

ASP.NET Active Dynamic Web Pages

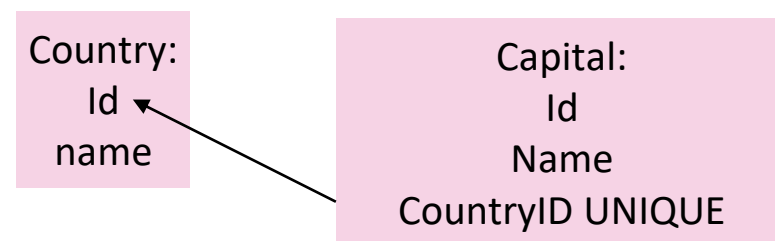
יחסי גומלין

תבנית העיצוב Repository

יחסי גומלין – אחד לאחד

דוגמא – לכל מדינה יש עיר בירה אחת – עיר בירה שייכת למדינה אחת.
לכל עובד יש כתובת מגורים אחת. כתובת מגורים שייכת לעובד אחד.

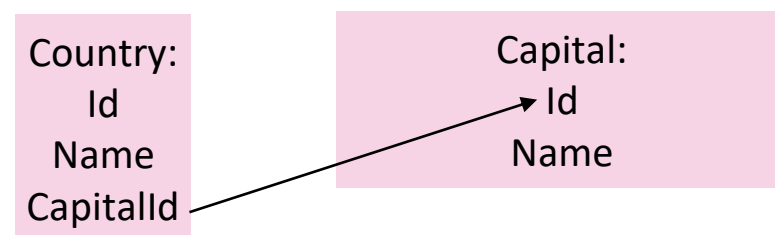
תיאור סכמטי:



במערכת יחסים של 1:1

אפשר לשים את המפתח הזר על עיר הבירה
או על המדינה

כי זה לא משנה – למדינה יש עיר בירה אחת
לעיר בירה יש מדינה אחת



יחסי גומלין – אחד לאחד

מימוש בC#

```
public class Country
{
    [Key]
    10 references
    public int Id { get; set; }

    [Required]
    12 references
    public required string Name { get; set; }

    0 references
    public Capital? Capital { get; set; }
}
```

```
public class Capital
{
    [Key]
    0 references
    public int Id { get; set; }

    [Required, MinLength(1), MaxLength(30)]
    0 references
    public required string Name { get; set; }

    //One to One Relationship - Foreign Key:
    0 references
    public int CountryId { get; set; }

    //Navigation Properties:
    0 references
    public required Country Country { get; set; }
}
```

יחסי גומלין – אחד לרבים

דוגמא – לכל עובד יש מנהל אחד – למנהל יש מספר עובדים.
דוגמא – באלבום יש הרבה שירים – שיר שייך לאלבום אחד.

תיאור סכמטי:



במערכת יחסים של 1:*

אפשר לשים את המפתח הזר רק על השיר

יחסי גומלין – אחד לרבים

מימוש ב-C#

```
public class Album
{
    [Key]
    12 references
    public int Id { get; set; }

    [Required]
    14 references
    public required string Name { get; set; }

    //Navigation Property:
    0 references
    public ICollection<Song> Songs { get; set; } = [];
}
```

```
public class Song
{
    [Key]
    11 references
    public int Id { get; set; }

    [Required]
    12 references
    public required string Title { get; set; }

    //One to Many Relationship - Foreign Key:
    0 references
    public int? AlbumId { get; set; }

    //Navigation props:
    3 references
    public Album? Album { get; set; }
}
```

באחד לרבים – מכיוון שיש רק דרך אחת להגדיר את היחס – לא חייבים לכלול במודל את AlbumId - הספרייה Entity Framework תדע ליצור את הטבלאות נכון. כלומר גם בלי לכלול אותה במודל – בטבלאות בדטה-בייס יהיה AlbumId

אבל זה נוח להחזיק את התכונה הזאת למטרות Debugging והבנה.

יחסי גומלין – רבים לרבים

דוגמא – כל מוצר שייך למספר קטגוריות – בקטגוריה יש הרבה מוצרים
דוגמא – כל סטודנט משתתף בהרבה קורסים – בקורס יש הרבה סטודנטים
דוגמא – כל סרט שייך למספר ג'אנרים – בג'אנר יש הרבה סרטים

Movie:
Id
Title

Genre:
Id
Name

תיאור סכמטי:

במערכת יחסים של **:*

MovieGenres:
MovieId
GenreId

יש רק דרך אחת להגדיר זאת – טבלה מקשרת.

טבלה מקשרת

יחסי גומלין – רבים לרבים

מימוש ב-C#

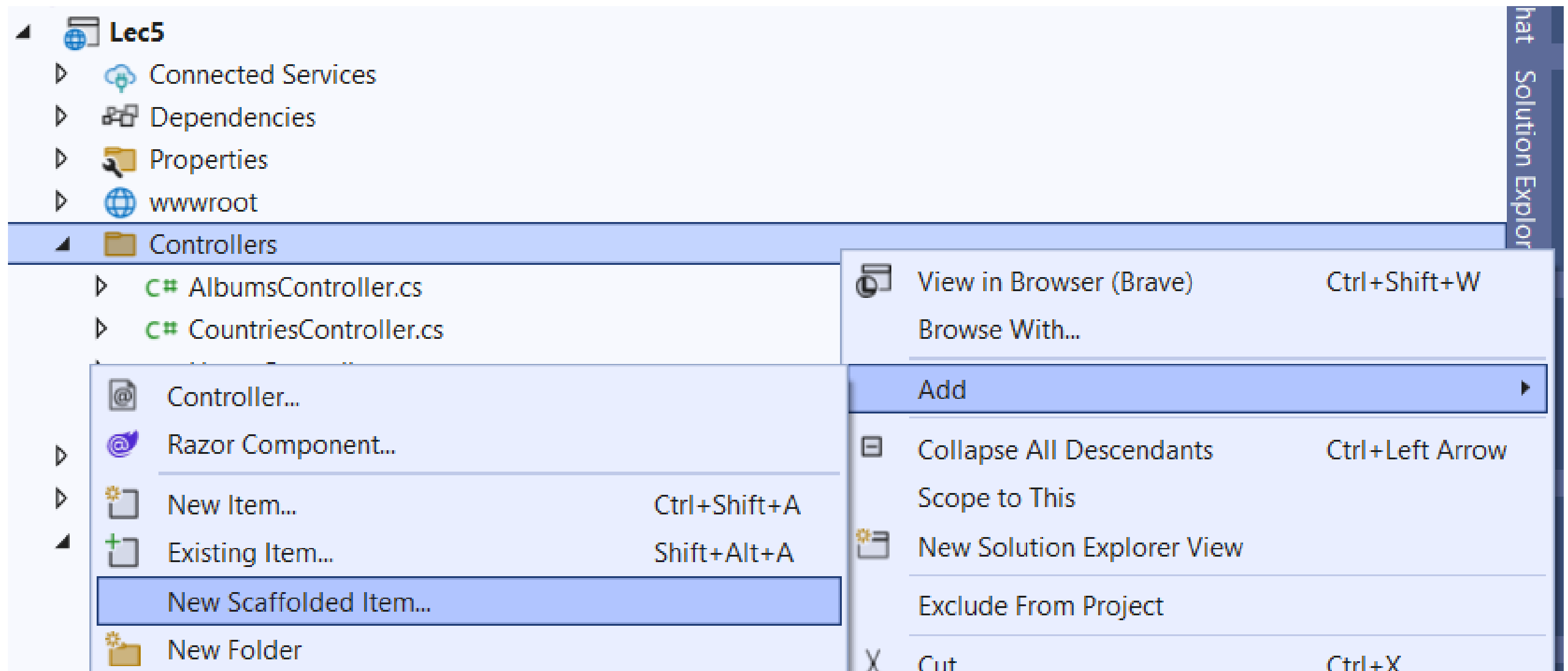
```
public class Movie
{
    0 references
    public int Id { get; set; }
    0 references
    public required string Title { get; set; }

    //Navigation Properties
    0 references
    public ICollection<Genre> Genres { get; set; } = [];
}
```

```
public class Genre
{
    0 references
    public int Id { get; set; }
    0 references
    public required string Name { get; set; }

    //Navigation Properties
    0 references
    public ICollection<Movie> Movies { get; set; } = [];
}
```

יצירת Controller כולל DbContext והוספת ספריות של Entity Framework



יצירת Controller כולל DbContext והוספת ספריות של Entity Framework

Add New Scaffolded Item

▲ Installed

- ▲ Common
 - API
 - ▲ MVC
 - Controller**
 - View
 - Razor Component
 - Razor Pages
 - Identity
 - Layout

MVC Controller - Empty

MVC Controller with read/write actions

MVC Controller with views, using Entity Framework

MVC Controller with views, using Entity Framework
by Microsoft
v1.0.0.0

An MVC controller with actions and Razor views to create, read, update, delete, and list entities from an Entity Framework data context.

Id: MvcControllerWithContextScaffolder

יצירת Controller כולל DbContext והוספת ספריות של Entity Framework

×

Add MVC Controller with views, using Entity Framework

Model classCountry (Lec5.Models)▼

DbContext class<Required>▼+

×

Add Data Context

New data context typeLec5.Data.Lec5Context

AddCancel

Controller nameCountryController

AddCancel

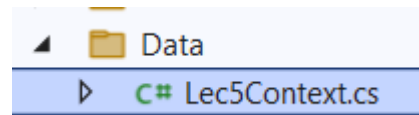
עדכון של Connection String שיתאים לדטה-בייס שלנו

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "Lec5Context": "Server=DESKTOP-GHIEC0F\\SQLEXPRESS;Database=Lec5DB;Trusted_Connection=True;TrustServerCertificate=True"
  }
}
```

שם השרת

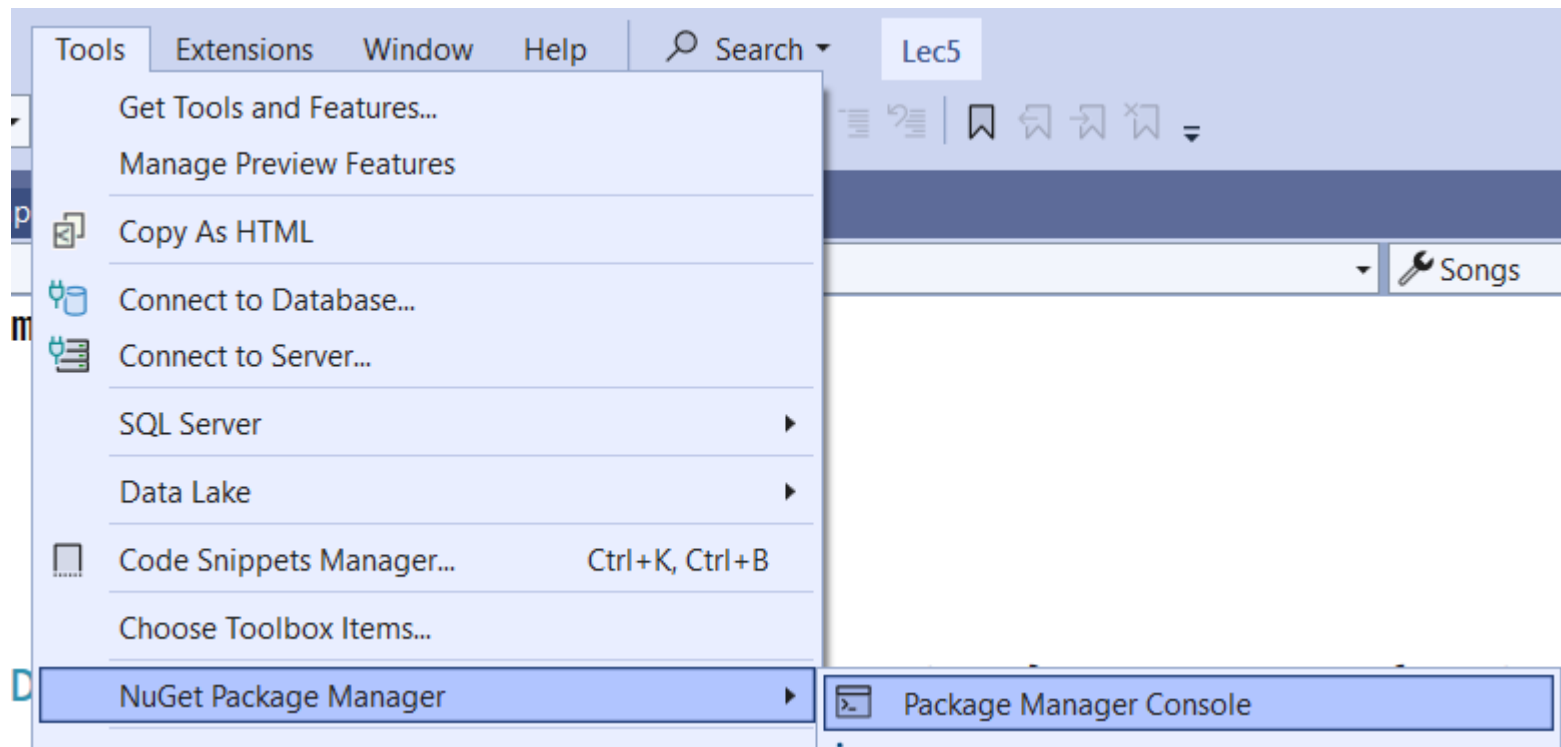
שם הדטה-בייס

עדכון הדטה-בייס לפי המודלים:



```
public class Lec5Context(DbContextOptions<Lec5Context> options) : DbContext(options)
{
    7 references
    public DbSet<Country> Countries { get; set; } = default!;
    0 references
    public DbSet<Capital> Capitals { get; set; } = default!;
    7 references
    public DbSet<Song> Songs { get; set; } = default!;
    8 references
    public DbSet<Album> Albums { get; set; } = default!;
    0 references
    public DbSet<Movie> Movies { get; set; } = default!;
    0 references
    public DbSet<Genre> Genres { get; set; } = default!;
}
```

עדכון הדטה-בייס לפי המודלים:



```
add-migration DBInit
```

```
update-database
```

הצגה של יחסי גומלין בממשק משתמש:

```
// GET: Songs
3 references
public async Task<IActionResult> Index()
{
    var songs = await context.Songs.Include(s => s.Album).ToListAsync();

    return View(songs);
}
```

מקביל ל

```
Select * FROM Songs
Join Albums ON Songs.AlbumId = Albums.Id
```

Index.cshtml הצגת שם האלבום עם פרטי השיר

```
<thead>
  <tr>
    <th>
      @Html.DisplayNameFor(model => model.Title)
    </th>
    <th>
      @Html.DisplayNameFor(model => model.Album)
    </th>
    <th></th>
  </tr>
</thead>

<tbody>
  @foreach (var item in Model) {
    <tr>
      <td>
        @Html.DisplayFor(modelItem => item.Title)
      </td>
      <td>
        @{
          var album = item.Album?.Name ?? "No Album";
        }
        @album
      </td>
      <td>
        <a asp-action="Edit" asp-route-id="@item.Id">Edit</a> |
        <a asp-action="Details" asp-route-id="@item.Id">Details</a> |
        <a asp-action="Delete" asp-route-id="@item.Id">Delete</a>
      </td>
    </tr>
  }
</tbody>
```

הצגת יחסי גומלין בממשק משתמש: הצגת טופס ליצירת שיר עם בחירת אלבום

```
// GET: Songs/Create
```

0 references

```
public async Task<IActionResult> Create()
```

```
{
```

```
    var albums = await context.Albums.ToListAsync();
```

```
    //pass the albums to the view  
    ViewBag.Albums = albums;
```

```
    return View();
```

```
}
```

נשלוף את כל האלבומים כך
שנוכל להציג את הפרטים
בבחירת אלבום בעת יצירת
שיר

העברה של פרטי האלבום לView

שימוש ב ViewBag מאפשר לנו להעביר
תכונות נוספות לView בנוסף להעברה
הרגילה ע"י View(model)

הטיפוס של ViewBag הוא דינאמי ואפשר
להעביר באן כל אובייקט

הצגת יחסי גומלין בממשק משתמש:

Create.cshtml הצגת טופס ליצירת שיר עם בחירת אלבום

```
@model Lec5.Models.Song
```

```
@{  
    ViewData["Title"] = "Create";  
    List<Album> albums = ViewBag.Albums;  
}
```

חילוץ המידע על האלבומים
מבאג

```
<h4>Add a Song</h4>  
<hr />  
<div class="row">  
    <div class="col-md-4">  
        <form asp-action="Create">  
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>  
            <div class="form-group">  
                <label asp-for="Title" class="control-label"></label>  
                <input asp-for="Title" class="form-control" />  
                <span asp-validation-for="Title" class="text-danger"></span>  
            </div>  
            <div class="form-group">  
                <select name="albumId">  
                    <option>Pick A Value</option>  
                    <option>Single</option>  
                    @foreach (var album in albums)  
                    {  
                        <option value="@album.Id">@album.Name</option>  
                    }  
                </select>  
            </div>  
        </form>  
    </div>  
</div>
```

הצגה של
לבחירת פרטי האלבום

שיעורי בית

צרו פרוייקט חדש והוסיפו בו מודל עבור שחקן
לכל שחקן יש שם פרטי, שם משפחה וId

צרו Controller עם add new Scaffold item
ותוך כדי התהליך צרו DbContext ותוסיפו את הספריות

שימו לב לעדכן את ה Connection String בקובץ
appsettings.json

בצעו מיגרציה ועדכנו את הדטה-בייס

שיעורי בית

יחסי גומלין – צרו מחלקה עבור סרט:

לכל סרט יש כותר ו ID

כל שחקן יכול לשחק בהרבה סרטים ובסרט יש הרבה שחקנים –
רבים לרבים.

ממשו את יחסי הגומלין ע"י הוספת Navigation Properties

עדכנו את DbContext

בצעו מיגרציה

עדכנו את הדטה-בייס

בדקו בדטה-בייס שהכל עודכן לשביעות רצונכם.

שיעורי בית

יחסי גומלין – צרו מחלקה עבור במאי ראשי.
לכל במאי יש ID ושם.

לכל סרט יש במאי ראשי
לבמאי יש הרבה סרטים.
(אחד לרבים)

ממשו את יחסי הגומלין ע"י הוספת Navigation Properties
עדכנו את DbContext
בצעו מיגרציה
עדכנו את הדטה-בייס
בדקו בדטה-בייס שהכל עודכן לשביעות רצונכם.

שיעורי בית

יחסי גומלין – צרו מחלקה עבור פרס אוסקר לסרט הטוב ביותר:
לכל פרס יש שנה ומזהה של הסרט הנבחר.

לכל סרט יש פרס אוסקר אופציונלי

כלומר – יחסי גומלין של אחד לאחד.

ממשו את יחסי הגומלין ע"י הוספת Navigation Properties

עדכנו את DbContext

בצעו מיגרציה

עדכנו את הדטה-בייס

בדקו בדטה-בייס שהכל עודכן לשביעות רצונכם.

שיעורי בית

צרו קונטרולר להוספת סרט
צרו קונטרולר להוספת שחקן
צרו קונטרולר להוספת פרס
צרו קונטרולר להוספת במאי ראשי

הכניסו מידע לדוגמא לדטה-בייס
הכניסו מספר סרטים, מספר במאים, מספר פרסי אוסקר, מספר
שחקנים.

דאגו למלא את יחסי הגומלין ב data ששמור במסד הנתונים.

הציגו את יחסי הגומלין בעמודי ה index השונים של
ה Controllers שיצרתם.

בהצלחה!