

Vizuelni alati baza podataka

DO SADA STE UČILI O NORMALIZACIJI, OSNOVNIM OBJEKTIMA BAZA PODATAKA I PRINCIPIMA BAZA PODATAKA. PRE NEGO ŠTO POČNETE DA GRADITE APLIKACIJE BAZA PODATAKA KORIŠĆENJEM VB.NET-A ZA PRISTUP BAZI, TREBA DA POGLEDATE VIZUELNE ALATE BAZA PODATAKA. VISUAL STUDIO .NET (VS.NET) PRAVI OGROMAN KORAK KA UNIFICIRANJU INTEGRISANOG RAZVOJNOG OKRUŽENJA (IDE) SVIH JEZIKA. U OVOM POGLAVLJU ĆETE UČITI O DELJENOM OKRUŽENJU ZA RAZVOJ U VS.NET-U I ISTRAŽIĆETE TRADICIONALNE ALATE BAZA PODATAKA, KAO ŠTO SU ENTERPRISE MANAGER I QUERY ANALYZER U SQL SERVERU 2000. UČIĆETE KAKO DA SE POVEŽETE SA BAZOM PODATAKA IZ VS.NET-A I KAKO DA KORISTITE VS.NET ALATE ZA GENERISANJE VB.NET KODA KOJI DAJE PODATKE IZ SQL SERVERA.

ZAMISLITE OVO POGLAVLJE KAO PREGLED. ČITAJTE GA TAKO DA RAZUMETE ŠTA SVAKI ALAT RADI, PRATITE PRIMERE I NEMOJTE BRINUTI AKO SE POJAVI NEKO LIKO PITANJA. ALATI KOJI SE OVDE OBRAĐUJU ĆE VAM POMOĆI DA PRATITE PRIMERE U POGLAVLJIMA KOJA DOLAZE I OLAKŠAĆE VAM RAZVOJ APLIKACIJA BAZA PODATAKA. SQL QUERY DESIGNER JE, NA PRIMER, ALAT KOJI OMOGUĆAVA DA GENERIŠETE SQL ISKAZE POKAŽI-I-KLIKNI OPERACIJAMA. AKO VAM NISU POZNATI SQL ISKAZI, QUERY DESIGNER JE ODLIČNA POLAZNA TAČKA. KORISTITE GA ZA UČENJE KAKO SE PIŠU KOMPLIKOVANI SQL ISKAZI POMOĆU VIZUELNIH ALATA ZA DIZAJNIRANJE. POŠTO OVLADATE GENERISANJEM UPITA POMOĆU QUERY DESIGNEROM, MOĆI ĆETE DA KUCATE SVOJE SQL ISKAZE BEZ VIZUELNIH ALATA.

NAPOMENA

Da bismo ovo poglavlje napravili još korisnijim, uvodimo mnoge teme koje su detaljno obrađene u sledećim poglavljima. Pored toga, u ovom poglavlju smo obezbedili odličan pregled vizuelnih alata koji su sada dostupni u VS.NET-u. Ako su Vam već poznate teme kao što su SQL iskazi, snimljene procedure i slične, ovo poglavlje je i pregled i zaključak, a ako ranije niste dizajnirali bazu podataka, ili ako ne poznajete SQL, pažljivo ga pročitajte. Učičete kako da dizajnirate pomoću vizuelnih alata, a u sledećim poglavljima učičete i stvarno programiranje. ■

Da biste izgradili efikasnu, opštu aplikaciju pomoću SQL Servera, morate da ovladate različitim alatima koje nude Visual Studio .NET i SQL Server. Uglavnom se fokusiramo na novim VS.NET alatima, jer je Microsoft napravio odličan napredak u ovoj oblasti. Data Adapter Configuration Wizard je glavni kandidat za demonstriranje kako ADO.NET radi. Ni jedan drugi alat nije toliko intuitivan i lak za generisanje DataAdapter i Command objekata, eksponencijalno smanjujući vreme razvoja.

Naš fokus je uglavnom na radu sa SQL Serverom kao back-end delom. Ne samo što .NET Framework prihvata SQL Server pristup podataka u svoje osnovne klase, već Microsoft takođe prodaje SQL Server 2000 kao jedan od .NET Servera. Ovi alati su "prijateljski", laki za upotrebu i krajnje moćni. Pošto naučite T-SQL, moći ćete da napišete sopstvene snimljene procedure i pokretačke događaje. U međuvremenu ćete moći da uradite baš sve pomoću vizuelnih alata baza podataka.

Počemo sa pregledom različitih alata za pristup podacima u VS.NET-u. Kratko ćemo pomenuti alate koje nudi SQL Server 2000, uglavnom se fokusirajući na novije tehnologije. Zatim ćete učiti kako da se povežete sa postojećom SQL Server bazom podataka iz VB.NET-a korišćenjem ovih alata. Na kraju, videćete uzbudljiv primer kako možete da radite sa ADO.NET objektima da biste dobili podatke iz SQL Servera.

NAPOMENA

Iako se ovo poglavlje primarno fokusira na VS.NET vizuelne alate baza podataka, videćete da je većina VS.NET alata razvijena iz onih u SQL Serveru 2000. Microsoft "ne otkriva točak", omogućavajući Vam da trošite više vremena na ovladavanje unificiranim setom alata, nego da učite novi za svaki IDE. ■

Prvi pogled na vizuelne alate baze podataka

Vizuelni alati baza podataka VS.NET-a i SQL Servera pojednostavljuju operacije kao što su dizajn baza podataka i izvlačenje podataka iz baze. Korišćenjem prevuci-i-spusti operacija možete se brzo povezati sa bazom podataka i pronaći podatke. VS.NET vizuelni alati baza podataka potpuno podržavaju nove osobine SQL Servera 2000, kao što su indeksirani pregledi i korisnički definisane funkcije. Na taj način, sada imate moć SQL Servera 2000 direktno iz Visual Studio .NET IDE-a. Pored toga, VS.NET alati apstrahuju ADO.NET i XML model u jednu lako razumljivu vizuelnu reprezentaciju. Ovo Vam umnogome olakšava da naučite `System.Data` i `System.Xml` opsege imena ako ste početnik sa .NET Frameworkom.

Što više radite sa ovim alatima, kasnije ćete morati da pišete manje koda. To nije slično izgradnji regularnih aplikacija pomoću VB.NET-a, ako na to mislite. Da li ste ikada uzimali u obzir koliko funkcionalnosti unosite u svoje aplikacije kada spustite kontrolu iz boksa sa alatima na formu, pa podesite svojstva kontrole u Properties prozoru?

Ovo poglavlje smo fokusirali na vizuelne alate za pristup podacima; međutim, imajte na umu da sve što Vam pokažemo može biti urađeno programski. Lepota VS.NET-a je u tome što izlaže sav izvorni kod koji generišu ove vizuelne prevuci-i-spusti operacije. Ovo je korisno, jer pregledanjem koda koji generiše VS.NET možete učiti, naročito ako upravo učite novu jezičku sintaksu koju nude VB.NET i C#.

NAPOMENA

Ako ste radili sa prethodnim izdanjem SQL Servera, ovim poglavlje usmeravamo Vašu pažnju na nove vizuelne alate koje nudi SQL Server 2000 i pravimo poređenja između alata VS.NET-a i SQL Servera 2000. Ako niste radili sa SQL Serverom, takođe ćete moći da pratite detaljne primere date u knjizi. ■

SAVET

Ovo poglavlje je vrlo interaktivno. Možda ćete sada želeći da pokrenete VS.NET IDE da biste mogli da pratite objašnjenja. Ako je ovo prvi put da pokrećete VS.NET, moraćete da koristite čarobnjak za konfigurisanje svoje My Profile strane. Pošto konfigurišete svoja podešavanja, sledeće pokretanje će Vas odvesti direktno na startnu stranu. ■

Visual Studio .NET alati

Microsoftovo izdanje Visual Basic 6 je napravilo veliki napredak u unificiranju razvojnog okruženja za aplikacije i kod baza podataka. Data Environment Designer i T-SQL Debugger su omogućili da se pristupa objektima baza podataka i kodu direktno iz VB IDE-a. VS.NET unapređuje ovu unifikaciju korišćenjem deljenog okruženja za razvoj, koje povezuje sve alate za razvoj koda u jedinstven IDE. Ovo omogućava da koristite isti dizajner koda sa više jezika, kao što su VB.NET, C#, XML i SQL. Jedinstveni dizajner koda postavlja uslove za debugovanje između jezika.

Pored toga, VS.NET blisko sarađuje sa SQL Server vizuelnim alatima. U stvari, videćete mnogo istih interfejsa u VS.NET-u koje ste koristili u SQL Serveru. Ovo Vam štedi vreme, jer se možete više fokusirati na kod, a manje na učenje novih alata.

Iako postoji mnogo novih alata koji dolaze sa VS.NET-om, fokusiramo se na one koje možete koristiti za pristup podacima. Uostalom, ovo je knjiga o programiranju baza podataka. Počinjemo sa Solution Explorerom.

Solution Explorer

Solution Explorer zamenjuje Project Explorer prozor u Visual Basicu 6. Kroz Solution Explorer prozor možete pregledati različite fajlove koji pripadaju projektu, uključujući XML podatke i fajlove šema. Meta podaci projekta su smešteni u fajlu rešenja, koji ima .sln ekstenziju. Fajlovi rešenja su analogni sa grupama fajlova VB6 projekta, koji imaju .vbg ekstenziju.

Iako možete raditi sa bazama podataka iz ovog prozora, to je na neki način ograničavajuće, jer možete raditi samo sa *referencama baza podataka*, a ne sa stvarnom bazom. Reference baze podataka ne izlažu čitavu hijerarhiju objekata baze podataka, ograničavajući Vas samo na pokazivanje veza, a ne daju stvarnu mogućnost da uđete u bazu i manipulišete objektima u okviru te veze. Međutim, Server Explorer, koga ćemo obraditi u narednom odeljku, obezbeđuje ovu mogućnost.

NAPOMENA

Solution Explorer je prilično intuitivan za korišćenje, čak i ako ste novi sa VS.NET-om. Poznat Vam je ako ste radili sa Project Explorerom u prethodnoj verziji Visual Studio. Nećemo trošiti mnogo vremena na obradu Solution Explorera u ovom poglavlju. ■

Server Explorer

Jedan od najmoćnijih alata koji obezbeđuje VS.NET je Server Explorer. On je kontejner za bilo koje servise koje pokrećete na svojoj lokalnoj, ili udaljenoj mašini. Da biste pozvali ovaj prozor, kliknite Server Explorer karticu na levoj strani VS.NET IDE-a. Biće prikazan Server Explorer prozor, koji radi kao Auto-hide (automatsko skrivanje) osobina Windows Start menija. Kliknite malu ikonu da biste ga zadržali na mestu.

Server Explorer proširuje koncept Data View koji ste imali u VB6. VS.NET proširuje mogućnost manipulisanja objektima baza podataka sa potpunom mogućnošću manipulisanja svim vrstama servisa. I sve to možete uraditi preko jedne konzole! Ako gledate ovaj prozor, videćete sve Vaše servise (na primer, Database Services i Message Queue Services), kao što je prikazano na slici 3.1. Server Explorer takođe omogućava da prevučete i spustite (drag-and-drop) objekte baza podataka u dizajner komponente da biste pojednostavili programerske zadatke. Zamislite da generišete ADO.NET DataSet iz tabele baze podataka sa jednim prevuci-i-spusti pokretom.

Možete izvršavati mnoge značajne zadatke baza podataka iz Server Explorera, dodavati i editovati veze sa bazama podataka i, čak, kreirati novu bazu podataka od početka!



SLIKA 3.1 Tipični Server Explorer prozor

Toolbox - kutija sa alatima

Možete pristupiti Toolboxu na isti način na koji ste pozvali Server Explorer - samo kliknite Toolbox karticu na levoj strani VS.NET IDE-a. Na drugi način to možete uraditi tako što ćete ga izabrati iz View menija. Toolbox će izgledati poznato onima koji su radili sa Visual Studio 6 proizvodima.

NAPOMENA

Toolbox je jednostavan za korišćenje, čak i ako ste početnik sa VS.NET-om. Onima koji su radili sa Toolboxom u prethodnim verzijama Visual Studia treba da je poznat. Nećemo trošiti mnogo vremena obrađujući Toolbox u ovom poglavlju. ■

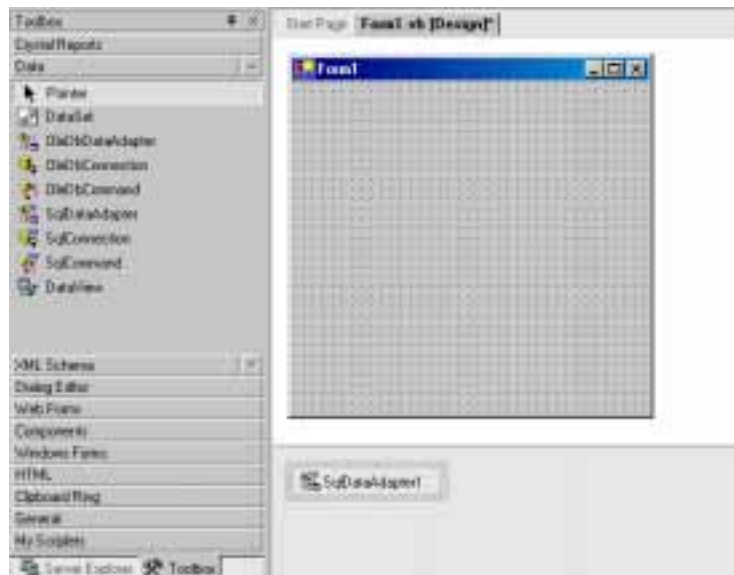
Kao što možete videti na slici 3.2, Toolbox ima novu Data karticu koja sadrži različite .NET komponente za pristup podacima, kao što su SqlDataAdapter i DataSet. Radićete sa tim komponentama na kraju ovog poglavlja.



SLIKA 3.2 Kartica Toolbox Data

Component Designer

Component Designer obezbeđuje vizuelni interfejs za manipulisanje .NET Framework objektima, kao što su ADO.NET komponente. Da biste radili sa Component Designerom, prevucite i spustite komponente iz Toolboxa u prostor za rad. Ova funkcionalnost za nevizuelne kontrole nije postojala u prethodnim izdanjima Visual Studia, jer Component Designer radi kao prostor za skladištenje, a ne kao pravi korisnički interfejs. Ovde su sačuvane samo nevizuelne komponente, kao što je Timer objekat. Tradicionalne vizuelne kontrole su prikazane na stvarnoj formi, kao u prethodnim verzijama VS.NET-a. Na slici 3.3 je pokazano da SqlDataAdapter izgleda kao da se nalazi u Component Designeru. Kada koristite DataAdapter objekte sa Component Designerom, Visual Studio .NET će automatski generisati šemu podataka za Vas. Ako želite, možete takođe koristiti Component Designer za generisanje DataSetova bez napisane linije koda.

SLIKA 3.3 *Component Designer*

Database Designer

Kao što ime pokazuje, Database Designer je alat za dizajniranje postojećih baza podataka. On radi u vezi sa Server Explorerom, koji omogućava da kreirate nove baze podataka, tabele, snimljene procedure i dijagrame. Pošto kreirate bazu podataka, možete koristiti Database Designer za izgradnju baze podataka (za definisanje njenih tabela, relacija i tako dalje).

Eksperimentišimo sa kreiranjem nove baze podataka iz VS.NET-a. Prvo, proširite Servers čvor u Server Exploreru. "Zaronite" u SQL Server, proširite svoje SQL Server ime i izaberite New Database, kao na slici 3.4. Definišite ime, kao što je TestDb, i kliknite OK. Database Designer će se priključiti direktno na SQL Server, kreirajući SQL Server bazu podataka umesto Vas. Od Vas će biti zatraženo da unesete informacije za autentikaciju. Imate opciju da koristite autentikaciju SQL Servera, definišući login ime i password. Alternativa je korišćenje Windows NT integrisanog obezbeđenja.

