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

Павлова А.А.

Преподаватель

Кожевников К. Д.

Казань 2024

**Цель** – Научиться выполнять резервное копирования и восстановление базы данных из резервной копии.

**Практическая часть**

**Шаг 1**

С помощью SQL Server Management Studio подключаемся к используемому экземпляру SQL Server. Проверяем установленный на сервере режим аутентификации.

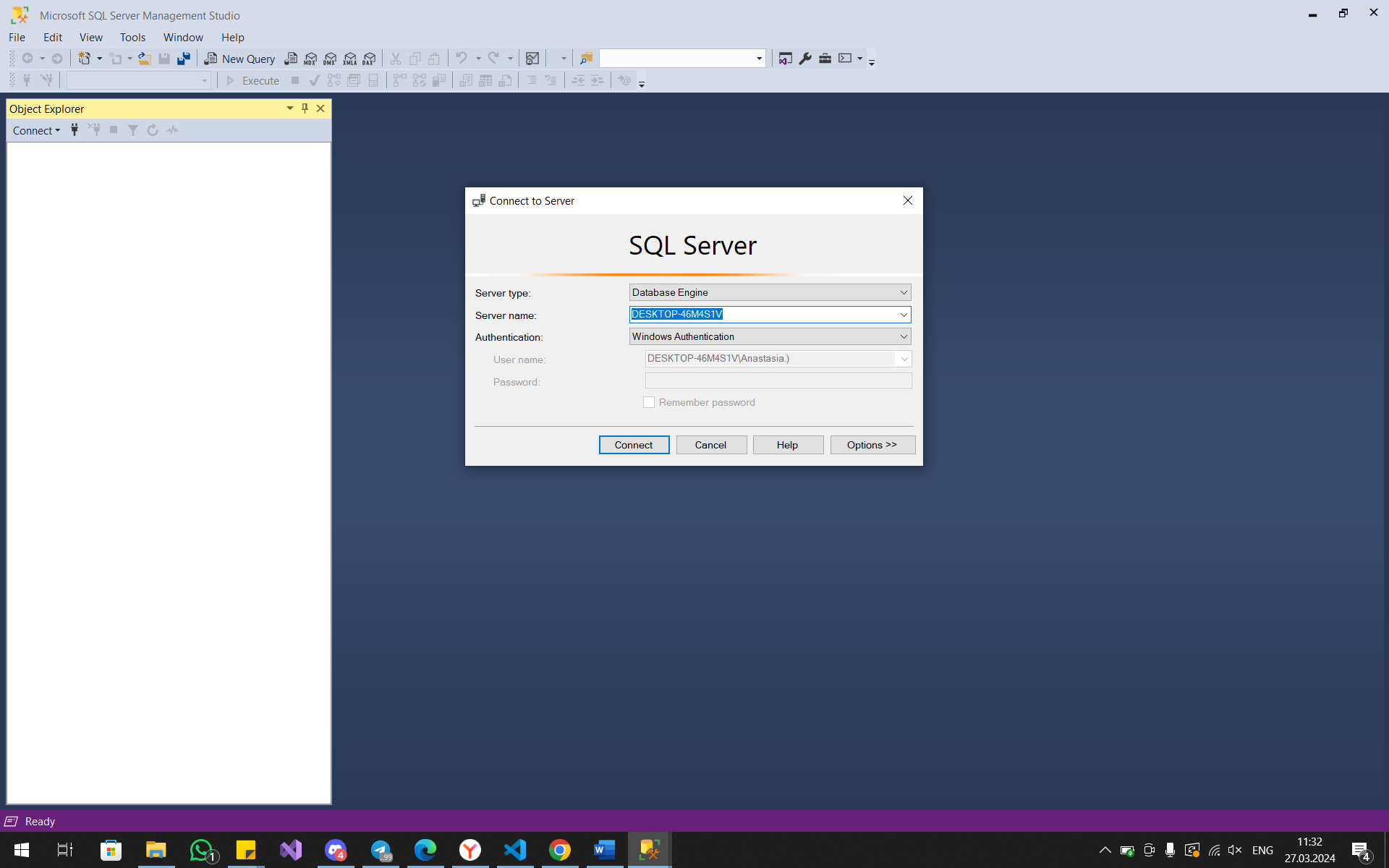


Рисунок 1. – Соединение с сервером.

Заходим в свойства**.**

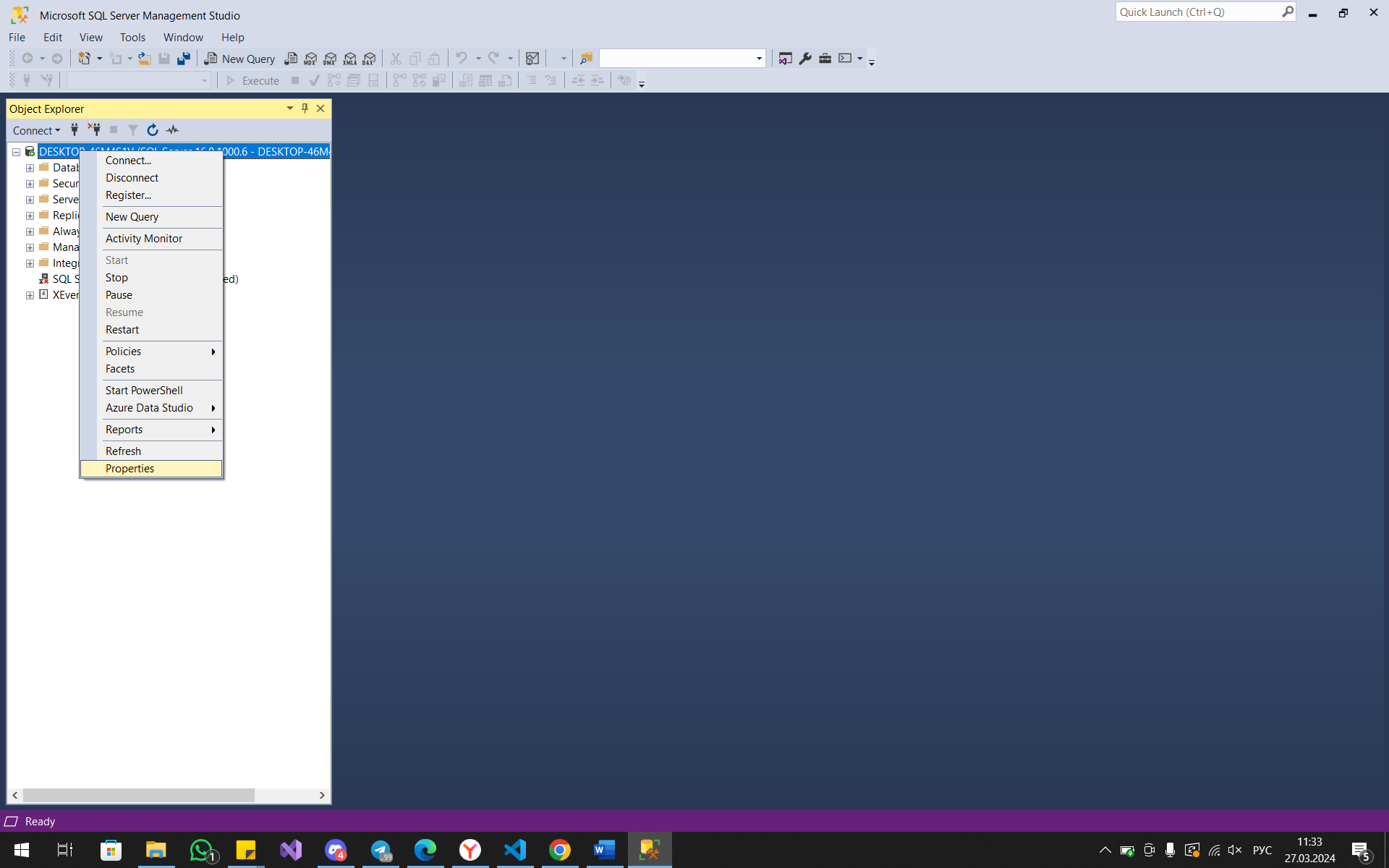


Рисунок 2. – Свойства.

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

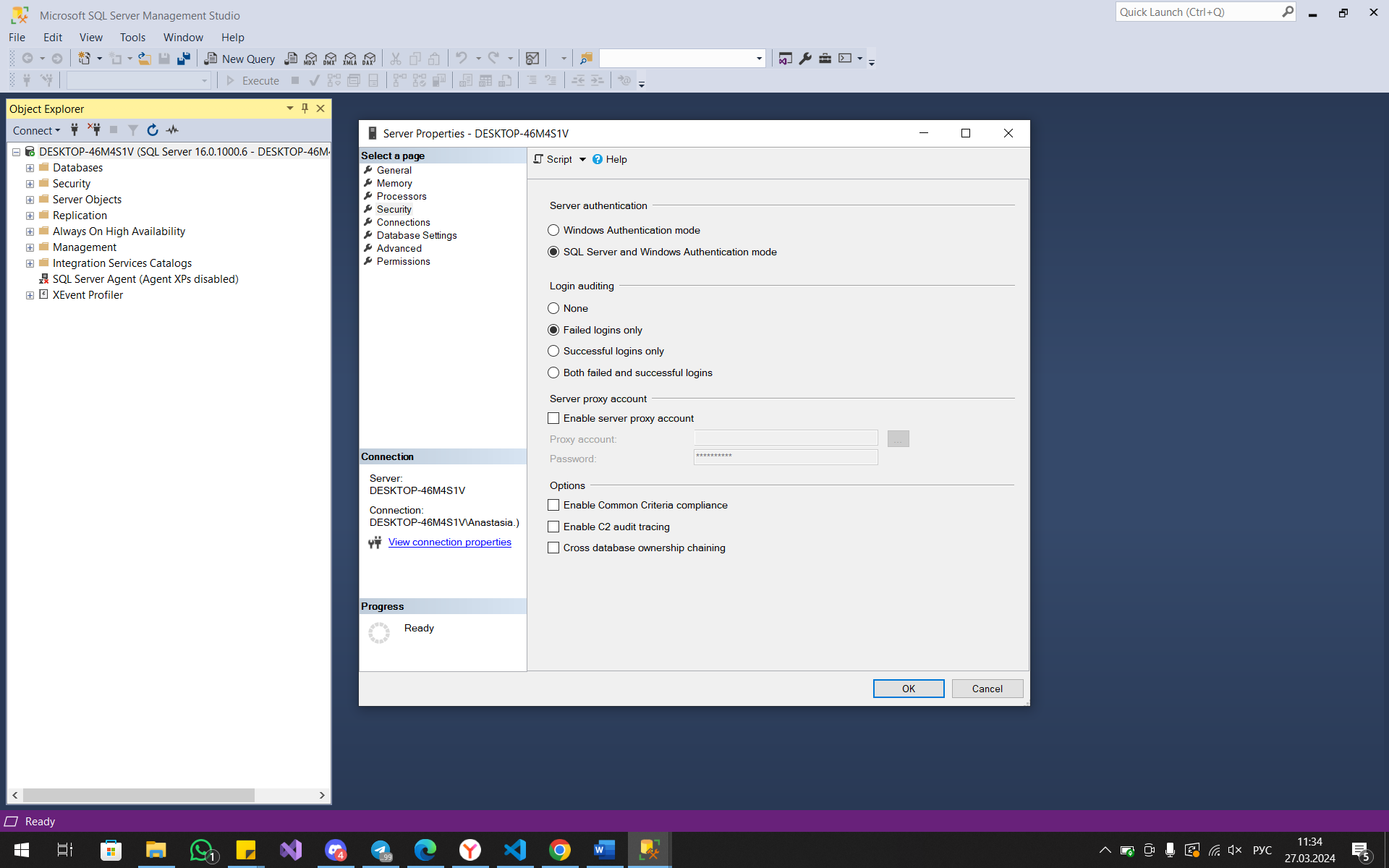




Рисунок 3. – Проверка подлинности.

