

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебный предмет
«Конструирование программ и языки программирования»

Инструкция
по выполнению лабораторной работы №31
«Исследование возможностей технологий доступа к данным. Организация
доступа к данным»

Минск 2024 г.

Лабораторная работа № 31

Тема работы: «Исследование возможностей технологий доступа к данным. Организация доступа к данным»

1 Цель работы

Сформировать умения и навыки исследование возможностей технологий доступа к данным. Организация доступа к данным

2 Задание

Реализовать базу данных из основных теоретических сведений посредством встроенной в Visual Studio СУБД MS SQLServer. Подключить БД к проекту.

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

К основным технологиям доступа к данным в C# и .NET относят следующее:

1. ADO.NET (ActiveX Data Objects .NET):

- ADO.NET является базовой технологией доступа к данным в .NET Framework. Он предоставляет набор библиотек и классов для работы с различными источниками данных, такими как базы данных, XML-документы и другие.

- Преимущества: Простота использования, высокая производительность, поддержка различных источников данных.

- Недостатки: Требуется ручное написание SQL-запросов, менее удобен для объектно-ориентированного программирования.

2. Entity Framework:

- Entity Framework (EF) - это ORM (Object-Relational Mapping) фреймворк, который позволяет работать с данными в терминах объектов C# вместо традиционных SQL-запросов.

- Преимущества: Упрощение взаимодействия с базами данных, поддержка LINQ (Language-Integrated Query), автоматическое создание SQL-запросов.

- Недостатки: В некоторых случаях может вызывать проблемы с производительностью, требует изучения.

3. Dapper:

- Dapper - это легковесный и быстрый микро-ORM для .NET. Он предоставляет простой способ выполнения запросов и маппинга результатов на объекты.

- Преимущества: Высокая производительность, легкость использования, поддержка хранимых процедур.

- Недостатки: Меньше функциональности по сравнению с Entity Framework.

4. LINQ to SQL:

- LINQ to SQL предоставляет ORM функциональности для работы с базами данных, используя LINQ-запросы.

- Преимущества: Интегрирован с LINQ, прост в использовании для простых сценариев доступа к данным.

- Недостатки: Больше не развивается, рекомендуется использовать Entity Framework.

5. WCF Data Services:

- WCF Data Services (ранее ADO.NET Data Services) позволяет создавать службы данных, которые могут быть использованы для запросов и обновлений данных через HTTP.

- Преимущества: Стандартизированный доступ к данным через веб-службы, поддержка OData протокола.

- Недостатки: Устаревший, рекомендуется использовать ASP.NET Web API.

6. ASP.NET Core Data Access:

- ASP.NET Core предоставляет новый подход к работе с данными, используя Entity Framework Core и другие технологии.

- Преимущества: Переносимость и поддержка множества баз данных, включая SQL Server, PostgreSQL, MySQL и др.

- Недостатки: Могут возникнуть проблемы с производительностью в некоторых случаях.

База данных (БД) – упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.

База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Существует большое количество вариантов подключения БД к проекту. Все зависит от того, какая конкретно СУБД используется.

В Microsoft Visual Studio существует встроенная СУБД – Microsoft SQL Server, которая позволяет не устанавливать дополнительное ПО для СУБД и разрабатывать базы данных прямо в Visual Studio.

Для того, чтобы создать БД в Microsoft Visual Studio, в первую очередь, необходимо создать новый проект WinForms (рисунок 1).

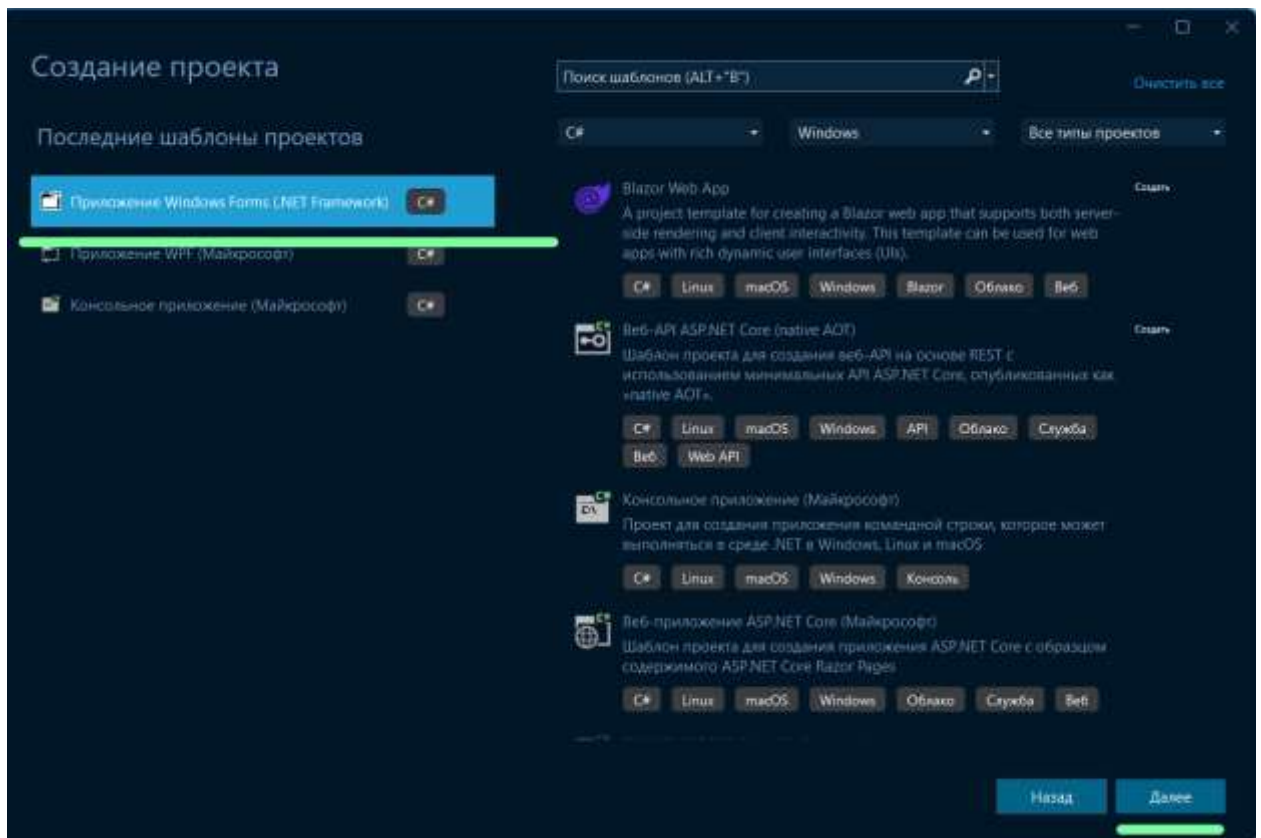


Рисунок 1 – Выбор типа проекта

Далее необходимо назвать проект, выбрать платформу и нажать на кнопку Создать (рисунок 2).

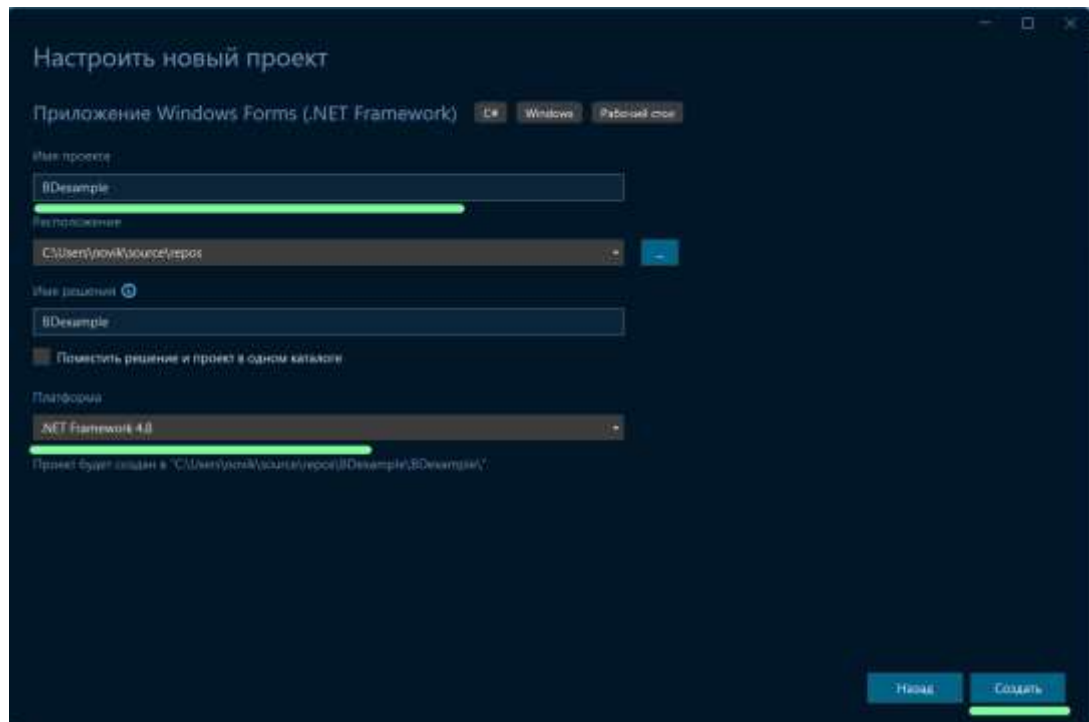


Рисунок 2 – Настройка проекта

После создания проекта Вы увидите окно, представленное на рисунке 3.

