PROGRAM 9

Consider the schema for Movie Database:

  ACTOR (Act_id, Act_Name, Act_Gender)

  DIRECTOR (Dir_id, Dir_Name, Dir_Phone)

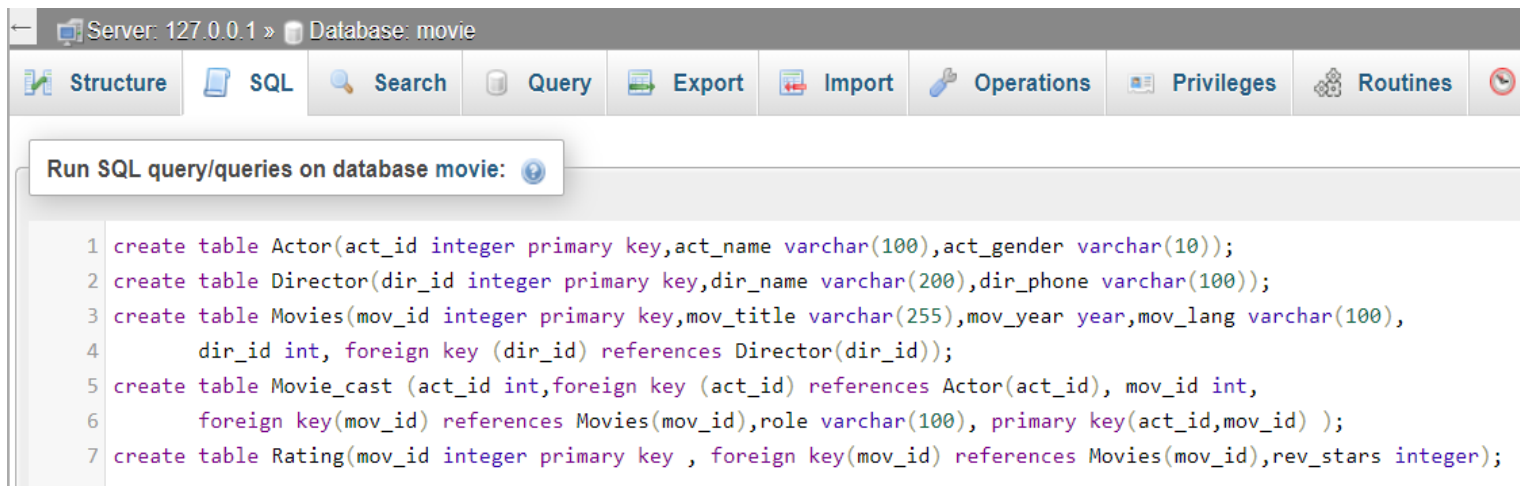 MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

  MOVIE_CAST (Act_id, Mov_id, Role)

