Administrowanie systemami bazodanowymi

 1 Wprowadzenie do systemów relacyjnych baz danych

  1.1 Podstawowe pojęcia

 1.2 Architektura systemu zarządzania bazą danych

 1.3 Structured Query Language (SQL)

  1.4 Główne obiekty bazy danych

 1.5 Przykłady RDBMS

 2 Architektura systemu zarządzania bazą danych (SQL Server)

2.1 Instancje programu SQL Server

  2.2 Narzędzia SQL Server

  2.3 Podłączanie do serwera SQL

 3 Architektura bazy danych

  3.1 Pliki bazy danych

  3.1.1 Strony i zakresy

  3.1.2 Grupy plików

  3.1.3 Bazy danych systemu

 3.1.4 Bazy danych zdefiniowanych przez użytkownika

 4 Tworzenie obiektów bazy danych

  4.1 Tworzenie tabel

 4.2 Główne typy danych SQL Server

  4.3 Zmiany tabel

 4.4 Spadek tabel

 4.5 Liczba wyświetleń

 4.6 Indeksy

  4.6.1 Tworzenie indeksów

 4.7 Zmiana indeksów

 4.8 Praca z indeksami

 5 optymalizacja kwerendy

 5.1 Główne elementy planów realizacji programu SQL Server

 6 Przechowywane procedury, funkcje i wyzwalacze

  6.1 Oświadczenia warunkowe

  6.2 Pętle

  6.3 Kursory

  6.4 Procedury przechowywane

 6.5 Funkcje

  6.6 Triggers

 7 Zabezpieczenia i autoryzacja użytkowników w programie SQL Server

7.1 Uwierzytelnianie serwera SQL

 7.2 Autoryzacja SQL Server

 7.3 Łańcuchy własnościowe

 8 kopii zapasowych w programie SQL Server

 8.1 Pełna kopia zapasowa bazy danych

 8.2 Różnicowe kopie zapasowe

 8.3 Kopia zapasowa dzienników transakcji

 9 Przywracanie baz danych SQL Server

 9.1 Proces przywracania

 9.2 Przywracanie do punktu awarii

  9.3 Przywracanie do punktu w czasie

 9.4 Przywracanie do zaznaczonych transakcji

  9.5 Przywracanie bazy danych systemów

  9.6 Przypomnienie o odzyskiwaniu kopii zapasowych

  9.7 Procedura odzyskiwania próbek

 10 Wysoka dostępność

 10.1 Log Wysyłka

  10.2 Dublowanie bazy danych

  10.3 Grupy Dostępność AlwaysOn

  10.4 AlwaysOn Instancje klastra pracy awaryjnej

 11 Bibliografia

**1 Wprowadzenie do systemów relacyjnych baz danych**

**1.1 Podstawowe pojęcia**

Baza danych składa się z plików przechowywanych na dyskach. System zarządzania bazą danych (DBMS) obejmuje oprogramowanie odpowiedzialne za zarządzanie tymi plikami. Systemy zarządzania bazą danych muszą dotyczyć następujących kwestii:

• Skuteczny dostęp do danych.

• Transakcje (ACID).

• Integralność danych.

• Nadzór konkurencji.

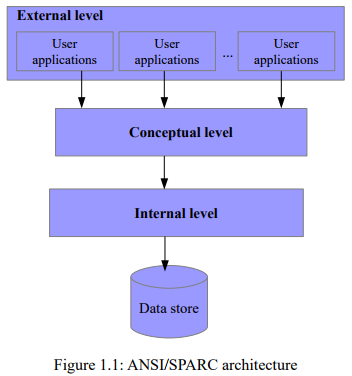
• spójność danych.

• Mechanizmy tworzenia kopii zapasowych i ich przywracania.

• Efektywne przetwarzanie zapytań i optymalizacja.

**1.2 Architektura systemu zarządzania bazą danych**

Najpopularniejszym systemem zarządzania bazą danych jest ANSI / SPARC [TK78, RG03]. Zapewnia trzy poziomy: zewnętrzne, koncepcyjne i wewnętrzne. Pierwszy poziom definiuje model danych dla użytkowników i aplikacji klienckich. Składa się z schematów użytkowników, uprawnień, tabel, widoków itp. Poziom pojęciowy stanowi model danych dla całego przedsiębiorstwa. Obejmuje katalog bazy danych i jego schemat. Poziom wewnętrzny zapewnia fizyczne przechowywanie danych.



***System zarządzania bazą danych*** obejmuje następujące składniki:

• *Query parser* - wykonuje syntakcyjną i semantyczną analizę zapytań użytkowników.

• *Query optimizer* (Optymalizator kwerend )- analizuje różne plany realizacji danego zapytania i wybiera plan o najmniejszym koszcie.

• *Query processor* (Procesor zapytania) - uruchamia odpowiedni plan zapytania i zwraca wynik.

• *Resource manager* (Menedżer zasobów) - odpowiada za zarządzanie rekordami w plikach.

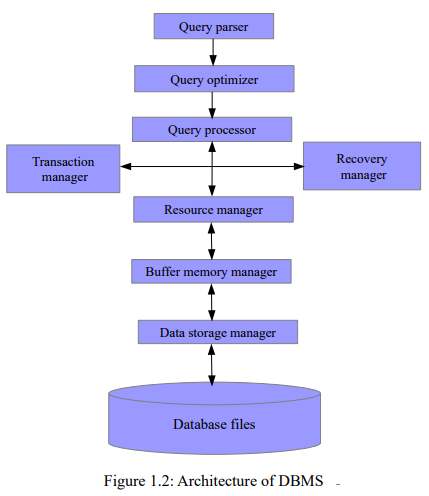
• *Buffer memory manager* (Menedżer pamięci buforowej) - przenosi strony danych z dysku na bufory pamięci.

*• Data storage manger* (Menedżer przechowujący dane) - umożliwia uzyskanie danych z dysku.

• *Transaction manager* - zapewnia atomowość, izolację, spójność i trwałość

transakcji.

• *Recovery manager* (Menedżer odzyskiwania) - jest odpowiedzialny za odzyskiwanie bazy danych w przypadku awarii.



**1.3 Structured Query Language (SQL)**

Każdy RDBMS definiuje język do manipulowania obiektami bazy danych i ich danymi. Nazywa się SQL (Structured Query Language). Ma kilka dialektów (np. T-SQL). SQL obejmuje następujące składniki:

• Language Definition Definition (DDL) - służy do tworzenia, modyfikowania i usuwania obiektów bazy danych (na przykład tabel, indeksów, widoków).

• Language Manipulation Language (DML) - służy do wybierania, wstawiania, aktualizowania i usuwania danych z tabel.

• Language Data Control Language (DCL) - służy do przyznawania i cofania uprawnień do bazy danych.

**Przykład 1.1:** Przykładowe zapytanie SQL.

Poniższe zapytanie zawiera informacje o imigrantach, których wiek jest większy niż 21, uzyskując informacje (imię, nazwisko i płeć).



**1.4 Główne obiekty bazy danych**

Najważniejsze obiekty bazy danych to:

• Tabela - jest to zestaw wierszy, w których każdy wiersz ma taką samą liczbę pól (kolumn).

• Ograniczenie - zawiera zasady egzekwowania integralności bazy danych.

• Widok - to jest wirtualny stół (zapytanie o nazwie).

• Indeks - jest strukturą pomocniczą (nadmiarową), która umożliwia przyspieszenie przetwarzania zapytań.

• Procedura - jest to zapisana kolekcja instrukcji SQL.

• Funkcja - jest to procedura przechowywana, która musi zwracać wartość.

• Trigger - jest to specjalna procedura przechowywana, która automatycznie uruchamia się, gdy wystąpi odpowiednie zdarzenie w bazie danych.

**1.5 Przykłady RDBMS**

Najbardziej znanymi systemami zarządzania relacyjnymi bazami danych są:

• Microsoft SQL Server (https://www.microsoft.com/sqlserver)

• Oracle Server (www.oracle.com/database)

• DB2 (www.ibm.com/db2)

• Sybase (www.sybase.com)

• MySQL (http://www.mysql.com/)

• PostgreSQL (<http://www.postgresql.org/>)

**2 Architektura systemu zarządzania bazą danych (SQL Server)**

SQL Server to bardzo potężny system zarządzania relacyjnymi bazami danych. Zapewnia między innymi następujące elementy:

• Mechanizmy wysokiej dostępności (AlwaysOn Availability Groups, AlwaysOn Failover Cluster Instances, Log Shipping, Database Mirroring).

• Zarządzanie transakcjami.

• Mechanizmy uwierzytelniania i autoryzacji.

• Mechanizmy tworzenia kopii zapasowych i przywracania.

Istnieje kilka wersji programu SQL Server [BOL12]:

• Datacenter (najpotężniejszy pod względem skalowalności i wirtualizacji).

• Enterprise (pozwala używać większej liczby procesorów i zużywa więcej pamięci niż standardowa).

• Standardowy.

• Developer (dedykowany programistom).

• Express (bezpłatna edycja).

**2.1 instancje SQL Server**

Administrator bazy danych może zainstalować więcej niż jedną kopię serwera SQL na tym samym serwerze. Każda kopia jest nazywana instancją SQL Server. Każda instancja składa się z zestawu konkretnych usług. Istnieją dwa typy instancji:

• instancja domyślna (zwana MSSQLSERVER),

• named instance (użytkownik musi podać nazwę podczas instalacji).

Najczęstszymi usługami SQL Server są [BOL12]:

• Usługa SQL Server (silnik bazy danych).

• Agent programu SQL Server - pozwala zautomatyzować niektóre zadania administracyjne (np. Tworzenie kopii zapasowych).

• Usługi integracyjne (ETL - Extract-Transform-Load).

• Usługi analizy.

• Usługi raportowania.

Uruchamianie nazwanej instancji:

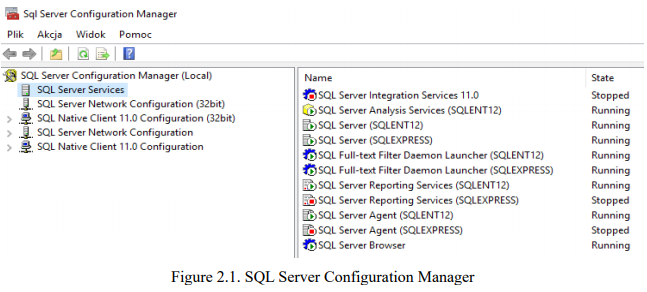


Uruchamianie domyślnego wystąpienia:



**2.2 Narzędzia SQL Server**

SQL Server Configuration Manager (SCCM) [BOL12] umożliwia konfigurowanie usług instancji i protokołów sieciowych.



Każda usługa ma konto usługi. SSCM pozwala zmodyfikować kont usługi. Obsługiwane są następujące konta usług:

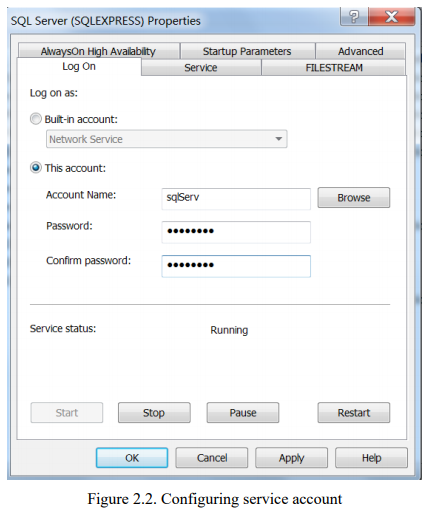
• Network Service (Usługa sieciowa).

• Local Service (Usługa lokalna).

• Local System (System lokalny).

• Local Windows account (Lokalne konto systemu Windows).

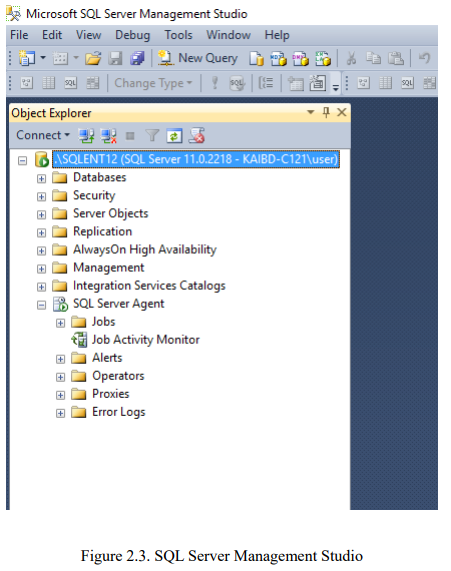
• Domain Windows Account (Konto systemu Windows w domenie).



**SQL Server Management Studio**

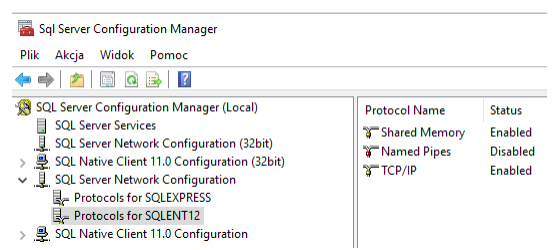
SQL Server Management Studio (SSMS) [BOL12] is an application which allows to

manage all SQL Server components.



**2.3 Podłączanie do serwera SQL Server**

SQL Server obsługuje następujące protokoły sieciowe:

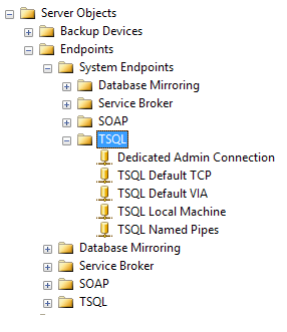


• Pamięć współużytkowana - dedykowana do połączeń lokalnych.

• TCP / IP - dotyczy połączeń użytkowników realizowanych za pośrednictwem sieci.

