**Practice on online compiler as the reduceFunction is depreciated**

**MapReduce is a data processing paradigm that enables you to perform computations on large data sets in two main steps:**

1. **Map: Applies a mapping function to each document in a collection, typically generating key-value pairs.**
2. **Reduce: Combines the output of the map function, aggregating values associated with the same key.**

**MapReduce for Employee Salary Calculation Example:**

db.map.insertMany([

{ "\_id": 1, "product": "laptop", "quantity": 5, "price": 1000 },

{ "\_id": 2, "product": "phone", "quantity": 10, "price": 500 },

{ "\_id": 3, "product": "tablet", "quantity": 3, "price": 700 },

{ "\_id": 4, "product": "laptop", "quantity": 2, "price": 1000 },

{ "\_id": 5, "product": "phone", "quantity": 7, "price": 500 }

]

)

var mapFunction=function(){

emit(this.product,this.quantity\*this.price)}

var reduceFunction = function(key,value){

return Array.sum(value)}

db.map.mapReduce(mapFunction,reduceFunction,{out:"totalrevenue"})

db.totalrevenue.find()

clg example

db.doc.insertMany([

... { docId: 1, name: 'Prajwal', degree: 'MBBS', price: 100 },

... { docId: 2, name: 'Prasad', degree: 'Homopathy', price: 150 },

... { docId: 3, name: 'Sakshi', degree: 'MBBS', price: 200 }

... ]);

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("670e2c5225a9ad848b94be09"),

ObjectId("670e2c5225a9ad848b94be0a"),

ObjectId("670e2c5225a9ad848b94be0b")

]

}

> var mapFunction1 = function () {

... emit(this.degree, this.price);

... };

var reduceFunction1 = function (key, values) {

var total = 0;

var count = values.length;

values.forEach(function(value) {

total += value; // Sum up the prices

});

return total / count; // Return average

};

> db.doc.mapReduce( mapFunction1, reduceFunction1, { out: "output1" } );

{

"result" : "output1",

"timeMillis" : 371,

"counts" : {

"input" : 3,

"emit" : 3,

"reduce" : 1,

"output" : 2

},

"ok" : 1

}

> db.output1.find();

{ "\_id" : "Homopathy", "value" : 150 }

{ "\_id" : "MBBS", "value" : 150 }

> var mapFunction = function () {

... emit(this.degree, this.price);

... };

>

> var reduceFunction = function (key, values) {

... return Array.sum(values);

... };

> db.doc.mapReduce(

... mapFunction,

... reduceFunction,

... {

... out: "output\_sum"

... }

... );

{

"result" : "output\_sum",

"timeMillis" : 331,

"counts" : {

"input" : 3,

"emit" : 3,

"reduce" : 1,

"output" : 2

},

"ok" : 1

}

> db.output\_sum.find().pretty();

{ "\_id" : "Homopathy", "value" : 150 }

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