



UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET



SIGURNOST MySQL BAZE PODATAKA

Seminarski rad

Predmet: Sistemi za upravljanje bazama podataka

Student:

Marko Kocić, br. ind. 1478

Mentor:

Doc. dr Aleksandar Stanimirović

Niš, maj 2023. godina

Sadržaj

UVOD.....	3
GENERALNE PREPORUKE I SMERNICE.....	4
Smernice koje se odnose na klijentsko programiranje.....	6
KREIRANJE I UPRAVLJANJE NALOZIMA.....	8
Bezbednost lozinki.....	11
Zaključavanje naloga.....	13
Ograničenje resursa naloga.....	16
SISTEMSKE PROMENLJIVE I OPCIJE.....	19
Bezbednosna razmatranja za <i>LOAD DATA LOCAL</i>	22
ZAKLJUČAK.....	28
LITERATURA.....	29

1. UVOD

Kako su napadi sve češći, nije bitna činjenica samo da nešto funkcioniše, već da je to nešto bez bezbednosnih propusta, potpuno sigurno, bez mogućnosti kompromitovanja podataka. Bezbednost je holistička, što bi značilo da nije dovoljno imati samo aplikaciju čiji kod nije ranjiv, već je neophodno nemati bezbednosnih propusta niti u hardveru, niti u operativnom sistemu, kao ni u mreži. Kada svaka navedena komponenta nije ranjiva, kao i veze između njih, tada možemo reći da posedujemo jedan bezbedan sistem. Napadaču je dovoljan pronalazak samo jednog ranjivog toka i već može naneti štetu sistemu. Treba biti svestan činjenice da je bezbednost proces, a ne proizvod i da od „najbezbednijeg” sistema postoji bezbedniji. Iako je sistem danas bezbedan, to ne znači da će i sutra biti.

Kada se razmišlja o bezbednosti *MySQL* baze podataka, potrebno je razmotriti širok spektar mogućih tema i njihov uticaj na sam *MySQL* server:

- Opšti faktori koji utiču na bezbednost – izbor jakih lozinki, davanje samo neophodnih privilegija korisnicima („*least privilege*” princip), sprečavanje *SQL injection*-a (injektiranja *SQL* naredbi) i *data corruption*-a (oštećenja podataka).
- Sigurnost same instalacije – datoteke sa podacima, *log* datoteke, kao i sve ostale datoteke instalacije treba biti zaštićene od neovlašćenog (neautorizovanog) čitanja ili pak pisanja po istim.
- Kontrola pristupa i bezbednost unutar samog sistema baze podataka – korisnicima omogućen pristup određenim bazama podataka, pogledima, kao i uskladištenim programima.
- *Feature*-i (dodatne funkcije/karakteristike) koji su u ponudi u okviru *plugin*-ova koji se odnose na sigurnost.
- Mrežna bezbednost *MySQL*-a i samog sistema – bezbednost se odnosi na odobrenja koja se daju individualnim korisnicima, ali i na restrikcije koje mogu postojati u vidu dostupnosti *MySQL*-a samo na čvoru na kome se nalazi *MySQL* server ili pak na ograničenom skupu drugih čvorova.
- Adekvatne i odgovarajuće rezervne kopije datoteka baze podataka, konfiguracija, kao i *log* datoteka – treba imati i rešenje za oporavak radi uspešnog povratka informacija iz *backup*-ova.

2. GENERALNE PREPORUKE I SMERNICE

Treba biti u toku sa savetima po pitanju bezbednosti radi izbegavanja čestih grešaka. Potrebno je razmotriti zaštitu celog servera (a ne samo *MySQL* servera) od svih vrsta napada (kako pasivnih, tako i aktivnih) – *eavesdropping* (prisluškivanje), *altering* (menjanje podataka), *playback* (reprodukcija), kao i *denial of service* napada. Sigurnost se zasniva na *ACL* listama (*Access Control Lists*) za sve konekcije, upita, kao i ostale operacije koje korisnici mogu izvesti. Takođe postoji podrška za *SSL* šifrovanu konekciju između *MySQL* klijenta i servera. Generalno, mnogi koncepti koji će biti pomenuti u ovom radu su opšti, nisu specifični samo za *MySQL*.

Nakon pokretanja *MySQL*-a, pratiti sledeće smernice:

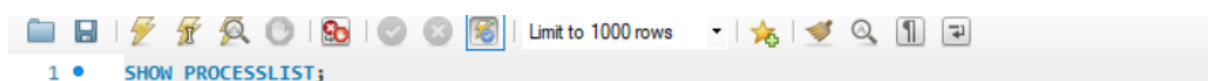
- Ne davati nikome (osim *MySQL root* nalozima) pristup sistemskoj tabeli *user*.
- Koristiti *GRANT* i *REVOKE* naredbe za kontrolu pristupa *MySQL*-u (naredbe za davanje i ukidanje privilegija korisničkim nalozima). Ne davati veće privilegije nego što je potrebno. Nikada ne davati privilegije svim *host*-ovima.
- *Root* korisnik je korisnik sa svim privilegijama i neophodno je da njegova lozinka uvek bude tražena (ne smemo se uspešno konektovati na server bez iste). *Root* korisnik je sistemski nalog i može izvesti bilo koju operaciju.
- Koristiti naredbu *SHOW GRANTS* radi provere privilegija različitih korisnika i po potrebi ukinuti one koje nisu neophodne (pozivom naredbe *REVOKE*).
- Ne čuvati lozinke u bazi podataka kao *plaintext*, već koristiti *hash* funkciju (primera radi *SHA-2*) na čiju se izlaznu vrednost dodaje *salt* i onda se ta vrednost ponovo *hash*-ira – *hash(hash(password)+salt)*.
- Lozinka treba biti dovoljno duga da se metodom grube sile (*brute-force*) ne može doći do iste. Takođe treba se sastojati od malih i velikih slova, brojeva i specijalnih karaktera koji nemaju nikakvog smisla u vezi samog korisnika. Jedini problem ovde je lakoća pamćenja iste.
- Uložiti u *firewall* koji štiti od najmanje 50% svih eksploatacija u bilo kom softveru. Port na kome radi *MySQL* (podrazumevano 3306) treba biti blokiran za korisnika kome se ne veruje.
- Aplikacije koje pristupaju *MySQL*-u ne treba da slepo veruju podacima koje unose korisnici, već je neophodno vršenje određenih provera.
- Ne prenositi podatke kao *plaintext* kroz mrežu, već ih šifrovati.

Prilikom povezivanja sa *MySQL* serverom prenosi se šifra u vidu šifrovanog teksta. Razlog za to je sumnja postojanja treće osobe koja prisluškuje saobraćaj na mreži. Primera radi, pomoću *Wireshark* aplikacije moguće je snimati čitav saobraćaj na ciljanoj mreži. Ukoliko je poruka enkriptovana, u slučaju zapažanja iste, napadač ne bi trebalo da može da je dekriptuje. Obično upiti i rezultati istih putuju kroz mrežu u čitljivom obliku. Ako smatramo da je mreža nepouzdan kanal komunikacije, možemo koristiti *SSL* podršku koju obezbeđuje

sam *MySQL* interno. Alternativno se može koristiti i *SSH* (mrežni protokol) radi dobijanja šifrirane *TCP/IP* konekcije između *MySQL* klijenta i *MySQL* servera.

Kako bismo učinili *MySQL* sistem sigurnim, potrebno je poslušati sledeće predloge:

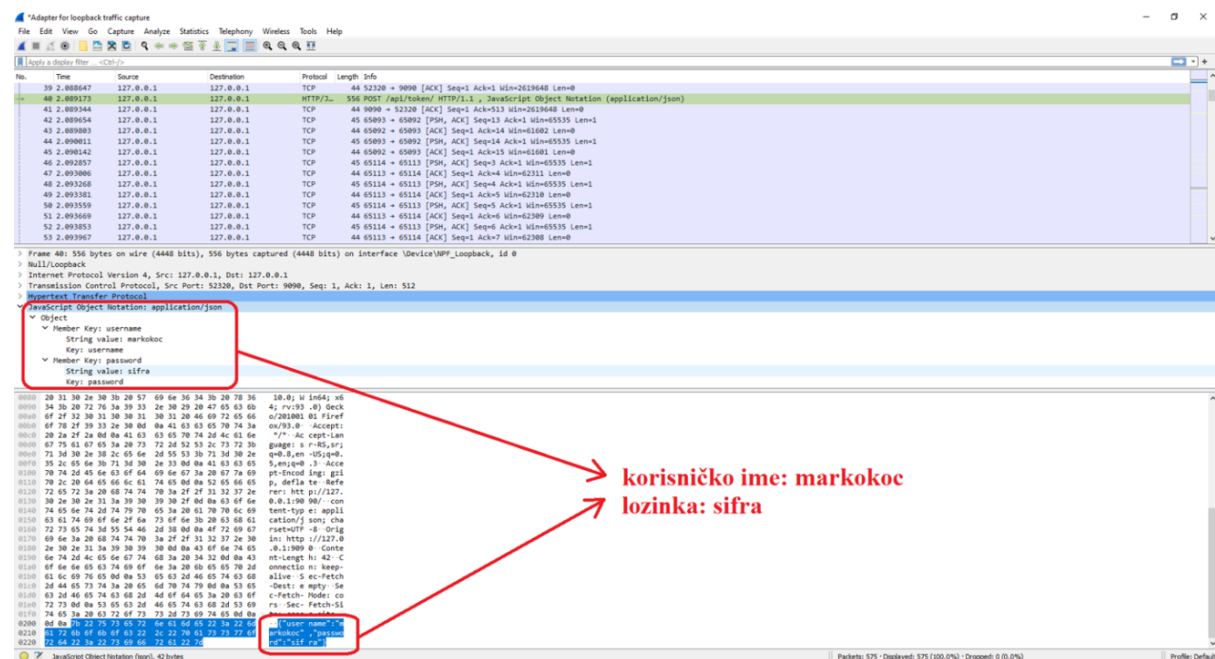
- Zahtevati da svi korisnici imaju lozinku u cilju poimanja servera da je reč o pravom identitetu korisnika.
- Uveriti se da samo *UNIX* korisnički nalozi sa privilegijama čitanja i pisanja pokreću *MySQL* server.
- Nikada ne pokretati *MySQL* server kao *UNIX root* korisnik jer može biti opasno. Predlaže se da ga pokreće običan, neprivilegovani korisnik koji je napravljen za tu namenu. Ovaj nalog bi trebalo da se koristi za administraciju *MySQL*-a.
- Ne dodeljivati *FILE* privilegiju korisnicima koji nisu administratori. Svako ko ima ovu privilegiju može pistati po datotekama i modifikovati ih na čitavom *file system*-u. To uključuje i serverov direktorijum podataka koji sadrži datoteke koje implementiraju tabele privilegija. Takođe, moguće je i čitati i bilo koju tabelu u bazi podataka što nije dobro po pitanju bezbednosti.
- Enkriptovati binarne *log* datoteke radi zaštite potencijalnih osetljivih podataka koji se mogu naći u istim.
- Ne dodeljivati *PROCESS* ili *SUPER* privilegije korisnicima koji nisu administratori. *SUPER* privilegija se može koristiti za okončanje klijentskih konekcija, promenu vrednosti sistemskih promenljivih, kao i za kontrolu replikacije servera. Pomoću naredbe *SHOW PROCESSLIST* mogu se videti sve naredbe svih korisnika koje se trenutno izvršavaju. Rezultat ove naredbe je prikazan sa slici 2.1. na kojoj se vidi da je korisnik *markodic* povezan na bazu, ali da trenutno ne izvršava nijednu naredbu i to već 324 sekunde.
- Ukoliko se ne veruje *DNS* serveru, trebalo bi u tabelama za dodelu privilegija koristiti *IP* adrese umesto imena *host*-ova.



