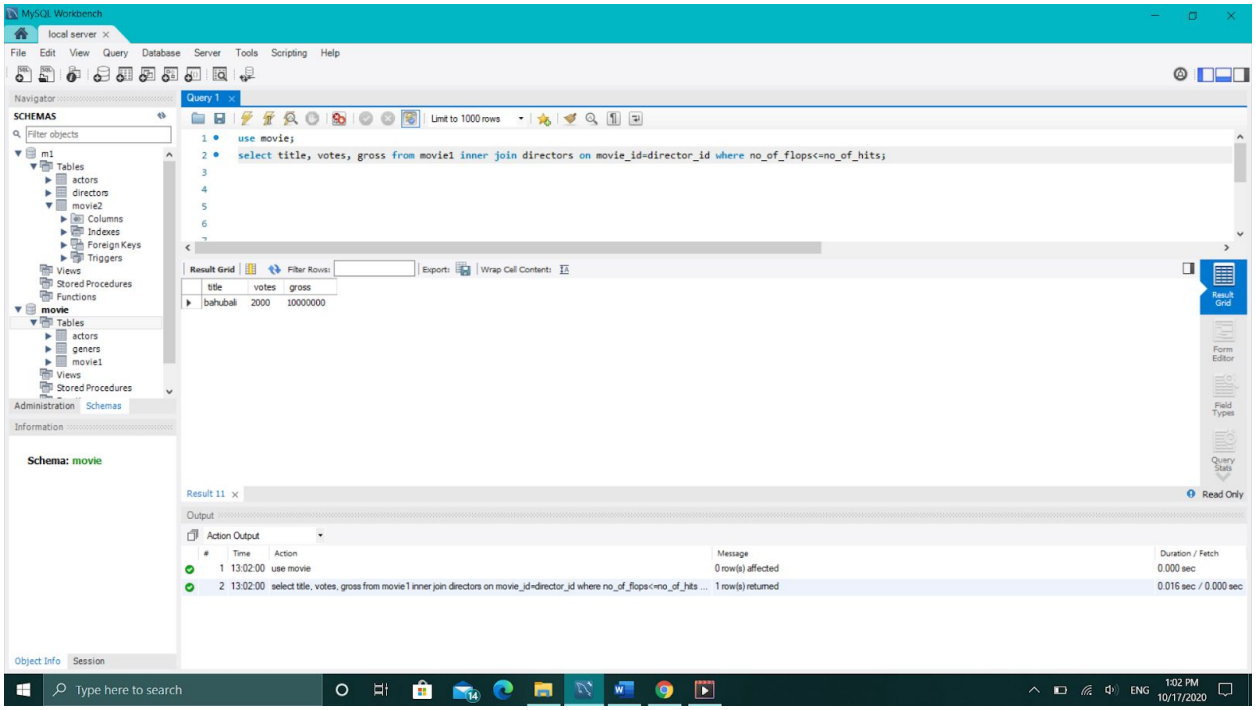
1.INNER JOIN

a) SQL QUERY

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

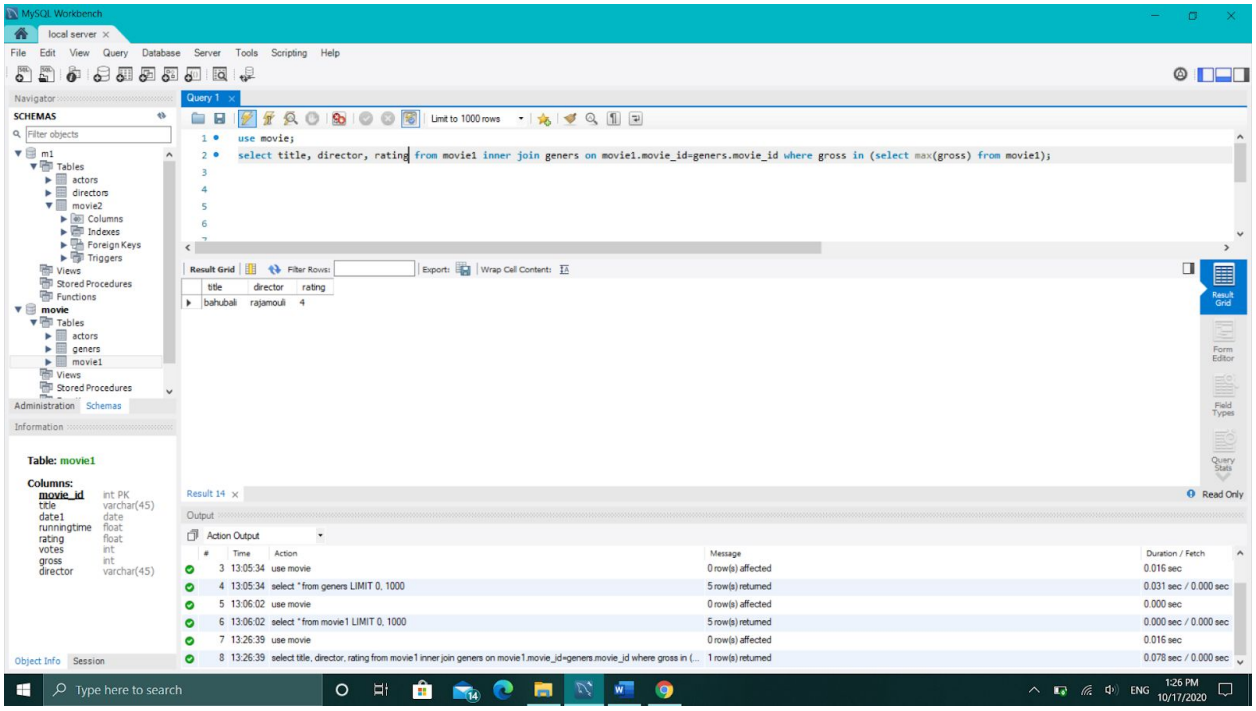
OUTPUT:



b)SQL QUERY

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

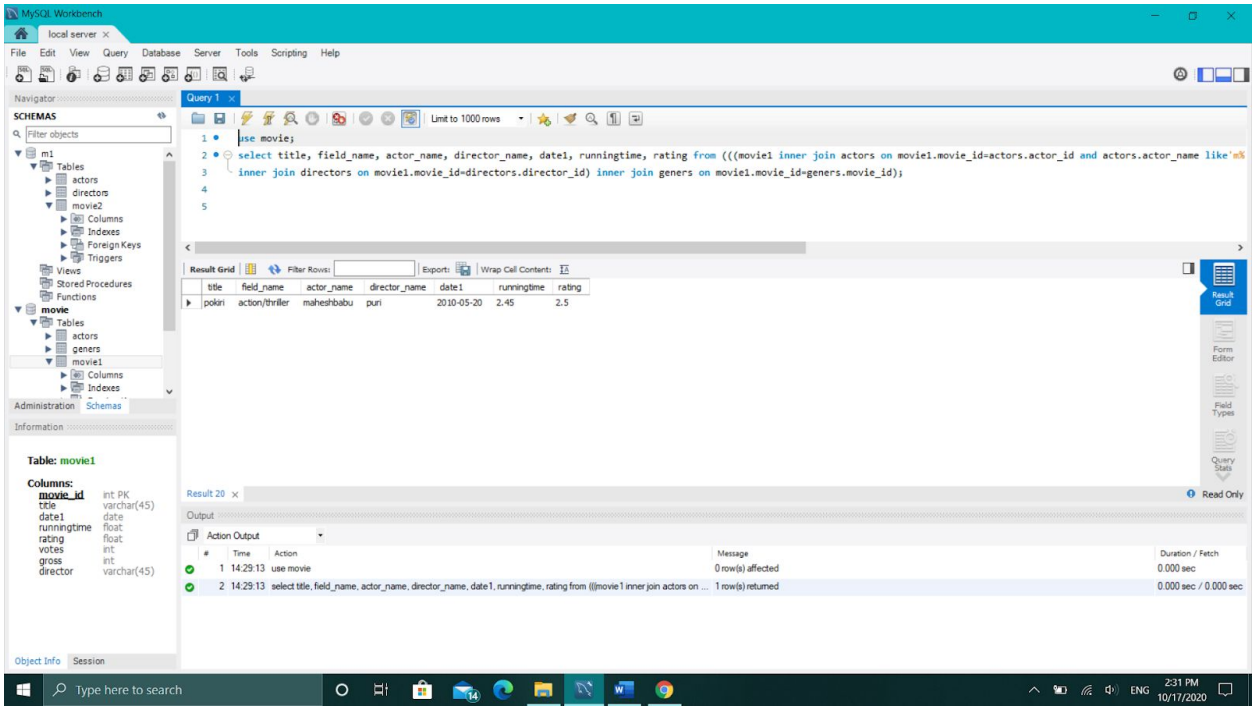
OUTPUT:



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=geners.movie_id);

OUTPUT:

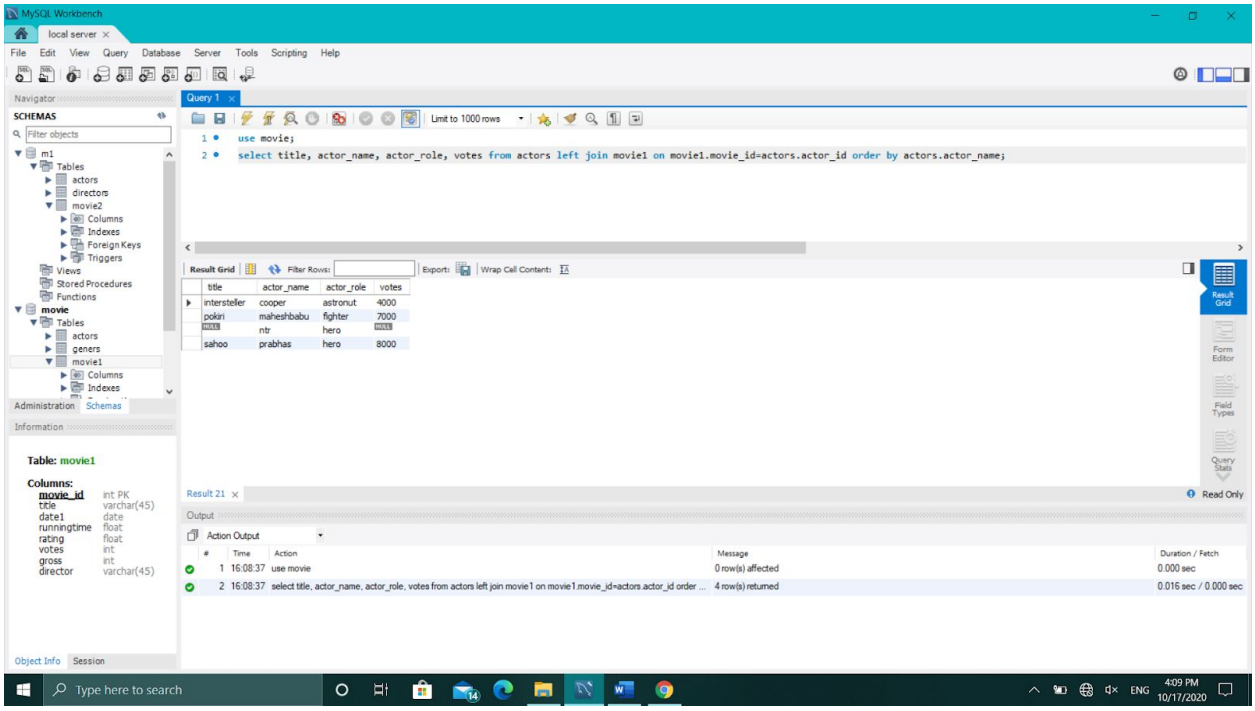

MySQL Workbench
local server
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
m1
Tables
actors
directors
movie2
Columns
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
movie
Tables
actors
geners
movie1
Columns
Indexes
Administration Schemas
Information
Table: movie1
Columns:
movie_id int PK
title varchar(45)
date1 date
runningtime float
rating float
votes int
gross int
director varchar(45)
Object Info Session
Query 1
Limit to 1000 rows
use movie;
select title, field_name, actor_name, director_name, date1, runningtime, rating from (((movie1 inner join actors on movie1.movie_id=actors.actor_id and actors.actor_name like'n%
inner join directors on movie1.movie_id=directors.director_id) inner join geners on movie1.movie_id=geners.movie_id);
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
title field_name actor_name director_name date1 runningtime rating
pokiri action/thriller maheshbabu puri 2010-05-20 2.45 2.5
Result Grid
Form Editor
Field Types
Query Stats
Result 20
Read Only
Output
Action Output
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, runningtime, rating from (((movie1 inner join actors on ... 1 row(s) returned 0.000 sec / 0.000 sec
Type here to search
ENG
2:31 PM
10/17/2020

