**DBMS Lab**

**06/10/2020 Assignment No. 12 (Map-Reduce on Bank Collection)**

**Sanjay Rawat(7341)**

**Aim:** To study and implement MapReduce operation in MongoDB

**Title:** Write and execute simple queries to demonstrate MapReduce operation.

**Theory:**

1. **Create collection bank**

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

orders 0.000GB

> use bank

switched to db bank

2. **Insert following documents into it.**

> db.bank.insert({custid:"c01",custname:"Sanjay",Bank:"SBI",Balance:3000,Location:"Pune"})

WriteResult({ "nInserted" : 1 })

> db.bank.insert({custid:"c02",custname:"Anish",Bank:"AXIS",Balance:4000,Location:"Pune"})

WriteResult({ "nInserted" : 1 })

> db.bank.insert({custid:"c03",custname:"Sanjay",Bank:"AXIS",Balance:5000,Location:"Mumbai"})

WriteResult({ "nInserted" : 1 })

> db.bank.insert({custid:"c04",custname:"Sanjay",Bank:"AXIS",Balance:5000,Location:"Nagpur"})

WriteResult({ "nInserted" : 1 })

> db.bank.insert({custid:"c05",custname:"Atul",Bank:"AXIS",Balance:6000,Location:"AGM"})

WriteResult({ "nInserted" : 1 })

> db.bank.insert({custid:"c06",custname:"Golu",Bank:"HDFC",Balance:7000,Location:"RPG"})

WriteResult({ "nInserted" : 1 })

> db.bank.insert({custid:"c07",custname:"Nitin",Bank:"HDFC",Balance:8000,Location:"Dangi"})

WriteResult({ "nInserted" : 1 })

> show dbs

admin 0.000GB

bank 0.000GB

config 0.000GB

local 0.000GB

orders 0.000GB

> db.bank.find().pretty()

{

"\_id" : ObjectId("5f7bf5528edc47bc7525cc27"),

"custid" : "c01",

"custname" : "Sanjay",

"Bank" : "SBI",

"Balance" : 3000,

"Location" : "Pune"

}

{

"\_id" : ObjectId("5f7bf5818edc47bc7525cc28"),

"custid" : "c02",

"custname" : "Anish",

"Bank" : "AXIS",

"Balance" : 4000,

"Location" : "Pune"

}

{

"\_id" : ObjectId("5f7bf5b68edc47bc7525cc29"),

"custid" : "c03",

"custname" : "Sanjay",

"Bank" : "AXIS",

"Balance" : 5000,

"Location" : "Mumbai"

}

{

"\_id" : ObjectId("5f7bf5da8edc47bc7525cc2a"),

"custid" : "c04",

"custname" : "Sanjay",

"Bank" : "AXIS",

"Balance" : 5000,

"Location" : "Nagpur"

}

{

"\_id" : ObjectId("5f7bf6038edc47bc7525cc2b"),

"custid" : "c05",

"custname" : "Atul",

"Bank" : "AXIS",

"Balance" : 6000,

"Location" : "AGM"

}

{

"\_id" : ObjectId("5f7bf63a8edc47bc7525cc2c"),

"custid" : "c06",

"custname" : "Golu",

"Bank" : "HDFC",

"Balance" : 7000,

"Location" : "RPG"

}

{

"\_id" : ObjectId("5f7bf6508edc47bc7525cc2d"),

"custid" : "c07",

"custname" : "Nitin",

"Bank" : "HDFC",

"Balance" : 8000,

"Location" : "Dangi"

}

3**. Perform aggregation using Map-Reduce to display bank wise balance** . **Display the result of new collection bankMR**

> var mapfun1=function(){emit(this.Bank,this.Balance);};

> var reducefun1=function(key,values){ return Array.sum(values); };

> db.bank.mapReduce( mapfun1,reducefun1, {out:"mapreduceq3"} )

{ "result" : "mapreduceq3", "ok" : 1 }

> db.mapreduceq3.find().pretty()

{ "\_id" : "AXIS", "value" : 20000 }

{ "\_id" : "SBI", "value" : 3000 }

{ "\_id" : "HDFC", "value" : 15000 }

5. **Display the balance of HDFC bank only from bankMR collection**

> db.mapreduceq3.find({\_id:"HDFC"}).pretty()

{ "\_id" : "HDFC", "value" : 15000 }

6. **Display the balance of SBI bank only from bankMR collection**

> db.mapreduceq3.find({\_id:"SBI"}).pretty()

{ "\_id" : "SBI", "value" : 3000 }

7. **Display the balance of AXIS bank only from bankMR collection**

> db.mapreduceq3.find({\_id:"AXIS"}).pretty()

{ "\_id" : "AXIS", "value" : 20000 }