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 2000 dla każdego



Autorzy: Richard Waymire, Rick Sawtell
Tłumaczenie: Jowita Gmytryk, Jacek Mozdyniewicz
Tytuł oryginału: Teach Yourself Microsoft SQL Server 2000 in 21 Days
ISBN: 83-7197-410-8
Nośnik: CD
Seria: dla każdego
Liczba stron: (około) 850

Poznawanie technik i pojęć z naszą książką sprawi, że błyskawicznie staniesz się ekspertem w wykorzystaniu Microsoft SQL Server 2000. A zatem – wykorzystaj swoją wiedzę w praktyce! W książce tej znajdziesz, co konieczne do skonfigurowania i efektywnej pracy z Microsoft SQL Serverem 2000. Podręcznik zawiera wszelkie potrzebne informacje, poczynając od podstawowych, a kończąc na bardzo zaawansowanych; spróbuj zatem opanować podstawy działania platformy Microsoft SQL Server 2000 oraz poznaj jego wszystkie, najważniejsze możliwości. Przeanalizuj praktyczne przykłady, dzięki którym nauczysz się efektywnie korzystać z najnowszych narzędzi Microsoft SQL Servera 2000. Zapoznaj się z poradami ekspertów w dziedzinie wdrażania SQL Server 2000 w środowiskach korporacyjnych.

Książka została napisana w taki sposób, by sprostać oczekiwaniom większości czytelników pragnących szybko i efektywnie poznać SQL Server 2000. Możesz po kolei czytać rozdziały oraz analizować podane w nich, szczegółowo wyjaśnione przykłady, możesz także wybrać tylko te rozdziały, które Cię interesują. A zatem:

- Dowiedz się, jak w pełni wykorzystać możliwości nowych narzędzi Microsoft SQL Server 2000 służących do zarządzania danymi.
- Rozszerz swoje umiejętności projektowania baz danych, tworząc tabele, procedury zachowane, widoki oraz inne obiekty baz danych.
- Poznaj język T-SQL, którego elastyczność i łatwość wielokrotnego użycia kodu da Ci możliwości, którymi nie dysponują kreatory oraz graficzne narzędzia projektowania.
- Poznaj i wykorzystaj nowe narzędzia Microsoft SQL Server 2000 – Analysis Services oraz technologie English Query.
- Dowiedz się, jak publikować dane w Internecie przy wykorzystaniu języka XML.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

O Autorach 15

Wstęp 17
O tej książce 18
Dla kogo jest ta książka? 20

Rozdział 1. Wprowadzenie do relacyjnych baz danych i Microsoft SQL Server 2000 21
SQL Server to jest to! 21
Historia SQL Servera 22
Co to jest baza danych? 23
Obiekty relacyjnej bazy danych 24
Projektowanie relacyjnych baz danych 25
Rozmowa z klientem 26
Grupowanie obiektów 27
Normalizacja danych 28
Tworzenie bazy danych i tabel 29
SQL Server i model klient-serwer 29

Rozdział 2. Instalacja Microsoft SQL Server 2000 31
Jakie rodzaje serwerów SQL są dostępne? 31
Standard Edition 32
Enterprise Edition 32
Personal Edition 32
Developer Edition 33
Evaluation Edition 33
Microsoft SQL Server 2000 Desktop Edition (MSDE) 33
Wymagania serwera SQL 33
Opcje programowe i sprzętowe 33
Instalacja SQL Servera 2000 37
Rozpoczęcie procesu instalacji 37
Warunki wstępne systemu operacyjnego 38
Instalacja SQL Servera 2000 (120-dniowa wersja Evaluation Edition) 38
Instalacja innych, opcjonalnych składników SQL Servera 55
Microsoft SQL Server Analysis Services 55
Microsoft English Query 55
Po instalacji: co zyskałeś? 56
Usługi SQL Servera 56
Foldery instalacyjne 57

Wpisy w rejestrach systemu Windows ... 58
Domyślne bazy danych SQL Servera ... 60
Domyślne identyfikatory (ID) logowania do serwera SQL ... 61
Kopie jawnie nazwane kontra kopie domyślne ... 62
Kopia domyślna ... 62
Kopia jawnie nazwana ... 63
Składniki wspólne ... 63
Instalacja kopii jawnie nazwanej ... 63
Uaktualnienie z wcześniejszej wersji SQL Servera ... 64
Uaktualnienie z wersji SQL Server 6.5 ... 64
Rozwiązywanie problemów z instalacją ... 65
sqlstp.log ... 65
Pliki z rozszerzeniem *.OUT ... 65
Dziennik błędów SQL Servera ... 65
Dziennik zdarzeń aplikacji Windows NT/2000 ... 66
Usuwanie serwera SQL ... 66

Rozdział 3. Narzędzia SQL Servera 2000 i jego usługi ... 67
Grupa programowa Microsoft SQL Server ... 67
Plik readme.txt ... 68
Books Online ... 68
Service Manager ... 73
Narzędzie Client Network ... 76
Narzędzie Server Network ... 80
Query Analyzer ... 82
Enterprise Manager ... 92
Import and Export Data ... 97
Configure SQL XML Support in IIS ... 97
Profiler ... 97
Grupa programowa Microsoft SQL Server — Switch ... 98
SQL Server Version Upgrade Wizard ... 98
Microsoft SQL Server 6.x Switch ... 98
Uninstal SQL Server 6.x ... 99
Inne narzędzia-usługi SQL Servera ... 99
Narzędzia związane z połączeniami ... 99
Narzędzia diagnostyczne i konserwacyjne serwera ... 104
BCP — jedyny w swojej klasie ... 110

Rozdział 4. Tworzenie baz danych i plików danych oraz administracja nimi ... 111
Tworzenie bazy danych ... 112
Zbieranie informacji o bazie danych ... 118
Ustawianie opcji bazy danych ... 120
Zmiana rozmiaru bazy danych ... 124
Rozszerzenie bazy danych ... 125
Zmniejszanie bazy danych ... 128
Zmiana nazwy bazy danych ... 131
Grupy plików bazy danych ... 132
Usuwanie bazy danych ... 135

Rozdział 5. Ustawienia zabezpieczeń logowania i użytkownika ... 139
Model bezpiecznego dostępu do SQL Servera ... 139
Uwierzytelnianie Windows ... 140
Uwierzytelnianie logowania do SQL Servera ... 142
Nazwa użytkownika bazy danych SQL Servera ... 142
Prawa dostępu ... 142

Tryby zabezpieczeń SQL Servera (z uwzględnieniem logowania) 142
Mixed Mode 143
Windows Authentication Mode 148
Ustawienia Security Mode 154
Konta logowania — zarządzanie graficzne 155
Użytkownicy bazy danych 158
Dodawanie użytkownika do bazy danych 158
Nazwa użytkownika — Guest 161
Dodawanie aliasu 162
Zmiana właściciela bazy danych 163
Role 164
Rola Public 164
Role o zasięgu serwera 164
Role bazy danych 166
Role aplikacji 171

Rozdział 6. Ustawianie uprawnień bazy danych 175
Potrzeba używania uprawnień 175
Implementacja uprawnień w bazie danych 176
Typy uprawnień 177
Pierwszeństwo uprawnień 177
Specjalne uprawnienia SQL Servera 177
Role stałe i związane z nimi uprawnienia 178
sa 181
Ustalone role bazy danych 181
Właściciel bazy danych (dbo) 184
Uprawnienia właściciela obiektów bazy danych (dboo) 185
Uprawnienia użytkownika 187
Rola public 187
Instrukcje uprawnień 188
Uprawnienie CREATE DATABASE 189
Uprawnienia CREATE TABLE, VIEW FUNCTION, PROCEDURE, DEFAULT i RULE 189
Uprawnienie polecenia BACKUP DATABASE i BACKUP LOG 189
Przydzielanie uprawnień polecenia 190
Administracja uprawnieniami polecenia za pomocą SQL Server Enterprise Managera 192
Uprawnienia obiektu 193
Przyznawanie uprawnień obiektu 194
Uprawnienia dotyczące widoków, procedur składowych i funkcji 199
Uprawnienia do widoków 200
Uprawnienia do procedur składowych 201
Uprawnienia do funkcji zdefiniowanych przez użytkownika 201
Łańcuchy własności 202
Łańcuch pojedynczego właściciela 202
Przerwane łańcuchy własności 202
Projektowanie strategii uprawnień 204
Najlepsze działania 204
Wskazówki „należy... nie należy...” 204
Generacja skryptów bezpieczeństwa 206

Rozdział 7. Tworzenie kopii bezpieczeństwa baz danych 209
Ochrona danych za pomocą lustrzanego odbicia, dupleksowania i paskowania 210
RAID 211

Sposób działania kopii bezpieczeństwa 214
Typy kopii bezpieczeństwa 215
Transakcje i punkty kontrolne 216
Inne strategie tworzenia kopii bezpieczeństwa 217
Terminologia związana z kopiami bezpieczeństwa 217
Rozważania dotyczące tworzenia kopii bezpieczeństwa 218
Kto odpowiada za archiwizację danych? 218
Jak będzie sprawdzana poprawność utworzonych kopii bezpieczeństwa? 218
Które bazy danych należy archiwizować? 219
Gdzie będą przechowywane kopie bezpieczeństwa? 219
Jak często należy tworzyć kopie bezpieczeństwa baz danych? 220
Jak długo należy przechowywać kopie bezpieczeństwa? 220
Ile czasu zajmuje odtwarzanie danych z kopii bezpieczeństwa? 220
Czy mam plan odzyskiwania danych po awarii? 220
Wybór taśm 222
Tworzenie kopii bezpieczeństwa baz użytkownika 222
Modele odtwarzania baz danych i kopie bezpieczeństwa 223
Tworzenie kopii bezpieczeństwa systemowych baz danych 225
Baza danych master 225
Baza danych MSDB 226
Baza danych model 226
Baza danych tempdb 227
Dystrybucyjna baza danych 227
Przygotowanie do implementacji archiwizacji 227
Tworzenie narzędzia archiwizacyjnego 227
Sprawdzanie spójności bazy danych 232
Tworzenie kopii bezpieczeństwa bazy danych 235
Opcje nośnika archiwizacyjnego 236
Polecenie BACKUP DATABASE dla całej bazy danych 237
Polecenie BACKUP:DATABASE dla plików i grup plików 241
Polecenie BACKUP LOG 242
Tworzenie kopii bezpieczeństwa z użyciem SQL Server Enterprise Managera 244

Rozdział 8. Odtwarzanie baz danych 249
Odtwarzanie bazy danych 249
Odzyskiwanie automatyczne 250
Odzyskiwanie ręczne 253
Scenariusze odtwarzania 271
Odtwarzanie po utracie dysku 271
Odtwarzanie po utracie bazy danych master 273

Rozdział 9. Typy danych i tworzenie tabel 277
Tabele 277
Kolumny 278
Identyfikatory SQL Servera 278
Typy danych 280
Wartość NULL 281
Sposób kodowania 281
Zmienne tekstowe 281
Dane binarne 282
Liczby całkowite 283
Przybliżone i dokładne liczbowe typy danych 284
Typy danych liczbowych przybliżonych 284
Dokładne typy danych liczbowych 285

Specjalne typy danych....286
bit....286
text i image....287
RowGUID....287
sql_variant....288
table....289
Typy danych daty i czasu....289
Waluta....291
Typ danych automatycznie wzrastający rowversion (timestamp)....292
Praca z ANSI i Unicode....292
Tworzenie typów danych zdefiniowanych przez użytkownika....293
Kolumny wyliczeniowe....295
Tworzenie tabel....296
Używanie wyrażenia CREATE TABLE....296
Tworzenie tabel za pomocą SQL Server Enterprise Managera....297
Usuwanie tabel....298

