

SOURCE CODE

```
public class RainbowSchoolDBContext : DbContext
{
    public RainbowSchoolDBContext() : base("name=RainbowSchoolDBContext")
    {
    }

    public DbSet<MVCAplicationSchoolDB.Models.Class> Class { get; set; }

    public DbSet<MVCAplicationSchoolDB.Models.StudentClass> StudentClass {
get; set; }

    public DbSet<MVCAplicationSchoolDB.Models.Subject> Subject { get; set; }
```

```
[Table("Class")]
public class Class
{
    [Key]
    public int ClassId { get; set; }
    [Required]
    [StringLength(50)]
    public string CName { get; set; }
    public virtual ICollection<Subject> Subjects { get; set; }
    public virtual ICollection<StudentClass> StudentClasses { get; set; }
}
```

```
[Table("StudentClass")]
public class StudentClass
{
    [Key]
    public int Id { get; set; }

    [Required]
    [StringLength(50)]
    public string FName { get; set; }

    [Required]
    [StringLength(50)]
    public string LName { get; set; }
    public int CId { get; set; }
    public virtual Class Classes { get; set; }
    public int SubId { get; set; }
    [JsonIgnore]
    public virtual Subject Subjects { get; set; }
}
```

SOURCE CODE

```

[Table("Subject")]
public class Subject
{
    [Key]
    public int SubId { get; set; }

    [Required]
    [StringLength(50)]
    public string SubName { get; set; }
    public int CId { get; set; }
    public virtual Class Classes { get; set; }
    public virtual ICollection<StudentClass> StudentClasses { get; set; }
}

}

public class ClassesController : Controller
{
    private RainbowSchoolDBContext db = new RainbowSchoolDBContext();

    // GET: Classes
    public ActionResult Index()
    {
        return View(db.Class.ToList());
    }

    // GET: Classes/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Class @class = db.Class.Find(id);
        if (@class == null)
        {
            return HttpNotFound();
        }
        return View(@class);
    }

    // GET: Classes/Create
    public ActionResult Create()
    {
        return View();
    }

    // POST: Classes/Create
    // To protect from overposting attacks, enable the specific properties
    // you want to bind to, for
    // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include = "ClassId,CName")] Class
@class)
    {
        if (ModelState.IsValid)
        {
            db.Class.Add(@class);
            db.SaveChanges();
        }
    }
}

```

SOURCE CODE

```

        return RedirectToAction("Index");
    }

    return View(@class);
}

// GET: Classes/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Class @class = db.Class.Find(id);
    if (@class == null)
    {
        return HttpNotFound();
    }
    return View(@class);
}

// POST: Classes/Edit/5
// To protect from overposting attacks, enable the specific properties
you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "ClassId,CName")] Class @class)
{
    if (ModelState.IsValid)
    {
        db.Entry(@class).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(@class);
}

// GET: Classes/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Class @class = db.Class.Find(id);
    if (@class == null)
    {
        return HttpNotFound();
    }
    return View(@class);
}

// POST: Classes/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Class @class = db.Class.Find(id);
    db.Class.Remove(@class);
    db.SaveChanges();
    return RedirectToAction("Index");
}

```

SOURCE CODE

```
protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
```

SOURCE CODE

```

@model MVCApplicationSchoolDB.Models.Class

@{
    ViewBag.Title = "Create";
}

<h2>Create</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Class</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.CName, "Class Name", htmlAttributes:
new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CName, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.CName, "", new { @class
= "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </div>
        </div>
    </div>

    <div>
        @Html.ActionLink("Back to List", "Index")
    </div>

    @section Scripts {
        @Scripts.Render("~/bundles/jqueryval")
    }

@model MVCApplicationSchoolDB.Models.Class

@{
    ViewBag.Title = "Delete";
}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Class</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.CName)
        </dt>

```

SOURCE CODE

```

        <dd>
            @Html.DisplayFor(model => model.CName)
        </dd>
    </dl>

    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default" /> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>

```

```

@model MVCApplicationSchoolDB.Models.Class

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
    <h4>Class</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.CName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.CName)
        </dd>
    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.ClassId }) |
    @Html.ActionLink("Back to List", "Index")
</p>