• Rurki nazwane - umożliwia dostęp do wystąpienia programu SQL Server.

Dla każdego protokołu sieciowego tworzone są specjalne obiekty o nazwie punktów końcowych. Punkty końcowe mają następujące nazwy [BOL12]:



• TSQL Local Machine (for the Shared Memory protocol),

• TSQL Named Pipes,

• TSQL Default TCP/IP,

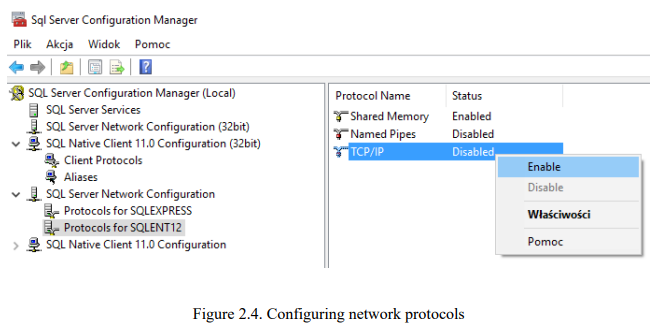
• TSQL Default VIA.

**Konfigurowanie protokołów sieciowych**

Aby uruchomić protokół TCP / IP dla instancji SQLENT12, konieczne są następujące kroki:

• Rozwiń *SQL Server Network Configuration* w lewym panelu.

• Kliknij *Protocols for SQLENT12*.

• Kliknij prawym przyciskiem na protokół TCP/IP i wybierz Enable.

**Łączenie się z bazą danych SQL Server z SSMS.**

1. Uruchom SSMS.

2. Podaj nazwę serwera (np. \ SQLENT12 - umożliwia nawiązanie połączenia z nazwą SQLENT12, która znajduje się na lokalnym komputerze).

3. Wybierz tryb uwierzytelniania.

4. Podaj nazwę użytkownika i hasło.



**3 Architektura bazy danych**

**3.1 Pliki bazy danych**

Każda baza danych SQL Server zawiera co najmniej dwa pliki:

• podstawowy plik bazy danych - przechowuje dane i ma rozszerzenie \* .mdf,

• plik dziennika transakcji - zapisuje wszystkie modyfikacje przeprowadzane przez transakcje bazy danych.

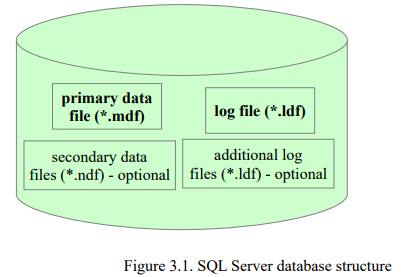
Baza danych może zawierać kilka plików danych. Dodatkowe pliki danych (wtórne pliki danych) posiadają rozszerzenie \* .ndf. Zastosowanie wielu plików danych pozwala na:

• zwiększenie wydajności,

• używać większej liczby rdzeni procesora,

• wykonywać mniejsze kopie zapasowe.

Na poniższej ilustracji przedstawiono strukturę bazy danych SQL Server:



**3.1.1 Strony i zakresy**

Strona [BOL12] jest podstawową jednostką składowania danych. Ma rozmiar 8kB w SQL Server. Wszystkie operacje odczytu i zapisu są preformowane na poziomie strony. Strona składa się z:

• nagłówek,

• obszar, w którym przechowywane są dane - ma rozmiar 8060 bajtów.

Część składa się z ośmiu stron. Rozszerzenia są wprowadzane w celu uproszczenia zarządzania przestrzenią.

**3.1.2 Grupy plików**

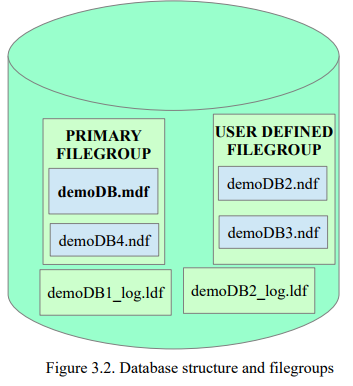
Grupy plików [BOL12] to kontenery do plików baz danych. Pozwalają sobie na następujące zadania:

• kontrolowanie, w jakich obiektach baz danych plików (np. Tabele, indeksy) są przechowywane,

• uprościć i przyspieszyć procedury tworzenia kopii zapasowych i ich przywracania.

Każda baza danych SQL Server ma tzw. Podstawową grupę plików. Podstawowym plikiem bazy danych jest członek tej grupy. Inne pliki danych mogą należeć do głównej grupy plików lub innych grup.

Na poniższej ilustracji przedstawiono strukturę bazy danych programu SQL Server zawierającą sześć plików: cztery pliki danych i dwa pliki dzienników.



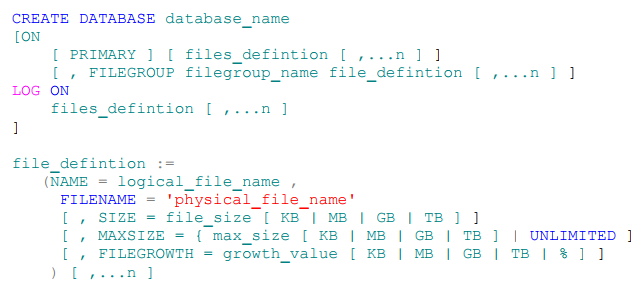
**3.1.3 Bazy danych systemu**

SQL Server udostępnia następujące bazy danych systemu [BOL12]:

|  |  |
| --- | --- |
| system | Przechowuje metadane instancji i informacje o konfiguracji. |
| model | Jest to szablon dla baz danych zdefiniowanych przez użytkownika. |
| msdb | Składa historię tworzenia kopii zapasowych i przywracania, obiekty programu SQL Server Agent. |
| resource | Jest to tylko baza danych do odczytu, która przechowuje wszystkie obiekty systemowe. |
| tempdb | Zapisuje tymczasowe obiekty i pośrednie wyniki wyszukiwania. |

**3.1.4 Bazy danych zdefiniowanych przez użytkownika**

SQL Server pozwala tworzyć bazy danych zdefiniowane przez użytkownika i zarządzać nimi. Podstawowa składnia CREATE DATABASE jest następująca: BOL12:



*database\_name* - reprezentuje nazwę nowej bazy danych.

*ON* - pozwala wyraźnie zdefiniować pliki bazy danych.

*PRIMARY* - wskazuje, że pierwszy plik przedstawiony w *file\_definition* staje się podstawowym plikiem danych.

*LOG ON* - umożliwia wyraźne zdefiniowanie plików dziennika.

*file\_definition* - określa listę plików i ich właściwości.

*NAME* = *logical\_file\_name* - określa nazwę logiczną.

*FILENAME* = *'physical\_file\_name'* - określa fizyczną nazwę pliku.

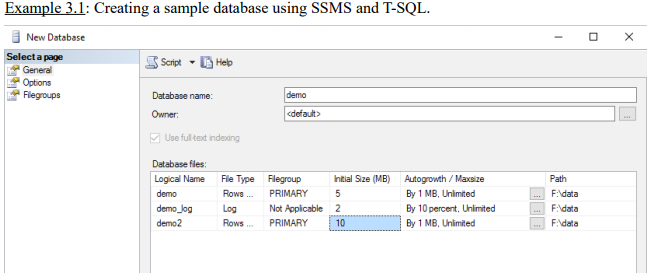
*SIZE* = *file\_szie* - określa rozmiar pliku.

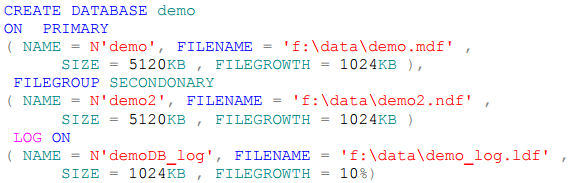
*MAXSIZE* = *max\_szie* - określa maksymalny rozmiar pliku. Plik może wzrastać do osiągnięcia

max\_size.

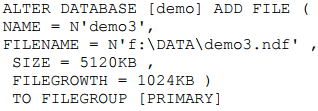
*FILEGROWTH* = *growth\_value* - automatycznie zwiększa plik, gdy nie ma wystarczającej ilości miejsca

Przykład 3.1: Tworzenie przykładowej bazy danych przy użyciu SSMS i T-SQL.





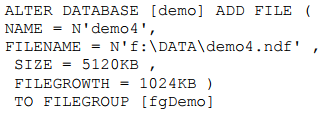
Przykład 3.2: Dodawanie pliku do demo bazy danych.



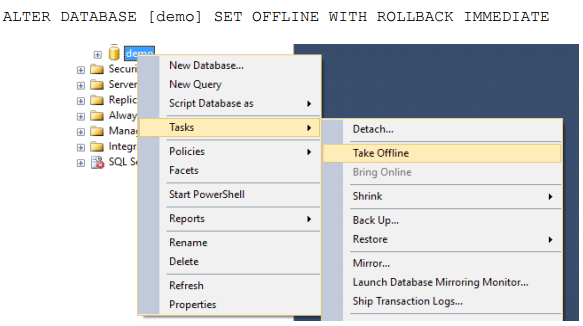
Przykład 3.3: Dodawanie grupy plików o nazwie fgDemo.



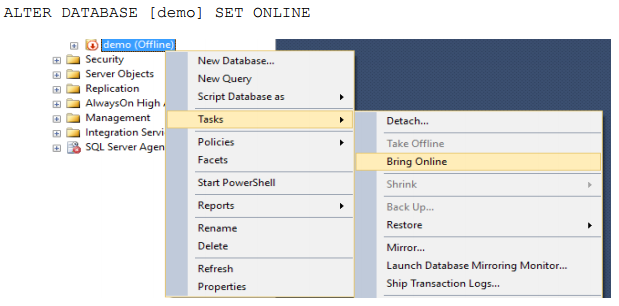
Przykład 3.4: Dodawanie nowego pliku do fgDemo



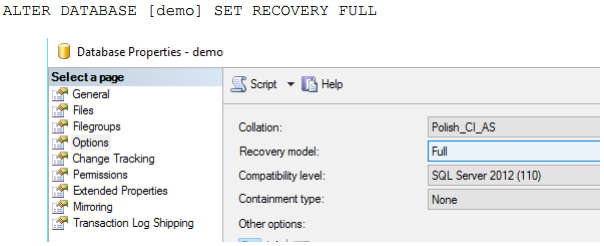
Przykład 3.5: Przenieś bazę danych do stanu offline.



Przykład 3.6: Przenieś bazę danych do stanu online

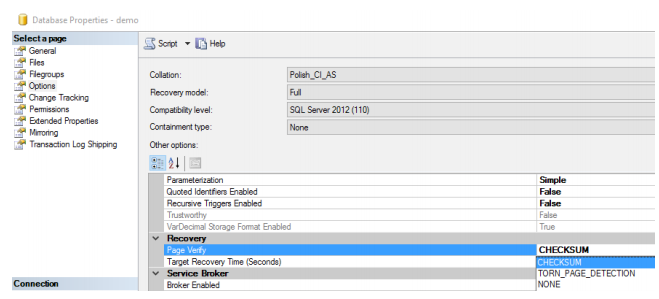


Przykład 3.7: Ustaw pełny model odzyskiwania.



Przykład 3.8: Ustaw stronę opcji w celu sprawdzenia sumy kontrolnej.

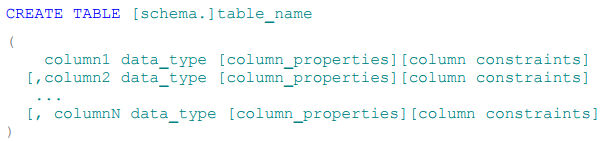




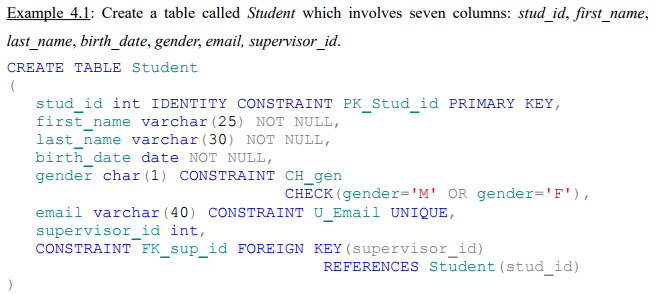
**4 Tworzenie obiektów bazy danych**

**4.1 Tworzenie tabel**

Instrukcja CREATE TABLE jest stosowana do tworzenia tabeli w bazie danych [BOL12]:



Example 4.1: Create a table called Student which involves seven columns: stud\_id, first\_name, last\_name, birth\_date, gender, email, supervisor\_id.



Właściwości kolumn

Najbardziej popularnymi właściwościami kolumny są:

• IDENTITY(start\_with, increment\_by)- automatycznie generuje unikatowy ciąg

liczby.

• NULL - pozwala na wartości null w kolumnie. Domyślnie NULL.

• NOT NULL - zabrania wartości null w kolumnie.

Ograniczenia kolumn

• PRIMARY KEY - wymusza unikalność kolumny i zabrania wartości NULL.

• KLUCZ ZAGRANICZNY - wymaga tylko wartości, które występują w odnośnym odnośniku

kolumna.

• UNIKALNE - gwarantuje, że wartości w kolumnie są różne.

• CHECK - ogranicza wprowadzone wartości do wartości zgodnych z określonym

stan.

• DEFAULT - pozwala zdefiniować domyślną wartość kolumny.

