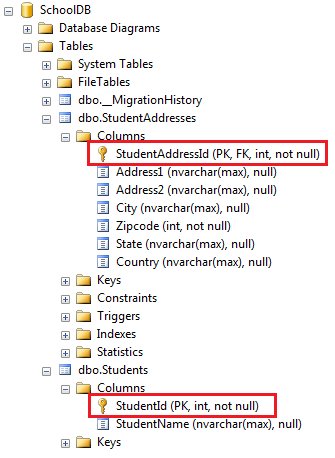
# Entity Framework 6 Code-First

## One-to-One

La situazione e quella mostrata nella figura seguente



public class Student

{

public int StudentId { get; set; }

public string StudentName { get; set; }

public virtual StudentAddress Address { get; set; }

}

public class StudentAddress

{

[ForeignKey("Student")]

public int StudentAddressId { get; set; }

public string Address1 { get; set; }

public string Address2 { get; set; }

public string City { get; set; }

public int Zipcode { get; set; }

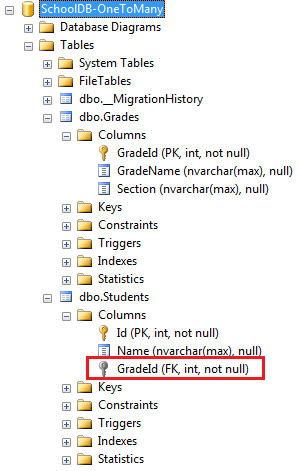
public string State { get; set; }

public string Country { get; set; }

public virtual Student Student { get; set; }

}

## One-to-Many



public class Student

{

public int Id { get; set; }

public string Name { get; set; }

**public int GradeId { get; set; }**

**public Grade Grade { get; set; }**

}

public class Grade

{

public int GradeId { get; set; }

public string GradeName { get; set; }

**public ICollection<Student> Student { get; set; }**

}

## Relazione many-to-many

Supponiamo di avere un’entità Gruppo di attività (ActivityGroup) e un’entità Attività (Activities) per la quale possono essere attribuite n gruppi di attività.

In questo scenario è possibile utilizzare una struttura many-to-many che, in EF6 code-first, si realizza come segue:

public class ActivityGroup : BaseEntityMultiTenant

{

[Required]

[MaxLength(500)]

public string Description { get; set; }

public List<Activities> Activities { get; set; }

}

e

public class Activities : BaseEntityMultiTenant

{

[Required]

[MaxLength(500)]

public string Description { get; set; }

[Range(0.01, 9999.99)]

public decimal Price { get; set; }

public decimal PercDiscounted { get; set; }

[Range(0.01, 9999.99)]

public decimal PriceDiscounted { get; set; }

public List<ActivityGroup> ActivityGroups { get; set; }

public Activities()

{

ActivityGroups = new List<ActivityGroup>();

}

}

Nel db questa struttura si concretizza nelle due entità ActivityGroup e Activities ma da sole non bastano occorre una terza entità, costruita in automatico da EF6, che gestisca la relazione ed è così fatta :



Questa entità non è presente nel modello EF6 ma solo nel DB.

### Come si utilizza una struttura many-to-many

La proprietà di navigazione ActivityGroups consente di avere a disposizione tutti gli activityGroup indicati per l’attività.

Di seguito l’utilizzo per rimappare le entità in un dto :

result = (from item in entityList

select new jActivity()

{

Id = item.Id,

Description = item.Description,

Price = item.Price,

PercDiscounted = item.PercDiscounted,

PriceDiscounted = item.PriceDiscounted,

ActivityGroup = (from d in item.ActivityGroups

select new jActivityGroup

{

Id = d.Id,

CompanyId = d.CompanyId,

Description = d.Description,

}).ToList()

}).ToList();

