IT-University of Copenhagen
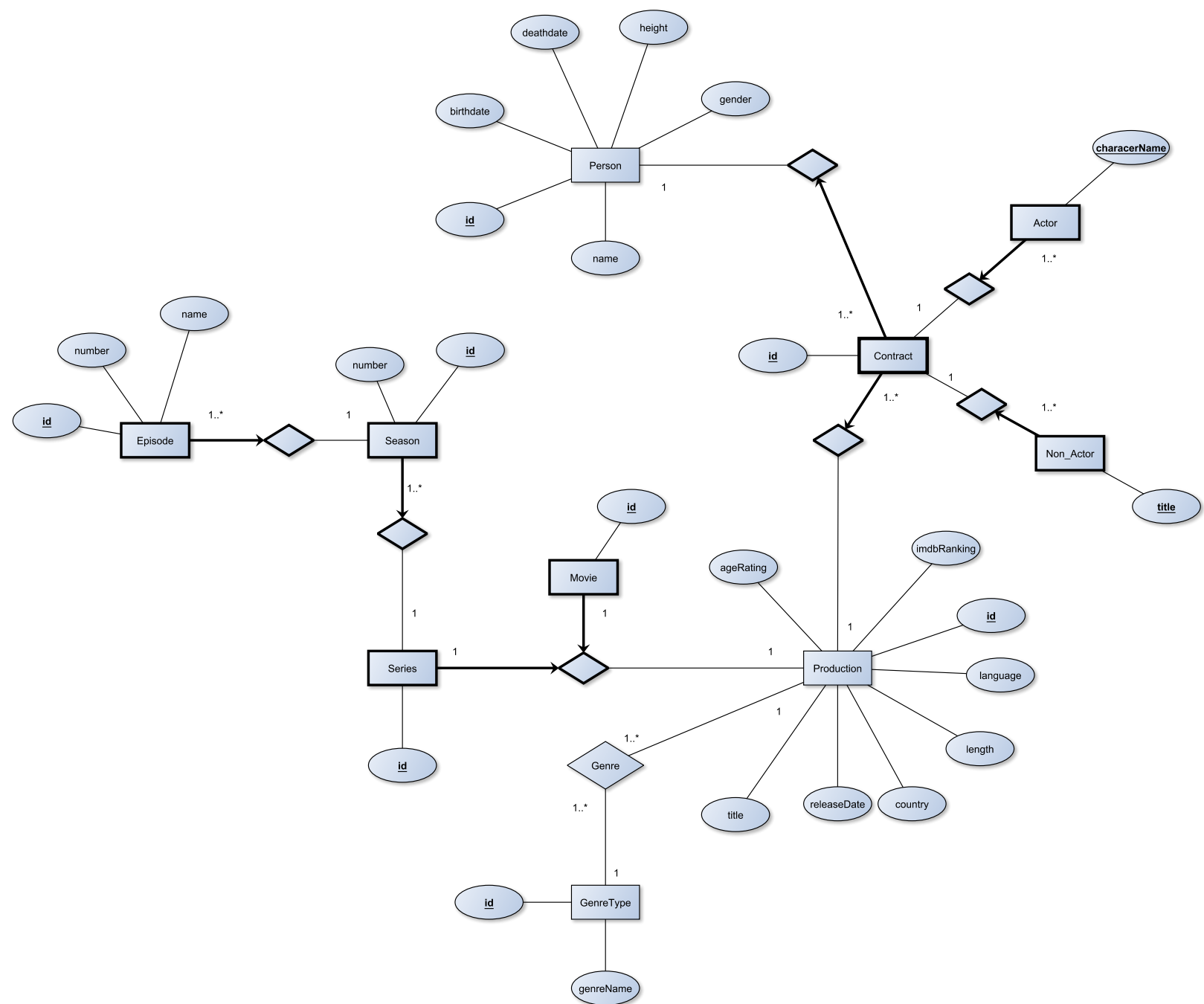
BIDD

Assignment 01

Group: Godfather

Alekxander Binderup(amab@itu.dk)

Kenneth Ry Ulrik(kulr@itu.dk)

Malthe Kirkro(maek@itu.dk)

Mikkel Gaub(mikg@itu.dk)

Omar Khan(omsh@itu.dk)

Entity-Relationship Diagram

Person entity:
- deathdate
- height
- birthdate
- gender
- **id** (key)
- name

Actor entity:
- **characerName** (key)

Contract entity:
- **id** (key)

Non_Actor entity:
- **title** (key)

Episode entity:
- number
- name
- **id** (key)

Season entity:
- number
- **id** (key)

Movie entity:
- **id** (key)

Series entity:
- **id** (key)

Production entity:
- ageRating
- imdbRanking
- **id** (key)
- language
- length
- title
- releaseDate
- country

Genre relationship

GenreType entity:
- **id** (key)
- genreName

Relationships and cardinalities

- Person (1) — Contract (1..*)
- Actor (1..*) — Contract (1)
- Non_Actor (1..*) — Contract (1)
- Contract (1..*) — Production (1)
- Episode (1..*) — Season (1)
- Season (1..*) — Series (1)
- Series (1) — Production (1)
- Movie (1) — Production (1)
- Production (1) — Genre (1..*)
- Genre (1..*) — GenreType (1)

1 Assumptions and choices

- Any production is assumed to have a title.

- A person has a name.

- A person is either a man or a woman, represented as a value of either 1 or 0 respectively.

- A contract is made between a single person and a single production.

- A person can have several different roles within a production through a contract. These roles are defined as either an actor or not an actor (ex. writer or director).

- A production can have more than one genre.

- A production is either a movie or a series.

- A series consists of one or more seasons, which consists of one or more episodes.

2 Analysis of DBMS design

Instead of going through every table in the SQL-Schema we have decided to discuss the areas that we believe could seem to interfere with the 'Boyce Codd' normal form.

2.1 Genre

The relationship between the tables GenreType and Production is done with the Genre relationship, which contains references to the aforementioned tables' primary keys, as foreign key constraints. This could seem redundant as a lot of GenreTypes will reoccur in the Genre table but the alternative, which is not having the "Genre" relationship, would force us to repeat genre names multiple times as well as not allowing us to look up the genre types that can be associated with a production. Having the relationship adds this possibility and replaces the duplication of genre names with id values.

2.2 Production

It is possible that multiple attributes such as the 'country' attribute in the relation "Production" should be treated as we treated GenreType, but since it would involve the creation of several additional tables, which would exceed the constraints of the assignment(maximum 15 tables) we have decided to showcase GenreType as an example of optimizing the schema.

2.3 Inheritance

The combination of the relations; Production, Movie, Series (hereunder Season and Episode), contributes to avoid a major amount of redundancy, as the alternative of storing loads of information in each table would introduce. This is solved by containing all common data fields in the Production relation, and having the previously mentioned Movie and Series tables relate to a single entry in the Production relation. Following this, Season and Episode contain little to no additional information, albeit a single Episode will usually have an individual name, setting it apart from the other episodes produced within the series.