

# OSNOVE BAZA PODATAKA

---

(5)

Standardni upitni jezik - SQL

# Ulaganje upita

---

- Ulaganje upita nad jednom relacijom u upit nad drugom relacijom, ili
- Dinamička zamena rezultata jednog upita u WHERE klauzuli drugog upita.

Primer: Prikazati Prezime, Ime i Zvanje svih Nastavnika koji imaju isto zvanje kao nastavnik 'RADOVIC NIKOLA'.

```
Select S_Nas, Prezime_Ime
      From Nastavnik
      Where S_Nas In (Select Distinct S_Dir
                      From Nastavnik)
```

# Ulaganje upita

---

- Prvi upit naziva se spoljašnji upit, a drugi (koji se navodi u zagradama) naziva se unutrašnji upit.
- Prilikom izvršavanja upita sa uloženim upitom prvo se izvršava unutrašnji upit a zatim spoljašnji upit.

Zadatak: Prikazati Prezime, Ime, Zvanje i Platu nastavnika sa zvanjem 'DOCENT' koji ne predaju nijedan predmet.

## Ulaganje upita - Zadatak

---

```
Select Prezime_Ime, Zvanje, Plata
  From Nastavnik
     Where Zvanje = 'Docent' And
           S_Nas Not In (Select Distinct S_Nas
                        From Predaje)
```

### Rezultat:

Prezime_Ime	Zvanje	Plata
-----	-----	-----
SIMIC SIMA	DOCENT	11500.0000

## Ulaganje upita (Dinamička zamena)

---

Zadatak: Prikazati Šifru nastavnika, Prezime i Ime nastavnika koji imaju najmanje jednog nastavnika koji im je podređen.

```
Select S_Nas, Prezime_Ime
      From Nastavnik
      Where S_Nas In (Select Distinct S_Dir
                      From Nastavnik)
```

### Rezultat:

S_Nas	Prezime_Ime
-----	-----
1	RADOVIC NIKOLA
6	SAVIC ILIJA

# Ulaganje upita (Dinamička zamena)

---

Zadatak 1: Prikazati Šifru nastavnika i Prezime\_Ime nastavnika koji predaje predmet pod nazivom 'Osnove racunarstva'.

Zadatak 2: Prikazati Šifru nastavnika, Prezime\_Ime i platu nastavnika za nastavnike koji imaju platu vecu od prosecne plate svih nastavnika.

Zadatak 3: Prikazati Šifru nastavnika, Prezime\_Ime i platu nastavnika za nastavnike u zvanju 'R Prof' koji imaju platu vecu od prosecne plate nastavnika u zvanju 'DOCENT'.

Zadatak 4: Prikazati S\_Nas, Prezime\_Ime i Zvanje nastavnika koji imaju platu veću od prosečne plate svog zvanja.

# Zadaci za vezbu

---

Zadatak 5: Za sve predmete prikazati Šifru predmeta, Naziv predmeta i broj nastavnika koji predmet predaju.

Zadatak 6: Prikazati Šifru predmeta, Naziv predmeta i broj nastavnika koji predmet predaju za predmete koje predaje više od jednog nastavnika.

Zadatak 7: Prikazati Šifru nastavnika, Prezime\_Ime, Zvanje, i broj predmeta koji nastavnik predaje za nastavnike u zvanju 'R PROF' koji predaju više od jednog predmeta.

Zadatak 8: Prikazati Šifru predmeta, Naziv predmeta i broj nastavnika koji predmet predaju za predmete koji u nazivu imaju RS.

# Zadaci za vezbu

---

Zadatak 9: Prikazati Šifru predmeta i velikim slovima Naziv predmeta.

Zadatak 10: Za sve nastavnike prikazati šifru nastavnika i Ime nastavnika.