**Шаг 2**

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

Для этого выполняем следующие действия:

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

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

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

Щелкаем правой кнопкой мыши по нашей учетной записи и выбираем опцию "Properties" (Свойства).

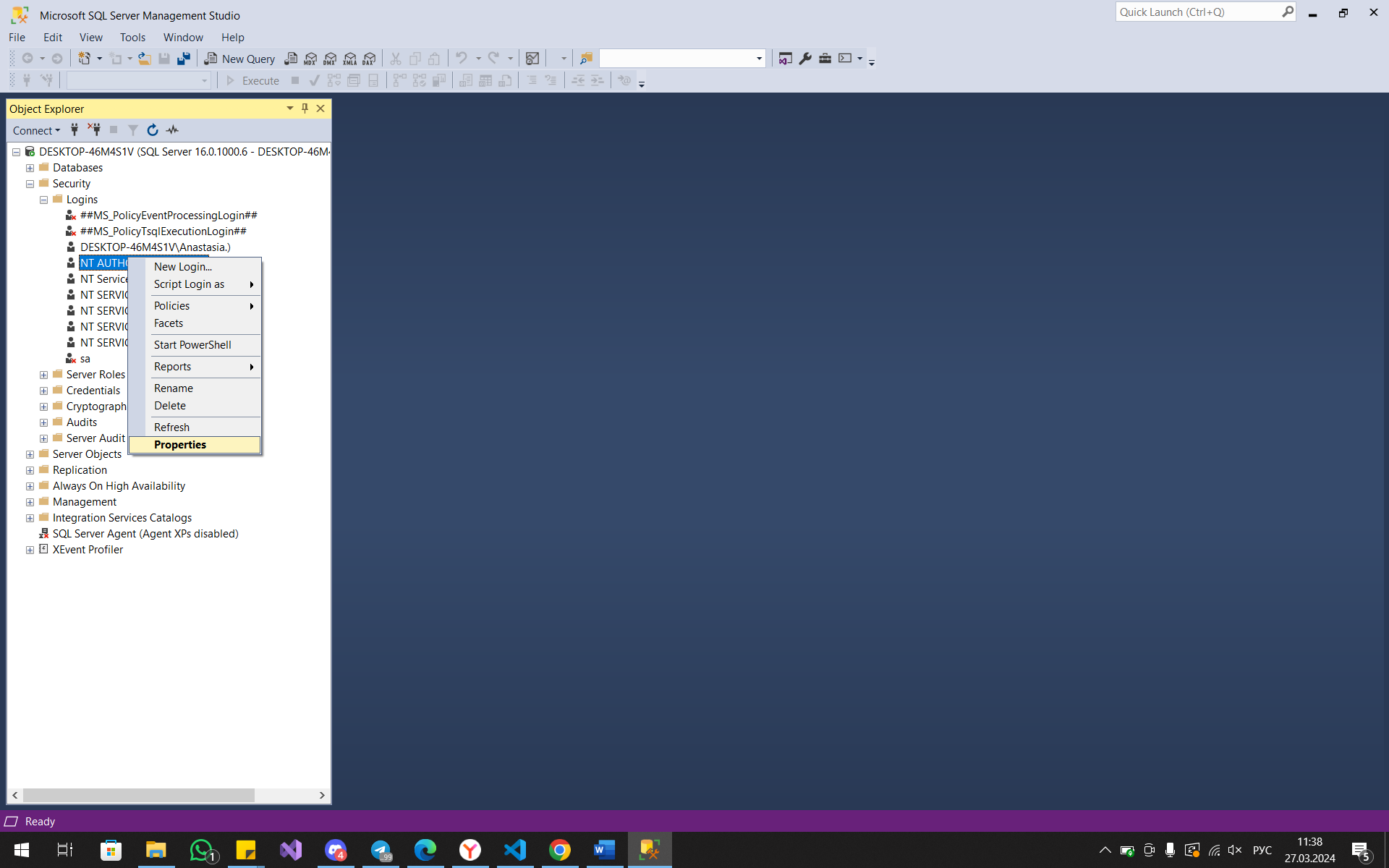


Рисунок 4. – Свойства.

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

Переходим 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.

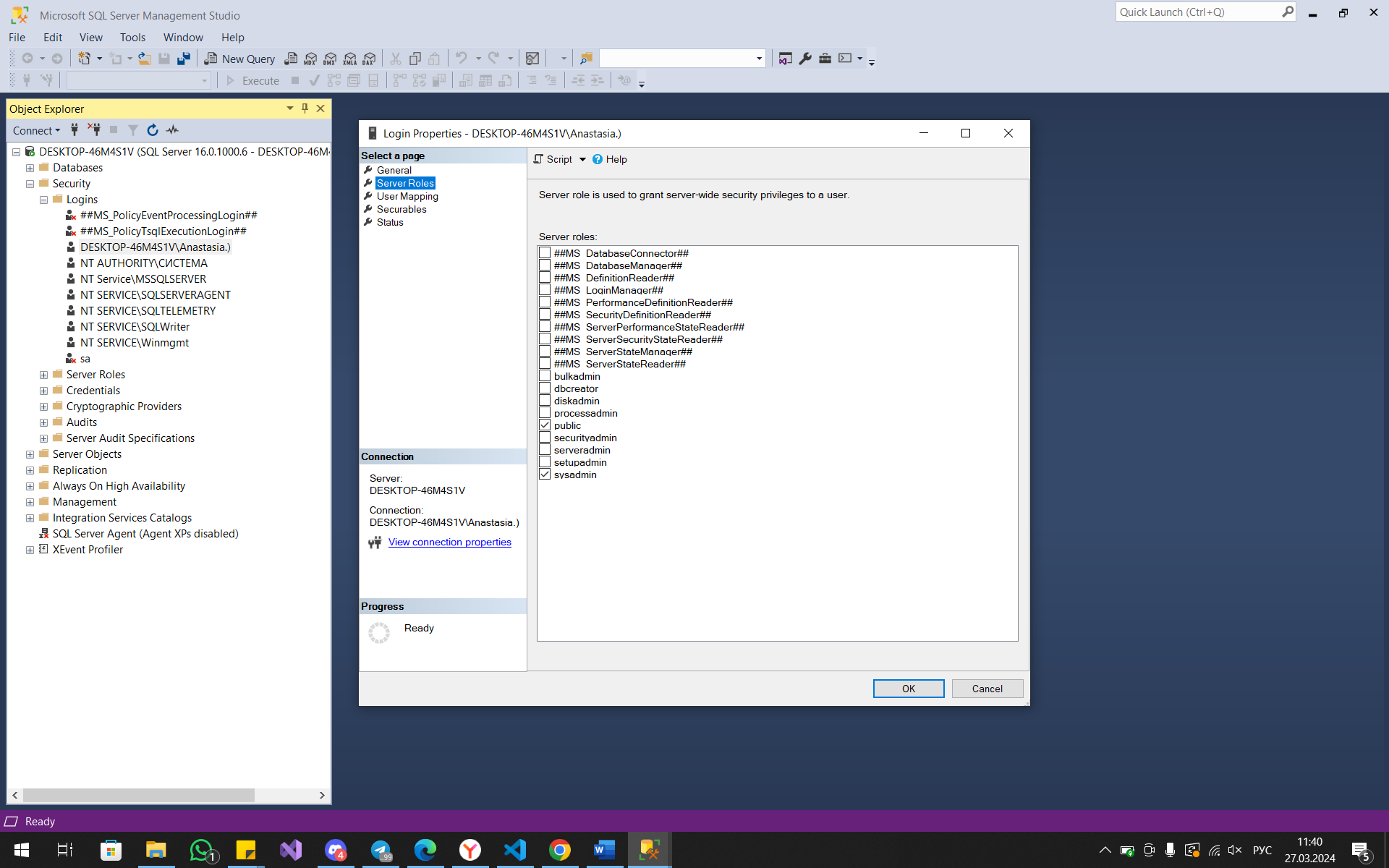
****

Рисунок 5. – Проверка роли сервера.

**Шаг 3**

Находим в окне Object Explorer наш сервер SQL и раскрываем его для просмотра баз данных.

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

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

Для этого выполняем следующие действия:

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

Разворачиваем папку "Security" внутри выбранной базы данных.

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

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

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

Выбираем нашего пользователя в списке.

Щелкаем правой кнопкой мыши на пользователе и выбираем опцию "Properties" (Свойства).

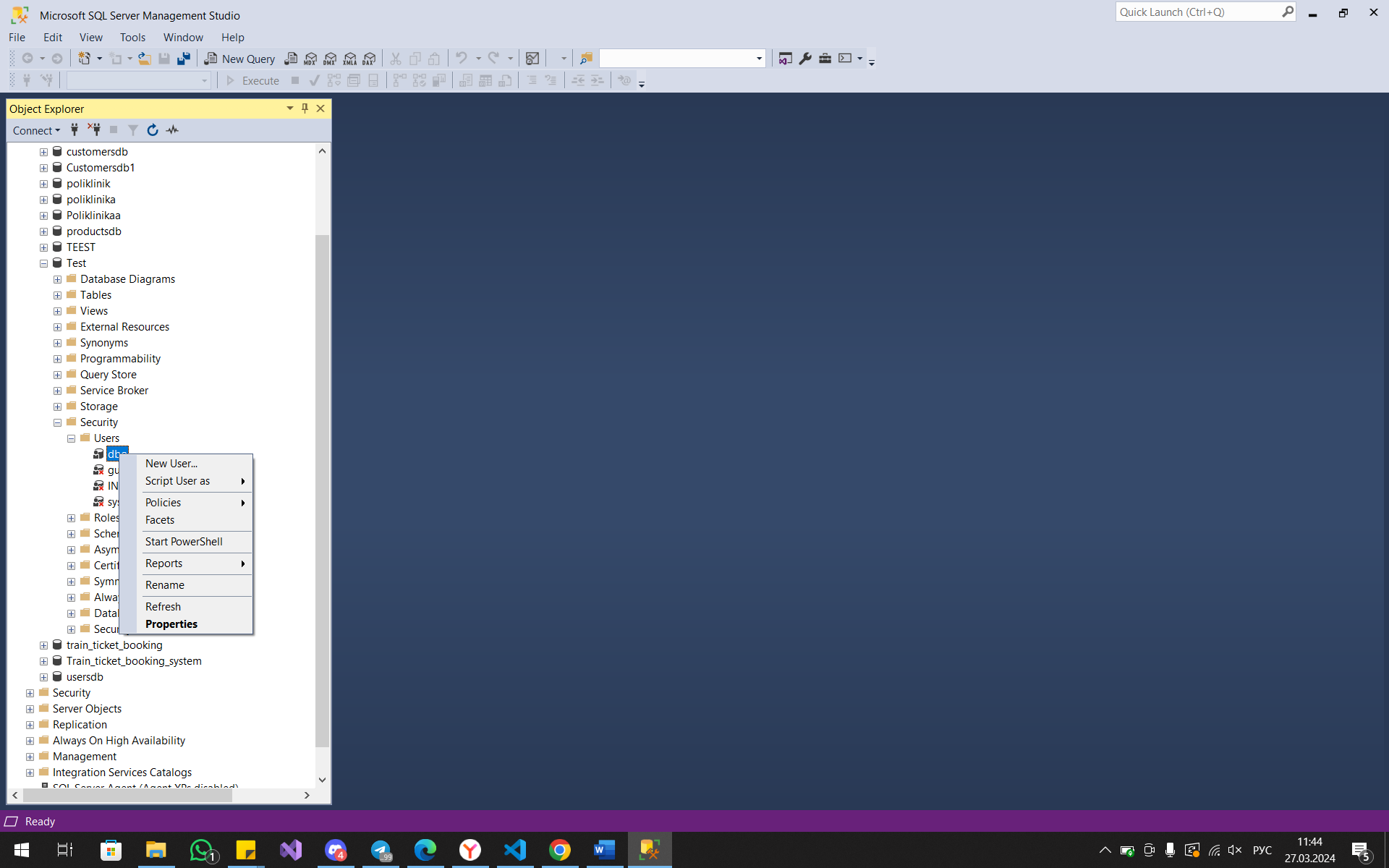


Рисунок 6. – Свойства.

