

# Bases de Datos 2020

## Examen Final - 21/12/2020

Sergio Canchi, Cristian Cardellino,  
Ramiro Demasi, Diego Piloni

### Introducción

El examen cuenta de 2 partes: SQL y NoSQL. Cada parte cuenta con un total de 5 ejercicios, de los cuales 4 son básicos para todos, y 1 es para libres únicamente. Esto hace que el total de ejercicios sea 10, 8 para regulares/libres y 2 más sólo para libres. Al final de la consigna se detallan las condiciones a tener en cuenta para la correcta entrega del examen.

### Primera Parte - SQL

Se dispone de la base de datos `sakila` cuyo diagrama se puede encontrar adjunto en el archivo `diagram.png`. Los archivos `sakila-schema.sql` y `sakila-data.sql` contienen el esquema y los datos de la BD respectivamente. En caso de tener la BD en el sistema, se recomienda borrarla y volver a cargarla para evitar inconvenientes. Para cargar los datos se utiliza los siguientes comandos:

```
$ mysql -h localhost -u root < sakila-schema.sql
$ mysql -h localhost -u root < sakila-data.sql
```

### Consignas

#### Ejercicio 1

Modifique la tabla `customer` de manera que la columna `email` pase a tener la restricción de unicidad (i.e. deberá ser única). Tenga en cuenta los valores originales con los que se creó dicha columna (viendo el archivo con el esquema) y haga los cambios que considere pertinentes.

## Ejercicio 2

- A. Cree una nueva tabla `premium_customer` que tendrá los siguientes campos:
- `customer_id`: Un campo que funcionará como clave primaria y será a su vez el campo que conecta la tabla `premium_customer` a la tabla `customer`. A la hora de definir el tipo revise la tabla relacionada.
  - `total_credits`: Un campo decimal positivo de 8 dígitos con 2 dígitos de precisión, deberá ser no nulo, 0 por defecto.
- B. Agregue en la tabla `premium_customer` todos aquellos clientes que tengan un total de alquileres hechos (según la tabla `rental`) mayor o igual a 40. En el valor `total_credits` de `premium_customer` será igual a la suma de pagos hechos por el cliente (de acuerdo a la tabla `payment`) multiplicado por 100.

## Ejercicio 3

Cree un trigger `new_premium_customer` que se ejecutará luego de que haya una nueva entrada en `rental`. Si el cliente no es parte de la tabla `premium_customer` y con el nuevo alquiler totaliza un número mayor o igual a 40 en alquileres totales, se agrega dicho cliente a la tabla `premium_customer` con `total_credits` igual al total de pagos registrados por dicho cliente en la tabla `payment` multiplicado por 100.

## Ejercicio 4

Listar el título, descripción, año de lanzamiento y cantidad total de alquileres de aquellas películas que cumplan con alguna de las siguientes condiciones:

- Tengan mayor o igual número de alquileres que el promedio de alquileres del top 100 de películas alquiladas de acuerdo a la tabla `rental`.
- Tengan menor o igual número de alquileres que el promedio de alquileres del bottom 100 de películas alquiladas de acuerdo a la tabla `rental`.

La consulta deberá devolver los valores ordenados de mayor a menor de acuerdo al número de alquileres.

## Ejercicio 5 (Sólo Para Libres)

Crear una función `top_spending_city` que tomará como parámetro de entrada un valor `minimum_spent` del mismo tipo que `payment.amount`. Deberá devolver el nombre de la ciudad (revisar que coincida el tipo de valor devuelto de la función) que tenga la mayor suma de pagos hechos en la base de datos, sólo considerando los pagos que tengan un gasto mayor o igual al parámetro `minimum_spent`.

## Segunda Parte - NoSQL

La base de datos `mflix` contiene datos sobre películas y salas de cine. Adicionalmente incluye datos de usuarios y comentarios sobre las películas. A continuación se describen todas las colecciones de la base de datos `mflix`:

Colección	Descripción
movies	Contiene información de las películas, incluyendo título, año de estreno, y director.
theaters	Contiene ubicaciones de salas de cine.
users	Contiene información del usuario.
comments	Contiene comentarios asociados a películas específicas
sessions	Campo de metadatos. Contiene JSON Web Token de los usuarios.

Para cargar la base de datos se utiliza el siguiente comando:

```
$ mongorestore --host localhost --drop --gzip --db mflix ./mflix
```

Si ya posee la BD se recomienda eliminarla y volverla a cargar.

