MOVIE DATABASE

Final Project for SQL Module by Sayyed Mohd. Maaz

Database Description:

The sample database represents some of the data storage and retrieval about a movie-related industry. Most people love to watch movies, and for all of them we, are providing sample information on the movie-related questions coming to their mind. This design of the database will make it easier for movie lovers to know the curiosities about the movies.

The purpose of the Movie Database:

● Help users to find interesting movies easier:

··Allow the user to search movies by typing in the keyword of the movie.

··Allow the user to search movies by typing in the name of the director.

··Allow the user to search movies by typing in an actor’s name.

··Allow the user to search movies by typing in the movie's name.

● Help inform the user about popular movies

··Show the rating for each movie.

··Offer a brief description of each movie.

● Movies in the database will be movies that we feel would be worth watching.

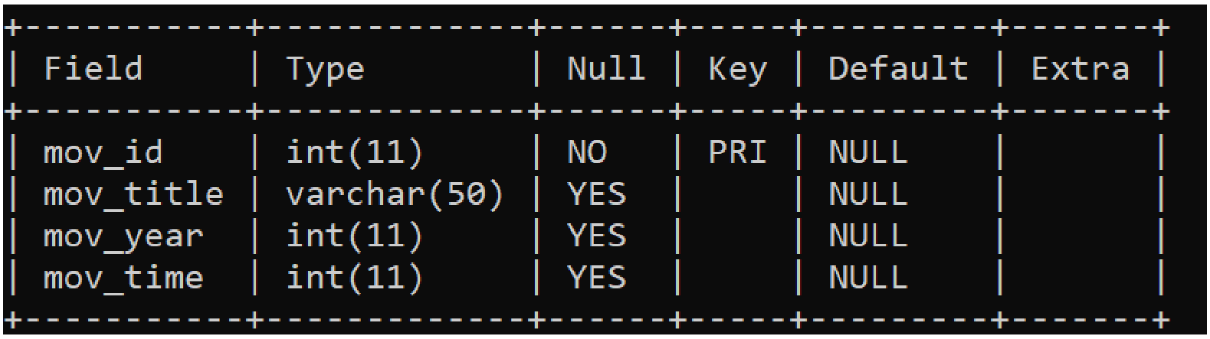
This database contains 7 tables:

1. movie
2. actor
3. genres
4. director
5. reviewer
6. rating
7. cast

Description of tables:

Movie:

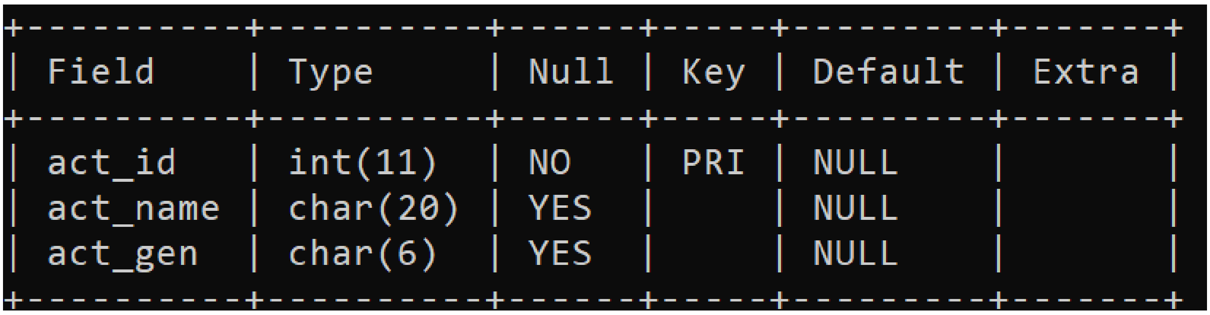
- mov_id – this is the unique ID for each movie
- mov_name – this column represents the name of the movie
- mov_year – this is the year of making the movie
- mov_time – duration of the movie i.e. how long it was running



```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| mov_id    | int(11)     | NO   | PRI | NULL    |       |
| mov_title | varchar(50) | YES  |     | NULL    |       |
| mov_year  | int(11)     | YES  |     | NULL    |       |
| mov_time  | int(11)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

Actor:

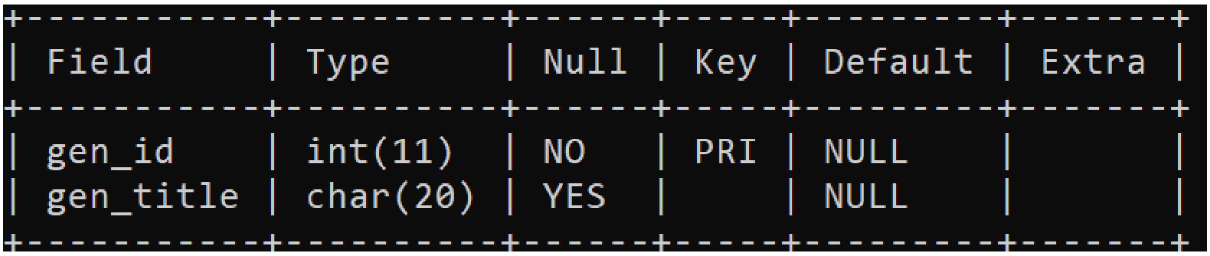
- act_id – this is a unique ID for each actor
- act_name – this is the name of each actor
- act_gen– this is the gender of each actor



```
+----------+----------+------+-----+---------+-------+
| Field    | Type     | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| act_id   | int(11)  | NO   | PRI | NULL    |       |
| act_name | char(20) | YES  |     | NULL    |       |
| act_gen  | char(6)  | YES  |     | NULL    |       |
+----------+----------+------+-----+---------+-------+
```

Genres:

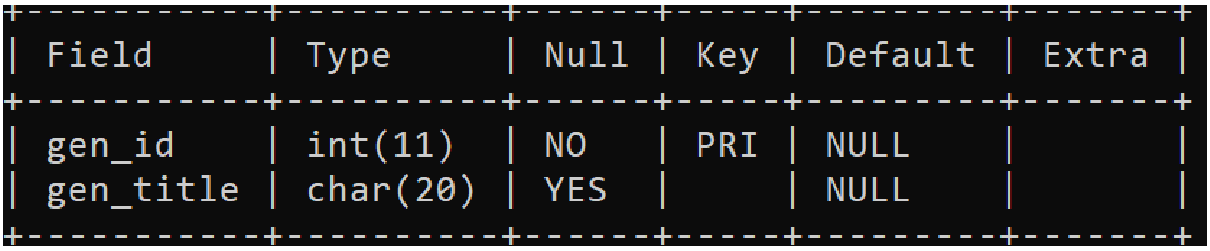
- gen_id – this is a unique ID for each genre
- gen_name – this is the description of the genres



