

## ASSIGNMENT - B3

31332

TITLE :-

Implement aggregation and indexing with Mongo DB.

PROBLEM STATEMENT :-

Implement aggregation and indexing with example using Mongo DB.

OBJECTIVE :-

Understanding indexing & aggregation concept in Mongo DB.

S/W & H/W Requirement.

MongoDB.

Operating System (Linux)

THEORY :-

### 1> Indexing

- Indexing supports the efficient resolution of queries.
- Without indexes, MongoDB must scan every document of a collection to select those documents that query the statement.
- This scan is highly inefficient & requires the MongoDB to process a large volume of data.

Indexes use special data structures which stores small portion of data set in an easy way to traverse.

Indexing can be achieved on any field in a document using `createIndex`

SYNTAX :-

```
db.collection.ensureIndex({key:1})
```

Here, key is name of field on which you want to create Index.

## Aggregation

Aggregation operations process data records and then returns computed results.

Aggregation operations group values from multiple documents together & can perform a variety of operation on the group data to return a single result.

SYNTAX :-

```
db.collection.aggregate (Aggregate operation)
```

i) Aggregate operation could be finding sum on a particular field from various documents in a single collection.

\*) Operations

1. `$sum` : sums up the destined value from all documents in the collection.

- 2) \$max : returns maximum of desired values
- 3) \$avg : returns average of a defined values.

### SQL Terms functions

### Mongo DB Aggregation Operators.

a) WHERE	\$ match
b) GROUP BY	\$ group
c) HAVING	\$ match
d) SELECT	\$ project
e) ORDER BY	\$ sort
f) LIMIT	\$ limit
g) SUM ()	\$ sum
h) COUNT ()	\$ sum.

### CONCLUSION

The indexing & aggregation queries were successfully performed.