Administracja i programowanie pod Microsoft SQL Server 2000

Paweł Rajba

pawel@ii.uni.wroc.pl
http://www.kursy24.eu/

Zawartość modułu 12

- Bezpieczeństwo SQL Servera
  - Tryby uwierzytelniania
  - Zarządzanie kontami
  - Regulacja dostępu do baz danych
  - Domyślne konta serwerowe
  - Konta bazy danych
  - Role, role serwera, role baz danych, role aplikacji
  - Uprawnienia
  - Zarządzanie bezpieczeństwem

Tryby uwierzytelniania

- Windows
  - Autentykacja możliwa jest tylko na podstawie kont systemu Windows
  - Przebieg autentykacji
    - klient otwiera połączenie do SQLServera i przesyła informacje identyfikujące go w systemie Windows (credentials)
    - SQLServer sprawdza, czy konto jest na liście kont w tabeli sysxlogins. Jeśli tak, akceptuje połączenie.
  - Uwaga na usunięcie i ponowne utworzenie konta w Windows
  - Wbudowane konto ma grupa Administratorzy

Tryby uwierzytelniania

- Mieszane
  - Autentykacja możliwa jest tylko na podstawie kont systemu Windows i kont SQLServera
  - Przebieg autentykacji
    - SQLServer sprawdza na podstawie tabeli sysxlogins, czy klient używa poprawnego identyfikatora konta SQLServer
    - Jeśli tak, sprawdza hasło i akceptuje połączenie lub nie
    - Jeśli nie, SQLServer sprawdza, czy konto jest na liście kont w tabeli sysxlogins. Jeśli tak, akceptuje połączenie.
  - Wbudowane konta SQLServera: sa, guest i dbo

Tryby uwierzytelniania

- Zalety uwierzytelniania Windows
  - pozwala na lepsze zarządzanie hasłem: wygaśnięcie, minimalna długość, itp.
  - łatwiej zarządzać kontami, np. dodając login dajemy hasło do grupy serwerów, lub dodając grupę dajemy dostęp grupie użytkowników
  - wygodniejsze dla użytkownika – ma tylko jedno hasło do wszystkiego
- Zalety uwierzytelniania mieszanego
  - pozwala na autoryzację użytkowników spoza systemu Windows, np. użytkowników z Internetu

Zarządzanie kontami

- Wprowadzenie
  - Konta są przechowywane w tabeli master.sysxlogins
  - Z kontem jest związana domyślna baza danych
  - Utworzenie konta nie daje jeszcze praw do dostępu do baz danych
  - Nazwa konta lub grupy może mieć co najwyżej 128 znaków
  - Przy zmianie członków w grupie Windows, w SQL Serverze nie ma potrzeby niczego aktualizować – konto grupy jest w SQL-u pojedynczym kontem

Zarządzanie kontami

- Wprowadzenie
  - Usunięcie konta w Windows nie powoduje usunięcia konta w SQL Serverze
    - jeśli chcemy usunąć konto definitywnie, najpierw powinniśmy to zrobić w Windowsie, a dopiero potem w SQL Serverze
  - Funkcja SUSER_SNAME() pobiera nazwę konta aktualnie zalogowanego użytkownika
  - Konta Windows są dostępne w SQL Serverze w postaci DOMAIN\UserName
  - Zarządzać kontami mogą członkowie ról:
    - sysadmin i securityadmin

Zarządzanie kontami

- Tworzenie konta w Enterprise Manager
  - (local) | Security | Logins
  - prawy p. myszy, New Login
- Blokada konta (tylko konto typu Windows)
  - Enterprise Manager
    - (local) | Security | Logins
    - prawy p. myszy na koncie, Właściwości
    - Grant access / Deny access

Zarządzanie kontami

- Tworzenie konta za pomocą T-SQL
  - konto typu Windows
    - sp_grantlogin [@loginame =] '*login*‘
    - sp_denylogin [@loginame =] '*login*'
    - powyższe procedury służą również do blokowania/odblokowania konta
  - konto typu SQL Server
    - sp_addlogin [@loginame =] '*login*'
      [, [@passwd =] '*password*']
      [, [@defdb =] '*database*']
      [, [@deflanguage =] '*language*']
      [, [@sid =] *sid*]
      [, [@encryptopt =] '*encryption_option*']

