<https://www.youtube.com/watch?v=D14wWW9EEx8>

¿Cómo funcionan?

En caso de db.collection(“books”.find({“rating”:10}), MongoDB intenta buscar todos los documentos que tengan “rating”:10. Si hay un montón de datos, no es eficiente.

En MongoDB, puede hacer index basado en rating field en colección de books, “por ejemplo, “Book\_Rating\_Index”.

Como ya tiene index de rating, sería más fácil buscar porque MongoDB sólo busca en el Index.

¿Cómo se aplican?

db.books.createIndex({ rating: 8})

y luego sale rating\_8 <- index

Para saber cuál Index está aplicado en bbdd,

db.books.getIndexies()

¿qué son los índices compuestos?

<https://www.mongodb.com/docs/manual/core/indexes/index-types/index-compound/>

Los índices compuestos recopilan y ordenan datos de dos o más campos en cada documento de una colección. Los datos se agrupan por el primer campo en el índice y luego por cada campo subsiguiente.

¿en qué casos debemos usarlos?

Si su aplicación ejecuta repetidamente una consulta que contiene múltiples campos, puede crear un índice compuesto para mejorar el rendimiento de esa consulta. Por ejemplo, un gerente de supermercado a menudo necesita buscar artículos de inventario por nombre y cantidad para determinar cuáles están bajos en stock. Puede crear un índice compuesto en los campos de artículo y cantidad para mejorar el rendimiento de la consulta.

¿son gratis o tienen algún coste? en caso de tener costes, ¿cuales son?

<https://www.mongodb.com/docs/manual/applications/indexes/>

The best indexes for your application must take a number of factors into account, including the kinds of queries you expect, the ratio of reads to writes, and the amount of free memory on your system. When developing your indexing strategy you should have a deep understanding of your application's queries. Before you build indexes, map out the types of queries you will run so that you can build indexes that reference those fields. Indexes come with a performance cost, but are more than worth the cost for frequent queries on large data sets. Consider the relative frequency of each query in the application and whether the query justifies an index.