj podatke

Tabela se sastoji od **polja** u koja se upisuju podaci. Slaganjem polja u jednom redu tabele dobijamo jedan **slog, red ili n-torku**. Skup svih redova (slogova), svih n-torki neke tabele čini **tijelo tabele**. Naziv atributa u jednoj tabeli može prema tome biti naziv druge tabele.

Preporučuje se da ime tabela kao i atributa treba da asociraju na prirodu procesa koji se u toj tabeli prati podacima. Atribut mora biti tako odabran (definisan) da se može iskazati samo jednom izričnom rečenicom, tj definisati jednim elementarnim (atomarnim) podatkom. Ako je za opis atributa potrebno više podataka, onda nije više u pitanju atribut, nego novi entitet sa svojim atributima.

Veze među objektima

Danas razlikujemo tri tipa veze među objektima

- **Veza tipa 1:1**
- **Veza tipa 1:N**
- **Veza tipa N:M**

Veza tipa 1:1

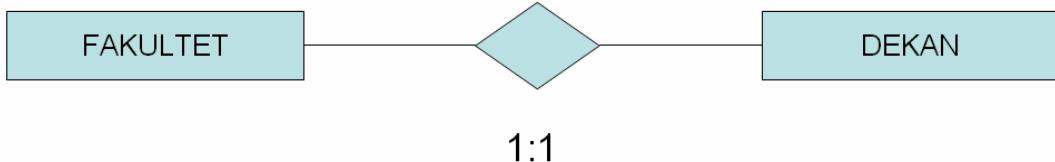
Pretpostavimo da je na nekom univerzitetu uspostavljen informacioni sistem u kome između ostalog postoje i 2 objekta: **fakultet i dekan**. Prva tabela FAKULTET sadrži atributе kojima se opisuju fakulteti tog univerziteta:

FAKULTET <naziv, adresa, telefon,>

DEKAN sadrži atributе koji pobliže definišu dekane fakulteta.

DEKAN <šifra_dekana, ime, prezime, adresa, telefon,..>

Ako je zakonom određeno da svaki fakultet može da ima samo jednog dekana, a da samo jedan profesor može biti dekan, onda je veza među tabelama tipa 1:1.

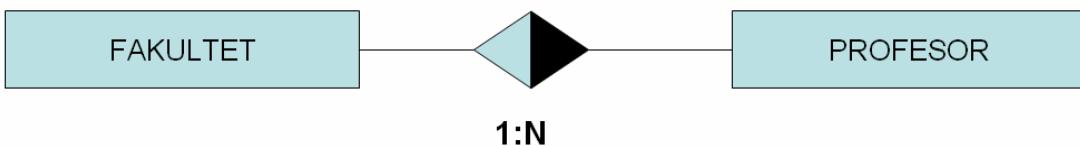


Veza tipa 1:N

U informacionom sistemu univerziteta može da postoji i objekat PROFESOR čiji atributi treba bliže opišu profesora.

PROFESOR <šifra_profesora, ime, prezime, zvanje, adresa,>

Ako zakonski propisi propisuju da jedan profesor može biti u radnom odnosu samo na jednom (1) fakultetu, a da svaki fakultet angažuje više (N) profesora, onda je veza između objekata FAKULTET i PROFESOR tipa 1:N.

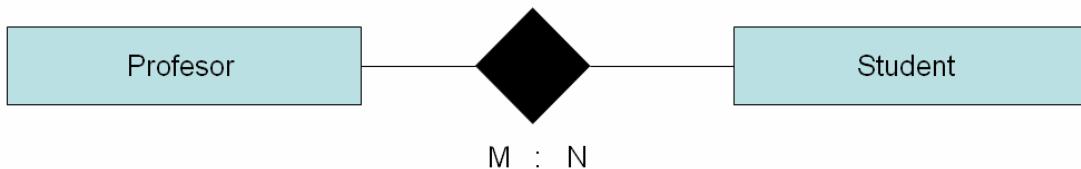


Veza tipa N:M

Proširenje ovog univerzitetskog informacionog sistema može se odnositi i na studente. O svakom studentu treba imati neke podatke.

STUDENT <br_{_ind}, ime, prezime, god_rod, adresa, itd>

Kako u toku studija svaki student dolazi u kontakt sa više (M) profesora, ali i svaki profesor drži predavanja većem broju (N) studenata, to je veza među objektima PROFESOR i STUDENT tipa M:N.



