Consultas SQL a la base de datos IMBD

Valdez Miranda Elias

Ingeniería en Sistemas de Información, Universidad de Sonora

Clave 4116: Bases de Datos I

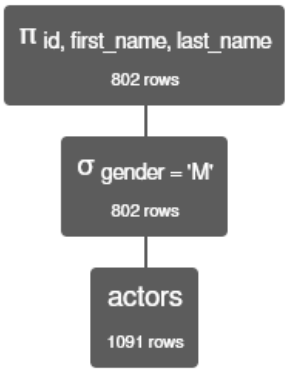
Profesor Navarro Hernández Rene Francisco

14 de septiembre de 2025

Consultas SQL a la base de datos IMBD

1.- Id, nombre y apellido de actores de género masculino

```
SELECT DISTINCT
        id,
        first_name,
        last_name
FROM actors
WHERE gender = 'M';
-- Se omiten 79 páginas de resultados
```



π id, first_name, last_name σ gender = 'M' actors

Execution time: 0 ms

actors.id	actors.first_name	actors.last_name
10963	'Chris'	'Anastasio'
32638	'Michael'	'Beach'
33949	'John'	'Bedford Lloyd'
42278	'Michael'	'Biehn'
57051	'Captain Kidd'	'Brewer Jr.'
64610	'Leo'	'Burmester'
69977	'Mike (I)'	'Cameron'
70419	'J. Kenneth'	'Campbell'
81503	'Michael (I)'	'Chapman'
108530	'Phillip'	'Darlington'

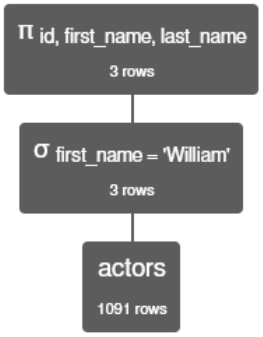
‹ 1 2 3 ›

actors.id	actors.first_name	actors.last_name
473634	'Lawrence'	'Tierney'
518086	'Steven (I)'	'Wright'

‹ 79 80 81 ›

2.- Id, nombre y apellido de actores de nombre ‘William’

```
SELECT DISTINCT
        id,
        first_name,
        last_name
FROM actors
WHERE first_name = 'William';
```



π id, first_name, last_name σ first_name = 'William' actors

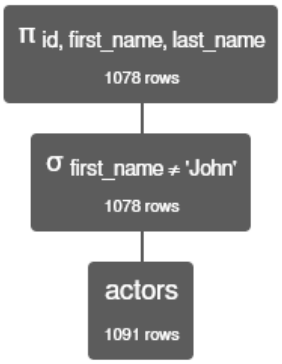
Execution time: 0 ms

actors.id	actors.first_name	actors.last_name
514734	'William'	'Wisher Jr.'
213646	'William'	'Hope'
462532	'William'	'Sylvester'

‹ 1 ›

3.- Id, nombre y apellido de actores que no se llaman 'John'

```
SELECT DISTINCT
        id,
        first_name,
        last_name
FROM actors
WHERE first_name != 'John';
```



π id, first_name, last_name σ first_name ≠ 'John' actors

Execution time: 0 ms

-- Se omiten 106 páginas de resultados

actors.id	actors.first_name	actors.last_name
10963	'Chris'	'Anastasio'
32638	'Michael'	'Beach'
42278	'Michael'	'Biehn'
57051	'Captain Kidd'	'Brewer Jr.'
64610	'Leo'	'Burmester'
69977	'Mike (I)'	'Cameron'
70419	'J. Kenneth'	'Campbell'
81503	'Michael (I)'	'Chapman'
108530	'Phillip'	'Darlington'
130508	'Thomas F.'	'Duffy'

‹ 1 2 3 ›

actors.id	actors.first_name	actors.last_name
368399	'Chris (I)'	'Penn'
447813	'Michael'	'Sottile'
452774	'David (I)'	'Steen'
473634	'Lawrence'	'Tierney'
518086	'Steven (I)'	'Wright'
575999	'Suzanne'	'Celeste'
688902	'Laurie'	'Latham'
803695	'Maria'	'Strova'

‹ 106 107 108 ›

4.- Nombre y calificación de las películas con ranking menor a 6

```
SELECT DISTINCT
    name,
    rank
FROM movies
WHERE rank < 6;
```

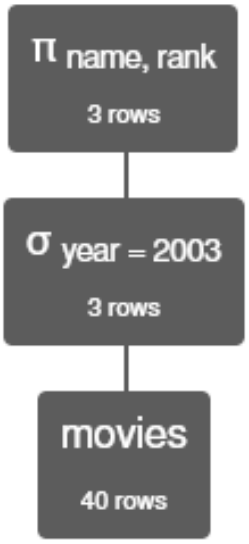
π name, rank σ rank < 6 movies

Execution time: 0 ms

movies.name	movies.rank
'Earthship.TV'	5.6
'Piranha Part Two: The Spawning'	2.8
'Xenogenesis'	3.9
'Flying Padre'	5.9
'Four Rooms'	5.9
'My Best Friend's Birthday'	3.9

5.- Nombres y ranking de todas las películas filmadas en el año 2003

```
SELECT DISTINCT
        name,
        rank
FROM movies
WHERE year = 2003;
```



π name, rank σ year = 2003 movies

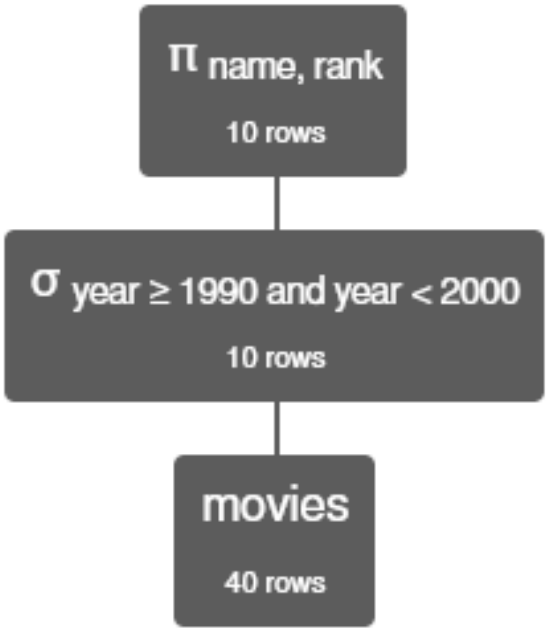
Execution time: 0 ms

movies.name	movies.rank
'Ghosts of the Abyss'	6.7
'Kill Bill: Vol. 1'	8.4
'Jimmy Kimmel Live!'	6.7

