ADO.Net

Entity Framework

To connect FE with BE

FE BE (Any Data source)

Web Application ADO.Net Database (Sql Server MySql, Oracle)

Console Entity Framework

MS Excel , XML

Windows

In ADO.Net , There is no mapping between classes (model) and ur database tables

Entity Framework : ORM

Object Relational Mapper

1. Step 1:

Using System.Data.SqlClient;

1. Step 2:

Create object of SqlConnection class , within this object , you will pass

SqlConnection con = new SqlConnection(connectionString);

Using System.Configuration

COnfigurationManager.connectionStrings[“con”].ToString();

Step 3:

Create object of SqlCommand(“query” , con);

Step 4:

Con.open();

Step 5: Execute Query

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Configuration;

namespace ConsoleApp35

{

class Program

{

static SqlConnection \_sqlConnection;

SqlCommand \_sqlCommand;

static SqlDataReader \_sqlDataReader;

static void Main(string[] args)

{

Program program = new Program();

string ch = "y";

try

{

while (ch == "y")

{

Console.WriteLine("Menu");

Console.WriteLine("1. Display all Records");

Console.WriteLine("2. Insert Record");

Console.WriteLine("3. Delete Record");

Console.WriteLine("4. Edit Record");

Console.WriteLine("5. Search Record");

Console.WriteLine("Enter choice");

int choice = Convert.ToByte(Console.ReadLine());

switch (choice)

{

case 1: program.GetEmployees(); break;

case 2:

{

int salary;

string name, department;

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Department");

department = Console.ReadLine();

Console.WriteLine("ENter Salary");

salary = int.Parse(Console.ReadLine());

int res = program.InsertEmployee(name, department, salary);

if (res == 1)

Console.WriteLine("Record Inserted");

else

Console.WriteLine("Record not inserted");

break;

}

case 3:

{

int id;

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

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

int res = program.DeleteEmployee(id);

if (res == 1)

Console.WriteLine("Record Deleted");

else

Console.WriteLine("Record not deleted");

break;

}

case 4:

{

int id;

string name;

string department;

int salary;

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

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

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Department");

department = Console.ReadLine();

Console.WriteLine("ENter Salary");

salary = int.Parse(Console.ReadLine());

int res = program.EditEmployee(id, name, department, salary);

if (res == 1)

Console.WriteLine("Record Updated");

else

Console.WriteLine("Record not updated");

break;

}

case 5:

{

int id;

string name;

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

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

name = program.GetEmployeeById(id);

if (name == null)

{

Console.WriteLine("Such Record does not exist");

;

}

else

{

Console.WriteLine("Name is " + name);

}

break;

}

}

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

ch = Console.ReadLine();

}

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

}

public string GetConnectionString()

{

return (ConfigurationManager.ConnectionStrings["con"].ToString());

}

public SqlConnection GetConnection()

{

\_sqlConnection = new SqlConnection(GetConnectionString());

return \_sqlConnection;

}

public void GetEmployees()

{

//\_sqlCommand = new SqlCommand();

//\_sqlCommand.CommandText = "Select \* from Employees";

//\_sqlCommand.Connection = \_sqlConnection;

using (\_sqlConnection = GetConnection())

{

using (\_sqlCommand = new SqlCommand("Select \* from Employees", \_sqlConnection))

{

\_sqlConnection.Open();

\_sqlDataReader = \_sqlCommand.ExecuteReader();

if (\_sqlDataReader.HasRows)

{

while (\_sqlDataReader.Read())

{

Console.WriteLine(\_sqlDataReader["id"]

+ "\t" + \_sqlDataReader["name"]

+ "\t" + \_sqlDataReader["department"]

+ "\t" + \_sqlDataReader["salary"]);

}

}

else

{

Console.WriteLine("No records found");

}

\_sqlConnection.Close();

}

}

}

public string GetEmployeeById(int id)

{

string name = string.Empty;

//\_sqlCommand = new SqlCommand();

//\_sqlCommand.CommandText = "Select \* from Employees";

//\_sqlCommand.Connection = \_sqlConnection;

using (\_sqlConnection = GetConnection())

{

using (\_sqlCommand = new SqlCommand("Select name from Employees" +

" where id=@id", \_sqlConnection))

{

\_sqlCommand.Parameters.AddWithValue("@id", id);

