МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«Харківський політехнічний інститут»

Кафедра стратегічного управління

ЗВІТ

з лабораторної роботи №5

з дисципліни «Стек технологій .NET»

Перевірила

Виконав:

Харків – 2019

**Тема:** Доступ к даним в .NET. Технологія ADO.NET

**Ціль:** Ознайомитися з технологією доступу к даним ADO.NET

**Варіант:** №2 - Склад

**Опис схеми бази даних:**

Схема бази даних зображена на рисунку 1.

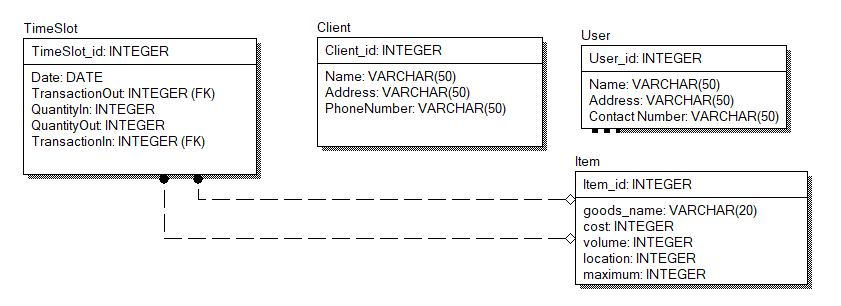


Рисунок 1 – Схема бази даних

Таблиця Client зберігає звичайних клієнтів. Таблиця User призначена для зберігання адміністраторів. В таблиці TimeSlot зберігається інформація щодо товарів поставлених або відгружених зі складу у різні проміжки часу. Таблиця Item призначена для зберігання інформації про товар.

Код програми, що містить функції для з’єднання з базою даних за допомогою ADO .NET (для компонента Клієнт) наведено нижче:

Файл MockClientRepository.cs

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.SqlTypes;

namespace lab4\_empty.Models

{

public class MockClientRepository : IClientRepository

{

private List<Client> \_clientList;

private string connectionString = "Data Source = localhost\\SQLExpress; Initial Catalog = lab5; Integrated Security = True";

public MockClientRepository()

{

}

public Client Add(Client client)

{

client.Id = \_clientList.Max(e => e.Id) + 1;

//\_clientList.Add(client);

using (SqlConnection con = new SqlConnection(connectionString))

{

String query = @"insert into Clients values (" + client.Id +

@", '" + client.Name + "'" +

@", '" + client.Address + "'" +

@", '" + client.PhoneNumber + "')";

SqlCommand cmd = new SqlCommand(query, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

return client;

}

public void Delete(int id)

{

using (SqlConnection con = new SqlConnection(connectionString))

{

String query = "delete from Clients where Id = " + id;

SqlCommand cmd = new SqlCommand(query, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

}

public Client GetClient(int Id)

{

Client client = new Client();

using (SqlConnection con = new SqlConnection(connectionString))

{

String query = "select \* from Clients where Id = " + Id;

SqlCommand command = new SqlCommand(query, con);

try

{

con.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

client.Id = Convert.ToInt32(reader[0]);

client.Name = reader[1].ToString();

client.Address = reader[2].ToString();

client.PhoneNumber = reader[3].ToString();

}

con.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

return client;

//return this.\_clientList.FirstOrDefault(e => e.Id == Id);

}

public IEnumerable GetClients()

{

List<Client> \_clientList = new List<Client>();

using (SqlConnection con = new SqlConnection(connectionString))

{

String query = "select \* from Clients";

SqlCommand command = new SqlCommand(query, con);

try

{

con.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Client client = new Client();

client.Id = Convert.ToInt32(reader[0]);

client.Name = reader[1].ToString();

client.Address = reader[2].ToString();

client.PhoneNumber = reader[3].ToString();

\_clientList.Add(client);

Console.WriteLine(client);

}

con.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

return \_clientList;

}

}

}

Висновки: в ході виконання лабораторної роботи було розглянуто та успішно реалізовано на реальному прикладі технологію ADO.NET.