3. Insert all the records from the given file into the **users** collection.

   1) Query: db.users.insertMany(*All the records*)

   2) Result:


```
{
    "acknowledged" : true,
    "insertedIds" : [
        ObjectId("5dea356c9202fec00b8b092a"),
        ObjectId("5dea356c9202fec00b8b092b"),
        ObjectId("5dea356c9202fec00b8b092c"),
        ObjectId("5dea356c9202fec00b8b092d"),
        ObjectId("5dea356c9202fec00b8b092e"),
        ObjectId("5dea356c9202fec00b8b092f"),
        ObjectId("5dea356c9202fec00b8b0930"),
        ObjectId("5dea356c9202fec00b8b0931"),
        ObjectId("5dea356c9202fec00b8b0932"),
        ObjectId("5dea356c9202fec00b8b0933")
    ]
}
```


4. Retrieve all the users sorted by name.

   1) Query: db.users.find().sort({"Name":1})

   2) Result:


```
{ "_id" : ObjectId("5dea356c9202fec00b8b092b"), "Name" : "Aguirre Fox", "Address" : { "StreetNumber" : 540, "streetName" : "High Street", "city" : "Bloomington", "state" : "SC", "ZIPCode" : 29823 }, "Friends" : [ "Glenn Mcbride", "Marlene Macias", "Constance Arnold", "Beard Dotson", "Hester Lowe" ], "Active" : true, "DOB" : "Sat Mar 15 2014 06:04:01 GMT+0000 (UTC)", "Age" : 49 }
{ "_id" : ObjectId("5dea356c9202fec00b8b0932"), "Name" : "Aimee Mcintosh", "Address" : { "StreetNumber" : 145, "streetName" : "Boardwalk ", "city" : "Mahtowa", "state" : "MA", "ZIPCode" : 35051 }, "Friends" : [ "Chase Wyatt", "Kelly Hewitt", "Michael Rodriguez" ], "Active" : false, "DOB" : "Sun Mar 10 1996 19:54:41 GMT+0000 (UTC)", "Age" : 38 }
{ "_id" : ObjectId("5dea356c9202fec00b8b092d"), "Name" : "Cooke Schroeder", "Address" : { "StreetNumber" : 246, "streetName" : "Huntington Street", "city" : "Allendale", "state" : "NH", "ZIPCode" : 64947 }, "Friends" : [ "Rebekah Winters", "Grace Lewis", "Stephanie Hyde" ], "Active" : false, "DOB" : "Wed Oct 19 2016 15:38:31 GMT+0000 (UTC)", "Age" : 32 }
{ "_id" : ObjectId("5dea356c9202fec00b8b092a"), "Name" : "Craft Parks", "Address" : { "StreetNumber" : 397, "streetName" : "Hudson Avenue", "city" : "Glenbrook", "state" : "UT", "ZIPCode" : 96867 }, "Friends" : [ "Love Short", "Dickerson Brock", "Berg Levy", "Lottie Pickett" ], "Active" : false, "DOB" : "Tue May 21 1996 13:17:58 GMT+0000 (UTC)", "Age" : 42 }
{ "_id" : ObjectId("5dea356c9202fec00b8b0933"), "Name" : "Hayes Weaver", "Address" : { "StreetNumber" : 722, "streetName" : "Diamond Street", "city" : "Belva", "state" : "OH", "ZIPCode" : 90952 }, "Friends" : [ "Cecilia Oneill", "Trujillo Wilkins", "Clara Day", "Briana Ellis" ], "Active" : true, "DOB" : "Thu Sep 21 2000 06:45:36 GMT+0000 (UTC)", "Age" : 44 }
{ "_id" : ObjectId("5dea356c9202fec00b8b0931"), "Name" : "Levine Johnston", "Address" : { "StreetNumber" : 737, "streetName" : "McKinley Avenue", "city" : "Helen", "state" : "DE", "ZIPCode" : 48935 }, "Friends" : [ "Willis Morrow", "Sutton Massey", "Gibson Herrera" ], "Active" : false, "DOB" : "Sat Jan 09 1988 00:04:53 GMT+0000 (UTC)", "Age" : 44 }
{ "_id" : ObjectId("5dea356c9202fec00b8b092e"), "Name" : "Patrick Thornton", "Address" : { "StreetNumber" : 152, "streetName" : "Newkirk Avenue", "city" : "Summertown", "state" : "ID", "ZIPCode" : 95160 }, "Friends" : [ "Small Barlow", "Carson Cherry", "Leah Sherman", "Joyner Buckner" ], "Active" : true, "DOB" : "Sat Jan 12 2002 19:11:20 GMT+0000 (UTC)", "Age" : 25 }
{ "_id" : ObjectId("5dea356c9202fec00b8b092c"), "Name" : "Sandy Oneil", "Address" : { "StreetNumber" : 819, "streetName" : "Pierrepont Place", "city" : "Gibsonia", "state" : "AR", "ZIPCode" : 10832 }, "Friends" : [ "Lorraine Berry", "Ola Brewer", "Sharlene Franklin", "Melanie Lynn", "Chandra Gilliam" ], "Active" : false, "DOB" : "Thu Aug 07 1980 05:30:31 GMT+0000 (UTC)", "Age" : 41 }
{ "_id" : ObjectId("5dea356c9202fec00b8b0930"), "Name" : "Susan Graham", "Address" : { "StreetNumber" : 797, "streetName" : "Doone Court", "city" : "Frierson", "state" : "IL", "ZIPCode" : 15612 }, "Friends" : [ "Clark Sharpe", "Guerra Goodman", "Vinson Jones", "Swanson Avery", "Socorro Morse" ], "Active" : false, "DOB" : "Fri Mar 17 1972 15:42:29 GMT+0000 (UTC)", "Age" : 41 }
{ "_id" : ObjectId("5dea356c9202fec00b8b092f"), "Name" : "Workman Holloway", "Address" : { "StreetNumber" : 955, "streetName" : "Covert Street", "city" : "Sylvanite", "state" : "GA", "ZIPCode" : 95335 }, "Friends" : [ "Sullivan Blackwell", "Ratliff Mccray" ], "Active" : false, "DOB" : "Thu Nov 12 1981 05:06:00 GMT+0000 (UTC)", "Age" : 29 }
```


