

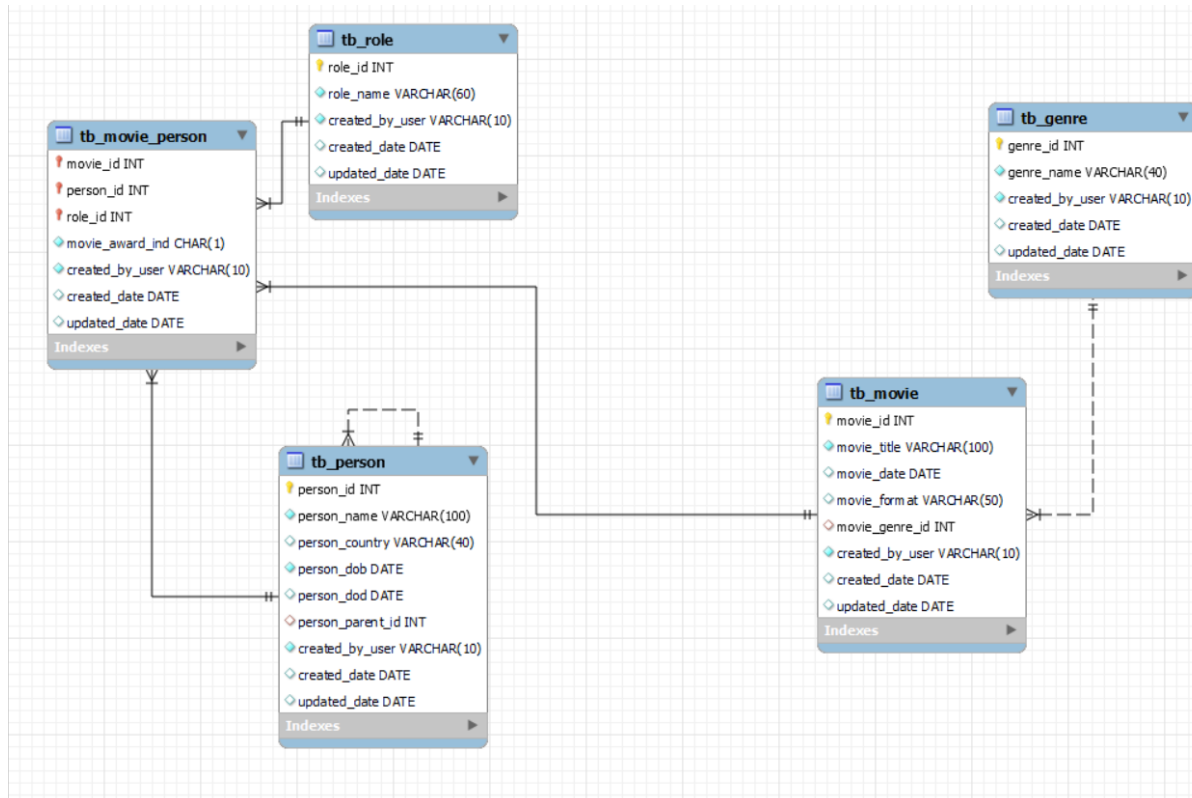
ITAcademy Especialització en Data Science

Sprint 1

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.



El esquema *movies* està format per cinc taules. Dos de les quals *tb_genre* i *tb_role* no contenen claus forànies d'altres taules, una d'elles *tb_person* és recursiva en tant que conté una clau de sí mateixa al camp *person_parent_id* i les altres dues *tb_movies* i *tb_movie_person* contenen claus forànies.

Totes les taules tenen tres atributs en comú *created_by_user*, *created date* i *updated_date* per fer referència respectivament a l'usuari que ha creat el registre, en quina data i en quina data s'ha modificat el registre. Així doncs, aquest tres camps no s'anomenaran quan es parli a continuació dels atributs de les taules.

La taula *tb_genre*, conté un llistat de gèneres filmogràfics amb els següents atributs:

- *genre_id* un atribut enter únic que actua com a clau primària
- *genre_name* un camp de text on figurarà el nom del gènere.

