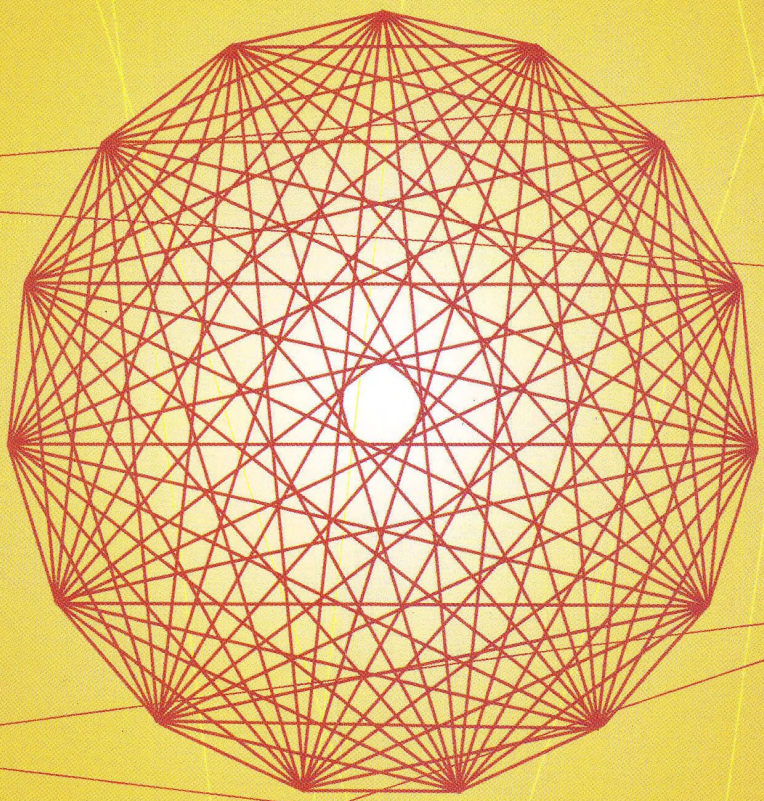
Baze podataka

Čičin-Šain, Marina

PROSINAC 2007.

CYBERNETICA

BILTEN DRUŠTVA KIBERNETIČARA



MARINA ČIČIN-ŠAIN:

BAZE PODATAKA

CYBERNETICA

Bilten Društva kibernetičara - Rijeka

MARINA ČIČIN-ŠAIN

BAZE PODATAKA

Prosinac 2007

Izdavač: Društvo kibernetičara
Broj primjeraka: 300
Tisak: Futura d.o.o. Rijeka

Lektorica: Nadija Budija, prof

Design ovitka: Jelena Babić dipl. diz.
Publikacija se izdaje uz financijsku potporu Grada Rijeka – Odjela gradske uprave za sport i tehničku kulturu i Zajednice tehničke kulture Rijeka

---

CIP - Katalogizacija u publikaciji
SVEUČILIŠNA KNJIŽNICA
RIJEKA

UDK 004.658

ČIČIN-Šain, Marina
Cybernetica - Baze podataka / Marina Čičin-Šain. - Rijeka : Društvo kibernetičara, 2007.

ISBN 978-953-95663-1-7

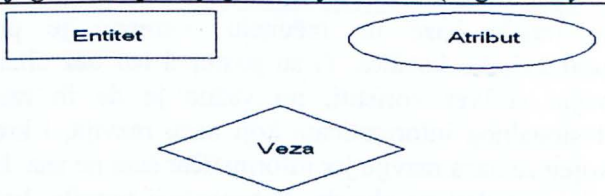
I. Baze podataka -- Upravljanje

111022064

Sadržaj

1. Priprema za korištenje baza podataka

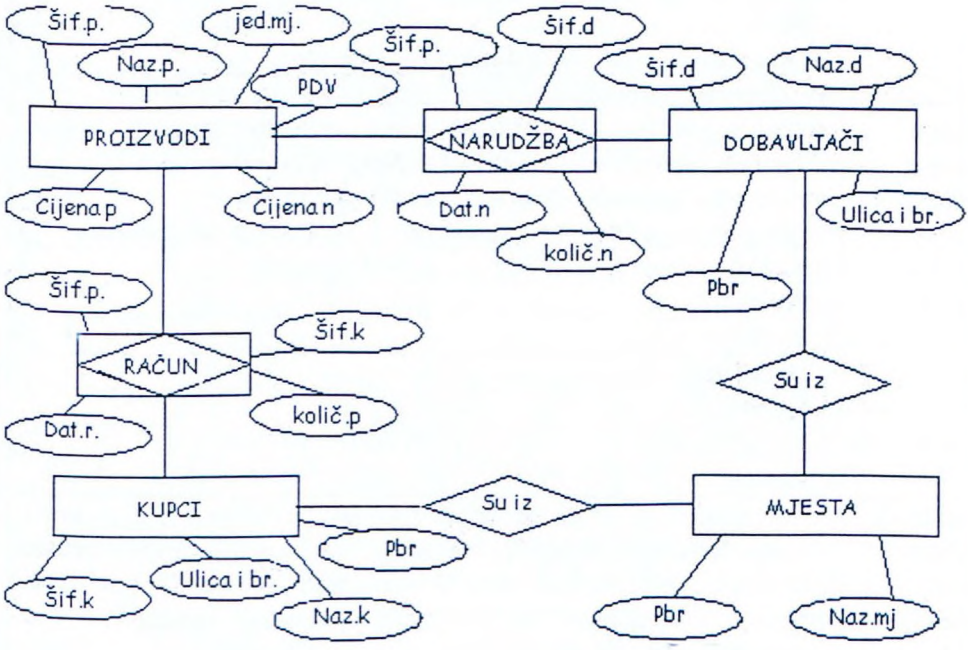
Što su baze podataka?	**Baza podataka** je termin koji se koristi za dva pojma. 1. Baze podataka su skupovi podataka i metoda koje služe da se ti podaci organiziraju, pohrane, ažuriranju, obrađuju i koriste. use store update process 2. Termin „baza podataka" koristi se i za razvojni softver koji omogućava kreiranje i korištenje baze podataka. Postoji više vrsta baza podataka u tom značenju riječi, a danas se najviše koriste **relacijske baze podataka**. Postoji softver za kreiranje, održavanje i korištenje baza podataka, a najčešće su danas korištene baze: Access (uz neku inačicu operativnog sustava Windows), Dbase 3+ i Clipper (za starija računala koja koriste operativni sustav MS DOS) i još neke. Pripreme za izradu baze podataka ovdje su dane smo na primjeru. Detaljnije definicije osnovnih pojmova i pravila naći ćete u udžbeniku *Informatika za informatičko poslovanje*.
Pripreme za izradu baze	Prije izrade baze na računalu potrebno je provesti neke **organizacijske korake.** Ti su postupci isti bez obzira koji će se razvojni softver koristiti, no važno je da ih razumije, osim profesionalnog informatičara koji bazu razvija, i krajnji korisnik za kojeg se baza razvija jer informatičar sam ne zna koji su podaci važni i potrebni za obradu. Poznavajući pravila, krajnji korisnik može osim toga provjeriti radi li informatičar po pravilima struke ili ne jer se sve anomalije u organizaciji posljedično negativno reflektiraju na posao pa će krajnjem korisniku, ako baza nije kreirana kako treba, biti teže ili čak nemoguće obaviti neke od poslova pri njenom korištenju.
Relacijska baza podataka	**Relacijske baze podataka** su baze kod kojih su podaci smješteni u jednu ili više tablica koje su međusobno povezane. Takve baze mogu biti na jednom računalu za samo jednog korisnika ili na jednom od računala u računalnoj mreži kojoj može pristupiti više korisnika. Postoje baze gdje se podaci nalaze raspoređeni (distribuirani) na više računala u mreži. U tom slučaju svi korisnici mogu imati pristup podacima bez obzira gdje se nalaze, naravno u skladu sa svojim ovlastima. Svaki od korisnika ne mora imati ista ovlaštenja. Neki korisnici neke podatke ne smiju ni vidjeti, drugi ih smiju vidjeti, ali ih ne smiju mijenjati, a najviša razina ovlaštenja, osim pristupa, dozvoljava i promjenu podataka.
Relacijski	Kod relacijskog se modela podaci nalaze u tablicama koje su

model	međusobno povezane ključevima (u relaciji). Time se uklanja **redundancija**. Redundancija je pojava kada se podaci javljaju u kopiji ili ako se čuva podatak koji se može dobiti obradom drugih podataka. Redundancija može biti korisna (za osiguranje podataka ili pri provjeri prijenosa podataka), ali kod baze je podataka redundancija najčešće štetna jer uzrokuje nepotrebno trošenje memorije i otežano održavanje podataka iz razloga što pri promjeni treba promijeniti podatak na više mjesta.
Prototip modela	Prave baze koje se koriste u praksi obično su vrlo složene pa ne bi bile dobre kao primjer za proučavanje kada se tek uči kako formirati bazu podataka. Zato se koriste **prototipi modela** koji su pojednostavljeni modeli, na kojima se mogu objasniti glavne značajke građe i funkcija modela. Kod prototipa se dakle, da bi se objasnilo funkcioniranje baze, ne uzimaju svi podaci koji se pojavljuju u bazi, nego samo oni posebno važni i uglavnom razumljivi. Prototip se kasnije može dograditi do pravog modela ili se pravi model gradi ispočetka na osnovi prototipa.
Dijagrami	Kod pripreme se za izradu baze koriste i grafički prikazi baze ili **dijagrami**, npr. dijagram objekt-veza (engl. *Entity-relationship*).
Gradbeni elementi dijagrama objekt--veza	 Za dijagrame objekt-veza koriste se osnovni gradbeni elementi. Ovaj se dijagram koristi kada imamo podatke koje želimo čim svrsishodnije spremiti u onoliko datoteka koliko je najbolje za tu bazu. Koriste se simboli: • Pravokutnik za entitet • Elipsa za atribut • Romb za vezu
Entitet	Podaci se skupljaju o nekom objektu ili **entitetu**. Entitet može biti bilo što o čemu se prikupljaju podaci.
Atribut	Podatak koji opisuje entitet zove se **atribut**. Taj se termin uvodi da bi se razlikovao općeniti podatak od pojedinačnog podatka. Primjerice jedan od atributa je *Naziv dobavljača*, a *Podravka* je naziv jednog od dobavljača.
Veza	Među entitetima se uspostavlja **veza**. Veza ovisno o kardinalnosti

PODACI → VRIJEDNOSTI ATRIBUTA

	može biti: 1:1 1:n n:1 n:n U slučaju 1:1 veze, ona se uspostavlja putem primarnog ključa oba entiteta, a možda je poželjno da se ta dva entiteta spoje u jedan. U slučaju 1:n i n:1 veze, ista se uspostavlja putem primarnog ključa na strani 1 i stranog na stani n. U slučaju n:n veze, veza se uspostavlja pomoću vezne tablice. U dijagram se upisuje kardinalnost koja se ponekad i ucrta u dijagram posebnim simbolima.

Baza podataka *Trgovina* – dijagram objekt-veza:



Osnovni entiteti ove baze su:

- Proizvodi
- Dobavljači
- Kupci
- Mjesta

U kakvoj su vezi entitet *Kupci* i *Mjesta* ovisno o njihovoj kardinalnosti?
Svaki kupac može imati sjedište u jednom mjestu, a u jednom mjestu sjedište može imati n kupaca pa kažemo da je ta veza n:1.
U kakvoj su vezi entitet *Kupci* i *Proizvodi* ovisno o njihovoj kardinalnosti?

Svaki kupac može kupiti n raznih proizvoda, a svaki proizvod može kupiti n raznih kupaca, pa kažemo da je ta veza n:n.
U slučaju n:n veze, ona se među entitetima uspostavlja pomoću vezne tablice pa se kaže da veza postaje entitet. Iz dijagrama je vidljiva veza da kupci kupuju proizvode uspostavljene pomoću vezne tablice *Račun*.

Pitanja:
Kako se između entiteta *Proizvodi* i *Dobavljači* može uspostaviti veza?
Koji su entiteti u vezi n:1?

Koliko će baza imati tablica?	Iz dijagrama se objekt-veza može vidjeti koliko će baza imati tablica. Baza će imati onoliko tablica koliko je u dijagramu entiteta. Koliko baza *Trgovina* ima entiteta? Ima ih šest: *Proizvodi*, *Kupci*, *Dobavljači*, *Narudžbe*, *Računi* i *Mjesta*. Dakle u bazi će biti šest tablica s tim imenima.
Koliko će tablice imati stupaca?	Iz dijagrama se može vidjeti i koliko će tih šest tablica imati stupaca. Svaka će tablica imati onoliko stupaca koliko odgovarajući entitet ima atributa. Npr. entitet *Proizvodi* ima šest atributa pa će tablica *Proizvodi* imati šest stupaca. Dopuni donju tablicu podacima o broju stupaca za sve tablice.

Koliko stupaca ima tablica	Broj stupaca
Proizvodi	6
Kupci	
Dobavljači	
Mjesta	
Narudžbe	
Računi	

Koliko će tablice imati redaka?	Iz dijagrama se ne može vidjeti i koliko će tih šest tablica imati redaka. Svaka će tablica imati onoliko redaka koliko bude u praksi potrebno, npr. tablica *Kupci* će imati redaka koliko ima registriranih kupaca, a tablica *Proizvodi* onoliko redaka koliko u bazi ima raznih proizvoda.
Slog ili n--torka	Svaki se redak iz tablice zove **n-torka** ili **slog** (engl. *Record*), a sadrži n podataka (u slučaju *Proizvoda* npr. 6), odnosno onoliko koliko ima atributa. Iz dijagrama je vidljivo koliko tablice imaju stupaca, a redaka će biti onoliko koliko trenutno ima proizvoda, kupaca, dobavljača, narudžbi, računa i poštanskih brojeva, što se na temelju dijagrama ne vidi.
Primarni ključ	Neki atribut ili grupa atributa može biti izabran za **primarni ključ**. Ključ se koristi za identifikaciju n-torke podataka. Kandidat za **primarni ključ** je atribut ili niz atributa koji moraju

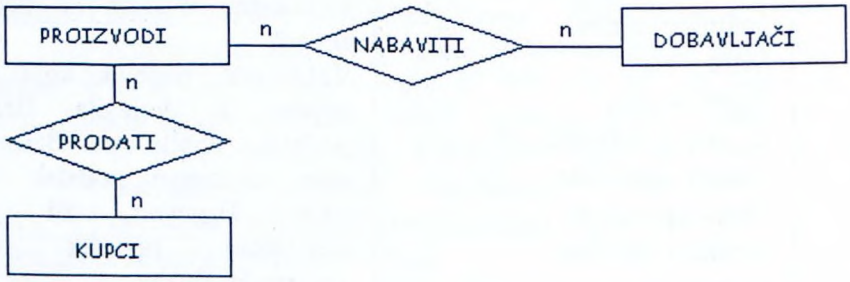
	ispunjavati dva svojstva: - u svakom retku postoje podaci za te atribute (ne smiju nedostajati) - za svaki je redak ključ jedinstven, odnosno ne smiju postojati dva retka s istim ključem
Primjer ključa	Za tablicu *Proizvodi* dobar kandidat za primarni ključ je *Šifra proizvoda* jer ispunjava oba svojstva. *Šifra proizvoda* se može, dakle, odrediti da bude ključ. Tablica ne mora imati ključ, ali se preporučuje da ga ima.
Štetna redundancija	Ako ponovo pogledate podatke u tablici *Dobavljača* i *Kupaca*, primijetit ćete da nema podataka za grad. Da smo te podatke spremili u tablicu *Dobavljača* i *Kupaca*, oni bi se ponavljali za n--torke iz istih gradova, pa bismo imali redundantne podatke. Umjesto toga je bolje podatke o gradu smjestiti u novu tablicu.
Strani ključ	Primarni je ključ te nove tablice *Poštanski broj*. Taj se ključ pojavljuje i u tablici *Dobavljači* i *Kupci* i u njima se zove **strani ključ**. Strani se ključ u tablici može pojaviti više puta, u ovom slučaju uvijek ako je kupac ili dobavljač iz tog grada.
Kardinalnost i relacije n:1	Kako ćemo povezati te tablice? Možemo ih povezati preko atributa *Poštanski broj*. Taj je atribut kandidat za primarni ključ u tablici *Mjesta*. U tablici *Dobavljača* on ne može biti primarni ključ jer se ponavlja u raznim recima. On je, dakle, tu atribut koji sadrži podatak o primarnom ključu neke druge tablice, a zove se **strani ključ**. Te su dvije tablice povezane relacijom **n:1**, odnosno više redaka iz prve tablice odgovara jednom (i samo jednom) retku iz druge tablice (dijagram *Trgovina*). Koristi se i termin da je **kardinalnost** među tablicama n:1. Svakom retku iz druge tablice može odgovarati n redaka iz prve tablice, pri čemu n može biti 0, 1 ili više redaka.
Razlozi za uređivanje (normalizaciju) tablica	Tablice je prije kreiranja potrebno urediti. Taj se postupak zove **normalizacija podataka**. Zašto je bolje podatke imati u dvjema povezanim tablicama, nego u jednoj proširenoj, tj. zašto ne bismo u tablicu *Dobavljača* dodali stupac za grad?
Ušteda prostora (izbjegavanje redundancije)	Na taj način štedimo prostor. Zamislite da imate tablicu *Dobavljača* od nekoliko stotina redaka, a neka je većina iz Rijeke. Za sve biste njih u slučaju jedne tablice morali u odgovarajući redak upisivati da su iz Rijeke i tako na stotine puta. U slučaju se dviju tablica u tablicu *Dobavljača* upisuje samo poštanski broj, a u tablici se *Mjesta* ime grada nalazi samo jedanput.
Manje pogrešaka (čuva se	Na taj način umanjujemo mogućnost pogreške. Ako smo grad ispravno upisali u tablicu *Mjesta*, taj će se ispravni naziv dobiti kad god ga pozovemo preko ključa, tj. podatka *Poštanski broj*.

integritet podataka)	Ako bismo sve pisali u jednu tablicu, mogli bismo na nekom mjestu pogriješiti i napisati da je npr. poštanski broj Bakra 51261, što je poštanski broj Bakarca. Ako podaci u bazi nisu kontradiktorni, kažemo da se čuva integritet podataka.
Lakše ispravke	Olakšavamo ispravke. Ako se ime nekog mjesta promijeni, to će se trebati ispraviti samo na jednom mjestu, a vrijedit će za sve.
Lakši i neovisan upis podataka	U tablicu *Gradova* možemo unijeti podatke i za gradove iz kojih nije niti jedan dobavljač. U našem primjeru možemo odmah unijeti podatak za Bakarac. Da imamo sve podatke u jednoj tablici, taj podatak ne bismo mogli upisati sve dok se ne bi pojavio barem jedan dobavljač ili kupac iz Bakarca.
Spajanje tablica	Netko može prigovoriti da je podatke preglednije imati u jednoj tablici. To je istina, ali u relacijskim se bazama može, kada je god to potrebno, podatke iz dviju tablica prikazati samo u jednoj tablici (kao u spojenoj tablici). Operacija kojom se iz dviju tablica uparuju odgovarajući podaci i prikazuju kao treća zove se **spajanje** (engl. *Join*).
Zbirka tablica	Poslužitelj baza podataka može raspolagati s više tablica. Skup svih tablica u jednoj bazi čini **zbirku** (engl. *Collection*) **tablica**. Takva zbirka i sama postaje entitet, a poslužitelj se baze podataka može brinuti za neka svojstva svojih entiteta, npr. za referencijalni integritet baze.
Referencijalni integritet baze	**Referencijalni** je **integritet** baze svojstvo da se promjenom podataka u bazi (brisanjem, dodavanjem ili mijenjanjem) ne narušavaju odnosi u bazi. U bazi *Trgovina* iz našeg primjera vrijedi pravilo da se podaci o mjestima nalaze u tablici *Mjesta* pa se ne smije dogoditi da se u tablici *Kupaca* ili *Dobavljača* pojavi poštanski broj koji nije u bazi *Mjesta*. Ni u bazi *Računi* ne smije se npr. koristiti šifra proizvoda koji nije u bazi *Proizvodi*.
Referencijalni se integritet baze brine da: - Ne možemo obrisati slog u tablici *Mjesta* ako je on strani ključ barem jednom retku bez da obrišemo i sve povezane slogove u tablici *Kupaca* ili *Dobavljača*. - Ne možemo mijenjati podatak o poštanskom broju u tablici *Mjesta* ako je on strani ključ barem jednom retku, bez da promijenimo i sve povezane slogove u tablici *Kupaca* ili *Dobavljača*. - Ne možemo dodati slog u tablicu *Kupaca* ili *Dobavljača* ako za njihov poštanski broj ne postoje podaci u tablici *Mjesta*. - Ne možemo promijeniti u tablici *Kupaca* ili *Dobavljača* poštanski broj ako za taj poštanski broj ne postoje podaci u tablici *Mjesta*.	
Primjer iz prakse	**Izrada baze na primjeru relacijske baze *Trgovine*.** Za daljnji ćemo rad koristiti bazu podataka o proizvodima, kupcima, dobavljačima i njihovim obavljenim transakcijama. Kako doći do dijagrama objekt-veza, kakav je npr. dijagram

	objekt-veza *Trgovina*? Taj dijagram djeluje složeno i nije nastao odjednom, nego u nekoliko koraka. Prvi je korak izrade neke baze podataka utvrđivanje koji su podaci dovoljno **važni (relevantni)** da bi ušli u tu bazu i u **koje tablice** će ući koji podaci. Podaci i tablice se u fazi projektiranja dogovaraju s korisnicima jer ni projektant baze podataka ni programer ne mogu znati koji su podaci relevantni za neku obradu, kakvi su ti podaci i kako utječu jedni na druge. Prirodu podataka poznaje korisnik, ali on često ne razumije što ga projektant baze ili korisnik pita. Zato je za korisnika jako važno razumjeti jezik kojim se projektant baze i programer služe. Dakle, prije izrade tablica u nekoj bazi podataka (npr. u Accessu) treba znati koje nam tablice trebaju, koja polja sadrže, što je ključ za svaku pojedinu tablicu i u kakvoj su vezi te tablice.
Relevantni podaci	**1. korak: Odabir relevantnih podataka za obradu** Relevantni podaci za *Trgovinu*: Šifra proizvoda Naziv proizvoda Jedinica mjere Šifra kupca Naziv kupca Cijena proizvoda Datum narudžbe Datum prodaje Prodana količina Adresa dobavljača Naručena količina Adresa kupca Relevantan je podatak za obradu onaj podatak koji je za tu obradu važan. Ne postoje apsolutno relevantni podaci, nego su podaci uvijek relevantni obzirom na neku obradu. Tako npr. podatak koju je ocjenu iz kolegija Baze podataka dobila Ana Banana nije relevantan podatak za bazu *Trgovina*, ali je relevantan podatak za studentsku službu. Relevantne podatke poznaje korisnik koji i inače radi taj posao, u ovom slučaju referenti prodaje i nabave u trgovačkom društvu, pa korisnik pomaže informatičaru u definiciji. Neka su nakon diskusije s korisnikom definirani relevantni podaci iz okvira *Relevantni podaci za trgovinu*. Kasnije je ustanovljeno da se za proizvod razlikuje nabavna i prodajna cijena, da se PDV od 22% obračunava na proizvode, ali ne na sve, i da se govori da će se možda uvesti još neka stopa PDV-a, ali to još nije zakonski određeno.
Koliko	**2. korak: Smještanje podataka u tablice**

tablica?	U početku nije jasno u koliko tablica treba spremiti relevantne podatke. U jednu, dvije, tri ili više? Kako izbjeći redundanciju podataka, kako podatke organizirati na način da se podaci mogu čim lakše obrađivati i, što je najvažnije, da se mogu unositi kada je god to potrebno. U tu svrhu može poslužiti dijagram objekt-veza koji poznaje simbole pravokutnik, elipsu i romb. Podsjetimo se da su entiteti naši predmeti promatranja, dakle ono o čemu prikupljamo podatke. Kako doći do dijagrama objekt-veza koji je dobro polazište za kreiranje baze? U našem slučaju uočavamo primjerice tri entiteta: • Proizvodi • Kupci • Dobavljači

Početak gradnje dijagrama objekt-veza *Trgovine* može izgledati ovako:



	Oko entiteta se grupiraju njihova obilježja koja nazivamo atributima. Atribute u ovom slučaju tražimo u okviru *Relevantni podaci*. Pokušajte rasporediti one atribute za koje je jasno da pripadaju entitetima sa slike. Jeste li uspjeli smjestiti sve atribute? Koje niste?
Rješavanje n:n veze	Proizvode ne možemo direktno povezati s dobavljačima ni kupcima jer bismo u tom slučaju imali **n:n** vezu. Jedan se proizvod prodaje raznim kupcima (veza n), a jedan kupac kupuje n raznih proizvoda. Zašto je veza proizvoda s dobavljačima također n:n? U slučaju n:n veze ne možemo ni na jednu stranu dodati strani ključ jer bismo morali dodati više od jednog stranog ključa, što je, kako ćemo poslije vidjeti u postupku normalizacije, neprihvatljivo.

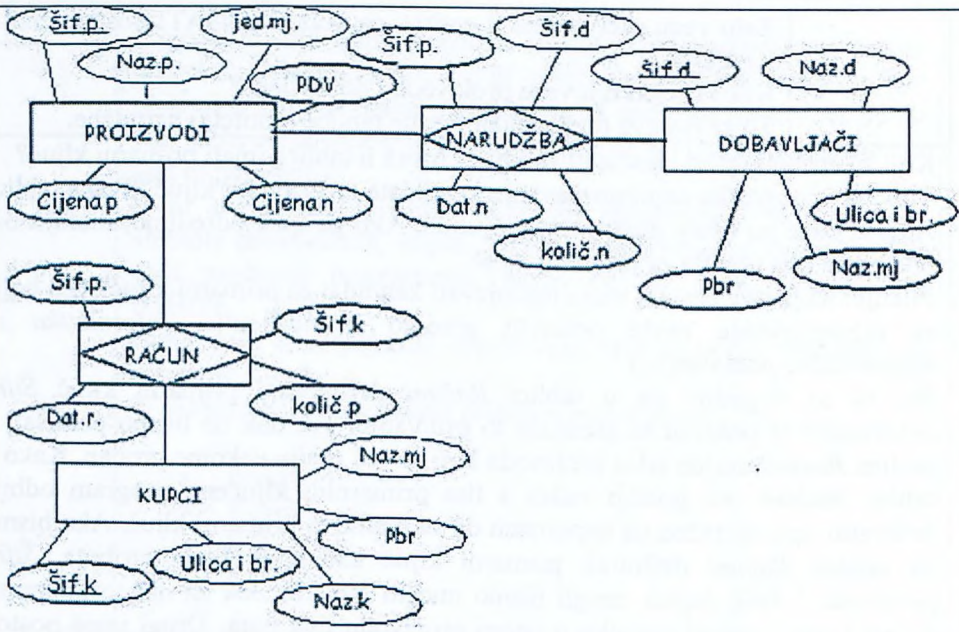
	Zato vezu pretvaramo u entitet, pa veza PRODATI postaje entitet RAČUN. Kako riješiti n:n vezu proizvoda i dobavljača? To se rješava dodavanjem vezne tablice (entiteta) narudžbe.

Koji su atributi dobri kandidati za ključ? Mora li tablica imati primarni ključ? Tipična je pogreška neprepoznavanje kandidata za primarni ključ. Svaka tablica Access baze ne mora imati primarni ključ. Ako ga ipak odredimo, moraju biti poštivana oba pravila za primarni ključ.

Pitanje: Koja dva pravila mora ispunjavati kandidat za primarni ključ? (Za bolje se razumijevanje može ponoviti gradivo iz udžbenika *Informatika za informatičko poslovanje*.)

Što bi se dogodilo da u tablici *Računi* definiramo primarni ključ *Šifra proizvoda*? U početku bi računalo to prihvatilo, sve dok ne bismo pokušali u tablicu *Računi* unijeti šifru proizvoda koji je već ranije nekome prodan. Kako u tablici *Računi* već postoji redak s tim primarnim ključem, program odbija prihvatiti upis za račun uz napomenu da se duplicira primarni ključ. Ako bismo za tablicu *Računi* definirali primarni ključ kao spoj dvaju atributa, *Šifra proizvoda* i *Šifra kupca*, mogli bismo unositi podatke dok za nekog kupca ne bismo trebali unijeti podatke o istom proizvodu dva puta. Drugi unos postoji kad kupac isti proizvod kupi više puta, što se u praksi stalno događa. Podatke bi o drugoj kupnji istog proizvoda program odbijao prihvatiti. Ključ složen od tri atributa (*Šifra proizvoda*, *Šifra kupca* i *Datum računa*) neće stvarati probleme dok se ne dogodi da isti kupac isti proizvod u istom danu kupuje dva puta, što je također realno očekivati pa ni to ne bi bio dobar kandidat za ključ. Dakle tablicu *Računi* može se ili definirati kao tablicu bez ključa, ili uvesti neki novi atribut, npr. *Broj računa* kao njezin ključ. Za naše prototipno rješenje ostavit ćemo je bez ključa.

Definicija primarnog ključa	Potrebno je definirati i primarni ključ za svaki od entiteta ili odlučiti da će tablica biti bez primarnog ključa. Po definiciji primarnog ključa oba svojstva za primarni ključ za entitet **PROIZVODI** zadovoljava atribut **ŠIFRA PROIZVODA.** Za entitet **DOBAVLJAČI** je primarni ključ **ŠIFRA DOBAVLJAČA .** Odredite primarni ključ za entitete **KUPCI, RAČUNI** i **NARUDŽBE.** (Rješenje: **ŠIFRA KUPCA** je dobar kandidat za entitet **KUPCI,** a ostale dvije nemaju dobrih kandidata za primarni ključ).



Poštujući pravila dobivamo:

a) Tablice *Proizvodi*, *Dobavljači*, *Kupci*, *Računi* i *Narudžbe*.
b) Zaglavlja stupaca postaju atributi iz dijagrama, npr. za tablicu *Kupci* imat ćemo pet stupaca: *Šifra kupca*, *Naziv kupca*, i tri koja se odnose na adresu: *Poštanski broj*, *Naziv mjesta* i *Ulicu i broj*.
c) Tablice *Računi* i *Narudžbe* su vezne tablice. Veze se u tim veznim tablicama uspostavljaju pomoću stranih ključeva. Tablica *Računi* je u n:1 relaciji s tablicom *Kupci* i *Proizvodi* (jedan se slog računa odnosi uvijek na jednog kupca i na jedan proizvod). Zato je *Šifra proizvoda* u tablici *Računi* strani ključ koji tu tablicu povezuje s tablicom *Proizvodi*, a *Šifra kupca* u tablici *Računi* strani ključ koji tu tablicu povezuje s tablicom *Kupci*.

Pitanja:

1. U kojoj se tablici, osim u tablici *Računi*, *Šifra proizvoda* pojavljuje kao strani ključ?
2. U toj se tablici pojavljuje još jedan strani ključ. Koji?
3. U kakvoj je relaciji tablica *Narudžbe* s tablicama *Proizvodi* i *Dobavljači*? Obrazloži.

Tablice *Proizvodi*, *Kupci*, *Dobavljači*, *Računi* i *Narudžbe*

Proizvodi : Table

Šifra proizvoda	Naziv proizvoda	Jedinica mjere	pdv	Prodajna cijena	Nabavna cijena
0001	Kruh bijeli	kom	0	6,00 kn	5,00 kn
0002	Pecivo	kom	0,22	1,00 kn	0,80 kn
0003	Keksi	kg	0,22	26,00 kn	20,00 kn
0004	Mlijeko	l	0	4,90 kn	4,00 kn
0005	Sir	kg	0,22	56,00 kn	48,00 kn

računi : Table

Šifra kupca	Šifra proizvoda	kol	datr
01	0001	2	1.5.2006
01	0002	2	1.4.2006
01	0001	1	2.5.2006
02	0001	1	1.6.2006
05	0001	1	1.3.2006

narudzbe : Table

sp	sd	Naručena količi	Datum narudžb
0001	01	200	1.2.2006
0002	01	100	1.2.2006
0001	02	50	2.2.2006
0003	03	150	3.2.2006
0003	04	32	3.2.2006
0004	05	24	4.2.2006
		0	

1 of 6

kupci : Table

Šifra kupca	Naziv kupca	pbr	mjesto	Ulica i broj
01	Pero	10000	Zagreb	Vinička 22
02	Edo	51000	Rijeka	Luki 33
03	Ada	52000	Pula	Starčevićeva 7
04	Kiki	51000	Rijeka	Krimeja 8
05	Vera	10000	Zagreb	Ilica 26

dobavljaci : Table

Šifra dobavljača	Naziv dobavljača	mjesto	Poštanski broj	Ulica i br
01	Agroprom	Zagreb	10000	Kostelska 5
02	Bionjam	Rijeka	51000	Tići 7
03	Ekonjup	Pula	52000	Nazorova 6
04	Finesa	Zagreb	10000	Mokrička 18
05	Kravica	Pula	52000	Livadska 8

Normalizacija podataka	**3. korak : Provjera tablica – postupak normalizacije** Sad se još pitamo jesu li podaci normalizirani i nalaze li se u trećoj normalnoj formi. Prije postupka normalizacije potrebno je odlučiti koji su atributi dobri kandidati za ključ.
Definicija prve normalne forme	Podaci su u **prvoj normalnoj formi** ako se mogu prikazati pomoću tablice u kojoj se nijedan atribut ili grupa atributa ne ponavljaju u istome retku, niti imamo slučaj da se u neko polje tablice upisuje više od jednog atributa (pojava tablice u tablici). U našem je slučaju to pravilo zadovoljeno. Za bolje se razumijevanje može ponoviti gradivo iz udžbenika *Informatika za informatičko poslovanje*.
Nedostaci ako podaci nisu u prvoj normalnoj formi	- Najveći nedostatak: Neke podatke koji se realno mogu pojaviti **nemamo gdje upisati**. Ako je predviđeno n mjesta za unos podatka, problem se javlja kod n+1 podataka. Ako se primjerice predvidi da se u tablicu *Proizvodi* upisuju i podaci o kupcu koji je taj proizvod kupio i predvidi se pet mjesta, može se unijeti pet kupaca za taj proizvod, ali se šesti nema gdje upisati. - Drugi nedostatak: Takvu je nenormaliziranu tablicu **teško pretraživati** jer za isti podatak treba pretraživati više stupaca.

	- Treći nedostatak: Takvu je nenormaliziranu tablicu **teško** obrađivati, npr. ažurirati.
Definicija druge normalne forme	Podaci su u **drugoj normalnoj formi** ako su u prvoj normalnoj formi, a osim toga za sve atribute vrijedi pravilo da ovise o cijelome ključu, a ne samo o dijelu ključa (u našem slučaju baze *Trgovine* i to je pravilo zadovoljeno).
Definicija treće normalne forme	Podaci su u **trećoj normalnoj formi** ako su u drugoj normalnoj formi (dakle i u prvoj), a osim toga vrijedi pravilo da nema funkcionalne zavisnosti među atributima. Ako pogledamo naše podatke i upitamo se u kojoj se normalnoj formi nalaze, vidjet će se da se oni nalaze drugoj, ali ne i u trećoj normalnoj formi.

Tablica *Kupci* je u drugoj, ali nije u trećoj normalnoj formi:

kupci				
Šifra kupca	**Naziv kupca**	**pbr**	**mjesto**	**Ulica i broj**
01	Pero	10000	Zagreb	Vinička 22
02	Edo	51000	Rijeka	Luki 33
03	Ada	52000	Pula	Starčevićeva 7
04	Kiki	51000	Rijeka	Krimeja 8
05	Vera	10000	Zagreb	Ilica 26

Pitanja za analizu tablica	Pokušajte na pitanja odgovoriti bez zirkanja u odgovore, a nakon što odgovorite, usporedite svoje odgovore s odgovorima iz knjige. 1. Zašto ova tablica nije u trećoj normalnoj formi? 2. Koji je atribut primarni ključ ove tablice? 3. Postoje li atributi koji funkcionalno zavise jedan o drugome mimo ključa? 4. Koji su to atributi? 5. Postoji li još jedna tablica u bazi s funkcionalnom zavisnosti atributa? 6. Koja je to tablica i koji su atributi zavisni? 7. Kako se ove tablice mogu svesti na treću normalnu formu?
Odgovori	1. Ova tablica nije u trećoj normalnoj formi jer u njoj postoje atributi koji ovise jedan o drugome. 2. Primarni ključ je *Šifra kupca*. 3. Da, atributi *Poštanski broj* i *Mjesto* ovise jedan o drugome. 4. To su *Poštanski broj* i *Mjesto*. 5. Da, postoji. 6. To je tablica *Dobavljači*, a atributi su također *Poštanski broj* i *Mjesto*.

Svođenje na treću normalnu formu:

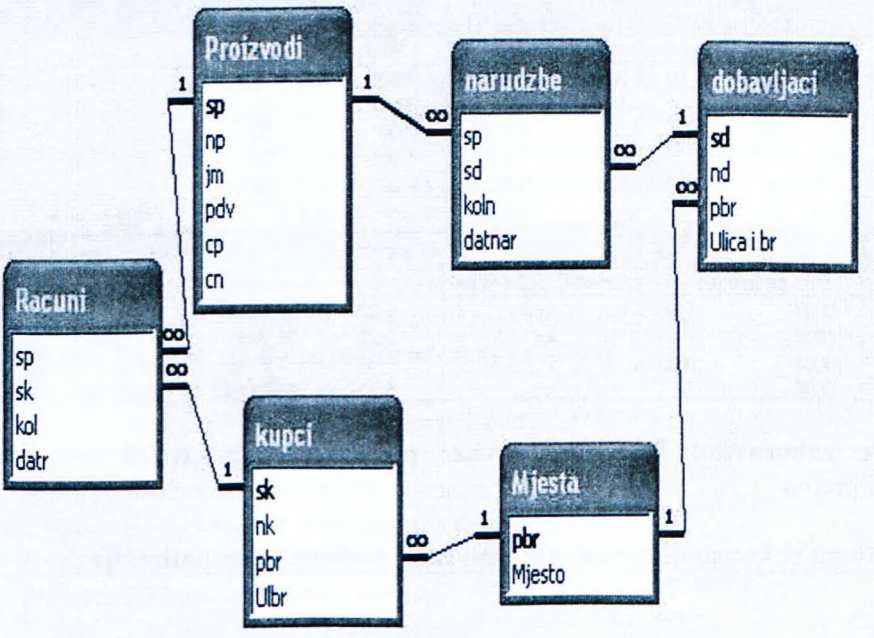
Problem se rješava tako da se tablica koja je u drugoj, ali ne i u trećoj normalnoj formi, rastavlja na dvije tablice: na tablicu *Kupci* (bez naziva mjesta) kojoj se dodaje tablica *Mjesta* u kojoj se naziv mjesta pojavljuje samo jedanput.

Mjesta	
Poštanski broj	**Mjesto**
10000	Zagreb
51000	Rijeka
52000	Pula

Ta će ista tablica *Mjesta* poslužiti i za svođenje na treću normalnu formu tablice *Dobavljači*.

Tablice koje zadovoljavaju pravila za projektiranje baze podataka prikazane su ovdje na dva načina.

1. Prozor veza (engl. *Relationhips*) iz programa Access



2. Tablice programa Access popunjene izmišljenim podacima.

Dobavljači : Table

Šifra dobavljača	Naziv dobavljača	Poštanski broj	Ulica i br
01	Agroprom	10000	Kostelska 5
02	Bionjam	51000	Tići 7
03	Ekonjup	52000	Nazorova 6
04	Finesa	10000	Mokrička 18
05	Kravica	52000	Livadska 8

Mjesta : Table

Poštanski broj	Mjesto
10000	Zagreb
51000	Rijeka
52000	Pula

Record: 1

Kupci : Table

Šifra kupca	Naziv kupca	pbr	Ulica i broj
01	Pero	10000	Vinička 22
02	Edo	51000	Luki 33
03	Ada	52000	Starčevićeva 7
04	Kiki	51000	Krimeja 8
05	Vera	10000	Ilica 26

Narudžbe : Table

sp	sd	Naručena količi	Datum narudžb
0001	01	200	1.2.2006
0002	01	100	1.2.2006
0001	02	50	2.2.2006
0003	03	150	3.2.2006
0003	04	32	3.2.2006
0004	05	24	4.2.2006

Racuni : Table

Šifra proizvoda	Šifra kupca	kol	datr
0001	01	2	1.5.2006
0001	01	1	1.6.2006
0001	02	1	1.3.2006
0002	03	2	1.4.2006
0001	05	1	2.5.2006

Record: 5 of 5

Proizvodi : Table

Šifra proizvoda	Naziv proizvoda	Jedinica mjere	pdv	Prodajna cijena	Nabavna cijena
0002	Pecivo	kom	0,22	1,00 kn	0,80 kn
0003	Keksi	kg	0,22	26,00 kn	20,00 kn
0004	Mlijeko	l	0	4,90 kn	4,00 kn
0005	Sir	kg	0,22	56,00 kn	48,00 kn

Ne zaboravite! Prije izrade baze podataka treba izvršiti organizacijske pripreme.

Pritom se koristi **dijagram objekt-veza** i **postupak normalizacije**.

Pitanja za ponavljanje uz poglavlje 1

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–g).

Pojam	*Objašnjenje pojma*
1. Normalizacija b	a) Razvojni softver za baze podataka.
2. Dijagram objekt-veza j	b) Postupak kojim se uređuju tablice na način da se otklanja štetna redundancija.
3. Entitet g	c) Atribut ili više atributa koji se odabiru radi jednoznačne definicije retka u tablici.
4. Atribut i	d) Primarni ključ druge tablice u bazi koji se koristi za uspostavljanje veze među tablicama.
5. Primarni ključ c	e) Svojstvo koje osigurava da se ne obrišu potrebni podaci.
6. Strani ključ d	f) Višak podataka.
7. Referencijalni integritet podataka e	g) Ono o čemu se podaci prikupljaju. U dijagramu se označava pravokutnikom, a u bazi postaje tablica.
8. Redundancija podataka f	h) Podaci i metapodaci organizirani za prikupljanje, pronalaženje i obradu podataka. Osim toga i razvojni softver za baze podataka.
9. Baze podataka h	i) Svojstvo koje opisuje entitet. U dijagramu se označava elipsom, a u bazi postaje naslov stupca u tablici.
10. Access, Oracle i DB 3+ a	j) Koristi se za dizajniranje baze podataka.

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Što je primarni ključ?
2. Treba li svaka tablica imati primarni ključ? NE
3. Što je strani ključ?
4. Kada se podaci nalaze u prvoj normalnoj formi?
5. Kada se podaci nalaze u drugoj normalnoj formi?
6. Kada se podaci nalaze u trećoj normalnoj formi?
7. Na temelju čega se kreiraju tablice? ~~ATRIBUT~~ DIN. DIA OBJ-V
8. Koliko će tablica biti u bazi podataka? KOLIKO IM-EM.
9. Koliko će stupaca imati tablica?
10. Čemu služi dijagram objekt-veza? SLUŽI ZA DIZAJNIRANJE BAZE P.
11. Koja je razlika između pojma podatak i atribut?

Pitanja za ponavljanje na temelju gradiva iz poglavlja i udžbenika *Informatika za informatičko poslovanje*:

1. Na kojoj se strani kod relacije n:1 nalazi primarni, a na kojoj strani ključ?
2. Koji se problemi javljaju ako podaci nisu u prvoj normalnoj formi?
3. Koji se problemi javljaju ako podaci nisu u drugoj ili trećoj normalnoj formi?

4. Jesu li podaci bolje uređeni kada se nalaze u prvoj ili u trećoj normalnoj formi?
5. Mogu li podaci biti u drugoj normalnoj formi, a da nisu u prvoj?
6. Mogu li podaci biti u drugoj normalnoj formi, a da nisu u trećoj?
7. Može li ključ biti složen od više atributa?
8. Kada je dobro tablicu raščlaniti na dvije tablice?
9. Koji postupak kao rezultat nudi raščlanjivanje tablice u više tablica?
10. Kako se u praksi rješava slučaj n:n veze?

Dobro pogledajte ove tablice i odgovorite na pitanja:

Proizvodi : Table

Šifra proizvoda	Naziv proizvoda	Jedinica mjere	pdv	Prodajna cijena	Nabavna cijena
0001	Kruh bijeli	kom	0	6,00 kn	5,00 kn
0002	Pecivo	kom	0,22	1,00 kn	0,80 kn
0003	Keksi	kg	0,22	26,00 kn	20,00 kn
0004	Mlijeko	l	0	4,90 kn	4,00 kn
0005	Sir	kg	0,22	56,00 kn	48,00 kn

računi : Table

Šifra kupca	Šifra proizvoda	kol	datr
01	0001	2	1.5.2006
01	0002	2	1.4.2006
01	0001	1	2.5.2006
02	0001	1	1.6.2006
05	0001	1	1.3.2006

narudzbe : Table

sp	sd	Naručena količi	Datum narudžb
0001	01	200	1.2.2006
0002	01	100	1.2.2006
0001	02	50	2.2.2006
0003	03	150	3.2.2006
0003	04	32	3.2.2006
0004	05	24	4.2.2006
		0	

1 of 6

kupci : Table

Šifra kupca	Naziv kupca	pbr	mjesto	Ulica i broj
01	Pero	10000	Zagreb	Vinička 22
02	Edo	51000	Rijeka	Luki 33
03	Ada	52000	Pula	Starčevićeva 7
04	Kiki	51000	Rijeka	Krimeja 8
05	Vera	10000	Zagreb	Ilica 26

dobavljaci : Table

Šifra dobavljača	Naziv dobavljača	mjesto	Poštanski broj	Ulica i br
01	Agroprom	Zagreb	10000	Kostelska 5
02	Bionjam	Rijeka	51000	Tići 7
03	Ekonjup	Pula	52000	Nazorova 6
04	Finesa	Zagreb	10000	Mokrička 18
05	Kravica	Pula	52000	Livadska 8

1. Koji bi atribut mogao biti primarni ključ u tablici *Proizvodi*? Zašto? D	a. *Šifra kupca*, jer ga sve n-torke imaju i različit je za svaki redak.
2. Koji bi atribut mogao biti primarni ključ u tablici *Kupci*? Zašto? A	b. Ne, jer se ponavlja ista u više redaka.
3. Koje tablice sadrže strani ključ? Koji su to atributi? E	c. Ne, jer postoje kupci s istim imenom, npr. Pero, Ivo, Ada...
4. Bi li *Naziv kupca* mogao biti primarni ključ u tablici *Kupaca*? Zašto? C	d. *Šifra proizvoda*, jer ga sve n--torke imaju i različit je za svaki redak.
5. Bi li *Mjesto* moglo biti primarni ključ u tablici *Dobavljača*? Zašto? B	e. *Narudžbe* (*Šifra dobavljača, Šifra proizvoda*) i *Računi (Šifra kupca* i *Šifra proizvoda*).

6. Bi li *Prodajna cijena* mogla biti primarni ključ u tablici *Proizvodi*? H	f. U drugoj, a ne u trećoj jer postoji ovisnost atributa *Mjesto* i *Poštanski broj* za tablice *Kupci* i *Dobavljači*.
7. U kojoj su normalnoj formi ove tablice? F	g. n:n
8. Koja je kardinalnost među tablicama *Dobavljači* i *Proizvodi*? G	h. Ne, jer se ponavlja isto u više redaka.

Pitanje za razmišljanje:

Kako biste tablice sveli na treću normalnu formu?

Primjeri iz prakse za vježbu:

1. Razgovorom s krajnjim korisnicima ustanovljeno je da se neki kupci (ne svi) pojavljuju i kao dobavljači. Također se neki dobavljači (opet ne svi) pojavljuju i kao kupci. Preradite dijagram objekt-veza na način da se u tom slučaju slog kupca koji je ujedno dobavljač, odnosno dobavljača, koji je ujedno kupac pojavljuje samo jednom, ali da postoji oznaka radi li se o kupcu, dobavljaču ili je kupac ujedno i dobavljač.

2. Izaberite primjer iz prakse za izradu baze podataka.
   - Izradite dijagram objekt-veza
   - Odredite primarne ključeve
   - Odredite kardinalnosti veza
   - Izvršite normalizaciju tablica
   - Tablice popunite izmišljenim podacima

Napomena: Ova će se baza u poglavljima koja slijede stalno dograđivati pa je ovaj zadatak važno izraditi čim pažljivije.

Preporučena literatura za daljnje istraživanje:

1. Čičin-Šain, M, Vukmirović, S, Čapko, Z, *Informatika za informatičko poslovanje*, Rijeka, Fortuna d.o.o, 2006.
2. Laudon, K.C, Laudon, J.P, *Esentials of Management Information Systems*, New Jersey, Prentice Hall Corporation, 2003.
3. www.prenhall.com/laudon
4. Varga, M, *Baze podataka; Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb, DRIP, 1994.

2. Microsoft Access baza podataka – uvod

Microsoft Access je interaktivni sustav za razvoj, izgradnju, korištenje i održavanje relacijskih baza podataka u Windows okruženju. Za razliku od drugih programa za rad s bazama podataka, u Accessu se cijela baza podataka nalazi u jednoj datoteci s nastavkom *.mdb* (engl. *Microsoft Data Base*).	
Osnovne mogućnosti Accessa	Access omogućava prikupljanje, spremanje i organiziranje podataka, te njihovu obradu. - Podaci se spremaju i čuvaju u tablicama (engl. *Tables*). - Radi lakšeg unosa podataka koriste se obrasci (engl. *Forms*) koji omogućavaju da se podaci unose putem obrazaca čija se slika vidi na ekranu računala. - Prikaz podataka iz jedne ili više tablica uz obradu podataka (sortiranje, filtriranje, ažuriranje, dodavanje, brisanje, zbrajanje, računanje vrijednosti po formulama i sl.) moguć je pomoću upita (engl. *Queries*). - Za ispis dokumenata na pisaču koriste se izvješća (engl. *Reports*). - Uz Access se može koristiti i pisanje programskog koda na više načina: pomoću makronaredbi (ili akcija), pomoću jezika upita (SQL) i pomoću jezika VBA (engl. *Visual Basic for Aplications*).
Relacijske baze podataka	Access je program za rad s relacijskim bazama podataka. U relacijskoj su bazi podaci vezani relacijama i strukturirani tako da se osigurava minimalna zalihosnost (redundancija, ponavljanje) podataka. Time se osigurava postojanost i integritet podataka u bazi.
Osnovni objekti u Accessu	Baza podataka Access se sastoji od šest osnovnih objekata. To su: - Tablice (engl. *Tables*) - Upiti (engl. *Queries*) - Maske za unos podataka (engl. *Forms*) - Izvješća (engl. *Reports*) - Makroprogrami koji sadrže kreirane rutine i procedure (engl. *Macros*) - Moduli – dijelovi koji sadrže izvorni kod Access Basic jezika (engl. *Modules*).
Pomoć	Za Access bazu postoji mnogo literature, a ona uvijek pristupačna je u izborniku Access pomoći (engl. *Help*). Ta je literatura dostupna uvijek kad se koristi program i dobro je naviknuti se na nju čim prije.

Otvaranje programa Access	Najbrži način otvaranja programa Access je klik na ikonu programa Access ili na neku Access bazu. U tom je slučaju potrebno spomenutu ikonu ili njezinu prečicu smjestiti na radni stol (engl. *Desktop*). Drugi je način otvaranje programa u izborniku *Programs* na gumbu *Start*.
Kreiranje baze podataka	Nakon što je poslovanje istraženo i kada je odlučeno koji će se podaci nalaziti u kojoj tablici, tek tada ima smisla kreirati bazu podataka. Kreirati bazu znači dati bazi ime i pripremiti je da može primiti svoje objekte. Svaka Access baza ima nastavak *.mdb*.
Primjer kreiranja baze	Nakon što je Access program pozvan, nova se baza kreira sljedećim postupkom: 1. Izabrati za novu bazu *Blank Database*. 2. U prozor *File Name* upisati (dodijeliti) naziv bazi podataka poduzeca1.mdb. 3. Kliknuti na gumb *Create*. Time je kreirana ljuska prazne baze podataka koja u sebi nema ni jedan objekt. Nakon kreiranja bazu možemo spremiti, pozivati i puniti je objektima. Napomena: Iako Access prihvaća i naša slova poput č i ć, bolje je za sve nazive objekata koristiti slova engleske abecede jer mogu nastupiti problemi pri prijenosu baze s računala na računalo.
Kreiranje baza uz pomoć čarobnjaka	Postoji mogućnost kreiranja Access baza uz pomoć čarobnjaka ili predložaka. Nude se gotove baze koje korisnik može izabrati i koristiti. U fazi učenja rada s bazama korisno je istražiti koje se mogućnosti nude, ali u praksi ta mogućnost nije previše upotrebljiva jer je ono što se nudi često nepotrebno, a ono što je potrebno se ne nudi.

Pitanja za ponavljanje uz poglavlje 2

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–a).

Pojam	*Objašnjenje pojma*
1. Access d	a) Automatizirani postupci koji pomažu izgraditi objekte.
2. .mdb k	b) Veličina Access baze ne može prijeći 2 GB.
3. Čarobnjaci a	c) Objekti koji služe za ispis podataka na pisač.
4. Pomoć j	d) Objekti koji služe za izradu grafičkog sučelja na ekranu monitora.
5. Izvješća c	e) Objekti koji služe za spremanje i čuvanje podataka.
6. Upiti f	f) Objekti koji služe za prikaz podataka iz jedne ili više tablica.
7. Tablice e	g) Postupak koji imenuje bazu i stvara okvir za prihvaćanje drugih objekata iz baze.
8. Moduli i	h) Objekti sastavljeni iz postupaka zvanih akcije.
9. Makroi h	i) Objekti sastavljeni iz *visual basic* naredbi.
10. Kreiranje baze g	j) Upute za rad sa bazama koji se nalaze u razvojnom alatu Access.
11. Ograničenje baze b	k) Nastavak uz ime Access baze.

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Kako se poziva program Access?
2. Kako se kreira nova baza podataka? FILE, NEW, BLANK DATE
3. Koji nastavak ima svaka Access baza?
4. Čemu služi pomoć unutar Access programa?
5. Koji su osnovni gradbeni objekti Access programa?
6. Kako se u Accessu može programirati?
7. Može li postojati baza koja ne sadrži ni jednu tablicu? DA
8. Može li ime baze sadržati slova hrvatske abecede?
9. Zašto se preporuča da nazivi objekata baze ne sadrže naša posebna slova, poput č i ć?
10. Što je .mdb?
11. Postoje li ograničenja na veličinu baze podataka?

Pitanja za ponavljanje na temelju gradiva iz poglavlja i udžbenika *Informatika za informatičko poslovanje* i druge preporučene literature:

1. Koje programe za baze podataka poznajete osim Accessa?
2. Pročitajte u Access pomoći (engl. *Help*) osnovne napomene uz Access.
3. Što je relacijska baza podataka?
4. Može li se Access baza graditi na osnovu postojeće baze?
5. Koje su anomalije moguće ako bazu počnemo graditi bez prethodne normalizacije?

Primjeri iz prakse za vježbu:

1. Kreirajte praznu ljusku baze *Trgovina* i spremite je pod imenom Trgovina1.mdb.
2. Zatvorite bazu koju ste upravo kreirali.
3. Pozovite bazu.
4. Preimenujte bazu tako da se zove Trgovina.mdb.
5. Kreirajte praznu ljusku baze koju ste zamislili unutar zadataka 1.1, primjer iz prakse 2.
6. Istražite mogućnosti koje nudi čarobnjak za kreiranje baze. Opišite mogućnosti koje se nude (one koje ste razumjeli).

Preporučena literatura za daljnje istraživanje:

1. Čičin-Šain, M, Vukmirović, S, Čapko, Z, *Informatika za informatičko poslovanje,* Rijeka, Fortuna d.o.o, 2006.
2. Laudon, K. C, Laudon, J. P, *Esentials of Management Information Systems*, New Jersey, Prentice Hall Corporation, 2003.
3. www.prenhall.com/laudon
4. Varga, M, *Baze podataka; Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb, DRIP, 1994.
5. Pomoć unutar programa Access.

3. Tablice

Tablice	Prvi objekti koji se dodaju u bazu najvjerojatnije su **tablice**. U bazi će biti onoliko tablica koliko ima entiteta u dijagramu objekt-veza nakon postupka normalizacije. Tablice (engl. *Table*) su skupovi podataka o pojedinom entitetu, npr. o poduzećima, djelatnicima, robama, kupcima. Svaki je entitet predstavljen atributima (obilježjima) koji ga opisuju. Tablice se sastoje od redaka (engl. *Rows*) i stupaca (engl. *Columns*). Prije kreiranja tablice korisno je izraditi dijagram objekt-veza. Tablice su, dakle, osnovni objekt relacijskih baza podataka koji je sastavljen od redaka i stupaca te sadrži podatke i metapodatke, pri čemu su metapodaci podaci o podacima (npr. naziv tablice, naziv polja, vrsta polja, podaci o ključu i sl).
Stupci	Stupci tablice sadržavaju istovrsne podatke i predstavljaju obilježja (atribute) tablice (npr. nazivi poduzeća, djelatnosti, prihodi, rashodi). Tablica će imati onoliko stupaca koliko atributa postoji za taj entitet u dijagramu objekt-veza.
Redak	Tablica se puni podacima tako da se upisuje redak po redak. Iz dijagrama objekt-veza nije vidljivo koliko će redaka imati tablica. Svaki se redak sastoji od n podataka pa se redak zove i n-torka, a koriste se još i sinonimi zapis ili slog (engl. *Record*). Reci sadrže podatke o jednom subjektu (npr. proizvodu pecivo), a stupci o obilježju ili atributu (npr. o prodajnoj cijeni)

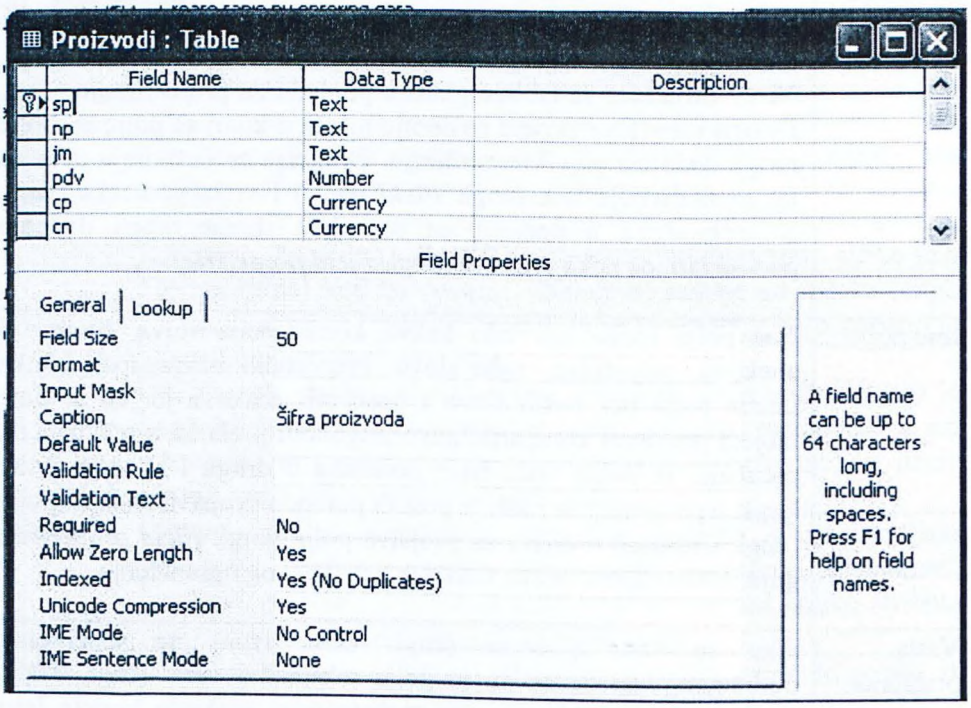
Primjer: Tablica *Poduzeća*

Proizvodi					
Šifra proizvoda	**Naziv proizvoda**	**Jedinica mjere**	**pdv**	**Prodajna cijena**	**Nabavna cijena**
0001	Kruh bijeli	kom	0	6,00 kn	5,00 kn
0002	Pecivo	kom	0,22	1,00 kn	0,80 kn
0003	Keksi	kg	0,22	26,00 kn	20,00 kn
0004	Mlijeko	l	0	4,90 kn	4,00 kn
0005	Sir	kg	0,22	56,00 kn	48,00 kn

Kreiranje strukture tablice	Prije unosa tablice potrebno je kreirati strukturu tablice, dakle upisati naslove stupaca odnosno polja i definirati ta polja po tipu i eventualno po još nekom svojstvu. Pritom se definira i atribut ili grupa atributa koji će biti primarni ključ.

Primjer: Kreiranje tablice Poduzeća	Na primjeru to izgleda ovako: U bazi *Trgovina.mdb* želimo kreirati tablicu *Proizvodi*. U bazi se bira objekt tablica (engl. *Tables*) i opcija za novu tablicu (engl. *New*). Struktura se tablice gradi u pogledu za projektiranje (engl. *Design view*). Na ekranu se dobije mreža u kojoj se polje po polje mogu dodavati pojedina obilježja. Obilježja se definiraju tako da im se dodjeljuje ime (engl. *Field name*) i vrsta podataka (engl. *Data type*), a u donjem im se dijelu sheme mogu dodatno podešavati još neka svojstva (engl. *Field properties*). 1. PITANJE
Ime polja	Ime polja može biti bilo kakva kombinacija slova, brojeva i znakova, uključujući naša slova. Preporučili bismo ipak kratka imena polja bez naših slova i posebnih znakova jer će se time izbjeći problemi kompatibilnosti prigodom prelaska s računala na računalo, iz jedne vrste baze podataka u drugu i sl. Ako želite ljepši ispis polja, to možete postići pomoću svojstva za ispis polja (engl. *Caption*) u dijelu za svojstva polja (engl. *Field properties*) gdje možete koristiti sve znakove bez ikakvog ograničenja.
Vrsta podataka	Ako se vrsta podatka (engl. *Data type*) ne specificira, podrazumijevat će se da je polje tekstualne vrste (engl. *Text*). Vrste podataka koje se, osim tekstualnih, najčešće koriste jesu brojčane (engl. *Numeric*), logičke (engl. *Yes/No*), datumske (engl. *Date*), novčane (engl. *Currency*) i još neke. Kao brojčana potrebno je definirati polja s kojima se računa, a ne sva polja koja sadrže brojke. Tako npr. telefonski broj nije dobro definirati kao brojčano polje jer se ne bi mogle upisati nule na početak broja, a to nije ni potrebno jer se s njim ionako ne obavljaju računske operacije.

Primjer: Kreiranje tablice *Proizvodi* (ime tablice – *Proizvodi*)



Biranje opcije unosa	Postupak je sljedeći: 1. Kliknuti na gumb *Tables* i gumb *New*. 2. U okviru za dijalog *New Table* kliknuti na opciju *Design View*. Nudi se i opcija upisa direktno u tablicu, ali se ona ne preporuča jer se tako ne može definirati tip podataka ni primarni ključ. 3. Otvara se okvir za upis strukture tablice (engl. *Table Design*). Gornji se dio okvira sastoji od tri stupca u koji se upisuju nazivi obilježja (engl. *Field Name*), definira tip polja za svako pojedino obilježje (engl. *Data Type*) i upisuju komentari, tj. opisi obilježja (engl. *Description*).
Upis imena polja	4. U prvi redak stupca *Field name* upisati naziv obilježja *sp* (*Šifra proizvoda*). Time smo dali spretno i kratko ime polja. Za ispis želimo ipak opširniji opis npr. *Šifra proizvoda*. Taj ćemo natpis upisati u polje natpisa (engl. *Caption*) u donji dio opisa svojstva polja (engl. *Field properties*). Naziv obilježja je mogao biti dulji i sadržavati naša slova, ali je takve dugačke nazive nespretno koristiti, npr. u programima, a s hrvatskim slovima može doći do problema pri prelasku s računala na računalo, pri prelasku na drugu bazu podataka i sl.

Izbor vrste polja	5. U prvi redak stupca *Data Type* definirati tip polja za obilježje *sp*. Klikom se na trokutić u polju *Data Type* javljaju ponuđeni tipovi podataka: *Text* (znakovni), *Number* (brojčani), *Date/Time* (datumski), *Yes/No* (logički) i ostali tipovi podataka. Za obilježje *sp* izabrat će se *Text*. Iako je šifra broj, s tim se brojem ne računa, a da smo izabrali brojčani tip, ne bi bilo moguće upisati vodeće nule.
Svojstva polja	6. U donjem su dijelu prozora *Table design* nabrojene karakteristike svojstava (engl. *Properties*) za upisana obilježja. Unos i promjena podataka u poljima svojstava vrijedi za ono obilježje (engl. *Field*) u kojem je pokazivač pozicioniran. Ta su svojstva različita ovisno o tipu podataka. Ovisno o zadanim karakteristikama obilježja definiraju se (popunjavaju) svojstva u svojstvima ili se polja tih svojstava ostavljaju nepopunjena (nepromijenjena).
Promjena veličine polja	7. U primjeru za obilježje *sp* u svojstvu *Field Size* (veličina polja) umjesto ponuđene vrijednosti 50 (znakova) upisat će se vrijednost 4.

