COMP 20 WEB PROGRAMMING

NOSQL DATABASES

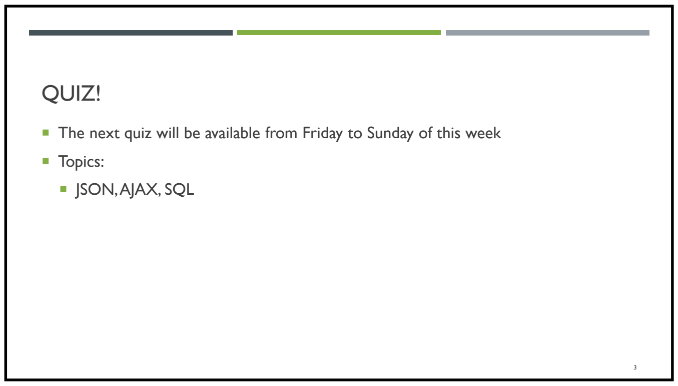
1

OFFICE HOURS

Sunday 11am
Mon 7pm
Tues 3:30pm
Thurs 8pm

Plus: 8pm on Wednesday with Lisa

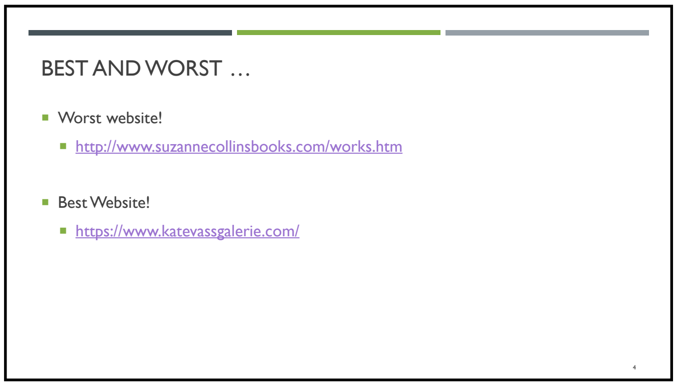
2



QUIZ!

- The next quiz will be available from Friday to Sunday of this week
- Topics:
  - JSON, AJAX, SQL

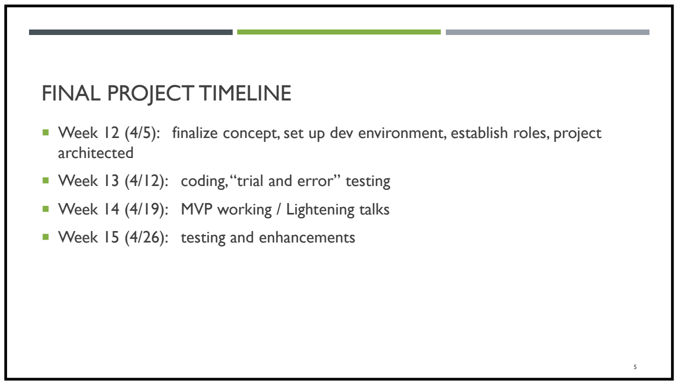
3



BEST AND WORST …

- Worst website!
  - http://www.suzannecollinsbooks.com/works.htm
- Best Website!
  - https://www.katevassgalerie.com/

4

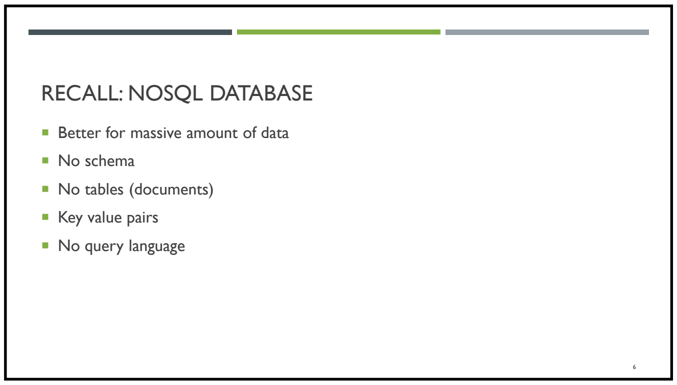


FINAL PROJECT TIMELINE

- Week 12 (4/5): finalize concept, set up dev environment, establish roles, project architected
- Week 13 (4/12): coding, "trial and error" testing
- Week 14 (4/19): MVP working / Lightening talks
- Week 15 (4/26): testing and enhancements

