

Za potrebe vježbi kreirana je baza nalaziti pod nazivom **knjiznica**. Baza za sada ima otprilike 800 mjesta, 5500 autora, 350 izdavača i oko 700 kataložnih zapisa. Ovo je mala baza, nazovimo je bazica marica 😊.

SELECT [na jednoj tablici]

```
SELECT * from mjesto;
```

Ovo je ponovljena naredba iz prošle vježbe. Cilj SQL naredbe nije izlistati sve podatke pa onda u rezultatima tražiti onaj koji nas zanima. Cilj je dobiti iz baze samo one podatke koji nas konkretno zanimaju. Idemo malo detaljnije upoznati SQL select naredbu.

```
SELECT stupac, konstanta ili izraz [* za sve kolone i redove]  
FROM naziv_tablice  
[ WHERE izraz_za_filtriranje ]
```

Primjećujemo dodatak naredbi u uglatim zagradama – WHERE. Where je neobavezni ali ipak ključni dio select naredbe jer se u tom dijelu filtriraju podaci i baza vraća samo one željene podatke.

```
SELECT * from mjesto  
WHERE sifra=2;
```

Zvijezdica (*) označava sve kolone (uzima se redosljed zadan definiranjem tablice – create table naredba).

```
SELECT naziv, postanskiBroj from mjesto  
WHERE sifra=2;
```

Sada više nemamo sve kolone već redom prvo naziv zatim poštanski broj.

```
SELECT postanskiBroj, 'fiksni tekst', length(naziv) from  
mjesto  
WHERE sifra=2;
```

Kao što smo naveli u definiciji select naredbe u select listu može ići kolona (**postanskiBroj**), konstanta (**'fiksni tekst'**) i izraz (**length(naziv)**).

```
SELECT postanskiBroj as 'Poštanski broj',  
length(naziv) as 'Broj znakova u nazivu mjesta' from mjesto  
WHERE sifra=2;
```

Svakoj koloni možemo dati ime s naredbom **as**. Ako ne koristimo naredbu **as** onda će kolona imati ime koje joj je zadano tijekom kreiranja tablice.

Izvesti sjedeće SQL naredbe

```
SELECT sifra, ime, prezime, datumRodjenja from autor  
WHERE sifra=2;
```

```
SELECT naziv, aktivan from izdavac  
WHERE sifra=2;
```

Where dio se sastoji od uvjeta za pretraživanje (filtriranje podataka). Filtriranje se izvodi na osnovu jedne ili više kolona. Za filtriranje se koriste sljedeći operatori:

1. Operatori uspoređivanja
 =, !=, <, >, <=, >=
2. Logički operatori
 and, or, not
3. Ostali operatori
 like, in, between, is null, is not null

Primjeri koji koriste sve navedene operatore

```
SELECT naziv, postanskiBroj from mjesto  
WHERE sifra>2 and postanskiBroj is null;
```

```
SELECT naziv, postanskiBroj from mjesto  
WHERE not (sifra>2 and postanskiBroj is not null);
```

```
SELECT ime, prezime from autor  
WHERE sifra!=2 and datumRodjenja between '1980-01-01' and  
'1980-02-01';
```

```
SELECT ime, prezime from autor  
WHERE (sifra=2 and sifra<=4) or (sifra>15452 and  
sifra<15460);
```

```
SELECT naziv from izdavac  
WHERE sifra in (414,490,696);
```

```
SELECT naziv from izdavac  
WHERE naziv like 'a%';
```

```
SELECT naziv from izdavac  
WHERE naziv like '%nt';
```

```
SELECT naziv from izdavac  
WHERE naziv like 'a%k';
```

```
SELECT naziv from izdavac  
WHERE naziv like '%obrt%';
```

Cijela logika SQL-a je da je čovjek u stanju verbalno uvijet za dohvaćanje podataka vrlo jednostavno pretočiti u SQL naredbu. U praksi to izgleda ovako: Trebaju mi sve knjige čiji naslov počinje s slovom a. Naredba glasi: `select * from katalog where naslov like 'a%'`

