

Chapter 1

Bases de dades relacionals (SQL)

1.1 Tasca M1 T01

Ejercicio 1.1. *A partir de los documentos adjuntos (estructura y datos), cree una base de datos con MySQL. Muestre las características principales del esquema creado y explique las diferentes tablas y variables que hay.*

Solución. Se puede ver que en la base de datos **movies** las tablas están relacionadas en las cuales existen claves primarias y foraneas.

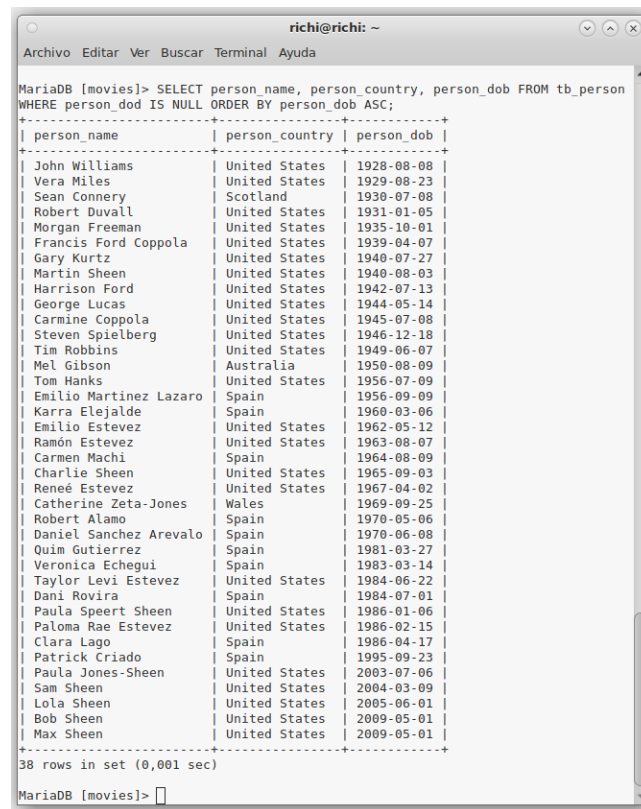
La tabla `tb_person` tiene una **PRIMARY KEY** única no nula y es un modelo uno a uno. Dado que, a cada persona le corresponde un solo identificador. Este mismo razonamiento también sucede con las tablas `tb_role`, `tb_movies` y `tb_genre`.

En la tabla `tb_movie_person` existen una relación de uno a muchos. Dado que, una persona puede ejercer diferentes roles. Lo mismo sucede con el género pues existen muchas películas para cierto género. □

Ejercicio 1.2. Tiene que obtener el nombre, el país y la fecha de nacimiento de aquellas personas por las cuales no conste una fecha de muerte y ordenar los datos de la persona más vieja a la persona más joven.

Solución.

```
SELECT person_name, person_country, person_dob FROM tb_person
WHERE person_dod IS NULL ORDER BY person_dob ASC;
```



The screenshot shows a terminal window titled 'richi@richi: ~' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal displays a SQL query executed in MariaDB [movies]:

```
MariaDB [movies]> SELECT person_name, person_country, person_dob FROM tb_person
WHERE person_dod IS NULL ORDER BY person_dob ASC;
```

The result is displayed as a table with three columns: person_name, person_country, and person_dob. The data is sorted by birth date (person_dob) in ascending order. The table contains 38 rows of data, including names like John Williams, Vera Miles, Sean Connery, Robert Duvall, Morgan Freeman, Francis Ford Coppola, Gary Kurtz, Martin Sheen, Harrison Ford, George Lucas, Carmine Coppola, Steven Spielberg, Tim Robbins, Mel Gibson, Tom Hanks, Emilio Martinez Lazaro, Karra Elejalde, Emilio Estevez, Ramón Estevez, Carmen Machi, Charlie Sheen, René Estevez, Catherine Zeta-Jones, Robert Alamo, Daniel Sanchez Arevalo, Quim Gutierrez, Veronica Echegui, Taylor Levi Estevez, Dani Rovira, Paula Speert Sheen, Paloma Rae Estevez, Clara Lago, Patrick Criado, Paula Jones-Sheen, Sam Sheen, Lola Sheen, Bob Sheen, and Max Sheen. The terminal also shows '38 rows in set (0,001 sec)' and the prompt 'MariaDB [movies]>'.

