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.4237

Фурсова А. М.

Преподаватель

Кожевников К. Д.

Казань 2024

**Цель** – Научиться реализации доступа пользователей к базе данных.

**Задачи:**

* изучить теоретическую часть;
* выполнить практические указания;
* составить отчет по лабораторной работе.

**Практическая часть**

**Шаг 1**

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

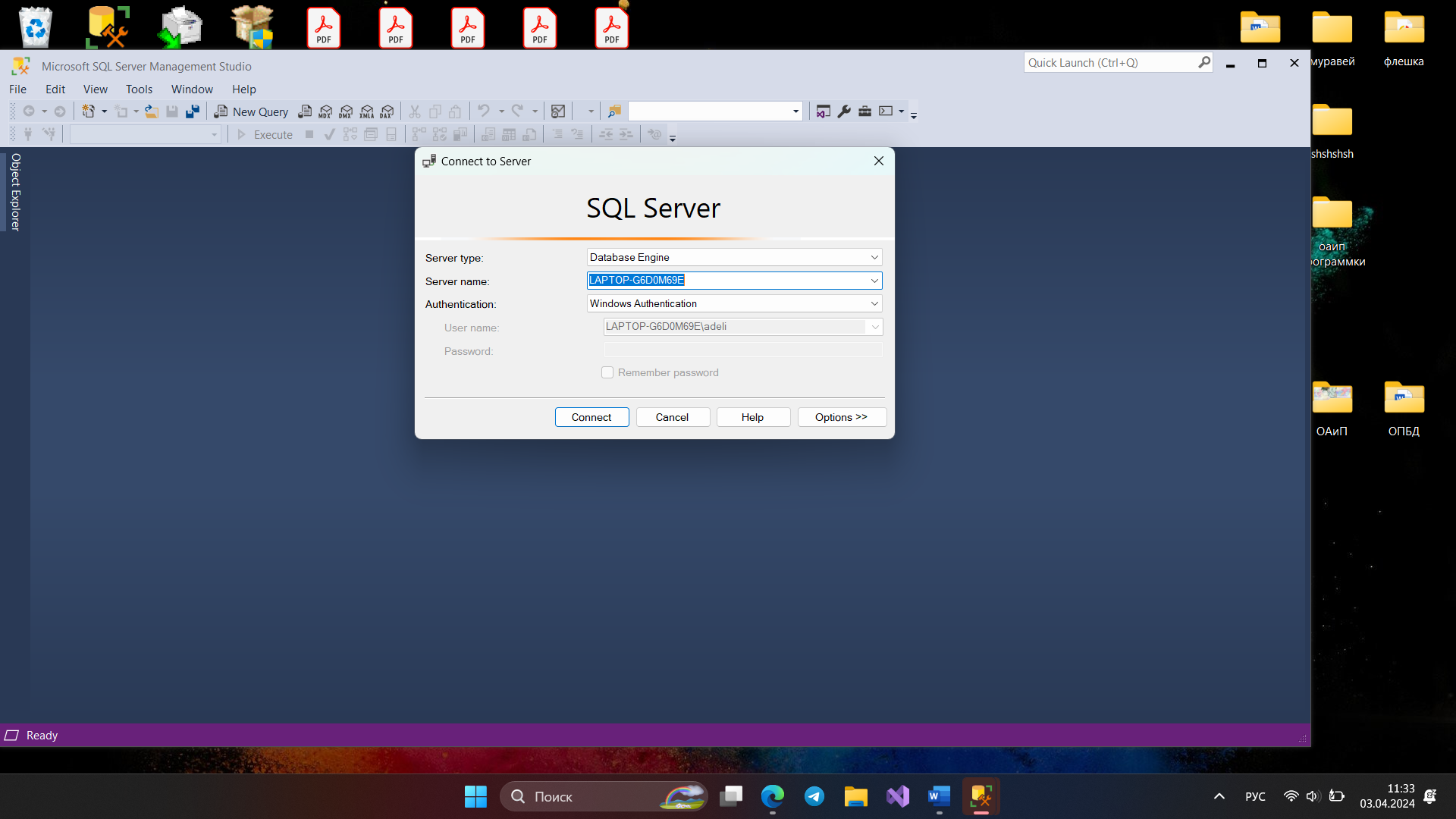


Рисунок 1. – Соединение с сервером.

Зашли в свойства.

