 UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET

**REPLIKACIJA MYSQL BAZA PODATAKA KAO MEHANIZAM OPORAVKA I OBEZBEĐIVANJA DOSTUPNI PODATAKA**

Seminarski rad II

Studijski program: Računarstvo i informatika

Modul: Softversko inženjerstvo

Student: Mentor:

Bojan Mitić, br. indeksa: 1918 Prof. dr. Aleksandar Stanimirović

Niš, septembar 2025. Godina

SADRŽAJ

1. UVOD……………………………………………………………………………………………………………………………………………..3

1.1 Pojam replikacije baza podataka……………………………………………………………………………………………………………...3

1.2 Ciljevi replikacije……………………………………………………………………………………………………………………………..3

1.3 Značaj replikacije u informacionim sistemima……………………………………………………………………………………………….4

2. OSNOVNI POJMOVI REPLIKACIJE…………………………………………………………………………………………………………....4

2.1 Definicija replikacije………………………………………………………………………………………………………………………….5

2.2 Vrste replikacije……………………………………………………………………………………………………………………………….6

2.3 Prednosti i izazovi replikacije…………………………………………………………………………………………………………………8

3. REPLIKACIJA U MYSQL-U……………………………………………………………………………………………………………………...9

3.1 Arhitektura MySQL replikacije………………………………………………………………………………………………………………..9

3.2 Mehanizam rada replikacije u MySQL-u…………………………………………………………………………………………………….10

3.3 Tipovi replikacije u MySQL-u………………………………………………………………………………………………………………..12

4. REPLIKACIJA KAO MEHANIZAM OPORAVKA I DOSTUPNOSTI…………………………………………………………………………13

4.1 Failover i failback mehanizmi………………………………………………………………………………………………………………..13

4.2 Disaster recovery scenariji……………………………………………………………………………………………………………………14

4.3 Skalabilnost i balansiranje čitanja…………………………………………………………………………………………………………….14

5. PRAKTIČNI DEO- DEMONSTRACIJA REPLIKACIJE U MYSQL-U………………………………………………………………………..17

5.1 Priprema okruženja…………………………………………………………………………………………………………………………...17

5.2 Podešavanje master servera…………………………………………………………………………………………………………………..18

5.3 Podešavanje replica servera…………………………………………………………………………………………………………………..18

5.4 Testiranje replikacije………………………………………………………………………………………………………………………….19

5.5 Rezultati testiranja na SQL client-u…………………………………………………………………………………………………………..20

6. SIGURNOST I IZAZOVI REPLIKACIJE………………………………………………………………………………………………………..21

6.1 SSL/TLS enkripcija komunikacije…………………………………………………………………………………………………………....21

6.2 Izazovi replikacije…………………………………………………………………………………………………………………………….22

6.3 Monitoring replikacije………………………………………………………………………………………………………………………..22

6.4 Kompletan proces konfiguracije i testiranja MySQL replikacije…………………………………………………………………………….24

7. ZAKLJUČAK..........................................................................................................................................................................................................26

7.1 Prednosti replikacije u MySQL-u.....................................................................................................................................................................26

7.2 Ograničenja i izazovi........................................................................................................................................................................................28

7.3 Dalji pravci unapređenja...................................................................................................................................................................................28

8. REFERENCE...........................................................................................................................................................................................................30

1. **UVOD**

U digitalnom dobu, gde se informacije obrađuju i razmenjuju u realnom vremenu, dostupnost i pouzdanost podataka postaju ključni faktori za stabilnost i uspešnost informacionih sistema. Baze podataka predstavljaju temelj većine aplikacija, servisa i poslovnih procesa, te svaki prekid u radu ili gubitak podataka može dovesti do ozbiljnih posledica — od tehničkih zastoja do finansijskih gubitaka i narušavanja poverenja korisnika.

Kako bi se obezbedila neprekidna dostupnost podataka i otpornost sistema na greške, razvijeni su različiti mehanizmi za zaštitu i oporavak baza podataka. Među njima, replikacija zauzima posebno mesto kao efikasan i fleksibilan pristup koji omogućava sinhronizaciju podataka između više instanci baze. U kontekstu MySQL-a, replikacija se koristi kako u malim, tako i u velikim sistemima, zahvaljujući jednostavnoj konfiguraciji i mogućnosti prilagođavanja različitim arhitekturama.

* 1. **Pojam replikacije baza podataka**

Replikacija baze podataka podrazumeva proces automatskog kopiranja i sinhronizacije podataka sa jedne baze (najčešće glavne – master) na jednu ili više drugih baza (sekundarnih – slave). Ove replike mogu služiti kao rezervne kopije, kao aktivne baze za čitanje, ili čak kao baze koje preuzimaju rad u slučaju pada glavne instance. U MySQL sistemima, najčešće se koristi **master-slave** model, gde se sve promene sa glavne baze propagiraju ka replikama, ali postoje i naprednije konfiguracije poput **master-master** i **multi-source** replikacije.

Replikacija može biti sinhrona ili asinhrona, u zavisnosti od potreba sistema. Sinhrona replikacija obezbeđuje da se podaci istovremeno upisuju u sve instance, dok asinhrona omogućava veću fleksibilnost i brže izvršavanje operacija, uz mogućnost minimalnog kašnjenja u sinhronizaciji.

