MongoDB:
A high level introduction and demonstration.

- Features
- Strengths
- Gotchas
- Demonstration
- Would you like to know more?

Bryan Nehl
@k0emt

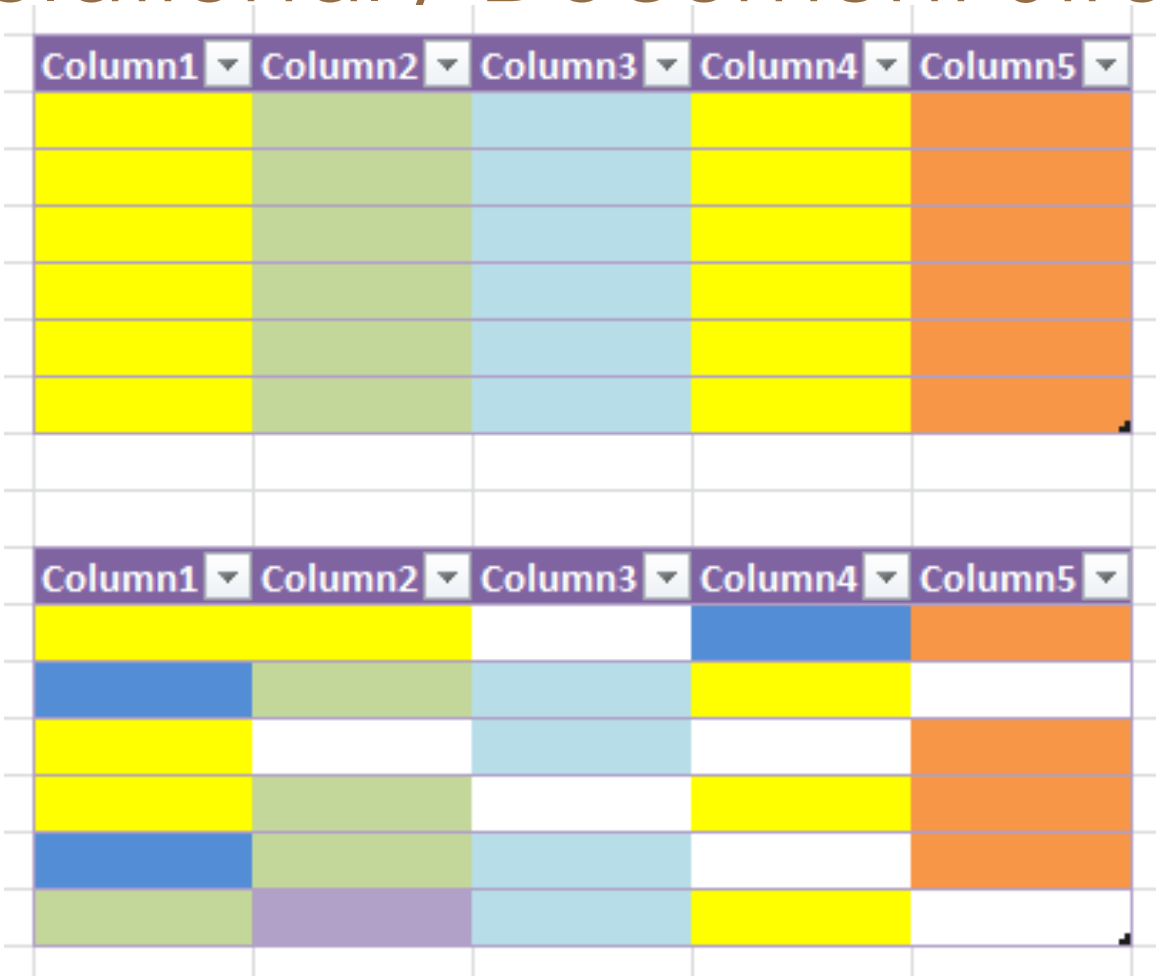
What is MongoDB?

