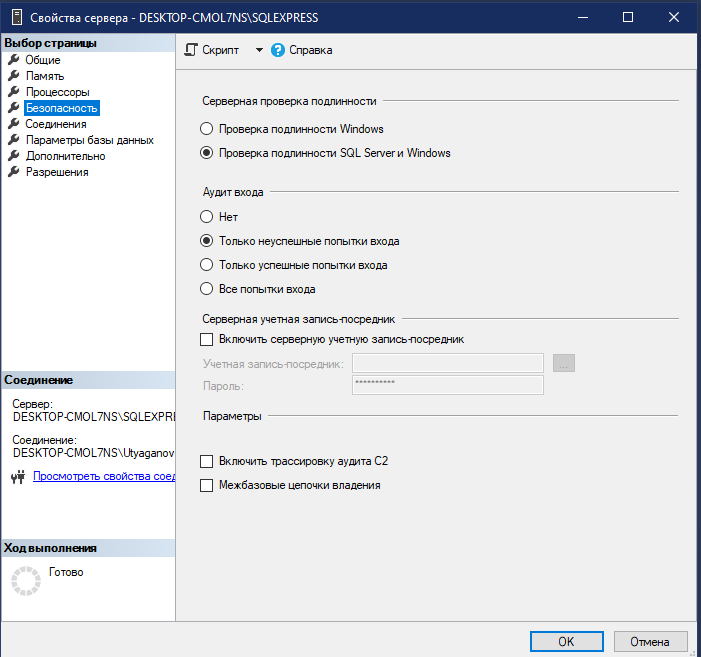
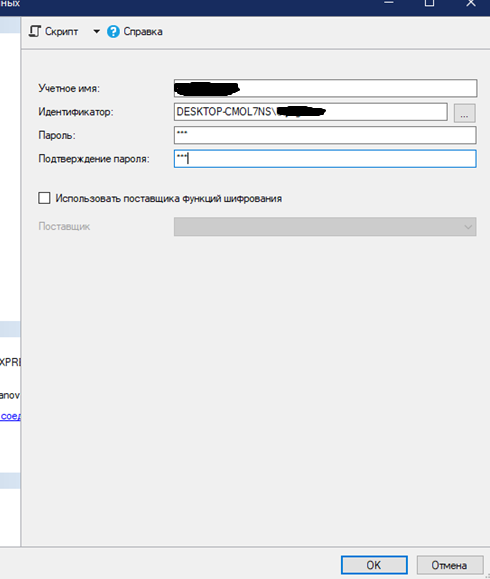
Лабораторная работа 4. Безопасность SQL Server

**Яновский Евгений б1-ИФСТ-21**

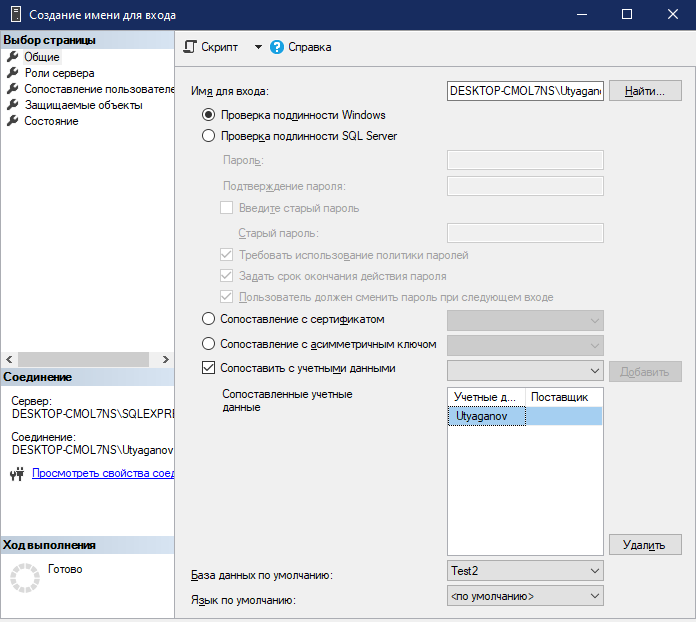
1.Измените, если это возможно, режим аутентификации SQL-сервера на смешанный режим.