Proizvodi : Table

Field Name	Data Type	Description
sp	Text	
np	Text	
jm	Text	
pdv	Number	
cp	Currency	
cn	Currency	

Field Properties

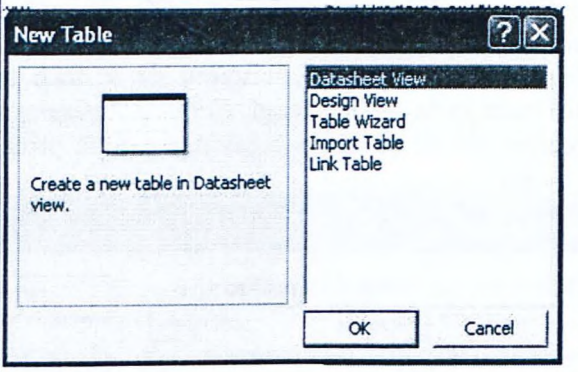
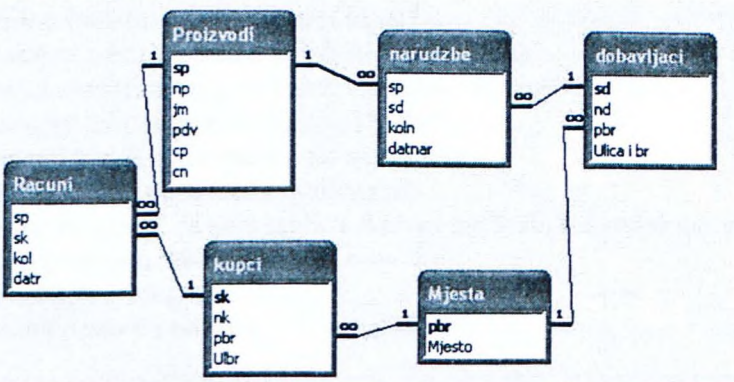
General | Lookup

Field Size	4
Format	
Input Mask	
Caption	Šifra proizvoda
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None

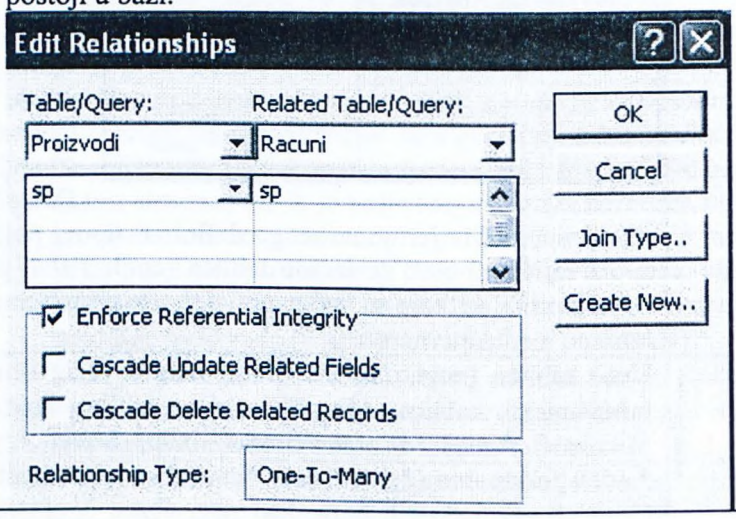
A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Polje za naslove	8. U polju se *Caption* može upisati naziv obilježja koje će se javiti na maski tablice (primjerice *Šifra proizvoda*, tako da u maski umjesto *sp*, koji je upisan kao ime obilježja nakon otvaranja tablice, u zaglavlju piše *Šifra proizvoda*).
Sljedeće obilježje	9. Nakon popunjavanja svojstava za prvo upisano obilježje (u primjeru *sp*), klikne se u novi redak stupca *Field Name* i u skladu se s opisanim postupkom upisuju podaci za sljedeće obilježje.

Primarni ključ	10. Nakon što se upišu sva obilježja i za njih definiraju potrebna svojstva, označava se primarni ključ tablice koji jedinstveno identificira svaki zapis u tablici. Primarni se ključ može sastojati od samo jednog obilježja ili od više obilježja čija kombinacija vrijednosti jednoznačno određuje zapise u tablici. Primarni ključ ne dopušta da se nađu dva ista zapisa u tablici. U dizajnu tablice u primjeru će se kao primarni ključ označiti obilježje *sp*. Ključ se označava tako da se označi redak obilježja *sp* i klikne na ikonicu ključa. Sličica ključa pojavljuje se uz to obilježje.
Zatvaranje i spremanje	11. Tablica se zatvara klikom na znak x u gornjem desnom kutu. U prozoru se *Table Name* upisuje naziv tablice: *Proizvodi*. Napomenimo još jednom da Access prihvaća i nazive tablice s našim slovima, ali ih je bolje izbjegavati u nazivima objekata jer može doći do problema pri prelasku s računala na računalo ili na novu inačicu softvera. Time je tablica kreirana, ali ona u tom trenutku ima samo zaglavlja, a nema još podatke za pojedine proizvode (u našem primjeru pet proizvoda).
Unos podataka u tablicu	Za unos je podataka potrebno tablicu otvoriti (engl. *Open*). Na ekranu se pojavljuje tablica spremna za unos podataka. Podaci se unose redak po redak i na kraju se tablica zatvori.
Promjena strukture tablice	Struktura se tablice može naknadno mijenjati, npr. dodati ili ukloniti neki stupac, promijeniti tip podatka ili njegov naziv. Kad se god mijenja struktura tablice, poziva se rad u pogledu za oblikovanje (engl. *Design view*).
Obrada tablica	Što se može raditi s tablicom? Kada smo napravili tablicu, možemo je spremiti, ponovo učitati, podatke iz tablice učitati, ispisati, sortirati, mijenjati, prepisati u Excel ili Word... No prava se korist od tablica javlja tek u suradnji s drugim Accessovim objektima, a to su upiti, obrasci, izvješća, makroi i moduli.

Kako se još može doći do tablica?	Kada se izabere opcija za gradnju nove tablice (engl. *New Table*), iz okvira se tablice (engl. *Tables*) javlja nekoliko mogućnosti.  Do sada smo koristili način gradnje u pogledu za gradnju (engl. *Design View*). Način direktnog upisa, gdje se tablica kreira stvarajući je (engl. *Datasheet View*), nije preporučljiv jer se ne mogu definirati tipovi polja ni njihova svojstva. Čarobnjak za izradu tablica (engl. *Table Wizard*) nudi nekoliko tablica koje su već napravljene za prototipnu bazu, što je korisno u edukativne svrhe.
Uvoz tablica	Uvoz tablica (engl. *Import Table*) moguć je iz drugih baza ili proračunskih tablica. Moguć je uvoz tablica koje su formata Microsoft Acces s nastavcima (*.mdb; *.adp; *.mda; *.mde; *.ade), tekstualne (*.txt; *.csv; *.tab; *asc), proračunskih tablica (*.xls) i nekih drugih baza, kao npr. dbase 3+ (+.dbf), Paradox baza, SQL baza i sl. Prednost uvoza tablica je da se u tom slučaju mogu koristiti već ranije uneseni podaci, što je korisno za velike baze. Rezultat je unosa kreiranje tablice s podacima u Access bazi koja je spremna za uporabu.
Povezivanje tablica	Povezivanje tablica (engl. *Link Table*) slično je uvozu, samo u tom slučaju tablica ostaje tamo gdje je kreirana, a u bazi se koristi preko veze kad je potrebna.
Prozor veza 	Nakon što su tablice kreirane, moguće ih je međusobno povezati u prozoru veza (engl. *Relationships*). Tablice baze *Trgovine* povezane su kako se vidi na slici: 

Potrebno je otvoriti prozor veza izborom ikone veza ili opciju *Relationships* u izborniku alata (engl. *Tools*). Veze se uspostavljaju povlačenjem miša od polja do polja, npr. *sp* u tablicama *Proizvodi* i *Računi*. U prozoru uređivanja veza (engl. *Edit Relationships*) može se definirati da se traži referencijalni integritet baze za te tablice (engl. *Enforce Referencial Integrity*), što osigurava da se ne može pojaviti račun za proizvod koji ne postoji u bazi.



Pitanja za ponavljanje uz poglavlje 3

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–e).

Pojam	*Objašnjenje pojma*
1. Tablica g	a) Postupak za brzo generiranje preddefiniranih tablica.
2. Stupac f	b) Podaci s kojima se namjerava računati.
3. Redak e	c) Označi se atribut ili grupa atributa koji su podesni za primarni ključ pa se na traci s alatima izabere ikona ključa.
4. Čarobnjak za izradu tablica a	d) U Accessu je moguća iako se ne preporuča.
5. Numerički podaci b	e) N-torka podataka koji su isti po vrsti, ali različiti po sadržaju.
6. Naziv polja j	f) U naslovu je naziv obilježja (atribut) ili drugi uneseni natpis, a slijede pojedini podaci za to obilježje.
7. Definicija primarnog ključa c podataka	g) Osnovni objekt relacijskih baza podataka, sastavljen od redaka i stupaca, a sadrži podatke i metapodatke.
8. Tablica bez primarnog ključa d	h) Access prihvaća te znakove, ali ih je bolje izbjeći.
9. Naša slova u nazivu polja h	i) Svojstvo koje osigurava da se ne brišu slogovi čiji su podaci korišteni u povezanim slogovima drugih tablica.
10. Referencijalni integritet i	j) Sažeto ime obilježja koje se koristi u cijeloj bazi.

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Kako se kreira nova tablica?
2. Čemu služi čarobnjak za izradu tablica? ZA POJED.
3. Kako se dodaje već postojeća tablica koja sadrži podatke koje ne želimo nanovo upisivati? UVOZ
4. Koje podatke definiramo kao brojčane (numeričke)?
5. Koja pravila vrijede za nazive polja (atribute) u tablicama?
6. Što se upisuje u svojstvo natpisa (engl. *Caption*)?
7. Kako se atribut označava kao primarni ključ? DESN KLIK NA REDAK i ~~[illegible]~~ EDIT → PRIMARY KEY
8. Može li postojati tablica koja ne sadrži niti jedan redak?
9. Može li postojati tablica koja ne sadrži niti jedan stupac? NE
10. Može li postojati tablica bez primarnog ključa?
11. Može li ime tablice ili polja sadržavati naša slova?
12. Kako se uspostavljaju veze među tablicama?
13. Kako možemo osigurati da se u tablicu *Računi* može dodati redak samo za kupce koji postoje u tablici *Kupci*? ENFORCE
14. Što je referencijalni integritet?
15. Što su u tablici *Računi* podaci, a što metapodaci?

Pitanja za ponavljanje na temelju gradiva iz poglavlja i udžbenika *Informatika za informatičko poslovanje* i druge preporučene literature:

1. Kako u tablicu možete dodati redak?
2. Kako u tablicu možete dodati stupac?
3. Kako možete obrisati redak tablice? Koje retke u tablicama nećete moći izbrisati?
4. Kako možete obrisati stupac tablice? Koje stupce u tablicama nećete moći izbrisati?
5. Koje vrste polja postoje?
6. Koliko tablica može postojati u bazi?

Primjeri iz prakse za vježbu:

1. Pročitajte u Access pomoći (engl. *Help*) osnovne napomene uz tablice.
2. Kreirajte tablice baze *Trgovina.mdb* i popunite ih podacima na temelju dijagrama objekt-veza i normaliziranih tablica baze iz poglavlja *Priprema za korištenje baze*.
3. Popunite praznu ljusku baze koju ste zamislili unutar zadataka 1.1, primjer iz prakse 2.

Preporučena literatura za daljnje istraživanje:

1. Čičin-Šain, M, Vukmirović, S, Čapko, Z, *Informatika za informatičko poslovanje*, Rijeka, Fortuna d.o.o, 2006.
2. Laudon, K. C, Laudon, J. P, *Esentials of Management Information Systems*, New Jersey, Prentice Hall, Corporation, 2003.
3. www.prenhall.com/laudon
4. Varga, M, *Baze podataka; Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb, DRIP, 1994.
5. Pomoć unutar programa Access.

4. Upiti

Upiti	**Upiti** (engl. *Queries*) su objekti baze koji se grade na temelju podataka koji se nalaze u tablicama. Tablice su temeljne sastavnice relacijske baze podataka. Tablice se međusobno povezuju relacijama. Upit služi za dobivanje željenih informacija koje su pohranjene u bazi podataka. Podaci koji čine potrebnu informaciju najčešće su razmješteni u više tablica pa je, da bi se ta informacija dobila, potrebno povezati tablice. Upit omogućava prikupljanje podataka iz više tablica kako bi se dobila relevantna (tražena) informacija. Odgovor na svaki upit može biti nova tablica. Ta je tablica rezultat upita, dinamičke je prirode i zato se zove dinamički skup podataka (engl. *Dynaset*). Vrijednosti upita nisu fiksne, nego se mijenjaju ovisno o promjeni podataka u tablicama nad kojima je sastavljen upit. Primjerice, u upit se mogu povezati tablice o proizvodima i kupcima. Ako se pojavi novi kupac, dobavljač ili proizvod i ako se obavi prodaja ili nabava robe, evidentiranje će se provesti u odgovarajućim tablicama, a automatski će se prikazati i u upitima koji se nanovo stvaraju na osnovi podataka iz tablica.
Relacije	Povezivanje tablica omogućava da se na temelju zadanog upita prikupljaju relevantni podaci koji su razmješteni u različitim tablicama. Pri povezivanju tablica potrebno je definirati odnos (relaciju) između tablica. Osnovni tipovi relacija su: - 1:1 - n:1 - 1:n - n:n.
Ključevi	Povezivanje se tablica uspostavlja pomoću ključeva, pri čemu je: - primarni ključ na strani 1 - strani ključ na strani n. Primjerice u bazi *Trgovina* neke tablice imaju primarni ključ, a neke nemaju. Za vježbu se podsjetite što je primarni ključ koje od tablica i dopunite tablicu primarnim ključevima: Tablica | Primarni ključ Proizvodi | Kupci | Dobavljači | Računi Narudžbe
Odgovori: Proizvodi – Šifra proizvoda, Kupci – Šifra kupca, Dobavljači – Šifra dobavljača, Računi i Narudžbe – Tablice bez primarnog ključa	

Dobavljači : Table

Šifra dobavljača	Naziv dobavljača	Poštanski broj	Ulica i br
01	Agroprom	10000	Kostelska 5
02	Bionjam	51000	Tići 7
03	Ekonjup	52000	Nazorova 6
04	Finesa	10000	Mokrička 18
05	Kravica	52000	Livadska 8

Mjesta : Table

Poštanski broj	Mjesto
10000	Zagreb
51000	Rijeka
52000	Pula

Record: 1

Kupci : Table

Šifra kupca	Naziv kupca	pbr	Ulica i broj
01	Pero	10000	Vinička 22
02	Edo	51000	Luki 33
03	Ada	52000	Starčevićeva 7
04	Kiki	51000	Krimeja 8
05	Vera	10000	Ilica 26

Narudžbe : Table

sp	sd	Naručena količi	Datum narudžb
0001	01	200	1.2.2006
0002	01	100	1.2.2006
0001	02	50	2.2.2006
0003	03	150	3.2.2006
0003	04	32	3.2.2006
0004	05	24	4.2.2006

Racuni : Table

Šifra proizvoda	Šifra kupca	kol	datr
0001	01	2	1.5.2006
0001	01	1	1.6.2006
0001	02	1	1.3.2006
0002	03	2	1.4.2006
0001	05	1	2.5.2006

Record: 5 of 5

Proizvodi : Table

Šifra proizvoda	Naziv proizvoda	Jedinica mjere	pdv	Prodajna cijena	Nabavna cijena
0002	Pecivo	kom	0,22	1,00 kn	0,80 kn
0003	Keksi	kg	0,22	26,00 kn	20,00 kn
0004	Mlijeko	l	0	4,90 kn	4,00 kn
0005	Sir	kg	0,22	56,00 kn	48,00 kn

Na temelju ovih tablica moguće je odgovoriti na pitanje koje su proizvode kupili pojedini kupci?

Za odgovor na ovo pitanje treba pogledati sve tablice. Pregledom tablice *Računi* doznaje se količina, pregledom tablice *Kupci* naziv kupca, tablice *Proizvodi* naziv proizvoda i prodajna cijena, a tablice *Mjesta* naziv mjesta iz kojeg je kupac. Za kratke se tablice od nekoliko kupaca, proizvoda i računa odgovor može dati i ručnim pregledavanjem tablica. No ručno bi pregledavanje tablica i traženje odgovora bilo vrlo mukotrpno za tablice od nekoliko stotina ili tisuća kupaca i stotinu ili više proizvoda i nekoliko tisuća računa. Ovdje može pomoći automatska tehnika traženja odgovora pomoću objekata koji se zovu upiti.

Vrste upita	Postoje razne vrste upita. Možemo ih podijeliti • ovisno o načinu nastanka • ovisno o njihovoj funkciji • ovisno o načinu upotrebe, odnosno trenutku kada se upisuje podatak s kojim se uvjet izvodi
Podjela po načinu nastanka	Po načinu nastanka upiti se dijele na: a) Upite kreirane pomoću razvojne mreže za gradnju upita (engl. *Design Grid*) koja se zove i mreža upita pomoću primjera (engl. *Query by Example – QBE*).

	b) Upite kreirane pomoću strukturiranog jezika upita (engl. *Structured Query Language-SQL*) Suvremene baze podataka koriste tehnike koje olakšavaju gradnju upita pomoću raznih tehnika prilagođenih GUI sučelju. Tako Access ima pomoć za gradnju upita pomoću razvojne mreže u koju se donose tablice, polja, upisuju uvjeti i sl. U pozadini se zatim automatski stvara kod u jeziku upita SQL. Oni korisnici koji znaju koristiti SQL jezik mogu, naravno, upit odmah napisati u SQL jeziku. Svaki upit generiran u mreži ima svoj SQL kod, ali obrat ne vrijedi. Samo se jednostavniji upiti stvoreni u SQL jeziku mogu predstaviti u QBE mreži. Dakle, u SQL je jeziku moguće stvoriti upite koji su presloženi da bi se prikazali u mreži. Takvi upiti zovu se i upiti svojstveni za SQL (engl. *SQL Specific Query*).
Podjela po funkciji upita	Upiti koji se grade pomoću razvojne mreže za gradnju upita (engl. *Design Grid*), koja se zove i mreža upita pomoću primjera (engl. *Query by Example – QBE*), postaju sve popularniji jer je takve upite korisniku dosta lako napraviti. Moguće je sagraditi dosta složen upit koji koristi nekoliko tablica ili drugih upita, pri čemu se koriste uvjeti koji neka polja i slogove isključuju, neka polja prikazuju zbirno i sl, a pritom ne treba ništa programirati, nego samo izabrati među ponuđenim opcijama ono što korisniku treba. Po funkciji se Access upiti dijele na: a) Upit izborom (engl. *Select Query*) b) Ukršteni upit (engl. *Crostab Query*) c) Djelatni upit (engl. *Action Query*) c1) Upit koji stvara tablicu (engl. *Make Table Query*) c2) Upit koji mijenja podatke (engl. *Update Query*) c3) Upit koji dodaje slogove (engl. *Append Query*) c4) Upit koji briše slogove (engl. *Delete Query*) Upiti izborom i ukršteni upiti po prirodi su dinamički skupovi podataka (engl. *Dynaset*). U bazi je upit zapisan u jeziku upita, a odgovor se generira kad se upit poziva. Tako nastaje odgovor koji sliči tablici, ali se ne upisuje kao tablica primjerice na čvrsti disk, nego postoji u memoriji računala samo toliko dugo dok se upit ne zatvori. Djelatni upiti kao rezultat upita ostavljaju trag na tablicama nakon što se upit izvrši. Ime je djelatnog upita ujedno i opis toga što pojedini djelatni upit radi tablicama.

Parametarski upiti	Kod parametarskih se upita uvjet ne upisuje u polje uvjeta, nego se upis ostavlja za kasnije, za trenutak korištenja, pa upit postaje šire primjenjiv. Parametarski upiti mogu biti upiti izborom, ukršteni ili djelatni; radi se samo o tome da se podatak s kojim se ispituje uvjet upisuje nakon pozivanja uvjeta, a ne prilikom njegove gradnje. Parametarske je upite moguće sagraditi i u SQL jeziku.
Čarobnjak za izradu upita	Postoji nekoliko čarobnjaka za izradu upita. Za vježbu je korisno pokušati sagraditi nekoliko upita pomoću njih. Ipak, slično kao i kod tablica, u praksi je bolje upite graditi u pogledu za projektiranje (engl. *Design view*) jer u tom slučaju korisnik dobije upravo one podatke koje je tražio i u obliku kakvom je tražio.
Povezivanje tablica	Tablice je međusobno moguće povezati u trenutku gradnje upita, bez obzira na koji se način grade. Ipak je mnogo bolje da se tablice povežu u kartici za povezivanje tablica (engl. *Relationship*).
Naziv upita	Naziv upita može sadržavati bilo kakve znakove uključujući naša slova, ali je bolje da sadrži samo slova engleske abecede. To je ograničenje dobro poštivati da se izbjegnu problemi pri prijenosu baze s jednog računala na drugo.

Pitanja za ponavljanje uz poglavlje 4

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 8–e).

Pojam	*Objašnjenje pojma*
1. Upit f	a) Postupak za brzo generiranje preddefiniranih upita.
2. QBE g	b) Upit koji uzrokuje promjene u tablicama.
3. SQL i	c) Mogu se koristiti, ali ih je bolje izbjegavati.
4. Čarobnjak za izradu upita a	d) Ako se koristi, rezultirat će ispisom samo onih podataka koji ga ispunjavaju
5. Djelatni upit b	e) Može sadržavati naša slova, ali je bolje izbjegavati ih.
6. Parametarski upit h	f) Access objekt koji nastaje na osnovi podataka iz jedne ili više tablica.
7. Dinamički skup podataka j	g) Upit primjerom – postupak za generiranje upita bez programiranja.
8. Naziv upita e	h) Upit gdje se podatak za provjeru uvjeta zadaje nakon pozivanja upita.
9. Naša slova u nazivu upita c	i) Programski jezik za postavljanje uvjeta.
10. Uvjet b	j) Rezultat upita sličan tablici koji postoji samo dok je upit aktivan.

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Čemu služe upiti?
2. Kakve vrste upita postoje?
3. Što je QBE?
4. Što je SQL?
5. Može li se svaki upit stvoren pomoću QBE mreže prikazati u SQL?
6. Može li se svaki upit stvoren pomoću SQL prikazati u QBE mreži?
7. Čemu služi parametarski upit?
8. Stvara li upit izborom novu tablicu u bazi tipa .mdb?
9. Postoji li upit koji stvara novu tablicu?
10. Što je dinamički skup podataka i kako nastaje?
11. Zašto se preporuča izbjegavati naša slova u nazivu upita?
12. Što su upiti svojstveni za SQL?

Zadaci za proširenje znanja i razmišljanje na temelju gradiva iz druge preporučene literature:

1. U pomoći programa Access pročitajte članak o upitima (engl. *About types of queries*).
2. Koja je tehnika stvaranja upita spretnija za korisnika – pomoću razvojne mreže (engl. *Design Greed* ili *QBE*) ili pomoću SQL-a?

Primjeri iz prakse za vježbu:

1. Kakve biste upite, osim u poglavlju opisanih, mogli zamisliti nad podacima baze *Trgovina.mdb*?
2. Možete li zamisliti neke upite nad podacima vaše baze ako ste je kreirali nakon obrade poglavlja 3?

Preporučena literatura za daljnje istraživanje:

1. Čičin-Šain, M, Vukmirović, S, Čapko, Z, *Informatika za informatičko poslovanje*, Rijeka, Fortuna d.o.o, 2006.
2. Laudon, K. C, Laudon, J. P, *Esentials of Management Information Systems*, New Jersey, Prentice Hall Corporation, 2003.
3. www.prenhall.com/laudon
4. Varga, M, *Baze podataka; Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb, DRIP, 1994.
5. Pomoć unutar programa Access.
6. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.

4.1 Upiti izborom

Upiti izborom	**Upiti izborom** najjednostavniji su i najčešće korišteni upiti, pa ako se počne graditi upit bez izbora vrste upita, podrazumijeva se da se gradi upit izborom. Upit izborom može se graditi pomoću SQL jezika, čarobnjaka ili pomoću mreže za gradnju upita (engl. *Design Grid* ili *QBE Query by Example*). Razvoj pomoću mreže je jednostavan, a prilično moćan pa će primjeri biti obrađeni na taj način.

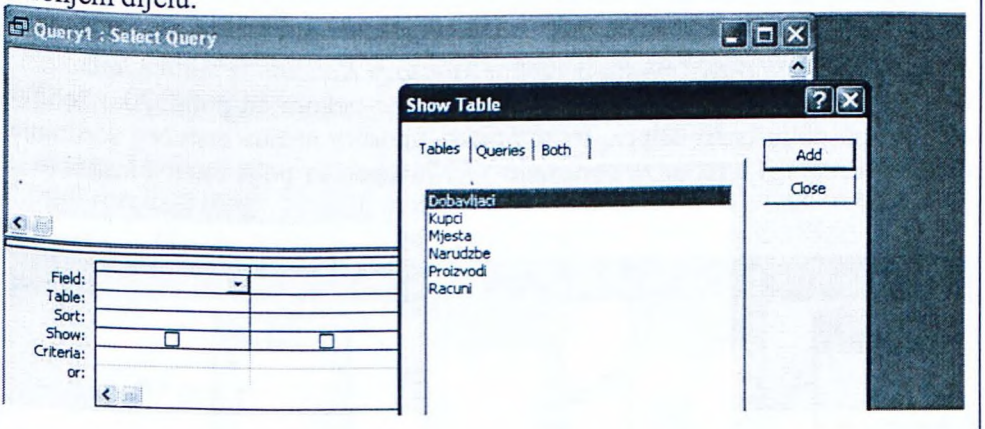
Koje su proizvode kupili pojedini kupci?

tkosto

Naziv kupca	Naziv proizvoda	Mjesto	kol	Prodajna cijena
Pero	Kruh bijeli	Zagreb	2	6,00 kn
Edo	Kruh bijeli	Rijeka	1	6,00 kn
Ada	Pecivo	Pula	2	1,00 kn
Pero	Kruh bijeli	Zagreb	1	6,00 kn
Vera	Kruh bijeli	Zagreb	1	6,00 kn

Podatke koje smo prije tražili u trima tablicama imamo sada u jednoj. Kako doći do takve tablice? U osnovnom je okviru Access potrebno izabrati upite (engl. *Queries*). Zatim je potrebno izabrati gumb za novi upit (engl. *New*) i unutar njega opciju vlastitog razvoja upita (engl. *Design* view).

Pojavljuje se radna površina za izradu upita preko primjera (engl. *QBE – Query by Example*) koja ima prostor za prikaz tablica u gornjem dijelu i mrežu za polja u donjem dijelu.



U početku se pojavljuje i okvir za izbor tablica sa svim tablicama koje postoje za bazu. Izaberu se tablice nad kojima se upit gradi tako da se pomoću gumba dodaj (engl. *Add*) dodaju na radnu površinu unutar QBE mreže, a na kraju se taj okvir zatvara gumbom zatvori (engl. *Close*).

Taj je dio isti za sve upite pa se prikazuje samo za ovaj prvi primjer.

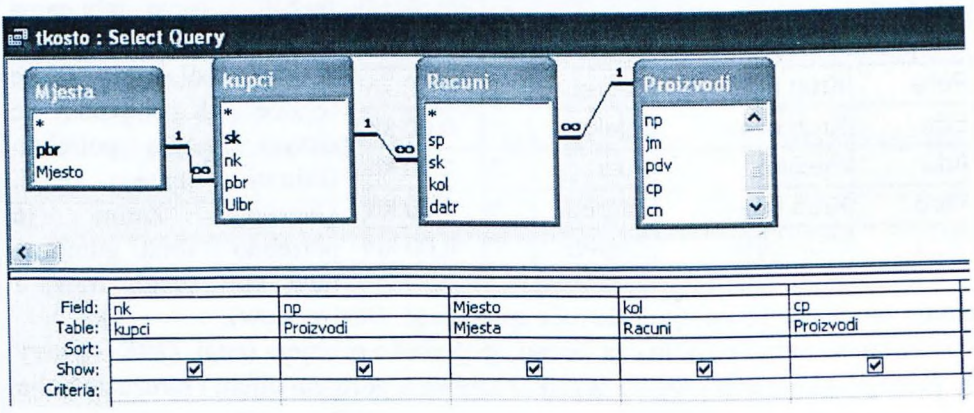
Ako naknadno želimo dodati neku tablicu, ponovno se otvara prozor za prikaz tablica (engl. *Show Table*) pomoću gumba na traci s alatima.

Kako se povezuju ključevi?

Ključevi se povezuju automatski ako imaju isti naziv (npr. *Šifra proizvoda* sp iz tablica *Proizvodi* i *Računi*); ako li nemaju isti naziv, potrebno ih je povezati ručno. Mogu se povezati tako da se u prozoru upita polja iz tablica spoje mišem, tako da se klikne prvo polje i bez otpuštanja tipke spoji s drugim poljem.

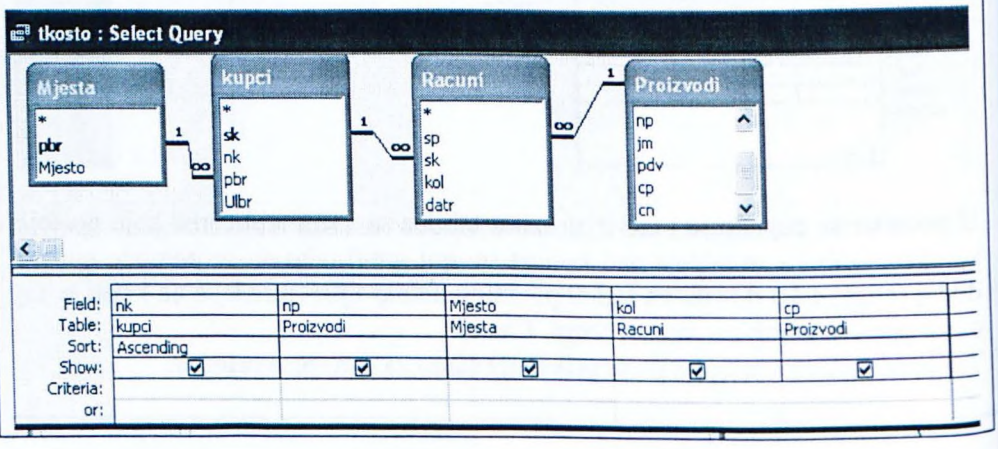
Osim toga moguće je vezu uspostaviti u prozoru veza (engl. *Relationships*), što je bolji način jer vrijedi za sve objekte u bazi i ne treba ih povezivati svaki put, pa su u ovom primjeru povezani na taj način.

Nakon izbora tablica, potrebno je metodom *izaberi* i *dovuci* izabrati polja iz tablica koja želimo u našem upitu. Ako se sve ispravno napravi, rezultat je:



Vidimo da je polje *Naziv kupca* iz tablice *Kupci*, *Naziv proizvoda* i *Cijena* iz tablice *Proizvodi*, *Naziv mjesta* iz tablice *Mjesta*, a *Količina* iz tablice *Računi*.

Upit se može doraditi, npr. tako da se sortira po nekom od polja. Ako želimo sortirati po polju naziv kupca, to se postiže unosom načina rastućeg sortiranja (engl. *Ascending*) u retku za sortiranje *Sort* i stupcu za polje naziva kupca (*nk*) iz mreže.



Nakon izrade je potrebno upit spremiti i imenovati, a kad ga se pozove, dobije se rezultat sličan prijašnjem, samo sortiran po nazivu kupca.

tko sto				
Naziv kupca	**Naziv proizvoda**	**Mjesto**	**kol**	**Prodajna cijena**
Ada	Pecivo	Pula	2	1,00 kn
Edo	Kruh bijeli	Rijeka	1	6,00 kn
Pero	Kruh bijeli	Zagreb	1	6,00 kn
Pero	Kruh bijeli	Zagreb	2	6,00 kn
Vera	Kruh bijeli	Zagreb	1	6,00 kn

Taj se upit sprema u bazu podataka, ali ne i tablica, koja se dobiva kao njegov rezultat. Tablica se pri svakom pozivu upita generira u memoriji računala i ne upisuje se ni u jednu tablicu na bilo kojem nositelju podataka.

Na temelju tablica iz baze moguće je primjerice odgovoriti i na druga pitanja:

Tko je što kupovao, ali zbirno? (da se npr. Pero ne pojavljuje dvaput)
Koji su kupci kupili pecivo?
Koliko su pojedini kupci ukupno potrošili?
Koliko je bilo kupaca iz Zagreba?
itd.

Na svako se od ovih pitanja može odgovoriti pomoću Access upita.

