MOVIE RECOMMENDATION SYSTEM (SQL MODULE PROJECT)

By

ANUVHAB BASU

DESCRIPTION:

1) The following database schema is designed to function as a backend storage database for an OTT platform.
2) By storing information in a relational database, all the tasks related to analyzing the data based on user preferences are done much more easily. SOME OF

THE BENEFITS of using this system over traditional RDBMS are as follows:

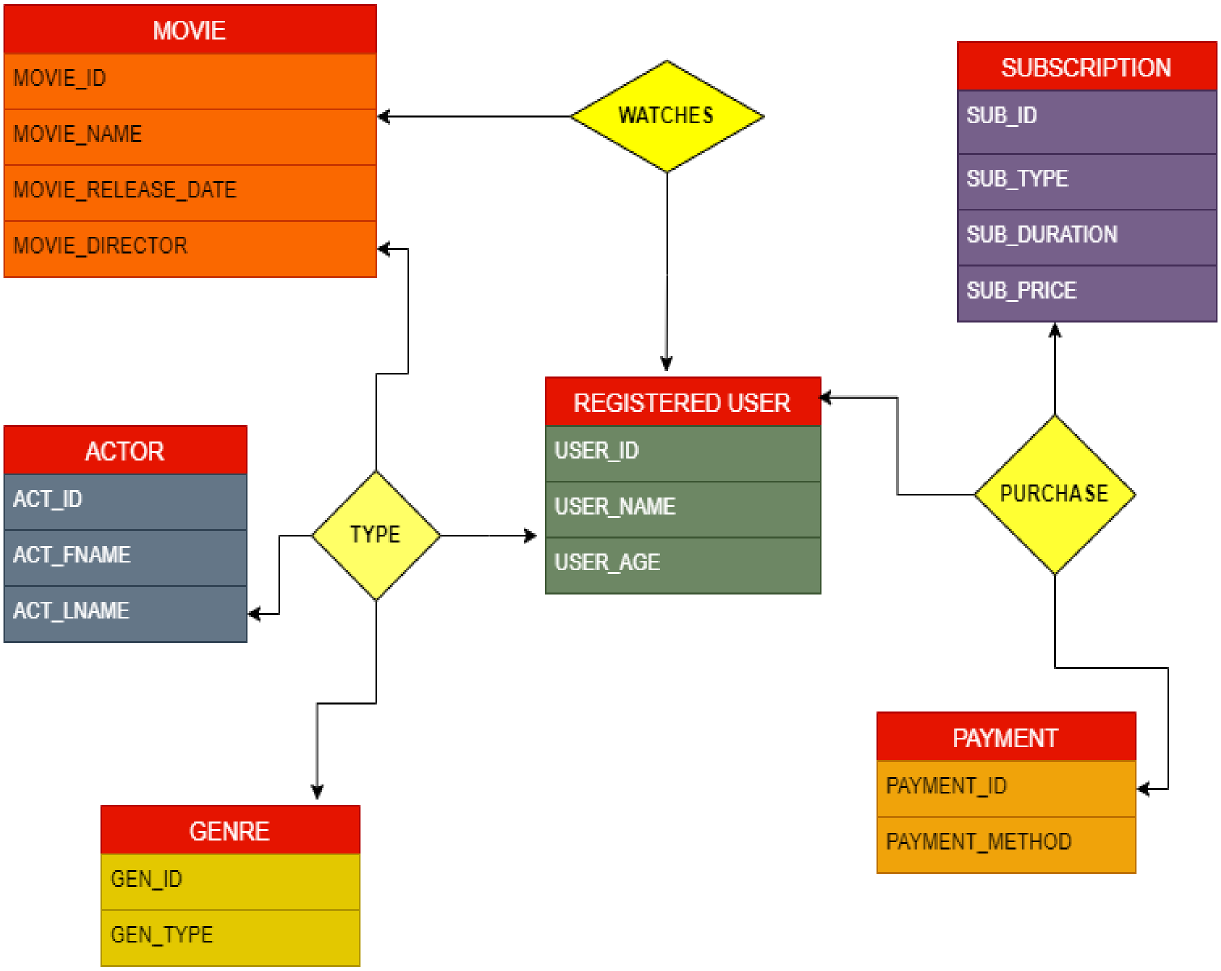
a) Updating and modifying Movie recommendations according to popularity and user choices.
b) Maintaining customer(user) details, their accounts and their activities is easy and efficient.
c) Calculations based on payment and subscriptions bought by the user are done very efficiently.
d) RDBMS provides many ways to analyze available data, thus helping in making suggestions and choices based on the choice of user and helps in making more informed decisions on other aspects of OTT movie recommendation system as well.

THE DATABASE CONTAINS 6 TABLES:

a) REGISTERED USER
b) MOVIE
c) ACTOR
d) GENRE
e) SUBSCRIPTION
f) PAYMENT

ENTITY RELATION DAIGRAM:

THE FOLLOWING TABLE IN THE NEXT PAGE SHOWS THE RELATION BETWEEN 6 TABLES IN THE DATABASE



MOVIE
MOVIE_ID
MOVIE_NAME
MOVIE_RELEASE_DATE
MOVIE_DIRECTOR

SUBSCRIPTION
SUB_ID
SUB_TYPE
SUB_DURATION
SUB_PRICE

ACTOR
ACT_ID
ACT_FNAME
ACT_LNAME

REGISTERED USER
USER_ID
USER_NAME
USER_AGE

PAYMENT
PAYMENT_ID
PAYMENT_METHOD

GENRE
GEN_ID
GEN_TYPE

1)*COMMANDS REQUIRED:*

1)CREATE TABLE, CREATE DATABASE.

