Maryam Soleimani 401222075

Introduction

In this project I designed database for a Media Streaming Service. The main goal is to create a system that manages data related to user profiles, subscription details, payment histories, and personalized watch lists. Additionally I want to ensure that media assets including both movies and series are organized and easily accessible complete with relevant information such as genre, production year, ratings, and associated production companies.

The Entities and the relations are accurately visible in ER diagram that is attached.

A few notes to consider and additional assumptions:

> In subscription entity I considered sub_tier a composite attr consists of two plans : 1 month and 12 month. In which user can choose to pay for 1 month and use the service or 12 month. The only point here is that it's not possible to have both plans active at the same time.

Here there is detailed information related to Entities and Relationships:

USER:

1. **User_id**
    - **Data Type: Integer (or UUID for a unique identifier)**
    - **Constraints:**
        - **Primary Key**
        - **Not Null**
2. **User_name**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**
        - **Unique**
3. **Age**
    - **Data Type: Integer**
    - **Constraints:**
        - **Not Null**
        - **Check: Age should be greater than or equal to 10**
4. **Email**
    - **Data Type: Varchar**

- **Constraints:**
    - **Not Null**
    - **Unique**
    - **Format Check: we can implement a regex pattern check for valid format.**

5. **Password**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**
        - **Minimum Length: we can enforce a minimum length**
        - **unique**

SUBSCRIPTION:

1. **subscription_id**
    - **Data Type**: Integer (or UUID for uniqueness)
    - **Constraints**:
        - **Primary Key**
        - **Not Null**
2. **start_date**
    - **Data Type**: Date
    - **Constraints**:
        - **Not Null**
        - **Check**: Should not be in the future (CHECK (start_date <= CURRENT_DATE))
3. **end_date**
    - **Data Type**: Date
    - **Constraints**:
        - **Not Null**
        - **Check**: Should be greater than start_date (CHECK (end_date >= start_date))
4. 1_month_plan
    - Data type: Boolean

5. 12_month_plan:
    - Data type: Boolean

Note: between 4 and 5 only one is allowed and after the duration it can be updated but both at the same time is not possible (it can be considered as a constraint)

Data types are Boolean because there are only 2 possibilities for each (a user chooses that or not so there is no other possibilities)

6. **user_id**
    - **Data Type**: Integer (or UUID for uniqueness) (it must be the same type with the User entity)
    - **Constraints**:
        - **Foreign Key**: (references User_id in the User table)
        - **Not Null**

7. **payment_id**
    - **Data Type**: Integer (it must be the same type with the Payment entity)
    - **Constraints**:
        - **Foreign Key**: (references Payment_id in the Payment table)
        - **Not Null**

Relation between USER and SUBSCRIBTION: (has)

One User to Many Subscriptions:

- **A single user can have multiple subscriptions over time (they might switch between different plans).**
- **Each subscription is linked to one specific user.**

Participation:

1. **User Participation:**
    - **Participation Type: Total Participation**
    - **Description: Every user must have at least one subscription. In this case, the participation of the User entity is total because it is expected that all users will engage in at least one subscription.**

      This means that there cannot be a user in the system without a corresponding subscription record. If a user is created, they should immediately or eventually have a subscription.

2. **Subscription Participation:**
    - **Participation Type: Total Participation**
    - **Description: every subscription must be linked to a user.**

      This means that while a user must have at least one subscription, a subscription cannot exist without being associated with a user. However, it's possible to have multiple subscriptions for a single user.

PAYMENT_HISTORY:

1. **Payment_id**
    - **Data Type**: Integer
    - **Constraints**:
        - Primary Key
        - Not Null
2. **Date**
    - **Data Type**: DATETIME
    - **Constraints**:
        - Not Null
3. **Amount**
    - **Data Type**: DECIMAL(10, 2)
    - **Constraints**:
        - Not Null

Relation between PAYMENT_HISTORY and SUBSCRIBTION: (has)

1 to 1:

Each payment history is associated with exactly one subscription and vice versa

The participation for both is total because there is no payment history without associated subscription and vice versa.