La taula *tb_movie* conté registres de pel·lícules amb els següents atributs:

- *movie_id* un nombre enter que actua com a clau primària
- *movie_title* un camp de text on figurarà el nom de la pel·lícula
- *movie_date* un camp datetime on figurarà la data en que es va estrenar la pel·lícula
- *movie_format* un atribut categòric que es diu si la pel·lícula es de format físic (Film) o digital (Digital)
- *movie_genre_id* un enter que conté una clau primària de la taula *tb_genre* per indicar a quin gènere correspon la pel·lícula.

La taula *tb_role* conté una relació de les diferents feines que es poden desenvolupar a una pel·lícula (actor, director, productor,...) i té els següents atributs:

- *role_id* el número enter que fa la funció de clau primària
- *role_name* un camp de text on es troba el nom de la feina en qüestió

La taula *tb_person* conté les dades personals de persones lligades al món de les pel·lícules, sense detallar quines feines han fet. Té els següents atributs:

- *person_id* l'enter que fa la funció de clau primària
- *person_name* camp de text que conté nom i cognoms de la persona
- *person_country* camp de text que conté el país on va neixer la persona
- *person_dob* camp datetime que reflexa la data de naixement de la persona
- *person_dod* camp datetime que reflexa la data de defunció de la persona si procedeix.
- *person_parent_id* un camp enter que fa referència a una de les claus primàries d'aquesta mateixa taula, per indicar qui és el pare/mare d'aquesta persona.

La taula *tb_movie_person* serveix per plasmar la relació entre una persona, quina tasca va desempenyar(role) i a quina pel·lícula. En aquest cas la clau primària és composta per la combinació de tres camps (*movie_id*, *person_id*, *role_id*). Té els atributs següents:

- *movie_id* enter clau forània que referencia la taula *tb_movie*
- *person_id* enter clau forània que referencia la taula *tb_person*
- *role_id* enter clau forània que referencia la taula *tb_role*
- *movie_award_ind* un caracter (Y/N) per indicar si es va rebre un premi pel rol a la pel·lícula

- 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 • SELECT person_name as Nom, person_country as PaísNaixement, person_dob as DataNaixement
2 FROM tb_person
3 WHERE person_dod is NULL
4 ORDER BY person_dob
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Nom	PaísNaixement	DataNaixement
▶	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

tb_person 7 x

- 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 tb_genre.genre_name, COUNT(*) as total_movies
2 FROM tb_movie INNER JOIN tb_genre
3 ON tb_movie.movie_genre_id=tb_genre.genre_id
4 GROUP BY movie_genre_id
5 ORDER BY total_movies DESC
```

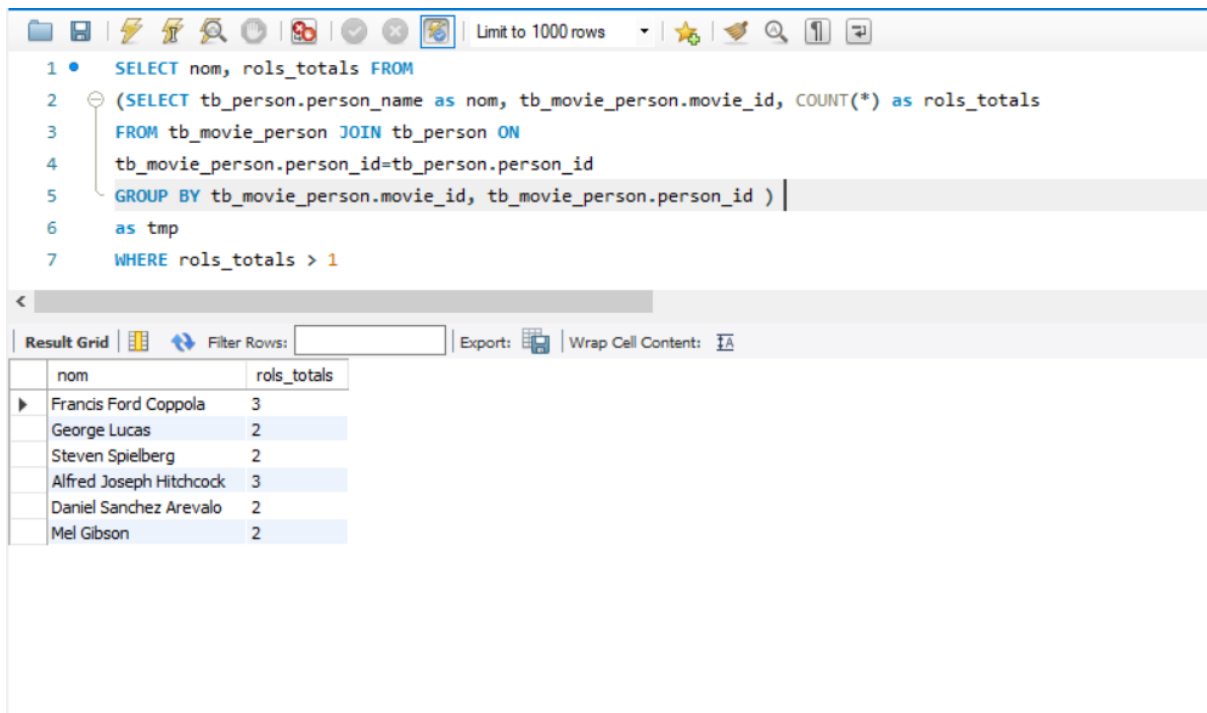
<		
Result Grid		
Filter Rows:		
Export:		
Wrap Cell Content:		
	genre_name	total_movies
▶	Comedia	3
	Acción	2
	Ciencia Ficción	2
	Drama	2
	Romance	2
	Suspense	2
	Fantasía	1
	Terror	1
	Bélico	1

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.



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 • SELECT nom, rols_totals FROM
2 (SELECT tb_person.person_name as nom, tb_movie_person.movie_id, COUNT(*) as rols_totals
3 FROM tb_movie_person JOIN tb_person ON
4 tb_movie_person.person_id=tb_person.person_id
5 GROUP BY tb_movie_person.movie_id, tb_movie_person.person_id )
6 as tmp
7 WHERE rols_totals > 1
```