‹ 1 ›

6.- Nombres y ranking de todas las películas de la década de los 90s

```
SELECT DISTINCT
    name,
    rank
FROM movies
WHERE year >= 1990 AND year < 2000;
```



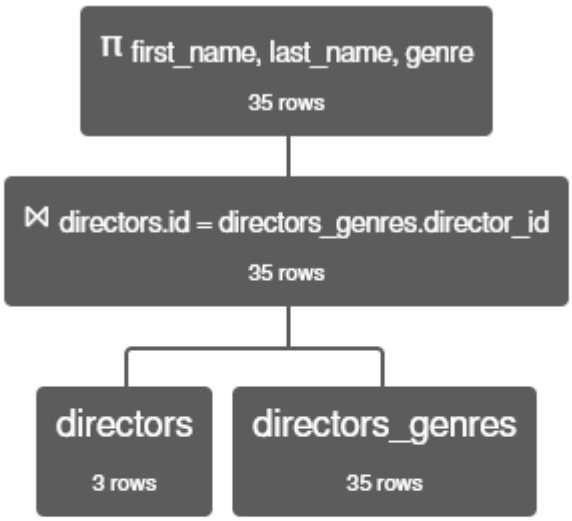
π name, rank σ year ≥ 1990 and year < 2000 movies

Execution time: 0 ms

movies.name	movies.rank
'T2 3-D: Battle Across Time'	7.4
'Terminator 2: Judgment Day'	8.1
'Titanic'	6.9
'True Lies'	7
'Eyes Wide Shut'	7
'Four Rooms'	5.9
'Jackie Brown'	7.5
'Pulp Fiction'	8.7
'Reservoir Dogs'	8.3
'ER'	7.7

7.- Nombres de los directores y los géneros de sus películas.

```
SELECT DISTINCT
        first_name,
        last_name,
        genre
FROM directors
JOIN directors_genres ON directors.id = directors_genres.director_id;
```



π first_name, last_name, genre (directors ⋈ directors.id = directors_genres.director_id directors_genres)

Execution time: 0 ms

directors.first_name	directors.last_name	directors_genres.genre
'James (I)'	'Cameron'	'Action'
'James (I)'	'Cameron'	'Adventure'
'James (I)'	'Cameron'	'Comedy'
'James (I)'	'Cameron'	'Documentary'
'James (I)'	'Cameron'	'Drama'
'James (I)'	'Cameron'	'Family'
'James (I)'	'Cameron'	'Fantasy'
'James (I)'	'Cameron'	'Horror'
'James (I)'	'Cameron'	'Romance'
'James (I)'	'Cameron'	'Sci-Fi'

‹ 1 2 3 ›

directors.first_name	directors.last_name	directors_genres.genre
'James (I)'	'Cameron'	'Short'
'James (I)'	'Cameron'	'Thriller'
'Stanley'	'Kubrick'	'Action'
'Stanley'	'Kubrick'	'Adventure'
'Stanley'	'Kubrick'	'Comedy'
'Stanley'	'Kubrick'	'Crime'
'Stanley'	'Kubrick'	'Documentary'
'Stanley'	'Kubrick'	'Drama'
'Stanley'	'Kubrick'	'Film-Noir'
'Stanley'	'Kubrick'	'Horror'

‹ 1 2 3 4 ›

directors.first_name	directors.last_name	directors_genres.genre
'Stanley'	'Kubrick'	'Music'
'Stanley'	'Kubrick'	'Mystery'
'Stanley'	'Kubrick'	'Romance'
'Stanley'	'Kubrick'	'Sci-Fi'
'Stanley'	'Kubrick'	'Short'
'Stanley'	'Kubrick'	'Thriller'
'Stanley'	'Kubrick'	'War'
'Quentin'	'Tarantino'	'Action'
'Quentin'	'Tarantino'	'Comedy'
'Quentin'	'Tarantino'	'Crime'

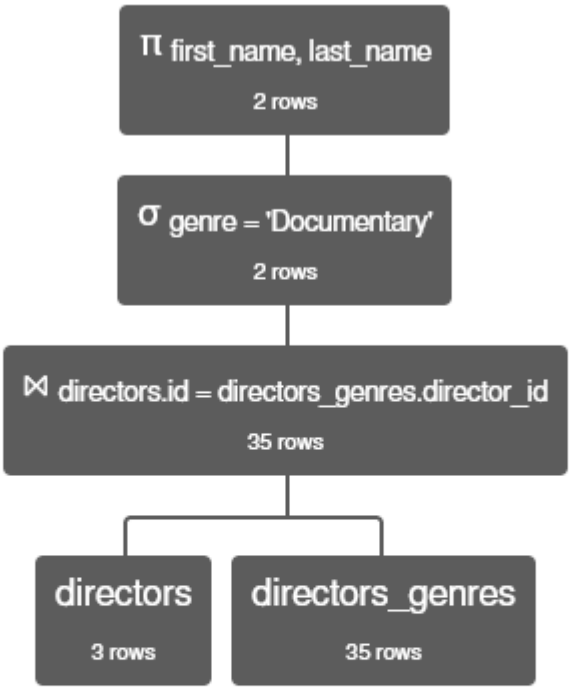
‹ 1 2 3 4 ›

directors.first_name	directors.last_name	directors_genres.genre
'Quentin'	'Tarantino'	'Drama'
'Quentin'	'Tarantino'	'Mystery'
'Quentin'	'Tarantino'	'Romance'
'Quentin'	'Tarantino'	'Thriller'
'Quentin'	'Tarantino'	'War'

‹ 2 3 4 ›

8.- Nombres de los directores que han dirigido películas del género documental.

```
SELECT DISTINCT
        first_name,
        last_name
FROM directors
JOIN directors_genres ON directors.id = directors_genres.director_id
WHERE genre='Documentary';
```



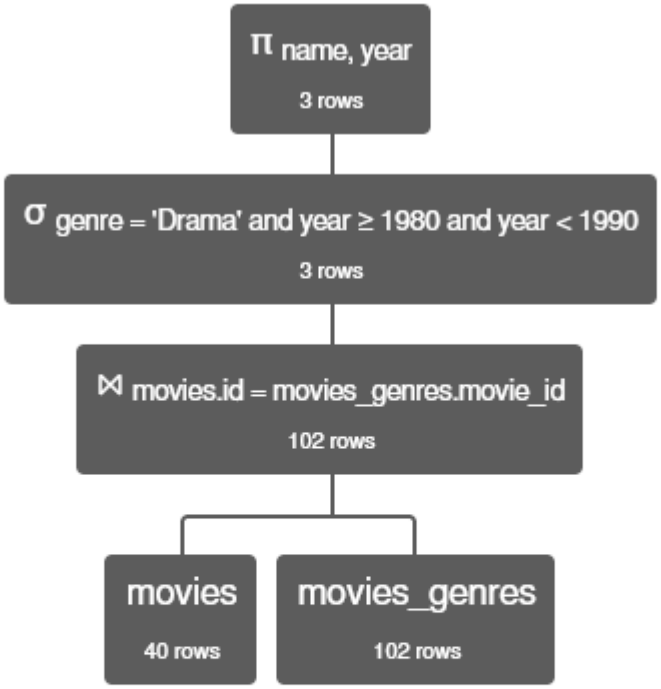
π first_name, last_name σ genre = 'Documentary' (directors ⋈ directors.id = directors_genres.director_id directors_genres)

Execution time: 0 ms

directors.first_name	directors.last_name
'James (I)'	'Cameron'
'Stanley'	'Kubrick'

9.- Nombres de películas del género drama producidas en los 80s

```
SELECT DISTINCT
    name,
    year
FROM movies
JOIN movies_genres ON movies.id = movies_genres.movie_id
WHERE genre='Drama' AND year >= 1980 AND year < 1990;
```



π name, year σ genre = 'Drama' and year ≥ 1980 and year < 1990 (movies ⋈ movies.id = movies_genres.movie_id movies_genres)

Execution time: 0 ms

movies.name	movies.year
'Abyss, The'	1989
'Full Metal Jacket'	1987
'My Best Friend's Birthday'	1987

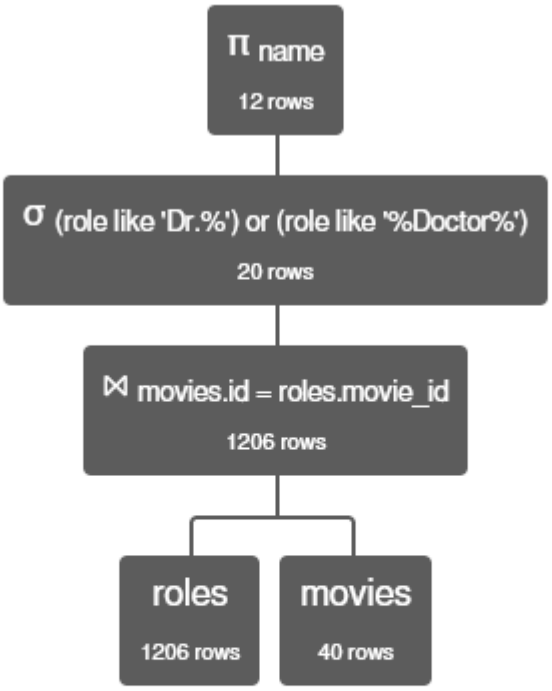
10.- Encontrar los nombres de películas en las que hay un 'Doctor' en el reparto.

```
SELECT DISTINCT name

FROM roles

JOIN movies ON movies.id = roles.movie_id

WHERE (role LIKE 'Dr.%') OR (role LIKE '%Doctor%');
```



π name σ (role like 'Dr.%') or (role like '%Doctor%') (roles ⋈ movies.id = roles.movie_id movies)

Execution time: 0 ms

movies.name
'Abyss, The'
'Aliens'
'T2 3-D: Battle Across Time'
'Terminator 2: Judgment Day'
'Terminator, The'
'2001: A Space Odyssey'
'Barry Lyndon'
'Clockwork Orange, A'
'Eyes Wide Shut'
'Lolita'

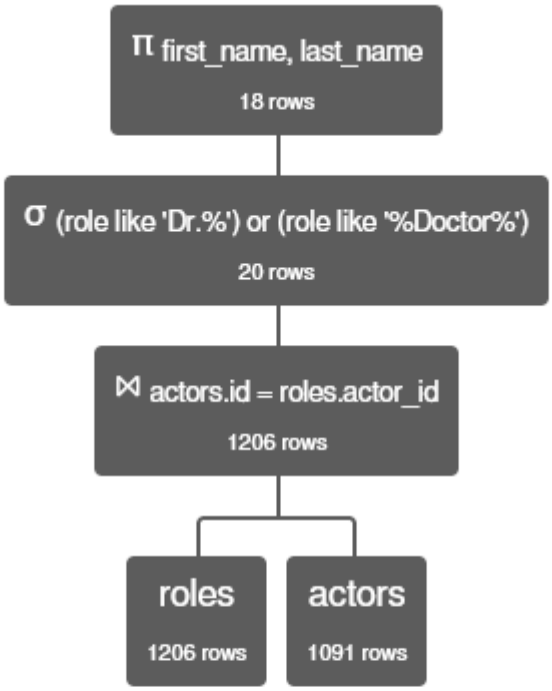
‹ 1 2 ›

movies.name
'Paths of Glory'
'Shining, The'

‹ 1 2 ›

11.- Encontrar los actores que personifican a un 'Doctor' en alguna película.

```
SELECT DISTINCT
        first_name,
        last_name
FROM roles
JOIN actors ON actors.id = roles.actor_id
WHERE (role LIKE 'Dr.%') OR (role LIKE '%Doctor%');
```



π first_name, last_name σ (role like 'Dr.%') or (role like '%Doctor%') (roles ⋈ actors.id = roles.actor_id actors)

Execution time: 0 ms

actors.first_name	actors.last_name
'Michael (I)'	'Chapman'
'Blain'	'Fairman'
'Earl'	'Boen'
'Joe'	'Morton'
'Robert (I)'	'Beatty'
'Keir'	'Dullea'
'Gary'	'Lockwood'
'Leonard'	'Rossiter'
'Sean (I)'	'Sullivan'
'William'	'Sylvester'

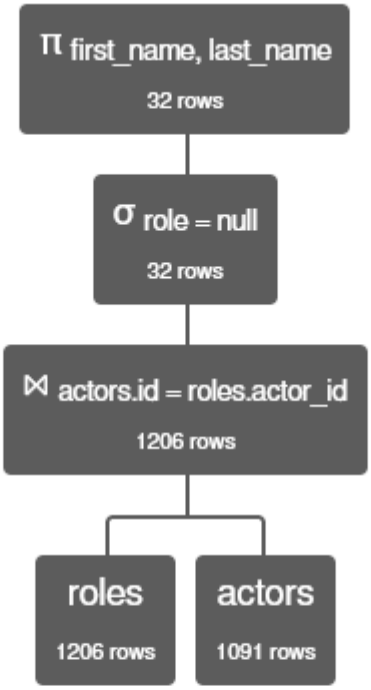
‹ 1 2 ›

actors.first_name	actors.last_name
'Geoffrey'	'Chater'
'Carl'	'Duering'
'Craig (I)'	'Hunter'
'Madge'	'Ryan'
'Tom'	'Cruise'
'Cec'	'Linder'
'Halder'	'Hanson'
'Anne'	'Jackson'

