Mongo DB

Steve Iannaccone • 05.02.2017

Overview

Objectives

- Understand differences of Mongo DB
- Install Mongo

- Run queries with pyMongo

Who likes good news? Everyone?

The Good News

There is NoSQL in this lecture

The Bad News

NoSQL stands for “*Not Only SQL*”

NoSQL

How is this Different?

Traditional RDBMS

- Requires a Schema to map relations between all the tables within your DB

ID	Date_Joined	Has_Beeper	Fax_Number
003241	1987-01-16	True	(555)111-2222
...	...	...	...

Restructuring or updating this sucks...

How is this Different?

Traditional RDBMS

- Scales vertically so making queries and joins can be done quickly
- This usually requires that all the data live on the same server
- Doesn’t *“shard”* well

How is this Different?

NoSQL

- Doesn’t need to know what is being stored in advance
- No schema required
- Auto-sharding
- Easily replicated across many servers or the cloud

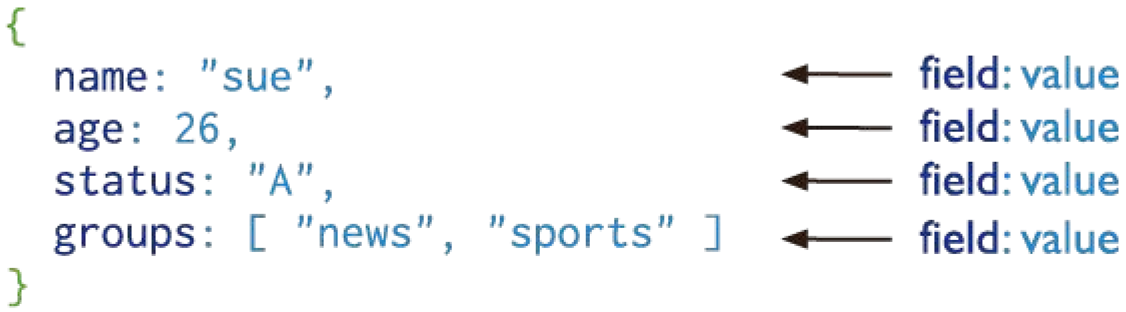
MongoDB

A flavor of NoSQL *(like Postgres is a flavor of SQL)*

MongoDB

Records = Documents

Records are stored as *“documents”* with **`field:value`** pairs



```
{
   name: "sue",
   age: 26,
   status: "A",
   groups: [ "news", "sports" ]
}
```

MongoDB

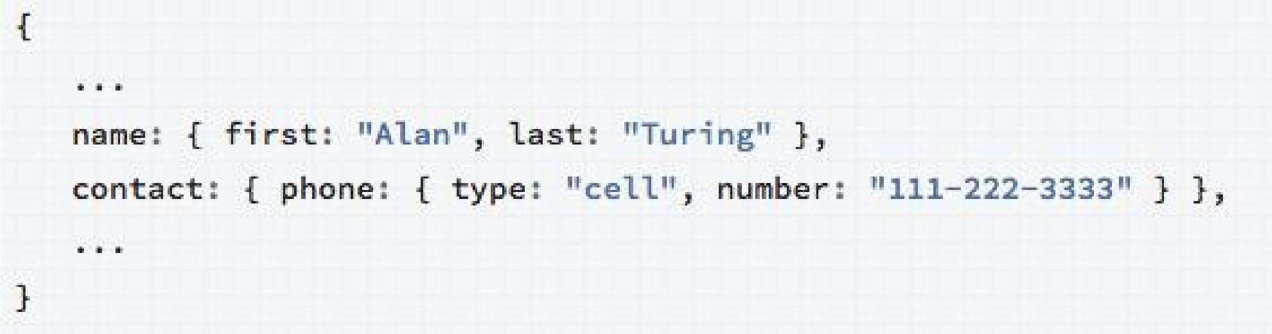
Field Names

- Must be strings
- Can’t start with the dollar sign (`$`) character
- Can’t contain the dot (`.`) character
- Can’t contain the **`null`** character
- **`_id`** is reserved for use as a primary key

