Backup i odtwarzanie

Kopie zapasowe, odtwarzanie i naprawa uprawnień po restore.

- Backup baz danych
- Odtwarzanie baz danych
- Naprawa uprawnień użytkownika SQL po odtworzeniu baz

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

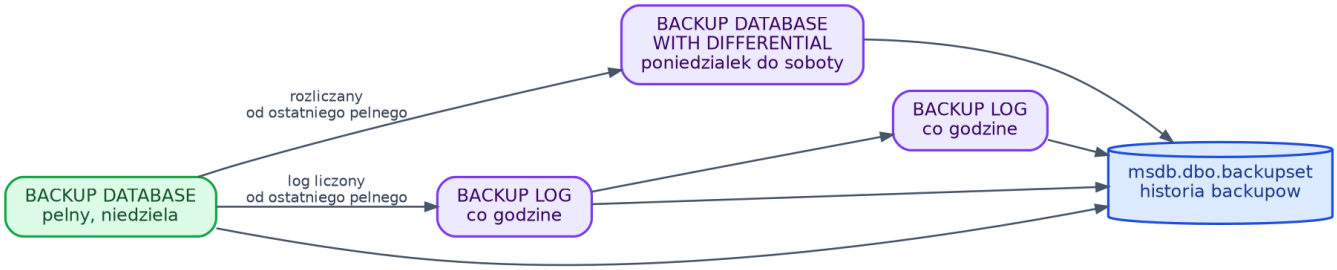
Z ŻYCIA WZIĘTE

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"

Odtwarzanie baz danych

Poziom: średni **Dotyczy: SQL Server 2016-2022** **Czas: ~25 min**

Odtwarzanie bazy z backupu pełnego, dokładanie kolejnych różnicowych i logów, śledzenie postępu długiego odtwarzania i wyjście ze stanu RESTORING.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
LSN	Log Sequence Number: rosnący numer pozycji w dzienniku transakcyjnym, po którym silnik rozpoznaje kolejność zmian
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Fiszka

ZMIENIA SYSTEM

Odtworzenie backupu pełnego z pozostawieniem bazy w stanie oczekiwania: NORECOVERY

```
RESTORE DATABASE NazwaBazy
FROM DISK = N'SciezkaDoPliku\NazwaBazy_full.bak'
WITH NORECOVERY, REPLACE;
```

Odtwarzamy backup pełny, ale zostawiamy bazę w stanie `RESTORING...`, gotową na dołożenie kolejnych backupów — różnicowego i logów. Opcja `REPLACE` pozwala nadpisać istniejącą bazę o tej samej nazwie, nawet jeśli jej struktura plików różni się od tej zapisanej w backupie; bez niej odtwarzanie na miejsce istniejącej bazy kończy się błędem bezpieczeństwa, chroniącym przed przypadkowym nadpisaniem.

Przykład:

```
Processed 1048576 pages for database 'NazwaBazy', file 'NazwaBazy' on file 1.
Processed 3 pages for database 'NazwaBazy', file 'NazwaBazy_log' on file 1.
```

```
RESTORE DATABASE successfully processed 1048579 pages in 38.140 seconds (220.442
MB/sec).
```

◫◫ ZMIENIA SYSTEM

Dołożenie backupu różnicowego

```
RESTORE DATABASE NazwaBazy
FROM DISK = N'SciezkaDoPliku\NazwaBazy_diff.bak'
WITH NORECOVERY;
```

Dokładamy najnowszy backup różnicowy na odtworzony wcześniej backup pełny — nadal z `NORECOVERY`, bo po nim mogą przyjść jeszcze backupy logów.

◫◫ ZMIENIA SYSTEM

Dołożenie backupu logu

```
RESTORE LOG NazwaBazy
FROM DISK = N'SciezkaDoPliku\NazwaBazy_log.trn'
WITH NORECOVERY;
```

Dokładamy kolejny backup logu — powtarzamy to polecenie dla każdego pliku logu w kolejności chronologicznej, aż do ostatniego dostępnego albo do momentu, w którym chcemy zatrzymać odtwarzanie.

◫◫ ZMIENIA SYSTEM

Odtworzenie do konkretnego momentu w czasie: STOPAT

```
RESTORE LOG NazwaBazy
FROM DISK = N'SciezkaDoPliku\NazwaBazy_log.trn'
WITH NORECOVERY, STOPAT = '2026-09-25T14:32:00';
```

Odtwarzamy log tylko do wskazanego momentu, zamiast do samego jego końca — kluczowa opcja przy odtwarzaniu do punktu tuż przed błędną operacją, opisanym szerzej na stronie Odtwarzanie do punktu w czasie (PITR). `STOPAT` działa wyłącznie w obrębie zakresu czasowego pojedynczego pliku logu — jeśli wskazany moment leży poza tym zakresem,

restore tego pliku kończy się bez efektu i trzeba sięgnąć po plik logu obejmujący właściwy przedział.

PITR = Point-In-Time Recovery: odtworzenie bazy do dowolnej chwili, np. 14:31

ZMIENIA SYSTEM

Zmiana lokalizacji plików przy odtwarzaniu na inny serwer: MOVE

```
RESTORE DATABASE NazwaBazy
FROM DISK = N'SciezkaDoPliku\NazwaBazy_full.bak'
WITH MOVE 'NazwaBazy' TO 'D:\MSSQL\DATA\NazwaBazy.mdf',
     MOVE 'NazwaBazy_log' TO 'E:\MSSQL\LOG\NazwaBazy_log.ldf',
     RECOVERY, REPLACE;
```

Odtwarzamy bazę wskazując nowe ścieżki fizyczne dla każdego pliku logicznego — potrzebne, kiedy odtwarzamy na serwer o innym układzie dysków niż oryginalny. Nazwy logiczne po lewej stronie `MOVE` muszą dokładnie odpowiadać nazwom zwróconym przez `RESTORE FILELISTONLY`, opisane na stronie Backup baz danych.

MSSQL = Microsoft SQL Server: silnik bazy danych Microsoftu, na nim w Zielonej stoi księgowość Optima

ZMIENIA SYSTEM

Wyjście ze stanu RESTORING: WITH RECOVERY

```
RESTORE DATABASE NazwaBazyDanych
WITH RECOVERY;
```

Kończymy sekwencję odtwarzania i przełączamy bazę ze stanu `RESTORING...` do normalnego stanu `ONLINE` — po tym poleceniu nie da się już dołożyć kolejnego backupu logu do tej samej sekwencji odtwarzania; jeśli trzeba dodać jeszcze jeden log, cały proces odtwarzania trzeba rozpocząć od nowa od backupu pełnego.

Przykład:

```
RESTORE DATABASE successfully processed 0 pages in 0.012 seconds (0.000 MB/sec).
```

ZMIENIA SYSTEM

Sprawdzenie postępu trwającego odtwarzania: sys.dm_exec_requests

```
SELECT
    session_id,
    command,
    percent_complete AS ProcentUkonczenia,
    DATEADD(MILLISECOND, estimated_completion_time, GETDATE()) AS SzacowanyKoniec
FROM sys.dm_exec_requests
WHERE command = 'RESTORE DATABASE'
    AND start_time IS NOT NULL;
```

Sprawdzamy, ile procent dużego, długo trwającego odtwarzania już się wykonało i kiedy silnik przewiduje jego zakończenie — zapytanie wykonujemy z poziomu bazy `master`, bo baza odtwarzana jest w tym momencie niedostępna do połączenia.

Przykład:

```
session_id   command           ProcentUkonczenia   SzacowanyKoniec
------------ ----------------- ------------------- -----------------------
58           RESTORE DATABASE  67.42               2026-09-25 15:04:12.000


(1 rows affected)
```

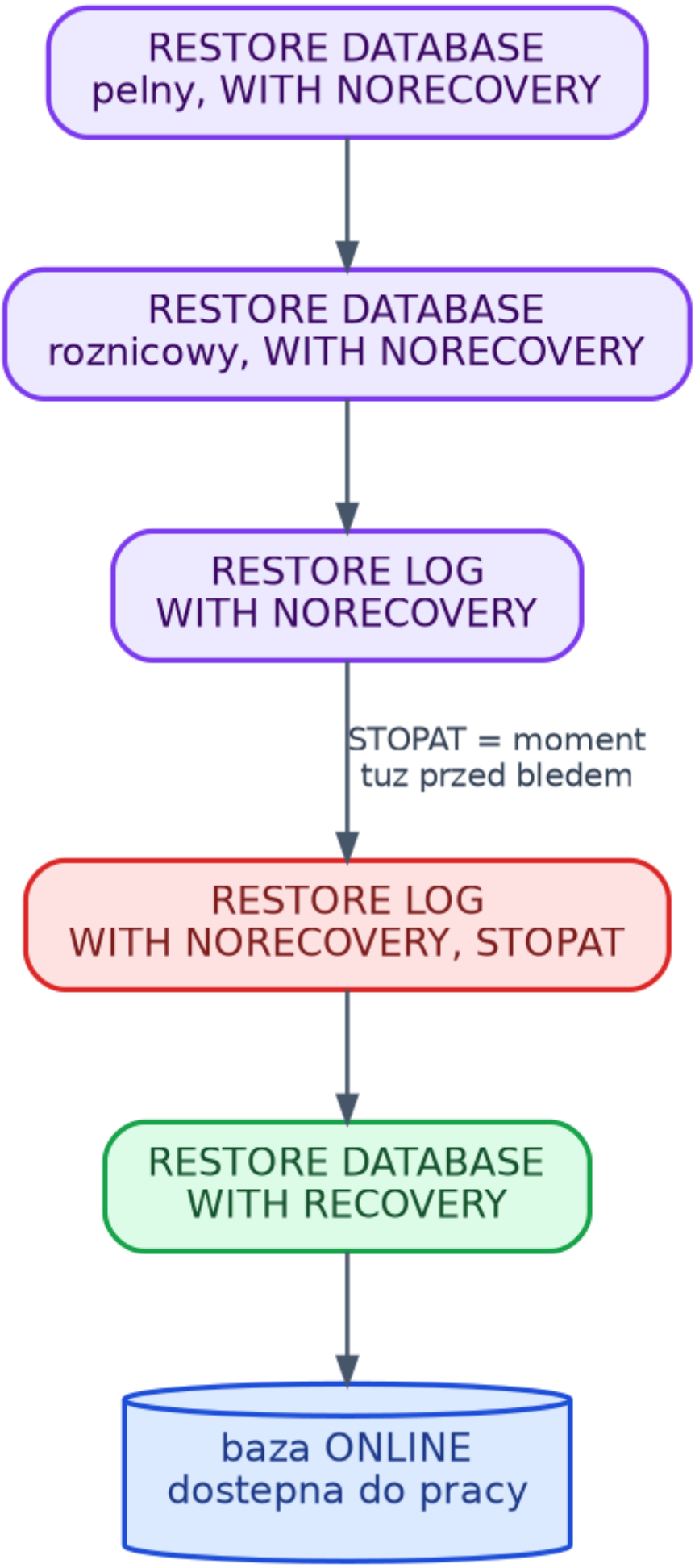
Z życia wzięte

Z ŻYCIA WZIĘTE

We wtorek o 11:20 jeden z programistów Oli uruchamia skrypt czyszczący dane testowe, ale przez pomyłkę odpala go na CDN_Zielona zamiast na kopii testowej, kasując DELETE bez WHERE połowę tabeli Zamowienia. Zespół natychmiast przełącza bazę w tryb SINGLE_USER i zaczyna odtwarzanie: najpierw backup pełny z niedzieli, potem różnicowy z poniedziałku, a na końcu logi aż do momentu tuż przed feralnym DELETE. Odtworzenie z STOPAT ustawionym na 11:19 przywraca dane sprzed pomyłki, tracąc tylko te kilka minut normalnej pracy po tym momencie. Ola przypomina zespołowi, żeby każdy taki skrypt zaczynał się od jawnego sprawdzenia nazwy bazy.

Po co to istnieje

> **Pojęcie: stan RESTORING.** Baza, dla której wykonano `RESTORE ... WITH NORECOVERY`, znajduje się w przejściowym stanie oczekiwania na kolejne pliki backupu. W tym stanie baza jest widoczna na liście `sys.databases`, ale całkowicie niedostępna do zapytań — nawet do samego `SELECT` na jej strukturze. Jest to stan celowy, umożliwiający dołożenie backupu różnicowego i kolejnych logów bez ponownego odtwarzania backupu pełnego od zera.


RESTORE DATABASE
pelny, WITH NORECOVERY
RESTORE DATABASE
roznicowy, WITH NORECOVERY
RESTORE LOG
WITH NORECOVERY
STOPAT = moment
tuz przed bledem
RESTORE LOG
WITH NORECOVERY, STOPAT
RESTORE DATABASE
WITH RECOVERY
baza ONLINE
dostepna do pracy

Kolejne odtworzenie backupu pelnego, roznicowego i logow z zatrzymaniem STOPAT przed bledna operacja (kliknięcie otwiera pełny rozmiar)

Częste błędy

Zapomnienie o `WITH RECOVERY` na końcu sekwencji. Baza zostaje trwale w stanie `RESTORING...`, niedostępna dla aplikacji, mimo że wszystkie potrzebne backupy zostały już odtworzone — pierwszy objaw to zgłoszenie „baza nie działa" mimo braku błędów w logach silnika.

Odtworzenie backupu różnicowego z niewłaściwego zestawu pełnego. Backup różnicowy jest powiązany z konkretnym backupem pełnym przez znacznik LSN — próba dołożenia różnicowego wykonanego po innym backupie pełnym niż ten właśnie odtworzony kończy się błędem `Msg 3136, Level 16, State 1, This differential backup cannot be restored because the database has not been restored to the correct earlier state`.

Mylenie `STOPAT` z filtrowaniem po numerze LSN. `STOPAT` odtwarza log aż do wskazanego czasu zegarowego zapisanego w metadanych transakcji — przy odtwarzaniu bardzo precyzyjnym, tuż przed konkretną znaną transakcją, dokładniejsze jest użycie `STOPBEFOREMARK` z nazwaną transakcją, jeśli backup logu ją obejmuje.

Powiązane: Backup baz danych · Odtwarzanie do punktu w czasie (PITR) · Naprawa uprawnień użytkownika SQL po odtworzeniu baz · Tryby pracy bazy: read-only, single user, offline

Źródła: dokumentacja Microsoft Learn, sekcja „RESTORE (Transact-SQL)" i „sys.dm_exec_requests"

Naprawa uprawnień użytkownika SQL po odtworzeniu baz

Typ: procedura **Dotyczy: SQL Server 2016–2022, uwierzytelnianie mieszane**

Czas: ~15 min

Efekt: użytkownik bazy danych ponownie połączony z prawidłowym loginem na instancji docelowej, bez potrzeby jego usuwania i tworzenia od nowa.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
SID	w Oracle: System Identifier, nazwa instancji; w Windows: Security Identifier, numer konta
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Zanim zaczniemy

Z ŻYCIA WZIĘTE

Po odtworzeniu świeżej kopii CDN_Zielona na serwerze testowym programiści Oli zgłaszają, że aplikacja testowa nie może się zalogować, mimo że hasło na pewno jest poprawne. Sprawdzamy sp_change_users_login z opcją Report i widzimy jednego użytkownika osieroconego, bo login o tej samej nazwie na serwerze testowym powstał niezależnie i ma inny SID niż na produkcji. Po naprawieniu powiązania poleceniem ALTER USER aplikacja loguje się bez żadnej zmiany hasła. Ola zapamiętuje, żeby po każdym odtworzeniu bazy na inny serwer sprawdzać ten sam problem od razu.

Wymaganie	Jak sprawdzić
Baza już odtworzona na docelowej instancji	`SELECT name, state_desc FROM sys.databases WHERE name = N'NazwaBazy';`

Wymaganie	Jak sprawdzić
Login o docelowej nazwie istnieje na instancji, do której odtwarzano	`SELECT name FROM sys.server_principals WHERE type IN ('S','U') AND name = N'NazwaLoginu';`
Dostęp `sysadmin` albo rola `securityadmin` na instancji docelowej	wymagane do wykonania `ALTER USER` / `sp_change_users_login`

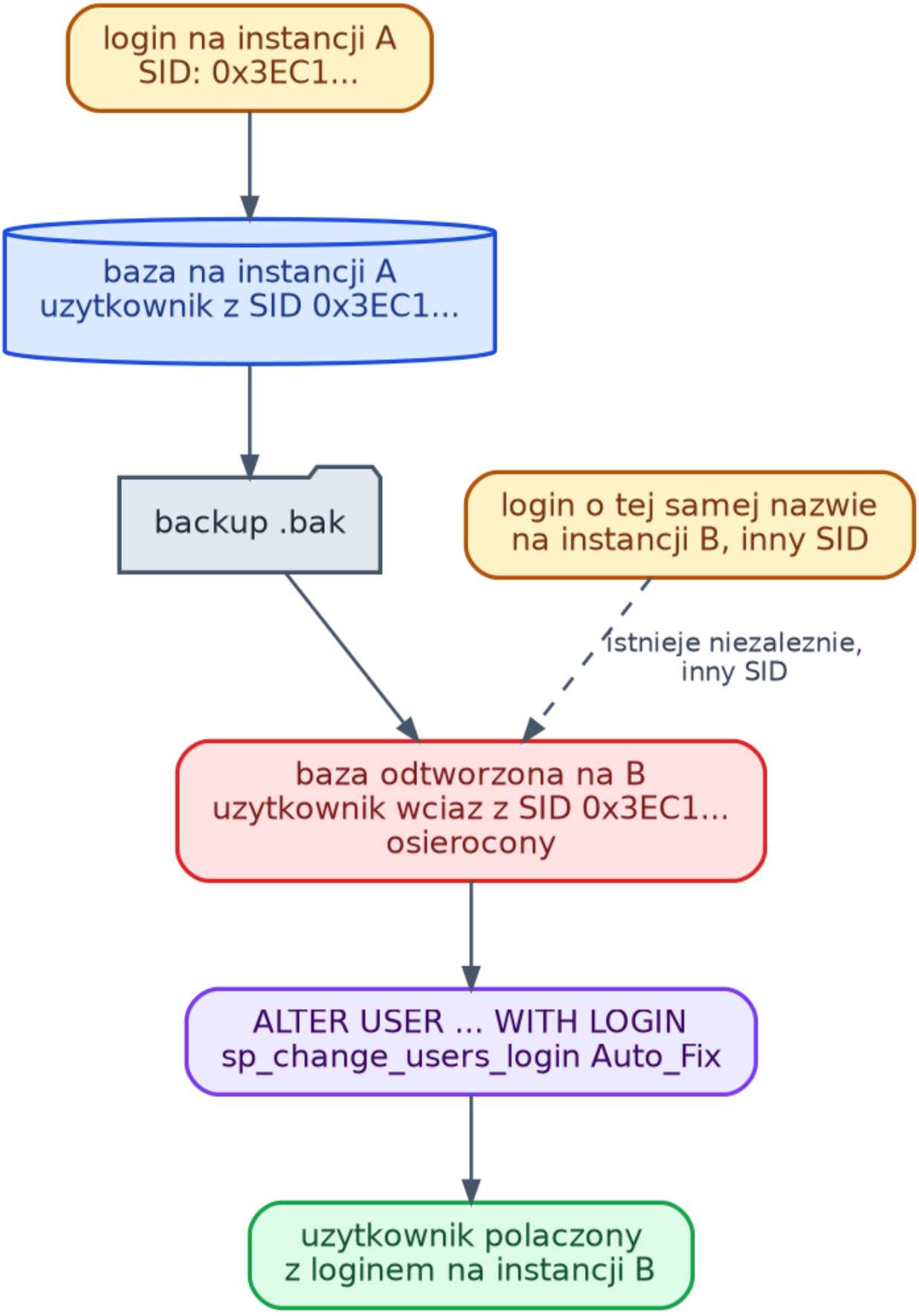
Uważaj: procedura dotyczy wyłącznie loginów SQL Server (uwierzytelnianie mieszane). Konta domenowe Windows nie osieracają się w ten sposób — identyfikują użytkownika po SID domenowym, niezależnym od instancji, więc problem opisany niżej dotyczy typowo migracji i odtworzeń między środowiskami z odrębnymi loginami SQL.

Przebieg

1. Wykrycie osieroconych użytkowników

```
USE NazwaBazy;
GO
EXEC sp_change_users_login @Action = 'Report';
```

> **Pojęcie: osierocony użytkownik (orphaned user).** Użytkownik bazy danych, którego wewnętrzny identyfikator SID nie odpowiada SID-owi żadnego loginu istniejącego na bieżącej instancji. Powstaje typowo wtedy, gdy bazę odtwarzamy na innej instancji niż ta, na której powstał backup — login o tej samej nazwie mógł zostać utworzony na instancji docelowej niezależnie, w innym momencie, i dostał inny SID, mimo identycznej nazwy tekstowej.


login na instancji A
SID: 0x3EC1...
baza na instancji A
uzytkownik z SID 0x3EC1...
backup .bak
login o tej samej nazwie
na instancji B, inny SID
istnieje niezaleznie,
inny SID
baza odtworzona na B
uzytkownik wciaz z SID 0x3EC1...
osierocony
ALTER USER ... WITH LOGIN
sp_change_users_login Auto_Fix
uzytkownik polaczony
z loginem na instancji B

Uzytkownik odtworzony z backupu zostaje osierocony, bo login o tej samej nazwie na nowej instancji ma inny SID (kliknięcie otwiera pełny rozmiar)

Przykład:

```
UserName      UserSID
------------- ----------------------------------------
nazwa_usera   0x3EC12A5B8F7D4C4A9B6E1234567890AB

(1 rows affected)
```

Pusty wynik oznacza, że żaden użytkownik w bazie nie jest osierocony i dalsze kroki nie są potrzebne.

