

1. Šta je MySQL?

MySQL je najpopularniji sistem otvorenog koda (*Open Source*) za upravljanje bazama podataka. Proizvod je kompanije MySQL AB poreklom iz Švedske.

- **MySQL je sistem za upravljanje bazama podataka.** Baza podataka je strukturirana kolekcija podataka. Može biti sve od najjednostavnije liste za kupovinu do kolekcije ogromne količine podataka neke korporacije. Za dodavanje, pristup i obradu podataka koji su smešteni u bazi podataka potreban je sistem za upravljanje bazama podataka kao što je MySQL server.
- **MySQL je sistem za upravljanje relacionim bazama podataka.** U relacionoj bazi podataka se podaci smeštaju u više međusobno povezanih tabela. Ovim se dobija na brzini i fleksibilnosti. SQL deo naziva "MySQL" potiče od "*Structured Query Language*" (strukturirani jezik za upite). SQL je najrasprostranjeniji standardizovani jezik koji se koristi za pristup bazama podataka i definisan je ANSI/ISO SQL standardom.
- **MySQL softver je softver otvorenog koda (*Open Source*).** Ovo znači da svako može koristiti i modifikovati softver. Svako može preuzeti MySQL softver sa interneta i koristiti ga bez plaćanja. Svaki korisnik može proučiti izvorni kod softvera i izmeniti ga u skladu sa svojim potrebama.
- **MySQL server je veoma brz, pouzdan i jednostavan za korišćenje.** MySQL server je prvobitno bio razvijen radi mnogo bržeg upravljanja velikim bazama podataka u odnosu na postojeća rešenja i uspešno se koristi u visokozahtevnim proizvodnim okruženjima veći broj godina. Iako pod stalnim razvojem, MySQL server danas nudi bogat i koristan skup funkcija. Konektivnost, brzina i sigurnost čine MySQL server jako pogodnim za pristup bazama podataka na internetu.
- **MySQL server radi u klijent/server sistemima.**
- **Postoji jako puno softverskih rešenja koja podržavaju rad sa MySQL serverom.**

2. Instaliranje MySQL-a

Prilikom instaliranja MySQL-a trebalo bi koristiti nalog koji ima administratorska prava.

2.1. Izbor instalacionog paketa

Za MySQL 5.0 moguće je izabrati jedan od tri instalaciona paketa:

- **The Essentials Package (osnovni paket):** Ovaj paket sadrži minimalan skup fajlova koji je neophodan da bi se instalirao MySQL, uključujući čarobnjaka za konfigurisanje. Ovaj paket ne uključuje opcione komponente, kao što su ugrađeni (*embedded*) server i paket za testiranje performansi (benchmark suite). Ugrađeni server predstavlja ugrađenu biblioteku koja može da radi sa više niti (*multi-threaded*) i koja se može povezati u aplikaciji da bi se dobio manji, brži i za upravljanje lakši samostalan proizvod. Paket za testiranje performansi služi da da korisniku informaciju koje operacije data SQL implementacija izvodi dobro, a koje loše. Na ovaj način se utvrđuje gde su uska grla aplikacije i baze podataka.
- **The Complete Package (kompletan paket):** Ovaj paket sadrži sve fajlove potrebne za kompletno instaliranje, uključujući čarobnjaka za konfigurisanje. Ovaj paket uključuje opcione komponente, kao što su ugrađeni server i paket za testiranje performansi.
- **The Noninstall Archive:** Ovaj paket sadrži sve fajlove koji se nalaze u prethodnom paketu sa izuzetkom čarobnjaka za konfigurisanje. Ovaj paket ne uključuje automatski alat za instaliranje i mora biti manuelno instaliran i konfigurisan.

Osnovni paket je preporučan za većinu korisnika.

U daljem tekstu biće dat prikaz instaliranja MySQL servera korišćenjem kompletnog paketa. Biće instaliran MySQL Community Server (trenutna stabilna verzija je 5.0.27) koji je besplatan.

2.2. Preuzimanje i pokretanje MySQL čarobnjaka za instaliranje

MySQL instalacioni paketi se mogu preuzeti sa adrese <http://dev.mysql.com/downloads/>. Pokretanje čarobnjaka (za kompletan paket) se vrši dvostrukim pritiskom levim tasterom miša na setup.exe fajl. Pojavljuje se prva stranica čarobnjaka.



Pritiskom na dugme **Next** prelazi se na sledeću stranicu čarobnjaka na kojoj se bira tip instalacije.

2.3. Izbor tipa instalacije

Moguće je izabrati jedan od tri tipa instalacije: **Typical (tipična)**, **Complete (kompletna)** i **Custom (prilagođena)**.

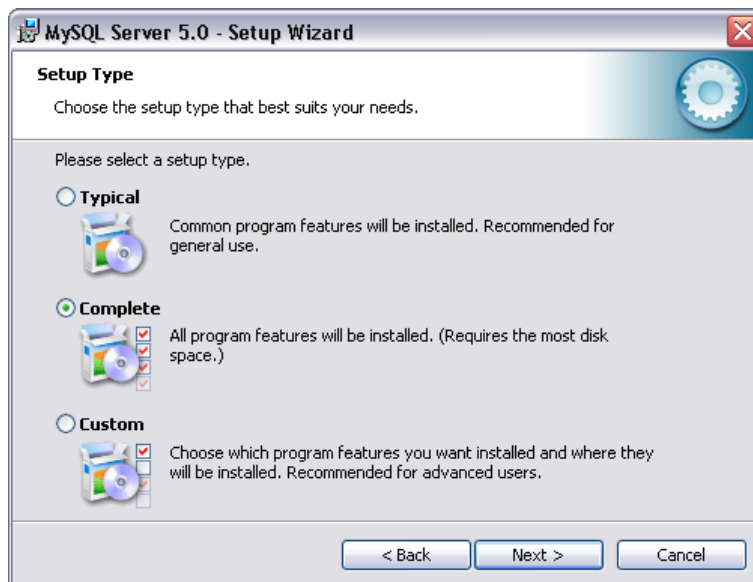
Izborom tipične instalacije instalira se MySQL server, mysql klijent za rad sa komandne linije i još neki alati za rad sa komandne linije.

Izborom kompletne instalacije instaliraju se sve komponente koje su uključene u instalacioni paket.

Prilagođeni tip instalacije daje korisniku potpunu kontrolu nad tim koji će paketi biti instalirani i koja će biti lokacija i naziv instalacionog foldera.

Izborom tipične ili kompletne instalacije i pritiskom na dugme **Next** prelazi se na prozor za potvrdu gde se još jednom mogu proveriti izabrane opcije i otpočeti instalacija.

Potrebno je izabrati opciju **Complete** i pritisnuti na dugme **Next**.



2.4. Prozor za potvrdu

Nakon izbora tipa instalacije i eventualnog izbora komponenti koje će biti instalirane (ako je izabran Custom tip instalacije) prelazi se na prozor za potvrdu. Ovde se mogu videti tip instalacije i instalacioni folder.

Ako su podešavanja zadovoljavajuća pritiskom na dugme **Install** otpočinje instalacija.



Nakon završetka instalacije pojavljuje se prozor koji nudi mogućnost registracije korisnika na MySQL sajtu. Registracija daje mogućnost potpunog korišćenja MySQL foruma, zajedno sa mogućnošću prijavljivanja bagova i mogućnošću pretplate na MySQL bilten. Na ovom prozoru treba izabrati opciju **Skip Sign-Up** (preskakanje registracije) i pritisnuti na dugme **Next**.

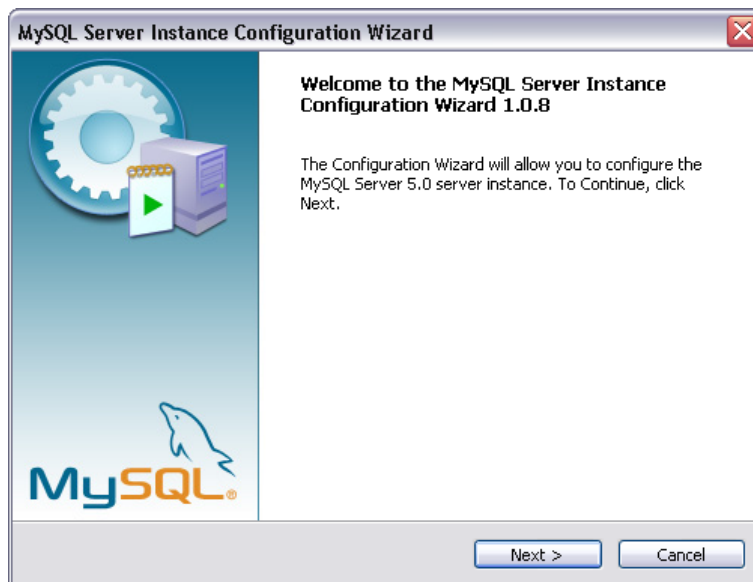


Poslednji prozor čarobnjaka daje obaveštenje o završetku instalacije i daje mogućnost pokretanja MySQL čarobnjaka za konfigurisanje koji se koristi za kreiranje konfiguracionog fajla, instaliranje MySQL servisa i konfigurisanje sigurnosnih podešavanja. Na ovom prozoru treba zadržati unapred čekiranu opciju **Configure the MySQL Server now** i pritisnuti na dugme **Finish**.



2.5. Čarobnjak za konfigurisanje MySQL servera

Čarobnjak za konfigurisanje MySQL servera automatizuje proces konfigurisanja servera. On kreira prilagođeni MySQL konfiguracioni fajl (my.ini ili my.cnf) postavljanjem niza pitanja korisniku i primenom niza odgovora na šablon. Ovaj čarobnjak se pokreće automatski kao deo instalacionog procesa. Nakon ovog prvog pokretanja korisnik može ponovo pokrenuti čarobnjaka kada je potrebno da promeni konfiguracione parametre svog servera. my.ini fajl se može modifikovati i otvaranjem u nekom od tekstualnih editora i vršenjem potrebnih izmena. Čarobnjak smešta my.ini fajl u instalacioni folder MySQL servera (C:\Program Files\MySQL\MySQL Server 5.0).



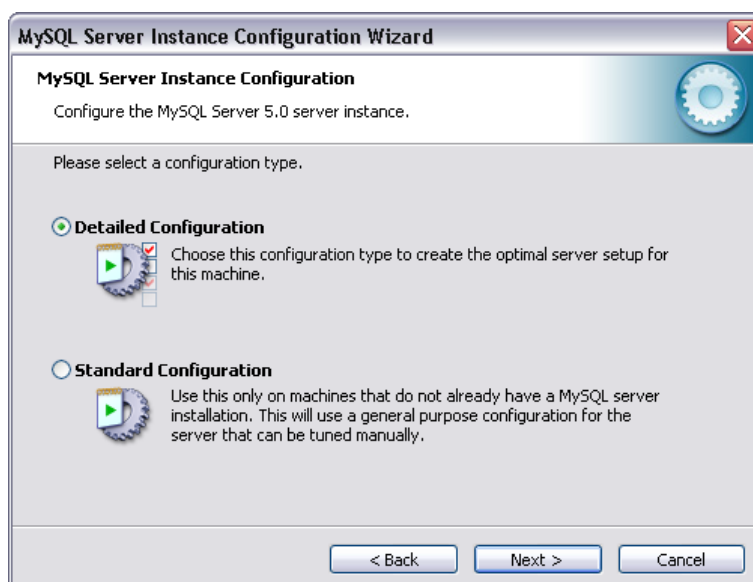
Pritiskom na dugme **Next** prelazi se na sledeću stranicu čarobnjaka.

2.6. Izbor tipa konfiguracije

Postoje dva osnovna tipa konfiguracije: Detailed Configuration (detaljna konfiguracija) i Standard Configuration (standardna konfiguracija). Standardna konfiguracija je namenjena za nove korisnike koji žele da započnu rad sa MySQL-om brzo bez potrebe za donošenjem mnogo odluka vezano za konfiguraciju servera. Detaljna konfiguracija je namenjena za napredne korisnike.