## Consignas

### Ejercicio 6

Agregar la película "Midnight Cowboy", con los siguientes atributos:

- ``title``: "Midnight Cowboy"
- ``year``: 1969
- ``type``: "movie"
- ``cast``: "Dustin Hoffman", "Jon Voight", "Sylvia Miles", "John McGiver"
- ``runtime``: 113
- ``released``: 16/06/1969
- ``rated``: "R"

Si la película existiese en la BD (bajo el mismo título, año y tipo) se deberá hacer un update en lugar de insertarla (Hint: averiguar sobre ``upsert``).

## Ejercicio 7

Listar, sin utilizar el pipeline de agregación el título, año y cantidad de "tomates podridos y tomates frescos" de aquellas películas que tengan "tomates frescos y tomates podridos" de acuerdo a la calificación de "Rotten Tomatoes" (Hint: atributo `tomatoes`` de `movie``), y que además tengan mayor valor en "tomates frescos" que en "tomates podridos" (Hint: necesitarán de las funciones `$exists`` y `$expr``).

## Ejercicio 8

Devolver el promedio de ratings de acuerdo a IMDB por cada género, de aquellas películas que posean un rating de imdb, tengan a Alemania como parte de sus países de origen, tengan su "Rotten Tomato Meter" tanto para `viewer`` como para `critic`` y dicho `meter`` sea mayor en `critic`` que en `viewer``.

El esquema final de lo devuelto por la consulta debe ser el siguiente:

```
[
  {
    "Genre": <GENRE_NAME>,
    "IMDB Average Rating": <IMDB_RATING_VALUE>
  }
]
```

## Ejercicio 9

Se desea llevar registro de las funciones de ciertas películas en diversos cines (theaters). Para ello, deberá tener en cuenta algunas condiciones buscadas:

- Contar la cantidad de cines en los que se está proyectando una película debe ser una operación eficiente.
- Los cines pueden proyectar funciones de una sola película y saber que película está proyectando debe ser una operación eficiente.
- Las películas estarán limitadas en la cantidad de cines en los que se pueden proyectar de manera simultánea.

En base a esto, modele una relación en la base de datos de MFlix de manera que se cumplan con las condiciones establecidas. Deberá justificar claramente su respuesta. Además, deberá hacer los cambios correspondientes en la base de datos para agregar funciones de al menos 3 películas en 5 cines distintos (quedarán a su criterio las películas y/o cines seleccionados mientras existan en la BD).

## Ejercicio 10 (Sólo para Libres)

Especifique las reglas de validación para el esquema de la colección `comments``. Justifique las decisiones tomadas.

## Puntos a tener en cuenta

- Los ejercicios marcados como "Sólo Para Libres" no suman pero si restan. Quienes rindan libres y no hagan dichos ejercicios podrán acceder a un porcentaje máximo del 80% del final.
- Algunos nombres de columnas en SQL o tablas pueden ser palabras reservadas de SQL. Para evitar problemas, utilizar ``backticks``:
  - E.g. `SELECT u.username FROM `user` u;`
- Mostrar únicamente los campos pedidos en la consigna y en el orden en el que se los pide (tanto a nivel fila como a nivel columna).
- Buscar hacer la consulta de la forma más sencilla posible.
  - No hacer uniones de tablas salvo que la información de dicha tabla sea estrictamente necesaria.
- Se evaluará el correcto formato de las soluciones:
  - El código entregado debe ser legible.
  - Utilizar indentación uniforme de 2 o 4 espacios.
  - No escribir toda la consulta en una sola línea. Usen buen criterio para separar partes de la consulta.
  - Utilizar mayúsculas para denotar palabras clave de SQL (e.g. ``SELECT``, ``INSERT``, ``FROM``, etc.).
  - Utilizar CTEs (i.e. ``WITH``).

## Entrega

- Se entregará un archivo comprimido ``soluciones.zip`` (con ``soluciones.sql`` y ``soluciones.js`` adentro) con las soluciones de los ejercicios. Separar las soluciones mediante comentarios de SQL/Javascript.
- La entrega se hará mediante el [Aula Virtual](#) en el [correspondiente apartado](#).
  - Tendrán hasta las 18:30 para que se considere una entrega completa. La recomendación es empezar a subir el archivo a las 18 para evitar cualquier eventualidad.
  - Si se entrega después de esa hora, el límite serán las 19:00 y se descontará 1 punto por entrega tardía.
  - Después de las 19:00 se cerrará la entrega y el final se considerará desaprobado.