5. List *only the id and names* sorted in reverse alphabetical order by name (Z-to-A).

   1) Query: db.users.find({}, {Name:1, _id:1}).sort({Name:-1})

   2) Result:


```
{ "_id" : ObjectId("5dea356c9202fec00b8b092f"), "Name" : "Workman Holloway" }
{ "_id" : ObjectId("5dea356c9202fec00b8b0930"), "Name" : "Susan Graham" }
{ "_id" : ObjectId("5dea356c9202fec00b8b092c"), "Name" : "Sandy Oneil" }
{ "_id" : ObjectId("5dea356c9202fec00b8b092e"), "Name" : "Patrick Thornton" }
{ "_id" : ObjectId("5dea356c9202fec00b8b0931"), "Name" : "Levine Johnston" }
{ "_id" : ObjectId("5dea356c9202fec00b8b0933"), "Name" : "Hayes Weaver" }
{ "_id" : ObjectId("5dea356c9202fec00b8b092a"), "Name" : "Craft Parks" }
{ "_id" : ObjectId("5dea356c9202fec00b8b092d"), "Name" : "Cooke Schroeder" }
{ "_id" : ObjectId("5dea356c9202fec00b8b0932"), "Name" : "Aimee Mcintosh" }
{ "_id" : ObjectId("5dea356c9202fec00b8b092b"), "Name" : "Aguirre Fox" }
```


6. Is the attribute name case-sensitive? Try with the previous query.

   1) Query: db.users.find({}, {Name:1, _id:1}).sort({NAME:1}) ; db.users.find({}, {NAME:1, _id:1}).sort({NAME:1})

   2) Result: We can see from the result that the attribute name is case-sensitive.


```
{ "_id" : ObjectId("5dea356c9202fec00b8b092a"), "Name" : "Craft Parks" }    { "_id" : ObjectId("5dea356c9202fec00b8b092a") }
{ "_id" : ObjectId("5dea356c9202fec00b8b092b"), "Name" : "Aguirre Fox" }    { "_id" : ObjectId("5dea356c9202fec00b8b092b") }
{ "_id" : ObjectId("5dea356c9202fec00b8b092c"), "Name" : "Sandy Oneil" }    { "_id" : ObjectId("5dea356c9202fec00b8b092c") }
{ "_id" : ObjectId("5dea356c9202fec00b8b092d"), "Name" : "Cooke Schroeder" }{ "_id" : ObjectId("5dea356c9202fec00b8b092d") }
{ "_id" : ObjectId("5dea356c9202fec00b8b092e"), "Name" : "Patrick Thornton" }{ "_id" : ObjectId("5dea356c9202fec00b8b092e") }
{ "_id" : ObjectId("5dea356c9202fec00b8b092f"), "Name" : "Workman Holloway" }{ "_id" : ObjectId("5dea356c9202fec00b8b092f") }
{ "_id" : ObjectId("5dea356c9202fec00b8b0930"), "Name" : "Susan Graham" }    { "_id" : ObjectId("5dea356c9202fec00b8b0930") }
{ "_id" : ObjectId("5dea356c9202fec00b8b0931"), "Name" : "Levine Johnston" }{ "_id" : ObjectId("5dea356c9202fec00b8b0931") }
{ "_id" : ObjectId("5dea356c9202fec00b8b0932"), "Name" : "Aimee Mcintosh" } { "_id" : ObjectId("5dea356c9202fec00b8b0932") }
{ "_id" : ObjectId("5dea356c9202fec00b8b0933"), "Name" : "Hayes Weaver" }   { "_id" : ObjectId("5dea356c9202fec00b8b0933") }
```


7. Repeat #5 but do not show the _id field.

   1) Query: db.users.find({}, {Name:1, _id:0}).sort({Name:-1})

   2) Result:


```
{ "Name" : "Workman Holloway" }
{ "Name" : "Susan Graham" }
{ "Name" : "Sandy Oneil" }
{ "Name" : "Patrick Thornton" }
{ "Name" : "Levine Johnston" }
{ "Name" : "Hayes Weaver" }
{ "Name" : "Craft Parks" }
{ "Name" : "Cooke Schroeder" }
{ "Name" : "Aimee Mcintosh" }
{ "Name" : "Aguirre Fox" }
```


8. Insert the following document to the collection. {Name: {First: “David”, Last: “Bark”}}

1) Query: db.users.insertOne({Name:{First:"David",Last:"Bark"}})

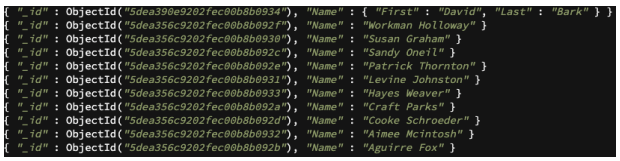
2) Result:

```
{ "_id" : ObjectId("5dea390e9202fec00b8b0934"), "Name" : { "First" : "David", "Last" : "Bark" } }
```

9. Rerun query #5. Where do you expect the new record to be located in the sort order? Verify the answer and explain.

   1) Answer: The new record should be the first one. Because the ascii of the punctuation "{" is the largest.

   2) Result:



10. Insert the following object into the users collection. {Name: [“David”, “Bark”]}

    1) Query: db.users.insert({Name:["David","Bark"]})

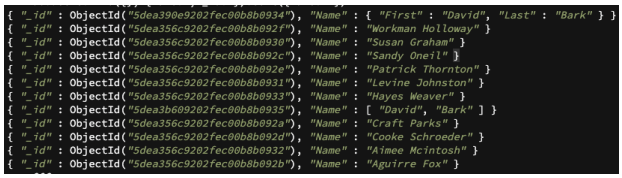
    2) Result:

```
{ "_id" : ObjectId("5dea3b609202fec00b8b0935"), "Name" : [ "David", "Bark" ] }
```

11. Repeat #5. Where do you expect the new object to appear in the sort order. Verify your answer and explain after running the query.

    1) Answer: The result should be the second, since the "[" is the second largest.

    2) Result:

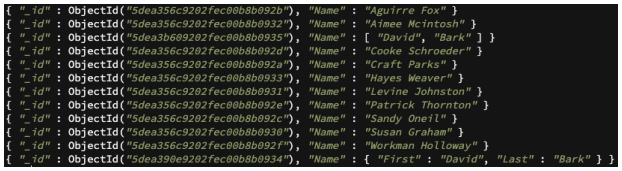


    I am so sorry about that, after two days of thinking this problem, I still couldn't figure out why this is not in the second position. I used to think about the objectid, however the answer is still wrong if sorted by objectid.

12. Repeat #5 again but this time sort the name in ascending order. Where do you expect the last inserted record to appear this time? Does it appear in the same position? Explain why or why not.

    1) Answer: The result should be the last second, since the "[" is the second largest.

    2) Result:



    Still sorry that I couldn't figure this problem out.

13. Build an index on the Name field for the users collection. Is MongoDB able to build the index on that field with the different value types stored in the Name field?

    1) Query: db.users.createIndex({Name:1}). Yes, MongoDB can build the index on the field with different value types.

    2) Result: