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

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.

Вариант предметных областей

1. Смирнов Я.

Предметная область: *Личные данные о студентах.* Минимальный список характеристик:

* код студента,
* ФИО студента,
* курс,
* специальность,
* дата рождения студента,
* номер группы,
* сумма стипендии студента,
* год зачисления.

Ход выполнения работы:

1.Запуск IDE VisualStudio

2.Создание проекта типа WPF (DataBaseUI), а также библиотеки классов WPF (DataBaseLibrary)

3.Описание классов

Класс Students: public class Students

{

public Students(string FIO, int Age, int Kyrs,int SpecialnostId, string DateBrithDay, string NumberGroup, int Stipendiya, int YearPostypleniya)

{

this.FIO = FIO;

this.Age = Age;

this.Kyrs = Kyrs;

this.SpecialnostId = SpecialnostId;

this.DateBrithDay = DateBrithDay;

this.NumberGroup = NumberGroup;

this.Stipendiya = Stipendiya;

this.YearPostypleniya = YearPostypleniya;

}

public int Id { get; set; }

public string? FIO { get; set; }

public int Age { get; set; }

public int Kyrs { get; set; }

public int SpecialnostId { get; set; }

public string? DateBrithDay { get; set; }

public string? NumberGroup { get; set; }

public int Stipendiya { get; set; }

public int YearPostypleniya { get; set; }

public Specialnost? Specialnosts { get; set; }

}

Класс Specialnost: public class Specialnost

{

public int Id { get; set; }

public string? Name { get; set; }

List<Students> Students { get; set; } = new List<Students>();

}

5.Установим пакеты NuGet Microsoft.EntityFrameWorkCore.SqlServer

6. Application context(чтоб работала бд и все остальное с данными): public class ApplicationContext : DbContext

{

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

public DbSet<Specialnost> Specialnosts { get; set; } = null!;

public ApplicationContext()

{

Database.EnsureCreated();

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

try { optionsBuilder.UseSqlServer(@"Data Source=DESKTOP-1NMG4AP\SQLEXPRESS;Initial Catalog=ZXCInfStudents;Integrated Security=True;"); }

catch { optionsBuilder.UseSqlServer(@"Data Source=44-6\SQLEXPRESS01;Initial Catalog=ZXCInfStudents;Integrated Security=True;"); }

}

7.логика создания студента: public AddStudents()

{

InitializeComponent();

using (var db = new ApplicationContext())

{

cmbSpecialnost.ItemsSource = db.Specialnosts.ToList();

cmbSpecialnost.SelectedValuePath = "Id";

cmbSpecialnost.DisplayMemberPath = "Name";

}

}

private void btnSavechanges\_Click(object sender, RoutedEventArgs e)

{

string studName = txtNameUser.Text;

int studAge = Convert.ToInt32(txtAgeUser.Text);

int studKyrs = Convert.ToInt32(txtKyrsUser.Text);

string studDateBrithDay = txtDateBrtithDayUser.Text;

string studNumbGroup = txtNumberGroupUser.Text;

int studStipendiya = Convert.ToInt32(txtStipendiyaUser.Text);

int studYearPostypleniya = Convert.ToInt32(txtYearPostypleniyaUser.Text);

Specialnost SpecId = (Specialnost)cmbSpecialnost.SelectedItem;

using (ApplicationContext context = new ApplicationContext())

{

// string studSpecialnostName = cmbSpecialnost.SelectedItem?.ToString() ?? "";

// var specialnost = context.Specialnosts.FirstOrDefault(s => s.Name == studSpecialnostName);

// if (specialnost == null)

// {

// MessageBox.Show("Специальность не найдена.");

// return;

// }

Students students = new Students(studName, studAge, studKyrs,SpecId.Id, studDateBrithDay, studNumbGroup, studStipendiya, studYearPostypleniya);

context.Students.Add(students);

context.SaveChanges();

MessageBox.Show("Студент добавлен");

}

}

8.Отображение данных с бд: xmal- <DataGrid Name="dtgUsers" AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="Id"

Binding="{Binding Id}" Width="50"/>

<DataGridTextColumn Header="FIO"

Binding="{Binding FIO}" Width="220"/>

<DataGridTextColumn Header="Age"

Binding="{Binding Age}" Width="50"/>

<DataGridTextColumn Header="Kyrs"