Rozdział 10. Pobieranie danych za pomocą wyrażenia SELECT....301

Proste polecenia SELECT....301
Zmiana nagłówków kolumn....303
Używanie literałów....305
Manipulacja danymi....305
Operatory arytmetyczne....305
Funkcje matematyczne....307
Funkcje ciągu znaków....308
Funkcje daty....310
Funkcje systemowe....312
Konwersja danych....312
Wybór wierszy....315
Operatory porównania....315
Zakresy....316
Listy....317
Łańcuchy znaków....318
Nieznane wartości....319
Stosowanie złożonych kryteriów do pobierania wierszy....320
Eliminacja nadmiarowych informacji....322
Sortowanie danych za pomocą klauzuli ORDER BY....323
Przegląd zagadnień....324
Funkcje agregujące....325
GROUP BY i HAVING....327
COMPUTE i COMPUTE BY....328
Super aggregates (ROLLUP i CUBE)....330
Korelacja danych....333
Implementacja złączeń....333
Złączenia wewnętrzne (inner joins)....334
Złączenia naturalne (natural joins)....334
Złączenia skośne lub nieograniczone (cross lub unrestricted joins)....335
Złączenia zewnętrzne (outer join)....337
Złączenia własne (self join)....338
Podzapytania....339
Stosowanie podzapytań....339
Typy podzapytań....340
Podzapytania skorelowane....341
SELECT INTO....342
Operator UNION....344

Rozdział 11. Modyfikacja danych 347
Rejestracja transakcji 347
Wpisywanie danych 348
Polecenie INSERT VALUES 350
Wartości domyślne 352
Wpisywanie danych za pomocą polecenia SELECT 352
Korzystanie z procedur składowych do wpisywania danych 354
Usuwanie danych 355
Usuwanie danych za pomocą tabeli przeglądowej (lookup) 356
Używanie TRUNCATE TABLE 357
Aktualizacja danych 358
Polecenie UPDATE 359
Wykonywanie polecenia UPDATE z wykorzystaniem tabeli przeglądowej 360

Rozdział 12. Rozszerzenia języka Transact-SQL 363
Pliki wsadowe (batches) 363
Skrypty 364
Transakcje 365
Typy transakcji 366
Transakcje rozproszone 375
Elementy języka kontroli przepływu 377
Blok BEGIN...END 377
Polecenie PRINT 378
Blok IF...ELSE 378
Wyrażenia CASE 380
Polecenie WHILE 384
Blokowanie 385
Typy blokad 386
Kontrola blokowania 387
Obserwacja blokad 388

Rozdział 13. Indeksowanie w celu poprawy wydajności 395
Dlaczego używać indeksów? 395
Struktury indeksu 396
Opcje indeksu 398
Indeksy SQL Servera 401
Opcje fillfactor i pad_index 403
Opcja DROP_EXISTING 405
Opcja STATISTICS_NORECOMPUTE 406
Opcja SORT_IN_TEMPDB 406
Opcja IGNORE_DUP_KEY 406
Korzystanie z indeksu 407
Kiedy indeks będzie używany? 407
Jak sprawdzić, czy indeks jest używany? 408
Omijanie działań optymalizatora 413
Pielęgnacja indeksu 415
Statystyki 415
Polecenie DBCC UPDATEUSAGE 418
Polecenie DBCC SHOWCONTIG 418
Polecenie DBCC DBREINDEX 420
Polecenie DBCC INDEXDEFRAG 421
Widoki indeksowane 422
Zarządzanie indeksami za pomocą SQL Server Enterprise Managera 424

Wybór indeksu 427
Wybór indeksu zgrupowanego 428
Wybór indeksu niezgrupowanego....429
SQL Server Index Tunning Wizard 429
Indeksowanie pełnotekstowe....430

Rozdział 14. Zapewnienie integralności danych 435
Jak zapewnić integralność danych?....435
Typy integralności....436
Integralność dziedziny 436
Integralność referencyjna....436
Integralność encji 437
Tradycyjne metody integralności....437
Typy danych zdefiniowane przez użytkownika....437
Wartości domyślne....440
Reguły 445
Zapewnienie poprawności danych za pomocą więzów ANSI i mechanizmów integralności deklaratywnej 448
Właściwość IDENTITY....448
Więzy ANSI....453
Usuwanie więzów 462
Zapewnienie poprawności danych za pomocą SQL Server Enterprise Managera....463
Opóźnianie stosowania więzów 468
Wyłączanie więzów 469

Rozdział 15. Tworzenie widoków, procedur wyzwalanych, procedur składowych i funkcji 471
Tworzenie widoków i operacje na nich....472
Tworzenie widoków....473
Pobieranie informacji o widokach 476
Typy widoków 477
Modyfikacja danych w widokach 479
Zmienianie widoków 479
Usuwanie widoków....480
Procedury składowe 480
Tworzenie procedur składowych 482
Pobieranie informacji o procedurach składowych....483
Tworzenie grup procedur....484
Używanie parametrów w procedurach składowych 485
Opcja WITH RECOMPILE....487
Automatyczne uruchamianie procedur składowych podczas uruchomienia SQL Servera....488
Opcja WITH ENCRYPTION 488
Zdalne procedury składowe 488
Procedury wyzwalane (triggers)....489
Tabele inserted i deleted 490
Tworzenie procedur wyzwalanych za pomocą polecenia CREATE TRIGGER....490
Wymuszanie integralności danych za pomocą procedur wyzwalanych 493
Procedury wyzwalane typu INSTEAD OF....497
Pobieranie informacji o procedurach wyzwalanych....497
Funkcje zdefiniowane przez użytkownika 498
Funkcje skalarne 499
Funkcje tabelowe o wartości inline....500
Funkcje tabelowe o wartościach multistatement 501

Rozdział 16. Metodologie projektowania replikacji....503
Czym jest replikacja?....503
Współczynniki rozproszonych danych....504
Metodologie dystrybucji....505
Replikacja łączona....505
Replikacja migawkowa (snapshot)....507
Replikacja transakcyjna....508
Subskrypcje możliwe do uaktualniania....509
Replikacja migawkowa z uaktualnianiem subskrybentów....511
Replikacja transakcyjna z uaktualnianiem subskrybentów....512
Transakcje rozproszone....512
Metafora wydawca-subskrybent....513
Artykuły....513
Publikacje....515
Jak działa replikacja?....517
Jak działa replikacja łączona?....518
Wewnętrzne procesy replikacji migawkowej....520
Wewnętrzne procesy replikacji transakcyjnej....521
Wewnętrzne procesy uaktualniania kolejkowanego....522
Scenariusze replikacji....523
Publikator centralny....523
Publikator centralny ze zdalnym dystrybutorem....524
Republikator....524
Centralny subskrybent....525
Wiele publikatorów i subskrybentów....526
Transformacja publikowanych danych....527
Publikowanie baz danych SQL Servera w Internecie....527
Publikowanie poprzez VPN....528
Publikowanie przez Microsoft Proxy Server....528
Publikowanie przez File Transfer Protocol (FTP)....528
Zalecane działania....529
Replikacja w środowisku heterogenicznym....529
Publikowanie danych do baz danych innych niż SQL Server 2000....530
Publikowanie danych innych SZBD na SQL Server 2000....530

Rozdział 17. Implementacja metod replikacji....533
Kwestie zabezpieczeń....534
Konfiguracja serwerów....535
Instalacja replikacji....536
Instalacja serwera dystrybucji....536
Tworzenie publikacji....543
Tworzenie subskrypcji....548
Synchronizacja ręczna....553
Weryfikacja synchronizacji początkowej....554
Testowanie replikacji....554
Rozważania dotyczące replikacji....556
Publikowanie....556
Subskrybent....557
Wydajność....557
Zarządzanie replikacją....558
Używanie monitora replikacji....558
Zarządzenie serwerem dystrybucji....561
Skrypty replikacji....564
Monitor wydajności i replikacja....567

Używanie procedur składowych ... 568
Rozważania dotyczące administracji ... 569
Rozwiązywanie problemów ... 570
Odtwarzanie replikacji ... 571

Rozdział 18. Planowanie zadań przy pomocy SQL Server Agenta ... 573
Dlaczego automatyzować SQL Server? ... 573
Które narzędzie wybrać do planowania? ... 574
Usługa harmonogramu Windows 2000 ... 574
Usługa SQL Server Agent ... 575
Integracja z SQLMail ... 582
Konfiguracja profilu Microsoft Outlook Messaging ... 582
Konfiguracja SQL Mail dla SQL Servera ... 587
Konfiguracja SQL Mail do używania z SQL Server Agentem ... 588
Wysyłanie poczty elektronicznej z SQL Servera ... 588
Planowanie zadań ... 590
Tworzenie nowego zadania ... 590
Konfiguracja kroków zadania ... 591
Planowanie zadania ... 593
Powiadamianie o zadaniach ... 596
Ustawianie opcji zadań systemowych ... 596
Zadania Multiservera ... 597
Ustawianie komunikatów i operatorów ... 598
Konfiguracja operatorów ... 598
Tworzenie komunikatów ... 599
Ustawianie opcji komunikatów systemowych ... 602
Zalety integracji Monitora wydajności ... 603
Korzystanie z kreatora Database Maintenance Plan ... 604

Rozdział 19. Migracja danych pomiędzy serwerami baz danych ... 611
Przesuwanie danych przy pomocy metody backup-and-restore ... 612
Używanie BCP do ładowania danych ... 613
Uprawnienia BCP ... 614
Używanie programu BCP ... 614
Tryb macierzysty (native) a tryb znakowy ... 618
Pliki formatu ... 619
Rozmiary zadań wsadowych ... 621
Szybki BCP ... 621
Polecenie BULK INSERT ... 623
Data Transformation Services ... 625
Pakiety transformacji danych (DTP) ... 626
Eksport danych za pomocą DTS Import and Export Wizards ... 630

Rozdział 20. Podstawy i narzędzia monitorowania wydajności ... 637
Dobór i dostrajanie sprzętu komputerowego ... 638
Pamięć operacyjna ... 638
Procesor ... 638
Sieć komputerowa ... 639
Dyski ... 639
Parametry konfiguracji systemu Windows 2000 ... 641
Liczniki monitora wydajności systemu Windows 2000 ... 642
Pamięć ... 643
Procesor ... 643
Sieć ... 643
Dysk ... 643

Konfigurowanie systemu SQL Server....644
Opcje konfiguracyjne procesora....646
Opcje konfiguracyjne pamięci....648
Opcje wejścia-wyjścia....650
Opcje zapytań i indeksów....650
Monitorowanie wydajności systemu SQL Server....651
Liczniki wydajności systemu SQL Server....651
Monitorowanie wydajności za pomocą systemowych procedur składowanych....653
Okno bieżącej aktywności programu SQL Server Enterprise Manager....654
Korzystanie z programu SQL Server Profiler....655
Kreator dostrajania indeksów — Index Tuning Wizard....663

Rozdział 21. XML i SQL Server....667
Czym jest XML?....667
Dostęp do SQL Servera za pośrednictwem serwera internetowych usług informacyjnych....669
Katalogi wirtualne i SQL Server....670
Korzystanie z zasobów systemu SQL Server za pośrednictwem adresów URL....676
Odczyt i zapis XML w systemie SQL Server....683
Prawidłowe instrukcje SELECT....684
Stosowanie klauzuli FOR XML....684

Skorowidz....689

Rozdział 4.

Tworzenie baz danych i plików danych oraz administracja nimi

W rozdziale 3. „Narzędzia SQL Servera 2000 i jego usługi" zostały omówione różnorodne narzędzia, dostarczane wraz z SQL Serverem 2000, np. SQL Server Enterprise Manager, narzędzie z Microsoft Management Console (MMC). W tym rozdziale zostaną wykorzystane narzędzia SQL Server Enterprise Manager i SQL Server Query Analyzer.

Rozdział ten pokazuje, jak tworzyć bazę danych, zmieniać ją i usuwać. Po utworzeniu baza danych jest przechowywana w co najmniej dwóch osobnych plikach. Jeden z nich zawiera dane, tabele systemowe i inne obiekty bazy danych, drugi — dziennik transakcji. W SQL Serverze 2000 można określić przyrost bazy danych przez ustalenie parametrów przyrostu pliku bazy danych lub dziennika transakcji.



