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 №4

по дисциплине

“Информационная безопасность”

# Тема: “ Установка приоритетов”

Работу выполнил

Студент гр.4233

Гарифуллин И. И.

Преподаватель

Кожевников К. Д.

Казань 2024

**Цель работы**

Изучить способы установки приоритетов.

**Задачи на лабораторную работу**

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

**Результат выполнения работы**

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

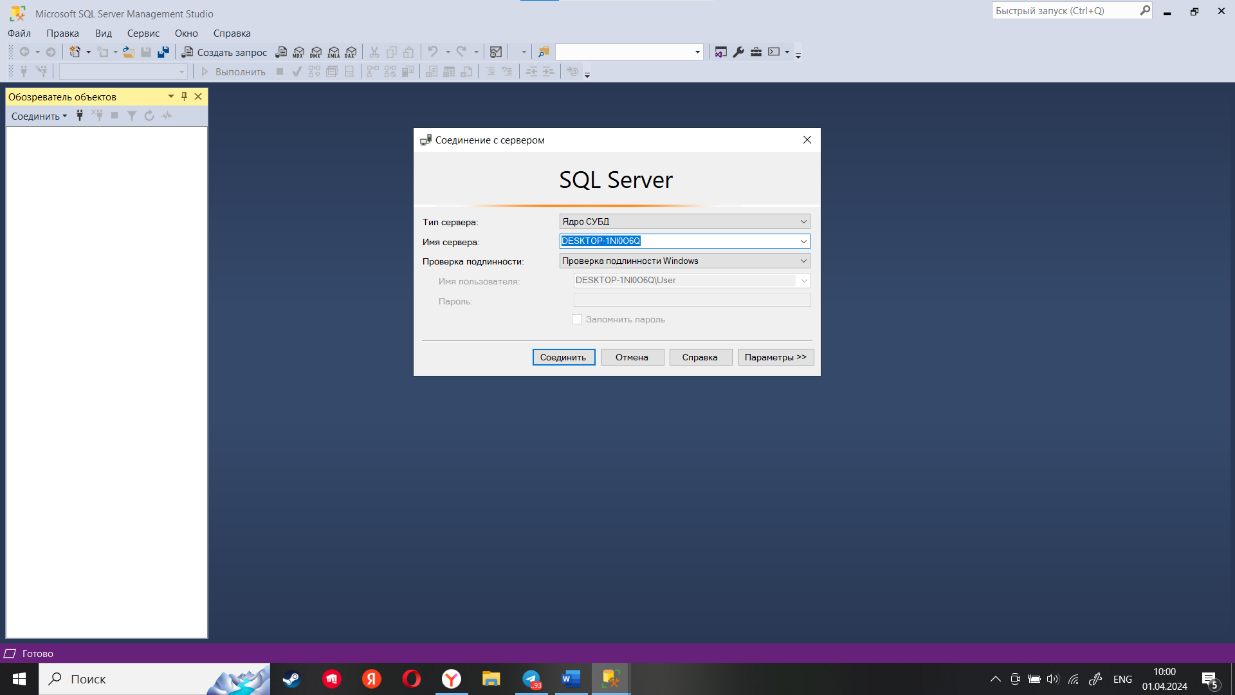


Рисунок 1 - Соединение с сервером

Заходим в свойства.