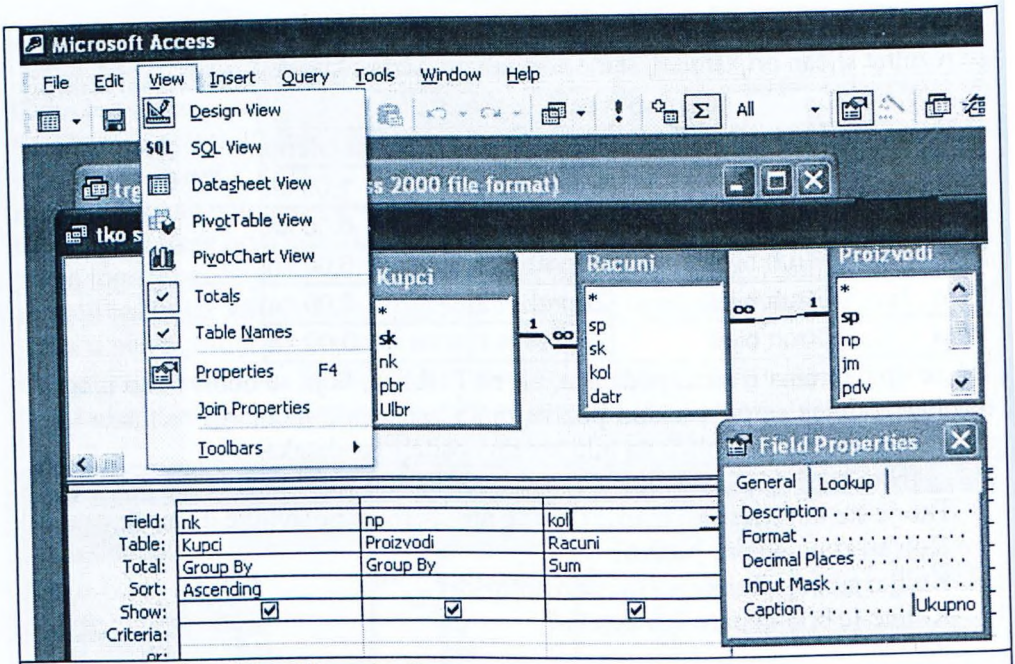
Tko je sto kupovao, ali zbirno?

tko sto zbirno		
Naziv kupca	**Naziv proizvoda**	**ukupno**
Ada	Pecivo	2
Edo	Kruh bijeli	1
Pero	Kruh bijeli	3
Vera	Kruh bijeli	1

Slično prethodnome, ovaj upit pokazuje tko je što kupovao, ali se za kupca istog artikla pojavljuje samo jedan redak pa Pero ima podatak da je kupio tri kruha, a ne kao prije da je kupio dva, pa posebno još jedan.

Na početku se gradnje upita pozivaju tablice, odabiru polja i kriterij sortiranja. Dodatno se u izborniku pogleda (engl. *View*) aktivira opcija za prikaz zbirnih rezultata (engl. *Totals*), čime se u mreži QBE pojavljuje redak za zbirne rezultate. U tom se retku za polje količine izabere zbroj (engl. *Sum*). Za isto polje se u svojstva polja (engl. *Field Properties*) za natpis unese *Ukupno* da bi to bio natpis nad tim stupcem u prikazu.

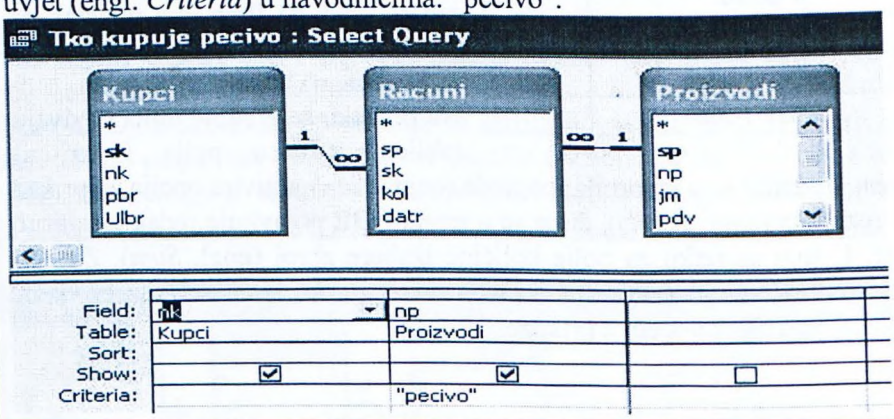
CAPTION



Koji su kupci kupili pecivo?

Tko kupuje pecivo	
Naziv kupca	Naziv proizvoda
Ada	Pecivo

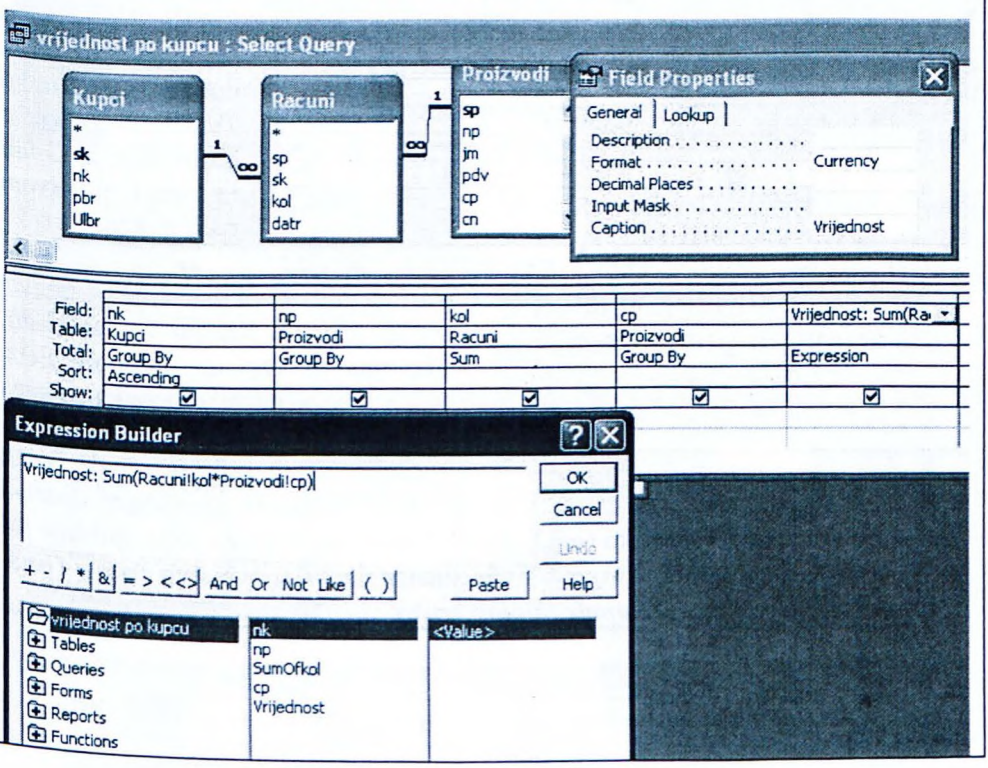
Kako dobiti ispis samo onih kupaca koji kupuju jedan od artikala, npr. pecivo? Na početku se gradnje upita pozivaju tablice, odabiru polja i u polje *Naziv proizvoda* dodaje uvjet (engl. *Criteria*) u navodnicima: "pecivo".



Koliko su pojedini kupci ukupno potrošili?

vrijednost po kupcu				
Naziv kupca	Naziv proizvoda	Ukupno količina	Prodajna cijena	Vrijednost
Ada	Pecivo	2	1,00 kn	2,00 kn
Edo	Kruh bijeli	1	6,00 kn	6,00 kn
Pero	Kruh bijeli	3	6,00 kn	18,00 kn
Vera	Kruh bijeli	1	6,00 kn	6,00 kn

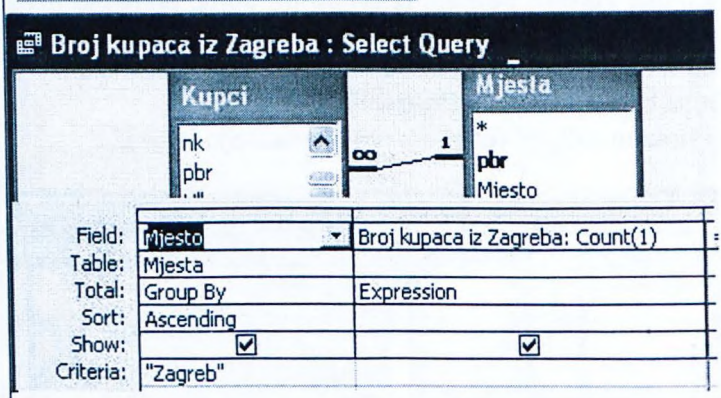
Za taj je upit, pored tehnika koje su poznate iz prijašnjih upita, potrebno još i kreirati polje *Vrijednost* koje se dobiva množenjem količine i prodajne cijene. Za to se koristi graditelj izraza (engl. *Expression Builder*). Količina se uzima iz tablice *Računi*, a prodajna cijena iz tablice *Proizvodi*.
Polja se biraju mišem i tipkom zalijepi (engl. *Paste*) iz graditelja izraza.



Koliko je bilo kupaca iz Zagreba?

Broj kupaca iz Zagreba	
Mjesto	**Broj kupaca iz Zagreba**
Zagreb	2

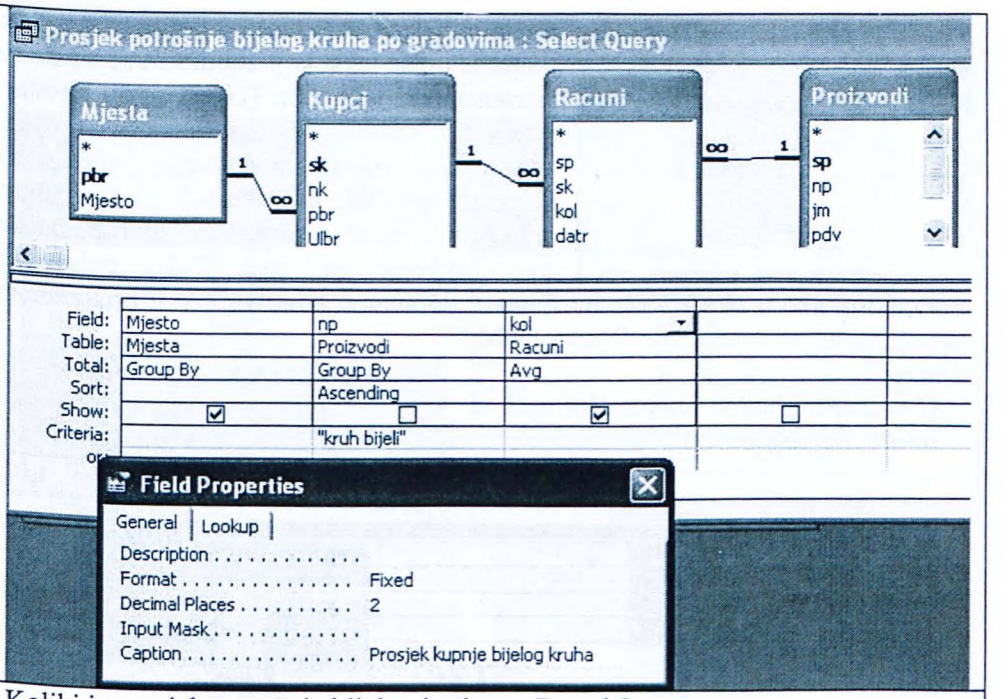
Ovaj se upit gradi na temelju podataka iz tablica *Kupci* i *Mjesta*. Iz *Mjesta* se koristi naziv mjesta, a iz *Kupci* samo veza. U polje koje se gradi izrazom upiše se bilo što (ovdje upisano 1) i upotrijebi *Count* iz *Totals*. Na temelju se toga automatski dobije da se u retku *Totals* pojavi *Expresion*, a *Count* uz izraz u prvom redu mreže.



Koliko prosječno kruha kupuje kupac iz raznih gradova?

Prosjek potrošnje bijelog kruha po gradovima	
Mjesto	**Prosjek kupnje bijelog kruha**
Rijeka	1,00
Zagreb	1,33

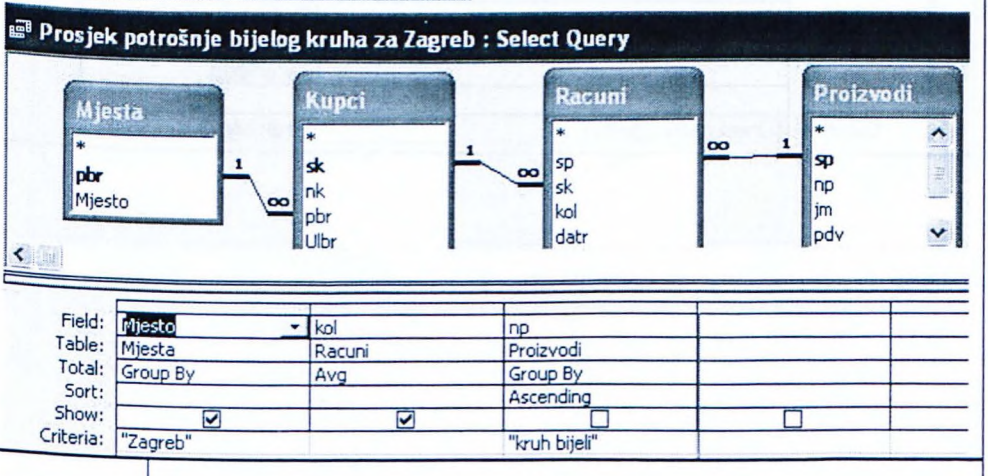
Ovaj se upit gradi na temelju tablica *Mjesta*, *Kupci*, *Računi* i *Proizvodi*. Kako nam u svakom retku ne treba pisati naziv proizvoda, u retku prikaza (engl. *Show*) deaktiviramo prikaz polja. U redak zbirnih prikaza (engl. *Total*) za polje količina iz tablice *Računi* izaberemo računanje prosjeka (engl. *Avg*). U okvir za svojstva polja upišemo još da se radi o decimalnom broju s dvije znamenke, a u svojstvo natpis (engl. *Capture*) upiše se *Prosjek kupnje bijelog kruha*.



Koliki je prosjek potrošnje bijelog kruha za Zagreb?

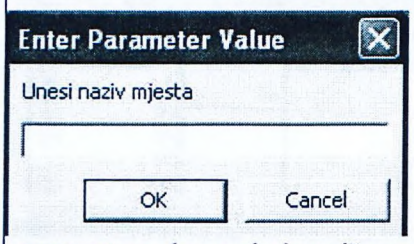
Prosjek potrošnje bijelog kruha za Zagreb	
Mjesto	**Prosjek kupnje bijelog kruha**
Zagreb	1,33

Ovaj se upit od prethodnog razlikuje samo po tome što je uz polje naziva mjesta dodan uvjet (engl. *Critera*) "Zagreb".



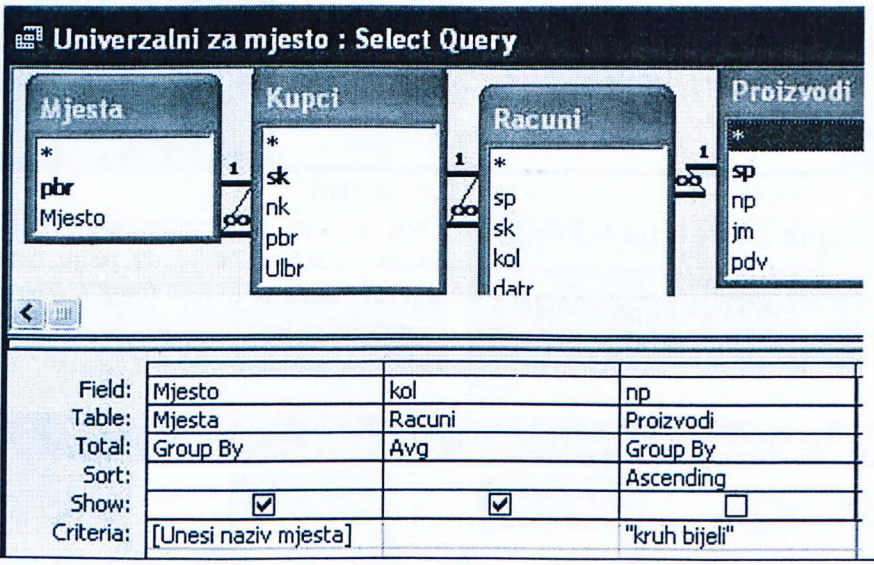
Parametarski upit	Korisniji bi bio univerzalni upit gdje se kriterij ne ugrađuje, nego ostavlja slobodan za trenutak obrade. Takav se upit zove **parametarski upit**.

Gradi se isto kao prethodni, samo se u polje uvjeta umjesto "Zagreb" u uglate zagrade upiše tekst koji želimo da se ispiše uz parametarski upit, npr. [Unesi naziv mjesta]. Kada se otvori upit, najprije se pojavljuje okvirić za unos parametarske vrijednosti (engl. *Enter Parameter Value*), a nakon upisa mjesta i kompletan dinamički skup podataka kao odgovor za upit. Tako ovisno o parametru, ako u okvir upišemo Zagreb, dobivamo odgovor kao u prethodnom primjeru, ali ako unesemo Rijeka, dobit ćemo:



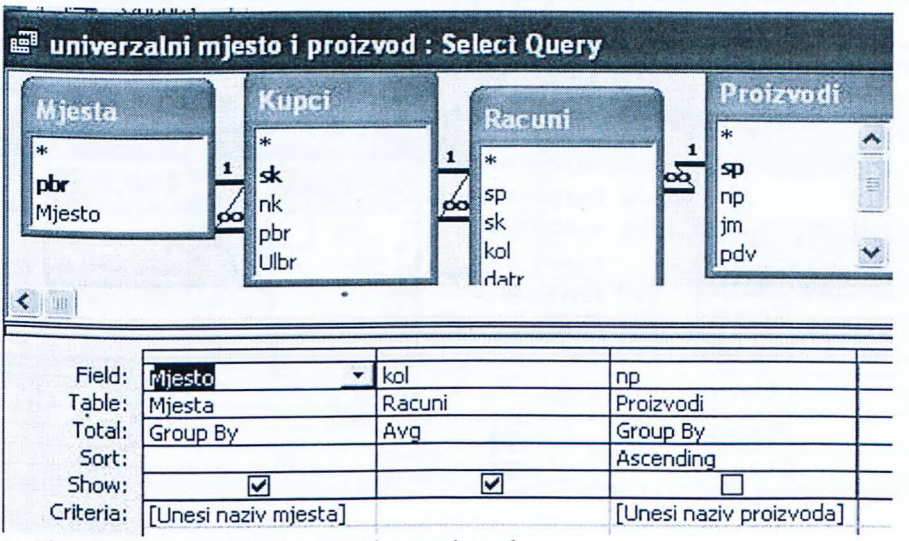
Univerzalni za mjesto	
Mjesto	**Prosjek kupnje bijelog kruha**
Rijeka	1,00

U pogledu gradnje univerzalni upit za mjesto izgleda ovako:



Field:	Mjesto	kol	np
Table:	Mjesta	Racuni	Proizvodi
Total:	Group By	Avg	Group By
Sort:			Ascending
Show:	☑	☑	☐
Criteria:	[Unesi naziv mjesta]		"kruh bijeli"

Moguće je za isti upit u retku kriterija postaviti više parametara pa npr. upit:

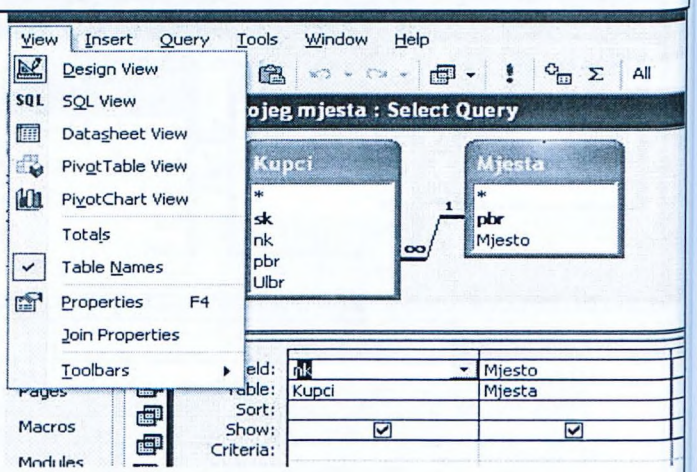


traži najprije unos mjesta, a zatim proizvoda.

SQL jezik je strukturirani jezik upita (engl. *Structured Query Language*). Taj je jezik vrlo raširen i koristi se unutar mnogih alata za rad s bazama podataka.

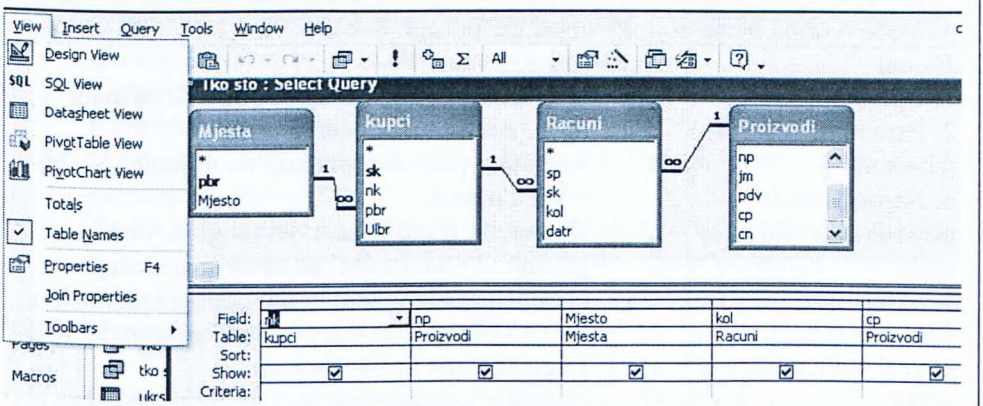
Interesantno je pogledati kako upiti stvoreni unutar Access alata izgledaju u SQL jeziku.

Iz kojeg mjesta	
Naziv kupca	**Mjesto**
Pero	Zagreb
Vera	Zagreb
Edo	Rijeka
Kiki	Rijeka
Ada	Pula



Vrlo jednostavan upit "Iz kojeg mjesta" gradi se na temelju dvije tablice, Mjesta i Kupci. Ako nas zanima vidjeti upit u SQL jeziku postupak je slijedeći:
Iz izbornika pogleda (engl. *View*) izabere se SQL pogled i dobije se upit u SQL jeziku:
SELECT Kupci.nk, Mjesta.Mjesto FROM Mjesta INNER JOIN Kupci ON Mjesta.pbr = Kupci.pbr;

Slično se može pogledati SQL prikaz svih upita iz ovog poglavlja. Npr. kako u SQL-u izgleda upit "Tko sto"? Otvorimo taj upit u pogledu gradnje.



Zatim se u izborniku pogled (engl. *View*) izabere SQL pogled. Time se dobije taj isti upit u SQL jeziku.

SELECT kupci.nk, Proizvodi.np, Mjesta.Mjesto, Racuni.kol, Proizvodi.cp FROM Proizvodi INNER JOIN (Mjesta INNER JOIN (kupci INNER JOIN Racuni ON kupci.sk = Racuni.sk) ON Mjesta.pbr = kupci.pbr) ON Proizvodi.sp = Racuni.sp;

Pitanja za ponavljanje uz poglavlje 4.1

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 8–e).

Pojam	*Objašnjenje pojma*
1. Upit izborom	a) Upit izborom kod kojeg se podatak za uvjet upisuje u trenutku poziva upita.
2. Parametarski upit izborom	b) Rezultat upita izborom koji ne mijenja sadržaj tablica u bazi.
3. Najbolji način povezivanja tablica u upitu	c) Upisuje se u redak za uvjet (engl. *Criteria*) pri gradnji uvjeta. Podaci se za uvjet u slučaju parametarskog uvjeta mogu upisati i naknadno.
5. SQL	d) Pomoću kartice za veze (engl. *Relationship*).
4. Najlakši način povezivanja tablica u upitu	e) Englesko ime upita izborom.
5. Dinamički skup podataka	f) Najčešće korišten upit koji se podrazumijeva ako nije drugačije definirano. Stupci su izabrana polja iz tablica ili upita nad kojima se upit gradi ili se dobivaju računskim putem, a reci se odabiru ovisno o ispunjenom uvjetu.
6. Naša slova u nazivu upita	g) Povezivanje mišem prilikom gradnje upita ili automatski ako se polja isto zovu.
7. Uvjet	h) Dozvoljena, ali se ne preporučuju.
8. Select Query	i) Jezik upita

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Čemu služe upiti izborom?
2. Kako se grade upiti izborom?
3. Nad kojim se objektima grade upiti izborom?
4. Može li se upit graditi nad samo jednom tablicom?
5. Može li se upit graditi nad drugim upitom?
6. Može li se svaki SQL upit prikazati u QBE mreži? SAMO JEDNOSTAVNE
7. Može li se upit graditi bez osnove u tablicama ili drugim upitima?
13. Može li upit izborom biti parametarski? M
14. Mora li upit imati ime?
15. Kako se povezuju tablice koje se koriste u upitu?
16. Može li se upit izborom prikazati u SQL obliku? MOŽE
17. Stvara li upit izborom novu tablicu u bazi tipa .mdb?
18. Kako glasi englesko ime upita izborom? SELECT QUERY
19. Može li se u upitu izborom računati kumulativne vrijednosti? MOGU
20. Može li se u upit izborom upisivati formula za računanje? MOŽE
21. Koja je razlika između parametarskog upita i upita dobivenog unosom uvjeta?

Zadaci za proširenje znanja i razmišljanje na temelju gradiva iz poglavlja i iz druge preporučene literature:

1. Kreirajte sve upite iz poglavlja 4.1.
2. U pomoći programa Access pročitajte članak o upitima izborom (engl. *Create a select or crosstab query*).
3. Kreirajte upite iz poglavlja 4.1. pomoću čarobnjaka za izradu upita. Koje niste mogli kreirati?
4. Usporedite SQL prikaze upita koje ste kreirali pomoću mreže i čarobnjaka. Jesu li isti?
5. Na temelju tablica koje ste razvili kao rezultat zadatka nakon poglavlja 3 o tablicama izgradite barem tri razna upita koristeći i uvjete, parametarski upit i graditelj izraza.

Preporučena literatura za daljnje istraživanje:

1. Čičin-Šain, M, Vukmirović, S, Čapko, Z, *Informatika za informatičko poslovanje*, Rijeka, Fortuna d.o.o, 2006.
2. Varga, M, *Baze podataka; Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb, DRIP, 1994.
3. Pomoć unutar programa Access.
4. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
5. Šavle, S, *Access*, Rijeka, Adamić, 2004.
6. Pejić-Bach, M, Spremić, M, Vlahović, N, *Priručnik za vježbe iz informatike*, Zagreb, Mikrorad, 2003.

4.2 Ukršteni upiti

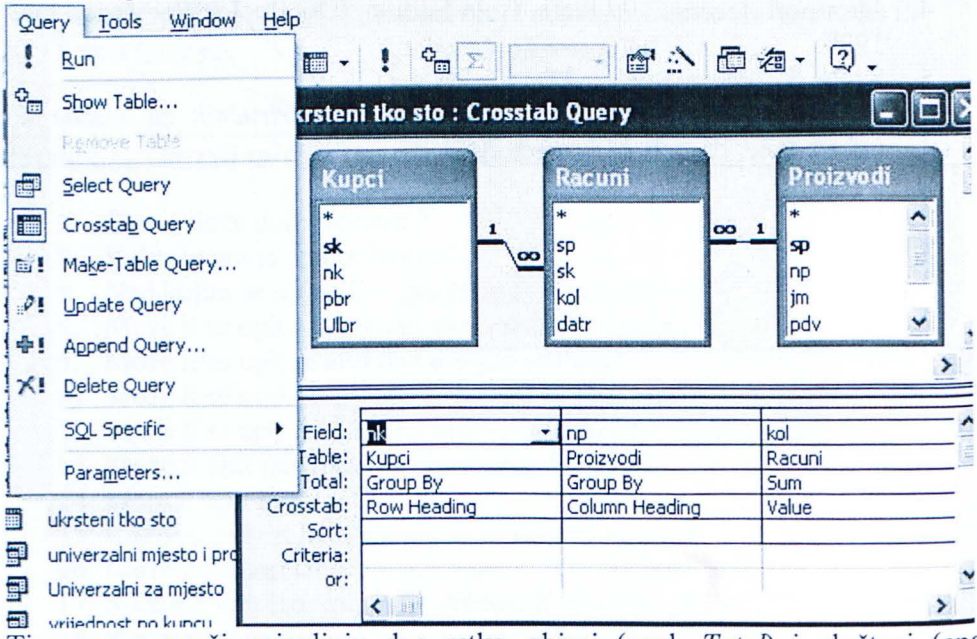
Ukršteni upiti	U ukrštenom upitu podaci su prikazani na način koji može biti povoljniji za analizu. Vrijednosti u tablici prikazane su kao zbroj, prosjek, maksimum, minimum i sl., a na što se vrijednost odnosi vidi se na početku retka i na vrhu stupca. Naslovi redaka i stupaca dobivaju se na osnovi podataka iz tablica.

ukrsteni tko sto

Naziv kupca	Kruh bijeli	Pecivo
Ada		2
Edo	1	
Pero	3	
Vera	1	

Primjer ukrštenog upita sadrži podatke koji su mogli biti prikazani i upitom izborom, ali su u ukrštenom upitu preglednije prikazani. Vidimo da se podaci o kupcu javljaju na početku retka, a podaci o proizvodu u zaglavlju stupca. Na presjeku retka i stupca javlja se vrijednost, koja se može računati na više načina, najčešće kao zbroj (engl. *Sum*).

Kako se gradi ukršteni upit? Početak gradnje je kao za upit izborom, i to onaj dio koji se odnosi na izbor tablica nad kojima se upit gradi. Zatim treba iz izbornika upita (engl. *Query*) izabrati ukršteni upit (engl. *Crosstab Query*).



Time se u mreži pojavljuju dva retka, zbirni (engl. *Total*) i ukršteni (engl. *Crosstab*). Za svaku ukrštenu tablicu barem jedan podatak mora biti definiran kao naslov retka (engl. *Row Heading*), točno jedan kao naslov stupca (engl. *Column Heading*) i točno jedan kao vrijednost (engl. *Value*). Za polje koje je vrijednosno, potrebno je izabrati i pravilo računanja u zbirnom retku (engl. *Totals*).

Ukršteni upit iz ovog primjera izražen u SQL jeziku bio bi:

criteria dosli is null otici > 0

```
TRANSFORM Sum(Racuni.kol) AS SumOfkol
SELECT Kupci.nk
FROM Proizvodi INNER JOIN (Kupci INNER JOIN Racuni ON Kupci.sk =
Racuni.sk) ON Proizvodi.sp = Racuni.sp
GROUP BY Kupci.nk
PIVOT Proizvodi.np;
```

Pitanje: Kako možemo dobiti SQL prikaz ovog upita?
Odgovor: Upit koji smo izgradili pomoću mreže za razvoj otvorimo u obliku za gradnju. Iz upitnika pogleda (engl. *View*) izaberemo SQL View.

Zadatak: Izradi upit izborom s istim podacima koji su se pojavljivali u ukrštenom i usporedi SQL prikaze ukrštenog upita i upita izborom.

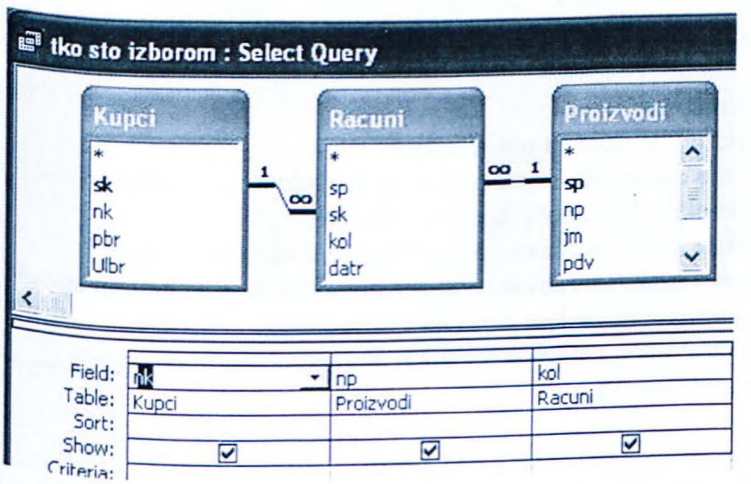
Odgovor:
Upit

tko sto izborom

Naziv kupca	Naziv proizvoda	kol
Pero	Kruh bijeli	2
Edo	Kruh bijeli	1
Ada	Pecivo	2
Pero	Kruh bijeli	1
Vera	Kruh bijeli	1

dobili smo gradnjom:
Pero je dva puta kupovao bijeli kruh, pa se u upitu izborom pojavljuje u dva retka, jednom za dva kuha, a drugi put za jedan.

U ukrštenom upitu u retku za Peru i stupcu za bijeli kruh je zbroj količina koje je Pero ukupno kupio, odnosno broj tri.



