



TEHNIČKO VELEUČILIŠTE U ZAGREBU  
POLYTECHNICUM ZAGRABIENSE

Stručni studij informatike  
Stručni studij računarstva

# Napredne baze podataka

## Zadaci za vježbu

---

**Priprema za drugi međuispit**

**Akadska godina: 2013/14**

1. Postaviti okidač koji će prilikom brisanja zapisa iz tablice *mjesto* provjeriti je li to mjesto referencirano u tablici *klijent* ili *radnik*. Ako je referencirano, tada će postaviti vrijednosti tih zapisa u tablicama *klijent* i *radnik* na NULL.

```
DROP TRIGGER mjesto;
DELIMITER $$
CREATE TRIGGER mjesto BEFORE DELETE ON mjesto
FOR EACH ROW
BEGIN
    UPDATE klijent SET pbrreg=NULL WHERE pbrreg=OLD.pbrmjesto;
    UPDATE klijent SET pbrKlijent=NULL
        WHERE pbrKlijent=OLD.pbrmjesto;
    UPDATE radnik SET pbrStan=NULL WHERE pbrStan=OLD.pbrmjesto;
END;
$$
DELIMITER ;
```

2. Izvedite isti okidač pozivom procedure.

```
DROP PROCEDURE IF EXISTS radi_mjesto;
DELIMITER $$
CREATE PROCEDURE radi_mjesto(IN pbr INT)
BEGIN
    UPDATE klijent SET pbrreg=NULL WHERE pbrreg=pbr;
    UPDATE klijent SET pbrKlijent=NULL WHERE pbrKlijent=pbr;
    UPDATE radnik SET pbrStan=NULL WHERE pbrStan=pbr;
END;
$$
DELIMITER ;

DROP TRIGGER mjesto;
DELIMITER $$
CREATE TRIGGER mjesto BEFORE DELETE ON mjesto
FOR EACH ROW
BEGIN
    CALL radi_mjesto(OLD.pbrMjesto);
END;
$$
DELIMITER ;
```

3. Napisati okidač koji će se pokrenuti prilikom unosa novog zapisa u tablicu *rezervacija*. Okidač će provjeriti polje *satServis* i ako je unesen podatak 0 postaviti će mu vrijednost na 10.

```
DROP TRIGGER rezervac;
DELIMITER $$
CREATE TRIGGER rezervac BEFORE INSERT ON rezervacija
FOR EACH ROW
BEGIN
```

```

        IF NEW.satServis=0 THEN SET NEW.satServis=10;
    END IF;
END;
$$
DELIMITER ;

```

4. Napisati okidač koji će se pokretati prilikom ažuriranja zapisa u tablici *kvar*. Ako se unosi vrijednost atributa *brojRadnika* manji od dosadašnje vrijednosti, potrebno je unijeti novu vrijednost. U suprotnom je potrebno ostaviti dosadašnju vrijednost atributa.

```

DROP TRIGGER mjesto;
DELIMITER $$
CREATE TRIGGER pazi BEFORE UPDATE ON kvar
FOR EACH ROW
BEGIN
    IF new.brojRadnika>old.brojRadnika
        THEN SET new.brojRadnika=old.brojRadnika;
    END IF;
END;
$$
DELIMITER ;

```

```

UPDATE kvar SET brojRadnika=9 WHERE sifkvar=1;
UPDATE kvar SET brojRadnika=1 WHERE sifkvar=1;

```

5. Napisati okidač koji će prilikom unosa novog klijenta provjeriti njegov JMBG. Ako je klijent maloljetan, tada je potrebno postaviti njegov JMBG na '00000000000000'.

```

DELIMITER //
DROP TRIGGER IF EXISTS punoljetnost //
CREATE TRIGGER punoljetnost BEFORE INSERT ON klijent
FOR EACH ROW
BEGIN
    DECLARE dob, d, m, g, tmp, td, tm, tg INT DEFAULT NULL;
    SELECT SUBSTRING(new.jmbgKlijent, 1, 2) INTO d;
    SELECT SUBSTRING(new.jmbgKlijent, 3, 2) INTO m;
    SELECT SUBSTRING(new.jmbgKlijent, 5, 3) INTO tmp;
    IF tmp<500 THEN SELECT CONCAT('2',tmp) INTO g;
        ELSE SELECT CONCAT('1',tmp) INTO g;
    END IF;
    SELECT DAY(CURDATE()) INTO td;
    SELECT MONTH(CURDATE()) INTO tm;
    SELECT YEAR(CURDATE()) INTO tg;
    IF tg<g THEN SET dob=NULL;
        ELSEIF tm<m OR (tm=m AND td<d) THEN SET dob=tg-g-1;
        ELSE SET dob=tg-g;
    END IF;
    IF dob<18 THEN SET new.jmbgKlijent='00000000000000';
    END IF;

