

Dr Snežana R. Popović
Mr Miloš Milosavljević

SQL PROGRAMIRANJE

– sa primerima u SQL server bazi podataka

SQL programiranje – sa primerima u SQL server bazi podataka

Dr Snežana R. Popović

Mr Miloš Milosavljević

ISBN 978-86-7991-432-3

Copyright © 2020. Računarski fakultet, Beograd i CET, Beograd.

Sva prava zadržana. Nijedan deo ove knjige ne može biti reprodukovan, snimljen, ili emitovan na bilo koji način: elektronski, mehanički, fotokopiranjem, ili drugim vidom, bez pisane dozvole izdavača. Informacije korišćene u ovoj knjizi nisu pod patentnom zaštitom. U pripremi ove knjige učinjeni su svi napor da se ne pojave greške. Izdavač i autori ne preuzimaju bilo kakvu odgovornost za eventualne greške i omaške, kao ni za njihove posledice.

Recenzent Bratislav Dimitrov

Lektura Sandra Zlatanović

Tehnički urednik Vesna Petrinović

Gl. i odg. urednik Jovana Ristić

Izdavači **Računarski fakultet**
Beograd, Knez Mihaila 6/VI
tel. (011) 2627-613, 2633-321
www.raf.edu.rs

CET
Computer Equipment and Trade
Beograd, Skadarska 45
tel/fax: (011) 3243-043, 3235-139, 3237-246
www.cet.rs

Za izdavača Dragan Stojanović, direktor

Tiraž 1000

Štampa SaTCIP, Vrnjačka Banja

Beograd, 2020.

Knjiga nas postupno i usmereno vodi ka sticanju osnovnih, a kasnije i naprednih znanja i koncepata baza podataka i SQL servera. Autori na koncizan način, sa odmerenim i jasnim primerima uspevaju da čitaoca uvedu u apstraktni svet ukazujući na vezu između baza podataka i realnih koncepata koje one implementiraju.

Dugogodišnje iskustvo na realnim projektima i implementacijama utkano je u tekst ove knjige i sve date primere, pružajući čitaocu uvid u realne probleme i njihova rešenja. Autori na ovaj način uspevaju da čitaoca na najbolji način pripreme na realne situacije sa kojima će se u praksi susretati.

Knjiga „SQL programiranje – sa primerima u SQL server bazi podataka” zbog jedinstvenog pristupa autora, uspeva ne samo da čitaoca nauči SQL programiranju, već i da prenese način razmišljanja i pristup potreban za uspešno SQL programiranje i projektovanje baza podataka. Upravo to čini ovu knjigu jedinstvenom i dragocenom za sve one koji žele ući u svet SQL programiranja.

*Bratislav Dimitrov
recenzent*

Sadržaj

Uvod.....	11
1. SQL upitni jezik	15
1.1 Elementi SQL jezika.....	16
1.2 SQL naredbe	20
2. Tipovi podataka	23
2.1 Numerički tipovi	23
2.2 Datumski i vremenski tipovi.....	27
2.3 Karakter tipovi.....	29
2.4 Ostali tipovi	35
2.5 Konverzija tipova podataka	36
3. Baze podataka.....	40
3.1 Vrste baza podataka	40
3.2 Kreiranje korisničkih baza podataka	42
3.3 Collation	44
3.4 Zadaci za vežbanje	45
4. Tabele	47
4.1 Kreiranje tabela.....	47
4.2 IDENTITY i SEQUENCE.....	52
4.3 Privremene tabele	54
4.4 Zadaci za vežbanje	55
5. SELECT naredba.....	57
5.1 Pisanje SELECT naredbe	57

5.2 Logički redosled izvršavanja SELECT naredbe	59
5.3 Eliminisanje duplih redova pomoću DISTINCT	61
5.4 Korišćenje pseudonima (alijasa) za kolone i tabele	62
5.5 Pisanje CASE izraza u SELECT naredbi	63
5.6 Zadaci za vežbanje	65
6. Pretraživanje više tabela	67
6.1 Terminologija i sintaksa	67
6.2 INNER JOIN	68
6.3 OUTER JOIN	70
6.4 CROSS JOIN	74
6.5 SELF JOIN	76
6.6 Zadaci za vežbanje	78
7. Sortiranje i filtriranje podataka	80
7.1 ORDER BY	80
7.2 WHERE	82
7.3 TOP	86
7.4 OFFSET-FETCH	87
7.5 Filtriranje NULL oznaka	90
7.6 Zadaci za vežbanje	92
8. Grupisanje podataka	94
8.1 Agregatne funkcije	94
8.2 GROUP BY i HAVING	100
8.3 Zadaci za vežbanje	104
9. Podupiti	106
9.1 Skalarni podupiti	107
9.2 Viševrednosni podupiti	109
9.3 Tabelarni podupiti	113
9.3.1 Izvedene tabele	113

