helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

SQL Server 2005. Programowanie. Od podstaw

Autor: Robert Vieira
Tłumaczenie: Piotr Balczyński, Maria Chaniewska, Grzegorz Kostek
ISBN: 83-246-0653-X
Tytuł oryginału: Beginning SQL Server 2005 Programming
Format: B5, stron: 728



Rozpocznij pracę z SQL Server 2005

- Dowiedz się, jak działają systemy RDBMS
- Poznaj narzędzia SQL Server 2005
- Naucz się obsługiwać bazy danych za pomocą SQL Server 2005

SQL Server 2005 to najnowsza wersja niezawodnego, wydajnego i wysoce skalowalnego systemu zarządzania relacyjnymi bazami danych (RDBMS) autorstwa Microsoftu. Podobnie jak wcześniejsze wersje tego produktu, SQL Server 2005 bazuje na języku T-SQL, ale zapewnia lepszą obsługę XML, danych definiowanych przez użytkownika oraz platformy .NET, a ponadto udostępnia dodatkowe usługi. Dzięki swym możliwościom doskonale nadaje się do tego, by być podstawą rozmaitych aplikacji potrzebujących dostępu do bazy danych.

Książka „SQL Server 2005. Programowanie. Od podstaw" przeznaczona jest dla programistów, którzy chcą rozpocząć pracę z SQL Server 2005. Dzięki niej poznasz podstawy funkcjonowania systemów RDBMS oraz języków SQL i T-SQL. Nauczysz się korzystać z narzędzi udostępnianych przez SQL Server 2005 oraz dowiesz się, jak wykonywać zarówno podstawowe, jak i bardziej zaawansowane operacje na bazach danych. Ta książka pozwoli Ci szybko opanować możliwości, jakie daje SQL Server 2005, i przystąpić do pisania stabilnych oraz wydajnych aplikacji bazodanowych.

- Wprowadzenie do systemów RDBMS
- Języki SQL i T-SQL
- Narzędzia dostępne w SQL Server 2005
- Tworzenie i modyfikowanie tabel
- Korzystanie ze złączeń i ograniczeń
- Normalizacja podstaw projektowania
- Tworzenie skryptów, programów wsadowych i procedur składowanych
- Obsługa transakcji i blokad
- Używanie wyzwalaczy
- Raporty
- Obsługa danych XML

Twórz niezawodne i wydajne aplikacje bazodanowe za pomocą SQL Server 2005

Wydawnictwo Helion
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl

Spis treści

Wstęp 15

Rozdział 1. Podstawy RDBMS — z czego składa się baza danych SQL Server 21

Przegląd obiektów bazy danych 22
Obiekt Baza danych 22
Dziennik transakcji 27
Najbardziej podstawowy obiekt bazy danych — tabela 27
Grupy plików 29
Diagramy 29
Widoki 31
Procedury składowane 31
Funkcje zdefiniowane przez użytkownika 32
Użytkownicy i role 32
Reguły 33
Wartości domyślne 33
Typy danych zdefiniowane przez użytkownika 33
Katalogi wyszukiwania pełnotekstowego 33
Typy danych w systemie SQL Server 34
Wartości NULL 38
Identyfikatory obiektów w systemie SQL Server 39
Nazwy 39
Konwencje nazewnicze 39
Podsumowanie 40

Rozdział 2. Narzędzia 41

Books Online 42
SQL Server Configuration Manager 43
Zarządzanie usługami 44
Konfiguracja sieci 44
Protokoły 46
Klient 48
SQL Server Management Studio 50
Rozpoczęcie pracy 51
Okno zapytań 55
SQL Server Integration Services (SSIS) 61
Bulk Copy Program (bcp) 62
SQL Server Profiler 62
sqlcmd 63
Podsumowanie 64

Rozdział 3. Podstawowe polecenia języka T-SQL 65
Podstawowe polecenie — SELECT 66
Polecenie SELECT i klauzula FROM 66
Klauzula WHERE 70
ORDER BY 73
Agregacje danych za pomocą klauzuli GROUP BY 77
Tworzenie warunków za pomocą klauzuli HAVING 85
Generowanie XML-a za pomocą klauzuli FOR XML 87
Wykorzystanie wskazówek za pomocą klauzuli OPTION 88
Predykaty DISTINCT i ALL 88
Wprowadzanie danych za pomocą polecenia INSERT 90
Polecenie INSERT INTO...SELECT 95
Wprowadzanie zmian za pomocą polecenia UPDATE 97
Polecenie DELETE 100
Podsumowanie 101
Ćwiczenia 102

Rozdział 4. Złączenia (JOINs) 103
Złączenia JOIN 104
Złączenia wewnętrzne (INNER JOIN) 105
Dlaczego INNER JOIN przypomina klauzulę WHERE 110
Złączenia zewnętrzne 114
Proste złączenie zewnętrzne 115
Bardziej skomplikowane złączenia zewnętrzne 120
Spojrzenie w obie strony za pomocą złączeń pełnych 124
Złączenia krzyżowe 126
Alternatywne składnie złączeń 127
Alternatywne INNER JOIN 127
Alternatywne OUTER JOIN 128
Alternatywne CROSS JOIN 129
Unia 130
Podsumowanie 135
Ćwiczenia 135

Rozdział 5. Tworzenie i modyfikacja tabel 137
Nazwy obiektów w systemie SQL Server 137
Nazwa schematu 138
Nazwa bazy danych 141
Nazwa serwera 141
Przegląd ustawień domyślnych 141
Polecenie CREATE 142
CREATE DATABASE 142
CREATE TABLE 148
Polecenie ALTER 161
ALTER DATABASE 161
ALTER TABLE 164
Polecenie DROP 168
Wykorzystanie narzędzia GUI 169
Tworzenie bazy danych za pomocą SQL Server Management Studio 169
Powrót do kodowania — podstawy tworzenia skryptów za pomocą SQL Server Management Studio 175
Podsumowanie 176
Ćwiczenia 176

Rozdział 6. Ograniczenia ... 177

Rodzaje ograniczeń ... 178
Ograniczenia domeny ... 178
Ograniczenia encji ... 179
Ograniczenia integralności referencyjnej ... 179
Nazewnictwo ograniczeń ... 180
Ograniczenia kluczy ... 181
Ograniczenia PRIMARY KEY ... 182
Ograniczenia FOREIGN KEY ... 184
Ograniczenia UNIQUE ... 195
Ograniczenia CHECK ... 196
Ograniczenia DEFAULT ... 198
Definiowanie ograniczenia DEFAULT w poleceniu CREATE TABLE ... 199
Dodawanie ograniczenia DEFAULT do istniejącej tabeli ... 199
Wyłączanie ograniczeń ... 200
Ignorowanie niewłaściwych danych podczas tworzenia ograniczenia ... 200
Tymczasowe wyłączenie istniejącego ograniczenia ... 202
Reguły i wartości domyślne — kuzyni ograniczeń ... 204
Reguły ... 204
Wartości domyślne ... 206
Określanie tabel wykorzystujących konkretne reguły i wartości domyślne ... 207
Wyzwalacze a integralność danych ... 208
Czego używać ... 208
Podsumowanie ... 209

Rozdział 7. Zwiększanie możliwości zapytań ... 211

Co to jest podzapytanie ... 212
Tworzenie podzapytania zagnieżdżonego ... 213
Podzapytania skorelowane ... 216
W jaki sposób działają podzapytania skorelowane ... 216
Podzapytania skorelowane w klauzuli WHERE ... 217
Praca z wartościami NULL — funkcja ISNULL ... 221
Tabele pochodne ... 222
Operator EXISTS ... 224
Inne sposoby wykorzystania EXISTS ... 226
Mieszanie typów danych — funkcje CAST i CONVERT ... 227
Kwestie wydajności ... 230
Lepiej używać złączeń, podzapytań czy czegoś innego? ... 230
Podsumowanie ... 232
Ćwiczenia ... 232

Rozdział 8. „Być normalnym" — normalizacja i inne kwestie podstaw projektowania ... 233

Tabele ... 234
Utrzymywanie danych „normalnymi" ... 234
Zanim zaczniemy ... 235
Pierwsza postać normalna ... 237
Druga postać normalna ... 241
Trzecia postać normalna ... 242
Inne postacie normalne ... 244
Relacje ... 245
Jeden do jednego ... 245

Jeden do jednego lub wielu ... 247
Wiele do wielu ... 249
Diagramy ... 253
Tabele ... 255
Dodawanie i usuwanie tabel ... 256
Relacje ... 262
Denormalizacja ... 266
Poza normalizacją ... 267
Prostota ... 267
Wybór typów danych ... 267
Błędy w sposobie przechowywania ... 268
Szkic prostego przykładu ... 269
Tworzenie bazy danych ... 269
Dodawanie diagramu i tabeli początkowej ... 269
Dodawanie relacji ... 274
Dodawanie ograniczeń ... 276
Podsumowanie ... 278
Ćwiczenia ... 278

Rozdział 9. Struktury danych i indeksów w SQL Server ... 279

Struktury danych w SQL Serverze ... 279
Baza danych ... 279
Zakres ... 280
Strona ... 280
Wiersze ... 281
Podstawy indeksów ... 281
B-drzewa ... 283
Sposób udostępniania danych w SQL Serverze ... 286
Tworzenie, modyfikacja i usuwanie indeksów ... 295
Polecenie CREATE INDEX ... 295
Tworzenie indeksów XML ... 302
Indeksy wywiedzione tworzone z ograniczeniami ... 303
Mądry wybór — jaki indeks, gdzie i kiedy go stosować ... 303
Selektywność ... 304
Przyglądanie się kosztom — gdy mniej znaczy więcej ... 305
Wybieranie właściwego indeksu klastrowego ... 305
Kolejność kolumn ... 308
Usuwanie indeksów ... 308
Korzystanie z Database Engine Tuning Advisor ... 309
Utrzymywanie indeksów ... 309
Fragmentacja ... 310
Identyfikacja fragmentacji a prawdopodobieństwo podziału stron ... 310
Podsumowanie ... 314
Ćwiczenia ... 315

Rozdział 10. Widoki ... 317

Proste widoki ... 317
Widoki jako filtry ... 321
Bardziej złożone widoki ... 323
Wykorzystanie widoków do zmiany danych bez użycia wyzwalaczy INSTEAD OF ... 326
Edycja widoków za pomocą T-SQL-a ... 330

Usuwanie widoków ... 330
Tworzenie i edycja widoków w SQL Server Management Studio ... 330
Edycja widoków w SQL Server Management Studio ... 334
Sprawdzanie — wyświetlanie istniejącego kodu ... 334
Ochrona kodu — szyfrowanie widoków ... 336
Słowo o schemacie powiązania ... 337
Upodabnianie widoków do tabel za pomocą VIEW_METADATA ... 338
Widoki indeksowane (materializowane) ... 338
Podsumowanie ... 342
Ćwiczenia ... 342

Rozdział 11. Tworzenie skryptów i programów wsadowych ... 343

Podstawy skryptów ... 344
Instrukcja USE ... 344
Deklaracja zmiennych ... 345
Użycie @@IDENTITY ... 349
Użycie @@ROWCOUNT ... 352
Batch ... 353
Błędy w programach wsadowych ... 355
Kiedy korzystać z programów wsadowych ... 356
SQLCMD ... 359
Dynamiczny SQL — tworzenie kodu „w locie” za pomocą polecenia EXEC ... 363
Zastosowania EXEC ... 364
Podsumowanie ... 368
Ćwiczenia ... 369

Rozdział 12. Procedury składowane ... 371

Tworzenie procedury składowanej — podstawy składni ... 372
Przykład prostej procedury ... 372
Zmiana procedury składowanej za pomocą ALTER ... 373
Usuwanie procedur składowanych ... 374
Określanie parametrów ... 374
Deklaracja parametrów ... 374
Instrukcje sterujące przepływem danych ... 379
Polecenie IF ... ELSE ... 380
Instrukcja CASE ... 390
Zapętlanie przy użyciu instrukcji WHILE ... 397
Instrukcja WAITFOR ... 398
Bloki TRY/CATCH ... 399
Potwierdzanie powodzenia lub niepowodzenia za pomocą wartości zwrotnych ... 399
Jak używać RETURN ... 400
Obsługa błędów ... 402
Dotychczasowe sposoby... ... 403
Bloki TRY/CATCH ... 409
Obsługa błędów przed ich wystąpieniem ... 412
Ręczne zgłaszanie błędu ... 415
Dodawanie własnych komunikatów o błędzie ... 419
Co oferuje procedura składowana ... 422
Tworzenie wywoływalnych procesów ... 423
Używanie procedur składowanych w celu zapewnienia bezpieczeństwa ... 424
Procedura składowana a wydajność ... 425

