Administrowanie bazami danych

# Zadania administratora bazy danych

Administrator bazy danych to osoba odpowiedzialna za tworzenie i instalowanie baz na serwerze i za zarządzanie bazami. Zarządzanie bazą danych obejmuje tworzenie kopii zapasowych danych i archiwizowanie danych z bazy.

Administrator zobowiązany jest do tworzenia odpowiednich zabezpieczeń bazy, aby w jak największym stopniu wyeliminować sytuacje jej uszkodzenia lub dostępu do informacji przez osoby niepowołane. W tym celu wdraża procedury:

* Uwierzytelniania użytkownika;
* Autoryzacji użytkownika
* Zarządzania dostępem do danych
* Replikacji bazy danych
* Tworzenia kopii bazy danych
* Archiwizowania danych z bazy

Administrator zajmuje się także porządkowaniem danych za pomocą określonych procedur i programów sortujących oraz zarządzaniem przechowywaniem danych archiwalnych.

# Podstawowe zadania konfiguracyjne administratora

Po instalacji serwera SQL, należy wykonać kilka zadań konfiguracyjnych. Zadania administratora w tym względzie to:

* Przegląd zainstalowanych usług i ich stanu
* Przegląd opcji konfiguracyjnych
* Rejestracja serwera
* Utworzenie grup serwerów
* Określenie trybu uwierzytelnienia
* Utworzenie hasła dla konta Server Administrator (sa)

## Usługi serwera

Po zainstalowaniu serwera SQL mamy możliwość uruchamiania lub zatrzymywania kilku usług serwera. Ich zestaw usług zależy od opcji zaznaczonych w trakcie instalacji. Do najważniejszych należą:

* SQL Server – podstawowa usługa MS SQL Server. Jej zatrzymanie spowoduje, że instancja serwera bazy danych nie będzie działała i odpowiadała na żądania użytkowników
* SQL Server Agent – usługa niezbędna do tego, aby serwer mógł automatycznie uruchamiać zaplanowane na określony termin zadania. Jeżeli nie definiujemy tego typu zadań, włączanie tej usługi nie jest konieczne
* SQL Server Browser – jest usługą dostarczającą klientom szczegółowych informacji o połączeniu z serwerem
* SQL Reporting Services – usługa udostępniająca pełen zakres gotowych do użycia narzędzi ułatwiających tworzenie, wdrażanie i zarządzanie raportami
* SQL Full-text Filter Daemon Launcher – usługa wyszukiwania pełno tekstowego

## Uruchamianie i zatrzymywanie usług serwera

Poprzez okno programu SQL Server Configuration Manager, każdą usługę serwera SQL można zatrzymać, uruchomić lub po prostu zrestartować za pomocą polecenia Restart dostępnego w menu podręcznym. Usługi można także uruchamiać lub zatrzymywać za pomocą narzędzia Usługi dostępneo w programie Narzędzia Administracyjne. Główną usługę serwera, czyli usługę SQL Server, a tym samym cały serwer można uruchomić za pomocą polecenia w wierszu poleceń systemu Windows (uruchomionego jako administrator):

NET START MSSQL$NazwaInstancjiSerwera

Z kolei zatrzymanie serwera umożliwia polecenie:

NET STOP MSSQL$NazwaInstancjiSerwera

## Przegląd opcji konfiguracyjnych serwera

## Rejestrowanie serwera

Aby wygodnie zarządzać lokalnym lub zdalnym serwerem przy użyciu programu Management Studio, należy zarejestrować go i umieścić w grupie serwerów. Serwer można zarejestrować przed połączeniem lub podczas połączenia z nim.

## Tworzenie grup serwerów

Po zarejestrowaniu serwera można go umieścić w jedne z istniejących grup serwerów lub utworzyć nową grupę. Grupa serwerów jest sposobem ich organizowania w celu realizacji zadać administratorskich.