---



# Zadaci za vezbu

---

Zadatak 7: Prikazati Šifru predmeta i velikim slovima Naziv predmeta.

```
Select S_Nas, Substring(Prezime_Ime,  
    CharIndex(' ', Prezime_Ime)+1, Len(Prezime_Ime) -  
    CharIndex(' ', Prezime_Ime)) As Ime  
From Nastavnik
```

---

# Dinamička zamena

---

Zadatak: Prikazati Šifru i Prezime\_Ime Nastavnika koji imaju najmanje jednog nastavnika koji im je podređen.  
(Koristiti operator EXISTS.)

# Ulaganje upita (Dinamička zamena)

---

Povezivanje spoljašnjeg i unutrašnjeg upita.

Zadatak: Prikazati Šifru, Preime i ime Nastavnika koji imaju najmanje jednog nastavnika koji im je podređen. (Koristiti operator EXISTS.)

```
Select S_Nas, Prezime_Ime
      From Nastavnik A
      Where Exists (Select *
                    From Nastavnik B
                    Where B.S_Dir = A.S_Nas)
```

# Ulaganje upita (Dinamička zamena)

---

Primer: Prikazati Prezime, Ime, Zvanje i Platu nastavnika koji imaju platu veću od prosečne plate svih nastavnika.

```
Select Prezime_Ime, Zvanje,Plata
      From Nastavnik
      Where Plata > (Select Avg(Plata)
                      From Nastavnik)
```

Prezime_Ime	Zvanje	Plata
-----	-----	-----
RADOVIC NIKOLA	R PROF	14500.0000
PETRIC JANKO	R PROF	13500.0000

# **Dinamička zamena**

---

Primer: Prikazati Prezime, Ime, Zvanje i Platu nastavnika koji imaju platu veću od prosečne plate nastavnika u svom zvanju.

# Ulaganje upita (Dinamička zamena)

---

Primer: Prikazati Prezime, Ime, Zvanje i Platu nastavnika koji imaju platu veću od prosečne plate nastavnika u svom zvanju.

```
Select Prezime_Ime, Zvanje,Plata
  From Nastavnik A
    Where Plata > (Select Avg(Plata)
                  From Nastavnik B
                    Where B.Zvanje = A.Zvanje)
```

# Dinamicka zamena - Vezba

---

Primer: Prikazati Prezime, Ime, Zvanje i Platu nastavnika koji imaju platu veću od prosečne plate svih nastavnika.

Agregacione funkcije se ne mogu naći u WHERE klauzuli, već samo kao elementi selekcije i u HAVING klauzuli GROUP BY klauzule.

Primer: Prikazati Prezime\_Ime, Zvanje i Platu nastavnika koji imaju platu veću od prosečne plate nastavnika u svom zvanju.

## Ulaganje upita – (UNION, INTERSECT i MINUS)

---

- Spoljašnji i unutrašnji upit mogu biti sastavljeni od dva ili više blokova međusobno povezanih operatorima UNION, INTERSECT ili MINUS.
- Da bi navedene operacije mogle biti primenjene rezultati upita SELECT blokova koji učestvuju u operaciji moraju:
  - Imati isti broj rezultujućih kolona i
  - Kolone moraju biti istog tipa.

Zadatak: Prikazati Prezime\_Ime, Zvanje i platu nastavnika sa zvanjem 'Docent' i 'R Prof', koristeći operator UNION.



## Ulaganje upita – (UNION, INTERSECT i MINUS)

---

```
Select Prezime_Ime, Zvanje, Plata
      From Nastavnik
      Where Zvanje = 'Docent'
Union
Select Prezime_Ime, Zvanje, Plata
      From Nastavnik
      Where Zvanje = 'R Prof'
```

- U prethodnom primeru operacija UNION nema puno smisla jer se radi o upitu nad jednom relacijom, pa se isti rezultati mogu dobiti korišćenjem logičkih operatora AND, OR i NOT.
- Primer je naveden kao ilustracija i naravno da korišćenje operatora UNION, INTERSECT ili MINUS ima smisla (i koriste se) u upitima nad različitim relacijama.

