**Uvod u relacione baze podataka**

**Neformalni pogled na relacioni model**

Intuitivno, **relacioni model** predstavlja način gledanja na podatke. Sadrži pravila za predstavljanje podataka (preko tabela) i pravila za rad sa tim podacima (izdvajanje, spajanje...)

**Aspekti relacionog modela**

* Aspekt strukture – svi podaci u bazi se korisniku predstavljaju isključivo u obliku tabela
* Aspekt integriteta – tabele zadovoljavaju izvesna ograničenja (primarni i spoljašnji ključevi...)
* Aspekt obrade – operatori koji su na raspolaganju korisnicima za obradu tabela su takvi da izvode tabele iz tabela

**Primeri relacionih operatora**

**Restrikcija** (selekcija) – izdvaja pojedinačne redove iz tabele

DOSIJE WHERE

DATUM\_RODJENJA > ’28.02.1995’

**Projekcija** – izdvaja pojedinačne slogove iz tabele

DOSIJE over Indeks, Ime, Prezime

**Spajanje** – kombinuje dve tabele na osnovu zajedničkih vrednosti u zajedničkoj koloni

DOSIJE AND ISPIT OVER Indeks (primer prirodnog spajanja)

* + Zajedničke vrednosti se javljaju samo u jednom redu
  + Javljaju se sve kombinacije uparenih vrednosti
  + Ne javljaju se redovi sa vrednostima koje se ne nalaze u obe tabele koje učestvuju u spajanju

**Karakteristike relacione baze**

* Relacioni sistemi zahtevaju samo da se baza prikaže korisniku u obliku tabele, način smeštanja i čuvanja na medijumima nije specifiran
* Informacioni princip – Celokupan informacioni kontekst baze se prikazuje na tačno jedan način kao eksplicitne vrednosti u pozicijama vrsta i kolona tabele

Posledica ovog principa je da nema pokazivača koji međusobno povezuju tabele, ali pokazivači mogu da postoje na fizikom nivou.

!! Ovo je jedna od razlika u odnosu na nerelacione modele gde su pokazivači deo struktura koje se prikazuju korisnicima i koje korisinik koristi u obradi

**Efekat relacionih operatora**

* Rezultat primene svakog operatora je tabela – **osobina zatvorenja relacionih sistema**. Rezultat primene operatora je istog tipa kao i njegov argument iz čega sledi da mogu da se pišu ugnježdeni relacioni izrazi.
* Sve operacije se primenjuju istovremeno na ceo skup, a ne samo na pojedinačni red. Rezultat operacije nikada nije pojedinačni red već je uvek kompletna tabela koja sadrži skup redova.

**Formalni pogled na relacioni model**

Relacioni model se sastoji od:

* Otvorenog skupa skalarnih tipova (koji uključuje i tip logičkih vrednosti boolean) – otvoren skup znači da korisnik može da definiše svoje tipove
* Generatora relacionih tipova i njihove odgovarajuće interpretracije
* Mogućnosti definisanja relacionih promenljivih za generisane relacione tipove
* Operacije relacione dodele kojom se dodeljuju relacione vrednosti definisanim relacionim promenljivim
* Otvorenog skupa opštih relacionih operatora („relaciona algebra“) za izvođenje relacionih vrednosti iz drugih relacionih vrednosti

**Terminologija**

Codd je prilikom formulisanja principa relacionih baza uveo novu terminologiju koja se razlikovala od neodređene terminologije tog vremena

Relacija – matematički termin za tabelu

Torka – red u tabeli

Atribut – kolona u tabeli

Kardinalnost – broj torki

Stepen – broj atributa

Primarni ključ – atribut ili kombinacija atributa koja jedinstveno identifikuje tabelu

Domen – skup važećih vrednosti (tipova)

**Relacije i relacione promenljive**

U definiciji baze studenata, predmeta i ispita, DOSIJE i ISPIT su relacione promenljive, odnosno promenljive čije su vrednosti relacione vrednosti.

**Terminologija:**

Relvar = relaciona promenljiva

Relacija = specifična relaciona vrednost

**Optimizacija**

Relacioni jezici su **neproceduralni**

!! Za relacioni sistem često se kaže da vrši **automatsku „navigaciju“** nad sačuvanim podacima, dok je u nerelacionim sistemima korisnik odgovoran za navigaciju.

