DBMS
Movie Database

Dharmsinh Desai University
Faculty Of Technology, Nadiad

By,

Parth Shah K. IT - 114

Parth Shah N. IT - 115

9th September, 2018

Contents

Introduction

This project is a database that stores data for an app that enables users to discover new movies, get information about various movies, search movies using different filters, etc. Once the movie is found the user can get information like genres, runtime, plot, cast, crew, ratings, user reviews, release date, posters, etc. Moreover, users can review the movies they have watched and contribute to the rating accuracy which in turn can help other users discover good movies.

Uses & Advantages

Using this application users can discover movies of the genres they like or have watched before. Users can simply tune various filters and receive get receive search results based on them. Furthermore users will get recommendations about the movies that they might like based on their previously liked or watched movies using some Machine Learning algorithm. These recommendations get better with increase in the use of application by the user.

Scope

This application can further be scaled to store and retrieve data for music in a similar fashion. All the current uses can be implemented to the newly added music data.

E-R Diagram
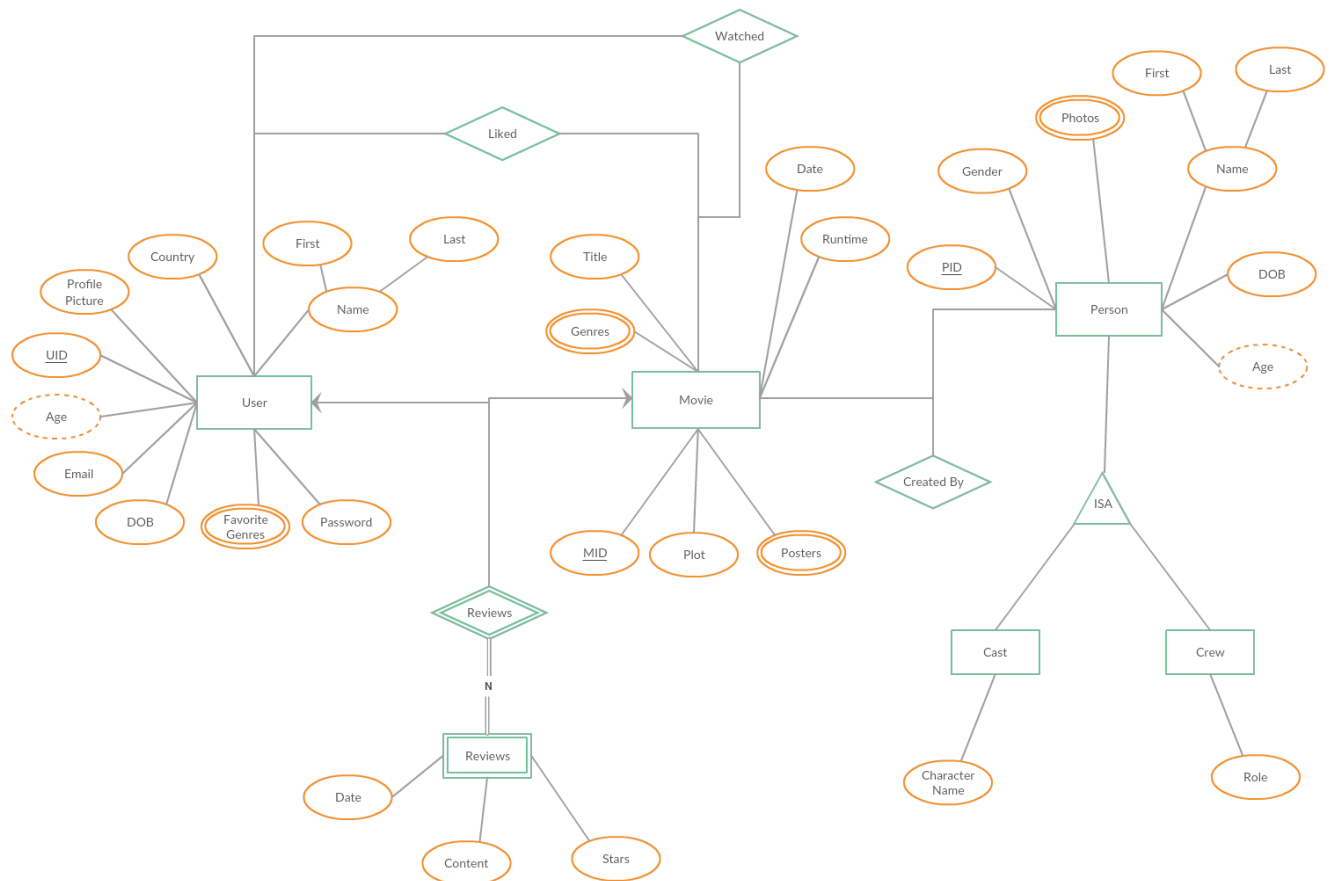
Database Schema

Schemas derived using Strong Entities:

Users (<u>UID,</u> FirstName, LastName, Country, ProfilePicture, Email, Password, DOB)

Movies (<u>MID, Title, ReleaseDate, Plot, Runtime</u>)

Person (<u>PID, FirstName, LastName, Gender, DOB</u>)

Schemas derived from multivalued attributes:

Genres (<u>MID, Name</u>)

Posters (<u>MID, FileName</u>)

Photos (<u>PID, FileName</u>)

FavGenres (<u>UID, Name</u>)

Schemas derived from weak entity set:

Reviews (<u>UID, MID</u>, Date, Stars, Content)

Schemas derived from relationships involving strong entity sets:

Watched (<u>UID, MID</u>)

Liked (<u>UID, MID</u>)

CreatedBy (<u>PID, MID</u>)

Schemas derived from ISA relationships:

Cast (<u>MID, PID</u>, CharacterName)

Crew (<u>MID, PID</u>, Role)

