using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Configuration;

namespace ADODemo

{

class Program

{

static SqlConnection connection;

static SqlCommand command;

static SqlConnection GetConnection()

{

//string connectionString = @"data source=adminvm\SQLEXPRESS;initial catalog=Practice;user id=sa;password=pass@123";

string connectionString = ConfigurationManager.ConnectionStrings["MyConnection"].ToString();

connection = new SqlConnection(connectionString);

return connection;

}

static void Menu()

{

Console.WriteLine("1. Insert");

Console.WriteLine("2. Edit");

Console.WriteLine("3.Delete");

Console.WriteLine("4. Search");

Console.WriteLine("5. List");

Console.WriteLine("6. Get No of Employee");

}

static void Main(string[] args)

{

char choice = 'y';

while (choice == 'y')

{

Menu();

Console.WriteLine("Enter Choice");

int ch = Byte.Parse(Console.ReadLine());

switch (ch)

{

case 1:

{

Console.WriteLine("Enter ID");

int id = Byte.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

string name = Console.ReadLine();

Console.WriteLine("Enter Dept");

string dept = Console.ReadLine();

InsertEmployee(id, name, dept);

break;

}

case 2:

{

Console.WriteLine("Enter ID for which to edit record");

int id = Byte.Parse(Console.ReadLine());

Console.WriteLine("Enter Dept");

string dept = Console.ReadLine();

EditEmployee(id, dept);

break;

}

case 3:

{

Console.WriteLine("Enter ID for which to delete record");

int id = Byte.Parse(Console.ReadLine());

DeleteEmployee(id);

break;

}

case 4:

{

Console.WriteLine("Enter ID for which to search record");

int id = Byte.Parse(Console.ReadLine());

GetEmployeeById(id);

break;

}

case 5:

{

List<Employee> list = new List<Employee>();

list = GetEmployees();

foreach(Employee employee in list)

{

Console.WriteLine(employee.Id + " " + employee.Name + " " + employee.Dept); ;

}

break;

}

case 6:

{

GetEmployeesCount();

break;

}

default:

{

Console.WriteLine("Invalid Choice");

break;

}

}

Console.WriteLine("Do you want to repeat");

choice = Convert.ToChar(Console.ReadLine());

}

}

static List<Employee> GetEmployees()

{

List<Employee> list = new List<Employee>();

connection = GetConnection();

command = new SqlCommand();

command.CommandText = "Select \* from employee";

command.Connection = connection;

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

// Employee employee = new Employee();

list.Add(new Employee() { Id = (int)reader[0], Name = reader[1].ToString(), Dept = reader[2].ToString() });

// Console.WriteLine(reader[0].ToString() + reader[1]);

}

}

connection.Close();

command.Dispose();

connection.Dispose();

return list;

}

static void GetEmployeeById(int id)

{

using (connection = GetConnection())

{

using (command = new SqlCommand())

{

command.CommandText = "Select \* from employee where id=@id";

command.Parameters.AddWithValue("@id", id);

command.Connection = connection;

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

Console.WriteLine(reader[0].ToString() + reader[1]);

}

}

connection.Close();

}

}

}

static void InsertEmployee(int id, string name, string dept)

{

int flag = 0;

using (connection = GetConnection())

{

using (command = new SqlCommand())

{

command.CommandText = "Insert into employee(id,name,dept) values(@id,@name,@dept)";

command.Parameters.AddWithValue("@id", id);

command.Parameters.AddWithValue("@name", name);

command.Parameters.AddWithValue("@dept", dept);

command.Connection = connection;

connection.Open();

flag = command.ExecuteNonQuery();

connection.Close();

}

}

if(flag>0)

Console.WriteLine("Record inserted");

}

static void EditEmployee(int id, string dept)

{

connection = GetConnection();

command = new SqlCommand();

command.CommandText = "Update employee set dept=@dept where id=@id";

command.Parameters.AddWithValue("@id", id);

command.Parameters.AddWithValue("@dept", dept);

command.Connection = connection;

connection.Open();

int flag = command.ExecuteNonQuery();

connection.Open();

if (flag > 0)

Console.WriteLine("Record updated");

}

static void DeleteEmployee(int id)

{

connection = GetConnection();

command = new SqlCommand();

command.CommandText = "Delete employee where id=@id";

command.Parameters.AddWithValue("@id", id);

command.Connection = connection;

connection.Open();

int flag = command.ExecuteNonQuery();

connection.Close();

if (flag > 0)

Console.WriteLine("Record deleted");

}

static void GetEmployeesCount()

{

connection = GetConnection();

command = new SqlCommand();

command.CommandText = "Select count(\*) from employee";

command.Connection = connection;

connection.Open();

int count = Convert.ToByte(command.ExecuteScalar());

connection.Close();

Console.WriteLine("No of Employees are" + count);

}

}

}