Za sljedeće zahtjeve kreirajte SQL naredbe

1. Izlistajte sva imena i prezimena autora
2. Izlistajte sve naslove knjiga u katalogu koje u sebi sadrže niz znakova LJUB
3. Izlistajte sva mjesta koja imaju definiran poštanski broj
4. Izlistajte nazive mjesta čija šifra nije između 2000 i 3000
5. Postoji li u Hrvatsko mjesto koje počinje s slovom Y?
6. Koliko je autora rođeno 7. prosinca 1980?
7. Izlistajte sve aktivne izdavače čiji naziv završava s znakom . (točka)
8. Izlistajte sve neaktivne izdavače koji nemaju šifre 346, 234, 589
9. Izlistajte sve kataložne zapise čija je šifra veća od 3000 ili naslov u sebi sadrži riječ ljubav

Dodatne mogućnosti select naredbe na jednoj tablici (koje ćemo obraditi) su:

1. `distinct` [vraća samo jedinstvene rezultate]
`SELECT distinct length(ime) from autor;`

Select naredba na samo jednoj tablici nam daje mogućnost filtriranja i dohvaćanja podataka samo iz te jedne tablice. Vrlo rijetko će se u praksi pokazati potreba da dohvaćate podatke iz samo jedne tablice. Zašto? Zato što su tablice međusobno povezane vanjskim ključevima i podatkovno su ovisne jedne od drugima. Najčešći slučaj dohvaćanja podataka je primjer uvjeta u jednoj tablici a ispisa rezultata iz druge tablice. U našem katalogu primjer bi bio da želim znati koje je sve knjige napisao August Šenoa? Idemo prvo vidjeti naš „stari način“ gdje radimo select na jednoj tablici. Prvo moramo izvesti naredbu:

```
SELECT sifra from autor  
where ime='August' and prezime like '%enoa';
```

i dobijemo jedinstveni rezultat broj 2. Što sada? Sada idemo u tablicu katalog i izvodimo sljedeći select:

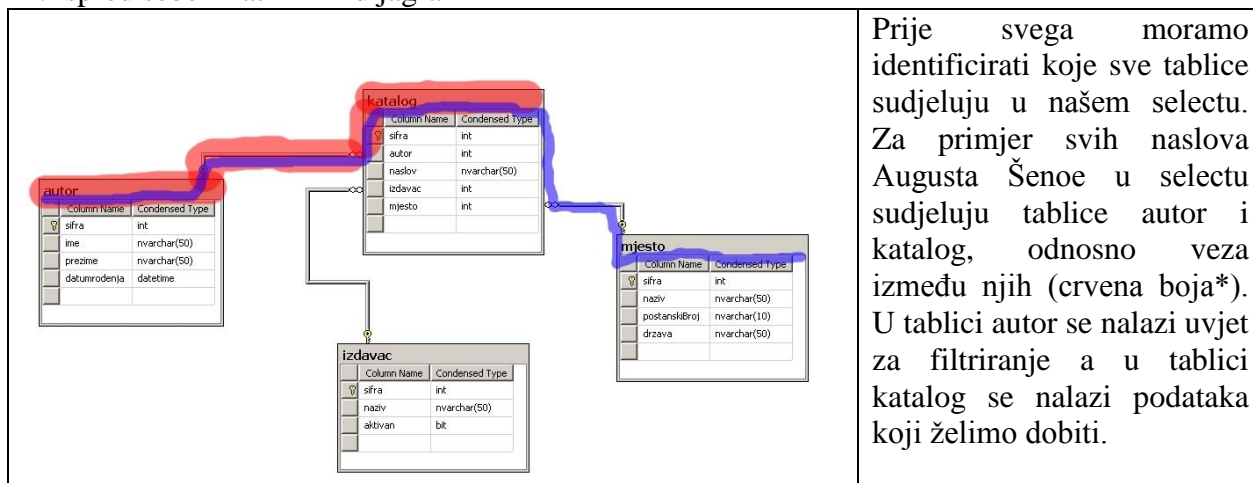
```
SELECT naslov from katalog  
where autor=2;
```

jeeeee, dobijemo dva rezultata, Čuvaj se senjske ruke i Zlatarevo zlato. To je točno. Mi u bazi imamo samo dvije knjige od Augusta. Ovo nije način dohvaćanja podataka kroz više tablica! Ako imamo potrebu dohvatiti podatke kroz više tablica pratimo sljedeću kuharicu.

