

# SBA 3 - MySQL

Adithya K Prabhu - 211480

---

1. Create Database project\_movie\_data and deploy table data according to ER diagram.

Sol : create database project\_movie\_data;

2. Create 9 tables which are presented in ER model and load the data with their foreign key and primary key values.

Soln :

### CREATING THE TABLES

```
create table actor(act_id int primary key,act_fname
char(20),act_lname char(20),act_gender char(1));
desc actor;
```

Field	Type	Null	Key	Default	Extra
act_id	int	NO	PRI	NULL	
act_fname	char(20)	YES		NULL	
act_lname	char(20)	YES		NULL	
act_gender	char(1)	YES		NULL	

```
create table movie_cast(act_id int,mov_id int, role
char(30), foreign key (act_id) references
actor(act_id),foreign key (mov_id) references
movie(mov_id));
desc movie_cast;
```

Field	Type	Null	Key	Default	Extra
act_id	int	YES	MUL	NULL	
mov_id	int	YES	MUL	NULL	
role	char(30)	YES		NULL	

```
create table director(dir_id int primary key,dir_fname
char(20),dir_lname char(20));
```

```
desc director;
```

Field	Type	Null	Key	Default	Extra
dir_id	int	NO	PRI	NULL	
dir_fname	char(20)	YES		NULL	
dir_lname	char(20)	YES		NULL	

```
create table movie(mov_id int primary key,mov_title
char(50),mov_year int,mov_time int,mov_lang
char(50),mov_dt_rel date,mov_rel_country char(5));
desc movie;
```

Field	Type	Null	Key	Default	Extra
mov_id	int	NO	PRI	NULL	
mov_title	char(50)	YES		NULL	
mov_year	int	YES		NULL	
mov_time	int	YES		NULL	
mov_lang	char(50)	YES		NULL	
mov_dt_rel	date	YES		NULL	
mov_rel_country	char(5)	YES		NULL	

```
create table movie_direction(dir_id int,mov_id int,foreign
key (dir_id) references director(dir_id), foreign
key(mov_id) references movie(mov_id));
```

```
desc movie_direction;
```

Field	Type	Null	Key	Default	Extra
dir_id	int	YES	MUL	NULL	
mov_id	int	YES	MUL	NULL	

```
create table reviewer(rev_id int primary key, rev_name
char(20));
```

```
desc reviewer;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
rev_id	int	NO	PRI	NULL	
rev_name	char(20)	YES		NULL	

```
create table genres(gen_id int primary key,gen_title
char(20));
```

```
desc genres;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
gen_id	int	NO	PRI	NULL	
gen_title	char(20)	YES		NULL	

```
create table mov_genres(mov_id int,gen_id int, foreign key
(mov_id) references movie(mov_id),foreign key (gen_id)
references genres(gen_id));
```

```
desc mov_genres;
```

Result Grid

Filter Rows:




Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
	mov_id	int	YES	MUL	NULL	
	gen_id	int	YES	MUL	NULL	

```
create table rating(mov_id int,rev_id int, rev_stars
decimal(8,2), num_o_ratings int,foreign key (mov_id)
references movie(mov_id), foreign key (rev_id) references
reviewer(rev_id));
```

desc rating;

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

Field	Type	Null	Key	Default	Extra
mov_id	int	YES	MUL	NULL	
rev_id	int	YES	MUL	NULL	
rev_stars	int	YES		NULL	
num_o_ratings	int	YES		NULL	

