Database Systems, spring 2014
Mini Project

Elias Obeid
`eobeid11@student.aau.dk`

Kent Caspersen
`kcaspe11@student.aau.dk`

Martin Madsen
`mbma11@student.aau.dk`

`d601f14, 1.1.01`
6th February, 2014 to 12th March, 2014

1 Self Study 1: Preliminary Database Modeling

Deadline: Wednesday 12th February, 2014

As stated in the assignment, we have decided to look at different possible attributes and models by looking at the structure of movie pages on IMDB. Initially, we think it would require many join tables, as we've identified a few many-to-many relationships among structures we've discussed. These structures are: *actors*, *directors*, *writers*, *movies*, *awards*, *ratings*, and *users*.

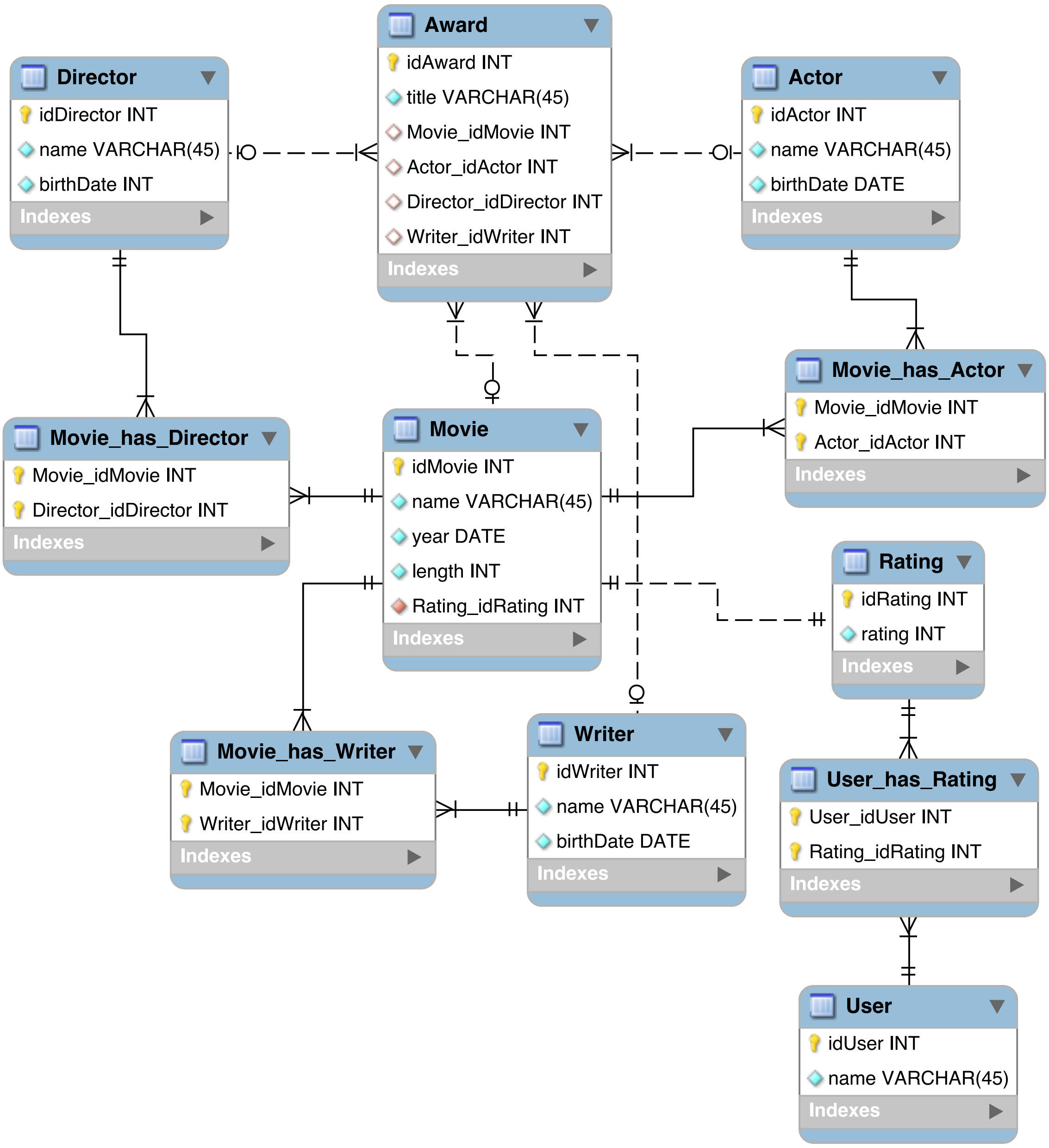

Figure 1: *Enhanced entity-relationship (EER) model diagram of a simplified movie database.*

We spent time on figuring out how to map the relationships between tables instead of focusing on the attributes. In our opinion it is easy to just add a birthdate if that should be necessary.

