

UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET

KATEDRA ZA RAČUNARSTVO

**I SEMINARSKI RAD**

Predmet: Sistemi za upravljanje bazama podataka

Tema: Evaluacija operatora koji se koriste prilikom obrade upita kod MySQL baze podataka

Mentor: Student:

Doc. dr Aleksandar Stanimirović Sandra Stojiljković 1537

Sadržaj

[1. Uvod u baze podataka 3](#_Toc133182039)

[2. MySQL baza podataka 4](#_Toc133182040)

[2.1 Upotreba MySQL-a 5](#_Toc133182041)

[2.2 MySQL alati 5](#_Toc133182042)

[2.3 SQL (Structured Query Language) 6](#_Toc133182043)

[2.3.1 SQL Sintaksa 6](#_Toc133182044)

[2.3.2 Tekstualne konstante 7](#_Toc133182045)

[2.3.3 Komentari 7](#_Toc133182046)

[2.3.4 SQL Naredbe 8](#_Toc133182047)

[3. Praktična primena upita u Mysql bazi podataka 9](#_Toc133182048)

[3.1 Kreirane tabele u MySQL bazi podataka 9](#_Toc133182049)

[3.2 Evaluacija operatora prilikom obrade upita 11](#_Toc133182050)

[3.2.1 Matematički operatori 12](#_Toc133182051)

[3.2.2 Relacioni operatori (operatori za poređenje) 16](#_Toc133182052)

[3.2.3 Predikati 20](#_Toc133182053)

[3.2.4 Logički operatori 22](#_Toc133182054)

[4. Zaključak 25](#_Toc133182055)

[5. Literatura 26](#_Toc133182056)

# 

# Uvod u baze podataka

Podatak predstavlja bilo koju sekvencu simbola koja dobija značenje specifičnim činom interpretacije. Digitalni podaci su oni nad kojima operacije izvodi računar i koji su smešteni na neku vrstu memorijskih jedinica (magneti, optički, mehanički medij) i koji se prenose u formi električnih signala. U kompjuterskim naukama pojam podatak odnosi se na sve ono što se nalazi u formi pogodnoj za korišćenja na računaru. Podaci mogu biti nestruktuirani (slike, muzika, sadrzaji direktorijuma) i struktuirani (tabele).

Baza podataka predstavlja kolekciju podataka organizovanih tako da se podacima lako može manipulisati. Ukoliko uopšteno posmatramo, baze podataka bi se mogle podeliti prema tipu podataka koje sadrže pa bi tako postojale tekstualne, numeričke ili baze nekih drugih tipova podataka.

Kompjuterske baze podataka najčešće se klasifikuju u zavisnosti od njihovog organizacionog pristupa. Organizacioni pristup se drugačije naziva model baze podataka. Može se reći da model baze podataka određuje njenu logičku strukturu, kao i način na koji su podaci skladišteni, organizovani i način na koji se njima manipuliše.

Postoje različiti modeli baze podataka kao sto su flet baza, relaciona baza, indekser i inverzni indeks, document i key value store.

Relacioni model je model po kome su podaci u bazi podataka organizovani u formi rednih lista (ordered list), a grupisani relacijama. Redne liste su u stvari tabele sa redovima i kolonama, a povezanosti među tabelama nazivaju se relacije.

Database menadžment sistemi predstavljaju sisteme za upravljanje bazama podataka koji je specijalno dizajnirana softverska aplikacija koja omogućava interakciju korisnika, drugih aplikacija sa jedne strane i same baze podataka sa druge strane. Drugim rečima RDBMS (relational database management system) omogućava definisanje, kreiranje, pretragu, ažuriranje i administraciju jedne baze podataka.

Najpoznatiji sistemi za upravljanje relacionim bazama podataka su MicrosoftSQLServer, Oracle, MySQL, PostgreSQL, SQLite, MariaDB, SAP HANA, dBASE, FoxPro, IBM DB2, LibreOffice Base, FileMaker Pro i Microsoft Access.

