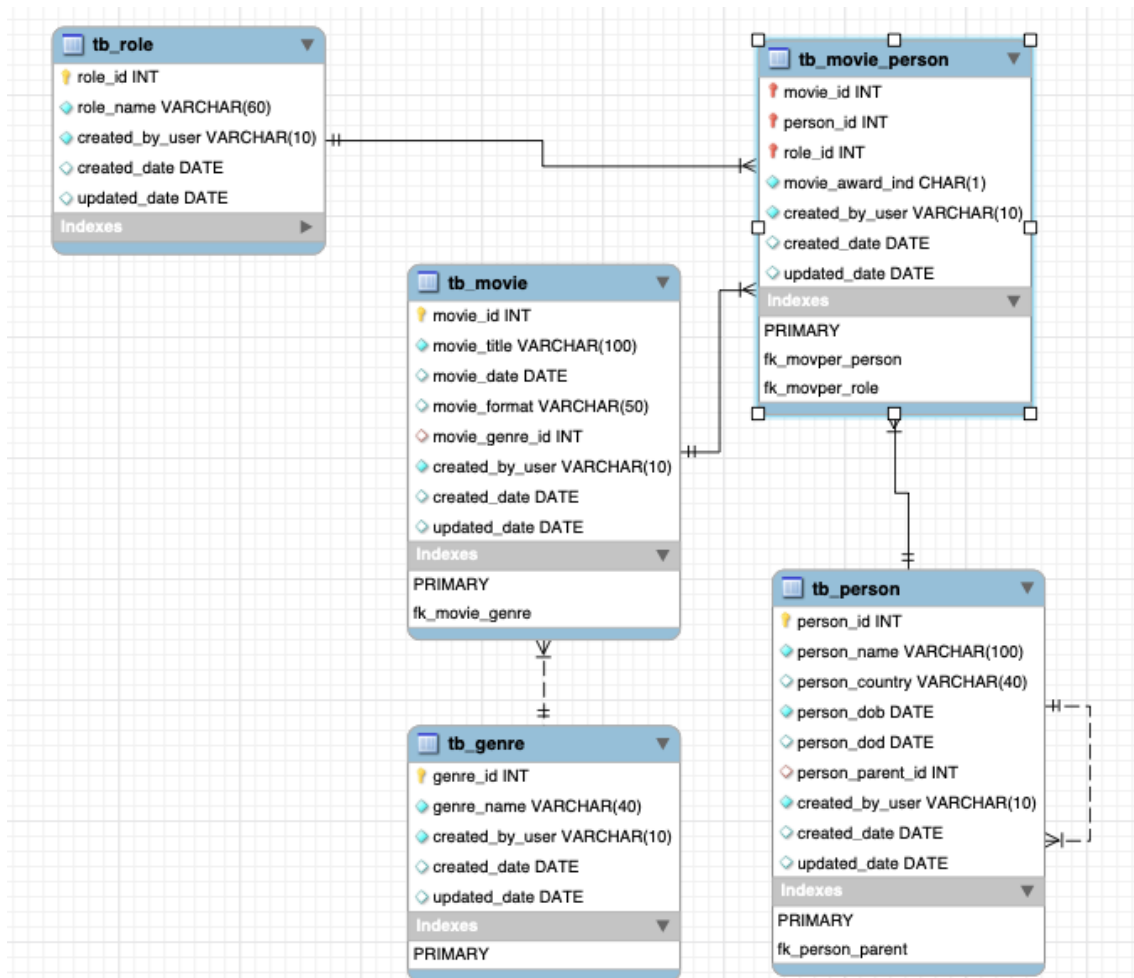


Tasca M1 T01

- 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.

Esquema:



Luego de crear la estructura de base de datos llamada "movies" con los datos proporcionados por la profesora, podemos identificar 5 tablas las cuales describiremos a continuación:

Primera tabla: tb_genre

Esta formada por 5 columnas en donde se destacan genre_id y genre_name relacionando cada nombre de genero con su numero de id.

The screenshot shows the SQL Studio interface. On the left, the 'SCHEMAS' pane displays the database structure: firstdatabase > movies > Tables > tb_genre. The 'Columns' pane for tb_genre lists: genre_id (int PK), genre_name (varchar(10)), created_by_user (varchar(10)), created_date (date), and updated_date (date). The main window shows a query result grid for 'select * from tb_genre'. The data is as follows:

genre_id	genre_name	created_by_us...	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
89	Documentales	OS_SGAD	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Segunda tabla tb_movie:

Está formada por ocho columnas en donde las principales son movie_id , movie_title y movie_genre_id donde se asigna una id a cada película y a su vez un numero de id para su genero, además podemos encontrar información como la fecha de lanzamiento entre otras.

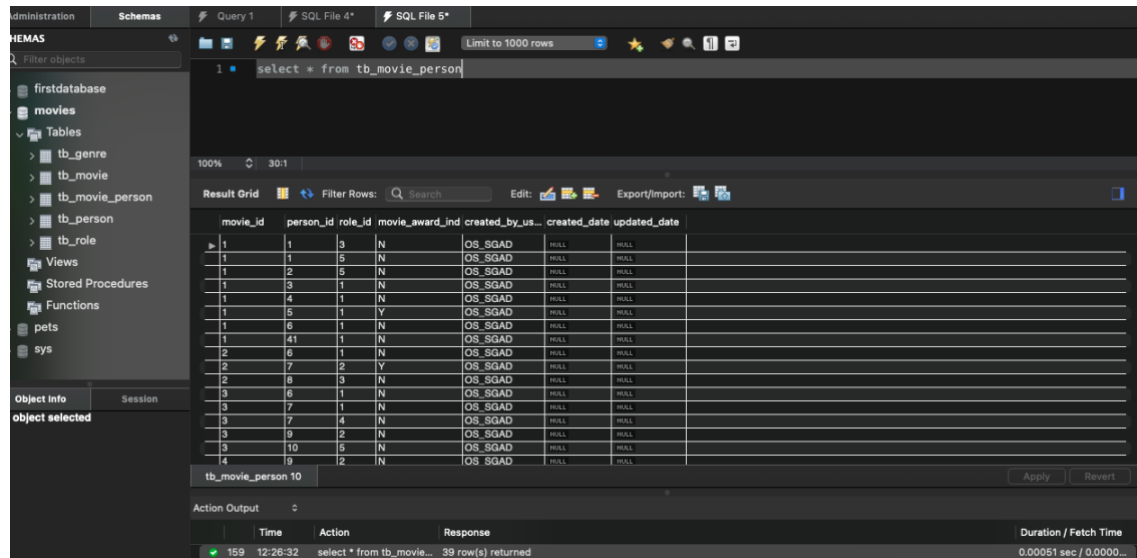
The screenshot shows the SQL Studio interface. On the left, the 'SCHEMAS' pane displays the database structure: firstdatabase > movies > Tables > tb_movie. The 'Columns' pane for tb_movie lists: movie_id (int PK), movie_title (varchar(100)), movie_date (date), movie_format (varchar(10)), movie_genre_id (int), created_by_user (varchar(10)), created_date (date), and updated_date (date). The main window shows a query result grid for 'select * from tb_movie'. The data is as follows:

movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_us...	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	El Financiero	1982-07-25	Film	5	OS_SGAD	NULL	NULL
7	Psycho	1980-05-08	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	3	OS_SGAD	NULL	NULL
10	El día de la bestia	2002-09-04	Digital	8	OS_SGAD	NULL	NULL
12	El día 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	NULL

Tercera tabla: tb_movie_person

En esta tabla podemos encontrar siete columnas de las cuales las principales son movie_id, person_id y role_id en donde podemos relacionar las películas, con las

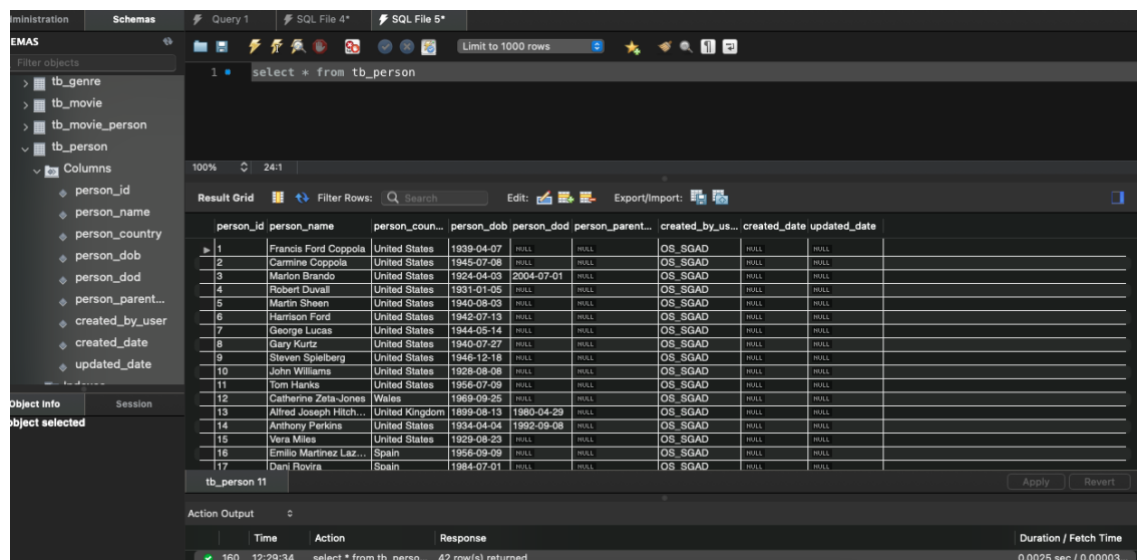
personas y su rol dentro de cada película ya que puede ser el caso que un actor sea a su vez productor en una misma película.



