

Ejercicio 1

1.1. Calcular el rating máximo y promedio por género.

```
ALTER TABLE data_movies  
RENAME COLUMN imdb_rating TO rating;  
  
SELECT DISTINCT genre kind_genre, MIN(rating) AS min_rating,  
ROUND(AVG(rating),2) AS avg_rating, MAX(rating) AS max_rating  
FROM data_movies  
WHERE genre IS NOT NULL AND rating IS NOT NULL  
GROUP BY genre;
```

	kind_genre	min_rating	avg_rating	max_rating
1	action	5.8	7.41	9
2	comedy	4.4	7.14	8.6
3	horror	4.6	6.39	8.4
4	romance	4.2	6.71	8.8
5	drama	5.9	7.56	8.7

1.2. Cantidad de películas que no tienen ratings

```
SELECT COUNT(name) as n_movie_snrtg  
FROM data_movies  
WHERE rating IS NULL;
```

	n_movie_snrtg
1	4

1.3 Películas que tiene el menor ratings, sin considerar las películas que no poseen ratings.

```
SELECT name, rating  
FROM data_movies  
WHERE rating IS NOT NULL AND rating < 5  
ORDER BY rating ASC;
```

	name	rating
1	Fifty Shades of Grey	4.2
2	Alvin and the Chipmunks: The Squeakquel	4.4
3	Anaconda	4.6
4	The Twilight Saga: New Moon	4.6
5	The Haunting	4.9
6	Scooby-Doo 2: Monsters Unleashed	4.9
7	The Twilight Saga: Eclipse	4.9
8	The Twilight Saga: Breaking Dawn - Part 1	4.9

1.4 Que año fue el que obtuvo el mayor rating promedio, considerando que se tengan a lo menos 6 películas ese año.

```
SELECT DISTINCT year year_movie, COUNT(year) as num_movie, ROUND(AVG(rating),2)
as avg_rating
FROM data_movies
WHERE year IS NOT NULL AND rating IS NOT NULL
GROUP BY year
HAVING num_movie > 6
ORDER BY avg_rating DESC;
```

	year_movie	num_movie	avg_rating
1	2014	13	7.42
2	2012	12	7.17
3	2015	8	7.16
4	2013	14	7.08
5	2003	7	7.01
6	2008	10	7
7	2001	9	6.97
8	2010	11	6.93
9	2002	8	6.91
10	2005	11	6.86
11	2011	10	6.82
12	2007	11	6.78
13	2000	7	6.73
14	2006	8	6.71
15	2009	14	6.68
16	1999	11	6.61
17	2004	7	6.49
18	1997	8	6.49

1.5. Calcular el ratings promedio por año, que posea una puntuación media entre 6,5 y 7,9. Muestre los primeros 5 años que tienen un mayor rating promedio.

```
SELECT DISTINCT year year_movie, COUNT(year) as num_movie, ROUND(AVG(rating),2)
as avg_rating

FROM data_movies

WHERE year IS NOT NULL AND rating IS NOT NULL

GROUP BY year

HAVING num_movie >= 6.5 AND num_movie <= 7.9

ORDER BY avg_rating DESC;
```

	year_movie ↕	num_movie ↕	avg_rating ↕
1	2003	7	7.01
2	2000	7	6.73
3	2004	7	6.49

Ejercicio 2

2.1. Mostraremos los 5 pilotos ganadores de las últimas 5 carrera disponibles en la tabla de resultados, con el nombre de la carrera y la respectiva fecha.

```
WITH five_races as
(
    -- Ultimas 30 carreras
    SELECT races.raceId, name, date
    FROM races
    WHERE races.raceId IS NOT NULL AND date IS NOT NULL
    ORDER BY date DESC
    LIMIT 30
),
five_winers as
(
    -- 5 ganadores de las ultimas 5 carreras
    SELECT results.raceId, driverId, positionOrder
    FROM results
    WHERE positionOrder = 1
),
five_drivers as
(
    -- 5 ganadores en las ultimas 5 carreras
    SELECT name, date, driverId
    FROM five_winers
    RIGHT JOIN five_races
    ON five_winers.raceId = five_races.raceId
    WHERE driverId IS NOT NULL
    ORDER BY date DESC
    LIMIT 5
)

-- Tabla resultado Nombre Carrera, fecha y Piloto Campeón
SELECT name, date, forename, surname
FROM five_drivers
LEFT JOIN drivers
ON five_drivers.driverId = drivers.driverId;
```

	name	date	forename	surname
1	Abu Dhabi Grand Prix	2017-11-26	Valtteri	Bottas
2	Brazilian Grand Prix	2017-11-12	Sebastian	Vettel
3	Mexican Grand Prix	2017-10-29	Max	Verstappen
4	United States Grand Prix	2017-10-22	Lewis	Hamilton
5	Japanese Grand Prix	2017-10-08	Lewis	Hamilton

