

MySQL Trigeri

Predmetni asistent:

Jovana Stojanović, dipl. inž. el. i rač.

jovana.stojanovic@akademijanis.edu.rs

Šta su zapravo Trigeri?

- Triger je imenovani SQL objekat koji se **automatski aktivira** (okida) kao odgovor na određeni događaj nad tabelom.
- **Događaji koji aktiviraju triger:**
 - INSERT (dodavanje novog reda)
 - UPDATE (izmena postojećeg)
 - DELETE (brisanje)
- Glavna svrha: Automatizacija, revizija i sprečavanje nevalidnih unosa koje CHECK ograničenja ne mogu da uhvate.
 - **Metafora:** Triger je kao alarmni sistem na kući. Vi ne palite alarm ručno svaki put kad neko provali; alarm je programiran da "reaguje" čim se otvore vrata (događaj).

Anatomija Trigera (Kada i Šta?)

- Da bismo definisali triger, moramo odgovoriti na dva pitanja:
 - **Vreme (Timing):** Da li triger radi **PRE** (BEFORE) ili **POSLE** (AFTER) događaja?
 - **Operacija:** Nad kojom komandom? (INSERT, UPDATE ili DELETE)
- **Kombinacije:**
 - BEFORE INSERT – Provera podataka pre nego što uopšte uđu u bazu.
 - AFTER UPDATE – Zapisivanje stare vrednosti u istoriju (log) nakon što je promena izvršena.

OLD i NEW (Ključne reči)

- Trigeri vide dve verzije reda:
 - **NEW**: Referenca na nove podatke koji se unose ili menjaju (koristi se kod INSERT i UPDATE).
 - **OLD**: Referenca na stare podatke koji su bili u tabeli pre promene (koristi se kod UPDATE i DELETE).

Operacija	OLD	NEW
INSERT	✗ (ne postoji staro)	✓ (novi red)
UPDATE	✓ (pre izmene)	✓ (posle izmene)
DELETE	✓ (šta brišemo)	✗ (ništa novo)

Praktičan primer – Automatska validacija

```

DELIMITER //
CREATE TRIGGER pre_unosa_uplate
BEFORE INSERT ON payment
FOR EACH ROW
BEGIN
    IF NEW.amount < 0 THEN
        SET NEW.amount = 0;
    END IF;
END //
DELIMITER ;
  
```

- **Scenario:** Sprečiti da se u tabelu payment unese negativan iznos. Ako je iznos negativan, automatski ga postavi na 0.
- **Zašto BEFORE?**
 - Jer želimo da "popravimo" podatak pre nego što se trajnoupiše na disk.

Praktičan primer – Audit Log (Istorija promena)

- **Scenario:** Svaki put kada se kupcu (customer) promeni adresa, upiši staru adresu u tabelu za reviziju.

```

DELIMITER //
CREATE TRIGGER nakon_izmene_kupca
AFTER UPDATE ON customer
FOR EACH ROW
BEGIN
    IF OLD.address_id <> NEW.address_id THEN
        INSERT INTO customer_log(customer_id, old_address, change_date)
        VALUES (OLD.customer_id, OLD.address_id, NOW());
    END IF;
END //
DELIMITER ;
    
```

- **Zašto AFTER?**

- Jer reviziju radimo tek kada smo sigurni da je primarna tabela uspešno ažurirana.

Ograničenja i oprez

- Trigeri su moćni, ali opasni ako se zloupotrebe:
 - **Skrivena logika:** Programer koji gleda samo kod aplikacije neće znati zašto se podaci menjaju "sami od sebe".
 - **Performanse:** Trigeri usporavaju INSERT/UPDATE/DELETE operacije jer baza mora da odradi dodatni posao.
 - **Nema direktnog pozivanja:** Triger ne možete pozvati sa CALL. On se "budi" isključivo akcijom nad tabelom.