

OSNOVE BAZA PODATAKA

Standardni upitni jezik - SQL

SELECT SQL naredba

- Koristi se za pretraživanje baze podataka
 - Putem postavljanja SQL upita

```
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]];
```

SELECT SQL naredba

□ Šablon uobičajene SELECT naredbe

```
SELECT { * | element_selekcije [alias], ...}  
[FROM relacija [alias] [, relacija [alias] ]... ]  
[WHERE uslov];
```

□ Koji delovi SQL naredbe su obavezni?

- Zavisi od proizvođača SUBP-a
 - U Microsoft SQL Server-u, samo je SELECT klauzula obavezna
-

Zadatak

- ❑ Prikazati kompletan sadržaj relacija NASTAVNIK, PREDMET i PREDAJE.

```
select *  
from nastavnik;
```

```
select *  
from predmet;
```

```
select *  
from predaje;
```

Zadatak

- Prikazati šifru nastavnika, njegovo prezime i ime, i njegovu godišnju platu.

```
select s_nas, prezime_ime, plata * 12  
from nastavnik;
```

```
select s_nas, prezime_ime, plata * 12 as "Godisnja Plata"  
from nastavnik;
```

Alijasi

- ❑ Elementima selekcije moguće je dodeliti privremene nazive (aliyasi, sinonimi)
- ❑ Alijasi se navode iza imena elementa selekcije i ključne reči **AS**, između znakova ""
 - Ključna reč **AS** se može i izostaviti
 - Ako se privremeni naziv sastoji od jedne reči, znaci "" mogu se izostaviti

```
select s_nas, prezime_ime, plata * 12 GodPlata  
from nastavnik;
```

WHERE klauzula SELECT naredbe

- ❑ Služi za selekciju n-torki
 - ❑ Omogućava:
 - Selekciju određenih n-torki tabele
 - Selekciju n-torki koje zadovoljavaju višestruke uslove
 - ❑ operator **AND**
 - Selekciju n-torki koje zadovoljavaju bar jedan od više uslova
 - ❑ operator **OR**
 - Selekciju n-torki koje ne zadovoljavaju određene uslove
 - ❑ operator **NOT** i operator nejednakosti
-

WHERE klauzula SELECT naredbe

- ❑ Služi za selekciju n-torki
 - ❑ Omogućava:
 - Selekciju n-torki unutar definisanog raspona
 - ❑ operatori [NOT] BETWEEN
 - Selekciju n-torki koje zadovoljavaju vrednosti u listi vrednosti
 - ❑ operator [NOT] IN
 - Selekciju n-torki koje zadovoljavaju određenu kombinaciju karaktera
 - ❑ operator LIKE
-

WHERE klauzula SELECT naredbe

- U WHERE klauzuli mogu se koristiti sledeći operatori:

=	jednako
<>, !=	različito
>, >=, <, <=	veće, veće ili jednako, manje, manje ili jednako
[NOT] IN	provera da li se vrednost pojavljuje u listi vrednosti
[NOT] BETWEEN x AND y	provera da li je vrednost u rasponu [x, y]
[NOT] EXISTS	istinito ako se kao rezultat dobije bar jedna torka
[NOT] LIKE	provera poklapanja sa sekvencom karaktera
IS [NOT] NULL	provera NULL vrednosti
NOT	negacija logičkog rezultata
AND	logičko I
OR	logičko ILI

Zadatak

- Prikazati sve predmete koji se predaju u 8. semestru

```
select *  
from predmet  
where semestar = 8;
```

Zadatak

- ❑ Prikazati prezimena, imena, plate i dodatke nastavnika koji primaju dodatak

```
select prezime_ime, plata, dodatak  
from nastavnik  
where dodatak is not null;
```

Operator IN

- Omogućava izbor n-torki sa vrednošću obeležja jednakom jednoj od vrednosti navedenih u listi

```
SELECT ...  
FROM ...  
WHERE obel IN ( lista vrednosti )
```

Zadatak

- Prikazati sve predmete sa šifrom 2 ili 3.

```
select *  
from predmet  
where s_pred = 2 or s_pred = 3;
```

```
select *  
from predmet  
where s_pred in (2, 3);
```

Operator BETWEEN

- Omogućava izbor n-torki sa vrednošću obeležja u zadanom intervalu

```
SELECT ...  
FROM ...  
WHERE obe1 BETWEEN vred1 AND vred2
```

Zadatak

- Prikazati sve nastavnike čija je plata između 11500 i 12000.

```
select *  
from nastavnik  
where plata >= 11500 and plata <= 12000 ;
```

```
select *  
from nastavnik  
where plata between 11500 and 12000;
```

Operator LIKE

- Omogućava izbor n-torki koje imaju parcijalno definisan sadržaj određenog obeležja (određenu kombinaciju karaktera)

```
SELECT ...  
FROM ...  
WHERE obe1 LIKE šablon
```

- U šablonu se mogu pojaviti „džoker“ znakovi
 - % - bilo koji niz karaktera, bilo koje dužine
 - _ - jedan bilo koji karakter
-

Zadatak

- Prikazati nastavnike čije prezime počinje na „P“.

```
select *  
from nastavnik  
where prezime_ime like 'P%';
```

Zadatak

- Prikazati nastavnike koji kao treće slovo prezimena imaju slovo „I”.

```
select *  
from nastavnik  
where prezime_ime like '__I%';
```

Zadatak

- Prikazati nastavnike čije ime počinje na „P“.

```
select *  
from nastavnik  
where prezime_ime like '% P%';
```

ORDER BY klauzula

- Služi za definisanje redosled n-torki u rezultatu pretrage
 - U prethodnim primerima SELECT naredbe redosled n-torki u rezultatu određen je od strane SUBP-a
 - Korišćenjem ORDER BY klauzule možemo kontrolisati redosled n-torki rezultata
-

Zadatak

- ❑ Prikazati platu, prezime, ime i zvanje nastavnika, uređeno u rastućem redosledu prema iznosu plate.

```
select plata, prezime_ime, zvanje  
from nastavnik  
order by plata;
```

```
select plata, prezime_ime, zvanje  
from nastavnik  
order by plata asc;
```

Zadatak

- ❑ Prikazati sve predmete, izuzev predmeta koji se predaju u prvoj godini. Urediti predmete po mestu u kome se održavaju. Predmete koji se održavaju na istom mestu urediti po nazivu, u rastućem redosledu.

```
select s_pred, naziv, mesto, semestar  
from predmet  
where semestar >= 3  
order by mesto asc, naziv asc;
```

Zadatak

- ❑ Prikazati sve predmete, izuzev predmeta koji se predaju u prvoj godini. Urediti predmete po mestu u kome se održavaju. Predmete koji se održavaju na istom mestu urediti po nazivu, u rastućem redosledu.

```
select s_pred          as „Sifra predmeta“,  
       naziv           as „Naziv predmeta“,  
       mesto           as „Mesto predmeta“,  
       semestar        as „Semestar predmeta“  
from predmet  
where semestar >= 3  
order by „Mesto predmeta“ asc,  
        „Naziv predmeta“  asc;
```

- ❑ Pokušati upotrebu alijasa u WHERE klauzuli.
-

ORDER BY klauzula

- ❑ 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.

DISTINCT klauzula

- Služi za eliminisanje istih n-torki iz rezultata upita
 - Koristimo je kada želimo prikazati samo različite vrednosti koje uzima neko obeležje (ili skup obeležja)
 - Koristi se neposredno iza SELECT klauzule, pre liste obeležja i izraza
-

Zadatak

- Prikazati zvanja nastavnika koja su evidentirana u sistemu.

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

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, ...
-

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*)
-

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

- ❑ Pronaći ukupan broj nastavnika, njihovu ukupnu platu, ukupan dodatak, i ukupan broj različitih zvanja svih nastavnika.

```
select count(*)           as "Broj nastavnika",  
       sum(plata)         as "Ukupna plata",  
       sum(dodatak)       as "Ukupan dodatak",  
       count(distinct zvanje) as "Broj zvanja"  
from nastavnik;
```

- ❑ Kako NULL vrednosti utiču na skupovne funkcije?
-

Zadatak

- ❑ Pronaći ukupan broj **docenata**, njihovu ukupnu platu, ukupan dodatak, kao i broj dodataka datih docentima.

```
select count(*)           as "Broj docenata",  
       sum(plata)         as "Ukupna plata" ,  
       sum(dodatak)       as "Ukupan dodatak",  
       count(dodatak)     as "Broj dodataka"  
from nastavnik  
where zvanje = 'Docent';
```

- ❑ Kako NULL vrednosti utiču na skupovne funkcije? Koja je razilka u odnosu na prethodni primer?
-

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

- ❑ Pronaći „stare studente”.
 - Uzeti u obzir da redovni studenti u indeksu imaju godinu 2021.
 - ❑ Pronaći prosečnu ocenu studenta sa šifrom 1000.
 - Koristiti funkciju *CAST* (obeležje *AS FLOAT*) za konvertovanje celobrojnih vrednosti u realne.
 - ❑ Koliko predmeta je *položio* student sa šifrom 4000?
 - ❑ Pronaći šifre studenata koji su polagali predmet ali još uvek nemaju ocenu iz predmeta.
-