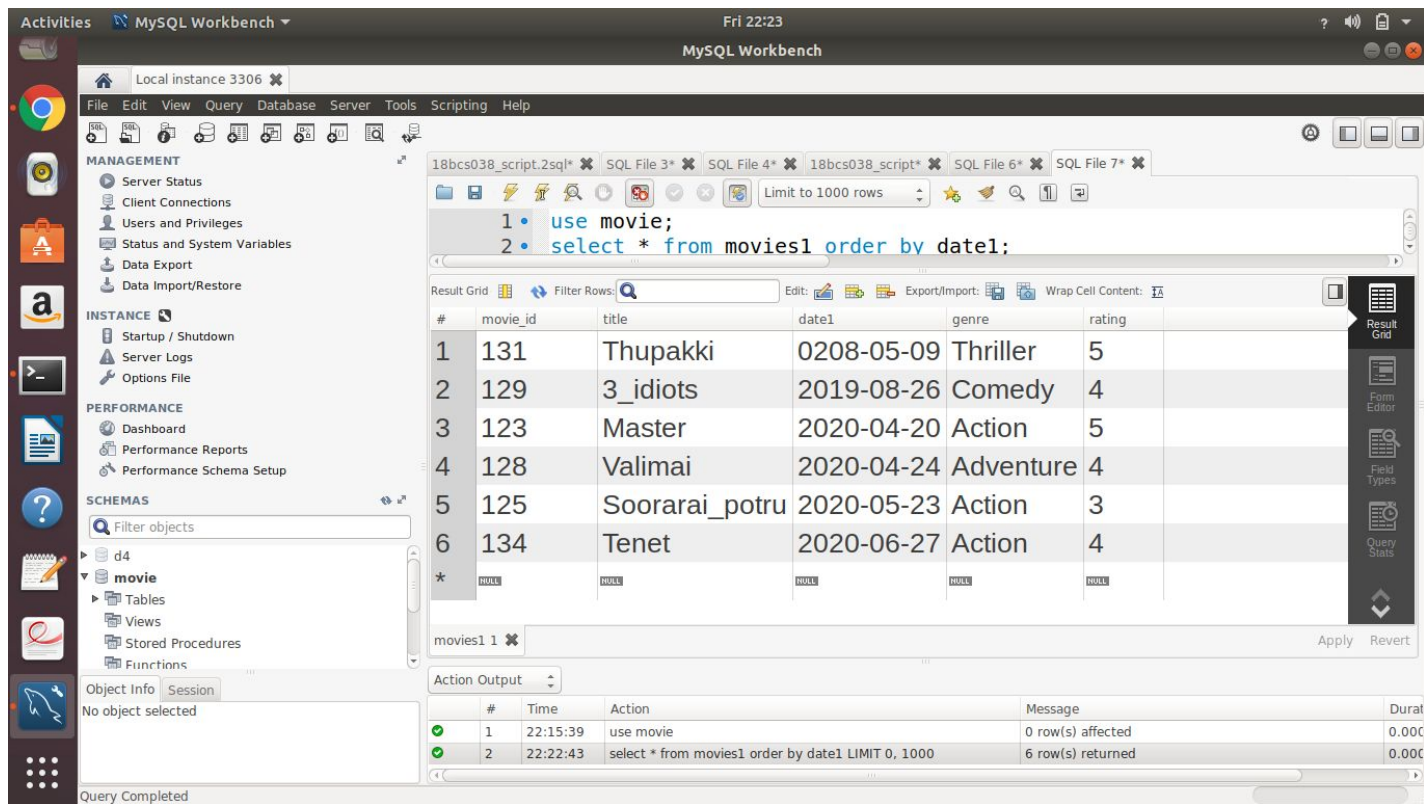


1. Order by clause.

Ans. select \* from movies1 order by date1;



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 • use movie;  
2 • select * from movies1 order by date1;
```

The query has been executed, and the results are displayed in the Result Grid. The results are ordered by the date1 column.

