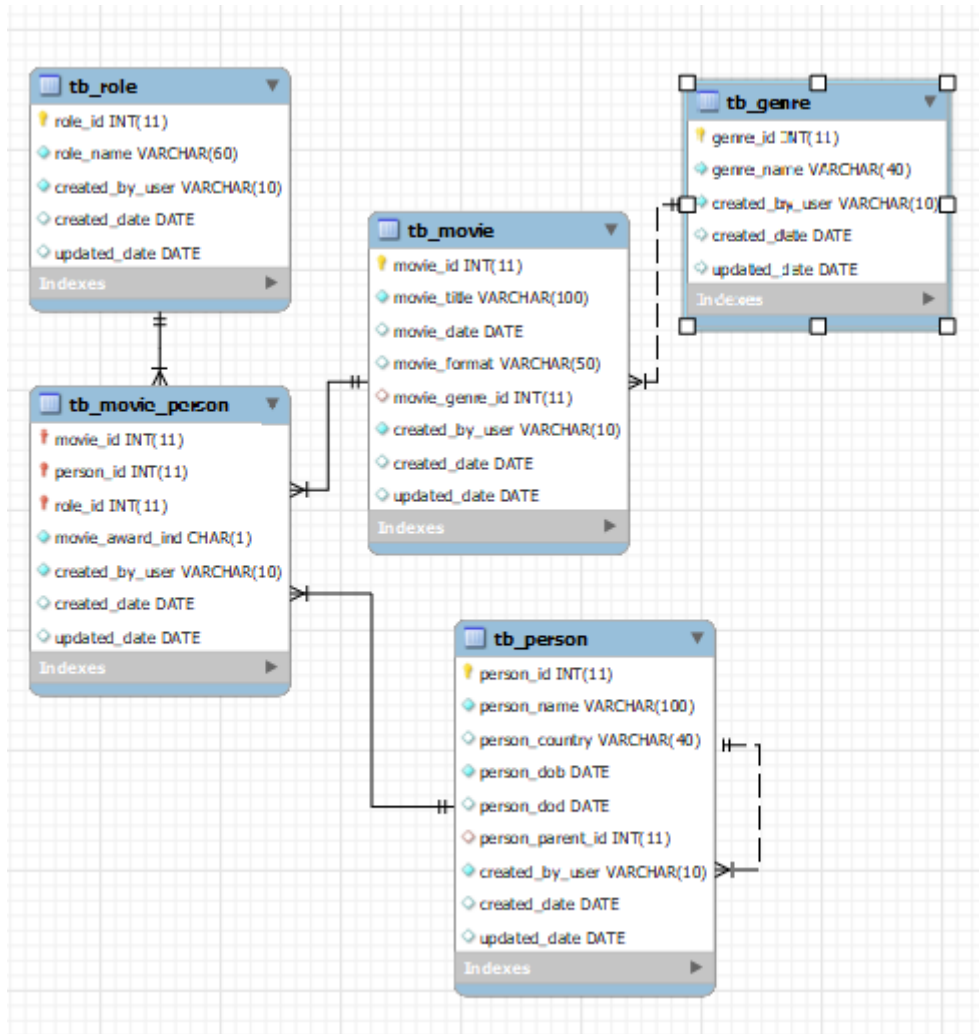


## Exercici 1

A partir dels documents adjunts (estructura i dades), crea una base de dades amb MySQL. Mostra les característiques principals de l'esquema creat i explica les diferents taules i variables que hi ha.



**tb\_role:** Identifica los diferentes roles de cada una de las personas que aparecen en la tabla *tb\_person*. Tiene el atributo *role\_id* (primary key) que identifica cada uno de los roles y el atributo *role\_name* con el nombre de cada uno de estos roles.

**tb\_movie:** Es un listado de películas, donde cada una de ellas está identificada por la *movie\_id*(primary key). En este listado aparece el título de la película (*movie\_title*), una fecha sin identificar sobre la película(*movie\_date*), el formato de la misma(*movie\_format*) y el género al que pertenece(*movie\_genre*).

**tb\_genre:** Listado de los diferentes géneros, con un *genre\_id*(primary key) y un nombre de género(*genre\_name*).

**tb\_person:** Listado de diferentes personas identificadas por una *person\_id*(primary key), donde aparece el nombre (*person\_name*), país de nacimiento(*person\_country*), fecha de

naixement(*person\_dob*), fecha de fallecimiento si existe(*person\_dod*) y un campo que en el caso de existir parentesco con otra de las personas, nos indica su *person\_parent\_id*

**tb\_movie\_person:** Es la tabla que relaciona a las tablas *tb\_person*, *tb\_movie* y *tb\_role* y además tiene un atributo que nos indica por cada uno de los registros, si esa persona en ese rol y para esa película, ha tenido un premio individual(*movie\_award\_id*).

