PARTE 2 MySQL Práctica

1. Consultas Where:
2. Obtener las películas **(movies)** hechas entre Octubre 1999 y Diciembre 2004, que muestre las películas más nuevas de primero. **(usar release\_date)**
3. Obtener las películas **(movies)** que tengan duración entre 45 minutos y 2 horas y media, ordenarlo por esta columna. **(usar length)**

**SELECT \***

**FROM movies**

**WHERE length BETWEEN '45'AND '150'**

**ORDER BY length ASC;**

1. Obtener los actores **(actors)**que el nombre empiece con ‘HE’ hasta los que empiezan con ‘TO’, ordenarlas como desees. **(usar first\_name)**

**SELECT \***

**FROM actors**

**WHERE first\_name LIKE 'HE%' AND first\_name LIKE 'TO';**

1. Obtener las películas **(movies)** que empiezan con la letra ‘T’, ordenarlas como desees. **(usar title)**

**SELECT \***

**FROM movies**

**WHERE title LIKE 'T%'**

**ORDER BY title ASC;**

1. Obtener las películas **(movies)** que empiezan con la letra ‘T’ y terminen con la letra ‘C’, ordenarlas como desees. **(usar title)**

**SELECT \***

**FROM movies**

**WHERE title LIKE 'T%C'**

**ORDER BY title ASC;**

1. Obtener las películas **(movies)** hechas en el año 2000, ordenarlos por nombre de la película. **(usar release\_date y title)**

**SELECT \***

**FROM movies**

**WHERE release\_date BETWEEN '2000-01-01'AND '2000-12-31'**

**ORDER BY title;**

1. Obtener las películas **(movies)** que en el título contenga un guión “-”, ordenarlas como desees. **(usar title)**

**SELECT \***

**FROM movies**

**WHERE title LIKE '%-%'**

**ORDER BY title ASC;**

1. Obtener las películas **(movies)** hechas en el mes de Octubre o en el año 1999, ordenar que el año sea el primer ordenamiento y el título de mayor a menor. **(usar release\_date y title)**

**SELECT \***

**FROM movies**

**WHERE MONTH (release\_date)='10' OR YEAR (release\_date)='1999'**

**ORDER BY release\_date, title DESC;**

1. Obtener los actores **(actors)** que de nombre tengan: inicie con ‘J’ con cuatro letras de comodín(usar el guión bajo para ello) y finalicen con ‘Y’, ordenarlas como desees. **(usar first\_name)**

**SELECT \***

**FROM actors**

**WHERE first\_name LIKE 'J\_\_\_Y'**

**ORDER BY first\_name ASC;**

1. Obtener los actores **(actors)** cuyos nombres terminen en ‘AM’, ordenarlos por el  apellido y por el nombre de menor a mayor **(usar first\_name y last\_name)**.

SELECT \*

FROM actors

WHERE first\_name LIKE '%AM'

ORDER BY last\_name, first\_name DESC;

1. Obtener las películas **(movies)** que contengan en su título la letra ‘Y’ como conjunción y además contenga ‘LA’, ordenarlas como desees **(usar title)**.

SELECT \*

FROM movies

WHERE title LIKE '% y %' AND title LIKE '% la %'

ORDER BY title ASC;

1. Obtener los actores **(actors)** que contengan en el apellido ‘de’ ó ‘ll’ y en el nombre ‘a’, ordenarlas como desees **(usar first\_name y last\_name)**.

**SELECT \***

**FROM actors**

**WHERE (last\_name LIKE '%de%' OR last\_name LIKE '%ll%') AND first\_name LIKE '%a%'**

**ORDER BY last\_name ASC;**

1. Obtener las películas **(movies)** que sean de la saga de ‘Toy Story’ o  y las películas de la saga de ‘Harry Potter’ con duración de 2 horas, ordenarlas por nombre ascendente y luego por duración en descendente **(usar title y length)**.

**SELECT \***

**FROM movies**

**WHERE (title LIKE 'Toy%' OR title LIKE 'Harry%') AND length LIKE '120'**

**ORDER BY title ASC, length DESC;**

1. Obtener todas las películas **(movies)** que tengan de rating ‘8.3’, ‘9.1’ y ‘9.0’, ordenarlas como desees **(usar rating)**.

**SELECT \***

**FROM movies**

**WHERE rating IN ('8.3','9.1','9.0')**

**ORDER BY rating ASC;**

1. Obtener todas las películas **(movies)** que tengan 2, 5 o 7 premios, ordenarlas de manera que muestre los que tengan mayores premios primero **(usar awards)**.

