

Sistemi baza podataka

Dr Ivan Luković

Dr Slavica Kordić

Vladimir Dimitrieski

Nikola Todorović

Milica Todorović

Sistem ocenjivanja

- Ukupno: 100 bodova

	Predispitne obaveze	Završni ispit	Σ
Teoretski deo (predavanja)	15	30	45
Praktični deo (vežbe)	55	0	55
Σ	70	30	100

Praktični deo - vežbe

- 55 bodova
 - U toku nastave
 - 1 zadatak / vežbe –složeni oblici vežbi (Z1)
 - Realizacija zadatka na vežbama
 - » proceduralna proširenja SQL
 - » serversko programiranje -realizacija ograničenja šeme
 - Maks. 25 bodova
 - 1 projekat (P)
 - Projektovanje i realizacija NoSQL šeme baze podataka
 - Maks. 30 bodova

Praktični deo - vežbe

- Pravila realizacije obaveza
 - Zadaci / vežbe – složeni oblici vežbi (Z1)
 - 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
 - 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 (db2015):
 - Username: **raXY**
 - X – broj indeksa, Y – godina upisa
 - Password: **ftn**
 - Host Name: **192.168.0.102**
 - Port Number: **1522**
 - Oracle SID, database name: **db2015**

Rad u učionici

- SQL Developer
 - Local Disk C:\Program Files\sqldeveloper -> SQLDeveloper:
 - sqldeveloper.exe
- Materijali
 - www.acs.uns.ac.rs
 - sistemi baza podataka
 - repozitorijum
 - 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
– 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
– CLOB	Karakter do 4 GB
– BLOB	
– BFILE	binarni podaci smešteni u eksternom fajlu do 4 GB
– ROWID	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)
  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

```
CREATE TABLE projekt
(
  Spr integer not null,
  Ruk integer not null,
  Nap varchar(30),
  Nar varchar(30),
  CONSTRAINT projekt_PK PRIMARY KEY (Spr),
  CONSTRAINT projekt_FK FOREIGN KEY (Ruk)
  REFERENCES Radnik (Mbr),
  CONSTRAINT projekt_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
```

Pogled

CREATE [OR REPLACE] VIEW view

[(alias[, alias]...)]

AS subquery

[WITH CHECK OPTION [CONSTRAINT constraint]]

[WITH READ ONLY [CONSTRAINT constraint]]

DROP VIEW view

Pogled

- Napraviti pogled koji će za sve radnike prikazati Mbr i ukupan broj sati angažovanja radnika na projektima na kojima radi. Za radnike koji nisu angažovani ni na jednom projektu, ukupan broj sati angažovanja treba da iznosi 0.
- Nakon toga, napraviti pogled koji će za svakog šefa (rukovodioca radnika) prikazati njegov matični broj, prezime, ime, ukupan broj radnika kojima rukovodi i njegovo ukupno angažovanje na svim projektima, na kojima radi. Koristiti prethodno definisani pogled.
- Koliko je ukupno angažovanje svih šefova na projektima?

Pogled

```
CREATE OR REPLACE VIEW angaz_po_radnicima (Mbr, SBrC) AS
  SELECT r.Mbr, NVL(SUM(rp.BrC), 0)
  FROM radnik r, radproj rp
  WHERE r.Mbr = rp.Mbr (+)
  GROUP BY r.Mbr
```

```
CREATE VIEW angaz_sefova (Mbr, Prz, Ime, BrRad, BrSat) AS
  SELECT DISTINCT r.Sef, r1.Prz, r1.Ime, COUNT(*), a.SBrC
  FROM radnik r, radnik r1, angaz_po_radnicima a
  WHERE r.Sef = r1.Mbr AND r.Sef = a.Mbr
  GROUP BY r.Sef, r1.Prz, r1.Ime, a.SBrC
```

```
SELECT SUM(BrSat) AS UkAngSef
FROM angaz_sefova
```