Result Grid Filter Rows: Export: Wrap Cell Content:								
	Id	User	Host	db	Command	Time	State	Info
▶	5	event_scheduler	localhost	NULL	Daemon	849516	Waiting on empty queue	NULL
	47	root	localhost:55880	sys	Sleep	196		NULL
	48	root	localhost:55881	sys	Query	0	init	SHOW PROCESSLIST
	49	markodic	localhost:55882	realparsmodel	Sleep	324		NULL
	50	markodic	localhost:55883	realparsmodel	Sleep	324		NULL

Slika 2.1. Rezultat izvršenja *SHOW PROCESSLIST* naredbe od strane administratora

Na slici ispod je prikazan primer rada *Wireshark*-a ukoliko transportni kanal ne bi bio enkriptovan.



Slika 2.2. Korišćenje *Wireshark*-a radi snimanja mrežnog saobraćaja

2.1. Smernice koje se odnose na klijentsko programiranje

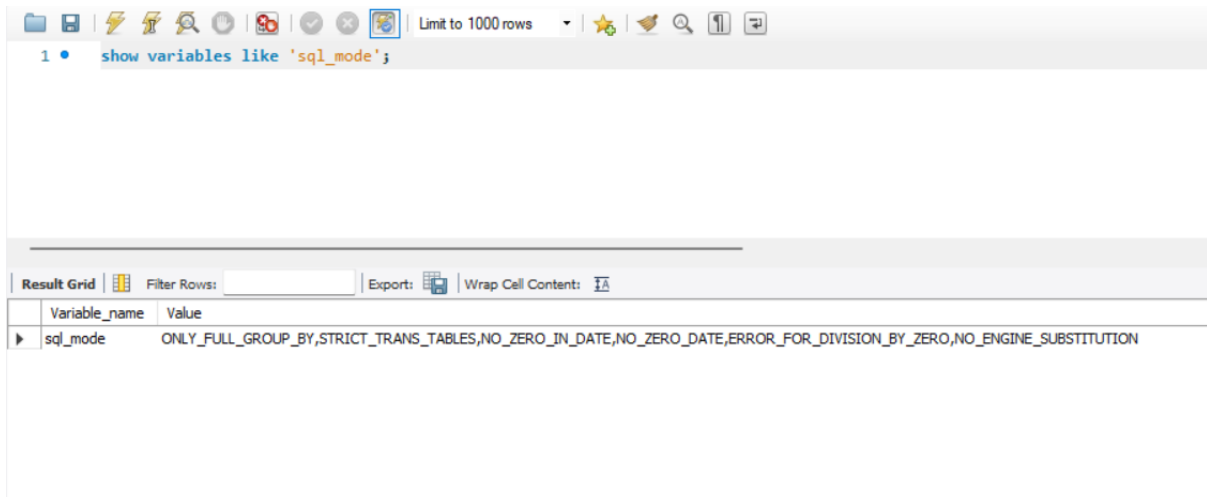
U ovom delu rada biće reči o preporukama koje se odnose na klijentsko programiranje – klijentske aplikacije koje se povezuju na *MySQL* server.

Treba biti oprezan i ne verovati unosima krajnjih korisnika koji mogu biti zlonamerni i pokušati, primera radi, izvođenje *SQL injection*-a u cilju nanošenja štete našem sistemu. Česta greška je zaštita samo *string* vrednosti, a ne recimo i brojevnih vrednosti (pokušaj parsiranja vrednosti), te tako korisnik može nadovezati na x vrednost i *OR 1=1* i da jedna takva naredba bude nesmetano izvršena. Takođe, postoji mišljenje da ne moramo štititi tabele koje sadrže javno dostupne podatke. Ovaj stav nije dobar, jer iako podaci nisu oseljivi i možemo vratiti sve torke korisniku, ipak treba postojati zaštita od recimo *denial of service* napada pri kome server bespotrebno troši resurse i ne može opslužiti legitimne korisnike (napadač je uspeo naneti štetu našem sistemu).

U nastavku su date neke od smernica:

- omogućiti striktan *SQL mode* (podrazumevani režim je prikazan na slici ispod – *STRICT_TRANS_TABLES* je uključen);
- pokušati sa unosom ' ili pak " u svim input poljima aplikacije i ukoliko dobijemo *MySQL* grešku, odmah videti u čemu je problem;
- pokušati sa modifikacijom dinamičkih *URL*-a dodavanjem %22 (") , %23 (#) i %27 (');

- pokušati sa modifikacijom tipova podataka u dinamičkom *URL*-u – sistem bi trebalo da bude otporan na ove napade;
- umesto numeričkih vrednosti, unositi specijalne karaktere – slanje neproverenih vrednosti *MySQL*-u može biti veoma opasno;
- proveriti veličinu podataka pre slanja *MySQL* serveru;
- koristiti *API* u cilju izbegavanja specijalnih znakova koji mogu naneti štetu sistemu.



Slika 2.3. Prikaz vrednosti *SQL mode*-a

Ukoliko se desi neka greška na serveru, korisniku vratiti samo da je došlo do interne greške na serveru, nikako ne izveštavati o detaljima greške (primera radi o kojoj tabeli je reč i slično). Detalji greške se beleže u posebnim datotekama kojima mogu pristupati samo administratori sistema.

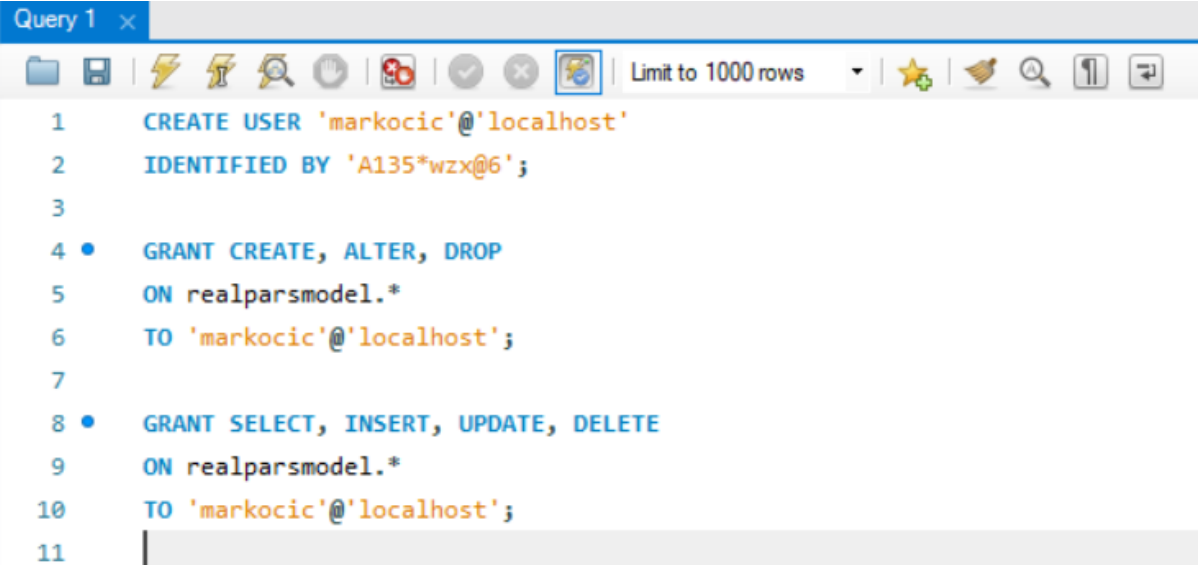
3. KREIRANJE I UPRAVLJANJE NALOZIMA

MySQL omogućava kreiranje naloga koji pruža klijentima povezivanje na server i pristupanje podacima kojima server upravlja. Primarna funkcija *MySQL* sistema privilegija je autentifikacija korisnika koji se povezuje sa datog *host*-a i povezivanje istog sa privilegijama u bazi podataka. Dodatna funkcionalnost uključuje mogućnost davanja privilegija za administrativne operacije.

Postoje neke stvari koje se ne mogu odraditi koristeći *MySQL* sistem privilegija:

- ne može se eksplicitno zabraniti pristup određenom korisniku (naći korisnika, a zatim odbiti konekciju);
- ne može se povezati lozinka sa tačno određenom bazom podataka, tabelom ili pak rutinom, već se lozinka globalno primenjuje na nalog.

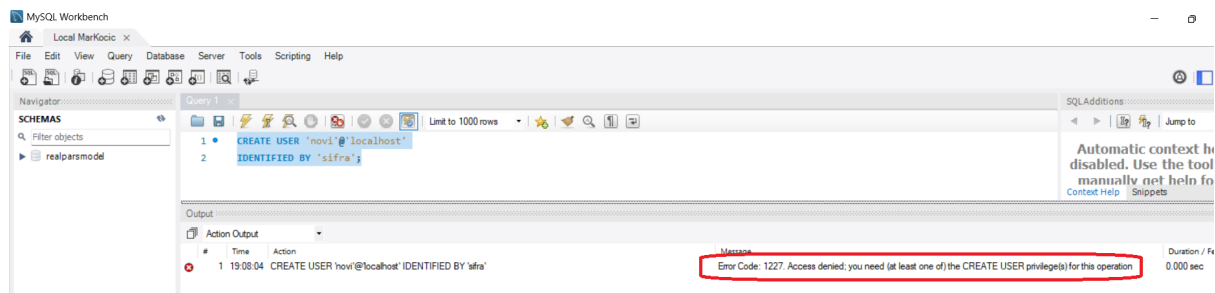
Na slici ispod, prijavljeni smo kao *root* korisnik i imamo privilegije za pravljenje novih naloga. Kreirali smo novog korisnika *markocic* na *localhost* serveru i postavili lozinku *A135*wzx@6*. Nakon toga, dodelili smo mogućnosti kreiranja, izmene i odbacivanja svih tabela u bazi podataka *realparsmodel* (s obzirom da imamo *** nakon naziva baze podataka što ukazuje na sve tabele), kao i mogućnosti selektovanja, dodavanja, ažuriranja i brisanja torki iz svih tabela takođe u prethodno pomenutoj bazi podataka.



```
Query 1 x
1 CREATE USER 'markocic'@'localhost'
2 IDENTIFIED BY 'A135*wzx@6';
3
4 • GRANT CREATE, ALTER, DROP
5 ON realparsmodel.*
6 TO 'markocic'@'localhost';
7
8 • GRANT SELECT, INSERT, UPDATE, DELETE
9 ON realparsmodel.*
10 TO 'markocic'@'localhost';
11
```

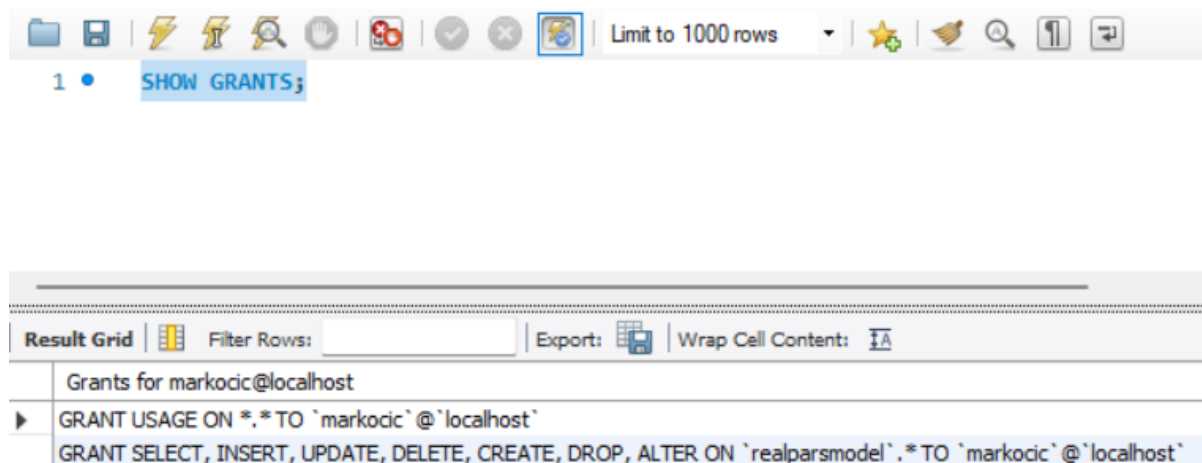
Slika 3.1. Kreiranje novog korisnika i dodela određenih privilegija

Na slici ispod je prikazan neuspešan pokušaj kreiranja još jednog korisnika. Razlog ovakvog ishoda je nepostojanje privilegija za kreiranje korisničkog naloga od strane *markocic*-a.



