

Fakultet tehničkih nauka, Novi Sad

Predmet:

Uvod u baze podataka

dr Milan Čeliković

dr Vladimir Dimitrieski

Aleksandar Jeremić

Jelena Hrnjak

Milica Vućinić

Vladimir Jovanović

Rad u učionici (1/2)

- Baze podataka:
 - Studentska korisnička šema (user schema)
 - pod nazivom **prXY**
 - **User name: prXY**
 - **Password: ftn**
 - gde je *X* broj indeksa, a *Y* godina upisa

Rad u učionici (2/2)

- Podaci potrebni za konektovanje na bazu podataka:

	MI A2-1, MI A2-2, MI A2-3	Ostale učionice	Kući
Host Name	192.168.0.102	192.168.7.204	localhost
Port Number	1522	1521	1521
Oracle SID, database name	db2016	bp1	xe

SQL (Structured Query Language)

- **SQL**
 - standardni jezik relacionih sistema za upravljanje bazama podataka
 - jezik visokog nivoa deklarativnosti
 - objedinjuje funkcije jezika za definiciju podataka, jezik za manipulaciju podacima i upitni jezik

SQL (Structured Query Language)

- **Namena i zadaci SQL-a u okviru sistema za upravljanje bazama podataka**
 - administratorima baze podataka za obavljanje poslova administracije
 - programerima za izradu aplikacija nad bazom podataka
 - krajnjim korisnicima, za postavljanje upita nad bazom podataka

SQL (Structured Query Language)

- **SQL se javlja u formama:**
 - interaktivnog jezika sistema za upravljanje bazama podataka
 - ugrađenog jezika u jezik III generacije
 - sastavnog dela jezika IV generacije

SQL (Structured Query Language)

- **Saglasno nameni i vrstama korisnika koji ga upotrebljavaju, SQL obezbeđuje realizaciju sledećih zadataka:**
 - izražavanje upita putem upitnog jezika (naredba SELECT)
 - ažuriranje baze podataka putem jezika za manipulaciju podacima (naredbe INSERT, DELETE i UPDATE)
 - realizacija implementacione šeme baze podataka i definisanje fizičke organizacije baze podataka (naredbe CREATE, DROP i ALTER)
 - automatsko održavanje rečnika podataka

SQL (Structured Query Language)

- **Saglasno nameni i vrstama korisnika koji ga upotrebljavaju, SQL obezbeđuje realizaciju sledećih zadataka:**
 - transakcijska obrada podataka (naredbe COMMIT, ROLLBACK, SAVEPOINT)
 - zaključavanje resursa (naredba LOCK TABLE)
 - zaštita podataka od neovlašćenog pristupa (naredbe GRANT, REVOKE)
 - praćenje zauzeća resursa i performansi rada sistema za upravljanje bazama podataka (naredbe AUDIT, EXPLAIN PLAN)
 - obezbeđenje proceduralnog načina obrade podataka "slog po slog" (naredbe za rad sa kursorom: OPEN, FETCH, CLOSE)

SQL (Structured Query Language)

- Sintaksa SQL-a zavisi od proizvođača sistema za upravljanje bazama podataka.

Primer

**radnik({Mbr, Ime, Prz, Sef, Plt, God,Pre}, {Mbr}),
projekat({Spr, Ruk, Nap, Nar}, {Spr}),
radproj({Spr, Mbr, Brc}, {Spr + Mbr}),**

**radnik[Sef] \subseteq radnik[Mbr],
projekat[Ruk] \subseteq radnik[Mbr],
radproj[Mbr] \subseteq radnik[Mbr],
radproj[Spr] \subseteq projekat[Spr].**

Tabela radnik

- **Mbr** - maticni broj radnika
- **Ime** - ime radnika
- **Prz** - prezime radnika
- **Sef** - maticni broj direktno nadredjenog rukovodioca - radnika
- **Plt** - mesecni iznos plate radnika
- **God** - Datum rodjenja radnika
- **Pre** – godišnja premija na platu radnika

Obeležja Mbr, Ime, Prz ne smeju imati null vrednost.
Plata ne sme biti manja od 500