person_name	person_country	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
Carmine Coppola	United States	1945-07-08
Steven Spielberg	United States	1946-12-18
Tim Robbins	United States	1949-06-07
Mel Gibson	Australia	1950-08-09
Tom Hanks	United States	1956-07-09
Emilio Martinez Lazaro	Spain	1956-09-09
Karra Elejalde	Spain	1960-03-06
Emilio Estevez	United States	1962-05-12
Ramón Estevez	United States	1963-08-07
Carmen Machi	Spain	1964-08-09
Charlie Sheen	United States	1965-09-03
René Estevez	United States	1967-04-02
Catherine Zeta-Jones	Wales	1969-09-25
Robert Alamo	Spain	1970-05-06
Daniel Sanchez Arevalo	Spain	1970-06-08
Quim Gutierrez	Spain	1981-03-27
Veronica Echegui	Spain	1983-03-14
Taylor Levi Estevez	United States	1984-06-22
Dani Rovira	Spain	1984-07-01
Paula Speert Sheen	United States	1986-01-06
Paloma Rae Estevez	United States	1986-02-15
Clara Lago	Spain	1986-04-17
Patrick Criado	Spain	1995-09-23
Paula Jones-Sheen	United States	2003-07-06
Sam Sheen	United States	2004-03-09
Lola Sheen	United States	2005-06-01
Bob Sheen	United States	2009-05-01
Max Sheen	United States	2009-05-01

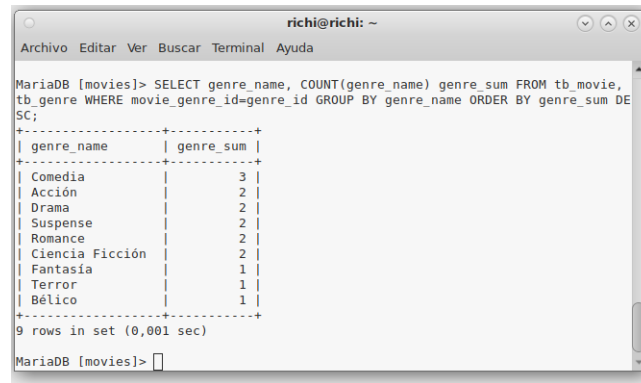
Figure 1.1:

□

Ejercicio 1.3. Tiene que obtener el nombre del género y el número total de películas de este género y ordenarlo por orden descendente de número total de películas.

Solución.

```
SELECT genre_name, COUNT(genre_name) genre_sum FROM tb_movie, tb_genre
WHERE movie_genre_id=genre_id
GROUP BY genre_name ORDER BY genre_sum DESC;
```



```
richi@richi: ~
Archivo Editar Ver Buscar Terminal Ayuda

MariaDB [movies]> SELECT genre_name, COUNT(genre_name) genre_sum FROM tb_movie,
tb_genre WHERE movie_genre_id=genre_id GROUP BY genre_name ORDER BY genre_sum DE
SC;
+-----+-----+
| genre_name | genre_sum |
+-----+-----+
| Comedia    | 3         |
| Acción     | 2         |
| Drama      | 2         |
| Suspense   | 2         |
| Romance    | 2         |
| Ciencia Ficción | 2         |
| Fantasía   | 1         |
| Terror     | 1         |
| Bélico     | 1         |
+-----+-----+
9 rows in set (0,001 sec)

MariaDB [movies]> 
```

Figure 1.2:

□

Ejercicio 1.4.

- Tiene que obtener, para cada persona, su nombre y el número máximo de roles diferentes que ha tenido en una misma película.
- Posteriormente, muestre únicamente aquellas personas que hayan asumido más de un rol en una misma película.

Solución.

- ```
SELECT tb_person.person_name, COUNT(role_id) max_role
FROM tb_movie_person
INNER JOIN tb_person ON tb_movie_person.person_id=tb_person.person_id
GROUP BY tb_person.person_name;
```



```
richi@richi: ~
Archivo Editar Ver Buscar Terminal Ayuda

MariaDB [movies]> SELECT tb_person.person_name, COUNT(role_id) max_role FROM tb
movie_person INNER JOIN tb_person ON tb_movie_person.person_id=tb_person.person
id GROUP BY tb_person.person_name;
+-----+-----+
| person_name | max_role |
+-----+-----+
Alfred Joseph Hitchcock	3
Anthony Perkins	1
Carmen Machi	2
Carmine Coppola	1
Catherine Zeta-Jones	1
Charlie Sheen	1
Clara Lago	2
Dani Rovira	2
Daniel Sanchez Arevalo	2
Emilio Martinez Lazaro	3
Francis Ford Coppola	3
Gary Kurtz	1
George Lucas	3
Harrison Ford	3
Hector Colome	1
John Williams	1
Karra Elejalde	2
Marlon Brando	1
Martin Sheen	1
Mel Gibson	2
Morgan Freeman	1
Patrick Criado	1
Quim Gutierrez	1
Robert Alamo	1
Robert Duvall	1
Steven Spielberg	5
Tim Robbins	1
Tom Hanks	1
Vera Miles	1
Veronica Echegui	1
+-----+-----+
30 rows in set (0,001 sec)
```