```
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| gen_id    | int(11)  | NO   | PRI | NULL    |       |
| gen_title | char(20) | YES  |     | NULL    |       |
+-----------+----------+------+-----+---------+-------+
```

Director:

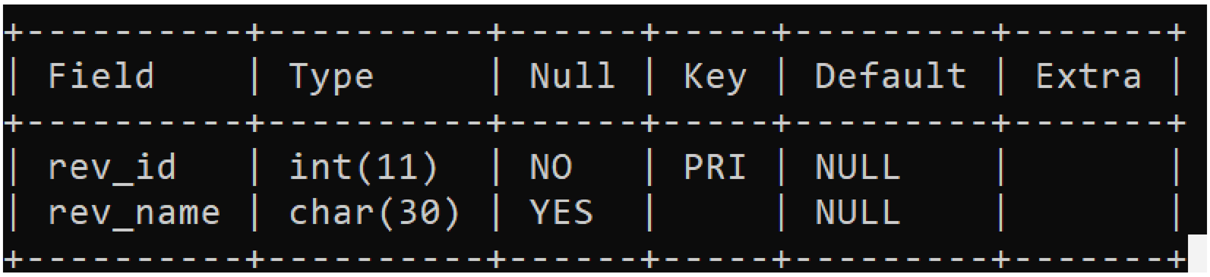
- dir_id – this is a unique ID for each director
- dir_fname – this is the first name of the director
- dir_lname – this is the last name of the director



```
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| gen_id    | int(11)  | NO   | PRI | NULL    |       |
| gen_title | char(20) | YES  |     | NULL    |       |
+-----------+----------+------+-----+---------+-------+
```

Reviewer:

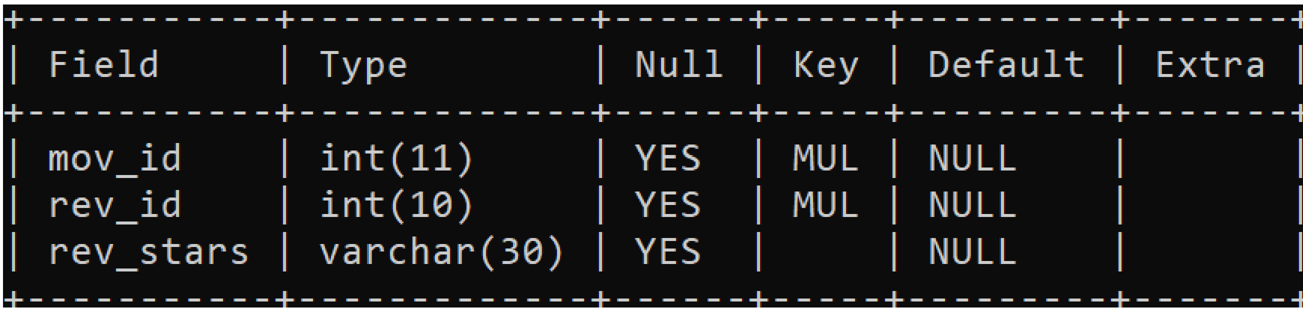
- rev_id – this is the unique ID for each reviewer
- rev_name – this is the name of the reviewer



```
+----------+----------+------+-----+---------+-------+
| Field    | Type     | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| rev_id   | int(11)  | NO   | PRI | NULL    |       |
| rev_name | char(30) | YES  |     | NULL    |       |
+----------+----------+------+-----+---------+-------+
```

Rating:

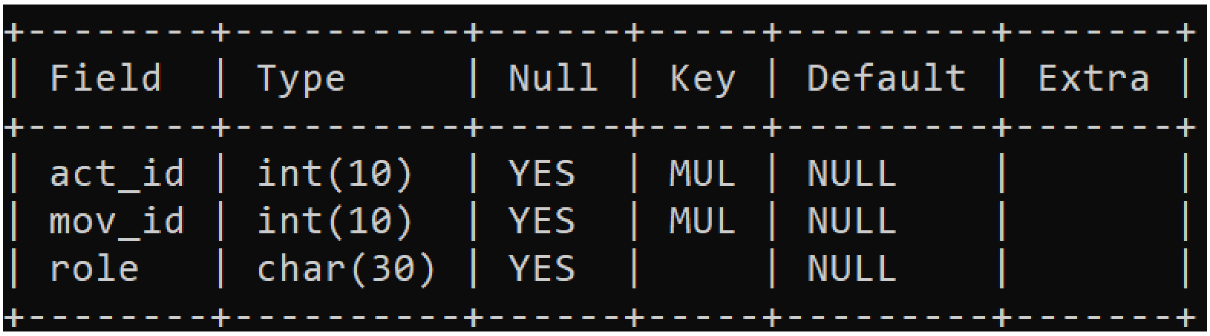
- mov_id –this is the ID of the movie, which is referencing the mov_id column of the table movie
- rev_id – this is the ID of the reviewer, which is referencing the rev_id column of the table reviewer
- rev_stars – this is indicating how many stars a reviewer rated for a review of a movie