Create table REGISTERED_ USER();

Create table MOVIE ();

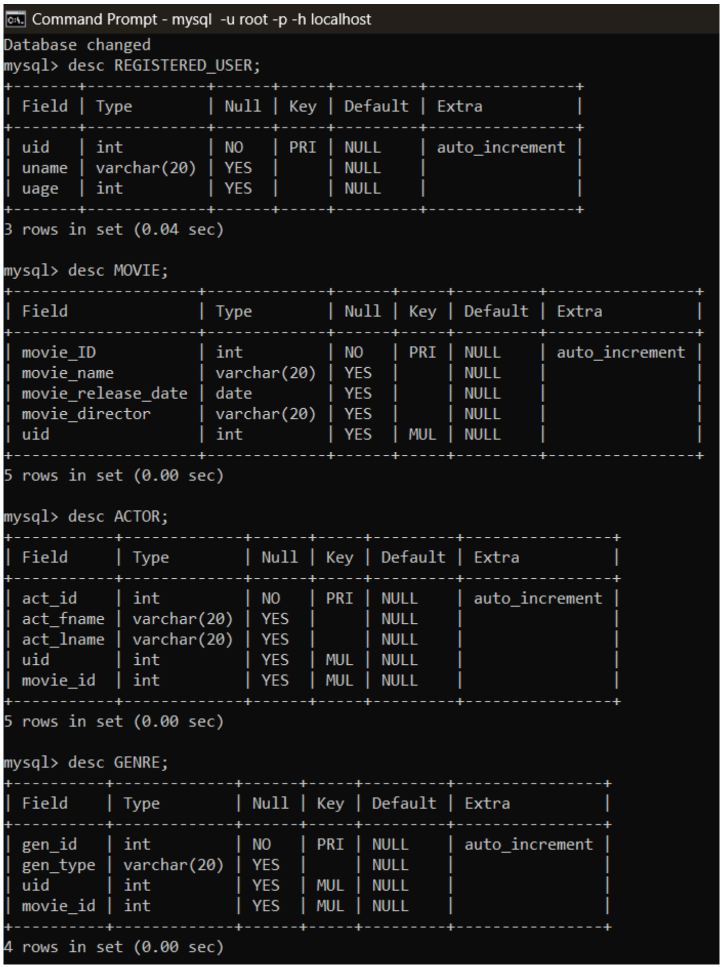
Create table ACTOR ();

Create table GENRE ();

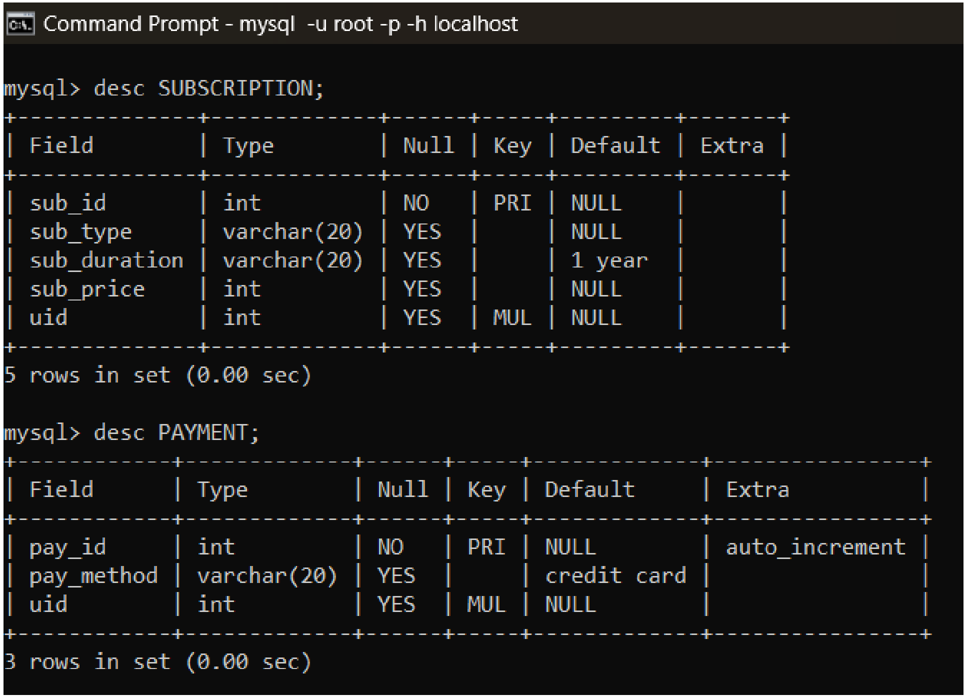
Create table SUBSCRIPTION ();

Create table PAYMENT ();

The following diagram in next page describes 6 tables as well as the keys and attributes used in tables:



```
Database changed
mysql> desc REGISTERED_USER;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| uid   | int         | NO   | PRI | NULL    | auto_increment |
| uname | varchar(20) | YES  |     | NULL    |                |
| uage  | int         | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
3 rows in set (0.04 sec)

mysql> desc MOVIE;
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| movie_ID           | int         | NO   | PRI | NULL    | auto_increment |
| movie_name         | varchar(20) | YES  |     | NULL    |                |
| movie_release_date | date        | YES  |     | NULL    |                |
| movie_director     | varchar(20) | YES  |     | NULL    |                |
| uid                | int         | YES  | MUL | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc ACTOR;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| act_id    | int         | NO   | PRI | NULL    | auto_increment |
| act_fname | varchar(20) | YES  |     | NULL    |                |
| act_lname | varchar(20) | YES  |     | NULL    |                |
| uid       | int         | YES  | MUL | NULL    |                |
| movie_id  | int         | YES  | MUL | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc GENRE;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| gen_id   | int         | NO   | PRI | NULL    | auto_increment |
| gen_type | varchar(20) | YES  |     | NULL    |                |
| uid      | int         | YES  | MUL | NULL    |                |
| movie_id | int         | YES  | MUL | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

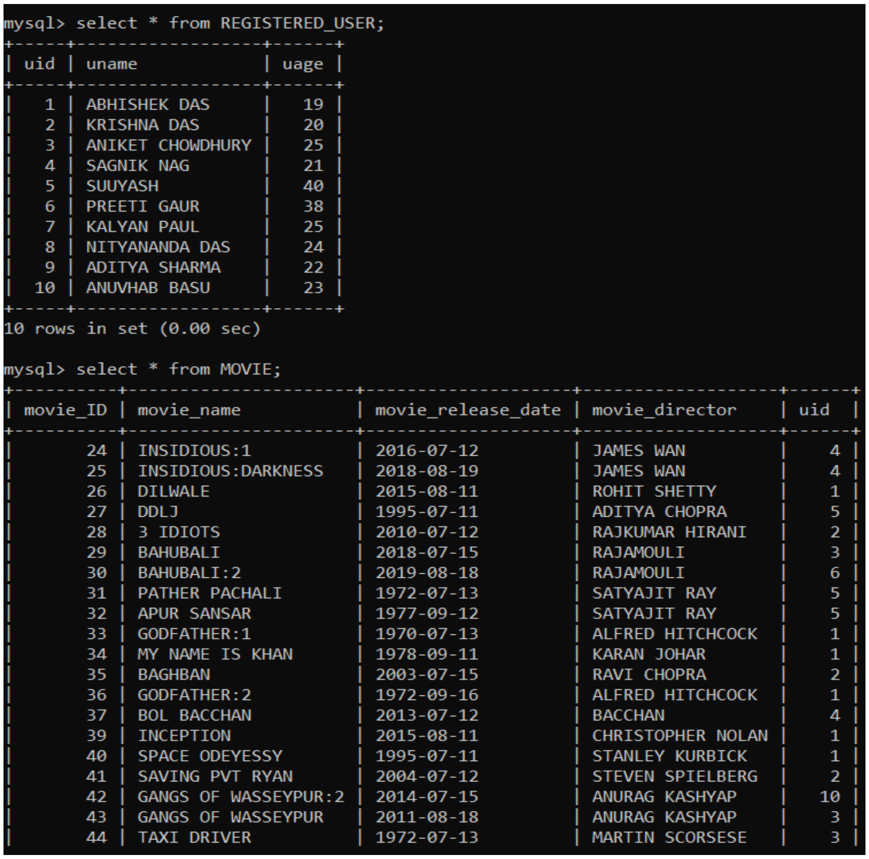