MongoDB

Values

- Can be heterogenous mixture of other documents, arrays, and arrays of documents

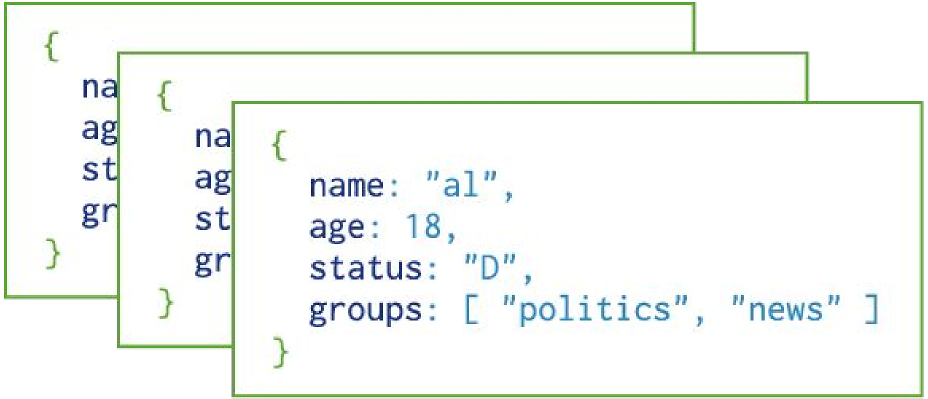


```
{
    ...
    name: { first: "Alan", last: "Turing" },
    contact: { phone: { type: "cell", number: "111-222-3333" } },
    ...
}
```

MongoDB

Tables = Collections

Databases are sets of collections of documents



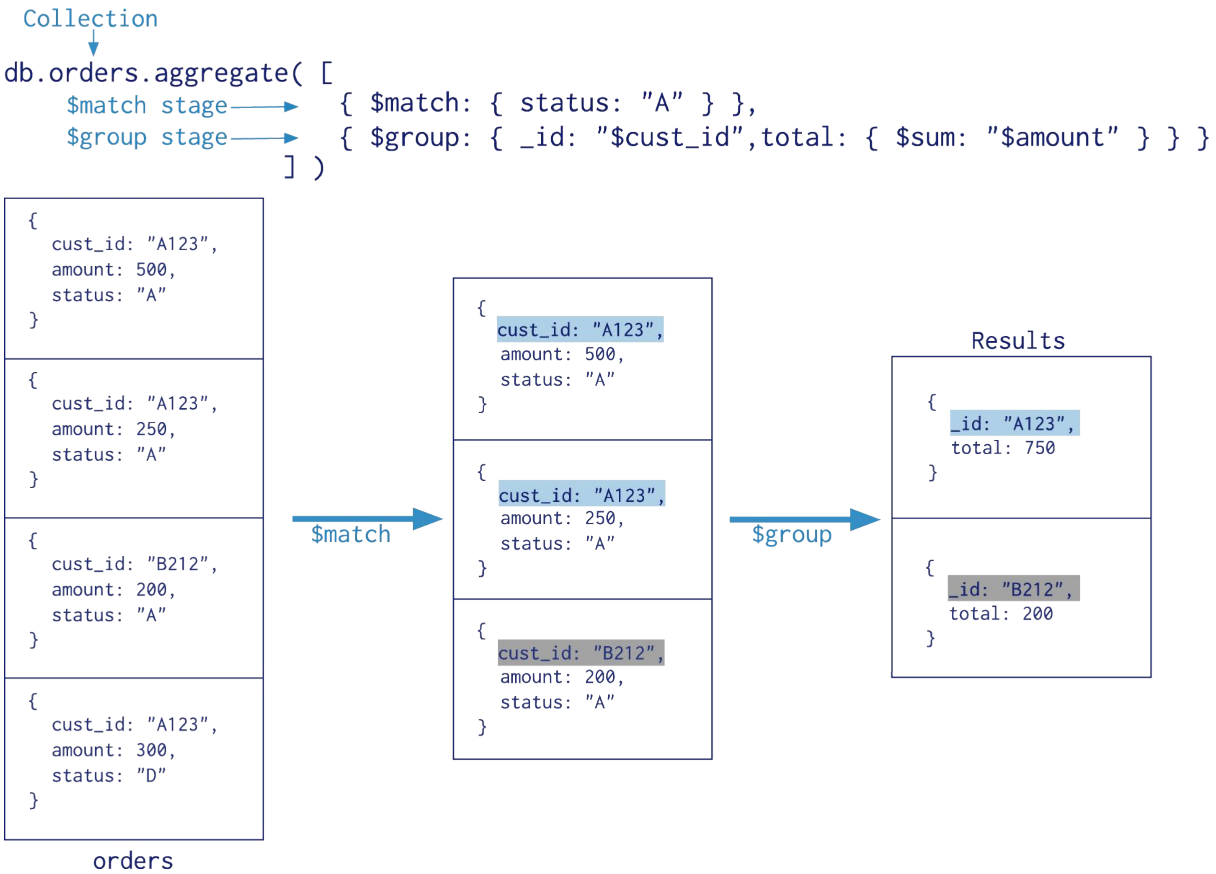
```
{
   name: "al",
   age: 18,
   status: "D",
   groups: [ "politics", "news" ]
}
```

