

Nadoknada laboratorijske vežbe 6. Trigeri

1. Kreirati triger trg_actor_caps 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) čuvaju isključivo velikim slovima, bez obzira kako korisnik unese podatke.

Proveriti sa:

```
INSERT INTO actor (first_name, last_name)
VALUES ('ana', 'anic');
```

```
SELECT *
FROM actor
WHERE last_name = 'ANIC';
```

Očekivan rezultat je na slici 1:

	actor_id	first_name	last_name	last_update
▶	202	ANA	ANIC	2026-05-14 19:14:01
•	NULL	NULL	NULL	NULL

Slika 1.

2. Kreirati triger trg_film_min_duration koji se aktivira pre unosa novog filma u tabelu film. Ako korisnik unese vrednost manju od 1 za kolonu rental_duration, triger treba automatski da postavi vrednost na 5 dana.

Proveriti sa:

```
INSERT INTO film (title, language_id, rental_duration, rental_rate,
replacement_cost)
VALUES ('Akcioni Film', 1, 0, 3.99, 15.99);
```

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

Očekivan rezultat je na slici 2:

	title	rental_duration
▶	Akcioni Film	3

Slika 2.

3. Potrebno je pratiti izmene email adresa zaposlenih. Prvo kreirati pomoćnu tabelu employee_email_log:

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

Zatim kreirati triger trg_staff_email_history koji će nakon svake izmene u tabeli staff proveriti da li je promenjena email adresa. Ako jeste, prethodna email adresa treba da se upiše u log tabelu.

Proveriti sa:

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

```
SELECT * FROM employee_email_log;
```

Očekivan rezultat je na slici 3.

	log_id	staff_id	old_email	changed_at
▶	1	1	novi.email@sakila.com	2026-05-14 19:19:38
*	NULL	NULL	NULL	NULL

Slika 3.

4. Kreirati triger trg_block_active_customer_delete koji se aktivira pre brisanja iz tabele customer. Ako je korisnik aktivan (active = 1), triger treba da zabrani brisanje i prikaže poruku: Brisanje aktivnih korisnika nije dozvoljeno!

Proveriti sa:

```
SELECT customer_id, first_name  
FROM customer  
WHERE active = 1  
LIMIT 1;
```

Očekivan rezultat je na slici 4.1:

	customer_id	first_name
▶	1	MARY
✱	NULL	NULL

Slika 4.1.

Proveriti sa:

```
DELETE FROM customer  
WHERE customer_id = 1;
```

Očekivan rezultat je na slici 4.2:

✖	14	19:22:37	DELETE FROM customer WHERE customer_id ...	Error Code: 1644. Nije dozvoljeno brisanje aktivni...	0.016 sec
---	----	----------	--	---	-----------

Slika 4.2.

5. Kreirati triger trg_address_last_modified koji se aktivira pre svake izmene u tabeli address. Triger treba automatski da upiše trenutno vreme (NOW()) u kolonu last_update.

Proveriti sa:

```
UPDATE address  
SET phone = '555-999'  
WHERE address_id = 1;
```

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

Očekivan rezultat je na slici 5:

	address_id	phone	last_update
▶	1	555-999	2026-05-14 19:24:17
✱	NULL	NULL	NULL

Slika 5.