Standardni upitni jezik SQL

Osnovna ili standardna verzija SQL programskog paketa je verzija previđena za manipulisanje (rukovanje) podacima u relacionim bazama i sastavni je deo u svim programskim paketima za obradu podataka.

SQL je jezik upita koji omogućava realizaciju proizvoljnih funkcija – upita nad ma. U svojoj sintaksi u svim verzijama (SQL-8 , SQL-89, SQL-92 (SQL2), SQL:1999, SQL:2003 itd.) ima za sve osnovne operacije spajanje, razliku, proizvod, restrikciju i projekciju ekvivalentne SQL naredbe. Unija, presek i deljenje se ne smatraju osnovnim jer se te operacije mogu izvesti kombinovanjem pomenutih. U relacionoj algebri definisane operacije omogućuju da se dobije željena (tabela) iz skupa datih (tabela) tako je rezultat i SQL (tabela).

SQL (Structured Query Language) je standardni upitni jezik relacionog modela podataka koji omogućava korisnicima da postavljaju upite i dobijaju odgovore bez saradnje sa programerom. SQL jezik se sastoji od komandi za rad sa tabelama, komandi za razne vrste manipulacije nad podacima i komandi za pretraživanje podataka. Svi relacioni DBMS softveri imaju podršku za SQL. Zahvaljujući jednostavnom rečniku, SQL je lak za učenje. Recimo, dovoljna je samo jedna SQL komanda za kreiranje tabele sa kompleksnom strukturom.

SQL je neproceduralni jezik, što znači da korisnik treba da navede *šta* treba da se uradi, ali ne i *kako* to treba uraditi. Za korišćenje SQL komandi, krajnji korisnici i programeri ne moraju da poznaju fizički format zapisa datoteke na medijima, niti kompleksne aktivnosti koje se izvode pozivanjem jedne SQL komande. SQL je vrlo upotrebljiv za manipulaciju nad podacima (dodavanje, izmena, brisanje i pretraživanje) i administraciju podataka (kreiranje tabela, indeksa i pogleda*).*

SQL je u stalnom razvoju. Na početku je bio prilično jednostavan, blizak korisniku i u velikoj meri deklarativan (neproceduralan).Danas se za SQL može reći da je kompleksan, proceduralno/deklarativan jezik. Zaključno sa SQL-92 standardom SQL naredbe su svrstavane u jednu od sledeće tri kategorije:

- naredbe za manipulisanje (rukovanje) podacima (data manipulation statements)
- naredbe za definisanje podataka (data definition statements)
- naredbe za kontrolne (upravljačke) funkcije (data control statements).

  - Naredbe za manipulisanje (rukovanje) podacima omogućuju ažuriranje i prikaz podataka baze:

- SELECT (prikaz sadržaja relacione baze podataka)
- UPDATE (izmena vrednosti kolona tabele)

- DELETE (izbacivanje redova tabele)
- INSERT (dodavanje redova postojećoj tabeli).

• Naredbe za definisanje podataka omogućuju definisanje objekata baze:

- CREATE TABLE (kreiranje tebele baze podataka)
- CREATE VIEW (kreiranje virtuelne tabele - "pogleda")
- CREATE INDEX (kreiranje indeksa nad kombinacijom kolona tabele)
- ALTER TABLE (izmena definicije tabele)
- DROP TABLE (izbacivanje tabele iz baze podataka).

• Naredbe za kontrolne (upravljačke) funkcije omogućuju oporavak, konkurentnost, sigurnost i integritet relacione baze podataka:

- GRANT (dodela prava korijenja sopstvene tabele drugim korisnicima)
- REVOKE (oduzimanje prava korijenja sopstvene tabele od drugih korisnika)
- COMMIT (prenos dejstava transakcije na bazu podataka)
- ROLLBACK (poništavanje dejstava transakcije).

Dva osnovna načina korišćenja SQL-a su:

- direktno (interaktivno) korišćenje SQL-a i
- povezivanje SQL-a sa klasičnim programskim jezicima ("ugrađeni" SQL).

SQL je programski paket koji se zasniva na relacionoj algebri i relacionom računu.

1.1 Osnove relacione algebre

Relacionu algebru uveo je E.F. Codd u svojim radovima iz 70-tih godina XX veka. Reč je o teorijskoj (matematičkoj) notaciji, a ne o praktičnom jeziku kojeg bi ljudi zaista neposredno koristili. Relaciona algebra se svodi na izračunavanje algebarskih izraza, građenih od relacija unarnih i binarnih operatora (čiji operandi su relacija a rezultat je opet relacija). Svaki algebarski izraz predstavlja jedan upit (pretraživanje).

