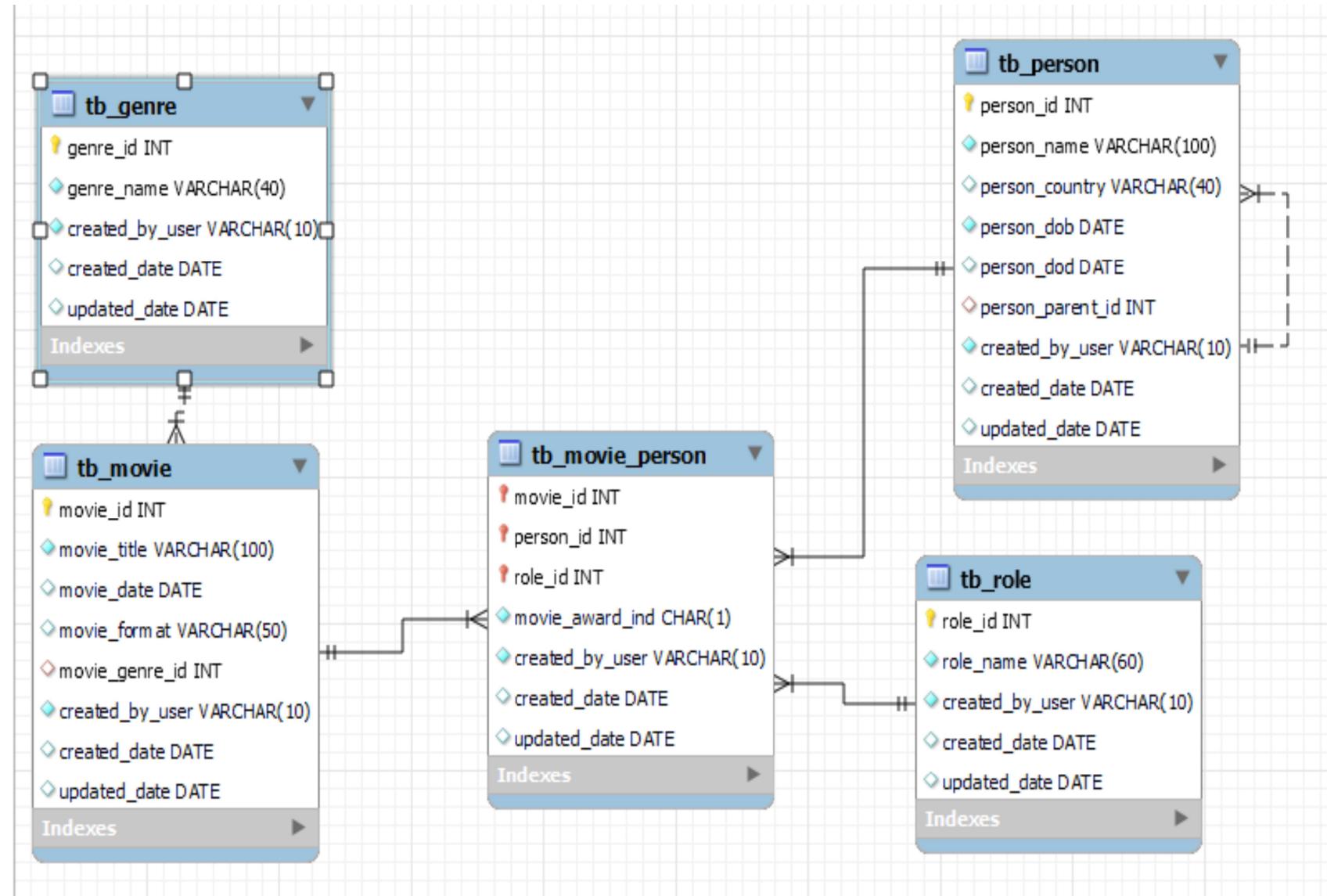


Exercici 1

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.