UPOZORENJE

Budite svesni ograničenja Server Explorera. Iako možete kreirati nove baze podataka iz SQL Server čvora, ne možete da ih brišete. Moraćete da koristite SQL Server Enterprise Manager za brisanje bilo koje baze podataka. ■



SLIKA 3.4 Kreiranje nove SQL Server baze podataka iz Server Explorera

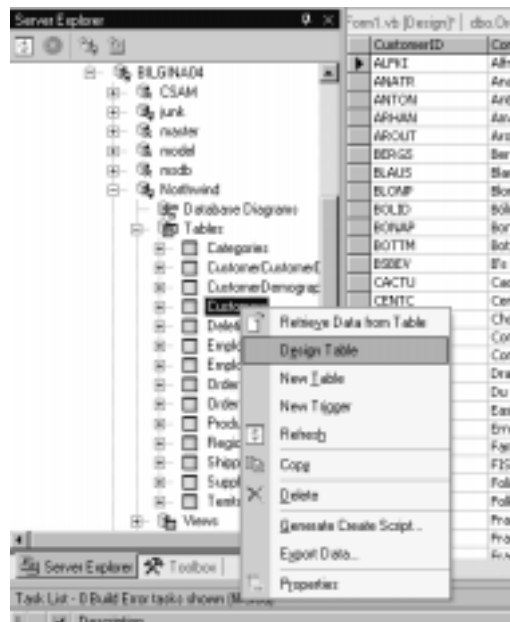
Table Designer

Pored toga što možete kreirati novu bazu podataka, možete raditi sa objektima baza podataka u okviru baze podataka, kao što su tabele, snimljene procedure i korisnički definisane funkcije. Proširite čvor za Northwind bazu podataka. Videćete različite objekte ispod čvora baze. Proširite Tables čvor da biste videli Northwind tabele. Duplo kliknite Customers tabelu, što će omogućiti prikaz svih zapisa u toj tabeli. Ovo je pogodan način da se edituju postojeći redovi, ili da se dodaju novi redovi tabeli. Ovo funkcioniše upravo kao SQL Server 2000 Enterprise Manager.

Kao i sa SQL Serverom, možete da kliknete desnim tasterom miša iznad Customers tabele i izaberete jednu iz mnoštva akcija koje možete izvršiti nad tabelom, kao što je prikazano na slici 3.5.

Kao što možete videti na iskaćućem meniju, možete dodati nove tabele, ili pokretačke događaje, ili izvoziti podatke direktno iz Visual Studio .NET-a. Izaberite Design Table iz iskaćućeg menija da biste pokrenuli Table Designer. Promene koje ovde izvršite čuvaju se u povezanoj bazi podataka.

Microsoft je postigao neka poboljšanja na Table Designeru u VS.NET-u. Oni koji su radili sa njim u Visual Studiu 6 (VS6) prvo će primetiti poboljšani korisnički interfejs, konzistentan sa osobinama interfejsa koje ćete naći u VS.NET IDE-u. Mrežastom interfejsu Microsoft dodaje prozor sa svojstvima koji možete koristiti da pregledate sve detalje kolona (pogledajte sliku 3.6). Mreža čuva prostor prikazivanjem samo najčešće korišćenih svojstava kolona.



SLIKA 3.5 Opcije za dizajn tabela iz Server Explorera



SLIKA 3.6 Poboļjšani VS.NET Table Designer

U prethodnim verzijama bili ste ograničeni na mrežasti pregled atributa tabele. Interfejs je bio vrlo glomazan, pošto je zahtevao puno horizontalnog pomeranja da bi se pogledala svojstva kolona. Programeri VS.NET-a su dobili inspiraciju za najbolji alat za upravljanje SQL Serverom: za SQL Server Enterprise Manager. Kao što možete videti, VS.NET Table Designer umnogome radi kao alat za dizajniranje tabela u SQL Enterprise Manageru i čak koristi sličan interfejs. Dizajner prezentuje kolone u mreži na vrhu. Klikom na kolonu ispod se prikazuje povezani detalj (kao što će kasnije biti prikazano na slici 3.16).

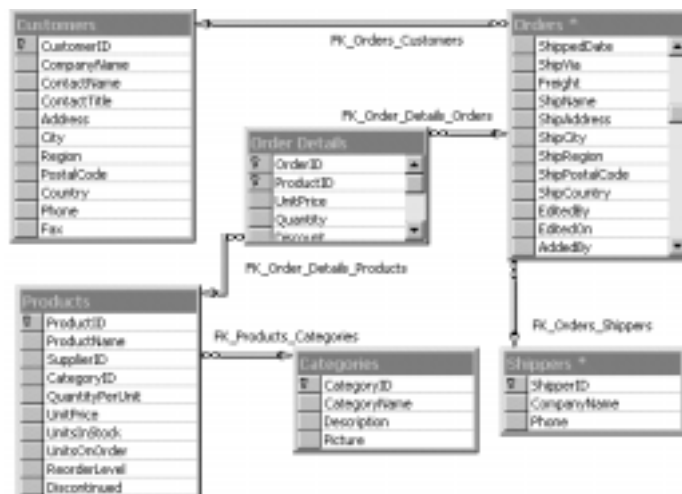
Database Diagram Designer

Database Diagram Designer je zamenio koncept dizajnera relacija; međutim, možete videti da se nije promenio mnogo od Visual Studia 6. Možete pristupiti ovom dizajneru proširivanjem Database Diagrams čvora u Server Exploreru.

UPOZORENJE

Ako nemate nikakve dijagrame u bazi podataka, nećete moći da radite sa Diagram Designerom. Biće Vam potrebno da kreirate novi dijagram klikom desnim tasterom miša iznad Database Diagrams čvora u Server Exploreru i da izaberete New Diagram. ■

Kao Table Designer, Diagram Designer oponaša funkcionalnost Enterprise Managera i izgleda identično sa Diagram Designerom, koji se nalazi u SQL Server 2000 Enterprise Manageru, kao što možete videti na slici 3.7. Kada kliknete relaciju, imate mogućnost da editujete njena svojstva. Možete takođe da iskoristite nove osobine relacija SQL Servera 2000, kao što su kaskadno ažuriranje i brisanje.



SLIKA 3.7 Koristite Database Designer za pregledanje strukture tabela i za uspostavljanje relacija između njih.

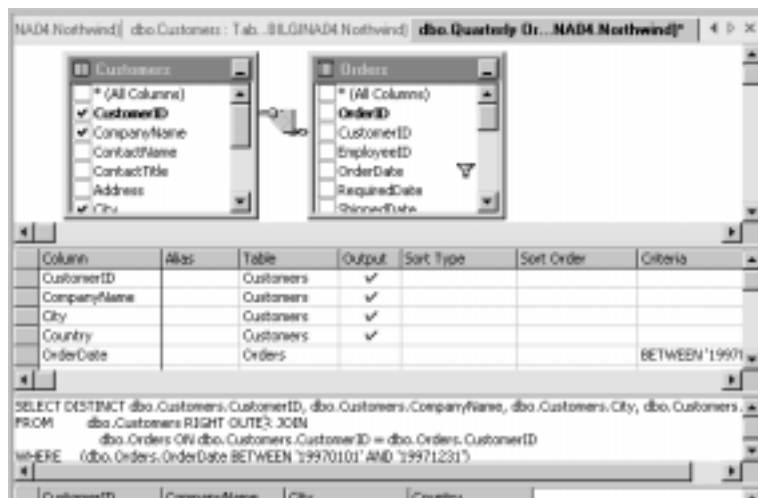
Query Designer i View Designer

Query Designer je zasnovan na prilično staroj tehnologiji, poznatoj kao query by example (QBE). Bez obzira što je staromodan, ovo je još uvek odličan vizuelni alat, naročito za ljude kojima nije poznat SQL. Query Designer, prikazan na slici 3.8, omogućava da gradite SQL iskaze pomoću pokaži-i-klikni operacija. Vi, jednostavno, stavite tabele koje želite da uključite u upit na pano Query Designera, izaberete polja koja želite da koristite u upitu i, na kraju, definišete ograničenja. Query Designer je "inteligentan" i fleksibilan alat i verovatno ćete ga više koristiti od bilo kog drugog vizuelnog alata baze podataka. Čak i programeri kojima su SQL iskazi poznati koriste Query Designer za dizajniranje upita, naročito ako se koriste mnoga imena polja iz mnogo tabela, da bi se smanjile greške pri kucanju.

Zbog toga što pregledi i upiti imaju mnogo čega zajedničkog, Query Designer interfejs može biti korišćen za rad i sa pregledima.

NAPOMENA

View Designer koristi Query Designer za dizajniranje pregleda, koje možete videti ako kliknete desnim tasterom miša iznad pregleda u Server Exploreru, a, zatim, izaberete Design View opciju. ■



SLIKA 3.8 Tipični Query i View Designer prozor

XML Designer

Kao što ste možda već pretpostavili, XML igra važnu ulogu u Visual Studio .NET-u i .NET Frameworku. Da bi podržao ovu ideju, VS.NET obezbeđuje vizuelne alate za rad sa XML-om i XML šemama. Jedan od najvažnijih alata u ovoj oblasti je XML Designer. Zbog toga što se ADO.NET DataSetovi skladište kao XML, možete da manipulišete DataSetovima, XML Schema Definitions(XSD) i XML fajlovima podataka korišćenjem XML Designera. XML Designer omogućava one akcije koje ne možete sprovesti korišćenjem drugog alata, kao što je vizuelno dodavanje DataRelation objekata između različitih DataTables.

XML Designer obezbeđuje moćan alat za kombinovanje podataka iz različitih izvora podataka u ADO.NET DataSet, bez kucanja linije koda. Korišćenjem XML Designera možete da vidite predstavljenu šemu podataka i da editujete XSD fajl u kome je šema sačuvana.

XML Designer obezbeđuje tri pregleda DataSeta: mrežasto predstavljanje podataka, neobrađeni XML i pregled šeme. Ne govorimo mnogo o XML Designeru u ovom poglavlju. Detaljno ćemo obraditi XML kod u Poglavlju 7, "ADO.NET programiranje", gde ćete raditi sa XML šema fajlovima, i u Poglavlju 8, "Data Aware kontrole", gde ćemo pokazati kako da dodate DataRelations iz XML Designera praćenjem primera Shopping Cart. Osim toga, ponovo obrađujemo XML Designer u Poglavlju 10, "Uloga XML-a".

SQL Editor

Editor koda u VS.NET-u ne podržava samo .NET jezike, kao što su VB.NET i C#, već može da rukuje i SQL kodom. SQL Editor omogućava da kreirate novu snimljenu proceduru i pokretačke događaje, ili da editujete postojeće. Možete, takođe, da editujete i debugujete snimljene procedure iz Visual Basic okruženja. SQL Editor nije pravi vizuelni alat, ali, zbog toga što je blisko povezan sa Query Designerom, uključujemo ga u ovo poglavlje.

NAPOMENA

VS.NET povezuje razvojno okruženje i sa integrisanim editorom koda. SQL Editor se češće naziva Code Editor, pošto je to unificirani kod i tekst editor za više jezika. Nazivamo ga SQL Editor u ovoj knjizi da bismo ga razlikovali između editovanja SQL koda nasuprot VB.NET kodu. Editor koda se prilagođava prema jeziku koji koristite. Zbog ovoga, određene osobine, kao što je IntelliSense, ne aktiviraju se pri editovanju SQL koda. ■

VS.NET unifikacija jezika se ne zaustavlja kod prostog editovanja. Možete, takođe, debugovati kod u kome su pomešani jezici i sve to iz istog IDE-a. Debugovanje više jezika ne omogućava samo debugovanje koda između VB.NET-a i SQL-a, već i između bilo kog drugog VS.NET jezika, kao što je C#. Visual Studio 6 nije podržavao ovu (sa mnogih strana zahtevanu) osobinu.

Alati za debugovanje su isti između jezika, omogućavajući da debugujete SQL snimljene procedure, pokretačke događaje i funkcije kao da su regularan VB.NET kod. Ako želite da debugujete snimljenu proceduru, proširite Stored Procedure čvor iz Server Explorera. Kliknite desnim tasterom miša iznad snimljene procedure koju želite da debugujete i izaberite opciju Step Into Stored Procedure iz iskaćućeg menija, kao što je prikazano na slici 3.9. Ako imate bilo kakve ulazne parametre, pojaviće se okvir za dijalog, tražeći vrednosti za njih. Zatim će Vas VS.NET debager provesti kroz snimljene procedure. Integrisani VS.NET debager menja T-SQL Debugger koji ste koristili u prošlosti.

UPOZORENJE

Ako ne možete da editujete i debugujete snimljene procedure, možda koristite Professional izdanje Visual Studia. Da biste editovali i debugovali snimljene procedure, mora biti instaliran Enterprise Architect. Možda želite da razmislite o nadgradnji Enterprise Architect izdanja posle pregledanja njegovih osobina, koje su prikazane u MSDN-u. ■



SLIKA 3.9 VS.NET Debugger omogućava interaktivno debugovanje snimljenih procedura.

Alati SQL Servera 2000

Ovaj deo obezbeđuje pregled visokog nivoa najvažnijih alata sa kojima možete raditi u SQL Serveru 2000. Ne predstavljamo sve alate koji su dostupni, već samo one koji su potrebni da znate da biste radili sa primerima iz ove knjige. U kasnijim poglavljima uvode se drugi vizuelni alati i čarobnjaci dok budete prelazili na naprednije teme.

Enterprise Manager

Ako ste ranije radili sa SQL Serverom, verovatno Vam je poznat Enterprise Manager. Slično SQL Server čvoru u Server Exploreru, u VS.NET-u Enterprise Manager daje vizuelni interfejs za primenu deklarativnih podešavanja nad objektima baza podataka. Skoro sve što uradite u Enterprise Manageru može biti napisano u SQL kodu, tako da možete zamisliti kako Enterprise Manager smanjuje potrebno vreme za kodiranje.

Query Analyzer

Query Analyzer SQL Servera 2000 je mnogo robustniji od prethodnih verzija. Najveća promena koju ćete primetiti je u Object Browseru. Oni koji programiraju pomoću VB.NET-a mogu ovu osobinu da smatraju "osveženjem", pošto ona daje objektno-orijentisani pogled objekata baze podataka. Nema potrebe da koristite Alt+Tab da biste prelazili između pisanja koda upita i referenciranja imena objekta u Enterprise Manageru. Query Analyzer Vam čak štedi i vreme pri kodiranju, jer svi objekti baze podataka koje vidite u ovom prozoru mogu biti prevučeni i spuštteni u Vaš kod. Pored toga, on obezbeđuje robustan debager za snimljene procedure, koji se odlično integriše sa VS.NET okruženjem za debugovanje.

Ovde pominjemo ove SQL Server 2000 alate da biste mogli da razumete kako se porede sa VS.NET vizuelnim alatima baza podataka. Nećemo potrošiti mnogo vremena na SQL Server IDE u ovom poglavlju, jer je najbolji način za razumevanje alata SQL Servera 2000 rad sa njima, koga ćete se "latiti" pošto istražite T-SQL sintaksu u Poglavlju 4, "Structured Query Language", i Poglavlje 5, "Transact-SQL".

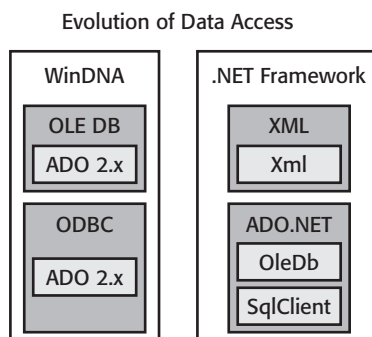
U sledećim odeljcima istražićete svaki od VS.NET alata. Prvo ćete pogledati različite strategije za uspostavljanje veza sa bazama podataka. Zatim ćete učiti kako da se povežete sa bazom podataka iz VS.NET-a. Posle svega, bez obzira šta ćete raditi sa bazom podataka, morate se na nju prvo povezati.

Strategije za pristup podacima

Pre nego što se stvarno povežete sa bazom podataka, potrebno je da prvo pregledate opcije za pristup podacima koje imate u .NET Frameworku. Vaša strategija za pristup podacima igra vitalnu ulogu ako želite da dizajnirate kvalitetno rešenje. Izborom odgovarajuće strategije za pristup podacima Vaša aplikacija će imati bolje performanse i skalabilnost. Razumevanjem ovih koncepata možete osigurati efikasnost svoje aplikacije.

U ovom odeljku obrađujemo strategije koje možete koristiti kada pristupate podacima korišćenjem .NET-a. Pokazujemo kako se arhitektura promenila od WinDNA arhitekture i objašnjavamo prednosti korišćenja .NET-a.

