ASSIGNMENT-2

JOIN

Consider the schema for MovieDatabase:

ACTOR (**Act_id**, Act_Name, Act_Gender)

DIRECTOR (**Dir_id**, Dir_Name, Dir_Phone)

MOVIES (**Mov_id**, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST (**Act_id, Mov_id**, Role)

RATING **(Mov_id,** Rev_Stars)

Write SQL queries to

1. List the titles of all movies directed by 'Hitchcock'.

2. Find the movie names where one actor acted in two or more movies.

3. List all actors who acted in a movie before 2000 and also in a movies after 2015 (use JOIN operation).

4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

5. Update rating of all movies directed by 'Steven Spielberg' to 5.

SCRIPT

```
CREATE DATABASE MOVIEDATABASE;

USE MOVIEDATABASE;

create table ACTOR(act_id int(10),act_name varchar(10),act_gender
varchar(5),primary key(act_id));
```

```
create table DIRECTOR(dir_id int(10),dir_name varchar(10),dir_phone int(10),primary key(dir_id));

create table MOVIES(mov_id int(10),mov_title varchar(10),mov_year int(10),mov_lang varchar(10),dir_id int(10),primary key(mov_id),foreign key(dir_id) references DIRECTOR (dir_id) on delete cascade);

create table MOVIE_CAST(act_id int(10),mov_id int(10),role varchar(10),foreign key(act_id) references ACTOR (act_id) on delete cascade,foreign key(mov_id) references MOVIES (mov_id) on delete cascade);

create table RATING(mov_id int(10),rev_stars float(5),foreign key(mov_id) references MOVIES (mov_id) on delete cascade);


insert into ACTOR values(101,"Coen","M");

insert into ACTOR values(102,"Raimi","M");

insert into ACTOR values(103,"Hanson","M");

insert into ACTOR values(104,"Hanks","M");


insert into DIRECTOR values(111,"Steven Spielberg",2541245);

insert into DIRECTOR values(112,"Hitchcock",415574);

insert into DIRECTOR values(113,"Hitchcock",145236);

insert into DIRECTOR values(114,"Steven Spielberg",968746);


insert into MOVIES values(1,"Fargo",1996,"english",111);

insert into MOVIES values(2,"Wonder Boys",1998,"hindi",114);

insert into MOVIES values(3,"Raising Arizona",2002,"english",112);

insert into MOVIES values(4,"Spiderman",2018,"english",113);
```

```
insert into MOVIE_CAST values(101,1,"hero");

insert into MOVIE_CAST values(102,2,"hero");

insert into MOVIE_CAST values(104,3,"villain");

insert into MOVIE_CAST  values(104,4,"hero");


insert into RATING values(1,4);

insert into RATING values(2,3);

insert into RATING values(4,5);

insert into RATING values(3,2);
```

1) SELECT MOVIES.mov_title

```
FROM MOVIES

INNER JOIN DIRECTOR ON MOVIES.dir_id=DIRECTOR.dir_id  WHERE
dir_name="Hitchcock";
```

2)select movies.mov_title,actor.act_id from

```
actor join movie_cast on actor.act_id=movie_cast.act_id

join movies on  movie_cast.mov_id=movies.mov_id

where actor.act_id=( select actor.act_id from

actor join movie_cast on actor.act_id=movie_cast.act_id

join movies on  movie_cast.mov_id=movies.mov_id group by actor.act_id
having count(mov_title)>1)
```

3)SELECT act_name

FROM ACTOR A

JOIN MOVIE_CAST C

ON A.act_id=C.act_id

JOIN MOVIES M

ON C.mov_id=M.mov_id

WHERE M.mov_year NOT BETWEEN 2000 AND 2015;

4)SELECT MOV_TITLE,MAX(REV_STARS)

FROM MOVIES

INNER JOIN RATING USING (MOV_ID)

GROUP BY MOV_TITLE

HAVING MAX(REV_STARS)>0

ORDER BY MOV_TITLE;

5)SET SQL_SAFE_UPDATES=0;