Zarządzanie kontami

- Przeglądanie kont
  - Enterprise Manager
    - (local) | Security | Logins
  - T-SQL
    - sp_helplogins [[@LoginNamePattern =] '*login*']

Zarządzanie kontami

- Usuwanie konta
  - Enterprise Manager
    - (local) | Security | Logins
    - prawy p. myszy na koncie, Usuń i potwierdzenie
  - T-SQL
    - sp_revokelogin [@loginame =] '*login*'
      - usunięcie konta z Windows
    - sp_droplogin [@loginame =] '*login*'
      - usunięcie konta z SQL Servera

Zarządzanie kontami

- Zmiana hasła dla konta SQL Servera
  - Enterprise Manager
    - (local) | Security | Logins
    - prawy p. myszy na koncie, Właściwości
  - T-SQL
    - sp_password [[@old =] '*old_password*' ,]
      { [@new =] '*new_password*' }
      [, [@loginame =] '*login*']
    - sp_password @new='haselko', @loginame='pawel'

Regulacja dostępu do baz danych

- Przypisywanie kontu dostęp do bazy danych
  - Enterprise Manager
    - (local) | Databases | <baza danych> | Users
    - prawy p. myszy, New Database User
  - T-SQL
    - sp_grantdbaccess [@loginame =] 'login'
      [,[@name_in_db =] 'name_in_db']

Regulacja dostępu do baz danych

- Odbieranie kontu dostęp do bazy danych
  - Enterprise Manager
    - (local) | Databases | <baza danych> | Users
    - prawy p. myszy na userze, Usuń
  - T-SQL
    - sp_revokedbaccess [@name_in_db =] 'name‘
      – można to robić tylko w bieżącej bazie

Regulacja dostępu do baz danych

- Dane o kontach bazy danych są w tabelach
  - sysusers – dane o kontach
  - sysprotects – dane o uprawnieniach
- Regulację dostępu do baz danych mogą robić:
  - członkowie roli sysadmin
  - właściciel bazy (database owner) – rola db_owner
  - administrator dostępu do bazy (database access administrators) – rola db_accessadmin and

Przykład 1

- Demonstracja

Domyślne konta serwerowe

- Grupa Administratorzy – BUILTIN\Administrators
  - Domyślnie grupa ta dostaje rolę serwerową System Administrators (sysadmin)
- Konto administratora SQLServera – sa
  - Domyślnie konto to dostaje rolę serwerową System Administrators (sysadmin)
  - Dostępne tylko w autentykacji typu mieszanego
    - należy się wtedy upewnić, że jego hasło nie jest puste
  - Jeśli konto nie jest niezbędne zaleca się je zablokować (gdyż jest ogólnie znane)

Konta bazy danych

- Konto dbo
  - Specjalne konto oznaczające właściciela bazy danych (Database Owner)
  - Członkowie roli sysadmin i login sa są automatycznie mapowani na użytkownika dbo
  - Właścicieli posiadają obiekty utworzone w bazach danych SQL Server
  - Przykład
    - yogi jest w sysadmin: jak utworzy tabelę, to jej właścicielem będzie dbo
    - yogi nie jest w sysadmin: właścicielem będzie yogi

Konta bazy danych

- Konto guest
  - Daje dostęp do bazy danych wszystkim użytkownikom posiadającym konto w SQL Server
  - Użytkownik będzie miał tożsamość i prawa tego konta, gdy zajdą następujące warunki:
    - login ma dostęp do SQLServera, ale nie ma dostępu do danej bazy danych
    - w bazie jest utworzone konto guest
  - Domyślnie konto to nie jest tworzone
  - W bazach master i tempdb konto guest jest i nie można go usunąć

Konta bazy danych

- Konto guest c.d.
  - Konto guest jest członkiem roli public i po niej dziedziczy uprawnienia
  - Przykład:
    - w bazie example pawel nie ma konta
    - logujemy się jako pawel; raz konto guest w bazie example jest usunięte, a raz dostępne