9.3.2 CTE izrazi	116
9.4 EXISTS	121
9.5 Zadaci za vežbanje	122
10. SET operatori.	124
10.1 UNION	124
10.2 EXCEPT i INTERSECT	126
10.3 APPLY operator	128
10.4 Zadaci za vežbanje	130
11. Prozorske funkcije.	132
11.1 Rangirajuće prozorske funkcije	133
11.2 Agregatne prozorske funkcije	138
11.3 Vrednosne prozorske funkcije	139
11.4 Zadaci za vežbanje	142
12. Unos, promena i brisanje podataka	144
12.1 Unos podataka	144
12.2 Promena podataka	149
12.3 Brisanje podataka	152
12.4 Zadaci za vežbanje	156
13. Integritet podataka.	158
13.1 Domenski integritet	158
13.2 Integritet entiteta	162
13.3 Referencijalni integritet	164
13.4 Zadaci za vežbanje	169
14. Dizajn i implementacija View objekata	171
14.1 Kreiranje i korišćenje View objekta	172
14.2 Opcije prilikom kreiranja View objekta	176
14.3 Lanac vlasništva i View objekti	179
14.4 Informacije o postojećim View objektima	180

14.5 Indeksirani View objekti	182
14.6 Partitionisani View objekti	184
14.7 Izmene podataka preko View objekta	185
15. Obrada grešaka u SQL programskom jeziku	186
15.1 Microsoft SQL server i njegove poruke	187
15.2 Strukturirana obrada greške	190
15.3 Korisnički definisane poruke na SQL Serveru i njihovo korišćenje	194
16. Transakcije i konkurentni - višekorisnički rad u SQL serveru	201
16.1 Transakcije	201
16.2 Ugnježdene transakcije	210
16.3 Konkurentni - višekorisnički rad na SQL Serveru	212
17. Uskladištene procedure	217
18. Korisnički definisane funkcije	234
18.1 Skalarne korisnički definisane funkcije	234
18.2 Inline Table-Valued funkcije	237
18.3 Multistatement Table-Valued funkcije	240
19. Okidači	243
19.1 DML okidači	243
19.2 DDL okidači	251
19.3 Logon okidači	257
20. Memorijske optimizovane tabele i prirodno kompajlirane uskladištene procedure	263
20.1 Memorijske optimizovane tabele	264
20.2 Prirodno kompajlirane uskladištene procedure	266
21. Integracija SQL Servera i koda napisanog u .NET Framework platformi	271

22. SQL Server i XML podaci	287
22.1 XML struktura	287
22.2 Skladištenje XML šema i podataka na SQL Serveru	290
22.3 XML schema kolekcije	292
22.4 XQuery upiti	296
22.5 XML Indeksi	301
22.6 Korišćenje SELECT upita sa klauzulama FOR XML i FOR JSON	304
23. Skladištenje velikih binarnih podataka u SQL serveru	313
23.1 FILESTREAM	315
23.2 FILESTREAM implementacija	316
23.3 FileTable tabela	319
24. Full Text pretraga i Full Text indeksiranje	322
24.1 Koraci za kreiranje Full Text indeksa	323
24.2 Korišćenje Full Text indeksa	325
Preduslovi za vežbanje	329
Prilog 1 – Microsoft SQL server instalacija	329
Prilog 2 - Instaliranje SSMS klijenta	356
Prilog 3 - Kreiranje WWI_SQL baze podataka.	356
Prilog 4 - Opis WWI_SQL baze podataka.	356
Rešenja zadataka	363

Uvod

Raspoloživi i dostupni podaci, smešteni u bazama podataka, svakodnevno podržavaju brojne aktivnosti savremenog društva. Baze podataka kreiraju i koriste državni organi i ustanove, finansijske, proizvodne i zdravstvene organizacije, mediji i društvene mreže, itd. Nezavisno od vrste podataka, bazama se ne pristupa direktno, već posredstvom različitih aplikativnih rešenja, koja u svom programskom kodu često koriste SQL jezik.

U ovom priručniku obrađen je Transact-SQL jezik, Microsoft-ova verzija SQL jezika za komunikaciju sa relacionim bazama podataka, uz korišćenje Microsoft SQL Server, verzija 2017, sistema za upravljanje bazama podataka (SUBP) (engl. Relational Database Management System - RDBMS).

Priručnik omogućava sistematsko sticanje znanja: od početnog do naprednog nivoa. Za dostizanje potrebnog i naprednog nivoa znanja potrebno je redom i pažljivo čitati obrađena poglavlja i samostalno rešavati postavljene zadatke. Imajući u vidu činjenicu da je sticanje svakog znanja proces koji traje, korisnici ovog priručnika postići će najbolje rezultate ukoliko obrađeno gradivo čitaju u kontinuitetu, bez preskakanja delova priručnika i brzinom, koju će sebi prilagoditi.

Priručnik sadrži dva dela. U prvom delu, od 1. do 13. poglavlja, obrađuju se osnovno pretraživanje podataka pomoću Transact-SQL jezika, dok drugi deo, od 14. do 24. poglavlja, obrađuje programiranje baze podataka.

U početnom poglavlju, „SQL upitni jezik”, predstavljene su osnovni elementi SQL jezika, sa osnovnim grupama SQL naredbi.

Jedna od važnijih odluka, koja se donosi u fazi projektovanja baze je korišćenje odgovarajućih tipova podataka pri definisanju kolona tabela baze. Zato je tipovima podataka posvećeno posebno poglavlje. Drugo poglavlje opisuje sistemske tipove podataka koje SQL Server koristi, kao i načine za konverziju jednog tipa podataka u drugi.

U trećem poglavlju uveden je pojam baza podataka, opisane su sistemske i korisničke baze podataka, kao i različite opcije za kreiranje korisničkih baza podataka.

Četvrto poglavlje obrađuje naredbe za kreiranje tabela, koje su osnovni objekti u relacionim bazama podataka. Takođe, poglavlje opisuje privremene tabele i načine za generisanje vrednosti surogat ključeva.

Poglavlje broj 5 obrađuje osnovne slučajeve korišćenja `SELECT` naredbe, koji su objašnjeni korišćenjem primera kreiranja upita nad jednom tabelom. U okviru poglavlja opisano je: eliminisanje duplih redova iz rezultata upita, korišćenje alijasa u `SELECT` listi i `FROM` klauzuli, kao i pisanje `CASE` izraza. Takođe, objašnjen je logički redosled operacija pri evaluaciji `SELECT` naredbe, s obzirom na to da je razumevanje redosleda operacija `SELECT` naredbe preduslov za pisanje ispravnog i efektivnog upita.

Šesto poglavlje obrađuje situacije, koje se često sreću u realnom okruženju – pretraživanje podataka koji se nalaze u više tabela. Za pisanje takvih upita koriste se `JOIN` operatori, koji omogućavaju preuzimanje podataka iz dve ili više tabela, najčešće na osnovu postojećih, uspostavljenih relacija između uključenih tabela.

U poglavlju 7. obrađuje se `ORDER BY` klauzula, koja se koristi za sortiranje rezultujućih podataka, dobijenih izvršenjem `SELECT` upita. Takođe, u okviru ovog poglavlja prikazuju se različiti načini za ograničenje broja redova, koje vraća upit, koristeći `WHERE`, `TOP` i `OFFSET – FETCH` klauzule.

Poglavlje broj 8 opisuje najčešće korišćene agregatne funkcije, kao i naredbe za kreiranje grupa podataka, agregaciju i filtriranje kreiranih grupa.

Deveto poglavlje objašnjava podupite, odnosno načine da se u okviru već postojeće naredbe doda nova `SELECT` naredba. Opisano je kreiranje različitih tipova podupita: skalarnih, viševrednosnih i tabelarnih. Takođe, opisan je mehanizam čiji je zadatak da proveri da li će ili ne, podupit vratiti bilo kakav rezultat.

U poglavlju 10. uvedeno je nekoliko tipova `SET` operatora, koji se primenjuju nad skupovima rezultujućih podataka. Objasnjeno je korišćenje `UNION`, `INTERSECT`, `EXCEPT` i `APPLY` operatora.

Poglavlje 11. obrađuje prozorske funkcije, koje se izvršavaju nad skupom redova i vraćaju jednu agregatnu ili rangirajuću vrednost za svaki red tabele.

