

Jezik za definisanje podataka (*engl.* DDL)

Projektovanje baza podataka
2020/21.

Sadržaj

- Tipovi podataka
- SQL DDL iskazi
- InnoDB mašina za skladištenje
- Alternativne mašine za skladištenje

Tipovi podataka

- 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 deklarisan 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.

Tipovi podataka

- 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.

Tipovi podataka

- 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.

Tipovi podataka

- Za razliku od aproksimativnih, precizni 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. 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**.

Tipovi podataka

- **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 tabele. 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 '2038-01-09 03:14:07' UTC. **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.

Tipovi podataka

- **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.

Tipovi podataka

- [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.
- Dakle, 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).
- 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.

Tipovi podataka

- **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 do 65535 karaktera. VARCHAR je sinonim za CHARACTER VARYING.
- 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.

Tipovi podataka

- 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. TEXT kolone su slične VARCHAR kolonama, a razlika je u tome što za TEXT kolonu nije moguće definisati podrazumevanu (DEFAULT) vrednost.

Tipovi podataka

- **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.
- **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). Omogućava skladištenje tekstualnih podataka dužih od onog što može da stane u tip VARCHAR.
- **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). Omogućava skladištenje tekstualnih podataka dužih od onog što može da stane u tip VARCHAR.

Tipovi podataka

- **ENUM**('value1','value2',...) [CHARACTER SET charset_name] [COLLATE collation_name] Ovaj tip podataka omogućava nabranje, tj. zadavanje liste mogućih vrednosti. Kolona tabele može sadržati samo jednu vrednost, izabranu iz liste vrednosti 'value1','value2',..., NULL ili specijalnu " 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 listevrednosti 'value1','value2',... SET kolona može imati maksimalno 64 člana. SET vrednosti su interno predstavljene kao celi brojevi.

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

SQL iskazi za definisanje podataka (*engl.* DDL)

- CREATE DATABASE
- CREATE TABLE
- CREATE TRIGGER
- CREATE VIEW
- ALTER DATABASE
- ALTER TABLE
- ALTER VIEW

SQL iskazi za definisanje podataka (*engl.* DDL)

- DROP DATABASE
- DROP TABLE
- DROP TRIGGER
- DROP VIEW
- RENAME TABLE
- TRUNCATE TABLE

SQL iskazi za definisanje podataka (*engl.* DDL)

- Sintaksa:

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

```
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) }
```

Ograničenja

- 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 indeksiranoj koloni zahteva promenu indeksa. Indeksirane kolone treba koristiti samo onda kada su stvarno neophodne.
- **KEY** (ključ) je 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.

Ograničenja

- **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. O mašinama biće reči dole u tekstu. 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.

Ograničenja

- 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.
- Nazivi indeksa tabele se mogu pogledati sa SHOW INDEX FROM tbl_name.
- 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 skladištenje dozvoljavaju definisanje tipa indeksa prilikom kreiranja indeksa.
- Primer:

```
CREATE INDEX ind_name ON lookup (name(10)) USING HASH;
```

Ograničenja

- Primeri:

```
CREATE TABLE tbl_name(  
    id INT,  
    INDEX USING BTREE (id)  
)
```

```
CREATE TABLE tbl_name(  
    id INT PRIMARY KEY  
)
```

```
CREATE TABLE tbl_name(  
    id INT,  
    name VARCHAR(20),  
    PRIMARY KEY (id, name)  
)
```

Ograničenja

- 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 redosledu.
- Moguće je kreirati specijalne **FULLTEXT** indekse koji se koriste za Full-text (tekstualna) pretraživanja. Samo MyISAM mašina za skladištenje 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 deklarisanе kao NOT NULL.

Ograničenja

- 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.

Ograničenja

- 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, SET NULL, NO ACTION, RESTRICT**.

Ograničenja

- **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
- 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.

Izmena tabele

- **ALTER TABLE** omogućava promenu strukture postojeće tabele. Na primer, moguće je dodati ili izbrisati kolone, kreirati ili uništiti indekse, promeniti tip podataka za postojeće kolone 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.

InnoDB sistem za skladištenje

- **InnoDB** je sistem za skladištenje opšte namene u kome su usklađeni visoka pouzdanost i visoke performanse
- 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
- Ukoliko se ne navede drugačije ovo je u CREATE TABLE iskazu podrazumevan izbor
- Omogućava rad sa transakcijama, B-stablo indeksima, klasterovanim indeksima, stranim ključevima, prostornim podacima, ...

Alternativni sistemi za skladištenje

- **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 dođe do kvara sistema.
- **CSV** - podaci se smeštaju u fajlove tako da su vrednosti razdvojene zapetom
- **ARCHIVE** - mašina za skladištenje se koristi za smeštanje velikih količina podataka bez indeksa
- **BLACKHOLE** - mašina koja čita podatke ali ih nigde ne smešta, sličan efekat ima kao i kada se proslede podaci u /dev/null
- **MERGE** - kolekcija mašina za skladištenje MyISAM koje se mogu koristiti kao jedna
- **FEDERATED** - mašina koja omogućava pravljenju jedne objedinjene logičke sheme za više udaljenih baza podataka, pogodna je za distribuirane baze podataka
- **EXAMPLE** - mašina koja služi za primer kako da se napravi mašina za skladištenje
- Osnovne karakteristike se mogu izlistati komandom SHOW ENGINES