Warszawska Wyższa Szkoła Informatyki

Bazy danych – Semestr 2
Zajęcia nr 2

Uprawnienia

UPRAWNIENIA

- W świecie, w którym wszyscy ludzie byliby idealni, samo tworzenie kopii zapasowych wystarczyłoby do zapewnienia bezpieczeństwa
- Niestety ludzie mogą zepsuć coś przez pomyłkę lub co gorsze świadomie zniszczyć, zmienić coś, bądź wykraść dane
- Dlatego obecnie ogranicza się dostęp do danych (bazy danych), dopuszczając tylko osoby do tego powołane
- Określona (wybrana, wyznaczona) osoba powinna mieć tylko takie uprawnienia jakie są wymagane do wykonywania przez nią pracy

PROBLEM

- Z bazy danych może korzystać wielu użytkowników, którzy mogą dysponować różnymi prawami dostępu do różnych obiektów w bazie danych
- Najgorszym rozwiązaniem jakie może istnieć to pełne zaufanie do osób korzystających z bazy (przydzielenie im pełnych praw)
- Ze względu na bezpieczeństwo danych zawartych w bazie, warto stosować zasadę ograniczonego zaufania do użytkownika
- Należy zastanowić się: jakie uprawnienia należy przydzielić poszczególnym użytkownikom lub grupom użytkowników
- Dobrze jest wiedzieć, w jaki sposób można nadawać, odmawiać i cofać prawa dostępu do danych (wybranych danych)
- W kolejnym kroku należy rozważyć, czy użytkownik powinien mieć dostęp do całego obiektu typu tabela lub widok, czy może wystarczy mu dostęp do poszczególnych kolumn

Weryfikacja uprawnień do BD

- Weryfikacja uprawnień do BD przebiega w dwóch etapach:
  - **W pierwszym** – sprawdzana jest tożsamość osoby łączącej się do BD
  - **W drugim** – sprawdzane są uprawnienia które dana osoba posiada i czy pozwalają one na wykonanie żądanej akcji
- Tak więc ***tożsamość***, ***uprawnienia*** i ***akcja*** są 3 elementami systemu uprawnień

Typy tożsamości

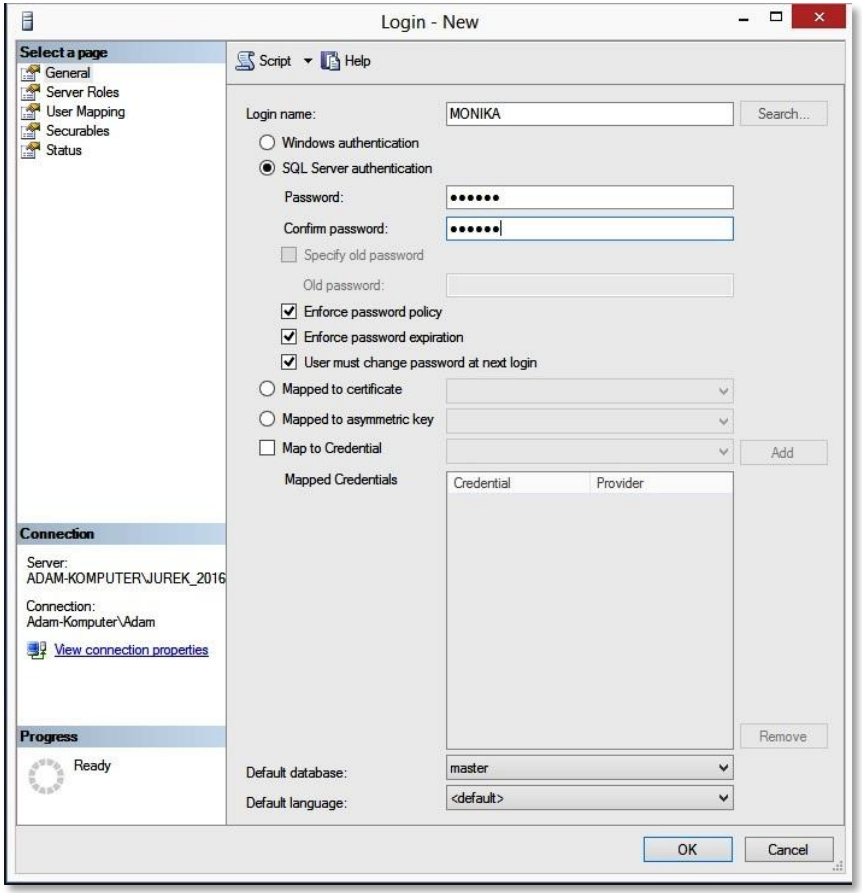
Poziom	Typ tożsamości
System operacyjny	**Konto lokalne** – konto założone na komputerze na którym znajduje się SQL Serwer **Konto domenowe** – konto założone w Active Directory **Grupa Windows** – konto może mieć dostęp do SQL Serwera poprzez przynależność do danej grupy
Instancja	**Konto SQL** – Konto może być uwierzytelniane hasłem może odpowiadać kontu lub grupie systemu operacyjnego
Baza danych	**Użytkownik bazy danych** – użytkownik odpowiadający kontu SQL **Rola bazy danych** – rola przypisana do użytkownika BD

