Name: Kelly Rhodes

This is a final exam for ADB. The final will cover MongoDB and will be worth 35% of your total grade. Each task is worth 10 points. Partial credit will be awarded.

Please read the entire questions. If you are asked for multiple parts in an answer (Statement & Results), each part is worth 50% of the total question value.

In the examples below, text in ALL CAPS needs to be replaced with a value.

IE: Replace VENDOR NAME with Apple, Samsung, GNC or Acme in the statement:

```
_id: VENDOR NAME
```

Text is ALL CAPITALS needs to be replaced with some data of your choice.

Do not put spaces in the _id fields.

Start Exam

1. Start Mongo and login to your “ADB” database using the adbOwner user. ALL data will be in one collection. We will be storing products from vendors.

2. Import the Json file called “finalData.js” into the “ADB” database, inventory collection. Paste the import statement below. (Hint: Use mongoImport.)

```
mongoimport -u adbOwner -p adb -d adb -c inventory -file /media/sf_VBShare/finalData.js
```

3. Review the data imported in question 2.

Create documents for vendors in the 'inventory' collection. (Vendors can be fictional.)

Two vendor documents total. **Create the object first** in the format below, **then save the object** to the database.(2 steps). There should be no spaces in the _id or name fields. We're using a natural key for the _id field. The phone number type should be "main", "customer service", or "fax". Two phone numbers are required per vendor. Also, remember MongoDB is case sensitive.

Paste all 4 statements below.

```
_id: VENDOR NAME
name: VENDOR NAME
type:"vendor"
address: ADDRESS
city: CITY
state: STATE
ll: [LONGITUDE,LATTITUDE]        (Must be an array of numeric values)
phone: [
            {type:TYPE, number:NUMBER},
            {type:TYPE, number:NUMBER}
    ]
```

Use the data below to create your vendors.

Address	Longitude	Latitude
4092 Eastgate Drive, Orlando, FL	-79.441833	44.012893
451 E Altamonte Dr, Altamonte Springs	-81.375883	28.667207
One Microsoft Way, Redmond, WA	-122.131378	47.638197
381 Brea Canyon Road, Walnut, CA	-117.844840	34.013444

```
******** VENDOR 1 ***************
>var x = {
        _id:'Faro',
        name:'Faro',
        type:'vendor',
        address: '4092 Eastgate Drive',
        city: 'Orlando',
        state: 'FL',
        ll:[-79.441833, 44.012893],
        phone:[
```

```
                {type:'main', number:'407-123-4567'},
                {type:'fax',number:'407-123-1468'}
        ]
}

>db.inventory.save(x)

******** VENDOR 2 ***************
>var x = {
        _id:'AAA',
        name:'AAA',
        type:'vendor',
        address: '451 E Altamonte Dr',
        city: 'Altamonte Springs',
        state: 'FL',
        ll:[-81.375883, 28.667207],
        phone:[
                {type:'main', number:'407-123-4569'},
                {type:'fax',number:'407-123-1470'}
        ]
}

>db.inventory.save(x)
```

4. Create documents for products in the 'inventory' collection for your 2 new vendors.

4 products for 1 vendor

3 products for 1 vendor

Seven products over 2 vendors. Save the data to the database using a **single statement** for each document. Use the format below. Ex: Vendor=Apple, Product=IPad

Features should be an array of features (strings). Each array should be different and contain 1-3 features.

Feature Ex: ["bluetooth","WiFi","Retina Display","Shock Resistant"]

Paste the 7 statements below.

_id: PRODUCT NAME (No Spaces)
name: PRODUCT NAME
type: "product"
vendor: VENDOR NAME
category: CATEGORY (string - Electronic, TV, Clothing, Health, Furniture…)
features []

```
>db.inventory.save({
        _id:'EDGE',
        name:'EDGE',
        type:'product',
        vendor:'Faro',
        category: 'metrology',
        features:['touchScreen','9ft','7axis']
})

>db.inventory.save({
        _id:'GAGE',
        name:'GAGE',
        type:'product',
        vendor:'Faro',
        category: 'metrology',
        features:['4ft','6axis']
})

>db.inventory.save({
```

```
        _id:'FUSION',
        name:'FUSION',
        type:'product',
        vendor:'Faro',
        category: 'metrology',
        features:['kinematicProbe']
})

>db.inventory.save({
        _id:'PRIME',
        name:'PRIME',
        type:'product',
        vendor:'Faro',
        category: 'metrology',
        features:['6ft','7axis']
})

>db.inventory.save({
        _id:'TripTik',
        name:'TripTik',
        type:'product',
        vendor:'AAA',
        category: 'map',
        features:['directions','coupons']
})

>db.inventory.save({
        _id:'cruise',
        name:'cruise',
        type:'product',
        vendor:'AAA',
        category: 'travel',
        features:['7nights','pool']
})

>db.inventory.save({
        _id:'insurance',
```

```
        name:'insurance',
        type:'product',
        vendor:'AAA',
        category: 'insurance',
        features:['deductible']
})
```