```

```

@model MVCApplicationSchoolDB.Models.Class

@{
    ViewBag.Title = "Edit";
}

```

SOURCE CODE

```

    }

    <h2>Edit</h2>

    @using (Html.BeginForm())
    {
        @Html.AntiForgeryToken()

        <div class="form-horizontal">
            <h4>Class</h4>
            <hr />
            @Html.ValidationSummary(true, "", new { @class = "text-danger" })
            @Html.HiddenFor(model => model.ClassId)

            <div class="form-group">
                @Html.LabelFor(model => model.CName, "Class Name", htmlAttributes:
new { @class = "control-label col-md-2" })
                <div class="col-md-10">
                    @Html.EditorFor(model => model.CName, new { htmlAttributes = new
{ @class = "form-control" } })
                    @Html.ValidationMessageFor(model => model.CName, "", new { @class
= "text-danger" })
                </div>
            </div>

            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <input type="submit" value="Save" class="btn btn-default" />
                </div>
            </div>
        </div>

        <div>
            @Html.ActionLink("Back to List", "Index")
        </div>

        @section Scripts {
            @Scripts.Render("~/bundles/jqueryval")
        }
    }

```

```

    @model IEnumerable<MVCApplcationSchoolDB.Models.Class>

    @{
        ViewBag.Title = "Index";
    }

    <h2>Index</h2>

    <p>
        @Html.ActionLink("Create New", "Create")
    </p>
    <table class="table">
        <tr>
            <th>

```

SOURCE CODE

```

        @Html.DisplayNameFor(model => model.CName)
    </th>
    <th></th>
</tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.CName)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.ClassId }) |
            @Html.ActionLink("Details", "Details", new { id=item.ClassId }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.ClassId })
        </td>
    </tr>
}

</table>

public class StudentClassesController : Controller
{
    private RainbowSchoolDbContext db = new RainbowSchoolDbContext();

    // GET: StudentClasses
    public ActionResult Index()
    {
        var studentClass = db.StudentClass.Include(s => s.Subjects);
        return View(studentClass.ToList());
    }

    // GET: StudentClasses/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        StudentClass studentClass = db.StudentClass.Find(id);
        if (studentClass == null)
        {
            return HttpNotFound();
        }
        return View(studentClass);
    }

    // GET: StudentClasses/Create
    public ActionResult Create()
    {
        ViewBag.SubId = new SelectList(db.Subject, "SubId", "SubName");
        return View();
    }

    // POST: StudentClasses/Create
    // To protect from overposting attacks, enable the specific properties you
    want to bind to, for
    // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]

```


SOURCE CODE

```

[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,FName,LName,CId,SubId")]
StudentClass studentClass)
{
    if (ModelState.IsValid)
    {
        db.StudentClass.Add(studentClass);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.SubId = new SelectList(db.Subject, "SubId", "SubName",
studentClass.SubId);
    return View(studentClass);
}

// GET: StudentClasses/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    StudentClass studentClass = db.StudentClass.Find(id);
    if (studentClass == null)
    {
        return HttpNotFound();
    }
    ViewBag.SubId = new SelectList(db.Subject, "SubId", "SubName",
studentClass.SubId);
    return View(studentClass);
}

// POST: StudentClasses/Edit/5
// To protect from overposting attacks, enable the specific properties you
want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Id,FName,LName,CId,SubId")]
StudentClass studentClass)
{
    if (ModelState.IsValid)
    {
        db.Entry(studentClass).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.SubId = new SelectList(db.Subject, "SubId", "SubName",
studentClass.SubId);
    return View(studentClass);
}

// GET: StudentClasses/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    StudentClass studentClass = db.StudentClass.Find(id);
    if (studentClass == null)
    {

```

SOURCE CODE

```

        return HttpNotFound();
    }
    return View(studentClass);
}

// POST: StudentClasses/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    StudentClass studentClass = db.StudentClass.Find(id);
    db.StudentClass.Remove(studentClass);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}

```

```
@model MVCApplicationSchoolDB.Models.StudentClass
```

```

@{
    ViewBag.Title = "Create";
}

```

```
<h2>Create</h2>
```

```

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>StudentClass</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.FName, "First Name", htmlAttributes:
new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.FName, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.FName, "", new { @class
= "text-danger" })
            </div>
        </div>

        <div class="form-group">

