**Univerzitet u Nišu, Elektronski fakultet**

**Katedra za računarstvo**

****

**SEMINARSKI RAD**

**MySql - Backup i restore**

**Predmet: Sistemi za upravljanje bazama podataka**

**Mentor: Doc. dr Aleksandar Stanimirović Student: Vesna Stojanović 1339**

**Niš 2022.**

**Sadržaj**

**Uvod**

**Razlozi pravljenja backup-a**

**Vrste rezervnih kopija i oporavka**

**Alati za backup i oporavak**

**Backup i restore MySql baze podataka pomoću komande mysqldump**

**Zaključak**

**Literatura**

# Uvod

Prvo ćemo razjasniti neke od ključnih termina koje ćemo koristiti u ovom radu. To su vraćanje (*eng. restore*) i oporavak (*eng. recover*). Vraćanje označava pribavljanje podataka iz backup-a i bilo njihovo učitavanje u MySQL ili postavljanje fajlova tamo gde MySQL očekuje da budu. Oporavak obično označava ceo proces spasavanja sistema ili dela sistema, nakon što je nešto pošlo naopako. Ovo uključuje vraćanje podataka iz backup-ova, kao i sve neophodne korake da bi server ponovo upotpunosti funkcionisao, kao što je restartovanje MySQL-a, promena konfiguracije i tako dalje.

Za mnoge ljude, oporavak samo označava popravku oštećenih tabela posle pada sistema. Ovo nije isto kao i oporavak celog servera. Oporavak od pada uređaja za skladištenje usklađuje njegove podatke i log fajlove. Ovo osigurava da fajlovi podataka sadrže samo promene urađene od izvršenih transakcija, i ovo reprodukuje transakcije iz log fajlova koje još nisu primenjene na fajlove podataka. Ovo bi moglo biti deo celokupnog procesa oporavka. Međutim, ovo nije isto kao i oporavak koji treba da uradimo na primer nakon slučajne naredbe DROP TABLE.

Mogu postojati situacije kada se podaci izgube ili oštete, na primer, kada su korisnici slučajno izbrisali ili prepisali podatke. Iz ovih razloga, preporučuje se redovno pravljenje rezervne kopije. Kroz ovaj proces možete osigurati konzistentnost podataka u vašim bazama podataka i biti sigurni da se ništa kritično neće dogoditi vašim podacima i da neće biti nepovratno izgubljeni. Pored toga, rezervna kopija podataka pomaže da podaci budu ažurirani i omogućava vam da ih vratite u bilo koje vreme u stanju u kom ste napravili rezervnu kopiju pre nego što su podaci oštećeni.

Postoje i *hot, warm i cold backup-ove*. Ljudi generalno koriste ove termine za označavanje uticaja backup-ova: „hot“ backup-ovi ne treba da zahtevaju bilo kakav prekid sistema. Međutim, neki alati čak koriste reč „hot“ i kada definitivno ne podrazumevaju ono što mi smatramo hot backup-ovima.

Hot rezervna kopija, poznata i kao dinamička ili onlajn rezervna kopija, je pravljenje rezervne kopije podataka dok je baza podataka aktivno onlajn i dostupna korisnicima. Hot rezervna kopija je standardni način pravljenja većine rezervnih kopija baze podataka.

Cold rezervna kopija, koja se naziva i rezervna kopija van mreže, je rezervna kopija baze podataka tokom koje je baza podataka van mreže i nije dostupna za ažuriranje. Ovo je najsigurniji način pravljenja rezervne kopije jer izbegava rizik od kopiranja podataka koji su možda u procesu ažuriranja.

U warm rezervnoj kopiji, server je uključen, ali ne obavlja nikakav posao, ili se uključuje s vremena na vreme da bi dobio ažuriranja sa servera za koji se pravi rezervna kopija. Tople rezervne kopije se obično koriste za preslikavanje ili replikaciju.

# Razlozi pravljenja backup-a

Evo nekoliko razloga zašto su backup-ovi važni:

* **Oporavak od katastrofe** – Oporavak od katastrofe je ono što radimo kada hardver padne, neka greška ošteti podatke ili server i njegovi podaci postanu nedostupni ili neupotrebljivi iz nekog drugog razloga. Moramo biti spremni na sve, od nekog ko se slučajno poveže na pogrešan server i uradi ALTER TABLE, do zlonamernih napadača ili MySQL greške, iako su šanse za bilo koje posebno katastrofalno stanje prilično male.
* **Ljudi menjaju svoja mišljenja** – Bićemo iznenađeni koliko često ljudi namerno obrišu podatke i onda ih žele nazad.
* **Revizija** – Ponekad moramo da znamo kako su naši podaci ili šema izgledali u nekom trenutku u prošlosti. Na primer, možda ćemo otkriti grešku u našoj aplikaciji i biće potrebno da pogledamo šta je kod nekada koristio.
* **Testiranje** – Jedan od najlakših načina za testiranje sa realnim podacima je periodično osvežavanje test servera sa najnovijim podacima iz produkcije. Ako pravimo backup-ove, to je lako, samo vratimo backup na test server.

**Vrste rezervnih kopija i oporavka**

MySql nudi niz strategija rezervnih kopija od kojih možete izabrati metode koje najbolje odgovaraju vašim zahtevima.