\_sqlConnection.Open();

name = \_sqlCommand.ExecuteScalar().ToString();

\_sqlConnection.Close();

}

}

return name;

}

public int InsertEmployee( string name, string department, int salary )

{

int res = 0;

using (\_sqlConnection = GetConnection())

{

using (\_sqlCommand = new SqlCommand("Insert into Employees( name, department, salary) " +

"values ( @name, @department,@salary)", \_sqlConnection))

{

\_sqlCommand.Parameters.AddWithValue("@name", name);

\_sqlCommand.Parameters.AddWithValue("@department", department);

\_sqlCommand.Parameters.AddWithValue("@salary", salary);

\_sqlConnection.Open();

res = \_sqlCommand.ExecuteNonQuery();

}

}

return res;

}

public int DeleteEmployee(int id)

{

int res = 0;

using (\_sqlConnection = GetConnection())

{

using (\_sqlCommand = new SqlCommand("Delete Employees where id=@id"

, \_sqlConnection))

{

\_sqlCommand.Parameters.AddWithValue("@id", id);

\_sqlConnection.Open();

res = \_sqlCommand.ExecuteNonQuery();

}

}

return res;

}

public int EditEmployee(int id, string name, string department, int salary)

{

int res = 0;

using (\_sqlConnection = GetConnection())

{

using (\_sqlCommand = new SqlCommand("update Employees " +

"set name = @name, department=@department , salary" +

"= @salary where id=@id", \_sqlConnection))

{

\_sqlCommand.Parameters.AddWithValue("@id", id);

\_sqlCommand.Parameters.AddWithValue("@name", name);

\_sqlCommand.Parameters.AddWithValue("@department", department);

\_sqlCommand.Parameters.AddWithValue("@salary", salary);

\_sqlConnection.Open();

res = \_sqlCommand.ExecuteNonQuery();

}

}

return res;

}

}

}

**Calling Stored procedures**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Configuration;

namespace ConsoleApp35

{

class Program

{

static SqlConnection \_sqlConnection;

SqlCommand \_sqlCommand;

static SqlDataReader \_sqlDataReader;

static void Main(string[] args)

{

Program program = new Program();

string ch = "y";

try

{

while (ch == "y")

{

Console.WriteLine("Menu");

Console.WriteLine("1. Display all Records");

Console.WriteLine("2. Insert Record");

Console.WriteLine("3. Delete Record");

Console.WriteLine("4. Edit Record");

Console.WriteLine("5. Search Record");

Console.WriteLine("Enter choice");

int choice = Convert.ToByte(Console.ReadLine());

switch (choice)

{

case 1: program.GetEmployees(); break;

case 2:

{

int salary;

string name, department;

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Department");

department = Console.ReadLine();

Console.WriteLine("ENter Salary");

salary = int.Parse(Console.ReadLine());

int res = program.InsertEmployee(name, department, salary);

if (res == 1)

Console.WriteLine("Record Inserted");

else

Console.WriteLine("Record not inserted");

break;

}

case 3:

{

int id;

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

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

int res = program.DeleteEmployee(id);

if (res == 1)

Console.WriteLine("Record Deleted");

else

Console.WriteLine("Record not deleted");

break;

}

case 4:

{

int id;

string name;

string department;

int salary;

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

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

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Department");

department = Console.ReadLine();

Console.WriteLine("ENter Salary");

salary = int.Parse(Console.ReadLine());

int res = program.EditEmployee(id, name, department, salary);

if (res == 1)

Console.WriteLine("Record Updated");

else

Console.WriteLine("Record not updated");

break;

}

case 5:

{

int id;

string name;

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

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

name = program.GetEmployeeById(id);

if (name == null)

{

Console.WriteLine("Such Record does not exist");

;

}

else

{

Console.WriteLine("Name is " + name);

}

break;

}

}

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

ch = Console.ReadLine();

}

}

catch(Exception ex)

{

Console.WriteLine(ex.Message);

}

}

public string GetConnectionString()

{

return (ConfigurationManager.ConnectionStrings["con"].ToString());

}

public SqlConnection GetConnection()

{

\_sqlConnection = new SqlConnection(GetConnectionString());

return \_sqlConnection;

}

create proc GetEmployees

As

Begin

Select \* from Employees

End

public void GetEmployees()

{

//\_sqlCommand = new SqlCommand();

//\_sqlCommand.CommandText = "Select \* from Employees";

//\_sqlCommand.Connection = \_sqlConnection;

using (\_sqlConnection = GetConnection())

{

**using (\_sqlCommand = new SqlCommand("GetEmployees", \_sqlConnection))**

**{**

**\_sqlCommand.CommandType = System.Data.CommandType.StoredProcedure;**

**\_sqlConnection.Open();**

\_sqlDataReader = \_sqlCommand.ExecuteReader();

if (\_sqlDataReader.HasRows)

{

while (\_sqlDataReader.Read())

{

Console.WriteLine(\_sqlDataReader["id"]

+ "\t" + \_sqlDataReader["name"]

+ "\t" + \_sqlDataReader["department"]

+ "\t" + \_sqlDataReader["salary"]);

}

}

else

{

Console.WriteLine("No records found");

}

\_sqlConnection.Close();

}

}

}

create proc GetEmployeeById (@id int , @name varchar(20) output)

As

Begin

Select @name = name from Employees where id=@id

End

public string GetEmployeeById(int id)

{

string name = string.Empty;

//\_sqlCommand = new SqlCommand();

//\_sqlCommand.CommandText = "Select \* from Employees";

//\_sqlCommand.Connection = \_sqlConnection;

using (\_sqlConnection = GetConnection())

{

**using (\_sqlCommand = new SqlCommand("GetEmployeeById", \_sqlConnection))**

**{**

**\_sqlCommand.CommandType = System.Data.CommandType.StoredProcedure;**

**\_sqlCommand.Parameters.AddWithValue("@id", id);**

**SqlParameter para = new SqlParameter("@name" , System.Data.SqlDbType.VarChar, size:20);**

**para.Direction = System.Data.ParameterDirection.Output;**

**\_sqlCommand.Parameters.Add(para);**

**\_sqlConnection.Open();**

**\_sqlCommand.ExecuteNonQuery();**

**name = \_sqlCommand.Parameters["@name"].Value.ToString();**

\_sqlConnection.Close();

}

}

return name;

}

create proc InsertEmployee(@name varchar(20), @department varchar(20), @salary int)

as

begin

insert into enployee(name, department, salary) values(@name , @department, @salary)

return 1

End

public int InsertEmployee( string name, string department, int salary )

{

int res = 0;

using (\_sqlConnection = GetConnection())

{

**using (\_sqlCommand = new SqlCommand("InsertEmployee", \_sqlConnection))**

**{**

**\_sqlCommand.CommandType = System.Data.CommandType.StoredProcedure;**

**\_sqlCommand.Parameters.AddWithValue("@name", name);**

**\_sqlCommand.Parameters.AddWithValue("@department", department);**

**\_sqlCommand.Parameters.AddWithValue("@salary", salary);**

**SqlParameter para = new SqlParameter("@res", System.Data.SqlDbType.Int);**

**\_sqlCommand.Parameters.Add(para);**

**\_sqlConnection.Open();**

**\_sqlCommand.ExecuteNonQuery();**

**res = Convert.ToByte(\_sqlCommand.Parameters["@res"].Value);**

**}**

**}**

return res;

}

Similarly you will do for Edit & Delete

public int DeleteEmployee(int id)

{

int res = 0;

using (\_sqlConnection = GetConnection())

{

using (\_sqlCommand = new SqlCommand("Delete Employees where id=@id"

, \_sqlConnection))

{

\_sqlCommand.Parameters.AddWithValue("@id", id);

\_sqlConnection.Open();

res = \_sqlCommand.ExecuteNonQuery();

}

}

return res;

}

public int EditEmployee(int id, string name, string department, int salary)

{

int res = 0;

using (\_sqlConnection = GetConnection())

{

using (\_sqlCommand = new SqlCommand("update Employees " +

"set name = @name, department=@department , salary" +

"= @salary where id=@id", \_sqlConnection))

{

\_sqlCommand.Parameters.AddWithValue("@id", id);

\_sqlCommand.Parameters.AddWithValue("@name", name);

\_sqlCommand.Parameters.AddWithValue("@department", department);

\_sqlCommand.Parameters.AddWithValue("@salary", salary);

\_sqlConnection.Open();

res = \_sqlCommand.ExecuteNonQuery();

}

}

return res;

}

}

}