**Залялетдинов Фаниль**

1. Задание:

1. Настроить рабочую среду для работы с курсом (MS SQL Server + SSMS + Visual Studio и инструменты .NET).

2. Создать однотабличную БД используя средства SSMS (от 3 до 5 полей не считая id) на произвольную тему. Заполнить базу записями (5-10 записей).

3. Написать программу, которая:

Выполняет подключение к БД, созданное в п.2 задания

Выводит все записи из таблицы БД

Отключается от БД

1. Исходный код:

**Visual Studio**

namespace DZ1

{

internal class Program

{

static SqlConnection OpenDbConntction()

{

string connectionString = ConfigurationManager.ConnectionStrings["LocalDBCollection"].ConnectionString;

SqlConnection conn = new SqlConnection(connectionString);

conn.Open();

return conn;

}

static void ReadQueryResult(SqlDataReader reader)

{

for (int i = 0; i < reader.FieldCount; i++)

{

Console.Write($"{reader.GetName(i)}{(i == reader.FieldCount - 1 ? "" : " - ")}");

}

Console.WriteLine();

if (!reader.HasRows)

{

Console.WriteLine("нет строк");

}

Object[] res = new Object[reader.FieldCount];

while (reader.Read())

{

reader.GetValues(res);

Console.WriteLine(String.Join(" - ", res));

}

Console.WriteLine();

}

static void SelectAll()

{

SqlConnection conn = null;

SqlDataReader reader = null;

try

{

conn = OpenDbConntction();

string queryString = "SELECT r.id, r.recordingDate, c.FIO, c.numberPhone, t.name as 'name of work'\r\n" +

"FROM RegistrationInTheService r\r\nJOIN Clients c on c.id=r.ClientsID\r\nJOIN TypeOfWork t on t.id=r.TypeOfWorkID";

SqlCommand sqlCommand = new SqlCommand(queryString, conn);

reader = sqlCommand.ExecuteReader();

ReadQueryResult(reader);

}

catch (Exception e)

{

Console.WriteLine($"ошибка подключения \n {e.Message}");

}

finally

{

reader?.Close();

conn?.Close();

}

}

static void SelectByID(int id)

{

SqlConnection conn = null;

SqlDataReader reader = null;

try

{

conn = OpenDbConntction();

string queryString = "SELECT r.id, r.recordingDate, c.FIO, c.numberPhone, t.name as 'name of work'\r\n" +

"FROM RegistrationInTheService r\r\nJOIN Clients c on c.id=r.ClientsID\r\nJOIN TypeOfWork t on t.id=r.TypeOfWorkID\r\n" +

$"where r.id=@ID";

SqlCommand sqlCommand = new SqlCommand(queryString, conn);

sqlCommand.Parameters.AddWithValue("@ID", id);

reader = sqlCommand.ExecuteReader();

ReadQueryResult(reader);

}

catch (Exception e)

{

Console.WriteLine($"ошибка подключения \n {e.Message}");

}

finally

{

reader?.Close();

conn?.Close();

}

}

static void Insert(string recordingdate, string FIO, string NameOfWork)

{

SqlConnection con = null;

try

{

con = OpenDbConntction();

string queryString = $"INSERT INTO RegistrationInTheService(recordingDate,TypeOfWorkID,ClientsID)\r\n" +

$"values(@R,(SELECT id FROM TypeOfWork\r\nwhere name like @NW)," +

$"(SELECT id FROM Clients\r\nwhere FIO like @FIO))";

SqlCommand sqlCommand = new SqlCommand(queryString, con);

sqlCommand.Parameters.AddWithValue("@R", recordingdate);

sqlCommand.Parameters.AddWithValue("@NW", NameOfWork);

sqlCommand.Parameters.AddWithValue("@FIO", FIO);

int rowsAffect = sqlCommand.ExecuteNonQuery();

if (rowsAffect == 1)

{

Console.WriteLine("запись успешно добавлена");

