Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

По дисциплине «СПП»

за 6-й семестр

Выполнил:

студент 3 курса

группы ПО-4 (2)

Яковчик И.А.

Проверил:

Монтик Н.С.

Брест, 2022

**Цель работы:** приобрести практические навыки разработки баз данных и начальной интеграции БД с кодом C# с помощью EntityFramework.

**Вариант:** 24

База данных «Аренда автомобилей».

**Задание:**

Реализовать базу данных из не менее 5 таблиц на заданную тематику. При реализации продумать типизацию полей и внешние ключи в таблицах. Визуализировать разработанную БД с помощью схемы, на которой отображены все таблицы и связи между ними (пример, схема на рис. 1). На языке Java с использованием JDBC реализовать подключение к БД и выполнить основные типы запросов, продемонстрировать результаты преподавателю и включить тексты составленных запросов в отчет. Основные типы запросов – 1. На выборку/на выборку с упорядочиванием (SELECT); 2. На добавление (INSERT INTO); 3. На удаление (DELETE FROM); 4. На модификацию (UPDATE). Базу данные можно реализовать в любой СУБД (MySQL, PostgreSQL, SQLite и др.)

**Код программы:**

**RentContext.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

namespace Lab\_3.\_1

{

sealed class RentContext : DbContext

{

public DbSet<Rent> Rents { get; set; }

public DbSet<Rate> Rates { get; set; }

public DbSet<Provider> Providers { get; set; }

public DbSet<Client> Clients { get; set; }

public DbSet<Car> Cars { get; set; }

public RentContext()

{

Database.EnsureDeleted();

Database.EnsureCreated();

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer("Server = (localdb)\\mssqllocaldb; Database = RentalCars; Trusted\_Connection = True;");

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Provider>().HasData(

new Provider() { Id = 1, Name = "Audi", Country = "Germany" },

new Provider() { Id = 2, Name = "Lada", Country = "Russia" });

modelBuilder.Entity<Car>().HasData(

new Car() { Id = 1, Name = "Audi-100", Year\_release = "1987", ProviderId = 1 },

new Car() { Id = 2, Name = "Audi-80", Year\_release = "1995", ProviderId = 1 },

new Car() { Id = 3, Name = "Lada Granta", Year\_release = "2000", ProviderId = 2 });

modelBuilder.Entity<Rate>().HasData(

new Rate { Id = 1, Name = "Usual", Price = "150" },

new Rate { Id = 2, Name = "Vip", Price = "300" });

modelBuilder.Entity<Rent>().HasData(

new Rent() { Id = 1, Name = "Month", CarId = 1, ClientId = 1, RateId = 2 },

new Rent() { Id = 2, Name = "Half year", CarId = 2, ClientId = 2, RateId = 2 },

new Rent() { Id = 3, Name = "Day", CarId = 1, ClientId = 3, RateId = 2 },

new Rent() { Id = 4, Name = "Week", CarId = 3, ClientId = 1, RateId = 1 });

modelBuilder.Entity<Client>().HasData(

new Client { Id = 1, Name = "Andrey", Sourname = "Guzarevich", Passport = "AB239842" },

new Client { Id = 2, Name = "Stas", Sourname = "Kotashevich", Passport = "AB1111111" },

new Client { Id = 3, Name = "Alexey", Sourname = "Lud", Passport = "A2222222" },

new Client { Id = 4, Name = "Vlad", Sourname = "Shuk", Passport = "A3333333" });

}

}

}

**Program.cs:**

using System;

using System.Linq;

using Lab\_3.\_1;

using Microsoft.EntityFrameworkCore;

namespace Lab\_3.\_1

{

public class Program

{

public static void Main(string[] args)

{

using RentContext context = new RentContext();

Console.WriteLine("\"SELECT\" Cars:");

var cars = context.Cars.ToList();

foreach (var car in cars)

{

Console.WriteLine($"Name: {car.Name}\n" +

$"Release year: {car.Name}\n");

}

Console.WriteLine("\"SELECT ORDER BY\" Cars:");

var cars1 = context.Cars

.OrderBy(r => r.Year\_release)

.ToList();

foreach (var car in cars1)

{

Console.WriteLine($"Name: {car.Name}\n" +

$"Sourname: {car.Name}\n");

}

Console.WriteLine("\n\"INSERT\" Client:");

Console.WriteLine("\"INSERT\" Before:");

foreach (var client in context.Clients.ToList())

{

Console.WriteLine($"Name: {client.Name}\n" +

$"Passport: {client.Passport}\n" +

$"Sourname: {client.Sourname}\n");

}

context.Clients.AddRange(

new Client { Name = "Andrey", Sourname = "Guzarevich", Passport = "AB239843" },

new Client { Name = "Stas", Sourname = "Kotashevich", Passport = "AB1111112" });

context.SaveChanges();

Console.WriteLine("\"INSERT\" After:");

foreach (var client in context.Clients.ToList())

{

Console.WriteLine($"Name: {client.Name}\n" +

$"Passport: {client.Passport}\n" +

$"Sourname: {client.Sourname}\n");

}

Console.WriteLine("\n\"DELETE\" Client:");

Console.WriteLine("\"DELETE\" Before:");

foreach (var client in context.Clients.ToList())

{

Console.WriteLine($"Name: {client.Name}\n" +

$"Passport: {client.Passport}\n" +

$"Sourname: {client.Sourname}\n");

}

var delete = context.Clients.Where(c => c.Id == 5 || c.Id == 6);

context.Clients.RemoveRange(delete);

context.SaveChanges();

Console.WriteLine("\"DELETE\" After:");

foreach (var client in context.Clients.ToList())

{

Console.WriteLine($"Name: {client.Name}\n" +

$"Passport: {client.Passport}\n" +

$"Sourname: {client.Sourname}\n");

}

Console.WriteLine("\n\"UPADATE\" Before:");

var update1 = context.Clients.First(t => t.Id == 3);

Console.WriteLine($"Passport: {update1.Passport}");

Console.WriteLine("\"UPADATE\" After:");

update1.Passport = "AB0000000";

context.Clients.Update(update1);

context.SaveChanges();

update1 = context.Clients.First(t => t.Id == 3);

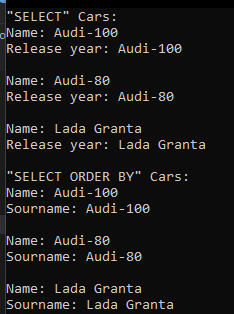
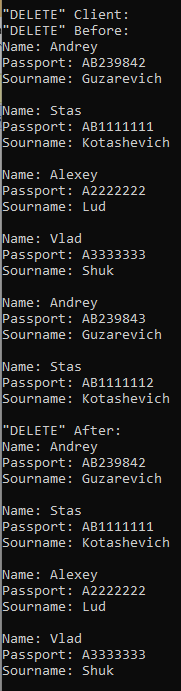
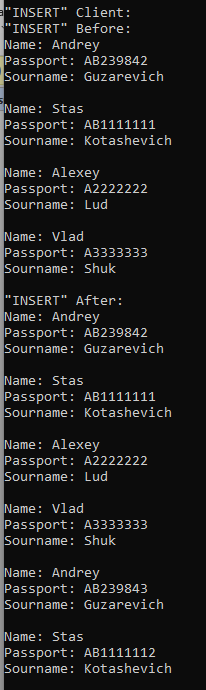
Console.WriteLine($"Passport: {update1.Passport}");

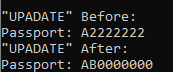
}

}

}

**Результаты работы:**

** **

****

**Выводы:** в ходе выполнения лабораторной работы были получены практические навыки разработки баз данных и начальной интеграции БД с кодом C# с помощью EntityFramework.