Roll no: 31126

Assignment B2

BAJAJ

Date: / / Page No.

Title: MongoDB - Aggregation and Indexing

Problem Statement:

Design and develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB

Objectives:

- Understand aggregation & indexing in Mongo DB with suitable example.

SW and HW Requirements:

Windows 10 64bit, 8GB RAM,
MongoDB installed on machine.

Theory:

Aggregation:

Aggregation operation process data records and return computed result. Aggregation operations group values from multiple documents together, and can peform a variety of operations on the grouped data to return single result. In SQL count(*) and with group is equivalent to MongoDB aggregation.

syntax:

db.collection_name.aggregate(AGGREGATE_OPERATION)

eg. db.records.aggregate([{$group: {_id:"$rollno", num_cources: {$sum: 1}}}])

this will return number of cources enrolled by each student.

Date: / / Page No.

Indexing:

Indexes support efficient resolution of queries. Without indexes, MongoDB must scan every document of a collection to select those documents that match the query statement. This scan is highly inefficient and require MongoDB to process a large volume of data.

Indexes are special data structures, that store a small portion of data of the set in easy-to-traverse form.

SYNTAX:

db.collection_name.createIndex({KEY:1})

- key is name of field on which you want to create Index and 1 is ascending order.
-1 for decending order.

queries:

db.records.createIndex({"rollno":1})

db.records.dropIndex({"rollno":1})

db.records.getIndexes()

Conclusion:

Successfully designed & implemented aggregation & indexing techniques in MongoDB.

```
show dbs

use assignment10

db.records.insertMany([
{
rollno: 31126,
name: "Omkar Gaikwad",
course: "ML"
},
{
rollno: 31126,
name: "Omkar Gaikwad",
course: "Data Science"
},
{
rollno: 31126,
name: "Omkar Gaikwad",
course: "Databases"
},
{
rollno: 31127,
name: "Vimal Galani",
course: "Python"
},
{
rollno: 31127,
name: "Vimal Galani",
course: "Java"
}
])
```

db.records.find().pretty()

db.records.aggregate([{$group: {_id: "$rollno", num_courses: {$sum: 1}}}])

db.records.createIndex({rollno: -1})

db.records.dropIndexes({rollno: -1})

db.records.getIndexes()

db.dropDatabase()


> db.records.aggregate([{$group: {_id: "$rollno", num_courses: {$sum: 1}}}])
{ "_id" : 31127, "num_courses" : 2 }
{ "_id" : 31126, "num_courses" : 5 }
> db.records.createIndex({rollno: -1})
"createdCollectionAutomatically" : false,
"numIndexesBefore" : 1,
"numIndexesAfter" : 2,
"ok" : 1
> db.records.getIndexes()
"v" : 2,
"key" : {
"_id" : 1
},
"name" : "_id_"
"v" : 2,
"key" : {
"rollno" : -1
},
"name" : "rollno_-1"
> db.records.dropIndexes({courses: 1})
uncaught exception: Error: error dropping indexes : {
"ok" : 0,
"errmsg" : "can't find index with key: { courses: 1.0 }",
"code" : 27,
"codeName" : "IndexNotFound"