MEDIA:

1. **Media_id**
    - **Data Type: Integer (or UUID for uniqueness)**
    - **Constraints:**
        - **Primary Key**
        - **Not Null**
2. **Genre**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**
3. **Title**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**
4. **Rate**
    - **Data Type: Integer**
    - **Constraints:**
        - **Not Null**
        - **Check: Rate should be within a valid range (0 to 10)**
5. **Director**
    - **Data Type: Varchar**

- **Constraints:**
    - **Not Null**

6. **Location_id**
    - **Data Type: Integer (or UUID)**
    - **Constraints:**
        - **Foreign Key: (references location_id in the Storage_Location table)**
        - **Not Null**

7. **Company_id**
    - **Data Type: Integer (or UUID)**
    - **Constraints:**
        - **Foreign Key: (references company_id in the Company table)**
        - **Not Null**
8. **List_of_actors (Multivalued Attribute)**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Can contain multiple entries (a comma-separated list)**
        - **Not null**
9. **List_of_producers (Multivalued Attribute)**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Can contain multiple entries (a comma-separated list)**
        - **Not null**

Note: for simplicity we don't need to consider separate Entities or tables for list_of_actors and list_of_producers because these don't have high level of importance for example we don't really need to keep a record of the age of an actor or his/her hair color or related attributes we only need the name so only a list of names will be enough here.

Note: Media can be either Movie or Series therefor we apply a disjoint here. Media is the parent for Movie and Series entities. So we can conclude that Media and Series both have the same attrs as Media plus some specific for their own.

MOVIE:

1. **Production_year**
    - **Data Type: Integer (or Year)**
    - **Constraints:**
        - **Not Null**
        - **Check: Should be a valid year (greater than a reasonable limit for example the year that the first film was made)**
2. **Run_time**
    - **Data Type: Integer (representing minutes)**
    - **Constraints:**
        - **Not Null**
        - **Check: Should be a positive integer**

SERIES:

1. **Number_of_seasons**
    - **Data Type: Integer**
    - **Constraints:**
        - **Not Null**
        - **Check: Should be a positive integer (greater than or equal to 1)**
2. **Start_date**
    - **Data Type: Date**
    - **Constraints:**
        - **Not Null**
        - **Check: Should not be in the future (CHECK (start_date <= CURRENT_DATE))**
3. **End_date**
    - **Data Type: Date**
    - **Constraints:**
        - **Not Null**
        - **Check: Should be greater than or equal to start_date (CHECK (end_date >= start_date))**

4. **Number_of_episodes**
    - **Data Type: Integer**
    - **Constraints:**
        - **Not Null**
        - **Check: Should be a positive integer (greater than or equal to 1)**

Note: Series(strong entity) has episodes so it's in an identifying relationship with Episode entity (weak entity).

EPISODE:

1. **Episode_no**
    - **Data Type: Integer**
    - **Constraints:**
        - **Not Null**
        - **Check: Should be a positive integer (greater than or equal to 1)**
2. **Episode_name (Partial Key)**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**

Note: Since it's a weak entity for Series it needs a partial key which is derived from the primary key of its strong entity. In this context the choice of Episode_name as the partial key seems reasonable.

STORAGE_LOCATION:

1. **Loc_id**
    - **Data Type: Integer (or UUID for uniqueness)**
    - **Constraints:**
        - **Primary Key**
        - **Not Null**

2. **Path**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**
        - **Check: Should represent a valid file path format**
3. **Server**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**
        - **Check: Should represent a valid server name or IP address**

COMPANY:

1. **Comp_id**
    - **Data Type: Integer (or UUID for uniqueness)**
    - **Constraints:**
        - **Primary Key**
        - **Not Null**
2. **Comp_name**
    - **Data Type: Varchar**
    - **Constraints:**
        - **Not Null**
        - **Check: Should not be empty**
3. **Establish_year**
    - **Data Type: Integer (or Year)**
    - **Constraints:**
        - **Not Null**
        - **Check: Should be a valid year**
4. **Comp_email**
    - **Data Type: Varchar**
    - **Constraints:**

- **Not Null**
- **Check: Should follow a valid email format**

5. **Comp_phone**
    - **Data Type: Varchar (to accommodate different formats)**
    - **Constraints:**
        - **Not Null**
        - **Check: Should follow a valid phone number format**

COMMENT:

1. **Comment _id**
    - **Data Type**: Integer (or UUID for uniqueness)
    - **Constraints**:
        - Primary Key
        - Not Null
2. **Comment**
    - **Data Type**: Varchar (or Text)
    - **Constraints**:
        - Not Null
3. **Media_id** (Foreign Key)
    - **Data Type**: Integer (or UUID, depending on the Media entity)
    - **Constraints**:
        - Foreign Key: (references media_id in the Media table)
        - Not Null
4. **User_id** (Foreign Key)
    - **Data Type**: Integer (or UUID, depending on the User entity)
    - **Constraints**:
        - Foreign Key: (references user_id in the User table)
        - Not Null

