Using MapReduce in MongoDB solve following queries on given below collection.

```
   { _id: 1, cust_id: "Ant O. Knee", ord_date: new Date("2020-03-01"), price: 25, items: [ { sku: "o
status: "A" },
   { _id: 2, cust_id: "Ant O. Knee", ord_date: new Date("2020-03-08"), price: 70, items: [ { sku: "o
status: "A" },
   { _id: 3, cust_id: "Busby Bee", ord_date: new Date("2020-03-08"), price: 50, items: [ { sku: "ora
status: "A" },
   { _id: 4, cust_id: "Busby Bee", ord_date: new Date("2020-03-18"), price: 25, items: [ { sku: "ora
   { _id: 5, cust_id: "Busby Bee", ord_date: new Date("2020-03-19"), price: 50, items: [ { sku: "cho
   { _id: 6, cust_id: "Cam Elot", ord_date: new Date("2020-03-19"), price: 35, items: [ { sku: "carr
status: "A" },
   { _id: 7, cust_id: "Cam Elot", ord_date: new Date("2020-03-20"), price: 25, items: [ { sku: "oran
   { _id: 8, cust_id: "Don Quis", ord_date: new Date("2020-03-20"), price: 75, items: [ { sku: "choc
status: "A" },
   { _id: 9, cust_id: "Don Quis", ord_date: new Date("2020-03-20"), price: 55, items: [ { sku: "carr
"oranges", qty: 10, price: 2.5 } ], status: "A" },
   { _id: 10, cust_id: "Don Quis", ord_date: new Date("2020-03-23"), price: 25, items: [ { sku: "ora
```

Database inventory and collection orders

```
{ _id: 1, cust_id: "Ant O. Knee", ord_date: new Date("2020-03-01"), price:
25, items: [ { sku: "oranges", qty: 5, price: 2.5 }, { sku: "apples", qty: 5,
price: 2.5 } ], status: "A" },
```

Exercise 01

1. **Total price paid by each customer**

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

> var reduce = function(cust_id,price) { return Array.sum(price);};

> db.orders.mapReduce(map,reduce,{out :{inline:1}});
```

```
{
    "results" : [
        {
            "_id" : "Busby Bee",
```

```
            "value" : 125
        },
        {
            "_id" : "Ant O. Knee",
            "value" : 95
        },
        {
            "_id" : "Don Quis",
            "value" : 155
        },
        {
            "_id" : "Cam Elot",
            "value" : 60
        }
    ],
    "ok" : 1
}
```


2. **Count of orders placed by each customer**

```
> db.orders.mapReduce(function(){emit(this.cust_id,1)},function(key,values){return Array.sum(values)},{out :{inline:1}})
```


```
{
    "results" : [
        {
            "_id" : "Busby Bee",
            "value" : 3
        },
```

```
        {
            "_id" : "Ant O. Knee",
            "value" : 2
        },
        {
            "_id" : "Don Quis",
            "value" : 3
        },
        {
            "_id" : "Cam Elot",
            "value" : 2
        }
    ],
    "ok" : 1
}
```


3. Total qty ordered of each item

```
> var mapFunction2 = function() {
... ...                 for (var idx = 0; idx < this.items.length; idx++) {
... ...                     var key = this.items[idx].sku;
... ...                     var value = {
... ...                                 count: 1,
... ...                                 qty: this.items[idx].qty
... ...                                };
... ...                     emit(key, value);
```

```
... ...                    }
... ...                  };

> var reduceFunction = function(keySKU, countObjVals) {
...                   reducedVal = {qty: 0 };
...
...                    for (var idx = 0; idx < countObjVals.length; idx++) {
...                       reducedVal.qty += countObjVals[idx].qty;
...                    }
...
...                    return reducedVal;
...                  };
>  db.orders.mapReduce(mapFunction,reduceFunction,{out:{inline:1}})
{
      "results" : [
            {
                  "_id" : "apples",
                  "value" : {
                        "qty" : 35
                  }
            },
            {
                  "_id" : "oranges",
                  "value" : {
                        "qty" : 63
```

```
            }
        },
        {
            "_id" : "chocolates",
            "value" : {
                "qty" : 15
            }
        },
        {
            "_id" : "pears",
            "value" : {
                "qty" : 10
            }
        },
        {
            "_id" : "carrots",
            "value" : {
                "qty" : 15
            }
        }
    ],
    "ok" : 1
}
```


4. **Count the no of order placed for each item**

```
> var mapFunction3 = function() {
...                     for (var idx = 0; idx < this.items.length; idx++) {
...                         var key = this.items[idx].sku;
...                         var value = {
...                                 no_of_orders: 1
...                             };
...                          emit(key, value);
...                       }
...                   };
> var reduceFunction3 = function(keySKU, countObjVals) {
...                     reducedVal = { no_of_orders: 0 };
...
...                     for (var idx = 0; idx < countObjVals.length; idx++) {
...                       reducedVal.no_of_orders += countObjVals[idx].no_of_orders;
...                     }
...
...                     return reducedVal;
...                   };
```

```
> db.orders.mapReduce(mapFunction3,reduceFunction3,{out:{inline:1}})
{
        "results" : [
                {
                        "_id" : "apples",
                        "value" : {
                                "no_of_orders" : 4
                        }
                },
                {
                        "_id" : "chocolates",
                        "value" : {
                                "no_of_orders" : 3
                        }
                },
                {
                        "_id" : "pears",
                        "value" : {
```

```
                "no_of_orders" : 1
            }
        },
        {
            "_id" : "carrots",
            "value" : {
                "no_of_orders" : 2
            }
        },
        {
            "_id" : "oranges",
            "value" : {
                "no_of_orders" : 7
            }
        }
    ],
    "ok" : 1
}
```

5. Calculate Order and Total Quantity with Average Quantity Per Item

```
> var mapFunction2 = function() {
...                     for (var idx = 0; idx < this.items.length; idx++) {
...                         var key = this.items[idx].sku;
...                         var value = {
...                                 count: 1,
...                                 qty: this.items[idx].qty
...                             };
...                         emit(key, value);
...                     }
...                 };
> var reduceFunction2 = function(keySKU, countObjVals) {
...                     reducedVal = { count: 0, qty: 0 };
...
...                     for (var idx = 0; idx < countObjVals.length; idx++) {
...                         reducedVal.count += countObjVals[idx].count;
```

```
...                     reducedVal.qty += countObjVals[idx].qty;

...                   }

...

...                   return reducedVal;

...                 };
> var finalizeFunction2 = function (key, reducedVal) {

...

...                   reducedVal.avg = reducedVal.qty/reducedVal.count;

...

...                   return reducedVal;

...

...                 };
> db.orders.mapReduce( mapFunction2,

...                   reduceFunction2,

...                   {

...                     out: { merge: "map_reduce_example" },

...                     query: { ord_date:

...                                 { $gt: new Date('01/01/2012') }
```

```
...                         },
...                       finalize: finalizeFunction2
...                     }
...                   )
{ "result" : "map_reduce_example", "ok" : 1 }
> db.map_reduce_example.find()
{ "_id" : "pears", "value" : { "count" : 1, "qty" : 10, "avg" : 10 } }
{ "_id" : "carrots", "value" : { "count" : 2, "qty" : 15, "avg" : 7.5 } }
{ "_id" : "apples", "value" : { "count" : 4, "qty" : 35, "avg" : 8.75 } }
{ "_id" : "oranges", "value" : { "count" : 7, "qty" : 63, "avg" : 9 } }
{ "_id" : "chocolates", "value" : { "count" : 3, "qty" : 15, "avg" : 5 } }
```