МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А. Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

“Информационная безопасность”

# Тема: “ Реализация доступа пользователей к базе данных”

Работу выполнил

Студент гр.4235

Тазеев Искандер,

Шафигуллин Рамиль

Преподаватель

Кожевников К. Д.

Казань 2024

**Цель:** Научиться реализовывать доступ пользователей к базе данных.

**Практическая часть**

**Шаг 1**

С помощью SQL Server Management Studio подключился к используемому экземпляру SQL Server. Проверил установленный на сервере режим аутентификации.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание**

Рисунок 2. – Соединение с сервером.

Зашел в свойства

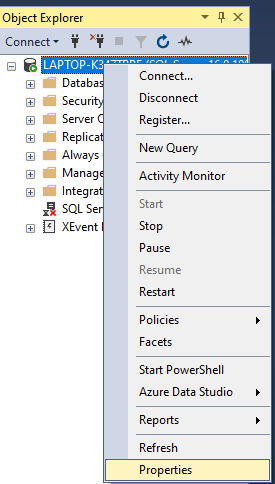
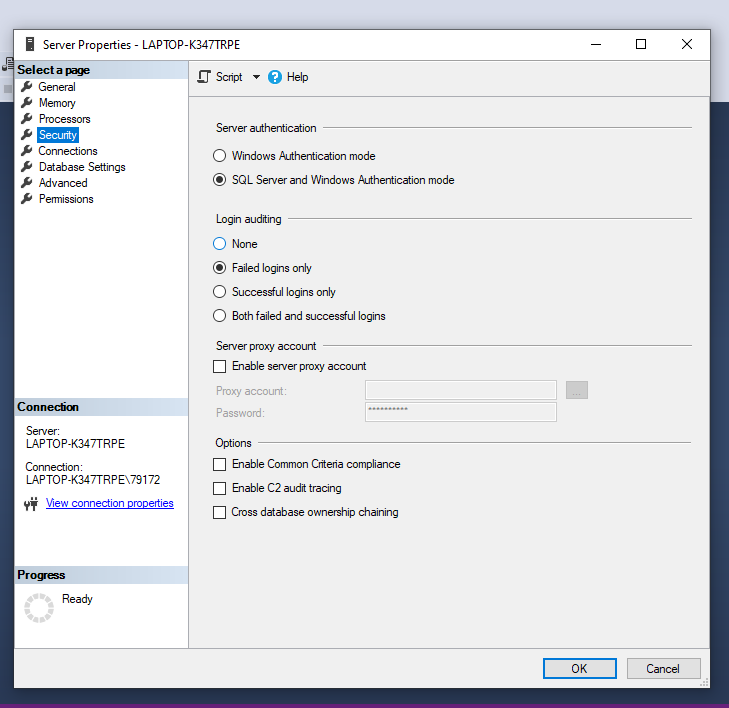
****

Рисунок 3. – Свойства.

Проверил какие настройки у нас установлены, в графе “Серверная проверка подлинности”. Изменил на проверку подлинности SQL Server и Windows

**Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание**



Рисунок 4. – Проверка подлинности.

**Шаг 2**

В окне Object Explorer нашел и развернул дерево объектов, чтобы увидеть различные папки и объекты, связанные с вашим сервером SQL.

Для этого выполнил следующие действия:

Нашел папку "Security" (Безопасность) в дереве объектов.

В папке "Security" разверниул подпапку "Logins" (Учетные записи).

Нашел вашу используемую учетную запись в списке учетных записей (logins).

Щелкнул правой кнопкой мыши на нашей учетной записи и выберал опцию "Properties" (Свойства).

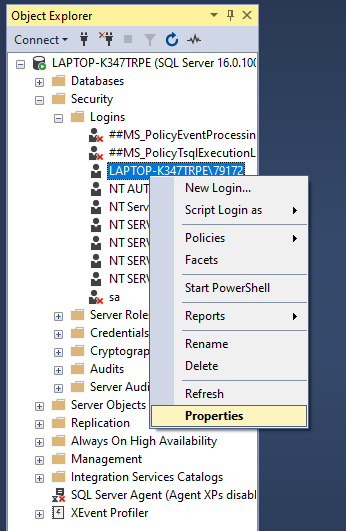


Рисунок 5. - Свойства

В открывшемся окне свойств учетной записи мы можем увидеть информацию о настройках и разрешениях нашей учетной записи.

