CountryWeb\Data\ApplicationDbContext.cs

Формируется контекст базы данных, где указано, что она состоит из трех таблиц:

Cities, Countries, Regions.

Вид и поля таблиц создаются в соответствии с моделями, закрепленными за таблицами, которые располагаются в CountryWeb\Models.

Узнать модель, которая закреплена за таблицей БД можно с помощью строки

public DbSet<City> Cities { get; set; }, где <City> это название класса из CountriesWeb.Models, а Cities это название таблицы в БД.

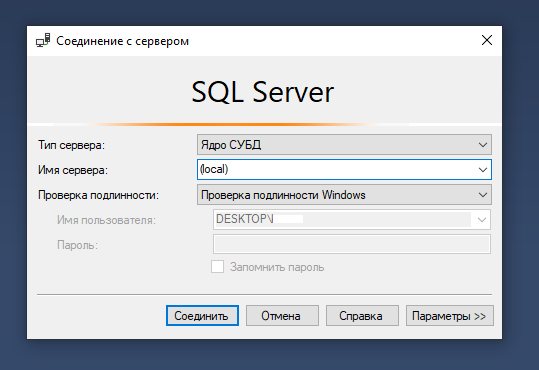
В CountryWeb\appsettings.json указан путь подключения к базе данных:

"ConnectionStrings": {

"DefaultConnection": "Server=(local); Database=CountriesDb; Trusted\_Connection=True; MultipleActiveResultSets=True"

},

Используется локальный сервер (Server=(local)). Это равносильно такому подключению к MSSQL:



Название базы данных CountriesDb(Database=CountriesDb)

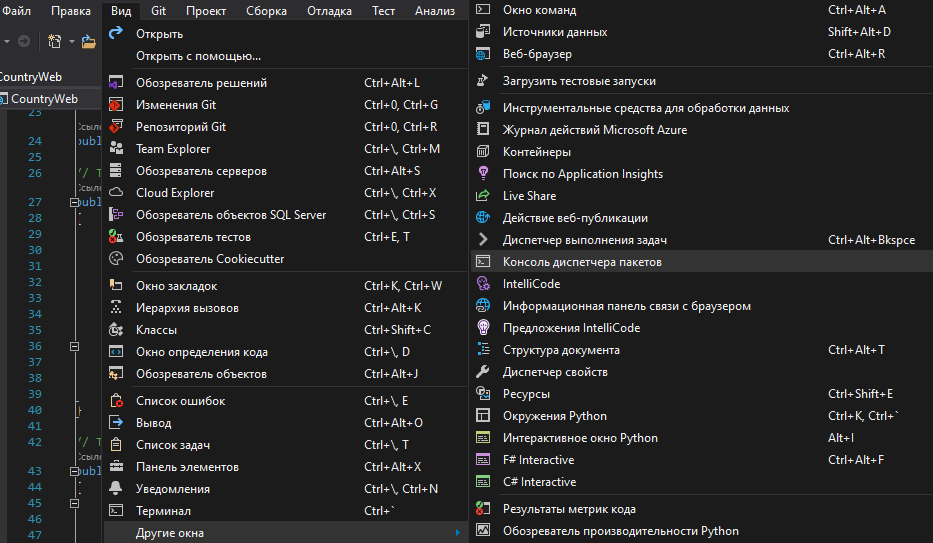
В CountryWeb\Startup.cs

public void ConfigureServices(IServiceCollection services)

происходит включение БД в проект:

services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

Через консоль диспетчера пакетов вводим команды для миграции базы данных.



add-migration <name>

После этой команды в папке Migrations создается файл для изменения БД

update-database

Изменение БД на основе файла миграции

Если ранее вы не создавали в MSSQL базу данных CountriesDb и таблицы, то после выполнения миграции и изменения базы данных вы можете выполнить соединение с сервером с помощью MSSQL Management Studio и у вас будут созданы база данных вместе с таблицами, параметры которых будут соответствовать подключенным в ApplicationDbContext моделям проекта