movie_id	person_id	role_id	movie_ward_ind	created_by_us...	created_date	updated_date
1	1	5	N	OS_SGAD	NULL	NULL
1	1	5	N	OS_SGAD	NULL	NULL
1	2	5	N	OS_SGAD	NULL	NULL
1	3	1	N	OS_SGAD	NULL	NULL
1	4	1	N	OS_SGAD	NULL	NULL
1	5	1	Y	OS_SGAD	NULL	NULL
1	6	1	N	OS_SGAD	NULL	NULL
1	41	1	N	OS_SGAD	NULL	NULL
2	6	1	N	OS_SGAD	NULL	NULL
2	7	2	Y	OS_SGAD	NULL	NULL
2	8	3	N	OS_SGAD	NULL	NULL
3	6	1	N	OS_SGAD	NULL	NULL
3	7	1	N	OS_SGAD	NULL	NULL
3	7	4	N	OS_SGAD	NULL	NULL
3	9	2	N	OS_SGAD	NULL	NULL
3	10	5	N	OS_SGAD	NULL	NULL
4	9	2	N	OS_SGAD	NULL	NULL

Cuarta tabla: tb_person

Esta tabla consta de 9 columnas en donde las principales son person_id y person_name para asignar un id a cada nombre de la persona y cuenta con otra información relevante como puede ser la fecha de nacimiento, difusión y país de nacimiento.



person_id	person_name	person_coun...	person_dob	person_dod	person_parent...	created_by_us...	created_date	updated_date
1	Francis Ford Coppola	United States	1939-04-07	NULL	NULL	OS_SGAD	NULL	NULL
2	Carmin Coppola	United States	1945-07-08	NULL	NULL	OS_SGAD	NULL	NULL
3	Marlon Brando	United States	1924-04-03	2004-07-01	NULL	OS_SGAD	NULL	NULL
4	Robert Duval	United States	1931-01-05	NULL	NULL	OS_SGAD	NULL	NULL
5	Martin Sheen	United States	1940-03-09	NULL	NULL	OS_SGAD	NULL	NULL
6	Harrison Ford	United States	1942-07-13	NULL	NULL	OS_SGAD	NULL	NULL
7	George Lucas	United States	1944-05-14	NULL	NULL	OS_SGAD	NULL	NULL
8	Gary Kurtz	United States	1940-07-27	NULL	NULL	OS_SGAD	NULL	NULL
9	Steven Spielberg	United States	1946-12-18	NULL	NULL	OS_SGAD	NULL	NULL
10	John Williams	United States	1932-08-08	NULL	NULL	OS_SGAD	NULL	NULL
11	Tom Hanks	United States	1956-07-09	NULL	NULL	OS_SGAD	NULL	NULL
12	Catherine Zeta-Jones	Wales	1969-09-25	NULL	NULL	OS_SGAD	NULL	NULL
13	Alfred Joseph Hitch...	United Kingdom	1899-08-13	1980-04-29	NULL	OS_SGAD	NULL	NULL
14	Anthony Perkins	United States	1934-04-04	1992-09-08	NULL	OS_SGAD	NULL	NULL
15	Vera Miles	United States	1929-08-23	NULL	NULL	OS_SGAD	NULL	NULL
16	Emilio Martinez Laz...	Spain	1956-09-09	NULL	NULL	OS_SGAD	NULL	NULL
17	Dani Rovira	Spain	1984-07-01	NULL	NULL	OS_SGAD	NULL	NULL

Quinta tabla: tb_role

Esta tabla formada por cinco columnas principalmente cumple la función de relacionar un role_id con un role_name para así relacionar cada rol de película con un id.