# MySQL baza podataka

MySQL je open source sistem za upravljanje bazama podataka i predstavlja najkorišćeniji sistem ove vrste. MySQL je projekat otvorenog koda, kreiran od strane švedske kompanije MySQL AB, u vlasništvu Oracle-a. Isporučuje se bez ikakvog alata sa grafičkim korisničkim okruženjem za manipulaciju podacima. Korisnici mogu da koriste integrisanu konzolu, odnosno komandni interfejs (CLI) ili da koriste neki od alata sa grafičkim korisničkim okruženjem koji dolaze odvojeno od samog MySQL sistema. MySQL je dostupan i za besplatno preuzimanje, međutim dostupno je i nekoliko plaćenih verzija koja nude dodatne funkcionalnosti. Kao što ime govori, MySQL je zasnovan na SQL standardnom jeziku za upis podataka unutar sistema za upravljanje relacionim bazama podataka. Deo "My" nazvan je po kćeri suosnivača Majkla Videnius-a, My. Kao što je gore navedeno, MySQL je relaciona baza podataka. To znači da su podaci smešteni unutar strukture sposobni da prepoznaju odnose između sačuvanih informacija. Svaka baza podataka sadrži tabele. Svaka tabela (koja se takođe naziva relacija) sadrži jednu ili više kategorija podataka pohranjenih u kolonama (koji se takođe nazivaju atributima). Svaki red (koji se takođe naziva zapis ili zbirka) sadrži jedinstven podatak (koji se inače naziva ključ) za kategorije definisane u kolonama. MySQL sadrži veliki broj ugrađenih operatora i funkcija. Većina njih se prvenstveno upotrebljava u SELECT i WHERE odredbama. U svim MySQL izrazima se mogu zadavati zagrade da bi se upravljalo redosledom kojim se izračunavaju pojedini delovi izraza, na isti način kao i u bilo kom drugom programskom jeziku.

## Upotreba MySQL-a

MySQL koriste neke od najvećih organizacija na svetu. Međutim, koristi se i za manje projekte, poput ličnih web sajtova ili blogova. MySQL je podrazumevana baza podataka za WordPress platformu - sistem za upravljanje sadržajem (CMS), u kome je napravljeno više od 60 miliona web sajtova. Ostali sistemi za upravljanje sadržajem koji koriste MySQL uključuju TYPO3, MODx, Joomla, phpBB, MyBB, i Drupal. Najčešća upotreba MySQL-a je u svrsi za web baze podataka. Može se koristiti za smeštanje bilo čega, od jednog zapisa podataka do celog popisa dostupnih proizvoda za Internet prodavnicu. U kombinaciji sa skriptnim jezikom kao što su npr. PHP ili Perl, moguće je kreirati web stranice koje će u realnom vremenu komunicirati sa MySQL bazom podataka kako bi se korisniku web sajta brzo prikazale kategorisane i informacije koje se mogu pretraživati. MySQL je trenutno u vlasništvu kompanije Oracle. Prvobitno je bio u vlasništvu švedske kompanije MySQL AB, koji je 2008. godine kupio Sun Microsistems. Nakon toga, 2010. godine Oracle Corporation je kupio Sun Microsistems.

## MySQL alati

MySQL uključuje brojne klijente i alate za rad sa MySQL bazama podataka. Postoje i mnogi drugi alati napisani od strane open source community-a, posebno za rad sa MySQL-om. Neki od najpopularnijih su:

* **MySQL Workbench** - grafički alat za rad sa MySQL-om. Može se koristiti za kreiranje i upravljanje bazama podataka, izvođenje upita, zaustavljanje ili pokretanje MySQL-a, konfigurisanje i pregled korisničkih privilegija i prava pristupa, izrada šema dijagrama, kreiranje relacija i provera statusa servera itd.
* **Mysqladmin** je program koji vam pruža MySQL jezik, koji vam omogućava da proverite konfiguraciju i trenutni status servera, kreirate i obrišete baze podataka itd.
* **Mysqldump** je uslužni program u komandnoj liniji koji pruža MySQL za logički backup MySQL baza podataka. Logički backup kopija je ona koja reprodukuje strukturu tabele i podatke, bez kopiranja stvarnih podataka u datotekama.
* **Myisamchk** je program koji pruža MySQL koji može uzeti informacije iz tabele iz baze podataka. Takođe ih može proveriti, popraviti i optimizovati. myisamchk radi posebno sa MyISAM tabelama (tabele koje sadrže .MID i .MII datoteke za čuvanje podataka i indeksa).
* **PhpMyAdmin** je besplatni softver napisan u PHP-u, namenjen upravljanju administracijom MySQL-a preko Interneta. Uobičajeni zadaci kao što su upravljanje bazama podataka, tabelama, kolonama, odnosima, indeksima, korisnicima, dozvolama itd. mogu se obavljati putem GUI-ja. PhpMyAdmin takođe pruža mogućnost pokretanja SQL izraza prema bazi podataka.