### INSERTING VALUES TO THE TABLES:







#### **Actor**

```

insert into actor values(101,'James','Stewart','M');
insert into actor values(102,'Deborah','Kerr','F');
insert into actor values(103,'Peter','Otoole','M');
insert into actor values(104,'Robert','De niro','M');
insert into actor values(105,'F Murray','Abraham','M');
insert into actor values(106,'Harisson','Ford','M');
insert into actor values(107,'Nicole','Kidman','M');
insert into actor values(108,'Stephen','Baldwin','M');
insert into actor values(109,'Jack','Nicholson','M');
insert into actor values(110,'Mark','Wahlberg','M');
insert into actor values(111,'Woody','Allen','M');
insert into actor values(112,'Claire','Danes','F');
insert into actor values(113,'Tim','Robbins','M');
insert into actor values(114,'Kevin','Spacey','M');
insert into actor values(115,'Kate','Winslet','F');
insert into actor values(116,'Robbin','Williams','M');
insert into actor values(117,'Jon','Voight','M');
insert into actor values(118,'Ewan','Mcgregor','M');
insert into actor values(119,'Christian','Bale','M');
insert into actor values(120,'Maggie','Gyllenhaal','F');
insert into actor values(121,'Dev','Patel','M');
insert into actor values(122,'Sigourney','Weaver','F');
insert into actor values(123,'David','Aston','M');
insert into actor values(124,'Ali','Astin','F');

```

```
select * from actor;
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit:   			
Export/Import:  			
Wrap Cell Content: 			
act_id	act_fname	act_lname	act_gender
101	James	Stewart	M
102	Deborah	Kerr	F
103	Peter	Otoole	M
104	Robert	De niro	M
105	F Murray	Abraham	M
106	Harisson	Ford	M
107	Nicole	Kidman	M
108	Stephen	Baldwin	M
109	Jack	Nicholson	M
110	Mark	Wahlberg	M
111	Woody	Allen	M
112	Claire	Danes	F
113	Tim	Robbins	M
114	Kevin	Spacey	M
115	Kate	Winslet	F
116	Robbin	Williams	M
117	Jon	Voight	M
118	Ewan	Mcgregor	M
119	Christian	Bale	M
120	Maggie	Gyllenhaal	F
121	Dev	Patel	M
122	Sigourney	Weaver	F
123	David	Aston	M
124	Ali	Astin	F

actor 18 x

## Director

```
insert into director values(201,'Alfred','Hitchcock');
insert into director values(202,'Jack','Clayton');
insert into director values(203,'David','Lean');
insert into director values(204,'Michael','Cimino');
insert into director values(205,'Milos','Forman');
insert into director values(206,'Ridley','Scott');
insert into director values(207,'Stanley','Kubrik');
insert into director values(208,'Bryan','Singer');
insert into director values(209,'Roman','Polanski');
insert into director values(210,'Paul','Thomas Anderson');
insert into director values(211,'Woody','Allen');
insert into director values(212,'Hayao','Mikazaki');
```

```

insert into director values(213,'Frank','Darabont');
insert into director values(214,'Sam','Mendes');
insert into director values(215,'James','Cameron');
insert into director values(216,'Gus','Van sant');
insert into director values(217,'John','Boorman');
insert into director values(218,'Danny','Boyle');
insert into director values(219,'Christopher','Nolan');
insert into director values(220,'Richard','Kelly');
insert into director values(221,'Kevin','Spacey');
insert into director values(222,'Andrei','Takovsky');
insert into director values(223,'Peter','Jackson');
select * from director;

```

	dir_id	dir_fname	dir_lname
▶	201	Alfred	Hitchcock
	202	Jack	Clayton
	203	David	Lean
	204	Michael	Cimino
	205	Milos	Forman
	206	Ridley	Scott
	207	Stanley	Kubrik
	208	Bryan	Singer
	209	Roman	Polanski
	210	Paul	Thomas Anderson
	211	Woody	Allen
	212	Hayao	Mikazaki
	213	Frank	Darabont
	214	Sam	Mendes
	215	James	Cameron
	216	Gus	Van sant
	217	John	Boorman
	218	Danny	Boyle
	219	Christopher	Nolan
	220	Richard	Kelly
	221	Kevin	Spacey
	222	Andrei	Takovsky
	223	Peter	Jackson
✱	NULL	NULL	NULL

## Movie

```

insert into movie
values(901,'Vertigo',1958,128,'English','19580824','UK');

insert into movie values(902,'The
innocents',1961,100,'English','19620219','Sw');

```