2. Создайте локальную учётную запись пользователя Windows на рабочей станции с SQL-сервером.

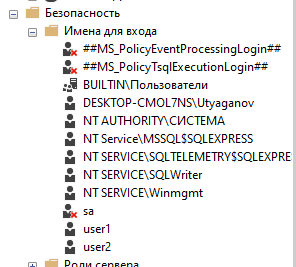
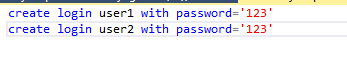


3. Разрешите вход на SQL-сервер под учётными данными созданного пользователя в режиме Windows-аутентификации.

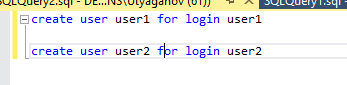


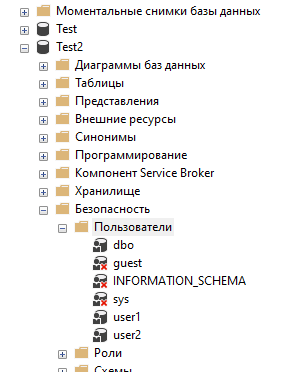


4. Создайте логин для входа на сервер в режиме SQL-аутентификации.

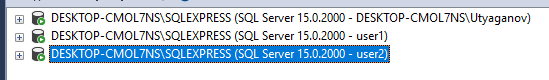


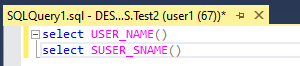
5. В тестовой базе данных создайте двух пользователей, соответствующих созданным ранее логинам.

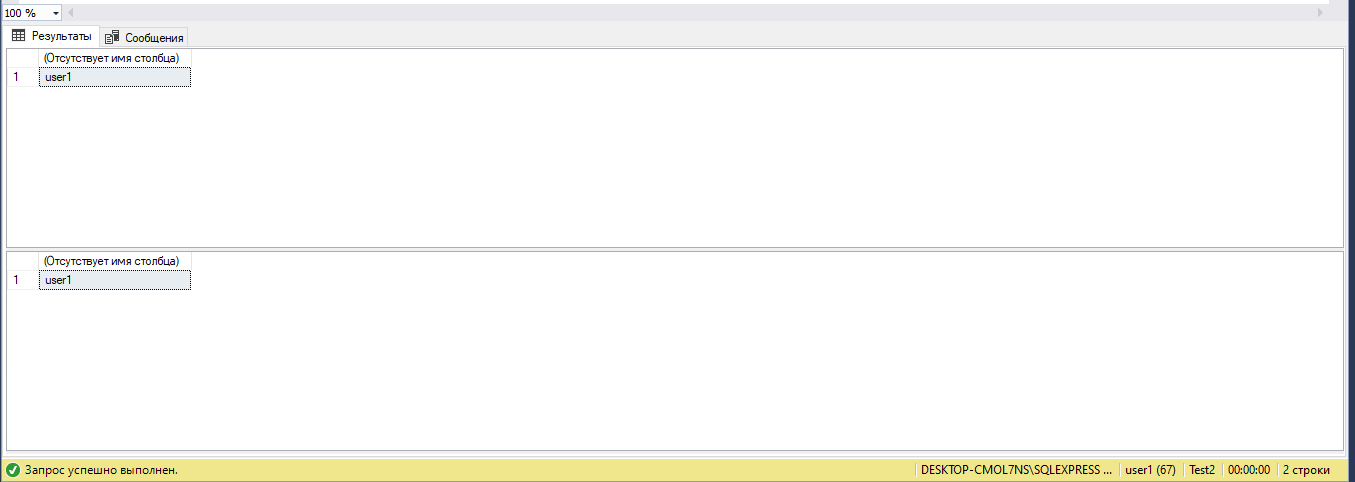




6. Протестируйте подключение к тестовой базе данных с использованием созданных логинов. Выведите при помощи запроса название логина, который использовался при аутентификации на сервере, а также название соответствующего пользователя базы данных.