* 1. **Ciljevi replikacije**

Glavni ciljevi replikacije u MySQL bazama podataka obuhvataju:

* **Visoku dostupnost (High Availability)** Replikacija omogućava da sistem nastavi sa radom čak i u slučaju pada glavne baze. Sekundarna baza može preuzeti funkciju glavne, čime se izbegava prekid u radu aplikacije.
* **Oporavak podataka (Disaster Recovery)** U slučaju gubitka podataka, greške u aplikaciji ili nepredviđenih situacija, replikovane baze mogu poslužiti kao izvor za vraćanje izgubljenih informacija.
* **Balansiranje opterećenja (Load Balancing)** Čitanje podataka može se distribuirati na više instanci baze, čime se smanjuje opterećenje glavne baze i poboljšavaju performanse sistema, posebno u okruženjima sa velikim brojem korisnika.

Ovi ciljevi čine replikaciju ne samo tehničkim rešenjem, već i strateškim pristupom u dizajnu skalabilnih i pouzdanih informacionih sistema.

* 1. **Značaj replikacije u savremenim informacionim sistemima**

U savremenim IT okruženjima, gde se očekuje neprekidna dostupnost servisa, brz odziv i sigurnost podataka, replikacija postaje neizostavan deo arhitekture baze podataka. Ona omogućava organizacijama da izgrade sisteme otporne na greške, da obezbede kontinuitet poslovanja i da se pripreme za nepredviđene situacije.

Posebno u kontekstu MySQL-a, koji se koristi u velikom broju web aplikacija, e-commerce platformi i analitičkih sistema, replikacija pruža jednostavan i efikasan način da se obezbedi sigurnost i dostupnost podataka bez potrebe za kompleksnim infrastrukturnim rešenjima.

1. **ŠTA JE REPLIKACIJA BAZA PODATAKA**

Replikacija baza podataka predstavlja proces kopiranja i sinhronizacije podataka sa jedne baze (najčešće glavne – master) na jednu ili više drugih baza (sekundarnih – slave), koje se nazivaju replikama. Cilj replikacije je obezbeđivanje visoke dostupnosti, otpornosti na greške, balansiranja opterećenja i bržeg pristupa podacima u distribuiranim informacionim sistemima.

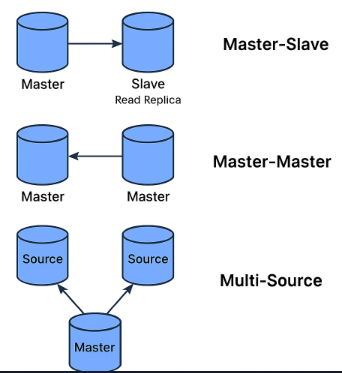
U modernim sistemima, replikacija omogućava da se podaci ne nalaze samo na jednom serveru, već da budu prisutni na više lokacija, čime se smanjuje rizik od gubitka informacija i povećava skalabilnost sistema.

* 1. **Definicija replikacije**

Replikacija baze podataka je proces u kojem se promene u jednoj instanci baze automatski prenose na druge instance, kako bi se obezbedila konzistentnost podataka. Ovaj proces može biti konfigurisan tako da se odvija u realnom vremenu, sa kašnjenjem, ili periodično, u zavisnosti od potreba sistema.

U MySQL okruženju, replikacija se najčešće koristi u arhitekturama kao što su:

* Master-Slave – jedna glavna baza šalje podatke replikama koje služe za čitanje.
* Master-Master – dve baze međusobno razmenjuju podatke i mogu istovremeno vršiti upis.
* Multi-Source – jedna baza prima podatke od više izvora.



Slika prikazuje tri najčešće arhitekture replikacije u MySQL-u:

* Master-Slave: jedan master šalje podatke ka jednom ili više slave servera (read-only).
* Master-Master: dve baze međusobno razmenjuju podatke i obe mogu pisati.
* Multi-Source: jedna baza prima podatke od više izvora (više master instanci).
  1. **Vrste replikacije**

**1. Sinhrona replikacija**

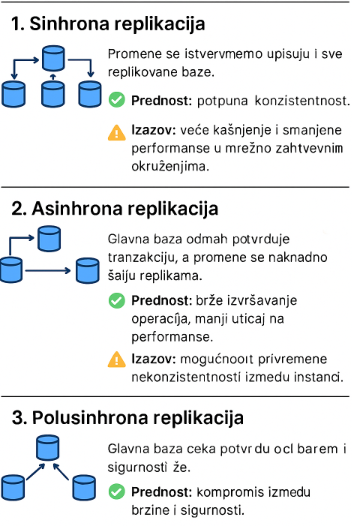
* Promene se istovremeno upisuju u sve replikovane baze.
* Glavna baza čeka potvrdu od svih replika pre nego što potvrdi transakciju.
* ✅ Prednost: potpuna konzistentnost.
* ⚠️ Izazov: veće kašnjenje i smanjene performanse u mrežno zahtevnim okruženjima.

**2. Asinhrona replikacija**

* Glavna baza odmah potvrđuje transakciju, a promene se naknadno šalju replikama.
* ✅ Prednost: brže izvršavanje operacija, manji uticaj na performanse.
* ⚠️ Izazov: mogućnost privremene nekonzistentnosti između instanci.

**3. Polusinhrona replikacija**

* Glavna baza čeka potvrdu od **barem jedne** replike pre nego što potvrdi transakciju.
* ✅ Prednost: kompromis između brzine i sigurnosti.
* ⚠️ Izazov: zahteva pažljivo balansiranje između dostupnosti i performansi.



