EXECUTE RAW QUERIES

**Writing queries for entity**

**Dbset.SqlQuery()**Returns entity instances

* Tracked by context(as if they were returned by LINQ query).

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DatabaseFirstConsole

{

class RawQueries

{

public static void Main(string[] args)

{

MyDbEntities db = new MyDbEntities();

string sql = "select \* from DEPTLOC where CITYID = 200";

IEnumerable<DEPTLOC> loclist = db.DEPTLOCs.SqlQuery(sql);

foreach (DEPTLOC aobj in loclist)

{

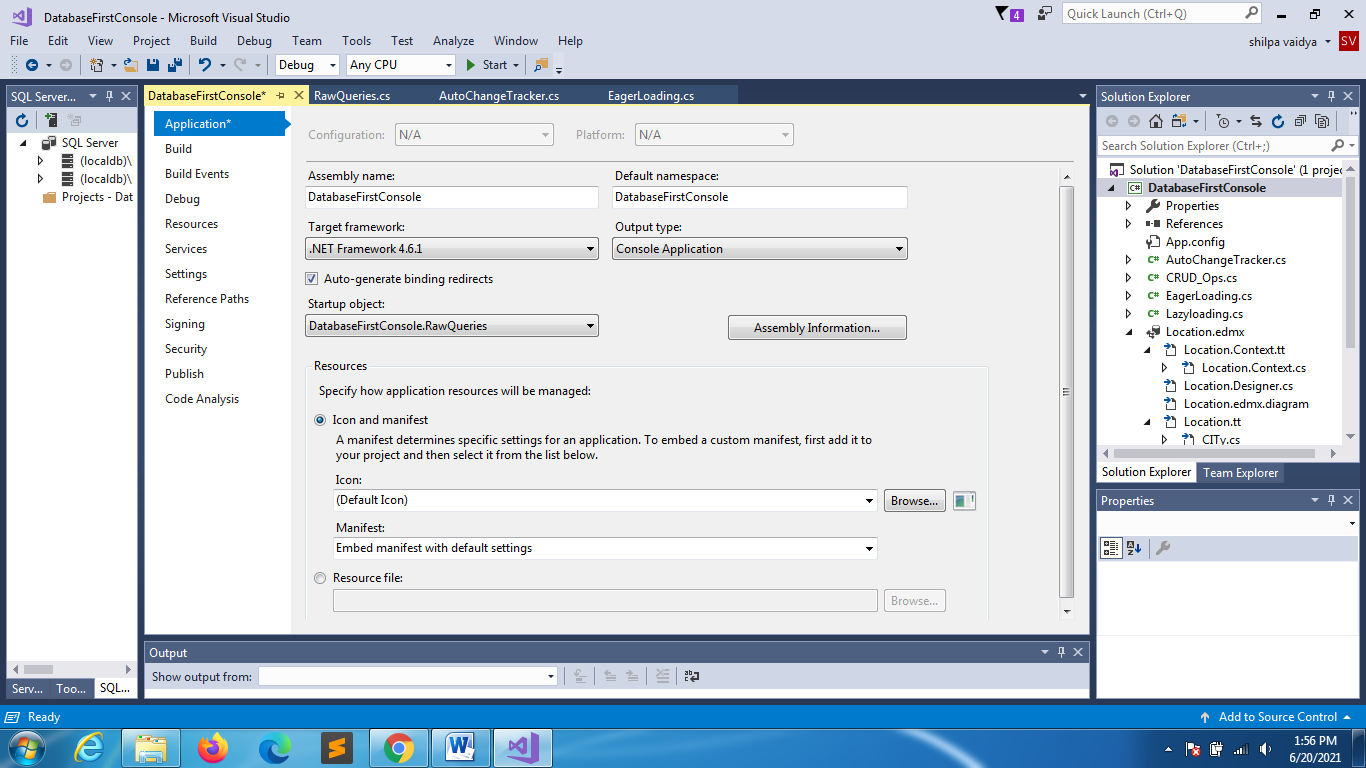
Console.WriteLine(aobj.AREANAME);

}

}

}

}



**If you want to pass parameter**

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DatabaseFirstConsole

{

class RawQueries

{

public static void Main(string[] args)

{

MyDbEntities db = new MyDbEntities();

// IEnumerable<DEPTLOC> loclist = db.DEPTLOCs.SqlQuery(sql);

SqlParameter p1 = new SqlParameter();

p1.ParameterName = "@id";

p1.Value = 200;

p1.SqlDbType = System.Data.SqlDbType.Int;

string sql = "select \* from DEPTLOC where CITYID = @id";

IEnumerable<DEPTLOC> loclist = db.DEPTLOCs.SqlQuery(sql,p1);

foreach (DEPTLOC aobj in loclist)

{

Console.WriteLine(aobj.AREANAME);

}

}

}

}

SQL QUERIES FOR NONENTITY TYPE

Create a class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DatabaseFirstConsole

{

class MyDeptloc

{

public string AreaName { get; set; }

public long Phone { get; set; }

}

}

Rawqueries.cs(modify)

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DatabaseFirstConsole

{

class RawQueries

{

public static void Main(string[] args)

{

MyDbEntities db = new MyDbEntities();

// IEnumerable<DEPTLOC> loclist = db.DEPTLOCs.SqlQuery(sql);

SqlParameter p1 = new SqlParameter();

p1.ParameterName = "@id";

p1.Value = 200;

p1.SqlDbType = System.Data.SqlDbType.Int;

string sql = "select \* from DEPTLOC where CITYID = @id";

IEnumerable<DEPTLOC> loclist = db.DEPTLOCs.SqlQuery(sql,p1);

Console.WriteLine("entityquery");

foreach (DEPTLOC aobj in loclist)

{

Console.WriteLine(aobj.AREANAME);

}

//nonentityquery

Console.WriteLine("nonentityquery");

var myLoclist = db.Database.SqlQuery<MyDeptloc>("select Areaname,phone from Deptloc");

foreach(MyDeptloc md in myLoclist)

{

Console.WriteLine("{0}\t{1}",md.AreaName,md.Phone);

}

}

}

}

**SENDING RAWQUERIES FOR EXECUTE COMMAND(NONQUERY)**

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DatabaseFirstConsole

{

class RawQueries

{

public static void Main(string[] args)

{

MyDbEntities db = new MyDbEntities();

// IEnumerable<DEPTLOC> loclist = db.DEPTLOCs.SqlQuery(sql);

SqlParameter p1 = new SqlParameter();

p1.ParameterName = "@id";

p1.Value = 200;

p1.SqlDbType = System.Data.SqlDbType.Int;

string sql = "select \* from DEPTLOC where CITYID = @id";

IEnumerable<DEPTLOC> loclist = db.DEPTLOCs.SqlQuery(sql,p1);

Console.WriteLine("entityquery");

foreach (DEPTLOC aobj in loclist)

{

Console.WriteLine(aobj.AREANAME);

}

//nonentityquery

Console.WriteLine("nonentityquery");

var myLoclist = db.Database.SqlQuery<MyDeptloc>("select Areaname,phone from Deptloc");

foreach(MyDeptloc md in myLoclist)

{

Console.WriteLine("{0}\t{1}",md.AreaName,md.Phone);

}

//updateentity

string sql2 = "update Deptloc set Phone=9876678987 where Areaid=34";

int res = db.Database.ExecuteSqlCommand(sql2);

Console.WriteLine("send rawcommand update");

if(res==1)

Console.WriteLine("Record updated");

else

Console.WriteLine("Error in executing query");

}

}

}