

# MYSQL sustav za upravljanje bazom podataka otvorenog koda

---

**Valjak, Silvio**

**Undergraduate thesis / Završni rad**

**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Polytechnic Nikola Tesla in Gospić / Veleučilište Nikola Tesla u Gospiću**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:107:942456>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-13**



*Repository / Repozitorij:*

[Polytechnic Nikola Tesla in Gospić - Undergraduate thesis repository](#)



VELEUČILIŠTE „NIKOLA TESLA“ U GOSPIĆU

Silvio Valjak

**MYSQL SUSTAV ZA UPRAVLJANJE BAZAMA PODATAKA  
OTVORENOG KODA**

Završni rad

Gospić, 2015.



VELEUČILIŠTE „NIKOLA TESLA“ U GOSPIĆU

POSLOVNI ODJEL

Stručni studij Ekonomika poduzetništva

**MYSQL SUSTAV ZA UPRAVLJANJE BAZAMA PODATAKA  
OTVORENOG KODA**

Završni rad

MENTOR

Dr. sc. Aleksandar Skendžić

STUDENT

Silvio Valjak

MBS: 0296002223/06

Gospić, rujan 2015.

POSLOVNI odjel

Gospić, \_\_\_\_\_ 20\_\_.

## ZADATAK

za završni rad

Pristupniku SILVIO VALJAK MBS: 0296002223/06

Studentu stručnog studija EKONOM. PODUZETNIŠTVA izdaje se tema završnog rada pod nazivom

MYSQL SVUSTAV ZA UPRAVLJANJE BAZAMA PODATAKA  
OTVORENOG KODA

Sadržaj zadatka :

1. BAZA PODATAKA, TIPOVI I PRIMJENE
2. OTVORENI KOD, ŠTO JE OTVORENI KOD I ZAŠTO KORISTITI BAZE PODATAKA OTVORENOG KODA
3. SVUSTAVI UPRAVLJANJA BAZAMA PODATAKA, NJIHOVA KVALIFIKACIJA I PRIMJERI
4. MYSQL, NJEGOVA SVOJSTVA I KARAKTERISTIKE
5. PRIMJER IZ PRAKSE, PREDNOST BAZE PODATAKA OTVORENOG KODA

Završni rad izraditi sukladno odredbama Pravilnika o završnom radu Veleučilišta „Nikola Tesla“ u Gospiću.

Mentor: DR. SC. ALEKSANDAR SKENDŽIĆ zadano: \_\_\_\_\_, \_\_\_\_\_  
(ime i prezime) (nadnevak) potpis

Pročelnik odjela: MR. SC. TOMISLAV ŽURČIĆ predati do: 30.9.2015., \_\_\_\_\_  
(ime i prezime) (nadnevak) potpis 16.09.2015

Student: SILVIO VALJAK primio zadatak: \_\_\_\_\_, \_\_\_\_\_  
(ime i prezime) (nadnevak) potpis

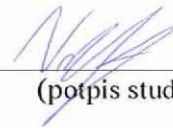
Dostavlja se:

- mentoru
- pristupniku

## IZJAVA

Izjavljujem da sam završni rad pod naslovom „MySQL sustav za upravljanje bazama podataka otvorenog koda“ izradio samostalno pod nadzorom i uz stručnu pomoć mentora Dr. sc. Aleksandra Skendžića.

Ime i prezime



---

(potpis studenta)

## **Sažetak**

U modernom poslovnom i institucionalnom svijetu upravljačka tijela raspolažu sa sve više resursa, odnosno sa sve većom količinom podataka. Kroz stvaranje baza podataka neophodno je olakšati prikupljanje podataka i skratiti put do traženih podataka. U ovom radu opisani su osnovni pojmovi vezani uz temu završnog rada radi lakšeg shvaćanja tematike koja obuhvaća izradu i implementaciju baze podataka, te problematike na koju nailaze osobe koje su na direktan način uključene u evidenciju i prodaju robe te prateću dokumentaciju u procesu poslovanja. Ideja za izradu baze podataka se sama po sebi nametnula zbog obima podataka koji se pohranjuju i obrađuju.

## Sadržaj:

1. Uvod .....	1
2. Baza podataka.....	2
2.1. Modeli baze podataka .....	4
2.1.1. Hijerarhijski model .....	4
2.1.2. Mrežni model.....	5
2.1.3. Relacijski model .....	7
2.1.4. Objektni model .....	8
2.2. Tipovi baza podataka.....	9
3. Otvoreni kod (eng. " <i>Open source</i> ").....	11
3.1. Dostupnost i prednosti otvorenog koda.....	14
4. Sustavi upravljanja bazama podataka.....	16
4.1. Primjeri sustava za upravljanje bazama podataka .....	17
4.1.1. MySQL .....	18
4.1.2. Microsoft SQL Server .....	18
4.1.3. Microsoft Office Access .....	18
4.1.4. Microsoft Visual FoxPro .....	19
4.1.5. PostgreSQL.....	19
4.1.6. SQLite.....	19
4.1.7. Firebird .....	20
4.1.8. Oracle.....	20
5. MySQL .....	21
5.1. Prednosti MySQL-a.....	22
5.1.1. Brzina .....	22



5.1.2. Pouzdanost.....	23
5.1.3. Rad na različitim platformama .....	23
5.1.4. Podržava veliku količinu programskih jezika .....	23
5.1.5. ODBC .....	24
5.1.6. Besplatno ili povoljno licenciranje .....	24
5.1.7. Povoljna komercijalna podrška .....	24
5.1.8. Dostupnost izvornog koda.....	25
5.1.9. Mala zahtjevnost za sistemskim resursima.....	25
5.2. Nedostaci MySQL-a.....	25
5.2.1. Nedostatak nekih mogućnosti SQL-a.....	25
5.2.2. Nedostatak ozbiljnih testiranja na određenim platformama .....	26
5.2.3. Složeno rukovanje izvornim kodom.....	27
6. Primjer MySQL baze podataka .....	28
6.1. Stvaranje tablica .....	29
6.2. Kreiranje veza između tablica .....	31
6.3. Unos podataka u kreirane tablice.....	33
6.4. Prebacivanje baze na poslužitelj.....	33
6.5. Prednosti MySQL baze podataka otvorenog koda .....	34
7. Zaključak .....	<b>Error! Bookmark not defined.</b>
Popis literature.....	36
Popis slika.....	39
Popis priloga.....	40

## 1. Uvod

**Predmet završnog rada:** MySQL sustav za upravljanje bazama podataka otvorenog koda, modeli baza podataka, zašto koristiti otvoreni kod.

Opisani su osnovni pojmovi vezani za baze podataka te njihove modele, sustave za upravljanje bazama podataka, otvoreni kod, te kako obraditi MySQL kao najpopularniji sustav za upravljanje bazama podataka otvorenog koda.

**Cilj završnog rada** je na temelju svih prikupljenih informacija i podataka ustanoviti važnost i efikasnost sustava za upravljanje bazama podataka otvorenog koda kao i njegovu isplativost. Na temelju podataka upoznati se sa radom u bazama podataka, te obraditi svaku cjelinu odnosno nešto detaljnije reći o njima.

### **Izvor podataka**

Pri izradi završnog rada korišteni su podaci iz dostupne literature na engleskom jeziku. Paralelno je korištena dostupna literatura. Rad se sastoji od 5 poglavlja u kojima su objašnjeni pojmovi i znanja vezana uz upravljanje bazama podataka otvorenog koda.

# 1. Baza podataka

Baze podataka i sustavi za upravljanje bazama podataka su neophodna komponenta života u suvremenom društvu: većina ljudi će naići na nekoliko aktivnosti svaki dan koje uključuju određene interakcije s bazom podataka. Kao primjer, odlazak u banku zbog uplate ili povlačenja sredstava, hotelske ili zrakoplovne rezervacije, pristupi elektroničkim katalozima knjižnica, ili online kupovina kao što su knjige, igračke ili računalo, šanse su da će aktivnosti ljudi uključivati nekoga ili neki računalni program koji pristupaju bazi podataka. Kupnja stvari u supermarketu često automatski ažurira bazu podataka koja sadrži inventar trgovine.

- Baza podataka je skup konstantnih podataka koji se koristi od strane aplikacijske sustava nekog određenog poduzeća. (C.J. Date, 2003, str. 11)

Pojam poduzeća ovdje je praktičniji generički naziv za bilo koju komercijalnu, znanstvenu, tehničku ili drugu organizaciju. Poduzeće može biti i pojedinac (s malom osobnom bazom podataka), ili kompletna korporacija ili slično veliko tijelo (s velikom zajedničkom bazom podataka), ili nešto između. Neki primjeri:

- Tvrtka za proizvodnju
- Banka
- Bolnica
- Sveučilište
- Odjel u vladi

Svako poduzeće mora nužno održavati masu podataka o svom radu. To su "konstantni podatci" iz definicije. Spomenute organizacije obično uključuju slijedeće među svojim osobnim podacima:

- Podatke o proizvodu
- Podatke o računu

- Podatke o pacijentu
- Podatke o studentima
- Podatke o planovima

Baza podataka sadrži povezane podatke, kao i opis tih podataka. Iz tog razloga, baza podataka je također definirana kao samo opisujuća kolekcija integriranih zapisa. Opis podataka je poznat kao sistemski katalog ili meta podatci. Na primjer, kada se pravi tablica baze podataka, tip podataka, veličina, oblik i ostali opis polja su navedeni. Meta podatci opisuju svojstva podataka koji su pohranjeni u polju tablice.

Podatci bilo koje organizacije su njegov sastavni dio. Podatci su vrlo važni za razvoj novih proizvoda i njihov marketing. Podaci moraju biti točni i dostupni kada je to potrebno. To je razlog da sve organizacije moraju organizirati i upravljati svojim podacima u bazama podataka. Gotovo sve organizacije i vladini odjeli svake države koriste baze podataka za održavanje svoje evidencije.

Primjer velike komercijalne baze podataka je Amazon.com. Ista sadrži podatke za preko 20 milijuna knjiga, CD-a, DVD-a, videa, igara, elektronike, odjeće i ostalih predmeta. Baza podataka zauzima više od 2 TB prostora i sprema se na 200 različitih servera. Oko 15 milijuna posjetitelja pristupi Amazon.com svaki dan i koristiti bazu za kupnju. Baza podataka neprestano se ažurira novim knjigama i drugim predmetima koji se dodaju u inventar zaliha, a količine se ažuriraju prilikom obavljanja transakcija. Oko 100 ljudi je odgovorno za upravljanje Amazon bazom podataka. (R. Elmasri, S. Navathe. 2010 str. 5)

## **1.1. Modeli baze podataka**

Model podataka određuje logičku strukturu baze podataka i bitno određuje na koji način podaci mogu biti pohranjeni, organizirani i manipulirani. Najpopularniji primjer modela baze podataka je relacijski model, koji se koristi formatom tablica i relacija među njima (R. Elmasri, S. Navathe. 2010. str. 44).

