Assignment no:12

Map Reduce in MongoDB

```
> db.book13.find()
{ "_id" : ObjectId("5d996xbd51ffbfd62b1e690bf"), "name" : "understanding java", "pages" : 200 }
{ "_id" : ObjectId("5d996bdc1ffbfd62b1e690c0"), "name" : "understanding xml", "pages" : 200 }
{ "_id" : ObjectId("5d996be91ffbfd62b1e690c1"), "name" : "understanding python", "pages" : 300 }
{ "_id" : ObjectId("5d996bfb1ffbfd62b1e690c2"), "name" : "understanding JSON", "pages" : 100 }
{ "_id" : ObjectId("5d996c331ffbfd62b1e690c3"), "name" : "understanding MYsql", "pages" : 400 }
{ "_id" : ObjectId("5d996cc81ffbfd62b1e690c4"), "name" : "understanding sql", "pages" : 250 }
```

Find the number of books having pages less 250 pages and greater than that using Map Reduce.

The map function.

```
> var map = function() {
var category;
if ( this.pages >= 250 )
category = 'Big Books';
else
category = "Small Books";
emit(category, {name: this.name});
};
```

The Reduce function.

```
> var reduce = function(key, values) {
var sum = 0;
values.forEach(function(doc) {
sum += 1;
});
return {books: sum};
};
```

Running MapReduce against the books collection.

```
> var count = db.book13.mapReduce(map, reduce, {out: "book_results"});
> db[count.result].find()
{ "_id" : "Big Books", "value" : { "books" : 4 } }
{ "_id" : "Small Books", "value" : { "books" : 3 } }
```