User:A3nm/SD202 2021 ex2

< User:A3nm

In this second exercise, we will query three tables: `movie`, `casting`, and `actor`.

The `movie` table describes movies, and contains the following fields:

- `id int(11)`, an identifier
- `title varchar(50)`, the title of the movie
- `yr int(11)`, the year the movie was released
- `director int(11)`, the identifier of the director
- `budget int(11)`, the movie's budget (we will not use it)
- `gross int(11)`, the movie's gross revenue (we will not use it)

The `actor` table contains people (actors and directors) and contains the following fields:

- `id int(11)`, an identifier
- `name varchar(50)`, the name of the person

The `casting` describes actors starring in movies. It contains the following fields:

- `movieid int(11)`, the identifier of the movie in the `movie` table
- `actorid int(11)`, the identifier of the actor in the `actor` table
- `ord int(11)`, an integer describing the position of the actor in the film's starring list. The first actor (called the leading actor) has `ord` value 1, the second actor has `ord` value 2, and so on.

For instance, here are the first 5 movies of the `movie` table:

id	title	yr	director	budget	gross
10001	$	1971	3		

Cheat mode

10002	"Crocodile" Dundee	1986	19		328203506
10003	"Crocodile" Dundee II	1988	36	15800000	239606210
10004	'Til There Was You	1997	49	10000000	
10005	'Til We Meet Again	1940	65		

Here are the `casting` rows corresponding to movie `10001`:

movieid	actorid	ord
10001	4	1
10001	5	2
10001	6	3
10001	7	4
10001	8	5
10001	9	6
10001	10	7
10001	11	8
10001	12	9
10001	13	10
10001	14	11
10001	15	12
10001	16	13
10001	17	14
10001	18	15

And here are the corresponding tuples in the `actor` table. Note that the name of the actor with id `6` has an encoding issue -- we will ignore these issues in the sequel.

Cheat mode

id	name

4	Warren Beatty
5	Goldie Hawn
6	Gert FrÃ¶be
7	Robert Webber
8	Scott Brady
9	Arthur Brauss
10	Robert Stiles
11	Wolfgang Kieling
12	Bob Herron
13	Christiane Maybach
14	Hans Hutter
15	Monica Stender
16	Horst Hesslein
17	Wolfgang Kuhlman
18	Klaus Schichan

Warning: Again, the table names are case-sensitive and must be written in lowercase.

Warning: The queries in this exercise may be slow to evaluate, which may cause the server to timeout. If you get the error "Lost connection to MySQL server during query", then please simply try again evaluating the query. If the problem persists, your query may not be efficient enough and may need to be optimized. To debug it and get partial results, try adding some additional selection conditions and/or a `LIMIT`.

Basic joins

1.

Compute the title of every Star Wars movie (starting with "Star Wars") and the name of its director. Sort the result by title.

Cheat mode

```
SELECT title,name FROM (SELECT title, director FROM movie WHERE title LIKE 'Star Wars%' ) as movie1 JOIN actor on movie1.director = actor.id ORDER BY title
```

Submit SQL

Restore default

result

2.

Cheat mode

Compute the list of the names of the actors starring in the movie "Jurassic Park" (1993), in the order in which they starred (i.e., by increasing `ord` value).

```
SELECT name FROM (SELECT id,title from movie WHERE title = 'Jurassic Park') AS table1
JOIN
casting
ON table1.id = casting.movieid
JOIN
actor ON casting.actorid = actor.id
```

Submit SQL

Restore default

result

Cheat mode

3.

Compute the list of the titles of the movies where "George Clooney" appeared, ordered by title.

```
SELECT title FROM (SELECT * FROM  actor WHERE name = 'George Clooney') AS table1 JOIN casting ON table1.id = casting.actorid JOIN movie ON movie.id = casting.movieid
```

Submit SQL

Restore default

result

Cheat mode

4.

Compute the list of the titles of all 1920 movies together with the name of their leading actor (the one with `ord` value of 1), ordered by title.

```
SELECT title,name FROM (SELECT id,title FROM movie WHERE yr = 1920) AS table1
LEFT JOIN (SELECT movieid,actorid FROM casting WHERE ord = 1) AS table2 ON table1.id = table2.movieid
LEFT JOIN actor ON table2.actorid = actor.id ORDER BY title
```

Submit SQL

Restore default

result

Cheat mode

Advanced joins

5.

Compute the list of the titles of all 1920 movies together with the name of their leading actor and with the name of their director, ordered by title.

```
SELECT title,name,director_name FROM (SELECT id,title,director FROM movie WHERE yr = 1920) AS table1
LEFT JOIN (SELECT movieid,actorid FROM casting WHERE ord = 1) AS table2 ON table1.id = table2.movieid
LEFT JOIN actor ON table2.actorid = actor.id
LEFT JOIN (SELECT id,name AS director_name FROM actor) as table3 ON table3.id = table1.director ORDER BY title
```

Submit SQL

Restore default

result

Cheat mode

6.

Compute the five movies in the database having the highest number of participating actors, and this number of participating actors, sorted by decreasing number of actors. **Warning:** beware of titles like "The Hunchback of Notre Dame" that are the titles of multiple movies!

```
SELECT title,COUNT(id) as number FROM (SELECT id,title FROM movie) AS table1
LEFT JOIN (SELECT movieid FROM casting) AS table2 ON table1.id = table2.movieid GROUP BY id ORDER BY COUNT(id) DESC LIMIT 5
```

Submit SQL

Restore default

result

Cheat mode

7.