}

else

{

Console.WriteLine($"Не удалось добавить запись, rowsAffect ={rowsAffect}");

}

}

catch (Exception e)

{

Console.WriteLine($"Ошибка добавления записи: {e.Message}");

}

finally

{

con?.Close();

}

}

static void Delete(int id)

{

SqlConnection sqlConnection = null;

try

{

sqlConnection = OpenDbConntction();

string queryString = $"DELETE FROM RegistrationInTheService where id=@ID";

SqlCommand sqlCommand = new SqlCommand(queryString, sqlConnection);

sqlCommand.Parameters.AddWithValue("@ID", id);

int rowsAffect = sqlCommand.ExecuteNonQuery();

if (rowsAffect == 1)

{

Console.WriteLine("запись успешно удалена");

}

else

{

Console.WriteLine($"Не удалось удалить запись, rowsAffect ={rowsAffect}");

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка удаления записи: {ex.Message}");

}

finally

{

sqlConnection?.Close();

}

}

static void Update(int id, string recordingdate, string FIO, string NameOfWork)

{

SqlConnection con = null;

try

{

con = OpenDbConntction();

string queryString = $"UPDATE RegistrationInTheService\r\n" +

$"SET recordingDate=@R, TypeOfWorkID=(SELECT id FROM TypeOfWork\r\n" +

$"where name like @NW),ClientsID=(SELECT id FROM Clients\r\n" +

$"where FIO like @FIO)\r\nwhere id=@ID";

SqlCommand sqlCommand = new SqlCommand( queryString, con );

sqlCommand.Parameters.AddWithValue("@ID", id);

sqlCommand.Parameters.AddWithValue("@R", recordingdate);

sqlCommand.Parameters.AddWithValue("@NW", NameOfWork);

sqlCommand.Parameters.AddWithValue("@FIO", FIO);

int rowsAffect= sqlCommand.ExecuteNonQuery();

if (rowsAffect == 1)

{

Console.WriteLine("запись успешно изменена");

}

else

{

Console.WriteLine($"Не удалось изменить запись, rowsAffect ={rowsAffect}");

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка изменения записи: {ex.Message}");

}

finally

{

con?.Close();

}

}

static void Main(string[] args)

{

// 1) вывести все записи

SelectAll();

// 2) вывести запись по id

SelectByID(8);

// 3) добавить запись

Insert("2023-12-26", "Nikiforov Dmitriy", "balancing");

// 4) удалить запись по id

// Delete(9);

// 5) изменить запись по id

Update(10, "2023-11-23", "Petrov Pavel", "Tire service");

// процедурный стиль - статические методы класса Program

SelectAll();

Console.ReadKey();

}

}

}

**SQL:**

create database CarService

use CarService

create table TypeOfWork(

id int primary key identity(1,1),

name varchar(100) unique not null

)

insert into TypeOfWork

values('washing'),('balancing'),('Wheel alignment'),('Tire service')

create table Clients(

id int primary key identity(1,1),

FIO varchar(100) not null unique,

numberPhone bigint not null unique

)

insert into Clients

values('Sidorov Ivan', 89369852365),('Petrov Pavel', 89287896532),('Bulkin Sasha', 89633216598),

('Ivanov Dima', 89526547856),('Nikiforov Dmitriy', 89265568945)

create table RegistrationInTheService(

id int not null primary key identity(1,1),

recordingDate date not null check(recordingDate>getdate()),

TypeOfWorkID int foreign key references TypeOfWork(id),

ClientsID int foreign key references Clients(id)

)

insert into RegistrationInTheService

values('2023-10-19',2,4),('2023-12-23',1,1),('2023-10-19',3,3)

select r.recordingDate, c.FIO, c.numberPhone, t.name

from RegistrationInTheService r

join Clients c on c.id=r.ClientsID

join TypeOfWork t on t.id=r.TypeOfWorkID

where r.recordingDate>'2023-10-20'

1. Скрины