1. Ispred sebe imati ERA dijagram
2. Napisati select i select listu ostaviti praznu
3. Napisati from dio i pospajati tablice prema uvjetima vanjskih ključeva
4. Napisati uvjete za filtriranje (where dio)
5. Napisati koje kolone želimo pokazati u select dio

Ključni dio kod spajanja tablica je pojam lanac vanjskih ključeva! Ako pogledate dijagram ispod vidjet ćete lanac. Primjer (slika plava boja*). Tablice autor, katalog i mjesto tvore lanac od tri tablice zato što su međusobno povezane vanjskim ključevima.

1. Ispred sebe imati ERA dijagram



* linije su neravne jer Hrvatske željeznice nemaju dobar sustav amortiziranja vagona ☹

2. Napisati select i select listu ostaviti praznu

```
SELECT
```

3. Napisati from dio i pospajati tablice prema uvjetima vanjskih ključeva

```
SELECT  
FROM autor a INNER JOIN katalog b  
on a.sifra = b.autor
```

4. Napisati uvjete za filtriranje (where dio)

```
SELECT
FROM autor a INNER JOIN katalog b
on a.sifra = b.autor
WHERE
a.ime='August' and a.prezime='Šenoa';
```

5. Napisati koje kolone želimo pokazati u select dio

```
SELECT b.naslov
FROM autor a INNER JOIN katalog b
on a.sifra = b.autor
WHERE
a.ime='August' and a.prezime like '%enoa';
```

Zašto preskačemo dijelove SQL naredbe? Zato što u trenutku pisanja select dijela naredbe ne znamo iz koje tablice ćemo htjeti podatke. Tablice u from dijelu dobivaju zamjenska imena (a,b,c,d,e,...). Ta imena mogu biti bilo koji niz znakova. Zamjenska imena dajemo zato da si olakšamo posao pisanja sql naredbe gdje umjesto da svaki put napišemo npr. katalog.sifra mi napišemo a.sifra jer smo tablici katalog dali zamjensko ime a. Drugi razlog davanja zamjenskih imena je potreba jedinstvenog identificiranja kolone kada su u igri više tablica. U tablici katalog i u tablici autor imamo kolonu sifra. Ako bi napisali u select dio samo sifra (bez zamjenskog imena ili stvarnog imena tablice) dobili bi grešku.

Primjer select naredbe kroz sve 4 tablice koje se nalaze u bazi.

Izlistajte mi sve naslove knjiga i nazive mjesta u kojima je izdana knjiga čiji autor je rođen 1976 i izdavač je još uvijek aktivan.

1. Staviti dijagram ispred sebe (preporučam tiskani oblik)

2. Napisati select i select listu ostaviti praznu

```
SELECT
```

3. Napisati from dio i pospajati tablice prema uvjetima vanjskih ključeva

```
SELECT
FROM autor a
INNER JOIN katalog b on a.sifra = b.autor
INNER JOIN izdavac c on b.izdavac = c.sifra
INNER JOIN mjesto d on b.mjesto = d.sifra
```

4. Napisati uvjete za filtriranje (where dio)

```
SELECT
FROM autor a
INNER JOIN katalog b on a.sifra = b.autor
INNER JOIN izdavac c on b.izdavac = c.sifra
INNER JOIN mjesto d on b.mjesto = d.sifra
WHERE
a.datumrođenja between '1976-01-01' and '1976-12-31'
and c.aktivan=1;
```

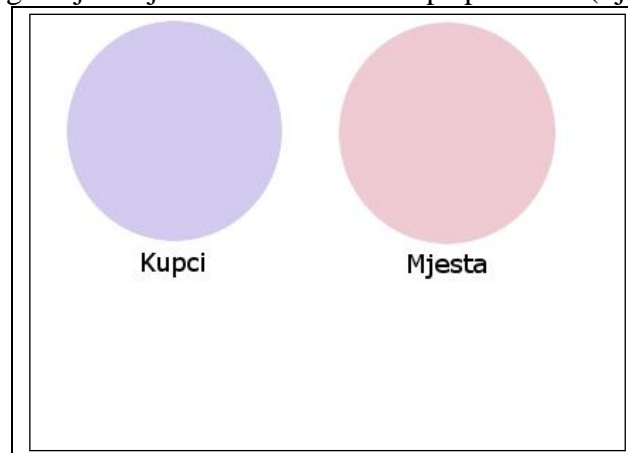
5. Napisati koje kolone želimo pokazati u select dio

```
SELECT b.naslov, d.naziv
FROM autor a
INNER JOIN katalog b on a.sifra = b.autor
INNER JOIN izdavac c on b.izdavac = c.sifra
INNER JOIN mjesto d on b.mjesto = d.sifra
WHERE
a.datumrodenja between '1973-01-01' and '1974-01-01'
and c.aktivan=1;
```

