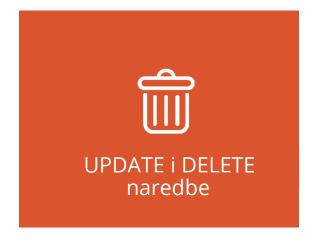
### Fakultet organizacije i informatike, Varaždin



Prof. dr. sc. Kornelije Rabuzin

# Sadržaj

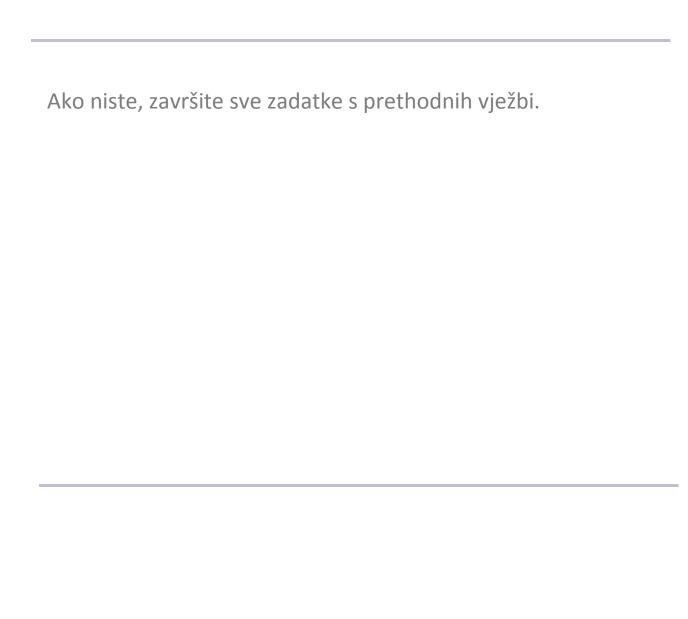




### Priprema

za vježbu U knjizi "Uvod u SQL" proučite 4. i 5. poglavlje i isprobajte

**4** sve primjere.



### Zadavanje upita

```
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
* | * | expression [ AS output_name ] [, ...]
[ FROM from_item [, ...] ]
[ WHERE condition ]
[ GROUP BY expression [, ...] ]
[ HAVING condition [, ...] ]
[ { UNION | INTERSECT | EXCEPT } [ ALL ] select ]
[ ORDER BY expression [ ASC | DESC | USING operator ] [, ] [, ...] ]
[ LIMIT { count | ALL } ]
[ OFFSET start ]
[ FOR UPDATE [ OF tablename [, ...] ] ]
```

## Primjeri

```
SELECT * FROM autor;

SELECT * FROM izdavac;

SELECT 3 * 4;

SELECT * FROM knjiga;

SELECT 3 * 4 AS produkt;

SELECT * FROM autor_knjige;

SELECT now();
```

### Upiti nad bazom podataka

Napišite upit koji će vraćati nazive knjiga iz baze:

Napišite upit koji će vraćati samo 4 retka iz tablice knjiga:

Napišite upit koji će vraćati samo 4 retka iz tablice knjiga i pritom preskočiti prva 4 retka:

Napišite upit koji će vraćati nazive i godine izdanja knjiga: **Što vraća Upper(naziv)?** 

Napišite upit koji će vraćati godine izdanja knjiga:

Napišite upit koji će vraćati godine izdanja knjiga (različite n-torke, bez ponavljanja):

Napišite upit koji će vraćati nazive i godine izdanja knjige za knjige izdane prije 1994.:

```
SELECT naziv FROM knjiga;
SELECT * FROM knjiga LIMIT 4;
SELECT * FROM knjiga LIMIT 4 OFFSET 4;
SELECT naziv, godina FROM knjiga;
SELECT godina FROM knjiga;
SELECT DISTINCT godina FROM knjiga;
SELECT naziv, godina FROM knjiga WHERE
```

godina < 1994;</pre>

### WHERE klauzula

- Služi za definiranje ograničenja nad upitima
- U njoj se mogu koristiti svi standardni operatori uspoređivanja koji postoje u PostgreSQLu, a to su:

A < B	A je manje od B	
A <= B	A je manje ili jednako od B	
A > B	A je veće od B	
A >= B	A je veće ili jednako od B	
A = B	A je jednako B	
A <> B (!=)	A je različito od B	
A BETWEEN B AND C	A je između B i C	
A LIKE B	Sličnost nizova (A je sličan B)	
A IN B	A je sadržan u B	
A IS (NOT) NULL	Je li vrijednost od A NULL ili ne	

Jednostavni se uvjeti međusobno povezuju logičkim veznicima AND, OR i NOT na standardan način

# Vježba

- 1) Napišite upit koji će vraćati godine izdanja i nazive knjiga izdanih između 1992 i 1999. godine.
- 2) Napišite upit koji će vraćati one nazive knjiga koje imaju definirani ISBN.
- 3) Napišite upit koji će vraćati one nazive knjiga koje imaju definirani ISBN i koje su izdane prije 1995.
- 4) Napišite upit koji će vraćati one nazive knjiga koje nemaju definirani ISBN i koje su izdane prije 1995.
- 5) Napišite upit koji će vraćati one nazive knjiga koji započinju slovom 'R'.