```
+-----------+-------------+------+-----+---------+-------
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------
| mov_id    | int(11)     | YES  | MUL | NULL    |       |
| rev_id    | int(10)     | YES  | MUL | NULL    |       |
| rev_stars | varchar(30) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------
```

Cast:

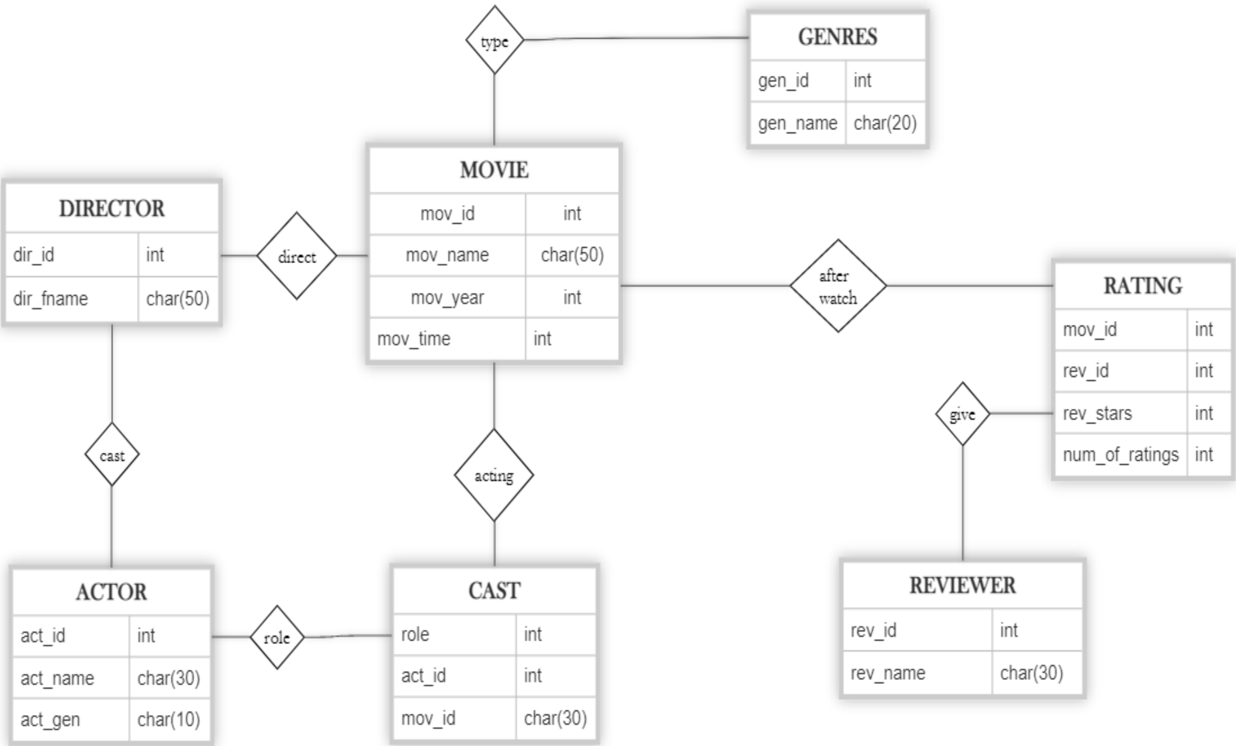
- act_id – this is ID of actor, which is referencing the act_id column of actor table
- mov_id – this is the ID of the movie, which is referencing the mov_id column of the table movie
- role – this is the name of a character in the movie, an actor acted for that character



```
+--------+----------+------+-----+---------+-------+
| Field  | Type     | Null | Key | Default | Extra |
+--------+----------+------+-----+---------+-------+
| act_id | int(10)  | YES  | MUL | NULL    |       |
| mov_id | int(10)  | YES  | MUL | NULL    |       |
| role   | char(30) | YES  |     | NULL    |       |
+--------+----------+------+-----+---------+-------+
```

How these tables/entities are related to each other is shown pictorially through ER diagram, i.e., Entity Relation (ER) Diagram.

ER DIAGRAM OF MOVIE DATABASE SYSTEM



Commands:

- **Create Databse:**

  ```
  Create database project;
  ```

- **Select Database;**

  ```
  Use project;
  ```

- **Create table named Movie:**

  ```
  create table movies1(mov_id int primary key auto_increment,
  mov_title char(50),
  mov_year int,
  mov_time int);
  ```

- **Create table named Actors:**

  ```
  create table actors (act_id int primary key,
  act_name char(20),
  act_gen char(6));
  ```

- **Create table named Directors:**

  ```
  create table directors (dir_id int primary key,
  dir_name char(20));
  ```

- **Create table named Genre:**

  ```
  create table genre (gen_id int primary key,
  gen_title char(20));
  ```

- **Create table named Rating:**

  ```
  create table rating (mov_id int,
  ```

```
rev_id int(10),
rev_stars int(5),
foreign key(mov_id) references movies1(mov_id),
foreign key (rev_id) references reviewers(rev_id));
```

- **Create table named Riviewers**:

```
create table reviewers (rev_id int primary key,
rev_name char(30));
```

- **Create table named Casts:**

```
create table casts (act_id int(10),
mov_id int(10),
role char(30),
foreign key (act_id) references actors(act_id),
foreign key(mov_id) references movies1(mov_id));
```

- **Populate "Movie" table:**

```
insert into movies1 values (1,"36_Farmhouse",2022,140),
(2,"Hai Tujhe Salaam India",2022,120),
(3,"Looop Lapeta",2022,125),
(4,"Gehraiyaan",2022,135),
(5,"Badhaai Do",2022,120),
(6,"Gangubai Kathiawadi",2022,110),
(7,"Love Hostel",2022,120),
(8,"Jhund",2022,120),
(9,"Toolsidas Junior",2022,105),
(10,"Radhe Shyam",2022,108),
(11,"The Kashmir File",2022,130),
(12,"Bachhan Pandey",2022,120),
(13,"Jalsa",2022,120),
(14,"Sharmaji Namkeen",2022,130),
(15,"Attack: Part1",2022,125),
(16,"Kaun Pravin Tambe?",2022,135),
(17,"CobaltBlue",2022,125),
(18,"Dasvi",2022,130),
(19,"Hurdang",2022,120),
(20,"Jersey",2022,130),
(21,"Operation Romeo",2022,120),
(22,"Runway 36",2022,125),
(23,"Heropanti 2",2022,120),
(24,"Mere Desh ki Dharti",2022,125),
(25,"Thar",2022,130),
(26,"Jayeshbhai Jorda",2022,110),
(27,"Bhool Bhulaiyaa 2",2022,120),
(28,"Dhaakad",2022,135),
(29,"Anek",2022,130),
(30,"Dehati Disco",2022,120);
```

- **Populate “actors” table:**

```
insert into actors values (001,'Vijay Raaz ','M'),
(002,'Aarya Babbar','M'),
(003,'Taapsee Pannu','F'),
(004,'Deepika Padukone','F'),
(005,'Rajkummar Rao','M'),
(006,'Alia Bhatt','F'),
(007,'Sanya Malhotra','F'),
(008,'Amitabh Bachchan','M'),
(009,'Sanjay Dutt','M'),
(010,'Prabhas','M'),
(011,'Mithun Chakraborty','M'),
(012,'Akshay Kumar','M'),
(013,'Vidya Balan','F'),
(014,'Rishi Kapoor','M'),
(015,'John Abraham','M'),
(016,'Shreyas Talpade','F'),
(017,'Prateik Babbar','M'),
(018,'Abhishek Bachchan','M'),
(019,'Nikhil Nagesh Bhat','M'),
(020,'Gowtam Tinnanuri','M'),
(021,'Shashant Shah','M'),
(022,'Ajay Devgn','M'),
(023,'Ahmed Khan','M'),
(024,'Faraz Haider','M'),
(025,'Faraz Haider','M'),
(026,'Raj Singh Chaudhary','M'),
(027,'Divyang Thakkar','M'),
(028,'Anees Bazmee','M'),
(029,'Razy Ghai','M'),
(030,'Anubhav Sinha  ','M');
```

- **Populate “Directors” table:**

```
insert into directors values(101,'Ram Ramesh Sharm'),
(102,'Avanish Kumar'),
(103,'Aakash Bhatia'),
(104,'Shakun Bhatia'),
(105,'Harshavardhan Kulkarni'),
(106,'Sanjay Leela Bhansali'),
(107,'Shanker Raman'),
(108,'Nagraj Manjule'),
(109,'Mridul '),
(110,'Radha Krishna Kumar'),
(111,'Vivek Agnihotri'),
(112,'Farhad Samji')
(113,'Suresh Triveni'),
(114,'Hitesh Bhatia'),
(115,'Lakshya Raj Anand'),
(116,'Jayprad Desai'),
(117,'Sachin Kundalkar'),
(118,'Tushar Jalota'),
(119,'Nikhil Nagesh Bhat'),
(120,'Gowtam Tinnanuri'),
(121,'Shashant Shah'),
(122,'Ajay Devgn'),
(123,'Ahmed Khan'),
(124,'Faraz Haider'),
(125,'Raj Singh Chaudhary'),
(126,'Divyang Thakkar'),
(127,'Anees Bazmee'),
(128,'Razy Ghai'),
(129,'Anubhav Sinha '),
(130,'manjo sharma');
```

- **Populate “Genre” table:**

```
insert into genre values (1001,'Action'),
(1002,'Adventure'),
(1003,'Animation'),
(1004,'Biography'),
(1005,'Comedy'),
(1006,'Crime'),
(1007,'Drama'),
(1008,'Horror'),
(1009,'Music'),
(1010,'Mystery'),
(1011,'Romance'),
(1012,'Thriller'),
(1013,'War'),
(1014,"Comedy"),
(1015,"Cricket"),
(1016,"Romance"),
(1017,"Cartoon"),
(1018,"Action"),
(1019,"Music"),
(1020,"War"),
(1021,"Romance"),
(1022,"Dance"),
(1023,"Darama"),
(1024,"Non Fiction"),
(1025,"Fiction"),
(1026,"Poetry"),
(1027,"War"),
(1028,"Action"),
(1029,"Darama"),
(1030,"Dance");
```

- **Populate “Reviewers” table:**

```
insert into reviewers values(901,"Maaz"),
(902,"anas"),
(903,"hamza"),
(904,"eesa"),
(905,"shadab"),
(906,"harish"),
(907,"zain"),
(908,"naaz"),
(909,"shaam"),
(910,"aftab"),
(911,"farhan"),
(912,'Paul'),
(913,'Woody'),
(914,'Hayao'),
(915,'Frank'),
(916,'Sam'),
(917,'James'),
(918,'Gus',),
(919,'John'),
(920,'Danny'),
(921,'Christopher'),
(922,'Richard'),
(923,'Kevin'),
(924,'Peter'),
(925,'Ridley'),
(926,'Kubrick'),
(927,'Singer'),
(928,'Polanski'),
(929,'Bryan'),
(930,'Roman');
```

- **Populate “rating” table:**

```
insert into rating values
(1,901,4.5),
(2,902,4.1),
(3,903,4.4),
(4,904,4.5),
(5,905,4.2),
(6,906,4.5),
(7,907,4.5),
(8,908,4.6),
(9,909,4.7),
(10,910,4.5),
(11,911,4.5),
(12,912,4.3),
(13,913,4.6),
(14,914,4.0),
(15,915,4.5),
(16,916,4.1),
(17,917,4.3),
(18,918,4.5),
(19,919,4.5),
(20,920,4.5),
(21,921,4.4),
(22,922,4.4),
(23,923,4.1),
(24,924,4.1),
(25,925,4.4),
(26,926,4.4),
(28,928,4.1),
(29,929,4.4),
(30,930,3.9);
```

- **Populate "casts" table:**

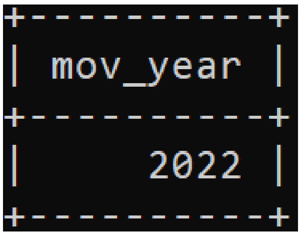
```
insert into casts values(1,1,"hero"),
(2,2,"hero"),
(3,3,"heroien"),
(4,4,"heroien"),
(5,5,"hero"),
(6,6,"heroien"),
(7,7,"heroien"),
(8,8,"hero"),
(9,9,"hero"),
(10,10,"hero"),
(11,11,"hero")
(12,12,"hero"),
(13,13,"heroien"),
(14,14,"hero"),
(15,15,"hero"),
(16,16,"heroien"),
(17,17,"hero"),
(18,18,"hero"),
(19,19,"hero"),
(20,20,"hero"),
(21,21,"hero"),
(22,22,"hero"),
(23,23,"hero"),
(24,24,"hero"),
(25,25,"hero"),
(26,26,"hero"),
(27,27,"hero"),
(28,28,"hero"),
(29,29,"hero"),
(30,30,"hero");
```

Basic Queries:

1.From the following table, write a SQL query to find when the movie 'looop lapeta ' released. Return movie year.

select mov_year from movies1 where mov_title='Looop Lapeta';

Output:

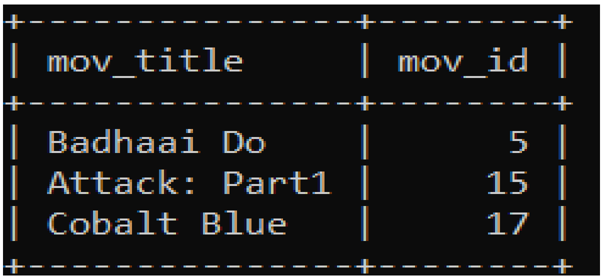