Na temelju toga dobije se SQL prikaz:
SELECT Kupci.nk, Proizvodi.np, Racuni.kol
FROM Proizvodi INNER JOIN (Kupci INNER JOIN Racuni ON Kupci.sk = Racuni.sk) ON Proizvodi.sp = Racuni.sp;
Vidimo da se taj dio koda nalazi i unutar ukrštenog upita, a u ukrštenom su dodane naredbe za zbrajanje, naslove retka i naslove stupca.

Pitanja za ponavljanje uz poglavlje 4.2

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 6–a).

Pojam	*Objašnjenje pojma*
1. Crostab Query	a) Način rada u kojem se projektira i gradi objekt. 4
2. Dynaset	b) Skup podataka prikazan u obliku privremene tablice koja nije zapisana na nosaču podataka.
3. SQL	c) Atribut ili grupa atributa koji se koriste za jednoznačnu identifikaciju sloga. 5
4. Design view	d) Primarni ključ druge tablice koji se koristi radi uspostavljanja veze. 6
5. Primarni ključ	e) Vrsta upita u kojem su neki atributi naslovi redaka, drugi naslovi stupaca, a jedan se atribut u mreži prikazuje kao podatak. 1
6. Strani ključ	f) Jezik za gradnju upita. 3
7. Zbirni redak	g) Redak u gradbenoj mreži u kojem se po nekom od ranije definiranih pravila mogu izračunavati podaci. 7

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Na koji se način u upitu povezuju primarni i strani ključ za dvije tablice?
2. Što je rezultat upita?
3. Kako se uspostavlja veza za n:n relacije?
4. Koliko atributa može sadržavati naslov retka u ukrštenom upitu?
5. Nad kojim se objektima može napraviti ukršteni upit?
6. Koliko atributa može sadržavati naslov stupca u ukrštenom upitu?
7. Koliko atributa može sadržavati vrijednosni podatak u ukrštenom upitu?
8. Kojeg tipa moraju biti podaci kojima će se računati?
9. Po kojim se pravilima najčešće računa u zbirnom retku?

Dodatna pitanja za istraživanje:

1. Nad koliko se tablica može napraviti ukršteni upit?
2. Istražite po kojim se sve pravilima može računati vrijednosno polje u ukrštenom upitu.
3. Postoji li naredba SELECT u ukrštenom upitu?
4. Postoji li naredba PIVOT u upitu izborom?
5. Čemu po vašem mišljenju služi naredba INNER JOIN?

Primjeri za vježbu:

1. Napravite upit izborom u bazi *Trgovina* koji pokazuje koliko je dobavljača iz Pule.
2. Napravite ukršteni upit za bazu *Trgovina* u kojem su nabrojeni svi proizvodi u recima, sva mjesta u stupcima, a u vrijednosti je naručena količina iz tog mjesta.
3. Nađite dobavljače koji nisu ništa isporučili.
4. Nađite kupce koji nisu ništa kupili.
5. Nađite proizvode koje nitko nije kupio.
6. Osmislite ukršteni upit za bazu koju ste vi kreirali.

Preporučena literatura za daljnje istraživanje:

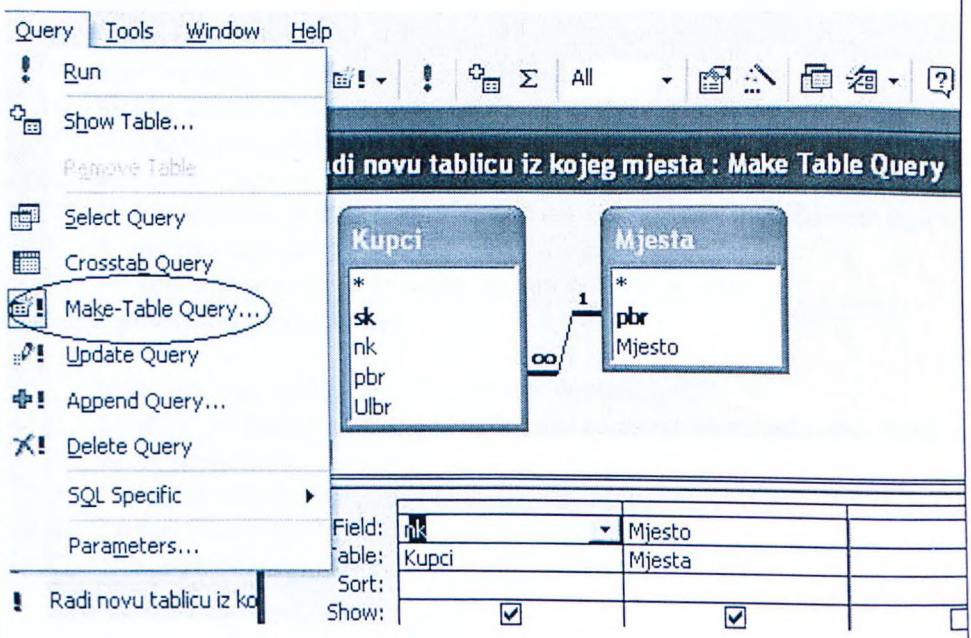
1. Čičin-Šain, M, Vukmirović, S, Čapko, Z, *Informatika za informatičko poslovanje*, Rijeka, Fortuna d.o.o, 2006.
2. Varga, M, *Baze podataka; Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb, DRIP, 1994.
3. Pomoć unutar programa Access.
4. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
5. Šavle, S, *Access*, Rijeka, Adamić, 2004.
6. Pejić-Bach, M, Spremić, M, Vlahović, N, *Priručnik za vježbe iz informatike*, Zagreb, Mikrorad, 2003.

4.3 Djelatni upiti

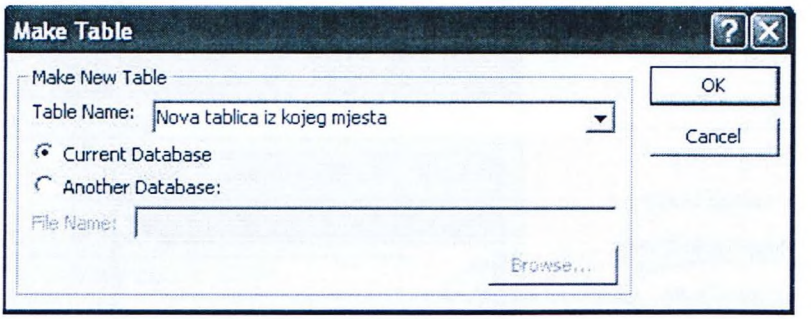
Djelatni upiti	Za razliku od upita izborom i ukrštenih upita koji služe za spretniji prikaz podataka, **djelatni upiti** služe da bi se stvorila nova tablica ili da bi se neka postojeća tablica mijenjala. Postoje četiri vrste djelatnih upita: 1. Upit koji gradi tablicu (engl. *Make-Table Query*) 2. Upit koji mijenja tablicu (engl. *Update Query*) 3. Upit koji dodaje slogove (engl. *Append Query*) 4. Upit koji briše slogove (engl. *Delete Query*)
Upit koji gradi tablicu	Rezultat upita koji gradi tablicu (engl. *Make-Table Query*), kao što mu ime kaže, je nova tablica. Ta nova tablica izgleda kao dinamički set podataka koji bi nastao da se upit gradio izborom s istim podacima.

Upit koji gradi tablicu gradi se kao upit izborom, samo se prethodno iz izbornika upita izabere upit za gradnju tablice.
Npr. upit kojim se stvara tablica slična upitu "Iz kojeg mjesta", koju smo naučili graditi u poglavlju 4.1, sastojala bi se iz koraka:

1. Početak gradnje kao upit izborom "Iz kojeg mjesta".
2. Izbor vrste upita iz izbornika.



3. Imenovanje nove tablice, npr. *Nova tablica iz kojeg mjesta* .



4. Imenovanje upita, npr. "Radi novu tablicu iz kojeg mjesta".
5. Kada se upit pozove, Access nas upozorava da će napraviti novu tablicu i nakon toga je napravi.

Djelatni upit iz ovog primjera izražen u SQL jeziku bio bi:
SELECT Kupci.nk, Mjesta.Mjesto INTO [Nova tablica iz kojeg mjesta]
FROM Mjesta INNER JOIN Kupci ON Mjesta.pbr = Kupci.pbr;

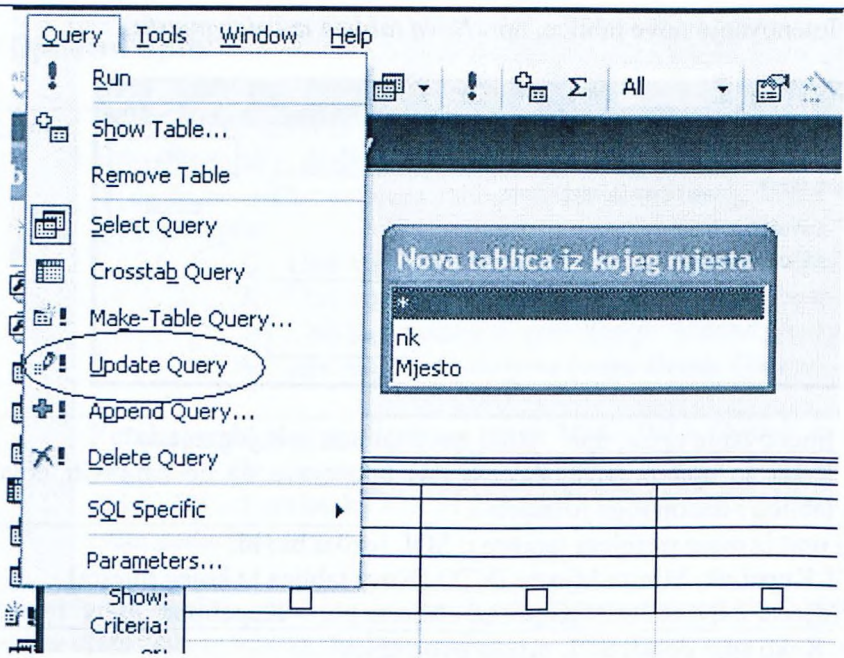
Pitanje: Kako smo dobili SQL prikaz ovog upita?

Zadatak: Izradite upit izborom s istim podacima koji su se pojavljivali u djelatnom upitu za izradu tablice i usporedite SQL prikaze djelatnog upita i upita izborom.

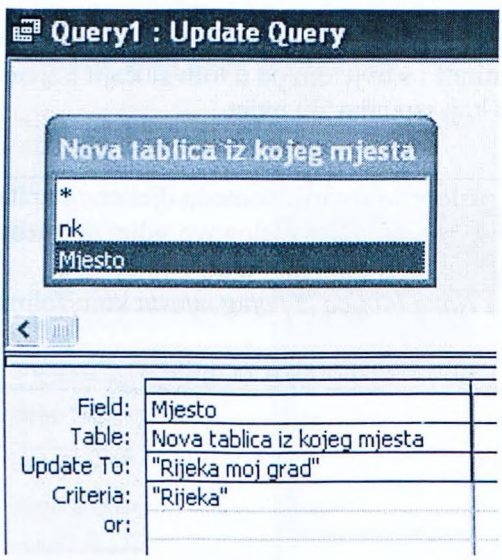
Upit koji mijenja tablicu	**Upit koji mijenja tablicu** (engl. *Update Query*) mijenja podatke u tablici, a može se kombinirati i s uvjetom pa u tom slučaju mijenja podatke samo slogovima koji ispunjavaju uvjet.

Primjerice u novoj tablici koju smo maloprije stvorili pomoću djelatnog upita, a koja se zove *Nova tablica iz kojeg mjesta*, za sve slogove gdje za mjesto piše "Rijeka" želimo podatak promijeniti u "Rijeka moj grad".

1. Za početak se izabere tablica *Nova tablica iz kojeg mjesta* koju želimo mijenjati.
2. Izabere se upit koji mijenja tablicu iz izbornika za upite.



3. U upit koji mijenja tablicu potrebno je dovući samo polje mjesto.



4. Upiše se uvjet "Rijeka".
5. U redak u koji se unosi promjena (engl. *Update To*) upiše se "Rijeka moj grad".
6. Upit se spremi pod imenom npr. "Upit koji mijenja tablicu".
7. Kada se upit pozove, tablica se promijeni uz prethodno upozorenje.

Rezultat je vidljiv u tablici *Nova tablica iz kojeg mjesta* koja sad izgleda ovako:

Nova tablica iz kojeg mjesta

nk	Mjesto
Edo	Rijeka moj grad
Kiki	Rijeka moj grad
Ada	Pula
Pero	Zagreb
Vera	Zagreb

SQL oblik ovog upita je:
UPDATE [Nova tablica iz kojeg mjesta] SET [Nova tablica iz kojeg mjesta].Mjesto = "Rijeka moj grad"
WHERE ((([Nova tablica iz kojeg mjesta].Mjesto)="Rijeka"));

Pitanje: Koja se nova SQL naredba pojavljuje u ovom upitu?

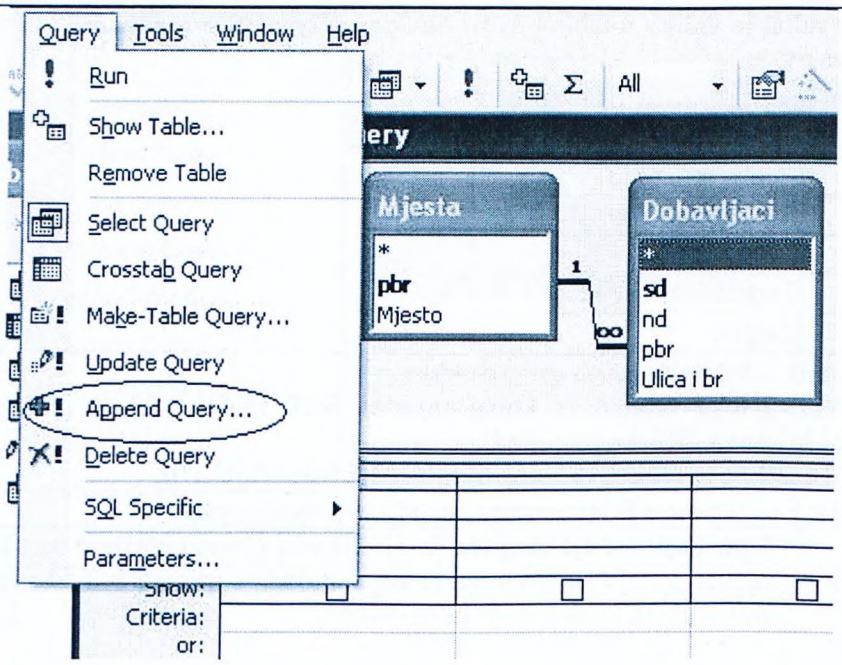
Upit koji dodaje slogove	**Upit koji dodaje slogove** (engl. *Append Query*) takav je upit koji tablici dodaje slogove. Polja koja se dodaju moraju biti iste vrste kao polja iz tablice na koju se dodaju.

Taj je upit koristan kada npr. želimo objediniti dvije tablice u jednu.

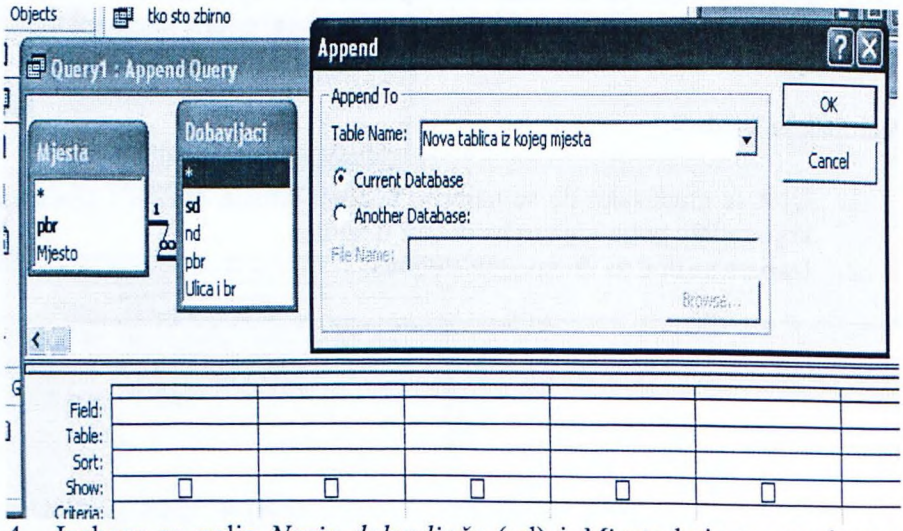
Za ovaj ćemo upit izraditi primjer tako da ćemo na tablicu *Nova tablica iz kojeg mjesta*, u kojoj su do sada bili upisani svi kupci iz baze s nazivom mjesta iz kojeg dolaze, dodati još i sve dobavljače, također s nazivom mjesta.

Postupak je sljedeći:

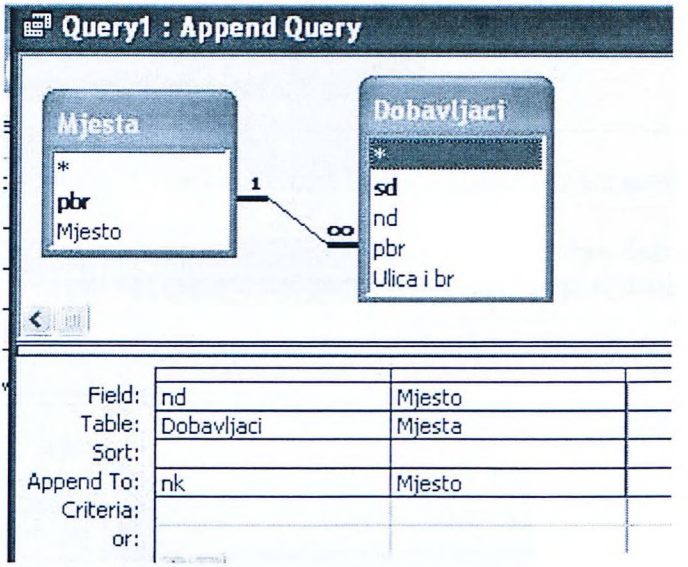
1. Upit se gradi tako da se najprije pozovu tablice *Mjesta* i *Dobavljači* koje sadrže polja koja treba dodati u tablicu.
2. Izabere se upit za dodavanje slogova.



3. Imenuje se tablica u koju se slogovi dodaju, u našem slučaju *Nova tablica iz kojeg mjesta*.



4. Izaberu se polja *Naziv dobavljača* (nd) i *Mjesto* koje se namjerava dodati. Kako se *Naziv dobavljača* pojavljuje u polju gdje je bio *Naziv kupca*, potrebno je unijeti i ime polja koje prima dodatni podatak u retku za dodavanje (engl. *Append To*). Kod polja *Mjesto* to nije potrebno jer se polje već zvalo *Mjesto*.



Rezultat je:

Nova tablica iz kojeg mjesta

nk	Mjesto
Edo	Rijeka moj grad
Kiki	Rijeka moj grad
Ada	Pula
Pero	Zagreb
Vera	Zagreb
Bionjam	Rijeka
Ekonjup	Pula
Kravica	Pula
Agroprom	Zagreb
Finesa	Zagreb

5. Po potrebi bi se moglo sortirati podatke ili dodati uvjet. Kako to nije bilo zadano, potrebno je samo imenovati i spremiti uvjet.
6. Kada se upit pozove, on će na tablicu dodati slogove, u našem slučaju pet slogova jer postoji pet dobavljača.

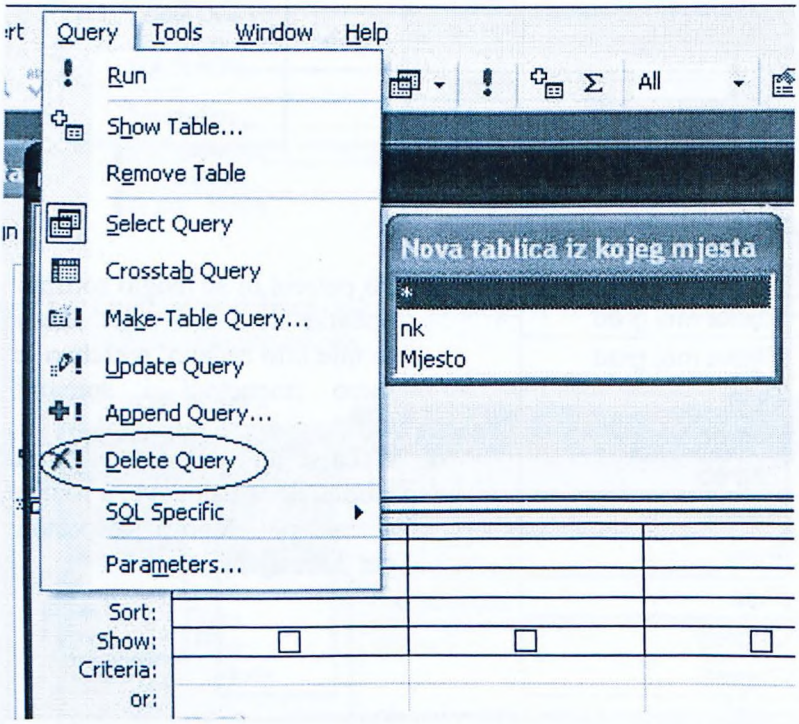
SQL oblik ovog upita je:
INSERT INTO [Nova tablica iz kojeg mjesta] (nk, Mjesto)
SELECT Dobavljaci.nd, Mjesta.Mjesto
FROM Mjesta INNER JOIN Dobavljaci ON Mjesta.pbr = Dobavljaci.pbr;

Pitanja:
Koja se nova SQL naredba pojavljuje u ovom upitu?
Zašto ovdje nema naredbe WHERE?
Koja naredba osigurava vezu između tablica *Mjesta* i *Dobavljači*?

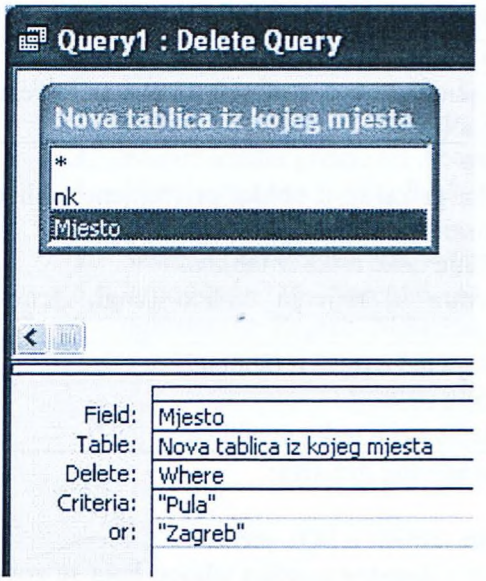
Upit koji briše slogove	**Upit koji briše slogove** (engl. *Delete Query*) iz postojeće tablice briše one slogove koji zadovoljavaju uvjet.

Primjer: Iz tablice *Nova tablica iz kojeg mjesta* želimo maknuti sve upise koji su iz Zagreba i Pule.
Upit se gradi na sljedeći način:

1. Pozove se gradnja upita i tablica *Nova tablica iz kojeg mjesta*.
2. Izabere se upit koji briše slogove.



3. Postavi se uvjet za brisanje slogova.



4. Upit se imenuje i spremi.
5. Kada se upit pozove, nakon upozorenja briše slogove iz tablice.

Rezultat rada tog upita je:

Nova tablica iz kojeg mjesta

nk	Mjesto
Edo	Rijeka moj grad
Kiki	Rijeka moj grad
Bionjam	Rijeka

SQL oblik ovog upita je:
DELETE [Nova tablica iz kojeg mjesta].Mjesto
FROM [Nova tablica iz kojeg mjesta]
WHERE ((([Nova tablica iz kojeg mjesta].Mjesto)="Pula" Or ([Nova tablica iz kojeg mjesta].Mjesto)="Zagreb"));

Pitanja:
Koja se nova SQL naredba pojavljuje u ovom upitu?
Koja je funkcija riječi *Or* u funkciji WHERE?
Koliko bi se slogova obrisalo da je umjesto *Or* pisalo *And*? Obrazložite.

Pitanja za ponavljanje uz poglavlje 4.3

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–f).

Pojam	*Objašnjenje pojma*
1. Djelatni upit	a. Skup podataka prikazan u obliku privremene tablice koja nije zapisana na nosaču podataka. 2
2. Dynaset	b. Upit koji uklanja neke retke iz tablica. 6
3. SQL	c. Upit koji stvara ili mijenja tablicu (engl. *Action Query*). 1
4. Update Query	d. Upit koji dodaje neke retke u tablicu. 5
5. Append Query	e. Jezik za gradnju upita. 3
6. Delete Query	f. Upit koji uklanja neke retke iz tablica. 4

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Može li se upit koji briše slogove izraziti u SQL jeziku?
2. Želite spojiti tablicu *Dobavljača* i *Kupaca* u jednu tablicu koja se zove *Spojena*. Kako biste to mogli učiniti?
3. Može li rezultat upita biti nova tablica?
4. Koja je razlika između upita izborom i upita koji gradi tablicu?
5. Koliko ima raznih djelatnih upita?

Dodatna pitanja za istraživanje:

1. Želite u tablici *Proizvodi* sve prodajne cijene uvećati za 10%. Možete li to napraviti pomoću upita? Pomoću kojeg i kako?
2. Postoji li naredba SELECT u upitu koji briše slogove?
3. Pokrenite dva puta uzastopce upit koji gradi tablicu. Što se dogodilo?
4. U Access pomoći pročitajte članke o djelatnim upitima.

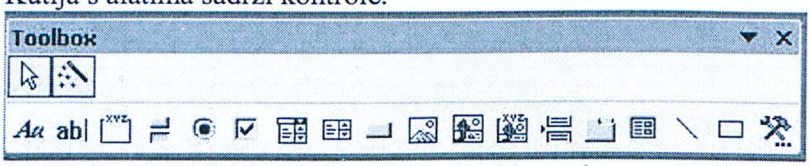
Primjeri za vježbu:

1. Izradite sve djelatne upite iz ovog poglavlja i pokrenite ih.
2. Osmislite djelatne upite za bazu koju ste vi kreirali.

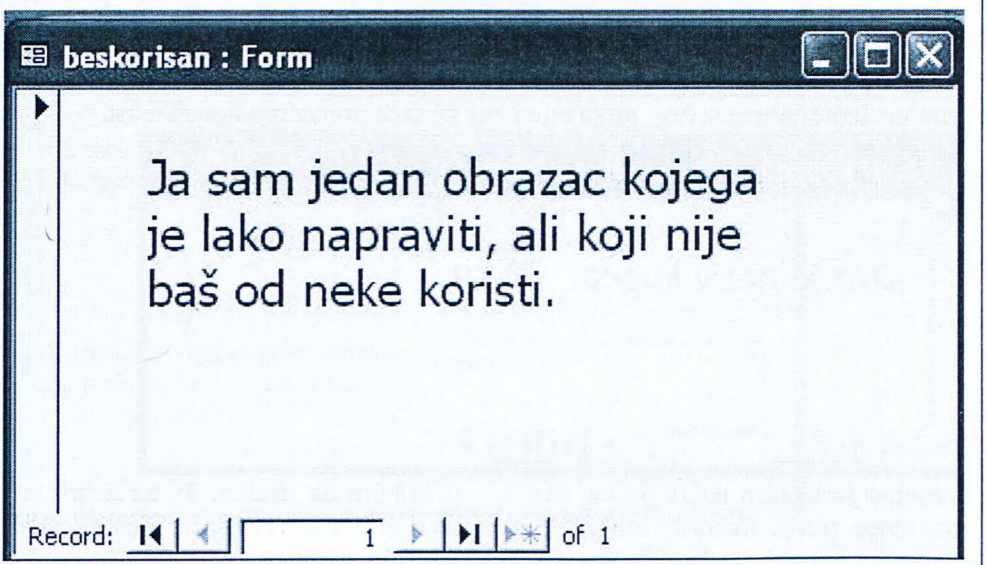
Preporučena literatura za daljnje istraživanje:

1. Čičin-Šain, M, Vukmirović, S, Čapko, Z, *Informatika za informatičko poslovanje*, Rijeka, Fortuna d.o.o, 2006.
2. Varga, M, *Baze podataka; Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb, DRIP, 1994.
3. Pomoć unutar programa Access.
4. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
5. Šavle, S, *Access*, Rijeka, Adamić, 2004.
6. Pejić-Bach, M, Spremić, M, Vlahović, N, *Priručnik za vježbe iz informatike,* Zagreb, Mikrorad, 2003.

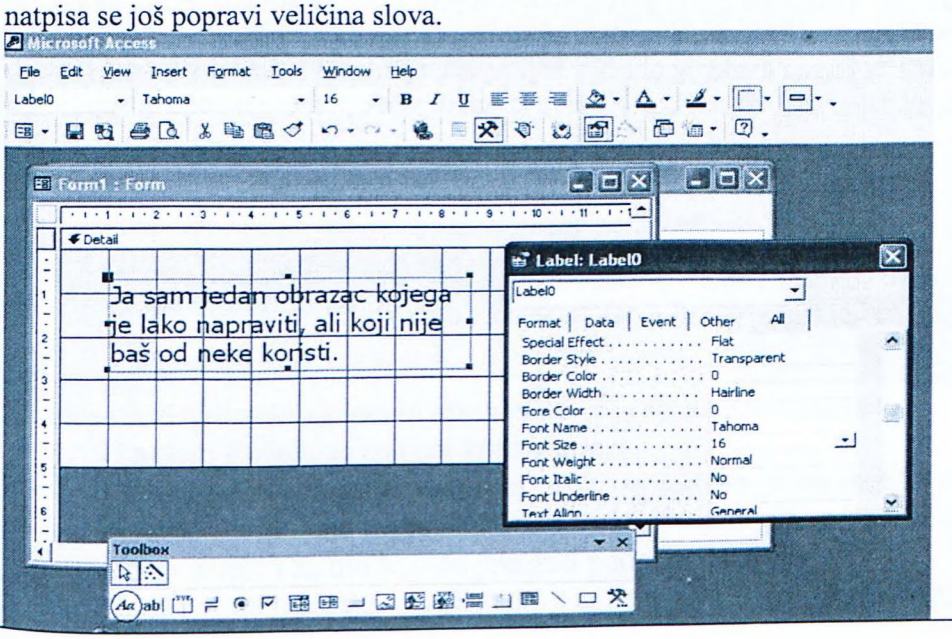
5. Obrasci

Obrasci Aa	**Obrasci** (engl. *Forms*) su objekti baze koji se najčešće koriste za izgradnju korisničkog sučelja na ekranu za rad s bazom podataka. Obrasci su korisni prilikom unosa podataka jer mogu nuditi dodatne informacije o tome koji se podatak treba unijeti, a mogu sadržavati i objekte poput gumba, izbornika i sl. koji olakšavaju unos i kontrolu unesenih podataka. Objekti koji se dodaju na obrazac nalaze se u kutiji s alatima (engl. *Toolbox*) i zovu se kontrole (engl. *Controls*).
Kontrole	Kutija s alatima sadrži kontrole.  Kontrole su objekti koji se mogu smjestiti na obrazac, izvješće ili na internetsku stranicu. Najčešće korištene kontrole su: • kontrola za natpise (engl. *Label*) • kontrola za prihvat podataka (engl. *Textbox*) • kontrolni gumb (engl. *Control Button*) pa će u ovome priručniku biti opisan rad s njima. Kontrole imaju svoju ikonu pomoću koje se pozivaju, svoj grafički prikaz na objektu u kojem se koriste i svoja svojstva koja korisnik može po potrebi podešavati.
Ostale kontrole	Postoje još neke kontrole koje omogućavaju korištenje nekih korisnih objekata poput kombiniranog okvira (engl. *Combo Box*), okvira za izbor (engl. *Check Box*), slike (engl. *Picture*) i još nekoliko raznih gumbova, npr. prekidač (engl. *Toggle Button*), izborni gumb (engl. *Option Button*) i drugi. Sve one mogu biti korisne u raznim situacijama pa se čitateljima i čitateljicama preporuča da istraže njihove mogućnosti.
Natpisi	Kontrola za natpise omogućava unos teksta na obrazac. Ikona natpisa je Aa, a svojstva natpisa su podaci o samom natpisu, kao npr. ime natpisa, sadržaj natpisa, vrsta i veličina slova i još dvadesetak raznih podataka koja spadaju u svojstva natpisa.

Primjer izrade prvog obrasca:
Za početak će se izraditi obrazac koji nije povezan ni sa jednom tablicom, poput ovoga koje se zove "beskoristan".

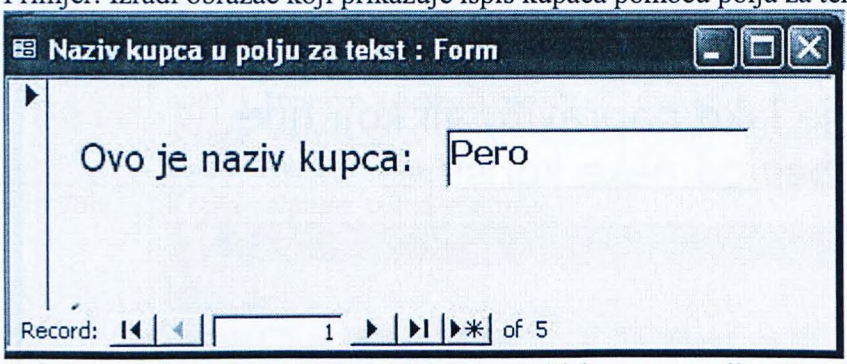


To se radi na sljedeći način. Iz okvira za obrasce (engl. *Forms*) se izabere opcija za izradu novog obrasca (engl. *New*). Zatim se iz kutije s alatima izabere ikona za natpise. Natpis se smjesti na obrazac. U njega se upiše tekst, a u svojstvima natpisa se još popravi veličina slova.