```
insert into movie values(903,'Lawrence of
arabia',1962,216,'English','19621211','UK');

insert into movie values(904,'The deer
hunter',1978,183,'English','19790308','UK');

insert into movie
values(905,'Amadeus',1984,160,'English','19850107','UK');

insert into movie values(906,'Blade
runner',1982,117,'English','19820909','UK');

insert into movie values(907,'Eyes wide
shut',1999,159,'English','', 'UK');

insert into movie values(908,'The usual
Suspects',1995,106,'English','19950825','UK');

insert into movie
values(909,'Chinatown',1974,130,'English','19740809','UK')
;

insert into movie values(910,'Boogie
Nights',1997,155,'English','19980216','UK');

insert into movie values(911,'Annie
Hall',1977,93,'English','19770420','USA');

insert into movie values(912,'Princess
monokonke',1997,134,'Japanese','20011019','UK');

insert into movie values(913,'The Shawshank
Redemption',1994,142,'English','19950217','UK');

insert into movie values(914,'American
beauty',1999,122,'English',null,'UK');

insert into movie
values(915,'Titanic',1997,194,'English','19980123','UK');

insert into movie values(916,'Good Will
Hunting',1997,126,'English','19980603','UK');

insert into movie
values(917,'Deliverance',1972,109,'English','19821005','UK
');

insert into movie
values(918,'Trainspotting',1996,94,'English','19960223','U
K');
```



```

insert into movie values(919,'The
Prestige',2006,130,'English','20061110','UK');

insert into movie values(920,'Donnie
Darko',2001,113,'English',null,'UK');

insert into movie values(921,'Slumdog
millionaire',2008,120,'English','20090109','UK');

insert into movie
values(922,'Aliens',1986,137,'English','19860929','UK');

insert into movie values(923,'Beyond The
Sea',2004,118,'English','20041126','UK');

insert into movie
values(924,'Avatar',2009,162,'English',null,'UK');

insert into movie values(926,'Seven
Samurai',1954,207,'Japanese','19540426','JP');

insert into movie values(927,'Spirited
Away',2001,125,'Japanese','20030912','UK');

insert into movie values(928,'BAck to the
future',1985,116,'English','19851204','UK');

insert into movie
values(925,'BraveHeart',1995,178,'English','19950908','UK'
);

select * from movie;

```

mov_id	mov_title	mov_year	mov_time	mov_lang	mov_release	mov_release_country
901	Vertigo	1958	128	English	1958-08-24	UK
902	The innocents	1961	100	English	1962-02-19	Sw
903	Lawrence of arabia	1962	216	English	1962-12-11	UK
904	The deer hunter	1978	183	English	1979-03-08	UK
905	Amadeus	1984	160	English	1985-01-07	UK
906	Blade runner	1982	117	English	1982-09-09	UK
907	Eyes wide shut	1999	159	English	NULL	UK
908	The usual Suspects	1995	106	English	1995-08-25	UK
909	Chinatown	1974	130	English	1974-08-09	UK
910	Boogie Nights	1997	155	English	1998-02-16	UK
911	Annie Hall	1977	93	English	1977-04-20	USA
912	Princess monokonge	1997	134	Japanese	2001-10-19	UK
913	The Shawshank R...	1994	142	English	1995-02-17	UK
914	American beauty	1999	122	English	NULL	UK
915	Titanic	1997	194	English	1998-01-23	UK
916	Good Will Hunting	1997	126	English	1998-06-03	UK
917	Deliverance	1972	109	English	1982-10-05	UK
918	Trainspotting	1996	94	English	1996-02-23	UK
919	The Prestige	2006	130	English	2006-11-10	UK
920	Donnie Darko	2001	113	English	NULL	UK
921	Slumdog millionaire	2008	120	English	2009-01-09	UK
922	Aliens	1986	137	English	1986-09-29	UK
923	Beyond The Sea	2004	118	English	2004-11-26	UK
924	Avatar	2009	162	English	NULL	UK
925	BraveHeart	1995	178	English	1995-09-08	UK
926	Seven Samurai	1954	207	Japanese	1954-04-26	JP
927	Spirited Away	2001	125	Japanese	2003-09-12	UK
928	BAck to the future	1985	116	English	1985-12-04	UK
NULL	NULL	NULL	NULL	NULL	NULL	NULL

### **Reviewer**

```
insert into reviewer values(9001,'Righty sock');
insert into reviewer values(9002,'Jack Malvern');
insert into reviewer values(9003,'Flagrant Baronessia');
insert into reviewer values(9004,'Alec Shaw');
insert into reviewer values(9005,null);
insert into reviewer values(9006,'Viictor Woeltjen');
insert into reviewer values(9007,'Simon Wright');
insert into reviewer values(9008,'Neal Wruck');
insert into reviewer values(9009,'Paul monks');
insert into reviewer values(9010,'Mike salvati');
insert into reviewer values(9011,null);
insert into reviewer values(9012,'Wesley S Walker');
insert into reviewer values(9013,'Sasha Goldstein');
insert into reviewer values(9014,'Josh Cates');
insert into reviewer values(9015,'Krug Stillo');
insert into reviewer values(9016,'Scott LeBrun');
insert into reviewer values(9017,'Hannah steele');
insert into reviewer values(9018,'Vincent Cadena');
insert into reviewer values(9019,'Brandt Sponsellor');
insert into reviewer values(9020,'Richard Adams');
select * from reviewer;
```

Result Grid			Filter Rows:
	rev_id	rev_name	
▶	9001	Righty sock	
	9002	Jack Malvern	
	9003	Flagrant Baronessia	
	9004	Alec Shaw	
	9005	NULL	
	9006	Victor Woeltjen	
	9007	Simon Wright	
	9008	Neal Wruck	
	9009	Paul monks	
	9010	Mike salvati	
	9011	NULL	
	9012	Wesley S Walker	
	9013	Sasha Goldstein	
	9014	Josh Cates	
	9015	Krug Stillo	
	9016	Scott LeBrun	
	9017	Hannah steele	
	9018	Vincent Cadena	
	9019	Brandt Sponsellor	
	9020	Richard Adams	

## Genres

```

insert into genres values(1001,'Action');
insert into genres values(1002,'Adventure');
insert into genres values(1003,'Animation');
insert into genres values(1004,'Biography');
insert into genres values(1005,'Comedy');
insert into genres values(1006,'Crime');

```

```

insert into genres values(1007,'Drama');
insert into genres values(1008,'Horror');
insert into genres values(1009,'Music');
insert into genres values (1010,'Mystery');
insert into genres values(1011,'Romance');
insert into genres values(1012,'Thriller');
insert into genres values(1013,'War');
select * from genres;

```

Result Grid | Filter Rows:

	gen_id	gen_title
▶	1001	Action
	1002	Adventure
	1003	Animation
	1004	Biography
	1005	Comedy
	1006	Crime
	1007	Drama
	1008	Horror
	1009	Music
	1010	Mystery
	1011	Romance
	1012	Thriller
	1013	War
●	NULL	NULL

### Movie\_direction

```

insert into movie_direction values(201,901);
insert into movie_direction values(202,902);
insert into movie_direction values(203,903);
insert into movie_direction values(204,904);
insert into movie_direction values(205,905);
insert into movie_direction values(206,906);
insert into movie_direction values(207,907);

```

```
insert into movie_direction values(208,908);
insert into movie_direction values(209,909);
insert into movie_direction values(210,910);
insert into movie_direction values(211,911);
insert into movie_direction values(212,912);
insert into movie_direction values(213,913);
insert into movie_direction values(214,914);
insert into movie_direction values(215,915);
insert into movie_direction values(216,916);
insert into movie_direction values(217,917);
insert into movie_direction values(218,918);
insert into movie_direction values(219,919);
insert into movie_direction values(220,920);
insert into movie_direction values(218,921);
insert into movie_direction values(215,922);
insert into movie_direction values(221,923);
select *from movie_direction;
```

dir_id	mov_id
201	901
202	902
203	903
204	904
205	905
206	906
207	907
208	908
209	909
210	910
211	911
212	912
213	913
214	914
215	915
216	916
217	917
218	918
219	919
220	920
218	921
215	922
221	923

### **Movie Genres**

