**Group B**

**ASSIGNMENT:-5**

**PROBLEM STATEMENT:**- Implement map reduce operation with suitable example using monoDB.

> db.createCollection('order')

{ "ok" : 1 }

> db.order.insert({cust\_id:'A1',Amount:400,Status:'A'})

WriteResult({ "nInserted" : 1 })

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

{

"\_id" : ObjectId("59ba3330616f0c36cc21c068"),

"cust\_id" : "A1",

"Amount" : 400,

"Status" : "A"

}

> db.order.insert({cust\_id:'B1',Amount:300,Status:'A'})

WriteResult({ "nInserted" : 1 })

> db.order.insert({cust\_id:'A1',Amount:200,Status:'A'})

WriteResult({ "nInserted" : 1 })

> db.order.insert({cust\_id:'C1',Amount:200,Status:'B'})

WriteResult({ "nInserted" : 1 })

> db.order.insert({cust\_id:'B1',Amount:700,Status:'B'})

WriteResult({ "nInserted" : 1 })

> db.order.insert({cust\_id:'B1',Amount:800,Status:'C'})

WriteResult({ "nInserted" : 1 })

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

{

"\_id" : ObjectId("59ba3330616f0c36cc21c068"),

"cust\_id" : "A1",

"Amount" : 400,

"Status" : "A"

}

{

"\_id" : ObjectId("59ba3358616f0c36cc21c069"),

"cust\_id" : "B1",

"Amount" : 300,

"Status" : "A"

}

{

"\_id" : ObjectId("59ba3371616f0c36cc21c06a"),

"cust\_id" : "A1",

"Amount" : 200,

"Status" : "A"

}

{

"\_id" : ObjectId("59ba3383616f0c36cc21c06b"),

"cust\_id" : "C1",

"Amount" : 200,

"Status" : "B"

}

{

"\_id" : ObjectId("59ba3393616f0c36cc21c06c"),

"cust\_id" : "B1",

"Amount" : 700,

"Status" : "B"

}

{

"\_id" : ObjectId("59ba33a3616f0c36cc21c06d"),

"cust\_id" : "B1",

"Amount" : 800,

"Status" : "C"

}

> db.order.mapReduce(function(){emit(this.cust\_id,this.Amount)},function(key,values){return Array.sum(values)},{query:{Status:"A"},out:"order\_totals"}).find().pretty()

{ "\_id" : "A1", "value" : 600 }

{ "\_id" : "B1", "value" : 300 }

>

> db.order.mapReduce(function(){emit(this.cust\_id,this.Amount)},function(key,values){return Array.avg(values)},{query:{Status:"A"},out:"order\_avg"}).find().pretty()

{ "\_id" : "A1", "value" : 300 }

{ "\_id" : "B1", "value" : 300 }

> db.order.mapReduce(function(){emit(this.cust\_id,this.Amount)},function(key,values){return Math.min.apply(Math,values)},{query:{Status:"A"},out:"order\_min"}).find().pretty()

{ "\_id" : "A1", "value" : 200 }

{ "\_id" : "B1", "value" : 300 }

> db.order.mapReduce(function(){emit(this.cust\_id,this.Amount)},function(key,values){return Math.max.apply(Math,values)},{query:{Status:"A"},out:"order\_max"}).find().pretty()

{ "\_id" : "A1", "value" : 400 }

{ "\_id" : "B1", "value" : 300 }

>

> db.order.mapReduce(function(){emit(this.cust\_id,this.Amount)},function(key,values){return Math.max.apply(Math,values)},{out:"order\_max"}).find().pretty()

{ "\_id" : "A1", "value" : 400 }

{ "\_id" : "B1", "value" : 800 }

{ "\_id" : "C1", "value" : 200 }

> db.order.mapReduce(function(){emit(this.cust\_id,this.Amount)},function(key,values){return Math.min.apply(Math,values)},{out:"order\_min"}).find().pretty()

{ "\_id" : "A1", "value" : 200 }

{ "\_id" : "B1", "value" : 300 }

{ "\_id" : "C1", "value" : 200 }

>

> db.order.mapReduce(function(){emit(this.cust\_id,this.Amount)},function(key,values){return Array.avg(values)},{out:"order\_avg"}).find().pretty()

{ "\_id" : "A1", "value" : 300 }

{ "\_id" : "B1", "value" : 600 }

{ "\_id" : "C1", "value" : 200 }