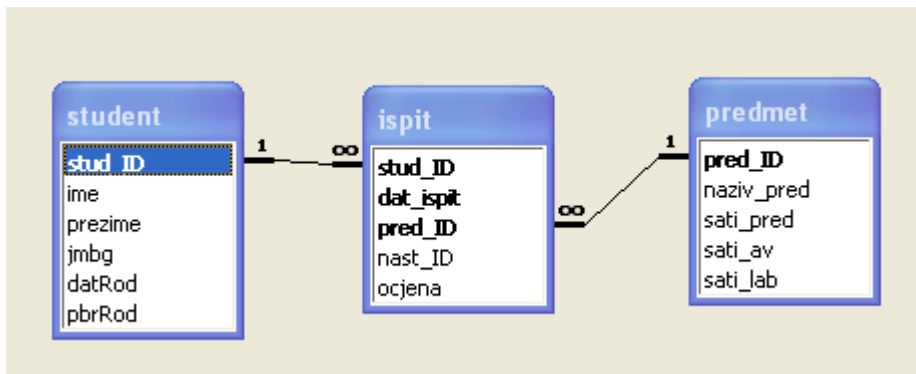


PRIMJERI SQL

Suzana Divić

PRIMJERI SQL	1
1. OSNOVE – PROJEKCIJA, SELEKCIJA, SPAJANJE TABLICA	2
1.1. PROJEKCIJA	2
1.2. SELEKCIJA	3
1.3. SELEKCIJA I PROJEKCIJA	4
1.4. SPAJANJE (JOIN)	5
1.5. SPAJANJE (JOIN) - ILUSTRACIJA RAZLIKE INNER I LEFT/RIGHT OUTER SPAJANJA	7
1.6. PRODUKT	8
1.7. UNIJA	9
2. PRIMJERI KORIŠTENJA IZRAZA I FUNKCIJA U SELECT DIJELU UPITA NAD JEDNOM TABLICOM	10
3. PRIMJERI KORIŠTENJA IZRAZA I FUNKCIJA U WHERE DIJELU UPITA NAD JEDNOM TABLICOM	11
4. PRIMJER - RING STRUKTURA	11
5. DODAVANJE PODATAKA U TABLICU – NAREDBA INSERT	12
6. AŽURIRANJE PODATAKA – NAREDBA UPDATE	14
7. BRISANJE PODATAKA – NAREDBA DELETE	15
8. PRIMJER - NAD BAZOM PRIKAZANOM NA SLICI NAPRAVITI SQL UPITE	16

1. Osnove – projekcija, selekcija, spajanje tablica



1.1. Projekcija

Ukoliko želimo vratiti sve atribute nekog entiteta, možemo korištenjem znaka * koji označava sve atribute:

```
select * from predmet;
select predmet.* from predmet;
select p.* from predmet p;
```

ili možemo nabrojiti svaki od atributa

```
select pred_id, naziv_pred, sati_pred, sati_av, sati_lab from predmet;
select predmet.pred_id, predmet.naziv_pred, predmet.sati_pred, predmet.sati_av,
predmet.sati_lab from predmet;
select p.pred_id, p.naziv_pred, p.sati_pred, p.sati_av, p.sati_lab from predmet p;
```

Kad želimo izlistati samo neke atribute (stupce tablice), govorimo o projekciji.

Primjer 1:

```
Select ime, prezime, jmbg from student;
```

Za sve studente (odn. sve retke upisane u tablicu "student"), upit vraca samo oznacene atribute

	stud_ID	ime	prezime	jmbg	datRod	pbrRod
+	10001	Ana	Jurić	1106980401236	11.06.1980	10000
+	10234	Mario	Klarić	0311981302215	22.05.1981	31000
+	11010	Ivan	Korkut	0209979502247	02.09.1979	21000
+	11234	Ante	Jurić	0412979214665	04.12.1979	48000
+	12244	Marija	Knjaz	1109980325234	11.09.1980	51000
+	13475	Kristina	Kušan	0202980122201	25.02.1980	31000
+	21112	Ivan	Županović	0311981376432	18.02.1982	31000
+	21331	Petar	Šoljić	1211981879356	21.05.1981	22000
+	22222	Mirta	Lončar	2709980127782	05.11.1987	52000
+	25367	Davor	Dujmić	2504982561131	24.02.1987	10000
+	27783	Franjo	Mušnjak	1105981200211	21.05.1981	33000
+	34211	Petra	Kreža	0506978657631	05.11.1986	10000
+	34678	Jurica	Domanovac	0606978543121	05.06.1989	10000
+	34879	Hrvoje	Petrović	1708982012003	17.08.1982	10000
+	35648	Igor	Ivčić	0203976370211	02.03.1976	21000
+	35765	Ana	Antolić	0808980452211	20.02.1980	32000
+	36781	Mislav	Bistričić	0311981356685	24.02.1985	20000
+	47678	Ljubica	Šaput	2412982700121	24.12.1982	44000
+	56778	Boris	Kundera	0203976443523	02.03.1976	23000
+	58445	Jelena	Župan	0209979122134	02.09.1979	21000
	0					0

Primjer 2:

Select pred_id, naziv_pred from predmet;

Za sve predmete (odn. sve retke upisane u tablicu "predmet"), upit vraća samo označene atribute

	pred_ID	naziv_pred	sati_pred	sati_av	sati_lab
+	1001	matematika	4	3	0
+	1002	baze podataka	3	2	2
+	1003	primjena računala	1	0	2
+	1004	operacijski sustavi	2	2	2
+	1005	programiranje	2	2	1
+	1006	računala i procesi	3	2	1
+	1007	elektronika	3	1	2
+	1008	algoritmi i strukture podataka	2	1	1
+	1009	računalna grafika	2	1	1
+	1010	programske paradigme i jezici	2	2	1
	0		0	0	0

1.2. Selekcija

Govorimo o selekciji kad želimo dohvatiti samo neke zapise iz tablice (samo one koji zadovoljavaju neki zadani kriterij)!