La tabla principal es “tb_movie_person” y tiene foreing keys de las otras tablas tb_movie, tb_role y tb_person. Hace de enlace de las otras tablas



Table Name: tb_movie_person Schema: movies

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
movie_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
person_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
role_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
movie_award_ind	CHAR(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
created_by_user	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	'OS_SGAD'					
created_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
updated_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL




Table Name: tb_movie_person Schema: movies

Foreign Key Name	Referenced Table	Column	Referenced Column
fk_movper_movie	'movies'.`tb_movie`	<input type="checkbox"/> movie_id	
fk_movper_person	'movies'.`tb_person`	<input checked="" type="checkbox"/> person_id	person_id
fk_movper_role	'movies'.`tb_role`	<input type="checkbox"/> role_id	
		<input type="checkbox"/> movie_award_ind	
		<input type="checkbox"/> created_by_user	
		<input type="checkbox"/> created_date	
		<input type="checkbox"/> updated_date	

La tabla principal es “tb_movie_person” y tiene foreing keys (FK) de las otras tablas tb_movie,tb_role y tb_person. Hace de enlace de las otras tablas

Aquí he dejado la tabla tb_movie_person para que se vea la conexión a través de la FK con las otras tablas.

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

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.

```
1 --- Has d'obtenir el nom, el país i la data de naixement
2 ---- d'aquelles persones per les quals no consti una data de mort i ordenar les dades
3 --- de la persona més vella a la persona més jove.
4
5
6 • use movies;
7
8 • select p.person_name, p.person_country, p.person_dod, p.person_dob
9   from tb_person p
10  where p.person_dod is null
11  order by p.person_dob asc;
12
13
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | 

	person_name	person_country	person_dod	person_dob
▶	John Williams	United States	NULL	1928-08-08
	Vera Miles	United States	NULL	1929-08-23
	Sean Connery	Scotland	NULL	1930-07-08
	Robert Duvall	United States	NULL	1931-01-05
	Morgan Freeman	United States	NULL	1935-10-01
	Francis Ford Coppola	United States	NULL	1939-04-07
	Gary Kurtz	United States	NULL	1940-07-27
	Martin Sheen	United States	NULL	1940-08-03
	Harrison Ford	United States	NULL	1942-07-13
	George Lucas	United States	NULL	1944-05-14
	Carmine Coppola	United States	NULL	1945-07-08
	Steven Spielberg	United States	NULL	1946-12-18
	Tim Robbins	United States	NULL	1949-06-07
	Mel Gibson	Australia	NULL	1950-08-09
	Tom Hanks	United States	NULL	1956-07-09
	Emilio Martínez Lázaro	Spain	NULL	1956-09-09

tb_person 33 ×

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 •  use movies;
2 •  select g.genre_id, g.genre_name, count( g.genre_id ) as Veces
3   from tb_genre g inner join tb_movie m on m.movie_genre_id = g.genre_id
4   group by g.genre_name
5   order by Veces desc;
6
7
```

Result Grid | Filter Rows: Export: Wrap Cell Content: 

	genre_id	genre_name	Veces
▶	3	Comedia	3
	1	Acción	2
	2	Ciencia Ficción	2
	4	Drama	2
	9	Suspense	2
	5	Fantasía	1
	8	Romance	1
	10	Terror	1
	11	Bélico	1



