



DBMS-> Project

MOVIE MANAGEMENT SYSTEM

Created by:- NAMAN GANDHI (2022UCS1536)

PRIYANSHU KUMAR SAH (2022UCS1519)

ARYAN GUPTA (2022UCS1533)

MUHAMMAD KHIZR KHAN (2022UCS1545)

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION TO PROBLEM AREA

Problem Area:*

1. *Information Overload:*

- With a vast number of movies, TV shows, and streaming options available, users often face information overload when trying to discover content.

2. *Lack of Centralized Information: *

- Movie-related information is dispersed across various platforms and sources, making it challenging for users to find comprehensive details about movies, actors, directors, and related data in one place.

3. *User Engagement and Recommendations: *

- Users often struggle to find personalized recommendations based on their preferences and viewing history. A centralized database can enhance user engagement by providing tailored suggestions.

4. *Franchise and Series Navigation: *

- Navigating through movie franchises or series and understanding the chronological order or relationships between movies can be confusing, especially for newcomers to a particular film series.

5. *Limited Access to Film Industry Data: *

- Filmmakers, researchers, and industry professionals may lack a centralized platform to access comprehensive data on box office performance, awards, and filming locations.

6. *Data Integrity and Reliability: *

- Users may encounter unreliable or outdated information on certain platforms. A centralized database ensures data integrity and provides a reliable source for movie-related information.

1.2 PROBLEM STATEMENT

A movie database is a structured collection of data related to movies, actors, directors, and other aspects of the film industry. It is designed to store, organize, and manage information about movies, their production, and audience engagement. Here are the key components and features typically found in a movie database:

1. Movie Information

- This section of the database includes details about individual movies. It encompasses data such as the movie's title, release date, genre, runtime, plot summary, and production studio. Movie information may also include posters, trailers, and promotional images.

2. Cast and Crew

- The database stores information about the cast and crew involved in making the movies. This includes actors, directors, producers, writers, and other key personnel. It tracks their names, birthdates, nationalities, and biographical details.

3. User Profiles

- User profiles are typically part of a movie database to enable user engagement. Users can create profiles with their usernames, email addresses, and other personal information. These profiles are used for activities like rating and reviewing movies, saving favorite films, and creating watchlists.

4. Movie Reviews and Ratings

- Users can write reviews and provide ratings for movies. The database stores these reviews, along with the user who wrote them, the movie being reviewed, and the corresponding rating. This user-generated content helps others make informed decisions about which movies to watch.

5. Recommendations and Watchlists

- Movie databases often offer features for users to create watchlists of movies they plan to watch and receive recommendations based on their viewing history and preferences. These features use algorithms to suggest relevant films to users.

6. Genres and Categories

- Movies are often categorized into genres or thematic categories. The database includes information about these genres, which is used for sorting and filtering movies. For example, genres like action, comedy, drama, and science fiction are common.

7. Release Information

- This section of the database tracks the release history of movies, including their release dates in different regions, DVD and streaming release dates, and international releases.

8. Box Office Data

- Some movie databases include box office information, such as box office earnings, budget, and financial performance metrics. This data is often used for ranking and analyzing the success of movies.

9. Awards and Nominations

- Information about awards, nominations, and recognitions received by movies, actors, and directors is stored in the database. This includes data on Academy Awards, Golden Globes, and other industry accolades.

10. Trailer and Multimedia Links

- Databases may link to multimedia content like movie trailers, interviews, and promotional videos related to the films.

11. Search and Recommendation Algorithms

- Movie databases employ search and recommendation algorithms to help users discover new movies, explore related content, and find movies that match their preferences.

12. User Interaction History

- To enhance user experiences, databases track user interactions, such as movie views, clicks, and searches, to improve recommendations and personalize content.

1.3 OBJECTIVE OF THE PROJECT

A movie database offers a wide range of uses and benefits, catering to different stakeholders within the film industry and movie enthusiasts. Here are some of the key uses and benefits of a movie database:

1. Movie Discovery

- Users can easily search and discover movies based on various criteria, such as genre, release date, actors, directors, or user ratings. This helps viewers find new movies to watch that match their preferences.

2. User Reviews and Ratings

- Users can read and contribute movie reviews and ratings. This user-generated content helps others make informed decisions about what to watch and provides valuable feedback to filmmakers.

3. Personalized Recommendations

- Movie databases often use recommendation algorithms to suggest movies to users based on their viewing history and preferences. This personalization enhances the user experience and encourages exploration.

4. Watchlists and Tracking

- Users can create watchlists of movies they intend to watch in the future. The database keeps track of these lists and provides reminders or notifications when selected movies become available.

5. Cinematic Information

- It offers comprehensive information about movies, including plot summaries, cast and crew details, release dates, genres, and trailers. This is valuable for both casual moviegoers and film enthusiasts.

6. Film Research and Analysis

- Researchers, academics, and film analysts can use the database to study trends in the film industry, analyze box office data, track awards and nominations, and perform in-depth research on movies and their impact.

7. Industry Insights

- Filmmakers, producers, and industry professionals can use the database to research the competition, understand market trends, and gain insights into the success of previous films.

8. Marketing and Promotion

- Movie studios and streaming platforms can use the database to promote their films, share trailers and promotional content, and target audiences based on user preferences.

9. Box Office Data

- Industry professionals and financial analysts can access box office earnings, budget data, and financial performance metrics to assess the profitability of movies.

10. Historical Data

- Film historians and enthusiasts can use the database to explore the history of cinema, track the evolution of movie-making techniques, and study the careers of actors and directors.

11. Awards and Recognitions

- The database provides information on awards and nominations, helping filmmakers and fans track the success and recognition received by movies, actors, and directors.

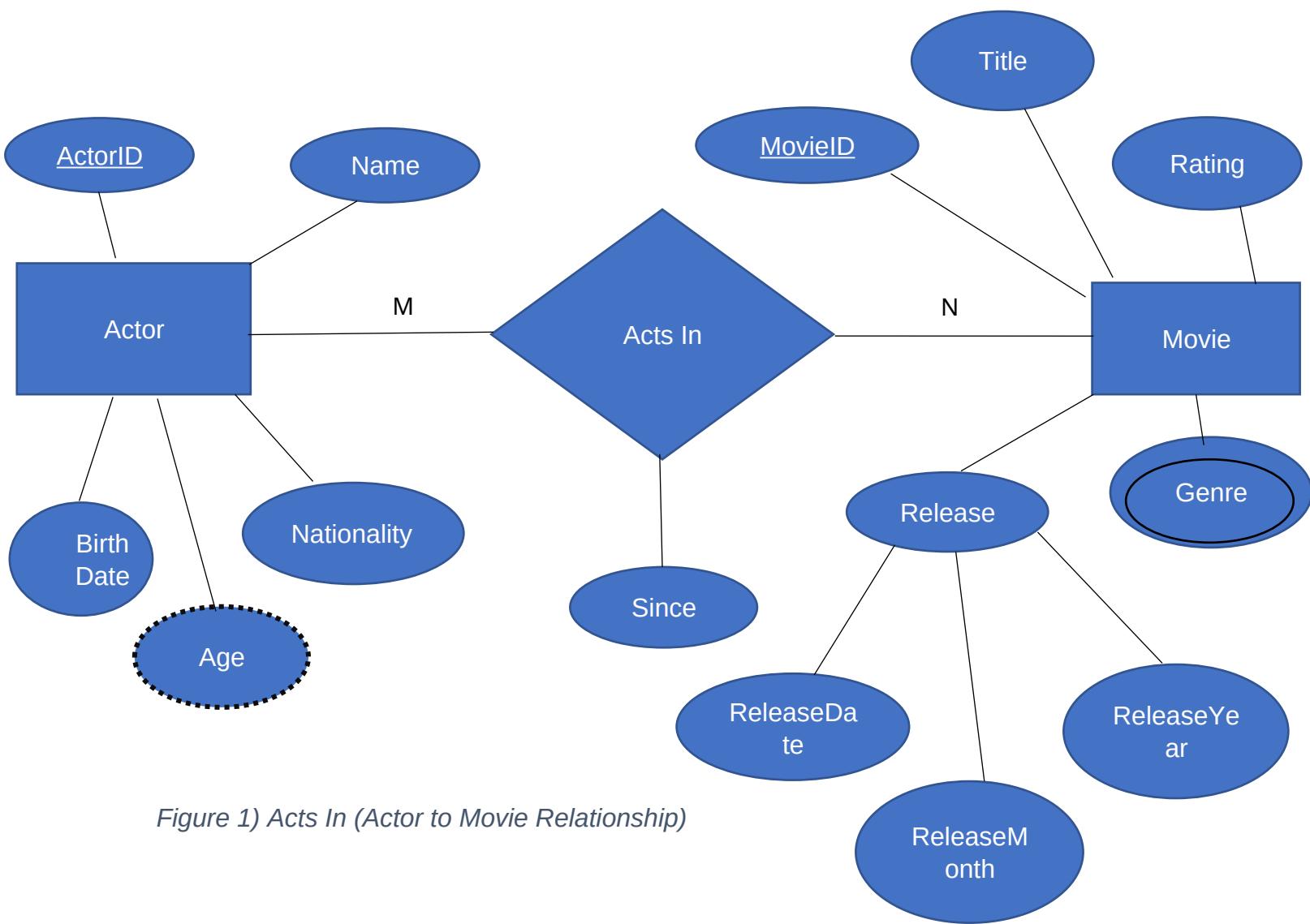
12. Content Curation

- Streaming platforms use movie databases to curate their content libraries, making it easier for users to find movies they want to watch and enjoy.