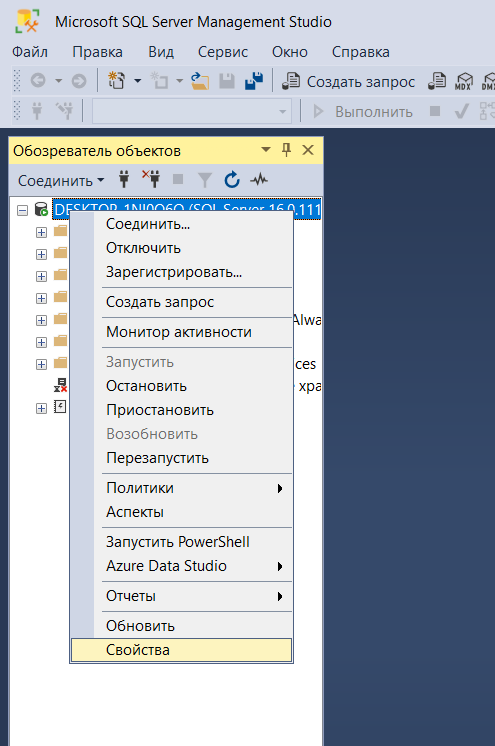


Рисунок 2 – Свойства

В открывшемся окне свойств сервера перейдем на вкладку "Процессоры".

В разделе "Потоки" установим флажок рядом с "Повысить приоритет SQL Server".

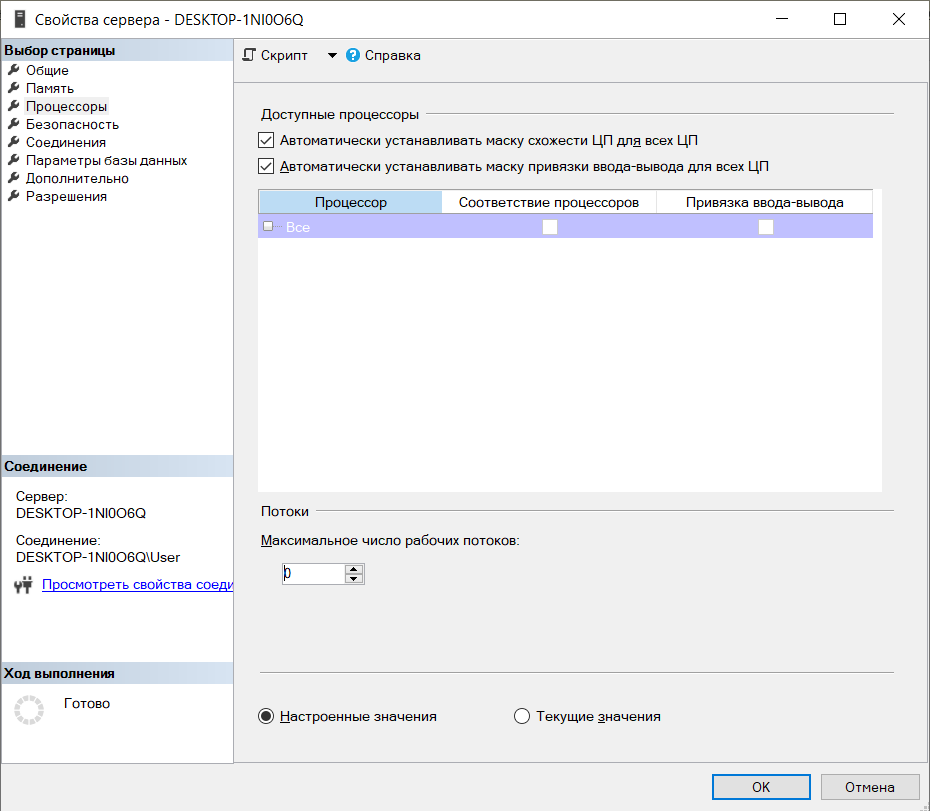


Рисунок 3 – Установка флажка

После этого остановим и снова запустим SQL Server, чтобы изменения вступили в силу. Эти действия позволят настроить параметр повышения приоритета через графический интерфейс SQL Server Management Studio.

Если флажка "Повысить приоритет SQL Server" нет, то это может быть связано с версией SQL Server или с настройками вашего сервера. Возможно, этот параметр не доступен в вашей текущей конфигурации.

Как это произошло и у меня. По это тому, ничего страшного, переходим ко второму шагу.

Напишем запрос, использование процедуры sp\_configure для задания значения параметра priority boost равным 1.