## Exercici 2

Realitza la següent consulta sobre la base de dades acabada de crear:

Has d'obtenir el nom, el país i la data de naixement d'aquelles persones per les quals no consti una data de mort i ordenar les dades de la persona més vella a la persona més jove.

***SELECT person\_name, person\_country, person\_dob***

***FROM tb\_person***

***WHERE person\_dod IS NULL***

***ORDER BY person\_dod DESC;***

The screenshot shows a database query interface. The query editor at the top contains the following SQL query:

```
1 SELECT person_name, person_country, person_dob
2 FROM tb_person
3 WHERE person_dod IS NULL
4 ORDER BY person_dod DESC;
```

Below the query editor, the 'Result Grid' displays the results of the query. The results are organized into three columns: *person\_name*, *person\_country*, and *person\_dob*. The results are ordered by *person\_dob* in descending order.

person_name	person_country	person_dob
Francis Ford Coppola	United States	1939-04-07
Veronica Echegui	Spain	1983-03-14
Patrick Criado	Spain	1995-09-23
Sean Connery	Scotland	1930-07-08
Mel Gibson	Australia	1950-08-09
Morgan Freeman	United States	1935-10-01
Tim Robbins	United States	1949-06-07
Charlie Sheen	United States	1965-09-03
Emilio Estevez	United States	1962-05-12
Ramón Estevez	United States	1963-08-07
René Estevez	United States	1967-04-02
Paula Speert Sheen	United States	1986-01-06
Bob Sheen	United States	2009-05-01
Max Sheen	United States	2009-05-01
Sam Sheen	United States	2004-03-09
Lola Sheen	United States	2005-06-01
Paula Jones-Sheen	United States	2003-07-06
Paloma Rae Estevez	United States	1986-02-15
Robert Alamo	Spain	1970-05-06
Quim Gubierrez	Spain	1981-03-27
Carmin Coppola	United States	1945-07-08
Robert Duvall	United States	1931-01-05
Martin Sheen	United States	1940-08-03
Harrison Ford	United States	1942-07-13
George Lucas	United States	1944-05-14
Gary Kurtz	United States	1940-07-27

At the bottom of the interface, the 'Output' section shows the execution details of the query:

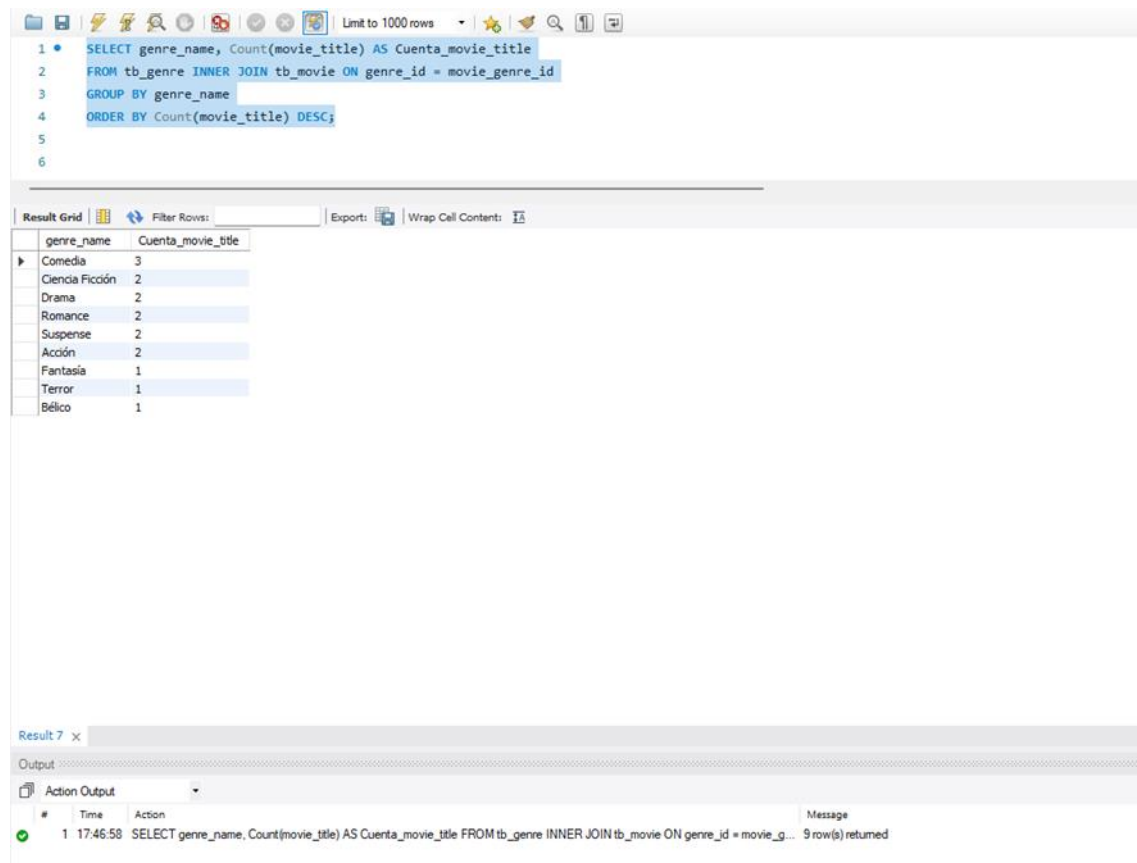
```
1 17:41:46 SELECT person_name, person_country, person_dob FROM tb_person WHERE person_dod IS NULL ORDER BY person_dod ... 38 row(s) returned
```