Rozszerzone procedury składowane (XPs) ... 427
Krótkie spojrzenie na rekurencję ... 428
Kolekcje .NET ... 430
Podsumowanie ... 432
Ćwiczenia ... 432

Rozdział 13. Funkcje definiowane przez użytkownika ... 435

Czym są funkcje użytkownika ... 435
Funkcje użytkownika zwracające wartości skalarne ... 437
Funkcje użytkownika zwracające tabele ... 440
Zrozumienie jednoznaczności ... 448
Debugowanie funkcji definiowanych przez użytkownika ... 450
Zastosowanie .NET w bazach danych ... 450
Podsumowanie ... 451
Ćwiczenia ... 451

Rozdział 14. Transakcje i blokady ... 453

Transakcje ... 453
BEGIN TRANSACTION ... 455
COMMIT TRANSACTION ... 455
ROLLBACK TRANSACTION ... 455
SAVE TRANSACTION ... 456
Jak działa dziennik SQL Servera ... 456
Awaria i odtwarzanie ... 458
Domniemanie transakcji ... 459
Blokady i współbieżność ... 460
Jakim problemom mogą zapobiegać blokady ... 461
Blokowalne zasoby ... 464
Rozwijanie blokad i wpływ blokad na wydajność ... 465
Tryby blokowania ... 466
Kompatybilność blokad ... 468
Określanie konkretnego typu blokady — wskazówki optymalizatora ... 469
Określanie poziomu izolowania ... 470
Obsługa zakleszczeń (lub błąd 1205) ... 473
W jaki sposób SQL Server określa występowanie zakleszczenia ... 474
W jaki sposób wybierane są ofiary zakleszczenia ... 474
Unikanie zakleszczeń ... 474
Podsumowanie ... 477

Rozdział 15. Wyzwalacze ... 479

Czym są wyzwalacze ... 480
Klauzula ON ... 482
WITH ENCRYPTION ... 482
Użycie klauzuli FOR|AFTER i INSTEAD OF ... 482
WITH APPEND ... 485
NOT FOR REPLICATION ... 486
AS ... 486
Użycie wyzwalaczy do określenia zasad integralności danych ... 486
Obsługa wymagań pochodzących z innych tabel ... 486
Użycie wyzwalaczy w celu sprawdzenia delty aktualizacji ... 488
Użycie wyzwalaczy w celu wywołania własnych komunikatów o błędach ... 490
Inne powszechne zastosowania wyzwalaczy ... 490

Inne kwestie związane z wyzwalaczami ... 491
Wyzwalacze można zagnieżdżać ... 491
Wyzwalacze mogą być rekurencyjne ... 491
Wyzwalacze nie zapobiegają zmianom architektury ... 492
Wyzwalacze można wyłączyć bez ich usuwania ... 492
Kolejność uruchamiania wyzwalaczy ... 493
Wyzwalacze INSTEAD OF ... 495
Rozważania o wydajności ... 496
Działanie wyzwalaczy jest raczej reakcją niż akcją ... 496
Wyzwalacze a problem współbieżności ... 496
Użycie IF UPDATE() i COLUMNS_UPDATED() ... 497
Twórz jak najprostsze wyzwalacze ... 499
Nie zapominaj o wyzwalaczach, gdy wybierasz indeksy ... 500
Unikaj cofania wewnątrz wyzwalaczy ... 500
Usuwanie wyzwalaczy ... 500
Podsumowanie ... 501

Rozdział 16. Krótki elementarz języka XML ... 503

Podstawy języka XML ... 504
Części składowe dokumentu XML ... 505
Przestrzeń nazw ... 513
Zawartość elementów ... 515
Poprawność strukturalna a poprawność składniowa — schemat XML i DTD ... 516
Co to ma wspólnego z systemem SQL Server? ... 517
Dostarczanie relacyjnych danych w formacie XML ... 517
RAW ... 519
AUTO ... 521
EXPLICIT ... 523
PATH ... 539
OPENXML ... 544
Dwa słowa o XSLT ... 550
Podsumowanie ... 552

Rozdział 17. Raport ze służby — spojrzenie na usługi raportowe ... 553

Usługi raportowe ... 554
Tworzenie prostego modelu raportu ... 554
Widok źródła danych ... 559
Tworzenie raportów ... 565
Projekt serwera raportowego ... 568
Instalowanie raportu ... 573
Podsumowanie ... 574

Rozdział 18. Integracja z usługami integracyjnymi ... 575

Zrozumienie problemu ... 576
Używanie Import and Export Wizard do tworzenia podstawowych pakietów ... 576
Uruchamianie pakietów ... 583
Używanie Execute Package Utility ... 583
Uruchamianie w SQL Server Business Intelligence Development Studio ... 586
Uruchamianie w SQL Server Management Studio ... 586
Edycja pakietów ... 587
Podsumowanie ... 589

Rozdział 19. Zabawa w administratora 591
Harmonogram zadań 592
Tworzenie kont operatorów 593
Tworzenie zadań 595
Tworzenie kopii zapasowych i odtwarzanie bazy 607
Tworzenie kopii zapasowej 607
Recovery Models (modele odtwarzania) 611
Odtwarzanie 612
Utrzymywanie indeksów 614
ALTER INDEX 615
Archiwizacja danych 617
Podsumowanie 618
Ćwiczenia 619

Dodatek A Rozwiązania ćwiczeń 621

Dodatek B Funkcje systemowe 631

Dodatek C Używanie właściwych narzędzi 689

Dodatek D Bardzo proste przykłady połączeń 697

Dodatek E Instalacja i użycie przykładów 701

Skorowidz 705

2

Narzędzia

Skoro poznaliśmy już różne rodzaje obiektów występujących w systemie SQL Server, powinniśmy dowiedzieć się, jak je znaleźć. Przyda się również wiedza na temat monitorowania systemu.

W niniejszym rozdziale zapoznamy się z narzędziami oferowanymi przez SQL Server. Niektóre spełniają tylko kilka wysoce wyspecjalizowanych funkcji, inne pozwalają wykonać całą masę różnych zadań. Większość z nich jest częścią systemu SQL Server od dawna. Jedno jest pewne: wszystko, co ma jakikolwiek związek z zestawem narzędzi systemu SQL Server, zostało poddane gruntownemu przeglądowi podczas opracowywania wersji 2005. Głównym celem zespołu zajmującego się projektowaniem narzędzi tej edycji systemu było uproszczenie problemu zawartego w pytaniu: „Gdzie można znaleźć to narzędzie?". Mogę powiedzieć, że z punktu widzenia początkujących użytkowników systemu cel ten został zrealizowany (oczywiście starzy wyjadacze, tacy jak ja, będą sobie zadawać pytanie: „Gdzie się to wszystko podziało?").

W tym rozdziale omówimy niektóre spośród poniższych narzędzi:

- SQL Server Books Online,
- SQL Server Configuration Manager,
- SQL Server Management Workbench,
- SQL Server Integration Services (SSIS) wraz z kreatorem importu i eksportu,
- Database Engine Tuning Advisor,
- Report Manager,
- Bulk Copy Program (bcp),
- SQL Server Profiler,
- sqlcmd.

Musisz uważać, jeśli zasięgasz rady znajomych, którzy posiadają doświadczenie w zakresie wersji systemu starszej niż SQL Server 2005. Zestaw narzędzi w najnowszej edycji systemu został opracowany na nowo i wiele rzeczy, do których przywykli użytkownicy wcześniejszych wersji serwera, występuje pod innymi nazwami, zostało przesuniętych lub nawet usuniętych w celu integracji z innym narzędziem.

Większość starych pojęć nadal gdzieś funkcjonuje, ale mogły one zmienić swoje miejsce.

Books Online

Czy *Books Online* to narzędzie? Wydaje mi się, że tak. Niezależnie od tego, ile razy przeczytasz tę lub inne książki dotyczące systemu SQL Server, nie zdołasz zapamiętać wszystkiego, co musisz o nim wiedzieć. SQL Server jest jednym z moich flagowych produktów, a i tak nie mogę wszystkiego zapamiętać. Books Online to po prostu jedno z podstawowych narzędzi systemu SQL Server.

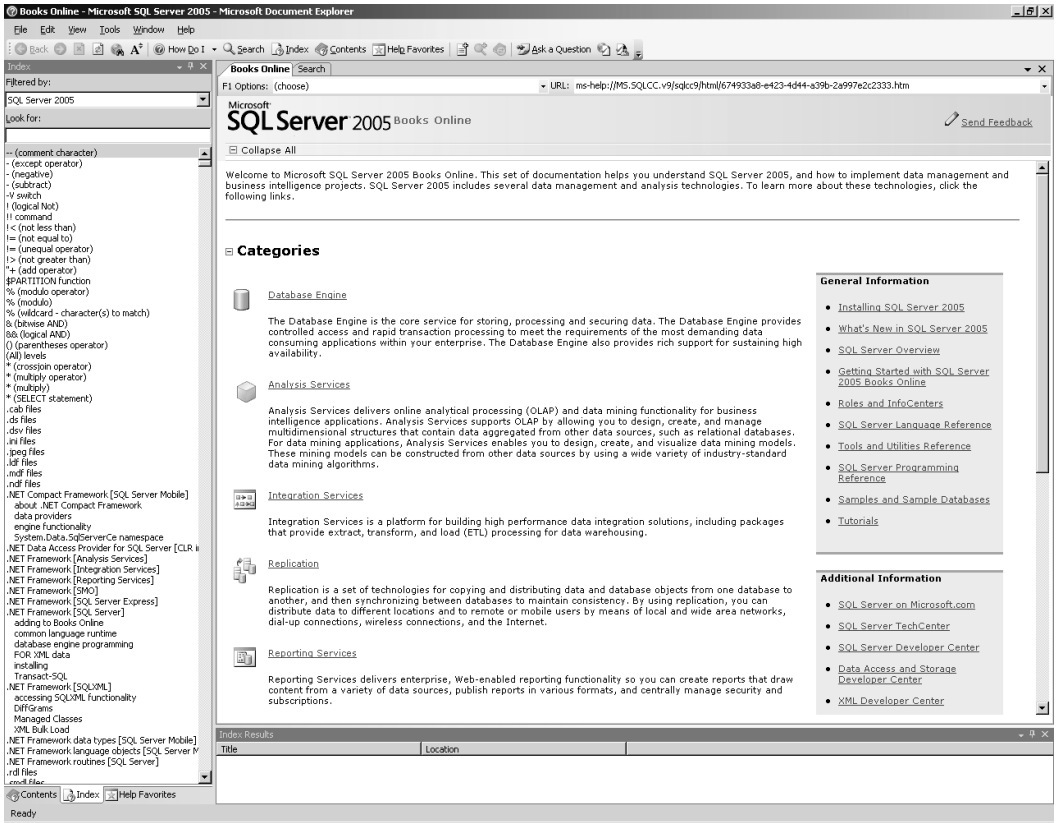
Uważam, że książek lub innych materiałów dotyczących programowania nigdy nie jest za wiele. Zacząłem programować mniej więcej w 1980 r. Wtedy można było zapamiętać większość rzeczy (choć nie wszystkie). Obecnie jest to po prostu niemożliwe. Jeśli jesteś wszechstronnym programistą (co jest dość trudne w obecnych czasach), masz po prostu zbyt wiele zagadnień do zapamiętania, zaś te rzeczy, których nie używasz każdego dnia, są tracone wraz z obumierającymi komórkami mózgu.

Krótka rada: nie staraj się nawet zapamiętać tego wszystkiego. Trzeba pamiętać, że coś się da zrobić. Trzeba pamiętać to, co stanowi integralną podstawę naszych działań. Trzeba pamiętać o tym, z czym mamy codziennie do czynienia. I trzeba w końcu pamiętać o budowaniu księgozbioru podręcznego (zaczynając od tej książki).

Books Online wykorzystują zmodernizowany interfejs pomocy online .NET, który zastępuje używany dotąd standardowy interfejs pomocy, stosowany w serii produktów technicznych Microsoft (BackOffice, MSDN i Visual Studio).

Wszystko działa tutaj zgodnie z oczekiwaniami, więc nie będę się zagłębiał w szczegóły obsługi systemu pomocy. Wystarczy powiedzieć, że SQL Server Books Online to wspaniały poradnik, z którego można korzystać niezależnie od tego, na jakiej maszynie się pracuje. Books Online zazwyczaj zawierają też uaktualnione informacje, których nie można znaleźć w drukowanej dokumentacji.

Nie w każdym systemie, na którym przyjdzie Ci pracować, zainstalowane będą Books Online (BOL). Dzieje się tak, gdyż można usunąć zaznaczenie opcji instalowania BOL w momencie instalacji serwera SQL. Zalecam jednak instalowanie BOL nawet wtedy, gdy dysponuje się mocno ograniczoną ilością miejsca na dysku. Nie zajmują one tak dużo miejsca, zwłaszcza jeśli rozważy się koszt jednego megabajta dysku. Dzięki dostępowi do tego poradnika na dowolnej maszynie z systemem SQL Server można zaoszczędzić mnóstwo czasu (na moim komputerze Books Online zajmują 100 MB).



Rysunek 2.1

SQL Server Configuration Manager

Z tego narzędzia najczęściej korzystają administratorzy odpowiedzialni za konfigurację komputerów w celu korzystania z bazy danych. Mimo to trzeba wiedzieć, jak ono działa.

SQL Server Computer Manager jest nowym narzędziem, które pojawiło się w systemie SQL Server 2005. W rzeczywistości połączono w nim kilka funkcji, które stosowano wcześniej w kilku różnych narzędziach. Elementy, którymi można zarządzać za pomocą modułu Configuration Manager, należą do dwóch dziedzin:

- zarządzania usługami,
- konfiguracji sieci.

Zarządzanie usługami

SQL Server to duży produkt. Jego różne części korzystają z szeregu usług działających w tle. Pełna instalacja obejmuje siedem usług. Wszystkimi można zarządzać z poziomu modułu SQL Server Configuration Manager.

Usługi, którymi można zarządzać z tego poziomu, obejmują:

- **SQL Server Analysis Services** — obsługuje silnik Analysis Services.
- **SQL Server FullText Search** — zgodnie z nazwą obsługuje silnik wyszukiwania pełnotekstowego.
- **SQL Server Reporting Services** — silnik obsługujący raportowanie.
- **SQL Server Agent** — główny silnik wykorzystywany przez wszystkie zadania zaplanowane w systemie SQL Server. Dzięki wykorzystaniu tej usługi można zaplanować wykonanie szerokiego zakresu prac (ang. *jobs*). Prace te mogą składać się z wielu etapów i mogą nawet rozdzielać się na różne zadania w zależności od wyników realizacji poprzednich zadań. Przykładami zadań wykonywanych przez SQL Server Agent mogą być wykonywanie kopii zapasowych lub rutynowe procesy importu i eksportu danych.
- **SQL Server** — podstawowy silnik baz danych, który operuje na magazynach danych oraz zapytaniach i jest odpowiedzialny za konfigurację systemu.
- **SQL Server Browser** — wspomaga rozgłaszanie systemu, aby umożliwić użytkownikom przeszukującym sieć lokalną sprawdzenie, czy na danej maszynie działa system SQL Server.
- **SQL Server Integration Services** — obsługuje silnik SSIS.

Konfiguracja sieci

Na ogół kwestie związane z połączeniami sieciowymi zależą od konfiguracji sieci klienta i dopasowania konfiguracji klienta do ustawień zastosowanych na serwerze.

SQL Server zapewnia kilka tzw. *bibliotek sieciowych* (ang. *Net-Libraries* lub *NetLibs*). Są to biblioteki dołączane dynamicznie (DLL), wykorzystywane przez SQL Server do komunikacji z niektórymi protokołami sieciowymi. NetLibs to pewnego rodzaju warstwa izolacji pomiędzy aplikacją kliencką a protokołem sieciowym, będącym w zasadzie językiem, za pomocą którego porozumiewają się między sobą karty sieciowe. Biblioteki te spełniają ten sam cel również na serwerze. NetLibs dostarczone w systemie SQL Server 2005 obejmują:

- Named Pipes,
- TCP/IP (domyślny),
- Shared Memory,
- VIA.

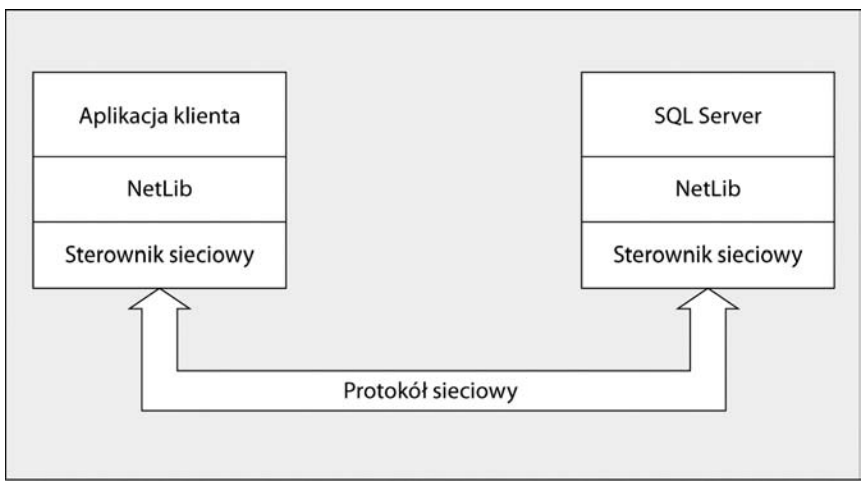
VIA to biblioteka sieciowa stworzona w celu współpracy z bardzo szczególnym (i drogim) sprzętem. Jeśli pracujesz w środowisku VIA, to z pewnością znasz jego szczególne wymagania. Tym, którzy nie operują w tym środowisku, wystarczy powiedzieć, że VIA zapewnia bardzo szybkie, choć drogie, rozwiązanie umożliwiające sprawną komunikację między serwerami. Z reguły jednak nie używa się go w przypadku zwykłego klienta.

Jeśli klient i serwer mają się porozumiewać między sobą poprzez protokół sieciowy, muszą posiadać tę samą bibliotekę NetLib. Jeśli serwer nie obsługuje biblioteki sieciowej używanej przez klienta, to nie uda się nawiązać połączenia i wygenerowany zostanie komunikat o błędzie *SQL Server Not Found* (nie odnaleziono serwera SQL Server).

Niezależnie od metody dostępu do danych i rodzaju zastosowanego sterownika (SQL Native Client, ODBC, OLE DB, DB-Lib) w każdym przypadku to właśnie sterownik będzie się komunikował z biblioteką NetLib. Przebieg tego procesu zilustrowano na rysunku 2.2. Jego kolejne etapy to:

1. Aplikacja kliencka nawiązuje kontakt ze sterownikiem (SQL Native Client, ODBC, OLE DB, DB-Lib).
2. Sterownik wywołuje bibliotekę NetLib klienta.
3. Biblioteka NetLib wywołuje odpowiedni protokół sieciowy i przesyła dane do biblioteki NetLib serwera.
4. Biblioteka NetLib serwera przekazuje żądania klienta do systemu SQL Server.

Rysunek 2.2

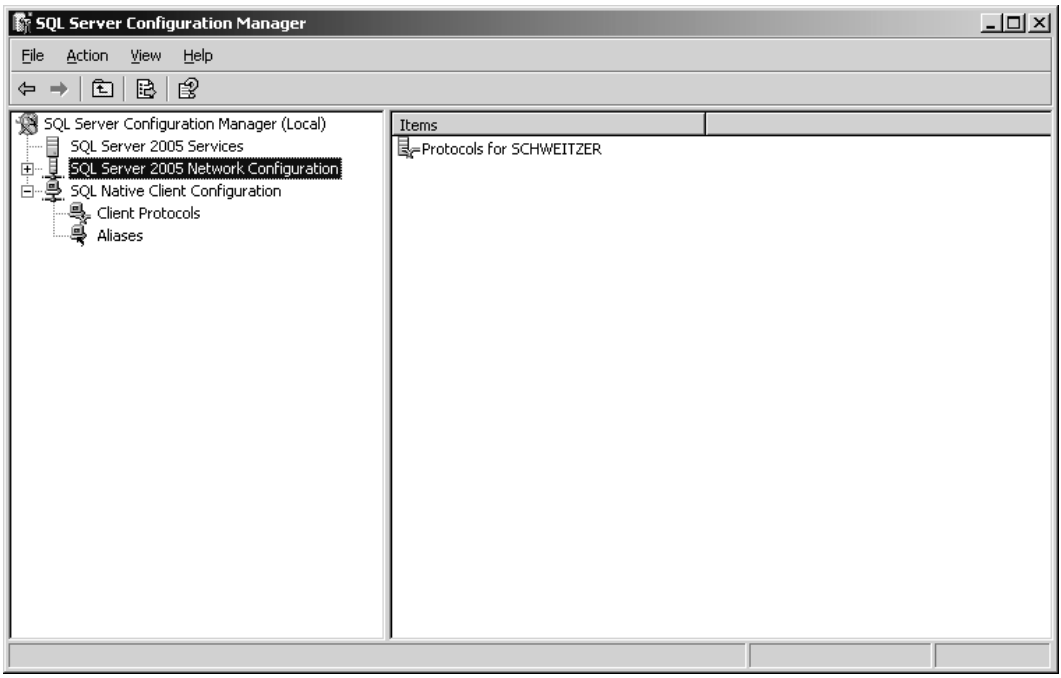


Odpowiedź z systemu SQL Server wysyłana jest w tym samym porządku, ale w przeciwnym kierunku.

> Jeśli znasz protokół TCP/IP, to powinieneś wiedzieć, że domyślnym portem, na którym będzie nasłuchiwać IP NetLib, jest 1433. Port można porównać do kanału radiowego — sygnały „skaczą dookoła" na różnych częstotliwościach, ale możesz z nich odnieść jakąś korzyść tylko wtedy, gdy „nasłuchujesz" na właściwym kanale.

Protokoły

Zacznijmy od pytania: „Z czego możemy wybierać?". Po uruchomieniu modułu *SQL Server Configuration Manager* i rozwinięciu listy drzewiastej *SQL Server 2005 Network Configuration* zobaczymy to, co zostało zaprezentowane na rysunku 2.3.