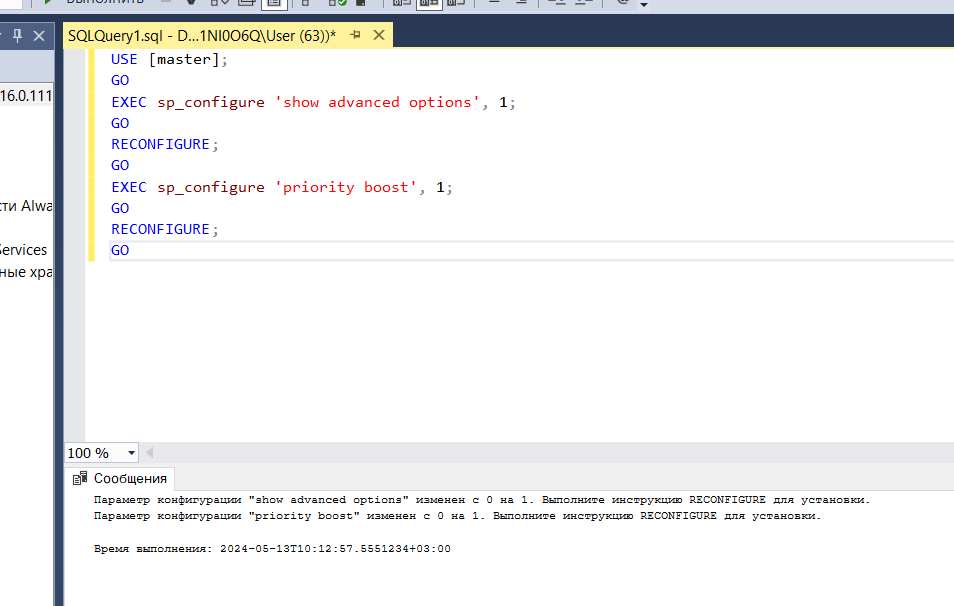


Рисунок 4 – Запрос

После перезагружаем сервер, чтобы изменения вступили в силу.

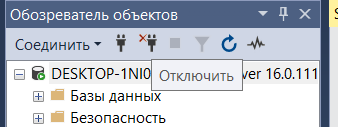


Рисунок 5 – Отключить.

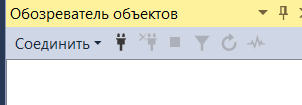


Рисунок 6 – Подключить.

После выполнения этого скрипта параметр приоритета будет настроен в соответствии с указанными значениями.

Напишем запрос для проверки значения параметра priority boost

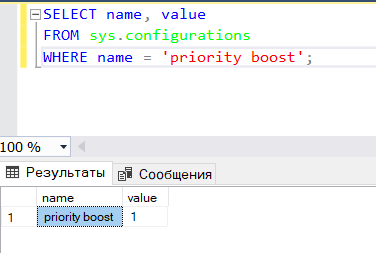


Рисунок 7 – Запрос.

Вывод

Мы изучили способы установки приоритетов.