Ova slika prikazuje tri vrste replikacije u MySQL-u:

* Sinhrona replikacija: svi replikovani serveri moraju potvrditi upis pre nego što se transakcija završi.
* Asinhrona replikacija: glavna baza odmah potvrđuje transakciju, a replike je primaju naknadno.
* Polusinhrona replikacija: potvrda se čeka od bar jedne replike, što predstavlja balans između brzine i sigurnosti.
  1. **Prednosti i izazovi replikacije**

**Prednosti:**

* Visoka dostupnost: sistem može nastaviti sa radom čak i ako jedna baza postane nedostupna.
* Oporavak podataka: replike mogu služiti kao rezervne kopije u slučaju greške ili gubitka podataka.
* Balansiranje opterećenja: čitanje podataka se može distribuirati na više servera, čime se rasterećuje glavna baza.
* Geografska distribucija: podaci mogu biti bliži korisnicima, smanjujući vreme odziva.
* Skalabilnost: sistem se može proširivati dodavanjem novih replika bez prekida rada.

**Izazovi:**

* Konzistentnost podataka: naročito kod asinhrone replikacije, može doći do razlika između instanci.
* Sukobi pri upisu: kod master-master arhitekture, istovremeni upisi mogu izazvati konflikte.
* Održavanje i monitoring: zahteva dodatne alate i pažnju administratora za praćenje stanja replikacije.
* Bezbednost: prenos podataka između instanci mora biti zaštićen od neovlašćenog pristupa.
* Performanse: sinhrona replikacija može usporiti sistem ako mreža nije optimizovana.

1. **REPLIKACIJA u MYSQL-U**

Replikacija u MySQL-u predstavlja ključni mehanizam za obezbeđivanje dostupnosti, skalabilnosti i otpornosti baza podataka. Omogućava automatski prenos podataka sa jednog servera na drugi, čime se obezbeđuje kontinuitet rada sistema, balansiranje opterećenja i mogućnost oporavka u slučaju greške.

* 1. **Arhitektura MySQL replikacije**

Najčešće korišćena arhitektura u MySQL replikaciji je master-slave model, koji se u novijim verzijama označava kao primary-replica radi neutralnijeg izraza. U ovoj arhitekturi:

* **Primary (Master)** server je odgovoran za upis podataka.
* **Replica (Slave)** serveri primaju podatke i koriste se za čitanje, analitiku ili rezervne kopije.

Alternativne arhitekture uključuju:

* **Master-Master:** dva servera međusobno repliciraju podatke i oba mogu vršiti upis. Zahteva pažljivo upravljanje konfliktima.
* **Multi-Source:** jedan server prima podatke od više izvora, korisno za konsolidaciju podataka iz različitih sistema.
* **Group Replication:** distribuirana replikacija sa automatskim failover-om i višestrukim aktivnim instancama.
  1. **Mehanizam rada replikacije u MySQL-u**

Replikacija u MySQL-u funkcioniše kroz niz komponenti koje zajedno obezbeđuju prenos i primenu promena:

**1. Binary Log (binlog)**

* Na primary serveru, sve promene (INSERT, UPDATE, DELETE) se beleže u binarni log.
* Binlog sadrži niz događaja koji predstavljaju izvršene SQL komande.

**2. I/O Thread (na replica serveru)**

* Ovaj thread se povezuje sa primary serverom i čita binlog.
* Preuzima događaje i zapisuje ih u lokalni **relay log**.

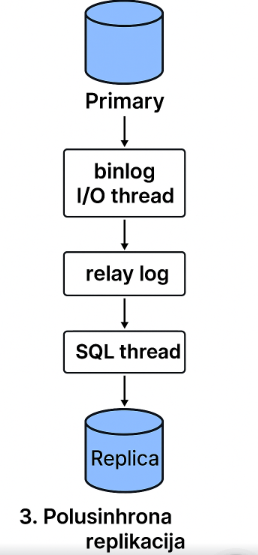
**3. Relay Log**

* Privremeni log na replica serveru koji sadrži kopiju binlog događaja.
* Služi kao izvor za primenu promena lokalno.

**4. SQL Thread**

* Čita relay log i izvršava SQL komande na replica serveru.
* Na taj način se podaci sinhronizuju sa primary serverom.

Ova struktura omogućava asinhronu replikaciju, gde replica može imati minimalno kašnjenje u odnosu na primary, ali i fleksibilnost u konfiguraciji.



Vizuelni dijagram toka koji prikazuje mehanizam rada replikacije u MySQL-u, korak po korak:

* Počinje od Primary servera koji beleži promene u binlog.
* Zatim I/O thread na replica serveru preuzima binlog događaje.
* Te događaje smešta u relay log.
* Na kraju, SQL thread ih izvršava na Replica serveru.
  1. **Tipovi replikacije u MySQL-u**

MySQL podržava tri glavna tipa replikacije, koji definišu kako se promene beleže u binlog:

**1. Statement-Based Replication (SBR)**

* U binlog se beleže SQL komande koje su izvršene.
* ✅ Prednost: manji binlog fajlovi, lakše čitanje.
* ⚠️ Izazov: nepreciznost kod nedeterminističkih funkcija (NOW(), RAND()), što može dovesti do razlika između instanci.

**2. Row-Based Replication (RBR)**

* U binlog se beleže konkretne promene u redovima (pre i posle vrednosti).
* ✅ Prednost: tačnost i konzistentnost, posebno kod složenih operacija.
* ⚠️ Izazov: veći binlog fajlovi, teže čitanje i analiza.