```
mysql> desc SUBSCRIPTION;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| sub_id       | int         | NO   | PRI | NULL    |       |
| sub_type     | varchar(20) | YES  |     | NULL    |       |
| sub_duration | varchar(20) | YES  |     | 1 year  |       |
| sub_price    | int         | YES  |     | NULL    |       |
| uid          | int         | YES  | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> desc PAYMENT;
+------------+-------------+------+-----+-------------+----------------+
| Field      | Type        | Null | Key | Default     | Extra          |
+------------+-------------+------+-----+-------------+----------------+
| pay_id     | int         | NO   | PRI | NULL        | auto_increment |
| pay_method | varchar(20) | YES  |     | credit card |                |
| uid        | int         | YES  | MUL | NULL        |                |
+------------+-------------+------+-----+-------------+----------------+
3 rows in set (0.00 sec)
```

INSERT VALUES INTO THE GIVEN TABLE:

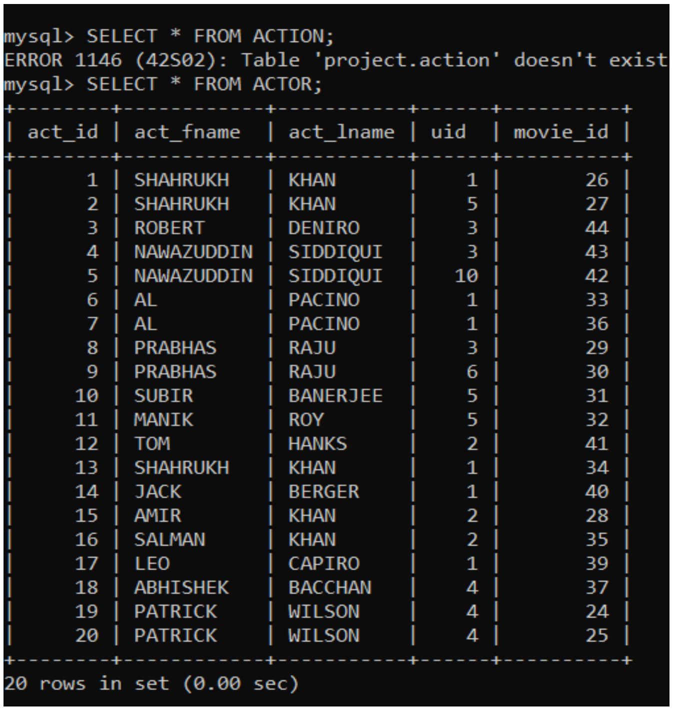
USING INSERT COMMAND TO INSERT VALUES IN THE TABLES. (using uid as a foreign key in all the tables)

THE FOLLOWING PICTURE SHOWS THE INSERTED VALUES IN THE TABLE:

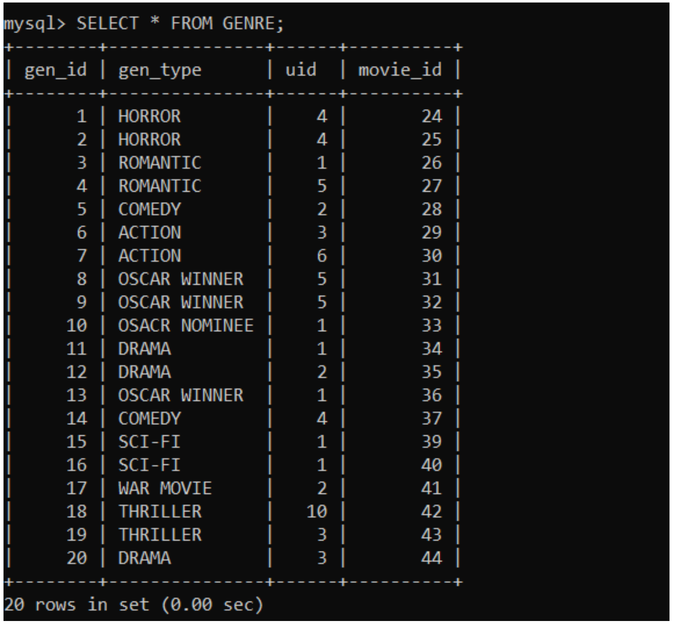