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

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

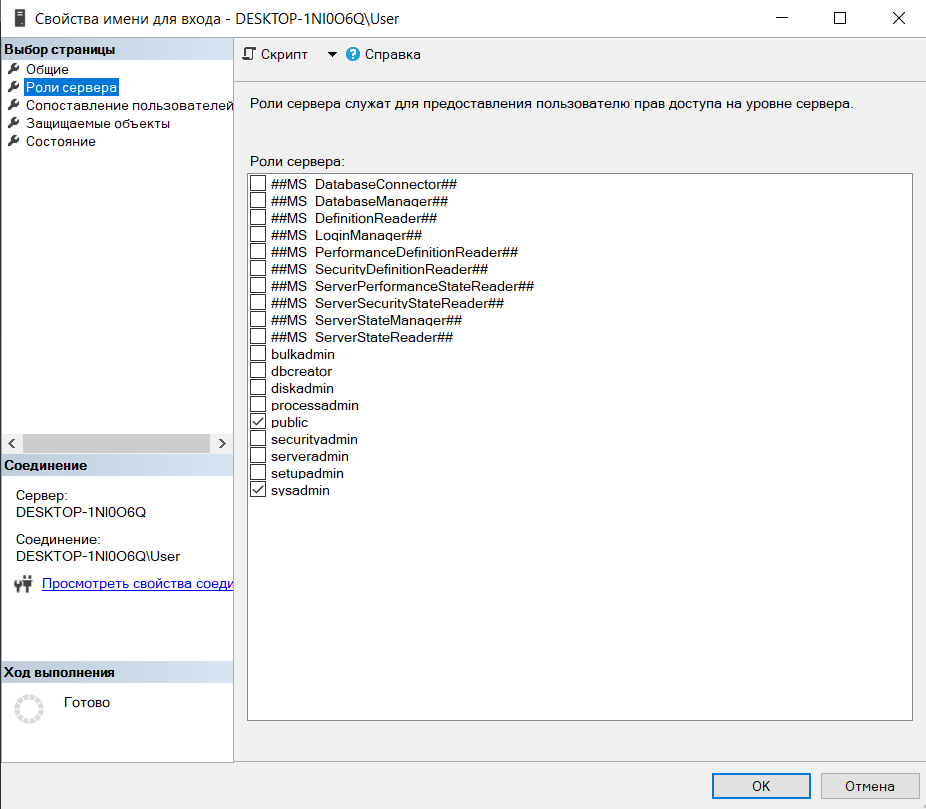


Рисунок 5 – Проверка роли сервера

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

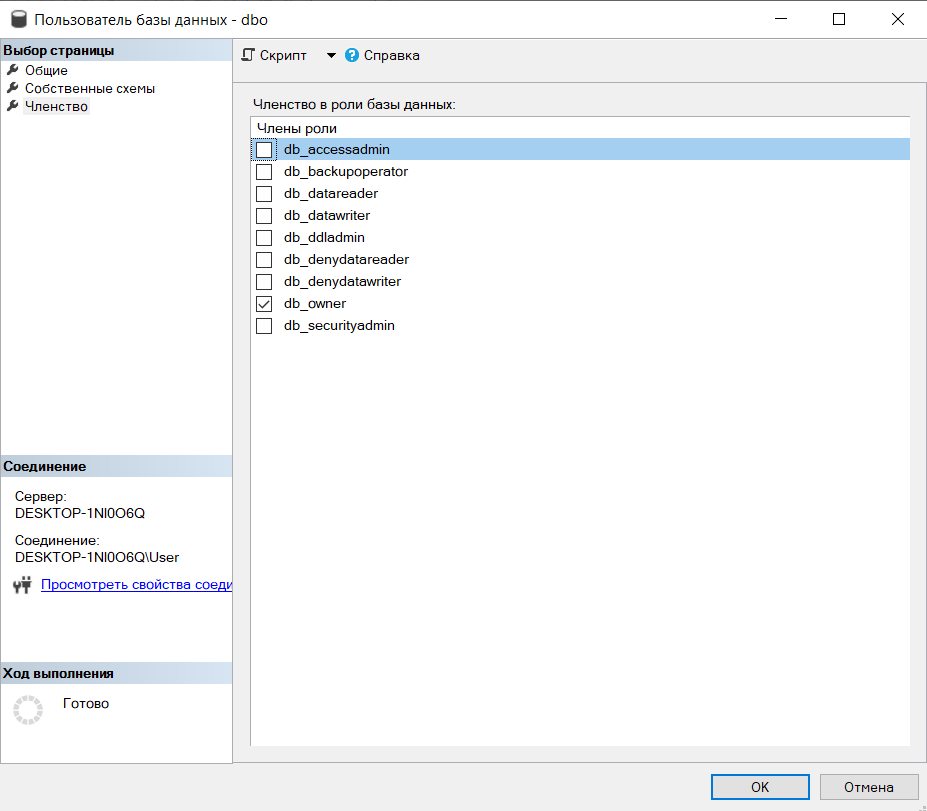


Рисунок 6 – Членство

Создадим новую базу данных.