Schema Diagram
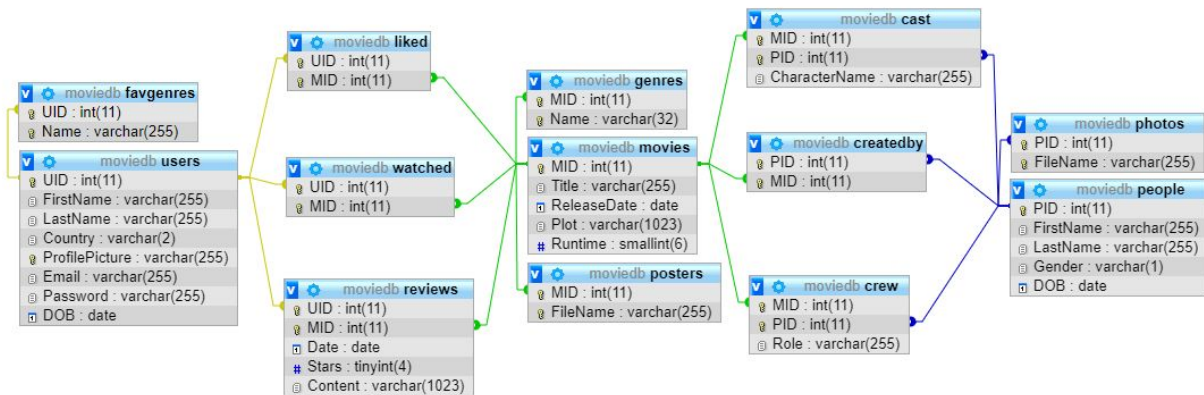
Table Definitions

```sql
CREATE TABLE Users (
    UID int NOT NULL,
    FirstName varchar(255) NOT NULL,
    LastName varchar(255) NOT NULL,
    Country varchar(2),
    ProfilePicture varchar(255),
    Email varchar(255) NOT NULL,
    Password varchar(255) NOT NULL,
    DOB Date,
    PRIMARY KEY (UID),
    UNIQUE(ProfilePicture)
);
```

```sql
CREATE TABLE Movies (
    MID int NOT NULL,
    Title varchar(255) NOT NULL,
    ReleaseDate Date NOT NULL,
    Plot varchar(1023),
    Runtime smallint,
    PRIMARY KEY (MID)
);
```

```sql
CREATE TABLE People (
    PID int NOT NULL,
    FirstName varchar(255),
    LastName varchar(255),
    Gender varchar(1),
    DOB Date,
    PRIMARY KEY (PID)
);
```

```
CREATE TABLE Genres (
    MID int NOT NULL,
    Name varchar(32) NOT NULL,
    FOREIGN KEY (MID) REFERENCES movies(MID),
    PRIMARY KEY (MID, Name)
);
```

```
CREATE TABLE Posters (
    MID int NOT NULL,
    FileName varchar(255) NOT NULL,
    FOREIGN KEY (MID) REFERENCES movies(MID),
    UNIQUE(FileName),
    PRIMARY KEY (MID, FileName)
);
```

```
CREATE TABLE Photos (
    PID int NOT NULL,
    FileName varchar(255) NOT NULL,
    FOREIGN KEY (PID) REFERENCES people(PID),
    UNIQUE(FileName),
    PRIMARY KEY (PID, FileName)
);
```

```
CREATE TABLE FavGenres (
    UID int NOT NULL,
    Name varchar(255) NOT NULL,
    FOREIGN KEY (UID) REFERENCES Users(UID),
    PRIMARY KEY (UID, Name)
);
```

```
CREATE TABLE Watched (
    UID int NOT NULL,
    MID int NOT NULL,
    FOREIGN KEY (UID) REFERENCES Users(UID),
    FOREIGN KEY (MID) REFERENCES Movies(MID),
    PRIMARY KEY (UID, MID)
);
```

```sql
CREATE TABLE Liked (
    UID int NOT NULL,
    MID int NOT NULL,
    FOREIGN KEY (UID) REFERENCES Users(UID),
    FOREIGN KEY (MID) REFERENCES Movies(MID),
    PRIMARY KEY (UID, MID)
);
```

```sql
CREATE TABLE CreatedBy (
    PID int NOT NULL,
    MID int NOT NULL,
    FOREIGN KEY (PID) REFERENCES People(PID),
