SQL Agent: zadania i harmonogramy

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

Sprawdzenie, jakie zadania są zdefiniowane, które akurat działają, jak wygląda ich historia i jak ręcznie uruchomić albo zatrzymać zadanie z poziomu T-SQL.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
SSMS	SQL Server Management Studio: graficzna konsola MSSQL
T-SQL	Transact-SQL: dialekt SQL używany w Microsoft SQL Server, wywodzący się z języka zapytań oryginalnego silnika Sybase

Fiszka

TYLKO ODCZYT

Sprawdzenie aktualnie działających zadań SQL Agent

```
WITH CTE_Sysession (AgentStartDate) AS (
    SELECT MAX(agent_start_date) FROM msdb.dbo.syssessions
)
SELECT
    sjob.name AS NazwaJoba,
    CASE WHEN sjob.enabled = 1 THEN 'Enabled' ELSE 'Disabled' END AS Stan,
    act.start_execution_date AS DataStartu,
    DATEDIFF(MINUTE, act.start_execution_date, GETDATE()) AS CzasTrwaniaMin,
    act.last_executed_step_id AS OstatniKrok
FROM msdb.dbo.syssessions AS sys1
```

```
JOIN CTE_Sysession AS sys2 ON sys2.AgentStartDate = sys1.agent_start_date
JOIN msdb.dbo.sysjobactivity act ON act.session_id = sys1.session_id
JOIN msdb.dbo.sysjobs sjob ON sjob.job_id = act.job_id
WHERE act.start_execution_date IS NOT NULL AND act.stop_execution_date IS NULL
ORDER BY act.start_execution_date DESC;
```

Sprawdzamy zadania, które są aktualnie w trakcie wykonywania — filtr `start_execution_date IS NOT NULL AND stop_execution_date IS NULL` wyłania dokładnie te wpisy z `sysjobactivity`, dla których zadanie wystartowało, ale jeszcze się nie zakończyło. Ograniczenie do najnowszej sesji agenta (`CTE_Sysession`) jest konieczne, bo `sysjobactivity` przechowuje też historyczne wpisy z poprzednich uruchomień samej usługi SQL Agent.

Przykład:

```
NazwaJoba                           Stan      DataStartu               CzasTrwaniaMin
OstatniKrok
----------------------------------- --------- ------------------------ --------------
-------------
IndexOptimize - USER_DATABASES      Enabled   2026-09-25 06:52:36.000  34
2
cdc_CDN_Zielona_capture                   Enabled   2026-09-25 06:41:53.000
45               1


(2 rows affected)
```

SQL = Structured Query Language: język zapytań do relacyjnych baz danych · **join** = połącz dwa pliki tekstowe po wspólnej kolumnie, jak złączenie tabel w bazie danych

TYLKO ODCZYT

Wylistowanie wszystkich zadań i ich statusu włączenia

```
SELECT name AS NazwaJoba, enabled AS Wlaczony, date_created AS DataUtworzenia
FROM msdb.dbo.sysjobs
ORDER BY name;
```

Sprawdzamy pełną listę zdefiniowanych zadań niezależnie od tego, czy akurat działają — pierwszy przegląd na nieznanej instancji, żeby zorientować się, jakie zadania konserwacyjne i integracyjne są w ogóle skonfigurowane.

TYLKO ODCZYT

Sprawdzenie historii wykonań zadania

```
SELECT
    j.name AS NazwaJoba,
    h.run_date AS DataUruchomienia,
    h.run_time AS GgodzinaUruchomienia,
    CASE h.run_status
        WHEN 0 THEN 'Failed'
        WHEN 1 THEN 'Succeeded'
        WHEN 3 THEN 'Canceled'
        WHEN 4 THEN 'InProgress'
    END AS Status,
    h.message AS Komunikat
FROM msdb.dbo.sysjobhistory h
JOIN msdb.dbo.sysjobs j ON j.job_id = h.job_id
WHERE j.name = N'NazwaJoba'
ORDER BY h.instance_id DESC;
```

Sprawdzamy historię ostatnich uruchomień konkretnego zadania, razem z komunikatem zwróconym przez ostatni krok — pierwszy krok diagnozy, kiedy zadanie zgłoszone jako nieudane trzeba zrozumieć bez zaglądania osobno w SSMS.

Przykład:

```
NazwaJoba          DataUruchomienia   GodzinaUruchomienia   Status      Komunikat
------------------ ------------------ --------------------- -----------
--------------------------------------
BackupPelny_CDN_Zielona 20260925           30000                 Succeeded   The job
succeeded.


(1 rows affected)
```

SSMS = SQL Server Management Studio: graficzna konsola MSSQL · **join** = połącz dwa pliki tekstowe po wspólnej kolumnie, jak złączenie tabel w bazie danych

TYLKO ODCZYT

Ręczne uruchomienie zadania: sp_start_job

```
EXEC msdb.dbo.sp_start_job @job_name = N'NazwaJoba';
```

Uruchamiamy zadanie natychmiast, poza jego harmonogramem — przydatne przy testowaniu nowo utworzonego zadania albo przy potrzebie jednorazowego, doraźnego wykonania.

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

TYLKO ODCZYT

Ręczne zatrzymanie zadania: sp_stop_job

```
EXEC msdb.dbo.sp_stop_job @job_name = N'NazwaJoba';
```

Zatrzymujemy zadanie, które akurat się wykonuje — silnik przerywa bieżący krok i oznacza uruchomienie jako `Canceled` w historii.

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

Z życia wzięte

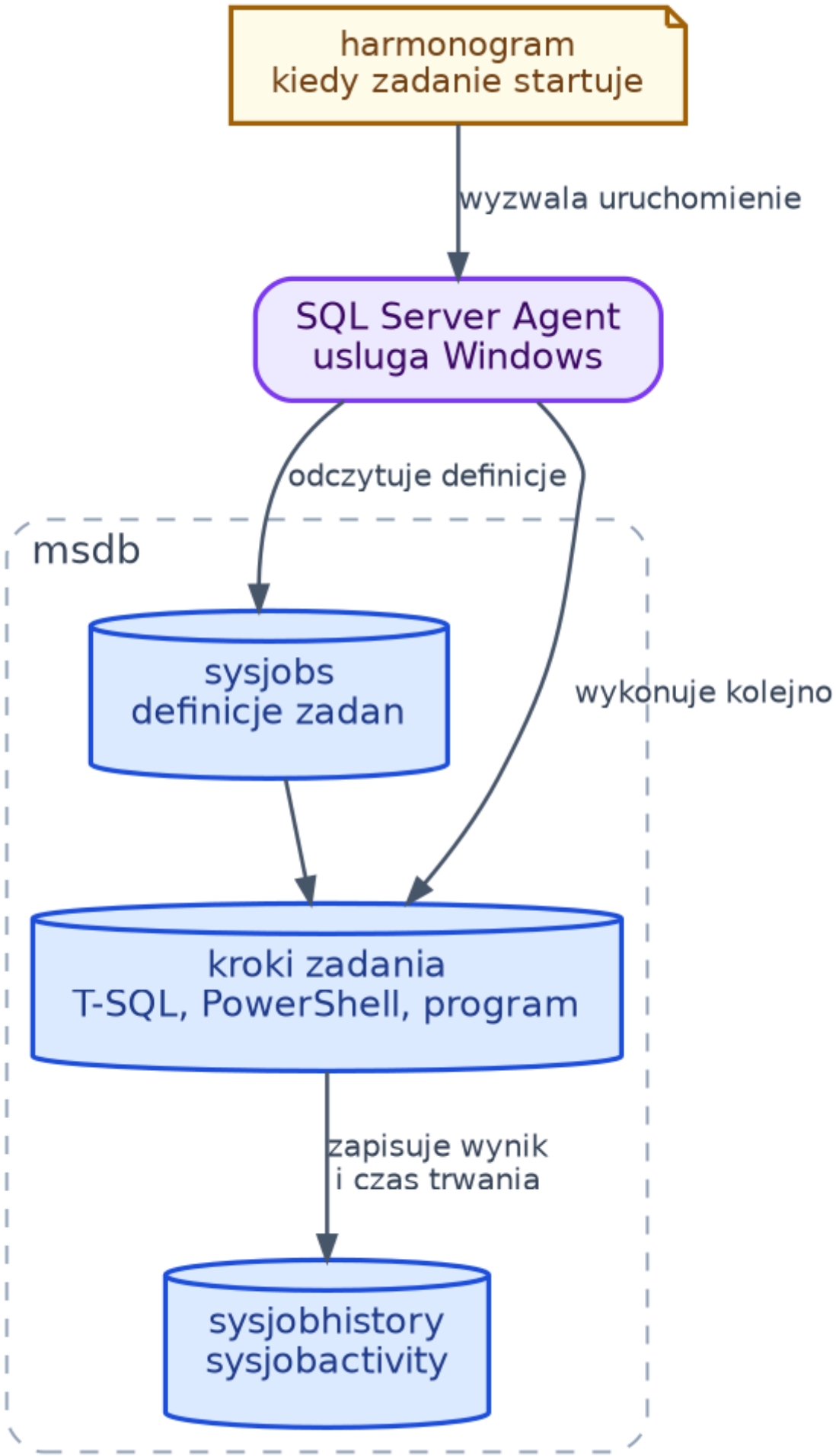
Z ŻYCIA WZIĘTE

W poniedziałek rano pani Halina zgłasza, że nocny import kursów walut do CDN_Zielona nie zadziałał, bo w Optimie widnieją wczorajsze kursy. Sprawdzamy historię zadania w sysjobhistory i widzimy, że job zakończył się statusem Failed już trzeci dzień z rzędu, ale nikt tego nie zauważył, bo powiadomienia mailowe SQL Agenta przestały działać po zmianie hasła konta serwisowego poczty. Ręczne uruchomienie zadania przez sp_start_job od razu pokazuje prawdziwy błąd: brak dostępu do zewnętrznego serwera z kursami. Po naprawie dostępu i sprawdzeniu konfiguracji powiadomień zadanie znowu działa zgodnie z harmonogramem.

Po co to istnieje

Pojęcie: SQL Server Agent. Osobna usługa Windows odpowiedzialna za wykonywanie zadań według harmonogramu — backupy, reindeksację, przetwarzanie wsadowe, alerty na zdarzenia silnika. Zadania (`Jobs`) składają się z kroków (`Steps`), a każdy krok może być poleceniem T-SQL, skryptem PowerShell albo zewnętrznym programem. Definicje zadań, ich