**4.2 Główne typy danych SQL Server**

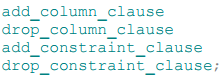
SQL Server udostępnia następujące typy danych [BOL12]:

|  |  |
| --- | --- |
| Data type | Description |
| char[(size)] | Typ danych o długości nieprzetworzonej, nie-Unicode o długości bajty (rozmiar <<, 8000>). Zawsze ma rozmiar pamięci rozmiar bajtów (nawet jeśli wstawiona wartość jest krótsza). Domyślny długość wynosi 1 bajt, MAX oznacza maksymalny rozmiar pamięci do 2 ^ 31 - 1 bajty. |
| varchar[(size) | MAX] | Typ danych o długości zmiennej, nieżododstawowy o długości  bajty (wielkość <1 8000>). Ma rozmiar pamięci równy rzeczywistemu długość włożonej wartości + 2 bajty. |
| nchar[(size)] | Typ danych o długości nieprzetworzonej, nie-Unicode o długości bajty (rozmiar ≤ 1, 4000>). Zawsze ma rozmiar pamięci dwa razy bajtów rozmiaru. |
| nvarchar[(size) | MAX] | Typ danych o długości zmiennej, nieżododstawowy o długości  bajty (rozmiar ≤ 1, 4000>). Ma rozmiar pamięci równy dwóm  razy rzeczywistej długości wstawionej wartości + 2 bajty. |
| int | Typ danych całkowitych z zakresu: 2 ^ 31 to 2 ^ 31-1 (magazyn: 4B). |
| smallint | Typ danych całkowitych w przedziale od 2 ^ 15 do 2 ^ 15-1 (przechowywanie: 2B). |
| tinyint | Typ danych całkowitych w przedziale od 0 do 255 (przechowywanie: 1B). |
| decimal(p, [s])  numeric(p, [s]) | Dziesiętne typy danych o stałej precyzji p (liczba maksymalna  cyfr dziesiętnych zarówno przed, jak i po przecinku dziesiętnym) i skala (maksymalna liczba cyfr po przecinku dziesiętnym punkt); (p <1, 38>, s <0, p>). Rozmiar pamięci mieści się w zakresie od 5 do 17 w zależności od p. |
| float[(m)] | Typ danych o przybliżonym numerze w zakresie: - 1,79E + 308 do -2,23E-308, 0 i 2,23E-308 do 1,79E + 308. Opcjonalne  parametr m oznacza mantysę (m <1, 53>). Rozmiar pamięci  zależy od m i jest równe 4B lub 8B. |
| real | Typ danych o przybliżonym numerze w zakresie: - 3,40E + 38 do -1,18E - 38, 0 i 1,18E - 38 do 3,40E + 38 (przechowywanie 4B). |
| money | Typ danych monetarnych w zakresie: -922.337.203.685.447.5808  do 922, 337, 203, 685477.5807 (przechowywanie: 8B). |
| smallmoney | Typ danych pieniądza w zakresie: - 214,748.3648 do  214,748.3647 (przechowywanie: 4B). |
| date | Typ danych daty z zakresu: 01.01.0001 do 31.12.9999  (dokładność: 1 dzień, przechowywanie: 3B). |
| time | Typ danych czasu z zakresu: 00: 00: 000000000000  23: 59: 59.9999999 (dokładność: 100 nanosekund, przechowywanie: od 3B do 5B). |
| datetime | Typ danych daty powiązany z czasem. Waha się od  Od 01.01.1753 do 31.12.9999 (zakres czasowy: 00:00:00 do  23: 59: 59,997). Rozmiar pamięci jest równy 8B. |
| binary[(n)] | Binarny typ danych o stałej długości o długości n bajtów (n <1, 8000>, przechowywanie: n bajtów). Domyślna długość to 1B. |
| varbinary[(n | MAX)] | Binarny typ długości o zmiennej długości o długości n bajtów (n <1 8000>). Ma rozmiar pamięci równy rzeczywistej długości wstawiona wartość + 2 bajty. MAX oznacza maksymalną pojemność rozmiar równy 2 ^ 31 - 1 bajty. |

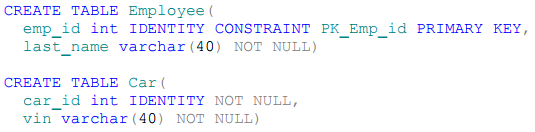
**4.3 Zmiany tabel**

Instrukcja ALTER TABLE jest używana głównie do dodawania, modyfikowania lub upuszczania kolumn / ograniczeń w pliku tabela bazy danych. Podstawowa składnia tej deklaracji jest następująca: BOL12:



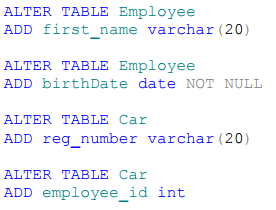


Każda klauzula ma odpowiedni formularz (patrz [BOL12]). Rozważmy następujące instrukcje SQL, które tworzą dwie tabele:



Dodawanie kolumn

Przykład 4.2: dodanie kilku kolumn do obu tabel.



Zmiana kolumn

Przykład 4.3: Zmiana kolumny Employee.last\_name (change the data type of last\_name from varchar(40) to varchar(60)).



Przykład 4.4: Ustaw NOT NULL na Car.employee\_id.



Dodawanie nowych ograniczeń

Przykład 4.5: Utwórz klucz podstawowy w kolumnie Car.car\_id.

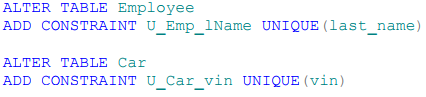


Przykład 4.6: Utwórz cudzysłowy klucz Car.employee\_id, który odnosi się do Employee.emp\_id.





Przykład 4.7: Utwórz UNIQUE ograniczenia na Employee.last\_name i Car.vin.



Upuszczanie kolumn i ograniczeń

Przykład 4.8: Usuń kolumnę Employee.birthDate.



Przykład 4.9: Usuń UNIQUE ograniczenie U\_Emp\_lName



**4.4 Spadek tabel**

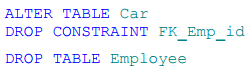
Polecenie DROP TABLE jest stosowane w celu usunięcia tabel bazy danych [BOL12]:



Example 4.10: Drop the table Employee.



Example 4.11: Drop the constraint FK\_Emp\_id and the table Employee.

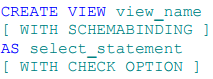


Example 4.12: Drop the table Car



**4.5 Widoki**

Podstawowe stwierdzenie CREATE VIEW ma następującą składnię [BOL12]:



WITH SCHEMABINDING- zapobiega modyfikacjom tabel widoku, które wpływają na jego

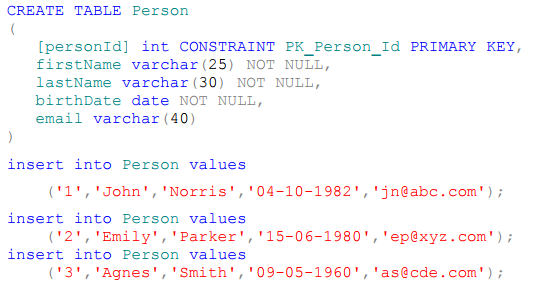
definicja.

select\_statement- definiuje instrukcję select, na której opiera się widok.

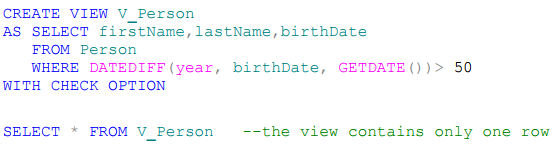
WITH CHECK OPTION - zabrania modyfikacji, która może generować rekordy, które nie są zawarte w zestawie wynikowym select\_statement.

Korzystanie z widoków

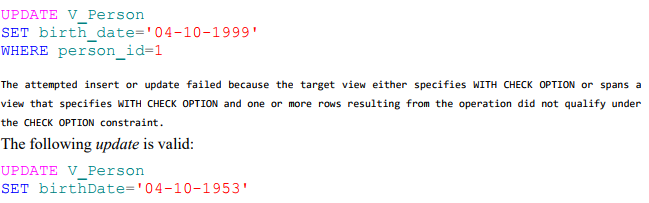
Rozważmy następującą tabelę Person:



Przykład 4.13: Utwórz widok o nazwie V\_Person, który przechowuje nazwy i daty urodzenia osób starszych niż 50.

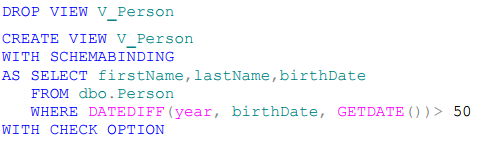


Przykład 4.14: Praca z opcją wyboru.





Example 4.15: Create a view with schemabinding



Z klauzulą schematyczną zabezpiecza przed następującą operacją:



Następujące oświadczenie jest poprawne (w kolumnie e-mail nie występuje w definicji widoku):



**4.6 Indeksy**

Indeksy ([GMUW08, RG03]) to nadmiarowe obiekty bazy danych służące do przyspieszania wyboru zdań. Indeks można uznać za tabelę dwóch kolumn. W pierwszej kolumnie są przechowywane tak zwane wartości kluczowe (np. Nazwiska), a drugie zawierają wartości inne niż kluczowe (np. Lokalizatory wierszy). Kluczowe wartości nie muszą być unikalne i są uporządkowane.

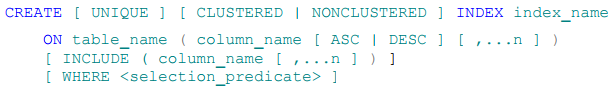
Aby zwiększyć wskaźniki wydajności, są zorganizowane jako zrównoważone drzewa (drzewo B) [GMUW08, RG03]. Indeks ma trzy poziomy. Pierwszy poziom nazywa się root. Drugi to poziom pośredni. Ostatni poziom nazywa się poziomem liści.

SQL Server udostępnia następujące typy indeksów [BOL12]:

* indeks klastrowany - przechowuje całe wiersze tabeli (strony daty) w węzłach liści.
* Tabela, która nie ma indeksu klastrowanego, nazywa się heap.
* Tabela zawierająca indeks klastrowy nazywa się tabelą klastrowaną.
* indeks nieklastrowany - przechowuje wartości kluczowe i wartości inne niż klucze (lokalizatory wierszy) w węzłach na każdym poziomie. Kluczowe wartości są zamawiane.
* indeks unikatowy - jest indeksem klastrowym lub nieklastrowanym, którego kluczowe wartości są unikalne
* indeks z dołączonymi kolumnami - jest indeksem nieklastrowanym, którego węzły liści przechowują również wartości z innych kolumn innych niż klucz.
* filtrowany indeks - jest indeksem nieklastrowanym, które wymagają warunków selekcji.
* indeks pełnotekstowy - obsługuje zapytania mające na celu znalezienie wyrafinowanych fraz w różnych tekstach.
* indeks przestrzenny - dedykowany do danych przestrzennych.
* indeks XML - obsługuje operacje wykonywane na dużych obiektach binarnych XML.

**4.6.1 Tworzenie indeksów**

Podstawowe instrukcje CREATE INDEX są następujące: [BOL12]:



UNIQUE - tworzy stół na unikatowym indeksie.

CLUSTERD - tworzy stół indeksowany na stole.

NONCLUSTERED - tworzy indeks nieklastrowany w tabeli (domyślnie opcja NONCLSTERD).

index\_name - jest to nazwa indeksu.

nazwa\_tabeli - jest to nazwa tabeli.

column\_name - jest to kolumna table\_name, na której tworzony jest indeks. ASC porządkuje klucze indeksu (wartości przechowywane w kolumnie nazwa\_) w porządku rosnącym. DESC sortuje te wartości w kolejności malejącej. ASC jest domyślnym porządkiem sortowania.

INCLUDE nazwa\_kolumny - określa, że wartości z kolumny column\_name będą przechowywane w węzłach liści indeksu klastrowanego.

WHERE selection\_predicate - powoduje, że tylko wiersze odpowiadające predykcie wyboru będą używane do utworzenia indeksu nieklastrowanego.

Indeksy próbek

Rozważmy następujący stół Student:

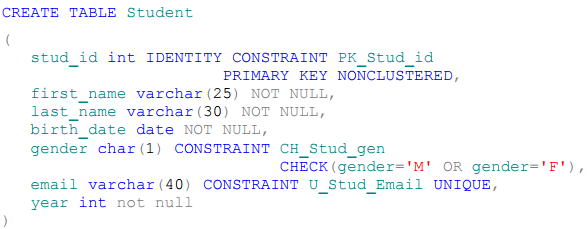
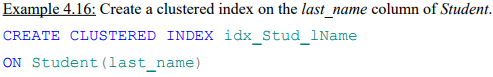


Tabela Student ma dwa indeksy:

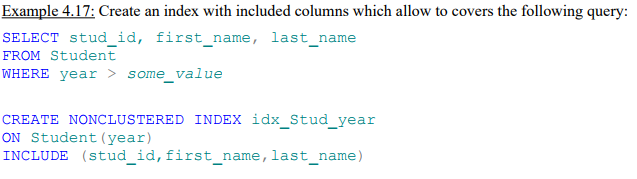
• unikatowy indeks nieklastrowany na kolumnie stud\_id

• unikatowy indeks nieklastrowany w kolumnie wiadomości e-mail

Przykład 4.16: Utwórz indeks klastrowany w kolumnie last\_name Student



Przykład 4.17: Utwórz indeks z dołączonymi kolumnami, które pozwalają pokrywać następujące zapytanie:

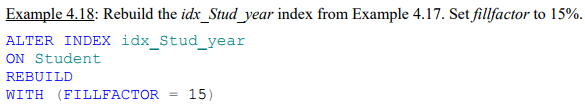


**4.7 Zmiana indeksów**

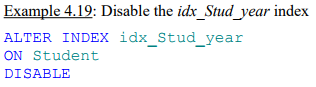
Odbudowywanie indeksów

Administratorzy baz danych powinni okresowo odbudowywać indeksy, aby defragmentować dane przez nich zajmowane.

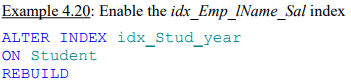
Przykład 4.18: Odbuduj indeks idx\_Stud\_year z przykładu 4.17. Ustawić wypełnienie do 15%



Przykład 4.19: Wyłącz indeks idx\_Stud\_year



Przykład 4.20: Włącz indeks idx\_Emp\_lName\_Sal



**4.8 Praca z indeksami**

Wykonaj następujące czynności:

1. Utwórz tabelę big1 w bazie danych demoDB:



2. Włącz statystyki wejść / wyjść:



3. Uwzględnij rzeczywisty plan wykonania (Ctrl + M) i uruchom kwerendę (1):



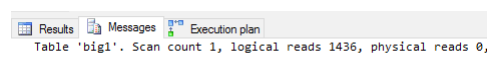
4. Sprawdź następujący plan wykonania zapytania (1) i statystyki I/ O.



5. Run the query (2):



6. Zbadaj następujące statystyki I / O (2).



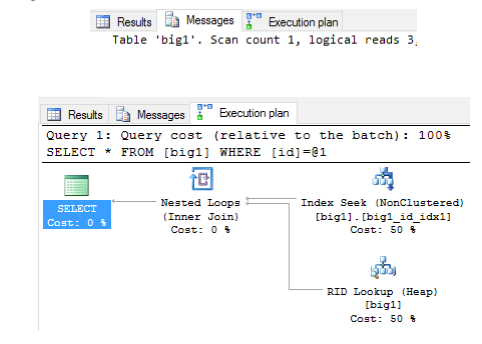
7. Porównaj plany realizacji (1) i (2) oraz ich statystykę I / O.

8. Utwórz indeks nieklastrowy w kolumnie id big1.



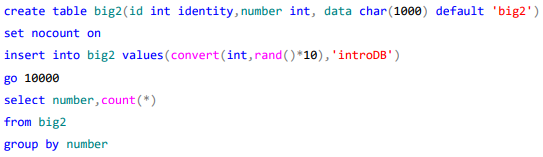
9. Run the queries (1) and (2).

10. Plan wykonania (1) nie powinien się zmienić. Plan i statystyki I / O dla zapytania (2) powinny być następujące:



11. Wyklucz bieżący plan wykonania (Ctrl + M) i wyłączyć statystyki I / O (ustawione statystyki io off).

12. Uruchom następujące instrukcje:



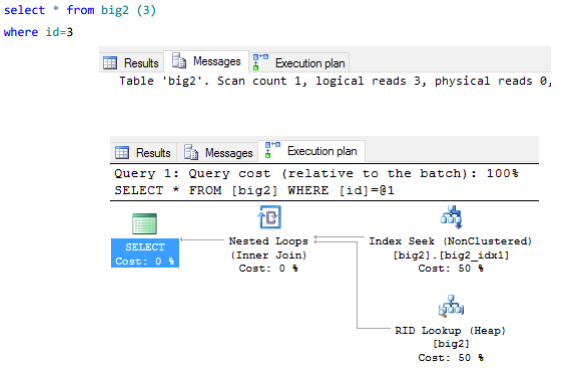
13. Utwórz następujące indeksy:





14. Uwzględnij rzeczywisty plan wykonania (Ctrl + M) i włącz statystyki I / O.

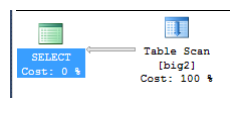
15. Plan wykonania kwerendy (3) i jej statystyk we / wy powinny być następujące:



15. Dlaczego zapytanie (4).



ma następujący plan wykonania?

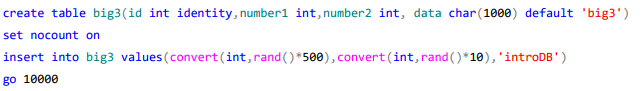


16. Zmień kwerendę (4), aby uzyskać następujący plan wykonania



17. Wyklucz bieżący plan wykonania (Ctrl + M) i wyłączyć statystyki I / O (ustawione statystyki io off).

18. Uruchom następujące instrukcje:



19. Utwórz następujące indeksy:

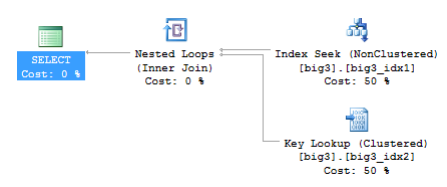


20. Dołącz plan realizacji rzeczywistych (Ctrl + M) i włącz statystyki I / O (ustaw statystyki io na).

21. Uruchom następujące zapytanie (5):



22. Interpretować plan realizacji (5). Powinien mieć następującą postać:



23. Napisz kwerendę zawierającą tabelę big3, która może mieć następujący plan wykonania.



24. Napisz kwerendę zawierającą tabelę big3, która może mieć następujący plan wykonania.



25. Wyświetl informacje o indeksach big3.



26. Wyświetl szczegółowe informacje o indeksach big3.



27. Zanotuj wartości zapisane w kolumnie avg\_fragmentation\_in\_percent of

sys.dm\_db\_index\_physical\_stats dla indeksu big3\_idx2.

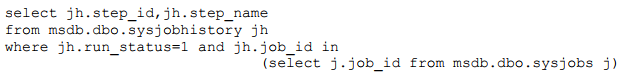
28. Uruchom następującą instrukcję aktualizacji i sprawdź procent fragmentacji (powinien on się zwiększyć):



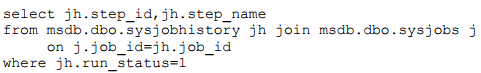
29. Odbuduj indeks big3\_idx2. Powinno zmniejszyć fragmentację fragmentu.

**5 Optymalizacja kwerend**

Istnieją dwa wspólne podejście do optymalizacji zapytań: optymalizacja reguł i optymalizacja kosztowa. Pierwsze podejście oparte jest na przepisywaniu zapytania. Optymalizator zapytania próbuje przepisać daną kwerendę do semantycznie równoważnego formularza, co powinno zagwarantować lepszą wydajność. Na przykład następujące zagnieżdżone zapytanie



można zastąpić następującym semantycznie odpowiednikiem formularza



W drugim zapytaniu operator zezwala na wyeliminowanie zagnieżdżonych zapytań. Współczesne systemy zarządzania bazami danych są zaprojektowane do efektywnego przetwarzania zapytań przy użyciu łączeń, a nie złożonych zagnieżdżonych zapytań. Optymalizator zapytania zazwyczaj próbuje przepisać zagnieżdżone zapytania.

Optymalizacja kosztów obejmuje między innymi następujące kwestie:

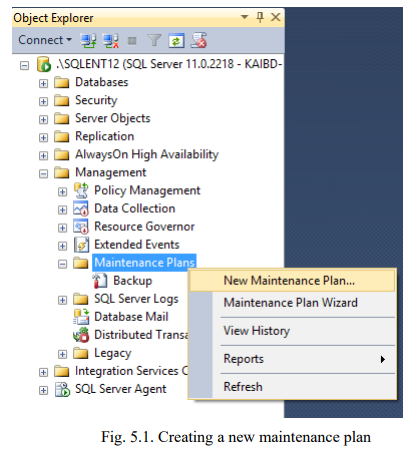
• Wybierz najlepsze indeksy.

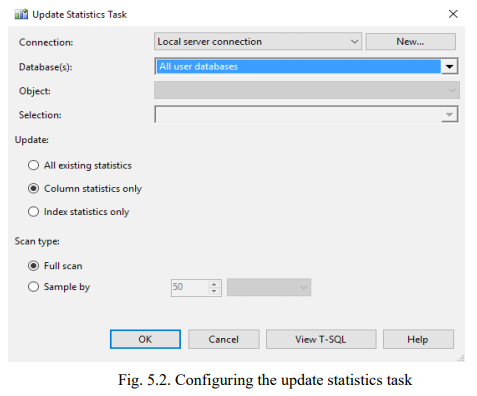
• Wybierz najlepszy zbiór.

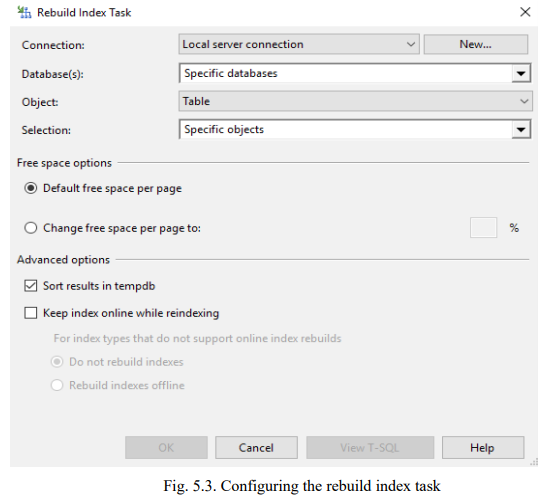
• Wybierz najlepszą operację łączenia (np. Zagnieżdżone pętle, łączenie łączenia, łączenie mieszań).

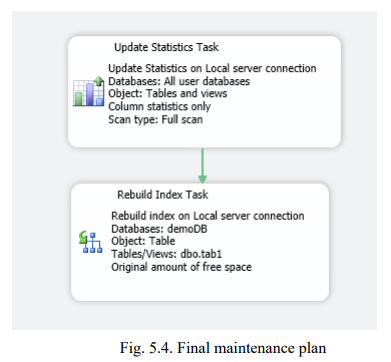
Optymalizator zapytań decyduje na podstawie statystyk. Administratorzy baz danych powinni regularnie odbudowywać statystyki bazy danych. Jeśli są nieaktualne, optymalizator zapytania może wybrać nieefektywny plan wykonania dla danego zapytania. Ważne jest również regularne odbudowywanie indeksów.

DBA może utworzyć tzw plan utrzymania w SQL Server Management Studio, który pozwala na odbudowę statystyk i indeksów bazy danych:





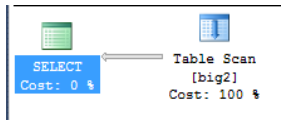




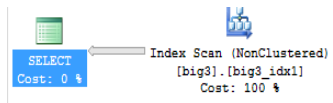
**5.1 Główne elementy planów realizacji programu SQL Server**

Serwer SSMS umożliwia wyświetlanie planów wykonania zapytań użytkowników. Takie plany mogą mieć następujące elementy:

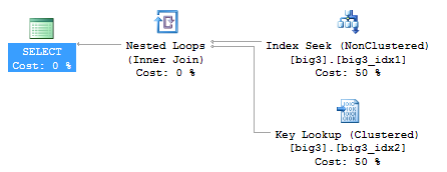
• Table Scan - wykonuje pełne skanowanie tabeli



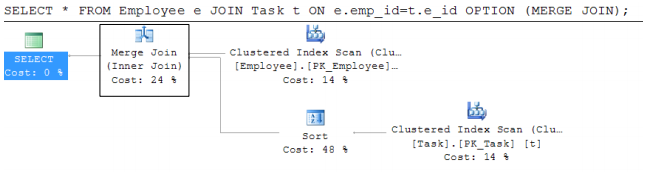
• Index Scan - wykonuje pełne skanowanie indeksu.



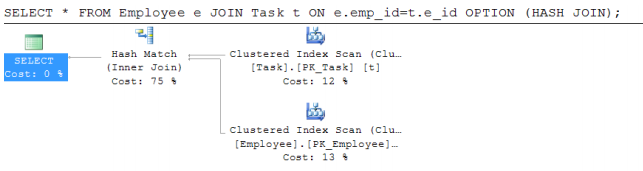
• Nested Loops [Pętle zagnieżdżone] - wykonuje operację łączenia za pomocą łączenia zagnieżdżonych pętli.



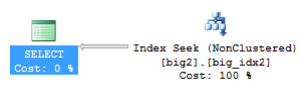
• Merge Join - wykonuje operację join przy użyciu połączenia merge merge.



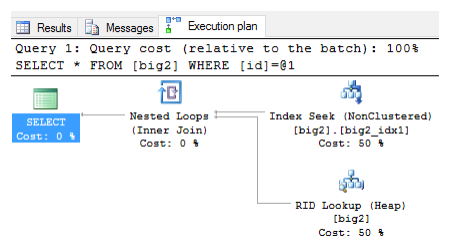
• Hash Join - wykonuje operację join przy użyciu połączenia mieszającego



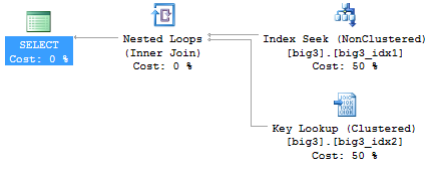
• Nonclustered index seek - przeszukuje indeks nieklastrowy i zwraca prawidłowy identyfikator wiersza (RID).



• RID Lookup (Heap) - zajmuje wiersz lokalizatora (RID) i wyszukuje sterty, aby uzyskać wiersz z tym RID. RID ma trzy elementy: identyfikator pliku, numer strony, numer wiersza na stronie



• Key Lookup (Clustered) - zakłada się, że tabela Tab3 ma indeks klastrowany. Wyszukiwanie kluczy działa na identyfikatorze wiersza zwracanym przez indeks wyszukiwania. Następnie wyszukuje indeks klastrowany, aby uzyskać wiersz z tym lokalizatorem

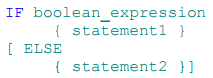


**6 Przechowywane procedury, funkcje i wyzwalacze**

**6.1 Oświadczenia warunkowe**

SQL Server udostępnia dwa oświadczenia warunkowe: if..else i case.

Instrukcja IF..ELSE [BOL12]

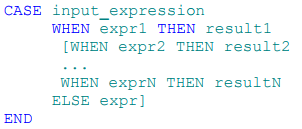


Przykład 6.1: Przykładowe instrukcje if..else.

