

# OSNOVE BAZA PODATAKA

---

(4)

STANDARDNI UPITNI JEZIK - SQL

# Standardni upitni jezik – SQL

---

SQL nije samo upitni jezik, već predstavlja kompletan jezik podataka koji sadrži jezike i za:

- definisanje podataka,
- ažuriranje,
- kontrolu konzistentnosti,
- konkurentni rad, i
- jezik za održavanje rečnika podataka.

Ugrađen je u većinu komercijalno raspoloživih SUBP, od personalnih do server računara.

---

# Standardni upitni jezik – SQL

---

Osnovne karakteristike SQL-a su:

- **Jednostavnost i jednoobraznost pri korišćenju**  
(relacije se kreiraju jednom izvršnom naredbom);
  - **Mogućnost interaktivnog i klasičnog programiranja**  
(koristeći SQL dobijaju se odgovori na na trenutno postavljene zahteve ili se SQL blokovi ugrađuju u neki viši programski jezik);
  - **Neproceduralnost** (odnosno, proceduralnost u minimalnom stepenu; SQL – je u velikoj meri neproceduralan jer se njime definiše **ŠTA** se želi dobiti, a ne kako se do rezultata dolazi).
-

# SQL naredbe

---

SQL naredbe mogu se grupisati u više grupa.  
Mi ćemo obraditi dve grupe naredbi SQL-a:

- Naredbe za definisanje podataka (Data Definition Statements – **DDL**)
  - Naredbe za manipulisanje podacima (Data Manipulation Statements - **DML**)
  - Naredbe za kontrolne (upravljačke) funkcije (Data Control Statements)
-

## SQL Naredbe za definisanje podataka:

---

- Kreiranje relacije (CREATE TABLE)
  - Kreiranje pogleda (CREATE VIEW)
  - Promena strukture relacije ( ALTER TABLE)
  - Brisanje relacije iz baze podataka (DROP TABLE)
  - Kreiranje indeksa (CREATE INDEX)
  - Brisanje indeksa iz baze podataka (DROP INDEX)
-

## SQL Naredbe za manipulisanje podacima:

---

- Dodavanje novih  $n$ -torki u relaciju (INSERT)
  - Pretraživanje relacione baze podataka (SELECT)
  - Izmena sadržaja relacije (UPDATE)
  - Brisanje  $n$ -torki relacije (DELETE)
-

# Notacija za predstavljanje sintakse SQL-a

---

- Ključne reči velikim slovima
  - Promenljive (reči) malim slovima
  - [ ] – Opcioni elementi
  - { } – Mogića pojava više vrednosti koje su u zagradi
  - < > - Složeni element – koji je dalje detaljno opisan
-

# Notacija za predstavljanje sintakse SQL-a

---

ALL – Podrazumevane vrednosti su podvučene.

| - ekskluzivno ili - [ A | B ] – A li B ali ne oba. Ako se sve izostavi podrazumeva se A.

- Konkretna SQL naredba se mogu pisati bilo velikim ili malim slovima
  - Elementi za predstavljanje sintakse se nikad ne pojavljuju u konkretnim SQL naredbama
-



# SQL – CREATE TABLE naredba

---

Naredba za definisanje tabele (šeme relacije)

Osnovni oblik sintakse

CREATE TABLE naziv\_tabele

```
(naziv_obeležja>    <tip_podatka>  
                    [<ograničenje_na_obeležje>]  
[, naziv_obeležja> <tip_podatka>  
                    [<ograničenje_na_obeležje>]]
```

. . . . .

```
[<ograničenje_na_tabelu>]  
)
```

---

# Numerički tipovi podataka

---

- Integer
    - Int
    - BigInt
    - SmallInt
  - Numeric(p,d)
  - Decimal(p,d)
  - Money
  - SmallMoney
  - Float
  - Real
-

# Tekstualni tipovi podataka

---

- Character(n) (ili Char(n))
  - Character Varying(n) (ili VARCHAR(n))
  - Character Large Object (ili Clob)
  - Text
-

# Binarni tipovi podataka

---

- Bit
  - Bit Varaying
  - Binary Varaying
  - Binary Large Object (BLOB)
  - Image
-