Również w tym rozdziale zostaną omówione podstawy używania grup plików. Ogólnie *grupa plików* pozwala na jednoznaczne wskazanie, w którym pliku bazy danych umieszczone zostaną obiekty takie, jak tabele i indeksy. Grupy plików mogą ułatwić pielęgnację bazy danych, np. przez zezwolenie na tworzenie kopii bezpieczeństwa jedynie grupy plików, a nie całej bazy danych. Grupy plików mają zalety zarówno jeśli chodzi o administrację, jak i o pielęgnację bazy danych, a także potencjalnie poprawiają wydajność większych kopii baz SQL Servera.

W tym rozdziale zostaną również omówione różne opcje konfiguracji bazy danych oraz ich wpływ na bazę.

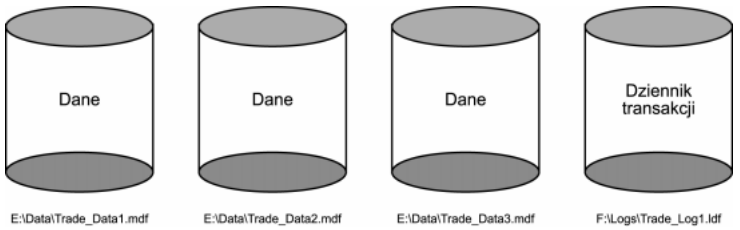
Tworzenie bazy danych

Aby utworzyć bazę danych w SQL Serverze, można skorzystać z:

- Database Creation Wizard,
- SQL Server Enterprise Managera,
- wyrażenia `CREATE DATABASE`.

Kiedy powstaje nowa baza danych, naprawdę tworzona jest kopia bazy danych `model`. Należy pamiętać, że wszystkie parametry bazy danych `model`, włączając w to opcje, które zostały ustawione przez użytkownika, pojawią się w każdej nowo utworzonej bazie danych. Po utworzeniu bazy danych przez przekopiowanie bazy `model` ta nowa wypełnia dodatkową przestrzeń pustym miejscem („rozszerza się" do żądanego rozmiaru).

W bazie danych muszą być pliki do fizycznego przechowywania danych na dysku. Po utworzeniu bazy danych powinno się określić co najmniej jeden plik do przechowywania danych i tabel systemowych oraz osobny plik do przechowywania dziennika transakcji. Baza danych i dziennik transakcji mogą obejmować wiele plików, jak pokazano na rysunku 4.1. Baza danych `Trade` w tym przykładzie ma trzy pliki danych i jeden plik na dziennik transakcji.



Rysunek 4.1. *Baza danych i dziennik transakcji mogą zajmować kilka plików*



Utworzone pliki bazy danych nie mogą być współdzielone z inną bazą danych lub dziennikiem transakcji.

W tej części zostanie omówione polecenie `CREATE DATABASE` i znaczenie każdego z jego parametrów. Po zrozumieniu działania tego polecenia będziesz mógł zobaczyć, jak utworzyć bazę danych za pomocą SQL Server Enterprise Managera. Składnia polecenia `CREATE DATABASE` wygląda następująco:

składnia

```
CREATE DATABASE database_name
[ON {[PRIMARY]
(NAME = logical_name,
FILENAME = 'phisical_name'
[,SIZE = size]
[,MAXSIZE = max_size |UNLIMITED]
[,FILEGROWTH = growth_increment])}
[,...n]]
[LOG ON
{(NAME = logical_name,
FILENAME = 'physical_name'\
```

```
[,SIZE = size] |UNLIMITED]
[,MAXSIZE = max_size |UNLIMITED]
[,FILEGROWTH = growth_increment])}
[,...n]]
[,COLLATE collation_name]
[FOR LOAD | FOR ATTACH]
```

W przypadku SQL Servera 2000 jedynym parametrem, jaki należy określić, jest logiczna nazwa bazy danych (NAME). Chociaż tworzenie bazy danych tym sposobem w SQL Serverze 2000 jest możliwe, nie jest jednak zalecane. Jeśli mimo to użytkownik skłania się ku wypróbowaniu tej metody, powinien określić koniecznie następujące parametry: logiczną nazwę bazy danych, nazwę pliku danych i jego rozmiar oraz nazwę pliku dziennika transakcji i jego rozmiar. Następująca lista opisuje dostępne parametry polecenia CREATE DATABASE:

- *database_name* odnosi się do bazy danych jako całości.
- ON PRIMARY określa, do których grup plików przynależy baza danych. Domyślną grupą plików jest Primary. Grupy plików zostaną omówione później.
- NAME określa nazwę logiczną, która będzie używana w SQL Serverze do odnoszenia się do fizycznego pliku bazy danych na dysku.
- FILENAME jest to ścieżka i nazwa pliku, określająca lokalizację danych na dysku twardym. Wymagany jest lokalny dysk twardy.
- SIZE określa, jak duży powinien być plik bazy danych. Wartość ta może być wyrażona w megabajtach lub kilobajtach. Domyślnym rozmiarem jest wielkość pliku Model. Aby określić jednostkę (megabajt lub kilobajt), należy dołączyć sufiks MB lub KB do parametru rozmiaru. Przykładowo, 10 MB utworzy plik o wielkości 10 megabajtów.



Megabajty mogą być używane tylko do określenia pełnych liczb. Aby utworzyć 2,5 megabajtową bazę danych, należy użyć kilobajtów, czyli 2 560 KB.

- MAXSIZE określa maksymalny rozmiar, do którego baza danych może się rozrastać. Jeżeli nie zostanie on określony i włączony jest automatyczny wzrost, baza danych może rozrastać się, aż zajmie cały dysk twardy. Parametr ten jest również wyrażony w megabajtach lub kilobajtach.
- FILEGROWTH określa wartość przyrostu, używanego do automatycznego powiększania pliku bazy danych. Parametr ten może być wyrażony w megabajtach, kilobajtach lub jako procent rozmiaru pliku w czasie powiększania go. Jeśli nie jest on określony, domyślnie przyjmuje się 1 MB. Parametr FILEGROWTH nie może przekroczyć wartości parametru MAXSIZE.
- LOG ON opisuje, gdzie położone są pliki dziennika transakcji i jaki jest ich rozmiar.
- COLLATE, nowa opcja w SQL Serverze 2000, określa językową stronę kodową w pojedynczej bazie danych. Może to być strona kodowa dostępna w SQL Serverze lub w systemie Windows. Jeżeli ten parametr nie zostanie określony, domyślnie przyjmowana jest wartość z kopii SQL Servera 2000. Strona kodowa może być również określona na poziomie tabeli i pojedynczej kolumny.

- `FOR LOAD` ustawia dla bazy danych własność DBO *Use Only*. Opcja ta jest wprowadzona jedynie dla zachowania zgodności wstecz z SQL Serverem 6.5 i nie powinna być używana w SQL Serverze 2000.
- `FOR ATTACH` dołącza ponownie zbiór plików, które tworzą bazę danych. Pliki bazy danych muszą być wcześniej utworzone i oddzielone od SQL Servera 2000. Opcja ta zostanie omówiona później.

Wydruk 4.1 pokazuje kod potrzebny do utworzenia bazy danych, która uruchamia się, rezerwując w sumie 25 MB — 20 MB dla obszaru danych i 5 MB dla dziennika transakcji. Pliki mogą rozrastać się do 115 MB — 100 MB dla danych i 15 MB dla dziennika transakcji. Baza ta korzysta również z kolejności sortowania, używanej przez SQL Server 2000.



W przypadku SQL Servera 2000 lepiej jest określić ilość przestrzeni, która jest potrzebna w danej chwili do przechowywania danych i dzienników, niż rezerwować całość przestrzeni dysku, jaka być może będzie potrzebna w przyszłości. Można skorzystać z parametrów `FILEGROWTH` i `MAXSIZE`, aby pozwolić w razie potrzeby na rozrastanie się bazy danych, a w danym momencie oszczędzić przestrzeń na dysku.



Wydruk 4.1. *Tworzenie bazy danych rezerwującej 25MB*

```
USE master
GO
CREATE DATABASE Frogger ON PRIMARY
( NAME = FroggerData,
 FILENAME =
    'D:\Program Files\Microsoft SQL Server\MSSQL\Data\FroggerData.mdf',
 SIZE = 20MB
 MAXSIZE = 100 MB
 FILEGROWTH = 10 MB )
LOG ON
(NAME = FroggerLog,
FILENAME =
    'D:\Program Files\Microsoft SQL Server\MSSQL\DATA\FroggerLog.ldf',
SIZE = 5MB,
MAXSIZE = 15MB,
FILEGROWTH = 1 MB)
GO
```



```
The CREATE DATABASE process is allocating 20.00MB on disk
'FroggerData'.
The CREATE DATABASE process is allocating 5.00MB on disk 'FroggerLog'.
```

Wydruk 4.2 pokazuje, jak utworzyć bazę danych, która zajmuje kilka plików dla danych i dziennika. Warto zauważyć, że pliki dzienników i danych używają rozszerzeń sugerowanych przez firmę Microsoft. Pierwszy plik danych powinien mieć rozszerzenie *.MDF*, a kolejne — *.NDF*. Pliki dziennika powinny używać rozszerzenia *.LDF*. Ponownie używane są domyślne ustawienia strony kodowej języka z SQL Servera 2000.

źródło **Wydruk 4.2.** *Tworzenie bazy danych rozmieszczonej w kilku plikach*

```
USE master
GO
CREATE DATABASE Leap ON PRIMARY
( NAME = LeapData1,
 FILENAME =
    'D:\Program Files\Microsoft SQL Server\MSSQL\Data\LeapData1.mdf',
 SIZE = 5,
 MAXSIZE = 20,
 FILEGROWTH = 1 ),
( NAME = LeapData2,
 FILENAME =
    'DL\Program Files\Microsoft SQL Server\MSSQL\Data\LeapData2.ndf',
 SIZE = 5,
 MAXSIZE = 20,
 FILEGROWTH = 5)
 LOG ON
(NAME = LeapLog1,
 FILENAME =
    'D:\Program Files\Microsoft SQL Server\MSSQL\Data\LeapLog1.ldf',
 SIZE = 2,
 MAXSIZE = 20,
 FILEGROWTH = 1 ),
(NAME = LeapLog2,
 FILENAME =
    'D:\Program Files\Microsoft SQL Server\MSSQL\Data\LeapLog2.ldf',
 SIZE = 2,
 MAXSIZE = 10,
 FILEGROWTH = 2 ),
GO
```

wynik

```
The CREATE DATABASE process is allocating 5.00 MB on disk 'LeapData1'.
The CREATE DATABASE process is allocating 5.00 MB on disk 'LeapData2'.
The CREATE DATABASE process is allocating 2.00 MB on disk 'LeapLog1'.
The CREATE DATABASE process is allocating 2.00 MB on disk 'LeapLog2'.
```

Przy określeniu wielu plików danych SQL Server automatycznie rozdziela informacje do wszystkich określonych plików. Podział danych pomiędzy kilka z nich może pomóc w redukcji niezgodności i błędów w bazie danych. Warto zauważyć, że SQL Server nigdy nie dzieli informacji pomiędzy pliki dziennika. Są one wypełniane sekwencyjnie i kiedy jeden z nich jest pełny, dane są kierowane do kolejnego pliku dziennika transakcji.

Wydruk 4.3 pokazuje, jak utworzyć bazę danych, która będzie sortować dane tak, jak określisz to poleceniem `COLLATE`. W tym przypadku SQL Server 2000 tworzy bazę danych, która korzysta ze strony kodowania Latin1 lub 1251, słownikowej kolejności sortowania (*General*) i jest „niewrażliwa" na wielkość liter (CI) oraz akcent (AI).



Jeżeli nie używasz macierzy RAID 5 (*redundant array of inexpensive disks*) lub w wyższej wersji, zaleca się umieszczenie dzienników transakcji na osobnych dyskach fizycznych. Rozmieszczenie ich w ten sposób zapewnia większe możliwości odzyskania danych w przypadku uszkodzenia dysku. Dodatkową zaletą tego rozwiązania jest to, że zapis do dziennika transakcji nie koliduje z zapisem do plików danych.