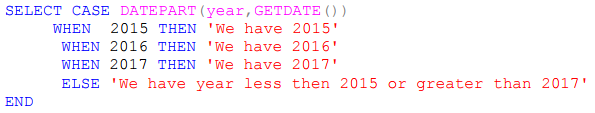




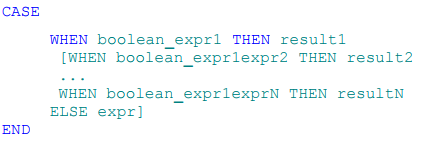
Oświadczenia przypadków mają dwie formy. Pierwsze formy są następujące: [BOL12]:



Przykład 6.2: Przykładowa instrukcja przypadku (pierwsza forma)

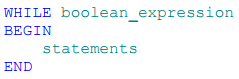


Druga forma orzecznictwa jest następująca: [BOL12]:

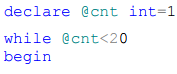


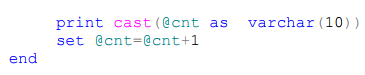
**6.2 Pętle**

Pętla umożliwia wykonywanie bloku instrukcji określonej liczby razy. SQL Server dostarcza klasyczne oświadczenie pętli while [BOL12]:



Przykład 6.3: Próbka podczas wypowiedzi.





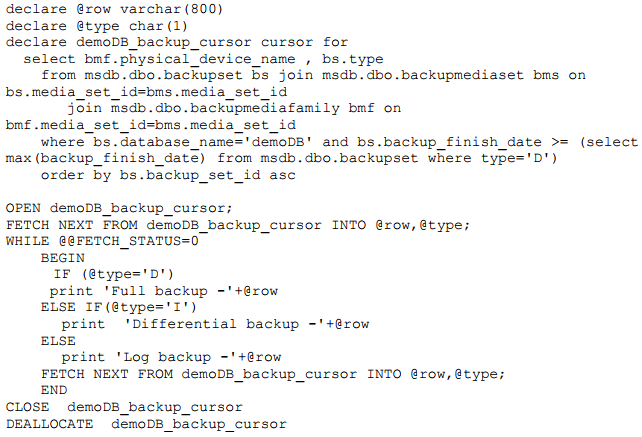
**6.3 Kursory**

Do odczytu obszaru, w którym program SQL Server zapisuje wynik zapytania, stosuje się kursor. Podstawowa składnia deklarująca kursor jest następująca:



Przykład 6.4: Przykładowy kursor

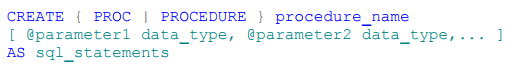
Poniższy kursor zwraca informacje o ostatnich kopiach zapasowych bazy danych demoDB.



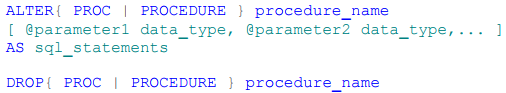
**6.4 Przechowywane procedury**

Procedura przechowywana jest ciągiem instrukcji T-SQL przechowywanych na serwerze.

Podstawowa składnia instrukcji CREATE PROCEDURE jest następująca: [BOL12]:



ALTER PROCEDURE i DROP PROCEDURE pozwalają odpowiednio modyfikować i usuwać procedury [BOL12].



Przykład 6.5. Tworzenie procedury bez parametrów.



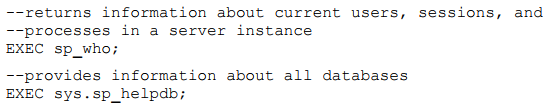
Przykład 6.6. Wywoływanie procedury.



**Procedury przechowywane w systemie**

SQL Server udostępnia wiele procedur przechowywanych w systemie. Są one głównie stosowane do celów administracyjnych.

Przykład 6.7. Wywoływanie procedur systemowych.

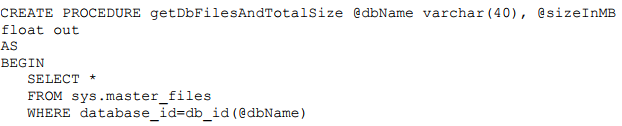


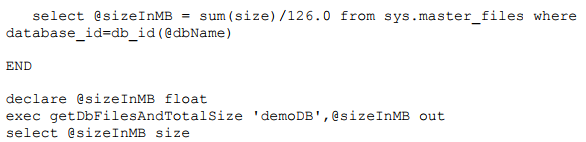
**Procedury z parametrami**

Przykład 6.8. Tworzenie i wywoływanie procedury z parametrem wejściowym.



Przykład 6.9. Tworzenie i wywoływanie procedury z parametrem wejściowym i wyjściowym.



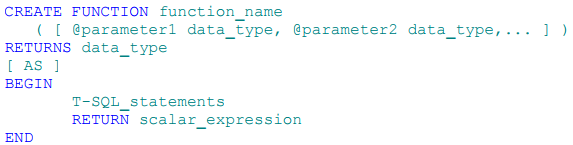


**6.5 Funkcje**

Funkcja to sekwencja instrukcji T-SQL przechowywanych na serwerze. Może pobierać parametry i zwracać pewną wartość.

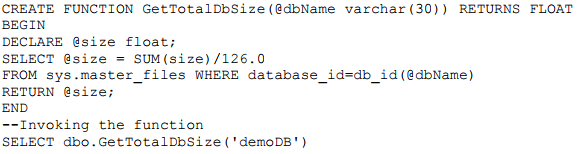
**Funkcje skalarne**

Podstawowa składnia wygląda następująco: [BOL12]:



Funkcje ALTER FUNCTION i DROP FUNCTION umożliwiają modyfikowanie i usuwanie funkcji.

Przykład 6.10. Przykładowa funkcja skalarna, która przyjmuje jeden parametr.



**Funkcje wycenione w tabeli**

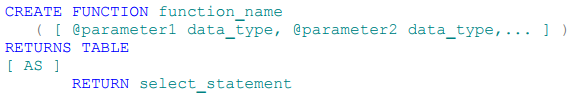
Są dwa typy tych funkcji [BOL12]:

• funkcje liniowe w tabeli

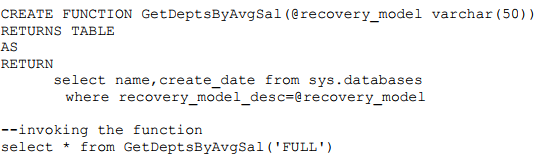
• funkcje wycenione w wielu tabelach

**Funkcje liniowe w tabeli**

Podstawowa składnia wygląda następująco: [BOL12]:

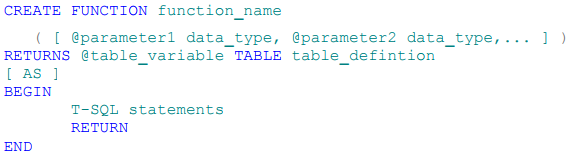


Przykład 6.11. Przykładowa funkcja inline table-valued z jednym parametrem.

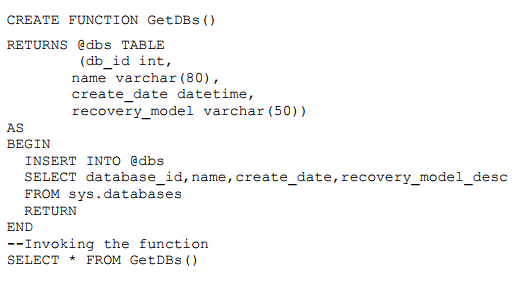


**Wielofirmowe funkcje tabeli**

Podstawowa składnia wygląda następująco: [BOL12]:



Przykład 6.12. Przykładowa wielo-deklarowana tabela wartość funkcji z jednym parametrem.



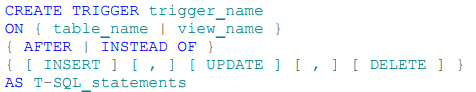
**6.6 Triggers**

Wyzwalacz jest rodzajem procedury przechowywanej, którą SQL Server uruchamia się automatycznie po prawidłowym wystąpieniu. Obsługiwane są następujące wyzwalacze:

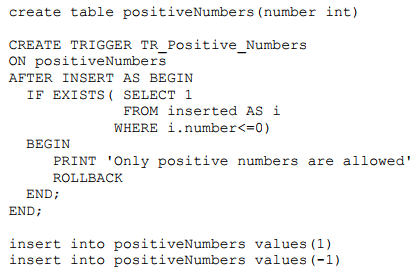
* DML triggers - są uruchamiane podczas wykonywania instrukcji DML (INSERT, UPDATE, DELETE) na tabelach lub widokach.
* DDL triggers - są uruchamiane, gdy użytkownicy wykonują różne instrukcje DDL (na przykład CREATE, ALTER, DROP)
* Logon triggers - są uruchamiane, gdy użytkownicy logują się do instancji SQL Server

**Wyzwalacze DML**

Składnia wygląda następująco: BOL12:

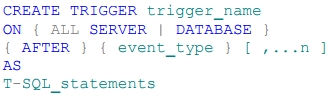


Przykład 6.13. Używając wyzwalacza DML, aby można było wstawić tylko pozytywne liczby do tabeli o nazwie positiveNumbers.



**Wyzwalacze DDL**

Podstawowa składnia wygląda następująco: [BOL12]:

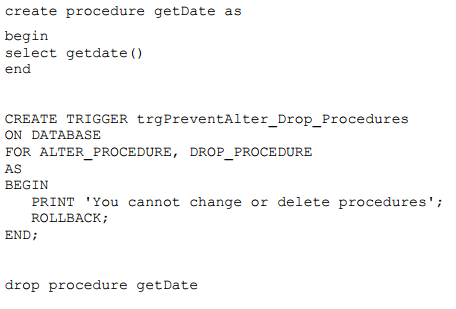


ALL SERVER – zakres wyzwalania obejmuje bieżące wystąpienie.

DATABASE – zakres wyzwalania obejmuje bieżącą bazę danych.

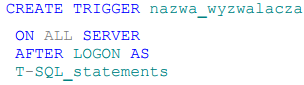
event\_type – definiuje zdarzenia DDL (np. CREATE TABLE, DROP TABLE, ALTER TABLE, 42 CREATE PROCEDURE, ALTER PROCEDURE, CREATE INDEX, ALTER INDEX, etc.)

Przykład 6.14. Używanie wyzwalacza DDL, aby zapobiec modyfikacji lub usunięciu indeksów. Utwórz tabelę tab1 (id int primary key)

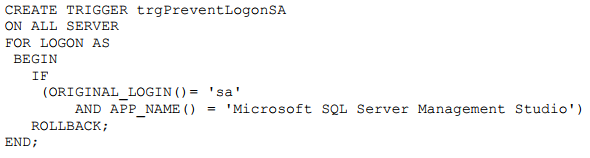


**Wyzwalacze logowania**

Podstawowa składnia wygląda następująco: [BOL12]:



Przykład 6.15. Użyj wyzwalacza LOGON, aby zapobiec logowaniu jako sa.



7 Bezpieczeństwo i autoryzacja użytkowników w programie SQL Server

SQL Server wprowadza następujące podmioty do zapewnienia bezpieczeństwa i kontroli dostępu

mechanizm [BOL12]:

• securables - reprezentują jednostki, do których może być kontrolowany dostęp,

• zleceniodawcy - reprezentują podmioty, które mogą uzyskać dostęp do sekurytyzacji.

Securables występują na trzech poziomach:

• poziom instancji - bazy danych, loginy, role serwera, punkty końcowe.

• poziom bazy danych - użytkownicy baz danych, schematy, role bazy danych, certyfikaty, klucze itp.

• poziom schematu - tabele, widoki, procedury, funkcje, typy itp.

Następujące poziomy są odpowiednie dla zleceniodawców:

• Poziom Windows - logowania i grupy systemu Windows.

• poziom instancji - logowanie SQL Server, role serwera.

• poziom bazy danych - użytkownicy baz danych, baza danych i role aplikacji.

7.1 SQL Server authentication

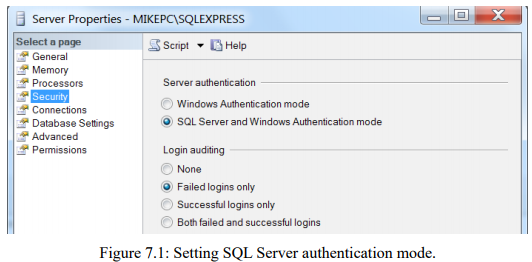
Authentication is aimed at checking an identity. During this process the system verifies whether a user is who he claims he is. SQL Server provides two authentication modes [BOL12]:

• Windows authentication mode – in this mode SQL Server accepts credentials obtained from Windows.

• Mixed mode (Windows and SQL Server authentication mode) – in this mode a user identity can be checked either by SQL Server or by Windows operating systems.

SQL Server Management Studio (SSMS) also allows to manipulate the authentication

mode (Right-click on the server instance – Select Properties and then select Security):



Biorąc pod uwagę powyższe tryby uwierzytelniania, istnieją dwa główne typy logów używane do łączenia się z wystąpieniem programu SQL Server:

• Logowania systemu Windows - są one oparte na kontach Windows lub grupach.

• Loginy SQL Server - są zarządzane bezpośrednio przez SQL Server.