13. Educational Purposes

- Educational institutions use movie databases for teaching film studies, cinematography, and the history of cinema.

ER Model for MOVIE DATABASE



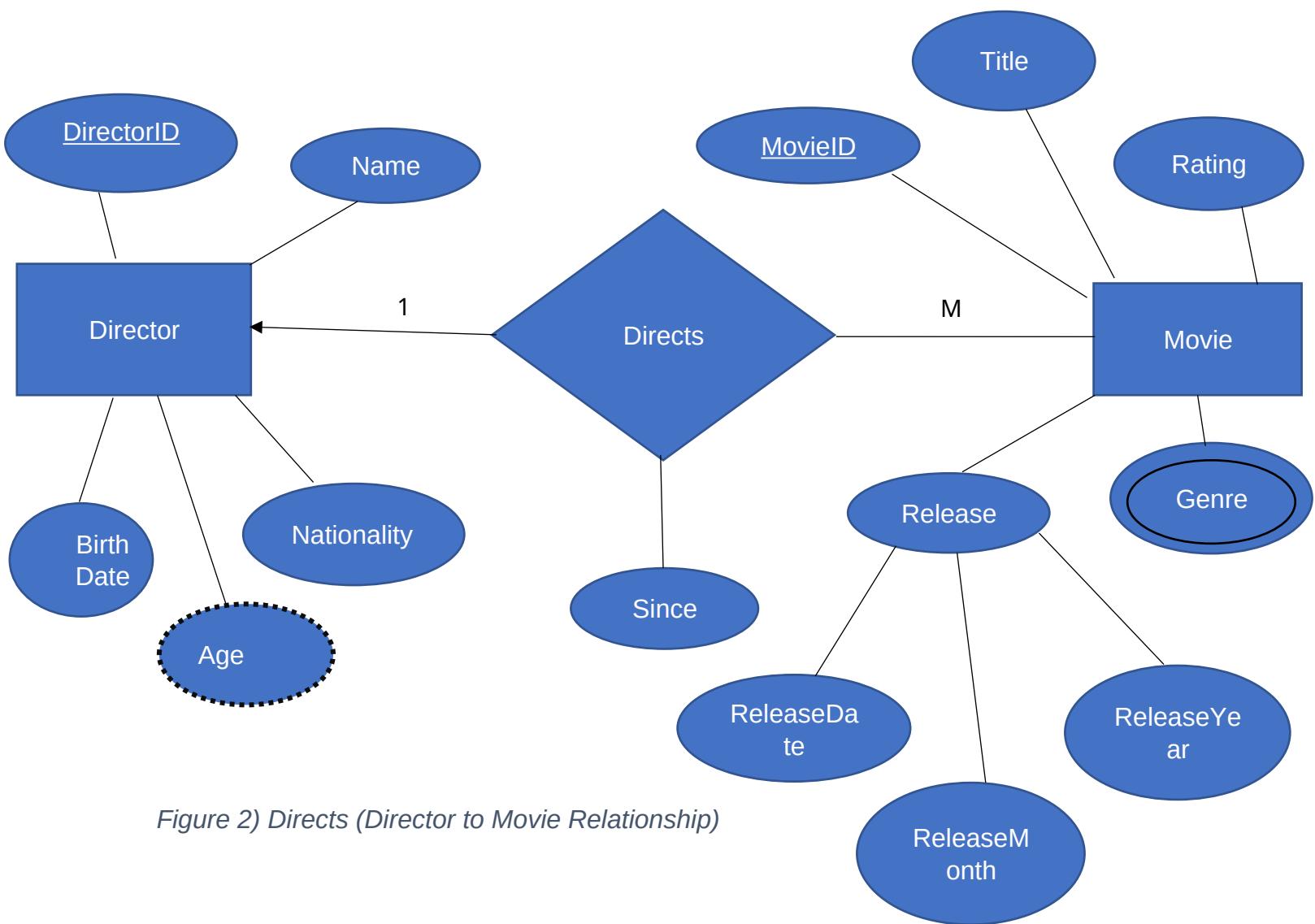


Figure 2) Directs (Director to Movie Relationship)

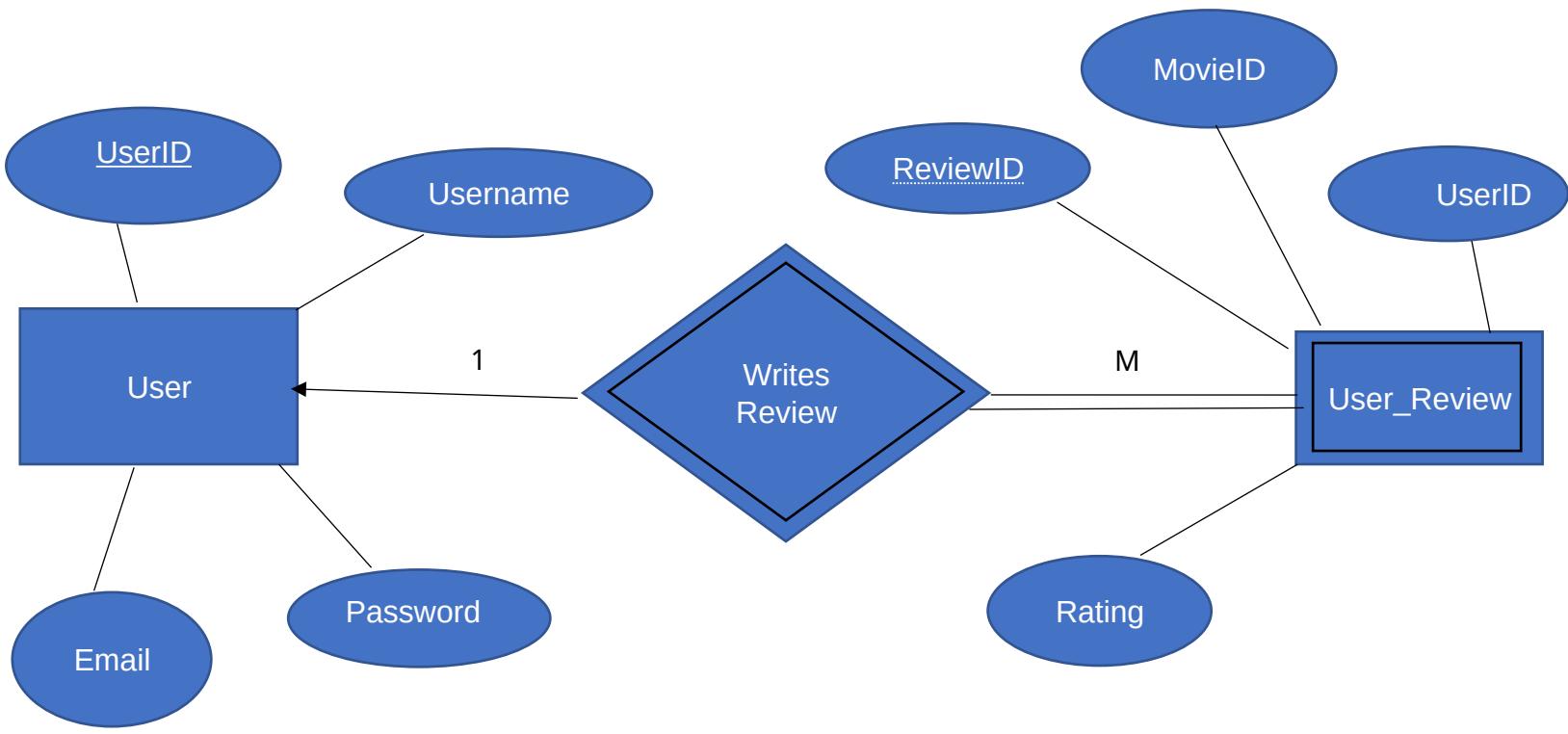


Figure 3) Writes Review (User to User-Review Relationship)