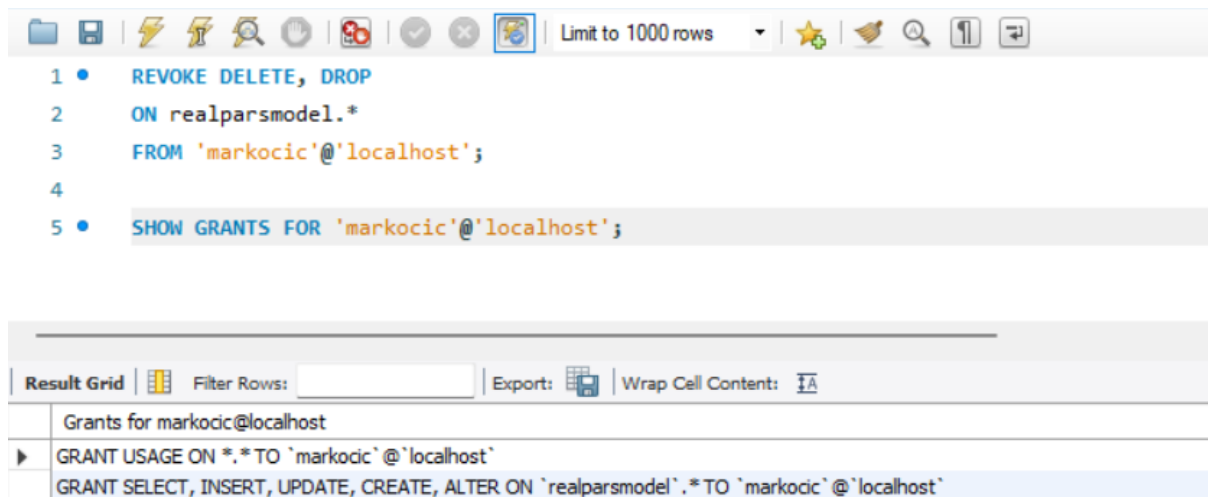
Slika 3.2. Odbijen pristup – nepostojanje privilegija za kreiranje novog korisnika

Na slici ispod je rezultat naredbe *SHOW GRANTS* za novokreiranog korisnika *markocic*. Iz ugla *root* korisnika isto se može vidite pomoću:
`SHOW GRANTS FOR 'markocic'@'localhost';`



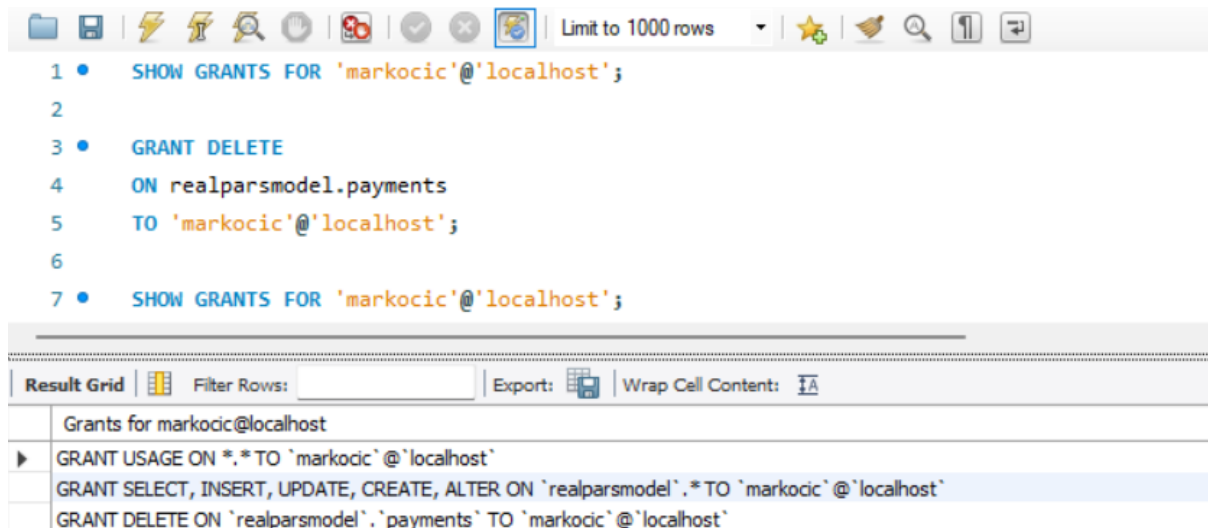
Slika 3.3. Privilegije korisnika *markocic* na serveru *localhost*

S obzirom da se vodimo principom dodele samo neophodnih privilegija i smatramo da korisnik *markocic* ipak ne treba imati privilegije brisanja torki tabela, kao i celih tabela nad bazom podataka *realparsmodel* (recimo da smo pogrešili prilikom inicijalne dodele), hoćemo da određene privilegije ukinemo (oduzmemo). To se postiže *REVOKE* naredbom koja je prikazana na slici ispod. Takođe, vidi se i uticaj same naredbe – korisnik više nema *DELETE* i *DROP* privilegije. Naredba je izvršena od strane sistemskog korisnika *root*.

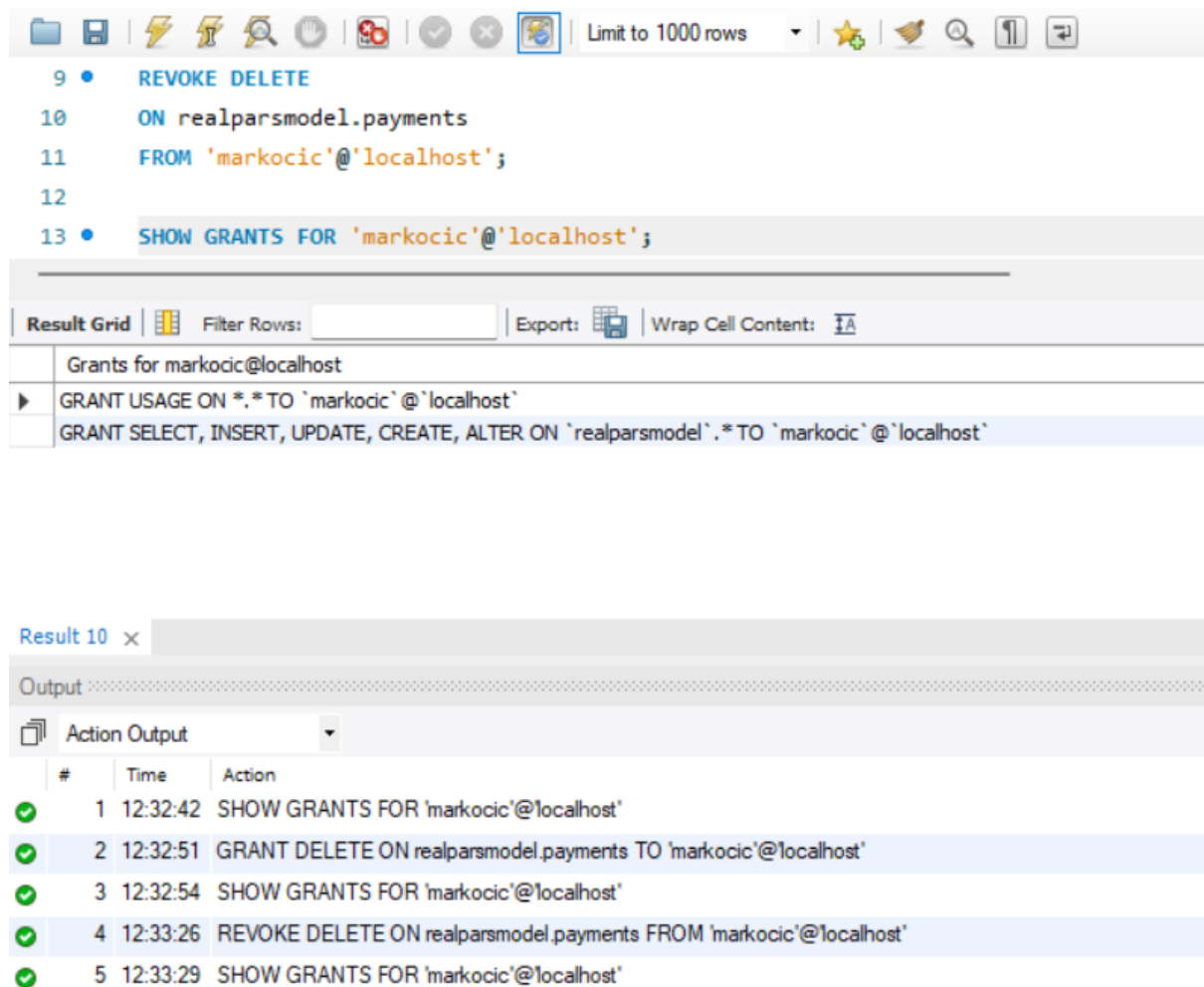


Slika 3.4. Izvršenje *REVOKE* naredbe – ukidanje određenih privilegija

Granični slučaj koji ovde postoji je nemogućnost ukidanja privilegija nad pojedinačnim tabelama baze podataka ukoliko kao takve ranije nisu dodate. Primera radi, privilegija *DELETE* je dodeljena nad celom bazom podataka, a ne nad pojedinačnom tabelom, te tako se ne može ukinuti sa pojedinačne tabele, već isključivo sa cele baze podataka. Ovo se objašnjava činjenicom da treba izbrisati torku iz tabele *grant*-ova koja ni ne postoji. Ukoliko želimo da ukinemo ovu privilegiju nad pojedinačnom tabelom, najpre ju je treba dodati, pa tek nakon toga oduzeti, što je prikazano na slikama ispod.



