

Sesión 5 – Manejo de una base de datos con información multimedia en MongoDB.

Esta sesión la dividiremos en dos actividades adicionales a las tres realizadas en la sesión anterior. En la cuarta relacionaremos la información de las farmacias y sus archivos multimedia, y en la quinta extraeremos la información multimedia de la base de datos.

Introducción.

Recordamos la base de datos a desarrollar. Estaba formada por:

1. Una tabla maestra con las farmacias, cuya estructura es:
 - "documentName" : Propietario de la farmacia
 - "documentDescription" : Dirección. Es la concatenación de las columnas "sanidadStreet" (calle), "sanidadPostalCode" (código postal), y "SanidadTown" (municipio), con un espacio en medio de cada dato para separar la información.
 - "sanidadCenterType" : Tipo de centro. Siempre es "Farmacia",
 - "sanidadCode" : Código de tipo de centro. Siempre es "farmac_far".
 - "sanidadName" : Nombre de la farmacia.
 - "sanidadPostalCode" : Código postal
 - "sanidadProvince" : Provincia
 - "sanidadRegion" : Región. Siempre está en vacío.
 - "sanidadStreet" : Calle.
 - "sanidadStreetSnap" : Siempre viene en blanco.
 - "sanidadTown" : Municipio.
2. Un conjunto de fotos de algunas de las farmacias (cada farmacia puede tener ninguna, una o varias fotografías). Las fotos pueden ser del interior o del exterior de la farmacia.

En la sesión de hoy implementaremos la colección "fotografías".

Preámbulo.

Sigue los pasos que se describen a continuación para crear el entorno necesario para realizar los ejercicios de esta sesión.




Paso 1: Descarga ConEmu de los materiales del campus virtual de la sesión anterior, y crea las mismas tres pestañas que usamos entonces. Esto es:

- La <1> la usaremos para tener el servidor de bases de datos activo.
- La <2> para ejecutar la aplicación cliente de manejo de la base de datos.
- La <3> para ejecutar comandos del sistema operativo desde la consola.

Paso 2: Mira la documentación de la sesión anterior, y pon en marcha el servidor MongoDB en la pestaña <1> de ConEmu. Recuerda que se hace ejecutando el comando `mongod.exe` en la carpeta donde está instalado el gestor de base de datos.

Paso 3: Mira la documentación de la sesión anterior, y pon en marcha el shell de MongoDB en la pestaña <2> de ConEmu. Recuerda que se hace ejecutando el comando `mongo.exe` en la carpeta donde está instalado el gestor de base de datos.

Paso 4: descarga el paquete de materiales que corresponde a esta sesión del campus virtual, y descomprímelos en una carpeta del escritorio de Windows. Abre la carpeta donde has depositado los materiales, y verás estos ficheros:

Nombre	Tipo	Tamaño comprimido	
 Exterior1	Archivo JPG	61 KB	I
 farmacias	Archivo JSON	30 KB	I
 Interior1	Archivo JPG	143 KB	I

Si no existe, crea la carpeta `c:\datosmongo` y copia estos tres ficheros ahí.

Paso 5: Comprueba si existe la base de datos denominada `dbdm` en el servidor. Para ello ve a la pestaña <2> de ConEmu y ejecuta el comando:

```
> show dbs
```

Si existiese, bórrala con estos comandos del shell:

```
> use dbdm  
> db.dropDatabase()
```

La base de datos de esta sesión es un poco diferente a la de la sesión anterior. Por eso la debes borrar y crearla de nuevo.

Paso 6: Ve a la pestaña <3> de ConEmu, e importa el archivo json a tu servidor de la misma forma que lo hiciste en la sesión anterior. Recuerda que para ello deberás usar el comando `mongoimport.exe` que vimos la sesión anterior.

Hecho esto, ya debes tener el entorno listo para realizar los ejercicios.

Actividad 4: Relacionando la información.

Ejercicio 9: Incorporación de información multimedia a la base de datos.

Abstract: Vamos a incorporar a la colección `farmacias` documentos con sus fotografías.

Selecciona la pestaña <3> de ConEmu. Ejecuta este comando:

cd "c:\Program Files\MongoDB\Server\3.2\bin" si usas tu propia máquina
cd "c:\mongodb\bin" si usas una máquina del laboratorio

MongoDB tiene una restricción importante a la hora de almacenar archivos grandes (los tipos de datos LOB que vimos en Oracle): El tamaño máximo de un documento es de 16 megabytes. Para solventar esto, se ha desarrollado una librería llamada Gridfs, que mediante el uso de dos colecciones almacena:

1. En la colección "files", los metadatos de los archivos guardados en nuestra base de datos.
2. En la colección "chunks", la propia información del archivo (texto o binaria), troceada en paquetes más pequeños (255 kilobytes por defecto), en caso de que el archivo a almacenar sea más grande que ese tamaño.

Vamos a incorporar archivos desde carpetas del sistema operativo hacia la base de datos, y vice versa. Almacenarás en la base de datos dos imágenes. La primera será la fachada de una de las farmacias, y la otra una fotografía del interior. Las dos fotografías se encuentran en la carpeta de datos de materiales, esto es c:\datosMongo. Ejecuta el comando que corresponda con estos parámetros:

NOMBRELOCAL = c:\datosMongo\Interior1.jpg
BASEDATOS = dbdm
PREFIXO = fotografías
ARCHIVO = **Interior1.jpg**

Con él, le estamos diciendo que incorpore a la base de datos llamada dbdm el archivo especificado, y en servidor al que estamos conectados (el que podemos ver en la pestaña de comandos <1>). Por ello no es necesario especificar ningún tipo de parámetro de conexión.