Przykład – scenariusz z życia

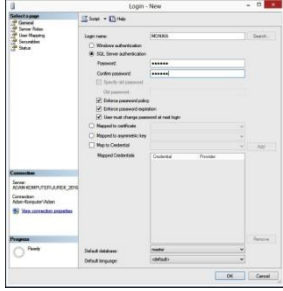
- Można utworzyć listę osób które mogą wejść do firmy
- Można stwierdzić też że:
  - ***wszyscy jej pracownicy*** - dostęp do wszystkich pomieszczeń firmy
  - ***oraz wszyscy goście*** – dostęp tylko do niektórych pomieszczeń
- Użycie **grup** i **ról** ułatwia znacznie administrację bezpieczeństwem
- Polecany model administracji to przydzielenie uprawnień

Jak stworzyć nowy login ?

- W ***SQL Management Studio*** klikamy węzeł **Databases | Security | Logins**. Klikamy prawym przyciskiem myszy i wybieramy **New Login**.

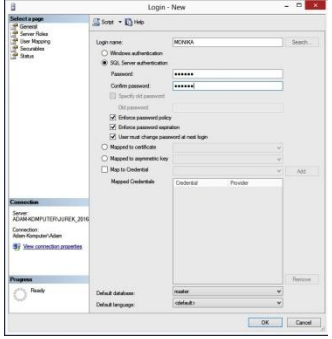


Jak stworzyć nowy login ?

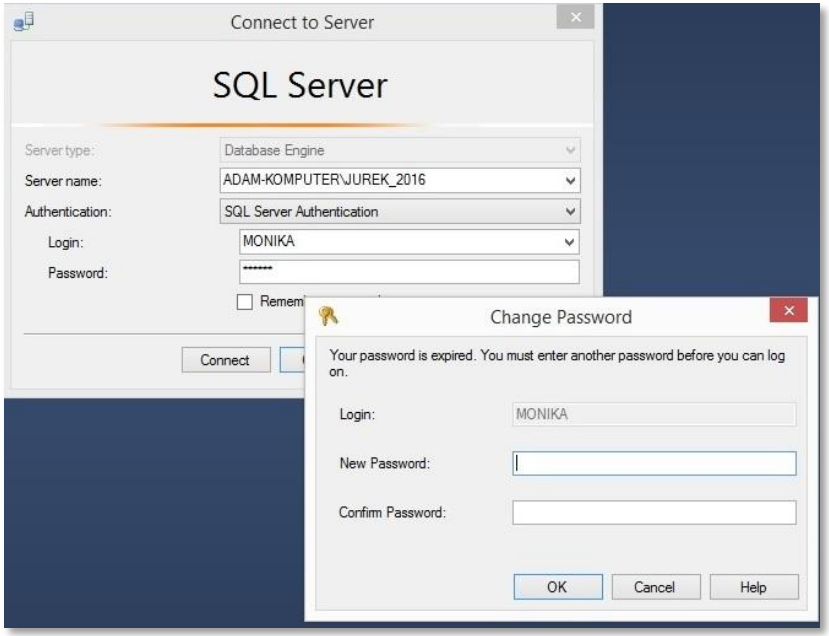


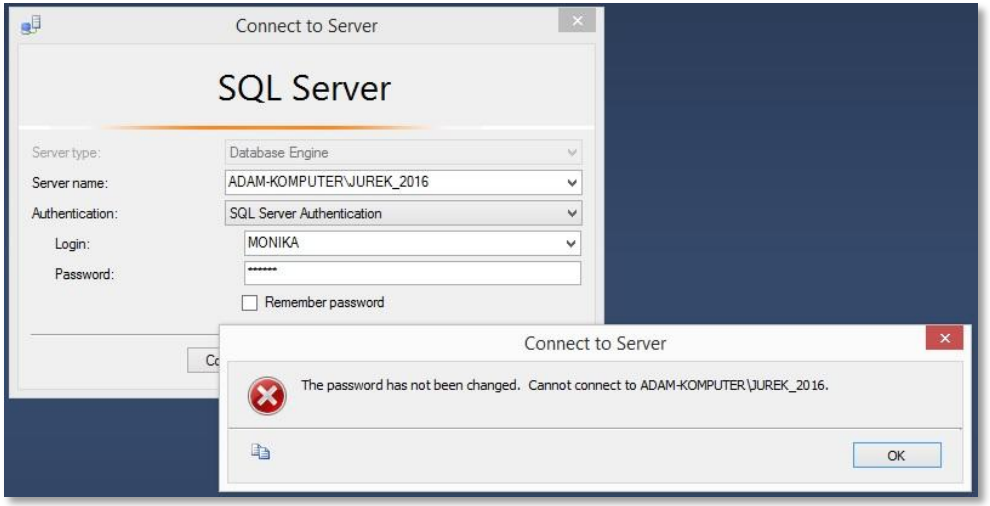
- Na głównej stronie określamy, czy będzie to użytkownik Windows (użytkownik Active Directory lub lokalne konto), czy będzie to użytkownik SQL.
  - **W celu wybrania autoryzacji Windows** –> zaznaczamy **Windows authentication**,
  - **w przypadku autoryzacji SQL** –> **SQL Server authentication**.
- W przypadku wybrania autoryzacji SQL, należy wprowadzić hasło dla loginu oraz poniżej wybrać jedną lub kilka z trzech opcji:
  - **Enforce password Policy** – hasło nie może zawierać w sobie części nazwy loginu i nie może być krótsze niż 7 znaków oraz powinno zawierać cyfry, duże i małe litery oraz znaki nie alfanumeryczne. Informacje pobierane są z polityki grupowej.
  - **Enforce password expiration** – hasło wygasa po przekroczeniu wartości określonej w polityce grupowej.
  - **User must change password at next logon** – wymaga zmiany hasła użytkownika przy kolejnym logowaniu.
- Sekcje **Default Database** oraz **Default language** są widoczne zarówno dla loginu z autoryzacją SQL jak i Windows. W **Default Database** określamy domyślną bazę danych dla loginu.

