mongo

MongoDB shell version: 2.6.10

connecting to: test

> use mmcoe

switched to db mmcoe

> db.createCollection("Product")

{ "ok" : 1 }

>
db.product.insertMany([{name:"soap",company:"dove",cost:25},{name:"chocolate",company:"5star
",cost:10},{name:"noodles",company:"maggie",cost:12}])

2017-09-26T10:07:33.469+0530 TypeError: Property 'insertMany' of object mmcoe.product is not a
function

>
db.Product.insertMany([{name:"soap",company:"dove",cost:25},{name:"chocolate",company:"5star
",cost:10},{name:"noodles",company:"maggie",cost:12}])

2017-09-26T10:07:51.783+0530 TypeError: Property 'insertMany' of object mmcoe.Product is not a
function

> db.Product.insertOne({name:"soap",company:"dove",cost:25})

2017-09-26T10:11:51.461+0530 TypeError: Property 'insertOne' of object mmcoe.Product is not a
function

> db.Product.insert({name:"soap",company:"dove",cost:25})

WriteResult({ "nInserted" : 1 })

> db.Product.insert({name:"chocolate",company:"5star",cost:10})

WriteResult({ "nInserted" : 1 })

> db.Product.insert({name:"noodles",company:"maggie",cost:11})

WriteResult({ "nInserted" : 1 })

> db.Product.insert({name:"conditioner",company:"lorealparis",cost:250})

WriteResult({ "nInserted" : 1 })

> db.Product.find().pretty()

{

        "_id" : ObjectId("59c9daa5b082c45ea5d1c6ec"),

        "name" : "soap",

        "company" : "dove",

```
                "cost" : 25
        }
        {
                "_id" : ObjectId("59c9dacbb082c45ea5d1c6ed"),
                "name" : "chocolate",
                "company" : "5star",
                "cost" : 10
        }
        {
                "_id" : ObjectId("59c9daeab082c45ea5d1c6ee"),
                "name" : "noodles",
                "company" : "maggie",
                "cost" : 11
        }
        {
                "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"),
                "name" : "conditioner",
                "company" : "lorealparis",
                "cost" : 250
        }
> db.Product.insert({name:"soap",company:"lorealparis",cost:150})
WriteResult({ "nInserted" : 1 })
> db.Product.find().pretty()
        {
                "_id" : ObjectId("59c9daa5b082c45ea5d1c6ec"),
                "name" : "soap",
                "company" : "dove",
                "cost" : 25
        }
        {
                "_id" : ObjectId("59c9dacbb082c45ea5d1c6ed"),
```

```
        "name" : "chocolate",

        "company" : "5star",

        "cost" : 10
}
{

        "_id" : ObjectId("59c9daeab082c45ea5d1c6ee"),

        "name" : "noodles",

        "company" : "maggie",

        "cost" : 11
}
{

        "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"),

        "name" : "conditioner",

        "company" : "lorealparis",

        "cost" : 250
}
{

        "_id" : ObjectId("59c9dbe1b082c45ea5d1c6f0"),

        "name" : "soap",

        "company" : "lorealparis",

        "cost" : 150
}
> db.Product.aggregate([{'$group':{'_id':$company','maxprice':{'$max':'$cost'},'minprice':{'$min':'$cost'}}}])

2017-09-26T10:27:26.035+0530 SyntaxError: Unexpected string

> db.Product.aggregate({'$group':{'_id':'$company','maxprice':{'$max':'$cost'},'minprice':{'$min':'$cost'}}})

{ "_id" : "lorealparis", "maxprice" : 250, "minprice" : 150 }

{ "_id" : "maggie", "maxprice" : 11, "minprice" : 11 }

{ "_id" : "5star", "maxprice" : 10, "minprice" : 10 }
```

```
{ "_id" : "dove", "maxprice" : 25, "minprice" : 25 }

> db.Product.aggregate({'$group':{'_id':'$company'}})

{ "_id" : "lorealparis" }

{ "_id" : "maggie" }

{ "_id" : "5star" }

{ "_id" : "dove" }

> db.Product.find()

{ "_id" : ObjectId("59c9daa5b082c45ea5d1c6ec"), "name" : "soap", "company" : "dove", "cost" : 25 }

{ "_id" : ObjectId("59c9dacbb082c45ea5d1c6ed"), "name" : "chocolate", "company" : "5star", "cost"
: 10 }

{ "_id" : ObjectId("59c9daeab082c45ea5d1c6ee"), "name" : "noodles", "company" : "maggie", "cost"
: 11 }

{ "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"), "name" : "conditioner", "company" : "lorealparis",
"cost" : 250 }

{ "_id" : ObjectId("59c9dbe1b082c45ea5d1c6f0"), "name" : "soap", "company" : "lorealparis", "cost"
: 150 }

> db.Prouct.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':1}})

> db.Prouct.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':1}})

> db.Prouct.aggregate({'$group':{'_id':'$company'}},{'$sort':{cost:1}})

> db.Prouct.aggregate({'$group':{'_id':'$company'}},{'$sort':{cost:1}})

> db.Prouct.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost:1'}})

2017-09-26T10:32:11.609+0530 SyntaxError: Unexpected token }

> db.Prouct.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':1}})

> db.Product.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':1}})

{ "_id" : "lorealparis" }

{ "_id" : "maggie" }

{ "_id" : "5star" }

{ "_id" : "dove" }

> db.Product.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':-1}})

{ "_id" : "lorealparis" }

{ "_id" : "maggie" }

{ "_id" : "5star" }
```

