```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\akash>mongosh
Current Mongosh Log ID: 653e42bb203823ad4d639143
Connecting to:
mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appN
ame=mongosh+2.0.2
Using MongoDB:          7.0.2
Using Mongosh:          2.0.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-10-22T13:56:36.462+05:30: Access control is not enabled for the database.
Read and write access to data and configuration is unrestricted
------

test> use experiment11
switched to db experiment11
experiment11> db.orders.insertMany([{ _id: 1, cust_id: "Ant O. Knee", ord_date: new
Date("2020-03-01"), price: 25, itemsexperiment11> db.orders.insertMany([{ _id: 1,
cust_id: "Ant O. Knee", ord_date: new Date("2020-03-01"), price: 25,
itemsexperiment11> db.orders.insertMany([{ _id: 1, cust_id: "Ant O. Knee",
ord_date: new Date("2020-03-01"), price: 25, itemsexperiment11>
db.orders.insertMany([{ _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" },{ _id: 2, cust_id: "Ant O.
Knee", ord_date: new Date("2020-03-08"), price: 70, items: [ {sku: "oranges", qty:
8, price: 2.5 }, { sku: "chocolates", qty: 5, price: 10 } ], status: "A" },{ _id:
3, cust_id: "Busby Bee", ord_date: new Date("2020-03-08"), price: 50, items: [ {
sku:"oranges", qty: 10, price: 2.5 }, { sku: "pears", qty: 10, price: 2.5 } ],
status: "A" },{ _id: 4, cust_id: "Busby Bee", ord_date: new Date("2020-03-18"),
price: 25, items: [ { sku:"oranges", qty: 10, price: 2.5 } ], status: "A" }, { _id:
5, cust_id: "Busby Bee", ord_date: new Date("2020-03-19"), price: 50, items: [ {
sku:"chocolates", qty: 5, price: 10 } ], status: "A"}, { _id: 6, cust_id: "Cam
Elot", ord_date: new Date("2020-03-19"), price: 35, items: [ { sku:"carrots", qty:
10, price: 1.0 }, { sku: "apples", qty: 10, price: 2.5 } ], status: "A" },{ _id: 7,
cust_id: "Cam Elot", ord_date: new Date("2020-03-20"), price: 25, items: [ {
sku:"oranges", qty: 10, price: 2.5 } ], status: "A" },{ _id: 8, cust_id: "Don
Quis", ord_date: new Date("2020-03-20"), price: 75, items: [ { sku:"chocolates",
qty: 5, price: 10 }, { sku: "apples", qty: 10, price: 2.5 } ], status: "A" },{ _id:
9, cust_id: "Don Quis", ord_date: new Date("2020-03-20"), price: 55, items: [ {
sku:"carrots", qty: 5, price: 1.0 }, { sku: "apples", qty: 10, price: 2.5 }, { sku:
"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:"oranges", qty: 10,
price: 2.5 } ], status: "A" }])
{
  acknowledged: true,
```

```
    insertedIds: {
      '0': 1,
      '1': 2,
      '2': 3,
      '3': 4,
      '4': 5,
      '5': 6,
      '6': 7,
      '7': 8,
      '8': 9,
      '9': 10
    }
}
experiment11> var mapFunction1 = function() {emit(this.cust_id, this.price);};

experiment11> var reduceFunction1 = function(keyCustId, valuesPrices) {return
Array.sum(valuesPrices);};

experiment11> db.orders.mapReduce(mapFunction1,reduceFunction1,{ out:
"map_reduce_example" })
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation
instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{ result: 'map_reduce_example', ok: 1 }

experiment11> db.map_reduce_example.find().sort( { _id: 1 } )
[
  { _id: 'Ant O. Knee', value: 95 },
  { _id: 'Busby Bee', value: 125 },
  { _id: 'Cam Elot', value: 60 },
  { _id: 'Don Quis', value: 155 }
]
```