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.

A) Características principales del Esquema

La Base de Datos "Movies" consta de cinco tablas, con los campos siguientes:

- tb_genre:

```
      1- genre_id: Índice del género de la película (*) (1,2,3....)
      2- genre_name: Tipo de género de la película. (1. Acción, 2.
Ciencia Ficción, 3. Comedia,etc..)
      3- created_by_user: Creador de la tabla
      4- created_date: Fecha de creación
      5- update_date: Fecha grabación
```

- tb_movie:

```
      1- movie_id: Índice de peliculas (**) (1,2,3...)
      2- movie_title: Título de la película
      3- movie_date: Fecha de la película
      4- movie_format: formato de la película. (Film ó Digital)
      5- movie_genre_id: código del género de la película (*) (1, 2,
3....)
      6- created_by_user: Creador de la tabla
      7- created_date: Fecha de creación
      8- update_date: Fecha grabació
```

- tb_movie_person:

```
      1- movie_id: Índice de peliculas (**)
      2- person_id: Índice de personas (***)
      3- role_id: Índice de roles (****) (1 a 5)
      4- movie_award_ind: premios
      5- created_by_user: Creador de la tabla
      6- created_date: Fecha de creación
      7- update_date: Fecha grabación
```

- tb_person:

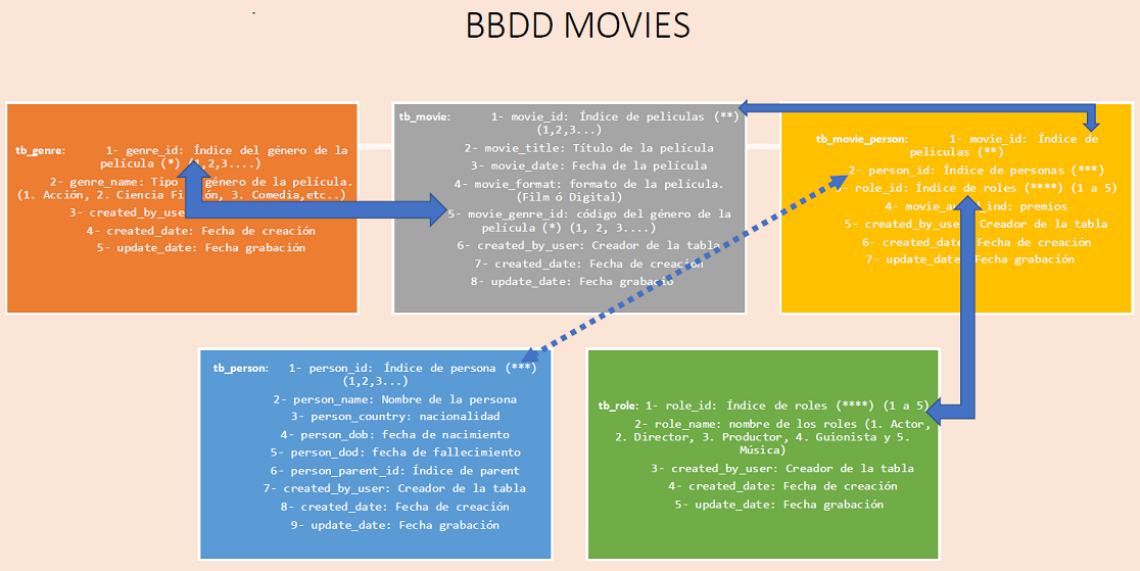
```
      1- person_id:Índice de persona (***) (1,2,3...)
      2- person_name: Nombre de la persona
      3- person_country: nacionalidad
      4- person_dob: fecha de nacimiento
      5- person_dod: fecha de fallecimiento
      6- person_parent_id: Índice de parent
      7- created_by_user: Creador de la tabla
```

```
        8- created_date: Fecha de creación
        9- update_date: Fecha grabación
```

- tb_role:

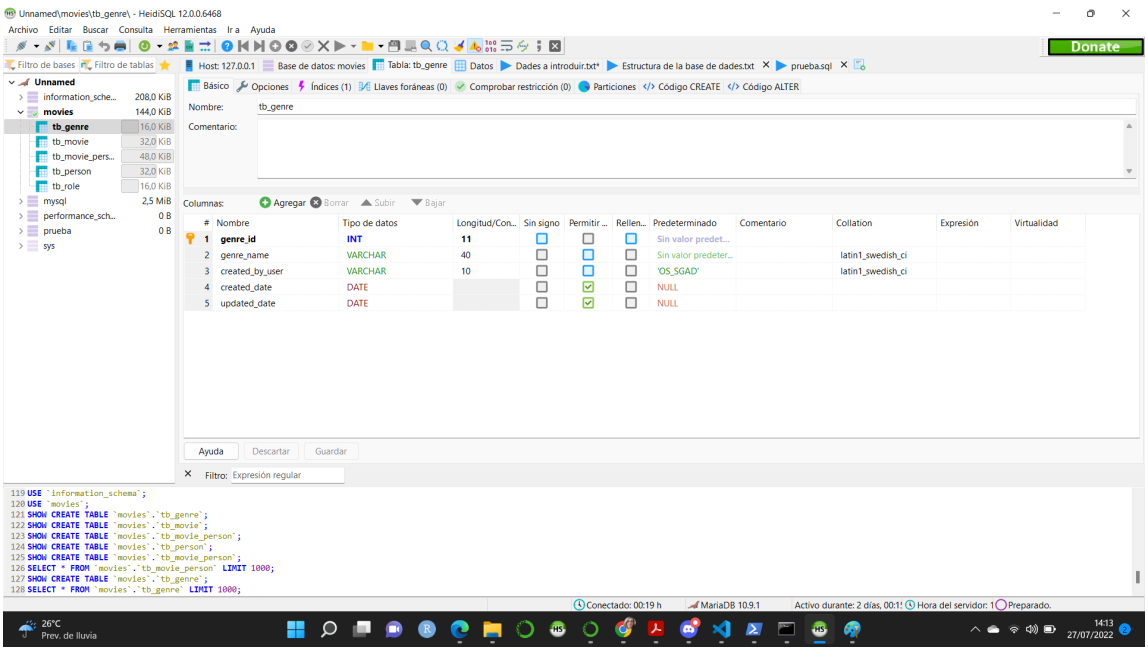
```
        1- role_id: Índice de roles (****) (1 a 5)
        2- role_name: nombre de los roles (1. Actor, 2. Director, 3.
    Productor, 4. Guionista y 5. Música)
        3- created_by_user: Creador de la tabla
        4- created_date: Fecha de creación
        5- update_date: Fecha grabación
```

