

MySQL savjeti i trikovi

Matko Kosmat | 1 travnja, 2026

Ne treba trošiti riječi da je MySQL najpopularnija baza podataka koja stoji iza mnogih web stranica i CMS suatava. Popularnost donosi i mnoštvo upita i problema, na koje ćemo pokušati odgovoriti na ovim stranicama.

U Službi za pomoć sistem-inženjerima često znamo dobiti upite kako promijeniti pretpostavljene vrijednosti u određenim servisima. Tako je prije nekog vremena jedan kolega želio otvoriti mrežni port preko kojeg bi mogao pristupiti svom MySQL poslužitelju, no to mu nije uspjelo. Port kojeg treba "osposobiti" je 3306. Taj port je iz sigurnosnih razloga zatvoren, jer je to još jedan način na koji udaljeni napadači mogu pokušati upasti na vaš poslužitelj.

Kao prvo, MySQL u osnovnoj konfiguraciji ne dopušta spajanje preko mrežnog *socket*a, nego isključivo preko Unix *socket*a, odnosno *socket*a na datotečnom sustavu. Da bismo omogućili pristup preko mreže, u konfiguraciji MySQL-a zakomentirajte sljedeće, ako već nije:

Ponovo pokrenite mysql poslužitelj u slučaju da ste morali napraviti ovu izmjenu:

No, to je samo predradnja za omogućavanje mrežne komunikacije. Nadalje je potrebno provjeriti što je upisano u datoteku `/etc/hosts.allow` (dio paketa **tcpd**):

Ovaj redak govori tcpd-u da dopusti spajanje na MySQL daemon samo računalima iz vaše domene i sa lokalnog poslužitelja. Ako se na MySQL želite spojiti s neke druge mreže, ili od kuće, ovdje možete dodati sva računala i mreže s kojih ćete se spajati, primjerice:

Ukoliko je ispred imena ".", smatra se da želite da se cijela poddomena može spajati na MySQL port. Ukoliko točke nema, smatra se da je to FQDN ime računala, i samo s njega će se moći spajati klijenti na MySQL port.

Ukoliko ne postoji redak "mysqld", to znači da je spajanje na MySQL daemon dopušten sa svih računala bilo gdje s Interneta, što naravno nikako nije poželjno. Upišite dakle barem "mysqld: .vasadomena.hr 127.0.0.1", što će omogućiti spajanje samo s lokalnog poslužitelja i vaše domene.

Napomena: pregledajte i datoteku `/etc/hosts.deny`, ukoliko tamo ima nekih unosa koji počinju sa "mysqld:" to znači da je za navedene domene i računala pristup **zabranjen**.

Zadnje što trebate provjeriti na samom poslužitelju je vatrozid, odnosno je li u vašim **iptables** pravilima propušten port 3306. Ovo ćete najlakše provjeriti tako da se probate spojiti na port 3306 s nekog računala izvan vaše lokalne mreže (možete i od kuće):

```
$ telnet mysql.domena.hr 3306
```

```
Trying 161.53.xxx.yyy...
```

```
telnet: Unable to connect to remote host: Connection refused
```

Ukoliko dobijete poruku poput ove, potrebno je otvoriti port 3306 za pristup izvana. Preporuka je otvoriti port za iste hostove koje ste naveli u `/etc/hosts.allow`.

Ukoliko i dalje imate problema sa spajanjem, provjerite MySQL-ove tablice s pravima pristupa korisnika i njihovim privilegijama. I ovdje trebate unijeti sve korisnike i računala s kojih se žele spajati (napomena: ovi korisnici u MySQL bazi nemaju nikakve veze s korisnicima na sustavu, nemojte to pobrkati!).

Primjerice, želite dopustiti korisniku "perica" spajanje s računala "perica.domena.hr" (ujedno kreirate tog korisnika):

Gornji redak dopušta spajanje korisniku perica, ali i sva prava nad tablicama u bazi "baza".