Neke funkcije za rad sa datumom i vremenom

- `ADD_MONTHS(d, n)` – koristi se za dodavanje ili oduzimanje broja meseci od određenog datuma
- `LAST_DAY(d)` – vraća datum poslednjeg dana u mesecu zadatog parametrom sa datumom
- `MONTHS_BETWEEN(d1, d2)` – utvrđuje broj meseci između dva datuma
- `NEXT_DAY(d, char)` – za zadati datum i dan u nedelji f-ja vraća prvi sledeći datum koji upravo predstavlja taj dan u nedelji
- `ROUND(d [,fmt])` – vraća datum i vreme zaokruženo na najveći sledeći deo datuma
- `SYSDATE` – vraća trenutni datum i vreme
- `TRUNC(d, [fmt])` – vraća vrednost tipa DATE odsečenu do vrednosti navedene u parametru koji predstavlja masku za formatiranje

d – datum

n – broj meseci

Funkcije za rad sa datumom i vremenom

- Primeri upotrebe datumskih funkcija

ROUND('29-JUL-2004', 'MONTH') → '01-AUG-2004'

TRUNC('29-JUL-2004', 'MONTH') → '01-JUL-2004'

ROUND('29-JUL-2004', 'YEAR') → '01-JAN-2005'

TRUNC('29-JUL-2004', 'YEAR') → '01-JAN-2004'

MONTHS_BETWEEN('29-JUL-2004', '28-JUN-2004') →
1.03225806

ADD_MONTHS('11-JAN-1995',6) → '11-JUL-1995'

NEXT_DAY('05-JUN-2004','SUNDAY') → '06-JUN-2004'

LAST_DAY('05-JUN-2004') → '30-JUN-2004'

Funkcije za rad sa datumom i vremenom

- Prikazati matični broj i razliku između današnjeg dana i datuma rođenja radnika, iskazanu u sedmicama, za svakog radnika.

```
SELECT MBR, trunc((sysdate-god)/7, 0) "Nedelje od  
rodjenja"  
FROM Radnik;
```

Neke karakter funkcije

- LOWER(char) – za konvertovanje svih znakova u mala slova
- UPPER(char) – za konvertovanje svih znakova u velika slova
- INITCAP(char) – prvo slovo svake reči u nizu znakova pretvara u veliko slovo, a ostatak reči u mala slova
- SUBSTR(char, m [,n]) – koristi se za izdvajanje dela niza znakova
- TRIM(LEADING / TRAILING / BOTH, trim_character FROM trim_source) – uklanja početne ili prateće znakove sa početka ili kraja niza znakova
- LENGTH(char) – vraća broj znakova u nizu

Neke karakter funkcije - primeri

LOWER ('Sva mala slova') → 'sva mala slova'

UPPER ('Sva velika slova') → 'SVA VELIKA SLOVA'

INITCAP('Velika početna slova') → ' Velika Početna Slova'

SUBSTR('DobroJutro',1,5) → 'Dobro'

TRIM('D' FROM 'DobroJutro') → 'obroJutro'

LENGTH('DobroJutro') → 10

Neke karakter funkcije - primer

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
WHERE UPPER(Prz) = 'PETRIC';
```

Neke funkcije za konverziju podataka

- `TO_CHAR(d [, fmt])` – transformiše vrednosti tipa `DATE` u `VARCHAR2`, po izboru uz navedeni format datuma
- `TO_CHAR(n [, fmt])` – transformiše vrednost brojanog tipa u `VARCHAR2`, po izboru uz navedeni format broja
- `TO_DATE(char [, fmt])` – za konvertovanje niza znakova u ekvivalentni datum
- `TO_NUMBER(char [,fmt])` – za konvertovanje znakovnih vrednosti u numeričke

Modeli za formatiranje datumskih podataka

- -/,,:;"text" Punctuation and quoted text is reproduced in the result.
- AD/A.D. AD indicator with or without periods.
- AM/A.M. Meridian indicator with or without periods.
- BC/B.C. BC indicator. with or without periods.
- CC/SCC Century; "S" prefixes BC dates with "-".
- D Day of week (1-7).
- DAY Name of day, padded with blanks to length of 9 characters.
- DD Day of month (1-31).
- DDD Day of year (1-366).
- DY Abbreviated name of day.
- IW Week of year (1-52 or 1-53) based on the ISO standard.
- IYY/IY/I Last 3, 2, or 1 digit(s) of ISO year.
- IYYY 4-digit year based on the ISO standard.
- HH/HH12 Hour of day (1-12).
- HH24 Hour of day (0-23).
- J Julian day; the number of days since January 1, 4712 BC. Number specified with 'J' must be integers.
- MI Minute (0-59).
- MM Month (01-12; JAN = 01)
- MONTH Name of month, padded with blanks to length of 9 characters.

Modeli za formatiranje datumskih podataka

- MON Abbreviated name of month.
- PM/P.M. Meridian indicator with and without periods.
- Q Quarter of year (1, 2, 3, 4; JAN-MAR = 1)
- RM Roman numeral month (I-XII; JAN = I).
- RR Last 2 digits of year; for years in other countries.
- SS Second (0-59).
- SSSSS Seconds past midnight (0-86399).
- WW Week of year (1-53) where week 1 starts on the first day of the year and continues to the seventh day of the year.
- W Week of month (1-5) where week 1 starts on the first day of the month and ends on the seventh.
- Y/YYYY Year with comma in this position.
- YEAR/SYEAR Year, spelled out; "S" prefixes BC dates with "-".
- YYYY/SYYYY 4-digit year; "S" prefixes BC dates with "-".
- YYYY/YY/Y Last 3, 2, or 1 digit(s) of year.
- TH Ordinal Number DDTH 4TH
- SP Spelled Number DDSP FOUR
- SPTH or THSP Spelled, ordinal number DDSPTH FOURTH
- FM "Fill mode". This modifier suppresses blank padding in the return value of the TO_CHAR function

Modeli za formatiranje numeričkih podataka

- 9 9999
Return value with the specified number of digits with a leading space if positive. Return value with the specified number of digits with a leading minus if negative. Leading zeros are blank, except for a zero value, which returns a zero for the integer part of the fixed point number.
- 0 09999990
Return leading zeros. Return trailing zeros.
- \$ \$9999
Return value with a leading dollar sign.
- B B9999
Return blanks for the integer part of a fixed point number when the integer part is zero (regardless of "0"s in the format model).
- MI 9999MI
Return negative value with a trailing minus sign "-". Returns positive value with a trailing blank.
- S S99999999S
Return negative value with a leading minus sign "-". Return positive value with a leading plus sign "+". Return negative value with a trailing minus sign "-". Return positive value with a trailing plus sign "+".
- PR 9999PR
Return negative value in <angle brackets>. Return positive value with a leading and trailing blank.

Modeli za formatiranje numeričkih podataka

- **D** 99D99
Return a decimal point (that is, a period ".") in the specified position.
- **G** 9G999
Return a group separator in the position specified.
- **C** C999
Return the ISO currency symbol in the specified position.
- **L** L999
Return the local currency symbol in the specified position.
- **,** (comma) 9,999
Return a comma in the specified position.
- **.** (period) 99.99
Return a decimal point (that is, a period ".") in the specified position.
- **V** 999V99
Return a value multiplied by 10^n (and if necessary, round it up), where n is the number of "9"s after the "V".
- **EEEE** 9.9EEEE
Return a value using scientific notation.
- **RNrn** RN
Return a value as Roman numerals in uppercase. Return a value as Roman numerals in lowercase. Value can be an integer between 1 and 3999.
- **FM** FM90.9
Returns a value with no leading or trailing blanks.

Neke funkcije za konverziju podataka

```
select MONTHS_BETWEEN(TO_DATE('29-Jul-2004',  
    'DD-Mon-YYYY'),TO_DATE('29-JUN-2004', 'dd-  
    MON-YYYY'))  
from dual
```

```
select Prz, TO_CHAR(God, 'fmDD Month YYYY') AS  
    DatRodj  
FROM   Radnik
```

```
select TO_CHAR(Plt, '$99,999.00') Plata  
FROM   Radnik
```


Primeri

- Prikazati Mbr i podstringove prezimena i imena radnika dužine 5, koji počinju od pozicije 3.

```
SELECT Mbr, SUBSTR(Prz, 3, 5) AS Prz_deo3_5,  
SUBSTR(Ime, 3, 5) AS Ime_deo3_5  
FROM radnik
```

Primeri

- Prikazati vrednosti obeležja Mbr, Prz, Ime i Sef za sve radnike, pri čemu se kao vrednost obeležja Sef, onda kada Sef ima NULL vrednost u tabeli, pojavljuje konstanta "glavni rukovodilac".

```
SELECT Mbr, Prz, Ime,  
       NVL(TO_CHAR(Sef), 'glavni rukovodilac') Sef  
FROM radnik
```

Primeri

- Kog dana u nedelji ste rođeni? Napomena: **iskoristiti SYS.DUAL**. Upit napisati tako da se datum rođenja može **interaktivno unositi**.

```
SELECT TO_CHAR(TO_DATE('&Dat_rodj',  
    'DD.MM.YYYY'), 'DAY') as "Dan rodjenja"  
FROM SYS.DUAL
```

Klauzula WITH

- dodela naziva bloku podupita
- blok može biti referenciran više puta unutar upita
- optimizacija upita
 - kao umetnuti pogled
 - kao privremena tabela
- cilj: **smanjenje kompleksnosti upita**

```
WITH naziv_upita (lista_obelezja) AS (  
    upit  
)
```

Primer

- Prikazati za svakog radnika angažovanog na projektu mbr, prz, ime, spr i broj drugih radnika koji su angažovani na istom projektu

```
select r.mbr, r.ime, r.prz, rp1.spr, count(rp2.mbr)-1 ostali
from radnik r, radproj rp1, radproj rp2
where r.mbr=rp1.mbr and rp1.spr=rp2.spr
group by r.mbr, r.ime, r.prz, rp1.spr;
```

```
with projinfo as (
    select rp.spr, count(rp.mbr) as rad_broj
    from radproj rp group by rp.spr
)
select r.mbr, r.ime, r.prz, rp.spr, pi.rad_broj-1 ostali
from radnik r, radproj rp, projinfo pi
where r.mbr=rp.mbr and rp.spr=pi.spr;
```

Zadatak

- Prikazati za svakog radnika angažovanog na projektu mbr, prz, ime, spr i udeo u ukupnom broju časova rada na tom projektu (zaokruženo na dve decimale).

```
with projinfo as (  
    select rp.spr, sum(rp.brc) as cas_suma  
    from radproj rp  
    group by rp.spr  
)  
select r.mbr, r.ime, r.prz, rp.spr, round(rp.brc/pi.cas_suma, 2) udeo  
from radnik r, radproj rp, projinfo pi  
where r.mbr=rp.mbr and rp.spr=pi.spr;
```

Zadatak

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

```
with projinfo as (  
    select spr, avg(brc) prosek  
    from radproj group by spr  
)  
select distinct r.mbr, r.ime, r.prz, r.plt  
from radnik r, radproj rp, projinfo pi  
where r.mbr=rp.mbr and rp.spr=pi.spr  
and rp.brc > prosek;
```

Zadatak

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

```
with projinfo as (  
    select spr, avg(brc) pros  
    from radproj group by spr  
)  
select distinct r.mbr, r.ime, r.prz, r.plt  
from radnik r, radproj rp, projinfo pi  
where r.mbr=rp.mbr and rp.spr=pi.spr  
    and rp.brc>(select avg(pros) from projinfo);
```


Primer

- Prikazati mbr, ime i prz rukovodilaca projekata kao i ukupan broj radnika kojima rukovode na projektima

with rukovodilac as (

select mbr, ime, prz, plt, spr
from radnik, projekat where mbr=ruk),

projinfo as (

select spr, count(mbr) ljudi
from radproj group by spr)

select ru.mbr, ru.ime, ru.prz, sum(pi.ljudi) ljudi
from rukovodilac ru, projinfo pi
where ru.spr=pi.spr
group by ru.mbr, ru.ime, ru.prz;

Zadatak

- Koliko je ukupno angažovanje svih šefova na projektima?

```
with angaz_po_radnicima (mbr, sbrc) as (  
    select r.mbr, nvl(sum(rp.brc), 0)  
    from radnik r, radproj rp  
    where r.mbr = rp.mbr (+)  
    group by r.mbr),  
angaz_sefova (mbr, prz, ime, brrad, brsat) as (  
    select distinct r.sef, r1.prz, r1.ime, count(*), a.sbrc  
    from radnik r, radnik r1, angaz_po_radnicima a  
    where r.Sef = r1.Mbr and r.Sef = a.Mbr  
    group by r.Sef, r1.Prz, r1.Ime, a.SBrc)  
select sum(brsat) as ukangsef  
from angaz_sefova;
```

WITH – Rekurzija

- blok podupita pomoću WITH
- blok sadrži dva upita vezana preko UNION ALL
 - prvi upit određuje početni skup podataka
 - drugi upit obezbeđuje rekurzivno proširenje skupa putem unije sa tekućim skupom
- postupak se zaustavlja kada ne dođe do promene skupa prilikom proširenja

```
WITH naziv_upita (lista_obeležja) AS  
(  
    upit1  
    UNION ALL  
    upit2  
)
```

Primer

- Prikazati za svakog radnika sve direktno i indirektno nadređene radnike