Pristup podacima se dramatično menja sa pojavom .NET Frameworka. Microsoft ne očekuje da promenite svoje tehnologije za pristup podacima "preko noći", tako da je dizajniran .NET Framework da bude unazad kompatibilan sa postojećim tehnologijama. Međutim, ako želite da Vaša aplikacija bude dovoljno skalabilna, treba da ozbiljno posmatrate prelazak na jednu od novih tehnologija za pristup podacima. Na slici 3.10 pokazana je evolucija strategije Microsofta za pristup podacima.



SLIKA 3.10 Evolucija pristupa podacima

Možete izabrati da koristite postojeće provajdere podataka, ili nove .NET Framework tehnologije za pristup podacima. Najčešći izbori kod pristupa podacima su:

- ODBC
- OLE DB
- ADO 2.x
- ADO.NET

ODBC

ODBC je tehnologija za pristup podacima kreirana za rad sa relacionim bazama podataka. ODBC je stariji drajver i mnoge nasleđene aplikacije ga danas koriste. To je vrlo rasprostranjen drajver i postoje ODBC drajveri za većinu baza podataka. Međutim, za svoje nove aplikacije treba da koristite OLE DB, jer je mnogo nezavisniji od svog izvora podataka nego ODBC. Pored toga, dok se ODBC arhitektura fokusira više na relacione baze podataka, OLE DB može biti korišćen i za relacione i nerelacione izvore podataka, kao što su poruke, XML i skladišta podataka koja su zasnovana na fajlu.

NAPOМЕНА

Pojava .NET platforme ne čini sve tehnologije za pristup podacima zastarelim. Dok novi .NET provajderi postaju dostupni, oni će biti dostupni za preuzimanje sa Microsoftove web prezentacije. Na primer, ODBC.NET je .NET provajder podataka koji možete koristiti za povezivanje sa nasleđenim ODBC sistemima. ■

OLE DB

Za razliku od ODBC-a, OLE DB podržava široku lepezu izvora podataka - relacionih i nerelacionih. Ovo ga čini odličnim izborom za povezivanje heterogenih izvora podataka. OLE DB još uvek postoji sa .NET Frameworkom; međutim, i dalje koristi stari COM OLE DB provajder podataka. Ovo znači da ćete imati slabije performanse, jer .NET provajderi podataka moraju da transliraju pozive korišćenjem COM upravljanja. Treba da koristite OLE DB kada pristupate verzijama SQL Servera koje su prethodile SQL Serveru 7, ili drugim izvorima podataka. Bez obzira da li koristite OleDb, ili SqlConnection opseg imena, ADO.NET daje uobičajeni model za generisanje DataSetova i DataReadera.

ADO 2.x

Klasičan ADO omogućava objektno-orijentisani interfejs za rad sa OLE DB-om. Zbog toga što može da podrži disconnected RecordSetove sa strane klijenta, ADO je koristan za Internet aplikacije. Iako je ADO originalno bio dizajniran za relacione podatke, možete još uvek da napravite hijerarhijsku reprezentaciju podataka korišćenjem OLE DB oblikovanja podataka, ili možete da konvertujete ADO RecordSetove u XML strukture.

Međutim, u poređenju sa ADO.NET-om, ADO 2.x je sasvim ograničavajući. Na primer, sa ADO RecordSetovima je nezgrapno raditi sa više tabela. Možete vratiti više tabela u jedan RecordSet, ili koristiti više RecordSetova za svaku tabelu. Pored toga, ADO RecordSetovi ne podržavaju koncept relacija između RecordSetova, koji dobijate kada koristite ADO.NET DataSetove.

Pored toga, klasični ADO se ne integriše blisko sa XML-om. ADO 2.x tretira XML kao još jedan mehanizam formata fajla, a ne koristi ga u osnovnom obliku. Zbog toga što je XML postao standard za razmenu podataka u poslu, važno je imati mehanizam za pristup podacima koji ga podržava u osnovi.

ADO 2.x RecordSetovi su malo prepunjeni, jer su napravljeni da rade kao disconnected i connected, da rade i sa relacionim i hijerarhijskim podacima i da rade sa XML-om, ili bez njega. Zbog ovoga su ADO RecordSetovi fleksibilni, ali teški. Dok su se pojavljivale Internet aplikacije, bilo je potrebe za slojem za pristup podacima koji je lakši, suštinski disconnected i blisko povezan sa XML-om. Ovo je prirodno vodilo prema evoluciji ADO.NET-a u okviru .NET Frameworka.

ADO.NET

ADO.NET prenosi apstrakciju podataka korak dalje od ADO-a, obezbeđujući jasno razdvajanje između sistema za pristup podacima (DataAdapter) i podataka (DataSet, DataReader). Pored toga, ADO.NET jasno razdvaja skladište podataka od šeme podataka, slično kao XML. Sa druge strane, ADO 2.x se fokusira na model zasnovan na bazi podataka, pre nego na pregled podataka nezavisno od njihovog izvora.

ADO.NET uvodi koncept upravljanih provajdera. Upravljeni provajderi u .NET-u menjaju neupravljanu OLE DB provajdere. Upravo sada aktuelno izdanje .NET Frameworka podržava dva upravljana provajdera - jedan za SQL Server, a drugi za OLE DB za korišćenje sa bilo kojom drugom vrstom izvora podataka.

ADO.NET koristi XML za čuvanje podataka, što aplikacijama olakšava da dele podatke. Zbog ovoga, ADO.NET simulatno skladišti podatke i u relacionom i hijerarhijskom formatu.

SAVET

Ako razmišljate da li da nadgradite ADO 2.x komponente na ADO.NET, dobar razlog da ovo uradite je da omogućite da Vaši podaci prolaze kroz firewall. Iako DCOM obezbeđuje prolazak kroz TCP/IP, za šta koristi Port 80, firewallovi mogu još uvek da filtriraju bilo koji saobraćaj koji nije HTTP, kao što je vlasničko COM upravljanje. ADO.NET koristi HTTP kao svoj protokol za transfer, koji je široko prihvaćen i svuda prisutan standard. Pored toga, uzmite u obzir prijemnu aplikaciju. Skoro sve aplikacije mogu da podrže XML podatke, pre nego binarni COM-zasnovani RecordSet. ■

Postoje dva upravljana provajdera koje obuhvata ADO.NET: opsezi imena SqlConnection i OleDb (oba se nalaze u System.Data opsegu imena).

SqlConnection opseg imena

Ako sigurno znate da je Vaš izvor podataka SQL Server, definitivno koristite ovaj opseg imena, umesto OleDb opsega. Dobićete mnogo više brzine kada pristupate svojim podacima, jer idete direktno kroz SQL Server fajl sistem, a ne kroz OLE DB sloj.

OleDb opseg imena

Koristite OleDb upravljeni provajder za sve ostale izvore podataka, uključujući nerelacione baze podataka, kao što su skladišta poruka i tekstualni fajlovi.

SAVET

OleDb upravljeni provajder nije toliko efikasan kao SQL upravljeni provajder, jer on još uvek mora da ide kroz COM meduoperativni sloj da bi komunicirao sa OLE DB korišćenjem COM-a. Upravljeni provajderi nude bolje performanse, pošto uopšteno izbegavaju COM i komuniciraju direktno sa izvorom podataka korišćenjem osnovnih API poziva. Nažalost, OleDb upravljeni provajder koji dolazi sa .NET-om ne radi ovo stvarno, jer su OLE DB provajderi zasnovani na COM-u. Ovo ga usporava, pošto mora da prevodi COM kod i tipove podataka u .NET kompatibilne objekte. U ovom trenutku je SQL Server jedini prvobitni provajder podataka koji se isporučuje sa .NET-om; međutim, Microsoft očekuje da ovaj broj poraste. ■

I OleDb i SqlConnection opsezi imena koriste isti mehanizam za skladištenje podataka: DataSet, ili DataReader. Ne morate da brinete o detaljima ADO.NET-a u ovom poglavlju. VS.NET vizuelni alati baza podataka rukuju većim delom posla, umesto Vas. U Poglavlju 7 ćete “zaviriti ispod haube” različitih ADO.NET objekata, kao što je DataAdapter.

Korišćenje stringova za povezivanje

U ADO.NET-u Connection objekat omogućava da komunicirate sa svojim izvorom podataka. Da bi Connection objekat “znao” kako da se poveže sa svojim izvorom podataka, potrebno je da definišete string za povezivanje, koji sadrži različite parametre veze. Imate nekoliko načina za kreiranje stringa za povezivanje. To možete uraditi vizuelno, ili programski korišćenjem Data Link Properties okvira za dijalog:

- Specificirajte ime eksternog fajla koji čuva informaciju o vezi. Ovaj fajl je poznat kao *data link file* i ima ekstenziju .udl. Problem sa ovim pristupom je što morate da pripremite različiti fajl za vezu za svakog korisnika na mreži.
- Koristite *Data Link Properties* okvir za dijalog iz VS.NET-a da vizuelno kreirate, ili editujete svoj string za povezivanje. Kasnije u ovom poglavlju biće prikazan ovaj okvir dodavanjem veze kroz Server Explorer. Ako kreirate vezu korišćenjem ovog okvira za dijalog, neće biti kreiran .udl fajl za Vas, već će biti dinamički povezivani atributi u string za povezivanje, što se može videti u svojstvima veze u VS.NET-u.
- Programski definišite *string za povezivanje*, koji prosleđuje sve informacije potrebne drajveru za uspostavljanje veze sa bazom podataka. Ova informacija uključuje ime mašine na kojoj se izvršava SQL Server, informacije o autentikaciji i ime baze podataka. Problem sa ovim prilazom je što morate da kodirate sve informacije potrebne za povezivanje sa bazom podataka. Međutim, možete postaviti svoj string za povezivanje na mesto na kome je lakše editovanje, kao što je *web.config* fajl. Radićete sa ovim opcijama kasnije u Delu IV, “Pristup podacima na Webu”, kada budete graditi web aplikacije.

Korišćenje data link fajlova

Data link fajlovi su tekstualni fajlovi koje možete kreirati iz Windows Explorera. Da biste kreirali data link fajl u Windowsu 2000, otvorite fasciklu gde želite da kreirate fajl i kliknite desnim tasterom miša iznad praznog prostora. U meniju koji se pojavi izaberite New ➤ Text Document. Promenite ekstenziju fajla na .udl. Ikona fajla se automatski menja u Microsoft Data Link ikonu. Zatim, kliknite desnim tasterom miša iznad data link fajla da biste otvorili meni. Izaberite opciju Properties. Ovo otvara Data Link Properties okvir za dijalog.

Korišćenje Data Link Properties okvira za dijalog

Pored korišćenja data link fajla, možete pristupiti ovom okviru za dijalog iz Server Explorera u VS.NET-u. Data Link Properties okvir za dijalog sadrži četiri kartice:

PROVIDER Ovo je mesto gde definišete provajder koji će biti korišćen za povezivanje sa bazom podataka. Ostale kartice će dinamički reflektovati svojstva koja se primenjuju na provajderu izabranom na ovoj kartici.

CONNECTION Ovo je mesto gde definišete izvor podatka veze (baza podataka na koju želite da se povežete).

ADVANCED Ova kartica omogućava da definišete neke napredne opcije povezivanja.

ALL Ova kartica sadrži sve atribute veze (većina od njih je definisana na druge tri kartice) i možete editovati njihove vrednosti.

Programsko deklarisanje stringa za povezivanje

Vizuelni alati će, umesto Vas, kreirati string za povezivanje. Alternativa je da možete da prosledite ovaj string za povezivanje direktno u ADO.NET Connection objekat. Na primer, ako želite da se povežete sa Northwind bazom podataka, kucate sledeće:

```
Dim nwindConn As SqlConnection = New SqlConnection  
("Integrated Security=SSPI; Initial Catalog=Northwind;Data Source=SA")
```

NAPOMENA

Data Source svojstvo predstavlja ime servera baze podataka; zbog toga, zamenite ovaj parametar imenom sopstvenog SQL Servera. ■

Kao što možete videti, atributi stringa za povezivanje odgovaraju poljima koja ste videli u Data Link Properties okviru za dijalog. Da biste vizuelno kreirali string za povezivanje, otvorite Server Explorer prozor, kliknete desnim tasterom miša iznad Data Connection ikone i iz menija izaberite Add Connection. Videćete Data Link Properties prozor, koji ima iste četiri ranije obrađene kartice.

Povezivanje sa bazom podataka

Sada, kada ste pregledali alate i stragetije za pristup podacima, spremni ste da počnete da generišete kod. Započecete sa najlogičnijim mestom za početak kodiranja aplikacije za pristup bazi podataka - sa VS.NET-om. U ovom odeljku prvo ćete naučiti kako se vizuelno kodiraju aplikacije. Zatim ćete pogledati kod generisan iza scene i naučiti svrhu svakog objekta.

Kasnije, u Poglavlju 7, učicete moćniji kod za manipulisanje objektima baze podataka. U svim primerima u ovom odeljku koristi se T-SQL kod koji već postoji iz Northwind baze podataka. U sledećih nekoliko poglavlja naučićete kako da pišete sopstveni korisnički T-SQL kod.

Povezivanje sa Northwind bazom podataka korišćenjem Windows projekta

Povežite se sa Northwind bazom podataka iz VB.NET-a i počnite rad sa novim alatima. Počnite sa pokretanjem VS.NET IDE-a (ako ga već niste pokrenuli) i kreiranjem novog projekta sa Start strane.

Pošto ste izabrali New Project dugme, videćete listu izbora projekta. Koristićete Windows Application i Database Project tipove u ovom poglavlju; ako Vas interesuju drugi tipovi projekata, tabela 3.1 pomaže da ih uporedite sa Visual Basic 6 tipovima projekata.

Tabela 3.1: VB.NET u odnosu na VB 6 tipove projekata

.NET tip projekta	VB6 poređenje
Windows Application	Slično VB6 izvršnom programu
Class Library	Slično ActiveX DLL-ovima i ActiveX EXE fajlovima
Windows Control Library	Slično ActiveX kontroli
ASP.NET Web Application	Slično InterDev web projektima
ASP.NET WebService	Nema direktnog poređenja (analogno sa udaljenom COM komponentom koja koristi HTTP i SOAP, a ne DCOM)
Web Control Library	Nema direktnog poređenja (analogija sa ActiveX kontrolama, ali za web stranu, pre nego za Windows aplikaciju)
Console Application	Nema poređenja sa VB6 (slično sa C++ aplikacijama sa komandne linije)
Windows Service	Nema poređenja sa VB6
Empty Project (prazan projekat)	Nema poređenja sa VB6
Empty Web Project (prazan web projekat)	Nema poređenja sa VB6
New Project In Existing Folder (novi projekat u postojećoj fascikli)	Nema poređenja sa VB6

Izaberite ime i putanju fajla za svoju Windows aplikaciju. Birajte `NWProject` za ime projekta. Solution Explorer će pokazati sve podrazumevane fajlove koji su kreirani pri kreiranju Vašeg novog projekta. Kao što smo istakli ranije, Solution Explorer je sličan Project Exploreru u VB6. Jedan od fajlova kreiranih pri kreiranju Vašeg projekta je standardna Windows forma, koju možete videti u prozoru dizajnera forme.

N A P O M E N A

Oni koji su radili sa prethodnim verzijama Visual Basica mogu da budu uplašeni kada vide da Visual Studio .NET automatski čuva projekat na disku pri kreiranju. Ranije ste mogli da gradite aplikaciju "za bacanje jednokratnu upotrebu" kreiranjem novog projekta i njegovim zatvaranjem bez čuvanja. Onima koji su radili sa Visual InterDevom ova osobina će biti poznata. ■

Pre nego što vidite vizuelne alate baza podataka u akciji, morate se povezati sa bazom podataka i kreirati nekoliko komandi koje pronalaze informacije iz baze podataka. Pratite naredne korake za uspostavljanje veze sa SQL Server Northwind bazom podataka da biste proverili da li možete da pratite primere u ovom poglavlju (prateći slične korake, možete se povezati na Oracle, ili Access bazu podataka).

1. Otvorite Server Explorer prozor i kliknite desnim tasterom miša iznad Data Connection čvora. Izaberite Add Connection iz menija.
2. Ovo će prouzrokovati pojavljivanje prozora gde možete da podesite Data Link svojstva. Ovo je slično Data Link prozoru, koji ste videli u prethodnim izdanjima.