The screenshot shows a database management interface with a sidebar on the left displaying a schema tree. The main area shows a query result for 'tb_role'.

role_id	role_name	created_by_us...	created_date	updated_date
1	Actor	OS_SGAD	NULL	NULL
2	Director	OS_SGAD	NULL	NULL
3	Producer	OS_SGAD	NULL	NULL
4	Guionista	OS_SGAD	NULL	NULL
5	Musica	OS_SGAD	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Below the table, the 'Action Output' section shows the execution details:

Time	Action	Response	Duration / Fetch Time
161 12:32:36	select * from tb_role LI...	5 row(s) returned	0.00042 sec / 0.0000...

- 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.

The screenshot shows a database management interface with a sidebar on the left displaying a schema tree. The main area shows a query result for 'tb_person'.

person_name	person_coun...	person_dob
John Williams	United States	1928-08-08
Vera Miles	United States	1929-08-23
Sean Connery	Scotland	1930-07-08
Robert Duvall	United States	1931-01-05
Morgan Freeman	United States	1935-10-01
Francis Ford Coppola	United States	1939-04-07
Gary Kurtz	United States	1940-07-27
Martin Sheen	United States	1940-08-03
Harrison Ford	United States	1942-07-13
George Lucas	United States	1944-05-14
Carmin Coppola	United States	1945-07-08
Steven Spielberg	United States	1946-12-18
Tim Robbins	United States	1949-06-07

Below the table, the 'Action Output' section shows the execution details:

Time	Action	Response
140 12:17:23	SELECT person_name, person_country, person_dob FROM movies.tb...	38 row(s) returned
141 12:17:43	SELECT person_name, person_country, person_dob FROM movies.tb...	38 row(s) returned
142 12:17:59	SELECT person_name, person_country, person_dob FROM movies.tb...	38 row(s) returned

Para realizar este ejercicio primero debemos obtener el nombre de la persona, el país y la fecha de nacimiento mediante un SELECT, indicándole a la maquina desde que tabla debe coger la información mediante un FROM.

Una vez obtenidos estos datos, le indicaremos mediante una cláusula WHERE que nos filtre los datos donde la fecha de difusión sea NULL ó sea que no exista para así descartar a cualquier actor en donde conste una fecha de difusión.

Por último, ordenaremos los datos obtenidos de forma ascendente mediante un ORDER BY y ASC indicando la fecha de nacimiento para visualizar de forma ordenada desde las personas mayores a las mas jóvenes.

- 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.

```
1 • SELECT genre_name, count(movie_id)
2   FROM tb_movie
3  inner join tb_genre
4  on movie_genre_id = genre_id
5  group by movie_genre_id
6  order by count(movie_id) desc
```

100% 30:6

Result Grid Filter Rows: Search Export:

	genre_name	count(movie_i...
▶	Comedia	3
	Acción	2
	Ciencia Ficción	2
	Drama	2
	Romance	2
	Suspense	2
	Fantasia	1
	Terror	1
	Bélico	1

Result 17

Action Output

	Time	Action	Response
✓	162 10:46:20	SELECT genre_name, count(movie_id) FROM tb_movie inner join tb_g...	9 row(s) returned
✓	163 10:47:02	SELECT genre_name, count(movie_id) FROM tb_movie inner join tb_g...	9 row(s) returned
✓	164 10:47:05	SELECT genre_name, count(movie_id) FROM tb_movie inner join tb_g...	9 row(s) returned

Para comenzar mediante un SELECT, seleccionaremos el nombre del género y el número total de películas de este género las cuales deben ser contadas con un COUNT por su número de id (movie_id). Luego relacionaremos las dos tablas en donde se

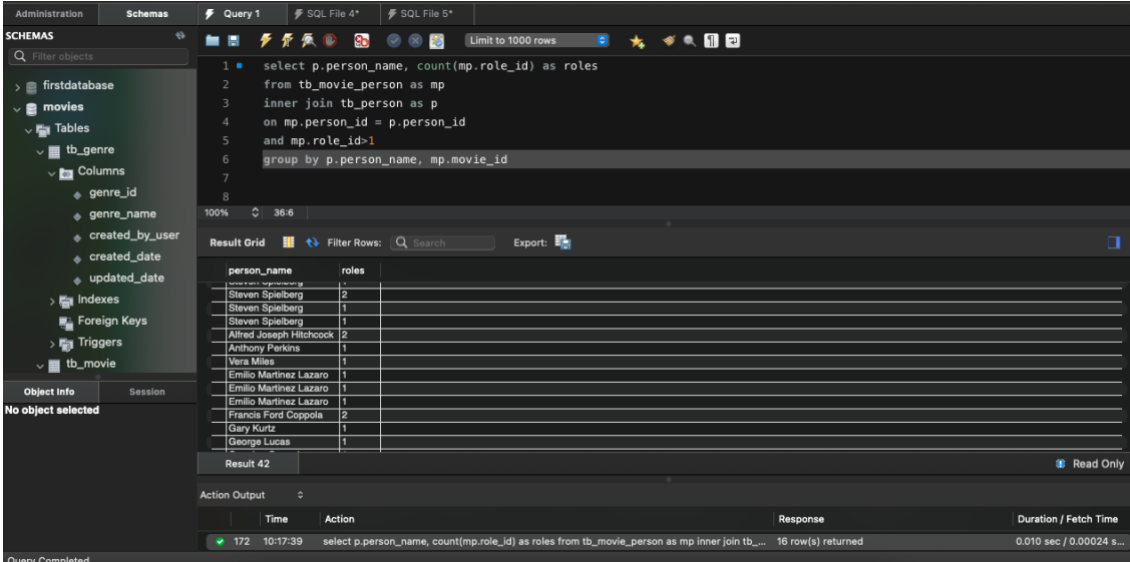
encuentran estos datos, tb_movie y tb_genre, esto lo hacemos mediante un INNER JOIN indicando la igualdad entre movie_genre_id y genre_id.
Para terminar, agrupamos los datos según (movie_genre_id) con GROUP BY y de damos un orden descendente con un ORDER BY (DESC).