5

5

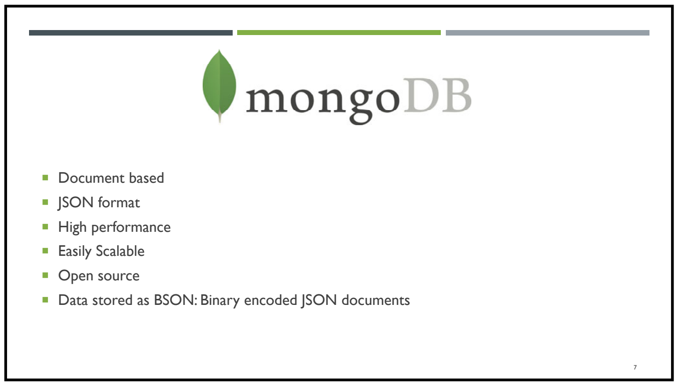


RECALL: NOSQL DATABASE

- Better for massive amount of data
- No schema
- No tables (documents)
- Key value pairs
- No query language

6

6

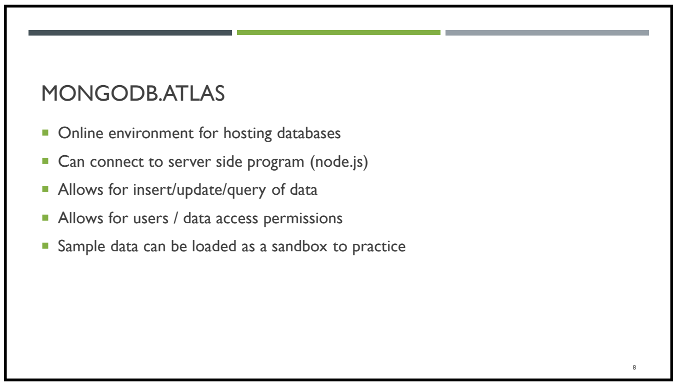


mongoDB

- Document based
- JSON format
- High performance
- Easily Scalable
- Open source
- Data stored as BSON: Binary encoded JSON documents

7

7



MONGODB.ATLAS

- Online environment for hosting databases
- Can connect to server side program (node.js)
- Allows for insert/update/query of data
- Allows for users / data access permissions
- Sample data can be loaded as a sandbox to practice

8

8

SET UP A MONGODB ATLAS ACCOUNT

- Go to https://www.mongodb.com/cloud/atlas
- Create a project
- Create a cluster
- Load sample data
- Add your own data

COLLECTIONS AND DOCUMENTS

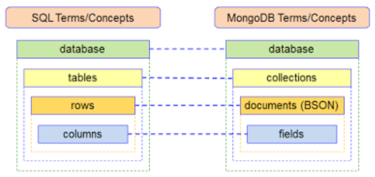


Image credit: https://www.analyticsvidhya.com/blog/2015/06/beginners-guide-mongodb/

EXAMPLE

Products
id, name, price

RDBMS: store data in a table called products

Id	Name	Price
10	Widget	3.5

MongoDB: create a collection which has only one document in this case

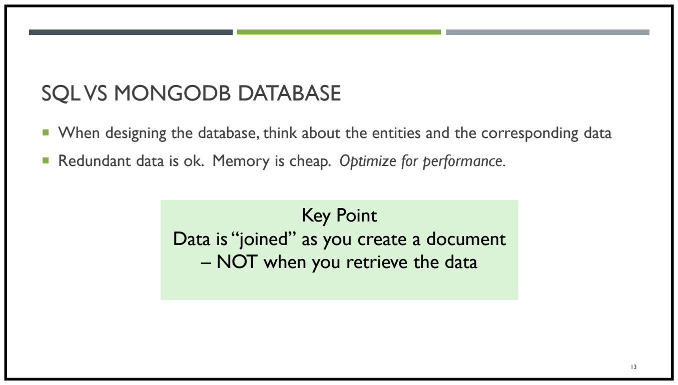
```
{
  id: 10,
  name: "Widget",
  price: 3.5
}
```