    FOREIGN KEY (MID) REFERENCES Movies(MID),
    PRIMARY KEY (PID, MID)
);
```

```sql
CREATE TABLE Reviews (
    UID int NOT NULL,
    MID int NOT NULL,
    Date Date NOT NULL,
    Stars tinyint NOT NULL,
    Content varchar(1023),
    FOREIGN KEY (UID) REFERENCES Users(UID),
    FOREIGN KEY (MID) REFERENCES Movies(MID),
    PRIMARY KEY (UID, MID)
);
```

```sql
CREATE TABLE Cast (
    MID int NOT NULL,
    PID int NOT NULL,
    CharacterName varchar(255) NOT NULL,
    FOREIGN KEY (MID) REFERENCES Movies(MID),
    FOREIGN KEY (PID) REFERENCES People(PID),
    PRIMARY KEY (MID, PID)
);
```

```
CREATE TABLE Crew (
    MID int NOT NULL,
    PID int NOT NULL,
    Role varchar(255) NOT NULL,
    FOREIGN KEY (MID) REFERENCES Movies(MID),
    FOREIGN KEY (PID) REFERENCES People(PID),
    PRIMARY KEY (MID, PID)
);
```

Functionalities with Implementations

- Find Movies Of A Particular Genre

```sql
SELECT m.Title, m.ReleaseDate, m.Plot, m.Runtime
FROM movies m
INNER JOIN genres g ON m.MID=g.MID
WHERE g.Name='Horror'
```

- Get All Reviews Of A Particular Movie

```sql
SELECT CONCAT(u.FirstName, ' ', u.LastName) AS User, r.Stars AS Rating,
r.Date, r.Content AS Review
FROM reviews r
INNER JOIN movies m ON r.MID=m.MID
INNER JOIN users u ON u.UID=r.UID
WHERE m.Title='The Secret Life'
ORDER BY r.Date DESC
```

- List All Cast Members Of A Movie

```sql
SELECT CONCAT(p.FirstName, " ", p.LastName) AS `Actor Name`,
c.CharacterName AS `Character Name` FROM createdby cb
INNER JOIN movies m ON cb.MID=m.MID
INNER JOIN cast c ON c.MID=cb.MID AND c.PID=cb.PID
INNER JOIN people p ON cb.PID=p.PID
WHERE m.title='The Avengers'
```

- List All Crew Members Of A Movie

```sql
SELECT CONCAT(p.FirstName, " ", p.LastName) AS `Crew Member Name`, c.Role
AS `Character Name`
FROM createdby cb
INNER JOIN movies m ON cb.MID=m.MID
INNER JOIN crew c ON c.MID=cb.MID AND c.PID=cb.PID
INNER JOIN people p ON cb.PID=p.PID
```

```
WHERE m.title='The Avengers'
```