```

SOURCE CODE

```

        @Html.LabelFor(model => model.LName, "Last Name", htmlAttributes: new
{ @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.LName, new { htmlAttributes = new
{ @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.LName, "", new { @class
= "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.CId, "Class ID", htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.CId, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model => model.CId, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.SubId, "Subject ID", htmlAttributes:
new { @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("SubId", null, htmlAttributes: new { @class =
"form-control" })
            @Html.ValidationMessageFor(model => model.SubId, "", new { @class
= "text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Create" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

SOURCE CODE

```

@model MVCApplicationSchoolDB.Models.StudentClass

@{
    ViewBag.Title = "Delete";
}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>StudentClass</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Subjects.SubName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Subjects.SubName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.FName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.FName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.LName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.LName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.CId)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.CId)
        </dd>

    </dl>

    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default" /> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>

```

SOURCE CODE

```
@model MVCApplicationSchoolDB.Models.StudentClass

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
    <h4>StudentClass</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Subjects.SubName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Subjects.SubName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.FName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.FName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.LName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.LName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.CId)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.CId)
        </dd>
    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.Id }) |
    @Html.ActionLink("Back to List", "Index")
</p>
```

SOURCE CODE

```

@model MVCApplicationSchoolDB.Models.StudentClass

@{
    ViewBag.Title = "Edit";
}

<h2>Edit</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>StudentClass</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.FName, "First Name", htmlAttributes: new
            { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.FName, new { htmlAttributes = new
                { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.FName, "", new { @class
                = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.LName, "Last Name", htmlAttributes: new
            { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.LName, new { htmlAttributes = new
                { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.LName, "", new { @class
                = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.CId, "Class Name", htmlAttributes: new
            { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CId, new { htmlAttributes = new {
                @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.CId, "", new { @class =
                "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.SubId, "Subject ID", htmlAttributes:
            new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("SubId", null, htmlAttributes: new { @class =
                "form-control" })
                @Html.ValidationMessageFor(model => model.SubId, "", new { @class
                = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">

```

SOURCE CODE

```
        <input type="submit" value="Save" class="btn btn-default" />
      </div>
    </div>
  </div>
}

<div>
  @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
  @Scripts.Render("~/bundles/jqueryval")
}
```

SOURCE CODE

```

@model IEnumerable<MVCApplcationSchoolDB.Models.StudentClass>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Subjects.SubName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.FName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.LName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.CId)
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Subjects.SubName)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.FName)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.LName)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.CId)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
                @Html.ActionLink("Details", "Details", new { id=item.Id }) |
                @Html.ActionLink("Delete", "Delete", new { id=item.Id })
            </td>
        </tr>
    }
</table>

```


SOURCE CODE

```

public class SubjectsController : Controller
{
    private RainbowSchoolDBContext db = new RainbowSchoolDBContext();

    // GET: Subjects
    public ActionResult Index()
    {
        return View(db.Subject.ToList());
    }

    // GET: Subjects/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Subject subject = db.Subject.Find(id);
        if (subject == null)
        {
            return HttpNotFound();
        }
        return View(subject);
    }

    // GET: Subjects/Create
    public ActionResult Create()
    {
        return View();
    }

    // POST: Subjects/Create
    // To protect from overposting attacks, enable the specific properties
    // you want to bind to, for
    // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include = "SubId,SubName,CId")] Subject
subject)
    {
        if (ModelState.IsValid)
        {
            db.Subject.Add(subject);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        return View(subject);
    }

    // GET: Subjects/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Subject subject = db.Subject.Find(id);
        if (subject == null)
        {
            return HttpNotFound();
        }
    }
}

```

SOURCE CODE

```

    }
    return View(subject);
}

// POST: Subjects/Edit/5
// To protect from overposting attacks, enable the specific properties
you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "SubId,SubName,CId")] Subject
subject)
{
    if (ModelState.IsValid)
    {
        db.Entry(subject).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(subject);
}

// GET: Subjects/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Subject subject = db.Subject.Find(id);
    if (subject == null)
    {
        return HttpNotFound();
    }
    return View(subject);
}

// POST: Subjects/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Subject subject = db.Subject.Find(id);
    db.Subject.Remove(subject);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}

```