

ITA-101 Bazy Danych

Włodzimierz Dąbrowski, Przemysław Kowalczyk, Konrad Markowski

Moduł 10

Wersja 2.0

Bezpieczeństwo w bazach danych

Spis treści

Bezpieczeństwo w bazach danych	1
Informacje o module.....	2
Przygotowanie teoretyczne	3
Przykładowy problem	3
Podstawy teoretyczne.....	3
Przykładowe rozwiązanie.....	6
Porady praktyczne	10
Uwagi dla studenta	10
Dodatkowe źródła informacji.....	10
Laboratorium podstawowe.....	12
Problem 1 (czas realizacji 45 min).....	12
Laboratorium rozszerzone	17
Zadanie 1 (czas realizacji 90 min).....	17

Informacje o module

Opis modułu

W tym module dowiesz się, jak należy rozumieć bezpieczeństwo baz danych oraz jakie są poziomy bezpieczeństwa. Ponadto dowiesz się, jakim zagrożeniom należy przeciwdziałać, a jakich nie da się uniknąć oraz jak należy planować implementację poszczególnych poziomów bezpieczeństwa w aplikacji bazodanowej.

Cel modułu

Celem modułu jest przedstawienie czytelnikowi typowych zagadnień związanych z zabezpieczeniami dostępu do danych w SQL Server 2008.

Uzyskane kompetencje

Po zrealizowaniu modułu będziesz:

- wiedział jakie mechanizmy uwierzytelniania wspiera SQL Server 2008
- potrafił dodać użytkownika i nadać mu odpowiednie prawa
- rozumiał czym są schematy zabezpieczeń

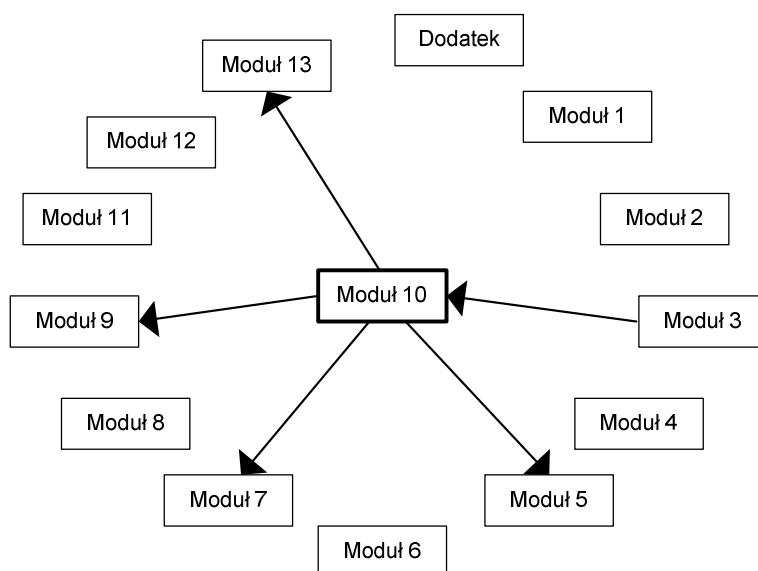
Wymagania wstępne

Przed przystąpieniem do pracy z tym modulem powinienś:

- wiedzieć, jak używać oprogramowania Microsoft Virtual PC
- znać podstawy obsługi systemu Windows 2000 lub nowszego
- znać podstawy obsługi SQL Server Management Studio

Mapa zależności modułu

Zgodnie z mapą zależności przedstawioną na rys. 1, istnieje konieczność wykonania wcześniej modułu 3.



Rys. 1 Mapa zależności modułu

Przygotowanie teoretyczne

Przykładowy problem

Firma National Insurance wdrożyła Microsoft SQL Server 2008. Założono na nim kilka baz i hurtowni danych. Ze względu na pilne potrzeby firmy szybko też zasilono bazy z dostępnych źródeł danych. Dostęp do poszczególnych baz danych zrealizowany jest poprzez dedykowane aplikacje łączące się z serwerem Microsoft SQL Server 2008 na prawach administratora. W pierwszych tygodniach po wdrożeniu system działał sprawnie, lecz później pojawiły się problemy z wydajnością i dostępem do danych. Niektórzy menadżerowie zauważyli szereg niezgodności, informacje jakie wprowadzali nie zgadzały się z danymi w raportach uzyskanych z systemu. Po krótkim czasie okazało się, że osoby nieuprawnione mają dostęp do poufnych danych, nie ma także żadnej polityki bezpieczeństwa dla serwerów bazodanowych.

Rozwiązanie tego problemu złożono na barki głównego informatyka, jako osoby kompetentnej i odpowiedzialnej za rozwój systemu bazodanowego firmy. W pierwszych krokach jakie podjął on po otrzymaniu zadania był dokładny przegląd stanu obecnego systemu i porównanie go z wytycznymi i najlepszymi praktykami z zakresu bezpieczeństwa bazodanowego.

Podstawy teoretyczne

Pojęcie bezpieczeństwa baz danych wiąże się nieodłącznie z bezpieczeństwem serwera baz danych. W hierarchii bezpieczeństwa takiego serwera stoi wyżej niż bezpieczeństwo pojedynczej bazy, ponieważ brak zabezpieczeń na tym poziomie pociąga za sobą brak zaufania do pojedynczych baz danych znajdujących się na serwerze.

Bezpieczeństwo serwera baz danych to:

- zapewnienie stabilnego i w miarę możliwości bezawaryjnego działania serwera baz danych
- zapewnienie uprawnionym użytkownikom dostępu do odpowiednich baz danych
- ograniczenie dostępu do danych dla użytkowników nieuprawnionych
- zapewnienie jak najmniejszej ingerencji serwera baz danych w działanie systemu operacyjnego komputera

Bezpieczeństwo baz danych natomiast dotyczy następujących aspektów:

- umożliwienie tylko autoryzowanym użytkownikom wykonywania odpowiednich operacji na bazie danych
- zapewnienie bezpieczeństwa fizycznego bazy danych (odpowiednia strategia kopii zapasowych)

Mówiąc o bezpieczeństwie należy rozróżniać dwa pojęcia: *uwierzytelnienie* oraz *autoryzacja*. Pierwsze pojęcie oznacza identyfikację użytkownika na podstawie jego nazwy i hasła. Z kolei autoryzacja jest fazą następującą po poprawnym uwierzytelnieniu i polega na określeniu uprawnień przypadających uwierzytelnionemu użytkownikowi.

Poziomy bezpieczeństwa

W najogólniejszym ujęciu można wyodrębnić następujące poziomy bezpieczeństwa:

- bezpieczeństwo fizyczne danych
- bezpieczeństwo sieci
- bezpieczeństwo domeny
- bezpieczeństwo maszyny lokalnej
- bezpieczeństwo serwera baz danych
- bezpieczeństwo bazy danych
- bezpieczeństwo aplikacji bazodanowej

Bezpieczeństwa doskonałego nie można w praktyce nigdy zapewnić, ale można podjąć kroki, by zapobiegać skutkom wszelkich awarii, katastrof lub niepożądanych ingerencji czynnika ludzkiego. Aby zadbać o globalne bezpieczeństwo, należy zaplanować strategię na każdym z wymienionych poziomów.