Modeli podataka

Pedestih godina prošlog veka pojavio se problem kako modelirati i objediniti objekte u jedan celoviti informacioni sistem. Do danas su razvijeni sledeći modeli:

- **Model prve generacije – bazirani na programskim jezicima**
- **Modeli druge generacije – hijerarhijski i mrežni**
- **Model treće generacije – relacioni model**
- **Model četvrte generacije – objektno orijentisani model**

Danas je u fazi projektovanja dominantan model **objekat-veze (MOV), (Entity-Relationship ili E-R model)**. Pri implementaciji na računar još uvek je dominantan relacioni model.

Hijerarhijski model podataka

Modeli, bazirani na programskim jezicima, održali su se kratko, takođe nije bilo ni vremena da se implementiraju iz razvojne u eksploracionu fazu. Prvi model koji je našao primenu i u praksi bio je **hijerarhijski model**.

Osnovni nedostatak hijerarhijskog modela je nepostojanje egzaktne matematičke teorije koja ga prati i koja bi omogućila njegovu punu implementaciju na računar. To je bio i podstrek E.F.Codd-u da početkom sedamdesetih godina prošlog veka definiše, dizajnira i prezentira danas najpopularniji *relacioni model*.

Mrežni model

Mrežni model je nastao kao proširenje i pokušaj poboljšanja hijerarhijskog. Osnovna razlika, u odnosu na hijerarhijski model, leži u činjenici da se dozvolilo da “naslednik” ima više “roditelja”, što znači da ovakav model prihvata i veze tipa M:N. I za mrežne strukture informacionih sistema važi većina nedostataka nevedenih kod hijerarhijskih, pa se zbog toga redje nalaze u praksi i gotovo da će za kratko vreme potpuno isčeznuti.

Relacioni model

Relacioni model je danas najrasprostranjeniji model informacionog sistema zahvaljujući sledećim osobinama:

- **Struktura modela je jednostavna a baza podataka je predstavljena samo skupom tabela i funkcionalnim vezama među njima**
- **Razvijena je matematička teorija koja omogućava jednostavnu interpretaciju ovog modela na računaru**

Ovaj model se zasniva na tabelama i relacijama među njima koje nisu određene hijerarhiski, ni mrežno, nego funkcionalno.

Objektni model

To je danas najrasprostranjeniji model informacionog sistema, pre svega u fazi projektovanja. Postoji čitav niz oblasti primene računara u kojima relacioni model nije pogodan za primenu. To su pre svega računarom podržano projektovanje, multimedijalni sistemi i baze znanja. On se zasniva na idejama objektno-orientisanih programskih jezika i semantičkih modela podataka.

Relacioni model – uvod

Principle i strukturu relacionog modela objavio je E.F. Codd 1970 u svom radu “*A Relation Model of Data for Large Data Banks*”. Osnovna prednost u odnosu na hijerarhijski i mrežni model, je u tome što se u potpunosti oslanja na matematičku disciplinu, tzv *relacionu algebru*, čime je omogućena računarska podrška, razvoj specifičnog softvera i obrada uz konzistentnost podataka.

Osnovne definicije

Za svaki objekat sistema koji je predstavljen tabelom:

- Kolone predstavljaju atributе (sa podacima)
- Redovi, n-torce ili slogove (sa podacima)

Osnovni pojmovi u relacionim bazama podataka:

- Relacija
- Entitet
- Atribut
- Domen
- Kardinalnost relacije
- Primarni ključ
- Sekundarni ključ

Pojam relacije u relacionom modelu

Relacija, osnovni element relacionog modela ustvari je sinonim (uz neka ograničenja) za tabelu ili datoteku. E.F.Codd – tabele koje se sastoje od kolona (atributa i vrednosti atributa – podataka) te redova (n-torki, slogova sa podacima). Postoje uslovi koje neka tabela mora da zadovolji da bi bila relacija:

- **Sve vrednosti podataka jednog atributa moraju biti istog tipa** – u relacionoj bazi podataka ne smeju postojati funkcionalne zavisnosti među atributima
- **Unutar jedne relacije ne smeju postojati dve identične n-torce** – dve n-torce sa identičnim vrednostima atributa – identičnim podacima. Dvostruko memorisanje podataka može biti štetno, a nije potrebno
- **Redosled n-torki u relaciji je proizvoljan**
- **Svi atributi unutar jedne relacije moraju imati različita imena, dok je redosled njihovog navođenja proizvoljan**

Entitet

Entitet je model dela relanog sveta opisanog ograničenim brojem atributa. Entitet je nešto što postoji i što se na jedinstven način može identifikovati. Svaki entitet mora imati jedno ili više svojstava (identifikator) koji ga jasno razlikuju od svih drugih entiteta.

