MONGODB - COVERED QUERIES

What is a Covered Query

As per the official MongoDB documentation, a covered query is a query in which:

- all the fields in the query are part of an index and
- all the fields returned in the query are in the same index

Since all the fields present in the query are part of an index, MongoDB matches the query conditions and returns the result using the same index without actually looking inside documents. Since indexes are present in RAM, fetching data from indexes is much faster as compared to fetching data by scanning documents.

Using Covered Queries

To test covered queries, consider the following document in **users** collection:

```
{
   "_id": ObjectId("53402597d852426020000002"),
   "contact": "987654321",
   "dob": "01-01-1991",
   "gender": "M",
   "name": "Tom Benzamin",
   "user_name": "tombenzamin"
}
```

We will first create a compound index for **users** collection on fields **gender** and **user_name** using following query:

```
>db.users.ensureIndex({gender:1,user_name:1})
```

Now, this index will cover the following query:

```
>db.users.find({gender:"M"},{user_name:1,_id:0})
```

That is to say that for the above query, MongoDB would not go looking into database documents. Instead it would fetch the required data from indexed data which is very fast.

Since our index does not include **_id** field, we have explicitly excluded it from result set of our query as MongoDB by default returns _id field in every query. So the following query would not have been covered inside the index created above:

```
>db.users.find({gender:"M"},{user_name:1})
```

Lastly, remember that an index cannot cover a query if:

- any of the indexed fields is an array
- any of the indexed fields is a subdocument