```
with hijerarhija(mbr,sef) as(
    select mbr, sef
    from radnik
    UNION ALL
    select r.mbr, h.sef
    from radnik r, hijerarhija h
    where r.sef = h.mbr and h.sef is not null
)
select *
from hijerarhija
order by mbr, sef;
```

Zadatak

- Prikazati za svakog radnika sve direktno i indirektno podređene radnike

```
with hijerarhija(mbr,pod) as (  
    select sef, mbr  
    from radnik  
    UNION ALL  
    select h.mbr, r.mbr  
    from hijerarhija h, radnik r  
    where h.pod = r.sef and h.mbr is not null  
)  
select *  
from hijerarhija  
order by mbr, pod;
```

Zadatak

- Prikazati za svakog radnika sve direktno i indirektno podređene radnike, ako nema podređenih prikazati null umesto oznake podređenog

Zadatak

```
with hijerarhija(mbr,pod) as (  
    select sef, mbr  
    from radnik  
    UNION ALL  
    select h.mbr, r.mbr  
    from hijerarhija h, radnik r  
    where h.pod = r.sef and h.mbr is not null)  
select r.mbr, h.pod  
from radnik r, hijerarhija h  
where r.mbr = h.mbr(+)  
order by r.mbr, pod;
```

WITH – Rekurzija – SEARCH

- klauzula SEARCH
- definiše poredak redova
 - BREADTH FIRST, DEPTH FIRST
- BY – poredak redova na istom nivou
- SET – vrednost pseudo-obeležja po redosledu redova
 - pseudo-obeležje automatski postaje deo rezultata

SEARCH BREADTH | DEPTH FIRST

BY lista_obeležja

SET pseudo-obeležje

Primer

- Prikazati hijerarhiju rukovođenja po nivoima

```
with hijerarhija(mbr,imeprz,sef,nivo) as (  
    select mbr, ime||' '||prz, sef, 1 as nivo  
    from radnik  
    where sef is null  
    union all  
    select r.mbr, r.ime||' '||r.prz, h.mbr, nivo+1  
    from radnik r, hijerarhija h  
    where r.sef = h.mbr)  
search depth first by imeprz set poredak  
select nivo, rpad(' ',3*nivo)||imeprz as imeprz, mbr, sef  
from hijerarhija  
order by poredak;
```

Primer

- Prikazati hijerarhiju rukovođenja po nivoima

```
with hijerarhija(mbr,imeprz,sef,nivo) as (  
    select mbr, ime||' '||prz, sef, 1 as nivo  
    from radnik  
    where sef is null  
    union all  
    select r.mbr, r.ime||' '||r.prz, h.mbr, nivo+1  
    from radnik r, hijerarhija h  
    where r.sef = h.mbr)  
search breadth first by imeprz set poredak  
select nivo, rpad(' ',3*nivo)||imeprz as imeprz, mbr, sef  
from hijerarhija  
order by poredak;
```

Zadatak

- Prikazati za svakog radnika lanac rukovođenja i glavnog radnika. Na primer:

IMEPRZ	MBR	SEF	LANAC	GLAVNI
Milan Milic	20	10	/Pera Peric/Milan Milic	Pera Peric
.....				

Zadatak

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as (  
    select ime||' '||prz, mbr, sef,  
           '/'||ime||' '||prz as lanac,  
           ime||' '||prz as glavni  
    from radnik  
    where sef is null  
    union all  
    select r.ime||' '||r.prz, r.mbr, r.sef,  
           h.lanac||'/'||r.ime||' '||r.prz as lanac, h.glavni  
    from radnik r, hijerarhija h  
    where r.sef = h.mbr )  
select imeprz,mbr,sef,lanac,glavni  
from hijerarhija;
```

WITH – Rekurzija – CYCLE

- klauzula CYCLE
- označava cikluse u rekurziji
 - prema proveru zadate liste obeležja
- oznaka prisustva ili odsustva ciklusa
 - pseudo-obeležje automatski postaje deo rezultata
 - jedan karakter

