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

Студенты гр.4237

Гарипов Тахир Рафилович,

Фадеев Петр Андреевич

Преподаватель

Кожевников К. Д.

Казань 2024

**МЕТОДИЧЕСКОЕ УКАЗАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ №3**

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

**Задачи:**

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

**Время на выполнение** – 4 часа.

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

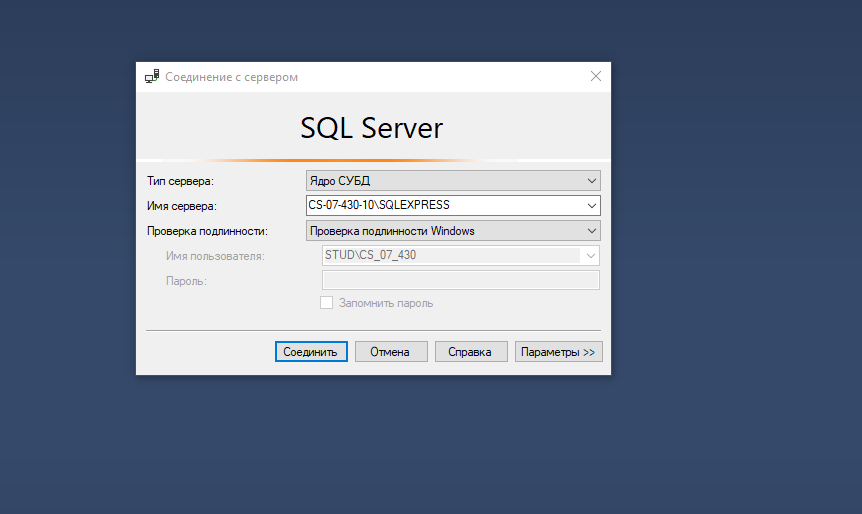


Рисунок 1. – Соединение с сервером.

Заходим в свойства,

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

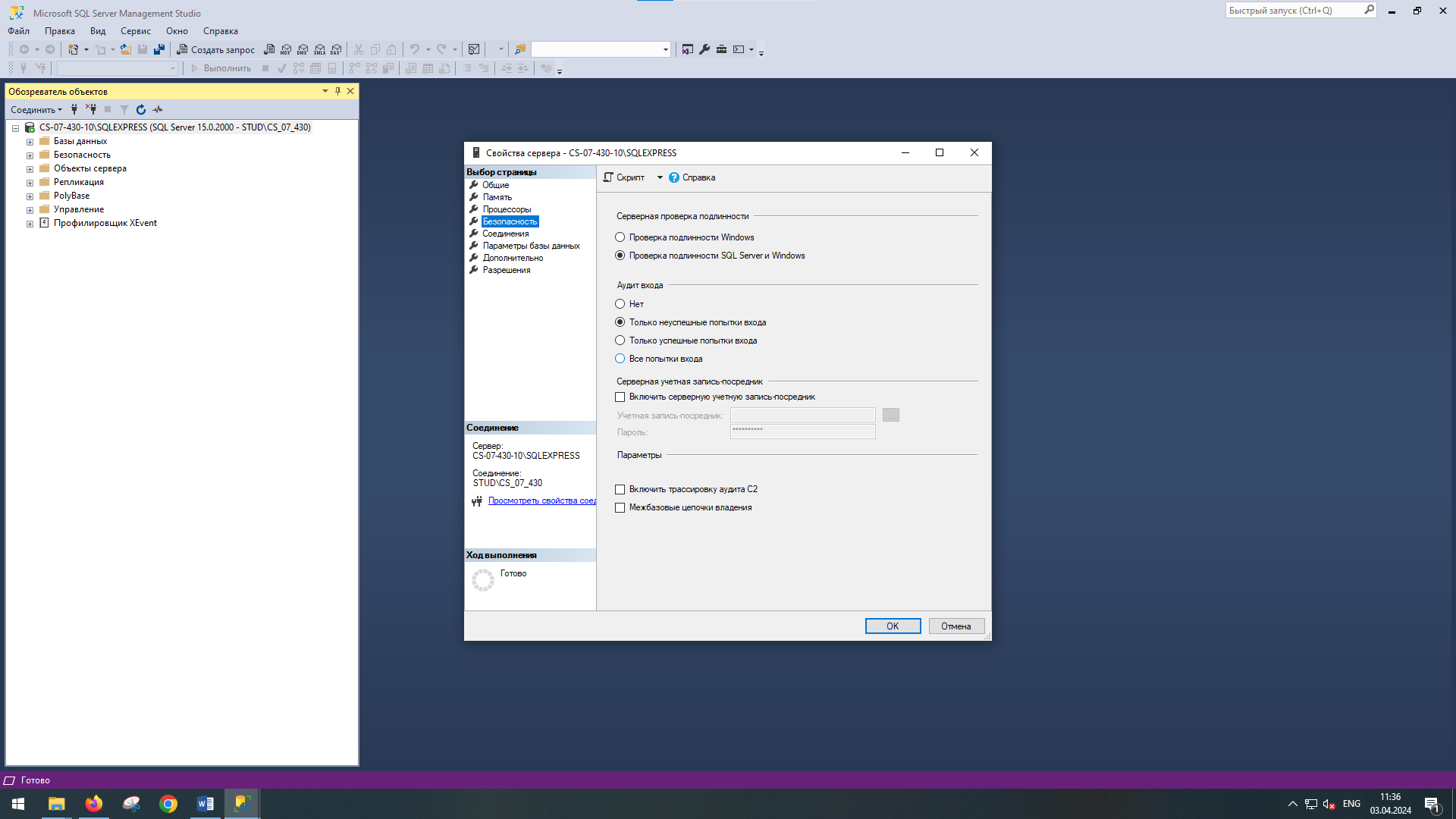


Рисунок 2. – Проверка подлинности.

В окне Object Explorer находим и разворачиваем дерево объектов, чтобы увидеть различные папки и объекты, связанные с вашим 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), нажимаем правой кнопкой мыши на нашу учетную запись и выбираем опцию "Properties" (Свойства).

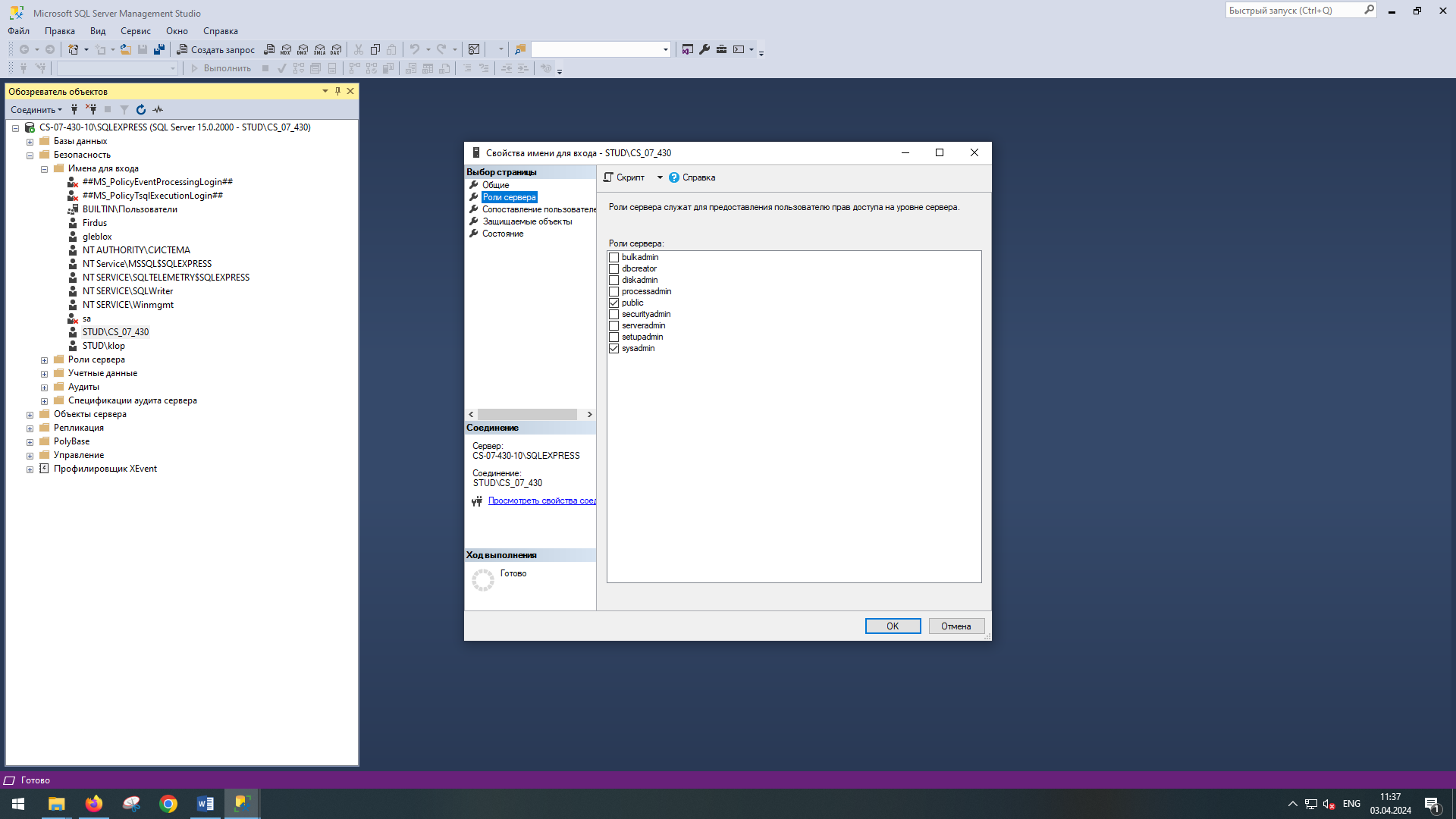


Рисунок 3. – Проверка роли сервера

Далее 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. В нашей учетной записи сопоставлены пользователи.

