

# Osnove programiranja u Accessu

- 1** Access kao alat za programiranje
- 2** Šta svaki programer treba da zna o bazama podataka i tabelama
- 3** Zavisnosti: Ključ za integritet podataka
- 4** Šta svaki programer treba da zna o osnovama zadavanja upita
- 5** Šta svaki programer treba da zna o formularima
- 6** Šta svaki programer treba da zna o izveštajima
- 7** VBA: Uvod
- 8** Objekti, svojstva, metodi i događaji
- 9** Napredne tehnike rada sa formularima
- 10** Napredne tehnike rada sa izveštajima
- 11** Napredne tehnike rada sa upitima
- 12** Napredne tehnike rada sa programskim jezikom VBA
- 13** Korišćenje prednosti modula klase
- 14** Šta su objekti ActiveX Data Objects i Data Access Objects i zbog čega su važni?



# Access kao alat za programiranje

### Zbog čega je ovo poglavlje važno

U razgovorima sa korisnicima i programerima sam saznala da je Access veoma pogrešno shvaćen proizvod. Mnogi ljudi smatraju da je to samo igračka za menadžere i sekretarice koji žele da se igraju podacima. Ostali korisnici misle da je to ozbiljan proizvod namenjen isključivo iskusnim programerima aplikacija. Ovo poglavlje će srušiti iluziju o Accessu. Pomoći će Vam da odgonetnete šta je zapravo Access, a šta nije. Pošto pročitate poglavlje, znaćete da li je Access alat koji treba da koristite, a kada se treba pozabaviti drugim proizvodima.

### Kakve tipove aplikacija možete napraviti u Accessu?

Često se nalazim u situaciji da sebi objašnjavam kakvi tipove aplikacija se mogu napraviti pomoću Microsoft Accessa. U Accessu postoji mnoštvo osobina koje zadovoljavaju različite potrebe baze podataka. Možete ih koristiti za pravljenje šest osnovnih tipova aplikacija:

- Personalne aplikacije
- Aplikacije za male firme
- Aplikacije za odeljenja firme
- Aplikacije korporacije
- Aplikacije koje su front end klijent/server aplikacija
- Aplikacije za Internet ili lokalnu mrežu

## Access kao razvojna platforma za personalne aplikacije

Na najnižem nivou, Access možete koristiti za pravljenje jednostavnih personalnih sistema za upravljanje bazama podataka. Ipak, ja upozoravam ljudе i ne nalazim da je ova ideja dobra. Ljudi koji Access kupuju kako bi automatizovali sve, od kolekcije vina do kućnih finansija, često su razočarani. Problem ležи u tome što je Access varljivo lagan za korišćenje. Njegovi sjajni ugrađeni čarobnjaci stvaraju utisak da je Access proizvod koji se lako može koristiti. Pošto odgovorite na niz pitanja, završili ste pravljenje razvodnih tabli aplikacije, ekrane za unos podataka, izveštaje i tabele u kojima se nalaze podaci. Zapravo, kada je Microsoft prvi put napravio Access, mnogi ljudi su me pitali da li se brinem da li će se moj posao programera i nastavnika smanjiti jer je Access apsolutno svakome omogućavao da napravi aplikaciju za bazu podataka. Iako jeste tačno da najjednostavnije Accessove aplikacije možete napraviti, a da uopšte ne razmišljate o strukturi i da ne napišete niti jedan red koda, za većinu aplikacija je ipak neophodno da napišete kod i da ih strukturirate.

Sve dok ste zadovoljni personalnim aplikacijama koje generiše čarobnjak, uz male izmene, neće se javiti nikakvi problemi. Problemi se mogu javiti kada poželite da značajno izmenite personalnu aplikaciju.

## Access kao razvojna platforma za poslovne aplikacije malih firmi

Access je odlična platforma za pravljenje aplikacija koje mogu koristiti male firme. Njegovi čarobnjaci omogućavaju programerima da brzo i lako naprave osnovu aplikacije. Mogućnost pravljenja modula koda omogućava programerima pravljenje biblioteka funkcija koje se mogu koristiti na više mesta, a mogućnost pravljenja koda formulara i izveštaja im omogućava da naprave moćne formulare i izveštaje.

Osnovna ograničenja korišćenja Accessa za pravljenje aplikacija za male firme jeste vreme i novac koji je potreban u postupku pravljenja aplikacije. Mnogi ljudi koriste Accessove čarobnjake kako bi započeli postupak pravljenja aplikacije, ali ih moraju prilagođavati na načine koji ne mogu sami da obave. Vlasnici malih firmi često imaju ovakav problem na mnogo višem nivou. Zahtevi aplikacije male firme su obično mnogo veći od zahteva personalne aplikacije. Mnogi doktori, advokati i drugi profesionalci su me zvali pošto su zapali u čorsokak postupka pravljenja aplikacije. Oni su uvek užasnuti saznanjem koliko novca je potrebno da bi njihove aplikacije mogle da se osposobe za korišćenje.

## Access kao razvojna platforma za aplikacije odeljenja

Access je savršen alat za pravljenje aplikacija za odeljenja velike korporacije. Relativno je lako nadograditi korisničke računare po odeljenjima tako da imaju odgovarajući hardver - na primer, mnogo je lakše kupiti još RAM memorije za 15 korisnika nego za 4.000!

Osim toga, Accessove performanse su adekvatne za većinu aplikacija koje bi se koristile po odeljenjima bez potrebe za korišćenjem klijent/server tehnologije. Najzad, većina odeljenja velikih korporacija ima razvojne budžete koji omogućavaju pravljenje dobrih aplikacija. Na sreću, u mnogim odeljenjima postoji stručnjak za računare koji je voljan da pomogne u pravljenju formulara i izveštaja. To odeljenjima stvara utisak vlasništva jer su dali svoj doprinos pravljenju aplikacije. To takođe moj posao programera čini lakšim. Ja se mogu usresrediti na teže probleme programiranja, ostavljajući pri tom neke poslove strukturiranja formulara i izveštaja talentima koji rade u firmi.

## Access kao razvojna platforma za aplikacije korporacije

Iako je Access možda najpogodniji za pravljenje aplikacija koje će se koristiti u odeljenjima firme, možete ga koristiti za pravljenje aplikacija koje će se koristiti u celoj firmi. Koliko je uloženi trud vredan, zavisi od korporacije. Postoji granica broja korisnika koji istovremeno (konkurentno) mogu da koriste Accessovu aplikaciju, a da pri tome ona zadrži prihvatljive performanse, a takođe postoji i ograničenje broja sloganova neke tabele, a da se značajno ne degradiraju performanse. Ovi brojevi zavise od sledećih faktora:

- Koliko mrežnog saobraćaja već postoji u mreži?
- Koliko RAM memorije i procesora postoji u serveru?
- Kako se server već koristi? Na primer, da li se aplikacije, kao što je Microsoft Office, učitavaju sa servera ili sa lokalnih radnih stanica?
- Kakve vrste poslova korisnici aplikacije obavljaju? Da li postavljaju upite, unose podatke, prave izveštaje i tako dalje?
- Gde se Access i Accessove aplikacije izvršavaju, na serveru ili radnoj stanici?
- Koji se mrežni operativni sistem koristi?

Moje opšte pravilo za Accessove aplikacije koje nisu klijent/server bazirane jeste da se loše performanse dobijaju kada ima više od 10-15 istovremenih korisnika i kada postoji više od 100.000 sloganova. Zapamtite, ovi brojevi u mnogome zavise od faktora koje sam pomenula, kao i od definicije prihvatljivih performansi koje ste Vi ili Vaši korisnici postavili. U Poglavlju 20, "Pravljenje višekorisničkih aplikacija i aplikacija za reduzeće", objašnjeno je kada treba preći na klijent/server bazu podataka. Više detalja o ovoj temi možete naći u mojoj knjizi "Ovladavanje klijent/server programiranjem u Accessu 11, Alison Balter" (*Alison's Balter's Mastering Access 11 Client/Server Development*), takođe u izdanju Samsa.

Programeri često ne razumeju šta je zapravo Access, a šta nije, kada se posmatra klijent/server platforma baze podataka. Ljudi mi često postavljaju pitanje: "Nije li Access klijent/server baza podataka?" Odgovor je da je Access neobičan proizvod jer je to aplikacija servera datoteka, ali se može koristiti kao front end za klijent/server bazu podataka. Ukoliko niste razumeli, evo objašnjenja: ukoliko kupite Access i napravite aplikaciju koja podatke čuva u Accessovoj bazi podataka na serveru datoteka, onda se na radnim stanicama obavlja sva obrada podataka. To znači da svaki put kada korisnik zada upit ili zatraži izveštaj, server datoteka radnoj stanici šalje sve podatke. Radna stanica zatim izvršava upit, a rezultat prikazuje u tabelarnom prikazu ili izveštaju. Ovakav postupak generiše značajan mrežni saobraćaj, naročito ukoliko više korisnika istovremeno traži izveštaje ili zadaje upite nad velikim Accessovim tabelama. Zapravo, ovakve operacije celu mrežu mogu potpuno zagušiti.

## Access kao front end za klijent/server aplikacije preduzeća

Klijent/server baza podataka, kakve su Microsoft SQL Server ili Oracle, upite obrađuje na serverskom računaru, a rezultate šalje radnoj stanici. Pošto serverski softver ne može korisniku prikazati podatke, tu u pomoć priskače Access. Ponašajući se kao front end, Access može u izveštajima, tabelama ili formularima prikazati podatke dobijene sa servera baza podataka. Ukoliko korisnik ažurira podatke u Accessovom formularu, radna stanica šalje izmene back-end bazi podataka. Ovaj postupak možete sprovesti ukoliko ostvarite vezu sa spoljašnjim bazama podataka tako da Vama i korisniku izgleda kao da radite sa Accessovim tabelama, ili korišćenjem tehnike kojima se klijent/server podacima direktno pristupa.

Koristeći Accessove datoteke projekta (nemojte ih pomešati sa datotekama Microsoft Project), možete napraviti aplikaciju koja je namenjena klijent/server okruženju. Ove datoteke projekta, koje su poznate kao Access Data Project (ADP) datoteke, zadržavaju formulare, izveštaje, makroe, module i stranice za rad sa podacima programa. Projekat je povezan sa back-end bazom podataka u kojoj se nalaze tabele, uskladištene procedure, pogledi i dijagrami baze podataka koje program koristi. Iz datoteke projekta lako možete menjati i manipulisati objektima koji se nalaze na serveru, koristeći Access grafički korisnički interfejs. ADP datoteke pomažu da brzo pravljenje aplikacija dovedete do pravljenja klijent/server aplikacija. Pošto Access 11 ima ugrađen način zapisivanja podataka (SQL Server 2000 Desktop Engine), klijent/server aplikacije možete napraviti na jednom računaru, a zatim ih lako prenesti u SQL Server bazu podataka. U Poglavlju 20 su ukratko objašnjeni alternativni načini i tehnike pravljenja klijent/server aplikacija. Detaljne informacije o pravljenju Accessovih projekata možete naći u knjizi "Ovladavanje klijent/server programiranjem u Accessu 11, Alison Balter".