Slika 3.5. Dodela privilegije nad pojedinačnom tabelom

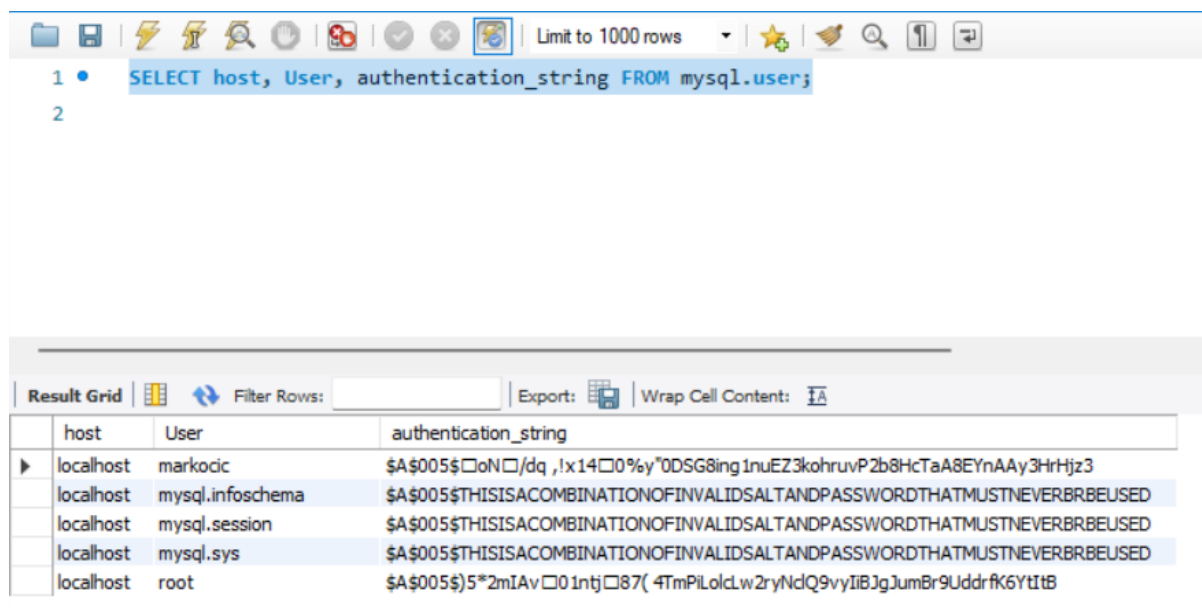


Slika 3.6. Ukidanje privilegije nad pojedinačnom tabelom

3.1. Bezbednost lozinki

Što se tiče lozinki, najsigurniji metod za krajnjeg korisnika je traženje iste od strane klijentske aplikacije. Prilikom unosa lozinke, savet je da se na ekranu ne pojavljuju karakteri koje unosi korisnik, već samo *. Umesto *, ne moramo prikazivati išta i na taj način se ne može zaključiti ni dužina lozinke. Lozinka se može čuvati i u opcionalnoj datoteci kojoj može pristupiti samo korisnik koji ju je stvorio. Ovaj način je takođe bezbedan.

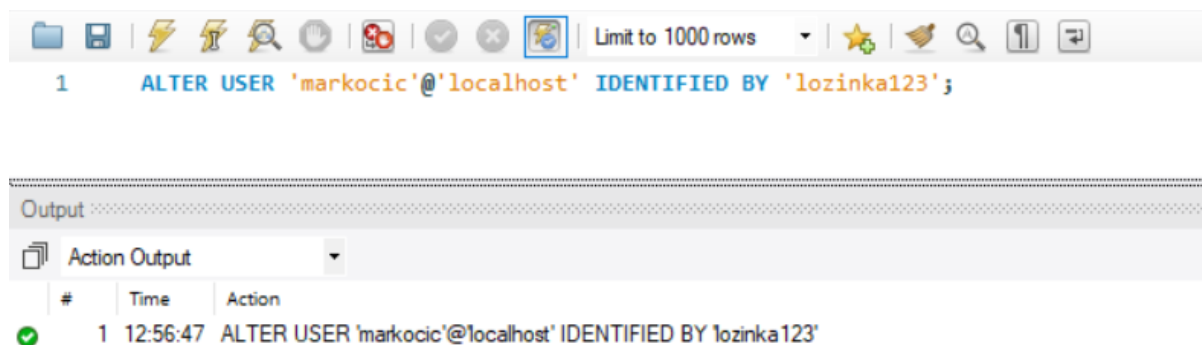
MySQL čuva lozinke za sve korisnike u tabeli *mysql.user* i to ne kao *plaintext*, već kao *hash* vrednost *hash* vrednosti same lozinke i *salt*-a (kolona *authentication_string*). Može se podestiti da lozinke ističu posle određenog vremena, tako da ih korisnici moraju obnavljati. Inače, pristup ovoj tabeli nikada ne treba biti dodeljen bilo kom korisniku koji nije administrator. Na slici ispod su prikazane neke od kolona prethodno pomenute tabele. Naredba je izvršena od strane *root*-a jer korisnik *markocic* nema privilegije.



host	User	authentication_string
localhost	markocic	\$A\$005\$oN□/dq ,!x14□0%y"0DSG8ing1nuEZ3kohruvP2b8HcTaA8EYnAAy3HrHjz3
localhost	mysql.infoschema	\$A\$005\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
localhost	mysql.session	\$A\$005\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
localhost	mysql.sys	\$A\$005\$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
localhost	root	\$A\$005\$5*2mIAv□01ntj□87(4TmPiLolclw2ryNdQ9vyIIBJgJumBr9UddrfK6YtItB

Slika 3.7. Korisnici i prikaz njihovih uskladištenih lozinki

Primer promene lozinke je dat na slici ispod. Ova naredba može biti izvršena od strane privilegovanih korisnika (*root* korisnik u našem slučaju), ali i od korisnika čije se ima javlja u samoj naredbi (*markocic*). S obzirom da je ova lozinka veoma slaba, odmah je zamenjena jačom koju nećemo prikazati u radu.



```
1 ALTER USER 'markocic'@'localhost' IDENTIFIED BY 'lozinka123';
```

#	Time	Action
1	12:56:47	ALTER USER 'markocic'@'localhost' IDENTIFIED BY 'lozinka123'

Slika 3.8. Promena lozinke korisnika *markocic*

Ukoliko server nije pokrenut korišćenjem opcije *--log-raw* (koja se ne preporučuje u produkciji), određene naredbe koje sadrže lozinku kao čitljiv, običan tekst, biće *log*-ovane sa modifikacijom (recimo u datoteci *the general query log*) i sam tekst će postati šifrovan. Neke od naredbi na koje se ovo odnosi su:

- *CREATE USER ... IDENTIFIED BY ...*
- *ALTER USER ... IDENTIFIED BY ...*
- *SET PASSWORD ...*

Naredbe koje počinju sa *INSERT* ili pak *UPDATE* beleže se u izvornom obliku, tako da ih treba izbegavati prilikom rada sa tabelom *mysql.user*. Ako se naredba ne može uspešno parsirati (primera radi postoji neka sintaksna greška), ista se ne beleži zbog bezbednosti jer sistem nije siguran da se u njoj ne nalazi lozinka. Ukoliko ipak želimo da se i nevalidne naredbe *log*-uju, pokrenuti server sa već pomenutom *--log-raw* opcijom i nakon toga se sve čuva kao običan tekst što je jako loše po pitanju sigurnosti.

3.2. Zaključavanje naloga

Informaciju o tome da li je određeni nalog otključan ili pak zaključan možemo dobiti čitanjem kolone *account_locked* iz sistemske tabele *mysql.user*. Ako imamo *flag Y*, nalog je zaključan, dok *flag N* ukazuje na otključan nalog. Inače, inicijalno (ukoliko se ništa ne navede) novokreirani nalog je otključan.

Ukoliko želimo da eksplicitno zaključamo neki nalog i time onemogućimo konekciju korisnika korišćenjem tog naloga, to možemo učiniti kao privilegovani korisnik – *root* korisnik izvršenjem sledeće naredbe:

ALTER USER određeni nalog ACCOUNT LOCK;

što je prikazano na slici ispod. Zatim, kako bismo se uverili da je zbilja nalog zaključan čitamo odgovarajuće kolone iz sistemske tabele *mysql.user* što se takođe može videti na slici ispod.

The screenshot displays a MySQL command-line interface. At the top, a toolbar includes icons for file operations and a 'Limit to 1000 rows' dropdown. The command history shows two executed queries:

```
1 • ALTER USER 'markocic'@'localhost' ACCOUNT LOCK;  
2  
3 • SELECT host, User, account_locked FROM mysql.user;  
4
```

Below the commands, the 'Result Grid' section shows the output of the second query. The grid has columns for 'host', 'User', and 'account locked'. The first row, where 'markocic' is locked, is highlighted with a red rectangle.

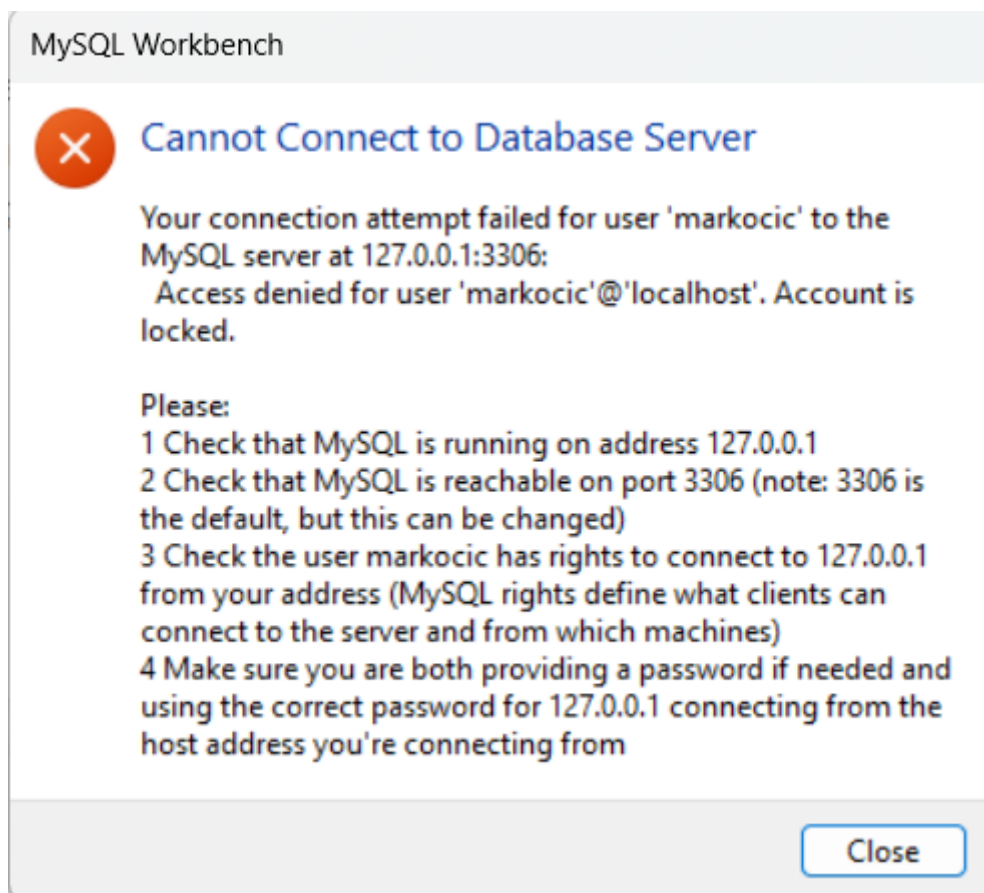
	host	User	account locked
▶	localhost	markocic	Y
	localhost	mysql.infoschema	Y
	localhost	mysql.session	Y
	localhost	mysql.sys	Y
	localhost	root	N

At the bottom, the 'Output' section shows the 'Action Output' log:

#	Time	Action
✓ 1	11:33:00	SELECT host, User, account_locked FROM mysql.user LIMIT 0, 1000
✓ 2	11:34:32	ALTER USER 'markocic'@'localhost' ACCOUNT LOCK
✓ 3	11:34:37	SELECT host, User, account_locked FROM mysql.user LIMIT 0, 1000

Slika 3.9. Eksplicitno zaključavanje određenog naloga

Ono što mi se ne sviđa po pitanju bezbednosti kod *MySQL*-a je to što nakon eksplicitnog zaključavanja naloga, već postojeće konekcije nisu podrazumevano prekinute, već korisnik nesmetano može raditi sa bazom podataka kao da ničeg nije ni bilo. Tek nakon pokušaja kreiranja nove konekcije, ova naredba ima efekta što se da videti na slici ispod.

