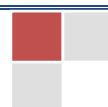


SADRŽAJ:

3 . MICROSOFT ACCESS 2007	3
3. 1. Objekti baze.....	3
Tabele.....	3
Upiti.....	4
Forme	4
Izveštaji.....	5
3. 2. Opis primera na kojem je priručnik baziran.....	5
3. 3. Pokretanje programa, pripreme radnje, kreiranje prve tabele.....	6
3. 4. Planiranje tabela	8
3. 5. Kreiranje prve tabele.....	9
3. 6. Povezivanje tabela	14
3. 7. Unošenje prvih zapisa	20
4. Kreiranje elektronskih obrazaca - formi.....	24
4. 1. Prva forma - Čitaoci.....	25



3. MICROSOFT ACCESS 2007

Access je sistem za upravljanje bazama podataka (data base management system DBMS), koristi se za skladištenje i učitavanje podatke, prikazivanje, a takodje i za automatizaciju poslova koji se često ponavljaju. Predstavlja odličnu kombinaciju produktivnosti programa za upravljanje bazom podataka i lakoće korišćenja Windows-a.

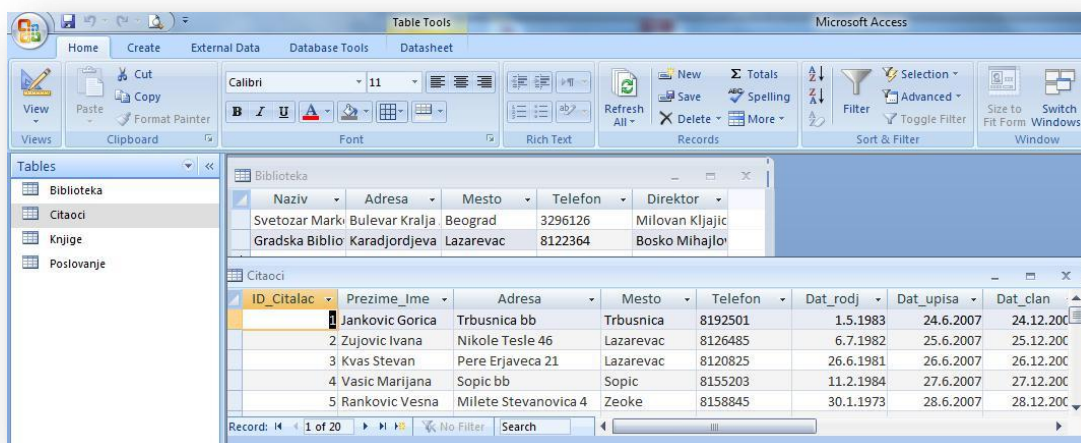
Kao zaokružena aplikacija, Access obezbeđuje potpuno upravljanje relacionom bazom podataka i integritet na nivou same mašine na kojoj se nalazi baza (što sprečava ne potpuno ažuriranje i brisanje). Tabelama je moguće pridružiti pravila koja proveravaju ispravnost i validnost podataka i sprečavaju unos neispravnih, a svako polje tabele može imati svoj format i podrazumevanu vrednost što znatno povećava produktivnost.

Omogućen je uvoz i izvoz podataka iz mnogih aplikacija: dBase, FoxPro, Excel, SQL Server, Oracle, Btrieve, tekstualni ASCII formati (polja sa fiksnom širinom i granicnicima), kao i podataka u HTML formatu. Access 2007 omogućava korišćenje svojih aplikacija za rad na Internetu. Izborom određene opcije omogućava se prebacivanje tabela, upita, izveštaja i prikaza u tabelarnom obliku u HTML format. Pomocu Data Access Pages (stranice za pristup podacima) mogu da se prave složeni obrasci na Internetu koji su povezani sa podacima u konkretnoj mreži.

3. 1. Objekti baze

Razvijena baza podataka u Accessu se sastoji od sledećih elemenata, tj. objekata: tabele, upiti, izveštaji; a može sadržati: Data Access strane, makroe i/ili VBA module.

Tabele



The screenshot shows the Microsoft Access 2007 interface. The 'Table Tools' ribbon is active, showing options for 'Table Tools', 'Database Tools', and 'Datasheet'. The 'Tables' pane on the left lists 'Biblioteka', 'Citaoci', 'Knjige', and 'Poslovanje'. The main window displays two tables:

Naziv	Adresa	Mesto	Telefon	Direktor
Svetozar Marki	Bulevar Kralja	Beograd	3296126	Milovan Kljajic
Gradska Biblio	Karadjordjeva	Lazarevac	8122364	Bosko Mihajlo

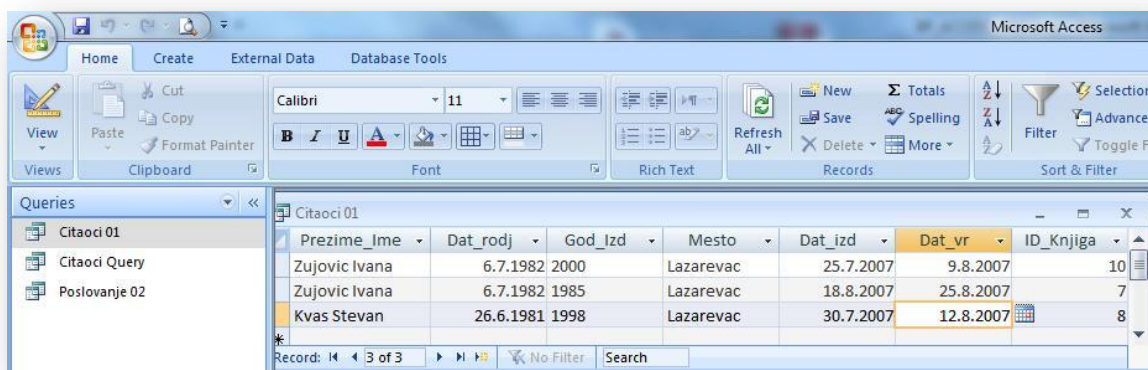
ID_Citalac	Prezime_Ime	Adresa	Mesto	Telefon	Dat_rodj	Dat_upisa	Dat_clan
1	Jankovic Gorica	Trbusnica bb	Trbusnica	8192501	1.5.1983	24.6.2007	24.12.2007
2	Zujovic Ivana	Nikole Tesle 46	Lazarevac	8126485	6.7.1982	25.6.2007	25.12.2007
3	Kvas Stevan	Pererjavca 21	Lazarevac	8120825	26.6.1981	26.6.2007	26.12.2007
4	Vasic Marijana	Sopic bb	Sopic	8155203	11.2.1984	27.6.2007	27.12.2007
5	Rankovic Vesna	Milete Stevanovica 4	Zeoke	8158845	30.1.1973	28.6.2007	28.12.2007

Slika 1

Tabele su nosioci podataka. Podaci se mogu nalaziti isključivo u tabelama. Svaka tabela se sastoji od kolona i redova, dok se presek kolone i reda naziva polje.

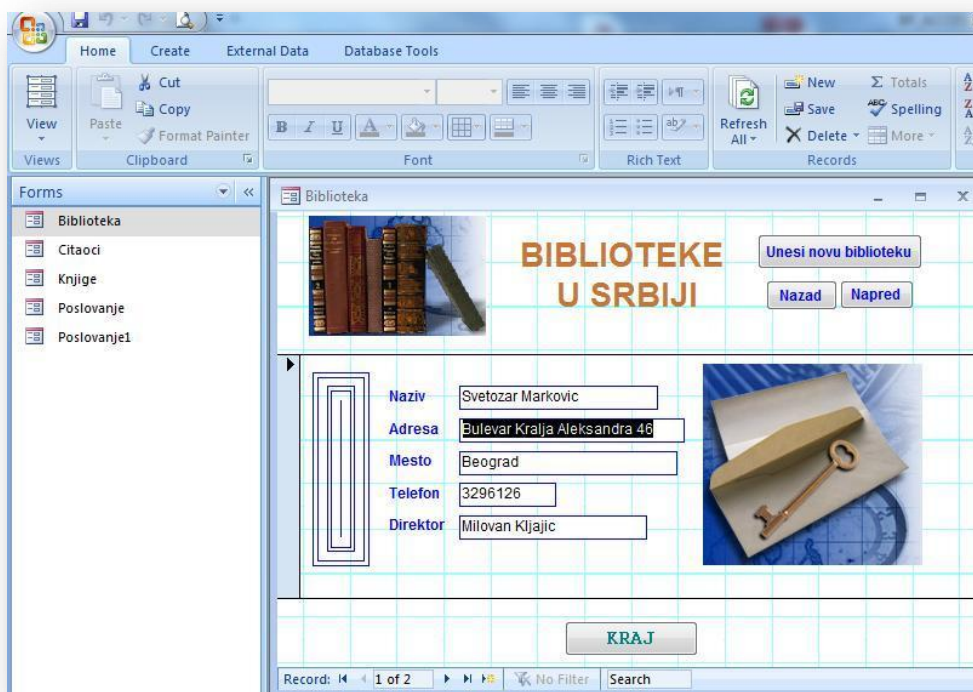
Upiti

Upiti su posebni pregledi kojima se postavlja neki upit nad tabelom i koji izdvajaju podatke za pregled na ekranu ili štampanje putem izveštaja.



Slika 2

Forme



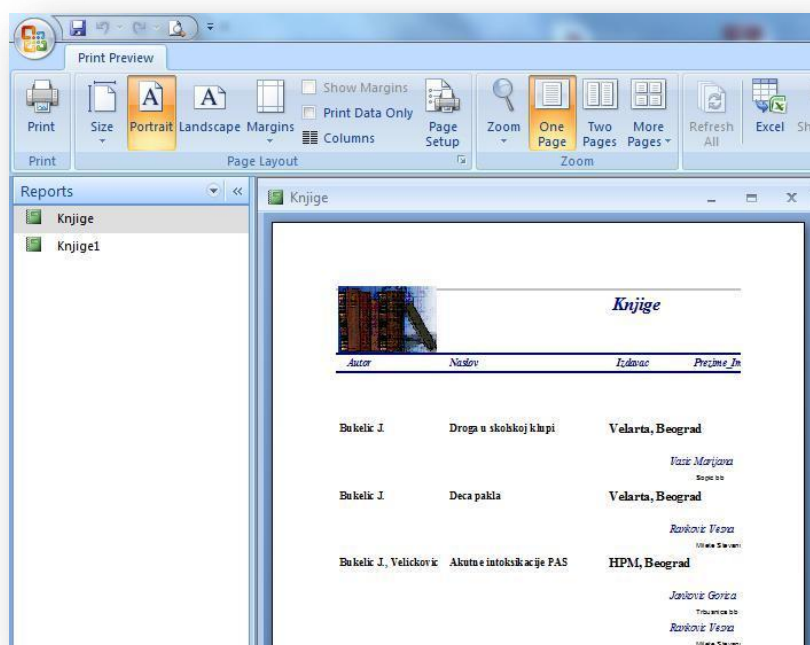
Slika 3

Forma je samo "pogled" na tabelu u čoveku prihvatljivijem izgledu. Nad formiranom tabelom koja sadrži podatke o nekom čitaocu formiramo masku za unos koja ima izgled papirnog obrasca - kartice, te manipulant ne mora da poznaje rad sa Accessom, nego samo koristi

elektronske obrasce (maske) i unosi podatke putem tastature, baš kao što je to ranije činio olovkom.

Izveštaji

Izveštaji su elektronski oblik klasičnih papirnih obrazaca koje možemo gledati na ekranu ili štampati na papir. Podaci prikazani u izveštaju mogu biti izvedeni iz tabela ili iz upita (filtrirani podaci). Dizajniranje izveštaja je veoma slično dizajniranju formi.



Slika 4

3. 2. Opis primera na kojem je priručnik baziran

Izgradnje baze počinje od osmišljavanja tabela i njihovog povezivanja. Za kreiranje programa **Biblioteka** za početak ćemo osmotriti celine - podatke koje treba grupisati u zasebnim tabelama.

Svaka biblioteka ima svoje čitaoce. Prva tabela će se stoga zvati **Čitaoci**.

Sama biblioteka ima svoje podatke kao što su adresa, telefon, ime direktora i slično. Druga tabela će se zvati **Biblioteka**.

Sama biblioteka ne bi bila to što jeste da nema knjiga. Za knjige je karakterističan inventarni broj, naslov knjige, autor itd. Treća tabela će se shodno tome zvati **Knjige**.

Interakcija između biblioteke i čitalaca se ogleda u izdavanju i vraćanju knjiga i plaćanju članarine. Četvrtu tabelu ćemo kreirati iz aktivnosti interakcije biblioteke i čitalaca (četvrta tabela - **Poslovanje**), a zapis o članarini ćemo ipak čuvati u tabeli **Čitaoci**.

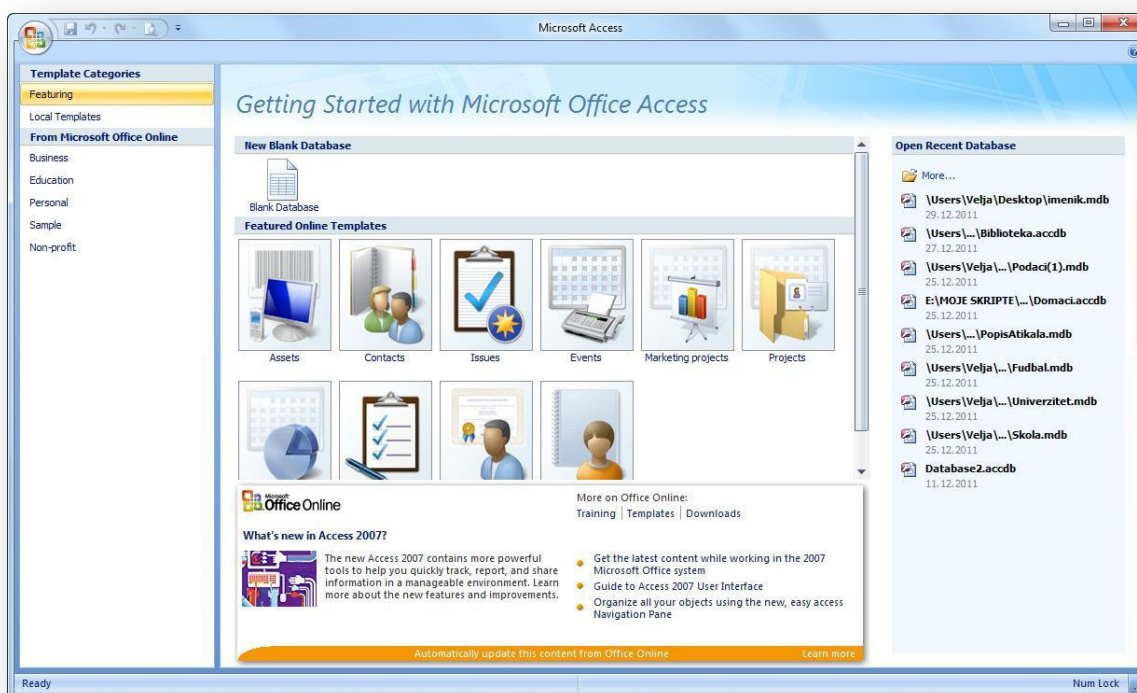
Zašto zasebne tabele? Jednostavno - zbog izbegavanja dupliranja podataka. Podaci kao što su telefon, adresa, ime i prezime, datum uplate članarine za čitaoca su jedinstveni i ne postoji niti jedan razlog da se oni ponavljaju. Ako bi pak kreirali samo jednu tabelu, u nju bi

morali da unosimo ime i prezime čitaoca (njegovu adresu, broj telefona, datum plaćanja članarine...) i naslov knjige za svako izdavanje i vraćanje (sa njegovim autorom, godinom izdanja, izdavačem...), te bi ovim nepotrebno duplirali podatke. Još gore: veoma su moguće greške u kucanju - nikad ne bismo bili načisto koliko je knjiga koji čitalac pročitao, odnosno koliko knjiga uopšte postoji u biblioteci.

Tabeli **Čitaoci** ćemo na kraju pridružiti i univerzalni identifikacioni broj, broj koji će se upisivati u tabelu **Poslovanje**. Sem toga, tabeli knjige ćemo pridružiti takođe nezavisan univerzalni identifikacioni broj, nalik ono što smo ranijih godina viđali po bibliotekama - inventarni broj. Operatoru u biblioteci je mnogo lakše da na vašu karticu upiše broj knjige koju ste uzeli, umesto da piše pun naslov knjige.

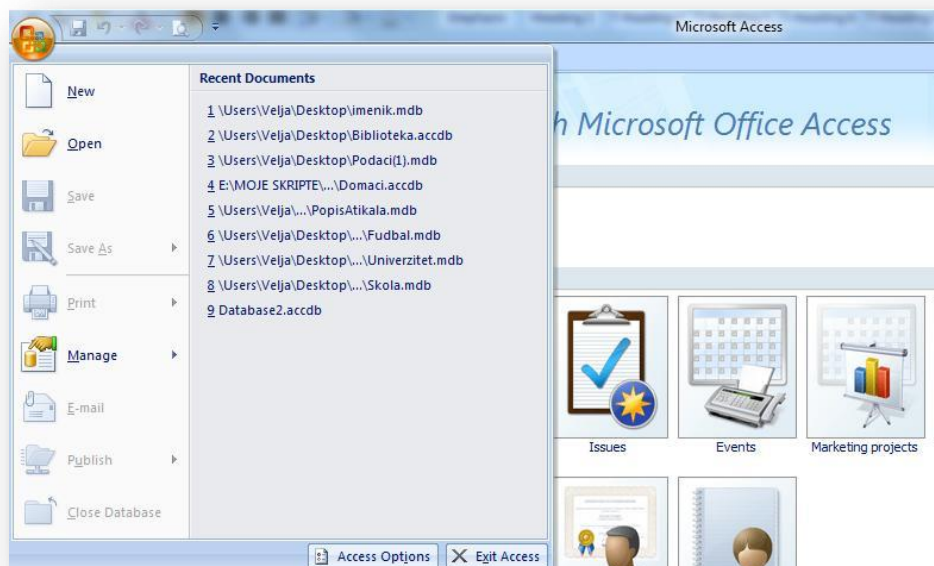
3. 3. Pokretanje programa, pripremne radnje, kreiranje prve tabele

Kada se pokrene Access 2007 otvora se prozor prikazan na sledećoj slici:



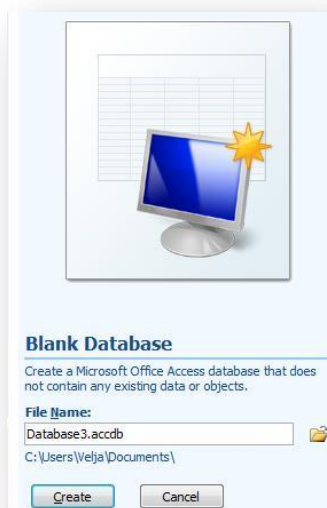
Slika 5

Klikom na Office dugme (Office Button) u gornjem levom uglu, otvara se meni gde je potrebno izabrati New, koja je prikazana na sledećoj slici:



Slika 6

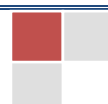
Sa leve strane pojaviće se forma za unos imena baze podataka, što je prikazano na sledejoj slici.



Slika 7

Nakon klika na dugme Create, otvara se novokreirana baza podataka, odnosno prostor u kome se kreiraju tabele za bazu podataka.

SVAKA baza podataka kreirana u Access 2007 dobija ekstenziju .accdb što je skraćeno od Access DataBase, za razliku od prethodnih verzija Access-a gde je ekstenzija bila .mdb (Microsoft DataBase).



3. 4. Planiranje tabela

Pošto smo stvorili sliku o potrebnom broju tabela i njihovoj nameni, možemo preći na sledeći korak - definisanje tabela. Broj tabela smo već utvrdili i on iznosi četiri (**Čitaoci**, **Biblioteka**, **Knjige** i **Poslovanje**). Za svaku tabelu valja definisati polja i adekvatne tipove podataka koji se mogu u njih unositi:

Tabela 'Čitaoci'

Naziv polja	Tip polja	Podvrsta / veličina polja	Komentar
ID_Citalac	Number	Integer (*)	
Prezime_Ime	Text	35	u isto polje upisujemo i ime i prezime
Adresa	Text	30	broj karaktera je restriktivno postavljen, ovo se kasnije može povećati po potrebi; uključuje naziv ulice i broj kuće
Mesto	Text	30	uključuje poštanski broj i naziv mesta
Telefon	Text	15	tip je tekst zbog interpunkcije formata xxx / AAA-BBBB
Dat_rodj	Date	Short Date (**)	potrebno za statističku obradu starosne strukture čitalaca
Dat_upisa	Date	Short Date	takođe potrebno za statističku obradu
Dat_clan	Date	Short Date	datum plaćanja članarine; potreban za uvid u finansijsko poslovanje i obaveštavanje čitalaca o isteku članarine
Napomena	Memo	(***)	

(*) Integer je tip polja Number koji omogućava upis brojeva od -32.767 do 32.767 a kako identifikacioni broj ne može ići u minus smatramo tip polja 0 do 32.767; ako će biti više čitalaca od tog broja, stavljamo tip Long Integer za brojeve od 0 do 2.147.483.648

(**) Short Date predstavlja prikaz datuma u obliku dd/mm/yy

(***) Memo je tip polja koji ne uzima fiksnu memoriju već zavisi od broja ukucanih karaktera

Tabela 'Knjige'

Naziv polja	Tip polja	Podvrsta / veličina polja	Komentar
ID_Knjiga	Number	Long Integer	Biće mnogo knjiga, treba nam više od 32.767
Autor	Text	30	
Naslov	Text	40	
Izdavac	Text	30	
God_izd	Text	5	format polja godine sa četiri cifre
Zanr	Text	1	usvojićemo oznake kao:f - fantastka, p - poezija i sl.
Status	Text	1	konvencija: r - raspoloživa, n - nije vraćena, z - nije za izdavanje...

Tabela 'Biblioteka'

Naziv polja	Tip polja	Podvrsta / veličina polja	Komentar
Naziv	Text	50	
Adresa	Text	40	
Mesto	Text	35	
Telefon	Text	15	
Direktor	Text	30	

Tabela 'Poslovanje'

Naziv polja	Tip polja	Podvrsta / veličina polja	Komentar
ID_Zapis	Autonumber		
ID_Citalac	Number	Integer	
ID_Knjiga	Number	Long Integer	
Dat_izd	Date	Short Date	datum izdavanja
Dat_vr	Date	Short Date	datum vraćanja

3. 5. Kreiranje prve tabele

Tabele u Access-u se mogu formirati na nekoliko načina:

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 retko koristi, jedino ako nije poznata struktura tabele ili ako je potrebno ubaciti kopirane podatke iz nekog drugog programa (npr. MS Excel).

Design View omogućava kreiranje tabele detaljnim izborom svojstava polja podataka, kao što su imena polja, tip podataka i veličina polja, određivanje primarnog ključa, i slično. To je najčešće korišćen pristup.

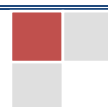
Table Wizard poziva čarobnjaka koji koristi biblioteku već formiranih tabela. Ova mogućnost je korisna kada se kreiraju standardni objekti kao što su adresari, računi u fazi učenja, ali se u praksi ne koristi mnogo.

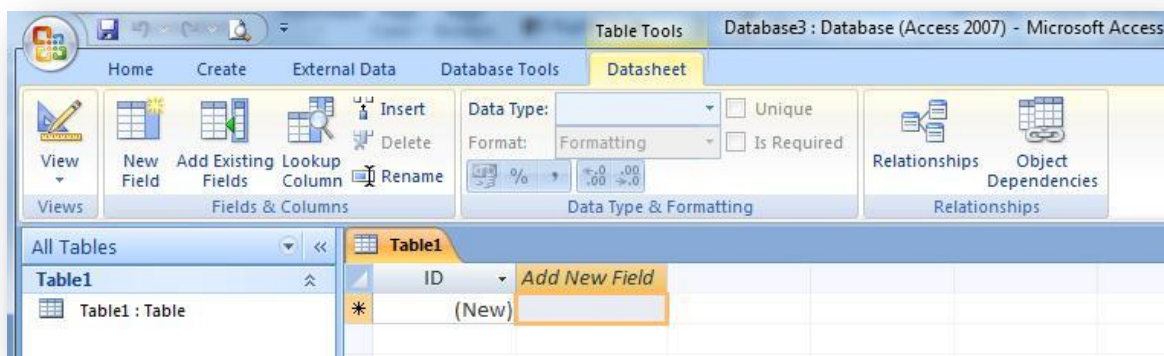
Import Table omogućava uvoz tabele iz nekog drugog izvora podataka, druge Access baze, MS Excel dokumenta, kao i fajlova tekstualnog formata.

Link Table obezbeđuje povezivanje sa tabelama iz druge MS Access baze (može i neki drugi format baze), pri čemu tabela ostaje u matičnoj bazi, a linkom se omogućuje veza na nju u cilju pregleda i eventualno ažuriranja podataka. U ovoj situaciji, pristup dizajnu tabele nije moguć iz baze koja ima link na datu tabelu, već samo iz matične (izvorne) baze. Dizajnu tabele se ne može pristupiti dok se svi aktivni linkovi na odnosnu tabelu ne isključe, što znači da pri dizajnu tabele niko (kroz mrežu) ne može da je ažurira.

Design View mod

Prostor u kome se kreiraju tabele prikazan je na sledećoj slici.





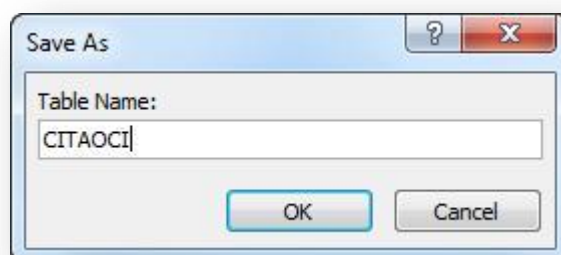
Slika 8

Desnim klikom miša na Table1 prikazuje se meni iz kojeg se tabela otvara u Design View modu, kao što se vidi na slici. Design View mod se otvara onda kada je potrebno uraditi podešavanja vezana za dizajn i funkcionalnost otvorenog prozora, bez obzira da li je u pitanju tabela, upit, forma, izveštaj, ili makro.

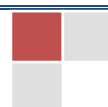


Slika 9

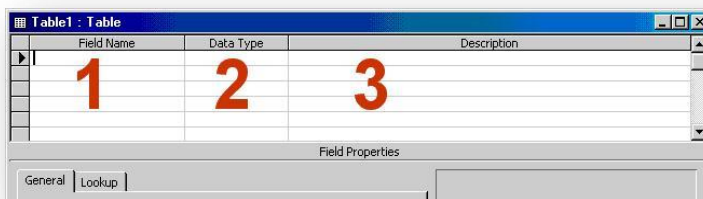
Prvo je potrebno upisati naziv table. Naziv table je CITAOCI. Dodeljivanje imena tabeli vrši se u prozoru prikazanom na sledećoj slici.



Slika 10



Zatim se otvara prozor za dizajniranje tabele.



Slika 11

Na slici, kolona obeležena brojem 1 predstavlja naziv kolone, broj 2 je tip kolone, a broj 3 je komentar koji se pojavljuje u statusnoj liniji Accessovog prozora. U **Field Name** preseka prve kolone i reda upisaćemo tekst **ID_Citalac**. U drugoj koloni istog reda ukucaćemo **Number**. U koloni broj 3 istog reda stavićemo komentar **Jedinstveni broj čitaoca**.

MS Access podržava sledeće tipove podataka (Data Type):

Field Name	Data Type
ID_Citalac	Number
Prezime_Ime	Text
Adresa	Memo
Mesto	Number
Telefon	Date/Time
Dat_rodj	Currency
Dat_upisa	AutoNumber
Dat_clan	Yes/No
Fotografija	OLE Object
	Hyperlink
	Attachment
	Lookup Wizard...

Slika 12

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

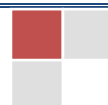
Memo - dugačak tekst (slova, brojevi i svi specijalni znaci) dužine do 65535 karaktera. Ovakav tip tekstualnog podatka ne može ući u neko sortiranje, pretraživanje ili grupisanje.

Number brojevi koji se upotrebljavaju u aritmetičkim izrazima, a veličina polja zavisi od "podtipa" koji može biti: Byte, Integer, Long Integer, Single, Double, Replication ID,

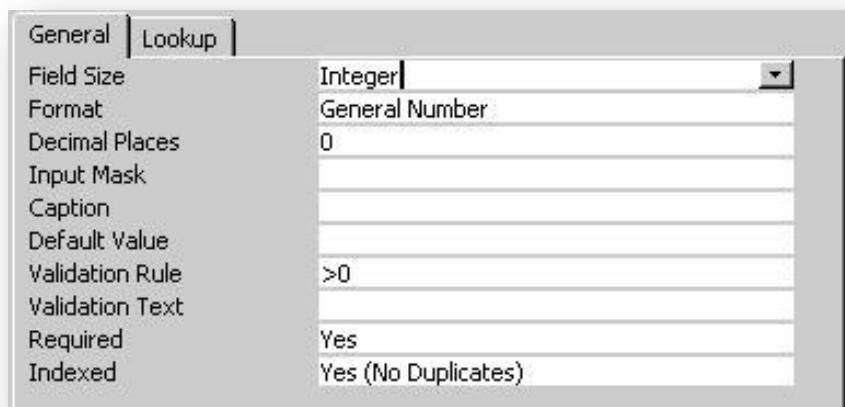
Date/Time, Currency, Auto Number (koji može biti Long Integer ili Replication ID opsega) ili **Yes/No**.

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

Lookup Wizard-om se kreiraju polja koja omogućavaju izbor vrednosti iz druge tabele ili iz (ograničene) liste vrednosti upotrebom ComboBox ili ListBox kontrole. Izborom ove stavke startuje se Lookup Wizard, kojim se definiše vrsta izvora, način popunjavanja i broj polja.

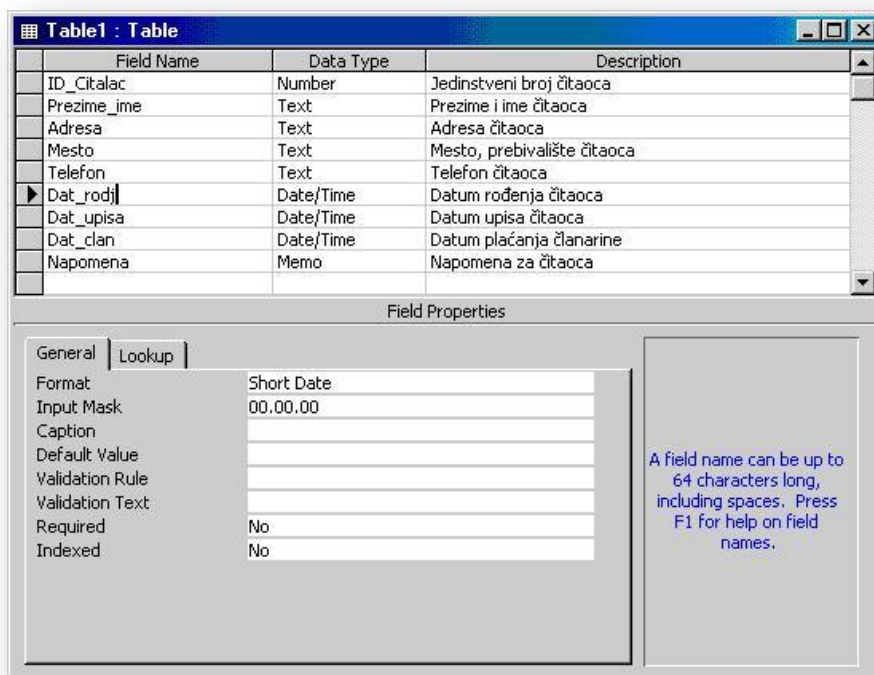


Primetimo da se na delu ekrana iznad kojeg piše **Field Properties** pojavila tabela sa različitim stavkama. Ako ste ispravno radili, na ekranu ćete imati prikaz sličan kao na ovoj slici:



Slika 13

Još jednom da objasnimo šta smo učinili. Započeli smo kreiranje tabele i nazivu prve kolone smo dodelili tekst **ID_Citalac**. Tip polja, kako smo ranije isplanirali, je **Number**. To znači da će u polje moći da se ukucavaju isključivo numeričke vrednosti, a sam tip polja ćemo postaviti na **Integer**. Na slici, ispod teksta **Field Properties**, na kartici **General** prva opcija je "veličina" (odnosno pod-tip) polja. Ovu vrednost ćemo promeniti na **Integer**. Postavimo i ostale osobine kao na slici:



Slika 14

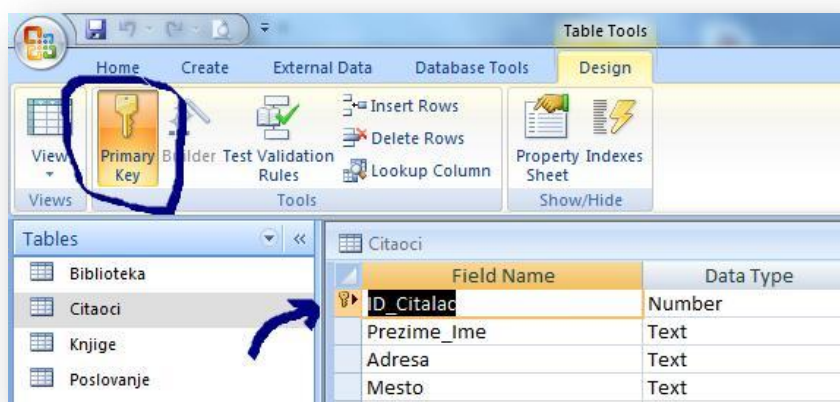
Primetimo da je na slici aktivno polje sa datumom. Ovde smo iskoristili osobinu **Input Mask** ("maska" za unos u polje), koju smo postavili na **00.00.00**. To nalaže operateru/bibliotekaru

da datum ukucava isključivo kao 6 cifara - dve za dan, dve za mesec i dve za godinu (2000. godina se ukucava kao 00).

Osobine koje se mogu postaviti za polje Number su odozgo na dole sledeće:

- ✦ **Field Size** - veličina, odnosno tip polja
- ✦ **Format** - način na koji će ukucani brojevi biti prikazani
- ✦ **Decimal Place** - broj decimala u cifri. Vrednost postavljamo na nula, pošto identifikacioni broj nema smisla sa decimalama
- ✦ **Input Mask** - "maska" za unos brojeva. Ako u ovo polje postavimo vrednost 00000, korisnik će morati da unese tačno pet cifara za identifikacioni broj čitaoca (npr. 00023)
- ✦ **Caption** - tekst koji se pojavljuje kao naziv kolone u pogledu za unos podataka (mi smo trenutno u dizajn-pogledu)
- ✦ **Validation Rule** - pravilo koje se mora ispoštovati pri unosu. Vrednost smo postavili na >0 kako se ne bi mogli uneti negativni brojevi.
- ✦ **Validation Text** - tekst koji se pojavljuje kao upozorenje kada korisnik pogreši pri unosu
- ✦ **Required** - određuje da li je unos neophodan ili ne. Vrednost postavljamo na Yes.
- ✦ **Indexed** - da li je polje indeksirano ili ne. U indeksiranom polju pretraga je brža. **No Duplicates** u zagradi znači da se ne smeju pojaviti (ukucati) dve iste šifre za dva različita čitaoca.

Sledeći i poslednji važan korak je postavljanje **primarnog ključa**. **Primarni ključ je jedno ili više polja preko kojih će tabela biti povezana sa drugim tabelama, preko koje će se podaci indeksirati i izbeći mogućnost dupliranih unosa.** U tabeli Čitaoci primarni ključ ćemo postaviti na polje ID_Citalac. Pozicionirajmo kursor na red u kome nalazi tekst ID_Citalac. Kliknimo na ikonu ključa na toolbaru. Izgled reda tabele ID_Citalac će biti sada ovakav:



Slika 15

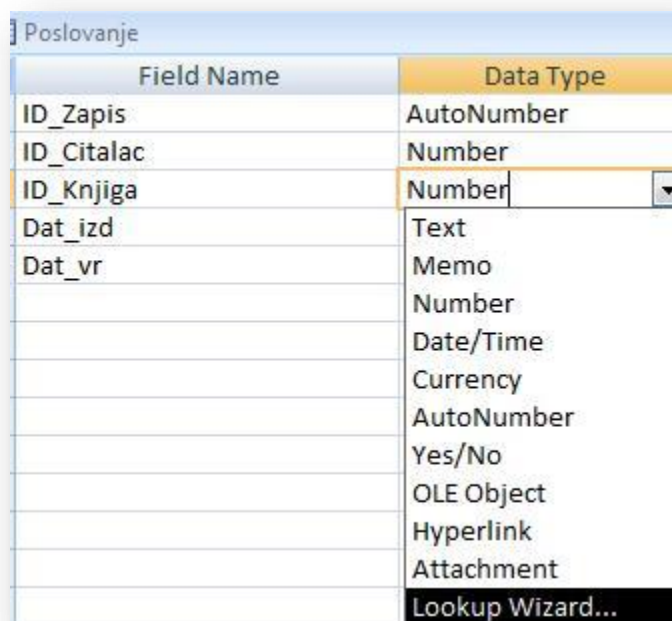
Ovim je prva tabela kreirana i snimljena. Na sličan način ćemo kreirati i ostale tabele.

3. 6. Povezivanje tabela

Iskoristićemo postojeće, ranije kreirane tabele i povezaćemo ih međusobno u **relacije**. Punu snagu Access ostvaruje iz relacionih tabela (otuda naziv relacione baze podataka). Iz relacionih tabela podatke je lako filtrirati i na njih primeniti razne kriterijume bez obzira iz koliko tabela podaci potiču.

Ako analiziramo spisak polja i strukturu tabela videćemo da međusobnog povezivanje tabela nema mnogo. Praktično, ima ih dva i oba se nalaze u tabeli **Poslovanje**. Polje **ID_Knjiga** bi valjalo povezati sa istoimenim poljem iz tabele **Knjige** i polje **ID_Citalac** sa istoimenim poljem iz tabele **Citaoci**.

Vraćamo se u Access i prozor baze. Selektujemo tabelu **Poslovanje** i kliknemo na taster **Design**. Sada se otvara već poznati dizajn-pogled tabele. Kliknimo u polje **Number** reda **ID_Knjiga**. Primetićemo da je polje "drop-down" tipa, odnosno da sadrži listu za izbor koja se može videti klikom na strelicu na dole ili pritiskom na taster F4.

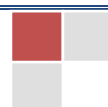


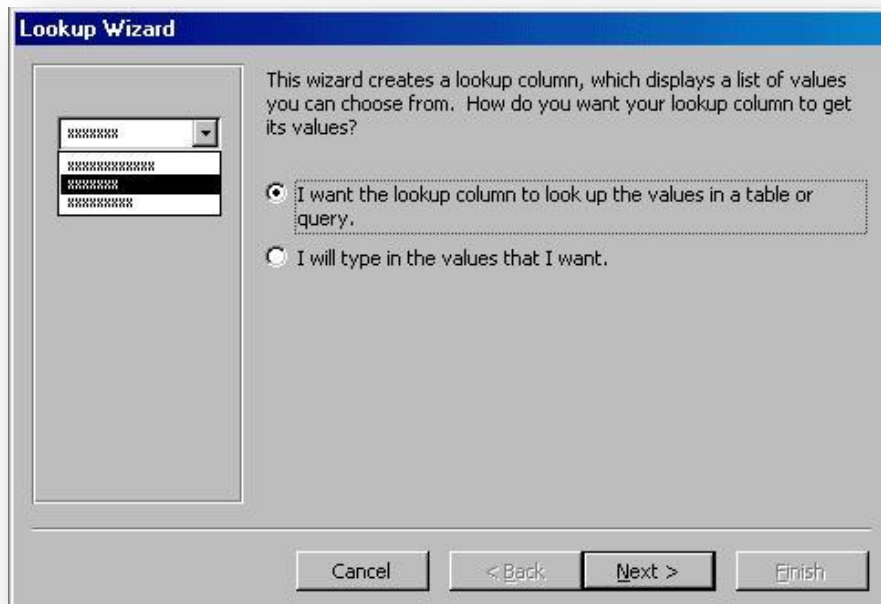
Field Name	Data Type
ID_Zapis	AutoNumber
ID_Citalac	Number
ID_Knjiga	Number
Dat_izd	Text
Dat_vr	Memo
	Number
	Date/Time
	Currency
	AutoNumber
	Yes/No
	OLE Object
	Hyperlink
	Attachment
	Lookup Wizard...

Slika 16

Ono što ćemo sada učiniti je da polja **ID_Knjiga** i **ID_Citalac** ograničimo na sadržaj tabela **Knjige** i **Citaoci**. Tip polja **Number** reda **ID_Knjiga** promenimo na **Lookup Wizard...**

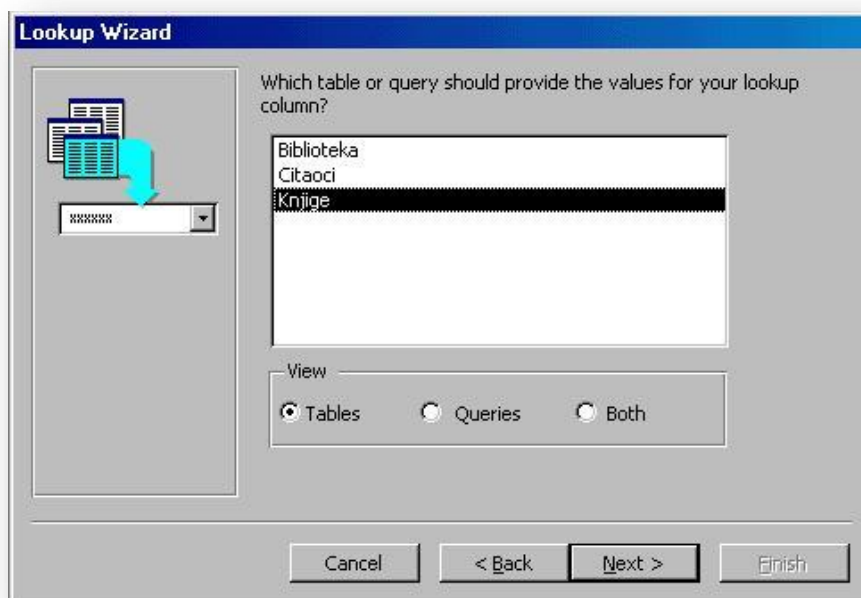
Pokreće se procedura koja će na osnovu naših odgovora na postavljena pitanja formirati relaciju umesto nas:





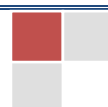
Slika 17

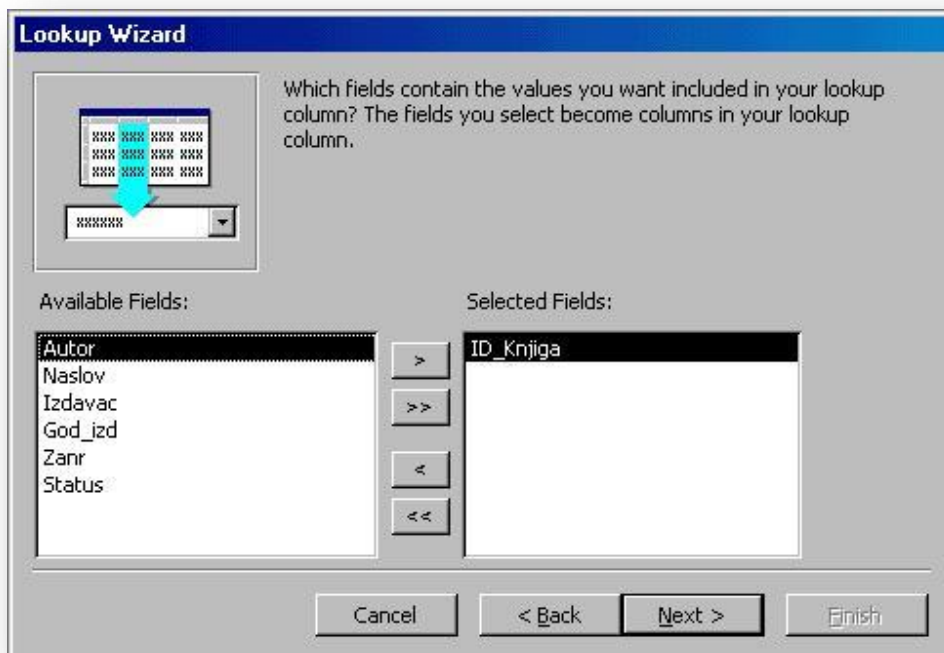
Na prvo pitanje odgovaramo sa prvom opcijom jer želimo da se u tabeli **Poslovanje** na mesto broja knjige pojavljuju samo vrednosti iz tabele knjiga, odnosno samo smisleni inventarni brojevi (biranjem druge opcije u koloni **ID_Knjiga** mogli bismo da kucamo bilo koji inventarni broj, pa i onaj nepostojeći). Klikom na **Next** (ili sa tastature **ALT+N**) sledi drugi korak:



Slika 18

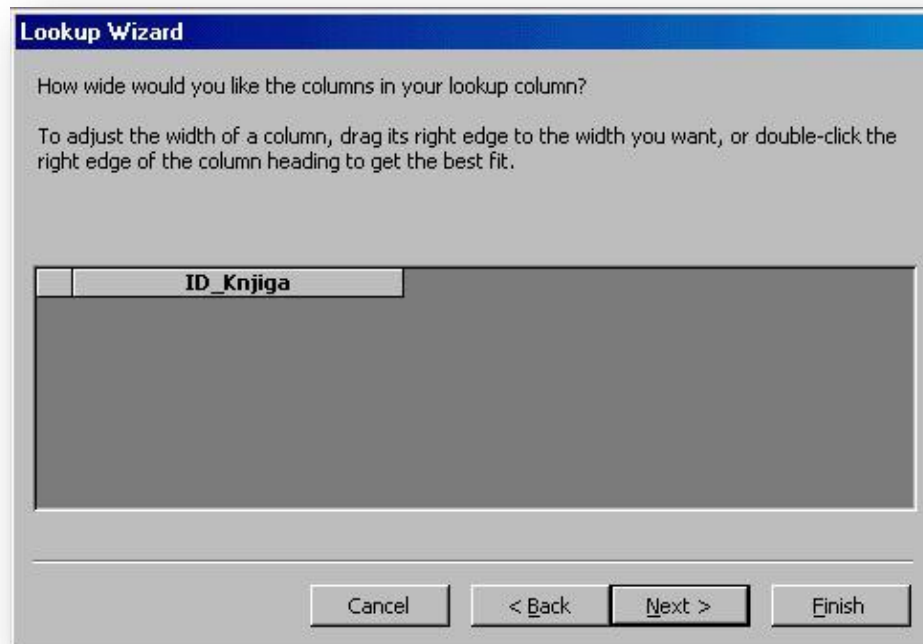
U ovom koraku biramo tabelu iz koje će se uzimati podaci, odnosno određujemo "izvorišnu" tabelu. Dalje, u trećem koraku određujemo kolonu (polja) koja će biti "izvor" podataka:





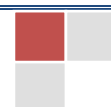
Slika 19

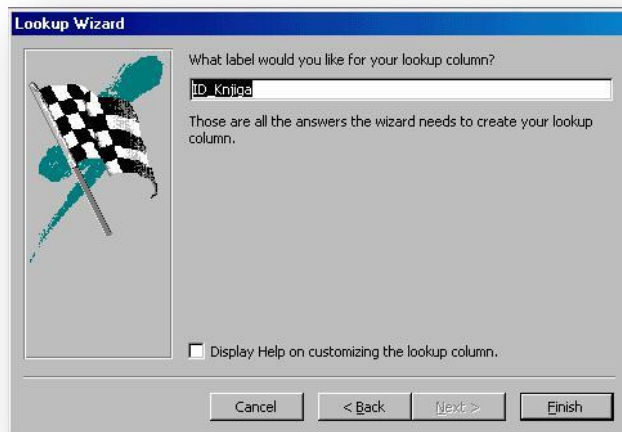
Preostala dva koraka nisu neophodna i već sada možemo kliknuti na taster **Finish**. Četvrti korak je "kozmetički" i ovdje određujemo širinu kolone koje će se prikazivati na ekranu:



Slika 20

U posljednjem koraku možemo uticati na naziv kolone, a kako nam naziv **ID_Knjiga** već odgovara, kliknućemo na taster **Finish**:



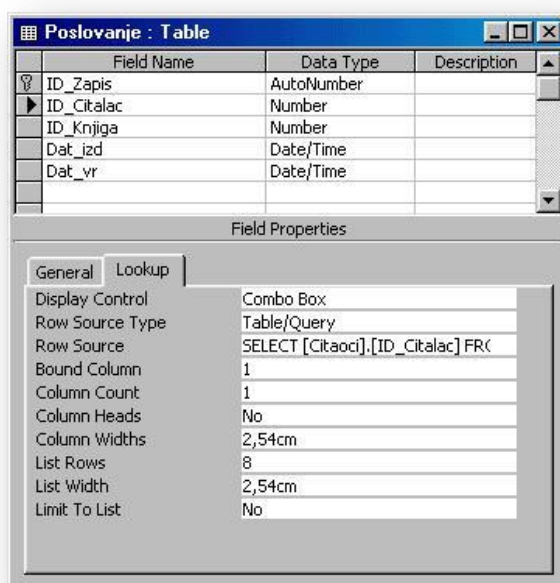


Slika 21

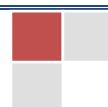
Ovim je kreirana prva tzv. relacija tabela **Poslovanje** i **Knjige**. Tabelu **Poslovanje** treba istom procedurom povezati sa tabelom **Citaoci**, što ćete prateći prethodni postupak sami uraditi.

Polja koja se povezuju nose isto ime u obe tabelle: **ID_Citalac**.

Još jednu, veoma bitnu, stvar moramo napraviti da bi ova veza funkcionisala kako valja: operateru se mora onemogućiti da u polja **ID_Knjiga** i **ID_Citalac** upisuje proizvoljne vrednosti, odnosno one vrednosti za koje ne postoji odgovarajuća šifra u tabelama **Knjige** i **Citaoci**. Otvorimo ponovo tabelu **Poslovanje** u dizajn-pogledu. Selektujemo red **ID_Knjiga** klikom miša. Ispod teksta **Field Properties** postoje dva tzv. taba - **General** i **Lookup**. Klikom na tab **Lookup** otvara se lista osobina za polja čiji sadržaj se gleda iz druge tabelle. Na ovoj listi za polja **ID_Knjiga** i **ID_Citalac** tabelle **Poslovanje** osobinu **Limit To List** postavimo na vrednost **Yes**.

