

# Sistemi baza podataka

---

*Dr Ivan Luković*

*Dr Slavica Kordić*

*Vladimir Dimitrieski*

# Sistem ocenjivanja

---

- Ukupno: 100 bodova

	Predispitne obaveze	Završni ispit	$\Sigma$
Teoretski deo (predavanja)	20	30	<b>50</b>
Praktični deo (vežbe)	50	0	<b>50</b>
$\Sigma$	<b>70</b>	<b>30</b>	<b>100</b>

# Praktični deo - vežbe

---

- 50 bodova
  - U toku nastave
    - 2 zadatka / vežbe – složeni oblici vežbi (Z1-Z2)
      - realizacija zadatka na vežbama
        - » proceduralna proširenja SQL
        - » serversko programiranje
      - Maks.  $2 \times 10$  bodova
    - 1 projekat (P1)
      - objedinjavanje prethodno stečenog znanja:
        - » projekat i realizacija šeme baze podataka
        - » specifikacija i realizacija ograničenja šeme BP
      - može biti priprema za završni (B.Sc.) rad
      - Maks.  $1 \times 30$  bodova

# Praktični deo - vežbe

---

- Pravila realizacije obaveza
  - Zadaci / vežbe – složeni oblici vežbi (Z1-Z2)
    - odbrana na času vežbi, koji je za to unapred planiran, pred asistentom
    - u toku semestra
    - student dobija primere i zadatke koji predstavljaju pripremu za njihovu realizaciju
    - ne postoji popravni termin
  - Projekat (P1)
    - realizuje se na nastavi i u samostalnom radu
    - odbrana: na nastavi, u toku semestra

# Rad u učionici

---

- Podaci o nalogima na računaru:
  - User name: **raX-Y**
    - X – broj indeksa, Y – godina upisa
  - Password: **Stud3nt**

# Rad u učionici

---

- Podaci potrebni za konektovanje na bazu podataka (db2014):
  - Username: **raXY**
    - X – broj indeksa, Y – godina upisa
  - Password: **ftn**
  - Host Name: **192.168.0.102**
  - Port Number: **1522**
  - Oracle SID, database name: **db2014**

# Rad u učionici

---

- SQL Developer
  - Local Disk C:\Program Files\sqldeveloper -> SQLDeveloper:
    - sqldeveloper.exe
- Materijali
  - [www.acs.uns.ac.rs](http://www.acs.uns.ac.rs)
    - sistemi baza podataka
  - nastavni materijali \ SBP \ Skriptovi

---

# **REKAPITULACIJA JEZIKA SQL**



# Kreiranje tabele

---

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

- 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

# SQL tipovi podataka

---

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

# Izmena definicije tabele

---

- **ALTER TABLE**

Alter table iskaz služi za:

- dodavanje nove kolone,
- modifikaciju postojeće kolone,
- definisanje podrazumevane vrednosti za novu kolonu,
- brisanje kolone
- dodavanje ograničenja.

# Izmena definicije tabele

---

**ALTER TABLE <naziv\_tabele>**

**ADD (<naziv\_kolone> <tip\_podatka> [DEFAULT izraz]**

**[, <naziv\_kolone> <tip\_podatka>]...);**

**ALTER TABLE <naziv\_tabele>**

**MODIFY (<naziv\_kolone> <tip\_podatka> [DEFAULT izraz] [,**

**<naziv\_kolone> <tip\_podatka>]...);**

**ALTER TABLE <naziv\_tabele>**

**DROP COLUMN (<naziv\_kolone>);**

**ALTER TABLE <naziv\_tabele>**

**ADD CONSTRAINT <naziv\_ogranicenja>**

**<definicija\_ogranicenja>;**

# Brisanje definicije tabele

---

```
DROP TABLE <naziv_tabele>;
```

# Primer - kreiranje

---

Kreirati sledeće tabele:

**radnik({Mbr, Ime, Prz, Sef, Plt, God}, {Mbr}),**

**projekat({Spr, Ruk, Nap, Nar}, {Spr}),**

**radproj({Spr, Mbr, Brc}, {Spr, Mbr}),**

**projekat[Ruk]  $\subseteq$  radnik[Mbr],**

**radproj[Mbr]  $\subseteq$  radnik[Mbr],**

**radproj[Spr]  $\subseteq$  projekat[Spr].**

Prethodno ih obrisati ako postoje u bazi podataka

# Tabela radnik

---

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

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

# Tabela projekat

---

- 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 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 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),
  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 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

---

```
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)
);
```

# Osnovni oblik SELECT naredbe

---

**SELECT** \*|{[DISTINCT] *column*|*expression* [[AS] *alias*],...}

**FROM** *table*

**WHERE** *condition*

**ORDER BY** {*column*|*expression*|*alias*|*order num* [ASC|DESC],...}

# Primeri

---

- Izlistati imena i prezimena radnika koji rade na projektu sa šifrom 30.

```
select ime, prz from radnik r, radproj rp  
where r.mbr=rp.mbr and spr=30;
```

# Primeri

---

- Prikazati radnike, čije prezime ili ime sadrži znak '-'.

```
SELECT Mbr, Ime, Prz, Plt FROM Radnik  
WHERE Ime LIKE '%-%' OR Prz LIKE '%-%'
```

- Prikazati radnike, čije prezime ili ime sadrži znak '\_'.

```
SELECT Mbr, Ime, Prz, Plt FROM Radnik  
WHERE Ime LIKE '%\_%' ESCAPE '\'  
OR Prz LIKE '%\_%' ESCAPE '\'
```

# Primeri

---

- Prikazati mbr, ime, prz, plt radnika koji zarađuju više od radnika sa matičnim brojem 40.

```
select r.mbr, r.prz, r.ime, r.plt
from radnik r, radnik r1
where r.plt>r1.plt and r1.mbr=40;
```

- Prikazati imena, prezimena i plate radnika koji zarađuju bar 1000 dinara manje od rukovodioca projekta na kom radnik radi.

```
select r1.ime, r1.prz, r1.plt, p.nap
from radnik r1, radnik r2, projekat p, radproj rp
where r1.mbr=rp.mbr and rp.spr=p.spr and
p.ruk=r2.mbr and r1.plt+1000<r2.plt;
```



# Primeri

---

- Izlistati nazive projekata na kojima se radi više od 15 časova.

```
select nap
from projekat p, radproj rp
where p.spr=rp.spr
group by p.spr, nap
having sum(brc)>15;
```

# Primeri

---

- Izlistati šifre i nazive projekata na kojima radi više od dva radnika.

```
select p.spr, p.nap  
from projekat p, radproj rp  
where rp.spr=p.spr  
group by p.spr, p.nap  
having count(mbr)>2;
```

# Primeri

---

- Izlistati u rastućem redosledu plate mbr, ime, prz i plt radnika koji imaju platu veću od prosečne.

```
select mbr, ime, prz, plt
from radnik
where plt > (select avg(plt) from radnik)
order by plt asc;
```

# Primeri

---

- Izlistati nazive i šifre projekata na kojima je prosečno angažovanje veće od prosečnog angažovanja na svim projektima.

```
select p.spr, p.nap
from projekat p, radproj rp
where rp.spr=p.spr
group by p.spr, p.nap
having avg(brc)>(select avg(brc) from radproj);
```

# Primeri

---

- Izlistati nazive i šifre projekata na kojima je prosečno angažovanje veće od prosečnog angažovanja na svakom projektu (najveće prosečno angažovanje).

```
select p.spr, p.nap
from projekat p, radproj rp
where rp.spr=p.spr
group by p.spr, p.nap
having avg(brc)>=all(select avg(brc)
from radproj group by spr);
```

# Primeri

---

- Prikazati mbr, ime, prz, plt radnika čiji je broj sati angažovanja na nekom projektu veći od prosečnog broja sati angažovanja na tom projektu.

```
select distinct r.mbr, ime, prz, plt, brc
from radnik r, radproj rp1
where r.mbr=rp1.mbr and
      rp1.brc>(select avg(brc) from radproj rp2
where rp2.spr=rp1.spr);
```

# Prirodno spajanje (NATURAL)

---

- Prikazati ime i prz radnika koji rade na projektu sa šifrom 30.

```
select ime, prz  
from radnik natural join radproj  
where spr=30;
```

**Spajanje se vrši na osnovu imena kolona.**

# Unutrašnje spajanje (INNER)

---

- Prikazati ime i prz radnika koji rade na projektu sa šifrom 30.

```
select ime, prz  
from radnik r inner join radproj rp on r.mbr=rp.mbr  
where spr=30;
```



# Spoljno spajanje (OUTER)

---

- Levo (LEFT)
- Desno (RIGHT)
- Potpuno (FULL)

# Spoljno spajanje (LEFT OUTER)

---

- Prikazati mbr, ime i prz radnika i šifre projekata na kojima rade. Prikazati, takođe, iste podatke i za radnike koji ne rade ni na jednom projektu, pri čemu za šifru projekta treba, u tom slučaju, prikazati nedostajuću vrednost.

```
select r.mbr,ime, prz, spr  
from radnik r left outer join radproj rp  
on r.mbr=rp.mbr;
```

# Spoljno spajanje (LEFT OUTER)

---

- Prikazati mbr, ime i prz svih radnika i nazive projekata kojima rukovode. Ukoliko radnik ne rukovodi ni jednim projektom ispisati: ne rukovodi projektom.

```
select r.mbr,ime, prz,
```

```
       nvl(nap, 'ne rukovodi projektom') Projekat
```

```
from radnik r left outer join projekat p on r.mbr=p.ruk;
```

# Spoljno spajanje (RIGHT OUTER)

---

- Prikazati nazive svih projekata i mbr radnika koji rade na njima. Ukoliko na projektu ne radi ni jedan radnik ispisati nulu.

```
select nvl(rp.mbr, 0) "Mbr radnika", nap  
from radproj rp right outer join projekat p  
    on rp.spr=p.spr;
```

```
select nvl(rp.mbr, 0) "Mbr radnika", nap  
from radproj rp, projekat p  
where rp.spr(+)=p.spr;
```

# Spoljno spajanje (FULL OUTER)

---

```
select nvl(rp.mbr, 0) "Mbr radnika", nap  
from radproj rp full outer join projekat p  
on rp.spr=p.spr;
```

# Primer

---

- Prikazati matične brojeve, imena i prezimena radnika, zajedno sa šiframa projekata na kojima rade. Prikazati, takođe, iste podatke i za radnike koji ne rade ni na jednom projektu, pri čemu za šifru projekta treba, u tom slučaju, prikazati nedostajuću vrednost.

# Rešenje

---

```
select r.mbr, r.prz, r.ime, rp.spr  
from radnik r, radproj rp  
where r.mbr = rp.mbr (+)
```

```
select r.mbr, r.prz, r.ime, rp.spr  
from radnik r left outer join radproj rp on r.mbr = rp.mbr
```

# Primer

---

- Prikazati za sve radnike i projekte na kojima rade Mbr, Prz, Ime, Spr i Nap. Za radnike koje ne rade ni na jednom projektu, treba prikazati Mbr, Prz i Ime, dok za vrednosti obeležja Spr i Nap treba zadati, redom, konstante 0 i "Ne postoji". Urediti izlazni rezultat saglasno rastućim vrednostima obeležja Mbr.



# Rešenje

---

```
SELECT r.Mbr, r.Prz, r.Ime, NVL(p.Spr, 0) AS Spr,  
NVL(p.Nap, 'Ne postoji') AS Nap  
FROM Radnik r, Radproj rp, Projekat p  
WHERE r.Mbr = rp.Mbr (+) AND rp.Spr = p.Spr (+)  
ORDER BY Mbr
```

```
SELECT r.Mbr, r.Prz, r.Ime, NVL(p.Spr, 0) AS Spr,  
NVL(p.Nap, 'Ne postoji') AS Nap  
FROM Radnik r LEFT OUTER JOIN Radproj rp ON  
r.Mbr = rp.Mbr LEFT OUTER JOIN Projekat p ON rp.Spr  
= p.Spr  
ORDER BY Mbr
```