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;

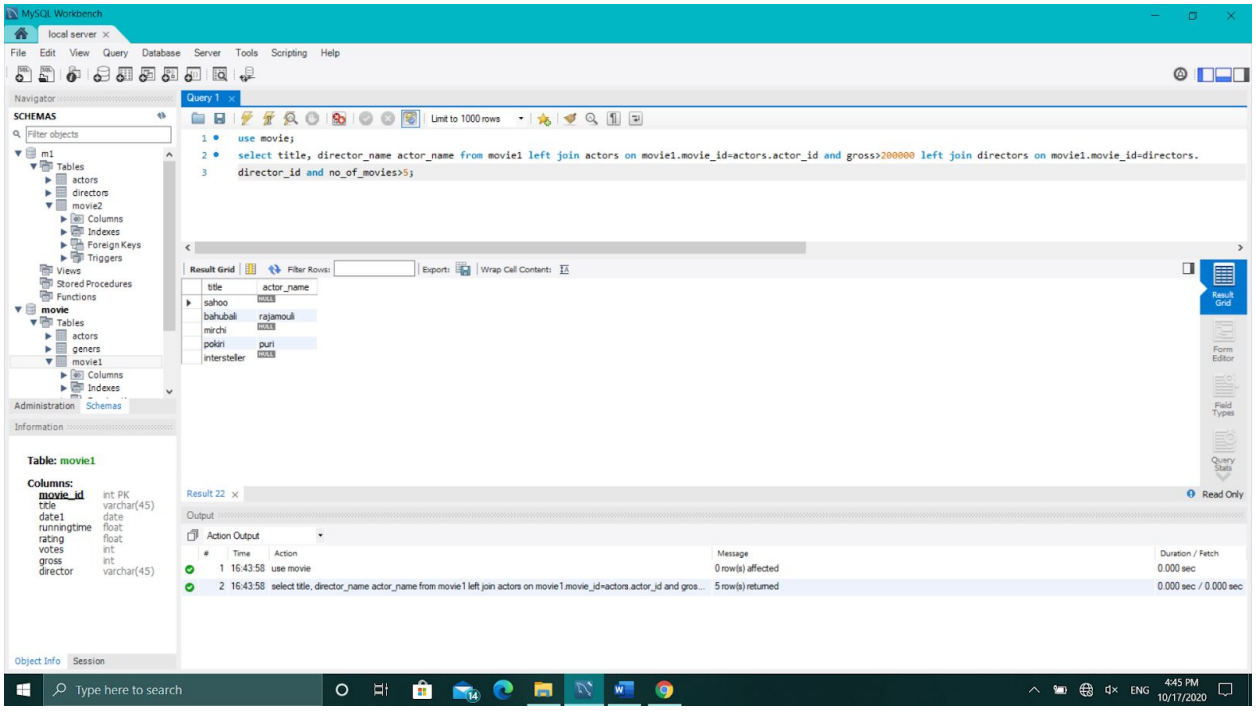
OUTPUT:



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;

