

AMPLIACION de BASES DE DATOS

(Profesor: Héctor Gómez Gauchía)

Práctica 5 semana 12: Utilizar una BD en MongoDB

→ QUÉ SUBIR al CV): un .zip con lo siguiente

- Un SOLO documento PDF que describa de cada apartado:
- la solución,
- el resultado después de la ejecución en formato texto

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

- Consultas: (*componente* es cada documento de la BD)
 - Obtener el nombre, el tema y la puntuación de los componentes mejor valorados (puntuaciones ≥ 9) pero no saques el identificador.
 - 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.
 - para cada una de las puntuaciones (10,9,8,7,6 y 5) por separado: Para cada puntuación, obtener los nombres de componentes que la tienen. (cuando ya funcione, incluye también el nombre su tema)
 - Lista de Apodos para cada afición (Tema)
- (para nota) A quien le gusta los mismos componentes?: Queremos obtener para cada Tema: Apodo, Nombre y Tema de los componentes (que tengan apodo distinto) en los que coincide al menos un nombre del mismo Tema.
- (para nota) Repite la búsqueda anterior para puntuaciones intermedias: más de 4 y menos de 9. Muestra la puntuación también.
- Describe al menos cuatro consultas interesantes para tí, descríbela primero el texto y luego su código y ejecútalas.
- Obtener todos los componentes de tu colección clasificados por tema.
 - 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.
 - Agrupar 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.
- 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`.
- (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:
 - Crear la colección *PorNivel*. Cada documento tendrá estos campos:
NomCal: Nombre del intervalo de calidad : 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 $70 < \text{nivel_4}$.
 - Carga en la colección *PorNivel* todos los componentes de la colección *aficiones* que correspondan.
 - Imprime el contenido de la colección *PorNivel* formateado, poniendo una línea de guiones entre cada nivel.
 - Consulta *PorNivel* para obtener los 5 elementos más baratos independientemente del nivel al que pertenezcan: su nombre, su precio y su *NomCal*
 - Elimina las 2 aficiones más caras 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., **no** hace falta que hagas el apartado 3.g.: calcula el nivel de calidad y asigna su valor en *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 quiere hacer. Teniendo 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?