źródło **Wydruk 4.3.** *Tworzenie bazy danych SQL Servera 2000 używającej niestandardowej kolejności sortowania*

```
USE master
GO
CREATE DATABASE Swim ON PRIMARY
( NAME = SwimData,
 FILENAME =
    'D:\Program Files\Microsoft SQL Server\MSSQL\Data\SwimData.mdf',
 SIZE = 20MB,
 MAXSIZE = 100MB,
 FILEGROWTH = 10MB )
 LOG ON
(NAME = SwimLog,
 FILENAME =
    'D:\Program Files\Microsoft SQL Server\MSSQL\Data\SwimLog.ldf',
 SIZE = 5MB,
 MAXSIZE = 15MB,
 FILEGROWTH = 1MB )
COLLATE Latin1_General_CI_AI
GO
```

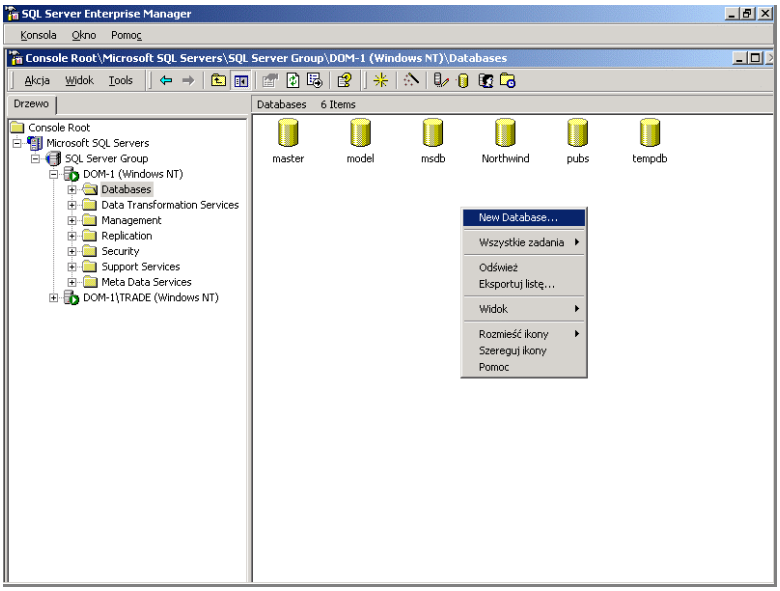
wynik

```
The CREATE DATABASE process is allocating 20.00 MB on disk 'SwimData'.
The CREATE DATABASE process is allocating 5.00 MB on disk 'SwimLog'.
```

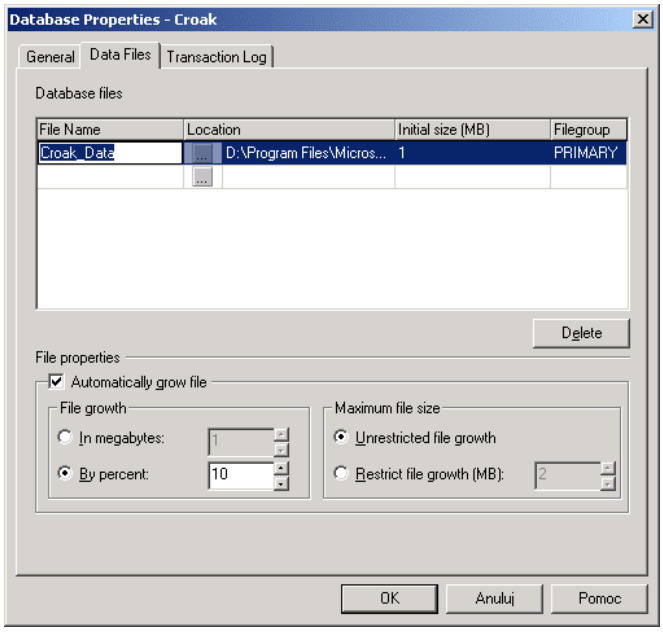
Można również używać SQL Server Enterprise Managera do tworzenia nowej bazy danych. Aby utworzyć bazę danych, postępuj według poniższych kroków:

1. Uruchom SQL Server Enterprise Managera — wybierz z menu *Start* opcję *Programy*, następnie *Microsoft SQL Server 2000* i *Enterprise Manager*.
2. Połącz się z kopią SQL Servera.
3. Rozwiń folder *Databases*, jak pokazano na rysunku 4.2.
4. Kliknij prawym klawiszem ikonę folderu *Databases* lub pustą przestrzeń w prawym panelu, następnie wybierz *New Database* z menu kontekstowego.
5. Pojawi się okno dialogowe *Database Properites* z aktywną zakładką *General*. Wpisz nazwę bazy danych (przykładowo `Croak`). Następnie możesz przejść do zakładki *Data Files* (zobacz rysunek 4.3), aby zobaczyć nowy plik bazy danych o nazwie `Croak_Data` o rozmiarze wstępnie ustawionym na 1 MB w domyślnym folderze *...\Data*. Warto zauważyć, że w bloku *File Properties* w dolnej części okna opcja *Automatically Grow File* jest włączona i ustawione są właściwości *File Growth*. Jest również ustawiony maksymalny rozmiar pliku jako *Unrestricted File Growth*.
6. Jeśli chcesz zmienić właściwości plików bazy danych, kliknij odpowiednie pole i wprowadź modyfikacje (w przykładzie pozostawiono wartości domyślne). Możesz dodać kolejne pliki bazy — przejdź do wolnego pola w kolumnie *File Name* i ustaw właściwości dodatkowego pliku.
7. Kliknij zakładkę *Transaction Log*. Warto zauważyć, że domyślną nazwą dziennika jest `Croak_Log`, a jego rozmiar to 1 MB.

Rysunek 4.2. *Prawy panel SQL Server Enterprise Managera pokazuje bazy danych i menu, pozwalające na rozmieszczanie ikon i wykonywanie zadań*

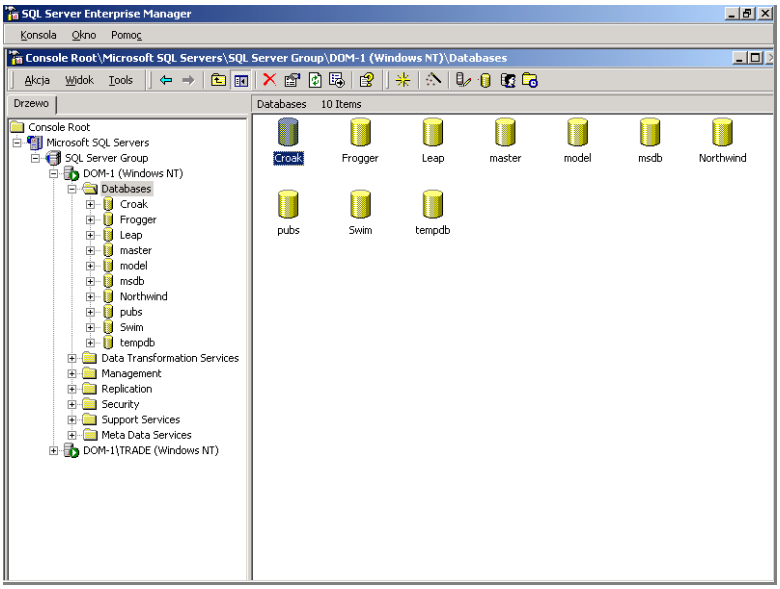


Rysunek 4.3. *Należy użyć tej zakładki, aby zobaczyć właściwości bazy danych, takie jak nazwa plików, ich rozmiar i grupa*



8. Kliknij *OK*. Okno powinno teraz wyglądać podobnie, jak pokazane na rysunku 4.4. Jeżeli bazy danych `Frogger`, `Leap` lub `Croak` nie są widoczne, kliknij prawym przyciskiem myszy folder *Databases* i wybierz z menu, które się pojawi, pozycję *Refresh*.

Rysunek 4.4.
Zostały dodane nowe bazy danych



Zbieranie informacji o bazie danych

Możesz zbierać informacje na temat bazy danych na kilka sposobów. Możesz używać SQL Server Enterprise Managera lub narzędzia SQL Server Query Analyzer albo też uruchomić systemowe procedury składowane.

Możesz skorzystać z procedury systemowej `sp_helpdb`, aby przetestować indywidualnie bazę danych lub zebrać podsumowujące informacje na temat wszystkich baz danych na SQL Serverze. Jeżeli procedura `sp_hepldb` zostanie uruchomiona bez określenia bazy danych, otrzymasz informacje na temat wszystkich baz danych konkretnego SQL Servera



Następujący przykład pokazuje część zwróconych danych. Oprócz przedstawionych tutaj otrzymuje się dodatkowo również informacje m.in. na temat możliwości odzyskiwania danych, ich uaktualniania i kolejności sortowania.



```
EXEC sp_helpdb
name        db_size     owner           dbid  created
Croak       2.00 MB     RHOME\rwaymi    10    Jul 29 2000
Frogger     25.00 MB    RHOME\rwaymi    9     Jul 29 2000
Leap        14.00 MB    RHOME\rwaymi    8     Jul 29 2000
master      13.88 MB    sa              1     Jul 27 2000
model       1.25 MB     sa              3     Jul 27 2000
msdb        14.00 MB    sa              4     Jul 27 2000
Northwind   4.25 MB     sa              6     Jul 27 2000
pubs        2.5 MB      sa              5     Jul 27 2000
Swim        25.00 MB    RHOME\rwaymi    7     Jul 29 2000
tempdb      8.75 MB     sa              2     Jul 29 2000
```

Warto zauważyć, że sp_helpdb daje podsumowujące informacje na temat baz danych SQL Servera. Przykładowo, baza danych Frogger ma 25 MB i jej właścicielem jest sa. Jeśli chcesz uzyskać więcej informacji na temat pojedynczej bazy danych, dodaj poniższy parametr w poleceniu sp_helpdb:



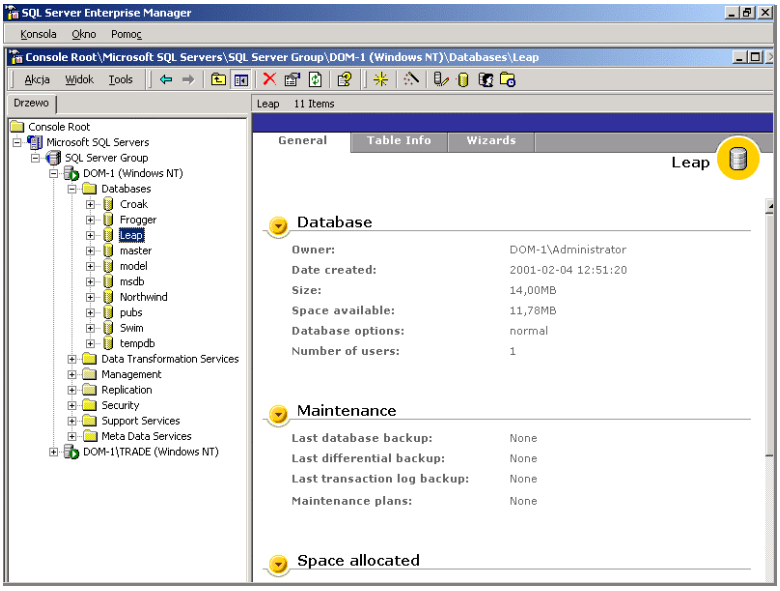
```
EXEC sp_helpdb Croak

name   Db_size  Owner  dbid  Created
Croak  2.00 MB  sa     9     Apr 14 1998

Name        Fileid  Filename
.....................................................................
Croak_Data  1       D:\PROGRAM FILES\MSSQL\data\Croak_Data.MDF
Croak_Log   2       D:\PROGRAM FILES\MSSQL\data\Croak_Log.LDF
```

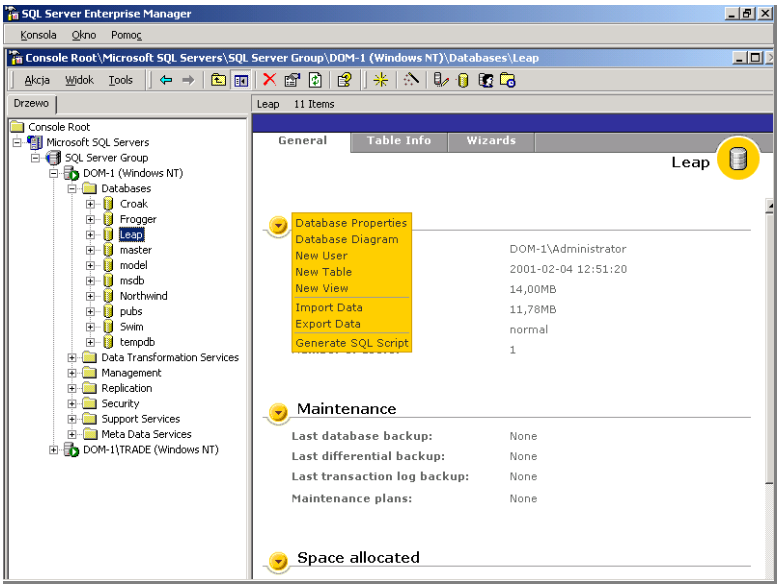
Tutaj możesz uzyskać nie tylko te dane, które zostały przedstawione w poprzednim przykładzie, ale także — dodatkowo — informacje o plikach bazy danych i ich lokalizacji. Przykładowy wydruk wyjściowy z wywołania powyższego polecenia pokazuje, że plik Croak_Data ma wielkość 2 MB i jest używany jedynie dla danych. Croak_Log ma 1 MB i jest używany jako dziennik transakcji.

