Sadržaj

Predgovor. 1
Uvod 3

1. Osnove izrade aplikacije 5
1.1 Tabele i ograničenja na nivou tabela 5
1.2 Upiti nad jednom i više tabela 14
1.3 Forme za unos, podforme i pokretačke forme 31
1.4 Izveštaji i podizveštaji 51
1.5 Makroi 64
1.6 Meniji i kontrolne table 74

2. Automatizacija događajima 81
2.1 Promenljive 81
2.2 Operatori 91
2.3 Upravljačke strukture 97
2.4 Referenciranje 108
2.5 Moduli i procedure 112
2.6 Funkcije 128
2.7 Načini deljenja Access baze podataka 138

Prilog A: Zadaci za vežbu 143
Prilog B: Neke korisničke funkcije. 157

Literatura 173

Predgovor

Ova knjiga predstavlja težnju autora da pretoči dugogodišnja iskustva koja se odnose na izradu onih aplikacija koje su vezane za različite poslovne informacione sisteme pretoči u jednu vrstu vodiča za buduće mlade programere.

Knjiga se razlikuje od mnoštva ostalih koje su usredsređene na opis tehnike izrade pojedinog objekta u aplikaciji. Ovde nećete naći opise tipa „Pokrenite dizajn mod, kliknite na ikonu, prevucite kursorom preko detalja….. i slično". Ova knjiga svakako nije za početnike koji nameravaju da od početka do kraja savladaju neki SUBP, odnosno Sistem za upravljanje bazama podataka. Za upotrebu ove knjige, nužno je poznavanje određenih elementarnih činjenica u vezi sa za programiranjem, ali i iskustvo u tehnikama rada.

U tekstu nije moglo da se izbegne pominjanje nekih termina pre nego što su naknadno objašnjeni u nastavku, odnosno u poglavljima koja slede Stoga, molim čitaoce za strpljenje i da sa pažnjom pročitaju celokupan tekst. Takođe su dati primeri iz najrazličitijih baza podataka, te je bilo bespredmetno postavljati uz knjigu izvore podataka na CD-u. Uostalom, cilj nije ni bio voditi čitaoca za ruku, korak po korak do rešenja nekog konkretnog zadatka, već da mu se ukaže na mogućnosti u izradi aplikacije, koje funkcionalnosti koristiti i koji redosled radnji treba ispoštovati.

Knjiga bi trebalo da zadovolji kriterijume vežbi iz predmeta IT Servis menadžment na Računarskom Fakultetu.

Autor

Napomena: Pojedine reči u tekstu koje označavaju nazive promenljivih, pisane su bez naših znakova (č,ć,đ,š,ž), jer ih VBA editor ne podržava. Nazivi i imenovani objekti su pisani pod navodnicima, dok su službene reči VBA programskog jezika i engleski nazivi pisani kurzivom.

Uvod

Aplikativno modeliranje je deo životnog ciklusa informacionog sistema zasnovanog na računaru. Kao takvo, pod nazivom „*faza razvoja*", dolazi nakon projektovanja, što podrazumeva da šema baze podataka na logičkom i fizičkom nivou već postoji. Stoga ćemo se nadalje ponašati kao da je faza projektovanja završena i da baratamo sa postojećom bazom u okviru odabranog SUBP-a, odnosno Sistema za upravljanje bazama podataka.

Za SUBP je ovom prilikom odabran MS ACCESS. Razlog tome je što ovaj sistem za upravljanje bazama podataka pruža najbolje pogodnosti za edukaciju, te sadrži u jednom kontejneru sve objekte kao što su tabele, obrasci, upiti, izveštaji, makroi i moduli. Pored toga, unutar ovog SUBP-a je inkorporiran editor za pisanje kôda programskog jezika VBA. Ovaj programski editor pomaže da se uradi automatizacija događaja nad pojedinim nabrojanim objektima.

Na narednim stranicama se nećemo baviti onim što je predmet informacionog modeliranja podržanog relacionom teorijom i upotrebom CASE alata zasnovanih na IDEF1X standardu i notaciji. Pretpostavka je da je čitalac kompletno savladao projektovanje informacionog sistema, pa da mu je preostala realizacija same aplikacije kroz fazu razvoja interfejsa.