User_RATE:

1. rate_id
    - **Data Type**: Integer (or UUID)
    - Constraints:
        - Primary Key
        - Not Null
2. Rate
    - Data Type: Integer
    - Constraints:
        - Check: Should be within a valid range (1 to 10)
        - Before a user can rate a media entity, check if a rating already exists for that user and media combination.
        - If no rating exists, allow the user to submit a rating, which will be stored in the Ratings table.
        - If a rating exists, inform the user that they can only rate once.
        - Not null
3. **Media_id** (Foreign Key)
    - **Data Type**: Integer (or UUID, depending on the Media entity)
    - **Constraints**:
        - Foreign Key: (references media_id in the Media table)
        - Not Null
4. **User_id** (Foreign Key)
    - **Data Type**: Integer (or UUID, depending on the User entity)
    - **Constraints**:
        - Foreign Key: (references user_id in the User table)
        - Not Null

WATCH_LATER_LIST:

1. **Watchlist_id**

- **Data Type**: Integer (or UUID for uniqueness)
- **Constraints**:
    - Primary Key
    - Not Null

2. **Media_id** (Foreign Key)
    - **Data Type**: Integer (or UUID, depending on the Media entity)
    - **Constraints**:
        - Foreign Key: (references media_id in the Media table)
        - Not Null
3. **User_id** (Foreign Key)
    - **Data Type**: Integer (or UUID, depending on the User entity)
    - **Constraints**:
        - Foreign Key: (references user_id in the User table)
        - Not Null

Relationship between USER and **watch_later_list:**

Cardinality: 1-1 : each user can create one watchlist and each watchlist can be created by one user

Participant: There exists users that have no watchlist so the participation is partial. All the watchlists are created by some users so it is total.

Relationship between MEDIA and **watch_later_list:**

Cardinality: M:N : in each watchlist there can be many media and each media can be inside many watchlists

Participant: there exists no such watchlist that Is lack of media so total. There can exist media that is not inside any watchlist so partial.

Relationship between MEDIA and **Comment:**

Cardinality: 1:N : Each Media can have many comments but each comment is special for one media.

Participant: There can exist a media that has no comment so partial. All the comments that are created must be associated with a media so total.

Relationship between User and **Comment:**

Cardinality: 1:N : each User can create multiple comments but each comment is created by one user.

Participation: There exists a user that leaves no comment so partial.

All the comments are created by some user so total.

Relationship between MEDIA and **user_rate:**

Cardinality: 1:N : Each Media can have many rates but each rate is special for one media.

Participant: There can exist a media that has no rate so partial. All the rates that are created must be associated with a media so total.

Relationship between User and **user_rate:**

Cardinality: 1:N : each User can create multiple rates(for different medias, note that each user-media combination can only have one rate) but each rate is created by one user.

Participation: There exists a user that leaves no rate so partial.

All the rates are created by some user so total.

Relationship between User and **Media:**

Cardinality: N:M : Each user can watch multiple media and each media can be watched by multiple users.

Participation: there exists a user that doesn't watch media and there exists a media that isn't watched so both side partial.

Relationship between Media and **company:**

Cardinality: 1:N : each Company can create multiple Media but each Media is belongs to only one company.

Participation: Every media belongs to some company and every company owns some media so both sides are total.

Relationship between Media and **storage_location:**

Cardinality: 1-1 : each Media has one specific location to be saved and each location is accommodated by a specific Media