Aby uzyskać informacje na temat baz danych, możesz skorzystać również z SQL Server Enterprise Managera. Otwórz Enterprise Managera i rozwiń folder *Databases*. Kliknij dwukrotnie bazę danych, aby wywołać okno właściwości bazy danych (pokazane wcześniej). Możesz także uzyskać informacje o bazie danych, klikając w lewym panelu, jak pokazano na rysunku 4.5 (wybierz *Taskpad* z menu *Widok*). Rysunek 4.5 pokazuje ogólne informacje na temat bazy danych Leap. Możesz uzyskać dodatkowe, bardziej szczegółowe informacje poprzez wybór niektórych opcji z ikony *Database*, *Maintenance* lub *Space Allocated*. Na rysunku 4.6 pokazano opcje ikony *Database*, dostępne w prawym panelu.



Rysunek 4.5. *Ogólne informacje na temat baz Leap*

Rysunek 4.6. *Dodatkowe opcje, dotyczące bazy danych*



Ustawianie opcji bazy danych

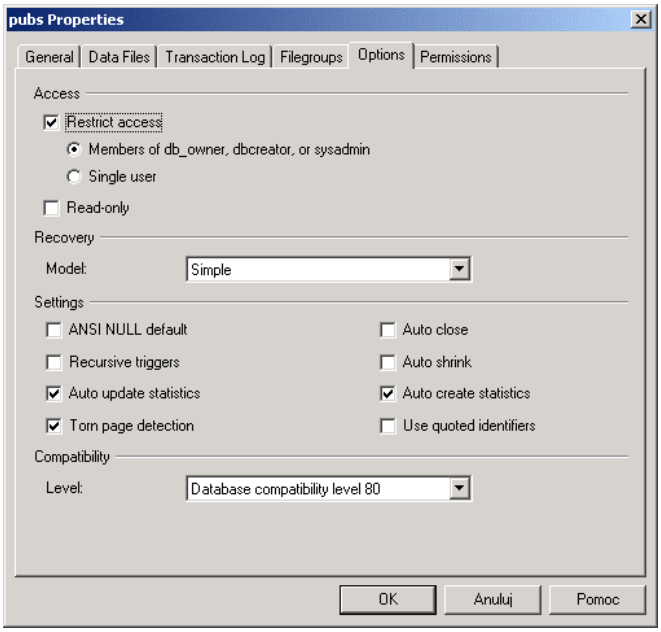
W części tej pojawi się więcej informacji na temat różnych opcji, jakie możesz stosować w bazie danych. Możesz modyfikować jej opcje za pomocą procedur składowanych w SQL Server Query Analyzerze lub poprzez SQL Server Enterprise Managera.

Jeśli chcesz przejrzeć i zmodyfikować opcje bazy danych za pomocą SQL Server Enterprise Managera, wybierz bazę danych, która będzie modyfikowana (w przykładzie baza `pubs`). Kliknij dwukrotnie wybraną bazę danych lub kliknij prawym klawiszem myszy i wybierz *Properties* z menu kontekstowego. Pojawi się arkusz *Properties* bazy `pubs`. Przejdź do zakładki *Options*, aby zapoznać się z różnymi opcjami, jak pokazano na rysunku 4.7.

Opcje są podzielone na trzy kategorie: *Access*, *Settings* i *Compatibility*. Poniżej przedstawiono znaczenie każdego z ustawień:

- *Members of db_owner, dbcreator or sysadmin* określa, że jedynie członkowie ustalonej roli bazy danych `db_owners` mogą używać tej bazy danych. Opcja ta jest często ustawiana, gdy administrator bazy danych przeprowadza jej konserwację i nie chce, aby zwyczajni użytkownicy korzystali z niej w tym czasie (ustawienie to zwane jest także DBO Use Only).
- *Single User* określa, że tylko jeden użytkownik w danym czasie może korzystać z bazy danych. Opcja ta de facto pozwala na połączenie z bazą tylko jednego użytkownika. Jest często używana podczas odtwarzania bazy danych.
- *Read-Only* oznacza, że w bazie danych nie można wprowadzać żadnych zmian (tryb tylko-do-odczytu).

Rysunek 4.7.
Zakładka Options w oknie właściwości bazy danych pubs



- *Recovery Model* określa, jak wiele transakcji zostanie zarejestrowanych i jakie rodzaje odtwarzania bazy danych są możliwe. Do wyboru są opcje: *Full*, *Simple* i *Bulk-Logged*. Wybierz *Full* — maksymalna możliwość odzyskiwania lub *Simple* — dla najłatwiejszej pielęgnacji.
- *ANSI NULL Default* — gdy opcja ta jest wybrana, powoduje definiowanie nowych kolumn tabeli jako domyślnie pustych (null) lub NOT NULL. Więcej na temat tabel i kolumn zostanie powiedziane w rozdziale 9. „Typy danych i tworzenie tabel".
- *Recursive Triggers* zezwala na używanie wyzwalaczy rekurencyjnych. *Wyzwalacz rekurencyjny* pojawia się, gdy modyfikacje danych w jednej tabeli (np.: w Tabeli A) wywołują wyzwalacz w innej tabeli (Tabela B). Następnie Tabela B uruchamia wyzwalacz w oryginalnej Tabeli A. SQL Server 2000 obsługuje rekurencję tylko do 32 poziomów zagnieżdżenia.
- *Auto Update Statistics* działa z *Auto Create Statistics*. Po pewnym czasie informacja w kolumnach zmieni się, jednak statystyka, dotycząca tych kolumn, nie zmieni się. Aby ominąć ten problem, należy co pewien czas uaktualniać statystyki. Opcja *Auto Update Statistics* wykonuje tę pracę automatycznie. Zaleca się włączenie tej opcji.
- *Torn Page Detection* wykrywa, kiedy nastąpił częściowy zapis strony danych na dysku (forma uszkodzenia danych). We wcześniejszych wersjach SQL Servera problem ten występował znacznie częściej, niż można się było tego spodziewać. Od wersji SQL Servera 7.0 uszkodzone strony przestały być kłopotliwe.

- *Auto Close* automatycznie zamyka bazę danych, kiedy ostatni użytkownik przestanie z niej korzystać.
- *Auto Shrink* automatycznie zmniejsza pliki danych i dziennika. Pliki dziennika kurczą się automatycznie po wykonaniu kopii zapasowej dziennika. Pliki bazy danych zmniejszają się, kiedy przy jej okresowym sprawdzaniu okaże się, że ma ona więcej niż 25% przydzielonej nieużywanej przestrzeni. Proces automatycznego zmniejszania bazy danych minimalizuje ją do rozmiaru, w którym baza ma 25% nieużywanej przestrzeni. Warto zauważyć, że proces automatycznego zmniejszania nie minimalizuje bazy danych do rozmiaru mniejszego niż jej oryginalny rozmiar.
- *Auto Create Statistics* tworzy automatycznie statystyki na podstawie dystrybucji wartości w kolumnie danych. Informacja ta jest używana przez SQL Server Query Optimizera do generowania planu zapytań, bazującego na kosztach używania różnych kolumn. Zaleca się włączenie tej opcji.
- *Use Quoted Identifiers* pozwala na używanie znaku podwójnego cudzysłowu jako części identyfikatora SQL Servera. *Identyfikator* jest nazwą obiektu; może to być zmienna, tabela lub jeszcze coś innego. Quoted Identifiers są przydatne, jeśli identyfikator (co się zdarza) jest zarezerwowanym słowem SQL lub zawiera spację — przykładowo, tabela `"Order Details"` lub tabela o nazwie `Table`.
- *Compatibility Level* umożliwia określenie, jaki poziom zgodności wstecz, z poprzednimi wersjami powinien być obsługiwany przez bazę danych. Poziom 80 jest wyznaczony dla SQL Servera 2000 (SQL Server w wersji 8.0), 70 dla SQL Servera 7.0 a 65 i 60 odpowiednio dla SQL Servera 6.5 i 6.0.

Możesz wykonać te same zadania za pomocą narzędzia SQL Server Query Analyzer i systemowej procedury składowanej `sp_dboption`. Przykładowo, aby ograniczyć dostęp do bazy danych `pubs` do *members of db_owner, dbcreator, or sysadmin*, uruchom kod, pokazany na rysunku 4.8. Jest to:

```
exec sp_dboption pubs, 'DBO Use Only', TRUE
```

Po uruchomieniu tego polecenia możesz sprawdzić, czy zostało faktycznie zastosowane poprzez uruchomienie systemowej procedury składowanej `sp_dboption` bazy `pubs` (zobacz rysunek 4.9). Jak widać z rysunku, w części status widnieje `dbo use only`. Opcja *trunc. log on chkpt.* (domyślna opcja systemowa po instalacji) i inne zostały również ustawione.

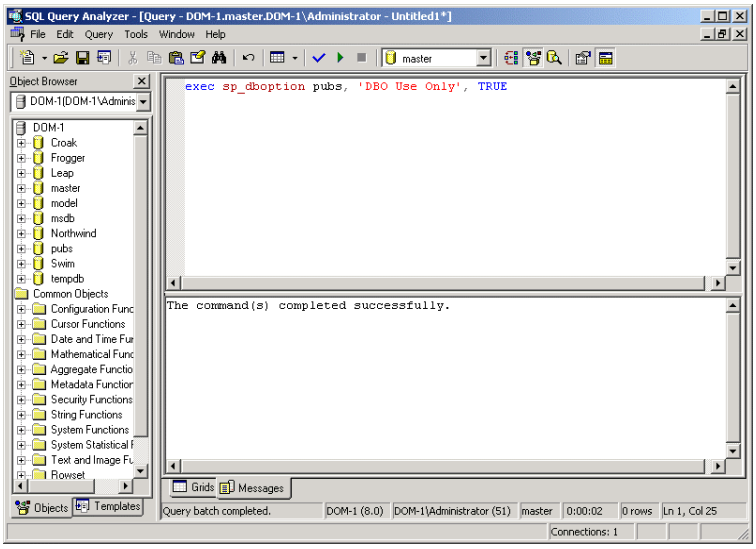
Aby wyłączyć status bazy danych DBO Use Only, uruchom następujący kod:

```
EXEC sp_dboption pubs, 'DBO Use Only', False
```

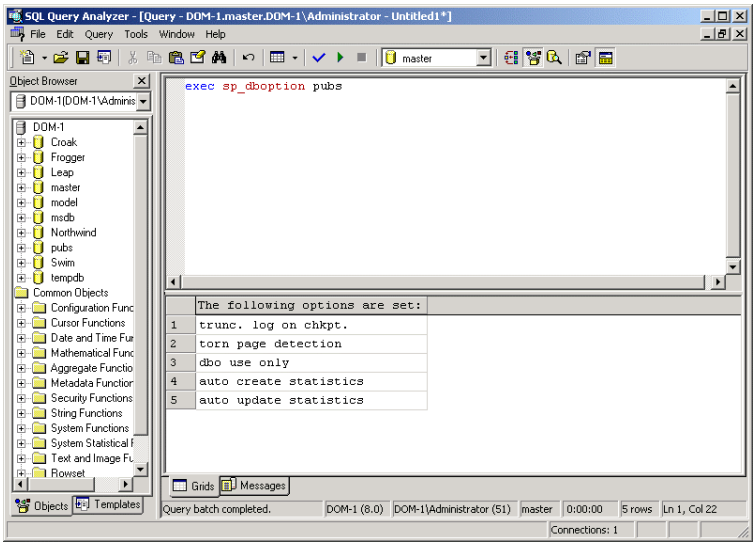
Jeżeli używana jest systemowa procedura składowana `sp_dboption`, możesz korzystać z kilku dodatkowych opcji, które nie są dostępne z Enterprise Managera:

- *Concat NULL Yields NULL* działa podobnie, jak mnożenie przez zero. Mnożenie przez zero zawsze zwraca zero. Concat NULL Yields NULL informuje, że wszystko, co zostanie dołączone do wartości NULL, da w wyniku NULL (zero). Przykładowo `'Hello world'+NULL = NULL`.