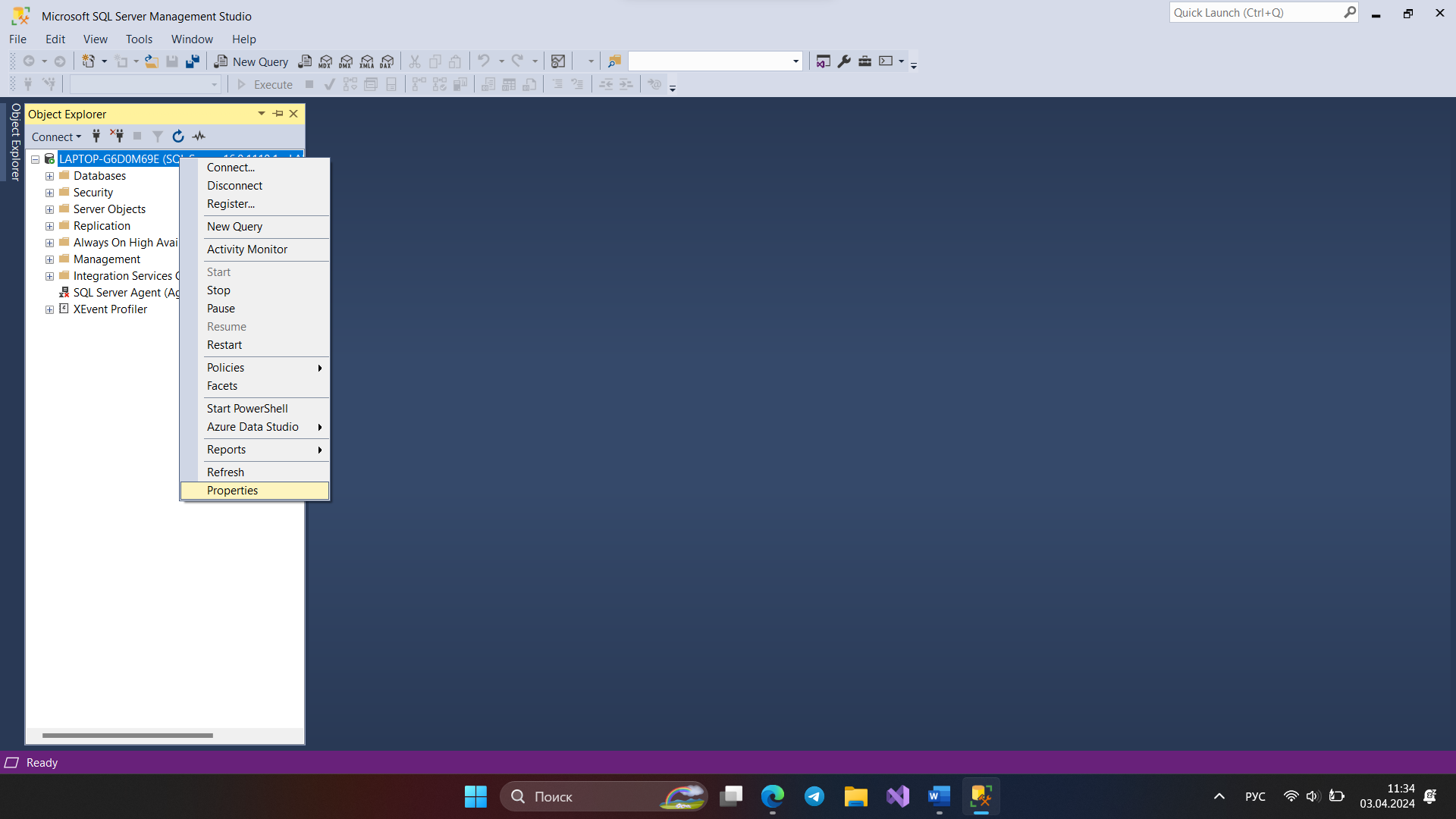


Рисунок 2. – Свойства.

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

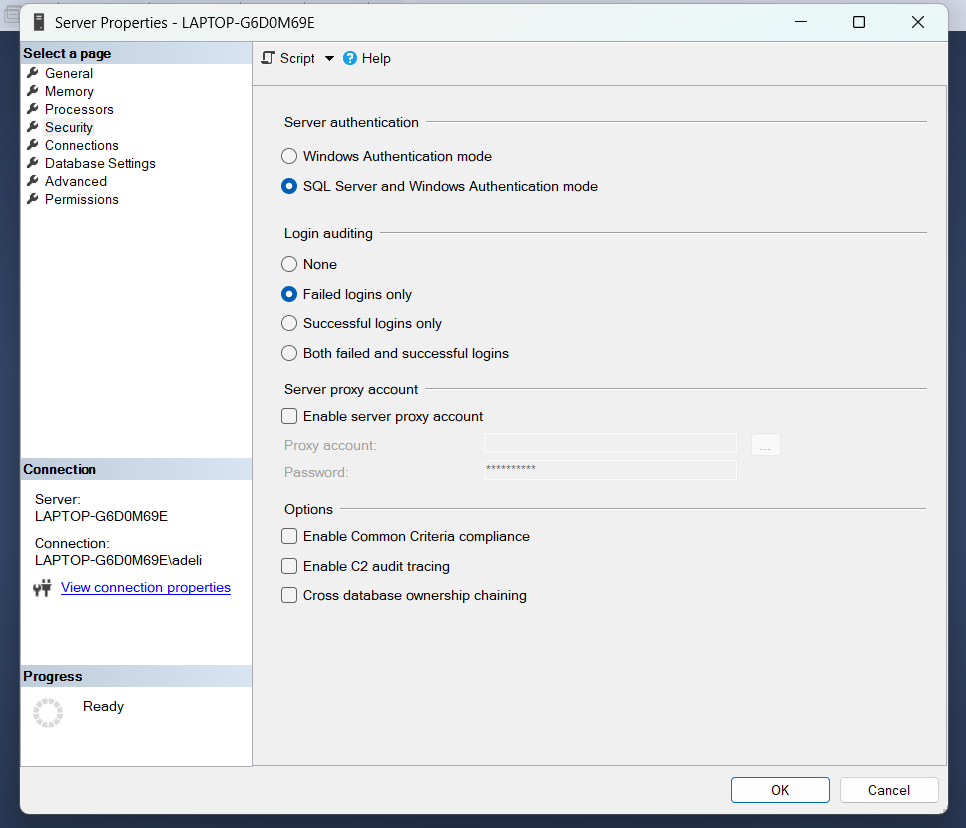


Рисунок 3. – Проверка подлинности.

