31341

Assignment B3

Title: Implement aggregation and indexing with suitable example using MongoDB

Problem Statement: Implement aggregation and indexing with suitable example using MongoDB

Objective: Understand indexing and aggregation concept in MongoDB

Learning Outcome:
Students will be able to implement indexing and aggregation concept in MongoDB

Hardware and software requirements:
MongoDB, Fedora 20, 8GB RAM, 500GB HDD, Monitor, Keyboard, Mouse

Concept related theory:

Indexing:

Indexes support the efficient resolution of queries. Without indexes, MongoDB must can every document of a collection to collect those documents that match the query statement. This scan is highly inefficient and requires MongoDB to process a large volume of data. Indexes are special data structures, which store a small portion of the data set in an easy to traverse form. The index stores the value of a specific field or set of fields, ordered by the value of the field as specified in index. Indexing can be achieved on any field in a document using ensureIndex( )method.

Example:

```
>db.ensureIndex({ id: 1, ename: -1 })
```

This is an example of indexing on multiple fields, i.e. composite index.

Aggregation:

Aggregation operations process data records and return computed results. Aggregation operations group values from multiple documents together, and can perform a variety of operations on the grouped data to return a single result.

In SQL, count(*) and with group by is an equivalent of MongoDB aggregation. The aggregate( )Method For the aggregation in MongoDB one should use aggregate( )method.

Syntax: db.collection.aggregate([ aggregate_operations> ])

Aggregate operation could be finding sum on particular field/key or taking an average or finding maximum or minimum values associated with particular field from various documents in a single collection.

The various available aggregation expressions are listed below:

$sum: Sums up the defined value from all documents in the collection

$avg: Computes average of defined value from all documents in the collection

$max: Displays the maximum of defined value from all documents in the collection

$min: Displays the minimum of defined value from all documents in the collection.

Conclusion:

Thus, we have understood indexing and aggregation in MongoDB, its applications and how to implement it.