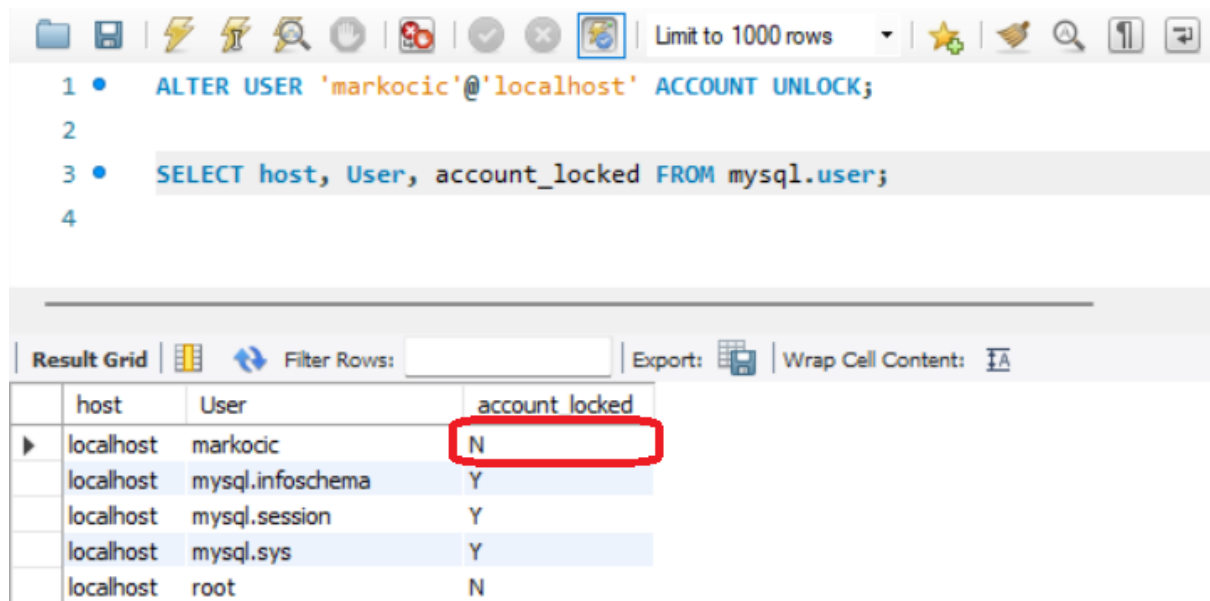


Slika 3.10. Nemogućnost povezivanja na server korišćenjem zaključanog naloga

Otključavanje i ujedno omogućavanje ponovnog povezivanja na server korišćenjem datog naloga, vrši se sledećom naredbom:

ALTER USER određeni nalog **ACCOUNT UNLOCK**;

što je prikazano na slici ispod. Treba napomenuti da ovu naredbu vrši *root* korisnik (korisnik koji je i napravio ovaj nalog). Vrednost kolone *account_locked* za korisnika *markocic* je sada *N* što ukazuje na činjenicu da nalog nije više zaključan i da se dati korisnik može ponovo konektovati na server.



Slika 3.11. Otključavanje naloga od strane *root* korisnika

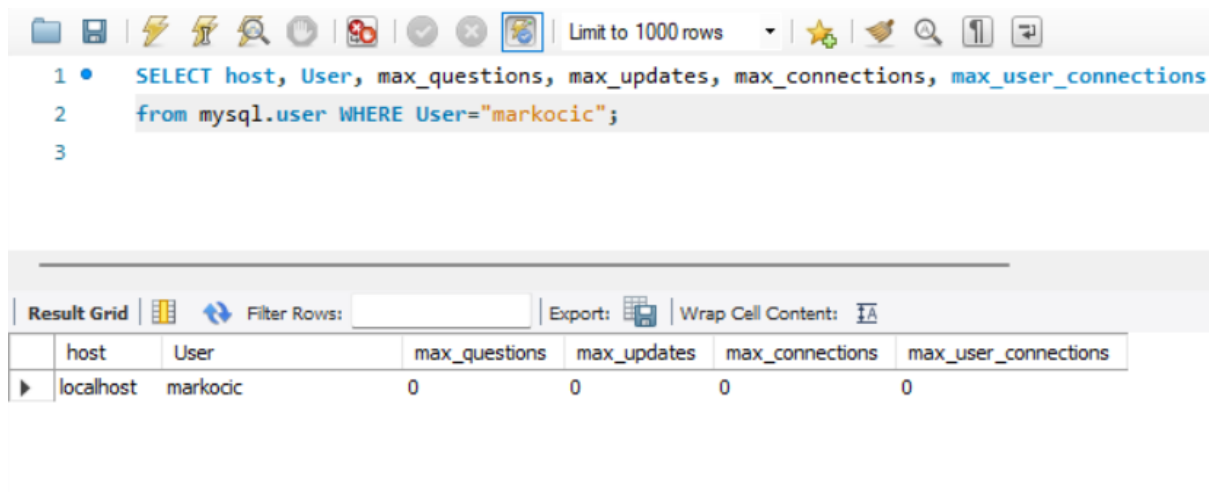
3.3. Ograničenje resursa naloga

MySQL omogućava postavljanje ograničenja korišćenja resursa servera po individualnom nalogu i to:

- broj upita po satu koji se mogu izdati;
- broj ažuriranja po satu koja se mogu izvršiti;
- broj konektovanja naloga na server po satu;
- broj istovremenih konekcija sa serverom.

Za limitiranje resursa koristi se *WITH* klauzula, bilo prilikom kreiranja ili pak modifikovanja korisnika. Podrazumevana vrednost za bilo koja ograničenje je 0 – nema ograničenja. Ograničenja se čuvaju u tabeli *mysql.user*. Ograničenja postavljamo koristeći privilegovani *root* nalog (nalog koji je kreirao nalog *markocic*).

Na slici ispod su prikazana ograničenja za nalog *markocic* (*max_questions* – broj upita po satu, *max_updates* – broj ažuriranja po satu, *max_connections* – broj konektovanja naloga na server po satu, *max_user_connections* – broj istovremenih konekcija). Sa slike vidimo da inicijalno nema nikakvih ograničenja.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

1 • SELECT host, User, max_questions, max_updates, max_connections, max_user_connections
2   from mysql.user WHERE User="markocic";
3

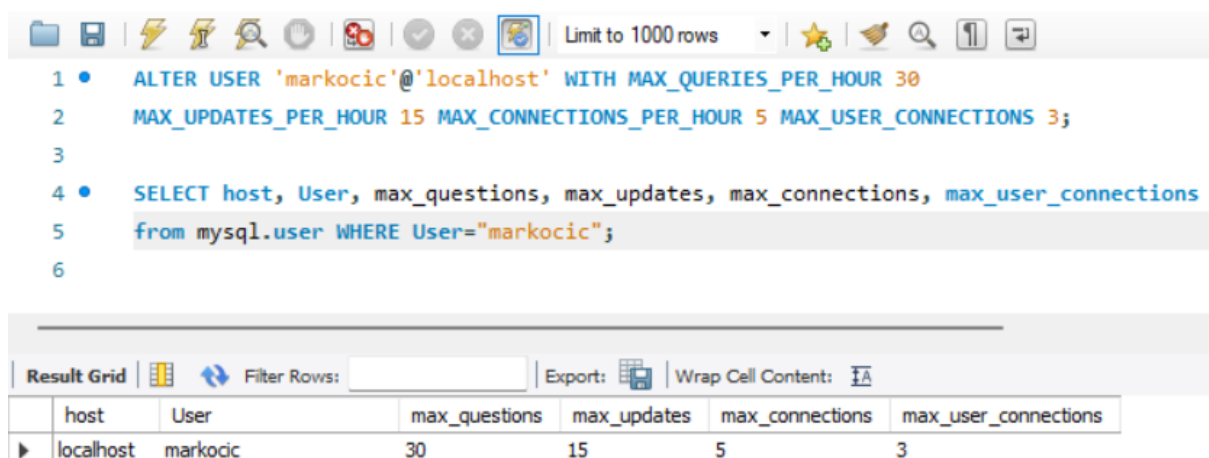
```

The Result Grid below shows the query results:

host	User	max_questions	max_updates	max_connections	max_user_connections
localhost	markocic	0	0	0	0

Slika 3.12. Inicijalna ograničenja za nalog *markocic*

Sada ćemo postaviti ograničenja za nalog *markocic* i to: 30 upita po satu, 15 ažuriranja po satu, 5 povezivanja na server po satu, kao i 3 istovremene konekcije sa naloga *markocic*. Rezultat izvršenja je prikazan na slici ispod.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```

1 • ALTER USER 'markocic'@'localhost' WITH MAX_QUERIES_PER_HOUR 30
2   MAX_UPDATES_PER_HOUR 15 MAX_CONNECTIONS_PER_HOUR 5 MAX_USER_CONNECTIONS 3;
3
4 • SELECT host, User, max_questions, max_updates, max_connections, max_user_connections
5   from mysql.user WHERE User="markocic";
6

```

The Result Grid below shows the query results:

host	User	max_questions	max_updates	max_connections	max_user_connections
localhost	markocic	30	15	5	3

Slika 3.13. Uspešno postavljena ograničenja za nalog *markocic*

Ukoliko želimo da ukinemo ranije postavljeno ograničenje, to se postiže postavljanjem vrednosti kolone na 0. Primer izvršenja naredbe dat je na slici ispod.

The screenshot shows a MySQL command-line interface with two SQL queries executed. The first query is `ALTER USER 'markocic'@'localhost' WITH MAX_QUERIES_PER_HOUR 0;` and the second is `SELECT host, User, max_questions, max_updates, max_connections, max_user_connections from mysql.user WHERE User="markocic";`. The result grid below shows the user 'markocic' at 'localhost' with `max_questions` set to 0, `max_updates` to 15, `max_connections` to 5, and `max_user_connections` to 3. The `max_questions` value of 0 is highlighted with a red rectangle.

```
1 • ALTER USER 'markocic'@'localhost' WITH MAX_QUERIES_PER_HOUR 0;
2
3 • SELECT host, User, max_questions, max_updates, max_connections, max_user_connections
4   from mysql.user WHERE User="markocic";
5
```

host	User	max_questions	max_updates	max_connections	max_user_connections
localhost	markocic	0	15	5	3

Slika 3.14. Ukidanje ograničenja broja upita po satu – sada je neograničen

4. SISTEMSKE PROMENLJIVE I OPCIJE

U nastavku je dat kratak opis pojedinih sistemskih promenljivih i opcija:

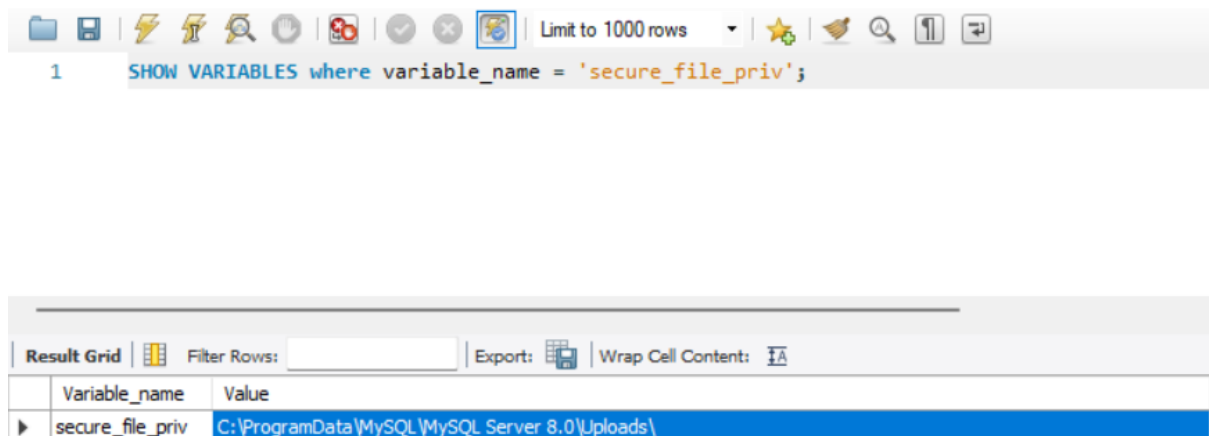
- *--allow-suspicious-udfs* – opcija koja kontroliše da li se učitavaju funkcije koje imaju samo xxx simbol za glavnu funkciju (podrazumevana vrednost je *OFF*);
- *automatic_sp_privileges* – sistemska promenljiva čija je podrazumevana vrednost *ON* i tada server automatski dodeljuje *EXECUTE* i *ALTER ROUTINE* privilegije korisniku koji je kreirao rutinu;
- *--chroot* – sistemskim pozivom *chroot()* postavljamo server u zatvoreno okruženje prilikom pokretanja (što se preporučuje);
- *local_infile* – promenljiva koja kontroliše *LOCAL* mogućnost za *LOAD DATA* naredbu na serverskoj strani;
- *--safe-user-create* – ukoliko je ova opcija omogućena, korisnik ne može kreirati nove korisnike koristeći naredbu *GRANT* osim ako nema privilegiju *INSERT* za sistemsku tabelu *mysql.user* (podrazumevana vrednost je *OFF*);
- *secure_file_priv* – promenljiva se koristi za ograničavanje efekata operacija uvoza i izvoza podataka, primera radi naredbe *LOAD DATA* i *SELECT ... INTO OUTFILE*, kao i funkcija *LOAD_FILE()* i ove operacije su dozvoljene samo korisnicima koji imaju privilegiju *FILE*. Ukoliko nema vrednost, promenljiva nema efekta, ukoliko joj je vrednost *NULL*, operacije uvoza i izvoza su onemogućene, dok ukoliko je vrednost naziv direktorijuma, server može raditi samo sa datotekama iz tog foldera;
- *skip-grant-tables* – uzrokuje da server ne čita tabele dodeljenih privilegija što znači da svaki korisnik ima neograničeni pristup svim bazama podataka;
- *skip_name_resolve* – ukoliko postavimo ovu sistemsku promenljivu na *ON*, *MySQL* će koristiti samo *IP* adrese, a ne imena *host*-ova, tako da se i u tabelama dodeljenih privilegija moraju naći takođe *IP* adrese;
- *skip_networking* – sistemska promenljiva se koristi za dozvoljavanje *TCP/IP* konekcija. S obzirom da je podrazumevano *OFF*, *TCP/IP* konekcija je dozvoljena, ali ukoliko želimo konekciju samo sa lokalnim klijentima, preporučuje se omogućavanje ove promenljive;
- *skip_show_database* – sistemska promenljiva koja kontroliše kome je dozvoljeno da koristi naredbu *SHOW DATABASES*.

Na slici ispod je prikazana tabela u kojoj se nalaze sistemske promenljive i opcije koje se odnose na bezbednost, kao i njihove karakteristike – da li se mogu koristiti u *command line interface*-u, u opcionalnoj datoteci, da li je globalni opseg u pitanju,...

Name	Cmd-Line	Option File	System Var	Status Var	Var Scope	Dynamic
allow-suspicious-udfs	Yes	Yes				
automatic_sp_privileges	Yes	Yes	Yes		Global	Yes
chroot	Yes	Yes				
local_infile	Yes	Yes	Yes		Global	Yes
safe-user-create	Yes	Yes				
secure_file_priv	Yes	Yes	Yes		Global	No
skip-grant-tables	Yes	Yes				
skip_name_resolve	Yes	Yes	Yes		Global	No
skip_networking	Yes	Yes	Yes		Global	No
skip_show_database	Yes	Yes	Yes		Global	No

Slika 4.1. Tabela karakteristika sistemskih promenljivih i opcija koje se odnose na bezbednost

Na slici ispod je prikazana trenutna vrednost promenljive *secure_file_priv*. Vrednost je *C:\ProgramData\MySQL\MySQL Server 8.0\Uploads* što znači da se uvoz i izvoz datoteka može vršiti isključivo u okviru direktorijuma *Uploads* koji se nalazi na zadatoj putanji.



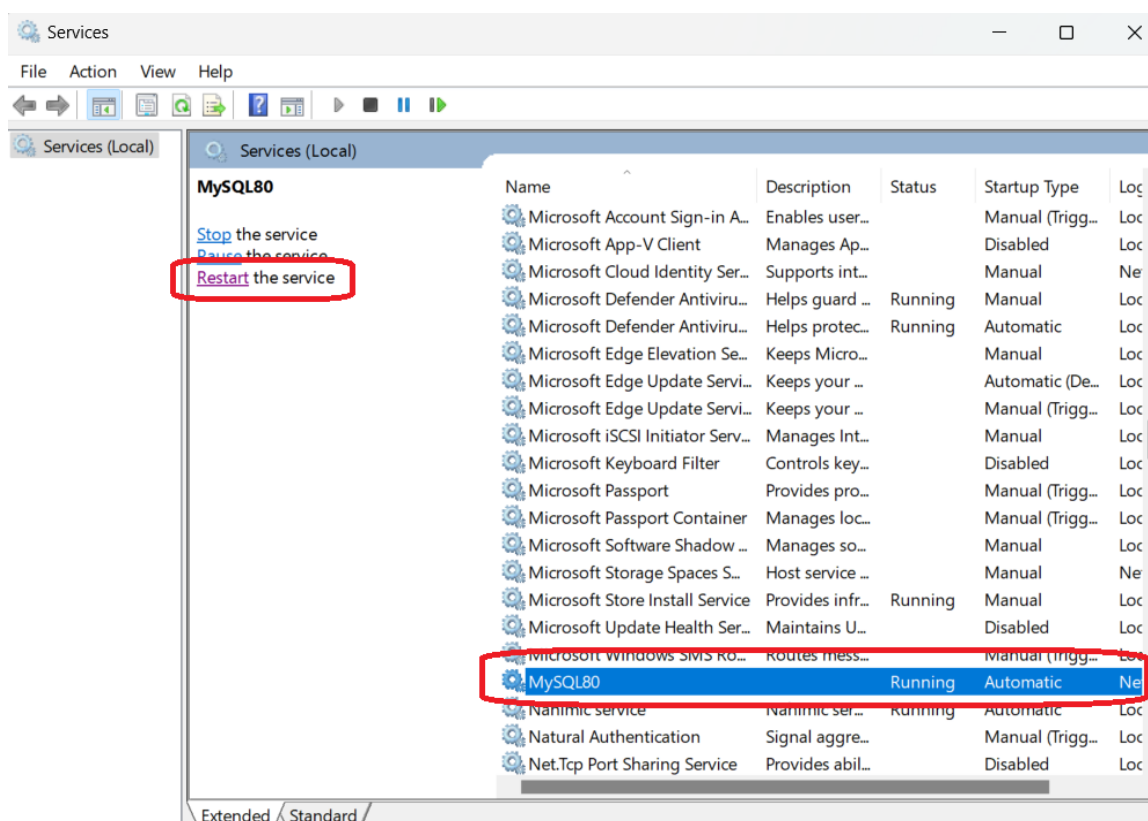
Slika 4.2. Vrednost promenljive *secure_file_priv* korišćenjem naredbe *SHOW VARIABLES*

Ukoliko želimo da izmenimo vrednost ove promenljive, to ne možemo uraditi direktno korišćenjem *SQL*-a, već je neophodno načiniti određene promene u konfiguracionoj datoteci. Ovaj *file* servera koji je pokrenut na *Windows* mašini nalazi se na lokaciji *C:\ProgramData\MySQL\MySQL Server 8.0\my.ini*. Na slici ispod je prikazan deo ove datoteke.

```
C:\ProgramData\MySQL\MySQL Server 8.0\my.ini - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
my.ini
151 #
152 # TABLE or CREATE DATABASE statement. Name comparisons are case-sensitive. You should not
153 # set this variable to 0 if you are running MySQL on a system that has case-insensitive file
154 # names (such as Windows or macOS). If you force this variable to 0 with
155 # --lower-case-table-names=0 on a case-insensitive file system and access MySQL table names
156 # using different lettercases, index corruption may result.
157 # Value 1 = Table names are stored in lowercase on disk and name comparisons are not case-sensitive.
158 # MySQL converts all table names to lowercase on storage and lookup. This behavior also applies
159 # to database names and table aliases.
160 # Value 2 = Table and database names are stored on disk using the lettercase specified in the CREATE TABLE
161 # or CREATE DATABASE statement, but MySQL converts them to lowercase on lookup. Name comparisons
162 # are not case-sensitive. This works only on file systems that are not case-sensitive! InnoDB
163 # table names and view names are stored in lowercase, as for lower_case_table_names=1.
164 lower_case_table_names=1
165 #
166 # This variable is used to limit the effect of data import and export operations, such as
167 # those performed by the LOAD DATA and SELECT ... INTO OUTFILE statements and the
168 # LOAD FILE() function. These operations are permitted only to users who have the FILE privilege.
169 secure-file-priv="C:/ProgramData/MySQL/"
170 #
171 # The maximum amount of concurrent sessions the MySQL server will
172 # allow. One of these connections will be reserved for a user with
173 # SUPER privileges to allow the administrator to login even if the
174 # connection limit has been reached.
175 max_connections=151
176 #
177 # The number of open tables for all threads. Increasing this value increases the number
178 # of file descriptors that mysqld requires.
179 table_open_cache=4000
180 #
181 # Defines the maximum amount of memory that can be occupied by the TempTable
182 # storage engine before it starts storing data on disk.
183 temptable_max_ram=1G
184 #
185 # Defines the maximum size of internal in-memory temporary tables created
186 # by the MEMORY storage engine and, as of MySQL 8.0.28, the TempTable storage
187 # engine. If an internal in-memory temporary table exceeds this size, it is
188 # automatically converted to an on-disk internal temporary table.
189 tmp_table_size=768M
190 #
191 # The storage engine for in-memory internal temporary tables (see Section 8.4.4, "Internal
192 # Temporary Table Use in MySQL"). Permitted values are TempTable (the default) and MEMORY.
193 internal_tmp_mem_storage_engine=TempTable
194 *** MySQL Specific options ***
MS ini file length: 15,816 lines: 346 Ln: 168 Col: 40 Pos: 6,817 Windows (CR LF) UTF-8 INS
```