Polja za tekst ab|	**Polja za tekst** (engl. *Text Box*), čija je ikona u kutiji s alatima ab |, služe da se obrazac poveže s izvorom podataka koji se nalazi u tblicama. Polja za tekst također imaju mnogo svojstava pomoću kojih se definira odakle se podaci pribavljaju i kakav je njihov prikaz na obrascu.

Primjer: Izradi obrazac koji prikazuje ispis kupaca pomoću polja za tekst.



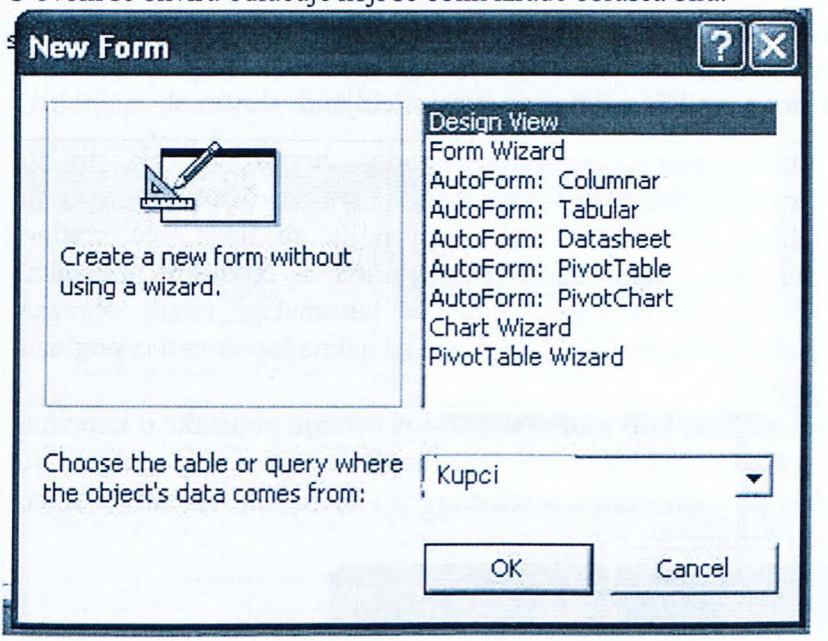
Trenutno je ispisan naziv prvog kupca, a pritiskom na strelicu ► uz natpis za broj sloga (engl. *Record*) mogli bi se ispisati i redom svi ostali. Postoji više načina za izradu ovakvog obrasca, koji ispisuje podatke o nazivu kupca iz tablice Kupci. Izabrat ćemo način kod kojeg korisnik može podesiti mnogo detalja.

Primjer: Izradi obrazac koji prikazuje ispis kupaca pomoću polja za tekst.

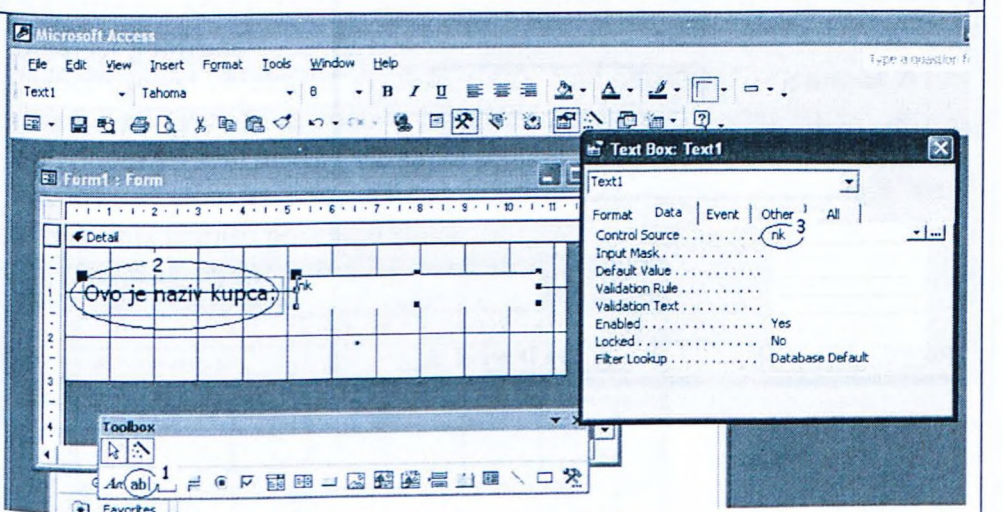
Trenutno je ispisan naziv prvog kupca, a pritiskom na strelicu ► uz natpis za broj sloga (engl. *Record*) mogli bi se ispisati i redom svi ostali. Postoji više načina za izradu ovakvog obrasca koji ispisuje podatke o nazivu kupca iz tablice *Kupci*. Izabrat ćemo način kod kojeg korisnik može podesiti mnogo detalja.

Najviše se detalja može podesiti ako se radi u pogledu gradnje (engl. *Design View*). Izbor će se obaviti u okviru za novi obrazac.

U ovom se okviru odlučuje koji se oblik izrade obrasca bira.



Najprije treba izabrati način rada, tj. pogled gradnje, a zatim i tablicu (ili upit) na temelju koje se obrazac gradi. U ovom slučaju to je tablica *Kupci*.

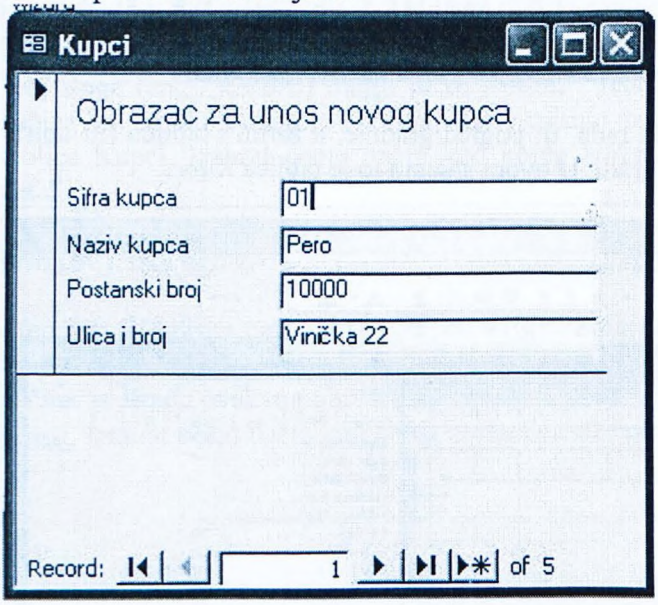


Zatim se koristi mreža za izradu obrasca na koju se nanose kontrole.

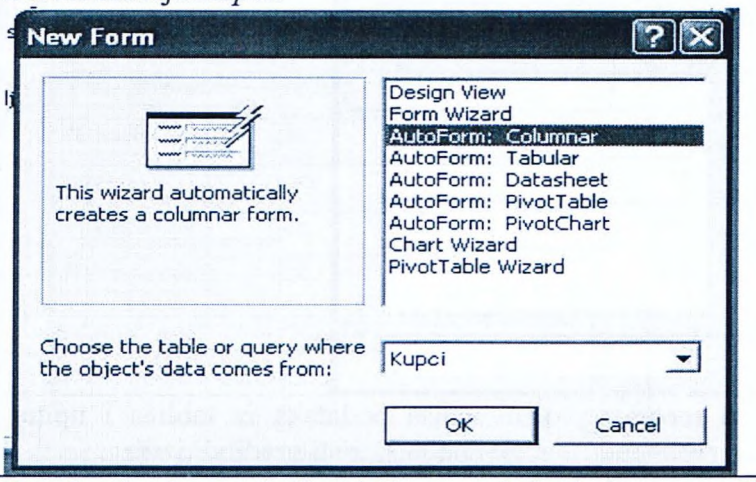
1. U našem slučaju se izabere kontrola ab|.
2. Kontrola ab| za unos polja za tekst. smještena je na mrežu za obrazac i podešena joj je veličina za oba njena dijela, odnosno za dio za natpis i za dio koji će preuzeti podatke iz tablice.
3. U kartici za podatke (engl. *Data*) za svojstva (engl. Properties)

	polja za tekst unese se u svojstvo izvora podatka (engl. *Control* Source) atribut nk, odnosno naziv polja od kuda se preuzimaju podaci. U kartici oblikovanja (Engl. *Format*) može se podesiti vrsta slova, veličina slova. opcija podebljanih slova i sl.
Izrada obrazaca pomoću čarobnja-ka.	Veliku pomoć prilikom izrade obrazaca mogu pružiti čarobnjaci. U tom slučaju nudi se izrada obrasca na temelju tablice ili upita u kojem su polja za tekst već uredno poravnata na obrascu i povezana s izvorom podataka. Korisna je tehnika kojom se automatski izradi obrazac pomoću čarobnjaka, a obrazac se naknadno doradi u pogledu gradnje.

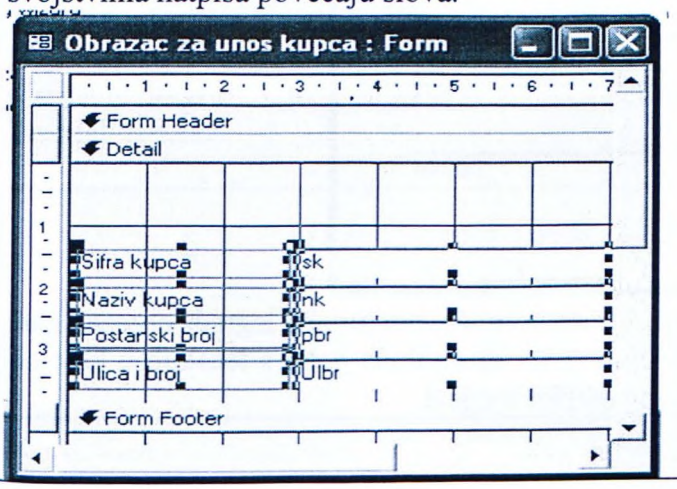
Primjer: Izradi obrazac koji može pomoći pri listanju podataka o kupcima i unosu novog kupca. Taj bismo obrazac mogli izraditi u pogledu gradnje dodajući jedno po jedno polje s tekstom, ali to bi bilo napornije nego izrada pomoću čarobnjaka.

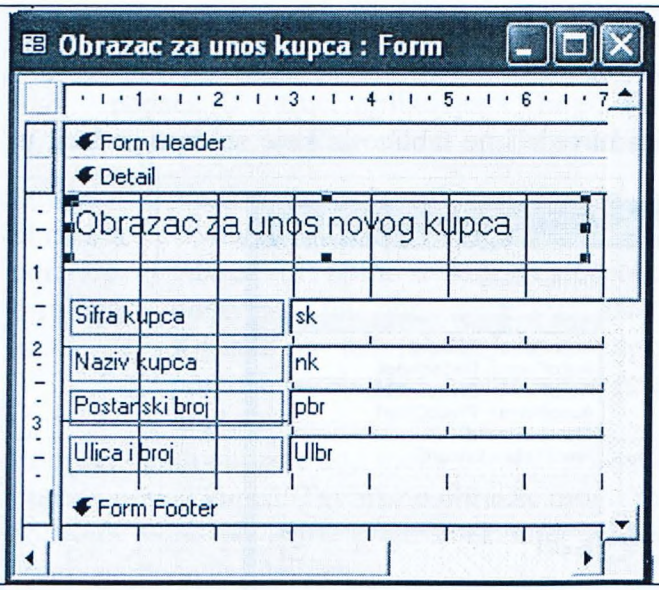


Koristeći čarobnjaka postupak je sljedeći:
Izabere se jedan od čarobnjaka za auto-matsku izradu obras-ca, npr. stupčasti (engl. *Columnar*).
U donjem se dijelu okvira navede ime tablice iz koje se crpe podaci, u našem slučaju *Kupci*.



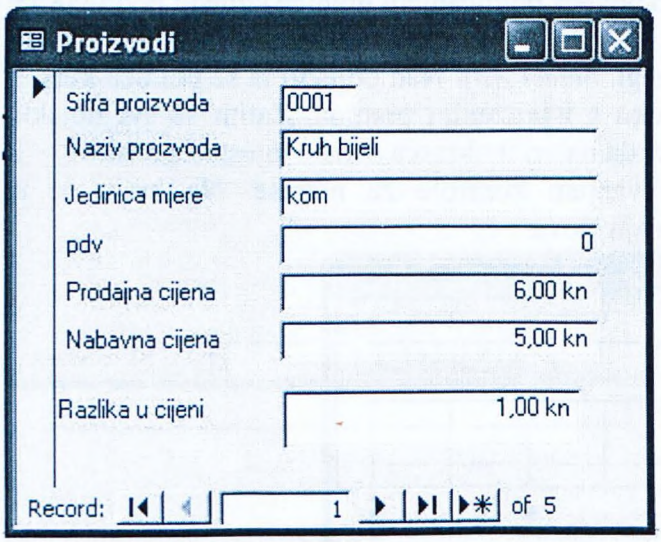
Rezultat je obrazac koji sadrži sve podatke koje smo tražili, jedino na vrh treba dodati naslov.
Za dodavanje je naslova potrebno obrazac doraditi u pogledu za gradnju:
Najprije se obrazac malo poveća pomicanjem granice između područja za detalje i podliska.
Izaberu se svi objekti (engl. *Select All*). Nad objektima se pomiče kursor dok se ne pretvori u ruku s ispruženim prstima. Zatim se svi objekti odjednom presele naniže, da na vrhu obrasca ostane mjesta za naslov.
Naslov se dodaje dodavanjem kontrole za natpise. Na kraju se u svojstvima natpisa povećaju slova.



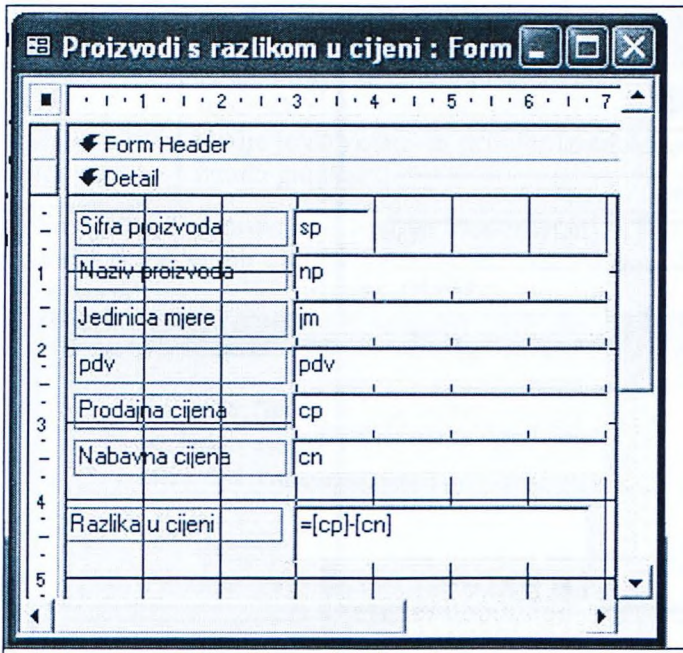


U polju s tekstom mogu se, osim ispisa podataka iz tablica i upita, ispisivati i rezultati računanja. Za izračun se koristi graditelj izraza.

Primjer:
Napravite obrazac koji za proizvode na ekranu, osim nabavne i prodajne cijene, ispisuje i njihovu razliku.



Taj se obrazac gradi tako da se pomoću čarobnjaka izgradi obrazac nad tablicom *Proizvodi*, a zatim se na njega doda polje s tekstom u koje se upiše formula za računanje razlike u cijeni.



Formula glasi: =[cp]-[cn]. Za upis formule može se koristiti graditelj izraza. Varijable cp i cn obavezno se upisuju u uglate zagrade, a formula mora započeti znakom jednakosti.

Nešto je složeniji zadatak ako se na obrascu želi prikazati ispis vrijednosti poreza na dodanu vrijednost. U tom se slučaju porez računa ovisno o parametru pdv. Ako je sadržaj parametra "Yes", porez se obračunava u iznosu od 22%, a ako je "No", porez se ne obračunava, odnosno iznosi 0 kuna.

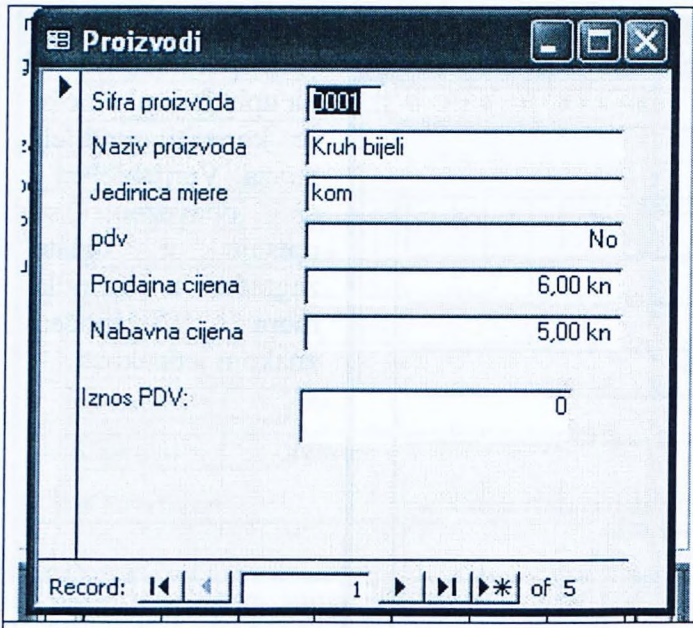
Slično kao u Excelu, za takav se izračun koristi formula odluke koja u Accessu izgleda ovako:

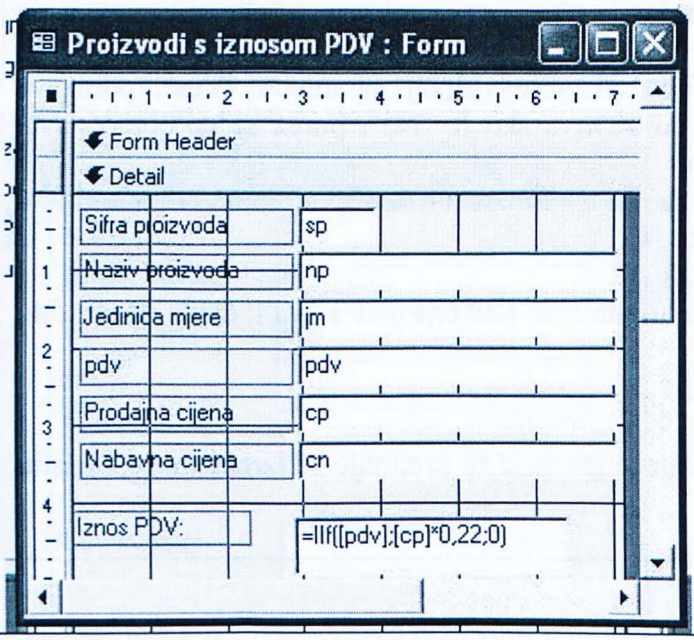
=IIf(uvjet;izraz1;izraz2)

Izraz je isti kao u Excelu, samo se funkcija piše s dva i, dakle Iif umjesto If.

=IIf([pdv];[cp]*0,22;0)

Varijable dolaze u uglate zagrade, a za kreiranje se izraza može koristiti graditelj izraza.

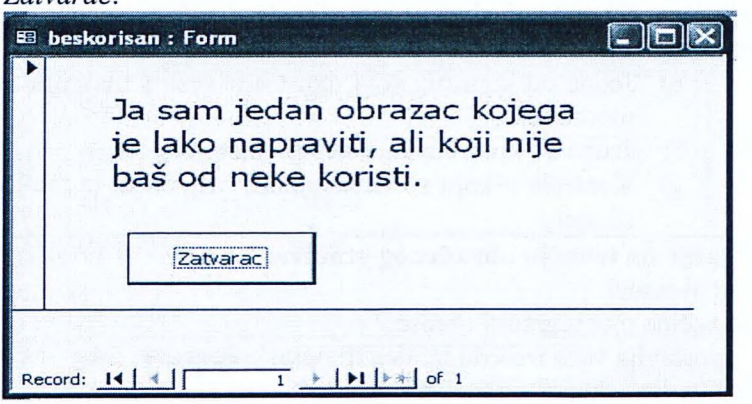




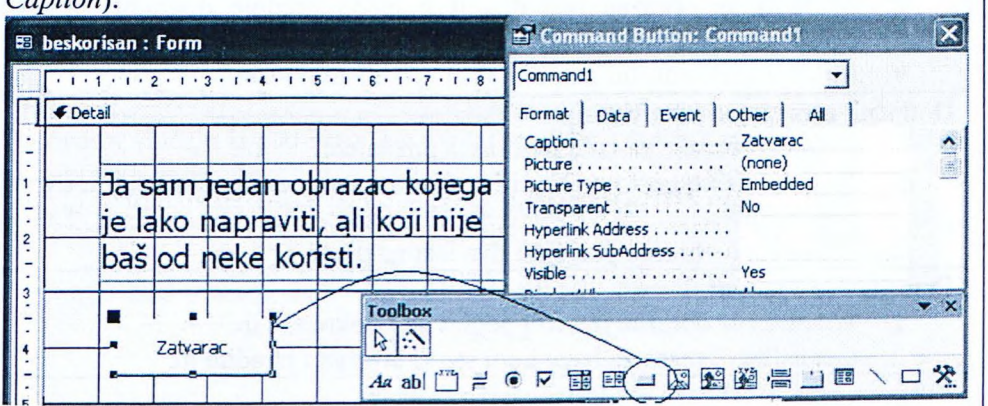
Iznos PDV-a uvjetovan je logičkom varijablom.

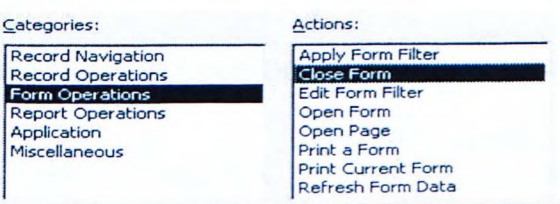
Gumb	**Gumb za naredbe** (engl. *Command Button*) je kontrola koja se upotrebljava za automatizaciju poslova, a imat će više smisla kada se nauče pisati makronaredbe ili programski moduli u Accessu. Ovdje je dan osnovni primjer izrade gumba pomoću čarobnjaka za izradu gumba.

Na postojeći obrazac koji se zove "beskoristan", a koji je sadržavao samo natpis, dodat ćemo gumb čija je svrha da zatvori obrazac. Taj će gumb nositi natpis *Zatvarač*.



Taj se zadatak rješava tako da se najprije obrazac otvori u pogledu za gradnju, iz kutije s alatima (engl. *Toolbox*) se na obrazac dovuče i smjesti gumb, pomoću čarobnjaka se odabere funkcija gumba (zatvaranje obrasca), a na kraju se u svojstvima makne slika s gumba i doda natpis u svojstvo natpisa (engl. *Caption*).





Pitanja za ponavljanje uz poglavlje 5

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–a).

Pojam	*Objašnjenje pojma*
1. Obrazac ×	a) Objekt koji se može dodati na obrazac ili izvješće. 1
2. Kutija s alatima ×	b) Engleski naziv za kutiju s alatima. 7
3. Kontrola	c) Objekt koji služi za izgradnju korisničkog sučelja. 3
4. Kontrolni gumb ×	d) Traka s alatima koja sadrži kontrole. 2
5. Natpis	e) Jedna od kontrola koja može biti veza s makroima i modulima. 4
6. Polja za tekst ×	f) Jedna od kontrola koja sadrži opise. 6
7. Control Box ×	g) Kontrola u koju se može upisati vrijednost iz tablice ili upita. 5

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Čemu služe obrasci?
2. Na koje se načine mogu graditi obrasci?
3. Kako se uspostavlja veza između tablice ili upita i obrasca?
4. Koja je razlika između polja za tekst i natpisa?
5. Koji su engleski nazivi za pojmove obrasci, kutija s alatima, kontrola, gumb, natpis i polje za tekst?
6. Kada se obrazac gradi u pogledu gradnje?
7. Može li se obrazac izgrađen pomoću čarobnjaka dograditi u pogledu gradnje?
8. Može li se obrazac izgrađen u pogledu gradnje dograditi pomoću čarobnjaka za izradu obrazaca?
9. Kako se na kontrolni gumb može napisati neki tekst?

Dodatna pitanja za istraživanje:

1. Koje čarobnjake za izradu obrasca poznajete?
2. Koje kontrole postoje u kutiji s alatima, osim onih obrađenih u ovome poglavlju?
3. Mogu li na obrascu postojati dva ista natpisa?

Primjeri za vježbu:

1. Izradite sve obrasce iz ovog poglavlja i pokrenite ih.
2. Osmislite obrasce za bazu koju ste vi kreirali i izradite ih.

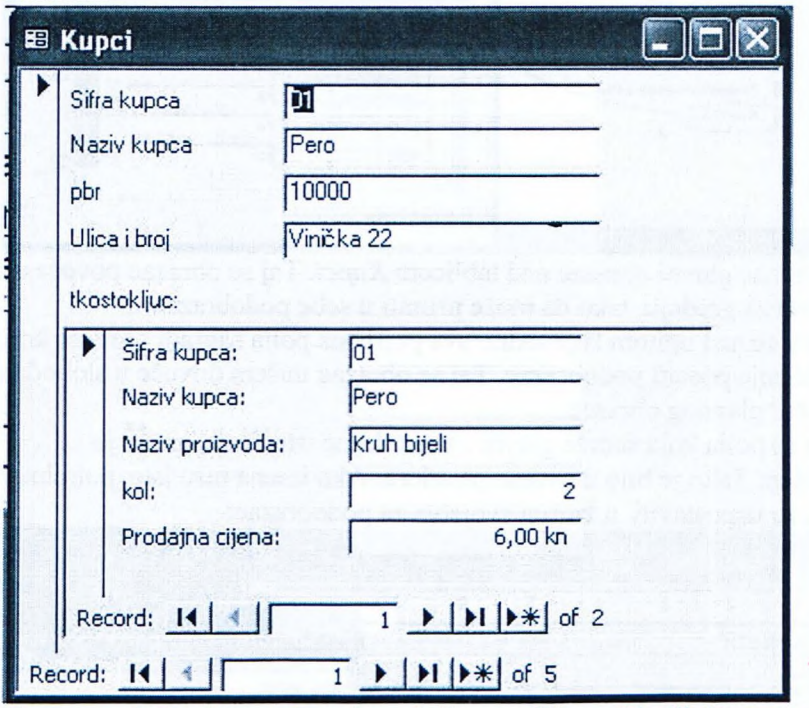
Preporučena literatura za daljnje istraživanje:

1. Pomoć unutar programa Access.
2. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
3. Šavle, S, *Access*, Rijeka, Adamić, 2004.
4. Pejić-Bach, M, Spremić, M, Vlahović, N, *Priručnik za vježbe iz informatike*, Zagreb, Mikrorad, 2003.

5.1 Podobrasci (engl. Subforms)

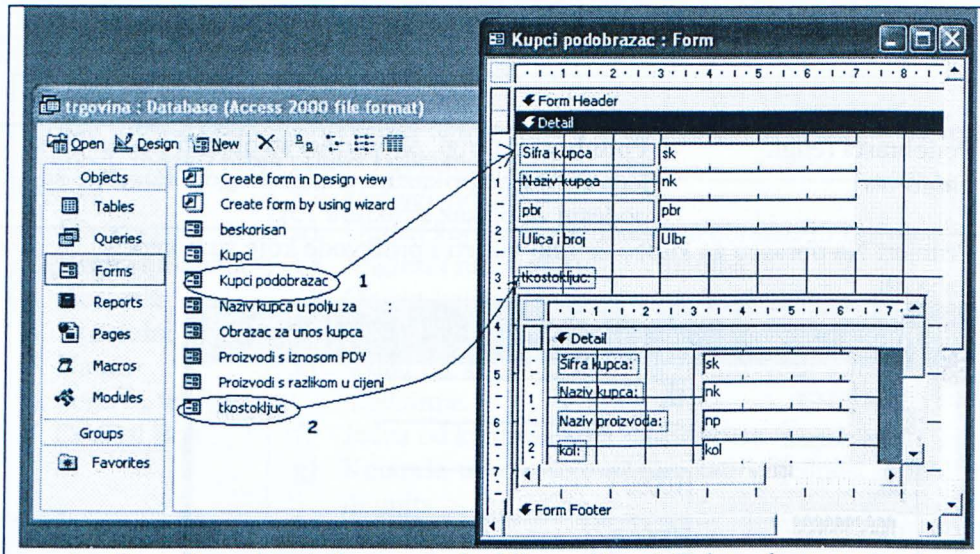
Podobrasci (engl. *Subforms*)	**Podobrasci** (engl. *Subforms*) su objekti kod kojih se jedan obrazac prikazuje unutar drugog. Posebno su podesni za prikaz struktura 1:n.

Primjer: Na obrascu za kupce želimo vidjeti i proizvode koje su ti kupci nabavili.

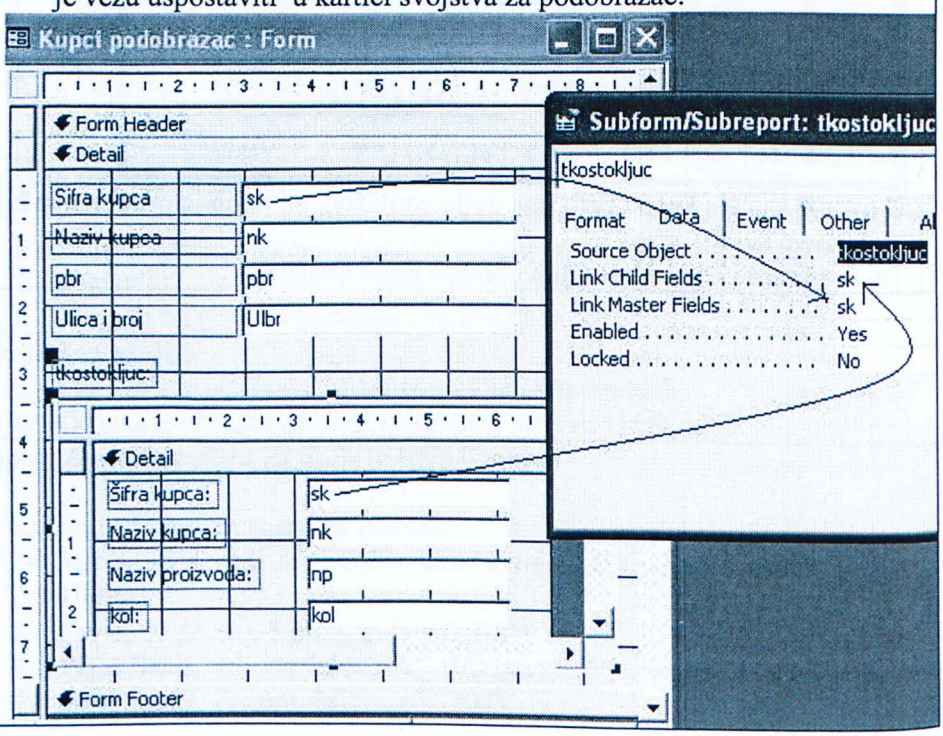


U ovome slučaju kupac Pero ima dvije narudžbe, od kojih je prva za dva bijela kruha po 6 kuna.

Taj se obrazac koji sadrži drugi obrazac gradi ovako:



1. Izgradi se glavni obrazac nad tablicom *Kupci*. Taj se obrazac poveća u načinu za gradnju, tako da može primiti u sebe podobrazac.
2. Zatim se nad upitom koji sadrži sva potrebna polja sagradi obrazac koji će kasnije postati podobrazac. Taj se obrazac mišem dovuče u slobodan prostor glavnog obrasca.
3. Ako se polja koja sadrže glavni i strani ključ isto zovu, posao je završen. Tako je bilo u ovome primjeru. Ako imena nisu ista, potrebno je vezu uspostaviti u kartici svojstva za podobrazac:



Polje se glavnog obrasca i podobrasca povezuje pomoću parametara *Polje gospodara* (engl. *Master Field*) i *Polje dijete* (engl. *Child Field*).

Pitanja za ponavljanje uz poglavlje 5.1

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–a).

Pojam	*Objašnjenje pojma*
1. Podobrazac	a) Polje koje sadrži poveznicu u podobrascu.
2. Polje gospodar	b) Engleski naziv za podobrazac.
3. Polje dijete	c) Polje koje sadrži poveznicu u glavnom obrascu.
4. Subform	d) Obrazac unutar obrasca.