5. Create an index (ascending) on the name field. Then run the command to list all of your indexes. Paste both commands and results.

```
> db.inventory.ensureIndex({ name : 1 })
RESULTS:
 {
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}

> db.inventory.getIndexKeys()
RESULTS:
[ { "_id" : 1 }, { "name" : 1 } ]
```

6. Create an index (descending) on the phone number field. Then run the command to list all of your indexes. Paste both commands and results.

```
> db.inventory.createIndex( { 'phone.number': -1 } )
RESULTS:
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}

> db.inventory.getIndexKeys()
RESULTS:
 [ { "_id" : 1 }, { "name" : 1 }, { "phone.number" : -1 } ]
```

7. Write a command to return one product document (type=”product”) by querying the vendor field (vendor = ???). Do not use findOne. Only one document should be returned. Paste command and results.

```
> db.inventory.find({vendor:'Faro'}).limit(1)
RESULTS:
{
        "_id" : "EDGE",
        "name" : "EDGE",
        "type" : "product",
         "vendor" : "Faro",
        "category" : "metrology",
        "features" : [ "touchScreen", "9ft", "7axis" ]
}
```

8. Write a command to return the products of 2 different vendors.

Paste command and results.

```
> db.inventory.find({vendor : {$in:['Faro', 'AAA']} } )
RESULTS:
{ "_id" : "EDGE", "name" : "EDGE", "type" : "product", "vendor" : "Faro", "category" : "metrology", "features" : [ "touchScreen", "9ft", "7axis" ] }
{ "_id" : "GAGE", "name" : "GAGE", "type" : "product", "vendor" : "Faro", "category" : "metrology", "features" : [ "4ft", "6axis" ] } { "_id" : "FUSION", "name" : "FUSION", "type" : "product", "vendor" : "Faro", "category" : "metrology", "features" : [ "kinematicProbe" ] }
{ "_id" : "TripTik", "name" : "TripTik", "type" : "product", "vendor" : "AAA", "category" : "map", "features" : [ "directions", "coupons" ] }
{ "_id" : "cruise", "name" : "cruise", "type" : "product", "vendor" : "AAA", "category" : "travel", "features" : [ "7nights", "pool" ] }
{ "_id" : "insurance", "name" : "insurance", "type" : "product", "vendor" : "AAA", "category" : "insurance", "features" : [ "deductible" ] }
 { "_id" : "PRIME", "name" : "PRIME", "type" : "product", "vendor" : "Faro", "category" : "metrology", "features" : [ "6ft", "7axis" ] }
```

9. Write a command to return vendors, by querying a specific phone number.

(Choose a number that exist in one of your vendor documents)

Your logic should be: Where type = VENDOR and phone number = NUMBER

Paste command and results.

```
> db.inventory.find({$and:[{type:'vendor'},{'phone.number':'407-123-1470'}]})
```

RESULTS:

{ "_id" : "AAA", "name" : "AAA", "type" : "vendor", "address" : "451 E Altamonte Dr", "city" : "Altamonte Springs", "state" : "FL", "ll" : [-81.375883, 28.667207], "phone" : [{ "type" : "main", "number" : "407-123-4569" }, { "type" : "fax", "number" : "407-123-1470" }] }

10. Add a new field called “rating” to any 8 product documents using the update command with $set. These values should not repeat (all ratings cannot be 7) and must be a numeric. All vendors should have a rating on at least 1 product. Paste ALL commands below.

rating: Number between 1-10 (This is the consumer rating.)

```
>db.inventory.update( {_id:'Battleship'}, {$set: {'rating':1}} )
>db.inventory.update( {_id:'PS3'}, {$set: {'rating':2}} )
>db.inventory.update( {_id:'Power_Glove'}, {$set: {'rating':3}} )
>db.inventory.update( {_id:'BigMac'}, {$set: {'rating':4}} )
>db.inventory.update( {_id:'ZpumpFusion'}, {$set: {'rating':5}} )
>db.inventory.update( {_id:'cruise'}, {$set: {'rating':6}} )
>db.inventory.update( {_id:'GAGE'}, {$set: {'rating':7}} )
>db.inventory.update( {_id:'FUSION'}, {$set: {'rating':8}} )
```