# Povezivanje više relacija

---

- U svim prethodnim primerima SELECT naredbe rezultat pretraživanja formiran je iz samo jedne relacije.
- Kako doći do rezultat pretraživanja koji nije smešten u jednoj relaciji?
- Povezivanje dve ili više relacija omogućava kombinovanje podataka sa prikazom rezultata u jednoj relaciji.

Zadatak: Pretpostavimo da želimo da znamo Prezime i Ime nastavnika koji predaje predmet sa Šifrom 1.

# Povezivanje više relacija

---

Zadatak: Prikazati Šifru predmeta i Prezime i Ime nastavnika koji predaje predmet sa Šifrom 1.

# Povezivanje više relacija

---

Primer: Prikazati Prezime\_Ime nastavnika i Šifru predmeta (S\_Pred) koji svaki nastavnik predaje.

# Povezivanje više relacija

---

- Posmatrajući naše relacije Nastavnik, Predmet i Predaje vidimo da do rezultata možemo doći u sledeća dva koraka.

1.Korak: Pretraživanje relacije Predaje dobićemo Šifru nastavnika koji predaje predmet sa Šifrom 1.

```
Select S_Pred, S_Nas
      From Predaje
      Where S_Pred = 1
```

S_Pred	S_Nas
1	2

# Povezivanje više relacija

---

2. Korak: Sada pretraživanjem relacije Nastavnik za šifru nastavnika 2 dobićemo Prezime i Ime nastavnika koji predaje predmet sa šifrom 1.

```
Select S_Nas, Prezime_Ime
      From Nastavnik
      Where S_Nas = 2
```

Rezultat:

S_Nas	Prezime_Ime
2	PETROVIC PETAR

# Povezivanje više relacija

---

- Do istog rezultata može se doći koristeći, umesto dva samo jedan upit.
- Uopšteno rečeno, da bi se izvršilo pretraživanje u kojem rezultat zavisi od sadržaja više relacija moraju se:
  - ☐ u FROM klauzuli navesti relacije koje treba pretražiti, a
  - ☐ u WHERE klauzuli obeležja i uslov za spajanje relacija.
- Pored uslova za spajanje u WHERE klauzuli mogu biti navedeni i uslovi selekcije n-torki.
- Ukoliko se izostavi uslov spajanja u WHERE klauzuli izvršava se dekartov proizvod relacija iza FROM klauzule (svaka n-torka prve spaja se sa svakom n-torkom druge relacije).

# Operacija SPOJ

---

Operacija spoja (*join*) je složena binarna operacija za koju se može reći da se izvodi u tri koraka:

- Korak – formiranje dekartovog proizvoda relacija;
- Korak – iz dekartovog proizvoda se izdvajaju  $n$ -torke koje zadovoljavaju postavljene uslove
- Korak – iz tabele dobijene u drugom koraku izdvajaju se određene kolone. Ovaj korak izvodi se samo u slučaju prirodnog spoja.



## Operacija SPOJ – PRIMER -

---

**Primer:** Neka su  $r(R)$  i  $p(P)$  relacije i neka obeležje relacije B pripada šemi relacije R, a obeležje C šemi relacije P. Operacija spoja relacija  $r$  i  $p$  na osnovu uslova  $B < C$  ima sledeći tok.

$r( A \ B)$

a 1

e 3

f 1

$p( C \ D)$

2 d

1 f

## Operacija SPOJ – PRIMER –

---

1. Korak:  $(\mathbf{r} \times \mathbf{p}) = \mathbf{r}' (A \ B \ C \ D)$

a 1 2 d

a 1 1 f

e 3 2 d

e 3 1 f

f 1 2 d

f 1 1 f

2. Korak:  $\sigma_{\mathbf{B} < \mathbf{C}}(\mathbf{r}') = \mathbf{r}'' (A \ B \ C \ D)$

a 1 2 d

f 1 2 d

# Operacija Prirodnog spoja

---

- ❑ Prirodnim spojem dve relacije spajaju se međusobno  $n$ -torke tih relacija na osnovu vrednosti obeležja koja se nalaze u obe šeme relacija.
- ❑ Neka su  $r(R)$  i  $p(P)$  relacije i neka je  $R \cup P = T$ . Prirodno spajanje relacija  $r$  i  $p$  u oznaci  $r \bowtie p$ , kao rezultat daje relaciju  $q(T)$ .
- ❑ Za svaku  $n$ -torku  $tq$  u relaciji  $q$  postoje  $n$ -torke  $tr \in r$  i  $tp \in p$  za koje važi  $tr = tq[R]$  i  $tp = tq[P]$  odnosno:

$$r \bowtie p = \{ tq \mid tq[R] = tr \wedge tr \in r \wedge tq[P] = tp \wedge tp \in p \}$$

## Operacija Prirodnog spoja

---

Operaciju prirodnog spoja možemo izraziti pomoću jednostavnih operacija:

- Dekartovog proizvoda,
- selekcije, i
- projekcije
- Neka su  $r(R)$  i  $p(P)$  relacije, neka je  $R \cap P = X$  i neka važi  $R.X \in R$  i  $P.X \in P$ . Tada važi:

$$r \bowtie p = \pi_{R \setminus X} \sigma_{R.X = P.X} (r \times p)$$

## Povezivanje više relacija

---

Potsetimo se našeg zadatka: Prikazati Prezime i Ime nastavnika koji predaje predmet sa Šifrom 1.

Rešenje s jednim upitom:

```
Select S_Pred, Predaje.S_Nas, Prezime_Ime
      From Predaje, Nastavnik
      Where Nastavnik.S_Nas = Predaje.S_Nas And
            S_Pred = 1
```

S_Pred	S_Nas	Prezime_Ime
-----	-----	-----
1	2	PETROVIC PETAR

Uslovi u WHERE klauzuli definišu uslove koji trebaju biti ispunjeni da bi se izvršilo spajanje n-torki relacije PREDAJE i NASTAVNIK.

---

# Povezivanje više relacija

---

Primer: Prikazati za svaku šifru nastavnika, odgovarajuće Prezime i Ime, šifru predmeta koji predaje i odgovarajući naziv predmeta u obliku tabele čije zaglavlje je niže prikazano.

S_Nas	Prezime_Ime	S_Pred	Naziv Predmeta
-------	-------------	--------	----------------

-----	-----	-----	-----
-------	-------	-------	-------

# Povezivanje više relacija

---

```
Select A.S_Nas, Prezime_Ime, C.S_Pred, Naziv
      From Nastavnik A, Predaje B, Predmet C
      Where A.S_Nas = B.S_Nas And B.S_Pred = C.S_Pred
```

## Rezultat:

S_Nas	Prezime_Ime	S_Pred	Naziv
-----	-----	-----	-----
1	RADOVIC NIKOLA	3	Osnove racunarstva
1	RADOVIC NIKOLA	5	Programiranje RS
2	PETROVIC PETAR	1	Informacioni sistemi
3	PEIC PETAR	5	Programiranje RS
5	ILIC JOVAN	2	Strukture i BP

# Povezivanje više relacija – SELF JOIN

---

Primer: Potrebno je za svakog nastavnika prikazati Prezime, Ime, Platu, kao i Prezime, Ime i Platu njegovog direktora. Redosled n-torki rezultata treba urediti po abecedi Prezimena i Imena nastavnika.

Prezime_Ime	Plata	Prezime_ImeDir	PlataDir
-----			
...			
...			



# Povezivanje više relacija – SELF JOIN

Primer: Potrebno je za svakog nastavnika prikazati Prezime, Ime, Platu, kao i Prezime, Ime i Platu njegovog direktora. Redosled n-torki rezultata treba urediti po abecedi Prezimenaa i Imena nastavnika.

```
Select A.Prezime_Ime, A.Plata, B.Prezime_Ime
       PrezimeImeDir, B.Plata PlataDir
From Nastavnik A, Nastavnik B
       Where A.S_Dir = B.S_Nas
Order By A.Prezime_Ime
```

