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 выполнили

Студенты гр.4235

Шапиров А.Г., Хасанзянов Ф.И.

Преподаватель

Кожевников К. Д.

Казань 2024

**Цель работы**

Научиться реализовывать доступ пользователей к базе данных.

**Практическая часть**

**Шаг 1**

С помощью SQL Server Management Studio подключимся к используемому экземпляру SQL Server.

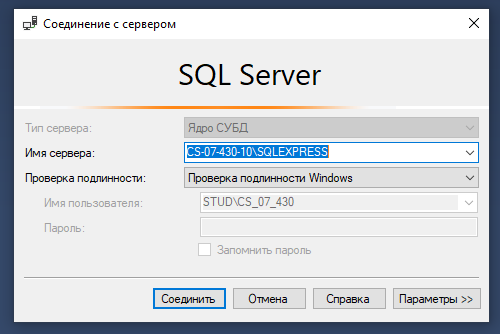


Рисунок 1 – Соединение с сервером.

Заходим в свойства

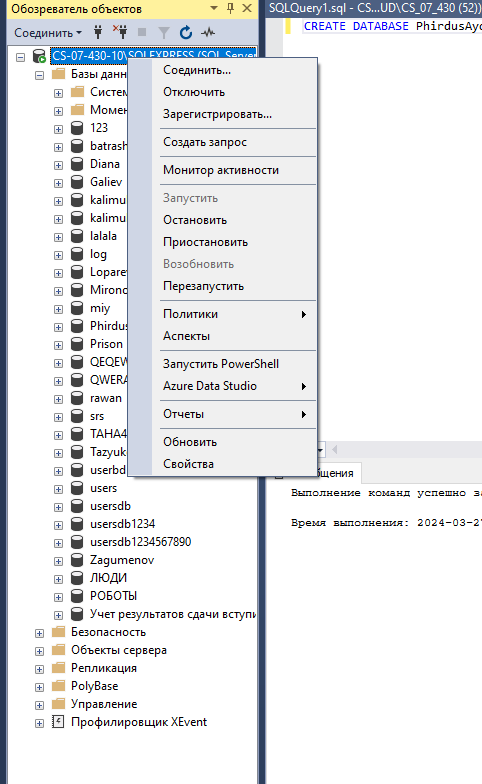


Рисунок 2 – Свойства.

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

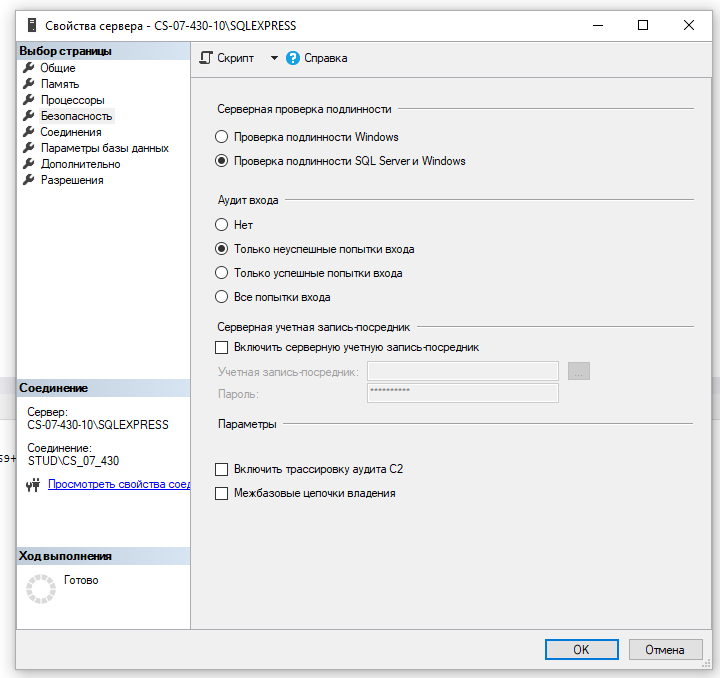


Рисунок 3 – Проверка подлинности.



**Шаг 2**



В окне Object Explorer найдём и развернём дерево объектов, чтобы увидеть различные папки и объекты, связанные с нашим сервером SQL.

Для этого выполним следующие действия:

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

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

Найдём нашу используемую учетную запись в списке учетных записей (logins).

Щелкнем правой кнопкой мыши на нашу учетную запись и выберем опцию "Properties" (Свойства).

