

Napredne računarske aplikacije

Predavanje broj: 12

Nastavne teme:

✓PRIPREMA ZA DRUGI KOLOKVIJUM

JOIN

PODUPITI

AKCIONI UPITI (INSERT, UPDATE, DELETE)

POGLEDI

DDL

FUNKCIJE

ZADATAK 1

Prikazati imena zaposlenih koji rade na istim projektima kao Andrija.

ZADATAK 1

```
SELECT ime
FROM radnik
WHERE id_radnika IN
    (SELECT Id_radnika FROM ucesce
     WHERE Id_projekta IN
        (SELECT Id_projekta FROM ucesce
         WHERE Id_radnika IN
            (SELECT Id_radnika FROM radnik
             WHERE ime='Andrija'))))
AND ime<>'Andrija';
```

ZADATAK 1 – II način

```
SELECT CONCAT(r.ime, ' ', r.prezime) AS 'Ime i prezime'  
FROM radnik r JOIN ucesce u  
ON r.Id_radnika=u.Id_radnika  
WHERE u.Id_projekta IN  
    (SELECT u.Id_projekta  
     FROM ucesce u  
     WHERE u.Id_radnika IN  
         (SELECT r.id_radnika  
          FROM radnik r  
          WHERE r.ime='Andrija'))  
AND r.ime<>'Andrija';
```

ZADATAK 2

Prikazati imena projekata na kojima rade zaposleni čija je kvalifikacija VSS.

ZADATAK 2

```
SELECT ime_proj  
FROM projekat  
WHERE Id_projekta IN  
    (SELECT Id_projekta  
     FROM ucesce  
     WHERE Id_radnika IN  
         (SELECT Id_radnika  
          FROM radnik  
          WHERE kvalif='VSS'))
```

ZADATAK 2 – II način

```
SELECT DISTINCT p.ime_proj  
FROM projekat p JOIN ucesce u  
ON p.id_projekta=u.Id_projekta  
JOIN radnik r  
ON u.id_radnika=r.id_radnika  
AND r.kvalif='VSS';
```

ZADATAK 3

Prikazati prosečna primanja zaposlenih po odeljenjima, samo za odeljenja u kojima je prosek veći od 2800.

ZADATAK 3

```
SELECT ime_od, ROUND(AVG(plata+IFNULL(premija,0)),1) AS Prosek  
FROM odeljenje,radnik  
WHERE odeljenje.Id_odeljenja=radnik.Id_odeljenja  
GROUP BY ime_od  
HAVING AVG(plata+IFNULL(premija,0))>2800
```

ZADATAK 4

Prikazati imena projekata i broj zaposlenih angažovanih na njima, za projekte na kojima je angažovano između 1 i 3 zaposlenih (uključujući i te vrednosti).

ZADATAK 4

```
SELECT ime_proj, COUNT(*) as broj  
FROM projekat, ucesce  
WHERE projekat.Id_projekta=ucesce.Id_projekta  
GROUP BY ime_proj  
HAVING COUNT(*) BETWEEN 1 AND 3;
```

ZADATAK 5

Prikazati imena zaposlenih i njihova ukupna primanja za zaposlene čija su ukupna primanja veća od prosečnih primanja u preduzeću. Rezultate urediti po imenima zaposlenih u rastućem redosledu.

ZADATAK 5

```
SELECT ime, plata+IFNULL(premija,0) as 'Primanja'  
FROM radnik  
WHERE plata+IFNULL(premija,0)>  
      (SELECT AVG(plata+IFNULL(premija,0))  
       FROM radnik)  
ORDER BY Primanja
```

ZADATAK 6

Prikazati sve podatke o odeljenjima u kojima ima zaposlenih radnika.

ZADATAK 6

```
SELECT *  
FROM ODELJENJE  
WHERE id_odeljenja IN  
    (SELECT Id_odeljenja  
     FROM radnik  
     WHERE Id_odeljenja IS NOT NULL);
```

ZADATAK 6 – II način

```
SELECT *  
FROM ODELJENJE  
WHERE EXISTS  
    (SELECT Id_odeljenja  
     FROM radnik  
     WHERE RADNIK.Id_odeljenja=ODELJENJE.Id_odeljenja);
```


ZADATAK 6 – III način

```
SELECT DISTINCT ODELJENJE.Id_odeljenja, Ime_od, mesto  
FROM ODELJENJE, RADNIK  
WHERE RADNIK.Id_odeljenja=ODELJENJE.Id_odeljenja  
AND RADNIK.Id_odeljenja IS NOT NULL
```

ZADATAK 7

Prikazati imena zaposlenih i njihova ukupna primanja za zaposlene čija su ukupna primanja manja od 2000 ili veća od 3000, ime im počinje slovom 'M' ili 'P', a zaposleni su u odeljenju Plan. Rezultate urediti po ukupnim primanjima u rastućem redosledu.

ZADATAK 7

```
SELECT ime, plata+IFNULL(premija,0) AS primanja
FROM radnik
WHERE ( (plata+IFNULL(premija,0))<2000 OR
(plata+IFNULL(premija,0))>3000)
AND (ime like 'm%' OR ime like 'p%')
AND Id_odeljenja IN
    (SELECT Id_odeljenja
     FROM odeljenje
     WHERE ime_od='Plan')
ORDER BY plata+IFNULL(premija,0);
```

ZADATAK 8

Prikazati imena zaposlenih čiji je rukovodilac Janko, a na projektima na kojima su angažovani imaju funkciju konsultanta.

ZADATAK 8

```
SELECT ime
FROM radnik
WHERE rukovodilac IN
    (SELECT Id_radnika
     FROM radnik
     WHERE ime='Janko')
AND Id_radnika IN
    (SELECT Id_radnika
     FROM ucesce
     WHERE funkcija='konsultant');
```

ZADATAK 9

Prikazati projekte na kojima radi Vanjin šef.

ZADATAK 9

```
SELECT ime_proj  
FROM projekat  
WHERE Id_projekta in  
  (SELECT Id_projekta  
   FROM ucesce WHERE Id_radnika in  
    (SELECT rukovodilac  
     FROM radnik  
     WHERE ime='Vanja'))
```

ZADATAK 10

Prikazati ime projekta i ukupan broj sati angažovanja svih zaposlenih za projekte na kojima je ukupan broj časova angažovanja između 5000 i 10000.

ZADATAK 10

```
SELECT PROJEKAT.ime_proj, SUM(UCESCE.br_sati) AS 'UKUPAN  
BROJ SATI'  
FROM PROJEKAT JOIN UCESCE  
ON PROJEKAT.id_projekta=UCESCE.Id_projekta  
GROUP BY PROJEKAT.ime_proj  
HAVING SUM(UCESCE.br_sati) BETWEEN 5000 AND 10000;
```

ZADATAK 11

Prikazati imena i poslove radnika, kao i broj i imena projekata na kojima rade uključujući i projekte na kojima ne radi ni jedan radnik.

ZADATAK 11

```
SELECT RADNIK.ime, RADNIK.posao, PROJEKAT.id_projekta,  
PROJEKAT.ime_proj  
FROM RADNIK JOIN UCESCE  
ON RADNIK.Id_radnika=UCESCE.Id_radnika  
RIGHT JOIN PROJEKAT  
ON PROJEKAT.Id_projekta=UCESCE.Id_projekta;
```

ZADATAK 12

Prikazati imena i poslove radnika, kao i broj i imena projekata na kojima rade, uključujući i radnike koji ne rade ni na jednom projektu.

ZADATAK 12

```
SELECT RADNIK.ime, RADNIK.posao, PROJEKAT.Id_projekta,  
PROJEKAT.ime_proj  
FROM RADNIK LEFT JOIN UCESCE  
ON RADNIK.Id_radnika=UCESCE.Id_radnika  
LEFT JOIN PROJEKAT  
ON PROJEKAT.Id_projekta=UCESCE.Id_projekta;
```

ZADATAK 13

Zaposlenima čija je kvalifikacija KV ukinuti angažovanje na projektima na kojima imaju funkciju izvršioca.

ZADATAK 13

```
DELETE
FROM ucesce
WHERE funkcija='izvršilac'
AND id_radnika IN (SELECT id_radnika
                   FROM radnik
                   WHERE kvalif='KV')
```

ZADATAK 14

Zaposlenima iz odeljenja 10 ukinuti angažovanje na projektu Izvoz.

ZADATAK 14

```
DELETE
FROM UCESCE
WHERE id_radnika in (SELECT id_radnika
                     FROM RADNIK
                     WHERE id_odeljenja=10)
and id_projekta in (SELECT id_projekta
                   FROM PROJEKAT
                   WHERE ime_proj='Izvoz');
```

ZADATAK 15

Zaposlenima u odeljenju 20 povećati platu 5%.

ZADATAK 15

```
UPDATE RADNIK  
SET plata=plata*0.95  
WHERE id_odeljenja=20
```

ZADATAK 16

Povećati sredstva 10% za projekte na kojima radi Pavle.

ZADATAK 16

```
UPDATE projekat
SET sredstva=sredstva*1.1
WHERE id_projekta in (SELECT id_projekta
                      FROM UCESCE
                      WHERE id_radnika in
                         (SELECT id_radnika
                          FROM radnik
                          WHERE ime ='Pavle'))
```

ZADATAK 17

Zaposlenim čija je kvalifikacija VSS i nemaju rukovodioca povećati platu za 3000 i dodeliti premiju od 4000.

ZADATAK 17

```
UPDATE radnik  
SET plata=plata+3000, premija=4000  
WHERE kvalif='VSS' AND rukovodilac IS NULL;
```

ZADATAK 18

Zaposlenim čija je kvalifikacija VKV i rukovodilac Janko Mančić povećati platu za 3450.

ZADATAK 18

```
UPDATE radnik  
SET plata=plata+3450  
WHERE kvalif='VKV' AND  
rukovodilac IN (SELECT id_radnik  
                FROM radnik  
                WHERE ime='Janko' AND prezime ='Mančić' );
```

ZADATAK 19

Dodati podatke o novom radniku Milan Milananoviću koji nema rukovodioca, ima platu od 33000 i premiju 5000. Ostali podaci nisu poznati.

ZADATAK 19

```
INSERT INTO radnik (id_radnika, ime, prezime, plata, premija)  
VALUES (7788,'Milan','Milanović', 33000, 5000)
```

ZADATAK 20

U već postojeću tabelu KVALIFIKACIJA, čiji su atributi IDBR, IME, POSAO, PLATA i PREMIJA dodati dodati podatke o radnicima koji imaju kvalifikaciju VKV ili KV.

ZADATAK 20

```
insert into kvalifikacija (idbr, ime, plata)
select id_radnika, ime, plata
from radnik
where kvalif in ('VKV','KV')
```

ZADATAK 21

U već postojeću tabelu STARI koja ima attribute ID_RADNIK, IME, PREZIME, DATUM dodati odgovarajuće podatke o zaposlenima koji su zaposleni pre 1.1.1980.

ZADATAK 21

```
insert into stari (id_radnik, ime, prezime, datum)
select id_radnika, ime, prezime, dat_zap
from radnik
where dat_zap < '1980/1/1'
```

ZADATAK 22

Napraviti backup podataka o radnicima koji rade u odeljenju 20 i nemaju premiju.

ZADATAK 22

```
CREATE TABLE radnici_b  
SELECT *  
FROM radnik  
WHERE id_odeljenja=20 and premija is null
```

ZADATAK 23

Kreirati pogled (VIEW) koji prikazuje ime zaposlenog, datum zaposlenja, platu i mesto u kome radi za zaposlene čija je kvalifikacija VKV ili ne primaju premiju. Rezultate urediti po plati u rastućem redosledu.

ZADATAK 23

```
CREATE VIEW zadatak AS
SELECT ime, dat_zap, plata, mesto
FROM odeljenje, radnik
WHERE odeljenje.id_odeljenja=radnik.id_odeljenja
AND (kvalif='VKV' OR premija is null)
ORDER BY plata;
```

ZADATAK 24

Kreirati pogled View_GodisnjaNagrada koji prikazuje ime, prezime radnika i nagradu po sledećem kriterijumu:

50% plate za 15 godina staža,

100% plate za 20 godina staža

200% plate za 30 godina staža.

ZADATAK 24

```
create view View_GodisnjaNagrada as
select ime, prezime, plata, year(curdate())-year(dat_zap) as 'Staz',
case
when year(curdate())-year(dat_zap)=15 then plata*0.5
when year(curdate())-year(dat_zap)=20 then plata
when year(curdate())-year(dat_zap)=30 then plata*2
end as 'GodisnjaNagrada'
FROM radnik;
```

```
select * from View_GodisnjaNagrada
where GodisnjaNagrada is not null;
```

ZADATAK 25

Student Jovan Jovanović koji se upisao na studijski program NRT 2016. godine i pri upisu dobio broj indeksa 14 polagao je ispit Programiranje 2 u januarskom ispitnom roku dana 02. februara 2018. kod redovnog profesora dr Mirka Milovanovića i dobio ocenu 10. U istom ispitnom roku, ispit iz Programiranja 2 položila je i studentkinja Nevena Pavlović sa brojem indeksa RT-34/16 i dobila ocenu 8.

Student Milica Mirković sa brojem indeksa NRT-76/17 polagala je ispit Baze podataka kod vanrednog profesora dr Danijele Milošević u januarskom ispitnom roku dana 31.januar 2019 i dobila ocenu 9, a ispit iz Programiranja 1, kod redovnog profesora docenta Petra Petrovića u februarском roku dana 09.februara 2018 i nije položila.

Student Miloš Pavlović sa brojem indeksa RT-66/17 prijavio je ispit iz Baza podataka kod vanrednog profesora dr Danijele Milošević u januarskom ispitnom roku dana 31.januar 2019, ali nije izašao na ispit.

Student Mirko Ranković sa brojem indeksa IS-19/15 prijavio je ispit iz Programiranja 1, kod redovnog profesora docenta Petra Petrovića u februarском roku dana 15.februara 2018 i udaljen je sa ispita zbog korišćenja mobilnog telefona.

ZADATAK 25

Kreirati relacioni model prema pravilima normalizacije i integriteta baze podataka.

Model treba da omogući da se:

- za svaki ispitni rok prikažu zapisnici sa polaganja ispita po predmetu sa podacima o predmetnom profesoru kao i ocene studenata .
- za uneti naziv ispita i datuma polaganja izračuna prosečna ocena ostvarena u ispitnom roku, ne uključujući studente koji nisu položili, nisu polagali ili su udaljeni sa ispita.

ZADATAK 25 – kreiranje relacionog modela

```
create database polaganje;  
use polaganje;
```

```
CREATE TABLE student(  
id_studenta INT NOT NULL AUTO_INCREMENT,  
br_indeksa INT NOT NULL,  
godina_upisa int NOT NULL,  
sp varchar(5) NOT NULL,  
ime varchar(20) not null,  
prezime varchar(20) not null,  
PRIMARY KEY(id_studenta),  
CONSTRAINT sp_ch check (sp IN('AVT','ASUV','ELITE','IS','NRT','NET',  
'RT')));
```


ZADATAK 25 – kreiranje relacionog modela

```
CREATE TABLE profesor (  
id_profesora INT NOT NULL AUTO_INCREMENT,  
ime varchar(20) not null,  
prezime varchar(20) not null,  
titula varchar(8) not null,  
zvanje varchar(10) not null,  
PRIMARY KEY(id_profesora),  
CONSTRAINT titula_ch check (titula IN('dr','docent','mr','magistar','doktor')));
```

ZADATAK 25 – kreiranje relacionog modela

```
CREATE TABLE ispit_rok (  
id_rok INT NOT NULL AUTO_INCREMENT,  
naziv varchar(30) not null,  
PRIMARY KEY(id_rok));
```

ZADATAK 25 – kreiranje relacionog modela

```
CREATE TABLE ispit (  
id_ispita INT NOT NULL AUTO_INCREMENT,  
naziv varchar(20) not null,  
datum_polaganja date not null,  
id_profesora int not null,  
id_rok int not null,  
PRIMARY KEY(id_ispita),  
CONSTRAINT fk_prof FOREIGN KEY(id_profesora) REFERENCES  
profesor(id_profesora),  
CONSTRAINT fk_rok FOREIGN KEY(id_rok) REFERENCES ispit_rok  
(id_rok));
```

ZADATAK 25 – kreiranje relacionog modela

```
CREATE TABLE zapisnik (  
id_ispita INT NOT NULL,  
id_studenta INT NOT NULL,  
ocena int not null,  
PRIMARY KEY(id_ispita, id_studenta),  
CONSTRAINT fk_ispit FOREIGN KEY(id_ispita) REFERENCES  
ispit(id_ispita),  
CONSTRAINT fk_student FOREIGN KEY (id_studenta) REFERENCES  
student (id_studenta),  
CONSTRAINT ocena_ch check (ocena IN (3,4,5,6,7,8,9,10))  
);
```

ZADATAK 25 – unos podataka

```
INSERT INTO student (br_indeksa, godina_upisa, sp, ime, prezime) values  
(14, 2016, 'NRT','Jovan', 'Jovanović'),  
(76, 2017, 'NRT','Milica', 'Mirković'),  
(66, 2017, 'NRT','Miloš', 'Pavlović'),  
(19, 2015, 'IS','Mirko', 'Ranković');
```

```
INSERT INTO profesor (ime, prezime, titula, zvanje) values  
( 'Mirko','Milovanović', 'dr','redovan'),  
( 'Danijela','Milošević', 'dr','vanredan'),  
( 'Petar','Petrović', 'docent','redovan');
```

ZADATAK 25 – unos podataka

```
INSERT INTO ispit_rok (naziv) values  
( 'januarski' ),  
( 'februarski' ),  
( 'junski' ),  
( 'septembarski' ),  
( 'oktobarski' );
```

```
INSERT INTO ispit (naziv,datum_polaganja,id_profesora,id_rok) values  
( 'Programiranje 2', '2018-02-09', 1, 1 ),  
( 'Baze podataka', '2019-01-31', 2, 1 ),  
( 'Programiranje 1', '2018-02-15', 3, 2 );
```

ZADATAK 25 – upit

Prikazati zapisnik sa ocenama za svakog profesora po ispitnom roku.

ZADATAK 25 – upit

```
SELECT CONCAT(p.ime,' ',p.prezime) as ' Ime i prezime
profesora', CONCAT(s.ime,' ',s.prezime) AS ' Ime i prezime
studenta', i.naziv, i.datum_polaganja, z.ocena, ir.naziv

FROM student s, profesor p, ispit i, zapisnik z, ispit_rok ir

WHERE s.id_studenta=z.id_studenta AND
i.id_ispita=z.id_ispita AND i.id_profesora=p.id_profesora
AND i.id_rok=ir.id_rok
```


ZADATAK 25 – funkcija

Kreirati funkciju `fun_prosek` koja vraća prosečnu ocenu na osnovu unetog naziva i datuma održavanja ispita.

ZADATAK 25 – funkcija

```
DELIMITER $$  
create function prosek(nazivl varchar(20), datum date)  
returns float  
begin  
declare prosecnaOc float;  
set prosecnaOc=(select AVG(ocena)  
from ispit, zapisnik  
where ispit.id_ispita=zapisnik.id_ispita and  
ocena NOT IN (3,4,5) and nazivl=naziv and datum=datum_polaganja);  
return prosecnaOc;  
END$$  
DELIMITER ;
```