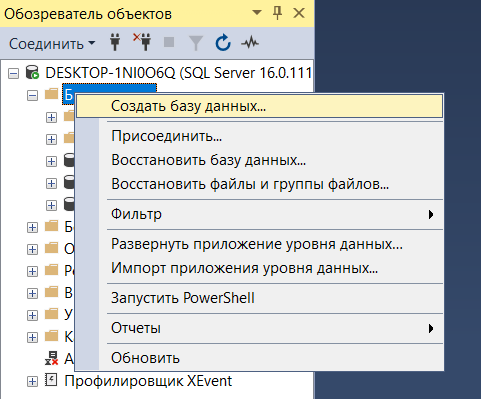


Рисунок 7 – Создание БД

В появившемся диалоговом окне введите имя для новой базы. Нажмем "OK", чтобы создать новую базу данных.

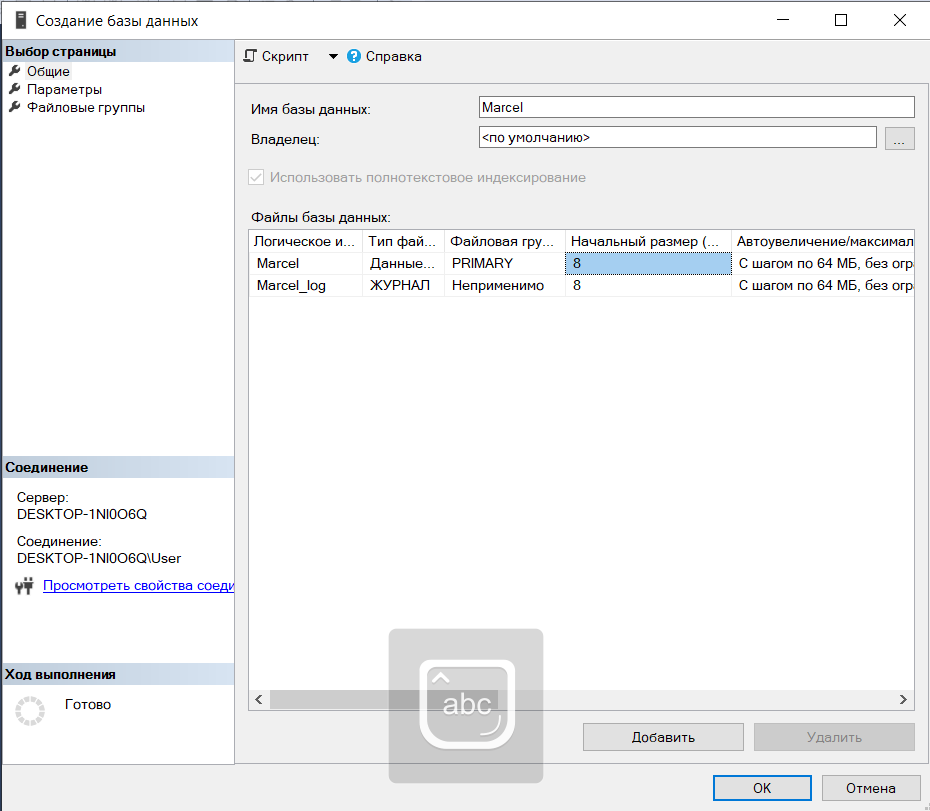


Рисунок 8 – Имя БД

Щёлкнем правой кнопкой мыши на пользователя "dbo" и выберем опцию "Properties" (Свойства). Перейдем на вкладку "Membership" (Членство) в окне свойств пользователя "dbo". Убедимся, что роль "db\_owner" присутствует в списке ролей, к которым этот пользователь авторизован.