**Шаг 2**

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

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

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

Щелкнули правой кнопкой мыши на нашу учетной записи и выбрали опцию "Properties" (Свойства).

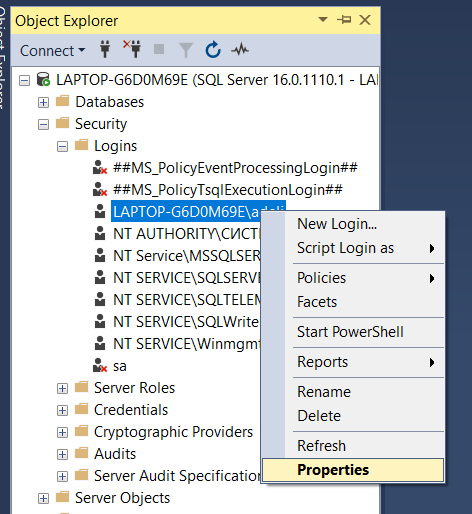


Рисунок 4. - Свойства

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

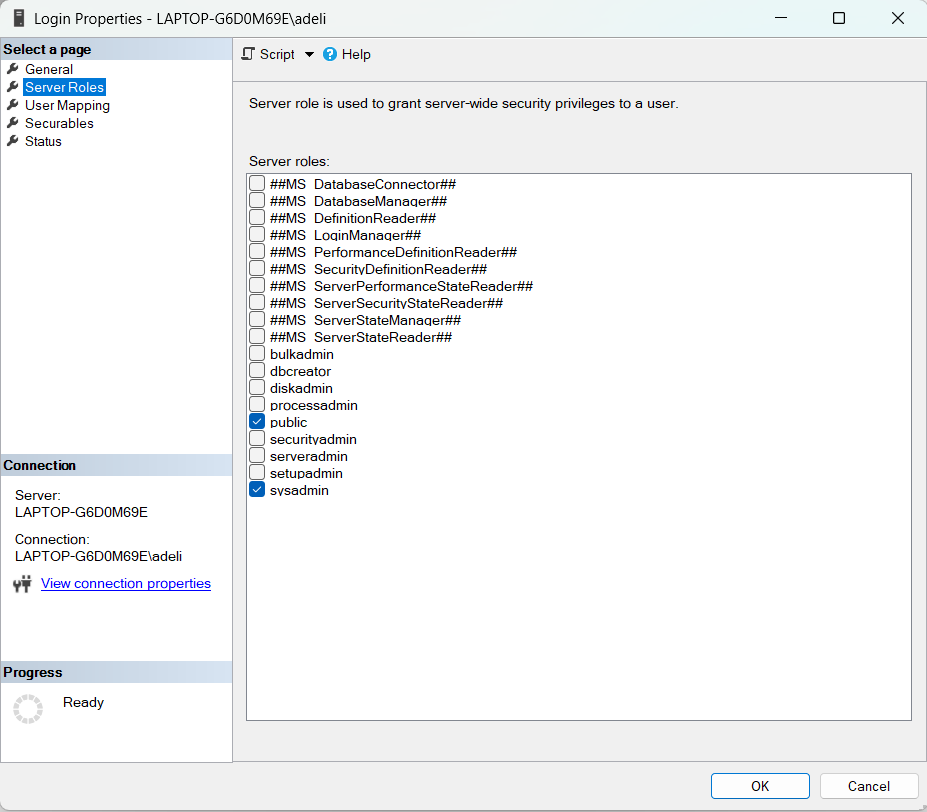
****

Рисунок 5. – Проверка роли сервера.

**Шаг 3**

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

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

Развернули папку "Security" внутри выбранной базы данных.

Под папкой "Security" нашли папку "Users" (Пользователи).

Выбрали нашего пользователя в списке.

Щелкнли правой кнопкой мыши на пользователя и выбрали опцию "Properties" (Свойства).

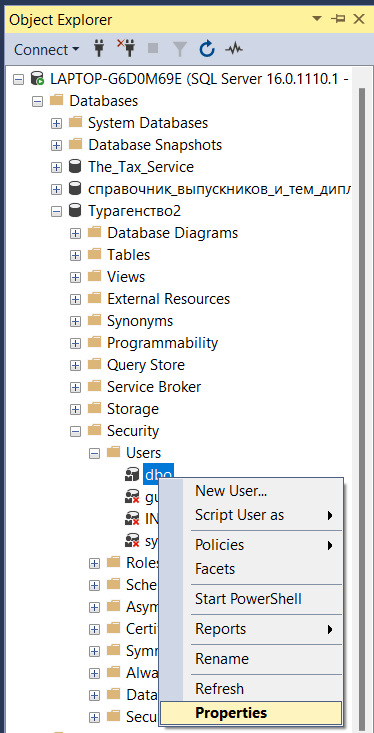


Рисунок 6. – Свойства.

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

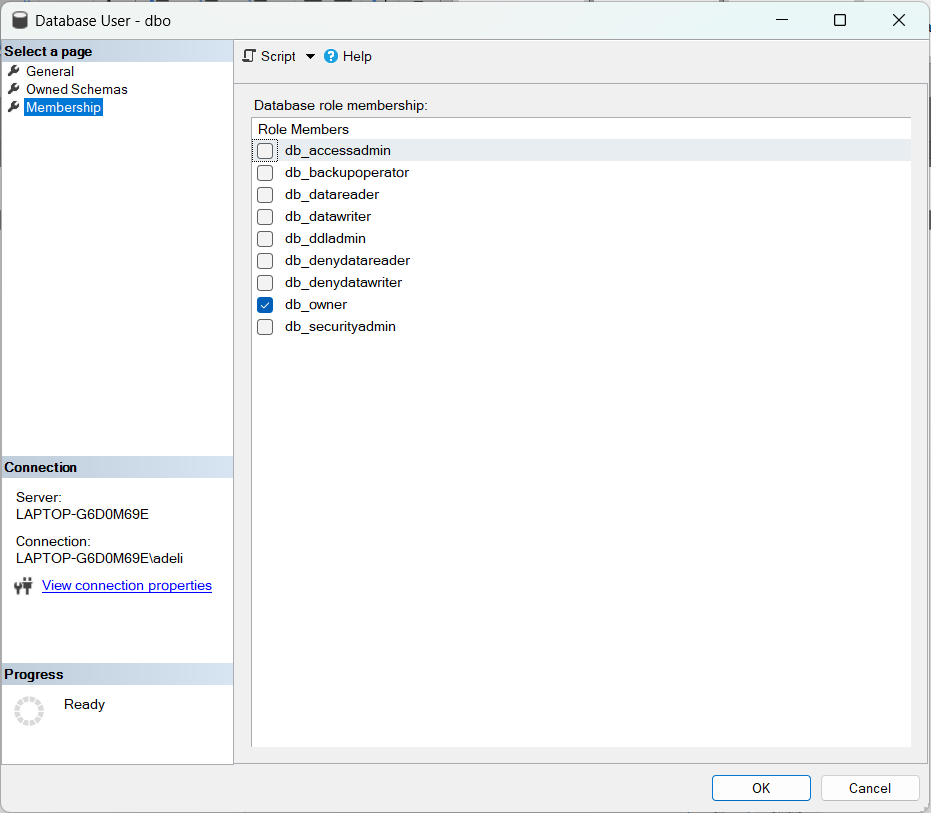


Рисунок 7. – Членство.

**Шаг 4**

В среде Management Studio создаем новую базу данных.

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

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

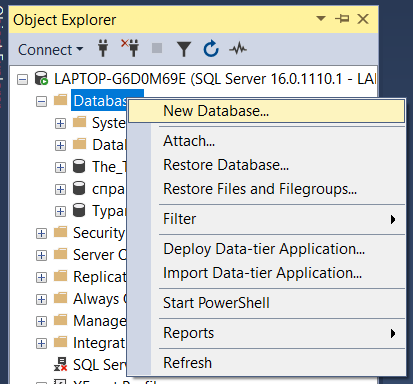
****

Рисунок 8. – Создание БД.

В появившемся диалоговом окне ввели имя для новой базы.

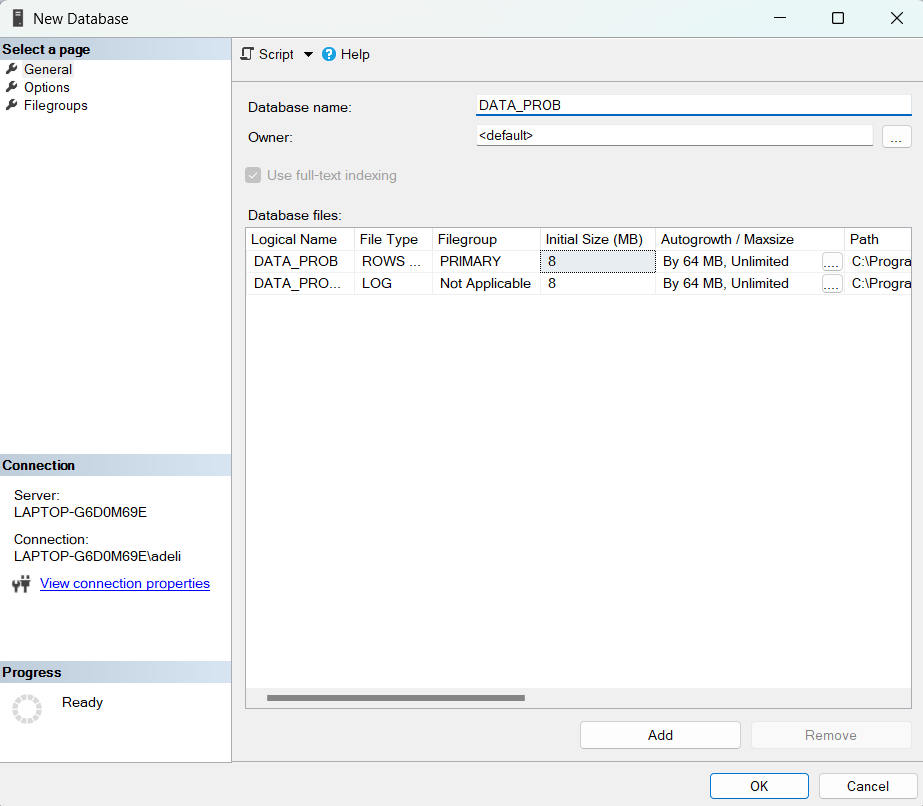
****

Рисунок 9. – Имя БД.

Развернули созданную нами новую базу данных в списке баз данных.

Нашли папку "Security" внутри вашей новой базы данных и развернули ее.

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

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

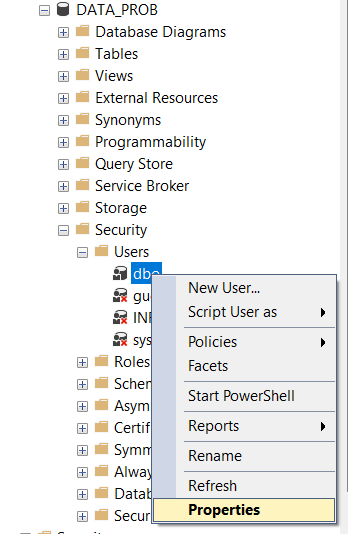
****

Рисунок 10. – Свойства dbo.

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

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

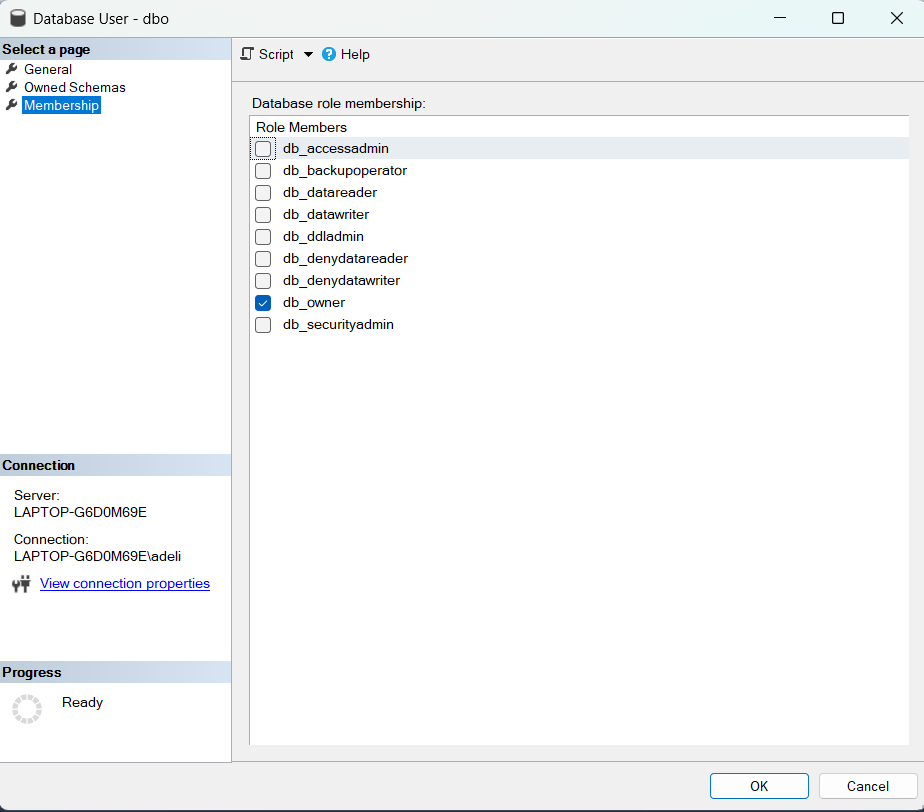
****

Рисунок 11. – Членство.

**Шаг 5.**

Используя приведенный ниже скрипт, создали в базе данных таблицы.

use DATA\_PROB

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]

)

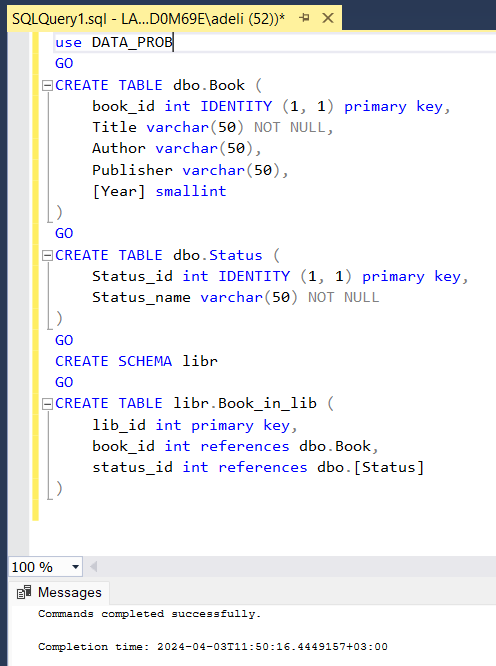


Рисунок 12. – Создание таблиц.

Проверили, что таблицы созданы в нашей базе данных.

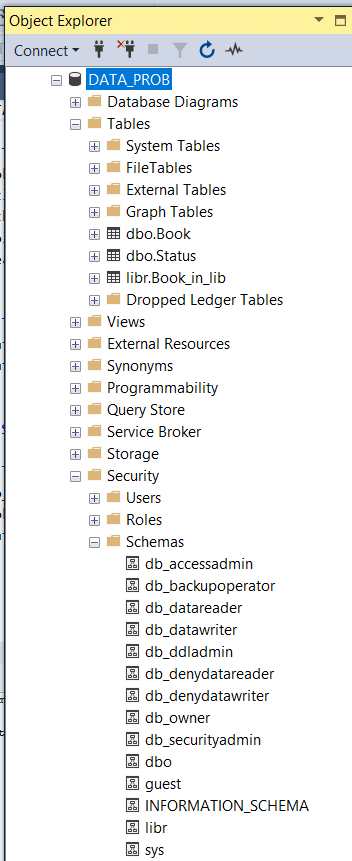


Рисунок 13. – Проверка.