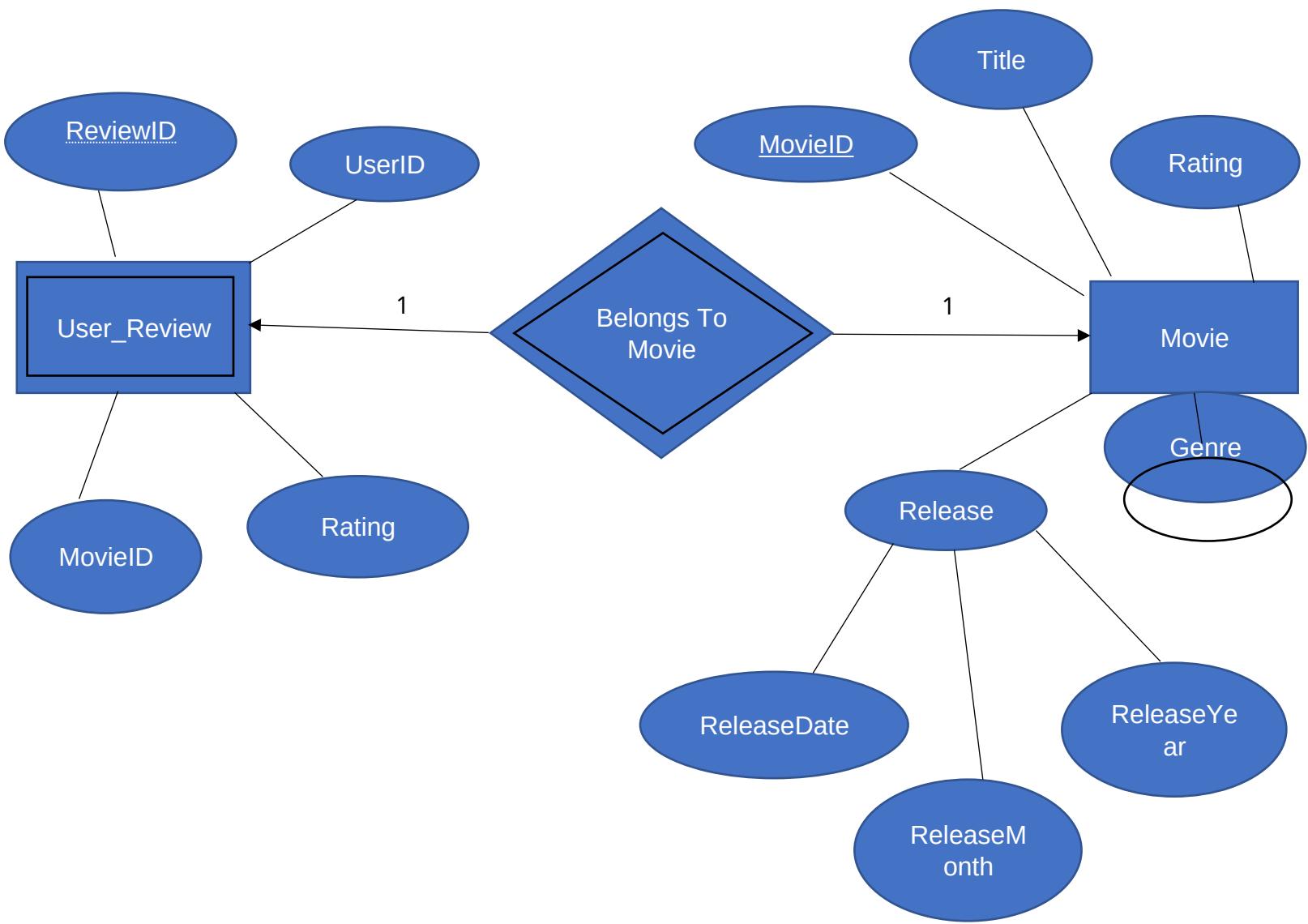


Figure 4) Belongs To Movie (User-Review to Movie Relationship)

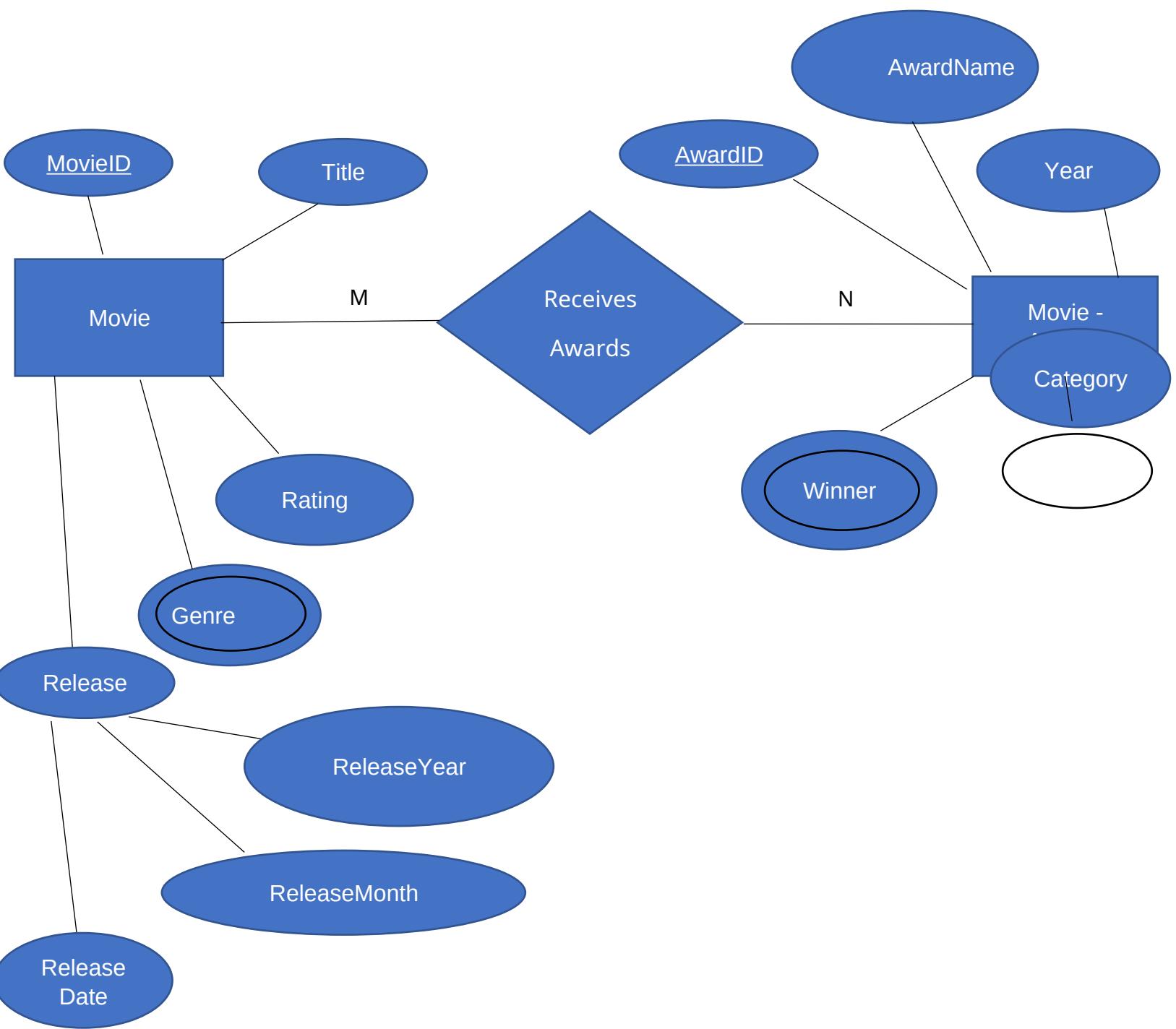


Figure 5) Receives Awards (Movie to Movie Awards Relationship)