Below the query editor, the 'Result Grid' is displayed with the following data:

nom	rols_totals
Francis Ford Coppola	3
George Lucas	2
Steven Spielberg	2
Alfred Joseph Hitchcock	3
Daniel Sanchez Arevalo	2
Mel Gibson	2

- 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.

Query 1

```
1 • INSERT INTO movies.tb_genre (genre_id, genre_name) VALUES (69, 'Documental')
```

Output

Action Output

#	Time	Action	Message
1	18:25:44	INSERT INTO movies.tb_genre (genre_id, genre_name) VALUES (69, 'Documental')	1 row(s) affected

1 SELECT * FROM tb_genre

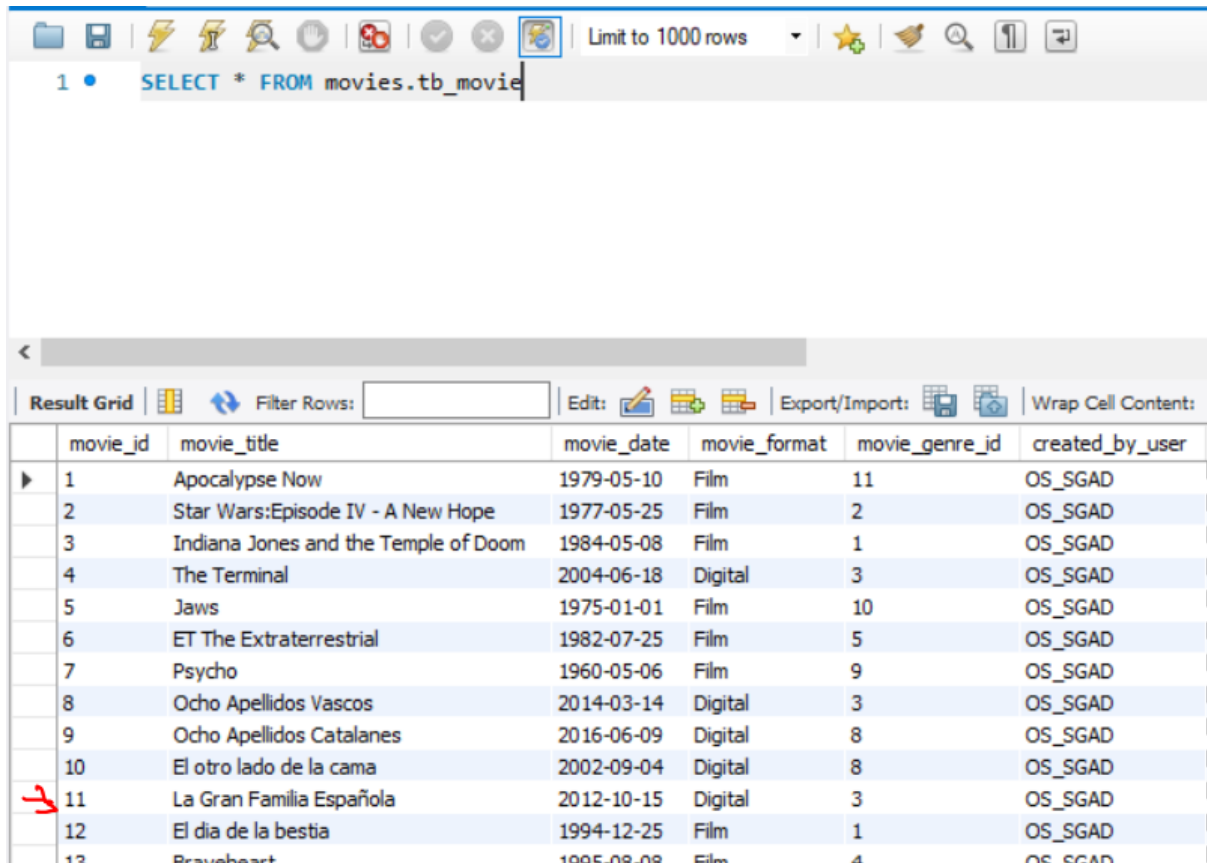
Result Grid

	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	NULL

- 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.

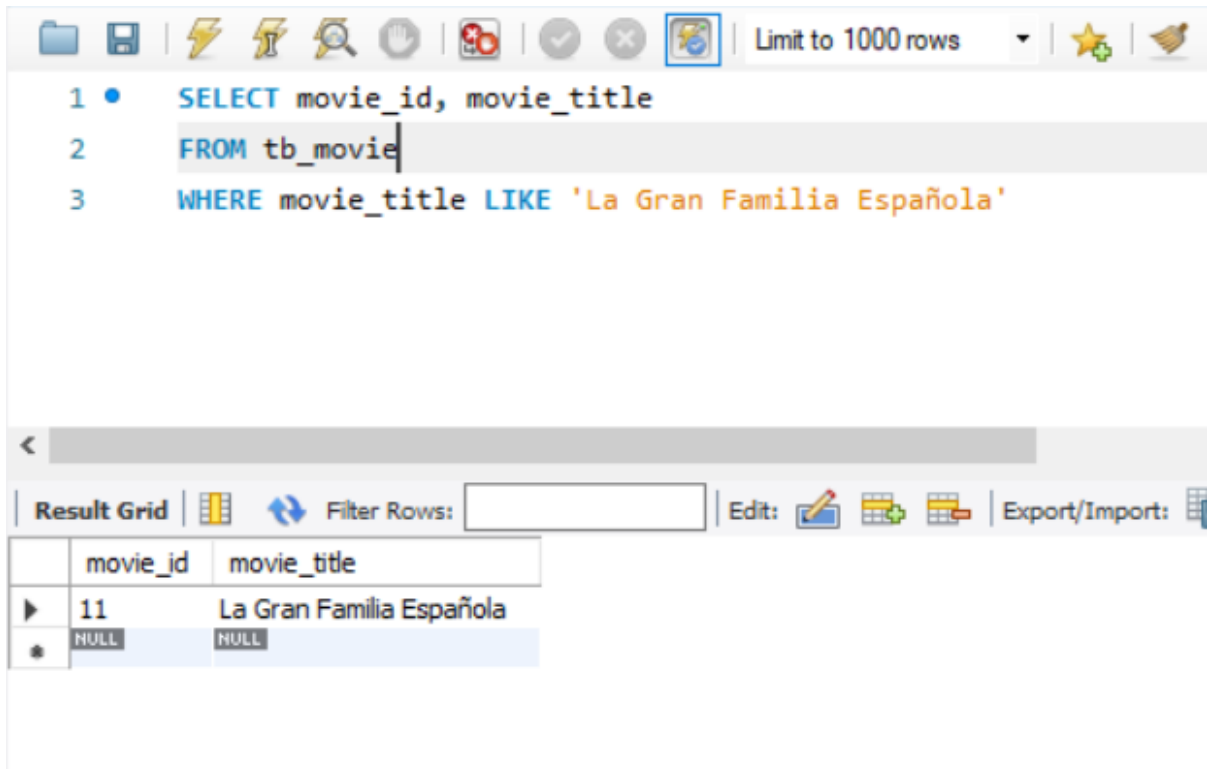


The screenshot shows a database management interface. At the top, there is a toolbar with various icons and a text input field containing the SQL query: `SELECT * FROM movies.tb_movie`. Below the query, there is a section labeled "Result Grid" which displays a table of movie data. The table has columns: `movie_id`, `movie_title`, `movie_date`, `movie_format`, `movie_genre_id`, and `created_by_user`. The data is as follows:

	movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_user
▶	1	Apocalypse Now	1979-05-10	Film	11	OS_SGAD
	2	Star Wars:Episode IV - A New Hope	1977-05-25	Film	2	OS_SGAD
	3	Indiana Jones and the Temple of Doom	1984-05-08	Film	1	OS_SGAD
	4	The Terminal	2004-06-18	Digital	3	OS_SGAD
	5	Jaws	1975-01-01	Film	10	OS_SGAD
	6	ET The Extraterrestrial	1982-07-25	Film	5	OS_SGAD
	7	Psycho	1960-05-06	Film	9	OS_SGAD
	8	Ocho Apellidos Vascos	2014-03-14	Digital	3	OS_SGAD
	9	Ocho Apellidos Catalanes	2016-06-09	Digital	8	OS_SGAD
	10	El otro lado de la cama	2002-09-04	Digital	8	OS_SGAD
→	11	La Gran Familia Española	2012-10-15	Digital	3	OS_SGAD
	12	El dia de la bestia	1994-12-25	Film	1	OS_SGAD
	13	Braveheart	1995-08-08	Film	4	OS_SGAD

