

## SQL Homework 2

Consider the following relational database from IMDB.

- ACTOR (id, fname, lname, gender)
- MOVIE (id, name, overview, release\_year, budget, rating)
- DIRECTOR (id, fname, lname, debut\_year)
- CAST (actor id, movie id, role)
- MOVIE\_DIRECTOR (director id, movie id)

**Q1 [10 pt]** List first name, last name, and gender of all the actors who played in the movie 'Officer 444' (1926)

**Solution:**

**SQL Query:**

```
select actor.fname,actor.lname,actor.gender from actor inner join cast on actor.id = cast.actor_id
inner join movie on movie.id = cast.movie_id where movie.name = 'Officer 444' and release_year
='1926';
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL editor contains the query: `select actor.fname,actor.lname,actor.gender from actor inner join cast on actor.id = cast.actor_id inner join movie on movie.id = cast.movie_id where movie.name = 'Officer 444' and release_year = '1926';`. The output is displayed in a table with columns: frame, lname, gender. The result shows one row: Peter, Pan, Male.

frame	lname	gender
Peter	Pan	Male

The bottom panel shows the 'Action Output' tab with a log of queries and their execution times. The final query (9) is the one executed, returning 1 row(s).

**Q2 [10 pt]** List all actor(s) in descending order of the number of films he/she played, the results should include their first name, last name, and number of movies played by this actor(s).

**Solution:**

**SQL Query:**

```
select actor.fname,actor.lname,count(cast.actor_id) as no_of_movies from actor
inner join cast on actor.id = cast.actor_id group by actor.fname,actor.lname order by no_of_movies
desc;
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • select actor.fname,actor.lname,count(cast.actor_id) as no_of_movies from actor
2 inner join cast on actor.id = cast.actor_id group by actor.fname,actor.lname order by no_of_movies des
3
```

The Results window displays the output of the query in a table format:

fname	lname	no_of_movies
Nicole	Stepe	7
Thomas	Cooke	6
Alexa	Stewart	5
Edward	Cooke	5
Miya	Greene	4
Alya	Williams	4
James	Hoog	3
Mc	Donald	3
Nail	Krite	2
Pit	Alex	2
Mc	Lyon	2
Peter	Pan	2
Jack	Sparrow	2
Tom	Ford	1
Wall		1

The Output window at the bottom shows the following message:

```
6 15:31:36 select movie.release_year,count(*) as female_only_movies_count from movie in... Error Code: 1054. Unknown column 'movie2.id' in 'field list' 0.000 sec
```

**Q3 [10 pt]** Find the film(s) with the smallest cast, return the movie name(s) and the size of the cast.

**Solution:**

**SQL Query:**

```
select movie.name,count(cast.actor_id) as cast_size from movie left join cast on movie.id=cast.movie_id
group by movie.name having cast_size
= (select min(cast_size) from (select count(actor_id) as cast_size from cast group by movie_id) as
subquery);
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 • select movie.name,count(cast.actor_id) as cast_size from movie left join cast on movie.id=cast.movie_id
2 = (select min(cast_size) from (select count(actor_id) as cast_size from cast group by movie_id) as sub
3
```

The Results window displays the following data:

name	cast_size
The 100	1
Legally Blonde	1
Officer 444	1
ZNMD	1
KGHH	1
KANK	1
KXKG	1
KXGH	1

The Output window shows the following message:

```
# Time Action Message Duration / Fetch
10 15:34:52 select actor.fname,actor.lname,count(cast.actor_id) as no_of_movies from acto... 15 row(s) returned 0.000 sec / 0.000 sec
```

The status bar at the bottom indicates "Query Completed".

**Q4 [10 pt]** Find the actor(s) who always have the same role in the movie he/she appeared.

**Solution:**

**SQL Query:**

```
select distinct actor.fname,actor.lname from actor inner join cast on actor.id=cast.actor_id inner join
movie on cast.movie_id=movie.id where not exists(select 1 from cast as cast2 where
cast2.actor_id=actor.id and cast2.movie_id <> cast.movie_id and cast2.char_role <> cast.char_role);
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • select distinct actor.fname,actor.lname from actor inner join cast on actor.id=cast.actor_id
2 inner join movie on cast.movie_id=movie.id where not exists(select 1 from cast as cast2
3 where cast2.actor_id=actor.id and cast2.movie_id <> cast.movie_id and
4 cast2.char_role <> cast.char_role);
```

The Results Grid displays the following data:

fname	lname
Peter	Pan
Jack	Sparrow
Tom	Ford
Jake	Hall
Thomas	Cooke
Edward	Cooke

The Output pane shows the following message:

```
14 15:40:59 select 'from movie LIMIT 0, 1000' 19 row(s) returned 0.000 sec / 0.000 sec
```

The status bar at the bottom indicates "Query Completed".

**Q5 [10 pt]** Find all actor(s) who acted only in films after 2020.

**Solution:**

**SQL Query:**

```
select distinct actor.fname, actor.lname from actor inner join cast on actor.id=cast.actor_id
inner join movie on cast.movie_id = movie.id where actor.id not in(select distinct actor.id
from actor inner join cast on actor.id=cast.actor_id inner join movie on cast.movie_id = movie.id
where movie.release_year<=2020);
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 select distinct actor.fname, actor.lname from actor inner join cast on actor.id=cast.actor_id
2 inner join movie on cast.movie_id = movie.id where actor.id not in(select distinct actor.id
3 from actor inner join cast on actor.id=cast.actor_id inner join movie on cast.movie_id = movie.id
4 where movie.release_year<=2020);
5
```

The Results window displays the following data:

fname	lname
Tom	Ford
Jake	Hall

The Output window shows the following message:

```
15 15:41:09 select * from cast LIMIT 0, 1000 49 row(s) returned 0.000 sec / 0.000 sec
```

The status bar at the bottom indicates "Query Completed".

**Q6[10 pt]** For every director, list the films he/she directed in his/her debut year. Sort the results by first name of the director.

**Solution:**

**SQL Query:**

```
select distinct director.fname from director inner join movie_director on director.id =  
movie_director.director_id inner join movie on movie_director.movie_id = movie.id where  
movie.release_year=director.debut_year order by director.fname;
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 • select distinct director.fname from director inner join movie_director on  
2 director.id = movie_director.director_id inner join movie on movie_director.movie_id = movie.id  
3 where movie.release_year=director.debut_year order by director.fname;  
4
```

The Results window displays the output of the query:

fname
Mike
Steve

The bottom status bar indicates "Query Completed".

**Q7[10 pt]** Output a list of all the actors ordered by the average ratings of their movies.

**Solution:**

**SQL Query:**

```
select actor.fname,actor.lname from actor inner join cast on actor.id=cast.actor_id inner join movie on  
cast.movie_id=movie.id group by actor.id  
order by AVG(movie.rating) desc;
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 select actor.fname,actor.lname from actor inner join cast on actor.id=cast.actor_id  
2 inner join movie on cast.movie_id=movie.id group by actor.id  
3 order by AVG(movie.rating) desc;  
4  
5
```

The Results Grid displays the following data:

fname	lname
Jake	Hall
Miya	Greene
Tom	Ford
James	Hoog
Nail	Krite
Thomas	Cooke
Nicole	Steppe
Mc	Donald
Pit	Alex
Mc	Lyon
Alexa	Stewart
Jack	Sparrow
Edward	Cooke
Alva	Williams
Dakota	Dan

The Output pane at the bottom shows the following message:

```
18 15:45:42 select distinct director.fname from director inner join movie_director on director... 2 row(s) returned  
Duration / Fetch 0.000 sec / 0.000 sec
```

Query Completed

**Q8[10 pt]** find out who is the most bankable director (has the highest budget sum for all the directed movies)

**Solution:**

**SQL Query:**

```
select director.fname, director.lname from director inner join movie_director on
director.id = movie_director.director_id inner join movie on movie_director.movie_id = movie.id
group by director.fname, director.lname order by sum(movie.budget) desc limit 1;
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
1 select director.fname, director.lname from director inner join movie_director on
2 director.id = movie_director.director_id inner join movie on movie_director.movie_id = movie.id
3 group by director.fname, director.lname order by sum(movie.budget) desc limit 1;
4
5
```

The Results window displays the output of the query:

fname	lname
James	Cullen

The bottom status bar indicates "Query Completed".



**Q9[10 pt]** Find all the actors who acted in films by at least 5 distinct directors (i.e. actors who worked with at least 5 distinct directors).

**Solution:**

**SQL Query:**

```
select actor.fname,actor.lname from actor inner join cast on cast.actor_id=actor.id inner join movie on  
cast.movie_id =movie.id inner join movie_director on movie_director.movie_id=movie.id group by  
actor.fname,actor.lname having count(distinct movie_director.director_id) >= 5;
```

**Output:**

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the 'imdb' database, including tables like 'actor', 'cast', 'director', 'movie', and 'movie\_director'. The main editor window contains a SQL query:   
1 • select actor.fname,actor.lname from actor inner join cast on cast.actor\_id=actor.id inner join  
2 movie on cast.movie\_id = movie.id inner join movie\_director on movie\_director.movie\_id=movie.id group  
The 'Result Grid' tab is active, showing a table with two columns: 'fname' and 'lname'. The first row of data is 'Edward' and 'Cooke'. The bottom status bar indicates 'Query Completed' and '1 row(s) returned'.

fname	lname
Edward	Cooke

**Q10 [10 pt]** For each year, count the number of movies in that year that had only female actors.

**Solution:**

**SQL Query:**

```
select movie.release_year,count(*) as female_only_movies_count from movie inner join cast on
movie.id = cast.movie_id left join actor on cast.actor_id = actor.id where actor.gender = 'female' and
movie.id not in (select distinct movie.id from movie inner join cast on movie.id = cast.movie_id inner
join actor on cast.actor_id = actor.id where actor.gender='male')group by movie.release_year
order by movie.release_year;
```

**Output:**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query from the previous block. The Results window shows the output of the query, which is a single row for the year 2011 with a count of 3. The bottom status bar indicates that the query was executed successfully, returning 1 row(s) in 0.000 seconds.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

imdb

- Tables
  - actor
  - cast
  - director
  - movie
  - movie\_director
- Views
- Stored Procedures
- Functions
- lab\_university
- sales
- sys

Administration Schemas

Information

No object selected

SQL Editor

Limit to 1000 rows

```
29 • select actor.fname,actor.lname from actor inner join cast on cast.actor_id=actor.id inner join movie
30 cast.movie_id=movie.id inner join movie_director on movie_director.movie_id=movie.id group by actor.
31 count(distinct movie_director.director_id) >= 5;
32
33 • select movie.release_year,count(*) as female_only_movies_count from movie
34 inner join cast on movie.id = cast.movie_id left join actor on cast.actor_id = actor.id
35 where actor.gender = 'female' and movie.id not in (select distinct movie.id from movie inner join
36 cast on movie.id = cast.movie_id inner join actor on cast.actor_id = actor.id where actor.gender='male')
37 order by movie.release_year;
```

Result Grid

release_year	female_only_movies_count
2011	3

Result 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:01:47	select movie.release_year,count(*) as female_only_movies_count from movie inn...	1 row(s) returned	0.000 sec / 0.000 sec