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Čemu služe podobrasci?
2. Na koji se način grade obrasci koji sadrže podobrasce?
3. Kako se uspostavlja veza između glavnog obrasca i podobrasca?
4. Kako se zovu polja pomoću kojih se uspostavlja veza između obrasca i podobrasca?
5. Kada se veza uspostavlja automatski?
6. Gdje se nalazi podatak za polje gospodar?
7. Gdje se nalazi podatak za polje dijete?

Dodatna pitanja za istraživanje:

1. Mogu li se pri izradi podobrasca koristiti čarobnjaci?
2. Može li obrazac imati više podobrazaca?
3. Koji su engleski nazivi za podobrazac, poveznicu, polje gospodar i polje dijete?

Primjeri za vježbu:

1. Izradite obrazac s podobrascem iz ovog poglavlja i pokrenite ga.
2. Osmislite obrazac s podobrascem za bazu koju ste vi kreirali.

Preporučena literatura za daljnje istraživanje:

1. Pomoć unutar programa Access.
2. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
3. Šavle, S, *Access*, Rijeka, Adamić, 2004.
4. Pejić-Bach, M, Spremić, M, Vlahović, N, *Priručnik za vježbe iz informatike*, Zagreb, Mikrorad, 2003.

6. Izvješća (engl. Reports)

Izvješća (engl. *Reports*)	**Izvješća** (engl. *Reports*) su objekti koji prikazuju rezultate pretrage ili obrade baze podataka. Posebno su podesni za pripremu podataka za ispis na pisaču.

Primjer: Želimo dobiti izvještaj o svim narudžbama uključujući vrijednost narudžbi.

Ukupna vrijenost narudžbi

Ukupna vrijenost narudžbi

sp	*Naziv proizvoda*	*Jedinica mjere*	*Nabavna cijena*	*Naručena količina*	*Vrijednost*
0001					
	Kruh bijeli	kom	5,00 kn	50	250,00 kn
	Kruh bijeli	kom	5,00 kn	200	1.000,00 kn
Summary for 'sp' = 0001 (2 detail records)					
Sum					*1.250,00 kn*
0002					
	Pecivo	kom	0,80 kn	100	80,00 kn
Summary for 'sp' = 0002 (1 detail record)					
Sum					*80,00 kn*
0003					
	Keksi	kg	20,00 kn	32	640,00 kn
	Keksi	kg	20,00 kn	150	3.000,00 kn
Summary for 'sp' = 0003 (2 detail records)					
Sum					*3.640,00 kn*
0004					
	Mlijeko	l	4,00 kn	24	96,00 kn
Summary for 'sp' = 0004 (1 detail record)					
Sum					*96,00 kn*
Grand Total					*5.066,00 kn*

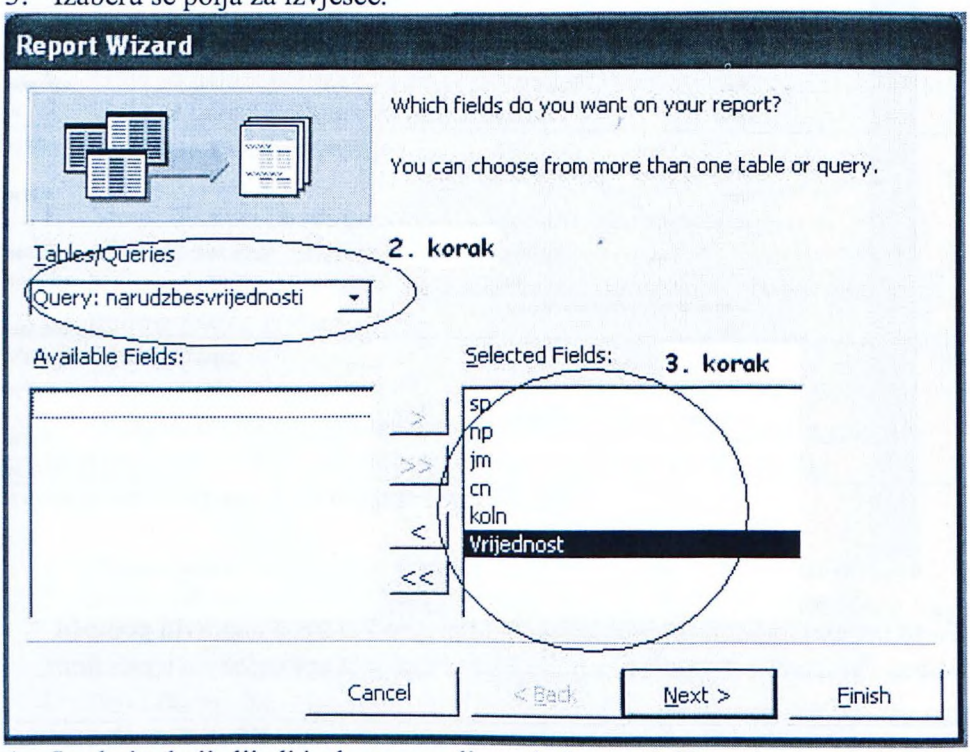
Page: 1

To se izvješće gradi na osnovi upita. Upit prethodno treba napraviti pomoću tablica *Proizvodi* i *Narudžbe*, a polje *Vrijednost* se u upitu dobiva izračunom.

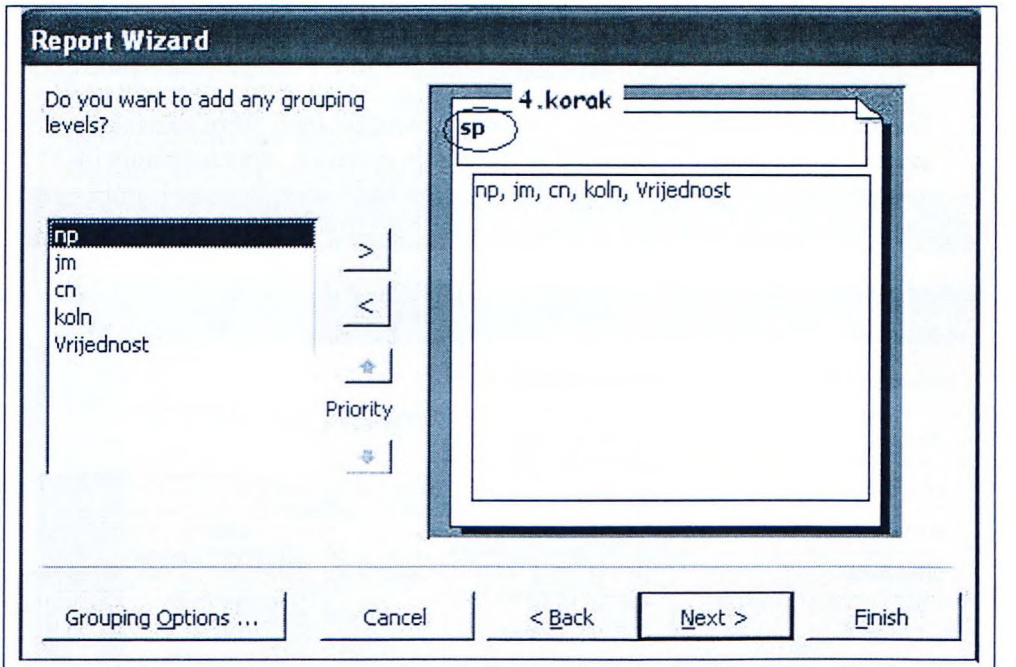
Šifra proizvoda	Naziv proizvoda	Jedinica mjere	Nabavna cijena	Naručena količina	Vrijednost
0001	Kruh bijeli	kom	5,00 kn	200	1.000,00 kn
0001	Kruh bijeli	kom	5,00 kn	50	250,00 kn
0002	Pecivo	kom	0,80 kn	100	80,00 kn
0003	Keksi	kg	20,00 kn	150	3.000,00 kn
0003	Keksi	kg	20,00 kn	32	640,00 kn
0004	Mlijeko	l	4,00 kn	24	96,00 kn

Za izradu izvješća korišten je čarobnjak. Radilo se u 11 koraka:

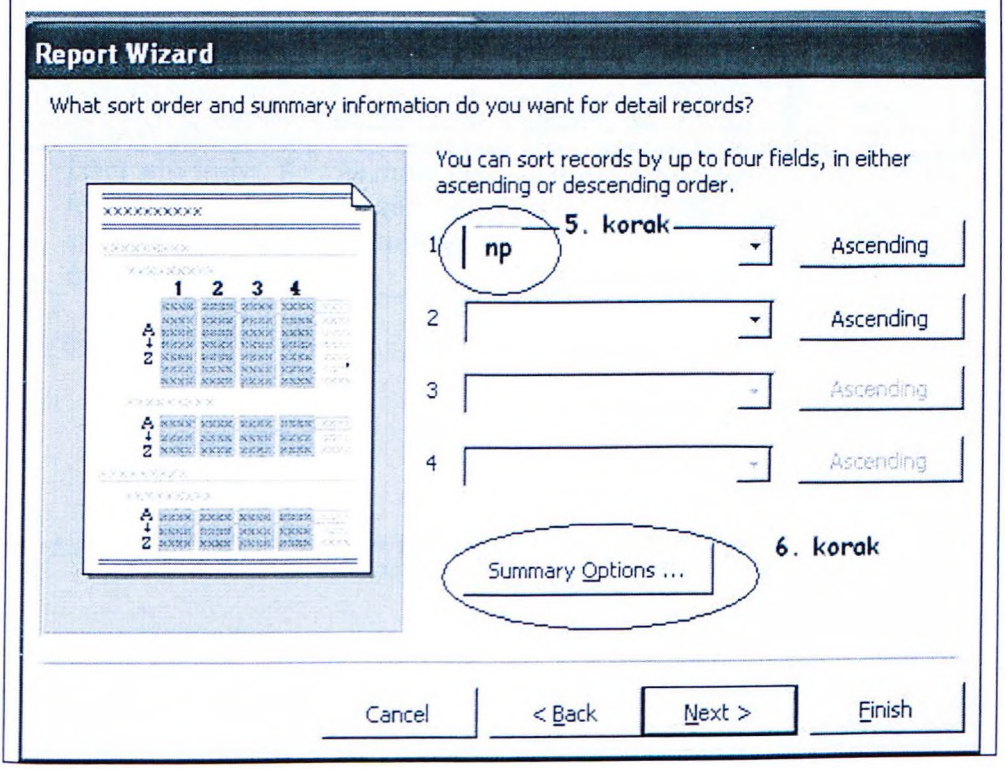
1. U okviru izvješća pozove se izrada izvješća pomoću čarobnjaka.
2. Izabere se osnova za izvješće, u našem slučaju upit o narudžbama s vrijednostima, koji se upravo kreiralo u polju s tekstom za izbor tablice ili upita.
3. Izaberu se polja za izvješće.



4. Iz okvira koji slijedi izabere se polje po kojem se želi grupirati podatke u izvješću.

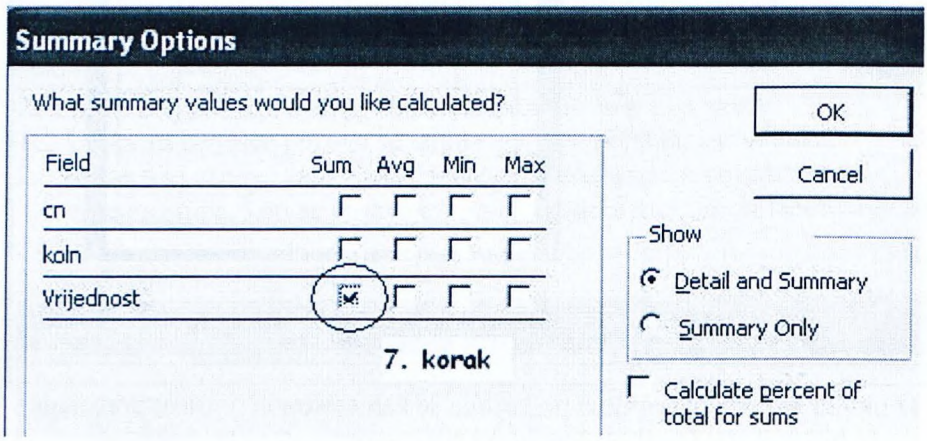


5. Iz okvira se izaberu pojmovi po kojima se želi sortirati izvješće. Sortiranje može biti uzlazno (engl. *Ascending*) i silazno (engl. *Descending*).

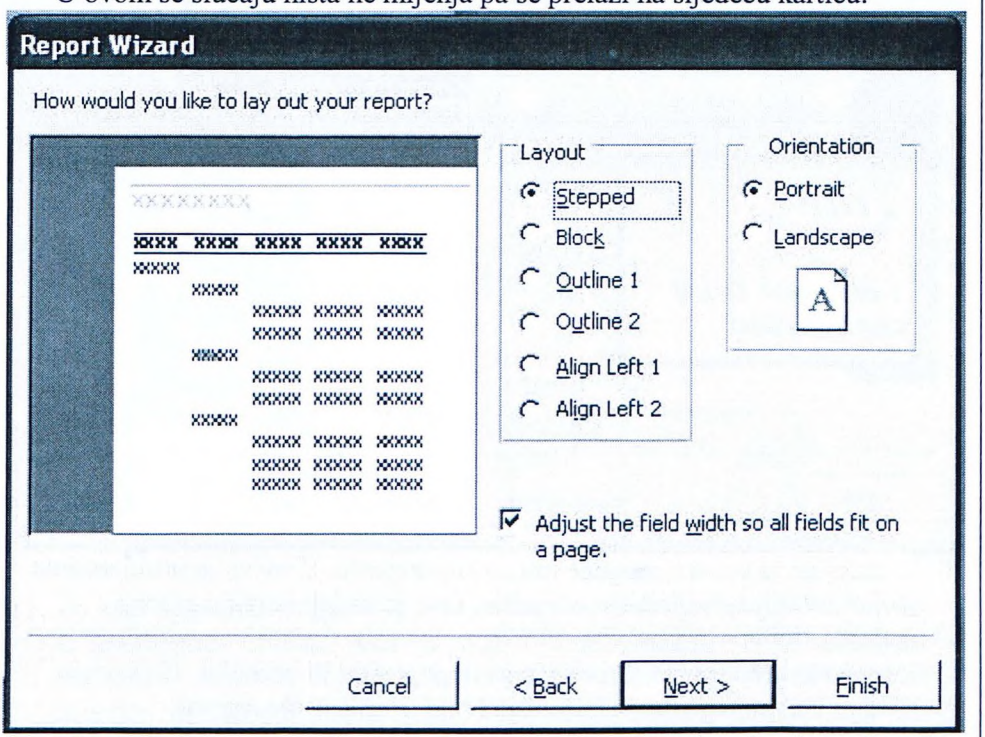


6. Prije prelaska na sljedeću karticu izabere se podešavanje opcija za grupe podataka.

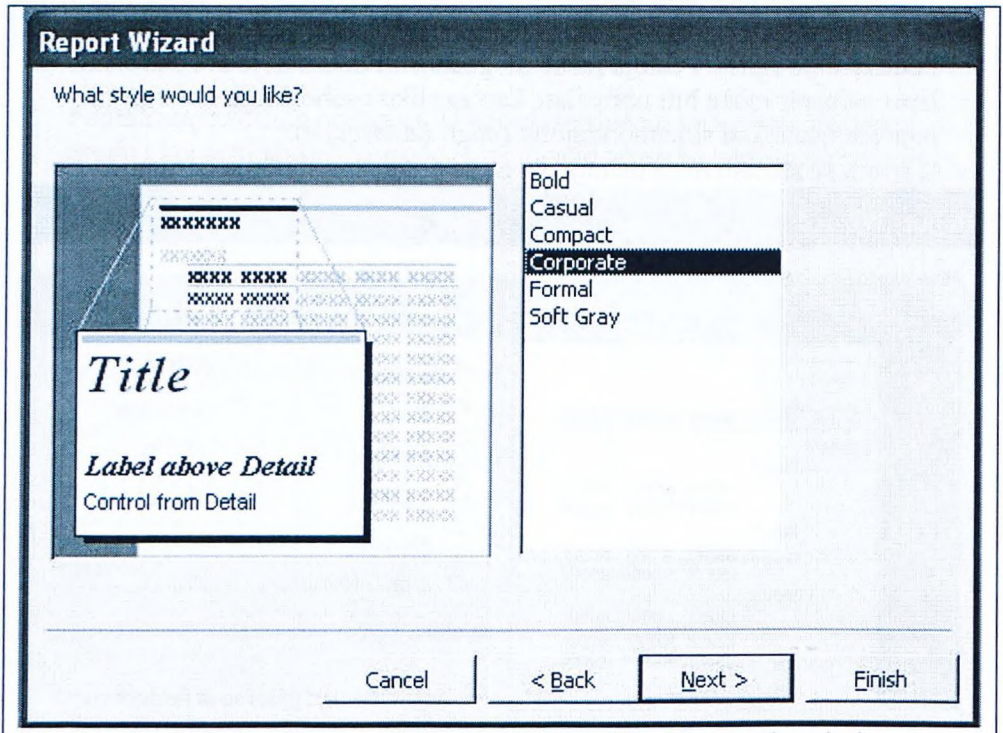
7. Podešavanje zbirnih podataka. U našem se slučaju traži zbroj za svaku grupu i konačni zbroj. Moguće je računanje prosjeka, ispis najmanje ili najveće vrijednosti za grupu, izračun postotaka i sl.



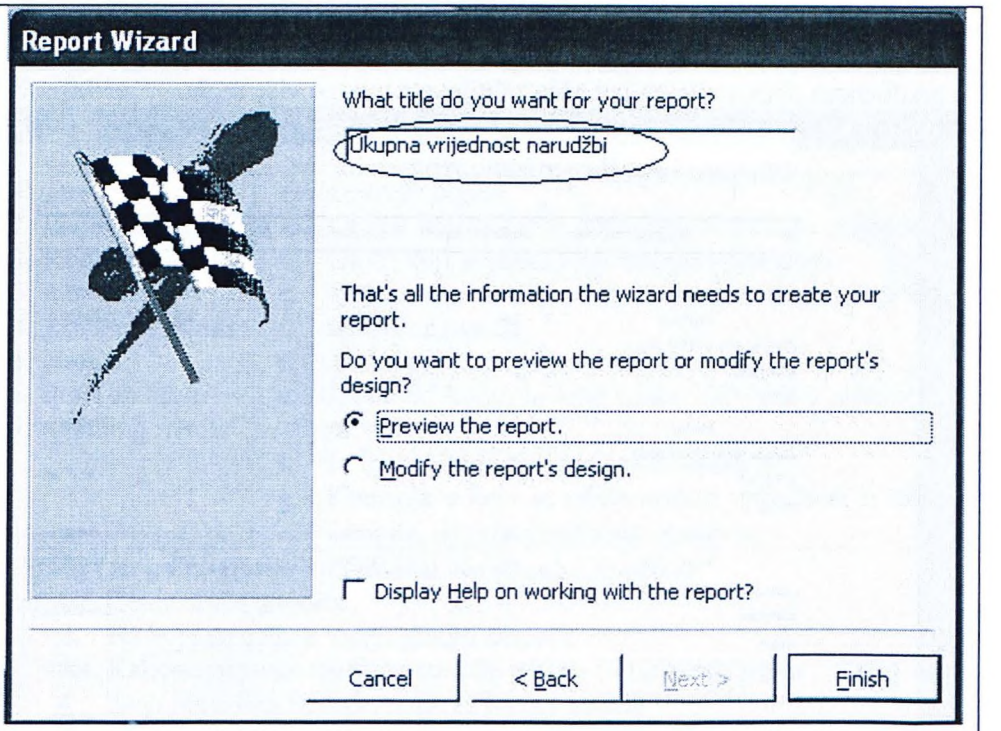
8. Podešavanje ispisa. Postoje razne mogućnosti i dobro ih je sve isprobati. Ispis na papir može biti normalan, kao za sliku osobe (engl. *Portrait*) ili poprečan, kao kad slikamo krajolik (engl. *Landscape*).
U ovom se slučaju ništa ne mijenja pa se prelazi na sljedeću karticu.



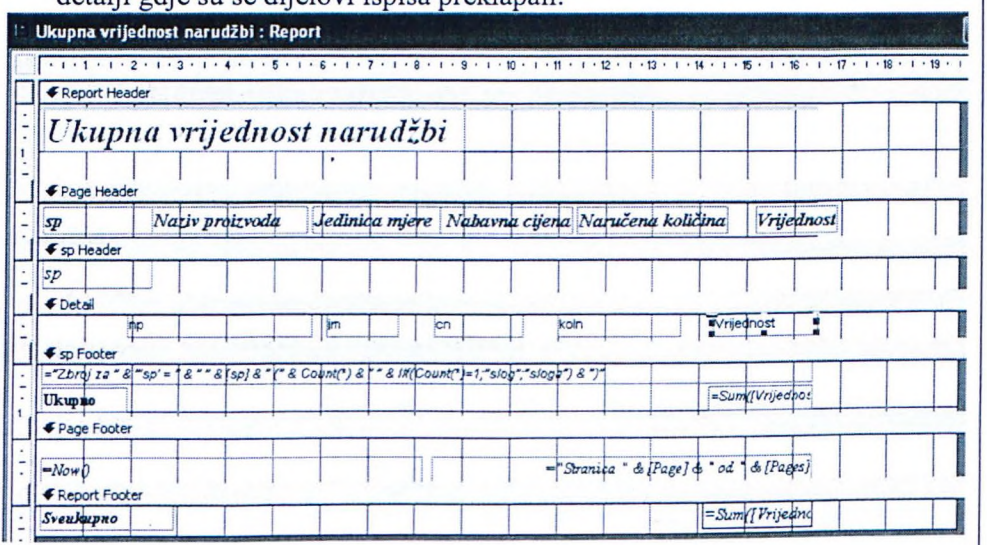
9. Izbor stila ispisa. Postoje razna slova za ispis naslova i podnaslova. I ovdje je korisno istražiti sve mogućnosti, a u ovom je slučaju izabran format za službena izvješća (engl. *Corporate*). Nakon toga se prelazi na sljedeći korak čarobnjaka.



10. U zadnjem je koraku moguće imenovati izvješće. U ovom je primjeru ime izvješća *Ukupna vrijednost narudžbi*. Ovo je zadnji korak u gradnji izvješća.
    Kada je izvješće gotovo, moguće ga je pogledati ili preraditi. U ovom je slučaju izabrano gledanje izvještaja (engl. *Preview the report*).



11. Po želji ili potrebi moguće je gotovo izvješće otvoriti u obliku za tiskanje ili u gradbenom obliku. Ako se otvori u gradbenom obliku, moguće je mijenjati i dodavati objekte u izvješće, pisati dijelove teksta na hrvatskom umjesto engleskom jeziku i slično. U ovom su primjeru u gradbenom načinu prevedeni dijelovi teksta, neka su polja proširena i dotjerani neki detalji gdje su se dijelovi ispisa preklapali.



Rezultat je ovog koraka konačno izvješće spremno za tisak. Postupak je dobivanja ovog izvješća pomoću čarobnjaka znatno olakšan jer bi ručna izrada u gradbenom obliku cijelog izvješća zahtijevala mnogo više truda.

Ukupna vrijednost narudžbi

Ukupna vrijednost narudžbi

sp	*Naziv proizvoda*	*Jedinica mjere*	*Nabavna cijena*	*Naručena količina*	*Vrijednost*
0001					
	Kruh bijeli	kom	5,00 kn	50	250,00 kn
	Kruh bijeli	kom	5,00 kn	200	1.000,00 kn
Zbroj za 'sp' = 0001 (2 stoga)					
Ukupno					*1.250,00 kn*
0002					
	Pecivo	kom	0,80 kn	100	80,00 kn
Zbroj za 'sp' = 0002 (1 stog)					
Ukupno					*80,00 kn*
0003					
	Keksi	kg	20,00 kn	32	640,00 kn
	Keksi	kg	20,00 kn	150	3.000,00 kn
Zbroj za 'sp' = 0003 (2 stoga)					
Ukupno					*3.640,00 kn*
0004					
	Mlijeko	l	4,00 kn	24	96,00 kn
Zbroj za 'sp' = 0004 (1 stog)					
Ukupno					*96,00 kn*
Sveukupno					*5.066,00 kn*

Page: 1

Pitanja za ponavljanje uz poglavlje 6

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–f).

Pojam	*Objašnjenje pojma*
1. Izvješća	a. Objekt koji se može dodati na obrazac ili izvješće.
2. Kutija s alatima	b. Okvir koji pomaže u računanju vrijednosti.
3. Kontrola	c. Objekt koji služi za izgradnju dokumenta koji će se tiskati na pisaču.
4. Kontrolni gumb	d. Traka s alatima koja sadrži kontrole.
5. Natpis	e. Jedna od kontrola koja može biti veza s makroima i modulima.
6. Polja za tekst	f. Jedna od kontrola koja sadrži opise.
7. Graditelj izraza	g. Kontrola u koju se može upisati vrijednost iz tablice ili upita, ili vrijednost koja se računa.

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Čemu služe izvješća?
2. Na koje se načine mogu graditi obrasci?
3. Kako se uspostavlja veza između tablice ili upita i izvješća?
4. Koja je razlika između polja za tekst i natpisa?
5. Koja je razlika između obrasca i izvješća?
6. Koji su engleski nazivi za pojmove obrasci, izvješća, kutija s alatima, kontrola, gumb, natpis i polje za tekst?
7. Kada se izvješće gradi u pogledu gradnje?
8. Može li se izvješće izgrađeno pomoću čarobnjaka dograditi u pogledu gradnje?
9. Može li se izvješće izgrađeno u pogledu gradnje dograditi pomoću čarobnjaka za izradu izvješća?

Dodatna pitanja za istraživanje:

1. Koje čarobnjake za izradu obrasca poznajete?
2. Koje kontrole postoje u kutiji s alatima, osim onih obrađenih u ovome poglavlju?
3. Mogu li na obrascu postojati dva ista natpisa?

Primjeri za vježbu:

1. Izradite sve obrasce iz ovog poglavlja i pokrenite ih.
2. Osmislite obrasce za bazu koju ste vi kreirali.

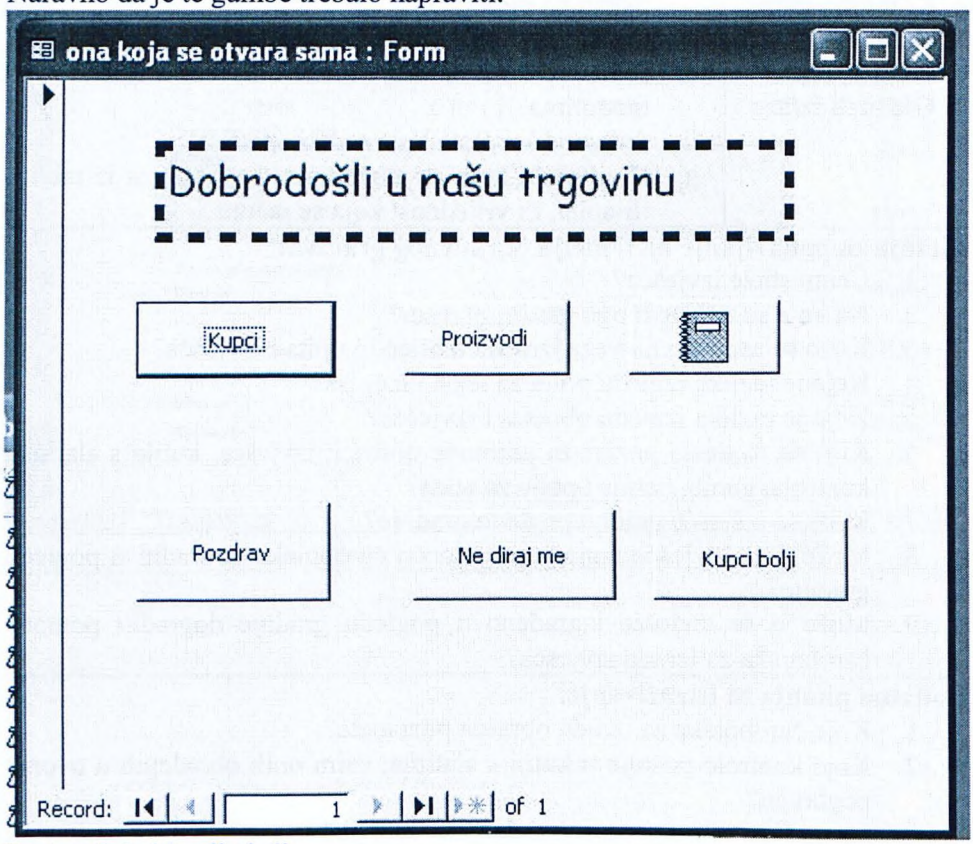
Preporučena literatura za daljnje istraživanje:

1. Pomoć unutar programa Access.
2. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
3. Šavle, S, *Access*, Rijeka, Adamić, 2004.
4. Pejić-Bach, M, Spremić, M, Vlahović, N, *Priručnik za vježbe iz informatike*, Zagreb, Mikrorad, 2003.

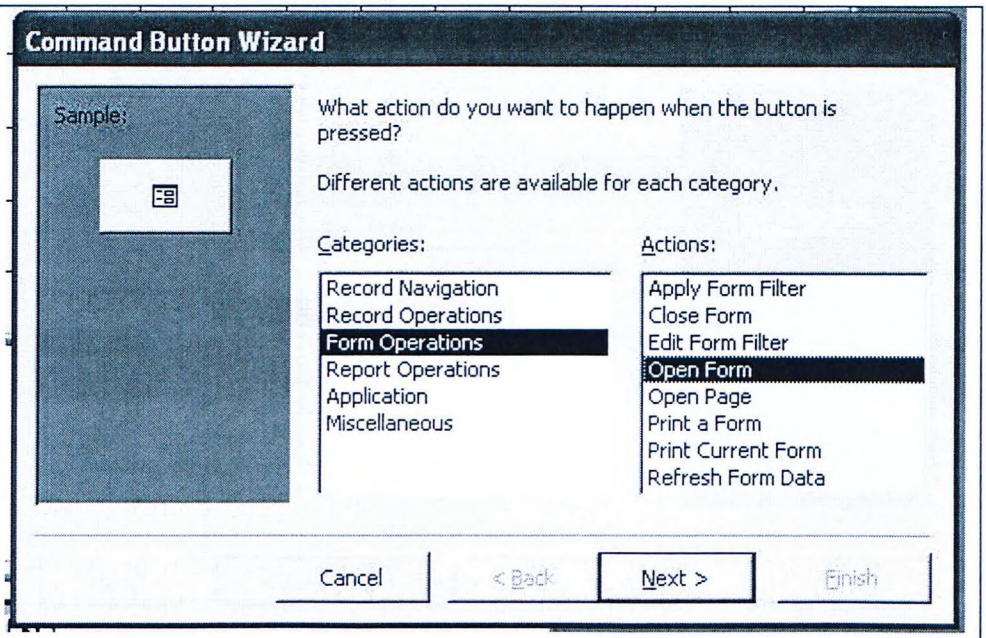
7. Makroi (engl. Macros)

Makroi (engl. *Macros*)	**Makroi** (engl. *Makros*) su objekti koji omogućavaju automatiziranu obradu unutar baza. Posebno su podesni za radnje poput otvaranja i zatvaranja prozora, prelazak na sljedeći objekt i sl.

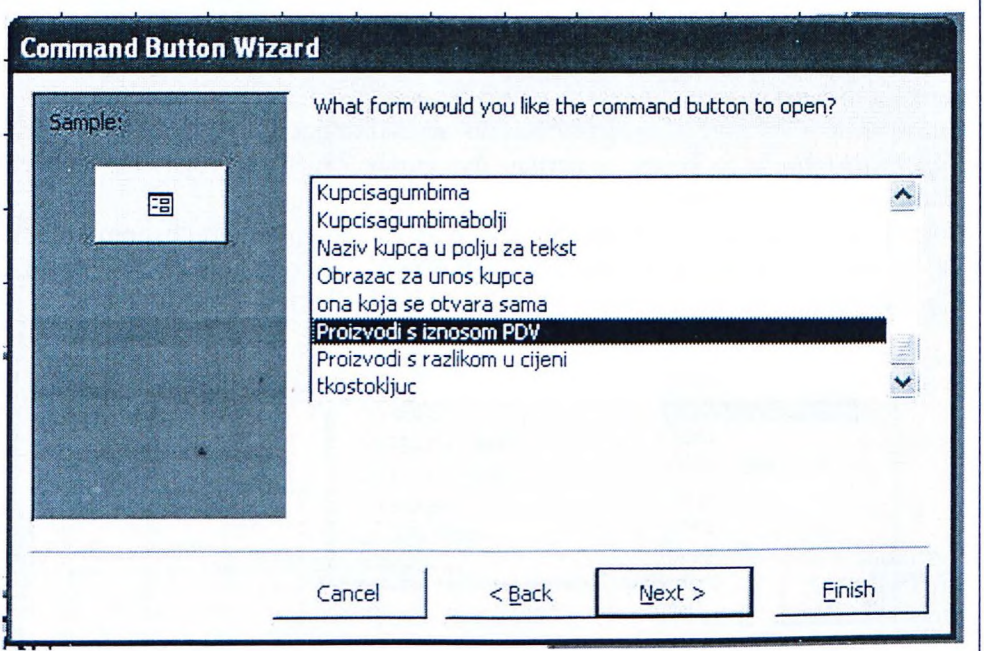
Primjer: Na obrascu koji se otvara čim se baza pokrene imamo niz gumbova. Naravno da je te gumbe trebalo napraviti.