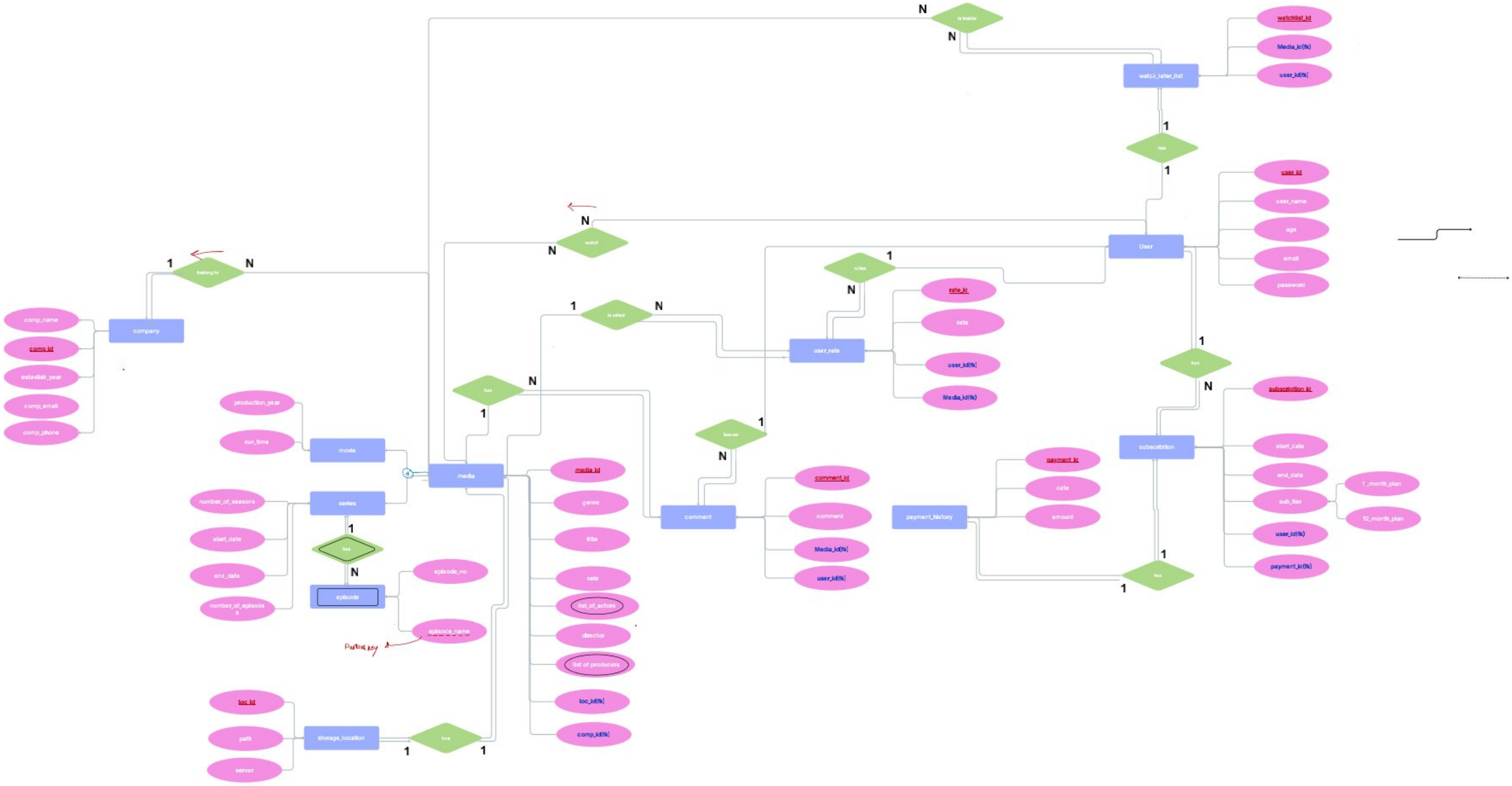
Participant: Both sides total: Every media is saved in a location and in each location is a media that is alredy saved.

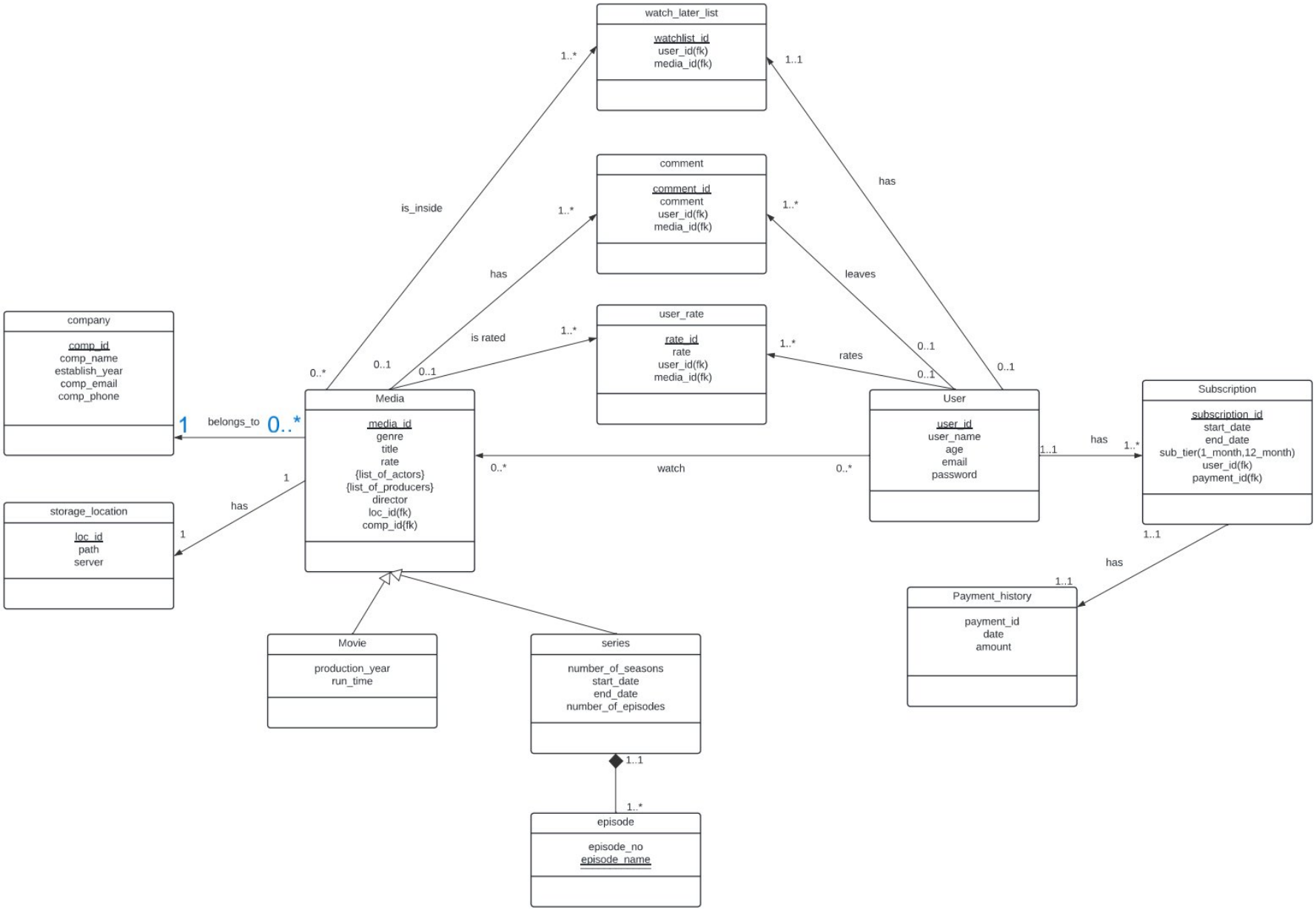
Relationship between Series and **Episode:**

Identifying Rel

Cardinality: 1:N : Each Series has multiple episodes but each episode is associated with 1 series

Participation: both sides total. Every Series has episodes and every episode is associated with a series


N
N
is invite
watch_later_list
watchlist_id
Media_id(fk)
user_id(fk)
1
has
1
user_id
user_name
age
email
password
N
N
watch
User
1
belong to
N
rates
1
N
rate_id
rate
user_id(fk)
Media_id(fk)
1
is rated
N
user_rate
1
has
N
1
leaves
1
N
comment
comment_id
comment
Media_id(fk)
user_id(fk)
payment_history
payment_id
date
amount
1
has
N
subscribtion
subscribtion_id
start_date
end_date
sub_tier
user_id(fk)
payment_id(fk)
1_month_plan
12_month_plan
1
has
1
company
comp_name
comp_id
estavlish_year
comp_email
comp_phone
production_year
run_time
movie
media
media_id
genre
title
rate
list_of_actors
director
list of producers
loc_id(fk)
comp_id(fk)
number_of_seasons
start_date
end_date
number_of_episode
s
series
1
has
N
episode
episode_no
episode_name
Partial key
loc_id
path
server
storage_location
has
1
1


watch_later_list
watchlist_id
user_id(fk)
media_id(fk)
comment
comment_id
comment
user_id(fk)
media_id(fk)
user_rate
rate_id
rate
user_id(fk)
media_id(fk)
company
comp_id
comp_name
establish_year
comp_email
comp_phone
Media
media_id
genre
title
rate
{list_of_actors}
{list_of_producers}
director
loc_id(fk)
comp_id(fk)
User
user_id
user_name
age
email
password
Subscription
subscription_id
start_date
end_date
sub_tier(1_month,12_month)
user_id(fk)
payment_id(fk)
storage_location
loc_id
path
server
Movie
production_year
run_time
series
number_of_seasons
start_date
end_date
number_of_episodes
episode
episode_no
episode_name
Payment_history
payment_id
date
amount
is_inside
has
has
leaves
is rated
rates
belongs_to
watch
has
has
has
1..*
1..1
0..*
0..1
1