Ejemplos de MapReduce en JavaScript

Ejemplo1

```
use prueba
base = [
{"name" : "Tim",     "country" : "USA",      "age" : 15},
{"name" : "Sandra",  "country" : "USA",      "age" : 18},
{"name" : "Alex",    "country" : "France",   "age" : 19},
{"name" : "Zhong",   "country" : "Taiwan",   "age" : 19},
{"name" : "Tom",     "country" : "USA",      "age" : 20},
{"name" : "Marc",    "country" : "France",   "age" : 20},
{"name" : "Hao",     "country" : "Taiwan",   "age" : 12},
{"name" : "Jennifer","country" : "USA",      "age" : 15},
{"name" : "Jean",    "country" : "France",   "age" : 17},
{"name" : "James",   "country" : "USA",      "age" : 17},
{"name" : "Peter",   "country" : "USA",      "age" : 20},
{"name" : "Jorge",   "country" : "Portugal", "age" : 20},]
db.students.drop()
db.students.insertMany(base)

function mr() {
    return db.students.mapReduce(

        // Función MAP
        function () {
            emit(this.age, { count: 1 });
        },

        // Función REDUCE
        function (key, values) {
            var reduced = { count: 0 };
            values.forEach(function (val) {
                reduced.count += val.count;
            });
            return reduced;
        },

        // Configurations
        {
            out: 'Stats'
        });
}
mr()
db.Stats.find().pretty()
```

Ejemplo2

```
textos =[{ text: "Peter Piper picked a peck of pickled peppers" },
{ text: "A peck of pickled peppers Peter Piper picked" },
{ text: "If Peter Piper picked a peck of pickled peppers" },
{ text: "Where's the peck of pickled peppers Peter Piper picked?" }]
db.borrame.drop()
db.borrame.insertMany(textos)
function wordMap() {
    // Genera un arreglo de palabras para cada documento
    var words = this.text.match(/\w+/g);
    if (words === null)
        return;
    for (var i = 0; i < words.length; i++) {
        //emite cada palabra con count 1
        emit(words[i], { count : 1 });
    }
}
function wordReduce(key, values) {
    var total = 0;
    for (var i = 0; i < values.length; i++) {
        total += values[i].count;
    }
    return { count : total };
}
db.borrame.mapReduce(wordMap, wordReduce,{out: 'Stats'})
db.Stats.find().pretty()
```

Ejemplo3

```
use prueba
db.orders.drop()
d1 = { cust_id: "A123", amount: 500, status: "A" }
d2 = { cust_id: "A123", amount: 250, status: "A" }
d3 = { cust_id: "B212", amount: 200, status: "A" }
d4 = { cust_id: "A123", amount: 300, status: "D" }
db.orders.insertMany([d1, d2, d3, d4])

db.orders.mapReduce(
    function () { emit(this.cust_id, this.amount); },
    function (key, values) { return Array.sum(values) },
    {
        query: { status: "A" }, out: "order_totals"
    }
)
db.order_totals.find().pretty()
```

Ejemplo4

```
use prueba
db.orders.drop()
d1 = {
    cust_id: "abc123",
    ord_date: new Date("Oct 04, 2012"),
    status: 'A',
    price: 25,
    items: [{ sku: "m1m1", qty: 5, price: 2.5 }, { sku: "n2n4", qty: 5,
price: 2.5 }]
}
db.orders.insert(d1)
var mapFun = function () {
    emit(this.cust_id, this.price);
};
var reduceFun = function (keyCust, valuePrice) {
    return Array.sum(valuePrice);
};
db.orders.mapReduce( mapFun, reduceFun, { out: "map_red_01"})

db.map_red_01.find().pretty()
```

Ejemplo5

```
use prueba
d1 = {
    cust_id: "abc123",
    ord_date: new Date("Oct 04, 2011"),
    status: 'A',
    price: 25,
    items: [{ sku: "mmm", qty: 5, price: 2.5 },
    { sku: "nnn", qty: 5, price: 2.5 }]
}
d2 = {
    cust_id: "abc456",
    ord_date: new Date("Nov 04, 2012"),
    status: 'A',
    price: 45,
    items: [{ sku: "mmm", qty: 10, price: 2.5 },
    { sku: "nnn", qty: 8, price: 2.5 }]
}

db.orders.drop()
db.orders.insertMany([d1, d2])
var mapFunction = function () {
    for (var idx = 0; idx < this.items.length; idx++) {
        var key = this.items[idx].sku;
        var value = {
            count: 1,
            qty: this.items[idx].qty
        };
        emit(key, value);
    }
};
var reduceFunction = function (keySKU, countObjVals) {
    reducedVal = { count: 0, qty: 0 };

    for (var idx = 0; idx < countObjVals.length; idx++) {
        reducedVal.count += countObjVals[idx].count;
        reducedVal.qty += countObjVals[idx].qty;
    }

    return reducedVal;
};

var finalizeFunction = function (key, reducedVal) {

    reducedVal.avg = reducedVal.qty / reducedVal.count;

    return reducedVal;
```

```
};

db.orders.mapReduce(mapFunction,
    reduceFunction,
    {
        out: { merge: "totalypromedio" },
        query: {ord_date:{ $gt: new Date('01/01/2012') }
        },
        finalize: finalizeFunction
    }
)
db.totalypromedio.find().pretty()
```

Ejemplo6

```
use prueba
base = [
    { "_id": ObjectId("5b3d366fa040bab1fff5ca3d"), "class": "Philosophy
101", "startDate": ISODate("2016-02-10T06:00:00Z"), "students": [{
"fName": "Dale", "lName": "Cooper", "age": 42 }, { "fName": "Lucy",
"lName": "Moran", "age": 35 }, { "fName": "Tommy", "lName": "Hill",
"age": 44 }], "cost": 1600, "professor": "Paul Slugman", "topics":
"Socrates,Plato,Aristotle,Francis Bacon", "book": { "isbn": "1133612105",
"title": "Philosophy : A Text With Readings", "price": 165.42 } },
    { "_id": ObjectId("5b3d3691a040bab1fff5ca3e"), "class": "College
Algebra", "startDate": ISODate("2016-02-11T06:00:00Z"), "students": [{
"fName": "Dale", "lName": "Cooper", "age": 42 }, { "fName": "Laura",
"lName": "Palmer", "age": 22 }, { "fName": "Donna", "lName": "Hayward",
"age": 21 }, { "fName": "Shelly", "lName": "Johnson", "age": 24 }],
"cost": 1500, "professor": "Rhonda Smith", "topics": "Rational
Expressions,Linear Equations,Quadratic Equations", "book": { "isbn":
"0321671791", "title": "College Algebra", "price": 179.4 } },
    { "_id": ObjectId("5b3d36aba040bab1fff5ca3f"), "class": "Astronomy
101", "startDate": ISODate("2016-02-11T06:00:00Z"), "students": [{
"fName": "Bobby", "lName": "Briggs", "age": 21 }, { "fName": "Laura",
"lName": "Palmer", "age": 22 }, { "fName": "Audrey", "lName": "Horne",
"age": 20 }], "cost": 1650, "professor": "Paul Slugman", "topics":
"Sun,Mercury,Venus,Earth,Moon,Mars", "book": { "isbn": "0321815351",
"title": "Astronomy: Beginning Guide to Univ", "price": 129.45 } },
    { "_id": ObjectId("5b3d36c1a040bab1fff5ca40"), "class": "Geology
101", "startDate": ISODate("2016-02-12T06:00:00Z"), "students": [{
"fName": "Andy", "lName": "Brennan", "age": 36 }, { "fName": "Laura",
"lName": "Palmer", "age": 22 }, { "fName": "Audrey", "lName": "Horne",
"age": 20 }], "cost": 1450, "professor": "Alice Jones", "topics":
"Earth,Moon,Elements,Minerals", "book": { "isbn": "0321814061", "title":
"Earth : An Introduction to Physical Geology", "price": 130.65 } },
    { "_id": ObjectId("5b3d36dca040bab1fff5ca41"), "class": "Biology
101", "startDate": ISODate("2016-02-11T06:00:00Z"), "students": [{
"fName": "Andy", "lName": "Brennan", "age": 36 }, { "fName": "James",
"lName": "Hurley", "age": 25 }, { "fName": "Harry", "lName": "Truman",
"age": 41 }], "cost": 1550, "professor": "Alice Jones", "topics":
"Earth,Cell,Energy,Genetics,DNA", "book": { "isbn": "0547219474",
"title": "Holt McDougal Biology", "price": 104.3 } },
    { "_id": ObjectId("5b3d3703a040bab1fff5ca42"), "class": "Chemistry
101", "startDate": ISODate("2016-02-13T06:00:00Z"), "students": [{
"fName": "Bobby", "lName": "Briggs", "age": 21 }, { "fName": "Donna",
"lName": "Hayward", "age": 21 }, { "fName": "Audrey", "lName": "Horne",
"age": 20 }, { "fName": "James", "lName": "Hurley", "age": 25 }], "cost":
1600, "professor": "Alice Jones", "topics": "Matter,Energy,Atom,Periodic
Table", "book": { "isbn": "0547219474", "title": "Chemistry : Matter and
Change", "price": 104.3 } }]
```

```
db.classes.drop()
db.classes.insertMany(base)

var mapFunc = function () {
    emit(this.professor, 1);
}
var redFunc = function (pro, cta) {
    return Array.sum(cta);
}
db.classes.mapReduce(mapFunc, redFunc,
    { query: { professor: "Alice Jones" }, out: "result" })

db.result.find().pretty()
```

Ejemplo7

```
use prueba
base = [
    { "_id": ObjectId("5b3d366fa040bab1fff5ca3d"), "class": "Philosophy
101", "startDate": ISODate("2016-02-10T06:00:00Z"), "students": [{
"fName": "Dale", "lName": "Cooper", "age": 42 }, { "fName": "Lucy",
"lName": "Moran", "age": 35 }, { "fName": "Tommy", "lName": "Hill",
"age": 44 }], "cost": 1600, "professor": "Paul Slugman", "topics":
"Socrates,Plato,Aristotle,Francis Bacon", "book": { "isbn": "1133612105",
"title": "Philosophy : A Text With Readings", "price": 165.42 } },
    { "_id": ObjectId("5b3d3691a040bab1fff5ca3e"), "class": "College
Algebra", "startDate": ISODate("2016-02-11T06:00:00Z"), "students": [{
"fName": "Dale", "lName": "Cooper", "age": 42 }, { "fName": "Laura",
"lName": "Palmer", "age": 22 }, { "fName": "Donna", "lName": "Hayward",
"age": 21 }, { "fName": "Shelly", "lName": "Johnson", "age": 24 }],
"cost": 1500, "professor": "Rhonda Smith", "topics": "Rational
Expressions,Linear Equations,Quadratic Equations", "book": { "isbn":
"0321671791", "title": "College Algebra", "price": 179.4 } },
    { "_id": ObjectId("5b3d36aba040bab1fff5ca3f"), "class": "Astronomy
101", "startDate": ISODate("2016-02-11T06:00:00Z"), "students": [{
"fName": "Bobby", "lName": "Briggs", "age": 21 }, { "fName": "Laura",
"lName": "Palmer", "age": 22 }, { "fName": "Audrey", "lName": "Horne",
"age": 20 }], "cost": 1650, "professor": "Paul Slugman", "topics":
"Sun,Mercury,Venus,Earth,Moon,Mars", "book": { "isbn": "0321815351",
"title": "Astronomy: Beginning Guide to Univ", "price": 129.45 } },
    { "_id": ObjectId("5b3d36c1a040bab1fff5ca40"), "class": "Geology
101", "startDate": ISODate("2016-02-12T06:00:00Z"), "students": [{
"fName": "Andy", "lName": "Brennan", "age": 36 }, { "fName": "Laura",
"lName": "Palmer", "age": 22 }, { "fName": "Audrey", "lName": "Horne",
"age": 20 }], "cost": 1450, "professor": "Alice Jones", "topics":
"Earth,Moon,Elements,Minerals", "book": { "isbn": "0321814061", "title":
"Earth : An Introduction to Physical Geology", "price": 130.65 } },
    { "_id": ObjectId("5b3d36dca040bab1fff5ca41"), "class": "Biology
101", "startDate": ISODate("2016-02-11T06:00:00Z"), "students": [{
"fName": "Andy", "lName": "Brennan", "age": 36 }, { "fName": "James",
"lName": "Hurley", "age": 25 }, { "fName": "Harry", "lName": "Truman",
"age": 41 }], "cost": 1550, "professor": "Alice Jones", "topics":
"Earth,Cell,Energy,Genetics,DNA", "book": { "isbn": "0547219474",
"title": "Holt McDougal Biology", "price": 104.3 } },
    { "_id": ObjectId("5b3d3703a040bab1fff5ca42"), "class": "Chemistry
101", "startDate": ISODate("2016-02-13T06:00:00Z"), "students": [{
"fName": "Bobby", "lName": "Briggs", "age": 21 }, { "fName": "Donna",
"lName": "Hayward", "age": 21 }, { "fName": "Audrey", "lName": "Horne",
"age": 20 }, { "fName": "James", "lName": "Hurley", "age": 25 }], "cost":
1600, "professor": "Alice Jones", "topics": "Matter,Energy,Atom,Periodic
Table", "book": { "isbn": "0547219474", "title": "Chemistry : Matter and
Change", "price": 104.3 } }]
```

```
db.classes.drop()
db.classes.insertMany(base)

var mapFunction = function () {
    for (var idx = 0; idx < this.students.length; idx++) {
        var key = this.students[idx].fName +" "+this.students[idx].lName;
        emit(key, 1);
    }
};

var reduceFunction = function (key, countSt) {
    reducedVal = 0;
    for (var idx = 0; idx < countSt.length; idx++) {
        reducedVal += countSt[idx];
    }
    return reducedVal;
};

db.classes.mapReduce(mapFunction, reduceFunction, { out: "result" })

db.result.find().pretty()
```