Primjer 1:

Select * from student where prezime = "Jurić";

Od svih zapisa u tablici student, upit će vratiti samo one označene

	stud_id	ime	prezime	jmbg	datRod	ponRod
+	35765	Ana	Antolić	0808980452211	20.02.1980	32000
+	36781	Mislav	Bistričić	0311981356685	24.02.1985	20000
+	34678	Jurica	Domanovac	0606978543121	05.06.1989	10000
+	25367	Davor	Dujmić	2504982561131	24.02.1987	10000
+	35648	Igor	Ivčić	0203976370211	02.03.1976	21000
+	11234	Ante	Jurić	0412979214665	04.12.1979	48000
+	10001	Ana	Jurić	1106980401236	11.06.1980	10000
+	10234	Mario	Klarić	0311981302215	22.05.1981	31000
+	12244	Marija	Knjaz	1109980325234	11.09.1980	51000
+	11010	Ivan	Korkut	0209979502247	02.09.1979	21000
+	34211	Petra	Krleža	0506978657631	05.11.1986	10000
+	56778	Boris	Kundera	0203976443523	02.03.1976	23000
+	13475	Kristina	Kušan	0202980122201	25.02.1980	31000
+	22222	Mirta	Lončar	2709980127782	05.11.1987	52000
+	27783	Franjo	Mušnjak	1105981200211	21.05.1981	33000
+	34879	Hrvoje	Petrović	1708982012003	17.08.1982	10000
+	47678	Ljubica	Šaput	2412982700121	24.12.1982	44000
+	21331	Petar	Šoljić	1211981879356	21.05.1981	22000
+	58445	Jelena	Župan	0209979122134	02.09.1979	21000
+	21112	Ivan	Županović	0311981376432	18.02.1982	31000

1.3. Selekcija i projekcija

→ kad želimo izlistati samo neke zapise iz tablice (samo one koji zadovoljavaju neki zadani kriterij), ali ne sve attribute već samo neke!

Primjer 1:

Select ime, prezime, jmbg from student where prezime = "Jurić";

Promotrimo sliku iz prethodnog primjera – ovdje se vrši ista selekcija kao u prošlom primjeru. Izdvajaju se 2 zapisa koja odgovaraju uvjetu da je za njih prezime Jurić. Za ta dva zapisa ispisat će se samo attribute ime, prezime i jmbg:

	ime	prezime	jmbg
	Ana	Jurić	1106980401236
	Ante	Jurić	0412979214665

Primjer 2:

**Select ime, prezime, jmbg from student
where prezime = "Jurić" and year(datRod) = 1985;**

Radi se kao u prethodnom primjeru, ali uz dodatak kriterija - želimo vratiti samo one koji su rođeni 1985. godine

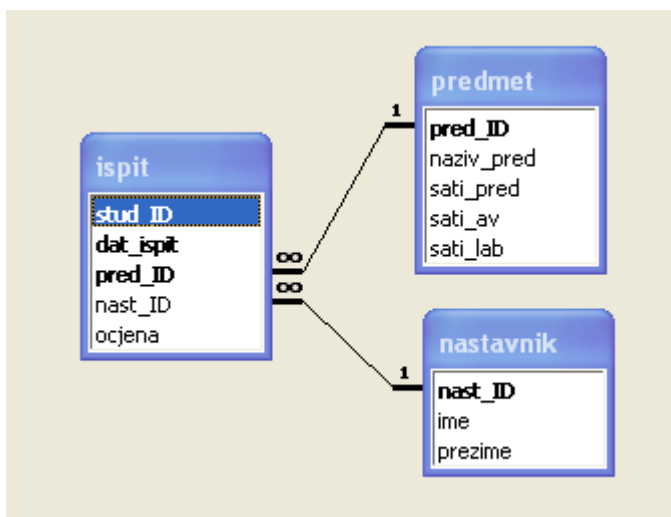
	stud_id	ime	prezime	jmbg	datRod	postro
+	35765	Ana	Antolić	0808980452211	20.02.1980	32000
+	36781	Mislav	Bistričić	0311981356685	24.02.1985	20000
+	34678	Jurica	Domanovac	0606978543121	05.06.1989	10000
+	25367	Davor	Dujmić	2504982561131	24.02.1987	10000
+	35648	Igor	Ivčić	0203976370211	02.03.1976	21000
+	11234	Ante	Jurić	0412979214665	04.12.1979	48000
+	10001	Ana	Jurić	1106980401236	11.06.1980	10000
+	10234	Mario	Klarić	0311981302215	22.05.1981	31000
+	12244	Marija	Knjaz	1109980325234	11.09.1980	51000
+	11010	Ivan	Korkut	0209979502247	02.09.1979	21000
+	34211	Petra	Krleža	0506978657631	05.11.1986	10000
+	56778	Boris	Kundera	0203976443523	02.03.1976	23000
+	13475	Kristina	Kušan	0202980122201	25.02.1980	31000
+	22222	Mirta	Lončar	2709980127782	05.11.1987	52000
+	27783	Franjo	Mušnjak	1105981200211	21.05.1981	33000
+	34879	Hrvoje	Petrović	1708982012003	17.08.1982	10000
+	47678	Ljubica	Šaput	2412982700121	24.12.1982	44000
+	21331	Petar	Šoljić	1211981879356	21.05.1981	22000
+	58445	Jelena	Župan	0209979122134	02.09.1979	21000
+	21112	Ivan	Županović	0311981376432	18.02.1982	31000

Za Antu Jurića datRod = '04.12.1979' → ne zadovoljava uvjet $\text{year}(\text{datRod}) = 1980$

Zato je rezultat upita ovo:

	ime	prezime	jmbg
▶	Ana	Jurić	1106980401236

1.4. Spajanje (join)



Povezivanje tablica (join) koristimo da bismo dohvatili podatke koji se nalaze u različitim tablicama, ali su logički povezani.

