

Lekcija 03

SQL: Naredba SELECT za rad sa više tabela

dr Svetlana Cvetanović



SQL: NAREDBA SELECT ZA RAD SA VIŠE TABELA

Uvod

01

02

03

04

Uvod

**Spajanje tabela
korišćenjem
podupita**

**JOIN za spajanje
tabela**

Zaključak

*Vežba: Spajanje
tabela korišćenjem
podupita*

*Vežba: JOIN za
spajanje tabela*

UVOD

Šta ćemo naučiti u ovoj lekciji?

Naredbom SELECT se mogu dobiti sadržaji i iz više tabela za šta se mogu primeniti dve metode:

- spajanje (JOIN) tabela i
- korišćenje podupita.

U predavanju su dati primeri korišćenja oba ova načina i ukazano je na njihove prednosti i nedostatke. Obzirom na ograničenja koja postoje prilikom korišćenja podupita, mnogo češći način za dobijanje podataka iz više tabela je spajanje (JOIN).

Mogu se primenjivati različite vrste JOIN-a kao što su: CARTESIAN JOIN, EQUIJOIN, OUTER JOIN i SELF JOIN.

Spajanje tabela korišćenjem podupita

□ *Vežba: Spajanje tabela korišćenjem podupita*

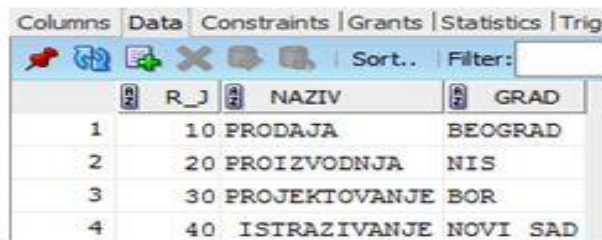
01

TABELE ZA PRAVLJENJE PODUPITA RADI SPAJANJA TABELA

Tabela RADNIK i RJ

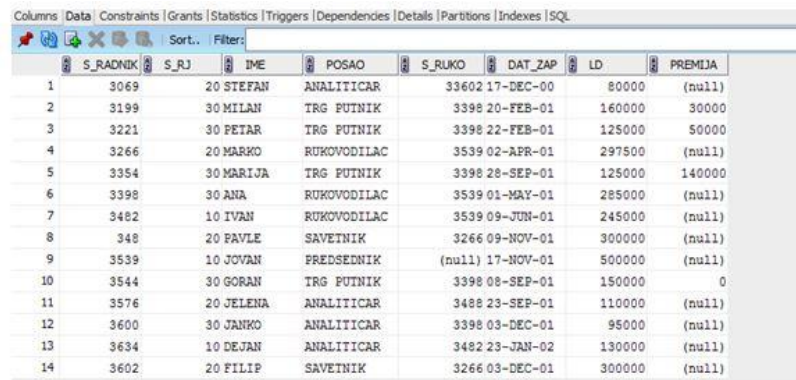
SQL raspolaže sa dve različite tehnike za pravljenje upita nad više tabela: korišćenje podupita tj. umetanje upita nad jednom relacijom u upit nad drugom i JOIN. Mada se obe tehnike koriste za rad sa više tabela, one se koriste u različitim situacijama.

Ovde će biti reči o pravljenju upita nad više tabela umetanje upita nad jednom relacijom u upit nad drugom što će biti ilustrirano na primeru relacionog modela sastavljenog od dve relacije: RJ (RADNA_JEDINICA i RADNIK koje su pretstavljene slikama 1. i 2.:



R_J	NAZIV	GRAD
10	PRODAJA	BEOGRAD
20	PROIZVODNJA	NIS
30	PROJEKTOVANJE	BOR
40	ISTRAZIVANJE	NOVI SAD

Slika 1. Tabela RJ



S_RADNIK	S_RJ	IME	POSAO	S_RUKO	DAT_ZAP	LD	PREMIJA
3069	20	STEFAN	ANALITICAR	33602	17-DEC-00	80000	(null)
3199	30	MILAN	TRG PUTNIK	3398	20-FEB-01	160000	30000
3221	30	PETAR	TRG PUTNIK	3398	22-FEB-01	125000	50000
3266	20	MARKO	RUKOVOĐILAC	3539	02-APR-01	297500	(null)
3354	30	MARIJA	TRG PUTNIK	3398	28-SEP-01	125000	140000
3398	30	ANA	RUKOVOĐILAC	3539	01-MAY-01	285000	(null)
3482	10	IVAN	RUKOVOĐILAC	3539	09-JUN-01	245000	(null)
348	20	PAVLE	SAVEINIK	3266	09-NOV-01	300000	(null)
3539	10	JOVAN	FREDSIEDNIK	(null)	17-NOV-01	500000	(null)
3544	30	GORAN	TRG PUTNIK	3398	08-SEP-01	150000	0
3576	20	JELENA	ANALITICAR	3488	23-SEP-01	110000	(null)
3600	30	JANKO	ANALITICAR	3398	03-DEC-01	95000	(null)
3634	10	DEJAN	ANALITICAR	3482	23-JAN-02	130000	(null)
3602	20	FILIP	SAVEINIK	3266	03-DEC-01	300000	(null)

Slika 2. Tabela RADNIK

KORIŠĆENJE PODUPITA ZA SPAJANJE TABELA

Primer u kojem se može upotrebiti podupit za spajanje tabela

Jedan od načina povezivanja tabela relacione baze podataka je umetanje upita nad jednom relacijom u upit nad drugom, odnosno, preciznije rečeno, dinamička zamena rezultata jednog upita u WHERE klauzuli drugog.

Primer: Prikazati ime i posao svakog radnika koji ima isti posao kao Dejan.

Najpre bi trebalo utvrditi koji posao obavlja Dejan. To se realizuje sledećom SQL naredbom kojom saznajemo da DEJAN obavlja posao analitičara (slika 3.)