Role

- W SQLServerze mamy następujące rodzaje ról:
  - Role serwerowe (Fixed Server Roles)
  - Role bazy danych (Fixed Database Roles)
  - Role bazy danych użytkownika (User-defined Database Roles)

Role serwera

- Cechy ról serwera (Fixed Server Roles)
  - nie można ich modyfikować
  - każdy użytkownik danej roli może do tej roli dodawać następnych użytkowników
- Lista ról serwera
  - sysadmin
    - Może wykonywać dowolne akcje
  - dbcreator
    - Tworzy i modyfikuje bazy danych
  - diskadmin
    - Zarządza plikami na dysku

Role serwera

- Lista ról serwera c.d.
  - processadmin
    - Zarządza procesami SQL Servera
  - serveradmin
    - Wykonuje różnego rodzaju konfiguracje serwera
  - setupadmin
    - Instaluje replikacje
  - securityadmin
    - Zarządza loginami na serwerze
  - bulkadmin
    - Może wykonywać instrukcje typu BULK INSERT

Role serwera

- Przydzielanie/odbieranie ról użytkownikom
  - Enterprise Manager
    - (local) | Security | Server Roles
    - (local) | Security | Logins
      prawy p. myszy na koncie, Właściwości, Server Roles
  - T-SQL
    - sp_addsrvrolemember
      [@loginame =] '*login*', [@rolename =] '*role*‘
    - sp_dropsrvrolemember
      [@loginame =] '*login*' , [@rolename =] '*role*'

Role baz danych

- Stałe role bazy danych
  - public
    - Zawiera domyślne uprawnienia
  - db_owner
    - Ma w bazie danych pełne uprawnienia
  - db_accessadmin
    - Pozwala dodawać i usuwać użytkowników, grupy i role bazy
  - db_securityadmin
    - Zarządza uprawnieniami do obiektów i ich własnością, rolami i członkostwem w rolach

Role baz danych

- Stałe role bazy danych c.d.
  - db_ddladmin
    - Pozwala wstawiać, modyfikować i usuwać obiekty bazy, ale nie może wykonywać poleceń GRANT, REVOKE, DENY
  - db_backupoperator
    - Może wydawać polecenia DBCC, CHECKPOINT, BACKUP
  - db_datareader
    - Może pobierać dowolne dane z tabel użytkowników

Role baz danych

- Stałe role bazy danych c.d.
  - db_datawriter
    - Może modyfikować dowolne dane w tabelach użytkowników
  - db_denydatareader
    - Nie może pobierać żadnych danych z tabel użytkowników
  - db_denydatawriter
    - Nie może modyfikować żadnych danych w tabelach użytkowników

Role baz danych

- Znaczenie roli public
  - Zawiera domyślne uprawnienia dla użytkowników bazy danych
  - Nie może zawierać użytkowników, ani innych ról, gdyż każdy domyślnie ma rolę public
  - Jest zawarta we wszystkich bazach
  - Nie można jej usunąć
  - Daje bardzo ograniczony zestaw uprawnień
  - Użytkownik dostaje uprawnienia tej roli poprzez konto gościa (guest)

Role baz danych użytkownika

- Rolę użytkownika tworzymy zwykle z dwóch powodów:
  - grupa użytkowników chce wykonywać podobne czynności w SQLServerze
  - nie mamy uprawnień do zarządzania kontami i grupami w systemie Windows

Role baz danych

- Zarządanie rolami w Enterprise Manager
  - tworzenie roli użytkownika
    - Enterprise Manager | Databases | <baza danych>
    - prawy p. myszy na Roles, New Database Role...
  - usuwanie roli użytkownika
    - Enterprise Manager | Databases | <bd> | Roles
    - prawy p. myszy na roli i Usuń

Role baz danych

- Zarządanie rolami za pomocą T-SQL
  - Zarządzanie rolą standardową
    - sp_addrole [@rolename =] 'role'
      [, [@ownername =] 'owner']
    - sp_droprole [@rolename =] 'role‘
    - sp_helprole [[@rolename =] 'role']
  - Edycja listy członków roli
    - sp_addrolemember [@rolename =] 'role' ,
      [@membername =] 'security_account‘
    - sp_droprolemember [@rolename =] 'role' ,
      [@membername =] 'security_account'
    - sp_helprolemember [[@rolename =] 'role']

