



REŠENJA

SQL upiti, Pogledi, Procedure

1 SQL UPITI

1.1. Koja tri profesora imaju najviše časova. Prikazati ime, prezime nastavnika i adresu nastavnika.

```
/* Za odabir troje profesora koji imaju najviše časova se prvo moraju sabrati časovi svih predmeta koje neki nastavnik predaje (sum broj casova), nakon toga se zbir časova poređa po opadajućem poretku (DESC) i uzmu se prva tri rezultata (TOP 3). Tabele iz kojih čitaju vrednosti su nastavnik i predaje*/
```

```
SELECT TOP (3) nastavnik.ime, nastavnik.prezime, mesto.naziv,  
SUM(predaje.brojCasova)  
FROM nastavnik INNER JOIN mesto  
ON nastavnik.ptt=mesto.ptt  
INNER JOIN predaje  
ON nastavnik.idNastavnika=predaje.idNastavnika  
GROUP BY nastavnik.ime, nastavnik.prezime, mesto.naziv  
ORDER BY SUM(predaje.brojCasova) DESC
```

1.2. Prikazati broj indeksa, prezime i ime studenta, godinu studija, broj položenih ispita i prosečnu ocenu studenata 2. i 3. godine studija koji imaju prosek veći od 7 i položili su najmanje 1 ispit.

```
/* Kada neki student polozi ispit njegov id se nadje u tabeli ispit. Da bi se znalo koliko je ispita neki student položio potrebno je prebrojati unutar tabele ispit (count) prosečna ocena se računa uz pomoć agregatne funkcije AVG  
Group BY odeljak mora postojati jer se unutar SELECT dela nalaze agregatne funkcije.  
group BY sadrži sve kolone iz select-a. Pošto postoji uslov da se trebaju prikazati studenti čiji je prosek ocena veći od 7, koristi se avg, imamo HAVING odeljak (HAVING se koristi kada agregatna funkcija ucestvuje u uslovu)*/
```

```
SELECT student.brojIndeksa, student.ime, student.prezime, student.godStudija,  
COUNT(ispit.brojIndeksa) AS [broj polozenih ispita],  
AVG(ispit.ocena) AS [prosek]  
FROM student INNER JOIN ispit  
ON student.brojIndeksa=ispit.brojIndeksa  
GROUP BY student.brojIndeksa, student.ime, student.prezime, student.godStudija  
HAVING AVG(ispit.ocena)>7 AND COUNT(ispit.brojIndeksa)>=1 AND student.godStudija IN (2,3)
```

1.3. Prikazati koliko studenata je položilo neki predmet. Prikazati podatke i za one predmete koje nijedan student nije (još) položio! Sortirati izveštaj po broju studenata koji su položili predmete.

```
/*Ako je neki student položio predmet njegov id i id predmeta će se naci u tabeli ispit.
COUNT(ispit.brojIndeksa) broji koliko je studenta položilo neki predmet. Left join
prikazuje one podatke koji se nalaze u obe tabele + sve podatke iz leve */
```

```
SELECT predmet.nazivPredmeta,
COUNT (ispit.brojIndeksa) AS [BROJ STUDENATA]
FROM predmet LEFT JOIN ispit
ON predmet.idPredmeta=ispit.idPredmeta
GROUP BY predmet.idKatedre,predmet.idPredmeta,predmet.nazivPredmeta
ORDER BY COUNT (ispit.brojIndeksa) DESC
```

2 PROCEDURE

2.1. Napisati uskladištenu proceduru koja treba da izdvoji i prikaže sledeće podatke za profesora na nekoj katedri: ime, prezime, zvanje, platu. Tip podatka za naziv katedre je nvarchar(30)

```
/*pošto procedura treba da prikaze sve profesore neke katedre to znaci da će parametar
biti katedra, jer je to ono što korisnik treba da unese. Pošto profesor predaje predmet,
a predmet je sa neke katedre onda treba povezati sve te tabele. Unutar where odeljka se
postavlja uslov sa parametrom. Potrebno je da se podaci grupisu, jer se spaja sa
predmetom i ukoliko neki profesor predaje više predmeta bez group by odeljka bi se
ispisao onoliko puta koliko predmeta predaje. */
```

```
CREATE PROCEDURE [profesori]
@naziv_katedre nvarchar(30)
AS
SELECT nastavnik.ime, nastavnik.prezime, nastavnik.zvanje, nastavnik.plata, katedra.naziv
FROM nastavnik INNER JOIN predaje
ON nastavnik.idNastavnika=predaje.idNastavnika
INNER JOIN predmet
ON predaje.idPredmeta=predmet.idPredmeta
INNER JOIN katedra
ON predmet.idKatedre=katedra.idKatedre
WHERE katedra.naziv=@naziv_katedre
GROUP BY nastavnik.ime, nastavnik.prezime, nastavnik.zvanje, nastavnik.plata,
katedra.naziv
```

2.2. Napisati SQL komandu za izvršavanje procedure.

```
EXECUTE profesori [Katedra za informatiku]
```

3 POGLEDI

3.1. Formirati spisak predmeta koje predaju profesori. prikazati imena i prezimena profesora koji predaju te predmete. Sortirati spisak po nazivu predmeta u rastućem poretku. Napisati komande za kreiranje, poziv i brisanje pogleda.

```
/*Profesor predaje predmet. Unutar tabele predaje se nalaze id predmeta i id profesora i time se definiše koji profesor predaje koji predmet. Za resenje problema potrebno je povezati ove tri tabele i u select delu oznaciti sta da se prikaze. U zadatku se trazi da se podaci sortiraju, pogledi ne sadrže order by odeljak, i zbog toga, ukoliko se trebaju prikazati sortirani podaci, to se radi kod pozivanja pogleda. */
```

```
CREATE VIEW [spisak]
AS
SELECT predmet.nazivPredmeta,nastavnik.ime, nastavnik.prezime
FROM predmet INNER JOIN predaje
ON predmet.idPredmeta=predaje.idPredmeta
INNER JOIN nastavnik
ON predaje.idNastavnika=nastavnik.idNastavnika
```

```
SELECT * FROM spisak
ORDER BY nazivPredmeta ASC
```

```
DROP VIEW spisak
```