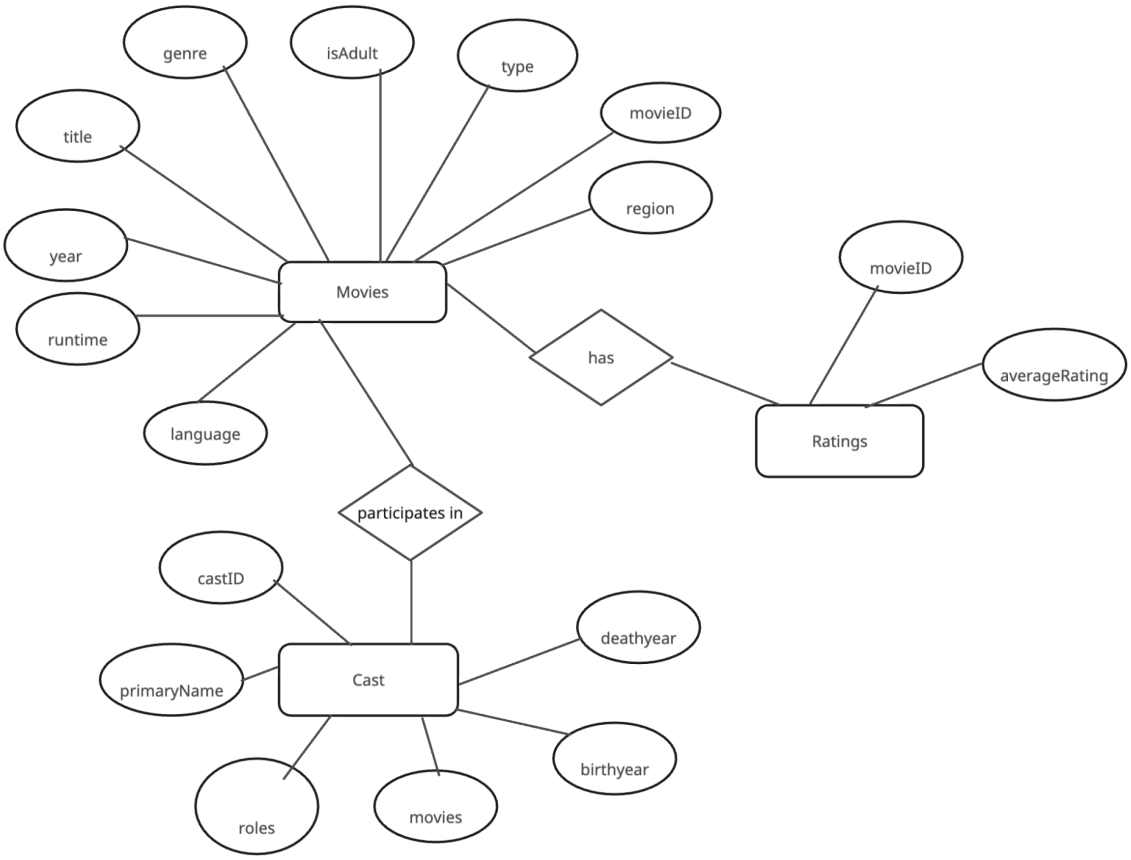
Project: Movie Recommendation System

Authors	Admire Madyira, Takudzwa Mhuru, and Josue Estin
Instructor	Professor Matteo

Final ER Schema



Final Relational Schema in Normal Form

```
-- Create movies table
CREATE TABLE movies (
    movieID INT PRIMARY KEY,
    genre VARCHAR(255),
    runtime INT,
    region VARCHAR(255),
    isAdult BOOLEAN,
    language VARCHAR(255),
    year INT,
    title VARCHAR(255),
   Type VARCHAR(255),
);


-- Create cast table
CREATE TABLE cast (
    castID INT PRIMARY KEY,
    movies VARCHAR(255),
    deathyear DATE,
```

```
    birthyear DATE,
    primaryName VARCHAR(255)
   roles VARCHAR(255)
);
-- Create ratings table
CREATE TABLE ratings (
    movieID INT,
    averageRating FLOAT,
    FOREIGN KEY (movieID) REFERENCES movies(movieID)
);
-- Create participates table to represent the many-to-many relationship between movies and cast members
CREATE TABLE participates (
    movieID INT,
    castID INT,
    PRIMARY KEY (movieID, castID),
    FOREIGN KEY (movieID) REFERENCES movies(movieID),
    FOREIGN KEY (castID) REFERENCES cast(castID)
);
```

Normal Form

- ***First Normal Form (1NF)***: The given schema is already in 1NF because it doesn't contain repeating groups, and each column holds atomic values.
- ***Second Normal Form (2NF)***: The schema is in 2NF if there are no partial dependencies on any candidate key. In this schema, the primary key for the movies table is movieID, and the attributes (genre, runtime, region, isAdult, language, startYear, title, endYear) are fully functionally dependent on the entire primary key. Therefore, it is in 2NF.
- ***Third Normal Form (3NF)***: The schema is in 3NF if there are no transitive dependencies. There are no transitive dependencies in this schema, so it is in 3NF as well.
- ***Summary***: The given schema is already in at least 3NF. There is no need for further normalization based