NoSQL and MongoDB

An Introduction (Part-III)

CREATE: insertMany() - ordered

Ordered clause is used to tell mongodb that whether insertMany() command should continue with insertion of other documents in case of an error occurs in insertion of a document.

```
db.mycollection.insertMany(
        [
                {“name”: “Tom”},
                {“name”:”John”}
        ],
        {ordered:false}
  )
```

Reading from a Sub-document

db.<collectionname>.find(
{
"<field1>.<sub-field1.1>.<subfield1.1.2>" : <value>
}
)

db.mycoll.find({“address.city”: “Mumbai”})

Projection: Limiting Fields

- 1 to include, and 0 to exclude

db.<collectionname>.find(
{ <field1>:<value1>, <field2>:<value2> },
{ <projectedfield1> : 1, <projectedfiled2> : 1 }
)

ex -->

db.mycollection.find(
{name: “Omkar”},
{name:1, job:1, _id:0}
)

Sorting

- 1 for Ascending, -1 for Descending

db.<collectionname>.find().sort({<keyvalue>:1})

ex -->

db.mycollection.find().sort({name:1})

db.mycollection.find().sort({sal:-1, name:1})

Indexing

- B-tree indexes are the default type of Indexes in MongoDB
- Increases read performance, slows down writes

Creating an Index

db.<collectionname>.createIndex({<filedname>: 1|-1})
db.mycoll.createIndex({name: 1});

db.mycoll.createIndex({desig: -1, name: 1});

where 1 or -1 represent sorting orders ascending and decending respectively.

Indexing

- List Indexes

db.<collectionName.getIndexes()

db.mycollection.getIndexes()

- Drop Index

db. <collectionName>.dropIndex({<fieldName>})

db.mycollection.dropIndex({“name”})