### Exercici 3

Realitza la següent consulta sobre la base de dades acabada de crear:

Has d'obtenir el nom del gènere i el nombre total de pel·lícules d'aquest gènere i ordenar-ho per ordre descendent de nombre total de pel·lícules.

```
SELECT genre_name, Count(movie_title) AS Cuenta_movie_title  
FROM tb_genre INNER JOIN tb_movie ON genre_id = movie_genre_id  
GROUP BY genre_name  
ORDER BY Count(movie_title) DESC;
```



The screenshot shows a database query editor with the following SQL query:

```
1 • SELECT genre_name, Count(movie_title) AS Cuenta_movie_title
2 FROM tb_genre INNER JOIN tb_movie ON genre_id = movie_genre_id
3 GROUP BY genre_name
4 ORDER BY Count(movie_title) DESC;
5
6
```

Below the query editor, the results are displayed in a table:

genre_name	Cuenta_movie_title
Comedia	3
Ciencia Ficción	2
Drama	2
Romance	2
Suspense	2
Acción	2
Fantasia	1
Terror	1
Bélico	1

At the bottom, the output log shows the execution of the query:

```
Result 7 x
Output
Action Output
# Time Action Message
1 17:46:58 SELECT genre_name, Count(movie_title) AS Cuenta_movie_title FROM tb_genre INNER JOIN tb_movie ON genre_id = movie_g... 9 row(s) returned
```

#### Exercici 4

Realitza la següent consulta sobre la base de dades acabada de crear:

Has d'obtenir, per a cada persona, el seu nom i el nombre màxim de rols diferents que ha tingut en una mateixa pel·lícula.

```
SELECT person_name, movie_title, Count(role_id) AS Cuenta_role_id  
  
FROM tb_movie INNER JOIN (tb_person INNER JOIN tb_movie_person ON  
tb_person.person_id= tb_movie_person.person_id) ON tb_movie.movie_id =  
tb_movie_person.movie_id  
  
GROUP BY person_name, movie_title  
  
ORDER BY Count(role_id) DESC;
```

The screenshot shows a database query editor with the following SQL query:

```
1 SELECT person_name, movie_title, Count(role_id) AS Cuenta_role_id
2 FROM tb_movie INNER JOIN (tb_person INNER JOIN tb_movie_person ON tb_person.person_id = tb_movie_person.person_id) ON tb_movie.movie_id = tb_movie_person.movie_id
3 GROUP BY person_name, movie_title
4 ORDER BY Count(role_id) DESC;
5
```

The results are displayed in a grid with the following columns: person\_name, movie\_title, and Cuenta\_role\_id. The results are sorted by Cuenta\_role\_id in descending order.

