ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программы

Код базы данных

CREATE DATABASE OrderTrack;

USE OrderTrack;

CREATE TABLE ClientTypes(

ClientTypeID INT PRIMARY KEY IDENTITY(1,1),

ClientType NVARCHAR(150) NOT NULL

);

CREATE TABLE Genders(

GenderID INT PRIMARY KEY IDENTITY(1,1),

Gender NVARCHAR(150) NOT NULL

);

CREATE TABLE Posts(

PostID INT PRIMARY KEY IDENTITY(1,1),

Post NVARCHAR(150) NOT NULL

);

CREATE TABLE Statuses(

StatusID INT PRIMARY KEY IDENTITY(1,1),

Status NVARCHAR(150) NOT NULL

);

CREATE TABLE Clients (

ClientID INT PRIMARY KEY IDENTITY(1,1),

FullName NVARCHAR(150) NOT NULL,

ClientTypeID INT DEFAULT 0,

Email NVARCHAR(100) UNIQUE,

Phone NVARCHAR(18) UNIQUE NOT NULL,

Address NVARCHAR(255),

INN NVARCHAR(10) DEFAULT '',

RegistrationDate DATE DEFAULT GETDATE(),

FOREIGN KEY (ClientTypeID) REFERENCES ClientTypes(ClientTypeID)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

FullName NVARCHAR(150) NOT NULL,

Phone NVARCHAR(18) UNIQUE NOT NULL,

Email NVARCHAR(100) UNIQUE,

GenderID INT DEFAULT 0,

PostID INT DEFAULT 0,

FOREIGN KEY (GenderID) REFERENCES Genders(GenderID),

FOREIGN KEY (PostID) REFERENCES Posts(PostID)

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY IDENTITY(1,1),

ClientID INT NOT NULL,

EmployeeID INT NOT NULL,

OrderDate DATE DEFAULT GETDATE(),

TotalAmount FLOAT DEFAULT 0.00,

StatusID INT DEFAULT 0,

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID) ON DELETE CASCADE,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID),

FOREIGN KEY (StatusID) REFERENCES Statuses(StatusID) ON DELETE CASCADE

);

CREATE TABLE Products (

ProductID INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(150) NOT NULL,

Description NVARCHAR(255),

Price FLOAT NOT NULL

);

CREATE TABLE OrderDetails (

OrderDetailID INT PRIMARY KEY IDENTITY(1,1),

OrderID INT NOT NULL,

ProductID INT NOT NULL,

Price FLOAT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE

);

CREATE TABLE Registration (

RegistrationID INT PRIMARY KEY IDENTITY(1,1),

UserLogin NVARCHAR(50) NOT NULL,

UserPassword NVARCHAR(50) NOT NULL,

IsAdmin BIT DEFAULT 0

);

CREATE TRIGGER trg\_InsertOrderDetailsPrice

ON OrderDetails

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

UPDATE od

SET od.Price = p.Price

FROM OrderDetails od

INNER JOIN inserted i ON od.OrderDetailID = i.OrderDetailID

INNER JOIN Products p ON i.ProductID = p.ProductID;

END;

CREATE TRIGGER trg\_UpdateOrderTotalAmount

ON OrderDetails

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

UPDATE o

SET o.TotalAmount = (

SELECT COALESCE(SUM(od.Price), 0)

FROM OrderDetails od

WHERE od.OrderID = o.OrderID

)

FROM Orders o

INNER JOIN (SELECT DISTINCT OrderID FROM inserted UNION SELECT DISTINCT OrderID FROM deleted) AS affectedOrders

ON o.OrderID = affectedOrders.OrderID;

END;

INSERT INTO ClientTypes (ClientType) VALUES ('Физическое лицо'), ('Юридическое лицо');

INSERT INTO Genders (Gender) VALUES ('Мужской'), ('Женский');

INSERT INTO Posts (Post) VALUES ('Менеджер по продажам'), ('Администратор');

