

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

Actividad en Clase

Uso de Índices en MongoDB

Objetivos

Esta actividad te permitirá profundizar en el uso de índices sobre la base de datos de MongoDB. Realizarás una serie de ejercicios guiados que te ayudarán a comprender mejor la utilidad que tienen los índices en una base de datos documental, así como sus restricciones de uso.

Descripción

Para realizar esta actividad, se asume que cuenta con la instalación de MongoDB en tu ordenador. Utiliza o bien el cliente mongo o MongoDB Compass.

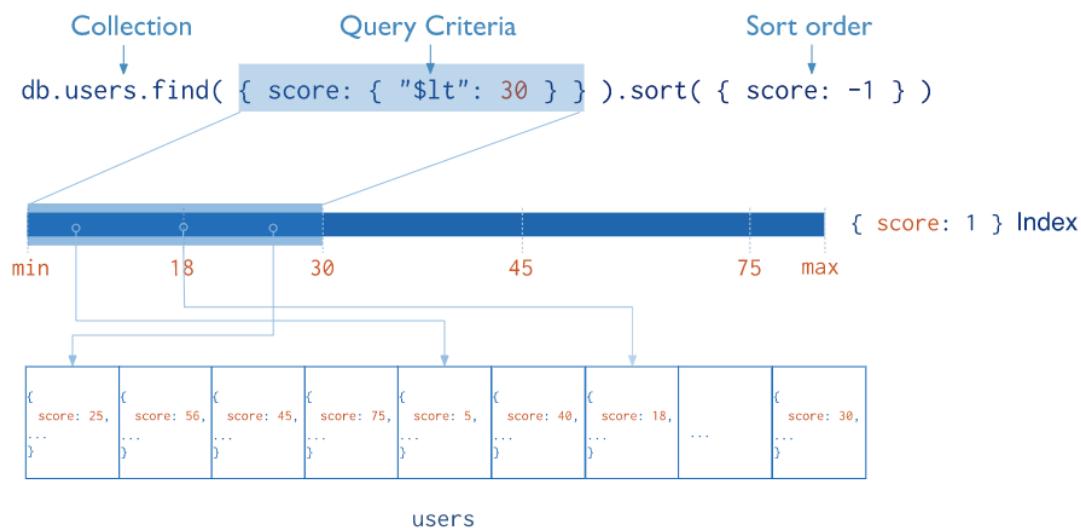
Índices

Los índices admiten la ejecución eficiente de consultas en MongoDB. Sin índices, MongoDB debe realizar un escaneo completo de la colección. Al escanear todos los documentos de una colección, consigue seleccionar aquellos documentos que coincidan con la declaración de consulta. Si existe un índice apropiado para una consulta, MongoDB puede usar el índice para limitar la cantidad de documentos que debe inspeccionar.

Los índices son estructuras de datos especiales que almacenan una pequeña parte del conjunto de datos de la colección en una forma fácil de recorrer. El índice almacena el valor de un campo específico o un conjunto de campos, ordenados por el valor del campo. **El orden de las entradas de índice admite coincidencias de igualdad eficientes y operaciones de consulta basadas en rangos.** Además, MongoDB puede devolver resultados ordenados utilizando el orden en el índice.

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

El siguiente diagrama ilustra una consulta que selecciona y ordena los documentos coincidentes mediante un índice:



Los índices en MongoDB son similares a los índices en otros sistemas de bases de datos. MongoDB define índices a **nivel de colección** y admite índices en cualquier campo o subcampo de los documentos de una colección MongoDB¹.

Recuerda: MongoDB crea un índice único en el campo `_id` durante la creación de una colección. El índice en `_id` evita que los clientes inserten dos documentos con el mismo valor para el campo `_id`.