‹ 1 2 ›

12.- Nombres de actores o actrices que aparecen en alguna película, pero no se les dio crédito.

```
SELECT DISTINCT
    first_name,
    last_name
FROM roles
JOIN actors ON actors.id = roles.actor_id
WHERE role IS NULL;
```



π first_name, last_name σ role = null (roles ⋈ actors.id = roles.actor_id actors)

Execution time: 0 ms

actors.first_name	actors.last_name
'Stacey'	'Hayes'
'Sheraton'	'Blount'
'Ann'	'Bormann'
'Julie'	'Croft'
'Penny'	'Francis'
'Jane'	'Hayward'
'Marcella'	'Markham'
'Kim'	'Neil'
'Jane'	'Pearl'
'Penny'	'Pearl'

‹ 1 2 3 ›

actors.first_name	actors.last_name
'Anthony'	'Dawes'
'Patrick'	'Dawson'
'Bernard'	'Hepton'
'Anthony'	'Herrick'
'Barry (I)'	'Jackson'
'Patrick'	'Laffan'
'Hans (I)'	'Meyer'
'Liam'	'Redmond'
'Frederick'	'Schiller'
'Roy'	'Spencer'

‹ 1 2 3 4 ›

actors.first_name	actors.last_name
'John (I)'	'Sullivan'
'Harry'	'Towb'
'Peter'	'Burton'
'Barrie'	'Cookson'
'Heather'	'Carter-Drake'
'Arthur'	'Feldman'
'Bill'	'Funaro'
'Shaun'	'O'Brien'
'Barbara'	'Brand'
'Roy'	'Engel'

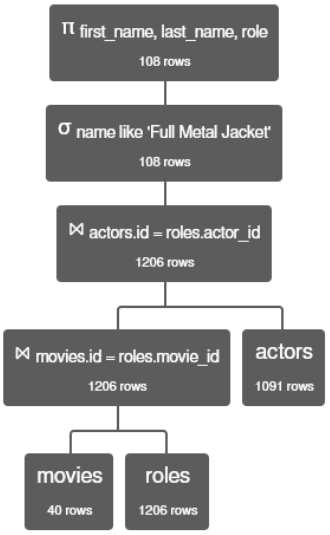
‹ 1 2 3 4 ›

actors.first_name	actors.last_name
'Richard'	'Farnsworth'
'Gordon (I)'	'Mitchell'

‹ 2 3 4 ›

13.- Mostrar el reparto (nombre y rol) de la película “Full Metal Jacket”.

```
SELECT DISTINCT
        first_name,
        last_name,
        role
FROM movies
JOIN roles ON movies.id = roles.movie_id
JOIN actors ON actors.id = roles.actor_id
WHERE name LIKE 'Full Metal Jacket';
-- Se omiten 9 páginas de resultados
```



π first_name, last_name, role σ name like 'Full Metal Jacket' ((movies ⋈ movies.id = roles.movie_id roles) ⋈ actors.id = roles.actor_id actors)

Execution time: 0 ms

actors.first_name	actors.last_name	roles.role
'Martin (I)'	'Adams'	'Marine'
'Kevin'	'Albridge'	'Marine'
'Del'	'Anderson'	'Marine'
'Philip'	'Bailey'	'Marine'
'Adam'	'Baldwin'	'Animal Mother'
'Louis'	'Barlotti'	'Marine'
'John'	'Beddows'	'Marine'
'Patrick'	'Benn'	'Marine'
'Bruce'	'Boa'	'Poge Colonel'
'Steve'	'Boucher'	'Marine'

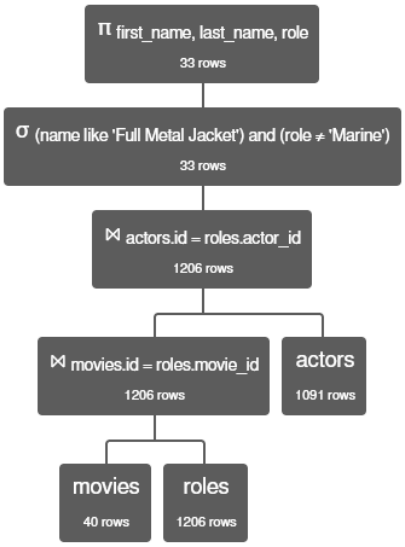
‹ 1 2 3 ›

actors.first_name	actors.last_name	roles.role
'Michael (III)'	'Williams'	'Marine'
'John (V)'	'Wilson'	'Marine'
'John'	'Wonderling'	'Marine'
'Laurie'	'Gomes'	'Marine'
'Leanne'	'Hong'	'Motorbike hooker'
'Vivian'	'Kubrick'	'News camera operator at mass '
'Ngoc'	'Le'	'V.C sniper'
'Papillon'	'Soo'	'Da Nang hooker'

‹ 9 10 11 ›

14.- Mostrar el reparto (nombre y rol) de la película “Full Metal Jacket”, mostrando únicamente los que no tuvieron el rol de ‘Marine’.

```
SELECT DISTINCT
    first_name,
    last_name,
    role
FROM movies
JOIN roles ON movies.id = roles.movie_id
JOIN actors ON actors.id = roles.actor_id
WHERE (name LIKE 'Full Metal Jacket') AND (role != 'Marine');
```



π first_name, last_name, role σ (name like 'Full Metal Jacket') and (role ≠ 'Marine') ((movies ⋈ movies.id = roles.movie_id roles) ⋈ actors.id = roles.actor_id actors)

Execution time: 0 ms

actors.first_name	actors.last_name	roles.role
'Adam'	'Baldwin'	'Animal Mother'
'Bruce'	'Boa'	'Poge Colonel'
'Costas Dino'	'Chimona'	'Chili'
'Tim'	'Colceri'	'Doorgunner'
'Marcus'	'D'Amico'	'Hand Job'
'Vincent (I)'	'D'Onofrio'	'Pvt. Pyle/Leonard Lawrence'
'Peter'	'Edmund'	'Snowball'
'R. Lee'	'Ermey'	'Gunnery Sgt. Hartman'
'Tan Hung'	'Francione'	'ARVN pimp'
'Dorian'	'Harewood'	'Eightball'

actors.first_name	actors.last_name	roles.role
'Keith'	'Hodiak'	'Daddy D.A.'
'Arliss'	'Howard'	'Pvt. Cowboy'
'Kevyn Major'	'Howard'	'Rafterman'
'Kieron'	'Jecchinis'	'Crazy Earl'
'Gil'	'Kopel'	'Stork'
'Stanley'	'Kubrick'	'Murphy'
'Sal'	'Lopez'	'T.H.E. Rock'
'Peter'	'Merrill'	'TV journalist'
'Gary Landon'	'Mills'	'Donlon'
'Matthew'	'Modine'	'Pvt. Joker'

