Section 601.615 - Database Final Project Phase I

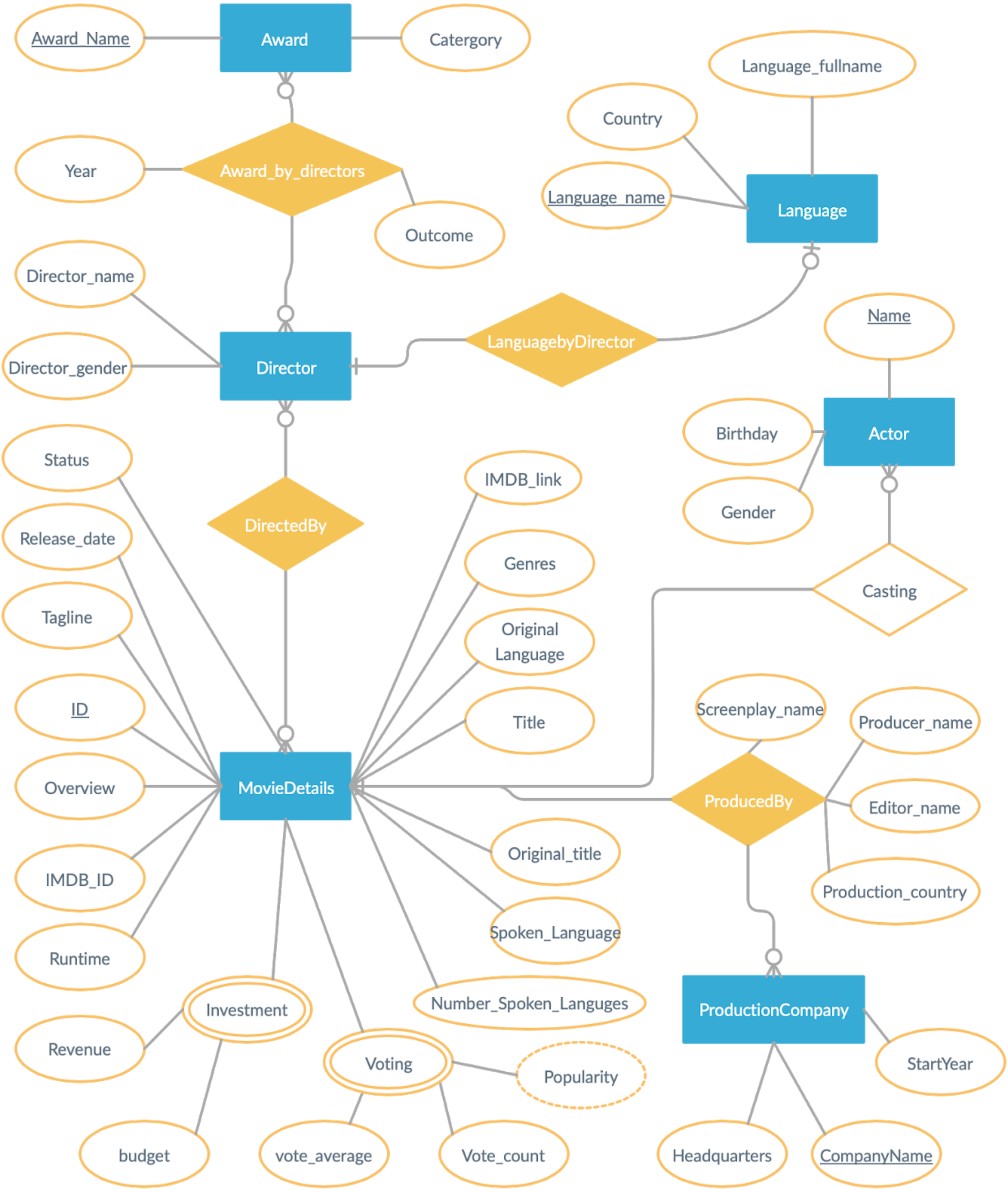
1. **Teammates: Feng Zhang, Boyang Zhang**
2. **Briefly describe your target domain**

   Our goal is to build a movie database that has around 350K movies, ranging from the end of 19th century to August 2017. Users can use this database to find movies that is best aligned with their interests.