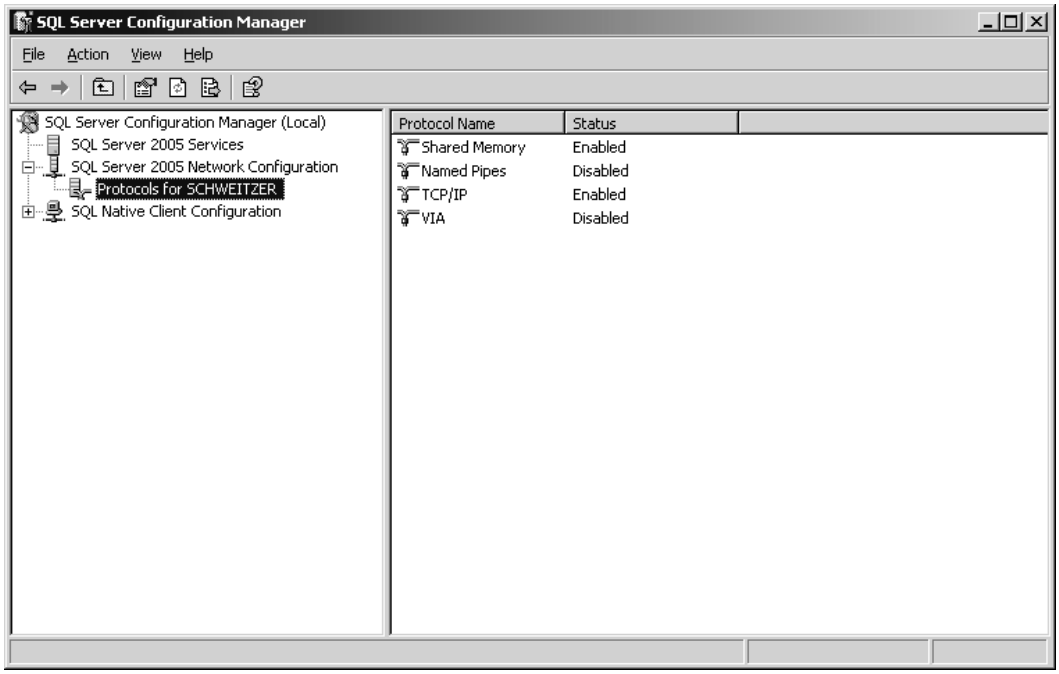


Rysunek 2.3

> Domyślnie włączona jest opcja *Shared Memory* i *TCP/IP*. W starszych wersjach produktu — w zależności od wersji systemu operacyjnego lub SQL Servera — włączone były domyślnie różne opcje.
>
> Aby móc zdalnie połączyć się z systemem SQL Server (np. z serwera internetowego lub z dowolnego klienta w sieci), należy włączyć przynajmniej jedną z pozostałych bibliotek NetLib.

Aby zobaczyć, czego nasz serwer *mógłby* nasłuchiwać, rozwiniemy element *Protocols for SCHWEITZER*, znajdujący się w grupie *SQL Server 2005 Network Configuration*, co zilustrowano na rysunku 2.4.

Należy pamiętać o tym, że jeśli klient ma nawiązać połączenie z serwerem, to serwer musi korzystać z tego samego protokołu co klient i nasłuchiwać na tym samym porcie. Dlatego też, gdybyśmy przebywali w środowisku potoków nazwanych (ang. *named pipes*), to być może musielibyśmy dodać nową bibliotekę. Aby tego dokonać, musielibyśmy przejść do elementu *Protocols*, kliknąć prawym przyciskiem protokół *Named Pipes* i wybrać polecenie *Enable* (włącz).



Rysunek 2.4

W tym momencie mógłbyś zapytać, dlaczego nie włączymy po prostu wszystkich bibliotek NetLib. Chyba dzięki temu nie musielibyśmy się o nic martwić? Niestety w tej sytuacji, podobnie jak w przypadku dodawania czegokolwiek do serwera, efekt jest taki sam — spadek wydajności. W tym przypadku z jednej strony spowolniłoby to serwer (w niewielkim stopniu, ale jednak), a z drugiej mogłoby obniżyć poziom bezpieczeństwa (po co zostawiać otwarte drzwi, jeśli nie zamierzamy ich używać?).

Zobaczmy teraz, które protokoły powinniśmy wybrać i dlaczego.

Named Pipes

Potoki nazwane mogą być użyteczne, gdy niedostępny jest protokół TCP/IP lub nie mamy dostępu do serwera DNS (ang. *Domain Name Server*) pozwalającego na stosowanie nazw serwerów w sieciach TCP/IP.

> Podłączenie się za pomocą adresu IP, a nie nazwy, do systemu SQL Server obsługującego protokół TCP/IP jest technicznie wykonalne. Działa to zawsze, nawet wtedy, gdy nie istnieje usługa DNS — pod warunkiem że istnieje ustalona trasa z klienta do serwera (jeśli ma adres IP, to nie potrzebuje nazwy).

TCP/IP

TCP/IP został uznany za standardowy protokół sieciowy i był używany jako domyślny już w wersji SQL Server 2000. Jest to zresztą jedyna opcja w przypadku łączenia się z systemem SQL Server przez internet, który wykorzystuje przecież tylko IP.

Nie należy mylić dwóch pojęć: czym innym jest udostępnienie serwera baz danych na użytek serwera internetowego, a czym innym umożliwienie bezpośredniego dostępu do serwera baz danych z internetu. Można wystawić w internecie serwer internetowy, który ma dostęp do niedostępnego bezpośrednio z internetu serwera baz danych (żądania do serwera baz danych przychodzące z internetu przechodzą przez serwer internetowy).

Bezpośrednie wystawienie serwera baz danych w internecie stanowi ogromne zagrożenie bezpieczeństwa. Jeśli koniecznie trzeba to zrobić (i istnieją uzasadnione powody takiego działania), to należy zwrócić szczególną uwagę na zapewnienie odpowiednich środków bezpieczeństwa.

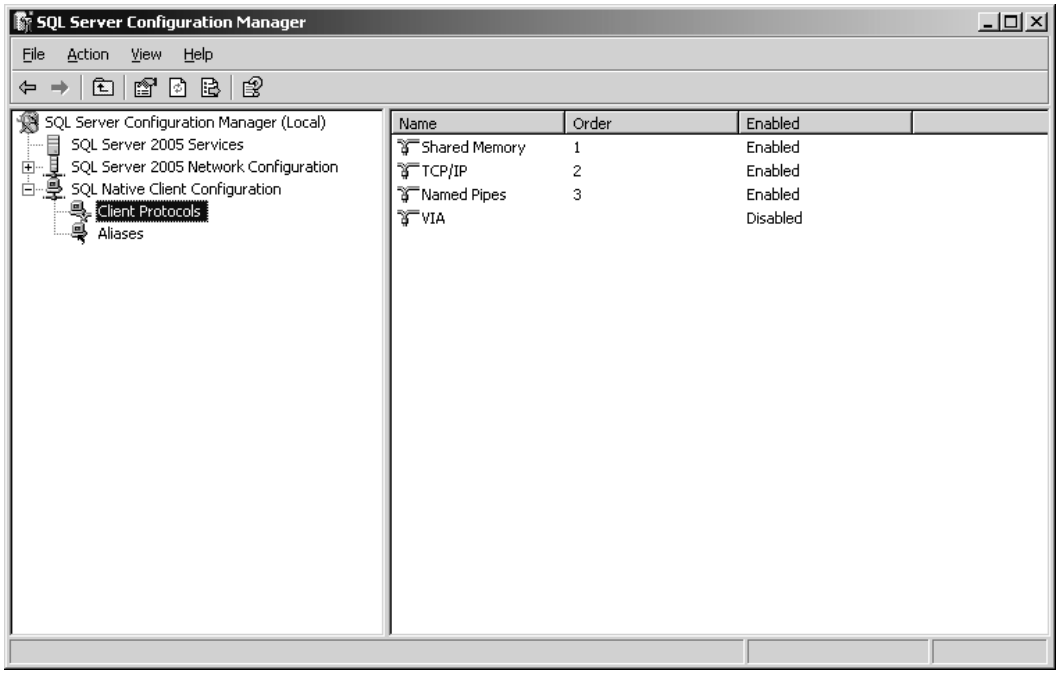
Shared Memory

Pamięć dzielona usuwa konieczność przekazywania parametrów między procesami, co jest sposobem pakowania informacji przed wysłaniem ich przez granice między procesami. Dochodzi do tego w sytuacji, gdy klient i serwer zostały uruchomione na tej samej maszynie. Klient ma bezpośredni dostęp do tego samego odwzorowanego w pamięci pliku, w którym serwer przechowuje dane. Takie rozwiązanie jest *bardzo* szybkie. Można je zastosować tylko w przypadku lokalnego dostępu do serwera (tzn. z serwera internetowego zainstalowanego na tej samej maszynie co serwer baz danych), ale wiąże się z tym duży wzrost wydajności.

Klient

Poznaliśmy już wszystkie dostępne protokoły i dowiedzieliśmy się, który należy wybrać. Kiedy już wiemy, co oferuje nam serwer, możemy przejść do konfiguracji klienta. Ustawienia domyślne z reguły będą działać doskonale. Spójrzmy jednak na posiadane przez nas możliwości. Po rozwinięciu elementu *SQL Native Client Configuration* należy wybrać *Client Protocols*, tak jak pokazano to na rysunku 2.5.

W wersji SQL Server 2005 Microsoft wprowadził możliwość uruchomienia klienta za pomocą jednego protokołu, a w przypadku niepowodzenia tej operacji wykorzystanie innego protokołu. W powyższym oknie najpierw korzystamy z *Shared Memory*, później *TCP/IP*, a następnie *Named Pipes*, w przypadku gdy TCP/IP nie działa zgodnie z definicją podaną w kolumnie *Order*. Jeśli nie zmienimy ustawień domyślnych (np. zmieniając priorytet za pomocą strzałek po wybraniu z menu kontekstowego danego protokołu polecenia *Order*), Shared Memory będzie pierwszą biblioteką NetLib wykorzystaną do połączenia z serwerem, którego nie ma na liście aliasów (następny element w *SQL Native Client Configuration*), kolejną będzie TCP/IP itd.



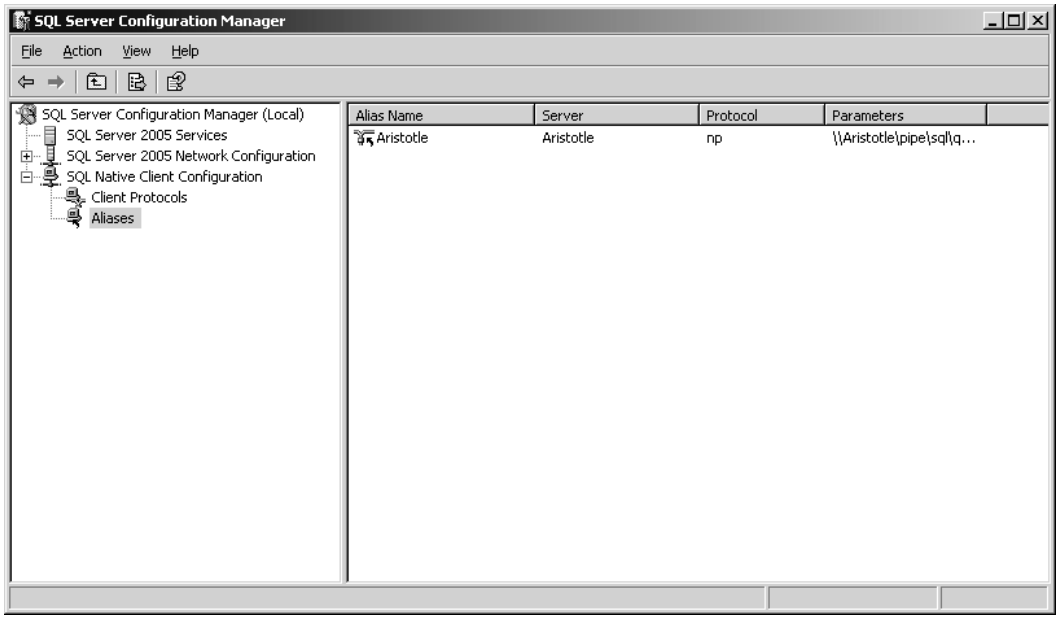
Rysunek 2.5

> Jeśli Twoja sieć obsługuje protokół TCP/IP, to Twój serwer też powinien go obsługiwać. IP wiąże się z mniejszymi narzutami czasowymi i po prostu działa szybciej. Nie ma powodu, dla którego miałbyś go nie wykorzystać. Należy jednak pamiętać, że w przypadku serwerów lokalnych (gdzie serwer i klient znajdują się na tym samym serwerze fizycznym) szybsza będzie biblioteka Shared Memory.

Lista *Aliases* prezentuje wszystkie serwery, dla których zdefiniowano konkretną bibliotekę do realizacji połączeń. Oznacza to, że z jednym serwerem można się kontaktować przy użyciu IP, a z innym korzystając z Named Pipes. Wszystko zależy od wymagań konkretnego serwera. Na rysunku 2.6 pokazano klienta wykorzystującego Named Pipes do komunikacji z serwerem `ARISTOTLE` i używającego dowolnych ustawień domyślnych dla połączeń z innymi serwerami SQL Server.