Jak stworzyć nowy login ?



User must change password at next logon – wymaga zmiany hasła użytkownika przy kolejnym logowaniu





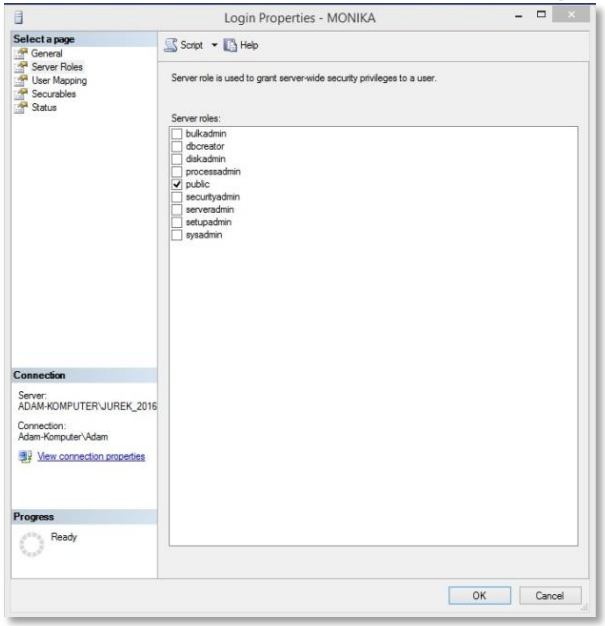
Podsumowanie: Typy tożsamości

- Uwierzytelnienie w trybie Windows
  - administrator systemu przydziela prawa dostępu poszczególnym kontom użytkowników WINDOWS
- Uwierzytelnianie w trybie SQL Serwer
  - dodatkowe uwierzytelnienie na poziomie SQL Serwera

Role serwerowe

Na stronie **Server Roles** wybieramy role serwerowe, przypisane dla tego loginu:

- **bulkadmin** – zezwala na operację masowego wstawiania danych (BULK INSERT)
- **dbcreator** – zezwala na tworzenie, usuwanie, modyfikację bazy danych oraz dodawanie do niej nowych członków (CREATE DATABASE)
- **diskadmin** – zezwala na zarządzanie plikami .mdf i .ldf (ALTER)
- **processadmin** – zezwala na kontrolę procesów (ALTER ANY CONNECTION oraz ALTER SERVER STATE)
- **securityadmin** – zezwala na zarządzanie loginami i uprawnieniami (ALTER ANY LOGIN)
- **serveradmin** – zezwala na konfigurację całego serwera (ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN)
- **setupadmin** –zezwala na zarządzanie serwerami połączonymi (ALTER ANY LINKED SERVER)
- **sysadmin** – zezwala na pełną kontrolę nad bazami danych (CONTROL SERVER with GRANT)



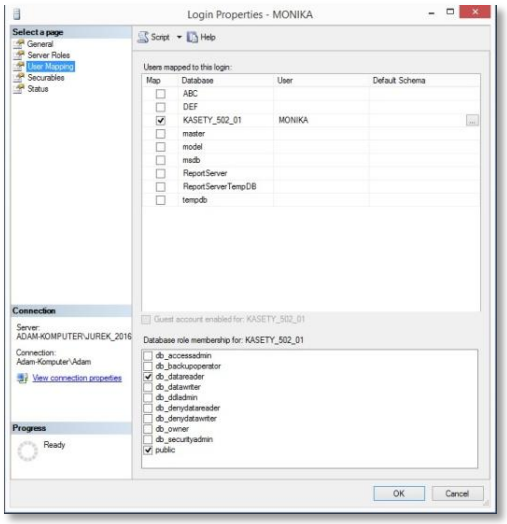
Podsumowanie: Role systemowe (serwerowe)

W przypadku instancji można wyróżnić role systemowe:

Nazwa roli	Pełna nazwa	Opis
BulkAdmin	Administratorzy masowego wstawiania	Mogą przeprowadzać operacje dotyczące dużej ilości danych i plików wstawiania (do masowego ładowania danych za pomocą BULK INSERT)
DbCreator	Twórcy bazy danych	Mogą tworzyć, zmieniać, lub usuwać bazę danych (polecenia CREATE, ALTER, DROP i RESTORE DATABASE
DiskAdmin	Administratorzy dysków	Mogą zarządzać plikami dyskowymi (wielkość, położenie plików danych i dzienników)
ProcessAdmin	Administratorzy procesów	Mogą zarządzać procesami działającymi w SQL Serwerze (pozwala zabijać procesy uruchomione wewnątrz SQL Serwera)
SecurityAdmin	Administratorzy systemów bezpieczeństwa	Rola umożliwia zarządzać loginami i ich właściwościami oraz uprawnieniami CREATE DATABASE. Członkowie tej grupy mogą również zmieniać hasła dla użytkowników
SerwerAdmin	Administratorzy serwera	Umożliwia rekonfigurację (wszędzie w serwerze mają prawo ustawiać opcje konfiguracyjne) oraz mogą zamykać instancję (serwer)
SetupAdmin	Administratorzy ustawień	Mogą zarządzać serwerami dołączonymi (pozwala dodawać i usuwać) , procedurami uruchamiającymi
SysAdmin	Administratorzy systemu	Mogą przeprowadzać dowolne działania w SQL Serwerze wewnątrz instancji.

Role w bazach danych (User Mapping)

- Na stronie **User Mapping** zaznaczamy, do której bazy użytkownik będzie posiadał uprawnienia (**Users mapped to this login**),
- następnie w sekcji **Database role membership for** : nazwa_bazy zaznaczamy role, które chcemy nadać:
  - **db_accessadmin** – zezwala na dodawanie i usuwanie kont,
  - **db_backupoperator** – zezwala na wykonywanie kopii zapasowych,
  - **db_datareader** – zezwala na odczyt baz danych,
  - **db_datawriter** – zezwala na zapisywanie i modyfikację baz danych,
  - **db_ddladmin** – zezwala na modyfikację i usuwanie obiektów baz danych,
  - **db_denydatareader** – nie zezwala na odczyt baz danych,
  - **db_denydatawriter** – nie zezwala na zapisywanie i modyfikację baz danych,
  - **db_owner** – zezwala na pełną kontrolę nad bazą danych,
  - **db_securityadmin** – zezwala na zarządzanie uprawnieniami oraz rolami baz danych,
  - **public** – rola domyślna, zapewniająca minimum uprawnień.



