ADBMS LAB

CO5- Programs

1.In database

Employee. use

db_name switched to db

db_name

> use emp switched

to db emp

> show collections empDetails

> db.empDetails.find()

{ "_id" : ObjectId("613726aa40cbedb84b494e0d"), "Name" : "Mohan", "Age" : 30, "Email" : "mohan12345@gmail.com", "salary" : 5000 }

{ "_id" : ObjectId("613726d340cbedb84b494e0e"), "Name" : "Raju", "Age" : 35, "Email" : "raju@gmail.com", "salary" : 7000 }

{ "_id" : ObjectId("613726fc40cbedb84b494e0f"), "Name" : "Bhuvan", "Age" : 25, "Email" : "bhuvan@gmail.com", "salary" : 10000 }

{ "_id" : ObjectId("6137272f40cbedb84b494e10"), "Name" : "Meera", "Age" : 27, "Email" : "meera@gmail.com", "salary" : 9000 }

{ "_id" : ObjectId("6137275040cbedb84b494e11"), "Name" : "Maya", "Age" : 28, "Email" : "maya@gmail.com", "salary" : 15000 }

db.empDetails.updateMany({Name :"Mohan"},{Dept : "Designer"}})

{ "acknolegement" :true, "matchedCount :1, "modified Count" : 1 }

db.empDetails.updateMany({Name :”Raju”},{Dept : “Tester”}})

{ “acknolegement” :true, “matchedCount :1, “modified Count” : 1 }

db.empDetails.updateMany({Name :”Bhuvan”},{Dept : “Developer”}})

{ “acknolegement” :true, “matchedCount :1, “modified Count” : 1 }

db.empDetails.updateMany({Name :”Meera”},{Dept : “Designer”}})

{ “acknolegement” :true, “matchedCount :1, “modified Count” : 1 }

db.empDetails.updateMany({Name :”Maya”},{Dept : “Tester”}})

{ “acknolegement” :true, “matchedCount :1, “modified Count” : 1 }

> db.empDetails.find()

{ "_id" : ObjectId("613726aa40cbedb84b494e0d"), "Name" : "Mohan", "Age" : 30, "Email" : "mohan12345@gmail.com", "salary" : 5000 ,”Dept “ : “Designer”}

{ "_id" : ObjectId("613726d340cbedb84b494e0e"), "Name" : "Raju", "Age" : 35, "Email" : "raju@gmail.com", "salary" : 7000,”Dept “ : “Tester” }

{ "_id" : ObjectId("613726fc40cbedb84b494e0f"), "Name" : "Bhuvan", "Age" : 25, "Email" : "bhuvan@gmail.com", "salary" : 10000,”Dept “ : “Developer”}

{ "_id" : ObjectId("6137272f40cbedb84b494e10"), "Name" : "Meera", "Age" : 27, "Email" : "meera@gmail.com", "salary" : 9000 ,”Dept “ : “Designer”}

{ "_id" : ObjectId("6137275040cbedb84b494e11"), "Name" : "Maya", "Age" : 28, "Email" : "maya@gmail.com", "salary" : 15000, ,”Dept “ : “Tester”}

1.find the average salary of each dept.

db.empDetails.aggregate([{$group:{_id:"$Dept","Avg_Salary":{$avg:"$Salary"}}}])

{ "_id" : "Designer", "Salary" : 10000 }

{ "_id" : "Developer", "Salary" : 10000 }

{ "_id" : "Tester", "Salary" : 8000 }

2. find the minimum salary of each dept.

db.empDetails.aggregate([{$group:{_id:"$Dept","Min_Salary":{$min:"$Salary"}}}])

{ "_id" : "Designer", "Salary" : 5000 }

{ "_id" : "Developer", "Salary" : 10000 }

{ "_id" : "Tester", "Salary" : 7000 }

3. find the maximum salary of each dept.

> db.empDetails.aggregate([{$group:{_id:"$Dept","Max_Salary":{$max:"$Salary"}}}])

{ "_id" : "Tester", "Max_Salary" : 9000 }

{ "_id" : "Designer", "Max_Salary" : 15000 }

{ "_id" : "Developer", "Max_Salary" : 10000 }

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

db.empDetails.aggregate([{$group:{id:"Dept","No of employees":{$sum:1}}}])

{"id":"Dept","No of employees" : 5 }

5. sort the collection empDetails in descending order of name

> db.empDetails.find().sort({"Name": -1})

{ "_id" : ObjectId("613726d340cbedb84b494e0e"), "Name" : "Raju", "Age" : 35, "Email" : "raju@gmail.com", "salary" : 7000 }

{ "_id" : ObjectId("613726aa40cbedb84b494e0d"), "Name" : "Mohan", "Age" : 30, "Email" : "mohan12345@gmail.com", "salary" : 5000 }

{ "_id" : ObjectId("6137272f40cbedb84b494e10"), "Name" : "Meera", "Age" : 27, "Email" : "meera@gmail.com", "salary" : 9000 }