Перешел на вкладку "Server Roles" (Роли сервера), чтобы увидеть, на выполнение каких серверных ролей авторизована наша учетная запись.

Здесь мы сможем увидеть список доступных серверных ролей (например, sysadmin, dbcreator, public и т.д.) и проверить, к каким из них относится наша учетная запись.

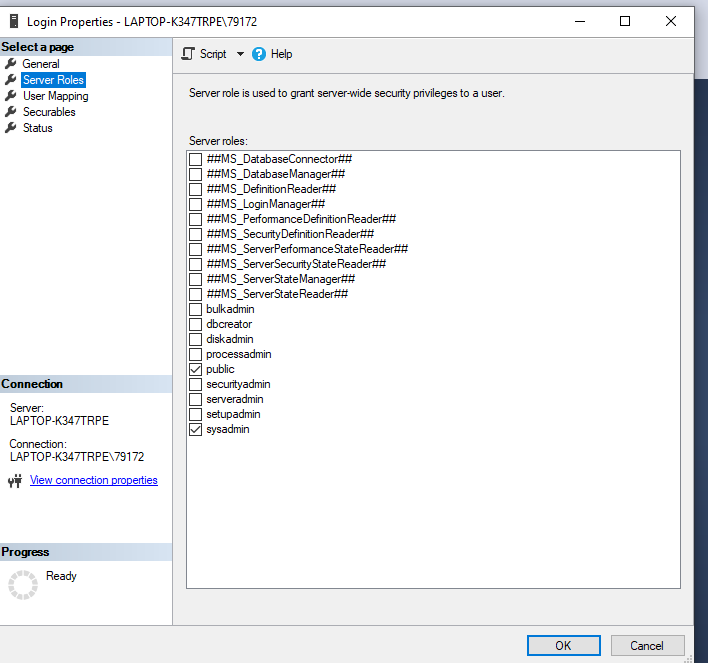
****

Рисунок 6. – Проверка роли сервера.

**Шаг 3**

В каких базах данных сервера вашей учетной записи сопоставлены пользователи? На выполнение каких ролей они авторизованы?

Нашли в окне Object Explorer ваш сервер SQL и раскрыли его для просмотра баз данных.

Развернули папку "Databases" (Базы данных), чтобы увидеть список всех баз данных, существующих на вашем сервере.

Нам нужно найти базы данных, в которых нашей учетной записи сопоставлены пользователи.

Для этого выполнили следующие действия:

Нашли базу данных, интересующую вас, в списке баз данных.

Развернули папку "Security" внутри выбранной базы данных.

Под папкой "Security" нашли папку "Users" (Пользователи). Здесь мы увидели список пользователей, сопоставленных с данной базой данных.

Нашли вашу учетную запись среди пользователей базы данных и проверили, на выполнение каких ролей она авторизована в рамках этой базы данных.

Чтобы узнать, на выполнение каких ролей пользователь авторизован в базе данных:

Выбрали нашего пользователя в списке.

Щелкнули правой кнопкой мыши на пользователе и выбрали опцию "Properties" (Свойства).

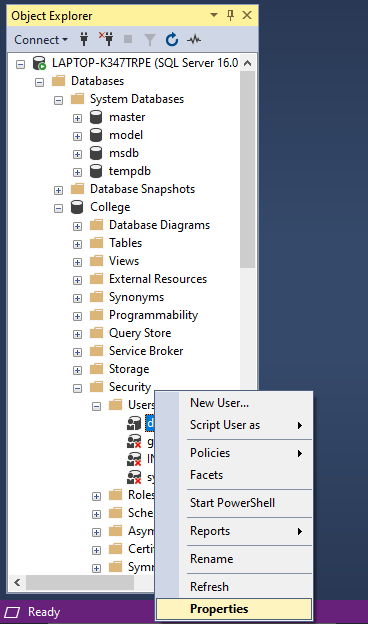


Рисунок 7. – Свойства.

Перешли на вкладку "Membership" (Членство), чтобы увидеть список ролей баз данных, к которым наш пользователь относится.

Ознакомились с информацией о ролях (например, db\_owner, db\_datareader, db\_datawriter и т.д.), к которым наш пользователь авторизован в данной базе данных.