### Napomena

ADP datoteke su se prvi put pojavile u Accessu 2000. Mnogi ljudi ih smatraju verzijom 1.0 te tehnologije. Microsoft je značajno radio na ADP datotekama u Accessu 10. Danas su ADP datoteke rešenje za pravljenje klijent/server aplikacija.

Kada smanjite količinu mrežnog saobraćaja tako što obradu upita prebacite na back-end, Access postaje mnogo moćnije razvojeno rešenje. Tada Access može obraditi velike količine podataka i veliki broj konkurentnih korisnika. Osnovna pitanja sa kojima se programeri koji žele da naprave takve Accessove aplikacije obično suočavaju su:

- Raznolikost operativnih sistema koje korisnici mogu da koriste.
- Problemi u pravljenju aplikacije.
- Način na koji svaki korisnik uspostavlja vezu sa aplikacijom.
- Vrsta hardvera svakog korisnika.

Iako se obrada upita u klijent/server aplikaciji obavlja na serveru, čime se značajno smanjuje mrežni saobraćaj, sama aplikacija se ipak mora nalaziti u memoriji računara svakog korisnika. To znači da svaki klijentski računar mora da bude takav da se na njemu može izvršavati odgovarajući operativni sistem i odgovarajuća verzija Accessa. Čak i kada se na klijentskom računaru izvršava odgovarajući operativni sistem i kada na njemu postoji odgovarajuća verzija Accessa, i dalje mogu nastati problemi. DLL (Dynamic Link Libraries) konflikti su često rezultat grešaka koje se teško otkrivaju i idiosinkrazija koje se javljaju u Accessovoj aplikaciji. Osim toga, Access nije najbolje rešenje za korisnike koji nemaju stalnu vezu sa bazom podataka, a koji preko Interneta moraju uspostaviti vezu sa aplikacijom i njenim podacima. Najzad, Access 11 ima velike zahteve za hardverom! Hardverski zahtevi Accessove aplikacije se objašnjavaju kasnije u ovom poglavlju. To znači da, pre nego što odlučite da prosledite veliku Accessovu aplikaciju, morate znati softversku i hardversku konfiguraciju svih sistemskih korisnika. Takođe, morate proceniti da li podrška koja se zahteva za tipičnu Accessovu aplikaciju zadovoljava zadati broj ljudi koji koriste sistem koji pravite.

## **Access kao razvojna platforma za aplikacije u lokalnoj mreži/Internetu**

Koristeći stranice za rad sa podacima, korisnici Interneta i lokalne mreže mogu ažurirati podatke aplikacije pomoću pretraživača (browser). Stranice za rad sa podacima su HTML dokumenti koji su direktno povezani sa podacima u bazi podataka. Iako Access stranice za rad sa podacima zapisuje van baze podataka, Vi ih koristite na isti način na koji koristite standardne Accessove formulare, izuzev što ih je Microsoft dizjanirao tako da se izvršavaju u Microsoft Internet Exploreru 5.5 ili boljoj verziji, a ne u Microsoft Accessu. Stranice za rad sa podacima za obavljanje posla koriste dinamički HTML. Pošto se mogu koristiti samo u Microsoft Internet Exploreru 5.5 ili boljoj verziji, stranice za rad sa podacima su mnogo pogodnije rešenje za lokalne mreže nego za Internet.

Pored korišćenja stranica za rad sa podacima, objekte baze podataka možete izdavati kao statičke ili dinamičke HTML stranice. Statičke stranice su standardni HTML dokumenti čiji se sadržaj može prikazati u bilo kom pretraživaču.

Objekte baze podataka možete izdavati dinamički u HTX/IDC formatu datoteka ili u ASP (Active Server Page) formatu datoteka. Web server HTX (HTML Extension)/IDC (Internet Database Connector) format datoteka izdaje dinamički te stoga ne zavisi od pretraživača. Web server takođe dinamički izdaje ASP datoteke koje se prave u Accessu, ali za ASP datoteke na klijentskom računaru morate koristiti Internet Explorer 4.0 ili bolju verziju.

U Accessu 10 je prvi put uvedena mogućnost pravljenja XML podataka i šema na osnovu Jetovih ili SQL Serverovih struktura i podataka. Takođe, iz XML dokumenata u Access možete uvoziti podatke i strukture podataka. To možete obaviti korišćenjem koda ili korisničkog interfejsa.

**Napomena**

U ovoj knjizi se donekle razmatra pravljenje aplikacija za lokalne mreže i Internet pomoću Microsoft Accessa. Detaljan opis pravljenja aplikacija za lokalne mreže i Internet pomoću Microsoft Accessa se razmatra u knjizi "Ovladavanje klijent/server programiranjem u Accessu 11, Alison Balter".

## Access kao proizvod promenljive veličine

Jedna od jačih strana Accessa jeste promenljiva veličina njegovih aplikacija. Aplikaciju možete prvo napraviti kao poslovnu aplikaciju koja se izvršava na jednom računaru pa je zatim proširiti u klijent/server aplikaciju firme. Ukoliko pravilno strukturirate aplikaciju, postupak njenog proširivanja možete obaviti tako da je malo ili nimalo ne prepravljate. Ova osobina čini Access odličnim izborom za firme koje proširuju svoje poslovanje, kao i za aplikacije koje testirate na nivou odeljenja imajući pri tom na umu da ih kasnije distribuirate na nivou cele firme.

Ono što Access čini izuzetnim jeste da, čak i kada se koristi kao front end i back-end sa podacima u Accessovim tabelama na serveru datoteka, omogućava izuzetnu bezbednost i mogućnost uspostavljanja pravila baze podataka koja su se ranije mogla koristiti samo u back-end bazama podataka. U Poglavlju 27, "Bezbednost baze podataka je lako ostvariti", i Poglavlju 28, "Napredne bezbednosne tehnike", naučićete kako da obezbedite svaki objekat baze podataka bilo na nivou korisnika ili nivou grupe korisnika. Pravila referencijalnog integriteta možete primeniti na nivou baze podataka, obezbeđujući (na primer) tako da korisnici ne mogu da unose narudžbine za klijente koji ne postoje. Pravila provere podataka možete primeniti bilo na nivou polja ili nivou sloga, održavajući tako integritet podataka baze podataka. Drugim rečima, mnoge osobine koje su ranije postojale samo serverima baza podataka sada se mogu koristiti zahvaljujući Accessovom načinu zapisivanja podataka.

## Šta je to ustvari baza podataka?

Termin baza podataka za različite ljude ima različito značenje. Tokom mnogo godina, u svetu xBasea (dBase, FoxPro, CA-Clipper), termin baza podataka se koristio za opisivanje kolekcije polja i slogova. (U Accessu se ovaj tip kolekcije naziva tabela.) U klijent/server okruženju, termin baza podataka se odnosi na sve podatke, šeme, indekse, pravila, okidače i uskladištene procedure koje su pridružene sistemu. U Accessu je baza podataka kolekcija svih tabela, upita, formulara, stranica za rad sa podacima, izveštaja, makroa i modula koji sačinjavaju kompletan sistem.

# Upoznavanje objekata baze podataka

Kao što je ranije rečeno, tabele, upiti, formulari, izveštaji, stranice za rad sa podacima, makroi i moduli zajedno čine Accessovu bazu podataka. Svaki od ovih objekata ima specijalnu namenu. Accessova aplikacija takođe sadrži nekoliko različitih objekata, uključujući zavisnosti, komandne linije, svojstva baze podataka i uvozne/izvozne specifikacije. Pomoću ovih objekata možete napraviti moćne, prijateljske, integrisane aplikacije. Na slici 1.1 vidite prozor Access Database. Obratite pažnju na sedam kategorija objekata prikazanih u kontejneru baze podataka. U narednom odeljku ćete se bliže upoznati sa objektima koji čine Accessovu bazu podataka.



Slika 1.1

U prozoru Access Database se prikazuju ikone za svaki tip objekta baze podataka.

## Tabele: Skladište podataka

Tabele su polazana tačka Vaših aplikacija. Bilo da su podaci smešteni u Accessovoj bazi podataka ili pomoću povezanih tabela koristite spoljašnje podatke, svi ostali objekti baze podataka direktno ili indirektno koriste tabele.

Da biste saznali koje se sve tabele nalaze u otvorenoj bazi podataka, kliknite ikonu Tables u listi Objects. (Zapamtite da se neće prikazati sakrivene tabele izuzev ako ste potvrdili polje Hidden Objects na kartici View okvira za dijalog Options.) Ukoliko želite da vidite podatke koji se nalaze u tabeli, dva puta kliknite ime tabele čije podatke želite da vidite. (Takođe, možete označiti tabelu pa kliknuti Open.)

Access podatke tabele prikazuje u tabelarnom obliku, što znači da se prikazuju sva polja i slogovi tabele (videti sliku 1.2). Možete menjati mnoge atribute tabelarnog prikaza, a možete čak i da vršite pretraživanje i filtriranje podataka u tabeli. Ukoliko je sa tabelom povezana druga tabela (recimo, kao što su povezane tabele Northwind Customers i Orders), možete otvoriti i zatvoriti podtabelu kako biste videli podatke koji se nalaze u dete-tabelama. U ovoj knjizi se ne obrađuju te tehnike. Objašnjenje tih tehnika možete naći u Accessovom uputstvu za korisnike ili knjigama za početnike, kao što je knjiga "Naučite sami da koristite Microsoft Office Access 2003 za 24 sata" (Teach Yourself Microsoft Office Access 2003 in 24 Hours, Sams).

	Customer ID	Company Name	Contact Name	Contact Title	
#	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Owner
#	ANATR	Ana Trujillo Emparedados y helados	Aña Trujillo	Owner	Asst.
#	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Manager
#	AROUT	Around the Horn	Thomas Hardy	Sales Representative	CO-OP
#	BERGS	Berglunds snögröd	Clemence Berglund	Order Administrator	Beginner
#	BLAUSU	Blauer See Delicatessen	Hanna Moos	Sales Representative	Firstie
#	BLOOMP	Blondie pâtisserie	Frédérique Closier	Marketing Manager	24, ph.
#	BOLID	Bólido Comidas preparadas	Matilde Sonner	Owner	Cl. Ana
#	BONAP	Bon app!	Lukasine Lubitsch	Owner	12, nr.
#	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	CO-Tech
#	BEEGEM	By Bevrijding	Wistaria Ashton	Sales Representative	Faxtel
#	CACTU	Cactus Comidas para llevar	Patricia Simpson	Sales Agent	Carmen
#	CENTC	Centro comercial Multicenter	Francisco Chang	Marketing Manager	Sierra
#	CHOPS	Chop-suey Chinese	Yang Wang	Owner	Haus
#	COMM	Comisario Inmobiliaria	Pedro Adáns	Sales Associate	Av. do
#	CORCH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Belize
#	DRACD	Drachenblut Delikatessen	Sven Ottke	Order Administrator	Wales
#	DUMON	Du monde entier	Jeanne Laberge	Owner	ST, nr.
#	EASTC	Easter Connection	Karen Daron	Sales Agent	36 Km
#	ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchg.
#	FAMIA	Familia Agualardo	Aña Cruz	Marketing Assistant	Rua O
#	PISSA	PISSA, Fabrica Inter. Delicias S.A.	Diego Rivel	Accounting Manager	Cl. Me.