Primjer 1:

Dohvatiti sve podatke nastavnika koji su držali ispit zajedno s nazivima predmeta koje su ispitivali.

- Želimo selectirati atribute iz 2 entitea koja nisu direktno povezana već 'između njih' postoji tablica veze (ispit) => moramo povezati sve 3 tablice da bismo ispravno povezali podatke o nastavniku i predmetu.
- Iz definicije zadatka vidi se popis atributa koje treba dohvatiti:
select nastavnik.*, predmet.naziv_Pred (A)

- u tablici ispit je veza predmeta i nastavnika. Dakle moramo uključiti i tu tablicu da bismo došli do odgovora na pitanje koji nastavnik je povezan uz koji predmet.
from nastavnik n, ispit i, predmet p (B)

- Zbog postojeće veze (FK) očekujemo da u tablici ispit ne može biti unesen ključ nepostojećeg nastavnika ili predmeta (u protivnom bi bio narušen referencijalni integritet baze. RDBMS ne bi dopustio upisivanje takvog zapisa)
- ukoliko ima redaka u tablici ispit za koje je nastavnik NULL, takvi nas zapisi u ovom slučaju ne zanimaju.
- Očekujemo da su nam u tablici predmeta podaci o svim predmetima, te u tablici nastavnika podaci o svim nastavnicima.
- Ne zanimaju nas niti nastavnici niti predmeti za koje ne postoji zapis u tablici ispit, jer ne odgovaraju na naše pitanje o povezanosti profesora i predmeta
where n.nast_id = i.nast_id and i.pred_id = p.pred_id (C)

Dakle:

```
select n.*, p.naziv_pred
from nastavnik n, ispit i, predmet p
where n.nast_id = i.nast_id and i.pred_id = p.pred_id;
```

ili

```
select n.*, p.naziv_pred
from predmet p inner join (nastavnik n inner join ispit i on n.nast_id = i.nast_id) on
p.pred_id = i.pred_id;
```

ili

```
select n.*, p.naziv_pred
from (nastavnik n inner join ispit i on n.nast_id = i.nast_id) inner join predmet p on
p.pred_id = i.pred_id;
```

ili

```
select n.*, p.naziv_pred
from nastavnik n inner join (ispit i inner join predmet p on p.pred_id = i.pred_id) on
n.nast_id = i.nast_id;          itd... (varijacije redoslijeda)
```

Ne može ovo: ... from ispit INNER JOIN (predmet INNER JOIN nastavnik ...)
JER NEMA KLJUČA KOJI SPAJA PREDMET I NASTAVNIK!!!!

Ovi upiti nam u pravilu daju 'duplice' jer tablica ispit ima i vezu prema studentu – dakle ukoliko je nastavnik Pero Perić držao ispit iz baza podataka 13.2.2010 na kojeg je izašlo 15 studenata, u tablici ispit biti će barem 15 zapisa koji će odgovarati tom

nastavniku i predmetu (i više ako je isti nastavnik iz istog predmeta držao ispit i na neke druge datume).

Dodamo li u sve ove upite nakon SELECT još i ključnu riječ DISTINCT, dupli zapisi se neće pojaviti:

```
select DISTINCT n.*, p.naziv_pred  
from nastavnik n, ispit i, predmet p  
where n.nast_id = i.nast_id and i.pred_id = p.pred_id
```

ili

```
select DISTINCT n.*, p.naziv_pred  
from predmet p inner join (nastavnik n inner join ispit i on n.nast_id = i.nast_id) on  
p.pred_id = i.pred_id
```

ili ...

Primjer 2:

Dohvatiti sve nastavnike:

```
select * from nastavnik;
```

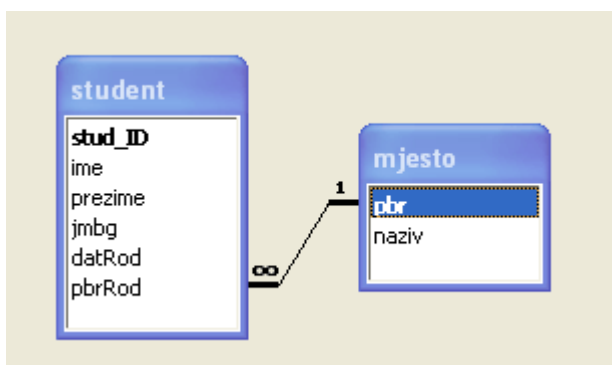
Za ovo nije potrebno spajati tablicu nastavnika s drugom tablicom!

Primjer 3:

Dohvatiti sve nastavnike koji su držali ispit

```
select nastavnik.* from nastavnik INNER JOIN ispit on nastavnik.nast_id = ispit.nast_id;
```

1.5. Spajanje (Join) - Ilustracija razlike INNER i Left/Right OUTER spajanja



Pretpostavimo da mjesto rođenja nije obavezan podatak za studenta. Tamo gdje je podatak nepoznat, vrijednost pbrRod je NULL. Tamo gdje podatak imamo, unesen je ključ iz tablice mjesto.