harmonogramy i cała historia wykonań żyją w bazie `msdb` — stąd wszystkie zapytania na tej stronie odwołują się właśnie do niej.


harmonogram
kiedy zadanie startuje
wyzwala uruchomienie
SQL Server Agent
usluga Windows
odczytuje definicje
msdb
sysjobs
definicje zadan
wykonuje kolejno
kroki zadania
T-SQL, PowerShell, program
zapisuje wynik
i czas trwania
sysjobhistory
sysjobactivity

SQL Server Agent uruchamia zadania wedlug harmonogramu i zapisuje historie w msdb (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: plany konserwacyjne (Maintenance Plans).** Graficzny, wbudowany w SSMS sposób definiowania typowych zadań konserwacyjnych — backupów, reindeksacji, sprawdzania integralności (`DBCC CHECKDB`) — bez pisania T-SQL ręcznie. Plany konserwacyjne tworzą pod spodem zwykłe zadania SQL Agent, więc zapytania do `sysjobs` i `sysjobhistory` pokazują je tak samo jak zadania utworzone ręcznie czy przez zewnętrzne skrypty (jak popularny zestaw procedur Ola Hallengren, wspomniany na stronie Rozpoznanie procedur składowanych).

Częste błędy

Sprawdzanie stanu zadania wyłącznie po fakcie, że jest `Enabled`. Włączone zadanie może mieć nieaktywny harmonogram albo brak przypisanego harmonogramu w ogóle — `enabled = 1` mówi tylko, że zadanie nie zostało ręcznie wyłączone, nie że faktycznie kiedykolwiek się uruchomi.

Odczytanie `run_duration` bez rozbicia na godziny/minuty/sekundy. Kolumna `run_duration` w `sysjobhistory` przechowuje czas trwania jako liczbę w formacie `HHMMSS` sklejoną w jedną liczbę całkowitą (np. `10234` oznacza godzinę, dwie minuty i trzydzieści cztery sekundy), a nie jako liczbę sekund — bezpośrednie porównania liczbowe tej kolumny dają błędne wnioski.

Powiązane: Rozpoznanie procedur składowanych · Backup baz danych · Indeksy: status, fragmentacja, przebudowa

Źródła: dokumentacja Microsoft Learn, sekcja „SQL Server Agent" i „sp_start_job (Transact-SQL)"

Revision #43
Created 2025-03-09 16:16:00 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:51 UTC by Tomasz Konieczny