TRY IT

- Create a MongoDB collection of textbooks
- Book 1: Competitive Swimming by Bob Smith
- Book 2: Coding for Dummies by John Jones

```
[
 {
 "title": "Competitive Swimming ",
 "author": "Bob Smith"
 }

 {
 "title": "Coding for Dummies",
 "author": "John Jones"
 }
]
```



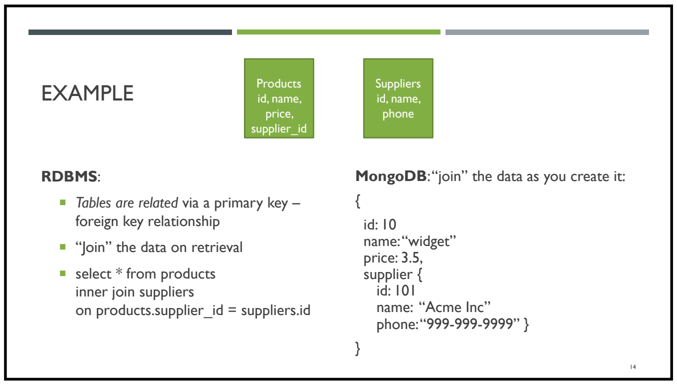
SQL VS MONGODB DATABASE

- When designing the database, think about the entities and the corresponding data
- Redundant data is ok. Memory is cheap. *Optimize for performance.*

Key Point
Data is "joined" as you create a document – NOT when you retrieve the data

13

13



EXAMPLE

Products
id, name, price, supplier_id

Suppliers
id, name, phone

RDBMS:

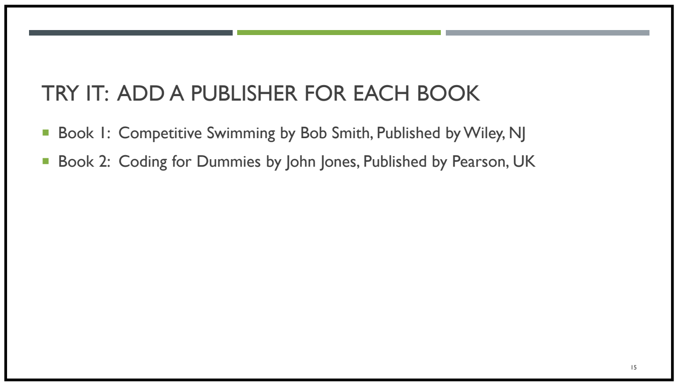
- *Tables are related* via a primary key – foreign key relationship
- "Join" the data on retrieval
- select * from products
  inner join suppliers
  on products.supplier_id = suppliers.id

MongoDB: "join" the data as you create it:

```
{
  id: 10
  name: "widget"
  price: 3.5,
  supplier {
    id: 101
    name: "Acme Inc"
    phone: "999-999-9999" }
}
```

14

14

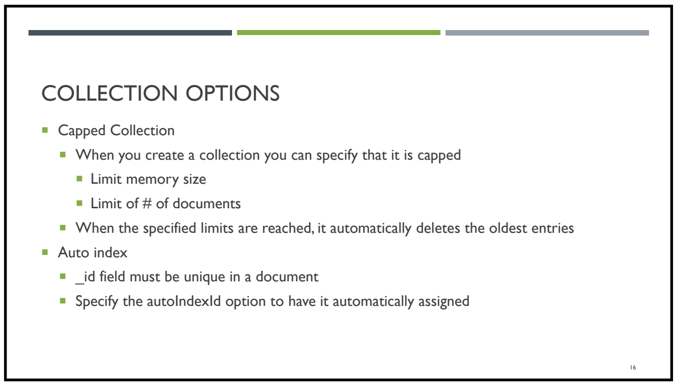


TRY IT: ADD A PUBLISHER FOR EACH BOOK

- Book 1: Competitive Swimming by Bob Smith, Published by Wiley, NJ
- Book 2: Coding for Dummies by John Jones, Published by Pearson, UK

15

15

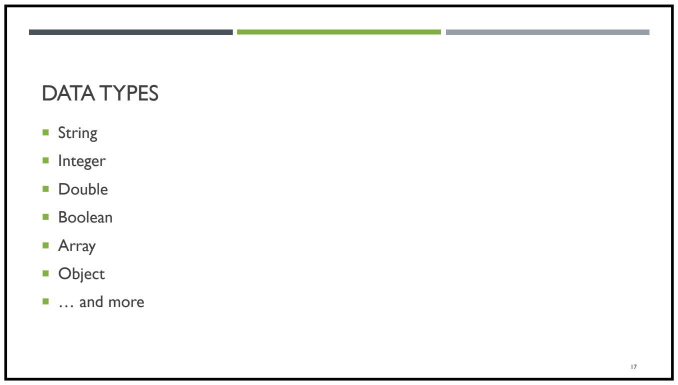


COLLECTION OPTIONS

- Capped Collection
  - When you create a collection you can specify that it is capped
    - Limit memory size
    - Limit of # of documents
  - When the specified limits are reached, it automatically deletes the oldest entries
- Auto index
  - _id field must be unique in a document
  - Specify the autoIndexId option to have it automatically assigned

16

16

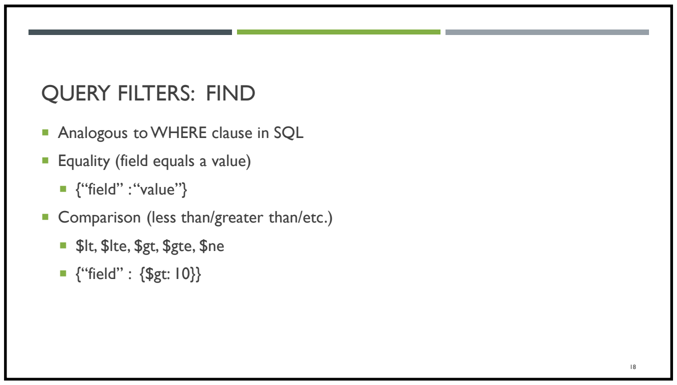


DATA TYPES

- String
- Integer
- Double
- Boolean
- Array
- Object
- … and more

17

17

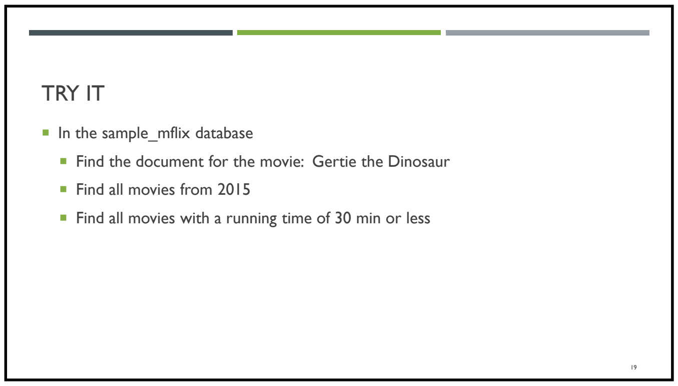


QUERY FILTERS: FIND

- Analogous to WHERE clause in SQL
- Equality (field equals a value)
  - {"field" :"value"}
- Comparison (less than/greater than/etc.)
  - $lt, $lte, $gt, $gte, $ne
  - {"field" : {$gt: 10}}

18

18

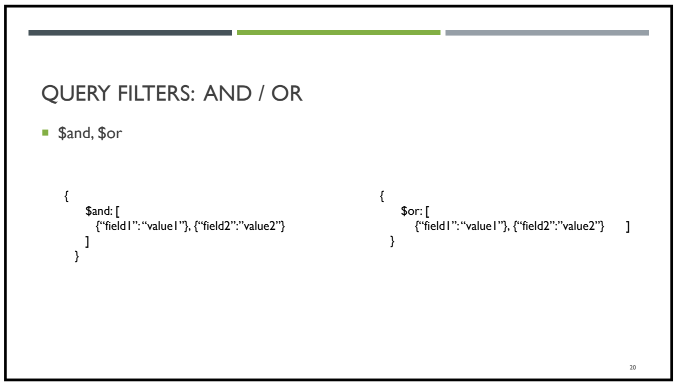


TRY IT

- In the sample_mflix database
  - Find the document for the movie: Gertie the Dinosaur
  - Find all movies from 2015
  - Find all movies with a running time of 30 min or less

19

19



QUERY FILTERS: AND / OR

- $and, $or

```
{
     $and: [
        {"field1":"value1"}, {"field2":"value2"}
     ]
  }
```

```
{
     $or: [
        {"field1":"value1"}, {"field2":"value2"}      ]
   }
```

20

20

TRY IT

- In the sample_supplies database
  - Find sales that occurred in Austin or Denver
  - Find all online sales from London

21

21

"LIKE" QUERIES

- To find a patterned value, use a regular expression:
- {"field": /val/} … find anywhere
- {"field": /^val/} … starts with

22

22

TRY IT

- In the sample_mflix database,
  - Find movies that start with the letter "L"
  - Find movies with the word "men" in the plot

23

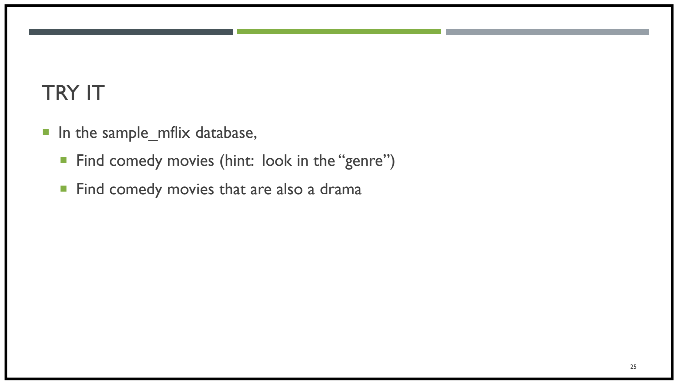
23

QUERYING ARRAYS

- Check if the array contains at least one document with a value
  - {"field":"val"}
- Find documents with the exact array structure
  - { "field": ["val1", "val2"] }
  - { "pet_name": ["fido", "fifi"] }
- Find documents that match all specified items: $all
  - Order does not matter
  - There can be other items in the array as well
  - { "field": { $all: ["val1", "val2"] } }

24

24

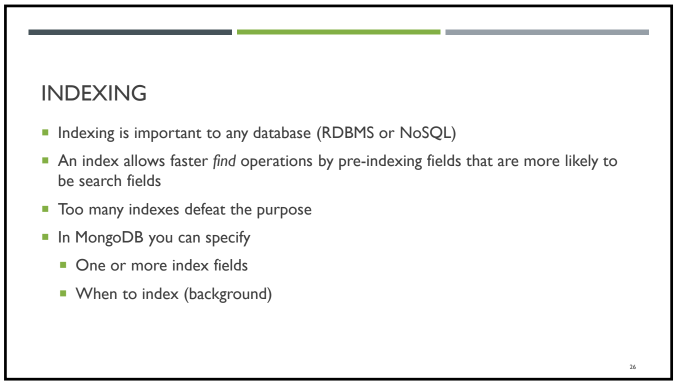


TRY IT

- In the sample_mflix database,
  - Find comedy movies (hint: look in the "genre")
  - Find comedy movies that are also a drama