**Шаг 6.**

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

USE DATA\_PROB

-- Создание логина с указанием пароля

CREATE LOGIN kiril WITH PASSWORD = '1111';

-- Создание пользователя и назначение схемы dbo

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

-- Добавление пользователя в роль db\_datareader

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

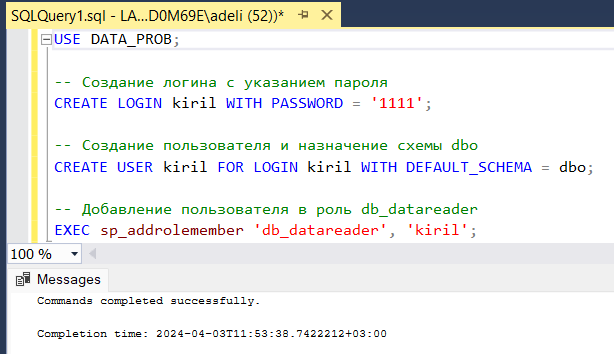
****

Рисунок 14. – Скрипт.

Тестирование:

При тестировании произошла ошибка.

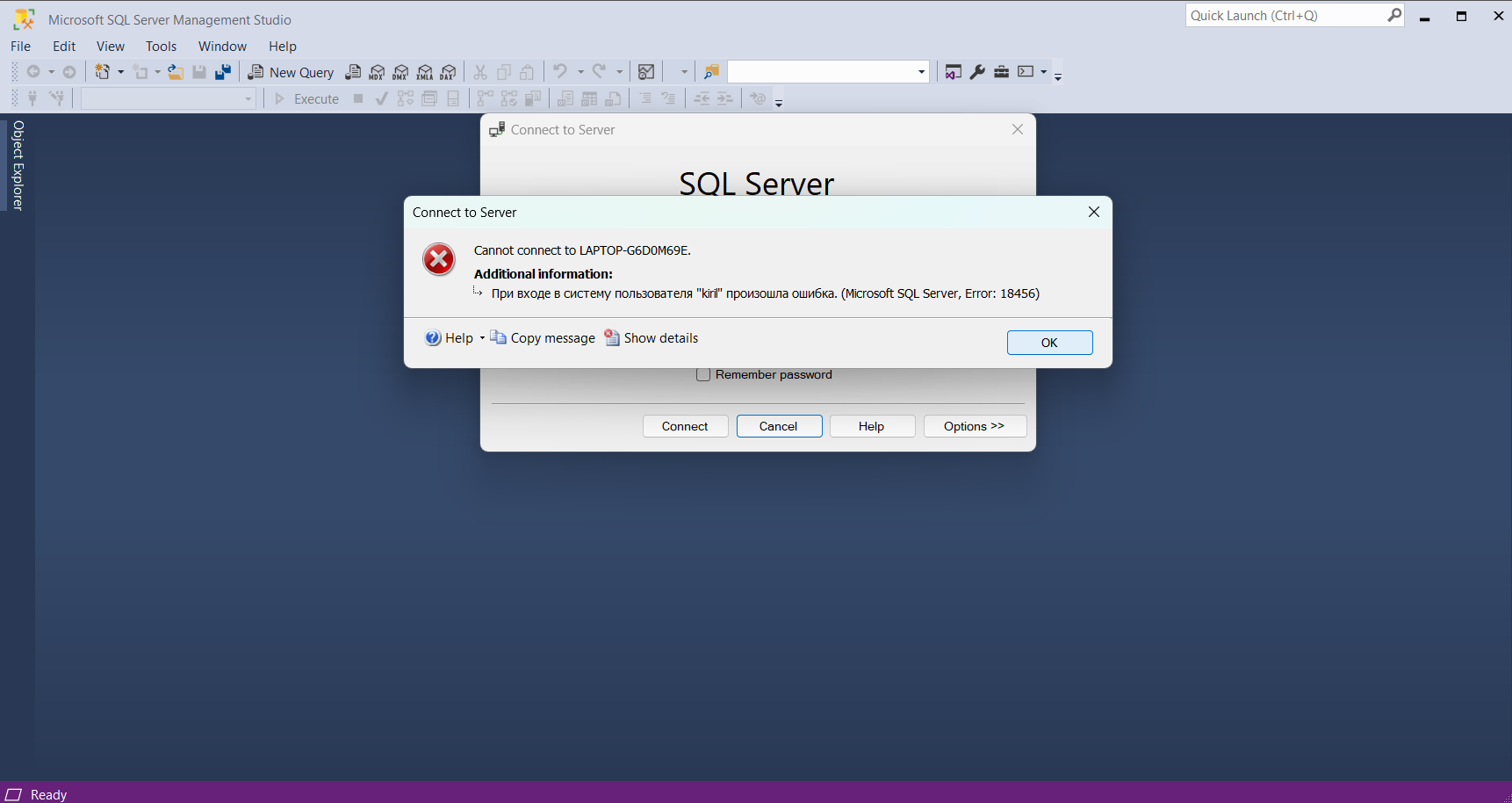
****

Рисунок 15. – Проверка со стороны пользователя.