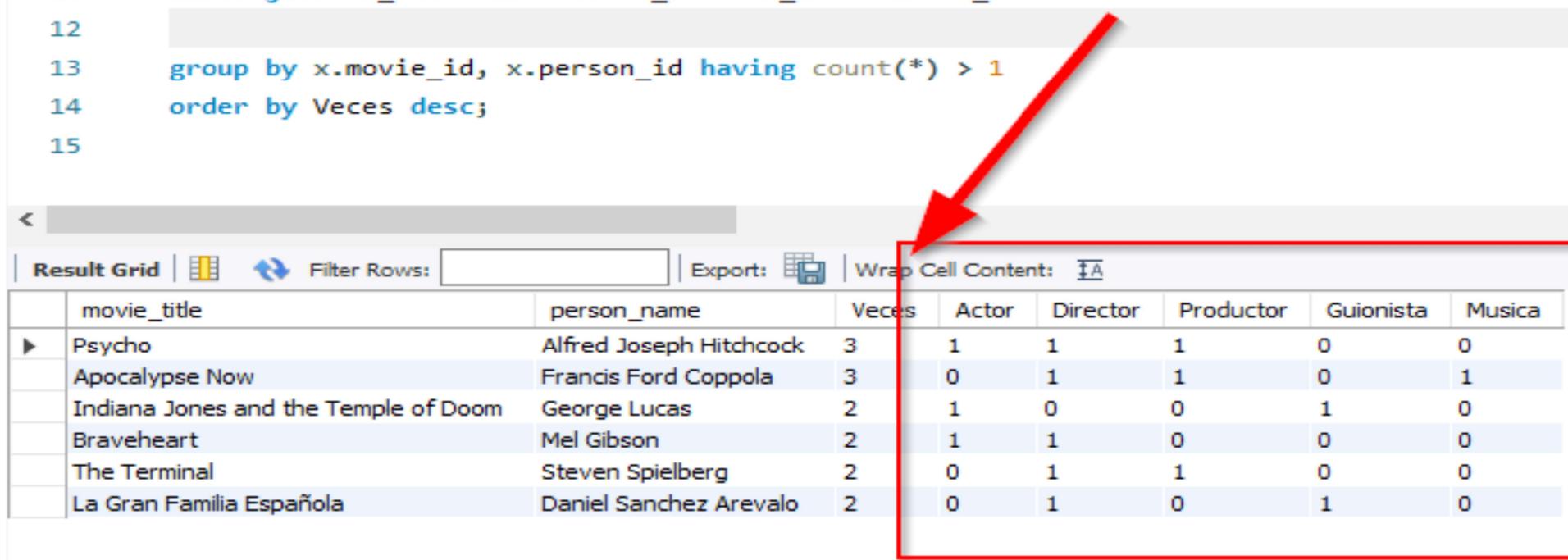
Exercici 4

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.

```
1 • use movies;
2
3     select tb_movie.movie_title, p.person_name, count(tb_role.role_name) as Veces,
4         sum( x.role_id = 1) as Actor,
5         sum( x.role_id = 2) as Director,
6         sum( x.role_id = 3) as Productor,
7         sum( x.role_id = 4) as Guionista,
8         sum( x.role_id = 5) as Musica
9     from tb_person p inner join tb_movie_person x on x.person_id = p.person_id
10    inner join tb_role  on x.role_id = tb_role.role_id
11    inner join tb_movie on x.movie_id = tb_movie.movie_id
12
13    group by x.movie_id, x.person_id having count(*) > 1
14    order by Veces desc;
15
```



The screenshot shows the MySQL Workbench interface with the results of the SQL query. The results are displayed in a grid with the following columns: movie_title, person_name, Veces, Actor, Director, Productor, Guionista, and Musica. The rows show data for Alfred Hitchcock, Francis Ford Coppola, George Lucas, Mel Gibson, Steven Spielberg, and Daniel Sanchez Arevalo, with their respective counts of roles and specific role types.

	movie_title	person_name	Veces	Actor	Director	Productor	Guionista	Musica
▶	Psycho	Alfred Joseph Hitchcock	3	1	1	1	0	0
	Apocalypse Now	Francis Ford Coppola	3	0	1	1	0	1
	Indiana Jones and the Temple of Doom	George Lucas	2	1	0	0	1	0
	Braveheart	Mel Gibson	2	1	1	0	0	0
	The Terminal	Steven Spielberg	2	0	1	1	0	0
	La Gran Familia Española	Daniel Sanchez Arevalo	2	0	1	0	1	0