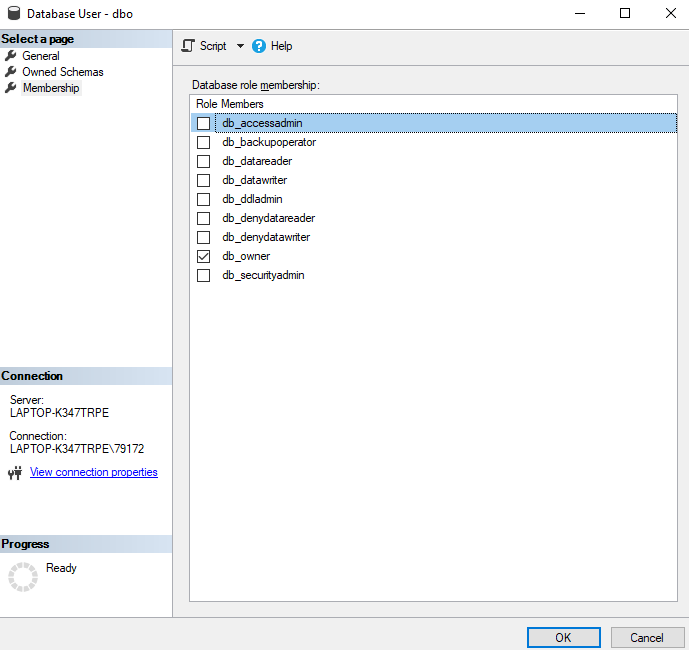


Рисунок 8. – Членство.

**Шаг 4**

В среде Management Studio создал новую базу данных. Открыл список пользователей и ролей. Убедился, что учетная запись, под которой я работаю, сопоставлена пользователю dbo, авторизованному на роль db owner.

Создание новой базы данных:

Щелкнул правой кнопкой мыши на папке "Databases" (Базы данных) в Object Explorer.

Выберал опцию "New Database" (Новая база данных).

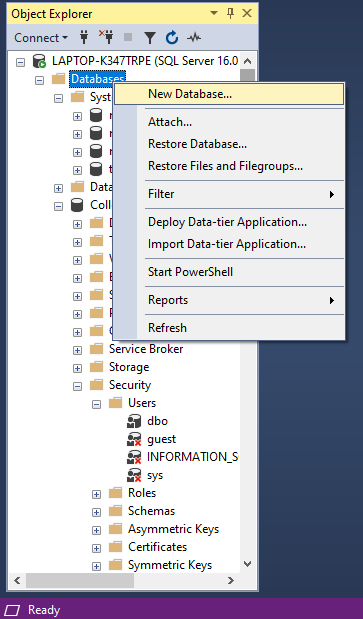
****

Рисунок 9. – Создание БД.

В появившемся диалоговом окне ввел имя для новой базы.

Нажал "OK", чтобы создать новую базу данных.

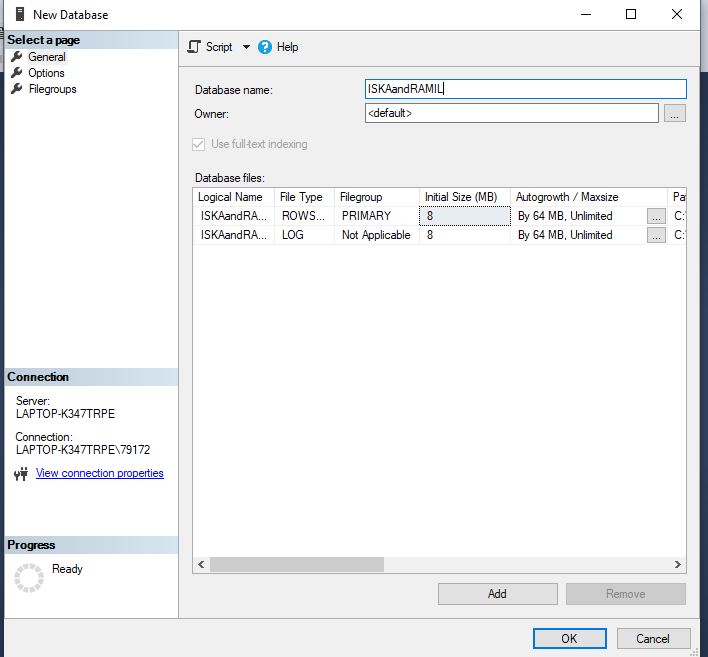
****

Рисунок 10. – Имя БД.

