Querying in MongoDB

Databases and Practice- Menna Khaliel

Find() Method

- The find method is used to perform queries in MongoDB
- Quiring returned a subset of documents in a collection, from no documents at all to the entire collection
- Which documents get returned is determined by the first argument defined, **which is a document specifying the query criteria**

Method Signature

db.collection.find()

Definition

```
db.collection.find(query, projection)
```

Parameter	Type	Description
`query`	document	Optional. Specifies selection filter using query operators. To return all documents in a collection, omit this parameter or pass an empty document (`{}`).
projection	document	Optional. Specifies the fields to return in the documents that match the query filter. To return all fields in the matching documents, omit this parameter. For details, see Projection.

Find() Crash Course

```
db.collection.find(query, projection)
```

- Passing an **empty argument (document)** for query -> matches everything in the collection

`db.c.find({})` or `db.c.find()` -> matches every document in collection ‘c’ and returns them in batches

If not given any arguments, default is {}

Find() Crash Course

- Adding key/value pairs to the query argument (document) allows us to restrict our search
  - Num match num
  - Bool match bool
  - Str match str
  - As simple as specifying the value we're looking for

`db.users.find({ "age" : 27})` -> find all documents where the value for "age" is 27

`db.users.find({ "username" : "Doggie"})` -> find all documents where the "username" is "Doggie"

Find() Crash Course

```
db.collection.find(query, projection)
```

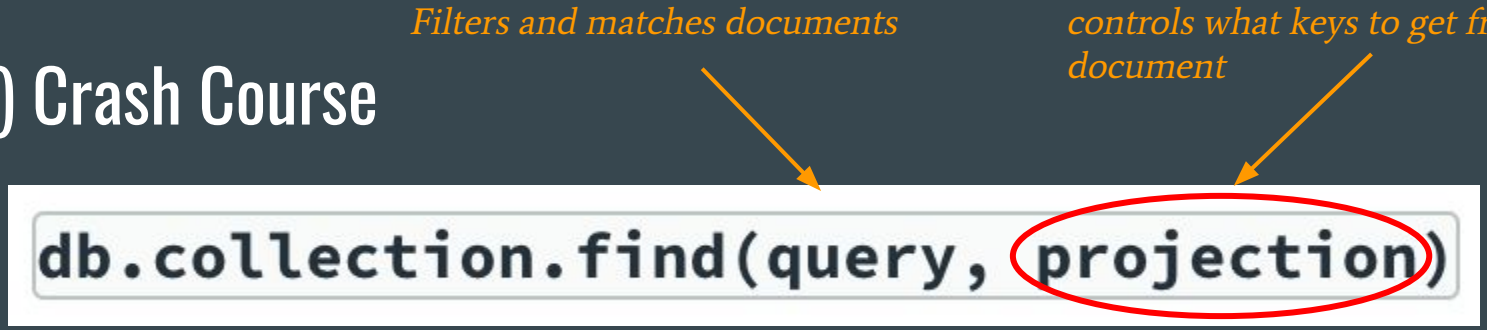
- Adding key/value pairs to the query argument (document) allows us to restrict our search
  - Multiple conditions can be strung together by adding more key/value pairs: **cond1 AND cond2 ... AND condN**

```
db.users.find({ "age" : 27, "username" : "Doggie"})
```

->

find all documents where the value for "age" is 27 with the "username" being "Doggie"

Find() Crash Course

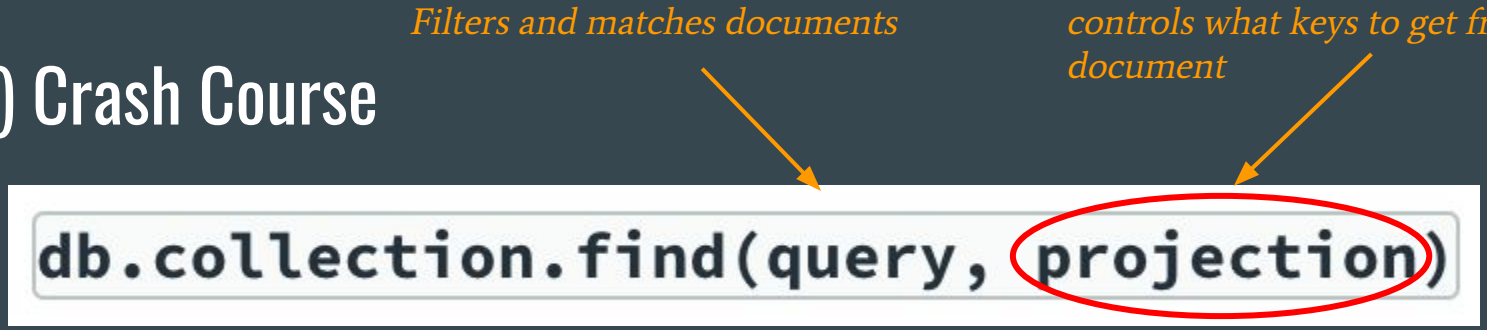


- To further specify the keys we want in the matching document, we can use the projection argument (document)

```
db.users.find({}, {"username" : 1 , "email" : 1}) ->
db.users.find({}, {"username" : 6 , "email" : 14})->
db.users.find({}, {"username" : true , "email" : true})->
```

ALL return the username and email keys only from the 'users' collection

Find() Crash Course



- To further specify the keys we want in the matching document, we can use the projection argument (document)

```
db.users.find({}, {"age" : false})->
```

ALL exclude the age key/value pair from the document returned

- *ID field is always returned by default, but can't be excluded if explicitly specified in the projection argument*

Find() Crash Course

- **Limitations**: value of query document can only be a constant, or a normal variable in your code, but can't refer to another key in the document

```
db.users.find({"age" : "cat_age"})
```
-> won't work!

Find() Crash Course: Query Conditionals

```
db.collection.find(query, projection)
```

- Queries can match more complex criteria; ranges, OR-clauses, negation
  - "$lt" <
  - "$lte" <=
  - "$gt" >
  - "$gte" >=
  - "$ne" != *can be used with any type*

```
db.users.find({"age" : {"$gte" : 18 , "$lte" : 30} }) ->
```

find all documents where the "age" field is greater than or equal to 18 AND less than or equal to 30

Find() Crash Course: Query Conditionals

```
db.collection.find(query, projection)
```

- Queries can match more complex criteria; ranges, OR-clauses, negation
  - OR clauses: can be used to
  - "$in" query for a variety of values for a single key
  - "$nin" opposite of "$in"
  - "$or" query for any of the given values across multiple keys

```
db.users.find({"age" : {"$in" : [3, "toddler"] }) ->
```

find all documents where the "age" field is equal to 3 or the "age" field equals "toddler"

Find() Crash Course: Query Conditionals

```
db.collection.find(query, projection)
```

- Queries can match more complex criteria; ranges, OR-clauses, negation
  - OR clauses: can be used to
  - "$in" query for a variety of values for a single key
  - "$nin" opposite of "$in"
  - "$or" query for any of the given values across multiple keys

```
db.users.find({"$or" : [ {"age" : 3}, {"weight" : 17}] )
```

->

"$in" can be used here as well !

find all documents where the "age" field is equal to 3 or all documents where "weight" field equals 17

THANK YOU, Questions?

References

- db.collection.find() — MongoDB Manual
- MongoDB: The Definitive Guide: Powerful and Scalable Data Storage