INST 327, Section 0202

Project Final Report

12/14/20

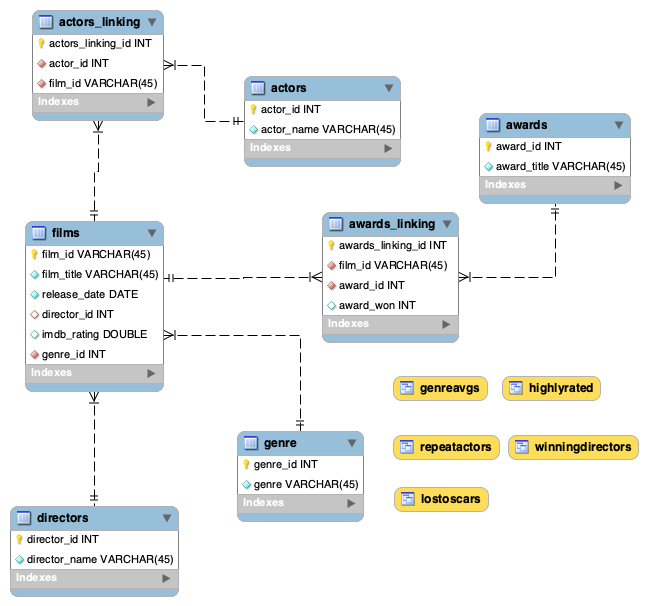
Team 3: Daniel Ely, John Gilligan, Gregory Sarafian, Shutian Wang, Sen Luo

**Introduction**

The topic our group has chosen to cover is movie ratings from 2018. It will cover 950 of the movies released that year, which were produced in the USA. This includes films from the box office as well as Netflix. We will cover topics such as each film’s main stars, director, and genre. The criteria in which our database rates movies is IMDB ratings, Oscar awards and nominations. We chose this topic because we felt it would be very interesting. With all of the hecticness going on at the moment, movies provide a healthy, uplifting distraction. Since a majority of movie production in 2020 has come to a halt, going back through 2018’s movies provides a good alternative to catch up to what has been popular recently.

This database will be particularly useful for doing advanced searches on movie data. It’s not challenging to find information about which movie from 2018 won best picture. But our database will be able to answer much more extensive questions such as, “Which Oscar nominated films had the highest IMDB ratings?” Being able to answer complex questions like this will help the user to discover a much more powerful movie recommendation for themself. Many people enjoy watching movies, and a lot of those people are curious about how different movies stack up against each other. Being able to compare movies to each other through several different facets will be an interesting and enlightening experience for them. Our database provides exactly that, we will explore its features throughout this report.

**Logical Design**



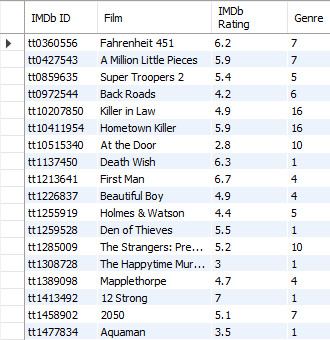
The key components of our database consist of seven tables, two of which are linking tables. The films table contains two foreign keys, connecting it to the genre and directors tables. The films table is also connected to the actors and awards table through their respective linking tables.

**Physical Design**

The database is designed to allow for easy joining between tables. This is because we do not store the names of the films in any table besides the films table. Therefore, we had to allow for all the other tables to access that information. This was also done to assist us in our sample queries, so we could easily create ones that were complex enough. Our database is packed with useful information as well. There are hundreds of titles from 2018 listed, along with every oscar award they won. Their 3 main lead actors are included, as well as the director and genre. With this information, it is easy to filter your results to find anything you need to know about films from 2018.

**Sample Data**

The data in our final database was retrieved from Kaggle website - IMDb movies extensive dataset (<https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset>). We built our database on the raw data we downloaded from this website, and successfully retrieved data on over 81k movies and 175k cast members scraped only from IMDb. We updated our database information to make it more clear and specific. We condensed the rating part because some of its detail is beyond what is necessary. During the reorganized part, one of the biggest challenges we faced was the Actor part. We found out each film comes with 0-3 actors and about 300 actor names appear more than one time in this dataset. To make each of the names an individual actor\_id is a little bit of a struggle. We decided to manually select the same actor, and mark them. Then we make them the same ID and reorder the actor\_id. Below is a sample data from our film table.



**Views & Queries**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Query** | **JOIN** x4 | **WHERE** x3 | **Aggregation** x2 | **Linking** x1 | **Subquery** x1 |
| winningDirectors | ✓ | ✓ |  | ✓ | ✓ |
| lostOscars | ✓ | ✓ | ✓ | ✓ |  |
| highlyRated | ✓ | ✓ |  |  |  |
| genreAvgs | ✓ |  | ✓ |  |  |
| repeatActors | ✓ |  | ✓ | ✓ |  |
| **Total** | **5** | **3** | **2** | **3** | **1** |

The following list outlines the intent behind each written query.

|  |  |  |
| --- | --- | --- |
| **Query 1:** | winningDirectors | Displays director information on Oscar winning movies. |
| **Query 2:** | lostOscars | Displays the number of oscars that a film was nominated for but did not win. |
| **Query 3:** | highlyRated | Displays movies, and their genres, which received 8 on IMDB or higher. |
| **Query 4:** | genreAvgs | Displays the average film score from each genre. |
| **Query 5:** | repeatActors | Shows a list of all actors who were major stars in at least one movie. |

**Changes from Original Design**

We made a lot of changes throughout the course project. One topic of these changes was figuring out which data to include or exclude. We were very ambitious to include as much data in our project, but, upon reflection, we realized a lot of this would either not be feasible or would not add value to achieving our scope. The criteria that we used to eliminate data was essentially, ‘does this data piece add value to our mission?’ One example of data that we cut was the Oscar presenter. Who presented the Oscar to the movie that won the award is not relevant to how good that movie is. It wouldn’t make much sense to say show me all movies that won oscars that were presented by Tom Hanks. Another example of this was our decision to strip budget and ticket sales. This was twofold. The first part was that our main data source did not have budget information for all movies, so the data itself was not extensive. Second, the budget does not necessarily influence a user’s decision making in finding a movie to watch. A user would not necessarily be interested in a movie with an enormous budget if the reviews were awful, or it was not in their genre of interest. Using this rule of thumb described, we condensed our data so that the remaining data would all be useful to our project’s scope.

A second topic of changes we made was our table design. We neglected to realize, in the beginning, that many of our relationships were going to be many-to-many and thus a linking table would be encouraged. For example, we didn’t think that there would be many repeat-film directors, but it turns out there were dozens of them. We also had decided to move ratings from its own separate table back to the films table. Since ratings were essentially defined by one value, it was redundant to have a ratings\_id number that pointed to another table in which it described another number. As our feel for which table relationships got better, we were able to pick up on these nuances to better our ERD and table structures.

**Lessons Learned**

The one challenge that we are facing is that unlike other assignments, group work is very important for this final assignment. A large challenge of this project was being able to coordinate and delegate different roles to different team members. Some team members were in different time zones. In order for everyone to have a clear understanding of what problem we are having and what the database would be like, we had to work remotely and share screens in order to be collaborative. We spend most of our meeting time talking about the database and discussing how to change it. Once everybody knows what to do we can have people adjust it during their private time. Another thing that we learned is the data import and export. With the data import process, we could have a better understanding about the relationship between all the different tables which allowed us to work on our ERD inspection.

A large lesson gained for all of our team members is our feel for normalization and database design. Throughout the course of this semester, this group project has allowed us to get a lot of hands on experience in design decisions. Some of us were not extremely familiar with what a linking table was in the initial phase of our project. Now, we all are familiar and proficient in concepts like this. Many of the concepts we learned were presented in class and in the textbook, but actually going through the process of getting our hands dirty with the data allowed us to really cement our understanding.

**Future Work**

While we achieved virtually all of our overarching goals with this project, there remain a few areas where we believe we could expand upon our database. As was addressed in the *Changes from Our Original Design* section, there were a few aspects that we originally intended to include but had to omit due to time and/or difficulty constraints. Namely, including foreign films in our database added an additional level of complexity, that although we had to omit, is a crucial component in recognizing Oscar winners. For instance films like Roma, which received critical acclaim should absolutely be included in future versions of this project. An additional metric we’d aim to include, to supplement IMDb ratings, would be Metacritic scores as well as Rotten Tomato scores. If we are able to include, in a future iteration, both of these types of ratings methods, we may be able to garner a better understanding of which films truly were the most well-received rather than relying on one metric. Additionally, including box office sales or first-week streams could be another way to further our understanding of which films were the most successful. One final way that we envision our database could be improved is via expanding upon our pool of actors and directors. Because of technical constraints, we had to limit the number of actors and directors associated with a film to three. Although given the scope of the project we felt this was the right decision, nonetheless it is important for us to aim to include more than just the three leading actors and directors as our work progresses. Overall, we are immensely proud of the work we’ve done and are incredibly grateful for the support we received from our professor, teaching assistants, and AMPs. The knowledge and experience garnered throughout this project are immensely valuable and we remain thankful for the opportunities afforded to us.

**Works Cited**

Leone, Stefano. IMDb Movies Extensive Dataset, Kaggle, 2020,

www.kaggle.com/stefanoleone992/imdb-extensive-dataset.