Проверка списка пользователей и ролей:

Развернул созданную нами новую базу данных в списке баз данных.

Найдшел папку "Security" внутри вашей новой базы данных и развернул ее.

Посмотрел список пользователей, сопоставленных с нашей новой базой данных.

Нашел пользователя "dbo" (Database Owner) среди списка пользователей.

Убедился, что наша учетная запись сопоставлена пользователю dbo, авторизованному на роль db owner:

Щелкните правой кнопкой мыши на пользователе "dbo" и выбрал опцию "Properties" (Свойства).

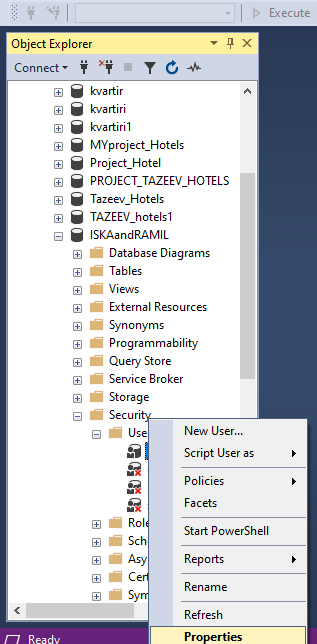
****

Рисунок 11. – Свойства dbo.

Перешел на вкладку "Membership" (Членство) в окне свойств пользователя "dbo".

Убедился, что роль "db\_owner" присутствует в списке ролей, к которым этот пользователь авторизован.

Если роль "db\_owner" отсутствует, то сделал пользователя "dbo" членом роли "db\_owner".

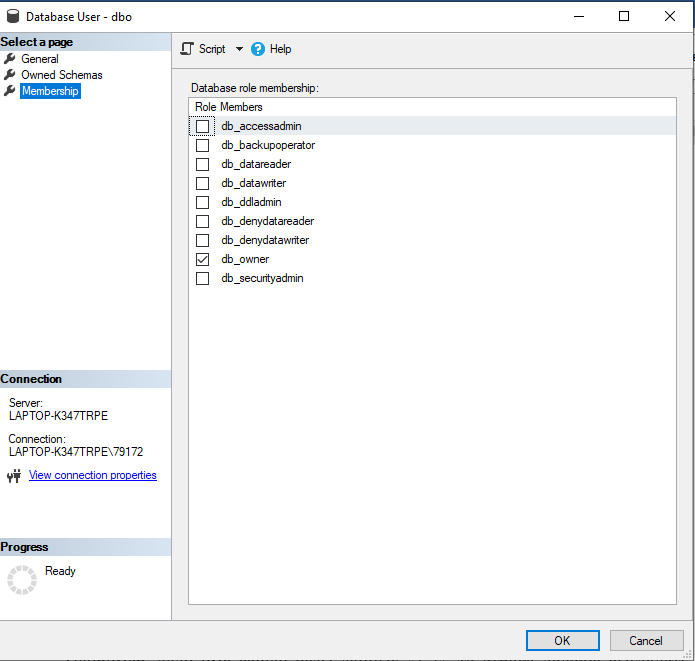
****

Рисунок 12. – Членство.

**Шаг 5.**

Используя приведенный ниже скрипт, создал в базе данных таблицы

После ключевого слова "use" указал имя нашей базы данных.

use ISKAandRAMIL

GO

CREATE TABLE dbo.Book (

book\_id int IDENTITY (1, 1) primary key,

Title varchar(50) NOT NULL,

Author varchar(50),

Publisher varchar(50),

[Year] smallint

)

GO

CREATE TABLE dbo.Status (

Status\_id int IDENTITY (1, 1) primary key,

Status\_name varchar(50) NOT NULL

)

GO

CREATE SCHEMA libr

GO

CREATE TABLE libr.Book\_in\_lib (

lib\_id int primary key,

book\_id int references dbo.Book,

status\_id int references dbo.[Status]

)