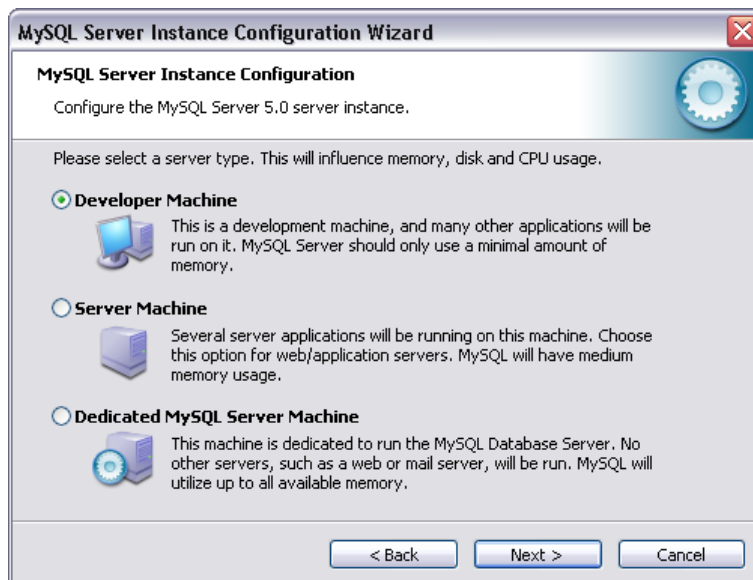
Korisnicima koji prvi put koriste MySQL server i žele server koji će biti konfigurisan kao jednokorisnička razvojna mašina, bi trebalo da odgovara standardna konfiguracija. Kod ovog tipa konfiguracije čarobnjak sve konfiguracione opcije podešava automatski osim opcija vezanih za servis i sigurnost.

Treba izabrati opciju **Detailed Configuration** i pritisnuti na dugme **Next**.



2.7. Izbor tipa servera

Postoje tri tipa servera. Tip servera koji korisnik izabere utiče na odluke koje pravi čarobnjak vezano za korišćenje memorije, hard diska i procesora.

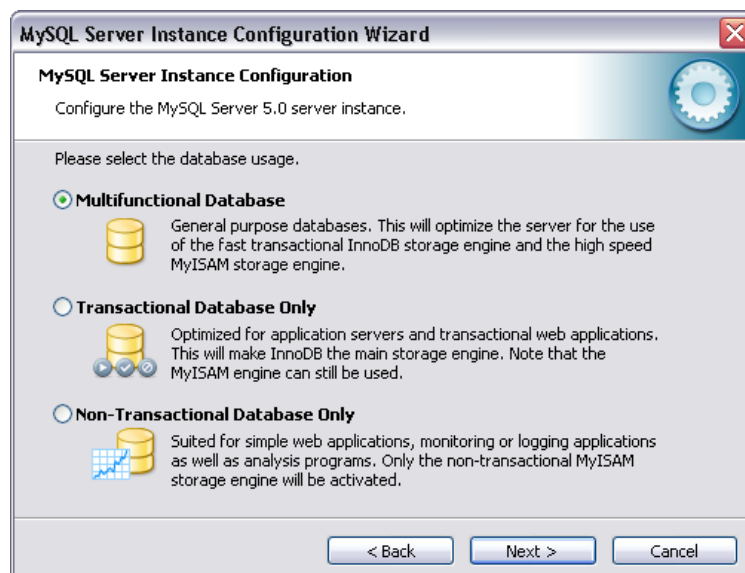


- Developer Machine (razvojna mašina): Ovu opciju treba izabrati za tipičnu desktop radnu stanicu gde je MySQL namenjen samo za lične potrebe. Pretpostavlja se da su mnoge druge aplikacije istovremeno aktivne. MySQL server je konfigurisan da koristi minimum sistemskih resursa.
- Server Machine (serverska mašina): Ovu opciju treba izabrati za serversku mašinu gde je MySQL server aktivan zajedno sa drugim serverskim aplikacijama, kao što su FTP, email i WEB serveri. MySQL server je konfigurisan za osrednje korišćenje sistemskih resursa.
- Dedicated MySQL Server Machine (serverska mašina namenjena samo za MySQL server): Ovu opciju treba izabrati za serversku mašinu na kojoj će biti aktivan samo MySQL server. Pretpostavlja se da druge aplikacije nisu aktivne. MySQL server je konfigurisan da koristi sve raspoložive sistemske resurse.

Treba izabrati opciju **Developer Machine** i pritisnuti na dugme **Next**.

2.8. Upotreba baze podataka

Ovaj prozor čarobnjaka omogućava korisniku da odluči koje će mašine za skladištenje koristiti prilikom kreiranja MySQL tabela. Izabrana opcija definiše da li je InnoDB mašina za skladištenje na raspolaganju i koji procenat resursa servera je na raspolaganju za ovu mašinu.



- Multifunctional Database (multifunkcionalna baza podataka): Ova opcija omogućava korišćenje i InnoDB i MyISAM mašine za skladištenje i deli resurse servera jednako između ove dve mašine. Ova opcija je preporučena za korisnike koji koriste obe mašine za skladištenje svakodnevno.
- Transactional Database Only (samo transakciona baza podataka): Ova opcija omogućava korišćenje i InnoDB i MyISAM mašine za skladištenje ali dodeljuje većinu resursa servera InnoDB mašini za skladištenje. Ova opcija je preporučena za korisnike koji skoro isključivo koriste InnoDB mašinu za skladištenje, a minimalno koriste MyISAM mašinu za skladištenje.
- Non-Transactional Database Only (samo netransakciona baza podataka): Ova opcija isključuje potpuno InnoDB mašinu za skladištenje i sve resurse servera dodeljuje MyISAM mašini za skladištenje. Ova opcija je preporučena za korisnike koji ne koriste InnoDB mašinu za skladištenje.

Tipovi tabela se nazivaju i mašine za skladištenje (*storage engine*). To odražava činjenicu da je za upotrebu nekih tipova tabela neophodna značajna količina posebnog programskog koda koji upravlja smeštanjem podataka, indeksiranjem, zaključavanjem podataka i pristupanjem disku. To takođe odražava suštinu baze podataka: skladištenje podataka.

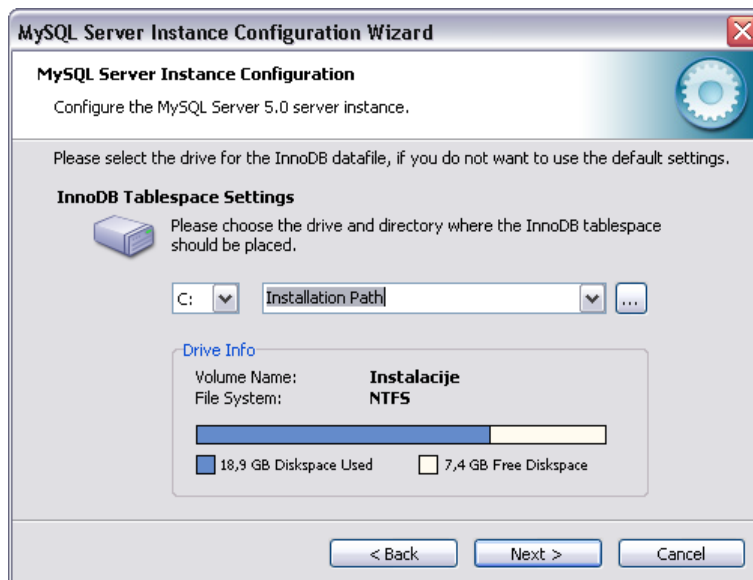
Tabele koje podržavaju transakcije omogućavaju da korisnik zada da je određena grupa upita nedeljiva jedinica obrade - transakcija. Trebalo bi da se cela transakcija obavi do kraja, a ako to nije moguće, baza podataka mora da je poništi (*roll back*), odnosno da se vrati u stanje u kojem je bila pre transakcije.

Može se zamisliti baza podataka o prometu na bankovnim računima. Ako se želi da se prebaci 1000 dinara s jednog računa na drugi, potrebna su najmanje dva SQL upita - jedan, koji na prvom računu smanjuje stanje za 1000 dinara i drugi, koji povećava stanje na drugom računu za 1000 dinara. Bila bi prava katastrofa kada bi se zbog nečeg (na primer, nestanka struje) prvi upit izvršio do kraja, ali ne i drugi. U takvim slučajevima bilo bi neuporedivo bolje da se izvrše ili oba upita, ili nijedan, jer baza podataka mora uvek biti u usklađenom stanju.

Treba izabrati opciju **Multifunctional Database** i pritisnuti na dugme **Next**.

2.9. Lokacija InnoDB Tablespace fajlova (fajlovi tabelarnog prostora)

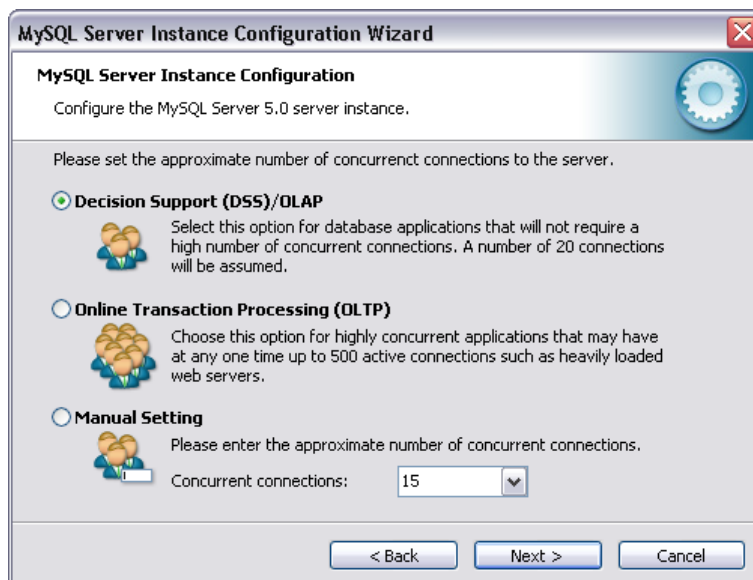
Dok MyISAM smešta svaku tabelu u zaseban fajl, InnoDB smešta sve tabele i indekse u tabelarni prostor, što znači da se delovi jedne tabele mogu nalaziti u više fajlova. Neki korisnici žele da lociraju InnoDB tablespace fajlove na neku drugu lokaciju različitu od MySQL server data foldera. Smeštanje tablespace fajlova na zasebnu lokaciju može biti poželjno ako sistem ima uređaje za skladištenje podataka većeg kapaciteta ili većih performansi, kao što je RAID sistem za skladištenje.



Da bi se promenila podrazumevana lokacija za InnoDB tablespace fajlove treba izabrati novu particiju ili novi hard disk iz padajuće liste i nakon toga izabrati novi put iz padajuće liste. Ako korisnik želi da sam kreira put treba da pritisne na dugme U ovom prozoru ne treba ništa menjati. Pritisnuti na dugme **Next**.

2.10. Broj istovremenih konekcija

Da bi se izbegle situacije u kojima bi server radio da nedovoljnim resursima, neophodno je ograničiti broj istovremenih konekcija na server koje se mogu uspostaviti. Ovaj prozor omogućava da se izabere očekivano korišćenje servera i u skladu sa tim podesi granica za broj istovremenih konekcija. Granica za broj istovremenih konekcija se može podesiti i manualno.

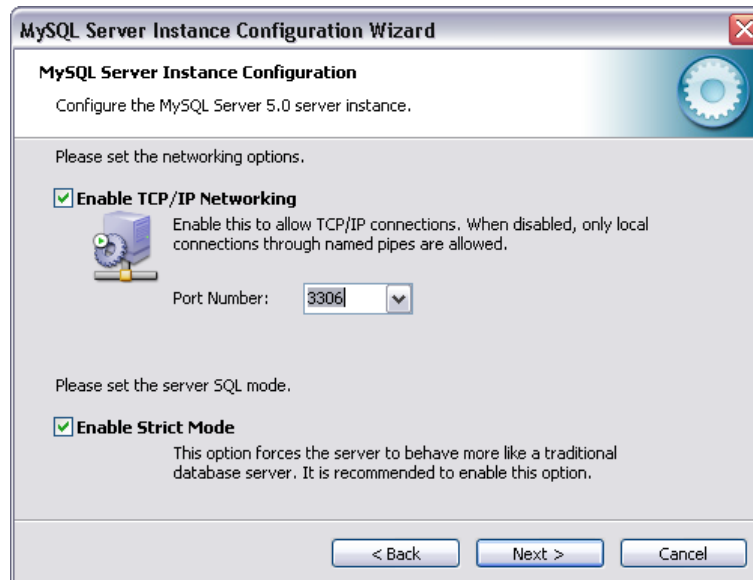


- Decision Support (DSS)/OLAP (podrška odlučivanju): Ovu opciju treba izabrati ako server ne zahteva veliki broj istovremenih konekcija. Maksimalan broj konekcija je podešen na 100, a pretpostavljeno da će biti prosečno 20 istovremenih konekcija.
- Online Transaction Processing (OLTP) (obrada online transakcija): Ovu opciju treba izabrati ako server zahteva veliki broj istovremenih konekcija. Maksimalan broj konekcija je podešen na 500.

- Manual Setting (manuelno podešavanje): Ovu opciju treba izabrati da bi se manuelno podesio maksimalan broj istovremenih konekcija. Treba izabrati maksimalan broj iz padajuće liste ili ga ukucati ako ne postoji u padajućoj listi. Treba izabrati opciju **Decision Support (DSS)/OLAP** i pritisnuti na dugme **Next**.

