SADRŽAJ

BAZE PODATAKA

Uvod

U svakodnevnom životu pojavljuje se čitav niz raznih evidencija, kartoteka, imenika, albuma koji predstavljaju manje ili više složene **baze podataka.** Neke od jednostavnijih evidencija, imenika i sl. možemo sasvim uspješno voditi i pomoću programa za tablične proračune koje u tom slučaju nazivamo **plošnim bazama** (jer se svi podaci nalaze samo u jednoj tablici tj. predstavljeni su dvodimenzionalno – plošno). Tu postoje znatna ograničenja (vidi poglavlje Tablični proračuni).

Koncepcija baze podataka polazi sa stajališta stvaranja jedinstvenog skupa podataka tako da između tih podataka postoje određeni odnosi. Jedan te isti skup podataka služi većem broju aplikacija odnosno korisnika. Baze podataka mogu se definirati kao *skup povezanih podataka* ili preciznije rečeno ***baza podataka*** *je organizirana i uređena cjelina međusobno povezanih podataka spremljenih bez nepotrebne* ***redundancije*** (***zalihosti***)[1].

U zavisnosti od strukture odnosa između entiteta[2] baze podataka i njihovog povezivanja razlikuju se tri osnovna modela baza podataka:

- Hijerarhijske[3]
- Mrežne[4]
- Relacijske

Od šezdesetih do osamdesetih godina prošlog stoljeća prevladavale su **hijerarhijske** i **mrežne** baze podataka, a od osamdesetih godina prošlog stoljeća to su **relacijske baze podataka** koje su danas gotovo isključivo u uporabi.

U nastavku bavit ćemo se samo relacijskim bazama podataka.

RELACIJSKE BAZE PODATAKA

Relacijska baza podataka sastoji se od skupa povezanihtablica odnosno relacija. Definicija jedne relacije naziva se **relacijska shema** i sastoji se od naziva relacije i popisa atributa (obilježja) koji su u sastavu te relacije. Definicija čitave baze podataka naziva se **relacijska shema baze podataka**, a sastoji se od relacijskih shema svih relacija koje ulaze u sastav baze podataka.
Relacijska struktura u velikoj je mjeri podudarna s predodžbom događaja stvarnog svijeta i podataka koje ti događaji generiraju. Zbog svoje jednostavnosti i prilagođenosti ljudskom shvaćanju podataka i odnosa među njima relacijski model ima prednost pred ostalim modelima podataka. Relacijska baza podataka danas je najčešće korišteni model baza podataka.

---

[1] Udvostručavanje ili umnožavanje
[2] Entitet predstavlja stvaran ili apstraktan objekt ili događaj o kojem se u informacijskom sustavu prikupljaju podaci.
[3] Hijerarhijske baze podataka zasnivaju se na hijerarhijskim strukturama podataka koji imaju oblik stabla.
[4] Mrežne baze podataka zasnivaju se na mreži podataka povezanih tako da ne postoje ni određeni ni osnovni segmenti.

Upravljanje bazom podataka vrši **Sustav za upravljanje bazom podataka – SUBP** (*engl*, *Database Management System* – **DBMS**) odnosno **RDBMS** za relacijske baze. RDBMS je program, zapravo programski sustav, koji pohranjuje podatke u obliku međusobno povezanih tablica, odnosno to je sustav za organiziranje baze podataka i rad s podacima u njoj. Ovaj sustav u osnovi obavlja dvije grupe poslova:

1. **Definiranje baze podataka** (*engl. Database Definition*)
2. **Rad s podacima** (*engl. Database Management*)

U sklopu ovoga obuhvaćeno je uz ostalo:

- Definiranje tablica i relacija
- Unošenje, uređivanje, prikazivanje, pretraživanje, sortiranje i filtriranje podataka
- Dohvat podataka
- Zaštita integriteta podataka
- Kontrola istovremenog pristupa podacima
- Zaštita od neovlaštenog korištenja
- Stvaranje izvještaja
- Kontrola baze podataka
- Obnova baze u slučaju "pada"

