

Databases Assignment 1:  
Internet Movie Database (IMDB.com)

DT228/4

BSc in Computer Science

Leslie Ducray

C10327999  
15 November 2013

# Introduction

Imdb.com is a massive online database storing all information relative to film, television and video games. The web site stores and displays the information on all actors, cast and staff related to each title. It also offers users the ability to rate and review titles, message boards, related news feeds, users to create lists of titles/actors/actress/celebrities. I have chosen to analyse and replicate only a small portion of the website’s functionality. The independent aspect I have developed encapsulates the data, entities and relationships to model the ability to:

* Login
* View user information
* View celebrities (actors, cast and crew, etc.)
* View the category of their involvement to a title (actor, writer, producer etc.)
* View titles they have been involved in
* view the characters they have played
* View what the other title the character has appeared in.
* View titles
* View the title’s genres
* Rate a title
* Review a title

The data for the above information is stored below in the implemented ERD design (fig.1).

The application business process I have chosen to mimic using data entities, triggers and set procedures is the ability for a user to rate a ‘Title’ which will update the Title’s average score and write a review on a title.

There are multiple aspects of the current system I have left out, this is due to the general magnitude of the application and that these features are outside of my desired business scope. Immediate aspects which could have been included to expand my design are: more celebrity information, family and relationships, biographies, personal quotes, trivia, trademark, character genres, taglines, plot keywords, filming locations, company credits, technical specifications, soundtracks and much more, but each of these would require another table or two for associations. Another aspect I have chosen not to implement in my design is "Full Episode Support" which will hold each independent episode of a television series, storing the cast and crew involved. This could have easily been implemented by associating a television title with a list of episodes.

In my implementation, primarily the tables are populated by means of hardcoded insert statements, otherwise tables which are dependant of User feedback (eg. The ’User\_Rating’ table, which records the User, title and the user’s score of that title and the ‘Review’ table) are populated via a stored procedure.   
After Insert into the ‘User\_Rating’ table, a trigger is called which calculates the Title’s average score and outputs it into the Title table where the title score is recorded.  
Another proceadure allows user’s to enter a score and review on a title, this is similar to the functionality of the IMDB website. However, how IMDB calculates the title’s average is not disclosed, I have found a formula according to Wikipedia is what is used to calculate a titles weighted average.  
The formula is : W = \frac{Rv + Cm}{v+m} , where W = the weighted average.  
R= the titles average score, v = the amount of votes on the title, C= the weighted average of all titles , m = the minimum votes required to enter into the Top 250 category (a feature of IMDB),this is set as 25000 votes . The score attached to a review of a title is not used in calculating a Title’s overall score; it is used just as an indication to other users reading a user’s review.

# ER Diagrams

# 

Figure 1: Conceptual

The rectangular entities represent the independent tables storing different general aspects of the system (eg. User information, Title information, Celebrity information , etc.).  
The diamond shaped entities represent the relationship tables which are dependant on the rectangle entities. The arrows illustrate the direction of dependancy (eg. ‘Character appearences’ is dependant on both ‘Title’ and ‘Character’). The diamond entities store the associations which which capture the websites primary function and linking the relative information and relationships between different entities.

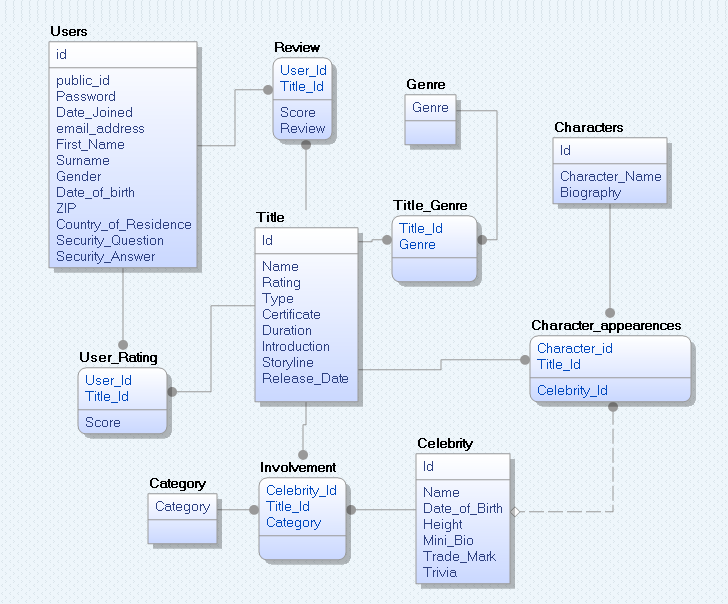


Figure 2: ER Diagram

The ER Diagram confirms that the Sql create statements developed implement the conceptual diagram. The diagram also includes the attributes and keys needed.

# Implementation

The design of the system began by studying and analysing the existing IMDB website. Once the target business aspect of the system was decided, it was then necessary to figure what data will be stored in which tables and the relationships between potential table entities.  
 It was decided to develop the scheme to incorporate a fairly normalized table layout, to ensure less dependency between attributes and more maintainability throughout the system.

The implementation process began at a conceptual level to detemine and illustrate the relationships between the required table entities. This was drawn out first on paper and then designed using an online prototyping tool. The Conceptual diagram is represented in fig:1.  
Once the conceptual diagram was designed, the SQL table creates were designed and attributes correctly constrained. The script was written and could then be reverse enginneered to produce a ER Diagram, to confirm the correctness of the implementation when compared against the conceptual ERD.

The primary business function of this application is to allow the user to rate against a title.  
When a user rates against a title, a trigger is activated this will calculate the title’s average rating and update the stored attribute. Unfortunately due to non-disclosure of the weighted average calculation used by IMDB for calculating a title’s rating, I have opted to calculate the shows average rating.  
If a user has already rated a title, the new score is updated on the ‘user\_rating’ table and the the new title average is calculated normally by the trigger.