Primer:

DOSIJE WHERE Indeks = Indeks(20100027) {Ime}

Za ovakav upit rezultat se dobija:

* Restrikcijom tekuće vrednosti relvara DOSIJE na redove u kojima je broj indeksa jednak 20100027
* Projekcijom dobijenih vrednosti na kolonu ime

Postoje bar dva načina pristupa obrade ovakvog upita:

1. Vršenje fizičkog skeniranja (sačuvane vrednosti) relvara DOSIJE dok se ne nađe traženi podatak
2. Ako postoji indeks nad (sačuvanom verzijom) kolone indeks (a obično postoji jer je indeks primarni ključ) tada se on koristi za direktan pristup traženim podacima

Odluka kako vršiti automatsku navigaciju spada u domen komponente SUBP koja se zove **optimizator**. U opštem slučaju optimizator određuje strategiju na osnovu razmatranja:

* Koji relvari su referencirani u zahtevu i koliko su oni veliki
* Koji indeksi postoje
* Kako su podaci fizički smešteni na disku

Itd.

**Katalog**

Katalog („rečnik“) se i sam sastoji od relvara – katalogu se pristupa na isti način kao i drugim relacionim promenljivim

Informacije o samom katalogu se nalaze u relvarima iz kataloga

**Osnovni relvari i pogledi**

Originalni relvari = **osnovni relvari**

Njihove vrednosti = **osnovne relacije**

**Izvedene relacije** = relacije koje nisu osnovne ali mogu da se dobiju iz osnovnih putem relacionih izraza

Osnovni relvari moraju da imaju ime.

Relacione baze imaju mehanizam za formiranje osnovnih relvara.

U SQL-u su osnovni relvari tabele. Njihovo kreiranje se vrši naredbom:

CREATE TABLE <ime\_osnovnog\_relvara>

Relacioni sistemi podržavaju i imenovane relvare koji su izvedene relacije – **poglede**

**Vrednost pogleda** je (trenutni) rezultat izvršavnja određenog relacionog izraza koji se navodi pri formiranju pogleda.

CREATE VIEW Studenti\_iz\_Kraljeva AS

(DOSIJE WHERE Mesto\_rodjenja = ‘Kraljevo’) {Indeks, Ime, Prezime}

Nad pogledima mogu da se vrše operacije kao nad osnovnim tabelama:

(Studenti\_iz\_Kraljeva WHERE Indeks > 20140023){Ime, Prezime}

Pri izvršavanju sistem modifikuje upit u:

((DOSIJE WHERE Mesto\_rodjenja=’Kraljevo’) {Indeks, Ime, Prezime})WHERE Indeks > 20100023){Ime, Prezime}

Odnosno, uprošćava u:

(DOSIJE WHERE Mesto\_rodjenja=’Kraljevo’AND Indeks > 20100023){Ime, Prezime}

Osnovni relvari predstavljaju podatke koji su **fizički smešteni** u bazi podataka (zaista postoje), dok pogledi prestavljaju **različit način gledanja na realne podatke** (ne postoje, virtualni relvari)

**Transakcije**

Transakcija je logička jedinica posla koja obično uključuje više operacija nad bazom

Svojstva:

1. Atomičnost – garantuje da će se izvršiti sve što se nalazi u transakciji ili ništa od toga
2. Trajnost – garantuje se da će po uspešnom izvršavanju (commit-u) sve promene ostati trajno upamćene u bazi, bez obzira na kasnije eventualne padove sistema
3. Izolovanost – Transakcije su međusobno izolovane u smislu da su efekti izvršavanja jedne transakcije nevidljivi za drugu sve do uspešnog izvršavanja COMMIT naredbe
4. Izvršavanje isprepletanog (u smislu početka i kraja) skupa transakcija će obično biti serijalizovano u smislu da se dobija isti rezultat kao da se te iste transakcije izvršavaju jedna po jedna u unapred neodređenom redosledu .

**Domeni**

Domen = tip podataka

**Ugrađen (sistemski predefinisan)** – Integer, Char, Smallint, Date...

**Korisnčki definisan** – INDEKS, IME, GOD\_RODJENJA…

Provera tipova – stroga tipiziranost.