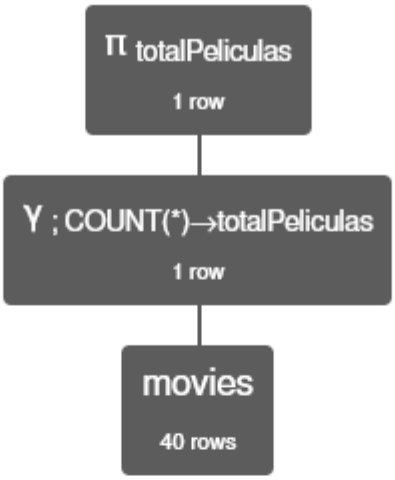
actors.first_name	actors.last_name	roles.role
'Herbert'	'Norville'	'Daytona Dave'
'Ed'	'O'Ross'	'Lt. Touchdown/Walter J. Schin'
'David'	'Palffy'	'Mass Grave soldier'
'Nguyen Hue'	'Phong'	'Camera thief'
'Jon'	'Stafford'	'Doc Jay'
'Duc Hu'	'Ta'	'Dead N.V.A.'
'Kirk'	'Taylor'	'Payback'
'John (I)'	'Terry'	'Lt. Lockhart'
'Ian'	'Tyler'	'Lt. Cleves'
'Leanne'	'Hong'	'Motorbike hooker'

actors.first_name	actors.last_name	roles.role
'Vivian'	'Kubrick'	'News camera operator at mass '
'Ngoc'	'Le'	'V.C sniper'
'Papillon'	'Soo'	'Da Nang hooker'

15.- Mostrar el número de películas registradas en la base de datos.

```
SELECT COUNT(*) AS totalPeliculas

FROM movies;
```



π totalPeliculas γ ; COUNT(*)→totalPeliculas movies

Execution time: 0 ms

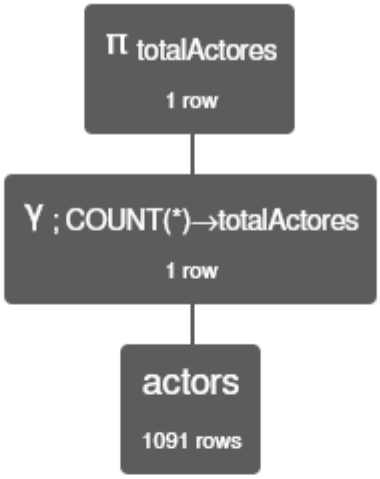
totalPeliculas
40

‹ 1 ›

16.- Mostrar el total de actores en la base de datos.

```
SELECT COUNT(*) AS totalActores

FROM actors;
```



π totalActores γ ; COUNT(*)→totalActores actors

Execution time: 0 ms

totalActores
1091

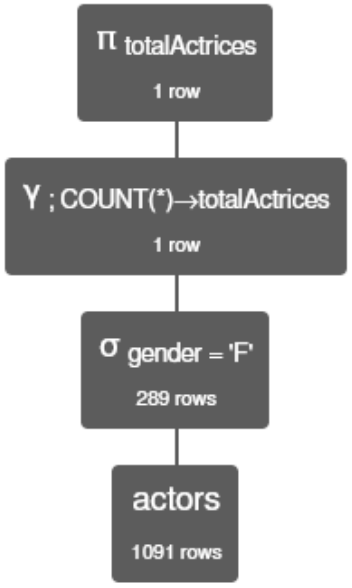
‹ 1 ›

17.- Mostrar el total de actrices en la base de datos.

```
SELECT COUNT(*) AS totalActrices

FROM actors

WHERE gender = 'F';
```



π totalActrices γ ; COUNT(*)→totalActrices σ gender = 'F' actors

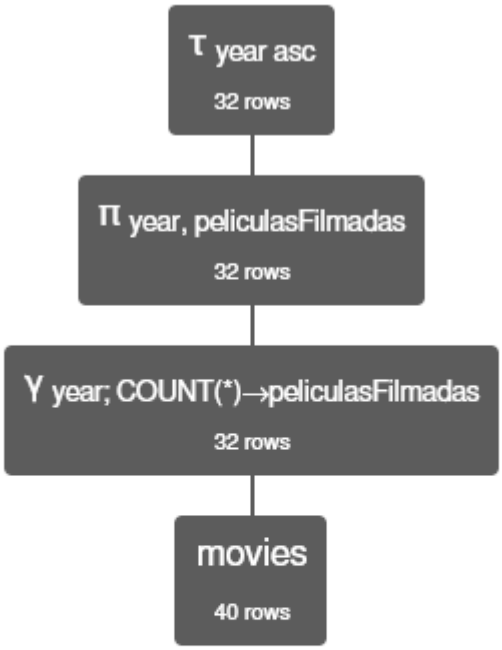
Execution time: 0 ms

totalActrices
289

‹ 1 ›

18.- Encontrar cuantas películas se filmaron en cada año.

```
SELECT
    year,
    COUNT(*) AS peliculasFilmadas
FROM movies
GROUP BY year
ORDER BY year;
```



τ year asc π year, peliculasFilmadas γ year; COUNT(*)→peliculasFilmadas movies

Execution time: 0 ms

movies.year	peliculasFilmadas
1951	2
1953	2
1955	1
1956	1
1957	1
1960	1
1962	1
1964	1
1968	1
1971	1

‹ 1 2 3 ›

movies.year	peliculasFilmadas
1975	1
1978	1
1980	1
1981	1
1984	1
1986	1
1987	2
1989	1
1991	1
1992	1

‹ 1 2 3 4 ›

movies.year	peliculasFilmadas
1994	3
1995	1
1996	1
1997	2
1999	1
2000	1
2001	1
2002	1
2003	3
2004	1