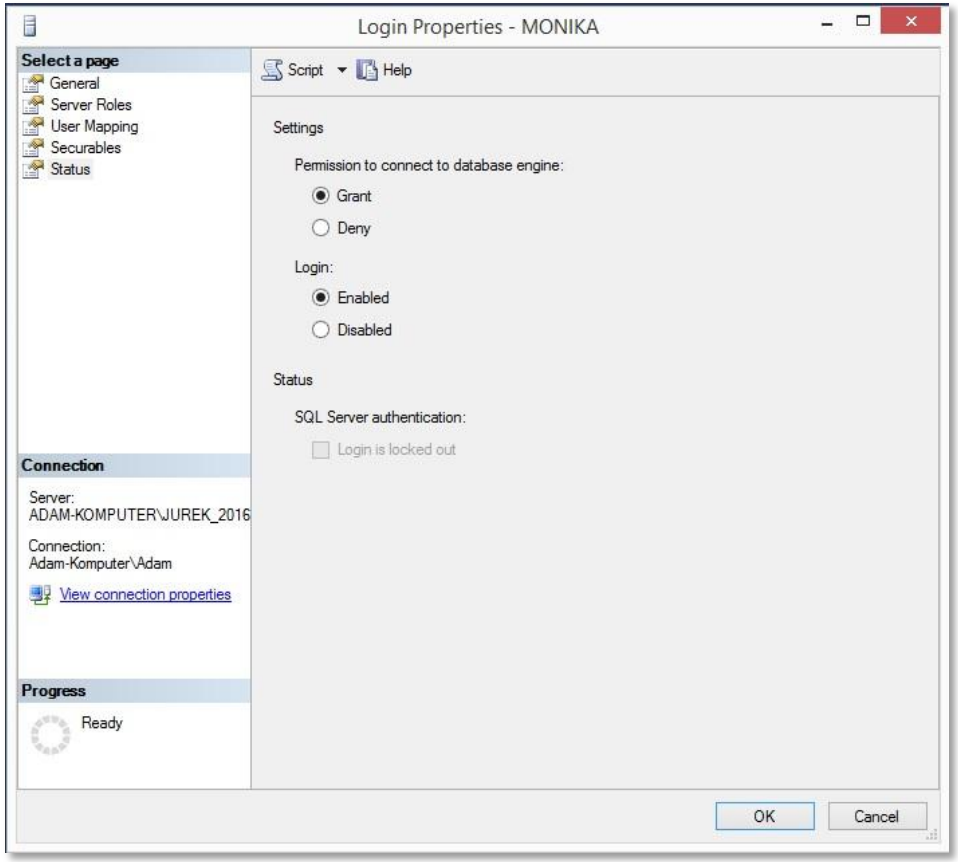
Podsumowanie: Wbudowane role bazodanowe

Nazwa roli	Pełna nazwa	Opis
Db_accessadmin	Administrator dostępu	Pozwala na dodawanie i usuwanie dostępu do bazy danych dla poszczególnych loginów
Db_backupoperator	Operator kopii zapasowej	Pozwala wykonywać kopie zapasowe bazy danych (BACKUP)
Db_datareader	Czytający dane	Pozwala na czytanie danych z dowolnego użytkownika. Za pomocą instrukcji SELECT może wybierać wszystkie dane z tabeli każdego użytkownika w bazie danych
Db_datawriter	Wpisujący dane	Pozwala na modyfikację (zapisywanie, usuwanie, modyfikację) dowolnych danych w tabeli każdego użytkownika w bazie danych (nie systemowych)
Db_denydatareader	Odmowa czytania danych	Uniemożliwia czytanie danych z tabel użytkownika. Nie ma prawa do wykonywania instrukcji SELECT na żadnym obiekcie bazy danych.
Db_denydatawriter	Odmowa pisania danych	Uniemożliwia zapis, usuwanie i modyfikację danych. Nie ma prawa do wykonywania instrukcji INSERT, UPDATE, DELETE na żadnym obiekcie bazy danych użytkownika
Db_dladmin	Administrator definiowania danych	Pozwala na tworzenie, modyfikację i usuwanie obiektów dowolnego typu wewnątrz bazy danych.
Db_owner	Właściciel bazy danych	Pozwala na wykonywanie dowolnej czynności administracyjnej wewnątrz bazy danych (również DROP DATABASE (brak w db_dladmin)
Db_securityadmin	Administrator systemu bezpieczeństwa	Pozwala zarządzać przynależnością do ról wewnątrz bazy danych zarządzać uprawnieniami
Public	Publiczna	Domyślna rola przypisywana każdemu użytkownikowi

Strona status

Strona **Status** służy do:

- nadawania lub odejmowania uprawnień dla loginu (użytkownika) do łączenia się z bazą danych (**Permission to connect to Database engine**)
- blokowania lub odblokowania konta



Skrypt do tworzenia powyższych operacji

W przypadku tworzenia LOGINU (jako użytkownika SQL) skrypt będzie wyglądal następująco.

```
USE MASTER
CREATE LOGIN MONIKA WITH PASSWORD='Pa$$w0rd' MUST_CHANGE, DEFAULT_DATABASE=[master],
CHECK_EXPIRATION=ON, CHECK_POLICY=ON
GO
USE MOJA_BAZA
CREATE USER MONIKA
GO
sp_addrolemember 'db_datareader', MONIKA
GO
sp_addrolemember 'db_datawriter', MONIKA
GO
sp_addrolemember 'db_owner', MONIKA
GO
```

W przypadku tworzenia LOGINU (jako użytkownika Windows), skrypt będzie wyglądal następująco.

```
USE [master]
GO
CREATE LOGIN [ORCA\pyszczek] FROM WINDOWS WITH DEFAULT_DATABASE=[master]
GO
EXEC master..sp_addsrvrolemember @loginame = 'ORCA\pyszczek', @rolename = 'sysadmin'
GO
USE [baza]
GO
CREATE USER [ORCA\pyszczek]
GO
USE baza
GO
sp_addrolemember 'db_datareader, 'ORCA\pyszczek'
GO
USE [baza]
GO
sp_addrolemember 'db_datawriter', 'ORCA\pyszczek'
GO
USE [baza]
GO
sp_addrolemember 'db_owner', 'ORCA\pyszczek'
GO
```

Tworzenie ról

- W SQL Serwerze 2014 nie ma możliwości tworzenia nowych ról serwerowych
- W SQL Serwerze 2014 można tworzyć nowe role bazodanowe (z poziomu interfejsu graficznego lub za pomocą języka T-SQL *Create role …)*
- Po utworzeniu roli zostają dwie czynności:
  - Przypisanie uprawnień do roli
  - Przydzielenie tożsamości do ról (z poziomu interfejsu graficznego lub za pomocą procedury *Exec sp_addrolemember …)*

Omawiane zagadnienia

- Tworzenie/usuwanie identyfikatora logowania SQL Serwera (użytkownika)
- Zarejestrowanie /usuwanie użytkownika bazy danych
- Tworzenie/usuwanie ról w bazie danych
- Przypisywanie użytkownikowi roli w bazie danych
- Odbieranie użytkownikowi roli z bazy danych

Tworzenie i usuwanie użytkownika

- **Tworzenie nowego użytkownika SQL SERWERA:**

  CREATE LOGIN ROBAK WITH PASSWORD ='123', CHECK_POLICY = OFF, CHECK_EXPIRATION = OFF
- **Tworzenie użytkownika WINDOWS w SQL Serwerze**

  CREATE LOGIN [ORCA\Z501_01] FROM WINDOWS;
- **Usuwanie użytkownika z SQL Serwera**