- List All Movies Of Favorite Genres

```
SELECT m.Title AS Movie, g.Name AS `Genre Match`
FROM movies m
INNER JOIN genres g ON m.MID=g.MID
INNER JOIN favgenres f ON f.Name=g.Name
INNER JOIN users u ON f.UID=u.UID
WHERE u.UID=1
```

- Get All Information Of A Movie

```
SELECT m.Title, m.ReleaseDate AS 'Release Date', m.Plot, m.runtime,
GROUP_CONCAT(g.Name) AS Genres
FROM movies m
INNER JOIN genres g ON m.MID=g.MID
WHERE m.Title = 'The Avengers'
GROUP BY m.MID
```

- Get Movies Released In A Particular Date Range

```
SELECT m.Title, m.ReleaseDate AS 'Release Date'
FROM movies m
WHERE m.ReleaseDate BETWEEN '2017-01-01' AND '2018-01-01'
ORDER BY m.ReleaseDate ASC
```

- Find All Movies A Person Has Starred In

```
SELECT m.Title AS Movie, c.CharacterName as 'Character Name'
FROM movies m
INNER JOIN cast c ON m.MID=c.MID
INNER JOIN people p ON c.PID=p.PID
WHERE p.PID=1
```

- Find All Movies A Person Has Worked In As Crew Member

```
SELECT m.Title AS Movie, c.Role
FROM movies m
INNER JOIN crew c ON m.MID=c.MID
INNER JOIN people p ON c.PID=p.PID
WHERE p.PID=1
```

- All Liked Movies Of A User

```
SELECT m.Title
FROM movies m
INNER JOIN liked l ON m.MID=l.MID
INNER JOIN users u ON u.UID=l.UID
WHERE u.UID=1
```

- All Watched Movies Of A User

```
SELECT m.Title
FROM movies m
INNER JOIN watched w ON m.MID=w.MID
INNER JOIN users u ON u.UID=w.UID
WHERE u.UID=1
```

- List All Reviews By A User

```
SELECT m.Title AS Movie, r.Date, r.Stars AS Rating, r.Content AS Review
FROM reviews r
INNER JOIN movies m ON r.MID=m.MID
WHERE r.UID=2
ORDER BY r.Date DESC
```

- All Cast Members Of User's Liked Movies

```
SELECT CONCAT(p.FirstName, ' ', p.LastName) AS Actor, m.Title AS Movie,
c.CharacterName AS 'Character Name'
FROM people p
```

```
INNER JOIN cast c ON p.PID=c.PID
INNER JOIN liked l ON l.MID=c.MID
INNER JOIN movies m ON c.MID=m.MID
WHERE l.UID=2
```

- List All Movies With A Rating Greater Than A Particular Rating

```
SELECT DISTINCT m.Title, r.Stars
FROM movies m
INNER JOIN reviews r ON r.MID=m.MID
WHERE r.Stars > 4
```

- List All Photos Of A Person

```
SELECT p.FileName
FROM photos p
WHERE p.PID=1
```

- List All Favorite Genre

```
SELECT f.Name AS Genres
FROM favgenres f
WHERE f.UID=1
```

- List All Genres Of A Movie

```
SELECT g.Name AS Genres
FROM genres g
WHERE g.MID=1
```

- List All Poster Of A Movie

```
SELECT p.FileName
FROM posters p
WHERE MID=1
```

Data Dictionary

cast

Column	Type	Null	Default	Links to	Comments	MIME
MID *(Primary)*	int(11)	No		movies -> MID		
PID *(Primary)*	int(11)	No		people -> PID		
CharacterName	varchar(255)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	MID	60	A	No	
				PID	60	A	No	
MID	BTREE	No	No	MID	60	A	No	
PID	BTREE	No	No	PID	60	A	No	

createdby

Column	Type	Null	Default	Links to	Comments	MIME
PID *(Primary)*	int(11)	No		people -> PID		
MID *(Primary)*	int(11)	No		movies -> MID		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	PID	45	A	No	
				MID	136	A	No	
PID	BTREE	No	No	PID	45	A	No	
MID	BTREE	No	No	MID	45	A	No	

crew

Column	Type	Null	Default	Links to	Comments	MIME
MID *(Primary)*	int(11)	No		movies -> MID		
PID *(Primary)*	int(11)	No		people -> PID		
Role	varchar(255)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	PID	76	A	No	
				MID	76	A	No	

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
MID	BTREE	No	No	MID	76	A	No	
PID	BTREE	No	No	PID	76	A	No	

favgenres

Column	Type	Null	Default	Links to	Comments	MIME
UID *(Primary)*	int(11)	No		users -> UID		
Name *(Primary)*	varchar(255)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	UID	34	A	No	
				Name	34	A	No	
UID	BTREE	No	No	UID	34	A	No	

genres

Column	Type	Null	Default	Links to	Comments	MIME
MID *(Primary)*	int(11)	No		movies -> MID		
Name *(Primary)*	varchar(32)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	MID	42	A	No	
				Name	42	A	No	
MID	BTREE	No	No	MID	42	A	No	

liked

Column	Type	Null	Default	Links to	Comments	MIME
UID *(Primary)*	int(11)	No		users -> UID		
MID *(Primary)*	int(11)	No		movies -> MID		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	UID	36	A	No	
				MID	36	A	No	
UID	BTREE	No	No	UID	36	A	No	

9/9/2018   Print view - phpMyAdmin 4.8.3

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
MID	BTREE	No	No	MID	36	A	No	

movies

Column	Type	Null	Default	Links to	Comments	MIME
MID *(Primary)*	int(11)	No				
Title	varchar(255)	No				
ReleaseDate	date	No				
Plot	varchar(1023)	Yes	*NULL*			
Runtime	smallint(6)	Yes	*NULL*			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	MID	20	A	No	

people

Column	Type	Null	Default	Links to	Comments	MIME
PID *(Primary)*	int(11)	No				
FirstName	varchar(255)	Yes	*NULL*			
LastName	varchar(255)	Yes	*NULL*			
Gender	varchar(1)	Yes	*NULL*			
DOB	date	Yes	*NULL*			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	PID	20	A	No	

photos

Column	Type	Null	Default	Links to	Comments	MIME
PID *(Primary)*	int(11)	No		people -> PID		
FileName *(Primary)*	varchar(255)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	PID	40	A	No	
				FileName	80	A	No	

http://localhost/phpmyadmin/db_datadict.php?db=moviedb&goto=db_structure.php   3/5

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
FileName	BTREE	Yes	No	FileName	80	A	No	
PID	BTREE	No	No	PID	40	A	No	

posters

Column	Type	Null	Default	Links to	Comments	MIME
MID *(Primary)*	int(11)	No		movies -> MID		
FileName *(Primary)*	varchar(255)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	MID	80	A	No	
				FileName	80	A	No	
FileName	BTREE	Yes	No	FileName	80	A	No	
MID	BTREE	No	No	MID	80	A	No	

reviews

Column	Type	Null	Default	Links to	Comments	MIME
UID *(Primary)*	int(11)	No		users -> UID		
MID *(Primary)*	int(11)	No		movies -> MID		
Date	date	No				
Stars	tinyint(4)	No				
Content	varchar(1023)	Yes	*NULL*			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	UID	46	A	No	
				MID	46	A	No	
UID	BTREE	No	No	UID	46	A	No	
MID	BTREE	No	No	MID	46	A	No	

users

Column	Type	Null	Default	Links to	Comments	MIME
UID *(Primary)*	int(11)	No				
FirstName	varchar(255)	No				
LastName	varchar(255)	No				
Country	varchar(2)	Yes	*NULL*			

ProfilePicture	varchar(255)	Yes	*NULL*			
Email	varchar(255)	No				
Password	varchar(255)	No				
DOB	date	Yes	*NULL*			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	UID	20	A	No	
ProfilePicture	BTREE	Yes	No	ProfilePicture	20	A	Yes	

watched

Column	Type	Null	Default	Links to	Comments	MIME
UID *(Primary)*	int(11)	No		users -> UID		
MID *(Primary)*	int(11)	No		movies -> MID		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	UID	84	A	No	
				MID	84	A	No	
UID	BTREE	No	No	UID	84	A	No	
MID	BTREE	No	No	MID	84	A	No	