## SQL (Structured Query Language)

SQL je akronim od Engleskih reči "Structured Query Language", što bi u prevodu značilo strukturalni upitni jezik. Dakle, SQL je upitni jezik koji koristimo za izdavanje komandi za manipulaciju podacima baze podataka, kao i za kompletno njeno kreiranje, izmenu ili brisanje, kako same baze, njenih tabela, tako i podataka koji se nalaze u bazi podataka.

SQL je jedan od prvih komercijalnih jezika Edgar F. Codd-ovog relacionog modela. SQL je standardizovan 1986. godine, pa slede dopune standarda 1989. i 1999., ali i pored toga nije u potpunosti prenosiv između različitih sistema za upravljanje relacionim bazama podataka. Ovo znači da njegova sintaksa nije usklađena i kompatibilna između različitih sistema. Tako da se SQL sintaksa delimično razlikuje kod MySQL-a i na primer SQLite-a.

### SQL Sintaksa

* SQL nije osetljiv na velika i mala slova;
* Svaki SQL upit (ukoliko ih ima više) mora da se završi oznakom ; (ovaj separator se može promeniti);
* Stringovi se pišu isključivo u navodnicima;
* MySQL prihvata i jednostruke i dvostruke navodnike za string, ali je najbolja praksa koristiti jednostruke;
* Objekti u SQL-u se mogu staviti u Backtick navodnike (select \* from `my table`);
* promenljive u SQL-u uvek počinju oznakom @

### Tekstualne konstante

String predstavlja sekvencu bajtova ili karaktera, koji su ograničeni ili jednostrukim navodnicima (') ili dvostrukim navodnicima (").

Generalno, postoje dva tipa stringova u MySQL-u i to su binarni stringovi i nebinarni stringovi, odnosno stringovi koji su sastavljeni od karaktera. Razlika između ove dve vrste stringova je u osnovnoj gradivnoj jedinici koja je za binarne stringove, naravno, bajt, dok je za nebinarne stringove to karakter. Još jedna razlika je veoma bitna, a to je činjenica da se za binarne stringove ne može definisati set karaktera ili kolacija, s obzirom da se čuvaju kao sekvenca bajtova.

### Komentari

MySQL server podržava tri vrste komentara:

* jednolinijski komentar, navođenjem karaktera #
* jednolinijski komentar navođenjem karaktera –
* višelinijski komentar koji je kreiran na isti način kao i u programskom jeziku C, korišćenjem karaktera /\* i \*/ gde sve između ovih oznaka predstavlja komentar.

MySQL server poznaje i jednu vrstu specijalnih komentara, koji se nazivaju uslovni komentari.

/\*! MySQL-specific code \*/

### SQL Naredbe

SQL naredbe se dele na tri seta:

* **Data Definition Language (DDL)** - Jezik za definiciju podataka

Definicione SQL naredbe (DDL) čine tri naredbe: CREATE, ALTER i DROP. Isključivo se tiču strukture, a ne samih podataka, što znači da, uz pomoć njih nikada nećete dodavati ni jedan jedini podatak u bazu. Kada želite da kroz SQL skriptu stvorite bazu podataka, ili neke njene entitete, koristićete upravo ovaj set naredba.

* **Data Manipulation Language (DML)** - Jezik za manipulaciju podacima

Definicione SQL naredbe (DML) čine četiri naredbe: SELECT, INSERT, UPDATE i DELETE. Jezik za manipulaciju podacima omogućava korisnicima da pristupe i preuzmu sadržaj baze podataka u interaktivnom i ugrađenom (u programskom jeziku domaćina) režimu. DML uključuje konstrukcije koje omogućavaju pronalaženje i manipulaciju sadržajem baze podataka. Češći i popularniji termin koji se koristi za DML je „jezik upita“. ISO standardni SKL jezik je najpopularniji DML za relacione baze podataka. DML može biti deklarativni.