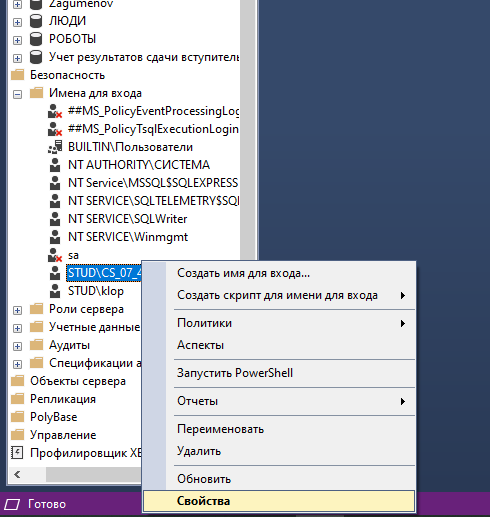


Рисунок 4 - Свойства

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

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

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

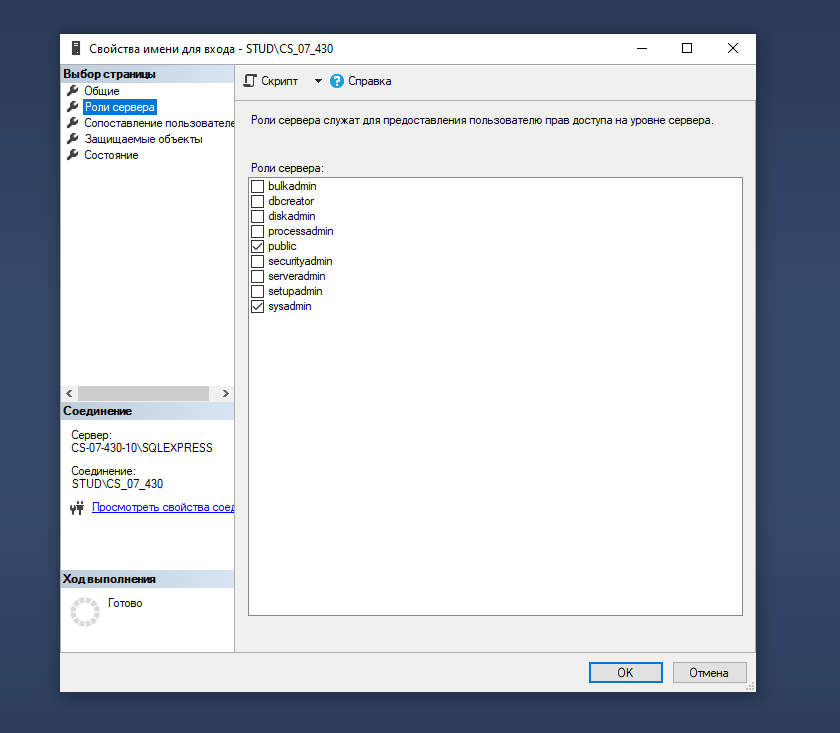
****

Рисунок 5 – Проверка роли сервера.

**Шаг 3**

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

Найдём в окне Object Explorer наш сервер SQL и раскроем его для просмотра баз данных.

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

Нам нужно найти базы данных, в которых наша учетная запись сопоставлена пользователям.

Для этого выполним следующие действия:

Найдём базу данных, интересующую нас, в списке баз данных.

Развернём папку "Security" внутри выбранной базы данных.

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

Найдём нашу учетную запись среди пользователей базы данных и проверим, на выполнение каких ролей она авторизована в рамках этой базы данных.

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

Выберем вашего пользователя в списке.

Щелкнем правой кнопкой мыши на пользователе и выберем опцию "Properties" (Свойства).

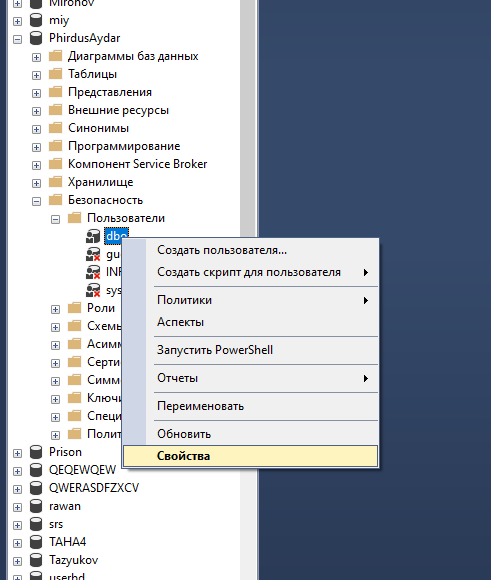


Рисунок 6– Свойства.

