

AMPLIACION de BASES DE DATOS

(Profesor: Héctor Gómez Gauchía)

Práctica 5 semana 11: Introducción a MongoDB, una base de datos noSQL

→ QUÉ CREAR (para conservar y poder estudiar para el examen):

- 1.- El dump de tu BD.
- 2.- Un documento que describa en cada apartado:
 - la solución,
 - el resultado después de la ejecución
- Incluye también lista de dudas que te han surgido y no has resuelto al terminar la práctica

APARTADO 1.- Conocer los elementos básicos para trabajar con MongoDB, ejecutando ejemplos

- Cómo arrancar y empezar: sigue las instrucciones del archivo *empezando-con-MongoDB.pdf*
- Para hacer la práctica se recomienda consultar:
 - o Ejemplos de instrucciones principales en el mismo archivo y en diapositivas de teoría.
 - o Ejemplos más elaborados en */ejemplosMongodb-en-Teoria/ejemplos-Libro-MongoDB-v3.pdf*

APARTADO 2.- Utilizar una BD de Aficiones *AficionesBD* para relacionar personas como en una Red Social

Objetivo: Queremos hacer una Red Social de datos de Hobbies o Aficiones. Para ello empezamos con la BD de las aficiones de los alumnos de la clase del año pasado. Luego añades tus aficiones y haces consultas para descubrir quienes tienen gustos parecidos, planear eventos por grupos afines, etc.

a) Cargar los datos de “aficiones del año pasado”: Para ello sigue estos pasos

- Una vez arrancada MongoDB, como se indica en el archivo *Empezando-con-MongoDB.pdf*:
- En la shell de Robo3T teclea: **use pracmongo** //para crear una BD con ese nombre
- En la ventana (CMD) en el labo es “entorno Mongo”: (en tu portatil recuerda poner el path correcto)

```
mongorestore --collection aficiones --db pracmongo
X:\...\dump-para-empezar-prac\aficiones.bson
```

- En la shell de Robomongo
db.aficiones.count(); // Debes tener bastantes documentos

b) Crear tus propios contenidos de la BD.:

- 1.- Escoge una de tus aficiones favoritas.
- 2.- Escoge tus componentes favoritos de esa afición (*cada componente es un documento distinto*): si es música pues tus grupos favoritos, si es cocinar pues tus recetas favoritas, si es esquiar pues tus estaciones favoritas
- 3.- En un editor de texto, escribe la información de cada componente en un formato que se pueda importar en MongoDB de acuerdo a las normas indicadas más abajo. Puedes ver el formato en el fichero *aficiones-restaurantes.json*

Normas para Formato de los Componentes de Aficiones: (puede ser una nueva o alguna de las aficiones ya creadas)

- Campos Obligatorios:
 - o *Tema:* sobre lo que es tu Afición. EJ: música, esquí, cine, cocina, fútbol
 - o *Apodo:* por privacidad, no uses tu nombre real porque los datos van a ser públicos.
 - o *Nombre:* único del componente. EJ: Nombre equipo, nombre de grupo, título de película o de libro
 - o *Puntuación:* de ese componente según tu gusto (máximo: 10)
 - o *Precio:* lo que cuesta disfrutar de eso. EJ: precio si es un CD, una película, del concierto de un grupo, etc.
- Campos específicos tuyos sobre tu componente: Debes incluir características propias de esa afición que te interesen representar.
 - o Si son películas: género, director, etc.

IMPORTANTE: Si tu afición es una de las que ya están creadas: usa el mismo nombre en el campo “Tema”.

- c) Carga los datos de tu Afición, insertando (sobre la BD con Botón dcho + insert) cada componente, que es un documento. Inserta al menos diez documentos.
- d) Hacer varias consultas básicas siguiendo las que hay en el archivo *Empezando-con-MongoDB.pdf*

Práctica 5 semana 12: 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 ≥ 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 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)
 - 4. Lista de Apodos para cada afición (Tema)
- b) (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.
- 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. Para imprimirlos de una manera ordenada, usa un cursor que llama a una función sin nombre (definida dentro del cursor). Esa función imprime cada ítem que, previamente hemos transformado a json (así le damos un formato más legible)
- 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. El valor se lo asignas tú de acuerdo a esta regla: cuanto mayor puntuación, menor % de descuento.
- g) (para nota) Crea una colección *PorNivel* donde vas a crear cuatro *niveles de calidad calculados* de acuerdo a esta fórmula: $(\text{puntuación} * 100) / \text{precio}$.
 - 1. Prueba a ajustar tú la ecuación para que sea realista y consiga valores distintos que estén dentro de 4 intervalos. Escoge esos cuatro intervalos de valores, p.e.: entre 0 y 30, más de 30 y menos de 50, etc.

La colección *PorNivel* tendrá los campos:

NomCal: Nombre de intervalo de calidad

Componentes: un array/vector que contenga, como elementos, los componentes de la colección *Aficiones* que correspondan a ese intervalo. Además, cada elemento del array, debe tener el *valor de calidad calculado* del componente.

- 2. Carga en la colección *PorNivel* todos los componentes de la colección *Aficiones* que correspondan.
- 3. Imprime su contenido formateado que se pueda leer bien, para comprobar que es correcto.
- 4. Consulta *PorNivel* para obtener los elementos más baratos: su nombre, su precio y su *NomCal*
- 5. Elimina las 2 aficiones más caras de 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. ¿Conviene normalizar o desnormalizar?. ¿Cómo debería quedar la representación de la colección?