Role na SQL Server

Role możemy rozumieć jako grupy użytkowników, do których możemy nadawać uprawnienia tak samo, jak by byli do normalni użytkownicy. Do roli możemy dodawać użytkowników lub inne role.

Rozważmy następującą sytuację: wszyscy pracownicy bazy Hurtownia mają identyczne uprawnienia w bazie. Zatrudniamy nowego pracownika – jeśli nie używamy ról, musimy nadać pracownikowi wszystkie uprawnienia od początku. Przy dużej liczbie obiektów staje się to kłopotliwe. Role usprawniają ten proces.

Rozróżniamy cztery rodzaje ról:

* Serwerowe
* Bazodanowe
* Definiowane przez użytkownika
* Aplikacyjne

# Role serwerowe

Do ról na poziomie serwera należą konta logowania, a nie użytkownicy, Role te znajdziemy w Management Studio po rozwinięciu gałęzi Security i Server Roles. Nie możemy tworzyć nowych ról tego typu. Istnieją tylko wbudowane role serwerowe. Najważniejsze z nich to:

* sysadmin – użytkownicy przypisani do tej roli mają pełne prawa administratora dla całej instancji
* dbcreator – użytkownicy przypisani do tej roli mogą tworzyć i modyfikować bazy danych
* diskadmin – użytkownicy przypisani do tej roli mogą zarządzać plikami
* processadmin – użytkownicy przypisani do tej roli mogą zarządzać procesami serwera
* serveradmin – użytkownicy przypisani do tej roli mogą wykonywać wiele czynności administracyjnych, lecz nie wszystkie (patrz dokumentacja SQL Server)
* securityadmin – użytkownicy przypisani do tej roli zarządzają kontami logowania
* setupadmin – użytkownicy przypisani do tej roli mogą zarządzać konfiguracją serwera
* bulkadmin – użytkownicy przypisani do tej roli uruchamiać polecenie BULK INSERT masowego wstawiania danych

Aby sprawdzić jakie szczegółowe uprawnienia kryją się za poszczególną rolą serwerową możemy wykonać procedurę składowaną sp\_srvrolepermission:

EXEC sp\_srvrolepermission ‘nazwaRoliSerwera’;

Z kolei aby dodać wybrane konto logowania do którejś z wymienionych ról administracyjnych można użyć programu Management Studio lub zapytania SQL z procedurą sp\_addsrvrolemember:

EXEC sp\_addsrvrolemember ‘kontoLogowania’, ‘nazwaRoli’;

Z kolei aby usunąć konto z roli administracyjnej można wykorzystać program Management Studio lub SQL i zapytania SQL i procedury sp\_dropsrvrolemember:

EXEC sp\_dropsrvrolemember ‘kontoLogowania’, ‘nazwaRoli’;

# Role bazodanowe

W każdej bazie danych istnieje pewien zestaw predefiniowanych ról. Role bazodanowe to zbiór uprawnień przyznawany do określonej bazy danych. Na serwerze SQL istnieją następujące role bazodanowe:

* public – należy do niej każdy użytkownik bazy danych. Domyślnie rola nie ma żadnych uprawnień. W odróżnieniu od konta guest aby skorzystać z uprawnień roli, konto, na które jesteśmy zalogowani musi mieć użytkownika w tej bazie
* db\_owner – użytkownik tej roli ma prawa administracyjne w danej bazie
* db\_accessadmin – użytkownik tej roli zarządza użytkownikami i rolami
* db\_securityadmin – użytkownik tej roli zarządza uprawnieniami obiektów
* db\_ddladmin – użytkownik może wykonywać instrukcje CREATE, ALTER, DROP
* db\_backupoperator – użytkownik roli może wykonywać kopię zapasową bazy
* db\_datareader – użytkownik może odczytywać dane ze wszystkich tabel
* db\_datawriter – użytkownik może modyfikować dane we wszystkich tabelach
* db\_denydatareader – użytkowni ma zabroniony odczyt z tabel bazy
* db\_denydanawriter – zabrania się użytkownikowi tej roli modyfikacji w tabelach bazy danych

Aby sprawdzić jakie szczegółowe uprawnienia kryją się za poszczególną rolą bazy danych możemy wykonać procedurę składowaną sp\_dbfixedrolepermission:

EXEC sp\_dbfixedrolepermission ‘nazwaRoli’;

# Tworzenie i usuwanie ról (role użytkownika)

## Tworzenie nowej roli:

CREATE ROLE nazwaRoli;

Rolę można też tworzyć starszą metodą

EXEC sp\_addrole nazwaRoli;

## Usuwanie roli:

DROP ROLE nazwaRoli;

Rolę można też usunąć starszą metodą

EXEC sp\_droprole nazwaRoli;

## Nadawanie uprawnień do roli

GRANT PRIVELEGE

ON objectName TO roleName;

## Odbieranie uprawnień z roli

REVOKE PRIVELEGE

ON objectName TO roleName;

## Modyfikowanie ról (role użytkownika)

Modyfikowanie roli umożliwia polecenie ALTER ROLE. Może ono przyjąć trzy postacie (od wersji SQL Server 2012):

ALTER ROLE roleName ADD MEMBER userName;

ALTER ROLE roleName DROP MEMBER userName;

ALTER ROLE roleName WITH NAME = roleName1;

## Dodanie użytkownika do roli

Dodanie użytkownika do wybranej roli użytkownika (lub roli bazodanowej) w programie Management Studio umożliwia gałąź Security | Roles. Możemy także użyć zapytania SQL z procedurą składowaną sp\_addrolemember:

EXEC sp\_addrolemember ‘nazwaRoli’, ‘nazwaUżytkownika’;

## Usunięcie użytkownika z roli

Usunięcie użytkownika z wybranej roli użytkownika (lub roli bazodanowej) w programie Management Studio umożliwia gałąź Security | Roles. Możemy także użyć zapytania SQL z procedurą składowaną sp\_droprolemember:

EXEC sp\_droprolemember ‘nazwaRoli’, ‘nazwaUżytkownika’;

# Role aplikacyjne

Nie można do nich przypisywać użytkowników. Służą one do tego, aby aplikacja korzystająca z bazy danych mogła wykonać specyficzną dla niej operację, do której nie ma uprawnień użytkownik pracujący z aplikacją. Przy tworzeniu roli tego typu podajemy login i hasło. Po standardowym połączeniu z bazą danych przez użytkownika, może on przełączyć kontekst na rolę aplikacyjną używając procedury sp\_setapprole i podając hasło. Role aplikacyjne znajdziem w Management Studio po rozwinięciu gałęzi Security | Roles | Application Roles w wybranej bazie danych.