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

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

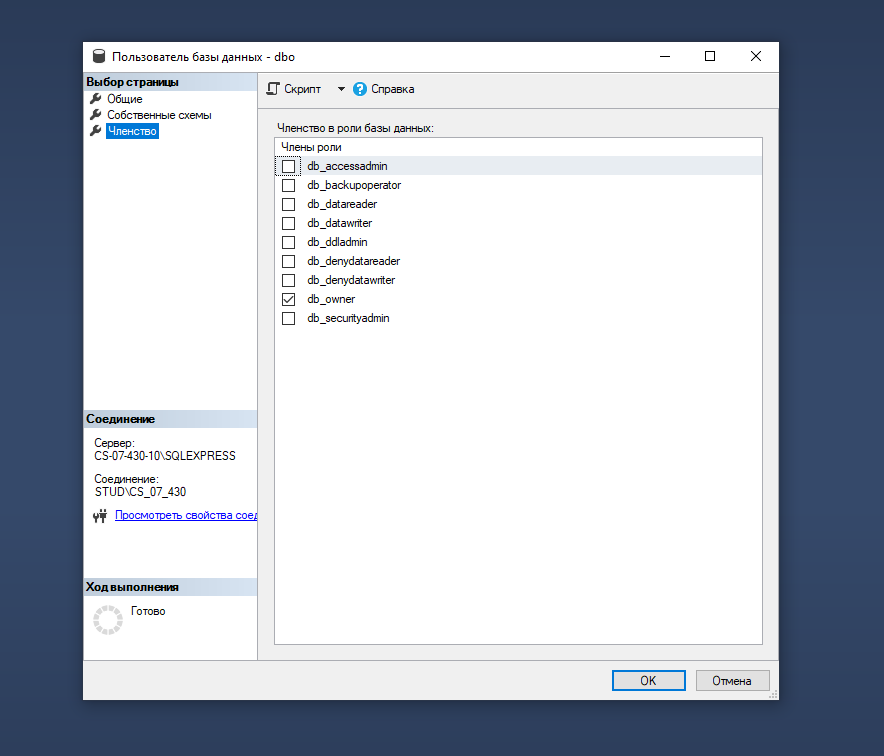


Рисунок 7 – Членство.

**Шаг 4**

В среде Management Studio создаём новую базу данных. Откроем список пользователей и ролей. Убедимся, что учетная запись, под которой мы работаем, сопоставлена пользователю dbo, авторизованному на роль db owner.

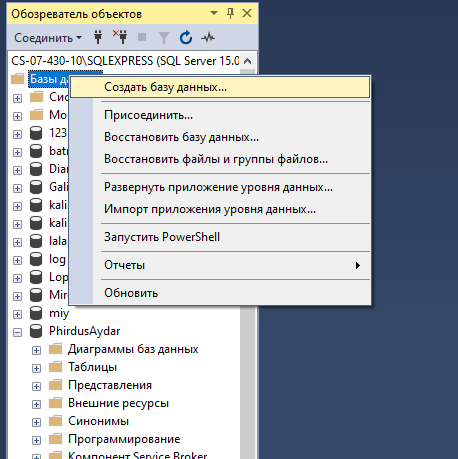
****

Рисунок 8 – Создание БД.

В появившемся диалоговом окне введём имя для новой базы.

Нажмём "OK", чтобы создать новую базу данных.

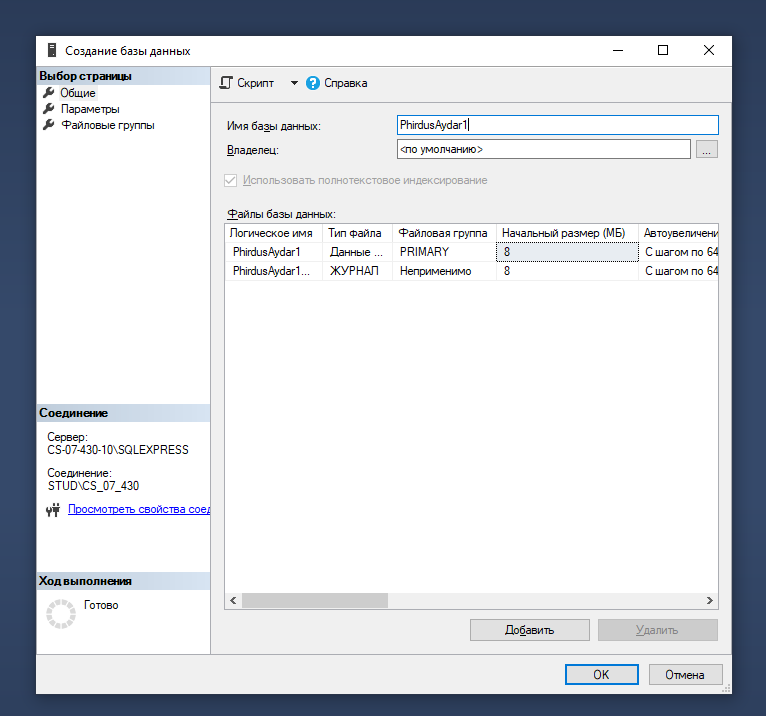
****

Рисунок 9 – Имя БД.

Проверка списка пользователей и ролей:

Развернём созданную нами новую базу данных в списке баз данных.

Найдём папку "Security" внутри нашей новой базы данных и развернём ее.

Посмотрим список пользователей, сопоставленных с нашей новой базой данных.