Переходим 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.

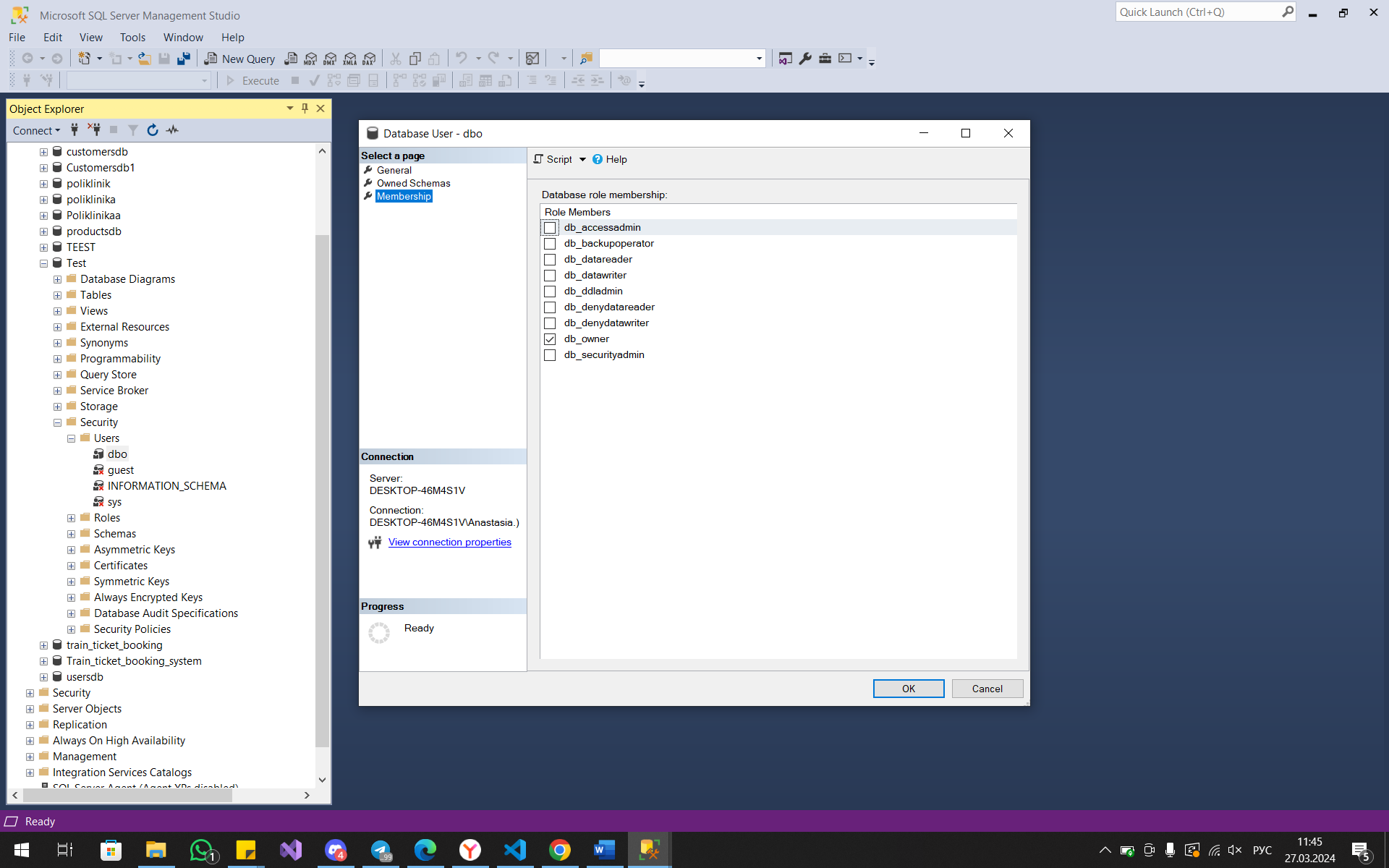


Рисунок 7. – Членство.

**Шаг 4**

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

Создаем новую базу данных:

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

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

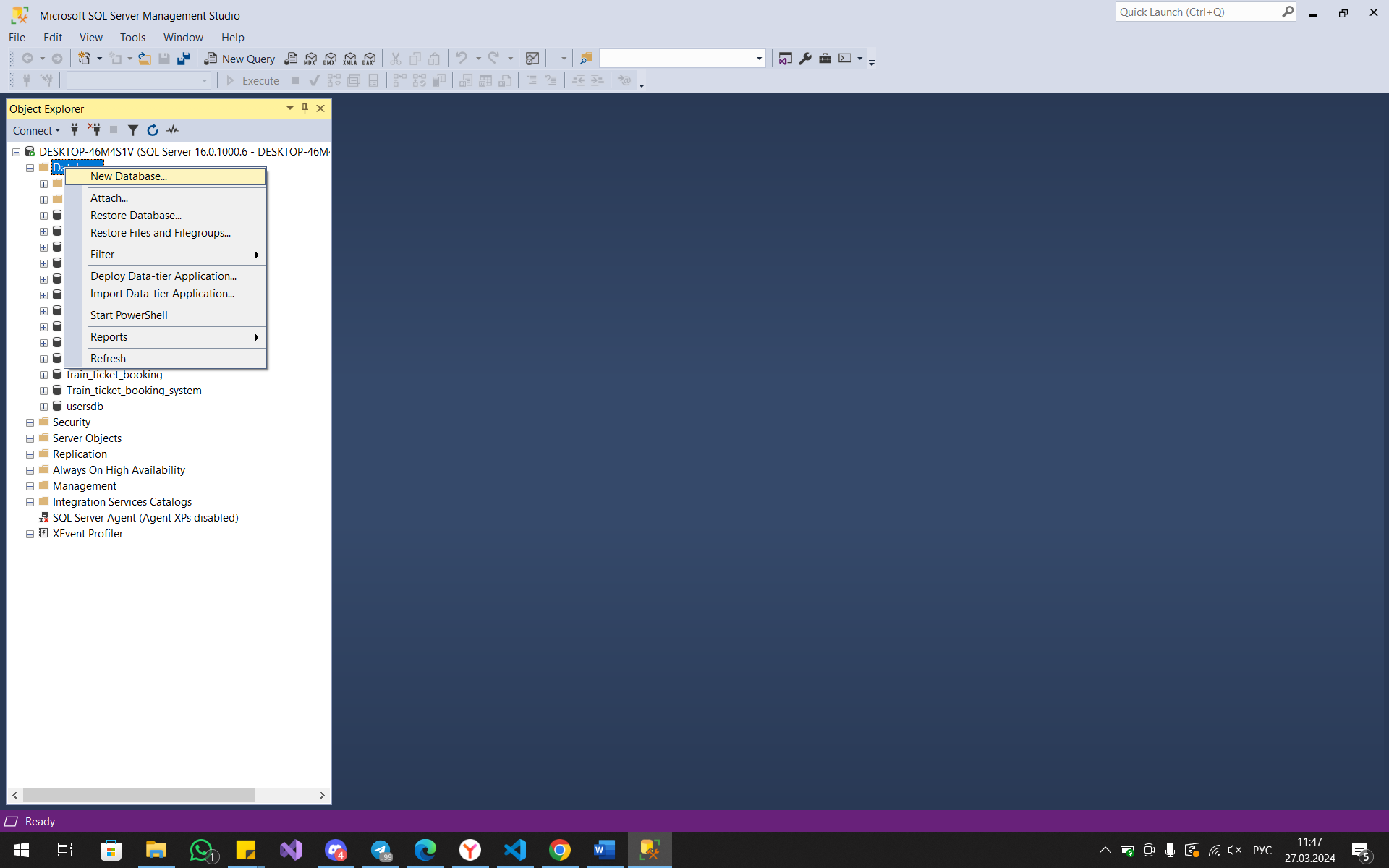
****

Рисунок 8. – Создание БД.

В появившемся диалоговом окне вводим имя для новой базы.

Нажимаем "OK", чтобы создать новую базу данных.

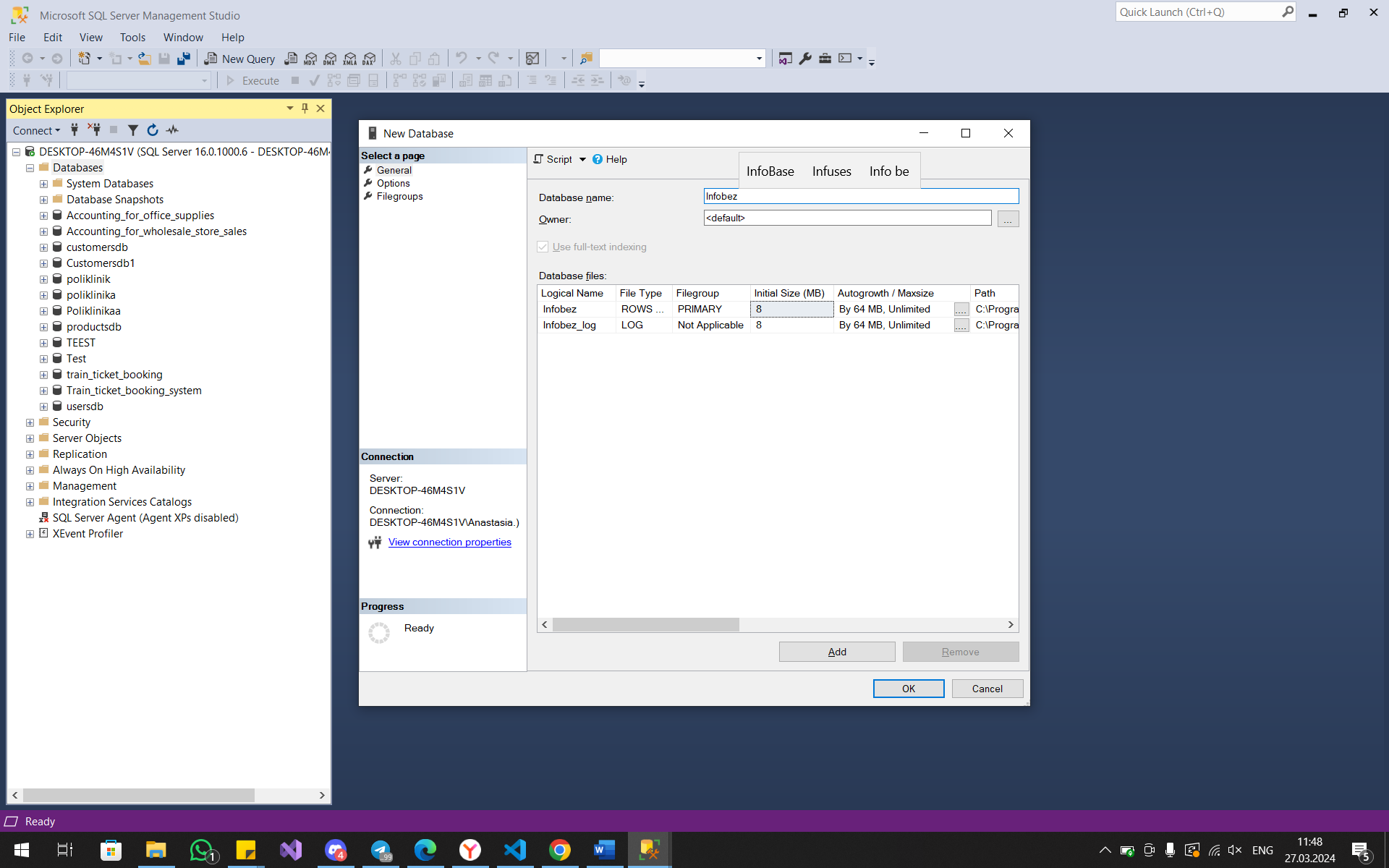
****

Рисунок 9. – Имя БД.