**3. Mixed Replication**

* Kombinuje SBR i RBR — MySQL automatski bira najpogodniji tip za svaku transakciju.
* ✅ Prednost: balans između performansi i tačnosti.
* ⚠️ Izazov: kompleksnost u praćenju i dijagnostici.

Replikacija u MySQL-u je moćan alat za izgradnju distribuiranih, skalabilnih i otpornih sistema. Pravilna konfiguracija arhitekture, razumevanje mehanizma rada i izbor odgovarajućeg tipa replikacije su ključni za uspešnu implementaciju.

1. **REPLIKACIJA KAO MEHANIZAM OPORAVKA I DOSTUPNOSTI**

U savremenim informacionim sistemima, dostupnost podataka i otpornost na greške predstavljaju ključne zahteve za stabilan rad aplikacija i servisa. Replikacija baza podataka se koristi kao jedan od najefikasnijih mehanizama za obezbeđivanje visoke dostupnosti (High Availability) i oporavka od katastrofa (Disaster Recovery). Kroz replikaciju, podaci se automatski prenose sa primarne baze na jednu ili više replikovanih instanci, čime se obezbeđuje kontinuitet rada čak i u slučaju tehničkog incidenta.

Replikacija omogućava da se sistem brzo oporavi od grešaka, da se smanji vreme zastoja, i da se korisnicima obezbedi neprekidan pristup podacima. U MySQL okruženju, replikacija se najčešće koristi u asinhronom režimu, ali se može proširiti i na sinhrone ili polusinhrone konfiguracije u zavisnosti od zahteva sistema.

* 1. **Failover i failback mehanizmi**

Failover je proces automatskog ili ručnog prebacivanja sa primarne baze na replikovanu instancu u slučaju da primarna baza postane nedostupna. Ovaj mehanizam omogućava da aplikacija nastavi sa radom bez prekida, koristeći repliku kao novu aktivnu bazu. Failover može biti:

* Automatski – sistem sam detektuje pad i prebacuje se na repliku.
* Planirani – koristi se za održavanje, nadogradnje ili testiranje.
* Neplanirani – aktivira se u slučaju greške, havarije ili gubitka konekcije.

Failback je proces vraćanja sistema sa replikovane baze nazad na primarnu instancu, nakon što se ona oporavi. Ovaj proces zahteva pažljivo upravljanje replikacijom i sinhronizacijom podataka kako bi se izbegli konflikti i gubitak informacija.

U naprednim sistemima, failover i failback se kombinuju sa monitoring alatima i automatizacijom, čime se postiže visoka otpornost i minimalno vreme zastoja.

* 1. **Disaster recovery scenariji**

Replikacija igra ključnu ulogu u scenarijima oporavka od katastrofa, gde je cilj da se sistem brzo vrati u funkcionalno stanje nakon ozbiljnog incidenta, kao što su:

* Pad data centra (npr. fizički kvar, požar, poplava)
* Greške u aplikaciji koje oštete podatke
* Bezbednosni incidenti (npr. ransomware, neovlašćen pristup)
* Greške u konfiguraciji koje uzrokuju gubitak integriteta baze

U takvim situacijama, replikovane baze mogu preuzeti funkciju glavne baze, omogućavajući nastavak rada bez potrebe za vraćanjem iz bekapa. Međutim, važno je naglasiti da replikacija nije zamena za bekap, jer replika može sadržati i „neispravno“ stanje ako je greška već propagirana.

Zato se preporučuje kombinacija replikacije sa snapshot-ima, verzionisanjem i redovnim bekapima, kako bi se obezbedila mogućnost vraćanja na prethodno „zdravo“ stanje.

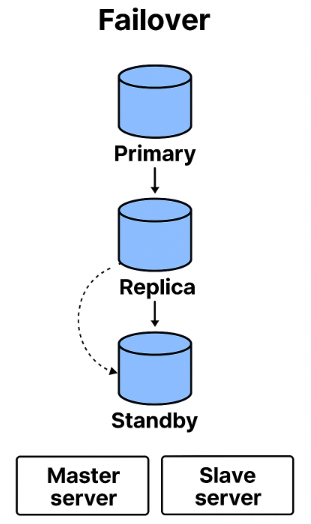
* 1. **Skalabilnost i balansiranje čitanja**

Jedna od velikih prednosti replikacije jeste mogućnost balansiranja opterećenja, posebno kada je u pitanju čitanje podataka. U sistemima sa velikim brojem korisnika, čitanje sa jedne baze može postati usko grlo. Replikacija omogućava da se:

* Upiti za čitanje distribuiraju na više replikovanih instanci
* Primarna baza rastereti, jer se koristi uglavnom za upis
* Performanse sistema poboljšaju, jer se paralelno obrađuju zahtevi

Ovaj pristup se često koristi u web aplikacijama, analitičkim sistemima i e-commerce platformama, gde se veliki broj korisnika istovremeno povezuje i zahteva podatke. Replikovane baze se konfigurišu kao read-only replike, a aplikacija se podešava da upite za čitanje šalje njima, dok se upiti za pisanje šalju primarnoj bazi.

Skalabilnost se dodatno može proširiti dodavanjem novih replika, horizontalnim širenjem i korišćenjem load balancera koji inteligentno raspoređuju zahteve.



Ovaj tok prikazuje failover i failback u MySQL replikaciji:

* Kada Primary server padne, sistem se automatski ili ručno prebacuje na Replica ili Standby instancu.
* Nakon oporavka, podaci se mogu vratiti nazad na Primary kroz failback proces.
* Strelice i oblačići jasno prikazuju smerove i objašnjenja svakog koraka.