11. Write a command to add “EMP Resistant” to the features array on one of the product documents. Paste the command.

```
> db.inventory.update({name:'EDGE'},{$push:{features:'EMP Resistant'}})
```

12. Write a command to create an index on the ll array for geospatial searching. Remember ll contains the longitude and latitude, use the appropriate type of index. Paste the command.

```
>db.inventory.ensureIndex({ll:'2d'})
```

13. Write a command to return the closest vendor using the ll array. For your current location use Full Sail 3300 University Boulevard, Winter Park, Fl, 32792, Long:-81.30151, Lat: 28.59716

Paste the command and results.

```
> db.inventory.find({ll:{$near:[-81.30151, 28.59716]}}).limit(1)
```

RESULTS:

{
    "_id" : "AAA",
    "name" : "AAA",

```
        "type" : "vendor",
        "address" : "451 E Altamonte Dr",
        "city" : "Altamonte Springs",
        "state" : "FL",
        "ll" : [ -81.375883, 28.667207 ],
        "phone" : [
                { "type" : "main", "number" : "407-123-4569" },
                { "type" : "fax", "number" : "407-123-1470" }
        ]
}
```


14. Write a command to return the count of all documents in the database. Paste the command and results.

```
> db.inventory.find().count({})
```
RESULTS:
23

15. Write a command to return the product with the second highest rating. Only one document should be returned.

Paste the command and results.

```
> db.inventory.find().sort({rating:-1}).skip(1).limit(1)
```
RESULTS:

```
{
        "_id" : "GAGE",
        "name" : "GAGE",
        "type" : "product",
        "vendor" : "Faro",
        "category" : "metrology",
        "features" : [ "4ft", "6axis" ],
        "rating" : 7
}
```


16. Write a command to return the average “rating” for all products (type=product) in the database by vendor. Only include documents with an “rating” field. IE: if there is not an “rating” field, do not count it as a 0. (Hint: Use the group command.)

Paste the command and results.

```
>db.inventory.group({
        cond: {$and:[{rating: { $exists: true }},{type:'product'}]},
        key: {type: true},
        initial: {totalRating: 0, count: 0},
        reduce: function(obj,prev){
                prev.totalRating += obj.rating;
                prev.count++; },
        finalize: function(out){ out.avgRating = out.totalRating / out.count; }
 })
```

RESULTS:

```
 [ {
        "type" : "product",
        "totalRating" : 31,
        "count" : 7,
        "avgRating" : 4.428571428571429
} ]
```

17. Write a command to return 1 vendor by the type and name field.
Your logic should be: where type = TYPE and name = NAME
Paste the command and results.

```
> db.inventory.find({$and:[{type:'vendor'},{name:'Mattel'}]})
```

RESULTS:

```
{
        "_id" : "Mattel",
       "name" : "Mattel",
       "type" : "vendor",
       "address" : "763 Runies Ct",
        "city" : "El Segundo",
       "state" : "CA",
       "ll" : [ -118.416465, 33.91918 ],
       "phone" : [
                { "type" : "main", "number" : "888-0=298-1236" },
                { "type" : "Cust Serv", "number" : "445-332-9987" }
       ]
}
```

18. Write a command to delete a single product. Filter on the document's unique id (_id).
Paste the command below.

```
> db.inventory.remove( { '_id' : 'ZpumpFusion' } )
```

19. Write a command to return the count of all product documents in your database.
Paste the command and results below.

```
> db.inventory.find({'type':'product'}).count()
RESULTS:
15
```

20. Write a command to remove all products for **one vendor** from your database. Just the products.
Paste the command.

```
> db.inventory.remove( { 'vendor' : 'Mattel' } )
```

21. Export your collection to a csv format using the mongoexport command. Only export the numeric and string fields for both products and vendors. (Do not export array fields)
Paste the command below.

```
mongoexport -u adbOwner -p adb -d adb -c inventory --csv --out
/media/sf_VBShare/inventory.csv --fields _id,name,type,vendor,category,address,city,state
```

FINISHING YOUR EXAM

1. Save your files. Convert this word doc to a PDF.
   Name your files (pdf & csv) FIRSTINITIAL LASTNAME.XXX
2. Submit the PDF & CSV to FSO under "Practical Exam".