Figure 1 shows relationships between the chosen models and their corresponding join tables. Dashed lines between tables repressent *non-identifying* relationships and solid lines between tables represent *identifying* relationships.

When lines branch toward a table then there is a “has many” relationship to that table. When the lines have two orthogonal dashes (or a orthogonal dash and a circle) by a table then there is a “has one” relationship to that table. If there is a circle then the relationship is non-identifying. For example one *Director* has many *Awards*. The relationship is also non-identifying because the tables can exist indenpendently of each other.

2 Self Study 2: Database Modeling

Deadline: Wednesday 12th March, 2014

Entity-relationship Diagram

In figure 2, we show an updated ER diagram based on concepts we’ve learned in the course.

Primary keys are underlined. Chen, min max, and arrows on lines represent the different cardinalities between entities and their relations. Circles are attributes and squares represent entities. Diamonds are relationships, just like we have learned in the course.

Schema

The entities and relationships have been mapped to relations in the diagram of figure 3. Attributes acting as foreign keys in relation A are marked by an ASCII arrow `->`, where the arrow points to the primary key(s) in relation B. The symbols to the left of each attribute signal whether the attribute can be null or not. When black, they cannot take on the null value, when hollow, the attribute can be null, like the *dateOfDeath* attribute on the *Person* relation, since we cannot know when living actors/directors will die.

Non-trivial considerations

The *Participate* relationship is 3-way due to the fact that many *People* (actors) can have many different roles in different movies, or even multiple roles in one movie. This relationship construct allows us to express both in the database.

Comparison of the previous and current solution

In our first attempt to construct a diagram for the movie database, we used the Enhanced Entity-Relationship (EER) model to construct the relevant information for the database. In this version of our database, we use the Entity-Relationship (ER) model as described in the course.

In this version we include Chen notation and min-max notation to emphasize to type of relations. This is also visualised in form of arrows or no arrows on each connection