  RATING (Mov_id, Rev_Stars)
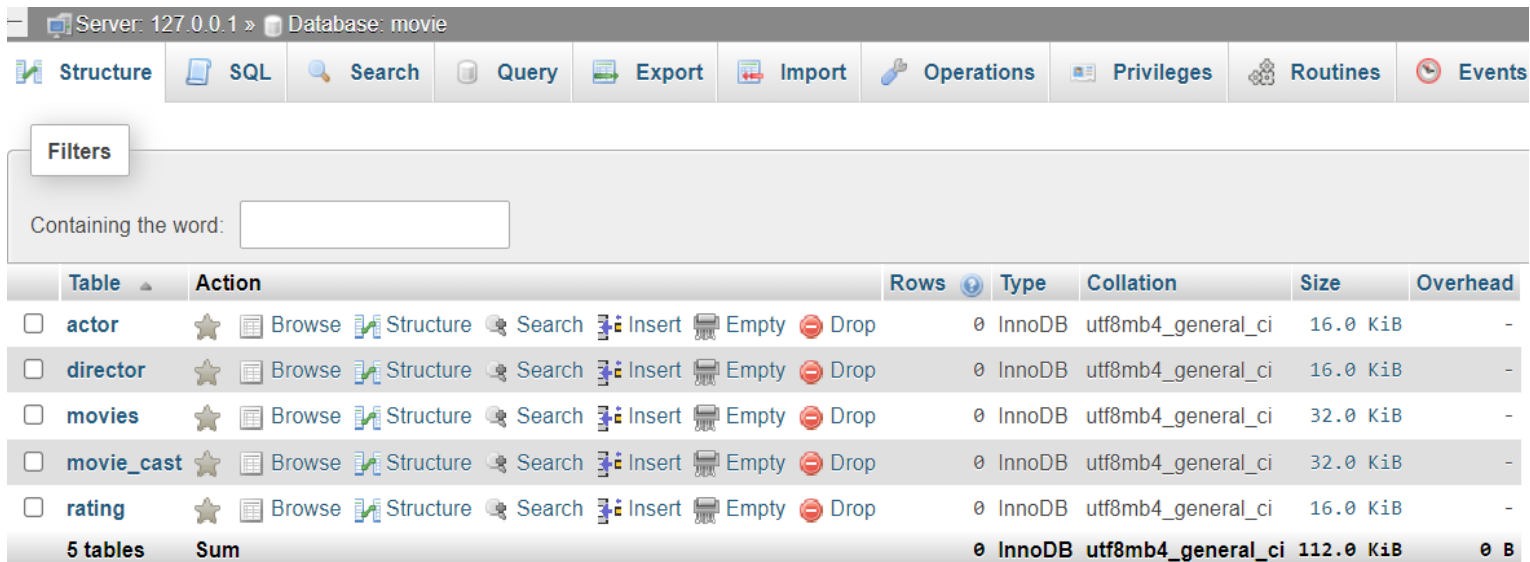
Write SQL queries to

Creation of tables:

Server: 127.0.0.1 » Database: movie

| Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Routines |

Run SQL query/queries on database movie:

```
1  create table Actor(act_id integer primary key,act_name varchar(100),act_gender varchar(10));
2  create table Director(dir_id integer primary key,dir_name varchar(200),dir_phone varchar(100));
3  create table Movies(mov_id integer primary key,mov_title varchar(255),mov_year year,mov_lang varchar(100),
4      dir_id int, foreign key (dir_id) references Director(dir_id));
5  create table Movie_cast (act_id int,foreign key (act_id) references Actor(act_id), mov_id int,
6      foreign key(mov_id) references Movies(mov_id),role varchar(100), primary key(act_id,mov_id) );
7  create table Rating(mov_id integer primary key , foreign key(mov_id) references Movies(mov_id),rev_stars integer);
```

Server: 127.0.0.1 » Database: movie

| Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Routines | Events |

Filters

Containing the word:

Table ▲	Action						Rows	Type	Collation	Size	Overhead
☐ actor	⭐	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐ director	⭐	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐ movies	⭐	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐ movie_cast	⭐	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐ rating	⭐	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
5 tables	Sum						0	InnoDB	utf8mb4_general_ci	112.0 KiB	0 B

Insertion into tables:

'ACTOR' table:

Run SQL query/queries on table movie.actor:

```sql
1  insert into Actor values (1001, 'Tom Crusie','M');
2  insert into Actor values (1002, 'Chris Hemsworth','M');
3  insert into Actor values (1003, 'Angelina Jolie','F');
4  insert into Actor values (1004, 'Margot Robbie','F');
5  insert into Actor values (1005, 'Kate Winslet','F');
6  insert into Actor values (1006, 'Robert Downey','M');
```

✔ Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

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

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

+ Options

	act_id	act_name	act_gender
☐ Edit Copy Delete	1001	Tom Crusie	M
☐ Edit Copy Delete	1002	Chris Hemsworth	M
☐ Edit Copy Delete	1003	Angelina Jolie	F
☐ Edit Copy Delete	1004	Margot Robbie	F
☐ Edit Copy Delete	1005	Kate Winslet	F
☐ Edit Copy Delete	1006	Robert Downey	M

'DIRECTOR' table:

Browse   Structure   SQL   Search   Insert   Export

Run SQL query/queries on table movie.director:

```
1 insert into Director values (9001, 'Hitchcock',9874562154);
2 insert into Director values (9002, 'Steven Spielberg',9874560054);
3 insert into Director values (9003, 'Joseph Levitan',9874562178);
4 insert into Director values (9004, 'Christopher Loyd',9874564454);
5 insert into Director values (9005, 'Yash Chopra',9874562994);
6 insert into Director values (9006, 'Tom Jones',9874503154);
```

Browse   Structure   SQL   Search   Insert   Export   Import

✔ Showing rows 0 - 5 (6 total, Query took 0.0005 seconds.)

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

☐ Show all | Number of rows: 25 ⌄   Filter rows: Search this table   Sort by

+ Options

	dir_id	dir_name	dir_phone
☐ ✏ Edit ⌄ Copy ⊖ Delete	9001	Hitchcock	9874562154
☐ ✏ Edit ⌄ Copy ⊖ Delete	9002	Steven Spielberg	9874560054
☐ ✏ Edit ⌄ Copy ⊖ Delete	9003	Joseph Levitan	9874562178
☐ ✏ Edit ⌄ Copy ⊖ Delete	9004	Christopher Loyd	9874564454
☐ ✏ Edit ⌄ Copy ⊖ Delete	9005	Yash Chopra	9874562994
☐ ✏ Edit ⌄ Copy ⊖ Delete	9006	Tom Jones	9874503154

'MOVIES' table:

Browse   Structure   SQL   Search   Insert   Export   In

Run SQL query/queries on table movie.movies:

```
1  insert into Movies values (101,'Iron Man',2014,'English',9001);
2  insert into Movies values (102,'Prosperity',2001,'Spanish',9001);
3  insert into Movies values (103,'Spiderman',1998,'English',9002);
4  insert into Movies values (104,'Star Wars',1999,'English',9003);
5  insert into Movies values (105,'Thor',2017,'English',9002);
6  insert into Movies values (106,'Captain America',1994,'English',9004);
```

Browse   Structure   SQL   Search   Insert   Export   Import   Priv

Showing rows 0 - 5 (6 total, Query took 0.0005 seconds.)

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

Show all | Number of rows: 25 ⌄   Filter rows: Search this table   Sort by key: Nor

+ Options

	mov_id	mov_title	mov_year	mov_lang	dir_id
☐ 🖉 Edit ᴤ Copy ⊝ Delete	101	Iron Man	2014	English	9001
☐ 🖉 Edit ᴤ Copy ⊝ Delete	102	Prosperity	2001	Spanish	9001
☐ 🖉 Edit ᴤ Copy ⊝ Delete	103	Spiderman	1998	English	9002
☐ 🖉 Edit ᴤ Copy ⊝ Delete	104	Star Wars	1999	English	9003
☐ 🖉 Edit ᴤ Copy ⊝ Delete	105	Thor	2017	English	9002
☐ 🖉 Edit ᴤ Copy ⊝ Delete	106	Captain America	1994	English	9004

'MOVIE CAST' table:

Server: 127.0.0.1 » Database: movie » Table: movie_cast

| Browse | Structure | SQL | Search | Insert |

Run SQL query/queries on table movie.movie_cast: ❓

```
1  insert into Movie_cast values (1001,101,'Joey');
2  insert into Movie_cast values (1001,102,'Conor');
3  insert into Movie_cast values (1002,102,'Tim');
4  insert into Movie_cast values (1003,103,'Kate');
5  insert into Movie_cast values (1004,104,'Claire');
6  insert into Movie_cast values (1006,105,'Sally');
7  insert into Movie_cast values (1005,106,'Jo');
8  insert into Movie_cast values (1002,106,'Craft');
9  insert into Movie_cast values (1002,104,'Josh');
10 insert into Movie_cast values (1005,105,'Roy');
```

Server: 127.0.0.1 » Database: movie » Table: movie_cast

| Browse | Structure | SQL | Search | Insert |

✔ Showing rows 0 - 9 (10 total, Query took 0.0005 seconds.)

