**LINQ (Language-Integrated Query)**

1. Linq Example – Use a method to pull one field from the database table Course and return it to the Student model. Calculate totalfees in Student model and display it on view.
2. The Linq method called PullCourseFees() uses linq function to access the table Courses from the Student table and pull the course fees based on the CourseId of both tables being equla and stores in a variable.
3. We use .Single to pull one single field from the record being accessed.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations.Schema;

namespace WebAppLinq2.Models

{

public class Student

{

[Key]

[DisplayName("Student Number")]

public int StudentId { get; set; }

[DisplayName("Name")]

public string name { get; set; }

[DisplayName("Surname")]

public string surname { get; set; }

[DisplayName("Course ID")]

public int CourseID { get; set; }

public virtual Course Course { get; set; }

public double TotalFees { get; set; }

public double PullCourseFees()

{

StudentContext database = new StudentContext();

var cfees = (from s in database.Courses

where s.CourseID == CourseID

select s.CourseFees).Single();

return (cfees);

}

public double calcTotalFees()

{

return (PullCourseFees() + 3000);

}

}

public class Course

{//open course class

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

[DisplayName("Course ID")]

public int CourseID { get; set; }

[DisplayName("Course Name")]

public string CourseName { get; set; }

[DisplayName("Course Fees")]

public double CourseFees { get; set; }

}//close course table

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data.Entity;

namespace WebAppLinq2.Models

{

public class StudentContext:DbContext

{

public StudentContext() : base("StudDatabase1")

{

}

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

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

}

}

1. The highlighted portion should only change in the Create action method on the Student Controller.

public ActionResult Create([Bind(Include = "StudentId,name,surname,CourseID,TotalFees")] Student student)

{

if (ModelState.IsValid)

{

student.TotalFees = student.calcTotalFees();

db.Students.Add(student);

db.SaveChanges();

student.calcCoursFees();

return RedirectToAction("Index");

}

**LINQ method using SingleOrDefault**

**1. In the Student model create a method highlighted below. This method called updateCourseFees uses a LINQ query syntax to access the table Course and check if the CourseId from both tables are the same. Then it selects t.)SingleOrDefault which actually returns the entire record with all fields. After the record has been returned, it is stored in variable called co. The variable is used to access and update the CourseFees located in the Course table. We then save the changes and go to the controller and call this method to update the course fees.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations.Schema;

namespace WebAppLinq2.Models

{

public class Student

{

[Key]

[DisplayName("Student Number")]

public int StudentId { get; set; }

[DisplayName("Name")]

public string name { get; set; }

[DisplayName("Surname")]

public string surname { get; set; }

[DisplayName("Course ID")]

public int CourseID { get; set; }

public virtual Course Course { get; set; }

public double TotalFees { get; set; }

public void updateCourseFees()

{

StudentContext db = new StudentContext();

Course co = (from t in db.Courses

where t.CourseID ==CourseID

select t).SingleOrDefault();

co.CourseFees = co.CourseFees + 10000;

db.SaveChanges();

}

public double PullCourseFees()

{

StudentContext db = new StudentContext();

var cofees = (from m in db.Courses

where m.CourseID == CourseID

select m.CourseFees).Single();

return (cofees);

}

public double calcTotFees()

{

return (PullCourseFees() + 3000);

}

}

public class Course

{//open course class

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

[DisplayName("Course ID")]

public int CourseID { get; set; }

[DisplayName("Course Name")]

public string CourseName { get; set; }

[DisplayName("Course Fees")]

public double CourseFees { get; set; }

}//close course table

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data.Entity;

namespace WebAppLinq2.Models

{

public class StudentContext:DbContext

{

public StudentContext() : base("StudDatabase1")

{

}

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

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

}

}

2. In the controller the highlighted line must only be changed and then you will see the output.

public ActionResult Create([Bind(Include = "StudentId,name,surname,CourseID,TotalFees")] Student student)

{

if (ModelState.IsValid)

{

student.updateCourseFees();

student.TotalFees = student.calcTotFees();

db.Students.Add(student);

db.SaveChanges();

return RedirectToAction("Index");

}