3. Podrazumevano stanje je da će aktivna kartica biti Connection, kao što možete videti na slici 3.11 (ovo je jedina kartica na koju se fokusiramo u ovom primeru; druge smo obradili prethodno u ovom poglavlju). Definišite ime Vašeg servera baze podataka i Vaš metod za autentikaciju i bazu podataka koju želite da koristite. U ovom slučaju koristite Northwind bazu podataka na svojoj lokalnoj mašini. Primetićete da se neće pojaviti ime baze podataka na inicijalnoj padajućoj listi baze podataka dok ne specificirate korisnički ID i password - naravno, ako ne izaberete opciju Windows NT integrisane bezbednosti.

NAPOMENA

VS.NET automatski konfiguriše provajder podataka, umesto Vas. Na kartici Provider okvira za dijalog Data Link Properties možete videti da provajder koristi Microsoft OLE DB provajder za SQL Server. ■



SLIKA 3.11 Povezivanje sa bazom podataka korišćenjem Connection kartice

4. Kliknite Test Connection dugme na dnu prozora da biste proverili da li sve funkcioniše, pa kliknite OK da biste zatvorili okvir za dijalog.

To je sve što morate da uradite da biste se povezali sa Northwind bazom podataka. Ako pogledate Server Explorer, videćete svoju novu vezu na stablu. Ako proširite fasciklu, videćete povezane objekte baze podataka, kao što su tabele, pregledi i snimljene procedure.

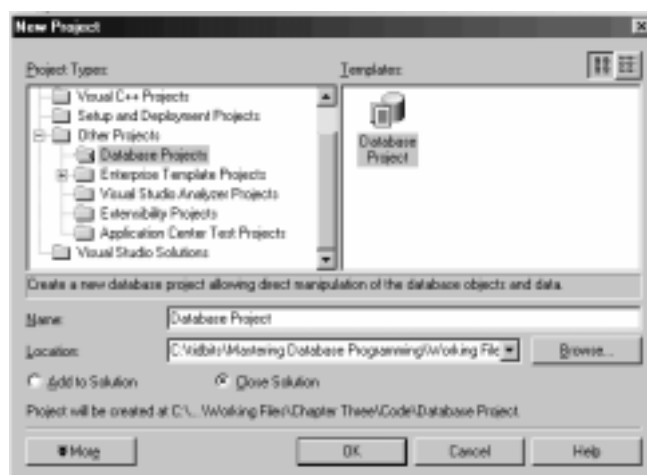
UPOZORENJE

Ne možete editovati postojeću vezu izborom Properties iz njenog menija. Properties okvir za dijalog će prikazati svojstva veze, ali će ona biti siva i read-only. Da biste promenili svojstva veze sa podacima, morate ukloniti postojeću vezu sa podacima i uspostaviti novu. ■

Povezivanje sa Northwind bazom podataka korišćenjem Data Projecta

Drugi način za kreiranje veza sa bazom podataka iz VS.NET-a je unapređivanje postojećeg tipa projekta nazvanog *Database Project*. Projekat baze podataka služi kao šablon koji sadrži različite objekte potrebne za povezivanje sa bazom podataka i njeno programiranje iz VB.NET koda. Pratite sledeće korake:

1. Kreirajte nove projekte izborom **File** → **New Project** iz Vašeg menija. U pregledu projekta tipa stabla pređite na fasciklu **Other Projects** i izaberite **Database Projects** fasciklu.
2. Izaberite ime projekta i putanju za Vaš fajl. Proverite da li ste ostavili potvrđenu opciju **Close Solution**. Kliknite šablon **Database Project**, kao što je prikazano na slici 3.12.



SLIKA 3.12 Izaberite novi Database Project

3. Sledeće što ćete videti je okvir za dijalog koji postavlja pitanje da li želite da koristite postojeću referencu na bazu podataka, ili ćete da kreirate novu. Zbog toga što ste već kreirali vezu sa Northwindom u prethodnom primeru, videćete je na listi u ovom boksu. Označite je i kliknite OK. Ako niste imali nikakve kreirane veze u Server Exploreru, videli biste **Data Link Properties** okvir za dijalog sa kojim ste radili ranije.

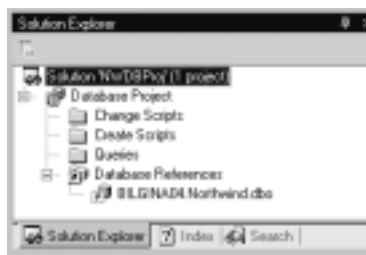
NAPOMENA

Imajte na umu da referenca baze podataka radi različito od veza u Server Exploreru. Referenca na bazu podataka je pokazivač na vezu u Server Exploreru. Osim toga, reference baze podataka su uskladištene sa projektom baze podataka. Kada ponovo otvorite projekat, videćete reference baza podataka. Ako veza sa podacima nije više važeća, VS.NET će, umesto Vas, automatski ponovo kreirati vezu. ■

Za razliku od navedenog primera Windows aplikacije, nećete videti vizuelni dizajner, kao što je forma. Takođe ćete primetiti razliku u Solution Explorer prozoru, kao što je prikazano na slici 3.13. U ovom prozoru videćete nekoliko fascikli: **Change Scripts**, **Create Scripts**, **Queries** i

Database References. Change Scripts fascikla omogućava da kreirate i modifikujete postojeće objekte baza podataka. Omogućava da uskladištite skript za modifikovanje za promenu objekata baze podataka. Vi biste kreirali skript za promenu kada niste sasvim spremni da implementirate promene direktno u bazu podataka. Create Scripts fascikla obezbeđuje mesto za skladištenje definicije skripta objekata baze podataka. Ova opcija nije bila dostupna u prethodnim verzijama Visual Studia. Fascikla Queries obezbeđuje mogućnost da pristupate Query Designeru, kao što smo objasnili ranije. Na ovaj način, možete kreirati iskaze za manipulisanje podacima da biste pristupili podacima.

Ako proširite Database References fasciklu, videćete da je, umesto Vas, VS.NET automatski kreirao referencu na Northwind bazu podataka - ona izgleda kao Connection objekat, koji ste ranije dodali u Server Exploreru. Međutim, za razliku od Server Explorera, ovo je samo pokazivač na vezu, tako da se ne možete udubiti u objekte koji se nalaze u okviru veze baze podataka iz reference baze podataka.



SLIKA 3.13 Komponente novog Database Project šablona

Zaključićete da je Database Projects korisno mesto za centralizovanje referenci baza podataka i odlično mesto za pokretanje SQL koda. Kasnije u ovom poglavlju naučićete kako da pokrenete upite i skriptove. Sačuvajte ovaj projekat kao Database Project, jer ćete ga koristiti ponovo u odeljku "Query Designer Window".

Korišćenje Server Explorera

Bez obzira da li ste koristili Windows Application, ili Database Project za kreiranje svoje veze, pošto ste ustanovili vezu sa bazom podataka, objekti definisane baze podataka će se automatski pojaviti u Server Exploreru. Objekti baza podataka su uskladišteni u fasciklama, prema njihovim tipovima. Za SQL Server i Oracle baze podataka videćete sledeće fascikle ispod Connection objekta baze podataka:

- Database Diagrams (dijagrami baza podataka)
- Tables (tabele)
- Views (pregledi)
- Stored Procedures (snimljene procedure)
- Functions (funkcije)

NAPOMENA

Svi objekti ispod Data Connection u Server Exploreru su uskladišteni u bazi podataka i nisu svojstveni za projekat. Čak i ako uklonite Data Connection iz ovog prozora, sve promene koje ste izvršili na bazi podataka pomoću te veze neće biti izgubljene. ■

Dijagrami baza podataka

Ova fascikla čuva relacione dijagrame baze podataka. Kao što znate, relacioni dijagram definiše kako se različite tabele odnose jedna prema drugoj. Da biste definisali novu relaciju, kliknite desnim tasterom miša iznad ikone Database Diagrams i izaberite New Diagram. Možete dodati tabelu koju želite svom dijagramu. Da biste editovali postojeći dijagram, samo duplo kliknite iznad njega.

Možete da imate više dijagrama baze podataka da biste grupisali tabele i njihove relacije. Ako baza podataka sadrži mnogo tabela, možete kreirati različite dijagrame za svaku grupu međusobno povezanih tabela. Individualne tabele mogu da pripadaju i većem broju dijagrama, sve dok su relacije u svim dijagramima konzistentne.

Ako proširite čvor za svoj dijagram, videćete potčvorove za sve tabele predstavljene u dijagramu. Nastavite proširivanje da biste videli kolone tabela, ili da biste izabrali da dizajnirate tabelu selekovanjem Design Table opcije iz iskaćućeg menija.

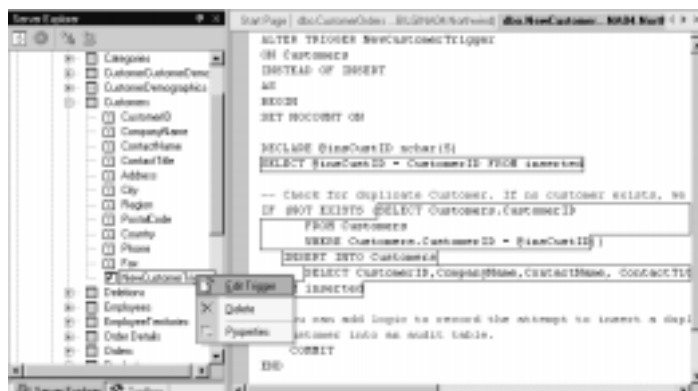
UPOZORENJE

Morate koristiti VS.NET Enterprise Architect izdanje da biste koristili ove funkcije za editovanje baza podataka. ■

Tabele

Ova fascikla sadrži sve tabele baze podataka, izuzev što je filtrirana iza scene, tako da sistemske tabele nisu vidljive, za razliku od onoga što vidite sa Enterprise Managerom. Videćete u sledećem odeljku kako možete dodati nove tabele bazi podataka, ili editovati postojeće. Ako ste se povezali sa bazom podataka koja je upravo bila kreirana, ali još uvek nije dizajnirana, možete dodati sve tabele i njihove relacije iz Server Explorera. Da biste dizajnirali nove tabele, kliknite desnim tasterom miša iznad Tables ikone i izaberite New Table. Da biste editovali postojeću, kliknite desnim tasterom miša iznad njenog imena i izaberite Design Table. Možete, takođe, duplo kliknuti tabelu i videti njene redove podataka u mreži. Možete, čak, editovati redove tabele na ovoj mreži, direktno iz VB.NET okruženja.

Osim toga, koristite Tables fasciklu za pristupanje pokretačkim događajima baze podataka. Pokretački događaji su specifični za tabele i oni su izlistani pod tabelom kojoj pripadaju, kao što je prikazano na slici 3.14. Ako proširite tabelu u Server Explorer prozoru, videćete njene kolone. Ako tabela sadrži pokretačke događaje, njihova imena će se takođe pojaviti posle liste polja tabele. Da biste dizajnirali novi pokretački događaj, kliknite desnim tasterom miša iznad Tables ikone i izaberite New Trigger. Da biste editovali postojeći pokretački događaj, kliknite desnim tasterom miša iznad željenog pokretačkog događaja i izaberite Edit Trigger opciju, takođe prikazanu na slici 3.14.



SLIKA 3.14 Rad sa tabelama i pokretačkim događajima u Server Exploreru

Pregledi

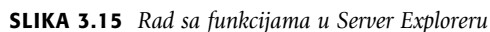
Ova fascikla sadrži preglede već pridodate bazi podataka. Ako proširite pregled, videćete njegove kolone. Da biste dizajnirali novi pregled, kliknite desnim tasterom miša iznad Views ikone i izaberite New View. Da biste editovali postojeći, kliknite desnim tasterom miša iznad njega i izaberite Design View. Ovo pokreće Query Designer interfejs, koji obrađujemo detaljno kasnije u ovom poglavlju, u odeljku "Korišćenje Query Designera". Možete, takođe, otvoriti prozor duplim klikom na njegovo ime; videćete njegove redove na mreži. Ako je pregled moguće ažurirati, možete editovati njegove redove (ili one koji su dozvoljeni za editovanje) direktno iz VB.NET okruženja.

Snimljene procedure

Ova fascikla sadrži snimljene procedure već pridodate bazi podataka. Ako kliknete plus znak ispred imena snimljene procedure, stavka će se proširiti da bi pokazala ulazne i izlazne parametre snimljene procedure.

Funkcije

Ova fascikla pokazuje korisnički definisane funkcije koje postoje u bazi podataka, kao što možete videti na slici 3.15. Ako proširite čvor za određenu funkciju, videćete njene ulazne i izlazne parametre.



U Poglavlju 5 ćete detaljnije istražiti tipove korisnički definisanih funkcija. Međutim, ovo je dobro mesto za objašnjenje zašto se neke od ikona funkcija pojavljuju drugačije od drugih, kao što možete videti na slici 3.15.

110

Korišćenje Database Designera

Ovaj deo obrađuje Database Designer i koristi Northwind bazu podataka za primere. Umesto dizajniranja novih tabela, editovaćete tabele Northwind baze podataka, pregledati postojeća ograničenja i relacije i dodati nove. Koristićete vezu koju ste ustanovili ranije u Northwindu. U prozorima Database Designera možete dizajnirati SQL Server i Oracle baze podataka. Ne postoji zaseban Database Designer prozor u vizuelnim alatima baze podataka. Postoji pet alata za dizajniranje objekata baze podataka:

- Design Table prozor, gde možete editovati atribut tabele
- Table Property Pages okvir za dijalog, gde možete pregledati i editovati svojstva tabele, kao što su indeksi, ograničenja i relacije
- Database Diagram prozor, gde možete pregledati i editovati relacije između tabela baze podataka
- Query Designer prozor, koji omogućava da vizuelno kreirate SQL iskaze za preglede i snimljene procedure
- SQL Editor prozor, gde možete editovati sirov SQL kod, slično kao što biste editovali i VB.NET kod.

Design Table prozor

Otvorite Server Explorer prozor, kreirajte Northwind Data Connection (ako to već niste uradili), pa proširite Data Connection i fascikle ispod nje. Da biste otvorili Products tabelu u dizajn modu, pronađite je u Tables fascikli, kliknite desnim tasterom miša iznad njene ikone i izaberite Design Table. Na slici 3.16 prikazana je Products tabela u dizajn modu.



SLIKA 3.16 Products tabela u dizajn modu

Svaka kolona tabele ima ime i tip podatka, a svaki tip podatka ima dužinu. Dužina većine tipova podataka je fiksna i ne možete je promeniti. Celi brojevi se čuvaju u četiri bajta, kolone datuma i vremena u osam bajtova i tako dalje. Tipovi podataka `varchar` i `nvarchar` su stringovi promenljive dužine, a njihovu maksimalnu dužinu definišete u `Length` koloni. Isto važi za binarne kolone. Neki numerički tipovi imaju `Precision` i `Scale` atribut. `Precision` atribut je broj cifara koji se koristi za predstavljanje numeričkog tipa podatka - ukupan broj cifara u broju. Atribut `Scale` je broj cifara iza decimalnog zareza.

NAPOMENA

Tipovi podataka su obrađeni u Poglavlju 2, "Osnovni koncepti relacionih baza podataka", gde je posebna pažnja posvećena novim SQL Server 2000 tipovima podataka. ■

`Allow Nulls` kolona mora biti poništena za polja koja ne mogu biti `Null`. Primarna ključna polja, na primer, ne mogu biti `Null`. Pre nego što definišete primarni ključ tabele, morate izbrisati njegov atribut `Allow Nulls`. U zavisnosti od podataka koje nameravate da skladištite u tabeli, drugim poljima koja nisu ključna može da ne bude dozvoljeno da prihvate `Null` vrednosti. Datum porudžbine, na primer, obično ne može da bude `Null`.

Da biste podesili primarno polje tabele (ako postoji), kliknite desnim tasterom miša iznad sivog boksa ispred imena kolone i izaberite `Set Primary Key` iz menija. Polje primarnog ključa tabele je identifikovano ikonom ključa.