- 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.

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



The screenshot shows a database management tool interface. On the left, a tree view shows the database structure: firstdatabase > movies > Tables > tb_genre > Columns (genre_id, genre_name, created_by_user, created_date, updated_date). The main area displays a SQL query:

```
1 select p.person_name, count(mp.role_id) as roles
2 from tb_movie_person as mp
3 inner join tb_person as p
4 on mp.person_id = p.person_id
5 and mp.role_id > 1
6 group by p.person_name, mp.movie_id
```

The results are shown in a table with two columns: person_name and roles. The data is as follows:

person_name	roles
Steven Spielberg	2
Steven Spielberg	1
Alfred Joseph Hitchcock	2
Anthony Perkins	1
Vera Miles	1
Emilio Martinez Lazaro	1
Emilio Martinez Lazaro	1
Francis Ford Coppola	2
Gary Kurtz	1
George Lucas	1

The bottom of the interface shows the query execution status: Query Completed, 172, 10:17:39, select p.person_name, count(mp.role_id) as roles from tb_movie_person as mp inner join tb_..., 16 row(s) returned, 0.010 sec / 0.00024 s...

Primero indicaremos a la máquina que las dos tablas de debemos relacionar las llamaremos de forma abreviada para hacer más sencillo el trabajo, esto se hace mediante un AS de este modo le decimos a la máquina que la tabla tb_movie_person la llame como mp y la tabla tb_person la llame como p.

Una vez realizadas estas abreviaciones mediante SELECT seleccionamos la columna de nombres poniendo la abreviación (p) seguida de un punto y el nombre de la columna (person_name), también queremos seleccionar el número máximo de roles por lo que seleccionamos con un COUNT la columna role_id indicando a que tabla pertenece mediante la abreviación (mp.role_id), por ultimo le indicaremos a la máquina que este conteo de los roles los llame con la abreviatura roles.

El segundo paso es relacionar las tablas en donde se encuentran los datos mediante un INNER JOIN relacionando las tablas tb_movie_person (mp) en donde se encuentran datos como movie_id, role_id y person_id y la tabla tb_person en donde se encuentran

datos como el nombre de las personas (person_name) en donde la igualdad se encontrará en las columnas person_id.