Dosije.indeks + Ispit.ocena --- pogrešno

Ispit.godina\_roka \* Ispit.ocena --- ispravno

Definisanje novog tipa:

TYPE <ime\_tipa> <moguca\_reprezentacija>

TYPE INDEKS REPREZENT (INTEGER);

TYPE IME REPREZENT (CHAR);

**SQL podška za domene**

Naredbe: CREATE DOMAIN, ALTER DOMAIN, DROP DOMAIN

U DB2: CREATE TYPE, ALTER TYPE, DROP TYPE

Istovremeno se definišu i odgovarajuće funkcije i operatori poređenja. Nema stroge tipiziranosti, ali postoji osnovni oblik provere tipova.

SQL podržava osam relacionih domena:

1. Brojevi (numbers)
2. Niske karaktera (character strings)
3. Niske bitova (bit strings)
4. Datumi (dates)
5. Vremena (times)
6. Kombinacija vremena i datuma (timestamps)
7. Intervali godina/mesec (year/month intervals)
8. Intervali dan/vreme (day/time intervals)

**Relacije**

Neka je dat skup od n tipova ili domena T, (i=1,2..n) ,pri čemu ne moraju svi tipovi da budu međusobno različiti. R je relacija nad tim tipovima ako se sastoji od dva dela, zaglavlja i tela i važi:

1. Zaglavlje je skup od n atributa Ai : Ti gde su Ai (koji svi moraju da budu različiti) imena atributa relacije R, a Ti odgovarajuća imena tipova
2. Telo je skup od m torki t gde je t skup komponenti oblika Ai : vi u kojima je vi vrednost tipa Ti

m – kardinalnost relacije R

n – stepen (arnost) relacije R

**Osobine relacije:**

* Nema ponovljenih (duplih) torki
* Torke su neuređene, od vrha ka dnu
* Atributi su neuređeni, sa leva u desno
* Svaka torka sadrži tačno jednu vrednost za svaki atribut. Za relaciju koja zadovoljava ovu osobinu kaže se da je normalizovana, odnosno da je u prvoj normalnoj formi.

**Da li su relacija i tabela jednake?**

Ne, jer

* Tabela može da sadrži duplirane redove, dok relacije ne može da sadrži duplirane torke
* Redovi u tabeli su uređeni od vrha ka dnu, dok za relaciju to ne važi
* Kolone u tabeli su uređene sa leva u desno, dok sa relaciju to ne važi

**Relacije i prazan skup**

Relacija koja ima **prazan skup torki** ima neprazno zaglavlje, ali je telo prazno.

Relacija koja ima **praznu torku** ima prazno zaglavlje i telo sa jednom torkom bez komponenti.

**Tipovi relacije**

Definicija tipa relacije ima sledeći oblik:

RELATION {<lista atributa razdvojenih zarezima>}

Atribut jer uređen par oblika:

<ime atributa> <ime tipa>

Npr:

Relation {Indeks INDEKS, Ime IME, Prezime PREZIME, God\_rodjenja GOD\_RODJENJA, Mesto\_rodjenja MESTO\_RODJENJA}

**Atributi i tipovi podataka**

* Svaki atribut ima neki tip
* Svaki tip može da bude ugrađen ili korisnički definisan
* Svaki tip može da bude atomski(skalarni) ili učauren (nije skalarni)
* Atributi relacije mogu da budu proizvoljnog tipa

**Nedostajuće vrednosti**

U svakodnevnoj praksi se često javlja problem nedostatka podataka, te postoji potreba da se u bazi čuva indikator o nedostatku vrednosti, kao i da se na odgovarajući način vrši obrada takvih podataka.

Najčešći pristup je korišćenje **‘nedostajuće vrednosti’ (NULL**) odnosno **trovalentne logike (3VL)**

Codd je predložio 4VL logiku jer postoje 2 vrste nedostajućih vrednosti:

1. Vrednost je nepozanata
2. Vredost nije primenljiva, vrednost ne postoji

**3VL logika**

3 vrednosti: Tačno, Netačno i Nepoznato

Operatori: I, ILI, NE, MOŽDA (AND, OR, NOT, MAYBE)

**Relaciona algebra**

Algebra je formalni matematički sistem koji se sastoji od skupa objekata i operacija nad tim objektima. Npr: Bulova algebra, algebra skupova...