Poglavlje 12. opisuje osnove korišćenja `INSERT` naredbe za dodavanje redova u tabele, korišćenje `UPDATE` naredbe za promenu postojećih podataka, kao i korišćenje `DELETE` i `TRUNCATE` naredbi za uklanjanje postojećih redova iz tabela.

Kvalitet podataka može značajno doprineti uspehu jedne organizacije. Kritičan preduslov za dostizanje kvaliteta podataka je njihov integritet. Poglavlje 13.

obrađuje raspoložive opcije za obezbeđenje i očuvanje integriteta podataka, kao i nivoje uspostavljanja integriteta.

Poglavlje 14. govori o kreiranju i korišćenju `View` objekata koji se često primenjuju za potrebe izveštavanja gde treba povezati više relacionih tabela radi dobijanja potrebnih podataka. `View` predstavlja enkapsulaciju složenih `SELECT` upita, a takođe olakšava administraciju bezbednosti i prava korišćenja podataka.

Prilikom izvršavanja različitih programskih struktura uvek se mogu dogoditi greške iz različitih razloga. Poglavlje 15. objašnjava kako se ovakve greške mogu obraditi u programskom kodu i, po potrebi, izvršiti notifikacija korisnika o nastalom problemu.

Poglavlje 16. objašnjava principe i korišćenje sistemskih i korisničkih transakcija. Takođe je objašnjen način kojim se odvija istovremeni pristup i rad više korisnika baze podataka.

Poglavlje 17. je posvećeno uskladištenim procedurama. One predstavljaju izuzetno moćan alat za rad sa podacima baze podataka. Veoma su zastupljene i koriste se za potrebe pregleda, modifikacije i generalne obrade podataka. Takođe se koriste i za različite administratorske namene.

Poglavlje 18. nastavlja sa programskim elementima baze podataka i objašnjava rad sa korisnički definisanim funkcijama. Obrađena su tri tipa funkcija: skalarne, linijske sa jednom naredbom i linijske sa više naredbi koje vraćaju virtualnu tabelu.

U bazi podataka uvek postoji potreba da se pre bilo koje akcije nad podacima izvrše dodatne provere, obrade, transfer podataka i slične operacije. Poglavlje 19. se bavi okidačima koji su jedan od načina da se realizuju ovakve potrebe. Obrađeni su `instead of` i `after`, `DDL` i `login` okidači.

Poglavlje 20. se bavi memorijski optimizovanim tabelama koje predstavljaju `C` strukture koje se koriste za čuvanje podataka. Takođe su prikazane prirodno kompajlirane uskladištene procedure - opcija da se `SQL` skript mapira u `C` programski jezik i kao takav kompajlira u prirodni - mašinski kod procesora. Obe funkcionalnosti su nastale zbog potrebe za izuzetno brzom manipulacijom podacima.

Poglavlje 21. objašnjava način na koji je moguće uraditi integraciju `.NET` platforme i `SQL` servera korišćenjem jezika `C#`.

`XML` kao način opisa i razmene podataka je uz `JSON`, `CSV` i još neke formate odavno postao standard i nalazi se u širokoj upotrebi.

Poglavlje 22. predstavlja načine na koje možemo upravljati XML strukturama i podacima, smeštati ih u relacionu bazu podataka i obrnuto, relacione podatke predstavljati u XML formatu.

Poglavlje 23. se bavi potrebom za skladištenjem velikih binarnih podataka kao što su razni dokumenti, slike, mediji i bilo koji drugi podaci u relacionu bazu. Prikazana je FILESTREAM tehnologija i prateći FileTable objekat.

Poglavlje 24. objašnjava implementaciju i upotrebe specijalnih Full Text indeksa koji proširuju mogućnosti pretrage kroz tekstualne podatke u relacionim tabelama.

Kada se uči SQL jezik, nije dovoljno da se čitaju knjige ili slušaju lekcije. Neophodno je da se vežba, prateći zadatke iz knjiga. Korisno je i da postavljate i rešavate zadatke, koje sami kreirate.