**Slika 1.2**

Pogled Datasheet za tabelu Customers baze podataka Northwind obuhvata sva polja i slogove tabele.

Najčešće će Vas kao programera zanimati struktura tabele koja je zapravo njen šematski plan ili šablon. Da biste videli strukturu tabele, kliknite ikonu Design kada je tabela označena (videti sliku 1.3). U pogledu Design možete videti ili izmeniti sva imena polja, tipove podataka i svojstva polja i tabele. Access Vam daje moć i fleksibilnost koji su Vam neophodni za prilagođavanje strukture tabela. U Poglavlju 2, "Šta svaki programer mora da zna o bazama podataka i tabelama", objašnjene su ove teme.

### Napomena

U ADP-u, iako Access tabele prikazuje u prozoru Database, ne zapisuje ih u Accessovoj datoteci projekta. Umesto toga, tabele zapisuje u SQL Serverovoj bazi podataka sa kojim povezujete ADP.

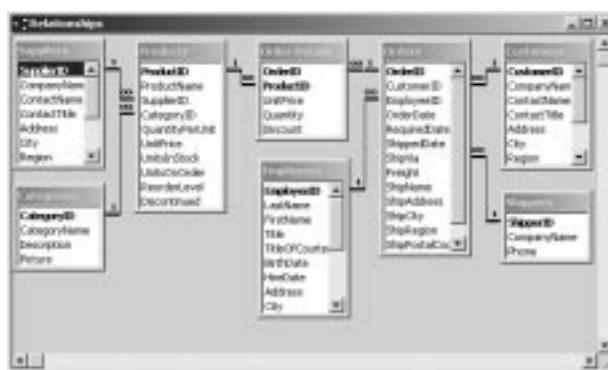
**Slika 1.3**

Struktura tabele Customers je njen šematski plan ili šablon.

### Zavisnosti: Povezivanje tabela

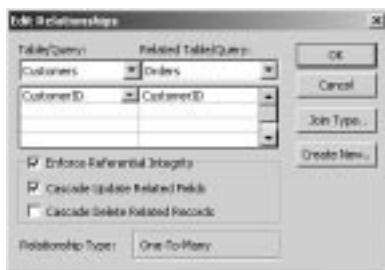
Da biste pravilno održavali integritet podataka i uprostili postupak rada sa drugim objektima baze podataka, morate definisati zavisnosti među tabelama baze podataka. To postižete korišćenjem prozora Relationships. Da biste prikazali prozor Relationships, dok je aktivan prozor Database, odaberite Tools, Relationships ili na paleti alata kliknite ikonu Relationships (videti sliku 1.4).

U ovom prozoru se prikazuju i održavaju zavisnosti u bazi podataka. Ukoliko ste Vi ili neki drugi programer definisali zavisnosti, ali u okviru za dijalog Relationships ne vidite nijednu, odaberite Relationships, Show All kako biste prikazali skrivene tabele i zavisnosti (možda ćećete morati da kliknete da biste otvorili meni za ovu opciju).

**Slika 1.4**

U prozoru Relationships prikazujete i održavate zavisnosti koje su definisane u bazi podataka.

Primećujete da mnoge zavisnosti koje vidite na slici 1.4 imaju linije koje spajaju tabele pored kojih стоји број 1 или знак за бесконачност (?). На тај начин је између табела означена зavisnost jedan-prema-više. Уколико два пута klikнете линiju, отвориће се оквир за дијалог Edit Relationships (видети слику 1.5). У овом оквиру за дијалог можете задати тачну приrodu зavisnosti између табела. Зavisnost између табела Customers и Orders, на пример, је зavisnost jedan-prema-više за коју се применије referencijalni integritet. То значи да корисник не може да дода (направи, унесе) наредњбине за купце који не постоје. Приметили сте да је поље Cascade Update Related Fields потврђено. То значи да ако корисник аžurира поље CustomerID, Access аžurира све слогове у табели Orders који садрже тај CustomerID. Пошто поље Cascade Delete Related Records nije потврђено, корисник не може да уклони купце из табеле Customers уколико за њих у табели Orders постоје одговарајуће наредњбине.



**Slika 1.5**

Оквир за дијалог Relationships Вам омогућава да одредите приrodu зavisnosti између табела.

У Поглављу 3, "Зavisnosti: Кљуč за integritet podataka", опширно се објашњава поступак definisanja i održavanja zavisnosti. У том поглављу се takođe objašnjavaju osnove структуре relacione baze podataka. За сада запамтите да зavisnosti morate uspostaviti konceptualno i doslovno што је ranije могуће у поступку структурирања. Зavisnosti су сastavni deo uspešnog структурирања i implementiranja aplikacije.



#### Napomena

У ADP-у, зavisnosti се у прозору Database приказују као диграми базе података. Не записују се у Accessовој datoteci projekta. Umesto тога се записују у SQL Serverовој бази података са којом пoveзujete ADP.

## Upiti: Uskladištena pitanja ili akcije koje применjujete na podatke

Upiti у Accessu су веома моћни и свестрани. Upiti odabiranja Вам омогућавају да прикажете, саžmete и обавите izračunavanja sa podacima koji se nalaze u tabelama. Akcioni upiti Вам омогућавају да унесете i аžurirate podatke u tabelu i izbrišete ih iz табеле.

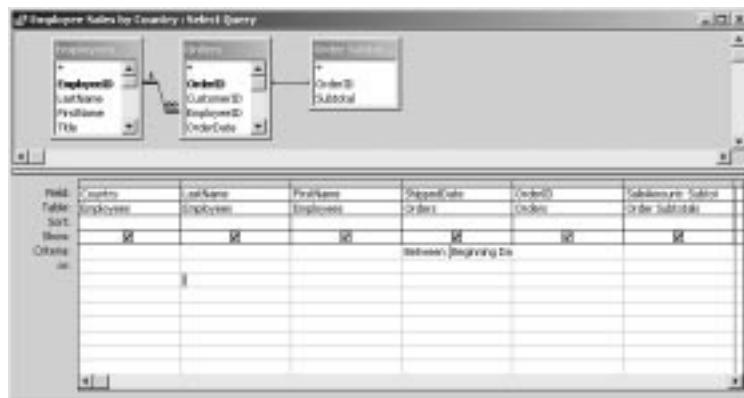
Da biste izvršili upit, iz spiska Objects odaberite Queries, dva puta kliknite upit koji želite da izvršite, ili označite upit koji želite da izvršite, a zatim kliknite Open. Pošto izvršite odabrani upit, u tabelarnom obliku će se prikazati sva polja koja su zadata upitom i svi slogovi koji zadovoljavaju kriterijum upita (videti sliku 1.6). Pošto pokrenete akcioni upit, Access obavlja zadatu akciju, recimo, pravljenje nove tabele ili dodavanje podataka u postojeću tabelu. U principu, podatke koje dobijate kao rezultat upita možete ažurirati jer je rezultat upita zapravo dinamički skup slogova, nazvan dynaset, na osnovu podataka tabele.

Country	Last Name	First Name	Shipped Date	Order ID	Sale Amount
USA	Callahan	Laura	10-Jan-1997	10380	\$1,313.92
USA	Fuller	Audrina	01-Jan-1997	10382	\$1,460.00
USA	Davolio	Francy	03-Jan-1997	10393	\$2,566.95
USA	Davolio	Nancy	03-Jan-1997	10394	\$4,624.00
UK	Deyerle	Michael	03-Jan-1997	10395	\$2,122.92
USA	Davolio	Nancy	06-Jan-1997	10396	\$1,003.00
UK	Docherty	Steven	03-Jan-1997	10397	\$716.72
USA	Fuller	Audrina	03-Jan-1997	10398	\$2,505.63
USA	Callahan	Laura	03-Jan-1997	10399	\$1,795.60
USA	Davolio	Nancy	10-Jan-1997	10400	\$3,083.00
USA	Davolio	Nancy	10-Jan-1997	10401	\$3,089.60
USA	Callahan	Laura	10-Jan-1997	10402	\$2,713.93
USA	Pacewski	Margaret	09-Jan-1997	10403	\$665.01
USA	Fuller	Annette	08-Jan-1997	10404	\$1,591.25
USA	Davolio	Nancy	20-Jan-1997	10405	\$490.00
UK	King	Robert	13-Jan-1997	10406	\$1,030.78
USA	Fuller	Annette	30-Jan-1997	10407	\$1,194.00
USA	Callahan	Laura	14-Jan-1997	10408	\$1,632.40
USA	Levenstein	Janet	18-Jan-1997	10409	\$219.20
USA	Levenstein	Janet	15-Jan-1997	10410	\$802.00
UK	Dachsbawt	Asra	21-Jan-1997	10411	\$966.00

**Slika 1.6**

Pošto pokrenete upit Employee Sales by Country, pojaviće se tabelarni prikaz podataka koji obuhvata sva polja zadata upitom i sve slogove koji zadovoljavaju kriterijum upita.

Pošto zapišete upit, samo njegovu definiciju, plan ili svojstva formatiranja, i tabelarni prikaz su zapravo zapisani u bazi podataka. Access Vam za strukturiranje uputa nudi intuitivni alat. Na slici 1.7 vidite prozor Query Design. Da biste otvorili ovaj prozor, u prozoru Database iz spiska Objects odaberite Queries, zatim odaberite upit koji želite da izmenite, pa kliknite Design. Upit koji vidite na slici izdvaja podatke iz tabele Employees, Orders i upita Order Subtotals. Upit prikazuje polja Country, LastName i FirstName tabele Orders, polja ShippedDate i OrderID tabele Orders i polje Subtotal upita Order Subtotals. Rezultat upita sadrži samo slogove koji zadovoljavaju uslov Shipped Date. Ovaj specijalni tip upita nazivamo parametarski upit (parameter query). U toku izvršavanja upit traži kriterijum izvršavanja i pomoću njega određuje koje slogove će prikazati u rezultatu. Upiti se obrađuju u Poglavlju 4, "Šta svaki programer mora da zna o osnovama upita", i Poglavlju 11, "Napredne tehnike postavljanja upita". Pošto su upiti osnova za većinu formulara i izveštaja, ja ću ih objašnjaviti u knjizi onako kako ih budem koristila za druge objekte baze podataka.

**Slika 1.7**

Ovim upitom se prikazuju podaci iz tabele Employees i Orders i upita Order Subtotals.

#### Napomena

U ADP-u, upiti se u prozoru Database prikazuju kao pogledi, uskladištene procedure i funkcije. Access ih ne zapisuje u Accessovoj datoteci projekta. Umesto toga ih zapisuje u SQL Serverovoj bazi podataka sa kojom ADP uspostavlja vezu.