Inner join tablica student i mjesto neće vratiti zapise o studentima za koje nema podatka o pbrRod u tablici student:

```
SELECT student.ime, student.prezime, mjesto.naziv  
FROM mjesto INNER JOIN student ON mjesto.pbr = student.pbrRod;
```

Ili

```
SELECT student.ime, student.prezime, mjesto.naziv  
FROM mjesto, student WHERE mjesto.pbr = student.pbrRod;
```

Ukoliko želimo dohvatiti podatke ime, prezime, mjesto rođenja o svim studentima ali tako da se naziv mjesta rođenja ispiše za one studente za koje ima podatka, moramo koristiti lijevi ili desni outer join na način da se uzme sve podatke iz tablice student (student LEFT JOIN mjesto ili mjesto RIGHT JOIN student):

```
SELECT student.ime, student.prezime, mjesto.naziv  
FROM mjesto RIGHT JOIN student ON mjesto.pbr = student.pbrRod;
```

```
SELECT student.ime, student.prezime, mjesto.naziv  
FROM student LEFT JOIN mjesto ON mjesto.pbr = student.pbrRod;
```

- Kada koristiti LEFT a kada RIGHT JOIN?
 - nema veze s tim što je na dijagramu nacrtano lijevo a što desno!!!
 - Ima veze s tim koja tablica je navedena prva (ona se smatra LIJEVOM tablicom). Pitanje je da li želimo sve podatke iz nje (lijeve) ili sve podatke iz one druge (desne) tablice. (U gornjem primjeru želimo sve podatke iz tablice student.)

1.6. Produkt

Ispisati za svakog studenta koji bi sve predmet mogao slušati: svakog studenta želimo povezati sa svakim predmetom!

```
SELECT student.ime, student.prezime, predmet.naziv_pred  
FROM student, predmet
```

Obzirom da nemamo naveden ključ spajanja tablica, spaja se svaki zapis iz tablice student sa svakim zapisom iz tablice predmet.

Ukoliko imamo 2 studenta: Pero Perić i Mara Marić
i 3 predmeta: matematika, fizika i baze podataka

Rezultat upita će biti 2*3 zapisa:

Pero Perić, matematika

Pero Perić, fizika
Pero Perić, baze podataka
Mara Marić, matematika
Mara Marić, fizika
Mara Marić, baze podataka

Da smo imali 100 studenata i 20 predmeta, upit bi vratio $100 \cdot 20 = 2000$ zapisa

1.7. Unija

Primjer 1:

Imamo 2 tablice:

Muskarci	= (muskarac_ID, ime, prezime, jmbg)
zene	= (zena_ID, ime, prezime, jmbg, datRod)

Želimo dohvatiti sve osobe:

```
select ime, prezime, jmbg from muskarci
union
select ime, prezime, jmbg from zene;
```

Zamislamo da je greškom Pero Perić unesen i u tablicu muškaraca i u tablicu žena. Gornji upit vratit će podatak o Peri Periću samo jednom.

Podatak o Peri Periću biti će izlistan dva puta ukoliko umjesto ključne riječi UNION napišemo UNION ALL:

```
select * from muskarci
union all
select zena_ID, ime, prezime, jmbg from zene;
```

Primjer 2:

Zadatak kao i u primjeru 1. ali želimo imati i informaciju o spolu osobe

```
select ime, prezime, jmbg, 'M' as spol from muskarci
union
select ime, prezime, jmbg, 'Ž' from zene;
```

Za Peru Perića koji je pogreškom unesen u obje tablice, ovdje će i operator union vratiti 2 zapisa jer zapisi koje dobijemo iz prvog i drugog selecta nisu isti – razlikuju se u vrijednosti atributa spol.

2. Primjeri korištenja izraza i funkcija u select dijelu upita nad jednom tablicom

Primjer1:

Tablica zaposlenika:

	ID_zaposlenog	placa	b_placa_god	bonus
	1	10000	12	3000
	2	15000	13	20000
	3	5000	12	0
	4	6000	12	1000

```
SELECT id_zaposlenog, placa * b_placa_god + bonus as total_godisnje  
from zaposlenici
```

	id_zaposlenog	total_godisnje
	1	123000
	2	215000
	3	60000
	4	73000

Primjer2:

Tablica zaposlenika:

	ID_zaposlenog	datum_zaposl
	1	01.11.1995
	2	15.02.2004
	3	11.01.2006
	4	07.07.2008
	(AutoNumber)	

```
SELECT id_zaposlenog, datum_zaposl,  
       year(datum_zaposl) as godina_zaposlenja,  
       round((date() - datum_zaposl)/365) as godina_zaposlen  
from zaposlenici
```

id_zaposlenog	datum_zaposl	godina_zaposlenja	godina_zaposlen
1	01.11.1995	1995	15
2	15.02.2004	2004	7
3	11.01.2006	2006	5
4	07.07.2008	2008	2

Objašnjenje:

Funkcija round zaokružuje decimalni broj na cjelobrojnu vrijednost.

$Round(1.2) = 1$
 $Round(1.834) = 2$

Funkcija date() vraća sistemski datum u formatu datetime

Razlika 2 datuma d2-d1 je broj dana između d1 i d2.

P.S. primjer je za ilustraciju. Točniji rezultat za broj godina koliko je zaposlenik zaposlen bismo dobili pomoću izraza **year(date()) – year(datum_zaposl)**

3. *Primjeri korištenja izraza i funkcija u where dijelu upita nad jednom tablicom*

Primjer 1:

Želimo dohvatiti samo one studente koji su rođeni 1985. godine

```
select * from student where year(datRodj) = 1985;
```

Primjer 2:

Tablica zaposlenika:

	ID_zaposlenog	placa	b_placa_god	bonus
	1	10000	12	3000
	2	15000	13	20000
	3	5000	12	0
	4	6000	12	1000

Želimo dohvatiti samo zaposlenike koji zarađuju preko 100.000 kuna godišnje:

```
SELECT *  
from zaposlenici  
where (placa * b_placa_god + bonus) > 100000;
```

4. *Primjer - Ring struktura*

Tablica povezana sama sa sobom...