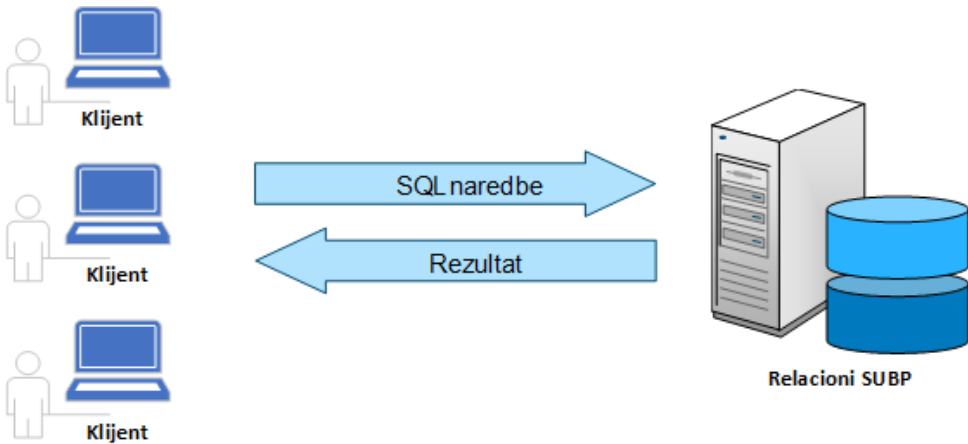
Samostalno učenje započecete instaliranjem SQL servera i kreiranjem baza za vežbanje. Na kraju priručnika je detaljno uputstvo za instalaciju SQL servera i klijentskog alata SQL server Managament Studio (SSMS) za komunikaciju korisnika sa SQL serverom, kao i opis baze podataka WWI_SQL, sa kojom se radi u ovom priručniku.

U okviru svakog poglavlja dati su primeri koji ilustruju određenu lekciju. Preporuka je da se primeri navedeni u tekstu izvrše u SSMS, koristeći funkcijski taster F5, ili dugme Execute. Takođe, preporuka je da se, pre prelaska na sledeće poglavlje, urade zadaci za vežbanje, koji se nalaze na kraju svakog poglavlja. Rešenja zadataka sa objašnjenjima nalaze se na kraju priručnika.

1. SQL upitni jezik

SQL je skraćenica za Structured Query Language. SQL upitni jezik koristi se za definisanje, organizaciju i korišćenje podataka u relacionim bazama podataka. Zahvaljujući zajedničkom radu Američkog nacionalnog instituta za standardizaciju (ANSI) i Međunarodne organizacije za standarde (ISO), SQL je postao standardni jezik za interakciju sa relacionim bazama podataka.

Naredbe SQL jezika koriste klijentske aplikacije da izvrše različite zadatke kod relacionih baza, kao što su: pregled, izmena i brisanje podataka, kreiranje i promena definicije objekata u bazi, davanje dozvola za pristup i korišćenje podataka. Kao rezultat izvršavanja SQL upita, relacioni SUBP vraća rezultat, najčešće u obliku skupa rezultujućih podataka.



Prva verzija standardizovanog SQL jezika nastala je 80-ih godina prošlog veka. Od tada, urađeno je više revizija. Različite verzije SQL jezika u svom nazivu sadrže godinu u kojoj su kreirane. Poslednja verzija, u momentu pisanja ovog priručnika, nosi naziv SQL:2016.

Korišćenje SQL jezika široko je rasprostranjeno u praksi i njegove specifikacije su potpuno ili delimično podržane od strane vodećih sistema za upravljanje relacionim bazama podataka. Tako Oracle implementira SQL jezik kao PL/SQL, Microsoft kao Transact-SQL, a IBM kao SQL PL.

Ovaj priručnik koristi Transact-SQL, ili skraćeno T-SQL, Microsoft implementaciju SQL standarda.

SQL jezik zasniva se na engleskom jeziku, što olakšava učenje i korišćenje. SQL jezik je deklarativni, a ne proceduralni jezik. Na primer, upit za preuzimanje podataka opisuje koji podaci su potrebni za prikaz, a ne opisuje se način, na koji se dolazi do podataka. Umesto pisanja proceduralne liste koraka, koje treba preduzeti, navode se atributi potrebnih podataka. Nakon toga, prepušta se sistemu za upravljanje bazama da odluči koji će fizički mehanizam koristiti za obradu upita napisanih SQL jezikom.

1.1 Elementi SQL jezika

SQL jezik sadrži više elemenata, uključujući:

- ključne reči,
- identifikatore,
- literale (konstante),
- operatore,
- izraze,
- funkcije,
- promenljive,
- komentare,
- NULL oznake,
- naredbe za upravljanje,
- tipove podataka.

Ključne reči su rezervisane reči, koje imaju predefinisano značenje u SQL jeziku. Koriste se za definisanje, manipulaciju i pristup bazi i objektima unutar baze podataka. Iako nije obavezno, uobičajeno je da se ključne reči pišu velikim slovima. Na primer, u naredbi `SELECT * FROM t1`; ključne reči su `SELECT` i `FROM`.