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

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

Щелкнем правой кнопкой мыши по пользователю "dbo" и выберем опцию "Properties" (Свойства).

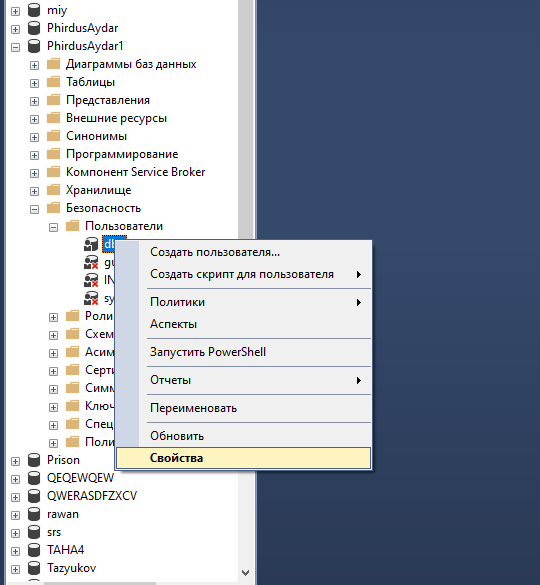
****

Рисунок 10 – Свойства dbo.

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

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

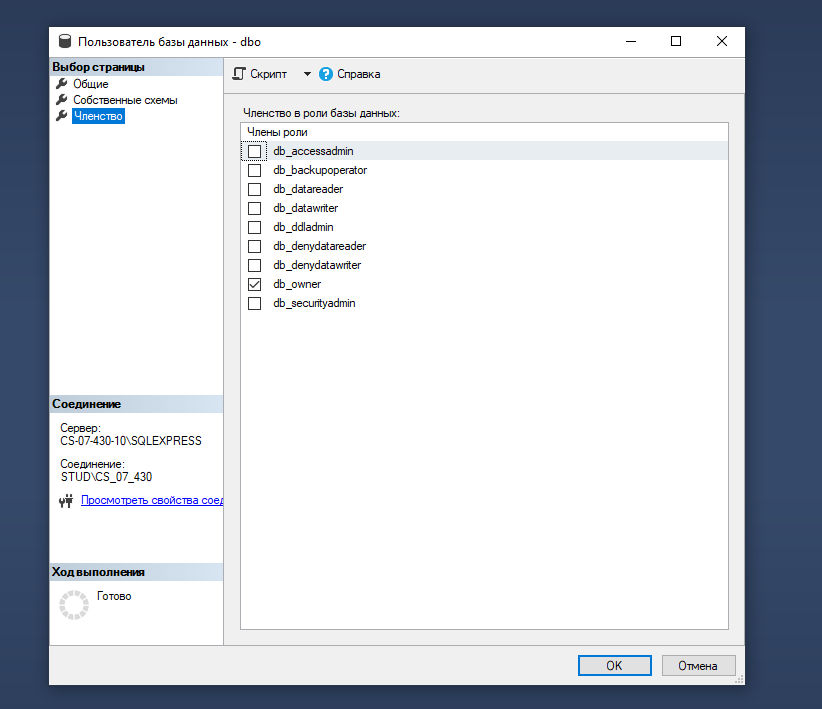
****

Рисунок 11– Членство.

**Шаг 5.**

Используя приведенный ниже скрипт, создаём в базе данных таблицы

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

use PhirdusAydar1;

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]

)

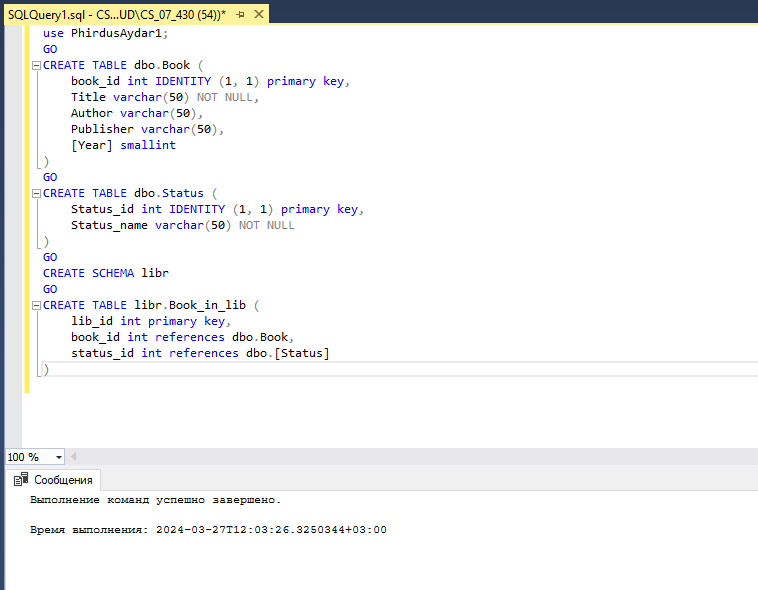


Рисунок 12 – Создание таблиц.

