

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 películas 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 películas de Imdb.com, cuya estructura es:
 - adult: Si tiene contenido para adultos o no
 - belongs_to_collection: Datos de la saga de la película, si pertenece.
 - budget: Presupuesto
 - genres: Género (Comedia, acción, etc.)
 - homepage: Página web.
 - id: Código único de la película.
 - imdb_id: Código único de la película en la base de datos IMDB.
 - original_language: Idioma original.
 - original_title: Título original.
 - overview: Resumen.
 - oopularity: Índice de popularidad.
 - poster_path: Nombre del archivo del póster (no lo tenemos).
 - production_companies: Productoras.
 - production_countries: Países de producción.
 - release_date: Fecha de lanzamiento.
 - revenue: Ingresos.
 - runtime: Duración.
 - spoken_languages: Idiomas.
 - status: Estado.
 - tagline: Etiqueta.
 - title: Título.
 - video: Booleano que indicas si es de cine o de video.
 - vote_average: Promedio de evaluaciones entre todos los usuarios
 - vote_count: número de usuarios que la han votado.
2. Un conjunto de fotos de los carteles de algunas de las películas (cada película puede tener ninguna, una o varias fotografías). Las fotos pueden estar de un idioma diferente.

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.

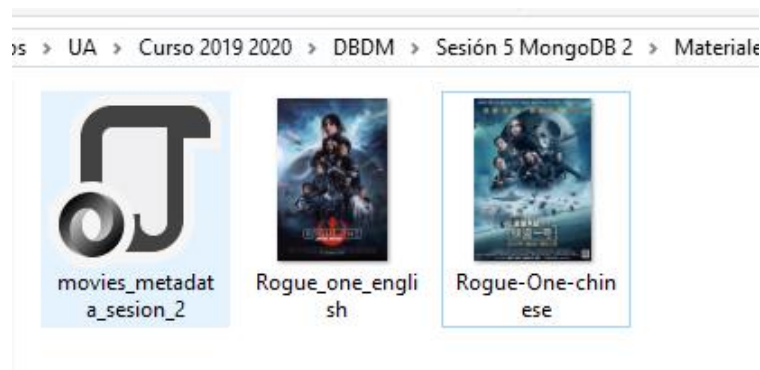
Paso 1: Descarga ConEmu de los materiales del campus virtual de la sesión anterior, y crea las mismas tres pestañas que usamos entonces. Esto es:

- 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:



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 entonces.

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 movies 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 (de 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 viceversa. Almacenarás en la base de datos dos imágenes. La primera será el póster de la película "Rogue One" en inglés, y la otra lo mismo, pero en chino. 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\poster_rogue_one_ingles.jpg**
BASEDATOS = dbdm
PREFIJO = fotografías
ARCHIVO = **poster_rogue_one_ingles.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\poster_rogue_one_chino.jpg**
BASEDATOS = dbdm
PREFIJO = fotografías
ARCHIVO = **poster_rogue_one_chino.jpg**

A continuación consulta los metadatos de las imágenes que se contienen en la colección fotografias.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 película con número de Id 330459 (El de Rogue One).

Nuestra base de datos actual contiene lo siguiente:

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

Vamos a implementar la relación uno a muchos entre la película 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 330459 de la colección movies.

Por ello obtén primeramente todos los datos del documento movies con id valor 330459. A continuación, incluye el array con las referencias a las imágenes, que llamaremos "imagenes". 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: 330459}

```
DATOS = { "adult": "False", "belongs_to_collection": "{ 'id': 10, 'name': 'Star Wars Collection', 'poster_path': '/ghd5zOQnDaDW1mxO7R5fXXpZMu.jpg', 'backdrop_path': '/d8duYyyC9J5T825Hg7grmaabfxQ.jpg' }", "budget": 200000000, "genres": "[ { 'id': 28, 'name': 'Action' }, { 'id': 12, 'name': 'Adventure' }, { 'id': 878, 'name': 'Science Fiction' } ]", "homepage": "http://www.starwars.com/films/rogue-one", "id": 330459, "imdb_id": "tt3748528", "original_language": "en", "original_title": "Rogue One: A Star Wars Story", "overview": "A rogue band of resistance fighters unite for a mission to steal the Death Star plans and bring a new hope to the galaxy.", "popularity": 36.567575, "poster_path": "/qjiskwlv1qQzRCjpV0cl9pEMF9a.jpg", "production_companies": "[ { 'name': 'Lucasfilm', 'id': 1 }, { 'name': 'Walt Disney Pictures', 'id': 2 }, { 'name': 'Allison Shearmur Productions', 'id': 47706 }, { 'name': 'Black Hangar Studios', 'id': 68738 }, { 'name': 'Stereo D', 'id': 86561 } ]", "production_countries": "[ { 'iso_3166_1': 'US', 'name': 'United States of America' } ]", "release_date": "2016-12-14", "revenue": 1056057273, "runtime": 133, "spoken_languages": "[ { 'iso_639_1': 'en', 'name': 'English' } ]", "status": "Released", "tagline": "A Rebellion Built on Hope", "title": "Rogue One: A Star Wars Story", "video": "False", "vote_average": 7.4, "vote_count": 5111, "imagenes": [ { "Id_imagen": ObjectId("IDIMAGEN1"), "poster_language": "en" }, { "Id_imagen": ObjectId("IDIMAGEN2"), "poster_language": "zh" } ] }
```

Consulta ahora el documento modificado usando la sentencia find, y filtrando por el id con valor 330459.



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 de documento distinta.

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

```
db.movies.find({$or:[{id:101665},{id: 407559}]}).pretty()
```

Identifica los atributos diferentes en ambos documentos de la colección movies.

¿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 = c:\datosmongo\poster_rogue_one_chino_extraida.jpg  
BASEDATOS = dbdm  
PREFIJO = fotografias  
ARCHIVO = poster_rogue_one_chino.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.