Sada se polazni zahtev svodi na prikazivanje imena i posla svakog radnika koji obavlja posao analitičara, što se realizuje sledećom naredbom (slika 4.)



```
SELECT IME, POSAO  
FROM RADNIK  
WHERE POSAO= 'ANALITICAR';
```

Slika 4. Naredba SELECT za prethodni primer



```
SELECT POSAO FROM RADNIK WHERE IME='DEJAN'
```

Izvršavanjem ove SELECT naredbe dobijamo izlaz

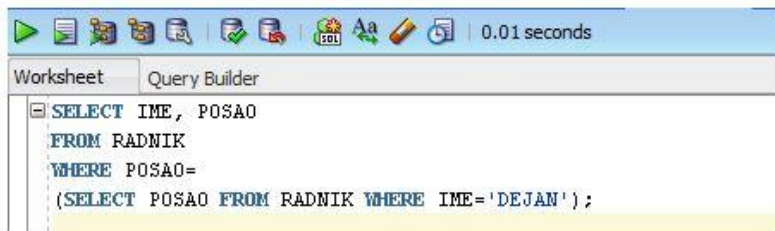
```
POSAO  
-----  
ANALITICAR
```

Slika 3. Naredba SELECT i njen izlaz za prethodni primer

KORIŠĆENJE PODUPITA ZA SPAJANJE TABELA

Primer upotrebe podupita za spajanje tabela

Povezivanje tabela dinamičkom zamenom rezultata jednog upita u WHERE klauzuli drugog se sastoji upravo u tome da se umesto rezultata prvog upita (u konkretnom slučaju 'ANALITIČAR') u WHERE klauzuli drugog, piše prvi upit tj. upit koji vraća taj rezultat (u konkretnom slučaju), tako da SELECT naredba ima sledeći izgled (slika 5.):



```
SELECT IME, POSAO
FROM RADNIK
WHERE POSAO=
(SELECT POSAO FROM RADNIK WHERE IME='DEJAN');
```

Iz prikazane SELECT naredbe se dobija izlaz:

IME	POS AO
STEFAN	ANALITICAR
JELENA	ANALITICAR
JANKO	ANALITICAR
DEJAN	ANALITICAR

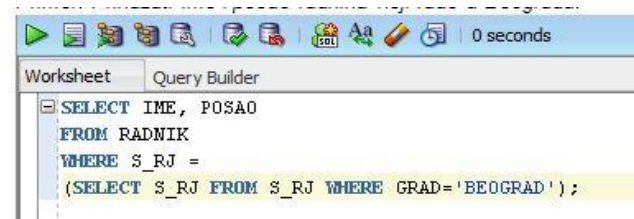
Slika 5. Naredba SELECT i njen izlaz za prethodni primer

Prvi upit, odnosno upit u zagradama se naziva unutrašnji upit i on se uvek izvršava prvi. Drugi upit se naziva spoljašnji upit. Pre

nego što počne njegovo izvršavanje unutrašnji upit je već završen i u zagradi se nalazi konkretna vrednost rezultata izvršavanja unutrašnjeg upita.

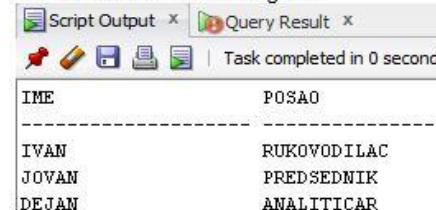
Navedeni mehanizam izvršavanja SELECT naredbe prouzrokuje pojavljivanje i n-torke o Dejanu u rezultujućoj tabeli. Ukoliko tu n-torku hoćemo da eliminišemo, u WHERE klauzulu spoljašnjeg upita treba dodati uslov: AND IME !='DEJAN'

Primer: Prikazati ime i posao radnika koji rade u Beogradu (slika 6.).



```
SELECT IME, POSAO
FROM RADNIK
WHERE S_RJ =
(SELECT S_RJ FROM S_RJ WHERE GRAD='BEOGRAD');
```

Izlaz iz SELECT naredbe glasi:



IME	POS AO
IVAN	RUKOVODILAC
JOVAN	PREDSEDNIK
DEJAN	ANALITICAR

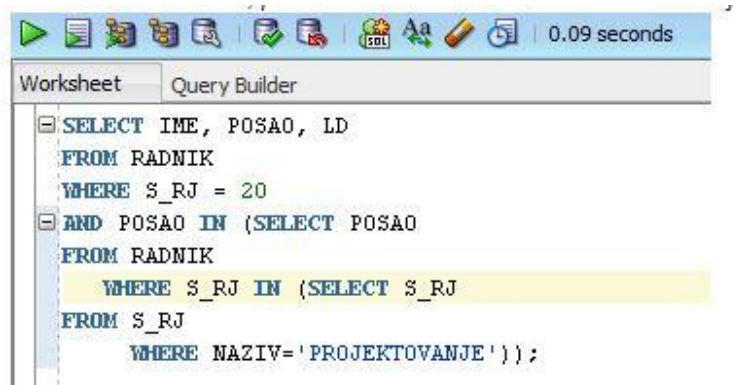
Slika 6. Naredba SELECT i njen izlaz za prethodni primer

KORIŠĆENJE PODUPITA ZA SPAJANJE TABELA

Primeri

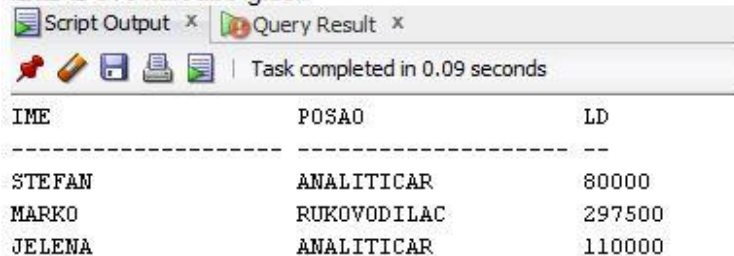
Primer: Prikazati ime, posao i lični dohodak radnika u RJ 20 koji imaju isti posao kao radnici RJ projektovanje (slika 7.).

Primer: Ko je najbolje plaćeni radnik u svakom odeljenju? Slika 8.



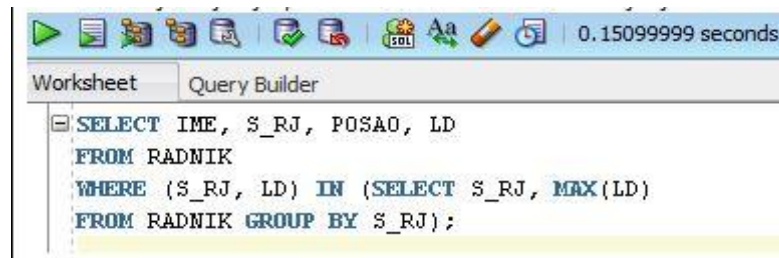
```
SELECT IME, POSAO, LD
FROM RADNIK
WHERE S_RJ = 20
AND POSAO IN (SELECT POSAO
FROM RADNIK
WHERE S_RJ IN (SELECT S_RJ
FROM S_RJ
WHERE NAZIV='PROJEKTOVANJE'));
```

Izlaz iz ove naredbe glasi:



IME	POSAO	LD
STEFAN	ANALITICAR	80000
MARKO	RUKOVODILAC	297500
JELENA	ANALITICAR	110000

Slika 7. Naredba SELECT i njen izlaz za prethodni primer



```
SELECT IME, S_RJ, POSAO, LD
FROM RADNIK
WHERE (S_RJ, LD) IN (SELECT S_RJ, MAX(LD)
FROM RADNIK GROUP BY S_RJ);
```

Slika 8. Naredba SELECT za prethodni primer

Iz ove naredbe se dobija sledeći izlaz:

IME	S_RJ	POSAO	LD
ANA	30	RUKOVODILAC	285000
PAVLE	20	SAVETNIK	300000
JOVAN	10	PREDSEDNIK	500000
FILIP	20	SAVETNIK	300000

Slika 9. Izlaz iz SELECT naredbe sa slike 8.

Ovaj primer pokazuje da spoljašnji i unutrašnji upit mogu biti povezani vrednostima više atributa.

Podupiti su veoma moćno sredstvo za spajanje dveju tabela, ali oni imaju i niz ograničenja: (1) mogu se selektovati podaci samo iz tabele koja se koristi na najvišem nivou; (2) podupit se ne može koristiti za dobijanje podataka iz više od jedne tabele. Zbog toga je u mnogim situacijama za selektovanje podataka iz više tabela potrebno koristiti JOIN.

Vežba: Spajanje tabela korišćenjem podupita

01

PRIKAZ TABELA STUDENTSKE BAZE

Studentska baza sadrži tabele DOSIJE, ISPIT, PREDMET i ISPITNI_ROK

SELECT * FROM `dosije`

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	INDEKS	IME	PREZIME	GOD_RODJENJA	MESTO_RODJENJA
<input type="checkbox"/>	20100021	Milos	Peric	1992	Beograd
<input type="checkbox"/>	20100022	Marijana	Savkovic	1993	Kraljevo
<input type="checkbox"/>	20100023	Sanja	Terzic	1991	Beograd
<input type="checkbox"/>	20100024	Nikola	Vukovic	1992	NULL
<input type="checkbox"/>	20100025	Ljubica	Savkovic	1991	Kraljevo
<input type="checkbox"/>	20100026	Zorica	Miladinovic	1993	Vranje
<input type="checkbox"/>	20100027	Milena	Stankovic	0	NULL

Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika-1. Prikaz tabele DOSIJE

SELECT * FROM `ispit`

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	indeks	id_predmeta	godina_roka	oznaka_roka	ocena	datum_ispita
<input type="checkbox"/>	20100021	1001	2011	jan	9	2020-01-20
<input type="checkbox"/>	20100021	1021	2011	apr	7	2003-04-20
<input type="checkbox"/>	20100021	2001	2011	jan	10	2025-01-20
<input type="checkbox"/>	20100021	3001	2011	jan	7	2027-01-20
<input type="checkbox"/>	20100022	1001	2011	jan	8	2020-01-20
<input type="checkbox"/>	20100022	1021	2011	apr	5	2003-04-20
<input type="checkbox"/>	20100022	2001	2011	jan	9	2025-01-20
<input type="checkbox"/>	20100023	1001	2011	jan	8	2020-01-20
<input type="checkbox"/>	20100023	1021	2011	apr	10	2003-04-20
<input type="checkbox"/>	20100023	2001	2011	jan	8	2025-01-20
<input type="checkbox"/>	20100023	3001	2011	jan	5	2027-01-20
<input type="checkbox"/>	20100024	1001	2011	jan	10	2020-01-20
<input type="checkbox"/>	20100024	1021	2011	apr	6	2003-04-20
<input type="checkbox"/>	20100024	2001	2011	jan	7	2025-01-20
<input type="checkbox"/>	20100024	3001	2011	jan	6	2028-01-20
<input type="checkbox"/>	20100025	1001	2011	jan	6	2020-01-20
<input type="checkbox"/>	20100025	2001	2011	feb	6	2010-02-20
<input type="checkbox"/>	20100025	2001	2011	jan	5	2025-01-20
<input type="checkbox"/>	20100026	1001	2011	feb	7	2010-02-20
<input type="checkbox"/>	20100026	1001	2011	jan	5	2020-01-20
<input type="checkbox"/>	20100026	1021	2011	apr	8	2003-04-20
<input type="checkbox"/>	20100026	2001	2011	feb	7	2010-02-20
<input type="checkbox"/>	20100026	3001	2011	jan	6	2028-01-20

