Assignment - B3

Title :- Implementation of aggregation & indexing with suitable example using MongoDB

objective :- To understand aggregation & indexing in MongoDB.

outcome :- Implementation of aggregation & indexing in mongoDB.

S/W & H/W :- MongoDB, 64 bit OS.

Theory :-

Aggregation :-
Aggregation Operation Process data records & return computed results Aggregation Operation group values from multiple documents together & can perform variety of operations on group -ed data to return a single result.

MongoDB provides three ways to perform aggregation :-

2020/11/21 18:13

① The aggregation pipeline :-
→ sort :-
sorts the function or fields
1 - Ascending order.
-1 - Descending order.

Indexes in MongoDB :-
Indexes support the diffi-
-cient execution of queries in MongoDB
Indexes are special Data structures
to store a small portion of collections
data set in an easy to traverse form

Indexes Function :-

1) Creation : creates an Index.
ex. db collection createIndex ({"name", -1})

2) Display : getting all Indexes.
ex. db collection getIndexes().

3) Deletion : Delete the index.
ex. db collection dropIndex ({"name of Index"})

Types of Indexes :-

1) Style fields :-
MongoDB supports the creation

2020/11/21 18:13

Date

2) map-reduce function
3) single process aggregation.

→ 1] Aggregation Pipeline :-
documents enters multistage pipeline that transfers documents into aggregated result.

ex. db collection, aggregate ([{ $match {condition} }, {$group { id: "file operation"{ $operation: "amount" }}}

we can use various operation along with $group in aggregation some of them are

1) $ sum - returns sum
2) $ avg - returns average
3) $ min - returns minimum
4) $ max - returns maximum.

→ 2] Count :-
returns count of fields satisfying condition.

db collection count ({'condition'})

① $ first :- returns first element
② $ last :- returns last element

2020/11/21 18:13

· DF user - Defined ascending & Descending indexes on a single field of document

ex. db collection, create Index ({"name"; 1})

2) Compound Index :-
MongoDB supports user-defined Indexes on multiple fields.

ex. db collection create Index ({
"name"; 1, "age"; -1 }).

3) Multikey - Index :-
MongoDB uses multi-key Indexes to index content stored in arrays.

ex. db collection create Index ({
"name frame age"})

Conclusion :-
We successfully implemented the aggregation & Indexing in MongoDB.

2020/11/21 18:14