Diseño de Bases de Datos Multimedia

Un gestor de BD NoSQL: MongoDB (II)

Grado en Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

Departamento de
Lenguajes y Sistemas Informáticos

ÍNDICE

- Recordando el almacenamiento de LOB en bases de datos.
- Diseño de BD en MongoDB. Modelado.
- LOB en MongoDB. La librería GridFS.
- Usos prácticos de NoSQL y MongoDB.
- Estructura de la práctica a realizar (II).
- Guion de la práctica.

Almacenamiento de tipos de datos LOB

- Recordemos que LOB significa "Large OBjects". Son tipos de datos utilizados para poder almacenar grandes volúmenes de información para un solo dato. Una sola columna de una fila podría tener un tamaño de varios Gigabytes, o incluso más.

- Recordemos también que podemos almacenar información de LOB en la propia base de datos (guardaríamos todos los datos dentro de ella), o como una referencia a la ubicación del archivo (guardaríamos la ubicación de dónde encontrar el fichero con los datos).

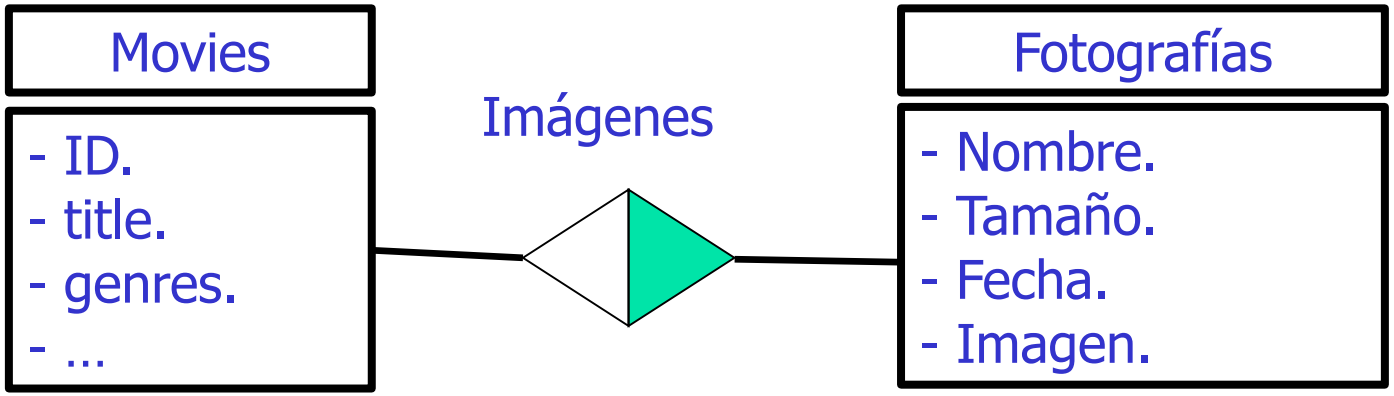
Diseño de BD MongoDB. Modelado.

Existen 2 patrones de modelado:

Embeber: incrusta documentos uno dentro de otro como parte del mismo documento, creando subdocumentos anidados o jerarquías de documentos.

Referenciar: imita el comportamiento de las claves ajenas en un SGBDR, esto es, incluimos el código de documento referenciado en el documento correspondiente.

Relaciones 1:N (1 de 3)



Podemos usar cualquiera de los dos patrones:

Embeber:

Dos opciones:

1) Incrustar los documentos tipo "Fotografías" dentro de los documentos tipo "movie" (un vector de fotografías para cada película).

2) Incrustar el mismo documento "movie" dentro de cada uno de los documentos tipo "Fotografía" (un documento movie por cada fotografía).

Relacionar:

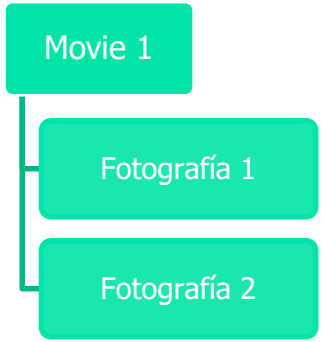
Dos opciones:

1) Referenciar el código de fotografía en cada documento tipo "movie".

2) Referenciar los códigos de película en el documento tipo "Fotografías".

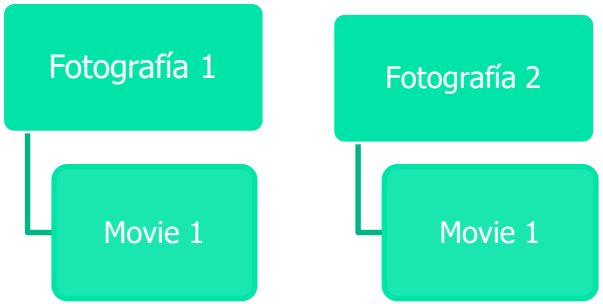
Relaciones 1:N (2 de 3)

Embeber opción 1:

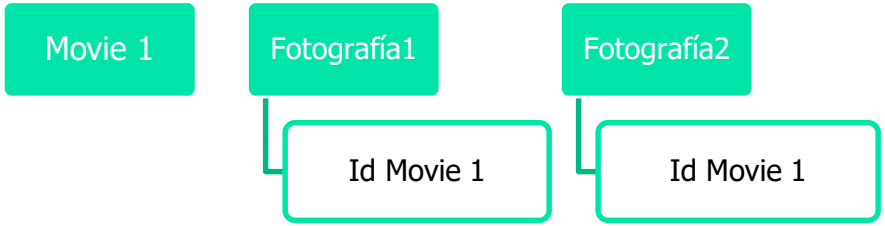


Relacionar opción 1:

Embeber opción 2:



Relacionar opción 2:



Relaciones 1:N (3 de 3)

¿Cuándo usar embeber o relacionar?

Intentaremos siempre utilizar embeber si no cambian mucho los documentos a incrustar. Tengamos en cuenta que para relacionar deberemos hacer dos consultas a la base de datos ya que en MongoDB no existen JOINS. En el ejemplo, usando embeber con una sola lectura obtendríamos una pelicula y todas sus fotografías.

Usaremos relacionar si cambian mucho los documentos a incrustar. En ese caso, es aconsejable utilizar la segunda opción antes indicada. En el ejemplo, incrustaríamos un vector de referencias a los documentos tipo "fotografías" dentro de cada documento tipo "movie"

LOB en MongoDB. GridFS

- GridFS es una librería que sirve para guardar ficheros grandes (LOB) en MongoDB.
  - Aunque MongoDB permite guardar datos binarios en objetos BSON, su límite en tamaño para cada documento es de 16 Megabytes.
- Ofrece un mecanismo para dividir de manera transparente al usuario un fichero grande partiéndolo en varios trozos.
  - Cada fichero que almacenemos tendrá un documento de metadatos en la colección `files`, y uno o más documentos chunk (trozos) en la colección `chunks`
- URL: http://www.mongodb.org/display/DOCS/GridFS

GridFS. Colección "files"

- Se usará para almacenar información de los ficheros.
- La estructura de la colección "files" es:

```
{   "_id" : <ObjectId>,   -- Identificador único de cada fichero
    "length" : <num>,     -- Longitud del fichero en bytes
    "chunkSize":<num>, -- Tamaño de cada trozo del fichero
    "uploadDate" : <timestamp>,  -- Fecha de carga del fichero
    "md5" : <hash>,       -- Checksum para comprobación de errores
    "filename" : <string>, -- Nombre del fichero
    "contentType" : <string>,  -- Tipo de contenido (MIME)
    "aliases" : <string array>,  -- Array de alias de nombre del fichero
    "metadata" : <dataObject>  -- Otros metadatos
}
```

GridFS. Colección “chunks”

- Se usará para almacenar los trozos de cada fichero.
- La estructura de la colección “chunks” es:

```
{   "_id" : <ObjectId>,          -- Identificador único de cada trozo
    “files_id" : <ObjectId>,     -- Clave ajena hacia la colección files
    “n“: <num>,                  -- Número de orden de cada trozo
    “data" : <binary>            -- Datos binarios del trozo
}
```

Usos prácticos. NoSQL y MongoDB.

- NoSQL es una alternativa a los gestores relacionales.
- Son útiles en determinados sistemas (no en todos los casos).
- Rápida puesta en marcha.
- Crecimiento flexible (replicación y particionado).
- GridFS: Almacenamiento de LOB en MongoDB.

Guion de la práctica

- Actividad 1: Configuración preliminar.
- Actividad 2: Puesta en marcha del sistema.
  - Ejercicio 1: Arranque del servidor.
  - Ejercicio 2: Monitorizar el servidor mediante un navegador Web.
  - Ejercicio 3: Instalar el shell de MongoDB.
- Actividad 3: Sentencias CRUD en MongoDB.
  - Ejercicio 4: Incorporar la colección (CREATE).
  - Ejercicio 5: Ejecución de consultas sencillas (READ).
  - Ejercicio 6: Ejecución de inserciones (INSERT).
  - Ejercicio 7: Modificando información (UPDATE).
  - Ejercicio 8: Borrando información (DELETE).
- Actividad 4: Relacionando la información.
  - Ejercicio 9: Incorporar información multimedia a la base de datos.
  - Ejercicio 10: Vincular imágenes a documentos.
  - Ejercicio 11: Consulta de documentos heterogéneos.
- Actividad 5: Extracción de información multimedia.
  - Ejercicio 12: Extraer un archivo desde BD hacia el Sistema operativo.

Guion para los ejercicios

- Para almacenar las fotografías, crearemos una serie de documentos anidados a los documentos de cada película.

- Para implementar nuestra base de datos de imágenes de las películass en MongoDB, usaremos los dos patrones:

  - El patrón "embeber": para incluir los datos de las fotografías de cada documento de la colección moviess.

  - El patrón "relacionar": dado que nuestros datos multimedia pueden tener un tamaño superior al máximo de 16 Megabytes admitidos por MongoDB. Vincularemos cada documento fotografía (dentro de la colección movies) al identificador del documento correspondiente en la colección "files" de la librería GridFS.

- Observa que GridFS usa el patrón "relacionar" en sus dos colecciones. "chunks" se relaciona con "files" por el atributo "files_id".

Comandos MongoDB

- Actividad 4: Creando los documentos de fotografías.
  - Ejercicio 9: Incorporar imágenes mediante GridFS.

mongofiles put --local NOMBRELOCAL --db BASEDATOS --prefix PREFIJO ARCHIVO

Donde:

NOMBRELOCAL = Nombre del archivo en el sistema operativo (debe existir, y debemos incluir la unidad de disco y la ruta).

BASEDATOS = Nombre de la base de datos MongoDB.

PREFIJO = Nombre del prefijo de las colecciones de GridFS. Si no ponemos este parámetro, por defecto la llamará fs.

ARCHIVO = Nombre archivo a almacenar en la colección files (la columna <filename> de esa colección tomará este valor).

Comandos MongoDB

- Actividad 5: Extrayendo información multimedia de la base de datos .
  - Ejercicio 12: Extraer imágenes mediante GridFS.

Mongofiles get --local NOMBRELOCAL --db BASEDATOS --prefix PREFIJO ARCHIVO

Donde:

NOMBRELOCAL = Nombre del archivo en el sistema operativo (incluyendo unidad y ruta destino).

BASEDATOS = Nombre de la base de datos MongoDB

PREFIJO = Nombre del prefijo de las colecciones de GridFS. Si no ponemos este parámetro, por defecto la llamará fs.

ARCHIVO = Nombre archivo a extraer de la colección files (la columna <filename> deberá tener este valor).