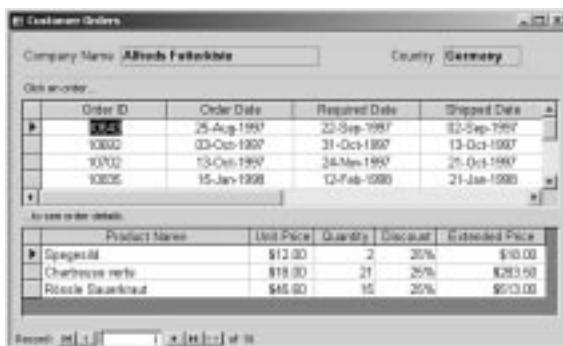
## Formulari: Načini prikazivanja, menjanja i dodavanja podataka

Iako u pogledu Datasheet tabele možete unositi i menjati podatke, ne možete dovoljno dobro kontrolisati akcije korisnika, niti možete učiniti mnogo toga kako biste olakšali postupak unošenja podataka. Zbog toga se koriste formulari. Accessovi formulari imaju mnogo osobina i veoma su moćni i fleksibilni.

Da biste prikazali bilo koji formular, iz spiska Objects odaberite Forms. Zatim dva puta kliknite formular koji želite da prikažete, ili obeležite formular koji želite da prikažete tako što ćete ga kliknuti pa zatim kliknute Open. Na slici 1.8 vidite formular u pogledu Form. Ovaj formular su zapravo tri formulara u jednom: jedan glavni formular i dva podformulara. U glavnem formularu se prikazuju informacije iz tabele Customers, a u podformularima se prikazuju informacije iz tabele Orders i Order Details. Kada korisnik prelazi sa kupca na kupca, u formularu se prikazuju narudžbine koje je kupac napravio. Kada korisnik odabedre narudžbinu tako što je klikne, u formularu se prikazuju proizvodi koji su deo narudžbine.

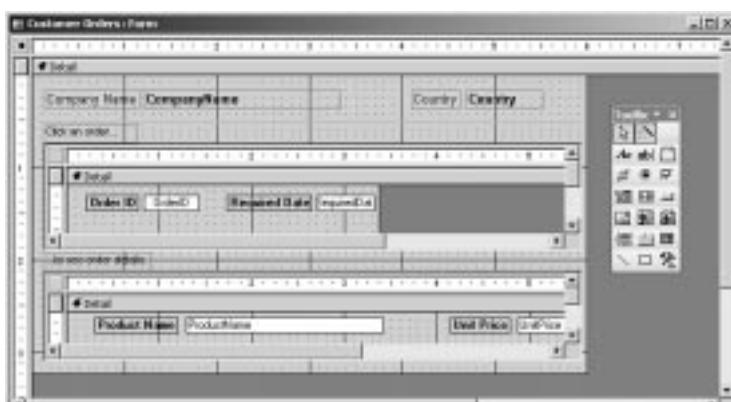
Kao što to možete da uradite za tabele i upite, i formulare možete prikazati u pogledu Design. Da biste videli strukturu formulara, iz spiska Objects odaberite ikonu Forms (videti sliku 1.1), odaberite formular čiju strukturu želite da promenite, pa zatim kliknite Design.

Na slici 1.9 vidite formular Customer Orders u pogledu Design. Sigurno ste primetili dva podformulara unutar glavnog formulara. Formulari se obrađuju u Poglavlju 5, "Šta svaki programer mora da zna o formularima", i Poglavlju 9, "Napredne tehnike rada sa formularima". Formulare će u knjizi, pored toga, obradivati i kada se budu odnosili na druge primere pravljenja aplikacije.



**Slika 1.8**

U formularu Customer Orders se prikazuju informacije o kupcu, narudžbini i njenim detaljima.



**Slika 1.9**

Iz strukture formulara Customer Orders se vidi da sadrži dva podformulara.

## Izveštaji: Pretvaranje podataka u informacije

Formulari Vam omogućavaju da unosite i menjate informacije, ali pomoću izveštaja možete prikazati informacije, obično štampane. Na slici 1.10 vidite izveštaj u režimu pregleda. Da biste pregledali bilo koji izveštaj, iz spiska Objects odaberite Reports. Dva puta kliknite izveštaj koji želite da pregledate ili odaberite izveštaj pa zatim kliknite Preview. Primetili ste grafiku u izveštaju i druge detalje, kao što je osenčena linija. Kao i formulari, izveštaji mogu biti detaljni, a da ipak sadrže informacije od koristi.

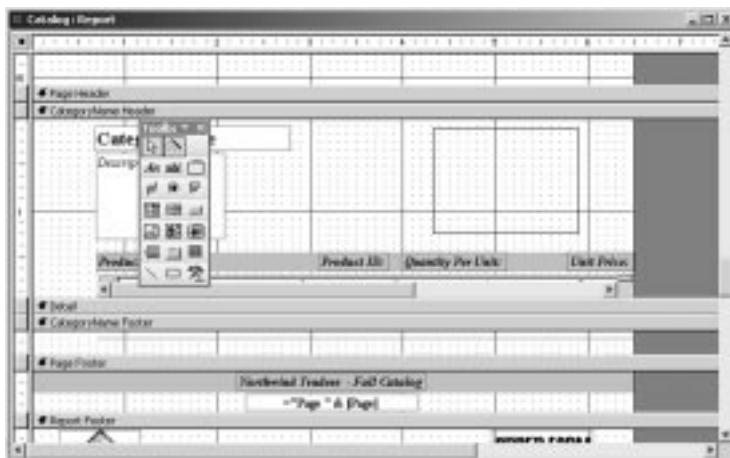
**Slika 1.10**

Ovaj pregled izveštaja Catalog prikazuje informacije u izveštaju.

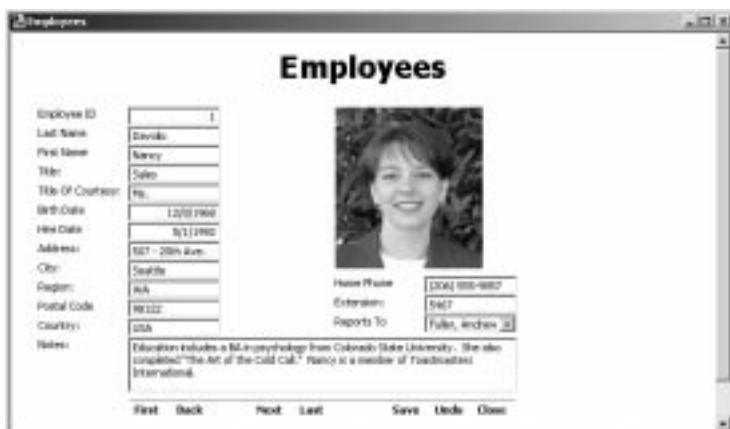
Ukoliko još uvek niste shvatili, izveštaje možete prikazati u pogledu Design, kao što vidite na slici 1.11. Da biste prikazali strukturu bilo kojeg izveštaja, iz spisk Objects odaberite Reports, pa zatim, pošto odaberete izveštaj, kliknite Design. Na slici 1.11 vidite izveštaj koji ima veći broj oblasti (odeljaka); na slici vidite Page Header, CategoryName Header, Detail Section, CategoryName Footer i Report Footer - to su samo neki od mnogih odeljaka izveštaja. Kao što formulari mogu sadržati podformulare, tako i izveštaji mogu sadržati podizveštaje. Izveštaji se obrađuju u Poglavlju 6, "Šta svaki programer mora da zna o izveštajima", i Poglavlju 10, "Napredne tehnike rada sa izveštajima". Izveštaje će takođe obradivati kako ih budem koristila u primerima.

## Stranice za rad sa podacima: Formulari koji se prikazuju u pretraživaču

Stranice za rad sa podacima, koje smo razmatrali ranije u ovom poglavlju, prvi put su se pojavile u Accessu 2000. One Vam omogućavaju da podatke baze podataka prikažete i izmenite pomoću pretraživača. Iako ih Access zapisuje izvan Accessove datoteke baze podataka (.mdb), Vi ih pravite i održavate na način koji je sličan radu sa formularima. Na slici 1.12 vidite stranicu za rad sa podacima koja je prikazana u Accessu. Iako su stranice za rad sa podacima namenjene korišćenju u pretraživaču, možete ih pregledati u okruženju Accessove aplikacije.

**Slika 1.11**

Pogled Design izveštaja Catalog Vam omogućava da vidite strukturu izveštaja.

**Slika 1.12**

Ovo je primer stranice za rad sa podacima koja je bazirana na tabli Employees.

Takođe, u pogledu Design možete prikazati i menjati stranice za rad sa podacima. Na slici 1.13 vidite stranicu za rad sa podacima u pogledu Design. Kao što vidite, pogled Design stranice za rad sa podacima je sličan istom pogledu za formulare. To rad sa stranicama za rad sa podacima i prosleđivanje aplikacije preko lokalne mreže čini lakim.

### **Savet**

U Accessu 10 se prvi put pojavila mogućnost zapisivanja Accessovih formulara kao stranica za rad sa podacima. Ta nova osobina je olakšavala pravljenje formulara koji su koristili korisnici Accessa i korisnici pretraživača.

**Slika 1.13**

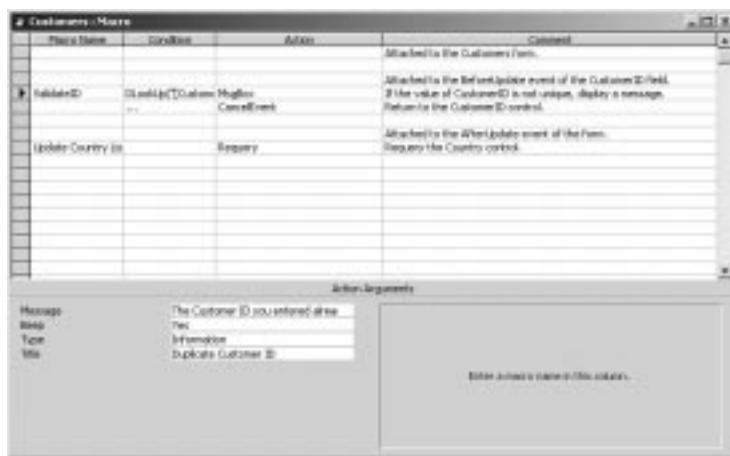
Ova stranica za rad sa podacima se prikazuje u pogledu Design.

## Makroi: Način za automatizovanje sistema

Makroi u Accessu nisu kao makroi u drugim članovima paketa Office. Ne možete ih snimiti, kao što to možete uraditi u Accessu ili Excelu, a Access ih ne zapisuje kao kod u programskom jeziku Visual Basic for Applications (VBA). Pomoću Accessovih makroa možete obaviti većinu poslova koje možete obaviti pomoću tastature, menija i paleta alata. Makroi Vam omogućavaju da u aplikaciju ugradite logiku. Uopšte, radije koristite VBA kod koji se nalazi u modulima, a ne u makroima, za obavljanje poslova koje aplikacija mora da obavi. To je zato što su moduli VBA koda fleksibilniji i moćniji od makroa. Iako su se u ranijim verzijama Accessa neki poslovi mogli obaviti samo pomoću makroa, programeri ih danas veoma malo koriste. Stoga se u ovoj knjizi neću baviti pravljenjem aplikacija pomoću makroa.