SELECT * FROM `movie_cast`

☐ Show all | Number of rows: 25 ⌄   Filter rows: Search this ta

+ Options

			act_id	mov_id	role
☐	Edit Copy Delete		1001	101	Joey
☐	Edit Copy Delete		1001	102	Conor
☐	Edit Copy Delete		1002	102	Tim
☐	Edit Copy Delete		1002	104	Josh
☐	Edit Copy Delete		1002	106	Craft
☐	Edit Copy Delete		1003	103	Kate
☐	Edit Copy Delete		1004	104	Claire
☐	Edit Copy Delete		1005	105	Roy
☐	Edit Copy Delete		1005	106	Jo
☐	Edit Copy Delete		1006	105	Sally

'RATING' table:

Run SQL query/queries on table movie.rating:

```
1  insert into Rating values (101,4);
2  insert into Rating values (102,3);
3  insert into Rating values (103,5);
4  insert into Rating values (104,2);
5  insert into Rating values (105,4);
6  insert into Rating values (106,3);
```

✔ Showing rows 0 - 5 (6 total, Query took 0.0005 seconds.)

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

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

+ Options

	mov_id	rev_stars
☐ 🖉 Edit ⧉ Copy ⊖ Delete	101	4
☐ 🖉 Edit ⧉ Copy ⊖ Delete	102	3
☐ 🖉 Edit ⧉ Copy ⊖ Delete	103	5
☐ 🖉 Edit ⧉ Copy ⊖ Delete	104	2
☐ 🖉 Edit ⧉ Copy ⊖ Delete	105	4
☐ 🖉 Edit ⧉ Copy ⊖ Delete	106	3

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

Query:

select mov_title from Movies where dir_id in (select dir_id from Director where dir_name='Hitchcock');

Run SQL query/queries on database movie:

```
1  select mov_title
2  from Movies
3  where dir_id in (select dir_id
4                   from Director
5                   where dir_name='Hitchcock');
```

Show query box

✔ Showing rows 0 - 1 (2 total, Query took 0.0013 seconds.)

select mov_title from Movies where dir_id in (select dir_id from Director where dir

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

+ Options

←T→	mov_title
☐ Edit Copy Delete	Iron Man
☐ Edit Copy Delete	Prosperity

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

Query:

select distinct m.mov_title,c.act_id from Movies m, Movie_cast c where m.mov_id=c.mov_id and c.act_id in (select act_id from Movie_cast group by act_id having count(mov_id)>1);

Server: 127.0.0.1 » Database: movie

Structure	SQL	Search	Query	Export	
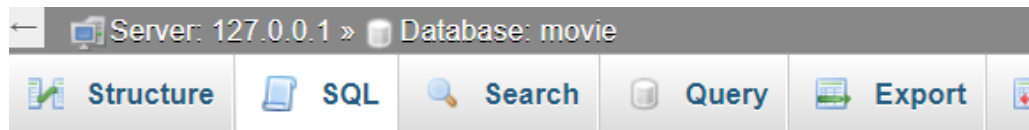
Run SQL query/queries on database movie:

```
1  select distinct m.mov_title,c.act_id
2  from Movies m, Movie_cast c
3  where m.mov_id=c.mov_id
4  and c.act_id in (select act_id
5                   from Movie_cast
6                   group by act_id
7                   having count(mov_id)>1);
```

Server: 127.0.0.1 » Database: movie

Structure	SQL	Search	Query	Export	Imp
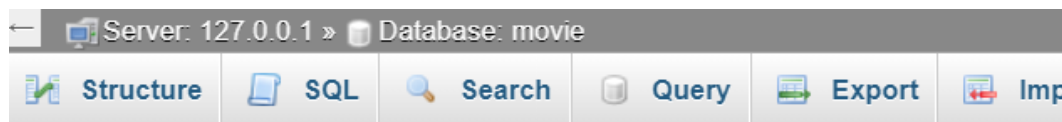
Show query box

✔ Showing rows 0 - 6 (7 total, Query took 0.0025 seconds.)

select distinct m.mov_title,c.act_id from Movies m, Movie_cast c whe count(mov_id)>1)

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

+ Options

mov_title	act_id
Iron Man	1001
Prosperity	1001
Prosperity	1002
Star Wars	1002
Thor	1005
Captain America	1002
Captain America	1005