(*....) Los asteríscos indican las relaciones de dependencia entre las diferentes tablas de la base de datos

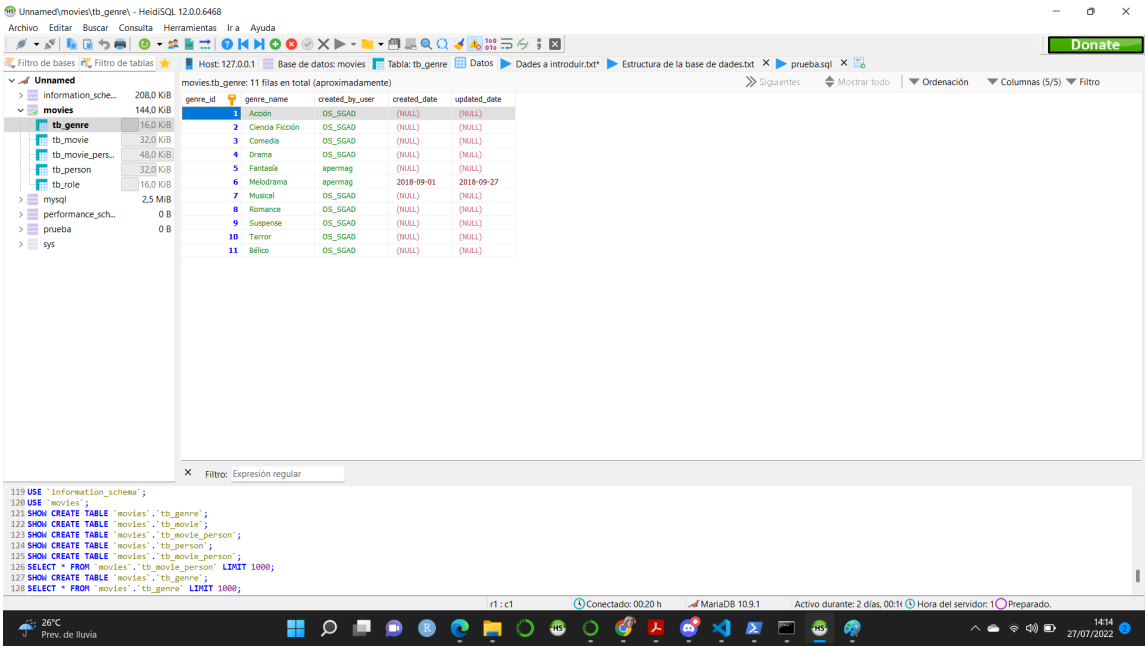


B) Capturas de Pantallas de las diferentes tablas y datos de Movies

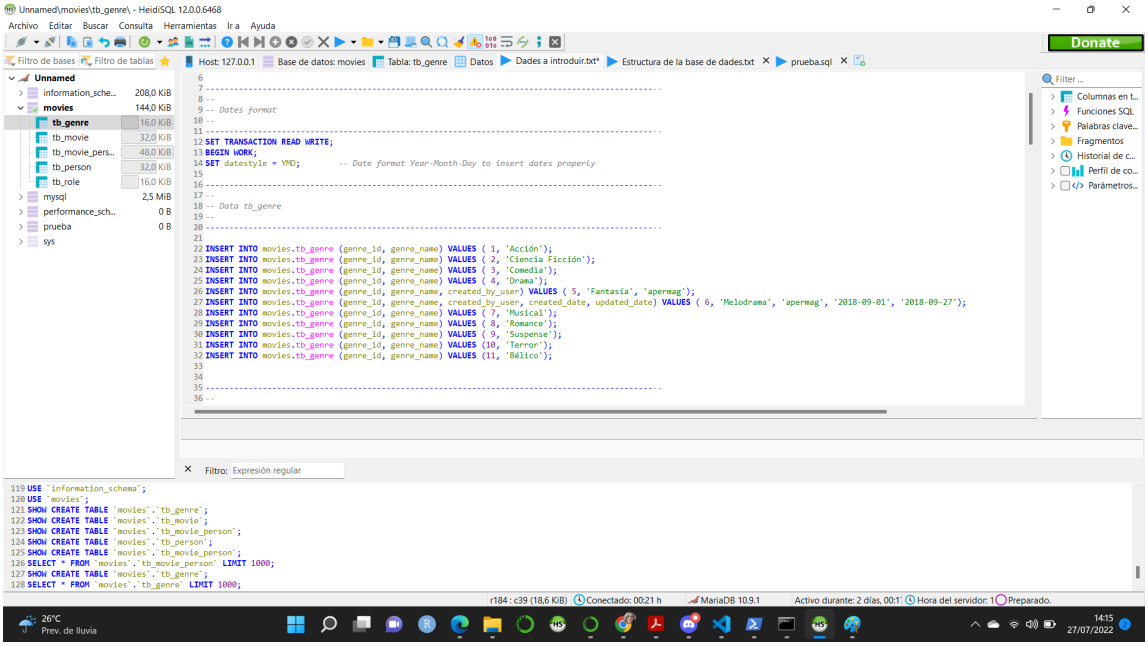