‹ 1 2 3 4 ›

movies.year	peliculasFilmadas
2005	1
2006	1

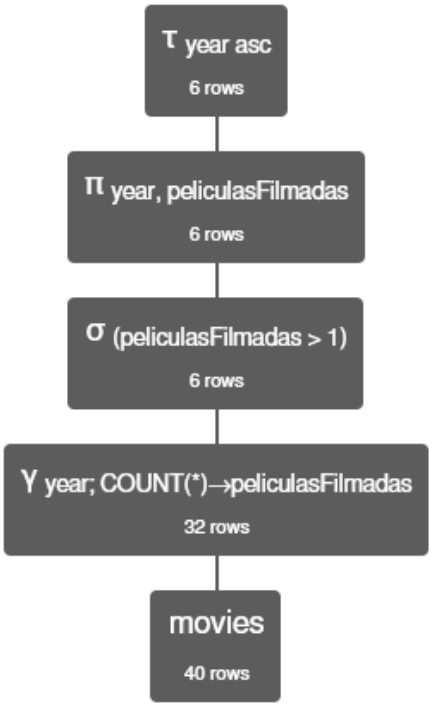
‹ 2 3 4 ›

19.- Encontrar en que años se filmaron más de 1 película.

```
SELECT
    year,
    COUNT(*) AS peliculasFilmadas
FROM movies
GROUP BY year
HAVING (peliculasFilmadas > 1)
ORDER BY year;
```

movies.year	peliculasFilmadas
1951	2
1953	2
1987	2
1994	3
1997	2
2003	3

‹ 1 ›

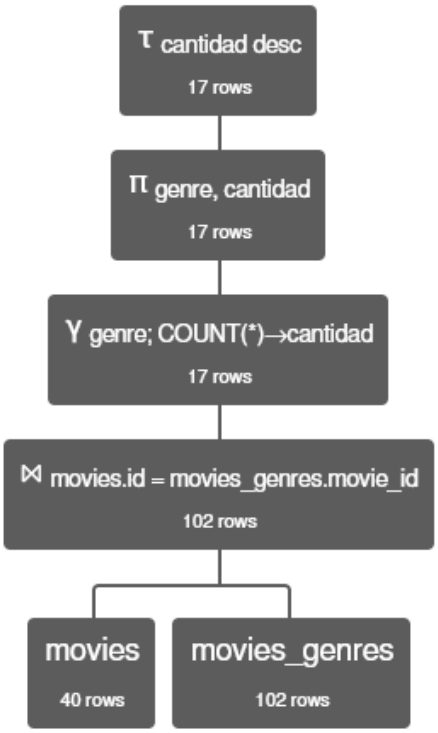


τ year asc π year, peliculasFilmadas σ (peliculasFilmadas > 1) γ year; COUNT(*)→peliculasFilmadas movies

Execution time: 0 ms

20.- Encontrar cuantas películas hay de cada género.

```
SELECT
    genre,
    COUNT(*) AS cantidad
FROM movies
JOIN movies_genres ON movies.id = movies_genres.movie_id
GROUP BY genre
ORDER BY cantidad DESC;
```



τ cantidad desc π genre, cantidad γ genre; COUNT(*)→cantidad (movies ⋈ movies.id = movies_genres.movie_id movies_genres)

Execution time: 0 ms

movies_genres.genre	cantidad
'Drama'	19
'Thriller'	12
'Action'	11
'Sci-Fi'	9
'Romance'	7
'War'	7
'Crime'	7
'Documentary'	6
'Short'	6
'Comedy'	5

‹ 1 2 ›

movies_genres.genre	cantidad
'Adventure'	3
'Horror'	3
'Mystery'	2
'Film-Noir'	2
'Family'	1
'Fantasy'	1
'Music'	1

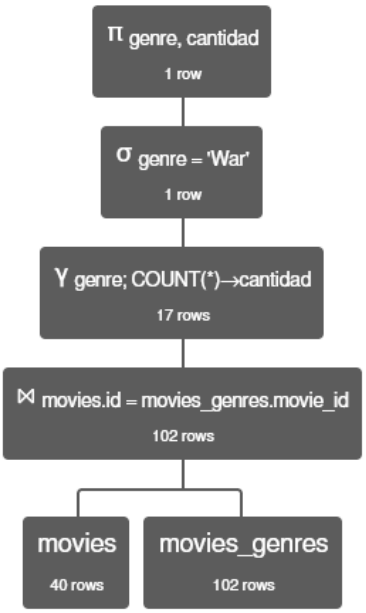
‹ 1 2 ›

21.- Encontrar cuantas películas hay del género ‘War’.

```
SELECT
        genre,
        COUNT(*) AS cantidad
FROM movies
JOIN movies_genres ON movies.id=movies_genres.movie_id
GROUP BY genre
HAVING genre = 'War';
```

movies_genres.genre	cantidad
'War'	7

‹ 1 ›

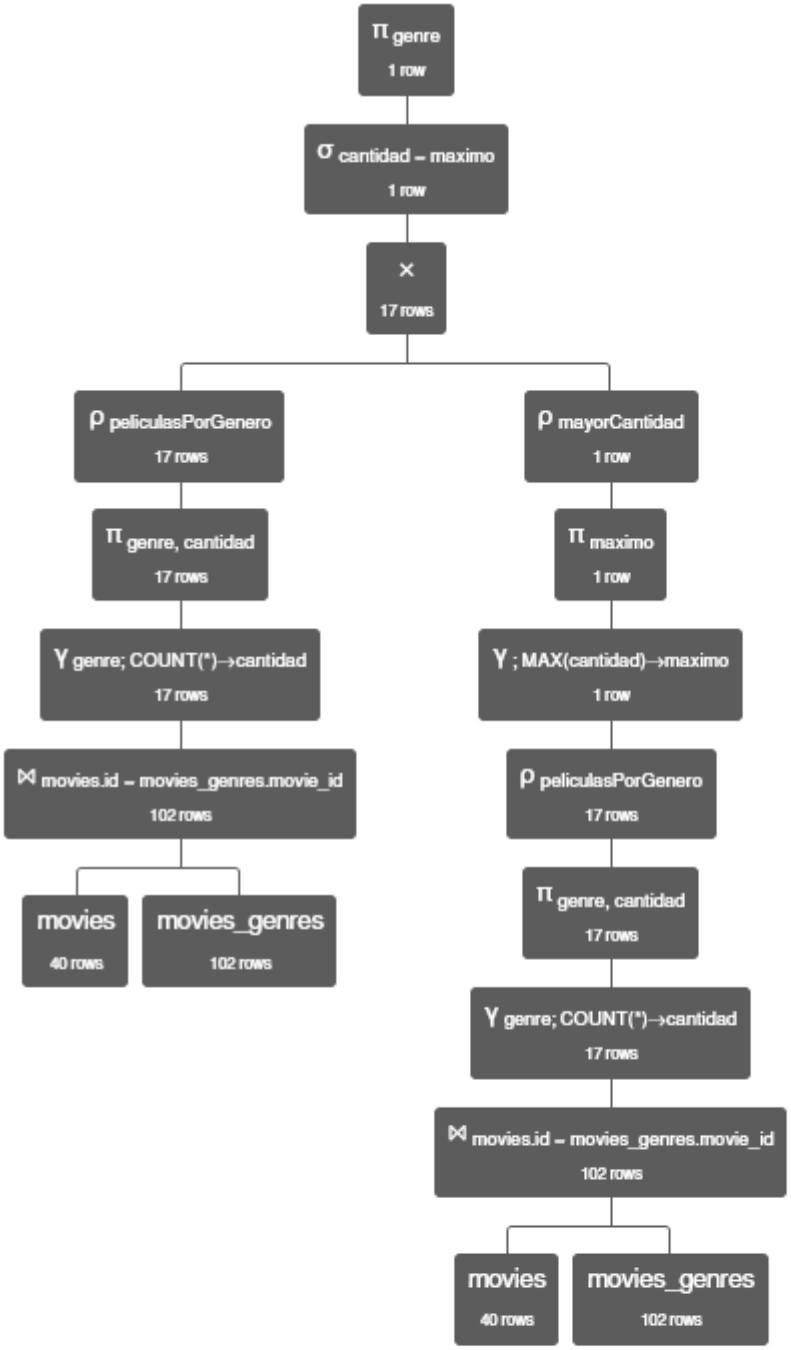


π genre, cantidad σ genre = 'War' γ genre; COUNT(*)→cantidad (movies ⋈ movies.id = movies_genres.movie_id movies_genres)

Execution time: 0 ms

22.- Encontrar de que genero hay más películas.

```
SELECT genre
FROM
	(SELECT DISTINCT
		genre,
		COUNT(*) AS cantidad
	FROM movies
	JOIN movies_genres ON movies.id = movies_genres.movie_id
	GROUP BY genre) AS peliculasPorGenero
CROSS JOIN (
	SELECT MAX(cantidad) AS maximo
	FROM (
		SELECT DISTINCT
			genre,
			COUNT(*) AS cantidad
		FROM movies
		JOIN movies_genres ON movies.id = movies_genres.movie_id
		GROUP BY genre) AS peliculasPorGenero) AS mayorCantidad
WHERE cantidad = maximo
```



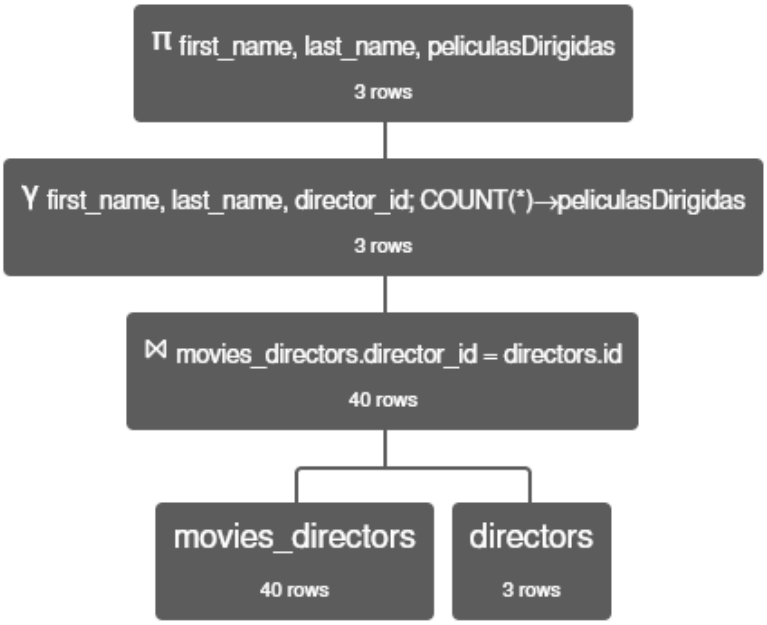
π genre σ cantidad = maximo (ρ peliculasPorGenero (π genre, cantidad γ genre; COUNT(*)→cantidad (movies ⋈ movies.id = movies_genres.movie_id movies_genres)) × ρ mayorCantidad (π maximo γ ; MAX(cantidad)→maximo ρ peliculasPorGenero (π genre, cantidad γ genre; COUNT(*)→cantidad (movies ⋈ movies.id = movies_genres.movie_id movies_genres))))

Execution time: 0 ms

peliculasPorGenero.genre
'Drama'

23.- Mostrar el número de películas que ha dirigido cada director.

```
SELECT DISTINCT
        first_name,
        last_name,
        COUNT(*) AS peliculasDirigidas
FROM movies_directors
JOIN directors ON movies_directors.director_id = directors.id
GROUP BY
        first_name,
        last_name,
        director_id;
```



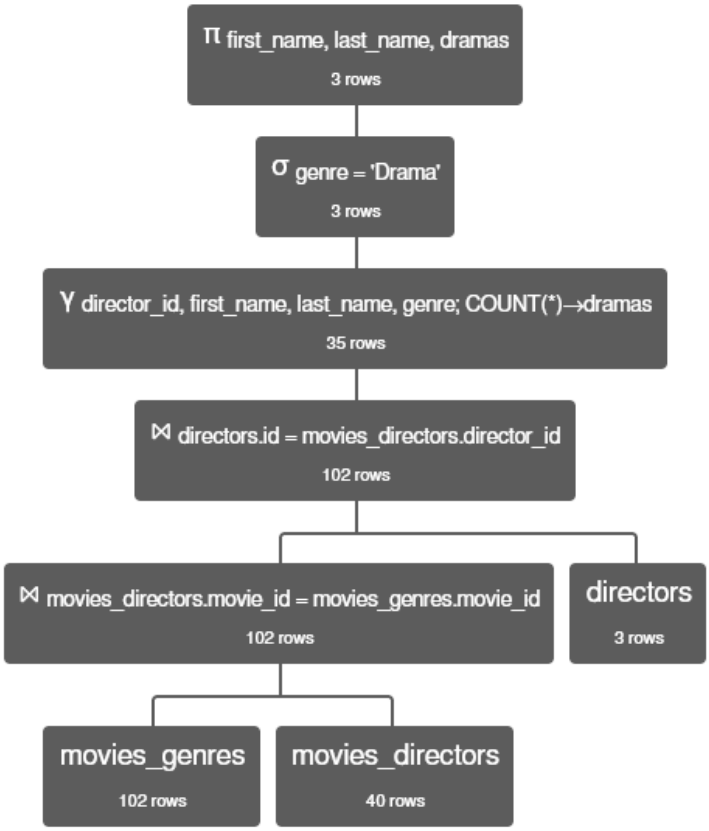
π first_name, last_name, peliculasDirigidas γ first_name, last_name, director_id; COUNT(*)→peliculasDirigidas (movies_directors ⋈ movies_directors.director_id = directors.id directors)

