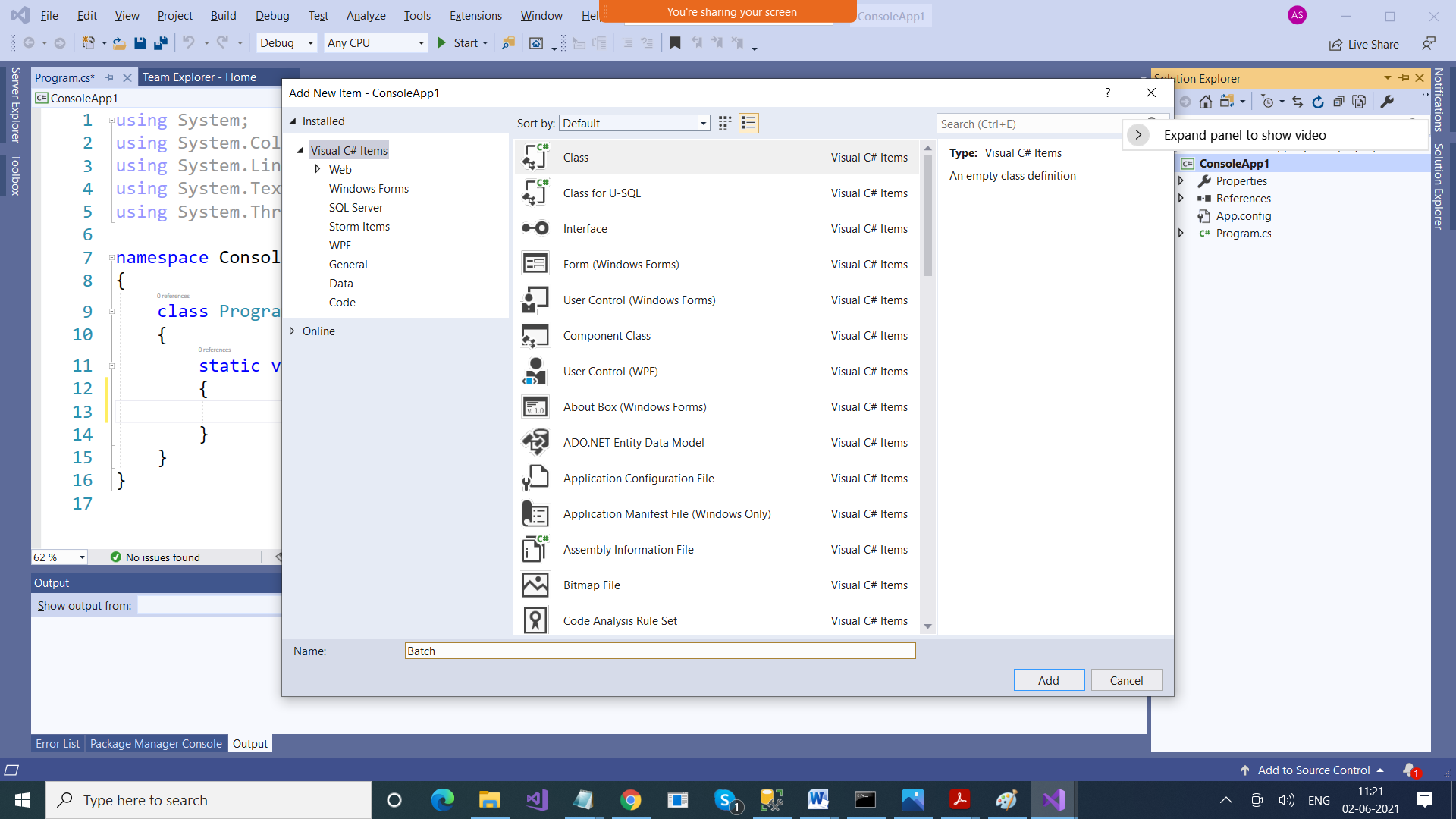
Code First Approach

From Class, you get table

Step 1: Create a class with properties



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Batch

{

public int Id { get; set; }

public string Name { get; set; }

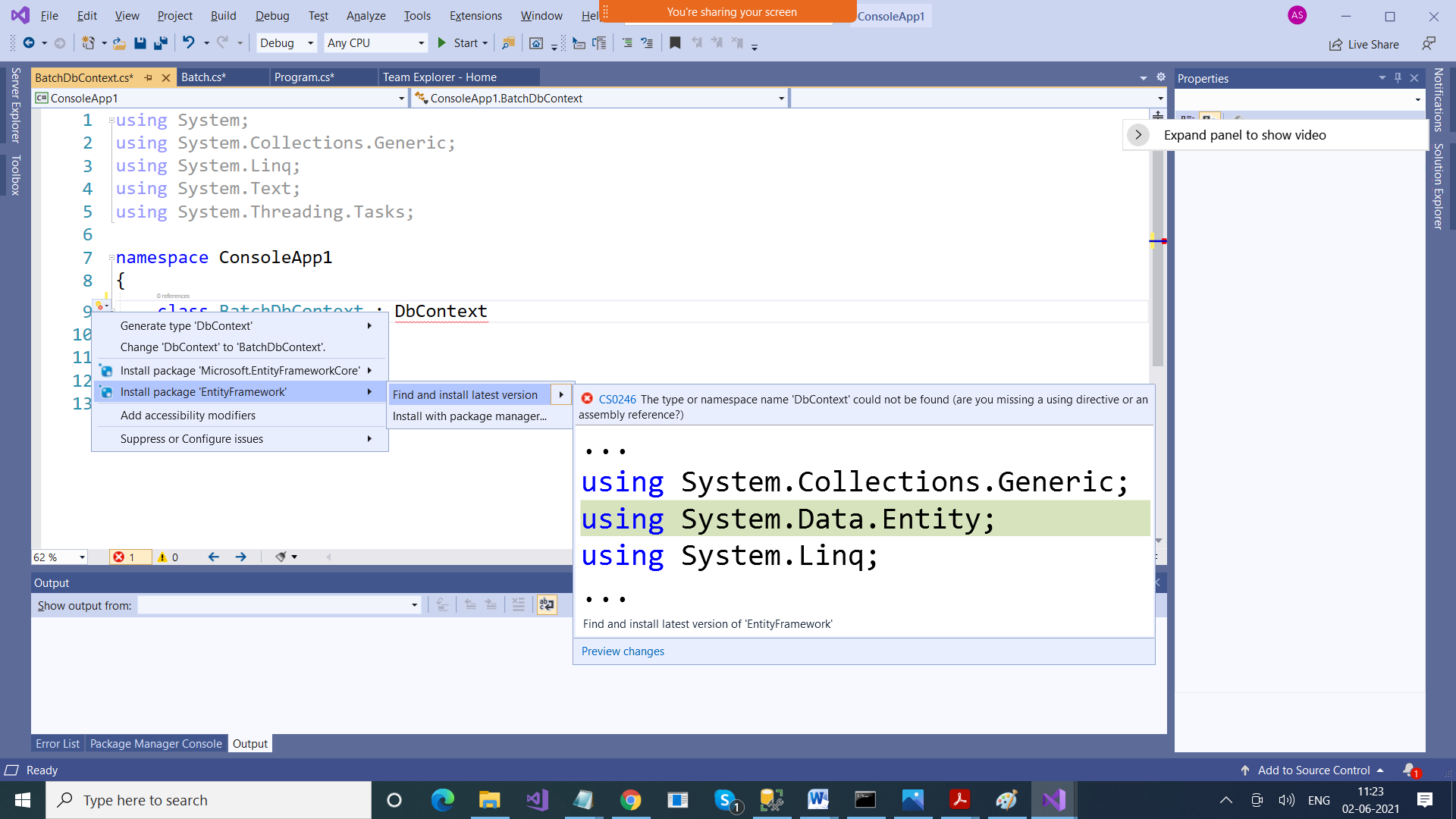
public string Course { get; set; }

public int Count { get; set; }

}

}

Step 2: Add one more class which should inherit from DbContext Class



using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class BatchDbContext : DbContext

{

public DbSet<Batch> batches{ get; set; }

}

}

Step 3: Add a connectionString in app.config file with same name as this class which inherits from DbContext Class

<connectionStrings>

<add name="BatchDbContext" connectionString="data source=LAPTOP-53S2KQS8; initial catalog=practicedb; integrated security=true" providerName="System.Data.SqlClient"/>

</connectionStrings>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

BatchDbContext context = new BatchDbContext();

List<Batch> batches = context.batches.ToList();

if (batches.Count > 0)

{

foreach (Batch temp in batches)

Console.WriteLine(temp.Id + " " + temp.Name + " " + temp.Course + " " + temp.Course);

}

else

Console.WriteLine("No records");

// Insertion

Batch batch = new Batch();

Console.WriteLine("Enter Id");

batch.Id = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Batch Name");

batch.Name = Console.ReadLine();

Console.WriteLine("Enter Course");

batch.Course = Console.ReadLine();

Console.WriteLine("Enter Student Count");

batch.Count = int.Parse(Console.ReadLine());

context.batches.Add(batch);

context.SaveChanges();

Console.WriteLine("Record Inserted");

// Updation

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

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

batch = context.batches.Find(id);

if (batch != null)

{

Console.WriteLine("Enter Batch Name");

batch.Name = Console.ReadLine();

Console.WriteLine("Enter Course");

batch.Course = Console.ReadLine();

Console.WriteLine("Enter Student Count");

batch.Count = int.Parse(Console.ReadLine());

foreach (Batch temp in batches)

{

if (temp.Id == id)

{

temp.Name = batch.Name;

temp.Count = batch.Count;

temp.Course = batch.Course;

}

}

context.SaveChanges();

Console.WriteLine("Record Updated");

}

context.batches.Add(batch);

context.SaveChanges();

Console.WriteLine("Record Inserted");

// Deletion

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

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

batch = context.batches.Find(id);

if (batch != null)

{

context.batches.Remove(batch);

context.SaveChanges();

Console.WriteLine("Record Updated");

}

}

}

}