**Assignment No. 02**

**Roll No: 3006**

**Title:** Implement Map-reduce and aggregation, indexing with suitable example in MongoDB.

Demonstrate the following:

• Aggregation framework

• Create and drop different types of indexes and explain () to show the advantage of the indexes.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1) Switch database**

> use assign1

switched to db assign1

**2) Create Collection**

> db.createCollection("Book");

{ "ok" : 1 }

**3) Insert documents into collection**

> db.Book.insert({id:"3586824624",title:"Database Management System",author:"S.Sudharshan",dept:"IT",subject:"ADBMS",price:756.76})

WriteResult({ "nInserted" : 1 })

> db.Book.insert({id:"8753698428",title:"Designing The User Interface",author:"Alan Dix",dept:"IT",subject:"HCI",price:876.75})

WriteResult({ "nInserted" : 1 })

> db.Book.insert([{id:"2376574365",title:"Machine Design",author:"Saul Fenster",dept:"ME",subject:"Machine Design",price:975},{id:"4868362945",title:"Digital Electronics",dept:"EE",price:832.765},{id:298374574,title:"Engineering Thermodyanamics",author:"P.K.Nag",dept:"ME",price:732.98}])

**4) Retrieve and Display Documents**

> db.Book.find().pretty()

{

"\_id" : ObjectId("64ece032848ae0682d696f52"),

"id" : "8753698428",

"title" : "Designing The User Interface",

"author" : "Alan Dix",

"dept" : "IT",

"subject" : "HCI",

"price" : 876.75

}

{

"\_id" : ObjectId("64eceb95caf728f8ddcefd13"),

"id" : "2376574365",

"title" : "Machine Design",

"author" : "Saul Fenster",

"dept" : "ME",

"subject" : "Machine Design",

"price" : 975

}

{

"\_id" : ObjectId("64eceb95caf728f8ddcefd14"),

"id" : "4868362945",

"title" : "Digital Electronics",

"dept" : "EE",

"price" : 832.765

}

{

"\_id" : ObjectId("64eceb95caf728f8ddcefd15"),

"id" : 298374574,

"title" : "Engineering Thermodyanamics",

"author" : "P.K.Nag",

"dept" : "ME",

"price" : 732.98

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Map Reduce Function\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

> var mf=function(){emit(this.dept,this.price)}

> var rf=function(Department,Total){return Array.sum(Total)}

> db.Book.mapReduce(mf,rf,{query:{dept:"IT"},out:"Total"})

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

> db.Total.find()

{ "\_id" : "IT", "value" : 4165.76 }

var mf=function(){emit(this.dept,this.price)}

> var rf=function(Department,Total){return Array.avg(Total)}

> db.Book.mapReduce(mf,rf,{query:{dept:"IT"},out:"Average"})

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

> db.Average.find()

{ "\_id" : "IT", "value" : 833.152 }

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Aggregate Operation (SUM)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

> db.Book.aggregate([{$match:{"dept":"IT"}},{$group:{\_id:{"department":"$dept"},total:{$sum:"$price"}}}])

{ "\_id" : { "department" : "IT" }, "total" : 750 }

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Aggregate Operation (AVG)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

> db.Book.aggregate([{$match:{"author":"Alan Dix"}},{$group:{\_id:{"department":"$dept"},total:{$avg:"$price"}}}])

{ "\_id" : { "department" : "IT" }, "total" : 750 }

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Aggregate Operation (MAX/MIN)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

> db.Book.aggregate([{$match:{"dept":"IT"}},{$group:{\_id:{"department":"$dept"},min:{$min:"$price"}}}])

{ "\_id" : { "department" : "IT" }, "min" : 701 }

>

> db.Book.aggregate([{$match:{"dept":"IT"}},{$group:{\_id:{"department":"$dept"},max:{$max:"$price"}}}])

{ "\_id" : { "department" : "IT" }, "max" : 986 }

>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Creating Index \*\*\*\*\*\*\*\*\*\***

> db.Book.createIndex({"price":1})

{

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"createdCollectionAutomatically" : false,

"ok" : 1

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Display Index\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

> db.Book.getIndexes()

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_"

},

{

"v" : 2,

"key" : {

"price" : 1

},

"name" : "price\_1"

}

]

>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Drop Index\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

> db.Book.dropIndexes({price:1})

{ "nIndexesWas" : 2, "ok" : 1 }

> db.Book.getIndexes()

[ { "v" : 2, "key" : { "\_id" : 1 }, "name" : "\_id\_" } ]

>