OPEN SOURCE DEPARTMENT

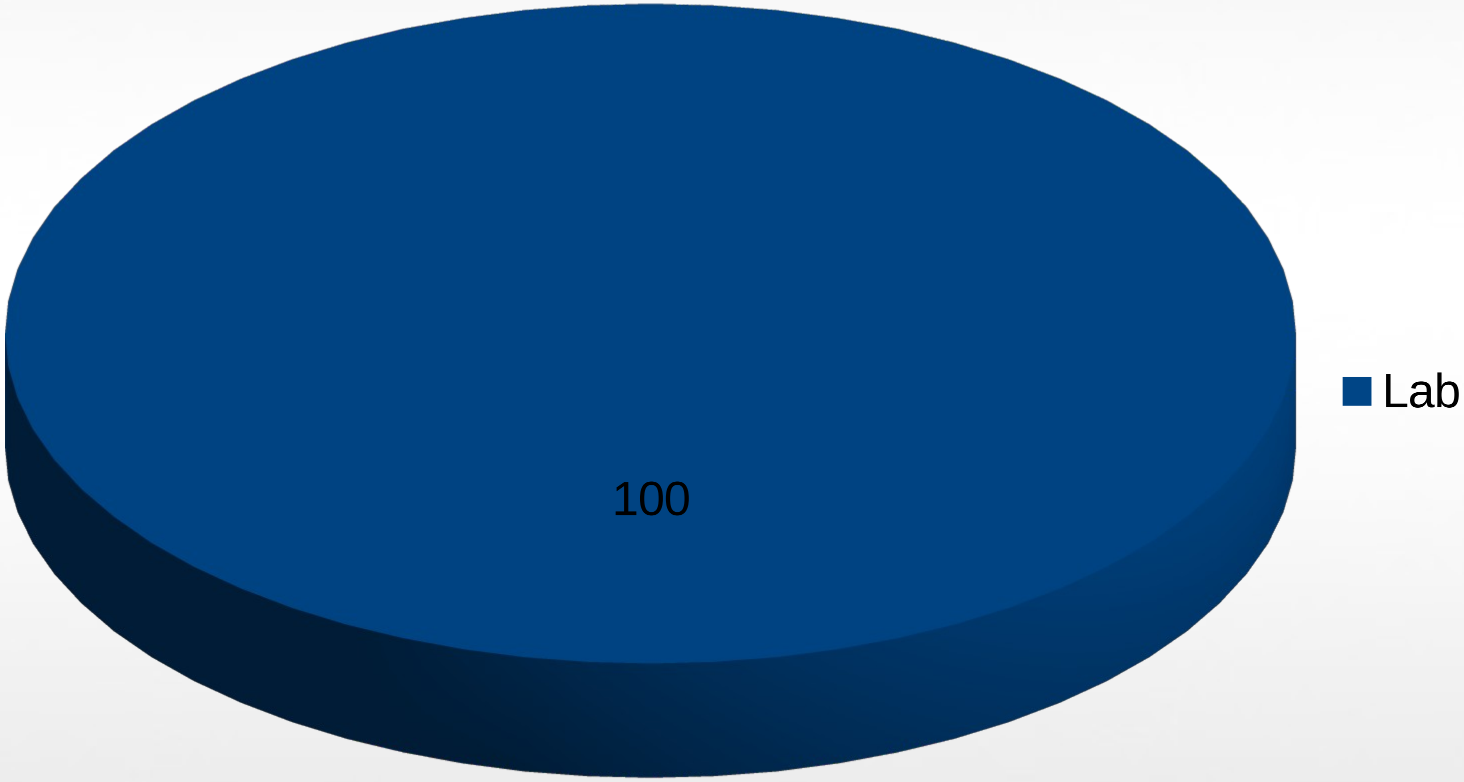
Mongo Database

Scale the web

Course Objectives

- Learn how to use Mongo database with the main operations and how to deal with the mongo shell.

Course Evaluation



Course Prerequisites

- Database Fundamentals
- Familiarity with Linux terminal
- Basic Knowledge of Javascript

Course Material

You can access course material via this URL:

http://tinyurl.com/iti-mongodb

Agenda

- What is NoSQL ?
- Why NoSQL?
- Why MongoDB ?
- Getting Started
- Mongo Basics

What is NoSQL

What is NoSQL

Definitions:

- **NoSQL** is a movement promoting a loosely defined class of non-relational data stores that break with a long history of relational databases.
- Non-relational next generation operational data-stores and databases.
- Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontal scalable.
- NoSQL is a term coined by **Carlo Strozzi** and repurposed by **Eric Evans** to refer to “some” storage systems.

The term should be used as

NOT ONLY SQL

What is NoSQL

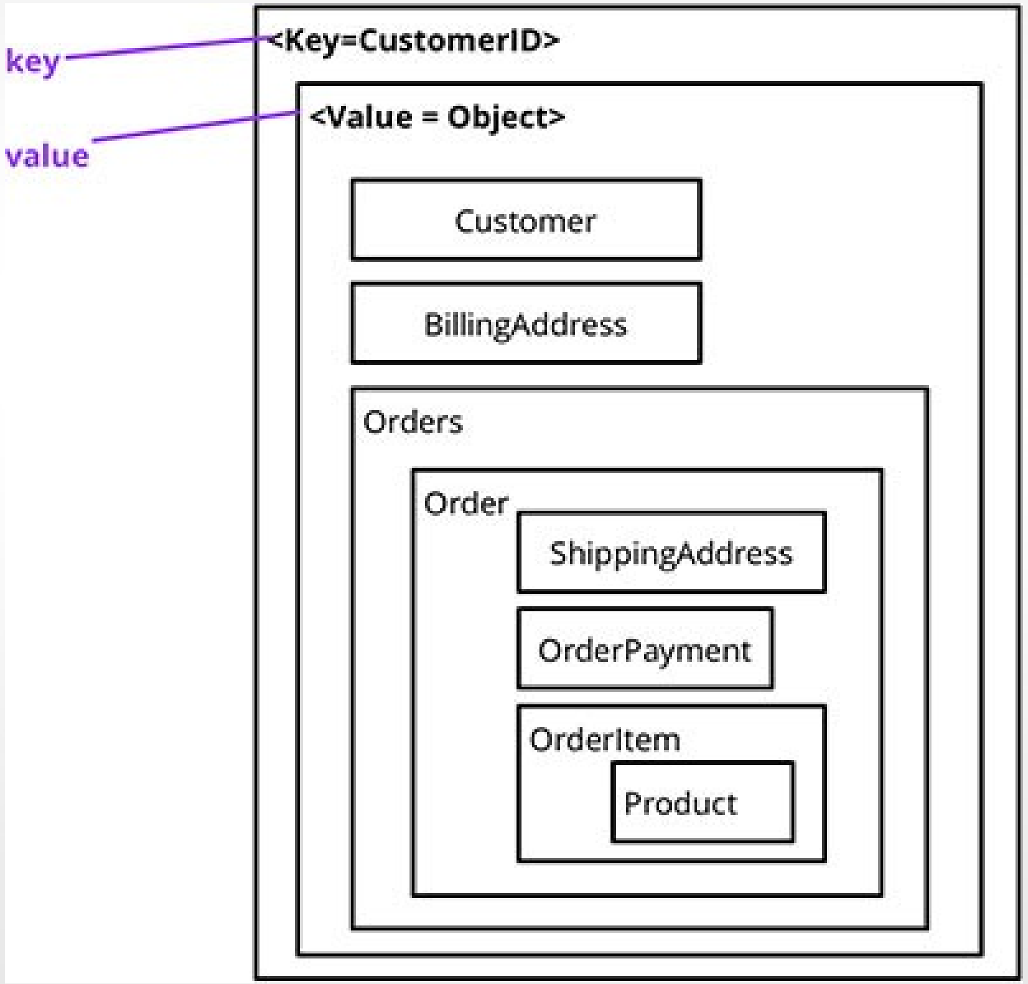
Types:

There have been various approaches to classify NoSQL databases

- **Column:** Cassandra
- **Document:** MongoDB
- **Key-value:** FoundationDB
- **Graph:** Neo4J

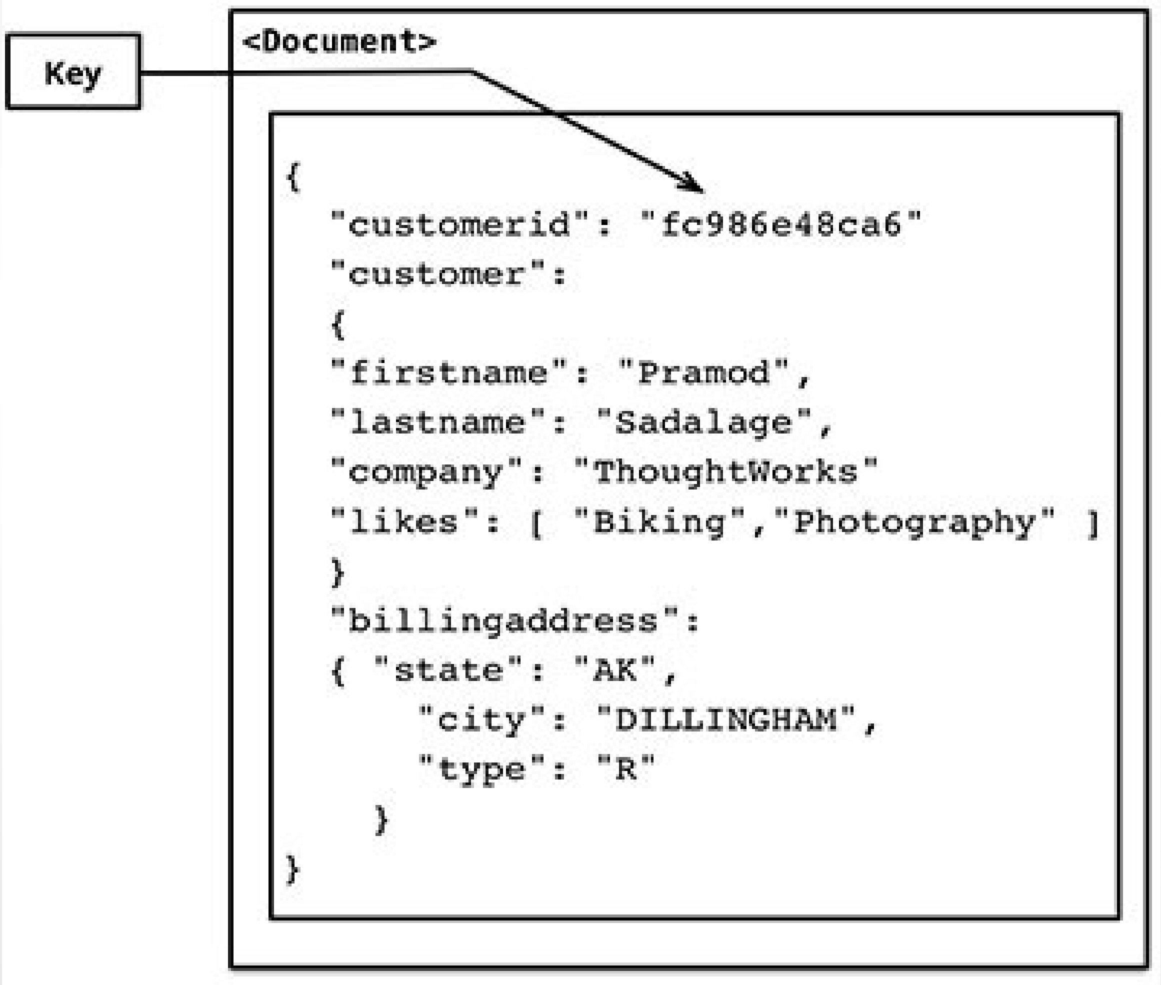
What is NoSQL

Types:



What is NoSQL

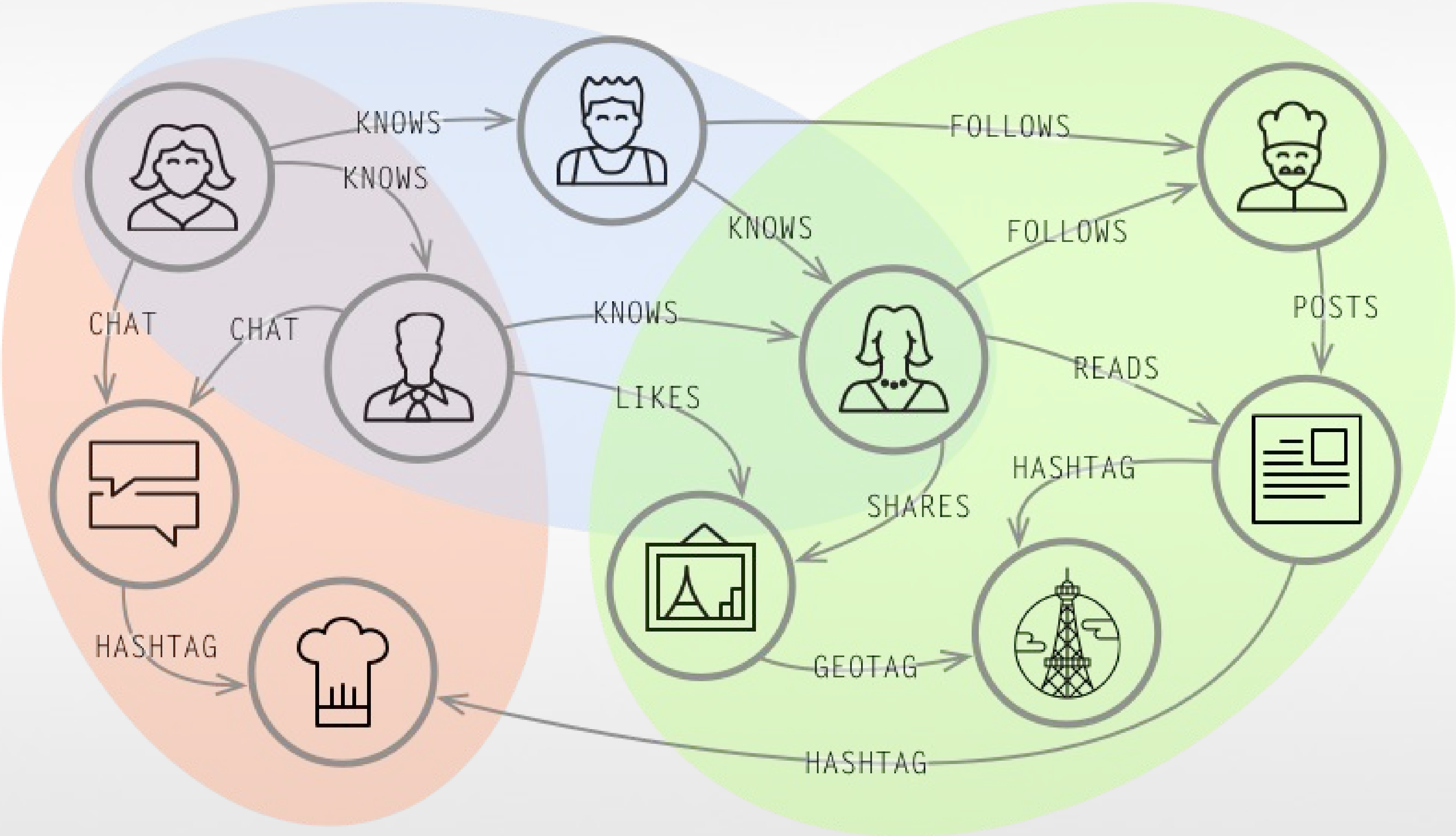
Types:



```
{
  "customerid": "fc986e48ca6"
  "customer":
  {
  "firstname": "Pramod",
  "lastname": "Sadalage",
  "company": "ThoughtWorks"
  "likes": [ "Biking","Photography" ]
  }
  "billingaddress":
  { "state": "AK",
      "city": "DILLINGHAM",
      "type": "R"
    }
}
```