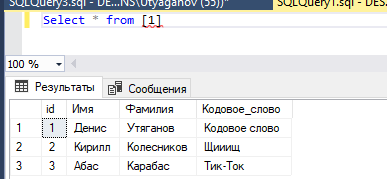




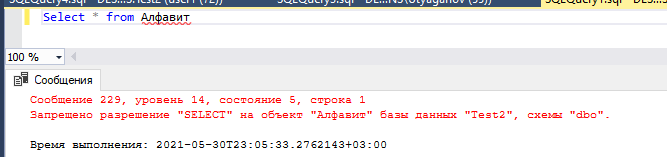


7. Разрешите доступ на чтение любой из таблиц первому пользователю, а второму пользователю – запуск любой созданной Вами хранимой процедуры. Протестируйте выданные разрешения.

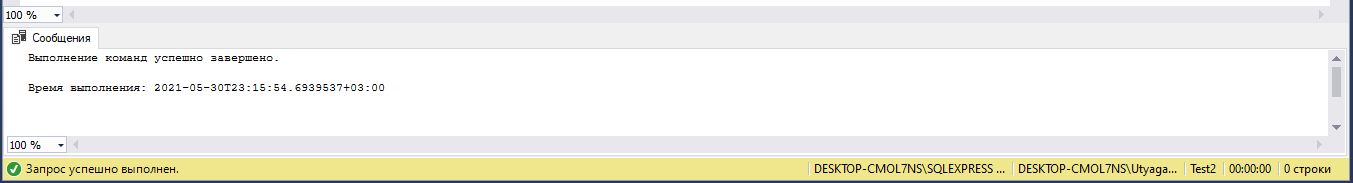


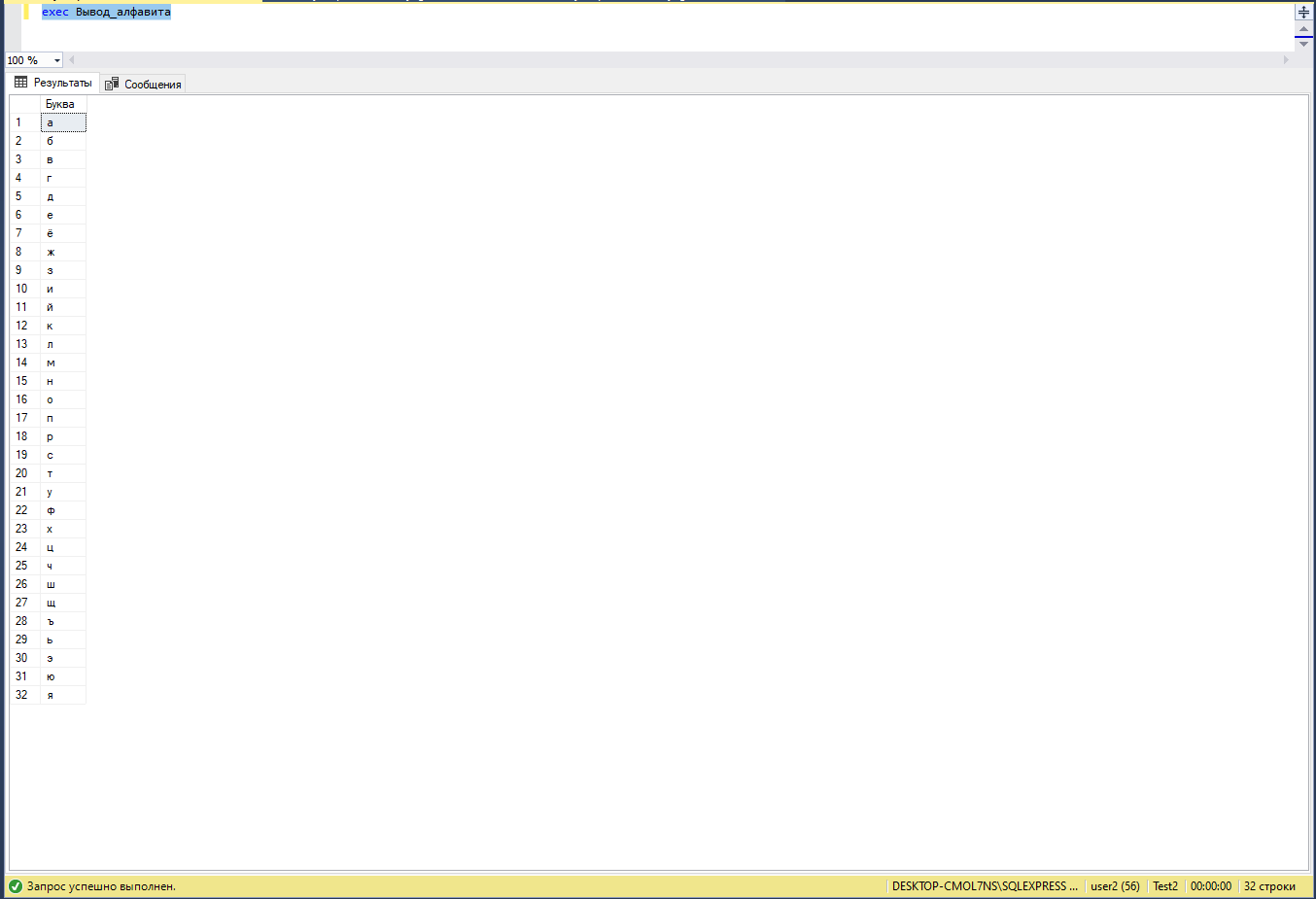




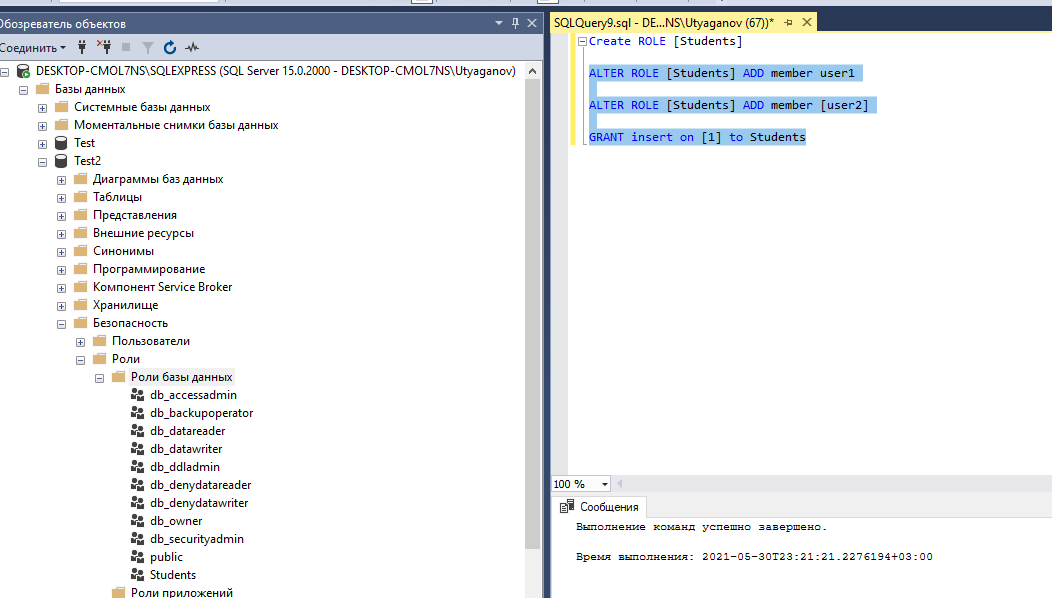








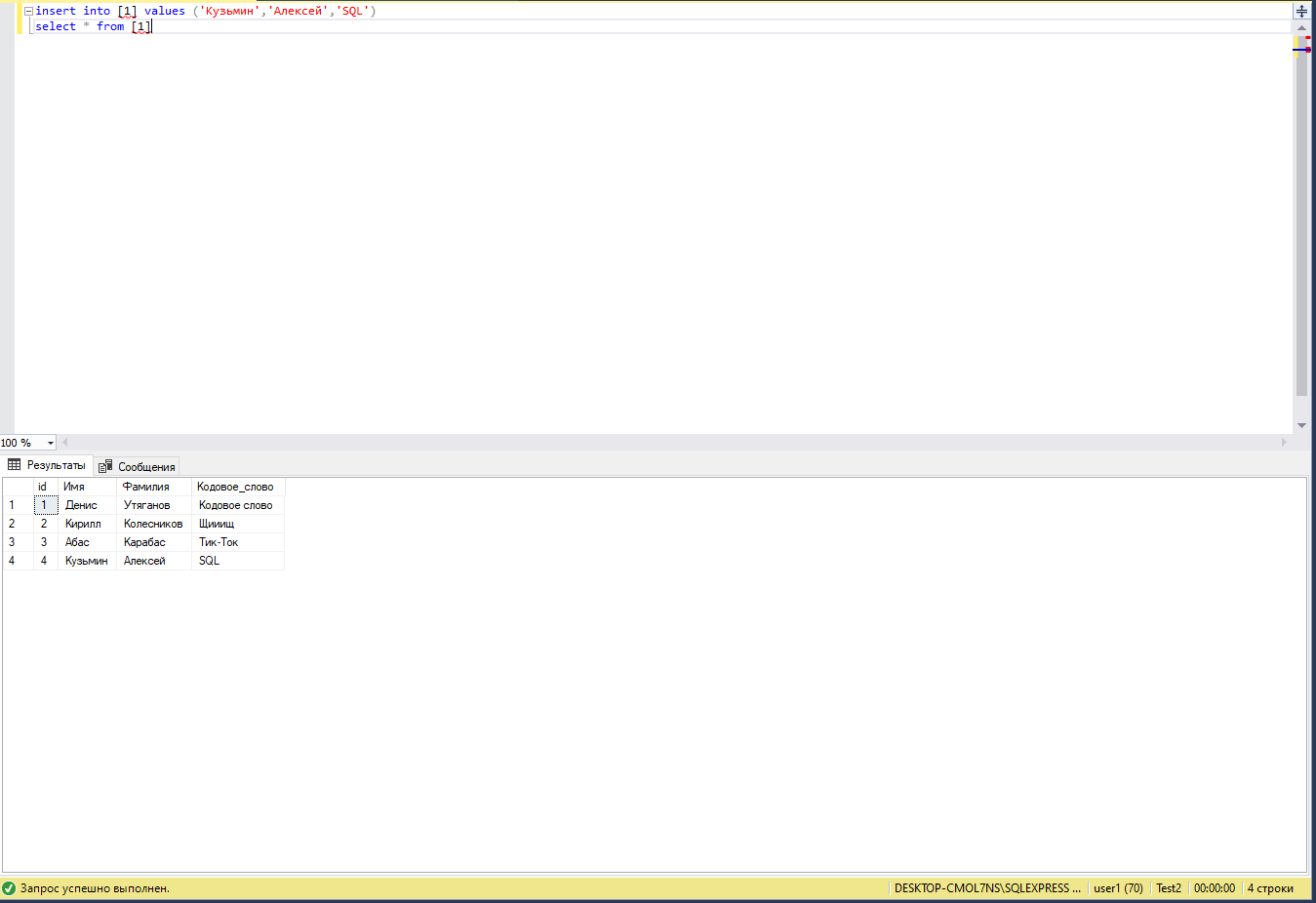
8. Создайте роль «students» и включите в неё двух созданных пользователей. Выдайте разрешение этой роли вставлять записи в ранее использованную таблицу.

