**DBMS Lab**

**7358 Umang Kumar**

**Assignment No. -3 (Group B)**

Title: Implement Map reduces operation with suitable example using MongoDB.

PS C:\Users\hp> mongo

MongoDB shell version v4.4.17

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("e2f94a9d-5581-407f-8eaa-2e8e34e7fa38") }

MongoDB server version: 4.4.17

---

The server generated these startup warnings when booting:

2022-10-27T17:42:48.501+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

---

---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display

metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you

and anyone you share the URL with. MongoDB may use this information to make product

improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()

To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

Umang 0.000GB

> use Umang

switched to db Umang

> db.createCollection('Journal');

{ "ok" : 1 }

> db.journal.insert({'book\_id':1,'book\_name':'OOP',amt':500,'status':'Available'});

uncaught exception: SyntaxError: missing : after property id :

@(shell):1:52

>db.Journal.insert({'book\_id':1,'book\_name':'JavacdOOP','amt':500,'status':'A vailable'});

WriteResult({"nInserted":1})

> db.Journal.insert({'book\_id':1,'book\_name':'JavaOOP','amt':400,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book\_id':1,'book\_name':'Java','amt':300,'s tatus':'Not Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book\_id':2,'book\_name':'Java','amt':300,'s tatus':'Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book\_id':2,'book\_name':'OPP','amt':200,'st atus':'Available'});

WriteResult({ "nInserted" : 1 }) >db.Journal.insert({'book\_id':2,'book\_name':'C+','amt':200,'status':'Available'} );

WriteResult({ "nInserted" : 1 }) >db.Journal.insert({'book\_id':3,'book\_name':'C+','amt':150,'status':'Available'} );

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':3,'book\_name':'C+ +','amt':200,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':4,'book\_name':'OPP C+ +','amt':300,'status':'Not Available'}); WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':5,'book\_name':'OPP C+ +','amt':400,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':5,'book\_name':'C+ +','amt':400,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':5,'book\_name':'C++ Java','amt':400,'status':'Not Available'}); WriteResult({ "nInserted" : 1 })

> var mapfunction=function(){ emit(this.book\_id,this.amt)};

> var reducefunction=function(key,value){return Array.sum(value);};

> db.Journal.mapReduce(mapfunction,reducefunction, {'out':'new'});

{ "result" : "new", "timeMillis" : 49,"counts" : { "input" : 12, "emit" : 12, "reduce" : 4, "output" : 5 }, "ok" : 1 }

> db.Journal.mapReduce(mapfunction,reducefunction, {'out':'new'}).find().pretty();

{ "\_id" : 1, "value" : 1200 }

{ "\_id" : 2, "value" : 700 }

{ "\_id" : 3, "value" : 350 }

{ "\_id" : 4, "value" : 300 }

{ "\_id" : 5, "value" : 1200 }

>

>

> db.new.find().pretty();

{ "\_id" : 1, "value" : 1200 }

{ "\_id" : 2, "value" : 700 }

{ "\_id" : 3, "value" : 350 }

{ "\_id" : 4, "value" : 300 }

{ "\_id" : 5, "value" : 1200 }