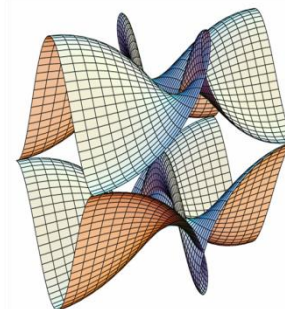




Sveučilište u Zagrebu
PMF – Matematički odsjek

BAZE PODATAKA
Predavanja 2019/2020



Poglavlje 5: Postavljanje upita u relacijskim bazama podataka

Sastavio: Robert Manger
06.04.2020

Općenito o postavljanju upita (1)

- U prethodnim poglavljima upoznali smo se s relacijskim modelom za bazu podataka.
- U ovom poglavlju zanima nas dinamički aspekt relacijskog modela, koji se očituje u postojanju fleksibilnih jezika za postavljanje upita.
- Postavljanje upita može se interpretirati kao primjena matematičkih operacija kojima se iz postojećih relacija dinamički stvaraju nove virtualne relacije.
- Jezici za postavljanje upita proučavaju se jednako dugo kao i same relacijske baze.

Općenito o postavljanju upita (2)

- Edgar Codd takve jezike smatra sastavnim dijelom relacijskog modela.
 - Relacijski model ne bi vrijedio mnogo bez svoje dinamičke komponente.
 - Baza od puno malih relacija ne bi bila upotrebljiva kad ne bi postojao fleksibilan i brz način pronalaženja i pregrupiranja podataka.
- Ovo poglavlje podijeljeno je u tri potpoglavlja.
 - Prva dva bave se relacijskom algebrom i relacijskim računom – upitnim jezicima koji su važni za teoriju.
 - Treće poglavlje predstavlja SQL - danas najrašireniji praktični jezik za rad s relacijskim bazama.
- Za zadavanje primjera upita služit će nam baza podataka o fakultetu sa sljedećim stanjem.

Općenito o postavljanju upita (3)

STUDENT

JMBAG	PREZIME	IME	GODINA_STUDIJA
0036398757	Marković	Marko	1
1191203289	Petrović	Petar	2
1192130031	Horvat	Dragica	2
1191205897	Janković	Marija	1
0165043021	Kolar	Ivan	3
0036448430	Grubišić	Katica	3
0246022858	Vuković	Janko	1

ZAVOD

IME_ZAVODA	OIB_PROCELNIKA	OPIS_DJELATNOSTI
Zavod za matematiku	25810043761	Ovaj zavod bavi se svim područjima teorijske i primijenjene matematike ...
Zavod za računarstvo	33571209458	Ovaj zavod bavi se računarskom znanosti i softverskim inženjerstvom ...

NASTAVNIK

OIB	PREZIME	IME	IME_ZAVODA	BROJ_SOB_E	PLACA
13257600947	Cantor	Georg	Zavod za matematiku	102	12000
25810043761	Goedel	Kurt	Zavod za matematiku	305	14000
33571209458	Codd	Edgar	Zavod za računarstvo	127	16000
44102179316	Klein	Felix	Zavod za matematiku	252	12000
50076128203	Pascal	Blaise	Zavod za matematiku	101	11000
67741205512	Turing	Alan	Zavod za računarstvo	315	13000

Općenito o postavljanju upita (4)

PREDMET

SIFRA_PREDMETA	NASLOV	IME_ZAVODA	OIB_NASTAVNIKA	SEMESTAR	ECTS_BODOVI
56001	Baze podataka	Zavod za računarstvo	33571209458	L	5
72001	Linearna algebra	Zavod za matematiku	44102179316	Z	6
72005	Matematička analiza	Zavod za matematiku	25810043761	L	6
56002	Programiranje u C-u	Zavod za računarstvo	67741205512	Z	5
72009	Analitička geometrija	Zavod za matematiku	44102179316	L	5

UPISAO

JMBAG	SIFRA_PREDMETA	DATUM_UPISA	OCJENA
0036398757	72001	2010-09-05	
1191203289	56001	2009-10-03	2
1191203289	56002	2009-10-03	2
1192130031	72001	2009-09-15	5
1192130031	56002	2009-09-15	4
1192130031	72009	2009-09-15	2
1191205897	72009	2010-09-05	
0165043021	56001	2008-10-03	3
0165043021	72001	2008-10-03	4
0165043021	56002	2009-06-10	
0036448430	56001	2009-09-04	5
0246022858	56001	2010-09-03	
0246022858	72001	2010-09-03	
0246022858	72005	2010-09-20	
0246022858	56002	2010-09-25	
0246022858	72009	2010-09-25	

Sadržaj Poglavlja 5

5.1. Relacijska algebra

5.2. Relacijski račun

5.3. Jezik SQL

Općenito o relacijskoj algebri

- Riječ je o matematičkoj notaciji, a ne o praktičnom jeziku.
- Uveo Edgar Codd u radovima iz 70-tih godina.
- Svodi se na izvrednjavanje algebarskih izraza građenih od operacija, operanda i zagrada.
 - Operandi u operacijama su relacije, a rezultati opet relacije.
 - Izraz predstavlja upit u bazu, a njegova vrijednost predstavlja odgovor na upit.
 - Odgovor je nova (virtualna) relacija dobivena iz postojećih primjenom algebarskih operacija.
- U nastavku opisujemo redom sve operacije.

Skupovne operacije (1)

- Relacije su skupovi n -torki. Zato na njima možemo primjenjivati uobičajene skupovne operacije.
- Neka R i S označavaju relacije. Tada je:
 - $R \text{ union } S$... unija od R i S , dakle skup n -torki koje su u R ili u S (ili u obje relacije).
 - $R \text{ intersect } S$... presjek od R i S , dakle skup n -torki koje su u R i također u S .
 - $R \text{ minus } S$... razlika od R i S , dakle skup n -torki koje su u R no nisu u S .
- Da bi se ove operacije mogle primijeniti, relacije R i S moraju biti *kompatibilne*, to jest jednako građene.