{ "_id" : "dove" }

> db.Product.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':'-1'}})

assert: command failed: {

    "errmsg" : "exception: $sort key ordering must be specified using a number or {$meta: 'textScore'}",

    "code" : 15974,

    "ok" : 0

} : aggregate failed

Error: command failed: {

    "errmsg" : "exception: $sort key ordering must be specified using a number or {$meta: 'textScore'}",

    "code" : 15974,

    "ok" : 0

} : aggregate failed

    at Error (<anonymous>)

    at doassert (src/mongo/shell/assert.js:11:14)

    at Function.assert.commandWorked (src/mongo/shell/assert.js:244:5)

    at DBCollection.aggregate (src/mongo/shell/collection.js:1149:12)

    at (shell):1:12

2017-09-26T10:33:38.498+0530 Error: command failed: {

    "errmsg" : "exception: $sort key ordering must be specified using a number or {$meta: 'textScore'}",

    "code" : 15974,

    "ok" : 0

} : aggregate failed at src/mongo/shell/assert.js:13

> db.Product.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':-1}})

{ "_id" : "lorealparis" }

{ "_id" : "maggie" }

{ "_id" : "5star" }

{ "_id" : "dove" }

> db.Product.aggregate({'$group':{'_id':'$company'}},{'$sort':{'cost':1}})

{ "_id" : "lorealparis" }

{ "_id" : "maggie" }

{ "_id" : "5star" }

{ "_id" : "dove" }

> db.Product.aggregate({'$limit':1})

{ "_id" : ObjectId("59c9daa5b082c45ea5d1c6ec"), "name" : "soap", "company" : "dove", "cost" : 25 }

> db.Product.aggregate({'$skip':1})

{ "_id" : ObjectId("59c9dacbb082c45ea5d1c6ed"), "name" : "chocolate", "company" : "5star", "cost" : 10 }

{ "_id" : ObjectId("59c9daeab082c45ea5d1c6ee"), "name" : "noodles", "company" : "maggie", "cost" : 11 }

{ "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"), "name" : "conditioner", "company" : "lorealparis", "cost" : 250 }

{ "_id" : ObjectId("59c9dbe1b082c45ea5d1c6f0"), "name" : "soap", "company" : "lorealparis", "cost" : 150 }

