Lab: MongoDB Queries

1. Go the testDB database using the following command

   use testDB

2. Use the following command to delete all the documents in the collection inventory

   db.inventory.deleteMany({})

3. Use insertMany() to populate the collection

```
db.inventory.insertMany( [
  { item: "canvas", qty: 100, size: { h: 28, w: 35.5, uom: "cm" }, status: "A" },
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
  { item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uom: "cm" }, status: "A" },
  { item: "mousepad", qty: 25, size: { h: 19, w: 22.85, uom: "cm" }, status: "P" },
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "P" },
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" },
  { item: "sketchbook", qty: 80, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
  { item: "sketch pad", qty: 95, size: { h: 22.85, w: 30.5, uom: "cm" }, status: "A" }
] );
```

4. Using query operators

   db.inventory.find({ status: { $in: ["A", "D", "P"] } })

   which is equivalent to
   db.inventory.find({ $or: [{status: "A"}, {status: "D"}, {status: "P"}] })

   db.inventory.find({ qty: { $lt: 50 } }) what does this do?

5. AND/OR conditions

   db.inventory.find({ status: "A", qty: { $lt: 30 } }) what does this do?
   db.inventory.find({ $or: [{ status: "A" }, { qty: { $lt: 30 } }] }) what does this do?

Combination of AND/OR

```
db.inventory.find( {
  status: "A",
  $or: [ { qty: { $lt: 30 } }, { item: /^p/ } ]
} )
```

6. Ascending/Descending Sort

```
db.inventory.find().sort({qty:1, item: -1})
```
what does this do?

7. Query on embedded/ nested documents

```
db.inventory.find( { size: { h: 14, w: 21, uom: "cm" } } )
```

Try the following, what will happen?

```
db.inventory.find( { size: { w: 21, h: 14, uom: "cm" } } )
```

8. Query on nested field

```
db.inventory.find( { "size.uom": "in" } )
db.inventory.find( { "size.h": { $gt: 20 } } )
```
what does this do?

```
db.inventory.find( { "size.h": { $lt: 15 }, "size.uom": "in", status: "D" } )
db.inventory.find( { $or: [ {"size.h": { $lt: 15 }}, {"size.uom": "in"}, {status: "D"}]})
```

What's the difference?

9. Query an array
   a. Delete the documents in the collection first
      ```
      db.inventory.deleteMany({})
      ```

   b. Populate the collection using the following documents
      ```
      db.inventory.insertMany([
        { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
        { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
        { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
        { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
        { item: "postcard", qty: 45, tags: ["blue"], dim_cm: [ 10, 15.25 ] }
      ]);
      ```

c. Match an array

```
db.inventory.find( { tags: ["red", "blank"] } )
db.inventory.find( { tags: ["blank", "red"] } )
```

What's the difference?

How about the following two?

```
db.inventory.find( { tags: { $all: ["red", "blank"] } } )
db.inventory.find( { tags: { $all: ["blank", "red"] } } )
```

10. Query an array for an element

```
db.inventory.find( { tags: "red" } )
db.inventory.find( { dim_cm: { $gt: 25 } } )
db.inventory.find( { dim_cm: { $gt: 15 } } )
```

11. Specify multiple conditions for array elements

```
db.inventory.find( { dim_cm: { $gt: 15, $lt: 20 } } )                    what does this mean?
db.inventory.find( { dim_cm: { $elemMatch: { $gt: 15, $lt: 20 } } } )    what does this mean?
```

Remember that dim_cm is an array, try the following

```
db.inventory.find( { "dim_cm.1": { $gt: 15 } } )
db.inventory.find( { "dim_cm.0": { $gt: 15 } } )
```

What's the difference?

