



**[1] OSNOVE UPRAVLJANJA
RELACIONIM BAZAMA PODATAKA**

R.S. Mikanović



Sadržaj

1 OSNOVNI POJMOVI I STANDARD	1
Standard u projektovanju baza podataka.....	3
ENTITETI I VEZE MEĐU NJIMA	4
Veza 1:1	4
Veza 1:n (n:1).....	4
Veza n:n	5
Rekursivna veza	5
Primer povezanosti entiteta jedne baze podataka.....	6
UVOD U MS ACCESS	7
Objekti MS Access-a	7
OTVARANJE baze iz MS Accessa	8
Database Wizard	8
Na prvi pogled.....	8
Database Window.....	9
Sistem menija.....	9
TABELA I NJENA ORGANIZACIJA	11
Tabela, polje, zapis	11
Atribut, domen i relacija	11
Kreiranje tabela Wizard-om.....	11
Indexiranje i ključevi	11
OSNOVE UPRAVLJANJA TABELAMA	13
Kreiranje tabela	13
Table Wizard	14
Dizajn tabele	15
Lookup Wizard.....	17
Imenovanje objekata	17
Sigurnost baze i brisanje objekata ^①	17
NORMALIZACIJA BAZE PODATAKA	20
Pravilo 1: Eliminisanje grupa podataka koje se ponavljaju	20
Pravilo 2: Eliminacija redundantnih podataka.....	20
Pravilo 3: Eliminisanje kolona koje ne zavise od primarnog ključa	20
Pravilo 4: Izolacija nezavisnih višestrukih relacija.....	21
Pravilo 5: Izolacija zavisnih višestrukih relacija.....	21
7 RELACIONI MODEL.....	22
Veza i relacija.....	22
Relacije tipa 1:n i 1:1.....	22
Rekursivna veza	22
Problemi pri definisanju relacije	24
Referencijalni integritet.....	24
UPRAVLJANJE UPITIMA.....	25
Kreiranje upita	25
Dizajn upita.....	26
OSNOVE SQL-a	27
Šta je to SQL?.....	27
Naredba SELECT..FROM.....	28
Klauzula ORDER BY.....	28
Klauzula WHERE.....	29
SQL funkcije za agregaciju.....	30
Klauzule DISTINCT i DISTINCTROW.....	31
Klauzule TOP n i TOP n PERCENT.....	31
Klauzula GROUP BY..HAVING	31
Tipovi SQL klauzule JOIN.....	32
UNION upiti.....	33
Unakrsni upiti klauzulom TRANSFORM..PIVOT.....	34
FORME	35
Kreiranje forme	35
Dizajniranje forme	36

Kontrole i tipovi kontrola.....	36
Sekcije forme.....	37
Subform (podforma).....	37
Programiranje forme ^①	38
IZVEŠTAJI	39
Kreiranje izveštaja	39
Dizajn izveštaja.....	39
Page Setup.....	40
IZRADA STANDARDNE BAZE PODATAKA	41
13 DODATAK	43
FAQs (Frequently Asked Questions).....	43
SPECIFIKACIJE ZA MS Access 97/2000.....	44
TASTERSKE PREĆICE.....	46

Pre svega,

dobro došli u svet lakog i, iznad svega, zanimljivog ostvarenja programerskih ideja. Svaki program zahteva obradu podataka, pa krenimo baš od toga – BAZE PODATAKA i naučimo to na najbolji način.

Kurs *MS Access –(1) Osnove upravljanja relacionim bazama podataka* je osmišljen kao jedan oblik ulaska u ovu kompleksnu materiju, ali na lakši način.

Da bi mogao da prati kurs na efikasan način, od polaznika se zahteva da ima ozbiljno iskustvo u obraditeksta (*MS Word*), kao i u obradi tabela (*MS Excel*), a pohađanje ovog kursa se savetuje:

- korisnicima razvijenih aplikacija pod *MS Access*-om,
jer će naučiti standard u korišćenju *MS Access* baza podataka i informacionih sistema
- programerima koji žele da u svoje buduće projekte uključe baze podataka, jer će naučiti da kvalitetno upravljaju onim objektima koji su i najznačajniji za kvalitetnu bazu podataka (tabele i upiti)
- onima koji žele da se profesionalno bave *MS Access*-om,
jer će naučiti kako se jednostavno mogu dizajnirati obrasci i izveštaji, a edukaciju mogu nastaviti kroz više kurseve *MS Access*-a.

U toku trajanja kursa polaznici bi trebalo da ovladaju osnovnim tehnikama korišćenja *MS Access*-a, ali i kreiranja jednostavnih relationalnih baza podataka. Pri radu polaznici mogu odabrat jednu od verzija *MS Access*-a verzija 8.0 iz paketa *MS Office 97 - Professional Edition* ili verzija 9.0 Microsoft Office 2000- Premium. Kroz teme kursa obrađuje se relationalni model baza podataka, što znači da će se izučavati objekti baze i veze između objekata (relacije).

Radi lakšeg razumevanja tematike Kurs je podeljen na grupe tema A) Osnovni pojmovi, principi i standard relationalnih baza podataka (gde će biti prikazano nekoliko primera edukativnih i poslovnih baza podataka), B) atim malo teorije, C) radno okruženje *MS Access*-a, organizacija tabela, njihovo kreiranje i dizajniranje, D) upiti (*Queries*) i SQL, E) osnove kreiranja obrazaca (*Forms*) i izveštaja (*Reports*).

Kako su najvažniji objekti baze podataka tabele (*Tables*) i upiti (*Queries*), od njihovog kvalitetnog kreiranja zavise svi drugi objekti u bazi, pa ćemo tako i mi težište našeg rada na ovom Kursu baciti baš na ove teme. Dizajn obrazaca (*Forms*) i izveštaja (*Reports*) u *MS Access*-u traži mnogo više vremena, pa će se na Kursu raditi samo osnove dizajna ovih objekata, a detaljna obrada ovih tema je predviđena na *MS Access* (2).

Zašto, za projektovanje baza podataka, koristimo baš *MS Access*? Pored mnogih razloga vredi pomenući: jednostavnost rada i lakoću ostvarenja zamišljenog, standardno *Windows* grafičko okruženje, lako dostupan softver, mogućnost korišćenja starijih modela baza podataka radi njegovog daljeg razvoja,...

Tokom kursa predviđen je i praktični rad u okviru kog ćemo zajednički kreirati bazu podataka "Kadrovi" i kroz njen razvoj učiti jednu po jednu temu, dok ćemo upite savladati kroz urađene primere u bazi "Magaza" sa unetim podacima. Da biste sagledali sve aspekte, primetili razlike i otkrili greške, mnogi primeri će biti prikazani u neispravljenom (sirovom) obliku, što je naročito pogodno za provociranje grešaka i/ili korigovanje primera.

Od polovine kursa, pa sve do poslednjeg termina, vaš će zadatak biti da, na osnovu jednog jednostavnog (šematski prikazanog) relationalnog modela, kreirate bazu podataka po svojim idjama i prema svojim potrebama, učemu ćete imati punu podršku predavača. Na kraju Kursa, Vaš zadatak će biti da, unosom eksperimentalnih podataka, prezentirate kolegama u grupi – Vašu bazu podataka, pa da zajednički sagledamo moguće puteve njenog daljeg razvoja.

Oblast projektovanja relationalnih baza podataka, razvoj aplikacija i informacionih sistema pod *MS Access*-om, gde je potrebno šire znanje, ali i iskustvo u radu, se izučava na višim kursevima i to:

MS Access (2) - Osnovne tehnike razvoja aplikacija
gde se obrađuje optimizacija baza podataka, dizajn formi i izveštaja, i

MS Access (3) - Razvoj Informacionih Sistema,
gde se polaznici obučavaju za poslove projektovanja, razvoja i održavanja IS-a.

Po završetku ova (1) i (2) stepena edukacije i uz vaše permanentno zalaganje bićete sposobni da izvršite sve zadatke u oblasti konkretnе primene *MS Access*-a u projektovanju poslovnih baza podataka.

Svi potrebni primeri koji su ovde pomenuti mogu se preuzeti od autora Mikac@BitsYu.net ili 064/1168683.

1 OSNOVNI POJMOVI I STANDARD

Podatak - nosilac realne slike stvarnosti koji daje opis nekog činjeničnog stanja (žut; beo; ceo; 1; F14; 354; \$10,50; Car Dušan; itd.). To je, u stvari, činjenica o nekom segmentu realnog sveta koju treba čuvati u elektronskom obliku na duži vremenski period i koja ima značenje za korisnike baze podataka.

Informacija – obradeni podaci povezanih grupa podataka ili podatak koji proširuje nivo znanja, odnosno smanjuje entropiju sistema (neodređenost). Za informaciju je bitno da bude jasná tačna i pravovremena. Tako bi rečenica: "Danas je naša firma ostvarila promet u iznosu od \$5000 sa zaradom od \$320" bila informacija, ako nas obaveštava o nečem što nismo znali.

Baza podataka - kolekcija povezanih podataka sa višestrukom namenom. To je skup podataka koji se odnose na jednu temu ili namenu. Možda je lakše shvatiti bazu podataka kao skladište opšte namene za smeštaj i obradu bilo koje vrste informacija, prema vašoj želji. Na primer, u bazu podataka možete smestati imena i adrese. Takođe biste mogli da skladištite podatke o vašoj videoteci, fonoteci, zahama i porudžbinama vašeg privatnog preduzeća, kao i svih oblika evidentiranja i prikazivanja podataka o poslovanju.

Baza podataka je više od samih podataka. Ona sadrži objekte koji će vam pomoći pri upravljanju podacima, kao što su obrasci (za unos ili ažuriranje podataka) i izveštaji (za prikaz i/ili štampanje probranih grupa podataka u odgovarajućem obliku) kao i jedna ili više tabele u koje skladištite podatke koji su značajni za vas.

Sistem baze podataka – je celina koju čine:

- sama baza podataka, koja može predstavljati jednu datoteku, dok je to u kompleksnijim modelima najčešće hijerarhijski i struktorno organizovan sistem više povezanih datoteka
- sistem za upravljanje bazom podataka (Database Management System - DBMS) i
- korisnici baze podataka.

Na slici desno je predstavljen opšti model sistema.

MS Access je kompleksan skup više zavisnih alata za kreiranje i razvoj baza podataka, pa je teško reći da je to tek jedan program, ali se u svakom slučaju može svrstati u kategoriju visoko razvijenih DBMS razvojnih alata (programa).

DBMS je jedan složen softverski alat koji treba da omogući:

- **Skladištenje** podataka sa minimalnom redundansom (ponavljanje istih podataka) i njihovu centralizaciju, čime se postiže jednostavna i efikasna organizacija podataka.
- **Korišćenje** zajedničkih podataka od strane svih ovlašćenih korisnika, kao i podelu korisnika sistema na grupe prema pravima pristupa.
- **Logičku i fizičku nezavisnost programa** od podataka. Bez obzira što se podaci fizički pamte, po pravilu, samo jednom, u jedinstvenoj fizičkoj organizaciji, svaki korisnik dobija svoju sopstvenu logičku sliku (logička struktura) podataka kakva mu najviše odgovara.
- **Jednostavno korišćenje** baze podataka preko grafičkog interfejsa bliskog korisniku, čime se krajnji korisnici lakše uključuju u razvoj informacionog sistema (IS), pa time njegovo korišćenje uglavnom ne predstavlja nikakav problem. Licima koja učestvuju u razvoju ovakvog sistema značajno se povećava produktivnost, odnosno rezultati se postižu za relativno kratko vreme.
- **Najracionalnije iskorišćenje** zauzetog prostora na spoljnoj memorijskoj jedinici.

Informacioni sistem - je visokouređeni skup metoda, procesa, i operacija za prikupljanje, čuvanje, obradu, prenošenje i distribuciju podataka u okviru jedne organizacije, uključujući i opremu koja se u te svrhe koristi i ljudi koji se tim aktivnostima bave. Kao primjeri mogu poslužiti informacioni sistemi poslovnih banaka i svih većih trgovinskih preduzeća, spoljno-trgovinskih preduzeća, projektnih biroa, avio kompanija i sl.čno.



Opšta arhitektura sistema za upravljanje bazom podataka koja omogućuje ostvarivanje navedenih ciljeva sastoji se iz tri nivoa:

Interni fizički nivo definije način na koji su podaci fizički organizovani na spoljnim memorijskim jedinicama (diskovima).

Konceptualni nivo definije opštu logičku strukturu baze podataka, sve podatke u modelu i njihove logičke odnose (veze), treba da omogući upravljanje podacima kao zajedničkim resursom u sistemu baze podataka.

Eksterni nivo (korisnički) na njemu se definije logička struktura podataka pogodna za specifične zahteve, odnosno programe.

Iako su tabele primarni izvori podataka, korisnici baze podataka nemaju direktni pristup ovim objektima, već obradili podataka pristupaju preko određenih obrazaca ili formulara (*Forms*) za unos i ažuriranje podataka, koji se nalaze u bazama koje mogu da predstavljaju aplikacije pod Access-om. Grafičke mogućnosti rada pruža operativni sistem *Windows*, a nazivamo je grafički korisnički interfejs (GUI).

Artikal	JM	Kol.	Cena	IZNOS	T
Olovke kompleti	kom.	250	10,00	2.500,00	9
Sijalice	kom.	1000	5,00	5.000,00	20

Na slici predstavljen je jedan primer obrasca (*Form*) u grafičkom interfejsu, a koji se koristi za ažuriranje podataka o otpremnicama i njihovim stavkama.

Korisnici i osoblje koje učestvuje u projektovanju i razvoju baze može se podeliti u nekoliko grupa:

Izvršioci projektanti baze podataka, administrator baze podataka, sistem analitičari, aplikativni programeri, kao i nekoliko lica angažovanih od strane investitora i krajnjih korisnika sistema.

Podrška projektanti i programeri DBMS-a, inženjeri u razvoju zaduženi za softverske alate i osoblje zaduženo za razvoj softvera koje obuhvata sistem baze podataka, hardvera i

Krajnji korisnici stalni, povremeni i slučajni korisnici sa različitim stepenom iskustva i prioritetom u pristupu podacima.

Standard u projektovanju baza podataka

Da bi baza podataka bila smatrana standardnom mora da ispunjava najmanje sledeće uslove:

- da se tekstualni podaci unose prema kodnim stranama: 1250, 1251 ili 1252 (YU tastatura, latinična ili cirilična za naš region), koje ne prave razliku između velikih i malih slova;
- da se za format ispisa svih (numeričkih, valutnih, datumskih, vremenskih,...) podataka oslanja na podešavanja na nivou OS-a (u *Regional Settings*) i/ili korisničkim opcijama u aplikaciji), a u potpunosti mora da odgovara korisniku;
- da se za veličinu polja u tabelama uzimaju racionalne dužine, prema opsegu i veličini samog podatka u tom polju;
- da ima mrežnu orientaciju, zaštićen relacioni model i zahtevani referencijski integritet;
- da ima neki oblik standardnog sistema menija;
- da se lako administrira, može dalje razvijati i ima period korišćenja duži od jednog poslovnog perioda (uglavnom 6 ili 12 meseci);
- da, po isteku test perioda, ne zavisi od autora, odnosno da je može administrirati liceobučeno za takve poslove u firmi korisnika;
- da korisniku pruža na raspolaganje sistem pomoći, arhiviranje rezervne kopije;
- da je urađena po projektnoj dokumentaciji (u slučaju IS-a) i da su sve izmene u modelu baze- dokumentovane.

Poštovanje standarda u razvoju baze podataka je od izuzetnog značaja, a ono što bi bilo dovoljno da vas usmeri ka tom cilju je:

- jednostavnost korišćenja i lakša implementacija,
- olakšane naknadne izmene i usklađivanje po zahtevu i potrebama korisnika,
- bezbednost podataka i veliki uticaj korisnika na oblik (format) njihovog prikaza,
- totalna korisnička orientacija, što znači da korisnik baze podataka, odgovore na sva pitanja o korišćenju, dobija od same baze (aplikacije).

Pregled podataka po sektorima						
Sektor	Subsektor	Datum	Preduzivac	Kontakt	Cena	Predator
Биохемикални производи						
1D155	01.01.1994.	Бак		ID 1,00 Dn.	10,00 Dn.	
		Бак A		5 1,00 Dn.	5,00 Dn.	
		Бак B		5 1,00 Dn.	5,00 Dn.	
		Задатак		1 1,00 Dn.	1,00 Dn.	
		Приједавачи и професори		1 1,00 Dn.	1,00 Dn.	
+001*	31.12.1994.	Радничко друштво		1 1,00 Dn.	1,00 Dn.	
+001*	06.05.1994.	Србска агенција за стручни рад		5 1,00 Dn.	5,00 Dn.	
		Економичар		1 1,00 Dn.	1,00 Dn.	
+0011	04.04.1994.	Србска агенција за стручни рад		10 15,00 Dn.	150,00 Dn.	
+0011	31.06.1994.	Срб		1 1,00 Dn.	1,00 Dn.	
+001*	17.07.1994.	Радничко друштво		10 1,00 Dn.	10,00 Dn.	
		Поликарпова		100 1,00 Dn.	100,00 Dn.	
1D155	01.09.1994.	Бак		ID 1,00 Dn.	10,00 Dn.	
		Маркет за хемикалије		1 1,00 Dn.	1,00 Dn.	
		Спјалас		1 1,00 Dn.	1,00 Dn.	
		Консултант		100 1,00 Dn.	100,00 Dn.	
		Богдановић А		1 17,50 Dn.	17,50 Dn.	
		Стојановић Ј		100 1,00 Dn.	100,00 Dn.	
		Стојановић Ј		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Стојановић		1 1,00 Dn.	1,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 2,00 Dn.	2,00 Dn.	
+0011	10.11.1994.	Радничко друштво		5 1,00 Dn.	5,00 Dn.	
		Бак Б		1 21,15 Dn.	21,15 Dn.	
		Бак А		10 1,00 Dn.	10,00 Dn.	
		Срб СД агенција		6 1,00 Dn.	6,00 Dn.	
		Бадић		1 1,00 Dn.	1,00 Dn.	
		Бак		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Стојановић		1 1,00 Dn.	1,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 2,00 Dn.	2,00 Dn.	
+0011	01.03.1995.	Радничко друштво		1 10,00 Dn.	10,00 Dn.	
		Бак		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		5 1,00 Dn.	5,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 2,00 Dn.	2,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.	1,00 Dn.	
		Адиле радионице		1 19,00 Dn.	19,00 Dn.	
		Банкар		1 33,00 Dn.	33,00 Dn.	
		Стојановић		ID 1,00 Dn.	10,00 Dn.	
		Бак А		1 1,00 Dn.		

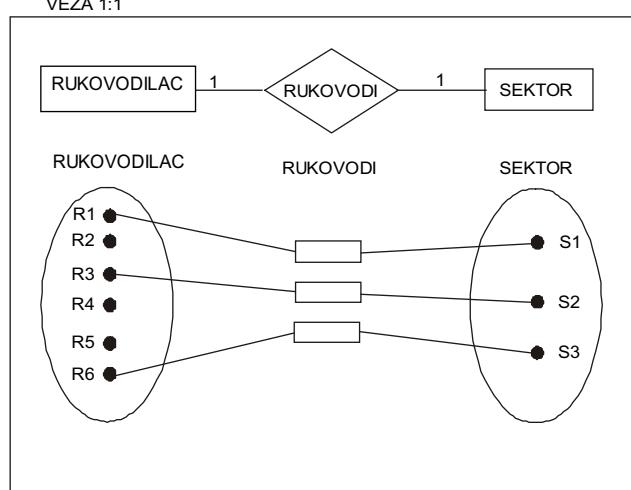
2 ENTITETI I VEZE MEĐU NJIMA

Entitet je element o kome se memorisu informacije. To je element koji postoji i koji se može razlikovati od ostalih elemenata. Entitet predstavlja bilo šta o čemu se mogu memorisati opisne informacije, što je u stanju da postoji nezavisno i što može biti jednoznačno određeno. Entitet može biti AUTOMOBIL, a njegovi atributi: marka, tip, boja, godina proizvodnje, i sl. Podaci u okviru baze nisu raspoređeni haotično, nego su grupisani po entitetima, što znači da svaki entitet sa sobom nosi određenu vrstu podataka.

Jedna baza se može sastojati od jednog ili više entiteta. Između entiteta mogu postojati određene veze koje su ilustrovane u sledećim primerima i mogu biti sledećih oblika:

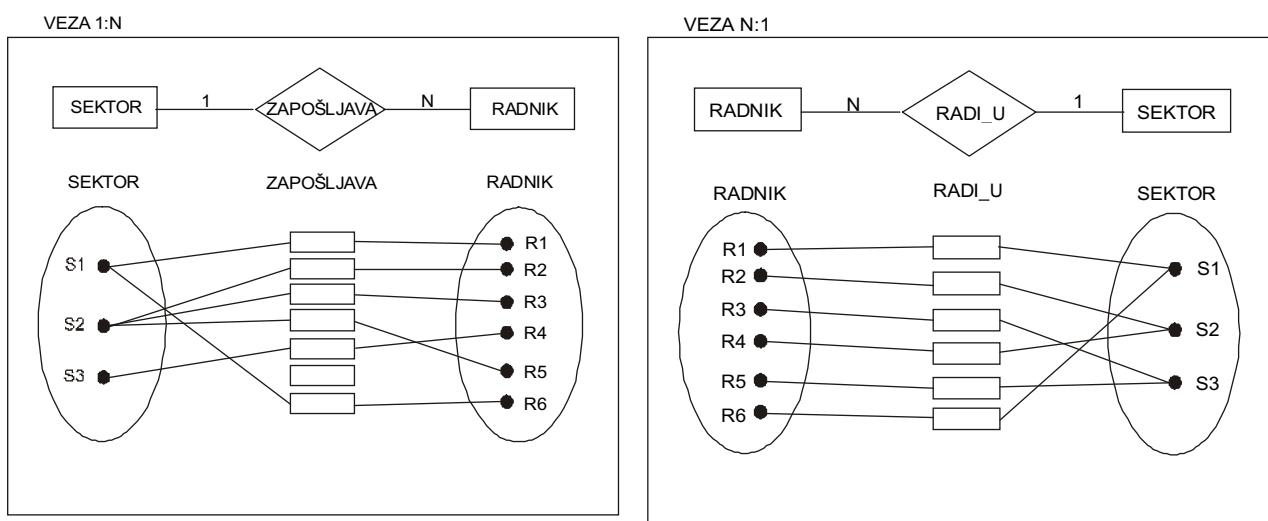
Veza 1:1

1:1 je najprostiji oblik veze, a za primer se može uzeti veza RUKOVODI između entiteta RUKOVODILAC i SEKTOR, gde se vidi da jedan rukovodilac može rukovoditi samo jednim sektorom, ali i obratno - jedan sektor ima samo jednog rukovodioca.



Veza 1:n (n:1)

je najčešći tip veze, a da li je veza 1:n ili n:1 zavisi od toga u kom smeru je posmatramo. Za primer ovog tipa veze se može uzeti veza SEKTOR ZAPOŠLJAVA RADNIKA, gde su entiteti SEKTOR i RADNIK u vezi 1:n, jer u jednom sektoru može biti zaposleno više radnika.

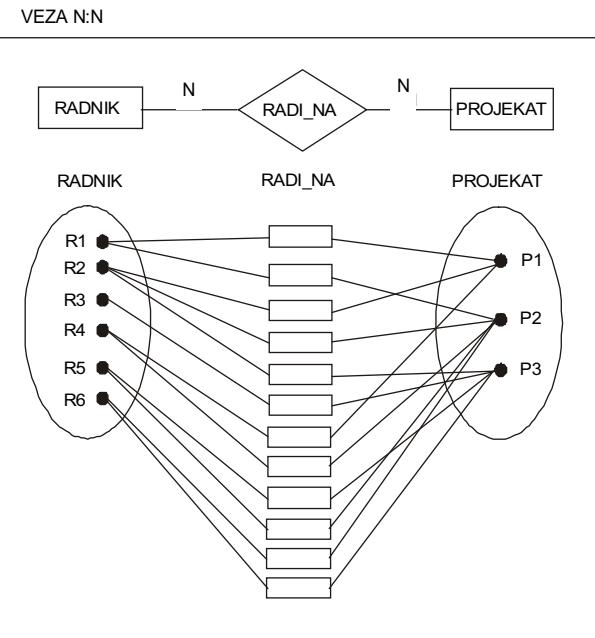


Posmatrajući ovu istu vezu u suprotnom smeru možemo učiti vezu RADNIK RADI_U SEKTORu, gde više radnika može raditi u jednom sektoru.

Veza n:n

je veza (više prema više) koja se u modelima vrločesto javlja, a ono što je za nju specifično je da takva kakva jeste - ne može da se direktno implementira u relacionom modelu baze podataka, jer bi dovela do nemogućnosti stroge definicije.

Problem veze n:n između dva entiteta se prevazilazi "razbijanjem" ove veze na dve veze tipa 1:n, pa se tako, prikazana veza RADNIK:PROJEKAT (n:n), deli na dve veze: PROJEKAT:P_R (1:n) i P_R:RADNIK (n:1), gde je P_R novi entitet (tabela u bazi) koji obezbeđuje strogu kontrolu modela.

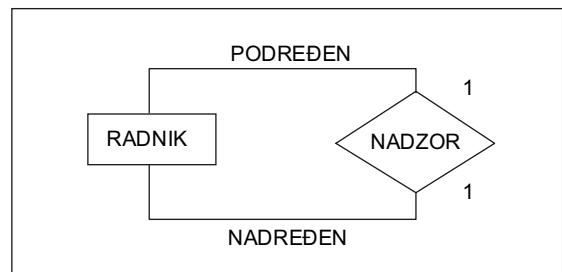


Rekurzivna veza

još jedan problematičan tip veze, a sreće se u situacijama kad se između dva (ili više) entiteta pojavi veza 1:n prema jedom paru atributa, a 1:1 ili n:1 prema drugom paru. Kao primer možemo uzeti vezu RADNIK:RUKOVODILAC.

Jedan radnik se, u tabeli RADNICI, može pojaviti samo jednom, dok se kod izbora nadzornika (rukovodioca) u tabeli RUKOVODIOCI, jedan radnik opet može pojaviti samo jednom, pa se uočava veza 1:1, ali pošto je jedan rukovodilac nadređen većem broju radnika, jasno je da postoji i veza RUKOVODIOCI:RADNICI = 1:n. Objedinjavajući ove dve veze u jednu relaciju - proizlazi da je ona tipa n:n.

REKURZIVNA VEZA

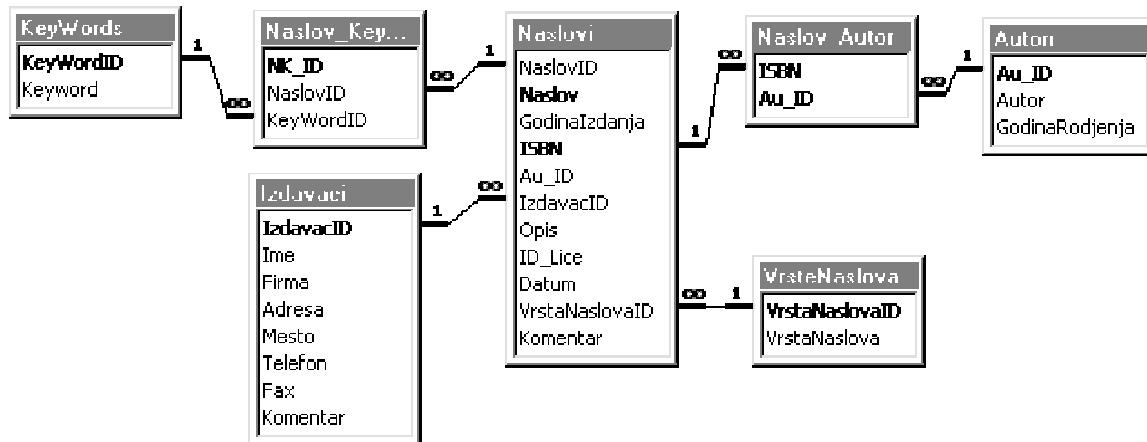


Ovaj tip veze je u bazi podataka moguće ostvariti, ali se model teško može održavati, pa se savetuje njeno obavezno eliminisanje, tj razbijanje na više entiteta sa vezama 1:n.

Primer povezanosti entiteta jedne baze podataka

Na sledećoj slici su predstavljeni entiteti baze podataka *Bibrios* (bibrios.mdb) i veze među njima. Na Web adresi <http://www.interbest.co.yu/source/access/> možete preuzeti (u ZIP formatu) neke od primera baza podataka koji su u skriptama izloženi, test pitanja, dodatke, index pojmove i slično, ili se obratite autoru na Mikac@BitsYu.net.

Entiteti u ovoj bazi su: Naslovi, Autori, Izdavaci, Naslov_Keywords, Naslov_Autor, Lica i VrsteNaslova.



Na slici se vidi da se izdavač u tabeli IZDAVACI može naći samo jednom, a u tabeli NASLOVI više puta. Naslov može imati jednog izdavača, a jedan izdavač se može naći u više naslova, pa je veza NASLOVI:IZDAVACI, preko polja (atributa) IzdavacID, tipa n:1 (oo:1).

Veza NASLOVI:AUTORI je tipa n:n (na istom naslovu se može naći više autora, a sam autor može učestvovati na više naslova), pa je dodatnom tabelom NASLOVI_AUTORI obezbjeđeno njen "razbijanje" na dve veze tipa 1:n.

Састав НАМИРНИЦА							
Групова	сироват	Активност/активност	сироват	Номер	сироват	Након исполнења	сироват
БУДВАР					ОПЕАНДА(Опек-коханео).....	2,00	X 0,0010
ЈАДЕ ВЕЛАНЦЕ	Испечени.....	20,00	Капијум.....0,0510	ОПЕАНДА(Опек-коханео).....	12,00	X 0,0010	
			Гарнју.....0,0730			02 0,0024	
						03 0,0034	
						04 0,0114	
						05 0,0115	
						06 0,0145	
						07 0,0170	
						08 0,0430	
						К 0,0530	
						ХОЛИН 0,2530	
						Б 0,3700	
ЈАЛЕЖКУЈАЦЕ	Испечени.....	20,00	Капијум.....0,0510	ОПЕАНДА(Опек-коханео).....	12,00	X 0,0010	
			Гарнју.....0,0730			02 0,0024	
						03 0,0034	
						04 0,0114	
						05 0,0115	
						06 0,0145	
						07 0,0170	
						08 0,0430	
						К 0,0530	
						ХОЛИН 0,2530	
ЈЕЧАЧ КОЉУШТ ГЕРШЛА	Глазенок.....1/2, Испечени.....10,00	СД (кохадо).....0,0250	ОПЕАНДА(Опек-коханео).....	2,00	0/5 0,0002		
			Капијум.....0,0430			С 0,0032	
			Гарнју.....0,0940			С 0,0032	
			Фордер.....0,0940			МОДИОЛ 0,1232	
			Гарнју.....0,0325			РАКА 0,1432	
			Макарон.....0,1130			ХОЛИН 1,5432	
			Макарон.....0,0318				
			Сунчеп.....0,0150				
						0,3307	
ЈАГОДА	Глазенок.....10,						
			Капијум.....0,1750				
			Капијум.....0,0910				
			Гарнју.....0,0430				
						0,2430	

Jun / 202

Страна 1 од 23



3 UVOD U MS ACCESS

Objekti MS Access-a

MS Access spada u grupu programa za upravljanje bazama podataka - DBMS (*Database Management System*). Kao što i samo ime kaže, ovi alati se koriste u projektovanju, razvoju i korišćenju baza podataka. Podaci koji se obrađuju mogu praktično biti sve što je, za korisnika baze podataka, prirodno i potrebno:

- Imena i adrese
- Informacije o zaposlenima i saradnicima
- Fakture, uplate, knjigovodstvo
- Poslovni kontakti, kupci, predviđanja prodaje
- Podaci o kretanjima i stanjima zalihami
- Planovi, rezervacije, projekti itd.

PRIMER: Pogledati kako su podaci organizovani u demo bazi MAGAZA (magaza.mdb u arhivi magaza.zip) ili neku od baza koje ste dobili uz instalaciju MS Access-a.

Jedna baza podataka u MS Access-u može da sadrži sledeće vrste objekata:

Tabela (*Table*) je osnovni tip objekta u bazi i predstavljaju direktni (primarni) izvor podataka. U tabelama se, po osmišljenim principima i vezama, čuvaju podaci kojima raspolaže korisnik i one su prvi objekti koje treba kreirati. Podaci u tabeli su smešteni u polja (kolone, eng. *Fields*), a sva definisana polja čine slog (zapis, red, eng. *Record*). Kvalitet baze podataka i IS-a leži u kvalitetnoj organizaciji podataka u tabelama baze, kao i njihovim dobrim vezama (relacionom modelu).

Upit (*Query*) je tip objekta za postavljanje pitanja o podacima iz tabela (ili drugih upita), a u cijlu njihovog ažuriranja kroz obrasce ili pregleda kroz izveštaje (na ekranu ili štampanu), pa se mogu definisati kao sredni izvori podataka. Na primer, upiti mogu da daju informacije (odgovore na pitanja) kao što su "koliko kupaca je iz Novog Sada, koja su njihova imena i brojevi telefona", s tom razlikom što vaš nabogavac govori srpski, a upit - jezikom SQL. Upiti se mogu koristiti za spajanje (engl. *join*) kolona iz više (relacija povezanih), tabela. Jedan upit bi mogao, na primer, da spoji tabele kupaca, porudžbina, stavki porudžbina i proizvoda da bi odgovorio na pitanje "koji kupci su naručili koje proizvode i koja je vrednost njihovih porudžbina". Upiti su, takođe, korisni prilikom izmene, brisanja ili unošenja velike količine podataka u jednom prolazu. Svaki korisnik MS Access-a bi trebalo da ga bar toliko poznaje da može bez problema iz njegove baze "izvući" potrebne podatke i to onako kako mu zada kolega ili nadgodavac. Svaki dalji rad u upravljanju bazom se zasniva na kreiranju objekata (forme izveštaji) koji ne mogu bez upita ili tabela, pa se njihovo kreiranje može znatno ubrzati samo pod uslovom da su izvori podataka (tabele i upiti) dobro kreirani.

Obrazac (*Form*) omogućava unos i prikazivanje podataka u prikladnom formatu koji lči na štampane obrasce u kojima treba popuniti prazna mesta. Vaši obrasci mogu biti jednostavni ili prilično složeni - sa grafikom, linijama, mogućnostima automatskog pretraživanja, koje unos podataka čine brzim i lakis. Obrasci mogu da sadrže i druge obrasce (nazvane pod-obrasci, engl. *subforms*) što omogućuje istovremeni unos podataka u više tabela.

Izveštaj (*Reports*) daje izvanredne mogućnosti pregleda i štampanja podataka. Kao i obrasci, izveštaji mogu da budu jednostavnici, ali i veoma kompleksni. Primeri izveštaja su: spiskovi, hronološki pregled poslovanja, cirkularna pisma, nalepnice sa adresama i fakture. Izveštaji se za podatke obraćaju tabelama, ali još češće upitima, a njihov osnovni zadatak je da te podatke predstave u obliku koji je lak za pregled, razumljiv i gde se mogu brzo učititi greške. Na primer, izveštajem se može dobiti "prodaja po mestima", "broj porudžbina u nekom periodu" i druge informacije koje se koriste za donošenje važnih poslovnih odluka. Primeri nekoliko izveštaja su prikazani kroz ovaskripta (na slobodnim prostorima).

Pages ili *Data Access Pages* (Web stranice za pristup podacima), omogućuju kreiranje složenih obrazaca na internetu kompanije koji su povezani s podacima u vašoj mreži. Čak i novajlja može da kreira Web stranice sa uvek ažurnim podacima. Hiperveze i vama i drugima omogućavaju pristup podacima koje ste objavili na Webu (i koje su drugi objavili), i to u formatu hipertekstualnih veza, direktno iz vaših MS Access obrazaca. Mnogi smatraju da posao objavljivanja podataka na Webu treba da bude prepusten isključivo administratoru Web lokacije (Webmasteru). MS Access 2000 je nepobitan dokaz da to nije tačno. Alatka *Data Access Pages* vodi vas kroz postupak izrade obrasca i njegovog povezivanja s podacima u izabranim objektima baze podataka, a zatim generisane HTML strane šalje na vašu Web lokaciju.

Makro (*Macro*) je niz MS Access komandi, a koristi se u slučajevima kada nekoliko komandi (određenim redosledom) treba pozvati na više mesta u MS Access aplikaciji (uglavnom iz forme). Kada pokrenete makro,

MS Access izvršava sve naredbe makroa u redosledu u kom su akcije navedene. Bez pisanja programskog koda, možete definisati makroe koji automatski otvaraju obrasce za bazu podataka, štampaju nalepnice sa adresama, obrađuju porudžbine i drugo. Makroi olakšavaju sastavljanje skupova tabela upita, obrazaca i izveštaja u kompletne aplikacije koje može da koristi bilo kočak i ako zna malo ili ne zna ništa o samom MS Access-u. Makroi se izbegavaju u profesionalnim MS Access aplikacijama, jer se oni ne mogu kompajlirati, pa ne daju mogućnosti kreiranja aplikacije.

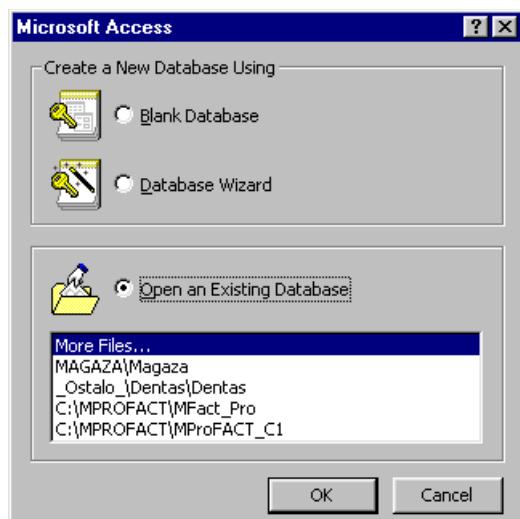
Modul (Module) kao i makro omogućava automatizaciju, ali mnogo napredniju, jer se u njemu, kao razvojna platforma, koristi VBA, u obliku procedura tipa *Sub* i *Function*, koje su na raspolaganju celom projektu. Za razliku od makroa moduli omogućavaju precizniju kontrolu nad preduzetim akcijama i zahtevaju škustvo u programiranju u VBA (Visual Basic for Applications), a ako uzmemu u obzir da forme i izveštaji imaju takođe modul kôda, jasno je da VBA predstavlja osnovnu programersku razvojnu platformu za MS Access. Na ovom kursu se nećemo baviti temama vezanim za VBA.

Kako su tabele izvori podataka, tako MS Access-fajl predstavlja bazu u slučaju da ima tabele, dok se u MS Access aplikacijama uglavnom nalaze ostali objekti, a veza sa tabelama se uspostavlja *Link-om*. Koncepcija odvajanja baze i aplikacije se obavezno primenjuje u profesionalnim projektima, a što sve treba uraditi i kakve se prednosti ovog odvajanja - radićemo na višem kursu MS Access (2).

OTVARANJE baze iz MS Accessa

Pokretanjem MS Access-a, (npr. na Start/Programs/Microsoft Access) dobîćete dijalog kao na slici i ponuđene sledeće opcije ulaza:

- *Create a New Database Using*
(kreiranje nove baze koristeći...)
 - *Blank Database*
(... praznu bazu)
 - *Database Wizard*
(automatsko generisanje baze)
- *Open an Existing Database*
(otvaranje postojeće baze podataka)



Database Wizard

Database Wizard pomaže početnicima da na lakši način uđu u rad na bazama i vide nekoliko primera aplikacija. Ovim se kreira cela baza za nekoliko minuta, a ponuđeno je više tipova baza podataka (adresari, porudžbine, vođenje inventara, videoteke, fonoteke...). Posle odabrane varijante, odlučujemo se o izboru tabela, i polja u tabelama, uključenju probnih podataka, izgledu aplikacije, izveštaja, i nazivu aplikacije. Sledi automatski proces kreiranja objekata i posle kraćeg vremena- baza je na raspolaganju, kao jednostavna aplikacija koja je jednostavna, ali vrlo korisna za učenje principa rada u MS Access-u.

Iako predstavlja dobar način da se vidi kako mogu izgledati aplikacije pod MS Access-om, ovaj način ipak nećete mnogo koristiti kod kreiranja svoje baze i aplikacije. Jedan od razloga je i njegovo pravilo da se imena polja u tabelama (koji se dalje prenose na sve ostale objekte) imenuju na engleskom jeziku.

Čarobnjaci u kreiranju svih drugih objekata, osim tabela, sigurno će vam mnogo znati u MS Access, pa ih od srca preporučujemo, a naročito ćemo ih koristiti kod kreiranja formi (obrazaca) i izveštaja. U toku rada u MS Access-u preporučuje se što više konsultovanja primera gotovih baza koje Microsoft isporučuje uz instalacionu verziju paketa (*Nwind, Orders, Solutions*)

Na prvi pogled

Radni prozor MS Access baze (Database Window) sadrži *CommandBar* (preko kojeg se otvaraju kartice željenih tipova objekata), kao i svoj *Toolbar*, sa nekoliko najčešćih komandi. U samom vrhu ovog prozora (naslovna traka) nalazi se ime baze koja se obrađuje.

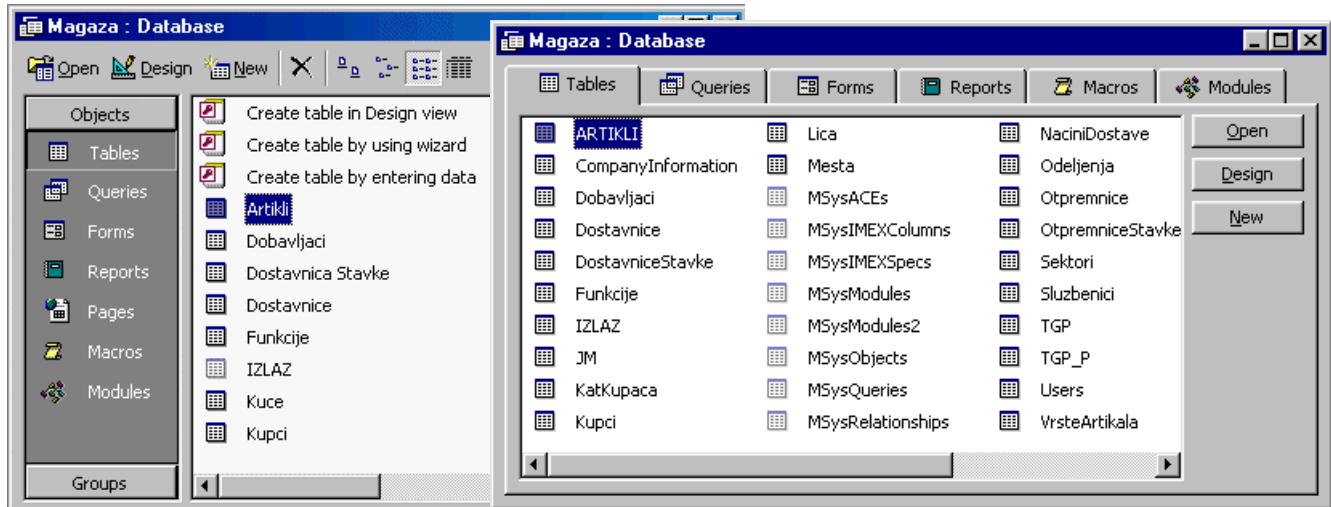
Sistem menija i paleta alatki u MS Access-u funkcioniše na sličan način kao i ostalim programima Microsoft Office-a, dok se sve ostalo smatra radnom površinom u kojoj se razvija i radi MS Access aplikacija. Na samom dnu MS Access prozora smeštena je statusna linija (*Status bar*) preko koje se korisniku prosleđuju dodatne informacije o tekućem objektu i informacije koje se odnose na tastere *Insert, Num Lock, Caps Lock, Scroll Lock*, ...

Database Window

Po otvaranju neke MS Access baze podataka na ekranu se pojavljuje *Database Window*. On omogućuje izbor jednog od šest (sedam) tipova objekata aktivne baze podataka, kao i režim (môd) rada u njima:

- Open** - otvaranje postojećeg,
- Design** - redizajniranje postojećeg, odnosno,
- New** - kreiranje novog objekta).

Kod izveštaja opcija *Open* je zamenjena odgovarajućom opcijom *Preview* (kojom se on otvara za pogled), a kod makroa i modula opcijom *Run* (kojom se on pokreće).



Desnim klikom na objekat možete dobiti sve moduse rada sa datim objektom, pa ovaj n in treba maksimalno koristiti kod otvaranja objekata za bilo kakvu obradu. Ne treba zanemariti ni konice na *Toolbar-u*, a posebno u situaciji kada treba preći iz jednog u drugu modus obrade objekta, a *Database Window* se nije vidljiv. *Toolbar* daje još jednu prednost – preko ikonice možete uvek preći u prepostavljeni (suprotan) modus (npr. iz *Open* u *Design* režim rada). Na prethodnim slikama je prikazan *Database Window* u MS Access 2000 i MS Access 97.

Sistem menija

File meni, osim standardnih stavki kao kod svih Windows aplikacija, sadrži i stavke:

- *Get External Data* omogućuje uvoz (*Import*) ili povezivanje (*Link Tables*) tabela iz druge MS Access baze ili nekog drugog kompatibilnog izvora podataka. Koristeći isti dijalog može se uvesti i bilo koji objekat iz druge MS Access baze. *Link*-om povezana MS Access tabela ne može se otvarati u dizajn modu (u tom m du joj se može pristupiti samo otvaranjem matčne baze (one u kojoj se tabela nalazi)).
- *Save As/ Export* omogućava čuvanje nekog od MS Access objekata, pod drugim imenom u teku oj ili drugoj bazi, kao i u obliku tabele u formatu druge aplikacije.
- *Database properties* je spisak svojstava MS Access baze podataka. Može da sadrži informacije o autoru, nazivu, predmetu, ključnim rečima aplikacije, statistike i spisak objekata u bazi, kao i mogu nost kreiranja korisnički definisanih (*Custom*) svojstava baze.

Edit meni sadrži standardne stavke za pokretanje komandi *Ažuriranje* (*Cut, Copy, Paste, Undo, Delete, Rename*) za rad sa objektima iz baze, kao i mogu nost kreiranja pre ice (*Shortcut-a*) na dati objekat. Za sve ove alate na raspolaganju su i odgovaraju e stavke *PopUp* menija koji dobijate desnim klikom na objekat, pa ih koristite za brzi i kvalitetniji rad, jer je MS Access, kao i OS Windows predstavlja objektno orijentisano radno okru enje.

View meni sadrži stavke izbora tipa objekta u *database* prozoru, određivanje tipa ikona i njihov raspored, kao i pregled svojstava (*Properties*) objekta i programskega koda objekta (forme i moduli). Preko stavke *Toolbars* omogu en je pristup korisničkom podešavanju sistema menija i paleta alata.

Insert meni sadrži opcije za kreiranje novog objekta u bazi, kao i automatsko generisanje forme ili izveštaja na osnovu selektovane tabele ili upita (kao izvora podataka).

Tools je daleko najznačajniji meni, a u svom prvom delu sadrži poznate *Microsoft Office* alate *Spelling* i *AutoCorrect*. *Spelling* izvršava pravopisnu proveru teksta (za srpski je potrebno instalirati poseban dodatak), dok *AutoCorrect* automatski ispravlja neke karakteristične greške (po želji). Korišćenje *AutoCorrect*-a je slično na nivou celog *Microsoft Office*-a, pa ga ovde nećemo posebno naglašavati. *Database Utilities* sekcija ovog menija, nudi mogućnost konvertovanja (*Convert*) fajla iz ranijih verzija *MS Access*-a u fajl novog formata, a (u verziji *MS Access 2000*) baza se može konvertovati i u fajl starijeg formata (za *MS Access 97*), mada ne funkcioniše korektno u slučaju aplikacije, ipak, predstavlja alat od izuzetnog značaja. *Compact* obezbeđuje sažimanje baze (kao i opravke fajla posle neregularnih izlazaka iz aplikacije - *Repair*). Sažimanje se može uporediti sa defragmentacijom diska s tim da se odnosi na *MS Access*-fajl. Radi se o sledećem: u toku rada na dizajnu baze, kao i u slučajevima većih zahvata na brisanju i dodavanju slogova, obrisani objekti izazivaju fragmentaciju fajla (samim tim on nepotrebno zauzima više prostora na disku), te ga sažimanje svodi na optimalnu veličinu. Ovo se preporučuje kod velikih izmena u fajlu. *Repair* se automatski startuje ako *MS Access* otkrije da nešto nije u redu prilikom otvaranja baze, dok je u verziji *MS Access 2000* ovaj alat integriran uz *Compact*. Ako se baza ponaša nepredvidljivo, ovaj alat se uvek može startovati i manuelno iz menija, a daje dobre rezultate samo u slučaju dobrog relacionog modela (sa što manje propusta u optimizaciji) i uključenog referencijalnog integriteta u što većem broju relacija.

Window meni sadrži stavke za aranžiranje prozora na ekranu (objekti koji se obrađuju).

Help meni ima uobičajene alate za pomoć.

4 TABELA I NJENA ORGANIZACIJA

Tabela, polje, zapis

Da bi obrada informacija u okviru neke aplikacije (unašem slučaju *MS Access*) bila moguća, moramo je prethodno raspodeliti na podatke, koje ćemo zatim smestiti u neku tabelu koja sadrži redove i kolone.

Tabela sadrži podatke o određenom subjektu. Na primer, baza podataka može da sadrži jednu tabelu sa podacima o kupcima (njihova imena, adrese, telefone...), drugu tabelu za proizvode i neku treću za porudžbine.

Polje predstavlja jedinični podatak, kao što je ime osobe, ili naziv ulice, poštanski broj ili naziv proizvoda. Polje se obično pojavljuje kao jedna kolona u tabeli.

Zapis sačinjavaju svi podaci pojedinačne stavke tabele, kao što su svi podaci o jednom kupcu iz tabele klijenta, ili o jednom proizvodu iz tabele proizvoda. Zapis se, u vizuelno predstavlja kao red. Zapisi se često nazivaju sloganima ili rekordima (*Records*) ili n-torkama, gde je n-broj polja u sloganu.

Atribut, domen i relacija

Atribut element informacije kojim je jednoznačno uređena vrsta svojstva.

Domen skup svih vrsta vrednosti koje određeni atribut može primiti.

Relacija skup uređenih n-torki.

Relacija ima sledeća svojstva:

- šema relacije ne sadrži dva jednakata naziva atributa,
- redosled kolona u relaciji nije bitan,
- relacija ne sadrži dve jednakate n-torce,
- redosled n-torki u relaciji nije bitan.

Stepen relacije broj uređenih n-torki.

Relaciona šema konačan skup naziva atributa zajedno sa skupom ograničenja.

Kreiranje tabela Wizard-om

MS Access, pored mogućnosti kreiranja baze podataka *Wizard-om* (čarobnjak), pruža na raspolaganje i *Wizard* za kreiranje tipskih tabela podataka. U fazi savladavanja i istraživanja u oblasti projektovanja baza podataka sigurno je da će korišćenje ovog alata doprineti bržem uočavanju mogućnosti *MS Access*, ali se u profesionalnom radu nameće potreba za kreiranjem sopstvenih tipskih tabela.

To znači da će se u svom daljem radu u *MS Access*-u sve više oslanjati na sopstvene snage, a manje na automatizme, bar kod upravljanja tabelama i upitima. Čarobnjaci kod izrade formi i izveštaja su nezaobilazni na svim nivoima poznavanja *MS Access*.

Indexiranje i ključevi

Indexiranjem se omogućava da se, u daljem korišćenju tabele kao izvora podataka, može brže sortirati, pretraživati i bravljivati u svakom drugom obliku – po polju podataka koje je indeksirano. Indeksiranje se preporučuje za polja numeričkih podataka i to samo u situacijama kada se nad njima planira neki oblik pretraživanja. Uključivanje indeksiranja po polju podatka može dovesti do znatnog uvećanja fajla baze podataka, a dešava se obično kod tabela koje imaju veliki broj sloganova podataka (u desetinama hiljada i više).

Isključivanje indeksiranja može prouzrokovati spor odziv upita kod kojih je primenjen neki od oblika sortiranja i kaskadnog grupisanja, pa je na programeru da odabere pravi način u konkretnom slučaju. Kada je po datom polju uključen primarni ključ – update podataka je nešto sporiji, ali se zato svaki upit po ovom polju drastično ubrzava.

Ovde od presudnog značaja može biti iskustvo kolega, pa ih, u svakom slučaju, saslušajte.

Kardinalnost - broj n-torki ili sloganova (predstavljeni po kolonama) jedne relacije (ili tabele).

Polje primarnog ključa - Ovo polje jednoznačno identificuje svaki zapis u tabeli. Na osnovu sadržaja ovog polja može se tačno odrediti o kom zapisu se radi. Iz toga sledi da je primarni ključ, u stvari, atribut neke tabele (relacije) koji identificuje jedan slog (n-torku) i naziva se još ključem relacije.

Prost ključ - kada u jednoj tabeli postoji polje koje u potpunosti može odrediti jedinstvenost sloga, onda se takav ključ naziva prostim.

Entitet RADNIK sadrži atribute RadnikID, Prezime, Ime, Zanimanje i MB.

RADNICI

RadnikID	Prezime	Ime	Zanimanje	MB
001	Petrović	Petar	novinar	1011963325641
002	Stojanović	Aleksandar	elektrotehničar	0209948025632
003	Slavković	Svetlana	prevodilac	2612975361258
004	Topolac	Životije	profesor	1809958521438
005	Stojanović	Aleksandar	geodeta	1111197211121

Ovde se vidi kako polje primarnog ključa (*RadnikID*) jednoznačno određuje slog; zamislimo da je kolona zanimanje u nekom upitu izostavljena. U tom slučaju, na bazi vrednosti polja *RadnikID* možemo zaključiti o kom Stojanović Aleksandru se radi, tj. koje je njegovo zanimanje, odnosno da li je u pitanju Stojanović Aleksandar- elektrotehničar ili neko drugi sa istim imenom i prezimenom ko je po zanimanju geodeta.

Složen ključ - Kada se jedinstvenost sloga ne može u potpunosti odrediti preko jednog polja, tada se mora koristiti slobzen primarni ključ. Ako u tabeli RADNICI imamo samo polja Prezime i Ime, tada bismo morali uvesti i polje SrednjeSlovo jer, kao što vidimo Prezime i Ime ne mogu u potpunosti odrediti jedinstvenost sloga (jer se mogu pojaviti dva radnika sa istim imenom i prezimenom). Tada primarni ključ čine zajedno Prezime, Ime i SrednjeSlovo, što predstavlja složeni primarni ključ, jer ga čini više polja.

Na primer, ako relaciji RADNICI dodamo atribut MB (matični broj), tada je i ovaj atribut kandidat za ključ, odnosno pomocu njega se takođe može odrediti jedinstvenost n-torke. Obično se, u ovakvim situacijama, jedan atribut bira za primarni ključ, a ostali neizabrani kandidati se nazivaju *alternativnim ključevima*.

Spoljni ključ - Pored primarnog ključa, relacija može imati i spoljni ključ. O čemu se zapravo radi?! Ponekad se dešava da jedna relacija ne može egzistirati bez druge, na primer stavka fakture ne može da postoji bez fakture. Recimo da relacija FAKTURE ima ovaj oblik:

Fakture (BrojFakture, Datum, Partner).

BrojFakture je primarni ključ jer određuje jedinstvenost sloga. Fakturna ima svoje stavke, pa iz toga proizilazi da relacija FaktureStavke ima ovakav oblik:

FaktureStavke (BrojFakture, BrojStavke, Artikal, JedinicaMere, Količina).