B.1) Tabla de Géneros: Tabla_tb_genre



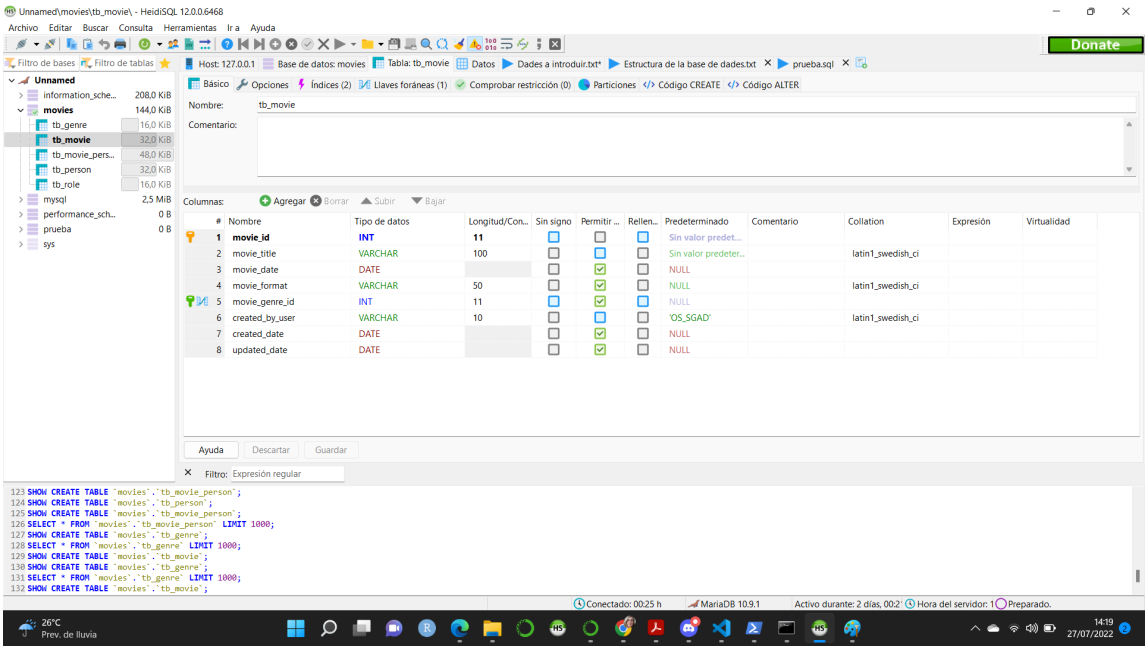
B.1.1) Tabla de Géneros: Datos_tb_genre



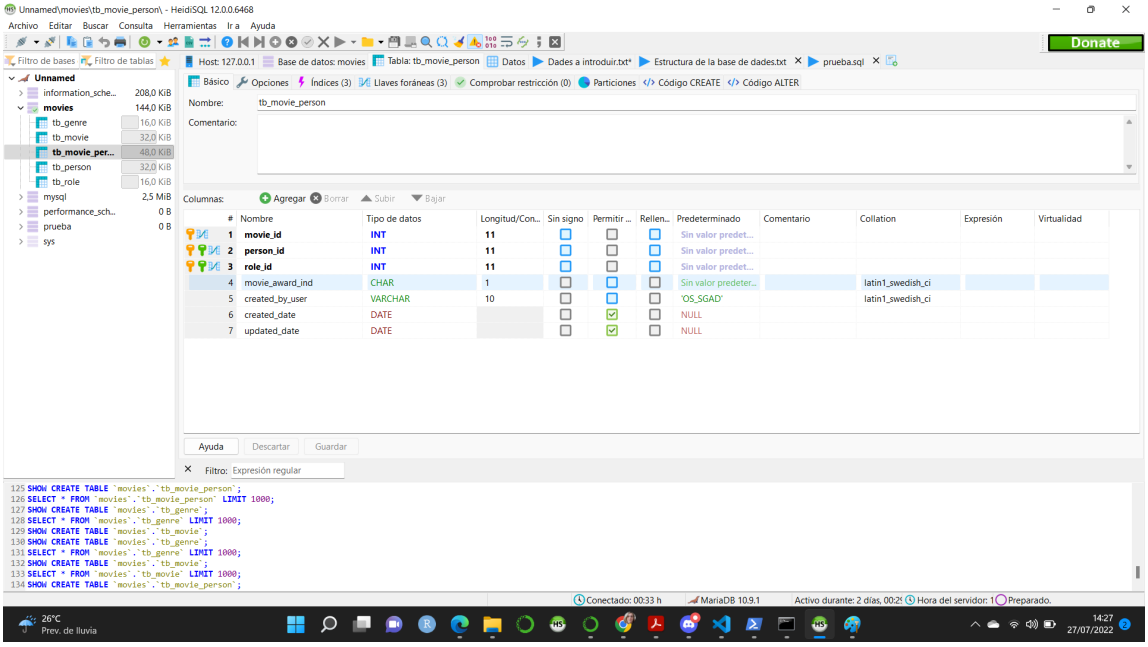
B.1.2) Tabla de Géneros: Dades a introduir.txt_tb_genre