Jeszcze raz przypominam, że ustawienia w *SQL Native Client Configuration* na maszynie działającej w sieci muszą zapewniać obsługę domyślnego protokołu obsługiwanego przez serwer albo na liście *Aliases* musi istnieć wpis pozwalający na wybór biblioteki NetLib obsługiwanej przez serwer.

Jeśli łączysz się z systemem SQL Server za pośrednictwem internetu (to naprawdę bardzo zły pomysł, ale niektórzy tak robią), to będziesz najprawdopodobniej stosował rzeczywisty adres IP serwera zamiast jego nazwy. Związane jest to z problemem rozwiązywania nazw, który może się pojawić w przypadku pracy z systemem SQL Server w środowisku internetowym. Trzeba pamiętać o konieczności ręcznej zmiany adresu IP serwera, jeśli otrzyma on nowe IP. Nie można w tym względzie polegać na serwerach DNS.



Rysunek 2.6

SQL Server Management Studio

SQL Server Management Studio jest głównym centrum administracji systemu SQL Server. Dostarcza ono szereg funkcji do zarządzania serwerem za pomocą stosunkowo łatwego w użyciu graficznego interfejsu użytkownika.

SQL Server Management Studio to zupełnie nowy moduł, który pojawił się w wersji SQL Server 2005. Środowisko to, przypominające nieco DevStudio IDE, zapewnia niezliczoną ilość funkcji realizowanych dotąd za pomocą osobnych narzędzi.

Tematyka niniejszej książki nie pozwala na omówienie wszystkich zagadnień związanych z SQL Server Management Studio. Dokonamy jednak szybkiego przeglądu jego możliwości:

- Tworzenie, edycja i usuwanie baz danych i obiektów bazy danych.
- Zarządzanie zadaniami, takimi jak wykonywanie kopii zapasowych i uruchamianie pakietów SSIS.
- Wyświetlanie informacji dotyczących bieżącej aktywności, np. zalogowanych użytkowników, blokad obiektów i podłączonych klientów.
- Zarządzanie bezpieczeństwem oraz takimi elementami jak role, loginy oraz zdalne i sprzężone serwery.
- Inicjacja i zarządzanie usługami pocztowymi bazy danych (ang. *Database Mail Service*).

- Tworzenie katalogów wyszukiwania pełnotekstowego i zarządzanie nimi.
- Zarządzanie ustawieniami konfiguracyjnymi serwera.
- Tworzenie i zarządzanie bazami publikatorów i subskrybentów na potrzeby replikacji baz danych.

Ponieważ w tej książce będziemy często używać narzędzia SQL Server Management Studio, przypatrzmy się teraz jego najważniejszym funkcjom.

Rozpoczęcie pracy

Gdy po raz pierwszy uruchamiamy SQL Server Management Studio, wyświetla się okno dialogowe, takie jak zaprezentowane na rysunku 2.7.

Rysunek 2.7



Ekran logowania może być nieco inny niż ten przedstawiony na rysunku i zależy od tego, czy użytkownik logował się wcześniej, na którą maszynę się logował i jakiego loginu używał. Chociaż większość opcji na ekranie logowania nie wymaga wyjaśnień, to omówimy kilka z nich.

Server Type

Typ serwera służy do wyboru podsystemu SQL Servera, do którego użytkownik chce się zalogować (zwykły serwer baz danych, Analysis Services, Reporting Services, SQL Server Mobile lub Integration Services). Należy upewnić się, że logujemy się do właściwego serwera, ponieważ wszystkie te rodzaje „serwerów" mogą być określane za pomocą tej samej nazwy.

SQL Server

Jak nietrudno zgadnąć, ustala się tu SQL Server, do którego chcemy się zalogować. W naszym przykładzie wybraliśmy serwer SCHWEITZER pracujący lokalnie. Chcemy się po prostu zalogować do domyślnej instancji systemu SQL Server działającej na tej samej maszynie, niezależnie od nazwy tej maszyny.

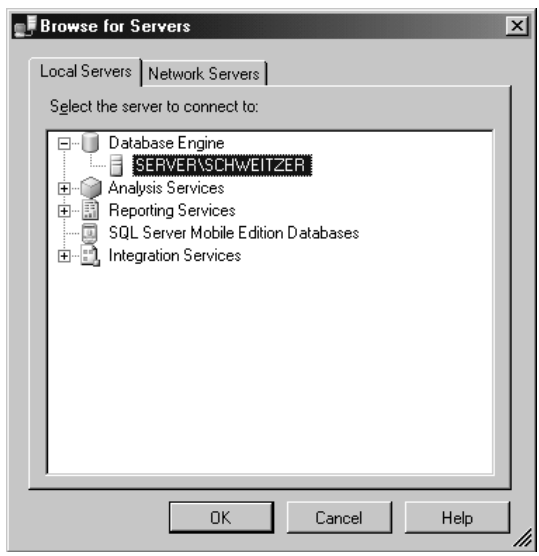
> SQL Server pozwala na jednoczesne działanie wielu „instancji" systemu SQL Server. Dzieje się tak dzięki ładowaniu do pamięci osobnych i niezależnych od siebie silników SQL Server.

Domyślna instancja serwera ma taką samą nazwę jak maszyna działająca w sieci. Oczywiście można zmienić nazwę serwera po instalacji, ale jest to co najmniej problematyczne, a czasami prowadzi do awarii serwera. Dodatkowe instancje serwera SQL będą nosić takie same nazwy jak instancja domyślna (w tej książce najczęściej SCHWEITZER lub ARISTOTLE). Dodatkowo w nazwie znajdzie się znak dolara i nazwa instancji, np. ARISTOTLE$POMPEII.

Serwer lokalny będzie korzystał z biblioteki Shared Memory, niezależnie od tego, którą bibliotekę wybierzemy do kontaktowania się z innymi serwerami. Ma to swoje dobre i złe strony. Niestety tracimy w ten sposób kontrolę (niezależnie od naszego wyboru SQL Server zawsze będzie używał Shared Memory). Z drugiej strony nie trzeba pamiętać, na którym serwerze obecnie się pracuje, a ponadto w związku z pracą na lokalnej maszynie wzrasta wydajność. Jeśli będziemy korzystać z rzeczywistej nazwy serwera, to komunikacja będzie się odbywać poprzez stos sieciowy. Chociaż praca nadal przebiega na tej samej fizycznej maszynie, spada wydajność, podobnie jak w przypadku komunikowania się z innym systemem.

A jeśli nie znamy nazwy serwera? Aby zobaczyć serwery, z którymi się ostatnio łączono, trzeba po prostu kliknąć strzałkę na liście rozwijanej serwerów, przewinąć listę i odnaleźć opcję *<Browse for more...>* (przeglądaj…). Po wybraniu tej opcji SQL Server zacznie odpytywać sieć w poszukiwaniu rozgłaszających się serwerów — jest to metoda, którą serwery wykorzystują, aby powiadamiać inne systemy o swojej obecności w sieci. Na rysunku 2.8 widać dwie zakładki: jedna pokazuje lokalne serwery (wszystkie instancje systemu SQL Server działające na maszynie, na której pracujemy), a druga wyświetla systemy SQL Server w sieci.

Rysunek 2.8



Wybierz serwer z jednej z zakładek i kliknij przycisk *OK*.

Korzystając z okna wyboru serwera, należy zachować ostrożność. Można bowiem tak skonfigurować serwer SQL, aby nie rozgłaszał swojej obecności w sieci. Jeśli serwer został tak właśnie skonfigurowany, to nie pojawi się na liście serwerów. Nie pojawią się tam również serwery, które prowadzą nasłuch z wykorzystaniem biblioteki sieciowej TCP/IP, ale nie posiadają wpisów DNS. W takim przypadku trzeba po prostu znać i zastosować adres IP serwera, z którym musimy się połączyć.

Authentication Type

Do wyboru mamy uwierzytelnianie Windows (dawniej NT) i SQL Server. Uwierzytelnianie Windows będzie dostępne zawsze, nawet jeśli serwer skonfigurowano do korzystania z uwierzytelniania SQL Server. System zaakceptuje loginy wykorzystujące nazwy użytkownika i hasła używane lokalnie w systemie SQL Server (który nie jest częścią szerszej sieci Windows) tylko wtedy, gdy włączono uwierzytelnianie SQL Server.

Windows Authentication

Uwierzytelnianie Windows — czym jest, każdy widzi. Mamy użytkowników i grupy Windows 2000+. W profilach użytkowników systemu Windows zamieszczono odnośniki kont użytkowników Windowsa do loginów w systemie SQL Server. Kiedy użytkownik loguje się do serwera SQL, jest on uwierzytelniany przez domenę Windows, a następnie przypisywany do odpowiednich ról na podstawie loginu systemu SQL Server. Role te określają prawa użytkownika.

Najbardziej istotne w tym modelu jest to, że użytkownik korzysta tylko z jednego hasła (jeśli zmienisz je w domenie Windows, to zostanie ono zmienione również w odniesieniu do loginów systemu SQL Server). Nie trzeba wypełniać żadnych pól, aby się zalogować. Do logowania służą po prostu informacje dostarczone podczas logowania się do sieci Windows. Ponadto administrator może zarządzać wszystkimi użytkownikami w jednym miejscu. Wadą jest to, że proces tworzenia odnośników może się skomplikować, a ponadto w celu administracji kontami użytkowników Windowsa musisz być administratorem domeny.

SQL Server Authentication

Bezpieczeństwo nie zależy od uprawnień użytkownika w sieci, ale od przyznanych mu uprawnień w systemie SQL Server. Proces uwierzytelniania nie korzysta z loginów używanych do logowania w sieci. Aby się zalogować, użytkownik musi podać login i hasło zdefiniowane w systemie SQL Server.

Ma to tę zaletę, że administrator serwera SQL nie musi być administratorem domeny (nie musi nawet posiadać konta w tej domenie), a mimo to może nadawać użytkownikom prawa do korzystania z systemu SQL Server. Proces ten jest też nieco prostszy niż w przypadku uwierzytelniania Windows. Oznacza to również, że jeden użytkownik może korzystać z kilku loginów, z których każdy daje mu inne prawa do przeprowadzania różnych operacji.

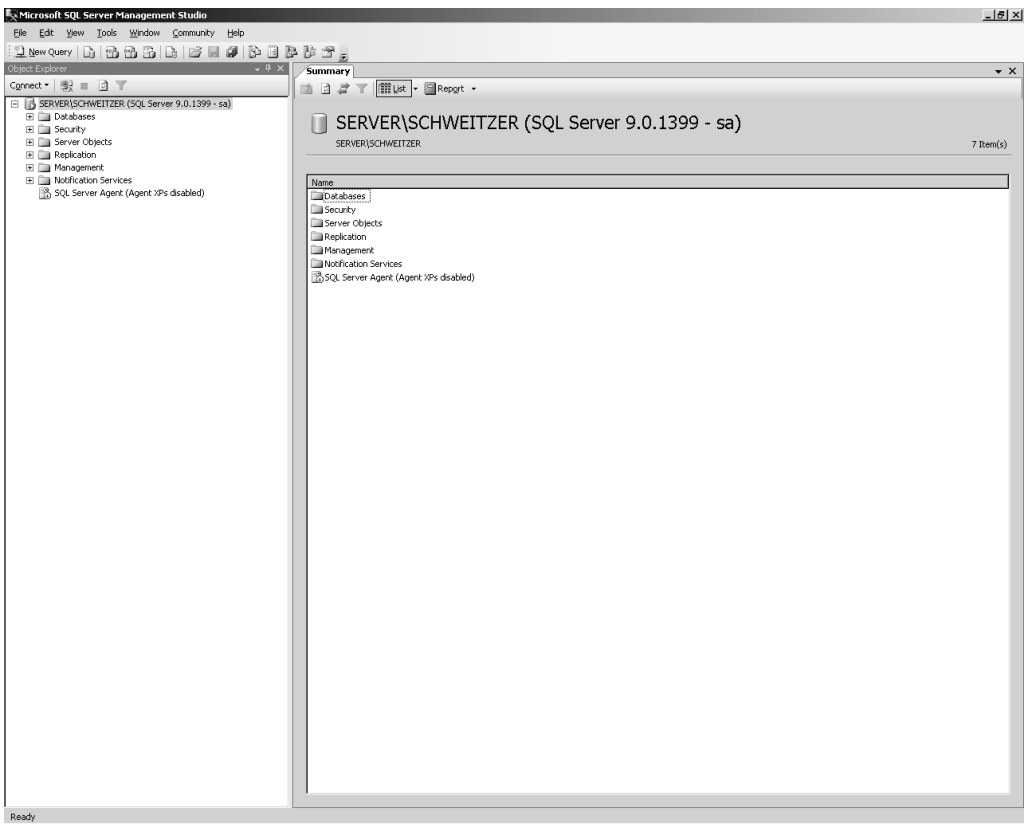
spróbuj sam Nawiązywanie połączenia

Zalogujmy się. Jeśli uruchamiasz SQL Server po raz pierwszy, musisz ustawić wszystko tak jak w naszym przykładzie.