Najveći broj slika prikazanih u ovoj knjizi biće iz verzije MS Access 2016. Iako je za ovu verziju dostupna lokalizacija na srpskom jeziku, autor je smatrao da prevod pojedinih opcija i funkcionalnosti nije najbolji, te je odlučio da se zadrži na originalnoj engleskoj verziji. Sam koncept knjige odskače od bilo kojeg priručnika iz MS Access-a koji ste do sada imali prilike da vidite. Pažnja je u mnogome posvećena načinu realizacije korisničkog interfejsa, kako bi se u što većoj meri zadovoljio krajnji korisnik. Iz tog razloga će za rešavanje istog problema biti predloženo više načina i varijanti.

Sve što bude bilo pomenuto i vezano za MS Access, izvodljivo je i u drugim sistemima za upravljanje bazama podataka.

1. Osnove izrade aplikacije

U osnovne elemente izrade korisničke aplikacije spadaju: tabele u kojima se čuvaju podaci, upiti koji najčešće predstavljaju isfiltrirane podatke u posebnim objektima, potom forme za unos, jer je udobnije i sigurnije to raditi nego direktno kroz tabele, izveštaji koji se zasnivaju na tabelama, a još češće na upitima, i na kraju dolaze makroi kao najjednostavniji nivo automatizacije. Osim pomenutih objekata, za konačni izgled aplikacije nužan je i korisnički interfejs za koji ćemo predložiti komandne table, menije i ribon trake. Pojam aplikacija se u ovom tekstu koristi da označi nešto što u sebi sadrži, osim progranskog kôda, još i prateće datoteke kao što su slike, zvuci, tekst i slično, dok se pojmom progam opisuje kôd koji saopštava računaru šta da radi.

1.1 Tabele i ograničenja na nivou tabela

Tabele nemaju svojstvo događaja, ali imaju mnoge druge osobine koje umanjuju potrebu za kasnijim programiranjem. Jasno je da nema smisla, a ni potrebe programirati nešto što je proizvođač već ugradio u ovaj SUBP.

Tip podataka određuje vrstu vrednosti koje korisnici mogu da skladište u poljima. Polja su presek kolona i zapisa u jednoj tabeli. Ukoliko prilikom generisanja šeme baze podataka u fizičku bazu niste potpuno zadovoljni tipom podatka i veličinom budućih polja, pravi je trenutak da, prilikom prvog ulaska u novo kreiranu bazu podataka, pristupite eventualnoj reviziji.

Čitaocima skrećemo pažnju da su tipovi i podtipovi podataka u različitim SUBP-ovima različiti više po nazivu nego suštinski, te da ako naučite skup iz jednog, imate dobre šanse da se za kratko vreme prilagodite drugom SUBP-u.

Prilikom razmatranja tipova podataka za kolone koje će biti primarni ključevi pojedinih tabela, valjalo bi da imate na umu koliki je eventualni ukupni broj budućih zapisa u toj tabeli. Najčešće ćete biti u prilici da se opredeljujete za Integer ili Long

Integer kao celobrojne vrednosti. Prvi bi trebalo koristiti kada pretpostavljate da vam neće trebati više od tridesetak hiljada zapisa, a drugi kada niste sigurni da taj broj neće biti prekoračen. Vrednost Long Integer polja je 4 bajta i duplo je veća od prethodnog, ali je opseg čak preko 2 milijarde zapisa, što je malo verovatno da ćete doseći.