person_name	movie_title	Cuenta_role_id
Francis Ford Coppola	Apocalypse Now	3
Alfred Joseph Hitchcock	Psycho	3
Steven Spielberg	The Terminal	2
Daniel Sanchez Arevalo	La Gran Familia Española	2
George Lucas	Indiana Jones and the Temple of Doom	2
Mel Gibson	Braveheart	2
Robert Alamo	La Gran Familia Española	1
Robert Duvall	Apocalypse Now	1
Karra Elejalde	Ocho Apellidos Vascos	1
Steven Spielberg	Jaws	1
Morgan Freeman	The Shawshank Redemption	1
George Lucas	Star Wars:Episode IV - A New Hope	1
Karra Elejalde	Ocho Apellidos Catalanes	1
Emilio Martinez Lazaro	Ocho Apellidos Vascos	1
John Williams	Indiana Jones and the Temple of Doom	1
Hector Colome	La Gran Familia Española	1
Martin Sheen	Apocalypse Now	1
Emilio Martinez Lazaro	Ocho Apellidos Catalanes	1
Steven Spielberg	ET The Extraterrestrial	1
Tim Robbins	The Shawshank Redemption	1
Gary Kurtz	Star Wars:Episode IV - A New Hope	1
Emilio Martinez Lazaro	El otro lado de la cama	1
Dani Rovira	Ocho Apellidos Vascos	1
Veronica Echegui	La Gran Familia Española	1
Harrison Ford	Apocalypse Now	1
Dani Rovira	Ocho Apellidos Catalanes	1
Harrison Ford	Indiana Jones and the Temple of Doom	1

The output section shows the following message:

```
1 17:33:34 SELECT person_name, movie_title, Count(role_id) AS Cuenta_role_id FROM tb_movie INNER JOIN tb_person INNER JOIN tb_... 42 row(s) returned
```

Posteriorment, mostra únicament aquelles persones que hagin assumit més d'un rol en una mateixa pel·lícula.

```
SELECT person_name, movie_title, Count(role_id) AS Cuenta_role_id
```

```
FROM tb_movie INNER JOIN (tb_person INNER JOIN tb_movie_person ON  
tb_person.person_id= tb_movie_person.person_id) ON tb_movie.movie_id =  
tb_movie_person.movie_id
```

```
GROUP BY person_name, movie_title
```

```
HAVING (((Count(role_id))>1))
```

```
ORDER BY Count(role_id) DESC;
```

The screenshot shows a database query editor with the following SQL query:

```
1 SELECT person_name, movie_title, Count(role_id) AS Cuenta_role_id
2 FROM tb_movie INNER JOIN (tb_person INNER JOIN tb_movie_person ON
3 tb_person.person_id= tb_movie_person.person_id) ON tb_movie.movie_id =
4 tb_movie_person.movie_id
5 GROUP BY person_name, movie_title
6 HAVING (((Count(role_id))>1))
7 ORDER BY Count(role_id) DESC;
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has three columns: person\_name, movie\_title, and Cuenta\_role\_id. The results are as follows:

person_name	movie_title	Cuenta_role_id
Alfred Joseph Hitchcock	Psycho	3
Francis Ford Coppola	Apocalypse Now	3
George Lucas	Indiana Jones and the Temple of Doom	2
Mel Gibson	Braveheart	2
Daniel Sanchez Arevalo	La Gran Familia Española	2
Steven Spielberg	The Terminal	2

At the bottom of the screenshot, there is an 'Action Output' section showing the execution of the query. It indicates that the query was executed at 17:37:44 and returned 6 rows.

## Exercici 5

Realitza la següent operació sobre la base de dades acabada de crear:

Has de crear un nou gènere anomenat "Documental" el qual tingui com a identificador el nombre 69.