Skupovne operacije (2)

- Promatramo relaciju NOVI_STUDENT građenu jednako kao STUDENT.
- To su studenti nekog drugog fakulteta, među njima ima onih koji su također upisali i naš fakultet.

NOVI_STUDENT

JMBAG	PREZIME	IME	GODINA_STUDIJA
1191205336	Drašković	Janko	2
1192130031	Horvat	Dragica	2
1191621335	Iveković	Ivka	1
0165043021	Kolar	Ivan	3
0248022869	Zalar	Mladen	1

- Primjenom skupovnih operacija na STUDENT i NOVI_STUDENT dobivamo sljedeće rezultate.

Skupovne operacije (3)

STUDENT union NOVI_STUDENT

JMBAG	PREZIME	IME	GODINA_STUDIJA
0036398757	Marković	Marko	1
1191203289	Petrović	Petar	2
1192130031	Horvat	Dragica	2
1191205897	Janković	Marija	1
0165043021	Kolar	Ivan	3
0036448430	Grubišić	Katica	3
0246022858	Vuković	Janko	1
1191205336	Dražković	Janko	2
1191621335	Iveković	Ivka	1
0248022869	Zalar	Mladen	1

STUDENT intersect NOVI_STUDENT

JMBAG	PREZIME	IME	GODINA_STUDIJA
1192130031	Horvat	Dragica	2
0165043021	Kolar	Ivan	3

STUDENT minus NOVI_STUDENT

JMBAG	PREZIME	IME	GODINA_STUDIJA
0036398757	Marković	Marko	1
1191203289	Petrović	Petar	2
1191205897	Janković	Marija	1
0036448430	Grubišić	Katica	3
0246022858	Vuković	Janko	1

Skupovne operacije (4)

- Skupovne operacije obično služe za kombiniranje jednostavnijih upita u složenije.
- Ako su R i S virtualne relacije koje sadrže odgovore na dva jednostavnija upita, tada:
 - R union S sadrži podatke koji zadovoljavaju kriterije barem jednog od tih jednostavnijih upita,
 - R intersect S podatke koji istovremeno zadovoljavaju oba upita,
 - R minus S podatke koji odgovaraju na prvi upit no ne odgovaraju na drugi.

Selekcija i projekcija (1)

- *Selekcija* je unarni operator koji izvlači uz relacije n -torke koje zadovoljavaju zadani Booleov uvjet.
- Selekcija na relaciji R u skladu s Booleovim uvjetom \mathcal{E} označavamo s *R where \mathcal{E}* .
- Uvjet \mathcal{E} je formula koja se sastoji od:
 - konstanti ili atributa,
 - operatora za uspoređivanje =, <, >, ≤, ≥, ≠,
 - logičkih operatora and, or, not.

Selekcija i projekcija (2)

- Upit 1: pronadi sve studente na prvoj godini.

ODGOVOR1 := STUDENT where GODINA_STUDIJA = 1.

- Upit2: pronadi sve predmete s naslovom Baze podataka.

ODGOVOR2 := PREDMET where NASLOV = 'Baze podataka'.

- Upit 3: pronadi one studente koji su iz predmeta sa šifrom 56001 dobili ocjenu veću od 2.

ODGOVOR3 := UPISAO where

((SIFRA_PREDMETA = 56001) and (OCJENA > 2)).

Selekcija i projekcija (3)

ODGOVOR1

JMBAG	PREZIME	IME	GODINA_STUDIJA
0036398757	Marković	Marko	1
1191205897	Janković	Marija	1
0246022858	Vuković	Janko	1

ODGOVOR2

SIFRA_PREDMETA	NASLOV	IME_ZAVODA	OIB_NASTAVNIKA	SEMESTAR	ECTS_BODOVI
56001	Baze podataka	Zavod za računarstvo	33571209458	L	5

ODGOVOR3

JMBAG	SIFRA_PREDMETA	DATUM_UPISA	OCJENA
0165043021	56001	2008-10-03	3
0036448430	56001	2009-09-04	5

Selekcija i projekcija (4)

- *Projekcija* je unarni operator koji iz relacije izvlači zadane attribute. U rezultirajućoj relaciji eliminiraju se n -torke duplikati.
- Projekciju relacije R na njezine attribute A_1, A_2, \dots, A_m označamo s $R[A_1, A_2, \dots, A_m]$.
- U složenim algebarskim izrazima smatramo da projekcija ima viši prioritet od ostalih operacija, osim ako nije drukčije označeno zagradama.

Selekcija i projekcija (5)

- Upit4: pronadi brojeve soba svih nastavnika.

ODGOVOR4 := NASTAVNIK [BROJ_SOBE].

- Upit5: pronadi OIB nastavnika koji predaje predmet sa šifrom 72009.

ODGOVOR5 :=

(PREDMET where SIFRA_PREDMETA = 72009) [OIB_NASTAVNIKA].

ODGOVOR4

BROJ_SOBE
102
305
127
252
101
315

ODGOVOR5

OIB_NASTAVNIKA
44102179316

Kartezijev produkt i dijeljenje (1)

- Neka su R i S relacije stupnja n_1 odnosno n_2 . Tada algebarski izraz *R times S* daje *Kartezijev produkt* od R i S , dakle skup svih (n_1+n_2) -torki čijih prvih n_1 komponenti čine n_1 -torku u R , a zadnjih n_2 komponenti čine n_2 -torku u S .
- Atribut u R times S ima isto ime kao odgovarajući atribut u R odnosno S . Po potrebi se to ime proširuje imenom polazne relacije i točkom.
- Kartezijev produkt je jedna od operacija koje nam omogućuju odgovaranje na složenije upite gdje je potrebno povezivanje podataka iz raznih relacija.

