

Title: MongoDB map-reduce operations

Problem: Implement map-reduce operations with suitable example using MongoDB

Objective:

To understand concept of map-reduce as data processing paradigm for condensing large volumes of data into useful aggregation results.

S/W H/H:

windows 10 64 bit, 8 GB RAM, 512 GB SSD
mongodb 5.0

Theory:

Map reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results.

Following is the syntax of mapreduce:

```
db.collection-name.mapReduce(  
  function() {  
    emit(key, value);  
  },  
  function(key, value) {  
    return reduceFunction;  
  },  
  { query: document,  
    out: document  
  }  
)
```

In the above syntax, map is a JS function that maps a value with a key and emits a key-value pair.

Reduce is a JS function that reduces or groups all the documents having the same key.

Out specifies the location of the map reduce query result.

Query specifies the optional selection criteria for selecting documents.

Conclusion:

Successfully implemented mapreduce operation in MongoDB.

```
show dbs
use assignment11
db
```

```
db.materials.insertMany([
{
  name:'aluminium',
  rate:200,
  weight:'Light'
},
{
  name:'steel, mild',
  rate:300,
  weight:'Medium'
},
{
  name:'steel, stainless',
  rate:100,
  weight:'Heavy'
},
{
  name:'abs plastic',
  rate:100,
  weight:'Light'
},
{
  name:'PVC',
  rate:200,
  weight:'Light'
}
])
```

```
db.materials.find().pretty()
```

```
db.materials.mapReduce(
function(){
  emit(this.name,this.rate);},
function(key, value){
  return Array.sum(value)},
{
  query:{weight:'Light'},
  out:'total_cost'
})
```

)

```
db.total_cost.find().pretty()
```

```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
> use assignment11
switched to db assignment11
> db
assignment11
>
> db.materials.insertMany([
... {
...   name:'aluminium',
...   rate:200,
...   weight:'Light'
... },
... {
...   name:'steel, mild',
...   rate:300,
...   weight:'Medium'
... },
... {
...   name:'steel, stainless',
...   rate:100,
...   weight:'Heavy'
... },
... {
...   name:'abs plastic',
...   rate:100,
...   weight:'Light'
... },
... {
...   name:'PVC',
...   rate:200,
...   weight:'Light'
... }
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("61972a3a250bf58329980b8b"),
    ObjectId("61972a3a250bf58329980b8c"),
    ObjectId("61972a3a250bf58329980b8d"),
    ObjectId("61972a3a250bf58329980b8e"),
    ObjectId("61972a3a250bf58329980b8f")
  ]
}
>
> db.materials.find().pretty()
{
  "_id" : ObjectId("61972a3a250bf58329980b8b"),
  "name" : "aluminium",
  "rate" : 200,
  "weight" : "Light"
}
{
  "_id" : ObjectId("61972a3a250bf58329980b8c"),
  "name" : "steel, mild",
  "rate" : 300,
  "weight" : "Medium"
}
{
  "_id" : ObjectId("61972a3a250bf58329980b8d"),
  "name" : "steel, stainless",
  "rate" : 100,
  "weight" : "Heavy"
}
{
  "_id" : ObjectId("61972a3a250bf58329980b8e"),
  "name" : "abs plastic",
  "rate" : 100,
  "weight" : "Light"
}
{
  "_id" : ObjectId("61972a3a250bf58329980b8f"),
  "name" : "PVC",
  "rate" : 200,
  "weight" : "Light"
}
>
> db.materials.mapReduce(
```

```
>
> db.total_cost.find().pretty()
{ "_id" : "abs plastic", "value" : 100 }
{ "_id" : "aluminium", "value" : 200 }
{ "_id" : "PVC", "value" : 200 }
>
```

C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe

```
> db.materials.mapReduce(
... function(){
...   emit(this.name,this.rate);},
... function(key, value){
...   return Array.sum(value)},
... {
...   query:{weight:'Light'},
...   out:'total_cost'
... }
... )
{ "result" : "total_cost", "ok" : 1 }
>
```