```
INSERT INTO tb_genre(genre_id, genre_name) VALUES(69, "Documental");
```

```
SELECT * FROM movies.tb_genre;
```

The screenshot shows a database management interface. At the top, there's a toolbar with various icons and a 'Limit to 1000 rows' dropdown. Below the toolbar, two SQL queries are listed:

- 1 • `INSERT INTO tb_genre(genre_id, genre_name) VALUES(69, "Documental");`
- 2 • `SELECT * FROM movies.tb_genre;`

Below the queries, there's a 'Result Grid' section. It contains a table with the following data:

genre_id	genre_name	created_by_user	created_date	updated_date
1	Acción	OS_SGAD	NULL	NULL
2	Ciencia Ficción	OS_SGAD	NULL	NULL
3	Comedia	OS_SGAD	NULL	NULL
4	Drama	OS_SGAD	NULL	NULL
5	Fantasia	apermag	NULL	NULL
6	Melodrama	apermag	2018-09-01	2018-09-27
7	Musical	OS_SGAD	NULL	NULL
8	Romance	OS_SGAD	NULL	NULL
9	Suspense	OS_SGAD	NULL	NULL
10	Terror	OS_SGAD	NULL	NULL
11	Bélico	OS_SGAD	NULL	NULL
69	Documental	OS_SGAD	NULL	NULL
*	NULL	NULL	NULL	NULL

Below the result grid, there's an 'Output' section. It shows the execution of the two queries:

#	Time	Action	Message
1	17:56:31	INSERT INTO tb_genre(genre_id, genre_name) VALUES (69, "Documental")	1 row(s) affected
2	17:56:36	SELECT * FROM movies.tb_genre LIMIT 0, 1000	12 row(s) returned

## Exercici 6

Fes la següent operació sobre la base de dades acabada de crear:

Elimina la pel·lícula "La Gran Familia Española" de la base de dades.

**DELETE FROM tb\_movie WHERE movie\_id =11;**

**SELECT\*FROM tb\_movie;**

```
1 • DELETE FROM tb_movie WHERE movie_id =11;
2 • SELECT*FROM tb_movie;
```

movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_user	created_date	updated_date
1	Apocalypse Now	1979-05-10	Film	11	OS_SGAD	NULL	NULL
2	Star Wars:Episode IV - A New Hope	1977-05-25	Film	2	OS_SGAD	NULL	NULL
3	Indiana Jones and the Temple of Doom	1984-05-08	Film	1	OS_SGAD	NULL	NULL
4	The Terminal	2004-06-18	Digital	3	OS_SGAD	NULL	NULL
5	Jaws	1975-01-01	Film	10	OS_SGAD	NULL	NULL
6	ET The Extraterrestrial	1982-07-25	Film	5	OS_SGAD	NULL	NULL
7	Psycho	1960-05-06	Film	9	OS_SGAD	NULL	NULL
8	Ocho Apellidos Vascos	2014-03-14	Digital	3	OS_SGAD	NULL	NULL
9	Ocho Apellidos Catalanes	2016-06-09	Digital	8	OS_SGAD	NULL	NULL
10	El otro lado de la cama	2002-09-04	Digital	8	OS_SGAD	NULL	NULL
12	El dia de la bestia	1994-12-25	Film	1	OS_SGAD	NULL	NULL
13	Braveheart	1995-08-08	Film	4	OS_SGAD	NULL	NULL
14	The Shawshank Redemption	1992-01-07	Film	4	OS_SGAD	NULL	NULL
15	Las brujas de Zugarramurdi	2009-10-07	Digital	9	OS_SGAD	NULL	NULL
16	Blade Runner	1982-12-25	Digital	2	OS_SGAD	NULL	NULL
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message
✓ 1	18:30:48	DELETE FROM tb_movie WHERE movie_id =11	1 row(s) affected
✓ 2	18:31:14	DELETE FROM tb_movie WHERE movie_id =11	0 row(s) affected
✓ 3	18:31:14	SELECT*FROM tb_movie LIMIT 0, 1000	15 row(s) returned

## Exercici 7

Realitza la següent operació sobre la base de dades acabada de crear:

Canvia el gènere de la pel·lícula "Ocho apellidos catalanes" perquè consti com a comèdia i no com a romàntica.

***UPDATE tb\_movie SET movie\_genre\_id = 3 WHERE movie\_id = 9;***

***SELECT\*FROM tb\_movie WHERE movie\_id=9;***

The screenshot shows a database management interface with the following components:

- SQL Editor:** Contains two queries:
  - 1 • `UPDATE tb_movie SET movie_genre_id = 3 WHERE movie_id = 9;`
  - 2 • `SELECT*FROM tb_movie WHERE movie_id=9;`
- Result Grid:** Displays the results of the second query. The first row shows the movie details for 'Ocho Apellidos Catalanes' with genre\_id 3. The rest of the grid is empty.
- Output Panel:** Shows the execution log with two actions:
  - 1 18:39:20 UPDATE tb\_movie SET movie\_genre\_id = 3 WHERE movie\_id = 9  
Message: 0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
  - 2 18:39:20 SELECT\*FROM tb\_movie WHERE movie\_id=9 LIMIT 0, 1000  
Message: 1 row(s) returned

movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_user	created_date	updated_date
9	Ocho Apellidos Catalanes	2016-06-09	Digital	3	OS_SGAD		