Tabela radnik

```
CREATE TABLE radnik(  
    Mbr integer NOT NULL,  
    Ime varchar(20) NOT NULL,  
    Prz varchar(25) NOT NULL,  
    Sef integer,  
    Plt decimal(10, 2),  
    Pre decimal(6, 2),  
    God date NOT NULL,  
    CONSTRAINT radnik_PK PRIMARY KEY (Mbr),  
    CONSTRAINT radnik_FK FOREIGN KEY (Sef)  
        REFERENCES Radnik (Mbr),  
    CONSTRAINT radnik_CH CHECK (Plt>500)  
);
```

Tabela projekt

- **Spr** - sifra projekta
- **Ruk** - rukovodilac projekta
- **Nap** - naziv projekta
- **Nar** - narucilac projekta

Obeležja Spr i Ruk ne smeju imati null vrednost, dok obeležje Nap mora imati jedinstvenu vrednost

Tabela projekat

```
CREATE TABLE projekat
(
    Spr integer not null,
    Ruk integer not null,
    Nap varchar(30),
    Nar varchar(30),
    CONSTRAINT projekat_PK PRIMARY KEY (Spr),
    CONSTRAINT projekat_FK FOREIGN KEY (Ruk)
        REFERENCES Radnik (Mbr),
    CONSTRAINT projekat_UK UNIQUE (Nap)
);
```

Tabela radproj

- **Spr** - sifra projekta
- **Mbr** - maticni broj radnika
- **Brc** - broj casova nedeljnog angazovanja na projektu

Sva tri obeležja ne smeju da imaju null vrednost

Tabela radproj

```
CREATE TABLE radproj
(
    Spr integer NOT NULL,
    Mbr integer NOT NULL,
    Brc integer NOT NULL,
    CONSTRAINT radproj_PK PRIMARY KEY (Spr, Mbr),
    CONSTRAINT radproj_rad_FK FOREIGN KEY (Mbr)
        REFERENCES radnik(Mbr),
    CONSTRAINT radproj_prj_FK FOREIGN KEY (Spr)
        REFERENCES projekat(Spr)
);
```


Kreiranje tabele

```
CREATE TABLE [šema.]<naziv_tabele>  
(<naziv_kolone> <tip_podatka> [DEFAULT  
izraz] [, ...]  
  CONSTRAINT <naziv_ogranicenja>  
<definicija_ogranicenja> [, ...]);
```

- šema – poklapa se sa nazivom korisnika
- **DEFAULT opcija:**
 - Specificira se predefinisana vrednost za kolonu, koja se koristi ukoliko se prilikom ubacivanja podataka izostavi vrednost za tu kolonu

Naziv tabele i kolone

- mora početi slovom,
- mora biti između 1 i 30 znakova dužine,
- mora sadržati samo velika i mala slova, cifre, _, \$ i #,
- ne sme se poklapati sa nazivom nekog drugog objekta koji je kreirao isti korisnik,
- ne sme biti rezervisana reč Oracle servera.
- Nazivi nisu case sensitive.

SQL tipovi podataka

Tip podatka	Opis
– VARCHAR2(size)	niz karaktera promenljive dužine, maksimalne dužine <i>size</i> ; minimalna dužina je 1, maksimalna je 4000
– CHAR(size)	Niz karaktera fiksne dužine od <i>size</i> bajtova; default i minimalna dužina je 1, maksimalna dužina je 2000
– NUMBER(p,s)	broj ukupnog broja cifara p, od čega je s cifara iza decimalnog zareza; p može imati vrednosti od 1 do 38
– DATE	vrednosti za vreme i datum
– LONG	niz karaktera promenljive dužine do 2 GB – za kompatibilnost sa starijim verzijama Oracle-a
– CLOB	niz karaktera promenljive dužine do 4 GB
– BLOB	binarni podaci do 4 GB
– BFILE	binarni podaci smešteni u eksternom fajlu do 4 GB
– ROWID	jedinstvena adresa vrste u tabeli

Izražavanje upita i osnovna struktura naredbe SELECT

- Sve vrste upita se u SQL-u izražavaju putem naredbe SELECT. Osnovna struktura SELECT naredbe je:

```
SELECT *| <lista_obeležja>  
FROM <lista_tabela>  
[WHERE <uslov_selekcije>]
```