Kako stavka bilo koje fakture ne može da postoji bez fakture kojoj pripada, tako ni relacija *FaktureStavke* ne može da egzistira bez relacije *Fakture*. Ove dve relacije povezane su preko ključa *BrojFakture* koji je zajednički za obe relacije. Na osnovu njega se zna koja stavka pripada kojoj fakturi. Zato se kaže da, u ovom slučaju relacija *FaktureStavke* nasleđuje atribut *BrojFakture* od relacije *Fakture* i on je za relaciju *BrojFakture* - spoljni ključ. Spoljni ključ služi za uspostavljanje relacija između tabela u relacionim bazama podataka.

VEŽBA: Kreirajte tabelu KUPCI(Naziv, Lice, Adresa, Mesto, Telefon). Složen ključ čine polja Naziv i Mesto, jer se u istom mestu (gradu) ne sme pojaviti kupac sa istim nazivom firme, ali se dozvoljava da ista firma postoji sa istim imenom u više mesta.

5 OSNOVE UPRAVLJANJA TABELAMA

Tabela predstavlja kolekciju podataka određene tematike i primarni izvor podataka za bazu. Sastoji se iz slogova (redova, eng. records) i polja (fields). Ona je osnovni objekat baze podataka. Podaci kojima se fizički manipuliše jedino postoje u tabelama, dok su ostali objekti baze samo način da se podaci iz baze, izabrani po nekom kriterijumu, sortirani i grupisani (po potrebi), prikažu na određeni način. Zato je kreiranje i definisanje karakteristika tabela, kao i uspostavljanje relacija između njih, jedan od najvažnijih zadataka prilikom projektovanja baze podataka.

ID_Firmi	Maticni	Naziv Fime	Adresa	ID_Mesto	Telefon
2168	601233	POTORIŠTE NA TERAZIJAMA	TERAZIJE 29	BEograd	011/334-037,338-907
2169	601101	POŠTANSKA ŠTEDIONICA	27. MARTA 71	BEograd	011/3248-111,3248-6110,32
2170	601104	PPT INŽENJERING, D.D.	BULEVAR VOJVODE MIŠIĆA 37-3	BEograd	011/235-1650,235-1890
2171	600801	PRAG, HOTELSKO UGOSTITELJSKO I TUR. PRE.	NARODNOG FRONTA 27	BEograd	011/687-355
2172	601231	PRAKTIČNA ŽENA, NIP	NEMANJINA 4	BEograd	011/641-155/631-653
2173	600118	PRAM, PP	ALEKSE NENADOVIĆA 32	BEograd	011/3220-188,459-406
2174	601201	PRAVNI FAKULTET UNIVERZITETA U BEOGRADU	BULEVAR REVOLUCIJE 67	BEograd	011/341-501
2175	601103	PRED. GRADSKIH PIJACA I TRŽNJIH CENTARA	ŽIVKA KARABIBEROVIĆA 3	BEograd	011/422-566
2176	601104	PREDNAPREGNUTI BETON, PRED.ZA PROJEKTOV.	UČITELJSKA 60/I	BEograd	011/4886-810,4887-004
2177	601103	PREDUZEĆE PLAVLJANAKA PLAV	VELIZARA KOŠANOVICIĆA 2	BEograd	011/413-858/410-054
2178	601109	PRENAD,D.O.O	DOSITEJEVA 1/III	BEograd	011/626-569,181-108
2179	600713	PРЕСЕДАН М, PP	MIŠKA KRALJICA 7	BEograd	011/583-346,582-493
2180	601231	PRESS EXPRES (DIREKTOR MITAR DANILOVIĆ)	ŠUMADIJSKI TRG 6A PF2024	BEograd	011/556-598,559-845
2181	600605	PREVOZ, PLEVULJA, PREDSTAVNIŠTVO	BALKANSKA 48	BEograd	011/687-164
2182	600134	PRIMA GRAFIKA	SARAJEVSKA 1	BEograd	011/642-666,339-422
2183	600501	PRIMA KOMERC	TADEUŠA KOŠČUŠKOG 70	BEograd	011/187-333,187-496,637
2184	600716	PRIMA MEDICAL, CO	CARA DUŠANA 72	BEograd	011/639-082,187-496,637
2185	600902	PRIMA, OMLADINSKA ZADRUGA	VLAJKOVIĆEVA 9	BEograd	011/347-516
2186	600720	PRIMAR EXPORT-IMPORT D.O.O.	CARA LAZARA 5	BEograd	011/628-653,627-345
2187	600716	PRIMAX, VELEDROGERIJE I APOTEKE	NEMANJINA 2	BEograd	011/644-968
2188	600702	PRIMERA	TEODORA MIRJEVSKOG 64	BEograd	011/444-7834,457-542,411
2189	601231	PRINCIP, IP	KOLARČEVA 7/6	BEograd	011/3248-496
2190	601102	PRIREDIVANJE IGARA NA SREĆU DP	SVETOZARA MARKOVIĆA 40	BEograd	CENTR. 011/334-876
2001	601231	PROMETNA ŠTAMPA, UZETAK, LEPOTA, ETC.	NUGOŠEĆKA 54	BEograd	011/334-876

Nije neophodno da svi podaci, koje planirate da vodite u bazi podataka, budu u jednoj tabeli. Obradom postojećih uvek se mogu sračunati potrebni podaci (i dobiti informacije), a razvrstavanjem po tabelama možete postići veće efekte u funkcionalnosti i lakov održavanju tabela, a samim tim i baze podataka.

Kreiranje tabela

Početni korak za kreiranje tabele je komanda *New* u *database* prozoru (kada je aktivna kartica *Tables*). Ovim se otvara dijalog koji nudi pet načina u kreiranju tabele i to:

Datasheet View omogućava direktno unošenje podataka u tabelu, pri čemu program sam određuje tip polja. Ime polja se daje opcijom *Rename Column* iz menija koji se dobija desnim klikom na naslov kolone. Ovaj način se koristi u dva slučaja i to: a) kada još nije poznata struktura tabele i b) kada je potrebno ubaciti kopirane podatke iz nekog drugog programa (npr. MS Excel). MS Access će svim poljima dati imena kao Field1, Field2,..., a na osnovu sadržaja unetih po poljima, pokušati da odredi neka svojstva polja, kao što je maksimalna dužina i format podatka. Ovde se podrazumeva da su sva polja tekstualnog formata i opcionalno unete dužine u samom MS Access-u. Ovaj pristup u kreiranju tabele treba izbegavati, jer se njime skoro ništa ne definiše.

Design View vodi u standardno kreiranje tabele detaljnim izborom svojstava polja podataka, kao što su imena polja, tip podataka i veličina polja, određivanje primarnog ključa, indeksiranje, graničnih uslova za unos podataka i slično. Poznavanje ovog pristupa je najvećeg značaja, a ujedno je to najkompleksniji, najčešći i najdetaljniji pristup.

Table Wizard prepušta formiranje tabela Wizard-u, koji to čini pozivajući se na biblioteku već formiranih tabela. Ova mogućnost je korisna kada želimo da kreiramo neke manjeviše standardne objekte kao što su adresari, podsetnici, računi, izveštaji, fakture, evidencije troškova i sl. Ovaj pristup je koristan u učenju i savladavanju "tajni" MS Access-a, ali je njegov nedostatak u tome što nudi mnogo toga, a vi najčešće ne možete da pronađete baš ono što vam treba.

Import Table daje mogućnost uvoza tabele iz nekog drugog izvora podataka kojije kreiran prema nekom od standarda za datoteke i baze podataka. Može biti rеч o MS Access bazi, ali i mnogo drugih organizovanih tabela, kreiranih programima kao što su: MS Excel, dBASE, Clipper, FoxPro,..., kao i fajlove tekstualnog formata, gde je za podelu na kolone (polja) potrebno definisati širinu svake kolone pojedinačno. Treba imati na umu da se na ovaj način stvara kopija importovane

tabele, pa se u slučaju rada u lokalnoj mreži ili IS-u može desiti decentralizacija (rasipanje) podataka, odnosno ponavljanje istih podataka na više mesta, što za održavanje takvog sistema može biti fatalno.

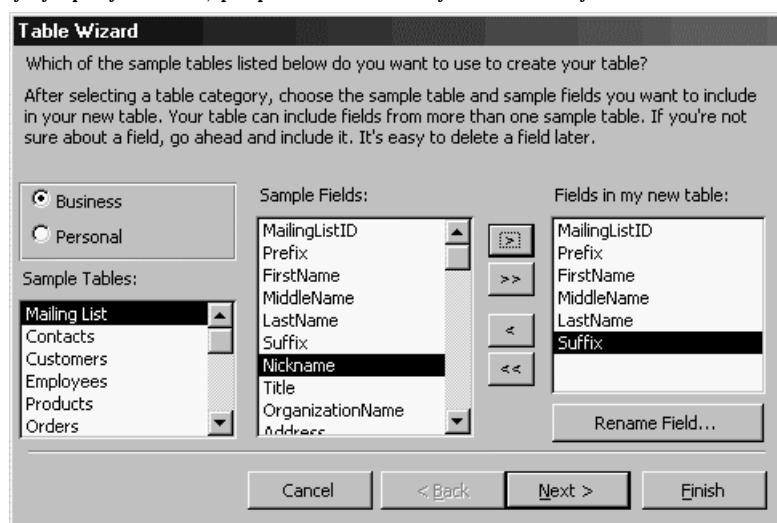
Link Table daje mogućnost povezivanja sa tabelom (obično više njih) iz druge MS Access baze (retko kada se koristi neki drugi format baze), pri čemu tabela ostaje u matičnoj bazi, a linkom se omogućuje veza na nju u cilju pregleda i ažuriranja podataka. U ovoj situaciji, pristup dizajn tabeli nije moguć iz baze koja ima link na datu tabelu, već samo iz matične (izvorne) baze. Dizajn tabele se ne može pristupiti dok se svi aktivni linkovi na odnosnu tabelu ne isključe, što znači da pri dizajnu tabele nikako (kroz mrežu) ne može da je ažurira. Kako mrežne aplikacije uglavnom rade na više tabele i više baza, u ovoj situaciji je potrebno uputiti korisnike da privremeno ne otvaraju podatke koji se nalaze u ovoj tabeli (jer će biti upozoren da to trenutno nije izvodljivo), a posle izvršenih ispravaka u dizajnu – nastavljaju sa uobičajenim radom. Ujedno ovo predstavlja i jedinu situaciju u kojoj, korisnik baze podataka kroz računarsku mrežu, sme da bude ometan u njenom korišćenju. Mnogi će vam priznati da bez Link-a nema ni mrežne orientacije baze podataka i da je to tek kamen temeljac za dobar informacioni sistem.

U kompleksnijim bazama podataka se obavezno praktikuje da nisve potrebne tabele ne budu u odnosnoj bazi, jer MS Access omogućava njihovo povezivanje (*Link*) kroz baze. Ovo je od ključnog značaja za projektovanje aplikacija za korišćenje u lokalnoj mreži ili informacionom sistemu. Održavanje i razvoj ovakvih sistemaja mnogo lakše za programera, a korisniku daje mogućnost kontinuiranog rada bez obzira na zahvate na administriranju.

Table Wizard

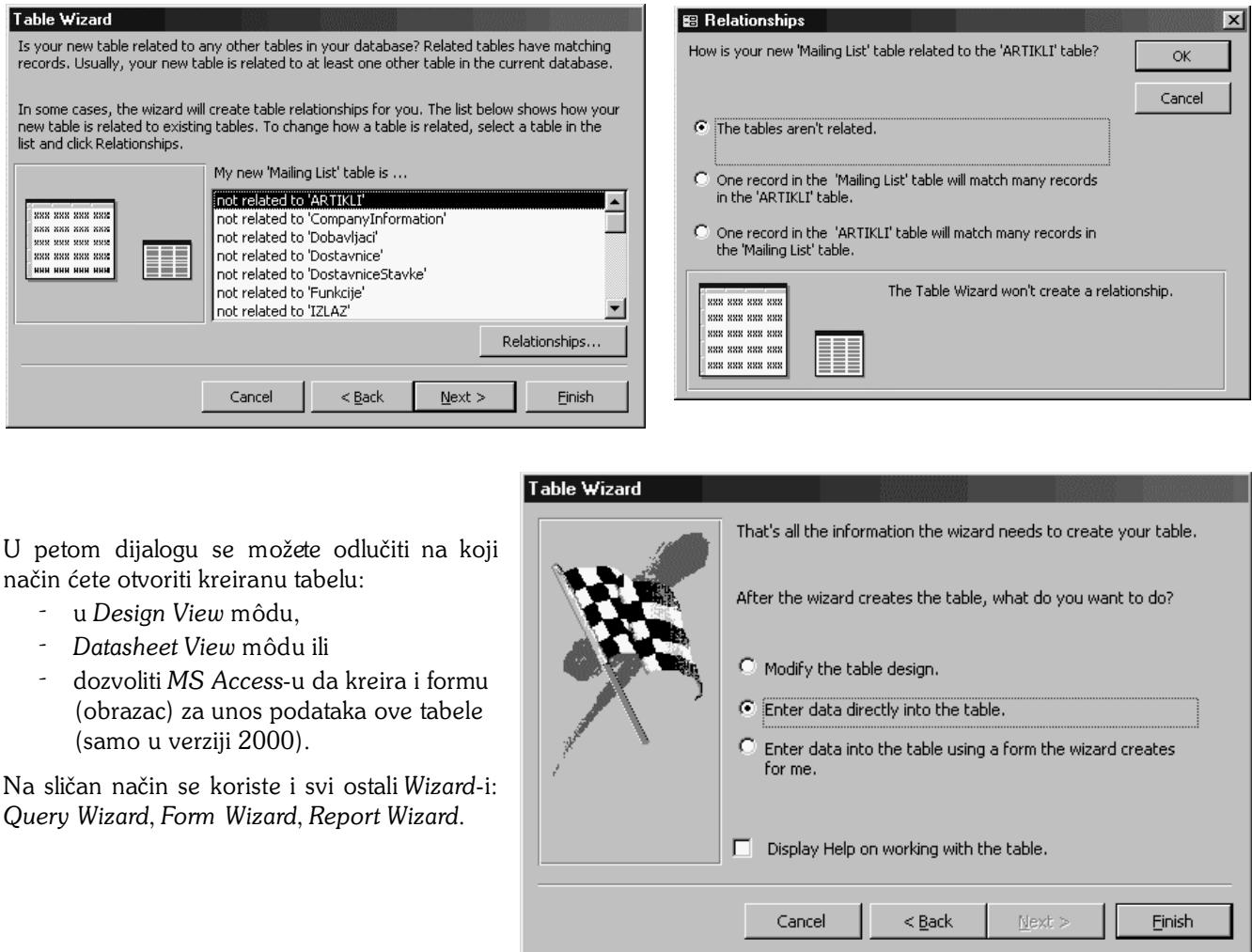
se koristi u fazi učenja, pa bi bilo dobro da ovde prikažemo njegovo korišćenje. Komercijalne (poslovne) baze podataka, koje se eksplorisu i po 24 na dan, moraju se ozbiljnije projektovati, pa prilikom kreiranja tabele u njima nećete moći (a ni želeti) da koristite *Wizard*-e. Tabele i upiti su tako značajni, objekti baze podataka, da traže više poštovanja, vremena i znanja, od pukog *Wizard*-a, jer se korišćenjem raspoloživih automatizama programer "otuđuje" od svoga dela.

U prvom dijalogu *Table Wizard*-a izaberite tip tabele (poslovna ili personalna), zatim neku od predloženih tabela, polja iz izabrane tabele i ... kreirali ste listu polja vaše nove tabele (poslednja lista), a preko komandnog tastera "Rename Field" možete promeniti ime izabranih polja.



U drugom dijalogu *Table Wizard* vam nudi da unesete ime tabele i (opciono) deklarišete polje primarnog ključa.

U trećem dijalogu *Table Wizard*-a možete pregledati relacije kreirane tabele sa drugim tabelama u bazi, a potom preko tastera "Relationships.." (eventualno) možete zadati parametre neke od relacija (kao u prikazanom četvrtom dijalogu).



U petom dijalu se možete odlučiti na koji način otvoriti kreiranu tabelu:

- u *Design View* módru,
- *Datasheet View* módru ili
- dozvoliti MS Access-u da kreira i formu (obrazac) za unos podataka ove tabele (samo u verziji 2000).

Na sličan način se koriste i svi ostali *Wizard*-i: *Query Wizard*, *Form Wizard*, *Report Wizard*.

Dizajn tabele

Način na koji se najčešće dizajniraju tabele je *Design View*. Tu je omogućen detaljan opis i definicija svakog polja nove tabele, kao i kasnije modifikacije postojeće tabele. U gornjem panelu otvorenog prozora upisuje se:

Field Name ime polja, kojim se definiše ime atributa u tabeli. Iz mnogih razloga (koje ovde nećemo pominiti) poštujte sledeća pravila kod zadavanja imena polja:

- a) koristite što kraće ime uz kombinovanje velikih i malih slova,
- b) ne koristite specijalne znake osim "_",
- c) ne koristite YU slova.

Iako, i bez ovih pravila, sve može da funkcioniše u lokalnom okruženju (vaš MS Access i vaš PC i vaša lokalna mreža) ipak kod distribucije (publikovanja) baze podataka ili njenog povezivanja sa nekom aplikacijom u razvoju (npr. pod VisualBasic-om ili operativni sistem UNIX) možete imati gde-nih problema. Ista pravila treba koristiti i kod imenovanja fajla baze i svih objekata u bazi. Ovo ne treba mešati sa unosom podataka i komentara, gde se može koristitibilo koje pismo ili jezik (omogućen operativnim sistemom).

Data Type tip podatka, koji se čuva u tom polju je najvažnije svojstvo polja, pa mu posvetite veliku pažnju. Ovde je važno što više izbegavati *Text* tip, jer se njegovim korišćenjem najmanje štedi prostor, a baza gubi na performansama (zauzeće, brzina rada, korisnčka orientacija,...). Ponekada ni sam korisnik to neće primetiti dok ne vidi i drugu situaciju. Primer za ovo može biti poštanski broj mesta, koji treba obavezno definisati kao broj (*Number*), a ne kao tekst (*Text*).

Description tekstualni opis podatka dužine do 255 karaktera, a njegovo upisivanje se savetuje, jer se on, u toku rada korisnika na datom polju podatka, ispisuje u statusnoj liniji i korisniku pruža dodatne informacije o tom polju.

MS Access podržava sledeće tipove podataka (*Data Type*), a u dodatku su dati i opsezi za numeričke tipove:

Text tekst (slova, brojevi i svi specijalni znaci) sa kojim se ne vrše računske operacije, a dužine do 255 karaktera.

Memo dugačak tekst (slova, brojevi i svi specijalni znaci) dužine do 65535 karaktera. Ovakav tip tekstualnog podatka ne može ući u neko sortiranje, pretraživanje ili grupisanje, što treba imati u vidu.

Number brojevi koji se upotrebljavaju u aritmetičkim izrazima, a veličina polja (*Field Size*) zavisi od "podtipa" koji može biti: *Byte*, *Integer*, *Long Integer*, *Single*, *Double*, *Replication ID*, *Date/Time*, *Currency*, *Auto Number* (koji može biti *Long Integer* ili *Replication ID* opsega) ili *Yes/No*.

OLE Object Objekat kao MS Word dokument, MS Excel tabela, zvuk, grafikon, slika... povezan je (*Linked*) ili ugrađen (*embedded*) u MS Access tabelu.

Lookup Wizard daje mogućnost da se kreiraju polja koja omogućavaju izbor vrednosti iz druge tabele ili iz (ograničene) liste vrednosti upotrebom grafičkih kontrola *ComboBox* ili *ListBox*. Izborom ove stavke startuje se *Lookup Wizard*, kojim se definiše vrsta izvora, način popunjavanja i broj polja. Zbog značaja i *Lookup Wizard*-a o njemu će biti reči u nastavku.

U zavisnosti od izbora tipa podatka u donjem panelu *Design View* prozora prikazuje se dodatna lista svojstava svrstanih u dve grupe (kartice): *General* i *Lookup*.

GENERAL kartica sadrži listu opštih svojstava polja:

Field Size je veličina polja; podrazumevani tip i dužina se podešava opcijama u MS Access-u. Najčešće je to 50 karaktera za tekst, a *Long Integer* za brojeve.

Decimal Places je svojstvo specifično za numeričke podatke i određuje broj cifara iza decimalne tačke.

Format definije poseban izgled tekstualnog polja, dok je za brojeve predloženo nekoliko standardnih tipova (*General Number*, *Currency*, *Fixed*, *Standard*, *Percent*, *Scientific*), kao i za *Date/Time* (*General**, *Long**, *Medium**, *Short**). Ovi formati u velikoj meri zavise od podešavanja u *Regional Settings* servisu *Control Panel*-a, pa je pravi trenutak da u tom servisu sve podesite na formate ispisa podataka u našem regionu i to:

1. *Number*: sa decimalnim zarezom i tačkom kao separatora za grupe, a tačka-zarez za separator liste
2. *Currency*: simbol za valutu podesite na "din", a ostalo kao i za *Number*
3. *Time*: "HH:mm:ss" i za separator dvotačku
4. *Date*: "dd.mm.yyyy", bez tačke na kraju i tačku za separator

Input Mask definije masku za unos podataka, koju korisnik mora ispoštovati prilikom unosa podatka, a korisna je u slučajevima unosa: broja telefona, broja pošte, matičnog broja i sl.

Caption po potrebi definije naslov polja. Ovaj tekst će biti ubačen pri kreiranju grafičko-upravljačke kontrole u nekoj formi koja prikazuje podatke iz ove tabele. Ukoliko se ovde ne unese ništa, uzima se već postavljeno ime polja u *Field Name*, pa se njegov unos savetuje u slučajevima kada, a ako ne bude uneto podrazumeva se da ima vrednost kao i ime polja (*Field Name*).

Default Value predstavlja vrednost koja će se automatski upisati u polje pri popunjavanju novog zapisa.

Validation Rule je izraz koji definije zahteve koje mora da ispunи uneti podatak. Obično su uslovi vezani za brojnu vrednost i opseg kome dati podatak pripada, a može se rešiti i pitanje da li je podatak uopšte unet, jer ako nije unet ima problematičnu "vrednost" *Null*.

Validation Text je tekst poruke koja se pojavljuje ukoliko uslov *Validation Rule* nije zadovoljen. Kreiranjem potpune validacije izbegavaju se komentari, koje korisnik dobija na engleskom.

Required određuje da li je u tom polju unos podatka obavezan (Yes) ili ne (No).

Allow

Zero Length je svojstvo specifično za tekstualne podatke i određuje da li je dozvoljeno postojanje stringa nulte dužine ("") u polju.

Indexed je svojstvo kojim se uključuje indeksiranje na određeno polje u cilju izbegavanja dupliciranja podataka i bržeg pretraživanja ili sortiranja po specificiranom polju. Efekatindeksiranja se može sagledati tek kod rada "velikih" tabela (par hiljada slogova) i baza podataka sa više ovakvih tabela, a ponuđene su tri opcije:

- 1) polje nije indeksirano (No),
- 2) polje je indeksirano, a duplikati vrednosti su dozvoljeni (*Yes-Duplicates OK*) ili
- 3) polje indeksirano, ali duplikati nisu dozvoljeni (*Yes-No Duplicates*).

Kod polja koje predstavlja primarni ključ mora biti odabранo (*Yes-No Duplicates*).

Unicode

Compression omogućava smanjenje prostora koji će biti zauzet *Unicode Text*-om. Ovo svojstvo je omogućeno u verziji MS Access 2000, a iz razloga što *Unicode Text* zauzima dvostruko veći memoriski prostor (1 znak zauzima 16, a ne 8 bita kao *ASCII Text*).

LOOKUP kartica sadrži svojstva u slučaju da se podataka u tekućem polju bira iz liste vrednosti. Sve započinje izborom tipa kontrole koja prikazuje listu raspoloživih vrednosti, tako da u zavisnosti od vrste podataka koje polječuva može biti *Text Box* (nije *Lookup* polje), *ComboBox*, i *List Box* (standardno), kao i *CheckBox* (za Yes/No tip podataka). MS Access omogućava da se ovde detaljnije odredi izvor podataka za odnosno polje, mada je *Lookup Wizard*, ipak, za početak jednostavniji.

Lookup Wizard

predstavlja skoro idealan način da inicirate prvu relaciju povezujući dva polja između dve tabele u vašoj bazi, ali obavezno ispoštujte sledeću proceduru:

- 1) kreirajte izvornu tabelu koja obavezno treba da sadrži primarni ključ sa *AutoNumber* ili *Number* tipom podatka i bar još jedno polje (obično tekstualno i male dužine);
- 2) u izvornu tabelu Zanimanja(ZanimanjaID, Zimanje) unesite nekoliko tekstualnih podataka (npr. "pravnik", "ekonomista", "novinar", "profesor" i "pekar");
- 3) kreirajte novu tabelu Radnici(Prezime, Ime, Zimanje), pa za jedno polje podatka (Zimanje) iskoristite *Lookup Wizard* i zadajte za izvor podataka liste – prethodno kreiranu tabelu (oba polja);
- 4) ako je potrebno zadajte drugo ime za polje koje ste kreirali, a zatim sačuvajte tabelu (dizajn);
- 5) otvorite istu tabelu za ažuriranje, pa unesite nekoliko slogova podataka.

Uočite da će se *AutoNumber* (ujedno i ključ) sam sakriti, pa ćete u korišćenju polja sa vezom na drugu tabelu videti samo tekstualni prikaz, a u polje podatka će biti upisana numerička vrednost odgovarajuće stavke koju odaberete iz liste. Pri tom, tabelu izvora podataka za listu možete ažurirati, što će se odražavati i na stavke u listi. Uslučaju da vam ovo ne uspe, preostaje da sami podesite parametre u kartici **LOOKUP**.

Imenovanje objekata

Imena objekata (fajl baze podataka, objekti u bazi i polja u tabelama) je najbolje zadavati primenjujući sledeća pravila:

- kombinovati VelikaMala slova bez odvajanja na više reči i bez specijalnih znakova
- VELIKIM slovima imenovati one važnije objekte (mada suštinske razlike nema, ali se brže učavaju u slučaju većeg broja objekata);
- primeniti imenovanje oblika '_IME'_ - kada objekat treba postaviti kao prvi u nizu (specijalni objekat od značaja za projektovanje baze ili aplikacije)
- primeniti imenovanje oblika 'Ime_Sub' - kada odnosni objekat (tabela, forma) predstavlja podobjekat
- objekte, planirane za uklanjanje (brisanje), imenovati sa prefiksom u obliku 'zzIME', jer će se naći na kraju liste objekata (ako je izabran uobičajen prikaz objekata po imenu - *By Name*).
- izbeći kreiranje tabele i upita sa istim imenom, jer može doći do greške prilikom pozivanja MS Access i tabelu i upit smatra izvorom podataka jednakog prioriteta), pa se praktikuje imenovanje upita uz prefiks 'qry' (kao npr. 'qryArtikli'). Mada se prefiksi koriste i kod imenovanja svih drugih objekata (tab, frm, rpt, mod i sl), kod imenovanja upita to nije od samo formalnog već od suštinskog značaja za korektan rad baze.

Sigurnost baze i brisanje objekataⁱ

Kada započinjete novi projekat (sav rad na novoj bazi ili aplikaciji) najpre se odlučite za kreiranje novog foldera (u okviru vaše zaštićene lokacije), a zatim u njemu kreirajte uobičajene foldere (kao npr: DOCs, Backup, Old, Trash,...).

MS Access baza podataka predstavlja jedan fajla, pa se lako može desiti da je, posle višemesecnog rada, i obrišete kao običan fajl. Posebno treba da izbegavate brisanje bez prebacivanja u *Recycle Bin* vašeg Windows-a, bilo preko <Shift> + <Delete> ili podešavanjem *Recycle Bin* na ovu nesretnu opciju, pa ako bazu i obrišete- zname da je u korpi. Interesantna je i činjenica da se fajl baze podataka mnogo lakše briše (preko *Windows Explorer-a*) od mnogih objekata same baze (preko MS Access-a), a da bi obezbedili bazu od brisanja iz *Windows Explorer-a* (koje bi mogao da obavi slučajni korisnik vašeg računara) nabavite neki od programa za zaštitu pristupa folderima ili iskoristite mogućnosti operativnog sistema u oblasti personalizacije (Passwords, Users, SaveSource i slično).

Pošto, objekti obrisani iz baze, ne mogu biti vraćeni, to nameće potrebu da se, prilikom uklanjanja nepotrebnog objekta, a u cilju sigurnijeg rada, primene neka pravila. Ono što je najvažnije- obratite pažnju na svaku poruku kojom vas MS Access upozorava da će objekat biti izbrisana.

Jedno od zlatnih pravila je da objektu dodelite neko ime koje će vas asocirati da je taj objekat planiran za brisanje, ali ga ne treba brisati dok god se ne uverite da sve u modelu baze funkcioniše korektno. Prilikom promene imena objekta treba voditi računa i o objektima sa kojima je preimenovani objekat u vezi, kao naprimjer: tabele u relaciji, ugnježdeni upiti, podobrasci (*Subform/Subreport*).

Drugi način je da, sve nepotrebne objekte prebacite u drugu bazu (*Copy/Paste*), pa ako vam neki od starih objekata (ili neki njegov deo) zatreba - znate da ga možete naći na sigurnom mestu. Ovo je navika mnogih profesionalaca, a ono o čemu treba da vodite računa je da svaki od projekata ima jednu ovakvu bazu. Važno je držati se i pravila kod imenovanja ovakvih (*Recycle*) baza, da ne dođete u situaciju da ne znate koja od baza je "ona prava".

Čak je i redosled brisanja objekata nekada od važnosti, pa se bezopasnim smatra brisanje sloga *pdata* (korisnički pristup). Ovo brisanje se može povratiti (preko *Paste*) ako je za brisanje korišćena akcija *Cut*, ali se ni u tom slučaju ne garantuje istovetnost brisanih sloganova (slučaj polja tipa *AutoNumber*).

Ostala brisanja su u nadležnosti administratora baze, gde treba voditi računa da se brisanjem objekata brišu i svi podaci koje oni čuvaju, a brisanje obavljati redom:

1. polje u tabeli (*Design View*)
2. relacija između dve tabele (*Relationships*)
3. polje primarnog ključa (*Design View*)
4. tabela (*Database Window*)

Brisanje formi i izveštaja nije ni izdaleka tako opasno, jer se za njihovo kreiranje mogu koristiti čarobnjaci, a koji zahtevaju postojanje izvora podataka (tabele i upiti). Brisanje upita ili modula (procedura kao njihovih delova) može predstavljati problem za one koji nemaju iskustva u njihovom kreiranju, a ono što je u ovom slučaju važno je da model baze (tabele i relacioni model) ne može biti narušen, pa se brisanje smatra bezopasnim.

SAVET: Polja podataka, koja se mogu dobiti obradom (izračunavanjem) podataka iz postojećih polja - ne treba kreirati. Tako npr. u tabeli Stavka(Artikal, Komada, Cena) ne treba kreirati polje Vrednost, jer se ono može uvek dobiti proizvodom polja Komada i Cena. Izračunata polja dovode do problema redudanse i suvišnog nakanadnog ažuriranja.

Na sledećim slikama dat je primer dizajniranja tabela, gde se mogu prepoznati značenja atributa i izgledi tabela u dizajn modu.

The screenshot displays four database tables in Microsoft Access:

- Radnici : Table** (Fields: RadnikID, Prezime, Ime, ZaniID, DatumRodjenja, LD, SektorID, Ocena, Aktivan)
- Plate : Table** (Fields: RadnikID, Mesec, Iznos)
- Projekti : Table** (Fields: PID, Projekat, DatumP, DatumZ, Vrednost, Opis)
- P_R : Table** (Fields: P_R)

Each table's field properties are shown in a separate window. A common validation rule for the birth date field (DatumRodjenja) is set to <Date()>. Another validation rule for the value field (Vrednost) is set to >0. A note on the right side of the interface states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names."

VEŽBA: Kreirajte bazu KADROVI i u njoj tabele Radnici, Zanimanja, Sektori, Plate, Projekti i P_R.

6 NORMALIZACIJA BAZE PODATAKA

Ovo je proces podešavanja strukture baze podataka kako bi njene performanse bile poboljšane, kako u pristupu podacima, tako i u pogledu integriteta i brzine procesiranja (sreće se i pod nazivom: kreiranje relacija normalizacijom). Sva pravila normalizacije podataka u nekoj bazi ne moraju se obavezno primeniti, ali u kompleksnijim i profesionalnim baza zama podataka teško ih je izbeći.

Svako od navedenih pravila podrazumeva pravila nižeg reda. Ova pravila bi trebalo da prepoznate u kreiranju baze (Kadrovi).

Pravilo 1: Eliminisanje grupa podataka koje se ponavljaju

Prvo pravilo normalizacije baze podataka sastoji se u tome, da morate napraviti odvojenu tabelu za svaki set kolona koje su u relaciji pa svakoj tabeli date primarni ključ. Za baze podataka koje poštuju ovo pravilo se kaže da su u prvoj normalnoj formi.

Dakle, ako u tabeli imate niz polja (sličnog imena) uklonite ovu ponavljajuću grupu tako što ćete kreirati novu tabelu za ponavljajuće podatke i povežite je relacijom sa ključnim poljem iz prve tabele.

PRIMER: Ako tabelu PLATE(RadnikID, Ime, Mesec, Sektor, Rukovodilac, LD)

svedemo na PLATE(RadnikID, Mesec, Sektor, LD) izbećićemo ponavljanje grupa podataka RadnikID i Ime, kao i Sektor i Rukovodilac.

Pravilo 2: Eliminacija redundantnih podataka

Redudantnim podacima se smatraju oni podaci koji su nepotrebno smešteni na više mesta u bazi podataka. Drugo pravilo normalizacije baza podataka kaže da, ako polje samo delimčno zavisi od više-vrednosnog ključa, premestite ga u drugu (posebnu) tabelu.

Dakle, nemojte smeštati iste podatke na dve različite lokacije (tabele). Ovo može voditi u greške ažuriranja ili brisanja. Ako su ekvivalentni elementi podataka uneseni u dva polja, uklonite drugi element podataka, kreirajte novu tabelu sa drugim elementom i pratećim ključnim poljem, a zatim smestite ključno polje kao relaciju prema lokaciji koja je sadržala oba objekta. Drugim rečima, ako imate potrebu da premestite dva polja da biste stvarno identifikovali slog, ali samo jedno od polja iz te tabele, koje je neophodno za izvršavanje pregleda tabele, potrebna Vam je nova tabela. Za baze podataka koje zadovoljavaju ovo pravilo se kaže da su u drugoj normalnoj formi.

U bazi Kadrovi, u tabelu Plate, radnik se ne unosi svojim imenom već identifikacijom preko njihovog ID broja (RadnikID). Na ovaj način se memorijski prostor za bazu uveliko smanjuje, održavanje baze se olakšava, a brzina operacija nad tabelama povećava.

PRIMER: Ako tabelu PLATE(RadnikID, Mesec, Sektor, Rukovodilac, LD) svedemo na

PLATE(RadnikID, Mesec, SektorID, LD), a podatak o tome ko je Rukovodilac sektora definišemo u tabeli SEKTORI(SektorID, Sektor, Rukovodilac), tada će tabela PLATE biti u drugoj normalnoj formi.

Pravilo 3: Eliminisanje kolona koje ne zavise od primarnog ključa

Polje, koje nije u direktnoj vezi sa indeksnim ključem tabele, bi trebalo da bude preneto u drugu tabelu. U jednoj tabeli podataka bi trebalo da budu smešteni samo elementi podataka koji su u direktnoj relaciji sa primarnim ključem tabele. Ovo pravilo se najčešće odnosi na izvedene podatke ili druga izračunavanja.

Dakle, ako imate elemente podataka koji nisu u direktnoj relaciji sa primarnim ključem tabele, ove elemente bi trebalo da prebacite u njihove sopstvene tabele.

Drugim rečima, ako kolona tabele stvarno ne treba da bude u toj tabeli, ona je verovatno neophodna u nekoj drugoj tabeli. Za baze podataka koje slede ovo pravilo se kaže da su u trećoj normalnoj formi.

PRIMER: Ako tabelu PLATE(RadnikID, Mesec, Sektor, LD) svedemo na

PLATE(RadnikID, Mesec, LD), a podatak o tome u kom je sektoru radnik

unesemo u tabelu RADNICI imaćemo: RADNICI(RadnikID, Ime, Prezime, MB, Sektor)

Dok prva tri pravila normalizacije uključuju eliminaciju ponavljajućih, redundantnih ili relevantnih polja podataka, preostala dva pravila uključuju izolaciju višestrukih relacija u cilju poboljšanja opšteg integriteta podataka. Prva tri pravila su

obično sve što treba da ispoštujete da biste došli do dobro dizajnirane baze podataka, međutim, u slučaju kompleksnijih modela je neophodno slediti i sledeća pravila, a detaljno se obrađuju na višem kursu - *MS Access* (2).

Pravilo 4: Izolacija nezavisnih višestrukih relacija

Ne može postojati tabela koja sadrži dve ili više 1:n ili n:n relacija koje nisu u direktnoj vezi. Ovo pravilo se koristi za poboljšanje relacionog modela baze podataka kada on sadrži više od jedne 1:n relacije.

Drugim rečima, ukoliko je podatak važan, ali nije u direktnoj relaciji sa drugim elementima sloga, moraćete da preneste ove podatke u novu tabelu. Za baze podataka koje slede ovo pravilo kaže se da su učetvrtoj normalnoj formi.

Pravilo 5: Izolacija zavisnih višestrukih relacija

Peto pravilo normalizacije podataka kaže, da je neophodno da izolujete nezavisne višestruke relacije u bazi podataka. Ovo pravilo se koristi za poboljšanje relacionog modela baze podataka kada on sadrži više od jedne n:n relacije.

Dakle, ako imate pravila koja zahtevaju višestruke reference prema istom polju ili setu polja, izolujte to polje u drugu tabelu i kreirajte jednu ili više tabele za povezivanje, koja će sadržati zahtevane reference (veze), a istovremeno zadržati integritet baze podataka.

Drugim rečima, ako nekoliko složenih relacija postoji u Vašoj bazi podataka, treba da izdvojite svaku od tih relacija u njihovu sopstvenu tabelu. Za baze podataka koje slede ovo pravilo kaže se da su u petoj normalnoj formi.

VEŽBA: Otvorite *MS Excel* fajl (roba.xls), pa prema podacima (kolonama u prvom radnom listu) kreirajte povezne tabele u vašoj bazi (roba.mdb), s tim da baza bude normalizovana primenom pravila 1, 2 i 3.

7 RELACIONI MODEL

Izborom Relationships sa Toolbar-a Database otvara se prozor u kome se definije relacioni model (veze polja između tabela). Iz dijaloga Show Table (koji se dobija sa Toolbar-a) biraju se tabele i upiti koji učestvuju u relacijama. Višestruko biranje objekata vrši se uz pomoć tastera Shift. Kada se tabele nađu na panelu, odgovarajuće relacije se kreiraju prenosom polja koje povezuju dva objekta od jednog do drugog. Spuštanjem na odgovarajuće polje u drugoj tabeli na panelu, pojavljuje se dijalog u kome se bira tip i osobine veze. MS Access će automatski prepoznati (sugerisati) vezu 1:N (One-To-Many) ili 1:1 (One-To-One). Ostaje još da se odredi da li treba uključiti referencijalni integritet i u okviru njega da li treba postaviti opcije za kaskadno brisanje i/ili kaskadno ažuriranje sadržaja tabele. Ostale tipove veza mogu se definisati u Join Type.

Veza i relacija

VEZA je izraz za liniju koja spaja dva polja u dve tabele. Tabele se dakle vezuju preko polja podataka a veza može biti višestruka (preko više polja). Višestruke relacije mogu uticati na povećanje broja redundantnih (ponovljenih) podataka u bazi kao i na nemogućnost ostvarenja referencijalnog integriteta date relacije, pa ih treba izbegavati.

Ako veza između dve tabele nema oznaku tipa (1 i ∞) to znači da je veza tek inicirana.

RELACIJA predstavlja jednostruku ili višestruku vezu između dve tabele preko njihovih polja. Sve veze između dve tabele moraju da se uklapaju u jedan tip relacije (1:n, n:1, 1:1), što znači da se u jednoj relaciji ne mogu uključiti veze različitih tipova. Relacija se smatra definisanim (uspostavljenom) tek po uključenju referencijalnog integriteta (dijalog Relationships) i tipa relacije (dijalog Join Properties). MS Access prepoznaće tip relacije, pa se u ovaj dijalog ulazi da bi se tip relacije, eventualno, promenio (najčešće kod upita).



Relacije tipa 1:n i 1:1

Relacija tipa 1:n (ili n:1) je najčešći tip relacije, ali bi se slobodno moglo i reći da se dobar relacioni model zasniva samo na ovom tipu relacije između tabela.

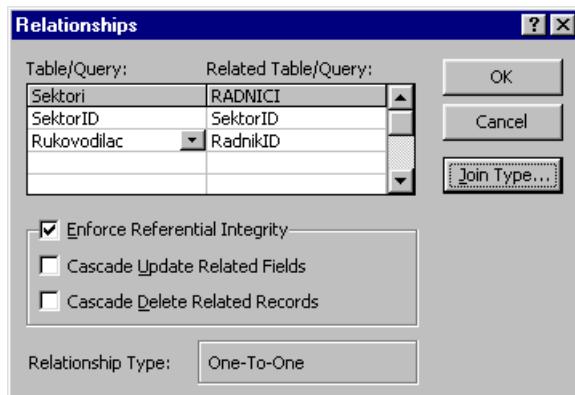
Relacija tipa 1:1 je redi tip, jer daje mogućnost da se obe tabele (koje su u ovoj relaciji) smeste u jednu. U slučajevima kada u jednu tabelu treba smestiti veliki broj polja (atributi jednog entiteta) onda se, zbog preglednosti relacionog modela i njegovog lakšeg održavanja, polja "grupišu" u više tabele, a između definije relacija 1:1. Kao primer za ovo se može užeti neki kompleksan entitet kao što je avion, gde se on mora podeliti na grupe atributa (motor, putnički prostor, kabina posade, teretni prostor, ...), pa treba kreirati tabele prema ovim entitetima, a sve njih povezati po jednom polju (atributu) koji egzistira u svima (npr. AvionID).

Rekurstivna veza

Rekurstivnom vezama se nazivaju ciklične veze između dve ili više tabele. Ovo je slučaj kada bi se, praćenjem smera niza veza u smeru strelica (n:1), formirao kružni tok.

U slučaju dve tabele ovakva relacija se može inicirati, ali je uključivanje referencijalnog integriteta moguće samo u slučaju kada se ova veza zasniva na relaciji 1:1. Za primer se može uzeti relacija između tabele RADNICI(RadnikID, Prezime, Ime, SektorID) i SEKTORI(SektorID, Sektor, Rukovodilac), gde bi trebalo ove tabele vezati jednom vezom 1:n, a drugom n:1.

Drugi slučaj, kada rekurstivnu vezu treba ostvariti između više tabela (npr. tri tabele: RADNICI, SEKTORI, RUKOVODIOCI) je moguće ostvariti, ali se nameće problem održavanja referencijalnog integriteta u slučajevima kaskadnih popravaka i brisanja. Vrlo lako se može doći u situaciju da se brisanjem samo jednog sloga u jednoj tabeli izvrši ("nepotrebno") brisanje velikog broja slogova u druge dve tabele. Drugi problem ove veze je slučaj obavez-



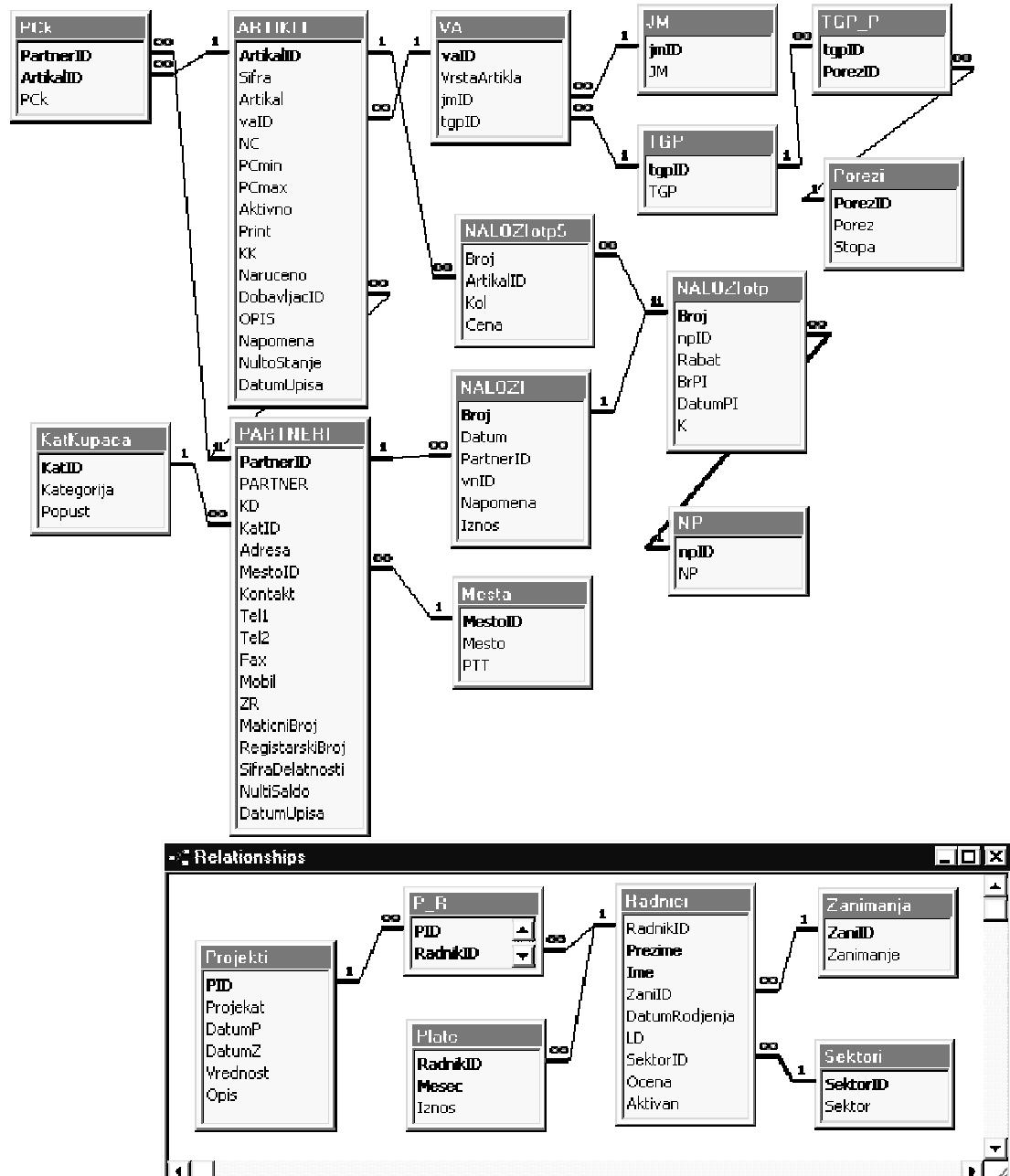
nog podatka koji se ne može upisati jer ne postoji u izvornoj tabeli, a koji na isti taj način ne može da se upiše zbog nedostatka izvornog podatka u trećoj tabeli.

Rekursivna veza se može ostvariti, ali je treba izbegavati zbog nemogućnosti ostvarivanja ispravne relacije, odnosno njenog teškog održavanja.

NAPOMENA: Ako se ukaže potreba da se promeni primarni ključ u tabeli koja već učestvuje u relacijama, pre toga moraju se ukinuti sve njene veze sa ostalim tabelama. Brisanje tabela izRelationships (panela sa relacijama) ne znači da su izbrisane i njene veze, već je ona samo sakrivena u cilju bolje pređnosti. Relacije ostaju čak i kada je tabela izbrisana sa panela, te ih treba brisati jednu po jednu, dok se veza briše – brisanjem vektora koji grafički povezuje tabele u Relationships.

Primer nešto složenijeg relacionog modela:

VEŽBA: Između tabela u bazi Kadrovi uvesti relacije kao na slici.



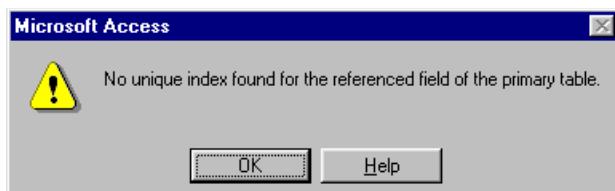
Problemi pri definisanju relacije

Relacija se ne može uključiti jer nije unapred zadovoljen referencijalni integritet, što se javlja u slučaju kada obrađuje model sa već postojećim podacima, pa referentno pdje (destinacija ili izvor, a nalazi se na strani gde je i strelica) ne sadži podatak koji je upisan u zavisno (*Lookup*) polje (sa druge strane veze).

Referentno polje nije primarni ključ niti je indeksirano kao *No Duplicates*.

Referencijalni integritet

Referencijalni integritet (*Referential Integrity - RI*) se u određenoj relaciji može, ali i ne mora uključiti. Svaka relacija u kojoj RI nije uključen predstavlja potencijalnu opasnost za očuvanje ispravnosti rada baze. Ako u relaciji, RI nije uključen, tada se svi mogući slučajevi eventualnog ugrožavanja integriteta baze moraju programirati, što je daleko teže od uključivanja RI. Za sve greške i nastale probleme ugrožavanja RI, u 99% slučajeva, odgovornost je na administratoru baze podataka, dok se tek za 1% moge "okriviti": računar, MS Access, lokalna mreža, napajanje el. energijom i drugi faktori.



Problemi sa integritetom ažuriranja

Integritet kaskadnog ažuriranja treba uključiti u situaciji kada je ažuriranje izvornog podatka korisniku omogućeno. U bazi Kadrovi ova situacija se može nametnuti kada polje RadnikID u tabeli Radnici ne bi bilo tipa *Auto Number*. Vrednost u polju RadnikID je izvorni (referentni) podatak za mnoge tabele u bazi, pa ako bi se njegova vrednost menjala, trebalo bi, na isti način, ispraviti (*update*) i sva polja koja imaju tu vrednost. Dakle, ako bi se RadnikID u tabeli Radnici ispravio sa 3 na 333, to bi trebalo da se uradi i u tabelama Plate i P_R, a kako broj tabela može biti znatno veći, bolje je ovaj posao prepustiti kaskadnom ažuriranju MS Access-a. U slučaju većih zahvata na kaskadnom ažuriranju, prethodno treba kreirati rezervnu kopiju ili se pouzdati u neprekidno napajanje električnom energijom preko solidnog UPS-a.

Ako se kaskadno ili manuelno ažuriranje ne obavi u potpunosti i na kvalitetan način, doći će do situacije pravidnog nepostojanja podataka. Pored toga što podataka nema tamo gde bi ih očekivali, oni se mogu pojaviti tamo gde ih ni slučajno ne bi očekivali, pa se ovo smatra najvećom mogućom greškom u održavanju baza podataka. Čak ni brisanje podataka nije tako opasno, jer se bar zna da ih nema, dok se u ovom slučaju javlja i problem njihovog nekontrolisanog "pojavljivanja" na pogrešnim mestima.

Problemi sa integritetom brisanja

Integritet kaskadnog brisanja se zahteva u slučaju da se izvorni podatak ne ispravlja već briše, jer u slučaju da se briše izvorni podatak, moraju se brisati i slogovi u drugim tabelama, čija polja se referenciraju na brisano polje (naјčešće slog). Pretpostavimo da iz tabele Radnici obrišemo jednogradnika, tada bi se morali brisati slogovi tabela: Plate i P_R, u kojima se on pominje. Dakle, posle brisanja radnika sa podatkom RadnikID = 1, moraju se brisati i svi slogovi iz Plate i P_R u kojima je RadnikID = 1. U suprotnom bi došlo do situacije da poljai slogovi sa referencom ostanu bez izvora, a time i neupotrebljivi. Ovakva polja se mogu uočiti po tome što umesto očekivanog teksta sadrže samo broj (isprobati u bazi Kadrovi).

Rešenje normalizacijom

Primenom navedenih pravila normalizacije treba izbeći bilo kakav oblik redudanse (ponavljanja podataka na više mesta u bazi), ali i pravilno odabrati polja primarnog ključa. Savetuje se unos bar po pet slogova u svaku izvornu tabelu i bar 10-ak u tabelu sa referencom na izvornu, jer se ovim testiranjem praktično uočavaju načinjene greške u dizajnu tabele i relacionom modelu. Baze u kojima je primenjena normalizacija i u svim relacijama uključen referencijalni integritet, su znatno manje, a njihov rad brži i sigurniji. Ako se baza koristi u višekorisnčkom okruženju, navedena pravila nisu dovoljna, pa je potrebno primeniti potpunu optimizaciju modela i radnog okruženja (server računar, parametri lokalne mreže, Jet Engine, ODBC ili SQL server), što se obrađuje na višim kursevima.

VEŽBA: Na ovom terminu polaznici treba da izlože relacioni model baze podataka koju uzimaju kao svoj zadatok, a predavač treba da im sugerise eventualne korekcije.

8 UPRAVLJANJE UPITIMA

Upiti služe kao posredni izvori podataka za izbor podataka na specifičan način. To podrazumeva filtriranje, grupisanje, pretraživanje, sumiranje podataka iz jedne tabele ili povezivanje više tabela po određenim kriterijumima, pa čak i kreiranje novih ili brisanje postojećih zapisa.

Upiti su sekundarni izvori podataka i osnova za forme (*Forms*) koje prikazuju njihove podatke na ekranu, ili izveštaje (*Reports*), kojima se isti prikazuju i štampaju, a osnovne vrste upita u MS Access-u su:

Select Query je najčešći i najjednostavniji tip upita, kojim se pregledaju, analiziraju podaci iz tabele. Podaci mogu da potiču iz jedne ili iz više tabele ili drugih upita.

Crosstab Query gde su podaci grupisani po kolonama i redovima, na način koji je uobičajen za unakrsne tabele. Pogodni su za sumiranje velikog broja informacija u preglednom obliku.

Action Query je specijalni tip upita koji menja veću grupu podataka jednom operacijom, a osnovni tipovi akcionih upita su:

Make Table Query omogućava kreiranje novih tabele izborom polja iz postojećih.

Delete Query briše podatke iz tabele u skladu sa navedenim uslovom.

Append Query dodaje izabrane podatke jedne tabele drugoj.

Update Query menja sadržaj odabralih polja u tabeli.

Union Query nadovezuje kombinujući odgovarajuća (sroдna) polja iz više tabele ili upita u jedinstveni rezultujući set podataka.

SQL Specific upiti karakteristični za SQL (*Structured Query Language*), koji se ne kreiraju u standardnom (*Design View*), već u SQL dizajn modu (*SQL View*):

Pass-Through Query šalje komande direktno na SQL database server, kao što je Microsoft SQL Server ili Sybase SQL Server. Sa tabelama se radi na serveru, umesto da se linkuju u aplikaciju.

Data-Definition Query slično tipu Action Query, menja i kreira nove tabele u bazi.

Kreiranje upita

Kreiranje novog upita započinjemo tako što, u prozoru baze biramo komandu *New*, posle čega sledi dijalog *New Query* (kao na slici) sa sledećim opcijama:

Design View startuje ekran za dizajniranje upita ili kreiranje upita dodavanjem potrebnih tabela ili drugih upita za upit koji se kreira.