TABLICA

Tablica *(engl. Table)* je osnovni objekt relacijske baze podataka i u njoj su pohranjeni podaci.
Redak tablice naziva se **Relacija** (*engl. Relation*), pojmovno je podudaran sa slogom podataka i predstavlja informaciju o jednom subjektu (učenik, zaposlenik, škola..), dakle **relacije** su pohranjene kao tablice**.**
Tablica (relacija) sastoji se od **redaka** (slogova) *(engl. Record)* i **stupaca** koji se nazivaju **atributi** i pojmovno odgovaraju **polju podataka** *(engl. Field*).

stupac – polje ↓

redak –slog ➔

ID_ucenik	OIB	ime	prezime	datum_r	mjesto_s	ulica	broj
1	19109803800	Ivica	Šimić	19.10.1980	Makarska	Sinjska	3
2	10099793800	Davor	Zorić	10.09.1979	Zadar	Bubalova	45a
3	01019813850	Nada	Janković	01.01.1981	Šibenik	Kninska	12
4	12019693851	Vlatka	Grgić	12.01.1969	Split	Bregovita	23
5	11129874252	Cvita	Jukić	11.12.1987	Split	Lička	6
6	25119793810	Juraj	Carić	25.11.1979	Omiš	Južna	45
...	...	...	...	...	...	...	...

Slika 1. – Tablica

Istovrsni objekti (recimo učenici) prikazani su u tablici redcima koji su opisani stupcima ili **poljima** (Ime, Prezime...).

Osnovne karakteristike tablice (relacije) su:

- ne postoje dva jednaka retka
- ne postoje dva jednaka stupca
- redoslijed redaka nije bitan
- redoslijed stupaca nije bitan

Primarni ključ

Polje ili više polja (Atributa) kojima se može jednoznačno definirati redak (slog) tablice naziva se **primarni ključ.**

Primjer: U tablici **učenik** polje **OIB** je ono koje jednoznačno definira redak te može biti primarni ključ jer ne postoje dvije ili više osoba s istim jedinstvenim matičnim brojem građana (OIB). Malo kasnije pokazat ćemo zašto je jednostavnije, a obično i nužno kreirati novo polje za identifikaciju slogova za primarni ključ, u ovom slučaju ID_ucenik [5](Slika 1.). Sam RDBMS ovom polju automatski dodjeljuje jedinstveni broj (za svaki novi slog zadnji dodijeljeni broj povećava se za 1).

TABLICA UČENIK

ID_ucenik	OIB	ime	prezime	datum_r	mjesto_s	ulica	broj
1	19109803800	Ivica	Šimić	19.10.1980	Makarska	Sinjska	3
2	10099793800	Davor	Zorić	10.09.1979	Zadar	Bubalova	45a
3	01019813850	Nada	Janković	01.01.1981	Šibenik	Kninska	12
4	12019693851	Vlatka	Grgić	12.01.1969	Split	Bregovita	23
5	11129874252	Cvita	Jukić	11.12.1987	Split	Lička	6
6	25119793810	Juraj	Carić	25.11.1979	Omiš	Južna	45
...	...	...	...	...	...	...	...

Slika 2. - Tablica Učenik

Primarni ključ upotrebljava se za povezivanje tablica i ima dvostruku ulogu: **jednoznačno definira retke tablice, a preko njega se ostvaruje i veza s drugim tablicama**

Primarni ključ mora zadovoljavati:

- vrijednost primarnog ključa mora biti jednoznačna
- primarni ključ ne može imati vrijednost NULL (ne može biti prazno polje)
- primarni ključ mora postojati kod kreiranja i spremanja sloga.

Uzmimo da nam je zadatak izraditi bazu podataka u kojoj će se pratiti učenici (njihovi osnovni podaci) te njihovo sudjelovanje na natjecanjima i uspjeh na istim.
Podaci koje želimo imati zapisane u bazi su:
Identifikacija sloga (retka), Ime, Prezime, OIB, Datum rođenja, Mjesto stanovanja, Adresa stanovanja (ulica i broj), Županija, Škola, Školska godina, Vrsta natjecanja, Predmet i Uspjeh. Kako smo ranije rekli, podaci se u relacijskim bazama pohranjuju u tablicama pa ćemo sukladno tome pokušati sve zahtijevane podatke pohraniti u tablicu prikazanu na Slici 3.