  DROP LOGIN ROBAK

  DROP LOGIN [ORCA\Z501_01]

Domyślnie SQL Server zawsze sprawdza zasadę haseł, ale można zawiesić jej egzekwowanie dla wybranych nazw użytkowników przy użyciu instrukcji CREATE LOGIN lub ALTER LOGIN, jak w poniższym fragmencie kodu:

CREATE LOGIN ADAM WITH PASSWORD = 'S%V7Vlv3c9Es8', CHECK_EXPIRATION = OFF, CHECK_POLICY = OFF

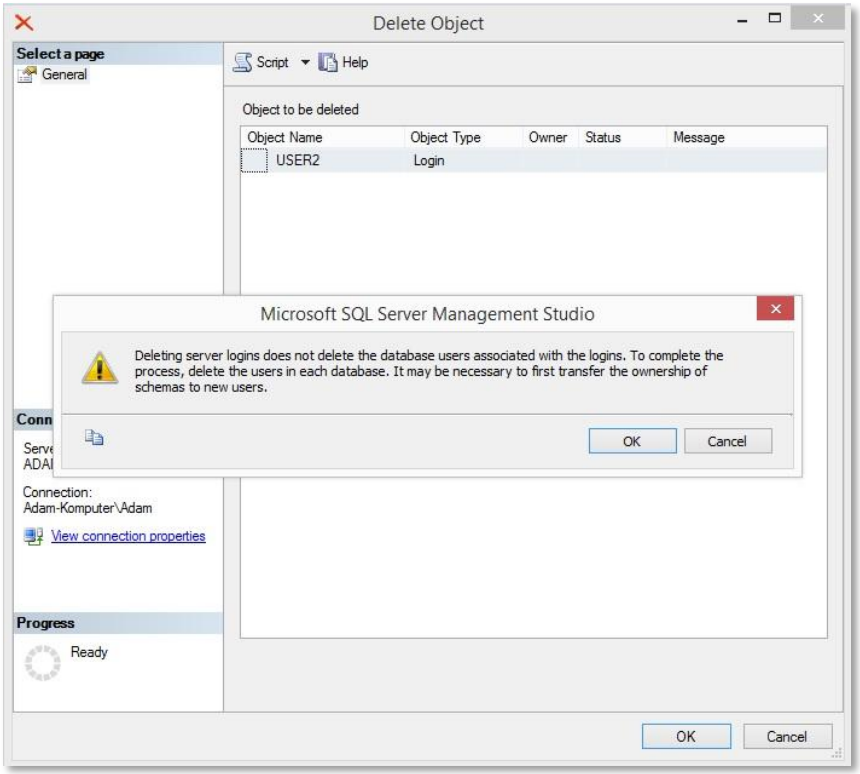
- **Enforce password Policy** – hasło nie może zawierać w sobie części nazwy loginu i nie może być krótsze niż 7 znaków oraz powinno zawierać cyfry, duże i małe litery oraz znaki nie alfanumeryczne. Informacje pobierane są z polityki grupowej.
- **Enforce password expiration** – hasło wygasa po przekroczeniu wartości określonej w polityce grupowej.

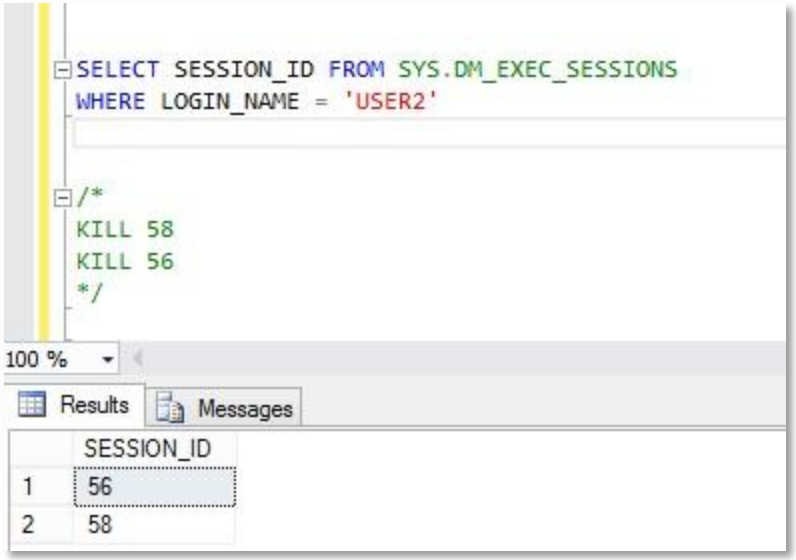
Usuwanie użytkownika

- **Usuwanie użytkownika z SQL Serwera**

  DROP LOGIN ROBAK

  DROP LOGIN [ORCA\Z501_01]





Dodawanie i odbieranie uprawnień

- **Dodanie uprawnień użytkownikowi ROBAK (ORCA/Z501_01) do bazy MOJA (z rolą PUBLIC)**

```
USE MOJA
GO
CREATE USER ROBAK
```

lub CREATE USER [ORCA\Z501_01]