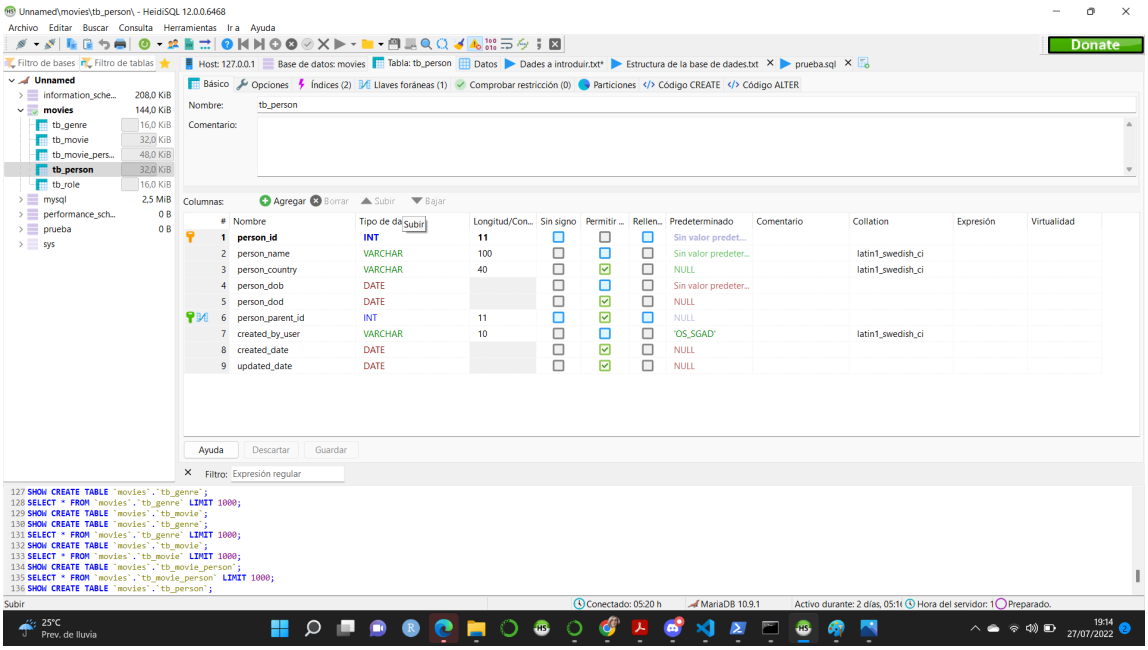
B.2) Tabla de películass: Tabla_tb_movie



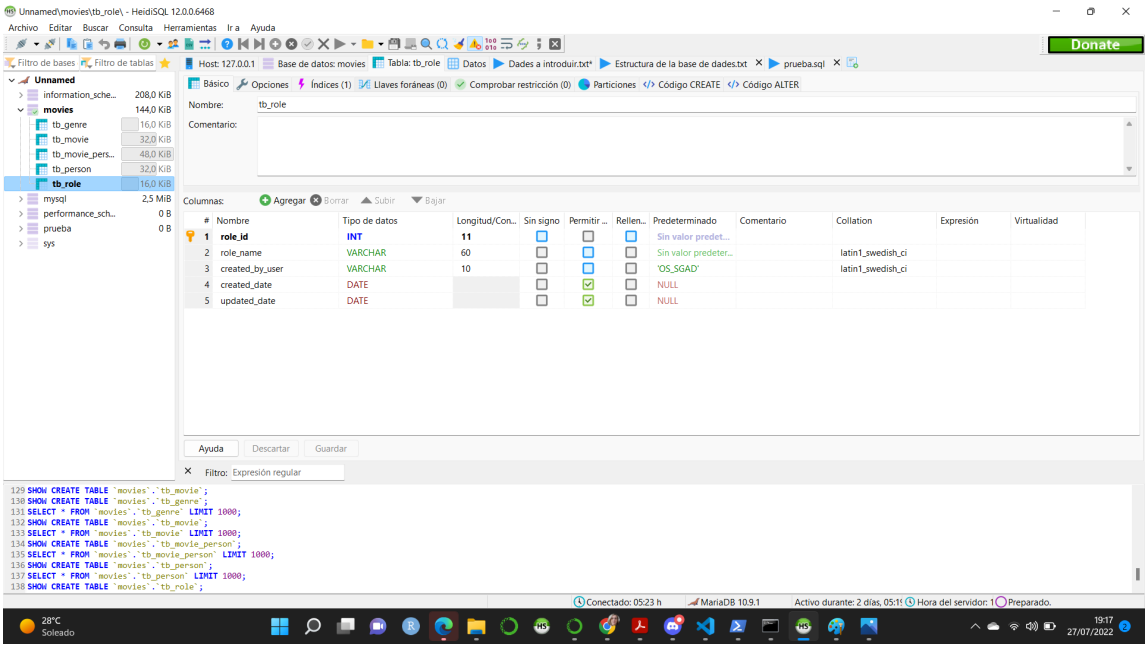
B.3) Tabla de películas y Personas: Tabla_tb_movie_person



B.4) Tabla de Personas: Tabla_tb_person



B.5) Tabla de Roles: Tabla_tb_role



C) Conexión MSQL - Python

In [1]:
```
import mysql.connector
import pandas as pd
import sqlalchemy

# Configuración warnings
# ============================================================================
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="hectorete1"
)
```

```
print(mydb)
```

```
<mysql.connector.connection_cext.CMySQLConnection object at 0x000001F76DBE5EB0>
```

C.1) Tabla de Géneros de Películas

In [3]:
```
engine = sqlalchemy.create_engine('mysql+mysqlconnector://root:hectorete1@localhos
data_genre = pd.read_sql_table("tb_genre", engine)
# check first rows of thcursor=connection.cursor()is table
data_genre
```

Out[3]:

	genre_id	genre_name	created_by_user	created_date	updated_date
0	1	Acción	OS_SGAD	NaT	NaT
1	2	Ciencia Ficción	OS_SGAD	NaT	NaT
2	3	Comedia	OS_SGAD	NaT	NaT
3	4	Drama	OS_SGAD	NaT	NaT
4	5	Fantasía	apermag	NaT	NaT
5	6	Melodrama	apermag	2018-09-01	2018-09-27
6	7	Musical	OS_SGAD	NaT	NaT
7	8	Romance	OS_SGAD	NaT	NaT
8	9	Suspense	OS_SGAD	NaT	NaT
9	10	Terror	OS_SGAD	NaT	NaT
10	11	Bélico	OS_SGAD	NaT	NaT
11	69	Documental	OS_SGAD	NaT	NaT

C.2) Tabla de Películas

In [4]:
```
data_movie = pd.read_sql_table("tb_movie", engine)
# check first rows of this table
data_movie
```

