

Name : Omkar.U.Hulawale

Class : TE-A

Roll No : 13165

Batch : A3

Aim : Implement Map reduce operation with suitable example using MongoDB

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS C:\Users\hulaw> mongosh
```

```
Current Mongosh Log ID: 6715c58a1b8b7ad4ca86b01c
```

```
Connecting to:
mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=
mongosh+2.3.2
```

```
Using MongoDB:      8.0.1
```

```
Using Mongosh:      2.3.2
```

For mongosh info see: <https://www.mongodb.com/docs/mongodb-shell/>

```
-----
```

The server generated these startup warnings when booting

2024-10-16T20:29:40.756+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

```
-----
```

```
test> show dbs
```

```
admin    40.00 KiB
```

```
bill     192.00 KiB
```

```
book     40.00 KiB
```

config 60.00 KiB

customer 108.00 KiB

local 40.00 KiB

test> // Use the TEB database

test> use TEB;

switched to db TEB

***Insert initial documents into the staff collection**

TEB> db.staff.insertMany([

... { tid: 1, tname: "shital", dname: "COMP", salary: 35000, hiredate: new Date('2014-08-23'),
bonus: 100, dob: new Date('1984-09-08') },

... { tid: 2, tname: "neha", dname: "IT", salary: 30000, hiredate: new Date('2012-02-02'), bonus:
200, dob: new Date('1980-07-18') },

... { tid: 3, tname: "pallavi", dname: "EnTC", salary: 40000, hiredate: new Date('2002-12-12'),
bonus: 150, dob: new Date('1975-01-08') },

... { tid: 4, tname: "saurabh", dname: "MECH", salary: 35000, hiredate: new Date('2010-10-18'),
bonus: 200, dob: new Date('1982-06-25') }
...]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6715c5981b8b7ad4ca86b01d'),

'1': ObjectId('6715c5981b8b7ad4ca86b01e'),

'2': ObjectId('6715c5981b8b7ad4ca86b01f'),

'3': ObjectId('6715c5981b8b7ad4ca86b020')

}

}

***Insert additional teachers with random data**

TEB> for (let i = 5; i <= 2000; i++) {

... db.staff.insert({

... tid: i,

```

...     tname: "Teacher" + i,
...     dname: "Department" + (Math.floor(Math.random() * 4) + 1), // Random department for
diversity
...     age: Math.floor(Math.random() * 120),
...     hiredate: new Date()
...   });
... }

```

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

```

{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6715c5a01b8b7ad4ca86b7ec') }
}

```

***MapReduce Example 1: Find total salary of teachers in each department**

```

TEB> var mapFunction1 = function() {
...   emit(this.dname, this.salary);
... };

```

```

TEB> var reduceFunction1 = function(key, values) {
...   return Array.sum(values);
... };

```

```

TEB> db.staff.mapReduce(mapFunction1, reduceFunction1, { out: 'ResultCollection' });

```

DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation instead.

See <https://docs.mongodb.com/manual/core/map-reduce> for details.

```

{ result: 'ResultCollection', ok: 1 }

```

```

TEB>

```

***Query the result of total salary by department**

```
TEB> print("Total Salary by Department:");
```

Total Salary by Department:

```
TEB> db.ResultCollection.find().pretty();
```

```
[
  { _id: 'MECH', value: 35000 },
  { _id: 'Department2', value: NaN },
  { _id: 'Department1', value: NaN },
  { _id: 'Department3', value: NaN },
  { _id: 'COMP', value: 35000 },
  { _id: 'EnTC', value: 40000 },
  { _id: 'Department4', value: NaN },
  { _id: 'IT', value: 30000 }
]
```

*** MapReduce Example 2: Find total number of teachers in each department**

```
TEB> var mapFunction2 = function() {
```

```
...   emit(this.dname, 1);
```

```
... };
```

```
TEB> var reduceFunction2 = function(key, values) {
```

```
...   return Array.sum(values);
```

```
... };
```

```
TEB> db.staff.mapReduce(mapFunction2, reduceFunction2, { out: 'resCollection' });
```

```
{ result: 'resCollection', ok: 1 }
```

***Query the result of total number of teachers by department**

```
TEB> print("Total Number of Teachers by Department:");
```

Total Number of Teachers by Department:

```
TEB> db.resCollection.find().pretty();
```

```
[
  { _id: 'MECH', value: 1 },
  { _id: 'Department2', value: 464 },
  { _id: 'Department1', value: 506 },
  { _id: 'Department3', value: 526 },
  { _id: 'COMP', value: 1 },
  { _id: 'EnTC', value: 1 },
  { _id: 'Department4', value: 500 },
  { _id: 'IT', value: 1 }
]
```

*** MapReduce Example 3: Calculating each key's value count**

```
TEB> var mapFunction3 = function() {
...   for (var key in this) {
...     emit(key, { "count": 1 });
...   }
... };
```

```
TEB> var reduceFunction3 = function(key, emits) {
...   let total = 0;
...   for (var i in emits) {
...     total += emits[i].count;
...   }
...   return { "count": total };
... };
```

```
TEB> db.staff.mapReduce(mapFunction3, reduceFunction3, { out: 'rescol' });  
{ result: 'rescol', ok: 1 }
```

***Query the result of each key's value count**

```
TEB> print("Count of Each Key:");
```

Count of Each Key:

```
TEB> db.rescol.find().pretty();  
[  
  { _id: 'salary', value: { count: 4 } },  
  { _id: '_id', value: { count: 2000 } },  
  { _id: 'tname', value: { count: 2000 } },  
  { _id: 'dob', value: { count: 4 } },  
  { _id: 'age', value: { count: 1996 } },  
  { _id: 'tid', value: { count: 2000 } },  
  { _id: 'bonus', value: { count: 4 } },  
  { _id: 'dname', value: { count: 2000 } },  
  { _id: 'hiredate', value: { count: 2000 } }  
]
```