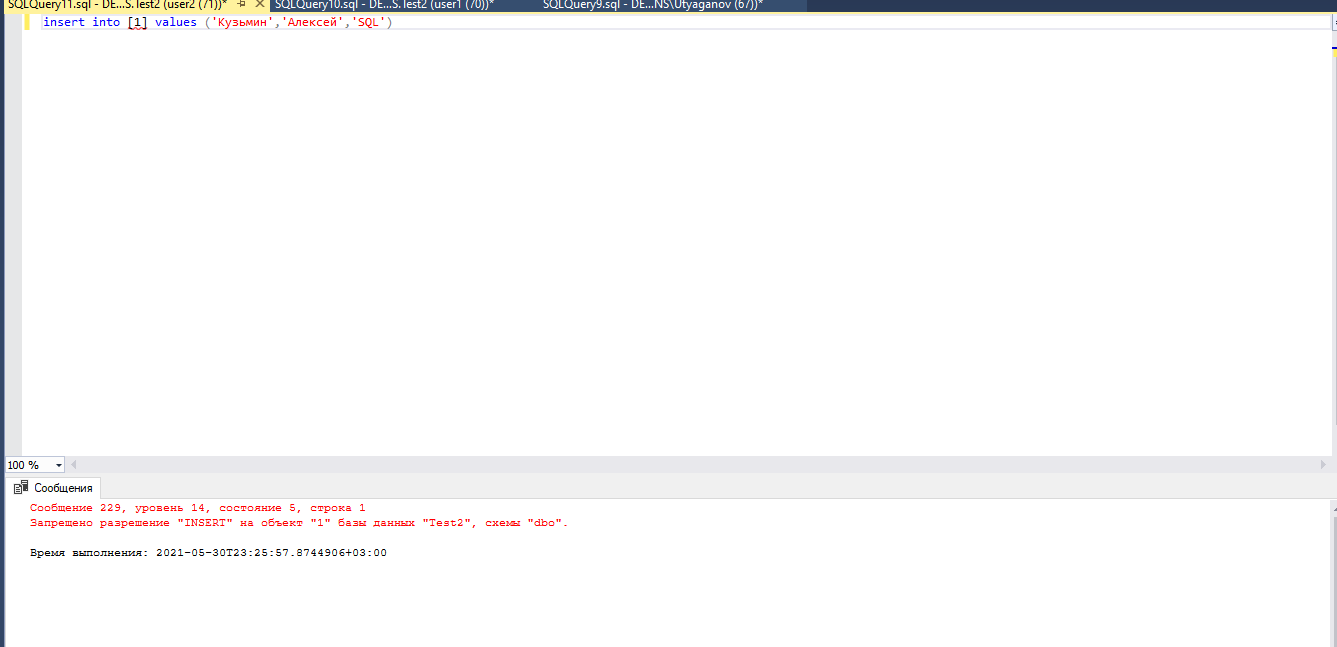


9. Задайте явный запрет вставки записи в ранее использованную таблицу второму пользователю.

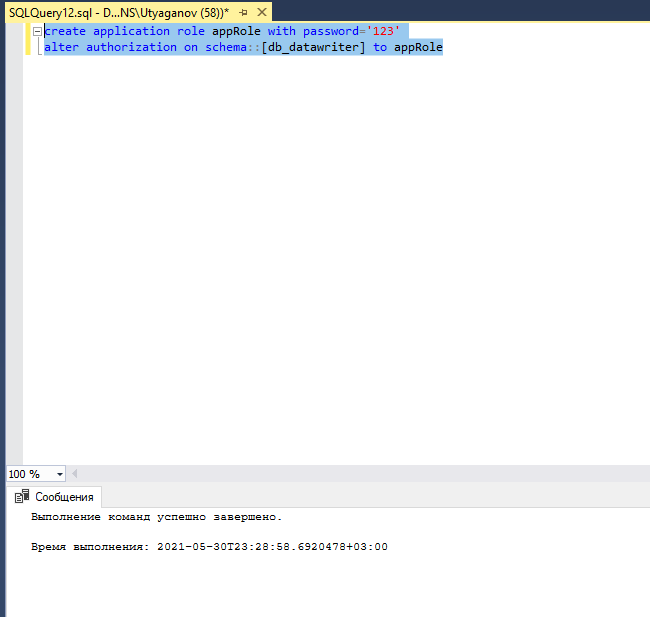


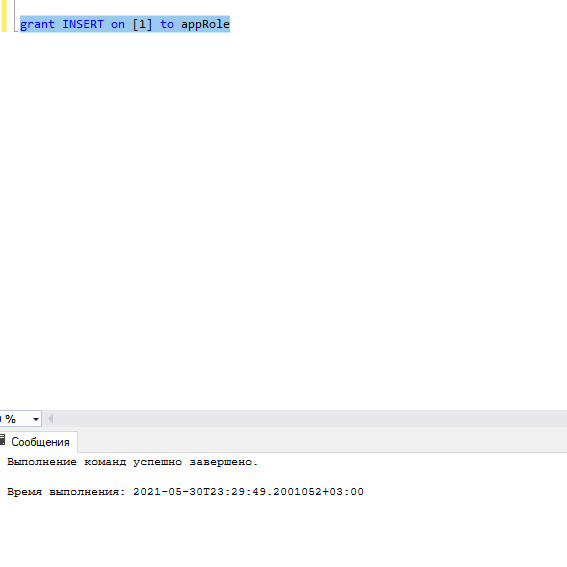
10. Протестируйте все ранее выданные разрешения под учётными данными обоих созданных пользователей.



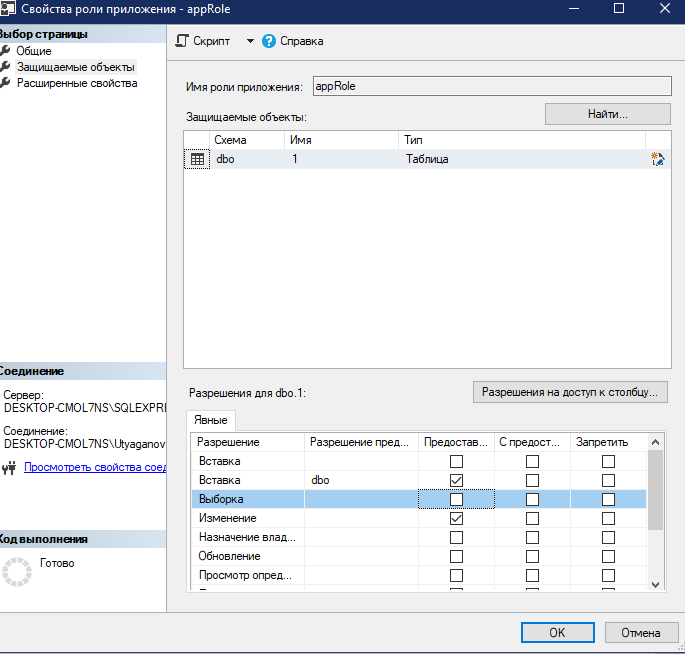
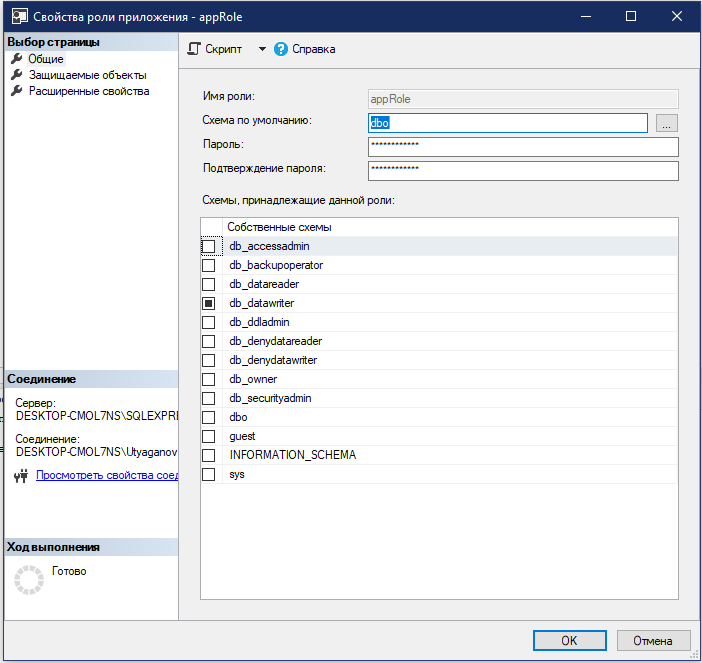


