

Administracja i programowanie pod Microsoft SQL Server 2000

Paweł Rajba

pawel@ii.uni.wroc.pl
<http://www.kursy24.eu/>

Zawartość modułu 7

- Indeksy
 - Wprowadzenie
 - Rodzaje indeksów
 - Wybór indeksów
 - wybór indeksu zgrupowanego
 - Tworzenie, usuwanie i przebudowanie indeksów
 - Pobieranie informacji o indeksach
 - Fragmentacja indeksów
 - wykrywanie i pozbywanie się fragmentacji
 - Tworzenie, odświeżanie i wyświetlanie statystyk

Wprowadzenie

- Zalety stosowania indeksów
 - Zwiększają szybkość wyszukiwania wierszy spełniających zadane warunki
 - Przyspieszają operacje złączenia, sortowania i grupowania
 - Pozwalają wymusić unikalność wierszy, jeśli indeks został zdefiniowany z odpowiednim parametrem
 - Dobrze jest zakładać indeks na kolumny o wysokim stopniu selektywności, czyli takie, gdzie wiersze się niewiele powtarzają

Wprowadzenie

- Wady stosowania indeksów
 - Zużywają miejsce na dysku
 - Ogólnie generują koszty związane z utrzymywaniem indeksów, stąd nie należy tworzyć indeksów, które będą rzadko używane

Rodzaje indeksów

- Indeksy zgrupowane
 - Indeks jest B-drzewem, w którym w liściach są nie wskaźniki ale właściwe strony danych z danymi
 - Liście B-drzewa tworzą podwójną listę zwaną łańcuchem stron
 - Kolejność stron w łańcuchu odpowiada kolejności liści w drzewie indeksowym
 - W związku z powyższym można utworzyć tylko jeden taki indeks i jest zalecane taki indeks utworzyć
 - To tak naprawdę nie jest tak, że indeksowane dane są spójne na dysku (jest to technicznie niemożliwe)

Rodzaje indeksów

- Indeksy niezgrupowane
 - Dane z wierszy są pamiętane w stronach, które są połączone w listę dwukierunkową (łańcuch)
 - Kolejność stron w łańcuchu nie jest ustalona
 - Indeks jest niezależnym B-drzewem, w którym liście zawierają wskaźniki na właściwe strony zawierające szukane rekordy
 - kiedy jest indeks zgrupowany, to jest wskaźnik do tego indeksu, a kiedy go nie ma, jest adres strony
 - Można utworzyć maksymalnie 249 indeksów niezgrupowanych

Wybór indeksów

- Co indeksować
 - klucze główne
 - klucze obce oraz kolumny biorące częsty udział w złączeniach tabel
 - kolumny, z których wyciągane są zakresy danych
 - kolumny, których wartości są sortowane
 - kolumny, na których wykonywane jest grupowanie i agregacja wartości

Wybór indeksów

- Czego nie indeksować
 - kolumn, które rzadko są wykorzystywane w zapytaniach
 - kolumn, które mają kilka wartości
 - kolumn o typach danych: text, ntext, image
 - tych kolumn nie da się indeksować

Wybór indeksu zgrupowanego

- Jak dobrać indeks zgrupowany
 - kiedy chcemy zoptymalizować operacje wstawiania do obciążonej tabeli, warto rozpatrzyć utworzenie indeksu na kolumnie będącej identyfikatorem (PK, int)
 - tabele często sortowane, grupowane lub przeszukiwane pod kątem zakresu danych warto indeksować po kolumnie względem której te operacje są wykonywane (np. raporty po dacie)
 - ogólne zasady
 - ograniczyć liczbę kolumn w indeksie
 - jak najmniej znaków w typach varchar i char
 - jak najmniejszy typ danych, np. tinyint zamiast int

Tworzenie indeksów

- Indeks stworzymy poleceniem CREATE INDEX
 - Składnia
CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED]
INDEX *nazwa*
ON { *table* | *view* } (*column* [ASC | DESC] [,...*n*])
[WITH <opcja> [,...*n*]]
[ON *filegroup*]
 - Wybrane opcje
 - FILLFACTOR
 - PAD_INDEX
 - DROP_EXISTING
 - SORT_IN_TEMPDB

Tworzenie indeksów

- Indeks...
 - Opcje
 - FILLFACTOR = fillfactor
 - opcja określa stopień zapełnienia liści w drzewie indeksu (stron indeksów) w momencie
 - tworzenia indeksu
 - przebudowania indeksu
 - opcja ma większe znaczenie przy tworzeniu indeksów zgrupowanych
 - wartości są od 0 do 100 (0 i 100 oznaczają 100% zapełnienie)
 - domyślna wartość to 0, można ją zmienić dla całego serwera poleceniem `sp_configure_fillfactor`

Tworzenie indeksów

- Indeks...
 - Opcje
 - PAD_INDEX
 - określa stopień zapełnienia wierzchołków wewnętrznych drzewa indeksu
 - wartość jest pobierana z opcji FILLFACTOR, stąd jej użycie ma sens wtedy, gdy użyta jest opcja FILLFACTOR
 - DROP_EXISTING
 - określa, że dany indeks powinien zostać usunięty i przebudowany w jednej transakcji (indeks musi istnieć)
 - szczególnie przydatne przy przebudowywaniu indeksu zgrupowanego – przy usuwaniu lub tworzeniu indeksu zgrupowanego przebudowywane są także indeksy niezgrup.