ID	ime	prez.	OIB	dat.rođ.	mjesto	adresa	županija	škola	šk. god.	vrsta natj.	predmet	uspjeh

Slika 3. – Tablica Učenici

[5]Prefiks ID koristimo radi lakšeg uočavanja polja koja su ključevi (primarni ili vanjski).

Primijetimo da u slučaju ovako formirane tablice OIB ne bi mogao biti primarni ključ i da ovdje treba postaviti posebno polje (ID), o čemu je bilo riječi prije, a čija vrijednost će jednoznačno definirati svaki slog.
Kad počnemo popunjavati tablicu podacima, vidimo (Slika 4) mnogostruko ponavljanje istih podataka (ime, prezime, OIB, datum rođenja, mjesto, adresa, županija, škola), odnosno gotovo cijelih slogova za pojedinog učenika za svaku školsku godinu, vrstu natjecanja te predmet (Slika 4).

ID	ime	prez.	OIB	dat.rođ.	mjesto	adresa	županija	škola	šk. god.	vrsta natj.	predmet	uspjeh
1	Ivica	Šimić	19109803800	19.10.1980	Split	Sinjska 3	Split.-dal.	Pomorska	2005/06	državno	Inform.	75%
2	Ivica	Šimić	19109803800	19.10.1980	Split	Sinjska 3	Split.-dal.	Pomorska	2006/07	županij.	Inform.	86%
3	Ivica	Šimić	19109803800	19.10.1980	Split	Sinjska 3	Split.-dal.	Pomorska	2006/07	općinsko	fizika	51%
4	Ivica	Šimić	19109803800	19.10.1980	Split	Sinjska 3	Split.-dal.	Pomorska	2007/08	državno	Inform.	48%
5	Davor	Zorić	10099793800	10.09.1979	Varaždin	Kninska	Varaždinska	Tehnička	2006/07	školsko	fizika	81%
6	Davor	Zorić	10099793800	10.09.1979	Varaždin	Kninska	Varaždinska	Tehnička	2006/07	općinsko	hrvatski	98%
7	Davor	Zorić	10099793800	10.09.1979	Varaždin	Kninska	Varaždinska	Tehnička	2007/08	općinsko	engleski	91%
8	Davor	Zorić	10099793800	10.09.1979	Varaždin	Kninska	Varaždinska	Tehnička	2007/08	županij	engleski	61%
9	Hrvoje	Anić	29019803805	29.01.1980	Split	Bubalova	Split.-dal.	Glazbena	2006/07	županij	engleski	78%

Slika 4. – Tablica Učenici s upisanim podacima o natjecanjima

Ovo je u suprotnosti s definicijom baze podataka koja glasi:
Baza podataka je skup međusobno povezanih podataka, spremljenih bez **redundancije (zalihosti**), a u kontekstu baze podataka **redundantnost (zalihost)** je pojava kad je ista činjenica nepotrebno zapisana više puta.
Nepotrebno ponavljanje istih podataka u bazi dovodi do niza poteškoća u radu s bazom. Poteškoće se očituju na razne načine i to počevši od nepotrebnog zauzimanja memorijskog prostora do pojava tzv. anomalija unosa, promjene i brisanja podataka. Vodi li se primjerice u našoj bazi adresa stanovanja pojedinog učenika na više mjesta, ukoliko dođe do promjene adrese jednog učenika to se mora upisati na svim mjestima na kojima je adresa zapisana. Ukoliko na jednom mjestu to ne bi bilo obavljeno, došli bismo u situaciju da imamo krive i nepouzdane podatke u bazi. Da bi se ovo izbjeglo, vrši se tzv. **normalizacija.**

Normalizacija

Normalizacija je postupak kojim se tablice u bazi strukturiraju tako da se izbjegne redundantnost i međuzavisnost te da se stvori što konzistentniji model podataka.

Iza normalizacije stoji složena matematička teorija kojom se mi ovdje, naravno, nećemo baviti, međutim krajnje pojednostavljeno **normaliziranje znači da se tablica u kojoj se nepotrebno ponavljaju podaci organizira u veći broj tablica**. U rješavanju ovog problema, veliki značaj ima i iskustvo dizajnera baze.
Pomoću tzv. **pravila dobrog dizajna tablice** moguće je utvrditi ima li dizajn tablice smisla i je li lako primjenjiv.
Pravila dobrog dizajna tablice su:

1. **Jedinstvenost polja** – Svako polje u tablici mora predstavljati jedinstveni tip informacije.
2. **Primarni ključevi** – Svaka tablica mora imati primarni ključ koji se sastoji od jednog ili više polja tablice.
3. **Funkcionalna ovisnost** – Vrijednosti stupca s podacima pridružene svakoj od jedinstvenih vrijednosti primarnog ključa moraju se odnositi na subjekt tablice i u potpunosti ga opisivati.

4. **Nezavisnost polja** – Mora postojati mogućnost mijenjanja podataka u bilo kojem polju (osim primarnog ključa), a da se pritom ne utječe na podatke u ostalim poljima.

Sukladno svemu rečenom problem tablice Učenici (Slika 4.) možemo riješiti njenim "razbijanjem" u nekoliko manjih. Dakle:
Iz tablice Ucenik izdvajamo dio koji se odnosi na učenika, a koji se ne mijenja bez obzira na to kad je, na kojima i je li uopće sudjelovao na natjecanjima (Slika 5.), dakle izdvajamo njegove nepromjenjive podatke.

Tablica Ucenik

ID_ucenik	ime	prez.	OIB	dat.rod.	ID_mjesto	adresa	ID_skola
1	Ivica	Šimić	19109803800	19.10.1980	1	Sinjska	

Slika 5. – Nova tablica Ucenik

U posebnu tablicu također možemo izdvojiti podatke o mjestu (gradu) stanovanja. U ovoj tablici možemo voditi podatke o nazivu mjesta, poštanskom i pozivnom broju mjesta te o županiji u kojoj je taj grad (Slika 6), nepromjenjive podatke koji su vezani za mjesto. Dodavanjem tablice Mjesto u tablici Učenici sada **umjesto naziva mjesta vodimo identifikaciju pripadajućeg mjesta (ID_mjesto)**. Ovo nam omogućuje da prema potrebi možemo uključiti i ostale podatke o mjestu u prikaz podataka o učeniku (O ovom će biti riječi kasnije u poglavlju o UPITIMA) .

Tablica Mjesto

ID_mjesto	mjesto	pošt.br	poz.br	ID-zupanija
1	Split	21000	021	17
2	Zagreb	10000	01	1

Slika 6. – Tablica Mjesto

Tablica Natjecanje

ID_natjecanje	Šk_god.	razred	ID_ucenik	ID_vrsta_natj	ID_predmet	rezultat
1	2005/06	Ia				
2	2000/01	Ib				

Slika 7. – Tablica Natjecanje

Tablica Natjecanje (Slika 7) središnja je tablica koja povezuje sve ostale tablice. U njoj se vode podaci o svim natjecanjima po školskim godinama, vrstama natjecanja, predmetima, učenicima i rezultatima natjecanja. Uočimo da većinu atributa (polja) ove tablice predstavljaju vanjski ključevi ostalih tablica.
Tablice Županija, Skola, Vrsta natjecanja te Predmet (Slika 8) su tzv. **šifarnici**, odnosno tablice u kojima su uneseni podaci (npr. nazivi županija) koji su nepromjenjivi (npr. u RH postoji točno određen broj županija s nepromjenjivim nazivima, a ako ipak dođe do promjene nekog naziva, on se mijenja samo na jednom mjestu – u šifarniku).

Tablica Škola

ID_skola	Naziv_skole
1	23
2	45

Tablica Predmet

ID_predmet	Naziv_predmet
1	Hrvatski
2	Informatika

Tablica Županija

ID_zupanija	zupanija
1	Split.-dalmat.
2	Varaždinska

Tablica vrsta natj.

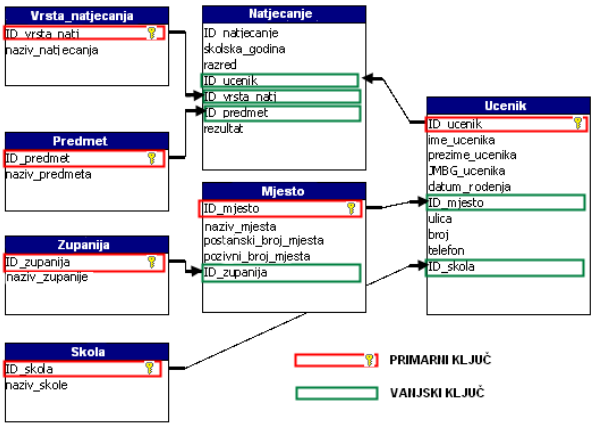
ID_vr_nat.	Naziv_nat.
1	Hrvatski
2	Informatika

Slika 8. – Tablice (šifarnici) Škole, Predmeti, Županije i vrsta natjecanja

Tako sad imamo sedam tablica kojima smo dodali polja za identifikaciju slogova (ID_ucenik, ID_mjesto, ID_natjecanje, ID_skola, ID_predmet, ID_zupanija i ID_vrsta) koja predstavljaju primarne ključeve odnosnih tablica.

Povezivanje tablica

Relacija (*engl. Relationship*) je po definiciji **odnos** ili **veza,** a u kontekstu relacijske baze podataka je povezivanje dviju tablica preko vrijednosti primarnog ključa, što znači da je ta vrijednost (primarni ključ) pohranjena u obje tablice. Povezane tablice sadrže iste vrijednosti s jedne strane (u jednoj tablici) u obliku **primarnog ključa** i s druge strane (u drugoj tablici) u obliku **vanjskog ključa** (stranog ključa) zbog čega moramo osigurati pravilno unošenje ovih vrijednosti čime štitimo **integritet** podataka u bazi. Jedno od osnovnih pravila zaštite integriteta podataka u bazi je pravilo **referencijalnog integriteta**, a ono je vrlo jednostavno i kaže: Vanjski ključ u povezanoj tablici **mora** odgovarati primarnom ključu osnovne tablice.
Pogledajmo primjer:
U tablici **Mjesto** polje **ID_mjesto** je primarni ključ koji jednoznačno definira svaki slog u ovoj tablici (primjer: Split - ovdje se može pojaviti samo jedanput). U tablici **Ucenik** (koja ima polje **ID_učenik** kao svoj primarni ključ) nalazimo isto polje **ID_mjesto** u kojem su sadržane vrijednosti primarnog ključa tablice **Mjesto**, a koje ovdje predstavlja **vanjski ili strani ključ** (*engl. Foreign key).* Za razliku od primarnog ključa strani ključ može biti ponavljan (u tablici **Ucenik** Split, odnosno njegov identifikacijski broj iz tablice mjesta, pojavljuje se u više slogova).
Ovo postaje potpuno jasno kad znamo da su u tablici **Mjesto** pohranjena mjesta u Republici Hrvatskoj u kojoj postoji samo jedan Split, dok su u tablici **Ucenik** pohranjeni podaci o učenicima, recimo Pomorske škole u Splitu, od kojih mnogi stanuju u Splitu.



Slika 9. – Veze među tablicama 9-1

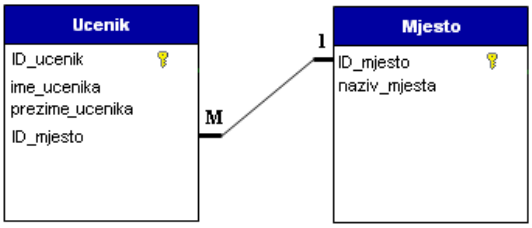
Odnosi između tablica - tipovi relacija

Odnosi[6] između raznih objekata u bazi predstavljeni su **vezama** (engl. *Relationship).* Postoje tri tipa veza, a to su:

[6] U lokaliziranoj (prevedenoj) verziji (inačici) MS Accessa 2007 **relacije** su prevedene kao **odnosi**

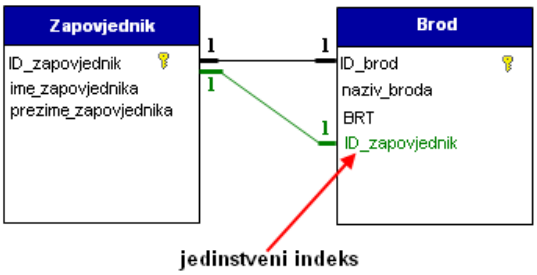
- jedan prema jedan (*one-to-one*) **1:1**
- jedan prema više (*one-to-many*) **1:M**
- više prema više (*many-to-many*) **M:M**