Simple Query Wizard je standardni "čarobnjak" za kreiranje jednostavnih upita (npr. grupisanje i sortiranje određenih polja, zapisa odabralih tabela ili ranije formiranih upita po nekom kriterijumu).

Cross-Tab Query Wizard omogućuje kreiranje upita tipa unakrsnih tabela. U toku dijaloga omogućeno je vezivanje nekih od agregatnih funkcija (*Sum, Min, Max, ...*) za polja koja postavimo u preseke redova i kolona, kao i sumiranje po redovima.

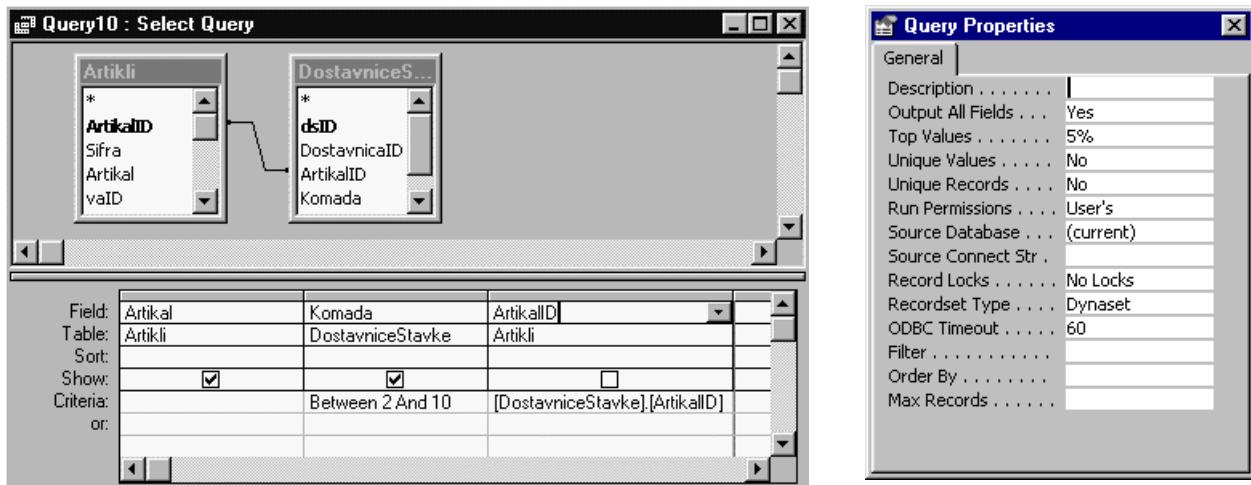
Find Duplicates Query kreira upit koji pretražuje izabranoj tabeli ili upit i nalazi zapise sa jednakim vrednostima u jednom ili više definisanih polja.

Find Unmatched Query pomaže da se kreira upit koji traži zapise koji postoje u jednoj tabeli, a ne slažu se sa pojima druge tabele na, koje bi trebalo da se referenciraju. Referenca (a zatim i relacija) se ne može ostvariti dok se ne obezbedi potpuno slaganje po ovim poljima, pa se ovaj alat koristi kada se pojave problemi sa uspostavljanjem relacije (uključivanje referencijalnog integriteta preko referentnih polja) između dve tabele.

Dizajn upita

Upit je objekat koji nudi tri mōda pristupa: *Design View* (vizuelni dizajn), *SQL View* (dizajn u kōdu SQL-a) i *Datasheet View* (izvršenje, pregled i ažuriranje rezultujućeg seta podataka). Na ovom nivou izučavanja upita mi ćemo se najviše zadržati na *Design View*, a *SQL View* ćemo konsultovati da bismo shvatili koncepciju SQL jezika.

Izborom *Design View*, pri kreiranju novog upita, otvara se prozor učijem zaglavlju je ispisano ime i tip upita (Query10-Select Query), a tu je i prozor *Query Properties* (koji se prema potrebi uključuje) za podešavanje nekih svojstava na nivou polja ili upita.

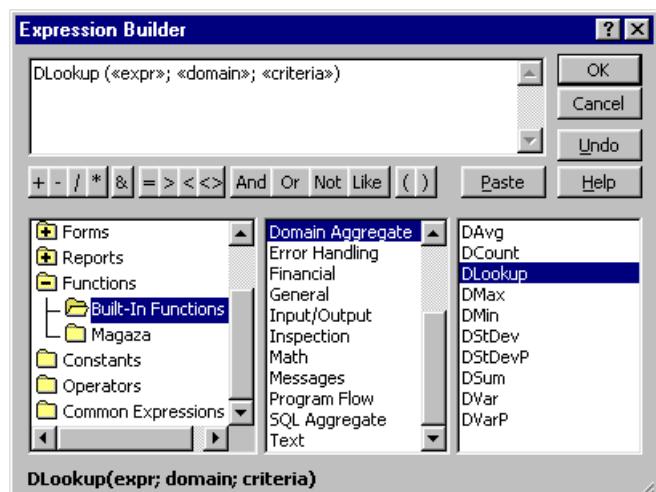


U ovom mōdu se najčešće dizajnira, odnosno edituje i isprobava postojeći upit. U gornjem panelu prikazuju se tabele ili upiti koji sačinjavaju upit koji se dizajnira, dok se u donjem panelu ubacuju polja koja sačinjavaju upit. Pri kreiranju novog upita automatski se otvara dijalog za izbor tabела ili upita koji će ga sačinjavati, a inače se može pozvati preko *Show Table* tastera. Željeni objekat se selektuje, a klikom na *Add*, prebacuje u gornji panel, a isto se može učiniti i dvostrukim klikom na željenu tabelu ili *query*. Polja koja će se naći u upitu, biraju se prenošenjem iz gornjeg panela na slobodno mesto (*Field*), ili dvostrukim klikom na njega u gornjem panelu.

Ako je u *View* meniju čekirana opcija *Table Names*, u donjem panelu se prikazuje i polje *Table*, u kome stoji ime objekta iz koga potiče dato polje. Ako upit treba da sadrži neke sumirajuće funkcije ili matematičke izraze, izborom opcije *Totals* u donjem panelu pojavljuje se i polje *Total* koje nudi (funkcije): *Group By, Sum, Avg, Min, Max, Count, StDev, Var, First, Last, Expression, Where*. Upiti sa totalima se ne mogu ažurirati, tj. preko njih se ne mogu menjati podaci u tablama.

Opcija *Sort* daje mogućnost sortiranja po izabranom polju (ili poljima) po rastućem (*Ascending*) ili opadajućem (*Descending*) poretku, a sa *Show* se bira da li će po izvršenju upita dato polje sa podacima biti vidljivo. U odeljku *Criteria* upisuju se granični uslovi za prikaz podataka iz odgovarajućeg polja. Uslovi može da bude više, i oni se upisuju jedan ispod drugog (*Or*), čineći logičku OR operaciju. Horizontalna sa uslovima za više polja čini logičku AND operaciju. Kriterijum najčešće čine logički i relacioni operatori, ali može biti i kombinacija konstanti, operatara, funkcija, literara, imena polja, kontrola i svojstava (*properties*) uz standardne džoker znake * i ?.

Expression Builder (iz upita se pokreće preko stavke *Build* u *PopUp* meniju) je alat MS Access-a koji će vam pomoći u korišćenju funkcija MS Access-a i SQL-a, u kreiranju izraza (formula), ali je njegova osnovna namena da vas uputi u osobine funkcija i objekata koji vam stoje na raspaganju, pa, iako je bez njega je uglavnom brže (ako poznajete nove programiranja u VBA), on predstavlja dobar edukativni alat.





9 OSNOVE SQL-A

Šta je to SQL?

SQL (Structured Query Language) je strukturni jezik za postavljanje upita i višestruku manipulaciju podacima koji se koristi u VisualBasic-u i MS Access Jet bazama podataka kao primarni metod za pristup podacima. SQL naredbe se mogu podeliti u dve osnovne kategorije i to: naredbe za definiciju podataka (*DDL*) inaredbe za manipulaciju podacima (*DML*).

DDL naredbe omogućavaju definisanje tabela, indeksa i relacija u bazi podataka, dok se **DML** naredbe koriste za selektovanje, sortiranje, sumiranje, izračunavanje u cilju dobijanja potrebnih informacija koje su smeštene u tabelama podataka. Zbog kompleksnosti i obimnosti gradiva u ovoj oblasti, izučavaćemo samo *DML* naredbe, a ostalo ostaviti za kasnije.

SQL je razvijen 1970 godine u IBMu sa idejom da obezbedi korisnicima računara standardni metod za odabir podataka iz različitih formata baza podataka. Namena je bila da se napravi jezik koji neće biti baziran ni na jednom do tada poznatom programskom jeziku, ali koji bi se, opet, mogao koristiti unutar bilo kojeg programskog jezika za ažuriranje i kreiranje upita u bazi podataka.

Većina SQL naredbi vraća rezultat u obliku sada slogova podataka koji se naziva prikaz ili rezultujući set. SQL nije preteđano korisnički orijentisan. Mnogi programi, koji koriste SQL iste naredbe skrivaju iza dijaloga, upita po primeru, ili uz pomoć drugog korisnički orijentisanog interfejsa. Ako su Vaši podaci smešteni u relacionu bazu podataka, Vi ćete koristiti SQL naredbe, znali Vi to ili ne.

U okviru svakog seta SQL standarda postoje 3 nivoa kompatibilnosti. Proizvod koji pretenduje da bude SQL kompatibilan mora da zadovolji 1. nivo kompatibilnosti. Nivoi 2. i 3. su opcioni i proizvodi koji se bave bazama podataka bi trebalo da ih podržavaju kako bi povećali međusobnu kompatibilnost u različitim sistemima relacionih baza podataka.

Mi ćemo SQL učiti preko sintakse Microsoft Jet SQL-a i njegovog testiranja u MS Access okruženju. Kada jednom savadate koncept koji je ovde prikazan, bićete u mogućnosti da te veštine koristite u praktično svim SQL orijentisanim programerskim alatima i alatima za projektovanje i upravljanje bazama podataka. Sve upite ćemo formirati u bazi podataka MAGAZA istim redosledom kojim su i ovde dati (od Query01 do Query24), ali ćemo, na osnovu primera u njima, kreirati i neke upite koje sami osmislimo.

U ovoj lekciji ćete naučiti:

- kako se koristi SELECT..FROM za odabir podataka iz jedne ili više tabela i prikaz tih informacija u jednoj tabeli koju možete listati ili ažurirati;
- naučićete i kako da limitirate podatke, koje treba selektovati, na samo one slogove podataka koji zadovoljavaju kriterijume date klauzulom WHERE;
- da jednostavno presortirate podatke u tabeli korišćenjem klauzule ORDER BY;
- da kreirate jednostavne naredbe koje će automatski sumirati i dati totale i podtotale koristeći klauzulu GROUP BY..HAVING;
- tipične SQL funkcije za manipulaciju brojevima i stringovima;
- da povežete "slične" tabele u jedinstvenu sa poklapanjem kolona klauzulom UNION;
- da koristite napredne tehnike JOIN i TRANSFORM..PIVOT klauzula i da uživate u lakoći njihovog korišćenja.

Napredne mogućnosti SQL-a se izučavaju na višim kursevima, a ako planirate da se profesionalno bavite baza podataka, preporučujem vam da potražite neki od referentnih (zvančno dokazanih, autorizovanih i potpunih) priručnika za SQL (kao jezik) ili SQL Server (kao alat).

Naredba SELECT..FROM

Ova naredba omogućava prikupljanje slogova podataka iz jedne ili više tabela baza podataka. Kao rezultat SELECT..FROM naredbe dobija se pregled podskupa izvornih podataka. U svojoj najjednostavnijoj formi SELECT..FROM sadrži dva dela:

- listu od jedne ili više kolona tabela,
- listu od jedne ili više tabela koje sadrže zahtevane kolone.

Najjednostavniji primer ispravne SQL naredbe je:

SELECT Artikal FROM Artikli;

Ova naredba proslediće Microsoft Jetu komandu da vreti (prikaže) objekat koji sadrži polje **Artikal** iz tabele **Artikli**, a rezultat pokretanja upita trebalo bi da izgleda kao na sledećoj slici.

Artikal	Vrsta Artikli
Alat za punjenje rashladnih ure	Alati
Alat za sečenje cevi	Alati
Klešta-sečice za kapilaru	Alati
Grejač 2 kW po porudžbini	Alati
Šuko priključnica dupla	Alati
Cev gr.I-1620 p=1100w	Alati
Kreda (stick)	Alati
▶ Krckalica	Alati
WD - 40	Alati
Brava	Alati
Šarka	Alati
Amort.vmg olimpik 016 prišt. 1	Amortizer
Amort.vmg olimpik 017 prišt. 2	Amortizer
Amort.vmg ps100 kraći prišt.	Amortizer
Amort.vmg ps664 duži prišt.	Amortizer
Amort. Vm gorenje kraci	Amortizer
Amort. Vm gorenje duzi	Amortizer
Amortizer suspa kraci	Amortizer
Amortizer suspa duzi	Amortizer
Amort. Vm niš	Amortizer
Automatski osigurač 25a dupli	Autom. osiq.

Query01 : Select Query	
Artikal	
▶ Alat za punjenje rashladnih uređaja	
Alat za sečenje cevi	
Klešta-sečice za kapilaru	
Grejač 2000W po porudžbini	
Šuko priključnica dupla	
Cev gr.I-1620 p=1100w	
Kreda (stick)	
Krckalica	
WD - 40	
Brava	
Šarka	
Amort.vmg olimpik 016 prišt. 1	
Amort.vmg olimpik 017.prišt. 2	

U ovom slučaju SELECT..FROM naredba je vratila sve redove tabele bez obzira da li tabela ima 10 ili 10000 slogova. Ovakva primena može biti opasna u smislu velikog usporenenja rada u mreži, prekoračenja memoriskog prostora ili eventualnog blokiranja računara u situaciju kada odabrana tabela ima ekstremno veliki broj slogova podataka.

SELECT naredba (Query02) koja prikazuje sve kolone tabele **Artikli** izgledala bi ovako:

SELECT * FROM Artikli;

Redosled u kojem navodite imena kolona u SELECT..FROM naredbi takođe upravlja redosledom kolona koje će se pojaviti u prikazu, kao na sledećoj slici u slučaju primene upita (Query03):

**SELECT Artikli.Artikal, Artikli.valid
FROM Artikli;**

Klauzula ORDER BY

Kada koristite naredbu SELECT..FROM rezultujući set podataka će biti vraćen u redosledu u kojem je pronađen u izvođenoj tabeli. Ako rezultate treba prikazati u nekom drugom redosledu treba koristiti klauzulu ORDER BY. Opcije ASC i DESC iza svakog polja unutar klauzule ORDER BY se koriste za sortiranje po rastu, odnosno opadanju. Ako nijedna opšta nije data, SQL podrazumeva sortiranje po rastućem redosledu.

Sledeći primer daje tabelu **Artikli** sa kolonama **Artikal** i **SerijskiBroj** sa opadajućim sortiranjem po polju **valid** (vrsta artikla) (Query04):

**SELECT Artikli.valid, Artikli.Artikal
FROM Artikli
ORDER BY Artikli.valid DESC;**

U okviru klauzule ORDER BY može se uneti više od jednog polja, kao npr. (Query05):

**SELECT Artikli.valid, Artikli.Artikal
FROM Artikli
ORDER BY Artikli.valid DESC , Artikli.Artikal;**

čime se dobija rastuće sortiranje po koloni **Artikal**, a opadajuće po koloni **valid** (vrsti artikla).

Vrsta Artikla	Artikal
Ventil vm	VM 2- kraki mp
Ventil vm	VM 1- kraki mp
Ventil vm	VM 2 - kraki ital
Ventil vm	VM 3 - kraki mp
Ventil vm	VM 1 - kraki ital
Term. t. a.	Monofazni 110 c
Term. t. a.	Trofazni 50-280c
Term. t. a.	Trofazni 20-100c
Term. t. a.	Monofazni 50-220 c
Term. t. a.	Trofazni 250 c nale
Ter. v. m. klikso	nz 90 c
Ter. v. m. klikso	nz 88 c
Ter. v. m. klikso	nz 60 c
Ter. v. m. klikso	no 40 c
Ter. v. m. klikso	VM nz 30/nz 57
Ter. v. m. klikso	VM nz38/nz90 4 izvoda
Ter. v. m. klikso	nz 27 / nz 50 3 izvoda

Klauzula WHERE

Jedan od najnaprednijih aspekata naredbe SELECT..FROM je njena mogućnost da upravlja sadržajem rezultujućeg seta podataka uz pomoć klauzule WHERE. Ova klauzula može upravljati sadržajem rezultujućeg seta podataka na dva načina:

- limitirajući sadržaj rezultujućeg seta podataka i
- povezivanjem dve ili više tabele u jedinstveni rezultujući set podataka.

Korišćenje klauzule WHERE za limitiranje rezultujućeg seta podataka

Klauzula WHERE omogućava logička poređenja podataka iz bilo koje kolone tabele podataka. Najprostiji oblik korišćenja ove klauzule je: **WHERE kolona = vrednost**

Vrlo je važno znati da WHERE klauzula uvek prethodi SELECT..FROM naredba, pa kreirajmo upit oblika(Query06):

```
SELECT DostavniceStavke.ArtikalID, DostavniceStavke.Komada
FROM DostavniceStavke
WHERE (((DostavniceStavke.Komada)=1));
```

koji će nam dati prikaz sloga kada je vrednost u polju **Komada** jednaka 1.

Različite WHERE klauzule se mogu povezati u jednu korišćenjem AND, OR i NOT operatora, kao u primeru(Query07):

```
SELECT Dobavljac.Dobavljac, Dobavljac.MestolID
FROM Dobavljac
WHERE (((Dobavljac.Dobavljac)<>'Print 93') AND ((Dobavljac.MestolID)=1));
```

gde će biti prikazani oni slogovi koji zadovoljavaju uslova da **Dobavljac** nije tekst **'Print 93'**. a vrednost u polju **MestolID**, u istom slogu, jednako je 1.

S obzirom da se SQL programiranje često zahteva i u VisualBasic-u obratite pažnju na korišćenje jednostrukih () navodnika u okviru SQL upita (MS Access podržava i dvostrukе), jer ćete na taj način olakšati njihovo održavanje na isti način kao i VisualBasic stringova, npr.:

```
frmER.datSecRS.RecordSource = "SELECT [Engleski] FROM [RECNIK] WHERE [Srpski]='' &
frmER.datPrimRS.Recordset![Srpski] & "'' & " Order_by [Srpski]""
```

Ovde je omogućeno selektovanje po promenljivom podatku frmER.datPrimRS.Recordset![Srpski] koga korisnik unosi, pa se on uklapa u jedinstveni SQL string, gde se jednostruki navodnici moraju navesti ispred i iza ove literarne vrednosti.

LIKE klauzula vraća sve redove u čijim kolonama se nalazi sadržaj sličan literaru koji je prosleđen funkcijom, a u okviru izraza se mogu koristi uobičajeni džokerski znaci * i ?, kao u sledećem primeru, gde ćemo dobiti one artiklečije ime počinje ili završava na 'A' (Query08a):

```
SELECT Artikli.valID, Artikli.Artikal
FROM Artikli
WHERE (((Artikli.Artikal) Like "A*")) OR (((Artikli.Artikal) Like "*a"));
```

BETWEEN klauzula daje slogove čije se deklarisano polje nalazi između datih vrednosti, kao u primeru (Query08b):

```
SELECT Dostavnice.DostavnicaBroj, Dostavnice.DobavljacID, Dostavnice.DatumPrispeca
FROM Dostavnice
WHERE (((Dostavnice.DatumPrispeca)
Between [Od datuma:] And [Do datuma:]))
ORDER BY Dostavnice.DatumPrispeca DESC;
```

gde se parametarskim upitom prikazuju podaci o dostavnicama za vremenski period između dva uneta datuma.

Korišćenje klauzule WHERE za povezivanje dve ili više tabela

Klauzula WHERE se može koristiti za poređenje kolona u različitim tabelama. Na ovaj način mogu se postaviti kriterijumi koji će povezati dve ili više tabele u jedinstveni rezultujući set podataka kao u primeru (Query09):

```
SELECT Artikli.Artikal, DostavniceStavke.Komada
FROM Artikli, DostavniceStavke
WHERE ((Artikli.ArtikalID)=[DostavniceStavke].[ArtikalID]));
```

gde su povezane tabele **Artikli** i **DostavniceStavke** i prikazani podaci **Artikal** i **Komada** u slučajevima kada polja **ArtikalID** iz jedne i druge tabele imaju istu vrednost. Rezultujući set će u ovom slučaju izgledati kao na slici desno.

Dok će upit (Query10):

```
SELECT Artikli.Artikal, DostavniceStavke.Komada
FROM DostavniceStavke INNER JOIN Artikli ON DostavniceStavke.ArtikalID = Artikli.ArtikalID
WHERE ((DostavniceStavke.Komada) Between 2 And 10) AND
((Artikli.ArtikalID)=[DostavniceStavke].[ArtikalID]));
```

dati samo one slogove gornjeg seta podataka, uz dodatni uslov da je vrednost u polju **Komada** između 2 i 10, pa samim tim verovatno dati kraći set podataka za prikaz. U ovom slučaju nije korišćeno uslovljavanje (WHERE) po polju **ArtikalID**, već je zahtevano unutrašnje slaganje (**INNER JOIN**) tabele po polju **ArtikalID** (što je često i jednostavnije).

SQL funkcije za agregaciju

SQL standardi definišu osnovni set funkcija koje su sadržane u svim SQL kompatibilnim sistemima, a poznate su pod imenom *funkcije za agregaciju*. Ove funkcije se koriste za brzo izračunavanje rezultata iz numeričkih podataka koji su smešteni po kolonama. SQL funkcije za agregaciju koje se koriste u MS Access Jet-u su:

Group by označava polje (ili jedno od polja) po kome se podaci u *Total* upitu grupišu.

Sum daje zbir (sumu) po polju u odnosu na prethodno *Group By* polje.

Avg daje aritmetičku sredinu vrednost za dato polje, u odnosu na polja po kome se grupiše.

Min daje minimalnu vrednost u datom polju.

Max daje maksimalnu vrednost u datom polju.

Count daje broj različitih vrednosti koje polje uzima, ne računajući nultu vrednost.

StDev daje standardnu devijaciju vrednosti u polju.

Var daje varijansu (kvadrat standardne devijacije).

First daje vrednost polja u prvom zapisu u setslogu koji vraća upit.

Last daje vrednost polja poslednjeg zapisu u setslogu koji vraća upit.

Expression Koristi se za prikaz izračunatih vrednosti, unošenjem funkcija nad poljima. Funkcija može biti jednostavna, jedna od agregatnih (*sum, avg, min, max, stdev, var*), aritmetičkih ili složenija za šta se obično koristi opcija *Build* iz menija koji se dobije desnim klikom na *poljeField*:. Potom se izabere jedna od internih, ili korisnički kreiranih, funkcija.

Where kriterijum uslov u vidu SQL izraza, za polje koje upit ne prikazuje u setslogu.

Primer korišćenja funkcija agregacije se može dati upitom (Query12):

```
SELECT Sum(DostavniceStavke.Komada) AS Zbir,
Min(DostavniceStavke.Komada) AS Najmanje, Max(DostavniceStavke.Komada) AS Najviše,
Count(DostavniceStavke.Komada) AS [Ukupan broj]
```

FROM DostavniceStavke

WHERE (((DostavniceStavke.Cena)<1));

VEŽBA: Kreirati upit kojim se dobija pregled sumiranja plata po radnicima.

Artikal	Komada	Record:
Gr.ta elind cigla 1750w	6	
Trofazni 20-100c	6	
Grejna ploča*145 termostatska	5	
Grejna ploča 220	4	
Gr.ta elind cigla 2500w 3iz	1	
Grejač 2 kW po porudžbini	2	
Nosač ležajeva vm Obodin	1	
Grejna ploča 100	200	
Gr.bojl 2000w koncar protocni	300	
Kreda (stick)	1	
Pumpa vm niš Mikron	100	
Razvodn. Tabla 30 os n/z met	500	
Pumpa vm Gorenje 663 sa plasti	1	
Pumpa vm Obodin Mikron	100	
Grejna ploča 145 expres	11	
Trofazni 50-280c	1	

Zbir	Najmanje	Najviše	Ukupan broj	Record:
2100	100	1000	3	

Klauzule DISTINCT i DISTINCTROW

Kada selektujete podatke iz tabele, postoje situacije u kojima se jedan red može pojaviti više puta. Na primer, kada želite da dobijete listu svih dobavljača koji su bar jednom izvršili dostavu. Problem se sastoji u tome da neki dobavljači imaju nekoliko dostava u tabeli dostavnica. Ako ne želite da se njihova imena ponavljaju u rezultujućoj tabeli treba da koristite klauzulu DISTINCT, kao npr.(Query13):

```
SELECT DISTINCT Dostavnice.DobavljacID FROM Dostavnice  
ORDER BY Dostavnice.DobavljacID;
```

Takođe postoje situacije kada treba prikupiti podatke o svim redovima koji se razlikuju u svim poljima. Umesto da koristite ključnu reč DISTINCT u svim poljima tabele, možete koristiti ključnu reč DISTINCTROW (Query15):

```
SELECT DISTINCTROW * FROM Dostavnice  
ORDER BY Dostavnice.DostavnicaBroj;
```

Klauzule TOP n i TOP n PERCENT

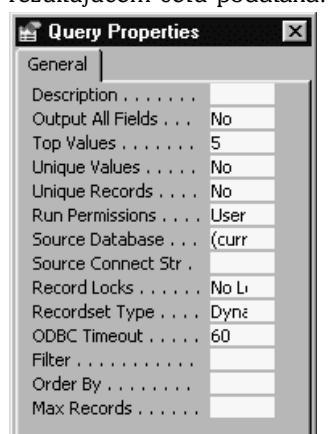
Ključne reči SQL-a TOP n i TOP n PERCENT se koriste za ograničavanje broja slogova u rezultujućem setu podataka. Prepostavimo da želite da dobijete listu od pet najprodavaniji proizvoda. Da biste ovo ostvarili iskoristite klauzulu TOP n, pa ćete dobiti samo prvih n slogova, a ako dva ili više proizvoda imaju istu vrednost prodaje, svi će biti prikazani. Upis broja n se može uneti i kao svojstvo 'Top Values' u prozoru 'Query Properties' (slika desno).

U sledećem prikazu traži se 5 najmanjih vrednosti po broju dostavnice (Query16):

```
SELECT TOP 5 Dostavnice.DostavnicaBroj, [Komada]*[Cena] AS  
Vrednost, Dostavnice.DatumPrispeca FROM Dostavnice  
INNER JOIN DostavniceStavke ON Dostavnice.DostavnicaID = DostavniceSta-  
vke.DostavnicaID  
ORDER BY [Komada]*[Cena] DESC;
```

Oblik: TOP n PERCENT neće vratiti samo 10 slogova nego 10% od ukupnog broja slovova iz rezultujućeg seta (Query17);

```
SELECT TOP 5 PERCENT * FROM Dostavnice  
ORDER BY Dostavnice.[Dostavnica Broj];
```



Klausula GROUP BY..HAVING

Ova klauzula omogućava korišćenje funkcija za agregaciju, ali uz dodatno uslovljavanje u pogledu rezultata ovih funkcija. Klauzula HAVING se ponaša na isti način kao i klauzula WHERE, s tom razlikom što se HAVING klauzula primjenjuje na rezultujuću kolonu koja će nastati korišćenjem GROUP BY klauzule, a nad originalnom kolonom. Sledеći upit će dati samo one articlje kojih je ukupno prodano preko 50 jedinica (Query18):

```
SELECT OtpremniceStavke.AritkalID,
Sum(OtpremniceStavke.Komada) AS SumOfKomada
FROM OtpremniceStavke
GROUP BY OtpremniceStavke.AritkalID
HAVING ((Sum(OtpremniceStavke.Komada))>50));
```

a rezultovače prikazom kao na slici.