- 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.

```
1
2      ---- Realitza la següent operació sobre la base de dades acabada de crear:
3
4      ---- Has de crear un nou gènere anomenat "Documental" el qual tingui com a identificador el nombre 69.
5
6 • use movies;
7 • DELETE FROM tb_genre
8     WHERE genre_id = 69;
9 • INSERT INTO movies.tb_genre (genre_id, genre_name, created_date)
10    select 69, 'Documental', '2022-07-19'
11    where Not EXISTS(select 1 from movies.tb_genre where genre_id = 69) ;
12 • select * from tb_genre;
13
14
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The grid displays the 'tb_genre' table with the following data:

	genre_id	genre_name	created_by_user	created_date	updated_date
1	1	Acción	OS SGAD	NULL	NULL
2	2	Ciencia Ficción	OS SGAD	NULL	NULL
3	3	Comedia	OS SGAD	NULL	NULL
4	4	Drama	OS SGAD	NULL	NULL
5	5	Fantasia	apermag	NULL	NULL
6	6	Melodrama	apermag	2018-09-01	2018-09-27
7	7	Musical	OS SGAD	NULL	NULL
8	8	Romance	OS SGAD	NULL	NULL
9	9	Suspense	OS SGAD	NULL	NULL
10	10	Terror	OS SGAD	NULL	NULL
11	11	bélico	OS SGAD	NULL	NULL
	69	Documental	OS SGAD	2022-07-19	NULL
	NULL	NULL	NULL	NULL	NULL

A red box highlights the last two rows of the grid, which correspond to the SQL operations shown above. The first row (id 11) is the new entry for 'Documental', and the second row (id 69) is the original entry that was deleted.

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.

The screenshot shows the MySQL Workbench interface with the following details:

SQL Editor:

```
1 ----- Fes la següent operació sobre la base de dades acabada de crear:  
2  
3 ----- Elimina la pel·lícula "La Gran Familia Española" de la base de dades  
4 • use movies;  
5  
6 • DELETE FROM tb_movie_person  
    WHERE movie_id = 11;  
8  
9 • DELETE FROM tb_movie  
10    WHERE movie_id = 11;  
11 • select * from tb_movie;
```

Result Grid:

movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_user	created_date	update
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	3	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

A pink arrow points to the last row of the grid, specifically the 'update' column for the movie with ID 11.

Action Output:

#	Time	Action	Message
93	17:16:31	select * from tb_movie LIMIT 0, 1000	16 row(s) returned
94	17:16:42	DELETE FROM tb_movie WHERE movie_id = 11	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails
95	17:19:40	DELETE FROM tb_movie_person WHERE movie_id = 11	7 row(s) affected
96	17:19:47	select * from tb_movie LIMIT 0, 1000	16 row(s) returned
97	17:19:53	DELETE FROM tb_movie WHERE movie_id = 11	1 row(s) affected
98	17:19:56	select * from tb_movie LIMIT 0, 1000	15 row(s) returned

¡Importante!

No se puede eliminar un registro de una tabla, si ese mismo aparece en otra como FK. Se produciría un problema de integridad referencial.

Para borrar un dato base, se deben eliminar todas las referencias a él desde otras tablas primero, pero si esos otros registros poseen otras dependencias, no se borrarán y tampoco la tabla base que planteas.

En síntesis, si tienes una dependencia del tipo A<-B<-C, no puedes borrar registros de A ni de B, sin borrar primero los de C.

- 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.

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

```
1 ----- Realitza la següent operació sobre la base de dades acabada de crear:  
2  
3 ----- el gènere de la pel·lícula "Ocho apellidos catalanes" perquè consti com a comèdia i no com a r  
4  
5 • use movies;  
6 • select * from tb_movie  
7 where movie_id = 9;  
8  
9 • UPDATE `movies`.`tb_movie`  
10 SET `movie_genre_id` = '3'  
11 WHERE (`movie_id` = '9');  
12 • select * from tb_movie where movie_id = 9;  
13
```

The result grid displays the following data:

	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	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The cell containing the value '3' in the 'movie_genre_id' column is circled in red.