**Применение Dapper**

При создании подключения вместо класса провайдера connection используется интерфейс IDbConnection, который Dapper дополняет методами Query или Execute

Алгоритм работы:

1. Создание подключения

using Microsoft.Data.SqlClient;

using System.ComponentModel;

using System.Data;

Console.WriteLine("Dapper");

// создание подключения

SqlConnectionStringBuilder builder = new()

{

DataSource = "PRSERVER\\SQLEXPRESS",

InitialCatalog = "ispp2111",

UserID = "ispp2111",

Password = "2111",

TrustServerCertificate = true

};

using IDbConnection connection = new SqlConnection();

2. Создание запроса (может быть несколько запросов, разделяются ;)

// создание запроса

string query = "...";

3. Выполнение запроса (Параметром может быть объект пользовательского типа или анонимного, параметр пишется без @)

// выполнение запроса

connection.МетодDapper(query, параметры);

await connection.МетодDapperAsync(query, new{Имяпараметра1 = знач1, Имяпараметра2 = знач2 });

Методы Dapper:

1. Query

Query<T> - вернет набор объектов типа(Т)

QuerySingle<T> - вернет единственную строку (если больше, то ошибка)

QueryFirst<T> - вернет первую строку

QuerySingleOrDefault / QueryFirstOrDefaul – вернет Null, если ничего не найдет

2. Execute DML( UPDATE, INSERT, DELETE)

ExecuteScalar<T> – вернет простое значение

ExecuteMultiple – позволяет получить данные или результаты нескольких запросов

using Dapper;

using Microsoft.Data.SqlClient;

using System.ComponentModel;

using System.Data;

using Lection2910;

Console.WriteLine("Dapper");

SqlConnectionStringBuilder builder = new()

{

DataSource = "PRSERVER\\SQLEXPRESS",

InitialCatalog = "ispp2111",

UserID = "ispp2111",

Password = "2111",

TrustServerCertificate = true

};

using IDbConnection connection = new SqlConnection(builder.ConnectionString);

string query = "SELECT \* FROM Category";

var categories = await connection.QueryAsync<Category>(query);

Console.WriteLine();

query = "SELECT \* FROM Category WHERE CategoryId = @Id";

var category = await connection.QuerySingleOrDefaultAsync<Category>(query, new { Id = 2});

Console.WriteLine();

query = "UPDATE Category SET Name = 'RPG' WHERE Name = @Name";

await connection.ExecuteAsync(query, new { Name = "РПГ" });

Console.WriteLine();

query = "INSERT INTO Category(Name) VALUES(@Name); SELECT SCOPE\_IDENTITY()";

Category arcada = new Category {Name = "аркада" };

int id = await connection.ExecuteScalarAsync<int>(query, arcada);

Console.WriteLine(id);