AGU COMP 204, GROUP 17

PROJECT

REPORT

By

GROUP 17

Burhan Özyılmaz

Mansur Muaz Ekici

30/05/2018

Kayseri / TURKEY

AGÜ

TABLE OF CONTENTS

LIST OF FIGURES

1. GENERAL DESCRIPTION OF THE PROJECT

Usually people who are addicted to watAch films prefer watching movies in a plan. To be more specific, they usually determine a director name. Then, they watch all the films of that director one by one or they can follow another way such as they can watch the films by following the awarded ones. In this project, we will provide different data about movies in a same platform. In that way, we are planning to make the searching of a film accurately.

We will store the title of the film, the duration, the director of the film, the name of the studio of the movie, the awards won by films, actors, or directors. As a group, we will normalize all the interactions among tables.

2. REQUIREMENT ANALYSIS

This project aims to help people who love watching films when they are at home. We will make a database to keep data about movies in a same platform and we will try to show films to our users according to their wants.

We may apply our project in several platforms such as Android, iOS, or we can use this on a website. According to our marketing review, we might identify the best way to reach the maximum number of users. When we determine the platform or platforms, we may use Android Studio, XCode or just a txt file to create our HTML code. We will make our technical feasibility work during the project term.

We also know we are surely aware of that our group should consider the legal usage of all the information about a certain movie. We should keep inside of laws while using them.

Up to now, we did our entity relationship diagram. For sure, we need to follow some of the improvements on our project After that, we firstly normalize our tables to prevent memorizing problems. We will create our database schemas. Before the final report, what we should work on is to provide some scripts, explanations on views, datasets and we will be sharing some of the screenshots of our work.

At the end, we will share the final model of our database. We will update all steps one by one. In that way, we will finish our project on the time.

3. SPECIFICATIONS

After that time, we will focus on our plan. The tables will be defined and normalized. Then, the database schemas will be provided. Some of the scripts will be shared. Additionally, several screenshots from our program is going to be shared. Finally, the project will be ended by sharing all the work done.

4. TOOLS/IDEs WE WILL USE

Smartdraw was used on web browser in order to draw er diagrams of the project. After we had drawn a diagram, we noticed that if we do not have full version of that service, we cannot save it as pdf because the name of the brand is

covered on the pdf file. In that way, we changed our approach to save our work by using screenshot of the diagram. Maybe, we may change the website that we used from smartdraw.com to lucidchart.com whether it is an easier program for drawing tables and interactions of each other.

We are supposed to learn sql. We determine a webpage that can be very helpful to learn the new query language. We are probably using the mySql. For the purpose of self-learning, we will need some sources that they can help us to code sql in an easier way. That's are all the programs that we are planning in the future.

We can execute our database by using different platforms. If we apply our database to use it on iOS platform. XCode will be used as our IDE. As a language, we will use Swift.

5. ER DIAGRAM

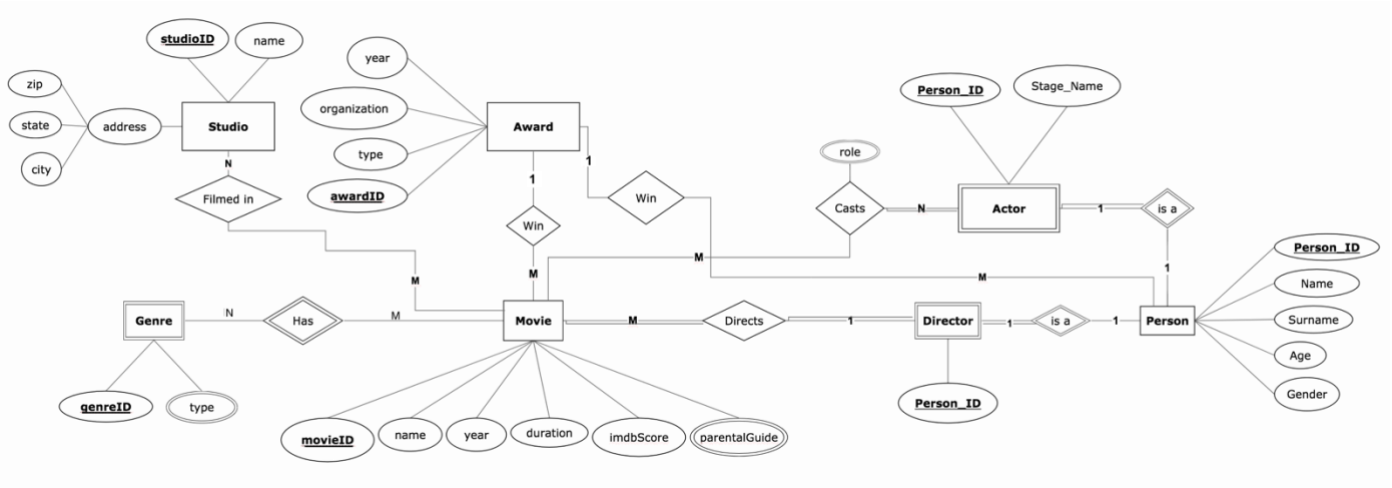


Figure 1: ER Diagram of the project (It is also attached into the zip file)

We defined "Genre" as weak entity. Because, genre don't express anything without a movie.

"Has" relationship between Genre and Movie is many to many because, one movie can have many genres and in there can be many movies in one genre.

Also, we defined "Type" as multivalued attribute in Genre, because, a movie can have more than one genre type like science-fiction and action.

We defined "parentalGuide" as multivalued attribute in Movie, because a movie can have more than one parental guide like +18 and violence & gore.

There exists a director for an every movie and every director direct at least one movie. So that, we defined both "Directs" relationship between Movie and Director as total participation relationship.

Every actor cast at least one movie so, the Actor side of the "Casts" relationship between Movie and Actor is defined as total participation. But on the other hand, there exist some animation movies which there is no actor has casted. So that, movie side of the "Casts" relationship is defined as partial participation relationship.

Also, we defined "Role" as a multivalued attribute of Casts relationship, because one actor can perform two different roles in one movie. For example, Arif and Komutan Logar characters of G.O.R.A movie is performed by Cem Yılmaz.

Director and Actor defined as weak entity, because they don't express anything without person.

We defined "Win" relationship between Award and Movie as 1 to many, because a movie can win more than one award and other side, an award can be received by just one movie. For example, The Shape of Water movie won 4 Oscar Awards in 90th academy awards. Also, same situation is possible for "Win" relationship between Person and Award.

6. DATABASE SCHEMA

1. Movies Table

Movies				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
movieID	INT	N	Sequence#	Unique identifier for a movie
movieName	Varchar(60)	N		Movie's Name
date	Datetime	Y		Movie's release date
duration	Varchar(15)	Y	#hour ##mins	Movie's duration
imdbScore	Decimal(1)	Y	#.#	Movie's IMDB score

Table 1 Movies Table

Example:

MovieID	movieName	date	duration	imdbScore
1	Inception	July 30, 2010	2h 28min	8.8

Table 2 Movies Table Example

Unique Index: movieID
Purpose: Stores the information about the names, release dates, durations and IMDB Scores of the movies.

2. ParentalGuides Table

ParentalGuides				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
parentalGuideID	INT	N	Sequence#	Unique identifier for a Parental Guide
parentalGuideName	Varchar(45)	N		Parental Guide's Description

Table 3 ParentalGuides Table

Example:

parentalGuideID	parentalGuideName
1	Violence & Gore

Table 4 ParentalGuides Table Example

Unique Index: ParentalGuideID
Purpose: Stores the information about the movie parental guides like +18 or profanity.

---

3. Genres Table

Genres				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
genreID	INT	N	Sequence#	Unique identifier for a genre
genreName	Varchar(45)	N		Genre's description

Table 5 Genres Table

Example:

genreID	genreName
1	Adventure

Table 6 Genres Table Example

Unique Index: genreID
Purpose: Stores the information about the movies' genres like adventure or comedy.

---

4. Studios Table

Studios				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
studioID	INT	N	Sequence#	Unique identifier for a Studio
studioName	Varchar(60)	N		Studio's Name

Table 7 Studios Table

Example:

studioID	studioName
1	Warner Bros.

Table 8 Studios Table Example

Unique Index: studioID
Purpose: Stores the information about the movie Studios.

---

5. Address Table

Address				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
addressID	INT	N	Sequence#	Unique identifier for an address
zipID	INT	N		Unique identifier for a zip
number	Varchar(10)	Y		The door number of the address
street	Varchar(45)	Y		Street name of the studio

Table 9 Address Table

Example:

addressID	zipID	number	street
1	3	221B	Lake Buena Vista

Table 10 Address Table Example

Unique Index: addressID

Purpose: Stores the information about the studios' address.

---

6. Zips Table

ZipTable				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
zipID	INT	N	Sequence#	Unique identifier for a Zip
zipCode	Varchar(15)	N		Zip Code of an address
cityID	INT	N		Unique identifier for a City

Table 11 Zips Table

Example:

zipID	zipCode	cityID
1	90028	1

Table 12 Zips Table Example

Unique Index: zipID
Purpose: Stores the information about the Zip Code, city and the streets.

---

7. Cities Table

Cities				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
cityID	INT	N	Sequence#	Unique identifier for a City
stateID	INT	N		Unique identifier for a state which city in.
cityName	Varchar(45)	N		City's name

Table 13 Cities Table

Example:

cityID	stateID	cityName
1	90028	1

Table 14 Cities Table Example

Unique Index: cityID
Purpose: Stores the information about the cities and its state.

---

8. States Table

States				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
stateID	INT	N	Sequence#	Unique identifier for a state
stateName	VARCHAR(45)	N		State's name

Table 15 States Table

Example:

stateID	**stateName**
1	California
2	Turkey

Table 16 States Table Example

Unique Index: stateID

Purpose: This stores data about state/country names.

---

9. AwardType Table

AwardType				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
typeID	INT	N	Sequence#	Unique identifier for a movie
typeName	VARCHAR(60)	N		Name of the given award

Table 17 AwardType Table

Example:

typeID	**typeName**

1	The best actor

Table 18 AwardType Table Example

Unique Index: typeID

Purpose: Stores information about award type.

---

10. AwardingOrganization Table

AwardingOrganization				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
organizationID	INT	N	Sequence#	Unique identifier for awarding organization
organizationName	VARCHAR(60)	N		Name of the organization that gives award

Table 19 AwardingOrganization Table

Example:

organizationID	organizationName
1	Academy Awards

Table 20 AwardType Table Example

Unique Index: organizationID

Purpose: This stores information about awarding organization

---

11. Jobs Table

Jobs				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
jobID	INT	N	Sequence#	Unique identifier for different jobs

jobName	VARCHAR(45)	N		Name of the job for people (Director, Actor, Actress)

Table 21 Jobs Table

Example:

jobID	jobName
1	Actor
2	Actress
3	Director

Table 22 Jobs Table Example

Unique Index: jobID

Purpose: This stores information about job type such as actor or director.

---

12. People Table

People				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
personID	INT	N	Sequence#	Unique identifier for people information
name	VARCHAR(45)	N		Name of the person
surname	VARCHAR(45)	N		Surname of the person
yearOfBirth	DATETIME	Y	####	The birth year of person
genderID	INT	Y		Genders unique identifier

Table 23 People Table

Example:

personID	name	surname	yearofBirth	genderID
1	Brad	Pitt	1963	1

Table 24 People Table Example

Unique Index: personID

Purpose: This stores information about person such as name, surname and genderID

13. Genders Table

Genders				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
genderID	INT	N	Sequence#	Unique identifier gender of person
gender	VARCHAR(45)	N		Gender for person

Table 25 Genders Table

Example:

genderID	gender
1	Male
2	Female
3	Other

Table 26 Genders Table Example

Unique Index: genderID

Purpose: This stores gender information for people

14. Users

Users				
Column	**Data Type**	**Nulls**	**Domain**	**Reference**
userID	INT	N	Sequence#	ID number of the user
username	VARCHAR(45)	N		Username of the user
password	VARCHAR(45)	N		Password of the user

Table 25 Genders Table

Example:

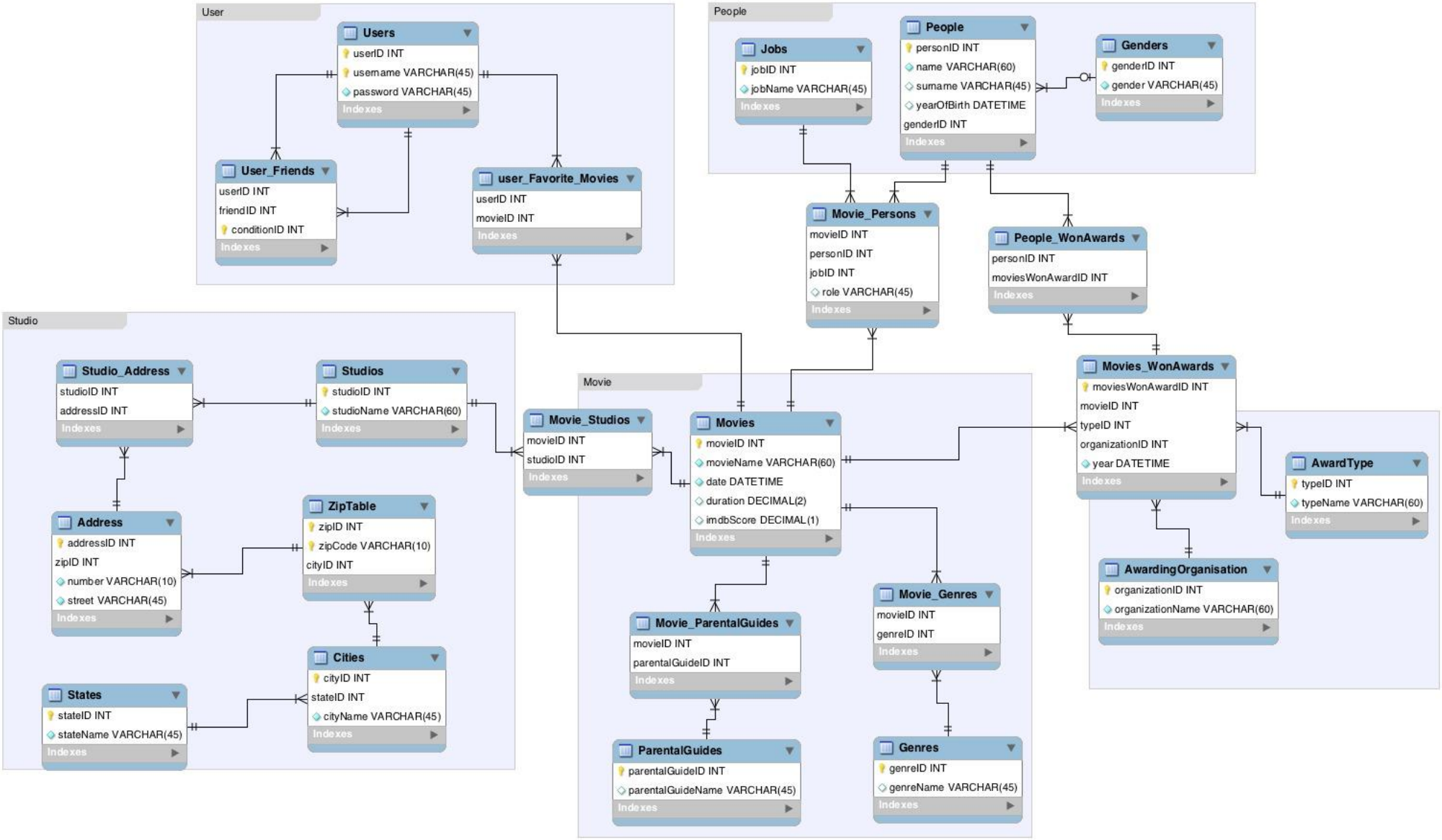
userdID	username	password
1	MCMickey	123123
2	gelsenbilsen	123456
3	Ahmet_soran	klKMnstfgk

Table 26 Genders Table Example

Unique Index: userID

Purpose: This stores user information for users class

EXTENDED ENTITY RELATIONAL DIAGRAM



7. SCRIPTS

```
-- -----------------------------------------------------


-- Table `Movies`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Movies` (
  `movieID` INT NOT NULL AUTO_INCREMENT,
  `movieName` VARCHAR(60) NOT NULL,
  `date` DATETIME NOT NULL,
  `duration` DECIMAL(2) NULL,
  `imdbScore` DECIMAL(1) NULL,
  PRIMARY KEY (`movieID`);


-- -----------------------------------------------------
-- Table `ParentalGuides`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `ParentalGuides` (
  `parentalGuideID` INT NOT NULL AUTO_INCREMENT,
  `parentalGuideName` VARCHAR(45) NULL,
  PRIMARY KEY (`parentalGuideID`);

-- -----------------------------------------------------
-- Table `Movie_ParentalGuides`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Movie_ParentalGuides` (
  `movieID` INT NOT NULL,
  `parentalGuideID` INT NOT NULL,
  INDEX `movieID_idx` (`movieID` ASC),
  INDEX `parentalGuideID_idx` (`parentalGuideID` ASC),
  PRIMARY KEY (`parentalGuideID`, `movieID`);

-- -----------------------------------------------------
-- Table `Genres`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Genres` (
  `genreID` INT NOT NULL AUTO_INCREMENT,
  `genreName` VARCHAR(45) NULL,
  PRIMARY KEY (`genreID`);


-- -----------------------------------------------------
-- Table `Movie_Genres`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Movie_Genres` (
```

```
  `movieID` INT NOT NULL,
  `genreID` INT NOT NULL,
  INDEX `movieID_idx` (`movieID` ASC),
  INDEX `genreID_idx` (`genreID` ASC),
  PRIMARY KEY (`movieID`, `genreID`);

-- -----------------------------------------------------
-- Table `Studios`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Studios` (
  `studioID` INT NOT NULL AUTO_INCREMENT,
  `studioName` VARCHAR(60) NOT NULL,
  PRIMARY KEY (`studioID`);


-- -----------------------------------------------------
-- Table `Movie_Studios`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Movie_Studios` (
  `movieID` INT NOT NULL,
  `studioID` INT NOT NULL,
  INDEX `movieID_idx` (`movieID` ASC),
  INDEX `studioID_idx` (`studioID` ASC),
  PRIMARY KEY (`movieID`, `studioID`);


-- -----------------------------------------------------
-- Table `States`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `States` (
  `stateID` INT NOT NULL AUTO_INCREMENT,
  `stateName` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`stateID`);


-- -----------------------------------------------------
-- Table `Cities`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Cities` (
  `cityID` INT NOT NULL,
  `stateID` INT NOT NULL,
  `cityName` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`cityID`, `stateID`);


-- -----------------------------------------------------
-- Table `ZipTable`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `ZipTable` (
```

```
  `zipID` INT NOT NULL,
  `zipCode` VARCHAR(10) NOT NULL,
  `cityID` INT NOT NULL,
  PRIMARY KEY (`zipID`, `cityID`, `zipCode`);


-- -----------------------------------------------------
-- Table `Address`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Address` (
  `addressID` INT NOT NULL AUTO_INCREMENT,
  `zipID` INT NOT NULL,
  `number` VARCHAR(10) NOT NULL,
  `street` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`addressID`, `zipID`);


-- -----------------------------------------------------
-- Table `Studio_Address`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Studio_Address` (
  `studioID` INT NOT NULL,
  `addressID` INT NOT NULL,
PRIMARY KEY (`studioID`, `addressID`);

-- -----------------------------------------------------
-- Table `AwardType`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `AwardType` (
  `typeID` INT NOT NULL AUTO_INCREMENT,
  `typeName` VARCHAR(60) NOT NULL,
  PRIMARY KEY (`typeID`);


-- -----------------------------------------------------
-- Table `AwardingOrganisation`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `AwardingOrganisation` (
  `organizationID` INT NOT NULL AUTO_INCREMENT,
  `organizationName` VARCHAR(60) NOT NULL,
  PRIMARY KEY (`organizationID`);


-- -----------------------------------------------------
-- Table `Movies_WonAwards`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Movies_WonAwards` (
  `moviesWonAwardID` INT NOT NULL,
  `movieID` INT NOT NULL,
```

```
  `typeID` INT NOT NULL,
  `organizationID` INT NOT NULL,
  `year` DATETIME NOT NULL,
PRIMARY KEY (`movieID`, `typeID`, `organizationID`, `moviesWonAwardID`);


-- -----------------------------------------------------
-- Table `Genders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Genders` (
  `genderID` INT NOT NULL AUTO_INCREMENT,
  `gender` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`genderID`);

-- -----------------------------------------------------
-- Table `People`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `People` (
  `personID` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(60) NOT NULL,
  `surname` VARCHAR(45) NULL,
  `yearOfBirth` DATETIME NULL,
  `genderID` INT NOT NULL,
  PRIMARY KEY (`personID`, `genderID`);

-- -----------------------------------------------------
-- Table `Jobs`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Jobs` (
  `jobID` INT NOT NULL,
  `jobName` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`jobID`));

-- -----------------------------------------------------
-- Table `Movie_Persons`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Movie_Persons` (
  `movieID` INT NOT NULL,
  `personID` INT NOT NULL,
  `jobID` INT NOT NULL,
  `role` VARCHAR(45) NULL,
PRIMARY KEY (`movieID`, `personID`, `jobID`);


-- -----------------------------------------------------
-- Table `Users`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Users` (
  `userID` INT NOT NULL,
```

```
  `username` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`userID`, `username`));


-- -----------------------------------------------------
-- Table `user_Favorite_Movies`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `user_Favorite_Movies` (
  `userID` INT NOT NULL,
  `movieID` INT NOT NULL,
  PRIMARY KEY (`userID`, `movieID`);


-- -----------------------------------------------------
-- Table `People_WonAwards`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `People_WonAwards` (
  `personID` INT NOT NULL,
  `moviesWonAwardID` INT NOT NULL,
  PRIMARY KEY (`personID`, `moviesWonAwardID`);


-- -----------------------------------------------------
-- Table `User_Friends`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `User_Friends` (
  `userID` INT NOT NULL,
  `friendID` INT NOT NULL,
  `conditionID` INT NOT NULL,
  PRIMARY KEY (`userID`, `friendID`, `conditionID`);
```

```
"SELECT * " +
        "FROM Movies " +
        "WHERE movieName LIKE ?;"
```

```
"SELECT * " +
    "FROM Casts " +
    "INNER JOIN Movies ON Casts.movieID = Movies.movieID " +
    "INNER JOIN People ON Casts.personID = People.personID " +
    "WHERE People.name LIKE ? OR People.surname LIKE ? OR Casts.role LIKE ? " +
    "GROUP BY Movies.movieName;"
```

```
SELECT * " +
    "FROM Directs " +
```

```
    "INNER JOIN Movies ON Directs.movieID = Movies.movieID " +
    "INNER JOIN People ON Directs.personID = People.personID " +
    "WHERE People.name LIKE ? OR People.surname LIKE ?" +
    "GROUP BY Movies.movieName;"
```

```
"SELECT * " +
    "FROM Movie_Studios " +
    "INNER JOIN Movies ON Movie_Studios.movieID = Movies.movieID " +
    "INNER JOIN Studios ON Movie_Studios.studioID = Studios.studioID " +
    "WHERE Studios.studioName LIKE ? " +
    "GROUP BY Movies.movieName;"
```

```
"SELECT * " +
    "FROM Movie_Genres " +
    "INNER JOIN Movies ON Movie_Genres.movieID = Movies.movieID " +
    "INNER JOIN Genres ON Movie_Genres.genreID = Genres.genreID " +
    "WHERE Genres.genreName =? " +
    "GROUP BY Movies.movieName;"
```

```
"SELECT * " +
    "FROM Movie_ParentalGuides " +
    "INNER JOIN Movies ON Movie_ParentalGuides.movieID = Movies.movieID " +
    "INNER JOIN ParentalGuides ON Movie_ParentalGuides.parentalGuideID = ParentalGuides.ParentalGuideID " +
    "WHERE ParentalGuides.ParentalGuideName =? " +
    "GROUP BY Movies.movieName;"
```

```
            "SELECT * " +
    "FROM Movies " +
    "WHERE imdbScore =?;"
```

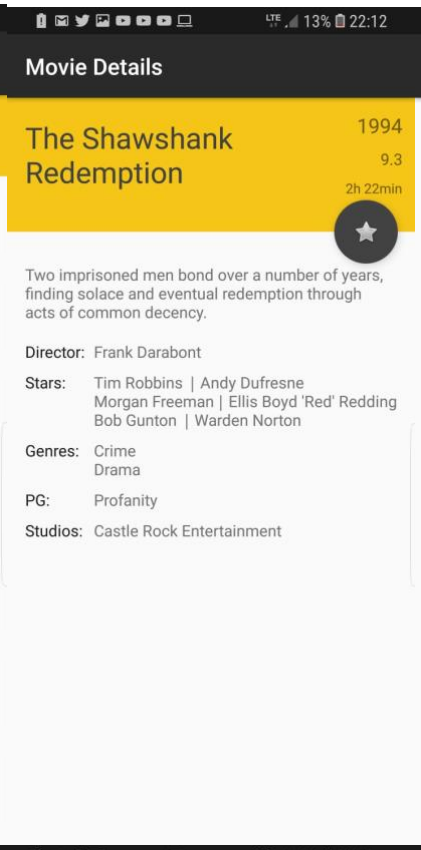
```
"SELECT Users.userID, Users.username " +
        "FROM Users " +
        "INNER JOIN User_Friends ON Users.userID = User_Friends.userID " +
        "WHERE User_Friends.friendID=? AND stateID=1 " +
        "ORDER BY Users.username DESC;"
```

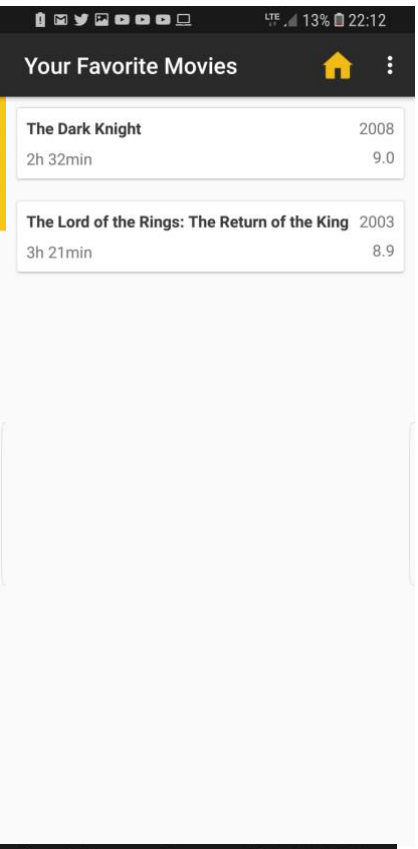
```
"SELECT * " +
"FROM Movies " +
 "ORDER BY imdbScore DESC;"
```

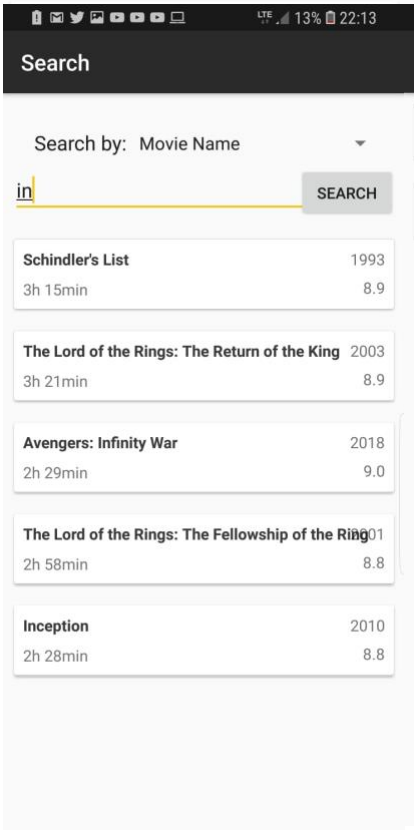
```
"SELECT * " +
"FROM Users" +
"WHERE userID <> ?" +
 "ORDER BY username DESC;"
```

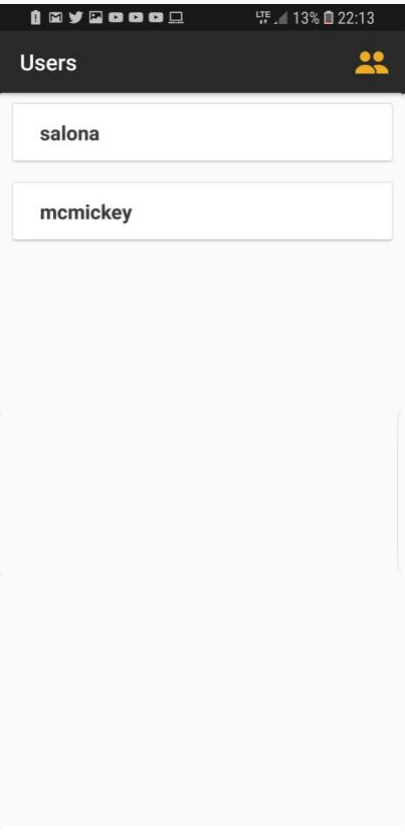
```
"SELECT Movies.movieName, Movies.date, Movies.imdbScore, Movies.duration, Movies.movieID " +
      "FROM Movies " +
      "INNER JOIN User_Favorites ON Movies.movieID = User_Favorites.movieID " +
      "WHERE userID=? " +
      "ORDER BY Movies.imdbScore DESC;"
```

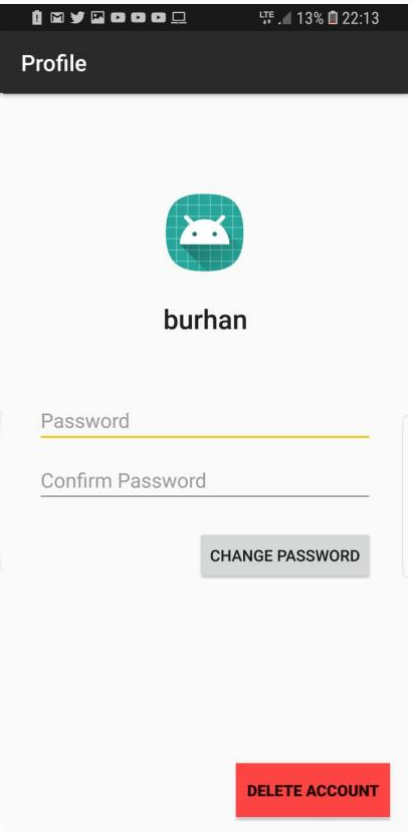
8. SCREENSHOOTS











9. DATASETS

Table: Casts

	personID	movieID	role
	Filter	Filter	Filter
1	1	1	Andy Dufresne
2	2	1	Ellis Boyd 'Red' ...
3	3	1	Warden Norton
4	4	2	Don Vito Corleone
5	5	2	Sonny Carlone
6	6	2	Clemenza
7	5	3	Michael
8	7	3	Vito Carleone
9	8	3	Tom Hagen
10	9	4	Bruce Wayne
11	10	4	Joker
12	11	4	Harvey Dent
13	12	5	Null
14	13	5	Null
15	14	5	Null
16	15	6	Oskar Schindler
17	16	6	Amon Goeth
18	17	6	Itzhak Stern
19	18	7	Frodo
20	19	7	Aragorn
21	20	7	Gandalf
22	21	8	Vincent Vega
23	22	8	Mia Wallace

Table: Genres

	genreID	genreName
	Filter	Filter
1	1	Action
2	2	Adventure
3	3	Comedy
4	4	Crime
5	5	Drama
6	6	History
7	7	Horror
8	8	Western
9	9	Science Fiction
10	10	Biography
11	11	Fantasy
12	12	Romance

Table: Users

	userID	username	password
	Filter	Filter	Filter
1	1	mcmickey	123123
2	2	burhan	123123

Table: Movies

New Record | Delete Record

	movieID	movieName	date	duration	imdbScore	topic
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	The Shawshank ...	1994	2h 22min	9.3	Two imprisoned men bond over a number of years, fi...
2	2	The Godfather	1972	2h 55min	9.2	The aging patriarch of an organized crime dynasty tr...
3	3	The Godfather 2	1974	3h 22min	9.0	The early life and career of Vito Corleone in 1920s N...
4	4	The Dark Knight	2008	2h 32min	9.0	When the menace known as the Joker emerges from ...
5	5	12 Angry Men	1957	1h 36min	8.9	A jury holdout attempts to prevent a miscarriage of j...
6	6	Schindler's List	1993	3h 15min	8.9	In German-occupied Poland during World War II, Oskar...
7	7	The Lord of the...	2003	3h 21min	8.9	Gandalf and Aragorn lead the World of Men against S...
8	8	Pulp Fiction	1994	2h 34min	8.9	The lives of two mob hitmen, a boxer, a gangster's w...
9	9	Il buono, il brutt...	1966	2h 41min	8.9	A bounty hunting scam joins two men in an uneasy al...
10	10	Fight Club	1999	2h 19min	8.8	An insomniac office worker, looking for a way to chan...
11	11	Avengers: Infinit...	2018	2h 29min	9.0	The Avengers and their allies must be willing to sacri...
12	12	The Lord of the...	2001	2h 58min	8.8	A meek Hobbit from the Shire and eight companions ...
13	13	Forrest Gump	1994	2h 22min	8.8	The presidencies of Kennedy and Johnson, Vietnam, ...
14	14	Star Wars: Epis...	1980	2h 41min	8.8	After the rebels are brutally overpowered by the Emp...
15	15	Inception	2010	2h 28min	8.8	A thief, who steals corporate secrets through the us...

Table: People

	personID	name	surname
	Filter	Filter	Filter
1	1	Tim	Robbins
2	2	Morgan	Freeman
3	3	Bob	Gunton
4	4	Marlon	Brando
5	5	Al	Pacino
6	6	James	Caan
7	7	Robert De	Niro
8	8	Robert	Duvall
9	9	Christian	Bale
10	10	Heath	Ledger
11	11	Aaron	Eckhart
12	12	Henry	Fonda
13	13	Lee J.	Cobb
14	14	Martin	Balsam

Table: ParentalGuides

	ParentalGuideID	ParentalGuideName
	Filter	Filter
1	1	Sex&Nudity
2	2	Violence&Gore
3	3	Profanity
4	4	Alcohol,Drugs&Smoking
5	5	Frightening&Intense Scenes

Table: Studios

	studioID	studioName
	Filter	Filter
1	1	Castle Rock Entertainment
2	2	Paramount Pictures
3	3	Alfran Productions
4	4	The Coppola Company
5	5	Warner Bros.
6	6	Legendary Entartainment
7	7	Syncopy
8	8	Orion-Nova Productions
9	9	Universal Pictures
10	10	Amblin Entertainment
11	11	New Line Cinema
12	12	WingNut Films
13	13	The Saul Zaentz Company
14	14	Miramax
15	15	A Band Apart
16	16	Jersey Films
17	17	Produzioni Europee Associate (PEA)
18	18	Arturo González Producciones Cinematográficas
19	19	Constantin Film