Figure 1.3:

- ```
SELECT tb_person.person_name, COUNT(role_id) max_role
FROM tb_movie_person
INNER JOIN tb_person ON tb_movie_person.person_id=tb_person.person_id
GROUP BY tb_person.person_name HAVING COUNT(tb_movie_person.role_id)>1;
```

□

```

richi@richi: ~
Archivo Editar Ver Buscar Terminal Ayuda

MariaDB [movies]> SELECT tb_person.person_name, COUNT(role_id) max_role FROM tb_movie_person INNER JOIN tb_person ON tb_movie_person.person_id=tb_person.person_id GROUP BY tb_person.person_name HAVING COUNT(tb_movie_person.role_id)>1;
+-----+-----+
| person_name | max_role |
+-----+-----+
| Alfred Joseph Hitchcock | 3 |
| Carmen Machi | 2 |
| Clara Lago | 2 |
| Dani Rovira | 2 |
| Daniel Sanchez Arevalo | 2 |
| Emilio Martinez Lazaro | 3 |
| Francis Ford Coppola | 3 |
| George Lucas | 3 |
| Harrison Ford | 3 |
| Karra Elejalde | 2 |
| Mel Gibson | 2 |
| Steven Spielberg | 5 |
+-----+-----+
12 rows in set (0,001 sec)

MariaDB [movies]> 

```

Figure 1.4:

Ejercicio 1.5. Tiene que crear un nuevo género llamado "Documental" el cual tenga como identificador el número 69.

Solución.

INSERT INTO movies.tb_genre (genre_id, genre_name) VALUES (69, 'Documental');

```

richi@richi: ~
Archivo Editar Ver Buscar Terminal Ayuda

MariaDB [movies]> select * from tb_genre;
+-----+-----+-----+-----+-----+
| genre_id | genre_name | created_by_user | created_date | updated_date |
+-----+-----+-----+-----+-----+
| 1 | Acción | 05_SGAD | NULL | NULL |
| 2 | Ciencia Ficción | 05_SGAD | NULL | NULL |
| 3 | Comedia | 05_SGAD | NULL | NULL |
| 4 | Drama | 05_SGAD | NULL | NULL |
| 5 | Fantasía | apermag | NULL | NULL |
| 6 | Melodrama | apermag | 2018-09-01 | 2018-09-27 |
| 7 | Musical | 05_SGAD | NULL | NULL |
| 8 | Romance | 05_SGAD | NULL | NULL |
| 9 | Suspense | 05_SGAD | NULL | NULL |
| 10 | Terror | 05_SGAD | NULL | NULL |
| 11 | Bélico | 05_SGAD | NULL | NULL |
| 69 | Documental | 05_SGAD | NULL | NULL |
+-----+-----+-----+-----+-----+
12 rows in set (0,000 sec)

MariaDB [movies]> 

```

Figure 1.5:

□

Ejercicio 1.6. Elimine la película "La Grande Familia Española" de la base de datos.

Solución. Primero obtenemos el ID de la película mediante la siguiente consulta:

```

SELECT tb_movie.movie_id, tb_movie.movie_title FROM tb_movie
WHERE tb_movie.movie_title='La Gran Familia Española'

```

```

richi@richi: ~
Archivo Editar Ver Buscar Terminal Ayuda

MariaDB [movies]> SELECT tb_movie.movie_id, tb_movie.movie_title FROM tb_movie WHERE tb_movie.movie_title='La Gran Familia Española';
+-----+-----+
| movie_id | movie_title |
+-----+-----+
| 11 | La Gran Familia Española |
+-----+-----+
1 row in set (0,000 sec)

MariaDB [movies]> 

```

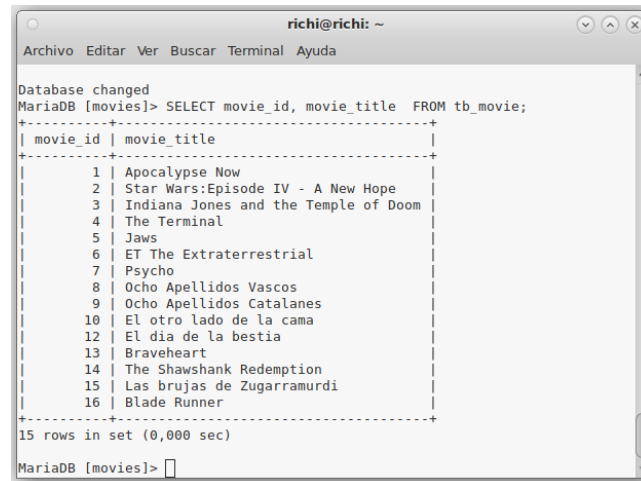
Figure 1.6:

A continuación, deshabilitamos la verificación de clave externa y eliminamos el registro con ID=11, luego volvemos a habilitar la verificación de clave externa de la siguiente manera:

```
SET FOREIGN_KEY_CHECKS=0;
```

```
DELETE FROM tb_movie WHERE tb_movie.movie_id=11;
```

```
SET FOREIGN_KEY_CHECKS=1;
```



The screenshot shows a terminal window titled 'richi@richi: ~'. The terminal output displays the result of a SQL query: 'SELECT movie_id, movie_title FROM tb_movie;'. The results are presented in a table with two columns: 'movie_id' and 'movie_title'. The table contains 16 rows of data, with the first row being '1 | Apocalypse Now' and the last row being '16 | Blade Runner'. The terminal also shows the message 'Database changed' and '15 rows in set (0,000 sec)'.

movie_id	movie_title
1	Apocalypse Now
2	Star Wars:Episode IV - A New Hope
3	Indiana Jones and the Temple of Doom
4	The Terminal
5	Jaws
6	ET The Extraterrestrial
7	Psycho
8	Ocho Apellidos Vascos
9	Ocho Apellidos Catalanes
10	El otro lado de la cama
12	El dia de la bestia
13	Braveheart
14	The Shawshank Redemption
15	Las brujas de Zugarramurdi
16	Blade Runner

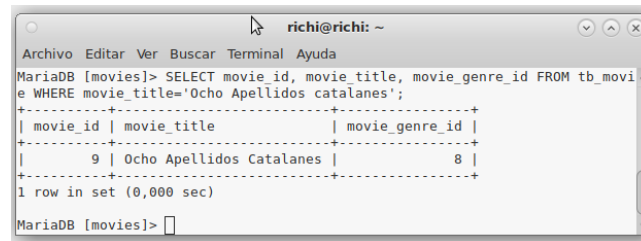
Figure 1.7:



Ejercicio 1.7. *Cambie el género de la película "Ocho apellidos catalanas" para que conste como comedia y no como romántica.*

Solución. Primero obtenemos los IDs de la película y de su género de la siguiente manera:

```
SELECT movie_id, movie_title, movie_genre_id FROM tb_movie
WHERE movie_title='Ocho Apellidos catalanes';
```



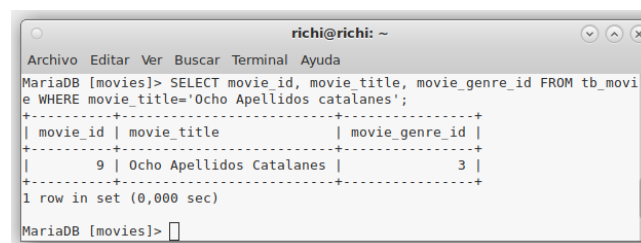
A screenshot of a terminal window titled 'richi@richi: ~'. The terminal shows a SQL query being executed in MariaDB: 'SELECT movie_id, movie_title, movie_genre_id FROM tb_movie WHERE movie_title='Ocho Apellidos catalanes';'. The result is displayed as a table with three columns: movie_id, movie_title, and movie_genre_id. The first row shows movie_id 9, movie_title 'Ocho Apellidos Catalanes', and movie_genre_id 8. Below the table, it says '1 row in set (0,000 sec)'. The prompt 'MariaDB [movies]>' is visible at the bottom.

movie_id	movie_title	movie_genre_id
9	Ocho Apellidos Catalanes	8

Figure 1.8:

Podemos ver que el ID de la película es 9 y el de su género es 8. Este 8 representa el género romance. Para cambiarlo a comedia actualizaremos el ID género a 3 de la siguiente manera:

```
UPDATE tb_movie SET movie_genre_id=3 WHERE movie_id=9;
```



A screenshot of a terminal window titled 'richi@richi: ~'. The terminal shows a SQL query being executed in MariaDB: 'SELECT movie_id, movie_title, movie_genre_id FROM tb_movie WHERE movie_title='Ocho Apellidos catalanes';'. The result is displayed as a table with three columns: movie_id, movie_title, and movie_genre_id. The first row shows movie_id 9, movie_title 'Ocho Apellidos Catalanes', and movie_genre_id 3. Below the table, it says '1 row in set (0,000 sec)'. The prompt 'MariaDB [movies]>' is visible at the bottom.

movie_id	movie_title	movie_genre_id
9	Ocho Apellidos Catalanes	3

Figure 1.9:

□