# Practical week 8 – Building a Front End on .NET

*This is a checkpoint. Due Friday, 21 September by 5.00 pm. You may work in pairs if you wish.*

The relational schema and data dictionary below define a simple database for cataloguing a DVD film collection. For this practical, you will implement the database and build a front end to allow the user to search and modify the database without having to write SQL.

First, create the tables and populate them with some simple test data (use imdb.com or rottentomatoes.com for test data inspiration).

You may then build this application either by:

1. Writing a desktop application in C# using ADO.NET and Windows Forms
2. Writing a web application in C# using ASP.NET and ADO.NET

I suggest that you don’t attempt to use ASP.NET MVC, or Microsoft’s Entity Framework (at least for now). These are both ORMs.

There’s a pretty good tutorial for writing your first ADO.NET application at:

<https://www.codeproject.com/Articles/361579/A-Beginners-Tutorial-for-Understanding-ADO-NET>

And a wealth of official Microsoft documentation at:

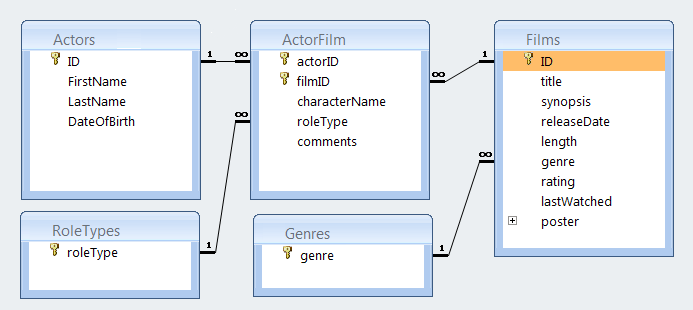
<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/>

1. Implement the database as shown. Feel free to add additional tables and/or attributes if you feel they would add value to your product.
2. Using your chosen method, design and build the following forms for your database:
   1. Data entry form allowing addition of a new actor
   2. Data entry form allowing addition of a new film
   3. Data entry facility (either a separate form or embedded controls in one of the preceding forms) to associate actors as cast members of movies.
3. Using your chosen method, design and build the following reports for your database:
   1. Summary of all films, organised by genre, not showing cast members.
   2. Summary of all actors, showing films and role type for each film.
   3. List of all films showing all cast members for each film.
4. Include a main navigation screen that allows the user to jump to any data entry form or report with a button click.

Make your forms and reports as clear and elegantly designed as you can. You may, of course, build any additional forms or reports that you feel would add value to your product.

Please populate your database with sufficient test data that the various forms and reports can be evaluated for layout and correctness.

Relational schema:



Data Dictionary:

|  |  |  |
| --- | --- | --- |
| Actors | | |
| ID | AutoNumber (Counter in SQL) | Surrogate ID |
| FirstName | Text | Actor’s first name |
| LastName | Text | Actor’s family name |
| DateOfBirth | Date | Actor’s birth date |
| Films | | |
| ID | AutoNumber (Counter in SQL) | Surrogate ID |
| title | Text | Title of Film |
| synopsis | Memo | Free text summary of film plot |
| releaseDate | Date | Date of commercial release |
| length | Number | Length in minutes |
| genre | Text | Genre. References Genres (LKP) |
| rating | Number | Your rating 1 to 5 |
| lastWatched | Date | Most recent viewing |
| poster | Attachment | Thumbnail of movie poster |
| ActorFilm | | |
| actorID | Number | FK. References Actors |
| filmID | Number | FK. References Films |
| characterName | Text | Name of character actor played |
| roleType | Text | Role. References Role (LKP) |
| comments | Memo | Free text comments on film |
| RoleTypes | | |
| roleType | Text | Look-up table. List of role types. |
| Genres | | |
| Genre | Text | Look-up table. List of genres |