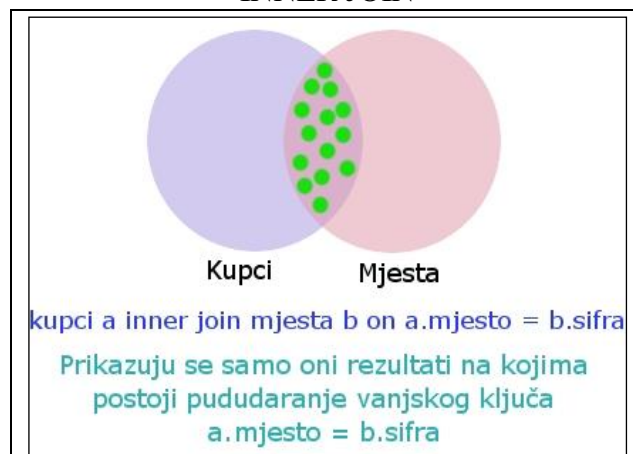
Do sada nam je ostala nepoznanica INNER JOIN. To je način spajanja tablica. Konkretno INNER JOIN definira spajanje samo onih podataka na kojima postoji podudaranje vanjskih ključeva. Nešto kasnije ćemo definirati i ostale načine spajanja. Želio bih vam prikazati jednu "omanju bazu". Ovaj izraz namjerno koristim da naglasim kako je ovo jako mala baza i da baze u srednjim i većim informacijskim sustavima mogu imati i po stotine tablice, ako ne i tisuću.

Kao što smo već spomenuli, svi dosadašnji načini spajanja su išli s INNER JOIN načinom. Postoje i ostali načini spajanja i sljedeće slike zorno prikazuju njihove funkcionalnosti.

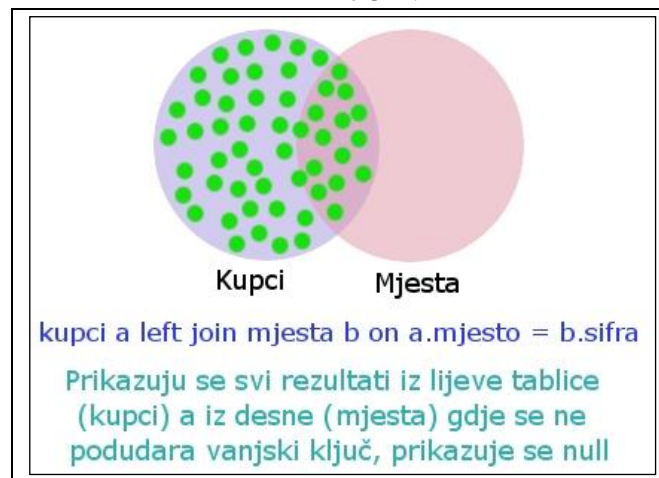
Uvijek se gledaju dvije tablice kao dva skupa podataka (lijeva i desna)



