Consider the schema for Movie Database:

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)

CREATING TABLES:

CREATE DATABASE MOVIE_DATABASE;

CREATE TABLE ACTOR (
ACT_ID INTEGER PRIMARY KEY,
ACT_NAME VARCHAR(20),
ACT_GENDER CHAR(1));

CREATE TABLE DIRECTOR(
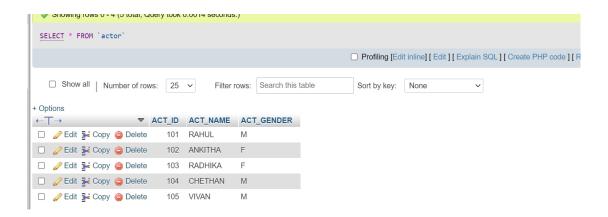DIR_ID INTEGER PRIMARY KEY,
DIR_NAME VARCHAR(20),
DIR_PHONE INTEGER);

CREATE TABLE MOVIES(
MOV_ID INTEGER PRIMARY KEY,
MOV_TITLE VARCHAR(25),
MOV_YEAR INTEGER,
MOV_LANG VARCHAR(15),
DIR_ID INTEGER,
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR(DIR_ID));

CREATE TABLE MOVIE_CAST(
ACT_ID INTEGER,
MOV_ID INTEGER,
ROLE VARCHAR(10),
PRIMARY KEY (ACT_ID,MOV_ID),
FOREIGN KEY (ACT_ID) REFERENCES ACTOR(ACT_ID),
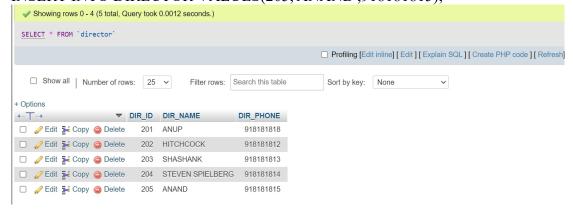FOREIGN KEY (MOV_ID) REFERENCES MOVIES(MOV_ID));

CREATE TABLE RATING(
MOV_ID INTEGER PRIMARY KEY,
REV_STARS VARCHAR(25),
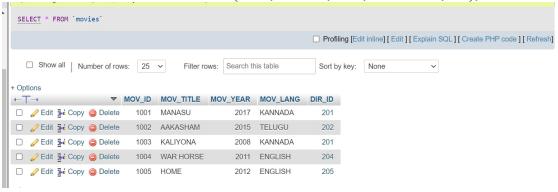FOREIGN KEY (MOV_ID) REFERENCES MOVIES(MOV_ID));

INSERTION:

INSERT INTO ACTOR VALUES(101,'RAHUL','M');
INSERT INTO ACTOR VALUES(102,'ANKITHA','F');
INSERT INTO ACTOR VALUES(103,'RADHIKA','F');
INSERT INTO ACTOR VALUES(104,'CHETHAN','M');
INSERT INTO ACTOR VALUES(105,'VIVAN','M');

```
SELECT * FROM `actor`
```

☐ Show all | Number of rows: 25 ⌄   Filter rows: Search this table   Sort by key: None ⌄

+ Options

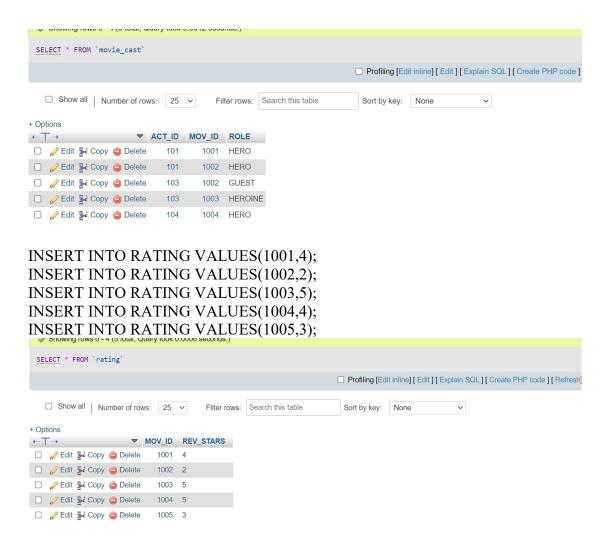|	ACT_ID	ACT_NAME	ACT_GENDER
☐ ✏ Edit ⌗ Copy ⊘ Delete	101	RAHUL	M
☐ ✏ Edit ⌗ Copy ⊘ Delete	102	ANKITHA	F
☐ ✏ Edit ⌗ Copy ⊘ Delete	103	RADHIKA	F
☐ ✏ Edit ⌗ Copy ⊘ Delete	104	CHETHAN	M
☐ ✏ Edit ⌗ Copy ⊘ Delete	105	VIVAN	M

INSERT INTO DIRECTOR VALUES(201,'ANUP',918181818);
INSERT INTO DIRECTOR VALUES(202,'HITCHCOCK',918181812);
INSERT INTO DIRECTOR VALUES(203,'SHASHANK',918181813);
INSERT INTO DIRECTOR VALUES(204,'STEVEN SPIELBERG',918181814);
INSERT INTO DIRECTOR VALUES(205,'ANAND',918181815);

```
SELECT * FROM `director`
```

☐ Show all | Number of rows: 25 ⌄   Filter rows: Search this table   Sort by key: None ⌄

+ Options

	DIR_ID	DIR_NAME	DIR_PHONE
☐ ✏ Edit ⌗ Copy ⊘ Delete	201	ANUP	918181818
☐ ✏ Edit ⌗ Copy ⊘ Delete	202	HITCHCOCK	918181812
☐ ✏ Edit ⌗ Copy ⊘ Delete	203	SHASHANK	918181813
☐ ✏ Edit ⌗ Copy ⊘ Delete	204	STEVEN SPIELBERG	918181814
☐ ✏ Edit ⌗ Copy ⊘ Delete	205	ANAND	918181815

