

MONGODB - ADVANCED INDEXING

http://www.tutorialspoint.com/mongodb/mongodb_advanced_indexing.htm

Copyright © tutorialspoint.com

Consider the following document of **users** collection:

```
{
  "address": {
    "city": "Los Angeles",
    "state": "California",
    "pincode": "123"
  },
  "tags": [
    "music",
    "cricket",
    "blogs"
  ],
  "name": "Tom Benzamin"
}
```

The above document contains an **address sub-document** and a **tags array**.

Indexing Array Fields:

Suppose that we want to search user documents based on his tags. For this, we will create an index on tags array in the collection.

Creating an index on array in turn creates separate index entries for each of its fields. So in our case when we create an index on tags array, separate indexes will be created for its values music, cricket and blogs.

To create an index on tags array, use the following code:

```
>db.users.ensureIndex({"tags":1})
```

After creating the index, we can search on the tags field of the collection like this:

```
>db.users.find({tags:"cricket"})
```

To verify that proper indexing is used, use the following **explain** command:

```
>db.users.find({tags:"cricket"}).explain()
```

The above explain command resulted in "cursor" : "BtreeCursor tags_1" which confirms that proper indexing is used.

Indexing Sub-Document Fields:

Suppose that we want to search documents based on city, state and pincode fields. Since all these fields are part of address sub-document field, we will create index on all the fields of the sub-document.

For creating index on all the three fields of the sub-document, use the following code:

```
>db.users.ensureIndex({"address.city":1,"address.state":1,"address.pincode":1})
```

Once the index is created, we can search for any of the sub-document fields utilizing this index as follows:

```
>db.users.find({"address.city":"Los Angeles"})
```

Remember that the query expression has to follow the order of the index specified. So the index

created above would support the following queries:

```
>db.users.find({"address.city":"Los Angeles","address.state":"California"})
```

It will also support the following query:

```
>db.users.find({"address.city":"LosAngeles","address.state":"California","address.pincode":  
:"123"})
```