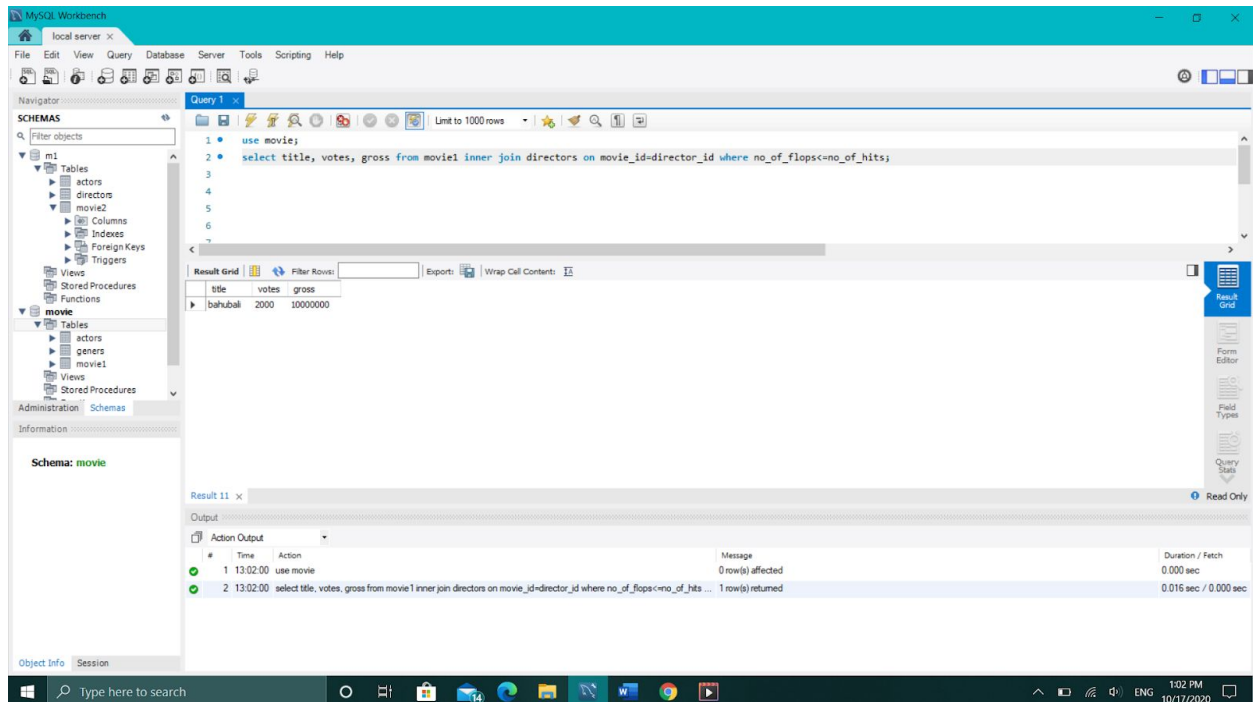


1.INNER JOIN

a) SQL QUERY

Select title, votes,gross from movie1 inner join directors on movie_id=director_id where no_of_flops<=no_of_hits;

OUTPUT:



The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1. use movie1;
2. select title, votes, gross from movie1 inner join directors on movie_id=director_id where no_of_flops<=no_of_hits;
```

The 'Result Grid' shows the following data:

title	votes	gross
bahubali	2000	10000000

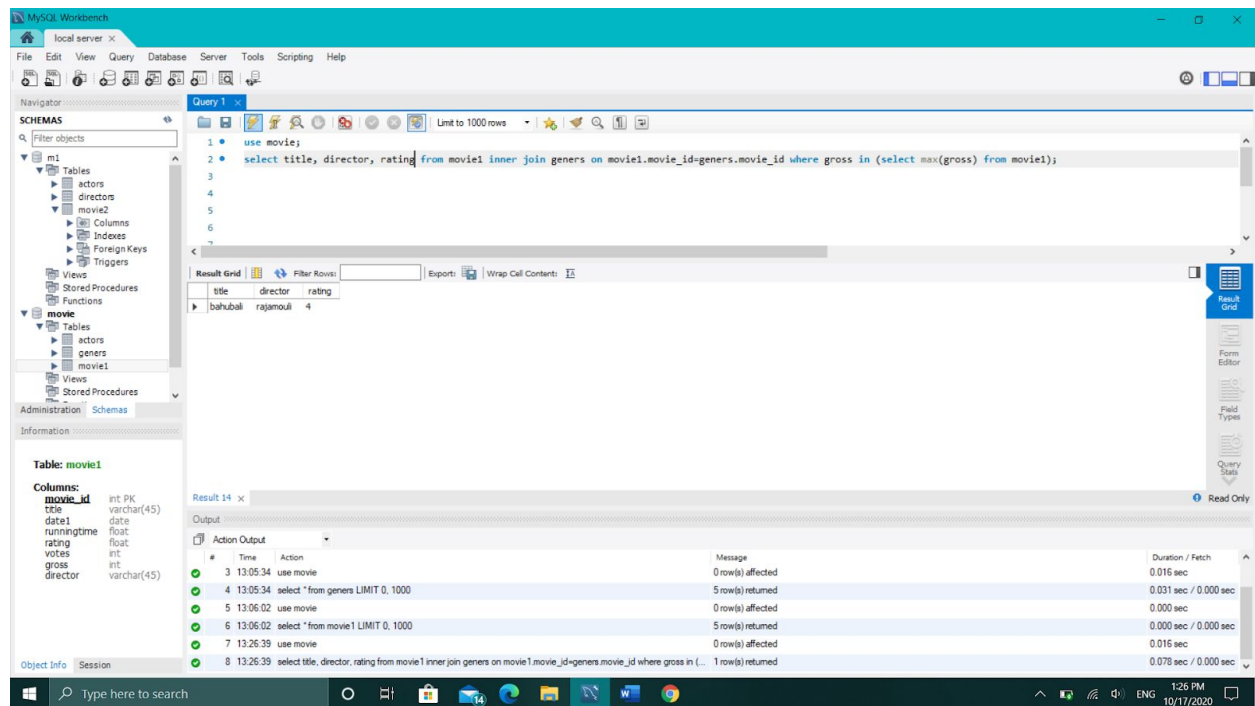
The 'Output' tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	13:02:00	use movie1	0 row(s) affected	0.000 sec
2	13:02:00	select title, votes, gross from movie1 inner join directors on movie_id=director_id where no_of_flops<=no_of_hits ...	1 row(s) returned	0.016 sec / 0.000 sec

b)SQL QUERY

Select title, director, rating from move1 inner join genres on movie1.movie_id where gross in (select max(gross) from movie1);

OUTPUT:



C) SQL QUERY

Select title,field_name, actor_name, director_name, date1,running time, rating from (((movie1 inner join actors on movie1.movie_id=actors.actor_id and actors.actor_name like 'm%' inner join directors on movie1.movie_id=directors_director_id) inner join genres on movie1.movie_id=genres.movie_id);

OUTPUT:

The screenshot displays the MySQL Workbench interface. The 'Query' tab is active, showing a SQL query that selects movie details from a database. The query is as follows:

```
1 use movie;
2 select title, field_name, actor_name, director_name, date1, runtime, rating from (((movie1 inner join actors on movie1.movie_id=actors.actor_id and actors.actor_name like 'pokri
3 inner join directors on movie1.movie_id=directors.director_id) inner join genres on movie1.movie_id=genres.movie_id);
4
5
```

The 'Result Grid' shows the output of the query, which is a single row of data:

title	field_name	actor_name	director_name	date1	runtime	rating
pokri	action/thriller	mareshbabu	puri	2010-05-20	2:45	2.5

The 'Table: movie1' section on the left lists the columns and their data types:

Columns:	movie_id	title	date1	runtime	rating	votes	gross	director
	int PK	varchar(45)	date	float	float	int	int	varchar(45)

The 'Output' section at the bottom shows the execution progress:

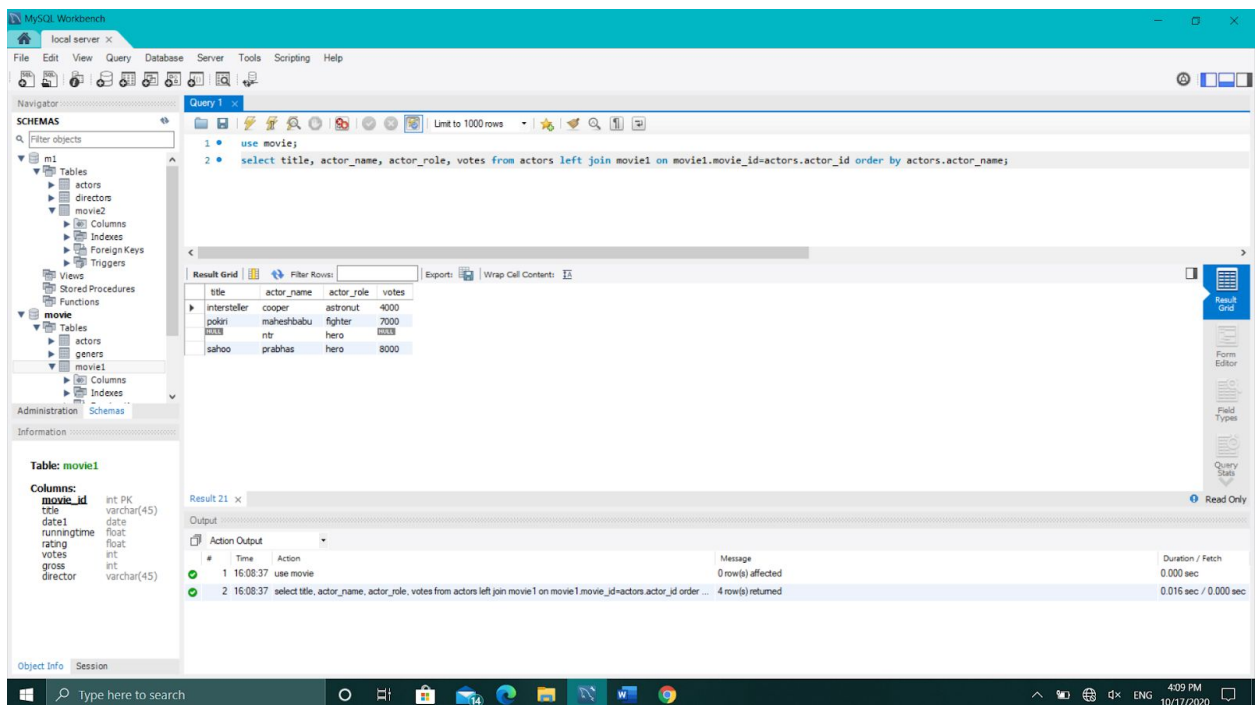
#	Time	Action	Message	Duration / Fetch
1	14:29:13	use movie	0 row(s) affected	0.000 sec
2	14:29:13	select title, field_name, actor_name, director_name, date1, runtime, rating from (((movie1 inner join actors on ...	1 row(s) returned	0.000 sec / 0.000 sec

2) LEFT OUTER JOIN

a) SQL QUERY

Select title, actor_name, actor_role, votes from actors left join movie1 on movie1.movie_id=actors.actor_id order by actors.actor_name;

OUTPUT:



The screenshot displays the MySQL Workbench interface. The 'Query' tab is active, showing the following SQL query:

```
1 • use movie1;
2 • select title, actor_name, actor_role, votes from actors left join movie1 on movie1.movie_id=actors.actor_id order by actors.actor_name;
```

The 'Result Grid' shows the output of the query, which is a table with 4 columns: title, actor_name, actor_role, and votes. The data is as follows:

title	actor_name	actor_role	votes
intersteller	cooper	astronut	4000
pokiri	maheishbabu	fighter	7000
...
sahoo	prabhas	hero	8000

The 'Table: movie1' structure is also visible in the left sidebar:

Columns:	movie_id	title	date1	runningtime	rating	votes	gross	director
	int PK	varchar(45)	date	float	float	int	int	varchar(45)

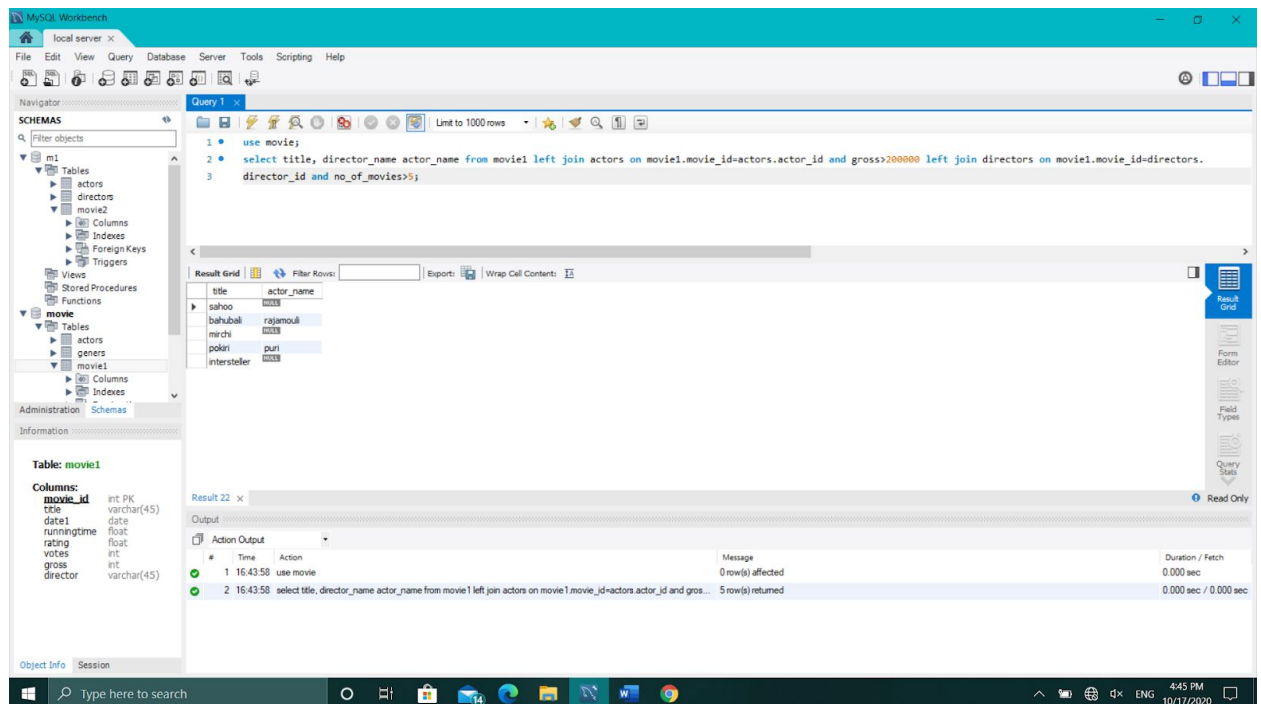
The 'Action Output' pane at the bottom shows the execution progress:

#	Time	Action	Message	Duration / Fetch
1	15:08:37	use movie1	0 row(s) affected	0.000 sec
2	15:08:37	select title, actor_name, actor_role, votes from actors left join movie1 on movie1.movie_id=actors.actor_id order by actors.actor_name;	4 row(s) returned	0.016 sec / 0.000 sec

B) SQL QUERY

Select title, director_name, actor_name from movie1 left join actors on movie1.movie_id=actors.actor_id and gross>200000 left join directors on movie1.movie_id=directors.director_id and no_of_movies>5;

OUTPUT:



The screenshot shows the MySQL Workbench interface. The 'Query 1' tab contains the following SQL query:

```
1. use movie;
2. select title, director_name actor_name from movie1 left join actors on movie1.movie_id=actors.actor_id and gross>200000 left join directors on movie1.movie_id=directors.
3. director_id and no_of_movies>5;
```

The 'Result Grid' shows the output of the query, displaying a table with columns 'title' and 'actor_name'. The results are as follows:

title	actor_name
sahoo	XXXX
bahubali	rajamouli
merchi	XXXX
pokiri	puri
intersteller	XXXX

The 'Action Output' tab shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
1	16:43:58	use movie	0 row(s) affected	0.000 sec
2	16:43:58	select title, director_name actor_name from movie1 left join actors on movie1.movie_id=actors.actor_id and gross>200000 left join directors on movie1.movie_id=directors. director_id and no_of_movies>5;	5 row(s) returned	0.000 sec / 0.000 sec

C) SQL QUERY

Select title, actor_name, director_name, field_name from movie1 left outer join actors on movie1.movie_id=actors.actor_id left outer join directors on movie1.movie_id=directors.director_id left join genres on movie1.movie_id=genres.movie_id;

OUTPUT;

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1. use movie1;
2. select title, actor_name, director_name, field_name from movie1 left join actors on movie1.movie_id=actors.actor_id left join directors on movie1.movie_id=directors.director_id
3. left join genres on movie1.movie_id=genres.movie_id;
```

