

# AMPLIACION de BASES DE DATOS

(Profesor: Héctor Gómez Gauchía)

## Práctica 5 Apartados 3, 4 y 5: Utilizar una BD en MongoDB

### APARTADO 3.- Trabajando con la colección *Aficiones* en MongoDB

- a) Consultas: (*componente* es cada documento de la BD)
1. Obtener el nombre, el tema y la puntuación de los componentes mejor valorados (puntuaciones  $\geq 9$ ) pero no saques el identificador.
  2. Obtener para cada tema (agrupado): su nombre, el total de cuanto te gastarías si vas a todos los componentes mejor valorados y cuántos componentes has incluido.
  3. para cada una de estas puntuaciones por separado: 10, 9, 8, 7, 6 y 5. Para cada puntuación, obtener los nombres de componentes que la tienen. (cuando ya funcione, incluye también el nombre su tema)
  4. Lista de Apodos para cada afición (Tema)
- b) (para nota) *A quien le gusta los mismos componentes?* -> Para cada Tema, queremos saber qué Apodos tienen un mismo componente, es decir el mismo *Nombre* (ej.: el mismo nombre de libro, CD, equipo de fútbol). Los atributos a mostrar de cada componente son: Apodo, Nombre y Tema. Para hacer pruebas inserta algún apodo y tema con componentes que ya existan en la BD.
- c) (para nota) Repite la búsqueda anterior para puntuaciones intermedias: más de 4 y menos de 9. Muestra la puntuación también.
- d) Describe al menos cuatro consultas interesantes para tí, descríbela primero el texto y luego su código y ejecútalas.
- e) Obtener todos los componentes de tu colección clasificados por tema.
- e.1) Queremos imprimir una línea para cada documento de este modo:  
TEMA: <su nombre> - NombreDoc: <del documento>.  
Usa un cursor que llama a una función sin nombre (definida dentro del cursor). Esa función imprime cada línea.
- e.2) Agrupa con *aggregate* para obtener un documento con el nombre de cada tema y una lista de nombres de sus documentos, además de cuántos documentos son.
- f) Rebaja un 10% al precio de todos los componentes peor valorados (puntuación  $< 7$ ). Y en la misma actualización añades el atributo *Descuento* a todas las aficiones: su valor no es un porcentaje fijo, sino que se lo asignas tú de acuerdo a esta regla: cuanto mayor puntuación, menor % de descuento (inventa la fórmula). Usa *forEach* y una *function* con *save*.
- g) (para nota) Queremos tener una colección *PorNivel* donde vas a crear (cuatro documentos) que representan cuatro *niveles de calidad calculados*, cada uno tendrá un vector con los documentos de *aficiones* que le correspondan. Sigue estos pasos:
1. Crear la colección *PorNivel*. Cada documento tendrá estos campos:  
*NomCal*: Nombre del intervalo de calidad, con uno de estos valores : nivel\_1, nivel\_2, nivel\_3 y nivel\_4  
*Componentes*: un array/vector que contenga, como elementos, los componentes (documentos) de la colección *aficiones* que correspondan a ese nivel. Además, cada elemento del array, debe tener un campo con el *valor de calidad calculado* del componente, obtenido multiplicando por 10 la *puntuación* del componente. Ese campo es el valor usado para asignarle un nivel, dentro de estos cuatro intervalos de valores:  $0 \leq \text{nivel\_1} \leq 30$ ,  $30 < \text{nivel\_2} \leq 50$ ,  $50 < \text{nivel\_3} \leq 70$  y  $70 < \text{nivel\_4}$ .
  2. Carga en la colección *PorNivel* todos los componentes de la colección *aficiones* que correspondan.
  3. Imprime el contenido de la colección *PorNivel* formateado, poniendo una línea de guiones entre cada nivel.
  4. Consulta *PorNivel* para obtener los 5 elementos más baratos independientemente del nivel al que pertenezcan: su nombre, su precio y su *NomCal*
  5. Elimina los 2 componentes más caros de cada intervalo (*NomCal*). Puedes hacer una operación separada para cada intervalo.

#### APARTADO 4.- Usando Colecciones limitadas (capped)

Queremos mantener en una colección *superGuai*, los 5 mejores componentes de la colección *aficiones*. Para ello hacemos lo siguiente:

- a) Crear dicha colección
- b) Crea las operaciones necesarias para poner los 5 mejores elementos de acuerdo al criterio de calidad explicado en el apartado 3.g., **pero usando la colección *aficiones* no uses *PorNivel***. Antes de obtener los mejores elementos: calcula el nivel de calidad y crea el atributo *Calidad* y su valor, actualizando la colección *aficiones*.
- c) Inserta un elemento a mano.
- d) Lista todos los componentes para comprobar que mantiene los último cinco introducidos

## **APARTADO 5.-**

Siguiendo las pautas para diseñar una BD no-sql en las diapositivas de la Teoría: diseña tú una BD de tema libre y describe qué operaciones quieres hacer. Debes tener en cuenta que sea un tema donde una BD tipo SQL **no sea adecuada**.

## **APARTADO 5.- EXTRA**

Deseamos introducir elementos compuestos, ej.: como en un equipo de futbol si incluimos cada jugador con sus datos personales. Y queremos hacer muchas consultas sobre esos elementos compuestos ej.: datos personales. ¿Conviene normalizar o desnormalizar?. ¿Cómo debería quedar la representación de la colección?