Si intentem fer l'eliminació directament ens avisa que aquest registre actua com a clau forània a la tabla `tb_movie_person`, així que hem de fer l'eliminació allí primer.

Començarem per obtenir la `movie_id` de la pel·lícula:



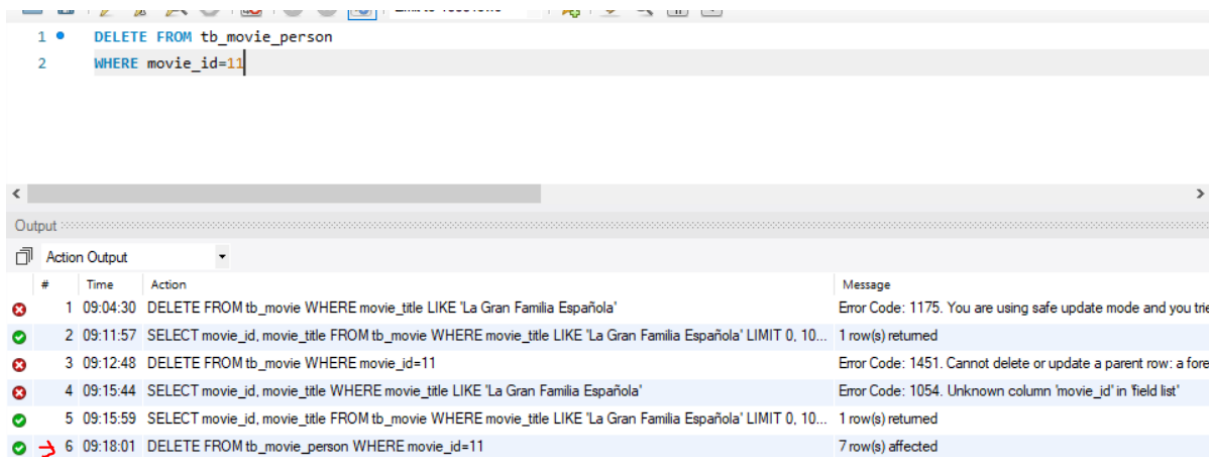
The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL query is entered in a text area:

```
1 • SELECT movie_id, movie_title
2 FROM tb_movie
3 WHERE movie_title LIKE 'La Gran Familia Española'
```

Below the query, there is a section labeled "Result Grid". It contains a table with two columns: "movie_id" and "movie_title". The table has one row with the values "11" and "La Gran Familia Española".

movie_id	movie_title
11	La Gran Familia Española

Sabent que el id de la pel·lícula és l'11 ja podem fer la eliminació a la taula *tb_movie_person*:



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL query is entered in a text area:

```
1 • DELETE FROM tb_movie_person
2 WHERE movie_id=11
```

Below the query, there is a section labeled "Output". It contains a table with three columns: "#", "Time", and "Action". The table has six rows, each representing a query execution. The last row shows the execution of the DELETE statement, which affected 7 rows.