Bezpieczeństwo fizyczne danych

Poziom bezpieczeństwa fizycznego danych określa, czy w przypadku awarii sprzętu, katastrofy (jako katastrofę rozumiemy nie tylko czynniki naturalne, jak np. powodzie, lecz także kradzieże i inne wpływy czynnika ludzkiego) lub fizycznego uszkodzenia plików danych jesteśmy w stanie odtworzyć dane i jak długo baza danych (lub serwer baz danych) będzie niedostępny dla użytkowników. Na tym poziomie należy też odpowiedzieć na pytanie, czy kopie danych są bezpieczne (m.in. czy niepowołane osoby nie mają do nich dostępu).

Bezpieczeństwo sieci

Poziom bezpieczeństwa sieci określa, czy dane są bezpiecznie przesyłane w sieci. Szczególnie dotyczy to ściśle poufnych danych, tj. numerów kart kredytowych czy danych personalnych klientów firmy.

Bezpieczeństwo domeny

Poziom bezpieczeństwa domeny określa, czy komputery w domenie (w szczególności kontrolery domeny) są odpowiednio zabezpieczone. W dobie integracji serwerów baz danych (np. Microsoft SQL Server) z systemami operacyjnymi, w przypadku braku zabezpieczeń w systemie operacyjnym bezpieczeństwo serwera baz danych spada do minimum.

Bezpieczeństwo serwera baz danych

Poziom bezpieczeństwa serwera baz danych określa, czy serwer baz danych jest odpowiednio zabezpieczony przed nieuprawnionymi użytkownikami (fizycznie – maszyna – oraz wirtualnie – odpowiednie mechanizmy uwierzytelniające).

Bezpieczeństwo bazy danych

Poziom bezpieczeństwa bazy danych określa, czy dostęp do bazy danych i ról w bazie danych jest odpowiednio skonfigurowany (na ogół jest to sprawa konfiguracji w systemie bazodanowym).

Bezpieczeństwo aplikacji bazodanowej

Poziom bezpieczeństwa aplikacji bazodanowej określa, czy kod aplikacji klienckiej współpracującej z bazą danych jest napisany w sposób bezpieczny (czy aplikacja nie umożliwia zmniejszenia bezpieczeństwa na którymkolwiek z pozostałych poziomów). Szczególnie należy tu zwrócić uwagę na dane wprowadzane przez użytkowników.

Implementacja różnych poziomów bezpieczeństwa

Każdy z poziomów bezpieczeństwa wymaga podjęcia określonych kroków przez administratorów systemów i baz danych.

Implementacja bezpieczeństwa fizycznego

Zadaniem administratora baz danych jest zapewnienie tolerancji błędów dysków fizycznych dla systemu i dla danych oraz zaplanowanie strategii sporządzania i przechowywania kopii zapasowych.

Tolerancję błędów dysków fizycznych można osiągnąć używając woluminów RAID (ang. *Redundant Array of Independent Disks*) typu RAID-1 lub RAID-5.

Implementacja RAID-1 polega na jednoczesnym przechowywaniu danych na dwóch fizycznych dyskach stanowiący jeden dysk logiczny (dwie kopie danych – w przypadku awarii jednego dysku, drugi nadal umożliwia dostęp do danych). Oznacza to, że 50% pojemności woluminu typu RAID-1

jest przeznaczone na przechowywanie danych, a druga połowa służy do przechowywania kopii danych.

RAID-5 to dysk logiczny składający się z co najmniej trzech dysków fizycznych (z każdego dysku wolumin zabiera tyle samo przestrzeni dyskowej). W woluminach typu RAID-5 część przestrzeni dyskowej jest poświęcana na zapis tzw. *danych parzystości* (niezbędnych do odzyskania danych w przypadku awarii jednego z dysków wchodzących w skład woluminu). Im więcej dysków wchodzi w skład woluminu, tym mniej przestrzeni dyskowej zajmują dane parzystości (mniejsza nadmiarowość danych).

Najlepszym rozwiązaniem w kwestii zapewnienia tolerancji błędów dysków fizycznych są sprzętowe woluminy RAID pracujące z kontrolerami SCSI, z uwagi na szybszą pracę niż RAID programowy. Niestety jest to jednocześnie najdroższe rozwiązanie.

Kopie bezpieczeństwa, zwane też kopiami zapasowymi (ang. *backup*), powinny być przechowywane bądź na zewnętrznym nośniku (taśmy, płyty CD lub inne nośniki) lub na innym komputerze niż ten, z którego kopiujemy dane. Ponadto nośniki z kopiami zapasowymi powinny być przechowywane w innym miejscu niż maszyna, z której pochodzą dane (zmniejszamy ryzyko utraty danych w przypadku pożarów czy powodzi).

Strategia kopii zapasowych powinna być zaplanowana przez administratora baz danych i administratora systemu operacyjnego. Należy zaplanować strategię, która odpowiada potrzebom firmy, tzn. należy odpowiedzieć na pytanie, czy ważniejsze jest szybkie sporządzanie kopii zapasowych, czy też istotniejsze jest jak najszybsze przywracanie danych po awarii. Na ogół strategia musi uwzględnić obie kwestie. Stąd najczęściej powtarzanym schematem sporządzania kopii zapasowych jest wykonywanie co tydzień kopii wszystkich danych oraz codzienne wykonywanie kopii przyrostowych (tylko dane zmodyfikowane danego dnia).

W budowaniu strategii kopii zapasowych należy też uwzględnić „godziny szczytu” pracy serwera (proces wykonywania kopii zapasowych pociąga za sobą dodatkowe obciążenie serwera). Dlatego na ogół operacje te są wykonywane w godzinach nocnych i są planowane w ten sposób, by nie kolidowały z czasem, gdy użytkowanie serwera przez klientów jest najintensywniejsze.

Implementacja bezpieczeństwa sieci

Przy planowaniu bezpieczeństwa sieci należy zadać sobie pytanie, czy dane przesyłane z naszego serwera baz danych są poufne. Jeśli tak, to możemy zastosować dostępne protokoły szyfrujące, takie jak SSH czy IPSec. Oprócz implementacji sieciowych protokołów szyfrujących do transmisji danych, należy ograniczyć ilość danych wysyłanych w świat do niezbędnego minimum (najlepiej nie „przedstawiać się” zbyt w sieci – ujawnienie oprogramowania serwera baz danych to pierwszy krok do zachwiania bezpieczeństwa naszego serwera).

Implementacja bezpieczeństwa komputerów i domen

Aby zapewnić komputerom i domenom niezbędny poziom bezpieczeństwa, należy trzymać się kilku zasad.