## Tryb uwierzytelnienia serwera

Jak już wiemy, podczas instalacji można określić tryb uwierzytelnienia na serwerze SQL. Istnieją dwa możliwe tryby:

* Windows Authentication – ten tryb (nazywany także zaufanym połączniem) akceptuje uwierzytelnienie systemu Windows i wyłącza uwierzytelnienie przez SQL Serwer. Jeżeli użytkownik uwierzyelniany w ten sposób należy do grupy administratorów lokalnych w systemie Windows, to autoamtyznie po zalogowaniu zostaje przypisany do roli (grupy) sysadmin. Ten tryb jest domyślny i nie może zostać wyłączony
* Mixed Mode – pozwala na tworzenie kont na serwerze i na Windows Authentication

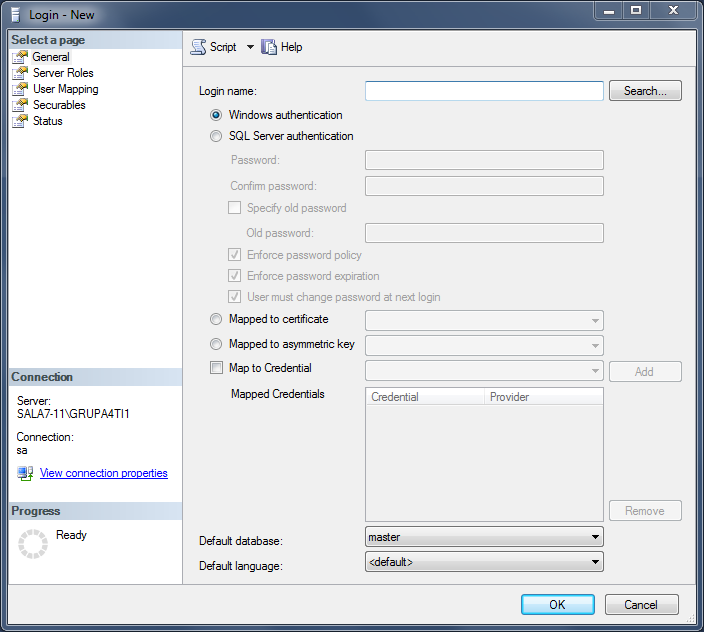
## Sprawdzanie trybu uwierzytelnienia

* Przez Właściwości serwera (zakładka Security)
* Poprzez zapytanie SQL i procedurę rozszerzoną SQL Servera:

EXEC xp\_loginconfig ‘login mode’;

# Tworzenie kont logowania

## Tworzenie kont logowania z poziomu Management Studio



## Tworzenie kont logowania z poziomu kodu SQL

CREATE LOGIN userName

FROM WINDOWS /\*or\*/ WITH PASSWORD = 'password';

# Tworzenie użytkownika bazy danych

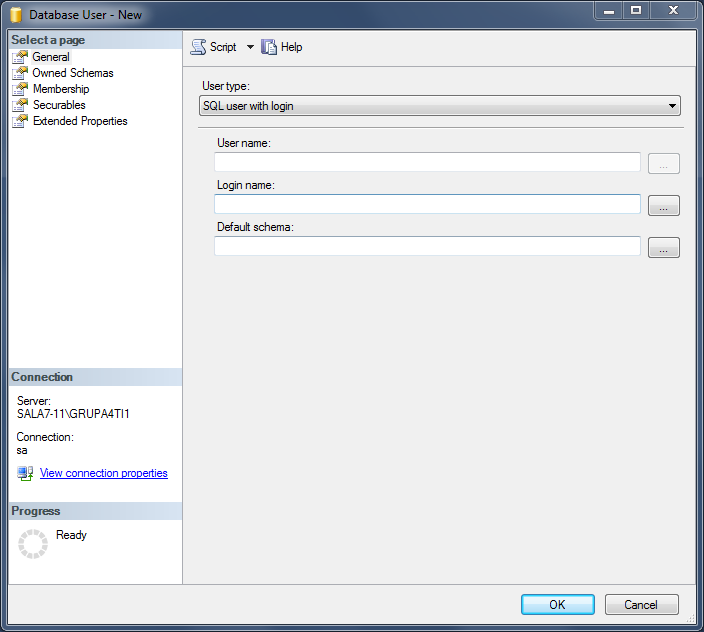
Gdy mamy już utworzone konto logowania, możemy go użyć do założenia użytkownika w poszczególnych bazach danych.