Atribut

Atribut je jedno od svojstava entiteta (objekata) kojeg posmatramo i o kojem sakupljamo podatke. Svi podaci u jednom redu, jednom slogu, tj jednoj n-torci, definišu jednu **jedinku** datog objekta, dok jedna kolona opisuje jedno **svojstvo** svih jedinki.

Na primer, u objektu

SLUŽBENIK <JMBG, ime, prezime, dat_rod, adresa...>

svaka n-torka sadrži gore navedene podatke o jednom službeniku, dok kolona JMBG sadrži jedinstvene matične brojeve svih službenika.

Atributi u trenutku skupljanja podataka u relacionoj bazi podataka ne moraju obavezno imati poznate vrednosti i tada ih nazivamo NULL – vrednostima, a treba ih izbegavati. Pri projektovanju informacionog sistema, treba pažljivo i studiozno iz mnoštva mogućih atributa odabrat i definisati samo one koji opisuju objekat na zadovoljavajući način.

Domen atributa

Domen atributa je skup vrednosti koje on može da poprimi. U relaciji SLUŽBENIK, atribut “pol” može da poprimi samo dve vrednosti, “muško” i “žensko”.

Primarni ključ

Primarni ključ ili često samo ključ, je atribut (ponekad i skup atributa, tj kombinacija atributa) čija vrednost jednoznačno određuje samo jednu n-torku, samo jedan slog u nekoj relaciji-tabeli. To znači da jednoznačno “izdvaja” samo jedan red-jedan slog.

Pri tome ne postoji nijedan podskup tog skupa atributa koji obezbeđuje jedinstvenost (osobina *minimalnosti*). U jednoj relaciji ne smeju postojati dve različite n-torce sa istom vrednošću ključnog atributa. Ukoliko ne postoji atribut koji zadovoljava uslove za ključni atribut, ključ se može kreirati kombinacijom dva ili više atributa.

Primarni ključ mora biti: jedinstven, nepromenljivi uvek raspoloživ. Primarni ključ se često obeležava (markira) prilikom definisanja relacije. U literaturi se ključni atribut najčešće označava znakom #, isključivo iz razloga da bi tabela-relacija bila preglednija za korisnika.

Npr, u relaciji SLUŽBENIK, kandidat za primarni ključ je matični broj građanina (JMBG) jer je samo on jedinstven. Relacija SLUŽBENIK sa obeleženim ključem:

SLUŽBENIK <JMBG#, ime, prezime, dat_rod, adresa, tel>

Ukoliko u nekoj relaciji nismo u stanju da “izdvojimo” kandidata za ulogu primarnog ključa, problem se rešava na dva načina:

- **Ili definisanjem grupe atributa (složeni ključ) koji onda imaju jedinstvenu vrednost za svaki slog**
- **Uvođenjem novog, dodatnog atributa, “šifra”**

Neki softverski paketi (ORACLE..) dopuštaju da relacija sadrži i identične n-torce tj ne mora biti definisan primarni ključ. Razlog za to je unos podataka iz drugih izvora, npr iz programa za rad sa tabelama (EXCELL) koji nemaju mehanizme koji bi sprečili pojavu duplikata

Spoljni (strani) ključ

Spoljni ključ (npr u relaciji A) je atribut (skupo atributa) koji u drugoj relaciji (npr B) ima ulogu primarnog ključa. Osnovna uloga spoljnih ključeva je uspostavljanje veza (relacija) među tabelama, a ne identifikacija n-torki. Spoljni ključ (naziva se još i strani ključ), može ali i ne mora se, posebno obeležavati (sa ! Ili \$).

To se koristi samo u tekstu, jer to doprinosi vizuelnoj preglednosti modela. Jedino ako je strani ključ ujedno i deo nekog primarnog složenog ključa, onda ga treba obeležiti uobičajenom oznakom (#) za primarni ključ.

Osnove relacione algebre

Za manipulaciju podacima i tabelama u relacionim bazama podataka potrebna su osnovna znanja iz relacione algebre. Relaciona algebra spada u matematičku oblast teorije skupova i na njoj se bazira relacioni model baze podataka.

Operatori relacione algebre dele se u dve grupe:

- **Osnovni operatori**
- **Operatori pridruživanja**

Relacioni operatori su sa stanovišta matematičke teorije operatori visokog nivoa jer operišu sa relacijama, dakle sa skupovima vrednosti (tabelama), a ne samo sa jednom i kao rezultat daju opet relaciju – skup vrednosti – novu relaciju. E.F. Codd je relacione operatore podelio u dve grupe:

- **Operatore koji su pogodni za ažuriranje**
- **Operatore koji su pogodni za izveštavanje**

Operatori pogodni za ažuriranje

Tradisionalni operatori izvode se nad minimum dve relacije:

- **Unija (UNION)**
- **Presek (INTERSECT)**
- **Razlika (DIFFERENCE)**
- **Proizvod (CARTESIAN PRODUCT)**

Unija

Unija dva skupa, dve relacije A i B, je relacija koja se sastoji od svih elemenata koji pripadaju relacijama A ili B. Svaka relacija je po definiciji skup n-torki, pa je i unija dve relacije skup n-torki, ali ne mora u opštem slučaju biti relacija. Relacija ne sme sadržavati različite tipove n-torki. Pa se može teoretski napraviti unija od dve relacije koja ima različite attribute. Rezultat je u tom slučaju tabela, ali nije relacija. Zato se definišu ograničenja koja moraju biti zadovoljena kako bi nad dve relacije bila izvodljiva operacija unija, a da rezultat pri tome opet bude relacija. Za takve relacije se kaže da su **union-kompatibilne**

Ta ograničenju su:

- **Obe relacije moraju imati iste attribute**
- **Isti atributi moraju biti definisani nad istim domenom**

Opracija unija nad relacijama A i B simbolički se označava sa: AUB.

Tabela A/

ŠIFRA #	PREZIME	IME	TEL. BROJ
3244	Aksentijević	Petar	0710 334 952
1772	Maksimović	Ilija	015 723 543

Tabela B

ŠIFRA #	PREZIME	IME	TEL. BROJ
3244	Aksentijević	Petar	0710 334 952
2345	Petrović	Dara	081 17318

Tabela C=A \cup B

ŠIFRA#	PREZIME	IME	TEL. BROJ
3244	Aksentijević	Petar	0710 334 952
1772	Maksimović	Ilija	015 723 543
2345	Petrović	Dara	081 17318

Presek

Presek dve relacije A i B (označava se $A \cap B$) je nova relacija koja sadrži sve n-torce koje su zajedničke za obe relacije.

C=A \cap B

ŠIFRA #	PREZIME	IME	TEL. BROJ
3244	Aksentijević	Petar	0710 334 952

Razlika

Razlika A – B dveju relacija (označava se i sa A/B) je nova relacija koja ima iste atribute kao i relacije A i B, a telo se sastoji samo od onih n-torki koje se nalaze u relaciji A, a ne nalaze u relaciji B. Za razliku važi pravilo $A - B = B - A$.

Tabela A/

ŠIFRA #	PREZIME	IME	TEL. BROJ
3244	Aksentijević	Petar	0710 334 952
1772	Maksimović	Ilija	015 723 543

Tabela B

ŠIFRA #	PREZIME	IME	TEL. BROJ
3244	Aksentijević	Petar	0710 334 952
2345	Petrović	Dara	081 17318

A – B (A / B)

ŠIFRA #	PREZIME	IME	TEL. BROJ
1772	Maksimović	Ilija	015 723 543

B – A (A / B)

ŠIFRA #	PREZIME	IME	TEL. BROJ
2345	Petrović	Dara	081 17318

Proizvod

Pojam proizvoda u relacionoj algebri je nešto širi pojam od pojma prostog Dekartovog proizvoda. Naime, Dekartov proizvod dva skupa A i B, definiše se kao skup uređenih parova u kojem prvi element pripada skupu A, a drugi skupu B.

U relacionoj algebri u vek želimo da dobijemo uređen skup n-torki, a ne uređen skup parova. Zbog toga se umesto skupa elemenata uzima skup n-torki, pri čemu je svaka tako novodobojena n-torka rezultat spajanja uređenog para n-torki. Kod izvođenja proizvoda dve relacije postoji opasnost da dođe do greške ukoliko te dve relacije imaju atribute sa istim imenima, a nemaju isto značenje.

Primer proširenog Dekartovog/Kartezijskog proizvoda relacija ALFA i BETA.

ALFA		BETA			ALFA*BETA				
A	B	C	D	E	A	B	C	D	E
a ₁	b ₁	c ₁	d ₁	e ₁	a ₁	b ₁	c ₁	d ₁	e ₁
a ₂	b ₂	c ₂	d ₂	e ₂	a ₁	b ₁	c ₂	d ₂	e ₂
a ₃	b ₃				a ₂	b ₂	c ₁	d ₁	e ₁
					a ₂	b ₂	c ₂	d ₂	e ₂
					a ₃	b ₃	c ₁	d ₁	e ₁
					a ₃	b ₃	c ₂	d ₂	e ₂

Ako bi želeli da napravimo proizvod relacija:

PROFESOR <šifra#, ime, prezime, zvanje, adresa..>

STUDENT <šifra#, ime, prezime, adresa>

to ne bi bilo moguće, jer se imena atributa (ime, prezime i adresa) ponavljaju.

Rešenje je u preimenovanju atributa u jednoj relaciji npr u relaciji STUDENT:

STUDENT <šifra#, imest, prezimest, adresast..>

Operatori pogodni za izveštavanje

U specijalne operatore spadaju

- **Selekcija**
- **Projekcija**
- **Spajanje**
- **Deljenje**

Selekcija

Selekcija (ograničenje ili restrikcija) izdvaja iz relacije samo one n-torke koje zadovoljavaju zadati kriterijum (uslov), koji je definisan logički. N-torke u kojoj je taj logički uslov zadovoljen, definišu onda novu relaciju.

Projekcija

Projekcija relacije daje novu relaciju koja se sastoji samo od određenih (ili samo jednog) atributa zadane relacije.. Rezultat operacije projekcija je podskup izabranih atributa neke relacije sa svim njenim n-torkama. Često se dešava da se primenom projekcije iz nesmotrenosti, mogu izgubiti neki podaci jer u novonastaloj tabeli mogu da se pojave identične n-torke (što u slučaju selekcije nije moglo da se desi), koje onda u nekim softverskim paketima bivaju bez upozorenja brisane, kako bi dobijeni rezultat bio relacija.

Spajanje (join)

Operacija spajanja ima više podvrsta od kojih su dve najvažnije:

- **Prirodno spajanje**
- **Spajanje pod nekim uslovom**

Prirodno spajanje relacija A i B daje relaciju AB koja ima sve atribute relacije A, i one atribute relacije B koje nema relacija A

A <x1,x2,..xn,y1,y2,...ym>

B <y1,y2,.....,ym,z1,z2,...zp>

AB <x1,x2,..,xn,y1,y2,...,ym,z1,z2,...zp>

Ako relacije koje se prirodno spajaju nemaju ni jedan zajednički atribut, onda operacija prirodnog spajanja prelazi u Kartizijev-Dekartov proizvod.

Šifrad#	Naziv	Mesto
D001	Comex	Toronto
D002	Unita	Vancouver
D003	Dual	Beograd

Šifrad#	Šifrap#	Broj kom.
D001	p991	324
D002	p678	23
D003	p007	12564

Šifrad#	Naziv	Mesto	Šifrap#	Broj kom.
D001	Comex	Toronto	p991	324
D002	Unita	Vancouver	p678	23
D003	Dual	Beograd	p007	12564

Spajanje pod nekim uslovom Ψ izvodi se nad relacijama samo onda kada one nemaju nijedan isti atribut. Rezultat spajanja je u tom slučaju Kartezijev proizvod tih relacija koji sadrži samo one n-torce koje zadovoljavaju logički uslov definsan izrazom Ψ .

Operacija deljenja

Deljenje se ne može izvesti sa proizvoljnim relacijama – tabelama. Da bi operacija A podeljeno sa B (A:B) bila izvodljiva, potrebno je da se svi atributi relacije B nalaze u A:

A <x₁,x₂,...x_n,y₁,y₂,...,y_m>
B <y₁,y₂,...,y_m>

Rezultat deljenja će biti relacija C koja ima samo x-tribute, a telo joj se sastoji od onih n-torki relacije A za koje se vrednosti y-tribute pojavljuju u relaciji B.

A		:		B		=		C	
X#	Y#			Y#				X#	
017	a22			a43				033	
033	a43							061	
077	a86							044	
061	a43								
044	a43								

Dodatni operatori

U modernoj relacionoj algebri postoji još nekoliko dodatnih, izvedenih, operatora koji su definisani posle E.F. Codd-a. Danas se koriste i operatori:

- **Proširenja**
- **Agregacije**
- **Uopštenog deljenja**
- **Spoljnog spajanja**
- **Uslovni operator (MAYBE)**

Operator MAYBE koji se koristi za manipulaciju Null vrednostima i predstavlja proširenje klasične logičke algebre. U klasičnoj logičkoj algebri postoje samo dve moguće vrednosti, dva stanja, koje logička promenljiva može uzeti *istina* (TRUE), *laž* (FALSE).

Uvođenje Null vrednosti definisana je treća mogućnost, označena sa U (unknown-nepoznato) pa logičke operacije AND, OR, NOT rezultiraju sada sa tri stanja:

- T (TRUE)
- F (FALSE)
- U (UNKNOWN)