Rysunek 4.8.
Kod, który ustawia własność DBO Use Only dla bazy pubs



Rysunek 4.9.
Siatka wyników, pokazująca zmianę i status



- *Cursor Close on Commit* określa, że wszelkie otwarte kursory są zamykane, gdy transakcja jest zakończona. Kursor jest to zbiór wynikowy z zapytania.
- *Defult to Local Cursor* określa, że jeśli kursor jest utworzony bez słowa kluczowego GLOBAL, jest dostępny jedynie dla lokalnego pliku wsadowego, wyzwalacza, procedury składowanej i innych, które wygenerowały kursor.
- *Merge Publish* sprawia, że baza danych może być używana jako publikator w scenariuszu łączonej replikacji. Scenariusze replikacji zostały omówione w rozdziale 16. „Metodologie projektowania replikacji”.
- *Offline* zamyka bazę danych. Jeżeli jest ona nieaktywna (offline), może być przeniesiona na nośnik przenośny i dystrybuowana.

- *Published* pozwala bazie danych na publikację artykułów dla replikacji.
- *Subscribed* określa, że baza danych jest odbiorcą publikowanych danych.

Zmiana rozmiaru bazy danych

Aby zmienić definicje plików i ustawienia rozmiarów konkretnej bazy danych, skorzystaj z polecenia ALTER DATABASE lub z SQL Server Enterprise Managera. W celu zmniejszenia bazy danych użyj polecenia DBCC SHRINKDATABASE lub DBCC SHRINKFILE. Aby dodać grupy plików do bazy danych, użyj polecenia ALTER DATABASE. W tym podrozdziale najpierw zostanie omówione polecenie ALTER DATABASE oraz zostaną wykonane pewne modyfikacje baz danych, stworzonych w tym rozdziale. Następnie zostanie przedstawione zmniejszenie bazy danych za pomocą polecenia DBCC SHRINKDATABASE. Na końcu zostaną w skrócie opisane grupy plików.



Aby skorzystać z polecenia ALTER DATABASE, użytkownik musi mieć uprawnienia CREATE DATABASE.

Składnia polecenia ALTER DATABASE wygląda następująco:



```
ALTER DATABASE database
{
ADD FILE File_specification [,...n] [TO FILEGROUP filegroup_name]
| ADD LOG FILE File_specification [,...n]
| REMOVE FILE logical_name
| ADD FILEGROUP filegroup_name
| MODIFY FILE File_specification
| MODIFY NAME = new_dbname
| MODIFY FILEGROUP filegroup_name, filegroup_property
| SET optionspec, ...n WITH termination
| COLLATE collation_name
}
File_specification
(NAME = logical_name,
FILENAME = 'physical name'
[, SIZE = size]
[, MAXSIZE = max_size| UNLIMITED]
[, FILEGROWTH = growth_increment])
```

Omówienie składni:

- *database* jest nazwą zmienianej bazy danych.
- ADD FILE określa plik danych, który ma zostać dodany.
- Wszystkie opcje *File_specification* są identyczne, jak w przykładach polecenia CREATE DATABASE.
- TO FILEGROUP określa grupę plików, do której będzie dodany plik. Jeżeli grupa nie zostanie określona, plik jest dodawany do domyślnej grupy (PRIMARY).

- ADD LOGFILE dodaje nowy plik dziennika do bazy.
- REMOVE FILE usuwa plik z bazy danych. Plik musi być pusty przed usunięciem. Można użyć polecenia DBCC SHRINKFILE z opcją EMPTYFILE, aby wyczyścić plik.
- ADD FILEGROUP dodaje nową grupę plików. Należy podać również nową nazwę grupy plików.
- REMOVE FILEGROUP usuwa grupę plików, jak również usuwa wszystkie pliki, należące do grupy. Pliki w grupie muszą być puste. Można użyć polecenia DBCC SHRINKFILE z opcją EMPTYFILE, aby wyczyścić pliki.
- MODIFY FILE pozwala na modyfikację własności pliku, m.in. opcji *physical_name*, FILEGROWTH i MAXSIZE. Jeżeli parametr MAXSIZE jest modyfikowany, nowy rozmiar musi być większy niż bieżący. Parametr FILENAME można zmienić jedynie dla plików, które rezydują w tempdb; zmiana ta będzie widoczna po ponownym uruchomieniu SQL Servera.



Mimo to, że można określić rozmiar pliku mniejszy niż bieżący, nie można w ten sposób zmniejszyć pliku. Należy użyć polecenia DBCC SHRINKDATABASE lub DBCC SHRINKFILE.

- MODIFY NAME zmienia nazwę bazy danych.
- MODIFY FILEGROUP pozwala zmienić własności grupy plików, m.in. READONLY, READWRITE i DEFAULT.
- SET pozwala na określenie READONLY, READWRITE i DEFAULT. Parametr termination określa, kiedy pojawi się wycofanie danych (rollback) — ROLLBACK AFTER n SECONDS lub ROLLBACK IMMEDIATE.
- COLLATE określa kolejność sortowania, zastosowaną w bazie danych. Musi to być nazwa SQL Collation. Jeżeli kolejność sortowania nie jest określona, domyślnie jest to kolejność stosowana przez SQL Server 2000.

Rozszerzenie bazy danych

Możesz rozszerzać bazy danych poprzez dodawanie nowych plików. Możesz dodać pliki do części danych, jak również do części dziennika bazy danych. Dopóki nie zostanie wyłączona opcja automatycznego rozrastania się bazy danych, pliki danych będą rosły, aż zajmą całą dostępną przestrzeń na dysku twardym. Należy pamiętać, że dane przechowywane w wielu plikach w bazie danych są automatycznie rozkładane do wielu plików. Należy się zastanowić, czy potrzebnych jest kilka plików bazy danych, skoro zapewnia się automatyczny rozrost pliku w razie potrzeby. Baza z wieloma plikami może powodować, że jej pielęgnacja stanie się mniej efektywna. Jednak są również zalety tego rozwiązania:

- Można umieścić pliki na odrębnych dyskach fizycznych.
- Można poprawić wydajność, ponieważ odczyty bazy danych i zapisy w niej mogą być weryfikowane przez odrębnych kontrolerów dysków.

- Kopie bezpieczeństwa poszczególnych plików bazy danych mogą być tworzone niezależnie.
- W przypadku korzystania z grup plików określone części danych mogą być umieszczone w odpowiednich plikach. Przykładowo, tabela `lista płac` może być umieszczona w jej własnej grupie plików, w osobnym pliku.



Należy określić maksymalny rozmiar bazy danych i dziennika transakcji. Jeżeli nie zostanie on określony ani nie został ograniczony rozrost automatyczny, baza danych może zająć cały dysk twardy. Kiedy partycja dysku twardego z systemem Windows zapełni się, system ten generuje błędy i może nie działać, dopóki nie zwolni się miejsce na tej partycji.

Wydruk 4.4 pokazuje, jak dodać nowy plik danych do bazy danych `Croak` za pomocą narzędzia SQL Server Query Analyzer.

źródło **Wydruk 4.4.** *Dodawanie nowego pliku danych do bazy Croak*

```
ALTER DATABASE croak
ADD FILE
( NAME = CroakData2,
 FILENAME =
 'D:\Program Files\Microsoft SQL Server\MSSQL\Data\CroakData2.ndf'
SIZE = 2,
MAXSIZE = 10,
FILEGROWTH = 2)
```

wynik

```
Extending database by 2.00 MB on disk 'CroakData2'.
```

Po tej operacji możesz uruchomić systemową procedurę składowaną `sp_helpdb`, aby sprawdzić, czy baza danych została poprawnie rozszerzona:

```
sp_helpdb croak
```

Baza danych powinna mieć wielkość 4 MB w plikach *Croak_Data*, *Croak_Log* i *CroakData2*.

Następnie możesz rozszerzyć plik dziennika bazy danych:

źródło

```
ALTER DATABASE croak
ADD LOG FILE
( NAME = CroakLog2,
  FILENAME =
  'D:\Program Files\Microsoft SQL Server\MSSQL\Data\CroakLog2.ndf',
SIZE = 2,
MAXSIZE = 10,
FILEGROWTH = 2)
```

wynik

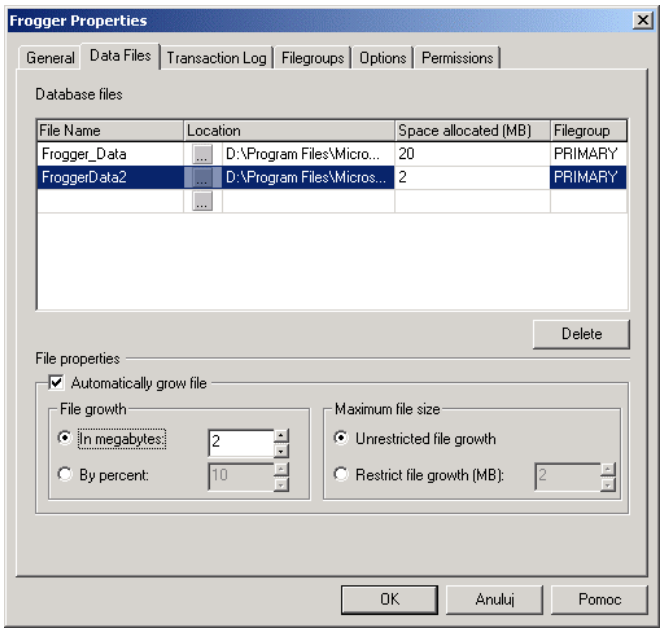
```
Extending database by 2.00 MB on disk 'CroakLog2'.
```

Wyniki możesz sprawdzić poprzez uruchomienie procedury `sp_helpdb Croak`. Baza danych powinna mieć wielkość 6 MB.

Podobne rezultaty powinno się osiągnąć, korzystając z SQL Server Enterprise Managera. Postępuj zgodnie z podanymi krokami, aby dokonać modyfikacji utworzonej wcześniej bazy danych `Frogger`:

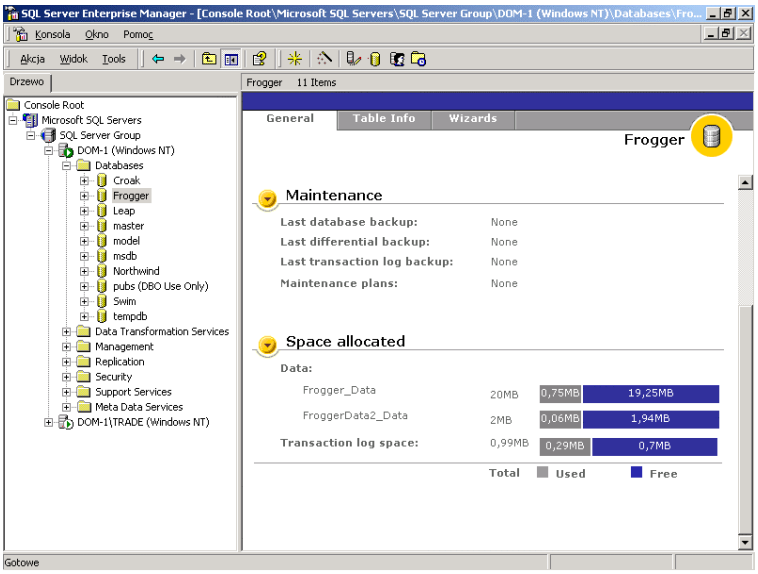
1. Uruchom SQL Server Enterprise Managera.
2. Rozwiń folder *Databases* i otwórz okno *Properties* bazy danych `Frogger` (możesz to zrobić, klikając prawym klawiszem bazę danych `Frogger` i wybierając z menu kontekstowego *Properties*).
3. Na zakładce *Data Files* tego okna kliknij puste pole w kolumnie *FroggerData* i dodaj plik *FroggerData2*, jak pokazano na rysunku 4.10.