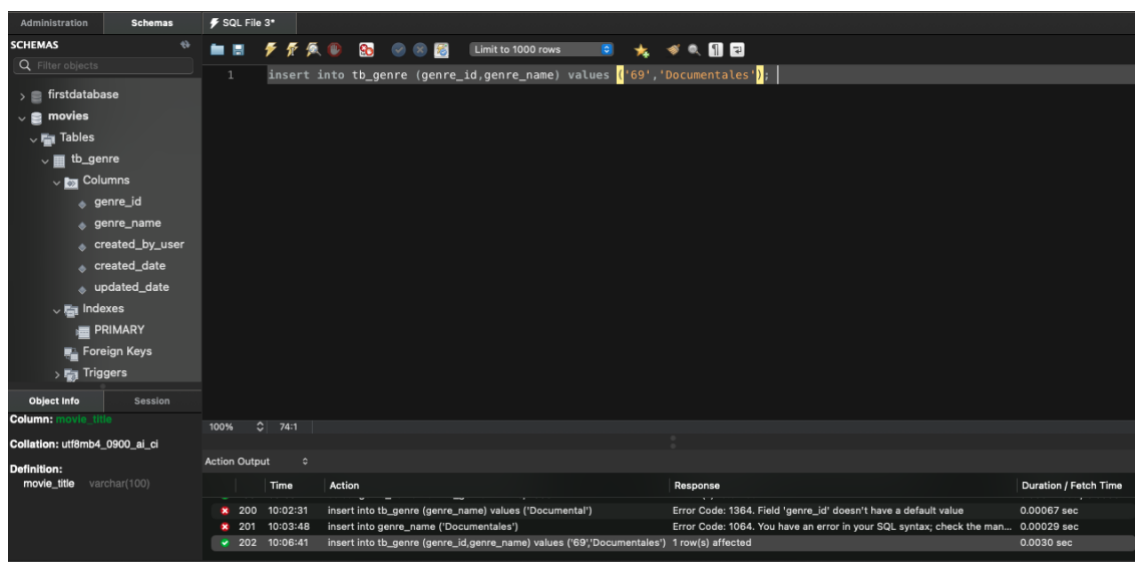
Una vez relacionadas las tablas y realizado el conteo del máximo de roles en el SELECT, le indicaremos a la maquina mediante un AND que solo muestre las personas que han cumplido más de un rol dentro de una misma película.

Por último, agruparemos la información según el nombre de la persona y el id de la película.

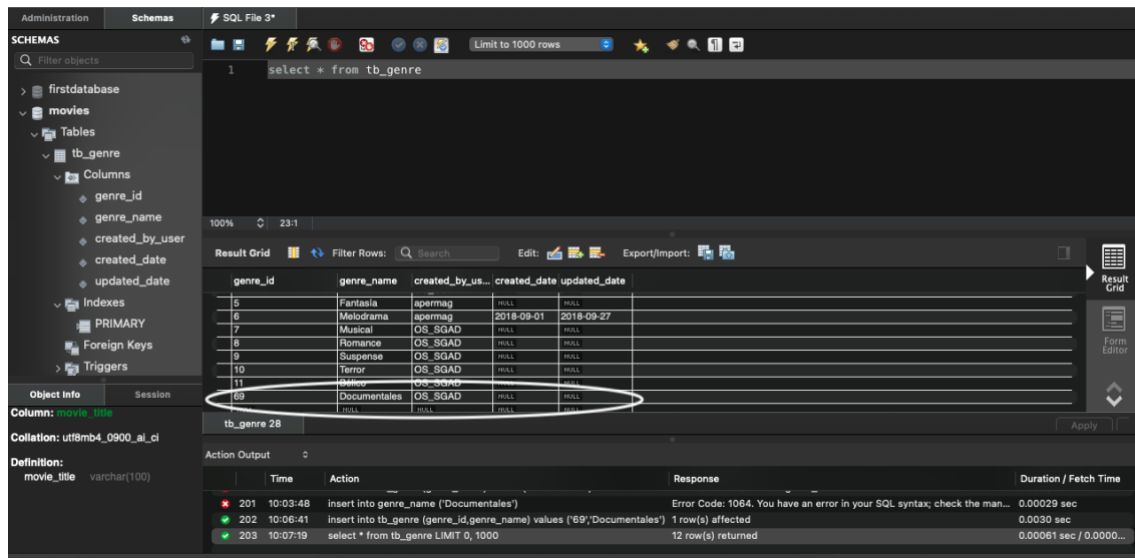
- Exercici 5

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.



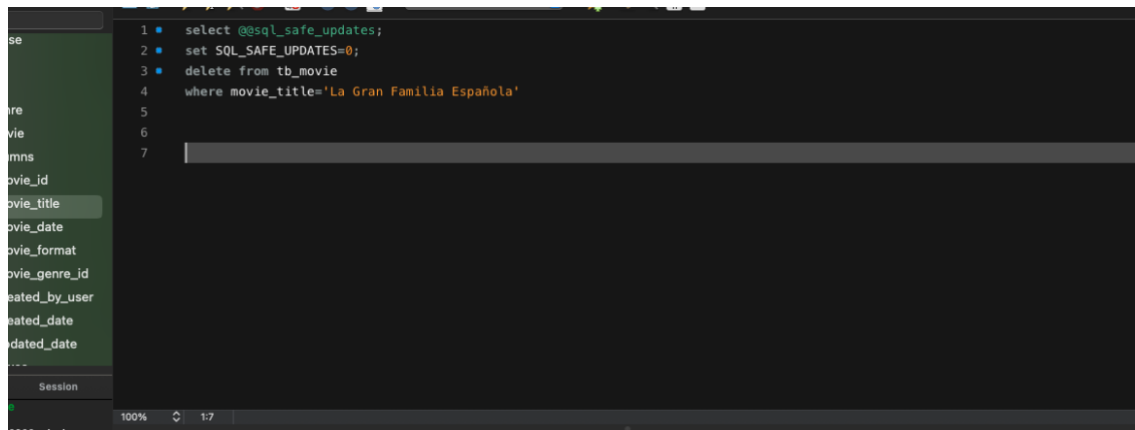
Para agregar información a un campo determinado dentro de una columna, debemos seguir la estructura con la cual fue formada dicha columna, de este modo para agregar un nuevo campo con datos dentro de la tabla de tb_genre lo hacemos mediante un INSERT INTO indicando el nombre de la tabla y posteriormente replicando el formato de dicha tabla en este caso (genre_id, genre_name) para después asignar los valores con un VALUE y aportando la información que queramos en este caso ('69', 'Documentales') siempre entre paréntesis, separado por comas y con la información entre comillas simples.