Da biste pokrenuli makro, iz spiska Objects odaberite Macros, kliknite makro koji želite da pokrenete, pa zatim kliknite Run. Access će zatim izvršiti akcije koje su navedene u makrou. Da biste videli strukturu makroa, iz spiska Objects odaberite Macro, obeležite makro koji želite da izmenite, a zatim kliknite design kako biste ga otvorili u prozoru Macro Design (pogledajte sliku 1.14). Makro koji vidite na slici ima četiri kolone. Prva kolona je kolona Macro Name (ime makroa) u kojoj možete zadati ime podrutine unutar makroa. U drugoj koloni možete zadati uslov. Akcija koja se nalazi u trećoj koloni makroa se neće izvršiti ukoliko nije zadovoljen uslov koji je zadat za tu akciju.

U četvrtoj koloni možete dokumentovati makro. U donjoj polovini prozora Macro Design zadajete argumente koji se odnose na obeleženu akciju. Na slici 1.14 obeležena je akcija MsgBox, koja prihvata četiri argumenta: Message, Beep, Type i Title.



**Slika 1.14**

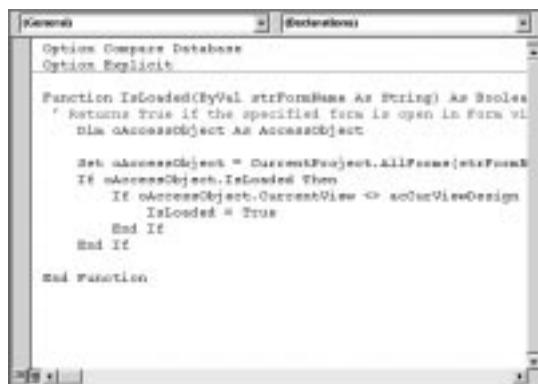
Struktura makroa Customers sadrži ime makroa, uslove, akcije i komentare.

## Moduli: Osnova postupka pravljenja aplikacije

Moduli, osnova svake aplikacije, Vam omogućavaju da napravite biblioteke funkcija koje možete koristiti u aplikacija. U modulima koje pravite obično smeštate podrutine i funkcije. Rezultat rada funkcije je uvek vrednost; podrutine kao rezultat rada ne daju vrednost. Korišćenjem modula koda možete uraditi sledeće:

- Obraditi greške
- Deklarisati i koristiti promenljive
- Raditi sa skupovima slogova
- Pozivati funkcije Windowsovog API-ja i drugih biblioteka
- Praviti i menjati sistemske objekte, kao što su tabele i upiti
- Obavljati obradu pomoću transakcija
- Obavljati mnoge funkcije koje ne možete obavljati pomoću makroa
- Testirati i debagovati složene procese
- Praviti biblioteku baza podataka

To su samo neki od poslova koje možete obavljati pomoću modula. Da biste videli strukturu postojećeg modula, u spisku Objects kliknite Modules, odaberite modul koji želite da izmenite, pa kliknite Module kako biste otvorili okvir za dijalog Module Design (videti sliku 1.15). Modul sa globalnim kodom koji vidite na slici 1.15 sadrži odeljak General Declarations i funkciju IsLoaded. Module i programski jezik VBA će obraditi u Poglavlju 7, "Programski jezik VBA: Uvod", i Poglavlju 12, "Napredne tehnike rada sa programskim jezikom VBA". Module i programski jezik VBA će detaljno obrađivati tokom cele knjige.



**Slika 1.15**

Modul sa globalnim kodom u pogledu Design sadrži odeljak General Declarations i funkciju IsLoaded.

## Pravila imenovanja objekata

Uspostavljanje skupa pravila imenovanja - i njihovo poštovanje - jedan je od ključeva za uspešno programiranje u Accessu ili bilo kom drugom programskom jeziku. Kada birate skup pravila imenovanja, obratite pažnju na sledeće tri osobine:

- Lakoću korišćenja
- Čitljivost
- Koliko su rasprostranjena među programerima

Redikova pravila imenovanja, koje je predložio Greg Redik (Greg Reddick), svakako su najbolji skup pravila imenovanja koji postoji u svetu programiranja.

Redikova pravila imenovanja daju standardizovani način za imenovanje objekata. Ova pravila su izvedena iz Lesinski-Redikovih pravila imenovanja koja su predložena u Accessovim verzijama 1.x i 2.0. Ti standardi su usvojeni i korišćeni u svetu programiranja i mogu se pronaći u većini dobrih knjiga i časopisa napisanih u nekoliko proteklih godina.

Redik je revidirao pravila imenovanja kako bi ih prilagodio potrebama ljudi koji programiraju pomoću Accessa, Visual Basica, Excela i drugih Microsoftovih proizvoda koji koriste programski jezik VBA. Ova pravila su dosledna metodologija koju je lako koristiti za imenovanje objekata u svim tim okruženjima.

U Dodatku B, "Pravila imenovanja", nalazi se sažeta i donekle izmenja verzija Redikovih pravila za imenovanje objekata. Ta pravila će koristiti u knjizi i istaći će neke njihove aspekte kada ih budem koristila.

## Hardverski zahtevi

Jedan od nedostataka Accessa je količina hardverskih resursa koji su mu neophodni. Zahtevi programera se razlikuju od zahteva krajnjih korisnika pa sam sistemske zahteve podelila u dva dela. Dok budete čitali zahteve, zabeležite šta ne zadovoljava preporučene minimalne zahteve.

### Kakav hardver treba da ima računar programera?

Prema dokumentaciji Microsofta, ovo su zvanični minimalni zahtevi za korišćenje Microsoft Accessa 11:

- Procesor Pentium III 133MHz ili bolji
- Windows 2000, Windows XP ili Windows 2003 operativni sistem
- 64MB RAM memorije (preporučuje se 128MB); 8MB za svaku aplikaciju koja se izvršava uporedo sa MS Accessom
- 245MB praznog prostora na disku; 115MB na disku na kojem je instaliran operativni sistem
- Super VGA ili bolja rezolucija
- CD-ROM uredaj
- Uredaj za rad sa pokazivačem

Kao da sav taj hardver nije dovoljan, moji zahtevi za računar na kojem se programira su mnogo veći, jer ćete pored Microsoft Accessa verovatno izvršavati druge aplikacije. Takođe, verovatno želite da izbegnete mogućnost usporavanja računara ili drugih problema do kojih može doći kada ima malo slobodne memorije. Ja Vam preporučujem da računar na kojem programirate (pored zahteva koje je postavio Microsoft) ima i:

- Procesor P5 90MHz ili bolji
- 512MB RAM memorije, ukoliko je moguće
- Monitor visoke rezolucije - što veći to bolje

**Upozorenje**

Ukoliko prilikom programiranja koristite monitor visoke rezolucije, formulare treba da strukturirate tako da se mogu prikazati na monitorima koji zadovoljavaju najmanji zajednički činilac. Iako prilikom programiranja možete iskoristiti prednosti velikih rezolucija, nemojte zaboraviti da će mnogi korisnici koristiti Vašu aplikaciju u nižoj rezoluciji.

Sve u svemu, što bolji hardver, to je bolje. Nikada nije dovoljno memorije ili praznog prostora na disku. Što je više memorije, to ćete lakše i udobnije koristiti Access.

## Kakav hardver treba da ima računar korisnika?

Iako računar korisnika ne mora da bude toliko napredan kao računar programera, ja ipak preporučujem da pored Microsoftovih zahteva zadovoljava i sledeće:

- Procesor Pentium III ili bolji
- 128MB RAM memorije, i 256MB ukoliko Vaša aplikacija podržava OLE, ili ukoliko će korisnik pored aplikacije koristiti i druge programe.

## Kako da počnem da pravim Accessovu aplikaciju?

Mnogi programeri veruju da, s obzirom na to da je Access okruženje za rapidno pravljenje aplikacija, ne postoji potreba za sistemskom analizom prilikom pravljenja aplikacije. Ne mogu opisati koliko se ne slažem sa takvim verovanjem. Kao što je rečeno ranije u ovom poglavljju, Accessove aplikacije se varljivo lako prave, ali bez pravilnog planiranja mogu postati katastrofalne.

### Analiza zadatka

Prvi korak u postupku pravljenja aplikacije jeste analiza zadataka, odnosno uzimanje u obzir svakog pojedinačnog procesa koji se odvija u svakodnevnom radu korisnika - dosadan je, ali neophodan posao. Kada sam prvi put počela da radim za veliku korporaciju kao programer glavnog računara, od mene se zahtevalo da dosledno sledim listu analize zadataka. Morala sam da saznam šta svaki korisnik sistema radi kako bi obavio svakodnevne poslove, dokumentujem svaku proceduru, utvrđim kako se sa jednog prelazi na drugi zadatak, uspostavim vezu svakog zadataka svakog korisnika sa njegovim ostalim zadacima kao i sa zadacima ostalih korisnika sistema, i da svaki zadatak povežem sa ciljevima korporacije. U savremenom svetu brzog pravljenja aplikacija i menjanja tehnologija, analiza zadataka u postupku pravljenja aplikacija nije više u prvom planu. Ja i dalje verujem da, ukoliko ovaj postupak pažljivo ne sprovedete bar na nekom nivou, moraćete da prepravite velike delove aplikacije.

## Analiza i struktura podataka

Pošto ste analizirali i dokumentovali sve zadatke koji postoje u sistemu, spremni ste da se posvetite fazi analize i strukturiranja podataka aplikacije. U ovoj fazi morate identifikovati svaki deo informacije koji je neophodan za obavljanje svakog zadatka. Te elemente podataka morate pridružiti subjektima, a svaki subjekat će postati zasebna tabela u bazi podataka. Na primer, subjekat može biti klijent; svaki podatak koji se odnosi na tog klijenta - ime, adresa, telefonski broj, limit kredita i bilo koja druga značajna informacija -postaće polja u tabeli klijenata.

Za svaki podatak morate da utvrdite sledeće:

- Odgovarajući tip podataka
- Neophodnu veličinu
- Pravila provere tačnosti podataka

Takođe, treba da utvrdite da li ćete korisniku omogućiti da ažurira svaki podatak i da li se podatak unosi ili izračunava. Zatim proveravate da li ste pravilno normalizovali strukture tabele.

### Normalizacija na lak način

Normalizacija je elegantan termin za postupak testiranja strukture tabele nizom pravila kojima se obezbeđuje da će aplikacija raditi što je efikasnije moguće. Ova pravila su izvedena na osnovu teorije skupova i prvobitno ih je postavio Dr E. F. Kod (E. F. Codd). Mada možete provesti godine proučavajući normalizaciju, njen glavni cilj jeste aplikacija koja se efikasno izvršava sa što je manje moguće rada sa podacima i kodiranja. U Poglavlju 4 se detaljno objašnjava normalizacija i strukturiranje baze podataka. Za sada evo šest osnovnih pravila normalizacije:

1. Polja treba da budu atomična - to jest, svaki podatak treba da se rastavi što je više moguće. Na primer, umesto da pravite polje Name (ime), napravili biste dva polja: jedno za lično ime, a drugo za prezime. Na ovaj način se mnogo lakše radi sa podacima. Ukoliko je potrebno da sortirate ili pretražujete po ličnom imenu, ne uzimajući u obzir prezime, na primer, možete to uraditi bez dodatnog truda.
2. Svaki slog treba da sadrži jedinstveni identifikator tako da imate bezbedan način za identifikovanje sloga. Na primer, ukoliko menjate informacije o kupcu, možete biti sigurni da menjate informacije koje se odnose na odgovarajućeg korisnika. Ovaj jedinstveni identifikator nazivamo primarni ključ (primary key).
3. *Primarni ključ* je polje, ili polja, pomoću kojeg se jedinstveno identificuje slog. Ponekad se može upotrebiti prirodni primarni ključ. Na primer, broj socijalnog osiguranja u tabeli zaposlenih treba jedinstveno da identificuje zaposlenog. U drugim slučajevima ćete verovatno morati da pravite primarni ključ. Pošto dva kupca mogu da imaju isto ime, na primer, ime kupca se ne može koristiti za jedinstveno identifikovanje. Može biti neophodno napraviti polje u kojem će se nalaziti jedinstveni identifikator za kupca, recimo polje CustomerID.

