Experiment -2

AIM: To make the ER Model on Movie WatchList Application.

Theory: The Entity-Relational (ER) Model is a method used to identify entities to be represented in a database and the relationships among those entities. The ER data model specifies an enterprise schema that represents the overall logical structure of a database graphically.

Key Components of the ER Model

1. Entity
   An entity may be any object, class, person, or place. In the ER diagram, an entity is represented as a rectangle.
2. Attribute
   Attributes are used to describe the properties of an entity. In the ER diagram:
   - Ellipses are used to represent attributes.

Types of Attributes

a. Key Attribute

- Represents the main characteristic of an entity and uniquely identifies it. - Represented as an ellipse with the text underlined.

b. Composite Attribute

- Composed of multiple attributes grouped together. - Represented as an ellipse connected to smaller ellipses.

c. Multivalued Attribute

- Can have more than one value for a single entity instance. - Represented as a double ellipse.

d. Derived Attribute

- Derived from other attributes. - Represented as a dashed ellipse.

Implementation:

This ER diagram models the key entities like User and Movies, and their relationship (watches) in a Movies Watchlist Application. The diagram captures an individual's interaction with their movie watchlist.

Entity: User

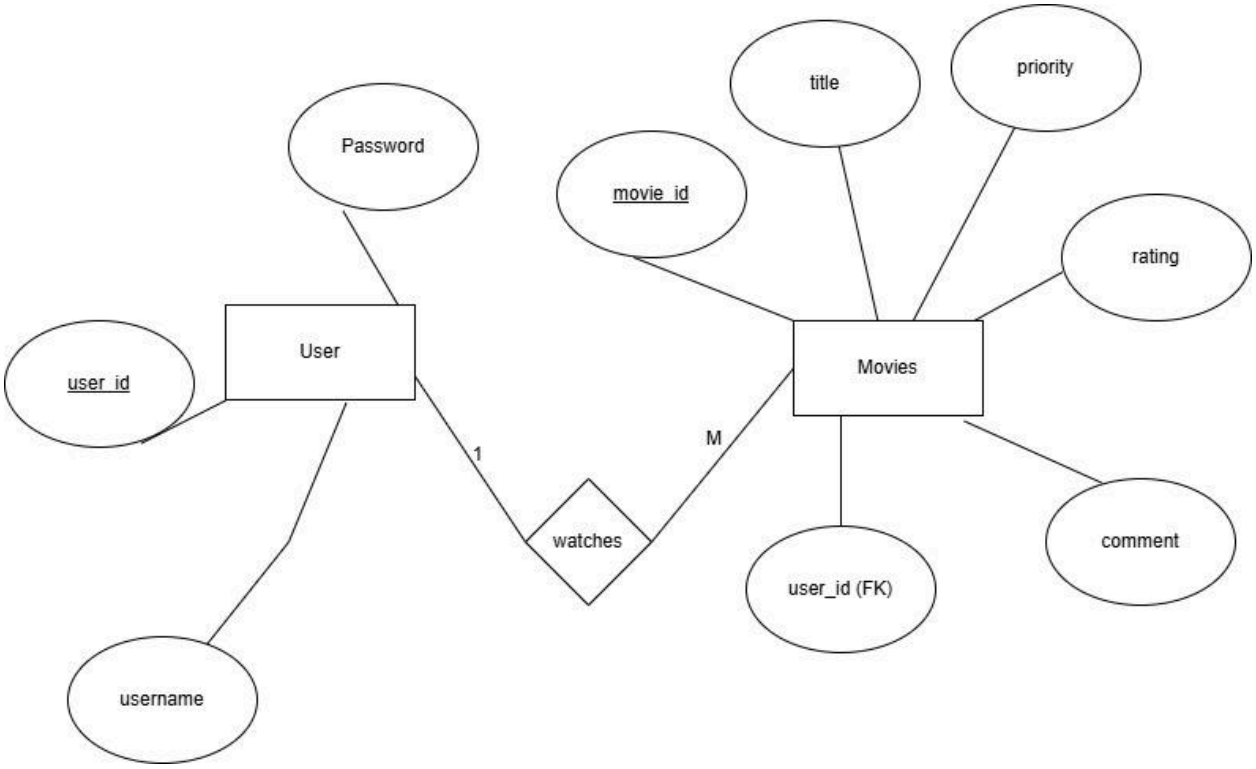
- **Attributes:**
    1. user_id (Primary Key): Unique identifier for each user.
    2. username: The name chosen by the user for their profile.
    3. password: The password for account authentication.

Entity: Movies

- **Attributes:**
    1. movie_id (Primary Key): Unique identifier for each movie.
    2. title: Name of the movie.
    3. priority: The priority level assigned by the user to the movie.
    4. rating: The rating given by the user after watching the movie.
    5. comment: User's review or comments on the movie.
    6. user_id (Foreign Key): Links to the user_id in the **User** entity to associate movies with specific users.

Relationship: watches

- The relationship **watches** connects the User entity to the Movies entity.
    - A user can watch **multiple movies** (1 relationship).
    - A movie is associated with **only one user** in this system.


Password
title
priority
movie_id
rating
User
user_id
Movies
1
M
watches
comment
user_id (FK)
username