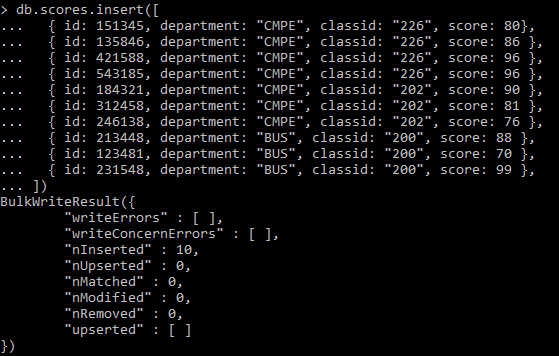
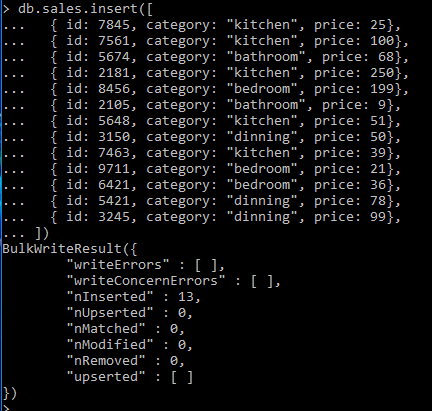
**Populating two collections**

Populating scores collection



Populating sales collection



**Pretty print collection 1**

> db.scores.find().pretty()

{

"\_id" : ObjectId("5901b1bcaf56317742058848"),

"id" : 151345,

"department" : "CMPE",

"classid" : "226",

"score" : 80

}

{

"\_id" : ObjectId("5901b1bcaf56317742058849"),

"id" : 135846,

"department" : "CMPE",

"classid" : "226",

"score" : 86

}

{

"\_id" : ObjectId("5901b1bcaf5631774205884a"),

"id" : 421588,

"department" : "CMPE",

"classid" : "226",

"score" : 96

}

{

"\_id" : ObjectId("5901b1bcaf5631774205884b"),

"id" : 543185,

"department" : "CMPE",

"classid" : "226",

"score" : 96

}

{

"\_id" : ObjectId("5901b1bcaf5631774205884c"),

"id" : 184321,

"department" : "CMPE",

"classid" : "202",

"score" : 90

}

{

"\_id" : ObjectId("5901b1bcaf5631774205884d"),

"id" : 312458,

"department" : "CMPE",

"classid" : "202",

"score" : 81

}

{

"\_id" : ObjectId("5901b1bcaf5631774205884e"),

"id" : 246138,

"department" : "CMPE",

"classid" : "202",

"score" : 76

}

{

"\_id" : ObjectId("5901b1bcaf5631774205884f"),

"id" : 213448,

"department" : "BUS",

"classid" : "200",

"score" : 88

}

{

"\_id" : ObjectId("5901b1bcaf56317742058850"),

"id" : 123481,

"department" : "BUS",

"classid" : "200",

"score" : 70

}

{

"\_id" : ObjectId("5901b1bcaf56317742058851"),

"id" : 231548,

"department" : "BUS",

"classid" : "200",

"score" : 99

}

**Pretty print collection 2**

> db.sales.find().pretty()

{

"\_id" : ObjectId("5901b1fdaf56317742058852"),

"id" : 7845,

"category" : "kitchen",

"price" : 25

}

{

"\_id" : ObjectId("5901b1fdaf56317742058853"),

"id" : 7561,

"category" : "kitchen",

"price" : 100

}

{

"\_id" : ObjectId("5901b1fdaf56317742058854"),

"id" : 5674,

"category" : "bathroom",

"price" : 68

}

{

"\_id" : ObjectId("5901b1fdaf56317742058855"),

"id" : 2181,

"category" : "kitchen",

"price" : 250

}

{

"\_id" : ObjectId("5901b1fdaf56317742058856"),

"id" : 8456,

"category" : "bedroom",

"price" : 199

}

{

"\_id" : ObjectId("5901b1fdaf56317742058857"),

"id" : 2105,

"category" : "bathroom",

"price" : 9

}

{

"\_id" : ObjectId("5901b1fdaf56317742058858"),

"id" : 5648,

"category" : "kitchen",

"price" : 51

}

{

"\_id" : ObjectId("5901b1fdaf56317742058859"),

"id" : 3150,

"category" : "dinning",

"price" : 50

}

{

"\_id" : ObjectId("5901b1fdaf5631774205885a"),

"id" : 7463,

"category" : "kitchen",

"price" : 39

}

{

"\_id" : ObjectId("5901b1fdaf5631774205885b"),

"id" : 9711,

"category" : "bedroom",

"price" : 21

}

{

"\_id" : ObjectId("5901b1fdaf5631774205885c"),

"id" : 6421,

"category" : "bedroom",

"price" : 36

}

{

"\_id" : ObjectId("5901b1fdaf5631774205885d"),

"id" : 5421,

"category" : "dinning",

"price" : 78

}

{

"\_id" : ObjectId("5901b1fdaf5631774205885e"),

"id" : 3245,

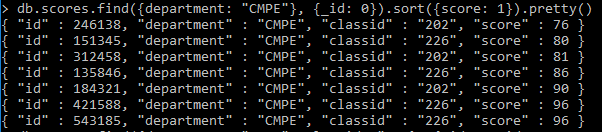
"category" : "dinning",

"price" : 99

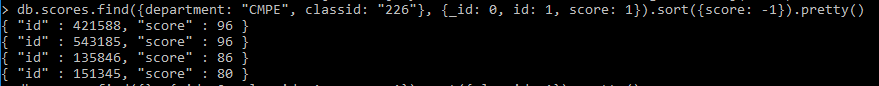
}

**Queries with various search criteria and sorting options for collection 1**

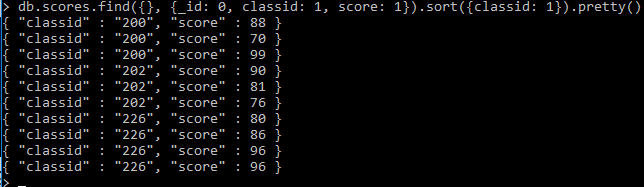
db.scores.find({department: "CMPE"}, {\_id: 0}).sort({score: 1}).pretty()



db.scores.find({department: "CMPE", classid: "226"}, {\_id: 0, id: 1, score: 1}).sort({score: -1}).pretty()

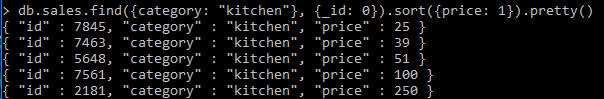


db.scores.find({}, {\_id: 0, classid: 1, score: 1}).sort({classid: 1}).pretty()

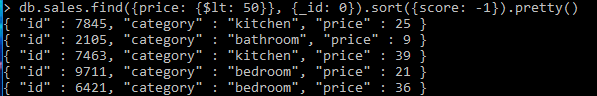


**Queries with various search criteria and sorting options for collection 2**

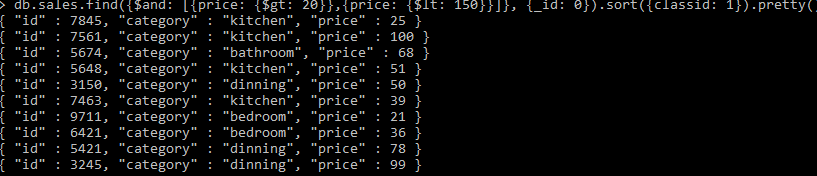
db.sales.find({category: "kitchen"}, {\_id: 0}).sort({price: 1}).pretty()



db.sales.find({price: {$lt: 50}}, {\_id: 0}).sort({score: -1}).pretty()

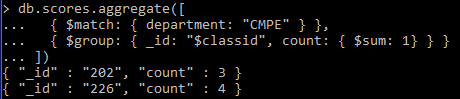


db.sales.find({$and: [{price: {$gt: 20}},{price: {$lt: 150}}]}, {\_id: 0}).sort({classid: 1}).pretty()



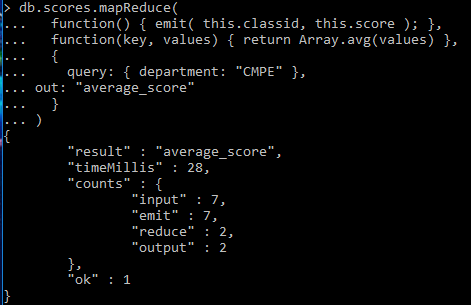
**Query that does “pipeline” aggregation for collection 1**

db.scores.aggregate([  
 { $match: { department: "CMPE" } },  
 { $group: { \_id: "$classid", count: { $sum: 1} } }  
])

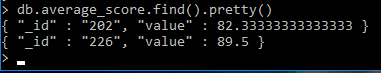


**Query that does aggregation using map-reduce for collection 1**

db.scores.mapReduce(  
 function() { emit( this.classid, this.score ); },  
 function(key, values) { return Array.avg(values) },  
 {  
 query: { department: "CMPE" },  
 out: "average\_score"  
 }  
)

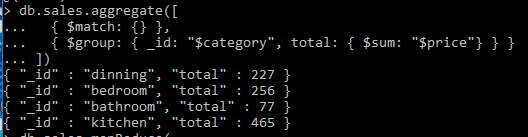


db.average\_score.find().pretty()



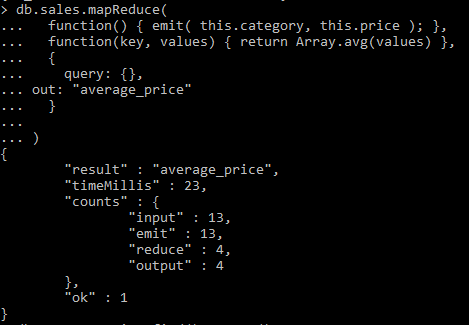
**Query that does “pipeline” aggregation for collection 2**

db.sales.aggregate([  
 { $match: {} },  
 { $group: { \_id: "$category", total: { $sum: "$price"} } }  
])



**Query that does aggregation using map-reduce for collection 2**

db.sales.mapReduce(  
 function() { emit( this.category, this.price ); },  
 function(key, values) { return Array.avg(values) },  
 {  
 query: {},  
 out: "average\_price"  
 }  
)



db.average\_price.find().pretty()

