Rozmiary tabel i indeksów

Poziom: podstawy **Dotyczy: SQL Server 2016–2022** **Czas: ~15 min**

Sprawdzenie, które tabele i indeksy faktycznie zajmują miejsce w bazie, zanim zaczniemy planować reindeksację albo szukać kandydata do archiwizacji.

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Fiszka

TYLKO ODCZYT

Sprawdzenie rozmiaru pojedynczej tabeli: sp_spaceused

```
EXEC sp_spaceused N'NazwaTabeli';
```

Sprawdzamy rozmiar konkretnej tabeli wraz z jej indeksami — wbudowana procedura systemowa, najszybszy sposób na zerknięcie w jeden obiekt bez pisania własnego zapytania.

Przykład:

```
name          rows      reserved    data      index_size   unused
------------- --------- ----------- --------- ------------ --------
Zamowienia    1048576   2145280 KB  1802112 KB 340992 KB   2176 KB


(1 rows affected)
```

exec = zastąp bieżący proces powłoki wskazanym programem, bez tworzenia nowego procesu

TYLKO ODCZYT

Sortowanie tabel po rozmiarze: sys.dm_db_partition_stats

```
SELECT
    t.name AS NazwaTabeli,
    SUM(ps.reserved_page_count) * 8 AS ZarezerwowaneKB,
    SUM(ps.used_page_count) * 8 AS ZajeteKB
FROM sys.tables t
JOIN sys.dm_db_partition_stats ps ON t.object_id = ps.object_id
GROUP BY t.name
ORDER BY SUM(ps.used_page_count) DESC;
```

Sprawdzamy, które tabele w bieżącej bazie zajmują najwięcej miejsca — zapytanie działa w kontekście bazy, na której jest wykonane, więc wymaga wcześniejszego `USE NazwaBazy`. `sys.dm_db_partition_stats` liczy strony osobno dla każdej partycji każdego indeksu, dlatego sumujemy je po `object_id` tabeli, żeby dostać jeden wiersz na tabelę zamiast jednego na indeks.

Przykład:

```
NazwaTabeli        ZarezerwowaneKB   ZajeteKB
------------------ ----------------- ----------
Zamowienia         2145280           1802112
PozycjeZamowien    987136            845056
Klienci            65536             52224

(3 rows affected)
```

□□ **join** = połącz dwa pliki tekstowe po wspólnej kolumnie, jak złączenie tabel w bazie danych

□□ **TYLKO ODCZYT**

Zajętość poszczególnych tabel wraz z indeksami: sys.allocation_units

```
SELECT TOP (30)
    SCHEMA_NAME(tab.schema_id) + '.' + tab.name AS Tabela,
    CAST(SUM(spc.used_pages * 8) / 1024.00 AS numeric(36, 2)) AS ZajeteMB,
    CAST(SUM(spc.total_pages * 8) / 1024.00 AS numeric(36, 2)) AS AlokowaneMB
FROM sys.tables tab
JOIN sys.indexes ind ON tab.object_id = ind.object_id
JOIN sys.partitions part ON ind.object_id = part.object_id AND ind.index_id =
```

```
part.index_id
JOIN sys.allocation_units spc ON part.partition_id = spc.container_id
GROUP BY SCHEMA_NAME(tab.schema_id) + '.' + tab.name
ORDER BY SUM(spc.used_pages) DESC;
```

Sprawdzamy trzydzieści największych tabel w bieżącej bazie, tym razem licząc bezpośrednio na poziomie jednostek alokacji, z rozróżnieniem miejsca faktycznie zajętego (`used_pages`) od zarezerwowanego dla obiektu (`total_pages`) — różnica między tymi dwiema kolumnami to zapas, który zwolniłby `DBCC SHRINKFILE`, gdybyśmy go wykonali, ale tego typowo nie robimy z powodów opisanych na stronie Log transakcyjny: rozmiar, shrink, zapchanie.