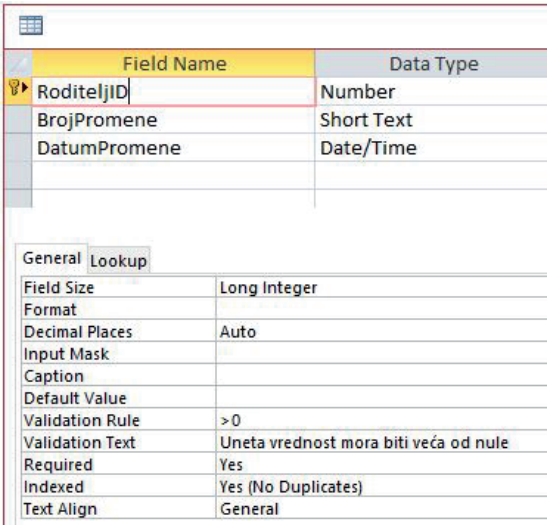
Treći tip celobrojnog podataka koji se najčešće koristi kod primarnih ključeva je AutoNumber i njega, u osobini *New Values*, možete definisati kao Incremet (sa priraštajem) ili Random (slučajni). AutonNumber je takođe dugački ceo broj kao i Long Integer, ali savetujem da AutoNumber koristite kod tabela koje su samostalne ili služe za testiranje, kada želite da sistem vodi računa o primarnom ključu i liši vas svake obaveze da ga unosite i o njemu razmišljate. Međutim, ova zadnja opcija nije dobra kod povezanih tabela, jer povezivanje traži da polja na obe strane veze (primarni i preneseni ključ) budu istog tipa i veličine.

Neke od osobina koje predlažem da razmotrite kada su tabele podataka u pitanju nalaze se u tekstualnoj tabeli na sledećoj strani, pod oznakom Tabela1. Sve što podesite sada, u situaciji dok koristite tabele baze podataka u Design View modu, višestruko će vam olakšati kasniji posao oko izrade interfejsa.

Prva osobina u nizu na donjoj slici, *Field Size* omogućava izbor i definisanje dužine za odabrani tip podatka. U zavisnosti od odabrane osobine podataka, kartica General menja svoj izgled u pogledu ponuđenih opcija, a desno u nastavku ove Slike 1, postoji objašnjenje svake od opcija.

Sledeća stvar na koju želim da vam skrenem pažnju je osobina *Format* na kartici General u Design View modu tabele. Ova osobina određuje oblik u kome se podatak pojavljuje na ekranu. U zavisnosti od tipa podataka (Date/Time, Number, Yes/No....) koji ste predvideli za pojedinu kolonu tabele, biće dostupne specificirane padajuće liste sa predefinisanim formatima. Kada je u pitanju kolona tipa Text, možete je ograničiti unosom vrednosti od 1 do 255 karaktera.

Savet: Ukoliko imate potrebu da resetujete polje AutoNumber u nekoj tabeli, onda je ispraznite i na ribon kartici Database Tools, primenite opciju Compact and Repair database.

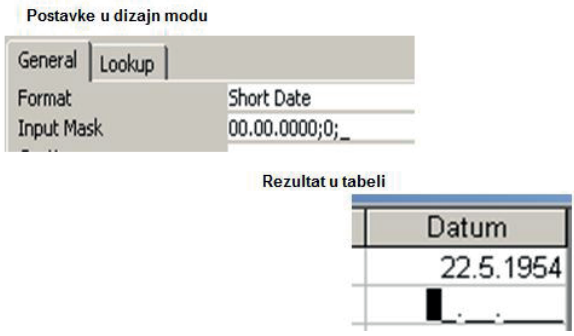


Slika 1.

Tabela 1.

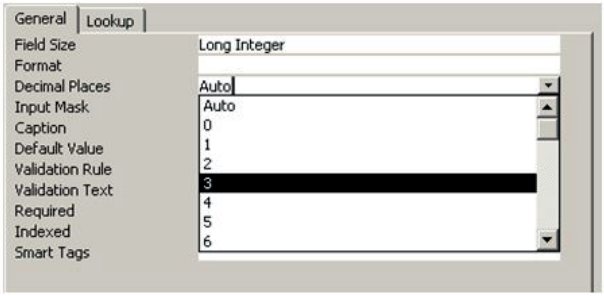
Field Size	Tekstualna vrednost: Ograničava širinu polja na zadati broj znakova (1-255), podrazumeva se 255.
Format	Određuje oblik u kome se podatak pojavljuje na ekranu.
Input Mask	Koristi se za unos podataka u unapred definisanom i proverenom formatu.
Decimal Places	Određuje broj decimalnih mesta.
Caption	Fakultativni tekst koji se pojavljuje umesto imena polja na obrscima i izveštajima ali i u tabelama.
Default Value	Vrednost koja se automatski upisuje u polje kada se doda novi zapis.
Validation Rule	Provera ispravnosti one vrednosti koja se unosi u polje na osnovu pravila koja definišete pomoću izraza ili makroa.
Validation Text	Tekst poruke koja se pojavljuje kada podatak u polju ne zadovoljava pravila ispravnosti.
Required	Određuje da li polje mora da bude popunjeno ili ne.
Allow Zero Lenght	Određuje da li je dozvoljen upis vrednosti " " u tekstualno polje radi razlikovanja od vrednosti Null.
Indexed	Ubrzava pristup podacima i (ako tako zadate) ograničava podatke na jedinstvne vrednosti.
Unicode Compression	Koristi se u aplkacijama koje treba da rade na više jezika. Potrebno je oko dva puta više mesta za smeštanje podataka, ali omogućava da se Office-ovi dokumenti, uključujući i Access-ove) izveštaje, prikazuju sa ispravnim znakovima, bez obzira na jezik ili pismo koje se koristi.

Osobina *Input Mask* se koristi za unos podataka u unapred definisanom i proverenom formatu. Datumska polja formatirajte sa Short date, a u *Input mask* staviti 00.00.00;0;0; da bi se pokazivale nule prilikom unosa u polje ili 00.00.00;0; da bi se pokazivale crtice prilikom unosa u polje, kao što je to slučaj u primeru na sledećoj slici.



Slika 2.

Osobina *Decimal Places* određuje broj decimalnih mesta. Kada su u pitanju Number tip podataka i formati predviđeni za njega, u zavisnosti od potrebe, možete birati broj decimala (od 0 do 15) kojima će Access operisati.



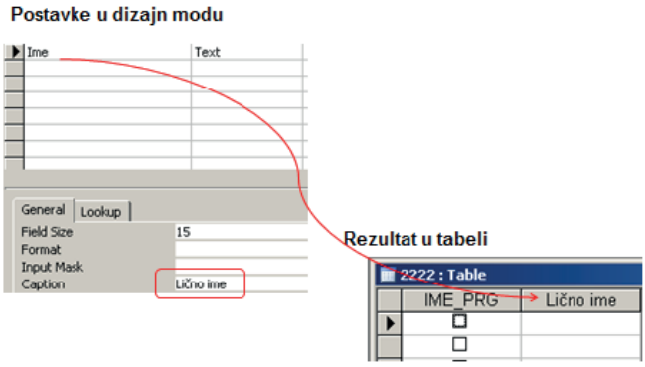
Slika 3.

Access fajl i importujte sve objekte iz starog fajla (kontejnera) u novi.

Savet: Polje za broj telefona napravite tako da u Text box ugradite Input mask sledeće sadržine: `\+###" ("##") "###\-####;0;_`

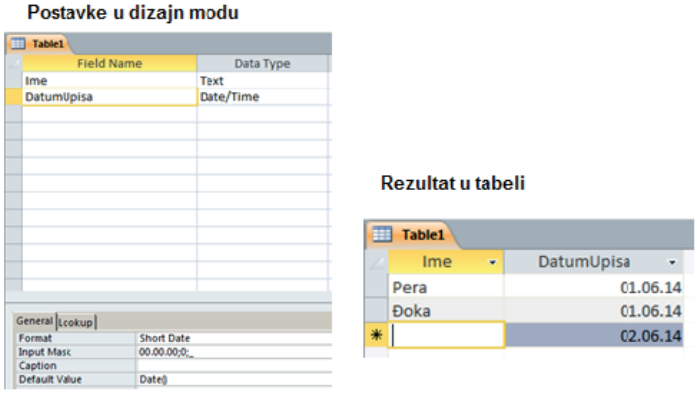
Savet: Ukoliko vam se dogodi da aplikacija nakon nekog vremena ne radi korektno i ne možete da otklonite grešku poznatim metodama, napravite novi

