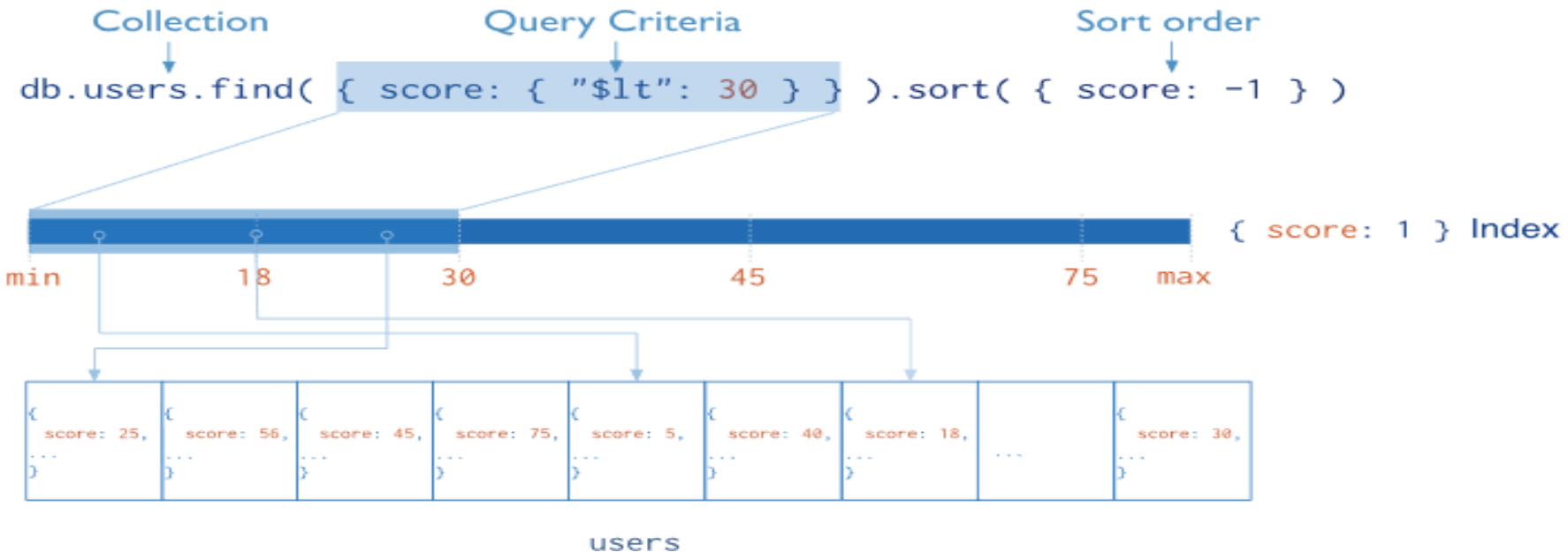
Indices en MongoDB

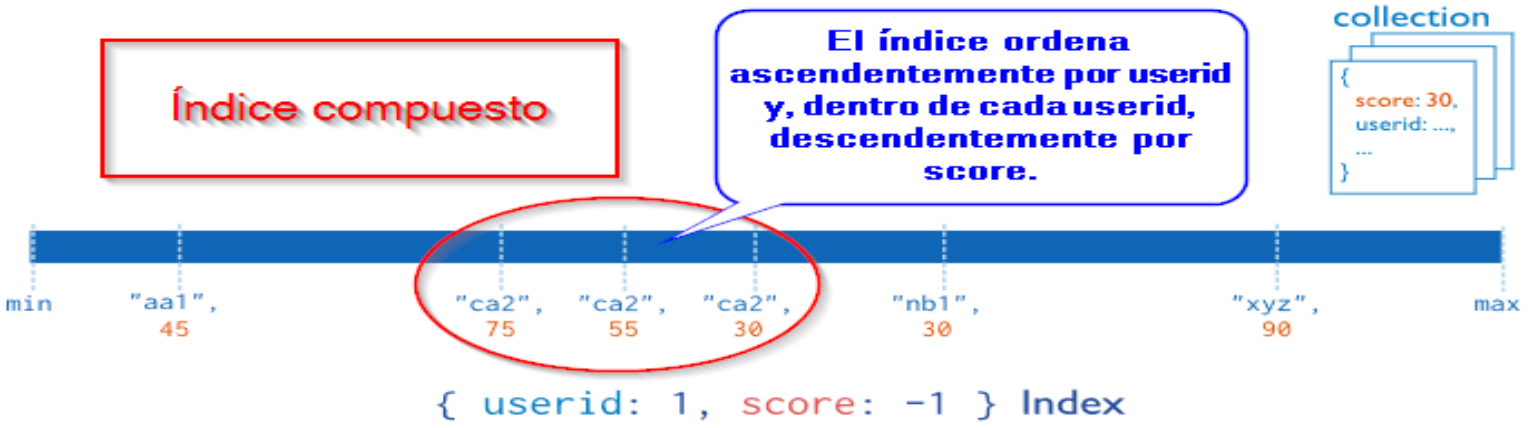
Los indices sirven para que las consultas de lectura sean mas rapidas.

– Un indice es una parte de los valores de los campos de una coleccion, almacenados en una estructura de tipo **B-tree** y **ordenados**, permitiendo a las consultas una rapida localizacion de los valores cuando se utilizan criterios de igualdad o de intervalo (rango).



Indices en MongoDB

- Todas las colecciones poseen por defecto un indice sobre el campo _id, que asegura su unicidad, y que no podemos eliminar.
- Tipos de indice:
  - De un solo campo (puede ser un campo de un subdocumento)
  - Compuestos: Indice sobre varios campos (max: 31), cada uno con su propio criterio de ordenacion (1: Ascendente; -1: Descendente)
  - Multiclave: Se aplican a los campos de tipo array
  - Geoespaciales



Indices en MongoDB

Vamos a comparar la diferencia entre usar un indice o no usarlo

- db.personas.find({"Nombre":{$lt:"L"}},{"Nombre":1}).**explain("executionStats")**

Indices en MongoDB

Crear un indice sobre el campo Nombre con orden ascendente

db.personas.createIndex({"Nombre":1})

db.personas.find({"Nombre":{$lt:"L"}},{"Nombre":1}).**explain("executionStats")**

Indices en MongoDB Geoespaciales

En MongoDB se pueden utilizar 2 formatos para los datos de tipo geoespacial:

- **Objeto GeoJSON** (opcion recomendada)
  - Es un documento compuesto por 2 campos:

    type: Indica el tipo de geometria (punto, linea, poligono…)

    coordinates: Es un array de arrays compuestos por 2 valores [longitud,latitud]

    ```
    Ubicacion:{"type": "LineString", "coordinates":[[0,0],[90,45]]}
    ```
  - Podemos usar geojson.io para visualizarlo
  - Los calculos de distancia se realizan sobre el geoide WGS84
- **Pares de coordenadas** (legacy; compatibilidad formatos antiguos)
- Ubicación:[longitud,latitud]
- Los cálculos de distancia se realizan sobre un plano euclideo

Indices en MongoDB Geoespaciales

Operadores geoespaciales

- **$geoIntersects:** Selecciona las geometrias que tienen intersección con la indicada.
- **$geoWithin:** Selecciona las geometrias que estan dentro de la especificada de tipo Polygon o MultiPolygon (o $centerSphere)
- **$near:** Devuelve los documentos ordenados desde el mas proximo al mas alejado del punto indicado. Requiere un index de tipo geoespacial (2d o 2dsphere, en funcion de que el punto se especifique en formato legacy o GeoJSON, respectivamente). **Si el indice es 2D realiza los calculos en proyeccion plana**.
- **$nearSphere:** Devuelve los documentos ordenados desde el mas proximo al mas alejado del punto indicado, **utilizando siempre geometria esferica** (incluso si el indice es 2d). Requiere un index de tipo geoespacial (2d o 2dsphere).

Indices en MongoDB Geoespaciales

Tipos de geometrias del objeto GeoJSON:

- **Point**
- **LineString**
- **Polygon**. Son poligonos cerrados; el ultimo par de coordenadas debe coincidir con el primero

  {type: "Polygon", coordinates: [[[0 , 0] , [3 , 6] , [6 , 1] , [0 , 0]]]}
- **MultiPoint**
- **MultiLineString**
- **MultiPolygon**
- **GeometryCollection**

- **Se crean con el operador $geometry**
  - **$geometry: {type: "Point" ,coordinates: [<longitude> , <latitude>]}**

Indices en MongoDB Geoespaciales

Ademas de $geometry, podemos usar los siguientes especificadores de geometria:

- **$centerSphere** (funciona con indices **2d** y **2dsphere**)

  { $centerSphere: [[<x>, <y>], <radius>] }

  El radio debe expresarse en radianes

- **$maxDistance**: Permite especificar una distancia maxima para los resultados de $near y $nearSphere (funciona con indices **2d** y **2dsphere**)

  { $near: [-74 , 40], $maxDistance: 10 }

  La distancia se expresa en metros

- **$minDistance**: Permite especificar una distancia mínima para los resultados de $near y $nearSphere (solo funciona con indices **2dsphere**)

{ $near:{$geometry: { type: "Point", coordinates: [-73.9667, 40.78] }, $minDistance: 1000, $maxDistance: 5000}}

  La distancia se expresa en metros

Indices en MongoDB Geoespaciales

Indices geoespaciales:

- **2dsphere:** Calcula las distancias en geometria esferica
- **2d**: Calcula las distancias en geometria plana

Intentar ordenar los documentos por la distancia a Toledo de su ubicación:

```
db.personas.find({Ubicacion:{$near:{$geometry:{type:"Point",coordinates:[-4.02,39.87]}}}})
```

No es posible porque no tenemos un indice geoespacial sobre Ubicacion

- Crear un indice de tipo 2dsphere sobre el campo personas.Ubicacion

```
db.personas.createIndex({Ubicacion: "2dsphere"})
```

Comprobar que ahora si se puede ejecutar la consulta

```
db.personas.find({Ubicacion:{$near:{$geometry:{type:"Point",coordinates:[-4.02,39.87]}}}})
```

Indices en MongoDB
Texto

- Sirven para realizar búsquedas "inteligentes" sobre textos, al estilo del buscador de Google.
- Estos indices se construyen mediante:
  - **Token**: En primer lugar el texto se descompone en tokens, que son aproximadamente equivalentes a las palabras
  - **Stemming**: Son algoritmos que reducen los tokens a la raíz de la que provienen. Por ejemplo, la raiz de "cocinar" seria "cocin"
  - **Stop words**: No se indexan ciertas palabras que por si solas no aportan "significado" al texto, como artículos, adverbios, ciertos verbos.

    `https://github.com/mongodb/mongo/blob/master/src/mongo/db/fts/stop_words_spanish.txt`
- Son extremadamente complejos y pueden crecer "brutalmente" por lo que **solo esta permitido tener un indice de texto en cada colección** (aunque puede ser un indice compuesto; es decir, que indexe varios campos)

Indices en MongoDB
Texto

MongoDB permite elegir un idioma predeterminado para el indice de texto.

https://docs.mongodb.com/manual/reference/text-searchlanguages/#text-search-languages

```
db.bibliotecas.createIndex({description: "text"},
                                        {default_language: "spanish"})
```

Indices en MongoDB
Texto

- Para realizar una busqueda hay que combinar los operadores **$text** y **$search**

```
db.bibliotecas.find({$text:{$search:"cocina"}},
                                {description:1,score:{$meta: "textScore"}})
```

- Cada coincidencia recibe una puntuación de similitud, que podemos ver mediante el operador **$meta** configurado con el valor **textScore** (que usa un orden **descendente** en combinación con sort)

```
db.bibliotecas.find({$text:{$search:"cocina"}},
                                {description:1,puntuacion:{$meta: "textScore"}})
```

Indices en MongoDB
Texto

Por defecto, las busquedas realizadas con $text no tienen sensibilidad de mayúsculas/minúsculas ni diacritica, pero podemos activarla asignando el valor true a las opciones **$caseSensitive** y **$diacriticSensitive**:

```
db.bibliotecas.find({$text:{$search:"Cuentos",$caseSensitive:true}},
                        {description:1,puntuacion:{$meta: "textScore"}}).count()
db.bibliotecas.find({$text:{$search:"Cuentos"}},
                        {description:1,puntuacion:{$meta: "textScore"}}).count()
db.bibliotecas.find({$text:{$search:"poesia",$diacriticSensitive:true}},
                                {description:1,puntuacion:{$meta: "textScore"}})
db.bibliotecas.find({$text:{$search:"poesía",$diacriticSensitive: true}},
                                    {description:1,puntuacion:{$meta:"textScore"}})
```

Indices en MongoDB
Texto

Cuando indicamos varios términos en $search, se consideran como un OR.

- Para realizar búsquedas de frases exactas tenemos que escapar las comillas con el caracter \"

```
db.bibliotecas.find({$text:{$search:"\"tradicion oral espanola\""}},
                                    {description:1,puntuacion:{$meta: "textScore"}})
db.bibliotecas.find({$text:{$search:"tradicion oral espanola"}},
                              {description:1,puntuacion:{$meta: "textScore"}}).count()
```

- Se puede incluir un - delante de un termino para excluirlo de la búsqueda

```
db.bibliotecas.find({$text:{$search:"rimas -colmos"}},
                                    {description:1,puntuacion:{$meta: "textScore"}})
 db.bibliotecas.find({$text:{$search:"rimas"}},
                              {description:1,puntuacion:{$meta: "textScore"}}).count()
```

Indices en MongoDB
Texto

Indices de textos compuestos

- En estos indices podemos ponderar el peso de cada campo mediante la opcion weight

```
db.bibliotecas.createIndex({title: "text", description: "text"})
```

– Recuerda que solo podemos tener un indice de texto en cada colección

```
db.bibliotecas.dropIndexes()
db.bibliotecas.find({$text:{$search:"historia"}},
                    {title:1,description:1,puntuacion:{$meta: "textScore"}}).
                                        sort( { puntuacion: { $meta: "textScore" } } ).limit(1)
 "puntuacion" : 1.0865800865800865
db.bibliotecas.createIndex({title: "text", description: "text"},{weights: {title:2,description:1}})
db.bibliotecas.find({$text:{$search:"historia"}},
                    {title:1,description:1,puntuacion:{$meta: "textScore"}}).
                                          sort( { puntuacion:{ $meta: "textScore" } } ).limit(1)
 puntuacion" : 1.658008658008658
```