

## **Proposal Review**

Due: April 6th, 2021

### **Introduction**

Due to detailed explanations with attributes like director, actors, and average review score, we could easily recognize a possible data structure that uses IMDb movie platform. After the potential application examples, the goal of the project was to compare how movie trends might have changed over time by using IMDb database to separate decades of movies into different tables.

### **Target Audience**

The flexibility of the database is extremely useful, but it can also be harmful. Team 3's database could include everyone as its target audience from recommending a movie, based on other users' similar preference. However, the Target Audience is identified first to shape the design and to meet the Target Audience's needs. Since the Target Audience shapes the initial design, the Target Audience should be someone who will use the database the most.

### **Sample Data**

Sample Data section provided 3 different, sampled entities that had many discrepancies in attribute names, ordering of attributes, and number of movies. Also, some columns like director's name, cast, genre, and movie\_country needed to be separated into two separate columns to prevent errors. Additionally, some columns had discrepancies in the type of data input like how a duration column can be both "4 Seasons" and "93 min". Due to many error possibilities and a lack of plan to build a sample data, the plan seems unrealistic.

### **Potential Entities**

With only one entity displayed with 8 attributes included, the team's entity setup looks very incomplete. Even though the team mentioned that it will use a different table for each decade of movies, the team's work is still incomplete. The team failed to include any information about how the relationship will be established from one table to another with primary key and foreign key identification to create a relational database. Also, there is no shared understanding on how the tables will be connected and how the tables will be compared. For example, table A joins table B using movie\_id will display X attributes. However, if there are 7 or more tables

with the same attributes, why bother making 7 other tables when you can just self join with itself? Thus, the team should redesign from the beginning and better devise a plan on how to create their database, because the current plan is incomplete and it is full of errors. Lastly, since IMDb only provides movie information, not demographic information, the team should explore on how to find the demographic information.

### **Entities That Will Not be Included in the Database**

Team 3 displayed a conceptual understanding on why unnecessary entities must be eliminated. However, the team failed to separate unnecessary and necessary entities correctly. The team had decided to exclude two entities, which are “actors” and “black and white or in color”. Firstly, for the entity “actors”, the team misjudged the relationship deduction with “actors” importance in a movie. Even though directors direct actors, most people only know the actors, not the directors. Since the team’s tables will require all information that might influence the user’s movie preference, the entity “actors” will play a huge role. Secondly, for the entity “black and white or in color”, the user can know whether the movie will be shown as black and white or as colored. The team is correct in the assumption that there are fewer black and white movies, but the team decided to assign each entity with a different decade until the beginning in 1920’s. During the 1920's, many movies were black and white, which would result in an entire entity with black and white. Thus, both entities “actors” and black and white or in color” are both necessary and should not be eliminated.

### **Questions the database will be able to answer**

Some questions are very simple and answered by selecting a specific attribute value. However, a few questions are complex enough like locating the most popular genre of movies during different time periods. Even though the complexity of the question is appropriate, the answer cannot be found, based on the potential entities. Since the entities don't include numerical rating, there cannot be an answer. The team should ask these complex questions during redesign, so that they can see how the database fails.