Za definiciju formalnog sistema je potrebno:

Prikazati sintaksu

Dati semantiku

Dati pravila izvođenja dokaza

**Relaciona algebra**

To je familija algebri sa dobro zasnovanom semantikom koja se koristi za modeliranje relacija (objekata) smeštenih u relacionoj bazi podataka i za definisanje upita nad njima. U suštini predstavlja skup operatora čiji su operandi i rezultati relacije.

Prvu verziju je dao Codd 1972. God.

Relacioni operatori:

Codd. Je prvobitno predložio 8 operatora:

* Restrikcija (selekcija)
* Projekcija
* Proizvod
* Unija
* Deljenje
* Presek
* Razlika
* (Prirodno) Spajanje

Kasnije su dodati:

* Promena imena
* Poluspajanje
* Polurazlika
* Ekskluzivna unija
* Proširenje
* Slika relacije
* Operatori agregata
* Sumarizacija
* ...

Minimalni skup operatora sadrži restrikciju (selekciju), projekciju, proizvod, uniju, razliku.

**Sintaksa**

Relacioni izraz je oblika:

ROP arg1, arg2....argn

ROP – relacioni operator

Argi – relacije koje su argumenti relacionog operatora

**Semantika**

U opisu semantike se koristi da su relacije matematički zasnovane i da predstavljaju skupove torki, kao i da su u pitanju operacije nad skupovima koji predstavljaju preslikavanje domena relacije u novi domen.

**Relaciono zatvorenje**

Osobina da su i argumenti i rezultat primene bilo kog relacionog operatora takođe relacije se naziva **relaciono zatvorenje.**

Zatvorenje znači da mogu da se pišu ugnježdeni relacioni izrazi, tj. relacioni izrazi čiji su operandi takođe relacioni izrazi.

Treba obezbediti da i novodbijene relacije imaju odgovarajuće zaglavlje (sa jedinstvenim nazivima atributa) i odgovarajuće telo, bez obzira da li su u pitanju osnovne ili izvedene relacije.

**Relacioni operatori**

**Restrikcija (selekcija)** – Traženje torki koje zadovoljavaju zadati uslov. Neka relacija A ima bar atribute X i Y i neka je teta operator (obično „=“, “<”, itd.) takav da je uslov X teta Y dobro definisan i da se izračunava kao istinitosna vrednost. Tada je teta restrikcija relacije A na atribute X i Y koja ima isto zaglavlje kao i A i telo koje sadrži sve torke t iz A za koje je vrednost uslova X teta Y tačno. Primer: dosije WHERE prezime = ‘Petrovic’

**Projekcija** – izdvajanje željenih atributa. Neka relacija A ima bar atribute X, Y, ... Z. Tada se projekcija relacije A na X,Y,...,Z označava sa A{X, Y,...Z} I predstavlja relacije čije:

Zaglavlje je izvedeno iz A uklanjanjem svih atributa koji se ne nalaze u skupu {X, Y,...Z}.

Telo sastoji od svih torki {X:x, Y:y,…,Z:z} pri čemu se svaka torka javlja u A sa X vrednošću x, Y vrednošću y, ..., Z vrednošću z. Ako su u listi navedeni svi atributi relacije A tada je projekcija identitet. Primer: dosije{ime, prezime, datum\_rodjenja}

**Prirodno spajanje** – Neka relacije A I B imaju sledeća zaglavlja:

A: {X1, X2, …, Xm, Y1, Y2,…, Yn}

B:{Y1, Y2,…,Yn, Z1, Z2,…Zp}

Tada je prirodno spajanje relacija A i B definisano sa A JOIN B = {{X:x, Y:y, Z:z} | {X:x, Y:y} e A ^ {Y :y, Z :z} e B}. Primer : dosije JOIN ispit

Postoji i teta spajanje za torke čiji atributi zadovoljavaju uslov X teta Y. Ako je teta = „=“ tada se ovo spajanje naziva jednakosno spajanje.