# Datumski tipovi podataka

---

- Date
  - Time
  - TimeStamp
  - DateTime
  - SmallDateTime
-

# Šema baze podataka

---

Na primer, neka imamo sledeću šemu baze podataka:

NASTAVNIK (S\_NAS, PREZIME\_IME, ZVANJE,  
S\_DIR, DATZAP, PLATA, DODATAK)

PREDMET( S\_PRED, NAZIV, MESTO)

PREDAJE (S\_NAS, S\_PRED, ČASOVA)

---

# SQL Script Šeme baze podataka

---

Create Table Nastavnik

(S_Nas	SmallInt Primary Key,
Prezime_Ime	Varchar(25) Not Null,
Zvanje	Char(6),
S_Dir	SmallInt,
DatZap	SmallDateTime Not Null,
Plata	Money Not Null,
Dodatak	Money)

-- Linijski komentar

Create Table Predmet

(S_Pred	SmallInt Primary Key,
Naziv	VarChar(35) Not Null,
Mesto	VarChar(25) Not Null,
Semestar	SmallInt Not Null)

/\* Blok komentar \*/

Create Table Predaje

(S_Nas	SmallInt
Foreign Key References	Nastavnik(S_Nas),
S_Pred	SmallInt
Foreign Key references	Predmet(S_Pred),
Casova	SmallInt Not Null,
Primary Key (S_Nas, S_Pred))	

---

# SQL INSERT Naredba

---

Naredba za dodavanje novih n-torki u relaciju:

```
INSERT [INTO] Naziv_Relacije
```

```
    [(Obl1, Obl2, ...)]
```

```
{VALUES (<vrednost1>, <vrednost2>, ...) |  
    <podupit>}
```

---



# INSERT Naredba – Primer:

---

Treba dodati podatke u relaciju Nastavnik za Petrovic Petra sa Sifrom 002, zvenjem DOCENT koji radi od 1. februara 1982, prima platu 11500, nema dodatak i kojem je rukovorilac Nastavnik sa sifrom 001.

**Insert Into Nastavnik Values**

```
(002, 'PETROVIC PETAR', 'DOCENT', 006, '1982-02-01',  
11500, Null)
```

---

# SELECT – SQL Naredba

---

Pretraživanje relacione baze podataka:

```
SELECT [ ALL | DISTINCT ] { * | relacija.* |  
    element_selekcije [alias], ...}  
FROM relacija [alias] [, relacija [alias] ]...  
[WHERE uslov]  
[CONNECT BY uslov [START WITH uslov]]  
[GROUP BY obeležje [, obeležje ...] [HAVING uslov]]  
[{UNION | INTERSECT | MINUS} SELECT ...]  
[ORDER BY obeležje [ASC | DESC] [,obeležje [ASC |  
    DESC]]...]  
[FOR UPDATE OF obeležje1 [ obeležje2]...[NOWAIT]];
```

---

# MS SQL Server - Management Studio

The screenshot displays the Microsoft SQL Server Management Studio interface. The top menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. Below the menu is a toolbar with various icons for file operations, query execution, and window management. The main window is divided into three panes:

- Object Explorer:** Located on the left, it shows the server hierarchy for 'DESKTOP-4HPGDHI (SQL Server 15.0, ^)'. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', '2021\_S', and 'BazaZaVezbe'. The 'BazaZaVezbe' database is selected, and its 'Tables' folder is expanded, listing 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', 'dbo.Nastavnik', 'dbo.Predaje', and 'dbo.Predmet'.
- Query Editor:** The central pane shows a SQL query in 'SQLQuery1.sql - DE...PGDHI\Bender (66))\*'. The query is: 

```
Select *  
from Nastavnik
```
- Results Pane:** Located at the bottom, it displays the results of the query in a table format. The table has 8 columns: S\_Nas, Prezime\_Ime, Zvanje, S\_Dir, DatZap, Pata, and Dodatak. The results are as follows:

	S_Nas	Prezime_Ime	Zvanje	S_Dir	DatZap	Pata	Dodatak
1	1	Radovic Nikola	R Prof	NULL	1961-11-01 00:00:00	14500.00	4000.00
2	2	Petrovic Petar	Docent	6	1982-02-01 00:00:00	11500.00	NULL
3	3	Peic Petar	Docent	6	1983-03-01 00:00:00	11500.00	NULL
4	4	Simic Sima	Docent	6	1984-04-01 00:00:00	11500.00	NULL
5	5	Ilic Jovan	Docent	6	1985-05-01 00:00:00	11500.00	NULL
6	6	Savic Ilija	V Prof	1	1975-06-01 00:00:00	12500.00	1000.00
7	7	Tot Ana	V Prof	1	1975-08-01 00:00:00	12500.00	1000.00
8	8	Savic Milan	R Prof	NULL	1960-10-01 00:00:00	13500.00	NULL
9	11	Petric Janko	R Prof	NULL	1960-10-01 00:00:00	13500.00	NULL

## Obavezni elementi SELECT Naredbe

---

```
SELECT { * | {<NazivRelacije>.* |  
        element_selekcije}}  
FROM <NazivRelacije>
```

---

# Prikaz kompletnog sadržaja relacije

---

Primer: Prikazati kompletan sadržaj  
relacija: Nastavnik, Predmet i Predaje

```
Select *  
      From Nastavnik
```

---

# Selekcija kolona relacije

---

Primer: Prikazati S\_Nas, Prezime\_Ime iz  
relacije Nastavnik

```
Select S_nas As Sifra, Prezime_Ime  
From Nastavnik
```

---

# Selekcija n-torki **WHERE** Klauzula

---

Za selekciju n-torki relacije potrebno je uključiti WHERE klauzulu. Korišćena u SELECT naredbi WHERE klauzula omogućava:

1. Selekciju određenih n-torki tabele.
  2. Selekciju n-torki koje zadovoljavaju višestruke uslove
  3. Selekciju n-torki koje zadovoljavaju bar jedan od više uslova.
  4. Selekciju n-torki koje ne zadovoljavaju određene uslove.
  5. Selekciju n-torki korišćenjem AND i OR logičkih operatora.
  6. Selekciju n-torki unutar definisanog raspona.
  7. Selekciju n-torki koje zadovoljavaju vrednosti u listi vrednosti.
  8. Selekciju n-torki koje zadovoljavaju određenu kombinaciju karaktera.
-

## U WHERE Klauzuli mogu se koristiti sledeći operatori:

---

- = - jednako
  - <> - različito
  - >;>=;<;<= - veće, veće ili jednako, manje
  - [NOT] IN - [nije] jednako nekoj vrednosti u listi
  - [NOT] BETWEEN x AND y - [nije]  $\geq x$  i  $\leq y$
  - EXISTS - istinito ako se kao rezultat dobije bar jedna n-torka
  - [NOT] LIKE - slaganje ili neslaganje sekvence karaktera
  - IS [NOT] NULL - je [nije] Null Vrednost
  - NOT - negacija logičkog rezultata
  - AND - logičko I
  - OR - logičko ILI
-



# WHERE - Klauzula

---

Prikazati sve predmete koji se predaju u 8. semestru.

```
Select *  
  From Predmet  
    Where Semestar = 8
```

S_Pred	Naziv	Mesto	Semestar
1	Informacioni sistemi	Novi Sad	8
4	Tehnike programiranja	Novi Sad	8

---

# Pretraživanje za vrednosti u listi -IN

Operator IN omogućava izbor n-torki koje sadrže vrednoost koja je jednaka **jednoj** od vrednosti navedenih u listi.

Primer: Prikazati sve predmete čija šifra je 2 ili 3.

```
Select *  
  From Predmet  
   Where S_Pred In (2, 3)
```

S_Pred	Naziv	Mesto	Semestar
2	Strukture i BP	Zrenjanin	3
3	Osnove racunarstva	Novi Sad	1

# Operator LIKE

---

Operator LIKE omogućava izbor n-torki koje imaju parcijalno definisan sadržaj određenog obeležja (određenu kombinaciju karaktera).

'%' – bilo koji niz karaktera, bilo koje dužine

'\_' – jedan bilo koji karakter

Primer: Prikazati Prezime i Ime svih nastavnika koji kao treće slovo u prezimenu imaju slovo "I".

```
Select Prezime_Ime
  From Nastavnik
   Where Prezime_Ime Like '___I%'
```

---

## Definisanje redosleda n-torki pretraživanja - ORDER BY

---

U prethodnim primerima SELECT naredbe redosled n-torki u rezultatu određen je od strane SUBP. Korišćenjem klauzule ORDER BY možemo kontrolisati redosled n-torki rezultata.