12. Array length

```
db.inventory.find( { tags: { $size: 3 } } )                              what does this mean?
```

13. Query for a document nested in an array
    a. Delete the documents in the collection first

```
db.inventory.deleteMany({})
```

    b. Populate the collection using the following documents

```
db.inventory.insertMany( [
  { item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] },
  { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },
  { item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 15 } ] },
  { item: "planner", instock: [ { warehouse: "A", qty: 40 }, { warehouse: "B", qty: 5 } ] },
  { item: "postcard", instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }
]);
```

Here, instock is an array of nested documents

c. Try the following commands. What's the difference?
```
db.inventory.find( { instock: { warehouse: "A", qty: 5 } } )
db.inventory.find( { instock: { qty: 5, warehouse: "A" } } )
```

d. Query on a field in an array of documents
```
db.inventory.find( { "instock.qty": { $lte: 20 } } )
db.inventory.find( { "instock.0.qty": { $lte: 20 } } )
db.inventory.find( { "instock.1.qty": { $lte: 20 } } )
```

What's the difference?

e. Specify multiple conditions on one field

```
db.inventory.find( { "instock.qty": 5, "instock.warehouse": "A" } )
db.inventory.find( { instock: { $elemMatch: { qty: 5, warehouse: "A" } } } )
```

and

```
db.inventory.find( { "instock.qty": { $gt: 10,  $lte: 20 } } )
db.inventory.find( { instock: { $elemMatch: { qty: { $gt: 10, $lte: 20 } } } } )
```

What's the difference?

14. Query for Null or Missing Fields

a. Delete the documents in the collection first
```
db.inventory.deleteMany({})
```

b. Populate the collection using the following documents
```
db.inventory.insertMany([
  { _id: 1, item: null },
  { _id: 2 }
])
```

c. Enter the following commands

```
db.inventory.find( { item: null } )                    What's the result?
db.inventory.find( { item : { $type: 10 } } )          What's the result?
db.inventory.find( { item : { $exists: false } } )     What's the result?
```

15. Project Fields to Return from Query

You may wonder how to specify fields as using SELECT in RDBMS since so far the result contains all the fields.

a. Delete the documents in the collection first
db.inventory.deleteMany({})

b. Populate the collection

```
db.inventory.insertMany( [
  { item: "journal", status: "A", size: { h: 14, w: 21, uom: "cm" }, instock: [ { warehouse: "A", qty: 5 } ] },
  { item: "notebook", status: "A",  size: { h: 8.5, w: 11, uom: "in" }, instock: [ { warehouse: "C", qty: 5 } ] },
  { item: "paper", status: "D", size: { h: 8.5, w: 11, uom: "in" }, instock: [ { warehouse: "A", qty: 60 } ] },
  { item: "planner", status: "D", size: { h: 22.85, w: 30, uom: "cm" }, instock: [ { warehouse: "A", qty: 40 } ] },
  { item: "postcard", status: "A", size: { h: 10, w: 15.25, uom: "cm" }, instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }
]);
```

c. A projection can explicitly include several fields by setting the <field> to 1 in the projection document

```
db.inventory.find( { status: "A" }, { item: 1, status: 1 } )
```

The operation corresponds to the following SQL statement:
SELECT _id, item, status from inventory WHERE status = "A"

How about the following?

```
db.inventory.find( { status: "A" }, { item: 0, status: 1 } )
```

You can remove the _id field

```
db.inventory.find( { status: "A" }, { item: 1, status: 1, _id: 0 } )
```

d. Exclude specific fields

```
db.inventory.find( { status: "A" }, { status: 0, instock: 0 } )
```

e. Return/exclude specific fields in embedded documents

```
db.inventory.find(   { status: "A" },    { item: 1, status: 1, "size.uom": 1 })
db.inventory.find(   { status: "A" },    { "size.uom": 0 })
db.inventory.find( { status: "A" },  { "instock.warehouse": 0 } )
```

f. Projection on embedded documents in an array

```
db.inventory.find( { status: "A" }, { item: 1, status: 1, instock: 1 } )
db.inventory.find( { status: "A" }, { item: 1, status: 1, "instock.warehouse": 1 } )
db.inventory.find( { status: "A" }, { item: 1, status: 1, "instock.qty": 1 } )
```

A question for you. How to list the item field?

```
db.inventory.find({ item: 1})
```

Does it work?

References

https://docs.mongodb.com/manual/introduction/
https://www.guru99.com/create-read-update-operations-mongodb.html
https://www.codeproject.com/Articles/1037052/Introduction-to-MongoDB