Kartezijev produkt i dijeljenje (2)

- Upit 6: za svakog studenta ispisujemo sve predmete koje on nije upisao:

```
SVE_KOMBINACIJE :=  
    STUDENT[JMBAG] times PREDMET[SIFRA_PREDMETA],  
ODGOVOR6 :=  
    SVE_KOMBINACIJE minus UPISAO[JMBAG, SIFRA_PREDMETA].
```

- Upit 7: ispisati sve parove prezimena studenata koji su na istoj godini studija.

```
STUDENT1 aliases STUDENT;  
ODGOVOR7 := ( ( STUDENT1 times STUDENT )  
    where ((STUDENT1.GODINA_STUDIJA = STUDENT.GODINA_STUDIJA)  
    and (STUDENT1.JMBAG < STUDENT.JMBAG)) )  
[STUDENT1.PREZIME, STUDENT.PREZIME].
```

Kartezijev produkt i dijeljenje (3)

ODGOVOR6

JMBAG	SIFRA_PREDMETA
0036398757	56001
0036398757	72005
0036398757	56002
0036398757	72009
1191203289	72001
1191203289	72005
1191203289	72009
1192130031	56001
1192130031	72005
1191205897	56001
1191205897	72001
1191205897	72005
1191205897	56002
0165043021	72005
0165043021	72009
0036448430	72001
0036448430	72005
0036448430	56002
0036448430	72009

ODGOVOR7

STUDENT1.PREZIME	STUDENT.PREZIME
Marković	Janković
Marković	Vuković
Vuković	Janković
Petrović	Horvat
Grubišić	Kolar

Kartezijev produkt i dijeljenje (4)

- Neka je R relacija stupnja n , a S relacija stupnja m , i neka se svi atributi od S pojavljuju i u R .

Rezultat *dijeljenja* R sa S , oznaka R divideby S , je skup svih $(n-m)$ -torki $\langle x \rangle$ takvih da se n -torke $\langle x, y \rangle$ pojavljuju u R za sve m -torke $\langle y \rangle$ u S .

- Ovdje x i y mogu predstavljati i skupine od jedne ili više vrijednosti atributa.
- Slijedi apstraktni primjer:

Kartezijev produkt i dijeljenje (5)

R1

<i>A</i>	<i>B</i>
<i>a</i> ₁	<i>b</i> ₁
<i>a</i> ₁	<i>b</i> ₂
<i>a</i> ₁	<i>b</i> ₃
<i>a</i> ₂	<i>b</i> ₁
<i>a</i> ₂	<i>b</i> ₂
<i>a</i> ₃	<i>b</i> ₁
<i>a</i> ₄	<i>b</i> ₄

R2

<i>B</i>
<i>b</i> ₁

R3

<i>B</i>
<i>b</i> ₁
<i>b</i> ₂
<i>b</i> ₃

R1 divideby *R2*

<i>A</i>
<i>a</i> ₁
<i>a</i> ₂
<i>a</i> ₃

R1 divideby *R3*

<i>A</i>
<i>a</i> ₁

Kartezijev produkt i dijeljenje (6)

- Upit 8: pronađi JMBAG-ove studenata koji su upisali sve predmete.

ODGOVOR8 := UPISAO[JMBAG, SIFRA_PREDMETA] divideby
PREDMET[SIFRA_PREDMETA].

- Upit 9: pronađi JMBAG-ove onih studenata koji su upisali barem one predmete koje je upisao student s JMBAG-om 1191203289.

ODGOVOR9 := (UPISAO[JMBAG, SIFRA_PREDMETA] divideby
(UPISAO where JMBAG = 1191203289)[SIFRA_PREDMETA]).

ODGOVOR8

JMBAG
0246022858

ODGOVOR9

JMBAG
1191203289
0165043021
0246022858

Prirodni spoj i slične operacije (1)

- *Prirodni spoj (natural join)* je binarna operacija primjenjiva na dvije relacije R i S koje imaju bar jedan zajednički atribut.
- R join S sastoji se od svih n -torki dobivenih spajanjem jedne n -torke iz R s jednom n -torkom iz S koja ima iste vrijednosti zajedničkih atributa.
- U rezultirajućoj relaciji zajednički atribut pojavljuje se samo jednom.
- Definicija se bolje može razumjeti na osnovu apstraktnog primjera sa sljedeće slike.

Prirodni spoj i slične operacije (2)

R

<i>A</i>	<i>B</i>	<i>C</i>
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₃	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₄	<i>b</i> ₂	<i>c</i> ₃

S

<i>B</i>	<i>C</i>	<i>D</i>
<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁
<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₂
<i>b</i> ₂	<i>c</i> ₃	<i>d</i> ₃

R join *S*

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₂
<i>a</i> ₂	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁
<i>a</i> ₂	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₂
<i>a</i> ₄	<i>b</i> ₂	<i>c</i> ₃	<i>d</i> ₃

- Prirodni spoj je operacija koja na najprirodniji način uspostavlja veze između podataka u raznim relacijama.

Prirodni spoj i slične operacije (3)

- Upit 10: pronadi prezimena i imena svih studenata koji su upisali predmet sa šifrom 56001.

ODGOVOR10 :=

```
( (UPISAO where SIFRA_PREDMETA = 56001) join STUDENT)
[PREZIME, IME].
```

ODGOVOR10

PREZIME	IME
Petrović	Petar
Kolar	Ivan
Grubišić	Katica
Vuković	Janko

Prirodni spoj i slične operacije (4)

- Upit 11: Pronađi OIB-e nastavnika koji predaju predmete koje je upisao bar jedan student na drugoj godini studija.

ODGOVOR11 :=

```
(( (STUDENT where GODINA_STUDIJA = 2) join UPISAO) join PREDMET)
[OIB_NASTAVNIKA] .
```

JMBAG	PREZIME	IME	GODINA_STUDIJA	SIFRA_PREDMETA	DATUM_UPISA	OCJENA
1191203289	Petrović	Petar	2	56001	2009-10-03	2
1191203289	Petrović	Petar	2	56002	2009-10-03	2
1192130031	Horvat	Dragica	2	72001	2009-09-15	5
1192130031	Horvat	Dragica	2	56002	2009-09-15	4
1192130031	Horvat	Dragica	2	72009	2009-09-15	2

JMBAG	SIFRA_PREDMETA	OIB_NASTAVNIKA	...
1191203289	56001	33571209458	...
1191203289	56002	67741205512	...
1192130031	72001	44102179316	...
1192130031	56002	67741205512	...
1192130031	72009	44102179316	...

Prirodni spoj i slične operacije (5)

- Neka su R i S relacije, prva ima atribut A , a druga atribut B .
- Definiramo *theta-spoj* kao poopćenje prirodnog spoja.
- $R \text{ join}_{A \theta B} S$ je skup n -torki Kartezijeva produkta R sa S za koje je predikat $R.A \theta S.B$ istina.
- Simbol θ predstavlja jedan od operatora za usporedbu: $=, <, >, \neq, \leq, \geq$.

Prirodni spoj i slične operacije (6)

- Upit 12: pronadi broj sobe nastavnika koji predaje predmet sa šifrom 72005.

```
ODGOVOR12 :=  
( (PREDMET where SIFRA_PREDMETA = 72005)  
join OIB_NASTAVNIKA = OIB NASTAVNIK) [BROJ_SOBE].
```

ODGOVOR12

BROJ_SOBE
305

Prirodni spoj i slične operacije (7)

- *Vanjski spoj (outer join)* je operacija slična prirodnom spoju. Primjenjiva je pod istim uvjetima i daje kao rezultat relaciju s istom shemom.
- Pored svih n -torki iz $R \text{ join } S$, $R \text{ outerjoin } S$ sadrži i sve „nesparene“ n -torke iz R odnosno S .
- Te nesporene n -torke na odgovarajući način su proširene nedostajućim (null) vrijednostima.
- Za isti apstraktni primjer R i S koji smo imali za prirodni spoj, $R \text{ outerjoin } S$ izgleda ovako:

Prirodni spoj i slične operacije (8)

R

<i>A</i>	<i>B</i>	<i>C</i>
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₂	<i>b</i> ₁	<i>c</i> ₁
<i>a</i> ₃	<i>b</i> ₂	<i>c</i> ₂
<i>a</i> ₄	<i>b</i> ₂	<i>c</i> ₃

S

<i>B</i>	<i>C</i>	<i>D</i>
<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁
<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₂
<i>b</i> ₂	<i>c</i> ₃	<i>d</i> ₃

R outerjoin *S*

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₂
<i>a</i> ₂	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁
<i>a</i> ₂	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₂
<i>a</i> ₄	<i>b</i> ₂	<i>c</i> ₃	<i>d</i> ₃
<i>a</i> ₃	<i>b</i> ₂	<i>c</i> ₂	-

- Vanjski spoj obično se rabi za traženje podataka koji ne zadovoljavaju neki uvjet.
- **Upit 13: pronadi naslove predmeta koje do 31.12. 2009. godine nije upisao ni jedan student.**

Prirodni spoj i slične operacije (9)

ODGOVOR13 :=

((PREDMET outerjoin (UPISAO where DATUM_UPISA < 2010-01-01))
where (JMBAG is null)) [NASLOV].

PREDMET outerjoin (UPISAO where DATUM_UPISA < 2010-01-01)

SIFRA_PREDMETA	NASLOV	...	JMBAG	DATUM_UPISA	...
56001	Baze podataka	...	1191203289	2009-10-03	...
	Baze podataka	...	0165043021	2008-10-03	
	Baze podataka	...	0036448430	2009-09-04	
72001	Linearna algebra	...	1192130031	2009-09-15	...
	Linearna algebra	...	0165043021	2008-10-03	
56002	Programiranje u C-u	...	1191203289	2009-10-03	...
	Programiranje u C-u	...	1192130031	2009-09-15	
	Programiranje u C-u	...	0165043021	2009-06-10	
72009	Analitička geometrija	...	1192130031	2009-09-15	...
72005	Matematička analiza	...	-	-	-

Sadržaj Poglavlja 5

5.1. Relacijska algebra

5.2. Relacijski račun

5.3. Jezik SQL

Općenito o relacijskom računu

- Uveo Edgar Codd u radovima iz 70-tih godina.
- Notacija zasnovana na predikatnom računu.
- Upit se izražava tako da zadamo predikat kojeg tražene n -torke moraju zadovoljavati.
- Postoje dvije varijante:
 - račun orijentiran na n -torke, osnovni objekti su n -torke,
 - račun orijentiran na domene, osnovni objekti su vrijednosti iz domena za attribute.
- Varijante obrađujemo u sljedeća dva odjeljka.
- Zadnji odjeljak objašnjava odnos relacijskog računa i relacijske algebre.

Račun orijentiran na n -torke (1)

- Izrazi koji izražavaju upit sastoje se od elemenata.
 - **Varijable n -torke**, poprimaju vrijednosti iz imenovane relacije. Ako je t varijabla koja „prolazi“ relacijom R , a A je atribut od R , tada $t.A$ označava vrijednost od A unutar t .
 - **Uvjeti oblika $x \theta y$** gdje je θ operator za uspoređivanje $=, <, >, \neq, \leq$ ili \geq . Bar jedno od x i y mora biti oblika $t.A$, a drugo može biti konstanta. Uvjeti također mogu biti oblika $R(t)$, što znači da je t n -torka u relaciji R .
 - **Dobro oblikovane formule (WFF)**. Građene su od logičkih veznika and, or, not, te egzistencijalnog kvantifikatora \exists i univerzalnog kvantifikatora \forall , u skladu sa sljedećim pravilima.

Račun orijentiran na n -torke (2)

- Svaki uvjet je WFF.
- Ako su f_1 i f_2 WFF, tada su to i f_1 and f_2 , f_1 or f_2 , i not f_1 .
- Ako je f jedna WFF u kojoj se t pojavljuje kao slobodna varijabla, tada su $\exists t (f)$ i $\forall t (f)$ također WFF.
- Pojava varijable u WFF je vezana ukoliko je ta varijabla uvedena kvantifikatorom, inače je slobodna.
- Ništa drugo nije WFF.
- Izraz računa orijentiranog na n -torke je oblika $\{ t.A, u.B, v.C, \dots \mid f \}$, gdje su t, u, v, \dots varijable n -orke, A, B, C, \dots su atributi odgovarajućih relacija, a f je WFF koja sadrži t, u, v, \dots kao slobodne varijable.

Račun orijentiran na n -torke (3)

- Upit 14: pronadi sve šifre predmeta.
 $\{ p.SIFRA_PREDMETA \mid PREDMET(p) \} .$
- Upit 15: pronadi JMBAG-ove svih studenata na prvoj godini.
 $\{ s.JMBAG \mid STUDENT(s) \text{ and } s.GODINA_STUDIJA = 1 \} .$
- Upit 16: Pronadi JMBAG-ove i prezimena studenata koji su upisali predmet sa šifrom 56002.
 $\{ s.JMBAG, s.PREZIME \mid STUDENT(s) \text{ and } \exists u (UPISAO(u) \text{ and } u.JMBAG = s.JMBAG \text{ and } u.SIFRA_PREDMETA = 56002) \} .$

Račun orijentiran na n -torke (4)

- Upit 17: pronadi šifre onih predmeta koje je upisao bar jedan student na drugoj godini.

$\{ p.SIFRA_PREDMETA \mid PREDMET(p) \text{ and } \exists u (UPISAO(u) \text{ and } u.SIFRA_PREDMETA = p.SIFRA_PREDMETA \text{ and } \exists s (STUDENT(s) \text{ and } s.JMBAG = u.JMBAG \text{ and } s.GODINA_STUDIJA = 2)) \} .$

- Upit 18: pronadi prezimena onih studenata koji su upisali sve predmete.

$\{ s.PREZIME \mid STUDENT(s) \text{ and } \forall p \exists u (PREDMET(p) \text{ and } UPISAO(u) \text{ and } p.SIFRA_PREDMETA = u.SIFRA_PREDMETA \text{ and } u.JMBAG = s.JMBAG) \} .$

Račun orijentiran na n -torke (5)

(Odgovor 14)

SIFRA_ PREDMETA
56001
72001
72005
56002
72009

(Odgovor 15)

JMBAG
0036398757
1191205897
0246022858

(Odgovor 16)

JMBAG	PREZIME
1191203289	Petrović
1191130031	Horvat
0165043021	Kolar
0246022858	Vuković

(Odgovor 17)

SIFRA_ PREDMETA
56001
56002
72001
72009

(Odgovor 18)

PREZIME
Vuković

Račun orijentiran na domene (1)

- Varijable „prolaze“ domenama a ne relacijama.
- Imamo **uvjete članstva** $R (A: v_1, B: v_2, C: v_3, \dots)$ gdje su A, B, C, \dots atributi od relacije R , a v_1, v_2, v_3, \dots su ili varijable ili konstante.
- Na primjer, uvjet
STUDENT (JMBAG: 1191203289, GODINA_STUDIJA: 2)
je istina ako i samo ako postoji n -torka u relaciji
STUDENT takva da je JMBAG = 1191203289 i
GODINA_STUDIJA = 2.
- Pravila za WFF su ista kao prije.

Račun orijentiran na domene (2)

- Upit 14: pronadi sve šifre predmeta.
{ S | PREDMET (SIFRA_PREDMETA: S) } .
- Upit 15: pronadi JMBAG-ove svih studenata na prvoj godini.
{ J | STUDENT (JMBAG: J, GODINA_STUDIJA: 1) } .
- Upit 16: Pronadi JMBAG-ove i prezimena studenata koji su upisali predmet sa šifrom 56002.
{ J, P | STUDENT (JMBAG: J, PREZIME: P) and
UPISAO (JMBAG: J, SIFRA_PREDMETA: 56002) } .

Račun orijentiran na domene (3)

Upit 17: pronadi šifre onih predmeta koje je upisao bar jedan student na drugoj godini.

$\{ S \mid \text{PREDMET (SIFRA_PREDMETA: } S) \text{ and}$
 $\exists J (\text{UPISAO (SIFRA_PREDMETA: } S, \text{ JMBAG: } J) \text{ and}$
 $\text{STUDENT (JMBAG: } J, \text{ GODINA_STUDIJA: } 2)) \} .$

- Upit 18: pronadi prezimena onih studenata koji su upisali sve predmete.

$\{ P \mid \exists J (\text{STUDENT (JMBAG: } J, \text{ PREZIME: } P) \text{ and}$
 $\forall S (\text{if PREDMET (SIFRA_PREDMETA: } S) \text{ then}$
 $\text{UPISAO (JMBAG: } J, \text{ SIFRA_PREDMETA: } S))) \} .$

Odnos računa prema algebri (1)

- Relacijski račun i relacijska algebra ekvivalentni su u smislu izražajnosti.
 - Svaki upit zapisan u relacijskoj algebri može se zamijeniti ekvivalentnim upitom u relacijskom računu.
 - Svaki upit zapisan pomoću relacijskog računa može se prevesti u ekvivalentan izraz u relacijskoj algebri.
 - Pritom je svejedno služimo li se računom orijentiranim na n -torke ili na domene.
- Postoje upiti koji se ne mogu zapisati ni u jednom od razmatranih jezika.

Odnos računa prema algebri (2)

- Kao ilustraciju, za nekoliko algebarskih operacija navodimo ekvivalente u relacijskom računu orijentiranom na n -torke:
- R_1 union $R_2 \dots \{ t \mid R_1(t) \text{ or } R_2(t) \}$,
- R_1 minus $R_2 \dots \{ t \mid R_1(t) \text{ and not } R_2(t) \}$,
- R_1 times $R_2 \dots \{ \langle t, r \rangle \mid R_1(t) \text{ and } R_2(r) \}$,
- R_1 where $\mathcal{B}(A) \dots \{ t \mid R_1(t) \text{ and } \mathcal{B}(t.A) \}$,
- $R_1[A] \dots \{ t.A \mid R_1(t) \}$.

Odnos računa prema algebri (3)

- Glavna razlika je u tome što je račun u mnogo većoj mjeri „neproceduralan“ od algebre.
 - Kod postavljanja upita u relacijskoj algebri, korisnik konstruira postupak odgovaranja na upit time što bira operacije i utvrđuje redoslijed njihovog izvršavanja.
 - Kod postavljanja upita u relacijskom računu, korisnik samo specificira podatke koje želi dobiti a ne zadaje način kako da se do tih podataka dođe.
- Svojstvo neproceduralnosti smatra se poželjnim za neposredne korisnike.

Odnos računa prema algebri (4)

- Praktični jezici za postavljanje upita više liče na relacijski račun nego na relacijsku algebru.
 - Jezik SQL uglavnom je zasnovan na računu orijentiranom na n -torke.
 - Jezik QBE (Query by Example) rabi račun orijentiran na domene.
- Bez obzira kako izgledao jezik za korisnike, DBMS prilikom interpretiranja tog jezika mora odrediti postupak pronalaženja traženih podataka.

Odnos računa prema algebri (5)

- Dakle DBMS mora postavljenu upit nekom vrstom *redukcijskog algoritma* prevesti u neki oblik relacijske algebre.
- Relacijski račun je matematički model za način kako bi **korisnici** trebali postavljati upite.
- Relacijska algebra je matematički model za način kako će **DBMS** dalje interpretirati i izvrđnjavati korisničke upite.

Sadržaj Poglavlja 5

5.1. Relacijska algebra

5.2. Relacijski račun

5.3. Jezik SQL

Općenito o jeziku SQL (1)

- Najrašireniji jezik za rad s relacijskom bazom. Kratica znači *Structured Query Language*.
- Nastao u 70-tim godinama 20. stoljeća.
 - Razvio ga IBM,
 - Proširio ga Oracle.
- Omogućuje
 - postavljanje upita,
 - stvaranje relacija,
 - unos, promjenu i mijenjanje podataka,
 - upravljanje transakcijama,
 - davanje i oduzimanje ovlaštenja korisnicima,
 - neke oblike računanja s podacima.

Općenito o jeziku SQL (2)

- Upitni dio SQL-a:
 - uglavnom je zasnovan na relacijskom računu,
 - matematička notacija zamijenjena ključnim riječima nalik na govorni engleski jezik.
 - lagano se realiziraju i sve operacije iz relacijske algebre.
- SQL je u izražajnom smislu **ekvivalentan relacijskoj algebri** odnosno **relacijskom računu**.
- DBMS odgovara na SQL upit tako da ga interno prevede u algebarski izraz i dalje ga izvrednjava.

Jednostavni upiti (1)

- Svi upiti u SQL-u postavljaju se fleksibilnom naredbom **SELECT**.
- **SELECT** nije isto što i operacija selekcije iz relacijske algebre, mada može obavljati i njenu zadaću.
- Rezultat izvođenja **SELECT** shvaća se kao bezimena i privremena relacija koja ispisuje na zaslonu ili prosljeđuje u aplikacijski program.

Jednostavni upiti (2)

- Upit 19: pronadi JMBAG-ove, prezimena i imena svih studenata na drugoj godini studija.

```
SELECT JMBAG, PREZIME, IME FROM STUDENT  
WHERE GODINA_STUDIJA = 2;
```

- Upit 20: ispiši OIB-e nastavnika koji predaju bar jedan predmet.

```
SELECT DISTINCT OIB_NASTAVNIKA FROM PREDMET;
```

- Upit 21: ispiši sve podatke o nastavnicima u obliku rang liste s obzirom na njihove plaće.

```
SELECT * FROM NASTAVNIK ORDER BY PLACA DESC;
```

Jednostavni upiti (3)

(Odgovor na Upit 19)

JMBAG	PREZIME	IME
1191203289	Petrović	Petar
1192130031	Horvat	Dragica

(Odgovor na Upit 20)

OIB_NASTAVNIKA
33571209458
44102179316
25810043761
67741205512

(Odgovor na Upit 21)

OIB	PREZIME	IME	IME_ZAVODA	BROJ_SOBE	PLACA
33571209458	Codd	Edgar	Zavod za računarstvo	127	16000
25810043761	Goedel	Kurt	Zavod za matematiku	305	14000
67741205512	Turing	Alan	Zavod za računarstvo	315	13000
13257600947	Cantor	Georg	Zavod za matematiku	102	12000
44102179316	Klein	Felix	Zavod za matematiku	252	12000
50076128203	Pascal	Blaise	Zavod za matematiku	101	11000

Jednostavni upiti (4)

- Upit 22: pronađi JMBAG-ove onih studenata koji su iz predmeta sa šifrom 56001 dobili ocjenu veću od 2.

```
SELECT JMBAG FROM UPISAO  
WHERE SIFRA_PREDMETA = 56001 AND OCJENA > 2;
```

ili

```
SELECT JMBAG FROM UPISAO  
WHERE SIFRA_PREDMETA = 56001 AND OCJENA IN (3, 4, 5);
```

Upit 23: ispiši imena studenata koja počinju na 'M'.

```
SELECT IME FROM STUDENT  
WHERE IME BETWEEN 'M' AND 'N';
```

ili

```
SELECT IME FROM STUDENT WHERE IME LIKE 'M%';
```

Jednostavni upiti (5)

- Upit 24: Ispišite prezimena svih nastavnika zajedno s udvostručenim plaćama, sortirano po prezimenima.

```
SELECT PREZIME, PLACA*2 FROM NASTAVNIK  
ORDER BY PREZIME;
```

(Odgovor na Upit 22)

JMBAG
0165043021
0036448430

(Odgovor na Upit 23)

IME
Marko
Marija

(Odgovor na Upit 24)

PREZIME	PLACA * 2
Cantor	24000
Codd	32000
Goedel	28000
Klein	24000
Pascal	22000
Turing	26000

Jednostavni upiti (6)

- U naredbi za postavljanje upita pojavljuju se riječi **SELECT**, **FROM**, **WHERE** i **ORDER BY**.
 - Između **SELECT** i **FROM** navode se imena atributa koje želimo ispisati,
 - iza **FROM** slijede imena relacija iz kojih se crpe podaci,
 - iza **WHERE** su uvjeti koje podaci moraju zadovoljiti.
 - **ORDER BY** omogućuje zadavanje načina kako će ispis biti sortiran,
 - **SELECT** i **FROM** su obavezne, ostale se koriste po potrebi.
- Naredba se može protezati kroz više redaka, no mora završiti s točka-zarezom.

Složeniji upiti (1)

- Upiti u SQL-u mogu poprimiti složenije oblike od onih koje smo vidjeli do sada.
 - Dozvoljeno je odjednom rabiti **više relacija**, te kombinirati podatke iz njih.
 - Jedna SELECT naredba može se **ugnijezditi** unutar druge, tako da rezultat prve naredbe služi kao dio uvjeta u drugoj.

Složeniji upiti (2)

- Upit 25: pronadi JMBAG-ove i prezimena studenata koji su upisali predmet sa šifrom 72009.

```
SELECT STUDENT.JMBAG, PREZIME  
FROM STUDENT, UPISAO  
WHERE STUDENT.JMBAG = UPISAO.JMBAG  
AND SIFRA_PREDMETA = 72009;
```

ili

```
SELECT JMBAG, PREZIME  
FROM STUDENT  
WHERE JMBAG IN  
  (SELECT JMBAG  
   FROM UPISAO  
   WHERE SIFRA_PREDMETA = 72009);
```

Složeniji upiti (3)

- Upit 26: pronadi JMBAG-ove i prezimena studenata koji su upisali barem jedan predmet koji predaje nastavnik s OIB-om 44102179316.

```
SELECT STUDENT.JMBAG, PREZIME  
FROM STUDENT, UPISAO, PREDMET  
WHERE STUDENT.JMBAG = UPISAO.JMBAG  
AND UPISAO.SIFRA_PREDMETA = PREDMET.SIFRA_PREDMETA  
AND OIB_NASTAVNIKA = 44102179316;
```

ili

```
SELECT JMBAG, PREZIME FROM STUDENT WHERE JMBAG IN  
(SELECT JMBAG FROM UPISAO WHERE SIFRA_PREDMETA IN  
(SELECT SIFRA_PREDMETA FROM PREDMET  
WHERE OIB_NASTAVNIKA = 44102179316));
```

Složeniji upiti (4)

- Upit 27: pronađi sve parove JMBAG-ova studenata koji su na istoj godini studija.

```
SELECT TEMP1.JMBAG, TEMP2.JMBAG  
FROM STUDENT TEMP1, STUDENT TEMP2  
WHERE TEMP1.JMBAG < TEMP2.JMBAG  
AND TEMP1.GODINA_STUDIJA = TEMP2.GODINA_STUDIJA;
```

- Da bi se ista relacija STUDENT bez dvoznačnosti mogla rabiti kao da je riječ o dvije relacije, morali smo uvesti nadimke (aliase) TEMP1 odnosno TEMP2 za njezinu prvu odnosno drugu pojavu.

Složeniji upiti (5)

- Upit 28: ispiši prezimena i plaće za sve nastavnike koji imaju veću plaću od nastavnika Cantora.

```
SELECT N2.PREZIME, N2.PLACA  
FROM NASTAVNIK N1, NASTAVNIK N2  
WHERE N1.PREZIME = 'Cantor'  
AND N2.PLACA > N1.PLACA;
```

ili

```
SELECT PREZIME, PLACA  
FROM NASTAVNIK  
WHERE PLACA >  
  (SELECT PLACA  
   FROM NASTAVNIK  
   WHERE PREZIME = 'Cantor');
```

Složeniji upiti (6)

(Odgovor na Upit 25)

JMBAG	PREZIME
1192130031	Horvat
1191205897	Janković
0246022858	Vuković

(Odgovor na Upit 26)

JMBAG	PREZIME
0036398757	Marković
1192130031	Horvat
1191205897	Janković
0165043021	Kolar
0246022858	Vuković

(Odgovor na Upit 27)

TEMP1.JMBAG	TEMP2.JMBAG
0036398757	1191205897
0036398757	0246022858
0246022858	1191205897
1191203289	1192130031
0036448430	0165043021

(Odgovor na Upit 28)

PREZIME	PLACA
Goedel	14000
Codd	16000
Turing	13000

Složeniji upiti (7)

- Upit 29: ispiši sve podatke o predmetima koje nije upisao student Kolar.

```
SELECT * FROM PREDMET
WHERE SIFRA_PREDMETA NOT IN
  (SELECT SIFRA_PREDMETA FROM UPISAO, STUDENT
   WHERE UPISAO.JMBAG = STUDENT.JMBAG
   AND PREZIME = 'Kolar');
```

- Upit 31: ispiši prezimena i imena nastavnika koji ništa ne predaju.

```
SELECT PREZIME, IME FROM NASTAVNIK
WHERE OIB NOT IN
  (SELECT OIB_NASTAVNIKA FROM PREDMET);
```

Složeniji upiti (8)

- Upit 30: pronadi JMBAG-ove, prezimena i imena onih studenata koji su upisali sve predmete.

```
SELECT JMBAG, PREZIME, IME FROM STUDENT
WHERE NOT EXISTS
  (SELECT * FROM PREDMET
   WHERE NOT EXISTS
     (SELECT * FROM UPISAO
      WHERE UPISAO.JMBAG = STUDENT.JMBAG
      AND UPISAO.SIFRA_PREDMETA =
        PREDMET.SIFRA_PREDMETA));
```

- Budući, da SQL nema univerzalni kvantifikator ali ima egzistencijalni, upit smo ovako morali preformulirati: pronadi one studente za koje ne postoji predmet kojeg oni nisu upisali.