25

25

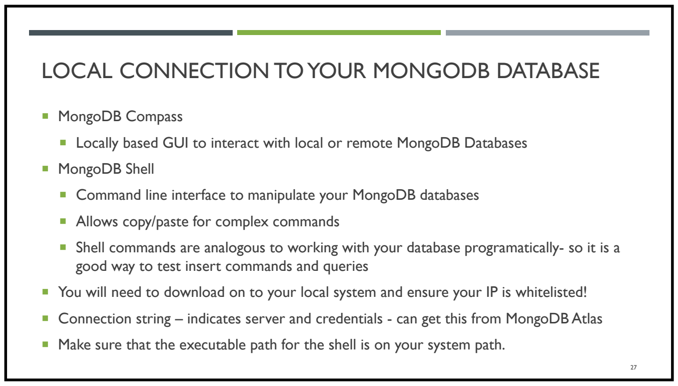


INDEXING

- Indexing is important to any database (RDBMS or NoSQL)
- An index allows faster *find* operations by pre-indexing fields that are more likely to be search fields
- Too many indexes defeat the purpose
- In MongoDB you can specify
  - One or more index fields
  - When to index (background)

26

26

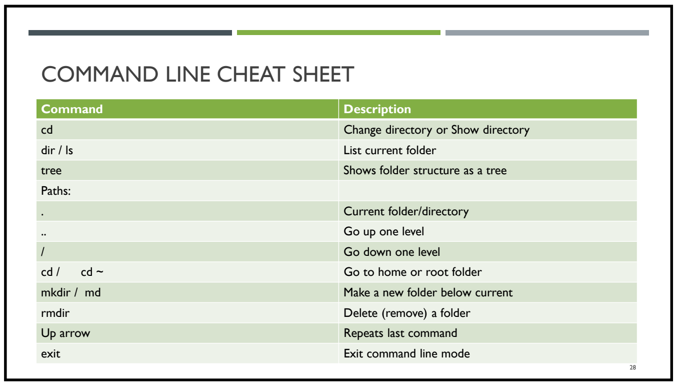


LOCAL CONNECTION TO YOUR MONGODB DATABASE

- MongoDB Compass
  - Locally based GUI to interact with local or remote MongoDB Databases
- MongoDB Shell
  - Command line interface to manipulate your MongoDB databases
  - Allows copy/paste for complex commands
  - Shell commands are analogous to working with your database programatically- so it is a good way to test insert commands and queries
- You will need to download on to your local system and ensure your IP is whitelisted!
- Connection string – indicates server and credentials - can get this from MongoDB Atlas
- Make sure that the executable path for the shell is on your system path.

27

27

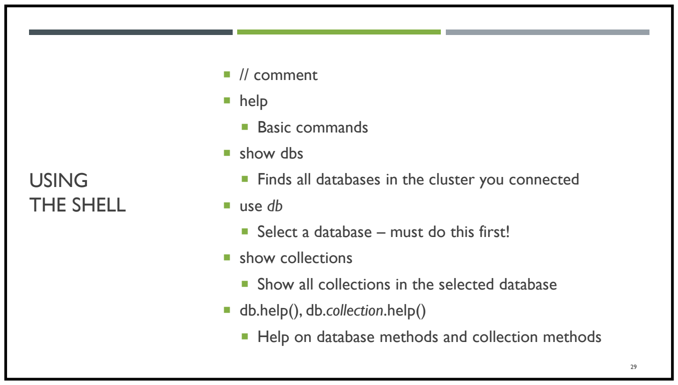


COMMAND LINE CHEAT SHEET

Command	Description
cd	Change directory or Show directory
dir / ls	List current folder
tree	Shows folder structure as a tree
Paths:	
.	Current folder/directory
..	Go up one level
/	Go down one level
cd / cd ~	Go to home or root folder
mkdir / md	Make a new folder below current
rmdir	Delete (remove) a folder
Up arrow	Repeats last command
exit	Exit command line mode