- Exercici 6

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

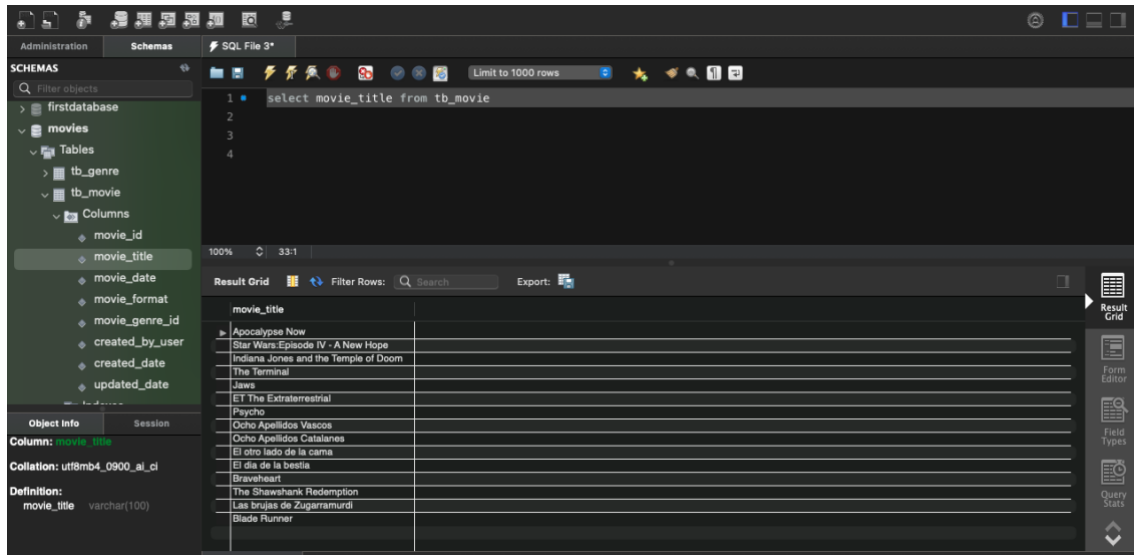
Elimina la pel·lícula "La Gran Familia Española" de la base de dades.



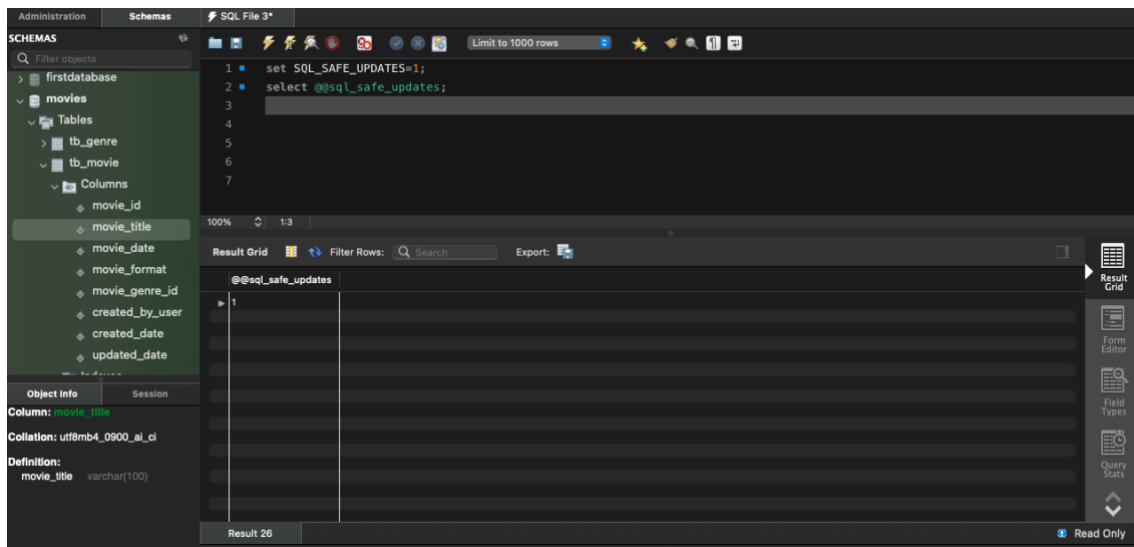
Las bases de datos por defecto tienen estas opciones deshabilitadas por seguridad ya que una vez eliminada la información esta no se podrá recuperar, para comprobar si esta opción esta habilitada o no debemos comprobarlo mediante SELECT @@sql_safe_updates siendo que si esta seteado en 1 esta activado y si esta seteado en 0 esta desactivado.