- Document Store
  - Jagged structured documents
  - BSON
- Considered NOSQL
- Web Scale

Structure

- Databases
- Collections
- Documents
- Fields

Relational / Document Structure



Features

JSON Review

- **json.org**
- {“key” : “values”}
- JSON types
  - string, number, object, array, true, false, null
- Lists
- Sub-documents

Features

- JSON/BSON (see bsonspec.org)
- JavaScript shell
- GridFS
- File System Style backups

Features

- Supports indexes
- Features for optimizing (perf/stat/top)
- Tools for monitoring (MMS)
- GeoJSON
- Full Text Search (default on v2.6)

Aggregation Framework

- Often compared/contrasted with:
  - Map-Reduce (lightweight alternative)
  - SQL (where, order by, group by)
  - Pipe line architecture
- In version 2.6:
  - returns a cursor
  - Can output to a collection
  - Can do disk based sorting

Scalable

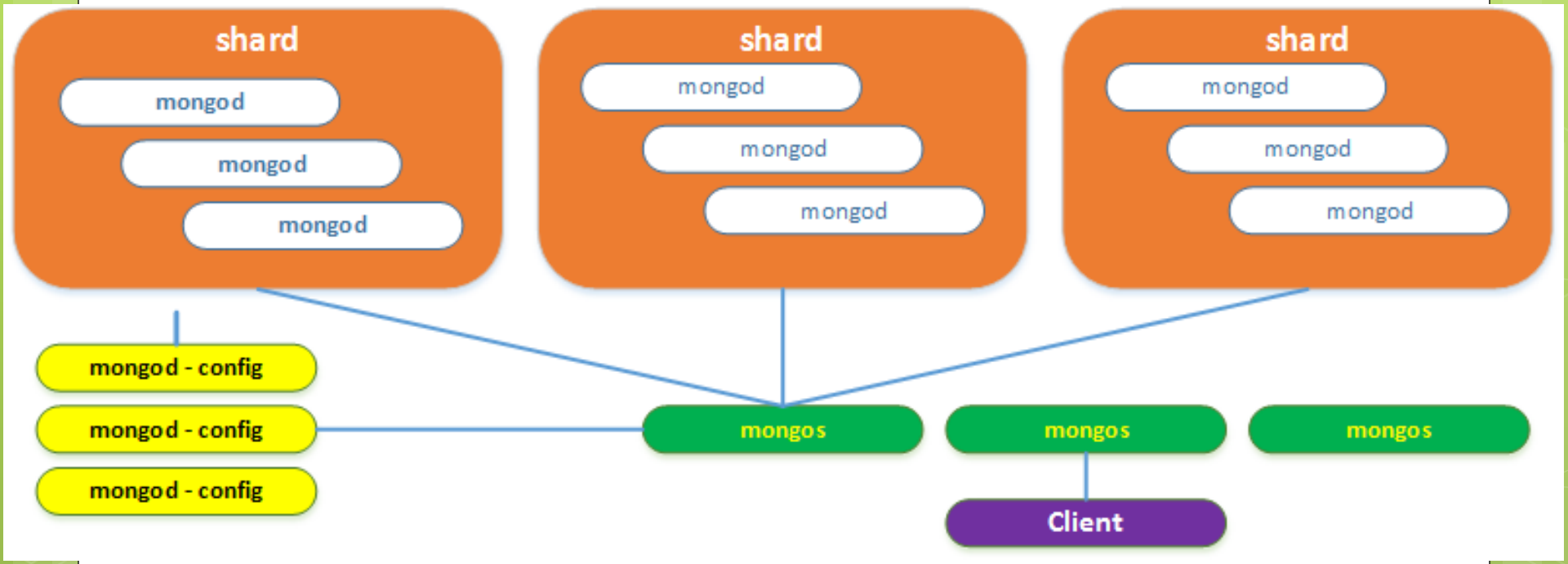
- Safety
  - Write safety (the w option)
  - Journal protection (the j option)
- Replication
- Sharding
- Commodity **PHYSICAL** machines with local storage

Sharding

- **Sharding distributes a single logical database system across a cluster of machines**
- **Shards**
  - Store a portion of the collection – size scalability
  - Balance read/write load and data across machines
  - Enabled per database and collection
- mongo**s**
  - used to access the shards
  - Utilize config servers which have metadata
    - About the cluster
    - About where the chunks are for the shards

http://docs.mongodb.org/manual/sharding/

Production Sharding Environment



Security

- Layered – start with your network
- Authentication
  - Challenge – Response
  - LDAP, Kerberos
  - Certificate
- Authorization
  - Role based access for users

http://docs.mongodb.org/manual/security/

Lots of Drivers

- Java / Groovy / JVM
- JavaScript / Node.js
- Python
- C / C++ / C#
- Go / Erlang
- Perl / PHP
- Ruby / Scala

http://docs.mongodb.org/ecosystem/drivers/

Example Java Code

```
MongoClient mongoClient = new MongoClient("localhost");
DB stuffDb = mongoClient.getDB("workshop");
DBCollection exampleCol = stuffDb.getCollection("examples");
DBObject document;

document = exampleCol.findOne(new BasicDBObject("someField",
5));
System.out.println(document);

DBCursor cursor = exampleCol.find();
while(cursor.hasNext()) {
	document = cursor.next();
	System.out.println("c doc: " + document);
}

mongoClient.close();
```

Example Python code

```
from pymongo import MongoClient

client = MongoClient("localhost")
db = client.workshop
col = db.examples

doc = col.find_one({"someField": 5})
print doc

for doc in col.find():
    print "c doc: " + str(doc)

client.close()
```

Strengths

Simplicity

- Installation
  - Download (archive or MSI)
  - Copy into place
  - Create `/data/db`
  - Start `mongod`
  - Interact with it using `mongo`

Interactive Shell

- Very powerful and easy to use
- Example
  - List databases
  - Create a document
    - Auto-creates database
  - Fields are not fixed across documents, in order, *type* or occurrence

Interactive Shell -- Demo

```
show dbs
use mydb
show dbs
show collections
db.inventory.insert({“car”:”Escape”})
db.inventory.find()
show dbs
show collections
db.inventory.insert({“car”:
{“make”:”Ford”,”model”:”Escape”}})
```

Gotchas

Gotchas

- If you misspell…
  - Database / collection / field
- Production defaults for Windows/Linux
  - Remember to configure your dev box!
- GUI interfaces are still immature
- Joins have to be done in code
- No Transactions (individual operations are atomic)
- Cut-n-paste those dang fancy “quotes”

Gotchas

- Give the shell something syntactically wrong and it'll eat it
- Isn't as strict about deleting multiple documents as it is about updating them
- Improperly repeat a field name in a query and it'll use the last criteria specified
- Differences between regular queries and the aggregation framework

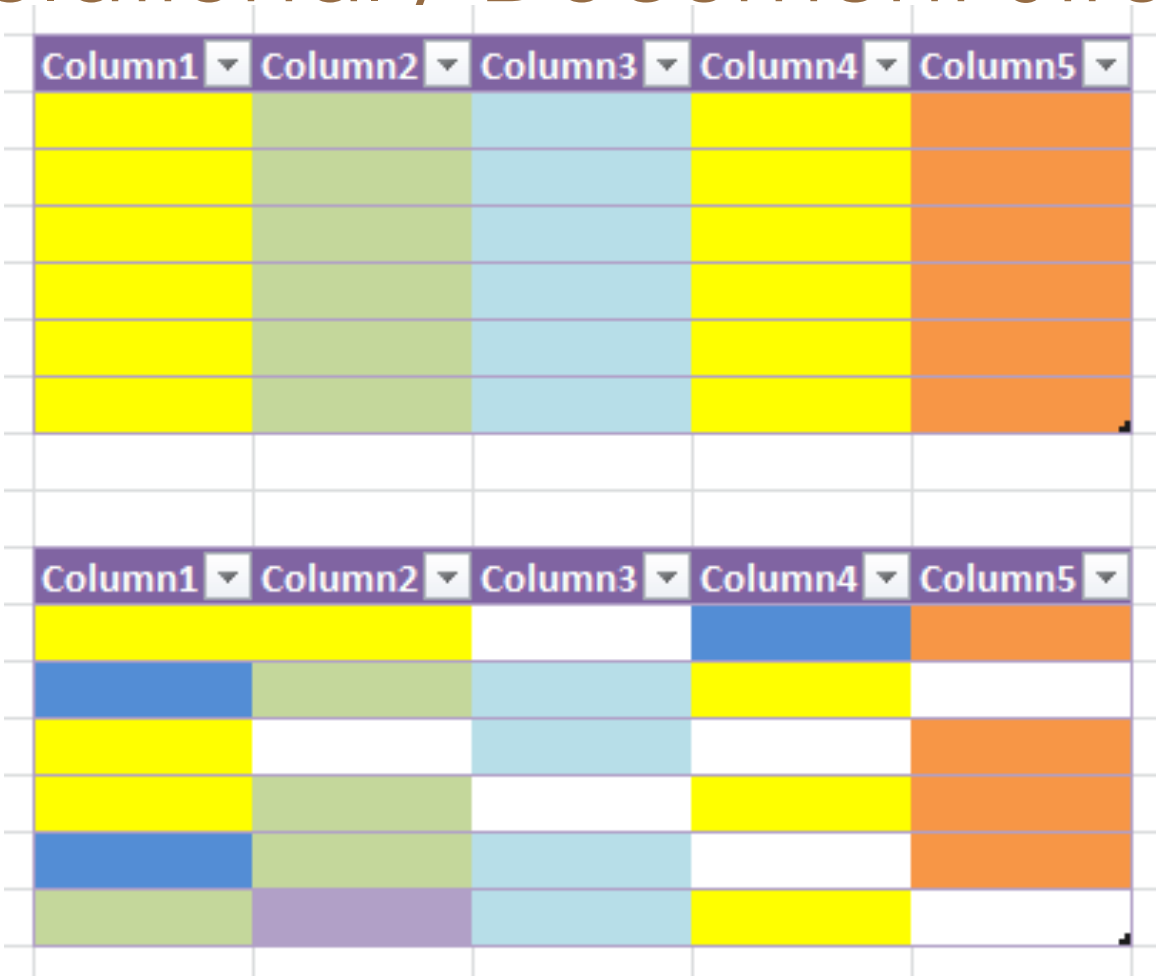
{“section” : “Schema”}

A demonstration of document oriented design

Structure

- Databases
- Collections
- Documents
- Fields

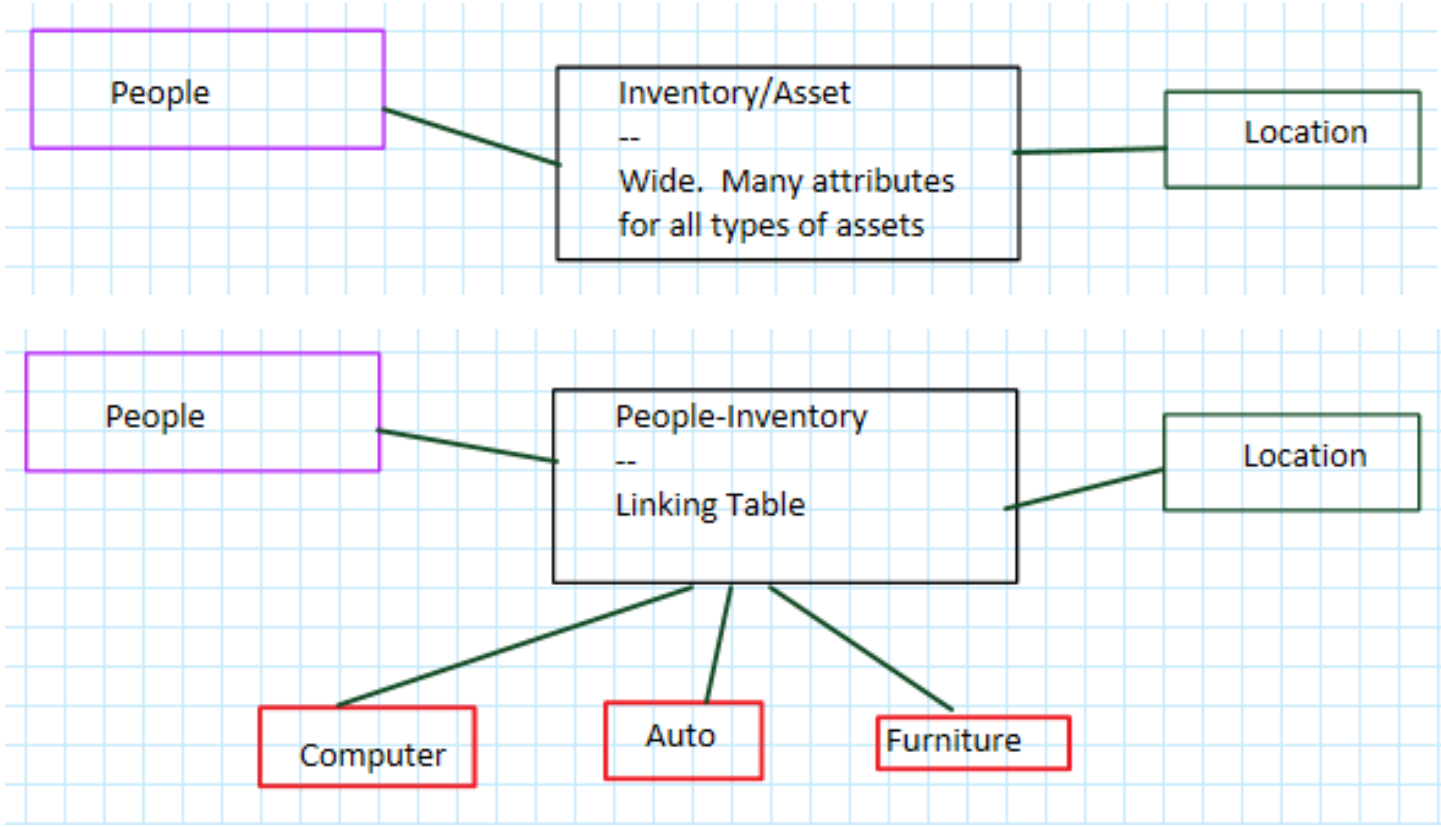
Relational / Document Structure



Relational Design Exercise

- Using standard relational techniques design an inventory management system that tracks assets.
- Example assets are: vehicles, computers, tables and chairs.
- I want to be able to store a lot of detail.
  - Where is the asset?
  - To whom is an asset assigned?
  - Vehicle detail like: make, model, VIN, color, etc.
  - Table detail like: material type, size, condition, color, etc.

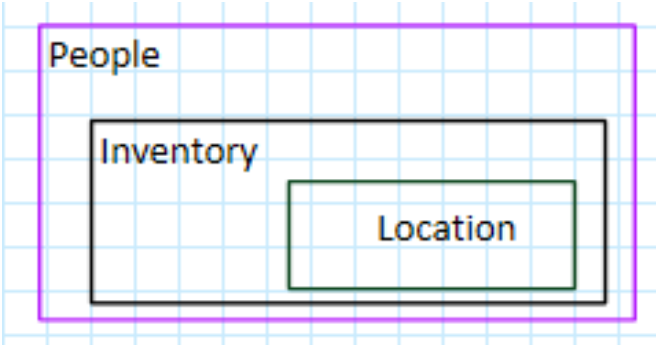
Relational

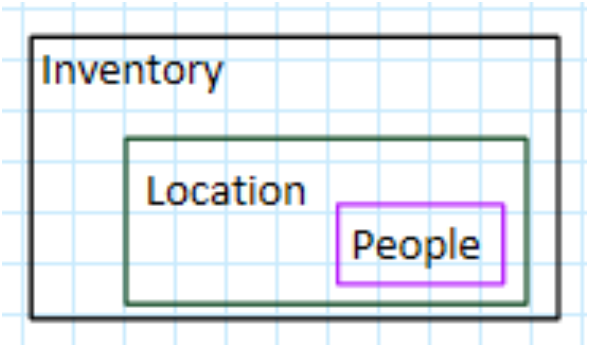

People
Inventory/Asset
--
Wide. Many attributes
for all types of assets
Location
People
People-Inventory
--
Linking Table
Location
Computer
Auto
Furniture

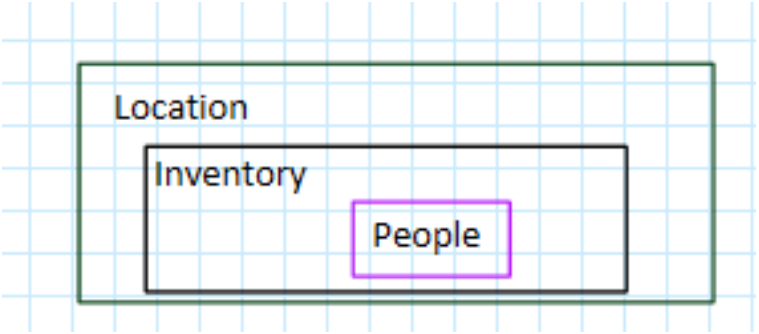
Document Design

- How would we organize this same sort of information in a document oriented system like MongoDB?
- ***Consider the access pattern(s).***
  - *Inventory part of Personnel System*
  - *Inventory part of Location Review*
  - *Inventory the primary focus*

Potential Document Design







Document Oriented Schema Design

- Naming
  - avoid the . (dot)
  - key name length matters
- No Joins
- **Consider the Access Pattern**

Would you like to know more?

Who is using MongoDB?

Lots of people

- Disney
- Expedia
- LinkedIn
- eHarmony
- Sourceforge
- SAP
- MTV

http://www.mongodb.org/about/production-deployments/

MongoDB Resources

- MongoDB University
  - Python, Java, node.js
  - Administration & Operations

https://university.mongodb.com/

http://docs.mongodb.org/manual/

http://mongodb.org

Conferences

- **MongoDB World**
- Strata
  - PyData & PyCon
  - Kansas City Developer Conference
  - BigDataSummitKC.org

Mentors

- From MongoDB
- User Groups and Forums
- Within your company

Experiment

- Set up a development environment
- Try out stuff
  - Work related
  - Something you are passionate about
- Share your experiences
  - blog, tweet, present
  - GitHub and Gists

MongoDB Questions?

Bryan Nehl

@k0emt

dbBear.com

(Links to Twitter, blog, GitHub, gists)