Prezime_Ime	Plata	PrezimeImeDir	PlataDir
ILIC JOVAN	11500.0000	SAVIC ILIJA	12500.0000
PEIC PETAR	1500.0000	SAVIC ILIJA	12500.0000
PETROVIC PETAR	11500.0000	SAVIC ILIJA	12500.0000
SAVIC ILIJA	12500.0000	RADOVIC NIKOLA	14500.0000
SAVIC MILAN	13500.0000	RADOVIC NIKOLA	14500.0000
SIMIC SIMA	11500.0000	SAVIC ILIJA	12500.0000
TOT ANA	12500.0000	RADOVIC NIKOLA	14500.0000

# Povezivanje više relacija – SELF JOIN

---

- U prethodnom primeru rezultat je dobijen iz spoja relacije Nastavnik same sa sobom (SELF JOIN).
- Oznake A i B su upotrebljene kao dva privremena naziva za relaciju Nastavnik.
- Spoj relacija A i B je izvršen za n-torke koje zadovoljavaju uslov jednakosti obeležja S\_Dir iz prve relacije sa S\_Nas iz druge relacije i
- Projekcije po obeležjima navedenim iza SELECT klauzule.
- Klauzulom ORDER BY definisan je rastući redosled n-torki rezultata po pobeležju Prezime\_Ime iz relacije A.

## Povezivanje više relacija – **SPOLJNO SPAJANJE (OUTER JOIN)**

---

- Uočava se da se u rezultatu ne dobijaju podaci o nastavnicima koji za vrednost obeležje `S_Dir` imaju *Null* – vrednost.
- U rezultatu nema nastavnika  
RADOVIC NIKOLE  
PETRIC JANKA
- Kod njih je vrednost obeležja `S_Dir` *Null* vrednost, tako da te n-torke nisu imale odgovarajuće n-torke s kojima bi se spojile po uslovu `A.S_Dir = B.S_nas`, jer nijedna n-torka u relaciji `Nastavnik` nema *Null* – vrednost obeležja `S_Nas` koje je primarni ključ u relaciji `Nastavnik`.
- Ako se žele prikazati n-torke koje za obeležje koje učestvuje u `WHERE` uslovu spajanja imaju *Null* – vrednosti potrebno je izvršiti **SPOLJNO SPAJANJE (OUTER JOIN)**.

## Povezivanje više relacija – **SPOLJNO SPAJANJE (OUTER JOIN)**

- Spoljno spajanje dve relacije realizuje se tako da se označenoj relaciji (relaciji koja ne sadži Null – vrednosti), za potrebe spajanja doda još jedna n-torka koja sadrži Null – vrednosti svih obeležja.
- Takva dodatna n-torka spaja se sa onim n-torkama za koje se ne pronađe odgovarajuća stvarna n-torka.
- Spoljno spajanje se realizuje korišćenjem sintakse u kojoj se oslov spajanja navodi u FROM klauzuli:

[FROM {<table\_source> <join\_type> <table\_source> ON <search\_condition>}

<join\_type> ::= [ INNER | { {LEFT | RIGHT | FULL} [OUTER] } ] JOIN

## Povezivanje više relacija – **SPOLJNO SPAJANJE (OUTER JOIN)**

---

- Spoljno spajanje (OUTER JOIN) može biti:
  - LEVO (LEFT OUTER JOIN) – kada se za potrebe spajanja u desnu relaciju, za potrebe leve, dodaje n-torka sa Null – vrednostima svih obeležja.
  - DESNO (RIGHT OUTER JOIN) – kada se za potrebe spajanje u levu relaciju, za potrebne desne, dodaje n-torka sa Null – vrednostima svih obeležja.
  - POTPUNO (FULL OUTER JOIN) – kada se u obe relacije (i levu i desnu) dodaje po jedna n-torka sa Null – vrednostima svih obeležja.
- Treba voditi računa da **SPOLJNO SPAJANJE** nije komutativna operacija, odnosno za rezultat nije svejedno da li se n-torka sa Null – vrednostima svih obeležja “dodaje” u levu ili desnu relaciju.