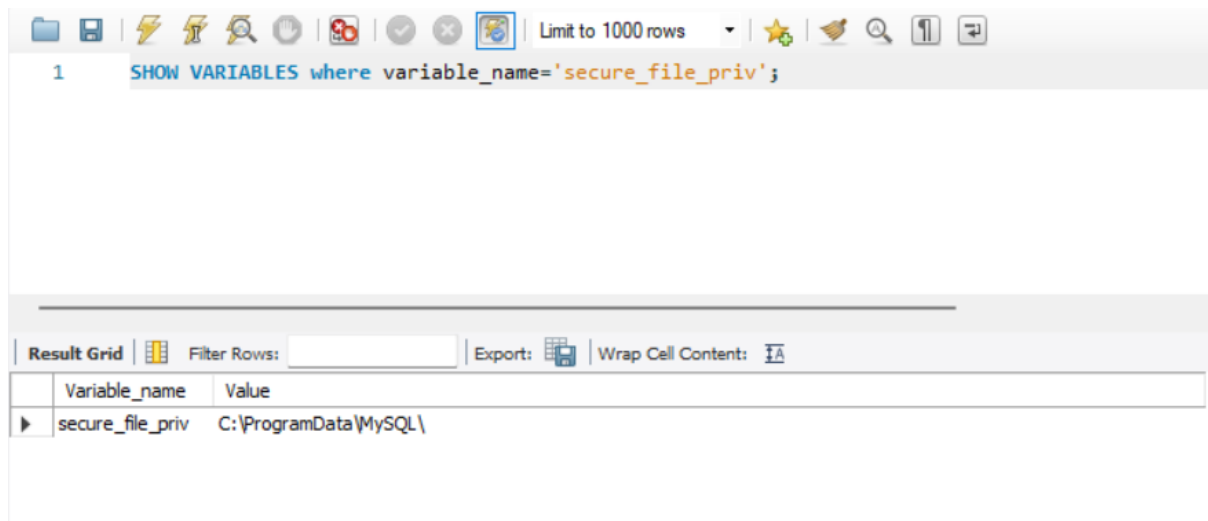
Slika 4.3. Deo konfiguracione datoteke *my.ini*

Nakon pronalaska promenljive, postavljamo joj novu željenu vrednost i čuvamo datoteku (sa privilegijama administratora). Potrebno je ponovo pokrenuti server kako bi konfiguracioni *file* bio iznova učitani. U aplikaciji *Services* nalazimo *MySQL* server i pritisnemo *Restart the service*.



Slika 4.4. Restart-ovanje *MySQL* servera

Na slici ispod je prikazana novopostavljena vrednost promenljive *secure_file_priv*. Vrednost je sada *C:\ProgramData\MySQL*.



Slika 4.5. Nova vrednost *secure_file_priv* je uspešno postavljena

4.1. Bezbednosna razmatranja za *LOAD DATA LOCAL*

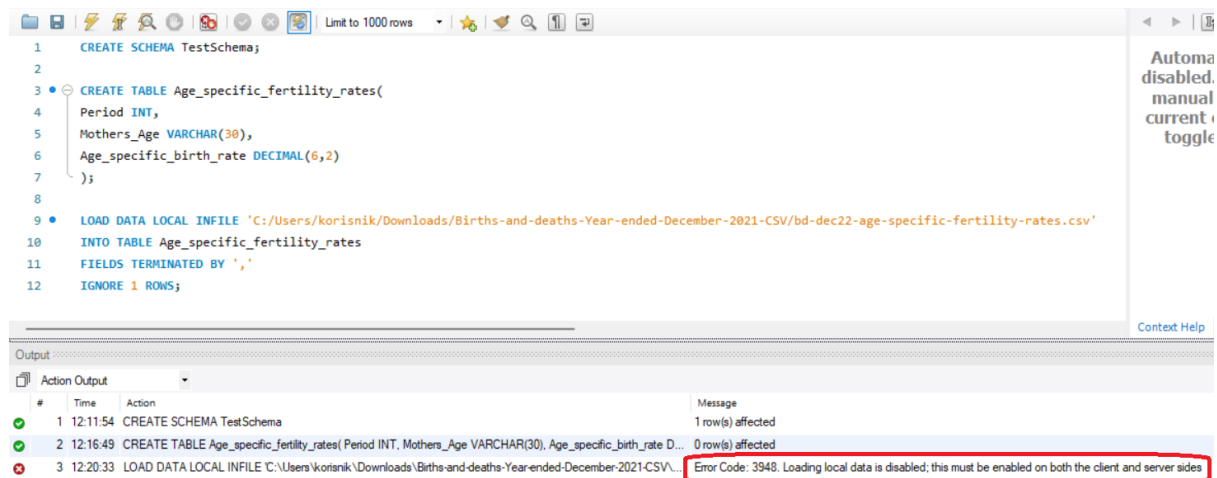
LOAD DATA naredba učitava datoteku u tabelu. Naredba *LOAD DATA LOCAL* učitava datoteku koja se nalazi na klijentskoj mašini i šalje je serveru. S obzirom da je *LOAD DATA LOCAL SQL* naredba, parsiranje iste se izvodi na strani servera i sam prenos datoteke je iniciran od strane servera. Maliciozan server može zahtevati datoteku koja se razlikuje od one koju je klijent specificirao u naredbi. Ukoliko klijent ima privilegije čitanja traženog *file-a*, isti će biti poslat serveru. Naime, ne mora se ni poslati tačno ova naredba, već bilo koja i server može zahtevati datoteke po želji i zato je generalni savet ne povezivati se sa serverima kojima ne verujemo. U cilju izbegavanja povezivanja sa nepouzdanim serverima glavnu ulogu igraju sertifikati. Klijent uspostavlja konekciju korišćenjem opcije *--ssl-mode=VERIFY_IDENTITY* pa odgovarajući *CA* sertifikat.

Na serverskoj strani, sistemska promenljiva *local_infile* je podrazumevano onemogućena – server odbija učitavanje lokalnih podataka od strane klijenata. Vrednost joj se može menjati u fazi izvršenja. Na klijentskoj strani, odbijanje ili dozvoljavanje učitavanja lokalnih datoteka zavisi od opcije *--local-infile* koja može biti postavljena na 0 ili 1 respektivno i podrazumevano je takođe onemogućena.

Ukoliko želimo da učitamo datoteku koja se nalazi lokalno, potrebno je da to bude omogućeno i na serverskoj i na klijentskoj strani.

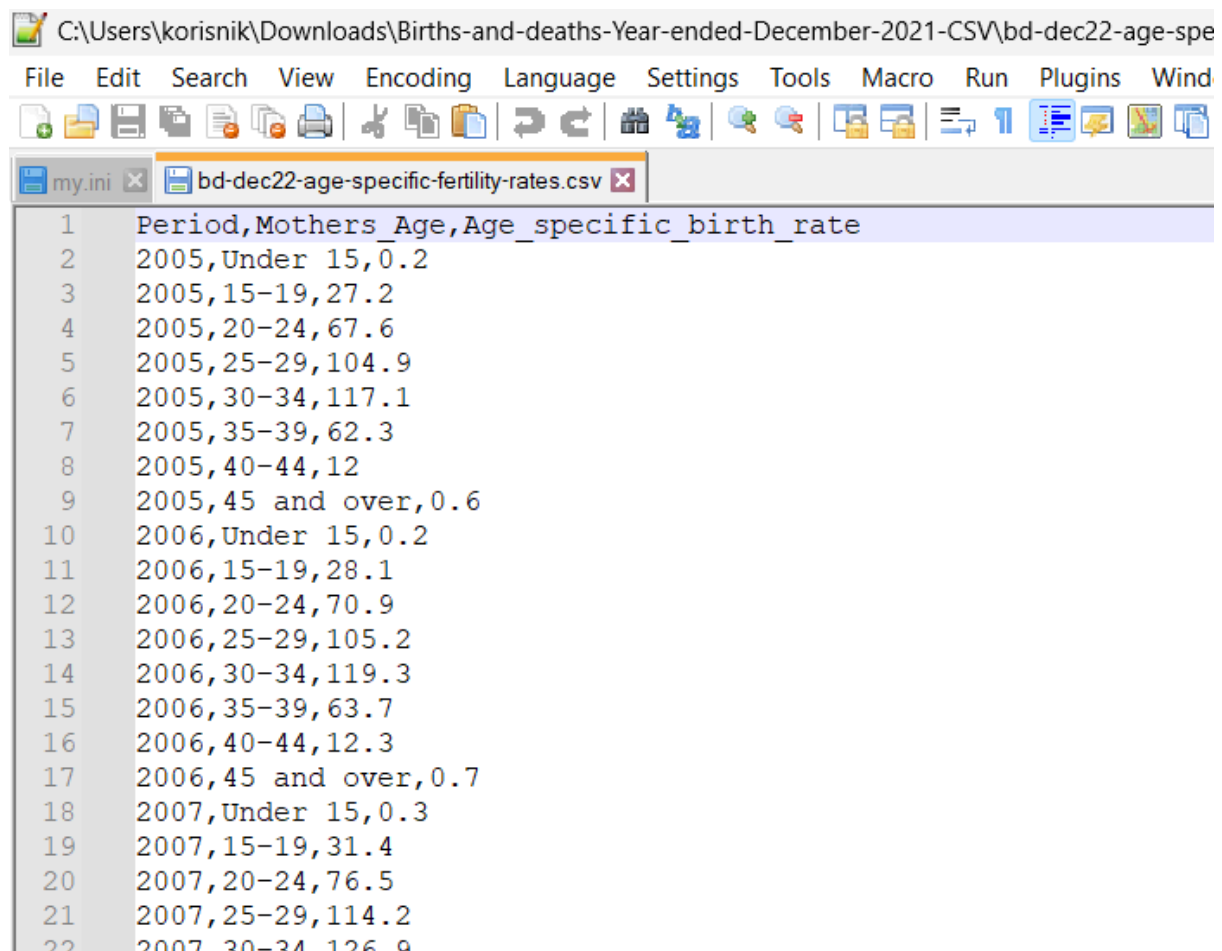
Na slici ispod je prikazano kreiranje nove sheme *TestSchema* i kreiranje nove tabele *Age_specific_fertility_rates* u koju želimo da učitamo podatke iz *.csv* datoteke. Nakon

pokušaja lokalnog učitavanja datoteke, dobili smo grešku da je učitavanje nemoguće i da mora biti dozvoljeno kako na klijentskoj, tako i na serverskoj strani.