Check All With selected: Change Delete Export

Slika-2. Prikaz tabele ISPIT

PRIKAZ TABELA STUDENTSKE BAZE

Studentska baza sadrži tabele DOSIJE, ISPIT, PREDMET i ISPITNI_ROK

`SELECT * FROM `ispitni_rok``

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	godina_ropa	oznaka_ropa	naziv
<input type="checkbox"/> Edit Copy Delete	2011	apr	April 2011
<input type="checkbox"/> Edit Copy Delete	2011	feb	Februar 2011
<input type="checkbox"/> Edit Copy Delete	2011	jan	Januar 2011
<input type="checkbox"/> Edit Copy Delete	2011	jun	Jun 2011
<input type="checkbox"/> Edit Copy Delete	2011	okt	Oktober 2011
<input type="checkbox"/> Edit Copy Delete	2011	sep	Septembar 2011

↑ Check All With selected: Change Delete Export

Number of rows: 25 Filter rows: Search this table

Slika-3. Prikaz tabele ISPITNI_ROK

`SELECT * FROM `predmet``

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	id_predmeta	sifra	naziv	bodovi
<input type="checkbox"/> Edit Copy Delete	1001	M111	Matematika 1	6
<input type="checkbox"/> Edit Copy Delete	1002	M112	Matemaitka 2	6
<input type="checkbox"/> Edit Copy Delete	1003	M113	Statistika sa verovatnocom	6
<input type="checkbox"/> Edit Copy Delete	1021	M131	Racunarske mreze	6
<input type="checkbox"/> Edit Copy Delete	1101	M105	Diskretne strukture 1	6
<input type="checkbox"/> Edit Copy Delete	1102	M106	Diskretne strukture 2	6
<input type="checkbox"/> Edit Copy Delete	2001	P101	Programiranje 1	8
<input type="checkbox"/> Edit Copy Delete	2002	P102	Programiranje 2	8
<input type="checkbox"/> Edit Copy Delete	2003	P103	Objektno orijentisano programiranje	6
<input type="checkbox"/> Edit Copy Delete	2004	P104	Algoritmi i strukture podataka	6
<input type="checkbox"/> Edit Copy Delete	3001	S1	Engleski jezik 1	5
<input type="checkbox"/> Edit Copy Delete	3002	S2	Engleski jezik 2	5
<input type="checkbox"/> Edit Copy Delete	4001	R101	Uvod u organizaciju racunara	5
<input type="checkbox"/> Edit Copy Delete	4002	R102	Uvod u Veb i Internet tehnologije	5

↑ Check All With selected: Change Delete Export

Slika-4. Prikaz tabele PREDMET

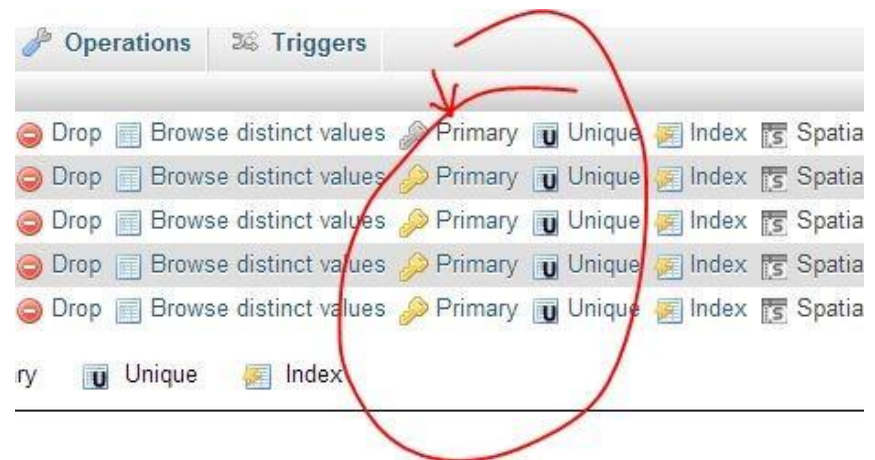
PRIMARNI KLJUČ

Primarni ključ (PRIMARY KEY) je ključno polje tabele kojim se kontroliše način na koji se informacije povezuju sa drugim tabelama

Pre svega potrebno je znati šta je primarni ključ.

Primarni ključ (PRIMARY KEY) je ključno polje tabele kojim se kontroliše način na koji se informacije povezuju sa drugim tabelama.

Primarni ključ na jedinstven način definiše zapis.



Slika-5. Prikaz kreiranja primarnog ključa

PRIMER 1.

Naredni primer služi za utvrđivanje pisanja podupita

Primer 1.

Pronaci sva imena i prezimena studenata iz tabele dosije koji žive u gradu u kome živi i student čiji je broj indeksa 20100021.

```
1 |select `IME`, `PREZIME`
2 |from DOSIJE
3 |where `MESTO_RODZENJA` =
4 |(SELECT `MESTO_RODZENJA` FROM DOSIJE WHERE `INDEKS`=20100021)
```

SELECT * SELECT INSERT UPDATE DELETE Clear

Slika-6. Prikaz upita

Osim na prikazani način, moguće je još uz pomoć **IN** klazule dobiti iste rezultate. Slika-8

Sort by key:

+ Options

IME	PREZIME
Milos	Peric
Sanja	Terzic

Slika-7. Prikaz rezultata upita

```
1 |select `IME`, `PREZIME`
2 |from DOSIJE
3 |where `MESTO_RODZENJA` IN
4 |(SELECT `MESTO_RODZENJA` FROM DOSIJE WHERE
  `INDEKS`=20100021)
```