**SELECT \***

**FROM movies**

**WHERE awards IN ('2','5','7')**

**ORDER BY rating DESC;**

1. Obtener todas las películas **(movies)** que sí tengan duración, mostrar primero las que tengan menor duración **(usar length)**.

**SELECT \***

**FROM movies**

**WHERE length is not null**

ORDER BY length ASC;

1. Obtener las películas **(movies)** que no sean del mes de Julio, ordenarlas como desees **(usar release\_date)**.

**SELECT \***

**FROM movies**

**WHERE MONTH (release\_date)<>'07'**

**ORDER BY title DESC;**

1. Obtener los actores **(actors)** que en el apellido no contengan la letra K, ordenarlas como desees **(usar last\_name)**.

**SELECT \***

**FROM actors**

**WHERE !(last\_name LIKE 'k%' OR last\_name LIKE '%k%' OR last\_name LIKE '%k')**

**ORDER BY last\_name DESC**

1. Obtener las películas **(movies)** que no tengan duración de 2 y 2 horas y media, muestren en orden ascendente los títulos. **(usar length y title)**

**SELECT \***

**FROM movies**

**WHERE (length NOT LIKE '120' AND length NOT LIKE'150')**

**ORDER BY title ASC;**

1. Alias de Campos y Tablas
2. Retornar de la tabla películas **(movies)** los valores de la columna **title** como ‘Titulo de la Pelicula’, ordenarlas como desees.

select m.title as "nombre de pelicula"

from movies as m

1. Obtener los campos ‘**id**’ como ‘id\_genero’, ‘**name**’ como ‘nombre\_genero’  de la tabla generos **(genres)**, ordenarlo por nombre\_genero de menor a mayor.

**SELECT g.id AS 'id\_genero', name AS 'nombre\_genero'**

**FROM genres AS g**

**ORDER BY 'nombre\_genero' ASC;**

1. Retornar el texto ‘Actor:’ delante del campo ‘**first\_name**’ de la tabla **actors**, ordenarlo por ese campo.

*Puedes usar el operador de concatenación || (doble pipes), para que mysql las pueda usar debes colocar antes de tu sentencia:*

set sql\_mode=PIPES\_AS\_CONCAT;

**set sql\_mode=PIPES\_AS\_CONCAT;**

**SELECT 'Actor' ' ' || first\_name FROM actors;**

1. Pongámonos a pensar qué más podemos hacer en alias.
2. Combinaciones - Table Reference
3. Retornar las películas **(movies)** y sus géneros **(genres)**, ordenando por nombre de película. **(usar title)**
4. Obtener las películas **(movies)** con sus actores **(actors)**, ordenar por nombre de pelicula y nombre de los actores. **(usar title y first\_name)**

**select m.title, a.first\_name, a.last\_name**

**FROM movies AS m, actor\_movie AS am, actors AS a**

**WHERE m.id=am.movie\_id AND am.actor\_id=a.id**

1. Obtener los actores **(actors)** y las películas **(movies)** a las que pertenecieron. **(usar first\_name, last\_name y title)**

# JOIN

**SELECT m.title. CONCAT (a.first\_name, “”, a.last\_name)AS actor**

**FROM movies AS m**

**LEFT JOIN actor\_movie AS am ON m.id=am.movie\_id**

**LEFT JOIN actors AS a ON am.actor\_id=a.id**

1. Retornar las películas **(movies)** y sus géneros **(genres)**, ordenando por nombre de película utilizando joins **(usar title)**.
2. Obtener las películas **(movies)** con sus actores **(actors)**, ordenar por nombre de pelicula y nombre de los actores utilizando joins **(usar title y first\_name)**.
3. Obtener los actores **(actors)** y las películas **(movies)** a los que pertenecieron utilizando joins **(usar first\_name, last\_name y title)**.
4. Retornar los géneros **(genres)** ordenados de mayor a menor por su nombre, con referencia a las películas **(movies)** utilizando joins **(usar name y title)**.
5. Obtener las películas **(movies)** con sus géneros **(genres)** y los actores **(actors)** que participen en cada una de ellas utilizando joins **(usar title, name, first\_name y last\_name)**.

SELECT movies.title, genres.name, CONCAT (actors\_first\_name,””,actors.last\_name)

FROM movies

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

INNER JOIN actor\_movie ON actor\_movie.movie\_id=movies.id

INNER JOIN actors ON actor.id=actor\_movie.actor\_id