5. Aim: Design and Implement Map-reduce operation with suitable example using MongoDB.

Problem Statement: Design a map-reduce operations on a collection “orders” that contains documents of the following prototype. Solve the following . A={

```
    cust_id: "abc123",
    ord_date: new Date("Oct 04, 2012"),
    status: 'A',
price: 25,
gender :’F’,
rating: 1

}
```

```
> db.customers.insert({cust_id:"Avani",ord_date:new Date("May
09,2014"),status:'C',price:550,gender:'F',rating:5});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({cust_id:"Adwait",ord_date:new Date("May
09,2014"),status:'B',price:200,gender:'M',rating:4});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({cust_id:"Ram",ord_date:new Date("Jan
09,2014"),status:'B',price:120,gender:'M',rating:3});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({cust_id:"Sayali",ord_date:new Date("Nov
12,2013"),status:'A',price:50,gender:'F',rating:2});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({cust_id:"abc123",ord_date:new Date("Oct
04,2012"),status:'A',price:25,gender:'F',rating:1});
WriteResult({ "nInserted" : 1 })
> db.customers.find().pretty();
{
        "_id" : ObjectId("653a0fb259f97ee03740934d"),
        "cust_id" : "Avani",
        "ord_date" : ISODate("2014-05-08T18:30:00Z"),
        "status" : "C",
        "price" : 550,
        "gender" : "F",
        "rating" : 5
}
{
        "_id" : ObjectId("653a0fe759f97ee03740934e"),
        "cust_id" : "Adwait",
        "ord_date" : ISODate("2014-05-08T18:30:00Z"),
        "status" : "B",
        "price" : 200,
        "gender" : "M",
        "rating" : 4
}
```

```
{
        "_id" : ObjectId("653a100259f97ee03740934f"),
        "cust_id" : "Ram",
        "ord_date" : ISODate("2014-01-08T18:30:00Z"),
        "status" : "B",
        "price" : 120,
        "gender" : "M",
        "rating" : 3
}
{
    "_id" : ObjectId("653a101e59f97ee037409350"),    "cust_id"
: "Sayali",
        "ord_date" : ISODate("2013-11-11T18:30:00Z"),
        "status" : "A",
        "price" : 50,
        "gender" : "F",
        "rating" : 2
}
{
        "_id" : ObjectId("653a103559f97ee037409351"),
        "cust_id" : "abc123",
        "ord_date" : ISODate("2012-10-03T18:30:00Z"),
        "status" : "A",
        "price" : 25,
        "gender" : "F",
        "rating" : 1
}
```


a) Write a Map Reduce operation to Return the Total Price Per Customer Id

```
Db.orders.mapreduce(
function(){emit(this.cust_id,this.price);},
function(key,value){return Array.sum(value);},
{out:'oder_total_price'}).find()
```

```
Db.orders.mapreduce(
function(){emit(this.cust_id,this.price);},
function(key,value){return Array.sum(value);},
{query:{gender:'F'}, out:'oder_total_price'}).find()
```

```
> var map=function(){emit(this.cust_id,this.price)};

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

```
> db.customers.mapReduce(map,reduce,{out:"Result"});
{
        "result" : "Result",
        "timeMillis" : 391,
        "counts" : {
                "input" : 5,
                "emit" : 5,
                "reduce" : 0,
                "output" : 5
        },
        "ok" : 1
}

> db.Result.find();
{ "_id" : "Adwait", "value" : 200 }
{ "_id" : "Avani", "value" : 550 }
{ "_id" : "Ram", "value" : 120 }
{ "_id" : "Sayali", "value" : 50 }
{ "_id" : "abc123", "value" : 25 }
>
```

B. Count the number of female (F) and male (M) respondents in the orders collection

```
> var map=function(){var category; if(this.gender=='F') category="female"; else
category="male"; emit(category,{cust_id:this.Cust_id})};
> var reduce=function(key,values){var sum=0;
values.forEach(function(doc){sum+=1;});return{count:sum};};
> var count=db.customer.mapReduce(map,reduce,{out:"Gender count"});
> db[count.result].find();
{ "_id" : "female", "value" : { "count" : 2 } }
{ "_id" : "male", "value" : { "count" : 4 } }
```

C. Count the number of each type of rating (1, 2, 3, 4 or 5) for each orders

```
> var map=function(){emit(this.rating,1)};
> var reduce=function(key,values){return Array.sum(values)};
> db.customer.mapReduce(map,reduce,{out:"Result"});
{
        "result" : "Result",
        "timeMillis" : 355,
        "counts" : {
                "input" : 6,
                "emit" : 6,
                "reduce" : 2,
                "output" : 4
        },
        "ok" : 1
}
```

```
> db.Result.find();
{ "_id" : 1, "value" : 2 }
{ "_id" : 2, "value" : 1 }
{ "_id" : 3, "value" : 1 }
{ "_id" : 4, "value" : 2 }
```