Compute the years where the actor "Rock Hudson" participated to strictly more than one movie, along with the number of movies to which he participated on that year, sorted by decreasing number of movies, then by ascending year.

```
SELECT yr,COUNT(table3.id) FROM (SELECT id,name FROM actor WHERE name = 'Rock Hudson') AS table1
LEFT JOIN (SELECT movieid, actorid FROM casting) AS table2 ON table1.id = table2.actorid
LEFT JOIN (SELECT yr,id FROM movie) as table3 ON table3.id = table2.movieid GROUP BY yr HAVING COUNT(table3.id)>1 ORDER BY COUNT(table3.id) DESC, yr
ASC
```

Submit SQL

Restore default

result

Cheat mode

8.

Compute the names of actors who were the leading actor in a movie where Harrison Ford appeared (and were not Harrison Ford himself). Order the results by actor name.

```
SELECT * FROM
(SELECT id FROM actor WHERE name = 'Harrison Ford') AS table1
JOIN
(SELECT movieid,actorid FROM casting) as table2 ON table1.id = table2.actorid
```

Submit SQL

Restore default

result

Cheat mode

9.

Compute the titles of movies which were both directed by Woody Allen and had Woody Allen appear as an actor, sorted in alphabetical order.

```
SELECT movieid,actorid FROM casting WHERE actorid = 1
```

Submit SQL

Restore default

result

Cheat mode

10.

Compute the titles of movies which were directed by Woody Allen or had Woody Allen appear as an actor (or both), sorted in alphabetical order.

```
SELECT title FROM movie, actor, casting WHERE casting.movieid = movie.id AND
casting.actorid = actor.id AND actor.name = 'Woody Allen'
UNION
SELECT title FROM movie, actor WHERE movie.director = actor.id AND actor.name =
'Woody Allen'
ORDER BY title
```

Submit SQL

Restore default

result

Cheat mode

11.

In which movie title did Alain Delon and Catherine Deneuve appear together?

```
SELECT title FROM movie,casting AS C1, casting AS C2, actor AS A1, actor AS A2 WHERE A1.id = C1.actorid AND A2.id = C2.actorid AND C1.movieid =
movie.id AND C2.movieid = movie.id AND A1.name = 'Alain Delon' AND A2.name = 'Catherine Deneuve'
```

Submit SQL

Restore default

result

Cheat mode

12.

Find the only actor who appeared in all Star Wars movies. (Note: this query is complex; if you get an error about losing connection to the server, please try again.)

```
SELECT name FROM actor WHERE NOT EXISTS (
  SELECT id FROM movie WHERE TITLE LIKE 'Star Wars%' AND NOT EXISTS (
    SELECT * FROM casting WHERE movieid=movie.id AND actorid=actor.id))
```

Submit SQL

Restore default

result

Cheat mode

13.

Find the only actor which only appeared in movies where Harrison Ford appeared, and appeared in strictly more than one movie.

```
SELECT name FROM actor, casting, movie WHERE casting.actorid = actor.id AND casting.movieid = movie.id AND name <> 'Harrison Ford' AND NOT EXISTS (
  SELECT id FROM movie, casting AS C1 WHERE C1.movieid = movie.id AND C1.actorid = actor.id AND NOT EXISTS
    (SELECT 1 FROM casting AS C2, actor AS A2 WHERE A2.name = 'Harrison Ford' AND C2.actorid = A2.id AND C2.movieid = C1.movieid)) GROUP BY name
HAVING COUNT(movie.id) > 1
```

Submit SQL

Restore default

result

Cheat mode

14.

For performance reasons, we limit ourselves to the movies released no later than 1930, and to the actor names starting with A, B, or C. We say that two actors X and Y are *challengers* if X was the leading actor in a movie where Y appeared *and* Y was the leading actor in a movie where X appeared. Compute all pairs X, Y of challengers (with X < Y, in alphabetical order).

```
WITH oldmov AS (SELECT id, yr FROM movie WHERE yr <= 1930),
aactor AS (SELECT id, name FROM actor WHERE name LIKE 'A%' OR name LIKE 'B%' OR name LIKE 'C%')
SELECT DISTINCT A1.name, A2.name FROM aactor AS A1, aactor AS A2, casting AS C1a, casting AS C1b, casting AS C2a, casting AS C2b, oldmov AS Ma,
oldmov AS Mb WHERE A1.id = C1a.actorid AND A1.id = C1b.actorid AND A2.id = C2a.actorid AND A2.id = C2b.actorid AND C1a.movieid = C2a.movieid AND
C1b.movieid = C2b.movieid AND C1a.ord = 1 AND C2b.ord = 1 AND C1a.movieid = Ma.id AND C1b.movieid = Mb.id AND A1.name < A2.name
```

Submit SQL

Restore default

result

Cheat mode

Acknowledgements

This exercise is inspired by the exercises More_JOIN_operations and by R. Ramakrishnan, J. Gehrke. Database Management Systems, 3rd ed., 2002. Exercise 4.3. If you reach the end of this exercise and still have time, you can complete the questions in the other JOIN tutorials which were not covered here.

Thanks to Andrew Cumming for his great work in maintaining SQLZoo!