Out[4]:

	movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_user	created
0	1	Apocalypse Now	1979-05-10	Film	11	OS_SGAD	2022-
1	2	Star Wars:Episode IV - A New Hope	1977-05-25	Film	2	OS_SGAD	2022-
2	3	Indiana Jones and the Temple of Doom	1984-05-08	Film	2	OS_SGAD	
3	4	The Terminal	2004-06-18	Digital	3	OS_SGAD	
4	5	Jaws	1975-01-01	Film	10	OS_SGAD	
5	6	ET The Extraterrestrial	1982-07-25	Film	5	OS_SGAD	
6	7	Psycho	1960-05-06	Film	9	OS_SGAD	
7	8	Ocho Apellidos Vascos	2014-03-14	Digital	3	OS_SGAD	
8	9	Ocho Apellidos Catalanes	2016-06-09	Digital	3	OS_SGAD	
9	10	El otro lado de la cama	2002-09-04	Digital	8	OS_SGAD	
10	12	El dia de la bestia	1994-12-25	Film	2	OS_SGAD	
11	13	Braveheart	1995-08-08	Film	4	OS_SGAD	
12	14	The Shawshank Redemption	1992-01-07	Film	4	OS_SGAD	
13	15	Las brujas de Zugarramurdi	2009-10-07	Digital	9	OS_SGAD	
14	16	Blade Runner	1982-12-25	Digital	2	OS_SGAD	

C.3) Tabla de ID (Películas - Personas - Roles)

In [5]:
```
data_movie_person = pd.read_sql_table("tb_movie_person", engine)
# check first rows of this table
data_movie_person
```

Out[5]:

	movie_id	person_id	role_id	movie_award_ind	created_by_user	created_date	updated_date
0	1	1	2	Y	OS_SGAD	NaT	NaT
1	1	1	3	N	OS_SGAD	NaT	NaT
2	1	1	5	N	OS_SGAD	NaT	NaT
3	1	2	5	N	OS_SGAD	NaT	NaT
4	1	3	1	N	OS_SGAD	NaT	NaT
5	1	4	1	N	OS_SGAD	NaT	NaT
6	1	5	1	Y	OS_SGAD	NaT	NaT
7	1	6	1	N	OS_SGAD	NaT	NaT
8	1	41	1	N	OS_SGAD	NaT	NaT
9	2	6	1	N	OS_SGAD	NaT	NaT
10	2	7	2	Y	OS_SGAD	NaT	NaT
11	2	8	3	N	OS_SGAD	NaT	NaT
12	3	6	1	N	OS_SGAD	NaT	NaT
13	3	7	1	N	OS_SGAD	NaT	NaT
14	3	7	4	N	OS_SGAD	NaT	NaT
15	3	9	2	N	OS_SGAD	NaT	NaT
16	3	10	5	N	OS_SGAD	NaT	NaT
17	4	9	2	N	OS_SGAD	NaT	NaT
18	4	9	3	N	OS_SGAD	NaT	NaT
19	4	11	1	N	OS_SGAD	NaT	NaT
20	4	12	1	N	OS_SGAD	NaT	NaT
21	5	9	2	N	OS_SGAD	NaT	NaT
22	6	9	2	N	OS_SGAD	NaT	NaT
23	7	13	1	N	OS_SGAD	NaT	NaT
24	7	13	2	N	OS_SGAD	NaT	NaT
25	7	13	3	N	OS_SGAD	NaT	NaT
26	7	14	2	N	OS_SGAD	NaT	NaT
27	7	15	2	N	OS_SGAD	NaT	NaT
28	8	16	2	N	OS_SGAD	NaT	NaT
29	8	17	1	N	OS_SGAD	NaT	NaT
30	8	18	1	N	OS_SGAD	NaT	NaT
31	8	19	1	N	OS_SGAD	NaT	NaT
32	8	20	1	N	OS_SGAD	NaT	NaT
33	9	16	2	N	OS_SGAD	NaT	NaT
34	9	17	1	N	OS_SGAD	NaT	NaT
35	9	18	1	N	OS_SGAD	NaT	NaT

	movie_id	person_id	role_id	movie_award_ind	created_by_user	created_date	updated_date
36	9	19	1	N	OS_SGAD	NaT	NaT
37	9	20	1	N	OS_SGAD	NaT	NaT
38	10	16	2	N	OS_SGAD	NaT	NaT
39	13	28	1	Y	OS_SGAD	NaT	NaT
40	13	28	2	N	OS_SGAD	NaT	NaT
41	14	29	1	N	OS_SGAD	NaT	NaT
42	14	30	1	N	OS_SGAD	NaT	NaT

C.4) Tabla de Personas

In [6]:
```
data_person = pd.read_sql_table("tb_person", engine)
# check first rows of this table
data_person
```

Out[6]:

	person_id	person_name	person_country	person_dob	person_dod	person_parent_id	created_
0	1	Francis Ford Coppola	United States	1939-04-07	NaT	NaN	C
1	2	Carmine Coppola	United States	1945-07-08	NaT	NaN	C
2	3	Marlon Brando	United States	1924-04-03	2004-07-01	NaN	C
3	4	Robert Duvall	United States	1931-01-05	NaT	NaN	C
4	5	Martin Sheen	United States	1940-08-03	NaT	NaN	C
5	6	Harrison Ford	United States	1942-07-13	NaT	NaN	C
6	7	George Lucas	United States	1944-05-14	NaT	NaN	C
7	8	Gary Kurtz	United States	1940-07-27	NaT	NaN	C
8	9	Steven Spielberg	United States	1946-12-18	NaT	NaN	C
9	10	John Williams	United States	1928-08-08	NaT	NaN	C
10	11	Tom Hanks	United States	1956-07-09	NaT	NaN	C
11	12	Catherine Zeta-Jones	Wales	1969-09-25	NaT	NaN	C
12	13	Alfred Joseph Hitchcock	United Kingdom	1899-08-13	1980-04-29	NaN	C
13	14	Anthony Perkins	United States	1934-04-04	1992-09-08	NaN	C
14	15	Vera Miles	United States	1929-08-23	NaT	NaN	C
15	16	Emilio Martinez Lazaro	Spain	1956-09-09	NaT	NaN	C
16	17	Dani Rovira	Spain	1984-07-01	NaT	NaN	C
17	18	Clara Lago	Spain	1986-04-17	NaT	NaN	C
18	19	Carmen Machi	Spain	1964-08-09	NaT	NaN	C
19	20	Karra Elejalde	Spain	1960-03-06	NaT	NaN	C
20	21	Daniel Sanchez Arevalo	Spain	1970-06-08	NaT	NaN	C
21	22	Quim Gutierrez	Spain	1981-03-27	NaT	NaN	C
22	23	Robert Alamo	Spain	1970-05-06	NaT	NaN	C
23	24	Hector Colome	Spain	1944-10-25	2015-02-28	NaN	C
24	25	Veronica Echegui	Spain	1983-03-14	NaT	NaN	C
25	26	Patrick Criado	Spain	1995-09-23	NaT	NaN	C
26	27	Sean Connery	Scotland	1930-07-08	NaT	NaN	C

	person_id	person_name	person_country	person_dob	person_dod	person_parent_id	created_
27	28	Mel Gibson	Australia	1950-08-09	NaT	NaN	C
28	29	Morgan Freeman	United States	1935-10-01	NaT	NaN	C
29	30	Tim Robbins	United States	1949-06-07	NaT	NaN	C
30	41	Charlie Sheen	United States	1965-09-03	NaT	5.0	C
31	42	Emilio Estevez	United States	1962-05-12	NaT	5.0	C
32	43	Ramón Estevez	United States	1963-08-07	NaT	5.0	C
33	44	Reneé Estevez	United States	1967-04-02	NaT	5.0	C
34	45	Paula Speert Sheen	United States	1986-01-06	NaT	41.0	C
35	46	Bob Sheen	United States	2009-05-01	NaT	41.0	C
36	47	Max Sheen	United States	2009-05-01	NaT	41.0	C
37	48	Sam Sheen	United States	2004-03-09	NaT	41.0	C
38	49	Lola Sheen	United States	2005-06-01	NaT	41.0	C
39	50	Paula Jones-Sheen	United States	2003-07-06	NaT	45.0	C
40	51	Paloma Rae Estevez	United States	1986-02-15	NaT	42.0	C
41	52	Taylor Levi Estevez	United States	1984-06-22	NaT	42.0	C

C.5) Tabla de Roles

In [7]:
```
data_role = pd.read_sql_table("tb_role", engine)
# check first rows of this table
data_role
```

Out[7]:

	role_id	role_name	created_by_user	created_date	updated_date
0	1	Actor	OS_SGAD	NaT	NaT
1	2	Director	OS_SGAD	NaT	NaT
2	3	Productor	OS_SGAD	NaT	NaT
3	4	Guionista	OS_SGAD	NaT	NaT
4	5	Música	OS_SGAD	NaT	NaT

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.

In [8]:
```
query = '''
SELECT person_name, person_country, person_dob,
       timestampdiff(YEAR, person_dob, current_date) as current_ages
FROM tb_person
WHERE person_dod is null
ORDER BY current_ages desc;

'''

current_ages_df = pd.read_sql_query(query, engine)
current_ages_df
```

Out[8]:

	person_name	person_country	person_dob	current_ages
0	John Williams	United States	1928-08-08	93
1	Sean Connery	Scotland	1930-07-08	92
2	Vera Miles	United States	1929-08-23	92
3	Robert Duvall	United States	1931-01-05	91
4	Morgan Freeman	United States	1935-10-01	86
5	Francis Ford Coppola	United States	1939-04-07	83
6	Gary Kurtz	United States	1940-07-27	82
7	Martin Sheen	United States	1940-08-03	81
8	Harrison Ford	United States	1942-07-13	80
9	George Lucas	United States	1944-05-14	78
10	Carmine Coppola	United States	1945-07-08	77
11	Steven Spielberg	United States	1946-12-18	75
12	Tim Robbins	United States	1949-06-07	73
13	Mel Gibson	Australia	1950-08-09	71
14	Tom Hanks	United States	1956-07-09	66
15	Emilio Martinez Lazaro	Spain	1956-09-09	65
16	Karra Elejalde	Spain	1960-03-06	62
17	Emilio Estevez	United States	1962-05-12	60
18	Ramón Estevez	United States	1963-08-07	58
19	Carmen Machi	Spain	1964-08-09	57
20	Charlie Sheen	United States	1965-09-03	56
21	Reneé Estevez	United States	1967-04-02	55
22	Robert Alamo	Spain	1970-05-06	52
23	Daniel Sanchez Arevalo	Spain	1970-06-08	52
24	Catherine Zeta-Jones	Wales	1969-09-25	52
25	Quim Gutierrez	Spain	1981-03-27	41
26	Veronica Echegui	Spain	1983-03-14	39
27	Taylor Levi Estevez	United States	1984-06-22	38
28	Dani Rovira	Spain	1984-07-01	38
29	Paloma Rae Estevez	United States	1986-02-15	36
30	Paula Speert Sheen	United States	1986-01-06	36
31	Clara Lago	Spain	1986-04-17	36
32	Patrick Criado	Spain	1995-09-23	26
33	Paula Jones-Sheen	United States	2003-07-06	19
34	Sam Sheen	United States	2004-03-09	18
35	Lola Sheen	United States	2005-06-01	17

	person_name	person_country	person_dob	current_ages
36	Bob Sheen	United States	2009-05-01	13
37	Max Sheen	United States	2009-05-01	13

Exercici 3

Realitza la següent consulta :

- 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.