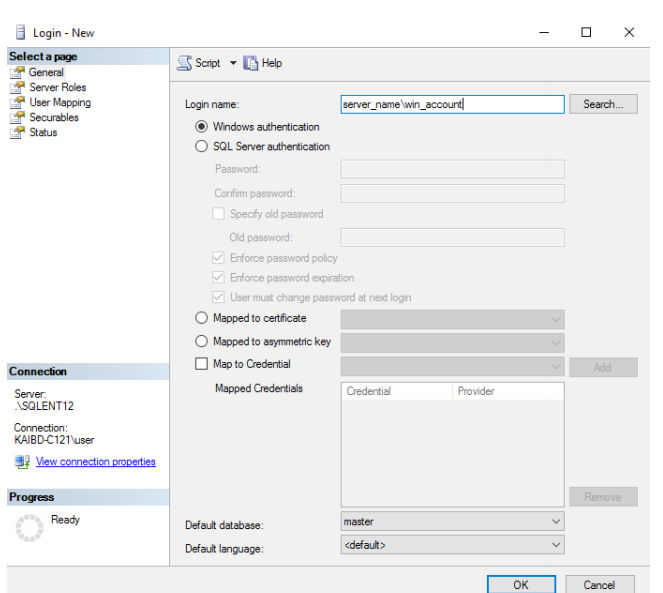
Logowania systemu Windows

Następująca składnia jest używana do utworzenia loginu systemu Windows [BOL12]:



Example 7.1: Creating Windows logins.

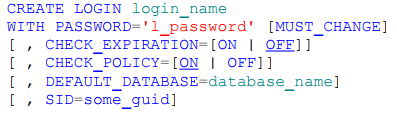






Loginy serwera SQL

Podstawowa składnia używana do tworzenia logowań serwera SQL jest następująca: BOL12:



login\_name - reprezentuje nazwę logowania SQL Server.

l\_password - jest to hasło logowania.

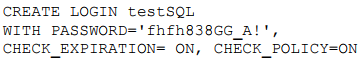
MUST\_CHANGE - zmusza wymianę hasła podczas pierwszego logowania.

CHECK\_EXPIRATION - zmusza zasady wygaśnięcia hasła do logowania.

CHECK\_POLICY - zmusza zasadę hasła systemu Windows.

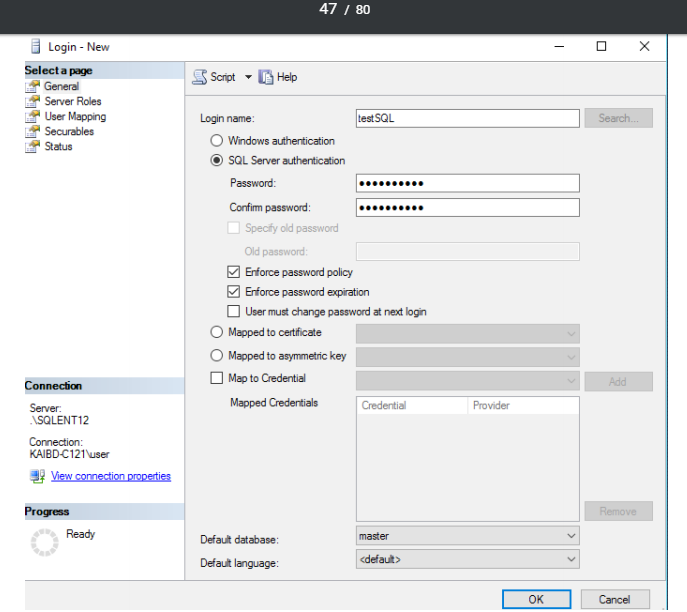
SID some\_guid - określa identyfikator globalny (GUID) dla logowania.

Example 7.2: Creating sample SQL Server login.



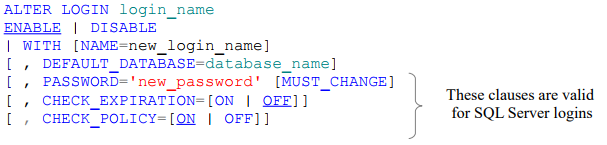
Example 7.3: Creating logins in SSMS.

Serwery SSMS umożliwiają tworzenie logów Windows i logowania serwera SQL Server. Kliknij prawym przyciskiem myszy Security folder w Object Explorer – Select New, a następnie wybierz Login – Fill the General page.



**Zmiana wpisów**

Instrukcja ALTER LOGIN jest używana do modyfikowania logowania systemu Windows i logowania serwera SQL Server [BOL12]:



Przykład 7.4: Zmiana nazwy logowania z loginOLD w celu zalogowaniaNEW.



Przykład 7.5: Zmiana hasła logowania.



Przykład 7.6: Wyłączenie logowania.



**Usuwanie logowań**

Polecenie DROP LOGIN jest używane do usuwania logowań:



7.2 Autoryzacja SQL Server

Konieczne jest przypisanie uprawnień do sekurytyzacji. Proces ten jest nazywany

upoważnienie. SQL Server udostępnia kilka metod implementacji autoryzacji [BOL12]:

• przypisać loginy do ról, które wymagają odpowiednich uprawnień,

• przypisać uprawnienia do logowania przy użyciu instrukcji GRANT,

• tworzenie użytkowników bazy danych i łączenie ich z logowaniami.

Zgoda może być przyznawana na następujących poziomach:

• poziom instancji (jest to najwyższy poziom),

• poziom bazy danych (drugi poziom),

• poziom schematu (poziom trzeci),

• poziom obiektu schema (czwarty poziom).

Role

Role mogą być rozumiane jako kontenery dla uprawnień. SQL Server obsługuje cztery typy ról:

• stałe role serwera,

• Role serwera zdefiniowane przez użytkownika,

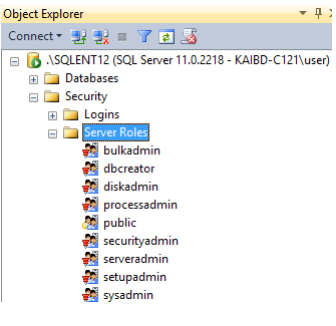
• stałe role bazy danych,

• ról bazy danych zdefiniowanych przez użytkownika.

Poprawiono role serwera

SQL Server udostępnia zestaw role zdefiniowane na serwerze. Poniższa tabela przedstawia roli serwera stałego i ich uprawnienia [BOL12]:

|  |  |  |
| --- | --- | --- |
| Role | Permission | Description |
| sysadmin | CONTROL SERVER | Umożliwia wykonywanie jakichkolwiek operacji w instancji |
| securityadmin | ALTER ANY LOGIN | Umożliwia zarządzanie loginami i ich uprawnieniami |
| dbcreator | ALTER ANY DATABASE | Pozwala modyfikować (tworzyć, zmieniać, upuszczać) i przywracać dowolną bazę danych |
| processadmin | ALTER ANY CONNECTION ALTER SERVER STATE | Pozwala zakończyć istniejące procesy |
| setupadmin | ALTER ANY LINKED SERVER | Umożliwia manipulowanie połączonymi serwerami |
| serveradmin | ALTER ANY ENDPOINT, ALTER RESOURCES, VIEW SERVER STATE, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN | Umożliwia manipulowanie opcjami konfiguracji serwera i wyłączenie serwera |
| diskadmin | ALTER RESOURCES | Umożliwia zarządzanie plikami dyskowymi |
| bulkadmin | ADMINISTER BULK OPERATIONS | Umożliwia zarządzanie operacjami masowymi |



Następujące procedury dotyczą dedykowanych roli serwera [BOL12]:

• sp\_addsrvrolemember 'login\_name', 'role\_name' - powoduje, że login jest członkiem roli.

• sp\_dropsrvrolemember 'login\_name', 'role\_name' - usuwa login z roli.

• sp\_helpsrvrolemember 'role\_name' - wyświetla członków roli.

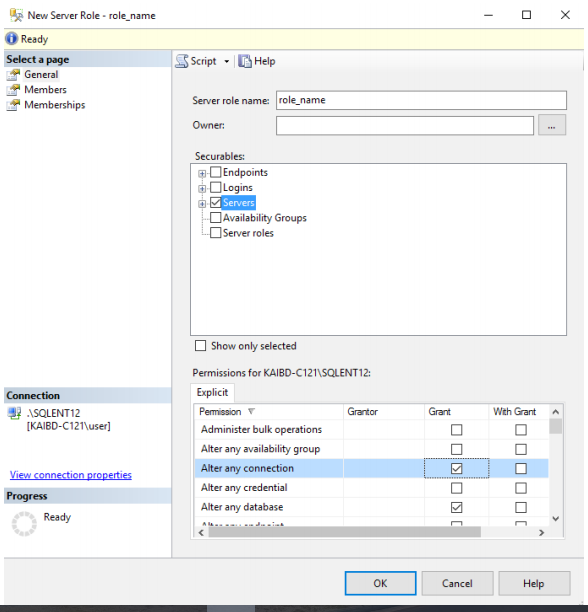
• sp\_srvrolepermission 'role\_name' - wyświetla uprawnienia należące do roli.

• IS\_SRVROLEMEMBER ('role\_name', 'login\_name') - ta funkcja zwraca wartość true, jeśli login jest członkiem roli; w przeciwnym razie zwraca fałsz.

**Rola serwera zdefiniowanego przez użytkownika**

Instrukcja CREATE SERVER ROLE jest stosowana do tworzenia ról serwera zdefiniowanych przez użytkowników:





Poniższa instrukcja umożliwia usunięcie ról serwera zdefiniowanych przez użytkownika:



**Zezwolenia na zakres serwera**

Najczęstsze uprawnienia na poziomie serwera to [BOL12]:

• SERWER SERWERA - umożliwia wykonywanie jakichkolwiek czynności w obrębie instancji.

• ALTER ANY DATABASE - pozwala zarządzać bazami danych w instancji.

• Utworzyć dowolną bazę danych - pozwala tworzyć bazy danych.

• ALTER ANY LOGIN - pozwala zarządzać loginami.

• PRZECZYTAJ OSTROŻNOŚĆ - umożliwia wyświetlanie metadanych dowolnych obiektów wewnątrz instancji.

• ALTER TRACE - pozwala na uruchomienie śladów przeciwko instancji.

• CONNECT SQL - umożliwia połączenie z instancją za pomocą odpowiedniego punktu końcowego.

Instrukcja GRANT umożliwia przyznawanie uprawnień na poziomie serwera [BOL12]:



permission - reprezentuje nazwę uprawnionej do serwera.

principal - reprezentuje główną nazwę (login systemu Windows lub SQL Server, rola serwera), który udziela zezwolenia.

WITH GRANT OPTION- niniejsza klauzula pozwala głównym zobowiązać się do udzielania zgody innym.

Oświadczenie DENY zaprzecza wcześniej przyznanym uprawnieniami na poziomie serwera [BOL12]:



CASCADE - ta opcja nie zgadza się z uprawnieniami permission\_name od osób, którym nadano principal\_name.

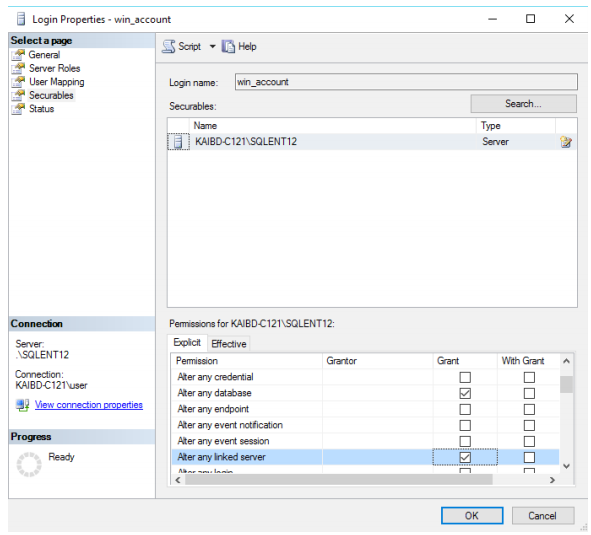
Instrukcja REVOKE usuwa uprzednio przyznane lub zabronione uprawnienia na poziomie serwera [BOL12]:



CASCADE - ta opcja usuwa uprawnienie permission\_name od innych zleceniodawców, którym nadano principal\_name lub jej odmowę.

Przykład 7.7: Przypisywanie uprawnień do serwera do logowania







Przykład 7.8: Tworzenie roli serwera zdefiniowanego przez użytkownika i przypisywanie jej uprawnień.



Przykład 7.9: Usuwanie uprawnień scentralizowanych przez serwer z logowania.



Przykład 7.10: Usuwanie uprawnień scentralizowanych przez serwer z roli serwera zdefiniowanego przez użytkownika.



**Tworzenie użytkowników bazy danych**

Najczęstszym sposobem na umożliwienie loginom dostępu do bazy danych jest tworzenie użytkowników bazy danych. Następująca instrukcja T-SQL jest stosowana do tworzenia użytkowników bazy danych [BOL12]:



db\_user\_name - reprezentuje nazwę użytkownika bazy danych.

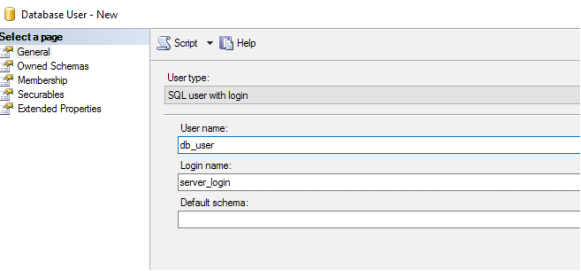
login\_name - reprezentuje nazwę loginu, dla którego użytkownik bazy danych jest db\_user\_name.

DEFAULT\_SCHEMA schema\_name - ustawia domyślny schemat (kontener dla obiektów). Domyślnie schemat dbo jest przypisany do użytkownika.

Przykład 7.11: Tworzenie użytkownika bazy danych.

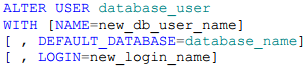
Poniższy skrypt tworzy użytkownika bazy danych o nazwie db\_user dla loginu o nazwie server\_login w bazie danych o nazwie demoDB:





**Zmiana użytkowników bazy danych**

Instrukcja ALTER USER jest stosowana do modyfikowania użytkowników bazy danych [BOL12]:



NAME = new\_db\_user\_name - ustawia nazwę użytkownika na new\_db\_user\_name.