**Fizičke (neobrađene) u odnosu na logičke rezervne kopije**

**Fizičke rezervne kopije** se sastoje od sirovih kopija direktorijuma i datoteka u kojima se čuva sadržaj baze podataka. Ova vrsta rezervne kopije je pogodna za velike, važne baze podataka koje je potrebno brzo oporaviti kada se pojave problemi.

**Logičke rezervne kopije** čuvaju informacije koje su predstavljene kao logička struktura baze podataka ([CREATE DATABASE](https://dev.mysql.com/doc/refman/8.0/en/create-database.html), [CREATE TABLE](https://dev.mysql.com/doc/refman/8.0/en/create-table.html) statements) i sadržaj (INSERT statement ili datoteke sa razgraničenim tekst fajlovima). Ova vrsta rezervne kopije je pogodna za manje količine podataka gde možete da izmenite vrednosti podataka ili strukturu tabele, ili da ponovo kreirate podatke na mašini druge arhitekture.

Metode fizičke rezervne kopije imaju ove karakteristike:

* Rezervna kopija se sastoji od tačnih kopija direktorijuma i datoteka baze podataka. Obično je ovo kopija celog ili dela MySql direktorijuma podataka.
* Metode fizičke rezervne kopije su brže od logičnih jer uključuju samo kopiranje datoteka bez konverzije.
* Izlaz(engl. output) je kompaktniji nego za logičku rezervnu kopiju.
* Pošto su brzina i kompaktnost rezervnih kopija važni za zauzete, važne baze podataka, MySql Enterprise Backup proizvod obavlja fizičke rezervne kopije.
* Granularnost pravljenja rezervnih kopija i vraćanja se kreće od nivoa celog direktorijuma podataka do nivoa pojedinačnih individualnih fajlova. Ovo može ili ne mora da obezbedi granularnost na nivou tabele, u zavisnosti od mehanizma za skladištenje. Na primer, InnoDB tabele mogu da budu u zasebnoj datoteci ili da dele skladište datoteka sa drugim InnoDB tabelama; svaka MyISAM tabela jedinstveno odgovara skupu fajlova.
* Pored baza podataka, rezervna kopija može uključivati sve povezane fajlove kao što su log ili konfiguracioni fajlovi.
* Teško je napraviti rezervnu kopiju za podatke iz MEMORY tabela na ovaj način jer se njihov sadržaj ne čuva na disku. (Proizvod MySql Enterprise Backup ima funkciju gde možete da preuzmete podatke iz MEMORY tabela tokom pravljenja rezervne kopije.)
* Rezervne kopije su prenosive samo na druge mašine koje imaju identične ili slične hardverske karakteristike.
* Pravljenje rezervnih kopija se može izvršiti dok MySql server nije pokrenut. Ako server radi, potrebno je izvršiti odgovarajuće zaključavanje kako server ne bi menjao sadržaj baze podataka tokom pravljenja rezervne kopije. MySql Enterprise Backup to zaključava automatski za tabele koje to zahtevaju.
* Alati za fizičku rezervnu kopiju uključuju **mysqlbackup**, MySql Enterprise Backup za InnoDB ili bilo koje druge tabele, ili komande na nivou sistema datoteka (kao što su **cp**, **scp**, **tar**, **rsync**) za MyISAM tabele.

Za vraćanje (restore):

* MySql Enterprise Backup vraća InnoDB i druge tabele za koje je napravio rezervnu kopiju.
* ndb\_restore vraća NDB tabele.
* Fajlovi kopirani na nivou fajl sistema mogu se kopirati nazad na svoje originalne lokacije pomoću fajl sistem komandi.

Metode logičke rezervne kopije imaju sledeće karakteristike:

* Pravljenje rezervne kopije se vrši ispitivanjem MySql servera radi dobijanja informacija o strukturi baze podataka i sadržaju.
* Rezervna kopija je sporija od fizičkih metoda jer server mora da pristupi informacijama baze podataka i da ih konvertuje u logički format. Ako je izlaz napisan na strani klijenta, server ga takođe mora poslati programu rezervne kopije.
* Izlaz je veći nego za fizičku rezervnu kopiju, posebno kada se čuva u tekstualnom formatu.
* Granularnost pravljenja rezervnih kopija i vraćanja dostupna je na nivou servera (sve baze podataka), nivou baze podataka (sve tabele u određenoj bazi podataka) ili nivou tabele. Ovo je tačno bez obzira na mehanizam za skladištenje.
* Rezervna kopija ne uključuje log i konfiguracione fajlove, ili druge fajlove u vezi sa bazom podataka koje nisu deo baza podataka.
* Rezervne kopije uskladištene u logičkom formatu su nezavisne od mašina i visoko prenosive.
* Logičke rezervne kopije se izvode kada je MySql server pokrenut. Server nije isključen.
* Logički alati za pravljenje rezervnih kopija uključuju program [**mysqldump**](https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html) i naredbu SELECT ... INTO OUTFILE. Oni rade za bilo koji mehanizam za skladištenje podataka, čak i za MEMORY.
* Da biste vratili logičke rezervne kopije, fajlovi dump-a u Sql formatu mogu se obraditi pomoću MySql klijenta. Da bi se učitali fajlovi sa razgraničenim tekstom, koristi se naredba LOAD DATA ili mysqlimport klijent.

