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. 4233  
Сахабутдинов К. Р.  
  
Принял  
Преподаватель Кожевников К. Д.

Казань 2024

###### **Цель работы**

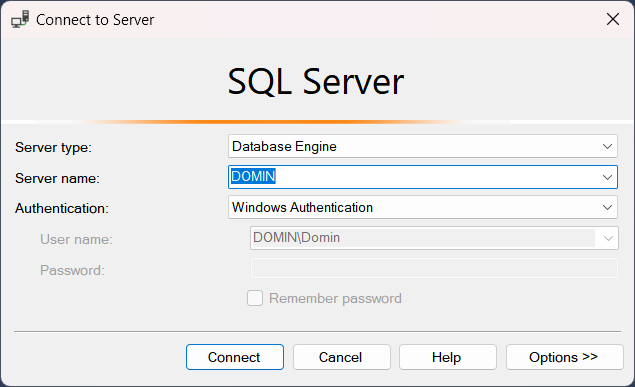
Научиться создавать пользователя в MSSQL и предоставлять ему права.

###### **Задачи**

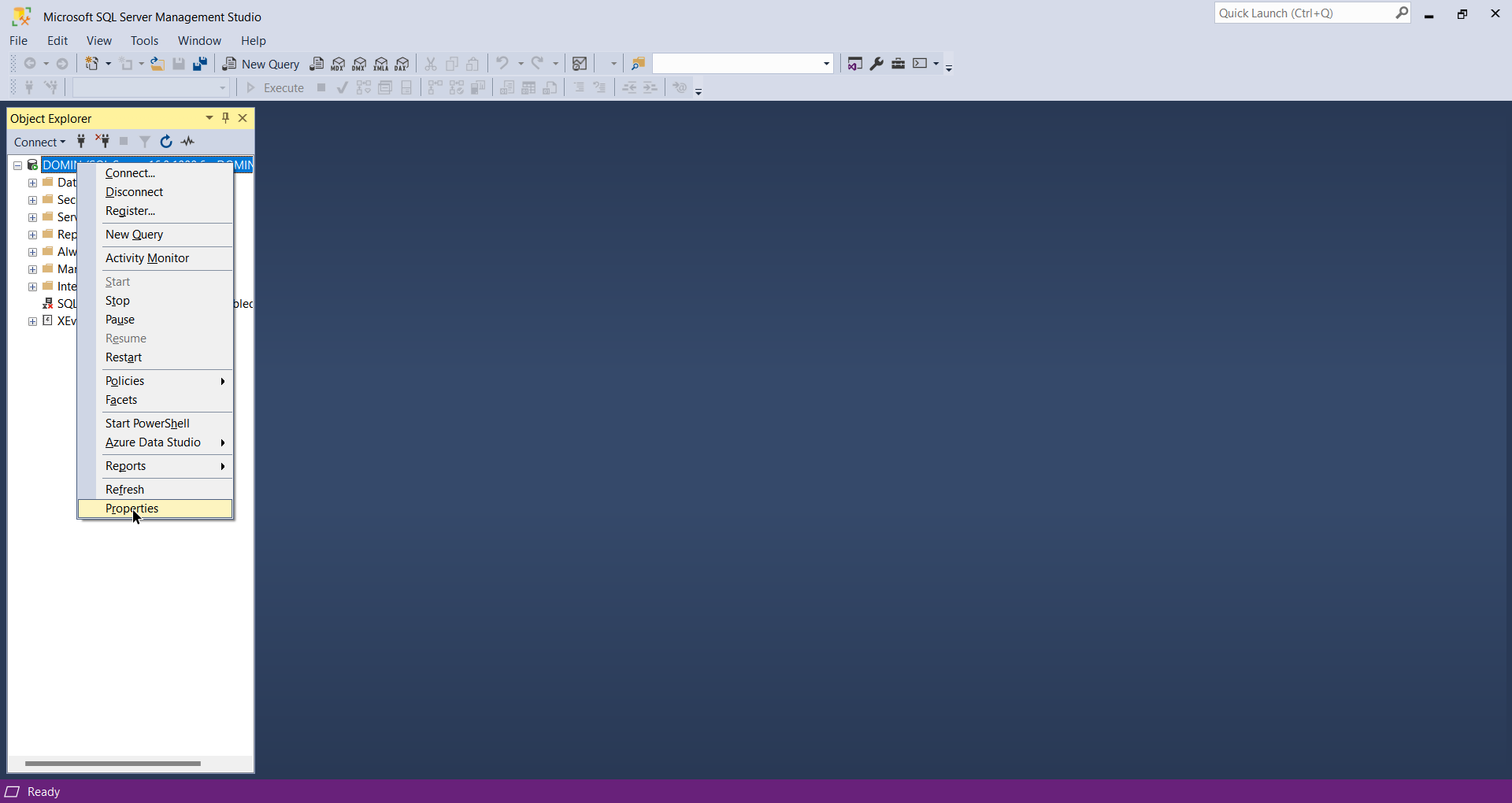
1. Изучить теоретическую часть;
2. Выполнить практические указания;
3. Составить отчет по лабораторной работе.