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

Переходим на вкладку "Membership" (Членство). Ознакамливаемся с информацией.

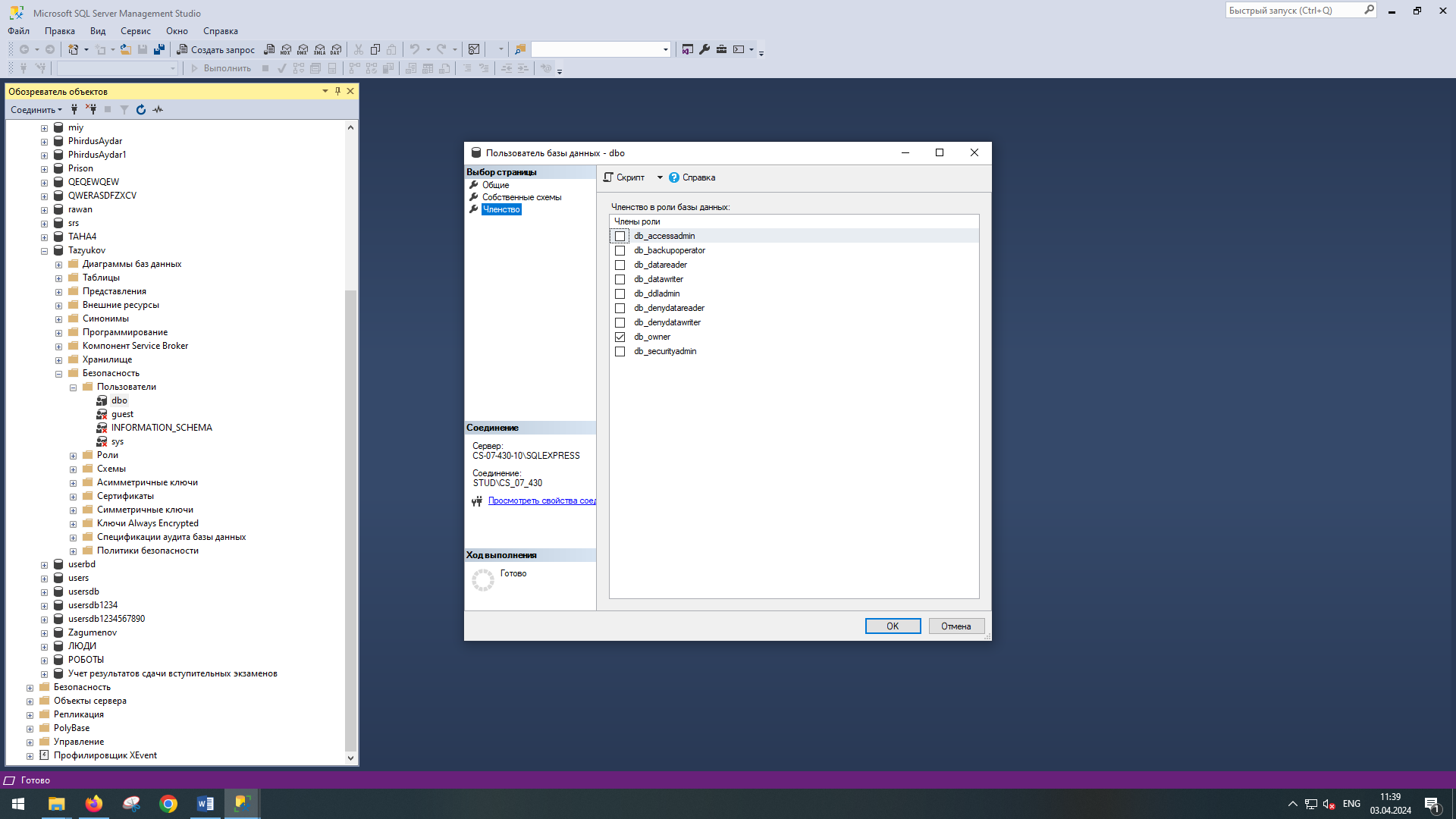


Рисунок 4. – Членство.

Создаем новую БД. Открываем список пользователей и ролей.

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

Выбираем опцию "New Database" (Новая база данных). В появившемся диалоговом окне вводим имя новой БД.

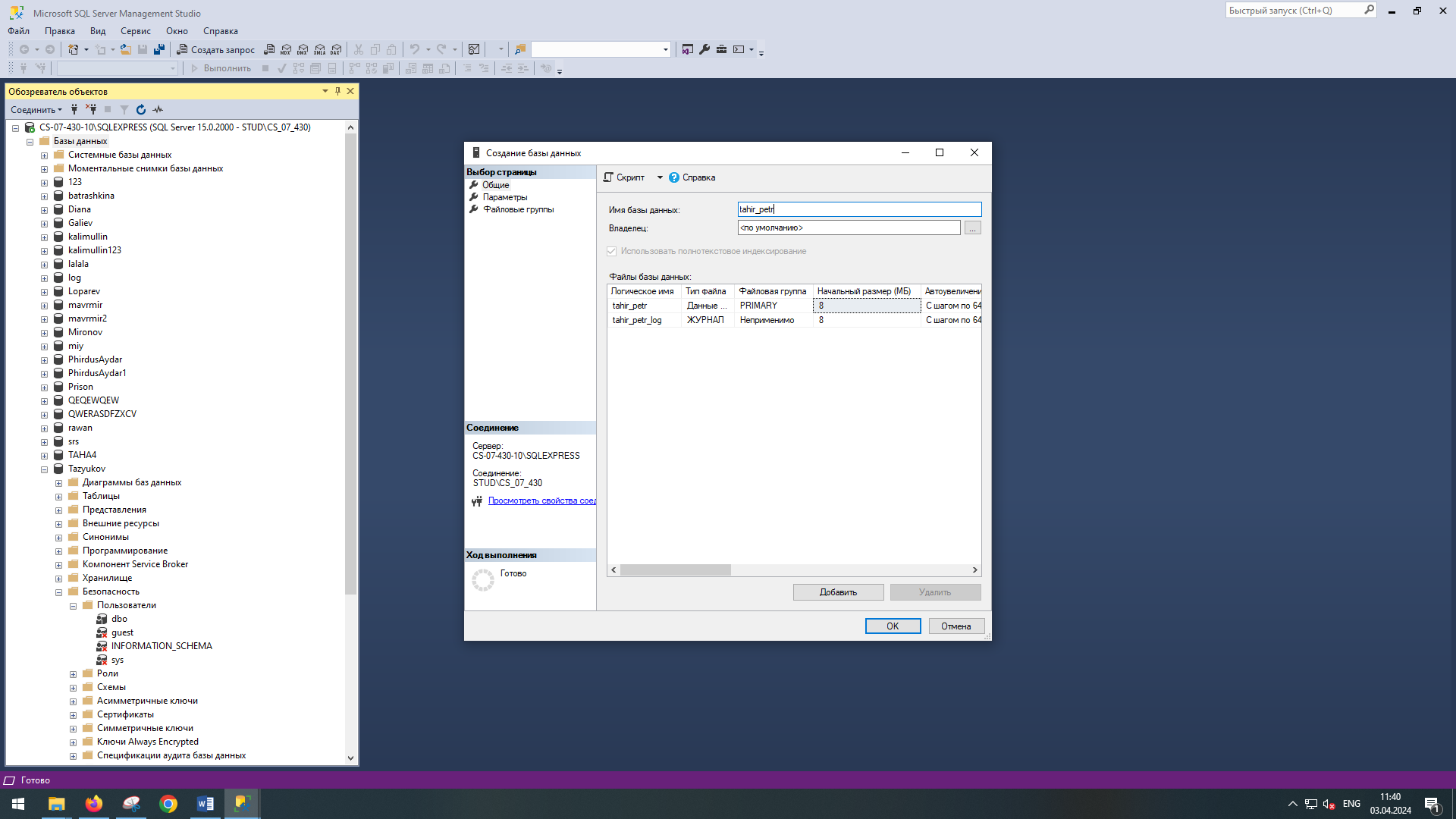


Рисунок 5 – Имя БД.

