Assignment No .11

Title of Assignment: MongoDB – Map-reduces operations:
Implement Map reduces operation with suitable example using MongoDB.
Course Objective:
To acquire the skills to use a powerful, flexible, and scalable general-purpose databases to handle Big Data

Course Outcome:

Implement NoSQL queries using MongoDB

Software Required: - Mongodb

Map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results. MongoDB uses **mapReduce** command for map-reduce operations. MapReduce is generally used for processing large data sets. In simple terms, the mapReduce command takes 2 primary inputs, the mapper function and the reducer function .

Working of Mapper and Reducer Function :

MapReduce is a two-step approach to data processing. First you map, and then you reduce. The mapping step transforms the inputted documents and emits a key=>value pair (the key and/or value can be complex). Then, key/value pairs are grouped by key, such that values for the same key end up in an array. The reduce gets a key and the array of values emitted for that key, and produces the final result. The map and reduce functions are written in JavaScript. A Mapper will start off by reading a collection of data and building a Map with only the required fields we wish to process and group them into one array based on the key. And then this key value pair is fed into a Reducer, which will process the values.

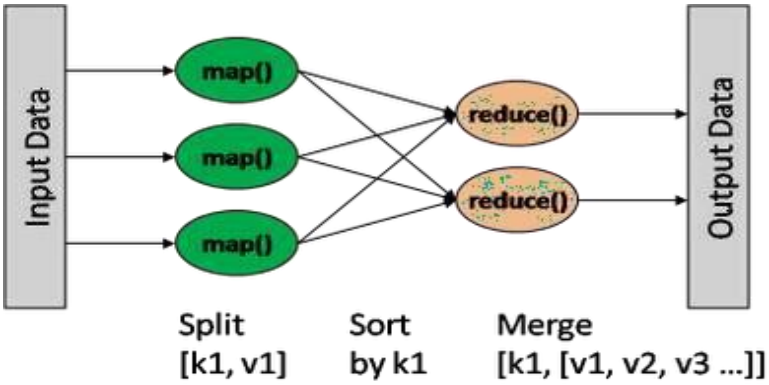
MapReduce Command:

syntax of the basic mapReduce command:

db.collection.mapReduce(function() {emit(key,value);}, //map function

function(key,values) {return reduceFunction}, //reduce function

{out: collection, query: document, sort: document, limit: number})



The map-reduce function first queries the collection, then maps the result documents to emit key-value pairs which is then reduced based on the keys that have multiple values. MapReduce Command:

syntax of the basic mapReduce command:

db.collection.mapReduce(function() {emit(key,value);}, //map function function(key,values) {return reduceFunction}, //reduce function

{out: collection, query: document, sort: document, limit: number}) The map-reduce function first queries the collection, then maps the result documents to emit key-value pairs which is then reduced based on the keys that have multiple values.

In the above syntax:

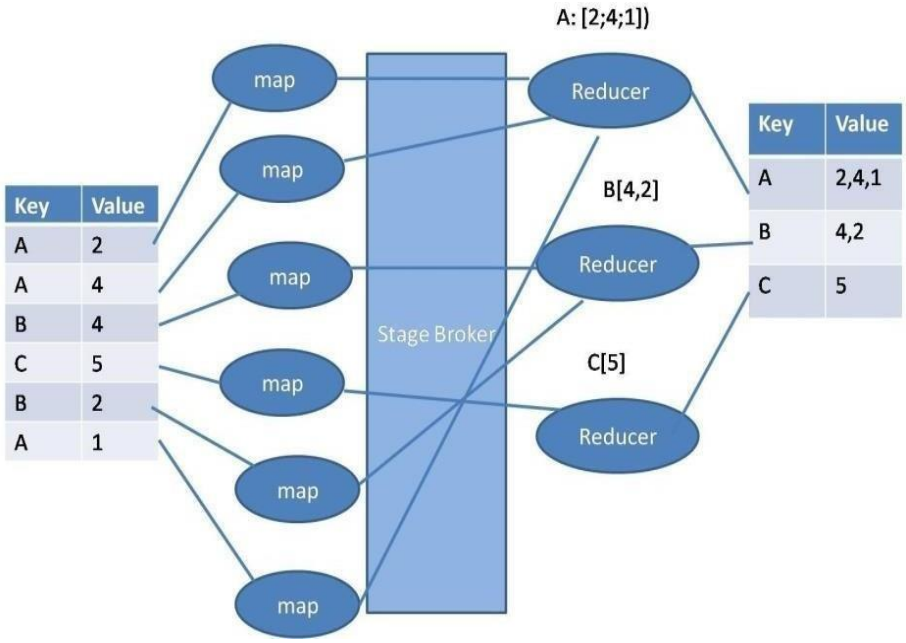
- **map** is a javascript function that maps a value with a key and emits a key-value pair
- **reduce** is a javascript function that reduces or groups all the documents having the same key
- **out** specifies the location of the map-reduce query result
- **query** specifies the optional selection criteria for selecting documents
- **sort** specifies the optional sort criteria
- **limit** specifies the optional maximum number of documents to be returned

Map Reduce Example

The below example is to retrieve the sum of total values related to particular key.

1. Insert data in *mapCollection.*

```
db.mapc.insert({key:"a", value:2})
db.mapc.insert({key:"a", value:4})
```



Conclusion: We have implemented Map reduce using Mongodb Successfully