```
mysql> select * from REGISTERED_USER;
+-----+------------------+------+
| uid | uname            | uage |
+-----+------------------+------+
|   1 | ABHISHEK DAS     |   19 |
|   2 | KRISHNA DAS      |   20 |
|   3 | ANIKET CHOWDHURY |   25 |
|   4 | SAGNIK NAG       |   21 |
|   5 | SUUYASH          |   40 |
|   6 | PREETI GAUR      |   38 |
|   7 | KALYAN PAUL      |   25 |
|   8 | NITYANANDA DAS   |   24 |
|   9 | ADITYA SHARMA    |   22 |
|  10 | ANUVHAB BASU     |   23 |
+-----+------------------+------+
10 rows in set (0.00 sec)

mysql> select * from MOVIE;
+----------+----------------------+--------------------+-------------------+-----+
| movie_ID | movie_name           | movie_release_date | movie_director    | uid |
+----------+----------------------+--------------------+-------------------+-----+
|       24 | INSIDIOUS:1          | 2016-07-12         | JAMES WAN         |   4 |
|       25 | INSIDIOUS:DARKNESS   | 2018-08-19         | JAMES WAN         |   4 |
|       26 | DILWALE              | 2015-08-11         | ROHIT SHETTY      |   1 |
|       27 | DDLJ                 | 1995-07-11         | ADITYA CHOPRA     |   5 |
|       28 | 3 IDIOTS             | 2010-07-12         | RAJKUMAR HIRANI   |   2 |
|       29 | BAHUBALI             | 2018-07-15         | RAJAMOULI         |   3 |
|       30 | BAHUBALI:2           | 2019-08-18         | RAJAMOULI         |   6 |
|       31 | PATHER PACHALI       | 1972-07-13         | SATYAJIT RAY      |   5 |
|       32 | APUR SANSAR          | 1977-09-12         | SATYAJIT RAY      |   5 |
|       33 | GODFATHER:1          | 1970-07-13         | ALFRED HITCHCOCK  |   1 |
|       34 | MY NAME IS KHAN      | 1978-09-11         | KARAN JOHAR       |   1 |
|       35 | BAGHBAN              | 2003-07-15         | RAVI CHOPRA       |   2 |
|       36 | GODFATHER:2          | 1972-09-16         | ALFRED HITCHCOCK  |   1 |
|       37 | BOL BACCHAN          | 2013-07-12         | BACCHAN           |   4 |
|       39 | INCEPTION            | 2015-08-11         | CHRISTOPHER NOLAN |   1 |
|       40 | SPACE ODEYESSY       | 1995-07-11         | STANLEY KURBICK   |   1 |
|       41 | SAVING PVT RYAN      | 2004-07-12         | STEVEN SPIELBERG  |   2 |
|       42 | GANGS OF WASSEYPUR:2 | 2014-07-15         | ANURAG KASHYAP    |  10 |
|       43 | GANGS OF WASSEYPUR   | 2011-08-18         | ANURAG KASHYAP    |   3 |
|       44 | TAXI DRIVER          | 1972-07-13         | MARTIN SCORSESE   |   3 |
```

*(WE USE uid as a FOREIGN KEY IN ALL THE TABLES TO FORM RELATION between them and the REGISTERED USER)

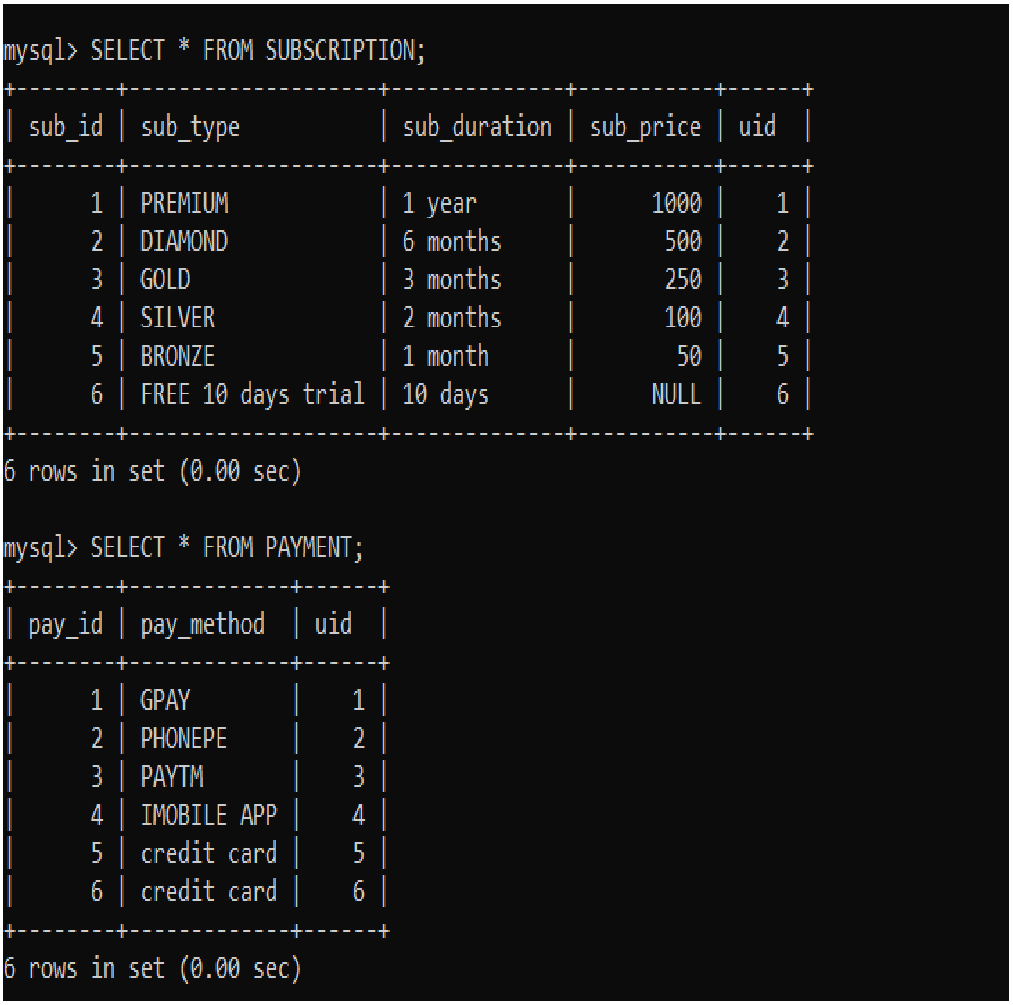