* **Data Control Language (DCL)** - Jezik kontrole podataka

DCL je najjednostavniji od SKL podskupova, jer se sastoji od samo tri komande: GRANT, REVOKE i DENI. Kombinovano, ove tri komande pružaju administratorima fleksibilnost da podese i uklone dozvole baze podataka na detaljan način. Jezik kontrole podataka je podskup jezika strukturiranih upita. Administratori baze podataka koriste DCL da konfigurišu bezbednosni pristup relacionim bazama podataka. On dopunjuje jezik definicije podataka, koji dodaje i se preuzimaju iz baze podataka ili proceduralni gde upotreba specificira kako se pristupiti podacima. Od modernih DBMS-a se očekuje da imaju deklarativne DML-ove (kao što je SKL), što omogućava DBMS-u da ih konvertuje u izraz relacione algebre pre nego što optimizuje njihovo izvršenje. Postoje i drugi jezici za manipulaciju podacima za specifične DBMS – npr. SPARKL za RDF podatke briše objekte baze podataka, i jezik za manipulaciju podacima, koji preuzima, umeće i menja sadržaj baze podataka.

# Praktična primena upita u Mysql bazi podataka

Najpre će biti više reči o kreiranim tabelama u MySQL bazi podataka, a zatim sledi objašnjenje o evaluaciji operatora nad napravljenom bazom podataka.

## Kreirane tabele u MySQL bazi podataka

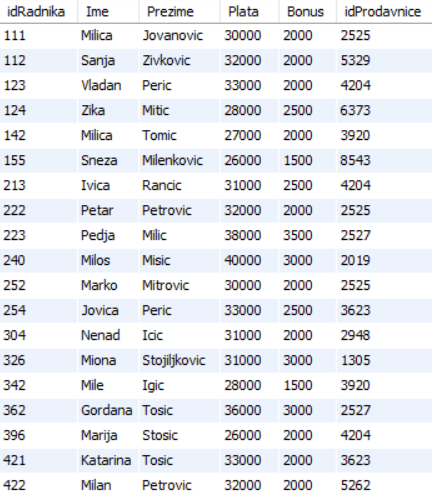
U MySQL bazi podataka, kreirane su tri tabele koje su međusobno povezane. Svaka tabela sadrzi primarni ključ (ID) i strani ključ kojim je povezana sa drugom tabelom. Baza podataka odnosi se na trgovinski lanac prodavnica u Nišu i nosi naziv “trgovina-Niš”. Kreirane tabele u bazi nose naziv prodavnice, radnici i fakture. Na slici 1, prikazana je tabela pod nazivom Prodavnice.

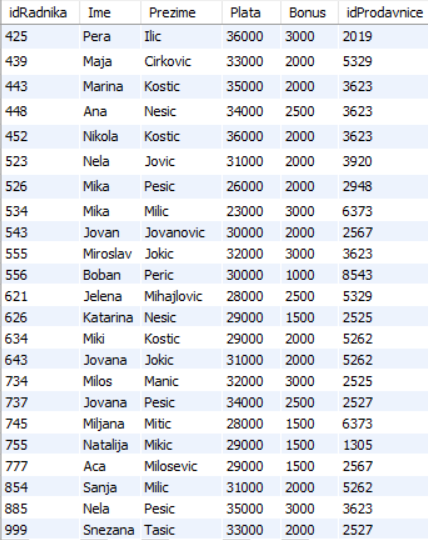


Slika 1. Tabela pod nazivom Prodavnice

Tabela prodavnice sadrži podatke kao što su id prodavnice koji predstavlja primarni ključ za ovu tabelu, šifra prodavnice, naziv prodavnice, mesto i broj radnika.

Na slici 2, prikazana je tabela pod nazivom Radnici.

