

Práctica I (Integradora)

En relación a la base de datos proporcionada, realizar estos ejercicios.

Utilizaremos estos temas ya vistos:

Consultas avanzadas: JOINS | DISTINCT | Funciones MySQL | Funciones de Agregación | GROUP BY | HAVING

Nos contactan desde **DigitalFlix** para realizar distintos informes enfocados en su catálogo, con el objetivo de realizar un análisis de mercado y proyectar futuras adquisiciones y despliegue en su público específico. Tomaremos como origen su propia base de datos, a la que nos dan acceso.

Deberemos enviarles el código SQL necesario para que ellos puedan realizar las consultas también en el futuro.

Como equipo tomamos la decisión de realizar las pruebas en WorkBench, para chequear contra la base de datos, si los resultados obtenidos con cada consulta son los esperados.

En esta primera etapa, solicitan:

1. Mostrar los géneros de las películas (*genres*), ordenados de mayor a menor por su nombre, con referencia a las películas (*movies*) utilizando joins. Solicitan que el informe muestre: *name* y *title*.
2. Mostrar las películas (*movies*) con sus géneros (*genres*) y los actores (*actors*) que participen en cada una de ellas utilizando joins. Solicitan que el informe muestre: *title*, *name*, *first_name* y *last_name*.
3. Mostrar los actores (*actors*) y las películas (*movies*) en las que participaron. Ordenar por nombre de actor. Mostrar: *first_name* y *title*.
4. Mostrar las películas (*movies*) con sus géneros (*genres*) si los posee y los géneros con todas las películas que le corresponden, ambas en una sola consulta, sin ordenamiento. Mostrar: *title* y *name*.

Mostrar también este informe utilizando un ordenamiento por título de la película (*title*), y por nombre del género (*name*).
Luego viceversa, por nombre del género (*name*) y por título de película (*title*).

5. Mostrar 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.
6. Mostrar todos los géneros (genres) y el promedio de rating (rating) de sus películas (movies). Considerar solamente las películas con rating mayor a 5.
7. Mostrar todas las series (series) y la cantidad de capítulos (episodes) que se emitieron en el 2016 (usar release_date).
8. Mostrar todas las series (series) y la cantidad de capítulos (episodes) que se emitieron en el año actual (usar release_date).

Práctica II (Integradora)

En relación a la base de datos proporcionada, realizar este ejercicio.

Utilizaremos estos temas ya vistos:

SUBQUERIES | EXISTS | SUBQUERIES con INNER JOIN

Complacidos con nuestro trabajo anterior, nos contactan nuevamente desde **DigitalFlix** para elaborar nuevos informes.

Nuevamente deberemos enviarles el código SQL necesario para que ellos puedan realizar las consultas también en el futuro y realizaremos las pruebas en Workbench.

En esta segunda etapa, solicitan:

1. Mostrar todos los actores (actors) cuya película favorita contenga la letra "T" (usar favorite_movie_id).
 2. Mostrar 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).
 3. Mostrar todos los géneros (genres) que tengan series (series) que se hayan estrenado en el 2013 o posterior (usar release_date).
-

4. Mostrar las películas (movies) que no sean películas preferidas (usar favorite_movie_id de la tabla actors).
5. Mostrar los géneros (genres) que estén en series (series) y en películas (movies).
6. Mostrar los géneros (genres) que no tengan series.