

OSNOVE BAZA PODATAKA

Standardni upitni jezik - SQL

Ponavljjanje

SQL izrazi i funkcije

- ❑ U SQL naredbama je moguće koristiti i razne ugrađene funkcije
 - Agregacione funkcije
AVG, SUM, MIN, MAX, COUNT, ...
 - Matematičke funkcije
ROUND, SQRT, COS, SIN, ...
 - String funkcije (nad nizovima karaktera)
LEN, SUBSTRING, UPPER, LOWER, ...
 - Datumske funkcije
DATEADD, DATEPART, ...

Ponavljjanje

Agregacione funkcije

- Funckije za dobijanje sumarnih informacija and numeričkim kolonama, kao što su:
 - srednja vrednost
AVG(ALL | DISTINCT] *obeležje*)
 - suma vrednosti
SUM(ALL | DISTINCT] *obeležje*)
 - minimalna vrednost
MIN(ALL | DISTINCT] *izraz*)
 - maksimalna vrednost
MAX(ALL | DISTINCT] *izraz*)

Ponavljanje

Agregacione funkcije

□ Funckija COUNT

- Definisana nad obeležjem bilo kod tipa

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

- Vraća broj ne-nula vrednosti obeležja, ili broj torki u slučaju upotrebe znaka *

Zadatak

- ❑ Odrediti srednju vrednost plate nastavnika svakog pojedinog zvanja.
 - Prvo proveriti koje različite vrednosti zvanja postoje u relaciji NASTAVNIK.

```
select distinct zvanje  
from nastavnik;
```

Zadatak

- Odrediti srednju vrednost plate nastavnika svakog pojedinog zvanja.
 - Potom pronaći prosek plate za svaku od njih.

```
select avg(plata)
from nastavnik
where zvanje = 'Docent';
```

```
select avg(plata)
from nastavnik
where zvanje = 'R Prof';
```

```
select avg(plata)
from nastavnik
where zvanje = 'V Prof';
```

GROUP BY klauzula

- Omogućava dobijanje agregiranih vrednosti na nivou *grupa* n-torki
 - Dobijanje informacija za *svaku različitu vrednost obeležja* po kome se vrši grupisanje

Zadatak

- ❑ Odrediti srednju vrednost plate nastavnika svakog pojedinog zvanja.
 - Koristiti GROUP BY klauzulu.

```
select zvanje, avg(plata)
from nastavnik
group by zvanje;
```


GROUP BY klauzula

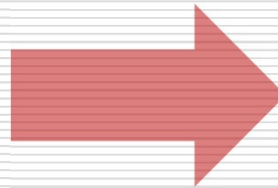
1 `select zvanje, avg(plata)
from nastavnik
group by zvanje;`

| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|----------------|--------|-------|
| 1 | Radovic Nikola | R Prof | 14500 |
| 2 | Petrovic Petar | Docent | 11500 |
| 3 | Peic Petar | Docent | 11500 |
| 4 | Simic Sima | Docent | 11500 |
| 5 | Ilic Jovan | V Prof | 11500 |
| 6 | Savic Ilija | V Prof | 12500 |

GROUP BY klauzula

```
select zvanje, avg(plata)
from nastavnik
2 group by zvanje;
```

| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|----------------|--------|-------|
| 1 | Radovic Nikola | R Prof | 14500 |
| 2 | Petrovic Petar | Docent | 11500 |
| 3 | Peic Petar | Docent | 11500 |
| 4 | Simic Sima | Docent | 11500 |
| 5 | Ilic Jovan | V Prof | 11500 |
| 6 | Savic Ilija | V Prof | 12500 |



| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|----------------|--------|-------|
| 1 | Radovic Nikola | R Prof | 14500 |

| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|----------------|--------|-------|
| 2 | Petrovic Petar | Docent | 11500 |
| 3 | Peic Petar | Docent | 11500 |
| 4 | Simic Sima | Docent | 11500 |

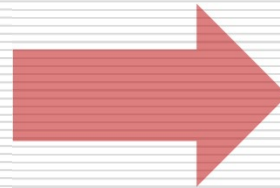
| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|-------------|--------|-------|
| 5 | Ilic Jovan | V Prof | 11500 |
| 6 | Savic Ilija | V Prof | 12500 |

GROUP BY klauzula

3 `select zvanje, avg(plata)
from nastavnik
group by zvanje;`

| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|----------------|--------|-------|
| 1 | Radovic Nikola | R Prof | 14500 |

| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|----------------|--------|-------|
| 2 | Petrovic Petar | Docent | 11500 |
| 3 | Peic Petar | Docent | 11500 |
| 4 | Simic Sima | Docent | 11500 |



| ZVANJE | avg(PLATA) |
|--------|------------|
| R Prof | 14500 |
| Docent | 11500 |
| V Prof | 12000 |

| S_NAS | PREZIME_IME | ZVANJE | PLATA |
|-------|-------------|--------|-------|
| 5 | Ilic Jovan | V Prof | 11500 |
| 6 | Savic Ilija | V Prof | 12500 |

