1.**In database Employee.**

1. find the average salary of each dept.

db.empDeatils.aggregate([{$group:{\_id:"$dept","avg salary":{$avg:"$salary"}}}])

{

"\_id" : "designer",

"avg salary" : 2300

}

{ "\_id" : "tester",

"avg salary" : 12250

}

{ "\_id" : "developer",

"avg salary" : 6000

}

1. find the minimum salary of each dept.

db.empDeatils.aggregate([{$group:{\_id:"$dept","minimum salary":{$min:"$salary"}}}])

{ "\_id" : "designer",

"minimum salary" : 2000

}

{ "\_id" : "tester",

"minimum salary" : 4500

}

{ "\_id" : "developer",

"minimum salary" : 6000

}

1. find the average salary of each dept.

db.empDeatils.aggregate([{$group:{\_id:"$dept","maximum salary":{$max:"$salary"}}}])

{ "\_id" : "designer",

"maximum salary" : 2600

}

{ "\_id" : "tester",

"maximum salary" : 20000

}

{ "\_id" : "developer",

"maximum salary" : 6000

}

1. find the no.of employees of each dept.

db.empDeatils.aggregate([{$group:{\_id:"$dept","no of emp":{$sum:1}}}])

{ "\_id" : "designer",

"no of emp" : 2

}

{ "\_id" : "tester",

"no of emp" : 2

}

{ "\_id" : "developer",

"no of emp" : 1

}

1. sort the collection empDetails in descending order of name

db.empDeatils.find().sort({"name":-1})

{

"\_id" : ObjectId("629acb7a9d9d94a998cfab38"),

"name" : "sanju",

"age" : 40,

"email" : "ghgh@gmail.com",

"phone" : 646768,

"salary" : 6000,

"dept" : "developer"

}

{

"\_id" : ObjectId("629acb7a9d9d94a998cfab36"),

"name" : "raju",

"age" : 21,

"email" : "dgv@gmail.com",

"phone" : 5768599,

"salary" : 20000,

"dept" : "tester"

}

{

"\_id" : ObjectId("629acb7a9d9d94a998cfab37"),

"name" : "mohan",

"age" : 52,

"email" : "mymail@gmail.com",

"phone" : 235578,

"salary" : 2600,

"dept" : "designer"

}

{

"\_id" : ObjectId("629acb7a9d9d94a998cfab34"),

"name" : "chinju",

"age" : 21,

"email" : "acbc@gmail.com",

"phone" : 2789084,

"salary" : 2000,

"dept" : "designer"

}

{

"\_id" : ObjectId("629f0c51a620e3bbad4328fc"),

"name" : "anu", "age" : 32,

"email" : "bc@gmail.com",

"phone" : 7655,

"salary" : 4500,

"dept" : "tester"

}

1. Create a text index for ‘name’ and search for names mohan and bhuvan

db.empDeatils.createIndex({name:"text"})

db.empDeatils.find({$text:{$search:"mohan anu"}})

{

"\_id" : ObjectId("629f0c51a620e3bbad4328fc"),

"name" : "anu",

"age" : 32,

"email" : "bc@gmail.com",

"phone" : 7655,

"salary" : 4500,

"dept" : "tester"

}

{

"\_id" : ObjectId("629acb7a9d9d94a998cfab37"),

"name" : "mohan",

"age" : 52,

"email" : "mymail@gmail.com",

"phone" : 235578,

"salary" : 2600,

"dept" : "designer"

}

2.create a database Inventory and create an orders collection. Apply MapReduce operation for finding the total purchase of each customer.

db.order.insert({custid:101,name:"anu",item:"sugar",price:80})

db.order.insert({custid:701,name:"manu",item:"wheat",price:180})

db.order.insert({custid:201,name:"maya",item:"sugar",price:80})

db.order.insert({custid:101,name:"anu",item:"biscuit",price:100})

db.order.insert({custid:701,name:"manu",item:"rice",price:280})

db.order.insert({custid:101,name:"anu",item:"sugar",price:80})

db.order.find()

{

"\_id" : ObjectId("629f1706a620e3bbad4328fd"), "custid" : 101,

"name" : "anu",

"item" : "sugar",

"price" : 80

}

{

"\_id" : ObjectId("629f178ca620e3bbad4328fe"), "custid" : 701,

"name" : "manu",

item" : "wheat",

"price" : 180

}

{

"\_id" : ObjectId("629f178ca620e3bbad4328ff"), "custid" : 201,

"name" : "maya",

"item" : "sugar",

"price" : 80

}

{

"\_id" : ObjectId("629f178ca620e3bbad432900"), "custid" : 101,

"name" : "anu",

"item" : "biscuit",

"price" : 100

}

{

"\_id" : ObjectId("629f178ca620e3bbad432901"), "custid" : 101,

"name" : "manu",

"item" : "rice",

"price" : 280

}

{

"\_id" : ObjectId("629f178da620e3bbad432902"), "custid" : 101,

"name" : "anu",

"item" : "sugar",

"price" : 80

}

var mapFunction=function(){emit(this.custid,this.price);};

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

db.order.mapReduce(mapFunction,reduceFunction,{'out':"map\_example"})

{ "\_id" : 101, "value" : 540 }

{ "\_id" : 201, "value" : 80 }

{ "\_id" : 701, "value" : 180 }

var mapFunction=function(){emit(this.custid,this.price);};

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

db.order.mapReduce(mapFunction,reduceFunction,{'query':{custid:{$gt:101}},'out':"map\_example"})

> db.map\_example.find()

{ "\_id" : 701, "value" : 180 }

{ "\_id" : 201, "value" : 80 }

**Example Queries**

>db.emp1.aggregate([{$group:{\_id:"$dept","avg sal":{$avg:"$sal"}}}])

> db.emp1.aggregate([{$group:{\_id:"$dept","minimum sal":{$min:"$sal"}}}])

> db.emp1.aggregate([{$group:{\_id:"$dept","maximum sal":{$max:"$sal"}}}])

> db.emp1.aggregate([{$group:{\_id:"$dept","no of emp":{$sum:1}}}])

> db.bookdata.find().sort({"Author":1})

> db.bookdata.find().sort({"Author":-1})

> db.bookdata.createIndex({Title:"text"})

> db.bookdata.find({$text:{$search:"DBMS COA"}})