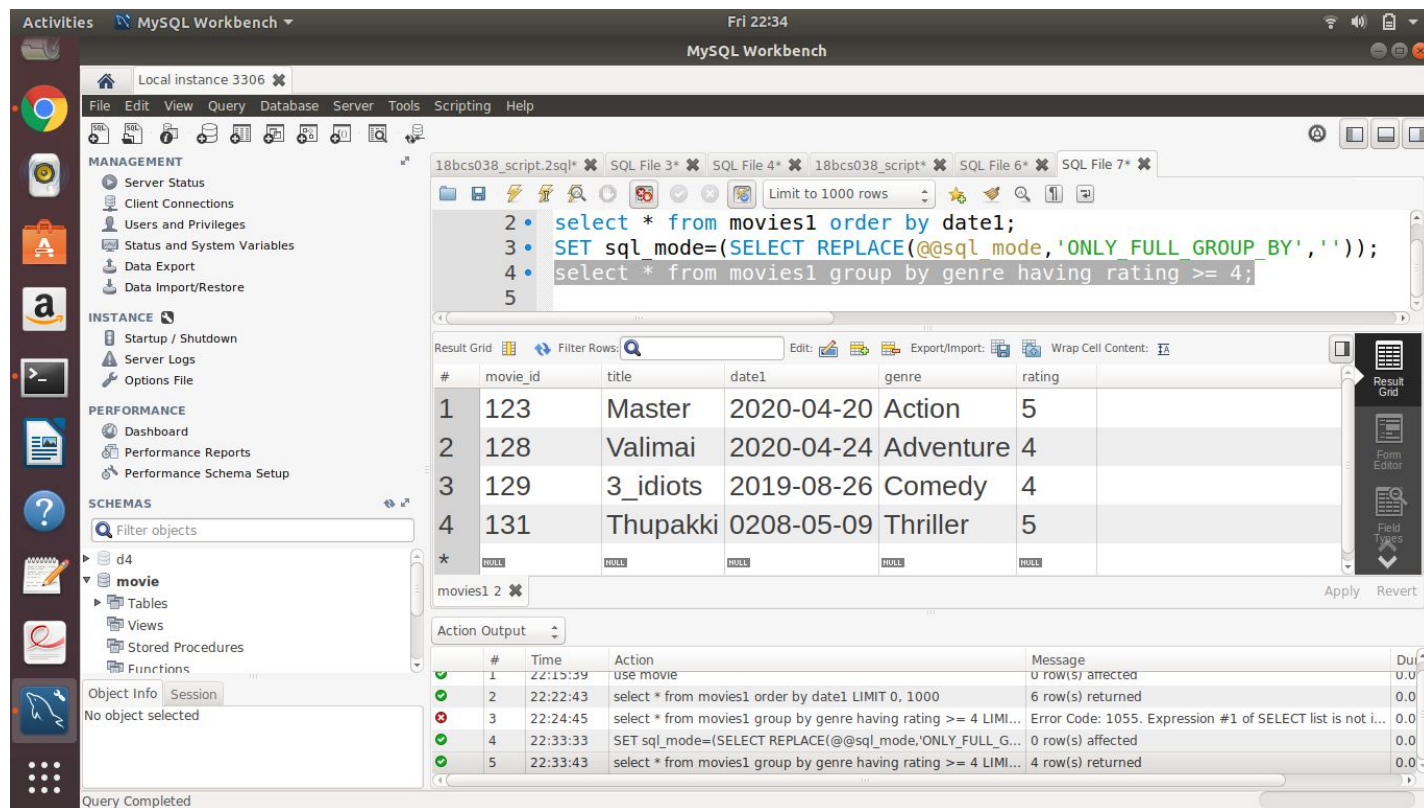
#	movie_id	title	date1	genre	rating
1	131	Thupakki	0208-05-09	Thriller	5
2	129	3 Idiots	2019-08-26	Comedy	4
3	123	Master	2020-04-20	Action	5
4	128	Valimai	2020-04-24	Adventure	4
5	125	Soorarai Potru	2020-05-23	Action	3
6	134	Tenet	2020-06-27	Action	4
*	NULL	NULL	NULL	NULL	NULL

The Action Output pane at the bottom shows the execution details:

#	Time	Action	Message	Duration
1	22:15:39	use movie	0 row(s) affected	0.000
2	22:22:43	select * from movies1 order by date1 LIMIT 0, 1000	6 row(s) returned	0.000

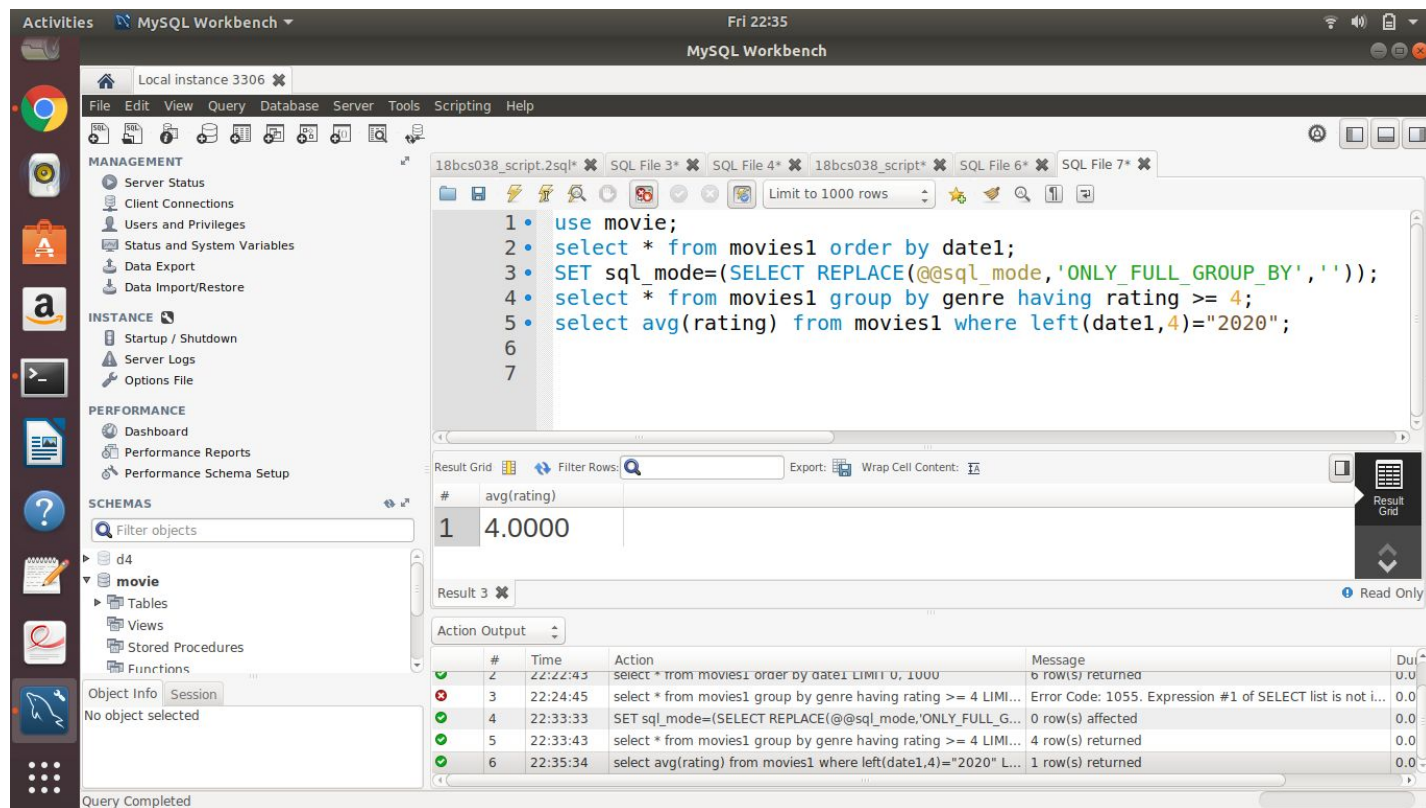
2. Group by and having.

Ans. select \* from movies1 group by genre having rating >= 4;



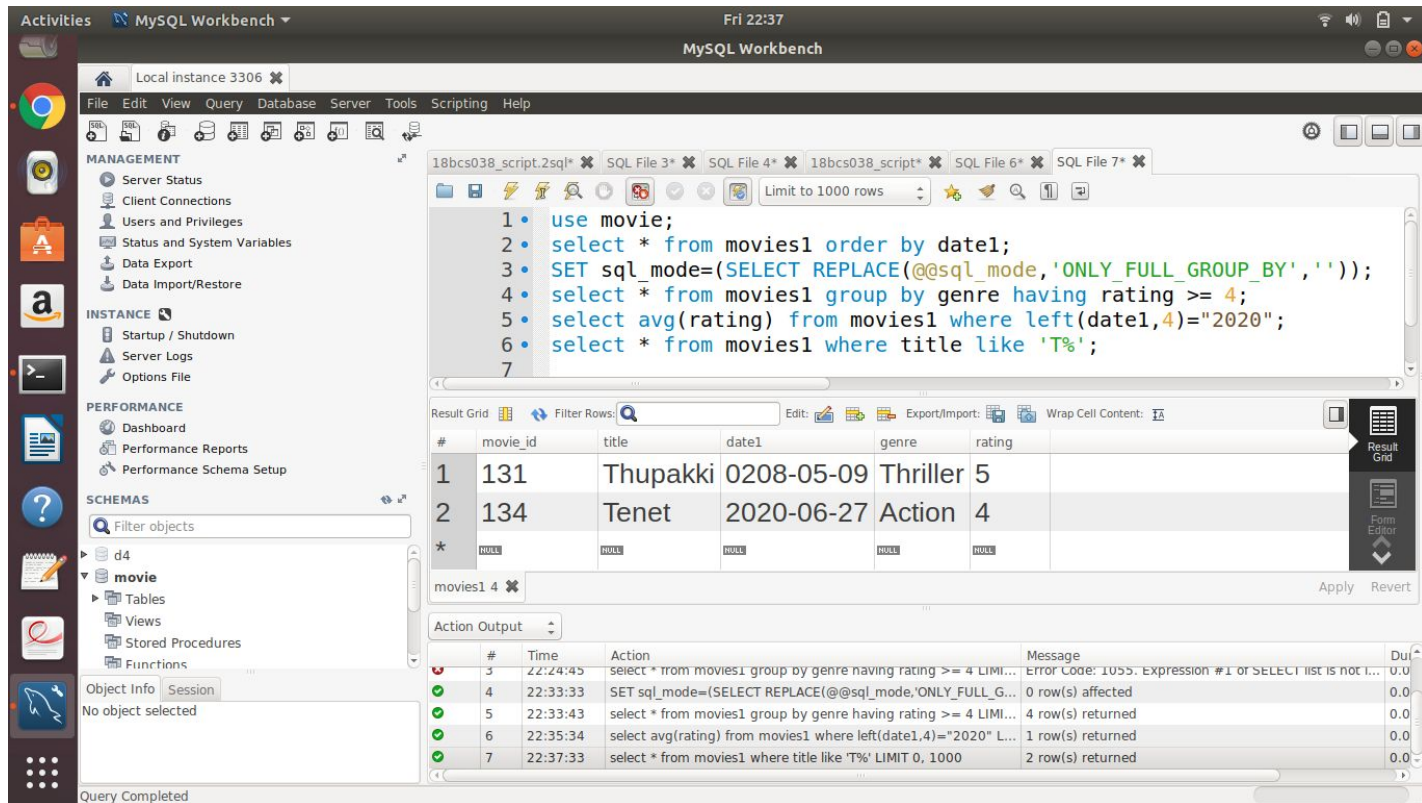
### 3. Aggregate functions.

Ans. select avg(rating) from movies1 where left(date1,4)="2020";



4. Logical operators especially with LIKE.

Ans. select \* from movies1 where title like 'T%';

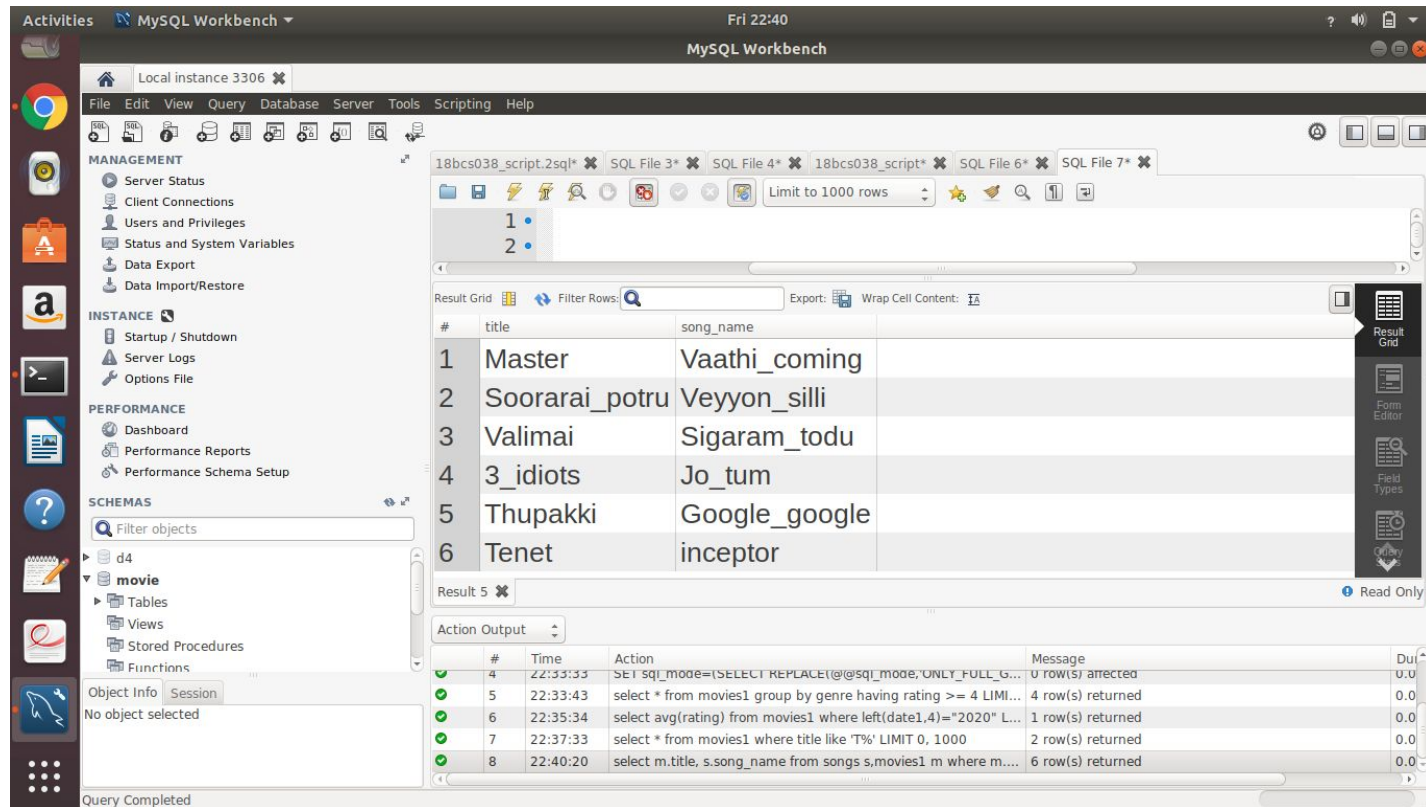


5. At least 4 nested queries specific to your Database, out of which at least 2 should have multiple subquery.

Ans.

A.

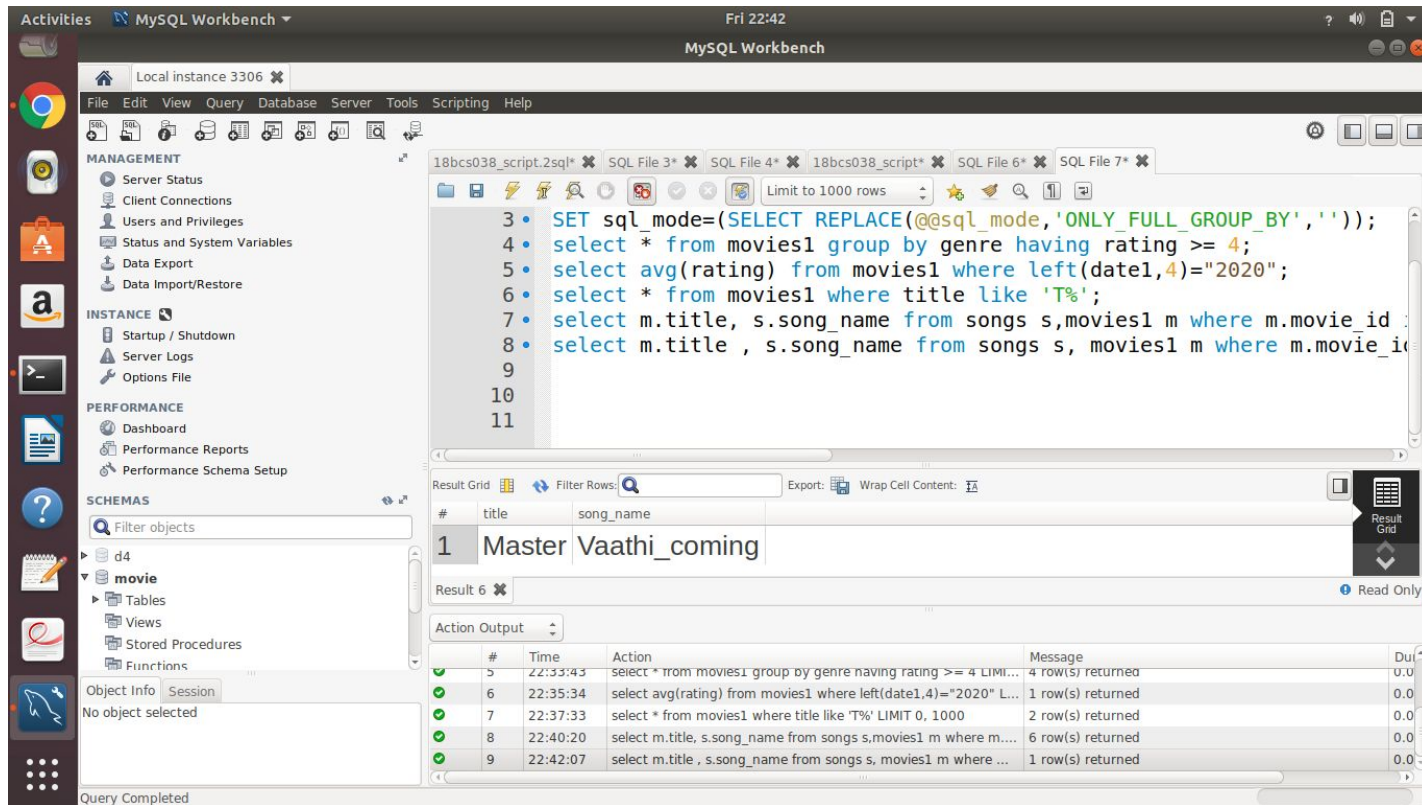
select m.title, s.song\_name from songs s, movies1 m  
where m.movie\_id in (select m.movie\_id where  
m.movie\_id=s.movie\_id);



B.

```
select m.title , s.song_name from songs s, movies1
m where m.movie_id=(select m.movie_id where
m.movie_id=s.movie_id and m.title="Master");
```





C.

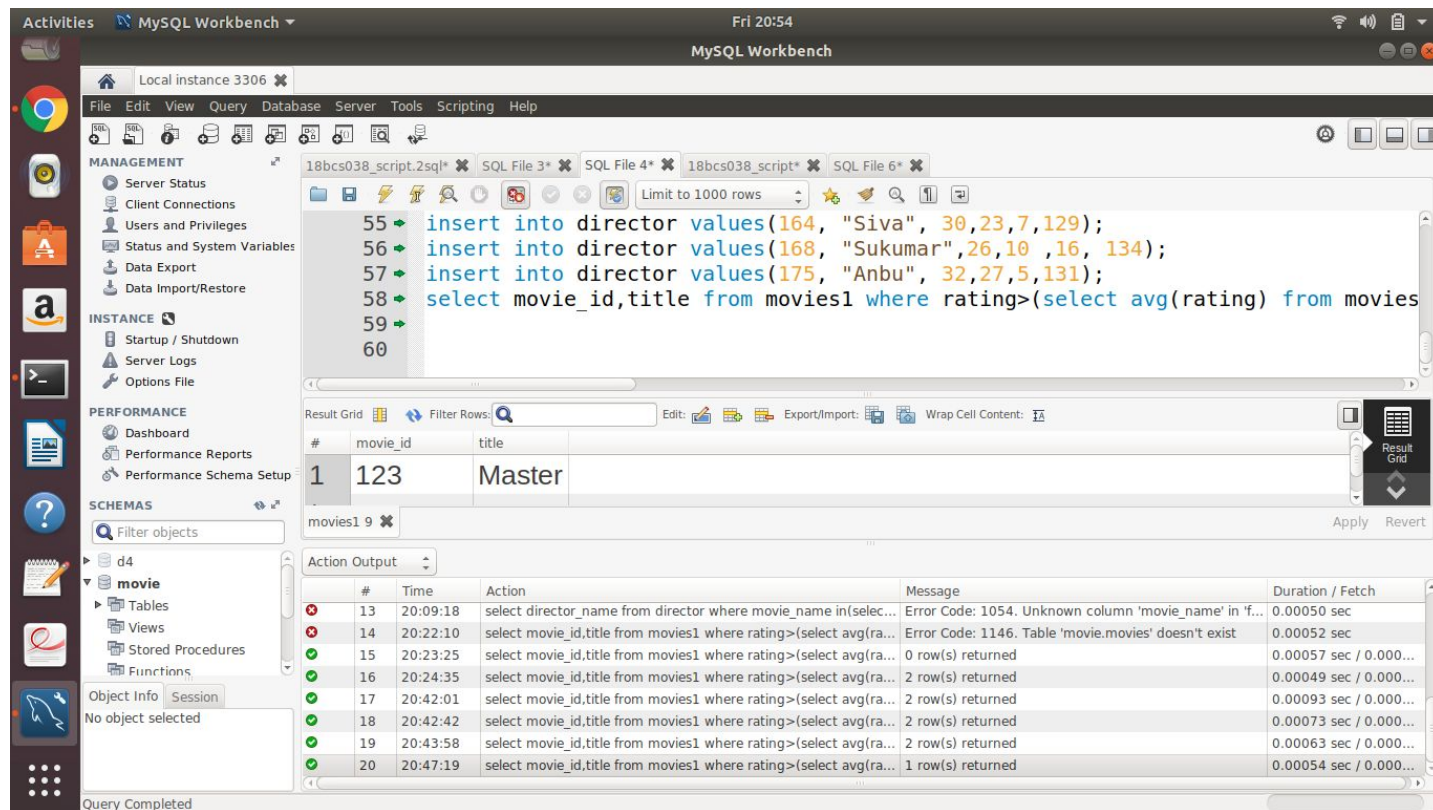
## Query:

select movie\_id,title from movies1 where  
rating>(select avg(rating) from movies1 where  
genre=(select genre from movies1 where  
movies1.movie\_id=134))and genre in(select genre  
from movies1 where movies1.movie\_id=134);

## Explanation:

/\*The inner query returns the average of the ratings of those movies with genre same as the genre of movies with movie\_id=134. The outer query displays the movie details with rating > the value returned by

the inner query and with genre same as that with  
movie\_id=134.\*/



D.

### Query:

```
select songs.song_name,songs.singer from songs
where songs.movie_id in(select movie_id from
director where no_of_flops >(select min(no_of_flops)
from director));
```

### Explanation:

/\*Displays the song name and singer from the songs  
of movies directed by directors with no\_of\_flops

greater than the minimum number of flops of any director from the director table\*/

The screenshot displays the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar contains a 'MANAGEMENT' section with options like Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore. Below this is an 'INSTANCE' section with Startup / Shutdown, Server Logs, and Options File. The 'PERFORMANCE' section includes Dashboard, Performance Reports, and Performance Schema Setup. The 'SCHEMAS' section has a search bar and a tree view showing 'd4' and 'movie' databases, with sub-items for Tables, Views, Stored Procedures, and Functions. The main editor window shows a SQL script with five insert statements into the 'director' table. The 'Result Grid' below the script displays a table with columns '#', 'song\_name', and 'singer'. The 'Action Output' pane at the bottom shows two error messages: 'Error Code: 1054. Unknown column 'movie\_name' in 'f...' and 'Error Code: 1146. Table 'movie.movies' doesn't exist'.

```
51 • insert into director values(135, "Gautham", 25, 18, 7, 125);
52 • use movie;
53 • insert into director values(143, "Shankar", 20, 17,3, 128);
54 • insert into director values(152, "Rajamouli", 45, 40,5, 123);
55 • insert into director values(164, "Siva", 30,23,7,129);
56 • insert into director values(168, "Sukumar",26,10,16, 134);
```

#	song_name	singer
1	Veyyon_silli	Harish
2	Vaathi_coming	Anirudh
3	Jo_tum	Taruk
4	inceptor	Jackie
5	Google_google	Vijay

#	Time	Action	Message	Du
13	20:09:18	select director_name from director where movie_name in(select...	Error Code: 1054. Unknown column 'movie_name' in 'f...	0.0
14	20:22:10	select movie_id,title from movies1 where rating>(select avg(ra...	Error Code: 1146. Table 'movie.movies' doesn't exist	0.0