Incorpora ahora la otra fotografía, la de la fachada exterior, actuando de manera similar.

NOMBRELOCAL= **c:\datosMongo\Exterior1.jpg**
BASEDATOS = dbdm
PREFIXO = fotografías
ARCHIVO = **Exterior1.jpg**

A continuación consulta los metadatos de las imágenes que se contienen en la colección fotografías.files. Para ello ejecuta una llamada a la función **find** para esa colección (esa función la vimos en la sesión anterior), en la ventana de comandos <2>. No pases ningún parámetro a la función de búsqueda, es decir, no uses ningún FILTRO, y MUESTRA todas las columnas de la colección.

Observa los metadatos que se almacenan almacena la colección fotografías.files.

Ejercicio 10: Vincular mediante el patrón "Relacionar" los archivos multimedia con un documento.

Abstract: Vamos a embeber y a relacionar las imágenes que hemos incorporado a nuestra base de datos, usando la farmacia con número de ID 1.

Nuestra base de datos actual contiene lo siguiente:

1. La información de las farmacias en la colección "Farmacias".
2. Dos archivos de imágenes, en las colecciones fotografias.files y fotografias.chunks.

Vamos a implementar la relación uno a muchos entre la farmacia y sus imágenes utilizando el patrón "Relacionar". Usaremos este patrón y no el de "incrustar" en este caso, dado que la información de las imágenes está almacenada en las colecciones fotografias.files y fotografias.chunks (en tablas distintas, traducido a lenguaje relacional). Tomemos para ello los IDs de las dos imágenes. Los podemos ver en el resultado del ejercicio anterior. Debes tomar los valores del atributo "_id" de cada imagen en la colección fotografias.files. Para poder seguir explicando el ejercicio, denominaremos a esos dos ids como **IDIMAGEN1** e **IDIMAGEN2**. Sustituye esos valores más adelante por los que hayas obtenido en la sentencia find que has ejecutado anteriormente sobre tu base de datos.

A continuación, incluye estos dos valores como un array embebido en el documento con "_id" con valor 1 de la colección farmacias.

Por ello obtén primeramente todos los datos del documento farmacias con _id valor 1. A continuación, incluye el array con las referencias a las imágenes, que llamaremos imágenes. La sentencia de modificación que ejecutaremos será update, tal como vimos en la sesión anterior. Recuerda que la función update en MongoDB sustituye un documento por otro, con que debemos incluir todos los datos del mismo. Los parámetros que debes usar son:

CONDICIONES = _id:1

```
DATOS = db.farmacias.update({_id: 1},{documentName: "PEREZ GARCIA, Olga Isabel",documentDescription : "Rodriguez Arias, 43 48011 Bilbao",sanidadCenterType : "Farmacia",sanidadCode : "farmac_far",sanidadName : "Farmacia: PEREZ GARCIA, Olga Isabel",sanidadPostalCode : "48011",sanidadProvince : "Bizkaia",sanidadRegion : "",sanidadStreet : "Rodriguez Arias, 43",sanidadStreetSnap : "",sanidadTown : "Bilbao",imagenes: [{Id_imagen: ObjectId("IDIMAGEN1"), Imagen_tipo: "Interior"}, {Id_imagen: ObjectId("IDIMAGEN2"), Imagen_tipo: "Exterior"}]})
```

Consulta ahora el documento modificado usando la sentencia find, y filtrando por el _id con valor 1.

Ejercicio 11: Extraer dos documentos de la base de datos, con estructura diferente.

Abstract: Vamos a realizar una consulta, que obtenga dos documentos de una misma colección, pero que tienen estructura distinta.

Fíjate que si consultamos con esta sentencia dos documentos de la misma colección, podremos comprobar que tienen estructura distinta:

```
db.farmacias.find({$or:[{_id:1},{_id:2}]}).pretty()
```

Identifica los atributos diferentes en ambos documentos de la colección farmacias. ¿Qué tendrías que haber hecho en una base de datos relacional para poder tener estructuras diferentes en filas de la misma tabla? ¿Habrías podido hacerlo?

Actividad 5: Extraer información multimedia de la base de datos.

Ejercicio 12: Obtener un archivo de la base de datos, y extraerlo a una carpeta del sistema operativo.

Abstract: Vamos a extraer un fichero de imagen que previamente hemos guardado en la colección de imágenes, y a guardarlo en una carpeta del sistema operativo.

Activa la ventana <3> de comandos, y asegúrate que estás en la carpeta que corresponde. Ejecuta el comando adecuado para obtener la imagen de la base de datos. Los parámetros de ese comando deben ser:

NOMBRELOCAL = Interior1extraida.jpg

BASEDATOS = dbdm

PREFIJO = fotografías

ARCHIVO = **Interior1.jpg**

Fin de la sesión.

Una vez terminados los ejercicios, deberemos cerrar adecuadamente el servidor. Para ello, ve a la pestaña <2> y ejecuta este comando:

```
db.getSiblingDB("admin").shutdownServer()
```

Los servidores de bases de datos deben ser desconectados de manera controlada, para que puedan terminar todas las transacciones pendientes, y guardar las cachés.

Hecho esto, cierra las tres sesiones de ConEmu.