  
Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Інформаційно-комунікаційні системі:

Бази даних

Студент: Бичок Вадим Вячеславович

Робота: 9

Назва: НАЛАШТУВАННЯ СИСТЕМИ БЕЗПЕКИ MS SQL SERVER.

СУБД: MSSQL

**Завдання для лабораторної роботи**

1. Напишіть команди Transact SQL для наступних дій:
   1. Створити новий обліковий запис;  
      Задамо умову створення логіна тільки в тому випадку якщо такого логіна вже не існує.  
      USE mssql;  
      IF NOT EXISTS (SELECT 1 FROM sys.server\_principals WHERE name = 'My\_Love')  
      BEGIN  
       CREATE LOGIN My\_Love  
       WITH PASSWORD = 'ilove',  
       DEFAULT\_DATABASE = mssql;  
      END;

GO

Те ж саме з створенням USER. Так роблю, тому що якщо у попередньому запиті створювати одночасно з створенням логіна, при зміні з тестової бази даних library на основну mssql старий код почав видавати помилки, тому перевірка буде виконуватися два рази.

IF NOT EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = 'My\_Love')  
BEGIN  
 CREATE USER My\_Love FOR LOGIN My\_Love;  
END;

GO

* 1. Назначити користувачу роль БД (будь-яку);

Перевіряємо чи є на нашому сервері такі Логін і user після створення.

SELECT \* FROM sys.server\_principals;  
SELECT \* FROM sys.database\_principals;  
EXECUTE sp\_addsrvrolemember "My\_Love", "sysadmin";

GO

І перевіряємо користувача чи є він членом групи sysadmin.  
SELECT IS\_SRVROLEMEMBER('sysadmin', 'My\_Love') AS IsSysadmin;

GO

* 1. Надати користувачу *Bill* повноваження на доступ до збереженої процедури (будь-якої);
  2. Відкликати всі надані користувачу *John* привілегії.

Ці два пункти я об’єднав тому що видалення виконується у тій самій перевірці де виконується і надання повноважень.

Для початку його створимо як і першого користувача.

IF NOT EXISTS (SELECT 1 FROM sys.server\_principals WHERE name = 'Bill')  
BEGIN  
 CREATE LOGIN Bill  
 WITH PASSWORD = 'Camaro',  
 DEFAULT\_DATABASE = mssql;  
END;

GO

IF NOT EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = 'Bill')  
BEGIN  
 CREATE USER Bill FOR LOGIN Bill;  
END;

GO

Перед створенням перевіримо чи існує USER якому ми хочемо надати повноваження на виконання процедури.

IF EXISTS (SELECT 1 FROM sys.server\_principals WHERE name = 'Bill')  
BEGIN  
 GRANT EXEC ON CalculateValue TO Bill;  
 REVOKE ALL TO Bill;  
END;

GO

1. Поясніть призначення наступних команд Transact SQL:
   * 1. Надання повноважень користувачу J\_Smith на додавання, і вибірку записів з таблиці SUPPLIES. Дозволяємо користувачу J\_Smith передавати отримані права іншим користувачам або ролям. Без цього параметра права будуть обмежені лише для J\_Smith. Права надаються від імені ролі Economists. Це означає, що роль Economists повинна мати права на SELECT і INSERT для таблиці SUPPLIES, і права передаються саме від цієї ролі.

GRANT SELECT, INSERT   
ON SUPPLIES   
TO J\_Smith   
WITH GRANT OPTION AS Economists

* + 1. Команда EXEC[UTE] sp\_addlogin викликає збережену процедуру sp\_addlogin, яка використовується для створення нового логіна в SQL Server (це старий спосіб управління логінами, який вже застарів і замінений іншими методами, наприклад CREATE LOGIN). Перший аргумент це ім’я. Другий пароль для логіна. Третій це База даних за замовчуванням. Четвертий це мова за замовчуванням для нового логіна. П’ятий параметр застарілий у контексті сучасних версій SQL Server, раніше він визначав мовну базу даних за замовчуванням (для збереження мовної інформації), але в сучасних версіях ігнорується. І останній шостий аргумент вказує на SID (унікальний ідентифікатор безпеки). Якщо вказано NULL, сервер самостійно створює SID для нового логіна.

EXEC[UTE] sp\_addlogin ‘king\_of\_the\_db',   
‘a2h7d0f7dg84mdf94',   
‘PROJECTS',   
‘Ukrainian',   
‘master',   
‘NULL'

* + 1. Видаляємо усі повноваження у M\_Ivanenko.  
       REVOKE ALL TO ‘M\_Ivanenko'   
       CASCADE

1. Зашифруйте зміст однієї з колонок таблиці в створеній БД.

GO

Були помили що мастер-ключ уже існує, тому я на перед переглядаю чи існує ключ.

SELECT \*

FROM sys.symmetric\_keys

WHERE name = '##MS\_ServiceMasterKey##';

GO

Якщо його не існує то створюємо його і пароль до нього.

IF NOT EXISTS (SELECT 1

FROM sys.symmetric\_keys

WHERE name = '##MS\_DatabaseMasterKey##')

BEGIN

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'KeyForMyLove';

END;

GO

Тепер так само тільки для сертифікату, і пояснення для чого потрібен сертифікат.

IF NOT EXISTS (SELECT 1

FROM sys.certificates

WHERE name = 'CertifiLove')

BEGIN

CREATE CERTIFICATE CertifiLove

WITH SUBJECT = 'Protect Contact';

END;

GO

Тепер створюємо конкретний ключ задаючи алгоритм шифрування, та до якого сертифіката прив’язати цей ключ.

IF NOT EXISTS (SELECT 1

FROM sys.symmetric\_keys

WHERE name = 'KeyForMyLove')

BEGIN

CREATE SYMMETRIC KEY KeyForMyLove

WITH ALGORITHM = AES\_128

ENCRYPTION BY CERTIFICATE CertifiLove;

END;

GO

Тепер дивимося чи існує стовпець EncryptContacts в Readers, якщо ні то створюємо його.

IF NOT EXISTS (SELECT 1

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'Readers'

AND COLUMN\_NAME = 'EncryptContacts')

BEGIN

ALTER TABLE Readers

ADD EncryptContacts VARBINARY(MAX);

END;

GO

Відкриваємо ключ.

OPEN SYMMETRIC KEY KeyForMyLove

DECRYPTION BY CERTIFICATE CertifiLove;

GO

Зашифровуємо стовпець Contacts і вписуємо результат у новододаний стовпець.

UPDATE Readers

SET EncryptContacts = EncryptByKey (Key\_GUID('KeyForMyLove'), Contacts)

FROM dbo.Readers;

GO

Зашифрувавши все що потрібно закриваємо ключ.

CLOSE SYMMETRIC KEY KeyForMyLove;

GO

SELECT \* FROM Readers;

GO

Відкриваємо знову для того щоб розшифрувати.

OPEN SYMMETRIC KEY KeyForMyLove

DECRYPTION BY CERTIFICATE CertifiLove;

GO

Розшифрувуємо 'EncryptContacts' і виводимо його і розшифрований стовпець.

SELECT

Ticket, EncryptContacts,

CONVERT(varchar, DecryptByKey(EncryptContacts))

AS 'Decrypted Contacts'

FROM dbo.Readers;

CLOSE SYMMETRIC KEY KeyForMyLove;