	Artikal	SumOfI
▶	Alat za punjenje rashladnih uređaja	57
	Tinjalica Ib-1.10 p crvena	100
	Lezaj 6204 Italija	90
	Okvir vrata vm Gorenje (Olimpik)	99
	Pumpa vm Gorenje Mikron	100
	Razvodna tabla 24 os n/z met	99
	Remenica vm El	99

Kako se ova klauzula koristi gotovo isto kao i WHERE klauzula, to znaci da i ovde mozemo koristiti više, logicki povezanih, uslova, kao u primeru (Query19):

```
SELECT OtpremniceStavke.ArtikalID  
FROM Artikli INNER JOIN OtpremniceStavke ON Artikli.ArtikalID = OtpremniceStavke.ArtikalID  
GROUP BY OtpremniceStavke.ArtikalID, Artikli.Artikal  
HAVING (((Artikli.Artikal) LIKE 'R*') AND ((Sum(OtpremniceStavke.Komada))>50));
```

gde se pored već nomenutog uslova traži da ime artikla počinje slovom "B".

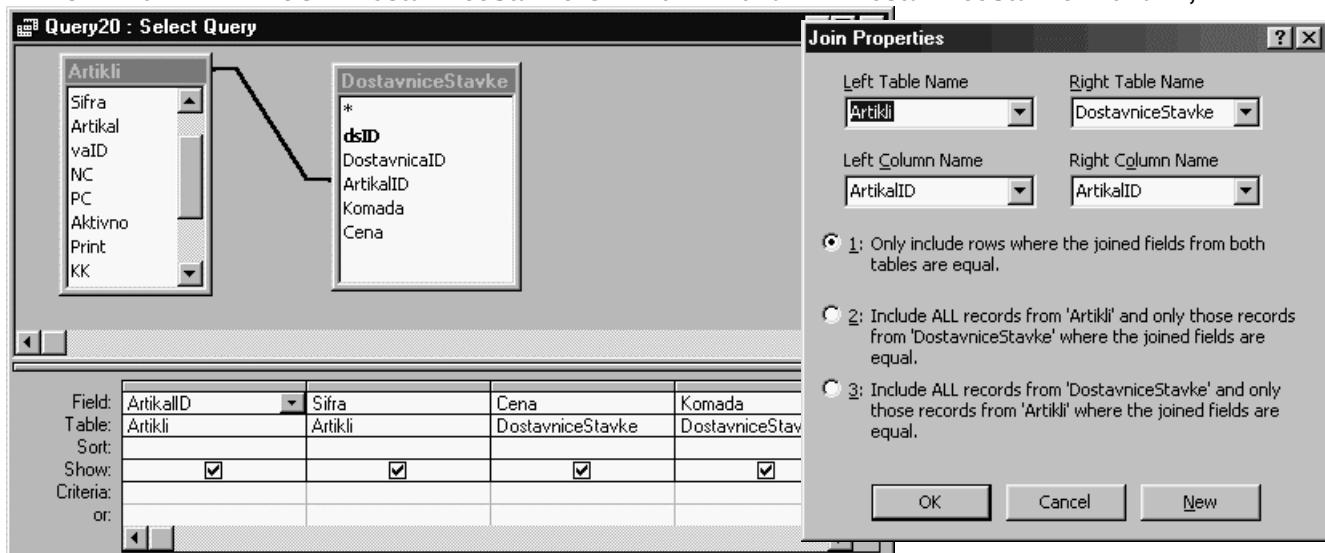
Tipovi SQL klauzule JOIN

Setovi podataka nastali kao rezultat korišćenja WHERE klauzule se ne mogu ažurirati, što se primenom jednog od tri tipa JOIN klauzule može prevazići, odnosno zahtevima za unutrašnjim INNER JOIN ili nekim od spoljašnjih slaganja LEFT JOIN, odnosno RIGHT JOIN. Shvatanje ovog slaganja u mnogome zavisi od vašeg poimanja relacije, jer se u svojstvima odnosne relacije (pored ostalog) deklariše i jedan od ovih oblika slaganja. Zbog kompleksnosti, ova tema se detaljnije obrađuje na razvojnom (višem) kursu., a ovde će biti samo prezentirana na primerima.

INNER JOIN

klauzula INNER JOIN može se koristiti za kreiranje rezultujućeg seta podataka koji sadrži samo one slogove koji odgovaraju jedni drugima u obe tabele, kao npr. kod upita(Query20):

```
SELECT Artikli.ArtikalID, Artikli.Sifra, DostavniceStavke.Cena, DostavniceStavke.Komada
FROM Artikli INNER JOIN DostavniceStavke ON Artikli.ArtikalID = DostavniceStavke.ArtikalID;
```

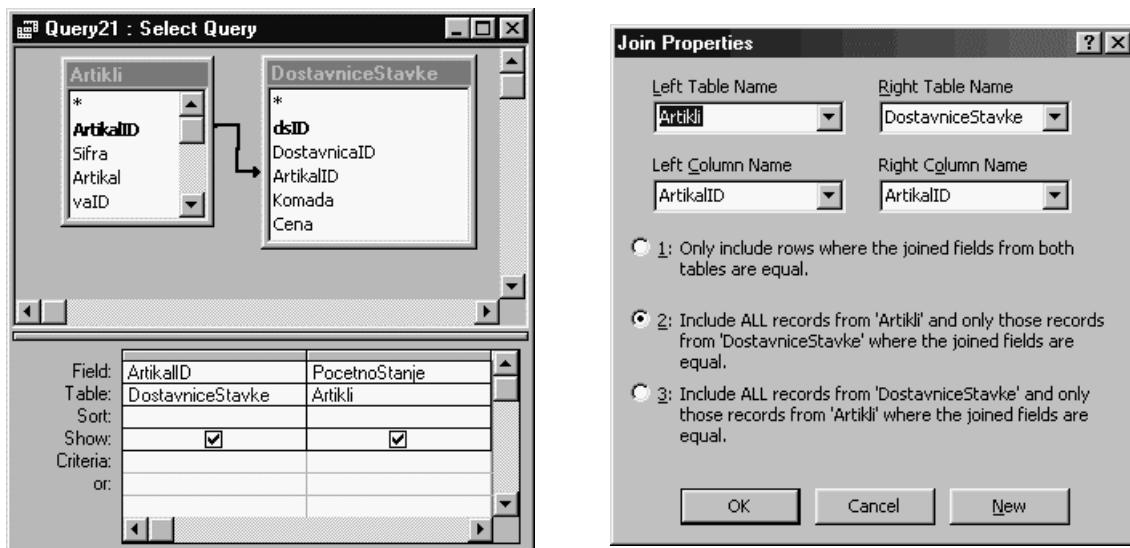


gde se zahteva jednakost polja **Artikli.ArtikalID** i **[DostavniceStavke].ArtikalID**.

LEFT JOIN

Klauzula INNER JOIN vraća samo one redove koji imaju odgovarajuću vrednost jednaku u obe tabele, dok năini spoljni povezivanja daju kao rezultat sve slogove podataka koji vraćaju vrednost sa jedne strane povezivanja, bez obzira da li postoji odgovarajući slog sa druge strane koja se povezuje. Klauzula LEFT JOIN će vratiti sve slogove iz prve tabele u listi (tabela sa leve strane) i odgovarajuće slogove iz tabele sa desne strane, ali samo one koji imaju odgovarajuću vrednost u ovoj koloni, kao u primeru (Query21):

```
SELECT DostavniceStavke.ArtikalID, Artikli.PocetnoStanje
FROM Artikli LEFT JOIN DostavniceStavke ON Artikli.ArtikalID = DostavniceStavke.ArtikalID;
```



RIGHT JOIN

Ova klauzula funkcioniše na sličan način kao prethodna, samo što se rezultujući set podataka bazira na drugoj (desnoj) tabeli, a koristi se na isti način (Query22).

```
SELECT Artikli.Artikal, OtpremniceStavke.Komada, Otpremnice.DatumOTP
FROM Otpremnice INNER JOIN (Artikli RIGHT JOIN OtpremniceStavke ON Artikli.ArtikalID =
OtpremniceStavke.ArtikalID) ON Otpremnice.OtpremnicalD = OtpremniceStavke.OtpremnicalD
ORDER BY Artikli.Artikal;
```

UNION upiti

Ova ključna reč SQL-a omogućava ujedinjenje dve tabele ili dva SQL upita koji sadrže slične podatke, ali nisu u relaciji. UNION vraća set podataka koji nije moguće ažurirati, pa su oni zgodni za izradu prikaza na ekranu, izveštaja, kao i za pripremu podataka za statističke prikaze i grafikone. Union tip upita je jedan od onih koji mogu da se dizajniraju (kreaju) samo u *SQL View*-u.

U sledećem primeru upit daje set podataka sa ujedinjenjem tabela **Dobavljac i Kupci** (Query23):

```
SELECT Dobavljac.Dobavljac, Dobavljac.KontaktOsoba, Dobavljac.Adresa
FROM Dobavljac
UNION SELECT Kupci.Kupac, Kupci.KontaktOsoba, Kupci.Funkcija
FROM Kupci;
```

čija se rezultujuća tabela vidi na slici, gde su i dobavljači i kupci smešteni u prvu kolonu.

Kada koristiti rad u SQL-u? Pored toga što ćete kad-tad morati da radite u SQL-u neki tip upita (kao što je Union) koji ne može da se kreira u *Design View* mōdu, pretpostavimo da imate veoma složen upit, ali da one ne funkcioniše zbog promjenjenog imena tabele, polja ili upita na koji se referenčira. Da biste lakše korigovali greške, stvorite upit u *SQL View*-u, pa ispravite objekata. Ako je greška ispravljena, upit će moći da se vidi i dizajnira i u *Design View* mōdu.

Dobavljac	KontaktOsoba	Adresa
Alarm Inženjeriing	Ivan Obradović	Direktor prodaje
BRANEX	-	Kralja Petra 24/I
DIABOLIS	Stolić Dragan	Menadžer prodaje
NIKKOM	Ostojić Mileva	Cara Dušana 144
Print 93	Dekić Slobodan	Majke Jevrosime 38
ProEnGiS	Kujović Slađana	Kneza Miloša 12
TALTEX Sistem	Kelović Željko	Svete Popovića 5
Top Shop	Beli	Direktor
TRAKS	Kujović Slađana	Direktor marketinga

VEŽBA: U cilju prikaza stanja artikala, kreirati upite:

- početna stanja artikala: qryPS(ArtikalID, PočetnoStanje), koristeći tabelu ARTIKLI
- ulaz artikala: qryULAZ(ArtikalID, Komada), koristeći tabele: ARTIKLI i DostavničStavke
- izlaz artikala: qryIZLAZ(ArtikalID, -Komada), koristeći tabele: ARTIKLI i OtpremniceStavke

Potom kreirati Union upit qryPROMET, kojim će se kreirati unija prethodna tri upita, a koji treba dazgleda kao:

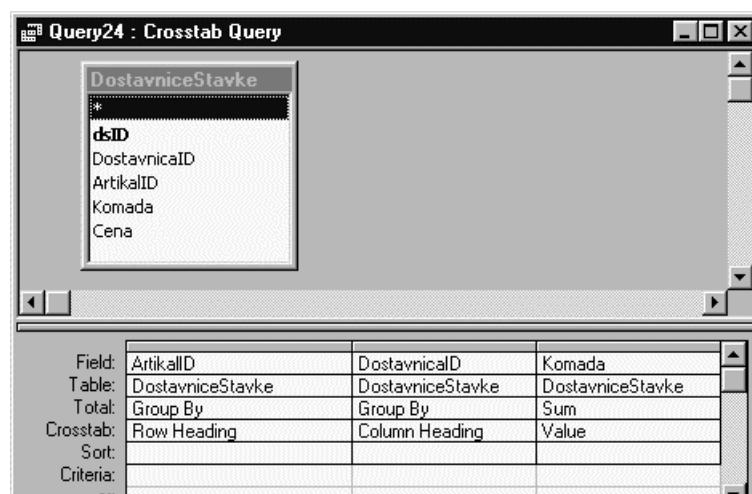
```
SELECT ALL * FROM qryPS
UNION SELECT ALL * FROM qryULAZ
UNION SELECT ALL * FROM qryIZLAZ;
```

Na kraju kreirati upit qrySTANJA(ArtikalID, Stanje), korišćenjem qryPROMET kao izvor podataka, a korišćenjem totala i sumiranje po polju Stanje.

Unakrsni upiti klauzulom TRANSFORM..PIVOT

Ovo je izuzetno složena alat koji omogućava kreiranje rezultujućih setova podataka koji sadrže sumirane podatke u formi koja se naziva *unakrsni upit*, što daje mogućnost međusobne zamene mesta između kolone i reda.

Da bi kreirali ovaj tip upita, najpre treba zadati njegov tip (*Crosstab Query*) iz palete alata. Primer koji će biti prikazan daje rezultujući pregled u obliku tabele sa sadržajem: DostavnicID u kolonama, ArtikallID po redovima i količinom u preseku odgovarajućeg reda i kolone:



TRANSFORM Sum(DostavniceStavke.Komada) AS SumOfKomada

SELECT DostavniceStavke.ArtikalID FROM DostavniceStavke

GROUP BY DostavniceStavke.ArtikalID

PIVOT DostavniceStavke.DostavnicaID;

kao na slici:

ArtikalID		1	2	3	4	5	6	11	12	16	18	19	20	21	22	23	25	26
Klešta-sećice za kapilaru																		
Grejač 2 kW po porudžbini																		
Kreda (stick)																		
Brava			22	26														48

SAVETI: Imena objekata koji se koriste u upitim (tabele, polja i upiti) treba da budu štampana i prepoznatljiva, a u slučaju da ime objekta sadrži neki od specijalnih karaktera, ime objekta mora se upisati u uglaste zagrade, kao npr. [Šifra Radnika], ali je bolje koristiti oblik: SifraRadnika. Komplikovana i duga imena tabela i upita mogu pretvoriti u noćnu moru, ako imamo u vidu da poslovne baze podataka imaju preko (obično) 10 tabela i mnogo više upita (zavisi od koncepcije programera).

10 FORME

11-12

ili obrasci služe za unos, pregled, ažuriranje, ali nekada i za štampanje podataka. Njihovanamena je pristup korisnika pregledu i ažuriranju podataka. Korisnik mora biti l-šen svake mogućnosti da unosi i ispravlja podatke u samoj tabeli, jer se na taj način mogu pojaviti neočekivani i nepredvidivi problemi za programera, ali i za

samog korisnika. Forma (ili obrazac) je najbolje i najlakše grafičko okruženje za korisnika, a ako ona realno oslikava pravi obrazac na koji je korisnik do tada navikao – biće mu mnogo lakše da se snađe u njoj i bez velikog objašnjavanja o načinu popunjavanju i uslovima.

Kreiranje forme

Kao i kod ostalih tipova MS Access objekata, i kod formi se kreiranje započinje preko komande *New* iz *Toolbar-a* prozora baze, ali se savetuje izbor tabele i upita (kao izbora podataka), pa iz osnovnog *Toolbar-a* na *New Object>Form*. U donjem delu dijaloga, koji se pojavljuje, u *ComboBox-u* se bira na koju će se postojecu tabelu ili *Query* (upit) pozvati nova forma. Biranje se vrši označavanjem stavke na listi, ili kucanjem prvih nekoliko karaktera imena tabele ili upita koji ih jedinstveno određuju. Kreiranje forme može započeti na sledeće načine:

Design View Kreiranje forme "od početka", odnosno praznog obrasca na koji se dodaju potrebne kontrole. (ne savetuje se jer predstavlja najsporiji način).

Form Wizard mehanizam ili "čarobnjak" opšte namene (savetuje se što češće).

AutoForm: Columnar Automatsko kreiranje forme (*Columnar*) u kojoj je svaki zapis na posebnoj strani obrasca.

AutoForm: Tabular Automatsko kreiranje forme (*Tabular*) u kojoj su zapisi prikazani tabelarno, po više njih na strani (ekranu).

AutoForm: Datasheet Podaci se prikazuju u obliku koji liči na tabelu u formi.

Chart Wizard Biranjem ove opcije pravi se forma sa OLE objektom (ulazi se u program *MS Graph* koji spada u "deljene aplikacije", odnosno ima istu ulogu u više programa, na primer *Word-u*) u kome se kreira grafikon na osnovu podataka iz neke tabele ili upita.

Pivot Table Wizard Kreiranje formi za prikaz tabela za interaktivno unakrsno izračunavanje, pri čemu se potrebne tabele kreiraju u *Excel-u*.

NAPOMENA: Forma se ne mora bazirati na upitu ili tabeli, što može biti primenjeno u slučaju izrade dijaloga, poruka, odnosno menija, gde podatke ne treba ažurirati već samo privremeno koristiti.

Dizajniranje forme

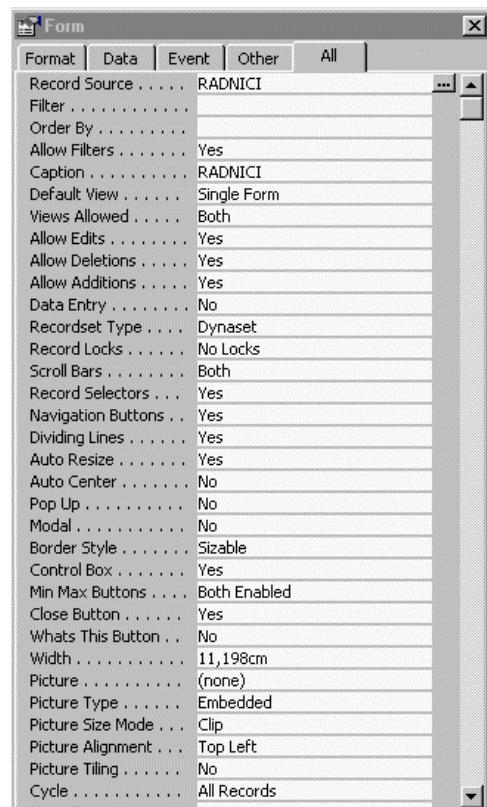
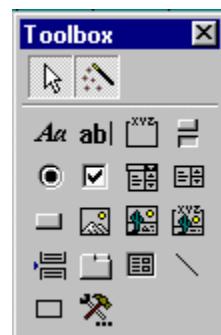
Da bismo dizajnirali formu potrebno je otvorimo uDesign View mōdu ili (ako je forma već otvorena uForm View mōdu), kliknuti na taster koji omogućuje *Design View*. Na Toolbar-u je moguć pregled forme u sva tri mōda (režima): *Form View* (ako je objekat vidljiv), *Design View* (ako u bazi nije zaštićen dizajn), *Datasheet View* (može biti onemogućen). Stavke iz sistema menija koje se koriste za rad u dizajn modu:

- Edit** Nudi standardne akcije ažuriranja u svim Windows aplikacijama (*Undo, Cut, Copy, Paste, Delete, Select All*), kao i mogućnost ubacivanja OLE objekata (*Paste Special*) i kreiranje kopija selektovanog objekta (*Duplicate*). Opcijom *OLE/DDE Links* osvežavamo, editujemo, kopiramo, ponovo uspostavljamo ili zatvaramo veze izabranog OLE objekta. *Object* se bavi svojstvima OLE objekta.
- View** meni nudi sadržaj prve grupe u zavisnosti od mōda rada u formi. U drugoj grupi su svojstva izabranog objekta, (*Properties*), lista polja (*Field List*) tabele ili upita nad kojim je data forma kreirana, mogućnost prelaska u pregled i editovanje koda vezanog za formu (*Code*) kao i *Tab Order* koji obezbeđuje definisanje redosleda prenošenja fokusa po kontrolama u formi.
- Insert** meni omogućava numeraciju strana kao i unošenje kontrole sa tekućim datumom i vremenom. Ovde su i komande za unošenje grafikona, slike, OLE objekata ili *Custom* kontrole.
- Format** meni u prvoj grupi opcija omogućava primenu predefinisanog modela forme ili izveštaja alatom *Auto Format* (nalazi se na Toolbar-u), postavljanje *Default* izgleda kontrole (*Set Default Controls*), kao i promenu izabrane kontrole koja prikazuje određeno polje u neku srodnju (*Change To*). Ostale opcije se tiču fizičkog rasporeda, poravnavanja i podešavanja dimenzija kontrola.

Kontrole i tipovi kontrola

Za kreiranje kontrola u radu vam pomaže *Wizard* koji se startuje iz "kutije sa alatom" (View/Toolbox ili na Toolbar-u) klikom na taster. Kontrole se biraju iz Toolbox-a. Access nudi sledeće standardne Windows kontrole (na slici predstavljene u okviru Toolbox-a):

- Label** (nalepnica) predstavlja statički tekst i upotrebljava se za natpisne naslove, imena drugih objekata i slično.
- TextBox** služi za unos, ispravljanje i prikaz podataka. Naјčešće odgovara jednom polju u tabeli ili upitu nad kojim je forma kreirana. Takođe služi i za prikaz izračunatih vrednosti, kao i za unošenje izbora korisnika preko dijaloga.
- OptionGroup** je kontrola čiji je zadatak da zauzme određenu površinu na formi i okupi *OptionButton*, *ToggleButton* i *CheckBox* na toj površini, koje služe za unos podatka biranjem neke od ponuđenih opcija.
- ToggleButton** (kao i *OptionButton* i *CheckBox*) je kontrola koja može imati jedno od dva stanja (*Yes/No*, *True/False*, *On/Off*, *-1/0*).
- ComboBox** daje mogućnost izbora stavke iz liste. Ponuđeni elementi liste mogu pripadati nekoj tabeli ili upitu, a mogu se i programski uneti.
- ListBox** omogućava izbor jednog od elemenata sa liste koju prikazuje. U listu se ne mogu direktno unositi nove vrednosti.
- CommandButton** je dugme koje omogućava pokretanje neke funkcije, makroa ili akcija napisanih u VBA-u. Ukoliko nemate iskustva u programiranju, postoji mogućnost kreiranja ovih komandnih tastera (za neke akcije) pomoću *Wizard*-a, gde se nude mogućnosti biranja vizuelnih, ali i komandnih svojstava komandnog tastera.
- Image** kontrola služi za unos slike koja može biti kao pozadina na formi.
- ObjectFrame** omogućava prikaz grafičkog ili nekog drugog OLE objekta koji nije u okviru baze podataka, već ima zaseban fajl. Ovako unet objekat može da se ažuriра (edituje).
- Bound**
- ObjectFrame** može da sadrži crtež, grafika ili neki drugi OLE objekt koji je deo neke od tabela u bazi.



PageBreak kontrola deli formu na stranice.

TabControl (kartice) predstavlja grafički objekat kojim se u velikoj meri može omogućiti radna površina, a u isto vreme pojednostaviti korišćenje aplikacije (bez otvaranja više prozora). Kombinacija ove kontrole sa Subform daje izvanredne rezultate i ne predstavlja veliki problem za dizajn.

Subform omogućava da se na formi prikaže neka druga forma (kao podforma). Ovo je izuzetno pogodno kada treba uštedeti na ekranskom prostoru, a da korisnik može sve da dobije brzo i najednom mestu. Poseban efekat se dobija kada treba prikazati više zapisa jedne tabele ili upita koji odgovaraju jednom zapisu na glavnoj formi. Na primer, često se koristi kod otpremnika, faktura, prijemnika i drugih dokumenata koji imaju stavke.

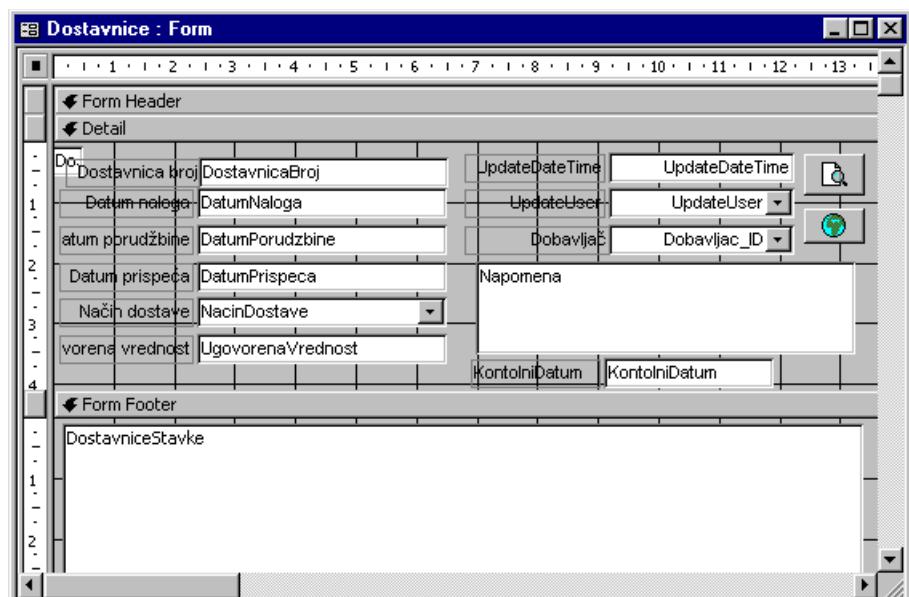
Line omogućava prikaz linija na formi.

Rectangle omogućava prikaz pravougaonika na formi.

U dizajn modu forme, mogu se podešavati svojstva (*Properties*) forme u celini ili za fokusirani (ili selektovani) objekat na formi. Pri tome se može otvoriti *Properties* prozor gde se mogu podešavati svojstva objekata (podeljene po karticama *Format*, *Data*, *Event*, *Other*) koji je prikazan na prethodnoj slici.

Kako izgleda forma na čijem se dizajnu tek radi, možete da vidite na slici desno, gde se kreira forma za ažuriranje podataka o dostavnicama.

VEŽBA: Kreirajte formu Dostavnice (kao na slici), podformu (DostavniceStavke) ćemo dodati kasnije.



Sekcije forme

omogućavaju lakši dizajn zaglavlja i totala:

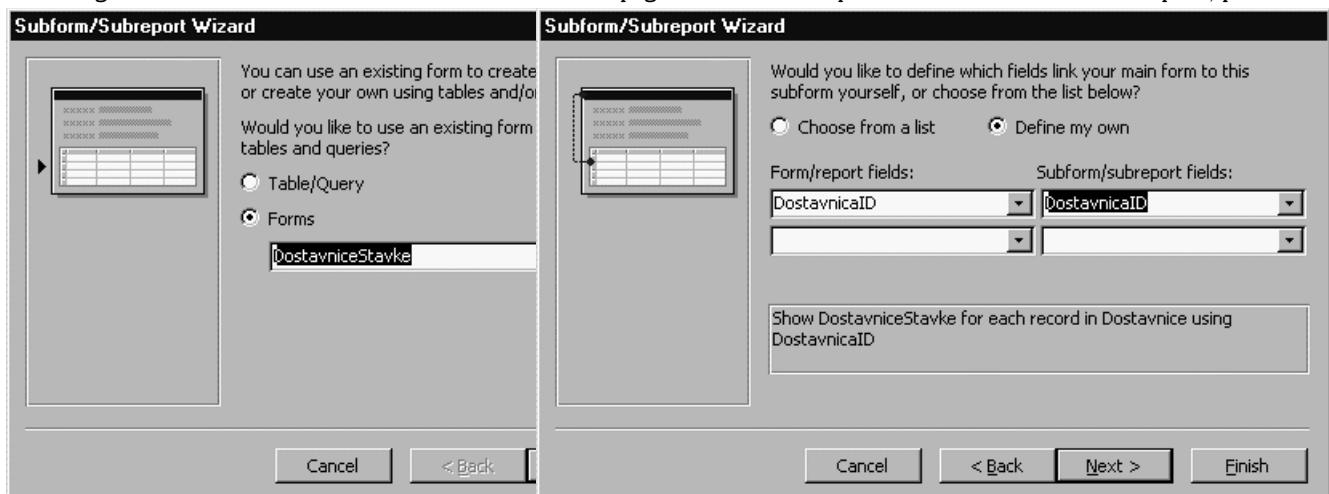
Form Header je zaglavljde forme. Obično se koristi za upis naziva logičkog dela aplikacije koji se obavlja ili podataka o datumu i vremenu.

Detail je centralni deo forme. Tu je smešten najveći deo kontrola i u njemu se najčešće nalazi najveći deo podataka iz tabele ili upita.

