/\*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem Statement:- Group-B

Implement Map reduces operation with suitable example using MongoDB.

**Student Name** :- **Roll\_No** :-

**BATCH** :- TEB-3 **Date** :-\_\_/\_\_/\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

gescoe@gescoe-desktop:~$ mongo

MongoDB shell version: 2.4.9

connecting to: test

> show dbs

local 0.03125GB

test (empty)

> use TEB61

switched to db TEB61

> db

TEB61

> show dbs

local 0.03125GB

test (empty)

//Creation of collection

> db.createCollection("sample")

{ "ok" : 1

> show collections

sample

system.indexes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Inserting data collections\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

>db.stud.insert({"name":"alpesh vasani","rollno":65,"cast":”open”, "address" : "nasik"})

WriteResult({ "nInserted" : 1 })

>db.stud.insert({"name":"salman","rollno":55,"cast":"OPan","address":

"nasik"})

WriteResult({ "nInserted" : 1 })

>db.stud.insert({"name":"rathod","rollno":49,"cast":"obc","address":

"jalaon"})

WriteResult({ "nInserted" : 1 })

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Displaying the data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> db.stud.find().pretty()

{

"\_id" : ObjectId("59a5257e7e602f6087591051"),

"name" : "alpesh vasni",

"rollno" : 65,

"cast" : "OPen",

"address" : "nasik"

}

{

"\_id" : ObjectId("59a526697e602f6087591052"),

"name" : "salman",

"rollno" : 55,

"cast" : "OPan",

"address" : "nasik"

}

{

"\_id" : ObjectId("59a5268f7e602f6087591053"),

"name" : "rathod",

"rollno" : 49,

"cast" : "obc",

"address" : "jalaon"

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*using Map reduce function\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> db.stud.mapReduce(

... function(){ emit( this.cast, this.rollno ); },

... function(key, values) { return Array.sum( values ) },

... {

... query: { address: "nasik" },

... out: "order\_totals"

... }

... )

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*output :\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

a]

{

"result" : "order\_totals",

"timeMillis" : 420,

"counts" : {

"input" : 2,

"emit" : 2,

"reduce" : 0,

"output" : 2

},

"ok" : 1

}

b]

> db.order\_totals.find().pretty()

{ "\_id" : "OPan", "value" : 55 }

{ "\_id" : "OPen", "value" : 65 }