Relaciona algebra je matematička disciplina koja se bavi manipulacijom podataka i na kojoj se bazira relacioni model baze podataka. Relaciona algebra definiše skup operacija pomoću kojih se, na proceduralan način, može dobiti željena tabela iz skupa datih.

1.1.1 Operacije za ažuriranje

Operacije koje su pogodne za ažuriranje su: Unija, Presek, Razlika, Proizvod. Navedene operacije izvode se nad bar dve relacije i primeniće se na primerima datih relacija R_1 i R_2 koje zadovoljavaju uslove kompatibilnosti.

R_1

BrInd	Student	Smer
152/01	Ana	Informatika
223/01	Nikola	Elektronika
425/02	Filip	Informatika

R_2

BrInd	Student	Smer
152/01	Ana	Informatika
240/01	Marko	Elektronika

Uslov kompatibilnosti relacija je da moraju imati isti broj atributa (isti stepen), a odgovarajući atributi su definisani nad istim domenom.

Kod izvođenja operacije proizvod nad relacijama može doći do greške ako relacije imaju atribute sa istim imenima, a nemaju isto značenje. U takvim slučajevima potrebno je primenovati atribute u jednoj relaciji.

U relacionoj algebri uvek se želi da se dobije uređen skup n-torki, a ne uređen skup parova kao kod Dekartovog odnosno Kartezijevog proizvoda kod koga se proizvod dva skupa rezultat definišekao skup uređenih parova u kojem prvi element pripada prvom skupu a drugi drugom skupu.

1.2 Naredbe za prikaz sadržaja baze podataka

Naredba SELECT služi za izdvajanje jednog ili više podataka iz jedne ili više tabela iz baze podataka. Rezultat ove naredbe je nova tabela sa atributima iz postojećih tabela relacije ili novokreiranim kada se izvršavaju agregatne funkcije nad kolonama tabela. Kada su potrebni podaci iz više tabela tada se moraju postvljati upiti nad obe relacije istovremeno i tada se koriste: operacije spajanje tabela (join) i podupiti (ugnježdeni) upiti. To omogućava da se podaci neponavljaju u svim tabelama – u čemu se i sastoji snaga relacionih baza podataka. Sintaksa osnovne naredbe SQL-a je:

```
SELECT [ lista kolona ]
FROM [ lista tabela ]
WHERE [ izraz za uslov selekcije ]
```

Rezultat upita je nova tabela čije kolone su definisane listom atributa iza reši SELECT. U lisiti relacija iza reči FROM pobrojane su tabele iz kojih se uzimaju podaci a kvalifikacioni izraz iza reči WHERE predstavlja logički izraz koji određuje osobine vrsta u rezultujućoj tabeli

Ostale sintakse SQL-a imaju oblik:

```
SELECT [ lista kolona ]
FROM [ lista tabela ]
WHERE [ kvalifikacioni izraz1 ]
GROUP BY [ lista kolona ]
HAVING [ kvalifikacioni izraz2 ]
ORDER BY[ lista kolona ].
```

Listom atributa zadaje se operacija PROJEKCIJE. Kvalifikacionim izrazom zadaju se uslovi SELEKCIJE i SPAJANJA, odnosno iskazi slični iskazima u relacionom računu. Odredbe SELECT i FROM su obavezne, dok odredbe WHERE i ostale nisu.

Naredbom se kreiraju upiti sledećih vrsta:

- Upiti nad jednom tabelom
- Upiti nad jednom tabelom za dobijanje novih vrednosti
- Upiti spajanje (join) nad dve ili više tabela istovremeno
- Upiti kao argument u drugom upitu,podupiti ili ugnježdeni upiti.

Primeri naredbi dati u daljem tekstu su urađeni nad tabelama Student i Predmet, datim sledećim relacionim šemama:

STUDENT (Broj Indeksa, Student, Smer, SmerID, Datum upisa, Ispitni poeni, Dodatni poeni, PredID)
PREDMET (PredID, Predmet, Profesor)

Student

BrInd	Student	Smer	SmerID	Datum upisa	Ispitni poeni	Dodatni poeni	PredID
113/12	Marko Marić	Menadžment	1	12.12.2012	75,5	20,5	7
114/10	Dragan Perić	Elektronika	2	31.3.2010	67	0	9
110/11	Nevena Simić	Menadžment	1	16.8.2011	85	16,5	8
211/10	Milica Tasić	Informatika	3	21.10.2010	77,5	20,5	4
220/09	Milan Ilić	Informatika	3	30.11.2010	72,5	23	5
315/11	Zoran Marić	Elektronika	2	10.10.2011	68,5	0	4
221/10	Sanja Lukić	Elektronika	2	9.10.2010	69	19,5	6
510/08	Mirko Spasić	Menadžment	1	11.11.2008	71,5	24	4
110/11	Tamara Dasić	Menadžment	1	31.10.2011	67,5	22,5	8
103/10	Željko Spasić	Informatika	3	9.10.2010	79	18,5	7

Predmet

PredID	Predmet	Profesor
1	Matematika	Marko Jović
2	Engleski jezik	Ana Perić
3	Informatika	Danka Dimić
4	Fizika	Lazar Ilić
5	Informacioni sistemi	Željka Marić
6	Programski jezici	Nikola Lazić
7	Operativni sistemi	Milica Arsić

PredID	Predmet	Profesor
8	Marketing	Janko Jović
9	Digitalna elektronika	Nenad Perić

1.2.1 Upiti nad jednom tabelom

Upiti nad jednom tabelom kojima se prikazuje prost, neizmenjen sadržaj tabele je najprostiji upit

```
SELECT kolona1 [,kolona2...]
FROM tabela
```

Primeri:

- Prikazati šifre, nazive i profesore svih predmeta iz relacije o predmetima.

```
SELECT PredID, Predmet, Profesor
FROM Predmet;
```

PredID	Predmet	Profesor
1	Matematika	Marko Jović
2	Engleski jezik	Ana Perić
3	Informatika	Danka Dimić
4	Fizika	Lazar Ilić
5	Informacioni sistemi	Željka Marić
6	Programski jezici	Nikola Lazić
7	Operativni sistemi	Milica Arsić
8	Marketing	Janko Jović
9	Digitalna elektronika	Nenad Perić

Kada se traže svi atributi neke tabele, umesto navođenja svakog atributa pojedinačno, moguće je koristiti znak "*" sa istim dejstvom.

```
SELECT *
FROM PREDMET;
```

PredID	Predmet	Profesor
1	Matematika	Marko Jović
2	Engleski jezik	Ana Perić
	...	
9	Digitalna elektronika	Nenad Perić

Pored prikazivanja svih kolona neke tabele (tj. atributa neke relacije) koristeći SELECT odredbu moguće je:

- izdvojiti specifične kolone tabele
- kontrolisati redosled pojavljivanja kolona
- sprečiti selekciju duplih n-torki

Primer:

- Prikaziti nazive i šifre svih predmeta.

```
SELECT Predmet,   PredID
FROM  Predmet;
```

Predmet	PredID
Matematika	1
Engleski jezik	2
Informatika	3
Fizika	4
Informacioni sistemi	5
Programski jezici	6
Operativni sistemi	7
Marketing	8
Digitalna elektronika	9

Primer:

- Prikazati sve smerove koje su studenti upisali.

```
SELECT Smer  FROM Student;
```

Smer
Menadžment
Elektronika
Menadžment
Informatika
Informatika
Elektronika
Elektronika
Menadžment
Menadžment
Informatika

Kako se u pojedinim kolonama ponavljaju podaci, to se opcijom DISTINCT prikazuju samo različiti podaci za izabranu kolonu. Dograđena sintaksa naredbe SELECT ima sledeći izgled:

```
SELECT [DISTINCT] kolona1 [,kolona2…]
FROM tabela
```

Primer:

- Prikazati samo različite smerove koristeći odredbu DISTINCT

```
SELECT DISTINCT Smer
FROM Student;
```

Smer
Menadžment
Elektronika
Informatika

- **Odredba WHERE**

Where odredba nije obavezna, a može se koristiti sa SELECT, UPDATE i DELETE naredbama. Korišćena u SELECT bloku ona omogućuje:

- Selekciju specifičnih redova tabele (n-torki relacije) koji zadovoljavaju neki uslov
- Selekciju redova koje zadovoljavaju višestruke uslove (AND)
- Selekciju redova koje zadovoljavaju bar jedan od više uslova (OR)
- Selekciju redova koje ne zadovoljavaju određene uslove (NOT, IS NOT)
- Selekciju redova koje zadovoljavaju složene uslove (istovremeno AND i OR)
- Selekciju redova unutar izvesnog raspona (BETWEEN)
- Selekciju redova koje zadovoljavaju vrednost u listi vrednosti (IN)
- Selekciju redova koje sadrže određenu kombinaciju karaktera (EXISTS) itd.

```
SELECT [DISTINCT] kolona [,kolona...]
FROM tabela
[WHERE uslov_selekcije]
```

Uslovi selekcije u WHERE odrebi sadrže:

- operatore poređenja (kao što su =, >, >=, <, <=),
  - jednako ("=") WHERE Smer = 'Menadžment'
  - različito ("<>") WHERE Smer <> 'Menadžment'
  - veće (">") WHERE Dodatni_poeni > 20
  - manje ("<") WHERE Dodatni_poeni < 20
  - veće ili jednako (">=") WHERE Datum_upisa >= '2011-09-08'
  - manje ili jednako ("<=") WHERE Datum_upisa <= '2011-09-08'
- operatore ranga (BETWEEN i NOT BETWEEN),

- liste (IN, NOT IN),
- uzorke (LIKE i NOT LIKE),
- nepoznate vrednosti (IS NULL i IS NOT NULL),
- višestruke uslove pretraživanja (AND,OR).

Primeri:

- Prikazati sve podatke o studentima kojima je smer elektronika.

```
SELECT * FROM Student
WHERE Smer = 'Elektronika';
```

Broj Indeksa	Student	Smer	SmerID	Datum upisa	Ispitni poeni	Dodatni poeni	PredID
114/10	Dragan Perić	Elektronika	2	31.3.2010	67	0	9
315/11	Zoran Marić	Elektronika	2	10.10.2011	68,5	0	4
221/10	Sanja Lukić	Elektronika	2	9.10.2010	69	19,5	6

- Prikaži broj indeksa, ime i prezime i ispitne poene studenta kojim je šifra smera 1 i koji imaju ispitne poene veće od 70 (korišćenje logičkog operatera AND).

```
SELECT Broj_indeksa, Student , Ispitni_poeni
FROM Student
WHERE SmerID=1  AND   Ispitni_poeni>70;
```

Broj Indeksa	Student	Ispitni poeni
113/12	Marko Marić	75,5
110/11	Nevena Simić	85
510/08	Mirko Spasić	71,5

- Prikaži sve podatke o studentima sa smera elektronika i informatika (korišćenje logičkog operatera OR):

```
SELECT * FROM Student
WHERE Smer = 'Elektronika'
OR Smer = 'Informatika';
```

Broj Indeksa	Student	Smer	SmerID	Datum upisa	Ispitni poeni	Dodatni poeni	PredID
114/10	Dragan Perić	Elektronika	2	31.3.2010	67	0	9
211/10	Milica Tasić	Informatika	3	21.10.2010	77,5	20,5	4

Broj Indeksa	Student	Smer	SmerID	Datum upisa	Ispitni poeni	Dodatni poeni	PredID
220/09	Milan Ilić	Informatika	3	30.11.2010	72,5	23	5
315/11	Zoran Marić	Elektronika	2	10.10.2011	68,5	0	4
221/10	Sanja Lukić	Elektronika	2	9.10.2010	69	19,5	6
103/10	Željko Spasić	Informatika	3	9.10.2010	79	18,5	7

- Prikaži broj indeksa, studenta, šifru smera i datum upisa za automatičare čija šifra predmeta nije 4 (korišćenje logičkog operatera NOT):

```
SELECT Broj_indeksa, Student, SmerID, Datum_upisa
FROM UPIS
WHERE Smer = 'Informatika'
AND NOT (PredID <> 4);
```

Broj Indeksa	Student	SmerID	Datum upisa
220/09	Milan Ilić	3	30.11.2010
103/10	Željko Spasić	3	9.10.2010

- Prikaži studenta, broj poena, smer i šifru predmeta za informatičare i menadžment kojima je šifra predmeta 4 (korišćenje zagrada da bi se definisao redosled ispitivanja uslova kod istovremene primene AND i OR logičkih operatora):

```
SELECT Student, Ispitni_poeni, Smer, PredID
FROM Student
WHERE (Smer = 'Informatika' OR Smer = 'Menadžment' ) AND PredID = 4;
```

Student	Ispitni poeni	Smer	PredID
Milica Tasić	77,5	Informatika	4
Mirko Spasić	71,5	Menadžment	4

- Prikaži studenta, smer i broj poena koji imaju između 70 i 75 poena (korišćenje odredbe BETWEEN, koja proverava da li je tražena vrednost atributa u definisanom rasponu):

```
SELECT Student, Smer, Ispitni_poeni FROM STUDENT
        WHERE Ispitni_poeni BETWEEN 70 AND 75;
```