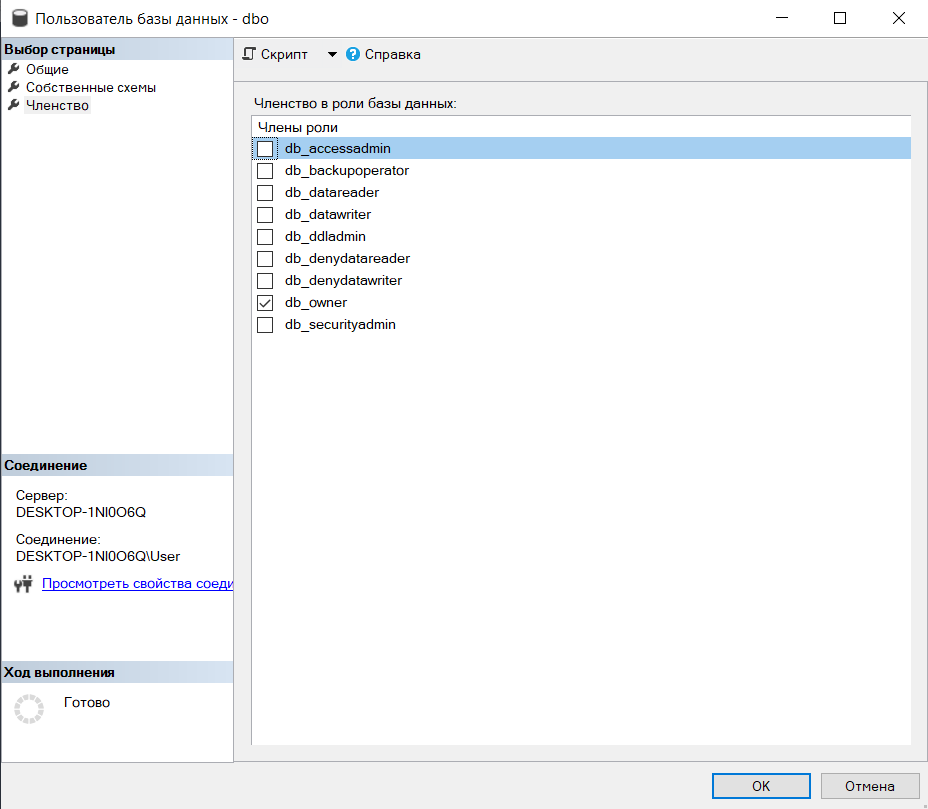


Рисунок 9 – Членство

Используя приведенный данный скрипт, создаем в базе данных таблицы

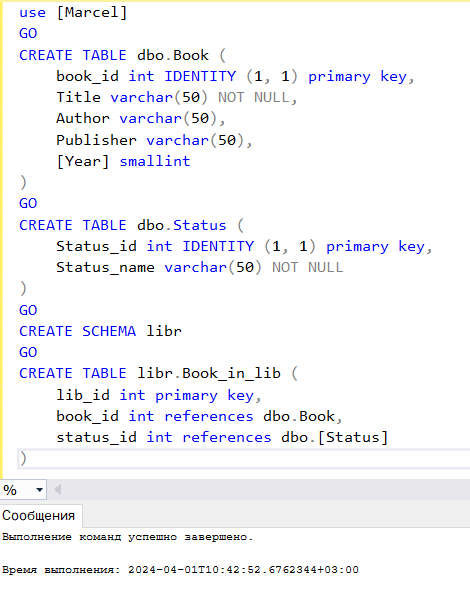


Рисунок 10 – Создание таблиц

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

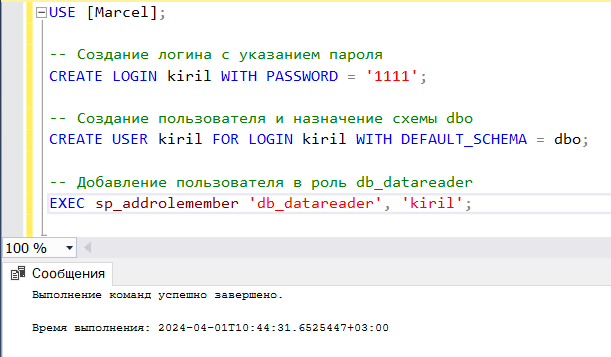
****

Рисунок 11 - Скрипт

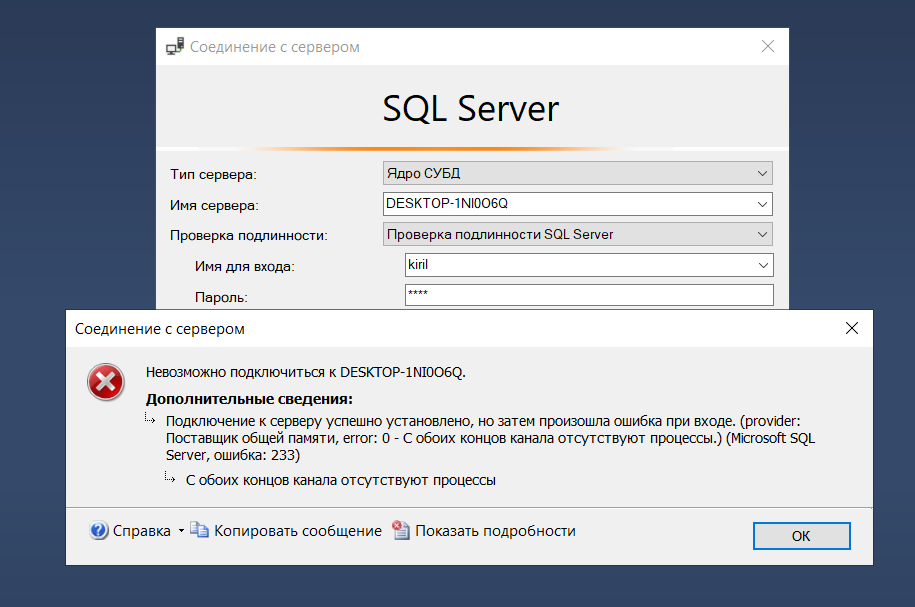


Рисунок 12 – Ошибка при тестировании