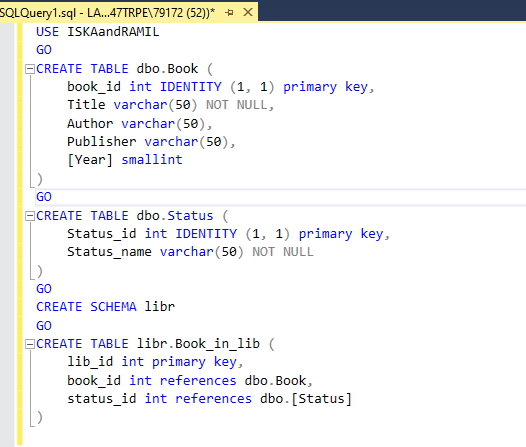


Рисунок 13. – Создание таблиц.

После успешного выполнения должны быть созданы таблицы: "Book", "Status", "Book\_in\_lib" и схема "libr".

Проверил, что таблицы созданы в нашей базе данных.

Мы можем посмотреть структуру созданных таблиц, а также связи между ними.

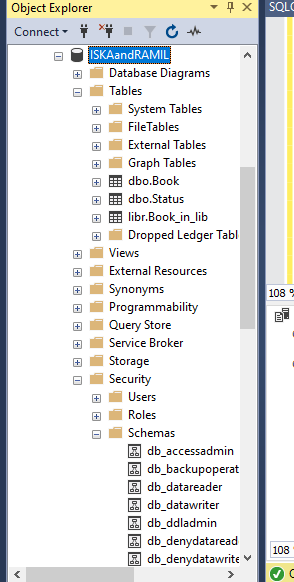


Рисунок 14. – Проверка.

**Шаг 6.**

Создание пользователя, выбор схемы dbo, добавление в роль db\_datareader:

Для создания пользователя, выбора схемы по умолчанию и добавления в роль db\_datareader, нужно выполнить следующий скрипт.

USE ISKAandRAMIL;

-- Создание логина с указанием пароля

CREATE LOGIN ISKAandRAMIL WITH PASSWORD = '1111';

-- Создание пользователя и назначение схемы dbo

CREATE USER ISKAandRAMIL FOR LOGIN ISKAandRAMIL WITH DEFAULT\_SCHEMA = dbo;

-- Добавление пользователя в роль db\_datareader

EXEC sp\_addrolemember 'db\_datareader', 'ISKAandRAMIL';

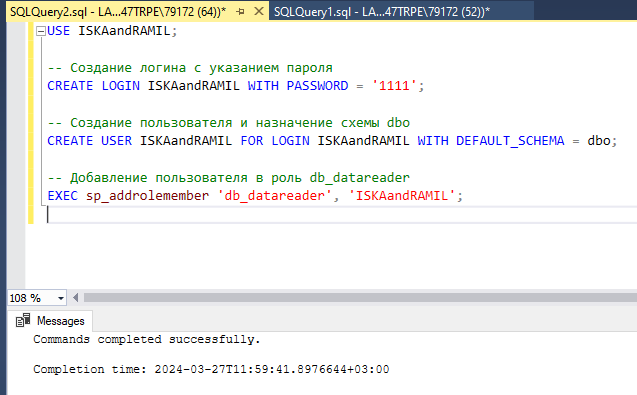
****

Рисунок 15. – Скрипт.

Тестирование:

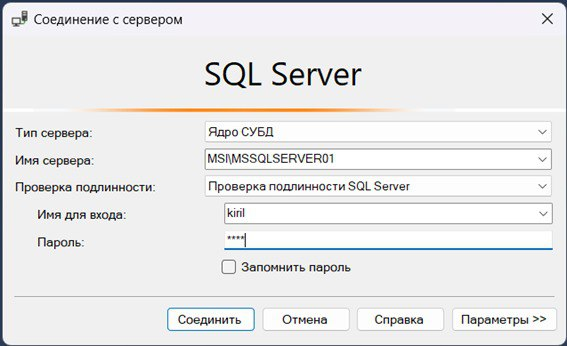


Рисунок 16. - Тестирование

Подключился к серверу с учетной записью другого пользователя.

Убедился, что новый пользователь имеет доступ только на чтение данных из всех таблиц базы данных.

Попробовал добавить или изменить данные в таблицах – у нас не должно быть соответствующих прав для этого пользователя.

**Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание**

Рисунок 17. – Проверка со стороны пользователя.

**Шаг 7.**

Создание новой роли и назначение разрешений:

Подключился к серверу базы данных:

Использовал административные учетные данные для подключения к серверу баз данных. Убедился, что мы подключены к той же базе данных, в которой создан пользователь "ISKAandRAMIL", или наше имя.

Создал новую роль уровня базы данных: Открыл новый запрос T-SQL и ввели следующий запрос:

CREATE ROLE libr\_writer;

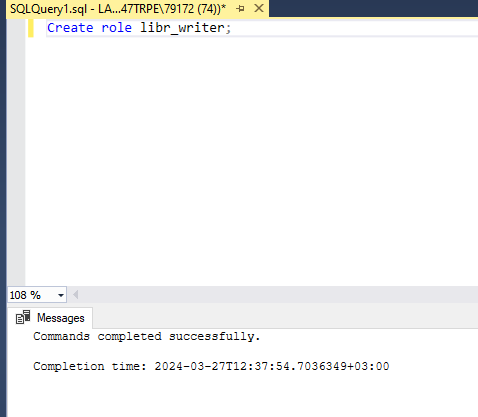


Рисунок 18. – Новая роль.

Этот запрос создаст новую роль с именем "libr\_writer".

Назначил разрешения роли для схемы "libr":

Ввели следующий запрос, чтобы предоставить разрешения роли "libr\_writer" на выполнение операций INSERT, UPDATE и DELETE для объектов в схеме "libr":

GRANT INSERT, UPDATE, DELETE ON SCHEMA::libr TO libr\_writer

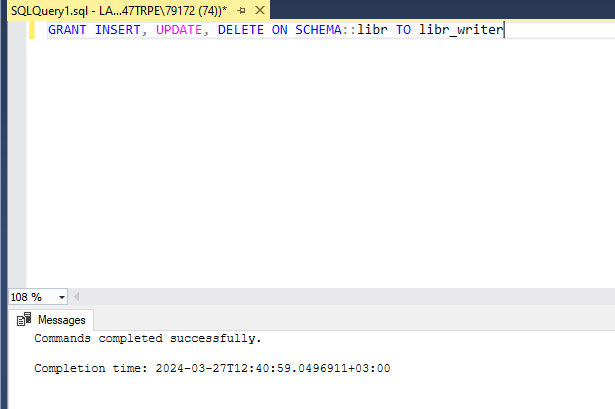


Рисунок 19. – Разращение роли.

Добавил пользователя в созданную роль:

Ввели следующий запрос, чтобы добавить пользователя "ISKAandRAMIL" в роль "libr\_writer":

Этот запрос назначает пользователя "ISKAandRAMIL" в роль "libr\_writer", что дает ему возможность изменять данные в объектах схемы "libr".

EXEC sp\_addrolemember 'libr\_writer', 'ISKAandRAMIL';

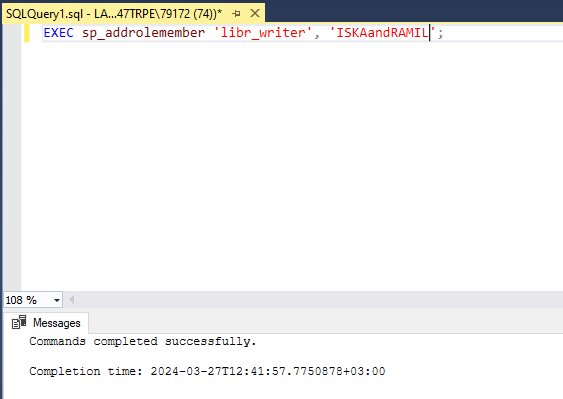


Рисунок 20. – Назначение роли.