```
+----------+
| mov_year |
+----------+
|     2022 |
+----------+
```

2.From the following table, write a SQL query to find the movies with ID 5 or 15 or 17. Return movie title.

select mov_title,mov_id from movies1 where mov_id in (5,15,17);

Output:

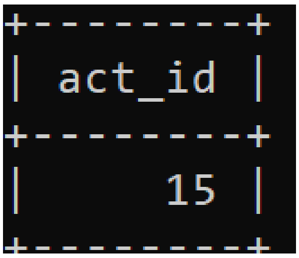


```
+---------------+--------+
| mov_title     | mov_id |
+---------------+--------+
| Badhaai Do    |      5 |
| Attack: Part1 |     15 |
| Cobalt Blue   |     17 |
+---------------+--------+
```

3. From the following table, write a SQL query to find those actors with the name 'john abraham' and gender 'M'. Return actor ID.

select act_id from actors where act_name='John Abraham' AND act_gen="M";

Output:



```
+--------+
| act_id |
+--------+
|     15 |
+--------+
```

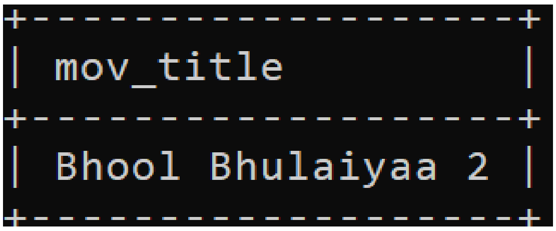
Sub-Queries:

1.From the following tables, write a SQL query to find the movies without any rating. Return movie title.

select mov_title from movies1 where mov_id not in

(select mov_id from rating);

Output:

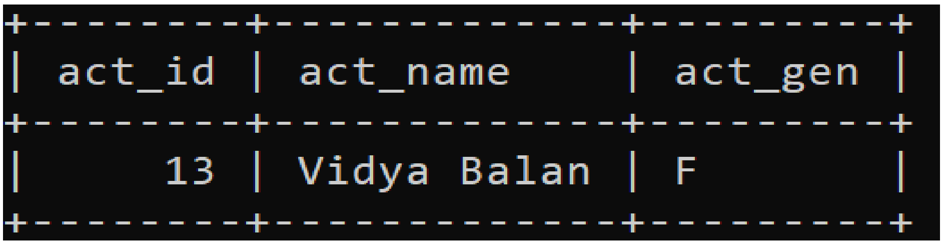


```
+--------------------+
| mov_title          |
+--------------------+
| Bhool Bhulaiyaa 2  |
+--------------------+
```

2.From the following tables, write a SQL query to find the actors who played a role in the movie 'jalsa'. Return all the fields of actor table.

select * from actors where act_id in (select act_id from casts where mov_id in (select mov_id from movies1 where mov_title='Jalsa'));

Output:

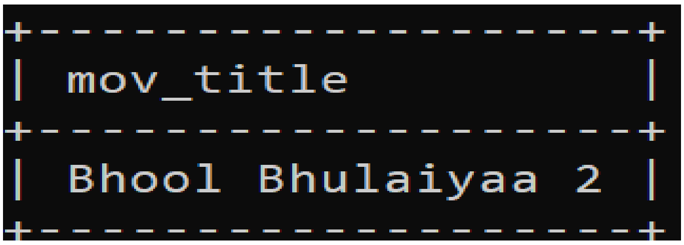


```
+--------+--------------+---------+
| act_id | act_name     | act_gen |
+--------+--------------+---------+
|     13 | Vidya Balan  | F       |
+--------+--------------+---------+
```

3. From the following table, write a SQL query to search for movies that do not have any ratings. Return movie title.

select distinct mov_title from movies1 where mov_id in (select mov_id from movies1 where mov_id not in (select mov_id from rating));

Output:

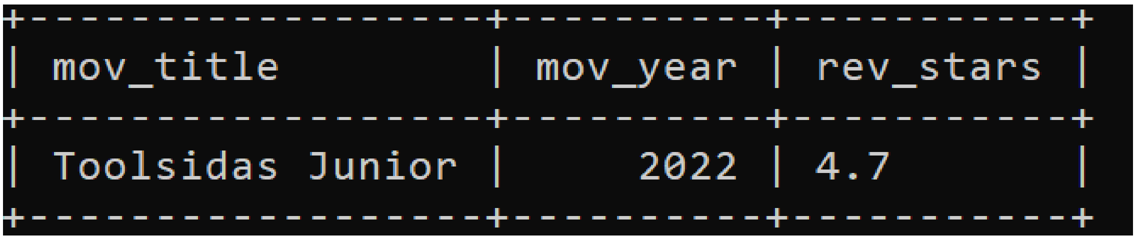


```
+-------------------+
| mov_title         |
+-------------------+
| Bhool Bhulaiyaa 2 |
+-------------------+
```

4. ***From the following tables, write a SQL query to find the highest-rated movies. Return movie title, movie year, review stars.***

select mov_title, mov_year, rev_stars from movies1 natural join rating where rev_stars = (select max(rev_stars) from rating);

Output:



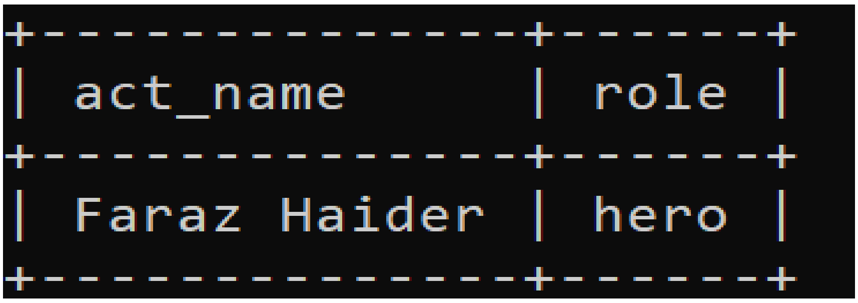
```
+------------------+----------+-----------+
| mov_title        | mov_year | rev_stars |
+------------------+----------+-----------+
| Toolsidas Junior |     2022 | 4.7       |
+------------------+----------+-----------+
```

Joins:

1. From the following table, write a SQL query to find out who was cast in the movie 'thar'. Return actor name and role

select act_name,role from actors join casts on actors.act_id=casts.act_id join movies1 on casts.mov_id=movies1.mov_id AND movies1.mov_title='Thar';

Output:



```
+--------------+------+
| act_name     | role |
+--------------+------+
| Faraz Haider | hero |
+--------------+------+
```

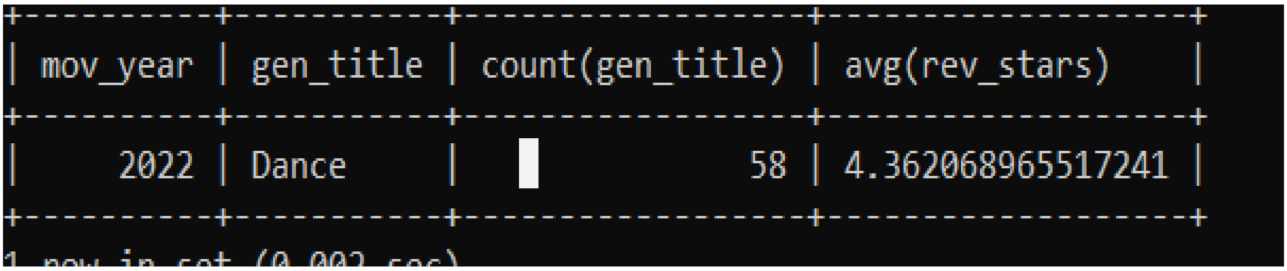
2. From the following tables, write a SQL query to find the years when most of the ‘Dance Movies’ produced.

Count the number of generic title and compute their average rating. Group the result set on

movie release year, generic title. Return movie year, generic title, number of generic title and average rating.

select mov_year,gen_title,count(gen_title), avg(rev_stars) from movies1 natural join genre natural join rating where gen_title='Dance' group by mov_year,gen_title;

Output:

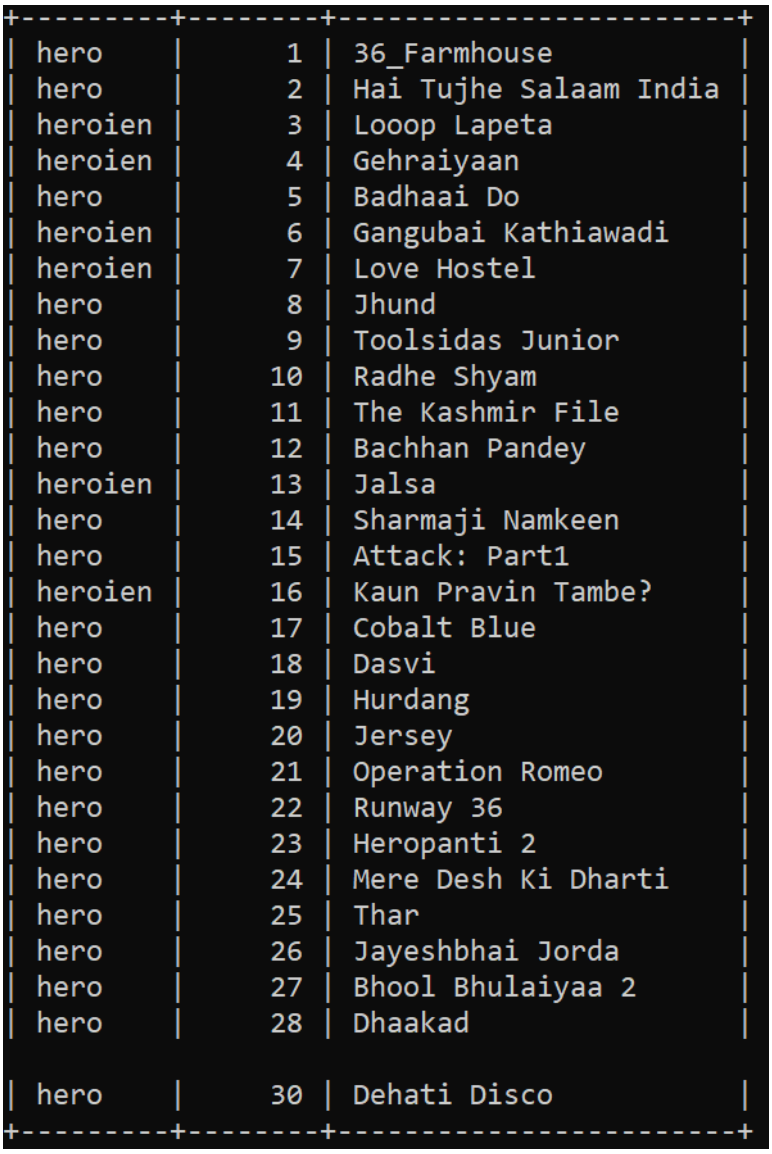