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

Проверим, что таблицы созданы в нашей базе данных.

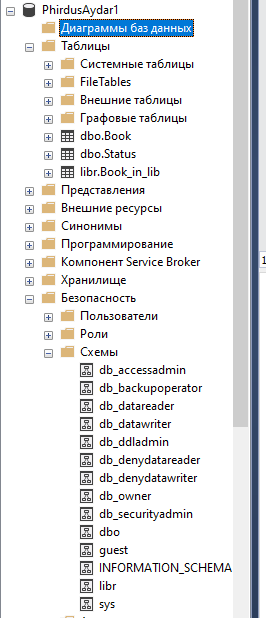


Рисунок 13 – Проверка.

**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, выполним следующий скрипт.

use PhirdusAydar1;

-- Создание логина с указанием пароля

CREATE LOGIN Firdus WITH PASSWORD = '11111111';

-- Создание пользователя и назначение схемы dbo

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

-- Добавление пользователя в роль db\_datareader

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

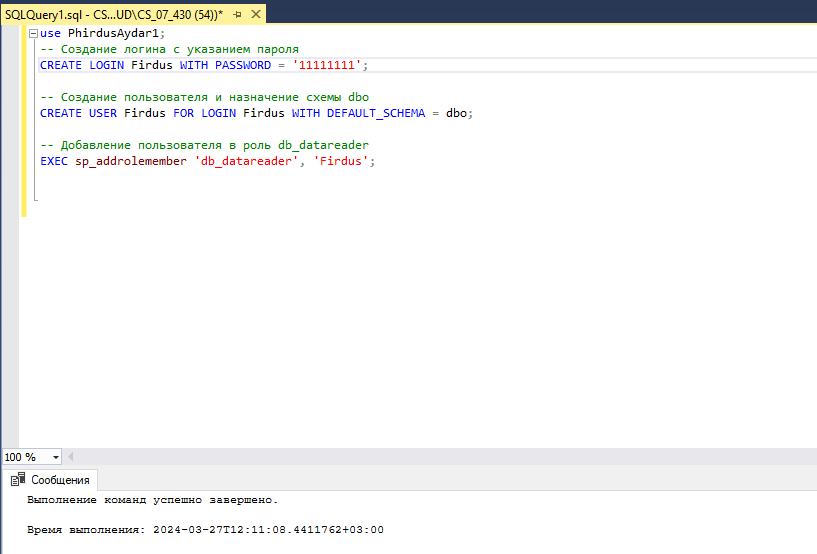
****

Рисунок 14 – Скрипт.

**Шаг 7.**

Создание новой роли и назначение разрешений:

Подключимся к серверу базы данных:

Используем административные учетные данные для подключения к серверу баз данных. Убедимся, что мы подключены к той же базе данных, в которой создан пользователь "Firdus".

Создаем новую роль уровня базы данных: Откроем новый запрос T-SQL и введём следующий запрос:

CREATE ROLE libr\_writer;

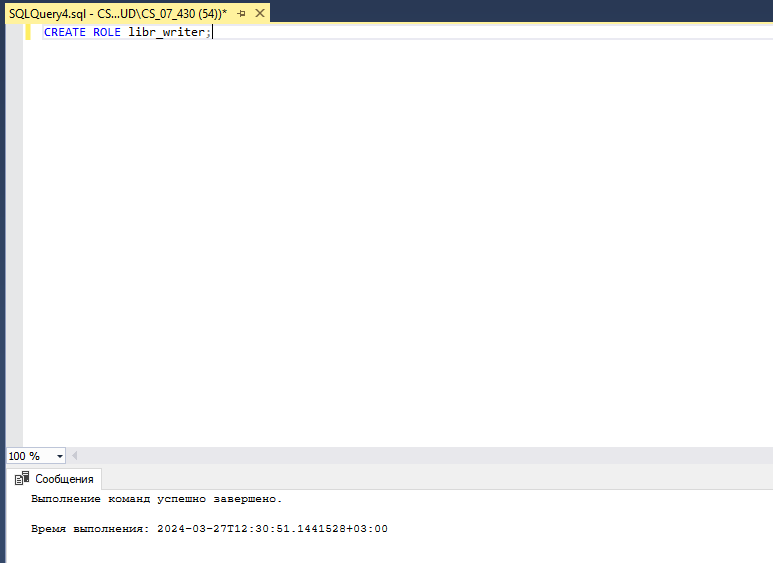


Рисунок 16 – Новая роль.