Prošli smo ukratko sve korake koje je potrebno učiniti kako bi omogućili spajanje na MySQL s

udaljenih mreža. Opet ćemo ponoviti da ovime otvarate više mogućnosti za potencijalnu provalu, ali pažljivim konfiguriranjem i praćenjem možete napraviti istovremeno fleksibilan, ali i siguran sustav. Većina sistem-inženjera vjerojatno ni nema potrebe za ovakvim načinom rada, jer im se svi servisi vrte na istom poslužitelju, ali dobro je znati kako podesiti mrežni način rada u slučaju potrebe.

Zbog uporabe u različitim CMS sustavima, Mysql je svakako postao najpopularnija baza podataka u različitim, obično manje zahtjevnim okruženjima. Uz velike količine dodatnih informacija i trikova koji se mogu naći po internet stranicama, taj status će vjerojatno još dugo uživati. Baš zbog svoje popularnosti i činjenice da mnogo toga ovisi o ispravnom radu mysqla, postupak nadogradnje treba pažljivo provesti.

Prvo, treba napraviti *dump* svih postojećih baza:

Time smo dobili manju ili veću dump datoteku, u ovisnosti kolika se količina podataka u bazi nalazi. Ukoliko je prevelika, možete je komprimirati i tako prebaciti. Na drugom poslužitelju (ili na istom poslužitelju nakon instalacije novog mysqla), napravite:

Naravno, podrazumijeva se da baze s istim imenima ne postoje u mysqlu. Ukoliko tako želite, možete dodati DROP naredbe u dump datoteku sa opcijama `--add-drop-table` i `--add-drop-database` kod operacije dumpa. Taj će korak obrisati podatke, ukoliko postoje baze i tablice s istim imenima.

Sljedeći, vrlo bitan korak, je pokrenuti **mysql_upgrade** nakon pokretanja mysqla, što će napraviti nužne promjene u tablicama kako bi ih pripremio za rad u novoj inačici mysqla:

U slučaju da mysql_upgrade ne radi, možete napraviti sljedeće:

Nešto više informacija možete potražiti na adresi http://openconcept.ca/mysql_dumps_and_imports i <http://dev.mysql.com/doc/refman/5.0/en/mysql-upgrade.html>

Puni popis promjena između ovih dviju inačica mysqla možete vidjeti na linku <http://dev.mysql.com/doc/refman/5.0/en/upgrading-from-4-1.html>

MySQL je danas vjerojatno najpopularniji od raspoloživih sustava baza podataka, što može zahvaliti svojom jednostavnošću, brzim razvojem i mnoštvom dobre dokumentacije i uputa. Unatoč tome, mišljenja smo da je na Portalu dobrodošla kuharica, točnije *cheat-sheet* za Mysql. Kompiliranjem raznih drugih izvora i upita na SysHelpu, došli smo do ovog podsjetnika. Nadamo se da će poslužiti u vašem svakodnevnom radu.

Napomena: simbol ">" označava odzivni znak (prompt) Mysqla, dok je znak "\$" oznaka za običnu korisničku ljsku (dakle, ne administrativnu). Naredbe Mysqla nisu *case-sensitive*.

Konfiguracijska datoteka Mysqla je `/etc/mysql/my.cnf`. U njoj možete podesiti osnovne paramtere Mysql sustava. Linux sustavi bazirani na Debianu uvode i drugu datoteku, `debian.cnf`, gdje se nalazi specifična konfiguracija (primjerice, uvodi se Mysql korisnik `debian-sys-maint` u svrhu održavanja mysqla. Tog korisnika nemojte brisati).

Mysql *cheat-sheet*

Logiranje na MySQL bazu (koristiti `-h` samo po potrebi, na lokalnom računalu ne morate navoditi):

Pravljenje baze:

Prikaz svih baza:

Prebacivanje na određenu bazu:

Prikaz svih tablica u bazi:

Prikaz opisa stupaca tablice:

Brisanje baze:

Brisanje tablice:

Prikaz svih podataka u tablici:

Prikaz podataka o stupcima tablice:

Prikaz određenih redaka gdje [ime polja] sadrži određeni pojam:

Prikaz svih zapisa koji sadrže ime "Ivica" i broj telefona '1234567'

Prikaz svih zapisa koji sadrže ime "Ivica" i broj telefona '1234567', posloženo po broju telefona:

Prikaz svih zapisa koji počinju s izrazom "lv" i brojem telefona '1234567', posloženo po broju telefona:

Pretraga zapisa preko regularnih izraza, ovaj izraz traži zapise u stupcu "rec" koji počinju s 'a':

Prikaz jedinstvenih zapisa:

Prikaz zapisa sortiranih po rastućem ili silaznom nizu:

Brojanje redaka:

Zapisivanje rezultata u datoteku:

Operacija JOIN na zajedničkim stupcima:

Prebacivanje na bazu Mysql. Dodavanje novog korisnika:

Promjena korisničkih zaporki (iz korisničke ljuske - *shell*)

Ovdje ključna riječ '**password**' označava naredbu, na tom mjestu ne trebate ukucavati staru zaporku. Za staru (trenutno vrijedeću) zaporku ćete biti upitani nakon izvršavanja ove naredbe.

Promjena korisničkih zaporki (u Mysqlu):

Prebacivanje na bazu "mysql". Dodjeljivanje privilegija za određenu bazu:

Osvježavanje podataka koji su već u tablici:

Brisanje redaka iz tablice:

Osvježavanje dozvola na bazama (u slučaju promjena):

Brisanje stupca:

Dodavanje novog stupca u tablicu:

Promjena imena stupca:

Promjena određenog podatka u stupcu

Učini stupac jedinstvenim (tako da ne mogu postojati duplikati):

Povećavanje stupca:

Brisanje jednoznačno označene (indexom) tablice:

Učitavanje/importiranje CSV datoteke u tablicu:

Dump svih baza (backup datoteka sadrži sve naredbe za uspješno vraćanje):

Dump svih baza za backup (backup datoteka ne sadrži DROP naredbe):

Dump samo jedne baze za backup:

Dump jedne tablice iz baze:

Vraćanje baze (ili tablice) iz backupa:

Vraćanje svih baza iz backupa (dump baze je napravljen sa --all-databases):

Optimiziranje i provjera tablica u svim bazama:

Provjera tipa tablica (koji je **database engine**):

ili iz naredbenog retka:

Pravljenje tablice primjer 1:

Pravljenje tablice primjer 2:

Pronašli ste ili sami napravili CMS sustav za potrebe vaše ustanove, instalirali i pustili u pogon. Jedan dio korisnika Vas zamoli da isti takav CMS postavite i za potrebe katedre ili predmeta. Budući da praktički imate već sve gotovo, odlučite rabiti zajedničku bazu (primjerice "**fakultet**") i zadržati iste nazive tablica (onako kako je u samom CMS-u predefinirano). Kako bi mogli koristiti istu bazu i zadržati iste tablice neophodno je koristiti prefiks ispred imena tablice, te malo dotjerati kôd kako bi se koristile tablice sa prefiksom.

Tipičan naziv tablice je primjerice "**user**", pa se možete odlučiti dodati prefiks "**web1**", što će na kraju izgledati ovako: "**web1_user**".

Za primjer, pretpostavimo da Vaš CMS ima 15 tablica koje su kreirane u bazi "**fakultet**". Svakom novom instalacijom CMS-a u istoj bazi broj tablica se povećava, te u slučaju da ste podigli četiri CMS sustava za različite kolegije ili katedre imat ćete 60 tablica u bazi.

Prilikom kreiranja rezervne kopije pomoću **mysqldump** obuhvatili bi cijelu bazu "**fakultet**" i svih 60 tablica, te napravili kopiju za 4 CMS sustava u jednom potezu.

No, "problem" se javlja kad morate raditi izmjene **samo za jedan** od instaliranih CMS-ova, a da ne ugrozite podatke i funkcionalnost ostalih CMS-ova. Najjednostavnije rješenje je eksportiranje smao tablica na kojima želite raditi izmjene, te vraćanje istih nazad u bazu.

To možemo obaviti tako da koristimo **zamjenske znakove** (*wildcards*) kod odabira prefiksa tablica koje želimo eksportirati. Pravi problem je što **mysqldump ne radi sa zamjenskim znacima**, no i za to postoji rješenje. Prijavimo se na mysql bazu, te napravimo sljedeće:

u primjeru vidimo prefikse tablica (web1, web2,...) koje možemo iskoristiti za eksport.

U naredbenoj liniji mysql-a iskoristit ćemo SQL operator "LIKE":

Iz primjera je vidljivo što sve moramo iskoristiti te ukomponirati sa mysqldump naredbom.

Krenimo korak po korak. U naredbenoj liniji prvo istestiramo cijelu proceduru (primjetite opciju "-e" koji nalaže izvršavanje SQL naredbe):

Ono što moramo izbaciti su okomite i vodoravne linije u Tables_in, tu koristimo grep -v.

Kompletan ovaj rezultat (listu) trebamo sad proslijediti naredbi mysqldump i to pomoću naredbe xargs i naravno sve to eksportirati u datoteku.

i svih 15 tablica sa prefiksom "web1" je eksportiranu u datoteku kopija.sql.

Bilo bi zgodno napraviti jednu skriptu da se izbjegne pisanje kompletne sintakse:

I to je to. Datatoteku nastalu nakon eksporta baze sada možete staviti na drugi server, promijeniti što god hoćete u dump datoteci, te vratiti u staru bazu ili jednostavno sačuvati kao backup na DVD-u ili nekom drugom mediju poput USB sticka.

Kako smo u članku o [alatu fsck](#) spominjali, oštećenja datotečnog sustava su iako ne prečesto, ali ipak moguća. Čak i ako se sustav prilikom podizanja oporavio, to još ne znači da će svi servisi proraditi od prve. Dobar primjer je MySQL, koji se vrlo često rabi kod raznih CMS sustava, pa je prestanak radabaze podatakatime postao ozbiljan problem, jer znači i prestanak rada web stranica.

U većini slučajeva, MySQL se može oporaviti sam, no ponekad je ipak potrebna intervencija sistem-inženjera. Ukoliko je ta intervencija potrebna, u logovima ćete moći vidjeti poruke poput ove:

Srećom, problem možemo probati riješiti na tri načina. Osim naredbenog retka MySQL-a ("**CHECK TABLE/REPAIR TABLE**"), postoje i alati **myisamchk**, te **mysqlcheck**. Alati myisamchk i mysqlcheck mogu detektirati i popravljati samo probleme u MyISAM tablicama, ali "CHECK TABLE"

može detektirati probleme u InnoDB tablicama (ne i popraviti).

MyISAM je osnovni "*storage engine*" MySQL-a, a rabi se još i InnoDB. MyISAM je MySQL-ova implementacija ISAM enginea, i vrlo je brza, ali i neotpornija na rušenja baze. InnoDB je bolji, ali zahtjeva više memorije i sporiji je. Više o svim enginima koje MySQL podržava pročitajte na <http://en.wikipedia.org/wiki/MySQL> i <http://en.wikipedia.org/wiki/MyISAM>.

1. način: "CHECK/REPAIR TABLE"

Sintaksa naredbe je:

Opcije koje možete rabiti su:

Opcija QUICK je najbrža i najpovršnija, dok je EXTENDED najsporija i najdetaljnija, dok su opcije između tih kompromis između brzine i detaljnosti. Koju ćete rabiti ovisi od slučaja do slučaja. Ukoliko vam je baza mala nema prepreka da odmah stavite opciju EXTENDED:

U ovom slučaju popravak ne treba (poruka je "OK"), no može se dogoditi da je oštećena indeksna datoteka, kao u ovom primjeru:

Naredba REPAIR ima nešto drugačije opcije nego opcija CHECK, ali je osnovna premisa ista. Postoje tri opcije: QUICK, EXTENDED i USE_FRM. Uobičajeno nećete rabiti **nijednu** opciju. Opciju QUICK ćete odabrati kod velikih baza kada je imperativ da baza proradi čim prije. Opcija EXTENDED se rabi kada nemate drugog izbora, i može generirati lažne zapise ("*garbage rows*").

Opcija USE_FRM je najkorisnija kod gore navedenog primjera, kada nedostaje indeksna datoteka, ili je oštećena:

2. način: "myisamchk"

Da bi alat myisamchk mogao raditi, potrebno je da MySQL **ne radi**, znači alat radi direktno nad datotekama baze:

Ukoliko se pokaže neka greška, možete je pokušati popraviti s opcijom "--repair":

Ukoliko dobijete poruku "clients are using or haven't closed the table properly", to znači da su tablice još uvijek otvorene, pa zaustavite MySQL daemon prije ponovnog pokušaja.

Naredba myisamchk također može rabiti različite razine provjere i popravka tablica. Najbrža je --quick, a najtemeljitija je --extend-check. Kod zadnje opcije preporučljivo je rabiti opciju --backup, jer može generirati lažne podatke pa je dobro imati kopiju stare (makar i oštećene) datoteke.

Zanimljiva je opcija --description, koja će vam dati podatke o tablici, poput statusa ili kodne stranice. Za više informacija uporabite je uz opciju --verbose, koju možete staviti više puta za još više informacija.

Na kraju, možete istovremeno provjeriti ili popraviti više baza/tablica, jednostavno upotrijebite zvjezdicu:

Naredba myisamchk se nalazi u paketu "mysql-server-5.0", odnosno pripadajućoj aktualnoj inačici paketa MySQL.

3. način: "mysqlcheck"

Treći način opravka je moguć pomoću alata mysqlcheck. Za ovaj način, mysql poslužitelj **mora biti pokrenut**. Mysqlcheck može i optimizirati tablice, ali to ostavljamo vama za "domaću zadaću".

Sintaksa ove naredbe je:

Uporaba je kao kod naredbi **mysql**, **mysqladmin** i sličnih:

Ukoliko neka od tablica nije u redu, možemo je probati popraviti sa opcijom "--repair", kao u primjeru. Ukoliko želite samo provjeriti tablice, a ukoliko se pojavi greška odmah je probati i popraviti, upotrijebite opciju "--auto-repair".

Dostupne su vam iste opcije kao i kod naredbenog retka u mysql-u, pa su tako opcije "--check-only-changed", "--extended", "--fast", "--medium-check" i "--quick" ekvivalentne opcijama koje možete upotrijebiti sa "CHECK TABLE".

Naredba mysqlcheck se nalazi u paketu "mysql-client-5.0", odnosno pripadajućoj aktualnoj inačici paketa.

Došli smo do kraja, a nadamo se da nećete morati rabiti niti jedan od ovih alata, jer to potencijalno može značiti gubitak podataka i *downtime*, koji korisnici, naravno, ne vole. Zato još jednom napominjemo da je aktualan *backup* jedino što vas može izvući i iz najvećih problema.

Nedavno smo na helpdesku pomogli korisniku koji je zaboravio administratorsku (root@localhost) lozinku za MySQL. Rješenje problema je da se MySQL pokrene bez tzv. GRANT tablica, odnosno da se zaobiđu tablice s dozvolama. To se radi opcijom --skip-grant-tables.

Prvo zaustavimo MySQL:

Zatim ga dignemo bez GRANT tablica, slijedećom naredbom:

Sada možemo pristupiti MySQL poslužitelju bez autentikacije, te izvršimo naredbe:

Za novije inačice MySQL-a (5.7.6+) i MariaDB-a (10.1.20+) rabi se naredba ALTER USER:

Napomena: Ako naredba 'ALTER USER' ne radi, vjerovatno se radi o nekom većem problemu. U svakom slučaju, još možete probati upotrijebiti naredbu "UPDATE ... SET":

Još nam ostaje restart MySQL-a:

Za novije Debiane naravno, rabit ćemo naredbu systemctl:

Provjerimo da li je sve u redu:

I to je to!