SELECT * SELECT INSERT UPDATE DELETE Clear

Slika-8. Prikaz upita korišćenjem operatora IN

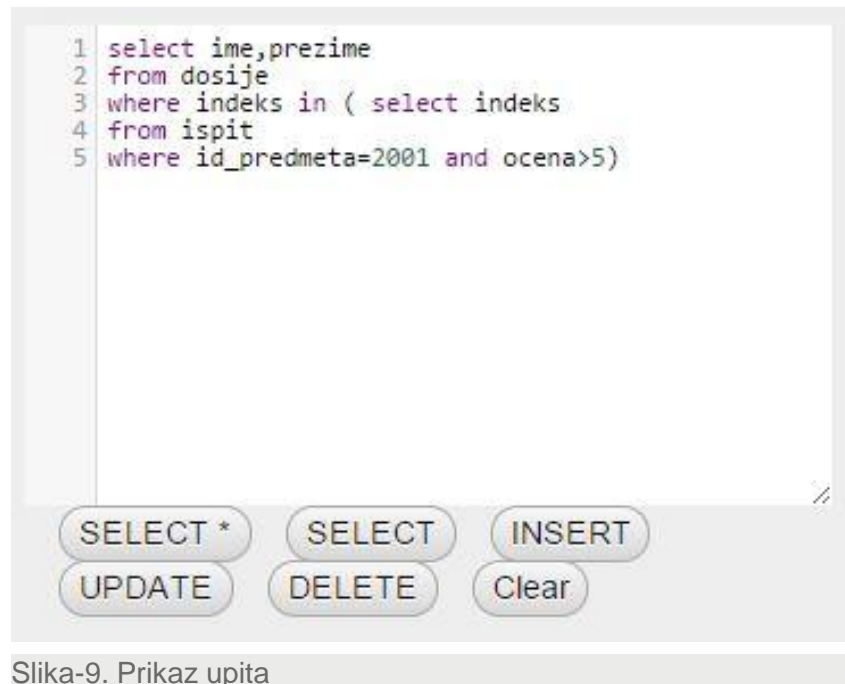
PRIMER 2.

Cilj primera jeste vežbanje korišćenja operatora in i pisanja podupita

Primer 2.

Izdvojiti imena i prezimena studenata koji su položili predmet čiji je id 2001.

```
1 select ime,prezime
2 from dosije
3 where indeks in ( select indeks
4 from ispit
5 where id_predmeta=2001 and ocena>5)
```



Slika-9. Prikaz upita

Sort by key:

+ Options

ime	prezime
Milos	Peric
Marijana	Savkovic
Sanja	Terzic
Nikola	Vukovic
Ljubica	Savkovic
Zorica	Miladinovic

Slika-10. Prikaz rezultata upita

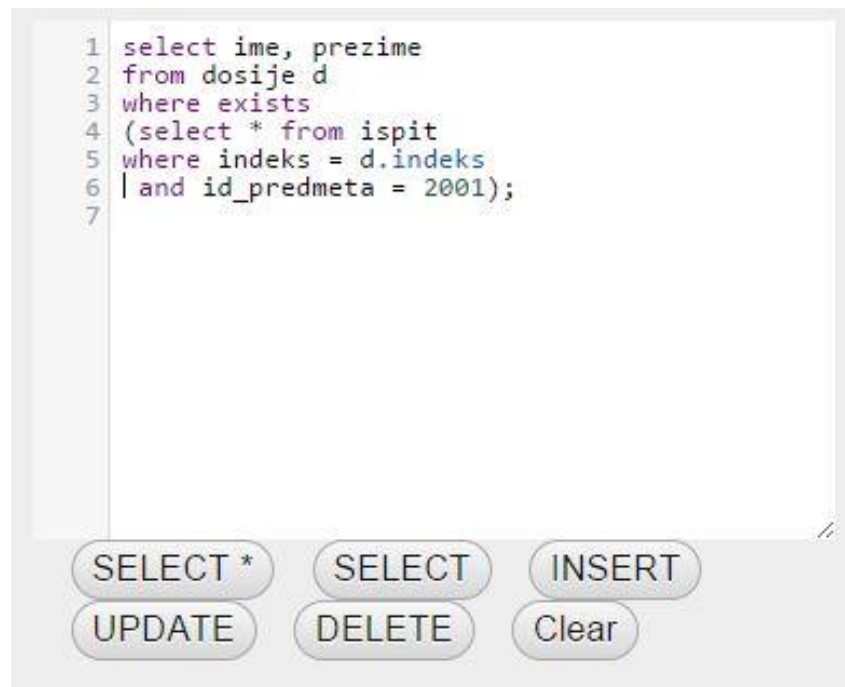
PRIMER 3.

Cilj primera jeste korišćenje operatora EXISTS i pisanja podupita

Primer 3.

Izdvojiti imena i prezimena studenata koji su položili predmet čiji je id 2001. Koristiti operator EXISTS

```
1 select ime, prezime
2 from dosije d
3 where exists
4 (select * from ispit
5  where indeks = d.indeks
6  | and id_predmeta = 2001);
7
```