28

28

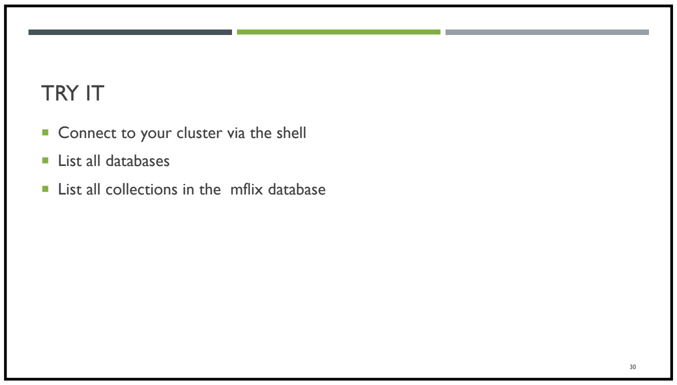


USING THE SHELL

- // comment
- help
  - Basic commands
- show dbs
  - Finds all databases in the cluster you connected
- use *db*
  - Select a database – must do this first!
- show collections
  - Show all collections in the selected database
- db.help(), db.*collection*.help()
  - Help on database methods and collection methods

29

29

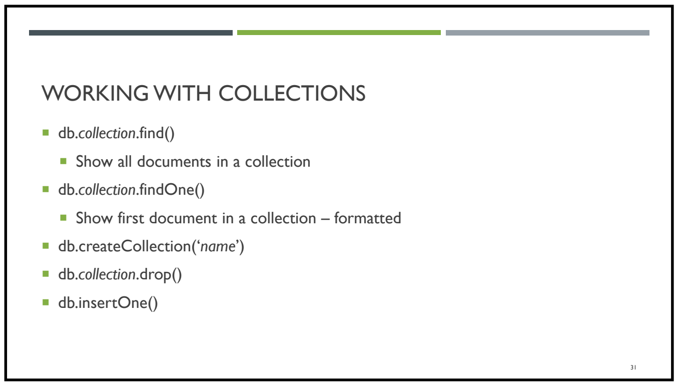


TRY IT

- Connect to your cluster via the shell
- List all databases
- List all collections in the mflix database

30

30

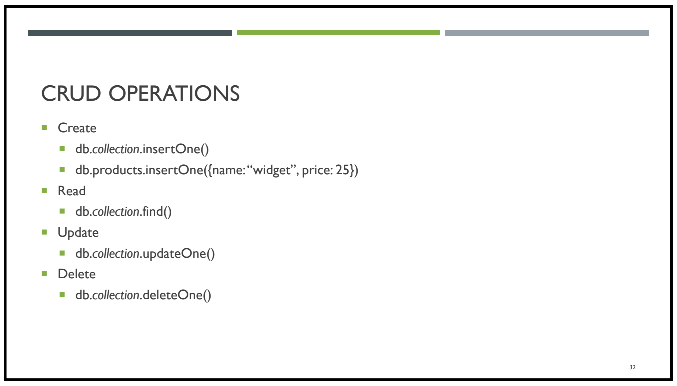


WORKING WITH COLLECTIONS

- db.*collection*.find()
  - Show all documents in a collection
- db.*collection*.findOne()
  - Show first document in a collection – formatted
- db.createCollection('*name*')
- db.*collection*.drop()
- db.insertOne()

31

31

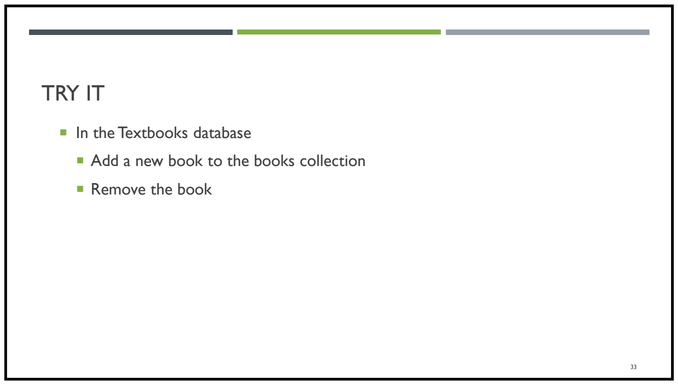


CRUD OPERATIONS

- Create
  - db.*collection*.insertOne()
  - db.products.insertOne({name: "widget", price: 25})
- Read
  - db.*collection*.find()
- Update
  - db.*collection*.updateOne()
- Delete
  - db.*collection*.deleteOne()

32

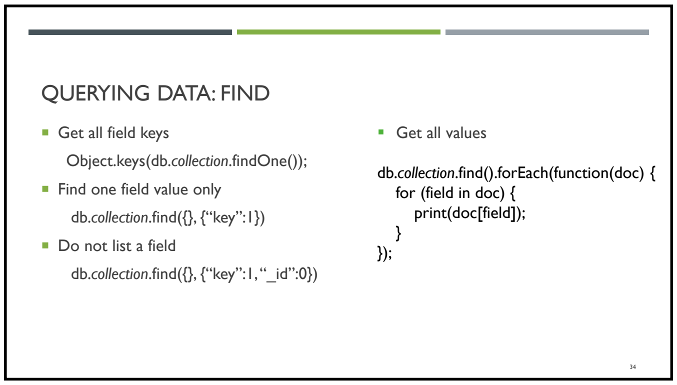
32



TRY IT

- In the Textbooks database
  - Add a new book to the books collection
  - Remove the book

33



QUERYING DATA: FIND

- Get all field keys

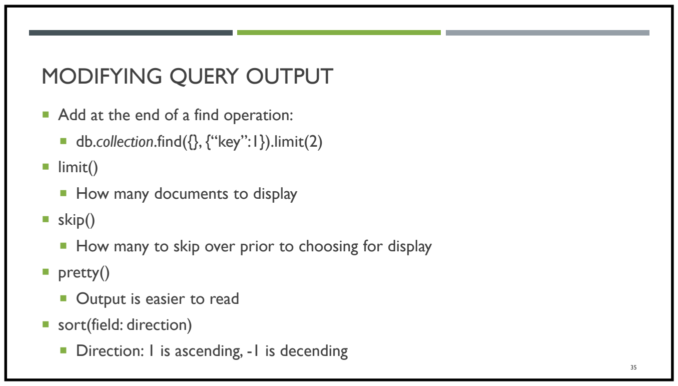
  Object.keys(db.*collection*.findOne());
- Find one field value only

  db.*collection*.find({}, {"key":1})
- Do not list a field

  db.*collection*.find({}, {"key":1, "_id":0})
- Get all values

```
db.collection.find().forEach(function(doc) {
   for (field in doc) {
       print(doc[field]);
   }
});
```

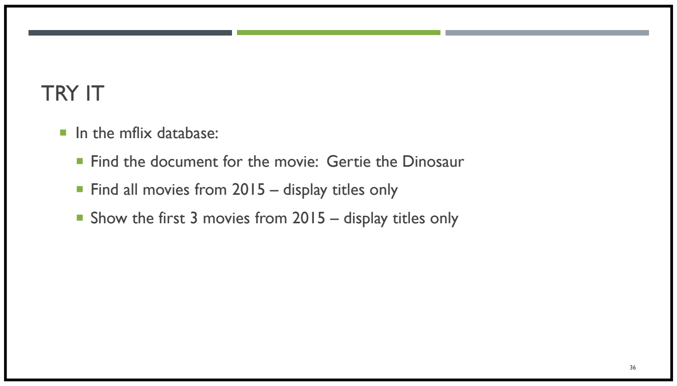
34



MODIFYING QUERY OUTPUT

- Add at the end of a find operation:
  - db.*collection*.find({}, {"key":1}).limit(2)
- limit()
  - How many documents to display
- skip()
  - How many to skip over prior to choosing for display
- pretty()
  - Output is easier to read
- sort(field: direction)
  - Direction: 1 is ascending, -1 is decending

35



TRY IT

- In the mflix database:
  - Find the document for the movie: Gertie the Dinosaur
  - Find all movies from 2015 – display titles only
  - Show the first 3 movies from 2015 – display titles only

36