Group B

Assignment No 12

Title of the Assignment: MongoDB – Map-reduces operations:

Implement Map reduces operation with suitable example using MongoDB.

Objective of the Assignment: To understand the concept of Map reduces operation.

Outcome: Students will be able to learn and understand concept Map reduces operation with examples.

Theory:

❖ **MongoDB - Map Reduce**

Map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results. Toperform map-reduce operations, MongoDB provides the mapReduce database command.
As per the MongoDB documentation, 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. MapReduceis generally used for processing large data sets.

- **MapReduce Command**

Following is the 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

Program:

1. create database

```
test> use map
switched to db map
```

2. create collection

```
map> db.createCollection("books")
{ ok: 1 }
```

3. insert documents

```
map> db.books.insertOne({'name':'JAVA','pages':'100'})
{
acknowledged: true,
insertedId: ObjectId("635d28cbeb14641fd96c5c23")
}
map> db.books.insertOne({'name':'PYTHON','pages':'200'})
{
acknowledged: true,
insertedId: ObjectId("635d28dfeb14641fd96c5c24")
}
map> db.books.insertOne({'name':'XML','pages':'300'})
{
acknowledged: true,
insertedId: ObjectId("635d28eeeb14641fd96c5c25")
}
map> db.books.insertOne({'name':'C++','pages':'350'})
{
acknowledged: true,
insertedId: ObjectId("635d28ffeb14641fd96c5c26")
}
```

```
map> db.books.insertOne({'name':'JAVASCRIPT','pages':'250'})
{
acknowledged: true,
insertedId: ObjectId("635d2917eb14641fd96c5c27")
}
```

4. display documents

```
map>
db.books.find().pr
etty()[
{
_id: ObjectId("635d28cbeb14641fd96c5c23"),
name: 'JAVA',
pages: '100'
},
{
_id: ObjectId("635d28dfeb14641fd96c5c24"),
name: 'PYTHON',
pages: '200'
},
{
_id:
ObjectId("635d28eeeb14641fd96c5c25"),
name: 'XML',
pages: '300'
},
{
_id:
ObjectId("635d28ffeb14641fd96c5c26"),
name: 'C++',
pages: '350'
},
{
_id:
ObjectId("635d2917eb14641fd96c5c27"),
name: 'JAVASCRIPT',
pages: '250'
}
]
```

5.write map function on books

```
map> var map = function()
{
var category;
if(this.pages>=250)
category='Big Books';
else category='Small Books';
emit(category,{name:this.name});
};
```

6. write reduce function on books

```
map> var reduce = function(key, values)
{
var sum =0;
values.forEach(function(doc)
{
sum = sum+1;
});
return{books:sum};};
```

7. write mapreduce function on books

```
map> var count = db.books.mapReduce(map,reduce, {out:"book_results"});
```

8. display the combined result

```
map> db[count.result].find()
[
{ _id: 'Big Books', value: { books: 3 } },
{ _id: 'Small Books', value: { books: 2 } }
]
```

Conclusion: Performed implementation of Map reduces operation.

Viva Question:

- What is MapReduce in MongoDB?
- What is indexing in MongoDB?
- Write reduce function on books?
- Write mapreduce function on books?
- how display the combined result?

Date:	
Marks obtained:	
Sign of course coordinator:	
Name of course Coordinator:	