4. Primarni ključ treba da je kratak, stabilan i jednostavan. Kada se kaže kratak, misli se da bude male veličine (a ne polje u koje se može smestiti 50 znakova). Polje tipa Long Integer je savršeno za primarni ključ. Kada se kaže stabilan, misli se da primarni ključ treba da bude polje čija se vrednost retko, verovatno nikad, menja. Na primer, mada je malo verovatno da će se menjati identifikator klijenta (kupca, kompanije), ime kompanije će se verovatno menjati. Kada se kaže jednostavan, to znači da korisnik može lako da ga koristi.
5. Svako polje tabele mora sadržati dodatne informacije o slogu koji primarni ključ jedinstveno identificuje. Na primer, svako polje u tabeli klijenata opisuje klijenta sa određenim identifikatorom.
6. Informacije u tabeli treba da se pojavljuju samo jednom. Na primer, ime klijenta sme da se pojavi samo u jednom slogu.

Pozabavimo se primerom. Tabela koju vidite na slici 1.16 je primer tabele koja nije normalizovana. Primetili ste da se polje CustInfo ponavlja u svakoj narudžbini, pa ako se adresa klijenta promeni, mora se promeniti u svakoj narudžbini koja je pridružena klijentu. Drugim rečima, polje CustInfo nije atomično. Ukoliko želite da obavite sortiranje po gradu, nemate sreće jer se ime grada nalazi u sredini polja CustInfo. Ukoliko se promeni ime elementa inventara, morate promeniti svaki slog u kojem se element nalazi (dakle, u svakoj narudžbini). Verovatno najgori problem u ovom primeru su elementi koji su naručeni. Sa ovakvom strukturom morate napraviti četiri polja za svaki element koji klijent naručuje: polje za ime, dobavljača, količinu i cenu. Ovakva struktura izuzetno otežava pravljenje izveštaja o prodaji i drugih izveštaja koji su korisnicima neophodni za efikasno poslovanje.

OrderID	CustInfo	OrderTotal	OrderDate	ItemName	ItemSupplier	ItemQuantity	ItemPrice	ItemCust
1	123 Any Street Anywhere, CA	\$80.00	11/15/2002	Widget	Good Supplier	5	\$1.50	Widget
2	223 Any Street Anywhere, CA	\$175.00	11/16/2002	Net	Same Supplier	3	\$6.00	Net
3	323 Any Street Anywhere, CA	\$125.00	12/03/2002	Stuff	Another Supplier	4	\$6.00	Telescope
4	423 Any Street Anywhere, CA	\$884.00	12/15/2002	Panel	Same Supplier	5	\$6.00	Ski Poles
5	545 Another Street Somewhere, CA	\$200.00	11/17/2002	Eraser	Another Supplier	9	\$6.00	Scissors
6	645 Another Street Somewhere, CA	\$299.00	10/14/2002	Computer	Good Supplier	2	\$6.00	Bolt
7	[A,B,C,D, E,F,G,H]	\$0.00				0	\$6.00	

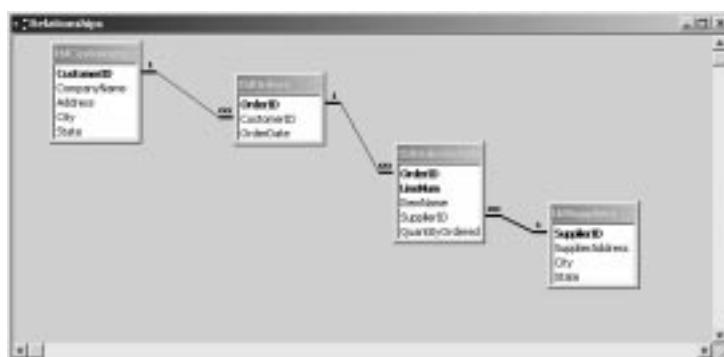
**Slika 1.16**

Ova tabela nije normalizovana.

Na slici 1.17 vidite iste podatke koji su ovog puta normalizovani. Primetili ste da sam ih razdvojila u nekoliko tabela: tabele `tblCustomers`, `tblOrders`, `tblOrderDetails` i `tblSuppliers`. U tabeli `tblCustomers` se nalaze podaci koji se odnose samo na klijente.

Svaki slog tabele sam jedinstveno identifikovala pomoću polja `CustID`, koje koristim za uspostavljanje odnosa sa tabelom narudžbina, `tblOrders`, i tabelom detalja o narudžbinama, `tblOrderDetails`. U tabeli `tblOrders` se nalaze samo informacije koje se odnose na celu narudžbinu, a ne na pojedinačne elemente narudžbine. U ovoj tabeli se nalazi polje `CustID` sa identifikatorom klijenta koji je izvršio narudžbinu i polje sa njenim datom; i ovu tabelu sam povezala sa tabelom `tblOrderDetails` pomoću polja `CustID`. U tabeli `tblOrderDetails` se nalaze informacije o svakom elementu narudžbine za zadati identifikator `OrderID`.

Ne postoji ograničenje broja elemenata koje korisnik može smestiti u nardžbinu. Korisnik može dodati onoliko elemenata koliko je potrebno tako što će u tabelu `tblOrderDetails` dodati nove slogove. Najzad, infromacije o dobavljaču sam smestila u posebnu tabelu, tabelu `tblSuppliers`, tako da, ukoliko se izmeni bilo koja informacija o dobavljaču, korisnik te izmene unosi samo na jednom mestu.



**Slika 1.17**

Podaci su normalizovani u četiri tabele.

## Pravljenje prototipova

Mada se faze analize zadataka i podataka u postupku pravljenja aplikacije nisu značajno promenile od vremena korišćenja centralnih računara, faza pravljenja prototipova se jeste izmenila. U radu sa centralnim računarima ili DOS-ovskim programskim jezicima bilo je veoma važno napraviti detaljne specifikacije za svaku masku i izveštaj. Pamtim da se od korisnika zahtevalo da se odjave iz svake maske i izveštaja. Čak i promena kakva je pomeranje polja u masci je značila promenu redosleda i proveru za veći broj sati. Pošto se korisnik odjavi iz maske ili izveštaja, programeri bi odlazili da danima naporno rade na svakoj masci i izveštaju. Zatim bi se vratili nakon mnogo meseci samo da bi saznali da je sve pogrešno. To je značilo da moraju da se vrati planiranju i da moraju da provedu mnogo radnih sati pre nego što korisnik ponovo može da razmotri aplikaciju.

Danas se ovaj postupak znatno promenio. Čim se skiciraju zadaci i završi se sa analizom podataka, programer može da strukturira tabele i da uspostavi zavisnosti između njih. Zatim može da počne postupak pravljenja prototipova formulara i izveštaja. Umesto da protekne mnogo nedelja i meseci u pravljenju pre nego što se ponovo kontaktira korisnik, programeru je potrebno samo nekoliko dana, koristeći Accessove čarobnjake, da brzo napravi prototipove formulara.

## Testiranje

Bez obzira na trajanje, testiranja nikada nije dosta. Preporučujem Vam da, ukoliko će se Vaša aplikacija izvršavati u operativnim sistemima Windows 98, Windows NT, Windows 2000 i Windows XP, testiranje obavite u svim okruženjima. Takođe Vam preporučujem da aplikaciju temeljno testirate na najmanjem zajedničkom harverskom činiocu - aplikacija se može izvrsno izvršavati na Vašem računaru, ali može dati neprihvatljive rezultate na računarima korisnika.

Obično je od pomoći da aplikaciju testirate u delovima i kao celinu. Zaposlite nekoliko ljudi da testiraju aplikaciju i da to bude raspon od ljudi koji se odlično razumeju u računare do ljudi koji su se najmanje navikli na korišćenje računara. Različiti korisnici će verovatno pronaći potpuno različite skupove problema. Najvažnije od svega je da Vi niste jedini koji testira aplikaciju jer ste Vi najverovatnije osoba koja neće pronaći greške u programima koje pravite.

## Implementiranje

Vaša aplikacija je napokn spremna da ugleda svetlost dana, ili se barem Vi tako nadate! Distribuirajte aplikaciju podskupu korisnika i postrajte se da znaju da je samo testiraju. Učinite ih ponosnim što učestvuju kao prvi korisnici sistema, ali ih upozorite da mogu iskrasnuti problemi i da na njima leži odgovornost da Vas o problemima obaveste. Ukoliko aplikaciju distribuirate velikom broju korisnika i dogodi se da se ona ne izvršava onako kako treba, biće teško ponovo stići poverenje korisnika. Zbog toga je toliko važno polako uvesti aplikaciju u eksplataciju.

## Održavanje

Pošto je Access okruženje za brzo pravljenje aplikacija, period održavanja teži da bude veći od perioda održavanja na centralnim računarima ili DOS-ovskim aplikacijama. Korisnici su mnogo zahtevniji; što im više pružate, to više traže. Za konsultanta je to odlična situacija. Nemojte se naći u situaciji statične ponude - s ozbirom na opseg menjanja aplikacije, možete biti gubitnik.

Postoje kategorije aktivnosti održavanja: ispravljenje grešaka, izmene specifikacija i ukrašavanje. Greške treba da otklonite što je pre moguće. Implikacije menjanja specifikacija moraju se jasno predočiti korisniku, uključujući vreme i troškove za zahtevane izmene. Što se tiče ukrašavanja, pokušajte da što je više moguće uključite korisnike tako što ćete ih naučiti kako da poboljšaju formulare i izveštaje i tako što ćete aplikaciju učiniti fleksibilnijom. Naravno, krajnji cilj svake aplikacije je zadovljiva grupa produktivnih korisnika.

## Šta je sve novo u Accessu 2002 i 2003?

U Accessu 2002 i 2003 postoje brojne nove osobine i svim tim osobina se vredi pozabaviti. Iako je Microsoft većinu novih osobina namenio klijent/server i web integraciji, postoje i mnoga druga korisna poboljšanja. U ovom odeljku se nalazi pregled novih osobina. Svaku od novih osobina ću detaljno objasniti u odgovarajućem poglavlju knjige.

