

## MySQL savjeti i trikovi

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.

- [Logirajte](#) [1] se za dodavanje komentara

## Kako omogućiti pristup MySQL-u preko mreže?



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:

```
#skip-networking
```

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

```
# /etc/init.d/mysql restart
```

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

```
mysqld: .domena.hr 127.0.0.1
```

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:

```
mysqld: .domena.hr .domena2.hr poslužitelj.negdje.hr 127.0.0.1 161.53.XXX.YYY
```

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):

```
$ mysql --user=root mysql --password
Password: <vaš password za root korisnika u MySQL-u>

mysql> GRANT ALL PRIVILEGES ON baza.* TO 'perica'@'perica.domena.hr'
-> IDENTIFIED BY 'neka_zaporka' WITH GRANT OPTION;
```

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.

- [Logirajte](#) [1] se za dodavanje komentara

čet, 2010-04-29 15:10 - Željko Boroš**Kuharice:** [Linux](#) [2]

**Kategorije:** [Servisi](#) [3]

**Vote:** 5

Vaša ocjena: Nema Average: 5 (1 vote)

## MySQL: kako napraviti upgrade s mysql 4 na mysql 5?



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:

```
# mysqldump -u root -pnekipassword --all-databases --opt > /somewhere/alldatabases.sql
```

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:

```
# mysql -u username -pnekipassword < /somewhere/alldatabases.sql
```

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:

```
# mysqlcheck --check --all-databases --auto-repair --password=nekipassword --user=root
# mysql_fix_privilege_tables --password=nekipassword --user=root
```

Nešto više informacija možete potražiti na adresi [http://openconcept.ca/mysql\\_dumps\\_and\\_imports](http://openconcept.ca/mysql_dumps_and_imports) i <http://dev.mysql.com/doc/refman/5.0/en/mysql-upgrade.html> [4]

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>

KEYWORDS: mysql mysql5 upgrade

- [Logirajte](#) [1] se za dodavanje komentara

čet, 2008-05-29 15:45 - Željko Boroš **Vijesti:** [Linux](#) [5]

**Kuharice:** [Za sisteme](#) [6]

**Kategorije:** [Servisi](#) [3]

**Vote:** 0

No votes yet

## MySQL: osnovne naredbe i operacije



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 ljusku (dakle, ne administrativnu). Naredbe Mysqla nisu *case-sensitive*.

Konfiguracijska datoteka Mysqla je `/etc/mysql/my.cnf`. U njoj možete podesiti osnovne parametre 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):

```
$ mysql -h poslužitelj -u root -pzaporka
```

Pravljenje baze:

```
> CREATE database [ime baze];
```

Prikaz svih baza:

```
> SHOW databases;
```

Prebacivanje na određenu bazu:

```
> USE [ime baze];
```

Prikaz svih tablica u bazi:

```
> SHOW tables;
```

Prikaz opisa stupaca tablice:

```
> DESCRIBE [ime tablice];
```

Brisanje baze:

```
> DROP database [ime baze];
```

Brisanje tablice:

```
> DROP table [ime tablice];
```

Prikaz svih podataka u tablici:

```
> SELECT * FROM [ime tablice];
```

Prikaz podataka o stupcima tablice:

```
> SHOW COLUMNS FROM [ime tablice];
```

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

```
> SELECT * FROM [ime tablice] WHERE [ime polja] = "pojam";
```

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

```
> SELECT * FROM [ime tablice] WHERE name = "Ivica" AND phone_number = '1234567';
```

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

```
> SELECT * FROM [ime tablice] WHERE name != "Ivica" AND phone_number  
= '1234567' order by phone_number;
```

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

```
> SELECT * FROM [ime tablice] WHERE name LIKE "Iv%" AND phone_number = '1234567';
```

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

```
> SELECT * FROM [ime tablice] WHERE rec RLIKE "^a$";
```

Prikaz jedinstvenih zapisa:

```
> SELECT DISTINCT [ime stupca] FROM [ime tablice];
```

Prikaz zapisa sortiranih po rastućem ili silaznom nizu:

```
> SELECT [col1],[col2] FROM [ime tablice] ORDER BY [col2] DESC;
```

Brojanje redaka:

```
> SELECT COUNT(*) FROM [ime tablice];
```

Zapisivanje rezultata u datoteku:

```
> SELECT * INTO OUTFILE "ispis.txt" FROM [ime tablice];
```

Operacija JOIN na zajedničkim stupcima:

```
> SELECT lookup.illustrationid, lookup.personid, person.birthday from lookup  
left join person on lookup.personid=person.personid=statement to join  
birthday in person table with primary illustration id;
```

Prebacivanje na bazu Mysql. Dodavanje novog korisnika:

```
> INSERT INTO mysql (Host,User,Password) VALUES( '%', 'user', PASSWORD( 'password' ));  
> FLUSH PRIVILEGES;
```

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

```
$ mysqladmin -u root -h hostname.domena.hr -p password 'novazaporka'
```

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):

```
> SET PASSWORD FOR 'user'@'hostname' = PASSWORD( 'passwordhere' );
```

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

```
> USE mysql;  
> INSERT INTO mysql (Host,Db,User,Select_priv,Insert_priv,  
Update_priv,Delete_priv,Create_priv,Drop_priv)  
VALUES ( '%', 'db', 'user', 'Y', 'Y', 'Y', 'Y', 'Y', 'N' );  
> FLUSH PRIVILEGES;
```

Osvježavanje podataka koji su već u tablici:

```
> UPDATE [ime tablice] SET Select_priv = 'Y', Insert_priv = 'Y', Update_priv = 'Y' whe  
re [ime polja] = 'user';
```

Brisanje redaka iz tablice:

```
> DELETE FROM [ime tablice] WHERE [ime polja] = 'pojam';
```

Osvježavanje dozvola na bazama (u slučaju promjena):

> FLUSH PRIVILEGES;

Brisanje stupca:

> ALTER table [ime tablice] drop column [ime stupca];

Dodavanje novog stupca u tablicu:

> ALTER table [ime tablice] add column [novo ime stupca] varchar (20);

Promjena imena stupca:

> ALTER table [ime tablice] change [staro ime stupca] [novo ime stupca] varchar (50);

Promjena određenog podatka u stupcu

> UPDATE [ime tablice] SET [ime stupca]=(REPLACE ([ime stupca], 'old', 'new'));

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

> ALTER table [ime tablice] add unique ([ime stupca]);

Povećavanje stupca:

> ALTER table [ime tablice] modify [ime stupca] VARCHAR(3);

Brisanje jednoznačno označene (indexom) tablice:

> ALTER table [ime tablice] drop index [ime stupca];

Učitavanje/importiranje CSV datoteke u tablicu:

```
> LOAD DATA INFILE 'filename.csv' replace INTO TABLE [ime tablice]
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' (field1,field2,field3);
```

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

```
$ mysqldump -u root -ppassword --all-databases --opt > alldatabases.sql
```

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

```
$ mysqldump -u root -ppassword --all-databases > alldatabases.sql
```

Dump samo jedne baze za backup:

```
$ mysqldump -u username -ppassword --databases databasename > databasename.sql
```

Dump jedne tablice iz baze:

```
$ mysqldump -c -u username -ppassword databasename tablename >
databasename.tablename.sql
```

Vraćanje baze (ili tablice) iz backupa:

```
$ mysql -u username -ppassword databasename < databasename.sql
```

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

```
$ mysql -u username -ppassword < alldatabases.sql
```

Optimiziranje i provjera tablica u svim bazama:

```
$ mysqlcheck -u root -ppassword --auto-repair --check --optimize --all-databases
```

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

```
> SHOW TABLE STATUS;
```

ili iz naredbenog retka:

```
$ mysqlshow --status ime_baze;
```

Pravljenje tablice primjer 1:

```
> CREATE TABLE [ime tablice] (firstname VARCHAR(20), middleinitial VARCHAR(3),
lastname VARCHAR(35), suffix VARCHAR(3), officeid VARCHAR(10),
userid VARCHAR(15), username VARCHAR(8), email VARCHAR(35), phone VARCHAR(25),
groups VARCHAR(15), datestamp DATE, timestamp time, pgpemail VARCHAR(255));
```

Pravljenje tablice primjer 2:

```
> CREATE TABLE [ime tablice] (personid int(50) not null auto_increment
primary key, firstname varchar(35), middlename varchar(50), lastname
varchar(50) default 'prezime');
```