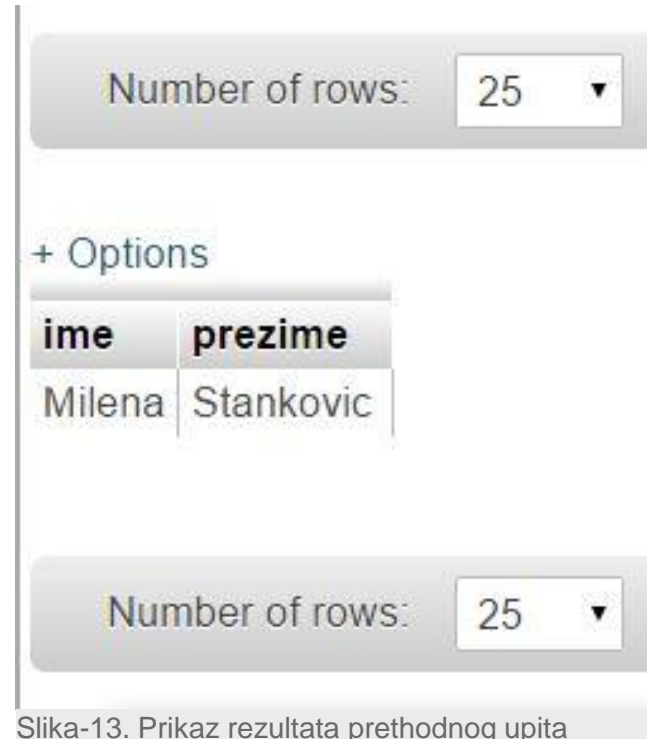
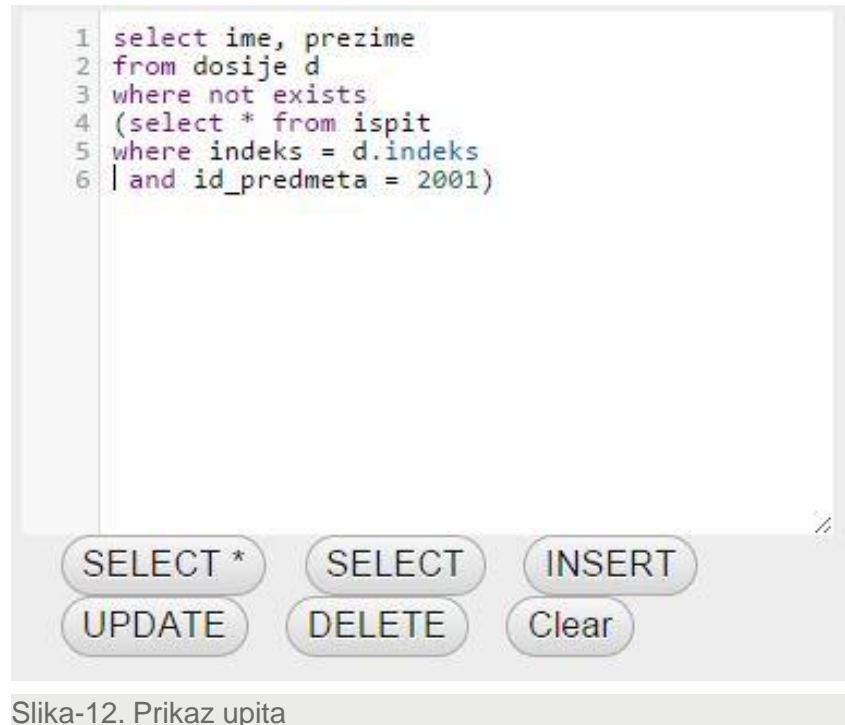
Slika-11. Prikaz upita

PRIMER 4.

Korišćenje NOT EXISTS klauze

Primer 4.

Izdvojiti imena i prezimena studenata koji nisu položili predmet čiji je id 2001



PRIMER 5.

Vežbanje pisanja podupita sa primenom IN operatora

Primer 5.

Izdvojiti brojeve indeksa, imena i prezimena studenta koji su polagali predmet koji nosi 5 bodova.

```
1 select indeks, ime, prezime
2 from dosije
3 where indeks in
4 ( select indeks from ispit
5 where id_predmeta in
6 ( select id_predmeta
7 from predmet
8 where bodovi=5));
9
```

SELECT * SELECT INSERT
UPDATE DELETE Clear

Slika-14. Prikaz upita

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

indeks	ime	prezime
20100021	Milos	Peric
20100023	Sanja	Terzic
20100024	Nikola	Vukovic
20100026	Zorica	Miladinovic

Number of rows: 25 Filter rows: Search this table

Slika-15. Prikaz rezultata upita

JOIN za spajanje tabela

Primeri: JOIN za spajanje tabela

Vežba: JOIN za spajanje tabela

02

FORMIRANJE UPITA NAD VIŠE TABELA KORIŠĆENJEM EQUIJOIN-A

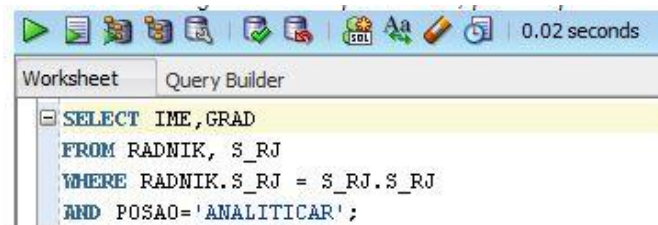
JOIN povezuje n-torke različitih tabela korišćenjem zajedničkih atributa i operatora spajanja =

JOIN povezuje n-torke različitih tabela korišćenjem zajedničkih atributa, odnosno atributa definisanih nad istim domenima. To su primarni ključ jedne tabele sa sekundarnim ključem druge između kojih obavezno postoji referencijalni integritet (npr. na slici 3 atribut S_RJ koji predstavlja primarni ključ tabele S_RJ se spaja sa atributom S_RJ u tabeli RADNIK koji predstavlja strani ključ te tabele). U primeru na slici 3 To je uobičajeni način povezivanja tabela relacione baze podataka.

Karakteristika EQUIJOIN-a je da se u WHERE klauzuli javlja uslov spajanja. **Pored potpune specifikacije naziva atributa po kojima se vrši spajanje tabela RADNIK i RJ navodi se i operator spajanja, u ovom slučaju jednakost (=).** Spajanje tabela moguće je vršiti u odnosu na bilo koji operator poređenja.

Primer: Za svakog analitičara prikaži ime, posao i podatke o RJ-u kojoj radi (slika 3.)

Karakteristično za ovaj upit je da se prvi put u SELECT listi atributa daje potpuna specifikacija naziva za attribute S_RJ tabele RADNIK i S_RJ tabele RJ. **Potpuna specifikacija naziva atributa sastoji se od naziva tabele, tačke i naziva kolone.** Ona se mora navoditi ukoliko naziv atributa (kolone) nije jedinstven u bazi podataka, tj. ukoliko se isti naziv koristi za kolone različitih tabela.



```
SELECT IME, GRAD
FROM RADNIK, S_RJ
WHERE RADNIK.S_RJ = S_RJ.S_RJ
AND POSAO='ANALITICAR';
```

Izlaz iz ove SELECT naredbe glasi:

IME	GRAD
STEFAN	BEOGRAD
STEFAN	NIS
STEFAN	BOR
STEFAN	NOVI SAD
JELENA	BEOGRAD
JELENA	NIS
JELENA	BOR
JELENA	NOVI SAD
JANKO	BEOGRAD
JANKO	NIS
JANKO	BOR
JANKO	NOVI SAD
DEJAN	BEOGRAD
DEJAN	NIS
DEJAN	BOR
DEJAN	NOVI SAD

