Read in MongoDB

Watch video: https://youtu.be/i7-yTCk8oV8?t=4791

find()

- Read operations in MongoDB use the find() method

- Example: this statement will find all documents in the bios collection

`db.bios.find()`

Adding criteria

- The following find() operation will find a document with an id of 5

```
db.bios.find( { _id: 5 } )
```

- We can also match multiple criteria, as shown here:

```
db.grades.find(
    {class_id: 19,student_id:5}
).pretty()
```

$and / $or operators

- The operators $and and $or can be used in find() queries.

- This example finds documents where the class_id is 19 **or** the student_id is 5.

```
db.grades.find(
  {$or: [
    {class_id: 19},
    {student_id:5}
  ]}
).pretty()
```

Comparison operators

- The following operation returns all the documents from the collection products where qty is greater than 25:

```
db.products.find( { qty: { $gt: 25 } } )
```

- The following operation returns documents in the bios collection where _id equals either 5 or ObjectId("507c35dd8fada716c89d0013"):

```
db.bios.find(
  {
    _id: { $in: [ 5, ObjectId("507c35dd8fada716c89d0013") ] }
  }
)
```

Comparison operators

- The operators available are:

Name	Description
$eq	Matches values that are equal to a specified value.
$gt	Matches values that are greater than a specified value.
$gte	Matches values that are greater than or equal to a specified value.
$in	Matches any of the values specified in an array.
$lt	Matches values that are less than a specified value.
$lte	Matches values that are less than or equal to a specified value.
$ne	Matches all values that are not equal to a specified value.
$nin	Matches none of the values specified in an array.

Query for ranges

- Combine comparison operators to specify ranges.

- The following operation returns documents with field between value1 and value2:

```
db.collection.find( { field: { $gt: value1, $lt: value2 } } );
```

Querying an array

Given a collection students that contains the following documents:

```
{ "_id" : 1, "score" : [ -1, 3 ] }
{ "_id" : 2, "score" : [ 1, 5 ] }
{ "_id" : 3, "score" : [ 5, 5 ] }
```

The following query:

```
db.students.find( { score: { $gt: 0, $lt: 2 } } )
```

Matches the following documents:

```
{ "_id" : 1, "score" : [ -1, 3 ] }
{ "_id" : 2, "score" : [ 1, 5 ] }
```

Querying an array of documents

- The following operation returns documents where an array (called awards) contains an embedded document element that has the fields 'award' and 'year'

```
db.bios.find(
    {
        awards: {
            $elemMatch: {
                award: "Turing Award",
                year: { $gt: 1980 }
            }
        }
    }
)
```