3. **List of questions that you would like to answer (at least 15 questions)**
   a. List the number of movies in each genre and their average rate.
   b. List total number of movies for each genre in each country in the year 2017.
   c. List top 5 highest rated movies in each genre and their average rate.
   d. List the director, actor/actress, movie name, genre, production company of top 10 highest rated movies.
   e. List all the “Best Director Award” winning director and total number of movies that they directed in that year.
   f. List the genre, director, actor/actress of most popular movies in each year.
   g. For each user that has ever rated a movie in year 2016, list the movie name and genre of highest rated movie they rated and recommend top 5 movies from that genre in year 2017.
   h. List sci-fi movies produced before 2010 with rating over 8.0
   i. List the movies with the highest rating in each genre in recent 10 years.
   j. List the movies with the highest rating produced in each country.
   k. List movies directed by Christopher Nolan that is not sci-fi and receives at least one award before year 2011.
   l. List the female director who has received the largest number of awards and her language is not English.
   m. List the actor or actress who plays the largest number of roles in the database and was born in America.
   n. List movies that received at least one reward but it's revenue is less than the investment.
   o. List the actors/actresses who played more than 100 movies in the database.
   p. List the production company whose movies receive the highest average rating.

4. Design and show a relational data model

ER Model



Relational form (Tabular)

AwardByDirector	director_name	ceremony	year	outcome		
	Aki KaurismÃ¤ki	ACCEC Awards	2004	Won		
Casting	id	actor_actress_name	gender			
	2	Turo Pajala	0			
	2	Susanna Haavisto	0			
	2	Matti PellonpÃ¤Ã¤	2			
	2	Eetu Hilkamo	0			
Director	id	director_name	director_gender			
	2	Aki KaurismÃ¤ki	2			
ProducedBy	id	producer_name	editor_name	screenplay_name	production_company	production_country
	2	Aki KaurismÃ¤ki	Raija Talvio	Aki KaurismÃ¤ki	Villealfa Filmproduction (	Finland
Language	Language	Country				
	fi	FIN				
LanguageByDirector	director_name	original_language	language_fullname			
	Cheung Chi-Sing	en	English			
	Cheung Chi-Sing	en	English			
Movies	id	genres	imdb_id	imdb_link	original_language	original_title
	2	Drama|Crime	tt0094675	https://www.imdb.com/title/tt0094675/	fi	Ariel
Investment	id	budget	revenue			
	2	unknown	unknown			
Votings	id	vote_average	vote_count	popularity		
	2	7.1	40	0.823904		
CountryInfo	Country	Country_Fullname				
	FIN	Finland				
Award	Award_Name	Category				
	ACCEC Awards	ACCEC Awards				
Actor	Name	Gender	Birthday	Country		
	Anthony Hopkins	Male	1937	England		
ProducationCompany	CompanyName	FoundYear	Headquarters			
	UniversalPictures	1912	Universal City			
DirectedBy	ID	Director				
	324901	Nolan				

Sample queries for creating table:

```
drop table if exists AwardByDirector;
create table AwardByDirector (
    Director_name    VARCHAR(50) NOT NULL PRIMARY KEY,
    Ceremony         VARCHAR(20),
    Year             INTEGER,
    Outcome          VARCHAR(10)
);

drop table if exists Casting;
create table Casting (
    ID               INTEGER NOT NULL PRIMARY KEY,
    Actor_name       VARCHAR(30) NOT NULL,
    Gender           VARCHAR(5)
);

drop table if exists Director;
create table Director (
    Director_name   VARCHAR(30) NOT NULL PRIMARY KEY,
    Director_gender VARCHAR(5)
);
```