Student	Smer	Ispitni poeni
Milan Ilić	Informatika	72,5
Mirko Spasić	Menadžment	71,5

Isti uslov moguće je realizovati i na sledeći način:

```
WHERE Ispitni_poeni >=70 AND Ispitni_poeni <= 75;
```

- Prikaži studenta, smer i šifru predmeta studenata koji nisu informatičari niti menadžment (korišćenje operatora IN koji menja višestruku primenu operatora OR):

```
SELECT Student, Smer, PredID FROM Student
WHERE Smer NOT IN ('Informatika','Menadžment');
```

Student	Smer	PredID
Dragan Perić	Elektronika	9
Zoran Marić	Elektronika	4
Sanja Lukić	Elektronika	6

- **Odredba LIKE**

Odredba LIKE omogućuje pretraživanje na osnovu "uzorka" odnosno dobijanje informacija i kada ne znamo potpun naziv (tj. vrednost) određenog atributa tipa character. Ona koristi dva specijalna karaktera sa sledećim značenjem:

"*" predstavlja string od 0 ili više karaktera

"?" predstavlja poziciju tačno jednog karaktera.

Ostali karakteri imaju uobičajeno značenje.

Primeri:

... gde se ime završava sa N.

```
WHERE IME LIKE '*N'
```

... gde je treći karakter imena R.

```
WHERE IME LIKE ' ??R*'
```

... gde je ime dugačko 5 karaktera.

```
WHERE IME LIKE '?????'
```

... gde ime nije dugačko 5 karaktera.

```
WHERE IME NOT LIKE '?????'
```

... gde je u imenu slovo G posle R.

```
WHERE IME LIKE '*RG*'
```

Primeri:

- Prikaži ime i prezime studenta i datum upisa za studente kojima ime počinje slovom M (korišćenje odredbe LIKE):

```
SELECT Student, Datum_upisa
FROM UPIS
WHERE Student LIKE 'M*';
```

Student	Datum upisa
Marko Marić	12.12.2012
Milica Tasić	21.10.2010
Milan Ilić	30.11.2010
Mirko Spasić	11.11.2008

- Prikazati sve podatke o studentima koji u nazivu smera imaju karakter '_':

```
SELECT * FROM Student
WHERE Smer LIKE '*_*' ;
```

Broj Indeksa	Student	Smer	SmerID	Datum upisa	Ispitni poeni	Dodatni poeni	PredID
158/10	Marko Marić	Proiz_masin	2	31.3.2010	67	23,5	9
217/10	Nikola Ilić	Proiz_masin	3	21.10.2010	77,5	20,5	4
229/09	Saša Perić	Proiz_masin	3	30.11.2010	72,5	23	5

- **Korišćenje NULL vrednosti**

Dva osnovna tipa NULL vrednosti

- još nepoznata vrednost
- neprimenjivo svojstvo

Za testiranje null vrednosti koristi se sintaksa:

- IS NULL
- IS NOT NULL

Primeri:

- Prikazati ime i prezime, smer i dodatne poene studenata koji imaju 0 dodatnih poena:

```
SELECT Student, Smer, Dodatni_poeni FROM Student
WHERE Dodatni_poeni IS NULL;
```

Student	Smer	Dodatni poeni
Dragan Perić	Elektronika	
Zoran Marić	Elektronika	

- Prikazati ime i prezime, smer i dodatne poene studenata koji su osvojili dodatne poene:

```
SELECT Student, Smer, Dodatni_poeni
FROM Student
WHERE Dodatni_poeni IS NOT NULL;
```

Student	Smer	Dodatni poeni
Marko Marić	Menadžment	20,5
Nevena Simić	Menadžment	16,5

Student	Smer	Dodatni poeni
Milica Tasić	Informatika	20,5
Milan Ilić	Informatika	23
Sanja Lukić	Elektronika	19,5
Mirko Spasić	Menadžment	24
Tamara Dasić	Menadžment	22,5
Željko Spasić	Informatika	18,5

- **ORDER BY odredba**

Korišćenjem ORDER BY odredbe moguće je sortirati rezultujuću tabelu po jednom ili više atributa u rastućem ili opadajućem redosledu.

Za specifikaciju rastućeg redosleda koristi se odredba ASC, a za specifikaciju opadajućeg redosleda odredba DESC. Rastući redosled se podrazumeva, pa odredbu ASC nije neophodno navoditi, za razliku od odredbe DESC koju uvek treba navesti kada se sortira u opadajućem redosledu.