Проверяем список пользователей и ролей:

Разворачиваем созданную нами новую базу данных в списке баз данных.

Находим папку "Security" внутри нашей новой базы данных и разворачиваем ее.

Посмотрим список пользователей, сопоставленных с нашей новой базой данных.

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

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

Щелкаем правой кнопкой мыши на пользователе "dbo" и выбираем опцию "Properties" (Свойства).

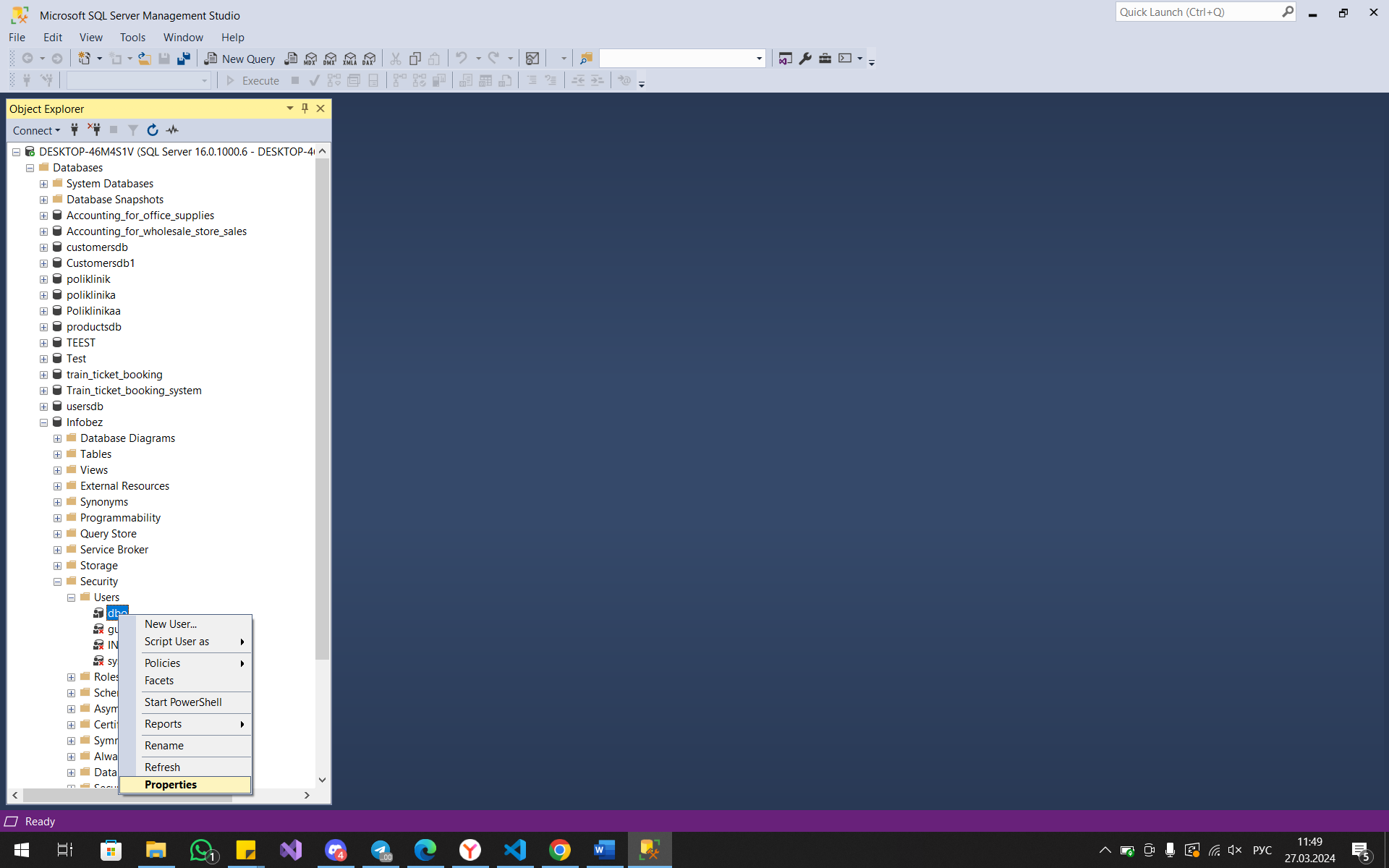
****

Рисунок 10. – Свойства dbo.

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

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

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

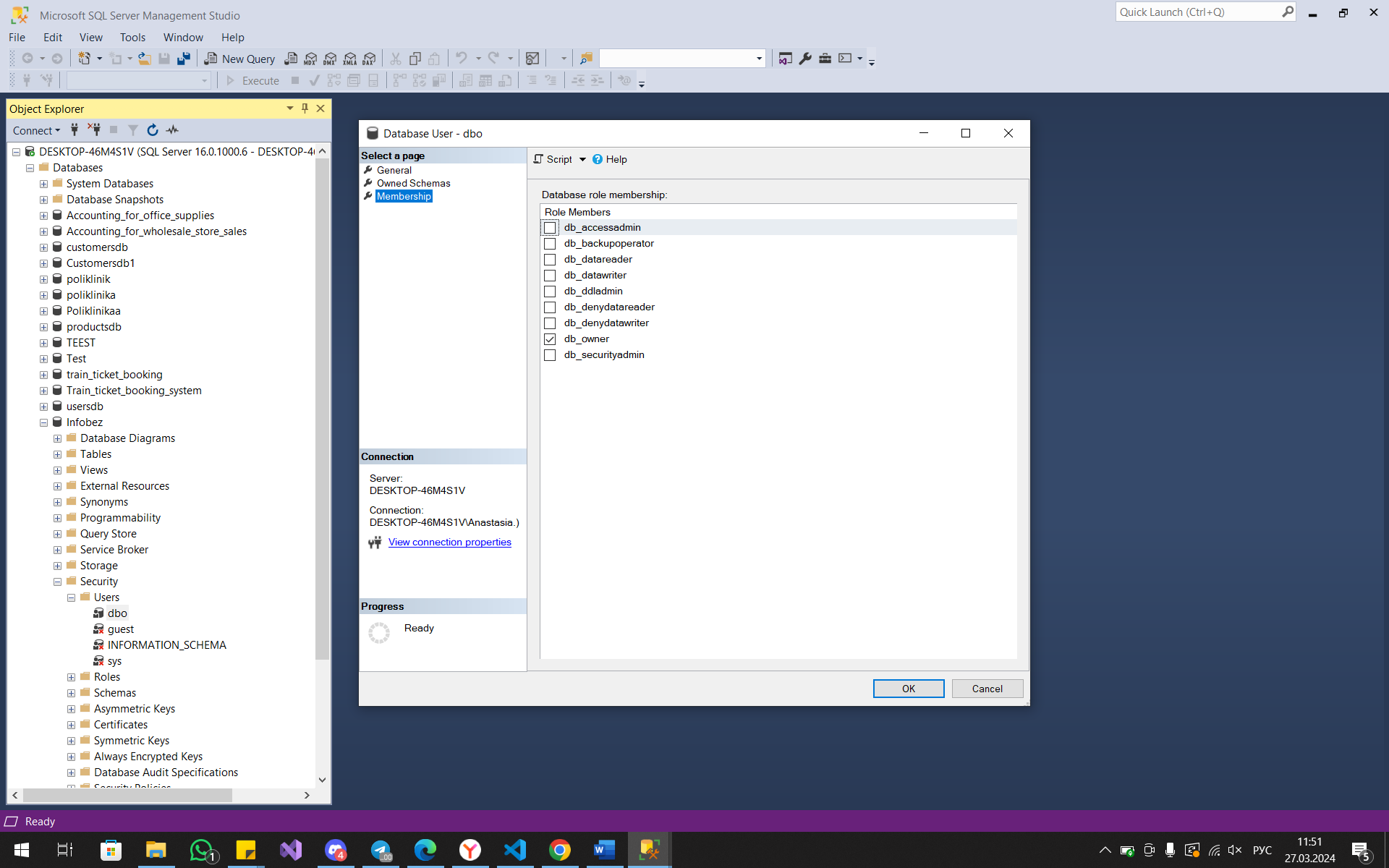
****

Рисунок 11. – Членство.

**Шаг 5.**

Используя приведенный ниже скрипт, создаем в базе данных таблицы.

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

use [Infobez]

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]

)

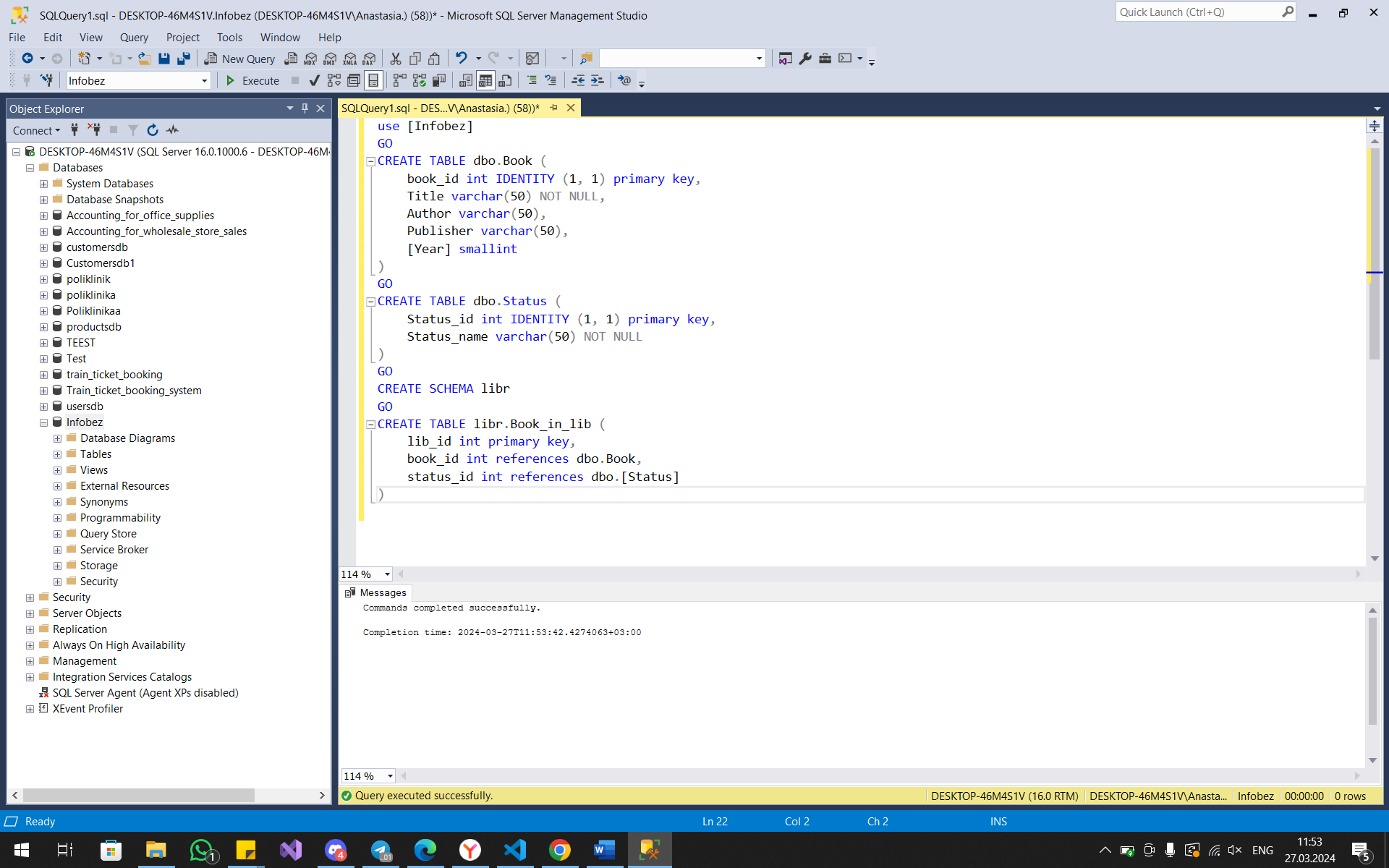


Рисунок 12. – Создание таблиц.