Создадим новую роль с помощью следующего запроса: CREATE ROLE libr\_writer;

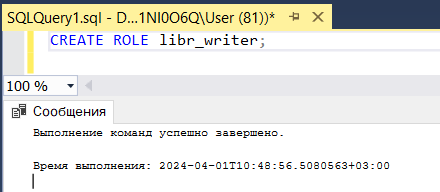


Рисунок 13 – Создание новой роли

Введем 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;

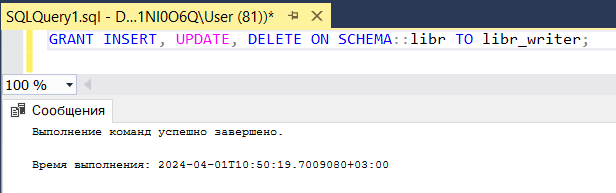


Рисунок 14 – Разрешения роли

Введем 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';

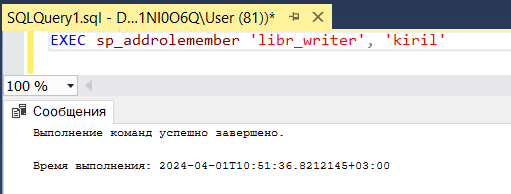


Рисунок 15 – Назначение роли

Назначим разрешения для пользователя “kiril”.