Collection

MongoDB

Queries

SQL	Mongo
`SELECT * FROM users`	`db.users.find()`
`SELECT * FROM users WHERE age = 33 ORDER BY name ASC`	`db.users.find({age: 33}).sort({name: 1})`
`SELECT COUNT(*) FROM users WHERE age > 30`	`db.users.find({age: {$gt: 30}}).count()`



```
db.orders.aggregate( [
    { $match: { status: "A" } },
    { $group: { _id: "$cust_id",total: { $sum: "$amount" } } }
] )
```

orders

```
{
  cust_id: "A123",
  amount: 500,
  status: "A"
}
{
  cust_id: "A123",
  amount: 250,
  status: "A"
}
{
  cust_id: "B212",
  amount: 200,
  status: "A"
}
{
  cust_id: "A123",
  amount: 300,
  status: "D"
}
```

$match

```
{
  cust_id: "A123",
  amount: 500,
  status: "A"
}
{
  cust_id: "A123",
  amount: 250,
  status: "A"
}
{
  cust_id: "B212",
  amount: 200,
  status: "A"
}
```

$group

Results

```
{
  _id: "A123",
  total: 750
}
{
  _id: "B212",
  total: 200
}
```

MongoDB

Aggregate

SQL	Mongo
SELECT age, SUM(1) AS counter FROM users WHERE country = “US” GROUP BY age	db.users.aggregate([{$match: {country: ‘US’} }, {$group: {‘_id’: ‘$age’, counter: {$sum:1}} }])

Break-Out Question

With a partner, on the whiteboard tables write out a SQL and a Mongo query to do the following:

- From a table / collection called “log” return the total number of records where the “country” is “India” grouped by “city” and rename it “counter”.

PyMongo

PyMongo

We can also interact with Mongo databases directly in Python by using the right drivers.

```
$ conda install pymongo
```

PyMongo

Now, in python we can just call:

```
from pymongo import MongoClient

client = MongoClient()

database = client[‘your_db’]

coll = database[‘your_collection’]
```

(after we start our Mongo server of course)

PyMongo

Queries look similar to those done inside Mongo Shell:

```
res = coll.find({‘name’: {$ne: ‘Frank’}})
```

This returns our query cursor object as a generator we can use in the normal fashion in python *(e.g. using* `.next()`*, etc)*.

Recap

1. What are the main differences between SQL and NoSQL?
2. What flavor of NoSQL are we using in class?
3. What data structure in python do NoSQL documents most resemble?

Resources

MongoDB Downloads:

https://www.mongodb.com/download-center#community

Mongo Docs:

https://docs.mongodb.com/getting-started/python/introduction/

PyMongo:

https://api.mongodb.com/python/current/installation.html