3. **Map reduce operation with suitable example using MongoDB.**
   Create an **orders collection with fields customer id, order date, status, price and items, quantity.** Execute following queries using mapreduce. Perform map reduce operation on orders collection . Perform following queries using map reduce.

i. Display total price per customer.

```
var map=function(){emit(this.Cust_id,this.Amt);}
 var red=function(keys,values){return Array.sum(values);}
db.orders.mapReduce(map,red,{'out':'neworders'})
db.neworders.find().pretty()
```

ii. Display total price per customer having status= D

```
db.orders.mapReduce(map,red,{'query':{Status:'D'},'out':'new1'})
db.new1.find().pretty()
```

iii. Display total price for Status =P

```
db.orders.mapReduce(map,red,{'query':{Status:'P'},'out':'new2'})
db.new2.find().pretty()
```

iv. Finding count of all keys in orders collection

db.orders.mapReduce(map1,red1,{'out':'new3'})

```
5. {
6. "result" : "new3",
7. "timeMillis" : 356,
8. "counts" : {
9. "input" : 6,
10. "emit" : 62,
11. "reduce" : 11,
12. "output" : 11
13. },
14. "ok" : 1
15. }
```

16. > db.new3.find().pretty()

```
17{ "_id" : "Amt", "value" : { "CountOfKey" : 6 } }
18. { "_id" : "Cust_id", "value" : { "CountOfKey" : 6 } }
19. { "_id" : "Cust_name", "value" : { "CountOfKey" : 6 } }
20. { "_id" : "DtOfOrder", "value" : { "CountOfKey" : 6 } }
21. { "_id" : "Email_id", "value" : { "CountOfKey" : 2 } }
22. { "_id" : "Item_name", "value" : { "CountOfKey" : 6 } }
23. { "_id" : "Order_id", "value" : { "CountOfKey" : 6 } }
24. { "_id" : "Phone_no", "value" : { "CountOfKey" : 6 } }
25. { "_id" : "Qty", "value" : { "CountOfKey" : 6 } }
26. { "_id" : "Status", "value" : { "CountOfKey" : 6 } }
27. { "_id" : "_id", "value" : { "CountOfKey" : 6 } }
```