

## Laboratorijska vežba br. 6.

# MySQL Trigeri

## 1. Cilj vežbe

- **Trigeri:** Kreiranje automatskih "okidača" na INSERT, UPDATE i DELETE događaje.
- **Integritet:** Automatska revizija (logovanje) i sprečavanje nevalidnih unosa podataka.

## 2. Neophodni preduslovi

- **Baza:** Instalirana **Sakila** (tabele: film, customer, staff, rental, actor).
- **Sintaksa:** Poznavanje SELECT upita, agregatnih funkcija i razlike između OLD i NEW stanja.
- **Alati:** Korišćenje DELIMITER komande za pisanje višelinjskih blokova koda.

## 3. Upiti

Napisati SQL naredbe za sledeće upite:

1. Kreirati triger trg\_actor\_uppercase koji će se aktivirati **pre unosa** novog reda u tabelu actor. Zadatak trigera je da osigura da se ime (first\_name) i prezime (last\_name) uvek čuvaju velikim slovima, bez obzira na to kako ih korisnik ukuca u aplikaciji.
  - a. Proveriti sa:

```
INSERT INTO actor (first_name, last_name)
VALUES ('marko', 'markovic');
SELECT * FROM actor WHERE last_name = 'MARKOVIC';
```

Očekivan rezultat – slika 1:

	actor_id	first_name	last_name	last_update
▶	201	MARKO	MARKOVIC	2026-03-25 20:11:38
★	NULL	NULL	NULL	NULL

Slika 1.

Rešenje:

```
DELIMITER //
```

```
CREATE TRIGGER trg_actor_uppercase
```

```
BEFORE INSERT ON actor
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    SET NEW.first_name = UPPER(NEW.first_name);
```

```
    SET NEW.last_name = UPPER(NEW.last_name);
```

```
END //
```

```
DELIMITER ;
```

2. Kreirati triger trg\_film\_check\_rental koji se aktivira **pre unosa** novog filma u tabelu film. Triger treba da proveriti kolonu rental\_duration. Ako je uneta vrednost manja od 1 (što nema smisla za iznajmljivanje), triger treba da je automatski postavi na minimalnu vrednost od 3 dana.

- a. Proveriti sa:

```
INSERT INTO film (title, language_id, rental_duration, rental_rate,  
replacement_cost)
```

```
VALUES ('Najnoviji Film', 1, 0, 4.99, 19.99);
```

```
SELECT title, rental_duration FROM film WHERE title = 'Najnoviji Film';
```

Očekivan rezultat – slika 2:

	title	rental_duration
▶	Najnoviji Film	3

Slika 2.

Rešenje:

```
DELIMITER //
```

```
CREATE TRIGGER trg_film_check_rental
```

```
BEFORE INSERT ON film
```

```
FOR EACH ROW
```



```
BEGIN
  IF NEW.rental_duration < 1 THEN
    SET NEW.rental_duration = 3;
  END IF;
END //
DELIMITER ;
```

3. Potrebno je pratiti promene plata ili statusa zaposlenih. Prvo kreirajte pomoćnu tabelu staff\_log sa kolonama staff\_id, old\_email i change\_date. Zatim kreirajte trigger trg\_staff\_email\_log koji će **nakon svake izmene** u tabeli staff proveriti da li je promenjena kolona email. Ako jeste, staru email adresu upisati u log tabelu.

Pomoćna tabela:

```
CREATE TABLE staff_log (
  log_id INT AUTO_INCREMENT PRIMARY KEY,
  staff_id INT,
  old_email VARCHAR(50),
  change_date DATETIME
);
```

- a. Proveriti sa:

```
UPDATE staff
SET email = 'novi.email@sakila.com'
WHERE staff_id = 1;
```

```
SELECT * FROM staff_log;
```

Očekivan rezultat – slika 3:

	log_id	staff_id	old_email	change_date
▶	1	1	Mike.Hillyer@sakilastaff.com	2026-03-25 20:14:45
*	NULL	NULL	NULL	NULL

Slika 3.

Rešenje:

```
DELIMITER //
CREATE TRIGGER trg_staff_email_log
AFTER UPDATE ON staff
FOR EACH ROW
```

```
BEGIN
  IF OLD.email <> NEW.email THEN
    INSERT INTO staff_log (staff_id, old_email, change_date)
    VALUES (OLD.staff_id, OLD.email, NOW());
  END IF;
END //
DELIMITER ;
```

4. Kreirati triger `trg_prevent_customer_delete` koji se aktivira **pre brisanja** iz tabele `customer`. Kako bismo sačuvali integritet istorije iznajmljivanja, triger treba da proveriti kolonu `active`. Ako je kupac označen kao aktivan (`active = 1`), triger mora da spreči brisanje i izbaciti poruku greške: 'Nije dozvoljeno brisanje aktivnih kupaca!'.
- a. Proveriti sa:
- i. `SELECT customer_id, first_name FROM customer WHERE active = 1 LIMIT 1;`

Očekivan rezultat – slika 4:

	customer_id	first_name
▶	1	MARY
✱	NULL	NULL

Slika 4.

- ii. `DELETE FROM customer WHERE customer_id = 1;`

Očekivan rezultat – slika 5:

20 20:17:09 DELETE FROM customer WHERE customer\_id = 1 Error Code: 1644. Nije dozvoljeno brisanje aktivnih kupaca!

Slika 5.

Rešenje:  
DELIMITER //  
CREATE TRIGGER `trg_prevent_customer_delete`

```
BEFORE DELETE ON customer
FOR EACH ROW
BEGIN
  IF OLD.active = 1 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Nije dozvoljeno brisanje aktivnih kupaca!';
  END IF;
END //
DELIMITER ;
```

5. Kreirati trigger `trg_update_last_update` koji se aktivira **pre svake izmene** u tabeli `address`. Zadatak triggera je da u kolonu `last_update` automatski upiše trenutno vreme (`NOW()`), kako bismo uvek znali kada je adresa kupca poslednji put ažurirana, bez manuelnog unosa tog podatka.

- a. Proveriti sa:

```
UPDATE address
SET phone = '123-456'
WHERE address_id = 1;
```

```
SELECT address_id, phone, last_update FROM address WHERE
address_id = 1;
```

Očekivan rezultat – slika 6:

	address_id	phone	last_update
▶	1	123-456	2026-03-25 20:18:57
•	NULL	NULL	NULL

Slika 6.

Rešenje:

```
DELIMITER //
CREATE TRIGGER trg_update_last_update
BEFORE UPDATE ON address
FOR EACH ROW
BEGIN
  SET NEW.last_update = NOW();
END //
DELIMITER ;
```

**Katedra za informaciono-komunikacione tehnologije**

**Predmet:** Administriranje baza podataka

**Asistent na predmetu:** Jovana Stojanović