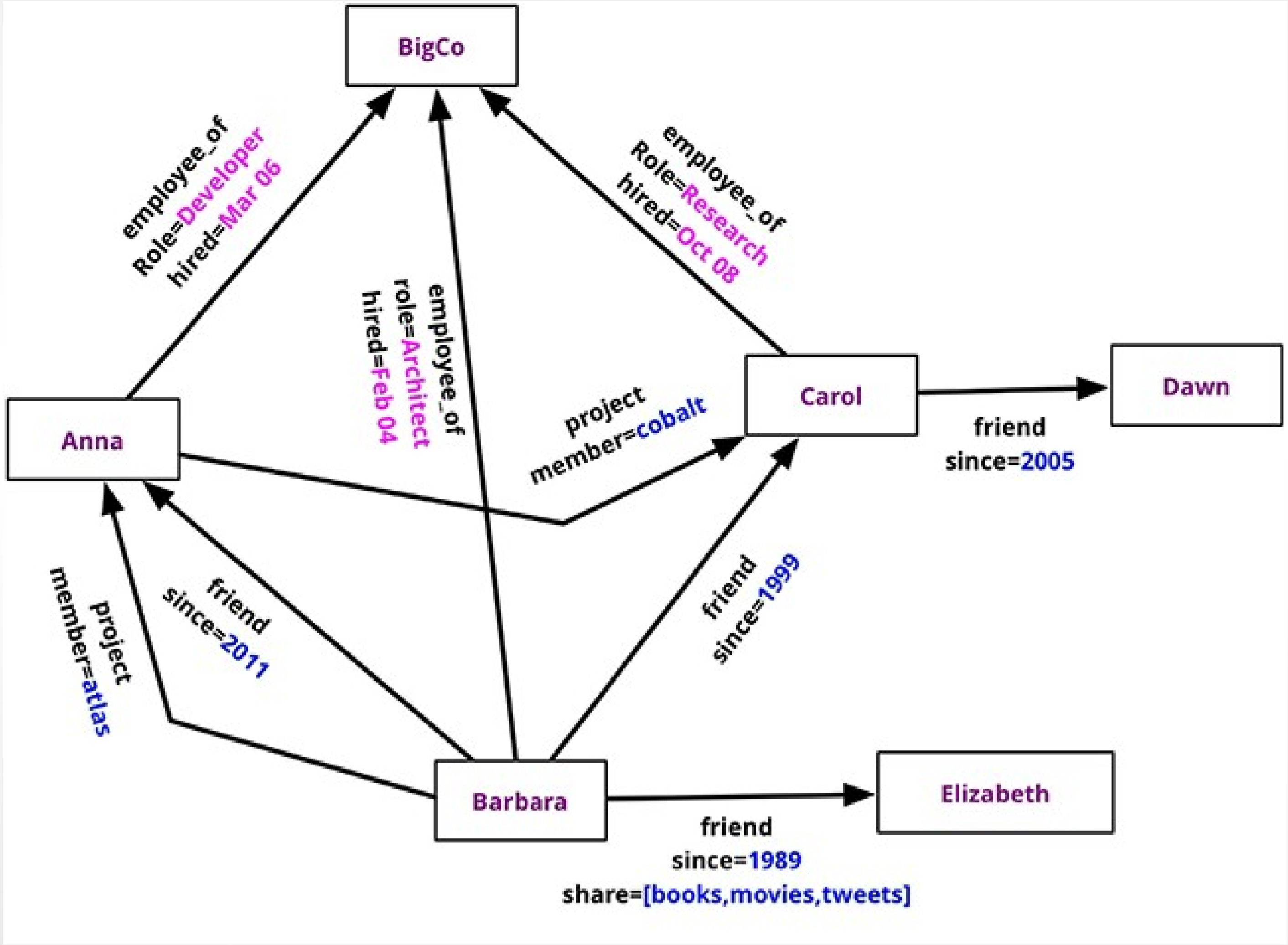
What is NoSQL

Types:



What is NoSQL

Types:



Why NoSQL?

Why NoSQL

- **Big Data is one of the key forces driving the growth and popularity of NoSQL for business.**

- **A Big Data project is normally typified by:**

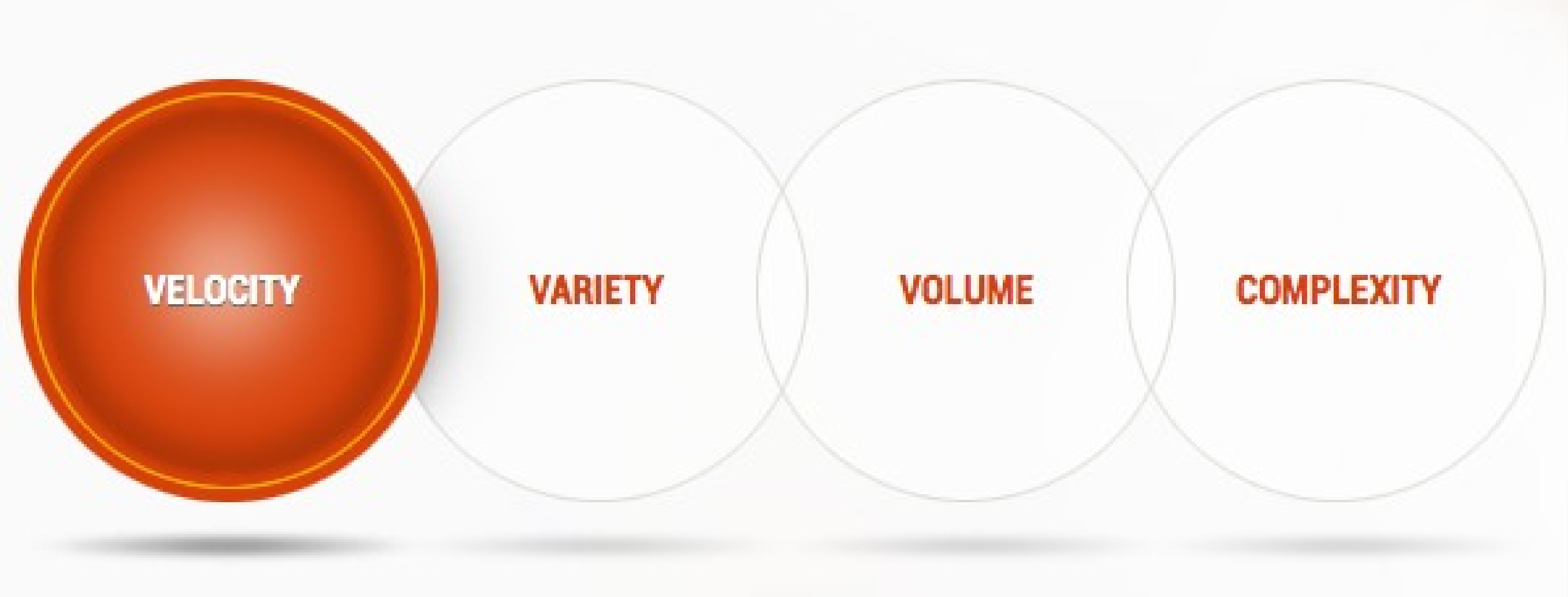
High data velocity: lots of data coming in very quickly, possibly from different locations.

Data variety: storage of data that is structured, semi-structured and unstructured.

Data volume: data that involves many terabytes or petabytes in size.

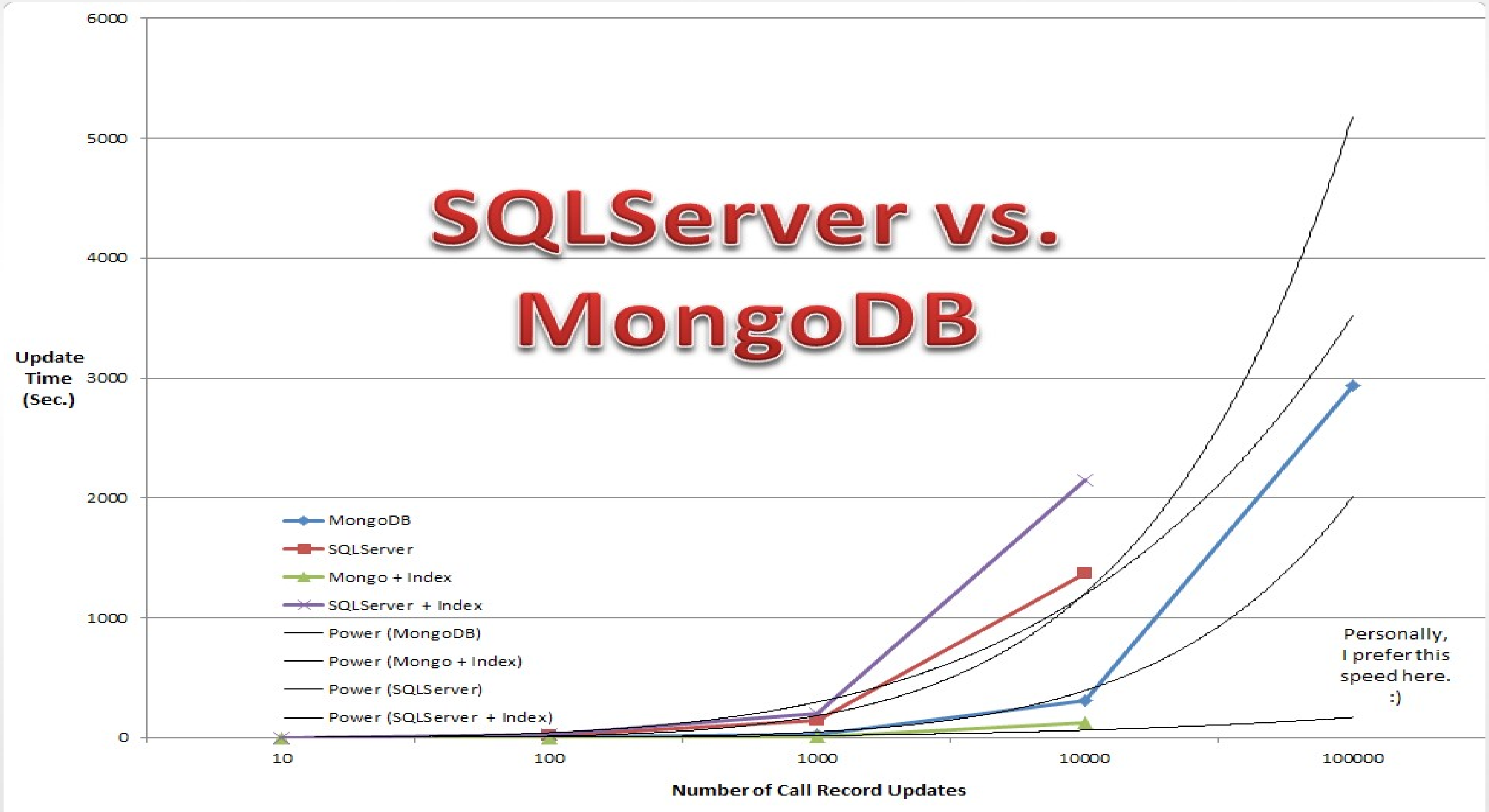
Data complexity: data that is stored and managed in different locations or data centers.

Why NoSQL

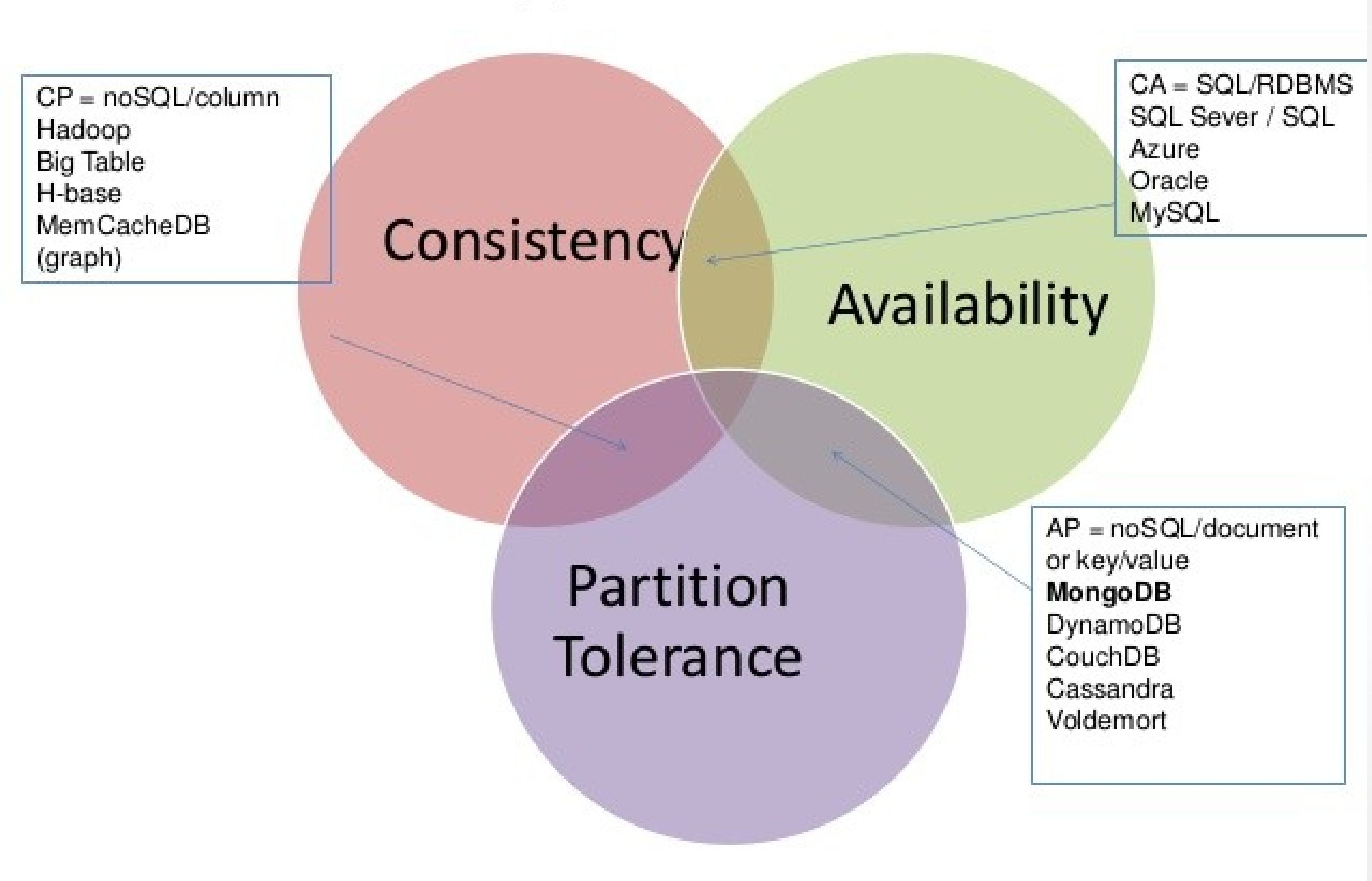


That is why we need NoSQL

Why NoSQL


SQLServer vs.
MongoDB
Update Time (Sec.)
6000
5000
4000
3000
2000
1000
0
10
100
1000
10000
100000
Number of Call Record Updates
MongoDB
SQLServer
Mongo + Index
SQLServer + Index
Power (MongoDB)
Power (Mongo + Index)
Power (SQLServer)
Power (SQLServer + Index)
Personally, I prefer this speed here. :)

Why NoSQL



CP = noSQL/column
Hadoop
Big Table
H-base
MemCacheDB
(graph)

CA = SQL/RDBMS
SQL Sever / SQL
Azure
Oracle
MySQL

AP = noSQL/document
or key/value
MongoDB
DynamoDB
CouchDB
Cassandra
Voldemort

CAP Theory

Why NoSQL

Scenarios where NoSQL SHOULD be used:

- Your relational database will not scale to your traffic at an acceptable cost.
- In a NoSQL database, there is no fixed schema and no joins. NoSQL can take advantage of “scaling out”. Scaling out refers to spreading the load over many commodity systems.
- It’s useful for creating prototypes or fast applications as it provides a tool to develop new features easily.
- You have local data transactions which do not have to be very durable. **e.g. “liking”** items on websites.
- Agile sprints, quick iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible

Why NoSQL

Scenarios where NoSQL SHOULD NOT be used:

- It cannot necessarily guarantee the ACID(Atomicity,Consistancy,Isolation,Durability) properties for your transactions.
- Normally an interface is provided for storing your data. Do not try to use a complicated query in that interface.
- The developer should always keep in mind that NoSQL database is not built on tables and usually doesn’t use structured query language.
- If consistency is mandatory and there will be no drastic changes in terms of the data volume.

Why MongoDB?

Why MongoDB

- **It is Open Source :)**
- **High Write Load :**
  MongoDB by default prefers high insert rate over transaction safety. If you need to load tons of data lines with a low business value for each one.
- **High Availability in an Unreliable Environment :**
  Setting replicaSet (set of servers that act as Master-Slaves) is easy and fast. Moreover, recovery from a node (or a data center) failure is instant, safe and automatic.
- **You need to Grow Big:**
  Databases scaling is hard (a single MySQL table performance will degrade when crossing the 5-10GB per table).

Why MongoDB

- **Location Based:**
  MongoDB has built in spacial functions, so finding relevant data from specific locations is fast and accurate.

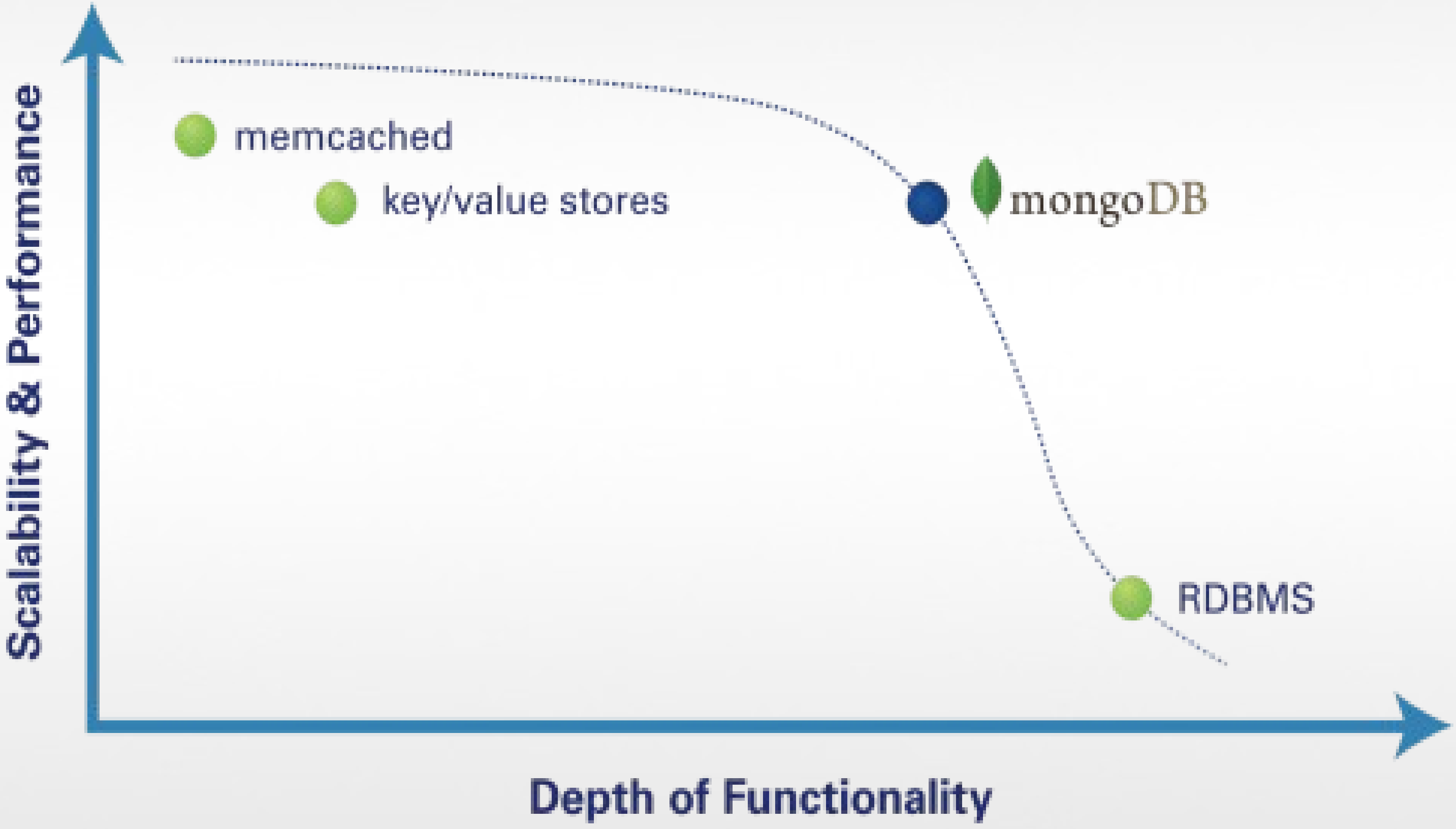
- **Data Set is Going to be Big (from 1GB) and Schema is Not Stable:**
  Adding new columns to RDBMS can lock the entire database in some database, or create a major load and performance degradation in other.

- **You Don't have a DBA:**
  If you don't have a DBA, and you don't want to normalize your data and do joins, you should consider MongoDB.

  Note: If you are expecting to go big, please notice that you will need to follow some best practices to avoid pitfalls.

http://www.mongodb.com/use-cases

Why MongoDB



Scalability & Performance
memcached
key/value stores
mongoDB
RDBMS
Depth of Functionality

Why MongoDB

Real World Case Study: Billing

BillRun

Billing on top of MongoDB

Ofer Cohen, S.D.O.C. Ltd.

MUG IL, February 2014

mongoDB

Getting Started

Getting Started

Install on Ubuntu:

- Packages MongoDB provides packages of the officially supported MongoDB builds in its own repository. This repository provides the MongoDB distribution in the following packages:

- mongodb-org
- mongodb-org-server
- mongodb-org-mongos
- mongodb-org-shell
- mongodb-org-tools

- **Control Scripts** : The mongodb-org package includes various control scripts, including the init script /etc/init.d/mongod. These scripts are used to stop, start, and restart daemon processes.

- The package **configures** MongoDB using the /etc/mongod.conf file in conjunction with the control scripts.

Getting Started

Step 1: Import the public key used by the package management system:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
```

Step 2: Create a list file for MongoDB. using the following command: Create the /etc/apt/sources.list.d/mongodb.list list file

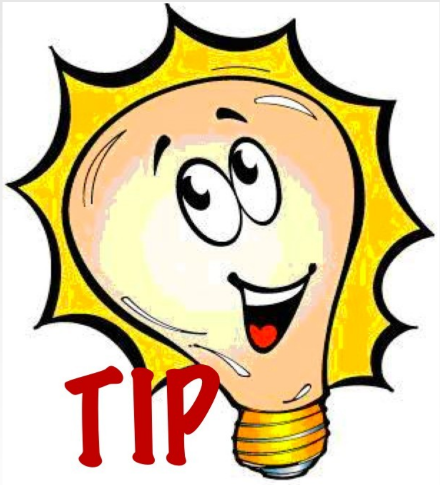
```
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo tee /etc/apt/sources.list.d/mongodb.list
```

Step 3: Reload local package database. Issue the following command to reload the local package database:

```
sudo apt-get update
```

Step 4: Install the MongoDB packages:

```
sudo apt-get install -y mongodb-org
```



If you truly wish to master a skill, nothing beats hands-on experience

SOOOO :)
Let's Start !

Mongo Basics

Mongo Basics

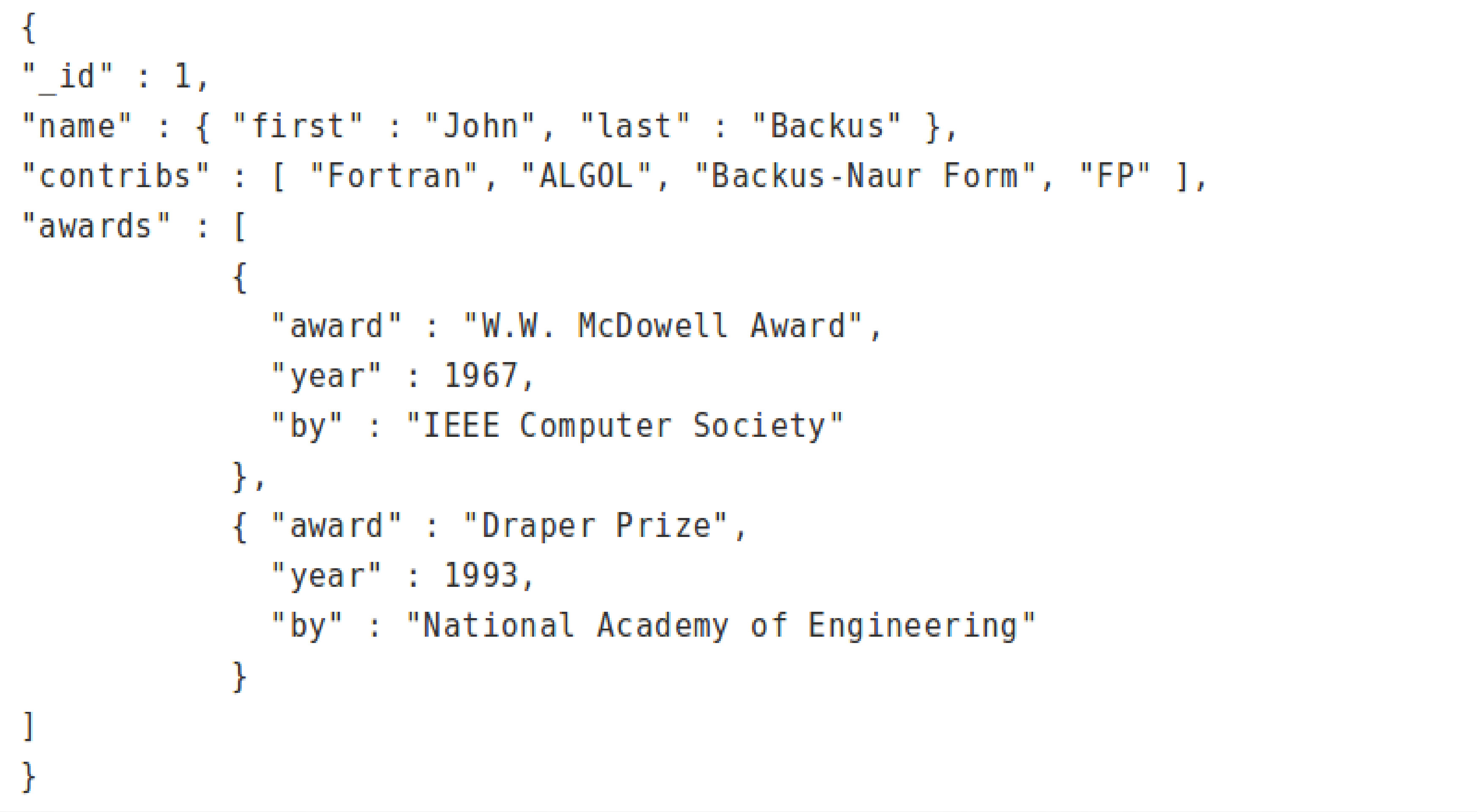
Binary JSON (BSON)

MongoDB represents JSON documents in binary-encoded format called BSON behind the scenes. BSON extends the JSON model to provide additional data types and to be efficient for encoding and decoding within different languages.

JSON

JavaScript Object Notation (JSON) is an open, human and machine-readable standard that facilitates data interchange, and along with XML is the main format for data interchange used on the modern web. JSON supports all the basic data types you’d expect: numbers, strings, and boolean values, as well as arrays and hashes.

Mongo Basics



```
{
"_id" : 1,
"name" : { "first" : "John", "last" : "Backus" },
"contribs" : [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
"awards" : [
           {
             "award" : "W.W. McDowell Award",
             "year" : 1967,
             "by" : "IEEE Computer Society"
           },
           { "award" : "Draper Prize",
             "year" : 1993,
             "by" : "National Academy of Engineering"
           }
]
}
```

Mongo Basics

Connect to a mongod:

From a system prompt, start mongo by issuing the mongo command, as follows:

```
$ mongo
```

By default, mongo looks for a database server listening on port 27017 on the localhost interface. To connect to a server on a different port or interface, use the --port and --host options.

Select a Database:

After starting the mongo shell your session will use the test database by default. At any time, issue the following operation at the mongo to report the name of the current database:

```
> db
```

Mongo Basics

From the mongo shell, display the list of databases, with the following operation:

```
> show dbs
```

Switch to a new database named mydb, with the following operation:

```
> use mydb
```

Confirm that your session has the mydb database as context, by checking the value of the db object, which returns the name of the current database, as follows:

```
> db
```

Mongo Basics

Display mongo Help

At any point, you can access help for the mongo shell using the following operation:

```
> help
```

Furthermore, you can append the ***.help()*** method to some JavaScript methods, any cursor object, as well as the db and db.collection objects to return additional help information.

Mongo Basics

Demo