Ispod tabele videćete detalje za određenu kolonu koju ste izabrali. Postoji mnogo različitih atributa koje možete izabrati za kolonu. U polju `Default Value` možete definisati podrazumevanu vrednost kolone, koja se automatski pridružuje svakom novom redu ako ni jedna druga vrednost nije uneta. Podrazumevana vrednost može da bude vrednost istog tipa kao i kolona, ili funkcija koja vraća isti tip podatka. Podrazumevana vrednost za `OrderDate` polje u `Orders` tabeli može da bude sledeći izraz koji vraća aktuelni datum i vreme:

```
GetDate()
```

Druga polja na `Columns` kartici uključuju `Identity`, `Identity Seed`, `Identity Increment` i `IsRowGuid`. Polja primarnog ključa su često podešena na vrednosti koje nisu celi brojevi i omogućavaju bazi podataka da dodeli jedinstvenu celobrojnu vrednost polju ključa svakog reda dodatog tabeli. Da biste definisali da SQL Server treba da automatski pridoda vrednosti polju, definišite `Yes` u padajućoj listi za `Identity` polje. Kada podesite `Identity` polje, možete dodatno da definišete inicijalnu vrednost (`Identity Seed`) kao i inkrement (`Identity Increment`). Sledeći boks (`IsRowGuid`) treba da bude podešen na `Yes` ako je određena kolona globalni identifikator, vrednost koja je jedinstvena ne samo u kontekstu tabele, već u čitavoj bazi podataka.

Ovako se dizajniraju tabele. Za sada, to je sasvim jednostavno, gotovo intuitivno. Međutim, tabele su više od kolekcija kolona. Da biste osigurali integritet baze podataka, treba da ustanovite ograničenja na vrednostima različitih kolona, definišete indekse i polja ključa i, na kraju, kreirate relacije između primarnog i polja stranih ključeva. Ovo možete uraditi korišćenjem `Property Pages` okvira za dijalog za tabelu.

Okvir za dijalog Table Property Pages

Da biste videli svojstva tabele, kliknite desnim tasterom miša negde na prozoru za dizajniranje tabele i izaberite Property Pages iz menija. Property Pages okvir za dijalog ima četiri kartice, koje ćemo sada objasniti.

SAVET

Da biste videli Properties Pages tabele, morate prvo da otvorite tabelu u dizajn modu, pa da kliknete desnim tasterom miša iznad prozora za dizajniranje tabele. Ako kliknete desnim tasterom miša iznad imena tabele u Server Exploreru i izaberete Properties, videćete okvir za dijalog sa imenom tabele i ime njenog vlasnika, a ne Table Properties prozor. ■

Tables kartica

Tables kartica, prikazana na slici 3.17, omogućava da modifikujete atribut izabrane tabele i promenite njihovo ime ukucavanjem imena u Table Name boks. Možete definisati vlasnika tabele korišćenjem padajuće liste. Primarni ključ tabele se često pojavljuje u Table Identity Column padajućem polju. Ako tabela ima globalni jedinstveni identifikator, on će biti na listi u Table ROWGUID Column polju. Sledeća dva polja će prikazati vrednost PRIMARY, ako administrator baze podataka nije podelio bazu podataka u više fajlova. Ostavite ove bokseve administratoru baze podataka. Ukratko, SQL Server baza podataka može da bude uskladištena u više fajlova, ali ovo neće uticati na Vaš kod. Konceptualni pogled na bazu podataka ostaje isti, bez obzira da li su te tabele skladištene u jednoj, ili više tabela. Na kraju, možete definisati opis tabele u poslednjem polju.



SLIKA 3.17 Koristite Tables karticu Property Pages okvira za dijalog tabele za definisanje ograničenja.

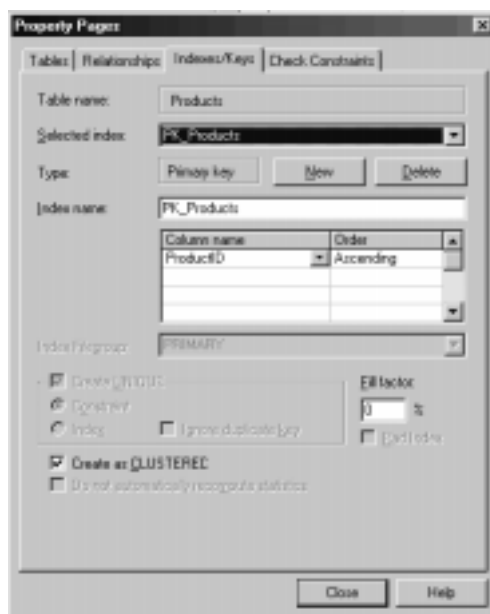
Relationships kartica

Relacije su najznačajniji aspekt u dizajniranju relacionih baza podataka. Ne morate samo da definišete tačne relacije za spojeve koje ćete kasnije izvršavati u upitima, već morate i ustanoviti pravila za integritet referenci između tabela. Ova kartica je objašnjena kasnije u ovom poglavlju, u odeljku "Korišćenje Database Diagrama".

Indexes/Keys kartica

Koristite Indexes/Keys karticu, prikazanu na slici 3.18, za manipulisanje indeksima i ključevima svojih tabela. Svaka tabela je indeksirana po jednoj, ili više kolona i obično ima jedan primarni ključ. Iz boksata Selected Index možete izabrati jedan od indeksa izabrane tabele. Da biste kreirali novi indeks, kliknite New dugme. Možete izbrisati postojeći indeks klikom na Delete dugme.

Svaki indeks ima ime, definiciju i nekoliko atributa, koji su definisani u donjem delu kartice. Definicija indeksa je ime jedne, ili više kolona. Ako pretpostavite da će Vaše aplikacije tražiti proizvode po njihovom drugom imenu, indeksirajte Customers tabelu po LastName polju. Kada indeksirate tabelu po polju, razlikujete porudžbinu prema polju uključenom u indeksu. Redove tabele nije obavezno preurediti, već se oni pronalaze po redu koji je definisan indeksom. Možete takođe indeksirati tabelu po dva, ili više polja. Ako polje po kome indeksirate nije jedinstveno, na primer, drugo ime, definišite dodatne ključeve. Moguće je indeksirati tabelu po poljima drugog i prvog imena, tako da su redovi sa istim drugim imenom grupisani zajedno i poredani prema prvom imenu.



SLIKA 3.18 Koristite Indexes/Keys karticu Property Pages okvira za dijalog tabele za definisanje novih polja indeksa i ključeva.

Polje za potvrdu Create UNIQUE omogućava da se definiše da je indeks jedinstven. Indeks po polju LastName, na primer, ne može biti jedinstven. Sasvim je moguće za dva, ili više kontakata u Customers tabeli da imaju isto drugo ime. Indeks zasnovan na polju, kao što je ID proizvoda, ili ISBN knjige, je jedinstven. Ovo polje se koristi kao primarni ključ, a primarni ključevi su po definiciji jedinstveni. Postoje situacije, međutim, kada polje ne može biti korišćeno kao primarni ključ, a mora biti jedinstveno. Recimo da održavate tabelu sa ID-ima učitelja i brojevima učionica. Zbog toga što učitelj ne može da bude u dve učionice u isto vreme, možete definisati da je kombinacija TeacherID + RoomNumber jedinstvena. Ako ne želite da kreirate indeks po ovim dvema kolonama, proverite Constraint opciju. Ako želite da pretražujete tabelu sa kombinacijom polja TeacherID + RoomNumber (ili samo sa poljem TeacherID), kreirajte novi indeks potvrđivanjem Index opcije.

Polje za potvrdu Create As CLUSTERED omogućava da definišete da će indeks biti kreiran kao grupisani indeks. Svaka tabela može da ima samo jedan grupisani indeks - on mora biti zasnovan na polju koje se koristi najčešće pri pretraživanju. Grupisani indeksi su vrlo efikasni, jer SQL Server čuva redove po istom redosledu kao što diktira grupisani indeks. Drugačije rečeno, fizički raspored redova je isti kao i logički raspored ključnih vrednosti. Rezultat ovoga je da SQL Server, kada pretražuje B-Tree strukturu indeksa, locira stvarne podatke, a ne pokazivače na podatke. Da biste održavali grupisane indekse, SQL Server radi malo teže svaki put kada se doda novi red, ali operacije koje uključuju grupisane indekse se izvršavaju vrlo efikasno.

Check Constraints kartica

U donjem delu Check Constraints kartice možete definisati bilo koji broj ograničenja za izabranu tabelu. Svako ograničenje je identifikovano imenom (tako da možete da ga uklonite i kasnije ponovo uspostavite) i definicijom. Da biste dodali novo ograničenje, kliknite New dugme, ukucajte ime ograničenja i ukucajte njegovu definiciju u odgovarajući tekst boks. Definicija ograničenja je izraz. Većina ograničenja su često upoređivanja, na primer:

```
Discount > 0 And Discount < 1
```

Ovaj izraz koji koristi relacione i logičke operatore "kaže" SQL Serveru da popust treba da ima vrednost između 0 i 1. Da biste definisali da polje Unit Price bude pozitivno, koristite sledeće ograničenje:

```
[Unit Price] >= 0
```

Tabela Employees sadrži malo kompleksnije ograničenje, koje zahteva da je datum rođenja zaposlenih "manji" (da se desio ranije) od aktuelnog datuma:

```
([BirthDate] < getdate())
```

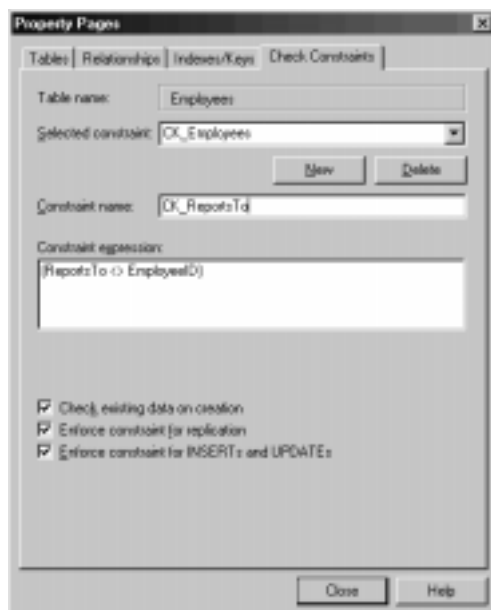
NAPOMENA

Ovo je primer iz knjige. Ograničenje kao što je ovo uopšte ne štiti Vaše podatke. ■

Možda niste primetili, ali Employees tabela je malo neobična: ona referencira samu sebe. Sadrži polje nazvano ReportsTo, koje predstavlja ID drugog zaposlenog. U većini korporacija zaposleni ne podnose izveštaje jedni drugima, tako da bi ograničenje sa više smisla za Employees tabelu bilo ono koje štiti ReportsTo polje da ne bude isto kao i EmployeeID polje. Možete li da zamislite šta bi se desilo da ste kreirali hijerarhiju zaposlenih zasnovanu na tome ko kome podnosi izveštaj

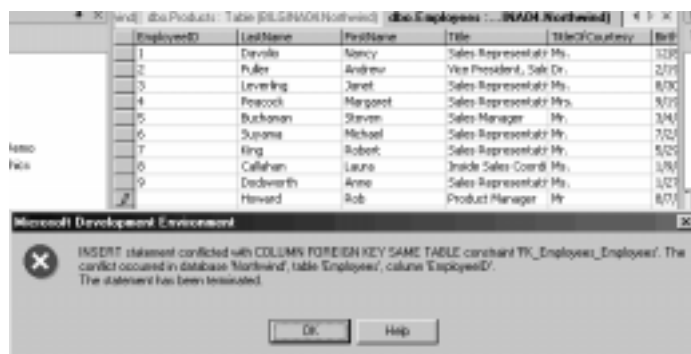
i da jedan od zaposlenih podnosi izveštaj samom sebi? Da biste sprečili ovu nepoželjnu situaciju, možete dodati novo ograničenje sa sledećom definicijom (takode prikazanom na slici 3.19):

ReportsTo <> EmployeeID)



SLIKA 3.19 Dodavanje novog ograničenja

Imenujte ovo ograničenje sa CK_ReportsTo i testirajte ga. Najpre, morate zatvoriti Employee mrežu za dizajniranje i složiti se da sačuvate svoje promene. Onda otvorite istu tabelu za editovanje duplim klikom iznad njenog imena. Ako pokušate da napravite ReportsTo polje jednako sa EmployeeID istog reda, ažuriranje će biti odbačeno, kao što je prikazano na slici 3.20.



SLIKA 3.20 Ova poruka o grešci će se pojaviti ako unesete vrednost koja ugrožava CK_ReportsTo ograničenje.

Korišćenje Database Diagram prozora

Da biste pregledali i editovali relacije u bazi podataka, pređite na Server Explorer i proširite Database Diagram fasciklu. Da biste pregledali relacije u bazi podataka, morate imati postojeći dijagram. Ako nema relacionog dijagrama za bazu podataka, kliknite desnim tasterom miša iznad Database Diagrams ikone i izaberite New Diagram.

Database Diagram prozor se pojavljuje na ekranu. Sada spustite tabele koje želite da uključite u dijagram iz Data View prozora na Database Diagram prozor. Tabele su predstavljene boksevima koji sadrže imena polja svake tabele. Polja primarnog ključa su označena ikonicom ključa ispred imena, kao što je prikazano na slici 3.21.



SLIKA 3.21 Customer-Order relacioni dijagram baze podataka Northwind baze podataka

Svaka relacija je predstavljena strelicom koja povezuje dve tabele u relaciji. Normalizovana baza podataka ima dva tipa relacija:

- jedan-prema-više (ili više-prema-jedan)
- jedan-prema-jedan

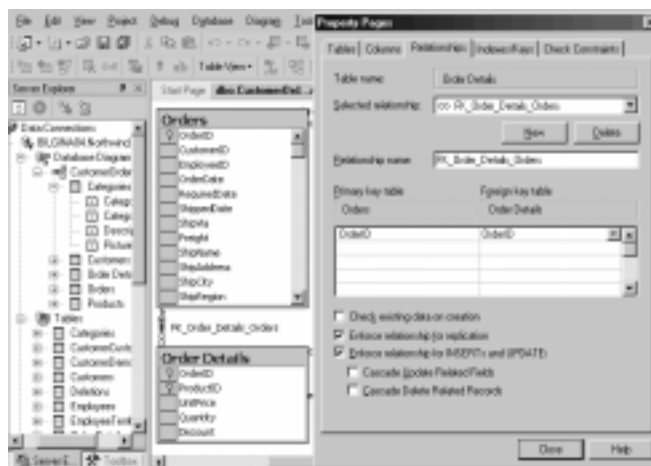
NAPOMENA

Postoje i više-prema-više relacije, ali se obično definišu korišćenjem dodatnih tabela za povezivanje. ■

Na jedinstvenoj strani relacije postoji ikona ključa, jer imate polje ključa. Na drugoj strani obično imate simbol beskonačnosti. Strana "više" relacije je strani ključ. Relacija između Customers i Orders je tipična jedan-prema-više relacija: svaki klijent se može pojaviti u više redova. Jedan-prema-jedan relacije nisu mnogo česte.

Ako zadržite pokazivač iznad relacije, videćete njeno ime i imena dve tabele koje povezuje. Da biste videli nazive relacija, kliknite desnim tasterom miša negde iznad panoa i izaberite Show Relationship Labels komandu iz menija.

Da biste videli više informacija o relaciji, kliknite desnim tasterom miša iznad linije i izaberite Property Pages. Videćete Table Property Pages okvir za dijalog sa Tables, Relationships, Check Constraints i Indexes/Keys karticama, koje smo već obradili. Relationships kartica je prikazana na slici 3.22 (ona slikovito pokazuje relaciju između tabela Orders i Order Details).



SLIKA 3.22 Relationships kartica Properties okvira za dijalog

Na kartici Relationships videćete ime izabrane relacije, dva polja uključena u relaciju (primarni i strani ključ) i tri opcije za uspostavljanje integriteta relacija na dnu. Opcije izgledaju ovako:

- Polje za potvrdu Check Existing Data On Creation “kaže” SQL Serveru da proveriti da li postojeći redovi ugrožavaju relaciju. Ako par primarni/strani ključ ugrožava relaciju (strani ključ u redu pokazuje na nepostojeći primarni ključ), biće prikazano upozorenje i relacija neće biti dodata u bazu podataka. Morate ručno da korigujete ove redove, pa da pokušate da uspostavite relaciju ponovo.
- Polje za potvrdu Enforce Relationship For Replication “kaže” SQL Serveru da primeni ograničenje kada se strana tabela replikuje u drugu bazu podataka.
- Treće polje za potvrdu Enforce Relationship For INSERTs And UPDATEs “kaže” SQL Serveru da prekine bilo koju operaciju ubacivanja, ili ažuriranja koja ugrožava integritet relacije. Ako potvrdite ovo polje, dve dodatne opcije će biti omogućene: Cascade Update Related Fields i Cascade Delete Related Records opcije. One omogućavaju da uspostavite kaskadna ažuriranja i brisanja sa povezanim tabelama, ako se bilo koja od ovih kolona promeni.

NAPOMENA

Osobina kaskadnog brisanja i ažuriranja je nova u SQL Serveru 2000. ■

Da biste ustanovili novu relaciju između dve tabele, prevucite jedno od dva polja koja su uključena u relaciju i spustite ga na drugo polje iz Database Designer prozora. Odmah pošto spustite polje, pojaviće se Create Relationship prozor. Database Designer je dovoljno "inteligentan" da "shvati" koje polje je primarno, a koje je strano. U većini slučajeva sve što treba da promenite je ime relacije.

SAVET

Kada vršite promene u okviru Database Diagram prozora za dizajniranje, pored imena tabela pojaviće se zvezdica čiji atributi su promenjeni. Kada zatvorite Database Diagram prozor, od Vas se traži da sačuvate promene u ovim tabelama. Ako izaberete da sačuvate ove promene, zvezdica se neće više pojaviti kada ponovo otvorite dijagram. ■

Korišćenje Query Designera

Najvažnija i najčešća operacija koju ćete izvršiti na bazi podataka je upit. Da biste postavljali upit bazi podataka, morate pisati komande u specijalnom jeziku Structured Query Languageu (SQL-u) i izvršiti ih nad bazom podataka. U ovom odeljku gradićete SQL iskaze korišćenjem drugog vizuelnog alata Query Designera. Query Designer je odličan alat, ali, kao programer, verovatno nećete želeti da se oslonite samo na vizuelne alate; na kraju, ovladaćete SQL-om tako da jednostavno možete da kucate svoje upite.

Sa Query Designerom, možete koristiti vizuelni dijagram upita da biste povlačili tabele u upit i definisali koje polje će se pojaviti u izlazu upita postavljanjem pokazivača i klikom. Pored toga, možete limitirati broj redova koje vraća upit definisanjem kriterijuma pretraživanja na mreži. Query Designer će generisati SQL iskaze koji implementiraju Vaš upit i prikazuju ga na odvojenom panou. Na kraju, možete izvršiti upit da biste proverili da li radi kao što je planirano. Upit će biti izvršen nad bazom podataka, tako da treba da proverite da li izvršavate *akcione upite* (upite koji modifikuju bazu podataka) nad produkcionom bazom podataka. Ako promenite bilo koju od baza podataka koje su date kao primeri izvršavanjem akcionih upita nad njima, možda nećete dobiti iste rezultate kao u primerima u sledećim poglavljima. Sasvim je sigurno, međutim, da izvršite *upite izbora* koji pronalaze redove i prikazuju ih u donjem panou prozora Query Designera. Mnogi programeri jednostavno unose SQL iskaze u srednji pano Query Designera i izvršavaju ih; ne morate da definišete iskaze vizuelno. Interesantna osobina Query Designera je da on može da napravi vizuelnu reprezentaciju SQL iskaza, a Vi možete da podesite upit pomoću vizuelnih alata.

Query Designer prozor

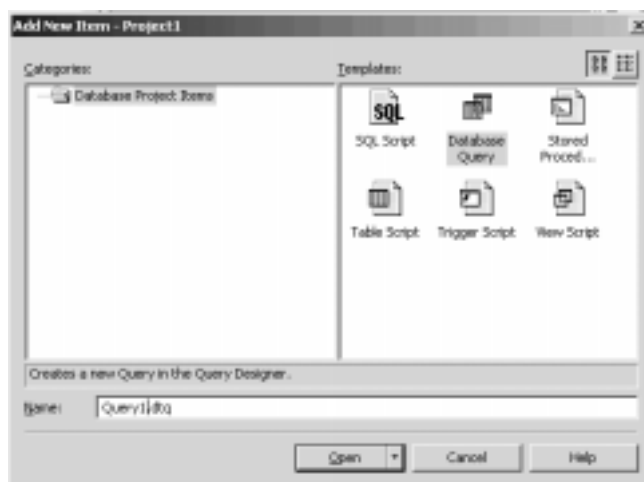
Da biste pokrenuli Query Designer, ponovo otvorite Database Project koji ste kreirali ranije. U Solution Exploreru kliknite desnim tasterom miša iznad *Queries* fascikle i izaberite Add Query. Ovo prikazuje Add New Item okvir za dijalog (vidite ga na slici 3.23), koji omogućava da birate iz različitih Database Project stavki, kao što su skriptovi, snimljene procedure i upiti. Izaberite Database Query Item.

RAZLIČITI NAČINI ZA POKRETANJE QUERY DESIGNERA

Postoji mnogo različitih načina za pokretanje Query Designera (na kraju, završavate sa istim funkcionalnostima). U ovom poglavlju pokazujemo kako da ga koristite iz Database Projecta. Alternativa je da ga možete pokrenuti editovanjem dizajna pregleda. Da biste to uradili, proširite Views čvor iz Server Explorera i kliknete desnim tasterom miša iznad postojećeg reda. Izaberite Design View opciju iz iskaćućeg menija, koji će da pokrene Query Designer. Izbor New View opcije će takođe da pokrene Query Designer.

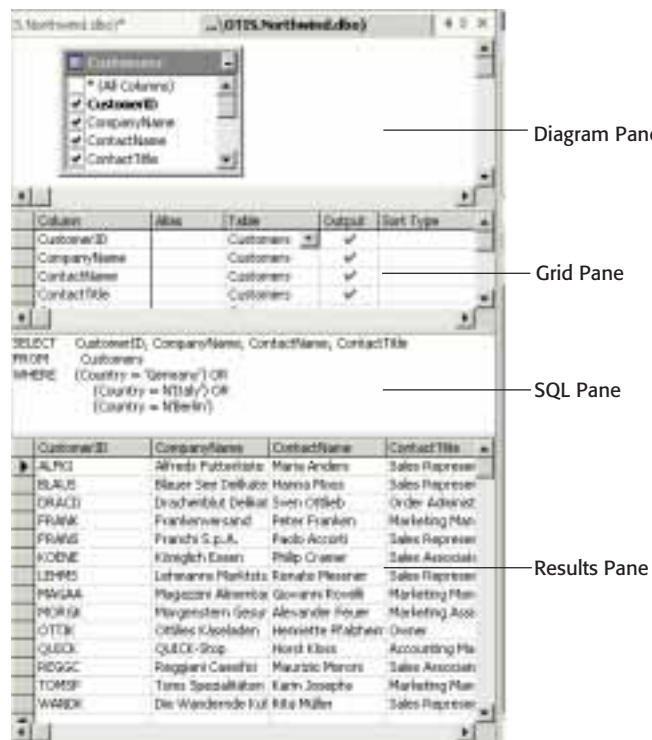
Drugi način za učitavanje Query Designer interfejsa je rad sa *SQL blokovima*. Ovo su definisani regioni SQL koda koji može biti editovan korišćenjem vizuelnog Query Designera. Učičete više o tome kasnije u ovom poglavlju, u odeljku "SQL Editor".

Možete, takođe, da pokrenete Query Designer iz unapred konfigurisane DataAdapter kontrole. Pokrenite prozor svojstva DataAdapter kontrole, ako nije već predstavljen, i proširite Select Command svojstvo. Ispod tog svojstva videćete CommandText svojstvo, koje prikazuje SQL iskaz povezan sa Command objektom. Ako kliknete elipse pored SQL iskaza, lansiraćete Query Designer prozor. Isto važi i za druge Command objekte, kao što su InsertCommand, DeleteCommand i UpdateCommand (obradićemo DataAdapter kontrole u Poglavlju 6, "Prvi pogled na ADO.NET").



SLIKA 3.23 Okvir za dijalog Add New Item

Sledeći okvir za dijalog traži od Vas da izaberete neke tabele, preglede, ili funkcije. Izaberite Customers tabelu. Kliknite Add dugme da biste prikazali tabele u Diagram panou. Query Designer prozor koji se otvori ima četiri panoa, koja su prikazana na slici 3.24.



SLIKA 3.24 Pano Query Designer prozora

Diagram pano

Na panou dijagrama prikazujete tabele uključene u upit. One su predstavljene boksevima, koji izlistavaju kolone tabela. Ako želite da dodate još tabela, kliknite desnim tasterom miša iznad ovog panoa i izaberite Add Table. Odmah pošto dodate novu tabelu, ona se automatski povezuje sa svojim povezanim tabelama na dijagramu. Podrazumeva se da Query Designer kreira unutrašnje spojeve (tipovi spojeva su objašnjeni u Poglavlju 4, ali je ovo važan detalj). Videćete uskoro različite tipove spojeva i kako da promenite podrazumevani tip spoja.

Grid pano

Grid pano sadrži mrežu sličnu kao kod programa za unakrsno izračunavanje, u koju definišete kriterijum za ograničavanje broja redova i kolona koje treba uključiti u upit (na primer, porudžbine napravljene u određenom intervalu, knjige koje je izdao određeni izdavač i tako dalje). Ako želite da proračunavate zbrove (zbrove, brojanja, proseke), definišete ih na Grid panou.

SQL pano

SQL pano prikazuje SQL iskaz za aktuelni upit. SQL iskaz implementira upit koji ste definisali vizuelno na Diagram panou. Možete editovati SQL iskaz, ili uneti sopstveni SQL iskaz od početka. Kada to uradite, Query Designer postavlja tabele koje uključuju upit na panou Diagrama i, umesto Vas, ispunjava kriterijum izbora automatski.

Ako je SQL za Vas novina, verovatno ćete koristiti Diagram pano za vizuelno kreiranje upita, pa ćete gledati SQL iskaze koje je automatski generisao Query Designer. Ako Vam je SQL poznat, verovatno ćete uneti SQL iskaze ručno (ili možete koristiti Diagram pano da biste izbegli unošenje dugačke liste imena polja, a, zatim, editovali inicijalni iskaz).

Results pano

Pošto ste definisali SQL iskaz, ili na Diagram panou, ili na SQL panou, možete izvršiti upit i pregledati redove koje vrati. Da biste izvršili upit, koristite Run komandu iz Query menija (ili pritisnite Ctrl+R). Možete, takođe, kliknuti desnim tasterom miša iznad SQL, ili Diagram panoa i izabrati Run iz menija. Pored toga, možete obezbediti validnost SQL iskaza klikom desnim tasterom miša na SQL panou i izborom Verify SQL Syntax iz menija. Ovo ne garantuje validnost Vaših rezultata, ali možete proveriti svoje SQL iskaze, naročito akcione upite.

Možete prevlačiti kolone iz Diagram panoa u Grid pano. Ovo automatski “osvežava” SQL pano. Query Designer generiše odgovarajući SQL iskaz i prikazuje ga na SQL panou. Ovo funkcioniše na oba načina. Ako ste uneli SQL iskaz, Query Designer će ažurirati Diagram pano na odgovarajući način. Kada pokrenete upit, on će prikazati odgovarajuće redove na Results panou na dnu prozora.

UPOZORENJE

Ako upit promeni jedan, ili više redova tabele (akcioni upit), neće vraćati nikakve redove i ništa neće biti prikazano na Results mreži. Međutim, baza podataka će biti ažurirana. Ako je potrebno, testirajte svoje akcione upite na privremenim tabelama, ili, čak, na kopiji baze podataka. Većina upita su upiti izbora i mogu biti sigurno testirani. Ali, mnogi od nas su tako naviknuti da testiraju upite izbora u Query Designeru gde možemo pozvati Run komandu, a da ne shvatimo da smo pokrenuli akcioni upit. ■

Izgradnja jednostavnih upita

Ako Vam nije poznat SQL, morate izgraditi nekoliko upita Query Designerom da biste videli šta on može uraditi za Vas. Ovo će Vam pomoći da počnete i možete izabrati da koristite ovaj alat na regularnoj osnovi.

Najjednostavniji tip upita je onaj koji pronalazi broj polja iz izabranih redova u pojedinačnoj tabeli. Da biste pronašli imena preduzeća u Nemačkoj i njihove kontakte iz Northwind baze podataka, pratite sledeće korake:

1. Postavite Server Explorer i Query Designer prozore na ekranu da budu vidljivi. Onda prevucite tabelu koja Vam je potrebna u svoj upit iz Server Explorer prozora i spustite ih na Diagram pano Query Designera. Za ovaj primer će Vam biti potrebna samo Customer tabela.

2. Proverite imena polja koja želite da se pojave u izlazu upita. Proverite polja CustomerID, CompanyName, ContactName i ContactTitle. Imena polja će se pojaviti u Grid panou, gde morate definisati ograničenja upita - doslovno, da želite da izaberete samo redove u kojima polje Country ima vrednost "Germany".

3. Kliknite prvu praznu liniju u Grid panou i izaberite Country polje u padajućoj listi od svih polja koja će se pojaviti. Zatim se prebacite na Criteria kolonu i unesite ovaj izraz:

= 'Germany'

Ovo je jedini kriterijum koji ćete primeniti u upitu. Zbog toga što će svi klijenti biti iz Nemačke, ne želite da uključite Country polje u izlaz. Da biste isključili polje iz izlaza upita, obrišite odgovarajući boks u Output koloni.

Odmah pošto prebacite kursor u drugu ćeliju, SQL pano će biti ažuriran. SQL iskaz koji implementira upit koji ste definisali pomoću vizuelnih alata izgleda ovako:

```
SELECT CompanyName, ContactName, ContactTitle
FROM Customers
WHERE (Country = 'Germany')
```

4. Otvorite Query meni i izaberite Run da bi upit bio izvršen. Redovi koji odgovaraju upitu u Customers tabeli će se pojaviti na Results panou, kao što je prikazano ranije na slici 3.24.

Ako želite da definišete više kriterijuma za isto polje, unesite ih u kolone sa Or... zaglavljem. Svi kriterijumi će biti kombinovani sa OR operatorom. Da biste izabrali sve klijente iz Nemačke i Italije, na primer, unesite sledeći izraz u kolonu desno od Criteria kolone:

= 'Italy'

Ako želite da definišete kriterijum sa različitim kolonama, ponovite korak 3 sa različitim imenima polja. Da biste izabrali klijente iz Nemačke i Berlina, izaberite City polje u sledećem dostupnom redu Grid panoa i unesite sledeći izraz u Criteria kolona:

= 'Berlin'

Nemojte zaboraviti da obrišete Output boks polja City, jer ne želite da uključite ime grada u izlaz.

Da biste dobili sve klijente iz Berlina, ne morate da definišete ograničenje zemlje. Uopšteno rečeno, imena gradova nisu jedinstvena, tako da treba da definišete i zemlju.

NAPOMENA

Možete primetiti da Grid pano prefiksuje Vaš kriterijum pretrage sa N pre vrednosti, kao što je: =N'Italy'. Ovo se pojavljuje, jer Grid pano konvertuje Vaše vrednosti u Unicode. Ovo se neće desiti ako direktno kucate svoj kriterijum na SQL pano. ■

Izgradnja upita sa agregatnim funkcijama

U ovom odeljku uključićete agregatne funkcije u SQL iskaze korišćenjem vizuelnih alata Query Designera. Međutim, prvo ukratko pogledajte agregatne funkcije.

SQL može da proračunava agregatne funkcije na izabranim poljima. Agregatna funkcija broji redove, proračunava zbirove i proseke i izvršava još nekoliko uobičajenih matematičkih operacija koje se koriste za zbrajanje podataka. SQL podržava sledeće agregatne funkcije:

AVG() Vraća prosek vrednosti u koloni.

COUNT() Vraća broj vrednosti u koloni.

MAX() Vraća najvišu vrednost u koloni.

MIN() Vraća najnižu vrednost u koloni.

SUM() Vraća zbir vrednosti u koloni.

NAPOMENA

U Poglavlju 4 pokazano je kako se ove funkcije koriste u SQL iskazima. ■

Jednostavne agregatne funkcije

Jednostavni agregatni upiti zajedno prezentuju proračunate informacije, kao što su zbrovi i proseci. Oni ne uključuju bilo kakve uslovne provere, ili kompleksne proračune. Kreirajte jednostavni agregatni upit za izvršavanje operacija brojanja.