Postoje četiri modela baze podataka a to su (Ramakrishnan R., Gehrke J. 2002. str. 12):

1. Hijerarhijski model
2. Mrežni model
3. Relacijski model
4. Objektni model

### **1.1.1. Hijerarhijski model**

Hijerarhijski model baze podataka je model podataka u kojima su podatci organizirani u strukturu poput stabla kao što je prikazano na slici 1. Struktura predstavlja podatke koristeći roditelj/dijete odnos: svaki roditelj može imati puno djece, ali svako dijete ima samo jednog roditelja.

U terminologiji matematičkog drveta, element na jednoj strani relacije “jedan-prema-više” naziva se element višeg nivoa ili roditelj, koji je povezan sa elementima nižeg nivoa. U hijerarhijskoj bazi podataka, korijen mora biti samo roditelj, a svaki dodatak na roditelja je dijete u točno jednoj relaciji roditelj-dijete (<http://www.extropia.com> 10. 05. 2015.).

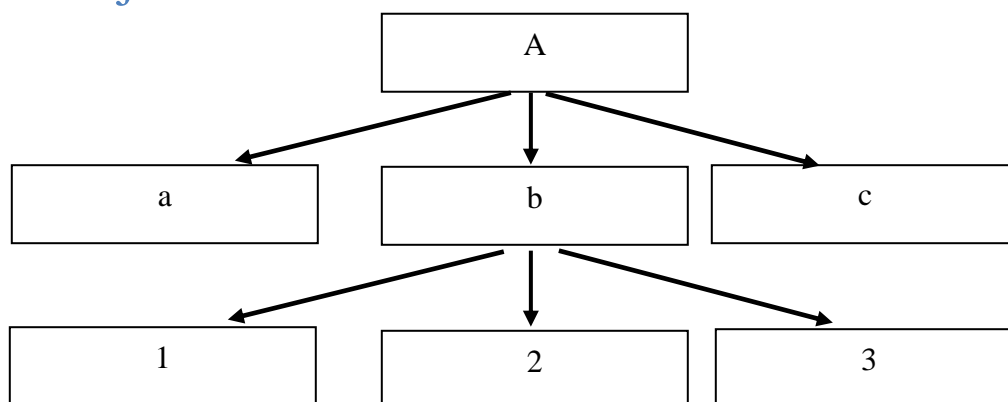
Hijerarhijska struktura je korištena u početcima sustava za upravljanje bazama podataka i još uvijek se koristi od strane nekih organizacija za izradu putnih rezervacija. Zapisuje se u obliku stabla, ova struktura je jednostavna, ali nepopustljiva jer odnos je ograničen na jedan-na-više. IBM Information System (IMS) i RDM Mobile primjeri su hijerarhijskog sustava baze podataka s višestrukim hijerarhijama nad istim podacima.

Prednost hijerarhijskih baza podataka je brže pristupanje i ažuriranje, jer su strukture kao stabla i odnosi između zapisa unaprijed definirani. Međutim ova značajka je dvosjekli mač. Nedostatak ove vrste strukture baze podataka je da svako dijete u stablu može imati samo jednog roditelja, a odnosi ili veze između djece nisu dopušteni, čak i ako oni imaju smisla iz logičkog stajališta. Hijerarhijske baze podataka su toliko krute svojim dizajnom da dodavanje novog polja ili zapisa zahtjeva da cijelu bazu podataka treba redefinirati.

Trenutno najčešće korišteni hijerarhijski modeli su baze podataka IMS razvijene od strane IBM-a i Windows Registry od strane Microsofta. Ovaj model je priznat kao prvi model baze podataka stvorene od strane IBM-a 1960-ih.

Slika 1. Primjer hijerarhijskog modela

## Hijerarhijski model



Izvor: Vlastita izrada 19. 07. 2015.

### 2.1.2. Mrežni model

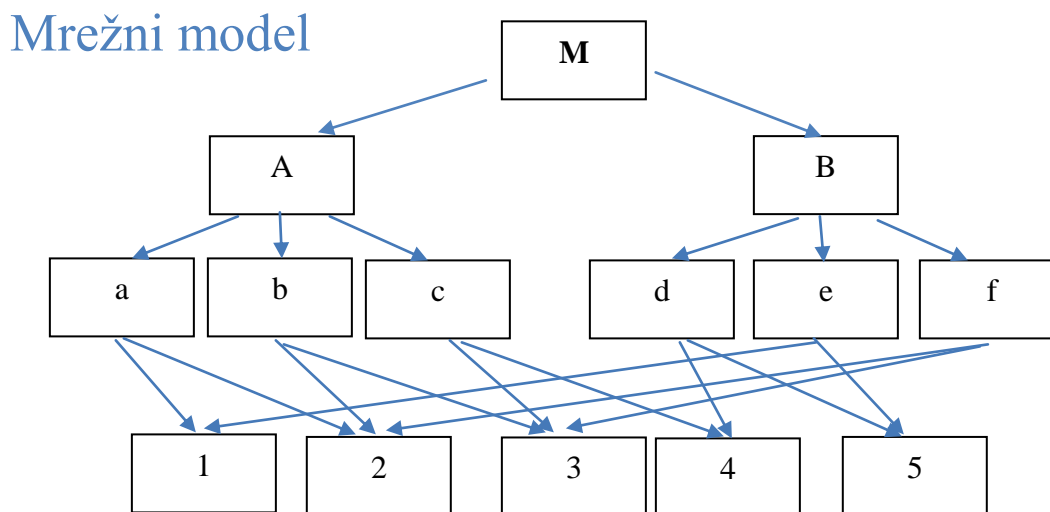
Mrežni model je model podataka zamišljen da na fleksibilan način predstavlja objekte i njihove odnose. Dok su kod hijerarhijskog modela baze podataka strukture podataka kao stablo zapisa, a svaki zapis ima jednog roditelja i puno djece, model mreže omogućuje svakom zapisu da ima više veza s roditeljem i djecom i tako stvara oblik strukture grafa.

Mrežni model baze podataka izumio je Charles Bachman 1969. kako bi se poboljšao postojeći hijerarhijski model baze podataka. On je stvorio ovaj model kako bi povećao fleksibilnost i učinio ga lakšim za razumjeti. Da bi to promijenio Bachman je razvio mrežni model baze podataka kako bi se omogućilo više zapisa koji će biti povezani s istim vlasnikom datoteke, vezama više-na-više, a ne jedan-na-više. Odnosi više-na-više su nastali iz razloga što se jedan vlasnik može povezati s puno datoteka i obratno. Da bi ubrzao proces Bachman je stvorio navedene veze da omogući povezivanje datoteka direktnim pokazivačima, a to je ujedno i najbrži način da bi se prošlo kroz mrežni model baze podataka (<http://databasemanagement.wikia.com> 20. 05. 2015).

Dva ograničenja moraju se uzeti u obzir prilikom korištenja ove vrste baze podataka. Slično hijerarhijskim bazama podataka, mrežne baze podataka moraju biti unaprijed definirane. Tu je i limit na broj veza koje mogu biti između zapisa.

Neki od poznatijih sustava koji koriste mrežni model su: Integrated Data Store, TurboIMAGE i Univac DMS-1100.

Slika 2: Primjer mrežnog modela



Izvor: Vlastita izrada 19. 07. 2015.

### 2.1.3. Relacijski model

Relacijska baza podataka temelji se na relacijskom modelu razvijenom od strane E. F. Codd. Relacijska baza podataka omogućuje definiranje strukture podataka, pohranu i dohvat te ograničenja integriteta. U takvoj bazi podataka podaci i odnosi među njima su organizirani u tablice. Tablica je zbirka zapisa, a svaki zapis u tablici sadrži ista polja. Sadržaj u tablici može biti trajno spremljen za buduću uporabu (R. Elmasri, S. Navathe. 2010 str. 59).

Relacijska baza podataka je postala vrlo popularna iz dva razloga. Prvo, relacijska baza podataka može se koristiti s malo ili bez treninga. Drugo, zapisi se mogu mijenjati bez redefiniranja cijele strukture. Nedostatak korištenja relacijskih baza podataka odnosi se na činjenicu da potraga za podacima može potrajati više vremena za razliku od slučaja kada se koriste druge metode.

Prva komercijalna implementacija relacijskog modela počela je ranih 1980-ih, kao što su SQL / DS sustava na MVS operativnom sustavu od strane IBM-a i Oracle sustava za upravljanje bazama podataka. Od tada, model je implementiran u velikom broju komercijalnih sustava. Trenutni popularni relacijski sustavi za upravljanje bazama podataka uključuju DB2 i Informix Dynamic Server (od IBM-a), Oracle i RDB (od Oracle), Sybase sustav za upravljanje bazama podataka (od Sybase) i SQLServer i Access (od Microsoft). Osim toga, nekoliko *open source* sustava, kao što su MySQL i PostgreSQL su isto na raspolaganju. (R. Elmasri, S. Navathe. 2010 str. 59)



Slika 3: Primjer relacijskog modela

## Relacijski model



Izvor: Vlastita izrada 19. 07. 2015.

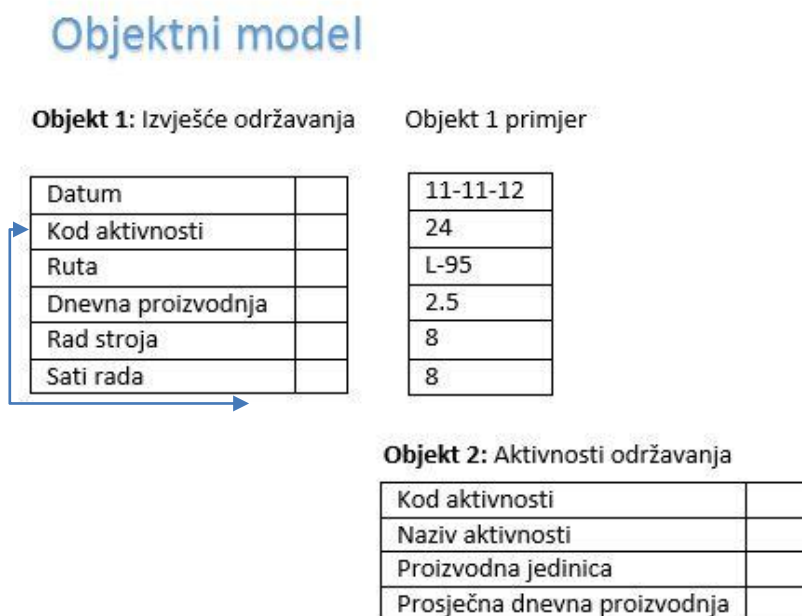
### 2.1.4. Objektni model

Objektne baze podataka (također objektno orijentirani sustav upravljanja bazom podataka) su sustavi za upravljanje bazom podataka u kojoj su podaci predstavljeni u obliku objekata kao što se koristi u objektno-orijentiranom programiranju. Objektne baze podataka se razlikuju od relacijskih baza podataka koje su orijentirane tablicama. Objektne baze podataka su uzete u obzir od ranih 1980-ih.