Konto logowania zakładamy na poziomie serwera, a użytkownika na poziomie bazy danych (każdej osobno!). Konto logowania pozwala połączyć się z serwerem, lecz dopiero użytkownik może mieć nadane prawa w poszczególnych bazach danych.

Dane konto logowania może mieć prawo dostępu w jednej bazie, a w innej nie.

Użytkownik może się nazywać inaczej niż konto logowania z nim związane. W dodatku użytkownicy stworzeni na bazie jednego konta logowania mogą inaczej się nazywać w różnych bazach danych.

## Tworzenie użytkownika z poziomu Management Studio



## Tworzenie użytkownika z poziomu kodu SQL

Utworzenie użytkownika w wybranej bazie danych i powiązanie go z wybranym kontem logowania umożliwia polecenie SQL:

USE databaseName;

CREATE USER userName

FOR LOGIN userAccountName;

# Uprawnienia (prawa dostępu)

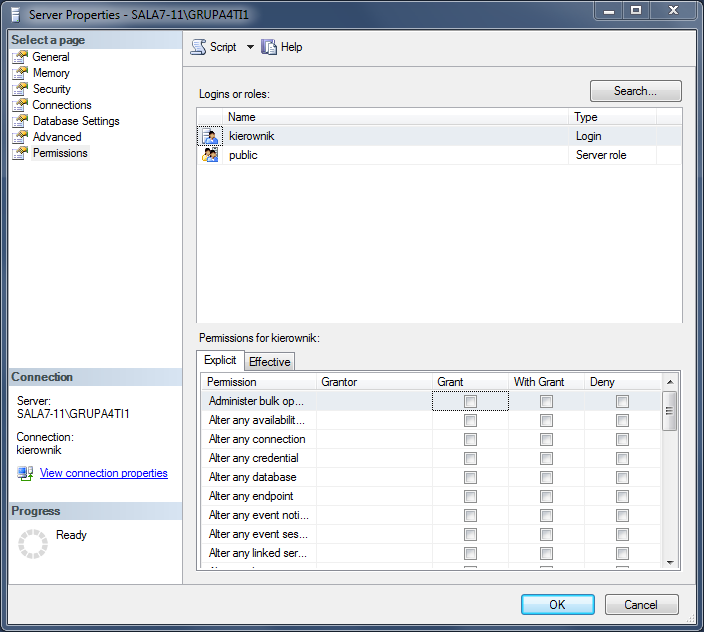
Uprawnienia możemy nadawać na różnych poziomach:

* Dla konta logowania możemy je przydzielać jedynie na poziomie serwera
* Dla użytkownika możemy je przydzielać na poziomie bazy danych, do obiektów, dla roli oraz do schematu