Slika 4.6. Greška prilikom pokušaja lokalnog učitavanja podataka

Deo sadržaja datoteke koju želimo da učitamo je prikazan na slici ispod.



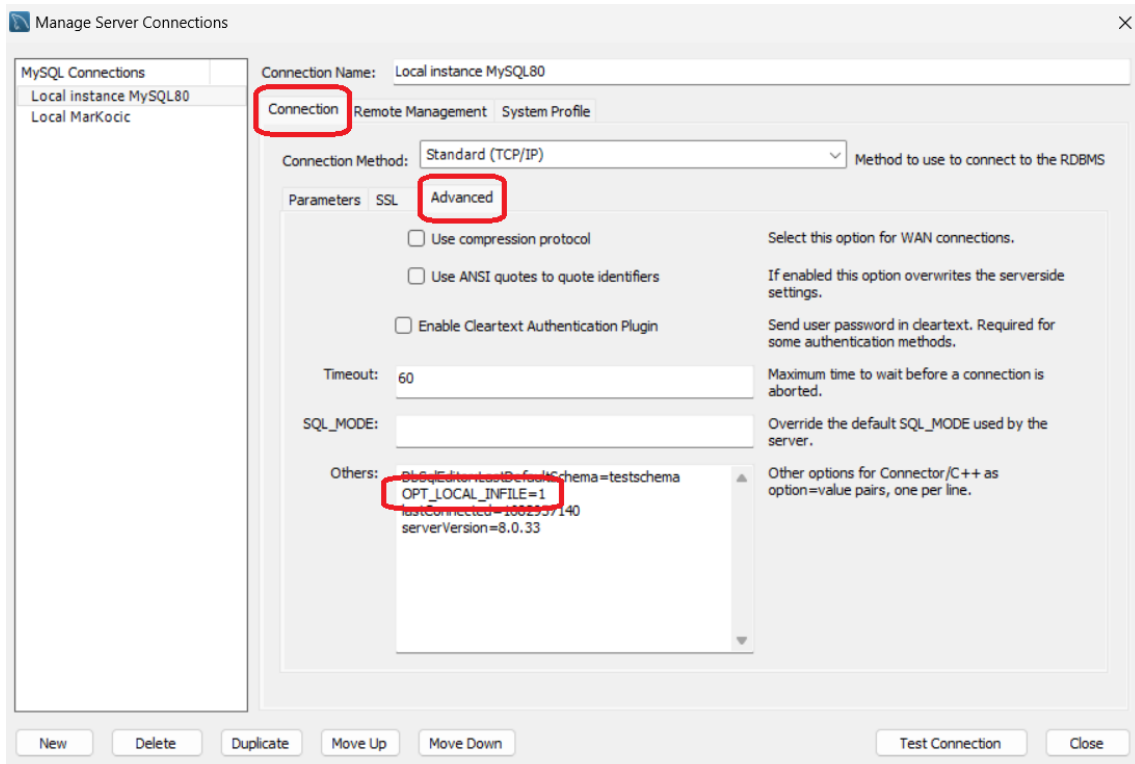
Slika 4.7. Csv datoteka koju želimo da učitamo u novokreiranu tabelu

Neophodno je postaviti *local_infile* sistemsku promenljivu na 1 (ON) i time smo omogućili rad sa lokalnim datotekama klijenata na serverskoj strani. Na slici ispod je prikazano uspešno postavljanje ove promenljive od strane *root* korisnika.



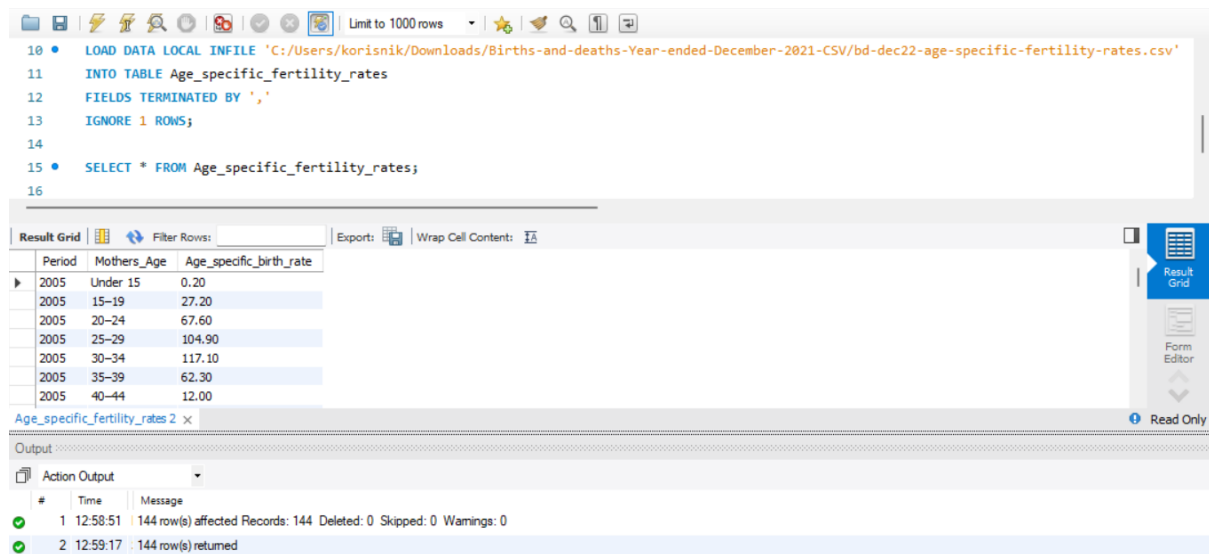
Slika 4.8. Provera i postavljanje sistemske promenljive *local_infile*

Jedan od načina postavljanja opcije učitavanja lokalnih datoteka na klijentskoj strani je dodavanje linije *OPT_LOCAL_INFILE=1* u sekciji *Others* tab *Advanced* koji se odnosi na konekciju. Prozor prikazan na slici ispod dobija se pritiskom desnog tastera miša na konekciju, a zatim izborom *Edit Connection...*



Slika 4.9. Omogućavanje lokalnog učitavanja na klijentskoj strani

Na kraju smo uspešno učitali podatke iz gore pomenute datoteke u tabelu što se da videti na slici ispod.



Slika 4.10. Uspešno učitavanje datoteke i prikaz tabele

Što se tiče bezbednosti, savetuje se specificiranje putanje na klijentskoj strani na kojoj se mogu naći datoteke koje mogu biti učitane. To se postiže postavljanjem opcije `--load-data-local-dir=dir_name` (`dir_name` predstavlja putanju) prilikom kreiranja konekcije. Bitno je napomenuti da `--local-infile` opcija treba biti postavljena na 0, tačnije onemogućena jer jedino u tom slučaju važi `--load-data-local-dir` opcija. Ukoliko je `--local-infile` opcija omogućena, specificirana putanja ne igra nikakvu ulogu i nismo dobili pojačanu sigurnost.

Na slici ispod je prikazano kreiranje nove konekcije sa pojedinim opcijama. Određeno je sa koje lokacije je moguće učitavati datoteke – `C:/Users/korisnik/Desktop/proba`. Novokreirana tabela je prazna što se da videti (*Empty set*). Najpre pokušavamo da učitamo datoteku sa *Desktop*-a koja se ne nalazi u direktorijumu *proba* i dobijamo grešku o postojanju restrikcija prilikom izvršenja `LOAD DATA LOCAL INFILE` naredbe. Nakon toga uspešno učitavamo datoteku koja se nalazi u direktorijumu *proba*. Uspešno učitani podaci se takođe mogu videti na slici.

```

C:\Users\korisnik>mysql -u root -p --local-infile=0 --load-data-local-dir=C:/Users/korisnik/Desktop/proba
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from testschema.age_specific_fertility_rates;
Empty set (0.00 sec)

mysql> LOAD DATA LOCAL INFILE 'C:/Users/korisnik/Desktop/bd-dec22-age-specific-fertility-rates.csv'
-> INTO TABLE testschema.age_specific_fertility_rates
-> FIELDS TERMINATED BY ','
-> IGNORE 1 ROWS;
ERROR 2068 (HY000): LOAD DATA LOCAL INFILE file request rejected due to restrictions on access.
mysql> LOAD DATA LOCAL INFILE 'C:/Users/korisnik/Desktop/proba/bd-dec22-age-specific-fertility-rates.csv'
-> INTO TABLE testschema.age_specific_fertility_rates
-> FIELDS TERMINATED BY ','
-> IGNORE 1 ROWS;
Query OK, 144 rows affected (0.00 sec)
Records: 144  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from testschema.age_specific_fertility_rates;
+-----+-----+-----+
| Period | Mothers_Age | Age_specific_birth_rate |
+-----+-----+-----+
| 2005 | Under 15 | 0.20 |
| 2005 | 15?19 | 27.20 |
| 2005 | 20?24 | 67.60 |
| 2005 | 25?29 | 104.90 |
| 2005 | 30?34 | 117.10 |
| 2005 | 35?39 | 62.30 |

```

Slika 4.11. Rad sa *--load-data-local-dir* opcijom

U nastavku prikazujemo posledice postavljanja *--local-infile* opcije na 1. Bez obzira na postavljanje putanje sa koje bi se jedino mogle učitati datoteke, ipak je moguće učitavanje istih sa bilo koje lokacije na klijentskoj mašini jer *--local-infile* opciju nismo postavili na 0, već na 1 i time praktično „pregazili” *--load-data-local-dir* opciju. Uspešno učitavanje datoteke koja se ne nalazi na putanji koja je navedena prilikom kreiranja konekcije, odnosno ovaj bezbednosni propust je prikazan na slici ispod.

```

C:\Users\korisnik>mysql -u root -p --local-infile=1 --load-data-local-dir=C:/Users/korisnik/Desktop/proba
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from testschema.age_specific_fertility_rates;
Empty set (0.00 sec)

mysql> LOAD DATA LOCAL INFILE 'C:/Users/korisnik/Desktop/bd-dec22-age-specific-fertility-rates.csv'
-> INTO TABLE testschema.age_specific_fertility_rates
-> FIELDS TERMINATED BY ','
-> IGNORE 1 ROWS;
Query OK, 144 rows affected (0.01 sec)
Records: 144  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from testschema.age_specific_fertility_rates;
+-----+-----+-----+
| Period | Mothers_Age | Age_specific_birth_rate |
+-----+-----+-----+
| 2005 | Under 15 | 0.20 |
| 2005 | 15-19 | 27.20 |
| 2005 | 20-24 | 67.60 |
| 2005 | 25-29 | 104.90 |
| 2005 | 30-34 | 117.10 |
| 2005 | 35-39 | 62.30 |

```

Slika 4.12. Uspešno učitavanje datoteke koja se ne nalazi na prethodno definisanoj putanji

5. ZAKLJUČAK

Na kraju seminarskog rada ostalo je izvući pojedine zaključke iz istog. Neophodno je pratiti sve preporuke i smernice iz zvaničnih dokumentacija, biti u toku sa svim aktuelnim dešavanjima. Kao što je već rečeno, bezbednost je proces i sve vreme treba raditi na istoj. U radu je dato mnogo primera koji se odnose na bezbednost *MySQL* baze podataka.

LITERATURA

[1] *MySQL*, <https://dev.mysql.com/>, maj 2023.