DEFAULT\_DATABASE = nazwa\_bazy\_danych - ustawia domyślną bazę danych na nazwę bazy danych.

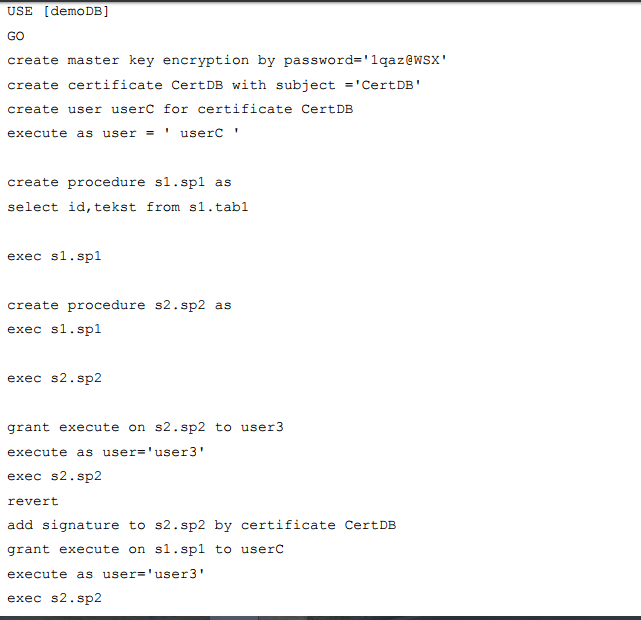
LOGIN = new\_login\_name - przypisuje użytkownikowi login o nazwie new\_login\_name.

Przykład 7.12: Zmiana nazwy użytkownika z userOLD na userNEW w bazie danych demoDB.



**Tworzenie użytkownika na podstawie certyfikatów**

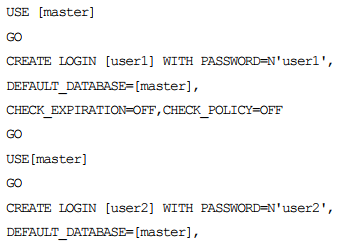
Sprawdź następujący kod i uruchom go na serwerze instancji.

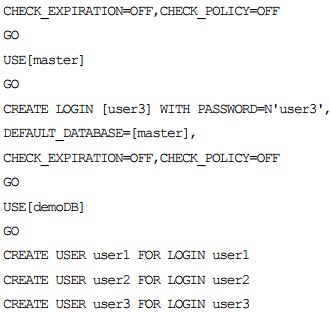


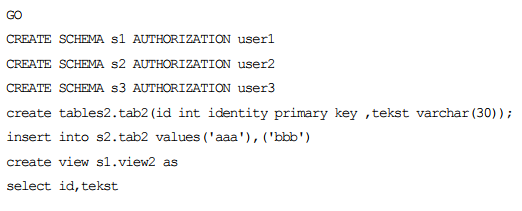


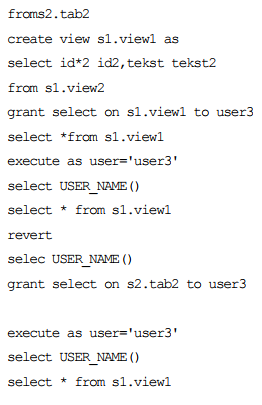
**7.3 Łańcuchy własnościowe**

Sprawdź następujący kod i uruchom go na serwerze instancji.





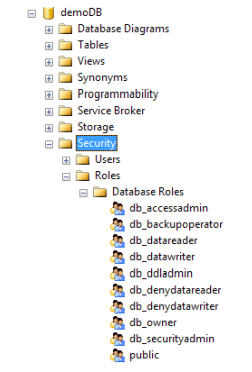




**Poprawiono ról bazy danych**

SQL Server udostępnia zestaw ról na poziomie bazy danych. Następujące predefiniowane role istnieją w każdej bazie danych [BOL12]:

|  |  |  |
| --- | --- | --- |
| Role | Permission | Description |
| db\_owner | CONTROL | Umożliwia zarządzanie całą bazą danych. |
| db\_securityadmin | ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION | Pozwala zarządzać uprawnieniami. |
| db\_accessadmin | ALTER ANY USER CREATE SCHEMA, CONNECT | Umożliwia zarządzanie dostępem do bazy danych. |
| db\_backupoperator | BACKUP DATABASE, BACKUP LOG, CHECKPOINT | Pozwala na tworzenie kopii zapasowych bazy danych. |
| db\_ddladmin | e.g. CREATE TABLE, CREATE VIEW, CREATE PROCEDURE, CREATE FUNCTION, etc. | Umożliwia wykonanie dowolnej instrukcji DDL (create, alter, drop) w bazie danych. |
| db\_datawriter | DELETE, INSERT, UPDATE | Umożliwia wykonanie dowolnej instrukcji DML (wstawianie, aktualizowanie, usuwanie). |
| db\_datareader | SELECT | Pozwala na wykonanie instrukcji SELECT na dowolnych obiektach bazy danych (tabele, widoki). |
| db\_denydatawriter | Denied: SELECT | Zakazuje operacji DML w bazie danych. |
| db\_denydatareader | Denied: DELETE, INSERT, UPDATE | Zakazuje wybranych operacji w bazie danych. |



Następujące procedury dotyczą dedykowanych ról bazy danych [BOL12]:

• sp\_addrolemember 'db\_user\_name', 'db\_role\_name' - powoduje, że użytkownik jest członkiem roli.

• sp\_droprolemember 'db\_user\_name', 'db\_role\_name' - usuwa użytkownika z roli.

• sp\_helprolemember 'db\_role\_name' - wyświetla członków roli.

• sp\_dbfixedrolepermission 'db\_role\_name' - wyświetla uprawnienia należące do roli.

• IS\_ROLEMEMBER ('db\_role\_name', 'db\_user\_name') - funkcja ta zwraca wartość true, jeśli użytkownik jest członkiem roli; w przeciwnym razie zwraca fałsz.

**Role bazy danych zdefiniowane przez użytkownika**

Instrukcja CREATE ROLE [BOL12] jest stosowana do tworzenia ról bazy danych zdefiniowanych przez użytkowników:



Poniższa instrukcja umożliwia usuwanie ról bazy danych zdefiniowanych przez użytkownika:



**Przypisywanie uprawnień użytkownikom bazy danych**

Instrukcja GRANT umożliwia przyznawanie uprawnień do obiektów bazy danych użytkownikom lub rolom bazy danych [BOL12]:



*permission\_name* - reprezentuje nazwę uprawnienia.

*column\_name* - reprezentuje nazwę kolumny

*s\_type* - reprezentuje obiekty takie jak schematy, tabele, procedury itp.)

WITH GRANT OPTION- niniejsza klauzula pozwala głównym zobowiązaniu do udzielenia zgody innym głównym.

Instrukcja REVOKE usuwa uprzednio przyznane lub zabronione uprawnienia [BOL12]:



Przykład 7.14: Udzielanie uprawnień (z opcją grantu) do tworzenia tabel w bazie danych demoDB

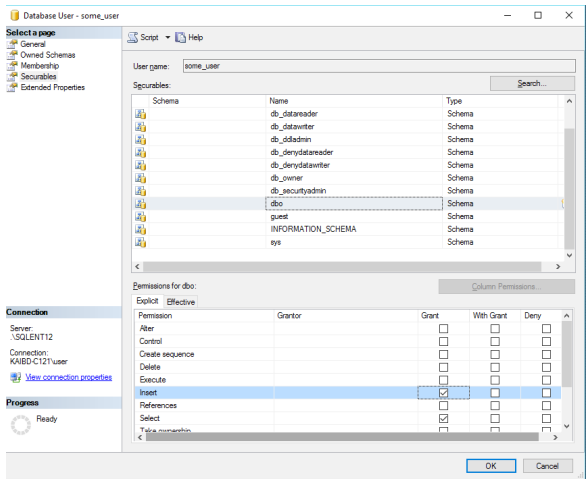


Przykład 7.15: Udziela się zgody na zmianę widoków w bazie danych demoDB.

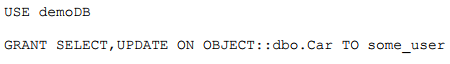


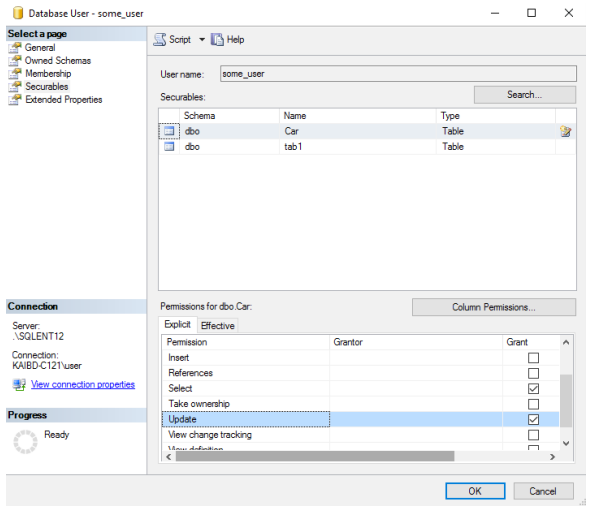
Przykład 7.16: Przyznawanie uprawnień SELECT i INSERT dla schematu dbo dla użytkownika4.





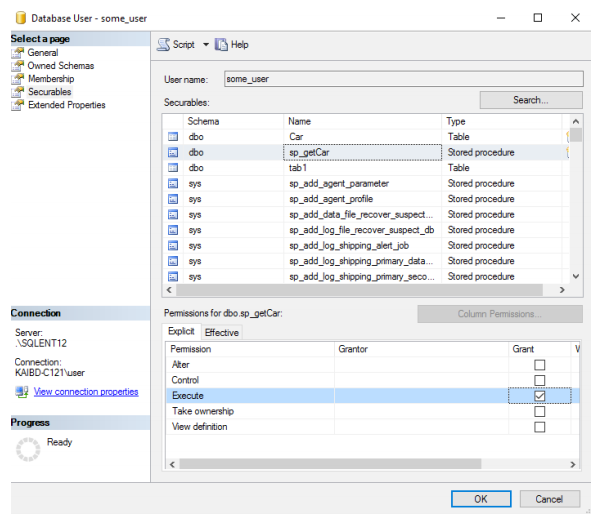
Przykład 7.17: Przyznawanie zgody SELECT i UPDATE w tabeli o nazwie Car.





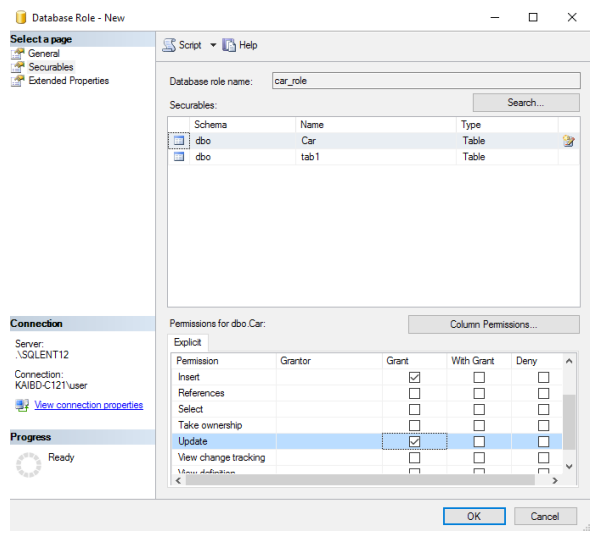
Przykład 7.18: Przyznawanie uprawnień EXECUTE do procedury o nazwie sp\_getCars





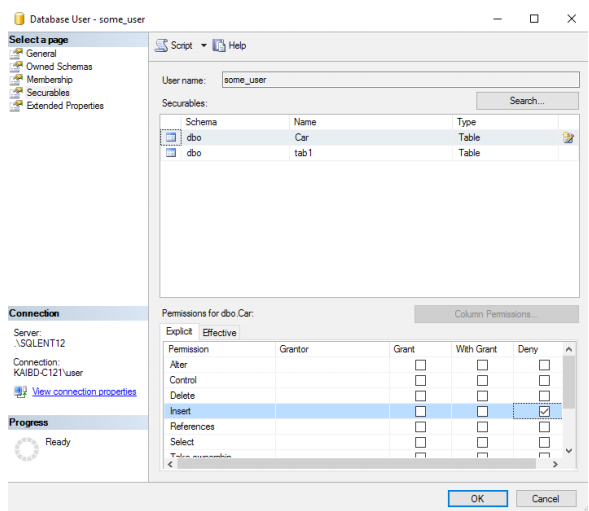
Przykład 7.19: Przyznawanie uprawnień INSERT i UPDATE do tabeli o nazwie Car do roli o nazwie car\_role.





Przykład 7.20: Odmowa uprawnienia INSERT na Car do some\_user.





Przykład 7.21: Odwołanie uprawnienia UPDATE dla schematu dbo do some\_user



Zadanie

1. Sprawdź tryb uwierzytelniania instancji.

2. Ustaw tryb uwierzytelniania w trybie mieszanym.

3. Utworzyć login sa.

4. Zmień nazwę sa na sqlLogAdmin i ustaw silne hasło dla niego. Logowanie jako sqlLogAdmin.

5. Utwórz nowe konto systemu Windows o nazwie sql i zaloguj się do niego w usłudze SSMS.

6. Zaloguj się do instancji jako sql i sprawdź, czy ten login ma dostęp do instancji baz danych.

7. W bazie danych demoDB utworzyć użytkownika bazy danych o nazwie demo dla loginu sql.

8. Sprawdź, czy użytkownik demo może wyświetlać dane z tabel przechowywanych w demoDB.

9. Przyznawaj, wstaw, aktualizuj, usuń uprawnienia wszystkich tabel przechowywanych w schemacie dbo dla demo użytkownika.

10. Wycofaj uprawnienie insertu w schemacie dbo od demo.

11. Dodaj użytkownika demo do roli db\_owner w bazie danych demoDB.

12. Utwórz nową rolę o nazwie test\_rol w bazie danych demoDB.

13. Dodaj to uprawnienie do tej roli.

14. Usuń demo z roli db\_owner. Dodaj demo do roli test\_rol.

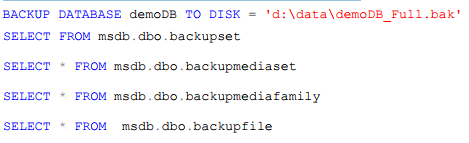
**8 Kopia zapasowa w programie SQL Server**

SQL Server obsługuje następujące kopie zapasowe [BOL12]:

* Pełna kopia zapasowa bazy danych - dotyczy wszystkich stron danych i aktywnej części pliku dziennika.
* Kopia zapasowa bazy danych różniczkowych - dotyczy tylko stron, które zostały zmienione od ostatniej pełnej kopii zapasowej.
* Częściowa kopia zapasowa - zawiera pliki z grupy plików PRIMARY, dowolne grupy plików mające status READ / WRITE, READ ONLY filegroups, jeśli zostaną podane.
* Kopia zapasowa plików grupowych.
* Kopia zapasowa grup plików.
* Kopia zapasowa dzienników transakcji - zawiera aktywną część dziennika i część, która nie występuje w poprzedniej kopii zapasowej dziennika (model odzyskiwania bazy danych musi być ustawiony na pełny lub zbiorczy).
* Kopiuj tylko kopie zapasowe (pełne lub dziennikowe tworzenie kopii zapasowych).

8.1 Pełna kopia zapasowa bazy danych

Wykonaj pełną kopię zapasową bazy danych i wyświetla informacje o nim.



**Przykładowa sekwencja tworzenia kopii zapasowej**

• Pełna kopia zapasowa - raz w tygodniu (w niedzielę o godzinie dwudziestej w nocy).

• Kopia zapasowa różnego rodzaju - raz dziennie od poniedziałku do soboty (o godzinie 8).

• Kopia zapasowa dzienników transakcji - co 5 minut od 9:00 do 19:00.

• Powyższa strategia gwarantuje, że w przypadku awarii tracimy transakcje z maksymalnie 5 minut.

**Nośniki kopii zapasowych**

• Zestaw kopii zapasowych - reprezentuje pojedynczą kopię zapasową.

• Urządzenie kopii zapasowej - pliki lub taśmy.

• Zestaw nośników - może zawierać do 64 urządzeń do tworzenia kopii zapasowych.

**Sprawdzanie walidacji**

• Instrukcja tworzenia kopii zapasowych może liczyć sumę kontrolną. Jest on przechowywany w nagłówku kopii zapasowej.

• Opcja CHECKSUM oznacza następujące operacje.

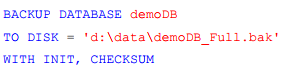
• Sprawdzanie każdej strony danych, jeśli opcja sprawdzania bazy danych strony jest włączona.

• Generowanie sumy kontrolnej dla całej kopii zapasowej.

• Domyślnie wartość NO\_CHECKSUM.

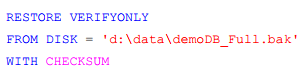
• Instrukcja tworzenia kopii zapasowej z opcją sumy kontrolnej powoduje awarię w przypadku wykrycia uszkodzonych stron danych.

• Opcja CONTINUE\_AFTER\_ERROR nie zatrzymuje operacji tworzenia kopii zapasowych.



• Polecenie RESTORE VERIFYONLY umożliwia sprawdzenie, czy kopia zapasowa jest kompletna i czytelna.

• Opcja CHECKSUM programu RESTORE VERIFYONLY sprawdza sumę kontrolną każdej strony i sumę kontrolną całej kopii zapasowej.

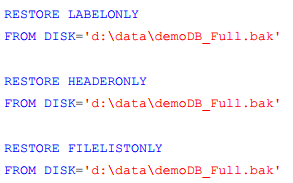


**Informacje o kopiach zapasowych**

• RESTORE LABELONLY - zwraca informacje o zestawach nośników.

• RESOTRE HEADERONLY - zwraca nagłówki zestawów kopii zapasowych z urządzeń kopii zapasowych.

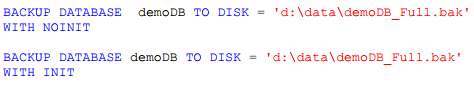
• RESTORE FILELISTONLY - zwraca listę plików danych i plików dzienników zapisanych w zestawie kopii zapasowych.



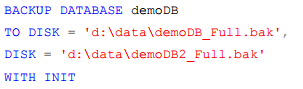
**Przykładowe opcje tworzenia kopii zapasowych**

INIT - zastępuje istniejące zestawy kopii zapasowych przez nowy zestaw kopii zapasowych.

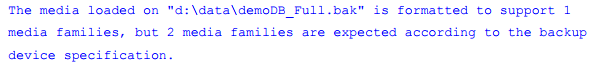
NOINIT - dołącza nowy zestaw kopii zapasowych do określonego zestawu mediów. Jest to ustawienie domyślne.



**Pełna kopia zapasowa na nośniku danych zawierającym dwa pliki danych**

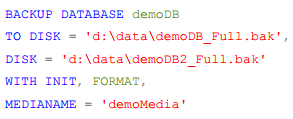


**Powyższe oświadczenie zwraca następujący błąd:**



Rozwiązanie:

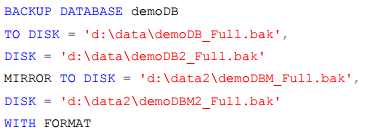
Opcja FORMAT jest potrzebna do modyfikacji struktury istniejących nośników. Powinien być oparty na dwóch plikach, a nie na jednym pliku.



**Opcja MIRROR TO**

• MIRROR TO pozwala wykonywać kilka kopii zapasowych tej samej bazy danych przy użyciu jednej instrukcji.

• Umożliwia określenie do trzech dodatkowych urządzeń kopii zapasowych.



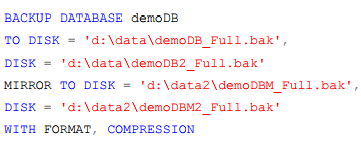
**Uwaga: wymagana jest opcja FORMAT, aby utworzyć zbiór kopii zapasowych**

**Skompresowane kopie zapasowe**

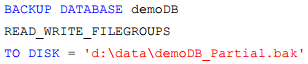
• Opcja COMPRESSION umożliwia utworzenie skompresowanej kopii zapasowej.

• Wymaga co najmniej Standard Edition SQL Server 2008 R2.

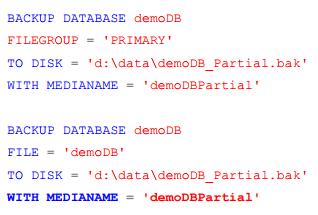
• To wymaga Enterprise Edition programu SQL Server 2008.



Częściowe kopie zapasowe, kopie zapasowe grup plików, kopie zapasowe plików





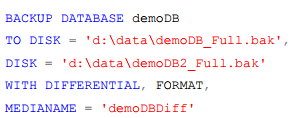


8.2 Różnicowe kopie zapasowe

• Opcja DIFFERENTIAL jest stosowana do wykonywania różnicowych kopii zapasowych.

• Kopia różnicowa obejmuje tylko rozszerzenia, które zostały zmienione od ostatniej pełnej kopii zapasowej.

• Przed wykonaniem różnicowej kopii zapasowej należy wykonać pełną kopię zapasową.



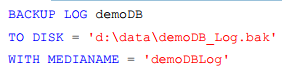
8.3 Kopie zapasowe dzienników transakcji

• instrukcja BACKUP LOG umożliwia wykonywanie kopii zapasowych dzienników transakcji.

• Model odzyskiwania musi być ustawiony na FULL lub BULK-LOGGED, aby wykonać kopie zapasowe dzienników.

• Jeśli model odzyskiwania jest ustawiony na prosty, niemożliwe jest wykonanie kopii zapasowych dzienników.

• Po wykonaniu kopii zapasowej dziennika program SQL Server obcina nieaktywną część pliku dziennika.



Kopia zapasowa Tail-log

• Kopia zapasowa ogona-logów jest archiwizacją dzienników, która obejmuje rejestry dzienników, które nie zostały jeszcze utworzone.

• Początkowo w przypadku awarii bazy danych administrator powinien wykonać kopie zapasowe dziennika.

• Po wykonaniu kopii zapasowej dziennika kopii zapasowych DBA może rozpocząć procedurę przywracania.

• Jeśli pliki danych są uszkodzone, opcja CONTINUE\_AFTER\_ERROR jest potrzebna do wykonywania kopii zapasowych dzienników kopii zapasowych.



**9 Przywracanie bazy danych SQL Server**

  9.1 Proces przywracania

Instrukcja RESTORE umożliwia:

• przywracanie całej bazy danych z pełnej kopii zapasowej (pełne przywrócenie). Po niej może nastąpić przywrócenie kopii zapasowych różnic i dzienników.

• przywracanie określonych grup plików lub plików (częściowe przywrócenie).

• przywracanie określonych stron (przywrócenie strony).

• przywracanie dziennika transakcji (przywrócenie dziennika transakcji).

Proces przywracania bazy danych obejmuje następujące etapy:

1. Faza kopiowania danych - tworzenie plików bazy danych i kopiowanie danych do nich.

2. Redo phase - stosowanie transakcji z plików dziennika do danych, aby przenieść je do określonego punktu przywracania.

3. Etap cofania - cofanie się transakcji bez ograniczeń (kabina zostanie wprowadzona w drugiej fazie).

Faza ponowień i cofania są nazywane odzyskiwaniem bazy danych.

Procedura przywracania próbki

9.2 Przywracanie do punktu awarii

Rozważmy następujący scenariusz:

Baza danych demoDB zawiera tablicę o nazwie tab1 (klucz identyfikacyjny id int, desc varchar (30)), który przechowuje dwa wiersze

• Wykonaj pełną kopię zapasową bazy danych.



• Wstaw dwa wiersze do tabeli tab1.

• Wykonaj różnicową kopię zapasową.



• Wstaw dwa wiersze do tabeli tab1.

• Wykonaj kopię zapasową dzienników transakcji.



• Wstaw trzy rzędy do tabeli tab1.

• Zamknij program SQL Server i usuń tylko pliki danych (przyjmuje się, że plik dziennika nie jest uszkodzony).

• Uruchom program SQL Server i przywróć demonstrację do punktu awarii.

• Wykonaj kopię zapasową dzienników typu tail-log (opcja norecovery - pozostawia bazę danych w stanie przywracania).



• Przywróć pełną kopię zapasową bazy danych (umożliwia kontynuowanie procedury przywracania - faza cofania nie jest wykonywana, możliwe jest przywrócenie kolejnych kopii zapasowych).



• Przywróć różnicową kopię zapasową.



• Przywróć kopię zapasową dziennika



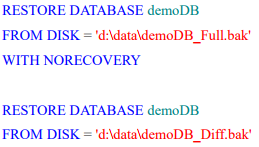
• Przywróć kopię zapasową dziennika zdarzeń (RECOVERY jest domyślnym, powoduje przywrócenie bazy danych z stanu odzyskiwania do stanu online).



9.3 Przywracanie do punktu w czasie

Opcja date\_time\_expression STOPAT w instrukcji RESTORE LOG umożliwia przywrócenie bazy danych do określonego punktu.

Korzystanie z opcji stopat (jeśli określony czas znajduje się w kopii zapasowej dziennika o liczbie k, proces odzyskiwania zostanie zatrzymany w kroku k):





9.4 Przywracanie do zaznaczonych transakcji

Opcja nazwa\_wersji STOPATMARK z RESTORE LOG umożliwia przywrócenie bazy danych do zaznaczonej transakcji (BEFOREMARK - przywraca bez zaznaczonej transakcji).

Używając opcji STOPATMARK:





9.5 Przywracanie baz danych systemu

Przywracanie głównej bazy danych

• DBA musi zawierać niektóre pliki danych i logów głównej bazy danych (wymagane są podstawowe pliki bazy danych, aby uruchomić instancję).

• Alternatywnie DBA może generować program master przy użyciu programu SQL Server Setup (ten proces zastępuje inne bazy danych systemu).

• Program DBA musi uruchomić instancję SQL Server w trybie pojedynczego użytkownika (dodanie -m w opcjach uruchamiania).

• Dba musi uruchomić następujące polecenie w sqlcmd.



• Opcja WYMIANA umożliwia przywrócenie bazy danych, jeśli istnieje już baza o tej samej nazwie.

• Usuń parametr -m z opcji uruchamiania.

• Uruchom instancję SQL Server.

Przywracanie modelu bazy danych

• DBA musi zawierać niektóre pliki danych i plików dziennika modelu bazy danych.

• DBA może generować model przy użyciu programu SQL Server Setup.

• Program DBA musi rozpocząć instancję programu SQL Server z parametrem T3608 (spowoduje to uruchomienie tylko głównej bazy danych).

• DBA musi działać następująco.



Przywracanie innych typów baz danych

• Tempdb - jest odtwarzany podczas uruchamiania instancji (instrukcje tworzenia kopii zapasowych i przywracania nie są dostępne).

• Zasób (baza danych tylko do odczytu, która przechowuje wszystkie obiekty systemu) - można ją przywrócić, używając kopii zapasowych systemów operacyjnych swoich plików lub SQL Server Setup (instrukcje tworzenia kopii zapasowych i przywracania nie są dostępne).

• Msdb - przywracanie tej bazy danych jest analogiczne do przywracania baz danych zdefiniowanych przez użytkownika.