ORDER BY je uvek poslednja odredba u SELECT bloku.

Primer:

- Prikaži ime i prezime, smer i broj poena studenata kojima je šifra predmeta 4 uređene u rastućem redosledu smerova i opadajućem redosledu broja poena:

```
SELECT Student, Smer, Ispitni_poeni
FROM Student
WHERE PredID =4
ORDER BY Smer ASC, Ispitni_poeni DESC;
```

Student	Smer	Ispitni poeni
Milica Tasić	Informatika	77,5
Zoran Marić	Elektronika	71.5
Mirko Spasić	Menadžment	68,5

Kada se sortiranje vrši po koloni koja sadrži NULL vrednosti, redosled prikaza n-torki sa NULL vrednostima u koloni sortiranja uslovljen je realizacijom SQL-a u konkretnom SUBP-u.

1.2.2 Upiti nad jednom tabelom za dobijanje novih vrednosti

Korišćenjem izraza, agregatnih funkcija i funkcija nad pojedinačnim redovima moguće je obraditi podatke baze podataka i prikazati izvedene podatke. Funkcije i izrazi pišu se u SELECT listi i ne menjaju sadržaj tabela baze podataka. Pored prikazivanja prostih vrednosti memorisanih u tabelama baze podataka, SQL ima više funkcija koje se koriste za dobijanje izvedenih, sumarnih informacija koje se nazivaju agregatnim funkcijama.

Primena agregatne funkcije zahteva da redovi tabele na koju se agregatna funkcija primenjuje budu grupisani na neki način. Svaka agregatna funkcija generiše jedan rezultujući red za svaku grupu redova tabele. Ukoliko grupisanje redova tabele nije eksplicitno specificirano, čitava tabela tretira se kao jedna grupa.

Najznačajnije agregatne funkcije su:

- AVG (naziv_kolone) - izračunava srednju vrednost
- SUM (naziv_kolone) - izračunava ukupnu vrednost
- MIN (naziv_kolone) - nalazi minimalnu vrednost
- MAX (naziv_kolone) - nalazi maksimalnu vrednost
- COUNT (*) - nalazi broj redova u grupi
- COUNT ([ALL] naziv_kolone) - nalazi broj definisanih (not null) vrednosti kolone
- COUNT (DISTINCT naziv_kolone) - nalazi broj različitih definisanih (not null) vrednosti kolone

Primeri:

- Naći minimalni broj poena studenata menadžmenta:

```
SELECT MIN   (Ispitni_poeni)
FROM Student  WHERE Smer = 'Menadžment';
```

MIN (Ispitni_poeni)
67

Ako treba da se promeni naziv kolone u kojoj se prikazuje rezultat korisiti se AS.

- Naći maksimalni broj poena studenata menadžmenta:

```
SELECT MAX   (Ispitni_poeni) AS Maksimalno poena
FROM Student  WHERE Smer = 'Menadžment';
```

Maksimalno poena
85

- Naći ukupan broj poena i ukupan broj dodatnih poena za informatičare:

```
SELECT  SUM   (Ispitni_poeni),   SUM    (Dodatni_poeni)
FROM Student
WHERE Smer = 'Informatika';
```

SUM (Ispitni_poeni)	SUM (Dodatni_poeni)
229	62

Pored toga što argumenti agregatnih funkcija mogu biti izrazi, i same agregatne funkcije mogu biti operandi izraza. To je ilustrovano sledećim upitom:

- Koliki je srednji broj poena od ukupanog broj poena studenata sa informatike:

```
SELECT  AVG (Ispitni_poeni + Dodatni_poeni)
FROM Student
WHERE Smer = 'Informatika';
```

AVG(Ispitni_poeni + Dodatni_poeni)
97

- Koliko je studenata na smeru Menadžment.

```
SELECT COUNT (*)
FROM Student
WHERE Smer = 'Menadžment‘
```

COUNT(*)
6

- Naći minimalni, srednji i maksimalni broj poena, kao i broj studenata menadžmenta:

```
SELECT MIN   (Ispitni_poeni),   AVG   (Ispitni_poeni),
MAX (Ispitni_poeni),  COUNT (*) AS Broj
FROM Student  WHERE Smer = 'Menadžment';
```

MIN (Ispitni_poeni)	AVG (Ispitni_poeni)	MAX (Ispitni_poeni)	Broj
67	73,3	85	4