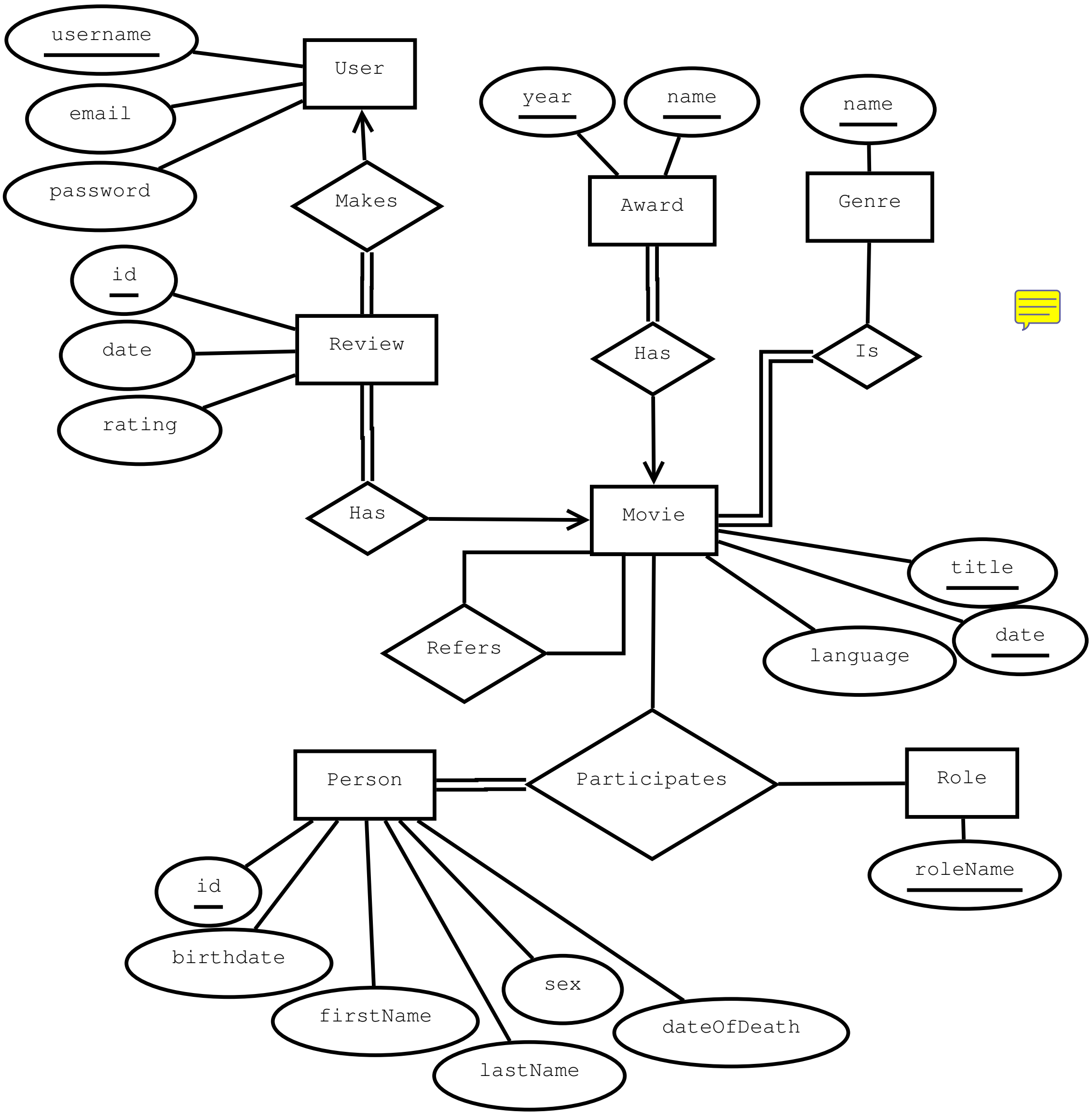

Figure 2: *ER diagram of an IMDB-like website.*

between relations. Another clear difference is that we include total participation for some of the relations. Actually, total and partial participation is expressed as identifying and non-identifying relations in the EER model. This is not covered in the course. We could also have included weak entities, but we did not find any which should be marked as weak.

We have removed redundancy, because an actor can also be a director in movies. We have introduced a relation called Role in which it is clear which role a person has in a given movie. We also considered an ISA relation between the roles in a movie. We chose not to use it, because there is nothing different between an actor and a director.

We have only included the necessary primary keys. If there was no need for a unique id, we have not included one.

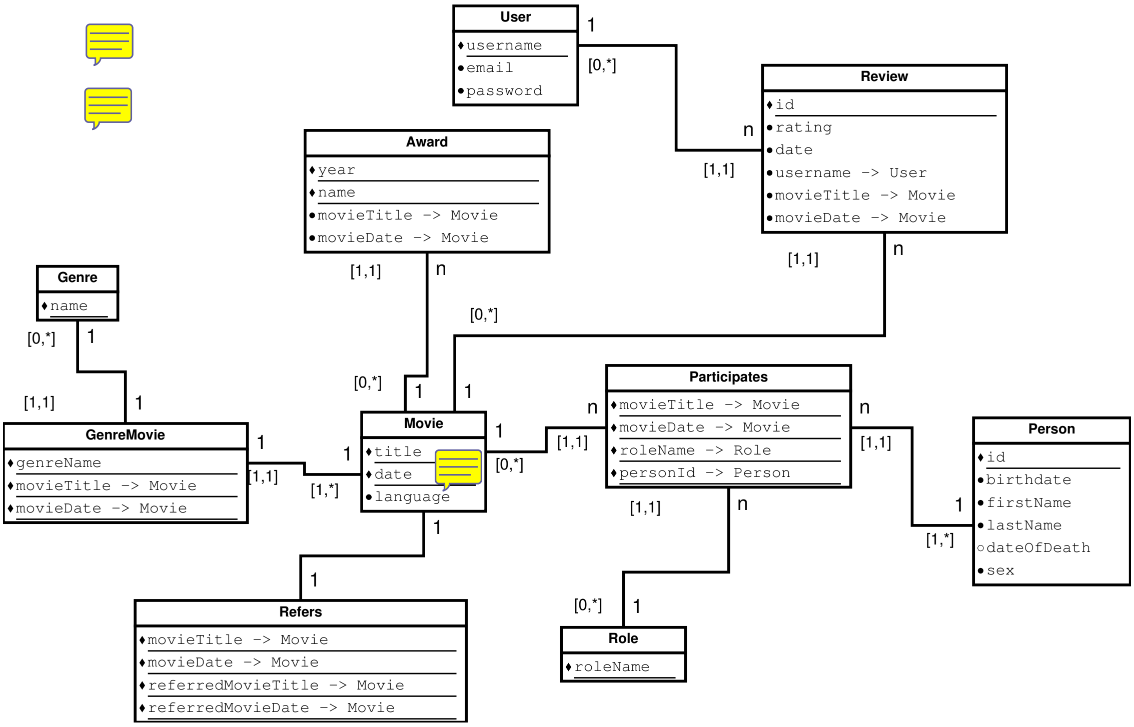

Figure 3: *The schema, i.e. mapped relations of an IMDB-like website.*