Проверил доступ пользователя к таблице:

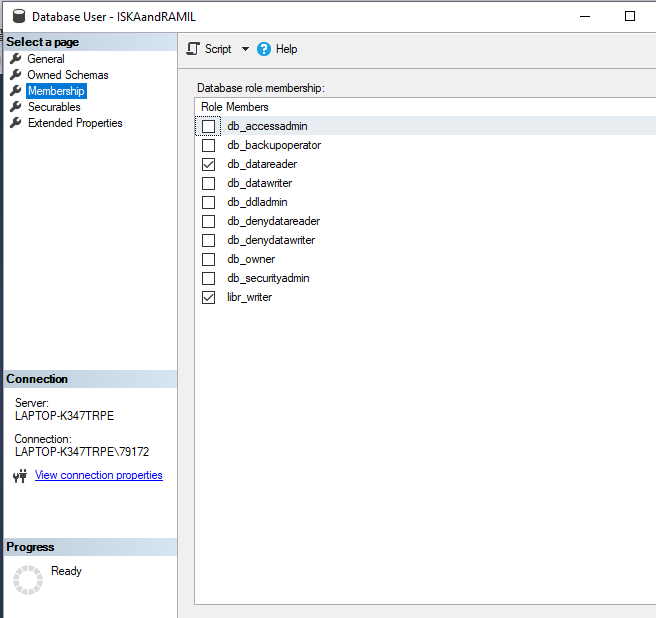


Рисунок 20 – проверка роли

Подключился к серверу базы данных с учетной записью пользователя "kiril".

Выполнил операции INSERT, UPDATE и DELETE для данных в таблице "Book\_in\_lib", принадлежащей схеме "libr". Зашел в свойства.

**Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание**

Рисунок 21. – Свойства.

Назначил разрешения для пользователя “ISKAandRAMIL” как на рисунке 22.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание**

Рисунок 22. – Настройка разращений.

**Шаг 8.**

Иногда нужно предоставить пользователю права на изменение отдельных столбцов. Как отмечается в документации SQL Server, на столбец могут быть предоставлены только разрешения SELECT, REFERENCES и UPDATE. Например: GRANT UPDATE ON dbo.Book(Title) TO libr\_writer Выполнил аналогичные действия в своей базе данных, проверил, что пользователь получил указанные разрешения.

Открыл новый запрос:

В меню навигации выбрал нашу базу данных, щелкнул правой кнопкой мыши и выбрал "New Query" (Новый запрос).

Выполнил запрос на предоставление прав:

В открытом запросе введили следующий запрос, чтобы предоставить пользователю "ISKAandRAMIL" право на обновление столбца "Title" в таблице "Book":

GRANT UPDATE ON dbo.Book(Title) TO ISKAandRAMIL;

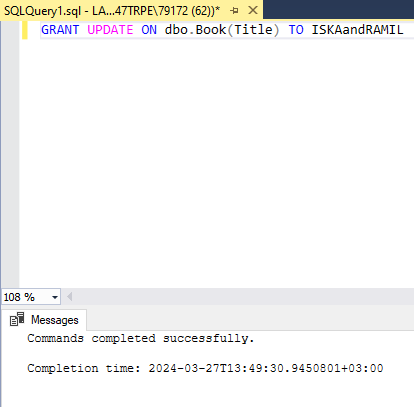


Рисунок 23. – Предоставление право.

После выполнения запроса попробуйте выполнить операцию обновления значения в столбце "Title" таблицы "Book".

UPDATE dbo.Book SET Title = 'Новое название' WHERE book\_id = 1;

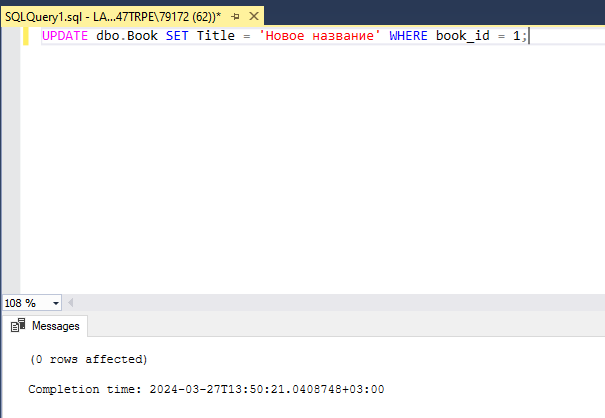


Рисунок 24. – Обновление столбцов

Проверил результат:

Если запрос обновления выполнится успешно без ошибок доступа, это означает, что пользователь "ISKAandRAMIL" успешно получил право на изменение столбца "Title".

**Шаг 9.**

Создал представления и предоставление прав на изменение и добавление записей:

Изучение формата оператора CREATE VIEW:

Первым делом, изучил формат оператора CREATE VIEW и особенности его использования. Обратил внимание на синтаксис и дополнительные параметры, такие как параметры безопасности.

