Lab 2 – Entity-Relation Diagram Design

This is a two-week lab

The objective of this lab is to make your project descriptions from Lab 1 more concrete and refined and turned into an Entity-Relation (ER) diagram. An ER diagram is a systematic tool for representing and describing the data elements that are available in a domain. For instance, it will help us create a data model of all the entities in my movie database application and will also explicitly define how the entities are related to each other.

Overview

An ER diagram has 3 components:

* Entities: An entity is a thing. In business domain terms, it’s a concept or glossary-level term. In relational database terms, as you will see later in this term, it is a table.
* Relationships: The real insight from this type of diagram comes when we see how entities relate to one another, or relationships. Relationships can be thought of as verbs that link two or more nouns.
* Attributes: For each entity, there can be more than one attribute. Attributes provide detailed information about the entity.

Steps

In order to create an ER diagram, there are three main steps that correspond to the identification of the above three components:

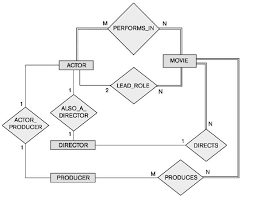
* Create boxes for each entity or primary business concept relevant to your model.
* Model the relationships between each by drawing lines to connect related entities. Label the relationships using verbs or a numeric notation.
* Identify relevant attributes for each entity. For a conceptual model, focus on the most important attributes. As your model evolves, make your attribute lists more specific.

The above steps identify entities, attributes, and relationships from your project description. It is also necessary to identify the cardinality ratios of the relationships found. For example, Movie and Director are two entities from my project description. Director has an attribute Name. There is also a relationship between directors and movies in that directors produce movies. This relationship is one-to-many because one director can make many movies in his lifetime.

When looking and thinking about identifying entities and attributes for your problem domain, you may want to consider the following information:

* Entities and attributes are often represented as nouns in our English writing, e.g. movie, director, etc.
* Attribute values are atomic (year) but entities are a collection of several atomic attributes (movie)
* Attributes cannot participate in relationships, but entities are engaged in relationships.
* Relationships are often described using verbs in our descriptions. So some of the verbs could be our potential relations. For instance, in ‘a director may produce many movies’ produce would be the relationship between director and movie entities.

Once you have identified your entities, attributes and relations, you can draw the corresponding ER diagram. In general, entities are drawn using rectangles, relations using diamonds and attributes using ovals.



Your model needs to have at least 5 entities, 5 relations and 20 attributes. Specify the cardinality of your relations. Ensure that you convert each of your n-ary relations into binary relations.

**Deliverables**

You should complete the steps described above. You can use ERDPlus.com (Refer to the demo we had in the class) for completing your ER diagrams. Then, you will prepare and submit the results in one single zip file (**YourName\_Lab2.zip**) containing the following item:

* A lab report document: The lab report should be prepared using a word processor, and should be stored as a single PDF file. This PDF document should be named as follows: **YourName\_Lab2\_Report.pdf**. This is what should be included in your PDF file:

1. Your name and student number
2. A short (one line) description for each of your entities, attributes and relationships.
3. An ERD diagram that consists of your entities, relations and attributes. Your model needs to have at least 5 entities, 5 relations and 20 attributes. You will need to also clearly specify the cardinality of the relations.