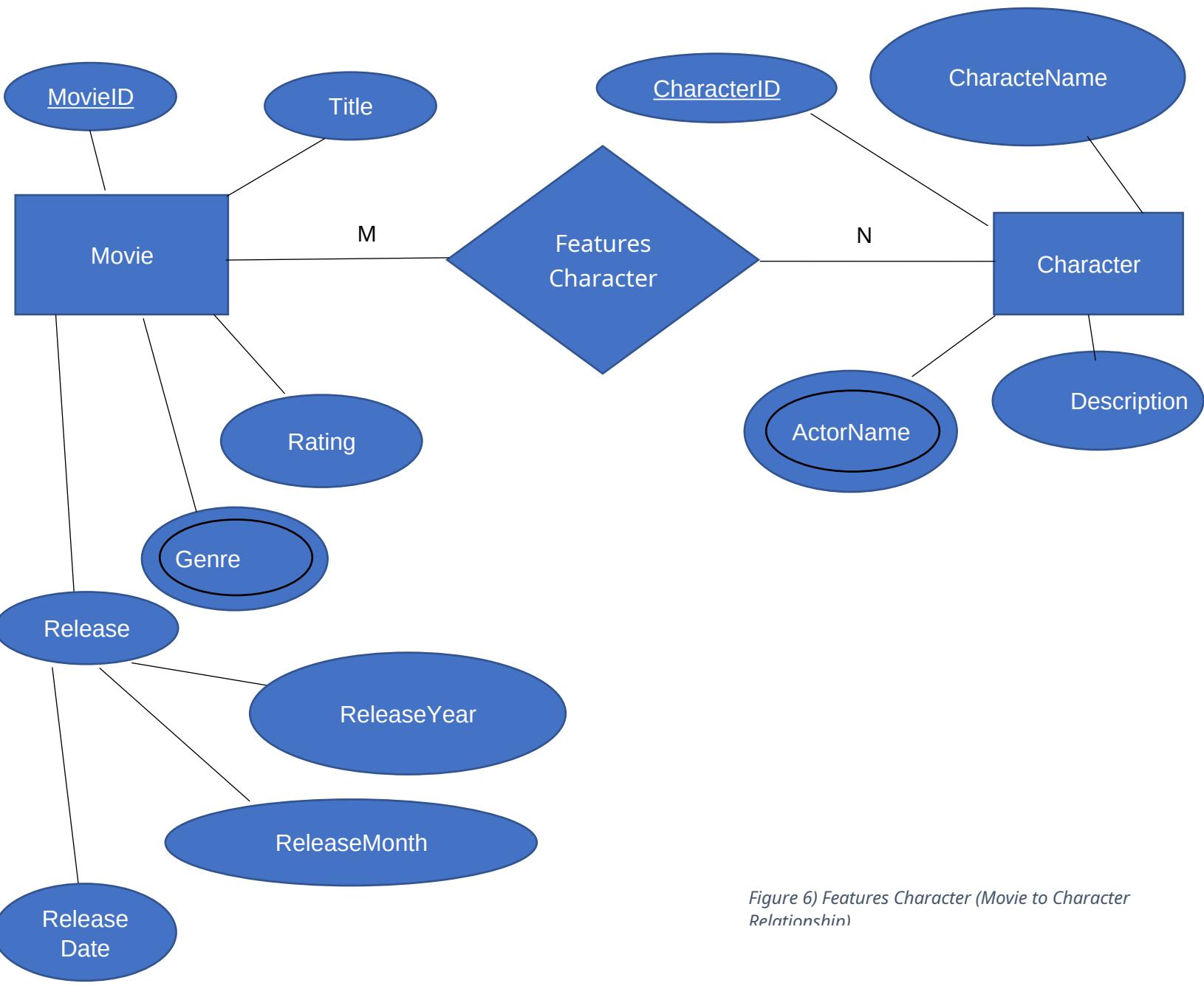


Figure 6) Features Character (Movie to Character Relationship)

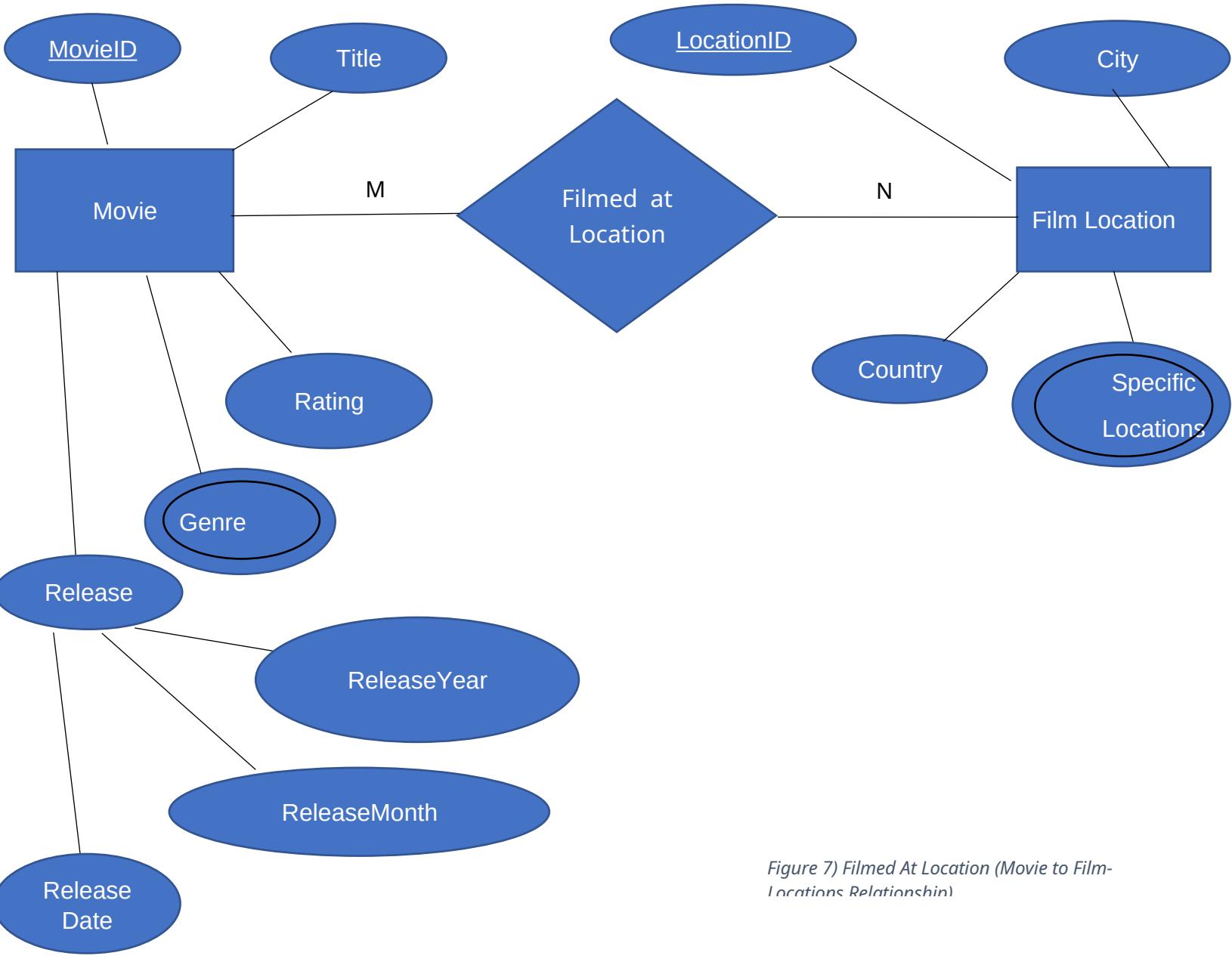


Figure 7) Filmed At Location (Movie to Film-locations Relationship)

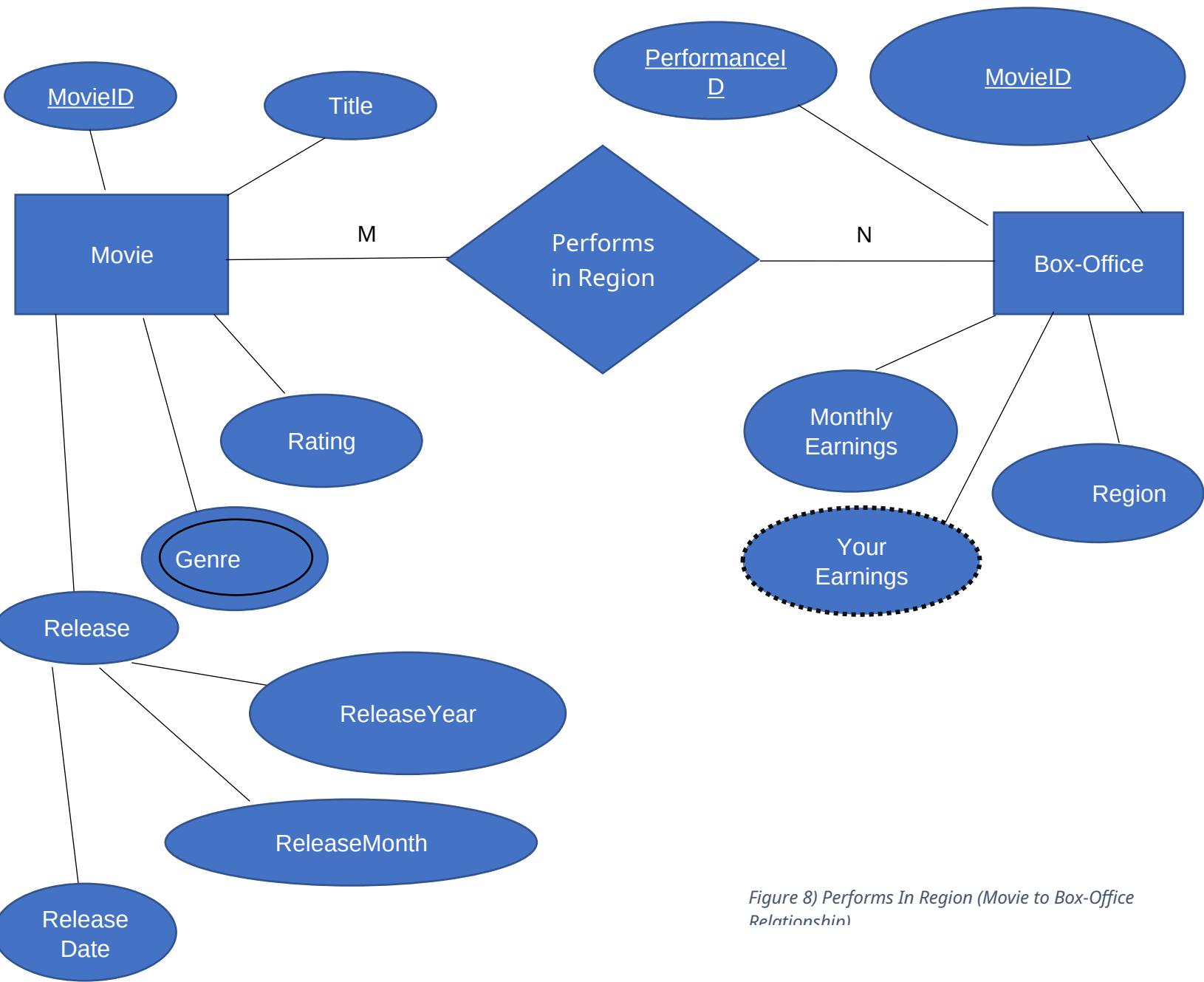


Figure 8) Performs In Region (Movie to Box-Office Relationship)

