

Nº de Alumno: 1530314

Apellido y Nombre: Suárez Abril

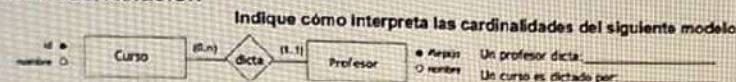
BBDD1 Parcial

2da fecha 4/12

Uso interno			
ER	AR	Norm	MySQL
✓	✓	A	A

CORREGIDO POR: Julián

Entidad/Relación



Un servicio de TV permite a sus usuarios organizar sus suscripciones a plataformas de contenidos digitales (ej. Netflix, Flow). De cada usuario se conoce su nombre y teléfono. Los contenidos ofrecidos por las plataformas consisten en películas y capítulos de series, pero pueden surgir otros tipos en el futuro. Cada contenido es exclusivo de una plataforma y tiene un nombre, duración y puntaje. Las películas además tienen un género, y de los capítulos de serie se conoce la serie a la que pertenecen. Cada serie a su vez tiene un nombre y un puntaje propio (independiente de los de sus capítulos). De cada plataforma se conoce el nombre y la url. Para cada suscripción que tiene un usuario con cada plataforma, se guardan el nombre de usuario y contraseña utilizados, y la fecha de vencimiento de la misma. También se conoce, para cada suscripción, cuáles son los contenidos vistos y los marcados como favoritos por el usuario en la plataforma (puede marcar como favorito un contenido de una plataforma aunque no lo haya visto aún). Además, de cada contenido visto por un usuario, se conoce el número de veces que se reprodujo y puede haber o no una reseña que incluya un comentario y puntaje otorgado. Un usuario no puede tener más de una suscripción con una misma plataforma.

Realizar el modelo E/R y la transformación al modelo relacional

Álgebra Relacional

Dados los siguientes esquemas:

CAPITULO(#capitulo, #serie, nombre, puntaje_promedio)
SERIE(#serie, nombre, puntaje_promedio)
RESEÑA(#reseña, #usuario, #capitulo, puntaje, comentario)
USUARIO(#usuario, email, nombre)

Hallar el #serie de las series que tienen todos sus capítulos con al menos una reseña.

Algoritmo para analizar la pérdida de d.f.

Res = X
Mientras Res cambia
Para i=1 to cant_de_particiones_realizadas
Res = Res U ((Res ∩ Ri) ∩ Ri)

Algoritmo para encontrar X'

Result = X
While (hay cambios en result) do
For (cada dependencia funcional Y → Z en F) do
if (Y ⊆ result) then result := result U Z

Normalización

Dado el siguiente esquema:

SUSCRIPCION(#suscripcion, email, nombre_usuario, #plan, nombre_plan, texto_condiciones, precio, email_adicional, nombre_adicional, #contenido, titulo, sinopsis, duracion, fecha_adicional)

Donde:

- Cada suscripción es realizada por un único usuario (identificado por el email) y un plan, pero además hay usuarios adicionales que la utilizan (email_adicional). De cada usuario adicional que se suma a la suscripción, se guarda la fecha.
- Un plan de suscripción tiene un nombre (que no puede garantizarse que sea único en el sistema), condiciones, y un precio mensual.
- Cada contenido tiene un título, sinopsis y duración. El #contenido es único en el sistema, pero del título no puede garantizarse que lo sea.
- De cada suscripción se sabe qué contenidos fueron reproducidos, sin distinción sobre qué usuario (titular o adicionales) reprodujo cada uno.

Aplicar y explicar el proceso de normalización. Considerar que el esquema ya se encuentra en 1FN.

MySQL

- Considerando las propiedades ACID de las transacciones y priorizando que esto se cumpla, ¿es necesario usar transacciones cuando se ejecuta una sentencia UPDATE? Justifique su respuesta.
- Explique 2 motivos por los cuales necesitaría definir una vista.
- En una base de datos con las siguientes tablas

CONTENIDO(#id_contenido, titulo, sinopsis)

VISUALIZACION(#id, timestamp, #id_contenido)

Hace falta crear un usuario para el frontend de la aplicación que sea capaz de obtener información sobre los contenidos, y registrar visualizaciones (pero no consultarlas). ¿Qué permisos le asignaría a cada tabla, para maximizar la seguridad de la BD?