```

```
END //
DELIMITER ;
```

Aktiviranje okidača:

```
INSERT INTO klijent (sifKlijent, jmbgKlijent) VALUES (4, '0101995228394');
```

6. Napisati okidač koji će prilikom unosa novog zapisa u tablicu *radnik* provjeriti jesu li dobro unesene vrijednosti atributa *pbrStan* i *sifOdjel* (referencira li se novi radnik na postojeće mjesto odnosno na postojeći odjel). Ako vrijednosti zadanih atributa nisu dobro unesene, tada ih je potrebno postaviti na NULL.

```
DELIMITER $$
CREATE TRIGGER tRadnik BEFORE INSERT ON radnik
FOR EACH ROW
BEGIN
    DECLARE st, odj INT;
    SELECT COUNT(*) INTO st
        FROM mjesto WHERE pbrMjesto=new.pbrStan;
    IF st=0 THEN
        SET NEW.pbrStan=NULL;
    END IF;
    SELECT COUNT(*) INTO odj
        FROM odjel WHERE odjel.sifOdjel=new.sifOdjel;
    IF odj=0 THEN
        SET new.sifOdjel=NULL;
    END IF;
END $$
DELIMITER ;
```

Aktiviranje okidača:

```
INSERT INTO radnik (sifRadnik, pbrStan, sifOdjel) VALUES (6, 90000, 76);
```

7. Napisati okidač koji se pokreće prilikom unosa novog zapisa u tablicu *nalog*. Potrebno je u novostvoreni atribut *ukupnoNaloga* u tablici *radnik* upisati koliki je ukupni broj naloga koje je radnik do sada zaprimio (novo stanje ukupnog broja naloga). Okidač je potrebno izvesti pozivom procedure.

```
ALTER TABLE radnik ADD ukupnoNaloga INT;
```

```
DELIMITER //
DROP PROCEDURE IF EXISTS brojiNaloge //
CREATE PROCEDURE brojiNaloge (IN zadaniRadnik INT)
BEGIN
    DECLARE n INT DEFAULT NULL;
    SELECT COUNT(*) INTO n FROM nalog WHERE sifRadnik=zadaniRadnik;
```

```
        UPDATE radnik SET ukupnoNaloga=n WHERE sifRadnik=zadaniRadnik;
END //
DELIMITER ;
```

**Primjer poziva procedure (za provjeru):**

```
CALL brojiNaloge(122);
```

```
DELIMITER //
DROP TRIGGER IF EXISTS pribrojiNalog //
CREATE TRIGGER pribrojiNalog AFTER INSERT ON nalog
FOR EACH ROW
BEGIN
    CALL brojiNaloge(new.sifRadnik);
END //
DELIMITER ;
```

**Primjer naredbe za aktiviranje okidača:**

```
INSERT INTO nalog VALUES (1257,26,122,CURDATE(),1,12);
```

#### 8. Zaključati tablicu *mjesto* za čitanje.

Pokušajte pročitati podatak iz tablice *mjesto*.

Pokušajte unijeti promjenu u tablicu *mjesto*.

Pokušajte pročitati podatak iz tablice *mjesto* u drugoj sesiji!

Pokušajte unijeti promjenu u tablicu *mjesto* u drugoj sesiji!

Otključajte tablicu.

Obratite pažnju što se događa u navedenim slučajevima!

**Prva sesija:**

```
LOCK TABLES mesto READ;
SELECT * FROM mesto WHERE pbrMjesto=10000;
INSERT INTO mesto SET pbrMjesto=10001;
```

**Druga sesija:**

```
SELECT * FROM mesto WHERE pbrMjesto=10000;
INSERT INTO mesto SET pbrMjesto=10002;
```

**Prva sesija:**

```
UNLOCK TABLES;
```

9. Zaključati tablicu *mjesto* za pisanje.

Pokušajte pročitati podatak iz tablice *mjesto* u istoj sesiji.