Далее проверяем список пользователей и ролей.

Разворачиваем созданную нами новую базу данных в списке баз данных.

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

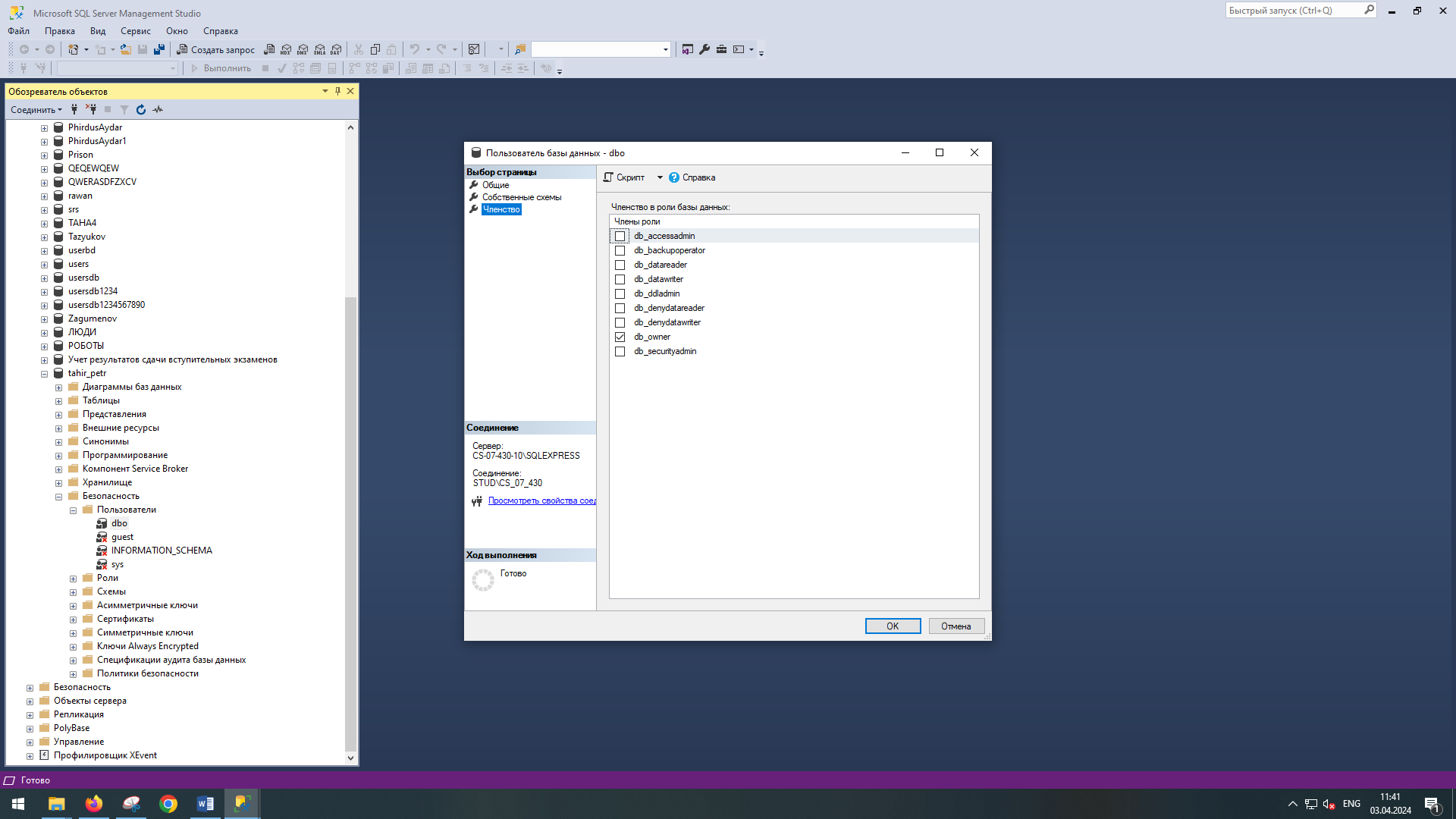


Рисунок 6. – Членство.

Создаем таблицы в созданной БД.

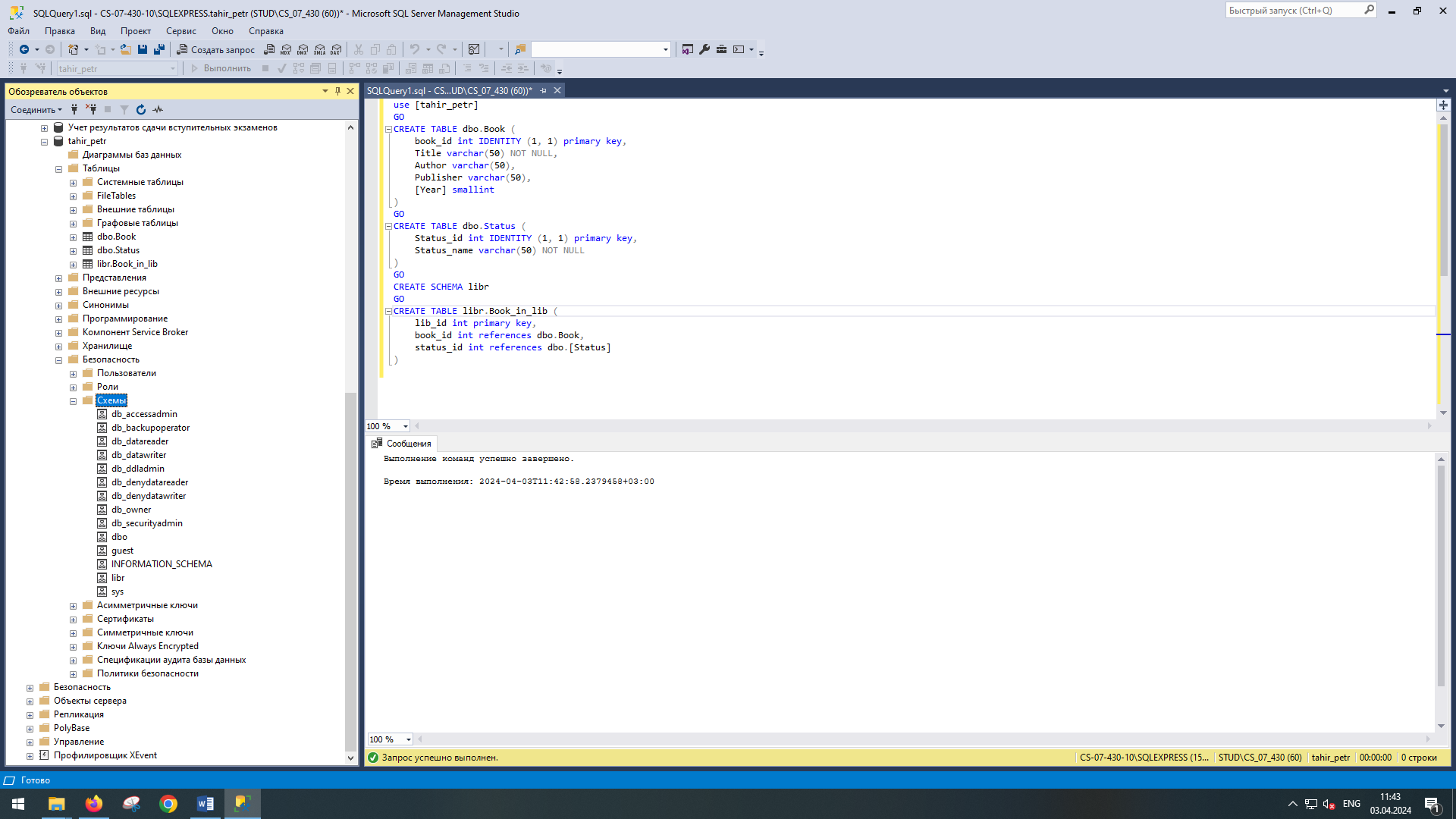


Рисунок 7. – Создание таблиц.

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.

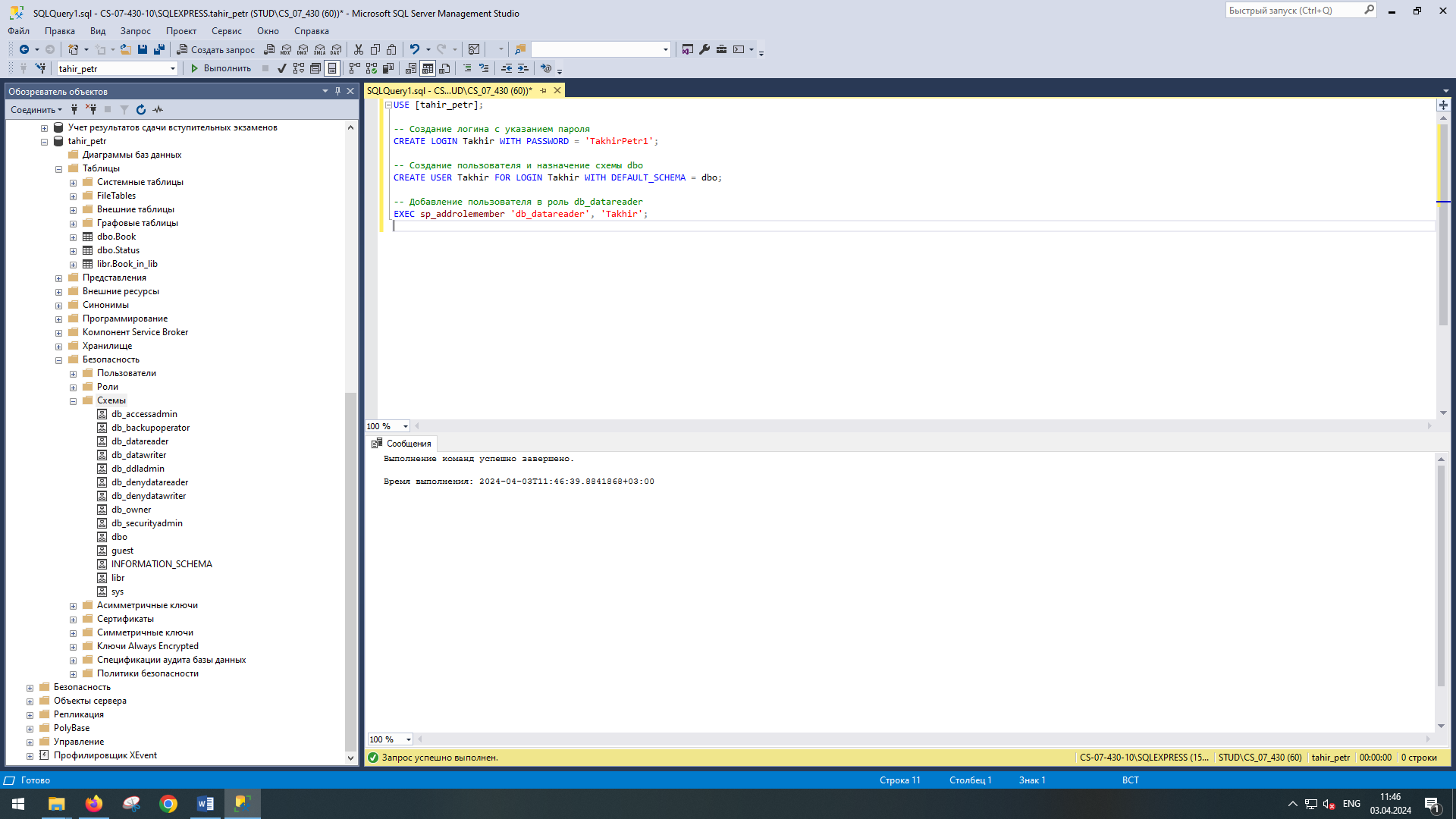


Рисунок 8. – Скрипт.

Далее подключаемся к серверу с учетной записью другого пользователя.

Попробуем добавить или изменить данные в таблицах.

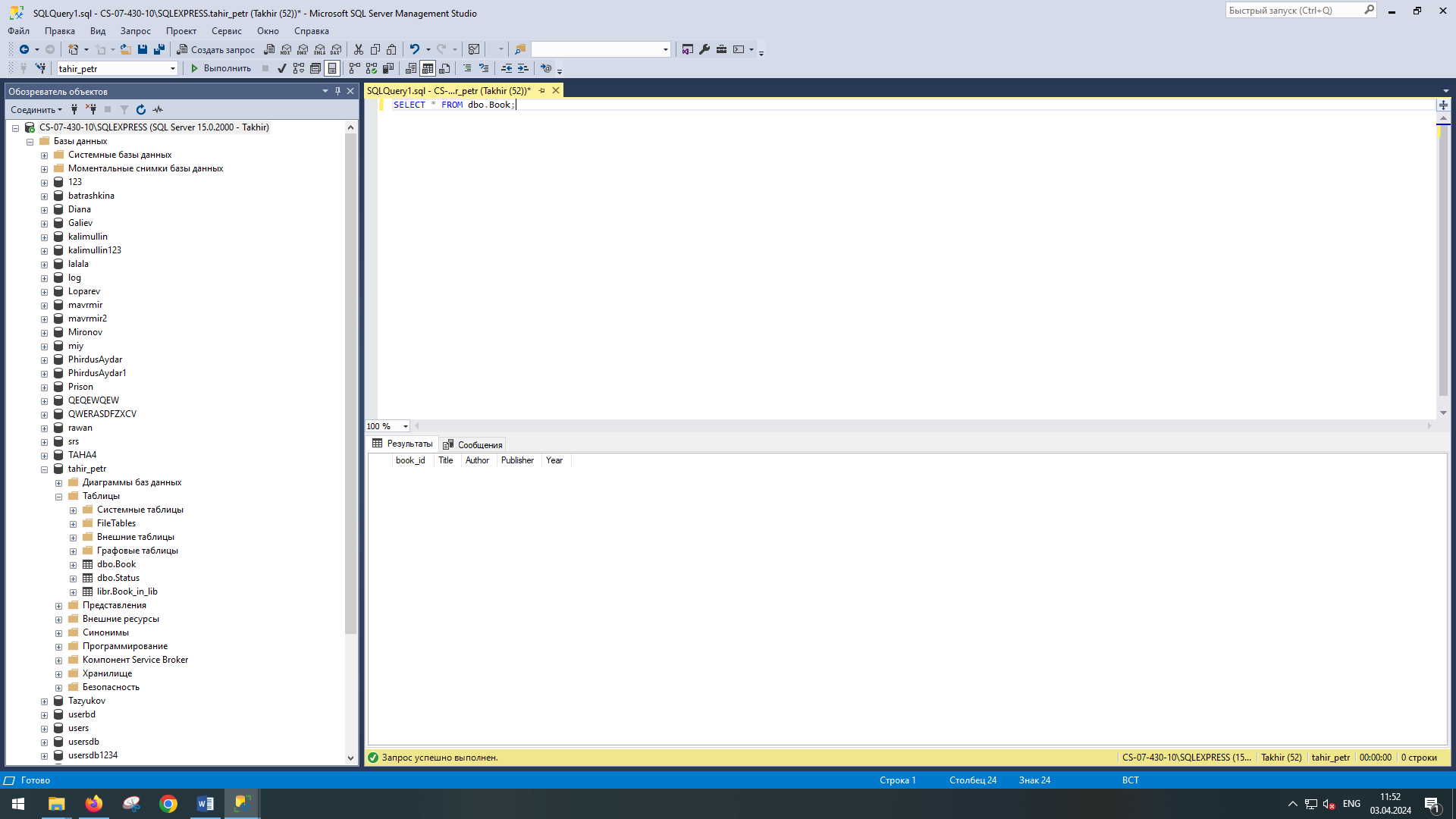


Рисунок 9. – Проверка со стороны пользователя.

Создание новой роли и назначение разрешений:

Подключитесь к серверу базы данных:

Используем административные учетные данные для подключения к серверу баз данных.