- **Odebranie uprawnień użytkownikowi z bazy**

```
USE MOJA
GO
DROP USER ROBAK
```

lub DROP USER [ORCA\Z501_01]

Nadanie i odbieranie roli

- **Nadanie roli (db_owner) użytkownikowi ROBAK i GOŚĆ (Windows) do bazy MOJA**

```
USE MOJA
GO
SP_ADDROLEMEMBER ‘DB_OWNER’,’ROBAK’
```

Lub SP_ADDROLEMEMBER ‘DB_OWNER’, ‘ORCA\Z501_01’

- **Zdjęcie (odebranie roli)**

```
USE MOJA
GO
SP_DROPROLEMEMBER ‘DB_OWNER’,’ROBAK’
```

Lub SP_DROPROLEMEMBER ‘DB_OWNER’,’ORCA\Z501_01’

Tworzenie i usuwanie roli

- **Tworzenie nowej Roli**

```
USE MOJA
GO
CREATE ROLE PROFESOR
```

- **Usuwanie roli**

```
USE MOJA
GO
DROP ROLE PROFESOR
```

- **Nadanie nowej roli (PROFESOR) użytkownikowi ROBAK**

```
USE MOJA
GO
SP_ADDROLEMEMBER 'PROFESOR','ROBAK'
```

Nazwa roli	Pełna nazwa
Db_accessadmin	Administrator dostępu
Db_backupoperator	Operator kopii zapasowej
Db_datareader	Czytający dane
Db_datawriter	Wpisujący dane
Db_denydatareader	Odmowa czytania danych
Db_denydatawriter	Odmowa pisania danych
Db_dladmin	Administrator definiowania danych
Db_owner	Właściciel bazy danych
Db_securityadmin	Administrator systemu bezpieczeństwa
Public	Publiczna

Język DCL

- **Język SQL został opracowany w 1987 roku z myślą o relacyjnych bazach danych**
- **Składa się on z trzech składowych:**
  - **języka operowania na danych (DML)**
  - **języka definiowania danych (DDL)**
  - **języka sterowania danymi (DCL- *D*ata *C*ontrol *L*anguage)**

Język DCL

- **Instrukcje języka DCL służą do zarządzania uprawnieniami dostępu do obiektów bazy**
- **Najważniejszymi poleceniami języka DCL są instrukcje:**
  - **GRANT – (Nadanie)** pozwala użytkownikowi lub roli na wykonywanie operacji określonej przez nadane uprawnienie
  - **DENY – (Odmowa)** odmawia użytkownikowi lub roli określonego uprawnienia i zapobiega odziedziczeniu go po innych rolach
  - **REVOKE – (Cofnięcie)** usuwa uprzednio nadane lub odmówione uprawnienie

Język DCL

Czym różni się odmowa uprawnień od cofnięcia uprawnień?

- **Są dwa rodzaje uprawnień (podobnie jak w systemie Windows XP/2000/2003/7)**
  - **uprawnienia pozytywne (coś można zrobić)**
  - **Uprawnienia negatywne (coś jest zabronione)**
- **Do nadawania uprawnień pozytywnych służy polecenie GRANT**
- **Uprawnienia negatywne nadajemy poleceniem DENY**
- **Polecenie REVOKE pozwala cofnąć uprawnienia nadane poleceniami GRANT i DENY**

Ważne

- ***Uprawnienia negatywne (nadane poleceniem DENY) nadpisują uprawnienia pozytywne (nadane przez polecenie GRANT)***

Uprawnienia - składnia

GRANT – przyznanie uprawnień

- **GRANT <lista praw> ON <element bazy danych > TO <lista użytkowników>**

DENY – zabronienie uprawnień

- **DENY <lista praw> ON <element bazy danych > TO <lista użytkowników>**

REVOKE – odwołanie uprawnienia przyznanego wcześniej poleceniem GRANT czy DENY

- **REVOKE <lista praw> ON <element bazy danych > FROM <lista użytkowników>**

Uprawnienia

- Po zweryfikowaniu **tożsamości** kolejnym elementem jest sprawdzenie **uprawnień** jakie posiada dana tożsamość
- Przydzielanie i odwoływanie uprawnień odbywa się za pomocą trzech poleceń
  - **GRANT** – przyznanie uprawnień
  - **DENY** – zabronienie uprawnień
  - **REVOKE** – odwołanie uprawnienia przyznanego wcześniej poleceniem GRANT czy DENY
- *Wybrane uprawnienia na poziomie tabel*
  - ***SELECT*** *odczytanie, wybieranie wierszy tabel*
  - ***INSERT*** *wstawianie wierszy do tabel*
  - ***UPDATE*** *aktualizacja wierszy tabeli*
  - ***DELETE*** *usuwanie wierszy tabeli*
  - ***EXECUTE*** *– wykonywania procedury składowanej*
  - ***ALL*** *wszystkie przywileje*

Zapewnienie dostępu do tabel

- **Przykład: przyznawanie praw dostępu do tabeli OSOBY**

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON OSOBY
TO PROFESOR
```

- **Przykład: odbieranie uprawnień (ograniczenie praw dostępu do tabel) do tabeli OSOBY**