Pokušajte unijeti promjenu u tablicu *mjesto* u istoj sesiji.

Pokušajte pročitati podatak iz tablice *mjesto* u drugoj sesiji!

Pokušajte unijeti promjenu u tablicu *mjesto* u drugoj sesiji!

Otključajte tablicu.

Obratite pažnju što se događa u navedenim slučajevima!

Prva sesija:

```
LOCK TABLES mesto WRITE;  
SELECT * FROM mesto;  
INSERT INTO mesto SET pbrMjesto=10003;
```

Druga sesija:

```
SELECT * FROM mesto;  
INSERT INTO mesto SET pbrMjesto=10004;
```

Prva sesija:

```
UNLOCK TABLES;
```

10. Postavite dozvole za čitanje i unos podataka u tablicu *rezervacija* za korisnika *marko* nad svojom bazom s lokalnog poslužitelja. Pokušajte se spojiti s novim podacima te provjerite koje su radnje omogućene korisniku *marko*.

```
GRANT SELECT,INSERT ON autoradionica.rezervacija TO 'marko'@'localhost';
```

11. Korisniku *marko* postavite lozinku na "12345". Pokušajte se spojiti s novim podacima.

```
SET PASSWORD FOR 'marko'@'localhost' = PASSWORD ('12345');  
FLUSH PRIVILEGES;
```

12. Koristeći naredbu *revoke* ukinite dozvolu unosa podataka za korisnika *marko*. Spojite se kao korisnik *marko* i provjerite stanje dozvola.

```
REVOKE INSERT ON autoradionica.rezervacija FROM 'marko'@'localhost';  
FLUSH PRIVILEGES;
```

13. Uvidom u odgovarajuću tablicu u bazi *mysql* te brisanjem odgovarajućeg zapisa, skinite korisniku *marku* preostale dozvole. Spojite se kao korisnik *marko* te provjerite koje su radnje omogućene korisniku *marko*.

```
DELETE FROM mysql.tables_priv
      WHERE tables_priv.host='localhost' AND tables_priv.user='marko';
FLUSH PRIVILEGES;
```

14. Obrišite korisnika *marko*.

```
DROP USER 'marko'@'localhost';
DELETE FROM mysql.user WHERE user.host='localhost' AND user.user='marko';
```

15. Korisniku *jura* dodijeliti sve dozvole sa svih adresa nad svim bazama te mu dodijeliti lozinku „pass123“. *Jura* mora imati mogućnost dodjeljivati dozvole drugim korisnicima.

```
GRANT ALL ON *.* TO 'jura'@'%' IDENTIFIED BY 'pass123' WITH GRANT OPTION;
```

16. Ukinuti dozvole dodijeljene u prethodnom zadatku.

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'jura'@'%';
```

17. Dodijeliti dozvole korisniku *jura* za brisanje i ažuriranje podataka u svim tablicama baze *studenti* s lokalnog poslužitelja, uz ograničenje da u satu može odraditi maksimalno 100 ažuriranja.

```
GRANT DELETE, UPDATE ON studenti.* TO 'jura'@'localhost' WITH
MAX_UPDATES_PER_HOUR 100;
```

18. Postaviti razinu izolacije zaključavanja podataka na čisto čitanje.

```
SET TRANSACTION ISOLATION LEVEL READ COMMITED;
```

19. Napisati PHP skriptu koja ispisuje ime i prezime klijenta koji je zadnji unesen u bazu (ima najveću šifru).

Baza: mojabaza

Poslužitelj: localhost

Username: user1

Lozinka: lozinka

```
<?php
$host="localhost";
$user="user1";
$pass="lozinka";

$link = mysql_connect ($host, $user, $pass)
    or die ("Neuspjesno spajanje na server");
$link2=mysql_select_db ("mojabaza",$link)
    or die ("Neuspjesno spajanje na bazu");
mysql_query("SET CHARACTER SET cp1250", $link);

$upit=mysql_query("SELECT imeKlijent, prezimeKlijent
    FROM klijent WHERE sifKlijent=
    (SELECT MAX(sifKlijent) FROM klijent);", $link);
$rez=mysql_fetch_array($upit);
echo ('Zadnje uneseni klijent je '.$rez[0].' '.$rez[1]);
?>
```

20. Napisati PHP skriptu koja u obliku tablice ispisuje ime i prezime te naziv županije stanovanja radnika za sve radnike koji imaju iznos osnovice veći od 2500kn. Podatke potrebne za konekciju potrebno je proizvoljno zadati.