Form Footer je donji deo forme u kome se obično nalaze polja za totale.

Subform (podforma)

je čisti podobjekat forme, a kreira se u situacijama kada u glavnoj formi treba omogućiti ažuriranje odataka stavki vezanih za tabelu glavne forme. Kada sa Toolbox-a kliknete na tip grafičke kontrole pod imenom Subform/Subreport, pa nacrtate

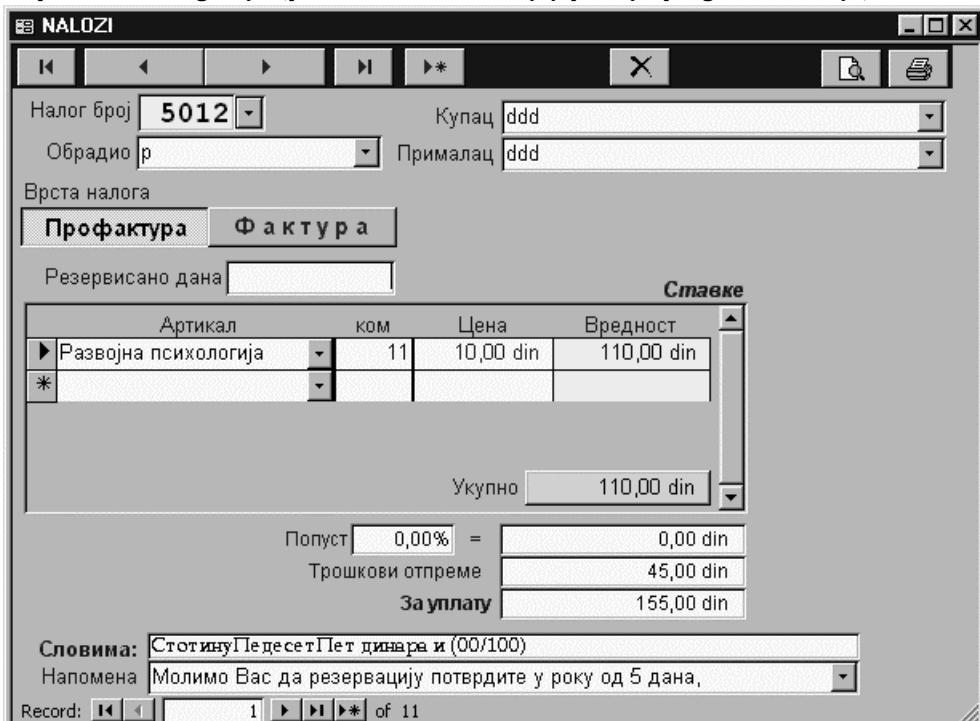


pravougaonik na mestu gde u glavnoj formi želite da smestite podformu, MS Access će pokrenuti Wizard-a. Ovde najpre treba odabratit formu (koja će biti podforma glavne forme), a zatim i polje po kojem će glavna *Master* biti povezan sa podformom (*Child*).

U našem slučaju (primer sa prethodne slike): Glavna forma je 'Dostavnice', podforma je 'DostavniceStavke', a polje po kojem se vezuju je 'DostavnicelID'.

Programiranje forme^①

Forme u MS Access su *Event-Driven* objekti što znači da se izvršavaju "vođene događajima". Da bi se programiralo na najlakši način, dovoljno je upoznati događaje (*Events*), bar malo VBA programiranje i upustiti se u kreiranje MS Access aplikacija, jer su baš forme - osnovni objekti aplikacije. Svaka akcija korisnika nad formom i/ili njenim grafičkim elementima pokreće neku proceduru događaja (pod uslovom da u njoj postoji programske linije) i izvršava je. Pisanje ovih



procedura (u VBA) obavlja se u prozoru kôda forme, a koji se poziva dvostrukim klikom na stavke *Event Procedure* u okviru pravila za polja i objekte dizajn moda forme.

Kodiranju (programiranju) se može pristupiti na više načina:

Expression Builder za kreiranje izraza ili kriterijuma u raznim vrstama polja uz korišćenje seta logičkih operatora kao i ugrađenih funkcija (samo za početak i upoznavanje objekata, svojstava i funkcija).

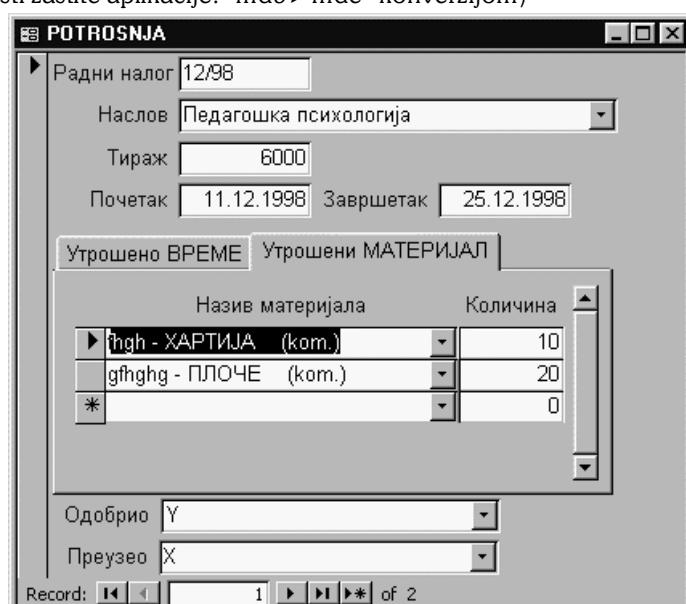
Macro Builder za kreiranje novog makroa

(prevaziđen način zbog nemogućnosti zaštite aplikacije: "mdb ► mde" konverzijom)

Code Builder za ulazak u modul kôda vezan za formu i pisanje VBA koda (procedura i funkcija).

Iz domena *Event-Driven* objekata i programiranja mi ćemo iskoristiti *Wizard-a* da kreiramo nekoliko komandnih tastera na formu (koju smo već kreirali), a preko kojih ćemo otvarati druge objekte (za sada neku formu, a kaniće i neki od izveštaja).

VEŽBA: Za tabele, koje ste već kreirali u vašoj bazi podataka, sada kreirajte i pojedinačne forme koristeći *Form Wizard*, a zatim ih redizajnirate u cilju podešavanja prema željama korisnika (konsultujte kolegu). U formama obavezno koristite *Subform*, *ComboBox*, *TabControl* tipove grafičko upravljačkih kontrola. U skriptama je prikazano nekoliko prostijih izgleda formi, pa možete da ih uzmete za primer.



13-14

11 IZVEŠTAJI

Reports (izveštaji) su objekti koji omogućuju pregled i štampanje traženih oblika lista i totala, od prostih do veoma složenih. Izveštaji su, po pristupu u dizajniranju, vrlo slični formama, ali se preko izveštaja ne omogućuje ažuriranje podataka od strane korisnika.

Kreiranje izveštaja

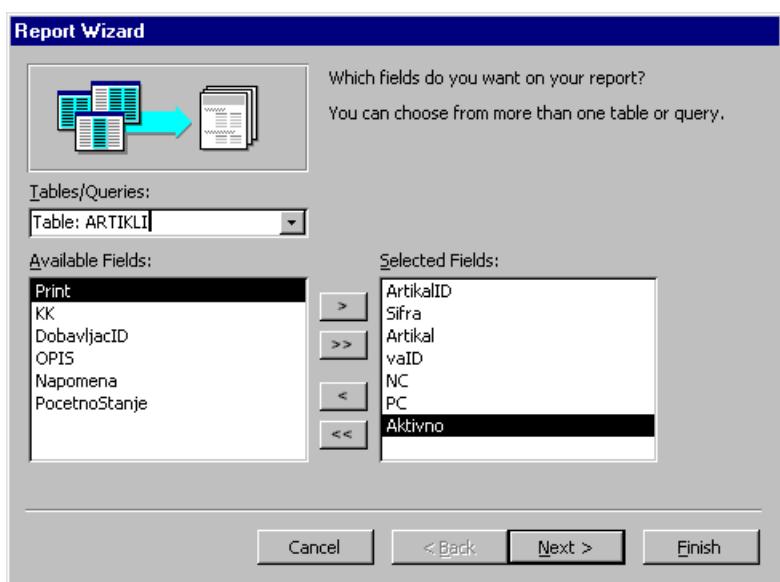
U okviru Database Window bira se komanda *New*, pri čemu se startuje sličan dijalog kao i u slučaju kreiranja forme. Za izveštaj, izvor podataka može biti tabela ili upit. Kod složenijih izveštaja načelo se koriste upiti, a kako *Report* ima velike mogućnosti grupisanja i kreiranja totala, mi ćemo za izvor uzimati postojeće tabele.

Pri kreiranju se nude sledeće opcije:

- *Design New*
- *Auto Report*
- *Report Wizard*
- *Chart Wizard*
- *Label Wizard*

Za razliku od *Form Wizard*-a, *Report Wizard* ima dodatne mogućnosti za grupisanje podataka u više nivoa, sortiranje po više ključeva, kao i za različite vrste sumiranja i primenu agregatnih funkcija. Na kraju se može izabrati jedan od nekoliko standardnih šablona za formatizovanje izveštaja. Dodatni dizajn izveštaja (podešavanja, doterivanja i usklađivanja), kao i kod formi, mogu se izvode se u *Design View*.

Label Wizard služi za kreiranje izveštaja oblika nalepnica i natpisa, a u ponudi se nalaze i mnogi poslovni i industrijski standardi. Ostavljena je mogućnost samostalnog podešavanja broja kolona po strani i veličine labela. Kroz ovaj *Wizard* moguće je podešavanje jedinice mere i vrste papira koja se ubacuje u štampač, izbor fonta, izbor polja za nalepnicu kao i određivanje po kom polju će ispis nalepnica biti sortiran.



Dizajn izveštaja

Pri dizajnu izveštaja, na *Toolbar*-u su na raspolaganju alati:

Auto Format Wizard (dostupna je i sa format menija) sadrži opcije za formatiranje postojećih izveštaja na nekoliko klasičnih načina. Svaki izveštaj se može kasnije doraditi u modu za dizajn, a mogu se kreirati i novi.

Sorting And Grouping (dostupna kroz *View* meni) je daleko najznačajniji alata - specifičan za izveštaje, a omogućava da se po svim sekcijama izveštaja postigne dodatno sortiranje i grupisanje podataka u više nivoa, kontrolisani prelom strana i kolona. *Field/Expression* omogućava biranje polja po kojima će se vršiti grupisanje, a *Sort Order* daje mogućnost biranja sortiranja po opadajućem ili rastućem redosledu.



Group Header	opcija zaglavlja grupe, pogodna za naslove ili imena polja.
Group Footer	zaglavljaju grupe, koristi se za međutotale i međurezultate.
Group On	određuje kako će se grupisanje vršiti. Moguće je izabrati jednu od dve vrednosti iz ComboBox-a: <i>Each Value</i> -grupisanje se vrši po svakoj vrednosti i <i>Prefix Characters</i> -grupisanje se vrši po prvih nekoliko karaktera u polju.
Group Interval	predstavlja broj prvih karaktera po kojima će se izvršiti grupisanje podatka.
Keep Together	opcija kojom se podešava da se, ukoliko je potrebno, elementi sekcijske grupe (<i>Header</i> , <i>Detail</i> , <i>Footer</i> ,...) drže zajedno na istoj strani izveštaja, a sa <i>No</i> se ovaj zahtev ne postavlja.
Whole Group	Obezbeđuje da svi delovi grupe budu odštampani na jednoj strani.
With First Detail	štampa se <i>Header</i> grupe sa najmanje jednim elementom <i>Detail</i> sekcijske grupe na istoj strani.

Page Setup

Za podešavanje opcija pri kreiranju izveštaja koristi se stavka *Page Setup* sa *File* menija. U *Page Setup* dijalogu podešavaju se parametri:

Margins za podešavanje marga. Opcija *Print Data Only* šalje na štampu samo podatke sa izveštaja, ali ne i njegov dizajn.

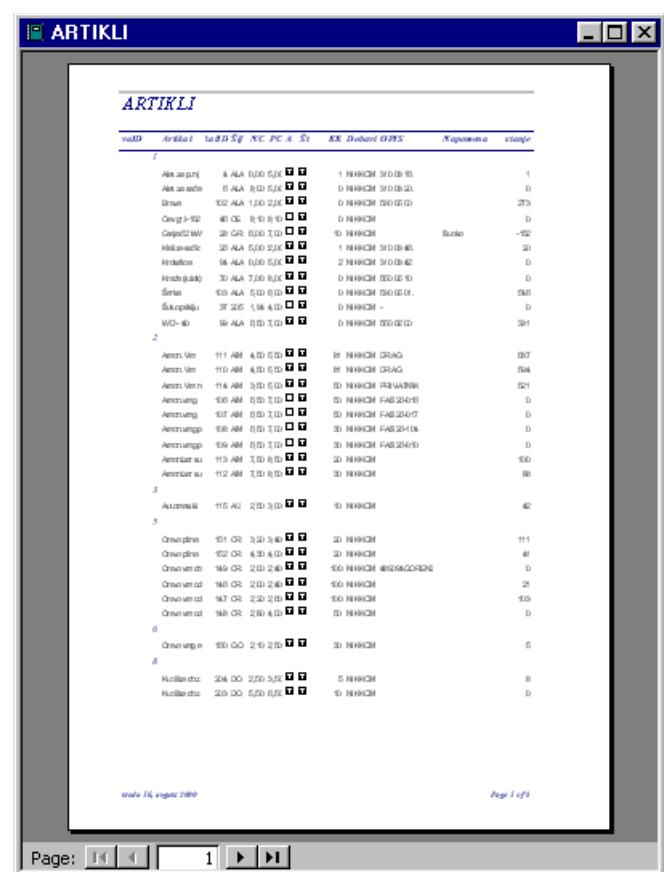
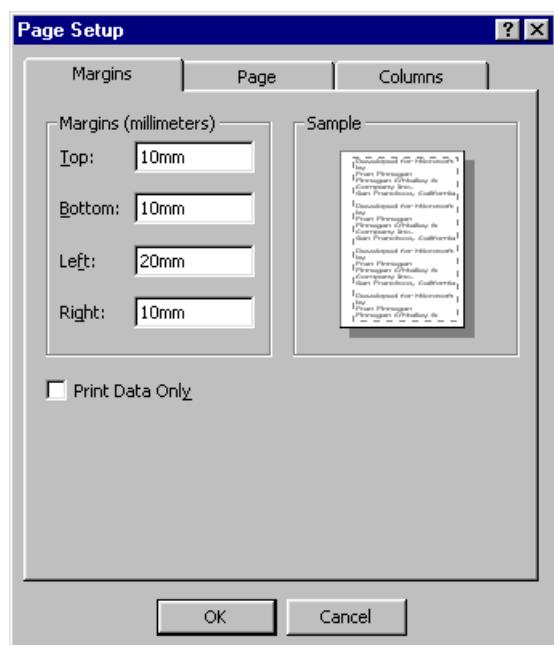
Page nudi izbor orijentacije (*Portrait* ili *Landscape*) strane, veličine papira i punjenja štampača, kao i izbor štampača.

Grid Settings podešava se broj kolona, razmak između kolona i razmak između redova

Item Size podešava se širina kolone i visina reda

Layout Columns redosled štampanja u slučaju više kolona.

U skriptima je (na delovima slobodnih strana) prikazano nekoliko tipova izveštaja.



Za pre-gled prikaza (pre ili bez štampanja) koriste se navigacioni tasteri u donjem levom uglu prozora izveštaja (slika desno), a print dijalog se pokreće preko <Ctrl> + <P> ili PopUp menua tekućeg izveštaja.

Za programiranje izveštaja potrebno je poznavanje VBA.

VEŽBA: Kreirajte izveštaje, slične prikazanim kroz ova skripta.

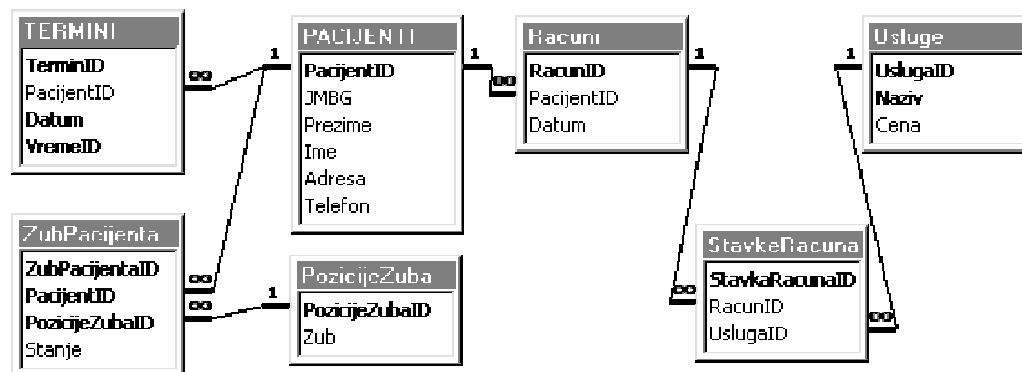
12 IZRADA STANDARDNE BAZE PODATAKA

Vaš zadatak, koji ste odabrali na sedmom terminu ovog kursa (bazu podataka), donećete na ovaj termin, izložiti relacioni model i prikazati kreirane objekte MS Access baze podataka. Ne zaboravite da Vaš fajl (*.mdb) obrišete po završenom prikazu, jer pravo na njega imate samo Vi.

Napravite bazu podataka koja će da sadrži podatke o nekim vašim poslovima, hobiju i slično. Baza treba da sadrži najmanje dve tabele u relaciji (savetuje se 5 do 10 tabela).

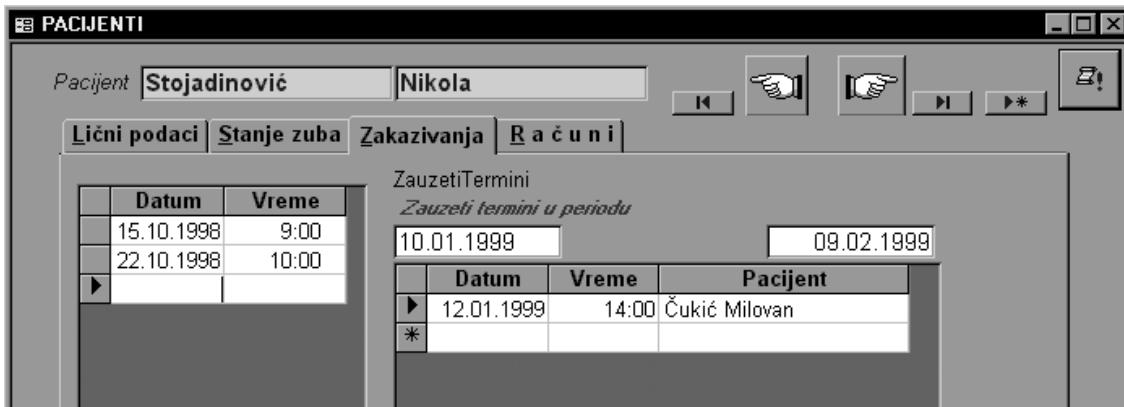
Primeri ovih tabela su na slikama dizajn moda ovih tabela (u nastavku teksta. Omogućiti unos podataka preko forme.

Koristiti prost relacioni model sličan sledećem:



Račun				
Pacijent	Račun br.	Datum	Usluga	Cena
Čukić Milovan	001	12.10.1998		
		Plombiranje		78,00 Din.
		Čišćenje kamenca		60,00 Din.
		Bušenje		5,00 Din.
			Ukupno:	143,00 Din.
	003	12.10.1998		
		Poliranje		26,00 Din.
		Čišćenje kamenca		60,00 Din.
			Ukupno:	86,00 Din.
			Ukupno:	229,00 Din.

U formama treba da bude omogućeno pretraživanje i manipulaciju sloganima baze po ugledu na sledeću:



Na kraju kreirati izveštaje (na osnovu prethodno kreiranog upita), koji će da sadrži grupisanja isumiranja, kao npr.:

Izaz po otprenimicama				
Dopravnik	Datum	Radnik	Aktivnost	Stanje
1	19.01.1998	10010	Rado	1
2	19.01.1998	10010	Analistički rad	10
3	11.01.1998	10010	Rado	6
4	11.01.1998	10010	Rado	3
5	11.01.1998	10010	Vlastimir Walter	1
6	11.01.1998	10010	Vlastimir Walter	1
7	11.01.1998	10010	Analistički rad	15
			Kabine odstupak imenice	+0
			Slanje	+0
			Anal.PTC	+0
			Anal.	+0
			Rado A5	+0
			Punkti odstupak imenice	1
8	11.01.1998	10010	Tiskarski uvezak	10
9	11.01.1998	10010	Vlastimir Walter	1
10	11.01.1998	10010	Punkti odstupak imenice	1
11	11.01.1998	10010	Tiskarski uvezak	10
12	11.01.1998	10010	Analistički rad	1
13	11.01.1998	10010	Punkti odstupak imenice	+0
14	11.01.1998	10045	Punkti odstupak imenice	15
15	11.01.1998	10110	Spolaz	1
16	11.01.1998	10101	Analistički rad	10
17	11.01.1998	10101	Vlastimir Walter	1
18	11.01.1998	10095	Vlastimir	1
19	11.01.1998	10095	Vlastimir	1
20	11.01.1998	10010	Vlastimir	1
21	15.01.1998	10010	Analistički rad	10
			Anal rad jeftinije	1
			Rado A5	+0
			Rado A5	+0
			Anal.	+0
			Anal.	+0
22	15.01.1998	10010	Analistički rad	10
			Anal rad jeftinije	1
23	11.01.1998	10010	Registar avanski	1
			Registar avanski	1
			Punkti odstupak	100

Makarne cene klijentih artikala				
Artikal	Utečajnici	Univer	Cena	Informacije
Blok A4				
	Srpska	06.01.1998	0,00 Dld.	
Faseta klas mehaničkih	Jedan	08.01.1998	7,10 Dld. (DII) 603-122	
Fotija sa triptikom	Jedan	16.01.1998	1,20 Dld. (DII) 603-122	
Jedan	18.01.1998	7,35 Dld. (DII) 603-122		
Fotokopir papir A4	Jedan	08.01.1998	63,00 Dld. (DII) 603-122	
Jedan	11.01.1998	63,00 Dld. (DII) 603-122		
Velikost	21.01.1998	49,00 Dld.		
Futrol za učenje	DP Pap.Foto	14.01.1998	33,00 Dld.	
Koverete u bojima	Jedan	16.01.1998	0,48 Dld. (DII) 603-122	
Srpska	29.01.1998	0,99 Dld.		
Srpska	08.01.1998	0,99 Dld.		
Koverete u bojivitke	Srpska	29.01.1998	1,37 Dld.	
Srpska	08.01.1998	1,20 Dld.		
Lepenke pisa	Zeml	06.01.1998	0,01 Dld.	
Mag-lite 1A baterijska lampja	Glosga	24.01.1998	264,00 Dld.	
Mag-lite 3D baterijska lampja	Glosga	24.01.1998	499,00 Dld.	
Obrazec I-II	Grafobasko	13.01.1998	1,00 Dld.	
Jedan	29.01.1998	465,00 Dld. (DII) 603-122		
Startek I	Zeml	07.01.1998	5,38 Dld.	
Tabolit I+II	Jedan	17.01.1998	243,00 Dld. (DII) 603-122	
Tabolit I+II	Jedan	17.01.1998	348,00 Dld. (DII) 603-122	
Tečnost za stolove	Papirus	11.01.1998	22,00 Dld. (DII) 3226-467	

Čuvajte Vaše radeve,

poštujte dela drugih i razmenjujte znanje sa onima koji imaju šta da Vam pruže,
trudeći se da, ovaj posao, svakim danom radite sve bolje,
jer će Vas u tom slučaju mnogi tražiti, a svi poštovati.

13 DODATAK

FAQs (Frequently Asked Questions)

Najčešća pitanja, na teme koje smo obradili u ovim skriptama, su data prema njihovoj čestalosti (broju pojavljivanja) na predavanjima, Internet konferencijama i razmeni iskustava sa drugima. Na pitanja je odgovorenno kratko, bez suvišnih komentara, a za detalje možete konsultovati autora.

Da li se od MS Access aplikacije može kreirati izvršni program?

Najkraći odgovor bi bio NE, ali ako zakupite prava na komplet razvojnih alata (npr *MS Access Developer Tools Kit*), biće Vam omogućeno da kreirate instalacionu verziju vašeg projekta i instalirate je kod korisnika koji nema obavezu da instalira MS Access.

Da li se podaci u bazi mogu u potpunosti zaštiti?

Odgovor bi bio DA, ali ga kvari mali procenat (oko 10%) mogućnosti da se, i potpuno zaštićena, baza razbije. I drugi alati za projektovanje baza podataka imaju ove probleme, jer se mnogi vrhunski programeri bave upravo ovom oblasti - razbijanje sistema zaštite, pa se događa i nelegalna distribucija algoritma zaštite (ključ zaštite).

Kako je ovo kompleksna oblast, najpre treba razlikovati oblike zaštite podataka: zaštita podataka od pregleda i zaštita distribucije podataka, pa ako se obezbedi prvi oblik, onda će biti obezbđen i drugi. Pristup bazi se najbolje rešava definisanjem korisničkih grupa i dozvola, što podrazumeva dobro poznavanje ovog alata, šifre korisnika i (uglavnom) rad u lokalnoj računarskoj mreži.

Znakovna konverzija podataka je oduvek bila dobar način da se zaštite podaci u bazama koje se distribuiraju uz komercijalne aplikacije. Pod pretpostavkom da je primenjen dobar model konverzije (oblika polinoma) i rekonverzije, mala je verovatnoća da bude razbijena. Ako baza bude i otvorena, za onoga koji želi podatke tek predstoji posao na mukotrpnom razbijanju algoritma konverzije, što se najčešće završava neuspehom.

Numerička konverzija ima brojne prednosti nad tekstualnom, jer daje mogućnosti kompleksnijeg algoritma i bržeg odvijanja konverzije, pa čak i primenu regresione analize polinomom. Bez obzira na tip i algoritam konverzije, treba imati u vidu da kompleksna konverzija usporava SQL, a da se jednostavna brzo razbije. Odlučiti se za neki algoritam nije lako, a velike kompanije obično zapošljavaju više poznavaca u ovoj oblasti kada treba da zaštite bazu u distribuciji.

Da li se aplikacija može zaštiti?

DA, kreiranjem mde fajla štiti se dizajn objekata tipa forme, izveštaji i moduli koda.

Da li se Access aplikacija (projekat) može prevesti u Visual Basic projekat?

Alati za ove zadatke postoje, a svi su na nivou DEMO AddIn aplikacija za MS Access. Problemi ovih dodataka su brojni i uglavnom vezani za *References* i imena objekata u *MS Access*-u (ver 9) i *VisualBasic*-u (ver 6).

SPECIFIKACIJE ZA MS Access 97/2000

U ovom dodatku data su ograničenja koja važe u MS Access-bazama podataka i objektima (tabelama, upitima, obrazcima, izveštajima i makroima).

Baze podataka

atribut	maksimalna vrednost
veličina MDB datoteke	2 Gb, Pošto baza podataka može da imaLink-om vezane tabele iz više datoteka, njena ukupna veličina ograničena je samo raspoloživim prostorom na spoljnim memorijskim jedinicama
broj objekata u bazi podataka	32768
broj modula	1000
broj znakova u imenima objekata	64
broj znakova za lozinku	14
broj znakova za korisničko ime ili ime grupe	20
broj istovremenih korisnika	255

Tabele

atribut	maksimalna vrednost
broj znakova u imenu tabele	64
broj znakova u imenu polja	64
broj polja u zapisu tabele	255
broj otvorenih tabela	2048 uključujući i tabele koje MS Access interno otvara
veličina tabele	1 GB
broj znakova u polju za tekst	255
broj znakova u Memo polju	65535 kada se podaci unose putem korisničkog interfejsa; 1 GB pri programskom unošenju podataka
veličina polja tipa OLE objekat	1 GB
broj indeksa po tabeli	32
broj polja u indeksu	10
broj znakova u poruci o pogrešno unetom podatu-	255
ku	
broj znakova u pravilu za proveru ispravnosti	2048
podataka	
broj znakova u opisu tabele ili polja	255
broj znakova u zapisu	2000 (bez polja tipa Memo i OLE objekat)
broj znakova koje može da sadrži svojstvo polja	255

Upiti

atribut	maksimalna vrednost
broj tabela u upitu	32
broj polja u skupu rezultata upita	255
veličina skupa rezultata upita	1 GB
ograničenje za sortiranje	255 znakova u jednom ili više polja
broj sortiranih polja po upitu	10
broj nivoa ugnježđenih upita	50
broj operatora AND u komandama WHERE i HAVING	40
broj znakova u SQL naredbi	64000

Obrasci i izveštaji

atribut	maksimalna vrednost
broj znakova u natpisu	2048
broj znakova u polju za tekst	65535
širina obrasca ili izveštaja	55,87 cm (22 inča)
visina odeljka	55,87 cm (22 inča)
visina svih odeljaka uvećana za zaglavla odeljaka (sekacija) u prikazu za dizajn	508 cm (200 inča)
broj nivoa ugnježdenih obrazaca ili izveštaja	3 (obrazac-podobrazac-podobrazac)
broj polja/izraza po kojima možete da sortirate ili grupišete	10 (samo u izveštajima)
broj zaglavlja ili podnožja u izveštaju	1 zaglavljepodnožje izveštaja; 1 zaglavljepodnožje stranice; 10 zaglavlja/podnožja grupe
broj odštampanih stranica izveštaja	65536
broj kontrolnih objekata ili odeljaka koje možete dodavati tokom životnog veka obrasca ili izveštaja	754

Makroi

atribut	maksimalna vrednost
broj akcija u makrou	999
broj znakova u uslovu	255
broj znakova u komentaru	255
broj znakova u argumentu akcije	255

Projekti u MS Access -u 2000

atribut	maksimalna vrednost
broj objekata u MS Access projektu (.adp)	32768
moduli (uključujući module obrazaca i izveštaja)	1000
broj znakova u imenu objekta	64

Tipovi podataka

Detalje o numeričkim tipovima podataka možete videti u sledećoj tabeli:

Tip podatka	Minimalna vrednost	Maximalna vrednost
Byte (celi brojevi)	0	255
Integer (celi brojevi)	-32.768	32.767
Long Integer (celi brojevi)	-2.147.483.648	2.147.483.647
Single (realni brojevi)	-3,402823E+38	3,402823E+38
Double (realni brojevi)	-1.79769313486231D+308	1.79769313486231D+308
Replication ID	16-bitni celi broj –GUID (globalni jedinstveni identifikator)	
Date/Time	datum i vreme za godine 100-9999	
Currency (valutni format)	maksimalno do 15 cifara u celobrojnem i 4 cifre u decimalnom delu	
Auto Number (automatski numerator)	jedinstveni sekvensijalni ili proizvoljan broj dužine 4 (ili 16 bita ako se koristi kao ReplicationID)	
Yes/No (logička promenljiva)	polje dužine 1 bit koje sadrži vrednost True/False, On/Off, -1/0	

TASTERSKE PREČICE

Globalne tasterske pre čice

Upravljanje fajlovima

Kreiranje nove baze	Ctrl+N
Otvaranje neke od postojećih baza	Ctrl+O

Štampanje i čuvanje

Za štampanje tekućeg objekta	Ctrl+P
Čuvanje objekta koji se trenutno dizajnira	Ctrl+S ili Shift+F12 ili Alt+Shift+F2
Za čuvanje ili Export objekta (Save As dialog)	F12 ili Alt+F2

Korišćenje ComboBox-a ili ListBox-a

Za Otvaranje ComboBox-a	F4 ili Alt+↓
Za osvežavanje sadržaja Lookup polja (ListBox ili ComboBox)	F9
Za pomeranje jednu liniju nadole	↓
Za pomeranje jednu stranu nadole	Page Down
Za pomeranje jednu liniju na gore	↑
Za pomeranje jednu stranu nagore	Page Up
Za izlaz iz objekta	Tab

Pretraživanje i zamena teksta ili podataka

Otvaranje Find dijaloga (Datasheet view, Form view, Module Window)	Ctrl+F
Otvaranje Replace dijaloga (Datasheet view, Form view, Module Window)	Ctrl+H
Traženje sledećeg traženog teksta specificiranog u Find ili Replace dijalogu (Datasheet view i Form view)	Shift+F4

Rad u Design view

Za prelazak između edit i navigacionog režima	F2
Za fokusiranje na formu ili report objekat	Ctrl+R
Za prelazak u Form view iz Form design	F5
Za prelazak između gornjeg i donjeg dela prozora (Design view tabele, makroa, upita i Advanced Filter/Sort prozora)	F6
Za dodavanje kontrole u sekciju (Form i Report Design view)	Shift+Enter

Window operacije

Prelazak na Database window u fokusu	F11 ili Alt+F1
Kretanje kroz otvorene prozore objekata baze	Ctrl+F6
Prikaz u Control menija	Alt+Space
Zatvaranje aktivnog prozora	Ctrl+W ili Ctrl+F4

Razno

Za prikaz cele adrese selektovanog Hyperlink-a	F2
Za Check Spelling	F7
Za otvaranje Zoom box-a (unos izraza ili tekst a u posebnom Input prozoru)	Shift+F2
Za prikaz liste (prozora) svojstava dizajniranog grafičkog objekta u fokusu	Alt+Enter
Za izlaz iz MS Access-a, zatvaranje dijalog-a, ili zatvaranje liste (prozora) svojstava	Alt+F4
Za pozivanje Builder-a	Ctrl+F2
Za prebacivanje sa kreiranog menija u Built-u menija	Ctrl+F11

Kretanje u polju podatka

Prelazak za jedno mesto desno	→
Prelazak za jednu reč desno	Ctrl+→
Prelazak za jedno mesto levo	←
Prelazak za jednu reč levo	Ctrl+←
Prelazak na kraj reda, u jednolinijskom polju	End
Prelazak na kraj polja, u višelinijijskom tekstualnom polju	Ctrl+End
Prelazak na početak reda, u jednolinijskom polju	Home
Prelazak na početak polja, u višelinijijskom tekstualnom polju	Ctrl+Home

Prikaz sistema pomoći

Za prikaz Office Assistant-a, Contents i Index-a	F1
Za prikaz pomoći po principu "Šta je ovo?"	Shift+F1

Pregled u VBA kôdu

Prikaz Object Browser-a	F2
Prikaz dijalog box-a Calls	Ctrl+L
Pregled deklaracije procedure, promenljive ili ključne reči	Shift+F2
Povratak u poslednju liniju prethodne pregledane procedure	Ctrl+Shift+F2
Prikaz prethodne procedure	Ctrl+↑
Prikaz sledeće procedure	Ctrl+↓
Prikaz Context-nog Help-a o tekućem objektu	F1
Prebacivanje fokusa na Object Box	Ctrl+F2
Prebacivanje fokusa na Procedure Box	Ctrl+F2 (zatim Tab)
Lista svojstava i metoda	Ctrl+J
Lista konstanti	Ctrl+Shift+J
Izbor selektovane stavke iz liste	Ctrl+Enter ili Tab
Izbor selektovane stavke iz liste i prelazak u sledeću liniju	Enter
Zatvaranje liste svojstava i metoda	ESC
Prikaz kratkih informacija	Ctrl+I
Prikaz informacija o parametrima	Ctrl+Shift+I
Kompletiranje započete ključne reči Visual Basic-a	Ctrl+Space

Rad u Visual Basic kôdu

Selektovanje sveg vidljivog kôda u prozoru modula	Ctrl+A
Traženje sledećeg slučaja traženog teksta kada je dijalog zatvoren	F3
Traženje prethodnog slučaja traženog teksta kada je dijalog zatvoren	Shift+F3
Traženje sledećeg slučaja selektovanog teksta	Ctrl+F3
Uvlačenje selektovanih redova	Tab ili Ctrl+M
Izvlačenje selektovanih redova	Shift+Tab ili Ctrl+Shift+M
Cut za tekući red (!)	Ctrl+Y

Run i debug Visual Basic kôda

Prikaz Debug prozora	Ctrl+G
Za pokretanje tekuće procedure (prozor modula)	F5
Prelazak između panela Debug prozora	F6
Izvršavanje u malim koracima (single step)	F8
Zaustavljanje izvršavanja iznad (ispred) procedure	Shift+F8
Zaustavljanje izvršavanja iza (ispod) procedure	Ctrl+Shift+F8
Pokretanje do tekuće programske linije	Ctrl+F8
Kreiranje trenutnog pregleda vrednosti selektovanog izraza	Shift+F9
Kontrolisanje prekidne tačke na tekućoj liniji	F9
Uklanjanje svih uvedenih prekidnih tačaka	Ctrl+Shift+F9
Pozicioniranje sledeću instrukciju (komandu)	Ctrl+F9
Zaustavljanje izvršavanja kôda ili macro-a	Ctrl+BREAK
Nastavak izvršavanja kôda ili macro-a	F5
Resetovanje (ponovno pokretanje) izvršavanja kôda ili macro-a	Shift+F5
Kontrolisanje prekidnih i neprekidnih grešaka koje nisu servisirane i izvršavanje do slijedeće instrukcije	Alt+F5
Kontrolisanje prekidnih i neprekidnih grešaka u Class modulima i nastavak izvršavanja	Alt+F8

Kopiranje, prebacivanje ili brisanje teksta

Kopiranje selekcije u Clipboard	Ctrl+C (Copy)
Brisanje i kopiranje selekcije u Clipboard	Ctrl+X (Cut)
Lepljenje sadržaja iz Clipboard-a na tekuću poziciju	Ctrl+V (Paste)
Brisanje selektovanog ili jednog karaktera na levo	Backspace
Brisanje selektovanog ili jednog karaktera na desno (!)	DEL

Poništiti poslednju izmenu

Poništavanje poslednjeg unosa u polju	Ctrl+Z ili Alt+ Backspace
Poništavanje izmena u tekućem polju ako je bilo promena	ESC
Poništavanje izmena u tekućem slogu	ESC, ESC

Unos podataka u tabele i forme

Ubacivanje tekućeg datuma	Ctrl+SEMICOLON (;)
Ubacivanje tekućeg vremena	Ctrl+COLON (:)
Ubacivanje podrazumevane vrednost za polje	Ctrl+Alt+Space
Ubacivanje vrednosti iz istog polja iz prethodnog sloga	Ctrl+APOSTROPHE (')
Dodavanje novog sloga	Ctrl+PLUS znak (+)
Brisanje tekućeg sloga	Ctrl+MINUS znak (-)
Čuvanje promena u tekućem slogu	Shift+Enter
Prelazak između opcija u Check Box ili Option Button	Space
Ubacivanje nove linije (u TextBox-u)	Ctrl+Enter

Osvežavanje polja sa trenutnim podacima

Rekalkulaciju polja u tekućem prozoru	F9
Osvežavanje upita povezanih tabela (u podformama)	Shift+F9
Osvežavanje sadržaja ili Lookup polja u ListBox ili ComboBox	F9

Prozori i dijalozi**Upravljanje prozorima**

Prelazak na sledeći aktivan program	Alt+Tab
Prelazak na prethodni aktivan program	Alt+Shift+Tab
Prikaz prozora Windows Start menu	Ctrl+ESC
Zatvaranje prozora tekućeg objekta (dokumenta)	Ctrl+W ili Ctrl+F4
Prelazak u sledeći prozor otvorenog objekta (dokumenta)	Ctrl+F6
Prelazak u prethodni prozor otvorenog objekta (dokumenta)	Ctrl+Shift+F6
Izbor foldera u Open ili Save As dijalogu	Alt+0 za selektovanje folder liste; kursorski tasteri za selektovanje foldera
Izbor tastera sa Toolbar-a u Open ili Save As dijalogu	Alt+ broj (1 za prvi, 2 za drugi, itd.)
Refresh liste foldera i fajlova u Open ili Save As dijalogu	F5

Korišćenje Dijalog Box-a

Prelazak u sledeću karticu dijalog-a	Ctrl+Tab ili Ctrl+Page Down
Prelazak u prethodnu karticu dijalog-a	Ctrl+Shift+Tab ili Ctrl+Page Up
Prelazak na sledeću stavku dijalog-a ili grupu grafčkih kontrola	Tab
Prelazak na prethodnu stavku dijalog-a ili grupu grafičkih kontrola	Shift+Tab
Kretanje kroz padajuću listu ili između opcija u grupi opcija	kursorski tasteri
Izvršiti akciju pritiska na fokusirani komandni taster; selektovanje ili brisanje u CheckBox ili opcionom dugmetu	Space
Prelazak na stavku po prvom slovu u listi ili folder prozoru	Taster prvog slova stavke koju želite (kada je padajuća lista u fokusu)
Izbor stavke ili opcije po podvučenom (HotKey) slovu	Alt+ slovni taster
Otvoriti padajuću listu	Alt+↓ (kada je padajuća lista u fokusu)
Zatvoriti padajuću listu	ESC (kada je padajuća lista u fokusu)
Izvršiti akciju pritiska na default taster	Enter
Otkazati sve opcije i zatvoriti dijalog	ESC

Unos kroz TextBox

Pozicioniranje na početak	Home
Pozicioniranje na kraj	End
Prelazak za jedno mesto uлево ili уdesno	← ili →
Prelazak za jednu reč uлево или уdesno	Ctrl+← ili Ctrl+→
Selektovanje do početka	Shift+Home
Selektovanje do kraja	Shift+End
Selektovanje ili deseletovanje jednog karaktera uлево	Shift+←
Selektovanje ili deseletovanje jednog karaktera уdesno	Shift+→
Selektovanje ili deseletovanje jedne reči uлево	Ctrl+Shift+←
Selektovanje ili deseletovanje jedne reči уdesno	Ctrl+Shift+→

Navigacija u Datasheet view

Odlazak na specificirani slog

Prelazak na specifikaciju traženog sloga (zatim otkucajte broj sloga i pritisnite Enter)	F5
--	----

Navigacija kroz polja i slogove

Prelazak na sledeće polje	Tab, Enter, ili →
Prelazak na prethodno polje tekućeg sloga	Shift+ Tab, ili ←
Prelazak na poslednje polje tekućeg sloga (u navigacionom režimu)	End
Prelazak na prvo polje tekućeg sloga (u navigacionom režimu)	Home
Prelazak u tekuće polje sledećeg sloga	↓
Prelazak na tekuće polje poslednjeg sloga (u navigacionom režimu)	Ctrl+↓
Prelazak na poslednje polje poslednjeg sloga (u navigacionom režimu)	Ctrl+ End
Prelazak na tekuće polje prethodnog sloga	↑
Prelazak na tekuće polje prvog sloga (u navigacionom režimu)	Ctrl+↑
Prelazak na prvo polje prvog sloga (u navigacionom režimu)	Ctrl+ Home

Navigacija kroz delove ekrana

Prelazak na donji deo ekrana	Page Down
Prelazak na gornji deo ekrana	Page Up
Prelazak na desni deo ekrana	Ctrl+ Page Down
Prelazak na levi deo ekrana	Ctrl+ Page Up

Prebacivanje kolona

Uključivanje Move moda (MOV uključen na desnoj strani statusne linije)	Ctrl+ F8
Prebacivanje selektovane kolone u desno za jedno mesto	→
Prebacivanje selektovane kolone uлево за jedno mesto	←
Isključivanje Move moda	ESC

Meniji

Korišćenje menija

Prikaz Shortcut (Pop Up) menija	Shift+ F10
Aktiviranje meni linije	F10
Prikaz Program Icon menija (u naslovnoj traci)	Alt+ Space
Izbor sledeće ili prethodne stavke u meniju	↓ ili ↑
Izbor menija uлево или udesno; ili prelazak u podmeni (ako postoji)	← ili →
Prelazak u prvu ili poslednju stavku menija	Home ili End
Zatvaranje otvorenog menija i podmenija u istom trenutku	Alt
Zatvaranje otvorenog menija; ili samo podmenija (ako je aktivan)	ESC

Navigacija u Form view

Odlazak na specificirani slog

Prelazak na specifikaciju traženog sloga (zatim otkucajte broj sloga i pritisnite Enter)	F5
---	----

Navigacija kroz polja i slogove

Prelazak na sledeće polje	Tab
Prelazak na prethodno polje	Shift+ Tab
Prelazak na poslednje polje tekućeg sloga (u navigacionom režimu)	End
Prelazak na poslednje polje poslednjeg sloga (u navigacionom režimu)	Ctrl+ End
Prelazak na prvo polje tekućeg sloga (u navigacionom režimu)	Home
Prelazak na prvo polje prvog sloga (u navigacionom režimu)	Ctrl+ Home
Prelazak u tekuće polje sledećeg sloga	Ctrl+ Page Down
Prelazak na tekuće polje prethodnog sloga	Ctrl+ Page Up

Navigacija kroz sekcije sloga

Unapred kroz sekcije	F6
Unazad kroz sekcije	Shift+ F6

Navigacija u formama sa više strana

Odlazak jednu stranu dole; na kraju sloga, prebacuje na ekvivalentnu stranu sledećeg sloga	Page Down
Odlazak jednu stranu gore; na početku sloga, prebacuje na ekvivalentnu stranu prethodnog sloga	Page Up

Navigacija između glavne forme i podforme

Prelazak iz radne forme u podformu	Tab
Prelazak iz podforme u sledeće polje radne forme	Shift+ Tab
Izlaz iz podforme i prelazak u sledeće polje ili sledeći slogmaster forme	Ctrl+ Tab
Izlaz iz podforme i prelazak u prethodno polje ili prethodni slog master forme	Ctrl+ Shift+ Tab

Selektovanje teksta i podataka**Selektovanje teksta u polju**

Proširenje za jedan karakter udesno	Shift+→
Proširenje za jednu reč udesno	Ctrl+Shift+→
Proširenje za jedan karakter uлево	Shift+←
Proširenje za jednu reč uлево	Ctrl+Shift+←

Napomena: Za otkaz selekcije, koristiti cursorske strelice (arrow keys).

Selektovanje polja ili sloga

Selektovanje sledećeg polja	Tab
Promena režima rada (<i>Edit</i> režim <-> <i>Navigacioni</i> režim)	F2
Prelazak između selektovanog prvog polja i selektovanog tekućeg sloga (<i>navigacioni</i> režim)	Shift+Space
Proširenje selekcije u prethodni slog, ako je tekući slog selektovan)	Shift+↑
Proširenje selekcije u sledeći slog, ako je tekući slog selektovan)	Shift+↓
Selektovanje svih slogova	Ctrl+A ili Ctrl+Shift+Space

Proširivanje selekcije

Uključivanje <i>Extend</i> moda (EXT u desnoj strani statusne linije); pritisnuti F8 više puta za proširenje selekcije na reč»polje» slog (<i>Datasheet view</i>), i svih slogova	F8
Proširenje selekcije za jedno polje u istom redu (<i>Datasheet view</i>)	←, ili →
Proširenje selekcije za jedan red (<i>Datasheet view</i>)	↑, ↓
Undo prethodnog proširenja selekcije	Shift+F8
Otkaz <i>Extend</i> moda	ESC

Selektovanje kolona u *Datasheet view*

Selektovanje tekuće kolone ili otkaz selekcije (navigacioni režim)	Ctrl+Space
proširenje selekcije udesno, kada je tekuća kolona selektovana	→
proširenje selekcije uлево, kada je tekuća kolona selektovana	←

Ažuriranje teksta i podataka**Kretanje kroz polje**

Za jedan karakter udesno	→
Za jednu reč udesno	Ctrl+→
Za jedan karakter uлево	←
Za jednu reč uлево	Ctrl+←
Prelazak na kraj polja (u jenolinijskom polju)	End
Prelazak na kraj polja (u višeliinijskom polju)	Ctrl+End
Prelazak na početak polja (u jenolinijskom polju)	Home
Prelazak na početak polja (u višeliinijskom polju)	Ctrl+Home

Napomena: Ako pozicija kursora nije vidljiva, pritisnite F2 za prikaz, tj. prelazak iz navigacionog u edit režim (važi i bratno).

Kopiranje, prebacivanje ili brisanje teksta

Kopiranje u Clipboard (Copy)	Ctrl+C
Isecanje selekcije i kopiranje u Clipboard (Cut)	Ctrl+X
Lepljenje sadržaja Clipboard-a na tekuću poziciju cursora (Paste)	Ctrl+V
Brisanje selekcije ili jednog karaktera uлево od tekuće pozicije cursora	BACK (Backspace)
Brisanje selekcije ili jednog karaktera udesno od tekuće pozicije cursora	DEL (Delete)

Poništavanje promena

Undo poslednjeg kucanja (unosa)	Ctrl+Z ili Alt+Backspace
Undo promena u tekućem polju ili tekućem slogu; prvo ESC poništava promene u polju, dok drugo poništava promene u tekućem slogu	ESC

Unos podataka kroz Datasheet ili Form view

Ubacivanje tekućeg datuma	Ctrl+SEMICOLON (:)
Ubacivanje tekućeg vremena	Ctrl+COLON (:)
Ubacivanje default vrednosti polja	Ctrl+Alt+Space
Ubacivanje vrednosti iz istog polja prethodnog sloga	Ctrl+APOSTROPHE (')
Dodavanje novog sloga	Ctrl+PLUS znak (+)
Brisanje tekućeg sloga	Ctrl+MINUS znak (-)
Čuvanje promena u tekućem slogu (neobavezno)	Shift+Enter
Promena izbora statusa (vrednosti) u Check Box ili Option Button	Space
Ubacivanje nove linije	Ctrl+Enter

Refresh

Rekalkulacija polja u prozoru	F9
Requery povezanih tabela; u podformi će biti osvežene samo tabele koje su vezane za nju	Shift+F9
Osvežavanje (Requery)sadržaja Lookup polja (ListBox ili ComboBox)	F9

Print Preview i Layout Preview**Dijalog box i prozori**

Print dijalog box	P ili Ctrl+P
Page Setup dijalog box	S
Zoom In ili Zoom Out dela strane	Z
Izlaz iz Print Preview ili Layout Preview	C ili ESC

Pregled drugih strana

Prelazak na specifikaciju traženog sloga (zatim otkucajte broj sloga i pritisnite Enter)	F5
Pregled sledeće strane (kada je izabran Zoom - Fit for Window)	Page Down ili ↓
Pregled prethodne strane (kada je izabran Zoom - Fit for Window)	Page Up ili ↑

Navigacija u Print i Layout Preview

Nadole za mali korak	↓
Nadole za ceo ekran	Page Down
Na dno strane	Ctrl+↓
Nagore za mali korak	↑
Nagore za ceo ekran	Page Up
Na vrh strane	Ctrl+↑
Udesno za mali korak	→
Desna ivica strane	End ili Ctrl+→
Donji desni ugao strane	Ctrl+End
Ulevo za mali korak	←
Leva ivica strane	Home ili Ctrl+←
Gornji levi ugao strane	Ctrl+Home

Moduli i Debug prozori

VBA kôd

Prikaz Object Browser	F2
Prikaz Calls dijalog box	Ctrl+L
Prikaz definicije procedure, promenljive ili ključne reči	Shift+F2
Povratak na poslednju liniju u prethodnoj prikazanoj proceduri	Ctrl+Shift+F2
Prikaz prethodne procedure	Ctrl+↑
Prikaz sledeći procedure	Ctrl+↓
Prikaz Context-nog Help-a o ključnoj reči u kojoj se nalazi kursor	F1
Fokusiranje na Object Box	Ctrl+F2
Fokusiranje na Procedure Box	Ctrl+F2 (zatim pritisnuti Tab)
Lista Properties i Methods	Ctrl+J
Lista konstanti	Ctrl+Shift+J
Unos selektovane stavke iz liste	Ctrl+Enter ili Tab
Unos selektovane stavke iz liste i prelazak kursorom na sledeću liniju	Enter
Zatvaranje liste	ESC
Prikaz brzih informacija	Ctrl+I
Prikaz informacija o parametru	Ctrl+Shift+I
Kompletiranje VBA ključne reči	Ctrl+Space

Unos VBA kôda

Selektovanje sadržaja aktivnog modula kôda	Ctrl+A
Traženje sledećeg specificiranog teksta, kada je Find ili Replace dijalog zatvoren	F3
Traženje prethodnog specificiranog teksta, kada je Find ili Replace dijalog zatvoren	Shift+F3
Traženje sledećeg selektovanog teksta	Ctrl+F3
Uvlačenje selektovanih linija	Tab ili Ctrl+M
Uklanjanje uvlačenja selektovanih linija	Shift+Tab ili Ctrl+Shift+M
Cut tekuće linije	Ctrl+Y

Run i debug VBA kôda

Prikaz Debug prozora	Ctrl+G
Pokretanje (Run) tekuće procedure, iz Modula kôda	F5
Kretanje kroz panele radnog prozora	F6
Izvršavanje procedure korak po korak	F8
Izvršavanje sledećeg koraka u proceduri	Shift+F8
Izvršavanje sledećeg koraka izvan procedure	Ctrl+Shift+F8
Izvršavanje kôda do tekuće pozicije i zaustavljanje na njoj	Ctrl+F8
Definisanje praćenja vrednosti selektovanog izraza	Shift+F9
Definisanje (on/off) prekidne linije (pauza u liniji kôda)	F9
Brisanje svih prekidnih linija	Ctrl+Shift+F9
Postavljanje na sledeću izvršnu instrukciju	Ctrl+F9
Prekidanje izvršavanja kôda ili makroa	Ctrl+BREAK
Nastavak izvršavanja kôda ili makroa	F5
Ponovno izvršavanje (reset) kôda ili makroa	Shift+F5
Isključivanje/uključivanje prekidne tačke na neservisiranim greškama	Alt+F5
Isključivanje/uključivanje prekidne tačke u Class modulu	Alt+F8

Toolbars

Korišćenje Toolbar-a

Aktiviranje menija	F10
Izbor sledećeg ili prethodnog Toolbar-a	Ctrl+Tab ili Ctrl+Shift+Tab
Izbor sledećeg ili prethodnog dugmeta ili menija na Toolbar-u	Tab ili Shift+Tab (kada je Toolbar aktivan)
Otvaranje menija	Enter (kada je meni fokusiran)
Pokretanje akcije pridružene dugmetu	Enter (kada je dugme fokusiran)
Enter text u a text box	Enter (kada je TextBox fokusiran)
Izbor opcije iz padajuće liste ili padajućeg menija dugmeta	Kursorski tasteri za kretanje kroz stavke i listi ili meniju; Enter za izbor tekuće stavke (kada je padajuća lista otvorena)

Database window

Ažuriranje i navigacija u listi objekata

Promena imena objekta	F2
Sledeća linija	↓
Prethodni ekran	Page Down
Poslednji objekat u listi	End
Prethodna linija	↑
Sledeći ekran	Page Up
Prvi objekat	Home

Navigacija i otvaranje

Prelazak u drugu grupu (tip) objekata	Ctrl+Tab (Shift+Ctrl+Tab)
Otvaranje tabele i upita u Datasheet view, forme u Form view mōdu	Enter
Otvaranje izveštaja u Print Preview-u	Enter
Pokretanje makroa	Enter
Otvaranje objekta u Design view mōdu	Ctrl+Enter ili Alt+D
Otvaranje svojstava (Properties) objekta	Alt + Enter (Alt+DoubleClick)

Office Assistant

Aktiviranje Office Assistant-a	Alt+F6 (ponavljati dok se ne aktivira)
Izbor teme iz Office Assistant-a	Alt+broj (1 za prvu, 2 za drugu, itd.)
See more Help tema	Alt+↓
Pregled prethodne teme	Alt+↑
Zatvaranje Office Assistant poruka	ESC
Help za Office Assistant-a	F1
Prikaz sledećeg saveta	Alt+N
Prikaz prethodnog saveta	Alt+B
Zatvaranje pregleda saveta	ESC