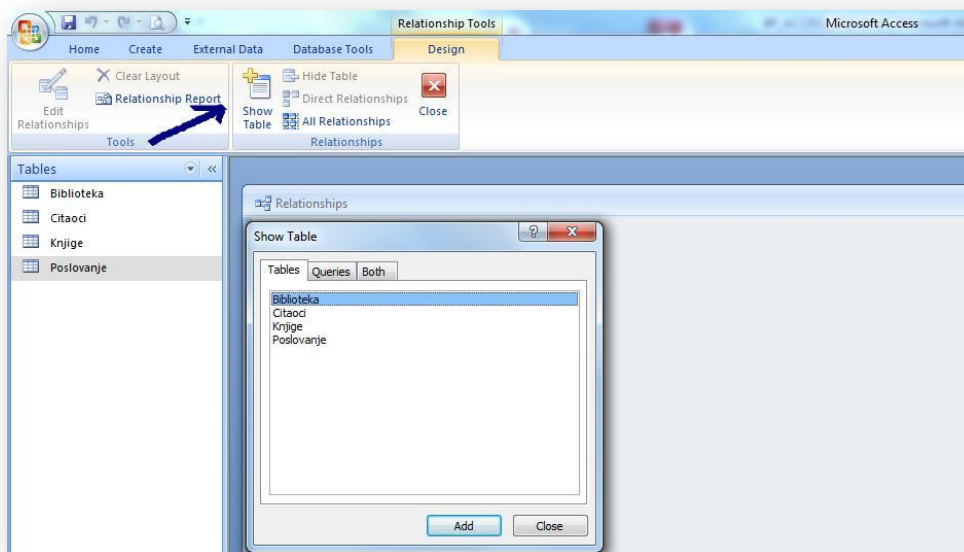


Slika 22



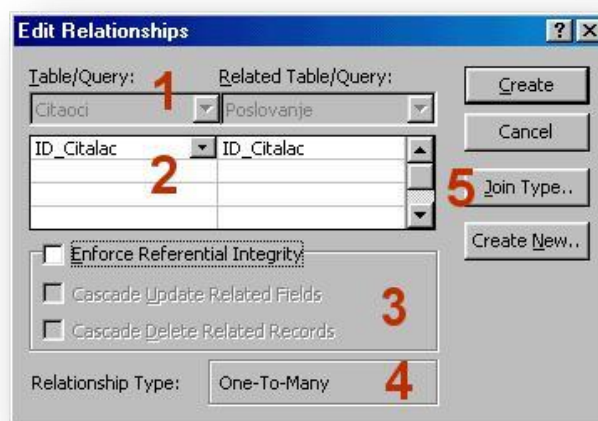
Snimimo i zatvorimo tabelu. Ovim smo ograničili sadržaj polja `ID_Citalac` i `ID_Knjiga` na sadržaj druge dve relevantne tabele - `Knjige` i `Citaoci`.

Pored ovih `Lookup` relacija napravićemo još i prave relacije između tabela. Sa kartice `DatabaseTools` biramo opciju `Relationships...`. U prazan prozor koji se pojavi, dodaćemo tabele koje ćemo povezati. Spisak postojećih tabela dobija se klikom na taster `Show Table`.



Slika 23

Dodajmo redom tabele: `Citaoci`, `Poslovanje` i `Knjige`. Iz tabele `Citaoci` prevucimo polje `ID_Citalac` u tabelu `Poslovanje` na polje `ID_Citalac`. U prozoru koji se pojavi, podešavamo osobine ove relacije.



Slika 24

Na slici, obeleženo brojem 1 su tabele koje će biti povezane. Polja koja su međusobno povezana su obeležena brojem 2. Pod brojem 3 su opcije koje će definisati ponašanje ciljne tabele u zavisnosti od promene podataka u izvorišnoj tabeli. Ove opcije ćemo sve uključiti jer će one omogućiti da se, ukoliko se npr. promeni inventarni broj knjige u tabeli `Knjige`, ova promena reflektuje na tabelu `Poslovanje`.

Primer:

Ako je čitalac sa šifrom 15 zadužio i razdužio knjigu sa inventarnim brojem 26, ove promene će ostati zapisane u tabeli **Poslovanje** i taj podatak može poslužiti u vođenju statistike o čitanosti knjige i broju pročitanih knjiga za čitaoca. Takođe, ukoliko se knjiga ošteti, može se na osnovu istorije izdavanja knjige utvrditi ko je od čitalaca iznajmljivao tu knjigu i tako eventualno pronaći krivca za oštećenje. U momentu kada se knjizi iz raznih razloga promeni njen inventarni broj, ili se knjiga rashoduje, ova promena se neće odraziti na tabelu **Poslovanje**, ukoliko pomenute opcije sa slike obeležene brojem 3 nisu uključene. U našem slučaju bismo rashodovali knjigu 26 i na mesto njenog broja postavili drugu knjigu, pa bismo dobili sasvim pogrešne podatke o izdavanju knjige i čitaocima koji su knjigu iznajmljivali.

Uključenjem opcija pod **Enforce Referential Integrity** će se promene podataka u tabelama **Citaoci** i **Knjige** automatski reflektovati na tabelu **Poslovanje**.

Na slici pod brojem 4 označen je tip relacija. U našem slučaju **One-to-many** znači da knjiga može imati jedinstven broj, da se ta knjiga može više puta izdati, te da jedan čitalac u isto vreme može uzeti više knjiga. Sem ove relacije postoji još dva tipa relacije.

Druga je **One-to-one** što bi značilo da jedan čitalac može uzeti samo jednu knjigu. Ova relacija bi bila primenljiva u, recimo, "Sega" klubovima, gde biste jednom članu dozvolili da iznajmi jednu konzolu za igranje.

Treća vrsta relacija je **Many-to-many**. Ova relacija se upotrebljava u slučajevima kada postoji osobina zajedničke pripadnosti. Za primer, recimo da dva radnika zaduže dva kompleta alata bez posebne zajedničke pripadnosti. Obojica su tada vlasnici i jednog i drugog kompleta alata.

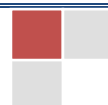
Pod brojem 5 na slici je označen taster **Join Type....** Klikom na taster ili prečicom **ALT+J** otvara se dijalog sa tri opcije:



Slika 25

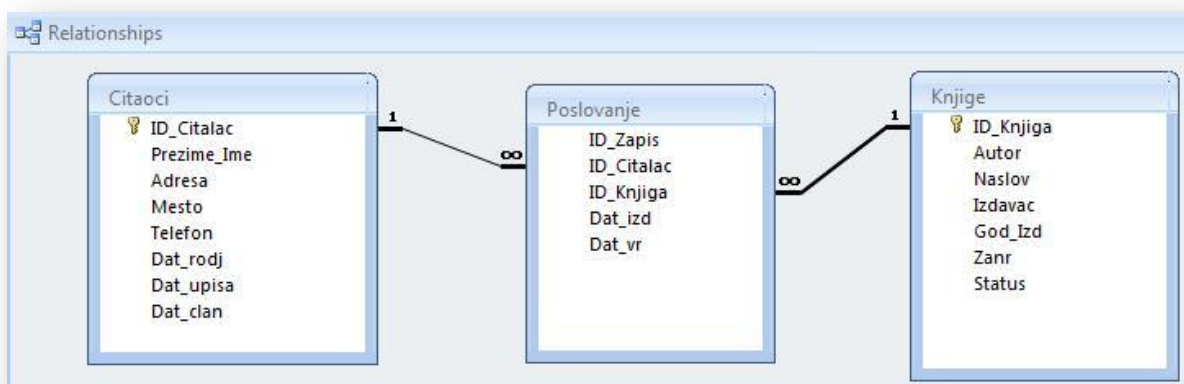
Prva opcija će dozvoliti pojavljivanje knjiga u tabeli **Poslovanje** samo za one inventarne brojeve koji već postoje u tabeli **Poslovanje**.

Biranjem druge opcije u tabeli **Poslovanje** će moći da se "pojave" svi inventarni brojevi knjiga.



Trećom opcijom bismo dozvolili (slično prvij) pojavljivanje inventarnih brojeva knjiga samo onih koji već postoje u tabeli **Poslovanje**. Praktično prva i treća opcija nemaju nekog smisla u našem slučaju. Stoga odabiramo drugu opciju i potvrđujem izbor klikom na taster **OK**. Takođe potvrdimo i promene osobina relacija još jednim klikom na taster **OK**.

Prateći postupak u kome smo kreirali relaciju između tabela **Citaoci** i **Poslovanje** gde smo povezali polja **ID_Citalac**, povežimo i tabele **Knjige** i **Poslovanje**. Nakon obavljene operacije imaćemo izgled relacija kao na sledećoj slici.



Slika 26

Konačno zatvorimo prozor relacija i odgovorimo potvrdno na postavljeno pitanje kojim snimamo naše promene na relacijama tabela.

3. 7. Unošenje prvih zapisa

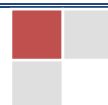
Pre nego što pređemo na kreiranje formulara za rad sa podacima, iskoristićemo priliku da unesemo podatke direktno u tabele. Način na koji Access smešta podatke u tabele je naizgled veoma jednostavan, a njegovo razumevanje je veoma važno za kasnije dizajniranje formulara i izveštaja.

Otvorimo Access i bazu podataka na kojoj radimo. Pređimo na tab **Table**. Od četiri tabele koje smo kreirali (**Biblioteka**, **Knjige**, **Citaoci**, **Poslovanje**) otvorimo za unos tabelu **Citaoci**. Tabela se "otvara" za unos jednim klikom na naziv tabele i zatim klikom na taster "Open" ili jednostavno - dvoklikom na naziv tabele.

Dvoklik na naziv tabele uz pritisnut taster CTRL otvara tabelu u dizajn-pogledu.

Popunimo prvi slog tabele sledećim podacima:

Prezime_ime	Petrović Petar
Adresa	Nemanjina 15
Mesto	21000 Novi Sad
Telefon	021/123-456
Dat_rodj	27.03.72
Dat_upisa	01.01.00 (00 ovde skraćeno označava 2000. godinu!)
Dat_clan	01.01.00



ID_Citalac	Prezime_ime	Adresa	Mesto	Telefon	Dat_rodj	Dat_upisa	Dat_clan	Napomena
1	Petrović Petar	Nemanjina 15	21000 Novi Sad	021/123-456	27.03.72	01.01.00	01.06.00	

Slika 27

Otvorimo zatim tabelu **Knjige** i unesimo, takođe, jedan slog:

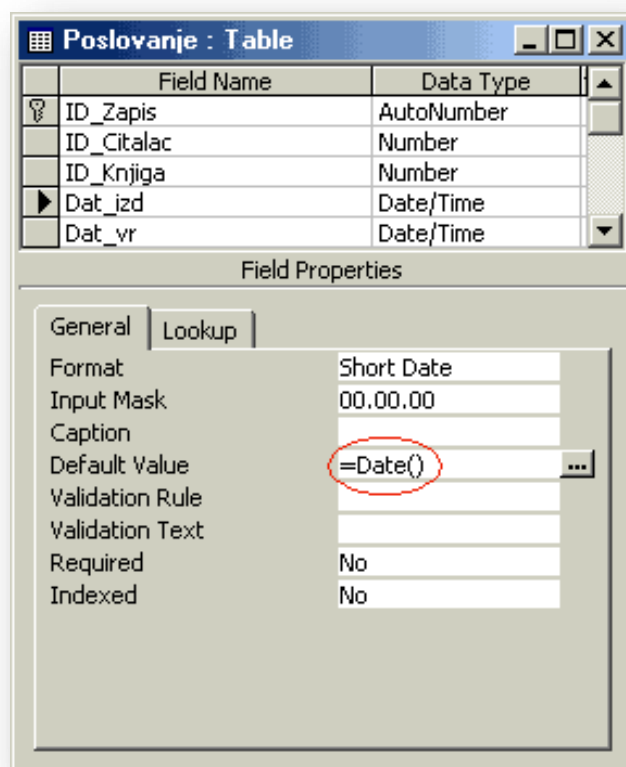
ID_Knjiga 1
Autor Grupa autora
Naslov Korak po korak - Microsoft Office 2000
Izdavac Microsoft Press
God_izd 1999
Zanr n (Naučna literatura)
Status u (U upotrebi)

Pošto postoji barem jedan slog u obe tabele možemo otvoriti i treću tabelu - **Poslovanje**, kako bismo čitaocu "iznajmili" knjigu. U polje **ID_Zapis** ne upisujemo ništa, uostalom Access to neće dozvoliti, jer smo to polje definisali kao **AutoNumber**. To polje Access sam popunjava inkrementirajućim brojem. Primetimo da kada započnemo unos u polje **ID_Citalac**, u polju **ID_Zapis** će pisati 1. Dakle, popunimo prvi slog tabele **Poslovanje** sledećim vrednostima:

ID_Citalac 1 (čitalac sa šifrom broj 1)
ID_Knjiga 1 (knjiga pod inventarnim brojem 1)
Dat_izd 01.11.00
Dat_vr 05.11.00

Sve što operater - bibliotekar treba da radi je da unese šifru knjige i čitaoca, kao i datum izmene. Možemo na ovom mestu ići i korak dalje - recimo da se u polju **Dat_izd** kao unapred ponuđena vrednost nađe tekući datum. Time bismo dodatno rasteretili operatera unosa datuma.

Zatvorimo tabelu **Poslovanje**. Uz pritisnut taster **CTRL** dvokliknimo mišem ponovo na tabelu **Poslovanje** (za izmenu dizajna tabele). Kliknimo na red u kome piše **Dat_izd** i upišimo u vrednost osobine Default Value (osnovna vrednost) funkciju: **=Date()**. Snimimo i zatvorimo tabelu.



Slika 28

U trenutku kada smo započeli unos u tabele **Knjige** i **Citaoci**, verovatno ste primetili znak + na levoj strani sloga. Klikom na taj znak Access prikazuje kaskadnu tabelu za koju je direktno vezana tabela sa kojom operišemo, odnosno prikazuje slog ili grupu slogova koji su u relaciji sa slogom otvorene tabele. Za tabelu **Citaoci** klikom na znak + dobija se tabela **Poslovanje** gde su slogovi u međusobnoj direktnoj relaciji:

ID_Citalac	Prezime_Ime	Adresa	Mesto	Telefon	Dat_rodj																
1	Jankovic Gorica	Trbusnica bb	Trbusnica	8192501	1.5.1983																
<table border="1"> <thead> <tr> <th>ID_Zapis</th> <th>ID_Knjiga</th> <th>Dat_izd</th> <th>Dat_vr</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>3</td> <td>30.6.2007</td> <td>13.7.2007</td> </tr> <tr> <td>2</td> <td>4</td> <td>30.6.2007</td> <td>17.7.2007</td> </tr> <tr> <td colspan="4">* (New)</td> </tr> </tbody> </table>						ID_Zapis	ID_Knjiga	Dat_izd	Dat_vr	1	3	30.6.2007	13.7.2007	2	4	30.6.2007	17.7.2007	* (New)			
ID_Zapis	ID_Knjiga	Dat_izd	Dat_vr																		
1	3	30.6.2007	13.7.2007																		
2	4	30.6.2007	17.7.2007																		
* (New)																					
2	Zujovic Ivana	Nikole Tesle 46	Lazarevac	8126485	6.7.1982																
3	Kvas Stevan	Pere Erjaveca 21	Lazarevac	8120825	26.6.1981																
<table border="1"> <thead> <tr> <th>ID_Zapis</th> <th>ID_Knjiga</th> <th>Dat_izd</th> <th>Dat_vr</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>8</td> <td>30.7.2007</td> <td>12.8.2007</td> </tr> <tr> <td colspan="4">* (New)</td> </tr> </tbody> </table>						ID_Zapis	ID_Knjiga	Dat_izd	Dat_vr	7	8	30.7.2007	12.8.2007	* (New)							
ID_Zapis	ID_Knjiga	Dat_izd	Dat_vr																		
7	8	30.7.2007	12.8.2007																		
* (New)																					
4	Vasic Marijana	Sopic bb	Sopic	8155203	11.2.1984																

Slika 29

Sa slike možemo videti da je čitalac sa šifrom 1 iznajmio knjigu sa inventarnim brojem 1. Ovo je zgodan način za testiranje ispravnosti relacija i tabela pre nego što se pređe na dizajniranje ostalih elemenata baze, naročito kada operišemo sa više tabela.

Postavlja se pitanje kako ćemo razdužiti čitaocu knjigu kada je bude vratio? To možemo izvesti na dva načina: ili ćemo obrisati slog u tabeli [Poslovanje](#) za čitaoca i knjigu koji više nisu u vezi (i pri tome iskopirati taj slog u pomoćnu tabelu za statističku obradu) ili ćemo ostaviti taj slog u tabeli, a knjige koje se pojavljuju u padajućoj listi tabele [Poslovanje](#), [ID_Knjiga](#) nekako ograničiti samo na raspoložive knjige.

Na sledećem času ćemo iskoristiti polje [Status](#) tabele [Knjiga](#) i u njega upisivati status knjige, a taj parametar ćemo iskoristiti za ograničenje liste raspoloživih knjiga. Takođe ćemo kreirati i ostale formulare (neki ih zovu i obrascima, šablonima, formama).