INNER JOIN

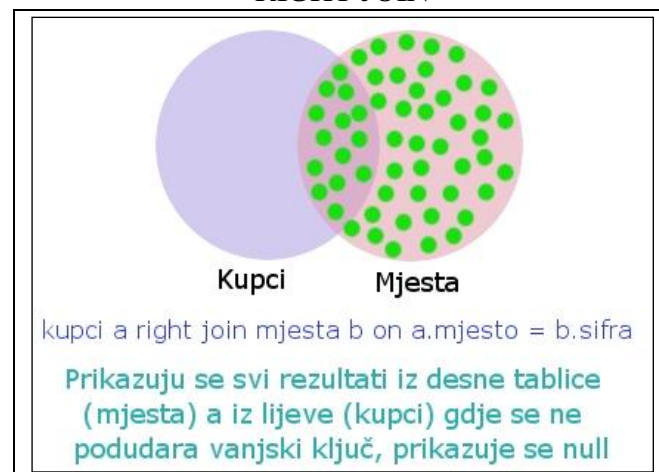


LEFT JOIN

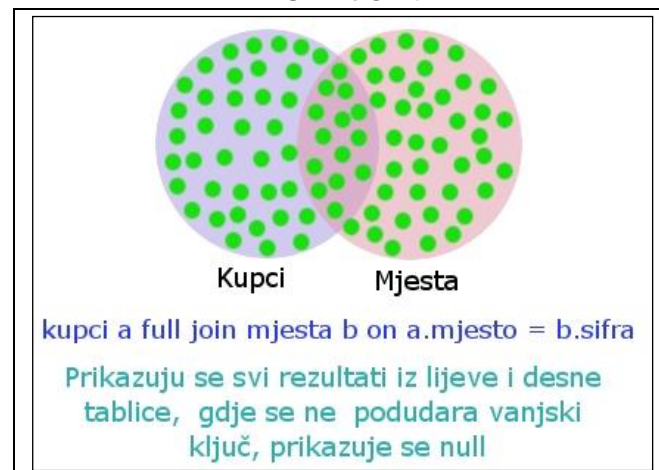


Ostali načini spajanja tablica (nećemo koristiti)

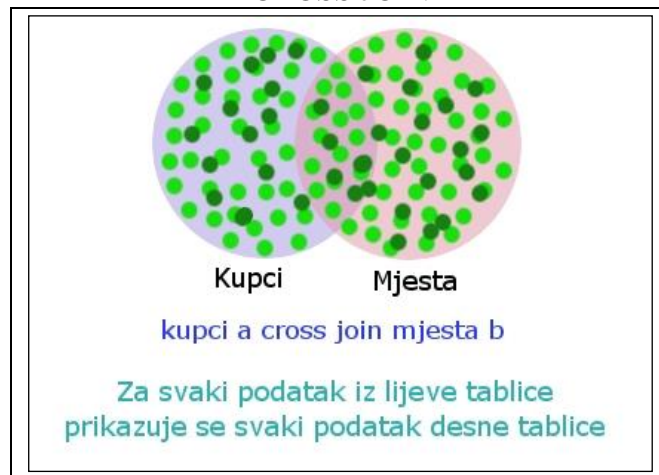
RIGHT JOIN



FULL JOIN



CROSS JOIN



Za vježbu se spojite na bazu **knjiznica** i vježbajte upite navedene u ovom dokumentu.

UPDATE [osnovno]

Update naredba (kao što joj i prijevod kaže) mijenja podatke u tablici. Jedna update naredba može promijeniti 0, 1 ili više podataka u tablici. Za sada ćemo samo prikazati mijenjanje jednog podatka. To mijenjanje je tkz. mijenjanje podataka po ključu. Sve ostale načine ćemo prikazati nakon obrade select naredbe. Update naredba može promijeniti podatak u jednoj ili više kolona tablice.

Primjer mijenjanja podatka u jednoj koloni na jednom redu

```
UPDATE mjesto set naziv='Kopenhagen'  
      where sifra=2;
```

Primjer mijenjanja podatka u više kolona na jednom redu

```
UPDATE mjesto set naziv='Osijek', drzava='Hrvatska',  
                  postanskiBroj='31000'  
      where sifra=4;
```

DELETE [osnovno]

Delete naredba (kao što joj i prijevod kaže) briše/miše podatke iz tablice. Jedna delete naredba može obrisati 0, 1 ili više redova u tablici. **Delete naredba briše cijeli red iz tablice!** Za sada ćemo samo prikazati brisanje jednog reda. To brisanje je tkz. brisanje reda po ključu. Sve ostale načine ćemo prikazati nakon obrade select naredbe.

Primjer brisanja podataka o kataloškom zapisu o „Priče iz davnine“

```
DELETE FROM katalog where sifra=2;
```