###### **Ход работы**

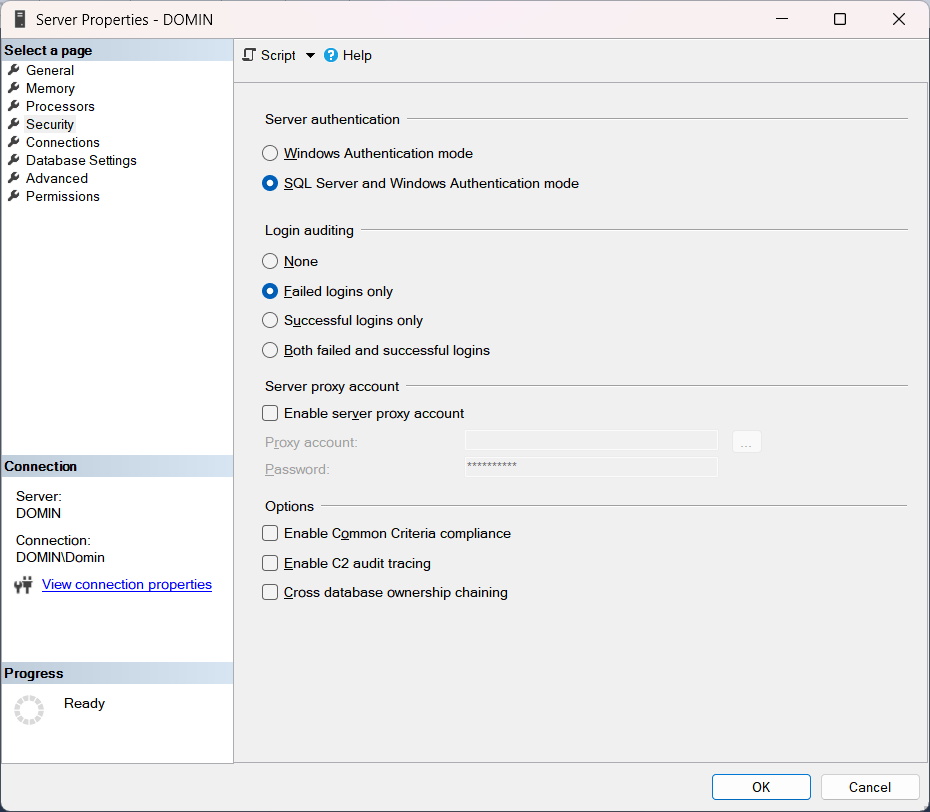
Открываем MSSQL и подключаемся к экземпляру (рисунок 1):

  
Рисунок 1 – Соединение с сервером

Заходим в свойства (рисунок 2):

  
Рисунок 2 – Свойства

Проверяем какие настройки установлены, в графе "Server authentication". Нужно изменить на "SQL Server and Windows Authentication mode" (рисунок 3):

  
Рисунок 3 – Проверка подлинности

Шаг 2

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

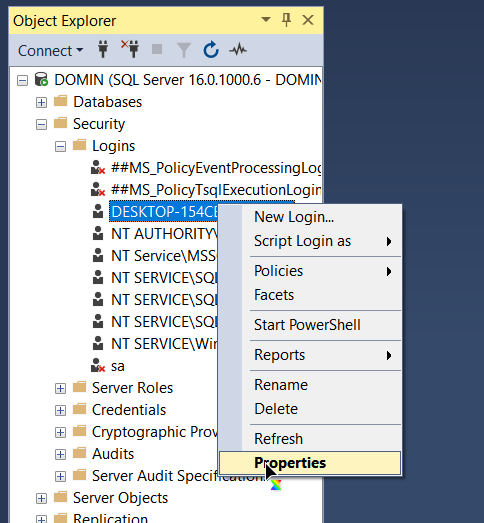
Для этого выполняем следующие действия:

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

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

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

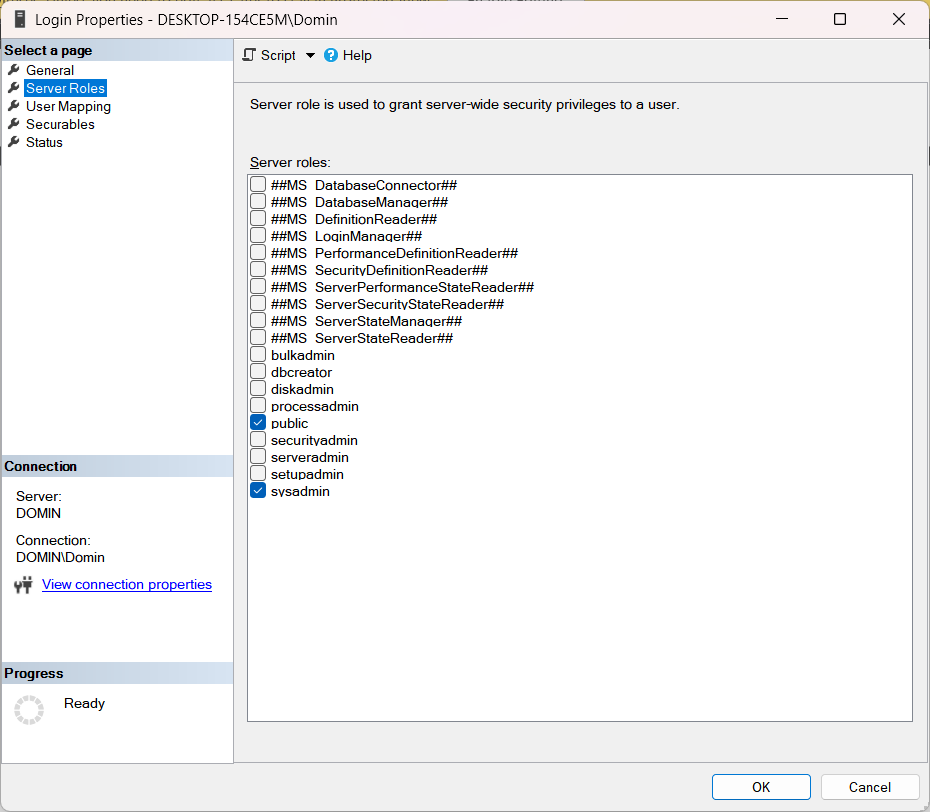
Щёлкаем правой кнопкой мыши на вашей учетной записи и выберите опцию "Properties" (Свойства) (рисунок 4):

  
Рисунок 4 – Свойства

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

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

Здесь мы сможем увидеть список доступных серверных ролей (например, sysadmin, dbcreator, public и т.д.) и проверить, к каким из них относится учетная запись (рисунок 5):

  
Рисунок 5 – Проверка роли сервера

Шаг 3

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

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

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

Необходимо найти базы данных, в которых в учетной записи сопоставлены пользователи.

Для этого выполняем следующие действия:

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

Развёртываем папку "Security" внутри выбранной базы данных.

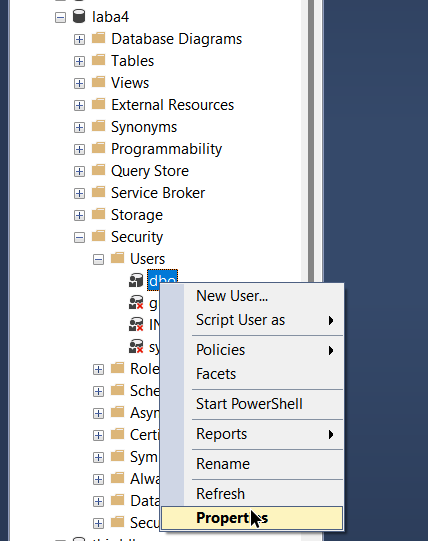
Под папкой "Security" находим папку "Users" (Пользователи). Здесь увидим список пользователей, сопоставленных с данной базой данных.

Находим 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:

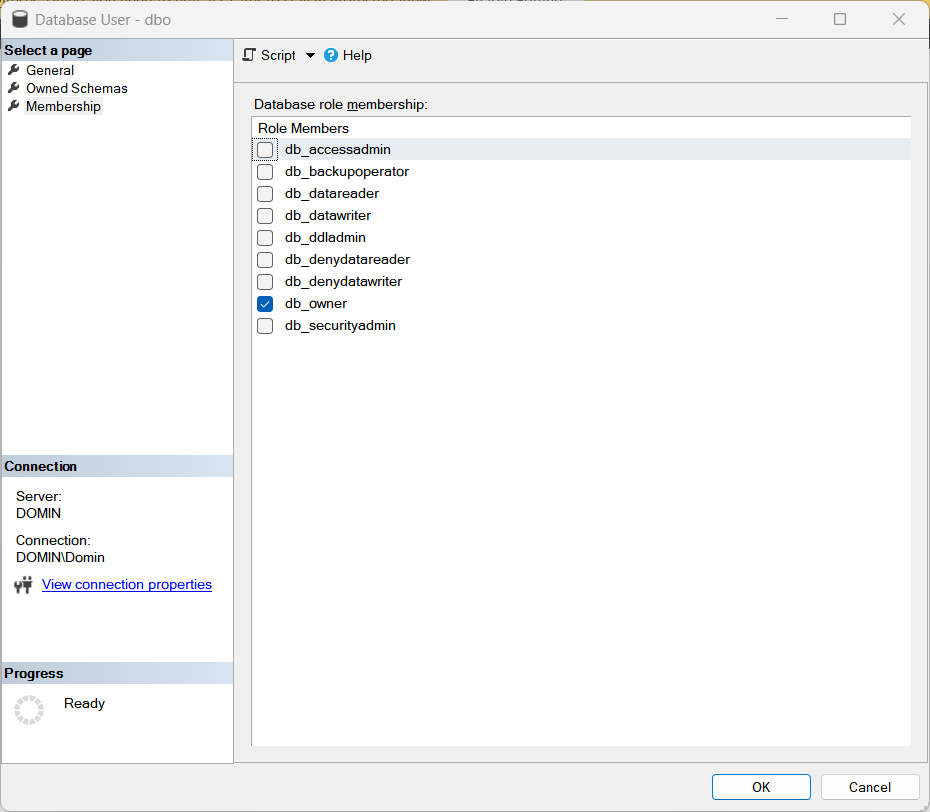
Выбираем пользователя в списке.

Щёлкаем правой кнопкой мыши на пользователе и выбираем опцию "Properties" (Свойства) (рисунок 6):

  
Рисунок 6 – Свойства

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

Ознакомиться 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):

  
Рисунок 7 – Членство

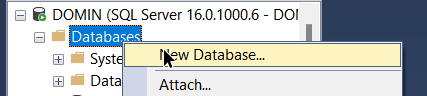
Шаг 4

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

Создание новой базы данных:

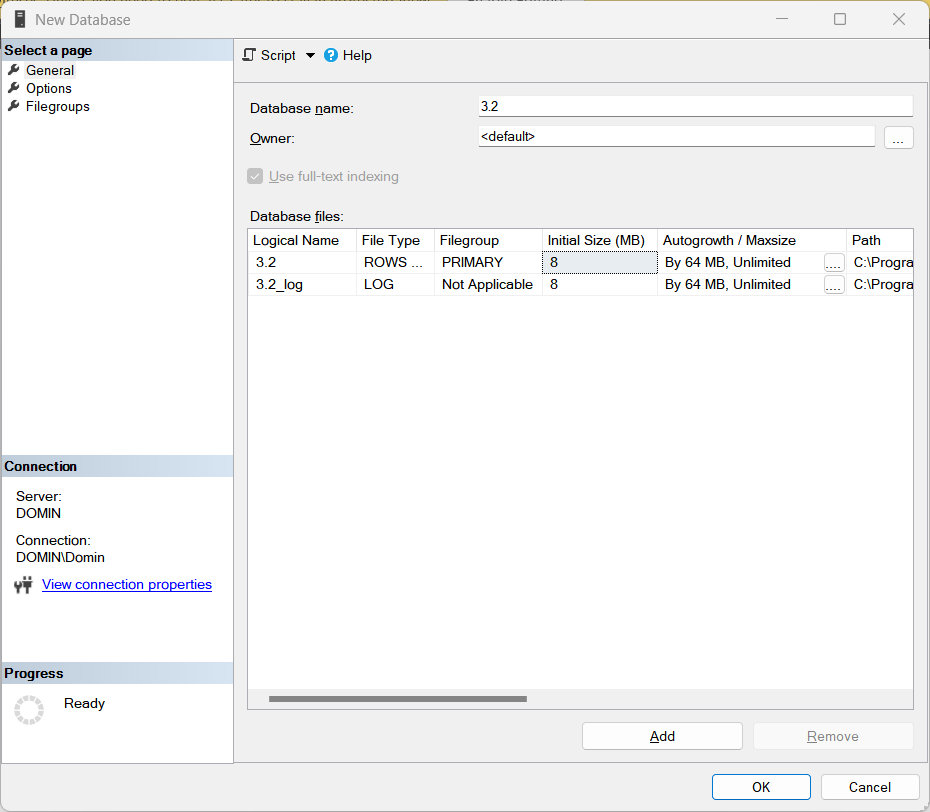
Щёлкаем правой кнопкой мыши на папке "Databases" (Базы данных) в Object Explorer.

Выбираем опцию "New Database" (Новая база данных) (рисунок 8):

  
Рисунок 8 – Создание БД

В появившемся диалоговом окне вводим имя для новой базы.