Execution time: 0 ms

directors.first_name	directors.last_name	peliculasDirigidas
'James (I)'	'Cameron'	14
'Stanley'	'Kubrick'	16
'Quentin'	'Tarantino'	10

24.- Mostrar el número de películas del género drama que ha dirigido cada director.

```
SELECT
    first_name,
    last_name,
    COUNT(*) AS dramas
FROM movies_genres
JOIN movies_directors ON movies_directors.movie_id = movies_genres.movie_id
JOIN directors ON directors.id = movies_directors.director_id
GROUP BY
    director_id,
    first_name,
    last_name,
    genre
HAVING genre = 'Drama'
```



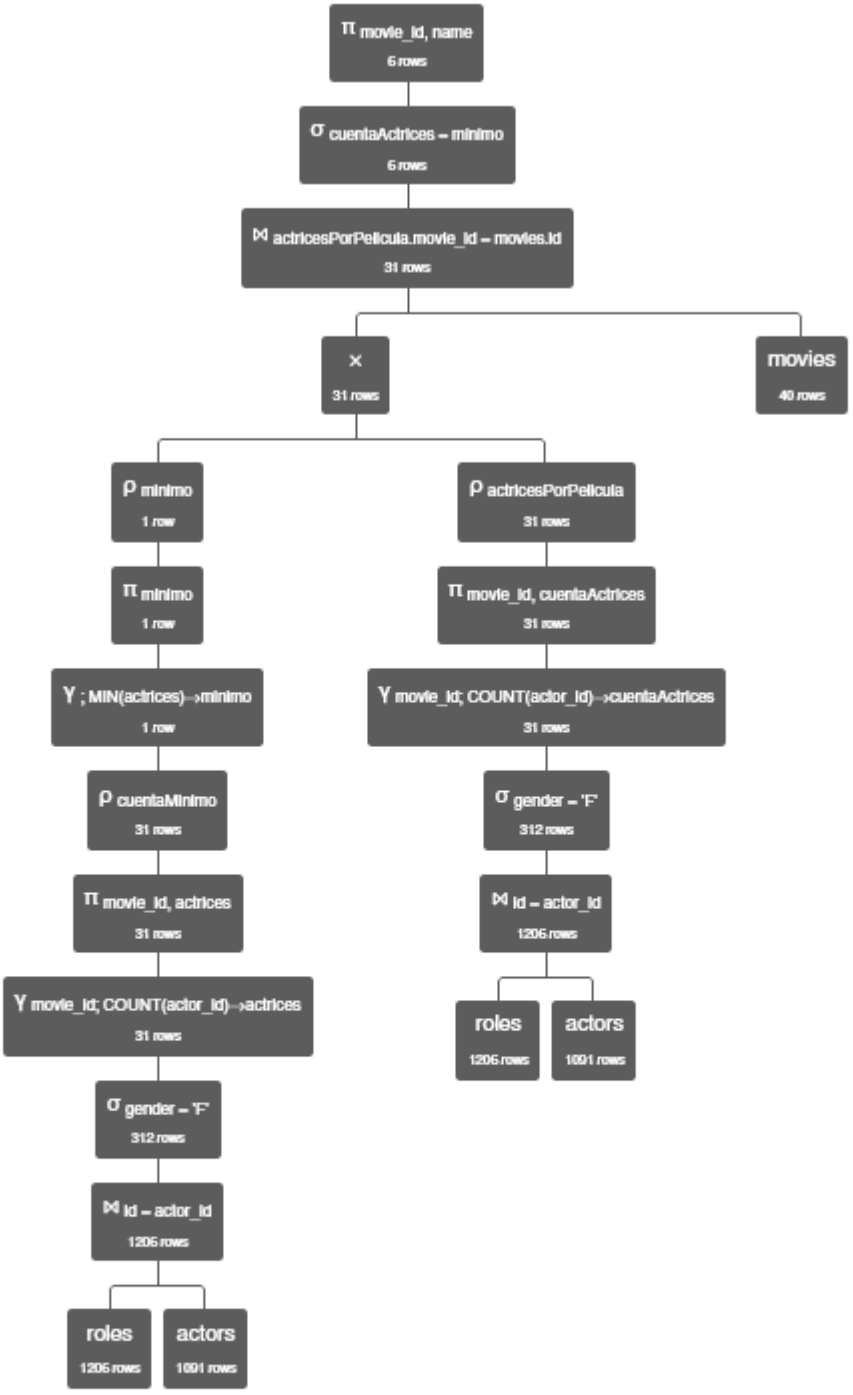
π first_name, last_name, dramas σ genre = 'Drama' γ director_id, first_name, last_name, genre; COUNT(*)→dramas ((movies_genres ⋈ movies_directors.movie_id = movies_genres.movie_id movies_directors) ⋈ directors.id = movies_directors.director_id directors)

Execution time: 0 ms

directors.first_name	directors.last_name	dramas
'James (I)'	'Cameron'	3
'Stanley'	'Kubrick'	10
'Quentin'	'Tarantino'	6

25.- Encontrar en que película(s) participaron el menor número de actrices.

```
SELECT
        movie_id,
        name
FROM
        (SELECT MIN(actrices) AS minimo
        FROM
                (SELECT
                        movie_id,
                        COUNT(actor_id) AS actrices
                FROM roles JOIN actors ON id = actor_id
                WHERE gender = 'F'
                GROUP BY movie_id) AS cuentaMinimo)
        AS minimo
CROSS JOIN
        (SELECT
                movie_id,
                COUNT(actor_id) AS cuentaActrices
        FROM roles
        JOIN actors ON id = actor_id
        WHERE gender = 'F'
        GROUP BY movie_id) AS actricesPorPelicula
JOIN movies ON actricesPorPelicula.movie_id = movies.id
WHERE cuentaActrices = minimo
```



π movie_id, name σ cuentaActrices = minimo ((ρ minimo (π minimo γ ; MIN(actrices)→minimo ρ cuentaMinimo (π movie_id, actrices γ movie_id; COUNT(actor_id)→actrices σ gender = 'F' (roles ⋈ id = actor_id actors))) × ρ actricesPorPelicula (π movie_id, cuentaActrices γ movie_id; COUNT(actor_id)→cuentaActrices σ gender = 'F' (roles ⋈ id = actor_id actors))) ⋈ actricesPorPelicula.movie_id = movies.id movies)

Execution time: 0 ms

actricesPorPelicula.movie_id	movies.name
322652	'T2 3-D: Battle Across Time'
369522	'Xenogenesis'
79846	'Day of the Fight'
92616	'Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb'
110246	'Fear and Desire'
250612	'Paths of Glory'