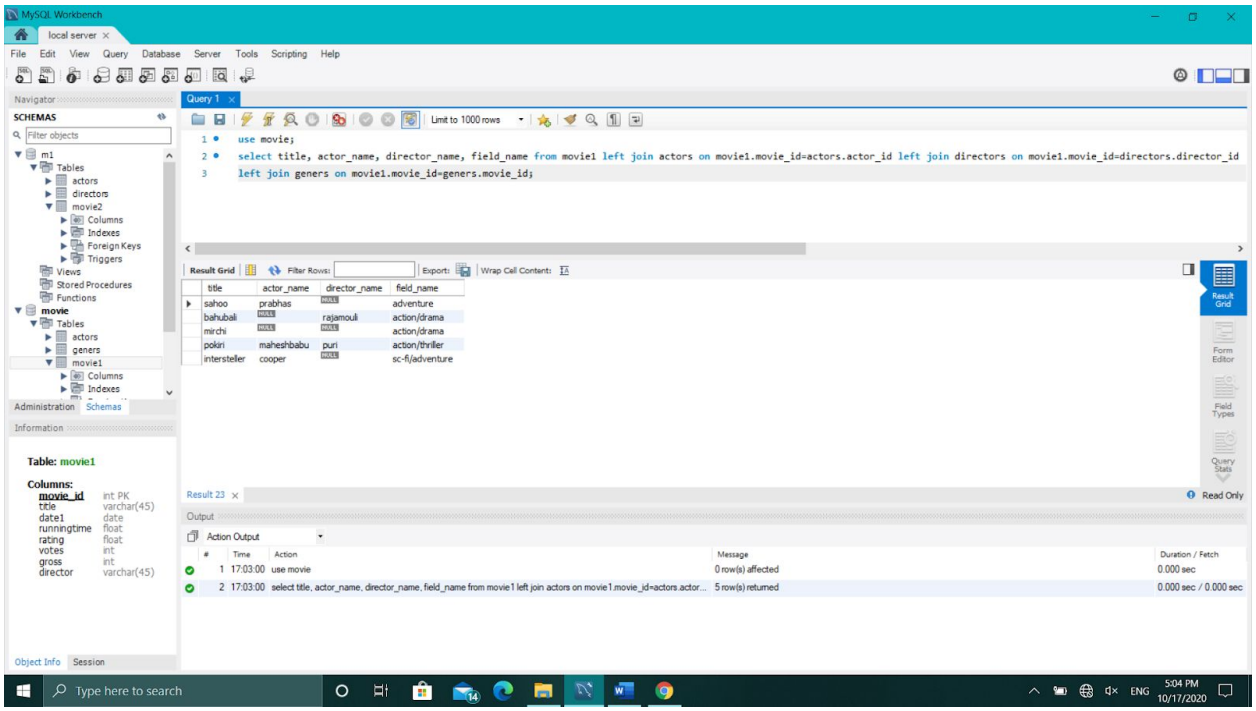
OUTPUT:



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=geners.movie_id;

OUTPUT;