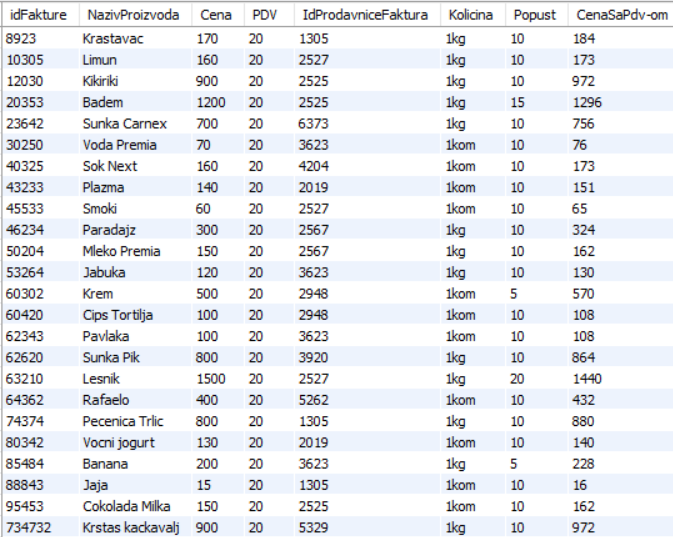




Slika 2. Tabela pod nazivom Radnici

Tabela radnici sadrži podatke kao što su id radnika koji predstavlja primarni ključ za ovu tabelu, ime i prezime radnika, plata, bonus, id prodavnice koji predstavlja strani ključ za ovu tabelu putem kojeg je ona povezana sa tabelom Prodavnice gde primarni ključ predstavljaju podaci iz ove kolone.

Na slici 3, prikazana je tabela Fakture.



Slika 3. Tabela pod nazivom Fakture

Tabela fakture sadrži podatke kao što su id fakture, koji predstavlja primarni ključ za ovu tabelu, naziv proizvoda, cena, PDV, id prodavnice faktura koji predstavlja strani ključ kojim je ova tabela povezana sa tabelom prodavnice, količina i popust.

Nakon kreiranja tabela, mogu se izvršavati upiti i manipulacije nad podacima iz tabela. U ovom radu biće detaljnije opisana evaluacija operatora nad podacima iz prikazanih tabela u bazi.

## Evaluacija operatora prilikom obrade upita

U upitima za selektovanje podataka se, u WHERE klauzuli definišu uslovi koje moraju da zadovoljavaju zapisi koji se traže. Sam uslov je logički izraz koji ima logičku vrednost tačno ili netačno (*True* ili *False*). Podaci će biti prikazani samo za one zapise koje zadovoljavaju taj uslov. Sam logički izraz može biti kombinacija manje složenih izraza koji su povezani logičkim operatorima AND, ORi NOT. Oni su samo jedna vrsta operatora o kojima će ovde biti više reči. Operatori omogućavaju da se dodaju brojevi, upoređuju vrednosti, spajaju stringovi (grupa tekstualnih karaktera) i kreiraju kompleksni relacioni izrazi. Operatori naznačavaju da je potrebno da se određena operacija sprovede nad jednim ili više elemenata. Razlikuju se sledeće vrste operatora:

* Matematički (aritmetički) operatori
* Relacioni operatori (za poređenje vrednosti)
* Logički operatori

### Matematički operatori

U matematičke operatore spadaju:

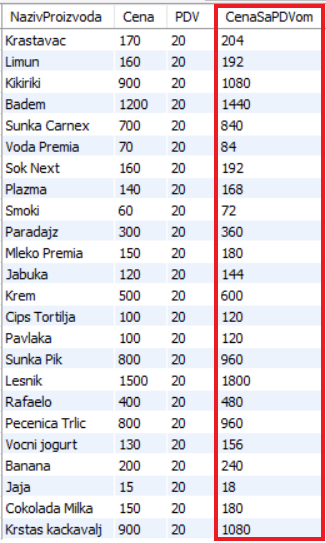
|  |  |
| --- | --- |
| \* | Množenje |
| + | Sabiranje |
| - | Oduzimanje |
| / | Deljenje |

Matematički operatori se koriste za rad sa brojevima, odnosno sa bilo kojim numeričkim tipom podataka. Broj može biti konstantna vrednost, vrednost promenljive ili sadržaj polja.

#### \* operator (množenje)

Na praktičnom primeru biće prikazano korišćenje ovog operatora za računanje cene sa PDV-om. U bazi podataka trgovina-Niš u tabeli fakture unose se podaci, izmedju ostalog, o ceni proizvoda i popustu za svaki proizvod. Na osnovu ovih podataka, moze se izračunati cena sa PDV-om za svaki proizvod prilikom čega se koristi operator množenje. U ovom slučaju kolona za izračunavanje će sadržati formulu Cena + (Cena \* PDV/100). Koristi se tabela fakture.

SELECT NazivProizvoda, Cena, PDV, Cena + (Cena \* PDV/100)

AS CenaSaPDVom

FROM fakture;

#### + operator (sabiranje)

Ukoliko želimo da formiramo kolonu koja sabira vrednosti iz drugih kolona, na primer, kolona Plata i kolona Bonus, tada bi se koristio izraz Plata+Bonus. Ova formula koristi operator za sabiranje radi sabiranja vrednosti iz obe kolone i prikazivanja rezultata u koloni koji sadrži formulu.

SELECT Ime, Prezime, Mesto, Plata, Bonus, (Plata+Bonus)

AS Plata\_sa\_bonusom FROM radnici





#### - operator (oduzimanje)

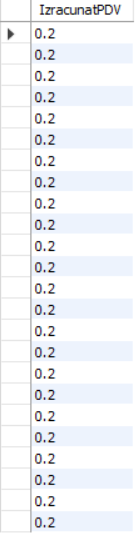
Primer korišćenja ovog operatora je oduzimanje popusta od konačne cene na fakturi. Formula bi bila: Cena - (Cena \* Popust/100). Iako zagrade nisu matematički operatori, one igraju važnu ulogu u radu sa operatorima, jer određuju redosled izvršavanja operacija.

SELECT NazivProizvoda, Cena, Popust, Cena - (Cena \* Popust/100)

AS CenaSaPopustom

FROM fakture;

#### / operator (deljenje)

Ovaj operator se može koristiti za deljenje dva broja i prikazivanje rezultata na željenom mestu. Na primer, može se iskoristiti za izračunavanje PDV-a koji je izražen u procentima.

SELECT PDV/100 AS IzracunatPDV

FROM fakture;

### Relacioni operatori (operatori za poređenje)

Postoji šest osnovnih relacionih operatora koji su poznati i kao operatori poređenja. Oni porede dve vrednosti ili izraza. Relacioni operatori su:

|  |  |
| --- | --- |
| = | jednako |
| < >, != | nije jednako |
| < | manje |
| < = | manje ili jednako |
| > | veće |
| > = | veće ili jednako |

Izrazi koji su formirani korišćenjem relacionih operatora vraćaju ili YES (tačno) ili NO (netačno) ili NULL (nepoznato/nema vrednosti). Ako bilo koja strana izraza koji koriste ove operatore ima NULL vrednost, rezultat će uvek biti NULL vrednost.

#### = operator (jednako)

Ovaj operator vraća logičku vrednost tačno ako su dva izraza koji se porede ista. BrojRadnika=3 vraća vrednost tačno ako je broj radnika 3, odnosno netačno ako se radi o broju radnika različitom od 3. Dakle, ako treba prikazati samo nazive prodavnica koje imaju po 3 radnika, koristi se sledeći upit:

SELECT NazivProdavnice, Mesto, BrojRadnika

FROM prodavnice

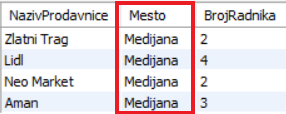
WHERE BrojRadnika=3;

U većini slučajeva, pri poređenju vrednosti znakovnog tipa u MySQL-u se ne pravi razlika između malih i velikih slova. Ako se želi da se znakovne vrednosti porede tako da se ipak pravi razlika između malih i velikih slova, treba ispred jedne od njih dodati prefiks BINARY. Primer pretrage po mestu firme koja ne pravi razliku izmedju malih i velikih slova:

SELECT NazivProdavnice, Mesto, BrojRadnika

FROM prodavnice

WHERE Mesto= "medijana";



Primer pretrage po mestu prodavnice koja pravi razliku izmedju malih i velikih slova:

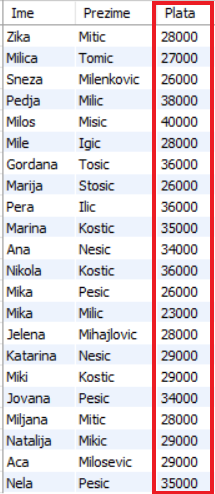
SELECT NazivProdavnice, Mesto, BrojRadnika

FROM prodavnice

WHERE Mesto= BINARY "medijana";

Ovakvim upitom neće se vratiti ni jedna vrednost jer ni jedna prodavnica nema naziv mesta medijana, već Medijana.

#### <> (nije jednako)

Ovaj operator je suprotnost operatora jednako. Kriterijum Plata<>31000 vraća vrednost tačno ako se radi o plati koja je različita od 31000. Za ovaj operator mogu se koristiti oznake <> i !=.

SELECT Ime, Prezime, Plata

FROM radnici

WHERE Plata<>30000

AND Plata<>31000

AND Plata<>32000 AND Plata<>33000;

#### < (manje)

Ovaj operator vraća vrednost tačno ako je leva strana izraza manja od desne.

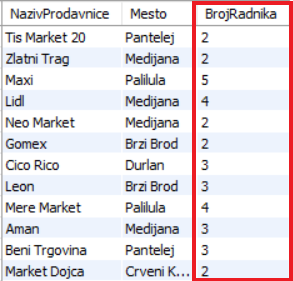


SELECT NazivProizvoda, Cena

FROM fakture

WHERE Cena < 500;

#### <= (manje ili jednako)

Ovaj operator vraća vrednost tačno ako je leva strana izraza manja ili jednaka u odnosu na desnu.

SELECT NazivProdavnice, Mesto, BrojRadnika

FROM prodavnice

WHERE BrojRadnika <=6;

#### > (veće)

Ovaj operator je suprotan od operatora manje. Ovaj operator vraća vrednost tačno ako je leva strana izraza veća od desne:



SELECT Ime, Prezime, Plata

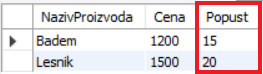
FROM radnici

WHERE Plata>33000;

#### >= (veće ili jednako)

Ovaj operator vraća vrednost tačno ako je leva strana izraza veća ili jednaka u odnosu na desnu.

SELECT NazivProizvoda, Cena, Popust

FROM fakture

WHERE Popust >=15;

### Predikati

Postoje i drugi operatori za poređenje koji se zovu predikati:

BETWEEN ... AND (između dve vrednosti)

IN (u skupu vrednost)

IS NULL (vrednost u polju ne postoji)

#### BETWEEN ... AND operator

Ovaj operator se koristi da se utvrdi da li se vrednost nalazi u definisanom opsegu vrednosti. Sintaksa je:

<Izraz > BETWEEN <vrednost 1> AND <vrednost 2>

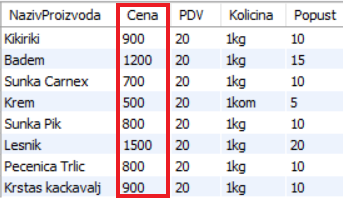
Ako je vrednost između navedenih vrednosti rezultat je tačno, a u suprotnom je rezultat netačno.

Sledeći upit prikazuje samo one stavke faktura kod kojih je cena između 500 i 1500:

SELECT NazivProizvoda, Cena, PDV, Kolicina, Popust

FROM fakture

WHERE Cena BETWEEN 500 AND 1500;



#### IN operator

Ovaj operator se koristi da bi se odredilo da li se vrednost poklapa sa nekom vrednošću iz liste vrednosti. Sintaksa je:

<Izraz> IN <vrednost 1, vrednost 2, vrednost 3, …>

Ako je vrednost nađena u listi rezultat je tačno, a u suprotnom je rezultat netačno.

Primer: Napraviti upit koji nalazi sve prodavnice koje su u mestu Pantelej, Medijana ili Palilula.



SELECT NazivProdavnice, Mesto

FROM prodavnice

WHERE Mesto IN ('Pantelej', 'Medijana','Palilula');

#### IS NULL

Koristi se da bi seodredilo da li objekat sadrži nešto u sebi, tj da li n ima vrednost NULL. Sintaksa glasi: n IS NULL

Primer: prodavnica koja nema ni jednu fakturu:

SELECT prodavnice. NazivProdavnice

FROM prodavnice LEFT JOIN fakture

ON prodavnice.id = fakture. IdProdavniceFaktura

WHERE IdProdavniceFaktura IS NULL;

### Logički operatori

Postoji više logičkih operatora. Oni se koriste za postavljanje uslova u izrazima. Logički operatori se koriste da se naprave kompleksni izrazi sa više uslova. Kao i relacioni operatori uvek vraćaju vrednosti tačno, netačno ili NULL. Logički operatori su:

|  |  |
| --- | --- |
| AND ili && | logičko **i** |
| OR ili || | logičko uključivo **ili** |
| XOR | logičko isključivo **ili** |
| NOT ili ! | logičko **ne** |

#### AND operator (logičko i)

Ovaj operator vraća vrednost tačno samo ako su oba uslova tačna. Sintaksa je:

<Izraz 1> AND <Izraz 2> *ime* = 'Nela' AND *prezime* = 'Jovic' je tačno samo ako su oba uslova tačna. Sledeći upit naći će prodavnice koje su u mestu Medijana i imaju broj radnika veći od 1.