GROUP BY klauzula

- ❑ U SELECT listi se kao elementi selekcije smeju pojaviti samo *obeležja po kojima se vrši grupisanje*, kao i *izrazi sa agregacionim funkcijama nad ostalim obeležjima*
 - Nije moguće ispisati vrednost obeležja bez agregacione funkcije, ako se to obeležje ne nalazi u GROUP BY klauzuli
- ❑ Grupisanje n-torki je moguće vršiti i po više obeležja
 - Sva obeležja po kojima se vrši grupisanje navode se iza GROUP BY klauzule, odvojena zarezom

GROUP BY klauzula

□ Ispravno:

```
select zvanje, avg(plata), count(*), sum(dodatak)
from nastavnik
group by zvanje;
```

```
select zvanje, plata, count(*)
from nastavnik
group by zvanje, plata;
```

Upit koji za postojeće kombinacije zvanja i plate ispisuje broj nastavnika koji imaju to zvanje i tu vrednost plate.

□ Neispravno:

```
select zvanje, plata, dodatak, prezime_ime
from nastavnik
group by zvanje;
```

Zadatak

- Za svakog nastavnika ispisati njegovu šifru, broj predmeta koje predaje, kao i prosečan broj časova na predmetima koje drži.

```
select s_nas, count(s_pred), avg(casova)
from predaje
group by s_nas;
```

```
select s_nas, count(s_pred), avg(cast(casova as float))
from predaje
group by s_nas;
```

HAVING klauzula

- Služi za filtriranje grupa
 - Korisi se u kombinaciji sa GROUP BY klauzulom

Zadatak

- Za svako mesto prikazati broj predmeta čija se nastava održava u tom mestu. Prikazati samo ona mesta u kojima se održava nastava iz bar 2 predmeta.

```
select mesto, count(s_pred)
from predmet
group by mesto
having count(s_pred) >= 2;
```


HAVING i WHERE

- HAVING definiše selekciju nad grupama
 - HAVING filtrira grupe
 - Za razliku od WHERE klauzule, koja definiše selekciju nad n-torkama (filtrira n-torke)
- Moguće je u okviru iste SELECT naredbe koristiti i WHERE i HAVING klauzulu
 - Prvo se definiše uslov selekcije n-torki putem WHERE klauzule, potom se filtrirane torke grupišu putem GROUP BY klauzule, i na kraju se eventualno može izvršiti selekcija grupa putem HAVING klauzule

GROUP BY i ORDER BY

- Moguće je koristiti ORDER BY u SELECT naredbi koja već poseduje GROUP BY klauzulu
 - U tom slučaju, uređivanje se vrši nad *grupama*, a ne nad n-torkama iz originalne tabele

GROUP BY i ORDER BY

- ❑ Redosled izvršavanja delova SQL SELECT naredbe

| ORDER | CLAUSE | FUNCTION |
|-------|----------|--|
| 1 | from | Choose and join tables to get base data. |
| 2 | where | Filters the base data. |
| 3 | group by | Aggregates the base data. |
| 4 | having | Filters the aggregated data. |
| 5 | select | Returns the final data. |
| 6 | order by | Sorts the final data. |
| 7 | limit | Limits the returned data to a row count. |

Zadatak

- Za svakog zvanje pronaći najmanju platu nastavnika tog zvanja. Urediti ispis po najmanjoj plati zvanja, u opadajućem redosledu.

```
select zvanje, min(plata)
from nastavnik
group by zvanje
order by min(plata) desc;
```

Zadatak

- Za svakog direktora pronaći broj podređenih nastavnika, kao i datum zaposlenja najskorije zapošljenog nastavnika koji mu je podređen. Urediti ispis po broju podređenih nastavnika, u rastućem redosledu.

```
select s_dir, count(s_nas), max(datzap)
from nastavnik
where s_dir is not null
group by s_dir
order by count(s_nas);
```

Aritmetički izrazi

- SQL naredbe mogu sadržati aritmetičke izraze sastavljene od:
 - Naziva obeležja
 - Konstanti
 - Aritmetičkih operatora (+, -, *, /)
 - Zagrada (za upravljanje redosledom izračunavanja)
 - Poziva funkcija

Zadatak

- ❑ Odrediti srednje ukupno godišnje primanje redovnih profesora.
 - Uzeti u obzir da su primanja u relaciji NASTAVNIK prikazana na mesečnom, a ne na godišnjem nivou.
 - Uzeti u obzir i dodatak pri računanju ukupnog primanja po nastavniku.

```
select s_nas,  
       prezime_ime,  
       (plata + dodatak) * 12  
from nastavnik;
```

Zadatak

- ❑ Odrediti srednje ukupno godišnje primanje redovnih profesora.
 - Uzeti u obzir da su primanja u relaciji NASTAVNIK prikazana na mesečnom, a ne na godišnjem nivou.
 - Uzeti u obzir i dodatak pri računanju ukupnog primanja po nastavniku.

```
select s_nas,  
       prezime_ime,  
       (plata + dodatak) * 12  
from nastavnik  
where zvanje = 'R Prof';
```


Zadatak

- ❑ Odrediti srednje ukupno godišnje primanje redovnih profesora.
 - Uzeti u obzir da su primanja u relaciji NASTAVNIK prikazana na mesečnom, a ne na godišnjem nivou.
 - Uzeti u obzir i dodatak pri računanju ukupnog primanja po nastavniku.

```
select s_nas,  
       prezime_ime,  
       avg((plata + dodatak) * 12)  
from nastavnik  
where zvanje = 'R Prof';
```

Aritmetički izrazi sa NULL vrednostima

- Da li je rešenje prethodnog zadatka ispravno?
 - Proveriti šta je rezultat aritmetičkih operacija nad NULL vrednostima

ISNULL funkcija

- *IsNull*(izraz, konstanta)
 - Rezultat (povratna vrednost) *IsNull* funkcije zavisi od vrednosti prosleđenog izraza:
 - Ako je vrednost izraza nedefinisana (*null* vrednost), rezultat funkcije je zadata konstanta
 - Ako je vrednost izraza definisana (nije *null* vrednost), rezultat funkcije je stvarna, neizmenjena vrednost izraza

Zadatak

- ❑ Odrediti srednje ukupno godišnje primanje redovnih profesora.
 - Uzeti u obzir da su primanja u relaciji NASTAVNIK prikazana na mesečnom, a ne na godišnjem nivou.
 - Uzeti u obzir i dodatak pri računanju ukupnog primanja po nastavniku.
 - Koristiti *IsNull* funkciju.

```
select s_nas,  
       prezime_ime,  
       avg((plata + isnull(dodatak, 0)) * 12)  
from nastavnik  
where zvanje = 'R Prof';
```

Zadatak

- Prikazati šifre, imena i prezimena svih nastavnika, kao i šifre njihovih direktora. Ako nastavnik nema direktora, za šifru direktora ispisati „Nema direktora“.
- Nije moguće u istoj koloni ispisati celobrojne i znakovne vrednosti
 - Koristiti *CAST(obeležje AS CHAR)* funkciju za konvertovanje celobrojnih vrednosti u znakovne.

```
select s_nas,  
       prezime_ime,  
       isnull(cast(s_dir as char), 'Nema direktora')  
from nastavnik;
```

Matematičke funkcije

- U SQL izrazima se mogu koristiti razne matematičke funkcije

| | |
|---------------------------------|--|
| ABS(<i>broj</i>) | nalazi apsolutnu vrednost od <i>broj</i> |
| POWER(<i>broj</i> , <i>x</i>) | stepenuje <i>broj</i> na <i>x</i> -ti stepen |
| ROUND(<i>broj</i> , <i>d</i>) | zaokružuje <i>broj</i> na <i>d</i> decimala |
| SQRT(<i>broj</i>) | izračunava kvadratni koren od <i>broj</i> |
| SIGN(<i>broj</i>) | daje +1 ako je <i>broj</i> >0, 0 ako je <i>broj</i> =0, -1 ako je <i>broj</i> <0 |
| ... | |

Zadatak

- ❑ 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 rezultata, zaradu po satu zaokružiti na 2 decimale.

```
select s_nas,  
       prezime_ime,  
       round((plata + isnull(dodatak, 0))/176, 2)  
       as zaradaPoSatu  
from nastavnik  
where (plata + isnull(dodatak, 0))/176 > 70;
```

String funkcije

- ❑ U SQL-u je definisan velik broj funkcija nad podacima znakovnog tipa (string tipa)

| | |
|--|--|
| LEN(<i>string</i>) | dužina stringa |
| LEFT(<i>string</i> , <i>n</i>) | selekcija prvih <i>n</i> karaktera stringa |
| RIGHT(<i>string</i> , <i>n</i>) | selekcija poslednjih <i>n</i> karaktera stringa |
| LTRIM(<i>string</i>) | uklanjanje <i>whitespace</i> karaktera sa početka stringa |
| RTRIM(<i>string</i>) | uklanjanje <i>whitespace</i> karaktera sa kraja stringa |
| LOWER(<i>string</i>) | prebacivanje u mala slova |
| UPPER(<i>string</i>) | prebacivanje u velika slova |
| SUBSTRING(<i>string</i> , <i>n</i> , <i>k</i>) | selekcija podniza karaktera iz stringa, dužine <i>k</i> , počevši od <i>n</i> -tog karaktera |
| <i>string1</i> + <i>string2</i> | konkatenacija stringova |
| ... | |

Zadatak

- Za svakog nastavnika prikazati njegovo zvanje velikim slovima, praćeno njegovim imenom i prezimenom (kao jedan element selekcije). Posle zvanja nastavnika koristiti zarez.

```
select upper(zvanje) + ', ' + prezime_ime  
from nastavnik;
```

Zadatak

- Za sve nastavnike koji su profesori prikazati prva 3 slova prezimena. Takođe prikazati i prvo slovo zvanja ovih nastavnika.

```
select left(prezime_ime, 3), substring(zvanje, 1, 1)
from nastavnik
where zvanje like '% Prof';
```

Zadatak

- ❑ Prikazati prezimena i imena svih nastavnika, pri čemu poslednje slovo imena treba da bude veliko.

```
select substring(prezime_ime, 1, len(prezime_ime)-1) +  
        upper(substring(prezime_ime, len(prezime_ime), 1))  
from nastavnik;
```

```
select left(prezime_ime, len(prezime_ime)-1) +  
        upper(right(prezime_ime, 1))  
from nastavnik;
```

Zadatak

□ Prikazati imena svih nastavnika.

| | |
|--|--|
| CHARINDEX(<i>char</i> , <i>string</i>) | vraća indeks zadatog karaktera <i>char</i> u stringu <i>string</i> |
|--|--|

```
select right(prezime_ime,  
            len(prezime_ime) - charindex(' ', prezime_ime))  
from nastavnik;
```

Domaći

□ Kreirati tabele:

STUDENT (S_STUD, IME, PRZ, BR_INDEKSA)

POLAZE (S_STUD, S_PRED, BR_BODOVA, OCENA)

| Obeležje | Značenje | Dozvoljene nedostajuće vrednosti |
|------------|--|--|
| S_STUD | Šifra studenta (koristi se kao identifikator studenta) | NE |
| IME | Ime studenta | NE |
| PRZ | Prezime studenta | NE |
| BR_INDEKSA | Broj indeksa studenta | NE |
| S_PRED | Šifra predmeta koji je student polagao | NE |
| BR_BODOVA | Broj bodova koji je student osvojio na predmetu | DA |
| OCENA | Ocena studenta iz predmeta | DA |

Domaći

□ Kreirati tabele:

STUDENT (S_STUD, IME, PRZ, BR_INDEKSA)

POLAZE (S_STUD, S_PRED, BR_BODOVA, OCENA)

```
create table student (  
    s_stud          smallint primary key,  
    ime            varchar(15) not null,  
    prz            varchar(20) not null,  
    br_indeksa     varchar(10) not null  
);  
  
create table polaze (  
    s_stud          smallint foreign key references student (s_stud),  
    s_pred          smallint foreign key references predmet (s_pred),  
    br_bodova       smallint,  
    ocena           smallint,  
    primary key (s_stud, s_pred)  
);
```

Domaći

- Popuniti tabele STUDENT i POLAZE podacima:

```
insert into student values (1000, 'Sava', 'Savic', 'SR1/2021');
insert into student values (2000, 'Mirko', 'Miric', 'SR2/2021');
insert into student values (3000, 'Marina', 'Maric', 'SR3/2021');
insert into student values (4000, 'Javor', 'Jelic', 'SR4/2021');
insert into student values (5000, 'Goran', 'Goric', 'SR5/2021');
insert into student values (6000, 'Jovan', 'Ilic', 'SR6/2021');
insert into student values (100, 'Marko', 'Markovic', 'SR7/2019');
```

```
insert into polaze values (1000, 2, 87, 9);
insert into polaze values (4000, 2, 40, 5);
insert into polaze values (6000, 2, 57, 6);
insert into polaze values (100, 2, 71, 8);
insert into polaze values (1000, 3, null, null);
insert into polaze values (4000, 3, 17, 5);
insert into polaze values (5000, 3, 77, 8);
insert into polaze values (1000, 1, 80, 8);
insert into polaze values (3000, 1, 97, 10);
```

Domaći

- ❑ Ispisati šifru i prosečnu ocenu svakog studenta.
- ❑ Ispisati šifre predmeta, za predmete koje je *položilo* bar 2 studenta.
- ❑ Ispisati podatke o studentima koji u imenu i u prezimenu imaju isto slovo na drugoj poziciji (npr. G**o**ran G**o**ric).
- ❑ Za svaku godinu upisa, ispisati broj studenata koji su te godine upisali studije.

- Godina upisa je sadržana u broju indeksa.

- Očekivani ispis:

| GodStud | BrStud |
|---------|--------|
| 2019 | 1 |
| 2021 | 6 |