```
mysql> SELECT * FROM ACTION;
ERROR 1146 (42S02): Table 'project.action' doesn't exist
mysql> SELECT * FROM ACTOR;
+--------+------------+-----------+------+----------+
| act_id | act_fname  | act_lname | uid  | movie_id |
+--------+------------+-----------+------+----------+
|      1 | SHAHRUKH   | KHAN      |    1 |       26 |
|      2 | SHAHRUKH   | KHAN      |    5 |       27 |
|      3 | ROBERT     | DENIRO    |    3 |       44 |
|      4 | NAWAZUDDIN | SIDDIQUI  |    3 |       43 |
|      5 | NAWAZUDDIN | SIDDIQUI  |   10 |       42 |
|      6 | AL         | PACINO    |    1 |       33 |
|      7 | AL         | PACINO    |    1 |       36 |
|      8 | PRABHAS    | RAJU      |    3 |       29 |
|      9 | PRABHAS    | RAJU      |    6 |       30 |
|     10 | SUBIR      | BANERJEE  |    5 |       31 |
|     11 | MANIK      | ROY       |    5 |       32 |
|     12 | TOM        | HANKS     |    2 |       41 |
|     13 | SHAHRUKH   | KHAN      |    1 |       34 |
|     14 | JACK       | BERGER    |    1 |       40 |
|     15 | AMIR       | KHAN      |    2 |       28 |
|     16 | SALMAN     | KHAN      |    2 |       35 |
|     17 | LEO        | CAPIRO    |    1 |       39 |
|     18 | ABHISHEK   | BACCHAN   |    4 |       37 |
|     19 | PATRICK    | WILSON    |    4 |       24 |
|     20 | PATRICK    | WILSON    |    4 |       25 |
+--------+------------+-----------+------+----------+
20 rows in set (0.00 sec)
```



```
mysql> SELECT * FROM GENRE;
+--------+---------------+------+----------+
| gen_id | gen_type      | uid  | movie_id |
+--------+---------------+------+----------+
|      1 | HORROR        |    4 |       24 |
|      2 | HORROR        |    4 |       25 |
|      3 | ROMANTIC      |    1 |       26 |
|      4 | ROMANTIC      |    5 |       27 |
|      5 | COMEDY        |    2 |       28 |
|      6 | ACTION        |    3 |       29 |
|      7 | ACTION        |    6 |       30 |
|      8 | OSCAR WINNER  |    5 |       31 |
|      9 | OSCAR WINNER  |    5 |       32 |
|     10 | OSACR NOMINEE |    1 |       33 |
|     11 | DRAMA         |    1 |       34 |
|     12 | DRAMA         |    2 |       35 |
|     13 | OSCAR WINNER  |    1 |       36 |
|     14 | COMEDY        |    4 |       37 |
|     15 | SCI-FI        |    1 |       39 |
|     16 | SCI-FI        |    1 |       40 |
|     17 | WAR MOVIE     |    2 |       41 |
|     18 | THRILLER      |   10 |       42 |
|     19 | THRILLER      |    3 |       43 |
|     20 | DRAMA         |    3 |       44 |
+--------+---------------+------+----------+
20 rows in set (0.00 sec)
```

*(WE USE MOVIE_ID AS A FOREIGN KEY IN TABLE_ACTION and TABLE_GENRE TO FIND RELATION BETWEEN THEM)



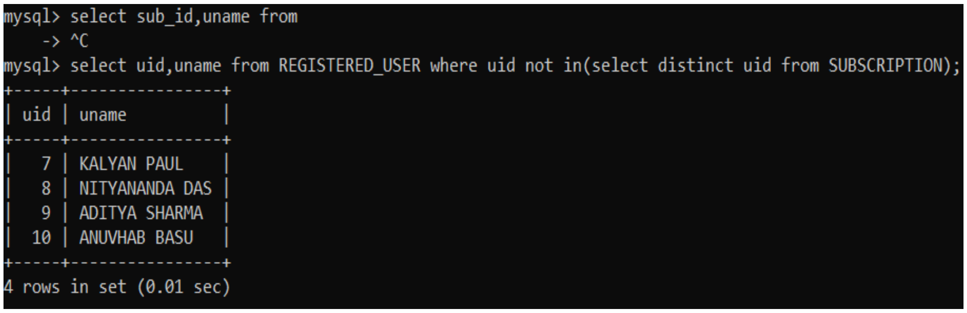
```
mysql> SELECT * FROM SUBSCRIPTION;
+--------+--------------------+--------------+-----------+------+
| sub_id | sub_type           | sub_duration | sub_price | uid  |
+--------+--------------------+--------------+-----------+------+
|      1 | PREMIUM            | 1 year       |      1000 |    1 |
|      2 | DIAMOND            | 6 months     |       500 |    2 |
|      3 | GOLD               | 3 months     |       250 |    3 |
|      4 | SILVER             | 2 months     |       100 |    4 |
|      5 | BRONZE             | 1 month      |        50 |    5 |
|      6 | FREE 10 days trial | 10 days      |      NULL |    6 |
+--------+--------------------+--------------+-----------+------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM PAYMENT;
+--------+-------------+------+
| pay_id | pay_method  | uid  |
+--------+-------------+------+
|      1 | GPAY        |    1 |
|      2 | PHONEPE     |    2 |
|      3 | PAYTM       |    3 |
|      4 | IMOBILE APP |    4 |
|      5 | credit card |    5 |
|      6 | credit card |    6 |
+--------+-------------+------+
6 rows in set (0.00 sec)
```

*(TABLES WITH VALUE INSERTED WITH USING 2 FOREIGN KEY, UID AND MOVIE_ID) *

2) ***SUB QUERIES***:

1) TO DISPLAY THE USERS WHO HAVE NOT TAKEN ANY SUBSCRIPTION:

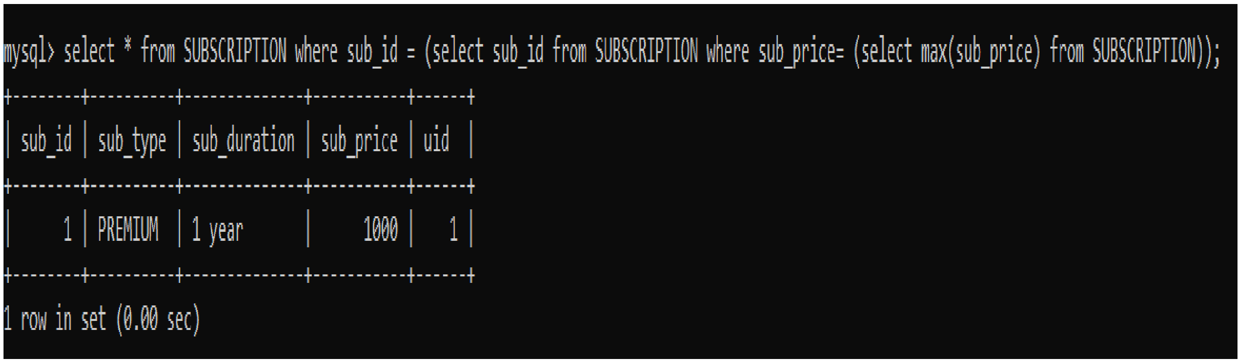
Select uid, uname from REGISTERED_USER where uid not in (select distinct uid from SUBSCRIPTION);