In [9]:
```
query = '''
SELECT genre_name, COUNT(*) movie_count

FROM tb_movie
INNER JOIN tb_genre
ON
tb_movie.movie_genre_id = tb_genre.genre_id
GROUP BY genre_name
ORDER BY movie_count desc;

'''

join_df = pd.read_sql_query(query, engine)
join_df.head(11)
```

Out[9]:

	genre_name	movie_count
0	Ciencia Ficción	4
1	Comedia	3
2	Suspense	2
3	Drama	2
4	Romance	1
5	Terror	1
6	Bélico	1
7	Fantasía	1

Exercici 4

Realitza la següent consulta:

- 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.

In [10]:
```
query = '''
SELECT person_name, role_name, movie_title
```

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

ORDER BY person_name asc;

'''

join2_df = pd.read_sql_query(query, engine)
join2_df
```

Out[10]:

	person_name	role_name	movie_title
0	Alfred Joseph Hitchcock	Productor	Psycho
1	Alfred Joseph Hitchcock	Actor	Psycho
2	Alfred Joseph Hitchcock	Director	Psycho
3	Anthony Perkins	Director	Psycho
4	Carmen Machi	Actor	Ocho Apellidos Catalanes
5	Carmen Machi	Actor	Ocho Apellidos Vascos
6	Carmine Coppola	Música	Apocalypse Now
7	Catherine Zeta-Jones	Actor	The Terminal
8	Charlie Sheen	Actor	Apocalypse Now
9	Clara Lago	Actor	Ocho Apellidos Catalanes
10	Clara Lago	Actor	Ocho Apellidos Vascos
11	Dani Rovira	Actor	Ocho Apellidos Catalanes
12	Dani Rovira	Actor	Ocho Apellidos Vascos
13	Emilio Martinez Lazaro	Director	Ocho Apellidos Vascos
14	Emilio Martinez Lazaro	Director	Ocho Apellidos Catalanes
15	Emilio Martinez Lazaro	Director	El otro lado de la cama
16	Francis Ford Coppola	Productor	Apocalypse Now
17	Francis Ford Coppola	Director	Apocalypse Now
18	Francis Ford Coppola	Música	Apocalypse Now
19	Gary Kurtz	Productor	Star Wars:Episode IV - A New Hope
20	George Lucas	Director	Star Wars:Episode IV - A New Hope
21	George Lucas	Guionista	Indiana Jones and the Temple of Doom
22	George Lucas	Actor	Indiana Jones and the Temple of Doom
23	Harrison Ford	Actor	Apocalypse Now
24	Harrison Ford	Actor	Star Wars:Episode IV - A New Hope
25	Harrison Ford	Actor	Indiana Jones and the Temple of Doom
26	John Williams	Música	Indiana Jones and the Temple of Doom
27	Karra Elejalde	Actor	Ocho Apellidos Vascos
28	Karra Elejalde	Actor	Ocho Apellidos Catalanes
29	Marlon Brando	Actor	Apocalypse Now
30	Martin Sheen	Actor	Apocalypse Now
31	Mel Gibson	Director	Braveheart
32	Mel Gibson	Actor	Braveheart
33	Morgan Freeman	Actor	The Shawshank Redemption
34	Robert Duvall	Actor	Apocalypse Now
35	Steven Spielberg	Productor	The Terminal

	person_name	role_name	movie_title
36	Steven Spielberg	Director	Indiana Jones and the Temple of Doom
37	Steven Spielberg	Director	The Terminal
38	Steven Spielberg	Director	Jaws
39	Steven Spielberg	Director	ET The Extraterrestrial
40	Tim Robbins	Actor	The Shawshank Redemption
41	Tom Hanks	Actor	The Terminal
42	Vera Miles	Director	Psycho

- En la salida figura Anthony Perkins como Director en Psycho, cuando el rol fué Actor, pero hemos verificado que el código de rol correspondiente al person_id 14 en la tabla de movie_person es: 2, que corresponde al de Director, así que es un error de codificación en tb_movie_person original.

In [11]:
```
query = '''
SELECT movie_title, person_name, COUNT(*) role_count

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


GROUP BY movie_title, person_name
HAVING role_count >1
ORDER BY movie_title asc

;


'''

join2_df = pd.read_sql_query(query, engine)
join2_df
```

Out[11]:

	movie_title	person_name	role_count
0	Apocalypse Now	Francis Ford Coppola	3
1	Braveheart	Mel Gibson	2
2	Indiana Jones and the Temple of Doom	George Lucas	2
3	Psycho	Alfred Joseph Hitchcock	3
4	The Terminal	Steven Spielberg	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.

In [12]:
```
# Import pymysql module
#import # Import pymysql module
import pymysql
```

In [13]:
```
# Connect to the database
connection = pymysql.connect(host='localhost',
                             user='root',
                             password='hectorete1',
                             db='movies')
```

In [14]:
```
cursor = connection.cursor()
```

In [15]:
```
sql = "INSERT INTO `movies.tb_genre` (genre_id, genre_name) VALUES (69, 'Documental


# the connection is not autocommited by default. So we must commit to save our char
connection.commit()
```

In [16]:
```
engine = sqlalchemy.create_engine('mysql+mysqlconnector://root:hectorete1@localhost
data_genre = pd.read_sql_table("tb_genre", engine)
# check first rows of thcursor=connection.cursor()is table
data_genre
```

Out[16]:

	genre_id	genre_name	created_by_user	created_date	updated_date
0	1	Acción	OS_SGAD	NaT	NaT
1	2	Ciencia Ficción	OS_SGAD	NaT	NaT
2	3	Comedia	OS_SGAD	NaT	NaT
3	4	Drama	OS_SGAD	NaT	NaT
4	5	Fantasía	apermag	NaT	NaT
5	6	Melodrama	apermag	2018-09-01	2018-09-27
6	7	Musical	OS_SGAD	NaT	NaT
7	8	Romance	OS_SGAD	NaT	NaT
8	9	Suspense	OS_SGAD	NaT	NaT
9	10	Terror	OS_SGAD	NaT	NaT
10	11	Bélico	OS_SGAD	NaT	NaT
11	69	Documental	OS_SGAD	NaT	NaT

In [17]:
```
# Close the connection
connection.close()
```

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.

In [18]:
```
import mysql.connector

