**Практическая работа №1. Создание БД в MS SQL Server. Подключение к СУБД.**

1. Задание.

1. Разработать систему классов по предметной области.
2. Реализовать эту систему в приложении WPF.Майкрософт (CodeFirst).
3. При запуске приложения должна создаваться БД MS SQL Server на локальном сервере.
4. Заполнить БД тестовыми данными.
5. Во втором приложении реализовать технологию BaseFirst для подключения к готовой БД.
6. В итоге: два приложения WPF.Майкрософт, скрипт БД, отчет.

2. Варианты предметных областей

1. Андреев

Предметная область: *Библиотека.* Минимальный список характеристик:

* номер читательского билета,
* ФИО читателя,
* адрес,
* телефон,
* название книги,
* дата выдачи книги читателю,
* дата сдачи книги читателем.

3. Ход выполнения работы:

1. Запуск IDE VisualStudio
2. Создание проекта типа WPF (FrontUniversity), а также библиотеки классов WPF (ClassUniversity)
3. Описание классов в ClassUniversity:

* Класс Exam.cs:

public class Exam

{

public Exam(string? examdate, int examevaluation)

{

this.ExamDate = examdate;

this.ExamEvaluation = examevaluation;

}

public Exam()

{

}

public int ExamId { get; set; }

public string? ExamDate { get; set; }

public int ExamEvaluation { get; set; }

public int SubjectId { get; set; }

public Subject? Subject { get; set; }

}

* Класс Group.cs:

public class Group

{

public Group(string? groupname)

{

this.GroupName = groupname;

}

public Group()

{

}

public int GroupId { get; set; }

public string? GroupName { get; set; }

public List<Subject> Subjects { get; set; } = new();

}

* Класс Teacher.cs:

public class Teacher

{

public Teacher(string? firstname, string? middlename, string? lastname)

{

this.FirstName = firstname;

this.MiddleName = middlename;

this.LastName = lastname;

}

public Teacher()

{

}

public int TeacherId { get; set; }

public string? FirstName { get; set; }

public string? MiddleName { get; set; }

public string? LastName { get; set; }

public List<Subject> Subjects { get; set; } = new();

}

Класс Subject.cs:

public class Subject

{

public Subject(string? subjectname, int subjectnumberofhours)

{

this.SubjectName = subjectname;

this.ESubjectNumberOfHours = subjectnumberofhours;

}

public Subject()

{

}

public int SubjectId { get; set; }

public string? SubjectName { get; set; }

public int ESubjectNumberOfHours { get; set; }

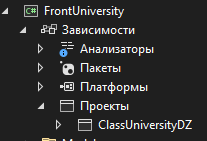
public Exam? Exa { get; set; }

public List<Group> Groups { get; set; } = new();

public List<Teacher> Teachers { get; set; } = new();

}

1. Добавляем зависимость на проект ClassUniversity для проекта FrontUniversity:



5. Создаем в проекте папку Models, а в ней класс Aplications устанавливаем таблицы и подключаемся к SQLServer:

public class ApplicationContext : DbContext

{

public DbSet<Exam> Exams { get; set; } = null!;

public DbSet<Group> Groups { get; set; } = null!;

public DbSet<Subject> Subjects { get; set; } = null!;

public DbSet<Teacher> Teachers { get; set; } = null!;

public ApplicationContext() => Database.EnsureCreated();