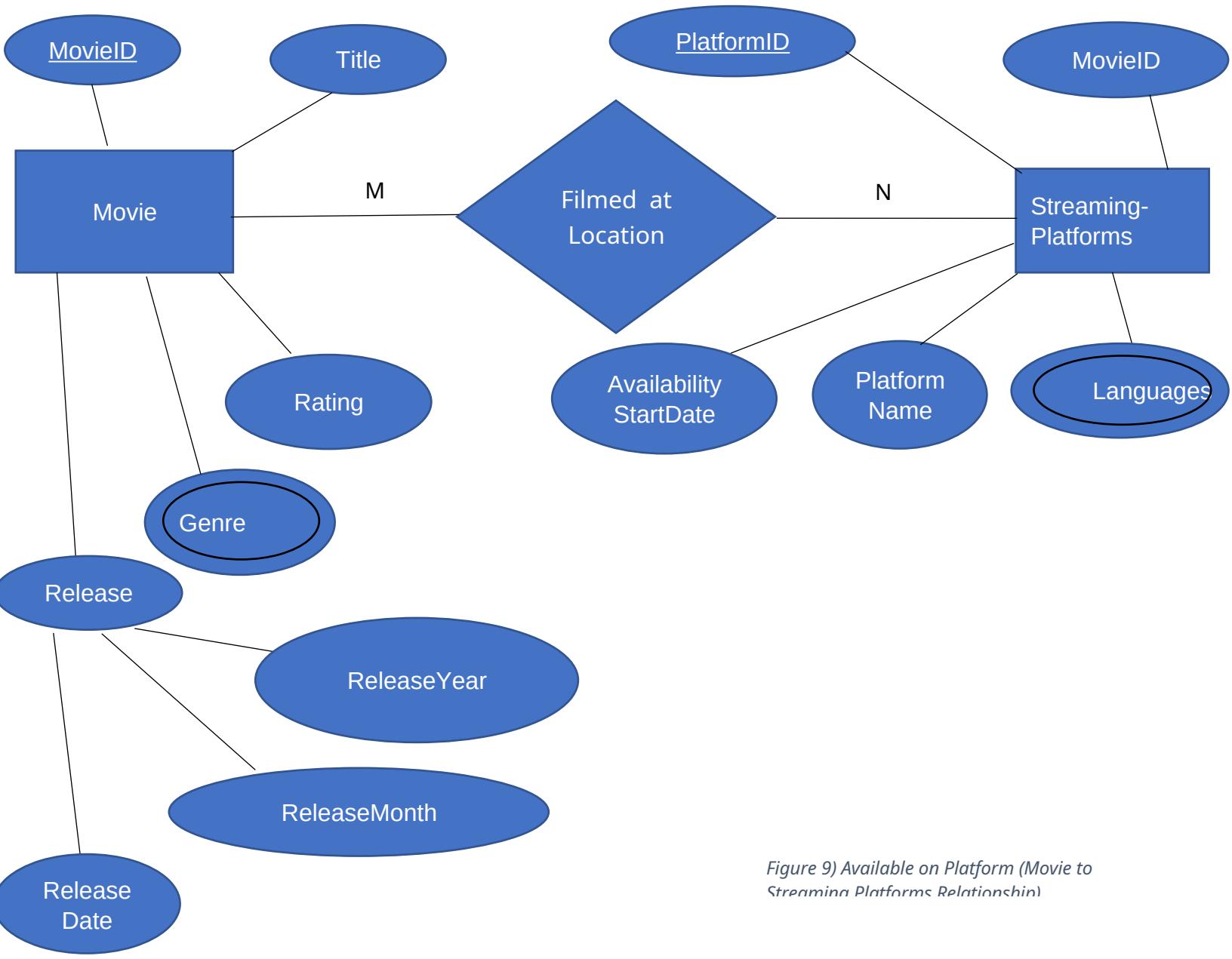
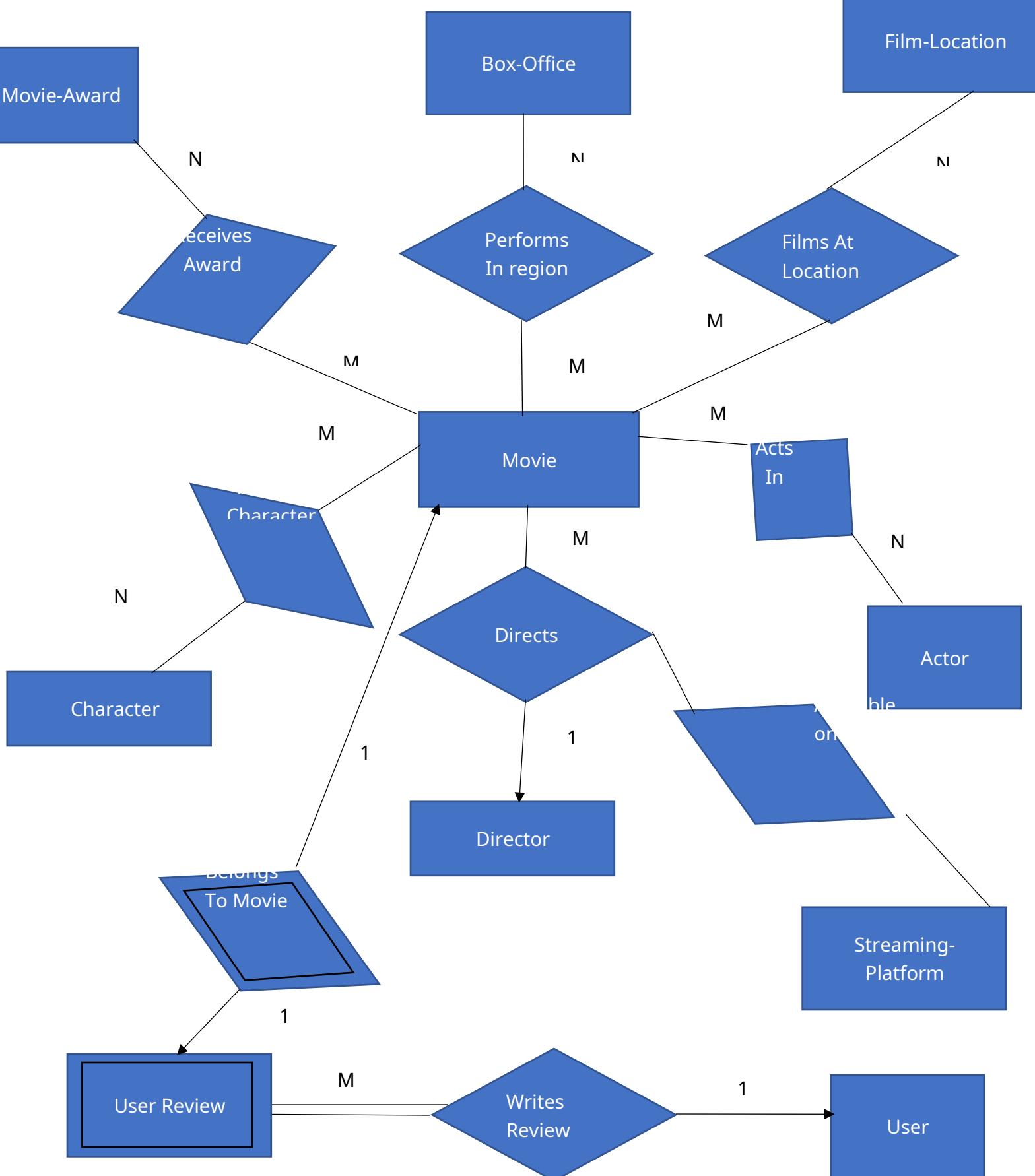
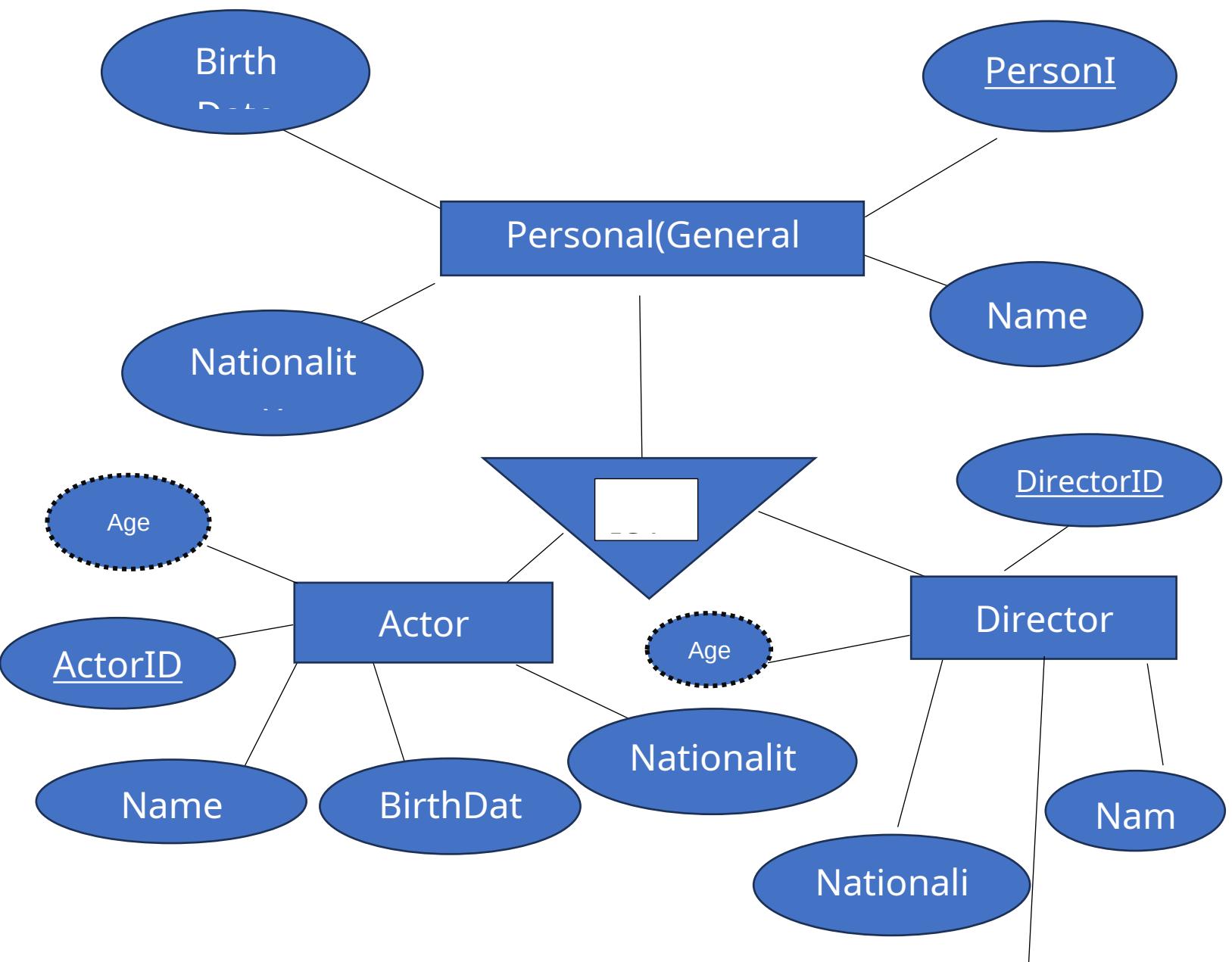
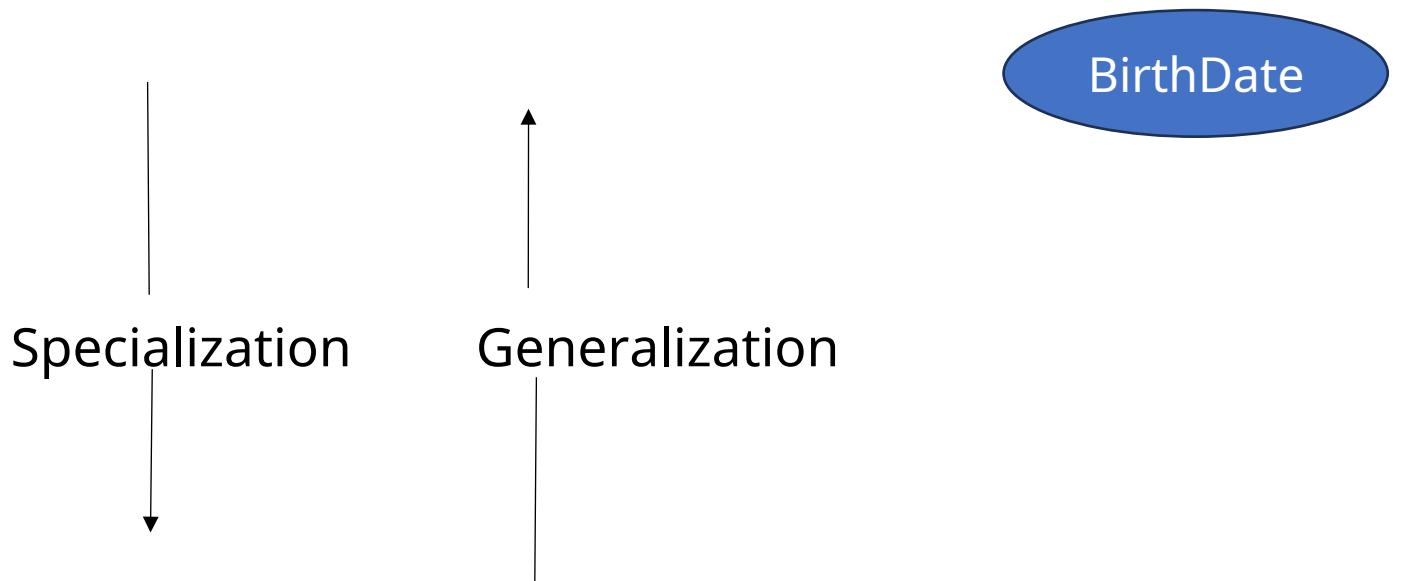