INSERT INTO MOVIES VALUES(1001,'MANASU',2017,'KANNADA',201);
INSERT INTO MOVIES VALUES(1002,'AAKASHAM',2015,'TELUGU',202);
INSERT INTO MOVIES VALUES(1003,'KALIYONA',2008,'KANNADA',201);
INSERT INTO MOVIES VALUES(1004,'WAR HORSE',2011,'ENGLISH',204);
INSERT INTO MOVIES VALUES(1005,'HOME',2012,'ENGLISH',205);

```
SELECT * FROM `movies`
```

☐ Show all | Number of rows: 25 ⌄   Filter rows: Search this table   Sort by key: None ⌄

+ Options

	MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
☐ ✏ Edit ⌗ Copy ⊘ Delete	1001	MANASU	2017	KANNADA	201
☐ ✏ Edit ⌗ Copy ⊘ Delete	1002	AAKASHAM	2015	TELUGU	202
☐ ✏ Edit ⌗ Copy ⊘ Delete	1003	KALIYONA	2008	KANNADA	201
☐ ✏ Edit ⌗ Copy ⊘ Delete	1004	WAR HORSE	2011	ENGLISH	204
☐ ✏ Edit ⌗ Copy ⊘ Delete	1005	HOME	2012	ENGLISH	205

INSERT INTO MOVIE_CAST VALUES(101,1002,'HERO');
INSERT INTO MOVIE_CAST VALUES(101,1001,'HERO');
INSERT INTO MOVIE_CAST VALUES(103,1003,'HEROINE');
INSERT INTO MOVIE_CAST VALUES(103,1002,'GUEST');
INSERT INTO MOVIE_CAST VALUES(104,1004,'HERO');

```
SELECT * FROM `movie_cast`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

☐ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options

		ACT_ID	MOV_ID	ROLE
☐	✏ Edit � Copy ⊖ Delete	101	1001	HERO
☐	✏ Edit ⌖ Copy ⊖ Delete	101	1002	HERO
☐	✏ Edit ⌖ Copy ⊖ Delete	103	1002	GUEST
☐	✏ Edit ⌖ Copy ⊖ Delete	103	1003	HEROINE
☐	✏ Edit ⌖ Copy ⊖ Delete	104	1004	HERO

INSERT INTO RATING VALUES(1001,4);
INSERT INTO RATING VALUES(1002,2);
INSERT INTO RATING VALUES(1003,5);
INSERT INTO RATING VALUES(1004,4);
INSERT INTO RATING VALUES(1005,3);

```
SELECT * FROM `rating`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options

		MOV_ID	REV_STARS
☐	✏ Edit ⌖ Copy ⊖ Delete	1001	4
☐	✏ Edit ⌖ Copy ⊖ Delete	1002	2
☐	✏ Edit ⌖ Copy ⊖ Delete	1003	5
☐	✏ Edit ⌖ Copy ⊖ Delete	1004	5
☐	✏ Edit ⌖ Copy ⊖ Delete	1005	3

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

SELECT MOV_TITLE
FROM MOVIES
WHERE DIR_ID = (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME='HITCHCOCK')

Show query box

```
SELECT MOV_TITLE FROM MOVIES WHERE DIR_ID = (SELECT DIR_ID FROM DIRECTOR WHERE DIR_NAME='HITCHCOCK')
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options

		MOV_TITLE
☐	✏ Edit ⌖ Copy ⊖ Delete	AAKASHAM

ii. Find the movie names where one or more actors acted in two or more movies.

```
SELECT MOV_TITLE
FROM MOVIES M,MOVIE_CAST MC
WHERE M.MOV_ID=MC.MOV_ID AND ACT_ID IN (SELECT ACT_ID
FROM MOVIE_CAST GROUP BY ACT_ID
HAVING COUNT(ACT_ID)>1)
GROUP BY MOV_TITLE
HAVING COUNT(*)>1;
```


Showing rows 0 - 0 (1 total, Query took 0.0086 seconds.)

SELECT MOV_TITLE FROM MOVIES M,MOVIE_CAST MC WHERE M.MOV_ID=MC.MOV_ID AND ACT_ID IN (SELECT ACT_ID FROM MOVIE_CAST GROUP BY ACT_ID HAVING COUNT(ACT_ID)>1) GROUP BY MOV_TITLE HAVING COUNT(*)>1

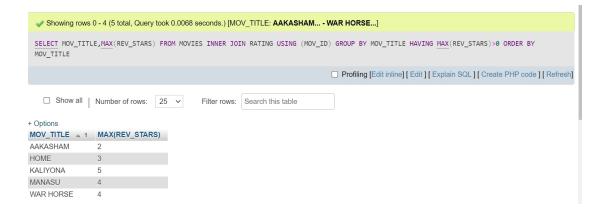Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ∨ Filter rows: Search this table

+ Options
MOV_TITLE
AAKASHAM

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

```
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;
```

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

Showing rows 0 - 0 (1 total, Query took 0.0057 seconds.)

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

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ∨ Filter rows: Search this table

+ Options
ACT_NAME
RAHUL

iv. 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.

```
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;
```

```
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
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ∨   Filter rows: Search this table

+ Options

MOV_TITLE ▲ 1	MAX(REV_STARS)
AAKASHAM	2
HOME	3
KALIYONA	5
MANASU	4
WAR HORSE	4

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

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'));

Show query box

✔ 1 row affected. (Query took 0.0105 seconds.)

```
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'))
```

[Edit inline] [Edit] [Create PHP code]