<lista_obeležja> sadrži obeležja nad kojima se formira rezultat upita, <lista_tabela> sadrži nazive tabela potrebne za realizaciju upita, <uslov_selekcije> izražava uslov selekcije podataka iz tabela koje su navedene iza službene reči FROM

Upiti nad jednom tabelom

- Izlistati sadržaj svih tabela.

select * from radnik;

select * from projekat;

select * from radproj;

Upiti nad jednom tabelom

- Prikazati imena i prezimena svih radnika.

```
select ime, prz  
from radnik;
```

DISTINCT

```
SELECT [DISTINCT] <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>
```

- Izlistati različita imena radnika.

```
select distinct ime from radnik;
```

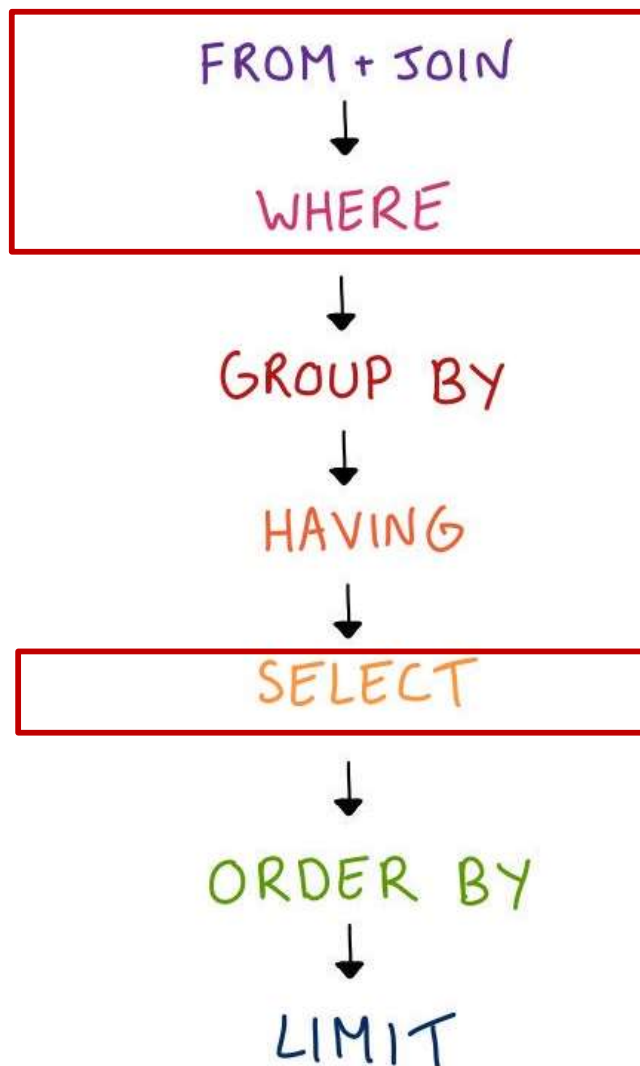
WHERE <uslov_selekcije>

- Izlistati mbr, ime i prezime radnika koji imaju platu veću od 25000.

```
select mbr, ime, prz  
from radnik  
where plt>25000;
```


Redosled izvršavanja klauzula

Redosled izvršavanja SQL klauzula



Aritmetički izrazi

- Izlistati godišnju platu svakog radnika.

```
select mbr, ime, prz, plt*12  
from radnik;
```

Null vrednost

x IS NULL – x je nula vrednost

x IS NOT NULL – x nije nula vrednost

- Izlistati mbr, ime, prz radnika koji nemaju šefa.

**SELECT mbr, ime, prz from radnik
where sef is null;**

BETWEEN

- Izlistati mbr, ime, prz radnika čija je plata između 20000 i 24000 dinara.

```
select mbr, ime, prz  
from radnik  
where plt between 20000 and 24000;
```

- Razlika između between i < >?

BETWEEN

- Izlistati ime, prz, god radnika rođenih između 1953 i 1975.

```
select ime, prz, god  
from radnik  
where god between '01-jan-1953' and '31-  
dec-1975';
```

NOT BETWEEN

- Izlistati ime, prz, god radnika koji nisu rođeni između 1953 i 1975.

select ime, prz, god

from radnik

**where god not between '01-jan-1953' and
'31-dec-1975';**

LIKE

<obeležje> LIKE <uzorak>

- Izlistati mbr, ime, prz radnika čije prezime počinje na slovo M.

```
select mbr, ime, prz  
from radnik  
where prz like 'M%';
```

NOT LIKE

- Izlistati mbr, ime, prz radnika čije ime ne počinje slovom A.

```
select mbr, ime, prz  
from radnik  
where ime not like 'A%';
```


LIKE

- Izlistati mbr, ime, prz radnika čije ime sadrži slovo a na drugoj poziciji.

```
select mbr, ime, prz  
from radnik  
where ime like '_a%';
```

LIKE

- Izlistati imena radnika koja počinju na slovo E. Imena ne bi trebalo da se ponavljaju.

```
select distinct ime  
from radnik  
where ime like 'E%';
```

LIKE

- Izlistati radnike koji u svom imenu imaju slovo E (e).

```
select mbr, ime, prz  
from radnik  
where ime like '%e%' or ime like  
'%E%';
```

IN

- Izlistati matične brojeve radnika koji rade na projektima sa šifrom 10, 20 ili 30.

```
select distinct mbr  
from radproj  
where spr in (10, 20, 30);
```

IN

- Izlistati matične brojeve radnika koji rade na projektu sa šifrom 10 ili rade 2, 4, ili 6 sati.

```
select distinct mbr  
from radproj  
where brc in (2, 4, 6) or spr='10';
```

NOT IN

- Izlistati matične brojeve radnika koji se ne zovu Ana ili Sanja.

```
select mbr, ime, prz  
from radnik  
where ime not in ('Ana', 'Sanja');
```

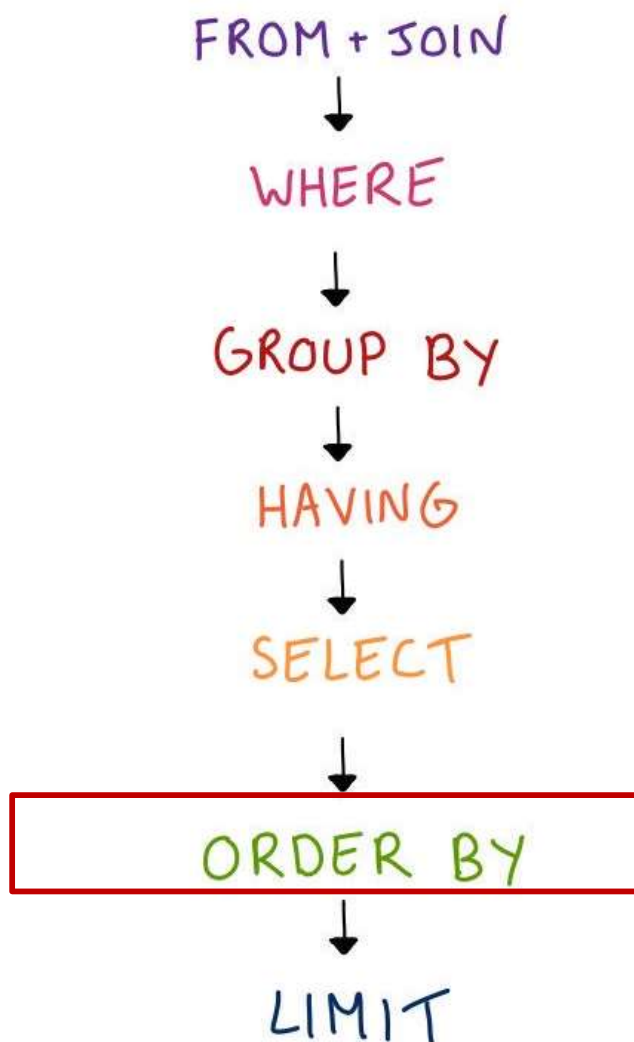
Uređivanje izlaznih rezultata

```
SELECT *| <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>  
ORDER BY <podlista_obeležja>
```

ORDER BY je uvek poslednja klauzula naredbe SELECT

Redosled izvršavanja klauzula

Redosled izvršavanja SQL klauzula



ORDER BY

- Prikazati radnike koji imaju šefa sortirano po prezimenu.

```
select mbr, ime, prz, plt  
from radnik  
where sef is not null  
order by prz asc;
```

ORDER BY

- Neki primeri upotrebe klauzule ORDER BY.

```
SELECT Mbr, Ime, Prz, Plt  
FROM Radnik ORDER BY Prz, Ime;
```

```
SELECT Mbr, Prz, Ime, Plt  
FROM Radnik ORDER BY Prz ASC, Ime ASC;
```

```
SELECT Mbr, Prz, Ime, Plt  
FROM Radnik ORDER BY Prz ASC, Ime DESC;
```

ORDER BY

- Neki primeri upotrebe klauzule ORDER BY.

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
ORDER BY 2, 3, Plt;
```

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
ORDER BY 2, 3, Plt * 1.17;
```

ORDER BY

- Prikazati matične brojeve, imena, prezimena i plate radnika, po opadajućem redosledu iznosa plate.

```
SELECT Mbr, Ime, Prz, Plt Plata  
FROM Radnik  
ORDER BY Plata DESC;
```

Uređivanje izlaznih rezultata

- Prikazati matične brojeve, **spojena** (konkatenirana) imena i prezimena radnika, kao i plate, uvećane za 17%.

```
SELECT Mbr,  
Ime || ' ' || Prz "Ime i prezime",  
Plt * 1.17 Plata  
FROM Radnik;
```

- **Concat funkcija**

Zadatak

- Prikazati radnike čije prezime sadrži ime.
Na primer Marko Marković, ili Djordje Karadjordjevic
 - Funkcije UPPER, LOWER

```
SELECT * from radnik where LOWER(prz)  
LIKE '%' || LOWER(ime) || '%'
```

ANY

$x \Theta \text{ANY} (<\text{lista_vrednosti}>)$

$\Theta \in \{<, >, <=, >=, !=, =\}$

Primer:

$x = \text{ANY} (<\text{lista_vrednosti}>)$

x je jednako makar jednoj vrednosti u
<listi_vrednosti>

ANY

- Prikazati matične brojeve radnika, imena i prezimena i platu radnika koji se zovu Pera ili Moma.

```
SELECT Mbr, Ime, Prz, Plt  
FROM Radnik  
WHERE Ime = ANY ('Pera', 'Moma');
```


ALL

$x \Theta \text{ALL} (<\text{lista_vrednosti}>)$

$\Theta \in \{<, >, <=, >=, !=, =\}$

Primer:

$x != \text{ALL} (<\text{lista_vrednosti}>)$

x je različito od svake vrednosti u
<listi_vrednosti>

ALL

- Prikazati matične brojeve radnika, imena i prezimena i platu radnika koji se ne zovu Pera ili Moma.

```
SELECT Mbr, Ime, Prz, Plt  
FROM Radnik  
WHERE Ime !=ALL ('Pera', 'Moma');
```

Upotreba skupovnih funkcija

- Prikazati matične brojeve radnika, kao i plate, uvećane za NULL vrednost.

SELECT Mbr, Plt + NULL FROM Radnik;

- Prikazati matične brojeve radnika, kao i plate, uvećane za godišnju premiju.

SELECT Mbr, Plt + Pre FROM Radnik;

Funkcija NVL(*izraz, konstanta*)

- Prikazati matične brojeve radnika, kao i plate, uvećane za godišnju premiju. Ukoliko za nekog radnika vrednost premije ne postoji, smatrati da ona iznosi 0.

```
SELECT Mbr, Plt + NVL(Pre, 0)  
FROM Radnik;
```

Funkcija count

- COUNT(*) – vraća ukupan broj selektovanih torki
- COUNT(<obeležje>) – vraća ukupan broj selektovanih torki, za koje vrednost <obeležja> nije nula vrednost
- COUNT(DISTINCT <obeležje>) – vraća ukupan broj različitih torki, za koje vrednost <obeležja> nije nula vrednost

Funkcija count

- Koliko ima radnika?

```
select count(*) from radnik;
```

- Koliko ima šefova?

```
select count(distinct sef) broj_sefova  
from radnik;
```

Funkcije max i min

- $\text{MAX}(\langle \text{obeležje} \rangle)$ – vraća maksimalnu vrednost za $\langle \text{obeležje} \rangle$, uzimajući u obzir sve selektovane torke
- $\text{MIN}(\langle \text{obeležje} \rangle)$ – vraća minimalnu vrednost za $\langle \text{obeležje} \rangle$, uzimajući u obzir sve selektovane torke

Funkcije max i min

- Prikazati minimalnu i maksimalnu platu radnika.

```
select min(plt) minimalna, max(plt) maksimalna  
from radnik;
```


Funkcija sum

- SUM(<obeležje>) – vraća zbir vrednosti datog <obeležja>, za sve selektovane torke, uključujući višestruko ponavljanje istih torke
- SUM(DISTINCT <obeležje>) – vraća zbir vrednosti datog <obeležja>, za sve različite selektovane torke

Funkcija sum

- Prikazati broj radnika i ukupnu mesečnu platu svih radnika.

```
select count(*) "Broj radnika",  
sum(plt) "Ukupna mesecna plata"  
from radnik;
```

- Šta je efekat sum funkcije kada u skupu ima null vrednosti?
 - Ignoriše null vrednosti

Funkcija avg

- `AVG(<obeležje>)` – vraća srednju vrednost datog <obeležja>, za sve selektovane torke, uključujući višestruko ponavljanje istih torki
- `AVG(DISTINCT <obeležje>)` – vraća srednju vrednosti datog <obeležja>, za sve različite selektovane torke

Funkcija avg

- Prikazati broj radnika, prosečnu platu i ukupnu godišnju platu svih radnika.
**select count(*) "Broj radnika",
avg(plt) "Prosecna plata",
12*sum(plt) "Godisnja plata"
from radnik;**
- Šta je efekat avg funkcije kada u skupu ima null vrednosti?
 - Ignoriše null vrednosti

Skupovne funkcije nad isključivo null vrednostima

- Prikazati ukupnu premiju svih radnika čiji je matični broj veći od 100.

**SELECT SUM(pre) FROM radnik WHERE
mbr >100;**

- Šta je rezultat sum, avg, max, min funkcija kada su u skupu sve null vrednosti?

Rezultat je null.

- Rezultat count funkcije u tom slučaju je vrednost 0.

Funkcija round

- ROUND(<izraz>, <broj_decimala>) – vraća zaokruženu vrednost datog <izraza> na dati <broj_decimala>

Funkcija round

- Prikazati prosečnu platu svih radnika pomnoženu sa koren iz 2 (1,41) zaokruženo na dve decimale.

```
select round(avg(plt *1.41), 2)  
from radnik;
```

Select naredba u listi tabela

```
SELECT * FROM (SELECT MBR,IME  
FROM radnik);
```


ROWNUM

Prikazati 10 radnika koji imaju najveću platu, sortiranih po plati u opadajućem redosledu

```
SELECT mbr, plt, rownum  
FROM RADNIK WHERE ROWNUM <= 10  
ORDER BY PLT DESC;
```

- **Da li je ovo ispravno?**

ROWNUM

- **Pseudokolona**
 - **ROWNUM vrednost koju torika dobija nastaje na osnovu redosleda kojim Oracle dobavlja torke**
 - **Vrednost za ROWNUM kreće od 1**

```
SELECT mbr, plt, rownum FROM  
(SELECT * FROM Radnik ORDER BY plt  
desc)  
WHERE ROWNUM <= 10;
```

Zadatak avg, round – select naveden u listi kolona

- Prikazati za svakog radnika red koji sadrži njegovu platu, prosečnu platu i apsolutnu (ABS) razliku prosečne plate i njegove plate.

	PLT	PROSECNA_PLATA	RAZLIKA
1	10000	20690.91	10690.91
2	20000	20690.91	690.91
3	15000	20690.91	5690.91
4	8000	20690.91	12690.91
5	40000	20690.91	19309.09

SELECT PLT,

**(SELECT ROUND(AVG(PLT), 2) FROM
radnik) as prosecna_plata,**

**ABS((SELECT ROUND(AVG(PLT), 2) from
RADNIK) – plt) as razlika FROM radnik;**