Rozpoznanie procedur składowanych

Poziom: podstawy **Dotyczy: SQL Server 2016–2022** **Czas: ~10 min**

Sprawdzenie, jakie procedury składowane istnieją w bazie i jak wygląda ich definicja, kiedy trafiamy na nieznaną wcześniej bazę bez dokumentacji.

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Fiszka

TYLKO ODCZYT

Wylistowanie procedur składowanych w bazie: INFORMATION_SCHEMA.ROUTINES

```
SELECT ROUTINE_SCHEMA, ROUTINE_NAME, ROUTINE_TYPE, CREATED, LAST_ALTERED
FROM NazwaBazy.INFORMATION_SCHEMA.ROUTINES
WHERE ROUTINE_TYPE = 'PROCEDURE'
ORDER BY ROUTINE_NAME;
```

Sprawdzamy, jakie procedury składowane istnieją we wskazanej bazie — `INFORMATION_SCHEMA.ROUTINES` obejmuje zarówno procedury (`PROCEDURE`), jak i funkcje (`FUNCTION`), stąd filtr na `ROUTINE_TYPE`. To pierwsze zapytanie na bazie, o której wiemy tylko, że jest zasilana logiką z poziomu procedur, a nie mamy do niej dokumentacji.

Przykład:

```
ROUTINE_SCHEMA   ROUTINE_NAME            ROUTINE_TYPE   CREATED
LAST_ALTERED
---------------- ----------------------- -------------- ----------------------
----------------------
dbo              IndexOptimize           PROCEDURE      2022-03-14 09:12:33.000 2023-
11-02 15:40:11.000
dbo              usp_ArchiwizujZamowienia PROCEDURE      2021-06-01 10:00:00.000 2021-
06-01 10:00:00.000

(2 rows affected)
```

TYLKO ODCZYT

Wyświetlenie definicji procedury: sp_helptext

```
EXEC sp_helptext N'NazwaProcedury';
```

Sprawdzamy pełną treść procedury — jej definicję `CREATE PROCEDURE` w oryginalnym sformatowaniu, tak jak została zapisana przy tworzeniu. Wykonujemy to zapytanie na bazie, która zawiera daną procedurę, więc wcześniej przełączamy kontekst poleceniem `USE NazwaBazy`.

Przykład:

```
Text
------------------------------------------------------------
CREATE PROCEDURE dbo.IndexOptimize
    @Databases NVARCHAR(MAX) = NULL,
    @FragmentationLow NVARCHAR(MAX) = NULL
AS
BEGIN
    SET NOCOUNT ON;
    -- ...

(18 rows affected)
```

Ten sam efekt daje odpytanie `sys.sql_modules` bezpośrednio, co bywa wygodniejsze w skryptach, bo zwraca definicję jako pojedynczy tekst w kolumnie `definition`, zamiast wielu wierszy jak `sp_helptext`:

```
SELECT definition FROM sys.sql_modules WHERE object_id = OBJECT_ID(N'NazwaProcedury');
```

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

Z życia wzięte

Z ŻYCIA WZIĘTE

Tomek dostaje zgłoszenie, że raport miesięczny w Optimie czasem się zawiesza, ale w aplikacji nie widać żadnego kodu odpowiedzialnego za jego generowanie. Sprawdza listę procedur składowanych w CDN_Zielona i znajduje usp_ArchiwizujZamowienia, o której nikt w zespole wcześniej nie słyszał. Próba odczytania jej treści przez sp_helptext kończy się pustym wynikiem, bo procedura została utworzona z opcją WITH ENCRYPTION. Dopiero rozmowa z Olą ujawnia, że to stary skrypt jednego z poprzednich programistów, zaszyfrowany na wszelki wypadek.

Po co to istnieje

Procedury składowane bywają jedynym miejscem, w którym opisana jest logika biznesowa działająca na bazie — harmonogramy konserwacyjne (jak popularna procedura `IndexOptimize` ze zbioru narzędzi Ola Hallengren, opisana szerzej na stronie Indeksy: status, fragmentacja, przebudowa), procedury archiwizujące dane albo integracje z zewnętrznymi systemami. Zanim zmienimy cokolwiek w bazie, której nie znamy, sprawdzamy, jakie procedury na niej działają i co dokładnie robią — `sp_helptext` jest do tego szybsze niż szukanie kodu źródłowego w repozytorium, którego może w ogóle nie być.

Częste błędy

Brak wyniku z `sp_helptext` dla procedury zaszyfrowanej. Procedura utworzona z opcją `WITH ENCRYPTION` nie zwraca definicji przez `sp_helptext` ani przez `sys.sql_modules` — silnik świadomie ukrywa treść, co bywa zaskoczeniem przy pierwszym natrafieniu na taki obiekt.

Szukanie procedury w złej bazie. `INFORMATION_SCHEMA.ROUTINES` i `sp_helptext` działają w kontekście bazy, na której akurat jesteśmy — procedura o tej samej nazwie może istnieć w kilku bazach naraz z zupełnie inną treścią, dlatego zawsze warto zacząć od jawnego `USE NazwaBazy`.

Powiązane: Rozpoznanie instancji i baz · Indeksy: status, fragmentacja, przebudowa · SQL Agent: zadania i harmonogramy

Źródła: dokumentacja Microsoft Learn, sekcja „sp_helptext" i „INFORMATION_SCHEMA.ROUTINES"

Revision #41
Created 2026-09-25 00:00:10 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:40 UTC by Tomasz Konieczny