Przykład:

```
Tabela               ZajeteMB    AlokowaneMB
-------------------- ----------- ------------
dbo.Zamowienia        1760.25     1784.50
dbo.PozycjeZamowien    825.50     841.75
dbo.Klienci             51.00      52.25

(3 rows affected)
```

DBCC = Database Console Commands: rodzina poleceń MSSQL do sprawdzania i naprawy spójności bazy, na przykład DBCC CHECKDB · **join** = połącz dwa pliki tekstowe po wspólnej kolumnie, jak złączenie tabel w bazie danych

ZMIENIA SYSTEM

Weryfikacja TotalIndexSize przed reindeksacją

```
SELECT
    SUM(p.used_page_count) * 8 / 1024.0 / 1024.0 AS TotalIndexSizeGB
FROM sys.dm_db_partition_stats p
JOIN sys.indexes i ON p.object_id = i.object_id AND p.index_id = i.index_id
WHERE i.type > 0;
```

Sprawdzamy łączny rozmiar wszystkich indeksów w bieżącej bazie, z pominięciem samych stert (`index_id = 0`, `type = 0`) — ta liczba jest orientacyjnym punktem odniesienia przy planowaniu reindeksacji, bo `ALTER INDEX ... REBUILD` w trakcie działania potrzebuje dodatkowego miejsca na dysku, w skrajnym przypadku zbliżonego do rozmiaru przebudowywanego indeksu, dopóki stara wersja nie zostanie zwolniona po zakończeniu

operacji.

Przykład:

```
TotalIndexSizeGB
------------------
42.780000


(1 rows affected)
```

Uważaj: przy rebuild bez `ONLINE` silnik trzyma jednocześnie starą i nową wersję przebudowywanego indeksu do samego końca operacji — na dysku, na którym zostało niewiele wolnego miejsca, duży rebuild może się nie zmieścić i przerwać z błędem `Msg 1101, Level 17, State 12, Could not allocate a new page for database`.

join = połącz dwa pliki tekstowe po wspólnej kolumnie, jak złączenie tabel w bazie danych

Z życia wzięte

Z ŻYCIA WZIĘTE

W ostatni dzień miesiąca, kiedy księgowość pracuje do nocy, monitoring zgłasza, że dysk z danymi CDN_Zielona ma zaledwie kilkanaście procent wolnego miejsca. Zamiast zgadywać, sprawdzamy zapytaniem do sys.dm_db_partition_stats, które tabele zajmują najwięcej miejsca. Okazuje się, że tabela Zamowienia urosła w tym miesiącu wyjątkowo mocno, bo dział sprzedaży importował archiwalne zamówienia z poprzednich lat. Ten wynik pozwala nam dokupić miejsce na dysku, zamiast w nocy walczyć z zapchaną bazą.

Po co to istnieje

Rozmiar tabeli i jej indeksów to jedna z pierwszych rzeczy, o które pytamy, zanim podejmiemy dowolną decyzję operacyjną — czy starczy miejsca na reindeksację, czy backup zmieści się w oknie serwisowym, czy tabela jest kandydatem do partycjonowania albo archiwizacji starych wierszy. Bez tych liczb każda taka decyzja jest zgadywaniem.

> **Pojęcie: strona (page) i extent.** Silnik SQL Server przechowuje dane w stronach po 8 KB — to najmniejsza jednostka odczytu i zapisu z dysku. Osiem kolejnych stron tworzy extent (64 KB); większość zapytań o rozmiar liczy właśnie strony i mnoży je przez 8 KB, żeby dostać wynik w kilobajtach, tak jak w zapytaniach powyżej.

Częste błędy

Mylenie rozmiaru tabeli z rozmiarem samych danych bez indeksów. `sp_spaceused` i zapytania powyżej liczą tabelę razem z jej indeksami — tabela z dużą liczbą indeksów wtórnych bywa mniejsza od nich łącznie niż same dane, co jest częstym zaskoczeniem przy pierwszym spojrzeniu na wynik.

Branie `AlokowaneMB`/`ZarezerwowaneKB` za rozmiar realnie potrzebny po reindeksacji. Ta kolumna zawiera zapas i fragmentację nagromadzoną przez lata — po świeżym `REBUILD` rozmiar zwykle spada, czasem znacząco, co jest normalne i nie oznacza utraty danych.

Powiązane: Rozpoznanie instancji i baz · Indeksy: status, fragmentacja, przebudowa · Fragmentacja i przebudowa indeksów

Źródła: dokumentacja Microsoft Learn, sekcja „sp_spaceused" i „sys.dm_db_partition_stats"

Revision #43
Created 2025-03-08 10:51:58 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:38 UTC by Tomasz Konieczny