1. Agregación de datos.

*Recordar que:*

*INNER JOIN significa “intersección”*

*el ON significa “cuando”…bla.bla..*

1. Traer todos los géneros **(genres)** y la cantidad de películas **(movies)** del mismo **(usar name y title)**.

**SELECT name, count(movies.id)**

**FROM genres**

**INNER JOIN movies**

**ON movies.genre\_id = genres.id**

**GROUP BY genres.id;**

1. De cada género, obtener el rating **(rating)** más bajo de las películas **(movies)** **(usar name de la tabla genres)**.

**SELECT** **min**(rating), genres.name

**FROM** movies

**INNER** **JOIN** genres

**ON** movies.genre\_id =genres.id

**GROUP** **BY** genres.id;

1. Obtener el rating **(rating)** más alto de las películas **(movies)** **por género** **(usar name de la tabla genres)**.

**SELECT** **max**(rating), genres.name

**FROM** movies

**INNER** **JOIN** genres

**ON** movies.genre\_id =genres.id

**GROUP** **BY** genres.id;

1. Obtener el promedio de ratings **(rating)** de películas **(movies)** por género **(usar name de la tabla genres)**.

**SELECT** **AVG**(rating), genres.name

**FROM** movies

**INNER** **JOIN** genres

**ON** movies.genre\_id =genres.id

**GROUP** **BY** genres.id;

1. Obtener los títulos **(title)** de las series **(series)** y la cantidad de temporadas **(usar number de la tabla seasons)**.

SELECT title

FROM movies

INNER JOIN series

ON series.title=season.id;

1. Obtener por cada temporada **(seasons)**, el título de la serie **(title)** y la cantidad de capítulos **(usar number de la tabla episodes)**.
2. Obtener por cada capítulo **(episodes)**, el número de temporada **(seasons)**, el nombre de la serie **(series)**, el género **(genres)** y la cantidad de actores **(actors)** que tiene.

SELECT series

1. Obtener todos los géneros **(genres)** y la cantidad de películas **(movies)** que tengan rating **(rating)** mayor a 5.

SELECT ~~movies.title, movies.rating,~~ genres.name, COUNT (movies.id)

FROM movies

INNER JOIN genres ON genres\_id=movies.genre\_id

WHERE movies.rating >5

GROUP BY movies.genre\_id

1. Traer todos los géneros **(genres)** y el promedio de rating **(rating)** de sus películas **(movies)**. Considerar solamente las películas con rating mayor a 5.
2. Traer todas las series **(series)** y su cantidad de capítulos **(episodes)**.
3. Traer todas las series **(series)** y la cantidad de capítulos **(episodes)** que se emitieron en el 2016 **(usar release\_date)**.
4. Traer todas las series **(series)**y la cantidad de capítulos **(episodes)** que se emitieron en el año actual **(usar release\_date)**.
5. **Subqueries**
6. Traer todos los actores **(actors)** cuya película favorita contenga la letra **t (usar favorite\_movie\_id)**.

**SELECT** a.id a,~~last\_name~~

**FROM** actors **as** a

**INNER** **JOIN** movies

**ON** a.favorite\_movie\_id=movies.id

**WHERE** movies.title **LIKE** '%t%';

1. Traer todos los actores **(actors)** y las películas **(movies)** en las que actúan siempre y cuando la película favorita del actor contenga la letra **t (usar favorite\_movie\_id de la tabla actors)**.

Lo anterior lo metió en un WHERE IN(…

1. Traer todos los géneros **(genres)** que tengan series **(series)** que hayan estrenado en el 2013 o posterior **(usar release\_date)**.
2. Exists
3. Traer las películas **(movies)** que no sean películas preferidas **(usar favorite\_movie\_id de la tabla actors)**.
4. Traer los géneros **(genres)** que estén en series **(series)** **y**en películas **(movies)**.
5. Traer los géneros **(genres)** que **no** tengan series.