### Nove osobine formulara

Videćete da u Accessu postoji nekoliko novih osobina formulara. Sledi pregled tih novih osobina. U Poglavljima 5 i 9 se detaljno objašnjavaju te nove osobine. Tri nova događaja olakšavaju rad sa formularima. Ti novi događaji su događaji OnUndo na nivou formulara i na nivou kontrola i događaj OnDirty. Događaj OnUndo na nivou formulara se dešava kada korisnik poništi sve izmene koje je načinio u formularu. Jednako kao parnjak na nivou formulara, tako se i događaj OnUndo na nivou kontrole dešava kada korisnik poništi izmene u kontroli. Najzad, kontrole sada imaju događaj OnDirty. To znači da možete reagovati kada je kontrola "prljava", baš kao što možete reagovati kada je formular "prljav".

Pored svih novih događaja formulara, u Accessu 2002 i 2003 postoji novo svojstvo i nov metod. Novo svojstvo je svojstvo Movable. Ovo svojstvo postoji kako za formulare tako i za kontrole. Možete ga koristiti kako biste saznali da li se formular ili kontrola mogu pomerati. Novi metod je metod Move. Pomoću ovog metoda možete programski pomerati formular.

Ukoliko ste radili sa verzijama Accessa koje prethode Accessu 2003, verovatno Vas je nerviralo što Vam čarobnjaci ComboBox i ListBox ne omogućavaju zadavanje poretka sortiranja. To je značilo da je jedna od prvih stvari nakon izvršavanja čarobnjaka koju mnogi ljudi rade promena svojstva RowSource i zadavanje redosleda sortiranja. Čarobnjaci ComboBox i ListBox u Accessu 2003 Vam omogućavaju da zadate redosled sortiranje po najviše četiri polja za podatke u poljima tipa ComboBox i ListBox. Na ovaj način ne morate obavljati dodatni korak koji je bio neophodan u prethodnim verzijama Accessa.

U mnogim slučajevima osobina AutoCorrect nije podesna za podatke koji se nalaze u bazi podataka. U Accessu 2003 možete bolje kontrolisati ponašanje osobine AutoCorrect. Na ovaj način ona postaje robusniji i korisniji alat u radu sa podacima tabele. Na primer, korišćenjem dugmeta AutoCorrect Options, veoma lako možete poništiti ispravke ili uključiti i isključiti opciju AutoCorrect.

Sledeća nova osobina Accessa 2003 se odnosi na korišćenje tema Windowsa XP. Unapred je definisano da Access za formulare koristi izgled koji koristi operativni

sistem. U Accessu 2003 postoji osobina koja Vam omogućava da sprečite da formulari naslede izgled od operativnog sistema.

Microsoft je u Access 2003 ugradio jednu od mojih najomiljenijih novih osobina. Ova nova grupa osobina, koja se opisuje kao pravila izgleda formulara i izveštaja, obezbeđuje proveru grešaka u formularima i izveštajima. Kada se aktiviraju, ove osobine ističu probleme sa nepridruženim oznakama (labelama) i kontrolama, novim nepridruženim labelama, greškama tastaturnih prečica (recimo, kada dve kontrole imaju istu tastaturnu prečicu) i nepravilnim svojstvima kontrole.

Najzad, Microsoft je prilikom dizajniranja značajno unapredio rad sa podformularima u Accessu 2002 i 2003. Najvažnije od svega je to što sada kontrolu podformulara možete kliknuti desnim tasterom miša i odabratи View, Subform in Own Window. Ova nekada neophodna komanda otvara podformular u posebnom prozoru u pogledu Design.

## **Nove osobine izveštaja**

Baš kao što za formulare postoji nekoliko novih osobina, tako i za izveštaje postoje nove osobine. Te osobine će detaljno objasnitи u Poglavlјima 6 i 10.

U Accessu 2002 i 2003 postoji nekoliko novih svojstava. Ta svojstva obuhvataju svojstva Modal, PopUp, BorderStyle, AutoResize, AutoCenter, MinMaxButtons, CloseButton i ControlBox. Sva ova svojstva se ponašaju kao njihovi parnjaci za formulare i služe da izveštajima daju fleksibilnost kakvu imaju formulari.

Microsoft je metodu OpenReport dodao dva nova argumenta. Argument WindowMode omogućava korisniku da kontroliše ponašanje prozora izveštaja (na primer, da bude neprioritetan). Argument i svojstvo OpenArgs Vam omogućavaju da izveštaju lako prosleđujete informacije, kao da je otvoren, i da iz izveštaja možete pročitati te informacije.

Kao što je Microsoft formularima dodao svojstvo Movable i metod Move, tako ih je učinio dostupnim i za izveštaje. Ove osobine Vam omogućavaju da saznate da li se izveštaj može pomerati i da na lak način napišete kod za pomeranje prozora izveštaja.

## **Uzbudljivi svet pogleda PivotTable i PivotChart**

Access 2002 i 2003 omogućava korisniku da prikaže svaku tabelu, upit, formular, ADP tabelu, ADP pogled, ADP uskladištenu proceduru ili ADP funkciju u pogledu PivotTable ili PivotChart.

Pogledi PivotTable i PivotChart omogućavaju korisniku da lako obavi prilično složene analize podataka. To znači da u Accessu možete obaviti mnoga složena analiziranja podataka koja su se nekada mogla obaviti u Microsoft Excelu. Pogledi PivotTable i PivotChart se mogu koristiti u stranicama za rad sa podacima i podformularima, i možete programski reagovati na događaje koje iniciraju.

## **Upoznajte se sa poboljšanjima programabilnosti u Accessu 2002 i 2003**

Microsoft je u Access 2002 i 2003 ugradio nekoliko novih poboljšanja programabilnosti. Ta poboljšanja obuhvataju sledeće:

- Sada sa Accessove komandne linije možete proslediti lozinku baze podataka.
- Svojstvima DateCreated i DateModifies se sada programski može pristupiti za sve Accessove objekte.
- Pomoći program Expression Builder se može koristiti u okruženju Visual Basic Environment (VBE).
- Novi objekat Printer i kolekcija Printers Vam omogućavaju da lako programski komunicirate sa štampačima.
- Metodi AddItem i RemoveItem najzad postoje za polja tipa ListBox i ComboBox.
- Uvedeni su metodi CompactRepair i ConvertAccessProject.
- Svojstvo BrokenReference Vam omogućava da lako saznate da li je bilo koja od referenci narušena.
- Zapisivanje i kompajliranje koda je "strujoliko", čime se poboljšavaju performanse prilikom pravljenja velikih aplikacija.

Kao što vidite, nekoliko ovih osobina dokazuju šta sve programer može da uradi u vreme izvršavanja. Ostala osobine poboljšavaju ili unapređuju performanse razvojnog okruženja. Sve ove osobine Vam pomažu da posao obavite efikasnije i efektnije. Svaku od ovih osobina ću detaljnije objasniti u odgovarajućim poglavljima knjige.

## **Access 2002 i 2003 i XML**

Microsoft je u Access 2002 i 2003 ugradio značajnu podršku za XML. Koristeći Microsoft Access, u XML možete lako izvoziti podatke i šeme i lako ih možete uvoziti iz XML-a, bilo programski, bilo pomoću korisničkog interfejsa. Access podatke i šeme izvozi kao snimke stanja stalnih tabela ili upita. Šeme izvozi u standardu W3C XSD. Access predstavljanje, odnosno format izlaza, zasniva na XSL dokumentu koji generiše HTML 4.0. XSL prezentovanje dokumenta i XML dokument podataka su povezani pomoću .htm datoteke koja se izvršava na računaru korisnika.

Ono što je novo u Accessu 2003 jeste mogućnost zadavanja datoteke transformacije prilikom uvoženja XML podataka ili izvoženja Accessovih podataka u XML. XML podatke možete transformisati prilikom njihovog uvoženja. Transformisanje se obavlja pre nego što Access napravi novu tabelu ili uveze podatke u novu tabelu. Kada podatke izvozite u XML, možete opcionalno izvoziti XSL datoteku. Access generisani XSL datoteku zasniva na informacijama o prezentovanju koje pronađe u Accessovom formularu, izveštaju ili tabeli koju izvozite.

U ovoj knjizi se ukratko objašnjava XML podrška u Accessu 2002 i 2003. Detaljan opis korišćenja XML-a možete naći u knjizi "Ovladavanje klijent/server programiranjem u Accessu 11, Alison Balter" (*Alisons Balter's Mastering Access 11 Client/Server Development*).

## Nove osobine ADP projekata

Mehanizam SQL Server 2000 Desktop Engine je zamena za mehanizam koji je ranije bilo poznat kao Microsoft Database Engine (MSDE). Ovaj robusni mehanizam ima iste osobine koje je Microsoft prvi put ugradio u SQL Server 2000. Nekoliko novih osobi na postoji u mehanizmu SQL Server 2000 Desktop Engine:

- Mogućnost pravljenje korisničkih funkcija
- Proširena podrška svojstava (na primer, lookup zavisnosti, teksta potvrđivanja, podtabela i tako dalje)
- Pogleda koji se mogu ažurirati
- Podrška za kopiranje baze podataka i transfer baze podataka

## Nove osobine stranica za rad sa podacima

Stranice za rad sa podacima u Accessu 2000 su imale ograničene osobine i sa njima je bilo teško raditi. Microsoft ih je značajno poboljšao u Accessu 2002 i 2003. Na primer, dizajner Data Access Page Designer u Accessu 2002 i 2003 podržava više novih poništavanja i ponovne primene. Ovaj dizajner Vam takođe omogućava da odaberete i radite sa više objekata pomoću miša ili tastature. Microsoft je ugradio mnoge opcije za promenu veličine i prevlačenje u novi i poboljšani dizajner. U dizajneru možete koristiti desni taster miša, a prozor Properties sada sadrži svojstva koja se mogu koristiti za odabrani objekat ili objekte. Kratko rečeno, dizajneri Accessa 2002 i 2003 su veoma slični dizajnerima formulara i nisu više kao zapostavljena deca.

U Accessu 2000 "trakaste" stranice za rad sa podacima imaju veoma ograničene mogućnosti. U Accessu 2002 i 2003, pomoću ovih stranica se mogu obavljati ažuriranja. Na nivou traka sada postoje mnoga nova svojstva. Među njima su svojstva AllowAdditions, AllowDeletions i Allow Edits. Nova osobina Autosum olakšava postupak pravljenja agregatnih izračunavanja.

Pored standardnih struktura stranica za rad sa podacima, u Accessu 2002 i 2003 postoje opcije Tabular, PivotChart i Spreadsheet. Pomoću ovih opcija možete lako svojim potrebama prilagoditi izgled stranica za rad sa podacima.

Najzad, stranice za rad sa podacima Accessa 2002 i 2003 je mnogo lakše praviti nego u Accessu 2000. Skript na nivou stranice obaveštava korisnike koji ne poseduju odgovarajući pretraživač. Relativne putanje (a ne absolutne putanje) se mogu zadati Accessovoj bazi podataka, a Office Data Connection (ODC) i Universal Data Link (UDL) Vam omogućavaju da centralizujete informacije o uspostavljanju veze sa bazom podataka. Pored toga, pošto Microsoft omogućava programabilan rad sa svojstvom Link, lako u vreme izvršavanja možete programski raditi sa stranicom.

