|  |
| --- |
| EMPLOYEE  0..1  1 |
| EmployeeId |
| EmpName |
| Salary |
| Email |
| PrimaryContact |
| SecondaryContact |
| EMPLOYEEADDRESS |
| EmployeeId |
| AddressLine1 |
| AddressLine2 |
| City |
| Zipcode |
|  |

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations.Schema;

namespace FluentAPI

{

class MyDbContext:DbContext

{

public MyDbContext():base("name:fluent")

{

}

public virtual DbSet<Department> Departments { get; set; }

public virtual DbSet<Employee> Employees { get; set; }

public virtual DbSet<EmployeeAddress> EmployeeAddresses { get; set; }

public DbSet<Project> Projects { get; set; }

public DbSet<Team> Teams { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

//Totable -> configure/set table name in sql server

modelBuilder.Entity<Department>().ToTable("Dept");

//HasKey (cause usually primary key will be either

//ID of DepartmentId in the class if it isnt so then we

//have to explicitly mention here

modelBuilder.Entity<Department>().HasKey(d=>d.DeptId);

//Property configuration "Employee" entity

// we dont want auto increment - >DatabaseGeneratedOption.None

modelBuilder.Entity<Employee>().Property(e=>e.EmployeeId).HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);

//EmpNAme - change name to Ename

modelBuilder.Entity<Employee>().Property(e => e.EmpName).HasColumnName("Ename")

.IsRequired()

.HasMaxLength(50)

.HasColumnType("varchar");

//Primary Contact

modelBuilder.Entity<Employee>().Property(e => e.PrimaryContact).IsRequired()

.HasMaxLength(10);

//employeeId is primary key in Employee Address

modelBuilder.Entity<EmployeeAddress>().HasKey(ea => ea.EmployeeId);

//one to one Employee-->EmployeeAddress

modelBuilder.Entity<Employee>().HasOptional(e => e.EmployeeAddress)

.WithRequired(a => a.Employee);

}

}

}

ONE TO MANY

**EMPLOYEE**

EMPLOYEEID

DEPTID

EMPNAME

SALARY

EMAIL

PRIMARY CONTACT

SECONDARY CONTACT

**DEPARTMENT**

DEPTID

DEPTNAME

1 1…M

**EMPLOYEE**

EMPLOYEEID

TEAMID

EMPNAME

SALARY

EMAIL

PRIMARY CONTACT

SECONDARY CONTACT

**TEAM**

TEAMID

TEAMNAME

SIZE

1 0…M

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Employee

{

public int EmployeeId { get; set; }

public string EmpName { get; set; }

public float Salary { get; set; }

public string Email { get; set; }

public string PrimaryContact { get; set; }

public string SecondaryContact { get; set; }

public int DeptId { get; set; }

public int? TeamId { get; set; }

//navigatio properties

public virtual Department Department { get; set; }

public virtual EmployeeAddress EmployeeAddress { get; set; }

//1 emp 1 team

public virtual Team Team { get; set; }

public virtual ICollection<Project> Projects { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations.Schema;

namespace FluentAPI

{

class MyDbContext:DbContext

{

public MyDbContext():base("name:fluent")

{

}

public virtual DbSet<Department> Departments { get; set; }

public virtual DbSet<Employee> Employees { get; set; }

public virtual DbSet<EmployeeAddress> EmployeeAddresses { get; set; }

public DbSet<Project> Projects { get; set; }

public DbSet<Team> Teams { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

//Totable -> configure/set table name in sql server

modelBuilder.Entity<Department>().ToTable("Dept");

//HasKey (cause usually primary key will be either

//ID of DepartmentId in the class if it isnt so then we

//have to explicitly mention here

modelBuilder.Entity<Department>().HasKey(d=>d.DeptId);

//Property configuration "Employee" entity

// we dont want auto increment - >DatabaseGeneratedOption.None

modelBuilder.Entity<Employee>().Property(e=>e.EmployeeId).HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);

//EmpNAme - change name to Ename

modelBuilder.Entity<Employee>().Property(e => e.EmpName).HasColumnName("Ename")

.IsRequired()

.HasMaxLength(50)

.HasColumnType("varchar");

//Primary Contact

modelBuilder.Entity<Employee>().Property(e => e.PrimaryContact).IsRequired()

.HasMaxLength(10);

//employeeId is primary key in Employee Address

modelBuilder.Entity<EmployeeAddress>().HasKey(ea => ea.EmployeeId);

//one to one Employee-->EmployeeAddress

modelBuilder.Entity<Employee>().HasOptional(e => e.EmployeeAddress)

.WithRequired(a => a.Employee);

//ONE-TO -MANY RELATIONSHIP DEPT--->EMPLOYEE

modelBuilder.Entity<Department>().HasMany(d => d.Employees)

.WithRequired(e => e.Department)

.HasForeignKey(e => e.DeptId);

//one to many relationship bet team & employee

modelBuilder.Entity<Team>().HasMany(t => t.Employees)

.WithOptional(e => e.Team)

.HasForeignKey(e => e.TeamId);

}

}

}

MANY TO MANY

**PROJECT**

PROJECTID

PROJECTNAME

PROGLANGUAGE

DATABASE

**EMPLOYEE**

EMPLOYEEID

EMPNAME

SALARY

EMAIL

PRIMARY CONTACT

SECONDARY CONTACT

\* \*

\*

**PROJECTOFEMPLOYEE**

EMPLOYEEID

PROJECTID

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations.Schema;

namespace FluentAPI

{

class MyDbContext:DbContext

{

public MyDbContext():base("name:fluent")

{

}

public virtual DbSet<Department> Departments { get; set; }

public virtual DbSet<Employee> Employees { get; set; }

public virtual DbSet<EmployeeAddress> EmployeeAddresses { get; set; }

public DbSet<Project> Projects { get; set; }

public DbSet<Team> Teams { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

//Totable -> configure/set table name in sql server

modelBuilder.Entity<Department>().ToTable("Dept");

//HasKey (cause usually primary key will be either

//ID of DepartmentId in the class if it isnt so then we

//have to explicitly mention here

modelBuilder.Entity<Department>().HasKey(d=>d.DeptId);

//Property configuration "Employee" entity

// we dont want auto increment - >DatabaseGeneratedOption.None

modelBuilder.Entity<Employee>().Property(e=>e.EmployeeId).HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);

//EmpNAme - change name to Ename

modelBuilder.Entity<Employee>().Property(e => e.EmpName).HasColumnName("Ename")

.IsRequired()

.HasMaxLength(50)

. HasColumnType("varchar");

//Primary Contact

modelBuilder.Entity<Employee>().Property(e => e.PrimaryContact).IsRequired()

.HasMaxLength(10);

//employeeId is primary key in Employee Address

modelBuilder.Entity<EmployeeAddress>().HasKey(ea => ea.EmployeeId);

//one to one Employee-->EmployeeAddress

modelBuilder.Entity<Employee>().HasOptional(e => e.EmployeeAddress)

.WithRequired(a => a.Employee);

//ONE-TO -MANY RELATIONSHIP DEPT--->EMPLOYEE

modelBuilder.Entity<Department>().HasMany(d => d.Employees)

.WithRequired(e => e.Department)

.HasForeignKey(e => e.DeptId);

//one to many relationship bet team & employee

modelBuilder.Entity<Team>().HasMany(t => t.Employees)

.WithOptional(e => e.Team)

.HasForeignKey(e => e.TeamId);

//MANY TO MANY RELATIONSHIP (EMPLOYEE --- PROJECT)

modelBuilder.Entity<Employee>().HasMany(e => e.Projects).

WithMany(p => p.Employees)

.Map(ep =>

{

ep.MapLeftKey("EmployeeId");

ep.MapRightKey("ProjectId");

ep.ToTable("ProjectsOfEmployee");

});

}

}

}

PROGRAM.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Program

{

static void Main(string[] args)

{

try

{

MyDbContext db = new MyDbContext();

Department d = new Department();

d.DeptId = 1;

d.DeptName = "Development";

db.Departments.Add(d);

db.SaveChanges();

Console.WriteLine("DB CREATED");

}

catch(Exception e)

{

Console.WriteLine("ERROR = " + e.ToString());

}

}

}

}