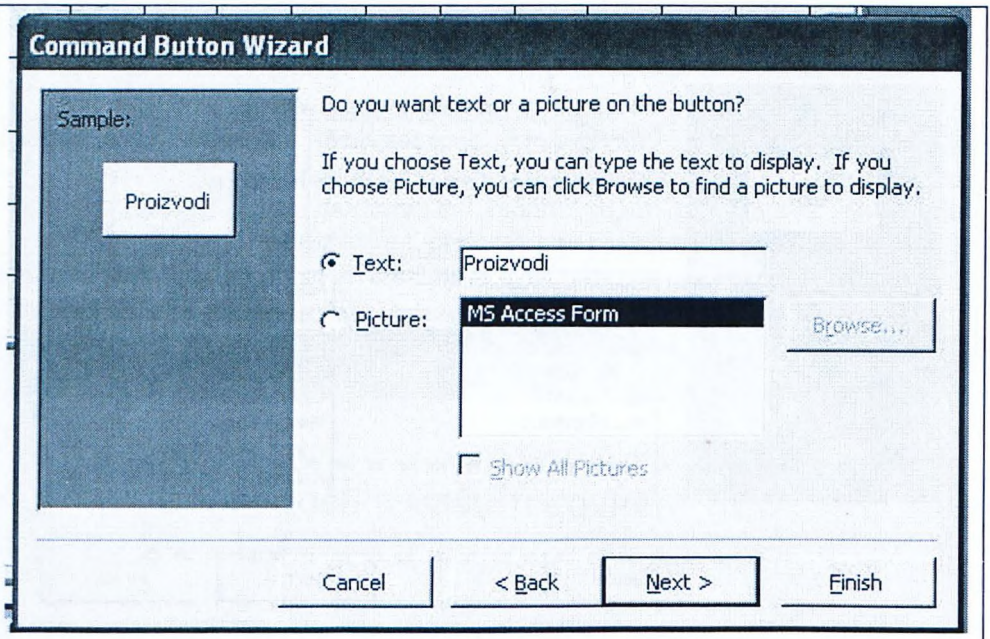
Postupak je bio sljedeći:
Gumb *Proizvodi* i onaj s nacrtanom fasciklom sačinjeni su pomoću čarobnjaka za izradu gumba.
Gumb *Proizvodi* načinjen je tako da se na formu u gradbenom pogledu doda gumb, a onda se taj gumb pomoću čarobnjaka poveže s operacijama koje se odnose na formu, i to na otvaranje forme.



Zatim se odabere forma među formama koje postoje u bazi, npr. *Proizvodi s iznosom PDV*, koja je ranije napravljena.



Na gumb se unese tekst *Proizvodi* (mogla se staviti i slika forme).

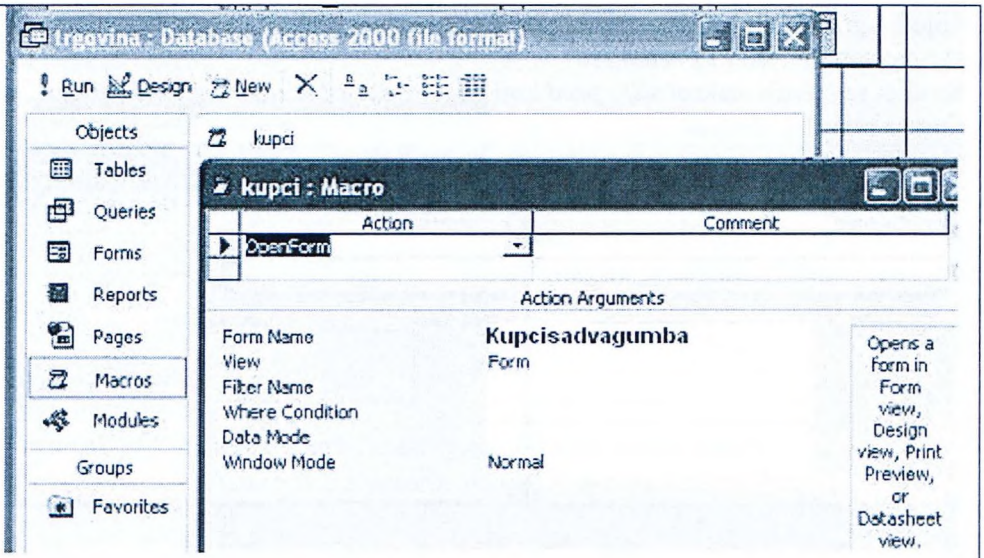


Gumb s nacrtanom fasciklom otvara izvješće *Ukupna vrijednost narudžbi* koje je izgrađeno u poglavlju o izvješćima. I taj je gumb sagrađen pomoću čarobnjaka. U ovom se slučaju otvaraju izvješća, a na gumbu je izabrano da bude slika umjesto teksta.

Ovaj postupak generira automatski kod u VBA, ali se isto moglo postići i pomoću makroa.

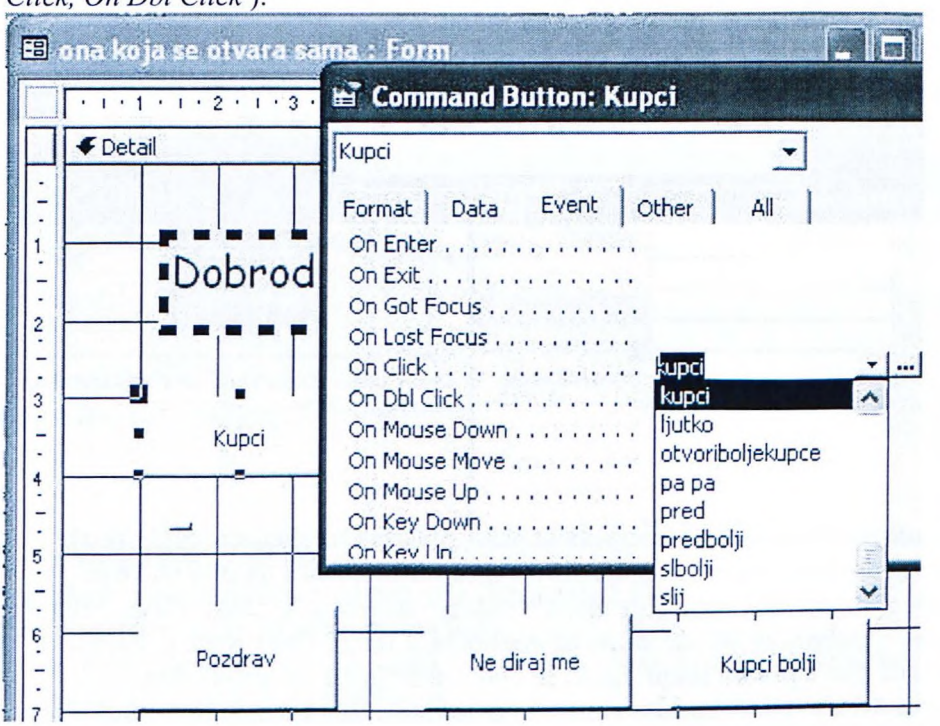
Gumb *Kupci* pokreće se pomoću makroa. Prethodila je izrada obrasca *Kupcisadvagumba* na kojem su ucrtana dva gumba i ručno u svojstvima obrasca uklonjen gumb za navigaciju.

Najprije se u makroprozoru pokrene izrada novog makroa u gradbenom obliku. Taj će makro biti sačinjen samo od jedne akcije, i to one koja otvara obrazac *Kupcisadvagumba*. Taj se makro sprema pod imenom *Kupci*.

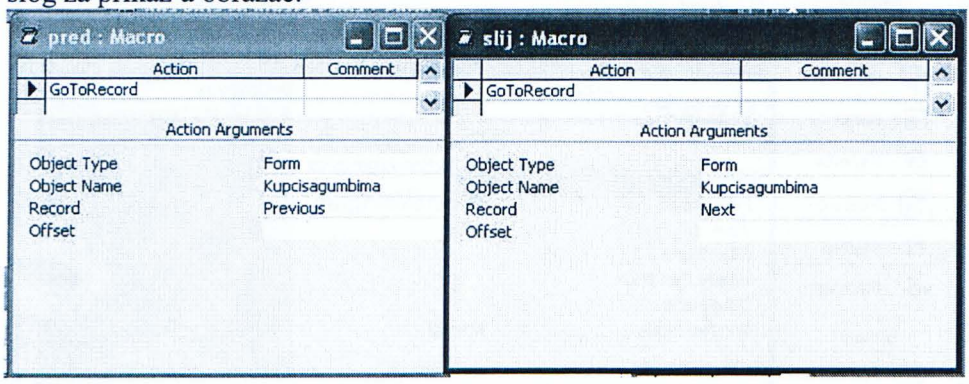


Sada je još potrebno taj makro povezati s gumbom na formi koja se otvara na početku rada baze.

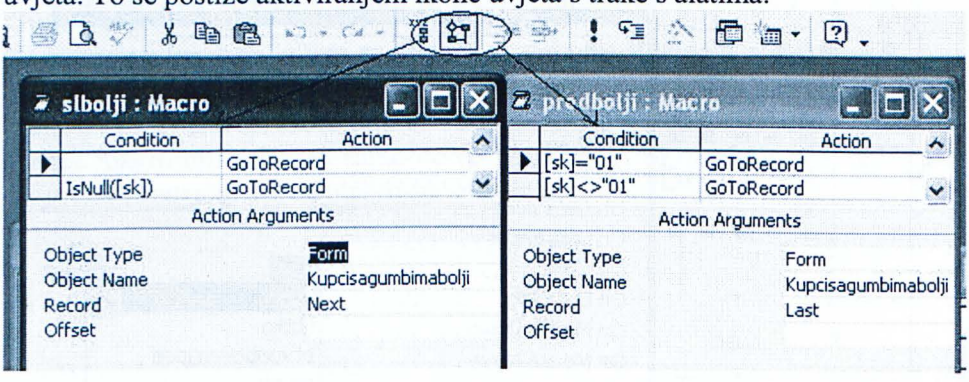
To se radi tako da se najprije napravi komandni gumb koji se ručno unutar svojstava objekta gumba *Kupci...* na kartici događaja (engl. *Event*) poveže s makroom *Kupci...* uz postupak pritiska i dvostrukog pritiska na miš (engl. *On Click, On Dbl Click*).



Slijedi ugrađivanje makroa na obrazac *Kupcisadvagumba*. On ima dva gumba sa natpisom *Sljedeći* i *Prethodni*.
Na slici se nalaze makroi *slij* i *pred* koji pozivaju sljedeći, odnosno prethodni slog za prikaz u obrazac.

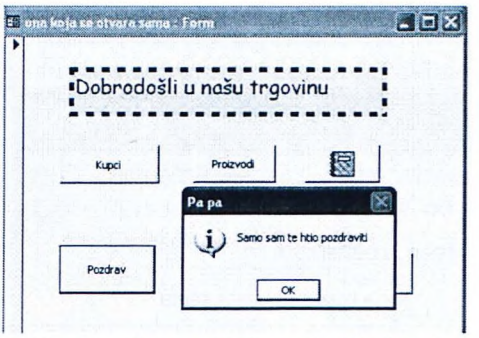


Oni se također trebaju povezati s gumbima u formi *Kupcisadvagumba* u kartici događaja unutar svojstva gumba.
Slijedi testiranje forme. Gumbi dobro funkcioniraju, osim kada se dođe na rub skupa, dakle na prvi i na zadnji slog. Ako se pokuša ići prije prvog ili iza zadnjeg, pojavljuju se poruke o pogrešci.
Pogreške se mogu izbjeći ako se makroi malo dorade. Na slici su bolji makroi koji se, ako se želi ići iza zadnjeg, ciklički nastavljaju na prvi slog, odnosno ako se želi ići prije prvog, prebacuje se na zadnji.
Da bi se upisale te naredbe, bilo je potrebno otvoriti mogućnost postavljanja uvjeta. To se postiže aktiviranjem ikone uvjeta s trake s alatima:



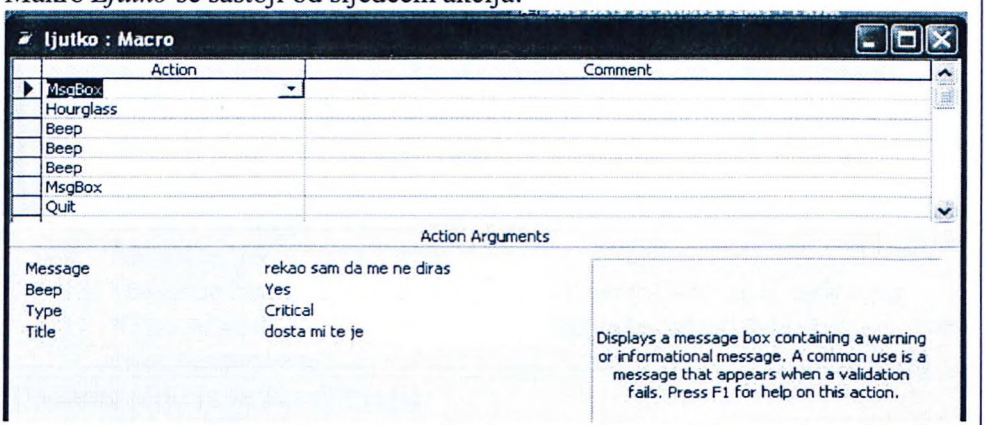
U makrou *slbolji* prva je akcija kao i prije, prelazak na sljedeći (engl. *Next*). Ako dođe do slučaja da je polje *Šifra kupca* prazno, znači da smo na kraju tablice. U tom se slučaju daje uputa o odlasku na prvi slog (engl. *First*). Slično tome, u makrou *predbolji* ide se na prethodnika (engl. *Last*), osim u slučaju kada se radi o prvom slogu. Samo se u tom slučaju ide na zadnji slog.
Rješenja za bolju navigaciju povezuju se nakon izrade s gumbom *Kupci bolji*.

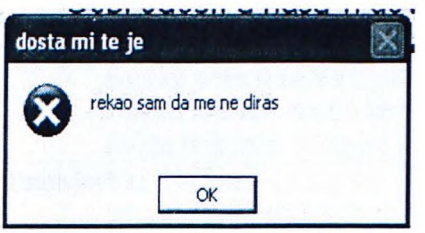
Gumb *Pozdrav* rezultira pozivanjem okvira s pozdravom.



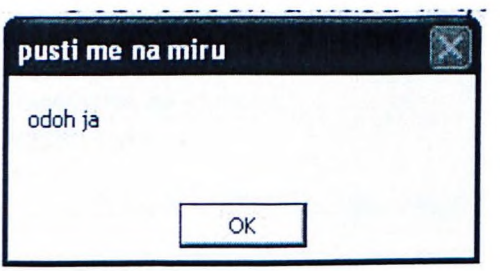
To se postiže izradom makroa od samo jedne akcije koja aktivira okvir sa pozdravom:
Akcija je MsgBox, čiji je tekst (engl. *Message*) "Samo sam te htio pozdraviti", a naslov (engl. *Title*) "Pa pa".

Pritisak na komandni gumb *Ne diraj* me pokreće makro *Ljutko*.
Makro *Ljutko* se sastoji od sljedećih akcija:



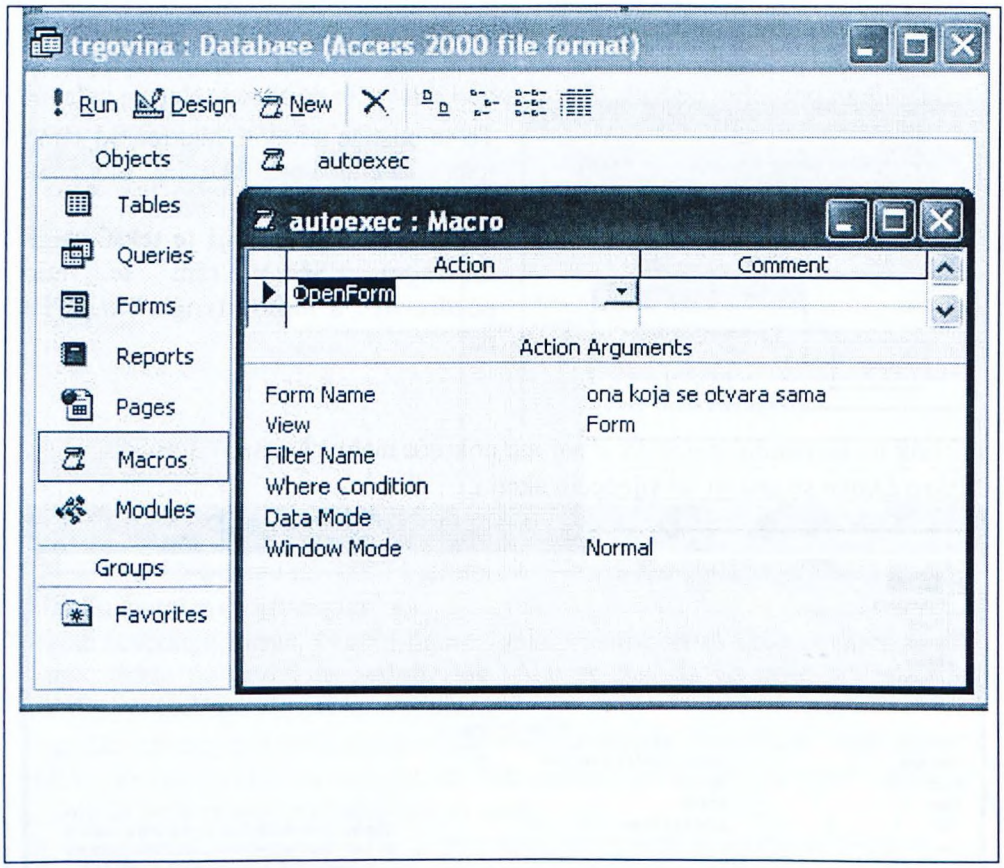


Poziva ga se komandnim gumbom *Ne diraj me*, a rezultat je uzastopna pojava dvaju okvira s porukama, nakon kojih se napušta Access.



Zadatak 1: Izradite makro *Ljutko* i povežite ga s gumbom *Ne diraj me*.

Ako se želi pokrenuti makro prilikom aktiviranja baze, potrebno je tom makrou dati ime *Autoexec*. Svaka baza može imati samo jedan makro pod tim imenom.
U slučaju naše baze *Autoexec* Makro će otvoriti obrazac na kojem se nalaze svi gumbi koji su povezani s makroima.


trgovina : Database (Access 2000 file format)
Run
Design
New
Objects
Tables
Queries
Forms
Reports
Pages
Macros
Modules
Groups
Favorites
autoexec
autoexec : Macro
Action
Comment
OpenForm
Action Arguments
Form Name
ona koja se otvara sama
View
Form
Filter Name
Where Condition
Data Mode
Window Mode
Normal

Pitanja za ponavljanje uz poglavlje 7

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 3–a).

Pojam	*Objašnjenje pojma*
1. Makroi	a. Makro koji se automatski izvršava prilikom pokretanja Access baze. 3
2. Akcije	b. Objekti od kojih se gradi makro.
3. Autoexec	c. Objekti koji omogućavaju automatizaciju nekih postupaka nad Access bazom. 1
4. Događaj	d. Podražaj koji omogućava pokretanje makroa.

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Čemu služe makroi?
2. Što su akcije?
3. Na koje se načine mogu graditi makroi?
4. Koji događaji mogu pokrenuti neki makro?
5. Kako se uspostavlja veza između makroa i događaja koji ga pokreće?
6. Može li postojati makro od samo jedne akcije?
7. Može li se na postojeći makro dodati akcija?
8. Može li se akcija u makrou uvjetovati?
9. Nabrojite nekoliko različitih akcija.
10. Nabrojite barem tri razna događaja koji mogu pokrenuti makro.
11. Kako se mora zvati makro koji se pokreće automatski čim se otvara neka Access baza?

Dodatna pitanja za istraživanje:

1. Postoje li čarobnjaci za izradu makroa?
2. Mogu li se akcije unutar makroa grupirati?
3. Koliko makro može imati akcija?
4. Navedite neku akciju koja nije spomenuta u ovom poglavlju.
5. Može li Access baza imati dva različita Autoexec makroa?

Primjeri za vježbu:

1. Izradite sve makroe iz ovog poglavlja i pokrenite ih.
2. Osmislite ti razna makroa za bazu koju ste vi kreirali.

Preporučena literatura za daljnje istraživanje:

1. Pomoć unutar programa Access.
2. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
3. Šavle, S, *Access*, Rijeka, Adamić, 2004.

8. Moduli (engl. Moduls)

Moduli (engl. *Moduls*)	**Moduli** (engl. *Moduls*) su objekti koji sadrže kod u VBA (*Visual Basic for Applications*) jeziku. Na taj se način može postići veći stupanj automatizacije rada nego pomoću makroa. Programiranje je u VBA dosta zahtjevno, tako da se ovdje daju samo neke osnovne napomene o programiranju.

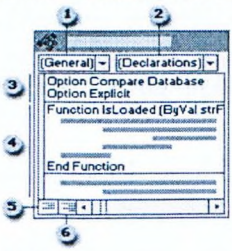
Show All

About modules

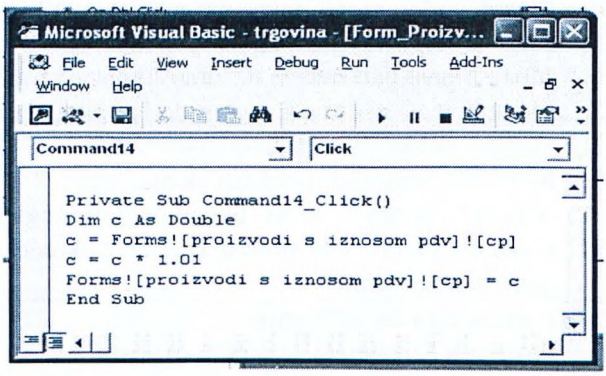
A module is essentially a collection of declarations, statements, and procedures stored together as one named unit to organize your Microsoft Visual Basic code. Microsoft Access has two types of modules: standard modules and class modules.

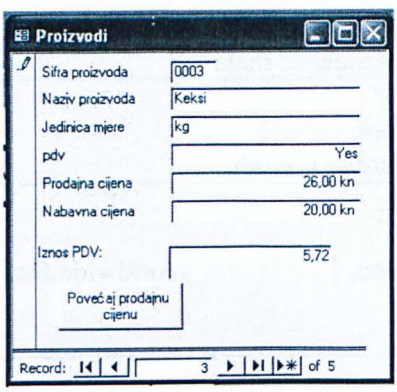
What is a module?

A module is a collection of Visual Basic declarations and procedures that are stored together as a unit.



1 **Object** box
2 **Procedure** box
3 Declarations
4 Procedures
5 **Procedure View** button
6 **Full Module View** button
▸ Class modules
▸ Standard modules



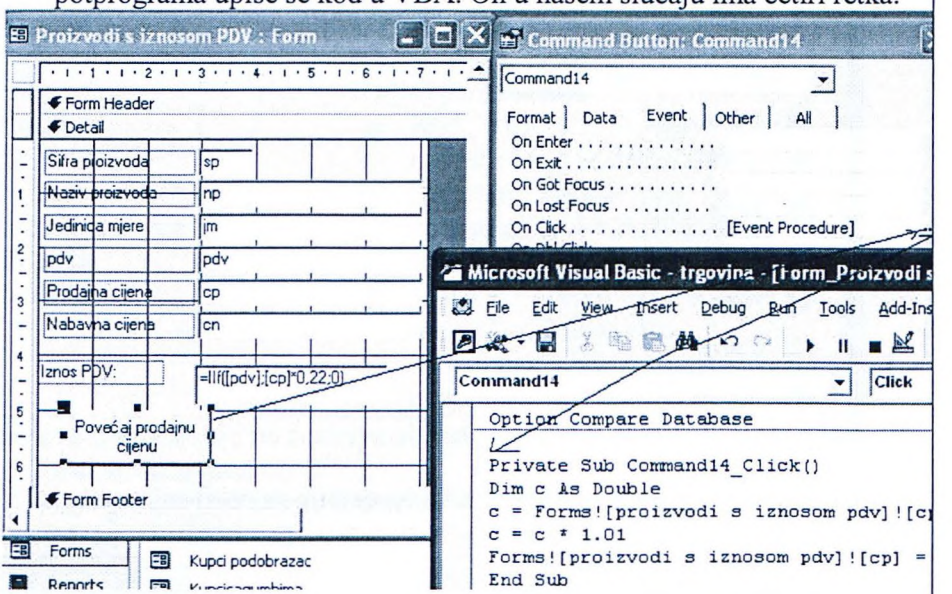


Primjer: Na obrascu za proizvode želimo dati mogućnost da se za neki proizvod prodajna cijena mijenja u koraku od 1%. To se može postići tako da se napiše program (modul) koji obavlja tu operaciju, koji će se aktivirati događajem, primjerice pritiskom na gumb.

Taj se obrazac koji sadrži gumb i kod za promjenu prodajne cijene gradi ovako:

1. Najprije se na obrazac u gradbenom obliku doda gumb.
2. Zatim se izabere gumb i za taj gumb i događaj unutar njega (*On Click*) aktivira unos koda ...

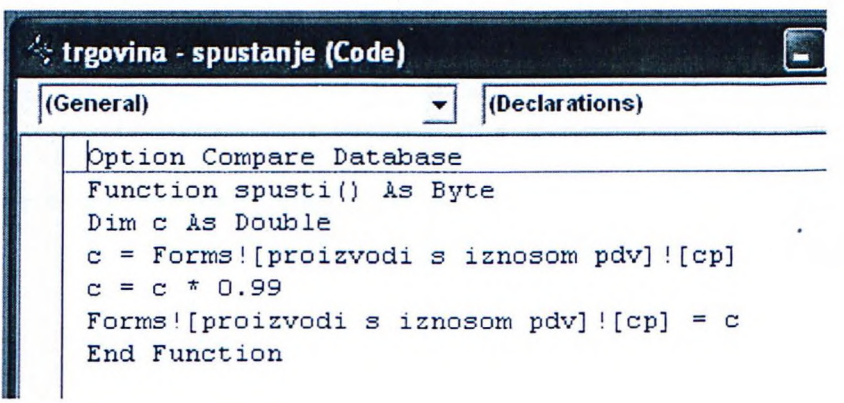
3. Između *Private Sub* i *End Sub* naredbi koje označavaju početak i kraj potprograma upiše se kod u VBA. On u našem slučaju ima četiri retka.



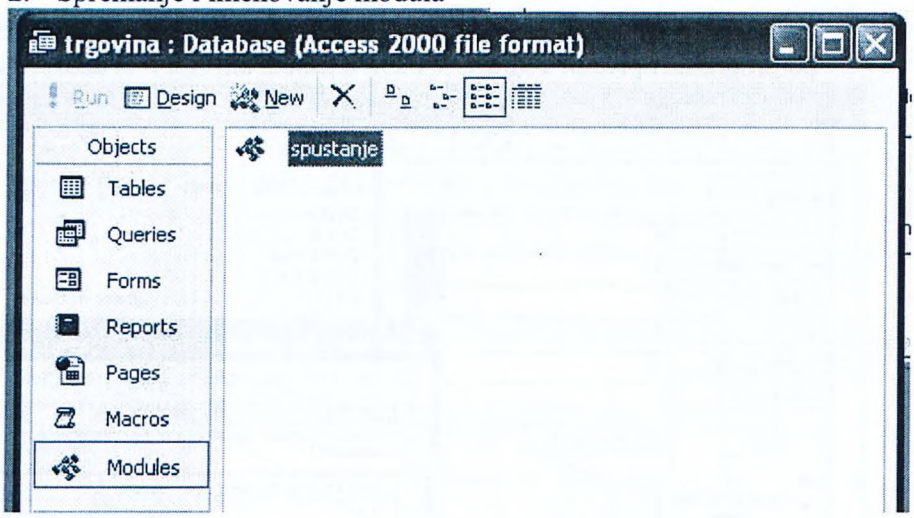
Za proizvod kojem se želi podignuti cijena za 1%, bit će dovoljno kliknuti gumb.

Drugi je način da se kod upiše u modul, a pomoću procedure se poziva modul.

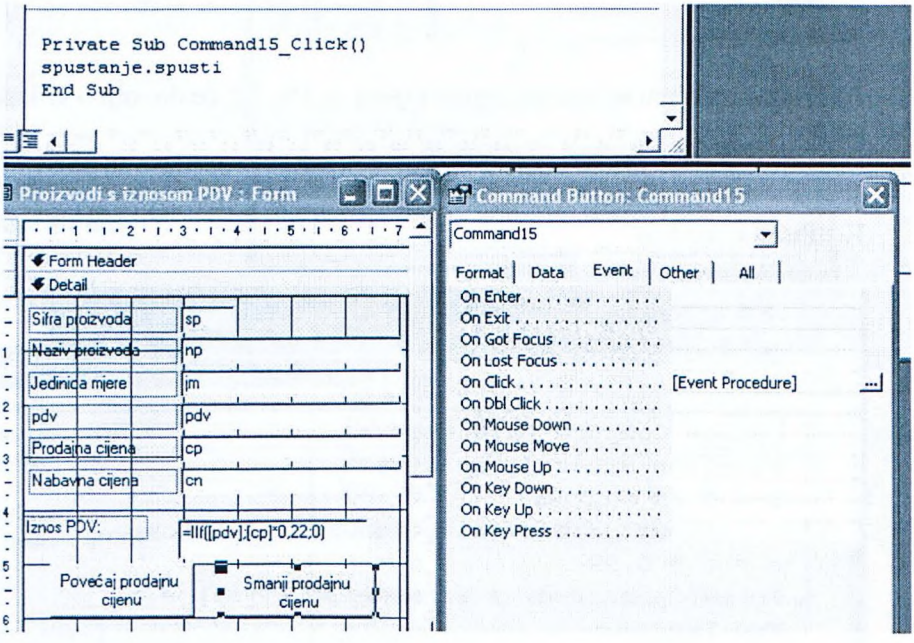
1. Pisanje modula



2. Spremanje i imenovanje modula



3. Kreiranje gumba i pisanje kratke procedure za pozivanje modula.



Prednost je ovoga drugog načina da se rutina koja se nalazi u modulu može pozivati iz raznih objekata, a ne samo iz jednog obrasca u kojem je kreirana.

Pitanja za ponavljanje uz poglavlje 8

Spajalica

Svaka riječ iz desnog stupca ima objašnjenje u lijevom stupcu. Uspostavite vezu pomoću rednog broja pojma i objašnjenja (npr. 5–e).

Pojam	*Objašnjenje pojma*
1. Moduli	a) Kod napisan u VBA jeziku koji se pokreće kada se događaj dogodi.
2. VBA	b) Program napisan u VBA jeziku i pohranjen u objekte "Moduli".
3. Gumb	c) Linija koda napisana u VBA.
4. Procedura uz događaj	d) Jedna od kontrola koja može biti veza s makroima i modulima.
5. Naredba	e) Programski jezik u kojem se pišu moduli i procedure. 2

Pitanja za ponavljanje na temelju obrađenog gradiva:

1. Čemu služe moduli?
2. U kojem se jeziku pišu moduli?
3. Može li se cijeli kod napisati kao procedura uz događaj?
4. Koja je prednost od pisanja programa u objektima moduli?

Dodatna pitanja za istraživanje:

1. Koja je razlika između makroa i modula?
2. Koristi li se modul ili makro za onaj postupak koji se aktivira automatski nakon otvaranja baze?
3. Hoćete li koristiti makro ili modul za otvaranje nekog obrasca?
4. Može li kontrolni gumb pokrenuti neki modul?
5. Može li kontrolni gumb pokrenuti neki makro?
6. Mogu li na istom obrascu postojati gumbi od kojih neki pokreću makroe, a drugi module?

Primjeri za vježbu:

1. Izradite sve module iz ovog poglavlja i pokrenite ih.
3. Osmislite module za bazu koju ste vi kreirali.

Preporučena literatura za daljnje istraživanje:

1. Pomoć unutar programa Access.
2. *Microsoft Access 2000 Bible*, Gold Edition, IDG Books Worldwide Inc, 1999.
3. Šavle, S, *Access*, Rijeka, Adamić, 2004.
4. Pejić-Bach, M, Spremić, M, Vlahović, N, *Priručnik za vježbe iz informatike*, Zagreb, Mikrorad, 2003.

9 789539 566317