UPDATED: 2012-08-18

- [Logirajte](#) [1] se za dodavanje komentara

pet, 2008-12-05 14:59 - Željko Boroš **Kuharice:** [Linux](#) [2]

**Kategorije:** [Servisi](#) [3]

**Vote:** 4.333335



Vaša ocjena: Nema Average: 4.3 (3 votes)

## Mysqldump - upotreba zamjenskih znakova

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 samo 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:

```
mysql> USE fakultet; // odabiremo bazu
Database changed
```

```
mysql> show tables; // prikaz svih tablica, primjetujemo 60 tablica
```

```
+-----+
| Tables_in_fakultet |
+-----+
| web1_addons         |
| web1_groups         |
| .....             |
| web2_addons         |
| .....             |
| web3_addons         |
| .....             |
+-----+
60 rows in set (0.00 sec)
```

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

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

```
mysql> show tables like "web1%"; // tablice s prefiksom web1
+-----+
| Tables_in_fakultet (web1%) |
+-----+
| web1_addons                  |
| web1_groups                  |
| .....                      |
| web1_mod_captcha            |
| web1_users                   |
+-----+
15 rows in set (0.00 sec)
```

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):

```
server# mysql fakultet -u korisnik -plozinka -e ' show tables like "web1%" '
+-----+
| Tables_in_fakultet (web1%) |
+-----+
| web1_addons                  |
| web1_groups                  |
| web1_mod_captcha            |
| .....                      |
| web1_users                   |
+-----+
```

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

```
server# mysql fakultet -u root -plozinka -e \
'show tables like "web1%"' | grep -v Tables_in

web1_addons
web1_groups
web1_mod_captcha_control
.....
web1_users
```

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

```
server# mysql fakultet -u korisnik -plozinka -e \
'show tables like "web1%"' | grep -v Tables_in \
| xargs mysqldump --opt fakultet -u korisnik -p lozinka > kopija.sql
```

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:

```
#!/bin/bash
echo -n "unesi ime baze "
read baza
echo -n "unesi korisnika "
read korisnik
echo -n "unesi zaporku "
read lozinka
echo -n "unesi prefix tablice "
read pref
echo -n "naziv datoteke "
read datoteka
mysql $baza -u$korisnik -p$lozinka -e 'show tables like "$pref%"' | \
grep -v Tables_in | xargs mysqldump --opt $baza -u $korisnik \
-p$lozinka > $datoteka
echo ""
echo "EXPORT BAZE $baza sa tablicama $pref ZAVRSEN"
echo ""
exit 0
```

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.

Zdravko Rašić

- [Logirajte](#) [1] se za dodavanje komentara

Čet, 2010-01-14 10:44 - Zdravko Rašić **Kuharice:** [Linux](#) [2]

**Kategorije:** [Servisi](#) [3]

**Vote:** 4.5

Vaša ocjena: Nema Average: 4.5 (2 votes)

## Popravlak oštećene MySQL baze



Kako smo u članku o [alatu fsck](#) [7] 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 rada baze podataka time 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:

## Table 'vocabulary' is marked as crashed and should be repaired

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> [8] i <http://en.wikipedia.org/wiki/MyISAM> [9].

### 1. način: "CHECK/REPAIR TABLE"

Sintaksa naredbe je:

```
mysql> CHECK TABLE tablica[, tablica2...] [opcija][, opcija2...]
```

Opcije koje možete rabiti su:

QUICK, FAST, CHANGED, MEDIUM, EXTENDED

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:

```
# mysql -u root -p
Enter password:*****
...
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> CHECK TABLE user EXTENDED;
+-----+-----+-----+-----+
| Table      | Op    | Msg_type | Msg_text |
+-----+-----+-----+-----+
| mysql.user | check | status   | OK       |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> REPAIR TABLE vocabulary;
+-----+-----+-----+-----+
| Table              | Op    | Msg_type | Msg_text |
+-----+-----+-----+-----+
| drupal.vocabulary | repair | error    | Can't find file: 'vocabulary.MYI' |
+-----+-----+-----+-----+
```