Tworzenie indeksów

- Indeks...
 - Opcje
 - SORT_IN_TEMPDB
 - decyduje, że pośrednie wyniki sortowania będą przechowywane w bazie tempdb
 - wpływa na poprawę szybkości tworzenia indeksu szczególnie wtedy, gdy bazę tempdb obsługuje np. inny dysk

Tworzenie indeksów

- Tworzenie tabeli a indeksy
 - Przy tworzeniu PRIMARY KEY
 - tworzony jest indeks na kolumny klucza
 - domyślnie tworzony indeks jest zgrupowany, różnowartościowy
 - Przy określeniu UNIQUE
 - również tworzony jest indeks na kolumny klucza
 - domyślnie tworzony indeks jest niezgrupowany, różnowartościowy
 - Przy tworzeniu kolumny tabeli można dodatkowe parametry: clustered, nonclustered, fillfactor

Tworzenie indeksów

- Przy tworzeniu indeksów UNIQUE
 - SQL Server sprawdza istniejące dane pod kątem różnych wartości. Jeśli są powtórzenia, zostanie zwrócony błąd na pierwszym powtarzającym się wierszu
 - Można wykorzystać poniższą procedurę do sprawdzenia, które wiersze ile razy się powtarzają
 - `SELECT kolumna, COUNT (kolumna)`
`FROM tabela`
`GROUP BY kolumna`
`HAVING COUNT(kolumna)>1 ORDER BY kolumna`

Tworzenie indeksów

- Tworzenie indeksów złożonych
 - jest to indeks oparty na co najmniej 2 kolumnach
 - kolumny muszą być z tej samej tabeli (chyba, że tworzymy indeks na widoku)
 - warto najpierw podać kolumnę bardziej unikalną (np. nazwisko, a potem imię)
 - w klauzuli WHERE należy podać pierwszą kolumnę indeksu, żeby indeks został wykorzystany
 - indeksy na (kol1, kol2) i na (kol2, kol1) są różne
 - użycie tych indeksów może zredukować ilość ogólnie utworzonych indeksów

Tworzenie indeksów

- Indeksy na kolumnach wyliczeniowych
 - Indeks można utworzyć, gdy
 - wyrażenie kolumnowe jest deterministyczne (np. funkcja `getdate()` nie jest deterministyczna)
 - przy tworzeniu tabeli, opcja połączenia `ANSI_NULL` jest ustawiona na `ON`
 - to, czy była włączona, można sprawdzić poleceniem

```
if objectproperty( object_id('tabela'), 'IsAnsiNullsOn')=1
print 'ANSI_NULL'
```
 - wyrażenie nie może być oparte na kolumnie o typie `text`, `ntext` lub `image`

Tworzenie indeksów

- Indeksy na kolumnach wyliczeniowych
 - Indeks można utworzyć, gdy
 - przy tworzeniu tabeli i przy operacjach, które modyfikują indeks (insert, update, delete) opcje połączenia
 - ANSI_NULLS
 - ANSI_PADDING
 - ANSI_WARNINGS
 - ARITHABORT
 - CONCAT_NULL_YIELDS_NULL
 - QUOTED_IDENTIFIER
 - powinny być włączone, natomiast opcja
 - NUMERIC_ROUNDABORT
 - powinna być wyłączona

Tworzenie indeksów

- Indeksy na kolumnach wyliczeniowych
 - Jeśli któryś z przedstawionych warunków nie będzie spełniony, procesor kwerend może indeks na kolumnie wyliczeniowej zignorować.