Tablica na nekom atributu ima definiran strani ključ koji 'pokazuje' na primarni ključ iste tablice

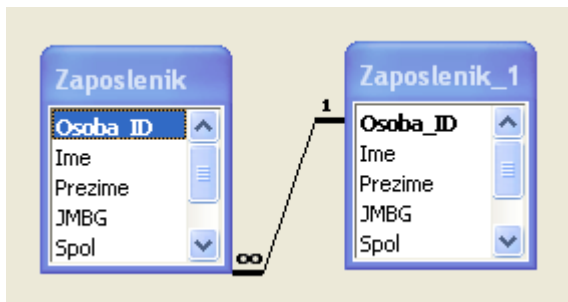
Primjer:

- Tablica zaposlenih, jedan od atributa svakog zaposlenika je informacija o tome tko mu je šef. 1 šef može imati više podređenih, ali svaki zaposlenik ima samo 1 šefa.

	Osoba_ID	Ime	Prezime	JMBG	Spol	Odjel	Sef_ID
+	1	Ana	Anić	111111111111	Ž	Kadrovska	1
+	2	Pero	Perić	111111222222	M	Kadrovska	1
+	3	Mate	Matić	2212221212334	M	Tehnika	3
+	4	Mirta	Mirtiće	3212734621471	Ž	Kadrovska	1
+	5	Mara	Marić	4238493242342	Ž	Tehnika	3
+	6	Ante	Antić	3249832058349	M	Tehnika	3
	(AutoNumber)						0

- Atribut sef_ID pokazuje na zapis o šefu koji se nalazi u istoj tablici

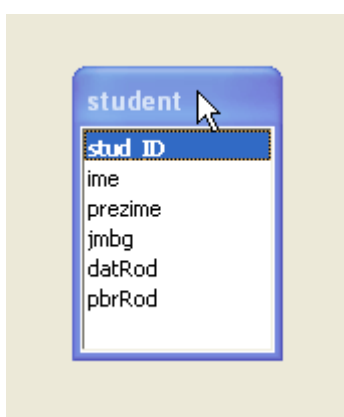
Na *Relationships* dijagramu u Accessu to izgleda ovako:



- Da bismo napisali upit koji koristi istu tablicu za 2 namjene (jednom da bismo dobili podatke o radniku a drugi put da bismo uz podatke o radniku dopisali podatke o njegovom šefu), trebamo koristiti alias naziv:

```
SELECT      Zaposlenik.Ime AS ime_zaposlenika,
            Zaposlenik.Prezime AS prezime_zaposlenika,
            Sef.Ime AS ime_nadredjenog,
            sef.Prezime AS prezime_nadredjenog
FROM Zaposlenik AS Sef INNER JOIN Zaposlenik
ON Sef.Osoba_ID=Zaposlenik.Sef_ID;
```

ime_zaposlenika	prezime_zaposlenik	ime_nadredjenog	prezime_nadredjenog
Ana	Anić	Ana	Anić
Pero	Perić	Ana	Anić
Mirta	Mirtiće	Ana	Anić
Mate	Matić	Mate	Matić
Mara	Marić	Mate	Matić
Ante	Antić	Mate	Matić



5. Dodavanje podataka u tablicu – naredba insert

Imamo tablicu student u kojoj su stud_id, ime i prezime obavezno podaci (required, odnosno not null).

Podatak o datumu i mjestu rođenja za studenta nije obavezan, kao ni jmbg.

Ne možemo ubaciti novi zapis (podatke za novog studenta) ukoliko ne upišemo sve obavezne podatke

Primjer 1:

Želimo u tablicu student dodati zapis o studentu Peri Periću iz Zagreba, rođenom 1.12.1987. godine

Možemo to napraviti na nekoliko načina:

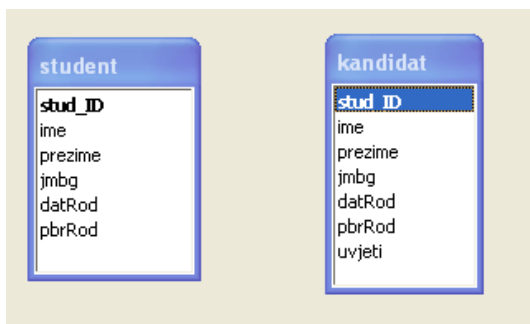
```
insert into student values (2010623, 'Pero', 'Perić', NULL, '01.12.1987', NULL);
```

```
insert into student(ime, stud_id, prezime) values ('Pero', 2010623, 'Perić');
```

Komentar – kad bi primarni ključ stud_id bio definiran kao sekvenca (Autonumber, COUNTER), ne bi bilo nužno u insert naredbi ubaciti vrijednost stud_id, već bi ga baza sama generirala prilikom upisivanja podataka.

