

MOVIE DATABASE OUTPUT

SHOW TABLES :-

```
3
4 • create table actor(act_id int, act_name varchar(30), act_gender enum('M','F'), primary key(act_id));
5 • create table director(dir_id int, dir_name varchar(30), dir_phone varchar(10), primary key(dir_id));
6 create table movies(mov_id int, mov_title varchar(30), mov_year year, mov_lang varchar(10), dir_id int, primary key(mov_id), foreign key(dir_id) references director(dir_id) on
7 create table moviecast(act_id int, mov_id int, part varchar(20), primary key(act_id, mov_id), foreign key(act_id) references actor(act_id) on delete cascade,
8 foreign key(mov_id) references movies(mov_id) on delete cascade);
9 • create table rating(mov_id int, rev_stars float, primary key(mov_id, rev_stars), foreign key(mov_id) references movies(mov_id) on delete cascade);
10
11 show tables;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Tables_in_moviedb

- actor
- director
- moviecast
- movies
- rating

Result Grid
Form Editor

SELECT * FROM ACTOR;






```
14
15 • insert into actor values(100, "Leo DiCaprio", 'M');
16 • insert into actor values(101, "Tom Hanks", 'M');
17 • insert into actor values(102, "Tom Cruise", 'M');
18 • insert into actor values(103, "Margot Robbie", 'F');
19 insert into actor values(104, "Jennifer Aniston", 'F');
20 insert into actor values(105, "Gal Gadot", 'F');
21
22 select * from actor;
```

Result Grid | Filter Rows: | Edit: | Export/Imp

	act_id	act_name	act_gender
▶	100	Leo DiCaprio	M
	101	Tom Hanks	M
	102	Tom Cruise	M
	103	Margot Robbie	F
	104	Jennifer Aniston	F
	105	Gal Gadot	F
*	NULL	NULL	NULL

SELECT * FROM DIRECTOR ;

```
23
24 • insert into director values(200, 'Steven Spielberg', '1649503470');
25   insert into director values(201, 'Alfred Hitchcock', '7989467865');
26 • insert into director values(202, 'James Cameron', '5218281077');
27   insert into director values(203, 'Kathryn Bigelow', '6157228013');
28   insert into director values(204, 'Niki Caro', '8976600547');
29   insert into director values(205, 'Sofia Coppola', '3949875040');
30
31   select * from director;
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit:   			
Export/Import:  			
	dir_id	dir_name	dir_phone
▶	200	Steven Spielberg	1649503470
	201	Alfred Hitchcock	7989467865
	202	James Cameron	5218281077
	203	Kathryn Bigelow	6157228013
	204	Niki Caro	8976600547
	205	Sofia Coppola	3949875040
*	NULL	NULL	NULL

SELECT * FROM MOVIES

```

33     insert into movies values(300, 'Avatar', 2010, 'EN', 202);
34 •   insert into movies values(301, 'Dial M For Murder', 1990, 'EN', 201);
35     insert into movies values(302, 'Jurassic Park 1', 1999, 'EN', 200);
36     insert into movies values(303, 'Jurassic Park 2', 2017, 'EN', 200);
37     insert into movies values(304, 'Vertigo', 1986, 'EN', 201);
38     insert into movies values(305, 'Zero Dark Thirty', 2012, 'EN', 200);
39
40 •   select * from movies;

```

Result Grid					Filter Rows:	Edit:	Export/Import:	Wr
mov_id	mov_title	mov_year	mov_lang	dir_id				
300	Avatar	2010	EN	202				
301	Dial M For Murder	1990	EN	201				
302	Jurassic Park 1	1999	EN	200				
303	Jurassic Park 2	2017	EN	200				
304	Vertigo	1986	EN	201				
305	Zero Dark Thirty	2012	EN	200				
NULL	NULL	NULL	NULL	NULL				

SELECT * FROM MOVIE CAST

```

46 insert into moviecast values(100, 302, 'actor');
47 insert into moviecast values(104, 302, 'actress');
48 insert into moviecast values(100, 303, 'actor');
49 • insert into moviecast values(104, 303, 'actress');
50 insert into moviecast values(102, 304, 'actor');
51 • insert into moviecast values(105, 304, 'actress');
52 insert into moviecast values(103, 305, 'actress');
53
54 select * from moviecast;

```

Result Grid			
Filter Rows:			
	act_id	mov_id	part
▶	100	302	actor
	100	303	actor
	101	300	actor
	102	301	actor
	102	304	actor
	103	301	actress
	103	305	actress
	104	302	actress
	104	303	actress
	105	300	actress
	105	304	actress

SELECT * FROM RATING;

```

56     insert into rating values(300, 4.5);
57     insert into rating values(301, 3);
58 •   insert into rating values(302, 4);
59     insert into rating values(303, 3.5);
60 •   insert into rating values(304, 5);
61     insert into rating values(305, 4);
62
63 •   select * from rating;

```

Result Grid | Filter Rows: Edit:

	mov_id	rev_stars
▶	300	4.5
	301	3
	302	4
	303	3.5
	304	5
	305	4
•	NULL	NULL

```

--
65     -- Queries
66
67     -- Q1
68 •   select m.mov_title from movies m, director d where m.dir_id=d.dir_id and d.dir_name='Alfred Hitchcock';

```

Result Grid | Filter Rows: Export: Wrap Cell Content: ☒

	mov_title
▶	Dial M For Murder
	Vertigo

```

70     -- Q2
71 •   select m.mov_title from movies m, moviecast mc where m.mov_id=mc.mov_id and act_id in( select act_id from moviecast group by act_id having count(act_id) > 1) group by m.mov_tit

```

Result Grid | Filter Rows: Export: Wrap Cell Content: ☒

	mov_title
▶	Dial M For Murder
	Jurassic Park 1
	Jurassic Park 2
	Vertigo



```

73 -- Q3
74 • select a.act_name, m.mov_title, m.mov_year from actor a, movies m, moviecast mc where a.act_id=mc.act_id and mc.mov_id=m.mov_id and m.mov_year not between 2000 and 2015;

```

Result Grid		
Filter Rows:		
act_name	mov_title	mov_year
Leo DiCaprio	Jurassic Park 1	1999
Leo DiCaprio	Jurassic Park 2	2017
Tom Cruise	Dial M For Murder	1990
Tom Cruise	Vertigo	1986
Margot Robbie	Dial M For Murder	1990
Jennifer Aniston	Jurassic Park 1	1999
Jennifer Aniston	Jurassic Park 2	2017
Gal Gadot	Vertigo	1986

```

76 -- Q4
77 • 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;

```

Result Grid	
Filter Rows:	
mov_title	max(rev_stars)
Avatar	4.5
Dial M For Murder	3
Jurassic Park 1	4
Jurassic Park 2	3.5
Vertigo	5
Zero Dark Thirty	4

Result Grid			
Filter Rows:			
	dir_id	dir_name	dir_phone
▶	200	Steven Spielberg	1649503470
	201	Alfred Hitchcock	7989467865
	202	James Cameron	5218281077
	203	Kathryn Bigelow	6157228013
	204	Niki Caro	8976600547
	205	Sofia Coppola	3949875040
✱	NULL	NULL	NULL