Uprawnienia

- Dostępne zestawy uprawnień
  - Uprawnienia na tabele i widoki
    - SELECT, INSERT, UPDATE i DELETE
  - Uprawnienia na kolumny
    - SELECT, UPDATE i REFERENCES
  - Uprawnienia do procedur
    - EXECUTE

Uprawnienia

- Zarządzanie uprawnieniami w T-SQL
  - GRANT – zezwala na wykonanie zadania
  - REVOKE – odbiera zezwolenie na wykonanie
  - DENY – zabrania wykonania zadania
  - Przegląd składni poleceń w helpie
- Przegląd uprawnień w Enterprise Manager
  - Uprawnienia do tabeli
  - Uprawnienia do bazy danych

Zarządzanie bezpieczeństwem

- Przy planowaniu zabezpieczeń warto rozważyć:
  - Ustawienia kont domyślnych
    - sa
      - jest to konto administratora i należy używać jak najrzadziej
    - BUILTIN\Administrators
      - grupa ta jest automatycznie członkiem roli sysadmin
      - jeśli nie chcemy dawać administratorom pełnego dostępu do SQLServera, możemy to konto usunąć z grupy sysadmin lub w ogóle usunąć z SQLServera
  - Ustawienie konta guest
    - w której bazie ma być, jakie powinny być jego uprawnienia i kto będzie z niego korzystał

Zarządzanie bezpieczeństwem

- Przy planowaniu zabezpieczeń warto rozważyć:
  - Ustawienia roli public
    - jest to rola, do której należy każdy użytkownik
    - warto przemyśleć zestaw uprawnień dla tej roli
  - Przyznawanie uprawnień do ról
    - warto uprawnienia zorganizować za pomocą hierarchi ról – upraszcza to zarządzanie uprawnieniami
    - należy unikać nadawania uprawnień pojedynczym użytkownikom
  - Tworzenie obiektów przez dbo
    -

Zarządzanie bezpieczeństwem

- Przy planowaniu zabezpieczeń warto rozważyć:
  - Tworzenie obiektów przez dbo
    - jest ważne żeby ustalić, kto może tworzyć obiekty w bazie danych
      - domyślnie są to członkowie ról: sysadmin, db_owner, db_ddladmin
    - wygodnie jest, jeżeli właścicielem wszystkich obiektów jest dbo
    - właściciela można zmienic za pomocą procedury
      - sp_changeobjectowner object, owner

Zarządzanie bezpieczeństwem

- Można także poprzez
  - Widoki
    - można dać uprawnienie do widoku nie dając uprawnienia od tabeli bazowej dla widoku
  - Procedury składowane

Przykład 2

- Demonstracja

Role aplikacji

- Rodzaje ról użytkownika
  - Standard
    - dostęp poprzez członkostwo w roli
  - Application
    - dostęp poprzez hasło
    - do przełączenia na tą rolę używamy procedury
      - sp_setapprole [@rolename =] '*role*' , [@password =] {Encrypt N '*password*'} | '*password*'
    - po przełączeniu na taką rolę użytkownik
      - traci wszystkie wcześniejsze prawa dostępu
      - ma dostęp do tylko tej jednej bazy lub do baz, w których jest konto guest
    - tworzona często w celu nadania dostępu do bazy aplikacjom zewnętrznym

Role aplikacji

- Zarządanie rolami za pomocą T-SQL
  - Zarządzanie rolą aplikacji
    - sp_addapprole [@rolename =] '*role*',
      [@password =] '*password*'
    - sp_dropapprole [@rolename =] '*role*‘
    - sp_setapprole [@rolename =] '*role*' ,
      [@password =] {Encrypt N '*password*'} | '*password*'
      [,[@encrypt =] '*encrypt_style*']
  - Przykład
    - EXEC sp_setapprole 'Test', {Encrypt N 'pswd'}, 'odbc'

Przykład 3

- Demonstracja