Primjer 2:

Osim tablice student iz prethodnog primjera, imamo dodatnu tablicu kandidat koja ima sve atribute tablice student uz dodatni atribut uvjeti. Kandidati za koje uvjeti="Da" upisuju fakultet i postaju studenti. Želimo ih prebaciti u tablicu student.



```
insert into student
select stud_id, ime, prezime, jmbg, datRod,
pbrRod
from kandidat where uvjeti = 'Da';
```

Tablica studenata prije ubacivanja podataka:

	stud_ID	ime	prezime	jmbg	datRod	pbrRod
+	10001	Ana	Jurić	1106980401236	11.06.1980	10000
+	10234	Mario	Klarić	0311981302215	03.11.1981	31000
+	11010	Ivan	Korkut	0209979502247	02.09.1979	21000
+	11234	Ante	Jurić	0412979214665	04.12.1979	48000
+	12244	Marija	Knjaz	1109980325234	11.09.1980	51000
+	13475	Kristina	Kušan	0202980122201	02.02.1980	31000
+	21112	Ivan	Županović	0311981376432	03.11.1981	31000
+	21331	Petar	Šoljić	1211981879356	12.12.1981	22000
+	22222	Mirta	Lončar	2709980127782	27.09.1980	52000
+	25367	Davor	Dujmić	2504982561131	25.04.1982	10000
+	27783	Franjo	Mušnjak	1105981200211	11.05.1981	33000
+	34211	Petra	Križić	0506978657631	05.06.1978	10000
+	34678	Jurica	Domanovac	0606978543121	06.06.1978	10000
+	34879	Hrvoje	Petrović	1708982012003	17.08.1982	10000
+	35648	Igor	Ivčić	0203976370211	02.03.1976	21000
+	35765	Ana	Antolić	0808980452211	08.08.1980	32000
+	36781	Mislav	Bistrić	0311981356685	03.11.1981	20000
+	47678	Ljubica	Šaput	2412982700121	24.12.1982	44000
+	58445	Jelena	Župan	0209979122134	02.09.1979	
+	2010623	Pero	Perić		01.12.1987	

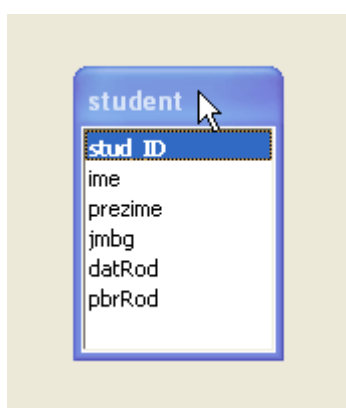
Kandidati:

	stud_ID	ime	prezime	jmbg	datRod	pbrRod	uvjeti
▶	536781	Mislav	Bilić	1311981356685		20000	Da
	510001	Anka	Matić	2106980401236	11.06.1980	10000	Da
	2010623	Pero	Petrić		01.12.1987		Ne
	558445	Jelena	Jelić	0209879122134	02.09.1979		Ne
	547678	Rade	Šuput	1412982700121	24.12.1982	44000	Ne
	510234	Marko	Jozić	1311981302215		31000	Ne

Tablica studenata nakon ubacivanja podataka sadrži i podatke o novim studentima:

	stud_ID	ime	prezime	jmbg	datRod	pbrRod
+	10001	Ana	Jurić	1106980401236	11.06.1980	10000
+	10234	Mario	Klarić	0311981302215	03.11.1981	31000
+	11010	Ivan	Korkut	0209979502247	02.09.1979	21000
+	11234	Ante	Jurić	0412979214666	04.12.1979	48000
+	12244	Marija	Knjaz	1109980325234	11.09.1980	51000
+	13475	Kristina	Kušan	0202980122201	02.02.1980	31000
+	21112	Ivan	Županović	0311981376432	03.11.1981	31000
+	21331	Petar	Šoljić	1211981879356	12.12.1981	22000
+	22222	Mirta	Lončar	2709980127782	27.09.1980	52000
+	25367	Davor	Dujmić	2504982561131	25.04.1982	10000
+	27783	Franjo	Mušnjak	1105981200211	11.05.1981	33000
+	34211	Petra	Krleža	0506978657631	05.06.1978	10000
+	34678	Jurica	Domanovac	0606978543121	06.06.1978	10000
+	34879	Hrvoje	Petrović	1708980212003	17.08.1982	10000
+	35648	Igor	Ivčić	0203976370211	02.03.1976	21000
+	35765	Ana	Antolić	0808980452211	08.08.1980	32000
+	36781	Mislav	Bistričić	0311981356685	03.11.1981	20000
+	47678	Ljubica	Šaput	2412982700121	24.12.1982	44000
+	58445	Jelena	Župan	0209979122134	02.09.1979	
▶	510001	Anka	Matić	2106980401236	11.06.1980	10000
+	536781	Mislav	Bilić	1311981356685		20000
+	2010623	Pero	Petrić		01.12.1987	
*	n					

6. Ažuriranje podataka – naredba update



Primjer 1

Imamo tablicu student u kojoj su stud_id, ime i prezime obavezno podaci (required, odnosno not null). Podatak o datumu i mjestu rođenja za studenta nije obavezan, kao ni jmbg.

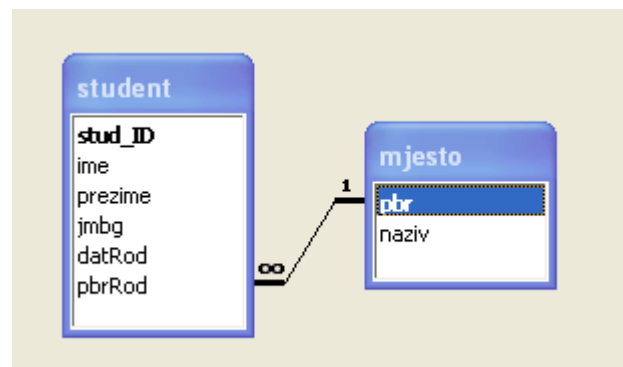
Svim studentima kojima je pbrRod = 10000 želimo postaviti prezime u Matić.

```
update student
set prezime = 'Matić'
where pbrRod = 10000;
```

Primjer 2

Želimo postaviti ime u „Bjelovarac“ svim studentima koji su rođeni u Bjelovaru.

```
update student
set ime = 'Bjelovarac'
where pbrRod in
(select pbr from mjesto where naziv = 'Bjelovar');
```