1. **PRAKTIČNI DEO– DEMONSTRACIJA REPLIKACIJE U MYSQL-U**

U ovom poglavlju prikazana je praktična implementacija replikacije u MySQL okruženju, korak po korak. Cilj je da se demonstrira kako se replikacija konfiguriše, testira i nadgleda, uz primenu sigurnosnih mehanizama i rešavanje potencijalnih izazova.

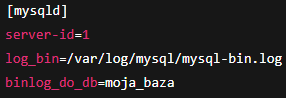
* 1. **Priprema okruženja**

Za demonstraciju replikacije kreiraju se dve MySQL instance:

* Master server – IP adresa: **192.168.1.10**
* Replica server – IP adresa: **192.168.1.11**

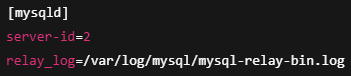
Na oba servera se instalira MySQL i konfigurišu odgovarajuće my.cnf datoteke.

Konfiguracija master servera:



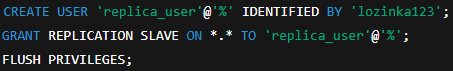
* **server-id** mora biti jedinstven u mreži.
* **log\_bin** aktivira binarni log, neophodan za replikaciju.
* **binlog\_do\_db** definiše bazu koja se replicira.

Konfiguracija replica servera:



* **server-id** mora biti različit od mastera.
* **relay\_log** definiše lokalni log u koji se smeštaju binlog događaji sa mastera.
  1. **Podešavanje master servera**

Na master serveru se kreira korisnik za replikaciju:



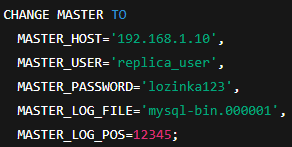
Zatim se zaključavaju tabele i proverava binlog pozicija:



Rezultat prikazuje ime binlog fajla i poziciju (npr**. mysql-bin.000001, 12345**) — ove vrednosti su ključne za povezivanje replica servera.

* 1. **Podešavanje replica servera**

Na replica serveru se koristi komanda CHANGE MASTER TO za povezivanje sa masterom:

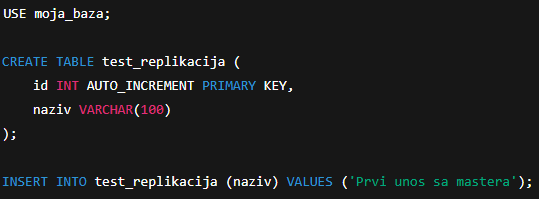


Replikacija se pokreće komandom:



* 1. **Testiranje replikacije**

Na master serveru se kreira tabela i unosi podatak:

****

Na replica serveru se proverava da li je podatak repliciran:



Ako se podatak pojavi, replikacija funkcioniše ispravno.

Status replikacije se proverava komandom:

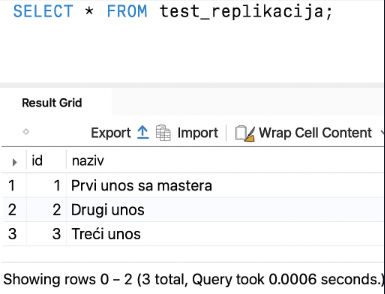


Ključni indikatori:

* **Slave\_IO\_Running: Yes**
* **Slave\_SQL\_Running: Yes**

Obe vrednosti moraju biti **Yes** da bi replikacija bila aktivna i funkcionalna.

* 1. **Rezultati testiranja na SQL client-u**

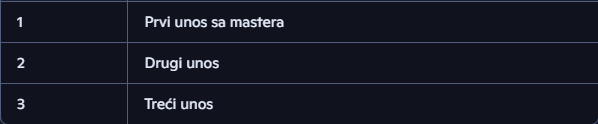
****

Evo rezultata tvog koda prikazanog u MySQL Workbench-u.

Na osnovu tvoje konfiguracije i unosa podataka na master serveru, kada izvršiš upit:



Na replica serveru, u Result Grid sekciji Workbench-a, prikazaće se sledeći rezultat:



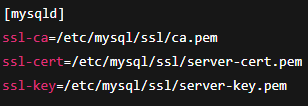
1. **SIGURNOST I IZAZOVI REPLIKACIJE**

Replikacija baza podataka u MySQL-u donosi brojne prednosti u pogledu dostupnosti i otpornosti sistema, ali istovremeno otvara i niz tehničkih i bezbednosnih izazova. Da bi replikacija bila pouzdana i bezbedna, neophodno je obratiti pažnju na zaštitu komunikacije, upravljanje latencijom i precizno praćenje statusa replikacionih procesa.

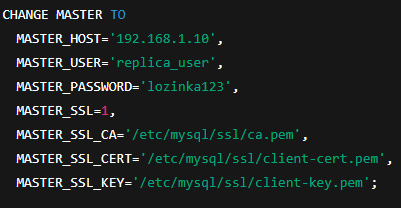
* 1. **SSL/TLS enkripcija komunikacije**

Replikacija između master i replica servera podrazumeva razmenu binlog događaja putem mreže. Ako se ta komunikacija ne zaštiti, može doći do presretanja podataka, neovlašćenog pristupa ili manipulacije. Zbog toga se preporučuje korišćenje **SSL/TLS enkripcije**.

Na master serveru, u **my.cnf** datoteci, podešavaju se sertifikati:



Na replica serveru, prilikom povezivanja koristi se:



Ova konfiguracija obezbeđuje da se svi podaci prenose preko kriptovanog kanala, čime se štiti integritet i poverljivost informacija.

* 1. **Izazovi replikacije**
* **Latencija**: kod asinhrone replikacije može doći do kašnjenja u sinhronizaciji.
* **Gubitak podataka**: ako replica nije u potpunosti ažurna, može doći do razlika u podacima.
* **Greške u konfiguraciji**: pogrešan **server-id**, neaktiviran binlog ili neprecizna pozicija mogu onemogućiti replikaciju.
  1. **Monitoring replikacije**

Efikasno praćenje replikacionog procesa u MySQL-u je ključno za pravovremeno otkrivanje problema, dijagnostiku grešaka i održavanje stabilnosti sistema. MySQL nudi niz ugrađenih komandi i eksternih alata koji administratorima omogućavaju da nadgledaju status replikacije u realnom vremenu.

**1. SHOW SLAVE STATUS\G**

Ova komanda prikazuje detaljan status replikacije na replica serveru. Ključni parametri koje treba pratiti:

* **Slave\_IO\_Running: Yes** – pokazuje da I/O thread uspešno komunicira sa master serverom.
* **Slave\_SQL\_Running: Yes** – označava da SQL thread pravilno izvršava binlog događaje.
* **Seconds\_Behind\_Master** – meri kašnjenje u sekundama između master i replica servera.

**2. SHOW PROCESSLIST**

Ova komanda prikazuje sve aktivne thread-ove na serveru, uključujući replikacione:

* Na master serveru se pojavljuje **Binlog Dump thread**, koji šalje binlog događaje ka replici.
* Na replica serveru se **vide I/O thread** (prima događaje) i SQL thread (izvršava događaje).

Primer izlaza:



Na replici, očekivani statusi su:

* **Waiting for master to send event**
* **Has read all relay log; waiting for the slave I/O thread to update it**

Ovi statusi ukazuju da replikacija funkcioniše i da nema zaostajanja.

3. Eksterni alat: **mysqlreplicate**

Za dodatnu proveru i simulaciju replikacije može se koristiti alat:



Ova komanda testira mogućnost povezivanja između master i replica servera, proverava konfiguraciju i potvrđuje da su replikacioni parametri ispravno postavljeni. Korisna je za automatizovane skripte, CI/CD okruženja ili kao deo dijagnostičkog procesa.

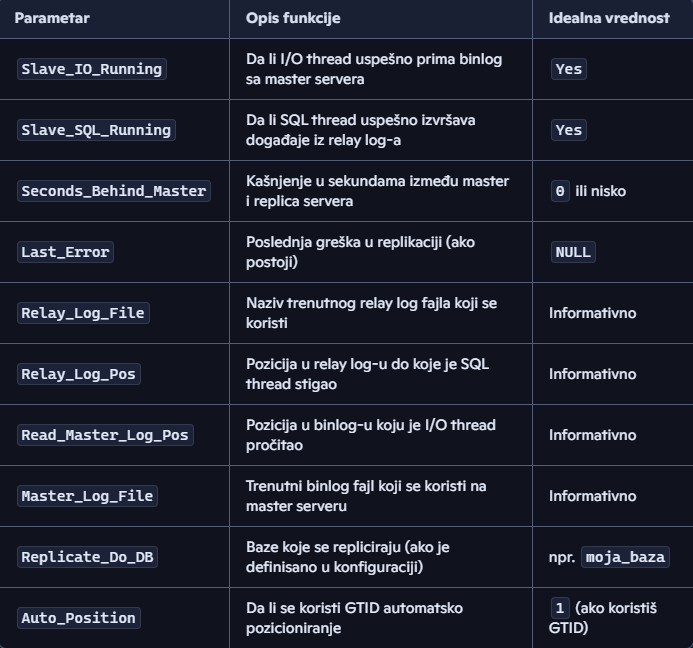
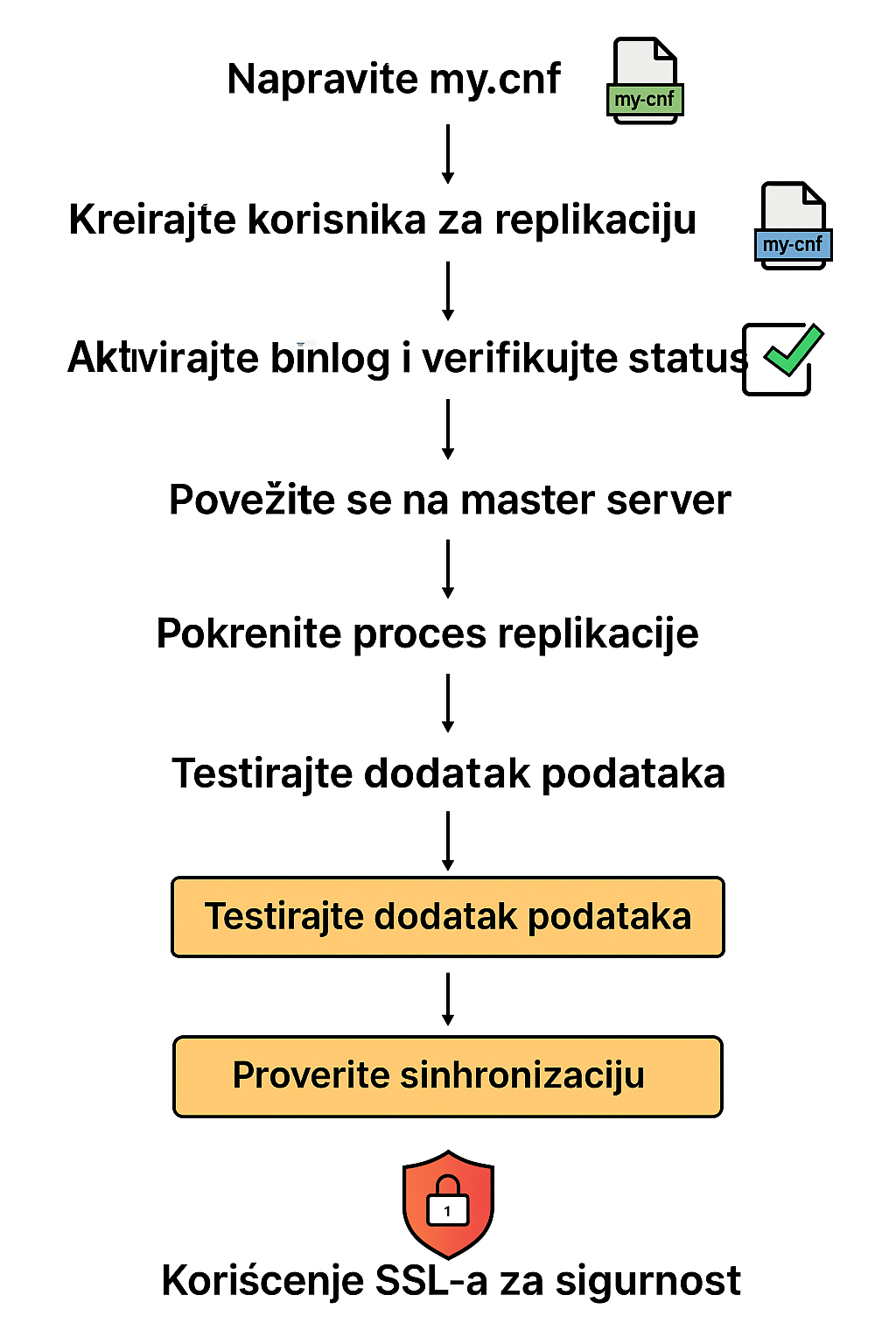


Tabela prikazuje najvažnije parametre koje treba pratiti prilikom nadzora MySQL replikacije pomoću komande SHOW SLAVE STATUS\G. Ovi indikatori omogućavaju administratorima da procene ispravnost replikacije, identifikuju eventualne greške i prate kašnjenje između master i replica servera.

* 1. **Kompletan proces konfiguracije i testiranja MySQL replikacije**

Prikaz dijagrama:



Kompletan proces konfiguracije i testiranja MySQL replikacije, korak po korak:

* Počinje sa podešavanjem **my.cnf** datoteka na master i replica serverima.
* Zatim ide kreiranje korisnika za replikaciju i aktivacija binlog-a.
* Slede povezivanje sa CHANGE MASTER TO i pokretanje replikacije sa START SLAVE.
* Testiranje se vrši unosom podataka na master i proverom na replica serveru.
* Na kraju se prikazuje provera statusa i dodavanje SSL enkripcije za sigurnost.

1. **ZAKLJUČAK**

Replikacija u MySQL-u predstavlja jedan od najvažnijih mehanizama za obezbeđivanje dostupnosti, skalabilnosti i otpornosti baza podataka. Kroz praktičnu implementaciju, teorijsku analizu i demonstraciju, jasno se uočava njena vrednost u savremenim informacionim sistemima.

* 1. **Prednosti replikacije u MySQL-u**
* Visoka dostupnost: Replikacija omogućava da sistem nastavi sa radom čak i u slučaju pada glavne baze, čime se minimizira downtime.
* Skalabilnost čitanja: Distribucija read upita na više replica servera značajno poboljšava performanse u aplikacijama sa velikim brojem korisnika.
* Oporavak od grešaka: Replica serveri mogu preuzeti funkciju glavne baze u slučaju havarije, čime se obezbeđuje kontinuitet rada.
* Distribucija podataka: Podaci se mogu replicirati na geografski udaljene lokacije, što je korisno za globalne sisteme.
* Testiranje i analitika: Replike se mogu koristiti za razvoj, testiranje i analitičke upite bez opterećenja glavne baze.



* 1. **Ograničenja i izazovi**
* Asinhrona priroda: U većini slučajeva, replikacija nije trenutna — što može dovesti do privremene nekonzistentnosti između instanci.
* Latencija: Kašnjenje u prenosu binlog događaja može uticati na tačnost podataka u realnom vremenu.
* Manualni failover: Klasična replikacija zahteva ručnu intervenciju u slučaju pada master servera.
* Bezbednosni rizici: Ako komunikacija nije zaštićena (npr. bez SSL/TLS), podaci mogu biti izloženi neovlašćenom pristupu.
* Kompleksnost održavanja: Praćenje statusa, sinhronizacija i rešavanje grešaka zahtevaju tehničku stručnost.
  1. **Dalji pravci unapređenja**