2.11. Mrežne i Strict Mode opcije

Mrežne opcije omogućavaju da se dozvoli ili zabrani kreiranje TCP/IP konekcija i da se konfiguriše broj porta koji će se koristiti za konektovanje na MySQL server.



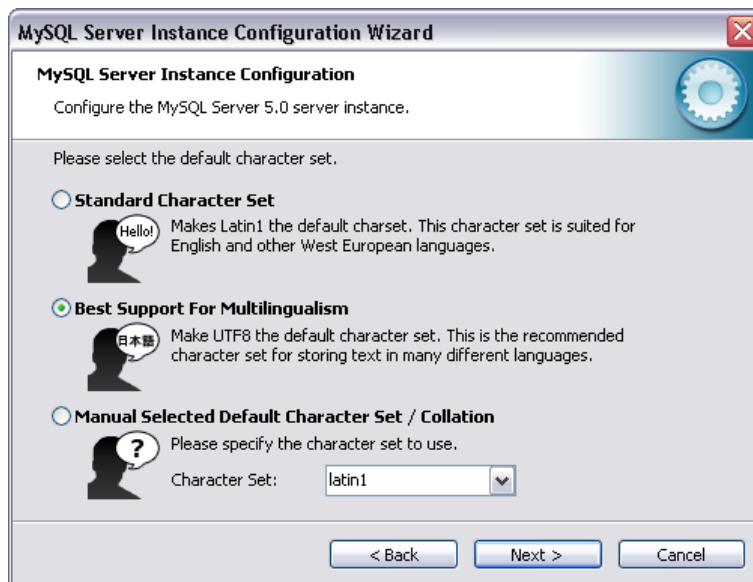
Kreiranje TCP/IP konekcija je podrazumevano dozvoljeno. Port 3306 je podrazumevana opcija. Da bi se promenio port potrebno ga je izabrati iz padajuće liste ili ga dokucati ako se ne nalazi u listi.

Strict Mode čini da se server ponaša kao drugi sistemi za upravljanje bazama podataka. Strict Mode kontroliše kako MySQL manipuliše ulaznim vrednostima koje su loše ili nedostaju.

Treba čekirati opciju **Enable TCP/IP Networking** i zadržati port **3306**. Takođe treba čekirati opciju **Enable Strict Mode** i pritisnuti na dugme **Next**.

2.12. Character Set opcija

MySQL server podržava različite setove karaktera i moguće je podesiti podrazumevani set karaktera za server koji će biti primenjen na sve tabele, kolone i baze podataka. Set karaktera je skup karaktera i kodiranja za te karaktere. Collation (uparivanje) je skup pravila za poređenje karaktera u okviru seta karaktera.



- Standard Character Set (standarni set karaktera): Izabрати ovu opciju ako se kao podrazumevani set karaktera želi koristiti latin1. Ovaj set karaktera se koristi za engleski i mnoge zapadnoevropske jezike.
- Best Support For Multilingualism (najbolja podrška za veliki broj jezika): Izabрати ovu opciju ako se kao podrazumevani set karaktera želi koristiti utf8. Ovo je Unicode skup karaktera i može smeštati karaktere iz mnogo različitih jezika.
- Manual Selected Default Character Set / Collation (manuelan izbor podrazumevanog seta karaktera / uparivanja): Izabрати ovu opciju ako se želi da se manuelno definiše podrazumevani set karaktera. Željeni set karaktera treba izabrati iz padajuće liste.

Treba izabrati opciju **Best Support For Multilingualism** i pritisnuti na dugme **Next**.

2.13. Opcije vezane za servis

Na Windows NT baziranim sistemima MySQL server može biti instaliran kao Windows servis. Kada je instaliran na ovaj način MySQL server može biti startovan automatski prilikom pokretanja sistema i čak restarstovan automatski od strane Windows-a u slučaju problema sa servisom.

Čarobnjak instalira MySQL server kao servis podrazumevano i daje naziv MySQL servisu. Ako se ne želi da se MySQL server instalira kao servis treba odčekirati opciju Install As Windows Service. Naziv servisa se može promeniti izborom drugog naziva iz padajuće liste ili ukucavanjem novog naziva ako se željeni ne nalazi u padajućoj listi.

Ako se želi da se MySQL server instalira kao servis ali da se servis ne startuje automatski prilikom pokretanja sistema treba odčekirati opciju Launch the MySQL Server automatically.



Treba čekirati opciju **Install As Windows Service**, zadržati ponuđeno ime servisa i čekirati opciju **Launch the MySQL Server automatically**. Treba čekirati opciju **Include Bin Directory in Windows PATH**. Sada je moguće pokrenuti bilo koji MySQL izvršni program kucanjem njegovog naziva u interpreteru komandi (Command Prompt) iz bilo kog foldera bez potrebe za prosleđivanjem putanje do bin foldera. Pritisnuti na dugme **Next**.

2.14. Sigurnosne opcije

Ovde je neophodno i jako preporučljivo definisati root password za MySQL server i čarobnjak podrazumevano zahteva od korisnika da to uradi. Ako korisnik ne želi da ovo uradi treba da odčekira opciju **Modify Security Settings**. root je podrazumevani administrativni nalog u MySQL sistemu za upravljanje pravima korisnika.



Da bi se definisao root password treba uneti password u polja **New root password** i **Confirm**.

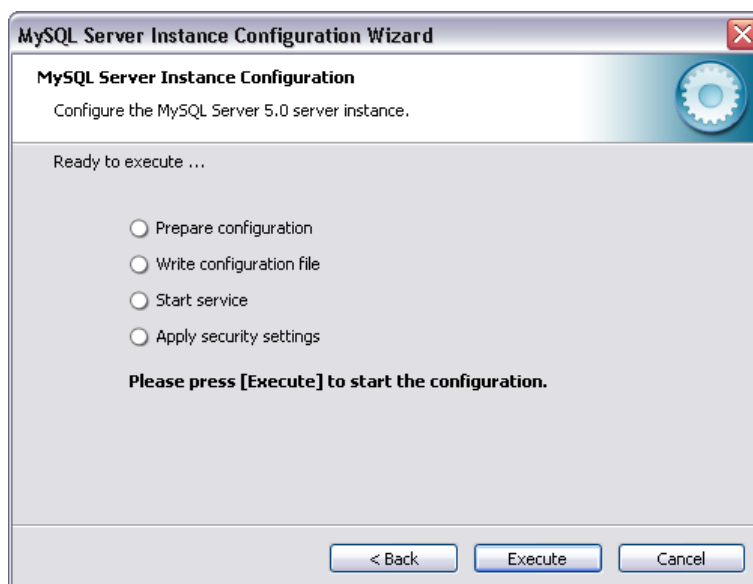
Da bi se sprečilo konektovanje na server preko root naloga sa drugih računara iz mreže treba odčekirati opciju **Enable root access from remote machines**. Ovo povećava sigurnost root naloga.

Da bi se kreirao anonimni nalog treba čekirati opciju **Create An Anonymous Account**. Za anonimne naloge nije potrebno zadavanje korisničkog imena i lozinke prilikom

prijavljivanja. Kreiranje anonimnog naloga može umanjiti sigurnost servera i izazvati probleme sa logovanjem i pravima pristupa. Iz ovih razloga nije preporučljivo. Treba čekirati opciju **Modify Security Settings**, uneti i potvrditi root password (kurs2006) i odčekirati opciju **Enable root access from remote machines**. Pritisnuti na dugme **Next**.

2.15. Prozor za potvrdu

Poslednji prozor čarobnjaka za konfigurisanje MySQL servera je prozor za potvrdu. Da bi otpočeo proces konfigurisanja treba pritisnuti dugme **Execute**.



Nakon završetka procesa konfigurisanja pojavljuje se prozor sa rezultatima. Pritiskom na dugme **Finish** izlazi se iz čarobnjaka.

2.16. Raspored foldera nastalih u toku instalacije

Za MySQL 5.0 na Windows-u podrazumevani instalacioni folder je: C:\Program Files\MySQL\MySQL Server 5.0. U instalacionom folderu se nalaze sledeći folderi:

Folder	Sadržaj foldera
bin	MySQL server i klijentski programi
data	Log fajlovi i baze podataka, odnosno podaci
Docs	Uputstvo u .chm formatu
examples	Programi i skriptovi za primer
include	Include (header) fajlovi ili fajlovi zaglavlja (koriste se prilikom kompajliranja)
lib	Biblioteke funkcija koje MySQL koristi
scripts	Skriptovi napisani u jeziku Perl koji obavljaju korisne poslove
share	Fajlovi sa tekstom poruka o greškama koje MySQL šalje
sql-bench	Programi za testiranje MySQL-a

2.17. Pregled izvršnih fajlova

MySQL-ovi izvršni fajlovi nalaze se u folderima bin i scripts. U folderu bin se može naći više fajlova koji u svom nazivu sadrže mysqld (recimo mysqld.exe, mysqld-nt.exe, mysqld-max.exe). To su programi različitih verzija mysql servera. U ovom folderu se nalazi i mysql.exe što je MySQL Monitor. Osim ovih najvažnijih programa mogu se naći i drugi:

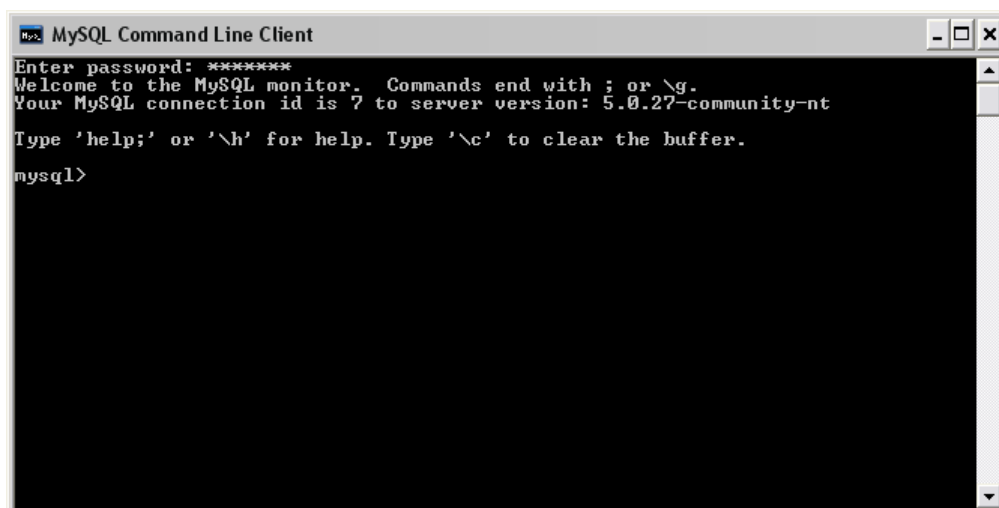
- mysqladmin.exe: Služi za obavljanje administrativnih funkcija

- `myisamchk.exe`: Služi za proveru i popravljanje ošteženih MyISAM tabela
- `mysqldump.exe`: Omogućava izradu rezervnih kopija baze podataka
- `mysqlbinlog.exe`: Služi za čitanje binarnih log fajlova ili dnevnika izmena gde se beleže podaci o svim izmenama na bazi. Beleženje izmena je veoma korisno u slučaju oporavljanja od katastrofalnih grešaka.
- `mysqlshow.exe`: Služi za prikazivanje podataka o bazama podataka i njihovim tabelama

3. Klijentski programi

3.1. MySQL klijent za rad sa komandne linije

MySQL klijent za rad sa komandne linije (`mysql`) (MySQL Monitor) služi za interaktivno izvršavanje SQL iskaza. Konfigurisan je da se konektuje na server sa root nalogom, pa se prilikom pokretanja od korisnika zahteva unos lozinke za root nalog ukoliko je definisana. Može se pokrenuti na sledeći način: **start** → **All Programs** → **MySQL** → **MySQL Server 5.0** → **MySQL Command Line Client**. Nakon pokretanja neophodno je uneti lozinku za root nalog i pritisnuti na taster **Enter**. Korisnik može MySQL-ove i SQL-ove komande upisivati direktno u MySQL Monitor.



```

MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 5.0.27-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>

```

Nakon prijave može se videti koje sve baze podataka postoje na serveru korišćenjem komande `SHOW`:

```
show databases;
```

Spisak bi trebalo da sadrži tri baze. Jedna od baza je `mysql`. To je sistemska baza podataka u kojoj se čuvaju podaci o korisničkim nalogima i njihovim pravima.

Većinu komandi koje korisnik otkuca u MySQL monitoru mora završiti znakom tačka i zarez (;) inače ih MySQL neće izvršiti.

Iz MySQL monitora korisnik se može odjaviti tako što otkuca `\q` (slovo `q` potiče od reči `quit`). Ova komanda se ne završava znakom tačka i zarez. Postoji grupa komandi koje počinju znakom `\` (obrnuta kosa crta ili *backslash*). Nijedna od njih se ne završava znakom tačka i zarez. Spisak tih komandi se može dobiti ako se otkuca `\h` (slovo `h` potiče od reči *help*).

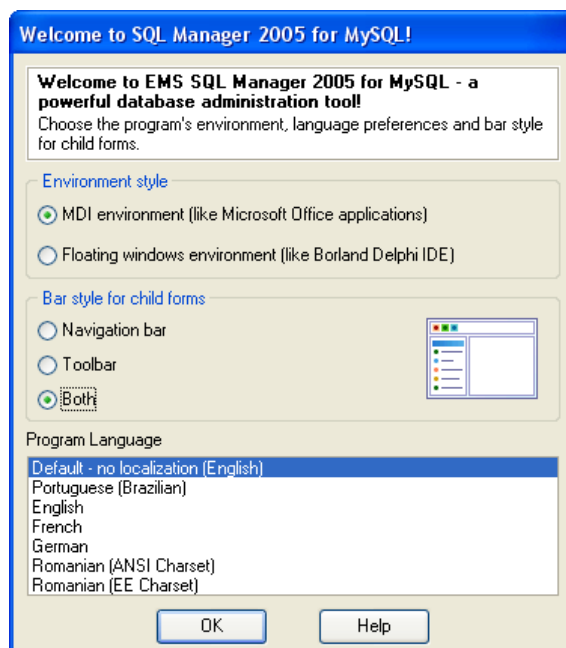
3.2. Klijentski programi sa grafičkim korisničkim okruženjem

Klijentski programi sa grafičkim korisničkim okruženjem služe za razvoj i administraciju baza podataka. Njihova glavna prednost su grafičko korisničko okruženje i moćni alati koji stoje na raspolaganju korisniku. Proizvođač MySQL-a takođe nudi klijentske programe

ovog tipa i oni su besplatni. Postoji jako puno klijentskih programa drugih proizvođača koji rade sa MySQL serverom i među njima ima i besplatnih rešenja i onih koja se plaćaju.

3.2.1 SQL Manager 2005 Lite for MySQL

SQL Manager Lite for MySQL je besplatan klijentski program. Proizvod je kompanije EMS (EMS Database Management Solutions). Trenutno aktuelna verzija je 3.7.7.1. Nakon instaliranja programa pojavljuje se pozdravni prozor na kome je potrebno izabrati neke od ponuđenih opcija.

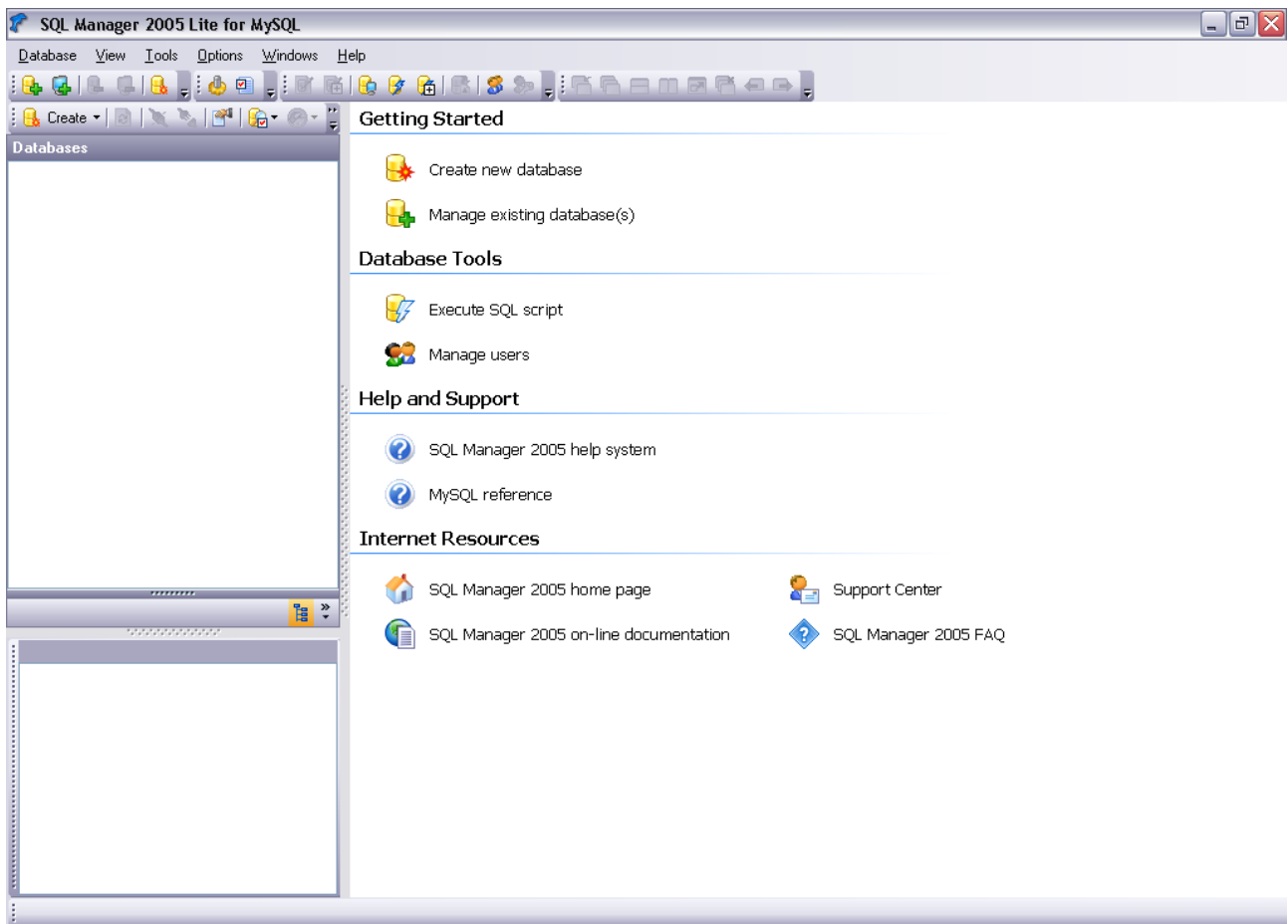


U ovom prozoru je u delu Environment style (stil okruženja) potrebno izabrati opciju **MDI environment (like Microsoft Office applications)**. Ovde se misli na stil korisničkog okruženja programa, odnosno ponašanje glavnog programskog prozora. Izborom navedene opcije definiše se da će okruženje biti nalik okruženju Microsoft Office aplikacija. Druga opcija (Floating windows environment (like Borland Delphi IDE)) daje okruženje sa plivajućim prozorima.

U delu Bar style for child forms definiše se lokacija akcionih dugmadi. Ona se mogu nalaziti u oknu za navigaciju (Navigation bar), na paletama sa alatkama (Toolbar) ili i u oknu za navigaciju i na paletama sa alatkama (Both). Ovde treba izabrati opciju **Both**.

U delu Program Language potrebno je izabrati jezik za interfejs iz liste ponuđenih jezika. Ovde treba izabrati **Default - no localization (English)**.

Pritiskom na dugme **OK** otvara se programski prozor.



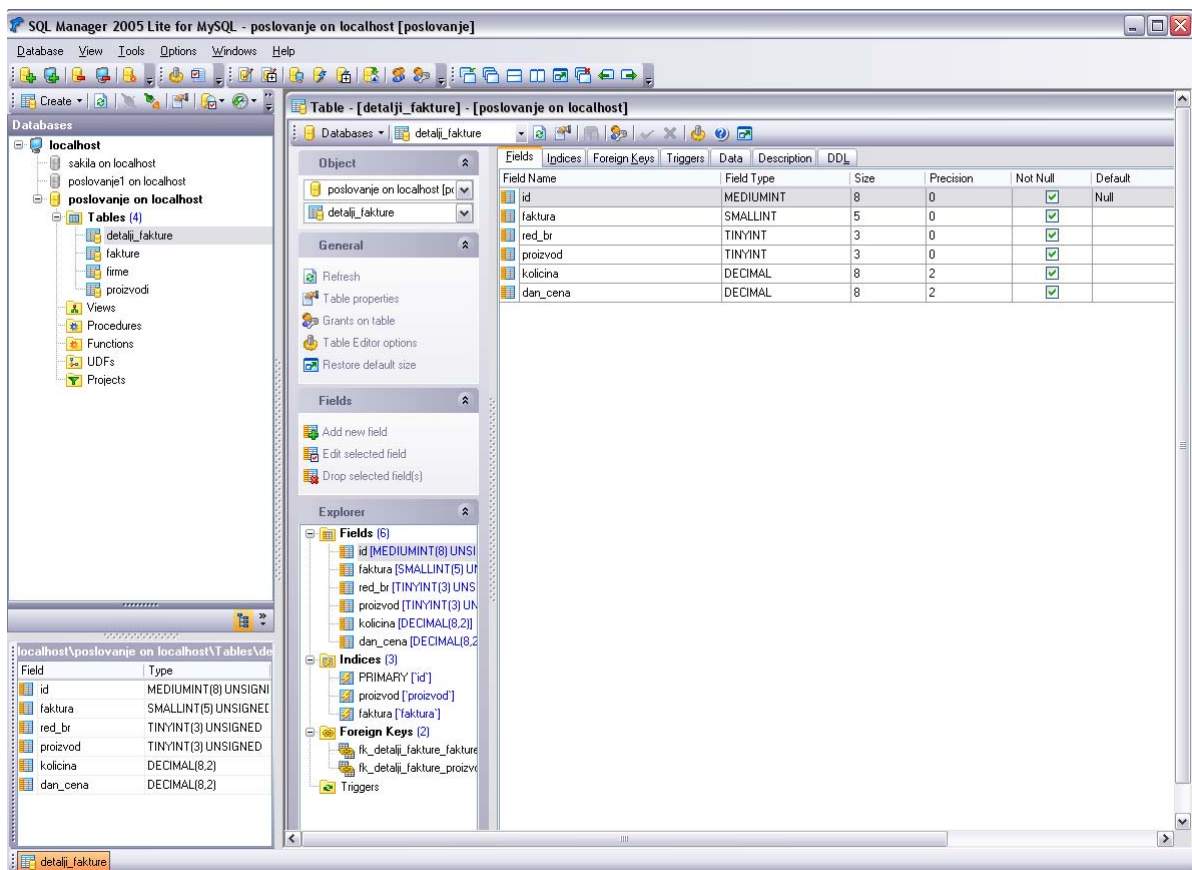
Inicijalno postoje dve mogućnosti: Kreiranje nove baze podataka (**Create new database**) ili korišćenje postojećih baza podataka na serveru (**Manage existing database(s)**). Izborom opcije **Manage existing database(s)** pokreće se čarobnjak za registrovanje baze podataka. Prva stranica služi za definisanje parametara za konekciju. Ovde treba zadržati ponuđene parametre.



Pritiskom na dugme **Next** prelazi se na sledeću stranicu čarobnjaka. Na ovoj stranici je potrebno izabrati željenu bazu podataka iz padajuće liste **Database name** i nakon toga pritisnuti na dugme **Finish**.



U okviru **Databases** sa leve strane prozora treba kliknuti desnim tasterom miša na novokreiranu konekciju i izabrati opciju **Connect to Database**. Nakon uspostavljene konekcije dvostrukim pritiskom na bilo koji od objekata baze podataka u okviru Databases otvara se prozor sa desne strane u kome se mogu videti detljne informacije za taj objekat podeljene po karticama.



4. Struktura jezika

Ovde će biti pomenuta pravila za pisanje sledećih elemenata SQL iskaza prilikom korišćenja MySQL-a:

- Literarne vrednosti, kao što su stringovi i brojevi
- Identifikatori, kao što su nazivi baza, tabela i kolona
- Komentari
- Rezervisane reči

4.1. Literarne vrednosti

U literarne vrednosti spadaju stringovi, brojevi, heksadecimalne vrednosti, boolean vrednosti i NULL.

4.1.1 Stringovi

String je niz karaktera ograđen bilo jednostrukim navodnicima ('), bilo dvostrukim navodnicima ("). Primeri:

'string'

"jos jedan string"

U okviru stringa određene sekvence imaju specijalno značenje. Svaka od ovih sekvenci počinje backslash karakterom (\) koji je poznat kao escape karakter. MySQL prepoznaje sledeće escape sekvence:

\'	Jednostruki navodnik (')
\"	Dvostruki navodnik (")
\b	Backspace karakter
\n	Prelazak u novu liniju
\t	Tab karakter
\\	Backslash karakter (\)

Za sve ostale escape sekvence backslash karakter se ignoriše. Na primer, \x se interpretira kao x.

Ove sekvence su osetljive na velika i mala slova. Na primer, \b se interpretira kao backspace, dok se \B tretira kao B.