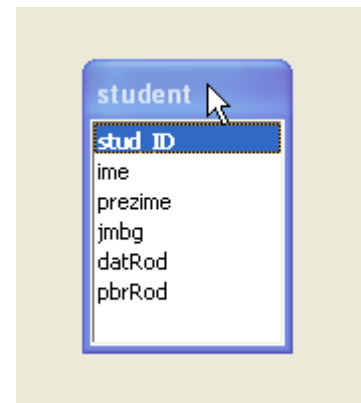


7. Brisanje podataka – naredba delete

Primjer 1

Želimo iz baze obrisati studenta sa šifrom studenta 12345 (kojem je stud_id = 12345).

```
delete from student
where stud_id = 12345;
```



Primjer 2

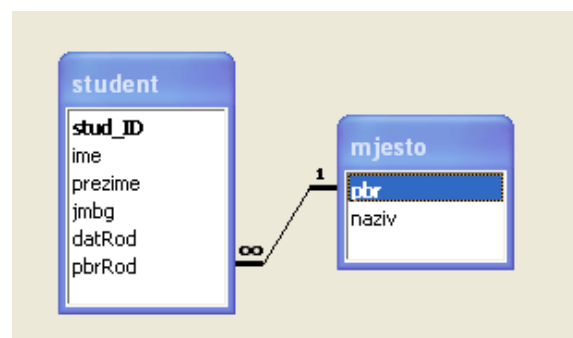
Želimo iz baze obrisati sve studente rođene prije 1950. godine

```
delete from student
where year(pbrRod) < 1950;
```

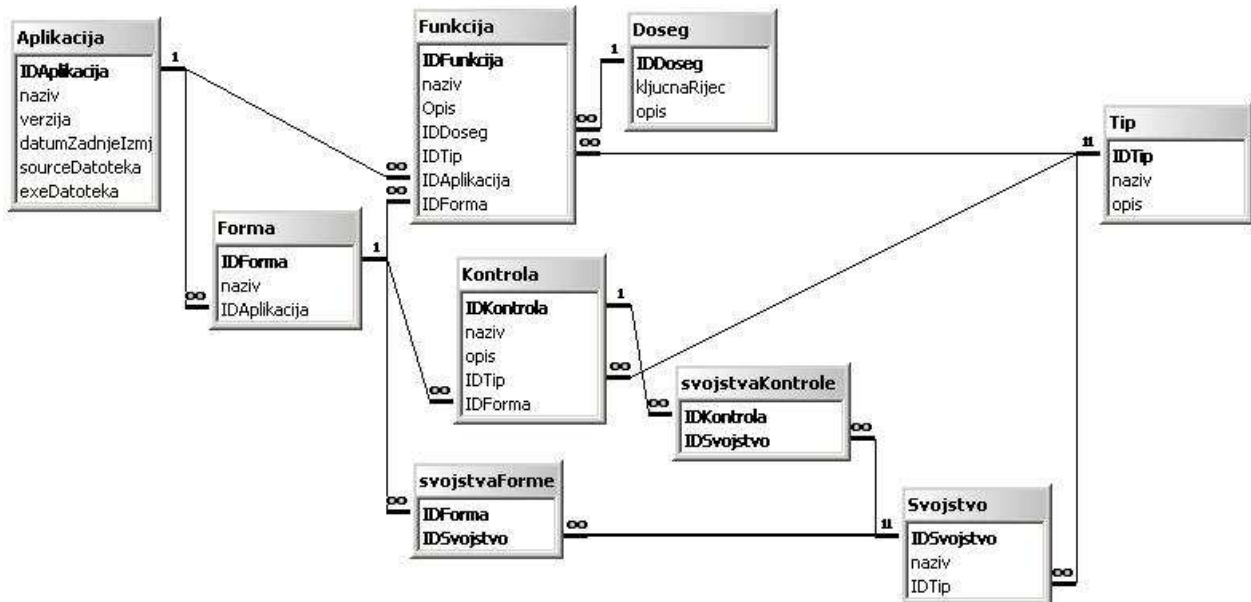
Primjer 3

Želimo obrisati sve studente iz Bjelovara.

```
delete from student
where pbrRod in
(select pbr from mjesto where naziv = 'Bjelovar');
```



8. Primjer - Nad bazom prikazanom na slici napraviti SQL upite



Uputa – iako baza izgleda kompleksno, kod rješavanja svakog od zadataka potrebno se fokusirati samo na jedan manji dio dijagrama.

Dobro pročitati zadatak, zaključiti koje tablice i veze je potrebno koristiti i promatrati samo taj dio dijagrama

a) Izlistati sve aplikacije (IDAplikacija i naziv iz tablice Aplikacija).

```
select IDAplikacija, naziv from aplikacija;
```

b) Izlistati popis aplikacija (sve podatke) s njihovih funkcijama (naziv i opis funkcije) uz pretpostavku da svaka aplikacija ima barem jednu funkciju.

```
select a.*, f.naziv, f.opis  
from aplikacija a INNER JOIN funkcija f on a.IDAplikacija = f.IDAplikacija;
```

c) Izlistati sve različite ključne riječi iz tablice Doseg

```
select DISTINCT kljucnaRijec  
from Doseg;
```

d) Ispisati nazive svih formi koje pripadaju aplikaciji „Dummy aplikacija“


```
select distinct Forma.naziv
from Forma inner join Aplikacija on Forma.IDAplikacija = Aplikacija.IDAplikacija
where aplikacija.naziv = „Dummy aplikacija“;
```

ili

```
select distinct Forma.naziv
from Forma, Aplikacija
where Forma.IDAplikacija = Aplikacija.IDAplikacija
and aplikacija.naziv = „Dummy aplikacija“;
```