**Шаг 7.**

Создали новую роль уровня базы данных: Открыли новый запрос T-SQL и ввели следующий запрос:

CREATE ROLE libr\_writer;

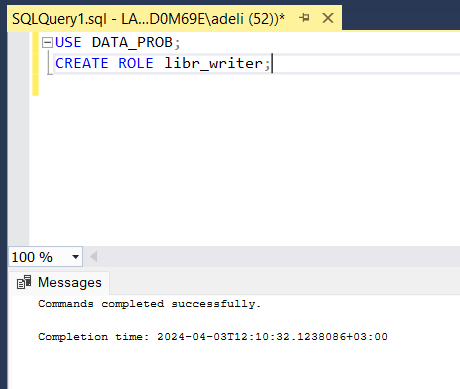


Рисунок 16. – Новая роль.

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

Назначили 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

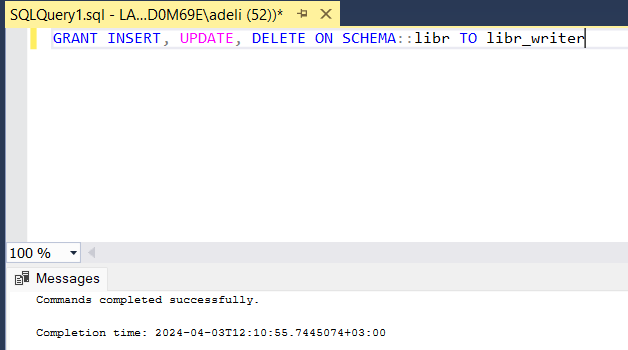


Рисунок 17. – Разращение роли.

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

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

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

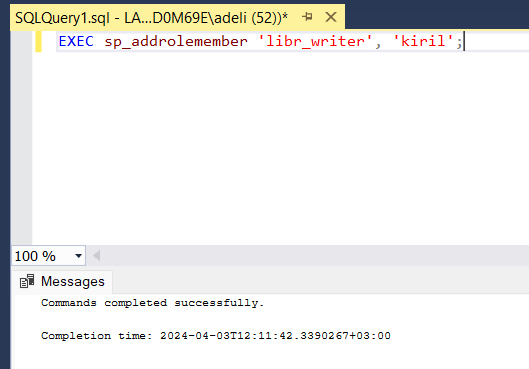


Рисунок 18. – Назначение роли.

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

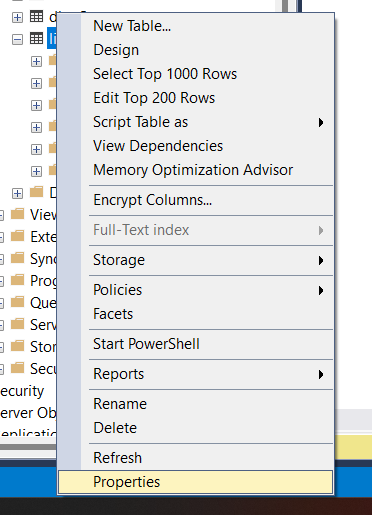


Рисунок 19. – Свойства.

Назначили разрешения для пользователя “kiril”.

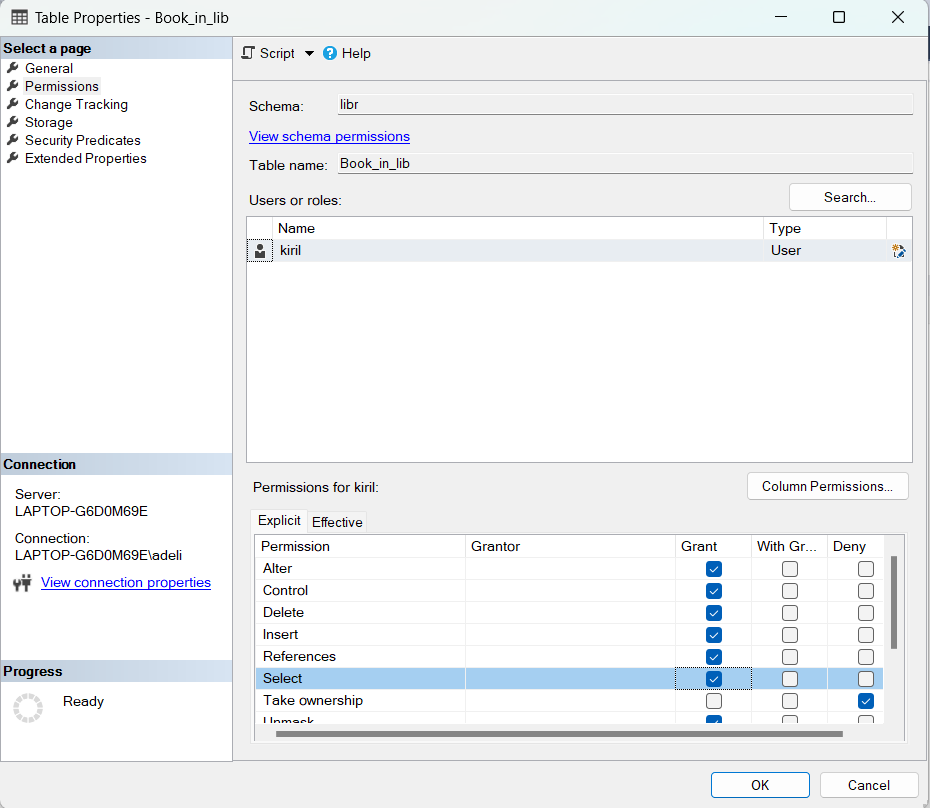


Рисунок 20. – Настройка разращений.