INSERT INTO Statuses (Status) VALUES ('В обработке'), ('Подтвержден'), ('Отменен'), ('Выполнен');

INSERT INTO Clients (FullName, ClientTypeID, Email, Phone, Address, INN)

VALUES

('Иванов Иван Иванович', 1, 'ivanov@mail.ru', '+79991234567', 'г. Москва, ул. Ленина, д. 10', ''),

('ООО "Ромашка"', 2, 'romashka@mail.ru', '+79991112233', 'г. Санкт-Петербург, пр. Невский, д. 25', '0987654321');

INSERT INTO Employees (FullName, Phone, Email, GenderID, PostID)

VALUES

('Петров Петр Петрович', '+7 (999) 888-77-66', 'petrov@mail.ru', 1, 1),

('Сидорова Анна Сергеевна', '+7 (999) 555-44-33', 'sidorova@mail.ru', 2, 2);

INSERT INTO Products (Name, Description, Price)

VALUES

('Квартира в доме 1', 'Дом 1', 85000000.00),

('Квартира в доме 12', 'Дом 2', 3500000.00),

('Квартира в доме 13', 'Дом 3', 1500000.00);

INSERT INTO Orders (ClientID, EmployeeID, StatusID)

VALUES

(1, 1, 1),

(2, 2, 2);

INSERT INTO OrderDetails (OrderID, ProductID)

VALUES

(1, 1),

(1, 2),

(2, 3);

INSERT INTO Registration (UserLogin, UserPassword, IsAdmin) VALUES

('admin', 'admin', 1),

('user', 'user', 0);

SELECT \* FROM ClientTypes;

SELECT \* FROM Genders;

SELECT \* FROM Posts;

SELECT \* FROM Statuses;

SELECT \* FROM Clients;

SELECT \* FROM Employees;

SELECT \* FROM Orders;

SELECT \* FROM Products;

SELECT \* FROM OrderDetails;

SELECT \* FROM Registration;

-- 1. Отчет по заказам клиентов

SELECT

o.OrderID AS 'Номер заказа',

c.FullName AS 'Клиент',

e.FullName AS 'Менеджер',

s.Status AS 'Статус',

o.OrderDate AS 'Дата заказа',

o.TotalAmount AS 'Сумма заказа'

FROM Orders o

JOIN Clients c ON o.ClientID = c.ClientID

JOIN Employees e ON o.EmployeeID = e.EmployeeID

JOIN Statuses s ON o.StatusID = s.StatusID

ORDER BY o.OrderDate DESC;

-- 2. Отчет по продажам за месяц

SELECT

YEAR(OrderDate) AS 'Год',

MONTH(OrderDate) AS 'Месяц',

COUNT(OrderID) AS 'Количество заказов',

SUM(TotalAmount) AS 'Общая сумма продаж'

FROM Orders

GROUP BY YEAR(OrderDate), MONTH(OrderDate)

ORDER BY YEAR(OrderDate) DESC, MONTH(OrderDate) DESC;

-- 3. Отчет по популярным товарам

SELECT

p.Name AS 'Товар',

COUNT(od.OrderDetailID) AS 'Количество продаж',

SUM(od.Price) AS 'Общая сумма'

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY p.Name

ORDER BY COUNT(od.OrderDetailID) DESC, SUM(od.Price) DESC;

DROP DATABASE OrderTrack;

Класс DataBase

using System.Data.SqlClient;

namespace OrderTrack

{

public class DataBase

{

private readonly SqlConnection sqlConnection = new(@"Data Source=DESKTOP-06ekml1\SQLEXPRESS;Initial Catalog=OrderTrack;Integrated Security=True");

/// <summary>

/// OpenConnection() вызывается при открытии соединения с базой данных

/// </summary>

public void OpenConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Closed)

{

sqlConnection.Open();

}

}

/// <summary>

/// CloseConnection() вызывается при закрытии соединения с базой данных

/// </summary>

public void CloseConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Open)

{

sqlConnection.Close();

}

}

/// <summary>

/// GetConnection() вызывается при получении соединения с базой данных

/// </summary>

/// <returns></returns>

public SqlConnection GetConnection()

{

return sqlConnection;

}

}

}

Класс Login

using System.Data;

using System.Data.SqlClient;

namespace OrderTrack

{

public partial class Login : Form

{

private readonly DataBase dataBase = new();

public Login()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

}

private void ButtonEnter\_Click(object sender, EventArgs e)

{

var loginUser = textBoxLogin.Text;

var passwordUser = textBoxPassword.Text;

SqlDataAdapter sqlDataAdapter = new();

DataTable dataTable = new();

string querystring = $"select RegistrationID, UserLogin, UserPassword, IsAdmin from Registration where UserLogin = '{loginUser}' and UserPassword = '{passwordUser}'";

SqlCommand sqlCommand = new(querystring, dataBase.GetConnection());

sqlDataAdapter.SelectCommand = sqlCommand;

sqlDataAdapter.Fill(dataTable);

if (dataTable.Rows.Count == 1)

{

MessageBox.Show("Вы успешно вошли!", "Успешно!", MessageBoxButtons.OK, MessageBoxIcon.Information);

Form1 form1 = new(Convert.ToBoolean(dataTable.Rows[0]["IsAdmin"]));

Hide();

form1.ShowDialog();

Show();

}

else

{

MessageBox.Show("Такого аккаунта не существует!", "Аккаунта не существует!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void LabelAuth\_Click(object sender, EventArgs e)

{

Signup formLogin = new();

Hide();

formLogin.ShowDialog();

Show();

}

private void ButtonClear\_Click(object sender, EventArgs e)

{

textBoxLogin.Text = "";

textBoxPassword.Text = "";

}

private void ButtonShow\_Click(object sender, EventArgs e)

{

if (textBoxPassword.PasswordChar == '•')

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword0;

textBoxPassword.PasswordChar = '\0';

}

else

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword1;

textBoxPassword.PasswordChar = '•';

}

}

}

}

Класс SignUp

using System.Data;

using System.Data.SqlClient;

namespace OrderTrack

{

public partial class Signup : Form

{

private readonly DataBase dataBase = new();

public Signup()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

buttonShow.BackgroundImage = Properties.Resources.ShowPassword0;

}

private void ButtonEnter\_Click(object sender, EventArgs e)

{

var login = textBoxLogin.Text;

var password = textBoxPassword.Text;

SqlDataAdapter sqlDataAdapter = new();

DataTable dataTable = new();

string querystringUsers = $"SELECT UserLogin FROM Registration WHERE UserLogin = '{login}'";

SqlCommand sqlCommandUsers = new(querystringUsers, dataBase.GetConnection());

sqlDataAdapter.SelectCommand = sqlCommandUsers;

sqlDataAdapter.Fill(dataTable);

if (dataTable.Rows.Count > 0)

{

MessageBox.Show("Такой логин уже существует!", "Ошибка!");

}

else

{

string querystring = $"INSERT INTO Registration(UserLogin, UserPassword) VALUES('{login}', '{password}')";

SqlCommand sqlCommand = new(querystring, dataBase.GetConnection());

dataBase.OpenConnection();

if (sqlCommand.ExecuteNonQuery() == 1 && textBoxLogin.Text != "")

{

MessageBox.Show("Аккаунт успешно создан!", "Успех!");

Login formLogin = new();

Close();

formLogin.ShowDialog();

}

else

{

MessageBox.Show("Аккаунт не создан!");

}

dataBase.CloseConnection();

}

}

private void ButtonClear\_Click(object sender, EventArgs e)

{

textBoxLogin.Text = "";

textBoxPassword.Text = "";

}

private void ButtonShow\_Click(object sender, EventArgs e)

{

if (textBoxPassword.PasswordChar == '•')

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword0;

textBoxPassword.PasswordChar = '\0';

}

else

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword1;

textBoxPassword.PasswordChar = '•';

}

}

}

}