Javier Pérez Clemente y Belén González Guío

07/04/2019

Bases de datos no convencionales

Máster en Data Science|URJC

Descripción breve

MongoDB y Neo4J para base de datos DBLP Computer Science Bibliography, considerada como la mayor recopilación existente de referencias bibliográficas  
académicas específicamente centrada en la informática. Almacena los datos relativos a la gran mayoría de las revistas científicas y congresos académicos sobre informática.

Tabla de contenido

[**INTRODUCCIÓN** 2](#_Toc5133773)

[**PARTE I : MONGODB** 2](#_Toc5133774)

[**1 .- CAPTURA Y PROCESAMIENTO DE DATOS** 2](#_Toc5133775)

[**2.- ALMACENAMIENTO DE DATOS** 3](#_Toc5133776)

[**3.- ANÁLISIS DE DATOS** 3](#_Toc5133777)

[**PARTE II: NEO4J** 3](#_Toc5133778)

[**1 .- CAPTURA Y PROCESAMIENTO DE DATOS** 3](#_Toc5133779)

[**2.- ALMACENAMIENTO DE DATOS** 3](#_Toc5133780)

[**3.- ANÁLISIS DE DATOS** 3](#_Toc5133781)

# **INTRODUCCIÓN**

La fuente de datos que se va a utilizar en esta práctica es una fuente de datos real que recopila **referencias bibliográficas académicas**, concretamente relacionadas con la **informática**. Esta fuente de datos es conocida como **DBPL** Computer Science Bibliography. Los datos que vamos a procesar y, más tarde, almacenar para que puedan ser consultados, han sido descargados manualmente en un único **fichero** en formato **XML** del siguiente link: <https://dblp.uni-trier.de/xml/>. Este fichero comprimido ocupa alrededor de 450 MB y descomprimido pesa acerca de **2 GB**. (ME SURGE LA DUDA DE SI EL CÓDIGO EJECUTA SI NO ESTÁN LOS DOS ARCHIVOS DBLP.DTD Y DBLP.XML)

La información relacionada con cada artículo que podemos encontrar en este fichero es título del artículo, autor/autores, páginas, fecha de publicación, etc. Los datos recogen **8 tipos de artículos**, pero en nuestro caso vamos a **trabajar con 3** de ellos: **artículos de revista** (article), **artículos en congresos** (inproceedings) y **artículos en libros** (incollection).

# **PARTE I : MONGODB**

En esta primera parte de la práctica se va a trabajar con la **base de datos no relacional** **MongoDB**, que es de **tipo documental**. Las bases de datos orientadas a documentos se caracterizan porque almacenan los **valores como documentos** (normalmente en formato JSON o XML). De esta manera, la bases de datos no relacionales documentales almacenan todos los atributos de una entidad en un único documento, en vez de almacenar cada atributo de la entidad con una clave separada. Además, existe una gran **flexibilidad en la estructura de los documentos**, a diferencia de las bases de datos relacionales.

## **1 .- CAPTURA Y PROCESAMIENTO DE DATOS**

Como se mencionó antes, la captura de los datos se ha hecho descargando manualmente el archivo XML comprimido que se encuentra en la página web. Al ser un archivo tan pesado, no se ha podido cargar directamente con Python. Por lo tanto, para procesar tantos datos se ha hecho uso de **iterparse**, que construye el árbol pero te permite eliminar partes del árbol que ya no se necesiten para dejar hueco y procesar el siguiente trozo. En otras palabras, el XML se procesa por trozos para no ir acumulando todos los datos y poder liberar espacio.

Antes de procesar todos los datos, se han mirado cuales eran las consultas más frecuentes para organizar el esquema en función de estas consultas y para no procesar datos o atributos que no sean necesarios y que no se vayan a usar.

Lo primero ha sido **especificar** en el iterparse qué **tags** queremos procesar y cuáles no son de nuestro interés puesto que, como se comentó, aunque hay 8 tipos de artículos, solo estamos interesados en 3 : **article**, **inproceedings** e **incollection** . Además, antes de procesar los datos, se han analizado las consultas para evitar procesar atributos que no se van a usar o que no van a ser consultados y para poder definir un esquema. Se han seleccionado finalmente los siguientes 5 **atributos**: **‘id’, ‘type’, ‘date’, ‘title’** y **‘authors’.**

En resumen, hemos filtrado por los tag ‘article’, ‘inproceedings’ y ‘incollection’ y hemos seleccionado los atributos ‘id’, ‘type’, ‘date’, ‘title’ y ‘authors’.

Una vez que hemos seleccionado los campos que vamos a almacenar, se ha definido un esquema mirando las consultas que se tienen que realizar para poder entender cómo se usarán los datos. Si bien una base de datos documental es flexible y no exige que la estructura de todos los documentos sea idéntica, es importante definir un esquema porque la estructura de los documentos aunque no sea idéntica, tienen que tener una estructura común. Para ello hay que definir cuantas colecciones vamos a definir en esta base de datos de MongoDB. Normalmente, se eligen el número de colecciones en función de cuántos tipos de documentos difieren en su estructura. En nuestro caso, como tenemos tres posibles tipos de documentos (articles, inproceedings, incollection) pero tienen una estructura muy similar, hemos decidido crear una sola colección.

## **2.- ALMACENAMIENTO DE DATOS**

## **3.- ANÁLISIS DE DATOS**

# **PARTE II: NEO4J**

## **1 .- CAPTURA Y PROCESAMIENTO DE DATOS**

## **2.- ALMACENAMIENTO DE DATOS**

## **3.- ANÁLISIS DE DATOS**