

Trigeri

(prema <http://www.w3resource.com/mysql/mysql-triggers.php>)

1.Zadatak Data je baza podataka **Test** sa sledećim relacijama:

```
student_mast(STUDENT_ID, NAME, ST_CLASS)
student_marks(STUDENT_ID, NAME, SUB1, SUB2, SUB3, SUB4, SUB5, TOTAL, PER_MARKS,
GRADE)
student_log(USER_ID, DESCRIPTION)
```

Primer trigera nad više atributa u relaciji:

Pretpostavimo da tabela student_marks sadrži podatke o studentima koji su se prijavili za polaganje testa. Kako test još nije završen, pretpostavimo da se u tabeli za svakog studenta nalaze samo vrednosti pripadajućih atributa STUDENT_ID i NAME dok su vrednosti svih ostalih atributa postavljene na podrazumevane vrednosti (0 ili NULL). Završen je test i studenti su dobili ocene za 5 različitih predmeta. Potrebno je izmeniti podatke o studentima u tabeli student_marks, pri čemu je za svakog studenta potrebno uneti ocene koje je dobio na testu za svaki od 5 predmeta. U skladu sa dobijenim ocenama potrebno je automatski izmeniti ukupnu ocenu za sve predmete (TOTAL), prosečnu ocenu (PER_MARKS) i ukupan uspeh na testu (GRADE). Vrednosti ovih atributa se izračunavaju na sledeći način:

Ukupna ocena: $TOTAL = SUB1 + SUB2 + SUB3 + SUB4 + SUB5$

Prosečna ocena: $PER_MARKS = (TOTAL)/5$

Uspeh: $GRADE =$

- If $PER_MARKS \geq 90$ -> 'EXCELLENT'
- If $PER_MARKS \geq 75$ AND $PER_MARKS < 90$ -> 'VERY GOOD'
- If $PER_MARKS \geq 60$ AND $PER_MARKS < 75$ -> 'GOOD'
- If $PER_MARKS \geq 40$ AND $PER_MARKS < 60$ -> 'AVERAGE'
- If $PER_MARKS < 40$ -> 'NOT PROMOTED'

Rešenje:

Zadatak ćemo rešiti korišćenjem trigera nad tabelom student_marks:

```
USE Test;
DELIMITER $$
CREATE TRIGGER student_marks_BUPD BEFORE UPDATE ON student_marks FOR EACH ROW
BEGIN
SET NEW.TOTAL = NEW.SUB1 + NEW.SUB2 + NEW.SUB3 + NEW.SUB4 + NEW.SUB5;
SET NEW.PER_MARKS = NEW.TOTAL/5;
IF NEW.PER_MARKS >=90 THEN SET NEW.GRADE = 'EXCELLENT';
ELSEIF NEW.PER_MARKS >=75 AND NEW.PER_MARKS <90 THEN SET NEW.GRADE = 'VERY
GOOD';
ELSEIF NEW.PER_MARKS >=60 AND NEW.PER_MARKS <75 THEN SET NEW.GRADE = 'GOOD';
ELSEIF NEW.PER_MARKS >=40 AND NEW.PER_MARKS <60 THEN SET NEW.GRADE =
'AVERAGE';
ELSE SET NEW.GRADE = 'NOT PROMOTED';
END IF;
```

```
END;  
$$
```

Primer trigera nad više relacija:

2.Zadatak Data je baza Test iz prethodnog zadatka. Potrebno je za sve studente iz tabele student_mast promeniti razred u sledeći. Kreirati triger koji proverava da uneti razred nije manji od starog razreda. Kreirati triger koji će obezbediti da se posle svake izmene pojedinačne n-torke u tabeli student_mast doda nova n-torka u tabelu student_log sa informacijom o korisniku (user_id) koji je izvršio izmenu i opisom u vezi izvršene izmene. Kreirati i triger koji nakon svakog brisanja n-torke iz tabele student_mast dodaje novu vrstu u tabelu student_log sa informacijom o korisniku koji vrši brisanje i kratkim opisom izvršenog brisanja.

Rešenje:

Use Test;

```
DELIMITER $$  
CREATE TRIGGER student_mast_BUPD BEFORE UPDATE ON student_mast  
FOR EACH ROW  
BEGIN  
DECLARE msg VARCHAR(255);  
IF (NEW.ST_CLASS < OLD.ST_CLASS)  
THEN  
SET msg='Greska: novi razred je manji od starog!';  
SIGNAL sqlstate '45000' SET message_text= msg;  
END IF ;  
END $$
```

```
CREATE TRIGGER student_mast_AUPD AFTER UPDATE ON student_mast FOR EACH ROW  
BEGIN  
INSERT into student_log VALUES (user(), CONCAT('Izmena podataka o student ', OLD.NAME,  
' Prethodni razred : ',OLD.ST_CLASS,' Novi razred : ', NEW.st_class));  
END$$
```

```
CREATE TRIGGER student_mast_ADEL AFTER DELETE ON student_mast FOR EACH ROW  
BEGIN INSERT into student_log VALUES (user(), CONCAT('Izbrisani podaci o studentu: '  
,OLD.NAME,' Razred :',OLD.ST_CLASS, '-> Datum brisanja : ', NOW()));  
END;  
$$  
DELIMITER ;
```

3.Zadatak Data je relacija emp_details kao deo baze ljudskih resursa:

```
emp_details(EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE,  
JOB_ID, SALARY , COMMISSION_PCT )
```

Kreirati triger koji omogućava da se nakon svakog unošenja podataka o zaposlenima u tabelu emp_details izvrši provera njihove zarade i ako je zarada veća od 20000 da se za tog

zaposlenog postavi provizija od urađenog posla (COMMISSION_PCT) na 0.1 a u suprotnom na 0.5.

Rešenje:

```
DELIMITER $$
CREATE TRIGGER emp_details_AINS AFTER INSERT ON emp_details FOR EACH ROW
BEGIN
IF NEW.SALARY >= 20000 THEN
UPDATE emp_details SET NEW.COMMISSION_PCT=0.1;
ELSEIF NEW.SALARY < 20000 THEN
UPDATE emp_details SET NEW.COMMISSION_PCT=0.5;
END IF;
END $$
DELIMITER;
```

(prema knjizi prof. Gordane Pavlović-Lažetić
<http://poincare.matf.bg.ac.rs/~gordana//FINALE.pdf>)

Data je aproksimacija izdavačke relacije baze sa sledećom shemom:

```
P (P_SIF, IME, BR_NASLOVA, DRZAVA)
I (I_SIF, NAZIV, STATUS, DRZAVA)
K (K_SIF, NASLOV, OBLAST)
KP (K_SIF, P_SIF, R_BROJ)
KI (K_SIF, I_SIF, IZDANJE, GODINA, TIRAZ)
```

Prve tri relacije predstavljaju tipove entiteta PISAC, IZDAVAC i KNJIGA redom. Relacija KP predstavlja apstraktni tip entiteta AUTOR, tj. odnos između tipova entiteta KNJIGA i PISAC, dok relacija KI predstavlja apstraktni tip entiteta IZDAVASTVO, tj. odnos između tipova entiteta KNJIGA i IZDAVAC.

4.Zadatak Prilikom registracije novog autorskog dela, odnosno prilikom unošenja n-torke u reklaciju KP, uvećati vrednost atributa BR_NASLOVA za jedan u okviru relacije P za pisca koji je autor unetog autorskog dela.

Rešenje u MySQL-u:

```
DELIMITER $$
CREATE TRIGGER KP_UNOS AFTER INSERT ON KP
FOR EACH ROW
BEGIN
DECLARE x INTEGER;
SET x = NEW.P_SIF;
UPDATE P
SET BR_NASLOVA=BR_NASLOVA+1 WHERE P_SIF=x;
END
$$DELIMITER ;
```

Na sličan način mogu se definisati trigeri pri brisanju, odnosno izmeni podataka o autorskom delu:

```
DELIMITER $$
CREATE TRIGGER KP_BRIS AFTER DELETE ON KP
FOR EACH ROW
BEGIN
DECLARE x INTEGER;
SET x = NEW.P_SIF;
UPDATE P
SET BR_NASLOVA=BR_NASLOVA-1 WHERE P_SIF=x;
END
$$
```

```
CREATE TRIGGER KP_AZUR AFTER UPDATE ON KP
FOR EACH ROW
BEGIN
DECLARE x,y INTEGER;
SET x = OLD.P_SIF;
SET y = NEW.P_SIF;
UPDATE P
SET BR_NASLOVA=BR_NASLOVA-1 WHERE P_SIF=x;
UPDATE P
SET BR_NASLOVA=BR_NASLOVA+1 WHERE P_SIF=y;
END
$$
DELIMITER ;
```