Создание представления:

В открытом запросе выполнил оператор CREATE VIEW для создания представления. Выберал из таблицы "Book" только те книги, которые были изданы не ранее 2000 года. Пример создания представления:

CREATE VIEW dbo.NewBooks AS

SELECT \*

FROM dbo.Book

WHERE [Year] >= 2000;

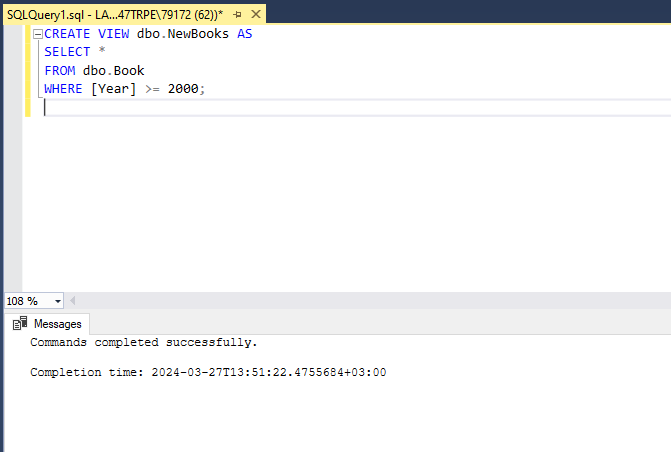


Рисунок 25. - Создания представления.

Предоставление прав на изменение и добавление записей:

Теперь предоставил пользователю "ISKAandRAMIL" права на изменение и добавление записей в представление "NewBooks", но только для книг, изданных не ранее 2000 года.

GRANT INSERT, UPDATE ON dbo.NewBooks TO ISKAandRAMIL;

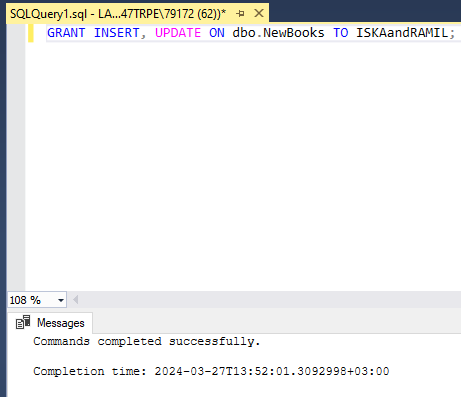


Рисунок 26. – Права на изменение записей.

Проверка прав пользователя:

Убедился, что пользователь "ISKAandRAMIL" имеет права на изменение и добавление записей в представление "NewBooks". Это можно проверить, выполнив следующий запрос:

SELECT HAS\_PERMS\_BY\_NAME('dbo.NewBooks', 'OBJECT', 'INSERT') AS can\_insert,

HAS\_PERMS\_BY\_NAME('dbo.NewBooks', 'OBJECT', 'UPDATE') AS can\_update;

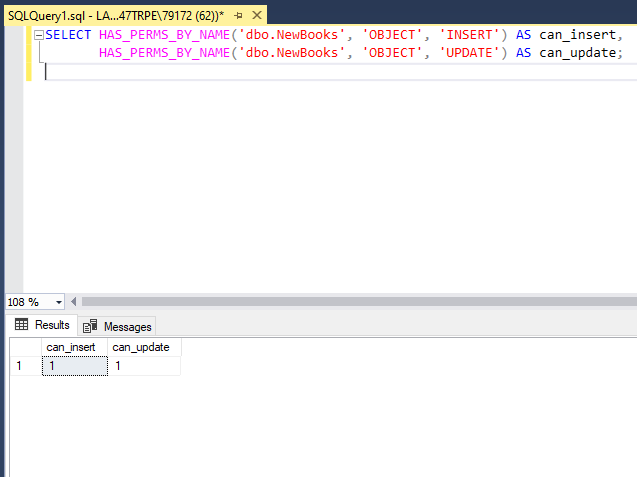


Рисунок 27. – Проверка прав пользователя.

Если для обеих операций выводится значение "1", это означает, что пользователь "ISKAandRAMIL" имеет соответствующие права.

**Вывод:** Научился реализовывать доступ пользователей к базе данных.