# L52

## Insert

var personas=new Persona(){id=2};

db.Persona.Attach(persona);

db.Direccion.Add(new Direccion()){Calle=”Calle ejemplo”, Persona=persona}

db.SaveChanges();

## Lazy loading (virtual)

Poñendo virtual na clase: public virtual List<Direccion>Direcciones { get; set; }

-----------------------------------------------------------------------------------------------------

var persona=db.Persona. FirstOrDefault(x=>x.Id==2);

var direcciones=persona.Direcciones;

#### L57 Lazy Loading

En este caso as propiedades relacionades estarán a nosa disposición cando intentemos accede e alas.

**1-Traer unha persoa cas dirección**

Var persona=db.Persona.FirstOrDefault();

Var PrimeraDireccion=persona.Direcciones[0];

**2-Trae todas las personas con sus direcciones**

Var presonasConDirecciones=db.Personas.ToList();

Var DireccionDeLaSegundaPersona=personasConDirecciones[1].Direcciones[0];

**3-Segundo nivel**

Var subCalle=persona.Direcciones[0].SubDireccion[0]SubCalle;

**PROBLEMA SERIALIZACIÓN:**

**Se pueden formar un bucle infinito ver** [**video L57**](L57-Lazy%20Loading.mp4)

Var persona JSON=Newtonsoft.Json.JsonConvert.SerializeObject(persona); **ERROR**

**PROBLEMA n+1**

**Ver** [**video L58**](L58-Error%20n+1%20Lazy%20loading.mp4)

|  |  |
| --- | --- |
|  | Codigo extremadamente lento porque al acceder 15000 veces a la dirección lanza 15000 consultas una a una más la de personas.  Se corrige con Eager loading utilizando include puesto que así no realiza 15001 consultas sino que lo hace en una. |

## Eager loading

Eliminamos o virutal.

Faino con un solo query polo que e mais rápido q Lazy loading

var persona=db.Persona.Include(“Direcciones”). FirstOrDefault(x=>x.Id==2);

var direcciones=persona.Direcciones;

### L56 Eager Loading

Partimos dunha tabla persoas que ten Direccions e esta ten subdireccions

**1-Include con Lambda**

var personasInclude=db.Pesonas.Include(x=>x.Direcciones).FirstOrDefault();

var primerDireccionInclude=personas.Direcciones[0];

**2-Include con string**

var personasConDirecciones= db.Persona.Include(“Direcciones”).ToList();

var DireccionDeLaSegundaPersona= personasConDirecciones[1].Direccion[0];

**3-Include Segundo nivel**

var presonasConDireecionesConSub=db.Personas.Include(

x=>x.Direcciones.Select(y=> y.SubDireccion)).FirstOrDefault();

var subCalle=personasConDireccionesConSub.Direcciones[0].SubDireccion[0].SubCalle;

return View(db.Persona.ToList());

# L53- Join- GroupJoin

Imaginémonos una relación entre personas y direcciones

OLLO: Ver video lección anterior L52 (52-Llaves foraneas y propiedades de navegacion (lazy loading-virtual y Eager loading-Include).mp4)

## Join

Indicamos as claves para o join

**Traemos unha dirección ca persoa correspondinte:**

var personaDireccion = db.Direccion.Join(db.Persona,dir=>dir.IdPersona,

per=>per.Id,(dir,per)=>new {dir,per}). FirstOrDefault(x=>x.dir.CodigoDireccion==1);

O que traemos. En este caso traemos todo as columnas, dir e per teñen que ir en orden co de arriba db.Direccion.Join (db.Persona

Podemos poñer Where en lugar FirstOrDefault.

Where(x=>dir.CodigoDireccion==1)

## GroupJoin

GroupJoin ejecuta un left join.

**Traer Persona cas Suas filas de direccions:**

var persona1ConSusDirecciones=db.Persona.GroupJoin(db.Direccion,per=>per.Id, dir=>dir.IdPersona, (per,dir)=>new{per,dir}).FirstOrDefault(x=>per.==1);

**Traer todas as persas cas suas direccion:**

var persona1ConSusDirecciones=db.Persona.GroupJoin(db.Direccion,per=>per.Id, dir=>dir.IdPersona, (per,dir)=>new{per,dir}).ToList();

return View(db.Pesona.ToList());

# L54-ver consultas antes de lanzalas

Para ver la consulta antes de ejecutala podemolo facer indicando ***.ToString()*** donde iría ***.ToList().***

***Ejemplo con group by:***

var personasSexoMasculino=db.Persona.GroupBy(x=>x.Sexo).ToString();

# L55-Query arbitrario

**Ejemplo 1:**

var personas=db.Persona.SqlQuery(“SELECT \* FROM dbo.Personas”).ToList;

**Ejemplo 2:**

Pasamos parámetros

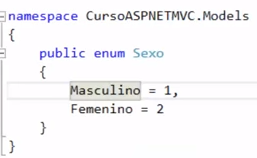
var direccion=db.Database.sqlQuery<Direccion>(

@“SELECT \* FROM dbo.Direcciones

WHERE CodigoDireccion=@Id”,new SqlParameter(“@Id”,1)).FirstOrDefault();

**Ejemplo 3:**

Queremos extraer as estadísticas por sexo.

Creamos unha clase estadistica :

namespace EfCodeFirst.Controllers

{

public class BlogPostController : Controller

{

private BlogContext db = new BlogContext();

class Estadistica

{

public Sexo Sexo { get; set; }

public int Cantidad { get; set; }

}

public ActionResult Index()

Utilizase db.Database porque se vai a usar un tipo que nn está mapeado como unha tabla da bbdd

{

var estadisticaDeSexo=db.Database.SqlQuery<Estadistica>(

@"SELECT Sexo,count(\*) as Cantidad FROM dbo.Personas GROUP BY Sexo

).ToList();