Rysunek 4.10. *Dodawanie pliku FroggerData2 o rozmiarze 2*



4. W kolumnie *Location* określ nową nazwę pliku *D:\Program Files\Microsoft SQL Server\MSSQL\DATA\FroggerData2.NDF* lub możesz przyjąć domyślną wartość *FroggerData2_data.ndf* (napęd i ścieżka mogą być inne).
5. W kolumnie *Space Allocated* (MB) wpisz liczbę 2.
6. Pozostaw *Filegroup* jako `PRIMARY`.
7. Upewnij się, czy pole *Grow File* w części *File Properties* jest zaznaczone.
8. Ustaw opcję *File Growth* na *In Megabytes* i wpisz wartość 2.
9. W części *Maksimum File Size* ustaw opcję *Restrict File Growth* (MB) na wartość 4.

10. Po dodaniu pliku danych możesz również rozszerzyć plik transakcji. Zmień wielkość miejsca zaalokowanego z 5 MB na 10 MB; raczej nie dodawaj nowego pliku.
11. Kliknij *OK* po zakończeniu konfiguracji.
12. Aby sprawdzić, czy baza została zmodyfikowana, kliknij bazę `Frogger` w lewym panelu, w folderze *Databases* Enterprise Managera. W prawym panelu możesz obejrzeć część *Space Allocated*, jak pokazano na rysunku 4.11 (może być potrzebne wybranie opcji *Taskpad* z menu *Widok*).



Rysunek 4.11. *Sprawdzenie, czy baza danych ma przydzieloną przestrzeń*

Zmniejszanie bazy danych

W celu zmniejszenia całej bazy danych skorzystaj z polecenia `DBCC SHRINKDATABASE`. Jego składnia jest następująca:

składnia

```
DBCC SHRINKDATABASE {
(database_name
[, target_percent]
[,{NOTRUNCATE | TRUNCATEONLY}])
}
```

Znaczenie składni:

- *database_name* jest nazwą bazy danych, która będzie zmniejszana.
- *target_percent* jest procentowym określeniem wolnej przestrzeni w bazie danych po zmniejszeniu.
- NOTRUNCATE wymusza pozostawienie wolnej przestrzeni w plikach bazy danych. Parametr *target_percent* jest ignorowany. Domyślnie zwolniona przestrzeń jest z powrotem dostępna dla systemu operacyjnego.

- TRUNKCATEONLY wymusza, aby nieużywana przestrzeń w plikach danych była oddawana do systemu operacyjnego. Po wybraniu TRUNCATEONLY parametr *target_percent* jest ignorowany i cała wolna przestrzeń jest zwracana do systemu.

Jakie skutki mają te polecenia? Polecenie SHRINKDATABASE zmniejsza wszystkie pliki danych w bazie, pozostawiając wyznaczony procent wolnej przestrzeni. Używane na końcu strony plików danych są przesuwane poniżej ustalonego procentowego progu. Na przykład w bazie danych o wielkości 10 MB, przechowującej jedynie 5 MB danych, można ustawić parametr *target_percent* na 20. Oznacza to, że 20% bazy danych ma być pozostawione jako wolna przestrzeń. Wszystkie wiersze utworzone w bazie danych są przemieszczane, tym samym wszelkie wolne miejsce na początku bazy danych jest usuwane. Proces ten jest podobny do defragmentacji dysków. Jest tylko 5 MB danych i ma pozostać 20% wolnej przestrzeni; 20% z 5 MB to 1 MB, czyli rozmiar bazy danych zostanie zmieniony na 6 MB, a pozostałe 4 MB zostają oddane z powrotem do systemu operacyjnego.



Nie można zmniejszyć bazy danych bardziej niż do obszaru, który jest potrzebny do przechowywania danych, nie można także jej zminimalizować do rozmiaru mniejszego niż baza model. Nie można również zmniejszyć bazy danych do rozmiarów mniejszych niż pierwotne, które były określone przy jej tworzeniu (za pomocą polecenia CREATE DATABASE). Aby ją jednak zmniejszyć, należy użyć polecenia DBCC SHRINKFILE.

W przypadku wybrania opcji NOTRUNCATE dane są przenoszone, aby wyczyścić przestrzeń na końcu pliku, ale nowo powstałe wolne miejsce jest wykorzystywane przez bazę danych, a nie oddawane do systemu operacyjnego.

Opcja TRUNCATEONLY zwalnia całą niezaalokowaną przestrzeń bez przesunięcia danych. Wszystkie puste strony przy końcu pliku są zwalniane dla systemu operacyjnego.

Możesz zmniejszać pojedyncze pliki bazy danych za pomocą polecenia DBCC SHRINKFILE. DBCC SHRINKFILE jest używane raczej do modyfikacji pojedynczych plików niż do wszystkich plików w bazie danych (jak to jest w przypadku polecenia DBCC SHRINKDATABASE).

Składnia polecenia DBCC SHRINKFILE jest następująca:



```
DBCC SHRINKFILE {
{ file_name |file_id }
[, target_size ]
[, {EMPTYFILE | NOTRUNCATE |TRUNCATEONLY}])
}
```

Używając polecenia DBCC SHRINKFILE należy określić nazwę pliku lub identyfikator *file_id*. Identyfikator *file_id* można znaleźć, uruchamiając procedurę systemową sp_helpdb, co zostało omówione wcześniej.

Parametr *target_size* został użyty w podobny sposób, jak to miało miejsce wcześniej, przy poleceniach SHRINKDATABASE. Polecenia TRUNCATEONLY i NOTRUNCATE są również takie same, jak opisywane wcześniej. Parametr EMPTYFILE jest interesujący, ponieważ przesuwa wszelkie używane strony w bieżącym pliku bazy danych do innych plików w grupie. Po tym, jak wszystkie strony zostaną przesunięte, plik będzie oznaczony jako pusty i nie można w nim przechowywać danych. Możliwość ta może być przydatna w przypadku korzystania z parametru REMOVE FILE lub REMOVE FILEGROUP w poleceniu ALTER DATABASE, ponieważ parametry te wymagają, aby plik był pusty.

Wydruk 4.5 pokazuje skrypt, który usuwa dane z pliku *FroggerData2* i używa opcji REMOVE FILE do usunięcia pliku z bazy danych Frogger.

źródło **Wydruk 4.5.** *Czyszczenie pliku i używanie REMOVE FILE*

```
USE frogger
GO
DBC SHRINKFILE (FroggerData2, EMPTYFILE)
GO
ALTER DATABASE Frogger REMOVE FILE FroggerData2
```

wynik

```
DbId  FileId    CurrentSize  MinimumSize   UsedPages EstimatePages
..... ........ ............  ............  ......... ..............
9     3         256          256           0         0

(1 row(s)affected)
DBCC execution completed. If DBCC printed error messages,
contact your system administrator.
The file 'FroggerData2' has been removed.
```

Jak można zauważyć, plik *FroggerData2* został najpierw wyczyszczony za pomocą opcji EMPTY FILE. Następnie została uruchomiona druga część polecenia wsadowego, czego skutkami są zmiany w bazie danych i usunięcie pliku. Aby upewnić się, że plik został usunięty, możesz uruchomić polecenie EXEC sp_helpdb Frogger:

źródło

```
EXEC sp_helpdb Frogger
```

wynik

```
name     Db_size  Owner   dbid Created     status
....... ......... ....... .... ............ ..................
Frogger 30.00 MB   sa     7    Jan 14 1999 no options set


name         Fileid    Filename                                      Filegroup
............ ......... ............. ......................................
FroggerData  1         D:\Program Files\MSSQL\data\FroggerData.mdf  PRIMARY
FroggerLog   2         D:\Program Files\MSSQL\data\FroggerLog.ldf   NULL

/...size      maxsize     growth    Usage
/...          ..........  ........  ..........
/...20480 KB  102400 KB   10240 KB  Data only
/...10240 KB  15360 KB    1024 KB   Log only
```

Warto zauważyć, że plik *FroggerData2* nie jest już skojarzony z bazą danych Frogger. Możesz sprawdzić za pomocą Eksploratora w katalogu ...*DATA*, że plik *FroggerData2.ndf* został również usunięty z systemu.

Zmiana nazwy bazy danych

Czasem zachodzi potrzeba zmiany nazwy bazy danych. Może być to spowodowane zmianami organizacyjnymi, takimi jak łączenie działów firmy lub przy okazji przesunięcia bazy danych do środowiska produkcyjnego. Bez względu na powód nazwę bazy danych możesz zmienić za pomocą poniższego polecenia. Uruchom systemową procedurę składowaną `sp_renamedb`:

```
EXEC sp_renamedb 'stara_nazwa', 'nowa_nazwa'
```

Zmieniając nazwę bazy danych, powinieneś mieć na uwadze kilka zasad:

- Użytkownik, który zmienia nazwę bazy, musi należeć do stałej roli systemowej serwera *sysadmin*.
- Właściwe działanie niektórych skryptów SQL może zależeć od nazwy bazy danych. Należy sprawdzić, czy w bazie są takie skrypty.
- Baza danych musi być w trybie pojedynczego użytkownika (single-user).
- Nazwy plików bazy danych i grup plików pozostają bez zmian po modyfikacji nazwy bazy.
- Procedurę systemową `sp_renamedb` należy uruchamiać z bazy `master`.

W tym przykładzie zostanie zmieniona nazwa bazy danych `Frogger` (ang. *żaba*) na `TsingWa` (*Tsing Wa* oznacza po chińsku żabę).

źródło

```
USE master
GO
EXEC sp_dboption Frogger, 'Single User', True
EXEC sp_renamedb 'Frogger', 'TsingWa'
EXEC sp_dboption TsingWa, 'Single User', False
GO
```

wynik

```
sp_dboption command succeeded
The database name 'TsinWa' has been set.
sp_dboption command succeded
```

Aby sprawdzić, czy nazwa została zmieniona, uruchom procedurę `sp_helpdb` (w przykładzie podano wyniki skrócone w celu zachowania przejrzystości):

źródło/wynik

```
EXEC sp_helpdb

name        db_size     owner         dbid         created
........    ..........  ........      .....    ........
Croak       6.00 MB     RHOME\rwaymi  10       Jul 29 2000
Leap        14.00 MB    RHOME\rwaymi  8        Jul 29 2000
master      13.88 MB    sa            1        Jul 27 2000
model       1.25 MB     sa            3        Jul 27 2000
msdb        14.00 MB    sa            4        Jul 27 2000
Northwind   4.25 MB     sa            6        Jul 27 2000
pubs        2.50 MB     sa            5        Jul 27 2000
Swim        25.00 MB    RHOME\rwaymi  7        Jul 29 2000
tempdb      8.75 MB     sa            2        Jul 29 2000
TsinWa      30.00 MB    RHOME\rwaymi  9        Jul 29 2000
```

Grupy plików bazy danych

Grupy plików umożliwiają bezpośrednie umieszczanie obiektów takich jak tabele i indeksy w określonym zbiorze plików bazy danych. Może to być przydatne dla administratorów, jak również korzystne dla wydajności. Na przykład, używając technologii RAID 5 (redundant array of inexpensive disks), można umieszczać grupy plików na osobnych dyskach fizycznych, a następnie rozmieszczać określone tablice, indeksy i inne obiekty bazy danych w plikach, należących do danej grupy plików. Sposób ten poprawia wydajność, ponieważ odczyty w bazie danych i zapisy w niej mogą się odbywać na osobnych dyskach fizycznych w tym samym czasie. Zaletą administracji jest to, że można tworzyć kopie bezpieczeństwa i odtwarzać pojedyncze pliki z danej grupy lub też całą grupę. Przykładowo, posiadając bazę danych o wielkości 50 GB, rozmieszczoną w pięciu grupach plików (po 10 GB każda), można utworzyć każdego wieczoru kopię bezpieczeństwa innej grupy plików, a nie całej bazy danych. Stale jednak pozostanie możliwość odzyskania danych.