2.2. Mostraremos los 5 pilotos lideres en puntos entre el año 2000 y el año 2017, con su fecha de nacimiento, nacionalidad y puntaje acumulado.

```
WITH race_years as
(
    -- carreras entre 2000 y 2017
    SELECT races.raceId
    FROM races
    WHERE year >= 2000 AND year <= 2017
),
race_drivers as
(
    -- Pilotos y puntos
    SELECT results.raceId, results.driverId, points
    FROM results
    WHERE points IS NOT NULL AND points != 0
),
drivers_point as
(
    SELECT driverId, SUM(points) puntaje_total
    FROM race_years
    LEFT JOIN race_drivers
    ON race_years.raceId = race_drivers.raceId
    GROUP BY driverId
    ORDER BY puntaje_total DESC
    LIMIT 5
)
SELECT forename, surname, dob, nationality, puntaje_total
FROM drivers_point
LEFT JOIN drivers
ON drivers_point.driverId = drivers.driverId
ORDER BY puntaje_total DESC;
```

	forename	surname	dob	nationality	puntaje_total
1	Lewis	Hamilton	07/01/1985	British	2610
2	Sebastian	Vettel	03/07/1987	German	2425
3	Fernando	Alonso	29/07/1981	Spanish	1849
4	Nico	Rosberg	27/06/1985	German	1594.5
5	Kimi	Räikkönen	17/10/1979	Finnish	1565

2.3. y 2.4. Considerando los años 1950 y 2017, mostraremos los 5 pilotos mas ganadores de la historia, y la cantidad de victorias del piloto más ganador.

```
WITH a_races as
(
    -- Registros de los campeones de cada carrera en (results)
    SELECT results.raceId, driverId, positionOrder
    FROM results
    WHERE positionOrder = 1
),
a_drivers as
(
    -- Todas las carreras entre 1950 y 2017
    SELECT races.raceId, date
    FROM races
    WHERE date >= 1950 AND date <= 2017
),
a_champion as
(
    -- suma de victorias en carreras por piloto
    SELECT driverId, SUM(positionOrder) num_victories
    FROM a_races
    LEFT JOIN a_drivers
    ON a_races.raceId = a_drivers.raceId
    GROUP BY driverId
    ORDER BY num_victories DESC
    LIMIT 5
)
-- tabla final con 5 pilotos mas ganadores y el total de sus victorias
SELECT forename, surname, num_victories
FROM drivers
RIGHT JOIN a_champion
ON drivers.driverId = a_champion.driverId;
```

	forename	surname	num_victories
1	Michael	Schumacher	91
2	Lewis	Hamilton	62
3	Alain	Prost	51
4	Sebastian	Vettel	47
5	Ayrton	Senna	41

2.5. Mostraremos el tiempo transcurrido entre la primera y la última victoria del piloto mas ganador, encontrado en el punto 2.3 y 2.4.

```
SET @idmax_winner = 30;
WITH win_mschu as
(
    -- Carreras en las que Michael Schumacher ganó
    SELECT results.raceId
    FROM results
    WHERE driverId = @idmax_winner AND positionOrder = 1
),
dates_win_mschu as
(
    SELECT name, date
    FROM win_mschu
    LEFT JOIN races
    ON win_mschu.raceId = races.raceId
    ORDER BY date ASC
)
SELECT ROUND(DATEDIFF (MAX(date), MIN(date)) / 365,0) `Años entre 1ra ...`
FROM dates_win_mschu;
```

	`Años entre 1ra y ultima victoria de M.Schumacher`
1	14

2.6. Mostraremos los 5 países con mas puntos.

```
WITH nat_point as
(
    SELECT results.driverId, SUM(points) puntos_total
    FROM results
    WHERE points IS NOT NULL AND points != 0
    GROUP BY results.driverId
    ORDER BY puntos_total DESC
),
nation as
(
    SELECT drivers.driverId, nationality
    FROM drivers
)
SELECT nationality, ROUND(SUM(puntos_total),0) puntos_total_nationality
FROM nat_point
LEFT JOIN nation
ON nat_point.driverId = nation.driverId
GROUP BY nationality
ORDER BY puntos_total_nationality DESC
LIMIT 5;
```

	nationality	puntos_total_nationality
1	British	8077
2	German	7146
3	Brazilian	3423
4	Finnish	3008
5	French	2806