```
+----------+-----------+------------------+------------------+
| mov_year | gen_title | count(gen_title) | avg(rev_stars)   |
+----------+-----------+------------------+------------------+
|     2022 | Dance     |               58 | 4.362068965517241 |
+----------+-----------+------------------+------------------+
1 row in set (0.002 sec)
```

3. From the following table, write a SQL query to display role,actor,and movie name using inner join.

select role,act_id,mov_title from movies1 inner join casts on movies1.mov_id=casts.mov_id;

Output:



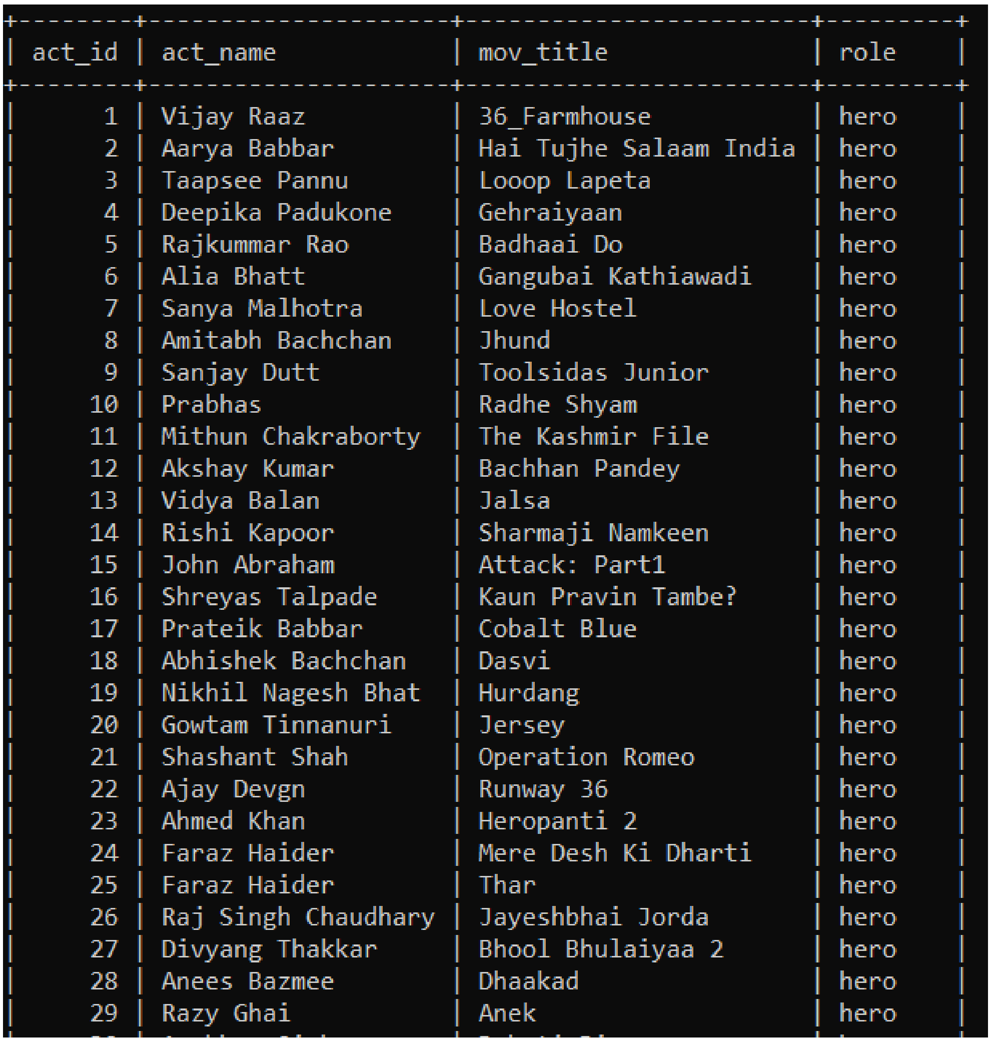
```
+---------+--------+-----------------------+
| hero    |      1 | 36_Farmhouse          |
| hero    |      2 | Hai Tujhe Salaam India |
| heroien |      3 | Looop Lapeta          |
| heroien |      4 | Gehraiyaan            |
| hero    |      5 | Badhaai Do            |
| heroien |      6 | Gangubai Kathiawadi   |
| heroien |      7 | Love Hostel           |
| hero    |      8 | Jhund                 |
| hero    |      9 | Toolsidas Junior      |
| hero    |     10 | Radhe Shyam           |
| hero    |     11 | The Kashmir File      |
| hero    |     12 | Bachhan Pandey        |
| heroien |     13 | Jalsa                 |
| hero    |     14 | Sharmaji Namkeen      |
| hero    |     15 | Attack: Part1         |
| heroien |     16 | Kaun Pravin Tambe?    |
| hero    |     17 | Cobalt Blue           |
| hero    |     18 | Dasvi                 |
| hero    |     19 | Hurdang               |
| hero    |     20 | Jersey                |
| hero    |     21 | Operation Romeo       |
| hero    |     22 | Runway 36             |
| hero    |     23 | Heropanti 2           |
| hero    |     24 | Mere Desh Ki Dharti   |
| hero    |     25 | Thar                  |
| hero    |     26 | Jayeshbhai Jorda      |
| hero    |     27 | Bhool Bhulaiyaa 2     |
| hero    |     28 | Dhaakad               |

| hero    |     30 | Dehati Disco          |
+---------+--------+-----------------------+
```

Views:

1.create a view using actor and director table, show act_name and dir_name

create view views AS select actors.act_id,actors.act_name,movies1.mov_title,casts.role from actors,movies1,casts where movies1.mov_id=actors.act_id;

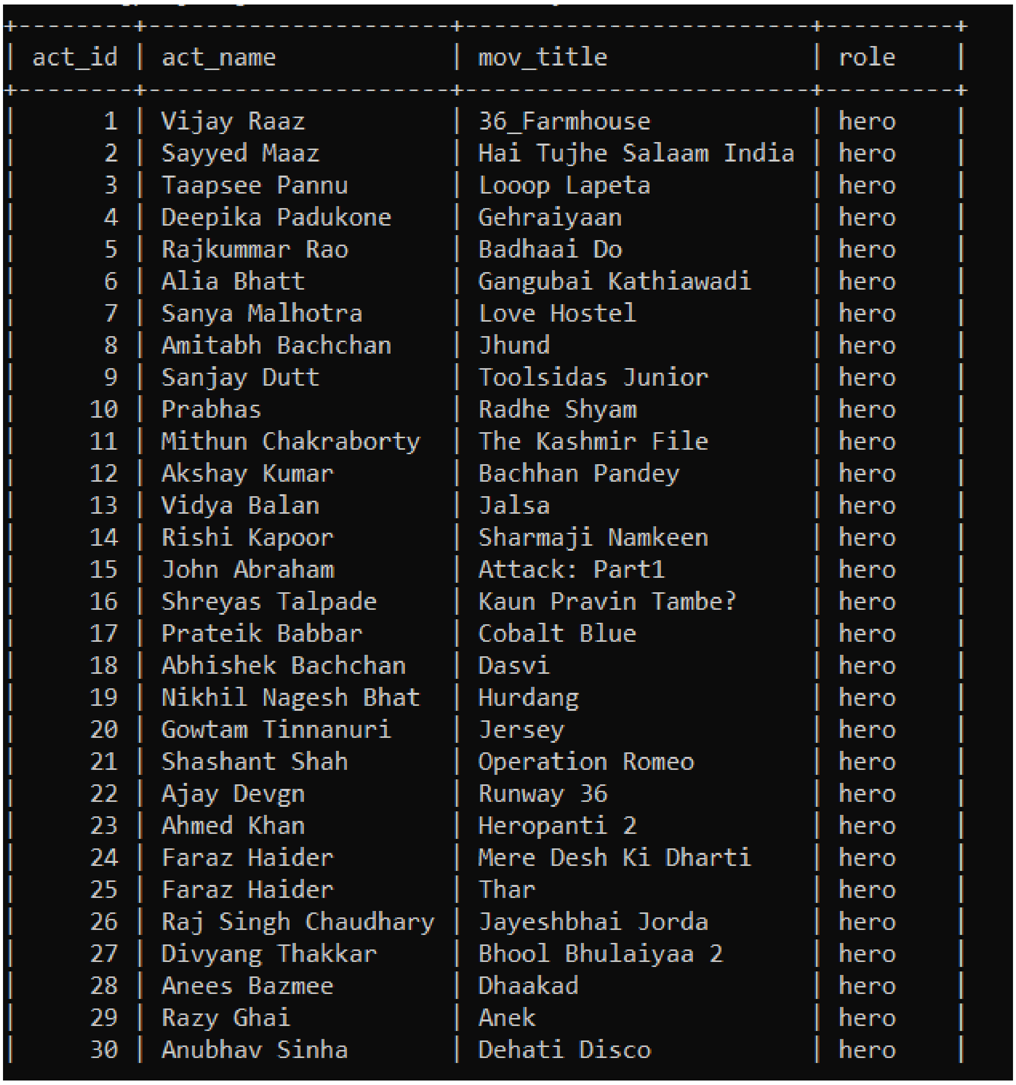


```
+--------+---------------------+------------------------+---------+
| act_id | act_name            | mov_title              | role    |
+--------+---------------------+------------------------+---------+
|      1 | Vijay Raaz          | 36_Farmhouse           | hero    |
|      2 | Aarya Babbar        | Hai Tujhe Salaam India | hero    |
|      3 | Taapsee Pannu       | Looop Lapeta           | hero    |
|      4 | Deepika Padukone    | Gehraiyaan             | hero    |
|      5 | Rajkummar Rao       | Badhaai Do             | hero    |
|      6 | Alia Bhatt          | Gangubai Kathiawadi    | hero    |
|      7 | Sanya Malhotra      | Love Hostel            | hero    |
|      8 | Amitabh Bachchan    | Jhund                  | hero    |
|      9 | Sanjay Dutt         | Toolsidas Junior       | hero    |
|     10 | Prabhas             | Radhe Shyam            | hero    |
|     11 | Mithun Chakraborty  | The Kashmir File       | hero    |
|     12 | Akshay Kumar        | Bachhan Pandey         | hero    |
|     13 | Vidya Balan         | Jalsa                  | hero    |
|     14 | Rishi Kapoor        | Sharmaji Namkeen       | hero    |
|     15 | John Abraham        | Attack: Part1          | hero    |
|     16 | Shreyas Talpade     | Kaun Pravin Tambe?     | hero    |
|     17 | Prateik Babbar      | Cobalt Blue            | hero    |
|     18 | Abhishek Bachchan   | Dasvi                  | hero    |
|     19 | Nikhil Nagesh Bhat  | Hurdang                | hero    |
|     20 | Gowtam Tinnanuri    | Jersey                 | hero    |
|     21 | Shashant Shah       | Operation Romeo        | hero    |
|     22 | Ajay Devgn          | Runway 36              | hero    |
|     23 | Ahmed Khan          | Heropanti 2            | hero    |
|     24 | Faraz Haider        | Mere Desh Ki Dharti    | hero    |
|     25 | Faraz Haider        | Thar                   | hero    |
|     26 | Raj Singh Chaudhary | Jayeshbhai Jorda       | hero    |
|     27 | Divyang Thakkar     | Bhool Bhulaiyaa 2      | hero    |
|     28 | Anees Bazmee        | Dhaakad                | hero    |
|     29 | Razy Ghai           | Anek                   | hero    |
```

2.update act_name =“aarya babbar” to “Sayyed Maaz” in view table:

update views set act_name='Sayyed Maaz' where act_id=2;

Output:



```
+--------+---------------------+-------------------------+---------+
| act_id | act_name            | mov_title               | role    |
+--------+---------------------+-------------------------+---------+
|      1 | Vijay Raaz          | 36_Farmhouse            | hero    |
|      2 | Sayyed Maaz         | Hai Tujhe Salaam India  | hero    |
|      3 | Taapsee Pannu       | Looop Lapeta            | hero    |
|      4 | Deepika Padukone    | Gehraiyaan              | hero    |
|      5 | Rajkummar Rao       | Badhaai Do              | hero    |
|      6 | Alia Bhatt          | Gangubai Kathiawadi     | hero    |
|      7 | Sanya Malhotra      | Love Hostel             | hero    |
|      8 | Amitabh Bachchan    | Jhund                   | hero    |
|      9 | Sanjay Dutt         | Toolsidas Junior        | hero    |
|     10 | Prabhas             | Radhe Shyam             | hero    |
|     11 | Mithun Chakraborty  | The Kashmir File        | hero    |
|     12 | Akshay Kumar        | Bachhan Pandey          | hero    |
|     13 | Vidya Balan         | Jalsa                   | hero    |
|     14 | Rishi Kapoor        | Sharmaji Namkeen        | hero    |
|     15 | John Abraham        | Attack: Part1           | hero    |
|     16 | Shreyas Talpade     | Kaun Pravin Tambe?      | hero    |
|     17 | Prateik Babbar      | Cobalt Blue             | hero    |
|     18 | Abhishek Bachchan   | Dasvi                   | hero    |
|     19 | Nikhil Nagesh Bhat  | Hurdang                 | hero    |
|     20 | Gowtam Tinnanuri    | Jersey                  | hero    |
|     21 | Shashant Shah       | Operation Romeo         | hero    |
|     22 | Ajay Devgn          | Runway 36               | hero    |
|     23 | Ahmed Khan          | Heropanti 2             | hero    |
|     24 | Faraz Haider        | Mere Desh Ki Dharti     | hero    |
|     25 | Faraz Haider        | Thar                    | hero    |
|     26 | Raj Singh Chaudhary | Jayeshbhai Jorda        | hero    |
|     27 | Divyang Thakkar     | Bhool Bhulaiyaa 2       | hero    |
|     28 | Anees Bazmee        | Dhaakad                 | hero    |
|     29 | Razy Ghai           | Anek                    | hero    |
|     30 | Anubhav Sinha       | Dehati Disco            | hero    |
```

CONCLUSIONS:

This is the report which includes the assumptions, relational schema and finally all the SQL queries. The time I consumed to properly complete the report has increased my knowledge towards the database management system and helped me gain a lot of ideas and skills about SQL queries. I am very grateful and would like to appreciate the consistency support and help that I was able to get from my tutor.