Primer: Prikazati platu, Prezime Ime i Zvanje nastavnika prema opadajućem iznosu plate.

```
Select Plata, Prezime_Ime, Zvanje  
      From Nastavnik  
      Order By Plata Desc
```

---

## Eliminisanje jednakih n-torki - *klauzula* DISTINCT

---

Ako želimo da prikazemo samo različite vrednosti koje uzima neko obeležje ili grupa obeležja koristimo klauzulu DISTINCT.

Primer: Prikazati skup vrednosti koje poprima obeležje Zvanje u relaciji Nastavnik.

```
Select Distinct Zvanje  
From Nastavnik
```

---

# Izrazi i funkcije

---

Pored prikazivanja vrednosti memorisanih u tabelama. SQL omogućava i korišćenje funkcija. Postoji više grupa funkcija. Mi ćemo obraditi:

- Agregacione funkcije
  - Matematičke funkcije
  - String funkcije (nad nizovima karaktera)
  - Datumske funkcije
-

# Agregacione Funkcije

---

To su funkcije za dobijanje sumarnih informacija nad numeričkim kolonama (obeležjima).

- Funkcija za izračunavanje srednje vrednosti
    - **AVG**(ALL | DISTINCT) **obeležje**)
  - Funkcija za izračunavanje ukupne vrednosti
    - **SUM**(ALL | DISTINCT) **obeležje**)
  - Funkcija za nalaženje minimalne vrednosti zraza.
    - **MIN**(ALL | DISTINCT) **izraz**)
  - Funkcija za nalaženje maksimalne vrednosti zraza.
    - **MAX**(ALL | DISTINCT) **izraz**)
-

# Funkcija COUNT

---

Definisana je nad obeležjem bilo kog tipa. Ima sledeći opšti oblik:

COUNT({\* | [ALL | DISTINCT] *obeležje*})

---



# Funkcija COUNT

---

Iz opšteg oblika funkcije COUNT mogu nastati sledeći oblici:

- COUNT(\*)
  - COUNT(*izraz*)
  - COUNT(DISTINCT *izraz*)
  - COUNT(ALL *izraz*)
-

# Agregacione Funkcije - Primer

---

**Primer:** Naći minimalnu, srednju i maksimalnu platu svih nastavnika, kao i ukupan broj nastavnika.

```
Select Min(Plata), Avg(Plata), Max(Plata), Count(*)  
From Nastavnik
```

---

11500.0000

12500.0000

14500.0000

9

---

## Alias – privremeni naziv elementa selekcije

---

- Može se primetiti da u prethodnom primeru rezultujuće kolone (obeležja) rezultata nemaju naziv.
- Ako se elementi selekcije žele imenovati ili im se želi promeniti naziv koristi se privremeno ime (sinonim) tako što se iza elementa selekcije navodi između dva znaka `".
- Ako se privremeni naziv sastoji od jedne reči znaci ` se ne moraju navoditi.

```
Select Avg(Plata) "Prosecna Plata", Count(*) BrNastavnika  
From Nastavnik
```

Prosecna Plata	BrNastavnika
12500.0000	9

---

# Agregacione Funkcije - Primer

---

Primer: Napisati SQL naredbu za prikaz ukupne plate i dodatka za sve nastavnike sa zvanjem DOCENT.

```
Select Sum(Plata) As 'Ukup.Plata', Sum(Dodatak) As 'Ukup. Dod.'  
From Nastavnik  
Where Zvanje = 'DOCENT'
```

Ukup.Plata

Ukup. Dod.

-----  
46000.0000

-----  
NULL

---

## Sumiranje informacija – klauzula GROUP BY

---

Upotreba GROUP BY klauzule omogućava dobijanje informacija za svaku različitu vrednost obeležja po kojem se vrši grupisanje.

Primer: Odrediti srednju vrednost plate i broj nastavnika svakog pojedinog zvanja.

