Create the following schema

Order(id,amount ,status)

Cus id Amount Status

A1 400 P

B1 300 D

A1 200 F

C1 200 F

B1 700 P

B1 800 P

Status: P=”Pending”, D= “Delivered”, F= “Failed”

Implement the following using Map Reduce function

1. Find the sum of amount of each customer whose status is P

2. Find the average amount of each customer

3. Find the min amount of each customer

4. Find the max amount of each customer whose status is F

============================================================

#CREATION OF COLLECTION : ORDER

>

db.createCollection("Order")

{ "ok" : 1 }

>

db.Order.insert({Customer\_id:'A1',Amount:400,Status:'P'})

WriteResult({ "nInserted" : 1 })

>

db.Order.insert({Customer\_id:'B1',Amount:300,Status:'D'})

WriteResult({ "nInserted" : 1 })

>

db.Order.insert({Customer\_id:'A1',Amount:200,Status:'F'})

WriteResult({ "nInserted" : 1 })

>

db.Order.insert({Customer\_id:'C1',Amount:200,Status:'F'})

WriteResult({ "nInserted" : 1 })

>

db.Order.insert({Customer\_id:'B1',Amount:700,Status:'P'})

WriteResult({ "nInserted" : 1 })

>

db.Order.insert({Customer\_id:'B1',Amount:800,Status:'P'})

WriteResult({ "nInserted" : 1 })

> db.Order.find().pretty()

{

"\_id" : ObjectId("5ba1dbe5691da4e812906374"),

"Customer\_id" : "A1",

"Amount" : 400,

"Status" : "P"

}

{

"\_id" : ObjectId("5ba1dbf8691da4e812906375"),

"Customer\_id" : "B1",

"Amount" : 300,

"Status" : "D"

}

{

"\_id" : ObjectId("5ba1dc06691da4e812906376"),

"Customer\_id" : "A1",

"Amount" : 200,

"Status" : "F"

}

{

"\_id" : ObjectId("5ba1dc0e691da4e812906377"),

"Customer\_id" : "C1",

"Amount" : 200,

"Status" : "F"

}

{

"\_id" : ObjectId("5ba1dc1d691da4e812906378"),

"Customer\_id" : "B1",

"Amount" : 700,

"Status" : "P"

}

{

"\_id" : ObjectId("5ba1dc24691da4e812906379"),

"Customer\_id" : "B1",

"Amount" : 800,

"Status" : "P"

}

1. Find the sum of amount of each customer whose status is P

>

var

mapfunction=function(){if(this.Status=='P')emit(this.Customer\_id,this.Amount)};

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

> db.Order.mapReduce(mapfunction,reducefunction,{'out':'Order\_total'}) {

"result" :

"Order\_total",

"timeMillis" : 558,

"counts" :

{

"input" : 6, "emit" : 3,

"reduce" : 1,

"output" : 2

}

, "ok" : 1 }

> > db.Order\_total.find() { "\_id" : "A1", "value" : 400 }

{ "\_id" : "B1", "value" : 1500 }

2. Find the average amount of each customer

> var

mapfunction=function()

{

if(this.Customer\_id=='A1')emit(this.Customer\_id,this.Amo unt)};

> var

reducefunction=function(key,values){return Array.avg(values);

};

> db.Order.mapReduce(mapfunction,reducefunction,{'out':'Order\_average\_A1'})

{

"result" :

"Order\_average\_A1",

"timeMillis" : 379, "counts" :

{

"input" : 6,

"emit" : 2,

"reduce" : 1,

"output" : 1 },

"ok" : 1 }

> db.Order\_average\_A1.find()

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

..............................................................................................

> var

mapfunction=function()

{

if(this.Customer\_id=='B1')emit(this.Customer\_id,this.Amo unt)};

> var

reducefunction=function(key,values){return Array.avg(values);};

> db.Order.mapReduce(mapfunction,reducefunction,

{'out':'Order\_average\_B1'})

{ "result" :

"Order\_average\_B1",

"timeMillis" : 443,

"counts" : { "input" : 6, "emit" : 3, "reduce" : 1, "output" : 1 },

"ok" : 1

}

> db.Order\_average\_B1.find() { "\_id" : "B1", "value" : 600 }

>

............................................................................................

> var

mapfunction=function()

{

if(this.Customer\_id=='C1')emit(this.Customer\_id,this.Amo unt)};

> var

reducefunction=function(key,values){return Array.avg(values);};

> db.Order.mapReduce(mapfunction,reducefunction,

{

'out':'Order\_average\_C1'})

{

"result" : "Order\_average\_C1",

"timeMillis" : 413,

"counts" :

{

"input" : 6, "emit" : 1, "reduce" : 0, "output" : 1 },

"ok" : 1 }

> db.Order\_average\_C1.find()

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

> var

mapfunction=function(){emit(this.Customer\_id,this.Amount)};

> var

reducefunction=function(key,values){return Array.avg(values);};

> db.Order.mapReduce(mapfunction,reducefunction,

{'out':'Order\_average'})

{

"result" :

"Order\_average",

"timeMillis" : 422,

"counts" :

{

"input" : 6, "emit" : 6, "reduce" : 2, "output" : 3

},

"ok" : 1 }

> db.Order\_average.find() { "\_id" : "A1", "value" : 300 }

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

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

3. Find the min amount of each customer

> var

mapfunction=function()

{

emit(this.Customer\_id,this.Amount)};

> var

reducefunction=function(key,values)

{

return Math.min.apply(Math,values);};

> db.Order.mapReduce(mapfunction,reducefunction,

{'out':'Order\_minimum'})

{

"result" :

"Order\_minimum",

"timeMillis" : 458,

"counts" :

{

"input" : 6, "emit" : 6, "reduce" : 2, "output" : 3 }, "ok" : 1 }

> db.Order\_minimum.find() { "\_id" : "A1", "value" : 200 }

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

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

4. Find the max amount of each customer whose status is F

> var

mapfunction=function()

{

if(this.Status=='F')emit(this.Customer\_id,this.Amount)};

> var

reducefunction=function(key,values)

{

return Math.max.apply(Math,values);};

> db.Order.mapReduce(mapfunction,reducefunction,

{'out':'Order\_Maximum\_F'})

{

"result" :

"Order\_Maximum\_F",

"timeMillis" : 427,

"counts" :

{

"input" : 6, "emit" : 2, "reduce" : 0, "output" : 2 }, "ok" : 1 }

> db.Order\_Maximum\_F.find()

{

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

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

.............................................................................................

> var

mapfunction=function()

{

if(this.Status=='F')emit(this.Customer\_id,this.Amount)};

> var

reducefunction=function(key,values)

{

return Math.max.apply(Math,values);};

> db.Order.mapReduce(mapfunction,reducefunction,{'out':'Order\_Maximum\_F'})

{

"result" : "Order\_Maximum\_F",

"timeMillis" : 419,

"counts" :

{

"input" : 8, "emit" : 4, "reduce" : 1, "output" : 3

},

"ok" : 1

}

> db.Order\_Maximum\_F.find()

{ "\_id" : "A1", "value" : 800 } { "\_id" : "B1", "value" : 800 } { "\_id" : "C1", "value" : 200

} >