The 'Result Grid' shows the output of the query, which includes the following data:

title	actor_name	director_name	field_name
sahoo	grabhas	adventure	
bahubali	rajamouli	action/drama	
merchi	maaheshbabu	action/thriller	
polni	cooper	sc-fi/adventure	

The 'Output' tab shows the execution details of the query:

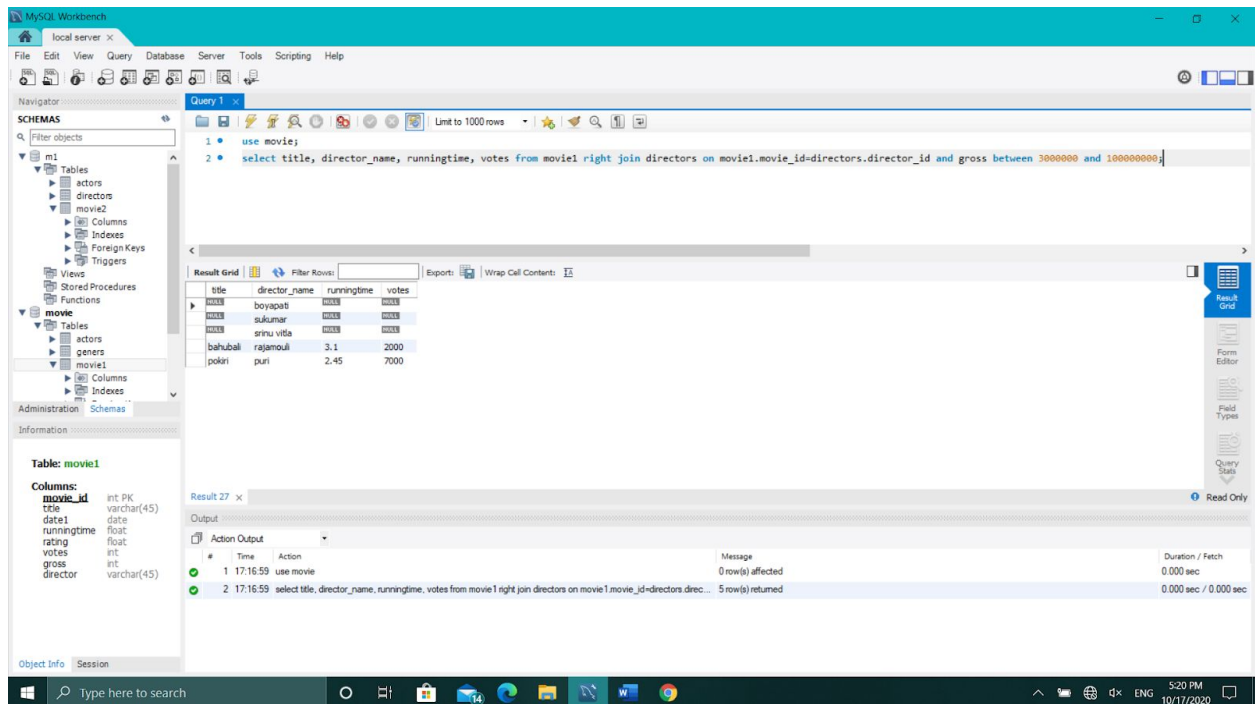
#	Time	Action	Message	Duration / Fetch
1	17:03:00	use movie1	0 row(s) affected	0.000 sec
2	17:03:00	select title, actor_name, director_name, field_name from movie1 left join actors on movie1.movie_id=actors.actor_id left join directors on movie1.movie_id=directors.director_id left join genres on movie1.movie_id=genres.movie_id;	5 row(s) returned	0.000 sec / 0.000 sec

3) RIGHT OUTER JOIN

a) SQL QUERY

Select title, director_name, running time, votes from movie1 right outer join directors on movie1.movie_id=directors_director_id and gross between 3000000 and 100000000;

OUTPUT:



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

```
1. use movie1;
2. select title, director_name, runningtime, votes from movie1 right join directors on movie1.movie_id=directors_director_id and gross between 3000000 and 100000000;
```

The results are displayed in a table with the following columns: title, director_name, runningtime, and votes. The data rows are:

title	director_name	runningtime	votes
boopas
sukumar
sinu vitta
bahubali	rajamouli	3.1	2000
pokiri	puri	2.45	7000