Stranice za rad sa podacima u Accessu 2002 i 2003 imaju funkcionalnost koju biste očekivali od "dizajnera formulara". Veoma je lako sa njima raditi i oni sadrže veliki broj osobina. Stranice za rad sa podacima se detaljno objašnjavaju u knjizi "Ovladavanje klijent/server programiranjem u Accessu 11, Alison Balter" (Alisons Balter's Mastering Access 11 Client/Server Development).

## Ostale nove osobine Accessa 2002 i 2003

Pored svega do sada pomenutog, ima još novih osobina. Ostale nove osobine Accessa 2002 i 2003 obuhvataju povećanu robusnost, dnevničenje grešaka konvertovanja, rasprostranjenost, pristupačnost i prepoznavanje govora.

Dva osnovna dela funkcionalnosti poboljšavaju robusnost Accessa 2002 i 2003. Prvi je značajno poboljšan pomoći program za sažimanje i oporavljanje baze podataka, a drugi je poboljšani mehanizam pretraživanja i postupak pronaalaženja grešaka za narušene reference. Kao pomoć pri konvertovanju Accessove 2000 aplikacije u Access 2002 i 2003, Access 2002 i 2003 obezbeđuju tabelu sa informacijama o problemima na koje možete naći tokom postupka konvertovanja.

Jedna od omiljenih novih osobina Accessa 2003 jeste mogućnost prikazivanja od čega sve objekat zavisi. Drugim rečima, pre uklanjanja upita, možete saznati koji formuli, izveštaji i upiti zavise od upita koji želite da uklonite.

Sledeća nova osobina Accessa 2003 jeste mogućnost uvoženja, izvoženja i povezivanja sa listama SharePoint Team Services. Takođe, iz SharePoint Team Servicesa možete uvoziti, izvoziti i povezivati Accessove podatke. Ove osobine u mnogome doprinose poboljšavanju integracije između Accessa i SharePoint Team Servicesa.

Microsoft je u Access 2003 ugradio novu sigurnosnu mrežu. Dok radite u Accessu, možete napraviti rezervnu kopiju baze podataka ili projekta pre nego što ga izmenite. Rezervnu kopiju možete zapisati na lokaciju koja je unapred za to predviđena ili je možete zapisati u tekući direktorijum. Restauriranje baze podataka zahteva samo njeno preimenovanje i otvaranje u Accessu.

Sledeća odavno zakasnela osobina koju je Microsoft ugradio u Access 2003 jeste mogućnost pravljenja lokalne kopije strukture i/ili podataka povezane tabele. Na taj način je rad sa povezanim tabelama "glatkiji".

Osnovna, ali dobrodošla osobina Accessa 2003 jeste mogućnost promene oblika i veličine slova SQL-a i pogleda za pravljanje upita za Accessove upite i upite SQL Servera sa kojima možete raditi pomoću Accessovog projekta.

Na ovaj način možete mnogo komotnije raditi sa upitima, pogledima, uskladištenim procedurama i funkcijama koje su neophodne aplikaciji.

U Accessu 2002 i 2003 postoje mnoge osobine koje dokazuju njihovu konkurentnost na međunarodnom tržištu. Te osobine obuhvataju

- Prikazivanje višejezičnog teksta u tabelama, formularima i izveštajima.
- Biranje opcija za proveru pravopisa koje odgovaraju govornom jeziku.
- Promenu pravca čitanja.

Što se tiče pristupačnosti, u Accessu 2002 i 2003 postoji nekoliko novih tastaturnih prečica, kao i mogućnost zumiranja. Najzad, nove opcije za prepoznavanje govora omogućavaju korisniku da diktira tekst i koristi menije zadavanjem glasovnih komandi!

## Praktični primeri: Struktura aplikacije za firmu za računarski konsalting

Zamislite firmu za računarski konsalting koja želi da ima evidenciju o vremenu i naplaćivanju pomoću Accessove aplikacije. Prvo, razmotrimo aplikaciju iz prespektive strukture.



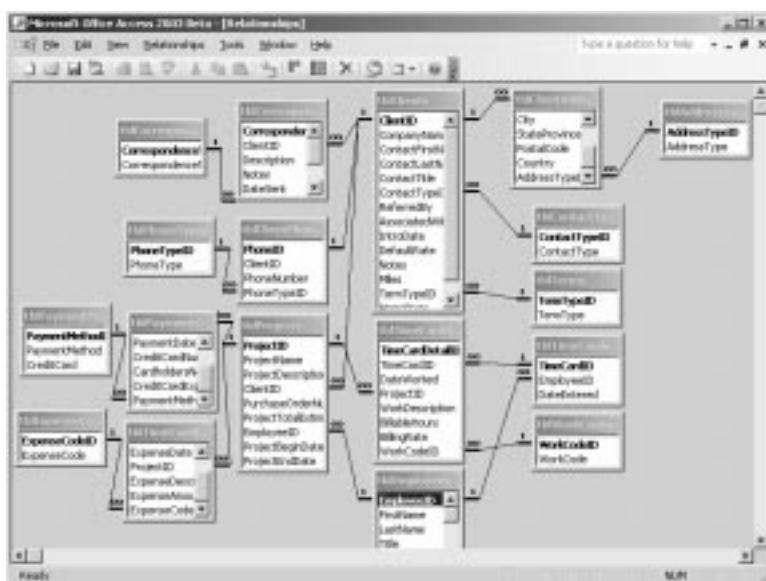
Sistem će voditi evidenciju o klijentima i projektima tih klijenata. Korisnicima će omogućiti da zapisuju sve sate koji se plaćaju, i troškove koji se javljaju, za svakog klijenta i projekat. Takođe, omogućice postojanja stalnih informacija o svakom zaposlenom ili podugovaraču. Tabele sistema sam zasnovala na tabelama koje je napravio čarobnjak Database Wizard. Te tabele sam malo izmenila i preimenovala kako bi odgovarale Redikovim pravilima imenovanja. U sistemu će postojati dvadeset i jedna tabela. Neke od tih tabele ćete napraviti u Poglavlju 2. Sve te tabele se nalaze bazama podataka aplikacije na propratnom CD-ROM-u:

- **tblClients** U ovoj tabeli se nalaze informacije o svakom klijentu; povezana je sa tabelom **tblProjects**, tabelom u kojoj se beleže informacije o projektima svakog klijenta.
- **tblClientAddresses** U ovoj tabeli se nalaze adrese klijenata; povezana je sa tabelama **tblClients** i **tblAddressTypes**.
- **tblAddressTypes** Ova tabela se koristi za pretraživanje. Sadrži sve važeće tipove adresa klijenta; povezana je sa tabelom **tblClientAddresses**.
- **tblClientPhones** U ovoj tabeli se nalaze telefonski brojevi klijenata; povezana je sa tabelama **tblClients** i **tblPhoneTypes**.

- **tblPhoneTypes** Ova tabela se koristi za pretraživanje. Sadrži važeće telefonske brojeve klijenta; povezana je sa tabelom **tblClients**.
- **tblTerms** Ova tabela se koristi za pretraživanje. Sadrži važeće termine plaćanja klijenta; povezana je sa tabelom **tblClients**.
- **tblContactType** Ova tabela se koristi za pretraživanje. Sadrži važeće tipove kontakta klijenta; povezana je sa tabelom **tblClients**.
- **tblProjects** U ovoj tabeli se nalaze sve stalne informacije o projektima; povezana je sa nekoliko tabela: **tblClients**, **tblPayments**, **tblEmployees**, **tblTimeCardHours** i **tblTimeCardExpenses**.
- **tblTimeCardHours** U ovoj tabeli se vodi evidencija o radnim satima provedenim na svakom projektu i za svakog zaposlenog; povezana je sa tabelama **tblProjects**, **tblTimeCards** i **tblWorkCodes**.
- **tblPayments** U ovoj tabeli se vodi evidencija o svim plaćanjima za neki projekat; povezana je sa tabelama **tblProjects** i **tblPaymentMethods**.
- **tblTimeCardExpenses** U ovoj tabeli se vodi evidencija o troškovima za svaki projekat i svakog zaposlenog; povezana je sa tabelama **tblProjects**, **tblTimeCards** i **tblExpenseCodes**.
- **tblEmployees** U ovoj tabeli se beleže informacije o zaposlenima; povezana je sa tabelama **tblTimeCards** i **tblProjects**.
- **tblTimeCards** U ovoj tabeli se vodi evidencija o radnim satima zaposlenih; predstavlja most između više-prema-više zavisnosti između tabela **tblEmployees** i **tblTimeCardExpenses**, kao i između tabela **tblEmployees** i **tblTimeCardHours**. Takođe je povezana sa tabelama **tblEmployees**, **tblTimeCardHours** i **tblTimeCardExpenses**.
- **tblExpenseCodes** Ova tabela se koristi za pretraživanje važećih kodova troškova; povezana je sa tabelom **tblTimeCardExpenses**.
- **tblWorkCodes** Ova tabela se koristi za pretraživanje važećih kodova poslova; povezana je sa tabelom **tblTimeCardHours**.
- **tblPaymentMethods** Ova tabela se koristi za pretraživanje važećih metoda plaćanja; povezana je sa tabelom **tblPayments**.
- **tblCorrespondence** Ova tabela se koristi za evidenciju poslovne prepiske koja se odnosi na projekat; povezana je sa tabelama **tblProjects** i **tblCorrespondenceTypes**.
- **tblCorrespondenceTypes** Ova tabela se koristi za pretraživanje važećih tipova poslovne prepiske; povezana je sa tabelom **tblCorrespondence**.
- **tblCompanyInfo** Ovo je sistemska tabela. Koristi se za zapisivanje informacija o kompaniji. Informacije iz ove tabele možete naći na formularima i izveštajima.

- **tblErrorLog** Ovo je sistemska tabela. U njoj beležimo greške koje dogode za vreme izvršavanja aplikacije.
- **tblErrors** Ovo je sistemska tabela. U njoj beležimo kodove grešaka i njihov opis.

Zavisnosti između tabela su detaljno objašnjene u Poglavlju 3, ali ih takođe vidite na slici 1.18.



**Slika 1.18**

Na slici vidite zavisnosti između tabela koje čine sistem za evidenciju radnih sati i plaćanja.

## Zaključak

Pre nego što savladate praktične aspekte programiranja u Accessu, potrebno je da razumete šta je Access i kako se uklapa u svet pravljenja aplikacija. Access je veoma moćan proizvod koji se koristi na razne načine; Accessove aplikacije možete videti svuda, od kućnih računara do korisnika u korporacijama, pa sve do klijent/server baza podataka.

Pošto shvatite šta je Access i šta može da obavi, spremni ste da ovладate njegovim brojnim objektima. Accessove aplikacije se sastoje od tabela, upita, formulara, izveštaja, stranica za rad sa podacima, makroa, modula, komandnih linija, ActiveX kontrola, zavisnosti i drugih objekata. Kada se pravilno strukturira, Accessova aplikacija efikasno kombinuje te objekte kako bi korisnik imao moćnu, robusnu i korisnu aplikaciju.