



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-Технічний Інститут

Лабораторна робота №9
Варіант -16

Виконав:
студент групи ФІ-73
Чіхладзе Вахтанг

Викладач:
Коломицев Михайло Володимирович

Київ 2020

Варіант № 16

Предметна область: Фільмотека .

Основні предметно-значимі сутності: фільми, актори.

Основні предметно-значимі атрибути сутності:

- фільми – назва фільму, жанр, тривалість, рік виходу, стрічка (кол. чи ч.б.).
- актори – прізвище, в яких фільмах грав, дата народження; стать

Основні вимоги до функцій системи:

- вибрати всіх акторів, що грали у фільмі вказаного жанру;
- підрахувати кількість фільмів, зіграних кожним актором;
- визначити актора, що зіграв найбільше ролей за останні 10 років.

Тригери:

1. На видалення запису з таблиці «Фільми». Якщо з цим фільмом пов'язані записи в таблиці «АкториФільми», видалити і ці записи.
2. На додавання / оновлення записів в таблиці «АкториФільми». Створити представлення «ІндексАктивності» з полями «код_актора», «ім'я», «Прізвище», «кількість_фільмів», «рік_виходу», де поле «кількість_фільмів» розраховується на кожний рік, коли знімався даний актор. При додаванні запису в таблицю «АкториФільми» і при оновленні запису, якщо поле «рік_виходу» змінювалось, оновлювати вид «ІндексАктивності».

Збережена процедура:

Процедура повинна повертати кількість фільмів конкретного жанру.

Ціль заняття: Набуття навичок налаштування системи безпеки MS SQL Server.

Результат заняття: Створення користувача, розмежування прав доступу.

Завдання для лабораторної роботи

1. Напишіть команди Transact SQL для наступних дій:

- 1) Створити новий обліковий запис;
- 2) Назначити користувачу роль БД (будь-яку);
- 3) Надати користувачу Bill повноваження на доступ к збереженої процедури (будь-якої);
- 4) Відкликати всі надані користувачу John привілеї.

2. Поясніть призначення наступних команд Transact SQL:

- a) GRANT SELECT, INSERT
ON SUPPLIES
TO J_Smith
WITH GRANT OPTION AS Economists
- b) EXEC[UTE] sp_addlogin 'king_of_the_db',
'a2h7d0f7dg84mdf94',
'PROJECTS',
'Ukrainian',
'master',
'NULL';
- c) REVOKE ALL TO 'M_Ivanenko';
CASCADE

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

Хід роботи

1)

-- deleting logins

DROP LOGIN John

DROP LOGIN Bill

CREATE LOGIN John

WITH PASSWORD = '123456789ABCDEF!'

DEFAULT_DATABASE = FilmLibraryDatabase;

CREATE LOGIN Bill

WITH PASSWORD = '123456789ABCDEF!'

DEFAULT_DATABASE = FilmLibraryDatabase;

USE FilmLibraryDatabase

GO

--deleting users

DROP USER Bill

DROP USER John

--creating users

CREATE USER Bill for LOGIN Bill

CREATE USER John for LOGIN John

-- give assign of role to Bill and John:

EXECUTE sp_addrolemember 'db_ddladmin', 'Bill';

EXECUTE sp_addrolemember 'db_ddladmin', 'John';

-- delete assign of role to Bill and John:

EXECUTE sp_droprolemember 'db_ddladmin', 'Bill';

EXECUTE sp_droprolemember 'db_ddladmin', 'John';

-- gives access to execute stored procedure addActor to Bill

GRANT EXECUTE ON addActor TO Bill;

-- revoke of permissions on executing of addActor procedure

REVOKE EXECUTE ON addActor TO Bill;

GRANT SELECT, INSERT ON actors TO John; -- add permissions for SELECT & INSERT inside of actors

REVOKE SELECT, INSERT ON actors TO John; -- revoke permissions for SELECT & INSERT inside of actors

2)

а)Команда надає користувачу J_Smith привілеї на здійснення вибірки та вставки даних у таблицю SUPPLIES як для ролі Economists

б)Команда створює ім'я входу king_of_the_db з паролем a2h7d0f7dg84md94, який одразу підключається до бази даних PROJECTS з українською мовою для входу з ідентифікатором безпеки master, де пароль передається без шифрування

c) Команда відзиває всі привілеї у користувача M_Ivanenko.

3)

USE master

GO

-- Showing existing symmetric keys that was assigned for database

SELECT *

FROM sys.symmetric_keys;

GO

USE FilmLibraryDatabase

GO

-- creating of new master-key

CREATE MASTER KEY ENCRYPTION BY PASSWORD = '1qazZAQ!';

GO

-- creating of certificate with metadata field 'Protect Data'

CREATE CERTIFICATE Certificate1

WITH SUBJECT = 'Protect Data'

GO

-- creating symmetric key

CREATE SYMMETRIC KEY SymmetricKey1

WITH ALGORITHM = AES_128

ENCRYPTION BY CERTIFICATE Certificate1

GO

-- adding new column for future encrypted data

ALTER TABLE actors

ADD actor_name_encrypt varbinary(MAX) NULL

GO

-- encryption

OPEN SYMMETRIC KEY SymmetricKey1

DECRYPTION

BY CERTIFICATE Certificate1;

GO

UPDATE actors

SET actor_name_encrypt = EncryptByKey (Key_GUID('SymmetricKey1'), actors.name)

FROM actors;

GO

-- close encryption procedure

CLOSE SYMMETRIC KEY SymmetricKey1;

GO

--end encryption

-- decryption:

```
OPEN SYMMETRIC KEY SymmetricKey1  
DECRYPTION BY CERTIFICATE Certificate1  
GO
```

```
SELECT  
A.actor_id,  
A.actor_name_encrypt AS 'encrypted actor name',  
CONVERT(varchar, DecryptByKey(A.actor_name_encrypt)) AS 'decrypted actor name'  
FROM actors A
```

```
CLOSE SYMMETRIC KEY SymmetricKey1  
GO  
--end decryption
```

Висновок: Отже, я навчився працювати з системою безпеки у MS SQL Server. Також я створив користувачів John та Bill та надав їм привілеї та відняв ці привілеї. Також я навчився працювати з системою шифрування і розшифрування даних у MS SQL Server.