Backup baz danych

Poziom: średni **Dotyczy: SQL Server 2016-2022** **Czas: ~20 min**

Wykonywanie backupu pełnego, różnicowego i logów, sprawdzanie historii wykonanych backupów oraz porządkowanie narastającej historii w bazie msdb.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
LSN	Log Sequence Number: rosnący numer pozycji w dzienniku transakcyjnym, po którym silnik rozpoznaje kolejność zmian
RPO	Recovery Point Objective: ile danych wolno stracić przy awarii
RTO	Recovery Time Objective: jak długo wolno nie działać przy awarii
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
SSMS	SQL Server Management Studio: graficzna konsola MSSQL

Fiszka

ZMIENIA SYSTEM

Backup pełny bazy: BACKUP DATABASE

```
BACKUP DATABASE NazwaBazy
TO DISK = N'SciezkaDoPliku\NazwaBazy_full.bak'
WITH COMPRESSION, CHECKSUM, INIT;
```

Wykonujemy pełny backup bazy do wskazanego pliku. `COMPRESSION` zmniejsza rozmiar pliku wynikowego i zwykle skraca czas backupu, bo mniej danych trzeba zapisać na dysk. `CHECKSUM` włącza weryfikację sum kontrolnych stron w trakcie backupu, co pozwala wykryć uszkodzone strony, zanim trafią do pliku backupu. `INIT` nadpisuje istniejącą zawartość pliku zamiast dopisywać kolejny backup na jego koniec.

Przykład:

```
Processed 1048576 pages for database 'NazwaBazy', file 'NazwaBazy' on file 1.
Processed 3 pages for database 'NazwaBazy', file 'NazwaBazy_log' on file 1.
BACKUP DATABASE successfully processed 1048579 pages in 42.318 seconds (194.213 MB/sec).
```

◫◫ ZMIENIA SYSTEM

Backup różnicowy: DIFFERENTIAL

```
BACKUP DATABASE NazwaBazy
TO DISK = N'SciezkaDoPliku\NazwaBazy_diff.bak'
WITH DIFFERENTIAL, COMPRESSION, CHECKSUM;
```

Wykonujemy backup wyłącznie tych stron, które zmieniły się od ostatniego backupu pełnego — mniejszy i szybszy niż pełny, ale bezużyteczny bez backupu pełnego, na którym się opiera. Odtworzenie wymaga zawsze najpierw backupu pełnego, a dopiero potem najnowszego różnicowego.

◫◫ TYLKO ODCZYT

Backup logu transakcyjnego: BACKUP LOG

```
BACKUP LOG NazwaBazy
TO DISK = N'SciezkaDoPliku\NazwaBazy_log.trn'
WITH COMPRESSION, CHECKSUM;
```

Wykonujemy backup logu transakcyjnego, dostępny wyłącznie dla baz w trybie recovery `FULL` albo `BULK_LOGGED` — opisanych na stronie Architektura MSSQL: instancja, baza, pliki MDF/NDF/LDF. Ten backup, w połączeniu z ostatnim pełnym i różnicowym, umożliwia odtworzenie bazy do dowolnego momentu w czasie, opisane na stronie Odtwarzanie do punktu w czasie (PITR), i jest jedynym mechanizmem, który regularnie obcina rosnący log — bez niego log rośnie bez końca, co opisujemy na stronie Log transakcyjny: rozmiar, shrink, zapchanie.

◫◫ **MSSQL** = Microsoft SQL Server: silnik bazy danych Microsoftu, na nim w Zielonej stoi księgowość Optima · **MDF** = Master Data File: główny plik danych bazy MSSQL · **PITR** = Point-In-Time Recovery: odtworzenie bazy do dowolnej chwili, np. 14:31

◫◫ ZMIENIA SYSTEM

Backup wyłącznie nowej bazy bez wcześniejszej historii: COPY_ONLY

```
BACKUP DATABASE NazwaBazy
TO DISK = N'SciezkaDoPliku\NazwaBazy_copy.bak'
WITH COPY_ONLY, COMPRESSION;
```

Wykonujemy backup, który nie wchodzi w normalny łańcuch backupów pełny/różnicowy/log — kolejny backup różnicowy nadal będzie liczony względem poprzedniego backupu pełnego, tak jakby tego `COPY_ONLY` w ogóle nie było. Sięgamy po tę opcję, kiedy potrzebujemy jednorazowej kopii bazy (np. do skopiowania na środowisko testowe), nie chcąc zaburzyć istniejącego harmonogramu backupów różnicowych.

