Architektura MSSQL: instancja, baza, pliki MDF/NDF/LDF

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

Z czego składa się instalacja SQL Server — instancja, usługi, bazy systemowe, pliki na dysku — i czym różnią się od siebie tryby recovery oraz edycje silnika.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
RPO	Recovery Point Objective: ile danych wolno stracić przy awarii
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
SSMS	SQL Server Management Studio: graficzna konsola MSSQL
TCP/IP	zestaw protokołów internetu: IP przenosi pakiety między adresami, TCP pilnuje, żeby dane doszły w całości i w kolejności
UDP	User Datagram Protocol: szybki protokół sieciowy bez potwierdzeń

Fiszka

TYLKO ODCZYT

Sprawdzenie, jakie instancje MSSQL są zainstalowane na serwerze

```
Get-Service -Name "MSSQL*"
```

Sprawdzamy, ile instancji silnika działa na danym serwerze Windows i pod jakimi nazwami usług. Nazwa usługi instancji domyślnej to zawsze `MSSQLSERVER`, a instancji nazwanej `MSSQL$NazwaInstancji`.

Przykład:

```
Status   Name               DisplayName
------   ----               -----------
Running  MSSQLSERVER        SQL Server (MSSQLSERVER)
Running  MSSQL$SQLBI        SQL Server (SQLBI)
Running  SQLSERVERAGENT     SQL Server Agent (MSSQLSERVER)
Running  SQLBrowser         SQL Server Browser
```

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

TYLKO ODCZYT

Sprawdzenie nazwy instancji i serwera z poziomu T-SQL

```
SELECT @@SERVERNAME AS NazwaSerwera, SERVERPROPERTY('InstanceName') AS NazwaInstancji;
```

Sprawdzamy, do jakiej instancji jesteśmy akurat podłączeni — przydatne, kiedy na jednym serwerze fizycznym siedzi kilka instancji naraz i łatwo pomylić się w SSMS.

Przykład:

```
NazwaSerwera        NazwaInstancji
------------------- --------------
SQL01\SQLBI       SQLBI


(1 rows affected)
```

T-SQL = Transact-SQL: dialekt SQL używany w Microsoft SQL Server, wywodzący się z języka zapytań oryginalnego silnika Sybase · **SSMS** = SQL Server Management Studio: graficzna konsola MSSQL

Z życia wzięte

Z ŻYCIA WZIĘTE

W poniedziałek o 8:15 Tomek próbuje połączyć się z bazą CDN_Zielona, wpisując w SSMS samą nazwę sql01, tak jak w starej instrukcji od poprzedniego administratora. Połączenie

kończy się błędem, chociaż serwer odpowiada na ping, a księgowość normalnie loguje się do Optimy. Dopiero sprawdzenie usług Windows pokazuje, że silnik dla CDN_Zielona działa jako instancja nazwana sql01\SQLBI, nie jako instancja domyślna. Po dopisaniu nazwy instancji do connection stringa Tomek łączy się bez problemu.

Po co to istnieje

Pojęcie: instancja. Osobna, samodzielna kopia silnika SQL Server działająca na tym samym serwerze fizycznym albo wirtualnym co inne instancje, ale ze swoim własnym zestawem baz systemowych, swoją konfiguracją pamięci, swoim portem i swoim kontem usługi. Instancje nie widzą nawzajem swoich baz — połączenie do jednej z nich nie daje dostępu do baz drugiej.

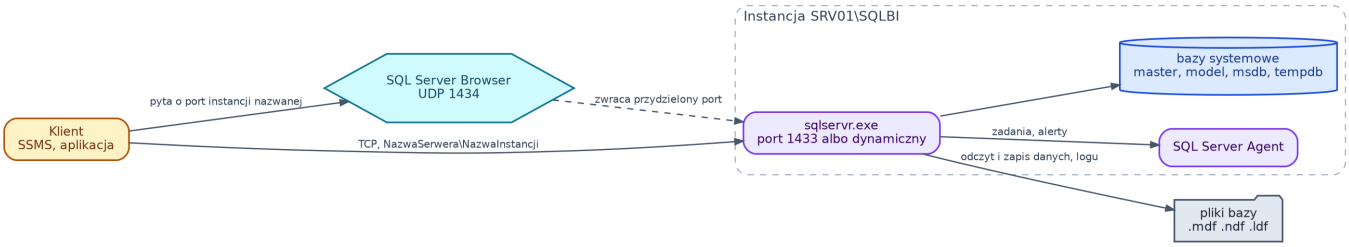
Pojęcie: instancja domyślna i instancja nazwana. Na serwerze może działać dokładnie jedna instancja domyślna, adresowana samą nazwą serwera (`SQL01`), oraz dowolna liczba instancji nazwanych, adresowanych jako `NazwaSerwera\NazwaInstancji` (`SQL01\SQLBI`). Instancja domyślna nasłuchuje zawsze na porcie 1433, instancje nazwane — na porcie przydzielonym dynamicznie, chyba że ustawimy port na stałe w Configuration Managerze.

Rozróżnienie instancja / baza danych / silnik jest tym, co najczęściej myli się na starcie: silnik to program `sqlservr.exe`, instancja to jedno uruchomienie tego programu ze swoim kontekstem, a baza danych to pojedynczy zbiór danych żyjący wewnątrz instancji. Jedna instancja trzyma zwykle wiele baz — systemowe plus te, które są przedmiotem naszej pracy.

Jak to działa

Rozwiń: jak to działa pod spodem

Pojęcie: usługi SQL Server. Instalacja silnika stawia na Windows kilka usług, z których każda odpowiada za inny fragment funkcjonalności.



Klient pyta SQL Server Browser o port instancji nazwanej, po czym łączy się z sqlservr.exe, który korzysta z baz systemowych i zapisuje dane do plików mdf, ndf i ldf (kliknięcie otwiera pełny rozmiar)

Usługa	Nazwa procesu / usługi	Za co odpowiada
SQL Server (silnik)	`sqlservr.exe`, `MSSQLSERVER` lub `MSSQL$NazwaInstancji`	wykonywanie zapytań, przechowywanie danych, logi transakcyjne
SQL Server Agent	`SQLSERVERAGENT` lub `SQLAGENT$NazwaInstancji`	zadania harmonogramowane, alerty, powiadomienia
SQL Server Browser	`SQLBrowser`	odpowiada klientom, na jakim porcie nasłuchuje dana instancja nazwana
SQL Server VSS Writer	`SQLWriter`	integracja z backupem na poziomie systemu plików (VSS)

Pojęcie: port TCP silnika. Instancja domyślna nasłuchuje standardowo na porcie 1433/TCP. Instancje nazwane dostają port dynamiczny przydzielany przy każdym starcie usługi, chyba że administrator wymusi port stały w Configuration Managerze, w zakładce protokołu TCP/IP. Klient, który zna tylko nazwę instancji, a nie zna portu, pyta o niego usługę SQL Server Browser działającą na porcie UDP 1434 — to dlatego wyłączenie tej usługi łamie połączenia po nazwie instancji, mimo że sam silnik działa poprawnie.

Pojęcie: bazy systemowe. Każda instancja, zaraz po instalacji, ma cztery bazy systemowe, które trzymają metadane samej instancji i nie są przedmiotem naszej pracy z danymi biznesowymi.

Baza	Co trzyma
`master`	konfigurację instancji, listę wszystkich baz danych, loginy na poziomie serwera, ustawienia konfiguracyjne
`model`	szablon, z którego kopiowana jest każda nowo tworzona baza — zmiana w `model` wpływa na wszystkie kolejne `CREATE DATABASE`

Baza	Co trzyma
`msdb`	historię i definicje zadań SQL Server Agent, historię backupów i restore'ów, plany konserwacyjne, alerty
`tempdb`	tabele tymczasowe, sortowania na dysku, wersje wierszy pod izolację `READ COMMITTED SNAPSHOT` — jedyna baza systemowa tworzona od nowa przy każdym starcie usługi

> **Pojęcie: plik danych i plik logu.** Fizycznie baza danych to zestaw plików na dysku, nigdy jeden monolityczny plik.

Rozszerzenie	Nazwa	Rola
`.mdf`	primary data file	pierwszy, obowiązkowy plik danych — trzyma też katalog systemowy bazy
`.ndf`	secondary data file	dodatkowe pliki danych, opcjonalne, używane do rozłożenia danych na kilka dysków albo grup plików
`.ldf`	log file	plik logu transakcyjnego — sekwencyjny zapis każdej zmiany, zanim trafi ona do plików danych

> **Pojęcie: grupa plików (filegroup).** Logiczny kontener grupujący jeden lub więcej plików danych (`.mdf` / `.ndf`), do którego przypisujemy tabele i indeksy. Grupa `PRIMARY` istnieje zawsze i zawiera plik `.mdf`. Dodatkowe grupy pozwalają rozłożyć duże tabele na osobne dyski albo trzymać archiwalne partycje na wolniejszym storage'u, bez zmiany ani jednego zapytania — to `CREATE TABLE ... ON NazwaGrupyPlikow` decyduje, gdzie fizycznie wyląduje tabela.

> **Pojęcie: autogrowth.** Automatyczne powiększanie pliku danych albo logu, kiedy zapełni się do końca, o wartość skonfigurowaną jako stały rozmiar (np. 256 MB) albo procent bieżącego rozmiaru. Domyślne wartości fabryczne (1 MB dla danych, 10% dla logu) są zbyt małe dla realnej produkcji — częste, drobne powiększenia fragmentują plik na dysku i chwilowo blokują zapis, dlatego ustawiamy autogrowth ręcznie na sensowną stałą wartość zaraz po utworzeniu bazy.