Informacija o svim različitim zvanjima dobojamo upitom:

```
Select Distinct Zvanje  
From Nastavnik
```

---

## Sumiranje informacija – klauzula GROUP BY

---

Zvanje

-----

DOCENT

R PROF

V PROF

Sada možemo za svaku vrednost zvanja formirati SQL  
SELECT naredbu

```
Select Zvanje, Avg(Plata), Count(*)  
      From Nastavnik  
      Where Zvanje = 'xxxxx'
```

Menjajući u WHERE klauzuli 'xxxxx' svaku od različitih  
vrednosti obeležja Zvanje.

---

## Sumiranje informacija – klauzula GROUP BY

---

Identičan rezultat možemo dobiti samo jednom SQL naredbom (umesto četiri) sledećeg oblika:

```
Select Zvanje, Avg(Plata) , Count(*)  
      From Nastavnik  
      Group By Zvanje
```

### Rezultat:

Zvanje		
-----	-----	-----
DOCENT	11500.0000	4
R PROF	13833.3333	3
V PROF	12500.0000	2

---

## Sumiranje informacija – klauzula GROUP BY

---

Dejstvo GROUP BY klauzule je identično ponavljanju SELECT naredbe sa različitim WHERE uslovima.

- U SELECT listi (kao elementi selekcije) pored Agregacionih funkcija mogu se pojaviti samo obeležja po kojima se vrši grupisanje (obeležja koja se nalaze u GROUP BY klauzuli).
  - Grupisanje n-torki može se vršiti po više obeležja koja se navode u GROUP BY klauzuli.
-



## Uslovi pretraživanja za grupe – klauzula HAVING

---

Kada se formiraju grupe sa GROUP BY klauzulom korišćenjem HAVING klauzule definiše se selekcija grupa.

Primer: Prikazati Zvanje i Broj nastavnika u zvanju za zvanja u kojima ima više od dva nastavnika.

```
Select Zvanje, Count(*) As BrNastavnika  
      From Nastavnik  
      Group By Zvanje Having Count(*) > 2
```

Rezultat:

Zvanje	BrNastavnika
-----	-----
DOCENT	4
R PROF	3

---

## WHERE i GROUP BY klauzule

---

- ❑ Mogu se koristiti u istoj SELECT naredbi pri čemu WHERE klauzula uvek ide pre GROUP BY klauzule.
  - ❑ Sa WHERE klauzulom se prvo definiše uslov selekcije n-torki, a zatim se selektovane n-torke grupišu GROUP BY klauzulom.
  - ❑ Na kraju se eventualno izvrši selekcija formiranih grupa sa HAVING klauzulom.
-

## Izrazi

---

SQL naredbe mogu sadržati aritmetičke izraze sastavljene od:

- Naziva obeležja
- Konstanti
- Povezanih aritmetičkim operatorima (+, -, \*, /)
- Zagrada () za izmenu redosleda izračunavanja
- Poziva funkcija

Primer: Odrediti srednje godišnje primanje redovnih profesora.

```
Select Zvanje, Avg(Plata+Dodatak) * 12 As GodPlata  
    From Nastavnik  
    Where Zvanje = 'R PROF'  
Group By Zvanje
```

---

# Matematičke funkcije

---

- $\text{ABS}(\textit{broj})$  - nalazi apsolutnu vrednost od *broj*
  - $\text{POWER}(\textit{broj}, x)$  - stepenuje *broj* na *x*-ti stepen
  - $\text{ROUND}(\textit{broj}, d)$  - zaokružuje *broj* na *d* decimala
  - $\text{SQRT}(\textit{broj})$  - izračunava kvadratni koren od *broj*
  - $\text{SIGN}(\textit{broj})$  - daje  $+1$  ako je  $\textit{broj} > 0$ ;  $0$  ako je  $\textit{broj} = 0$ ,  $-1$  ako je  $\textit{broj} < 0$ .
-

# Matematičke funkcije

---

Primer: Prikazati nastavnike koji zarađuju više od 70 dinara po satu. Zaradu po satu izračunati tako što se ukupan iznos Plate i Dodatka podeli sa 176 (broj radnih sati u mesecu). U prikazu zaradu po satu zaokružiti na ceo broj.

```
Select S_Nas, Prezime_Ime, Round((Plata+Dodatak)/176,0)
As      ZaradaPoSatu
From Nastavnik
Where (Plata+Dodatak)/176 > 70
```

S_Nas	Prezime_Ime	ZaradaPoSatu
1	RADOVIC NIKOLA	18500.0000
6	SAVIC ILIJA	13500.0000
7	TOT ANA	13500.0000
8	SAVIC MILAN	15500.0000

---

## Izrazi sa Null - vrednostima

---

Proverom relacije Nastavnik može se zapitati zašto u prethodnom rezultatu nema nastavnika Petrić Janka koji uma platu 13.500,00 dinara, Dodatak je *Null* vrednost i za kojeg je zarada po satu  $13.500,00/176 > 70$

- Razlog je u tome što se obeležja sa *Null* vrednostima ne koriste pri izračunavanju izraza i funkcija.
  - Da bi se izračunavanje omogućilo koristi se NVL funkcija (kod SQL – Servera IsNull funkcija).
  - IsNull funkcija privremeno menja *Null* vrednost sa vrednošću za koju se odlučimo, obično s vrednošću koja nema uticaja na željenu operaciju, odnosno rezultat
-

## IsNull Funkcija

---

`IsNull(obeležje, broj)` – Izvršava se na sledeći način:

- Ako je vrednost obeležja Null – vrednost, `Null` - vrednost se zamenjuje sa *broj*.
- *Ako je vrednost obeležja definisana, tada se uzima stvarna vrednost obeležja.*

```
Select S_Nas, Prezime_Ime,  
       Round((Plata+IsNull(Dodatak,0))/176,0) As ZaradaPoSatu  
From Nastavnik  
Where (Plata+IsNull(Dodatak,0))/176 > 70
```

---

## Primer korišćenja IsNull funkcije

---

```
Select S_Nas, Prezime_Ime,  
       Round((Plata+IsNull(Dodatak,0))/176,0) As ZaradaPoSatu  
From Nastavnik  
Where (Plata+IsNull(Dodatak,0))/176 > 70
```

S_Nas	Prezime_Ime	ZaradaPoSatu
1	RADOVIC NIKOLA	18500.0000
6	SAVIC ILIJA	13500.0000
7	TOT ANA	13500.0000
8	SAVIC MILAN	15500.0000
9	PETRIC JANKO	13500.0000

---



# Funkcije nad nizovima karaktera – String funkcije

---

U SQL je definisan veći broj funkcija nad podacima tipa *Char*

- LEN(niz)
  - LEFT(niz, n)
  - RIGHT(niz, n)
  - LTRIM(niz)
  - RTRIM(niz)
  - LOWER(niz)
  - UPPER(niz)
  - SUBSTRING(niz, n, k)
  - Niz1 + niz2 - konkatencija
-

# String funkcije

---

Primer: Prikazati Prezime Ime nastavnika iza kojeg treba neposredno da sledi zvanje (kao jedan element selekcije). Redosled n-torki rezultata treba da bude u rastucem redosledu obeležja Prezime\_Ime.

```
Nastavnik
```

```
-----
```

```
Radovic Nikola, R Prof
```

```
... .
```

```
... .
```

---

# String funkcije

---

Primer: Prikazati Prezime Ime nastavnika iza kojeg treba neposredno da sledi zvanje (kao jedan element selekcije). Redosled n-torki rezultata treba da bude u rastucem redosledu obeležja Prezime\_Ime.

```
Select Prezime_Ime + ', ' + Zvanje Nastavnik  
From Nastavnik  
Order By Prezime_Ime
```

---

# Ulaganje upita

---

Primer: Prikazati Prezime Ime i Zvanje svih nastavnika koji imaju isto zvanje kao RADOVIĆ NIKOLA.

Primer: Prikazati prezime Ime Nastavnika koji imaju najmanje jednog nastavnika koji im je podređen

---

## Dimamička zamena

---

Primer: Prikazati prezime 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)
```

Dinamička zamena rezultata jednog upita z WHERE klauzuli drugog upita.

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

---

# Dinamička zamena

---

- Primer: Prikazati prezime Ime Nastavnika koji imaju najmanje jednog nastavnika koji im je podređen (koristeći operator EXIST).
-