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

Назначим разрешения роли для схемы "libr":

Введём 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

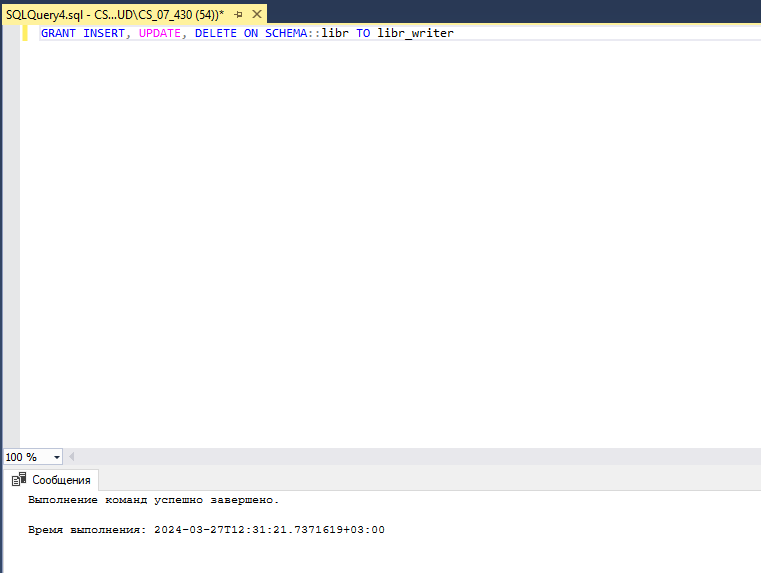


Рисунок 17 – Разращение роли.

Добавим пользователя в созданную роль:

Введём следующий запрос, чтобы добавить пользователя "Firdus" в роль "libr\_writer":

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

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

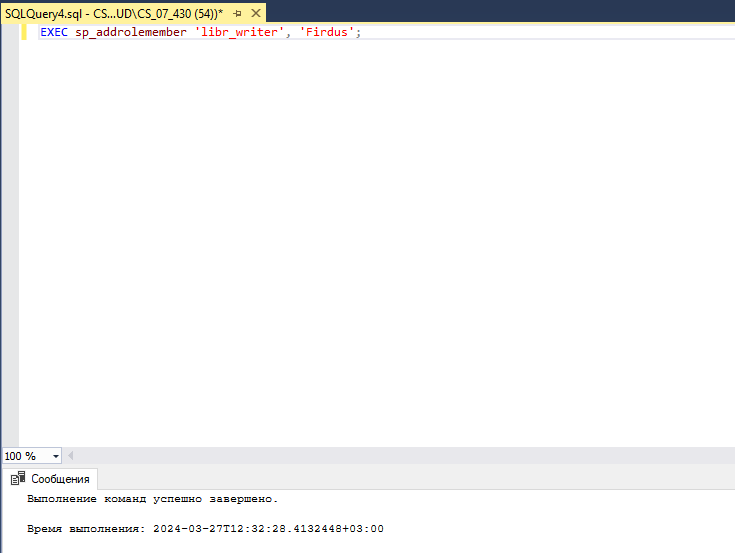


Рисунок 18 – Назначение роли.

Проверим доступ пользователя к таблице:

Подключимся к серверу базы данных с учетной записью пользователя "Firdus".

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

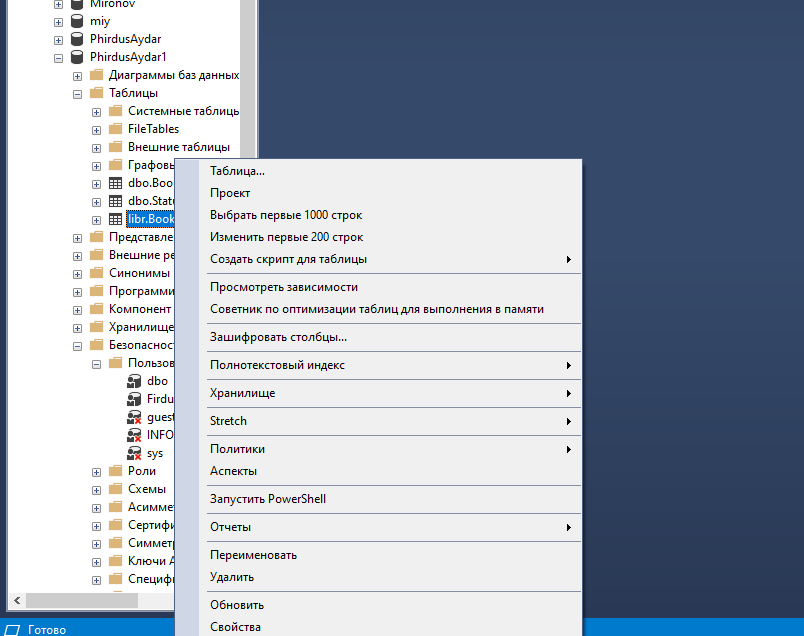


Рисунок 19– Свойства.

Назначим разрешения для пользователя “Firdus” как на рисунке 20.

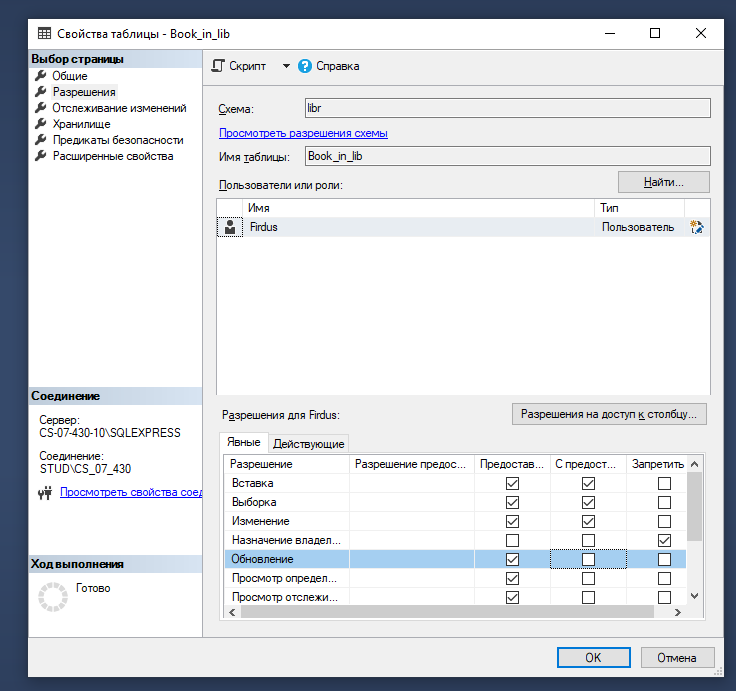


Рисунок 20– Настройка разращений.

**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 Выполним аналогичные действия в своей базе данных, проверим, что пользователь получил указанные разрешения.

Откроем новый запрос:

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

Выполним запрос на предоставление прав:

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

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

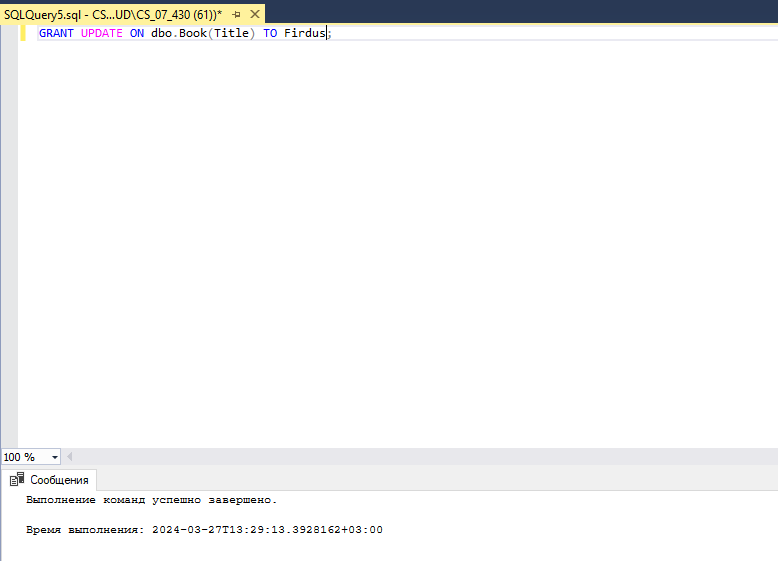


Рисунок 21 – Предоставление право.

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

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

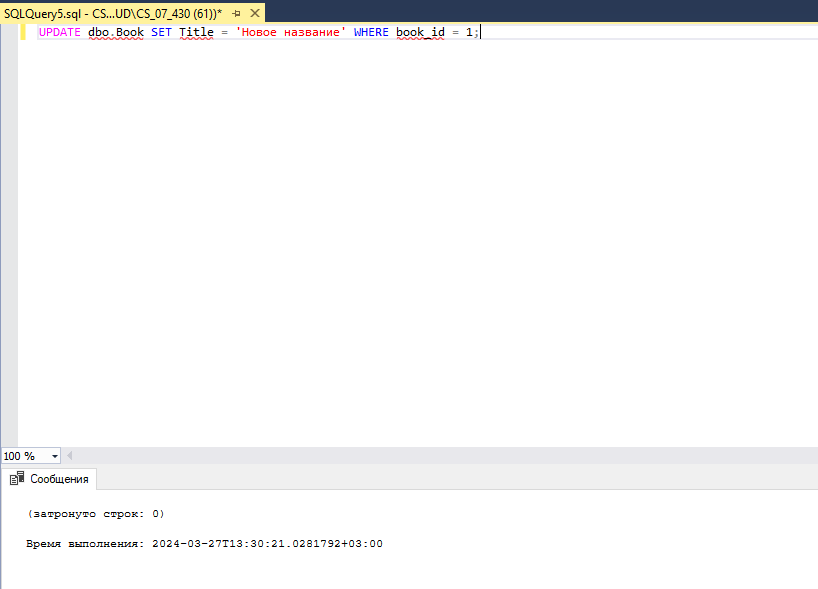


Рисунок 22 – Обновление столбцов

**Шаг 9.**

Создание представления и предоставление прав на изменение и добавление записей:

Изучение формата оператора CREATE VIEW:

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

Создание представления:

В открытом запросе выполним оператор CREATE VIEW для создания представления. Выберите 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;

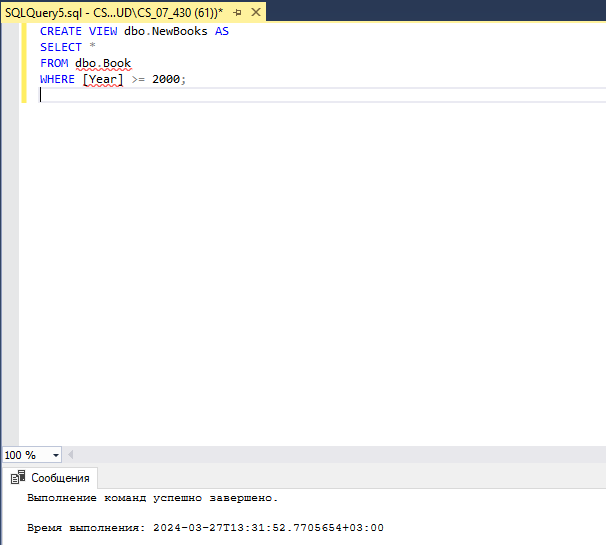


Рисунок 23 - Создания представления.

Предоставление прав на изменение и добавление записей:

Теперь предоставим пользователю "Firdus" 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 Firdus;

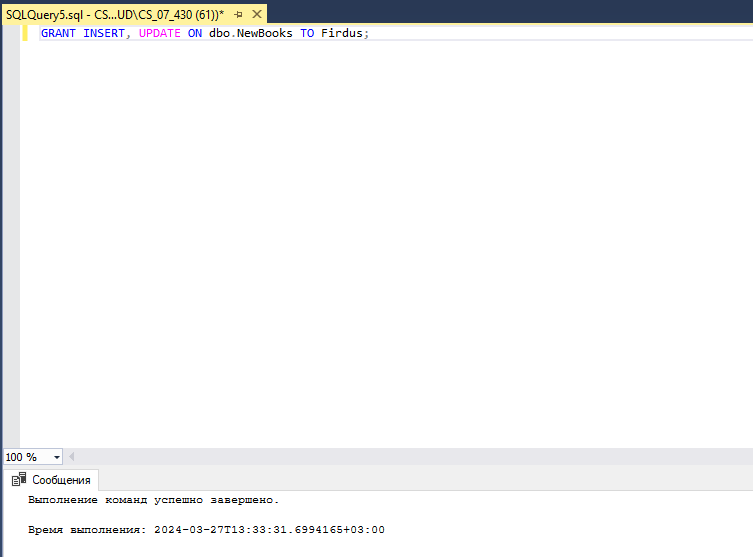


Рисунок 24 – Права на изменение записей.

Проверка прав пользователя:

Убедимся, что пользователь "Firdus" 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;

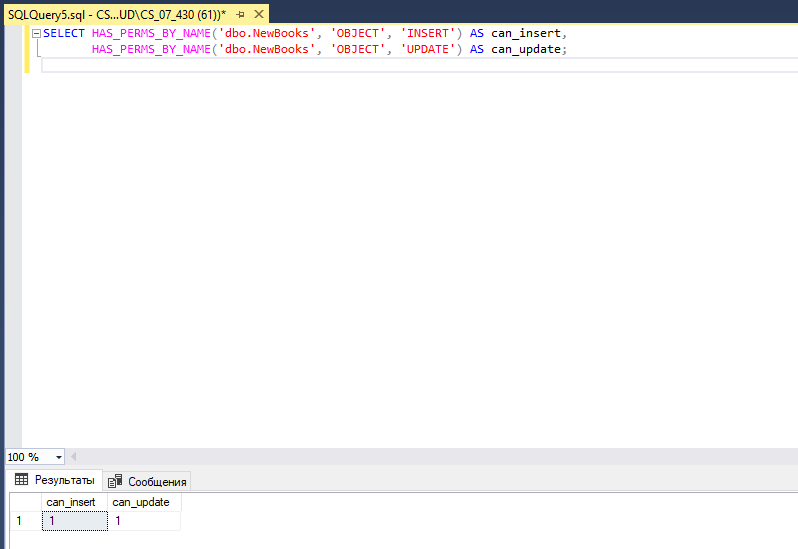


Рисунок 25 – Проверка прав пользователя.

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

**Вывод**: после проделанной лабораторной работы мы научились реализовывать доступ пользователей к базе данных.