ili

```
select naziv from Forma where idAplikacija IN (select A. idAplikacija from aplikacija A where
A.naziv = „Dummy aplikacija“);
```

e) Ispisati nazive svih formi

```
select naziv
from Forma;
```

f) Izlistati naziv i opis kontrola, te za svaku tip kontrole (naziv i opis)

```
select k.naziv, k.opis, t.naziv as tip_kontrole, t.opis as opis_tipa_kontrole
from kontrola k INNER JOIN tip t on k.IDtip = t.IDtip;
```

ili

```
select k.naziv, k.opis, t.naziv, t.opis
from kontrola k, tip t
where k.IDtip = t.IDtip;
```

g) Dohvatiti nazive tipova koji se koriste kao tipovi kontrole (jedan naziv ispisati samo jednom)

```
select distinct t.naziv
from tip t INNER JOIN kontrola k on t.IDTip = k.IDTip;
```

ili

```
select naziv from tip
where IDTip in (select IDTip from Kontrola);
```

ili

```
select distinct t.naziv
from Kontrola k, Tip t
```

where k.IDTip = t.IDTip;

h) Izlistati popis kontrola (naziv i opis) i njihovih svojstava (naziv svojstva).

```
select k.naziv, k.opis, s.naziv as naziv_Svojstva
from kontrola k INNER JOIN (svojstva s INNER JOIN svojstvaKontrole sk on s.IDSvojstvo
= sk.IDSvojstvo) on k.IDKontrola = sk.IDKontrola;
```

ili

```
select k.naziv, k.opis, s.naziv
from kontrola k, svojstva s, svojstvaKontrole sk
where s.IDSvojstvo = sk.IDSvojstvo and k.IDKontrola = sk.IDKontrola;
```

(gledamo samo dio dijagrama koji prikazuje tablice Kontrola, svojstvoKontrole, Svojstvo i veze među njima. Tablica svojstvoKontrole je tablica veze (više prema više) između entiteta Kontrola i entiteta Svojstvo)

i) Napisati naredbu kojom se tablici Aplikacija dodaje atribut datumKreiranja. Atribut nikada ne može biti prazan.

```
alter table aplikacija
add column datumKreiranja datetime NOT NULL;
```

j) Dohvatiti sve IDAplikacija i nazive aplikacija koje nisu mijenjane više od 2 godine, a sourceDatoteka im je nepoznata (nije upisana). Upit mora biti neovisan o vremenu izvođenja.

```
Select IDAplikacija, naziv
From Aplikacija
Where sourcedatoteka is null
and datumZadnjelzmj < dateserial(year(date())-2, month(date()), day(date()));
```

k) Dohvatiti sve forme zajedno s funkcijama koje im pripadaju, te odgovarajući tip funkcije. Na popisu trebaju biti sve forme (i one koje nemaju definiranu funkciju). Ispisati naziv forme, naziv funkcije i naziv odgovarajućeg tipa.

```
Select frm.IDForma, frm.naziv, fun.naziv, t.naziv
From forma frm left outer join (funkcija fun inner join tip t on fun.IDtip = t.IDtip) on
frm.IDForma = fun.IDForma;
```

- l) Napišite naredbe kojima bi se iz tablice svojstvo obrisala svojstva koja se ne koriste kao svojstva formi.

```
delete from svojstvo where IDSvojstvo not in (select IDSvojstvo from svojstvaForme);
```

- m) Za aplikaciju naziva „Dummy aplikacija“ postavite opis svih njezinih funkcija u „Dummy funkcija“.

```
update Funkcija  
set opis = „Dummy funkcija“  
where IDAplikacija in (select IDAplikacija from Aplikacija where naziv = „Dummy aplikacija“);
```

- n) Ispisati sve kontrole koje su povezane s aplikacijom „Dummy aplikacija“ (sve kontrole koje pripadaju formi koja pripada navedenoj aplikaciji).

```
Select distinct k.*  
From (kontrola k inner join forma f on k.idForma = f.idForma) inner join Aplikacija A on  
A.idAplikacija = f.idAplikacija  
where A.naziv = „Dummy aplikacija“;
```

ili

```
Select distinct k.*  
From kontrola k, forma f, Aplikacija A  
where k.idForma = f.idForma and A.idAplikacija = f.idAplikacija  
and A.naziv = „Dummy aplikacija“;
```

ili

```
select *  
from kontrola k  
where IDForma in  
    (select Idforma from forma where IDAplikacija IN  
        (select IDaplikacija from aplikacija where naziv = „Dummy aplikacija“));
```

- o) Ispisati tipove koji se ne koriste

```
Select t.*  
From Tip t  
Where IDTip not in  
    (select IDTip from Funkcija  
    Union  
    Select IDTip from Kontrola  
    Union  
    Select IDTip from Svojstvo);
```

```
select t.*
from Tip t
where IDTip not in (select IDTip from Funkcija)
and IDTip not in (select IDTip from Kontrola)
and IDTip not in (select IDTip from Svojstvo);
```

Objašnjenje: Pregledom slike možemo ustanoviti da je tablica Tip povezana s tablicama Funkcija, Kontrola i Svojstvo. Zaključujemo da neki definirani tip nije u upotrebi ukoliko trenutno niti jedna funkcija nije tog tipa, niti jedno svojstvo nije tog tipa, te niti jedna kontrola nije tog tipa.

p) U kojim sve zadacima je upotrebljena DML naredba?

Samo u zadatku i) gdje smo naredbom

alter table aplikacija add column datumKreiranja datetime NOT NULL;

dodali novi atribut (stupac) datumKreiranja u entitet (tablicu) Aplikacija. Uz novi atribut smo dodali i not null ograničenje.