33231

CLASSMATE
Date :
Page :

# ASSIGNMENT 15

**Aim :** Implement Map reduces operation with suitable example on above MongoDB database

**Problem Statement :** Implement Map reduces operation with suitable example on above MongoDB database

**Objective :**
- To understand concept of Map Reduce in MongoDB
- To implement the concept of document oriented database.

**Theory :**
- Implements the MapReduce model for processing large data sets
- can choose from one of several output options (inline, new collection, merge, replace, reduce)
- MapReduce functions are written in JavaScript
- Supports non sharded and sharded input collections
- Can be used for incremental aggregation over large collections
- MongoDB 2.2 implements much better support for sharded map reduce output
- MapReduce involves two steps :
  - first, map the data from collection specified
  - second, reduce the results

**Map Function**
- var mapFunction1 = function ()
- { emit (this.cust_id, this.amount); };

**Reduce Function**
- var reduceFunction1 = function (key, values)
- return Array.sum(values); };

33231

db.order.mapReduce
(mapFunction1, reduceFunction1, {query : {status : "A"},
out : "order_totals"});

db.Order.mapReduce (
Map function → function () {emit(this.cust_id, this.amount);},
Reduce function → function (key, values)
{return Array.sum (values)},
Query a { query : { status : "A" },
Output collection a out : "order_totals" })
{
"result" : "order_totals",
"timeMillis" : 27,
"counts" : {
"input" : 3,
"emit" : 3,
"reduce" : 1,
"output" : 2,
}, "ok" : 1, }

To display result of MapReduce function use collection created in OUT.
db.<collection_name>.find();
db.order_totals.find();

Conclusion :
Understood to use MapReduce operation in MongoDB.