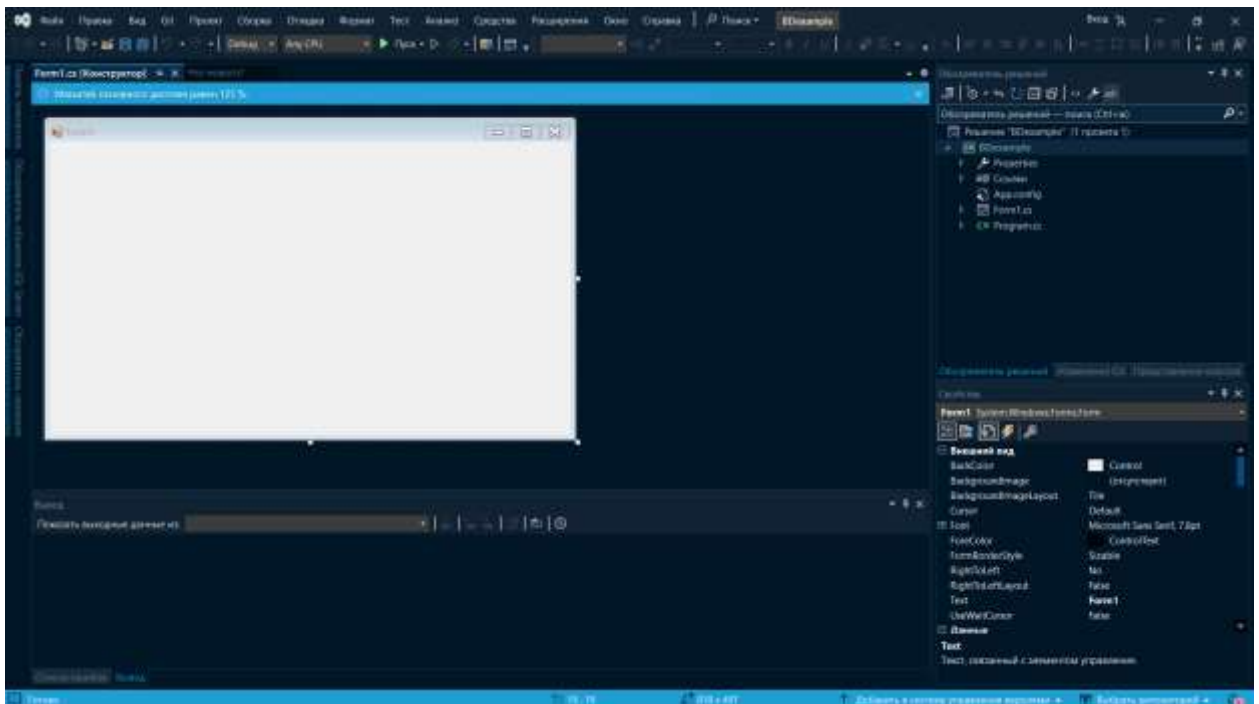


Рисунок 3 – Окно для работы с проектом

Данное окно содержит в себе конструктор формы, панель элементов для формы, обозреватель решений и панель свойств. Также, рядом с панелью элементов могут располагаться дополнительные панели – Обозреватель объектов SQL Server и Обозреватель серверов.

Для создания БД перейдем во вкладку Обозреватель объектов SQL Server (рисунок 4). Данная вкладка содержит информацию о существующих серверах, базах данных и т.д.

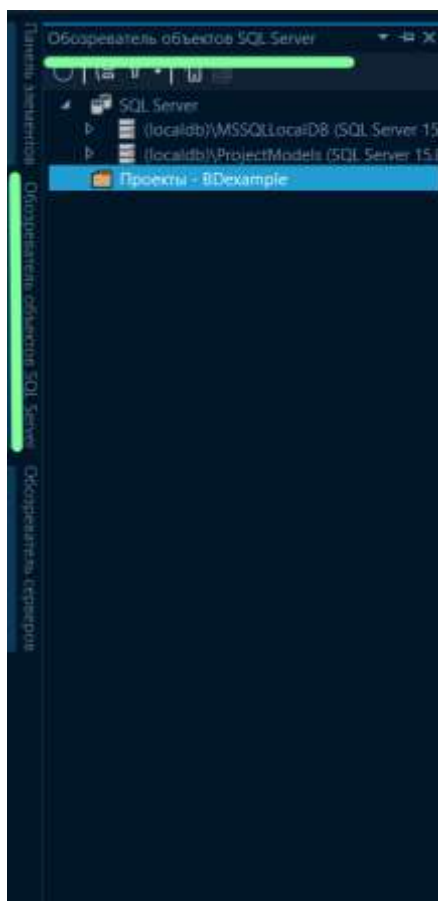


Рисунок 4 – Содержимое вкладки Обозреватель объектов SQL Server

Процесс создания БД на «чистом» проекте состоит в создании изолированного экземпляра SQL Server. Для его добавления необходимо нажать на кнопку Добавить SQL Server (рисунок 5).

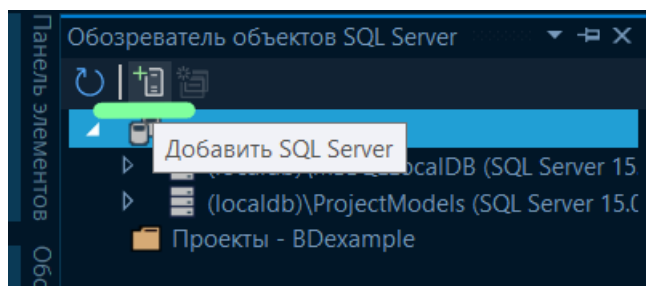


Рисунок 5 – Добавление SQL Server

Далее открывается окно с выбором и настройкой сервера (рисунок 6). Для выполнения лабораторной работы достаточно выбрать из двух вариантов локальных серверов и нажать кнопку Подключиться.