Нажимаем "OK", чтобы создать новую базу данных (рисунок 9):

  
Рисунок 9 – Имя БД

Проверка списка пользователей и ролей:

Развёртываем созданную нами новую базу данных в списке баз данных.

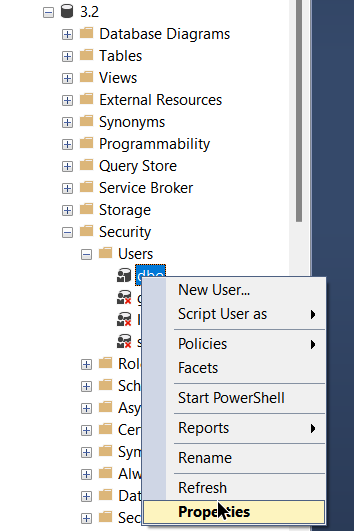
Находим папку "Security" внутри новой базы данных и развёртываем её.

Посмотреть на список пользователей, сопоставленных с новой базой данных.

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

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

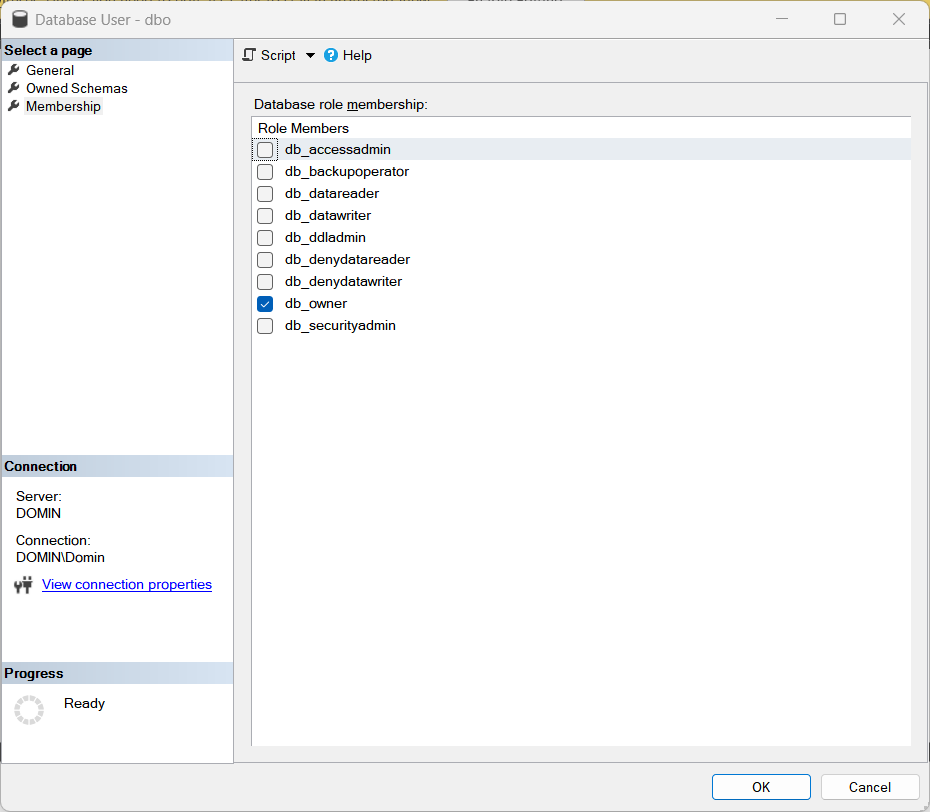
Щёлкаем правой кнопкой мыши на пользователе "dbo" и выбираем опцию "Properties" (Свойства) (рисунок 10):

  
Рисунок 10 – Свойства dbo

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

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

Если роль "db\_owner" отсутствует, то делаем пользователя "dbo" членом роли "db\_owner" (рисунок 11):

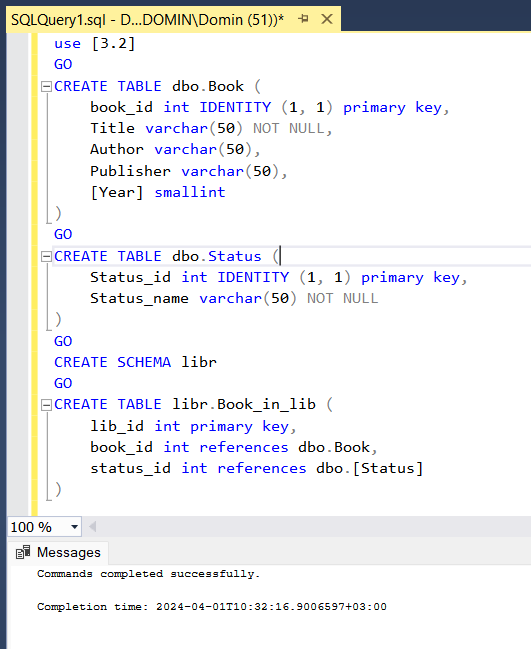
  
Рисунок 11 – Членство

Шаг 5

Используем приведенный ниже скрипт, создаём в базе данных таблицы.