11. Создайте роль приложения с произвольным именем и выдайте этой роли разрешение на удаление и чтение записей в используемой таблице.





12. Продемонстрируйте технологию выполнения вышеперечисленных шагов при помощи графического интерфейса Management Studio.



13. Продемонстрируйте на простейшем примере использование созданной роли приложения. Для этого создайте простой проект с использованием технологии ADO.NET, JDBC, либо любой другой аналогичной. Можно использовать любой язык программирования. Продемонстрируйте факт того, что приложение действует в контексте созданной Вами роли приложения, а также то, что приложению разрешено выполнять операции, разрешённые для созданной Вами роли приложения и не разрешено выполнять другие операции.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Data;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

ConnectionToRole();

//Select();

// Insert();

Delete();

Console.ReadKey();

}

static string connectionString =

@"

Data Source=DESKTOP-CMOL7NS\SQLEXPRESS;

Initial Catalog=Test2;

User ID=user1;

Password=123;

";

public static void ConnectionToRole()

{

SqlCommand comm = new SqlCommand();

comm.CommandText = "Exec sys.sp\_setapprole 'appRole', '123'";

return;

}

public static int Select()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand comm = new SqlCommand("Select Кодовое\_слово From [1] where id = '4'",connection);

connection.Open();

SqlDataReader reader = comm.ExecuteReader();

while (reader.Read())

{

Console.WriteLine(reader["Кодовое\_слово"].ToString());

}

}

return 0;

}

public static int Delete()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand comm = new SqlCommand("Delete from [1] where id = 3",connection);

comm.ExecuteNonQuery();

Console.WriteLine("Successfully deleted");

}

return 0;

}

public static int Insert()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("Insert into [1] values(@Имя, @Фамилия,@Кодовое\_слово)");

command.Connection = connection;

command.Parameters.Add(new SqlParameter("@Имя", "Илья"));

command.Parameters.Add(new SqlParameter("@Фамилия", "Кривенко"));

command.Parameters.Add(new SqlParameter("@Кодовое\_слово", "Староста"));

command.ExecuteNonQuery();

Console.WriteLine("Successfully added");

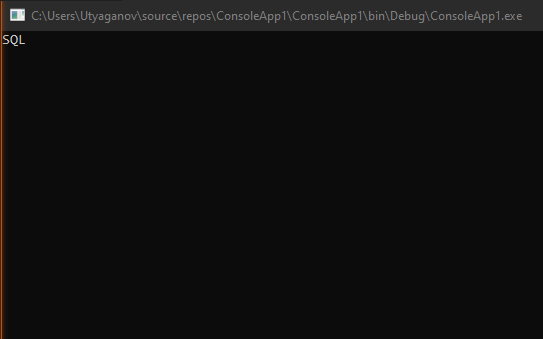
}

return 0;

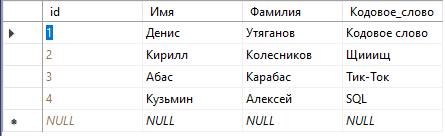
}

}

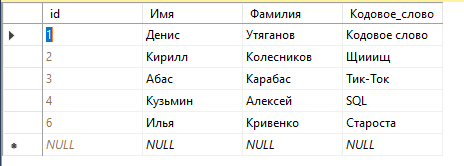
}



До всех изменений



После Insert



После DELETE:

