Konta serwisowe i uprawnienia do monitoringu

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

Nadawanie loginowi tylko tych uprawnień, których faktycznie potrzebuje — na przykładzie konta systemu monitorującego, które ma czytać stan instancji, a nie mieć dostęp do danych.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
SSMS	SQL Server Management Studio: graficzna konsola MSSQL

Fiszka

ZMIENIA SYSTEM

Nadanie uprawnienia do stanu serwera: VIEW SERVER STATE

```
GRANT VIEW SERVER STATE TO NazwaLoginu;
```

Nadajemy loginowi prawo odpytywania większości widoków dynamicznego zarządzania (`sys.dm_exec_*`, `sys.dm_os_*`) na poziomie całej instancji — bez tego uprawnienia zapytania do tych widoków zwracają puste wyniki albo błąd odmowy dostępu, mimo że sama składnia jest poprawna.

ZMIENIA SYSTEM

Nadanie uprawnienia do definicji obiektów: VIEW ANY DEFINITION

```
GRANT VIEW ANY DEFINITION TO NazwaLoginu;
```

Nadajemy prawo odczytu definicji obiektów (treści procedur, widoków, funkcji) na całej instancji, bez nadawania prawa do samych danych — przydatne dla narzędzi dokumentujących strukturę bazy albo systemów monitorujących, które muszą znać nazwy i definicje obiektów, ale nie mają czytać wierszy tabel.

□□ ZMIENIA SYSTEM

Nadanie odczytu do konkretnej tabeli systemowej: SELECT na obiekcie

```
GRANT SELECT ON OBJECT::msdb.dbo.sysjobs TO NazwaLoginu;
```

Nadajemy odczyt pojedynczej tabeli systemowej wewnątrz `msdb` — typowy krok przy podłączaniu zewnętrznego systemu monitoringu, który sam odpytuje stan zadań SQL Agent zamiast czekać na alert wysłany przez sam silnik.

□□ **SQL** = Structured Query Language: język zapytań do relacyjnych baz danych

□□ ZMIENIA SYSTEM

Nadanie prawa do wykonania procedury systemowej: EXECUTE na obiekcie

```
GRANT EXECUTE ON OBJECT::msdb.dbo.agent_datetime TO NazwaLoginu;
```

Nadajemy prawo wywołania konkretnej procedury systemowej bez nadawania szerszych uprawnień do całej bazy `msdb` — zasada najmniejszych uprawnień w praktyce: konto monitorujące dostaje dokładnie te trzy czy cztery uprawnienia, które są mu rzeczywiście potrzebne, zamiast roli `db_owner` na wyrost.

□□ ZMIENIA SYSTEM

Zestaw uprawnień dla konta monitorującego SQL Agent

```
USE msdb;
GO

GRANT SELECT ON OBJECT::msdb.dbo.sysjobs TO NazwaLoginu;
GRANT SELECT ON OBJECT::msdb.dbo.sysjobactivity TO NazwaLoginu;
```

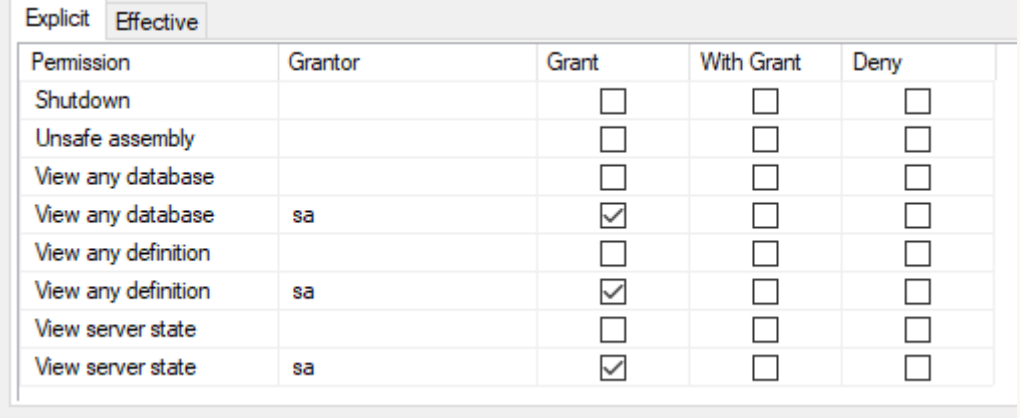
```
GRANT SELECT ON OBJECT::msdb.dbo.sysjobservers TO NazwaLoginu;
GRANT EXECUTE ON OBJECT::msdb.dbo.agent_datetime TO NazwaLoginu;
GO

USE master;
GO

GRANT VIEW SERVER STATE TO NazwaLoginu;
GRANT VIEW ANY DEFINITION TO NazwaLoginu;
GO
```

Komendy wykonujemy jako blok, w tej kolejności, ponieważ pierwsza część działa w kontekście `msdb`, a druga w kontekście `master` — to jedyna sytuacja, w której łączymy kilka poleceń `GRANT` w jeden blok, bo razem tworzą kompletny, gotowy do wklejenia zestaw uprawnień pod jedno konto serwisowe, na przykład konto systemu Zabbix albo innego narzędzia monitorującego instancję z zewnątrz.