CYCLE lista_obeležja

SET pseudo_obeležje

TO oznaka_ciklusa

DEFAULT oznaka_odsustva_ciklusa

Primer

- Prikazati za svakog radnika lanac rukovođenja, uz proveru postojanja ciklusa

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as
( select ime||' '||prz, mbr, sef,
  '/'||ime||' '||prz as lanac, ime||' '||prz as glavni
  from radnik where sef is null
  union all
  select r.ime||' '||r.prz, r.mbr, r.sef,
  h.lanac||'/'||r.ime||' '||r.prz as lanac, h.glavni
  from radnik r, hijerarhija h where r.sef = h.mbr )
search breadth first by imeprz set poredak
cycle mbr set ciklus to 'x' default 'o'
select imeprz,mbr,sef,lanac,glavni, ciklus from
hijerarhija;
```

Zadatak

- Prikazati sve podređene za radnika 70

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as
( select ime||' '||prz, mbr, sef,
  '/'||ime||' '||prz as lanac,
  ime||' '||prz as glavni
  from radnik where mbr = 70
  union all
  select r.ime||' '||r.prz, r.mbr, r.sef,
  h.lanac||'/'||r.ime||' '||r.prz as lanac, h.glavni
  from radnik r, hijerarhija h where r.sef = h.mbr )
search breadth first by imeprz set poredak
cycle mbr set ciklus to 'x' default 'o'
select imeprz,mbr,sef,lanac,glavni, ciklus from
hijerarhija;
```

Zadatak

- Promeniti šefa radnika 70 da bude radnik 140

```
update radnik  
set sef = 140  
where mbr = 70;
```


Zadatak

- Ponovo prikazati sve podređene za radnika 70

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as
( select ime||' '||prz, mbr, sef,
  '/'||ime||' '||prz as lanac, ime||' '||prz as glavni
  from radnik where mbr = 70
  union all
  select r.ime||' '||r.prz, r.mbr, r.sef,
  h.lanac|| '/'||r.ime||' '||r.prz as lanac, h.glavni
  from radnik r, hijerarhija h
  where r.sef = h.mbr )
search breadth first by imeprz set poredak
cycle mbr set ciklus to 'x' default 'o'
select imeprz, mbr, sef, lanac, glavni, ciklus
from hijerarhija;
```

Zadatak

- Poništiti promenu šefa radnika 70

rollback;

Sekvencer (Generator sekvence vrednosti)

CREATE SEQUENCE sequence

[INCREMENT BY n]

[START WITH n]

[{MAXVALUE n | NOMAXVALUE}]

[{MINVALUE n | NOMINVALUE}]

[{CYCLE | NOCYCLE}]

[{CACHE n | NOCACHE}]

ALTER SEQUENCE sequence ...

DROP SEQUENCE sequence

Primer upotrebe sekvencera

```
CREATE SEQUENCE SEQ_Mbr  
    INCREMENT BY 1  
    START WITH 300  
    NOCYCLE  
    CACHE 10
```

```
INSERT INTO radnik (Mbr, Prz, Ime, God)  
VALUES (SEQ_Mbr.NEXTVAL, 'Misic', 'Petar', SYSDATE)
```

Primer upotrebe sekvencera

```
SELECT SEQ_Mbr.CURRVAL  
FROM SYS.DUAL
```

Tabele u Oracle bazi podataka

- korisničke tabele
 - kolekcije tabela koje kreira i održava korisnik
 - sadrže korisničke informacije
- **Data Dictionary** (rečnik podataka)
 - kolekcija tabela koje kreira i održava Oracle server
 - sadrže informacije baze podataka
 - vlasnik svih tabela u rečniku je SYS korisnik
 - informacije smeštene u rečniku podataka obuhvataju imena korisnika Oracle servera, privilegije dodeljene korisnicima, nazive objekata baze podataka, ograničenja.
 - postoji nekoliko kategorija pogleda rečnika podataka; svaka od njih ima odgovarajući prefiks:
 - USER_ - ovi pogledi sadrže informacije o objektima čiji je vlasnik korisnik
 - ALL_ - ovi pogledi sadrže informacije o svim tabelama (objektnim i relacionim) koje su dostupne korisniku
 - DBA_ - ovi pogledi su zabranjeni, tj. dostupni su samo korisnicima koji imaju DBA ulogu

Tabele u Oracle bazi podataka

- Upiti u rečniku podataka se postavljaju kao i svi ostali upiti.
- Prikazati nazive tabela čiji je vlasnik korisnik.

```
SELECT table_name  
FROM user_tables;
```
- Prikazati različite tipove objekata čiji je vlasnik korisnik.

```
SELECT DISTINCT object_type  
FROM user_objects;
```
- Prikazati tabele, pogledе, sinonime i sekvence čiji je vlasnik korisnik.

```
SELECT *  
FROM user_catalog;
```