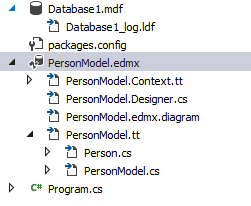
# Intro

## Database first

Add model by database – add new item->data->ADO data model-> generate from database



Model

namespace DataBase\_First

{

using System;

using System.Collections.Generic;

public partial class Person

{

public int Id { get; set; }

public string Name { get; set; }

public Nullable<int> Age { get; set; }

}

}

Context

namespace DataBase\_First

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class PersonEntities : DbContext

{

public PersonEntities()

: base("name=PersonEntities")

{

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public DbSet<Person> People { get; set; }

}

}

Program.cs

class Program

{

static void Main()

{

var context = new PersonEntities();

foreach (var person in context.People)

{

Console.WriteLine("{0}: {1,-6} - {2} years",person.Id,person.Name,person.Age);

}

}

}

## Adventure work

namespace AW\_DB\_First

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class AdventureWorksLT2012\_DataEntities : DbContext

{

public AdventureWorksLT2012\_DataEntities()

: base("name=AdventureWorksLT2012\_DataEntities")

{

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public DbSet<Address> Addresses { get; set; }

public DbSet<Customer> Customers { get; set; }

public DbSet<CustomerAddress> CustomerAddresses { get; set; }

public DbSet<SalesOrderDetail> SalesOrderDetails { get; set; }

public DbSet<SalesOrderHeader> SalesOrderHeaders { get; set; }

}

}

static void Main()

{

using (var context = new AdventureWorksLT2012\_DataEntities())

{

var query = from customer in context.Customers

where customer.CustomerAddresses.Any()

orderby customer.FirstName

select new { customer.FirstName, customer.CustomerAddresses };

foreach (var customer in query.Take(10))

{

Console.WriteLine(customer.FirstName);

foreach (var customerAddress in customer.CustomerAddresses)

{

Console.WriteLine("\t" + customerAddress.Address.StateProvince);

}

}

}

}

}