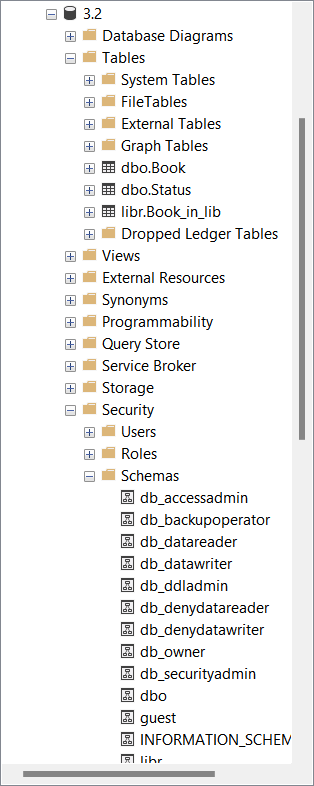
use [3.2]  
GO  
CREATE TABLE dbo.Book (  
 book\_id int IDENTITY (1, 1) primary key,  
 Title varchar(50) NOT NULL,  
 Author varchar(50),  
 Publisher varchar(50),  
 [Year] smallint  
)  
GO  
CREATE TABLE dbo.Status (  
 Status\_id int IDENTITY (1, 1) primary key,  
 Status\_name varchar(50) NOT 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]  
)

Запускаем скрипт (рисунок 12):

  
Рисунок 12 – Создание таблиц

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

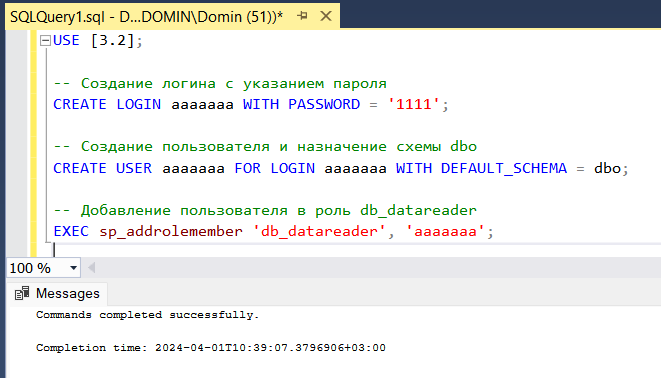
Проверяем, что таблицы созданы в вашей базе данных (рисунок 13):

  
Рисунок 13 – Проверка

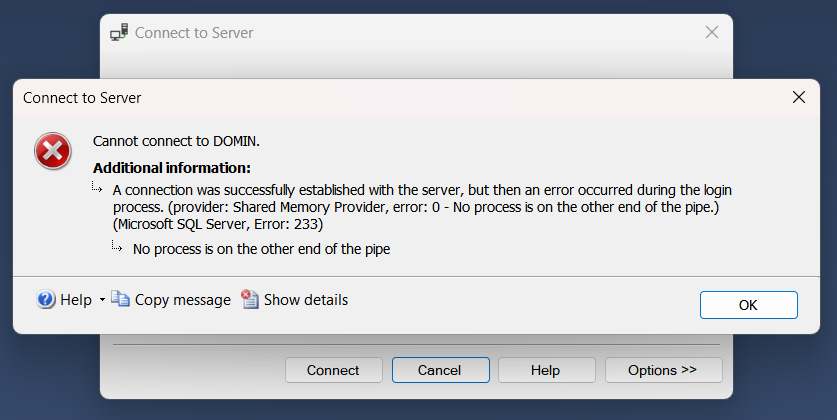
Шаг 6

Создание пользователя, выбор схемы dbo, добавление в роль db\_datareader (рисунок 14):

USE [3.2];  
  
-- Создание логина с указанием пароля  
CREATE LOGIN aaaaaaa WITH PASSWORD = '1111';  
  
-- Создание пользователя и назначение схемы dbo  
CREATE USER aaaaaaa FOR LOGIN aaaaaaa WITH DEFAULT\_SCHEMA = dbo;  
  
-- Добавление пользователя в роль db\_datareader  
EXEC sp\_addrolemember 'db\_datareader', 'aaaaaaa';

  
Рисунок 14 – Скрипт

При аутентификации с помощью логина получаем ошибку (рисунок 15):