Ten sam komplet uprawnień widać w oknie właściwości loginu w SSMS, na zakładce Securables — kolumna Grant pokazuje, co konto faktycznie dostało.



Zakładka Securables we właściwościach loginu monitorującego: nadane View any database, View any definition i View server state. (kliknięcie otwiera pełny rozmiar)

SQL = Structured Query Language: język zapytań do relacyjnych baz danych · **SSMS** = SQL Server Management Studio: graficzna konsola MSSQL

Z życia wzięte

Z ŻYCIA WZIĘTE

We wtorek Ola prosi o konto dla nowego eksportera Zabbixa, żeby monitorować kondycję sql01 razem z resztą infrastruktury. Zamiast dodać je od razu do roli sysadmin, bo tak najszybciej, nadajemy dokładnie te uprawnienia, których potrzebuje: VIEW SERVER STATE i odczyt kilku tabel w msdb. Miesiąc później hasło tego konta wycieka razem z konfiguracją Zabbixa skopiowaną na złe repozytorium, ale ryzyko ogranicza się do samego monitoringu, nie do całej instancji.

Po co to istnieje

Pojęcie: login i użytkownik bazy danych. W SQL Server to dwa osobne poziomy uprawnień. Login (`CREATE LOGIN`) istnieje na poziomie całej instancji i decyduje, czy dane konto w ogóle może się połączyć. Użytkownik bazy danych (`CREATE USER`) istnieje wewnątrz konkretnej bazy i mapuje się na login — dopiero on decyduje, do czego dane konto ma dostęp wewnątrz tej konkretnej bazy. Ten sam login może mieć różnych użytkowników — a więc różne uprawnienia — w różnych bazach na tej samej instancji. Powiązanie między loginem a użytkownikiem, oraz sytuacje, w których się ono rozjeżdża po odtworzeniu bazy z innego serwera, opisujemy na stronie Naprawa uprawnień użytkownika SQL po odtworzeniu baz.

Konta systemów monitorujących (Zabbix, Nagios, Prometheus z odpowiednim exporterem) potrzebują dostępu tylko do odczytu stanu instancji — nigdy do danych biznesowych. Nadanie takiemu kontu roli `sysadmin` na wyrost, bo „tak jest szybciej", jest najczęstszym błędem przy podłączaniu monitoringu i tworzy niepotrzebne ryzyko: kompromitacja hasła narzędzia monitorującego oznacza wtedy pełny dostęp administracyjny do wszystkich baz na instancji.

Jak to działa

Uprawnienia `GRANT ... ON OBJECT::` są uprawnieniami na poziomie pojedynczego obiektu, nie roli. Zamiast dodawać login do wbudowanej roli bazodanowej `db_datareader`, która dałaby odczyt do wszystkich tabel bazy `msdb` włącznie z tymi, których monitoring nigdy nie odpyta, nadajemy `SELECT` na dokładnie tych trzech, czterech tabelach, których faktycznie potrzebuje zapytanie widoczne po stronie narzędzia monitorującego.

> **Pojęcie: rola serwerowa i rola bazodanowa.** Rola serwerowa (`sysadmin`, `serveradmin`, `dbcreator`) działa na poziomie całej instancji. Rola bazodanowa (`db_owner`, `db_datareader`, `db_datawriter`, `db_backupoperator`) działa wewnątrz jednej bazy. Nadanie loginowi członkostwa w roli jest szybsze niż wypisywanie uprawnień obiekt po obiekcie, ale nadaje od razu cały zestaw uprawnień tej roli — dla kont serwisowych o wąskim zakresie odpowiedzialności wypisanie pojedynczych `GRANT` jest bardziej precyzyjne, choć bardziej pracochłonne przy pierwszym ustawieniu.

Częste błędy

Nadanie roli `sysadmin` kontu monitorującemu „na wszelki wypadek". To najszybsza droga do sytuacji, w której hasło jednego konta serwisowego, wycieknięte z konfiguracji narzędzia trzeciego, daje pełną kontrolę nad wszystkimi bazami na instancji.

Zapomnienie o `VIEW SERVER STATE` przy podłączaniu monitoringu. Zapytania do `sys.dm_exec_sessions` czy `sys.dm_os_performance_counters` bez tego uprawnienia zwracają pusty wynik zamiast błędu — co bywa mylące, bo wygląda jak brak aktywności na serwerze, a nie jak brak uprawnień.

Powiązane: Naprawa uprawnień użytkownika SQL po odtworzeniu baz · Konta, role i uprawnienia: GRANT, REVOKE, zasada najmniejszych uprawnień · SQL Agent: zadania i harmonogramy

Źródła: dokumentacja Microsoft Learn, sekcja „GRANT (Transact-SQL)" i „Server-Level Roles"

Revision #41
Created 2026-09-25 00:00:09 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:39 UTC by Tomasz Konieczny