



## Assumptions

### User – Favorite Relationship (Likes)

User\_id is a foreign key in the favorites table. The relationship cardinality for the Likes relationship is **one-to-many**. However, the relationship cardinality for the Favorite-User relationship is **one-to-one**. This is true because every user can have (Like) multiple favorite movies. But each favorite entry belongs to only one user. A user may also have no favorite movies/tv shows.

### User-Search Relationship

User\_id is a foreign key in the Search table. The relationship cardinality for the User-Search relationship is **one-to-many**. However, the relationship cardinality for the Search-User relationship is **one-to-one**. This is true because every user can have multiple searches. But each search entry belongs to only one user. A user may also have no searches.

### Search-Movie Relationship

Movie\_id is a foreign key in the Search table. The relationship cardinality for the Search-Movie relationship is **one-to-one**. Since only searches that result in a movie “hit” are logged. However, the relationship cardinality for the Movie-Search relationship is **one-to-many**, as a movie may be searched for multiple times (or not at all) by different or the same user.

### Favorite-Movie Relationship (Is a)

Movie\_id is a foreign key in the Favorite table. The relationship cardinality for the Favorite-Movie relationship is **one-to-one**. This is because only movies in the movie table can be favorited. The relationship cardinality for the Movie-Favorite relationship is **one-to-many**, because a movie can be a favorite for more than one user, or not a favorite at all.

### Movie-Availability Relationship (Available in)

Movie\_id is a foreign key in the Availability table. The cardinality for the Movie-Availability relationship is **one-to-many** with the stipulation that the movie can only be available if it is in the movie table. The inverse relationship, Availability-Movie relationship has a cardinality of **one-to-one**.

### Availability-Country Relationship (Named)

Country\_id is a foreign key in the Availability table. The relationship cardinality for the Availability-Country relationship is **one-to-many**, and any movie that is available, by definition, needs to have a country. The inverse relation also has a **one-to-many** cardinality. However, here it is not necessary that every country has a Netflix catalogue.

### Movie-Cast Relationship (Has)

Movie\_id is a foreign key in the Cast table. The cardinality for the Movie-Cast relationship is **one-to-many** since every movie has multiple cast members, and every movie has at least one cast member. The inverse relationship is **one-to-one** since the crew member plays their role only in a specific movie (counting sequels/prequels as separate).

### Movie Genre-Movie Relationship (Has)

Movie\_id is a foreign key in the Movie Genre table. The cardinality of the Movie Genre-Movie Relationship is **one-to-many**. A genre is not unique to a movie, and is only in the table if there is at least one movie of that genre present. The Movie-Movie Genre relationship is also **one-to-many** since a movie may have more than one genre but must have at least one.

### Logical Design (Relational Schema)

**Movie**(movie\_id:INT [PK], name:VARCHAR(255), media\_type:VARCHAR(255), release\_dt:DATE, synopsis:TEXT)

**Availability**(availability\_id:INT [PK], movie\_id:INT [FK to Movie.movie\_id], country\_id:int [FK to Country.country\_id])

**Country**(country\_id:INT [PK], country\_name:VARCHAR(255))

**Cast**(cast\_id:INT [PK], movie\_id:INT [FK to Movie.movie\_id], , role:VARCHAR(50), name:VARCHAR(255))

**Genre**(genre\_id:INT [PK], movie\_id:INT [FK to Movie.movie\_id], genre:VARCHAR(255))

**User**(user\_id:INT [PK], username:VARCHAR(50), email:VARCHAR(250))

**Favorite**(favorite\_id:INT [PK], user\_id:INT [FK to User.user\_id], movie\_id[FK to Movie.movie\_id])

**Search**(search\_id:INT [PK], user\_id:INT [FK to User.user\_id], search\_text:VARCHAR(255), movie\_id:INT [FK to Movie.movie\_id])