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

Проверяем, что таблицы созданы в нашей базе данных.

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

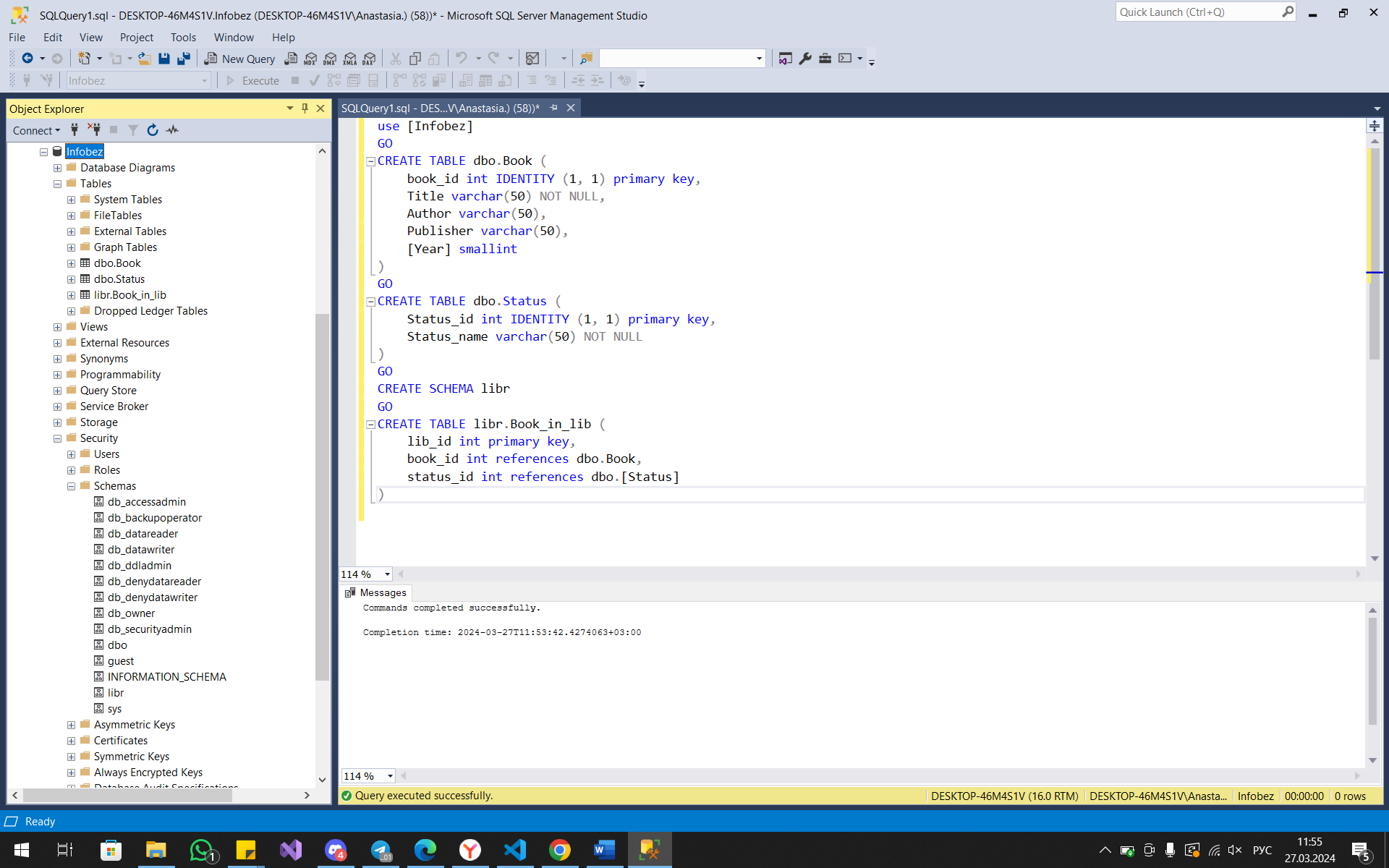


Рисунок 13. – Проверка.

**Шаг 6.**

Создаем пользователя, выбираем схему dbo, добавляем в роль db\_datareader:

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

USE [Infobez];

-- 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';

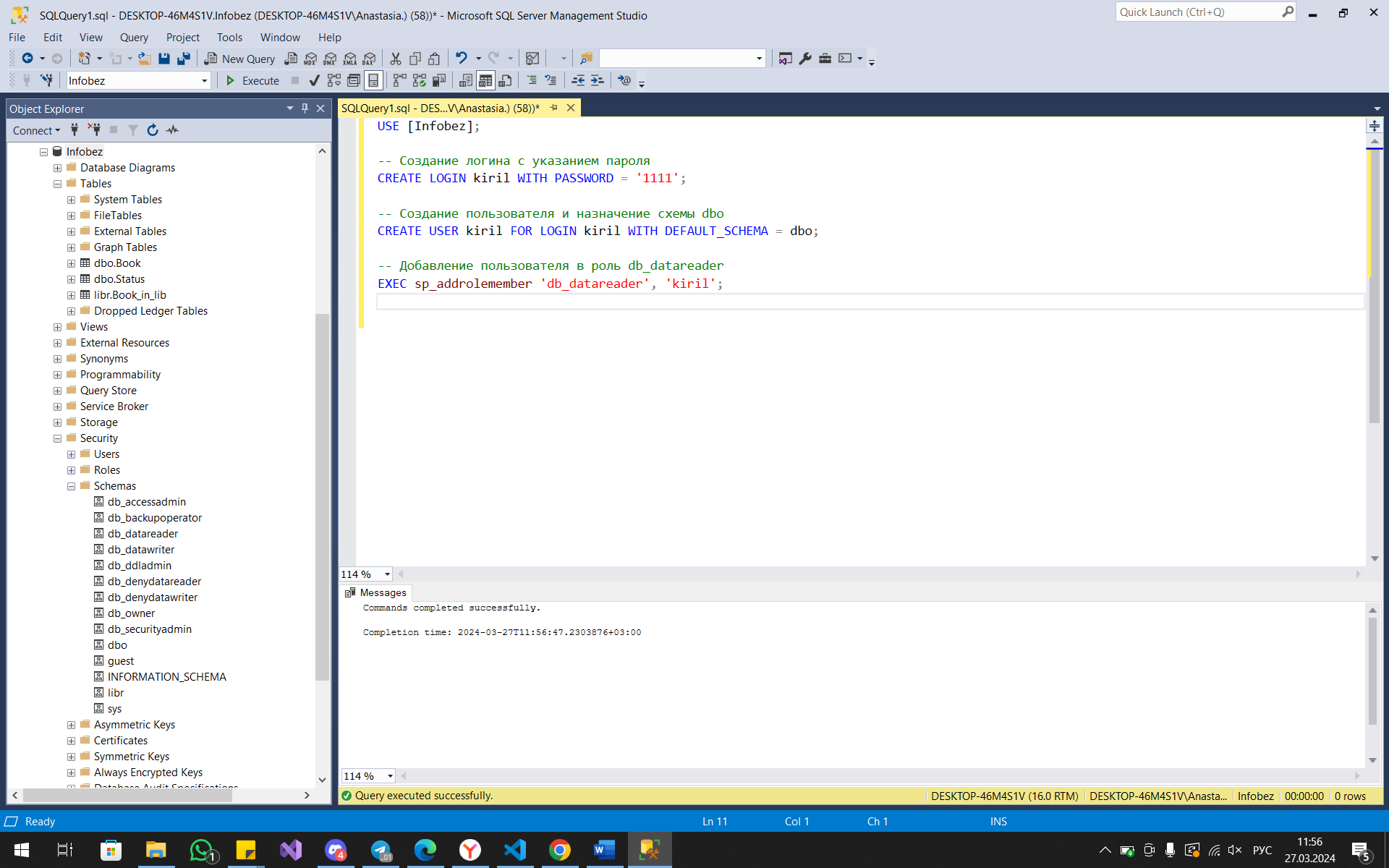
****

Рисунок 14. – Скрипт.

**Шаг 7.**

Создание новой роли и назначение разрешений:

Подключаемся 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 "kiril".

Создаем новую роль уровня базы данных: Открываем новый запрос T-SQL и вводим следующий запрос:

CREATE ROLE libr\_writer;

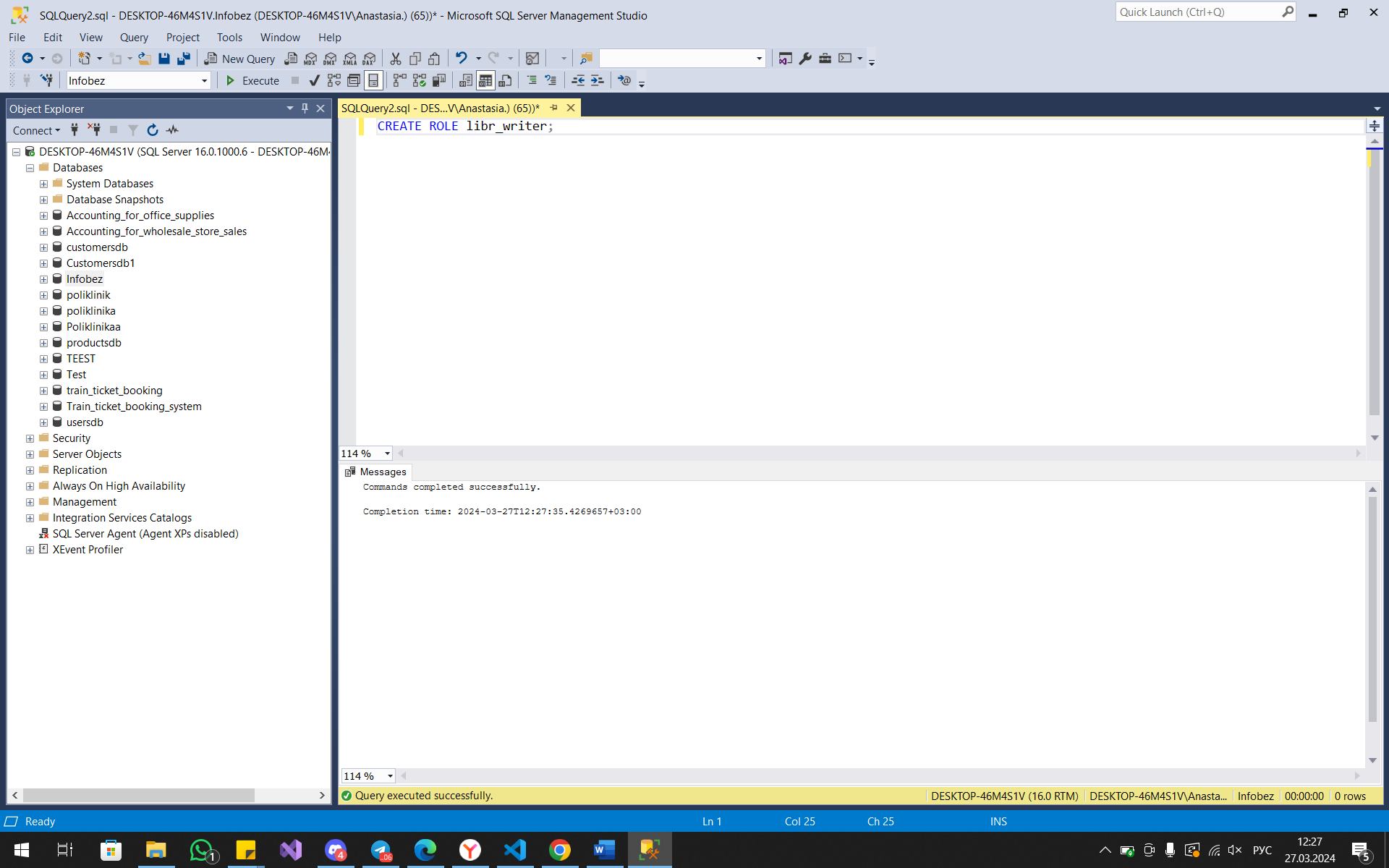


Рисунок 15. – Новая роль.