Nie należy instalować serwerów baz danych na serwerach kluczowych dla domeny (kontrolery domeny). Najlepsza struktura domeny to taka, w której każdy serwer pełni pojedynczą funkcję (np. serwer aplikacji, serwer plików, serwer baz danych itd.).

Niezbędna jest odpowiednia polityka administratorów systemu (lub domeny), czyli:

- utrzymywanie aktualnego poziomu zabezpieczeń systemu operacyjnego oraz serwera baz danych
- odpowiednia polityka bezpiecznych haseł użytkowników
- zmiana nazw kont administratorskich
- monitorowanie logowania do systemu (domeny)

- ograniczanie dostępu do plików i folderów
- nadawanie minimalnych wymaganych uprawnień dla użytkowników i grup
- jak najmniejsze wykorzystywanie kont administratorskich
- implementacja „zapór ogniowych” (ang. *firewall*)
- ograniczenie fizycznego dostępu do serwerów i kontrolerów domeny
- uruchamianie usług serwera baz danych przy użyciu konta użytkownika specjalnie stworzonego w tym celu (nie administratora) i zapewnienie stabilności tego konta (np. nigdy nie wygasające hasło)

Implementacja bezpieczeństwa serwera baz danych i samych baz

Pod hasłem bezpieczeństwa serwera baz danych rozumiemy umożliwienie korzystania z serwera tylko osobom do tego uprawnionym. Większość systemów zarządzania bazami danych oferuje uwierzytelnianie użytkowników na dwóch poziomach: na poziomie serwera (użytkownik może dostać się do serwera) oraz na poziomie bazy danych (użytkownik serwera ma dostęp do konkretnej bazy danych).

Mechanizmy uwierzytelniania i autoryzacji są różne i zależą od konkretnego środowiska bazodanowego. Zazwyczaj użytkownicy dzieleni są na role (grupy), natomiast rola nadawane są określone uprawnienia. Ponadto niezbędnym nawykiem administratora baz danych powinno być rejestrowanie i monitorowanie zdarzeń na serwerze w poszukiwaniu nietypowych zdarzeń.

Implementacja bezpieczeństwa aplikacji bazodanowej

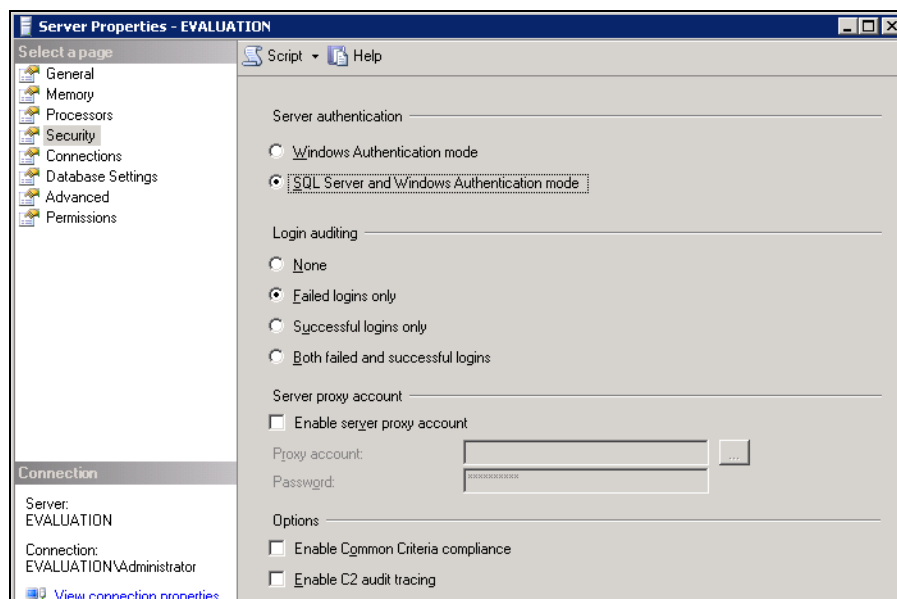
Piętą achillesową systemu informatycznego współpracującego z bazą danych często jest interfejs użytkownika (od strony programistycznej i implementacji logiki biznesowej). Szczególnie chodzi tu o umożliwienie użytkownikom oddziaływania na serwer baz danych lub nawet na system operacyjny serwera z poziomu aplikacji klienckiej.

Należy ze szczególną uwagą projektować aplikacje bazodanowe. Oto kilka zasad, którymi należy się kierować przy tworzeniu interfejsów dla tych aplikacji:

- zachowaj przezroczystość aplikacji i bazy danych (nie pokazuj informacji o źródle aplikacji i o strukturze bazy danych), szczególnie uważaj na komunikaty domyślne aplikacji (lepiej ustawić swoje, które powiedzą tylko, że wystąpił błąd)
- nigdy nie ufaj użytkownikowi aplikacji i wpisywanym przez niego wartościom
- sprawdzaj, czy wejście jest tym, czego oczekujesz i odrzucaj wszystko inne wartości
- walidację wejścia przeprowadzaj na wielu poziomach
- używaj wyrażeń regularnych
- staraj się nie używać konkatenacji do tworzenia zapytań SQL (zamiast tego użyj procedur z parametrami)
- łącz się z bazą danych używając w miarę najmniej uprzywilejowanego konta użytkownika

Przykładowe rozwiązanie

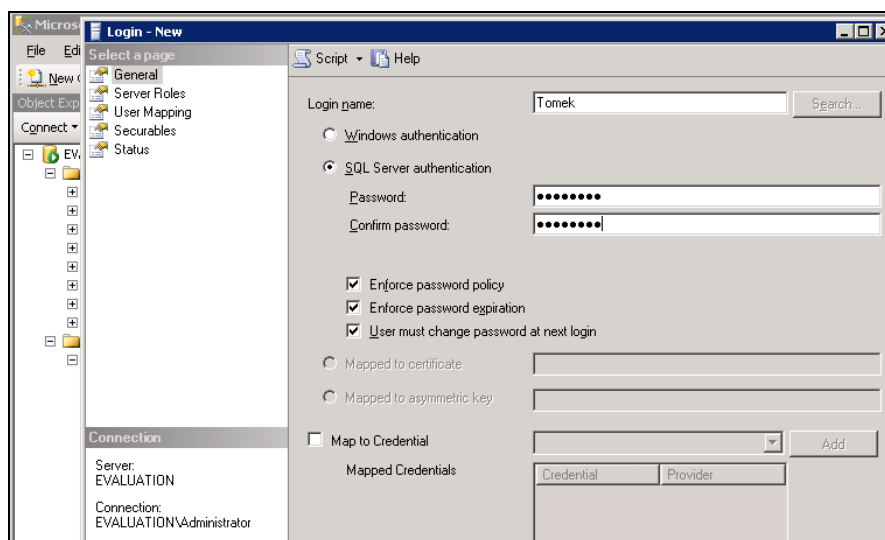
Zapewnienie bezpieczeństwa serwerowi bazodanowemu jest sprawą złożoną i rozciągniętą na kilka poziomów. Z punktu widzenia administratora systemu podstawowym poziomem jest kwestia autoryzacji użytkowników, którzy mają dostęp do serwera SQL. Narzędzie SQL Server Management Studio umożliwia kontrolę nad wieloma parametrami nie tylko bazy danych, ale też samego serwera. Jedną z grup interesujących nas parametrów jest sposób uwierzytelniania użytkowników, co ilustruje rys. 2. Serwer SQL może wykorzystywać do uwierzytelniania własne konta użytkowników lub dodatkowo honorować konta systemu operacyjnego.