```
insert into mov_genres values(922,1001);
insert into mov_genres values(917,1002);
insert into mov_genres values(903,1002);
insert into mov_genres values(912,1003);
insert into mov_genres values(911,1005);
insert into mov_genres values(908,1006);
insert into mov_genres values(913,1006);
insert into mov_genres values(926,1007);
insert into mov_genres values(928,1007);
insert into mov_genres values(918,1007);
insert into mov_genres values(921,1007);
insert into mov_genres values(902,1008);
insert into mov_genres values(923,1009);
insert into mov_genres values(907,1010);
insert into mov_genres values(927,1010);
insert into mov_genres values(901,1001);
insert into mov_genres values(914,1011);
insert into mov_genres values(906,1012);
insert into mov_genres values(904,1013);
select * from mov_genres;
```

	mov_id	gen_id
▶	922	1001
	917	1002
	903	1002
	912	1003
	911	1005
	908	1006
	913	1006
	926	1007
	928	1007
	918	1007
	921	1007
	902	1008
	923	1009
	907	1010
	927	1010
	901	1001
	914	1011
	906	1012
	904	1013



### Ratings

```
insert into rating values(901,9001,8.4,263575);
insert into rating values(902,9002,7.9,20207);
insert into rating values(903,9003,8.3,202778);
insert into rating values(906,9005,8.2,484746);
insert into rating values(924,9006,7.3,null);
```

```

insert into rating values(908,9007,8.6,779489);
insert into rating values(909,9008,null,227235);
insert into rating values(910,9009,3,195961);
insert into rating values(911,9010,8.1,203875);
insert into rating values(912,9011,8.4,null);
insert into rating values(914,9013,7,862618);
insert into rating values(915,9001,7.7,830095);
insert into rating values(916,9014,4,642132);
insert into rating values(925,9015,7.7,81328);
insert into rating values(918,9016,null,580301);
insert into rating values(920,9017,8.1,609451);
insert into rating values(921,9018,8,667758);
insert into rating values(922,9019,8.4,511613);
insert into rating values(923,9020,6.7,13091);
select * from rating;

```

Result Grid   Filter Rows: <input type="text"/>				
	mov_id	rev_id	rev_stars	num_o_ratings
▶	901	9001	8.40	263575
	902	9002	7.90	20207
	903	9003	8.30	202778
	906	9005	8.20	484746
	924	9006	7.30	NULL
	908	9007	8.60	779489
	909	9008	NULL	227235
	910	9009	3.00	195961
	911	9010	8.10	203875
	912	9011	8.40	NULL
	914	9013	7.00	862618
	915	9001	7.70	830095
	916	9014	4.00	642132
	925	9015	7.70	81328
	918	9016	NULL	580301
	920	9017	8.10	609451
	921	9018	8.00	667758
	922	9019	8.40	511613
	923	9020	6.70	13091



**Movie\_cast**

```
insert into movie_cast values(101,901,'John Scottie
Ferguson');
insert into movie_cast values(102,902,'MissGiddens');
insert into movie_cast values(103,903,'T E lawrence');
insert into movie_cast values(104,904,'Michael');
insert into movie_cast values(105,905,'Antonio Salieri');
insert into movie_cast values(106,906,'Rick deckard');
insert into movie_cast values(107,907,'Alice harford');
insert into movie_cast values(108,908,'Mcmanus');
insert into movie_cast values(110,910,'Eddie adams');
insert into movie_cast values(111,911,'Alvy singer');
insert into movie_cast values(112,912,'San');
insert into movie_cast values(113,913,'Andy Dufresne');
insert into movie_cast values(114,914,'Lester burham');
insert into movie_cast values(115,915,'Rose dewitt
bukater');
insert into movie_cast values(116,916,'Sean Maguire');
insert into movie_cast values(117,917,'Ed');
insert into movie_cast values(118,918,'Renton');
insert into movie_cast values(120,920,'Elizebath Darko');
insert into movie_cast values(121,921,'Older jamal');
insert into movie_cast values(122,922,'Ripley');
insert into movie_cast values(114,923,'Bobby Darin');
insert into movie_cast values(109,909,'J J Gittes');
insert into movie_cast values(119,919,'Alfred Borden');
```

```
select * from movie_cast;
```

Result Grid			
Filter Rows: <input type="text"/>			
	act_id	mov_id	role
▶	101	901	John Scottie Ferguson
	102	902	MissGiddens
	103	903	T E lawrence
	104	904	Michael
	105	905	Antonio Salieri
	106	906	Rick deckard
	107	907	Alice harford
	108	908	Mcmamus
	110	910	Eddie adams
	111	911	Alvy singer
	112	912	San
	113	913	Andy Dufresne
	114	914	Lester burham
	115	915	Rose dewitt bukater
	116	916	Sean Maguire
	117	917	Ed
	118	918	Renton
	120	920	Elizebath Darko
	121	921	Older jamal
	122	922	Ripley
	114	923	Bobby Darin
	109	909	J J Gittes
	119	919	Alfred Borden

movie\_cast 41 x

3. Write a query in sql to list the horror movies.

Sol :

```
select mov_title from movie
INNER JOIN mov_genres mg
ON movie.mov_id = mg.mov_id
INNER JOIN genres g
ON mg.gen_id = g.gen_id
WHERE g.gen_title = 'Horror';
```

```

315 ON a.act_id = mc.act_id
316 INNER JOIN movie m
317 ON m.mov_id = mc.mov_id
318 WHERE m.mov_title='Boogie Nights';
319
320 • select mov_title from movie
321 INNER JOIN mov_genres mg
322 ON movie.mov_id = mg.mov_id
323 INNER JOIN genres g
324 ON mg.gen_id = g.gen_id
325 WHERE g.gen_title='Horror';

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

mov_title
The innocents

4. Write a query in SQL to find the name of all reviewers who have rated 8 or more stars?

Sol:

```

select rev_name from reviewer r
INNER JOIN rating ra
ON r.rev_id = ra.rev_id
where ra.rev_stars >= 8.00;

```

```

262 • select rev_name from reviewer r
263 INNER JOIN rating ra
264 ON r.rev_id = ra.rev_id
265 where ra.rev_stars >= 8.00;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

rev_name
Righty sock
Flagrant Baronessia
Simon Wright
Mike salvati
Hannah steele
Vincent Cadena
Brandt Sponseller

5. Write a query in SQL to list all the information of the actors who played a role in the movie 'Deliverance'

Sol:

```

SELECT
a.act_id,a.act_fname,a.act_lname,a.act_gender,mo.mov_
title from actor a
INNER JOIN movie_cast m
ON a.act_id = m.act_id
INNER JOIN movie mo

```

```

ON mo.mov_id = m.mov_id
Where mo.mov_title = 'Deliverance';

```

262

263 • `select a.act_id,a.act_fname,a.act_lname,a.act_gender,mo.mov_title from actor a`

264 `INNER JOIN movie_cast m`

265 `ON a.act_id = m.act_id`

266 `INNER JOIN movie mo`

267 `ON mo.mov_id = m.mov_id`

268 `Where mo.mov_title = 'Deliverance';`

Result Grid

act_id	act_fname	act_lname	act_gender	mov_title
117	Jon	Voight	M	Deliverance

6. Write a query in SQL to find the name of the director (first and last names) who directed a movie that casted a role for 'Eyes Wide Shut'. (using subquery)

Sol :

```

select dir_fname,dir_lname from director
where dir_id IN (
select dir_id from movie_direction
where mov_id IN (
select mov_id from movie
where mov_title = 'Eyes wide shut'));

```

269

270 • `select dir_fname,dir_lname from director`

271 `where dir_id IN (`

272 `select dir_id from movie_direction`

273 `where mov_id IN (`

274 `select mov_id from movie`

275 `where mov_title = 'Eyes wide shut'));`

Result Grid

dir_fname	dir_lname
Stanley	Kubrik

7. Write a query in SQL to find the movie title, year, date of release, director and actor for those movies which reviewer is 'Neal Wruck'

Sol:

```

SELECT mov_title, mov_year, mov_dt_rel, dir_fname,
dir_lname, act_fname, act_lname
FROM movie m, movie_direction md, director d, rating
ra, reviewer r, actor a, movie_cast mc

```

