A
Project Report
on
Movie Database

Developed by
Parth Shah K. (IT-114) - Department of IT, DD University
Parth Shah N. (IT-115) - Department of IT, DD University

Guided by
Internal Guide:
Sunil K. Vithlani
Department of Information Technology
Faculty of Technology
DD University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad - 387001
October - 2018

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "Movie Database" is a bonafide report of the work carried out by

1) Mr. Parth Shah K., Student ID No: 16ITUOS143
2) Mr. Parth Shah N., Student ID No: 16ITUON022

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2018-2019.

Prof. Sunil K. Vithlani
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Prof. Sunil K. Vithlani as well as our head of department, Prof. Vipul Dabhi who gave us the golden opportunity to do this project on the topic Movie Database, which also helped us in doing a lot of Research and helped us learn many new things that we are really thankful to.

TABLE OF CONTENTS

SYSTEM OVERVIEW

1.1 CURRENT SYSTEM

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. The database stores information of users such as emails, passwords, favorite genres, country, DOB, identity, name, profile picture path, etc. Moreover, it stores information of movies which include but not limited to movie title, runtime, genres, plot, release date, path of poster and information of people who worked to create the movie. Furthermore the database also stores the data of the reviews and rating that are posted by individual users for any movie.

1.2 OBJECTIVES OF THE PROPOSED SYSTEM

The objective of the proposed system should be as follows:
It should allow users to search for movies using detailed and vivid search filters. The application should also recommend the user movies based on their previously watched and like movies, and based on ratings and reviews of other users. These recommendations get better with increase in the use of application by the user. It also allows users to read reviews and ratings posted by other users.

1.3 ADVANTAGES OF CURRENT SYSTEM

The system enables users to discover movies of their choice using the search filters. The recommendations that users get are also helpful in discovering new movies. Moreover, users can read reviews of other movies online. 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.

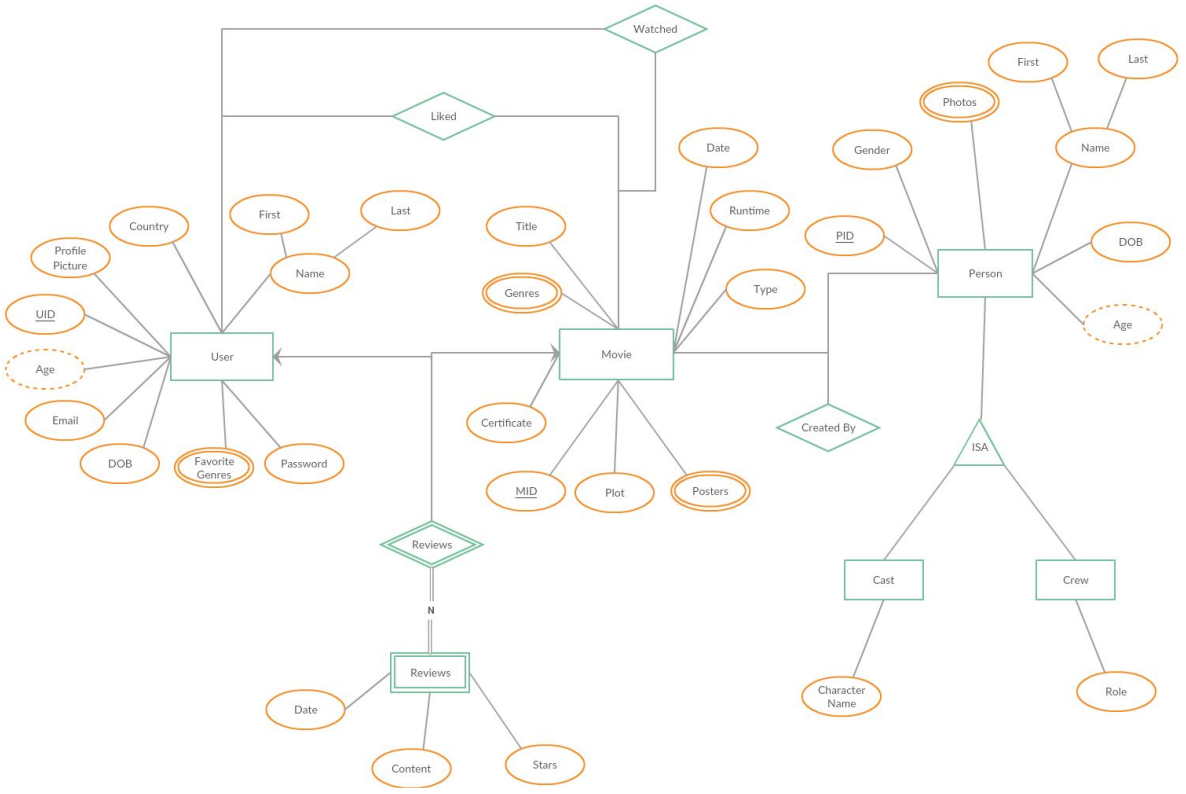
FUTURE ENHANCEMENTS OF THE SYSTEM

This system 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. Moreover it can be scaled to store more information for individual movies.

BIBLIOGRAPHY

- Database System Concepts - Fourth Edition by Silberschatz-Korth-Sudarshan

E-R DIAGRAM



2.1 ENTITIES

- User
- Movie
- Review (weak)
- Person
- Cast
- Crew

2.2 RELATIONSHIPS

- Liked (User - Movie)
- Watched (User - Movie)
- Reviews (User - Movie - Review)
- Createdby (Movie - Person)

- ISA (Person - Cast)
- ISA (Person - Crew)

2.3 MAPPING CONSTRAINTS

- Liked :- Many to Many
- Watched :- Many to Many
- Review - Movie :- Many to One
- Review - User :- Many to One
- Createdby :- Many to Many

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	
MID	BTREE	No	No	MID	76	A	No	

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
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	
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*			
Type	varchar(255)	Yes	*NULL*			
Certificate	varchar(255)	No				

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	
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	40	A	No	
				FileName	80	A	No	
FileName	BTREE	Yes	No	FileName	80	A	No	
MID	BTREE	No	No	MID	40	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	

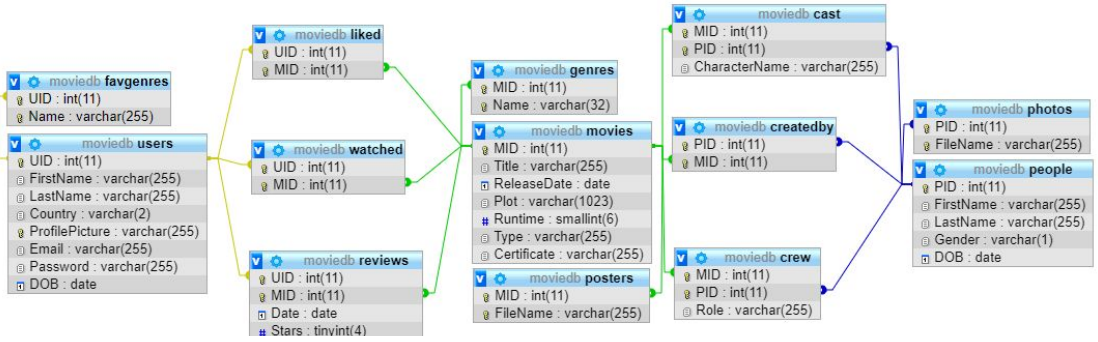
watched

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	42	A	No	
				MID	84	A	No	
UID	BTREE	No	No	UID	42	A	No	
MID	BTREE	No	No	MID	42	A	No	

SCHEMA DIAGRAM



DATABASE IMPLEMENTATIONS

5.1 CREATE SCHEMA

```
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)
);

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


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)
);

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)
```

```
);

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)
);

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)
);

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)
);
```

5.2 INSERT DATA VALUES