```
REVOKE SELECT,INSERT,UPDATE,DELETE
ON OSOBY
TO PROFESOR
```

- **SP_HELPROTECT**

  Komenda do przeglądania aktualnych uprawnień

Zapewnienie dostępu do pojedynczych kolumn

- Istnieje również możliwość przyznawania lub odmawiania praw dostępu do poszczególnych kolumn
- Uprawnienia dotyczące zarządzaniem kolumn tabeli:
  - **SELECT** Umożliwia wykonywanie selekcji na kolumnie
  - **UPDATE** Umożliwia aktualizowanie kolumny
  - **REFERENCE** Umożliwia odwoływanie się do kolumny z obcego klucza

Zapewnienie dostępu do pojedynczych kolumn

- **Przykład: przyznawanie praw dostępu do kolumn (nazwisko, imie) tabeli OSOBY**

```
GRANT SELECT, UPDATE (nazwisko,imie)
ON OSOBY
TO PROFESOR
```

- **Przykład: odbieranie uprawnień (ograniczenie praw dostępu do kolumn tabeli) tabeli OSOBY**

```
REVOKE UPDATE (imie)
ON OSOBY
TO PROFESOR
```

Podsumowanie – porady praktyczne (1/2)

- Przyznawanie lub odmawianie praw dostępu do poszczególnych kolumn zwiększa elastyczność w zarządzaniu dostępem na przykład do poufnych danych z niektórych kolumn
- Tworzenie użytkowników oraz nadawanie im uprawnień może odbywać się na dwa sposoby:
  - Pierwszy wymaga znajomości języka T-SQL.
  - Drugi sposób polega na wykorzystaniu graficznych narzędzi dostępnych w SQL Server Menagement Studio.
- W przypadku tworzenia użytkowników z poziomu języka T-SQL musimy znać odpowiednie procedury składowane, których należy w tym celu użyć
- W procedurach składowanych zapisane są ustawienia, które zostaną wprowadzone podczas ich użycia

Podsumowanie – porady praktyczne (2/2)

- Możemy nadawać uprawnienia do różnych obiektów w bazie danych. W przykładach pokazano w jaki sposób nadawać uprawnienia do obiektu typu tabela
- W analogiczny sposób możemy nadawać uprawnienia do obiekty typu: widok, funkcja czy procedura składowana
- Użytkowników możemy grupować według nadanych im uprawnień. Wówczas możemy założyć role i pogrupować użytkowników według ról, jakie pełnią w bazie danych
- W wyniku tego możemy przypisać dostęp do obiektów bazy danych nie tylko pojedynczemu użytkownikowi, ale również grupom użytkowników zapisanych w roli.

Przykład: Skrypt 1 - Tworzenia tabel bazy danych

```
CREATE DATABASE MOJA
-- W waszym przypadku baza szkolna jest już założona
 GO
-- ------------------------------------------------------
USE MOJA  (u was np. KASETY_501_01)
go

-- ****** Tworzenie Tabeli: OSOBY
-- ------------------------------------------------------
CREATE TABLE OSOBY
      (
      IDO                   int             PRIMARY KEY,
      NAZWISKO              char (30),
      IMIE                  char (15)
      )
GO

-- ****** Tworzenie Tabeli: HOBBY
-- ------------------------------------------------------
CREATE TABLE HOBBY
      (
      IDH                   int             PRIMARY KEY,
      NAZWA                 char (15)
      )
GO

-- ****** Tworzenie Tabeli: O-H
-- ------------------------------------------------------
CREATE TABLE O_H
      (
      IDO                   int,
      IDH                   int
      )
GO
```

OSOBY
IDO
NAZWISKO
IMIE

O-H
IDO
IDH

HOBBY
IDH
NAZWA

Przykład: Skrypt 2 - Usuwanie tabel bazy danych

```
USE MOJA
-- (u was np. USE KASETY_501_01)
GO
-- ****** Usuwanie Tabeli: O_H
-- ------------------------------------------------
DROP TABLE O_H
GO
-- ****** Usuwanie Tabeli: OSOBY
-- ------------------------------------------------
DROP TABLE OSOBY
GO
-- ****** Usuwanie Tabeli: HOBBY
-- ------------------------------------------------
DROP TABLE HOBBY
GO
```

Wprowadzenie przykładowych danych do tabel

```
-- ****** - Do tabeli OSOBY
-- ----------------------------------
INSERT INTO OSOBY VALUES (1,'JAN','NOWAK')
INSERT INTO OSOBY VALUES (2,'ANNA','BĄK')
----------------------------------
-- ****** - Do tabeli HOBBY
----------------------------------
INSERT INTO HOBBY VALUES (1,'Piłka Nożna')
INSERT INTO HOBBY VALUES (2,'Boks')
----------------------------------
-- ****** - Do tabeli O_H
----------------------------------
INSERT INTO O_H VALUES (1,1)
INSERT INTO O_H VALUES (2,1)
INSERT INTO O_H VALUES (1,2)
-- ----------------------------------
```