Rys. 2 Opcje bezpieczeństwa serwera

Mieszany tryb uwierzytelniania, który jest zaznaczony na rys. 2, pozwala na łączenie się z serwerem także użytkownikom, którzy nie mają kont w systemie Windows w sieci, w której pracuje serwer. Dobrą praktyką jest monitorowanie nieudanych prób logowania do systemu, co zapewnia opcja **Login auditing**.

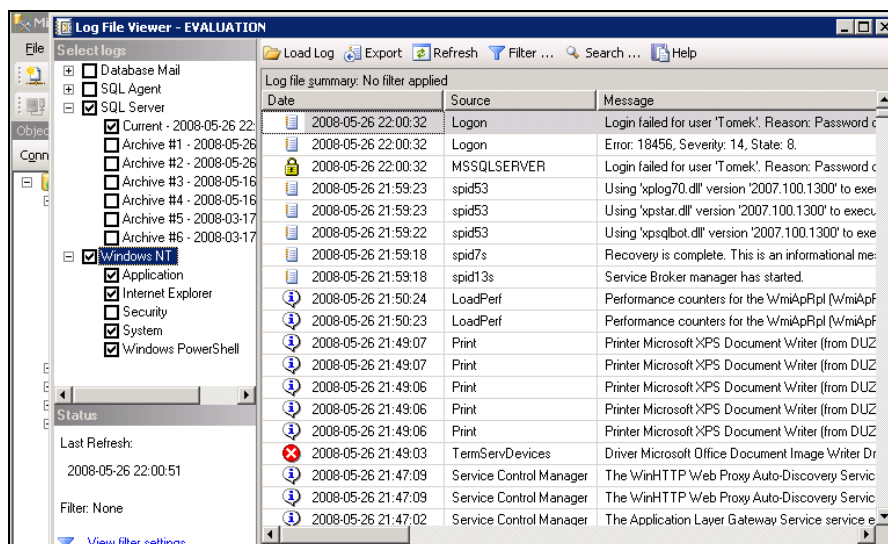
Inną interesującą grupą są obiekty związane wewnętrznymi kontami użytkowników serwera SQL dostępne w polu **Logins**. Procedura tworzenia nowego użytkownika jest podobna do analogicznej procedury w systemie operacyjnym co ilustruje rys. 3.



Rys. 3 Tworzenie nowego loginu

Jeśli Microsoft SQL Server 2008 zainstalowany jest na komputerze pracującym pod kontrolą systemu Microsoft Windows Server 2003, można wymusić odpowiednią politykę bezpieczeństwa haseł serwera baz danych dzięki polisom systemu operacyjnego.

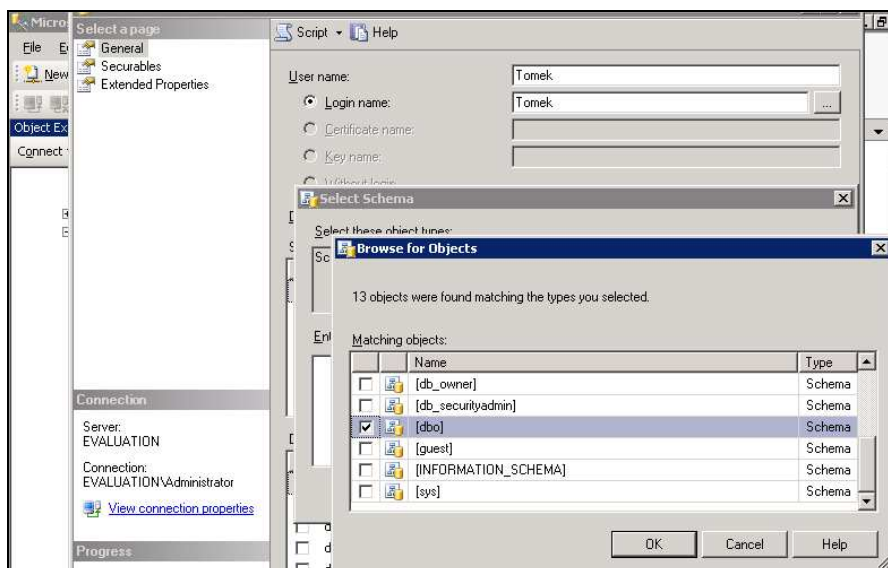
Po założeniu odpowiednich kont możemy sprawdzić, czy logowanie do serwera przebiegało pomyślnie czy też były z tym jakieś problemy. Do monitorowania aktywności serwera służy dziennik. Przykładową zawartość dziennika serwera SQL pokazuje rys. 4.



Rys. 4 Dziennik systemowy SQL Server 2008

Dobry administrator śledzi przynajmniej nieudane próby logowania do systemu. Dziennik pokazany na rys. 4 to dziennik to po prostu zdarzeń systemu Windows. Przeglądarka dziennika systemu SQL Server umożliwia jednocześnie przeglądanie wszystkich zapisywanych w tym systemie informacji.

Utworzenie konta dla danego użytkownika nie oznacza jeszcze przyznanie mu jakichkolwiek praw poza możliwością połączenia z serwerem. Aby dany użytkownik mógł skorzystać z baz danych należy w kontekście danej bazy przyznać mu prawo do połączenia się z nią. Po rozwinięciu drzewa bazy w obszarze **Object Explorer** widzimy pole **Security**, gdzie możemy ustalać, który użytkownik posiada dostęp i na jakich zasadach, co ilustruje rys. 5.

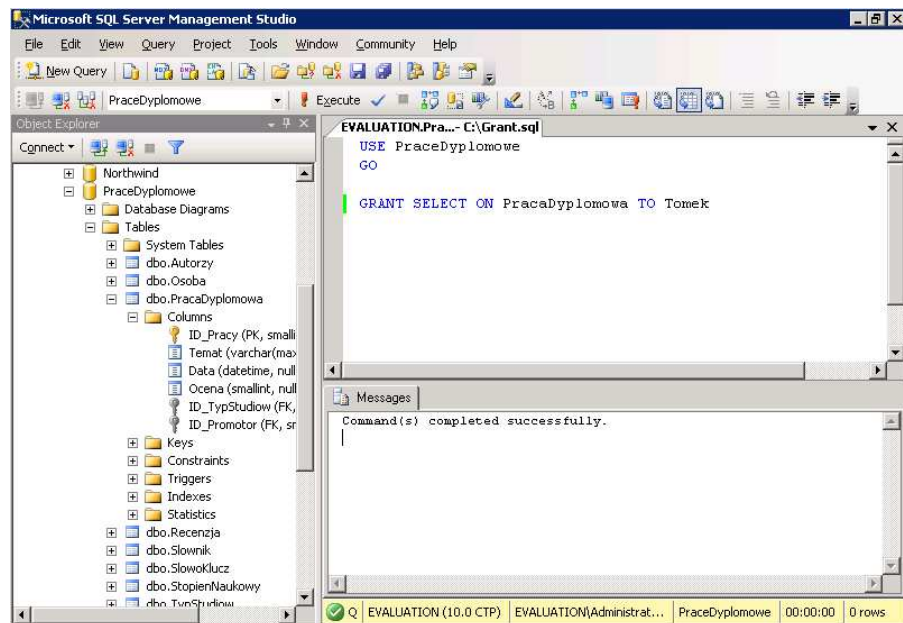


Rys. 5 Dodawanie użytkownika do bazy danych

Podobny efekt możemy uzyskać uruchamiając odpowiednią sekwencję kodu języka SQL. Do nadawania i odbierania uprawnień użytkownikom służą polecenia GRANT i REVOKE, tak jak to pokazuje rys. 6. Należy jednak zauważyć, że do uruchomienia danego kodu SQL musimy posiadać konto, które:

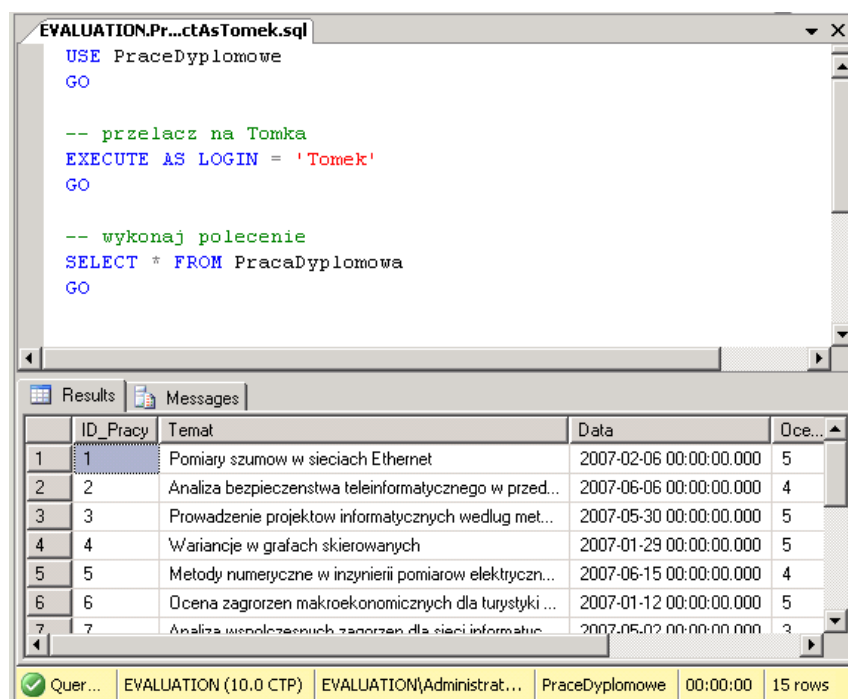
- ma dostęp do danej bazy
- posiada odpowiednie uprawnienia np. administracyjne w kontekście tej bazy

W innym wypadku kontrolowanie uprawnień z poziomu języka SQL nie powiedzie się.



Rys. 6 Nadanie uprawnień użytkownikowi Tomek

Jeżeli posiadamy w systemie kilka kont użytkowników, które są odpowiednio skonfigurowane i posiadają uprawnienia dostępu do bazy, możemy przełączać pomiędzy nimi kontekst wykonywanych poleceń SQL. Służy do tego polecenie EXECUTE AS, jak pokazano na rys. 7.



Rys. 7 Wykonanie kodu SQL w kontekście użytkownika Tomek

Komenda EXECUTE AS pozwala zmienić kontekst wykonywania poleceń na wybranego użytkownika. Powrót do pierwotnego kontekstu zapewnia polecenie REVERT.

W rzeczywistych systemach pojawia się wiele kont i grup użytkowników. Pomoc w zapanowaniu nad mnogością kont zapewniają *schematy*. Schematy to przestrzenie nazw lub pojemniki na obiekty w bazie danych. Upraszczają one zarządzanie uprawnieniami w bazie danych oraz stanowią element niezbędny do poprawnego rozwiązywania nazw w systemie Microsoft SQL Server 2008.

Schematy umożliwiają nadawanie uprawnień na wiele obiektów jednocześnie. Wystarczy umieścić je w jednym schemacie. Ponadto schematy pozwalają uniknąć sytuacji, w których usunięcie użytkownika z bazy jest niemożliwe, gdy jest on właścicielem obiektów w bazie danych (wcześniej trzeba zmienić właścicieli wszystkich obiektów, których właścicielem jest wspomniany użytkownik).

Do tworzenia schematów służy polecenie `CREATE SCHEMA`. Więcej informacji na temat tworzenia i zarządzania schematami znajduje się w laboratorium podstawowym i Books Online na stronie firmy Microsoft.

Porady praktyczne

Nigdy nie myśl, że system i serwer baz danych są bezpieczne. Jest to jedna z podstawowych zasad przy projektowaniu lub inspekcji mechanizmów zabezpieczających systemy, nie tylko informatyczne. Takie podejście znacznie zwiększa szanse na znalezienie luki lub potencjalnego problemu.

Nigdy nie ufaj temu, co użytkownik podaje na wejściu do systemu. Jeżeli przewidujesz możliwość wprowadzania danych przez użytkownika zawsze staraj się prawidłowo i uważnie obsługiwać pojawiające się informacje automatycznie odrzucając wartości, których się w danej sytuacji nie spodziewasz.

Zachowuj zasadę minimalnych uprawnień w stosunku do użytkownika. Prawidłowe podejście z punktu widzenia bezpieczeństwa to w pierwszym kroku zabranie użytkownikowi wszystkich uprawnień w systemie a dopiero później ostrożne przydzielenie mu takich jakie wydają się być niezbędne. Wprowadza to oczywiście wydłużenie czasu dostosowania systemu do pracy w pełnym wymiarze oraz swoiste niezadowolenie użytkowników ale jest niezbędne. Zasada ta jest szczególnie ważna dla użytkowników typu serwis systemowy i implikuję kolejną o nazwie „domyślnie zamknięte”. Obszary działania systemu takie jak porty dostępu, protokoły komunikacyjne czy same bazy danych jeżeli nie są w danej chwili używane powinny mieć status zamkniętych dla użytkownika. Dopiero formalna potrzeba użycia danej części systemu może ją aktywować. Eliminuje to znakomitą część prób włamań na nieużywane, „uśpione” ale ciągle aktywne zasoby.

Regularnie szukaj nieprawidłowości w systemie. Systemy informatyczne to najczęściej twory o silnej dynamice podlegające ciągłym zmianom. Zmiany te mogą tworzyć nowe, potencjalne „furtki” dla włamywaczy. Inną sprawą jest niedoskonałość samego oprogramowania. Co prawda dla rozwijanych systemów co jakiś czas wydawane są aktualizacje jednakże praktyka wskazuje, że łatki takie potrafią naprawiać jedną część a jednocześnie stwarzać luki gdzie indziej.

Bądź na bieżąco z technologiami i technikami programistycznymi aby wiedzieć jak reagować na potencjalne zagrożenia. Wiedza ta w przypadku systemów bazodanowych jest szczególnie cenna gdyż część funkcjonalności administrator może sam bezpośrednio oprogramować a co za tym idzie posiadać nad nią całkowitą kontrolę.

Uwagi dla studenta

Jesteś przygotowany do realizacji laboratorium jeśli:

- rozumiesz, co oznacza serwis systemowy, serwis bazy danych, instalacja serwisu
- rozumiesz zasadę działania uruchomienia serwisów w kontekście użytkownika
- umiesz wymienić i opisać podstawowe komponenty systemu bazodanowego
- umiesz podać przykład zastosowania systemu bazodanowego w praktyce

Pamiętaj o zapoznaniu się z uwagami i poradami zawartymi w tym module. Upewnij się, że rozumiesz omawiane w nich zagadnienia. Jeśli masz trudności ze zrozumieniem tematu zawartego w uwagach, przeczytaj ponownie informacje z tego rozdziału i zajrzyj do notatek z wykładów.

Dodatkowe źródła informacji

1. Kalen Delaney, *Microsoft SQL Server 2005: Rozwiązania praktyczne krok po kroku*, Microsoft Press, 2006

Podręcznik ten jest idealną pomocą dla użytkowników, którzy postawili już pierwsze kroki w systemach bazodanowych. Dużo ćwiczeń i kodów źródłowych odnoszących się do sytuacji spotykanych w praktyce jest doskonałą bazą do rozwiązywania problemów pojawiających się w rzeczywistości.

2. Edward Whalen, *Microsoft SQL Server 2005 Administrator's Companion*, Microsoft Press, 2006

Kompleksowe opracowanie na temat zaplanowania i wdrożenia system bazodanowego opartego o MS SQL Server 2005 w małym i średnim przedsiębiorstwie. Autorzy postawili na formułę przedstawiania wielu problemów z praktyki administratora baz danych oraz możliwych dróg do ich rozwiązania. Książka jest adresowana do praktykujących użytkowników.

3. Dusan Petkovic, *Microsoft SQL Server 2008: A Beginner's Guide*, McGraw-Hill, 2008

Pozycja adresowana do osób zaczynających przygodę z bazami danych. Znajdziemy tu wprowadzenie do relacyjnych baz danych, sposoby ich projektowania, optymalizacji i w końcu wdrożenia w najnowszej odsłonie serwera SQL w wersji 2008. Omówienie języka T-SQL w osobnej, dużej części książki jest kolejną mocną tej pozycji.


4. *Strona domowa SQL Server 2008*, <http://www.microsoft.com/sql/2008/default.msp>

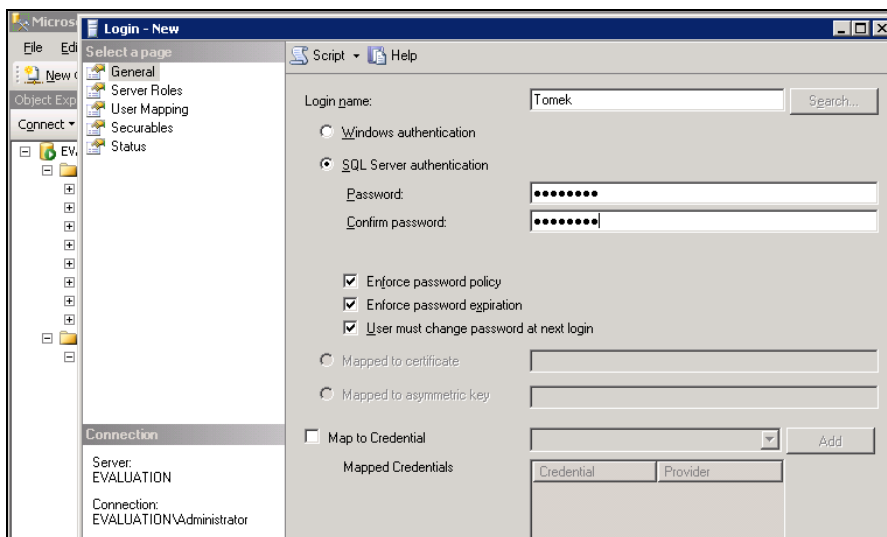
Tutaj znajdziemy wszystkie podstawowe informacje na temat MS SQL Server 2008 oraz nowości z nim związane.

Laboratorium podstawowe

Problem 1 (czas realizacji 45 min)

Pierwszym zadaniem, jakie sobie postawiłeś, jest zbadanie możliwości serwera Microsoft SQL Server 2008 pod względem tworzenia użytkowników, nadawania im uprawnień i kontroli nad tymi uprawnieniami. W celach testowych postanowiłeś wykorzystać roboczą bazę PracDypłomowe, założoną na serwerze Evaluation.

