

## Информация по подключению проекта

### План:

- Скачать проект с репозитория GitHub;
- Создать базу данных и запустить скрипт из репозитория;
- Поменять подключение сервера в программе;
- Запустить программу.

### Пункт 2(Создать базу данных и запустить скрипт)

- Запускаем SSMS, переносим скрипт в рабочую область;
- Создаем на сервере базу данных с название NotesDB;
- После чего запускаем скрипт;
- База данных создана, данные таблиц и диаграмма из скрипта перенесены.

### Пункт 3(Поменять подключение сервера в программе)

- Заходим в папку Notes Program;
- Запускаем файл Notes Program.sln в Visual Studio;
- Находим в проекте файл App.Config;
- В нем меняем значение source(название сервера) в строке подключение на название вашего сервера;

```
<connectionStrings>  
<add name="NotesDBEntities" connectionString="metadata=res://*/Model.Model.csdl|res://*/Model.Model.ssdl|res://*/Model.Model.msl;provider=System.Data.SqlClient;provider connection string="data source=ARTEM\SQLEXPRESS;
```

```
source=ARTEM\SQLEXPRESS;
```

- Запускаем программу.

**Если программа выдает ошибку связанную с GetContext(), то нужно:**

- Перейти в проекте в папку Model;
- Найти файл Model.Context.cs;
- В нем прописать GetContext(), рисунок ниже.

```
namespace Notes_Program.Model  
{  
    using System;  
    using System.Data.Entity;  
    using System.Data.Entity.Infrastructure;  
  
    public partial class NotesDBEntities : DbContext  
    {  
        public NotesDBEntities()  
            : base("name=NotesDBEntities")  
        {  
        }  
  
        private static NotesDBEntities _context;  
  
        public static NotesDBEntities GetContext()  
        {  
            if (_context == null)  
                _context = new NotesDBEntities();  
            return _context;  
        }  
  
        protected override void OnModelCreating(DbModelBuilder modelBuilder)  
        {  
            throw new UnintentionalCodeFirstException();  
        }  
  
        public virtual DbSet<Notes> Notes { get; set; }  
        public virtual DbSet<Role> Role { get; set; }  
        public virtual DbSet<sysdiagrams> sysdiagrams { get; set; }  
        public virtual DbSet<User> User { get; set; }  
    }  
}
```