### Online vs Offline Rezervne kopije

Online rezervne kopije se dešavaju dok je MySql server pokrenut, tako da se informacije baze podataka mogu dobiti sa servera. Offline rezervne kopije se dešavaju dok je server zaustavljen. Ova razlika se takođe može opisati kao hot” naspram cold” rezervne kopije; warm“ rezervna kopija je ona u kojoj server ostaje da radi, ali je zaključan od modifikacije podataka dok eksterno pristupate fajlovima baze podataka.

Metode online rezervnih kopija imaju ove karakteristike:

* Rezervna kopija je manje nametljiva za druge klijente, koji mogu da se povežu sa MySQL serverom tokom pravljenja rezervne kopije i mogu da pristupe podacima u zavisnosti od toga koje operacije treba da izvrše.
* Mora se voditi računa da se nametne odgovarajuće zaključavanje tako da ne dođe do modifikacija podataka koje bi ugrozile integritet rezervne kopije. MySql Enterprise Backup proizvod vrši takvo zaključavanje automatski.

Metode offline rezervnih kopija imaju ove karakteristike:

* Na klijente to može negativno uticati jer server nije dostupan tokom pravljenja rezervnih kopija. Iz tog razloga, takve rezervne kopije se često uzimaju iz replike koja se može preuzeti van mreže bez narušavanja dostupnosti.
* Procedura rezervne kopije je jednostavnija jer ne postoji mogućnost ometanja aktivnosti klijenta.

Veća je verovatnoća da će klijenti biti pogođeni onlajn oporavkom nego onlajn rezervnom kopijom jer oporavak zahteva jače zaključavanje. Tokom pravljenja rezervne kopije, klijenti će možda moći da čitaju podatke dok se prave rezervne kopije. Oporavak menja podatke i ne samo ih čita, tako da klijentima mora biti onemogućen pristup podacima dok se oni vraćaju(engl restore).

Lokalni i remote backup-ovi

Lokalna rezervna kopija se vrši na istom hostu na kome radi MySql server, dok se remote rezervna kopija radi sa drugog hosta. Za neke vrste rezervnih kopija, rezervna kopija se može pokrenuti sa remote hosta čak i ako je izlaz napisan lokalno na serveru .host.

* [**mysqldump**](https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html) može da se poveže sa lokalnim ili remote serverima. Za Sql output(CREATE i INSERT) mogu se napraviti lokalni ili udaljeni dumpovi i generisati izlaz na klijentu. Za izlaz razdvojenog teksta (sa --tab opcijom), datoteke sa podacima se kreiraju na hostu servera.
* SELECT ... INTO OUTFILE se može pokrenuti sa lokalnog ili udaljenog klijenta, ali se izlazna datoteka kreira na hostu servera.
* Metode fizičke rezervne kopije se obično pokreću lokalno na hostu MySql servera tako da server može da se prebaci van mreže, iako odredište za kopirane datoteke može biti remote.

Snapshot Backup-ovi

Neke implementacije fajl sistema omogućavaju pravljenje snapshot-a“. Oni obezbeđuju logičke kopije sistema datoteka u datom trenutku, bez potrebe za fizičkom kopijom celog fajl sistema. (Na primer, implementacija može da koristi tehnike kopiranja pri pisanju tako da samo delovi sistema datoteka modifikovani posle vremena snimanja moraju da se kopiraju.) Sam MySql ne pruža mogućnost za pravljenje snimaka sistema datoteka. Dostupan je preko rešenja nezavisnih proizvođača kao što su Veritas, LVM ili ZFS.

### Full i Incremental Backup-ovi

Potpuna rezervna kopija uključuje sve podatke kojima upravlja MySql server u datom trenutku. Inkrementalna rezervna kopija se sastoji od promena napravljenih u podacima tokom datog vremenskog perioda (od jedne do druge tačke u vremenu). Inkrementalne rezervne kopije su omogućene omogućavanjem binarnog logova servera, koji server koristi za beleženje promena podataka.

**Potpuni i inkrementalni oporavak**

Potpuni oporavak vraća sve podatke iz potpune rezervne kopije. Ovo vraća instancu servera u stanje koje je imala kada je napravljena rezervna kopija. Ako to stanje nije dovoljno aktuelno, potpuni oporavak može biti praćen oporavkom inkrementalnih rezervnih kopija napravljenih od pune rezervne kopije, kako bi se server doveo u ažuriranije stanje.

Inkrementalni oporavak je oporavak promena napravljenih tokom datog vremenskog perioda. Ovo se takođe naziva oporavak u trenutku, jer čini stanje servera aktuelnim do određenog vremena. Oporavak u trenutku zasniva se na binarnom logu i obično sledi potpuni oporavak iz datoteka rezervnih kopija koji vraća server u stanje kada je rezervna kopija napravljena. Zatim se promene podataka zapisane u binarnim log fajlovima evidencije primenjuju kao inkrementalni oporavak da bi se ponovo izvršile izmene podataka i doveo server do željenog trenutka.

Planiranje rezervnih kopija(engl. Backup Scheduling), kompresija i šifrovanje

Planiranje rezervnih kopija je dragoceno za automatizaciju procedura pravljenja rezervnih kopija. Kompresija izlaza rezervne kopije smanjuje zahteve za prostorom, a šifrovanje izlaza obezbeđuje bolju sigurnost od neovlašćenog pristupa podacima napravljenim rezervnom kopijom. Sam MySql ne pruža ove mogućnosti. MySql Enterprise Backup proizvod može komprimovati InnoDB rezervne kopije, a kompresija ili šifrovanje izlaza rezervne kopije se može postići korišćenjem uslužnih programa fajl sistema.

