ASSIGNMENT NO 11

Problem Statements:

Using MapReduce function comply the following requirements

1. Display the salary of employees department wise

2. Display the count of employees departmentwise

3. Consider the collection “Orders “ with the following types of doccuments

{

\_id: ObjectId("50a8240b927d5d8b5891743c"),

cust\_id: "abc123",

status: 'A',

price: 25,

}

Return the Total Price Per Customer

4. Display the total salatry of employees working in computer department.

5. Consider each document in the zipcode collection has the following form:

{

"\_id": "10280",

"city": "NEW YORK",

"state": "NY",

"pop": 5574,

}

Return the city wise population of the cities in the NY state.

MongoDB shell version: 2.6.12

connecting to: test

> use pccoe2

switched to db pccoe2

> db.createCollection("emp1")

{ "ok" : 1 }

> db.emp1.insert({"deptno":101,"deptname":"comp","empname":"jan","sal":2000})

WriteResult({ "nInserted" : 1 })

> db.emp1.insert({"deptno":101,"deptname":"comp","empname":"nina","sal":1000})

WriteResult({ "nInserted" : 1 })

> db.emp1.insert({"deptno":102,"deptname":"it","empname":"nina","sal":7000})

WriteResult({ "nInserted" : 1 })

> db.emp1.mapReduce(

function() { emit( this.deptno, this.sal);},

function(keydeptid,valuesamt) {return Array.sum(valuesamt)},

{out:"total",}).find()

{ "\_id" : 101, "value" : 3000 }

{ "\_id" : 102, "value" : 7000 }

> db.d.mapReduce(

function() { emit(this.dept,1); },

db.emp1.mapReduce(function() { emit( this.deptno,this.sal);},

function(keydeptid, valuesamt) {return Array.sum(valuesamt)},

{out:"total",}).find()

^C

> db.emp1.mapReduce(

function() { emit( this.deptno, 1);},

function(keydeptid,valuesamt) {return Array.sum(valuesamt)},

{out:"total",}).find()

{ "\_id" : 101, "value" : 2 }

{ "\_id" : 102, "value" : 1 }

> db.createCollection("cust")

{ "ok" : 1 }

> db.emp1.insert({"custid":102,"status":"a","price":7000})

WriteResult({ "nInserted" : 1 })

> db.emp1.insert({"custid":101,"status":"a","price":500})

WriteResult({ "nInserted" : 1 })

> db.emp1.insert({"custid":101,"status":"b","price":700})

WriteResult({ "nInserted" : 1 })

> db.cust.mapReduce(

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

function(keydeptid, valuesamt) {return Array.sum(valuesamt)},

{out:"total",}).find()

> db.cust.mapReduce(

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

function(keycustid, valuesamt) {return Array.sum(valuesamt)},

{out:"total",}).find()

> db.cust.find()

> db.cust.insert({"custid":101,"status":"b","price":700})

WriteResult({ "nInserted" : 1 })

> db.cust.insert({"custid":101,"status":"a","price":500})

WriteResult({ "nInserted" : 1 })

> db.cust.insert({"custid":102,"status":"a","price":7000})

WriteResult({ "nInserted" : 1 })

> db.cust.mapReduce(

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

function(keycustid, valuesamt) {return Array.sum(valuesamt)},

{out:"total",}).find()

{ "\_id" : 101, "value" : 1200 }

{ "\_id" : 102, "value" : 7000 }

> db.emp1.mapReduce(

function() { emit( this.deptno, 1);},

function(keydeptid,valuesamt) {return Array.sum(valuesamt)},

{query:{deptname:"comp",out:"total"}).find()

^C

> db.cust.mapReduce(

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

function(keycustid, valuesamt) {return Array.sum(valuesamt)},

{query:{deptname:"comp"},out:"total",}).find()

> db.emp1.mapReduce(

function() { emit( this.deptno, this.sal);},

function(keydeptid,valuesamt) {return Array.sum(valuesamt)},

{query:{deptname:"comp"},out:"total",}).find()

{ "\_id" : 101, "value" : 3000 }

> db.createCollection("zipcode")

{ "ok" : 1 }

> db.zipcode.insert({"city":"new york","state":"ny","pop":5000})

WriteResult({ "nInserted" : 1 })

> db.zipcode.insert({"city":"new york","state":"ny","pop":2000})

WriteResult({ "nInserted" : 1 })

> db.zipcode.insert({"city":"new york1","state":"ny","pop":3000})

WriteResult({ "nInserted" : 1 })

> db.zipcode.insert({"city":"new york1","state":"jan","pop":3000})

WriteResult({ "nInserted" : 1 })

> db.zipcode.mapReduce(function() { emit( this.city,

this.pop);},function(keycustid, valuesamt) {return Array.sum(valuesamt)},{query:

{state:"ny"},out:"total",}).find()

{ "\_id" : "new york", "value" : 7000 }

{ "\_id" : "new york1", "value" : 3000 }