UPDATE RATING

SET REV_STARS=5

WHERE MOV_ID IN (SELECT MOV_ID FROM MOVIES

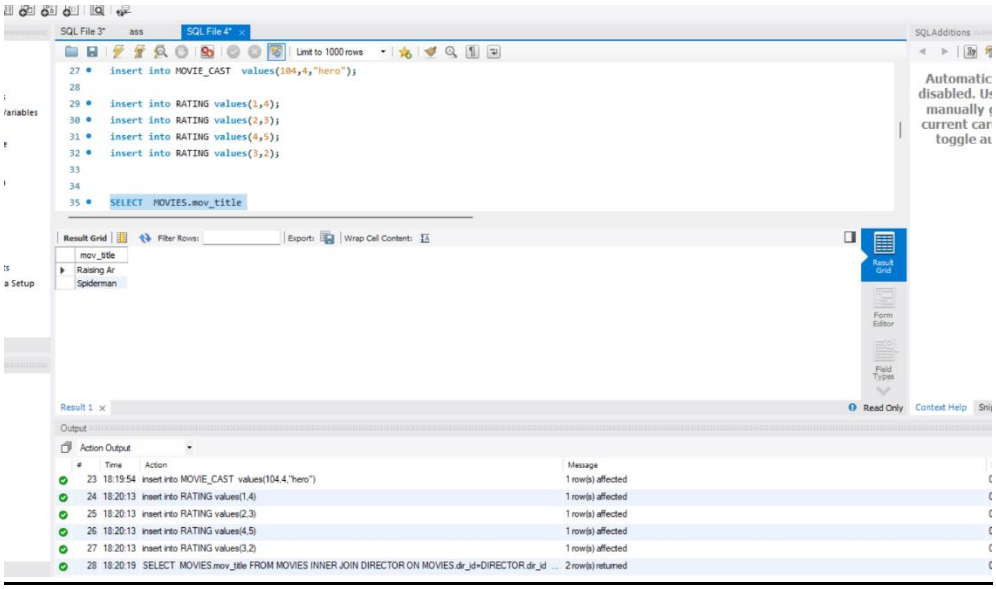
WHERE DIR_ID IN (SELECT DIR_ID

FROM DIRECTOR

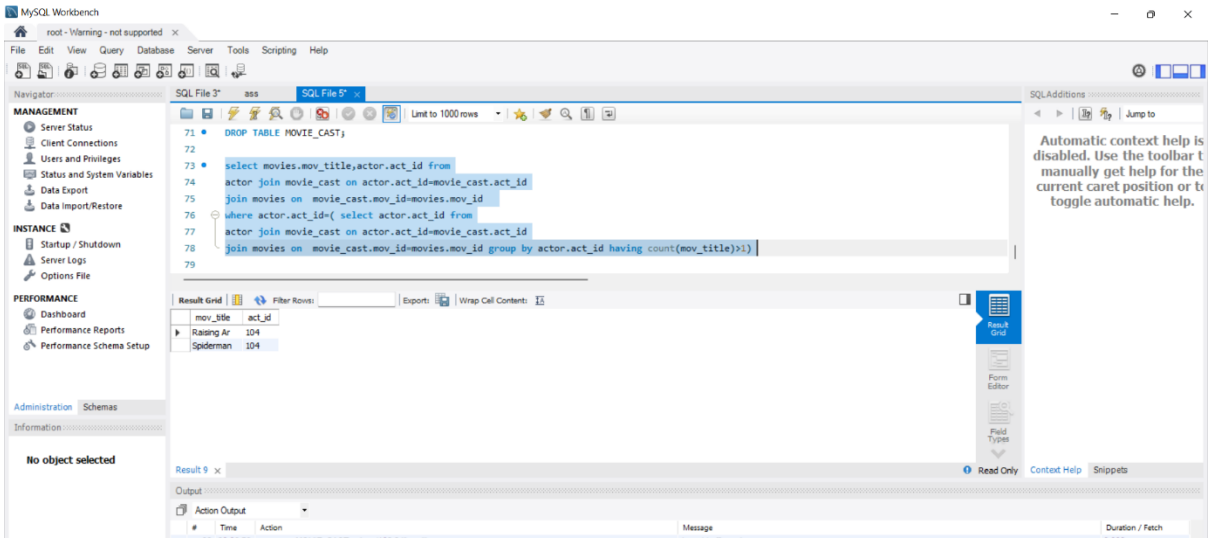
WHERE DIR_NAME='Steven Spielberg'));

OUTPUTS:

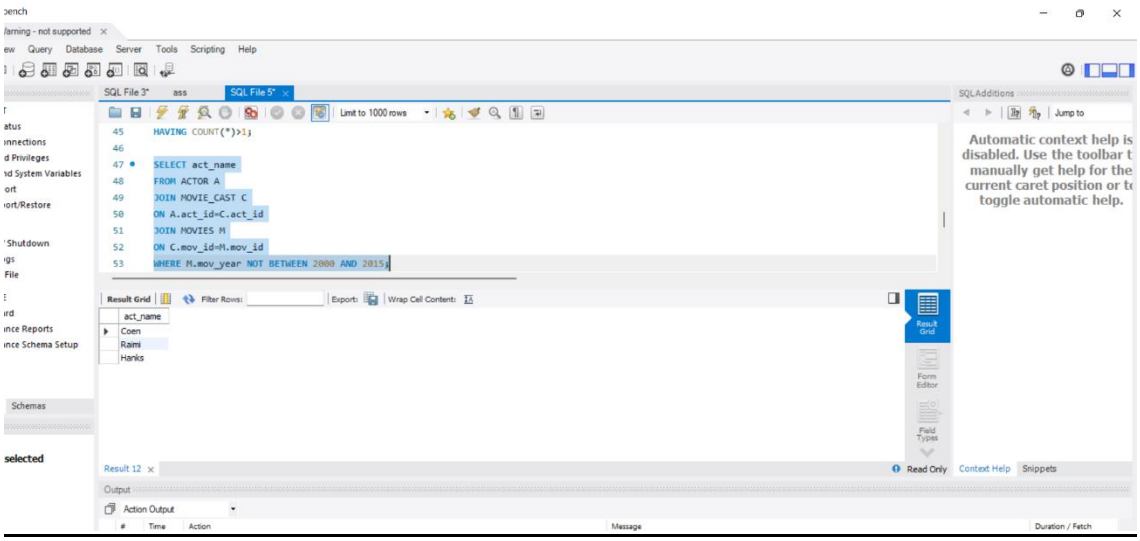
1


insert into MOVIE_CAST values(104,4,"hero");
insert into RATING values(1,4);
insert into RATING values(2,3);
insert into RATING values(4,5);
insert into RATING values(3,2);
SELECT MOVIES.mov_title
mov_title
Raising Ar
Spiderman
Action Output
23 18:19:54 insert into MOVIE_CAST values(104,4,"hero") 1 row(s) affected
24 18:20:13 insert into RATING values(1,4) 1 row(s) affected
25 18:20:13 insert into RATING values(2,3) 1 row(s) affected
26 18:20:13 insert into RATING values(4,5) 1 row(s) affected
27 18:20:13 insert into RATING values(3,2) 1 row(s) affected
28 18:20:19 SELECT MOVIES.mov_title FROM MOVIES INNER JOIN DIRECTOR ON MOVIES.dir_id=DIRECTOR.dir_id ... 2 row(s) returned

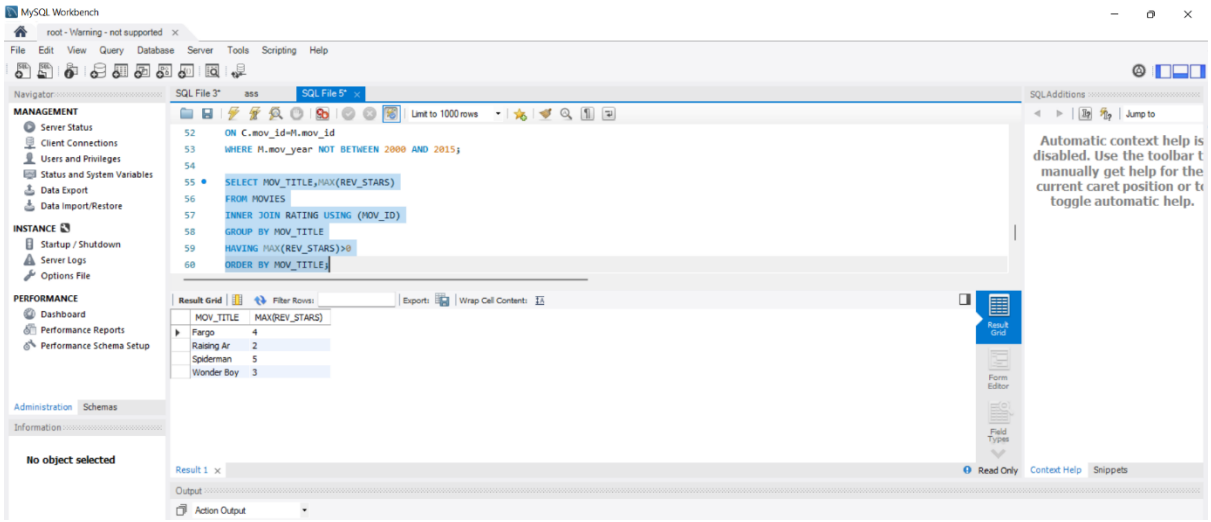
2


MySQL Workbench
DROP TABLE MOVIE_CAST;
select movies.mov_title,actor.act_id from
actor join movie_cast on actor.act_id=movie_cast.act_id
join movies on movie_cast.mov_id=movies.mov_id
where actor.act_id=(select actor.act_id from
actor join movie_cast on actor.act_id=movie_cast.act_id
join movies on movie_cast.mov_id=movies.mov_id group by actor.act_id having count(mov_title)>1)
mov_title act_id
Raising Ar 104
Spiderman 104
No object selected

3


HAVING COUNT(*)>1;
SELECT act_name
FROM ACTOR A
JOIN MOVIE_CAST C
ON A.act_id=C.act_id
JOIN MOVIES M
ON C.mov_id=M.mov_id
WHERE M.mov_year NOT BETWEEN 2000 AND 2015;
act_name
Coen
Raimi
Hanks

4


ON C.mov_id=M.mov_id
WHERE M.mov_year NOT BETWEEN 2000 AND 2015;
SELECT MOV_TITLE,MAX(REV_STARS)
FROM MOVIES
INNER JOIN RATING USING (MOV_ID)
GROUP BY MOV_TITLE
HAVING MAX(REV_STARS)>0
ORDER BY MOV_TITLE;
MOV_TITLE
MAX(REV_STARS)
Fargo 4
Raising Ar 2
Spiderman 5
Wonder Boy 3
No object selected