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

CREATE ROLE libr\_writer;

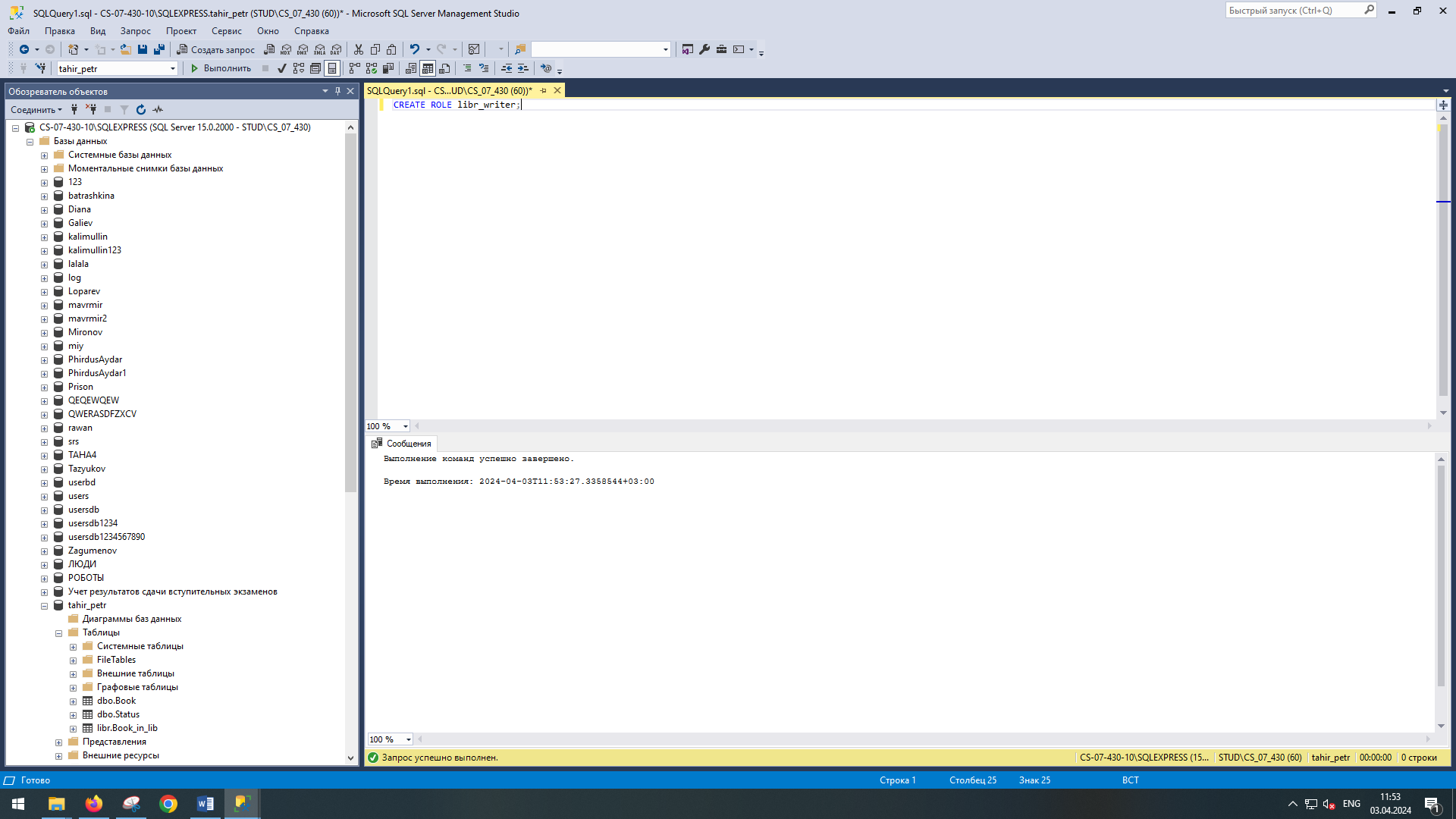


Рисунок 10. – Новая роль.

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

Назначаем разрешения роли для схемы "libr":

Введим 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

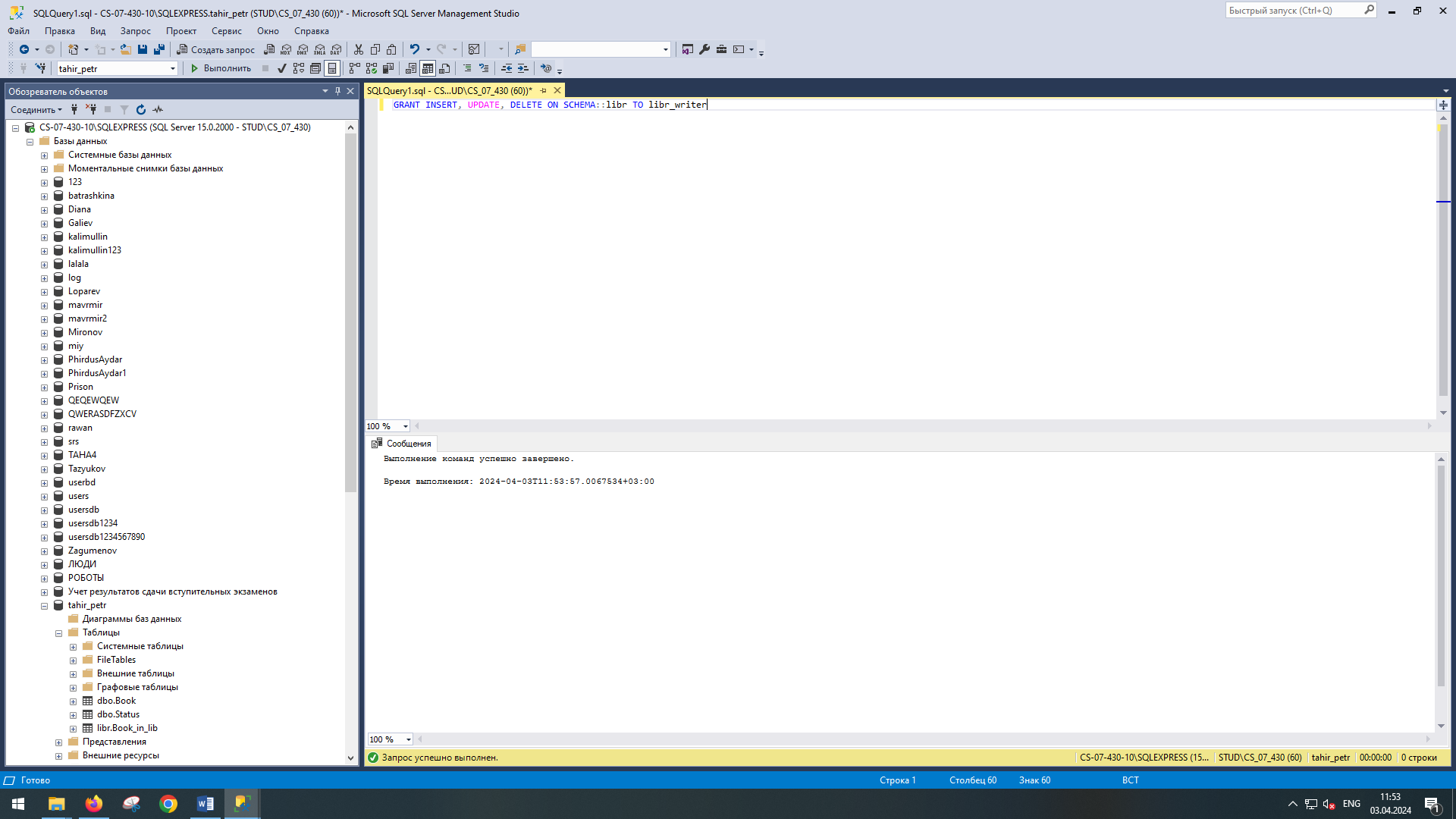


Рисунок 11. – Разращение роли.

Добавляем пользователя в созданную роль:

Введим следующий запрос, чтобы добавить пользователя "Takhir" в роль "libr\_writer":

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

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

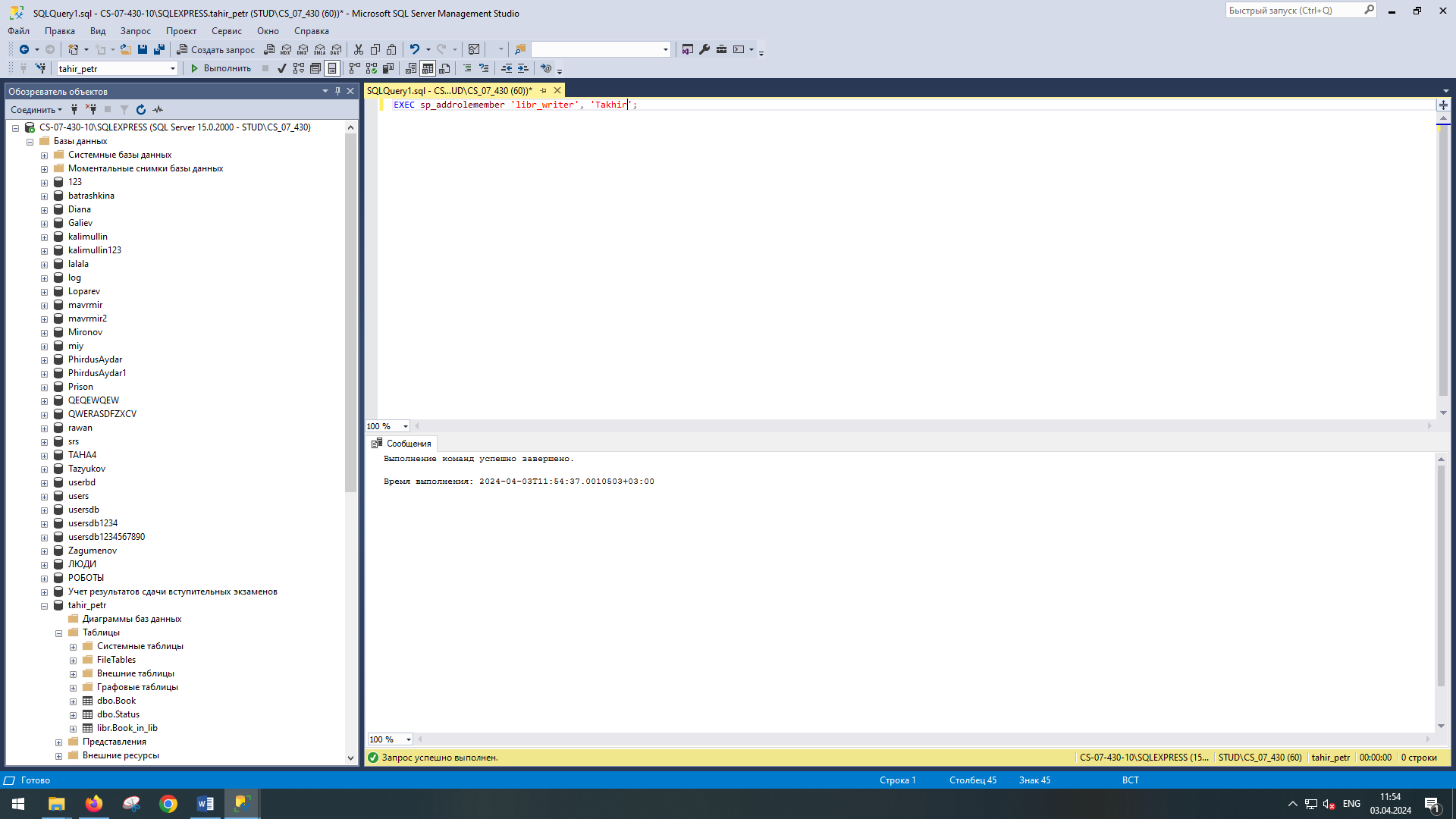


Рисунок 12. – Назначение роли.

Проверяем доступ пользователя к таблице:

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

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

Назначаем разрешения для пользователя “Takhir” как на рисунке 12.

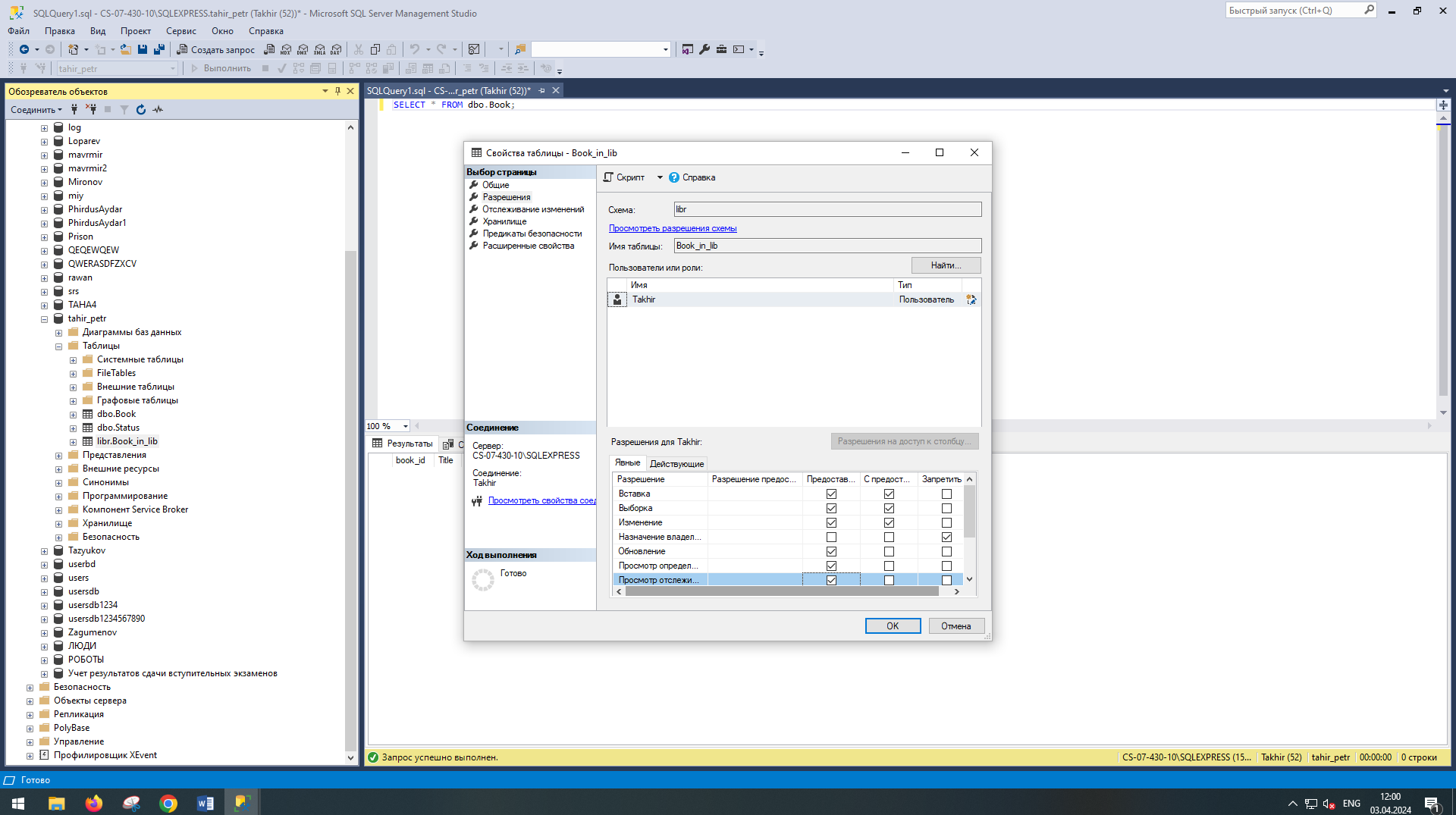


Рисунок 13. – Настройка разращений.

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, щелкаем правой кнопкой мыши и выбираем "New Query" (Новый запрос).

Выполняем запрос на предоставление прав:

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

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

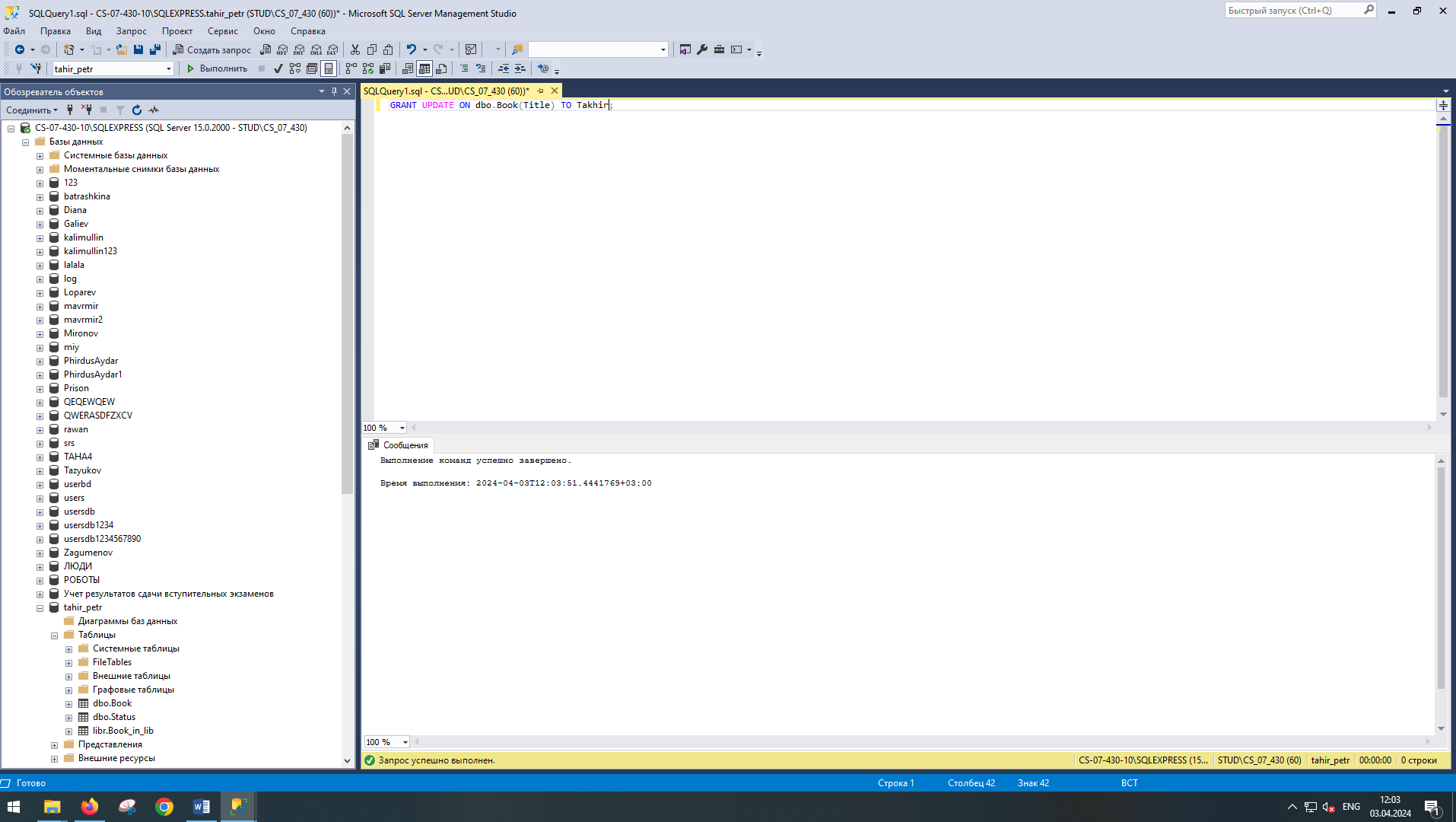


Рисунок 14. – Предоставление право.

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

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

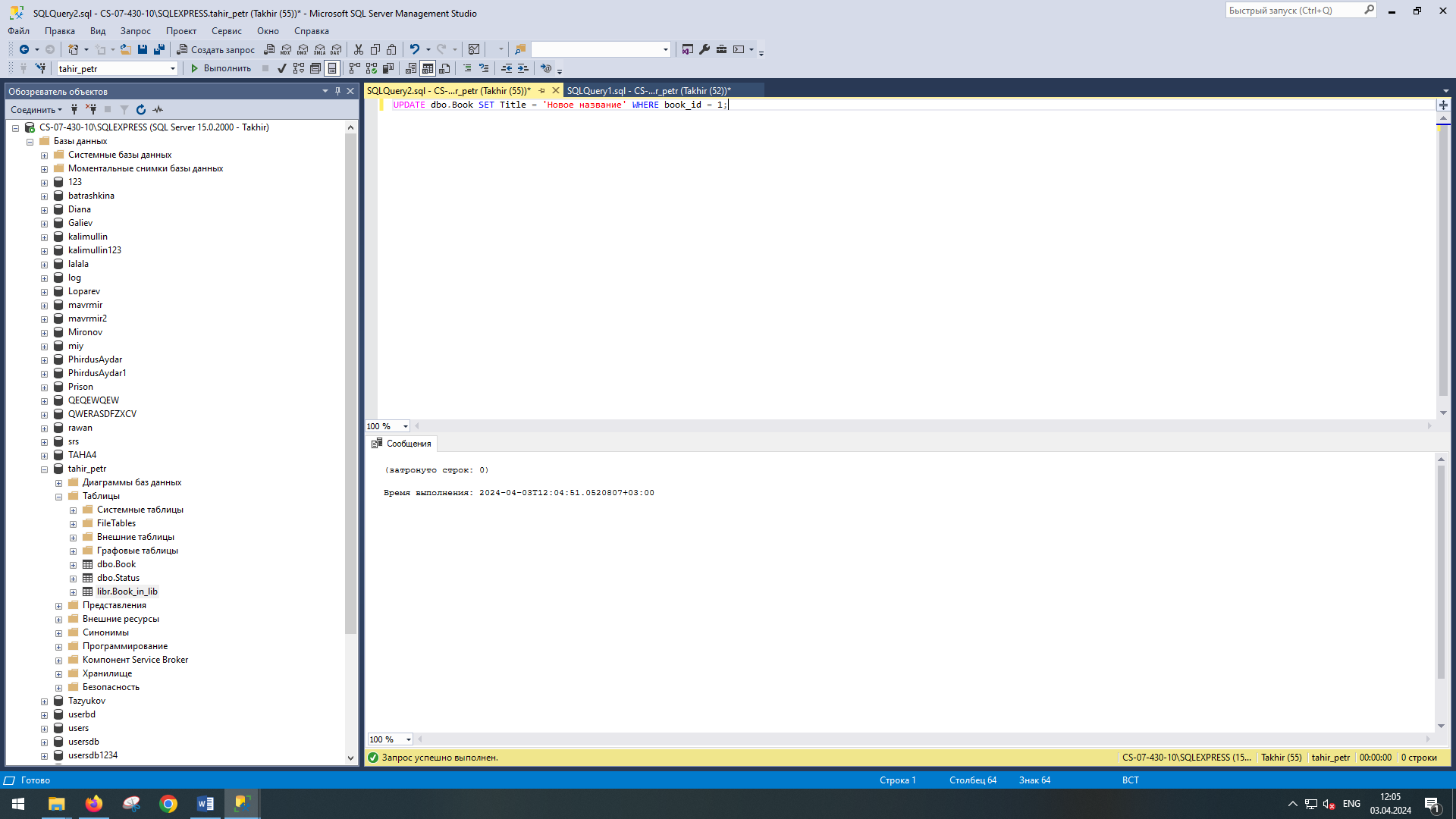


Рисунок 15. – Обновление столбцов

Проверяем результат:

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

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;

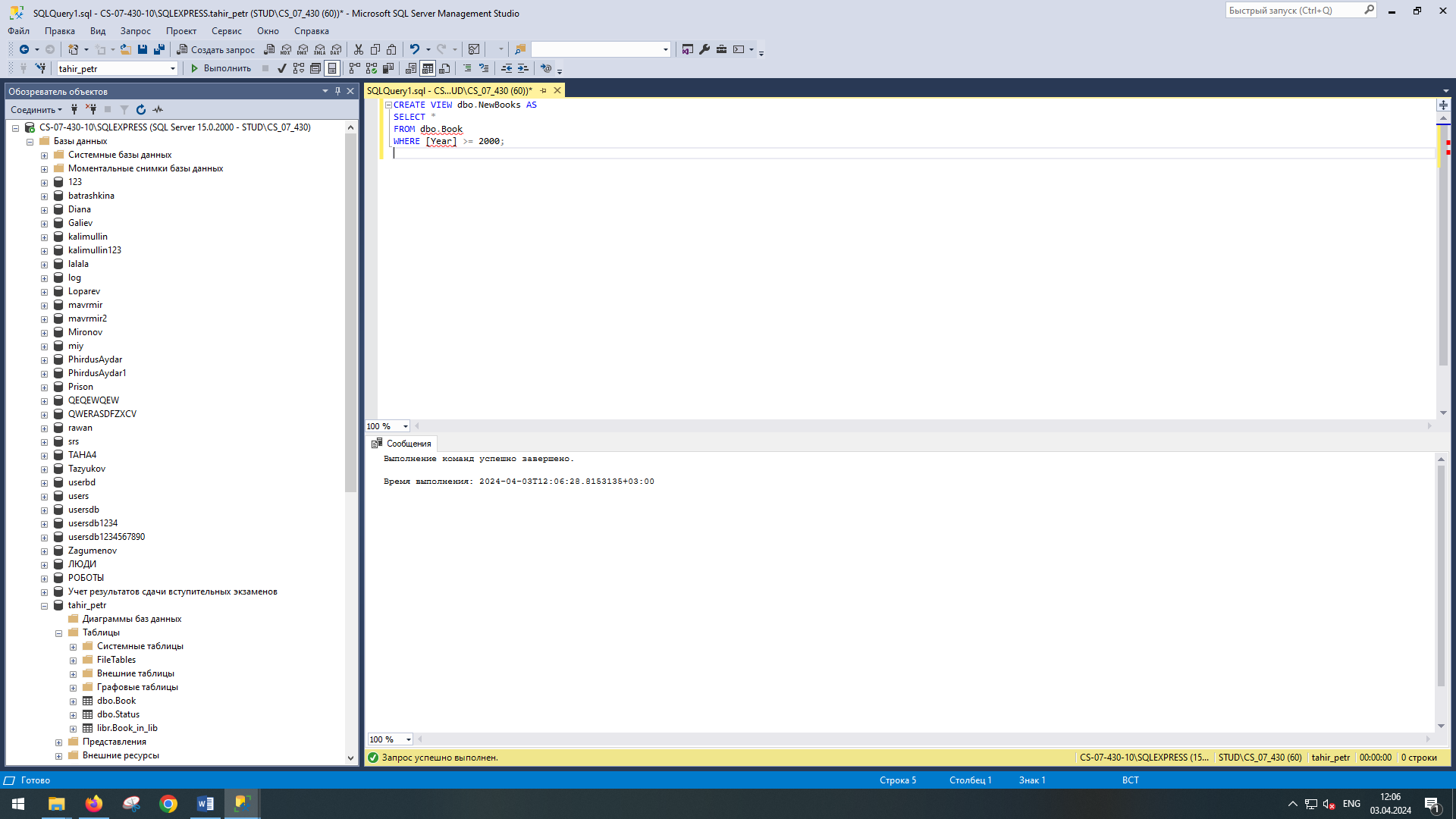


Рисунок 16. - Создания представления.