Deljenje – Sintaksa: A DIVIDEDBY B PER C. Semantika: Neka naredne tri relacija imaju sledeća zaglavlja:

A: {X1,X2,…Xm}

B: {Y1,Y2,…Yn}

C:{X1,X2,…Xm,Y1,Y2,…Yn}

Tada je deljenje A sa B po C definisano kao : DIV(A, B, C) = {{X :x} e A | V{Y :y} e B : {X :x, Y :y} e C}

**Unija** – Sintaksa : A UNION B. Semantika : A U B = {t | t e A V t e B}

**Presek** – Sintaksa: A INTERSECT B. Semantika: A ∩ B = {t | t e A ^ t e B}

**Razlika** – A DIFFERENCE B. Semantika: A – B = {t | t e A ^ t !e B}

**Mnozenje** (**Dekartov** **proizvod**) – Sintaksa: A TIMES B. Semantika: Neka relacije A i B imaju sledeća zaglavlja:

{A1,…Am}

{B1,…Bn}

Tada proizvod A TIMES B ima zaglavlje {A1,…Am,B1,…Bn} i važi A TIMES B = {t U t’ | t e A ^ t’ e B}

**Svrha relacione algebre**

* Pisanje relacionih izraza koji se koriste za:
  + Definisanje prostora za dohvatanja podataka
  + Definisanje prostora za ažuriranje podataka
  + Definisanje pravila integriteta
  + Definisanje izvedenih relaicja
  + Definisanje pravila zaštite
* Osnova za optimizaciju upita

**Relaciona kompletnost**

Jezik je relaciono kompletan ako je moćan isto kao i algebra, tj. ako bilo koja relacija predstavljena u algebri može da se predstavi i u tom jeziku.

SQL je relaciono kompletan jer postoje SQL izrazi za svaki od 5 primitvnih operatora relacione algebre.

A WHERE P – SELECT \* FROM A WHERE P

A {x, y,…z} – SELECT DISTINCT x, y,...y FROM A

A TIMES B – A CROSS JOIN B

A UNION B – SELECT \* FROM A UNION SELECT \* FROM B

A MINUS B – SELECT \* FROM A EXCEPT SELECT \*FROM B

Algebarski zakoni

Zakon asocijacije: (A UNION B) UNION C = A UNION (B UNION C), INTERSECT, TIMES, JOIN

Zakon komutacije: A UNION B = B UNION A, INTERSECT, TIMES, JOIN

**Relacioni račun**

To je opisan, neproceduralan jezik.

Logički ekvivalent relacione algebre ako se posmatra deo relacionog modela podataka za obradu podataka. Zasnovan je na predikatskom računu.

Dve varijante:

1. Račun orijentisan ka torkama
2. Račun orijentisan ka domenima – osnova za QBE

Predikatski račun

Predikat je istinitosno vrednosna funkcija sa argumentima. Kada se argumenti zamene vrednostima funkcija daje izraz koji se naziva predlog koji može biti tačan ili netačan.

Opseg promenljivih:

Promenljiva torki ima opseg iz skupa navedenih relacija i dopuštene vrednosti koje pripadaju torkama iz tih relacija.

Promenljiva domena ima opseg iz skupa navedenih domena i vrednosti koje pripadaju tim domenima.

Neka je x predikat. Tada se skup svih x takav da je je P tačno za x označava sa {x | P(x)}.

Postoje dva kvantifikatora: V – za svaki, E – postoji.

Relacioni račun torki

Slobodne i vezane promenljive. Kvantifikatori: FORALL V(P), EXISTS V(P).

Kvantifikacija i rad sa slobodnim i vezanim promenljivama su u skladu sa pravilima predikatskog računa.

Korišćenjem promenljivih torki traže se torke za koje je predikat tačan. Sintaksa ovog računa se u literaturi prikazuje na različite načine.

Primer:

RANGE OF <promenljiva> IS <tabele>

RETRIEVE <promenljiva>.<ime\_atributa>

[WHERE <uslovni izraz>]

Ili

RANGEVAR <promenljiva> RANGES OVER <tabela>  
<promenljiva>.<ime\_atributa>

[WHERE <uslovni\_izraz>]

RANGEVAR DOSIJEX RANGES OVER DOSIJE

{DOSIJEX.IME, DOSIJEX.DATUM\_RODJENJA}

WHERE DOSIJEX.MESTO\_RODJENJA = ‘Beograd’

AND DOSIJEX.INDEKS > 20110456

RANGEVAR DX RANGES OVER DOSIJE

RANGEVAR PX RANGES OVER PREDMET

RANGEVAR IX RANGES OVER ISPIT

DX.IME

WHERE EXISTS IX (DX.INDEKS = IX.INDEKS AND EXISTS PX(PX.ID\_PREDMETA = IX.ID\_PREDMETA AND PX.SIFRA = ‘P270’))

Relacioni račun domena

Opsega važenja promenljivih su domeni, a ne relacije.

Moguće je definisati uslov pripadnosti.

Oblik R (lista\_parova)

R – naziv relvara

Svaki par u listi je oblika A x gde je A naziv atributa u R, a x ili ime promenljive torki ili poziv selektora.

Uslov je tacan akko postoji torka u relaciji R takva da je za svaki konkretan par poredjenje A = x tačno.

Npr:

ISPIT {INDEKS INDEKS(20110456), ID\_PREDMETA ID\_PREDMETA(1001)}

Ima vrednost tačno akko postoji torka u ispitu koja ima vrednost 201100456 za atribut indeks I 1001 za atribut predmeta.

Algebra ili račun

Algebra I račun su semantički ekvivalentni. Kodov algoritam redukcije je prikaz da je algebra moćna bar koliko i račun i obratno. Neki upitni jezici su zasnovani na algebri, a neki na računu.

SQL ima osobine oba.

**Dodatni operatori relacione algebre**

Razlozi za uvođenje:

Jednostavnije rešavanje i zapis pojedinih upita

Zahtevi iz prakse

Potrebne osobine:

ROP arg1,..argn

Rezultat primene operacije je relacija – princip zatvorenja

Operatori:

* **Ekskluzivna** **unija** – Neka su relacije r1 i r2 istog tipa. Tada je njihova ekskluzivna unija u oznaci r1 XUNION r2 relacija istog tipa, sa telom koje sadrži sve torke t takve da pripadaju samo jednoj od relacija r1 ili r2. n-adična definicija: Neka su relacije r1, r2...rn (n>=0) istog tipa T. Tada izraz XUNION{r1, ...rn} označava relaciju tipa T čije telo sadrži skup svih torki t takvih da t pripada tačno m relacija ri, pri čemu je m paran broj (moguće različit za različite torke t)
* **Poluspajanje** – Poluspajanje relacija r1 i r2 (u tom redosledu), u oznaci r1 MATCHING r2 je relacija ekvivalentna relaciji koja se dobija kao rezultat: (r1 JOIN r2){H1} gde je H1 zaglavlje relacije r1.
* **Polurazlika** – Polurazlika relacija r1 i r2 (u tom redosledu), u oznaci r1 NOT MATCHING r2 je relacija ekvivalentna relaciji koja se dobija kao rezultat r1 MINUS (r1 MATCHING r2). Ako su r1 i r2 istog tipa, tada se polurazlia degenerise u razliku r1 MINUS r2.
* **Proširenje** – Neka je r relacija koja ne sadrži atrbitu sa imenom X. Tada je proširenje EXTEND r : {X := izraz} relacija čije zaglavlje uključuje zaglavlje relacije r prošireno atributom X, dok telo sadrži sve torke t relacije r proširene sa vrednošću atributa X dobijenom izračunavanjem vrednosti izraza izraz nad torkom t. Primedba: Kardinalnost nove relacije je ista kao i kardinalnost relacije r, dok je stepen za jedan veći. Tip novog atributa u relaciji je tip izraza izraz. Primer: ((EXTEND PREDMET : {DUPLI\_BODOVI := BODOVI \* 2})

WHERE DUPLI\_BODOVI > 12) {NAZIV, DUPLI\_BODOVI}

* Slika reacije – Neka su relacije r1 i r2 takve da mogu da se spoje (njihovi atributi sa istim imenom su istog tipa) i neka su t1 torka relacije r1 i t2 torka relacije r2 pri čemu imaju iste vrednosti zajedničkih atributa. Neka je relacija r3 restrikcija relacije r2 koja sadrži sve takve torke t2 i ništa sem njih, i neka je relacija r4 projekcija relacije r3 koja sadrži samo zajedničke atribute. Tada se za relaciju r4 kaže da je slika relacije u oznaci „!!“ koja odgovara t1 u odnosu na r2. Primer: DOSIJE

WHERE (!!ISPIT) {ID\_PREDMETA} = PREDMET {ID\_PREDMETA}

r1 I r2 – DOSIJE i ISPIT

WHERE uslov se računa za svakog studenta – torka t1

Izraz !!ISPIT označava sliku relacije r4 unutar r2 (sa atributima relacije ISPIT bez INDEKS-a)

Izraz (!!ISPIT) {ID\_PREDMETA} je projekcija slike na {ID\_PREDMETA} – skup predmeta koje je student polagao

