

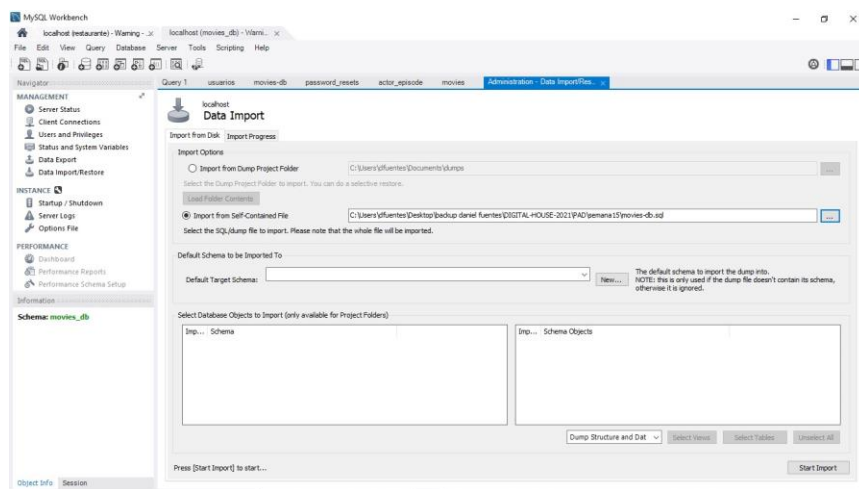
Uniones y funciones de agregación

Práctica integradora

Objetivo

Retomando la base de datos de películas vamos a practicar cómo realizar consultas consumiendo datos de diferentes tablas.

Si aún no tenemos la base de datos, podemos descargar el script haciendo [click acá](#). Recordemos que va a ser necesario importarlo y ejecutarlo en Workbench o algún programa similar.



1

A continuación, planteamos diferentes consultas que debemos realizar sobre esta base de datos.

Algunas sugerencias de cómo encarar el ejercicio:

- Empezar por generar una mirada rápida a las tablas y sus campos para tener una idea de dónde buscar la información.
- Recordar que, como en todo el resto de los ambientes, tenemos una consola y que, si MySQL da un error, va a detallar muy bien qué salió mal. 🤖👉

Sin más preámbulo, arranquemos con los micro desafíos.

¡Buena suerte! 😎👍🌟

Micro desafío - Paso 1:

Utilizando la base de datos de películas queremos conocer, por un lado, los títulos y el nombre del género de todas las series de la base de datos. Por otro, necesitamos listar los títulos de los episodios junto con el nombre y apellido de los actores que trabajan en cada uno de ellos.

```
SELECT series.title, genres.name FROM series INNER JOIN genres ON genre_id = genres.id
```

```
SELECT episodes.title, actors.first_name, actors.last_name FROM episodes INNER JOIN  
actor_episode ON episode_id = episodes.id INNER JOIN actors ON actor_id = actors.id
```



Micro desafío - Paso 2:

Para nuestro próximo desafío necesitamos obtener a todos los actores o actrices (mostrar nombre y apellido) que han trabajado en cualquier película de la saga de la Guerra de las galaxias, pero ¡cuidado!: debemos asegurarnos de que solo se muestre una sola vez cada actor/actriz.

```
SELECT DISTINCT first_name, last_name  
FROM movies  
INNER JOIN actor_movie ON movie_id = movies.id  
INNER JOIN actors ON actor_id = actors.id  
WHERE title LIKE "%Guerra%" /*Porque me tira: No data si escribo Guerra de las? Es  
porque antes de la G hay un espacio y no lo vi? >>> Preguntar*/
```



Micro desafío - Paso 3:

Debemos listar los títulos de cada película con su género correspondiente. En el caso de que no tenga género, mostraremos una leyenda que diga "no tiene género". Como ayuda podemos usar la función **COALESCE()** que retorna el primer valor no nulo de una lista.

Podemos encontrar más información haciendo [click acá](#).

```
SELECT title, COALESCE(name, 'no tiene género')
FROM movies
LEFT JOIN genres ON genre_id = genres.id
```



Micro desafío - Paso 4:

Necesitamos mostrar, de cada serie, la cantidad de días desde su estreno hasta su fin, con la particularidad de que a la columna que mostrará dicha cantidad la renombraremos:

“**Duración**”. Por ejemplo:

Título	Duración
Supernatural	130

```
SELECT title, DATEDIFF(end_date, release_date) AS Duración
FROM series
```



Micro desafío - Paso 5:

Listar los actores ordenados alfabéticamente cuyo nombre sea mayor a 6 caracteres.

```
SELECT first_name, last_name
FROM actors
WHERE LENGTH(first_name) = 6
ORDER BY first_name
```

Debemos mostrar la cantidad total de los episodios guardados en la base de datos.

```
SELECT COUNT(*)
FROM episodes
```

Para el siguiente desafío nos piden que obtengamos el título de todas las series y el total de temporadas que tiene cada una de ellas.

```
SELECT series.title, COUNT(seasons.title)
FROM series
INNER JOIN seasons ON serie_id=series.id
GROUP BY series.title
```

Mostrar, por cada género, la cantidad total de películas que posee, siempre que sea mayor o igual a 3.

```
SELECT genres.name, COUNT(movies.title)
FROM genres
INNER JOIN movies ON genre_id=genres.id
GROUP BY genres.name
HAVING COUNT(movies.title) > 3
```

Conclusión

Anteriormente, mencionamos que el manejo de bases de datos es toda una disciplina. No debemos preocuparnos si por ahora nos sentimos abrumados, lo importante es conocer las bases del tema y seguir practicando para no olvidar y adquirir nuevos conceptos sobre el tema.

¡Hasta la próxima!