Этот запрос создаст новую роль с именем "libr\_writer". Назначаем разрешения роли для схемы "libr":

Вводим 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

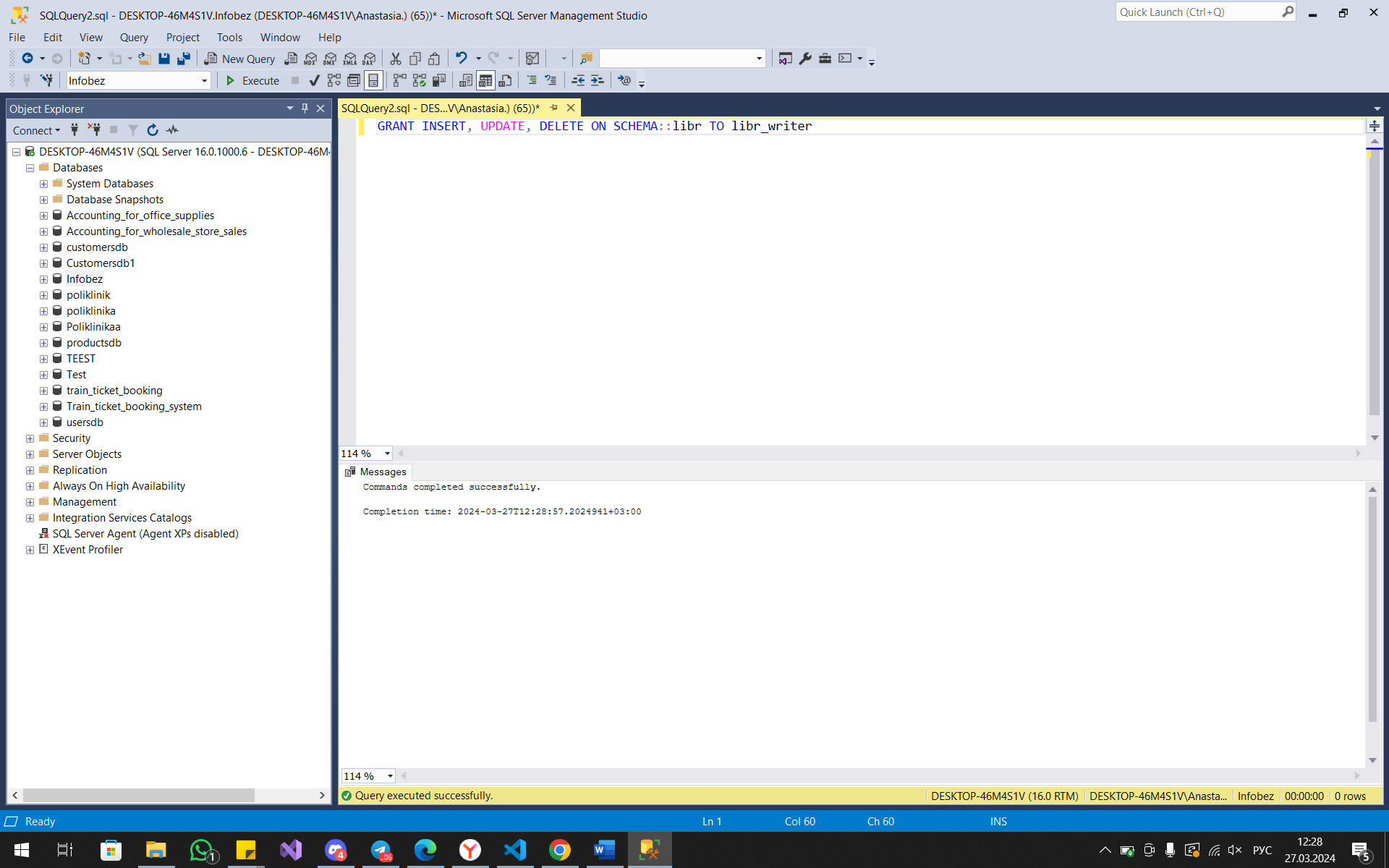


Рисунок 16. – Разрешение роли.

Добавляем пользователя в созданную роль:

Вводим 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';

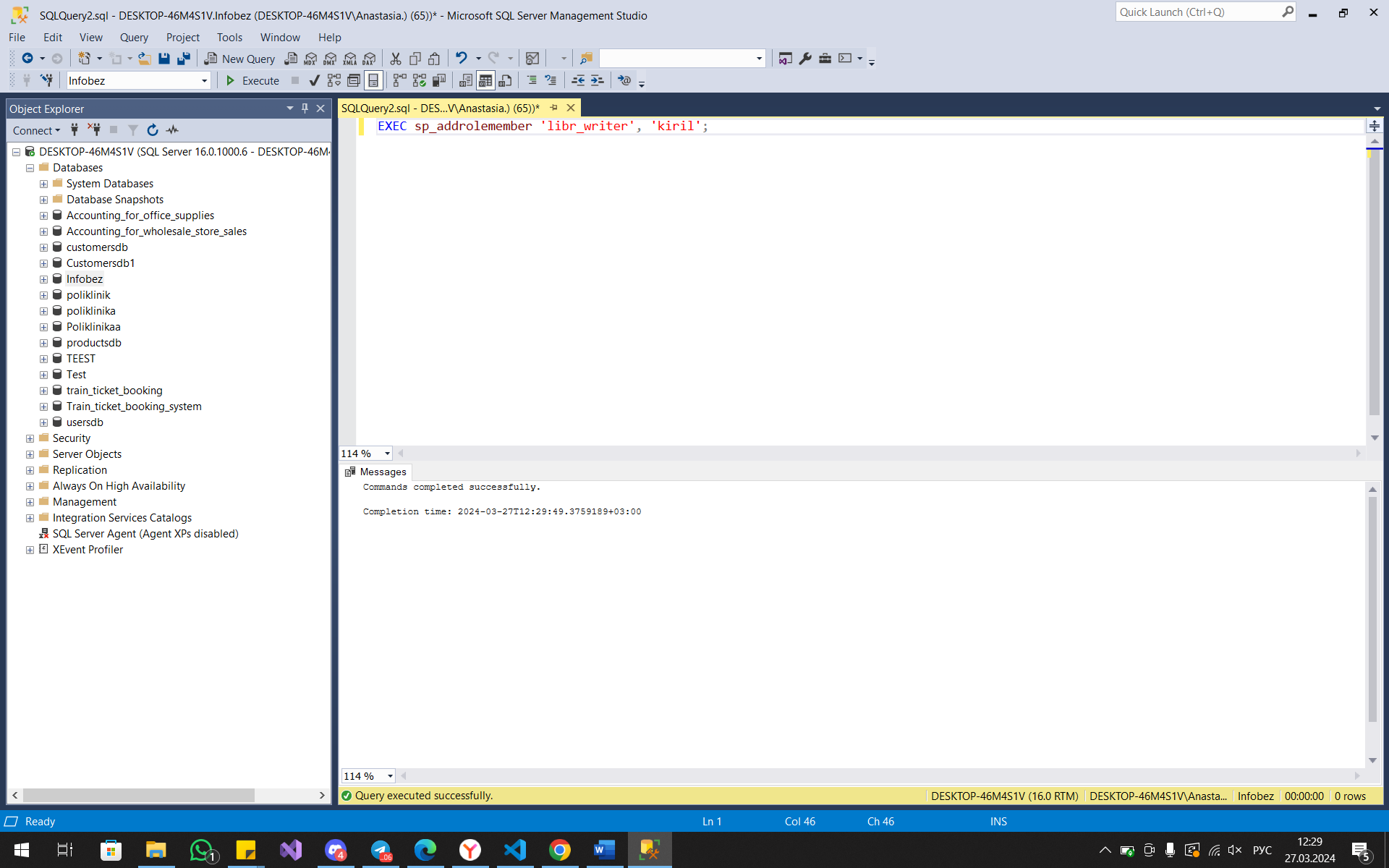


Рисунок 17. – Назначение роли.

Проверяем доступ пользователя к таблице:

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

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

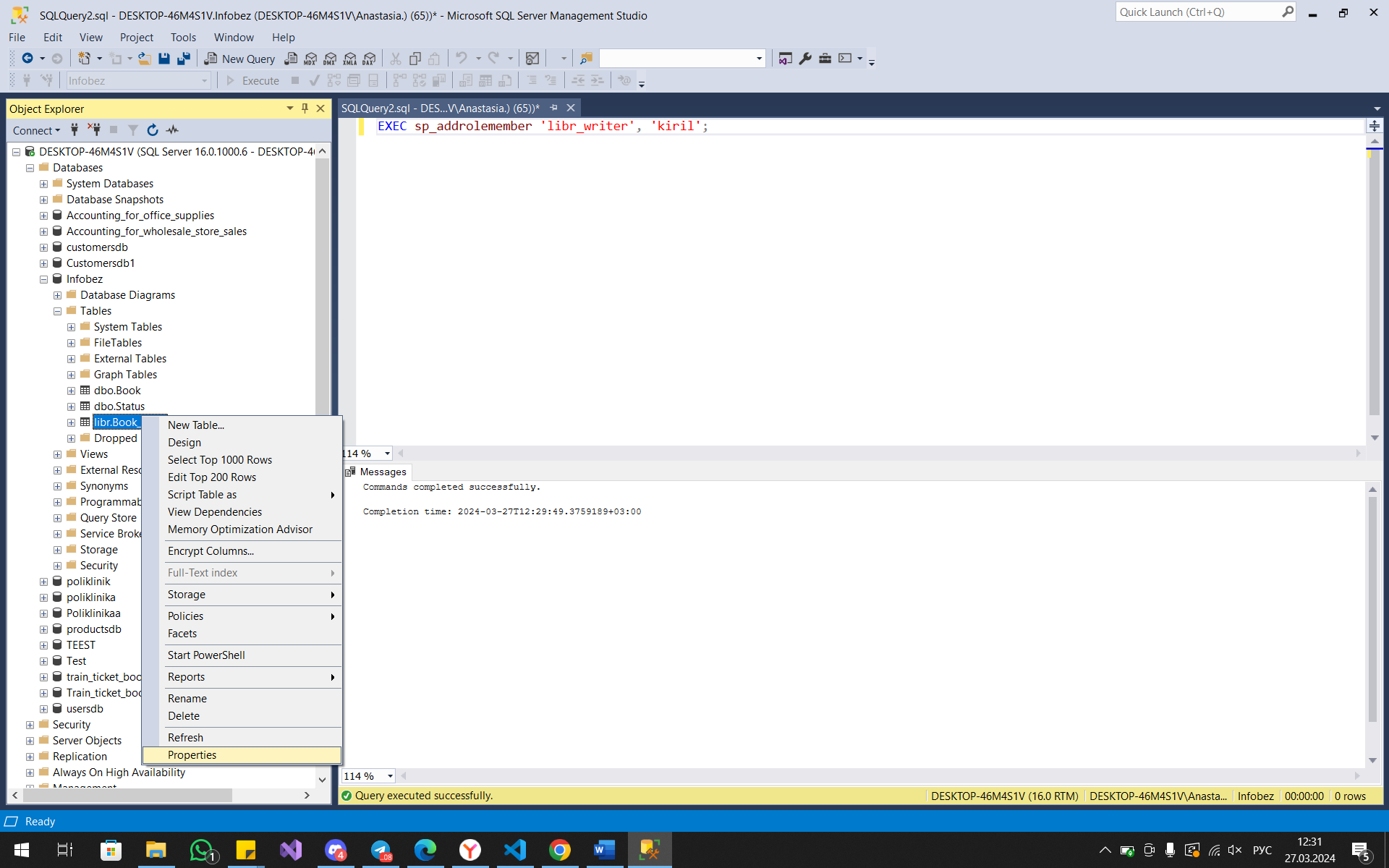


Рисунок 18. – Свойства.

Назначаем разрешения для пользователя “kiril” как на рисунке 19.

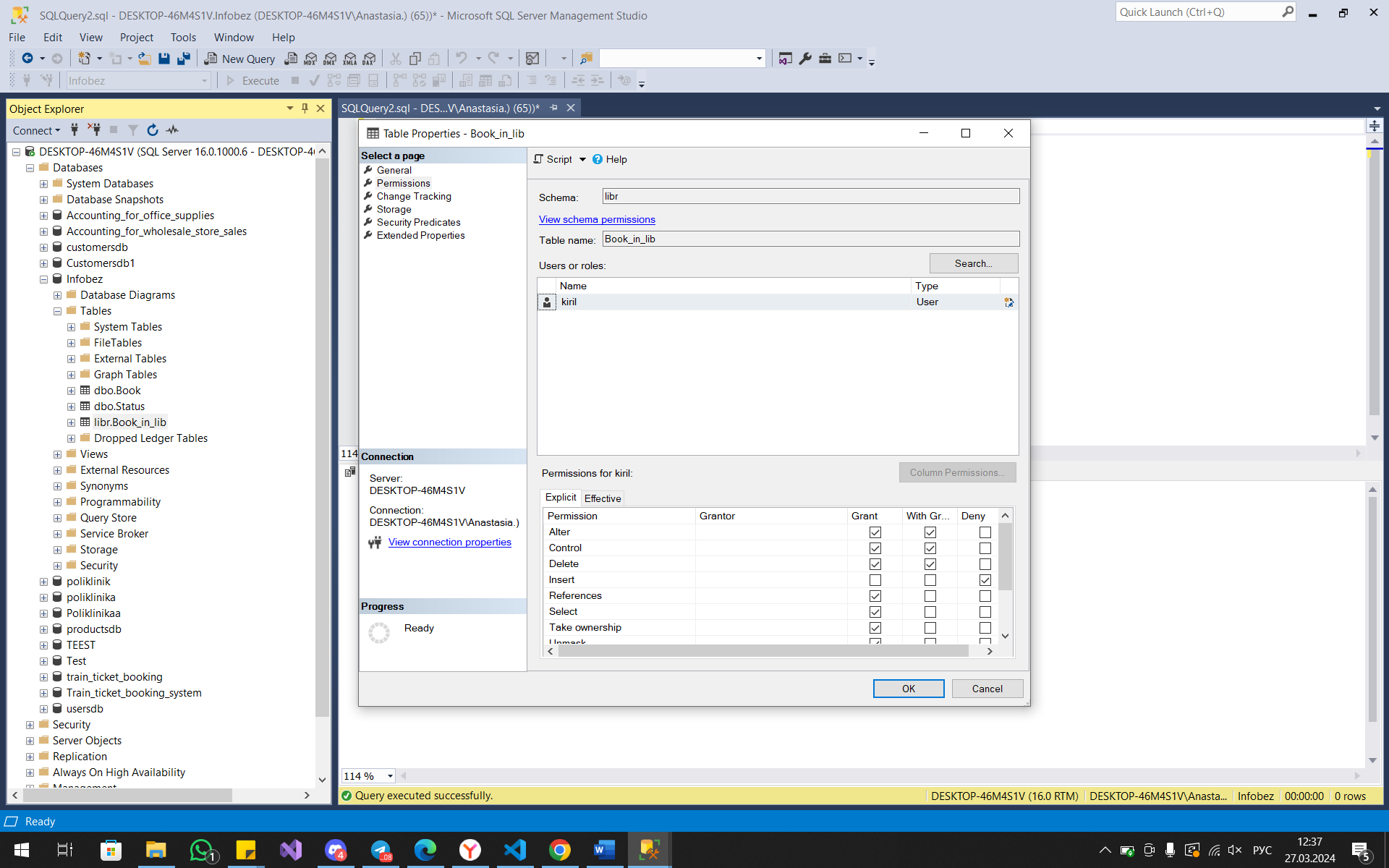


Рисунок 19. – Настройка разрешений.

**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 Выполняем аналогичные действия в своей базе данных, проверяем, что пользователь получил указанные разрешения.

Открываем новый запрос:

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

Выполняем запрос на предоставление прав:

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

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

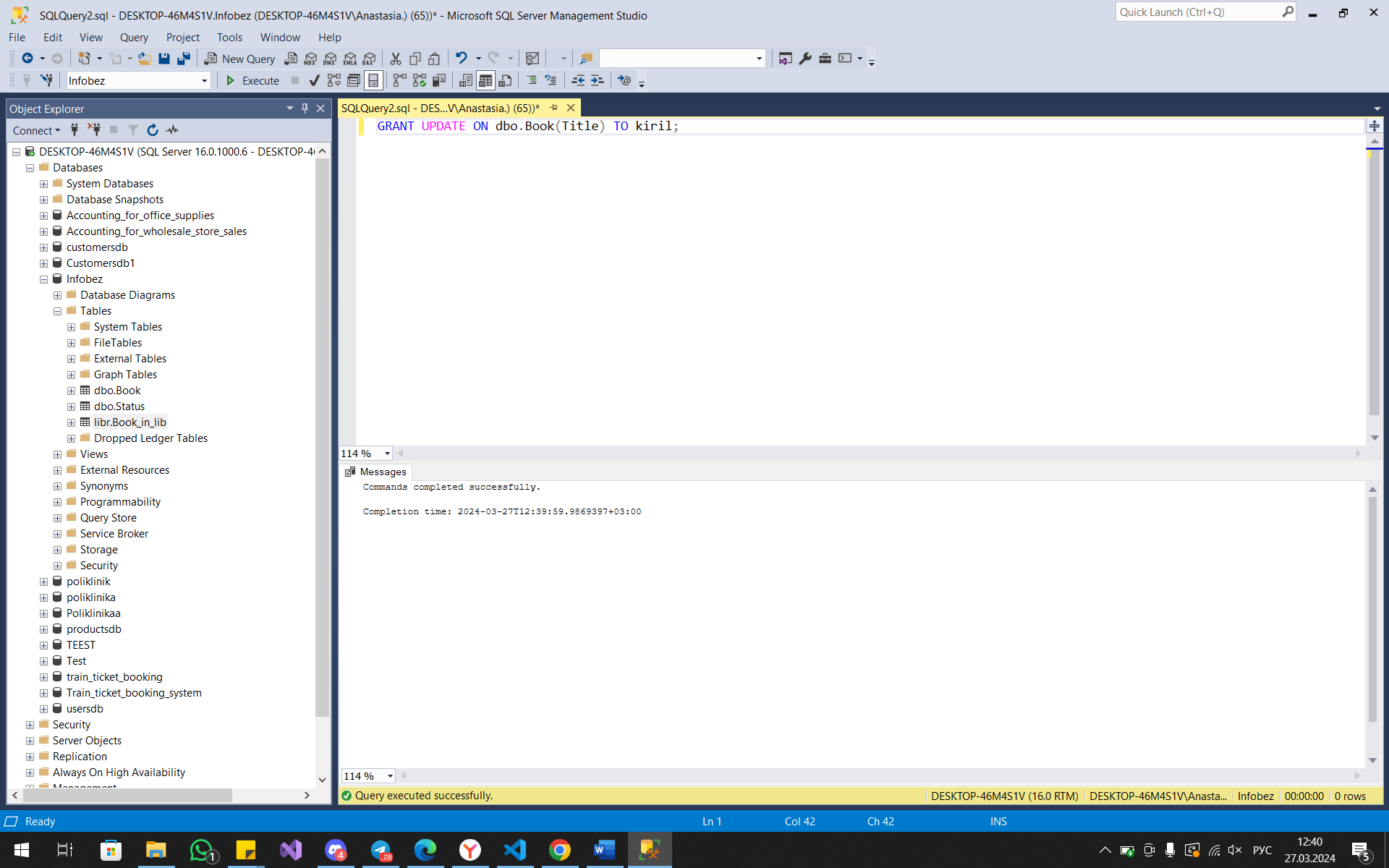


Рисунок 20. – Предоставление право.

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

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

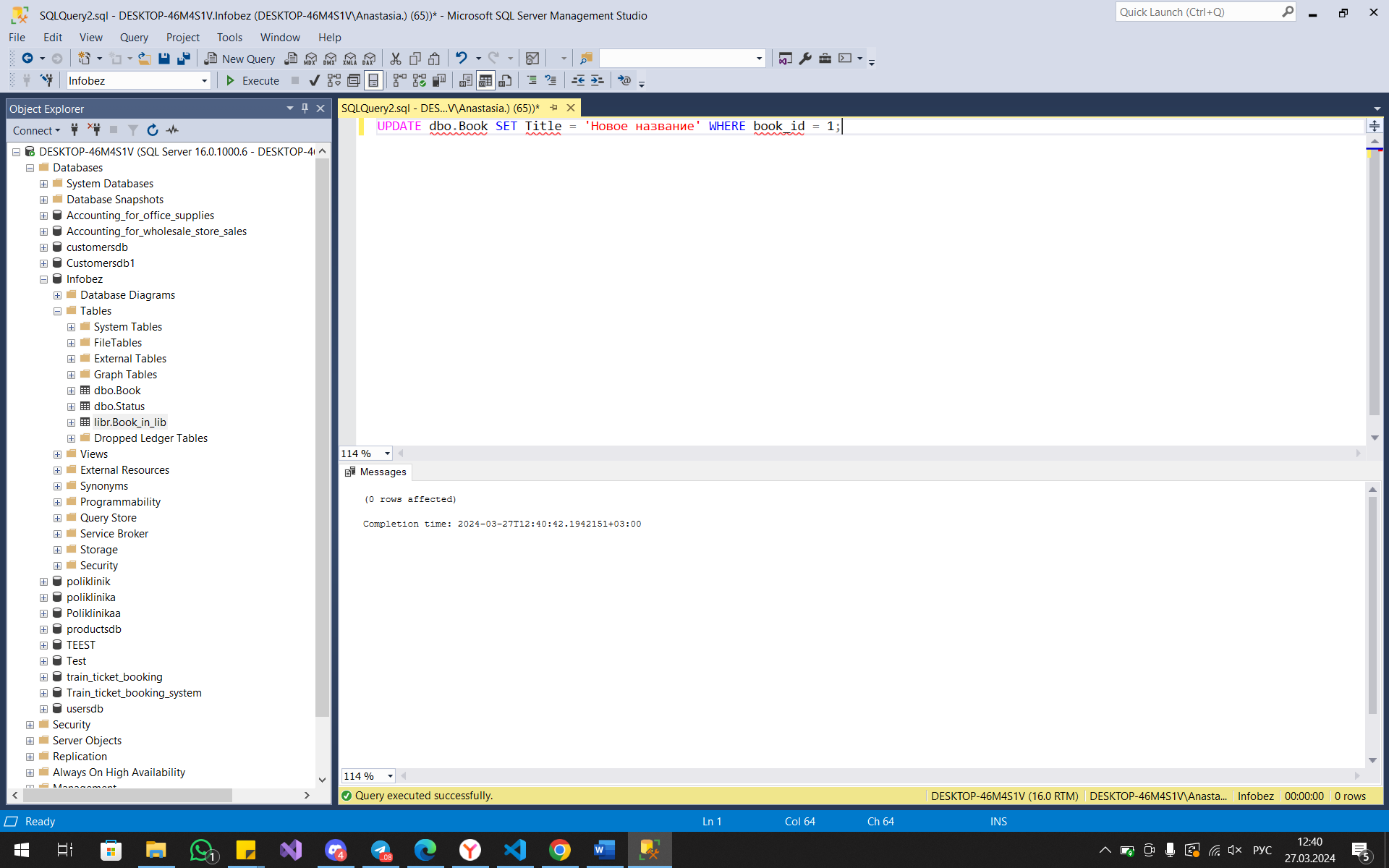


Рисунок 21. – Обновление столбцов.