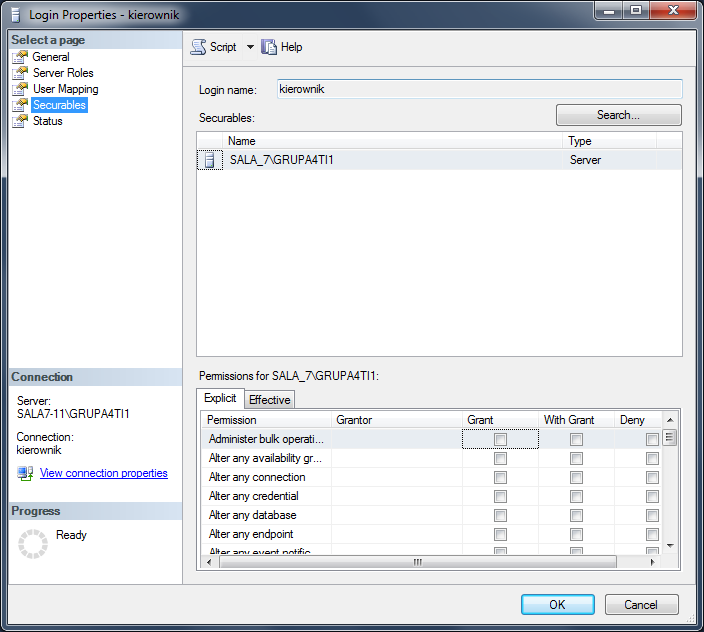
## Uprawnienia na poziomie serwera

Uprawnienia na poziomie serwera są jedynymi, które nadajemy dla konta logowania, a nie dla użytkownika. Są to prawa administracyjne, pozwalające wykonać określone czynności na instancji serwera bazy danych. Do uprawnień tych możemy się dostać na dwa sposoby:

* Wchodząc we właściwości instancji i wybierając zakładkę Permissions:



* Edytując właściwości konta logowania (zakładka Securables)



Przykłady uprawnień na poziomie serwera to:

* ALTER ANY DATABASE – pozwala na modyfikowanie ustawień dowolnej bazy danych. Umożliwia też tworzenie i usuwanie baz
* ALTER LOGIN – pozwala modyfikować i zakładać konta logowania
* CONNECT SQL – pozwala na połączenie z serwerem
* CREATE ANY DATABASE – pozwala zakładać bazy danych

Przykład nadania tego typu uprawnień z poziomu SQL:

GRANT UPRAWNIENIE TO nazwaKontaLogowania;

## Uprawnienia na poziomie bazy danych

Niektóre z uprawnień na poziomie bazy danych to:

* ALTER ANY USER – tworzenie i modyfikowanie użytkowników
* ALTER – tworzenie i modyfikowanie dowolnych obiektów
* BACKUP DATABASE – wykonywanie kopii zapasowej
* BACKUP LOG – wykonywanie kopii zapasowej dziennika transakcji
* CREATE TABLE – tworzenie tabel
* CREATE VIEW – tworzenie widoków
* SELECT – wykonywanie instrukcji SELECT na wszystkich tabelach bazy
* EXECUTE – wykonywanie wszystkich procedur składowanych w bazie
* INSERT – wykonywanie instrukcji INSERT na wszystkich tabelach bazy
* UPDATE – wykonywanie instrukcji UPDATE na wszystkich tabelach bazy
* DELETE – wykonywanie instrukcji DELETE na wszystkich tabelach bazy
* SHOW PLAN – wyświetlanie planu wykonania zapytania

Przykład nadania tego typu uprawnień z poziomu SQL:

GRANT UPRAWNIENIE TO nazwaKonta;

Uprawnienia te można też ustawić wchodząc we właściwości bazy danych i wybierając zakładkę Permissions.

## Uprawnienia na poziomie obiektów

Najważniejsze z uprawnień na poziomie obiektów to:

* SELECT – dostępne dla tabel i widoków, pozwala wykonać instrukcję SELECT. Możliwe jest nadanie prawa do całości lub do wybranych kolumn
* INSERT – dostępne dla tabel i perspektyw, pozwala wstawiać wiersze
* UPDATE – dostępne dla tabel i widoków lub ich podzbioru kolumn, pozwala modufikować rekordy
* DELETE – dostępne dla tabel i widoków
* EXECUTE – dostępne dla procedur
* ALTER
* CONTROL
* VIEW DEFINITION

Przykłady nadania tego typu uprawnień z poziomu SQL:

GRANT PRIVELEGE

ON objectName TO userName;

## Nadawanie I odbieranie uprawnień

Istnieją trzy instrukcje służące do zarządzania uprawnieniami:

* GRANT – nadaje uprawnienia
* DENY – zabrania dostępu
* REVOKE – czyści prawo lub zakaz dostępu

Wynikowe uprawnienia użytkownika są sumą uprawnień GRANT ze wszystkich poziomów, tzn. jeżeli na przykład użytkownik nie ma prawa SELECT na poziomie tabeli, lecz ma to prawo na poziomie bazy danych, to będzie mógł wykonać instrukcję SELECT na tej tabeli.

Inaczej jest w przypadku instrukcji DENY.Ta instrukcja jest silniejsza niż GRANT, z czego wynika, że jeżeli użytkownik ma DENY na którymkolwiek poziomie, to choćby miał GRANT na kilku innych poziomach, to i tak nie będzie mógł wykonać pewnych operacji. Przykład: użytkownik ma GRANT na instrukcji na poziomie bazy Hurtownia, lecz ma DENY na SELECT na tabeli Klienci. Wtedy nie będzie mógł wykonać instrukcji SELECT na tabeli Klienci.

Użytkownik, który otrzymał uprawnienia, domyślnie nie może przekazać ich dalej, lecz możemy to zmienić, przydzielając mu uprawnienia wraz z klauzulą WITH GRANT OPTION. Polecenie nadającego uprawnienia do obiektu bazy danych:

GRANT PRIVELEGE

ON objectName TO userName

WITH GRANT OPTION;

Z kolei polecenie odbierające nadane uprawnienia

REVOKE PRIVELEGE

ON objectName TO userName;

Zaś polecenie blokujące dostęp do wybranego obiektu:

DENY PRIVELEGE

ON objectName TO userName;

## Konta specjalne

Na SQL Server mamy dwa konta specjalne:

* guest - konto, które istnieje w każdej bazie, lecz domyślnie nie ma żadnych uprawnień. Każde konto logowania może wykonywać operacje, do których ma prawo guest, nawet jeśli z tym kontem nie jest skojarzony użytkownika w bazie. Domyślnie konto gościa nie ma nawet prawa Connect, czylni nie łączy się z bazą;
* dbo – właściciel bazy danych. Ma pełnie uprawnienia w bazie danych i może przekazywać swoje uprawnienia innym. Każde konto logowania należące do roli sysadmin ma uprawnienia dbo w każdej bazie danych.

# Uprawnienia dziedziczone

To uprawnienia przeniesiona na obiekt z obiektu nadrzędnego. Uprawnienia dziedziczone ułatwiają zarządzanie uprawnieniami i zapewniają spójność uprawnień we wszystkich obiektach z wybranego serwera czy bazy danych.

Obiekty bazodanwe tworzą hierarchię, zgodnie z którą działają mechanizmy dziedziczenia. Najwyżej w hierarchii znajdują się obiekty serwera bazodanowego, poniżej są obiekty bazy danych, jeszcze niżej mamy obiekty schematów, a na samym dole hierarchii są kolumny tabel.

# Właściciel obiektu

Każdy obiekt w bazie danych ma swojego właściciela, a domyślnie właścicielem obiektu jest użytkownik, który go utworzył. Na przykład zalogowany na serwerze użytkownik kierownik tworzący w bazie Hurtownia widok o nazwie vCenyRowerow staje się automatycznie jego właścicielem:

CREATE VIEW vCenyRowerow AS

SELECT NazwaRoweru, CenaJednostkowa

FROM Rowery

Możliwe jest jednak jawne określenie właściciela obiektu. Wystarczy podczas tworzenia obiektu jego nazwę poprzedzić nazwą nowego właściciela:

CREATE VIEW Pracownik.vCenyRowerow AS

SELECT NazwaRoweru, CenaJednostkowa

FROM Rowery