Objektne baze podataka pohranjuju podatke i složene odnose između podataka izravno bez mapiranja relacijskih redaka i stupaca, a to ih čini prikladnim za primjene u složenim bazama podataka. Objekti imaju puno odnosa i pristupa im se korištenjem pokazivača. Pokazivači su povezani s objektima da bi uspostavili odnos. Još jedna prednost objektnog modela je da se može programirati s malim proceduralnim razlikama bez utjecaja na cijeli sustav. To je vrlo korisno za one organizacije koje imaju veze među podacima koji nisu potpuno jasni ili moraju promijeniti te

odnose kako bi zadovoljili nove poslovne zahtjeve (A.Silberschatz, H.F. Korth, S.Sudarshan. 2011. Str 945).

Slika 4: Primjer objektnog modela



Izvor: Vlastita izrada 19. 07. 2015.

## 2.2. Tipovi baza podataka

Sustavi upravljanja bazama podataka mogu podržati različite vrste baza podataka. Baze podataka mogu se razvrstati prema broju korisnika, lokaciji baze podataka, te očekivanju vrsti i opsegu korištenja. Broj korisnika utvrđuje je li baza podataka klasificirana kao za jednog korisnika ili više korisnika. Tipovi baza podataka su (<http://databasemanagement.wikia.com> 20. 05. 2015.):

**Single-user baze podataka** - podržava samo jednog korisnika na vrijeme. Ako korisnik A koristi bazu podataka, korisnici B ili C moraju čekati dok A ne završi.

**Desktop baze podataka** - jedan korisnik baze podataka koji radi na osobnom računalu.

**Višekorisnička baza podataka** - Podržava više korisnika u isto vrijeme. Kad višekorisnička baza podataka podržava relativno mali broj korisnika (manji od 50).

**Baza podataka radne grupe** - Podržava mali broj korisnika (obično manje od 50) ili određeni odjel unutar organizacije.

**Poslovna baza podataka** – Koristi je cijela organizacija (obično stotine korisnika) u mnogim odjelima.

**Centralizirane baze podataka** - Podržava podatke koji se nalaze na jednom mjestu.

**Distribuirana baza podataka** - Podržava podatke raspoređene u nekoliko različitih mjesta.

**Operativna baza podataka** - Dizajnirana za podršku poslovanja tvrtke iz dana u dan. Također se spominje kao transakcijska ili proizvodna baza podataka.

**Skladišta podataka** - Usredotočena na spremanje podataka koji se koriste za generiranje informacije potrebne za taktičke ili strateške odluke.

### 3. Otvoreni kod (eng. "*Open source*")

Softver koji možete kupiti ili preuzeti sa interneta dolazi samo u kompiliranoj verziji spremnoj za korištenje. Kompiliran znači da je stvarni programski kod koji je programer stvorio, poznat kao izvorni kod, pokrenut preko posebnog programa pod nazivom *prevodilac* ili *jezični prevoditelj* koji prevodi izvorni kod u oblik koji računalo može razumjeti. Izuzetno je teško mijenjati kompiliranu verziju većine aplikacija i gotovo nemoguće je vidjeti točno kako je programer stvorio različite dijelove programa. Većina komercijalnih proizvođača softvera vidi to kao prednost koja sprečava druge tvrtke da kopiraju kod i koriste ga u konkurentskim proizvodima. Ta činjenica im također daje kontrolu nad kvalitetom i značajkama koje se nalaze u pojedinom proizvodu.

*Open source* softver je na suprotnom kraju spektra. Izvorni kod je uključen sa kompiliranom verzijom i izmjene ili prilagodbe se zapravo potiču. Softver programeri koji podržavaju *open source* koncept vjeruju da time što dopuštaju svakome tko je zainteresiran za izmjenu izvornog koda, program će biti korisniji i bez grešaka na dugi rok.

Pokret slobodnog softvera pokrenut je 1983, ali su postojali raniji projekti koji odgovaraju suvremenoj definiciji slobodnog softvera, odnosno, softver koji svi korisnici mogu slobodno koristiti, proučavati, mijenjati i redistribuirati ("slobodan kao sloboda"). Raniji projekti davali su te slobode bilo iz praktičnih razloga ili društvenih razloga, ali nisu bili dio organiziranog pokreta širenja prakse ili filozofije.

Pokret je pokrenuo Richard Stallman kao reakciju na rastući trend programera koji blokiraju te slobode objavljivanjem kompilirane verzije softvera, a ne izvorni kod koji se da mijenjati. Stallman tvrdi da je to društveni imperativ za sav distribuirani softver, a ne tehnički izbor koji se jednostavno dogodi, te da ima praktičnu vrijednost u nekim kontekstima (<http://www.gnu.org/> 12. 09. 2015.). Godine 1998, ljudi koji su se zalagali za besplatni softver, složili su se da je to društveni imperativ i počeli su koristiti termin "*open source* softver" za software predstavljajući ga kao da ima tehničke prednosti.

Da bi se softver uzeo u obzir kao *open source* softver u industriji razvoja softvera, određeni kriteriji moraju biti ispunjeni (<http://otvorenikod.weebly.com> 15. 08. 2015.):

**Slobodna redistribucija** - licenca ne ograničava niti jednu stranu u prodaji ili poklanjanju softvera kao sastavnice zajedničke distribucije softvera koja sadrži programe iz nekoliko različitih izvora. Licenca ne zahtijeva naknadu za autorsko pravo ili druge naknade kod navedenog oblika prodaje.

**Izvorni kod** - program mora uključivati izvorni kod te mora omogućiti distribuciju u izvornom kodu, kao i u kompiliranom obliku. Ako se neki oblik proizvoda ne distribuira s izvornim kodom, mora postojati jasno naznačen način pribavljanja izvornog koda za iznos koji ne premašuje razumne troškove reprodukcije, po mogućnosti preuzimanjem preko interneta bez naplate. Izvorni kod mora biti oblik kojemu programer daje prednost pri modifikaciji programa. Nije dopušteno namjerno skrivanje izvornog koda. Nisu dopušteni prijelazni oblici kao što su izlaz pred procesora ili prevoditelja.

**Izvedeni oblici programa** - licenca mora dopuštati modifikacije i izvedene oblike programa te mora dopuštati njihovu distribuciju pod istim uvjetima navedenima u licenci originalnog softvera.

**Integritet autorskog izvornog koda** - licenca može ograničavati distribuciju izvornog koda u modificiranom obliku isključivo ukoliko licenca dopušta distribuciju dodatnih “zakrpa” uz izvorni kod u svrhu modifikacije programa kod kompiliranja. Licenca mora izričito dopuštati distribuciju softvera kompiliranog iz modificiranog izvornog koda. Licenca može sadržavati odredbe prema kojima takvi modificirani programi moraju nositi naziv ili broj inačice koji se razlikuje od naziva ili broja originalnog softvera.

**Nije dopuštena diskriminacija osoba ili skupina** - licenca ne smije diskriminirati pojedinačne osobe ili skupinu osoba.

**Nije dopuštena diskriminacija djelatnosti** - licenca ne smije ograničavati korištenje programa za određenu djelatnost. Primjerice, ne može ograničavati korištenje programa u poslovnom okruženju, ili za genetska istraživanja.

**Distribucija licence** - prava vezana uz program moraju vrijediti za sve korisnike kojima se program dalje distribuira, bez izdavanja dodatne licence.

**Licenca ne smije biti ograničena na jedan proizvod** - prava koja se odnose na program ne smiju ovisiti o pripadnosti programa distribuciji određenog softvera. Ako se program odvoji od te distribucije i koristi ili distribuira u skladu s uvjetima licence programa, sve stranke kojima se program dalje distribuira trebale bi imati prava istovjetna pravima važećima za distribuciju originalnog softvera.

**Licenca ne smije ograničavati drugi softver** - licenca ne smije primjenjivati ograničenja na drugi softver koji se distribuira uz licencirani softver. Primjerice, licenca ne smije zahtijevati da svi drugi programi koji se distribuiraju na istom mediju moraju biti otvorenog koda.

**Licenca mora biti neutralna u odnosu na tehnologiju** - niti jedna odredba licence se ne može temeljiti na pojedinoj tehnologiji ili obliku sučelja.

Godine 1991, Linus Torvalds, student na Sveučilištu u Helsinkiju u Finskoj je razvio novi operativni sustav temeljen na Minix-u, izveden od Unix-a, koji je nazvao Linux. Torvalds je objavio verziju 0,02 Linuxa pod GNU *General Public License*, koja pruža dobru zakonsku definiciju *open source* softvera. Ljudi diljem svijeta preuzeli su Linux i počeli raditi s njim. Mnogi od tih korisnika su bili programeri koji su na svoju ruku napravili izmjene u izvornom kodu koji je Torvalds uključio. Tijekom iduće tri godine, Torvalds je primio ove modificirane verzije od ostalih programera i ugradio mnoge promjene u osnovnu verziju i objavio 1994. Linux verziju 1.0 (<http://www.wired.com/> 12. 09. 2015.).

Najčešća briga krajnjim korisnicima koji žele koristiti *open source* softver je nepostojanje jamstva i tehničke podrške. Licenca *open source* softvera potiče izmjenu i prilagodbu za što je

gotovo nemoguće imati tehničku podršku. Iz tog razloga Red Hat Software, osnovan 1994, stvorio je "Official Red Hat Linux", te je u stanju prodati ovaj inače "besplatan" softver. Glavna vrijednost koju Red Hat dodaje paketu je jamstvo i tehnička podrška. Za većinu tvrtki osiguranje tehničke podrške ključni je faktor u odluci za kupnju Linux umjesto da ga jednostavno preuzmu besplatno (u Hrvatskoj jedan od vodećih lanaca supermarketa koristi Red Hat verziju Linuxa na svojim blagajnama). Osim Red Hat, postoji i nekoliko drugih tvrtki koje su pakirale Linux, obično s dodatnim softverom, za daljnju prodaju.

Osim Linuxa, Mozilla (jezgra preglednika Firefox), Apache (web server), PERL (web jezik za skriptiranje) i PNG (grafički format) su sve primjeri vrlo popularnog softvera koji se temelji na *open source*-u.

### **3.1. Dostupnost i prednosti otvorenog koda**

Od 2008, 85% poduzeća se koristi *open source* softverom, prema Gartneru. (www.cnet.com 19. 08. 2014.). *Open source* programi su sada dostupni za svaku vrstu poslovnog softvera - od baza podataka, web poslužitelja, web preglednika i uredskih aplikacija do softvera za nadgledanje mreže i sigurnosnog softvera. Naravno, u svim tim kategorijama postoje alternative u komercijalnom softveru. Kvaliteta *open source* softvera isto ispunjava ili premašuje očekivanja 92% ispitanika u nedavnom istraživanju koje je provela tvrtka Forrester Research. U međuvremenu, 87% ispitanika izjavilo je da *open source* softver isporučuje uštedu kojoj su se nadali. Poduzeća smatraju da su dobili veliku vrijednost i željeni povrat ulaganja od *open source* softvera (www.cnet.com 19. 08. 2014.).