> db.Product.aggregate([{$match:{'company':'dove'}}])

{ "_id" : ObjectId("59c9daa5b082c45ea5d1c6ec"), "name" : "soap", "company" : "dove", "cost" : 25 }

> db.Product.aggregate([{$match:{'company':'lorealparis'}}]).pretty()

{

        "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"),

        "name" : "conditioner",

        "company" : "lorealparis",

        "cost" : 250

}

{

        "_id" : ObjectId("59c9dbe1b082c45ea5d1c6f0"),

        "name" : "soap",

        "company" : "lorealparis",

        "cost" : 150

}

> db.Product.aggregate({'$match':{'cost':{$gt:250}}}).pretty()

> db.Product.aggregate({'$match':{'cost':{$gt:25}}}).pretty()

{

```
        "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"),

        "name" : "conditioner",

        "company" : "lorealparis",

        "cost" : 250
}
{
        "_id" : ObjectId("59c9dbe1b082c45ea5d1c6f0"),

        "name" : "soap",

        "company" : "lorealparis",

        "cost" : 150
}
> db.Product.aggregate({'$match':{'cost':{$lt:25}}}).pretty()
{
        "_id" : ObjectId("59c9dacbb082c45ea5d1c6ed"),

        "name" : "chocolate",

        "company" : "5star",

        "cost" : 10
}
{
        "_id" : ObjectId("59c9daeab082c45ea5d1c6ee"),

        "name" : "noodles",

        "company" : "maggie",

        "cost" : 11
}
> db.Product.count()
5
> for(i=5;i<10;i++){db.Product.insert({_id:i,name:'soap'+i,cost:Math.floor(Math.random()*120)})};
WriteResult({ "nInserted" : 1 })
> db.Product.find();
{ "_id" : ObjectId("59c9daa5b082c45ea5d1c6ec"), "name" : "soap", "company" : "dove", "cost" : 25 }
```

{ "_id" : ObjectId("59c9dacbb082c45ea5d1c6ed"), "name" : "chocolate", "company" : "5star", "cost" : 10 }

{ "_id" : ObjectId("59c9daeab082c45ea5d1c6ee"), "name" : "noodles", "company" : "maggie", "cost" : 11 }

{ "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"), "name" : "conditioner", "company" : "lorealparis", "cost" : 250 }

{ "_id" : ObjectId("59c9dbe1b082c45ea5d1c6f0"), "name" : "soap", "company" : "lorealparis", "cost" : 150 }

{ "_id" : 5, "name" : "soap5", "cost" : 113 }

{ "_id" : 6, "name" : "soap6", "cost" : 28 }

{ "_id" : 7, "name" : "soap7", "cost" : 27 }

{ "_id" : 8, "name" : "soap8", "cost" : 6 }

{ "_id" : 9, "name" : "soap9", "cost" : 100 }

> for(i=1;i<5;i++){db.Product.insert({_id:i,name:'soap'+i,age:Math.floor(Math.random()*120)})};

WriteResult({ "nInserted" : 1 })

> db.Product.find();

{ "_id" : ObjectId("59c9daa5b082c45ea5d1c6ec"), "name" : "soap", "company" : "dove", "cost" : 25 }

{ "_id" : ObjectId("59c9dacbb082c45ea5d1c6ed"), "name" : "chocolate", "company" : "5star", "cost" : 10 }

{ "_id" : ObjectId("59c9daeab082c45ea5d1c6ee"), "name" : "noodles", "company" : "maggie", "cost" : 11 }

{ "_id" : ObjectId("59c9db2bb082c45ea5d1c6ef"), "name" : "conditioner", "company" : "lorealparis", "cost" : 250 }

{ "_id" : ObjectId("59c9dbe1b082c45ea5d1c6f0"), "name" : "soap", "company" : "lorealparis", "cost" : 150 }

{ "_id" : 5, "name" : "soap5", "cost" : 113 }

{ "_id" : 6, "name" : "soap6", "cost" : 28 }

{ "_id" : 7, "name" : "soap7", "cost" : 27 }

{ "_id" : 8, "name" : "soap8", "cost" : 6 }

{ "_id" : 9, "name" : "soap9", "cost" : 100 }

{ "_id" : 1, "name" : "soap1", "age" : 55 }

{ "_id" : 2, "name" : "soap2", "age" : 77 }

{ "_id" : 3, "name" : "soap3", "age" : 33 }

{ "_id" : 4, "name" : "soap4", "age" : 28 }

```
> db.Product.save({})
WriteResult({ "nInserted" : 1 })
> db.Product.find({name:'soap5'}).explain();
{
        "cursor" : "BasicCursor",
        "isMultiKey" : false,
        "n" : 1,
        "nscannedObjects" : 15,
        "nscanned" : 15,
        "nscannedObjectsAllPlans" : 15,
        "nscannedAllPlans" : 15,
        "scanAndOrder" : false,
        "indexOnly" : false,
        "nYields" : 0,
        "nChunkSkips" : 0,
        "millis" : 0,
        "server" : "mmcoe-desktop:27017",
        "filterSet" : false
}
> db.Product.ensureIndex({name:1});
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.Product.getIndexes();
[
        {
                "v" : 1,
                "key" : {
```

```
                                "_id" : 1
                        },
                        "name" : "_id_",
                        "ns" : "mmcoe.Product"
                },
                {
                        "v" : 1,
                        "key" : {
                                "name" : 1
                        },
                        "name" : "name_1",
                        "ns" : "mmcoe.Product"
                }
]
> db.Product.find({name:'soap5'}).explain();
{
        "cursor" : "BtreeCursor name_1",
        "isMultiKey" : false,
        "n" : 1,
        "nscannedObjects" : 1,
        "nscanned" : 1,
        "nscannedObjectsAllPlans" : 1,
        "nscannedAllPlans" : 1,
        "scanAndOrder" : false,
        "indexOnly" : false,
        "nYields" : 1,
        "nChunkSkips" : 0,
        "millis" : 60,
        "indexBounds" : {
                "name" : [
                        [
```

```
                                "soap5",

                                "soap5"

                        ]

                ]

        },

        "server" : "mmcoe-desktop:27017",

        "filterSet" : false

}
> db.Product.find({cost:25,name:{'$gt':'soap5','$lt':'soap8'}}).explain();

{

        "cursor" : "BtreeCursor name_1",

        "isMultiKey" : false,

        "n" : 0,

        "nscannedObjects" : 2,

        "nscanned" : 2,

        "nscannedObjectsAllPlans" : 2,

        "nscannedAllPlans" : 2,

        "scanAndOrder" : false,

        "indexOnly" : false,

        "nYields" : 0,

        "nChunkSkips" : 0,

        "millis" : 0,

        "indexBounds" : {

                "name" : [

                        [

                                "soap5",

                                "soap8"

                        ]

                ]

        },

        "server" : "mmcoe-desktop:27017",
```

```
          "filterSet" : false
}
> db.Product.ensureIndex({cost:25,tname:1});
{
          "createdCollectionAutomatically" : false,
          "numIndexesBefore" : 2,
          "numIndexesAfter" : 3,
          "ok" : 1
}
> db.Product.getIndexes();
[
          {
                    "v" : 1,
                    "key" : {
                              "_id" : 1
                    },
                    "name" : "_id_",
                    "ns" : "mmcoe.Product"
          },
          {
                    "v" : 1,
                    "key" : {
                              "name" : 1
                    },
                    "name" : "name_1",
                    "ns" : "mmcoe.Product"
          },
          {
                    "v" : 1,
                    "key" : {
                              "cost" : 25,
```

```
                "tname" : 1
            },
            "name" : "cost_25_tname_1",
            "ns" : "mmcoe.Product"
        }
    ]
>
```