Slika 3. Naredba SELECT i njen izlaz za prethodni primer

EQUIJOIN ZA SPAJANJE VIŠE TABELA: PRIMER 1

Primerom pokazano spajanje tabela **LOKACIJA**, **ODELJENJE**, **RADNIK**

Neka su date sledeće relacije:

LOKACIJA (Sifra_lokacije, Naziv_lokacije)

ODELJENJE (Sifra_odeljenja, Ime_odeljenja, *Sifra_lokacije*)

RADNIK (Sifra_radnika, Ime, Prezime, Zarada, *Sifra_odeljenja*)

Kreirati izveštaj sa kolonama **Naziv lokacije**, **Ime odeljenja**, **Radnik** (Prezime i ime) i **Zarada** na osnovu tabela **RADNIK**, **ODELJENJE** i **LOKACIJA** za sve radnike koji se nalaze na lokaciji **DALLAS**, **CHICAGO** ili **BOSTON** i čija je zarada veća od dvostruke vrednosti minimalne zarade. Izveštaj sortirati po koloni Naziv lokacije, po imenu odeljenja pa po opadajućoj vrednosti zarade.

```
SELECT NAZIV_LOKACIJE, IME||' '||PREZIME, ZARADA
FROM RADNIK, ODELJENJE, LOKACIJA
WHERE ODELJENJE.SIFRA_LOKACIJE =
LOKACIJA.SIFRA_LOKACIJE AND
RADNIK.SIFRA_ODELJENJA =
ODELJENJE.SIFRA_ODELJENJA AND
ZARADA > (SELECT 2*MIN(ZARADA) FROM RADNIK)
ORDER BY NAZIV_LOKACIJE, IME||' '||PREZIME, ZARADA
```

NAZIV_LOKACIJE	IME	PREZIME	ZARADA
CHICAGO	SOMMERS	DENISE	1850
CHICAGO	BLAKE	MARION	2850
CHICAGO	ALLEN	KEVIN	1600
DALLAS	LEWIS	RICHARD	1800
DALLAS	FORD	JENNIFER	3000
DALLAS	SCOTT	DONALD	3000
DALLAS	JONES	TERRY	2975
DALLAS	DENNIS	LYNN	2750

Slika 8. Izlaz iz naredbe SELECT za prethodni primer

EQUIJOIN ZA SPAJANJE VIŠE TABELA: PRIMER 2

*Primerom pokazano spajanje tabela **ODELJENJE** i **RADNIK***

Neka su date relacije:

ODELJENJE (Sifra_odeljenja, Ime_odeljenja)

RADNIK (Sifra_radnika, Ime, Prezime, Zarada, Sifra_odeljenja)

Kreirati izveštaj sa kolonama **Ime odeljenja** , **Prosečna zarada** i **Zbir zarada** na osnovu tabela **RADNIK** i **ODELJENJE** za sve odeljenja čija je prosečna zarada veća od 115 % prosečne zarade svih radnika.

```
select o.ime, avg(zarada) prosek, sum(zarada) suma  
from radnik r, odeljenje o  
where r.sifra_odeljenja = o.sifra_odeljenja  
group by o.ime_odeljenja  
having avg(zarada) > (select avg(zarada) * 1.15 from radnik )
```

IME	PROSEK	SUMA
ACCOUNTING	2916.6667	8750
RESEARCH	2283.3333	20550

Slika 9. Izlaz iz naredbe SELECT za prethodni primer

EQUIJOIN ZA SPAJANJE VIŠE TABELA: PRIMER 3

*Primerom pokazano spajanje tabela **RADNIK** i **POS AO***

Neka su date relacije:

RADNIK (Sifra_radnika, Ime, Prezime, Zarada, *Sifra_posla*)

POS AO (Sifra_posla, Funkcija)

Kreirati izveštaj sa kolonama **Funkcija**, **Ime i prezime** i **Zarada** na osnovu tabela **RADNIK** i **POS AO**. Prikazati samo one radnike koji imaju Funkciju **ANALYST** ili **SALESPERSON** i čija je zarada veća od 85% prosečne zarade. Podatke sortirati po koloni **Funkcija**, pa po koloni **Zarada** u opadajućem redosledu.

```
select funkcija, prezime||' '||ime "Prezime i ime", zarada
from radnik r, posao p
where r.sifra_posla = p.sifra_posla
and funkcija IN ( 'ANALYST', 'SALESPERSON') and
zarada > (select AVG(zarada)*0.85 from radnik)
order by funkcija, zarada desc
```


Vežba: JOIN za spajanje tabela

Vežba: JOIN za spajanje tabela

02

LEFT I RIGHT JOIN

Pre svega potrebno je objasniti LEFT i RIGHT JOIN.

LEFT JOIN daje kompletni skup zapisa iz tabel A sa odgovarajućim podacima iz tabele B.

Ukoliko nema podudaranja, desna strana će sadržati NULL vrednosti.

RIGHT JOIN daje kompletni skup zapisa iz tabele B sa odgovarajućim podacima iz tabele A.

Ukoliko nema podudaranja leva strana će sadržati NULL vrednosti.