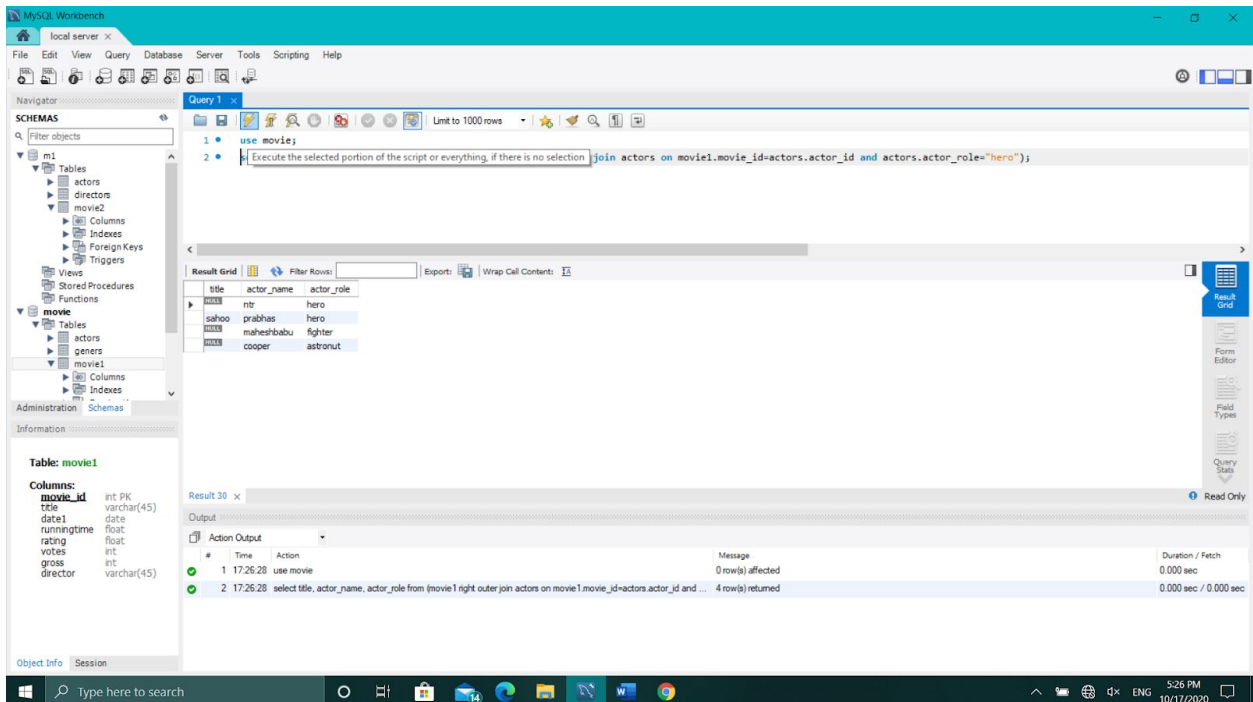
The bottom panel shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	17:16:59	use movie1	0 row(s) affected	0.000 sec
2	17:16:59	select title, director_name, runningtime, votes from movie1 right join directors on movie1.movie_id=directors_director_id and gross between 3000000 and 100000000;	5 row(s) returned	0.000 sec / 0.000 sec

B) SQL QUERY

Select title, actor_name, actor_role from (movie1 right outer join actors on movie1.movie_id=actors.actors_id and actors.actor_role="hero");

OUTPUT:



The screenshot shows the MySQL Workbench interface. The 'Query 1' tab contains the following SQL query:

```
1 use movie;
2 join actors on movie1.movie_id=actors.actors_id and actors.actor_role="hero";
```

The 'Result Grid' shows the output of the query:

	title	actor_name	actor_role
1	ntr	hero	
2	sahoo	prabhas	hero
3	mahe	mahe	fighter
4	cooper	astronut	

The 'Table: movie1' section shows the columns and their data types:

Columns:	movie_id	title	date1	runningtime	rating	votes	gross	director
	int PK	varchar(45)	date	float	float	int	int	varchar(45)