```

WHERE m.mov_id=md.mov_id
AND md.dir_id=d.dir_id
AND m.mov_id=ra.mov_id
AND ra.rev_id=r.rev_id
AND m.mov_id=mc.mov_id
AND mc.act_id=a.act_id
AND r.rev_name ='Neal Wruck';

```

```

278 • SELECT mov_title, mov_year, mov_dt_rel, dir_fname, dir_lname, act_fname, act_lname
279 FROM movie m, movie_direction md, director d, rating ra, reviewer r, actor a, movie_cast mc
280 WHERE m.mov_id=md.mov_id
281 AND md.dir_id=d.dir_id
282 AND m.mov_id=ra.mov_id
283 AND ra.rev_id=r.rev_id
284 AND m.mov_id=mc.mov_id
285 AND mc.act_id=a.act_id
286 AND r.rev_name ='Neal Wruck';

```

mov_title	mov_year	mov_dt_rel	dir_fname	dir_lname	act_fname	act_lname
Chinatown	1974	1974-08-09	Roman	Polanski	Jack	Nicholson

8. Write a query in SQL to find all the years which produced at least one movie and that received a rating of more than 4 stars.

Sol:

```

Select DISTINCT
mov_year
FROM movie
WHERE mov_id IN
( SELECT mov_id
FROM rating
WHERE rev_stars>4 )
ORDER BY mov_year;

```

mov_year
1958
1961
1962
1977
1982
1986
1995
1997
1999
2001
2004
2008
2009

9. Write a query in SQL to find the name of all movies who have rated their ratings with a NULL value

Sol:

```

SELECT re.rev_name as REVIEVER,m.mov_title as
MOVIEName
FROM movie m

```

```

INNER JOIN rating r
ON r.mov_id = m.mov_id
INNER JOIN reviewer re
ON re.rev_id = r.rev_id
WHERE r.rev_stars IS NULL;

```

```

295
296 • SELECT re.rev_name as REVIEWER,m.mov_title as MOVIEName
297 FROM movie m
298 INNER JOIN rating r
299 ON r.mov_id = m.mov_id
300 INNER JOIN reviewer re
301 ON re.rev_id = r.rev_id
302 WHERE r.rev_stars IS NULL;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
REVIEWER	MOVIEName			
▶ Neal Wruck	Chinatown			
▶ Scott LeBrun	Trainspotting			

10. Write a query in SQL to find the name of movies who were directed by 'David'

Sol:

```

select m.mov_title,d.dir_fname,d.dir_lname
FROM movie m
INNER JOIN movie_direction md
ON m.mov_id = md.mov_id
INNER JOIN director d
ON d.dir_id = md.dir_id
WHERE d.dir_fname = 'David';

```

```

303
304 • select m.mov_title,d.dir_fname,d.dir_lname
305 FROM movie m
306 INNER JOIN movie_direction md
307 ON m.mov_id = md.mov_id
308 INNER JOIN director d
309 ON d.dir_id = md.dir_id
310 WHERE d.dir_fname = 'David';

```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
mov_title	dir_fname	dir_lname				
▶ Lawrence of arabia	David	Lean				

11. Write a query in SQL to list the first and last names of all the actors who were cast in the movie 'Boogie Nights', and the roles they played in that production.

Sol :

```
select
a.act_fname,a.act_lname,mc.role,m.mov_title
FROM actor a
INNER JOIN movie_cast mc
ON a.act_id = mc.act_id
INNER JOIN movie m
ON m.mov_id = mc.mov_id
WHERE m.mov_title='Boogie Nights';
```

```
311
312 • select a.act_fname,a.act_lname,mc.role,m.mov_title
313 FROM actor a
314 INNER JOIN movie_cast mc
315 ON a.act_id = mc.act_id
316 INNER JOIN movie m
317 ON m.mov_id = mc.mov_id
318 WHERE m.mov_title='Boogie Nights';
```

<  Filter Rows:  Export: Wrap Cell Content:

	act_fname	act_lname	role	mov_title
▶	Mark	Wahlberg	Eddie adams	Boogie Nights

-----XX-----