Prednosti *Open Source* baza podataka (www.donos.hr 10. 09. 2014.):

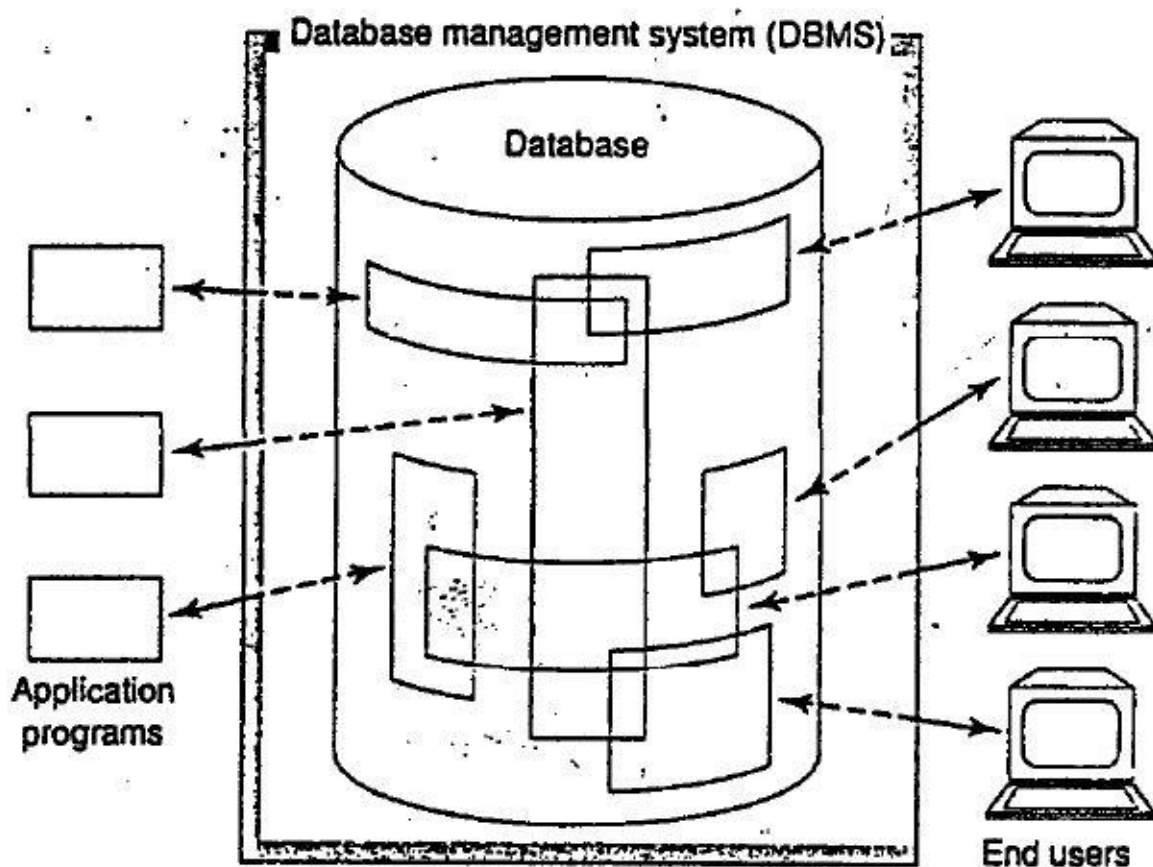
1. Sigurnost, jer više očiju brže uoči grešku
2. Kvalitetu, jer ih testiraju tisuće korisnika
3. Lakše se prilagođavaju potrebama svakog korisnika
4. Korisnik nije vezan za jednog programera ili kompaniju
5. Ne postoji potreba za stalnim plaćanjem nove verzije programa,
6. Jamstvo kvalitete daju ostali korisnici, a ne proizvođač
7. Svi problemi su vjerojatno već otkriveni i dokumentirani
8. Višestruko niža cijena jer nema plaćanja razvoja proizvoda,
9. Sva rješenja su provjerena od strane tisuća korisnika,
10. Za svaku nadogradnju vjerojatno već postoji gotovo rješenje.



## 4. Sustavi upravljanja bazama podataka

Sustavi upravljanja bazom podataka su posebno dizajnirani programi koji u interakciji s korisnikom, drugim aplikacijama, i samom bazom dohvaćaju podatke i analiziraju ih. Sustav upravljanja bazom podataka opće namjene je softverski sustav dizajniran kako bi se omogućilo definiranje, stvaranje, upiti, ažuriranje i administracija baza podataka.

Slika 5: Jednostavan prikaz sustava za upravljanje bazama podataka



Izvor: R. C.J. Date: An Introduction to Database Systems, 8th Edition. Addison-Wesley, Reading MA, 2003.. Str. 7.

Sustavi upravljanja bazama podataka mogu biti na temelju nekoliko različitih modela podataka: hijerarhijski, mrežni, relacijski ili objektni. Višestruki korisnici mogu pristupati različitim podacima. Isto omogućuje pristup iz više mjesta i sa više aplikacija.

Sustav upravljanja bazama podataka nudi sljedeće prednosti (<http://databasemanagement.wikia.com> 20. 05. 2015.):

- **Sigurnost podataka** - Pruža pristup podacima samo ovjerenim korisnicima. ID i lozinka koje služe za zaštitu podataka. Također, dijelovi baze podataka mogu se dati određenim korisnicima koje drugi možda nemaju.
- **Neovisnost podataka** - Pruža sigurnost ili izolaciju između stvarnih podataka i aplikacija koje ga koriste. Ti podatci se mogu promijeniti bez potrebe da se mijenja i program. To osigurava aplikacijsko programsko sučelje, kao što je SQL.
- **Integritet podataka** - Određuje vrstu podataka koji se koriste u poljima. Sustav za upravljanje bazama podataka može upravljati odnose među vrstama podataka.
- **Mehanizmi zaključavanja** - Važno za kontrolu transakcija i valentnost podataka. Sustav će pružiti opciju povratka podataka ako dođe do oštećenja.
- **Programsko balansiranje** - Uravnotežuje opterećenje rada unutar CPU-a i također unutar mrežne infrastrukture.
- **Evidencija promjena baze** - Evidencija o objektima baze podataka kada se mijenjanju. Ona bilježi svaku izmjenu podataka ukoliko nije drugačije navedeno. Evidencija se može koristiti da se promjene napravljene na bazi podataka vrate na staro stanje.

#### 4.1. Primjeri sustava za upravljanje bazama podataka

Na različite načine, sustavi su postali poput robe. Proizvodi se razlikuju po cijeni, performansama, jednostavnosti administracije baze podataka i funkcionalnosti.

Postoji i veliki izbor sustava za upravljanje bazom podataka što uključuje i pakirani i *open source*. Glavni dobavljači su Fujitsu, Hewlett-Packard, Hitachi, IBM, Microsoft, NCR Teradata, Oracle, Progress, SAS Institut i Sybase.

#### **4.1.1. MySQL**

Jedan od najutjecajnijih i široko raširenih *open source* sustava za upravljanje bazama podataka koji manipulira velike baze podataka, a može mu se pristupiti preko weba je MySQL. O njemu detaljnije u poglavlju 5.

#### **4.1.2. Microsoft SQL Server**

Microsoft SQL Server dizajniran je za stvaranje web, poslovne i desktop baze podataka, te se koristi s različitim ciljevima i na različitim razinama. MS SQL Server omogućuje pohranu velikih količina podataka koji obrađuju komponente kao što su video, fotografije, brojevi, tekst i ostalo. Microsoft SQL Server je razvijen za upravljanje terabajtima podataka u usporedbi s Microsoft Access koji se može nositi sa samo 2 gigabajta podataka (<https://support.office.com> 31. 08. 2015.).

#### **4.1.3. Microsoft Office Access**

Microsoft Office Access kombinira alate za razvoj softvera i relacijski Microsoft Jet Database Engine s jednostavnim korisničkim sučeljem za korištenje. MS Access je idealno rješenje za početnike zbog mogućnosti stvaranja struktura i odnosa u bazi. MS Access mogu koristiti mali poduzetnici, ogranci velikih korporacija i od programera koji iz razonode stvaraju posebne prilagođene sustave za manipulaciju stvaranja i upravljanja podacima. Microsoft Access baze podataka svakako su postale vrlo popularne upravo zbog svoje ključne prednosti za izgradnju softverskih sustava za upravljanje cijelog niz usluga vrlo brzo.

#### 4.1.4. Microsoft Visual FoxPro

Microsoft Visual FoxPro, skraćeno kao VFP, usko je integriran s vlastitom relacijskom bazom podataka koja proširuje FoxPro te je xbase u mogućnosti održavati SQL upite i upravljanje podacima. Za razliku od većine sustava za upravljanje bazama podataka, Visual FoxPro je dinamičan programski jezik koji ne zahtijeva korištenje dodatnog univerzalnog programskog okruženja, te se može koristiti za pisanje ne samo konvencionalnih aplikacija, nego i web aplikacijskih programa i middleware-a. Od 2007 Microsoft je odustao od ažuriranja FoxPro stoga se odnedavno podatci obično pretvaraju u jednom smjeru iz FoxPro-a u ostale sustave za upravljanje bazama podataka.

#### 4.1.5. PostgreSQL

Relacijski sustav koji mnogi programeri web aplikacija koriste radije kao back-end komponentu za upravljanje podacima. To se prvenstveno koristi od strane mnogih istaknutih organizacija koji ga primjenjuju za kritične aplikacije ili aplikacije širokih primjena. Domene .info i .org ga koriste kao svoje primarno spremište podataka. Isto koriste mnoge financijske institucije i velike tvrtke. Iako PostgreSQL nije pravi izbor za svaki projekt, njegove napredne značajke i ključne prednosti kao što je *open source*, vrlo niski troškovi implementacije i laka administracije čine ga odličnim izborom za one koji koriste web baze podataka.

#### 4.1.6. SQLite

SQLite je mala C knjižnica koja obavlja samostalnu, ugradbenu, SQL bazu podataka bez konfiguracije. Ova baza podataka pruža brz i jednostavan način za rad s ravnim bazama podataka. Za razliku od poslužiteljskog sustava za upravljanje bazama podataka, SQLite je sastavni dio programa. SQLite je idealno rješenje u situacijama u kojima su implementacija, održavanje i jednostavnost administracije važnije od bezbrojnih složenih značajki koje poslovni sustavi za upravljanje bazama podataka pružaju. SQLite je pravi izbor kao baza podataka za male

i srednje web stranice. SQLite često se koristi kao zamjena za poslovne sustave za upravljanje bazama podataka u smislu svrhe prikazivanja ili za testiranja.

#### **4.1.7. Firebird**

Firebird je potpuno opremljen i snažan relacijski sustav za upravljanje bazama podataka unatoč veličini instalacijske datoteke. Može manipulirati bazama podataka od samo nekoliko kilobajta do par gigabajta s dobrim izvršenjem i s gotovo besplatnom podrškom. Jednako je pogodan za komercijalne i *open source* aplikacije. Firebirdova sposobnost da radi u potpunosti automatiziran bez administrativne nužnosti ga čini idealnim za distribuciju baze podataka i aplikacija. On nudi gotovo svaku značajku dostupnu u bazama podataka visokih performansi bez značajnijeg utjecaja na postignuća i ima mogućnosti da radi dobro s *open source* alatima kao što su PHP i Perl.

#### **4.1.8. Oracle**

Oracle je jedan od vodećih poslovnih SQL relacijskih sustava za upravljanje bazama podataka. Dostupan je u različitim konfiguracijama od malih osobnih verzija do jakih poslovnih verzija. Oracle nudi puno mogućnosti i funkcionalnosti za rješavanje kompliciranih problema, zato se ističe u Fortune 100, a koristi se u srednjim i velikim poduzećima i skladištima. Ovaj moćan sustav zahtijeva puno znanja i vještine za obradu velikih količina podataka.

## 5. MySQL

Slika 6: MySQL logo



Izvor: <https://www.mysql.com> 14. 05. 2015.

MySQL je *open source* relacijski sustav za upravljanje bazama podataka. Trenutno je instaliran na više od šest milijuna sustava i ima prosječnu stopu preuzimanje od 40.000 na dan. MySQL radi kao poslužitelj koji pruža više-korisnički pristup brojnim bazama podataka. Ime je dobio po kćeri od suosnivača Montya Wideniusa, koja se zove My (<http://databasemanagement.wikia.com/> 12. 09. 2015).

MySQL je počeo s namjerom korištenja mSQL sustava baze podataka. Međutim, nakon testiranja, programeri su došli do zaključka da mSQL nije bio dovoljno brz i dovoljno fleksibilan za njihove potrebe. To je rezultiralo novim SQL sučeljem na njihove baze podataka, ali s gotovo istim sučeljem za programiranje aplikacija kao mSQL. Ovo Aplikacijsko programsko sučelje je dizajnirano kako bi se omogućilo kodu koji je napisan za korištenje s mSQL da bude prenesen jednostavno za korištenje s MySQL-om.

MySQL softwer nudi vrlo brz, višekorisnički i robustan SQL poslužitelj baze podataka. MySQL je namijenjen kritičnim, teško opterećenim proizvodnim sustavima, kao i ugrađivanju u masovno dijeljen softver.

MySQL ima dvojno licenciranje što znači da možete koristiti MySQL softver besplatno pod GNU općom javnom licencom. Također možete kupiti komercijalnu MySQL licencu ako ne želite biti vezani uvjetima opće javne licence. (M. Widenius, D. Axmark. 2002, Str. 2)

Isto se pridržava ANSI SQL 2003 specifikacije. ANSI je Američki nacionalni institut za standarde koji postavlja specifikacije za kod programskog jezika upravljanja strukturiranim upitima (SQL). Osim SQL, MySQL također ima sposobnost učiniti *if, else* (hrv. ako, drugo) izjave i *while/looping* (hrv. dok/ponavljanje) izjave. MySQL također podržava više programskih jezika za izgradnju prednjeg kraja za povezivanje s bazom podataka. Ti jezici su PHP, Perl, Java, .Net.

## **5.1. Prednosti MySQL-a**

Bilo da ste web programer ili administrator mreže s interesom za izgradnju aplikacija baze podataka MySQL je jednostavan za korištenje, ali iznimno moćan, siguran i skalabilan. Zbog svoje male veličine i brzine, to je idealna baza podataka za web stranice. Njegove prednosti uključuju slijedeće: brzina, pouzdanost, rad na različitim platformama, podržava veliku količinu programskih jezika, ODBC, besplatno ili povoljno licenciranje, povoljna komercijalna podrška, dostupnost izvornog koda, mala zahtjevnost za sistemskim resursima.

### **5.1.1. Brzina**

MySQL kod je pisan u cijelosti od nule sa temeljnim ciljem postizanja što boljih izvedbi. MySQL je u stvari nastao zbog frustracije Montya Wideniusa sporošću drugih programa pa je odlučio napisati MySQL. Mnogo novih korisnika je dovelo u pitanje sposobnost MySQL-a da zadovolji njihove osnovne potrebe. Kako bi se pokazalo što MySQL sve može napravljen je

*benchmark* program koji je testirao sve njegove mogućnosti. Korisnici su bili oduševljeni rezultatima, te su se sve više odlučivali na MySQL. MySQL je svakako dovoljno brz da zadovolji sve potrebe, pa čak i brži od onoga što je potrebno. Kao i svaki drugi program MySQL može biti spor u nekim situacijama, a do toga može doći ako se ne obraća pažnja na točnost komandi kada šaljemo upite. Navedenu situaciju možemo izbjeći shvaćanjem na koji način radi server.

### **5.1.2. Pouzdanost**

Nakon osnovnog podešavanja MySQL može raditi danima i mjesecima bez intervencije korisnika. Povremeno dolazi do otkrivanja grešaka unutar programa ili problema, što je uobičajeno za svaki server baza podataka. Ako MySQL server padne (što je rijetkost) vrlo se lako vraća u prethodno stanje. Prednosti MySQL-a privukle su pažnju nekoliko ozbiljnih korisnika koji su odlučili sa komercijalnih programa preći na MySQL. Neke od njih su: NASA, Yahoo! Finance, Motorola, Novell i mnogi drugi. Stručni tim je fokusiran na pouzdanost i stabilnost. Događalo se da su nove verzije MySQL-a odgađane zbog manjih grešaka u programu koje su bile otkrivene, ali ne i otklonjene. Razvoj bi se obustavio u slučaju otkrivanja veće greške i počinjalo bi se od nule, a svi korisnici bi primili javne isprike.

### **5.1.3. Rad na različitim platformama**

MySQL ima mogućnost rada na različitim operacijskim sustavima i arhitekturama. Operacijski sustavi koji se učestalo koriste su Linux, Solaris, FreeBSD i Windows. Također, MySQL radi i na Irix, Tru64, OpenBSD, NetBSD, Max OS X, AIX i SCO.

### **5.1.4. Podrška veliku količinu programskih jezika**

Prilikom rada na razvoju sustava za upravljanje bazama podataka, jedna od najvećih briga je da li će se sustav moći povezati sa serverom baze podataka koristeći određeni programski jezik. MySQL je i na ovom području bolji od konkurencije pošto programeri sa MySQL-om mogu komunicirati koristeći PHP, Java, Ruby, TCL, Python, Perl, Eiffel i C/C++.



### **5.1.5. ODBC**

Pored podrške za programske jezike, MySQL također ima podršku za ODBC. Ovo daje programeru mogućnost da kreira aplikacije koristeći *Open Database Coonectivity* (ODBC) standard. Podrška za ODBC pruža mogućnost korištenja MySQL-a sa drugim menadžerima baza podataka koji imaju ODBC. Neke od tih aplikacija su Microsoft Excel, Crystal Reports, Microsoft Access i mnogi drugi. Zahvaljujući ODBC-u MySQL može se koristiti u kombinaciji sa ColdFusion-om, BorlandBuilder-om, Visual Basic-om, Delphi-em i mnogim drugim razvojnim alatima i okruženjima.

### **5.1.6. Besplatno ili povoljno licenciranje**

MySQL se distributira pod odredbama GLP-a (*General Public License*) koje su propisane iz FSF-a (*Free Software Foundation*). Prema navedenoj licenci korisnik može koristiti software potpuno besplatno kako za komercijalne svrhe tako i za nekomercijalne svrhe. MySQL se distribuira korisnicima potpuno besplatno, osim kada se korisnik odlučuje za MySQL Enterprise za koji se plaća dodatna licenca. Licenciranje se može zatražiti od MySQL-a po povoljnim uvjetima u slučaju da je potrebna zbog softvera koji ne dozvoljava korištenje GPL proizvoda ili se korisnik odluči na Enterprise. Licenca za jedan server iznosi 200\$, ali cijena pada povećanjem broja licenci. Licence ne ograničavaju broj korisnika i izdaju se po serveru.

### **5.1.7. Povoljna komercijalna podrška**

Za korisnike koji žele koristiti MySQL za zahtjevnije poslove vrlo je važna profesionalna korisnička podrška. U slučaju da i ne dođe do problema korisnik će se osjećati sigurnije znajući da postoji netko kome se može obratiti ako dođe do problema. MySQL AB pruža širok raspon podrške po vrlo razumnim cijenama uključujući svakodnevnu telefonsku podršku. Na sva pitanja odgovaraju programeri pa se odgovori na tražena pitanja dobivaju puno brže.

### **5.1.8. Dostupnost izvornog koda**

Velika prednost je mogućnost pristupa MySQL-ovom izvornom kodu, pogotovo za korisnike koji imaju stručne C/C++ programere. To omogućuje izmjene, popravke, poboljšanja bez čekanja da izdavač softvera to učini. Prednost otvorenog koda je ta što je manja vjerojatnost da će doći do grešaka u kodu ili do lošeg koda, jer otvoreni kod mogu svi vidjeti.

### **5.1.9. Mala zahtjevnost za sistemskim resursima**

MySQL će maksimalno iskoristiti resurse koji mu se daju. Sa što više resursa izvedbe će biti bolje, ali minimalni resursi neće biti problem kao što znaju biti na nekim drugim serverima baza podataka. MySQL sasvim korektno radi i na konfiguracijama tipa Cpu od 300+ mhz i 64Mb RAM-a, ali bilo je iskustava i sa konfiguracijama koje su imale i slabije specifikacije od gore navedenih što ih nije omelo da rade.

## **5.2. Nedostaci MySQL-a**

MySQL ima određene nedostatke. To se obično ne otkriva u preliminarnim razinama uobičajenih scenarija. Nedostatak testiranja i složeno rukovanje izvornim kodom uzrokuje probleme administratorima MySQL baza. O tim problemima detaljnije u slijedećim poglavljima.

### **5.2.1. Nedostatak nekih mogućnosti SQL-a**

Predstavlja probleme u povezivanju MySQL-a i postojećih aplikacija. U slučaju da aplikacija koristi neku osobinu koju MySQL nema, prije povezivanja dijelovi aplikacije se moraju reprogramirati. Stručnjaci iz svijeta informatičke tehnologije smatraju da se neke opcije koje MySQL nema moraju dodati. Većina programera je navikla koristiti te opcije i one predstavljaju osnovni dio njihovog programiranja. Taj nedostatak ponekad zna odvratiti potencijalne korisnike. Reakcija korisnika može se usporediti sa čovjekom koji je nakon dužeg vremena vožnje auta sa automatskim mjenjačem prešao na ručni.

Iskusni MySQL korisnici su navikli na nedostatke pa čak i uživaju u pronalaženju načina da zaobiđu ove nedostatke. Pronašli su načine da najbolje iskoriste MySQL bez obzira na nedostatke, te podržavaju rasprostranjivanje MySQL-a. Isto kao što će vozači sa ručnim mjenjačem postići veću brzinu od onih sa automatskim. Zbog ovih nedostataka podijeljeno je mišljenje o upotrebi MySQL-a. Tu se ponovno postavlja pitanje da li je MySQL dovoljno dobar.

Kreativni i iskusni programeri jednostavno koriste MySQL za izgradnju baze podataka. Do problema dolazi kada je programer vezan i naučen na neki drugi program. Njegova predanost u radu sa MySQL-om nebi bila kao što je sa nekim drugim programom na koji su naučili. Zbog ovoga razloga nebi trebalo forsirati radnike da koriste MySQL, ali to bi moglo biti i korisno da za vrlo kratko vremensko razdoblje nađu rješenja. MySQL radi na tome da u sljedećih nekoliko godina otklone sve nedostatke.

### **5.2.2. Nedostatak ozbiljnih testiranja na određenim platformama**

MySQL svoje proizvode uvijek testira kroz nekoliko vrlo zahtjevnih testova. Sljedeće testiranje javlja se kada novu verziju preuzme 20000 korisnika dnevno (Alexander Sasha Pachev, 2003, str. 5). Tijekom tog procesa mogu se otkriti razne greške i problemi koji se nisu mogli ustanoviti prije. MySQL AB vrlo brzo otkloni pronađene greške, ali to puno ovisi o *feedbacku* korisnika.

Kao rezultat može se uočiti da stabilnost MySQL-a na određenoj platformi zavisi od broja korisnika. Što je više korisnika manja je šansa da će greške ostati neotkrivene. Iako se MySQL trudi da ne stavlja nijednu platformu ispred ostalih, razlika u broju korisnika dovodi do toga da neke platforme rade bolje od drugih. Prema proračunima 40% korisnika koristi MySQL na x86 Linux platformi, 25% korisnika ga koristi na Windows platformi, 15% na FreeBSD, 15% na SPARC Solaris, te 5% na ostalim platformama (Alexander Sasha Pachev, 2003, str. 5). Ovih 5% će uvijek zaostajati po kvaliteti, ali MySQL sasvim dobro radi i na tim platformama, što dolazi kao posljedica dobro napisanog izvornog koda. MySQL se trudi proširiti platforme na kojima vrši testove te provjerava sisteme kao što su HP-UX i AIX.

### 5.2.3. Složeno rukovanje izvornim kodom

Imati pristup izvornom kodu daje veliku fleksibilnost korisniku, ali pri radu sa kodom kojeg je netko drugi napisao prvo ga se mora razumjeti. Kod MySQL servera je prilično složen, sa time čak i iskusni C/C++ programeri imaju problema. Kod servera baze podataka optimiziran je da što je bolje moguće organizira podatke na disku, minimalizira uporabu diska, raščlanjuje upite i ubrza pristup memoriji.

Ovaj kod koliko god dobro napisan ipak je kompleksan. Najbolji programeri su na kod nailazili kao na vrlo velik izazov.

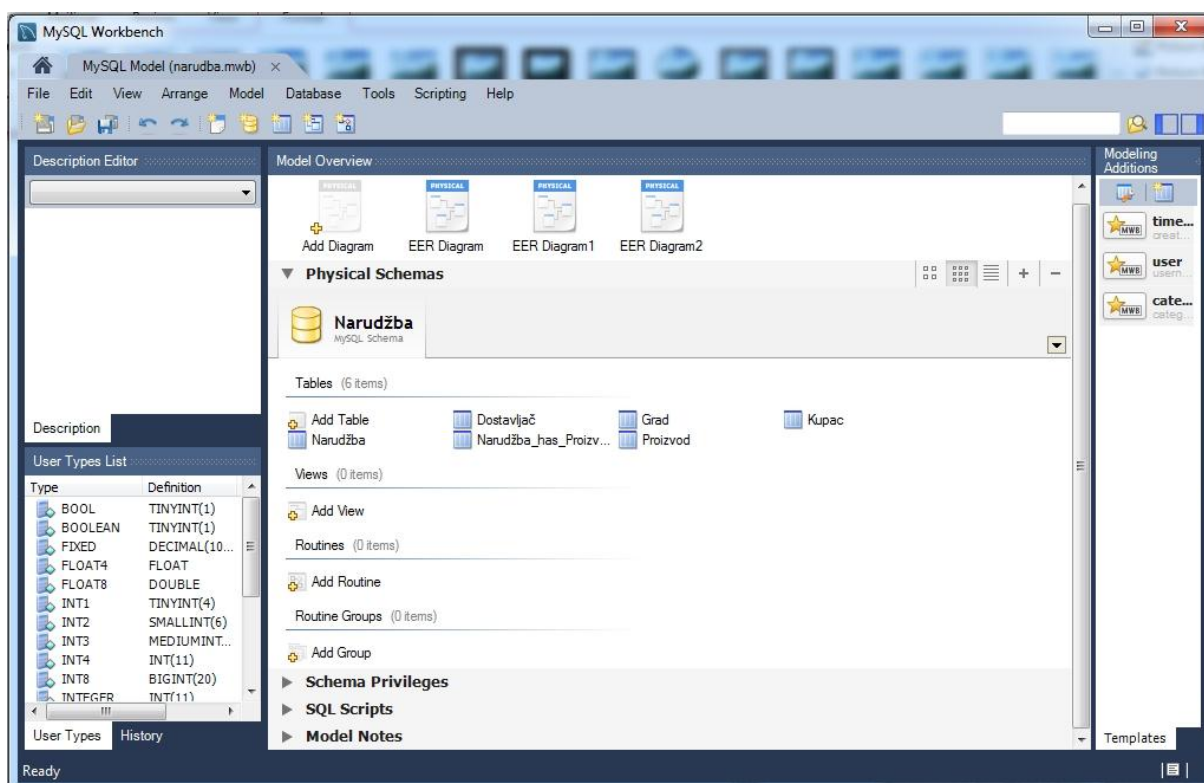
Sve ovo su samo neke od osnovnih prednosti i mana koje nudi MySQL sustav za upravljanje bazama podataka. Prednosti su velike i to je ono što privlači većinu velikih tvrtki da se odluče za MySQL baze podataka. Istovremeno, male tvrtke također nisu lišene tih značajki jer ima i opcija besplatne MySQL baze podataka.

Troškovi održavanja su prilično niski što ga čini omiljenim izborom za male tvrtke jer postoji puno besplatnih MySQL priručnika, koji omogućuju ljudima da uče i obrađuju MySQL baze podataka bez većih izazova ili problema. Ukratko, MySQL baze podataka su za sve i spremne su na sve vrste zahtjeva da zadovolje i male i velike korisnike.

## 6. Primjer MySQL baze podataka

Osnovu svake baze podataka čine tablice, u ovom slučaju (Slika 7.): kupac, narudžba, dostavljač, proizvod, grad i narudžba has proizvod. Na ovaj način uređene i definirane tablice postavljaju osnovu kojom se želi postići međuovisnost podataka unutar samih tablica na način da se izrade relacijski odnosi unutar tablica što će činiti lakše povezivanje te bolju vidljivost, a samim time će se definirati jasan cilj izrade cijele baze podataka.

Slika 7: Popis izrađenih tablica baze podataka



Izvor: Vlastita izrada 17. 05. 2015.

Dobro osmišljena baza podataka će imati unaprijed odabranu strukturu, a samim time i odluku koju vrstu podataka autor želi unositi i kako će ih u bazi najbolje posložiti, koje poslove želi informatizirati unosom podataka, odnosno koje radnje baza podataka mora činiti unosom i/ili obrađivanjem podataka te koje informacije su važne u pregledu stanja podataka, a sve u cilju izbjegavanja duplog unošenja podataka, kvalitetnog postavljanja elemenata baze, postavljanja

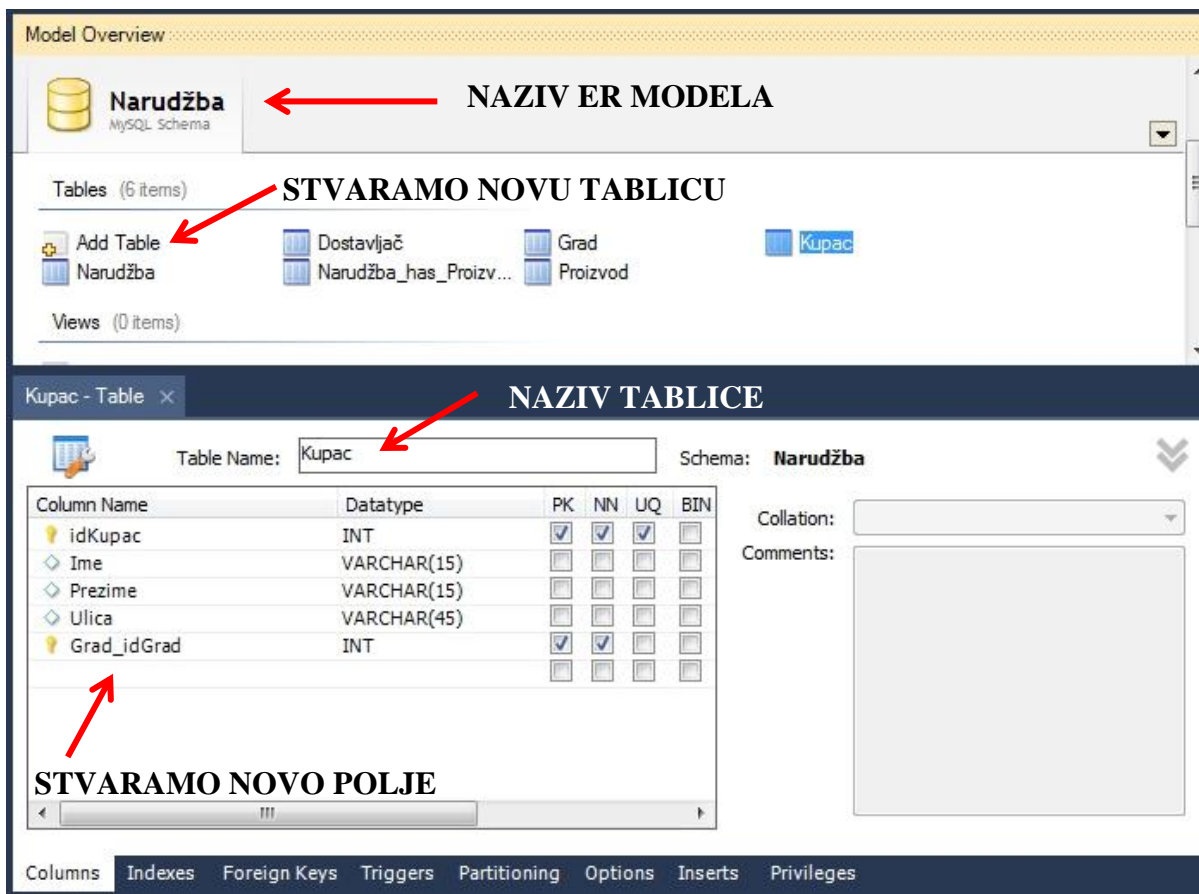
relacijskih odnosa unutar same baze te radi što svrsishodnijeg formiranja baze prema onim radnjama za koje će biti namijenjena.

U ovom slučaju primjer baze podataka jednog *webshop*-a, zbog same veličine baze i opsega raznih proizvoda kao i zbog broja različitih kategorija proizvoda kojih u jednom *webshop*-u ima bilo bi prekomjerno za potrebe ovog rada definirati i koncipirati takvu bazu koja bi pokrila sve proizvode te njihove kategorije. Stoga je je baza zamišljena na način da pokrije samo mali dio proizvoda u svrhu prikaza principa rada baze podataka kod jednog *webshop*-a, ali je ista zamišljena tako da se zaista i može primijeniti u stvarnom *webshop*-u, naravno uz neke dodatne izmjene naziva dijelova baze, dodavanja novih dijelova u bazi te dodatnim unošenjem podataka.

## **6.1. Stvaranje tablica**

U daljnjem nastavku ovoga rada najprije će se izvršiti provedba u praksi relacijskog modela podataka u MySQL baze podataka preko MySQL Workbench-a. Za početak smo instalirali MySQL Workbench te ga pokrenuli. Kroz slike ćemo postupno objasniti provedbu u praksi ER modela. ER model ćemo nazvati Narudžba. Tablice kreiramo na način na koji je prikazano na slici (Slika 8.). Klikom na Add Table pravimo novu tablicu. Sam nam program nudi naziv primarnog ključa na osnovu naziva tablice. U ovome slučaju idKupac.

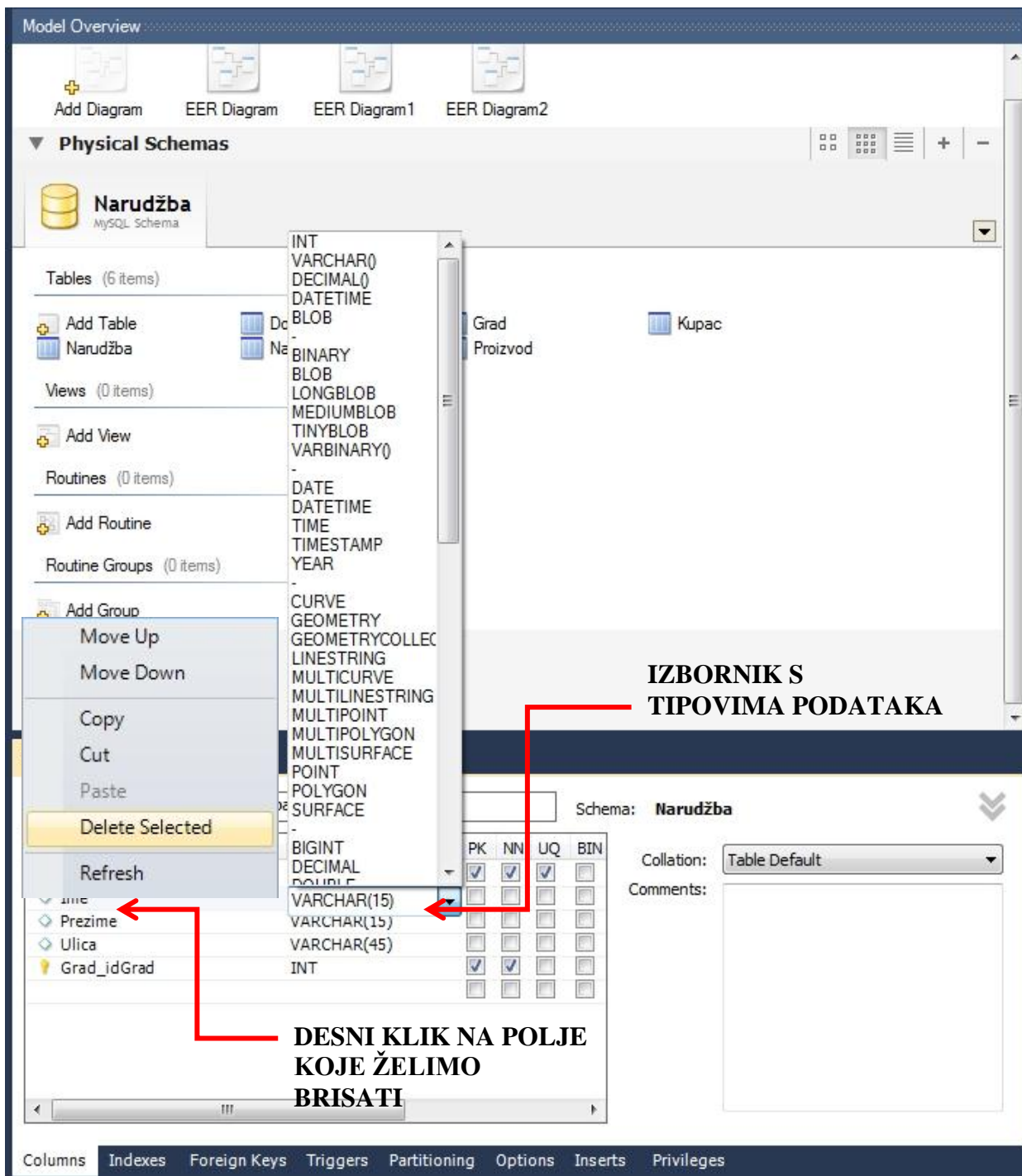
Slika 8. Kreiranje tablice



Izvor: Vlastita izrada 17. 05. 2015.

Vrstu podatka mijenjamo na način prikazan na slici ispod (Slika 9.), a također je prikazano kako izbrisati polje koje ne želimo.

Slika 9. Uređivanje i brisanje podataka unutar tablice



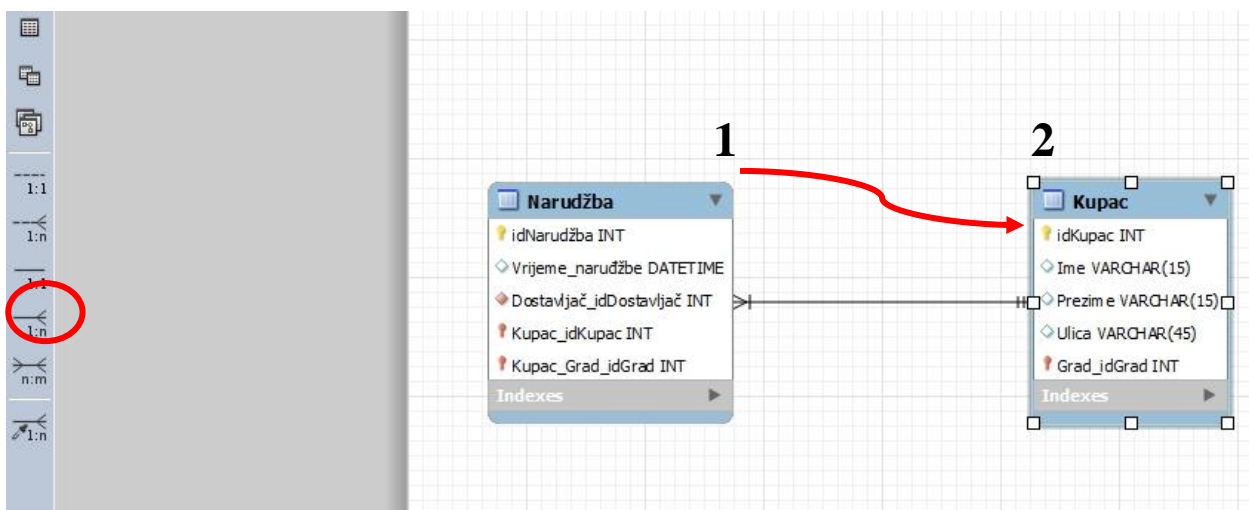
Izvor: Vlastita izrada 17. 05. 2015.



## 6.2. Kreiranje veza između tablica

Kada su tablice kreirane korištenjem već pripremljenih veza unutar samog programa jednostavno se povežu dvije tablice i strani ključevi se pojave. To je prikazano na slici ispod (Slika 10.). Klikom na vezu koja nam je potrebna u ovome slučaju (1:više), zatim kliknemo prvo na tablicu Narudžba pa tablicu Kupac i imamo vezu i postavljene strane ključeve.

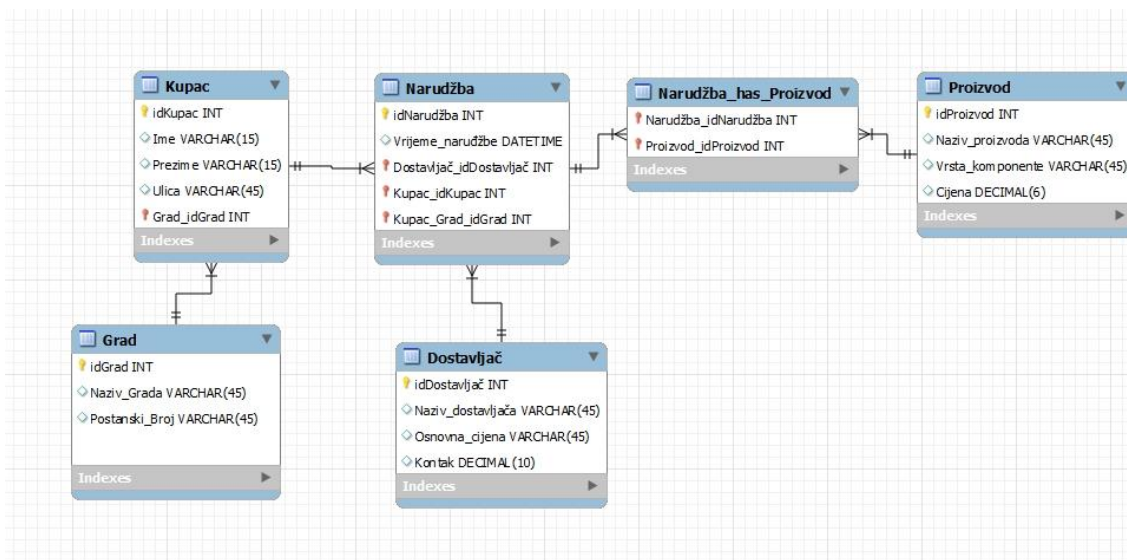
Slika 10. Povezivanje tablica



Izvor: Vlastita izrada 17. 05. 2015.

Nakon što smo povezali sve tablice dobijemo gotov ER dijagram koji je detaljan prikaz entiteta, veza i podataka radi bolje organizacije poslovanja. Kao na slici ispod (Slika 11.).

Slika 11. ER dijagram

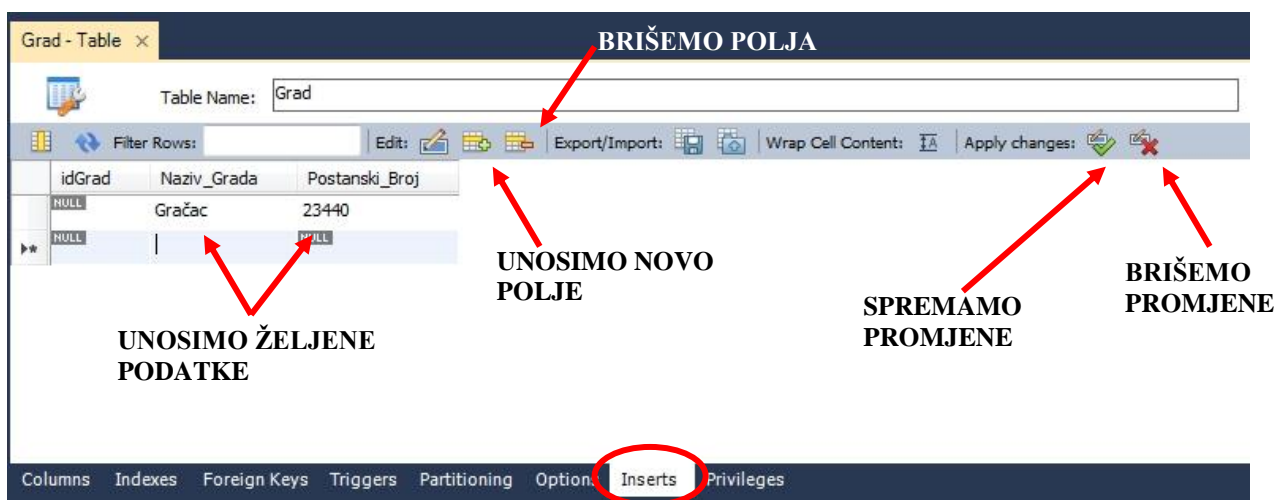


Izvor: Vlastita izrada 17. 05. 2015.

### 6.3. Unos podataka u kreirane tablice

U kreirane temeljne tablice unosimo podatke na način kako je prikazano. Na slici ispod (Slika 12.) je objašnjen postupak. Dvostrukim klikom otvorimo tablicu. Kliknemo na Insert.

Slika 12. Unos podataka u tablice

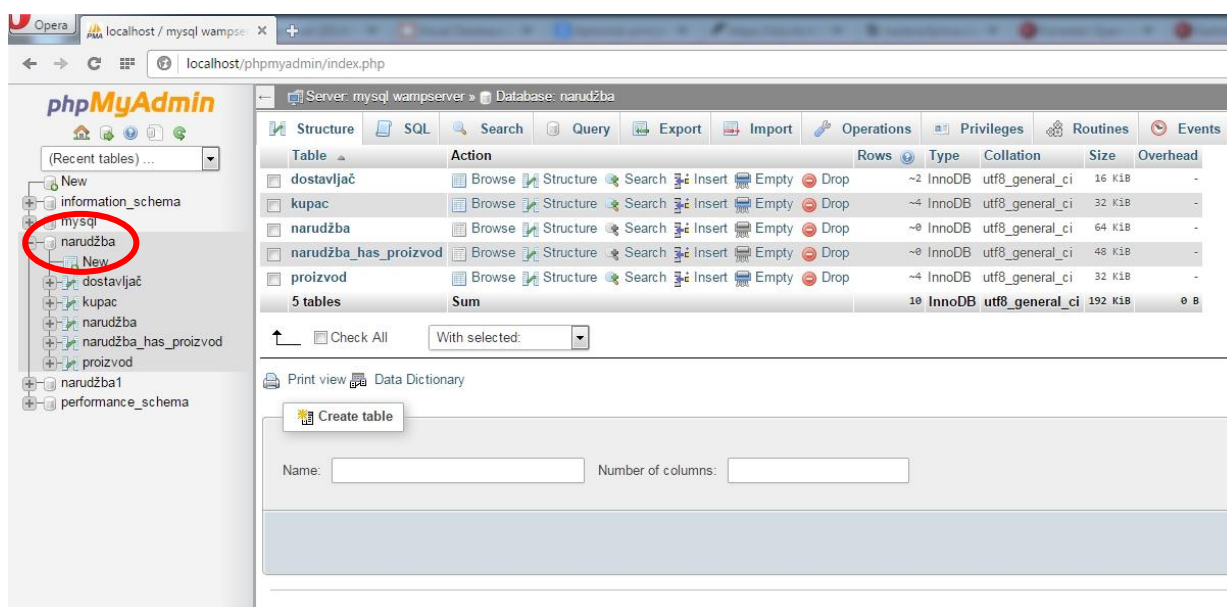


Izvor: Valastita izrada 17. 05. 2015.

## 6.4. Prebacivanje baze na poslužitelj

Gotovu bazu podataka prebacujemo na poslužitelj da bi se ona mogla koristiti od strane korisnika u našem slučaju bi to bili kupci. Pošto se poslužiteljski prostor inače zakupljuje mi ovom prilikom koristimo program WAMP koji emulira poslužiteljsko okruženje (Apache2, PHP, MySQL) sa phpMyAdmin korisničkim sučeljem za upravljanje bazama podataka.

Slika 13. phpMyAdmin sučelje



Izvor: Valastita izrada 17. 05. 2015.

Da bi ova baza podataka potpuno zaživjela treba napraviti web stranicu na koju bi korisnici tj. kupci mogli pristupati te tražiti i naručivati proizvode iz naše baze. Samim time bi našu bazu podataka stavili u funkciju za koju je i namijenjena.

## 6.5. Prednosti MySQL baze podataka otvorenog koda

Glavna i najveća prednost MySQL baze podataka i PHP programskog jezika koji smo koristili u ovom primjeru je to što su oni otvorenog koda. Ovdje ćemo sumirati neke od njih: otvoreni standardi i neovisnost koji nas osiguravaju u slučaju da MySQL baza koju smo koristili

u ovom primjeru i trenutačno nam odgovara, a u budućnosti ako nam to zatreba možemo se prebaciti na neku drugu vrstu baze (SQLite), ovako nešto ne bi bili u mogućnosti napraviti sa komercijalnim bazama podataka jer one u većini slučajeva nemaju interoperabilnost. Tu dolazimo do najveće prednosti otvorenog koda, a to je cijena pri čemu je MySQL koji koristimo u primjeru besplatan dok kod konkurentskog sustava za upravljanje bazama podataka Oracle, u najpovoljnijoj varijanti godišnja licenca iznosi 2.135,00 dolara (<https://shop.oracle.com> 09. 09. 2015).

## 7. Zaključak

Sustavi upravljanja bazom podataka kategorizirani su prema modelu koji podržavaju: orijentirani prema objektu, odnosni, mrežni itd. Najveći dio sustava za upravljanje bazama nije ovisan o modelu podataka već je zaokupljen upravljanjem čimbenika kao što su: performanse, integritet, obnove nakon hardverskih propusta i podudarnosti. U tim područjima su velike razlike među proizvodima.

Važnost sustava za upravljanje bazama podatka raste sa važnosti Interneta. Budući da veliki broj današnjih Web stranica nije statičan nego povezan sa bazama podataka u koje spremaju podatke koje prikupljaju, MySQL je imao važnu ulogu u stvaranju interneta. Procvat poslovanja na internetu krenuo je većim brzinama pristupa Internetu te pojavom *open-source* baza podataka. MySQL je olakšao korištenje baze podataka kod običnih korisnika koji s malim znanjem mogu stvarati Web aplikacije i Web stranice. Tvrtke mogu na pristupačnim proračunima izgraditi svoj informacijski sustav ili e-poslovanje s minimalnim naporima i troškovima.

Baze podataka daju nam mogućnost da sačuvamo podatke koji su nam potrebni. S pravom možemo reći da baze podataka pamte ono što mi možemo zaboraviti, a njima se to svakako ne smije i ne može dogoditi. Potrebno je samo ispravno rukovoditi računalom i određenim bazama.

U dijelu rada - Poglavlje 6. „Primjer MySQL baze podataka“ je prikazano kako kreirati MySQL bazu podatka. Bazi smo pristupali preko aplikacija MySQL Workbench 6.1 CE. Kada smo definirali kako bi baza morala izgledati osmislili smo model. U radu smo prošli kroz kreiranje tablica, veza između tablica, unos podataka, prikaz dijagrama i prikaz phpMyAdmin sučelja.

Na modelu baze podataka iz primjera u Poglavlju 6. prikazano je da male tvrtke ne moraju koristiti komercijalne i skupe proizvode nego je moguće koristiti *open source* rješenja kao što je MySQL koji može zadovoljiti osnovne potrebe tih tvrtki. *Open source* je preporučljiv tamo gdje je bitno imati niske informatičke troškove, pri čemu se zadržava funkcionalnost sustava, a omogućuju uštede u poslovanju.

## Popis literature

### Knjige:

1. C.J. Date: An Introduction to Database Systems, 8th Edition. Addison-Wesley, Reading MA, 2003.
2. R. Elmasri, S. Navathe: Fundamentals of Database Systems, 6th Edition. Addison-Wesley, Reading MA, 2010.
3. Alexander Sasha Pachev: MySQL enterprise solutions. Indianapolis, Ind. Wiley, 2003.
4. Ramakrishnan R., Gehrke J. Database Management Systems, 3rd Edition. 2002.
5. A.Silberschatz, H.F. Korth, S.Sudarshan, Database system concepts, 6th Edition, New York: The McGraw-Hill Companies, 2011.
6. M. Widenius, D. Axmark: MySQL Reference Manual. O'Reilly & Associates, Sebastopol CA, 2002.

### Internetski izvori:

1. <http://otvorenikod.weebly.com> (15. 08. 2014.)
2. <http://www.gnu.org/philosophy/open-source-misses-the-point.hr.html> (12. 09. 2015)
3. <http://www.wired.com/2009/08/0825-torvalds-starts-linux/> (12. 09. 2015.)
4. <http://www.donos.hr> (10. 09. 2014.)
5. <http://www.cnet.com/news/gartner-85-percent-of-companies-using-open-source/> (19. 08. 2014.)
6. <http://www.cnet.com/news/forrester-open-source-delivers-cost-and-quality-benefits/> (19. 08. 2014.)
7. [http://www.extropia.com/tutorials/sql/hierarchical\\_databases.html](http://www.extropia.com/tutorials/sql/hierarchical_databases.html) (10 05. 2015.)
8. [http://databasemanagement.wikia.com/wiki/Network\\_Database\\_Model](http://databasemanagement.wikia.com/wiki/Network_Database_Model) (20. 05. 2015.)
9. <https://support.office.com/en-us/article/Access-2010-specifications-1e521481-7f9a-46f7-8ed9-ea9dff1fa854> (31. 08. 2015)
10. <http://databasemanagement.wikia.com/wiki/DBMS> (20. 05. 2015.)

11. <https://shop.oracle.com/pls/ostore/f?p=700:6:0::NO::> (09. 09. 2015.)
12. <http://databasemanagement.wikia.com/wiki/MySQL> (12. 09. 2015)
13. <https://www.mysql.com> (14. 05. 2015.)

## Popis slika

Slika 1. Primjer hijerarhijskog modela.....	5
Slika 2. Primjer mrežnog modela .....	6
Slika 3. Primjer relacijskog modela.....	8
Slika 4. Primjer objektnog modela .....	9
Slika 5. Jednostavan prikaz sustava za upravljanje bazama podataka .....	16
Slika 6. MySQL logo.....	21
Slika 7. Popis izrađenih tablica baze podataka.....	28
Slika 8. Kreiranje tablice .....	30
Slika 9. Uređivanje i brisanje podataka unutar tablice .....	31
Slika 10. Povezivanje tablica.....	32
Slika 11. ER dijagram.....	33
Slika 12. Unos podataka u tablice .....	33
Slika 13. phpMyAdmin sučelje .....	34



## **Popis priloga**

Prilog 1. CD sa primjerom baze podataka *Webshop-a* u MySQL-u