Najčešća veza je veza **1:M** (jedno mjesto – više učenika) kod koje jednoznačna vrijednost primarnog ključa može povezivati jedan, više ili čak niti jedan slog u povezanoj tablici.
U našem primjeru tablica **mjesta** i tablica **učenik** povezane su vezom **1:M**



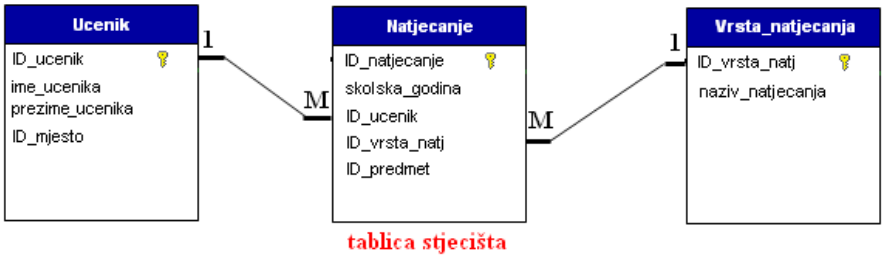
Slika 10. – Veza 1:M 10-1

Veza **1:1** (jedan brod – jedan kapetan) nije uobičajena jer informacije povezane na ovaj način najčešće su spremljene u istoj tablici (u tablici **Zapovjednik** ID_brod bi se nalazio kao vanjski ključ tablice **Brod**). Kod ove veze jednoznačna vrijednost primarnog ključa može povezivati samo jedan vanjski ključ (ili primarni ključ druge tablice), dakle svaki zapis u prvoj tablici može imati samo jedan odgovarajući zapis u drugoj tablici, a svaki zapis u drugoj tablici može imati samo jedan odgovarajući zapis u prvoj tablici. Da bismo dobili ovu vezu, moramo postaviti **jedinstveni indeks** (*engl. index*) na vanjski ključ što osigurava da se vanjski ključ ne može duplicirati (ponaša se kao primarni ključ).



Slika11. – Veza 1:1 11-1

Definiranje relacije **M:M** (više učenika sudjeluje na više natjecanja) riješeno je stvaranjem treće tablice, koja se često naziva **tablica stjecišta**, koja odnos **M:M** rastavlja u dva odnosa **1:M**. Primarni ključ iz svake od dviju tablica nalazi se u trećoj tablici. Kao rezultat treća tablica zapisuje svaku pojavu ili instancu odnosa.



Slika 12. – Veza M:M 12-1

Bazu može činiti jedna, ali u pravilu je čini **više međusobno povezanih tablica**.

Što zapravo dobivamo ovim povezivanjem tablica? Dobivamo mogućnost izdvajanja podataka iz jedne, dviju ili više tablica na način da, koristeći "upit" (***query***) ili "pogled" (**view**), prikazujemo slogove koji su sastavljeni od polja iz povezanih tablica, a fizički se ne moraju ponovo pohraniti. Upiti ili pogledi predstavljaju dakle virtualne tablice koje ne sadrže podatke, nego samo definicije takvih "tablica". Ovo i predstavlja glavnu snagu relacijskih baza.

Uz **tablice** i **upite** osnovni objekti baze podataka su još **obrasci** (*eng. Forms*) i **izvještaji** (*engl. Reports)* o čemu će biti više riječi u daljnjem tekstu.