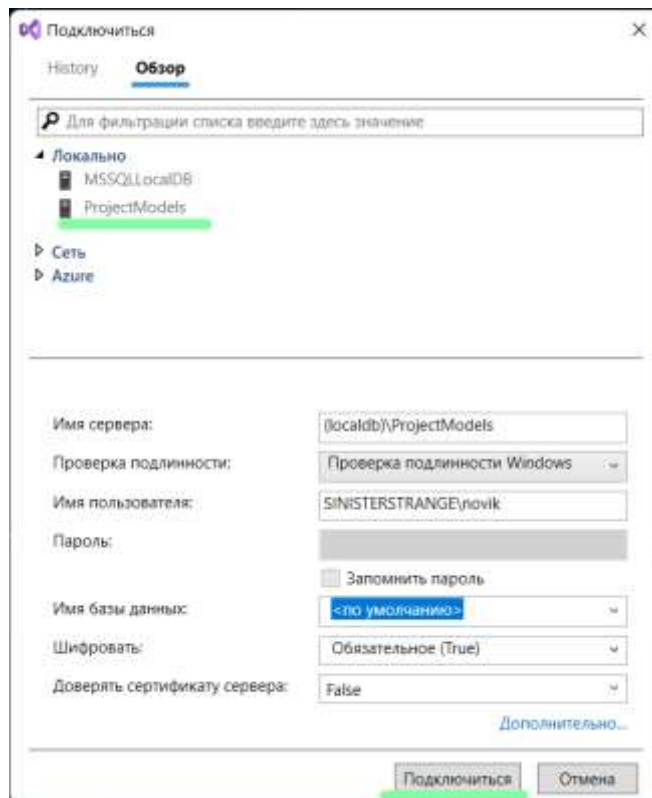


Рисунок 6 – Подключение к SQL Server

После того, как произошло подключение к серверу, его можно будет увидеть в списке объектов SQL Server. В подключении также будут указаны папки. Для дальнейшей работы нам необходима папка Базы данных. Для добавления новой БД на сервер, необходимо щелкнуть правой кнопкой мыши по папке и выбрать Добавить новую базу данных (рисунок 7).

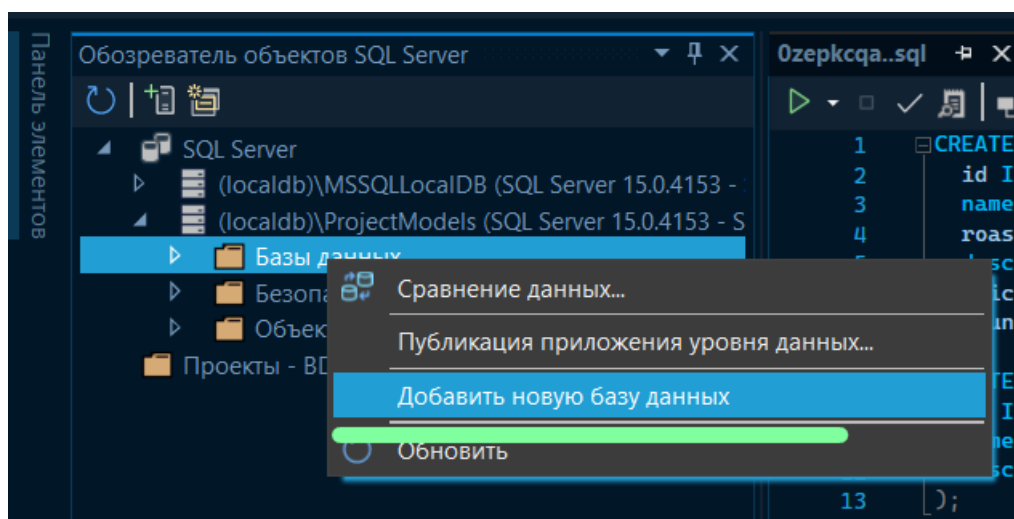


Рисунок 7 – Добавление новой базы данных

Далее, необходимо назвать новую БД в соответствующем окне (рисунок 8). Местоположение БД лучше не менять. После этого следует нажать кнопку ОК.

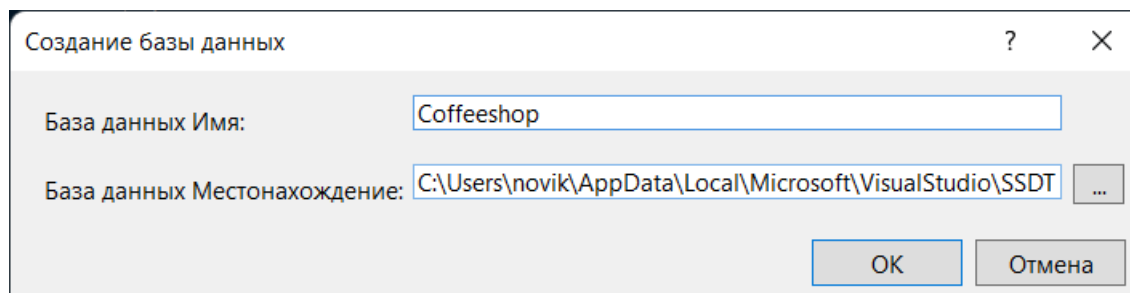


Рисунок 8 – Создание базы данных

Как только БД была создана, она добавится в дерево объектов для Вашего сервера (рисунок 9).

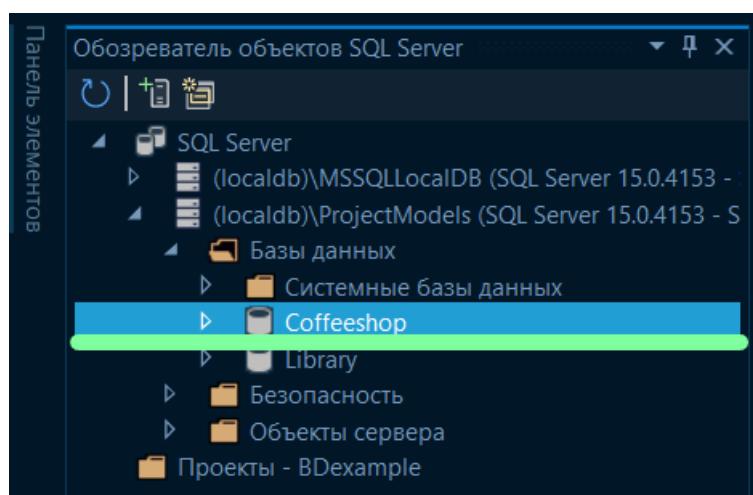


Рисунок 9 – Размещение ранее созданной БД на сервере

После этого необходимо добавить содержимое БД – таблицы, связи между ними и данные.

Для этого можно воспользоваться конструктором таблиц. Однако, это достаточно долго.

Для дальнейшей работы воспользуемся базой данных по тематике «Кофейня», состоящей из четырех связанных таблиц.

Для того, чтобы создать таблицы и заполнить их данными, воспользуемся SQL-запросами, код которых представлен ниже.

```
CREATE TABLE Coffee_Beans (  
  id INT IDENTITY PRIMARY KEY,  
  name VARCHAR(50) NOT NULL,  
  roast_level VARCHAR(20) NOT NULL,  
  description VARCHAR(200),  
  price DECIMAL(10,2) NOT NULL,
```

```

country_of_origin VARCHAR(50)
);
CREATE TABLE Brewing_Methods (
id INT IDENTITY PRIMARY KEY,
name VARCHAR(50) NOT NULL,
description VARCHAR(200)
);
CREATE TABLE Drinks (
id INT IDENTITY PRIMARY KEY,
name VARCHAR(50) NOT NULL,
description VARCHAR(200),
id_coffee_beans INT,
id_brewing_method INT,
price DECIMAL(10,2) NOT NULL,
volume INT NOT NULL,
FOREIGN KEY (id_coffee_beans) REFERENCES Coffee_Beans(id),
FOREIGN KEY (id_brewing_method) REFERENCES Brewing_Methods(id)
);
CREATE TABLE Orders (
id INT IDENTITY PRIMARY KEY,
date_time DATETIME NOT NULL,
id_drink INT,
quantity INT NOT NULL,
total DECIMAL(10,2) NOT NULL,
FOREIGN KEY (id_drink) REFERENCES Drinks(id)
);
INSERT INTO Coffee_Beans (name, roast_level, description, price,
country_of_origin)
VALUES
('Arabica Cattura', 'Medium', 'Balanced taste with fruity notes', 350,
'Brazil'),
('Robusta Vietnam', 'Dark', 'Strong and bitter coffee with chocolate
aroma', 280, 'Vietnam'),
('Ethiopia Yirgacheffe', 'Light', 'Floral aroma with citrus notes', 420,
'Ethiopia'),
('Guatemala Antigua', 'Medium', 'Chocolate taste with nutty aroma', 380,
'Guatemala');
INSERT INTO Brewing_Methods (name, description)
VALUES
('Espresso', 'Classic way of making coffee'),
('Americano', 'Espresso diluted with water'),
('Cappuccino', 'Espresso with milk foam'),
('Latte', 'Espresso with hot milk');
INSERT INTO Drinks (name, description, id_coffee_beans, id_brewing_method,
price, volume)
VALUES
('Espresso', 'Classic coffee', 1, 1, 150, 30),
('Americano', 'Espresso with water', 1, 2, 180, 120),
('Cappuccino', 'Coffee with milk foam', 2, 3, 220, 180),
('Latte', 'Coffee with milk', 3, 4, 200, 250);
INSERT INTO Orders (date_time, id_drink, quantity, total)
VALUES
('2024-02-14 12:00:00', 1, 2, 300),
('2024-02-14 14:30:00', 3, 1, 220),
('2024-02-14 17:00:00', 2, 1, 180);

```

Данный код необходимо вставить в редактор SQL-запросов и выполнить (рисунок 10-11).

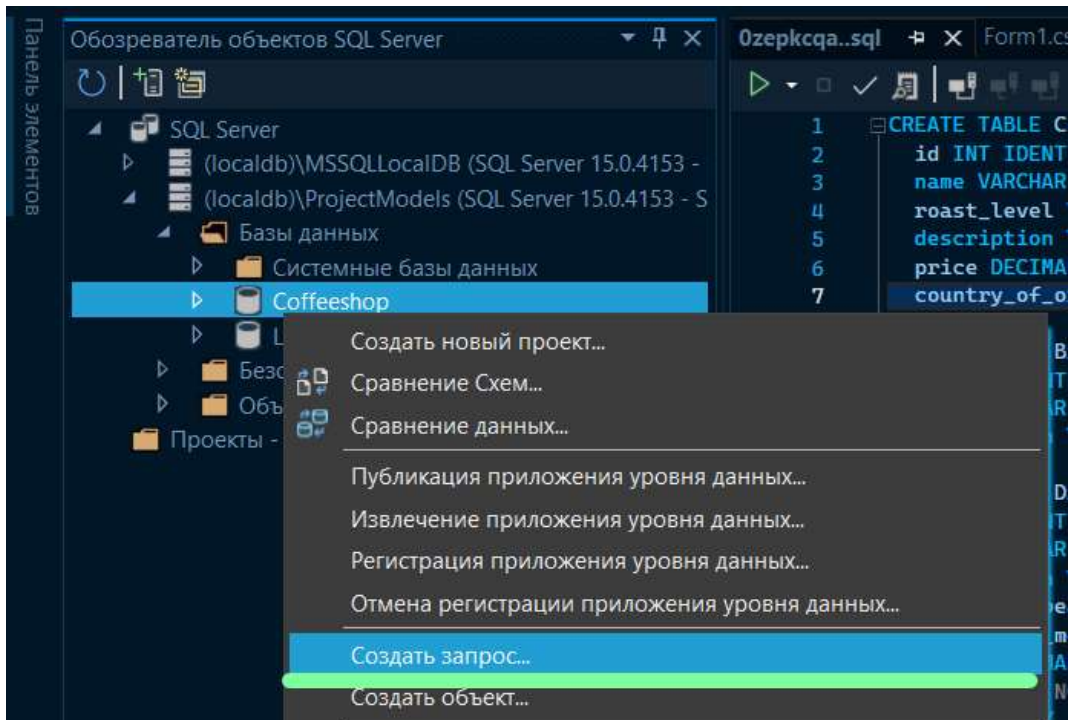


Рисунок 10 – Открытие окна для добавления запроса

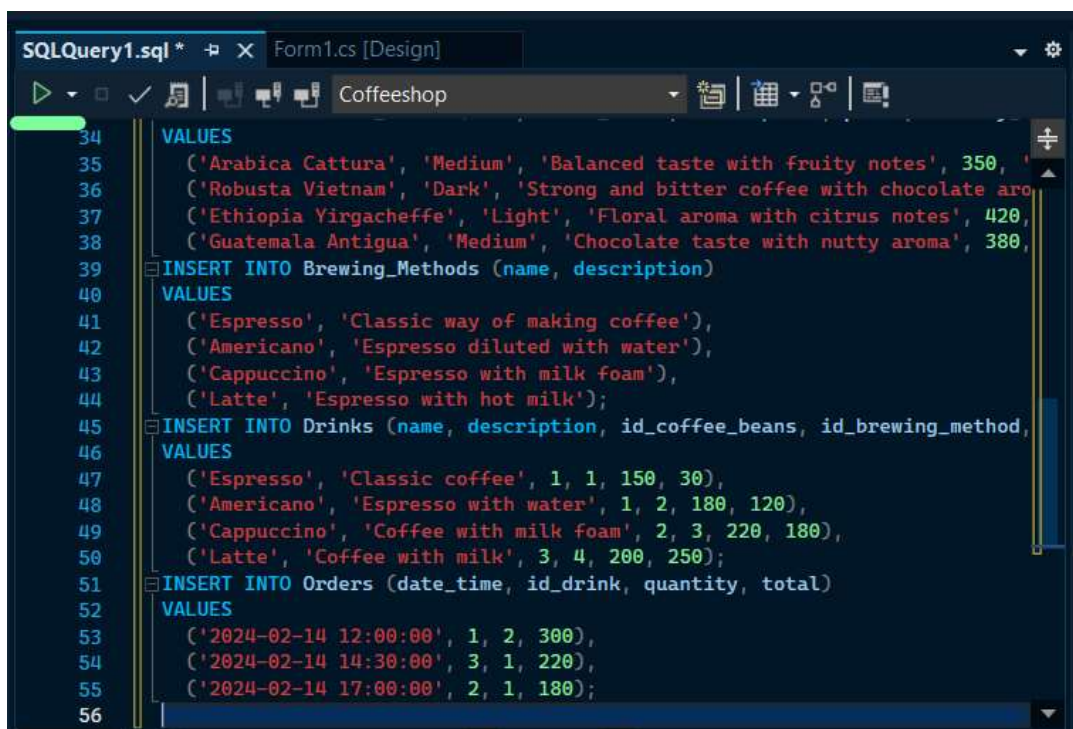


Рисунок 11 – Выполнение SQL-запроса

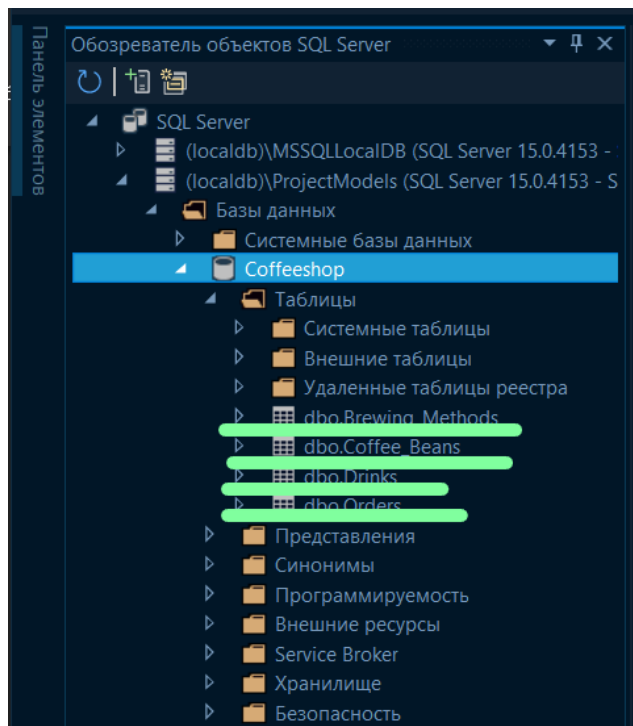


Рисунок 12 – Результат выполнения запроса

После успешного добавления таблиц и данных в базу, необходимо создать новое подключение для дальнейшей работы с базой данных. Для этого, необходимо перейти во вкладку **Обозреватель серверов** и нажать на кнопку **Добавить SQL Server** (рисунки 13-14). В настройках сервера необходимо созданную ранее базу данных.

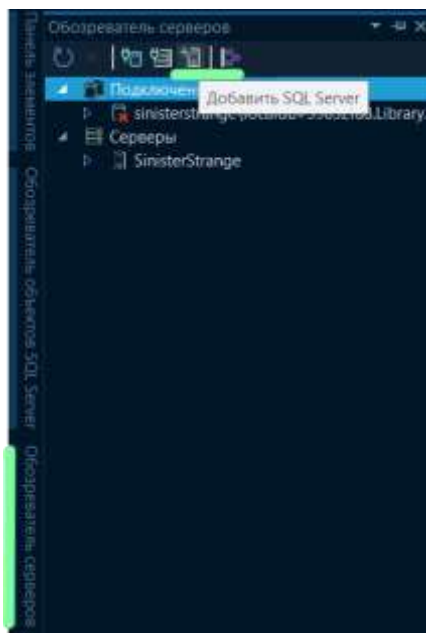


Рисунок 13 – Добавление SQL Server

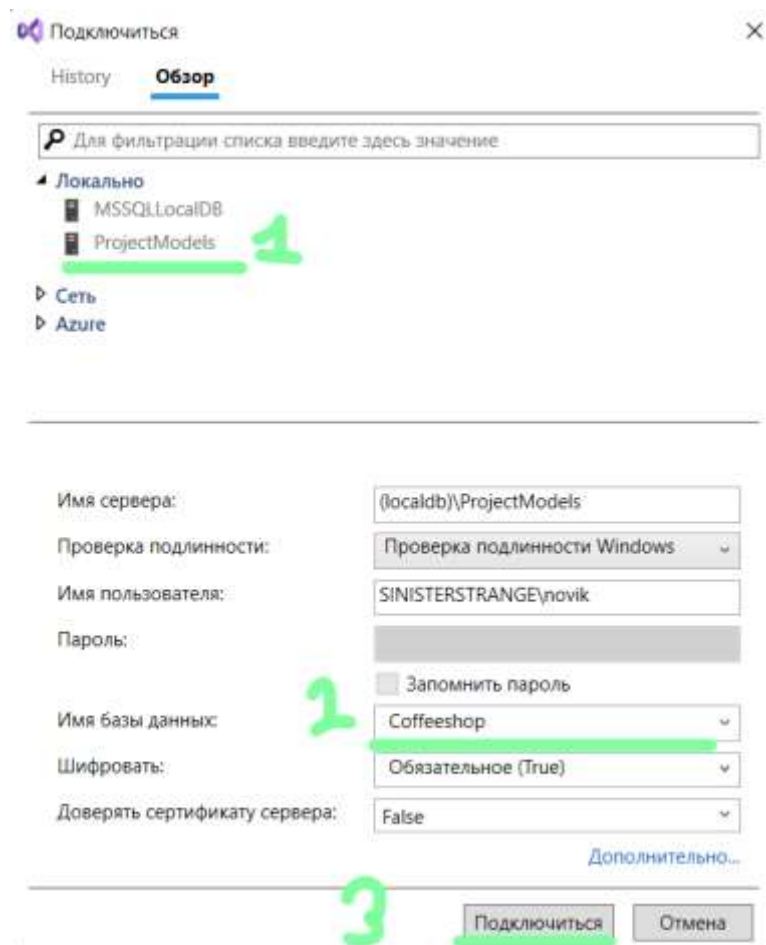


Рисунок 14 - Добавление SQL Server

Далее используем ПКМ на пункте Подключения данных и выберем пункт Добавить подключение (рисунок 15).

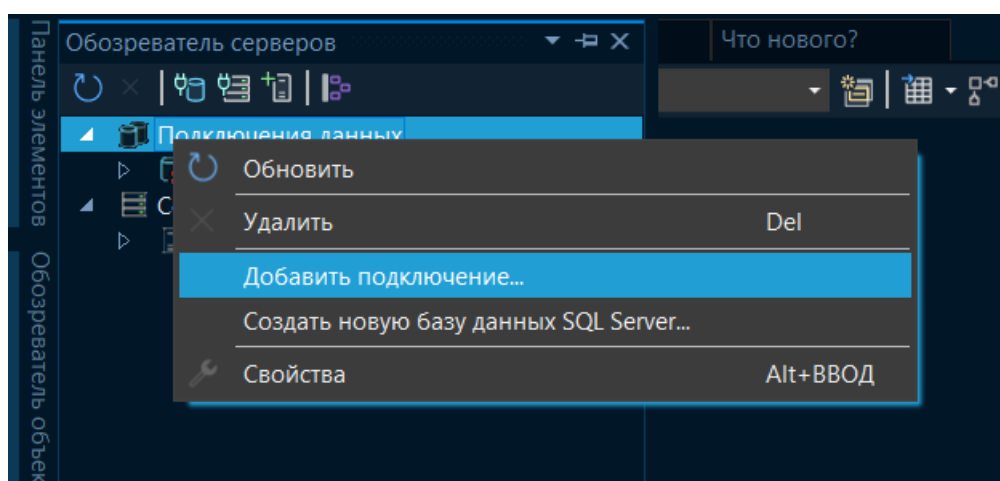


Рисунок 15 – Добавление нового подключения к БД

В открывшемся окне выбираем пункт Microsoft SQL Server (рисунок 16).

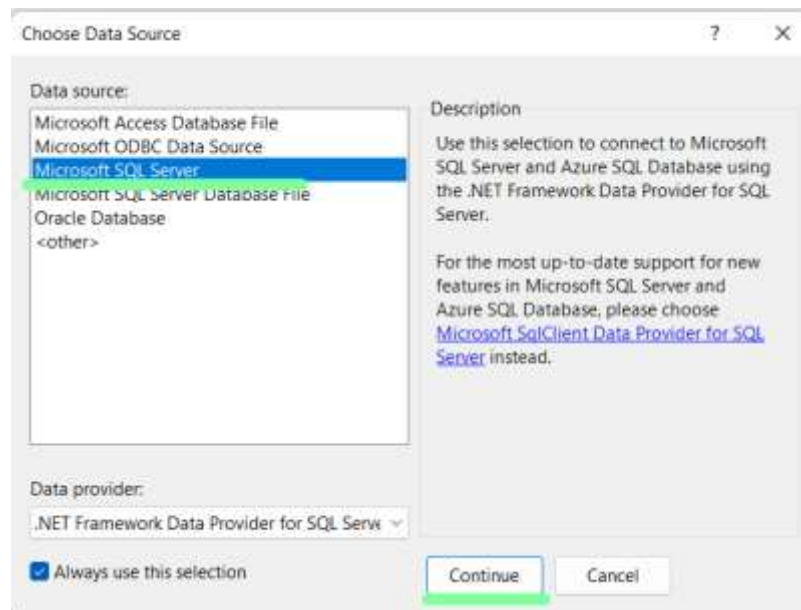


Рисунок 16 – Выбор источника данных

Если по каким-то причинам Вам выдает ошибку вида "The connection properties object must be of type SqlConnectionProperties or OleDbSqlConnectionProperties", Вам необходимо удалить из файлов machine.config, находящихся по двум путям – C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config и C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config, строку, начинающуюся на DbProviderFactories и перезагрузить IDE.

Далее необходимо подключиться к серверу и выбрать созданную БД. После это следует проверить подключение нажав на соответствующую кнопку. Если соединение прошло успешно, можно нажимать на кнопку ОК (рисунок 17).



Рисунок 17 – Добавление соединения

Если все прошло хорошо, Вы увидите новое подключение к БД (рисунок 18).

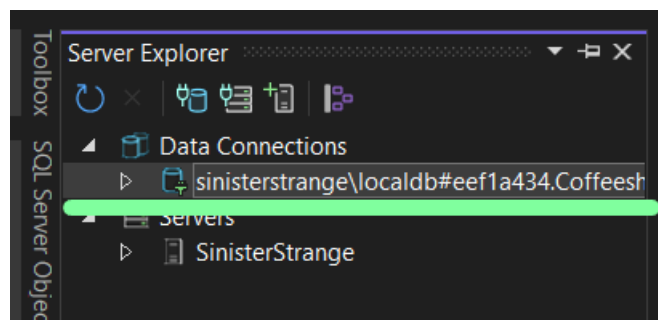


Рисунок 18 – Установленное подключение к БД

5. Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм и теоретическую объектную модель решения задачи.
3. Запрограммировать полученные алгоритмы и объектную модель.

6. Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Перечислите основные технологии доступа к данным.
2. Какая СУБД встроена в Visual Studio.
3. Дайте понятие определению «база данных»
4. Кратко опишите процесс создания БД средствами Visual Studio и подключение ее к проекту.

8. Рекомендуемая литература

1. **Рихтер, Дж.** CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. СПб. : Изд-во Питер, 2021. 896 с.
2. **Прайс, М. Дж.** C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.
3. **Васильев, А.Н.** Программирование на C# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.
4. **Фримен, А.** ASP.NET Core 3 с примерами на C# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.