Usuwanie indeksu

- Usuwanie indeksu odbywa się poprzez
 - DROP INDEX w przypadku indeksów utworzonych za pomocą CREATE INDEX
 - ALTER TABLE w przypadku indeksów utworzonych przy tworzeniu tabeli
- Przy usuwaniu tabeli usuwane są wszystkie związane z nią indeksy
- DROP INDEX nie może być użyty wobec tabel systemowych

Przebudowanie indeksu

- Do przebudowania indeksu jest polecenie **DBCC DBREINDEX**
 - Składnia
DBCC DBREINDEX
(['baza.właściciel.tabela' [, index [, fillfactor]]])
[WITH NO_INFOMSGS]
 - Uwagi
 - index – nazwa indeksu, jeśli podamy "", przebudowane będą wszystkie indeksy
 - opcja WITH NO_INFOMSGS – wyłącza pojawianie się komunikatów informacyjnych (poziom 0 do 10)

Przebudowanie indeksu

- Indeks jest przebudowywany automatycznie
 - przy tworzeniu indeksu zgrupowanego
 - przy usuwaniu indeksu zgrupowanego
 - przy użyciu opcji `DROP_EXISTING`

Pobieranie informacji o indeksach

- Można to robić za pomocą dwóch procedur
 - sp_help tabela
 - sp_helpindex tabela

Przykłady

- sqlserver-p07-01.sql
- sqlserver-p07-02A.sql
- sqlserver-p07-02B.sql
- sqlserver-p07-03.sql

Fragmentacja indeksów

- Fragmentacja pojawia się przy modyfikacji tabeli
- Rodzaje fragmentacji
 - wewnętrzna
 - oznacza stopień niewykorzystania stron indeksów
 - często pożądana, ustalana poprzez parametr fillfactor
 - zewnętrzna
 - pojawia się, gdy
 - logiczna kolejność stron nie odpowiada kolejności fizycznej
 - obszary należące do tabeli nie sąsiadują z sobą
 - kłopotliwa, gdy SQLServer np. skanuje wszystkie elementy tabeli lub indeksu

Wykrywanie fragmentacji

- Do wykrywania fragmentacji służy polecenie **DBCC SHOWCONTIG**

- Składnia

- **DBCC SHOWCONTIG**

```
[ ( { table_name | table_id | view_name | view_id }  
  [ , index_name | index_id ] ) ]  
[ WITH {  
    ALL_INDEXES  
  | FAST [ , ALL_INDEXES ]  
  | TABLERESULTS [ , { ALL_INDEXES } ]  
  [ , { FAST | ALL_LEVELS } ]  
  }  
]
```

Wykrywanie fragmentacji

- Polecenie DBCC SHOWCONTIG
 - Opcje polecenia
 - FAST – skanowanie będzie szybsze, ale pobrana będzie mniejsza ilość informacji
 - ALL_LEVELS
 - dualne do FAST
 - wyciąga wszystkie informacje
 - TABLERESULTS – wyniki będą w postaci tabeli wraz z dodatkowymi informacjami
 - ALL_INDEXES – informacje będą o wszystkich indexach niezależnie, czy został wybrany konkretny index

Wykrywanie fragmentacji

- Znaczenie pozycji wyniku polecenie DBCC
 - Page scanned – liczba stron w indeksie
 - Extents scanned – liczba obszarów w indeksie
 - Extent switches – liczba przejść między obszarami przy przeglądaniu kolejnych stron indeksu
 - Avg. pages per extent – liczba stron w obszarze w ramach łańcucha stron
 - Scan density [best count:actual count]
 - best count – optymalna liczba obszarów
 - actual count – bieżąca liczba obszarów

Wykrywanie fragmentacji

- Znaczenie pozycji wyniku polecenie DBCC
 - Logical scan fragmentation – procent stron dla których logiczna kolejność nie przenosi się na kolejność liści w drzewie
 - Extent scan fragmentation – stosunek liczb luk pomiędzy obszarami, a liczbą obszarów
 - obszar jest identyfikowany przez pierwszej jego strony
 - np. kolejne obszary indeksu to 8, 16, 24 i 40, zatem luka jest pomiędzy 24 i 40.
 - ponieważ liczba luk jest max. liczba obszarów-1, współczynnik nie osiąga nigdy 100%

Wykrywanie fragmentacji

- Znaczenie pozycji wyniku polecenie DBCC
 - Do wyznaczenia stopnia fragmentacji najbardziej odpowiednie są parametry:
 - Logical scan fragmentation
 - Extent scan fragmentation
 - I w trochę mniejszym stopniu:
 - Scan density
 - w przypadku, gdy tabela jest oparta na kilku plikach, parametr ten przestaje być wiarygodny

Pozbywanie się fragmentacji

- Sposób 1: przebudowanie
 - Przykładowe sposoby przebudowania
 - wykonać DROP INDEX i CREATE INDEX
 - CREATE INDEX z opcją DROP EXISTING
 - Pomijając inne wady, podstawowa jest taka, że na tabele są nakładane blokady i stają się one częściowo lub w ogóle niedostępne
 - Jeśli indeks ma wysoki stopień fragmentacji, ta metoda jest stosunkowo najszybsza

Pozbywanie się fragmentacji

- Sposób 2: defragmentacja
 - Służy do tego polecenie DBCC INDEXDEFRAG
 - Składnia
 - DBCC INDEXDEFRAG
({ database_name | database_id | 0 }
, { table_name | table_id | 'view_name' | view_id }
, { index_name | index_id }
) [WITH NO_INFOMSGS]
 - Opcje
 - Podając bazę o numerze 0, podajemy bazę bieżącą
 - NO_INFOMSGS – wyłącza generowanie komunikatów o wadze 0 do 10 (informacyjnych)

Przykład

- sqlserver-p07-04.sql

Zbieranie informacji do statystyk

- Przeglądanie wszystkich lub losowo wybranych wartości w kolumnach
- Przeglądanie losowej próbki jest domyślne przy
 - tworzeniu i aktualizacji statystyk
- Przeglądanie wszystkich wierszy jest domyślne
 - przy tworzeniu indexów
 - przy użyciu opcji FULLSCAN podczas tworzeniu lub aktualizacji

Tworzenie statystyk

- Automatyczne
 - Jeśli opcja bazy danych auto create statistics jest ustawiona na ON, automatycznie są tworzone statystyki dla
 - indeksowanych kolumn zawierających dane
 - nieindeksowanych kolumn używanych w złączeniach klauzuli where
 - Wyłączenie tej opcji może źle wpłynąć na wydajność, gdyż wtedy optymalizator kwerend nie będzie mógł z nich korzystać

Tworzenie statystyk

- Ręczne
 - Statystyki tworzymy za pomocą polecenia
 - CREATE STATISTICS statistics_name ON {table| view} (column [,...n])
 - Statystyki można utworzyć dla
 - nieindeksowanych kolumn
 - wszystkich kolumn poza pierwszą indeksu złożonego
 - dla kolumn wyliczanych, ale takich, dla których można by utworzyć indeks (czyli spełniających pewne warunki)
 - kolumn nie opartych na typach text, ntext, image
 - Tworzenie takich statystyk może być przydatne dla optymalizatora kwerend wtedy, gdy nie chcemy tworzyć indeksu

Odświeżanie statystyk

- Jest to ważne, ponieważ przy nieaktualnych danych optymalizator kwerend może działać nieoptymalnie
- Automatyczne
 - ma miejsce, gdy jest włączona opcja auto update statistics
 - aktualizacja jest wykonywana przy optymalizacji zapytań przez optymalizator kwerend

Odświeżanie statystyk

- Ręczne
 - można wywołać poleceniem
 - UPDATE STATISTICS table| view [index | (statistics_name[,...n])]
 - należy je wykonać w następujących sytuacjach
 - kiedy utworzymy indeks przed wstawieniem danych do tabeli
 - kiedy tabela jest obcięta (truncate)
 - kiedy wstawiamy dużo wierszy do tabeli (szczególnie z małą ilością danych) i zamierzamy od razu wykorzystywać ją w zapytaniach

Wyświetlanie statystyk

- Wyświetlamy statystyki za pomocą polecenia `DBCC SHOW_STATISTICS`
 - Składnia
 - `DBCC SHOW_STATISTICS (table, target)`
 - Opcje
 - `target` – czego statystyki chcemy wyświetlić
 - nazwa indeksu
 - nazwa statystyki

Przykład

- sqlserver-p07-05.sql

Indeksowanie widoków

- Indeksowanie widoków odpowiada widokom zmaterializowanym
- Pierwszym indeksem musi być unikalny indeks zgrupowany
- Podobnie jak w przypadku indeksów na kolumnach wyliczeniowych należy ustawić następujące opcje
 - SET ANSI_NULLS ON, SET ANSI_PADDING ON
 - SET ANSI_WARNINGS ON, SET ARITHABORT ON
 - SET CONCAT_NULL_YIELDS_NULL ON
 - SET QUOTED_IDENTIFIER ON
 - SET NUMERIC_ROUNDABORT OFF

Indeksowanie widoków

- Definicja widoku, który ma być indeksowany nie może zawierać
 - słów TOP, DISTINCT, UNION
 - kolumn typu text, ntext, image
 - funkcji MIN, MAX, COUNT(*), COUNT(wyr), AVG ...
 - innego widoku (tylko tabele bazowe)
 - klauzul COMPUTE, COMPUTE BY
 - klauzuli ORDER BY
 - ...

Indeksowanie widoków

- W przypadku, gdy chcemy użyć GROUP BY, na liście select należy użyć funkcji COUNT_BIG(*)
- GROUP BY nie może mieć klauzuli HAVING
- Fakt, czy widok jest indeksowalny można sprawdzić za pomocą właściwości IsIndexable funkcji OBJECTPROPERTY
- Widok musi być utworzony z opcją SCHEMABINDING – zapobiega to usunięciu tabel bazowych dla widoku

Indeksowanie widoków

- Wykorzystanie indeksów na widokach
 - kwerenda nie musi się bezpośrednio odwoływać do widok, żeby skorzystać z indeksu utworzonego dla tego widoku

Przykład

- sqlserver-p07-06.sql