SQL

Za obradu podataka u relacijskim bazama koristi se Strukturirani upitni jezik *(engl. **Structered Query Language***) **SQL,** odnosno njegove inačice ovisno o proizvođaču baze. Za većinu naših zahtjeva nije potrebno poznavanje ovog jezika za baze podataka jer grafičko sučelje koje nude MS Access, Open office Base i drugi samostalno koristi **SQL** koji se nalazi (radi) u njegovoj pozadini. Svi oni koji žele napredno koristiti relacijske baze, a to znači mnogo više od onoga što nudi grafičko sučelje, trebali bi poznavati SQL. Mi se nećemo posebno baviti SQL-om, ali pokazat ćemo jedan najjednostavniji primjer.
Jezgru SQL-a predstavlja naredba **SELECT** koja općenito ima oblik:

SELECT *<popis polja>*
FROM *<popis tablica>*
WHERE *<naredba za odabir slogova>*
GROUP BY *<odredbe za grupiranje slogova>*
HAVING *<odredbe za odabir grupa>*
ORDER BY *<odredbe za sortiranje slogova>*

tako npr. upit koji iz tablice **Ucenik**

ID_ucenik	OIB	ime	prezime	datum_r	mjesto_s	ulica	broj
1	19109803800	Ivica	Šimić	19.10.1980	Makarska	Sinjska	3
2	10099793800	Davor	Zorić	10.09.1979	Zadar	Bubalova	45a
3	01019813850	Nada	Janković	01.01.1981	Šibenik	Kninska	12
4	12019693851	Vlatka	Grgić	12.01.1969	Split	Bregovita	23
5	11129874252	Cvita	Jukić	11.12.1987	Split	Lička	6
6	25119793810	Juraj	Carić	25.11.1979	Omiš	Južna	45

Slika 13. - Tablica Ucenik

izdvaja sve učenike (ime, prezime i dtum rođenja) koji su rođeni poslije 01.01.1989. i sortirani po prezimenu (**Slika 14.**) izražen u SQL-u izgleda:

```
SELECT Ucenik.ime, Ucenik.prezime, Ucenik.datum_r
FROM Ucenik
WHERE (((Ucenik.datum_r)>#1/1/1989#))
ORDER BY Ucenik.prezime;
```

ime	prezime	datum_r
Hilda	Bitanić	30.12.1989
Ivo	Ivić	04.04.1991
Ante	Matić	12.12.1990
Ivica	Šimić	19.10.1989
Vlatka	Vulić	12.01.1989

Slika 14. - Izvršeni upit

IZRADA BAZE

Danas postoji čitav niz SUBP-a, a većina predstavlja i razvojnu okolinu tj. omogućuje stvaranje čitavih aplikacija s bazama podataka. Mnogi od njih zahtijevaju poznavanje nekog programskog jezika, a samim tim i iskustvo u programiranju. U tablici je prikazano nekoliko poznatijih proizvoda ove vrste. Odabir pojedinog proizvoda ovisi o mogućnostima i potrebama korisnika.

PROIZVOĐAČ	PROGRAM	OPERACIJSKI SUSTAV	PROGRAMSKI JEZIK
Oracle corporation	Oracle	MS Windows, Linux, Unix, Mac OS,..	SQL, Java..
IBM	DB2	MS Windows, Linux, Unix, Mac OS, VMS..	SQL, Java, Cobol
IBM	Informix	MS Windows, Linux, Unix, Mac OS..	SQL, Java..
Microsoft	MS SQL Server	MS Windows	SQL, C++
Microsoft	Access	MS Windows	SQL, Access Basic
My SQL AB	My SQL	Linux, Unix	SQL, C, PHP..
Sysbase Inc.	Sysbase	MS Windows, Linux, Unix	SQL, Cobol..
Sun (sada Oracle)	OpenOfficeBase	Windows, Linux, Mac OS..	SQL, Java
LibreOffice.org	LibreOfficeBase	Windows, Linux, Mac OS..	SQL, Java

Slika 15. – Tablica nekih proizvođača i njihovih programa za baze podataka

Od prikazanih za naše potrebe (školske) najprihvatljiviji su **OpenOfficeBase** (OpenOfficeBase) i **MS Access**.

Oba programa sličnih su mogućnosti koje u potpunosti zadovoljavaju naše potrebe. Sučelje oba programa vrlo je slično te onaj koji savlada rad na jednom od ovih programa lako se može prebaciti na rad s drugim. Moramo napomenuti da je MS Access (dio uredskog paketa MS Office) ipak nešto "snažniji", što je i normalno s obzirom da se radi o komercijalnom programu, za razliku od OpenOffice.org

Više o **OpenOffice.org Base (LibreOffice.org Base) u CARNetovoj e-knjižnici...**