Postoji više načina uključivanja navodnika unutar stringa:

- Jednostruki navodnik unutar stringa koji je ograničen jednostrukim navodnicima može biti napisan kao ''
- Dvostruki navodnik unutar stringa koji je ograničen dvostrukim navodnicima može biti napisan kao ""
- Korišćenjem escape sekvenci
- Jednostruki navodnik unutar stringa koji je ograničen dvostrukim navodnicima ne zahteva poseban tretman (ne zahteva dupliranje ili pisanje escape sekvence). Isto tako dvostruki navodnik unutar stringa koji je ograničen jednostrukim navodnicima ne zahteva poseban tretman.

4.1.2. Brojevi

Celi brojevi se predstavljaju nizom cifara. Realni brojevi koriste tačku (.) kao decimalni separator. Bilo koji tip brojeva može imati znak + ili znak - na početku radi označavanja pozitivnih ili negativnih vrednosti respektivno.

Primeri ispravno napisanih celih brojeva:

1232

0

-31

Primeri ispravno napisanih realnih brojeva:

128.48

-35.785e+10

120.00

Celi brojevi se mogu koristiti u izračunavanjima sa realnim brojevima i tada se tretiraju kao ekvivalentni realni brojevi.

4.1.3. Boolean vrednosti

Konstantama TRUE i FALSE odgovaraju vrednosti 1 i 0 respektivno. Nazivi konstanti se mogu pisati i velikim i malim slovima.

```
mysql> SELECT TRUE, true, FALSE, false;
```

```
-> 1, 1, 0, 0
```

4.1.4. NULL vrednosti

NULL vrednost znači "nema podataka". NULL se može pisati i velikim i malim slovima.

Treba imati na umu da je NULL vrednost različita od vrednosti 0 za numeričke tipove podataka ili praznog stringa za string tipove podataka.

4.2. Nazivi baza, tabela, indeksa, kolona i alias-i

Nazivi baza, tabela, indeksa, kolona i alias-i su identifikatori.

Sledeća tabela definiše maksimalne dužine za svaki tip identifikatora.

Identifikator	Maksimalna dužina
Baza podataka	64
Tabela	64
Kolona	64
Indeks	64
Alias	255

Postoje određena ograničenja po pitanju karaktera koji se mogu pojavljivati u identifikatorima:

- Dozvoljeno je koristiti navodnike u identifikatorima, ali je najbolje to izbegavati
- Nazivi baza, tabela i kolona nebi trebalo da se završavaju space karakterom
- Nazivi baza ne mogu sadržati karaktere / \ . ili karaktere koji nisu dozvoljeni u Windows-u prilikom davanja naziva folderima
- Nazivi tabela ne mogu sadržati karaktere / \ . ili karaktere koji nisu dozvoljeni u Windows-u prilikom davanja naziva fajlovima

Identifikator može biti sa navodnicima ili bez navodnika. Ako je identifikator rezervisana reč ili sadrži specijalne karaktere korisnik mora da piše identifikator sa navodnicima svaki put kada ga koristi. Znak navoda za identifikator je znak backtick (`). (Izuzetak: reč koja ide nakon tačke u kvalifikatoru mora biti identifikator, pa ne mora biti pod navodnicima ako je rezervisana reč). Specijalni karakteri su oni koji se ne nalaze u skupu alfanumeričkih karaktera trenutnog seta karaktera, _ i \$.

```
mysql> SELECT * FROM `select` WHERE `select`.id > 100;
```

4.2.1. Kvalifikatori identifikatora

MySQL dozvoljava nazive koji se sastoje od jednog identifikatora ili od više identifikatora. Ako naziv ima više identifikatora neophodno je da oni budu razdvojeni tačkom.

U MySQL kolona se može referencirati na jedan od sledećih načina:

Referenca kolone	Značenje
<i>col_name</i>	Kolona <i>col_name</i> iz bilo koje tabele korišćene u iskazu koja ima kolonu sa tim nazivom
<i>tbl_name.col_name</i>	Kolona <i>col_name</i> iz tabele <i>tbl_name</i> podrazumevane baze podataka
<i>db_name.tbl_name.col_name</i>	Kolona <i>col_name</i> iz tabele <i>tbl_name</i> baze podataka <i>db_name</i>

4.2.2. Osetljivost identifikatora na velika i mala slova

U MySQL-u bazama podataka na serveru odgovaraju folderi u okviru data foldera (C:\Program Files\MySQL\MySQL Server 5.0\data). Svakoj tabeli baze odgovara barem jedan fajl u folderu baze. Prema tome osetljivost na velika i mala slova kod operativnog sistema određuje osetljivost na velika i mala slova kod naziva baza i tabela. Ovo znači da nazivi baza i tabela nisu osetljivi na velika i mala slova u Windows-u.

Iako nazivi baza i tabela nisu osetljivi na velika i mala slova nebi trebalo na različitim mestima u okviru istog iskaza u nazivu baze ili tabele koristiti različita slova (velika ili mala).

```
mysql> SELECT * FROM moja_tabela WHERE MOJA_TABELA.kol=1;
```

Nazivi kolona i indeksa i aliasi kolona nisu osetljivi na velika i mala slova. Nazivi okidača su osetljivi na velika i mala slova.

Alias tabele nisu osetljivi na velika i mala slova.

Najbolje je usvojiti doslednu konvenciju, kao što je da se prilikom kreiranja i referenciranja baza i tabela koriste nazivi sa malim slovima.

4.3. Sintaksa komentara

MySQL server podržava tri stila za komentare:

- Od karaktera # do kraja linije
- Od sekvence -- do kraja linije. Neophodno je da nakon druge crte postoji barem jedan razmak ili kontrolni karakter (space, tab, nova linija, itd.).
- Od sekvence /* do sledeće /* sekvence. Ova sintaksa dozvoljava komentar u više redova, pošto početna i krajnja sekvenca ne moraju da budu u istom redu.

Sledi primer koji demonstrira sva tri stila:

```
mysql> SELECT 1+1; # Ovaj komentar se nastavlja do kraja linije
```

```
mysql> SELECT 1+1; -- Ovaj komentar se nastavlja do kraja linije
```

```
mysql> SELECT 1 /* ovo je komentar u liniji */ + 1;
```

```
mysql> SELECT 1+
```

```
/*
```

```
ovo je
```

```
komentar u više linija
```

```
*/
```

```
1;
```

4.4. Tretman rezervisanih reči u MySQL-u

U rezervisanim rečima jezika SQL ne pravi se razlika između malih i velikih slova. To je standardno u svim sistemima za rad sa bazama podataka.

Dozvoljeno je da se nazivi funkcija koriste kao identifikatori. Na primer, ABS je prihvatljivo kao naziv kolone. Međutim, treba znati da prilikom pozivanja funkcija nije dozvoljeno postojanje razmaka između naziva funkcije i otvorene zagrade koja sledi.

```
mysql> CREATE TABLE abs (val INT);
```

```
mysql> CREATE TABLE abs(val INT);
```

U drugom slučaju će se javiti greška u sintaksi pošto iskaz onda pokušava da pozove funkciju ABS().

Većina rezerviranih reči su zabranjene u standardnom SQL-u za nazive kolona i tabela.

5. Konvencije koje će biti korišćene u ovom materijalu

Rezervisane reči u SQL-u nisu osetljive na velika i mala slova. U ovom materijalu će rezervisane reči biti pisane velikim slovima.

U opisu sintaksi uglaste zagrade ([]) ukazuju na opcione reči ili klauzule.

```
DROP TABLE [IF EXISTS] tbl_name;
```

Kada element sintakse sadrži veći broj alternativa, alternative su razdvojene vertikalnim linijama (|). Kada jedna od alternativa može biti izabrana, alternative su smeštene unutar uglastih zagrada. Kada jedna od alternativa mora biti izabrana, alternative su smeštene unutar vitičastih zagrada ({}).

(...) ukazuje na izostavljanje dela iskaza, sa namerom da se prosledi kraća verzija kompleksnije sintakse. (...) takođe ukazuje da se prethodni element sintakse iskaza može ponavljati.

6. Data Definition Language (DDL)

Jezik za definisanje podataka (*Data Definition Language (DDL)*) omogućava formiranje strukture baze podataka.

6.1. Pravljenje baze podataka – CREATE DATABASE sintaksa

Nakon projektovanja strukture baze podataka, naredni korak je, sasvim logično, izdavanje komande MySQL-u za pravljenje nove baze podataka. To se radi pomoću SQL-ove komande CREATE DATABASE, na sledeći način:

```
CREATE DATABASE [IF NOT EXISTS] db_name;
```

CREATE DATABASE kreira bazu podataka sa datim nazivom. Ako već postoji baza podataka sa datim nazivom, a nije upotrebljeno IF NOT EXISTS javiće se greška.

Kreiranje baze podataka poslovanje:

```
CREATE DATABASE poslovanje;
```

Ako korisnik želi da proveriti da li je ova komanda uspešno izvršena može da izda komandu:

```
SHOW DATABASES;
```

Trebalo bi da se naziv nove baze podataka pojavi u spisku baza podataka na serveru.

Nazive baza podataka je najbolje pisati malim slovima. U nazivima baza ne treba koristiti srpske karaktere. U nazivima baza ne treba koristiti razmake. Ako se želi efekat razmaka treba koristiti donju crtu (_).

Sada na serveru postoji prazna baza podataka.

6.2. Biranje baze podataka – USE sintaksa

```
USE db_name;
```

USE *db_name* iskaz kaže MySQL-u da koristi bazu sa datim nazivom kao podrazumevanu (tekuću) bazu za iskaze koji slede. Baza sa datim nazivom ostaje podrazumevana do kraja sesije ili dok se ne prosledi neki drugi USE iskaz.

Proglašavanje neke baze podrazumevanom (tekućom) pomoću USE islaza ne sprečava korisnika da pristupa tabelama iz ostalih baza podataka na serveru.

6.3. Brisanje baze podataka – DROP DATABASE sintaksa

DROP DATABASE [IF EXISTS] *db_name*;

DROP DATABASE briše sve tabele u bazi podataka sa datim nazivom i nakon toga briše i samu bazu podataka. Treba biti jako oprezan sa ovim iskazom!

IF EXISTS se koristi da bi se sprečila pojava greške ako na serveru ne postoji baza podataka sa datim nazivom.

DROP DATABASE vraća broj tabela koje su bile obrisane.

6.4. Pravljenje tabele - CREATE TABLE sintaksa

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] *tbl_name*
(*create_definition*,...)
[*table_option* ...]

ili:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] *tbl_name*
[(*create_definition*,...)]
[*table_option* ...]
select_statement

ili:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] *tbl_name*
{ LIKE *old_tbl_name* | (LIKE *old_tbl_name*) }

create_definition:

column_definition

| [CONSTRAINT [*symbol*]] PRIMARY KEY [*index_type*] (*index_col_name*,...)

| {INDEX|KEY} [*index_name*] [*index_type*] (*index_col_name*,...)

| [CONSTRAINT [*symbol*]] UNIQUE [INDEX|KEY]

[*index_name*] [*index_type*] (*index_col_name*,...)

| {FULLTEXT|SPATIAL} [INDEX|KEY] [*index_name*] (*index_col_name*,...)

| [CONSTRAINT [*symbol*]] FOREIGN KEY

[*index_name*] (*index_col_name*,...) [*reference_definition*]

| CHECK (*expr*)

column_definition:

col_name data_type [NOT NULL | NULL] [DEFAULT *default_value*]

[AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]

[COMMENT '*string*'] [*reference_definition*]

index_col_name:

col_name [(*length*)] [ASC | DESC]

index_type:

USING {BTREE | HASH}

reference_definition:

REFERENCES *tbl_name* [(*index_col_name*,...)]

[MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]

[ON DELETE *reference_option*]

[ON UPDATE *reference_option*]

reference_option:

RESTRICT | CASCADE | SET NULL | NO ACTION

table_option:

{ENGINE|TYPE} [=] *engine_name*
| AUTO_INCREMENT [=] *value*
| AVG_ROW_LENGTH [=] *value*
| [DEFAULT] CHARACTER SET *charset_name*
| CHECKSUM [=] {0 | 1}
| COLLATE *collation_name*
| COMMENT [=] '*string*'
| CONNECTION [=] '*connect_string*'
| DATA DIRECTORY [=] '*absolute path to directory*'
| DELAY_KEY_WRITE [=] {0 | 1}
| INDEX DIRECTORY [=] '*absolute path to directory*'
| INSERT_METHOD [=] { NO | FIRST | LAST }
| MAX_ROWS [=] *value*
| MIN_ROWS [=] *value*
| PACK_KEYS [=] {0 | 1 | DEFAULT}
| PASSWORD [=] '*string*'
| ROW_FORMAT [=]
{DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}
| UNION [=] (*tbl_name* [, *tbl_name*]...)

select_statement:

[IGNORE | REPLACE] [AS] SELECT ...

