

# Bases de Datos 2018

## Laboratorio #1

Matias Bordone Carranza, Sergio Canchi, Ramiro Damasi,  
Paula Estrella, Ezequiel Orbe y Jorge Rafael

ShowHub es una startup de servicios de streaming basado en suscripción orientada a la oferta de películas y programas de televisión, tanto de producciones propias como de terceros.

Luego de haber probado con éxito un MVP del producto y de haber conseguido financiamiento, es momento de iterar para obtener una versión 'production-ready' del mismo. Para ello, el equipo de ingeniería necesita desarrollar un modelo de datos que de soporte al modelo de negocios de la empresa.

Debido a que los tiempos apremian y que el equipo de ingeniería todavía es muy pequeño, ShowHub ha decidido contratarlos para sumarse a el equipo de ingeniería y hacerse cargo del desarrollo del modelo de datos.

## Requerimientos

Luego de las primeras reuniones con los diferentes stakeholders se ha logrado condensar una primer versión de los requerimientos del negocio que dicho modelo de datos debe soportar. En particular, se han identificado lo siguiente:

- a) Existen dos tipos de usuarios en la plataforma de streaming: los clientes y los empleados. Cada cliente tiene un nombre, apellido, correo electrónico, contraseña, fecha de nacimiento, sexo, nombre de usuario, un número de teléfono y puede estar suscrito a un plan (básico, estándar o premium) determinado.  
Por otro lado, cada empleado tiene un nombre, apellido, correo electrónico, contraseña, uno o más teléfonos, y uno o más roles de administración (por ejemplo: "admin de usuarios clientes", "admin de usuarios empleados", "admin de carga de productos", etc.)
- b) Existen dos tipos de contenido dentro de la plataforma: películas y programas de televisión. Una película tiene un título, una descripción, una duración (en minutos), una fecha de lanzamiento y pertenece a al menos un género (por ejemplo, drama, suspenso). Una película tiene actores protagonistas y actores secundarios, es dirigida por uno o más directores, y es producida por una productora (por ejemplo, Warner Bros, Pol-ka). Además, cada película tiene subtítulos en uno o más idiomas (por ejemplo; inglés, italiano).  
Por su parte, un programa de televisión tiene un título, una descripción, un número de temporada (por ejemplo, 1, 2), el año de cada temporada (por ejemplo, 2017, 2018) y al menos un género. Cada temporada del programa de televisión está

compuesta por un conjunto de episodios (capítulos) donde cada episodio tiene un título, una descripción, una duración y una fecha de lanzamiento. Al igual que una película, un programa de televisión tiene actores protagonistas y actores secundarios, es dirigido por uno o más directores, es producido por una productora y pertenece a una cadena de televisión (por ejemplo, FOX, NBC, CN).

- c) Un actor está definido por un nombre, apellido, fecha de nacimiento, sexo y su sitio web oficial. Lo mismo aplica para un director. Un director puede actuar inclusive en la misma película que dirige.
- d) Finalmente, una de las características principales de esta plataforma de streaming es su motor de recomendaciones, el cual está alimentado a partir de los reviews que cada cliente haga sobre las películas o programas de televisión que ha visto. Cada review está compuesto por un título, descripción, fecha y una calificación (de 1 a 5 estrellas).

## Tareas

### Parte I

1. Crear un modelo ER que tenga en cuenta los requerimientos presentados y que permita responder las consultas que más adelante se presentan. Usar la notación de modelado visto en el teórico para crear el modelo ER.
2. Mapear el modelo ER desarrollado en una base de datos relacional MySQL (con sus respectivas restricciones).

### Parte II

1. Responder las siguientes consultas usando SQL:
  - a. Listar los datos de los clientes suscritos al plan PREMIUM con una determinada fecha de suscripción.
  - b. Listar los datos de las películas donde el actor 'X' fue protagonista.
  - c. Listar los episodios correspondientes a un programa de televisión X y un número de temporada N. Listar ordenados por fecha de lanzamiento.
  - d. Listar los reviews hechos por un cliente X dentro de un rango de fechas.
  - e. Listar los actores que hayan actuado en películas que pertenecen a géneros de drama o comedia, ordenados por orden alfabético.
  - f. Dada una película X, calcular para cada estrella el porcentaje de calificaciones recibidas en sus reviews.
  - g. Dada una película X, calcular su "calificación promedio". Para calcular la calificación promedio en términos de 5 estrellas usar la media ponderada como se describe en <http://goo.gl/YXYVak>

- h. Listar las películas mejor calificadas en los últimos 6 meses. Usar la “calificación promedio”.
- i. Listar las películas dirigidas por dos o más directores femeninos.
- j. Listar los actores que hayan actuado en todos los programas de televisión en donde actuó un actor X.

**Para cada consulta deberás generar al menos 1 caso de prueba**

2. Crear las siguientes rutinas almacenadas:

- a. El trigger “actualizar fecha de suscripción” actualiza la “fecha de suscripción” de un “cliente” a la fecha actual cada vez que se detecta que el plan de suscripción cambió (por ejemplo, de plan básico a premium). Se debe actualizar la “fecha de suscripción” solo y solo si hubo un cambio de plan.
- b. El trigger “actualizar calificación promedio de película” actualiza la “calificación promedio” de una “película” cada vez que se insertan nuevas críticas.
- c. El procedimiento almacenado “calcular métricas de calificación” recibe como parámetros de entrada el identificador de una película. El procedimiento debe obtener los porcentajes de calificaciones para cada estrella, calificación promedio y cantidad de críticas recibidas.
- d. El procedimiento almacenado “recientemente vistas” recibe como parámetros de entrada el identificador del usuario y un tipo de producto (película o programa de televisión). El procedimiento debe obtener un listado de los datos de los productos (datos de películas o datos de programas de televisión según corresponda) cuyos géneros pertenecen al de los productos con géneros que vio el usuario y que fueron vistos por otros usuarios en la última semana.

**Para cada rutina almacenada deberás generar al menos 1 caso de prueba.**

# Entrega

Fecha de Entrega: 28/09 00:00hs

## Repositorio

En el repositorio provisto por la materia deberán crear la siguiente estructura de carpetas y archivos:

```
lab-1/  
  db/  
    db.sql  
    der.jpg  
  consultas/  
    consulta-a/  
      consulta.sql  
      test-1.sql  
      ...  
      test-n.sql  
    ...  
    consulta-n/  
  rutinas/  
    rutina-a/  
      rutina.sql  
      test-1.sql  
      ....  
      test-n.sql
```

- En la carpeta **db**, deben crear un archivo **db.sql** con el DDL de las tablas y un archivo **der.jpg** (der.png) con el diagrama ER de la misma.
- En la carpeta **consultas (rutinas)** deberán crear una carpeta **consulta-x (rutina-x)** donde **x** es el nro. de la consulta en las consignas (eg, consulta-a, consulta-b, etc).
- En cada carpeta **consulta-x (rutina-x)** debe haber un archivo llamado **consulta.sql (rutina.sql)** con la consulta (rutina) y uno o más archivos llamados **test-i.sql**

# Tests

Los tests deberán tener el siguiente formato:

```
-- -----  
-- TEST CASE: Nombre del Test Case  
-- DESCRIPTION: Descripción del Test Case  
-- -----  
  
-- INPUT DATA SET  
INSERT INTO mytable1 VALUES (...);  
INSERT INTO mytable2 VALUES (...);  
  
-- EXPECTED DATASET  
  
-- CSV, TSV o algun tipo de descripción del resultado esperado
```

Deberán crear al menos un caso de prueba por query. Se sugiere probar un caso normal, y los border cases como mínimo.