```
mysql> select sub_id,uname from
    -> ^C
mysql> select uid,uname from REGISTERED_USER where uid not in(select distinct uid from SUBSCRIPTION);
+-----+----------------+
| uid | uname          |
+-----+----------------+
|   7 | KALYAN PAUL    |
|   8 | NITYANANDA DAS |
|   9 | ADITYA SHARMA  |
|  10 | ANUVHAB BASU   |
+-----+----------------+
4 rows in set (0.01 sec)
```

2)TO DISPLAY THE MAXIMUM SUBSCRIPTION PRICE:

Select * from SUBSCRIPTION where sub_id in (select sub_id from SUBSCRIPTION where sub_price=(select max(sub_price) from SUBSCRIPTION));

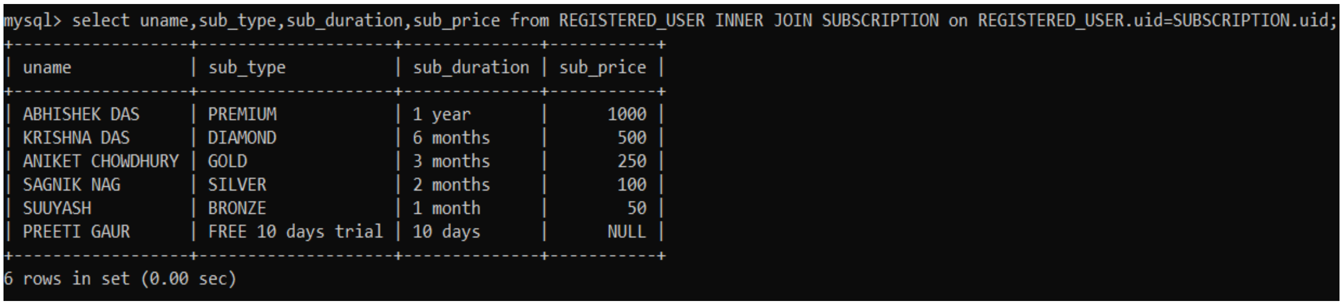


```
mysql> select * from SUBSCRIPTION where sub_id = (select sub_id from SUBSCRIPTION where sub_price= (select max(sub_price) from SUBSCRIPTION));
+--------+----------+--------------+-----------+-----+
| sub_id | sub_type | sub_duration | sub_price | uid |
+--------+----------+--------------+-----------+-----+
|      1 | PREMIUM  | 1 year       |      1000 |   1 |
+--------+----------+--------------+-----------+-----+
1 row in set (0.00 sec)
```

3) *JOINS*

1) TO DISPLAY THE USERS WHO HAVE BOUGHT THE SUBSCRIPTION ALONG WITH SUB_PRICE,SUB_DURATION:

SELECT uname,sub_type,sub_duration,sub_price from REGISTERED_USER inner join SUBSCRIPTION on REGISTERED_USER.uid=SUBSCRIPTION.uid;

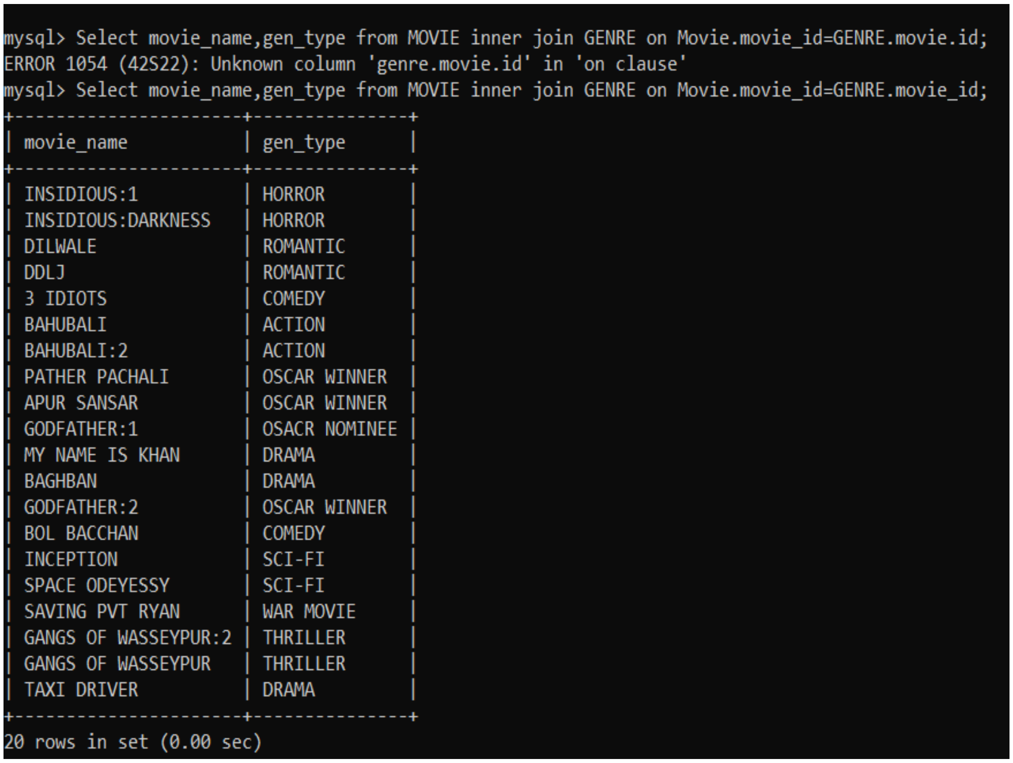