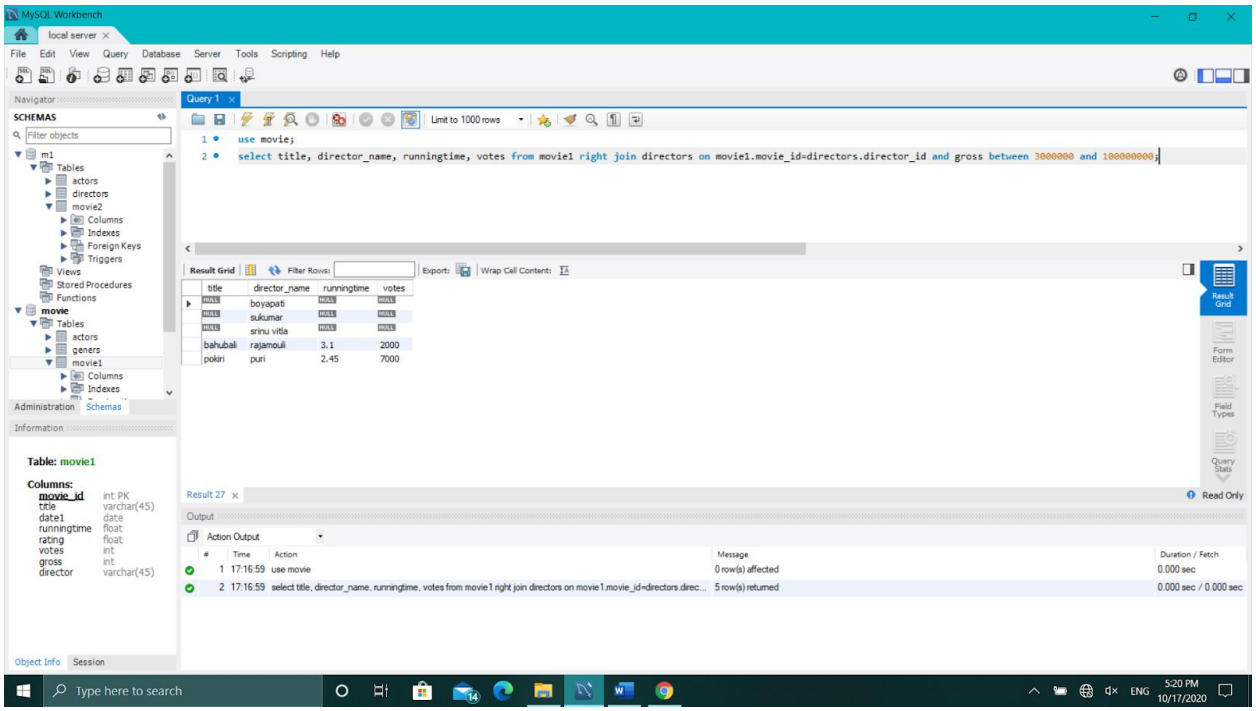


3) RIGHT OUTER JOIN

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;

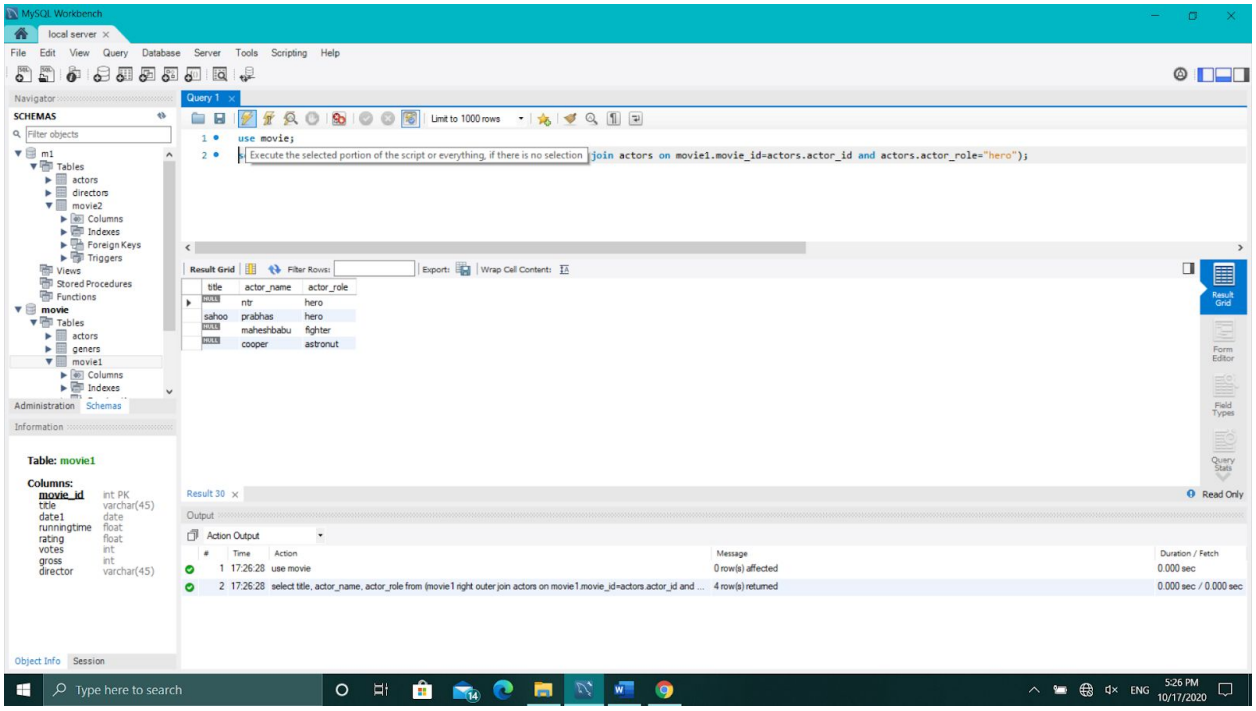
OUTPUT:



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:



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: