الكلاسات (Entities)  
using System.Collections.Generic;

namespace UniversityApp.Models

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

public string MgrName { get; set; }

// Relations

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

public ICollection<Teacher> Teachers { get; set; }

public ICollection<Course> Courses { get; set; }

}

public class Teacher

{

public int Id { get; set; }

public string Name { get; set; }

public decimal Salary { get; set; }

public string Address { get; set; }

// FK

public int DepartmentId { get; set; }

public Department Department { get; set; }

// Relation with Course

public ICollection<Course> Courses { get; set; }

}

public class Student

{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

// FK

public int DepartmentId { get; set; }

public Department Department { get; set; }

// Relation with Course (Many-to-Many through StuCrsRes)

public ICollection<StuCrsRes> StuCrsRes { get; set; }

}

public class Course

{

public int Id { get; set; }

public string Name { get; set; }

public string Degree { get; set; }

public int MinDegree { get; set; }

// FK

public int DepartmentId { get; set; }

public Department Department { get; set; }

// Relation with Teacher

public ICollection<Teacher> Teachers { get; set; }

// Relation with Student (Many-to-Many through StuCrsRes)

public ICollection<StuCrsRes> StuCrsRes { get; set; }

}

public class StuCrsRes

{

// Composite Key (StudentId + CourseId)

public int StudentId { get; set; }

public Student Student { get; set; }

public int CourseId { get; set; }

public Course Course { get; set; }

public int Grade { get; set; }

}

}  
  
  
  
DbContext  
using Microsoft.EntityFrameworkCore;

namespace UniversityApp.Models

{

public class UniversityContext : DbContext

{

public UniversityContext(DbContextOptions<UniversityContext> options)

: base(options)

{

}

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

public DbSet<Teacher> Teachers { get; set; }

public DbSet<Student> Students { get; set; }

public DbSet<Course> Courses { get; set; }

public DbSet<StuCrsRes> StuCrsRes { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

// Composite Key for StuCrsRes

modelBuilder.Entity<StuCrsRes>()

.HasKey(sc => new { sc.StudentId, sc.CourseId });

// One-to-Many: Department -> Students

modelBuilder.Entity<Student>()

.HasOne(s => s.Department)

.WithMany(d => d.Students)

.HasForeignKey(s => s.DepartmentId);

// One-to-Many: Department -> Teachers

modelBuilder.Entity<Teacher>()

.HasOne(t => t.Department)

.WithMany(d => d.Teachers)

.HasForeignKey(t => t.DepartmentId);

// One-to-Many: Department -> Courses

modelBuilder.Entity<Course>()

.HasOne(c => c.Department)

.WithMany(d => d.Courses)

.HasForeignKey(c => c.DepartmentId);

// Many-to-Many: Student <-> Course (through StuCrsRes)

modelBuilder.Entity<StuCrsRes>()

.HasOne(sc => sc.Student)

.WithMany(s => s.StuCrsRes)

.HasForeignKey(sc => sc.StudentId);

modelBuilder.Entity<StuCrsRes>()

.HasOne(sc => sc.Course)

.WithMany(c => c.StuCrsRes)

.HasForeignKey(sc => sc.CourseId);

}

}

}  
  
builder.Services.AddDbContext<UniversityContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));  
  
   
"ConnectionStrings": {

"DefaultConnection": "Server=.;Database=UniversityDB;Trusted\_Connection=True;TrustServerCertificate=True;"

}  
  
  
---------------------------------------------------------------------------------------------------------------------

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using UniversityApp.Models;

using System.Linq;

namespace UniversityApp.Controllers

{

public class StudentController : Controller

{

private readonly UniversityContext \_context;

public StudentController(UniversityContext context)

{

\_context = context;

}

// /student/ShowAll

public IActionResult ShowAll()

{

var students = \_context.Students

.Include(s => s.Department) // علشان يجيب بيانات القسم مع الطالب

.ToList();

return View(students);

}

// /student/ShowDetails?id=3

public IActionResult ShowDetails(int id)

{

var student = \_context.Students

.Include(s => s.Department)

.Include(s => s.StuCrsRes)

.ThenInclude(sc => sc.Course) // يجيب الكورسات اللي واخدها الطالب

.FirstOrDefault(s => s.Id == id);

if (student == null)

return NotFound();

return View(student);

}

}

}  
  
  
@model IEnumerable<UniversityApp.Models.Student>

<h2>All Students</h2>

<table class="table table-bordered">

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>Age</th>

<th>Department</th>

<th>Details</th>

</tr>

</thead>

<tbody>

@foreach (var student in Model)

{

<tr>

<td>@student.Id</td>

<td>@student.Name</td>

<td>@student.Age</td>

<td>@student.Department?.Name</td>

<td>

<a href="/Student/ShowDetails?id=@student.Id">View</a>

</td>

</tr>

}

</tbody>

</table>  
  
  
@model UniversityApp.Models.Student

<h2>Student Details</h2>

<p><strong>Id:</strong> @Model.Id</p>

<p><strong>Name:</strong> @Model.Name</p>

<p><strong>Age:</strong> @Model.Age</p>

<p><strong>Department:</strong> @Model.Department?.Name</p>

<h3>Courses & Grades</h3>

<table class="table table-striped">

<thead>

<tr>

<th>Course</th>

<th>Grade</th>

</tr>

</thead>

<tbody>

@foreach (var crs in Model.StuCrsRes)

{

<tr>

<td>@crs.Course.Name</td>

<td>@crs.Grade</td>

</tr>

}

</tbody>

</table>

<a href="/Student/ShowAll">Back to All Students</a>  
  
  
---------------------------------------------------------------------------------------------------------------------

A paper with text on it

AI-generated content may be incorrect.