Identifikatori predstavljaju nazive za objekte, kao što su: baze podataka, tabele, kolone, pogledi, funkcije, procedure. Pravila za određivanje imena zavise od SUBP. U opštem slučaju:

- ime ne bi trebalo da bude ključna reč;
- prvi karakter u imenu trebalo bi da bude slovo;

- pored slova, ime može da sadrži brojeve i donju crtu (engl. *underscore*);
- dužina imena ne prelazi 128 Unicode karaktera;
- identifikator mora da bude jedinstven u okviru određenog domena;
- kada ime sadrži neki od nepreporučenih karaktera (ključna reč i/ili specijalni znak), smatra se da je ispravno ukoliko se navede između duplih navodnika " " ili uglastih zagrada [].

Kada se pristupa objektima unutar baze podataka, potpuni naziv navodi se u obliku: `Server.Database.Schema.Object`. S obzirom na to da je uglavnom unapred definisano sa kojim se serverom i bazom podataka radi, preporuka je da se pri pristupu objektima navodi naziv šeme i naziv objekta, u obliku `Schema.Object`.

Literali su specifične, fiksne vrednosti. Format literala zavisi od tipa podatka, čiju vrednost predstavlja. Postoje četiri tipa literala:

- numerički (na primer: -100, -120, 544.03, -458.25, 3E2, 5E-2, 0xAE);
- karakter stringovi, koji se pišu između jednostrukih navodnika (na primer: 'Beograd');
- bulovi (na primer: TRUE, FALSE);
- datumski i vremenski, koji se, kao i karakter stringovi, pišu između jednostrukih navodnika (na primer: 'Jan-25-2020 21:12:40.00', '20201025', '14:30:24').

Operatori omogućavaju manipulisanje podacima i vraćanje rezultata. Koriste se u različitim SQL naredbama (na primer: `SELECT`, `INSERT`, `UPDATE`, `DELETE`), ili kod kreiranja objekata u bazi podataka (na primer: funkcija, pogleda, okidača, procedura). SQL jezik podržava različite tipove operatora. Najčešće se koriste:

	Operator	Opis	Primer
Aritmetički operatori	+	Sabiranje	$5 + 2 = 7$
	-	Oduzimanje	$5 - 2 = 3$
	*	Množenje	$5 * 2 = 10$
	/	Deljenje	$5.0 / 2.0 = 2.5$
	%	Moduo (ostatak od celobrojnog deljenja)	$5 \% 2 = 1$

Operatori poređenja	=	Jednako	5 = 2 je netačno
	!=	Nije jednako	5 != 2 je tačno
	<>	Različito (nije jednako)	5 <> 2 je tačno
	>	Veće od	5 > 2 je tačno
	<	Manje od	5 < 2 je netačno
	>=	Veće i jednako	5 >= 2 je tačno
	<=	Manje i jednako	5 <= 2 je netačno
	!<	Nije manje	5 !< 2 je tačno
	!>	Nije veće	5 !> 2 je netačno
Logički operatori	AND	Poredi dva izraza i vraća tačno, ako su oba izraza tačna	
	OR	Poredi dva izraza i vraća tačno, ako je bar jedan izraz tačan	
	NOT	Menja vrednost izraza sa tačno na netačno i obrnuto	
Operator dodeljivanja	=	Omogućava dodeljivanje vrednosti, stavljene sa desne strane operatora promenljivoj ili parametru sa leve strane	SET @x = 5; Promenljiva @x imaće vrednost 5 posle evaluacije
Nadovezivanje stringova (engl. concatenation)	+	Kombinuje dva ili više karakter tipova u jedan izraz	ime + ' ' + prezime
Specijalni operatori	IN	Proverava da li vrednost postoji u listi vrednosti	
	BETWEEN	Proverava da li se vrednost nalazi u definisanom opsegu (uključuje donju i gornju granicu)	
	LIKE	Poredi vrednost sa drugom vrednošću, iskazanom preko paterna; primenjuje se samo na karakter tipove podataka	
	EXISTS	Proverava postojanje rezultata podupita	

Izrazi u SQL jeziku formiraju se od različitih elemenata, kao što su: literali, promenljive, imena kolona, operatori, podupiti, funkcije. Koriste se u upitima, a njihovom evaluacijom dobija se jedan rezultat. Na primer, u `SELECT` naredbi, izrazi se mogu koristiti: da obave određene manipulacije sa kolonama, da