Una vez seteada la opción mediante SET SQL_SAFE_UPDATE=0

Procedemos a eliminar el campo deseado mediante un DELETE indicando primero la tabla a la cual nos referimos, en este caso tb_movie para luego indicar mediante una cláusula WHERE que campo de la columna deseamos eliminar, en este caso movie_title='La Gran Familia Española'



Comprobamos que el campo se haya eliminado.



Y por último es una práctica recomendable volver a setear la opción de eliminar datos a 1 para no cometer errores y eliminar información no deseada.

- 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.

Administration Schemas SQL File 3*

Limit to 1000 rows

1 set SQL_SAFE_UPDATES=0

2

3

Column: movie_title

Collation: utf8mb4_0900_ai_ci

Definition: movie_title varchar(100)

Action Output

	Time	Action	Response	Duration / Fetch Time
214	10:23:41	SQL_SAFE_UPDATES=0	Error Code: 1064. You have an error in your SQL syntax; check the man...	0.00031 sec
215	10:24:26	SET SQL_SAFE_UPDATES=0	Error Code: 1231. Variable 'sql_safe_updates' can't be set to the value...	0.0011 sec
216	10:24:35	set SQL_SAFE_UPDATES=0	0 row(s) affected	0.00029 sec

Administration Schemas SQL File 3*

Limit to 1000 rows

1 update tb_movie set movie_genre_id='3'

2 where movie_title='Ocho Apellidos Catalanes'

3

Column: movie_title

Collation: utf8mb4_0900_ai_ci

Definition: movie_title varchar(100)

Action Output

	Time	Action	Response	Duration / Fetch Time
219	10:26:03	select * from tb_movie LIMIT 0, 1000	15 row(s) returned	0.00089 sec / 0.000...
220	10:26:39	select * from tb_genre LIMIT 0, 1000	12 row(s) returned	0.00057 sec / 0.0000...
221	10:27:26	update tb_movie set movie_genre_id='3' where movie_title='Ocho Ap...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.0040 sec

Query Completed

Administration Schemas SQL File 3*

Limit to 1000 rows

1 select * from tb_movie

2

Column: movie_title

Collation: utf8mb4_0900_ai_ci

Definition: movie_title varchar(100)

Result Grid

movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_us...	created_date	updated_date
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	1960-09-08	Film	9	OS_SGAD	NULL	NULL
8	Ocho Apellidos Catalanes	2014-08-14	Digital	3	OS_SGAD	NULL	NULL
9	Ocho Apellidos Catalanes	2016-06-09	Digital	3	OS_SGAD	NULL	NULL
10	El otro lado de la cama	2002-08-04	Digital	8	OS_SGAD	NULL	NULL
11	El día de la bestia	1994-10-06	Film	1	OS_SGAD	NULL	NULL

tb_movie 34

Action Output

	Time	Action	Response	Duration / Fetch Time
220	10:26:39	select * from tb_genre LIMIT 0, 1000	12 row(s) returned	0.00057 sec / 0.0000...
221	10:27:26	update tb_movie set movie_genre_id='3' where movie_title='Ocho Ap...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.0040 sec
222	10:27:51	select * from tb_movie LIMIT 0, 1000	15 row(s) returned	0.00048 sec / 0.000...

Query Completed

De igual manera que en el anterior punto primero debemos activar la opción de poder actualizar campos ya añadidos para ser modificados.

Continuaremos con actualizar mediante la clausula UPDATE indicando primero la tabla (tb_movie) y luego con SET cual es el campo que deseamos cambiar (movie_genre_id='3') para posteriormente indicarle mediante un calusula WHERE cual es la fila a la que queremos asignar dicho campo en este caso movie_title='Ocho Apellidos Calatanés'.