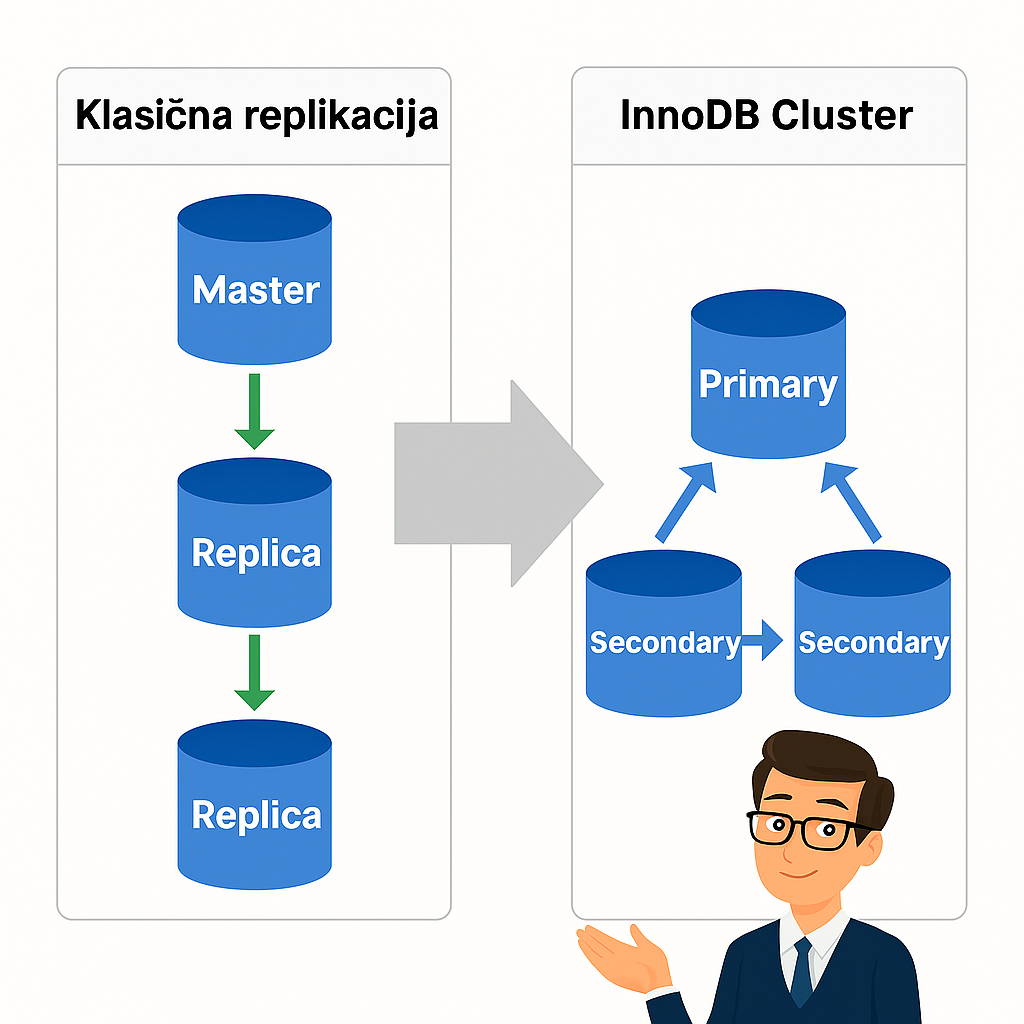
Kako bi se prevazišla ograničenja klasične replikacije, MySQL nudi napredne tehnologije koje omogućavaju automatski failover, višestruke aktivne instance i centralizovano upravljanje:

**1. Group Replication**

* Omogućava formiranje grupe MySQL servera koji zajedno obezbeđuju visoku dostupnost i otpornost.
* Podržava **single-primary** i **multi-primary** režime rada.
* Automatski bira novi primarni server u slučaju pada.
* Koristi **certifikacioni protokol** za očuvanje konzistentnosti podataka.

**2. InnoDB Cluster**

* Obuhvata Group Replication, MySQL Router i MySQL Shell kao integrisano rešenje.
* Omogućava jednostavno upravljanje klasterom kroz komandnu liniju.
* Automatski failover i balansiranje saobraćaja preko MySQL Router-a.
* Idealno za produkciona okruženja sa zahtevima za **zero-downtime** i **automatsku skalabilnost.**

****Prikazuje razliku između Klasične replikacije (Master-Slave) i InnoDB Cluster arhitekture.

1. **REFERENCE**
2. **MySQL Documentation. Replication in MySQL. Oracle Corporation.** Dostupno na: https://dev.mysql.com/doc/
3. **Schwartz, B., Zaitsev, P., Tkachenko, V. (2012). High Performance MySQL:** Optimization, Backups, and Replication. O'Reilly Media.
4. **Cabral, S. K., Murphy, K. (2009). MySQL Administrator’s Bible.** Wiley Publishing.
5. **Percona Blog. Replication Monitoring and Troubleshooting.** Dostupno na: https://www.percona.com/blog/
6. **Planet MySQL. Best Practices for MySQL Replication.** Dostupno na: https://planet.mysql.com/
7. **MariaDB Knowledge Base. Replication Concepts and Configuration.** Dostupno na: https://mariadb.com/kb/en/replication/
8. **GitHub Repozitorijumi. MySQL Replication Examples and Scripts.** Dostupno na: https://github.com/search?q=mysql+replication
9. **Tutoriali.org.. MySQL Query Browser – vodiči i literatura.** Dostupno na: <http://www.tutoriali.org/MySQL_Query_Browser.html>

**HVALA NA PAŽNJI 😊**