Tryby recovery

> **Pojęcie: tryb recovery (recovery model).** Ustawienie bazy danych decydujące o tym, jak długo silnik trzyma zapisy w logu transakcyjnym i jakie opcje odtwarzania są w ogóle dostępne. To jedna z pierwszych rzeczy, którą sprawdzamy na nieznanej bazie — opisane szerzej na stronie Log transakcyjny: WAL, redo, transaction log.

Tryb	Co dzieje się z logiem	Backup logu	Odtwarzanie do punktu w czasie
`SIMPLE`	log jest obcinany automatycznie po każdym checkpoincie, nie rośnie w nieskończoność	niedostępny — nie ma czego backupować	niedostępne
`FULL`	log rośnie, dopóki nie zrobimy jego backupu — dopiero backup logu zwalnia miejsce	wymagany do zachowania ciągłości odtwarzania	dostępne, z dokładnością do transakcji
`BULK_LOGGED`	operacje masowe (`BULK INSERT`, przebudowa indeksu, `SELECT INTO`) logowane są minimalnie, reszta jak w `FULL`	dostępny, ale backup logu obejmujący operację masową nie pozwala na PITR w jej trakcie	ograniczone — nie działa dla przedziału z operacją masową

Wybór trybu jest kompromisem między rozmiarem logu a możliwością odtworzenia bazy na sekundę przed awarią. Baza produkcyjna, dla której liczy się RPO bliski zeru, pracuje w `FULL` i ma regularny backup logów opisany na stronie Backup pełny, różnicowy, przyrostowy, logi. Baza testowa albo hurtownia zasilana raz dziennie, dla której strata dnia danych jest akceptowalna, może bezpiecznie pracować w `SIMPLE`.

Edycje silnika

> **Pojęcie: edycja (edition).** Wariant licencyjny silnika SQL Server, decydujący o limitach zasobów i dostępności niektórych funkcji — sprawdzenie zainstalowanej edycji opisujemy na stronie Rozpoznanie instancji i baz.

Edycja	Limit pamięci silnika	Limit rdzeni	Typowe zastosowanie
Express	1410 MB bufora	4 rdzenie	aplikacje wbudowane, małe wdrożenia, brak kosztu licencji
Standard	128 GB bufora	24 rdzenie	typowa produkcja średniej wielkości

Edycja	Limit pamięci silnika	Limit rdzeni	Typowe zastosowanie
Enterprise	brak limitu	brak limitu	duże instalacje, `ONLINE` rebuild indeksów, partycjonowanie tabel, Always On z wieloma repликami
Developer	jak Enterprise	jak Enterprise	funkcjonalnie pełna, ale licencyjnie wyłącznie do środowisk deweloperskich i testowych, nie do produkcji

Różnica między Standard a Enterprise dotyczy administratora najbardziej przy operacjach na indeksach — `ALTER INDEX ... REBUILD WITH (ONLINE = ON)` wymaga Enterprise, temat rozwinięty na stronie Indeksy: status, fragmentacja, przebudowa, oraz przy skalowaniu, gdzie limit rdzeni i pamięci w Standard bywa pierwszym wąskim gardłem, jaki napotyka rosnąca baza.

Częste błędy

Mylenie instancji z serwerem. Adres `SQL01` bez nazwy instancji łączy zawsze z instancją domyślną — jeśli na serwerze stoi wyłącznie instancja nazwana `SQL01\SQLBI`, samo `SQL01` zwróci błąd połączenia, mimo że serwer żyje i odpowiada na ping.

Traktowanie `tempdb` jak zwykłej bazy do trzymania danych roboczych na stałe. `tempdb` jest tworzona od zera przy każdym restarcie usługi — wszystko, co w niej zapisano, znika po restarcie silnika, łącznie z tabelami tymczasowymi utworzonymi ręcznie jako „stałe" miejsce robocze.

Zostawienie domyślnego autogrowth na produkcji. Plik danych rosnący o 1 MB przy bazie zapisującej gigabajty dziennie powiększa się tysiące razy, każde powiększenie to pauza w zapisie i fragmentacja pliku na poziomie systemu plików.

Powiązane: Rozpoznanie instancji i baz · Log transakcyjny: WAL, redo, transaction log · Rozmiar bazy: pliki, tablespace'y, autoextend · Indeksy: status, fragmentacja, przebudowa

Źródła: dokumentacja Microsoft Learn, sekcja „SQL Server instances" i „Database Files and Filegroups"

Revision #42
Created 2026-09-25 00:00:03 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:34 UTC by Tomasz Konieczny