

Práctica 1 (ejercitación simple)

En relación a la base de datos proporcionada, realizar estos ejercicios. Utilizaremos estos temas ya vistos: SELECT | WHERE | ORDER BY / ASC/DESC | LIMIT / OFFSET | BETWEEN / LIKE | Alias |

- a. Obtener todas las películas (movies) que se realizaron luego del año 2000 (usar release_date).
- b. Obtener todas las películas (movies) hechas entre el año 2000 y el año 2010 (release_date).
- c. Obtener todos los actores (actors) que no se llamen "Mark".
- d. Obtener todos los actores (actors) que se llamen "Mark" o "Sam".
- e. Obtener todas las películas (movies) que hayan sido hechas previo al año 2000 o posterior al año 2009 y además tengan id mayor a 10 (usar release_date e id).
- f. Obtener todos los actores (actors) ordenados alfabéticamente por nombre y en segundo lugar por su apellido (usar first_name y last_name).
- g. Obtener todas las películas (movies) ordenadas por la fecha de estreno, desde la más antigua a la más reciente. (usar release_date).
- h. Obtener aquellas películas (movies) hechas en el siglo XXI ordenadas por título (usar release_date y title).
- i. Obtener únicamente 3 películas (movies), a partir de la película 7 hechas en el siglo XXI (usar release_date).
- j. Obtener las películas (movies) hechas entre Octubre 1999 y Diciembre 2004, que muestre las películas más nuevas de primero (usar release_date).
- k. Obtener los actores (actors) que el nombre empiece con "HE" hasta los que empiezan con "TO", ordenarlo como desee (usar first_name).
- l. Obtener las películas (movies) que empiezan con la letra "T", ordenarlo como desee (usar title).
- m. 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).
- n. Obtener los actores (actors) que contengan en el apellido "DE" ó 'll' y en el nombre "A". Ordenarlo como desee (usar first_name y last_name).
- o. Obtener las películas (movies) que sean de la saga de "Toy Story" 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).

- p. Obtener todas las películas (movies) que tengan de rating “8.3”, “9.1” y “9.0”. Ordenarlas como desee (usar rating).
- q. Obtener las películas (movies) que no tengan duración de 2 y 2 horas y media. Mostrar en orden ascendente los títulos (usar length y title).
- r. Obtener los campos “id” como “id_genero”, “name” como “nombre_genero” de la tabla generos (genres). Ordenarlo por nombre_genero de menor a mayor.

Podés 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;

Práctica 2 (Ejercicio Integrador 1)

En relación a la base de datos proporcionada, necesitamos realizar tres informes. Realizá las consultas adecuadas para obtener los datos necesarios. Utilizaremos SELECT ... FROM ... WHERE

- s. Obtener las películas (movies) y sus géneros (genres), ordenado por nombre de película (usar title).
- t. Obtener las películas (movies) con sus actores (actors), ordenar por nombre de película y nombre de los actores (usar title y first_name).
- u. Obtener los actores (actors) y las películas (movies) a las que pertenecieron (usar first_name, last_name y title).
- v. Obtener la cantidad de temporadas de cada serie. Ordenado por la serie que más temporadas tenga, y luego por título en orden descendente.

Así:

Supernatural	12
The Big Bang Theory	10
...	...

w.