Sada pogledajte svoj prvi upit, tako da, umesto svih klijenata u određenoj zemlji, on vrati broj klijenata u svakoj zemlji. Da biste zamenili `CompanyName` polje brojem klijenata u svakoj zemlji, resetujte svoj upit korišćenjem sledećih koraka:

1. Počnite novi upit, prevucite i spustite `Customers` tabelu na Diagram pano i proverite polja `CompanyName` i `Country`. Ovaj upit će vratiti sve klijente zajedno sa imenom njihove zemlje.
2. Kliknite desnim tasterom miša iznad Diagram panoa da biste otvorili iskaćući Query meni i potvrdite `Group By` opciju. Nova kolona sa zaglavljem `Group By` se ubacuje između `Sort Order` i `Criteria` kolona u Grid panou. Svim ćelijama u ovoj koloni se pridružuje vrednost `Group By`.
3. U redu `CompanyName` i ispod `Group By` kolone izaberite `Count` opciju (kliknite ovu ćeliju mišem i izaberite `Count` iz padajuće liste). Ako gledate pano dijagrama, videćete da se simbol za sumiranje pojavljuje pored `CompanyName` polja, a simbol za grupisanje pored `Country` polja.

Pošto ste završili, pogledajte odgovarajući SQL iskaz na SQL panou:

```
SELECT COUNT(CompanyName) AS Expr1, Country
FROM Northwind.dbo.Customers
GROUP BY Country
```

Ovaj iskaz "kaže" SQL Serveru da grupiše sve redove Customers tabele po Country polju, tako da su korisnici iz iste zemlje grupisani zajedno. Zatim, mora da proračuna redove u svakoj grupi. Rezultat upita je lista imena zemalja i broja klijenata, kao što je prikazano na sledećoj strani (samo prvih nekoliko zemalja su uključene u listu).

3	Argentina
2	Austria
2	Belgium
8	Brazil
3	Canada

Iskaz AS Expr1 je alias za prvu kolonu rezultata. Da biste prikazali ime kolone sa imenom koje ima više smisla, preimenujte ga da koristi alias, kao, na primer, "Customers", ili "Number of Customers".

I druge agregatne funkcije su lake za upotrebu. Kad bi polje CompanyName bilo numeričko, mogli biste da proračunavate prosek pomoću AVG() funkcije, ili total za sve redove u grupi pomoću SUM() funkcije.

Agregatne funkcije nad više tabela

Izgradite još jedan upit sa agregatnom funkcijom (da on uključuje dve tabele). Ovaj upit će pronaći imena kategorija i broj proizvoda u svakoj kategoriji. Kliknite desnim tasterom miša iznad Queries fascikle u svom Solution Exploreru i izaberite Add Query, birajući Database Query stavku. Da biste izgradili ovaj iskaz, potrebne su Vam Categories i Products tabele. Dodajte ih iz Add Table dijaloga, ili prevucite ih iz Server Explorer prozora na Diagram pano Query Designera. Možete pokrenuti AddTable dijalog desnim klikom na Diagram pano i izborom AddTable. Želite listu imena kategorija i broj proizvoda u svakoj kategoriji, tako da treba da proverite polje CategoryName u Categories tabeli. Ne želite da uključite ID kategorija u izlaznoj listi i nećete uključiti bilo koja polja druge tabele - samo ćete brojati proizvode.

Otvorite meni Query u VS.NET-u i proverite opciju Group By. Ova komanda će grupisati rezultate prema izabranim poljima u Categories tabeli. Odmah pošto potvrdite Group By opciju, simbol za grupisanje se pojavljuje pored imena CategoryName polja u Categories tabeli. Izaberite padajuću listu iz prve ćelije u drugom redu panoa sa mrežom. Videćete stavke Count i Count_Big pored imena polja. Ove funkcije vraćaju broj stavki u aktuelnoj grupi. Izaberite Count opciju. Biće potrebno da ponovo izaberete CategoryName polje. Da li brojite jedno polje, ili sva polja ne predstavlja nikakvu razliku. Ponovo otvorite Query meni, izaberite Run i posmatrajte rezultate upita na donjem panou Query Designera. Odgovarajući SQL iskaz se pojavljuje u SQL panou prozora, kao što je prikazano ovde:

```
SELECT      Categories.CategoryName, COUNT(*) AS Expr1
FROM        Categories
INNER JOIN  Products
ON          Categories.CategoryID = Products.CategoryID
GROUP BY   Categories.CategoryName
```

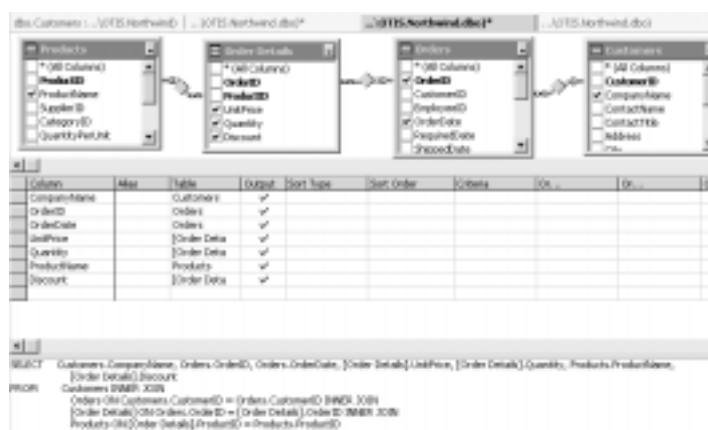
Iskaz **INNER JOIN** kombinuje redove iz dve tabele na osnovu zajedničkog polja (polja **CategoryID** u ovom primeru) i to je obrađeno u sledećem odeljku. Evo nekoliko linija izlaza koje proizvodi ovaj iskaz:

Beverages	12
Condiments	12
Confections	13

Izgradnja upita iz više tabela

Upiti su retko tako jednostavni. Veći deo njih uključuje više tabela povezanih međusobno i sa ograničenjima koja se primenjuju na polja, ili na više od jedne tabele. Izgradite upit koji pronalazi sve klijente, zajedno sa njihovim porudžbinama i proizvodima koje su kupili. Ovaj upit koristi četiri tabele: **Customers**, **Orders**, **Order Details** i **Products**; **Order Details** tabela sadrži ID-e proizvoda, a ne njihova stvarna imena. Specifikacija ovog upita zvuči komplikovano, ali izgradnja upita nije teška kada koristite vizuelne alate baza podataka. U ovom primeru videćete koliko brzo možete da generišete upit sa više tabela, izbegavajući mukotrpno kucanje.

Kliknite desnim tasterom miša iznad **Queries** fascikle u svom **Solution Exploreru** i izaberite **Add Query** iz menija. Izaberite **Database Query** stavku i dodajte četiri tabele izlistane gore na dijagramu. **Query Designer** ih automatski povezuje na osnovu postojećih relacija, kao što je prikazano na slici 3.25. On, takođe, priprema srž SQL iskaza i prikazuje ga na **SQL** panou. Primetićete da je lista izbora prazna. Počnite da potvrđujete imena polja koja želite da uključite u izlaz i gledajte kako lista izbora **SQL** iskaza raste. U tabeli 3.2 pokazana su imena polja koja je potrebno da izaberete.



SLIKA 3.25 Query Designer može da poveže tabele jednu sa drugom na osnovu definicije primarni/strani ključ u bazi podataka.

Tabela 3.2: Izabrana polja za upit

Tabela	Izabrana polja
Customers	CompanyName
Orders	OrderID, OrderDate
Order Details	UnitPrice, Quantity, Discount
Products	ProductName

Query Designer generiše dugačku listu izbora:

```
SELECT      Customers.CompanyName, Orders.OrderID, Orders.OrderDate,
           [Order Details].UnitPrice, [Order Details].Quantity, [Order Details].Discount,
           Products.ProductName
```

Ispred svih imena polja se nalazi prefiks sa imenom tabele kojoj pripadaju, sa tačkom između imena tabele i polja. Ključna reč **FROM** u SQL iskazu je dugačka lista unutrašnjih spojeva. Spojevi su obrađeni detaljno u Poglavlju 4, ali postoji kratko objašnjenje operacije spajanja: spoj kombinuje redove iz dve tabele na osnovu vrednosti polja koje je zajedničko za obe tabele (obično, kombinacija primarnog i stranog ključa). Redovi Customers i Orders tabela su spojeni ovako: SQL Server skenira sve redove Customers tabele. Svaki red Customers tabele se povezuje sa jednim, ili više redova Orders tabele, čije CustomerID polje ima istu vrednost kao CustomerID polje aktuelnog reda u Customers tabeli. Ako kupac sa ID-om "BLAUS" ima pet redova koji odgovaraju u Orders tabeli, biće dodato još pet novih redova izlazu. Ovih pet redova će imati isti CustomerID, ali različiti OrderID. Tabele se povezuju automatski jedna sa drugom u Diagram panou na osnovu relacija između tabela.

Query Designer takođe popunjava Grid pano poljima koja ste potvrdili, kao što je prikazano na slici 3.25. Sada kliknite desnim tasterom miša iznad Diagram panoa i izaberite Run iz menija da biste videli izlaz iz upita.

Specifikacija upita može da zvuči komplikovano, a SQL iskaz koji ga implementira je sasvim dugačak, ali možete da ga napravite jednostavnim postavljanjem četiri tabela na panou Diagrama i izborom imena polja koja treba uključiti u izlaz.

Korišćenje SQL Editora

Poslednji od vizuelnih alata baza podataka nije stvarno vizuelan. SQL Editor je tekstualni editor prilagođen za unošenje Transact-SQL iskaza; to je alat koji ćete koristiti za pisanje sopstvenih snimljenih procedura i pokretačkih događaja. Učićete kako da debugujete svoje snimljene procedure pomoću T-SQL Debuggera.

Snimljene procedure i pokretački događaji nisu jednostavna tema; u ovom odeljku su ukratko predstavljeni, a detaljno su obrađeni u Poglavlju 5. U ovom odeljku koristimo jednostavne primere da bismo pokazali šta T-SQL može da uradi za aplikaciju.

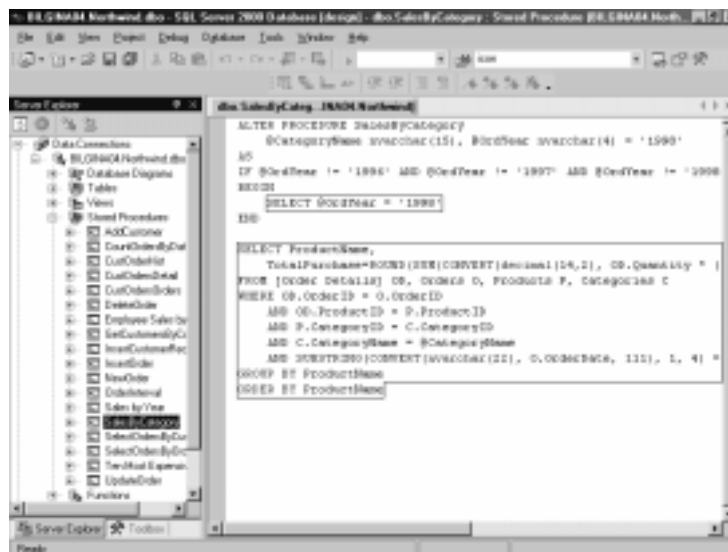
Rad sa snimljenim procedurama

Snimljena procedura je program u T-SQL-u koji postavlja upit, ili ažurira bazu podataka. U svojoj najjednostavnijoj formi snimljena procedura je upit koji je sačuvan u samoj bazi podataka. Ako često pozivate svoje upite, ili ako oni sadrže komplikovanu logiku, definitivno se isplati implementacija ovih snimljenih procedura i njihovo skladištenje u bazu podataka. SQL Server optimizuje snimljenu proceduru, što omogućava da bude izvršen brže nego ekvivalentan ad hoc SQL iskaz.

Da bismo demonstrirali kako da koristite unificirane VS.NET alate za debugovanje sopstvenih snimljenih procedura, koristićemo primere snimljenih procedura koje dolaze sa Northwind bazom podataka. Otvorite Stored Procedures fasciklu na Server Exploreru i duplo kliknite ikonu SalesByCategory snimljene procedure. Videćete Design prozor za snimljene procedure, kao što je prikazano na slici 3.26.

UPOZORENJE

Neke verzije VS.NET-a ne podržavaju editovanje snimljenih procedura iz Server Explorera. ■



SLIKA 3.26 Editovanje snimljenih procedura u Visual Basic razvojnom okruženju

Snimljene procedure su napisane u T-SQL-u, što je mešavina SQL-a i tradicionalnih struktura za programiranje, kao što su IF iskazi, petlje i tako dalje. Kao VB.NET programer, nećete imati problema prilikom učenja i korišćenja T-SQL-a.

UPOZORENJE

Ako debugujete udaljeni SQL Server, moraćete da podesite udaljeno SQL debugovanje, koje koristi DCOM. Konfigurisanje udaljenog SQL debugovanja korišćenjem DCOM-a je tema koja je izvan opsega ove knjige. Možete pronaći ovu informaciju u MSDN dokumentaciji koja se isporučuje sa VS.NET-om. ■

Snimljena procedura `SalesByCategory` prihvata ulazne parametre `@CategoryName` i `@OrdYear`. To su ime kategorije i godina u kojoj je zbir računat. Dve promenljive su deklarirane odmah posle deklaracije procedure. Primetićete da `@OrdYear` parametar ima podrazumevanu vrednost od 1998, koja se koristi ako pozivate snimljenu proceduru bez obezbeđivanja vrednosti za ovaj argument.

Prve tri linije pregledaju vrednost `@OrdYear` parametra. Ako nije u opsegu od 1996 do 1998, procedura ignoriše originalnu vrednost (što je greška) i proračunava zbir definisane kategorije za 1998:

```
IF @OrdYear != '1996' AND @OrdYear != '1997' AND @OrdYear != '1998'
BEGIN
    SELECT @OrdYear = '1998'
END
```

Iako imate još da istražujete T-SQL sintaksu, sve što treba da znate radi ovog vežbanja je da sva T-SQL imena promenljivih počinju znakom `@`.

Ostatak koda je SQL iskaz koji kombinuje `Products`, `Categories`, `Orders` i `Order Detail` tabele da bi proračunao ukupnu prodaju za određenu kategoriju u godišnjem periodu.

NAPOMENA

Možda ste primetili plave bokseve oko određenih segmenata SQL koda. Oni se nazivaju SQL blokovi. Ako kliknete desnim tasterom miša bilo gde u SQL Blocku i izaberete `Design Sql Block`, možete pokrenuti Query Designer da biste vizuelno modifikovali SQL kod. ■

Izvršavanje i debugovanje snimljene procedure

Sada ste spremni da testirate SQL iskaz pomoću Query Designera. Kliknite desnim tasterom miša bilo gde iznad koda snimljene procedure u okviru Designer prozora i izaberite `Step Into Stored Procedure` opciju da biste izvršili `SalesByCategory` snimljenu proceduru. Pre nego što se izvrši snimljena procedura, biće zatraženo da unesete vrednosti očekivanih parametara u `Run Stored Procedure` okviru za dijalog prikazanom na slici 3.27. Unesite željene vrednosti u poslednju kolonu i kliknite OK za nastavak.

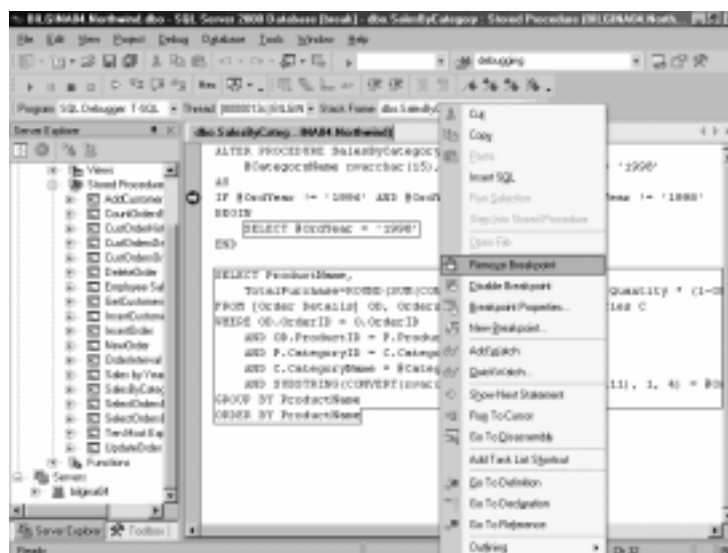


SLIKA 3.27 Obezbedite parametre snimljene procedure u prozoru *Parameters*.