# Alati za backup i oporavak

Dostupni su raznovrsni alati za pravljenje backup-a. Omiljeni su mylvmbackup za LVM snapshot backup-ovde i Percona XtraBackup (otvorenog koda) ili MySQL Enterprise Backup za „hot“ InnoDB backup-ove. Ne preporučuje se mysqldump za pravljenje backup-a bilo koje značajne količine podataka, zbog uticaja na server i nepredvidivo dugog vremena za povratak.

## MySQL Enterprice Backup

Ovaj alat, ranije poznat kao InnoDB Hot Backup ili ibbackup, je deo pretplate za MySQL Enterprice. Njegovo korišćenje ne zahteva zaustavljanje MySQL-a, postavljanje ključeva ili prekidanje normlne aktivnosti baze podataka (mada će to izazvati dodatno opterećenje na serveru). On podržava funkcije kao što su kompresovani backup-ovi i inkrementalni backup-ovi. Ovo je zvanični backup alat za MySQL.

## Percona XtraBackup

Percona XtraBackup je sličan MySQL Enterprice Backup-u na više načina, ali je on otvorenog koda i besplatan je. Pored osnovnog alata za izradu backupa, takođe postoji wrapper skripta pisana u Perlu koja poboljšava njegove funkcionalnosti za naprednije zadatke. On podržava funkcije kao što su protočne, inkrementalne, kompresovane i višenitne (paralelne) backup operacije. Takođe, on ima razne specijalne karakteristike za smanjenje uticaja backup-ova na opterećenim sistemima.

Percona XtraBackup funkcioniše tako što prati u stopu InnoDB log fajlove u pozadinskoj niti, a onda kopira InnoDB fajlove podataka. Ovaj proces uključuje posebne provere kako bi se obezbedilo da se podaci kopiraju konzistentno. Kada se svi fajlovi podataka kopiraju, nit za kopiranje loga se takođe završava. Rezultat je kopija svih podataka, ali u različitim vremenskim trenucima. Logovi sada mogu da se primene na fajlove podataka, korišćenjem InnoDB rutina oporavka od katastrofa, kako bi se svi fajlovi podataka doveli u konzistentno stanje. To se naziva procesom pripreme. Kada se jednom pripremi, backup je potpuno konzistentan i sadrži sve izvršene transakcije od krajnje tačke procesa kopiranja fajla. Sve ovo se dešava potpuno eksterno na MySQL-u, tako da nije potrebno da se ni na koji način konektujemo ili pristupamo MySQL-u.

## mylvmbackup

mylvmbackup je Perl skripta koja pomaže da se automatizuju MySQL backup-ovi preko LVM snapshot-a. On dobija globalni ključ za čitanje, kreira snapshot i oslobađa ključ. Onda kompresuje podatke pomoću tar i briše snapshot. On ima nekoliko naprednijih opcija, ali uopšteno je jednostavan alat za izvođenje LVM backup-ova.

## Zmanda Recovery Manager

Zmanda Recovery Manager za MySQL ili ZMR dolazi i u besplatnoj i u komercijalnoj verziji. U poslovnom izdanju dolazi sa menadžement konzolom koja pruža grafički web interfejs za konfigurisanje, izradu backupa, proveru, oporavak, izveštavanje i planiranje. Izdanje otvorenog koda nije ni na koji način uskraćeno, ali ne uključuje neke dodatke, kao što je web konzola. ZMR je zapravo menadžer za izradu rezervnih kopija i oporavka, a ne samo jedan alat. On umotava svoju funkcionalnost oko standardnih alata i tehnika, kao što su mysqldump, LVM snapshot-ovi i Percona XtraBackup. On automatizuje mnogo napornog rada pravljenja rezervnih kopija i oporavka.

## mydumper

Nekoliko MySQL inženjera su napravili mydumper kao zamenu za mysqldump. On je skup alata za višenitno (paralelno) pravljenje backupa i oporavka za MySQL sa mnogo dobrih karakteristika. Mnogi ljudi smatraju brzinu višenitne izrade backup-ova i vraćanja kao najatraktivniju karakteristiku ovog alata. Malo ljudi ga koristi u produkciji.

## mysqldump

Većina ljudi koristi programe koji se isporučuju sa MySQL-om, tako da je uprkos nedostacima, mysqldump najčešći izbor za krieranje logičkih backup-ova podataka i šema. Ovo je alat opšte namene koji može da se koristi za mnoge zadatke, kao što je kopiranje tabele sa jednog servera na drugi, što je prikazano na slici 32.



*Slika 32*

* Pravljenje logičkog backup-a svega na serveru u jedan fajl, za sve tabele u svakoj bazi podataka napraviti backup u istom vremenskom trenutku:



*Slika 33*

* Pravljenje logičkog backup-a samo jedne baze podataka:



*Slika 34*

* Pravljenje logičkog backup-a samo jedne tabele:



*Slika 35*

Podrazumevane opcije za mysqldump nisu dobre za većinu namena backup-a. Možemo da specifiramo neke opcije za promenu izlaza. Ovo su opcije koje se često koriste da bi mysqldump bio efikasniji i da bi njegov izlaz bio lakši za upotrebu:

* --opt – Omogućuje grupu opcija koje onemogućavaju baferovanje i upis više podataka u manje SQL naredbi u kopiji kako bi efikasnije učitali podatke. Ako onemogućimo ovu grupu opcija, mysqldump će čuvati svaku tabelu koju odlažemo u memoriju pre nego što je upiše na disk, što je nepraktično za velike tabele.
* --allow-keywords, --quote-names – Omogućavaju odlaganje i vraćanje tabela koje koriste rezervisane reči kao nazive.
* --complete-insert – Omogućava premeštanje podataka između tabela koje nemaju identične kolone.
* --tz-utc – Omogućava premeštanje podataka između servera u različitim vremenskim zonama.
* --lock-all-tables – Koristi FLUSH TABLES WITH READ LOCK za dobijanje globalno konzistentnog backup-a.
* --tab – Odlaže fajlove pomoću SELECT INTO OUTFILE
* --skip-extended-insert – Prouzrokuje da svaki red podataka ima svoju sopstvenu INSERT naredbu. Ovo nam može pomoći da selektivno vraćamo određene redove ako je potrebno.

Ako koristimo mysqldump opcije –databases ili –all-databases, rezultujući odloženi podaci će biti konzistentni u svakoj bazi podataka, jer mysqldump će istovremeno zaključati i odložiti sve tabele u bazi podataka. Međutim, tabele iz različitih baza podataka možda neće biti međusobno konzistentne. Korišćenje opcije –lock-all-tables rešava ovaj problem.

Za pravljenje backup-a InnoDB-a, treba dodati opciju –single-transaction, koja koristi InnoDB-ove MVCC karakteristike za kreiranje konzistentnog backupa u jednom trenutku, umesto da koristi LOCK TABLES. Ako dodamo opciju –master-data, backup će takođe sadržati koordinate binarnog loga servera u trenutku pravljenja backup-a, što je veoma korisno za oporavak u određenom trenutku i podešavanje replika. Međutim, treba da imamo na umu da će on za zamrzavanje servera koristiti FLUSH TABLES WITH READ LOCK da bi mogao da dobije koordinate.

**Backup i restore MySql baze podataka pomoću komande mysqldump**

**Mysqldump** je uslužni program komandne linije koji se koristi za generisanje rezervne kopije MySql logičke baze podataka kao jedne .skl datoteke sa skupom Sql naredbi. Pomoćni program vam pomaže da izbacite MySql tabele, više baza podataka ili njihove objekte. Imajte na umu da nije moguće napraviti rezervnu kopiju MySql baza podataka ili podataka u odvojene .skl datoteke pomoću uslužnog programa miskldump. Za više informacija o tome kako da napravite rezervnu kopiju MySql baza podataka i podataka, pogledajte Različiti načini pravljenja rezervnih kopija MySql baza podataka i tabela.

Osim toga, morate imati pristup pokrenutom MySql serveru da biste koristili komande miskldump i miskl.

Sada je vreme da vidimo kako da vratite MySql bazu podataka. Prvo morate da kreirate praznu MySql bazu podataka, a zatim da vratite MySql dump datoteku.

Korak 1: Kreirajte bazu podataka

U komandnoj liniji povežite se sa MySql serverom na kojem želite da kreirate bazu podataka i pokrenite komandu miskl:

Napomena: Ime baze podataka treba da bude isto kao i baza podataka koju želite da vratite.

mysql> create database sakila;

Korak 2: Vratite bazu podataka

Sada možemo da vratimo MySql bazu podataka iz dump datoteke. Sintaksa za komandu je sledeća:

mysql -u [user name] –p [target\_database\_name] < [dumpfilename.sql]

Parametri uključuju:

-u [ime korisnika] je korisničko ime za povezivanje sa MySql serverom.

-p je lozinka za korisničko ime koje koristite za povezivanje sa serverom.

target\_database\_name je ime prazne baze podataka u koju želite da učitate podatke iz datoteke rezervne kopije.

< je parametar koji se odnosi na proces vraćanja baze podataka.

[dumpfilename.skl] je putanja do dump datoteke.

Dakle, da biste vratili sakila bazu podataka, izvršite sledeću komandu:

mysql --host=dbfmylast --user=root --port=3306 -p sakila < D:\sakila.sql

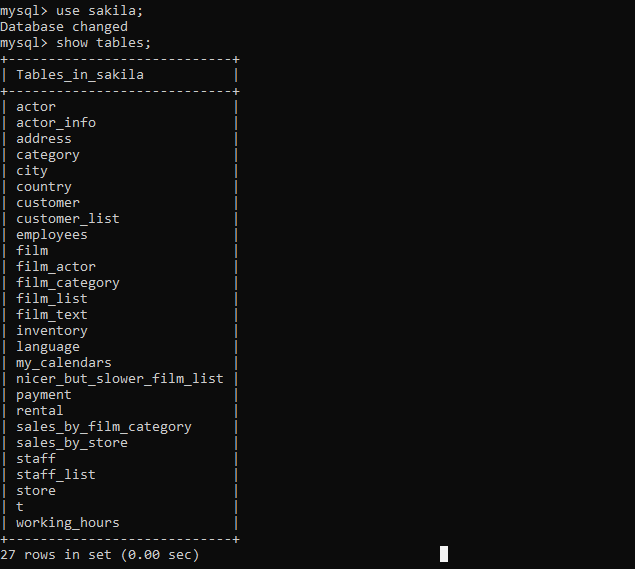
gde je dbfmilast ciljni server sa praznom bazom podataka u koju ćemo učitati podatke iz sakila.skl dump datoteke.

mysql> use sakila;

...

mysql> show tables;

U izlazu ćete videti listu tabela koje se nalaze u sakila bazi podataka.



Vratite sve baze podataka u MySql iz dump datoteke koristeći miskldump

Pomoću uslužnog programa miskldump možete vratiti ne samo jednu bazu podataka već i sve baze podataka na serveru na veliko.

Sintaksa za vraćanje svih MySql baza podataka iz dump datoteke je sledeća:

miskl -u root -p < alldatabases.skl

gde je alldatabases.skl putanja do dump datoteke koja sadrži rezervnu kopiju svih baza podataka na serveru.

Na primer, vratićemo sve MySql baze podataka iz backup\_all\_databases.skl damp datoteke. Za ovo, pokrećemo sledeću komandu:

miskl -u root -p < D:\backup\_all\_databases.skl

Koristite miskldump da vratite jednu tabelu iz dump datoteke

Pod nekim okolnostima, može biti potrebno da vratite određenu MySql tabelu iz baze podataka – kada je neko ispustio tabelu ili prepisao njene podatke.

U svrhu demonstracije, napravite rezervnu kopiju tabele vork\_hours iz sakila baze podataka tako što ćete izvršiti naredbu miskldump. Datoteka backup\_vorking\_hours\_table.skl je izlazna datoteka koja će sadržati rezervnu kopiju tabele.

miskldump --host=dbfmilast --user=root --port=3306 -p sakila radni\_sati > D:\backup\_vorking\_hours\_table.skl

Nakon što je rezervna kopija napravljena, ispustite tabelu vork\_hours sa naredbom DROP TABLE.

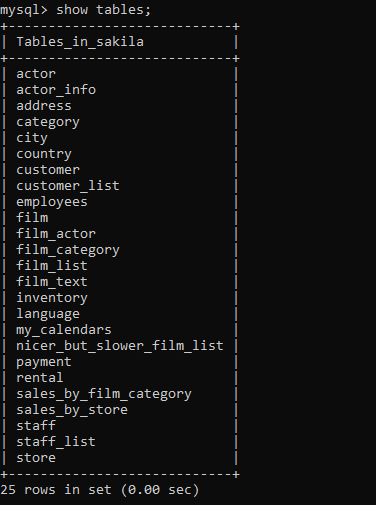
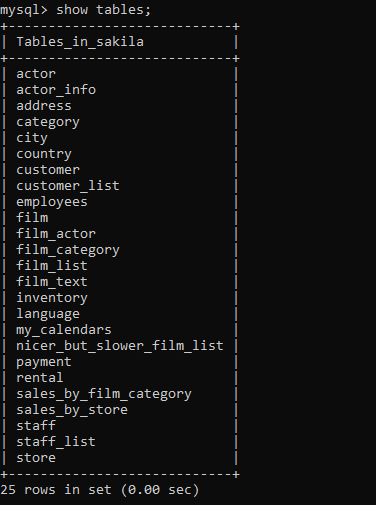
miskl> ispusti tabelu vork\_hours;

Proverite da li je tabela uklonjena iz baze podataka sakila:

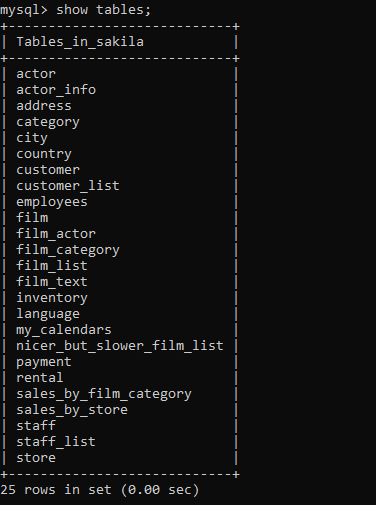
miskl> use sakila;

...

miskl> prikaži tabele;



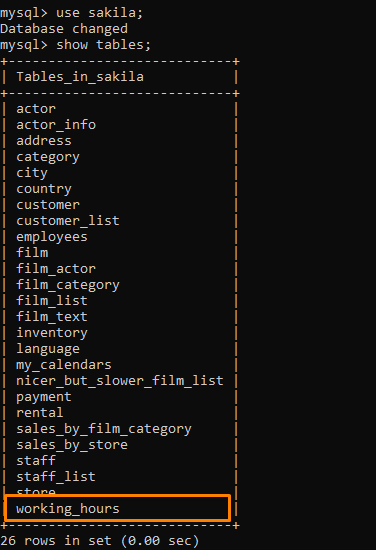
Komanda navodi MySql tabele dostupne u bazi podataka. Kao što vidite, ne postoji tabela radnih\_časova.



Sada vratite rezervnu kopiju tabele vork\_hours u sakila bazu podataka tako što ćete izvršiti komandu miskl:

miskl --host=dbfmilast --user=root --port=3306 -p sakila < D:\backup\_vorking\_hours\_table.skl

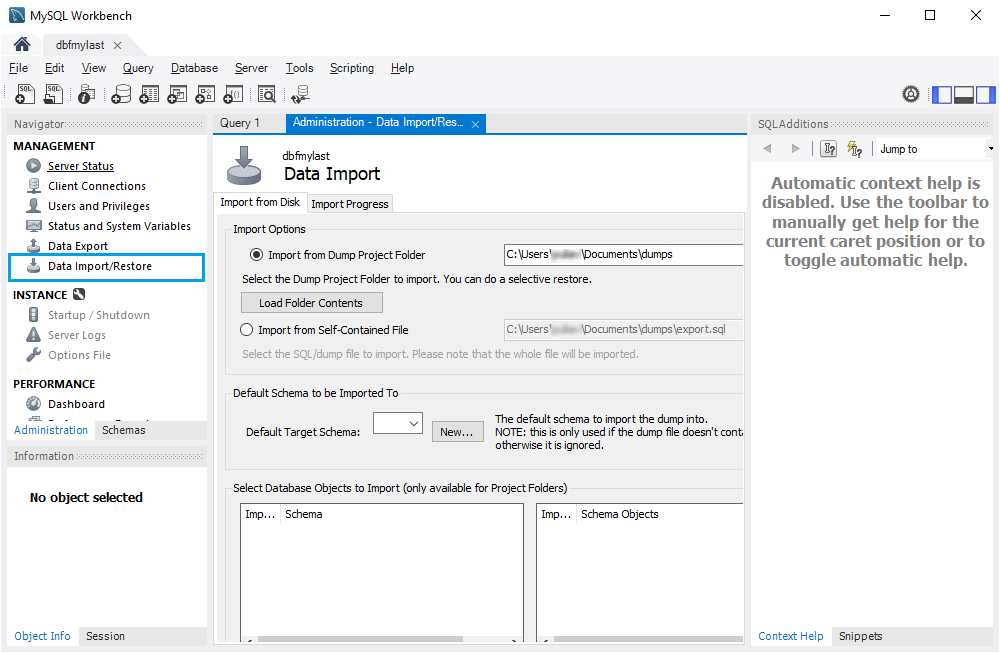
Tabela će biti uvezena u sakila bazu podataka:



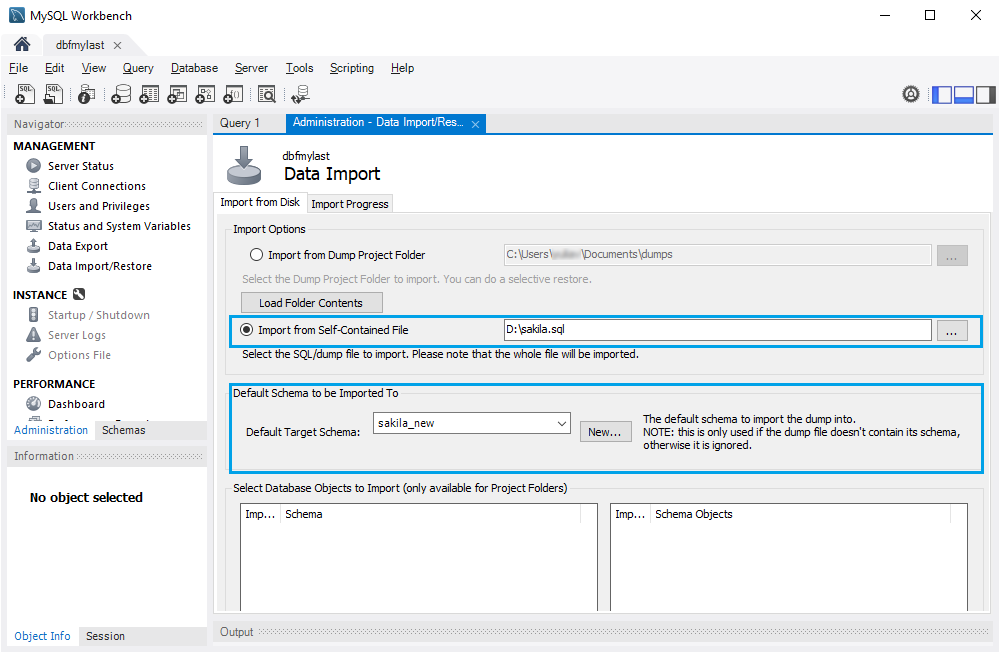
Vratite MySql bazu podataka iz .skl datoteke koristeći Vorkbench

Ovaj odeljak objašnjava kako da vratite MySql bazu podataka iz .skl datoteke koristeći MySql Vorkbench alat. Pre vraćanja baze podataka, napravite praznu bazu podataka na serveru na koji želite da se povežete.

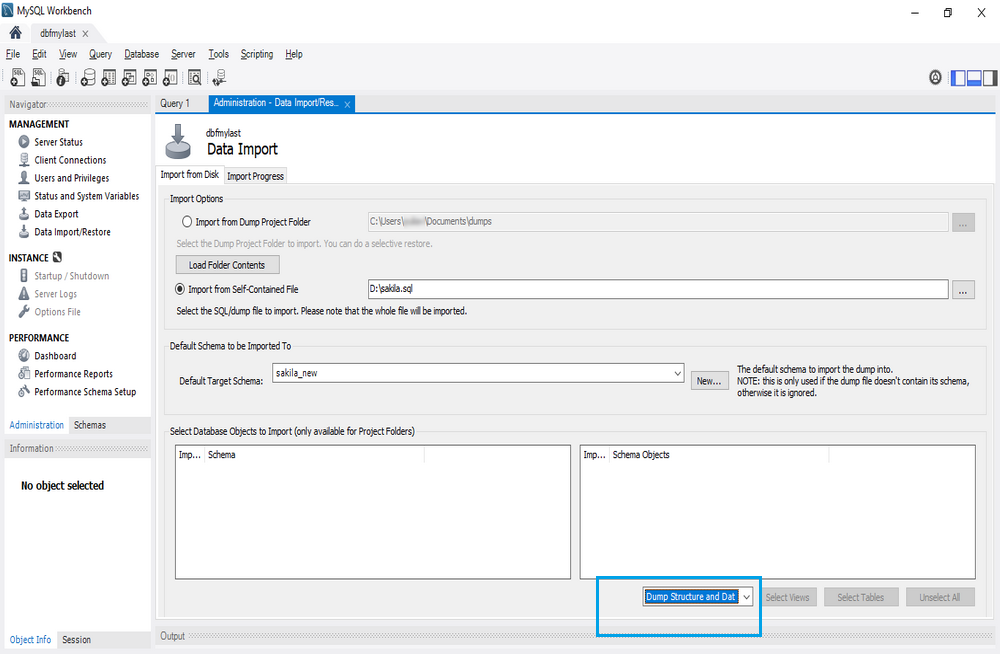
Otvorite MySql Workbench i kliknite na Data Import/Restore u Navigatoru.



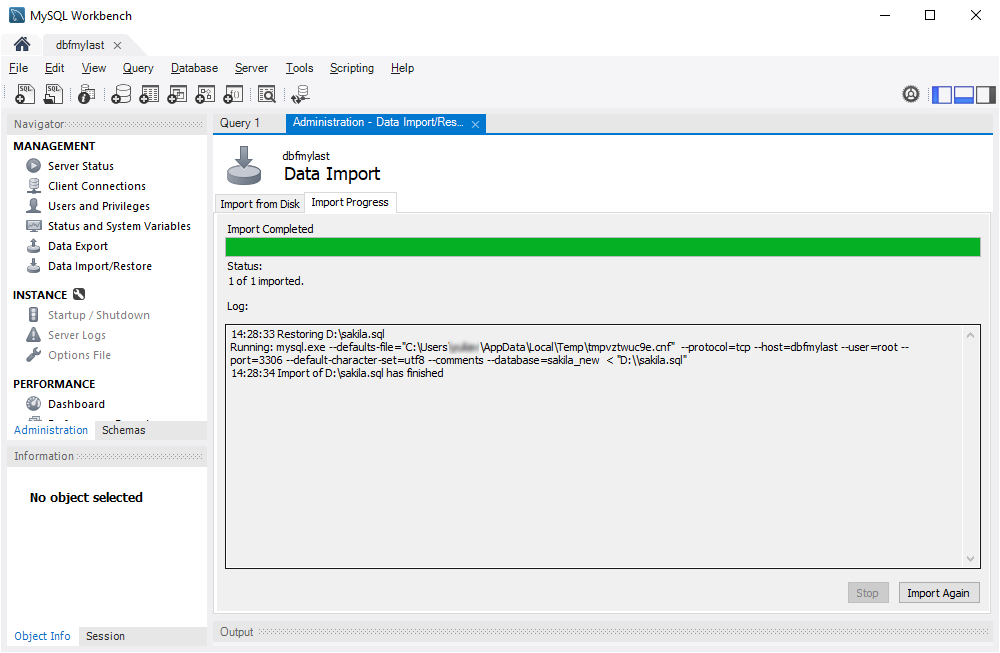
U dokumentu Administracija – uvoz/vraćanje podataka izaberite Uvezi iz samostalne datoteke i potražite .skl datoteku koja sadrži rezervnu kopiju sakila baze podataka. Zatim, u okviru Podrazumevana šema za uvoz, izaberite ciljnu bazu podataka u koju će biti učitana rezervna kopija izvorne baze podataka sa njenom strukturom i podacima.



Na dnu dokumenta izaberite Dump Structure and Data sa padajuće liste da biste vratili strukturu baze podataka i podatke.



Kada su svi parametri podešeni, pređite na karticu Napredak uvoza i kliknite na Start Import. Nakon što je uvoz završen, možete zatvoriti alatku.



**Zaključak**

Važno je napraviti rezervnu kopiju svojih baza podataka kako biste mogli da oporavite svoje podatke i da sistem bude ponovo u funkciji u slučaju da dođe do problema, kao što su pad sistema, kvarovi na hardveru ili korisnici koji greškom brišu podatke. Rezervne kopije su takođe neophodne kao zaštita pre nadogradnje MySql instalacije i mogu se koristiti za prenos MySql instalacije na drugi sistem ili za postavljanje replika servera.

**Literatura**

[**https://dev.mysql.com/doc/refman/8.0/en/backup-types.html**](https://dev.mysql.com/doc/refman/8.0/en/backup-types.html)

**Effective MySql: Backup and Recovery -Ronald Bradford**

High performance MySQL - Baron Schwartz, Peter Zaitsev, and Vadim Tkachenko

<https://blog.devart.com/how-to-restore-mysql-database-from-backup.html>

<https://www.sqlshack.com/how-to-backup-and-restore-mysql-databases-using-the-mysqldump-command/>