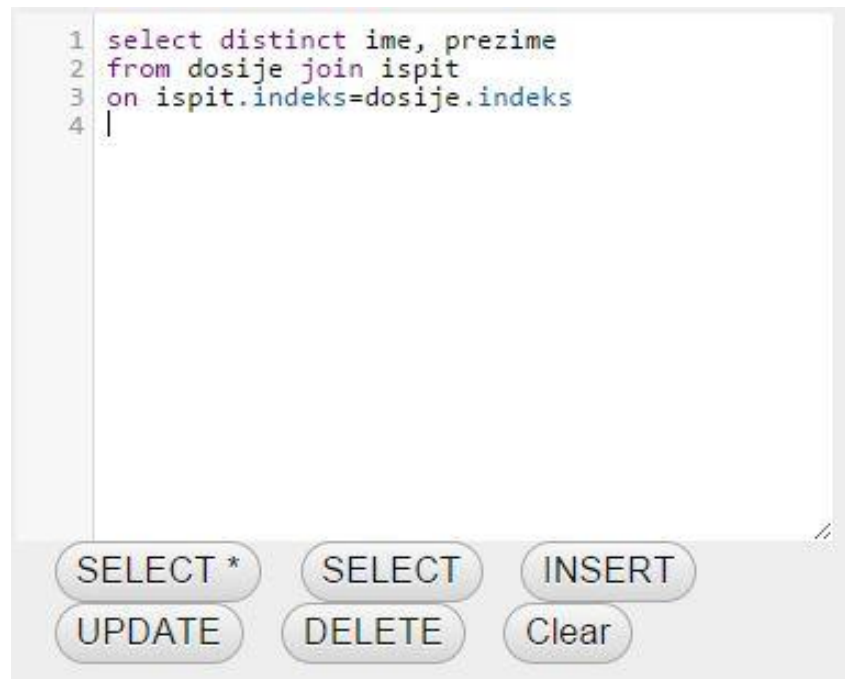
PRIMER 6.

Primer korišćenja JOIN operatora

Primer 6.

Prikazati ime i prezime svih studenata koji su položili bar jedan predmet.

```
1 select distinct ime, prezime
2 from dosije join ispit
3 on ispit.indeks=dosije.indeks
4 |
```



Slika-16. Prikaz upita korišćenjem JOIN operatora

Number of rows: 25 Filter rows: Search this table

+ Options

ime	prezime
Milos	Peric
Marijana	Savkovic
Sanja	Terzic
Nikola	Vukovic
Ljubica	Savkovic
Zorica	Miladinovic

Number of rows: 25 Filter rows: Search this table

Slika-17. Prikaz rezultata upita

PRIMER 7.

Primer korišćenja LEFT JOIN operatora

Primer 7.

Prikazati ime i prezime svih studenata bez obzira da li su položili neki predmet. Potrebno je koristiti LEFT JOIN operator.

```
1 select distinct ime, prezime
2 from dosije left join ispit
3 on ispit.indeks=dosije.indeks
4
```

SELECT * SELECT INSERT
UPDATE DELETE Clear

Slika-18. Prikaz upita

Number of rows: 25 Filter rows: Search this table

+ Options

ime	prezime
Milos	Peric
Marijana	Savkovic
Sanja	Terzic
Nikola	Vukovic
Ljubica	Savkovic
Zorica	Miladinovic
Milena	Stankovic

Number of rows: 25 Filter rows: Search this table

Slika-19. Prikaz rezultata upita

PRIMER 8.

Primer korišćenja JOIN operatora

Primer 8.

Izdvojiti broj indeksa i nazive svih predmeta koje je student polagao 2011 godine. Rezultat urediti prema broju indeksa, rastuće.

```
1 select indeks, naziv
2 from ispit i join predmet p
3 on i.id_predmeta = p.id_predmeta
4 where i.godina_roka =2011
5 order by indeks;
6
```

SELECT * SELECT INSERT
UPDATE DELETE Clear

Slika-20. Prikaz upita

+ Options

indeks	▲ 1	naziv
20100021		Matematika 1
20100021		Racunarske mreze
20100021		Programiranje 1
20100021		Engleski jezik 1
20100022		Matematika 1
20100022		Racunarske mreze
20100022		Programiranje 1
20100023		Matematika 1
20100023		Racunarske mreze
20100023		Programiranje 1
20100023		Engleski jezik 1
20100024		Matematika 1
20100024		Racunarske mreze
20100024		Programiranje 1
20100024		Engleski jezik 1
20100025		Matematika 1
20100025		Programiranje 1
20100026		Matematika 1
20100026		Racunarske mreze
20100026		Programiranje 1
20100026		Engleski jezik 1

Slika-21. Prikaz rezultata upita

PRIMER 9.

Primer spajanja tri tabele JOIN operatorima

Primer 9.

Izdvojiti brojeve indeksa, imena i prezimena studenta koji su polagali predmet koji nosi 5 bodova.

```
1 select d.indeks, d.ime, d.prezime
2 from dosije d join ispit i on
3   d.indeks=i.indeks
4   join predmet p on
5   i.id_predmeta=p.id_predmeta
6 where p.bodovi=5 ;
```

SELECT * SELECT INSERT
UPDATE DELETE Clear

Slika-22. Prikaz upita

Number of rows: 25 Filter rows: Search this table

+ Options

indeks	ime	prezime
20100021	Milos	Peric
20100023	Sanja	Terzic
20100024	Nikola	Vukovic
20100026	Zorica	Miladinovic

Number of rows: 25 Filter rows: Search this table

Slika-23. Prikaz rezultata upita

PRIMER 10.

Primer koji koristi agregatne funkcije, group by i having klauzule

Primer 10.

Za svakog studenta koji je skupio barem 20 bodova prikazati ukupan broj skupljenih bodova.

```
1 select indeks, sum(bodovi) ukupno
2 from ispit i join predmet p
3 on i.id_predmeta = p.id_predmeta
4 where ocena > 5
5 group by indeks
6 having sum(bodovi) >= 20
```

SELECT * SELECT INSERT
UPDATE DELETE Clear

Slika-24. Prikaz upita

Number of rows: 25 Filter rows: Search this table

+ Options

indeks	ukupno
20100021	25
20100023	20
20100024	25
20100026	25

Number of rows: 25 Filter rows: Search this table

Slika-25. Prikaz rezultatat prethodnog upita

Zaključak

ZAKLJUČAK

Šta smo naučili u ovoj lekciji?

U ovoj lekciji je kroz veliki broj primera prikazan način spajanja više tabela. Posebno su objašnjeni primeri za spajanje tabela korišćenjem podupita, naznačena ograničenja koja postoje u primeni ovakvog načina spajanja i posebno prikazani primeri za primenu različitih vrsta JOIN-a kao što su: CARTESIAN JOIN, EQUIJOIN, OUTER JOIN i SELF JOIN.