Posle klika OK dugmeta na prozoru `Run Stored Procedure`, videćete žutu strelicu na prvoj liniji izvršnog koda. To je upravo ono što biste videli kada bi to bio VB.NET kod! Možete, čak, koristiti iste skraćenice sa tastature: F5 za pokretanje i F11 za kretanje kroz kod. VS.NET debager omogućava

OGRANIČENJA PRI DEBAGOVANJU U VS.NET-U

Čak i ako podesite ovu opciju, editovanje izvornog koda u debug modu je još uvek teško. U prošlim verzijama Visual Studia možda ste uživali u osobini koja je omogućavala da modifikujete linije koda za vreme debugovanja. Ovo je bilo korisno kada ste želeli da eksperimentišete sa promenom u logici koda dok je aplikacija izvršavana. Na taj način, niste morali da zaustavite i ponovo pokrećete aplikaciju da bi izvršavala novu logiku. Nažalost, Edit And Continue opcija u VS.NET-u ne radi na isti način. Ako uključite opciju Allow Me To Edit VB Files While Debugging, ne znači da će Vaš novi kod biti izvršen. U stvari, neće. Editovanja, dodavanja, ili brisanja u kodu dok ste u modu za debugovanje neće se dinamički kompajlirati u ostatak Vašeg izvornog koda. Razlog je činjenica da VS.NET kompajler ne interpretira više kod za vreme izvršavanja. Kada pokrenete svoju aplikaciju za vreme dizajniranja, Vaš kod je potpuno kompajliran. Možete debugovati kod, jer se VS.NET debugger "kači" na izvorni kod, tako da možete pauzirati izvršavanje i testirati vrednost u Immediate prozoru. Međutim, stvarni izvorni kod se ne kompajlira ponovo kada izvršite promene u debug modu, zato što je Allow Me To Edit VB Files While Debugging podrazumevano isključeno. Ovo stvarno znači sledeće: Dozvolite mi da editujem VB fajlove dok debugujem, ali nastavite sa originalnim iskompajliranim izvornim kodom.



SLIKA 3.28 *Debugovanje snimljene procedure pomoću unificiranog VS.NET debagera*

DEBAGOVANJE KADA SU POMEŠANI JEZICI

Prethodne verzije Visual Studia korišćene pomoću T-SQL Debuggera za debugovanje SQL koda. T-SQL Debugger nije mnogo koristan sa kodom kod koga su pomešani jezici.

Sa VS.NET-om, možete podesiti debugovanje više pomešanih jezika iz VB.NET i ADO.NET koda. Možete ovo uraditi klikom desnim tasterom miša iznad VB.NET projekta i izborom Properties opcije. Odavde izaberite Debugging iz Configuration Properties fascikle. Potvrdite SQL Server Debugging polje za potvrdu i izaberite Apply. Sada, kada debugujete VB.NET i ADO.NET kod, automatski ćete ući u značajan SQL kod u okviru snimljenih procedura koje referencirate.

VS.NET komande za debugovanje su iste za sve jezike. Možete proći kroz iskaze snimljenih procedura, podesiti tačke preloma i pregledati vrednosti lokalnih promenljivih za vreme izvršavanja.

Naučili ste mnogo u poslednjih nekoliko odeljaka. Da sumiramo - Database Designer je kolekcija alata koja omogućava da:

- kreirate nove tabele i editujete postojeće
- pregledate i editujete redove postojeće tabele
- ustanovite ograničenja za pojedinačne kolone
- kreirate nove indekse i editujete postojeće
- ustanovite nove relacije između tabela i editujete postojeće
- debugujete SQL kod u okviru okruženja sa više jezika.

Ovo su operacije koje su potrebne za dizajniranje nove, ili za poboljšanje dizajna postojeće baze podataka. Čak i najjednostavniji DBMS-ovi obezbeđuju ove mogućnosti i možete dizajnirati baze podataka za različite DBMS-ove iz Visual Studio .NET okruženja. Database Designer može da rukuje Access, SQL i Oracle bazama podataka - imate zajednički alat za sve ove različite tipove baza podataka i kroz različite tipove jezika.

Korišćenje Component Designera

Mnoge osnovne teme su obrađene u ovom poglavlju, ali niste napisali ni jednu liniju VB.NET koda. Vizuelni alati baza podataka su integrisani u okruženje Visual Studia, ali kako oni pomažu VB.NET programeru? Kao što smo ranije pomenuli, što više radite sa alatima baza podataka, manje koda ćete morati da pišete. Podesite svoja ograničenja i relacije pravilno i nećete morati da brinete o održavanju referencijalnog integriteta iz svog koda. Napišite snimljene procedure za najčešće zadatke (na primer, za dodavanje klijenta, proizvoda, faktura i tako dalje) i možete pozivati ove procedure kao metode baze podataka. Na kraju, testiranjem i debugovanjem svojih upita unapred možete jednostavno kopirati SQL iskaze iz Query Designera i prebaciti ih u svoj kod.

Da biste dobili uvid kako mogu pomoći vizuelni alati baza podataka pri razvijanju aplikacija baza podataka, pregledaćete neke primere u ovom odeljku. Ovo su teme koje ćete istražiti detaljno u sledećim poglavljima, ali bismo želeli da završimo ovo poglavlje nekim "opipljivim" dokazom o korisnosti alata koji su obrađeni. Nećete pisati nikakav kod; nastavice vizuelni pri-

laz programiranju baze podataka. Cilj je da na kraju kreirate DataSet korišćenjem vizuelnih alata i čarobnjaka.

Konfigurisanje DataAdapter objekta

Kao što se sećate, kreirali ste Data Connection korišćenjem standardne Windows aplikacije ranije u ovom poglavlju, u odeljku "Povezivanje sa bazom podataka". Pogledajte ponovo kako možete da proširite taj Connection objekat i da vizuelno radite sa ADO.NET objektima. Pokazaćemo kako da vizuelno konfigurirate svoj DataAdapter objekat korišćenjem čarobnjaka. Ovaj čarobnjak će, umesto Vas, automatski generisati Vaše ADO.NET Connection i Command objekte.

NAPOMENA

ADO.NET Command objekat je bilo koji objekat baze podataka koji vraća jedan, ili više redova iz baze podataka. On može da bude snimljena procedura, tabela, ili SQL iskaz. Ako ste ranije radili sa ADO 2.x, Command objekti treba da su Vam poznati. ■

Najpre, uspostavite Data Connection (ako to već niste uradili) sa Northwind bazom podataka. Proces uspostavljanja veze baze podataka sa Connection objektom je isti kao što je objašnjeno ranije u ovom poglavlju, u odeljku "Povezivanje sa bazama podataka".

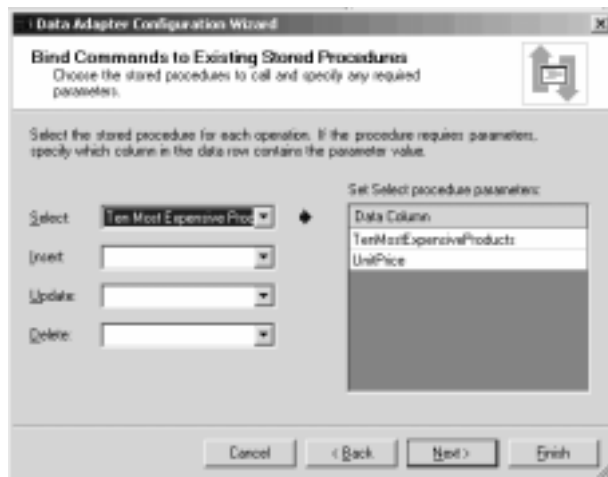
Ako nemate projekat sa Data Connection sa Northwind bazom podataka, sada kreirajte jedan:

1. Započnite novu Windows aplikaciju, otvorite njen Server Explorer prozor i dodajte Data Connection prema Northwind bazi podataka.
2. Otvorite Toolbox prozor. Iz Data kartice prevucite SqlDataAdapter i spustite ga na svoju formu. Ovo će pokrenuti Data Adapter Configuration Wizard. Izaberite Next da biste prešli sa uvodnog ekrana.
3. Prvi ekran čarobnjaka traži da odredite koju vezu da koristite. Možete koristiti postojeću, ili kreirati novu. Izaberite Northwind vezu koju ste kreirali ranije.
4. Sledeći ekran daje tri opcije koje možete koristiti za generisanje svojih Command objekata. U kasnijim poglavljima radićete sa svim ovim opcijama, ali, za sada, izaberite opciju Use Existing Stored Procedure (Koristi postojeću snimljenu proceduru).

SAVET

Uvek je dobro "držati se" snimljenih procedura, pre nego SQL iskaza, jer su one unapred optimizovane i brže će se izvršavati. ■

5. Pošto kliknete Next dugme, videćete ekran koji traži da izaberete snimljenu proceduru. U Select padajućoj listi izaberite snimljenu proceduru pod naslovom *Ten Most Expensive Products* (Deset najskupljih proizvoda). Pošto ovo uradite, videćete parametre koji se automatski popunjavaju, kao što možete videti na slici 3.29.



SLIKA 3.29 Mapiranje snimljenih procedura korišćenjem Data Adapter Configuration Wizarda

NAPOMENA

Ovaj ekran može da bude sasvim različit od onog koga ste videli ranije u prethodnim verzijama Visual Studia, jer predstavlja mogućnost ne samo da izaberete snimljenu proceduru, već i da mapirate snimljene procedure sa različitim akcijama (ovo ćemo obraditi detaljnije u Poglavlju 6). ■

6. Izaberite Next, pa Finish da biste napustili čarobnjak. Ako pogledate Component Designer pano, videćete dve nove kontrole: SqlDataAdapter i SqlConnection1.

Zbog toga što nema ADO.NET Connection iz Customers tabele, VS.NET je dodao novu SqlConnection kontrolu u Component Designer. Možete videti da je VS.NET automatski izabrao najpogodniji Connection tip zasnovan na provajderu koji ste izabrali. U ovom slučaju on koristi SqlConnection objekat. Ovaj SqlConnection1 objekat je zasnovan na originalnoj vezi Customers tabele u Server Exploreru (to je, samo tu, Data Connection čije je ime zasnovano na Vašem računaru i imenu baze podataka). Ako kliknete ovu kontrolu, videćete svojstva stringa za povezivanja izlistana u Properties prozoru.

SAVET

VS.NET omogućava da editujete string za povezivanje direktno iz ovog prozora. Ovo je odlično za pravljenje prototipova, jer možete da prepustite da, umesto Vas, čarobnjak generiše vezu, pa da je pomoću copy/paste iskopirate u svoj kod. ■

“Upregnite” sada moć Component Designera i pogledajte vizuelne ADO.NET kontrole. U ovom odeljku pokazujemo kako da na najbolji način iskoristite Component Designer za pristup podacima korišćenjem pokaži-i-klikni operacija za dinamičko kreiranje XML šeme i ADO.NET DataSeta. Takođe ćete videti XML Designer u akciji.

Korišćenje XML Designera

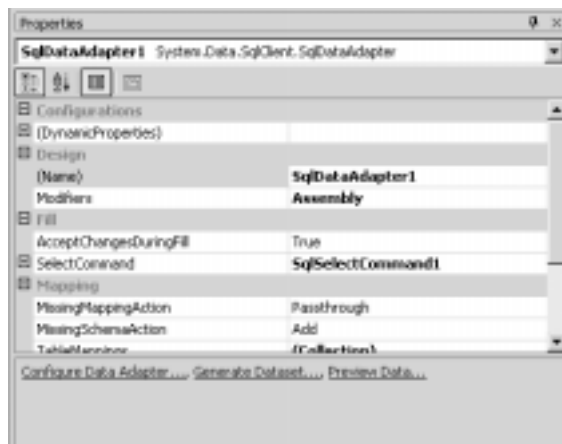
Sada, kada imate SqlConnection i SqlDataAdapter kontrole u svom Component Designeru, možete ići dalje i generisati DataSet. Popunićete DataSet na osnovu snimljene procedure koju ste izabrali u DataAdapteru.

1. Kliknite SqlDataAdapter kontrolu da biste prikazali njen prozor svojstava. Ispod Properties panoa treba da vidite mali sivi boks (kao na slici 3.30), koji sadrži nekoliko hiperveza: Configure Data Adapter, Generate Dataset i Preview Data.
2. Zbog toga što ste već konfigurisali svoj DataAdapter, izaberite drugu opciju Generate DataSet. Ovo pokreće okvir za dijalog koji traži da izaberete neke opcije.

NAPOMENA

Treća opcija Preview Data prikazuje vizuelni interfejs na kome možete da pregledate podatke koje vraća Vaš DataSet. Sa tim možete sada eksperimentisati, ako želite, ali nemojte da brinete - ovo ćemo detaljno obraditi u Poglavlju 7. ■

3. Izaberite New opciju i ukucajte ime dsExpensive za svoj DataSet. Ostavite Ten Most Expensive Products snimljenu proceduru kao Vaš kriterijum izbora, kao što je prikazano na slici 3.31. Takođe ćete želeći da ostavite potvrđenu opciju Add This DataSet To The Designer, tako da možete raditi sa njom.



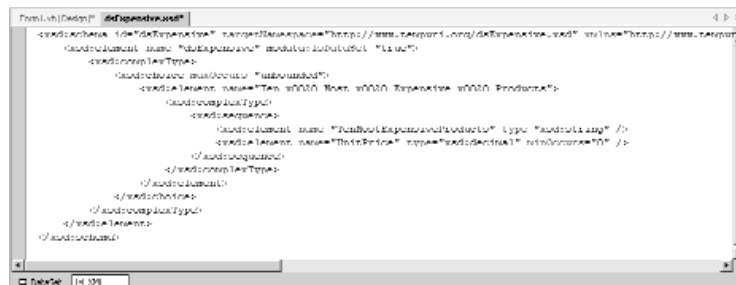
SLIKA 3.30 Rad sa DataAdapter opcijom

**SLIKA 3.31** Konfigurisanje DataSeta

4. Pošto kliknete OK, VS.NET, umesto Vas, automatski kreira ADO.NET DataSet. Ako pogledate Component Designer, videćete novu kontrolu nazvanu dsExpensive (ili sa imenom koje Vi odaberete).
5. Kliknite ovu dsExpensive DataSet kontrolu da biste prikazali svojstva.
6. Ispod Properties panoa videćete neke hiperveze. Izaberite View Schema opciju.

Izbor View Schema opcije pokreće XML Designer, koji smo pomenuli ranije. Ovo omogućava da vidite vizuelno predstavljanje Vašeg DataSeta u grid (mreža) formatu, kao što je prikazano na slici 3.32, ili XML formatu (klikom na XML karticu na dnu ekrana), kao što je prikazano na slici 3.33. Ne samo da možete videti podatke u XML formatu, već VS.NET takođe automatski kreira XML fajl šeme sa .xsd ekstenzijom, prigodno nazvan po Vašem DataSetu. Možete pogledati ovaj fajl u Solution Exploreru.

**SLIKA 3.32** Pregledanje DataSetova na vizuelnoj mreži



SLIKA 3.33 Pregledanje DataSetova u XML formatu

Kao što možete videti, Component Designer obezbeđuje moćan način za generisanje DataSetova "u hodu". Možete, čak, kombinovati podatke iz više tabela i heterogenih izvora podataka korišćenjem Data Adapter Configuration Wizarda. Pored toga, ovi alati omogućavaju da lako generišete XML fajlove šema koje možete distribuirati eksternim aplikacijama u univerzalno prihvaćenom formatu.

UPOZORENJE

XML Designer obezbeđuje indirektan način za rad sa DataSet šemama. Ako promenite svojstva DataAdaptera, moraćete da ponovo generišete DataSet, ili da ručno editujete XSD fajl. ■

Zaključak

Ovde je završeno predstavljanje vizuelnih alata baza podataka, koje ćete često koristiti u sledećim poglavljima. Kada dizajnirate novu bazu podataka, podešavate postojeću, ili kreirate upite, ili snimljene procedure, konstatovaćete da su ovi alati nezamenljivi. Novi vizuelni alati baza podataka obezbeđuju moćan način za rad sa ADO.NET strukturnim okvirom bez kucanja linije koda.

U sledećem poglavlju preći ćete na istraživanje jezika za manipulisanje bazama podataka - SQL-a (Structured Query Language). SQL je univerzalni jezik za manipulisanje bazama podataka (i njihovim sadržajem i strukturom) i to je ono što svaki programer baza podataka mora da zna. Query Designer je odličan alat, ali ne može kompletno da zameni razumevanje SQL iskaza i njegovu sintaksu.