BASES DE DATOS Fecha Entrega 31-05-2017

Profesor: Héctor Gómez Gauchía

Práctica: Introducción a MongoDB, una base de datos noSQL

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

Preparación para trabajar con MongoDB:

- Repasar y ejecutar **el** archivo *empezando-con-MongoDB.pdf* (no entregues nada)

- 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.

APARTADO 1.- Utilizar una BD de Aficiones *AficionesBD* para relacionar personas como en una Red Social

a) Crear tus propios contenidos de la BD. Puedes hacer antes el apartado **b)** para usarlo como modelo:

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:

- Campos Obligatorios:
  - *Tema*: sobre el que es tu *Afición*. EJ: música, esquí, cine, cocina, futbol
  - *Apodo*: no uses tu nombre real porque los datos van a ser públicos en las wikis
  - *Nombre*: del componente. EJ: Nombre equipo, nombre de grupo, título de película o de libro
  - *Puntuación:* de ese componente según tu gusto (máximo: 10)
  - *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: Características propias de esa afición que te interesen representar.
  - Si son películas: género, director, etc.

b) Cargar los datos de Aficiones del año pasado: Para ello sigue estos pasos

- Una vez arrancada MongoDB, como se indica en en *Empezando-con-MongoDB.pdf* :

- En la shell de Robomongo teclea: `use prac1617` //para crear esa BD

- En la ventana (CMD) del entorno Mongo: (en tu portatil recuerda poner el path correcto)

```
mongorestore
      --collection aficiones --db prac1617 A:\...dump-para-empezar-prac\aficiones
```

- En la shell de Robomongo

```
db.aficiones.count();
```
// Debes tener 89 documentos

c) Carga los datos de tu Afición, insertando cada componente (cada documento)

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

APARTADO 2.- Trabajando con la colección *Aficiones* en MongoDB

a) De cada *Tema* :
   1. Obtener los nombres de los componentes mejor valorados (puntuaciones > = 9)
   2. Calcula cuanto te gastarías si vas a todos los temas mejor valorados.
   3. Obtener los componentes valorados para cada uno de los valores (10,9,8,7,6 y 5) por separado
   4. Lista de Apodos con esa afición (Tema)

b) Lista los Apodos, Componente y Tema en los que coincide al menos un componente (el nombre) del mismo Tema.

c) 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 ti, escribe el 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) Crea una colección *PorNivel* donde vas a crear cuatro *niveles de calidad calculados* de acuerdo a esta fórmula:
   `(puntuación * 100) / 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 3.- 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 del apartado anterior (apartado 2.g).
c) Inserta un elemento a mano.
d) Lista todos los componentes para comprobar que mantiene los último cinco introducidos

APARTADO 4.- 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?