1. Wybierz uwierzytelnianie *SQL Server Authentication*.
2. Wybierz login *sa*, który jest skrótem od *system administrator* (administrator systemu), i zapamiętaj go. Możesz też zalogować się jako inny użytkownik, pod warunkiem że posiada on prawa administratora systemu.
3. Wprowadź hasło użytkownika *sa*, które zostało ustawione podczas instalacji systemu SQL Server. W serwerach rozróżniających wielkie i małe litery (ang. *case-sensitive*) należy wprowadzić login, używając małych liter.

Jeśli łączysz się z serwerem zainstalowanym przez kogoś innego lub zmieniłeś ustawienia domyślne, to będziesz musiał wprowadzić informacje logowania odpowiadające tym zmianom. Po kliknięciu przycisku *OK* powinieneś zobaczyć okno przedstawione na rysunku 2.9.

> Należy zwrócić uwagę na hasło użytkownika *sa*. Zarówno ten, jak i inni użytkownicy *sysadmin* to tzw. superużytkownicy, którzy mają pełny dostęp do całej bazy.



Rysunek 2.9

Jak to działa

Okno dialogowe *Connect to Server* pobiera wszystkie informacje konieczne do utworzenia połączenia. Następnie łączy te informacje w jeden *łańcuch połączenia* (ang. *connection string*), który zostaje wysłany do serwera. Połączenie zostaje przyjęte bądź odrzucone. Jeśli zostanie przyjęte, to uchwyt (ang. *handle*) połączenia przekazany będzie do bazy danych, co pozwala na wielokrotne wykorzystanie tego połączenia dla realizacji dowolnej ilości zapytań aż do momentu odłączenia od serwera.

Więcej informacji na temat tworzenia i formatowania łańcuchów połączenia można znaleźć w dodatku D.

Wiele elementów, z którymi możemy mieć tutaj do czynienia (*Nowy*, *Otwórz*, *Zapisz*, *Wytnij*, *Wklej* itd.), występuje w aplikacjach Windows i powinieneś je dobrze znać. Niektóre jednak są charakterystyczne tylko dla systemu SQL Server. Teraz należy jedynie zwrócić uwagę na to, że układ menu w SQL Server Management Studio zależy od kontekstu. Oznacza to, że dostępność i zawartość poszczególnych elementów będzie się zmieniać w zależności od tego, w jakim oknie SQL Server Management Studio aktualnie pracujemy. Sprawdź, jak zmienia się zawartość menu kontekstowych podczas pracy z różnymi częściami SQL Server Management Studio.

Okno zapytań

Ta część SQL Server Management Studio zastępuje oddzielne narzędzie, które w poprzednich wersjach występowało pod nazwą *Query Analizer*. Narzędzie to służy do prowadzenia interaktywnych sesji z konkretnym serwerem SQL. To właśnie tutaj można wykonywać polecenia za pomocą języka *Transact-SQL* (T-SQL). Z uporem wymawiam to jak ti-eskjuel, choć powinno się raczej mówić ti-sikuel. T-SQL to natywny język systemu SQL Server. Jest on dialektem strukturalnego języka zapytań SQL (ang. *Structured Query Language*), zgodnym ze standardem ANSI SQL 92 na poziomie wejściowym. Zgodność na poziomie wejściowym oznacza, że SQL Server spełnia wymagania pierwszego poziomu, co jest konieczne, aby zakwalifikować produkt jako zgodny ze standardem ANSI. Jak się przekonasz, większość systemów RDBMS jest zgodna ze standardem ANSI na poziomie wejściowym.

Szczerze mówiąc, nie jestem jakoś szczególnie oczarowany tym nowym narzędziem. Wydaje mi się, że w związku z ilością zadań wykonywanych za pomocą tego narzędzia w interfejsie użytkownika panuje spory bałagan i czasami ciężko jest znaleźć to, czego się szuka. Z drugiej strony Microsoft ma nadzieję, że będzie ono bardziej intuicyjne w użyciu z punktu widzenia nowych użytkowników.

Przyjrzyjmy się teraz bliżej narzędziu *Query window*, przy którym spędzimy sporo czasu podczas lektury tej książki, i zobaczmy, jak należy się nim posługiwać.

Rozpoczęcie pracy

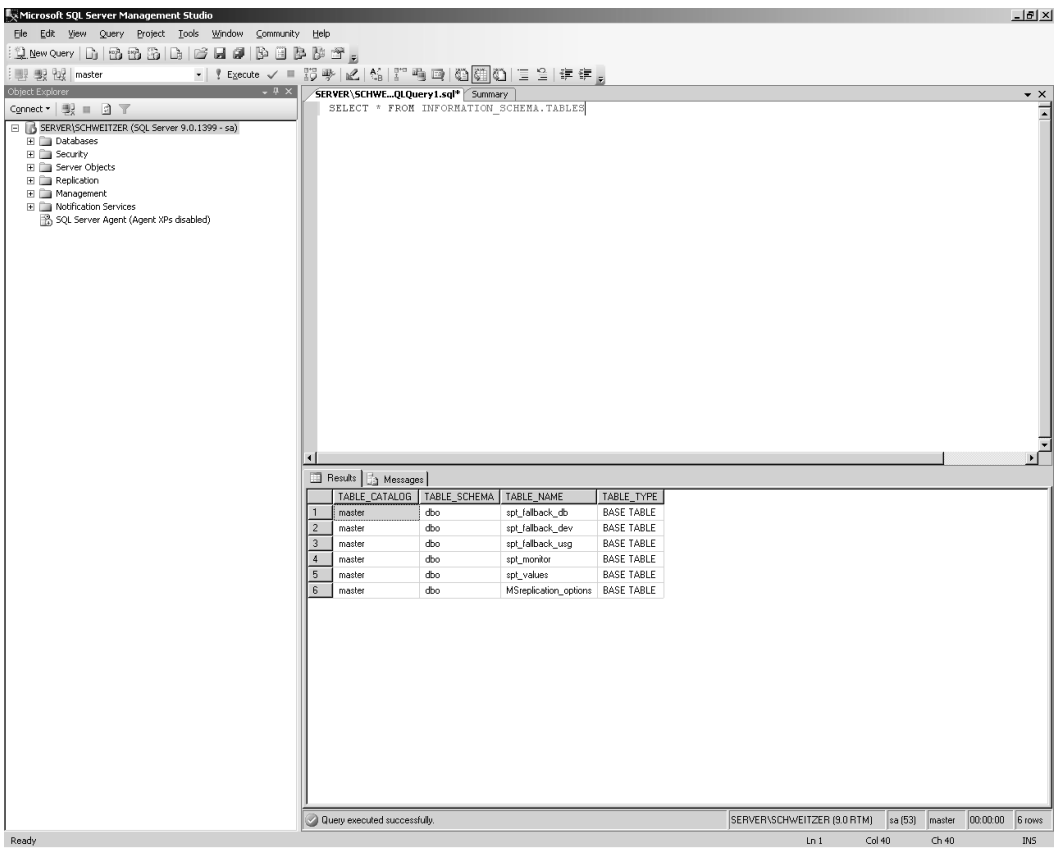
Sporo już powiedziałem o tym, co pojawi się na kartach tej książki. Najwyższy czas przejść do działania. W tym celu otwórzmy nowe okno zapytań, klikając przycisk *New Query* lub wybierając polecenie *File/New/Query With Current Connection*. Po otwarciu okna zapytań wyświetlone zostaną elementy menu podobne do tych, które znajdowały się w starym narzędziu Query Analizer. Za chwilę przyjrzymy się szczegółom, ale teraz zabierzmy się za nasze pierwsze zapytanie.

W głównym oknie zapytań wpisz następujący kod:

```
SELECT * FROM INFORMATION_SCHEMA.TABLES
```

Podczas wpisywania słów i fraz w oknie zwróć uwagę na ich kolory. Słowa kluczowe powinny zostać zaznaczone kolorem niebieskim. Elementy niezidentyfikowane, takie jak nazwy kolumn i tabel (tabele w poszczególnych bazach danych i na różnych serwerach mogą mieć odmienne nazwy), powinny być czarne, ale tabele systemowe będą zielone, łańcuchy znakowe — czerwone, funkcje systemowe — różowe, operatory — szare, zaś komentarze — zielone. Przyporządkowanie kolorów można zmienić za pomocą polecenia *Tools/Options/Environment/Fonts and Colors*. W tym celu z listy *Show settings for* należy wybrać pozycję *Text Editor*. Schematy te należy poznać i nauczyć się nimi posługiwać. Pomagają one bowiem wyłowić błędy jeszcze przed uruchomieniem zapytania (i wyświetleniem komunikatu o błędzie). Polecenie *Query/Parse* uruchamia kolejne narzędzie debugowania, które analizuje zapytanie, nie wykonując go. Jeśli w zapytaniu znajdują się błędy składniowe, narzędzie to powinno je odnaleźć. Innym narzędziem służącym do wyszukiwania błędów jest debugger. Zagadnienia te omówimy w rozdziale 12., zatytułowanym „Procedury składowane".

W celu wykonania zapytania musisz kliknąć teraz znajdujący się na pasku narzędzi przycisk z czerwonym wykrzyknikiem. Okno zapytań zmieni nieco wygląd (patrz: rysunek 2.10).

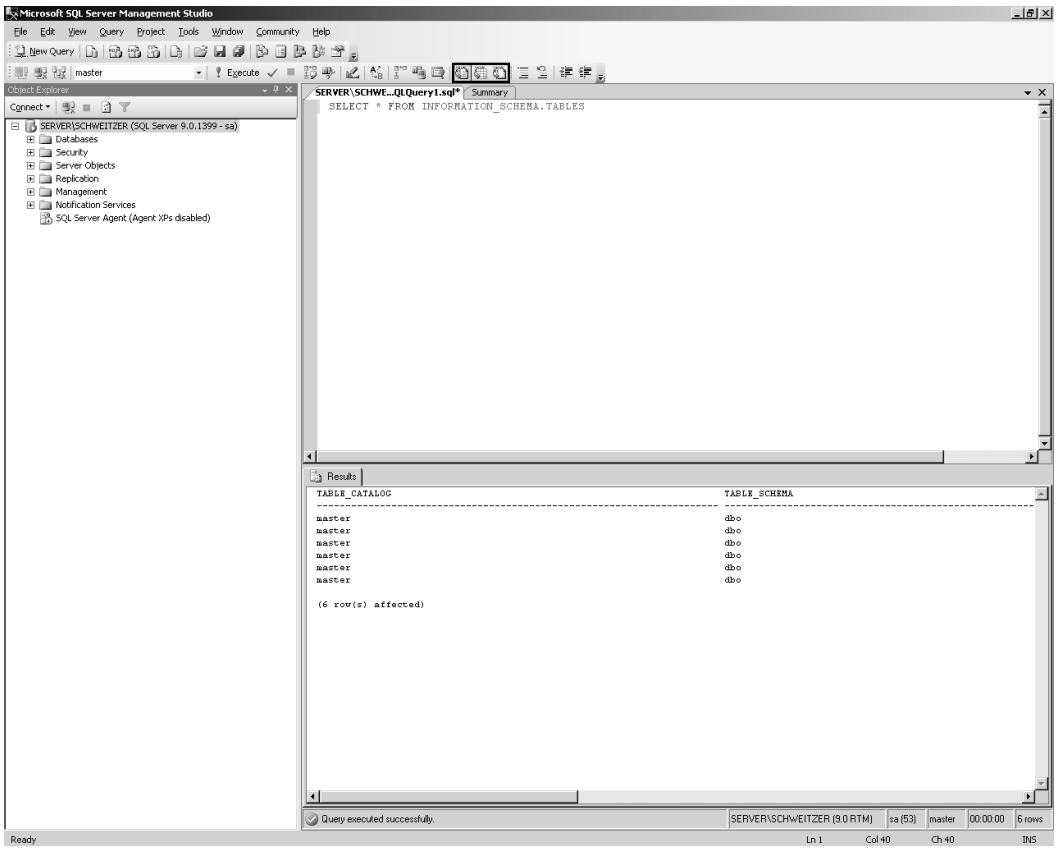


Rysunek 2.10

Okno główne zostało automatycznie podzielone na dwie części. W górnej znajduje się tekst zapytania, dolna nosi nazwę *panelu wynikowego*. Ponadto u góry panelu wynikowego znajduje się zakładka. Kiedy będziemy wykonywać zapytania zwracające kilka zbiorów danych, zobaczysz, że każdy z tych zbiorów można będzie wyświetlić na osobnej zakładce. Jest to dość wygodne w użyciu, zwłaszcza gdy nie wiadomo, jak duży jest zbiór danych lub zbiór wyników (ang. *result set*).