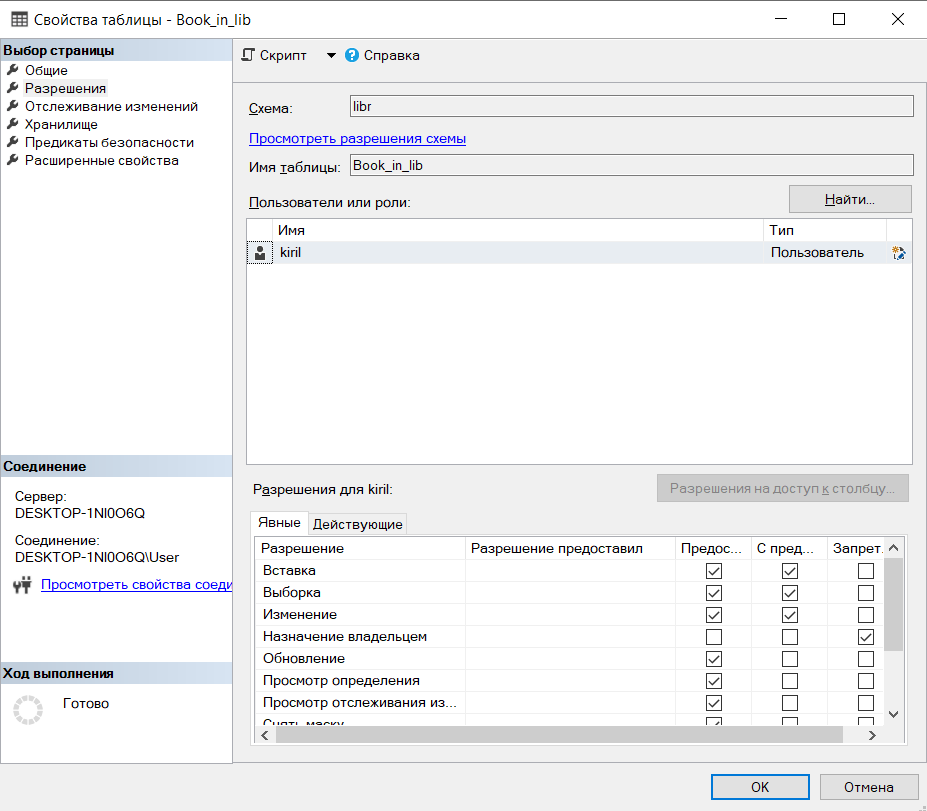


Рисунок 16 – Предоставление прав пользователя

Предоставим пользователю "kiril" право на обновление столбца "Title" в таблице "Book": GRANT UPDATE ON dbo.Book(Title) TO kiril;

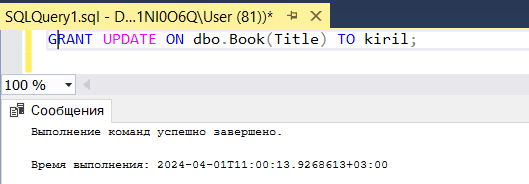


Рисунок 17 – Предоставление прав на обновление столбца

Выберем из таблицы "Book" только те книги, которые были изданы не ранее 2000 года. Создания представления:

CREATE VIEW dbo.NewBooks AS

SELECT \*

FROM dbo.Book

WHERE [Year] >= 2000;

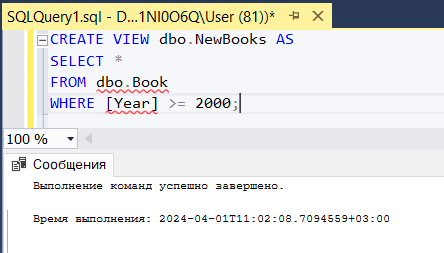


Рисунок 18 – Создание представления

Теперь предоставьте 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;

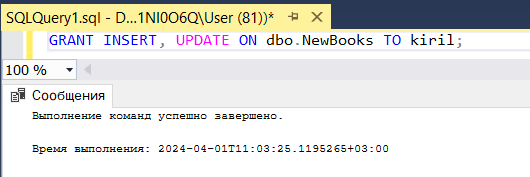


Рисунок 19 – Предоставление прав на изменение представления

**Вывод**

Мы научились создавать пользователя в MSQL и предоставлять ему права.