```
mysql> select uname,sub_type,sub_duration,sub_price from REGISTERED_USER INNER JOIN SUBSCRIPTION on REGISTERED_USER.uid=SUBSCRIPTION.uid;
+------------------+--------------------+--------------+-----------+
| uname            | sub_type           | sub_duration | sub_price |
+------------------+--------------------+--------------+-----------+
| ABHISHEK DAS     | PREMIUM            | 1 year       |      1000 |
| KRISHNA DAS      | DIAMOND            | 6 months     |       500 |
| ANIKET CHOWDHURY | GOLD               | 3 months     |       250 |
| SAGNIK NAG       | SILVER             | 2 months     |       100 |
| SUUYASH          | BRONZE             | 1 month      |        50 |
| PREETI GAUR      | FREE 10 days trial | 10 days      |      NULL |
+------------------+--------------------+--------------+-----------+
6 rows in set (0.00 sec)
```

2)TO DISPLAY LIST OF MOVIES ALONG WITH GENRE:

Select movie_name,gen_type from MOVIE inner join GENRE on Movie.movie_id=GENRE.movie.id;

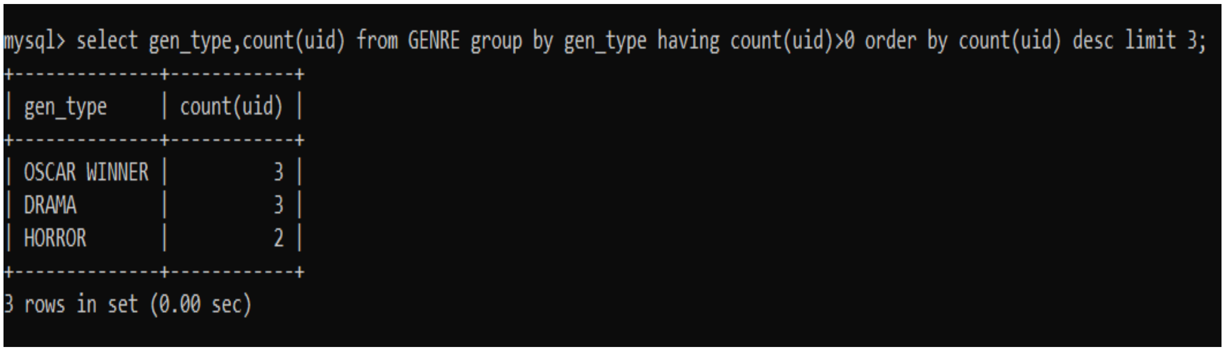
DIAGRAM on the next page (PTO)



```
mysql> Select movie_name,gen_type from MOVIE inner join GENRE on Movie.movie_id=GENRE.movie.id;
ERROR 1054 (42S22): Unknown column 'genre.movie.id' in 'on clause'
mysql> Select movie_name,gen_type from MOVIE inner join GENRE on Movie.movie_id=GENRE.movie_id;
+----------------------+---------------+
| movie_name           | gen_type      |
+----------------------+---------------+
| INSIDIOUS:1          | HORROR        |
| INSIDIOUS:DARKNESS   | HORROR        |
| DILWALE              | ROMANTIC      |
| DDLJ                 | ROMANTIC      |
| 3 IDIOTS             | COMEDY        |
| BAHUBALI             | ACTION        |
| BAHUBALI:2           | ACTION        |
| PATHER PACHALI       | OSCAR WINNER  |
| APUR SANSAR          | OSCAR WINNER  |
| GODFATHER:1          | OSACR NOMINEE |
| MY NAME IS KHAN      | DRAMA         |
| BAGHBAN              | DRAMA         |
| GODFATHER:2          | OSCAR WINNER  |
| BOL BACCHAN          | COMEDY        |
| INCEPTION            | SCI-FI        |
| SPACE ODEYESSY       | SCI-FI        |
| SAVING PVT RYAN      | WAR MOVIE     |
| GANGS OF WASSEYPUR:2 | THRILLER      |
| GANGS OF WASSEYPUR   | THRILLER      |
| TAXI DRIVER          | DRAMA         |
+----------------------+---------------+
20 rows in set (0.00 sec)
```

USING WHERE CLAUSE TO FIND TOP 3 GENRES VIEWED BY USERS:

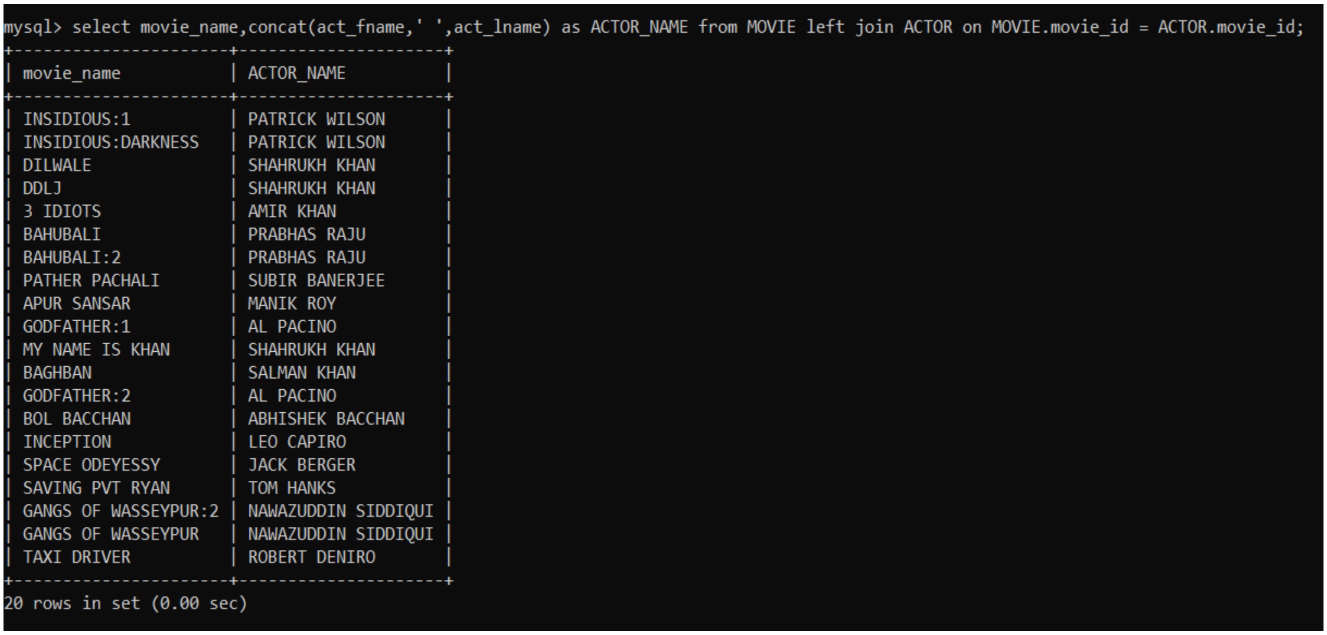
select gen_type,count(uid) from GENRE group by gen_type having count(uid)>0 order by count(uid) desc limit 3;



```
mysql> select gen_type,count(uid) from GENRE group by gen_type having count(uid)>0 order by count(uid) desc limit 3;
+--------------+------------+
| gen_type     | count(uid) |
+--------------+------------+
| OSCAR WINNER |          3 |
| DRAMA        |          3 |
| HORROR       |          2 |
+--------------+------------+
3 rows in set (0.00 sec)
```

USING LEFT JOIN TO VIEW LIST OF MOVIES ALONG WITH THE ACTORS:

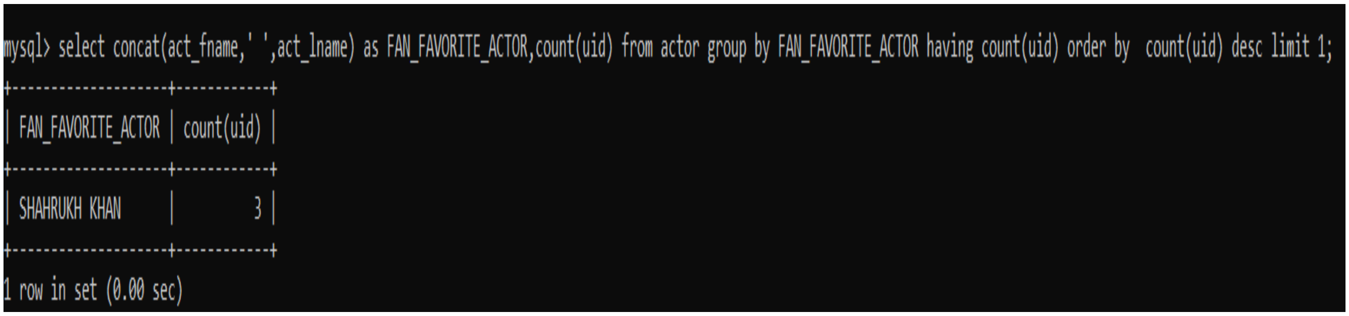
select movie_name,concat(act_fname,' ',act_lname) as ACTOR_NAME from MOVIE left join ACTOR on MOVIE.movie_id = ACTOR.movie_id;



```
mysql> select movie_name,concat(act_fname,' ',act_lname) as ACTOR_NAME from MOVIE left join ACTOR on MOVIE.movie_id = ACTOR.movie_id;
+----------------------+---------------------+
| movie_name           | ACTOR_NAME          |
+----------------------+---------------------+
| INSIDIOUS:1          | PATRICK WILSON      |
| INSIDIOUS:DARKNESS   | PATRICK WILSON      |
| DILWALE              | SHAHRUKH KHAN       |
| DDLJ                 | SHAHRUKH KHAN       |
| 3 IDIOTS             | AMIR KHAN           |
| BAHUBALI             | PRABHAS RAJU        |
| BAHUBALI:2           | PRABHAS RAJU        |
| PATHER PACHALI       | SUBIR BANERJEE      |
| APUR SANSAR          | MANIK ROY           |
| GODFATHER:1          | AL PACINO           |
| MY NAME IS KHAN      | SHAHRUKH KHAN       |
| BAGHBAN              | SALMAN KHAN         |
| GODFATHER:2          | AL PACINO           |
| BOL BACCHAN          | ABHISHEK BACCHAN    |
| INCEPTION            | LEO CAPIRO          |
| SPACE ODEYESSY       | JACK BERGER         |
| SAVING PVT RYAN      | TOM HANKS           |
| GANGS OF WASSEYPUR:2 | NAWAZUDDIN SIDDIQUI |
| GANGS OF WASSEYPUR   | NAWAZUDDIN SIDDIQUI |
| TAXI DRIVER          | ROBERT DENIRO       |
+----------------------+---------------------+
20 rows in set (0.00 sec)
```

TO DISPLAY THE MOST POPULAR ACTOR AMONG USERS:

select concat(act_fname,' ',act_lname) as FAN_FAVORITE_ACTOR,count(uid) from actor group by FAN_FAVORITE_ACTOR having count(uid) order by count(uid) desc limit 1;



```
mysql> select concat(act_fname,' ',act_lname) as FAN_FAVORITE_ACTOR,count(uid) from actor group by FAN_FAVORITE_ACTOR having count(uid) order by  count(uid) desc limit 1;
+--------------------+------------+
| FAN_FAVORITE_ACTOR | count(uid) |
+--------------------+------------+
| SHAHRUKH KHAN      |          3 |
+--------------------+------------+
1 row in set (0.00 sec)
```

T