**Kod operatora** 

- 6) Napišite upit koji će vraćati one autore čije prezime započinje na 'C'.
- 7) Napišite upit koji će vraćati one autore čije prezime započinje na 'C' i sadrži točno 5 slova. više

LIKE "\_" zamjenjuje jedan znak, "%" više njih!

### **ORDER BY klauzula**

- Služi za sortiranje rezultata upita
- U toj klauzuli definiraju se atributi po kojima se vrši sortiranje (najprije se sortira po prvom definiranom atributu, pa drugom, itd.)
  - Kod silaznog sortiranja se iza naziva atributa dodaje ključna riječ DESC

#### PRIMJER:

Napišite upit koji vraća sve nazive knjiga i pritom sortirajte rezultat uzlazno po polju "naziv":

```
SELECT naziv FROM knjiga ORDER BY naziv;

ILI

SELECT naziv FROM knjiga ORDER BY 1;
```

1 označava sortiranje po prvom atributu nakon riječi SELECT

## Primjer

Upit u kojem je sortiranje izvedeno najprije silazno po godinama, a unutar toga uzlazno po nazivu:

```
SELECT naziv, godina FROM knjiga
```

ORDER BY godina DESC, naziv;

ILI

```
SELECT naziv, godina FROM knjiga ORDER BY 2 DESC, 1;
```

### Grupiranje slogova

- Za grupiranje slogova u rezultatu upita koriste se određene funkcije i GROUP BY klauzula
- U GROUP BY klauzuli navode se atributi po kojima se vrši grupiranje, a grupovne funkcije moraju biti definirane nad poljima koja se ispisuju, a nisu navedena u GROUP BY klauzuli

MBr	Prezime	Ime	 Grad	Prosj_ocjena
1	-	-	 Varaždin	4
2	-	-	 Zagreb	3
3	-	+	 Zagreb	2
4	-	-	 Čakovec	3
5	-	-	 Varaždin	4
6	-	-	 Čakovec	5
7	-	-	 Zagreb	3
8	-	-	 Varaždin	4

### **Grupovne funkcije**

- SQL grupovne funkcije:
  - COUNT vraća broj slogova
  - SUM vraća sumu vrijednosti numeričkog polja
  - MIN vraća najmanju vrijednost polja
  - MAX vraća najveću vrijednost polja
  - AVG vraća prosječnu vrijednost numeričkog polja
  - STDDEV standardna devijacija
  - VARIANCE varijanca

## Primjeri

Upit koji vraća prosječnu starost knjiga u godinama:

```
SELECT AVG(godina) AS prosjek FROM knjiga;
```

Upit koji vraća godinu izdanja najstarije knjige:

SELECT MIN(godina) FROM knjiga;



Ako se ne koristi GROUP BY, cijela tablica je jedna grupa!

Upit koji vraća broj knjiga u bazi podataka:

```
SELECT COUNT(*) AS "Broj knjiga" FROM knjiga;
```

Što vraća COUNT(isbn)?



COUNT(\*) broji sve slogove, dok COUNT(atribut) broji

samo one slogove čija je vrijednost atributa poznata, tj. nije NULL

## Primjeri

Upit koji vraća broj knjiga prema godinama izdanja:

```
SELECT godina, COUNT(*) FROM knjiga GROUP BY godina;
```

Upit koji vraća broj knjiga prema godinama izdanja sortirano silazno prema broju knjiga:

```
SELECT godina, COUNT(*) FROM knjiga GROUP BY godina
ORDER BY COUNT(*) DESC;
```

Upit koji vraća broj knjiga prema šiframa izdavača:

```
SELECT izdavac, COUNT(*) FROM knjiga GROUP BY izdavac;
```

### **HAVING** klauzula

Kako definirati upit koji će vratiti broj knjiga prema godinama izdanja, ali samo ako je određene knjige izdano barem 2 knjige?

Problem: u WHERE klauzuli je dozvoljeno koristiti samo nazive atributa, ali ne i grupovnih funkcija!!

- Klauzula HAVING omogućuje definiranje uvjeta nad grupiranim atributima, tj. nad grupovnim funkcijama
- Eliminira grupe koje ne zadovoljavaju uvjete

```
SELECT godina, COUNT(*) FROM knjiga
GROUP BY godina HAVING COUNT(*) > 1;
```

### **UPDATE** i **DELETE** naredbe

- UPDATE naredba služi za ažuriranje slogova u tablicama
- OELETE naredba služi za brisanje slogova iz tablica

```
UPDATE [ ONLY ] table
SET column = { expression | DEFAULT } [, ...][ FROM fromlist ]
[ WHERE condition ]
```

Npr. UPDATE knjiga SET godina=1984 WHERE sifra=2;

```
DELETE FROM [ ONLY ] table[ WHERE condition ];
```

Npr. DELETE FROM knjiga WHERE sifra=2;

### **Zadaci**

Implementirajte tablice iz knjige (katedre i predmeti)

Kreirajte tablicu *literatura* koja se sastoji od sljedećih atributa:

- knjiga vanjski ključ na tablicu knjiga, imenovano tablično ograničenje (ako se knjiga briše, brišu se i zapisi u tablici literatura)
- predmet vanjski ključ na tablicu predmeti, tablično ograničenje (nema ime)
- status odabir jedne vrijednosti (obavezna ili dopunska)

<ul> <li>primarni ključ tablice čine stupci knjiga i predmet (imenovano tablično ograničenje)</li> </ul>
Popunite tablice podacima (barem 20 slogova)