## Povezivanje više relacija – **SPOLJNO SPAJANJE (OUTER JOIN)**

```
Select A.Prezime Ime, A.Plata, B.Prezime Ime
       PrezimeImeDir, B.Plata PlataDir
  From Nastavnik A Left Outer Join Nastavnik B
       On A.S_Dir = B.S_Nas
 Order By A.Prezime_Ime
```

Prezime_Ime	Plata	PrezimeImeDir	PlataDir
-----	-----	-----	-----
ILIC JOVAN	11500.0000	SAVIC ILIJA	12500.0000
PEIC PETAR	11500.0000	SAVIC ILIJA	12500.0000
PETRIC JANKO	13500.0000	NULL	NULL
PETROVIC PETAR	11500.0000	SAVIC ILIJA	12500.0000
RADOVIC NIKOLA	14500.0000	NULL	NULL
SAVIC ILIJA	12500.0000	RADOVIC NIKOLA	14500.0000
SAVIC MILAN	13500.0000	RADOVIC NIKOLA	14500.0000
SIMIC SIMA	11500.0000	SAVIC ILIJA	12500.0000
TOT ANA	12500.0000	RADOVIC NIKOLA	14500.0000

# Zadatak1

---

Prikazati šifru nastavnika, Prezime\_Ime, i broj prdmeta koji predaje za nastavnike koji predaju najmanje jedan predmet i imaju platu veću od prosečne plate svog zvanja.

## Zadatak1

---

Prikazati šifru nastavnika, Prezime\_Ime, i broj prdmeta koji predaje za nastavnike koji predaju najmanje jedan predmet i imaju platu veću od prosečne plate svog zvanja.

```
Select A.S_Nas, Prezime_Ime, Count(*) BrPredmeta
  From Nastavnik A, Predaje B
    Where A.S_Nas = B.S_Nas And
          Plata > (Select Avg(Plata)
                  From Nastavnik C
                  Where C.Zvanje = A.Zvanje)
 Group By A.S_Nas, Prezime_Ime
```



## Izmena sadržaja relacije – UPDATE Naredba

---

Opšti oblik naredbe:

a) UPDATE <naziv\_relacije> [<alias>]  
    SET obl1 = izraz1 [,obl2 =izraz2] . . .  
    [WHERE <uslov\_za\_izmenu n-torki>]

b) UPDATE <naziv\_relacije> [<alias>]  
    SET (OBL1 [,OBL2] . . .) = (podupit)  
    [WHERE <uslov\_za\_izmenu n-torki>]

alias – privremeni naziv relacije (sinonim, drugi naziv)

obl-i – i-to obeležje relacije kojem se menja sadržaj

izraz-i – izraz koji određuje vrednost koja se pridružuje obl-i

## Izmena sadržaja relacije – UPDATE Naredba

---

- Ovom naredbom menja se sadržaj navedenih obeležja onih n-torki koje zadovoljavaju uslov za izmenu n-torki u WHERE klauzuli.
- Ako se WHERE klauzula izostavi, izmena sadržaja vrši se u svim n-torkama relacije.
- Klauzula SET određuje obeležja koja podležu izmeni i vrednosti koje će im biti dodeljene.
- U svakoj n-torki koja zadovoljava WHERE klauzulu, obeležja levo od znaka jednakosti dobijaju vrednosti određene izrazom sa desne strane znaka jednakosti.
- Ako je u SET klauzuli naveden podupit on mora vratiti tačno jednu n-torku i broj vrednosti podupita mora odgovarati broju obeležja, a pridruživanje vrednosti je po redosledu navođenja.