{ "_id" : ObjectId("6137275040cbedb84b494e11"), "Name" : "Maya", "Age" : 28, "Email" : "maya@gmail.com", "salary" : 15000 }

{ "_id" : ObjectId("613726fc40cbedb84b494e0f"), "Name" : "Bhuvan", "Age" : 25, "Email" : "bhuvan@gmail.com", "salary" : 10000 }

6. Create a text index for 'name' and search for names mohan and bhuvan

db.empDetails.createIndex({Name : "text"})

{

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"createdCollectionAutomatically" : false,

"ok" : 1

}

> db.empDetails.find()

{ "_id" : ObjectId("6141db0be3945e33a7255b56"), "Name" : "Mohan", "Age" : 30, "Email" : "mohan@gmail.com", "Salary" : 5000 }

{ "_id" : ObjectId("6141db14e3945e33a7255b57"), "Name" : "Raju", "Age" : 35, "Email" : "raju@gmail.com", "Salary" : 7000 }

{ "_id" : ObjectId("6141db20e3945e33a7255b58"), "Name" : "Bhuvan", "Age" : 25, "Email" : "bhuvan@gmail.com", "Salary" : 10000 }

> db.empDetails.find({$text:{$search:"Mohan Bhuvan"}})

{ "_id" : ObjectId("6141db20e3945e33a7255b58"), "Name" : "Bhuvan", "Age" : 25, "Email" : "bhuvan@gmail.com", "Salary" : 10000 }

{ "_id" : ObjectId("6141db0be3945e33a7255b56"), "Name" : "Mohan", "Age" : 30, "Email" : "mohan@gmail.com", "Salary" : 5000 }

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

```
use inventory switched
to db inventory
> db.createCollection("orders")
{ "ok" : 1 }
> db.orders.insert({custid:200,name:"maya",item:"rice",price:340})
WriteResult({ "nInserted" : 1 })

> db.orders.insert({custid:201,name:"anu",item:"rice",price:340})
WriteResult({ "nInserted" : 1 })

> db.orders.insert({custid:202,name:"Meera",item:"sugar",price:35}) WriteResult({
"nInserted" : 1 })

> db.orders.insert({custid:201,name:"anu",item:"wheat",price:40})
WriteResult({ "nInserted" : 1 })

>                db.orders.insert({custid:202,name:"Meera",item:"teapower",price:250})
WriteResult({ "nInserted" : 1 })

> db.orders.insert({custid:203,name:"navya",item:"wheat",price:40})
WriteResult({ "nInserted" : 1 })
> db.orders.find()
{ "_id" : ObjectId("614251b0ec9cd36cd1db1f46"), "custid" : 200, "name" : "maya", "item"
: "rice", "price" : 340 }
{ "_id" : ObjectId("614251c0ec9cd36cd1db1f47"), "custid" : 201, "name" : "anu", "item" :
"rice", "price" : 340 }
{ "_id" : ObjectId("614251dcec9cd36cd1db1f48"), "custid" : 202, "name" : "Meera", "item"
: "sugar", "price" : 35 }
{ "_id" : ObjectId("614251f2ec9cd36cd1db1f49"), "custid" : 201, "name" : "anu", "item" :
"wheat", "price" : 40 }
```

{ "_id" : ObjectId("6142520dec9cd36cd1db1f4a"), "custid" : 202, "name" : "Meera", "item" : "teapower", "price" : 250 }

{ "_id" : ObjectId("61425230ec9cd36cd1db1f4b"), "custid" : 203, "name" : "navya", "item" : "wheat", "price" : 40 }

>var mapFunction1=function(){emit(this.custid,this.price);};

>var reduceFunction1=function(keyCustId,valuesPrices){return Array.sum(valuesPrices);};

> db.orders.mapReduce(mapFunction1,reduceFunction1,{out: "map_example"}) {

"result" : "map_example", "ok" : 1 }

> db.map_example.find()

{ "_id" : 201, "value" : 380 }

{ "_id" : 200, "value" : 340 }

{ "_id" : 203, "value" : 40 }

{ "_id" : 202, "value" : 285 }

> db.orders.find()

{ "_id" : ObjectId("61425321ec9cd36cd1db1f4c"), "custid" : 200, "name" : "maya", "item" : "rice", "price" : 340 }

{ "_id" : ObjectId("61425323ec9cd36cd1db1f4d"), "custid" : 201, "name" : "anu", "item" : "rice", "price" : 340 }

{ "_id" : ObjectId("61425325ec9cd36cd1db1f4e"), "custid" : 202, "name" : "Meera", "item" : "sugar", "price" : 35 }

{ "_id" : ObjectId("61425326ec9cd36cd1db1f4f"), "custid" : 201, "name" : "anu", "item" : "wheat", "price" : 40 }

{ "_id" : ObjectId("61425328ec9cd36cd1db1f50"), "custid" : 202, "name" : "Meera", "item" : "teapower", "price" : 250 }

{ "_id" : ObjectId("6142532bec9cd36cd1db1f51"), "custid" : 203, "name" : "navya", "item" : "wheat", "price" : 40 }