DELETE CASCADE

ALLOWS DELETING A ROW TRIGGERS THE DELETION OF RELATED ROWS AUTOMATICALLY.

EMPLOYEE

EMPLOYEEID

DEPTID

EMPNAME

SALARY

EMAIL

PRIMARY CONTACT

SECONDARY CONTACT

DEPARTMENT

DEPTID

DEPTNAME

DELETE A CHILD RECORDS AUTOMATICALLY WHEN ITS PARENT IS EXPLICITLY DELETED VIA DBCONTEXT

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)

.HasColumnName("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)

.WillCascadeOnDelete(false);

//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");

});

//

}

}

}