TYLKO ODCZYT

Sprawdzenie ostatnich backupów danej bazy: msdb.dbo.backupset

```
SELECT database_name, backup_finish_date AS DataZakonczeniaBackupu
FROM msdb.dbo.backupset
WHERE type = 'L'
    AND database_name = N'NazwaBazy'
ORDER BY backup_finish_date DESC;
```

Sprawdzamy historię backupów logu dla wskazanej bazy — typ backupu wybieramy filtrem `type`, gdzie każda litera oznacza inny rodzaj wykonanej operacji.

Wartość `type`	Rodzaj backupu
`D`	backup pełny (Full)
`I`	backup różnicowy (Differential)
`L`	backup logu (Log)
`F`	backup plikowy (File)
`G`	backup różnicowy plikowy (Differential File)
`P`	backup częściowy (Partial)
`Q`	backup różnicowy częściowy (Differential Partial)

Przykład:

```
database_name   DataZakonczeniaBackupu
--------------- ----------------------
```

```
CDN_Zielona          2026-09-25 03:15:22.000
CDN_Zielona          2026-09-25 02:15:19.000
CDN_Zielona          2026-09-25 01:15:14.000


(3 rows affected)
```

□□ TYLKO ODCZYT

Purge informacji o backupach z bazy: sp_delete_database_backuphistory

```
USE msdb;
GO
EXEC sp_delete_database_backuphistory N'NazwaBazy';
GO
```

Usuwamy z `msdb` historię backupów wskazanej bazy — same pliki `.bak`/`.trn` na dysku pozostają nietknięte, kasowana jest wyłącznie metadana o nich w tabelach `backupset`, `backupfile` i powiązanych. Sięgamy po to polecenie, kiedy historia w `msdb` narosła do rozmiaru wpływającego na jej własną wydajność, opisanego na stronie Rozpoznanie instancji i baz — samo `msdb` też trzeba czasem posprzątać.

Uważaj: po tym poleceniu tracimy możliwość odtworzenia z zestawu backupów starszych niż wykonana czystka bez ręcznego wskazania plików `.bak` przez `RESTORE ... FROM DISK` — sam silnik nie będzie już „pamiętał" tej historii przy budowaniu planu odtworzenia w SSMS.

□□ **SSMS** = SQL Server Management Studio: graficzna konsola MSSQL · **exec** = zastąp bieżący proces powłoki wskazanym programem, bez tworzenia nowego procesu

□□ ZMIENIA SYSTEM

Weryfikacja poprawności pliku backupu: RESTORE VERIFYONLY

```
RESTORE VERIFYONLY FROM DISK = N'SciezkaDoPliku\NazwaBazy_full.bak';
```

Sprawdzamy, czy plik backupu jest fizycznie kompletny i czytelny, bez faktycznego odtwarzania z niego bazy — nie weryfikuje to jednak logicznej spójności danych wewnątrz, tylko integralność samego pliku.

Przykład:

```
The backup set on file 1 is valid.
```

ZMIENIA SYSTEM

Sprawdzenie zawartości pliku backupu: RESTORE HEADERONLY i RESTORE FILELISTONLY

```
RESTORE HEADERONLY FROM DISK = N'SciezkaDoPliku\NazwaBazy_full.bak';
```

Sprawdzamy metadane zestawu backupów w pliku — nazwę bazy, typ backupu, datę wykonania, numer LSN — przydatne, kiedy dostajemy plik `.bak` bez opisu i chcemy wiedzieć, co dokładnie w sobie zawiera, zanim rozpoczniemy odtwarzanie.

```
RESTORE FILELISTONLY FROM DISK = N'SciezkaDoPliku\NazwaBazy_full.bak';
```

Sprawdzamy listę plików logicznych i fizycznych zapisanych w danym backupie — potrzebne przed odtworzeniem na inny serwer, gdzie ścieżki dysków mogą być inne niż na oryginalnym, co wymaga użycia opcji `MOVE` opisanej na stronie Odtwarzanie baz danych.