Проверяем результат:

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

**Шаг 9.**

Создание представления и предоставление прав на изменение и добавление записей:

Изучение формата оператора CREATE VIEW:

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

Создание представления:

В открытом запросе выполняем оператор CREATE VIEW для создания представления. Выбираем 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;

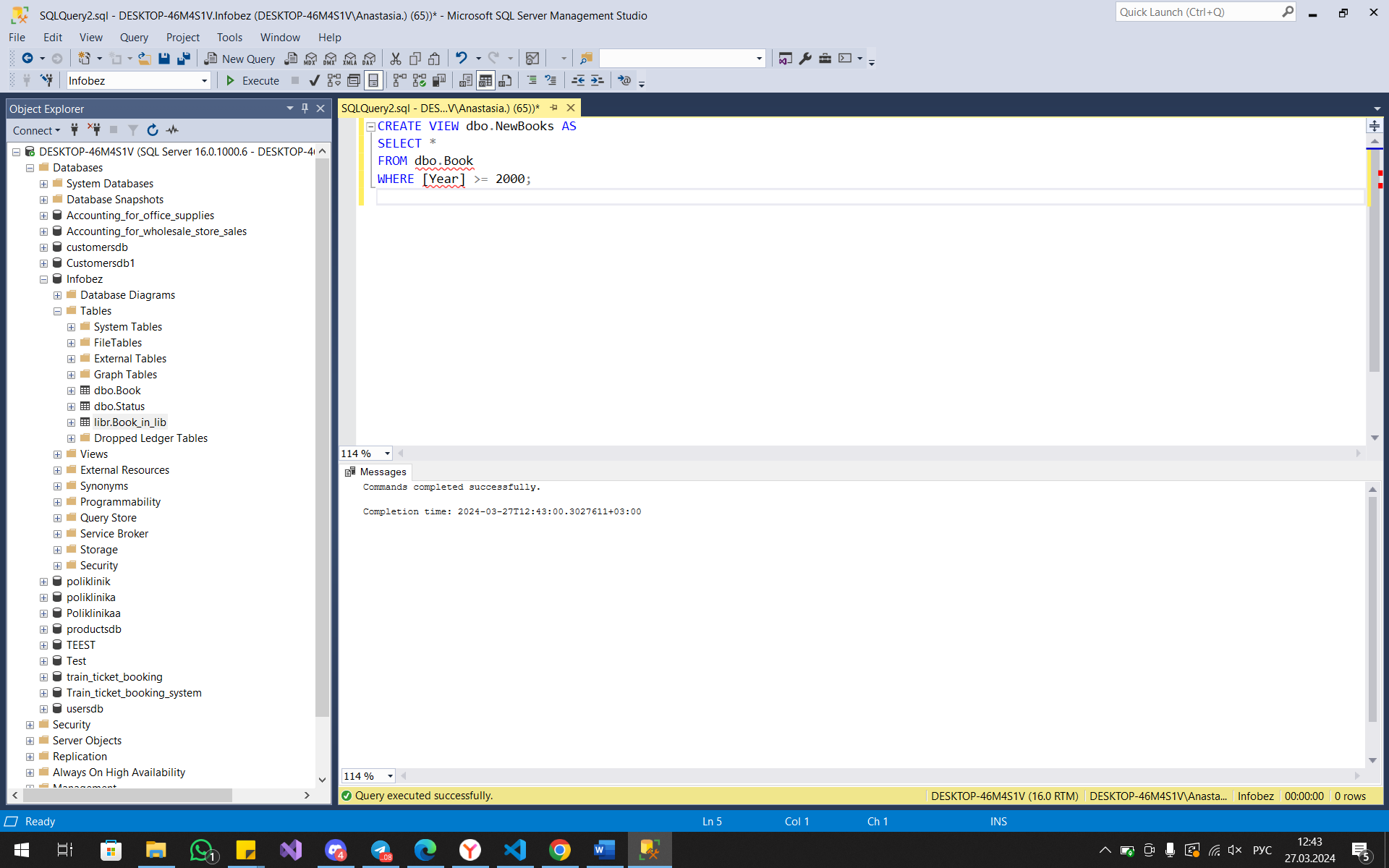


Рисунок 22. - Создания представления.

Предоставление прав на изменение и добавление записей:

Теперь предоставляем 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;

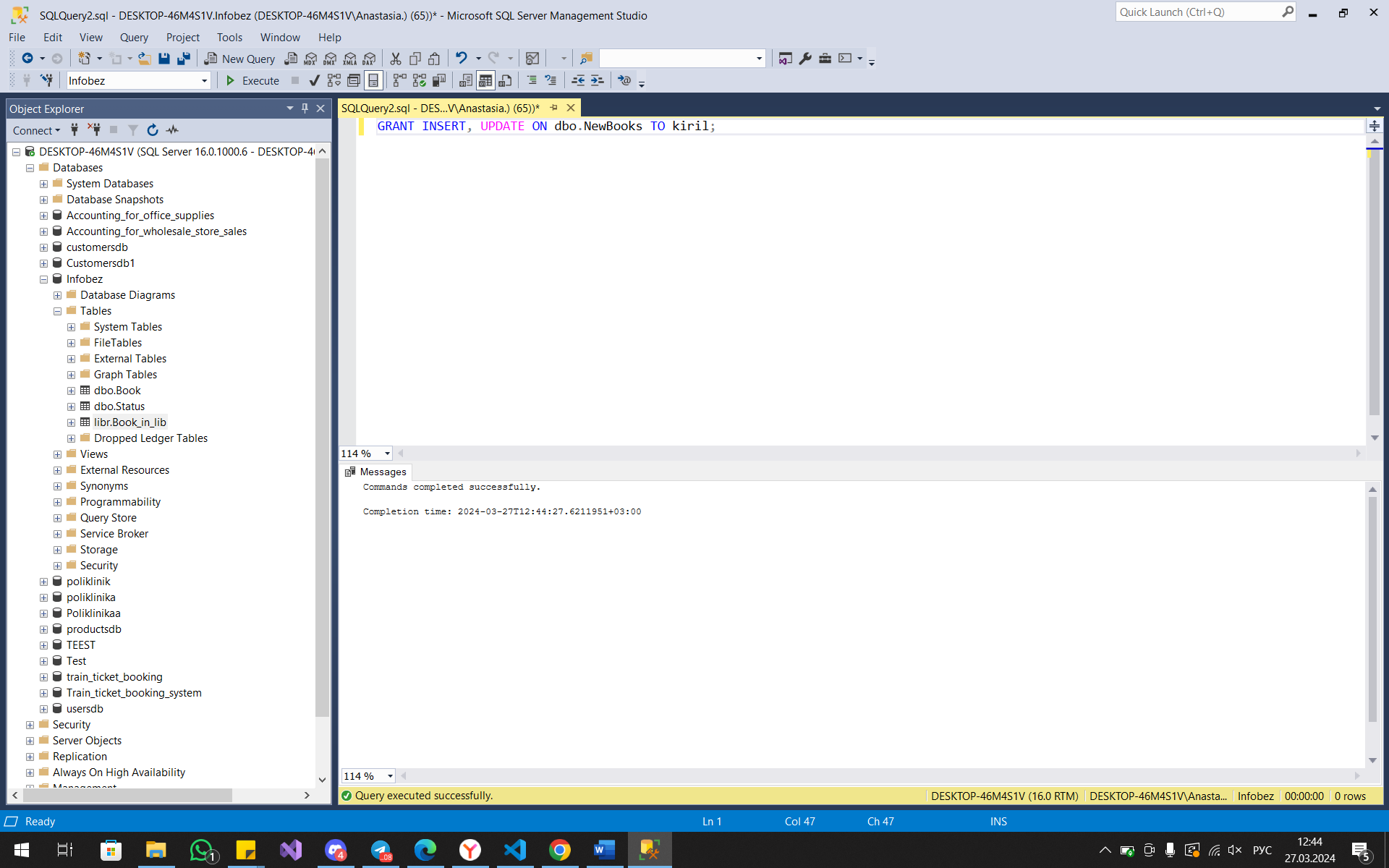


Рисунок 23. – Права на изменение записей.

Проверка прав пользователя:

Убедимся, что пользователь "kiril" 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;

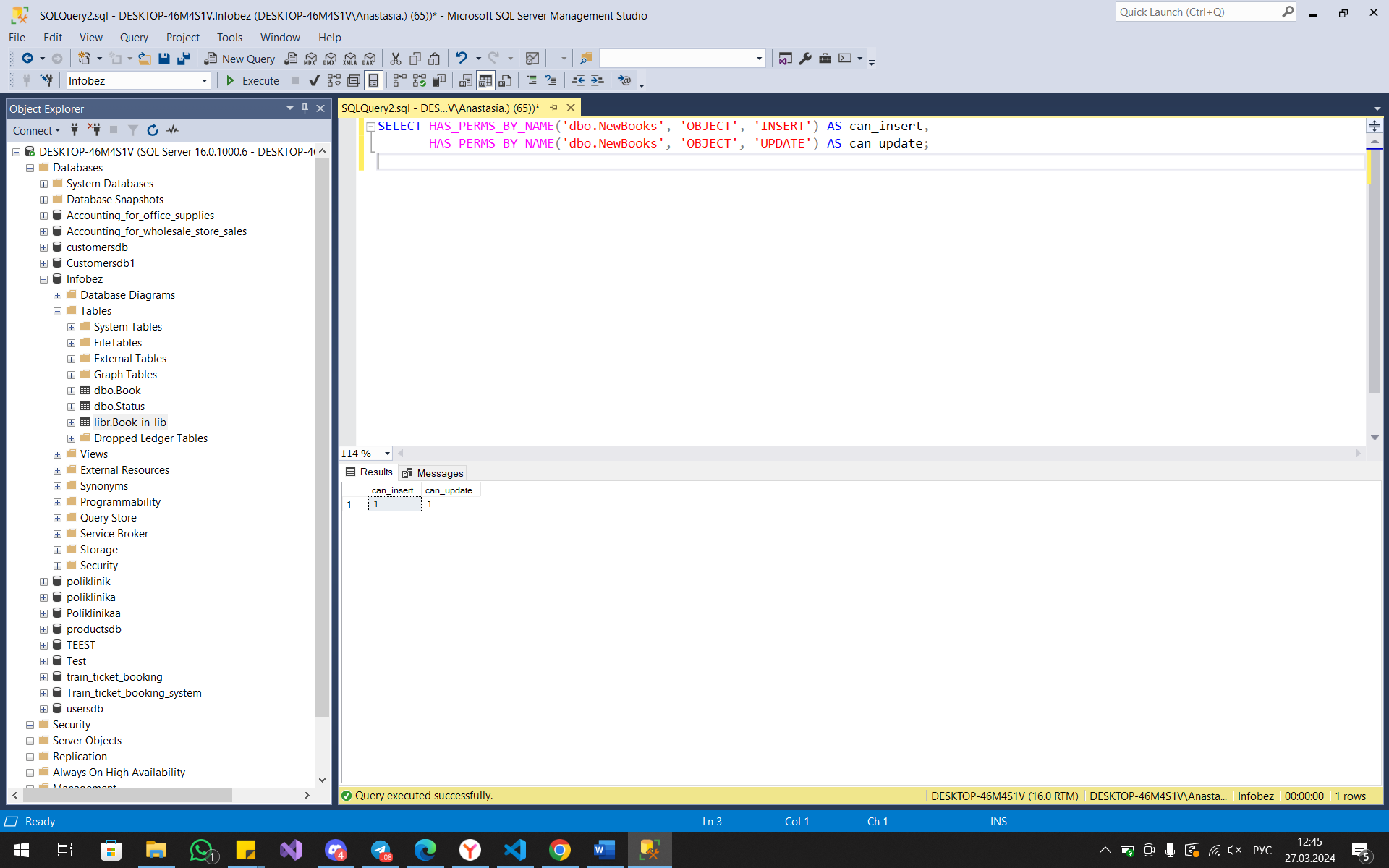


Рисунок 24. – Проверка прав пользователя.

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