SELECT NazivProdavnice, Mesto, BrojRadnika

FROM prodavnice

WHERE Mesto = 'Medijana' AND BrojRadnika > 1;

Pri tome treba obratiti pažnju na sledeće:

True AND True = True

False AND bilo šta = false

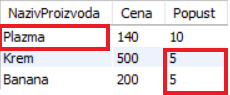
Svi ostali izrazi se svode na NULL.

#### OR operator (logičko uključivo ili)

Ovaj operator vraća vrednost tačno ako je bar jedan od izraza tačan. Sintaksa je:

<Izraz 1> OR <Izraz 2>

Donjim upitom biće pronađeni svi proizvodi čiji je naziv četka ili je jedinica mere kilogram:

SELECT NazivProizvoda, Cena, Popust

FROM fakture

WHERE NazivProizvoda = 'Plazma' OR popust = 5;

Pri tome, treba voditi računa o sledećem:

True OR bilo šta = True

NULL OR False = NULL

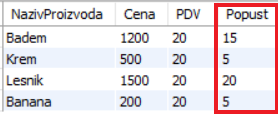
NULL OR NULL = NULL

False OR False = False

#### NOT operator (logičko ne)

Ovaj operator se koristi za negaciju. Operator vraća vrednost tačno ako uslov nije tačan. Ovaj operator menja logički rezultat izraza. Za NULL vraća NULL. Sintaksa glasi:

NOT <Izraz>

NOT Popust=10 vraća tačno ako je popust različit od 10, odnosno prikazuje samo redove sa popustom koji je različit od 10.

SELECT NazivProizvoda, Cena, PDV, Popust

FROM fakture

WHERE NOT Popust=10;

#### XOR operator (logičko isključivo ili)

Vraća vrednost tačno jedino u slučajevima da jedan od izraza ima vrednost tačno ili oba izraza imaju vrednost tačno. Ukoliko jedan od izraza ima vrednost NULL, vraća vrednost NULL. SELECT NazivProizvoda, Cena, PDV, Popust

FROM fakture

WHERE Popust=10 XOR NOT PDV=20;

# Zaključak

Cilj ovog rada je, da kroz praktične primere pokaže kako se koriste operatori u Mysql bazi podataka korišćenjem upita. U MySQL Workbench-u definisani su upiti korišćenjem operatora na osnovu čega dobijamo zahtevane rezultate iz bilo koje tabele u bazi. Korišćenjem operatora mogu se dobiti nove kolone sa potrebnim podacima, mogu se formirati složeni upiti i moguće je manipulisati podacima iz tabela. Bez korišćenja operatora ne bi bilo moguće formirati upite pomoću kojih pronalazimo tražene podatke iz tabele ili formiramo nove.

# Literatura

1. https://www.nkod.rs/latinica/blog/sta-je-mysql;

2. https://cubes.edu.rs/sr/37/obuke-i-kursevi/sta-je-mysql;

3. https://totohost.hr/2019/08/02/kreiranje-i-administriranje-mysql-baze-podataka;

4. https://link.springer.com/referenceworkentry/10.1007/978-1-4899-7993-3\_80654-1;

5. https://www.lifewire.com/data-control-language-dcl-1019477;

6. http://icit.masfak.ni.ac.rs/uploads/articles/icit2\_mysql\_6.pdf;

7. https://phoenixnap.com/kb/how-to-create-mysql-database-workbench;

8. https://www.codeblog.rs/clanci.php?p=relacione\_baze\_podataka;

9. https://www.careerride.com/MySQL-Operator.aspx;

10. http://tfzr.rs/Content/files/0/MySQL%20Tutorijal%20deo%204.pdf