Przykład:

```
LogicalName   PhysicalName                  Type   FileGroupName   Size
------------- ----------------------------- ------ --------------- -----------
NazwaBazy     D:\MSSQL\DATA\NazwaBazy.mdf   D      PRIMARY         8589934592
NazwaBazy_log E:\MSSQL\LOG\NazwaBazy_log.ldf L                      2147483648


(2 rows affected)
```

LSN = Log Sequence Number: rosnący numer pozycji w dzienniku transakcyjnym, po którym silnik rozpoznaje kolejność zmian

Z życia wzięte

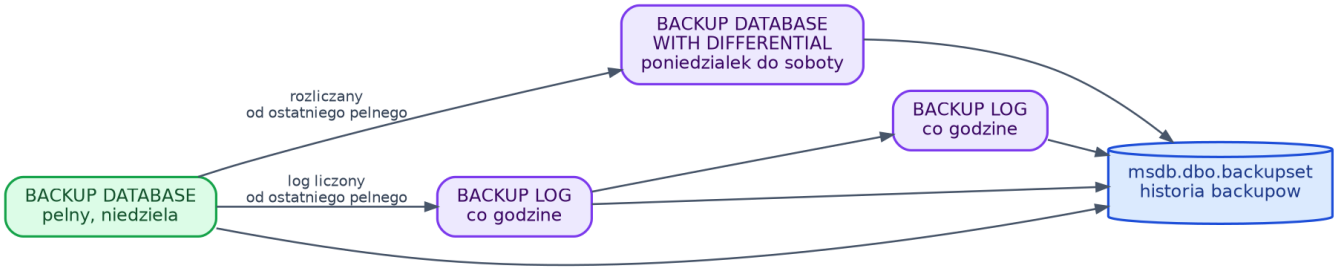
Z ŻYCIA WZIĘTE

Raz w roku pan Wojtek pyta, czy backupy CDN_Zielona faktycznie dają się odtworzyć, a nie tylko wykonują się bez błędu w harmonogramie. Sprawdzamy RESTORE VERIFYONLY na najnowszym pliku pełnym i różnicowym, a potem RESTORE HEADERONLY, żeby pokazać mu

> dokładną datę i rozmiar backupu. Przy okazji zauważamy, że backup różnicowy z piątku jest większy niż zwykle, bo w tym tygodniu księgowość importowała duży plik z rozliczeniami rocznymi. Pan Wojtek dostaje wydruk z datami i rozmiarami i temat na ten rok jest zamknięty.

Po co to istnieje

Backup jest jedynym środkiem, jakim dysponujemy, kiedy coś pójdzie nieodwracalnie źle — pomyłkowy `DELETE` bez `WHERE`, awaria dysku, uszkodzenie logiczne po błędnym wdrożeniu. Trzy rodzaje backupu (pełny, różnicowy, log) i sposób, w jaki się ze sobą łączą, opisujemy szerzej na stronie Backup pełny, różnicowy, przyrostowy, logi razem z pojęciami RPO i RTO, które decydują o tym, jak często backup w ogóle wykonujemy.



Backup pelny, po nim roznicowy i logi budujace lancuch odtwarzania (kliknięcie otwiera pełny rozmiar)

Częste błędy

Backup różnicowy bez wcześniejszego backupu pełnego. Backup różnicowy jest bezużyteczny bez punktu odniesienia — próba odtworzenia z samego różnicowego kończy się błędem, bo silnik nie ma bazowego stanu, do którego mógłby dołożyć zmiany.

Backup logu na bazie w trybie `SIMPLE`. `BACKUP LOG` na takiej bazie kończy się błędem `Msg 4208, Level 16, State 1, The statement BACKUP LOG is not allowed while the recovery model is SIMPLE` — log w tym trybie jest obcinany automatycznie i nie ma czego backupować.

Trzymanie backupu na tym samym dysku fizycznym co pliki bazy. Awaria dysku danych niszczy wtedy jednocześnie bazę i jej backup — reguła, którą łatwo przeoczyć na małych środowiskach, gdzie jest tylko jeden wolumen danych.

Powiązane: Odtwarzanie baz danych · Backup pełny, różnicowy, przyrostowy, logi · RPO i RTO · Log transakcyjny: rozmiar, shrink, zapchanie

Źródła: dokumentacja Microsoft Learn, sekcja „BACKUP (Transact-SQL)" i „msdb.dbo.backupset"

Revision #42
Created 2025-03-08 10:26:26 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:47 UTC by Tomasz Konieczny