Osobina *Caption* omogućuje fakultativni tekst koji se pojavljuje umesto imena polja na obrascima i izveštajima ali i u tabelama. Na slici 4 vidite kako se tekst: „Lično ime", upisan na *General* kartici u režimu *Design View* tabele, manifestuje kada tabelu pokrenete u režimu *Datasheet View*. Dakle,u strukturi tabele možete imati jedan naziv kao zvaničan, a taj naziv ćete koristiti kod bilo kojeg imenovanja i obraćanja pomenutoj tabeli prilikom programiranja, a drugi naziv za krajnjeg korisnika aplikacije. Ovo je veoma korisno kada imate nasleđeni sistem tabela iz nekog drugog SUBP-a, kao što su Clipper, Fox, dBASE i slični.



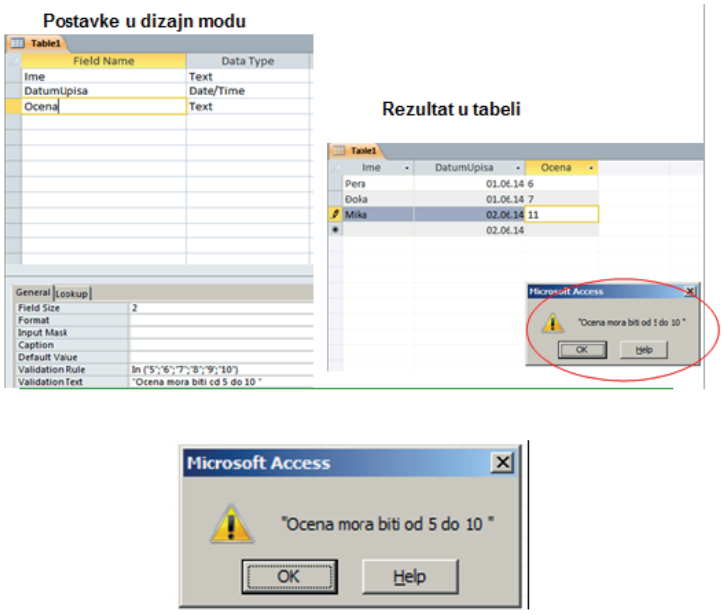
Slika 4.

Osobina *Default Value* je od pomoći krajnjem korisniku aplikacije. To je vrednost koja se automatski upisuje u polje kada se doda novi zapis. U primeru na slici, rezultat u tabeli je ponuđen tekući datum. U momentu unosa to je bio 02.06.2014. Korišćena je standardna funkcija `Date()`. U zadnjem prikazanom redu kartice General na slici ispod, korišćena je standardna funkcija `Date()`.



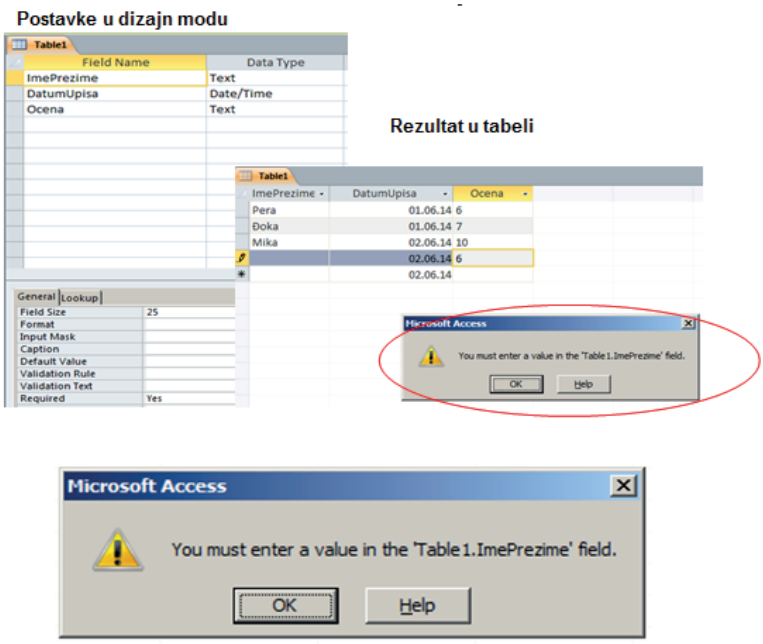
Slika 5.

Kako biste sprečili neispravan unos podataka u polje, koristite sledeće dve osobine. Provera ispravnosti te vrednosti je *Validation Rule*, koja se unosi u polje na osnovu pravila koja definišete pomoću izraza ili makroa. Ova osobina je usko povezana sa osobinom *Validation Text*, odnosno tekstom poruke koja se pojavljuje kada podatak u polju ne zadovoljava pravila ispravnosti. Na sledećoj slici vidite da je za pravilo provere ispravnosti vrednosti upotrebljena lista `IN()` i u njoj navedene ocene koje student može da dobije na ispitu. U polju *Validation Text* je upisan odziv koji Access upućuje krajnjem korisniku ukoliko pokuša da unese vrednost izvan liste. Ovaj odziv dobija krajnji korisnik u obliku poruke na ekranu nakon zaustavljanja izvršenja, što takođe vidite na donjoj slici.



Slika 6.

Možete insistirati da pojedina polja u tabeli moraju biti popunjena pre nego što korisnik aplikacije napusti tekući zapis. Tada na nivou tabele i njenih ograničenja treba da koristite osobinu *Required*. Ova osobina određuje da li polje mora da bude popunjeno ili ne. Ukoliko je ta osobina postavljena na *Yes*, pri pokušaju da se napusti zapis (red) u kojem zahtevano polje nije popunjeno, MS Access javlja poruku i ne dozvoljava da se zapis kao takav sačuva. Ova poruka je na engleskom (slika 7.) i s obzirom na to da je sistemska, vi možete da je promenite ukoliko želite. Ovo se radi kroz procedure za obradu grešaka, ali o tome ćete saznati više pri kraju ove knjige, kada se budemo bavili modulima i procedurama.



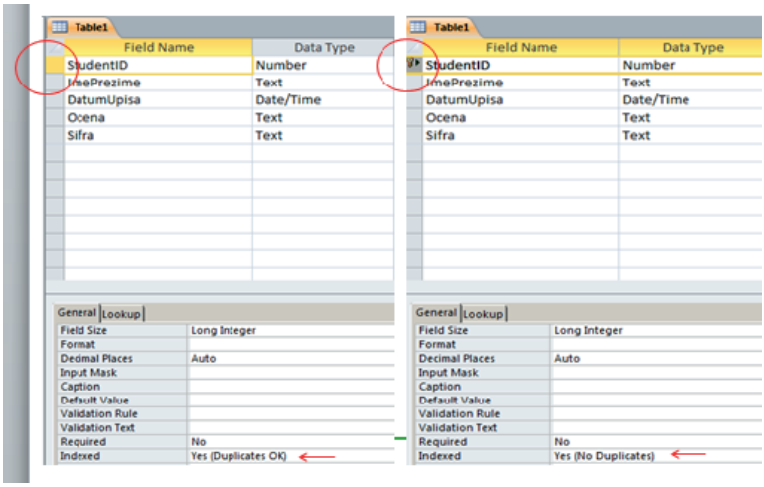
Slika 7.

Osobina *Allow Zero Lenght* određuje da li je dozvoljen upis vrednosti " " (prazan string) u tekstualno polje radi razlikovanja od vrednosti Null ili nepopunjenog polja. Ova se osobina retko upotrebljava ali može da bude korisna.

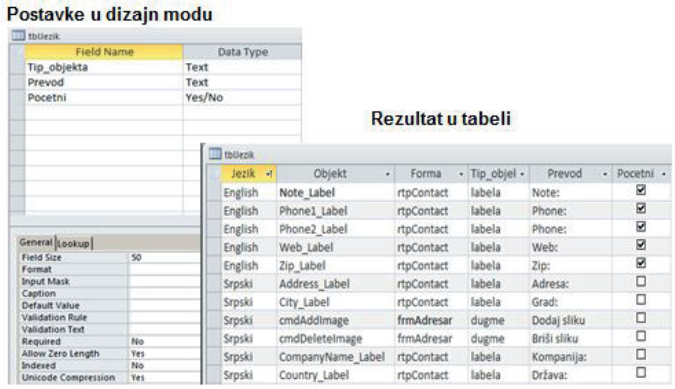
Osobina *Indexed* ubrzava pristup podacima i ako tako zadate, ograničava podatke na jedinstvene vrednosti. Ako kolona nije PK, odnosno primarni ključ, duplikati su dozvoljeni. Ukoliko je kolona PK, duplikati nisu dozvoljeni. Slika 8 pokazuje obe varijante.