* Sumarizacija – Neka su relacije r i s takve da s ima isto zaglavlje kao i neka projekcija od r, neka relacija r ne sadrži atribut X i neka su A,B,...C atributi relacije s. Tada sumarizacija

SUMMARIZE r PER (S) : {X := zbir}

je relacije čije zaglavlje uključuje zaglavlje relacije s prošireno atributom X, dok telo sadrži sve torke takve da je t torka relacije s proširena vrednošću x atributa X. Vrednost x se izračunava sumiranjem preko svih torki relacije r koje imaju iste vrednosti atributa A,B...C kao i t. Primedba: Kardinalnost nove relacije je ista kao i kardinalnost s, dok je stepen za 1 veći. Tip novog atributa je tip izraza zbir. SUMMARIZE ima funkciju vertikalnog izračunavanja, kao što EXTEND ima funkciju horizontalnog izračunavanja. GROUP BY u SQL.

* Deljenje (dodatna definicija) – Neka su relacije r1 i r2 takve da zaglavlje relacije r1 čine unija skupova atributa {X} i {Y}, pri čemu je {Y} skup atributa relacije r2. Tada je kolčnik relacija r1 i r2 u oznaci r1 DIVIDEBY r2 skraćenica za:

r1 {X} NOT MATCHING ((r1 {X} JOIN r2) NOT MATCHING r1)

Primer: (ISPIT{INDEKS, ID\_PREDMETA}

DIVIDEBY PREDMET{ID\_PREDMETA} JOIN DOSIJE

* Agregatni operatori – Agregatni operator u relacionom modelu je svaki operator koji izvodi jednu vrednost iz „agregata“ vrednosti koji se pojavljuju unutar nekog atributa neke relacije. Agregatni operatori nisu relacioni već skalarni, jer vraćaju pojedinačnu vrednost. Sintaksa: naziv(relacioni\_izraz, [, izraz]). Uobičajeni: COUNT, SUM, AVG, MAX, MIN, AND, OR i XOR
* Rekurzivno zatvorenje – Za datu relacionu vrednost pp relacije PP relacija rz koja predstavlja rekurzivno zatvorenje od pp može da se definiše na sl. način:

par (px, py) se javlja u rz akko:

se javlja u pp, ili

postoji neko pz tako da se par (pxz, pz) javlja u pp i par (pz, py) se javlja u rz

**Pitanja**

1. Navesti prednosti relacionog modela u odnosu na hijerarhijski.

2. Navesti i opisati osnovne komponente relacionog modela.

3. Opišite aspekte relacionog modela podataka.

4. Koja je funkcija okidača, a koja kataloga u relacionom modelu?

5. Opisati svojstva transakcija u relacionim bazama.

6. Koja je funkcija korisnički definisanih tipova?

7. Koja je svrha upotrebe korisnicki definisanih tipova? Da li isti efekat može da se postigne i bez njihove upotrebe? Objasniti odgovor.

8. Navesti formalne definicije bar 4 dodatna operatora relacione algebre.

9. Dati definiciju i opisati operator sumarizacije.

10. Nabrojite dodatne (ne Kodove) operatore relacione algebre.

11. Izraziti relacioni operator ekskluzivne unije pomoću Kodovih operatora kao i operator deljenja pomoću operatora poluspajanja.

12. Dokazati da za operatore relacione algebre važi da projekcija nije distributivna u odnosu na razliku.

13. Prikazati da je SQL relaciono kompletan jezik ako je zasnovan na skupu operatora relacione algebre kojipored Kodovih operatora uključuje i bar 4 proizvoljna operatora iz skupa dodatnih operatora.

14. Pokazati da pomoću operatora iz minimalnog skupa Kodovih operatora mogu da se izvedu operator deljenja i operator polurazlike.

15. Formalno dokazati da za operatore relacione algebre važi da je restrikcija distributivna u odnosu na uniju.

16. Formalno dokazati da za operatore relacione algebre važi da je prirodno spajanje distributivno u odnosu na uniju.

17. Kada se za neki jezik kaže da je relaciono kompletan? Navedite naredbe SQL jezika koje pokazuju da je on relacionokompletan, kao i odgovarajuće naredbe relacione algebre.

18. Dati formalne definicije levog, desnog i potpunog spoljašnjeg spajanja.