Теперь предоставляем пользователю "Takhir" 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 Takhir;

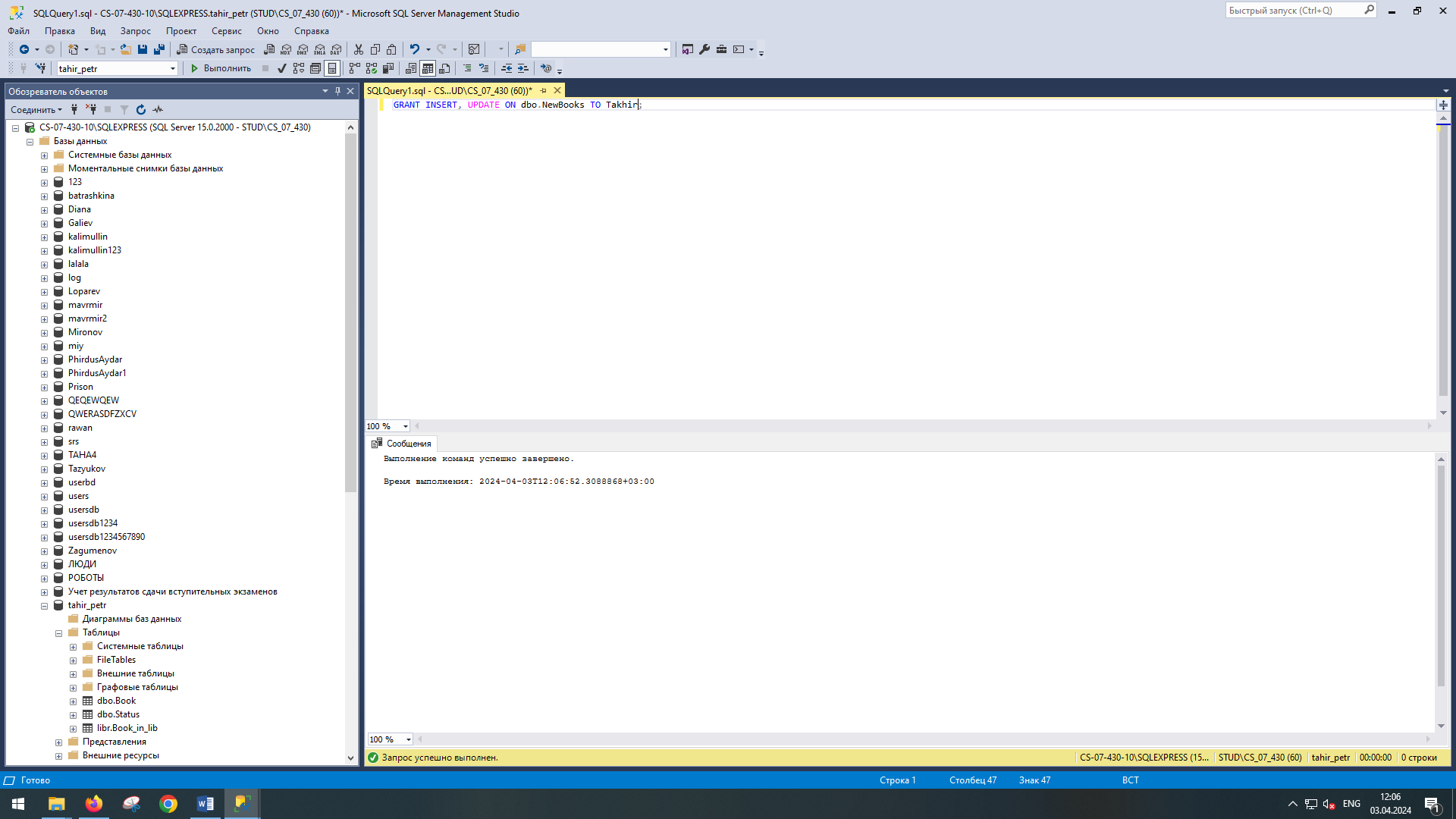


Рисунок 17. – Права на изменение записей.

Далее убеждаемся, что пользователь "Takhir" 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;

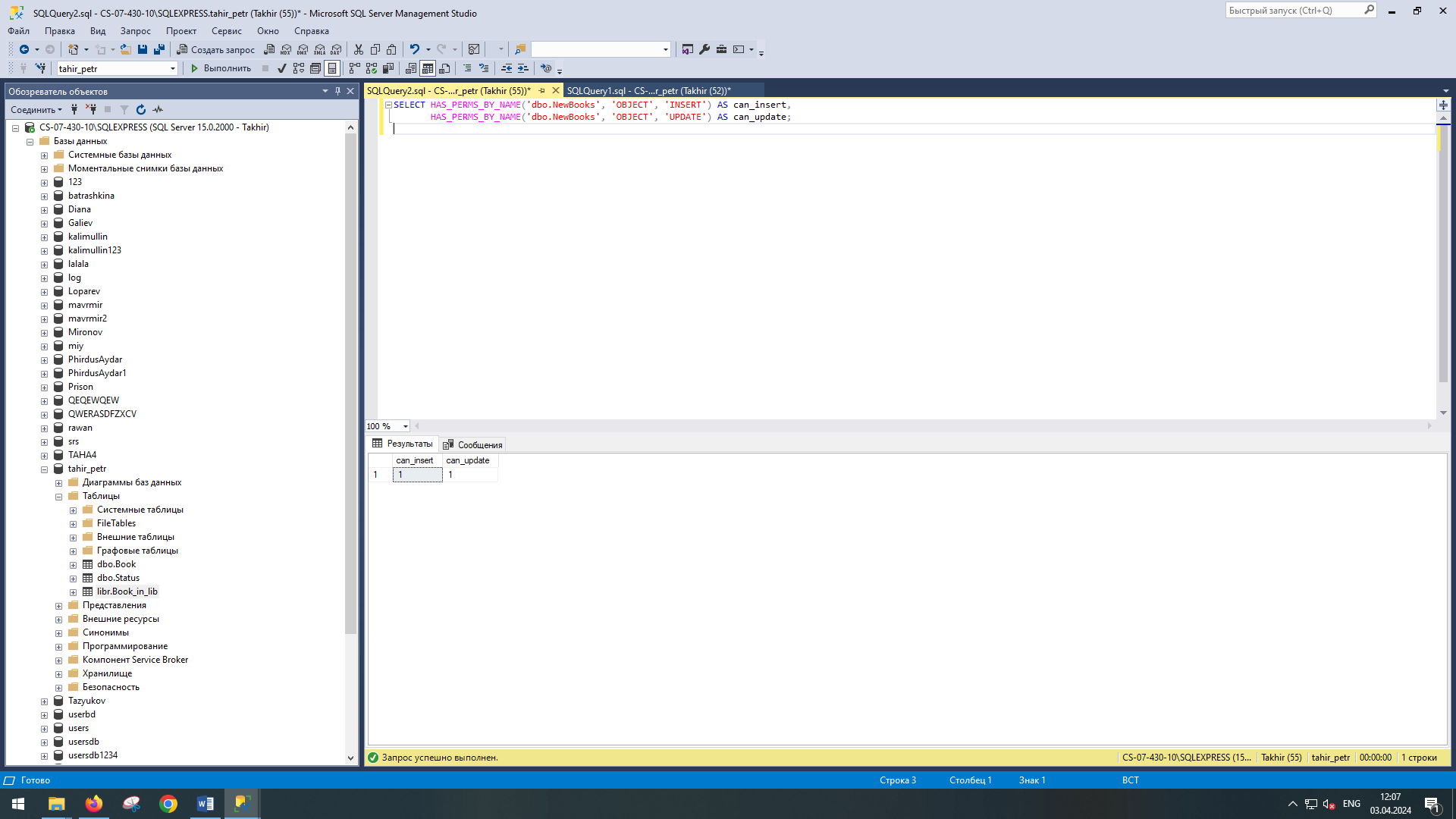


Рисунок 18. – Проверка прав пользователя.

**Вывод**

В процессе выполнения данной лабораторной работы мы научились реализовывать доступ пользователей к базе данных.