> Terminów „zbiór wyników" (ang. *result set*) i „zbiór rekordów" (ang. *recordset*) używa się często w odniesieniu do zbioru danych, który został otrzymany w wyniku wykonania jakiegoś polecenia. Oba słowa można stosować zamiennie.

Pozmieniaj teraz ustawienia i zobacz, jakie pojawią się różnice. Spójrz na pasek narzędzi nad oknem zapytań i zwróć uwagę na zestaw trzech ikon, które zaznaczono na rysunku 2.11.



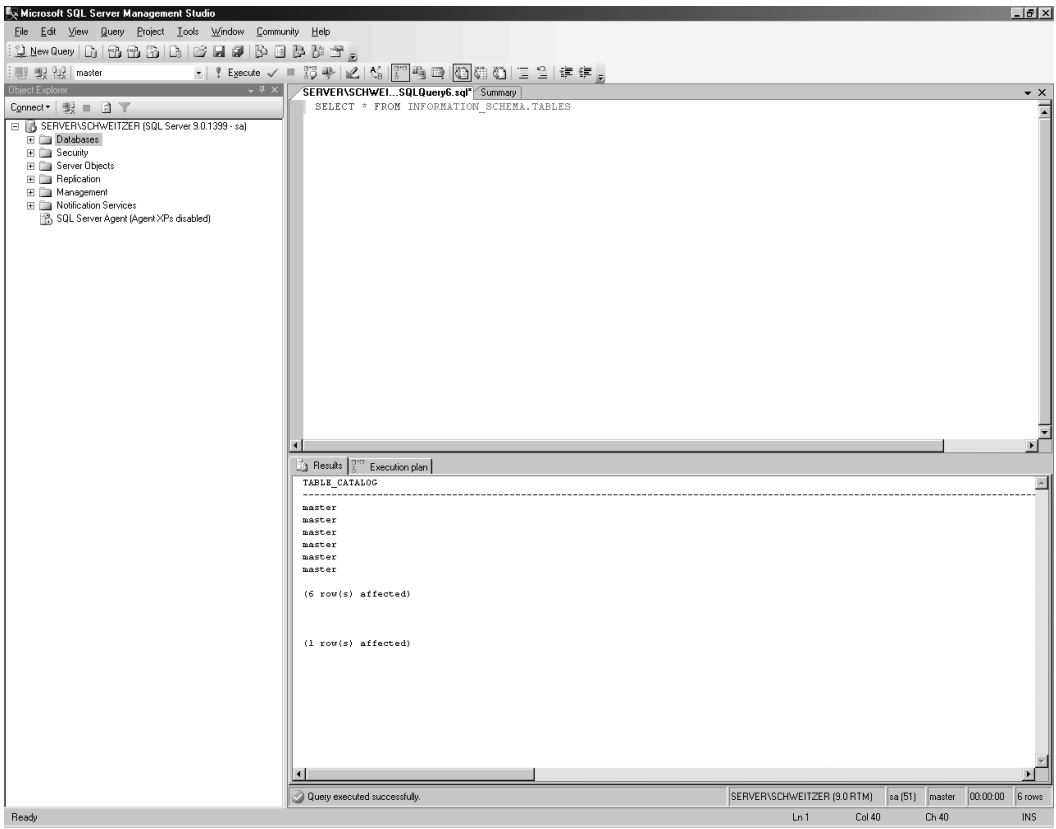
Rysunek 2.11

Kontrolki te pozwalają na określenie sposobu prezentacji wyników zapytania — mogą to być tekst, siatka lub plik. Ustawienia te można zmieniać również za pomocą polecenia *Results To* znajdującego się w menu *Query*.

Wyniki jako tekst

Opcja *Results to Text* pobiera wszystkie wyniki zapytania i umieszcza je na jednej stronie w formacie tekstowym. Długość tej strony może być niemal nieskończona (ogranicza ją ilość dostępnej w systemie pamięci).

Zanim przejdziemy dalej, wykonaj ponownie nasze poprzednie zapytanie i zobacz, co się stało. Wybierz opcję *Results to Text*, kliknij ikonę z zieloną strzałką *Include Actual Execution Plan* lub wybierz to polecenie z menu *Query* i wykonaj zapytanie jeszcze raz. Efekt pokazano na rysunku 2.12.



Rysunek 2.12

Dane wynikowe na zakładce *Results* są dokładnie takie same jak poprzednio. Pojawiły się tylko w innym formacie. Dodatkowo na zakładce *Execution plan* sprawdzić można koszty uzyskania wyników. Osobiście używam metody prezentacji wyników jako tekst, kiedy:

- Otrzymuję tylko jeden zbiór wyników, zaś wyniki mają dość wąskie kolumny.
- Chcę zapisać moje wyniki w pojedynczym pliku tekstowym.
- Otrzymam kilka niewielkich zbiorów wyników, więc chcę zobaczyć wszystkie na jednej stronie bez konieczności borykania się z kilkoma paskami przewijania.

Wyniki w siatce

Opcja *Results to Grid* dokonuje podziału kolumn i wierszy w widoku siatki. Poniżej zamieszczam listę możliwości, na które pozwala ta opcja:

- Zmiana rozmiaru kolumn poprzez umieszczenie wskaźnika myszy nad prawą krawędzią nagłówka kolumny, a następnie kliknięcie i przeciągnięcie krawędzi kolumny do nowego rozmiaru. Klikając dwukrotnie krawędź kolumny, można dokonać automatycznego dostosowania rozmiaru.
- Po zaznaczeniu kilku komórek, wycięciu ich i wklejeniu do innej siatki (np. Microsoft Excel) będą one traktowane jako osobne komórki (gdybyśmy korzystali z opcji *Results to Text*, wycięte dane zostałyby wklejone do jednej komórki).
- Można zaznaczyć kilka kolumn wyników (kiedy w *Results to Text* zaznaczymy kilka wierszy, zaznaczone zostaną kolumny wszystkich wierszy wewnętrznych; zaznaczenia można dokonać tylko w środku pierwszego i ostatniego wiersza).

Ponieważ zazwyczaj potrzebuję jednej z powyżej opisanych możliwości, najczęściej korzystam z tej właśnie opcji.

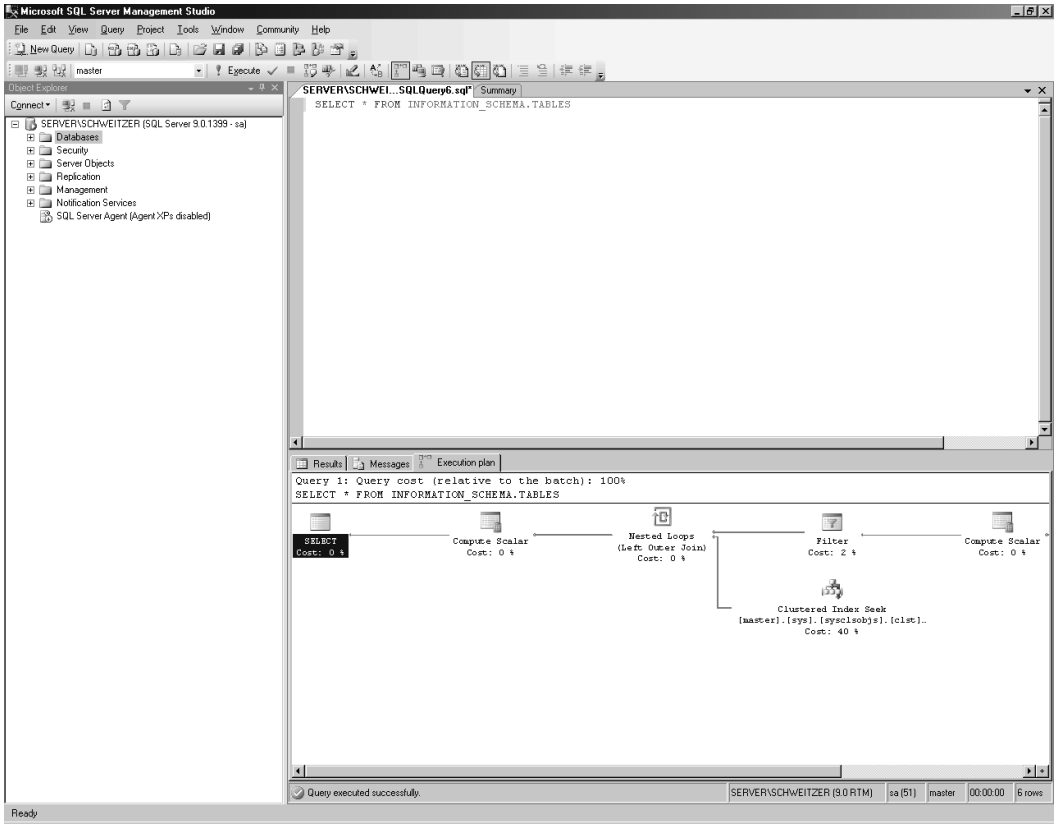
Plan wykonania zapytania

Przy każdym uruchomieniu zapytania SQL Server analizuje to zapytanie i wysyła je do optymalizatora zapytań (ang. *query optimizer*). Optymalizator zapytań to część systemu SQL Server ustalająca taki sposób wykonania zapytania, aby umożliwić szybkie otrzymanie wyników i w jak najmniejszym stopniu wpłynąć na pracę pozostałych użytkowników. Korzystając z opcji *Display Estimated Execution Plan*, otrzymamy graficzny obraz oraz dodatkowe informacje dotyczące tego, jak SQL Server zamierza wykonać zapytanie. Można też włączyć opcję *Include Actual Execution Plan* (dołącz rzeczywisty plan wykonania). Zazwyczaj nie będzie różnic między szacowanym a rzeczywistym planem wykonania. Czasami jednak różnice takie pojawiają się w związku ze zmianami dokonywanymi przez optymalizator podczas wykonywania zapytania i rzeczywistym kosztem wykonania zapytania, odmiennym od tego, który obliczył optymalizator.

Aby zobaczyć, jak wyglądał plan wykonania naszego prostego zapytania, należy kliknąć opcję *Include Actual Execution Plan* i ponownie wykonać zapytanie. Jeśli chcemy zobaczyć koszt wykonania zapytania, musimy kliknąć zakładkę *Execution Plan*. Pokazano to na rysunku 2.13.

Zwróćmy też uwagę, że wyniki zapytania są nadal wyświetlane w wybrany przez nas sposób. Wyniki działania opcji *Display Estimated Execution Plan* i *Include Actual Execution Plan* są niemalże identyczne, za wyjątkiem dwóch różnic:

- Otrzymujemy plan od razu, a nie dopiero po wykonaniu zapytania.
- Chociaż widzimy plan, to wszystkie informacje dotyczące kosztów są wynikiem szacunków, a nie rzeczywistego wykonania zapytania. Kiedy uruchamiamy zapytanie z opcją *Include Actual Execution Plan*, zapytanie jest fizycznie wykonywane i informacje dotyczące kosztów są rzeczywiste, a nie szacowane.



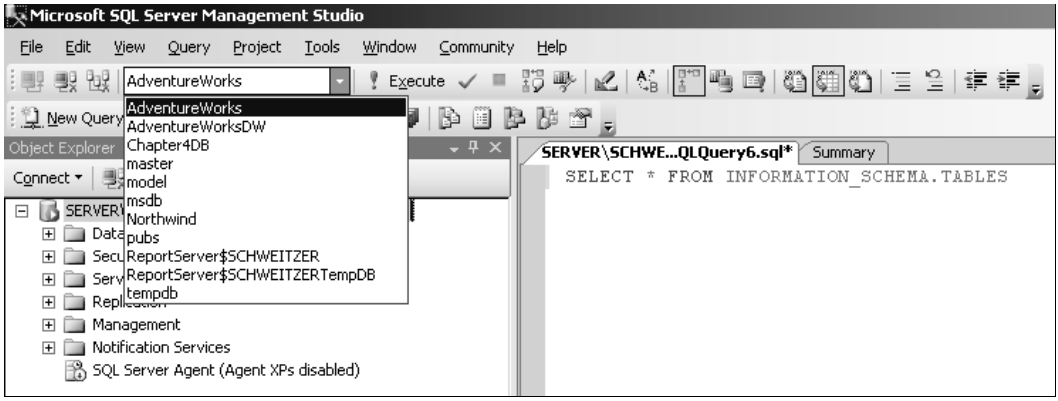
Rysunek 2.13

Lista wyboru bazy danych

Przejdźmy teraz do omówienia listy wyboru bazy danych. Służy ona do wyboru domyślnej bazy danych, na której będą operowały zapytania wykonywane w bieżącym oknie. W momencie uruchomienia okna zapytań domyślna baza to ta, do której zalogował się użytkownik (jeśli niczego nie zmieniono w systemie, to w przypadku użytkownika `sa` jest to baza `master`). Następnie można przejść do innej bazy, o ile wykorzystywany login ma do niej prawa dostępu. Ponieważ korzystamy z identyfikatora użytkownika `sa`, na liście baz danych powinny pojawić się wszystkie bazy danych, jakie znajdują się na bieżącym serwerze.

Zmieńmy bazę na `AdventureWorks` i ponownie wykonajmy poprzednie zapytanie, tak jak pokazano to na rysunku 2.14.

Jak zobaczysz, przedstawione dane pochodzą właśnie z wybranej bazy.



Rysunek 2.14

Okno Object Explorer

Okno *Object Explorer* (eksplorator obiektów) to małe, ale użyteczne narzędzie, które pozwala przemieszczać się w bazie danych, sprawdzać nazwy obiektów, a nawet przeprowadzać operacje, takie jak pisanie skryptów i przeglądanie danych.

Elementy listy można rozwijać aż do momentu, kiedy dojdziemy do listy tabel znajdujących się w wybranej bazie, np. `AdventureWorks`. Można posunąć się jeszcze dalej i przeglądać poszczególne kolumny, typy danych i inne właściwości. To bardzo poręczne narzędzie do przeglądania baz danych.

SQL Server Integration Services (SSIS)

SSIS (wcześniej znane jako *Data Transformation Services* lub *DTS*) to nasz przyjaciel. Za każdym razem, gdy spoglądam na tę funkcję systemu SQL Server, muszę usiąść z wrażenia. Aby nakreślić szersze tło, powiem, że na przestrzeni lat zrealizowałem kilka projektów systemów wspomagania decyzji (zazwyczaj są to systemy, w których nie mamy do czynienia z danymi wchodzącymi i wychodzącymi na bieżąco, ale raczej z pobieraniem danych z innych źródeł w celu wspomagania kierownictwa w podejmowaniu decyzji). Projekt taki pobiera dane z różnych źródeł i umieszcza je w centralnej bazie danych wykorzystywanej do scentralizowanego raportowania.

Tego typu projekty szybko stają się coraz droższe z uwagi na fakt, że nie każdy system tak samo definiuje dane. Może się pojawić cała masa problemów, które trzeba rozwiązać. Mogą one dotyczyć integralności danych (co zrobić, gdy w polu występuje wartość `NULL`, a my nie zezwalamy na występowanie takich wartości?) lub różnic w regułach biznesowych (jeden system rozwiązuje problemy związane z udzielaniem pożyczek, umożliwiając wprowadzenie ujemnej liczby zamówionych towarów, a inny na to nie pozwala). Ta lista może stawać się coraz dłuższa, a koszt coraz wyższy.

Dzięki zastosowaniu SSIS usunięto lub przynajmniej wyeliminowano konieczność tworzenia ogromnych ilości kodu pisanego zazwyczaj w języku obsługiwanym po stronie klienta. Kod ten musiał powstawać, aby obsłużyć tego typu sytuacje. SSIS pozwala pobrać dane z dowolnego źródła, takiego jak OLE DB lub .NET, i umieścić je w tabelach systemu SQL Server.

> Należy pamiętać, że istnieje specjalny dostawca OLE DB dla ODBC. Dostawca ten pozwala zamapować dostęp z OLE DB bezpośrednio do sterownika ODBC. Oznacza to, że za pomocą OLE DB (a dzięki temu również SSIS) można uzyskać dostęp do ODBC. Pozwala to też na uzyskanie dostępu do OLE DB przez ODBC.
>
> Skoro już przy tym jesteśmy, chciałbym podkreślić, że chociaż SSIS jest częścią systemu SQL Server, to może pracować na dowolnym źródle i celu OLE DB. Oznacza to, że SQL Server nie musi być w ogóle zaangażowany w ten proces i służy jedynie do „pompowania" danych. Można na przykład wysłać dane z bazy Oracle do arkusza Excel, a nawet z bazy DB/2 do MySQL.

Podczas przenoszenia danych możemy zastosować tzw. *transformacje danych*. Transformacje po prostu zmieniają dane zgodnie z pewnymi regułami logicznymi. Przekształcenie takie może być proste, jak np. zmiana nazwy kolumny, lub złożone, jak np. analiza integralności danych i w razie konieczności zastosowanie odpowiednich reguł. Aby sobie to wyobrazić, zastanów się nad podanym przeze mnie przykładem, w którym pobierano dane z pola zawierającego wartości `NULL` i wstawiano je do tabeli, która nie zezwalała na tego typu wartości. Korzystając z SSIS, podczas procesu przenoszenia danych można dokonać automatycznej zamiany wartości `NULL` na dowolnie wybraną wartość (np. zero w przypadku liczby lub „nieznany" w przypadku ciągu znaków).

Bulk Copy Program (bcp)

Jeśli SSIS jest naszym przyjacielem, to Bulk Copy Program (bcp) byłby tym starym przyjacielem, którego niezbyt często widujemy, ale za każdym razem cieszymy się na jego widok.

Narzędzie bcp to program wiersza poleceń, którego jedynym zadaniem jest masowe przenoszenie sformatowanych danych z i do serwera SQL Server. Istniał on na długo przed tym, nim wymyślono SSIS, i chociaż SSIS zastępuje bcp w większości zadań importu lub eksportu danych, to bcp nadal stanowi atrakcyjną alternatywę dla tych, którzy lubią narzędzia wiersza poleceń. Ponadto cała masa instalacji systemu SQL Server nadal polega na programie bcp, kiedy zachodzi konieczność szybkiego przesyłania danych.

SQL Server Profiler

Nie potrafię policzyć, jak często narządzie to ratowało mi skórę, mówiąc mi, co się dzieje z moim serwerem, gdy wszystkie inne narzędzia zawiodły. Nie jest to narzędzie, którym programista (a nawet administrator bazy danych) będzie się posługiwał codziennie. Jest ono jednak nadzwyczaj skuteczne i może Cię uratować, kiedy nic innego nie jest już w stanie pomóc.

SQL Server Profiler (uruchamiany z menu *Tools*) to w wielkim skrócie narzędzie śledzenia w czasie rzeczywistym. Performance Monitor (uruchamiany z menu *Start/Programs/Administrative Tools/Performance*) służy do śledzenia na poziomie makro spraw związanych z konfiguracją systemu. Profiler natomiast zajmuje się szczegółami, co ma swoje dobre i złe strony. W zależności od ustawień Profiler może dostarczyć dokładną składnię każdego zapytania wykonanego na serwerze. Wyobraź sobie, że trzeba przeprowadzić operację dostrajania wydajności w systemie, w którym pracuje tysiąc użytkowników. Łatwo można sobie wyobrazić ryzy papieru zużyte do wydrukowania poleceń wykonanych przez tych użytkowników w przeciągu paru minut. Na szczęście Profiler został wyposażony w szeroką gamę filtrów pomagających nakładać ograniczenia i umożliwiających śledzenie konkretnych problemów, takich jak długo wykonujące się zapytania albo poznanie dokładnej składni zapytania wykonanego wewnątrz procedury wykonywanej (co może być pomocne zwłaszcza wtedy, gdy w procedurze tej występują wyrażenia warunkowe i wykonanie zapytania może przebiegać odmiennie w różnych sytuacjach).

sqlcmd

Narzędzia tego próżno szukać w grupie programów SQL Server. W rzeczy samej to zdumiewające, jak wielu ludzi nie wie nawet o istnieniu tego narzędzia ani jego starszych braciach (osql i isql). Dzieje się tak dlatego, że jest to w zasadzie program konsoli, a nie Windowsa, uruchamiany z systemowego wiersza poleceń (menu *Start/Programs/Accessories/Command Prompt*).

Narzędzia tego należy używać wtedy, gdy w plikach wsadowych wywoływanych z wiersza poleceń chcemy wstawić polecenia SQL lub zadania związane z zarządzaniem. W wersjach wcześniejszych niż 7.0 i przed nadejściem czegoś, co wówczas określano terminem DTS (obecnie SSIS), sqlcmd było używane razem z narzędziem Bulk Copy Program (bcp) do realizowania operacji importu danych z systemów zewnętrznych. Zastosowanie to traci obecnie na znaczeniu, gdyż administratorzy i programiści poznali potęgę i prostotę SSIS. Mimo to nadal zdarzają się przypadki, gdy trzeba napisać skrypt będący częścią większego procesu realizowanego w wierszu poleceń. Możliwość tę zapewnia właśnie sqlcmd.

Narzędzie sqlcmd może być bardzo przydatne, zwłaszcza jeśli korzystasz z plików zawierających skrypty, które można uruchomić w sqlcmd. Należy jednak pamiętać, że istnieją narzędzia, które o wiele lepiej poradzą sobie z zadaniami realizowanymi przez sqlcmd. Narzędzia te posiadają też interfejs użytkownika, który bardziej pasuje do tego, nad czym pracujesz w systemie SQL Server.

> *Nadszedł czas na kolejną radę. Jej celem jest poznanie historii i umożliwienie zrozumienia ludzi, z którymi rozmawiasz o systemie SQL Server. Sqlcmd to kolejna nowa nazwa narzędzia występującego pod różnymi nazwami. Na początku nazywało się ono ISQL, zaś w SQL Server 2000 oraz 7.0 znane było pod nazwą osql.*

Podsumowanie

Nie będziesz codziennie używał większości przedstawionych tutaj narzędzi. Przeciętny programista będzie na co dzień korzystał z SQL Server Management Studio. Mimo to należy posiadać wyobrażenie na temat tego, jakie zadania każde z tych narzędzi może realizować. Każde może zaoferować jakąś ważną funkcjonalność. Na kartach naszej książki spotkamy się jeszcze z tymi narzędziami.

Dostępne są jeszcze inne narzędzia, wykorzystywane głównie w celach administracyjnych, do których nie ma skrótów w menu *Start* (narzędzia łączności oraz diagnostyki i konserwacji serwera).