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

Query:

select act_name from Actor where act_id in(select a.act_id from (select act_id from Movie_cast natural join Movies where mov_year<2000)a inner join (select act_id from Movie_cast natural join Movies where mov_year>2015)b on a.act_id=b.act_id);
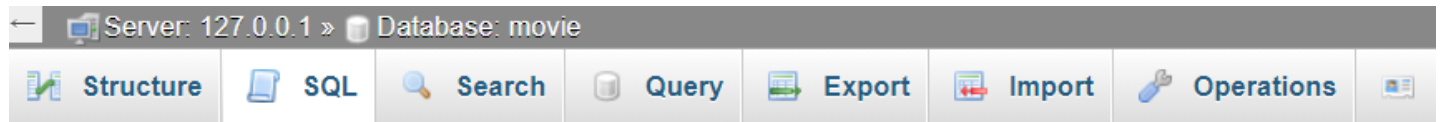
Run SQL query/queries on database movie:

```
 1  select act_name
 2  from Actor
 3  where act_id in (select a.act_id
 4                   from (select act_id
 5                         from Movie_cast
 6                         natural join Movies
 7                         where mov_year<2000)a inner join (select act_id
 8                                                           from Movie_cast
 9                                                           natural join Movies
10                                                           where mov_year>2015)b
11                   on a.act_id=b.act_id);
```

Show query box

✔ Showing rows 0 - 0 (1 total, Query took 0.1107 seconds.)

select act_name from Actor where act_id in (select a.act_id from (select act_id f Movie_cast natural join Movies where mov_year>2015)b on a.act_id=b.act_id)

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

+ Options

←T→   **act_name**

☐ 🖉 Edit 📋 Copy ⊖ Delete   Kate Winslet

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.

Query:

select mov_title,max(rev_stars) from Movies m,Rating r where m.mov_id=r.mov_id group by m.mov_title having count(r.rev_stars)>0;

Server: 127.0.0.1 » Database: movie

Structure | SQL | Search | Query | Export

Run SQL query/queries on database movie:

```
1 select mov_title, max(rev_stars)
2 from Movies m,Rating r
3 where m.mov_id=r.mov_id
4 group by m.mov_title
5 having count(r.rev_stars)>0;
```

Server: 127.0.0.1 » Database: movie

Structure | SQL | Search | Query | Export | Import

Show query box

✓ Showing rows 0 - 5 (6 total, Query took 0.1087 seconds.)

select mov_title, max(rev_stars) from Movies m,Rating r where m.mov_id=

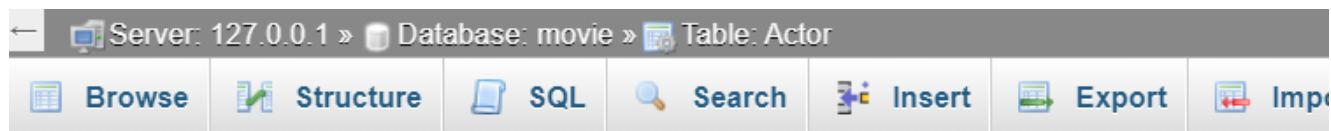☐ Show all | Number of rows: 25 ✓   Filter rows: Search this table

+ Options

mov_title	max(rev_stars)
Captain America	3
Iron Man	4
Prosperity	3
Spiderman	5
Star Wars	2
Thor	4

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

Query:

update Rating set rev_stars=5 where mov_id in (select mov_id from Movies inner join Director on Movies.dir_id=Director.dir_id and Director.dir_name='Steven Spielberg');

Server: 127.0.0.1 » Database: movie

Structure · SQL · Search · Query · Export · Im

Run SQL query/queries on database movie:

```
1  update Rating
2  set rev_stars=5
3  where mov_id in (select mov_id
4                   from Movies
5                   inner join Director
6                   on Movies.dir_id=Director.dir_id
7                   and Director.dir_name='Steven Spielberg');
```

Server: 127.0.0.1 » Database: movie » Table: rating

Browse · Structure · SQL · Search · Insert

✔ Showing rows 0 - 5 (6 total, Query took 0.0005 seconds.)

SELECT * FROM `rating`

☐ Show all | Number of rows: 25 ✔   Filter rows: Search

+ Options

			mov_id	rev_stars
☐	Edit	Copy	Delete	101	4
☐	Edit	Copy	Delete	102	3
☐	Edit	Copy	Delete	103	5
☐	Edit	Copy	Delete	104	2
☐	Edit	Copy	Delete	105	5
☐	Edit	Copy	Delete	106	3