CREATE TABLE kreira tabelu sa datim nazivom. Tabela se kreira u podrazumevanoj (tekućoj) bazi. Greška se javlja ako tabela postoji, ako ne postoji podrazumevana (tekuća) baza ili ako baza ne postoji.

Naziv tabele može biti specificiran kao *db_name.tbl_name* da bi se kreirala tabela u određenoj bazi podataka. Ovo funkcioniše nezavisno od toga da li postoji podrazumevana (tekuća) baza, pod uslovom da baza sa datim nazivom postoji.

Prilikom kreiranja tabele se može koristiti TEMPORARY ključna reč. Tabela kreirana sa ovom ključnom reči je vidljiva samo u tekućoj konekciji (sesiji) i biva automatski obrisana nakon prekida konekcije.

Ključna reč IF NOT EXISTS sprečava pojavljivanje greške ako tabela sa istim nazivom postoji.

Nazive tabela je najbolje pisati malim slovima. U nazivima tabela ne treba koristiti srpske karaktere. U nazivima tabela ne treba koristiti razmake. Ako se želi efekat razmaka treba koristiti donju crtu (_).

Pomoću odredbe LIKE *old_tbl_name* može se napraviti nova tabela koja ima istu šemu kao neka druga tabela.

U komandi CREATE TABLE deklarišu se unutar zagrada potrebne kolone, njihovi tipovi podataka i druge informacije koje se tiču strukture tabele. Najjednostavnija definicija kolone sastoji se samo od naziva kolone i tipa podataka u koloni.

Komanda CREATE TABLE se može završiti komandom SELECT. SELECT je SQL-ova komanda koja omogućava učitavanje redova iz jedne ili više tabela. Pomoću ove opcije se može napuniti nova tabela podacima koje učitava zadata komanda SELECT.

6.4.1. Tipovi podataka

data_type predstavlja tip podataka prilikom definisanja kolona tabele.

MySQL podržava veliki broj tipova podataka koji se mogu razvrstati u više kategorija: numerički tipovi, tipovi za datum i vreme i znakovni ili tekstualni tipovi.

Numerički tipovi se koriste za skladištenje brojeva.

BIT[(*M*)]

Služi za definisanje binarnog tipa podataka. *M* definiše broj bitova i može biti od 1 do 64. Ukoliko se *M* izostavi podrazumeva se da je 1.

TINYINT[(*M*)] [UNSIGNED] [ZEROFILL]

Veoma mali celi brojevi. SIGNED opseg je od -128 do 127. UNSIGNED opseg je od 0 do 255.

M predstavlja širinu prikazivanja za celobrojne tipove podataka. Širina prikazivanja se koristi za prikazivanje celobrojnih vrednosti čija je širina manja od specificirane širine za kolonu dodavanjem space karaktera (razmaka) sa leve strane. Širina prikazivanja ne ograničava opseg vrednosti koji se može prikazati u koloni. Kada se koristi zajedno sa opcionim atributom ZEROFILL vrši se dodavanje nula (vodeće nule) a ne space karaktera sa leve strane. Na primer, ako je kolona deklarirana sa INT(5) ZEROFILL, vrednost 4 se predstavlja sa 00004.

Svi celobrojni tipovi podataka mogu imati opcionu atribut UNSIGNED. Kada se navede ovaj atribut moguće je unositi samo nenegativne brojeve, a gornja granica dozvoljenog opsega biva povećana, pri čemu veličina opsega ostaje ista.

Ako se za kolonu sa numeričkim tipom podataka navede atribut ZEROFILL, MySQL automatski dodaje UNSIGNED atribut.

BOOL, BOOLEAN

Ovi tipovi su sinonimi za TINYINT(1). false (netačno) odgovara vrednosti 0. true (tačno) odgovara bilo kojoj vrednosti različitoj od nule.

SMALLINT[(*M*)] [UNSIGNED] [ZEROFILL]

Mali celi brojevi. SIGNED opseg je od -32768 do 32767. UNSIGNED opseg je od 0 do 65535.

MEDIUMINT[(*M*)] [UNSIGNED] [ZEROFILL]

Srednje veliki celi brojevi. SIGNED opseg je od -8388608 do 8388607. UNSIGNED opseg je od 0 do 16777215.

INT[(*M*)] [UNSIGNED] [ZEROFILL]

Normalni celi brojevi. SIGNED opseg je od -2147483648 do 2147483647. UNSIGNED opseg je od 0 do 4294967295.

INTEGER[(*M*)] [UNSIGNED] [ZEROFILL]

Ovaj tip je sinonim za INT.

BIGINT[(*M*)] [UNSIGNED] [ZEROFILL]

Veliki celi brojevi. SIGNED opseg je od -9223372036854775808 do 9223372036854775807. UNSIGNED opseg je od 0 do 18446744073709551615.

Aproksimirani numerički tipovi su FLOAT i DOUBLE.

FLOAT[(*M,D*)] [UNSIGNED] [ZEROFILL]

Mali (jednostruka preciznost) broj sa pokretnim zarezom. Dozvoljene vrednosti su od -3.402823466E+38 do -1.175494351E-38, 0 i od 1.175494351E-38 do 3.402823466E+38.

M je ukupan broj cifara, a *D* je broj cifara iza decimalne tačke. Ako se *M* i *D* izostave vrednosti se smeštaju u skladu sa hardverskim ograničenjima. Broj sa pokretnim zarezom jednostruke preciznosti je tačan do približno 7 decimalnih mesta.

Korišćenje ovog tipa podataka može dovesti do neočekivanih problema pošto se sva izračunavanja u MySQL-u rade sa dvostrukom preciznošću.

DOUBLE[(*M,D*)] [UNSIGNED] [ZEROFILL]

Normalan (dvostruka preciznost) broj sa pokretnim zarezom. Dozvoljene vrednosti su od 1.7976931348623157E+308 do -2.2250738585072014E-308, 0 i od 2.2250738585072014E-308 do 1.7976931348623157E+308.

M je ukupan broj cifara, a *D* je broj cifara iza decimalne tačke. Ako se *M* i *D* izostave vrednosti se smeštaju u skladu sa hardverskim ograničenjima. Broj sa pokretnim zarezom dvostruke preciznosti je tačan do približno 15 decimalnih mesta.

DOUBLE PRECISION[(*M,D*)] [UNSIGNED] [ZEROFILL], REAL[(*M,D*)] [UNSIGNED] [ZEROFILL]

Ovi tipovi su sinonimi za DOUBLE.

Tačni numerički tipovi su tip INTEGER i njegove varijante i DECIMAL.

DECIMAL[(*M,D*)] [UNSIGNED] [ZEROFILL]

Broj sa fiksnim zarezom. *M* je ukupan broj cifara (preciznost), a *D* je broj cifara iza decimalne tačke (the scale). Decimalna tačka i (za negativne brojeve) znak – se ne računaju u *M*. Ako je *D* jednako nuli brojevi nemaju decimalnu tačku i razlomljeni deo. Maksimalan ukupan broj cifara (*M*) za ovaj tip podataka je 65. Maksimalan broj cifara iza decimalne tačke (*D*) za ovaj tip podataka je 30. Ako se *D* izostavi, podrazumevana vrednost je 0. Ako se *M* izostavi, podrazumevana vrednost je 10.

Sva osnovna izračunavanja (+, -, *, /) sa ovim tipom podataka se rade sa preciznošću od 65 cifara.

Ovaj tip podataka se obično koristi za rad sa novčanim vrednostima.

DEC[(*M,D*)] [UNSIGNED] [ZEROFILL], NUMERIC[(*M,D*)] [UNSIGNED] [ZEROFILL], FIXED[(*M,D*)] [UNSIGNED] [ZEROFILL]

Ovi tipovi su sinonimi za DECIMAL.

DATE

Datum. Podržani opseg je od '1000-01-01' do '9999-12-31'. MySQL prikazuje vrednosti za ovaj tip podataka u formatu 'YYYY-MM-DD'.

DATETIME

Kombinacija datuma i vremena. Podržani opseg je od '1000-01-01 00:00:00' do '9999-12-31 23:59:59'. MySQL prikazuje vrednosti za ovaj tip podataka u formatu 'YYYY-MM-DD HH:MM:SS'.

TIMESTAMP

Ovo je koristan tip podataka za kolone tabelle. Ako se u određenom redu ne zada vrednost za kolonu ovog tipa u kolonu se upisuje vreme kada je red dodat tabeli ili kada je poslednji put izmenjen sadržaj reda.

Opseg je od '1970-01-01 00:00:01' UTC do 2037. godine. **TIMESTAMP** vrednosti su smeštene kao broj sekundi proteklih od epohe ('1970-01-01 00:00:00' UTC). **TIMESTAMP** vrednost se vraća kao string sa formatom 'YYYY-MM-DD HH:MM:SS' i sa širinom prikazivanja fiksiranom na 19 karaktera.

TIME

Vreme. Podržani opseg je od '-838:59:59' do '838:59:59'. MySQL prikazuje vrednosti za ovaj tip podataka u formatu 'HH:MM:SS'.

YEAR[(2|4)]

Godina kao dvocifreni ili četvorocifreni broj. Podrazumevani format je četvorocifreni broj. Kod četvorocifrenog formata podržani opseg je od 1901 do 2155 i 0000. Kod dvocifrenog formata podržani opseg je od 70 do 69, što odgovara godinama od 1970. do 2069. MySQL prikazuje vrednosti za ovaj tip podataka u formatu YYYY.

[NATIONAL] CHAR(*M*) [CHARACTER SET *charset_name*] [COLLATE *collation_name*]

String fiksne dužine kome prilikom smeštanja uvek bivaju dodati space karakteri sve do specificirane dužine. *M* predstavlja dužinu, odnosno broj znakova. Opseg za *M* je od 0 do 255 karaktera.

Kada se podatak tipa CHAR upiše u kolonu tabele, on uvek ima dužinu koja je zadata u definiciji kolone. To se postiže dopunjavanjem podatka u koloni razmacima. Ti razmaci se automatski uklanjaju pri učitavanju podatka iz kolone tipa CHAR.

Očigledno je da podaci tipa CHAR zauzimaju više prostora na disku od ekvivalentnih znakovnih vrednosti promenljive dužine. Prednost im je što se podaci brže učitavaju iz tabele čije su sve kolone fiksne dužine (recimo CHAR ili DATE).

Deklaracijama tipova CHAR i VARCHAR može slediti rezervisana reč BINARY, što znači da se pri poređenju znakovnih vrednosti pravi razlika između malih i velikih slova. Podrazumevani način poređenja je da se ta razlika ne pravi.

CHAR je skraćenica za CHARACTER. NATIONAL CHAR (ili odgovarajuća skraćenica NCHAR) je standardni način da se u SQL-u definiše da kolona sa tipom podataka koristi neki unapred definisani skup karaktera.

CHARACTER SET atribut specificira skup karaktera koji se koristi. COLLATE atribut specificira uparivanje za taj skup karaktera. Skup karaktera je skup simbola i šifriranja. Uparivanje je skup pravila za poređenje karaktera u skupu karaktera.

CHARSET je sinonim za CHARACTER SET.

CHAR [CHARACTER SET *charset_name*] [COLLATE *collation_name*]

Ovo je sinonim za CHAR(1).

[NATIONAL] VARCHAR(*M*) [CHARACTER SET *charset_name*] [COLLATE *collation_name*]

String varijabilne dužine. *M* predstavlja maksimalnu dužinu podataka. Opseg za *M* je od 0 do 65535 karaktera. VARCHAR je sinonim za CHARACTER VARYING.

BINARY(*M*)

BINARY tip podataka je sličan CHAR tipu podataka, ali smešta binarne stringove umesto nebinarnih stringova sa karakterima.

VARBINARY(*M*)

VARBINARY tip podataka je sličan VARCHAR tipu podataka, ali smešta binarne stringove umesto nebinarnih stringova sa karakterima.

TINYBLOB

BLOB kolona sa maksimalnom dužinom od 255 bajtova. BLOB je veliki binarni objekat koji može sadržati promenljivu količinu podataka. BLOB kolone se tretiraju kao binarni stringovi.

TINYTEXT [CHARACTER SET *charset_name*] [COLLATE *collation_name*]
TEXT kolona sa maksimalnom dužinom od 255 karaktera.

BLOB(*M*)

BLOB je skraćenica od Binary Large Object (veliki binarni objekat). BLOB kolona može da ima maksimalnu dužinu od 65535 bajta.

Za ovaj tip se može dati opcionalna dužina *M*. Ako se definiše MySQL kreira kolonu koja je najmanji BLOB tip dovoljno veliki da prima vrednosti koje su *M* bajtova dugačke.

TEXT(*M*) [CHARACTER SET *charset_name*] [COLLATE *collation_name*]

TEXT kolona sa maksimalnom dužinom od 65535 karaktera.

Za ovaj tip se može dati opcionalna dužina *M*. Ako se definiše MySQL kreira kolonu koja je najmanji TEXT tip dovoljno veliki da prima vrednosti koje su *M* karaktera dugačke.

Omogućava skladištenje tekstualnih podataka dužih od onog što može da stane u tipove CHAR i VARCHAR.

MEDIUMBLOB

BLOB kolona sa maksimalnom dužinom od 16777215 bajtova (16 MB).

MEDIUMTEXT [CHARACTER SET *charset_name*] [COLLATE *collation_name*]

TEXT kolona sa maksimalnom dužinom od 16777215 karaktera (16 MB).

LOB

BLOB kolona sa maksimalnom dužinom od 4294967295 bajtova (4 GB).

LONGTEXT [CHARACTER SET *charset_name*] [COLLATE *collation_name*]

TEXT kolona sa maksimalnom dužinom od 4294967295 karaktera (4GB).

ENUM('value1','value2',...) [CHARACTER SET *charset_name*]
[COLLATE *collation_name*]

Nabrajanje. Ovaj tip podataka omogućava zadavanje liste mogućih vrednosti. Kolona tabele može sadržati samo jednu vrednost, izabranu iz liste vrednosti 'value1','value2',..., NULL ili specijalna ' ' greška vrednost. ENUM kolona može imati maksimalno 65535 različitih vrednosti. ENUM vrednosti su interno predstavljene kao celi brojevi.

pol ENUM ('m', 'ž')

SET('value1','value2',...) [CHARACTER SET *charset_name*] [COLLATE *collation_name*]

Skup. Kolona tabele može sadržati jednu ili više vrednosti, a svaka od njih mora biti izabrana iz liste vrednosti 'value1','value2',... SET kolona može imati maksimalno 64 člana. SET vrednosti su interno predstavljene kao celi brojevi.

6.4.2. Ostali atributi

- Svaka kolona se može deklarirati kao NOT NULL ili NULL, što znači da se ne dozvoljava da kolona sadrži vrednost NULL (opcija NOT NULL) ili da se prihvata vrednost NULL (opcija NULL). Ako nisu specificirani ni NULL ni NOT NULL atribut, kolona se tretira kao da je NULL atribut specificiran.

- Kolone sa celobrojnim tipom podataka mogu imati dodatni atribut AUTO_INCREMENT. Kada se u indeksiranu AUTO_INCREMENT kolonu unese vrednost NULL (preporučeno) ili vrednost 0, vrednost u koloni se podešava na sledeću vrednost u nizu. Tipično to je *value+1* gde je *value* najveća vrednost u koloni trenutno u tabeli. AUTO_INCREMENT nizovi počinju brojem 1. Može postojati samo jedna AUTO_INCREMENT kolona po tabeli, mora biti indeksirana i ne može imati podrazumevanu (DEFAULT) vrednost. AUTO_INCREMENT funkcioniše ispravno ako sadrži samo pozitivne vrednosti.
- DEFAULT klauzula specificira podrazumevanu vrednost za kolonu. Sa jednim izuzetkom, podrazumevana vrednost mora biti konstanta; ne može biti funkcija ili izraz. Izuzetak je da se može specificirati CURRENT_TIMESTAMP kao podrazumevana vrednost za TIMESTAMP kolonu. Kolone tipa BLOB i TEXT ne mogu imati definisanu podrazumevanu vrednost.
- Komentar za kolonu može biti specificiran pomoću COMMENT opcije i može imati dužinu do 255 karaktera

6.4.3. Ograničenja (Constraints)

Ako se redovno sortiraju podaci u tabeli po jednoj istoj koloni ili kolonama potrebno je napraviti indeks za tu kolonu ili te kolone. Indeks je interna tabela vrednosti koja održava red, odnosno redosled zapisa. Na ovaj način, kada je potrebno da se sortiraju podaci ili kada je potrebno da se određeni podatak pronađe brzo MySQL može da vrši pretraživanje po indeks ključevima po poznatom redosledu, što je bolje od sekvencijalnog pretraživanja po podacima.

Pravljenje indeksa usporava unošenje podataka; svaki novi zapis, obrisani zapis ili promena vrednosti u indeksiranom polju zahteva promenu indeksa. Indeksirane kolone treba koristiti samo onda kada su stvarno neophodne.

KEY (ključ) je normalno sinonim za INDEX (indeks) i znači da će zadata kolona (ili kolone) biti indeksirana. Indeksi se koriste da bi se brzo pronašli redovi sa određenom vrednošću u koloni. Atribut PRIMARY KEY se može specificirati i kao KEY kada je dat u definiciji kolone.

UNIQUE (jedinstveni) indeks kreira ograničenje takvo da sve vrednosti u indeksu (odnosno koloni koja je indeksirana) moraju biti različite. Ukoliko korisnik pokuša da doda novi red u tabelu koji ima istu vrednost u koloni za koju je definisan UNIQUE indeks kao neki drugi red javiće se greška. Ovo ograničenje se ne odnosi na NULL vrednosti sem za BDB mašine za smeštanje podataka. Za ostale mašine za smeštanje podataka UNIQUE indeks dozvoljava višestruke NULL vrednosti za kolone koje mogu sadržati NULL vrednosti.

PRIMARY KEY (primarni ključ) je jedinstveni indeks gde sve kolone ključa moraju biti definisane kao NOT NULL. Ako nisu eksplicitno deklarisan kao NOT NULL, MySQL ih deklariše implicitno. Tabela može imati samo jedan primarni ključ. Na ovaj način se postavlja ograničenje nad tabelom. To znači da se upisom novog zapisa u kolonu koja je primarni ključ ne sme upisati vrednost koja već postoji. Za kolonu (ili kolone) koja je primarni ključ se automatski formira indeks.

U kreiranoj tabeli primarni ključ je smešten kao prvi, zatim dolaze svi UNIQUE (jedinstveni) indeksi i na kraju svi nejedinstveni indeksi.

Primarni ključ može biti indeks nad više kolona. Međutim, nije moguće kreirati indeks nad više kolona korišćenjem atributa PRIMARY KEY u specifikaciji kolone. Mora se koristiti posebna PRIMARY KEY(*index_col_name*, ...) klauzula.

U MySQL-u naziv primarnog ključa je PRIMARY. Za ostale indekse, ako im se ne dodeli naziv, indeksu se daje naziv prve indeksirane kolone, zajedno sa opcionim sufiksom (*_2*, *_3*, ...) da bi bio jedinstven.

Za mašine za skladištenje koje nisu tipa InnoDB je moguće prilikom definisanja kolone koristiti REFERENCES *tbl_name(col_name)* koja nema stvarni efekat i služi samo kao podsetnik ili komentar korisniku da se namerava da kolona koja se trenutno definiše ukazuje na kolonu neke druge tabele.

Neke mašine za smeštanje podataka dozvoljavaju definisanje tipa indeksa prilikom kreiranja indeksa.

Specifikacija za *index_col_name* se može završavati sa ASC ili DESC. Ove ključne reči su dozvoljene za buduće ekstenzije za specificiranje smeštanja vrednosti u indeksu po rastućem ili opadajućem redosledu. Trenutno se ove ključne reči ignorišu; vrednosti u indeksu su uvek smeštene po rastućem redolsledu.

Moguće je kreirati specijalne FULLTEXT indekse koji se koriste za Full-text (tekstualna) pretraživanja. Samo MyISAM mašina za smeštanje podataka podržava FULLTEXT indekse. Oni mogu biti kreirani samo za CHAR, VARCHAR i TEXT kolone.

Moguće je kreirati SPATIAL indekse za prostorne tipove podataka (dozvoljavaju generisanje, smeštanje i analizu geografskih obeležja). Ovi tipovi podataka su podržani samo u MyISAM tabelama i indeksirane kolone moraju biti deklarirane kao NOT NULL.

InnoDB tabele podržavaju ograničenja tipa strani ključ (FOREIGN KEY). Strani ključ je skup kolona (jedna ili više kolona) iz jedne tabele (sekundarna tabela ili dete tabela) čije vrednosti moraju da se 'slažu', odnosno da odgovaraju vrednostima primarnog ključa u nekoj drugoj tabeli (primarna tabela ili roditelj tabela). Definisanjem stranog ključa se u stvari uspostavlja relacija između tabela. Sintaksa za definisanje stranog ključa kod ovog tipa tabela izgleda ovako:

```
[CONSTRAINT symbol] FOREIGN KEY [id] (index_col_name, ...)
REFERENCES tbl_name (index_col_name, ...)
[ON DELETE {RESTRICT | CASCADE | SET NULL | NO ACTION}]
[ON UPDATE {RESTRICT | CASCADE | SET NULL | NO ACTION}]
```

Prilikom definisanja stranog ključa obe tabele moraju biti InnoDB tabele i ne smeju biti privremene (TEMPORARY) tabele.

InnoDB odbija bilo koju INSERT ili UPDATE operaciju koja pokušava da kreira vrednost za strani ključ u dete tabeli ako ne postoji odgovarajuća vrednost ključa u roditelj tabeli. Akcija koju preduzima InnoDB za bilo koju UPDATE ili DELETE operaciju koja pokušava da promeni ili obriše vrednost ključa u roditelj tabeli koji ima povezane redove u dete tabeli zavisi od referencijalne akcije specificirane korišćenjem ON UPDATE i ON DELETE podklauzula u FOREIGN KEY klauzuli. Kada korisnik pokuša da obriše ili promeni red u roditelj tabeli, a postoje jedan ili više povezanih redova u dete tabeli, InnoDB podržava 5 opcija vezano za akciju koja će biti izvršena:

- CASCADE: Brisanjem ili izmenom reda u roditelj tabeli se automatski brišu ili menjaju povezani redovi u dete tabeli.
- SET NULL: Brisanjem ili izmenom reda u roditelj tabeli se vrednost u koloni koja je strani ključ dete tabele za povezane redove postavlja na NULL. Ovo važi samo ako kolone koje su strani ključ nemaju specificiran NOT NULL kvalifikator.
- NO ACTION: U standardnom SQL-u NO ACTION znači da nema akcije u smislu da neće biti moguće obrisati ili promeniti vrednost primarnog ključa u roditelj tabeli ako postoji povezana vrednost stranog ključa u dete tabeli. InnoDB odbija operaciju brisanja ili izmene za roditelj tabelu.
- RESTRICT: Odbija se operacija brisanja ili izmene za roditelj tabelu.
- SET DEFAULT: Ova akcija je prepoznata od strane parsera, ali InnoDB odbija definicije tabela koje sadrže ON DELETE SET DEFAULT ili ON UPDATE SET DEFAULT klauzule.

Odgovarajuće kolone koje čine strani i primarni ključ moraju imati slične tipove podataka da bi mogle da budu poređene bez konverzije podataka. Veličina i znak celobrojnih tipova moraju biti isti. Dužine string tipova ne moraju biti iste.

6.4.4. Mašine za skladištenje i tipovi tabela

MySQL podržava nekoliko mašina za skladištenje koji rade kao manipulatori za različite tipove tabela. Među ovim mašinama postoje one koje rade sa transakcionim i one koje rade sa netransakcionim tabelama:

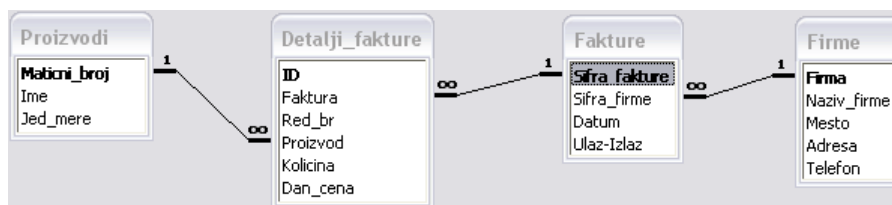
- MyISAM upravlja netransakcionim tabelama. Obezbeđuje visoku brzinu smeštanja i preuzimanja podataka kao i mogućnost fulltext pretraživanja.
- MEMORY mašina za skladištenje radi sa tabelama koje su smeštene u memoriji računara i nikada se ne upisuju na disk. Zahvaljujući tome te tabele su veoma brze, ali veličina im je ograničena i podaci iz njih se ne mogu restaurirati ukoliko se sistem pokvari. MERGE mašina za skladištenje, takođe poznata i kao MRG_MyISAM mašina omogućava rad sa kolekcijom identičnih MyISAM tabela kao sa jednom tabelom. Pod identičnim tabelama se podrazumeva da tabele imaju identične kolone i indekse. To se može iskoristiti kada operativni sistem ograničava maksimalnu veličinu fajlova – pa zbog toga i tabela. MEMORY i MERGE rade sa netransakcionim tabelama.
- InnoDB mašina za skladištenje radi sa transakcionim tabelama i ima mogućnosti izvršavanja, rollback-a i oporavka prilikom havarije. Ova mašina radi zaključavanje na nivou redova prilikom transakcija. Dizajnirana je za maksimum performansi kada obrađuje velike količine podataka. Ova mašina podržava spoljne ključeve. Ovo je podrazumevana mašina za skladištenje.
- ARCHIVE mašina za skladištenje se koristi za smeštanje velikih količina podataka bez indeksa.

Ovo su samo neke od ukupno 11 mašina skladištenje.

6.4.5. Kreiranje tabela u bazi podataka poslovanje

Izabran je primer baze podataka **poslovanje** koja služi za praćenje poslovanja jedne firme koja radi sa određenom grupom proizvoda i saraduje sa određenim brojem drugih firmi.

Baza podataka **poslovanje** ima četiri tabele i to: proizvodi, firme, fakture i detalji_fakture. Na sledećoj slici se mogu videti sve tabele sa nazivima kolona i primarnim ključevima (nazivi kolona ispisani polucrnim slovima), kao i uspostavljene relacije i strani ključevi u odgovarajućim tabelama. Može se videti da su sve uspostavljene relacije tipa jedan-prema-više.



Na sledećoj slici je prikazan primer tabela koje bi trebalo da budu povezane preko stranog ključa. Polje Faktura iz tabele Detalji_fakture može uzimati samo one vrednosti koje se pojavljuju u polju Sifra_fakture tabele Fature.

Detalji_fakture : Table						
ID	Faktura	Red_br	Proizvod	Kolicina	Dan_cena	
1	1	2	3	50,00	240,00	
2	2	1	4	50,00	25,00	
3	2	2	2	36,00	11,00	
4	2	3	3	50,00	23,00	
5	3	1	1	20,00	28,00	
6	3	2	2	35,00	35,00	
7	3	3	3	50,00	100,00	
8	3	4	4	50,00	15,00	
9	4	1	4	50,00	23,00	
10	4	2	2	50,00	18,00	
11	5	1	4	50,00	20,00	
12	5	2	4	50,00	500,00	
13	5	3	2	25,00	24,00	
14	6	1	4	50,00	220,00	
15	6	2	4	50,00	840,00	
16	7	1	2	26,00	470,00	
17	8	1	2	25,00	840,00	
18	8	2	3	50,00	1000,00	
19	6	3	2	24,00	250,00	
20	6	4	3	50,00	230,00	
21	6	5	5	50,00	12,00	
22	1	3	3	50,00	250,00	
23	9	1	3	25,00	26,00	
24	9	2	2	14,00	24,00	
25	10	1	4	150,00	150,00	
26	10	2	5	100,00	100,00	
27	10	3	3	125,00	23,00	
28	10	4	2	100,00	250,00	
29	10	5	3	1,00	1,00	
30	10	5	2	5,00	5,00	
* ber)	0	0	0	0,00	0,00	

Fakture : Table					
Sifra_fakture	Sifra_firme	Datum	Ulaz-Izlaz		
+	1	3	25.10.2006	1	
+	2	4	28.10.2006	1	
+	3	1	25.10.2006	2	
+	4	3	29.10.2006	2	
+	5	3	25.10.2006	1	
+	6	3	6.11.2006	2	
+	7	4	4.11.2006	2	
+	8	2	23.10.2006	1	
+	9	3	2.11.2006	2	
+	10	4	8.10.2006	1	
+	11	3	15.10.2006	1	
* 0	0	0	20.1.2007		

Na sledećoj slici prikazani su i dve preostale tabele baze podataka **poslovanje** sa unetim podacima.

Proizvodi : Table			
Maticni_broj	Ime	Jed_mere	
+	1	Četka	korn.
+	2	Lak	lit.
+	3	Stiropor	m2
+	4	Gips	kg
+	5	Destilovana voda	kg
+	6	Šmirgla	korn.
* 0			

Firme : Table					
Firma	Naziv_firme	Mesto	Adresa	Telefon	
+	1	BALKAN	Niš	Mokranjčeva 13	018-522-854
+	2	STIL	Beograd	Takovska 10	011-562-365
+	3	KOKOMAX	Niš	Dušanova 33	018-352-666
+	4	HELIO	Subotica	Nikole Tesle 55	027-555-125
* 0					

Kreiranje tabele **proizvodi**:

```
CREATE TABLE proizvodi (
  maticni_broj TINYINT(3) UNSIGNED NOT NULL,
  ime VARCHAR(50) NOT NULL,
  jed_mere VARCHAR(20) NOT NULL,
  PRIMARY KEY (maticni_broj)
);
```

Kreiranje tabele **firme**:

```
CREATE TABLE firme (
  firma TINYINT(3) UNSIGNED NOT NULL,
  naziv_firme VARCHAR(100) NOT NULL,
  mesto VARCHAR(50) NOT NULL,
  adresa VARCHAR(50) NOT NULL,
  telefon VARCHAR(20) NOT NULL,
```

```
PRIMARY KEY (firma)  
);
```

Kreiranje tabele **fakture**:

```
CREATE TABLE fakture (  
  sifra_fakture SMALLINT(5) UNSIGNED NOT NULL,  
  sifra_firme TINYINT(3) UNSIGNED NOT NULL,  
  datum DATE NOT NULL,  
  ulaz_izlaz CHAR(1) NOT NULL,  
  PRIMARY KEY (sifra_fakture),  
  KEY k_sifra_firme (sifra_firme),  
  CONSTRAINT fk_fakture_firme FOREIGN KEY (sifra_firme) REFERENCES firme (firma)  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Kreiranje tabele **detalji_fakture**:

```
CREATE TABLE detalji_fakture (  
  id MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,  
  faktura SMALLINT(5) UNSIGNED NOT NULL,  
  red_br TINYINT(3) UNSIGNED NOT NULL,  
  proizvod TINYINT(3) UNSIGNED NOT NULL,  
  kolicina DECIMAL(8,2) NOT NULL,  
  dan_cena DECIMAL(8,2) NOT NULL,  
  PRIMARY KEY (id),  
  KEY k_proizvod (proizvod),  
  KEY k_faktura (faktura),  
  CONSTRAINT fk_detalji_fakture_proizvodi FOREIGN KEY (proizvod) REFERENCES  
  proizvodi (maticni_broj) ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT fk_detalji_fakture_faktura FOREIGN KEY (faktura) REFERENCES fakture  
  (sifra_fakture) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

6.5. Pravljenje indeksa – CREATE INDEX sintaksa

Svi indeksi koji su neophodni najčešće će se praviti automatski prilikom kreiranja tabele. Za svaku kolonu koja se deklariše sa opcijom PRIMARY KEY, KEY, UNIQUE ili INDEX, automatski se formira i indeks. Ako se ustanovi da se koristi veći broj upita koji obuhvataju kolonu za koju nije definisan indeks novi indeks se može dodati pomoću komande CREATE INDEX:

```
CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX index_name  
[index_type]  
ON tbl_name (index_col_name,...)
```

index_col_name:

```
col_name [(length)] [ASC | DESC]
```

index_type:

```
USING {BTREE | HASH}
```

Primer:

```
CREATE INDEX k_naziv_firme ON firme (naziv_firme);
```

Ovde je bio kreiran indeks za kolonu *naziv_firme* tabele *firme*. Naziv indeksa je *k_naziv_firme*.

Zanimljivo je da se, pre izvršavanja, komanda CREATE INDEX preslikava u komandu ALTER TABLE.

Indeksi definisani nad kolonama tipa CHAR, VARCHAR, BINARY i VARBINARY mogu se ograničiti na prvih nekoliko znakova u polju. To se može uraditi tako što se iza imena indeksirane kolone zada između zagrada broj znakova koji želi da se indeksira:

```
CREATE INDEX k_naziv_firme ON firme (naziv_firme(5));
```

Pošto indeksi nad kolonama tekstualnog tipa nisu tako efikasni kao indeksi nad numeričkim kolonama, indeksiranje samo nekoliko početnih znakova poboljšava performanse.

6.6. **Brisanje tabele – DROP TABLE sintaksa**

```
DROP [TEMPORARY] TABLE [IF EXISTS]  
tbl_name [, tbl_name] ...  
[RESTRICT | CASCADE]
```

Komandom je moguće izbrisati više tabela ako se zada lista njihovih imena razdvojenih zarezima.

```
DROP TABLE detalji_fakture; (Ovo je samo primer korišćenja komande. Komandom se briše tabela detalji_fakture.)
```

6.7. **Brisanje indeksa – DROP INDEX sintaksa**

```
DROP INDEX index_name ON tbl_name;
```

Da bi se izbrisao indeks mora se zadati i naziv tabele kojoj je pridružen.

```
DROP INDEX k_naziv_firme ON firme;
```

Ovde je obrisan prethodno kreirani indeks *k_naziv_firme*.

6.8. **Izmena strukture postojeće tabele – ALTER TABLE sintaksa**

```
ALTER [IGNORE] TABLE tbl_name  
alter_specification [, alter_specification] ...
```

alter_specification:

```
ADD [COLUMN] column_definition [FIRST | AFTER col_name ]  
| ADD [COLUMN] (column_definition,...)  
| ADD {INDEX|KEY} [index_name] [index_type] (index_col_name,...)  
| ADD [CONSTRAINT [symbol]]  
PRIMARY KEY [index_type] (index_col_name,...)  
| ADD [CONSTRAINT [symbol]]  
UNIQUE [INDEX|KEY] [index_name] [index_type] (index_col_name,...)  
| ADD [FULLTEXT|SPATIAL] [INDEX|KEY] [index_name] (index_col_name,...)  
| ADD [CONSTRAINT [symbol]]  
FOREIGN KEY [index_name] (index_col_name,...)  
[reference_definition]  
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}  
| CHANGE [COLUMN] old_col_name column_definition
```



```

[FIRST|AFTER col_name]
| MODIFY [COLUMN] column_definition [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX|KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO] new_tbl_name
| ORDER BY col_name
| CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]
| [DEFAULT] CHARACTER SET charset_name [COLLATE collation_name]
| DISCARD TABLESPACE
| IMPORT TABLESPACE
| table_option ...

```

index_col_name:
col_name [(*length*)] [ASC | DESC]

index_type:
USING {BTREE | HASH}

ALTER TABLE omogućava promenu strukture postojeće tabele. Na primer, moguće je dodati ili izbrisati kolone, kreirati ili uništiti indekse, promeniti tip postojećih kolona ili preimenovati kolone ili celu tabelu.

Sintaksa za mnoge dozvoljene promene je slična sintaksi za klauzule CREATE TABLE iskaza.

U većini slučajeva ALTER TABLE radi tako što pravi privremenu kopiju originalne tabele. Izmene se vrše na kopiji, a zatim se originalna tabela briše a privremena tabela se preimenuje u originalnu.

Ako se koristi ALTER TABLE *tbl_name* RENAME TO *new_tbl_name* bez dodatnih opcija MySQL jednostavno preimenuje sve fajlove koji odgovaraju tabeli *tbl_name*. U ovom slučaju nema potrebe za pravljenjem privremene tabele.

Opcije CHANGE i MODIFY zapravo su jedna te ista opcija i omogućavaju izmenu definicije kolone ili njenog mesta u tabeli.

Opcija DROP COLUMN briše kolonu iz tabele, dok opcije DROP PRIMARY KEY i DROP INDEX brišu samo indeks pridružen koloni.

Opcija DISABLE KEYS nalaže MySQL-u da ne ažurira sadržaj nejedinstvenih indeksa, ali upotrebljiva je samo za MyISAM tabele. Opcija ENABLE KEYS uključuje ažuriranje indeksa.

Opcija ORDER BY će poređati zadatim redosledom redove tabele na koju je primenjena. Taj redosled neće biti očuvan kasnije kada se budu dodavali ili brisali podaci.

Primeri:

```

ALTER TABLE fakture
ADD COLUMN primio VARCHAR(50);

```

Ovde se dodaje nova kolona sa nazivom *primio* tabeli *fakture*.

```

ALTER TABLE fakture
DROP COLUMN primio;

```

Ovde se briše prethodno dodata kolona.

```
ALTER TABLE detalji_fakture  
MODIFY COLUMN kolicina DECIMAL(6,2);
```

Ovde se menja definicija kolone *kolicina* tabele *detalji_fakture*.

```
ALTER TABLE detalji_fakture  
MODIFY COLUMN kolicina DECIMAL(8,2);
```

Ovde se ponovo menja definicija kolone *kolicina* tabele *detalji_fakture* u smislu vraćanja na prvobitnu definiciju.

```
ALTER TABLE detalji_fakture  
CHANGE COLUMN dan_cena_cena DECIMAL(10,2);
```

Ovde se menja definicija kolone *dan_cena* tabele *detalji_fakture* tako što se naziv kolone iz *dan_cena* menja u *cena* i pored svega se menjaju podešavanja za tip podataka (u prvobitnoj definiciji je bilo DECIMAL(8,2)).