Osobina *Unicode Compression* je osobina tabele koja se koristi u aplkacijama koje treba da rade na više jezika. Potrebno je oko dva puta više mesta za smeštaj podataka, ali omogućava da se Office-ovi dokumenti, uključujući i Access-ove izveštaje, prikazuju sa ispravnim znakovima bez obzira na jezik ili pismo koje se koristi.

Upozorenje: Ne mogu da se održavaju relacije nad tabelama iz dve baze (dve .mdb ili dve .accdb datoteke). Zbog toga je najbolje primenjivati linkovanje tabela tako da se *back-end* tabele ulinkuju u *front-end* aplikaciju koja predstavlja korisnički interfejs.



Slika 8.



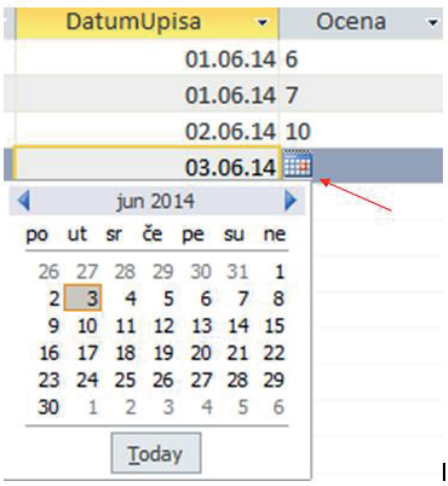
Slika 9.

Postoje još neke osobine tabela za koje nećemo dati primere, ali su jednako korisne u datim situacijama i potvrđuju da nema smisla ni potrebe programirati nešto što je proizvođač već ugradio u ovaj SUBP:

- *IME mode* – podešavanje režima za premeštanje fokusa u polju,
- *Smart Tags* – dostupne pametne oznake za tipove polja,
- *Tekst Allign* – poravnanje teksta u polju,
- *Show Data Picker* – prikazivanje pomoćnog kalendara.

Savet: Ako je baza *Read-Only*, potrebno je da ona bude ponovo sačuvana, korišćenjem sledeće opcije: File, Save As, kao normalnu arhivsku kopjiu, uz eventualnu promenu naziva.

Za ovu zadnju osobinu ćemo dati neke napomene. U starioj verziji 2003 MS Access-a, funkcionalnost DatePicker je bila moguća uz dodatnu formu i VBA programski kôd u modulu. *DataPicker* postoji kao ugrađena funkcionalnost od verzije MS Access 2007 pa nadalje. Potrebno je da u tabeli, na polje tipa *Date/Time*, podesiti osobinu *Show Date Picker* na vrednost *For Dates*. Što je, uostalom, i po defaultu (prev. *podrazumevano*). Kod starih fajlova, tipa .mdb, pokrenutih u novijim verzijama MSAccess-a, postojeća datumska polja u tabelama nemaju tu funkcionalnost. Međutim, ako se podaci iz starog datumskog polja prebace u novokreirano datumsko polje, potom se staro polje obriše a novom se dodeli naziv starog, tada sve funkcioniše kako treba.



Slika 10.

Sve ove osobine ćete naći i u drugim SUBP-ovima kao što su MS SQL Server, MySQL ili Oracle. Kada ovladate jednim, učenje sledećeg sistema za upravljanje bazama podataka će biti mnogo lakše.

Savet: Tokom razvoja aplikacije, redovno pravite arhivske kopije i imenujte ih tako da sadrže datum, a po potrebi, i vreme izrade. Primer: (VašaAplikacija 20130919.mdb). Tako ćete izbeći razne neprijatnosti koje donosi razvoj aplikacije.