Bases de Datos 2019

Parcial #2

Sergio Canchi, Cristian Cardellino,
Ramiro Demasi, Ezequiel Orbe

Trabajaremos sobre una base de datos de juegos y usuarios de la plataforma Steam. La BD consta de 2 colecciones: games y users.

Instrucciones para la carga de datos

Para los que utilicen las máquinas de la facultad, ya se han creado para sus respectivos usuarios de grupo (con correspondientes password) bases de datos que tienen la forma: `mongo_g`**`N`**`_steam`**`M`**, donde **N** es su número de grupo y **M** es un número entre 1 y 3 (de manera que cada integrante pueda utilizar una BD). En estas máquinas, los datos ya vienen precargados (i.e. las colecciones están creadas).
Para aquellos que opten por hacerlo en sus máquinas, deberán importar los datos desde los archivos **games.json** y **users.json** (que se encuentran en el archivo parcial.tar.bz2). Para ello deberán importar los datos ejecutando los siguientes comandos:

```
$ mongoimport --db steam --collection games --drop --jsonArray --file games.json
$ mongoimport --db steam --collection users --drop --jsonArray --file users.json
```

Estructura de las colecciones

Las colecciones tienen la siguiente estructura:

games

```
{
	_id : ObjectId,
	id : int,
	title : string,
	publisher : string,
	developer : string,
	genres : [
		string
	],
	release_date : date,
	price : float,
	sentiment : string
}
```

Donde `_id` es el identificador asignado por Mongo, mientras que `id` es el identificador del juego, asignado por Steam.

users

```
{
	_id : ObjectId,
	user_id : string,
	items : [
		{
			item_id: int,
			playtime_forever: int,
			playtime_2weeks: int
		}
	]
}
```

Donde `_id` es el identificador asignado por Mongo, `user_id` es un identificador del usuario en Steam e `item_id` es un identificador de un juego de acuerdo a la plataforma de Steam (es el valor del campo `id` de la colección anterior).

Consignas

Las consultas 1 a 4 se deben resolver utilizando las operaciones de consulta y crud básicas de MongoDB (i.e. no se puede utilizar el pipeline de agregación).

1. Mostrar el título, el precio y sentimiento de todos aquellos juegos que tengan algún tipo de sentimiento "positivo" (hint: los sentimientos positivos contienen la cadena "Positive") y un precio de hasta 5 dólares.
2. Contar la cantidad de usuarios tales que en un mismo juego llegaron a jugar más de 5000 minutos en las últimas dos semanas y más de 30000 minutos en total.
3. Agregar un campo `discount_price = 2.49` a todo juego desarrollado por Valve que cuesta 4.99 dólares.
4. Agregar el juego con el título "Portal 3", de la empresa Valve (desarrolla y publica), con los géneros de "Acción" y "Aventura", a salir el 11/11/2031, a un precio de 59.99, y con un sentimiento "Mixed". Tener en cuenta que si el juego con el título mencionado existe (publicado y desarrollado por el mismo equipo mencionado), deberá hacerse un update (Hint: usar la opción upsert).

Las consultas 5 a 8 se deben resolver utilizando el pipeline de agregación.

5. Listar los últimos 5 géneros de juegos del Top 10 de géneros con mayor cantidad de juegos. Es decir, listar los géneros del 6 al 10 del Top 10. Sólo mostrar los campos "Género" y "Cantidad".
6. Listar precio máximo, precio mínimo y precio promedio de los juegos desarrollados o publicados por Valve. No se debe listar campos adicionales a los requeridos.
7. Listar el año, el mes y la lista de juegos (mostrando solo el título y desarrollador) por mes y año de estreno (ordenados de manera descendente), de aquellos meses-años que tengan más de 200 juegos estrenados. (Hint: utilizar $project y $toDate para convertir `release_date` de string a date).
8. Listar el título, el precio, el desarrollador, y cantidad de minutos jugados de los 10 juegos que más se hayan jugado en las últimas 2 semanas considerando la totalidad de los minutos jugados por todos los usuarios.

Consigna de modelado de datos.

9. Luego del éxito alcanzado en sus juegos, Steam necesita agregar un sistema de recomendaciones para esto se pide agregar a su actual modelo de datos, los reviews de juegos posteados por sus usuarios, donde se necesita modelar la siguiente información: título, texto, y fecha del review, usuario que realizó el review, juego sobre el cual se realizó el review. Agregue algunos reviews como ejemplos (con 3 es suficiente) y Justifique algunas decisiones de diseños (por ejemplo, por que agrega o no una nueva colección, tipos de lo campos elegidos).

Cosas a tener en cuenta

- No hacer consultas extremadamente complejas. Siempre buscar la variación más sencilla.
- El archivo a entregar debe estar correctamente formateado:
    - Debe ser legible, y tener una indentación de 2 espacios.
    - Nunca escribir toda la consulta en una sola línea, pero tener criterios y no separar una línea más de lo necesario.
- Revisar que las proyecciones sean las estipuladas en el enunciado.
    - No mostrar campos no pedidos.
    - No hacer proyecciones innecesarias antes de tiempo.
    - Sea eficiente y tenga cuidado a la hora de hacer uniones de colecciones (i.e. no haga uniones sobre documentos que no se utilizarán).

Entrega

- Entregar el archivo **solutions.js** con todas las consultas de Mongo que resuelvan los 9 ejercicios. Para el punto 9, la justificación deberá estar dada por un comentario de JavaScript en el mismo archivo (entre **/*** y ***/**).