#	Time	Action	Message
1	09:04:30	DELETE FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española'	Error Code: 1175. You are using safe update mode and you trie
2	09:11:57	SELECT movie_id, movie_title FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española' LIMIT 0, 10...	1 row(s) returned
3	09:12:48	DELETE FROM tb_movie WHERE movie_id=11	Error Code: 1451. Cannot delete or update a parent row: a fore
4	09:15:44	SELECT movie_id, movie_title WHERE movie_title LIKE 'La Gran Familia Española'	Error Code: 1054. Unknown column 'movie_id' in 'field list'
5	09:15:59	SELECT movie_id, movie_title FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española' LIMIT 0, 10...	1 row(s) returned
6	09:18:01	DELETE FROM tb_movie_person WHERE movie_id=11	7 row(s) affected

Ara ja podem fer l'eliminació del registre a la taula *tb_movie*:


```
1 • DELETE FROM tb_movie
2 WHERE movie_id=11
```

Output



Action Output

#	Time	Action	Message
✗ 1	09:04:30	DELETE FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española'	Error Code: 1175. Yc
✓ 2	09:11:57	SELECT movie_id, movie_title FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española' LIMIT 0, 10...	1 row(s) returned
✗ 3	09:12:48	DELETE FROM tb_movie WHERE movie_id=11	Error Code: 1451. Ca
✗ 4	09:15:44	SELECT movie_id, movie_title WHERE movie_title LIKE 'La Gran Familia Española'	Error Code: 1054. Ur
✓ 5	09:15:59	SELECT movie_id, movie_title FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española' LIMIT 0, 10...	1 row(s) returned
✓ 6	09:18:01	DELETE FROM tb_movie_person WHERE movie_id=11	7 row(s) affected
✓ → 7	09:19:40	DELETE FROM tb_movie WHERE movie_id=11	1 row(s) affected

- Exercici 7

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

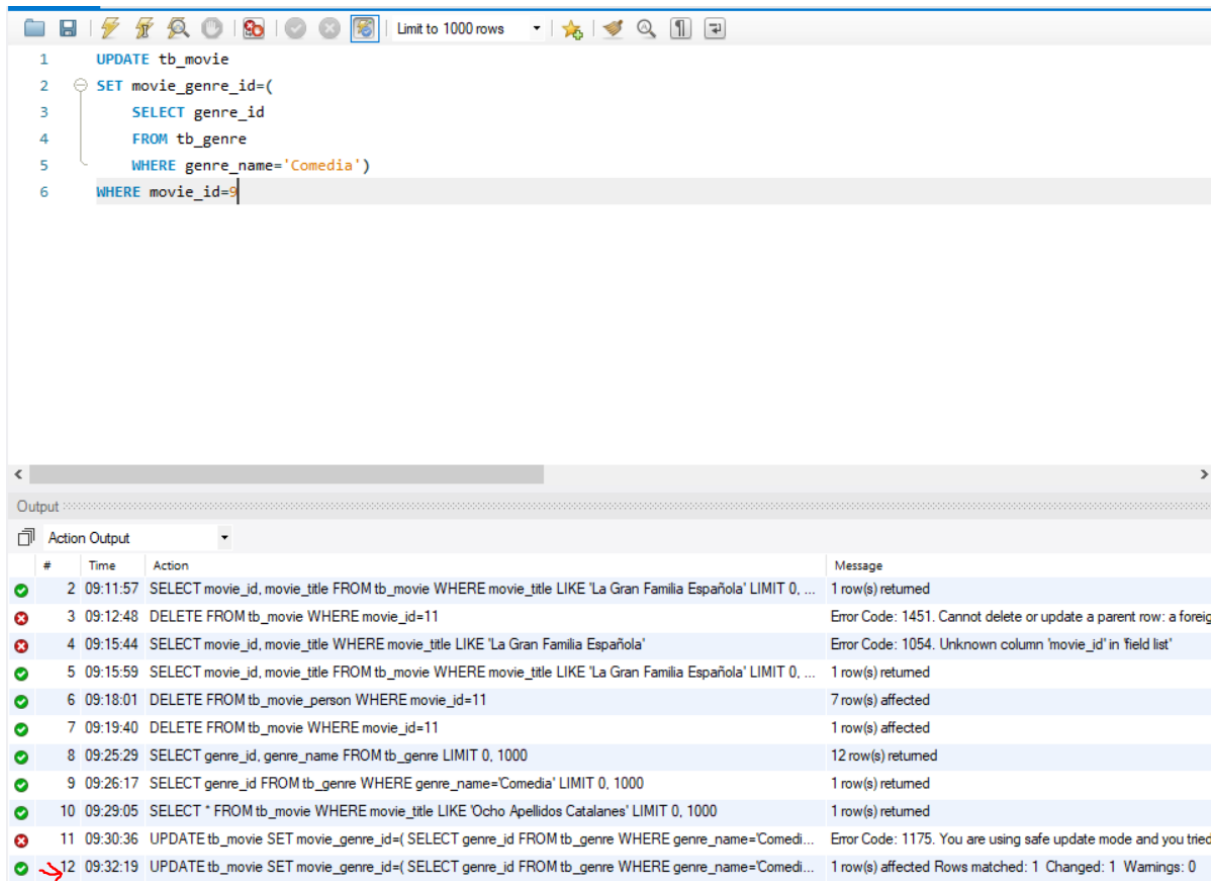
Canvia el gènere de la pel·lícula "Ocho apellidos catalanes" perquè consti com a comèdia i no com a romàntica.

Comencem consultant el registre a modificar:

```
1 • SELECT *  
2 FROM tb_movie  
3 WHERE movie_title LIKE 'Ocho Apellidos Catalanes'
```

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	8	OS_SGAD	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fem un update amb un select anidat a la part del WHERE hem de fer servir el *movie_id* pel safe update mode activat:



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. A dropdown menu shows "Limit to 1000 rows". The SQL editor contains the following query:

```
1 UPDATE tb_movie
2 SET movie_genre_id=(
3     SELECT genre_id
4     FROM tb_genre
5     WHERE genre_name='Comedia')
6 WHERE movie_id=9
```

Below the editor is the "Output" pane, which is set to "Action Output". It displays a log of database actions with columns for line number, time, action, and message.

#	Time	Action	Message
2	09:11:57	SELECT movie_id, movie_title FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española' LIMIT 0, ...	1 row(s) returned
3	09:12:48	DELETE FROM tb_movie WHERE movie_id=11	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails
4	09:15:44	SELECT movie_id, movie_title WHERE movie_title LIKE 'La Gran Familia Española'	Error Code: 1054. Unknown column 'movie_id' in 'field list'
5	09:15:59	SELECT movie_id, movie_title FROM tb_movie WHERE movie_title LIKE 'La Gran Familia Española' LIMIT 0, ...	1 row(s) returned
6	09:18:01	DELETE FROM tb_movie_person WHERE movie_id=11	7 row(s) affected
7	09:19:40	DELETE FROM tb_movie WHERE movie_id=11	1 row(s) affected
8	09:25:29	SELECT genre_id, genre_name FROM tb_genre LIMIT 0, 1000	12 row(s) returned
9	09:26:17	SELECT genre_id FROM tb_genre WHERE genre_name='Comedia' LIMIT 0, 1000	1 row(s) returned
10	09:29:05	SELECT * FROM tb_movie WHERE movie_title LIKE 'Ocho Apellidos Catalanes' LIMIT 0, 1000	1 row(s) returned
11	09:30:36	UPDATE tb_movie SET movie_genre_id=(SELECT genre_id FROM tb_genre WHERE genre_name='Comedia'...	Error Code: 1175. You are using safe update mode and you tried to update a table. No PRIMARY or UNIQUE key was used
12	09:32:19	UPDATE tb_movie SET movie_genre_id=(SELECT genre_id FROM tb_genre WHERE genre_name='Comedia'...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0