Složeniji upiti (9)

(Odgovor na Upit 29)

SIFRA_PREDMETA	NASLOV	IME_ZAVODA	OIB_NASTAVNIKA	SEMESTAR	ECTS_BODOVI
72005	Matematička analiza	Zavod za matematiku	25810043761	L	6
72009	Analitička geometrija	Zavod za matematiku	44102179316	L	5

(Odgovor na Upit 30)

JMBAG	PREZIME	IME
0246022858	Vuković	Janko

(Odgovor na Upit 31)

PREZIME	IME
Cantor	Georg
Pascal	Blaise

Grupirajući upiti (1)

- Odgovori na dosadašnje upite sastojali od vrijednosti koje se doslovno pojavljuju u pojedinim n -torkama pojedinih relacija baze.
- No često su nam potrebne i vrijednosti koje odgovaraju cijelim **grupama** (skupovima) n -torki.
- Takve grupne vrijednosti nastaju primjenom neke od **grupnih funkcija**.
- Na primjer, postoji grupna funkcija za
 - broj n -torki u grupi,
 - zbroj vrijednosti nekog izraza za sve n -torke u grupi,
 - minimum vrijednosti, maksimum, prosjek, i slično.

Grupirajući upiti (2)

- Upit 32: ispiši najmanju i najveću plaću nastavnika, te zbroj plaća za sve nastavnike.

```
SELECT MIN(PLACA), MAX(PLACA), SUM (PLACA)  
FROM NASTAVNIK;
```

Grupa je samo jedna i sastoji se od svih n -torki iz NASTAVNIK. Upit proizvodi jedan redak ispisa.

- Upit 33: koliko ima studenata na kojoj godini?

```
SELECT GODINA_STUDIJA, COUNT(*) FROM STUDENT  
GROUP BY GODINA_STUDIJA;
```

Skup n -torki iz STUDENT dijeli se u grupe s obzirom na vrijednosti GODINE_STUDIJA. Upit proizvodi onoliko redaka ispisa koliko ima različitih godina studija.

Grupirajući upiti (3)

- Upit 34: sastavi izvještaj koliko ima nastavnika u pojedinom zavodu.

```
SELECT IME_ZAVODA, COUNT(*) BROJ_NASTAVNIKA  
FROM NASTAVNIK  
GROUP BY IME_ZAVODA;
```

- Upit 35: odredi prosječnu plaću nastavnika za svaki pojedini zavod.

```
SELECT IME_ZAVODA, AVG(PLACA) PROSJEKNA_PLACA  
FROM NASTAVNIK  
GROUP BY IME_ZAVODA;
```

Grupirajući upiti (4)

(Odgovor na Upit 32)

MIN(PLACA)	MAX(PLACA)	SUM(PLACA)
11000	16000	78000

(Odgovor na Upit 33)

GODINA_STUDIJA	COUNT(*)
1	3
2	2
3	2

(Odgovor na Upit 34)

IME_ZAVODA	BROJ_NASTAVNIKA
Zavod za matematiku	4
Zavod za računarstvo	2

(Odgovor na Upit 35)

IME_ZAVODA	PROSJEKNA_PLACA
Zavod za matematiku	12250
Zavod za računarstvo	14500

Grupirajući upiti (5)

- Upit 36: ispiši popis JMBAG-ova, prezimena i imena svih studenata, te za svakog studenta broj koliko je predmeta on upisao. Popis treba biti leksikografski sortiran po prezimenima studenata.

```
SELECT STUDENT.JMBAG, PREZIME, IME,  
       COUNT(*) UPISAO_PREDMETA  
FROM STUDENT, UPISAO  
WHERE STUDENT.JMBAG = UPISAO.JMBAG  
GROUP BY STUDENT.JMBAG  
ORDER BY PREZIME;
```

Uvjet iza klauzule *WHERE* primjenjuje se prije grupiranja.
Klauzula *ORDER BY* primjenjuje se nakon grupiranja.

Grupirajući upiti (6)

- Upit 37: ispiši JMBAG-ove studenata koji su položili manje od dva predmeta, zajedno s brojem predmeta koje su položili.

```
SELECT JMBAG, COUNT(OCJENA) POLOZIO_PREDMETA  
FROM UPISAO  
GROUP BY JMBAG  
HAVING COUNT(OCJENA) < 2;
```

Klauzula **HAVING** je analogna klauzuli **WHERE**, no ona djeluje nakon grupiranja i njome je moguće izdvojiti samo neke grupe te izbaciti neželjene grupe.

Grupirajući upiti (7)

- Upit 38: ispiši prezimena i imena studenata s druge ili treće godine koji su skupili barem 10 ECTS-bodova, zajedno s brojem skupljenih ECTS-bodova. Ispis treba biti silazno sortiran prema broju ECTS-bodova.

```
SELECT PREZIME, IME, SUM(ECTS_BODOVI) ZBROJ_ECTS
FROM UPISAO, STUDENT, PREDMET
WHERE UPISAO.JMBAG = STUDENT.JMBAG
AND UPISAO.SIFRA_PREDMETA =
    PREDMET.SIFRA_PREDMETA
AND GODINA_STUDIJA >= 2
GROUP BY UPISAO.JMBAG
HAVING SUM(ECTS_BODOVI) >= 10
ORDER BY 3 DESC;
```

Grupirajući upiti (8)

(Odgovor na Upit 36)

STUDENT.JMBAG	PREZIME	IME	UPISAO_PREDMETA
0036448430	Grubišić	Katica	1
1192130031	Horvat	Dragica	3
1191205897	Janković	Marija	1
0165043021	Kolar	Ivan	3
0036398757	Marković	Marko	1
1191203289	Petrović	Petar	2
0246022858	Vuković	Janko	5

(Odgovor na Upit 37)

JMBAG	POLOZIO_PREDMETA
0036398757	0
1191205897	0
0036448430	1
0246022858	0

(Odgovor na Upit 38)

PREZIME	IME	ZBROJ_ECTS
Horvat	Dragica	16
Kolar	Ivan	16
Petrović	Petar	10