Binding="{Binding Kyrs}" Width="50"/>

<DataGridTextColumn Header="Specialnost"

Binding="{Binding Specialnosts.Name}" Width="100"/>

<DataGridTextColumn Header="DateBrithDay"

Binding="{Binding DateBrithDay}" Width="100"/>

<DataGridTextColumn Header="NumberGroup"

Binding="{Binding NumberGroup}" Width="100"/>

<DataGridTextColumn Header="Stipendiya"

Binding="{Binding Stipendiya}" Width="80"/>

<DataGridTextColumn Header="YearPostypleniya"

Binding="{Binding YearPostypleniya}" Width="100"/>

</DataGrid.Columns>

</DataGrid>

Cs- public Glav()

{

using (ApplicationContext zxc = new ApplicationContext())

{

students = zxc.Students.Include(s => s.Specialnosts)

.ToList();

}

InitializeComponent();

dtgUsers.ItemsSource = students;

}

9.Удаление: private void btnDeleteStudents\_Click(object sender, RoutedEventArgs e)

{

List<Students> studentsForRemoving = dtgUsers.SelectedItems.Cast<Students>().ToList();

if (studentsForRemoving.Count > 0)

{

foreach (Students user in studentsForRemoving)

{

using (ApplicationContext zxc = new ApplicationContext())

{

zxc.Students.Remove(user);

zxc.SaveChanges();

}

}

using (ApplicationContext zxc = new ApplicationContext())

{

dtgUsers.ItemsSource = zxc.Students.ToList();

}

MessageBox.Show($"Выбранные пользователи({studentsForRemoving.Count}) удалены");

}

else

{

MessageBox.Show("Вы не выбрали ни одного пользователя");

}

}

10.редактирование(я не обещаю что у вас это будет работать,поиск ошибки занял в районе 3ч.)

public partial class RedactStudent : Page

{

private Students \_student { get; set; }

public RedactStudent(Students students)

{

this.\_student = students;

InitializeComponent();

txtNameUser.Text = students.FIO;

txtAgeUser.Text = Convert.ToString(students.Age);

txtKyrsUser.Text = Convert.ToString(students.Kyrs);

using (var db = new ApplicationContext())

{

cmbSpecialnost.ItemsSource = db.Specialnosts.ToList();

cmbSpecialnost.SelectedValuePath = "Id";

cmbSpecialnost.DisplayMemberPath = "Name";

}

cmbSpecialnost.SelectedIndex = \_student.SpecialnostId-1;

txtDateBrtithDayUser.Text = students.DateBrithDay;

txtNumberGroupUser.Text = students.NumberGroup;

txtStipendiyaUser.Text = Convert.ToString(students.Stipendiya);

txtYearPostypleniyaUser.Text = Convert.ToString(students.YearPostypleniya);

}

private void btnBackRedact\_Click(object sender, RoutedEventArgs e)

{

this.NavigationService.Navigate(new Glav());

}

private void btnSaveRedact\_Click(object sender, RoutedEventArgs e)

{

\_student.FIO = txtNameUser.Text;

\_student.Age = Convert.ToInt32(txtAgeUser.Text);

\_student.Kyrs = Convert.ToInt32(txtKyrsUser.Text);

\_student.Specialnosts = cmbSpecialnost.SelectedItem as Specialnost;

\_student.DateBrithDay = txtDateBrtithDayUser.Text;

\_student.NumberGroup = txtNumberGroupUser.Text;

\_student.Stipendiya = Convert.ToInt32(txtStipendiyaUser.Text);

\_student.YearPostypleniya = Convert.ToInt32(txtYearPostypleniyaUser.Text);

using (ApplicationContext zxc = new ApplicationContext())

{

zxc.Students.Update(\_student);

zxc.SaveChanges();

}

MessageBox.Show("Пользователь изменён");

}

}

На этом первая часть закончена,все основные моменты с первой программы вынесены а для больше ознакомления советую самому просмотреть код

2 часть – Подключение к существующей бд

1. Вид - другие окна- консоль диспетчера пакетов- Scaffold-DbContext "Server=DESKTOP-1NMG4AP\SQLEXPRESS;Database=ZXCInfStudents;Trusted\_Connection=True;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer

После этого появляется –

Build started...

Build succeeded.

1. To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see <http://go.microsoft.com/fwlink/?LinkId=723263>.

Дальше появляется файл cs с названием вашей лок.бд с кодом public partial class ZXCInfStudentsContext : DbContext

{

public ZXCInfStudentsContext()

{

}

public ZXCInfStudentsContext(DbContextOptions<ZXCInfStudentsContext> options)

: base(options)

{

}

public virtual DbSet<Specialnost> Specialnosts { get; set; } = null!;

public virtual DbSet<Student> Students { get; set; } = null!;

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see http://go.microsoft.com/fwlink/?LinkId=723263.

optionsBuilder.UseSqlServer("Server=DESKTOP-1NMG4AP\\SQLEXPRESS;Database=ZXCInfStudents;Trusted\_Connection=True;TrustServerCertificate=True;");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Student>(entity =>

{

entity.HasIndex(e => e.SpecialnostId, "IX\_Students\_SpecialnostId");

entity.Property(e => e.Fio).HasColumnName("FIO");

entity.HasOne(d => d.Specialnost)

.WithMany(p => p.Students)

.HasForeignKey(d => d.SpecialnostId);

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

1. }

3.Отображение данных почти тоже самое только в cs: public List<Student> students = new List<Student>();

public MainWindow()

{

InitializeComponent();

using (ZXCInfStudentsContext zxc = new ZXCInfStudentsContext())

{

students = zxc.Students.ToList();

dtgUsers.ItemsSource = zxc.Students.Include(t => t.Specialnost).ToList();

}

С WPF Всё

3БД SQL: USE [master]

GO

/\*\*\*\*\*\* Object: Database [ZXCInfStudents] Script Date: 13.09.2024 22:08:35 \*\*\*\*\*\*/

CREATE DATABASE [ZXCInfStudents]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'ZXCInfStudents', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\ZXCInfStudents.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'ZXCInfStudents\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\ZXCInfStudents\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT, LEDGER = OFF

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [ZXCInfStudents].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [ZXCInfStudents] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [ZXCInfStudents] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [ZXCInfStudents] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [ZXCInfStudents] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [ZXCInfStudents] SET ARITHABORT OFF

GO

ALTER DATABASE [ZXCInfStudents] SET AUTO\_CLOSE ON

GO

ALTER DATABASE [ZXCInfStudents] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [ZXCInfStudents] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [ZXCInfStudents] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [ZXCInfStudents] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [ZXCInfStudents] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [ZXCInfStudents] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [ZXCInfStudents] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [ZXCInfStudents] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [ZXCInfStudents] SET ENABLE\_BROKER

GO

ALTER DATABASE [ZXCInfStudents] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [ZXCInfStudents] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [ZXCInfStudents] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [ZXCInfStudents] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [ZXCInfStudents] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [ZXCInfStudents] SET READ\_COMMITTED\_SNAPSHOT ON

GO

ALTER DATABASE [ZXCInfStudents] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [ZXCInfStudents] SET RECOVERY SIMPLE

GO

ALTER DATABASE [ZXCInfStudents] SET MULTI\_USER

GO

ALTER DATABASE [ZXCInfStudents] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [ZXCInfStudents] SET DB\_CHAINING OFF

GO

ALTER DATABASE [ZXCInfStudents] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [ZXCInfStudents] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [ZXCInfStudents] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [ZXCInfStudents] SET ACCELERATED\_DATABASE\_RECOVERY = OFF

GO

ALTER DATABASE [ZXCInfStudents] SET QUERY\_STORE = ON

GO

ALTER DATABASE [ZXCInfStudents] SET QUERY\_STORE (OPERATION\_MODE = READ\_WRITE, CLEANUP\_POLICY = (STALE\_QUERY\_THRESHOLD\_DAYS = 30), DATA\_FLUSH\_INTERVAL\_SECONDS = 900, INTERVAL\_LENGTH\_MINUTES = 60, MAX\_STORAGE\_SIZE\_MB = 1000, QUERY\_CAPTURE\_MODE = AUTO, SIZE\_BASED\_CLEANUP\_MODE = AUTO, MAX\_PLANS\_PER\_QUERY = 200, WAIT\_STATS\_CAPTURE\_MODE = ON)

GO

ALTER DATABASE [ZXCInfStudents] SET READ\_WRITE

GO