mydb = mysql.connector.connect(host="localhost",
                               user="root",
                               password="hectorete1",
                               database="movies")
```

In [19]:
```
mycursor = mydb.cursor()
```

In [20]:
```
sql=" ALTER TABLE `tb_movie_person` DROP CONSTRAINT `fk_movper_movie`;"
```

In [21]:
```
mycursor.execute(sql)
```

In [22]:
```
sql=" ALTER TABLE `tb_movie_person` ADD CONSTRAINT `fk_movper_movie` FOREIGN KEY (r
```

In [23]:
```
mycursor.execute(sql)
```

In [24]:
```
sql ="DELETE FROM tb_movie WHERE tb_movie.movie_title='La Gran Familia Española'"
```

In [25]:
```
mycursor.execute(sql)
```

In [26]:
```
# Close the connection
mydb.close()
```

In [27]:
```
data_movie = pd.read_sql_table("tb_movie", engine)
# check first rows of this table
data_movie
```

Out[27]:

	movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_user	created
0	1	Apocalypse Now	1979-05-10	Film	11	OS_SGAD	2022-
1	2	Star Wars:Episode IV - A New Hope	1977-05-25	Film	2	OS_SGAD	2022-
2	3	Indiana Jones and the Temple of Doom	1984-05-08	Film	2	OS_SGAD	
3	4	The Terminal	2004-06-18	Digital	3	OS_SGAD	
4	5	Jaws	1975-01-01	Film	10	OS_SGAD	
5	6	ET The Extraterrestrial	1982-07-25	Film	5	OS_SGAD	
6	7	Psycho	1960-05-06	Film	9	OS_SGAD	
7	8	Ocho Apellidos Vascos	2014-03-14	Digital	3	OS_SGAD	
8	9	Ocho Apellidos Catalanes	2016-06-09	Digital	3	OS_SGAD	
9	10	El otro lado de la cama	2002-09-04	Digital	8	OS_SGAD	
10	12	El dia de la bestia	1994-12-25	Film	2	OS_SGAD	
11	13	Braveheart	1995-08-08	Film	4	OS_SGAD	
12	14	The Shawshank Redemption	1992-01-07	Film	4	OS_SGAD	
13	15	Las brujas de Zugarramurdi	2009-10-07	Digital	9	OS_SGAD	
14	16	Blade Runner	1982-12-25	Digital	2	OS_SGAD	

- Se ha eliminado la película 11: "La Gran Familia Española"

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.

In [28]:
```
mydb = mysql.connector.connect(host="localhost",
                               user="root",
                               password="hectorete1",
                               database="movies")
```

```
mycursor = mydb.cursor()

sql = "UPDATE tb_movie SET  movie_genre_id= 3 WHERE  movie_title='Ocho Apellidos Ca

mycursor.execute(sql)

mydb.commit()

print(mycursor.rowcount, "registros afectado/s")
```

```
0 registros afectado/s
```

In [29]:
```
data_movie = pd.read_sql_table("tb_movie", engine)
# check first rows of this table
data_movie
```

Out[29]:

	movie_id	movie_title	movie_date	movie_format	movie_genre_id	created_by_user	created
0	1	Apocalypse Now	1979-05-10	Film	11	OS_SGAD	2022-
1	2	Star Wars:Episode IV - A New Hope	1977-05-25	Film	2	OS_SGAD	2022-
2	3	Indiana Jones and the Temple of Doom	1984-05-08	Film	2	OS_SGAD	
3	4	The Terminal	2004-06-18	Digital	3	OS_SGAD	
4	5	Jaws	1975-01-01	Film	10	OS_SGAD	
5	6	ET The Extraterrestrial	1982-07-25	Film	5	OS_SGAD	
6	7	Psycho	1960-05-06	Film	9	OS_SGAD	
7	8	Ocho Apellidos Vascos	2014-03-14	Digital	3	OS_SGAD	
8	9	Ocho Apellidos Catalanes	2016-06-09	Digital	3	OS_SGAD	
9	10	El otro lado de la cama	2002-09-04	Digital	8	OS_SGAD	
10	12	El dia de la bestia	1994-12-25	Film	2	OS_SGAD	
11	13	Braveheart	1995-08-08	Film	4	OS_SGAD	
12	14	The Shawshank Redemption	1992-01-07	Film	4	OS_SGAD	
13	15	Las brujas de Zugarramurdi	2009-10-07	Digital	9	OS_SGAD	
14	16	Blade Runner	1982-12-25	Digital	2	OS_SGAD	

In [30]:
```
query = '''
SELECT movie_title,genre_name

FROM tb_movie
INNER JOIN tb_genre
ON
tb_movie.movie_genre_id = tb_genre.genre_id

ORDER BY movie_title asc;

'''

join_df = pd.read_sql_query(query, engine)
join_df
```

Out[30]:

	movie_title	genre_name
0	Apocalypse Now	Bélico
1	Blade Runner	Ciencia Ficción
2	Braveheart	Drama
3	El dia de la bestia	Ciencia Ficción
4	El otro lado de la cama	Romance
5	ET The Extraterrestrial	Fantasía
6	Indiana Jones and the Temple of Doom	Ciencia Ficción
7	Jaws	Terror
8	Las brujas de Zugarramurdi	Suspense
9	Ocho Apellidos Catalanes	Comedia
10	Ocho Apellidos Vascos	Comedia
11	Psycho	Suspense
12	Star Wars:Episode IV - A New Hope	Ciencia Ficción
13	The Shawshank Redemption	Drama
14	The Terminal	Comedia