Figure 9) Available on Platform (Movie to Streaming Platforms Relationship)







'User', 'User_Review', 'Actor', and 'Director' schemas are in the form of BCNF, and the 'Movie' schema is in the 2NF form.

Entities for MOVIE

DATABASE:-

- 1) Movie (Strong entity)
 - MovieID (Primary Key)
 - Release (NOT NULL) (Composite)
 - Title (Unique)
 - Genre (Multivalued Attribute)
 - Rating (Constraint 1-10)

2) Actor (Strong entity)

- ActorID (Primary Key)

- Name

- BirthDate

- Nationality

3) Director (Strong entity)

- DirectorID (Primary Key)

- Name

- BirthDate

- Nationality

4) User (Strong entity)

- UserID (Primary Key)

- Username

- Email (UNIQUE) (NOT NULL)

- Password (UNIQUE) (NOT NULL)

5) User Review (Weak Entity)

- ReviewID (Primary Key)

- UserID (Foreign Key)

- MovieID (Foreign Key)

-Rating (Constraint 1-10)

6) Movie-Awards (Strong Entity)

-AwardID (Primary Key)

-AwardName

-Category (Multivalued Attribute)

-Year

-Winner (Multivalued)

7) Character (Strong Entity)

-CharacterID (Primary Key)

-CharacterName

-Description

-ActorName (Multivalued Attribute)

8) Film Location (Strong Entity)

-LocationID (Primary Key)

-City

-Country

SpecificLocations (Multivalued Attribute)

9) Box-Office (String Entity)

- PerformanceID (Primary Key)
- MovieID (Foreign Key)
- Region
- MonthlyEarnings (Multivalued Attribute)

10) Streaming-Platform (Strong Entity)

- PlatformID (Primary Key)
- MovieID (Foreign Key)
- PlatformName
- AvailabilityStartDate
- Languages (Multivalued Attribute)

PROJECT FRONT END

The image shows two screenshots of a web-based movies management system.

Top Screenshot: MOVIES MANAGEMENT SYSTEM - SEARCH

This screenshot displays a sidebar menu titled "MOVIES MANAGEMENT SYSTEM" with a "SEARCH" button. The sidebar lists various entities and their management functions:

- Movies**: Create Movie, Read Movie, Update Movie, Delete Movie
- Actors**: Create Actor, Read Actor, Update Actor, Delete Actor
- Directors**: Create Director, Read Director, Update Director, Delete Director
- Users**: Create User, Read User, Update User, Delete User
- User Reviews**: Create User Review, Read User Review, Update User Review, Delete User Review
- Awards**: Create Award, Read Award, Update Award, Delete Award
- Characters**: Create Character, Read Character, Update Character, Delete Character
- Film Locations**: Create Film Location, Read Film Location, Update Film Location, Delete Film Location
- Box Office**: Create Box Office, Read Box Office, Update Box Office, Delete Box Office
- Streaming Platforms**: Create Streaming Platform, Read Streaming Platform, Update Streaming Platform, Delete Streaming Platform

Bottom Screenshot: Universal Search

This screenshot shows a search interface with a header "Universal Search". It includes a search input field containing "god of war" and a "Search" button. Below the search bar, it displays "Movie Details:" for the movie "god of war".