  
Рисунок 15 – Ошибка при тестировании

Шаг 7

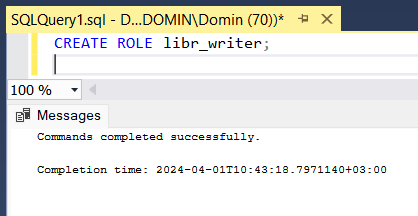
Создание новой роли и назначение разрешений:

Подключаемся к серверу базы данных:

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

Создаём новую роль уровня базы данных (рисунок 16):

CREATE ROLE libr\_writer;

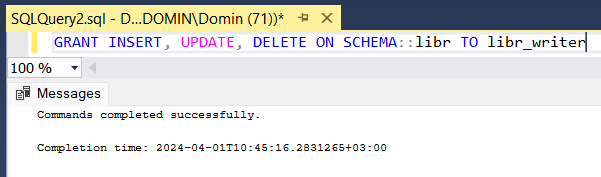
  
Рисунок 16 – Новая роль

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

Назначим 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" (рисунок 17):

GRANT INSERT, UPDATE, DELETE ON SCHEMA::libr TO libr\_writer

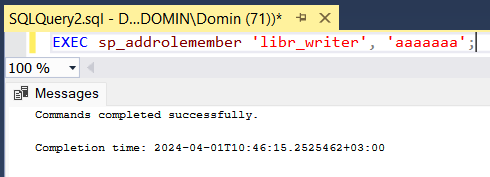
  
Рисунок 17 – Разрешения роли

Добавим пользователя в созданную роль:

Вводим следующий запрос, чтобы добавить пользователя "aaaaaaa" в роль "libr\_writer":

Этот запрос назначает пользователя "aaaaaaa" в роль "libr\_writer", что дает ему возможность изменять данные в объектах схемы "libr" (рисунок 18):

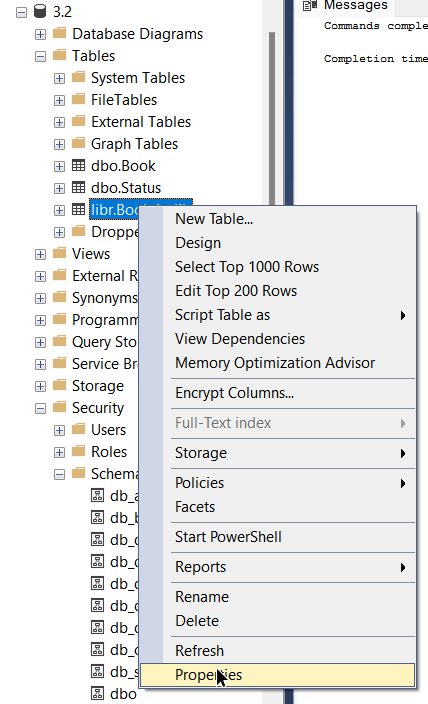
EXEC sp\_addrolemember 'libr\_writer', 'aaaaaaa';

  
Рисунок 18 – Назначение роли

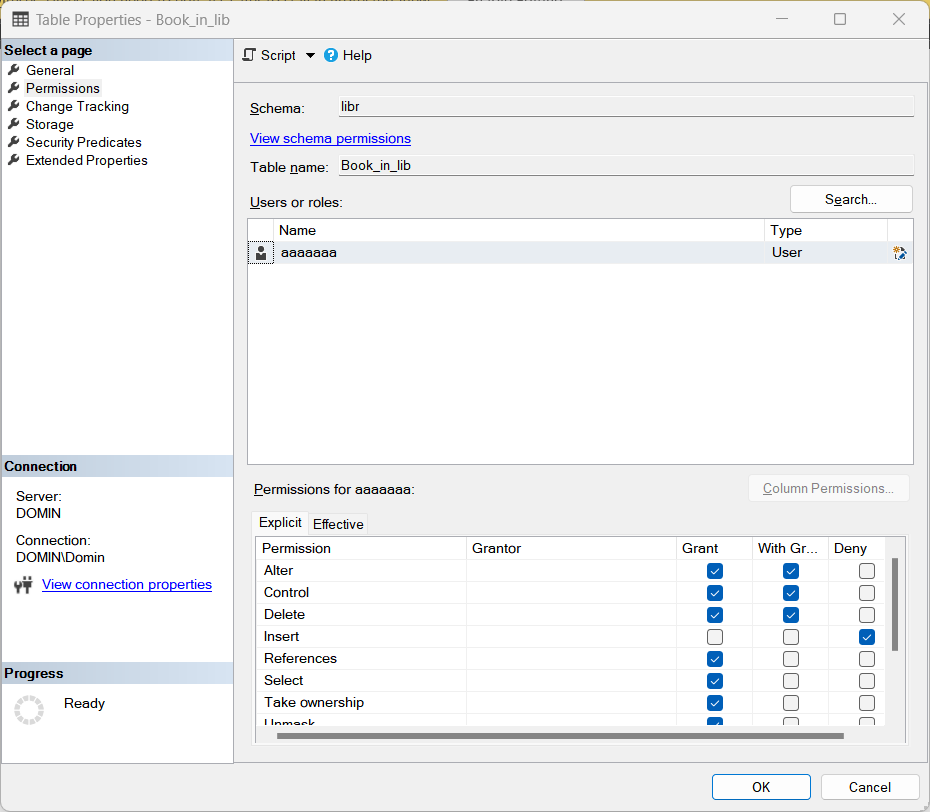
Проверяем доступ пользователя к таблице:

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

Выполняем операции INSERT, UPDATE и DELETE для данных в таблице "Book\_in\_lib", принадлежащей схеме "libr". Заходим в свойства (рисунок 19):

  
Рисунок 19 – Свойства

Назначаем разрешения для пользователя “aaaaaaa” (рисунок 20):

  
Рисунок 20 – 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.

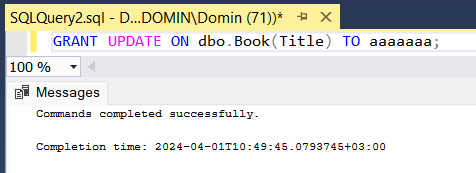
Открываем новый запрос:

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

Выполняем запрос на предоставление прав:

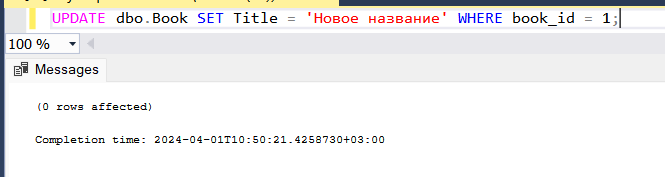
В открытом запросе вводим следующий запрос, чтобы предоставить пользователю "aaaaaaa" право на обновление столбца "Title" в таблице "Book" (рисунок 21):

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

  
Рисунок 21 – Предоставление право

После выполнения запроса попробуем выполнить операцию обновления значения в столбце "Title" таблицы "Book" (рисунок 22):

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

  
Рисунок 22 – Обновление столбцов

Проверяем результат:

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

Шаг 9

Создание представления и предоставление прав на изменение и добавление записей:

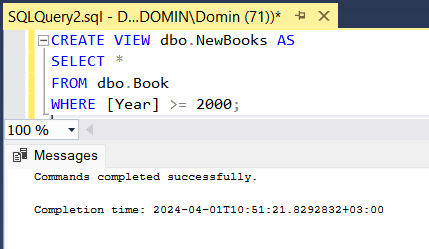
Изучение формата оператора CREATE VIEW:

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

Создание представления:

В открытом запросе выполним оператор CREATE VIEW для создания представления. Выбираем из таблицы "Book" только те книги, которые были изданы не ранее 2000 года. Пример создания представления (рисунок 23):

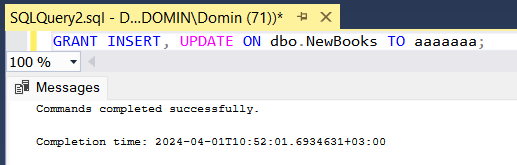
CREATE VIEW dbo.NewBooks AS  
SELECT \*  
FROM dbo.Book  
WHERE [Year] >= 2000;

  
Рисунок 23 – Создания представления

Предоставление прав на изменение и добавление записей:

Теперь предоставим пользователю "aaaaaaa" права на изменение и добавление записей в представление "NewBooks", но только для книг, изданных не ранее 2000 года (рисунок 24):

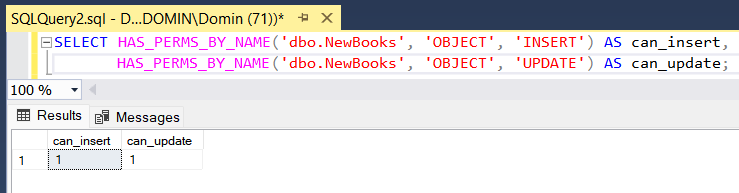
GRANT INSERT, UPDATE ON dbo.NewBooks TO kiril;

  
Рисунок 24 – Права на изменение записей

Проверка прав пользователя:

Убедимся, что пользователь "aaaaaaa" имеет права на изменение и добавление записей в представление "NewBooks". Это можно проверить, выполнив следующий запрос (рисунок 25):

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

###### **Вывод**

Я научился создавать пользователя в MSSQL и предоставлять ему права.