5. **A set of SQL statements that implements a representative sample of your target queries**

   Here, we present 5 SQL statements:

   a. List the female director who has received the largest number of awards and her language is not English.

```
select A.director_name from(
(select * from Director
where director_gender=1) as W
inner join
(select * from AwardByDirector
where outcome="Won") as A on A. director_name = W.director_name
inner join
(select * from LanguageByDirector
where origianl_language <> "en") as B on A. director_name = B. director_name)
group by A.director_name
having count(distinct ceremony, year) =
        (select count(distinct ceremony, year) as count from(
        (select * from Director
        where director_gender=1) as W
        inner join
        (select * from AwardByDirector
        where outcome="Won") as A on A. director_name = W.director_name
        inner join
        (select * from LanguageByDirector
        where origianl_language <> "en") as B on A. director_name = B. director_name)
        group by A.director_name
        order by count(distinct ceremony, year) desc limit 1)
```

b. List all the “Best Director Award” winning director and total number of movies that they directed in that year.

```
select director_name, count(distinct id) as total_movies from
((select director_name from AwardByDirector
where outcome = "Won" and category = "Best Director Award") as A
inner join Director as D on A.director_name = D.director_name
inner join (select * from MovieDetails as M on D.id = M.id where YEAR(release_date) =
        select year from AwardByDirector
        where outcome = "Won" and category = "Best Director Award")))
group by A.director_name
```

c. List the best sci fi movies produced by each country before 2005.

```
SELECT Movies.original_title, production_country, vote_average
FROM movies INNER JOIN Produceby ON movies.id = Produceby.id
INNER JOIN voting ON movies.id = voting.id
WHERE genres = "sci-fi" and vote_average = (
        SELECT MAX(Vote_average)
        FROM movies INNER JOIN Produceby ON movies.id = produceby.id
        INNER JOIN voting ON movies.id = voting.id
        WHERE genres = "sci-fi"
        GROUP BY production_country) AS bestMovies;
```

d. List Japanes movies that has a rating higher than the average rating of Japanese movies but it's revenue is less than the investment.

```
SELECT Movies.original_title, vote_average
FROM Movies INNER JOIN Produceby ON movies.id = Produceby.id
INNER JOIN voting ON movies.id = voting.id
WHERE production_country = "Japan" and vote_average > (
        SELECT MEAN(Vote_average)
FROM movies INNER JOIN Produceby ON movies.id = produceby.id
INNER JOIN voting ON movies.id = voting.id
WHERE production_country = "Japan") AS Average_score;
```

e. List the movies directed by female directors who doesn't speak English and the movie's voting is over 8.0 with over 500 hundreds people voting for it and the popularity is over 0.8. And list the corresponding directors.

```
SELECT Movies.original_title, Director_name
FROM Movies INNER JOIN voting ON movies.id = voting.id
INNER JOIN Directedby ON Directedby.id = movies.id
INNER JOIN LanguageByDirector ON director = Directedby.name
INNER JOIN Director ON Director_name = Directedby.Director
INNER JOIN Voting ON Voting.id = Movies.ID
WHERE Director_gender = "Female" and language_fullname <> "English" AND
Vote_average > 8.0 AND Vote_count > 500 AND Popularity > 0.8;
```

6. **A plan for how to load the database with values**
   We plan to download dataset from either Kaggle (https://www.kaggle.com/stephanerappeneau/350-000-movies-from-themoviedborg) or from public available datasource (such as http://www.omdbapi.com/ or https://www.themoviedb.org/documentation/api).

7. **Briefly describe the form/type of output or result**
   A website to allow users can either specify their interest (i.e. by movie genre) and view top recommended movies (ordered by average voting rate) by our database or query detailed information for movie(s).

8. **Specialized or advanced topics (major in one and minor in one specialization)**
   a. Advanced SQL topics (triggers)
   b. Complex data extraction issues from online data sources