## Izmena sadržaja relacije – UPDATE Naredba

---

Primer1: Za nastavnike sa zvanjem 'V Prof' upisati dodatak od 1200 dinara.

```
Update Nastavnik  
  Set Dodatak = 1200  
  Where Zvanje = 'V Prof'
```

Primer2: Za nastavnike sa zvanjem 'Docent' povećati platu 5% i ukinuti dodatak.

```
Update Nastavnik  
  Set Plata = 1.05 * Plata, Dodatak = 0  
  Where Zvanje = 'Docent'
```

## Izmena sadržaja relacije – UPDATE Naredba

---

Rezultata izvršenja prethodne naredbe može se jednostavno proveriti:

```
Select * From Nastavnik  
Where Zvanje = 'Docent'
```

S_Nas	Prezime_Ime	Zvanje	S_Dir	DatZap	Plata	Dodatak
2	PETROVIC PETAR	DOCENT	6	1982-02-01 00:00:00	12075.0000	.0000
3	PEIC PETAR	DOCENT	6	1983-03-01 00:00:00	12075.0000	.0000
4	SIMIC SIMA	DOCENT	6	1984-04-01 00:00:00	12075.0000	.0000
5	ILIC JOVAN	DOCENT	6	1985-02-01 00:00:00	12075.0000	.0000

## Izmena sadržaja relacije – UPDATE Naredba

---

Primer3: Svim Nastavnicima sa Zvanjem 'Docent' upisati dodatak u iznosu od 10% dodatka nastavnika 'RADOVIC NIKOLE':

```
Update Nastavnik
  Set Dodatak = (Select 0.1 * Dodatak
                  From Nastavnik
                  Where Prezime_Ime = 'Radovic Nikola')
Where Zvanje = 'Docent'
```

- Jednom naredbom UPDATE nije moguće izvršiti promenu u više od jedne relacije. Zbog toga se može susresti s problemom integriteta. To se rešava mehanizmom transakcija.

# Zadaci za vežbu

---

Zadatak 1. : Svim nastavnicima sa zvanjem 'Docent' povećati dodatak za 1000 dinara.

Zadatak 2.: Svim nastavnicima sa zvanjem 'V Prof' povećati platu za 20% prosečne plate nastavnika u zvanju 'Docent' .

Zadatak 3. : Ukinuti dodatak nastavnicima koji ne predaju nijedan predmet.

Zadatak 4. : Svim nastavnicima koji predaju više od jednog predmeta udvostručiti iznos dodatka.

## Brisanje n-torki relacije – DELETE Naredba

---

Opšti oblik naredbe:

```
DELETE FROM <naziv_relacije>  
      [WHERE <kriterijum_za_brisanje n-torki> ]
```

- Naredbom DELETE brišu se sve n-torke relacije koje ispunjavaju kriterijum za brisanje n-torki.
- Ako nije zadata klauzula WHERE brišu se sve n-torke relacije.

# Brisanje n-torki relacije – DELETE Naredba

---

Primer1: Izbrisati podatke o nastavniku sa šifrom 8. (Brisanje jedne n-torke.)

```
Delete From Nastavnik  
Where S_Nas = 8
```

Primer2: Izbrisati podatke o angažovanju svih nastavnika na predmetu sa šifrom 5 (brisanje više n-torki).

```
Delete From Predaje  
Where S_Pred = 5
```



# Brisanje n-torki relacije – DELETE Naredba

---

Primer: Obrisati sve n-torke relacije Predaje.

Delete From Predaje

- Naredbom se brišu sve n-torke relacije pošto nema WHERE klauzule.
- Naredbom DELETE ne ukida se definicija relacije kao DROP TABLE naredbom već se brišu sve n-torke relacije, a i dalje ostaje definicija relacije u rečniku relacione baze podataka bez ijedne n-torke.

# OSNOVE OBRADE TRANSAKCIJA

---

- **Transakcija** je vremenski urdeđeni niz nedeljivih radnji nad bazom podataka koje u celini ne remete uslove integriteta.
- Transakcija predstavlja **logičku jedinicu** rada nad bazom podataka.
- Sa aspekta definisanih uslova integriteta transakcija transformiše jedno konzistentno stanje baze podataka u drugo takođe konzistentno stanje baze podataka.
- Između ta dva konzistentna stanja dozvoljeno je da se baza podataka nađe u nekonzistentnom stanju.
- Bitno je da transakcija bude tako oformljena da iz bilo kojih razloga (otkaz hardvera ili softvera) ne ostavi trajno bazu podataka u nekonzistentnom stanju.

# OSNOVE OBRADE TRANSAKCIJA

---

- Sistemi koji podržavaju **transakcionu obradu** moraju garantovati da, ukoliko se transakcijom vršilo ažuriranje baze podataka i iz bilo kojih razloga transakcija nije normalno završila, ponište sva ažuriranja delimično izvršenih transakcija.
- Transakciona obrada odvija se pod kontrolom administratora transakcija kao dela programske podrške čije naredbe:
  - COMMIT i
  - ROLLBACKu potpunosti određuju način njegovog funkcionisanja.

# OSNOVE OBRADE TRANSAKCIJA

---

- Naredba **COMMIT** signalizira administratoru transakcija o uspešnom završetku transakcije.
- Baza podataka ponovo se nalazi u konzistentnom stanju i sva dejstva transakcije (privremena ažuriranja) mogu biti trajno preneti na bazu podataka.
- Naredba **ROLLBACK** signalizira administratoru transakcija o neuspešnom završetku transakcije.
- U tom slučaju potrebno je sva dejstva takve transakcije nad bazom podataka poništiti.
- Naredbe COMMIT i ROLLBACK se zadaju direktno od programera ili automatski od strane SUBP.

# OSNOVE OBRADJE TRANSAKCIJA

---

- Da bi se mehanizam trajnog prenošenja ažuriranja na bazu podataka ili poništavanja ažuriranja implementirao potrebno je čuvati informacije o BP pre i posle ažuriranja.
- Da bi se znao vremenski trenutak od kojeg počinje čuvanje informacija i kada prestaje, nad BP se postavljaju **tačke sinhronizacije (cleanpoint)**.
- **Tačka sinhronizacije** predstavlja graničnu tačku između dve serijske transakcije. To je tačka u kojoj se baza podataka nalazi u konzistentnom stanju.
- Naredba COMMIT signalizira uspešno izvršavanje transakcije i postavlja tačku sinhronizacije. Sva ažuriranja od prethodne tačke sinhronizacije postaju trajna.

# OSNOVE OBRADE TRANSAKCIJA

---

- Naredba ROLLBACK signalizira neuspešno izvršavanje transakcije i postavlja tačku sinhronizacije. Sva ažuriranja BP izvršena datim programom od prethodne tačke sinhronizacije se poništavaju
- Administrator transakcija mora garantovati da će posle uspešnog izvršavanja naredbe COMMIT od strane transakcije sva ažuriranja nad BP postati trajna, mada je moguće da se u sledećem trenutku dogodi otkaz računarskog sistema.
- Npr. Otkaz u trenutku posle naredbe COMMIT ali pre nego što ažuriranja budu fizički zapisana u BP.
- Tada se ažuriranja koja se u trenutku otkaza nalaze u baferima operativne memorije ne gube, jer procedura restarta računarskog sistema na osnovu žurnala (podataka o izvršenim promenama na BP) ipak izvrši trajno prenošenje ažuriranja na BP.

# RESTAURACIJA KONZISTENTNOG STANJA BP

---

- Administrator transakcija u sebi ima implementiran **protokol nedeljivog kompletiranja transakcije** kojim se sva ažuriranja BP upisuju u **log transakcije**.
- Mehanizam upisa u log transakcije može biti:
  - ☐ Upis novih vrednosti BP u log transakcije (**after image**)
  - ☐ Upis starih vrednosti BP u log transakcije (**before image**)
- Svaka tačka sinhronizacije se unosi log transakcije.
- Logovi pojedinačnih transakcija se kombinuju u jedinstven sistemski log koji nazivamo **dnevnik ažuriranja**, u kojem su pokazivačima povezani zapisi koji predstavljaju log jedne transakcije.
- Dnevnik ažuriranja omogućuje oporavak BP u kombinaciji sa sigurnosnom kopijom BP.