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:

```
mysql> REPAIR TABLE vocabulary USE_FRM;
```

| Table             | Op     | Msg_type | Msg_text                           |
|-------------------|--------|----------|------------------------------------|
| drupal.vocabulary | repair | warning  | Number of rows changed from 0 to 3 |
| drupal.vocabulary | repair | status   | OK                                 |

## 2. način: "myisamchk"

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

```
# myisamchk vocabulary.MYI
Checking MyISAM file: vocabulary.MYI
Data records:      0
Deleted blocks:    0
- check file-size
- check record delete-chain
- check key delete-chain
- check index reference
- check data record references index: 1
- check record links
```

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

```
# myisamchk --repair vocabulary.MYI
- recovering (with sort) MyISAM-table 'vocabulary.MYI'
Data records: 95
- Fixing index 1
- Fixing index 2
```

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 upotrijebite 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:

```
# cd /var/lib/mysql/baza_koju_zelite_provjeriti
```

```
# myisamchk --repair *.MYI
```

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:

```
mysqlcheck [opcije] ime_baze tablica [tablica2... ].
```

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

```
# mysqlcheck -u root -p<vasa_zaporka> --repair drupal
drupal.access                OK
drupal.accesslog             OK
drupal.aggregator_category   OK
drupal.aggregator_category_feed OK
drupal.aggregator_category_item OK
drupal.aggregator_feed       OK
drupal.aggregator_item       OK
...
```

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.

- [Logirajte](#) [1] se za dodavanje komentara

pon, 2010-04-26 10:23 - Željko BorošKuharice: [Linux](#) [2]

**Kategorije:** [Servisi](#) [3]

**Vote:** 5

Vaša ocjena: Nema Average: 5 (1 vote)

## Zaboravljena MySQL root lozinka



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:

```
# /etc/init.d/mysql stop
Stopping MySQL database server: mysqld
```

Zatim ga dignemo bez GRANT tablica, slijedećom naredbom:

```
# mysqld_safe --skip-grant-tables --skip-networking &
Starting mysqld daemon with databases from /var/lib/mysql
mysqld_safe[7754]: started
```

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

```
mysql -e "UPDATE mysql.user SET Password = PASSWORD('novipassword')
        WHERE User = 'root'"
mysql -e "FLUSH PRIVILEGES;"
```

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

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'novipassword';
FLUSH PRIVILEGES;
```

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

```
UPDATE mysql.user SET authentication_string = PASSWORD('novipassword') WHERE User='root' AND Host='localhost';
```

Još nam ostaje restart MySQL-a:

```
# /etc/init.d/mysql restart
STOPPING server from pid file /var/run/mysqld/mysqld.pid
mysqld_safe[8310]: ended
Starting MySQL database server: mysqld.
Checking for crashed MySQL tables in the background.
```

Za novije Debiane naravno, rabit ćemo naredbu systemctl:

```
# systemctl restart mariadb
```

Provjerimo da li je sve u redu:

```
mysql -e "select now();" -p
Enter password:
+-----+
| now() |
+-----+
| 2006-01-27 19:37:09 |
+-----+
```

I to je to!

- [Logirajte](#) [1] se za dodavanje komentara

pet, 2006-02-03 15:54 - Uredništvo **Kuharice:** [Za sistemce](#) [6]

**Kategorije:** [Servisi](#) [3]

**Vote:** 0

No votes yet

**story\_tag:** [mariadb](#) [10]

[mysql](#) [11]

[change root password](#) [12]

**Source URL:** <https://sysportal.carnet.hr/node/489>

### Links

[1] <https://sysportal.carnet.hr/sysportallogin>

[2] <https://sysportal.carnet.hr/taxonomy/term/17>

[3] <https://sysportal.carnet.hr/taxonomy/term/28>

[4] <http://dev.mysql.com/doc/refman/5.0/en/mysql-upgrade.html>

[5] <https://sysportal.carnet.hr/taxonomy/term/11>

[6] <https://sysportal.carnet.hr/taxonomy/term/22>

[7] <https://sysportal.carnet.hr/node/737>

[8] <http://en.wikipedia.org/wiki/MySQL>

[9] <http://en.wikipedia.org/wiki/MyISAM>

[10] <https://sysportal.carnet.hr/taxonomy/term/373>

[11] <https://sysportal.carnet.hr/taxonomy/term/374>

[12] <https://sysportal.carnet.hr/taxonomy/term/375>