**Шаг 8.**

Выполнили запрос на предоставление прав:

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

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

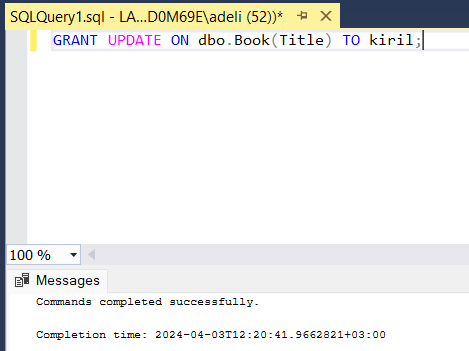


Рисунок 21. – Предоставление право.

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

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

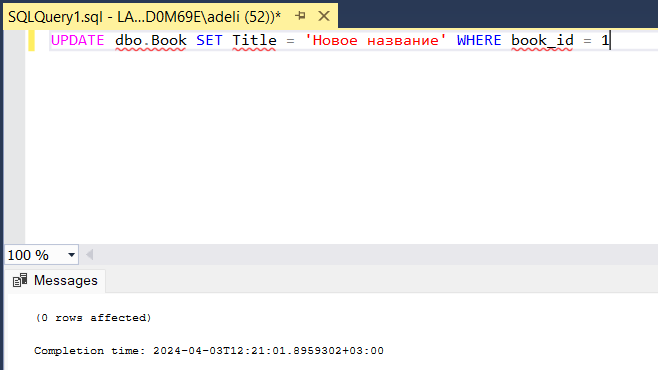


Рисунок 22. – Обновление столбцов

Запрос обновления выполнится успешно без ошибок доступа, это означает, что пользователь "kiril" успешно получил право на изменение столбца "Title".

**Шаг 9.**

Создание представления:

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

CREATE VIEW dbo.NewBooks AS

SELECT \*

FROM dbo.Book

WHERE [Year] >= 2000;

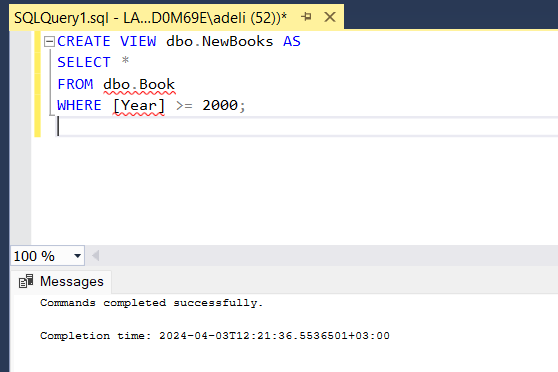


Рисунок 23. - Создания представления.

Теперь предоставили пользователю "kiril" 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 kiril;

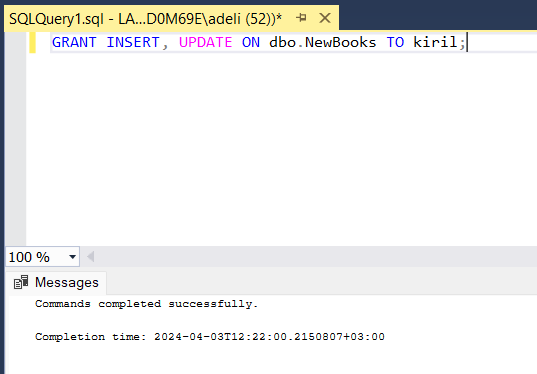


Рисунок 25. – Права на изменение записей.

Убедились, что пользователь "kiril" имеет права на изменение и добавление записей в представление "NewBooks". Это мы проверили, 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;

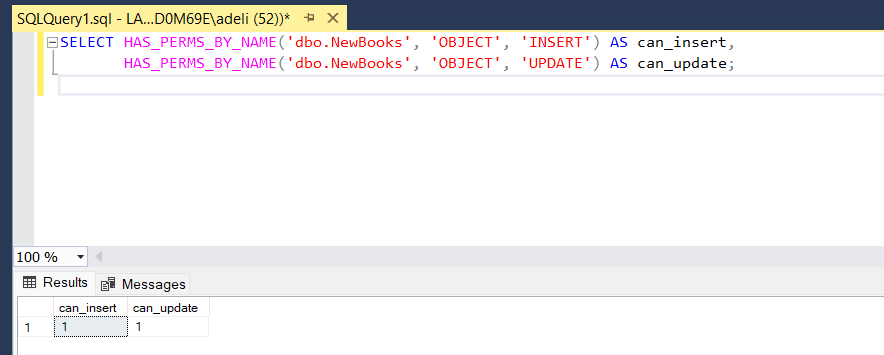


Рисунок 26. – Проверка прав пользователя.

Для обеих операций выводится значение "1", это означает, что пользователь "kiril" имеет соответствующие права.

**Вывод**

Мы научились реализации доступа пользователей к базе данных.