```
<?php
$host="localhost";
$user="root";
$pass="root";

$link = mysql_connect ($host, $user, $pass)
    or die ("Neuspjesno spajanje na server");
$link2=mysql_select_db ("autoradionica",$link)
    or die ("Neuspjesno spajanje na bazu");
mysql_query("SET CHARACTER SET cp1250", $link);

$upit=mysql_query("SELECT imeRadnik, prezimeRadnik, nazivZupanija
    FROM radnik JOIN mjesto ON pbrStan=pbrMjesto
    NATURAL JOIN zupanija
    WHERE iznosOsnovice>2000;", $link);
$a=mysql_num_rows($upit);
echo <<< EOT
    <table border="1" style="border-collapse:collapse;" align="center">
    <tr>
```



```

        <th> Ime: </th>
        <th> Prezime: </th>
        <th> JMBG: </th>
    </tr>
EOT;

for ($i=0;$i<$a;$i++)
{
    $row=mysql_fetch_array($upit);
    echo <<< EOT
        <tr>
            <td> {$row[0]} </td>
            <td> {$row[1]} </td>
            <td> {$row[2]} </td>
        </tr>
EOT;
}
echo ("</table>");
?>

```

21. Koristeći znanje iz MySQL, HTML, CSS i PHP tehnologija, napisati aplikaciju koja će primiti naziv kvara, te za zadani kvar ispisati naziv svih odjela te imena i prezimena radnika koji rade na zadanom kvaru.

### index.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Radnici koji popravljaju kvar</title>
</head>

<body>
    <form action="ispisi_zakvarove.php" method="post">
        Unesite naziv kvara:<br />
        <input type="text" name="kvar" />
        <input type="submit" value="dohvati" />
    </form>
</body>
</html>

```

## ispisi\_za\_kvarove.php

```
<?php

$link_server=mysql_connect("localhost", "root", "root")
    or die ("Ne mogu se spojiti na poslužitelj");
$link_baza=mysql_select_db("autoradionica", $link_server)
    or die ("Ne mogu se spojiti na bazu");
mysql_query("SET CHARACTER SET cp1250", $link_server);

$kvar=$_POST['kvar'];

/* primjeri sintakse WHERE klauzule
$dohvat1="SELECT nazivOdjel, imeRadnik, prezimeRadnik FROM kvar NATURAL
JOIN odjel NATURAL JOIN radnik WHERE nazivKvar REGEXP \"^Zamjena\"";

$dohvat1_1="SELECT nazivOdjel, imeRadnik, prezimeRadnik FROM kvar NATURAL
JOIN odjel NATURAL JOIN radnik WHERE nazivKvar REGEXP '^Zamjena'";

$dohvat2="SELECT nazivOdjel, imeRadnik, prezimeRadnik FROM kvar NATURAL
JOIN odjel NATURAL JOIN radnik WHERE nazivKvar='Zamjena prednjeg fara'";
*/

$dohvat3="SELECT nazivOdjel, imeRadnik, prezimeRadnik FROM kvar NATURAL
JOIN odjel NATURAL JOIN radnik WHERE nazivKvar='$kvar'";

$upit=mysql_query($dohvat3, $link_server) or die (mysql_error());

$n=mysql_num_rows($upit);

echo <<< ispis
    <h2>Kvar:
    $kvar</h2>
    <table border="1">
        <tr>
            <th style="width:150px;">Naziv odjela</th>
            <th style="width:150px;">Ime radnika</th>
            <th style="width:150px;">Prezime radnika</th>
        </tr>

ispis;

for($i=0;$i<$n;$i++)
{
    $row=mysql_fetch_array($upit);

echo <<< EOT
    <tr>
        <td> {$row['nazivOdjel']} </td>
        <td> {$row['imeRadnik']} </td>
        <td> {$row['prezimeRadnik']} </td>
    </tr>
```

```
EOT;  
}  
mysql_close($link_server);  
  
echo ("</table>");  
  
?>
```