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

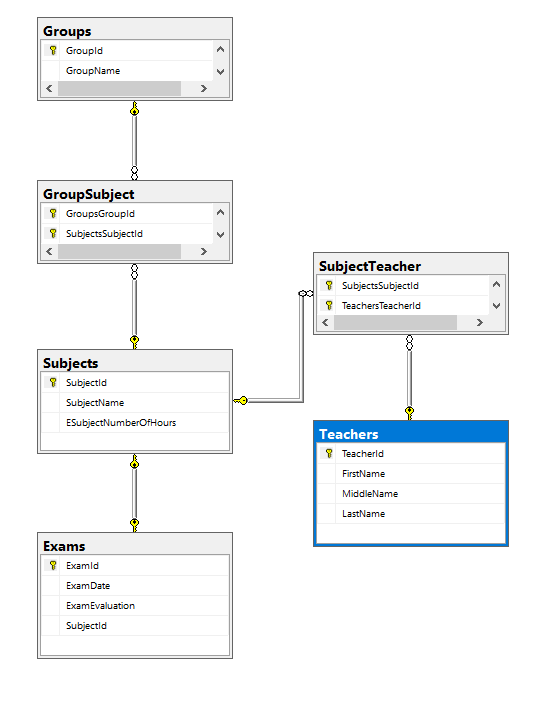
{

optionsBuilder.UseSqlServer(@"Server=44-3\MSSQLSERVER01;Database=university;TrustServerCertificate=True;Trusted\_Connection=True");

}

}

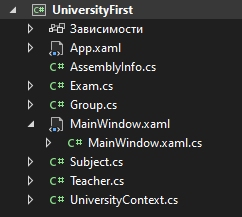
6. Запускаем проект после в SQLServer создается БД:



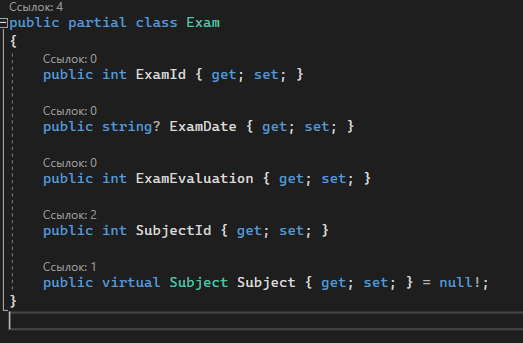
7. Создаем еще один проект UniversityFirst и реализовываем технологию BaseFirst Запуская консоль данного проекта и прописываем:

Scaffold-DbContext "Server=44-3\MSSQLSERVER01;Database=university;TrustServerCertificate=True;Trusted\_Connection=True" Microsoft.EntityFrameworkCore.SqlServer

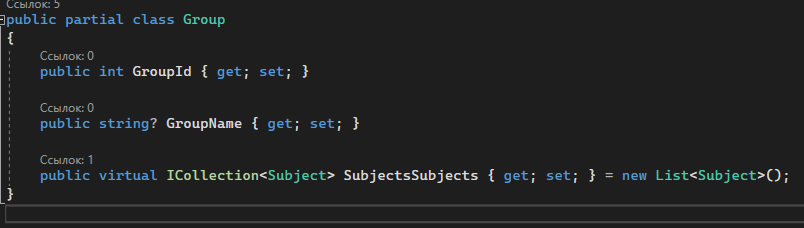
8. После этого в проекте создаются классы:



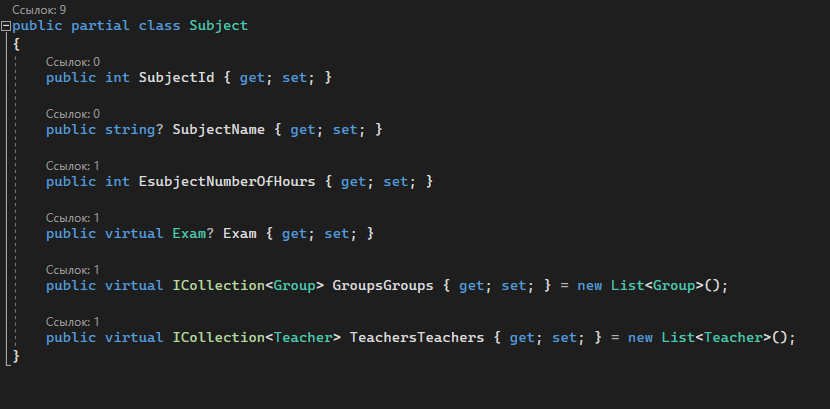
* Exam.cs:



* Group.cs:



* Subject.cs:



* Teacher.cs:

