```
use orders
switched to db orders
orders> db.orders.insertMany([
...
... { item: "apple", price: 10, quantity: 2, category: "fruit" },
...
... { item: "banana", price: 5, quantity: 5, category: "fruit" },
...
... { item: "carrot", price: 7, quantity: 3, category: "vegetable" },
...
... { item: "celery", price: 4, quantity: 7, category: "vegetable" },
...
... { item: "orange", price: 8, quantity: 4, category: "fruit" }
...
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68e7495ee410ccf56cba4815'),
    '1': ObjectId('68e7495ee410ccf56cba4816'),
    '2': ObjectId('68e7495ee410ccf56cba4817'),
    '3': ObjectId('68e7495ee410ccf56cba4818'),
    '4': ObjectId('68e7495ee410ccf56cba4819')
  }
}
```

Create

```
orders> var mapfunction=function(){emit(this.category,this.quantity);};

orders> var reducefunction=function(keyCategory,quantity){return Array.sum(quantity);};

orders> db.orders.mapReduce(mapfunction,reducefunction,{out:"total_quantity_per_category"});
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{ result: 'total_quantity_per_category', ok: 1 }

orders> db.total_quantity_per_category.find();
[ { _id: 'vegetable', value: 10 }, { _id: 'fruit', value: 11 } ]
```