Movie Details:

- Movie ID: 1
- Release Date: 2019-06-05
- Genre: action
- Rating: 7

User Reviews:

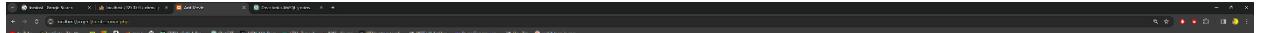
- Review ID: 1
- Rating: 8

Box Office Data:

- Performance ID: 1
- Region: india
- Monthly Earnings: 99999999.99

Streaming Platforms:

- Platform Name: NETFLIX
- Availability Start Date: 2023-06-14
- Languages: hindi,english



Add Movie

ID:

Title:

Release Date: dd-mm-yyyy

Genre:

Rating (1-10):

[Back to Home](#)



Movies

MovieID	Title	Release Date	Genre	Rating
1	god of war	2019-06-05	action	7
2	the amazing spider man	2022-06-05	thriller action	9
4	avenger	2019-01-01	action	9
5	conjuring	2018-12-12	thriller horror	9
10	thor	2023-02-06	action	9

[Back to Home](#)





Update Movie

Movie ID:

Title:

Release Date: dd-mm-yyyy 

Genre:

Rating (1-10):

[Back to Home](#)





Delete Movie

Movie ID:

[Back to Home](#)



Movies

MovieID	Title	Release Date	Genre	Rating
1	god of war	2019-06-05	action	7
4	avenger	2019-01-01	action	9
5	conjuring	2018-12-12	thriller horror	9
10	thor	2023-02-06	action	9

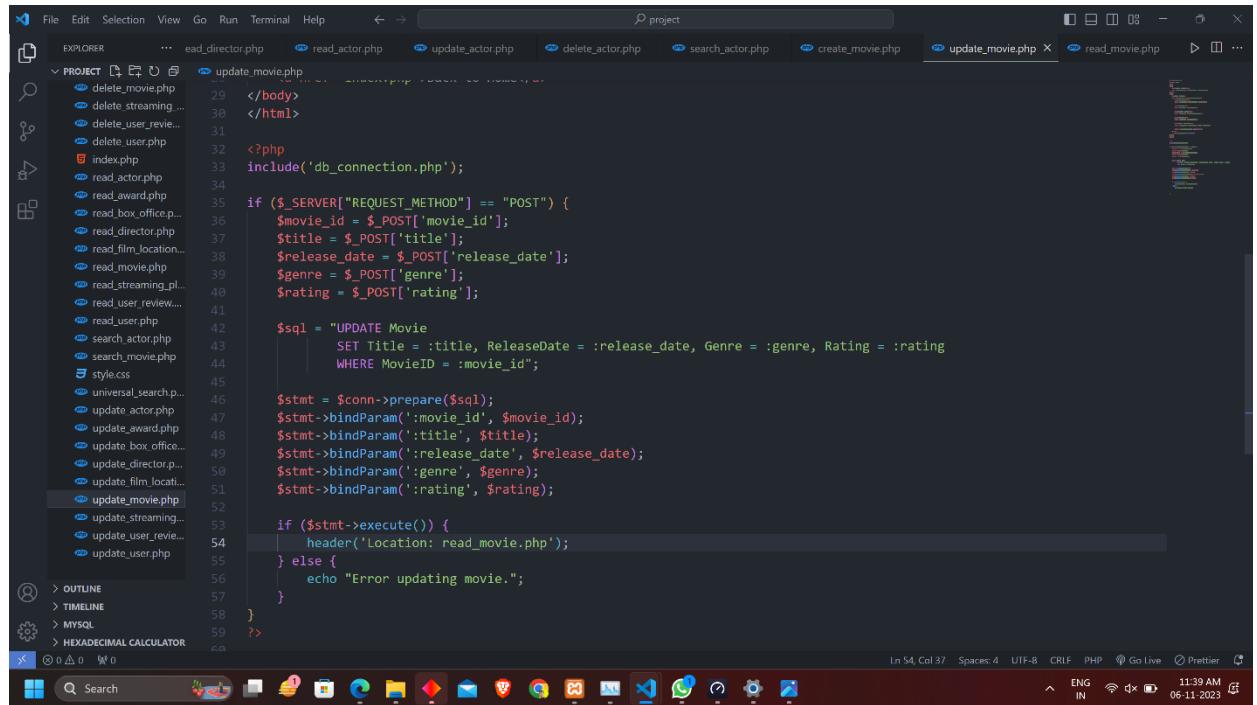
[Back to Home](#)



BACK END PHP FILES FOR ALL CRUD OPERATIONS ON DATABASE



BACK END SCREENSHOTS CONATINING



The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled 'project'. Below the menu is a tab bar with several tabs: 'ead_director.php', 'read_actor.php', 'update_actor.php', 'delete_actor.php', 'search_actor.php', 'create_movie.php', 'update_movie.php' (which is currently active), and 'read_movie.php'. The main area is the 'EXPLORER' view, showing a file tree for a project. The 'PROJECT' folder contains numerous PHP files such as 'delete_movie.php', 'read_streaming...', 'delete_user_revie...', 'delete_user.php', 'index.php', 'read_actor.php', 'read_award.php', 'read_box_office.p...', 'read_director.php', 'read_film_location...', 'read_movie.php', 'read_streaming_pl...', 'read_user_review...', 'read_user.php', 'search_actor.php', 'search_movie.php', 'style.css', 'universal_search.p...', 'update_actor.php', 'update_award.php', 'update_box_office...', 'update_director.p...', 'update_film_locati...', 'update_movie.php', 'update_streaming...', 'update_user_revie...', and 'update_user.php'. The bottom status bar displays 'In S4, Col 37' and other system information like battery level, signal strength, and date/time.

```
29     </body>
30   </html>
31
32   <?php
33     include('db_connection.php');
34
35   if ($_SERVER["REQUEST_METHOD"] == "POST") {
36     $movie_id = $_POST['movie_id'];
37     $title = $_POST['title'];
38     $release_date = $_POST['release_date'];
39     $genre = $_POST['genre'];
40     $rating = $_POST['rating'];
41
42     $sql = "UPDATE Movie
43           SET Title = :title, ReleaseDate = :release_date, Genre = :genre, Rating = :rating
44           WHERE MovieID = :movie_id";
45
46     $stmt = $conn->prepare($sql);
47     $stmt->bindParam(':movie_id', $movie_id);
48     $stmt->bindParam(':title', $title);
49     $stmt->bindParam(':release_date', $release_date);
50     $stmt->bindParam(':genre', $genre);
51     $stmt->bindParam(':rating', $rating);
52
53     if ($stmt->execute()) {
54       header('Location: read_movie.php');
55     } else {
56       echo "Error updating movie.";
57     }
58   }
59 ?>
```

MYSQL QUERY

A screenshot of a code editor interface, likely Visual Studio Code, displaying PHP code. The code is for reading movie information from a database. It includes HTML for a table header, a loop to fetch rows from the database, and an echo statement to print the table rows. The code editor has a dark theme with syntax highlighting for PHP and HTML. The status bar at the bottom shows file paths, line numbers (Ln 26, Col 41), and other system information.

```
<!DOCTYPE html>
<html>
<head>
    <title>View Movies</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <h1>Movies</h1>
    <table>
        <thead>
            <tr>
                <th>MovieID</th>
                <th>Title</th>
                <th>Release Date</th>
                <th>Genre</th>
                <th>Rating</th>
            </tr>
        </thead>
        <tbody>
            <?php
                include('db_connection.php');
                $sql = "SELECT * FROM Movie";
                $stmt = $conn->query($sql);
                while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                    echo "<tr>";
                    echo "<td>" . $row['MovieID'] . "</td>";
                    echo "<td>" . $row['Title'] . "</td>";
                    echo "<td>" . $row['ReleaseDate'] . "</td>";
                    echo "<td>" . $row['Genre'] . "</td>";
                    echo "<td>" . $row['Rating'] . "</td>";
                    echo "</tr>";
                }
            ?>
        </tbody>
    </table>
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying PHP code for deleting a movie. The code uses POST method handling, prepares a DELETE query, and executes it. It includes a back-to-home link in the HTML. The code editor has a dark theme with syntax highlighting for PHP and HTML. The status bar at the bottom shows file paths, line numbers (Ln 10, Col 45), and other system information.

```
<form>
    <a href="index.php">Back to Home</a>
</body>
</html>

<?php
include('db_connection.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $movie_id = $_POST['movie_id'];

    $sql = "DELETE FROM Movie WHERE MovieID = :movie_id";

    $stmt = $conn->prepare($sql);
    $stmt->bindParam(':movie_id', $movie_id);

    if ($stmt->execute()) {
        header('Location: read_movie.php');
    } else {
        echo "Error deleting movie.";
    }
}

?>
```

DATABASE STRUCTURE OF MYSQL

The screenshot shows the phpMyAdmin interface for the 'dbms' database. The left sidebar displays a tree view of databases and tables. The main area shows the 'user' table with one row:

	User ID	Username	Email	Password
<input type="checkbox"/>	1	naman	namangandhipersonal@gmail.com	root

Below the table, there are buttons for 'Edit', 'Copy', 'Delete', 'Check all', and 'With selected'. At the bottom, there are links for 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view) and a 'Console' button.

localhost / 127.0.0.1 / dbms

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=dbms&table=streamingplatform

Server: 127.0.0.1 > Database: dbms > Table: streamingplatform

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT * FROM `streamingplatform`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

	PlatformID	MovieID	PlatformName	AvailabilityStartDate	Languages
<input type="checkbox"/>	1	1	NETFLIX	2023-06-14	hindi,english

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / dbms

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=dbms&table=movie

Server: 127.0.0.1 > Database: dbms > Table: movie

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

```
SELECT * FROM `movie`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	MovieID	ReleaseDate	Title	Genre	Rating
<input type="checkbox"/>	1	2019-06-05	god of war	action	7
<input type="checkbox"/>	4	2019-01-01	avenger	action	9
<input type="checkbox"/>	5	2018-12-12	conjuring	thriller horror	9
<input type="checkbox"/>	10	2023-02-06	thor	action	9

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / dbms /

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=dbms&table=filmlocation

YouTube LeetCode - The Wo... Gmail ChatGPT MDN Web Docs HTML Tutorial 8085 - Course HTML attribute ref... NPTEL Hide/Show

phpMyAdmin

Server: 127.0.0.1 > Database: dbms > Table: filmlocation

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

SELECT * FROM `filmlocation`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

LocationID City Country SpecificLocation

Query results operations

Create view

localhost/phpmyadmin/index.php?route=/server/privileges&db=dbms&table=filmlocation&

localhost / 127.0.0.1 / dbms /

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=dbms&table=director

YouTube LeetCode - The Wo... Gmail ChatGPT MDN Web Docs HTML Tutorial 8085 - Course HTML attribute ref... NPTEL Hide/Show

phpMyAdmin

Server: 127.0.0.1 > Database: dbms > Table: director

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

SELECT * FROM `director`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←→ DirectorID Name BirthDate Nationality

Edit Copy Delete 1 director1 2023-10-31 Indian

Edit Copy Delete 3 director3 2023-11-15 Indian

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / dbms / character

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=dbms&table=character

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
SELECT * FROM `character`
```

CharacterID CharacterName Description ActorName

Create view

Console

localhost / 127.0.0.1 / dbms / boxoffice

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=dbms&table=boxoffice

Showing rows 0 - 0 (total, Query took 0.0003 seconds.)

```
SELECT * FROM `boxoffice`
```

PerformanceID MovieID Region MonthlyEarnings

1	1	India	99999999.99
---	---	-------	-------------

Edit Copy Delete

Show all Number of rows: 25 Filter rows: Search this table

Extra options

← → PerformanceID MovieID Region MonthlyEarnings

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / dbms / awards

localhost/phpmyadmin/index.php?route=/sql&db=dbms&table=awards&pos=0

Server: 127.0.0.1 > Database: dbms > Table: awards

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

SELECT * FROM `awards`

Profile Edit inline Edit Explain SQL Create PHP code Refresh

Show all Number of rows: 25 Filter rows: Search this table

Extra options

	AwardID	AwardName	Category	Year	Winner
<input type="checkbox"/>	1	random	horror	2019	1

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / dbms / actor

localhost/phpmyadmin/index.php?route=/sql&db=dbms&table=actor&pos=0

Server: 127.0.0.1 > Database: dbms > Table: actor

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

SELECT * FROM `actor`

Profile Edit inline Edit Explain SQL Create PHP code Refresh

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	ActorID	Name	BirthDate	Nationality
<input type="checkbox"/>	0	actor1	2023-11-08	Indian
<input type="checkbox"/>	1	actor2	2023-10-31	USA

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / dbms /

localhost/phpmyadmin/index.php?route=/sql&db=dbms&table=actor&pos=0

Server: 127.0.0.1 > Database: dbms > Table: actor

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

SELECT * FROM `actor`

Profile Edit inline Edit Explain SQL Create PHP code Refresh

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	ActorID	Name	BirthDate	Nationality
<input type="checkbox"/>	0	actor1	2023-11-08	Indian
<input type="checkbox"/>	1	actor2	2023-10-31	USA

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

localhost / 127.0.0.1 / dbms | +

localhost/phpmyadmin/index.php?route=/database/structure&db=dbms

YouTube LeetCode - The Wo... Gmail GPRM : GitHub Pro... ChatGPT MDN Web Docs HTML Tutorial 8085 - Course HTML attribute ref... NPTEL Hide/Show

phpMyAdmin

Server: 127.0.0.1 > Database: dbms

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Designer

Filters

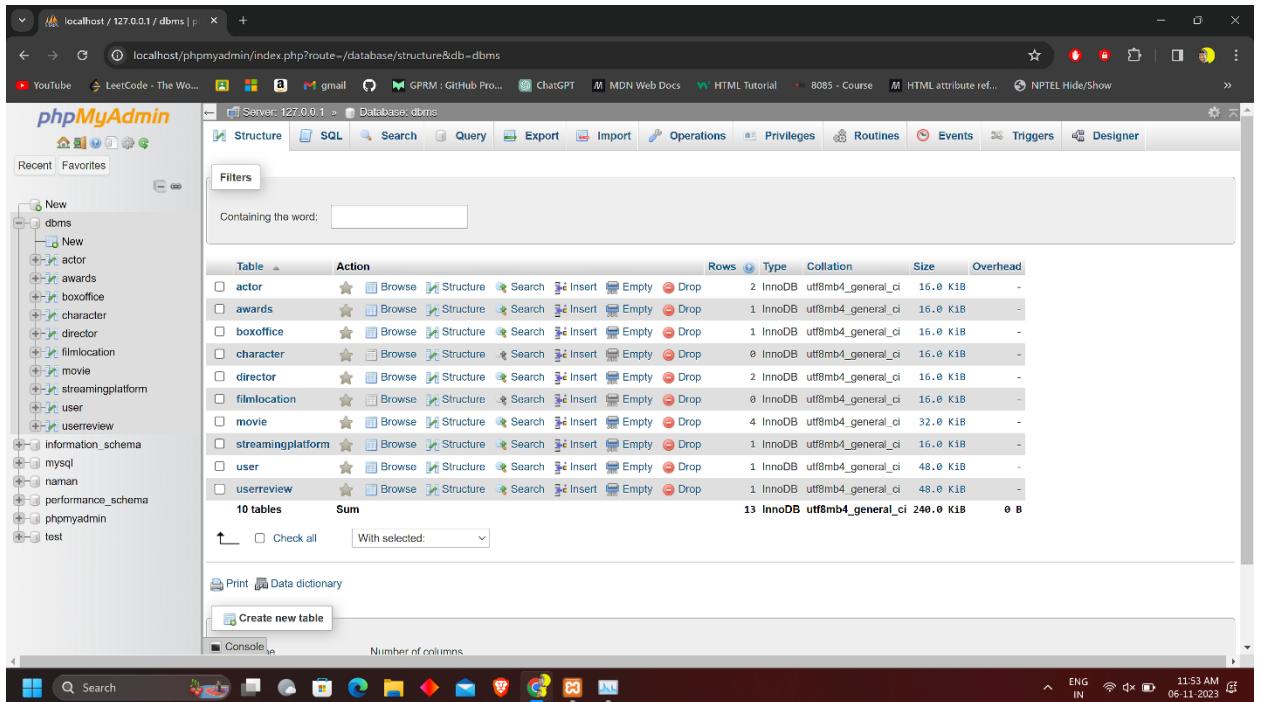
Containing the word: []

Table	Action	Rows	Type	Collation	Size	Overhead
actor	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 K1B	-
awards	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 K1B	-
boxoffice	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 K1B	-
character	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 K1B	-
director	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 K1B	-
filmlocation	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 K1B	-
movie	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 K1B	-
streamingplatform	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 K1B	-
user	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	48.0 K1B	-
userreview	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	48.0 K1B	-
10 tables	Sum	13	InnoDB	utf8mb4_general_ci	240.0 K1B	0 B

Check all With selected: []

Print Data dictionary Create new table

Console Number of columns: 11:53 AM 06-11-2023



localhost / 127.0.0.1 / dbms | +

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=dbms&table=userreview

YouTube LeetCode - The Wo... Gmail GPRM : GitHub Pro... ChatGPT MDN Web Docs HTML Tutorial 8085 - Course HTML attribute ref... NPTEL Hide/Show

phpMyAdmin

Server: 127.0.0.1 > Database: dbms > Table: userreview

Browse Structure SQL Search Insert Export Import Operations Triggers

Showing rows 0 - 0 (total. Query took 0.0003 seconds.)

SELECT * FROM `userreview`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

User ID Review ID Movie ID Rating

User ID	Review ID	Movie ID	Rating
1	1	1	8

Check all With selected: Edit Copy Delete

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console 11:54 AM 06-11-2023