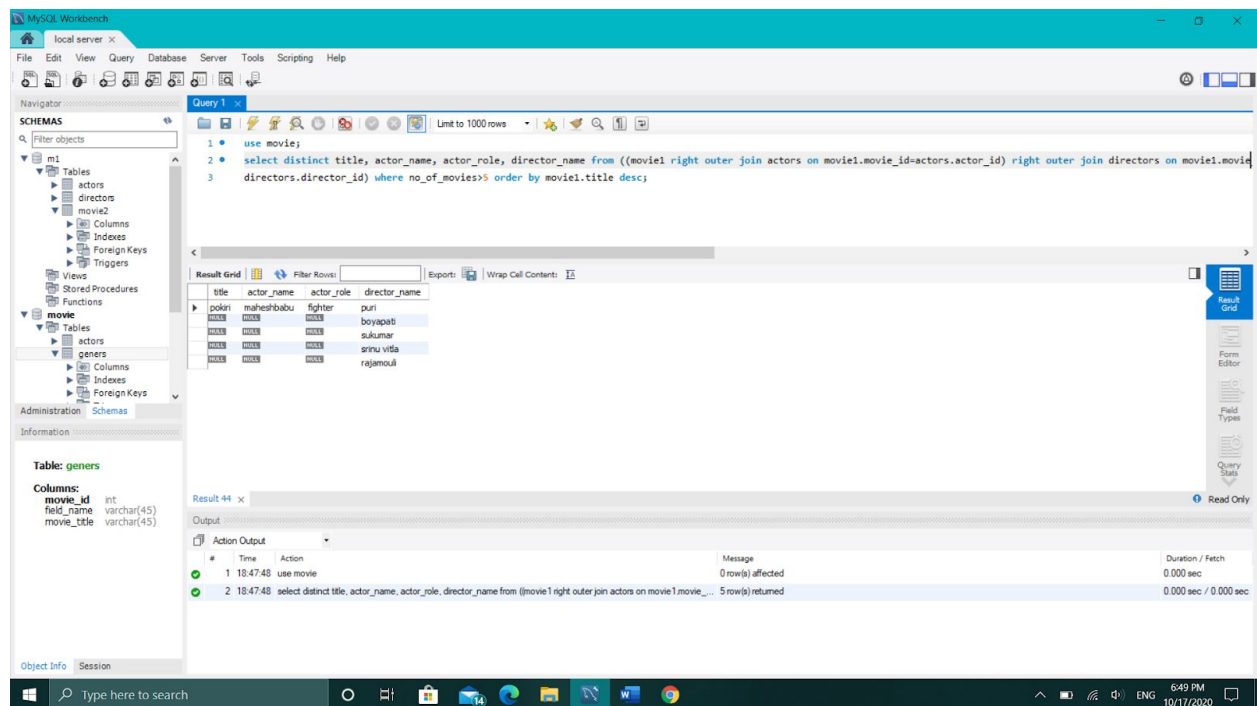
The 'Output' section shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	17.26.28	use movie	0 row(s) affected	0.000 sec
2	17.26.28	select title, actor_name, actor_role from (movie1 right outer join actors on movie1.movie_id=actors.actors_id and actors.actor_role="hero");	4 row(s) returned	0.000 sec / 0.000 sec

C) SQL QUERY

Select distinct title, actor_name, actor_role, director_name from ((movie1 right outer join actors on movie1.movie_id=actors.actor_id) right outer join directors on movie1.movie_id= directors.directors_id) where no_of_movies>5 order by movie1.title desc;

OUTPUT:



The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'm1' expanded, containing tables 'actors', 'directors', 'movie1', and 'movie2'. The 'movie1' table is selected, showing its columns: 'movie_id' (int), 'title' (varchar(45)), and 'no_of_movies' (varchar(45)).

The main query editor shows the following SQL query:

```
1. use movie;
2. select distinct title, actor_name, actor_role, director_name from ((movie1 right outer join actors on movie1.movie_id=actors.actor_id) right outer join directors on movie1.movie_id= directors.directors_id) where no_of_movies>5 order by movie1.title desc;
```

The 'Result Grid' shows the output of the query, displaying 5 rows of data:

title	actor_name	actor_role	director_name
poker	naheshbabu	fighter	puri
...
...
...
...

The 'Action Output' pane at the bottom shows the execution progress:

#	Time	Action	Message	Duration / Fetch
1	18:47:48	use movie	0 row(s) affected	0.000 sec
2	18:47:48	select distinct title, actor_name, actor_role, director_name from ((movie1 right outer join actors on movie1.movie_id=actors.actor_id) right outer join directors on movie1.movie_id= directors.directors_id) where no_of_movies>5 order by movie1.title desc;	5 row(s) returned	0.000 sec / 0.000 sec