2. Naprawa powiązania: sp_change_users_login

```
USE NazwaBazy;
GO
EXEC sp_change_users_login 'Auto_Fix', 'NazwaUsera';
```

Naprawiamy powiązanie użytkownika bazy z loginem o tej samej nazwie na bieżącej instancji — opcja `Auto_Fix` odnajduje login o identycznej nazwie i podmienia SID użytkownika w bazie na SID tego loginu.

Przykład:

```
The row for user 'nazwa_usera' will be fixed by updating its login link to a login already in
existence.
The number of orphaned users fixed by updating users was 1.
```

3. Naprawa przez ALTER USER (odpowiednik nowoczesny)

```
ALTER USER NazwaUsera WITH LOGIN = NazwaLoginu;
```

Ten sam efekt co `sp_change_users_login`, składnią zalecaną od nowszych wersji silnika — `sp_change_users_login` pozostaje wspierana, ale dokumentacja Microsoft oznacza ją jako

przeznaczoną głównie do zgodności wstecznej. Różnica praktyczna: `ALTER USER` pozwala też jawnie wskazać inny login niż ten o identycznej nazwie, gdy nazwa użytkownika w bazie i nazwa loginu na instancji celowo się różnią.

Weryfikacja

```
EXEC sp_change_users_login @Action = 'Report';
```

Powtarzamy zapytanie z kroku 1 — poprawnie naprawiony użytkownik nie powinien się już pojawić na liście osieroconych.

Sprawdzamy też, że użytkownik faktycznie może się połączyć i wykonać zapytanie w kontekście naprawionej bazy, logując się bezpośrednio danym loginem:

```
SELECT USER_NAME(), SUSER_SNAME();
```

Przykład:

```
(No column name)   (No column name)
------------------ ------------------
nazwa_usera         NazwaLoginu


(1 rows affected)
```

Wycofanie zmian

Naprawa powiązania SID jest odwracalna wyłącznie przez ponowne wskazanie innego loginu tym samym poleceniem — nie usuwa ona ani użytkownika, ani loginu, więc nie ma efektów ubocznych do wycofania poza samym powiązaniem. Jeśli chcemy przywrócić poprzedni, osierocony stan (np. na potrzeby diagnostyki), wystarczy wskazać z powrotem oryginalny login:

```
ALTER USER NazwaUsera WITH LOGIN = NazwaPoprzedniegoLoginu;
```

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa

`Msg 15291, Level 16, ... @LoginName is not valid`	login o wskazanej nazwie nie istnieje jeszcze na instancji docelowej	tworzymy login (`CREATE LOGIN NazwaLoginu WITH PASSWORD = 'Haslo'`) przed próbą naprawy
`sp_change_users_login` zwraca pustą listę mimo błędów logowania aplikacji	użytkownik nie jest osierocony technicznie, ale login nie ma nadanych uprawnień do bazy	sprawdzamy role użytkownika (`sp_helprotect`) zamiast powiązania SID
Po naprawie aplikacja nadal zgłasza `Login failed for user`	login jest wyłączony (`DISABLED`) albo ma zablokowane hasło po stronie polityki Windows	`ALTER LOGIN NazwaLoginu ENABLE;` i sprawdzenie polityki haseł
Wielu użytkowników osieroconych naraz po masowej migracji baz	loginy na instancji docelowej tworzone niezależnie od backupu, z innymi SID dla całej grupy kont	skrypt pętli po wyniku `sp_change_users_login @Action = 'Report'` dla każdej bazy, wykonujący `Auto_Fix` dla każdego wiersza

Powiązane: Odtwarzanie baz danych · Konta serwisowe i uprawnienia do monitoringu · Konta, role i uprawnienia: GRANT, REVOKE, zasada najmniejszych uprawnień

Źródła: dokumentacja Microsoft Learn, sekcja „sp_change_users_login (Transact-SQL)" i „ALTER USER (Transact-SQL)"