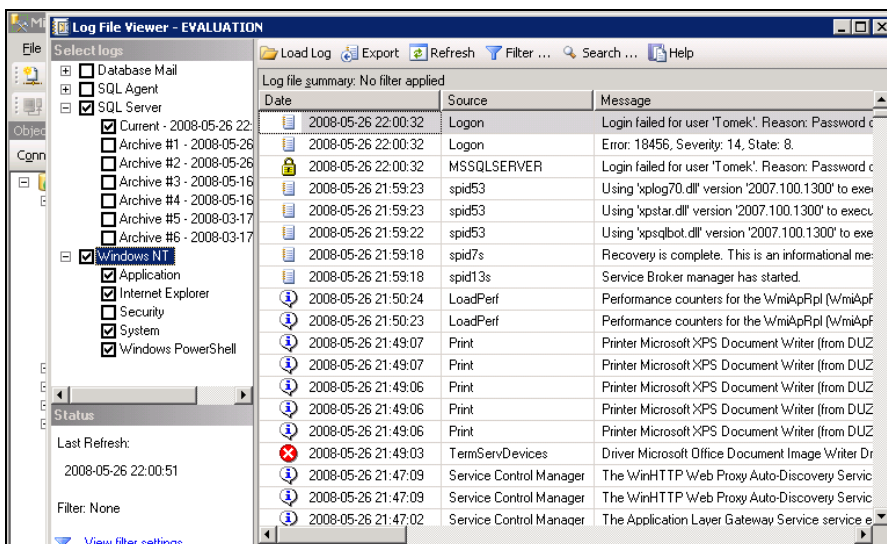
Zadanie	Tok postępowania
1. Zmień tryb uwierzytelniania	<ul style="list-style-type: none">• Uruchom maszynę wirtualną BD2008.<ul style="list-style-type: none">– Jako nazwę użytkownika podaj Administrator.– Jako hasło podaj P@ssw0rd.•  Jeśli nie masz zdefiniowanej maszyny wirtualnej w Microsoft Virtual PC, dodaj nową maszynę używając wirtualnego dysku twardego z pliku D:\VirtualPC\Dydaktyka\BD2008.vhd.• Kliknij Start. Z grupy programów Microsoft SQL Server 2008 uruchom SQL Server Management Studio.• Po lewej stronie ekranu w oknie Object Explorer kliknij prawym przyciskiem myszy nazwę serwera (EVALUATION) i z menu kontekstowego wybierz opcję Properties.• W lewej części okna z listy Select a page wybierz Security.• Zaznacz opcję SQL Server and Windows Authentication Mode.• Kliknij OK.• Kliknij OK w oknie informującym o tym, że nowe ustawienia wymagają restartu usługi serwera.• Dokonaj restartu maszyny i połącz się ponownie z serwerem SQL.
2. Utwórz loginy	<ul style="list-style-type: none">• W oknie Object Explorer rozwiń zawartość folderu Security.• Prawym przyciskiem myszy kliknij Logins i z menu kontekstowego wybierz New Login.• W polu Login name wpisz Tomek.• Zaznacz opcję SQL Server authentication.• W polach Password i Confirm password wpisz P@ssw0rd.• Odznacz opcję User must change password at next login.• Kliknij OK.



Rys. 8 Tworzenie nowego loginu

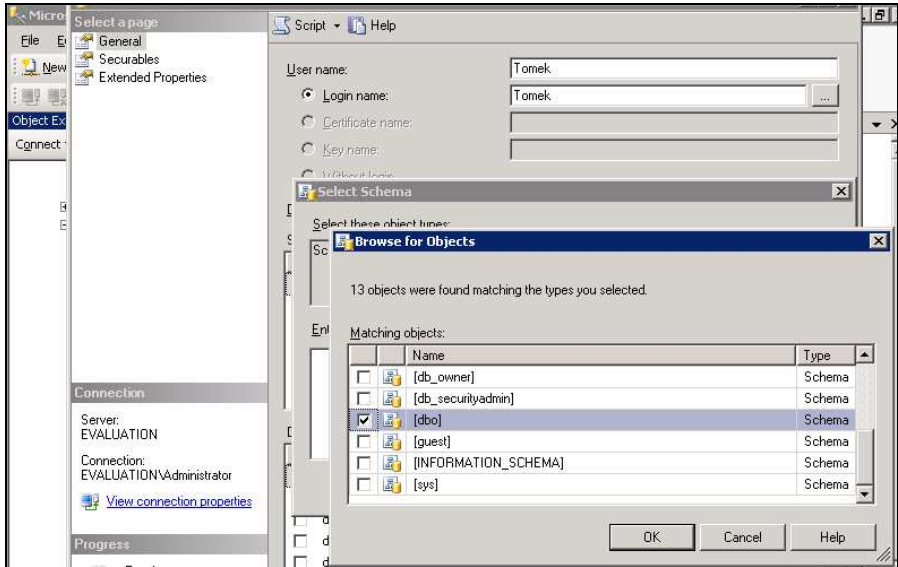
3. Przeprowadź
audyt prób
logowania

- Z menu głównego wybierz **File -> New -> Database Engine Query**.
- W oknie **Connect to Database Engine** z listy **Authentication** wybierz **SQL Server Authentication**.
- W polu **Login** wpisz **Tomek**, zaś pole **Password** pozostaw puste.
- Kliknij **OK**.
- W oknie komunikatu o nieudanej próbie logowania kliknij **OK**.
- Zamknij okno logowania klikając **Cancel**.
- W oknie **Object Explorer** rozwiń zawartość folderu **Management**.
- Prawym przyciskiem myszy kliknij **SQL Server Logs** i wybierz **View -> SQL Server and Windows Log**.
- Przeczytaj informację o nieudanej próbie logowania użytkownika **Tomek**.
- Zamknij okno dziennika systemowego.



Rys. 9 Dziennik systemowy SQL Server 2008

- Utwórz jeszcze jeden login w systemie. Po skonfigurowaniu wszystkich opcji loginu (nazwa, hasło, itd.) w górnej części okna wybierz **Script**. Przyjrzyj się składni polecenia, które pojawi się w oknie edytora.

<p>4. Dodaj użytkownika do bazy danych</p>	<ul style="list-style-type: none"> • W oknie Object Explorer rozwiń zawartość folderu Databases. • Rozwiń zawartość bazy danych PraceDyplomowe. • W bazie PraceDyplomowe rozwiń zawartość folderu Security. • Prawym przyciskiem myszy kliknij folder Users i wybierz New User. • W oknie Database User - New w pola User name i Login name wpisz Tomek (klikając na przycisku z trzema kropkami masz możliwość wyboru istniejącego loginu z listy), a w polu Default schema wpisz Sales. • W górnej części okna kliknij Script. • Kliknij OK i obejrzyj skrypt, który został wygenerowany.  <p style="text-align: center;"><i>Rys. 10 Dodawanie użytkownika do bazy danych</i></p>
<p>5. Nadaj uprawnienia użytkownikowi</p>	<ul style="list-style-type: none"> • Z menu głównego wybierz File -> Open -> File. • Odszukaj plik Grant.sql i kliknij Open. • Wciśnij F5, aby uruchomić kod. Kod ten nadaje uprawnienia do wykonywania polecenia SELECT na tabeli PracaDyplomowa użytkownikowi Tomek.
<p>6. Wykorzystaj stworzonego użytkownika</p>	<ul style="list-style-type: none"> • Z menu głównego wybierz File -> Open -> File. • Odszukaj plik SelectAsTomek.sql i kliknij Open. • Wciśnij F5, aby uruchomić kod. Wykona on polecenie SELECT jako użytkownik Tomek, któremu odpowiednie uprawnienia nadałeś w kroku 5.