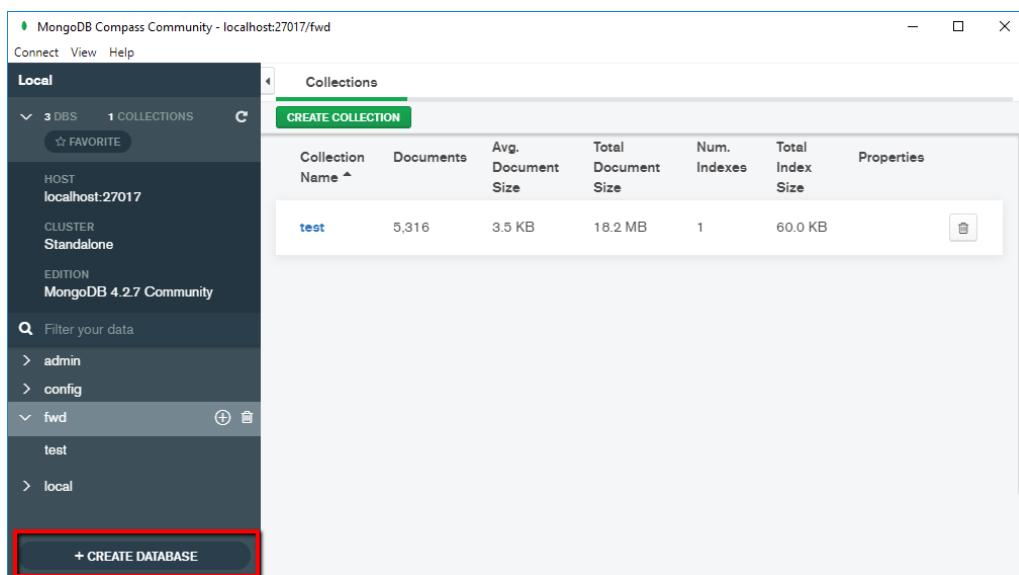
Prepara los datos

Se asume que cuenta con la instalación de MongoDB Compass, un IDE que viene incorporado con MongoDB y que facilita mucho más la interacción con la base de datos (*prohibido decir que es mejor esto a usar la terminal o consola*).

¹ Todo el apartado “Índices” es tomado de la documentación oficial de MongoDB <https://docs.mongodb.com/manual/indexes/>

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos: Nombre:	

- 1) Consulte la sección de datos abiertos de la Comunidad de Madrid. URL
https://datos.comunidad.madrid/catalogo/dataset/calidad_aire_estaciones
- 2) Descargue el fichero de los datos de calidad del aire bien en CSV o en JSON (a tu elección).
- 3) Abra MongoDB Compass, conéctese a su Base de Datos local y cree una base de datos nueva llamada *CalidadAire*.



MongoDB Compass Community - localhost:27017/fwd

Connect View Help

Local

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 4.2.7 Community

Filter your data

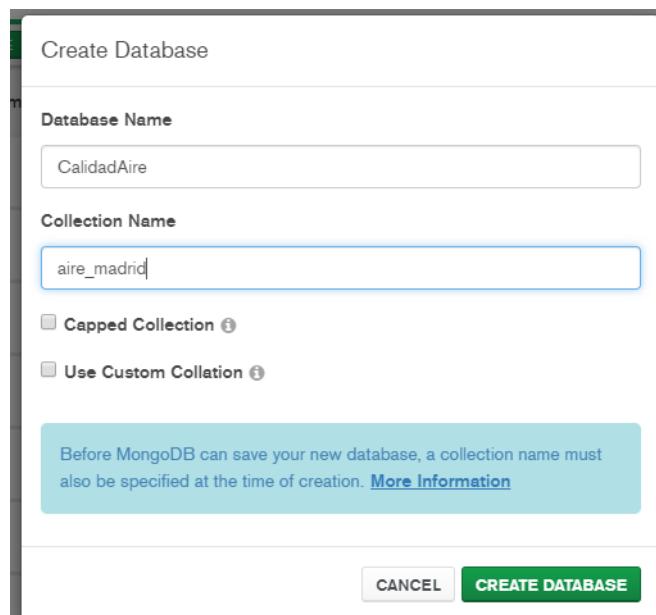
- > admin
- > config
- < fwd < + ☰
- > test
- > local

+ CREATE DATABASE

Collections

CREATE COLLECTION

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
test	5,316	3.5 KB	18.2 MB	1	60.0 KB	⚙️



Create Database

Database Name

CalidadAire

Collection Name

aire_madrid

Capped Collection ⓘ

Use Custom Collation ⓘ

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

CANCEL CREATE DATABASE

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos: Nombre:	

- 4) Cree una colección llamada *aire_madrid* e inserte el fichero descargado previamente. Asigna el tipo de datos correspondiente a cada campo del fichero.

estacion_codigo	zona_calidad_aire_descripcion	estacion_municipio	estacion_fecha
1 28005002	Zona 2 Corredor del Henares	Alcalá de Henares	2000/03/16 01
2 28006004	Zona 2 Corredor del Henares	Alcobendas	2000/01/19 01
3 28007004	Zona 3 Urbana Sur	Alcorcón	2008/09/11 01
4 28009001	Zona 2 Corredor del Henares	Algete	2007/03/31 01
5 28013002	Zona 3 Urbana Sur	Aranjuez	2001/10/06 01
6 28014002	Zona 2 Corredor del Henares	Arganda del Rey	2006/04/25 01
7 28016001	Zona 5 Sierra Norte	El Atazar	2006/08/22 01
8 28045002	Zona 4 Urbana Noroeste	Colmenar Viejo	2001/10/06 01
9 28047002	Zona 4 Urbana Noroeste	Collado Villalba	2006/05/31 01
10 28049003	Zona 2 Corredor del Henares	Coslada	2007/06/05 01

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

Trabaja con índices

Crear un **índice simple**:

- ▶ `db.collection.createIndex({ campo: -1 })` #1 ascendente -1 descendente

Crear un **índice compuesto**:

- ▶ `db.collection.createIndex({ campo1: 1, campo2: -1 }, { name: "indice_campo1_campo2" })`

Crear un **índice multiclave**:

- ▶ `db.collection.createIndex({ campo_array: 1 })` # donde campo_array es un array.

Crear un **índice de texto** simple y compuesto:

- ▶ `db.collection.createIndex({ campo: "text" })` #simple
- ▶ `db.collection.createIndex({ campo1: "text", campo2: "text" })`

5) Sobre la colección aire_madrid, cree un índice simple sobre cada uno de los campos:

- estacion_codigo
- zona_calidad_aire_descripcion
- estacion_municipio
- estacion_fecha_alta
- estacion_tipo_area
- estacion_tipo_estacion

6) Cree los índices compuestos sobre los campos:

- estacion_codigo y estacion_coord_UTM_ETRS89_x
- estacion_codigo y estacion_coord_UTM_ETRS89_y
- estacion_codigo y estacion_altitud

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos: Nombre:	

- 7) Sobre la colección `aire_madrid`, haga lo siguiente:
- `db.aire_madrid.find({$text: {$search: "Parque Europa"}})` ¿Qué ocurre?
 - Cree un índice de texto sobre el campo `estacion_direccion_postal`.
 - Vuelva a ejecutar la consulta anterior. ¿Qué ocurre ahora?
- 8) Ejecute ahora esta consulta:
- `db.aire_madrid.find({$text: {$search: "parque europa", $caseSensitive: true}})` ¿funciona?
 - `db.aire_madrid.find({$text: {$search: "parque europa", $caseSensitive: false}})` ¿y ahora?
- 9) Ejecute esta instrucción: `db.aire_madrid.getIndexes()` ¿Qué observas?
- 10) Ahora esta instrucción: `db.aire_madrid.createIndex({"$**": "text"})` ¿Funciona? ¿Hay error? ¿Podrías decir qué ocurre?
- 11) Borra el índice de texto creado antes:
- ```
db.aire_madrid.dropIndexes("estacion_direccion_postal_text")
```
- ¿Se llama así el índice?
- 12) Ejecuta nuevamente la instrucción del punto 10. ¿Funciona ahora? ¿Qué ocurría? ¿Cómo se llama el índice que has creado en ahora?

## Conclusiones

Una buena alternativa para mejorar el rendimiento de tus consultas es definir índices sobre los campos que más utilizarás en dichas operaciones.