W SQL Serverze używa się dwóch typów grup plików: `PRIMARY` (domyślna) i zdefiniowana przez użytkownika (user-defined). Grupa plików `PRIMARY` musi zawierać podstawowy plik danych i wszystkie pliki, które nie są przypisane do innej grupy. Grupy plików, zdefiniowane przez użytkownika, tworzy się przy modyfikacji bazy danych i dodawaniu plików do określonych grup.

Przy korzystaniu z grup plików należy pamiętać o kilku zasadach:

- Plik nie może należeć do więcej niż jednej grupy.
- Podstawowy plik danych musi być przypisany do grupy `PRIMARY`.
- Można przypisać tabele, indeksy, tekst, ntext i dane graficzne do grupy plików.
- Wszystkie tabele systemowe muszą znajdować się w grupie `PRIMARY` w podstawowym pliku danych.
- Jeżeli grupa `PRIMARY` przepełni się, nowe strony danych nie będą automatycznie przenoszone do grup plików, zdefiniowanych przez użytkownika.



Wyrażenia ntext, nchar i nvarchar odnoszą się do danych typu Unicode. Używa on kilku bajtów (na ogół 2) do reprezentowania każdego z wyświetlanych znaków, podczas gdy ASCII (char, varchar i text) używa jedynie 1 bajtu do przechowywania pojedynczego znaku. Więcej na temat typów danych zostanie powiedziane w rozdziale 9.

Następujące przykłady pokazują utworzenie nowej grupy plików, a następnie dodanie do niej pliku danych. Potem baza danych zostanie ponownie zmieniona — zostaną zmodyfikowane właściwości nowo utworzonej grupy plików, a także będzie ona dołączona do domyślnej grupy plików. Domyślna grupa plików `PRIMARY` powstaje podczas tworzenia bazy danych. Tylko jedna grupa plików w bazie może być domyślna. Jeżeli nowa grupa plików zostanie zaznaczona jako domyślna, nowe pliki dodawane do bazy danych zostaną umieszczone w tej grupie, jeśli nie zostanie dopisane jawnie `PRIMARY`. Ostatni z przykładów pokazuje usuwanie grupy plików.

Aby utworzyć grupę plików, uruchom polecenie:



```
ALTER DATABASE Leap
ADD FILEGROUP LeapGroup1
GO

The command(s) completed successfully.
```

Aby sprawdzić, czy grupa plików jest częścią bazy `Leap`, uruchom procedurę:



```
use Leap
GO
EXEC sp_helpfilegroup

Groupname    Groupid    Filecount
...........  .......    ..............
PRIMARY      1          2
LeapGroup1   2          0
```

Następnie trzeba wprowadzić zmiany w bazie, dodając nowy plik do grupy plików:



```
ALTER DATABASE Leap
ADD FILE
( Name = LeapDataG1,
 FILENAME =
       'D:\PROGRAM FILES\Microsoft SQL Server\MSSQL\DATA\LeapDataG1.ndf',
SIZE = 2)
TO FILEGROUP LeapGroup1
GO

Extending database by 2.00 MB on disk 'LeapDataG1'.
```

Ponownie uruchom procedurę `sp_helpfilegroup`. Wyniki powinny być podobne do przedstawionych:



```
USE Leap
GO
EXEC sp_helpfilegroup

Groupname      Groupid    Filecount
.............  .........  ................
PRIMARY        1          2
LeapGroup1     2          1
```

Możesz dołączyć zdefiniowaną przez użytkownika grupę plików `LeapGroup1` do domyślnej grupy plików poprzez zmianę jej właściwości. Możesz również ustawić atrybut grupy plików na `READONLY` lub `READWRITE`. Więcej informacji na temat opcji `READONLY` lub `READWRITE` możesz znaleźć w podręcznikach SQL Server Books Online.

Aby zaznaczyć nową grupę plików jako grupę domyślną, uruchom polecenia:



```
USE Leap
GO
ALTER DATABASE Leap
MODIFY FILEGROUP LeapGroup1 DEFAULT

The filegroup property 'DEFAULT' has been set.
```

Aby przetestować nową grupę domyślną, dodaj nowy plik bez określania grupy plików:

źródło/wynik

```
USE pubs
GO
ALTER DATABASE Leap
ADD FILE
( NAME = LeapDataTest,
FILENAME =
    'D:\PROGRAM FILES\Microsoft SQL Server\MSSQL\DATA\LeapDataTest.ndf',
 SIZE = 2)
GO

Extending database by 2.00 MB on disk 'LeapDataTest'.
```

Aby sprawdzić, do której grupy należy konkretny plik, trzeba uruchomić procedurę `sp_helpfile`.

Aby usunąć grupę plików, trzeba po pierwsze ustawić grupę `PRIMARY` jako grupę domyślną. Następnie należy usunąć zawartość plików ze starej grupy. Po wyczyszczeniu zawartości plików możesz bezpiecznie usunąć pliki, a następnie grupę plików, jak w poniższym przykładzie:

źródło

```
USE Leap
GO

ALTER DATABASE Leap
MODIFY FILEGROUP [PRIMARY] DEFAULT
GO

ALTER DATABASE Leap
REMOVE FILE LeapDataG1
GO
ALTER DATABASE Leap
REMOVE FILE LeapDataTest
GO

ALTER DATABASE Leap
REMOVE FILEGROUP LeapGroup1
GO
```

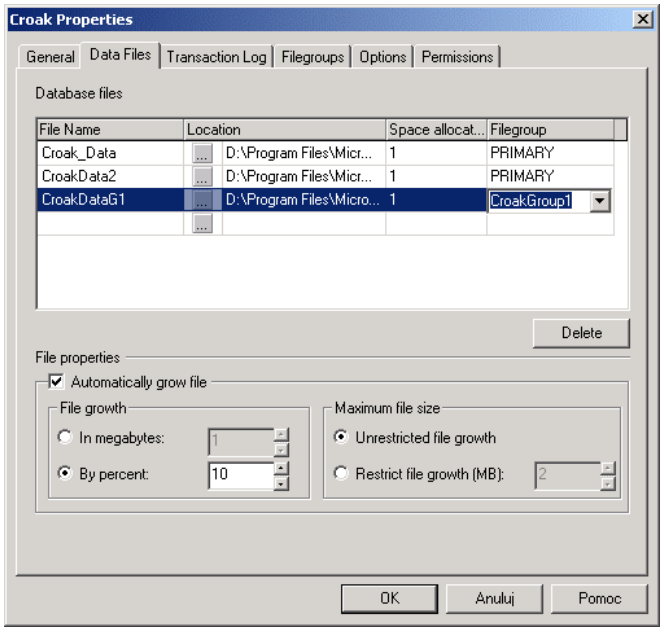
Powinny się ukazać komunikaty, informujące o usunięciu grupy plików `LeapGroup1` i plików z tej grupy. Możesz uruchomić ponownie procedurę `sp_helpfile`, aby sprawdzić, czy pliki i grupa zostały pomyślnie usunięte.

Możesz wykonać operację usunięcia grupy z poziomu SQL Server Enterprise Managera w podobny sposób, jak przy tworzeniu plików. Postępuj zgodnie z przedstawionymi krokami, aby utworzyć plik bazy danych i dodać go do nowej grupy plików:

1. Przejdź do bazy danych `Croak` i otwórz arkusz właściwości tej bazy na zakładce *Data Files*.
2. Aby utworzyć nową grupę i dodać do niej plik, kliknij puste pole poniżej `CroakData2` i dodaj `CroakDataG1`.

3. W lokalizacji podaj *D:\Program Files\Microsoft SQL Server\MSSQL\Data\CroakDataG1.ndf*.
4. W kolumnie *Space Allocated* (MB) wpisz 1.
5. W polu *Filegroup* dodaj `CroakGroup1`, jak pokazano na rysunku 4.12.

Rysunek 4.12. *Wybór grupy plików okna właściwości bazy danych*



6. Kliknij *OK* po zakończeniu.
7. Aby sprawdzić, otwórz ponownie okno właściwości dla bazy `Croak` i kliknij zakładkę *Filegroups*. Wśród pokazanych grup powinna się też znaleźć grupa `CroakGroup1` (z wyjątkiem grupy `PRIMARY`).

Część ta omawia pobieżnie grupy plików i ich tworzenie. Mogą być one bardzo złożone i nie są tematem tej książki. Więcej informacji na temat grup plików możesz znaleźć w podręcznikach SQL Server Books Online lub publikacji „Microsoft SQL Server 2000 Unleashed”, wydanej przez Sams Publishing.

Usuwanie bazy danych

Czasem konieczne staje się usunięcie bazy danych z systemu. Może ona na przykład nie być już dłużej potrzebna. W każdym przypadku usuwanie bazy danych z SQL Servera jest względnie proste. Można ją usunąć zarówno za pomocą SQL Server Enterprise Managera, jak też języka Transact-SQL.

Zanim usunie się bazę danych, należy uświadomić sobie kilka istotnych zależności:

- Usunięcie bazy danych powoduje wyeliminowanie informacji na temat tabel systemowych oraz usunięcie z systemu plików danych i dziennika.
- Usunięta baza danych może być przywrócona jedynie za pomocą odtwarzania z kopii bezpieczeństwa.
- Podczas usuwania z bazy nie może korzystać żaden użytkownik.
- Aby usunąć bazę danych, należy mieć uprawnienia roli `db_owner` (lub serwer sysadmin).
- Polecenie `DROP DATABASE` należy wydawać po zalogowaniu do bazy `master`.

Składnia polecenia `DROP DATABASE` wygląda następująco:

składnia

```
DROP DATABASE database_name, database_name2.....
```

Polecenie `DROP DATABASE` umożliwia usunięcie jednorazowo wielu baz danych. W tym przykładzie zostaną usunięte bazy `TsingWa` (uprzednio `Frogger`) i `Croak`. Należy uruchomić poniższy kod z SQL Server Query Analyzera:

źródło

```
USE MASTER
GO
DROP DATABASE TsingWa, Croak
GO
```

wynik

```
Deleting database file
'D:\Program Files\Microsoft SQL Server\MSSQL\data\Croak_Data.MDF'.

Deleting database file
'D:\Program Files\Microsoft SQL Server\MSSQL\data\Croak_Log.LDF'.

Deleting database file
'D:\Program Files\Microsoft SQL Server\MSSQL\data\CroakData2.ndf'.
Deleting database file
'D:\Program Files\Microsoft SQL Server\MSSQL\data\CroakLog2.ldf'.

Deleting database file
'D:\Program Files\Microsoft SQL Server\MSSQL\data\CroakDataG1.NDF'.
Deleting database file
'D:\Program Files\Microsoft SQL Server\MSSQL\data\froggerdata.mdf'.

Deleting database file
'D:\Program Files\Microsoft SQL Server\MSSQL\data\froggerlog.ldf'.
```

Jak możesz zauważyć, wszystkie pliki, które były związane z tymi dwiema bazami danych, zostały usunięte — to samo stało się z bazami danych (zostały usunięte z systemu). Aby to sprawdzić, możesz uruchomić `sp_helpdb`.

Możesz również użyć SQL Server Enterprise Managera do usunięcia bazy danych. Postępuj zgodnie z krokami:

1. Uruchom SQL Server Enterprise Managera i przejdź do folderu *Databases*.
2. Kliknij prawym klawiszem myszy bazę danych `Leap` i wybierz *Usuń* z menu kontekstowego.
3. W polu potwierdzenia *Delete Database* wybierz *Yes*.

To wszystko. Baza danych nie jest już częścią systemu. Pliki danych i dziennika zostały również usunięte.