EVALUATION.Pr...ctAsTomek.sql

```
USE PraceDyplomowe
GO

-- przełącz na Tomka
EXECUTE AS LOGIN = 'Tomek'
GO

-- wykonaj polecenie
SELECT * FROM PracaDyplomowa
GO
```

	ID_Pracy	Temat	Data	Oce...
1	1	Pomiary szumów w sieciach Ethernet	2007-02-06 00:00:00.000	5
2	2	Analiza bezpieczeństwa teleinformatycznego w przed...	2007-06-06 00:00:00.000	4
3	3	Prowadzenie projektów informatycznych według met...	2007-05-30 00:00:00.000	5
4	4	Wariancje w grafach skierowanych	2007-01-29 00:00:00.000	5
5	5	Metody numeryczne w inżynierii pomiarów elektryczn...	2007-06-15 00:00:00.000	4
6	6	Ocena zagrożeń makroekonomicznych dla turystyki ...	2007-01-12 00:00:00.000	5
7	7	Analiza współczesnych zagrożeń dla sieci informatyc...	2007-05-02 00:00:00.000	3

Quer... EVALUATION (10.0 CTP) EVALUATION\Administrat... PraceDyplomowe 00:00:00 15 rows

Rys. 11 Wykonanie kodu SQL w kontekście użytkownika Tomek

7. Utwórz właściciela schematu

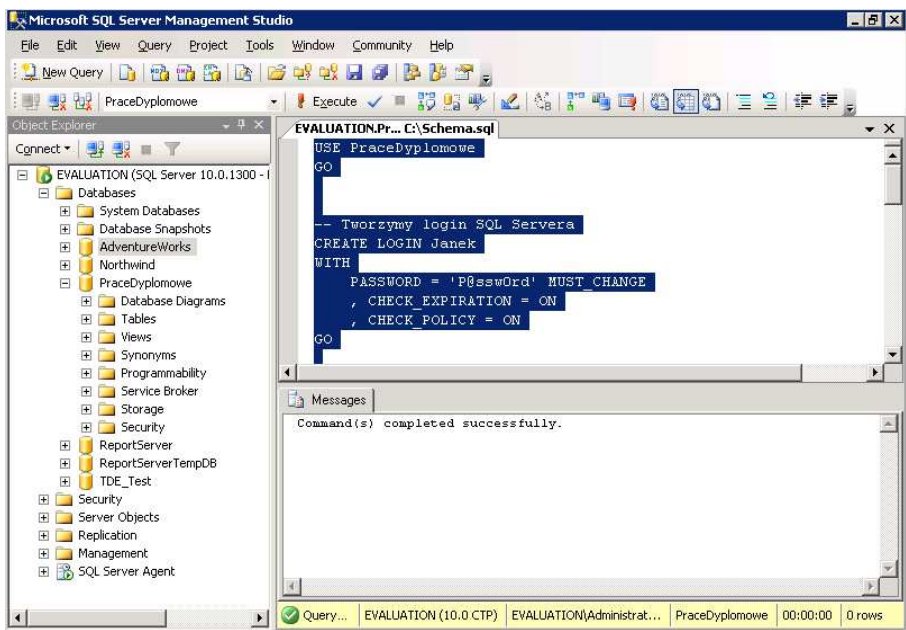


- Z menu głównego wybierz **File -> Open -> File**.
- Odszukaj plik **Schema.sql** i kliknij **Open**.
- Zaznacz kod, który tworzy użytkownika **Janek**, będącego właścicielem nowego schematu:

```
USE PraceDyplomowe
GO

CREATE LOGIN Janek
WITH
PASSWORD = 'P@ssw0rd' MUST_CHANGE
, CHECK_EXPIRATION = ON
, CHECK_POLICY = ON
GO
```

```
CREATE USER Janek
FOR LOGIN Janek
WITH DEFAULT_SCHEMA = dbo
GO
```

- Wciśnij **F5**, by uruchomić zaznaczony kod.

	 <p style="text-align: center;"><i>Rys. 12 Tworzenie nowego użytkownika SQL Server</i></p>
<p>8. Utwórz schemat</p>	<ul style="list-style-type: none"> Zaznacz i uruchom (F5) poniższy fragment kod, który tworzy schemat NewSchema dla użytkownika Janek, tabelę NewTable w tym schemacie oraz nadaje uprawnienia do wykonywania polecenia SELECT na tabeli użytkownikowi Tomek: <pre>--Tworzymy schemat CREATE SCHEMA NewSchema AUTHORIZATION Janek CREATE TABLE NewTable(col1 int, col2 int) GRANT SELECT ON NewTable TO Tomek GO</pre>
<p>9. Uzyskaj dostęp do danych</p>	<ul style="list-style-type: none"> Zaznacz kod, który przełączy kontekst użytkownika na login John: <pre>-- Zmieniamy kontekst (w SQL 2000 - setuser 'John') EXECUTE AS LOGIN = 'John';</pre> Wciśnij F5, by uruchomić zaznaczony kod. Zaznacz kod, który wykona próbę dostępu do danych: <pre>-- Error!!! Nie ma Sales.NewTable ani dbo.NewTable SELECT * FROM NewTable</pre> Wciśnij F5, by uruchomić zaznaczony kod. <p> Próba wykonania powyższego kodu spowoduje wyświetlenie komunikatu o błędzie, ponieważ nie istnieje obiekt o nazwie Sales.NewTable (Sales to domyślny schemat dla użytkownika John) ani obiekt o nazwie dbo.NewTable.</p> Zaznacz kod, który wykona ponownie próbę dostępu do danych: <pre>-- Ok. SELECT * FROM NewSchema.NewTable</pre> Wciśnij F5, by uruchomić zaznaczony kod. <p> Powyższy kod zostanie poprawnie wykonany, ponieważ Janek ma uprawnienia do wykonywania operacji na schemacie NewSchema, będąc jego właścicielem.</p>

Laboratorium rozszerzone

Zadanie 1 (czas realizacji 90 min)

Stworzenie użytkowników i powiązanie ich z odpowiednimi prawami do danych w bazach firmy National Insurance znacznie zwiększyło poziom bezpieczeństwa systemu bazodanowego. Monitoring dzienników systemowych pokazał, które aplikacje klienckie i którzy użytkownicy sprawiali problemy. Widać było także wyraźne rezultaty zabezpieczeń w postaci odrzuconych nieuprawnionych prób dostępu do danych. Wyniki tych działań zachęciły zarząd firmy do wsparcia dalszych prac nad bezpieczeństwem systemu bazodanowego. Jako główny administrator masz za zadanie zapoznać się z możliwościami szyfrowania danych zawartymi w Microsoft SQL Server 2008 oraz wdrożyć je w firmie.

- Zapoznaj się z możliwościami wykorzystania elementów kryptografii w SQL Server 2008 (plik **cryptography.sql**).
- Spróbuj wykorzystać szyfrowanie i certyfikaty cyfrowe do zabezpieczenia poszczególnych tabel bazy danych.

Wszystkie eksperymenty mają być dokonywane na bazie PracDyplomowe. Jako rezultat masz napisać raport zawierający przykłady zastosowania możliwości szyfrowania danych i autoryzacji dostępu do nich.