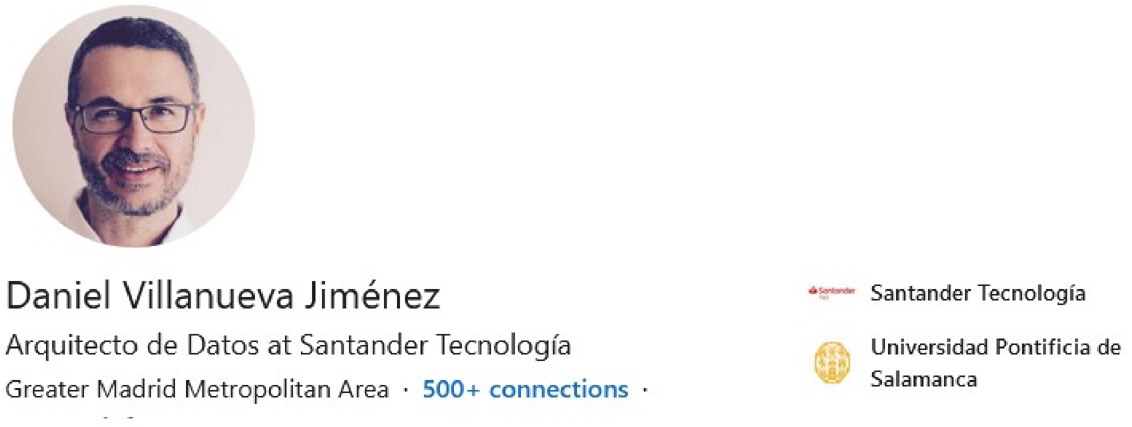
Programming and frameworks for ML

Introduction to NoSQL Databases

About Me

Big Data Consultant at Santander / Big Data Lecturer

- More than 20 years of experience in different environments, technologies, customers, countries ...
- Passionate about data and technology
- Enthusiastic about Big Data world and NoSQL



Daniel Villanueva Jiménez
Arquitecto de Datos at Santander Tecnología
Greater Madrid Metropolitan Area · 500+ connections ·

Santander Tecnología
Universidad Pontificia de Salamanca

daniel.villanueva@immune.institute

IMMUNE
CODING INSTITUTE

Objectives

- Tour of different database models
- Comparison of a relational database with NoSql databases
  - Key/Value
  - Documents
  - Column oriented
  - Graphs

IMMUNE CODING INSTITUTE

Agenda

- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- MongoDB
- Apache Cassandra
- Neo4j

Material - Virtual Machine

jupyterlab TNG 2.2.9 http://localhost:8001/

mongoDB 4.2.5 http://localhost:3100/

neo4j 3.5.11 http://localhost:7474/

riak 2.2.3 http://localhost:8098/

ssh http://localhost:2222/

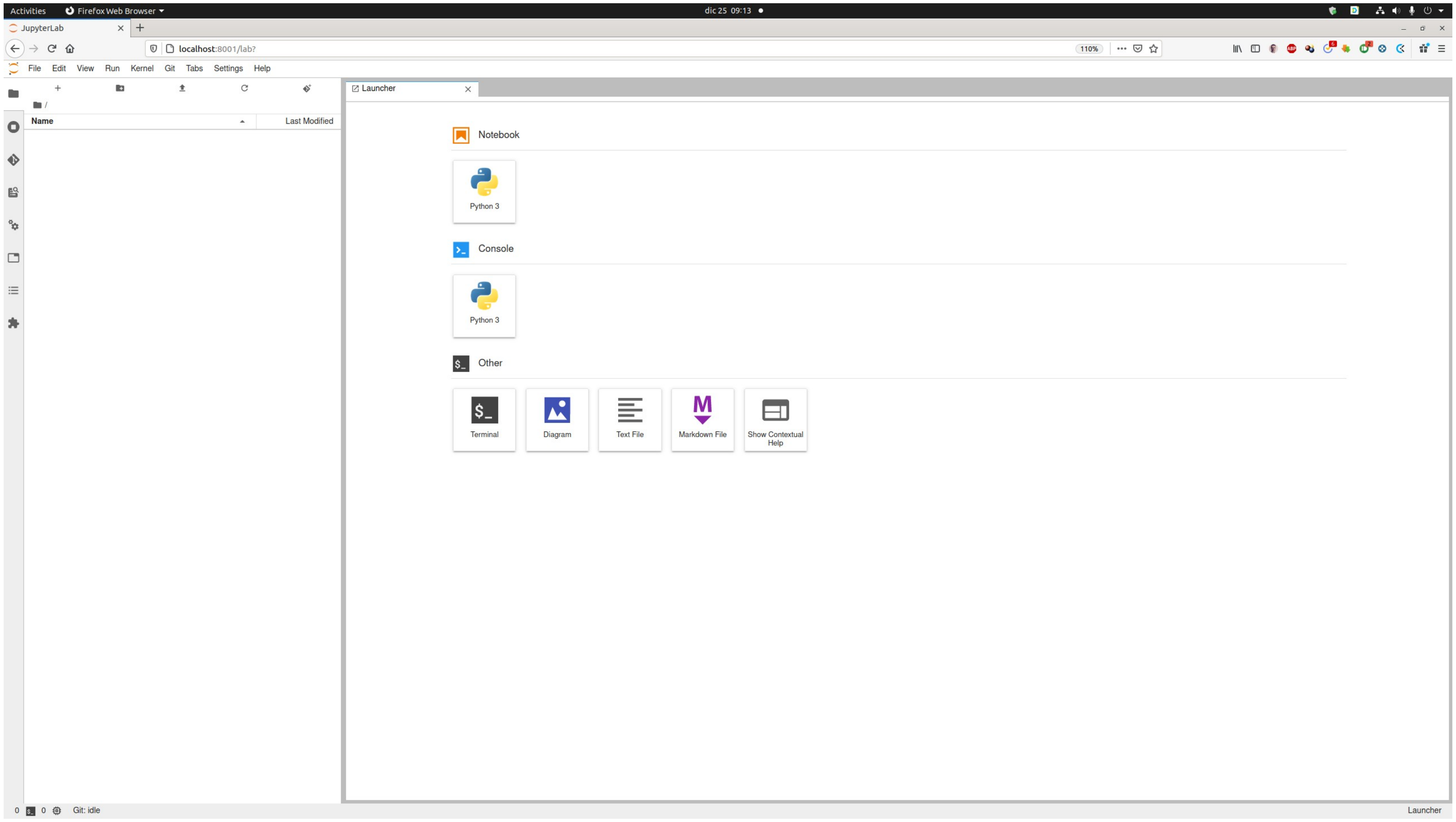
PostgreSQL 12.2

Cassandra 3.11

learner/learner

- https://github.com/dvillaj/NoSQL-box

Material - Virtual Machine



What are databases?

IMMUNE CODING INSTITUTE

What are databases?

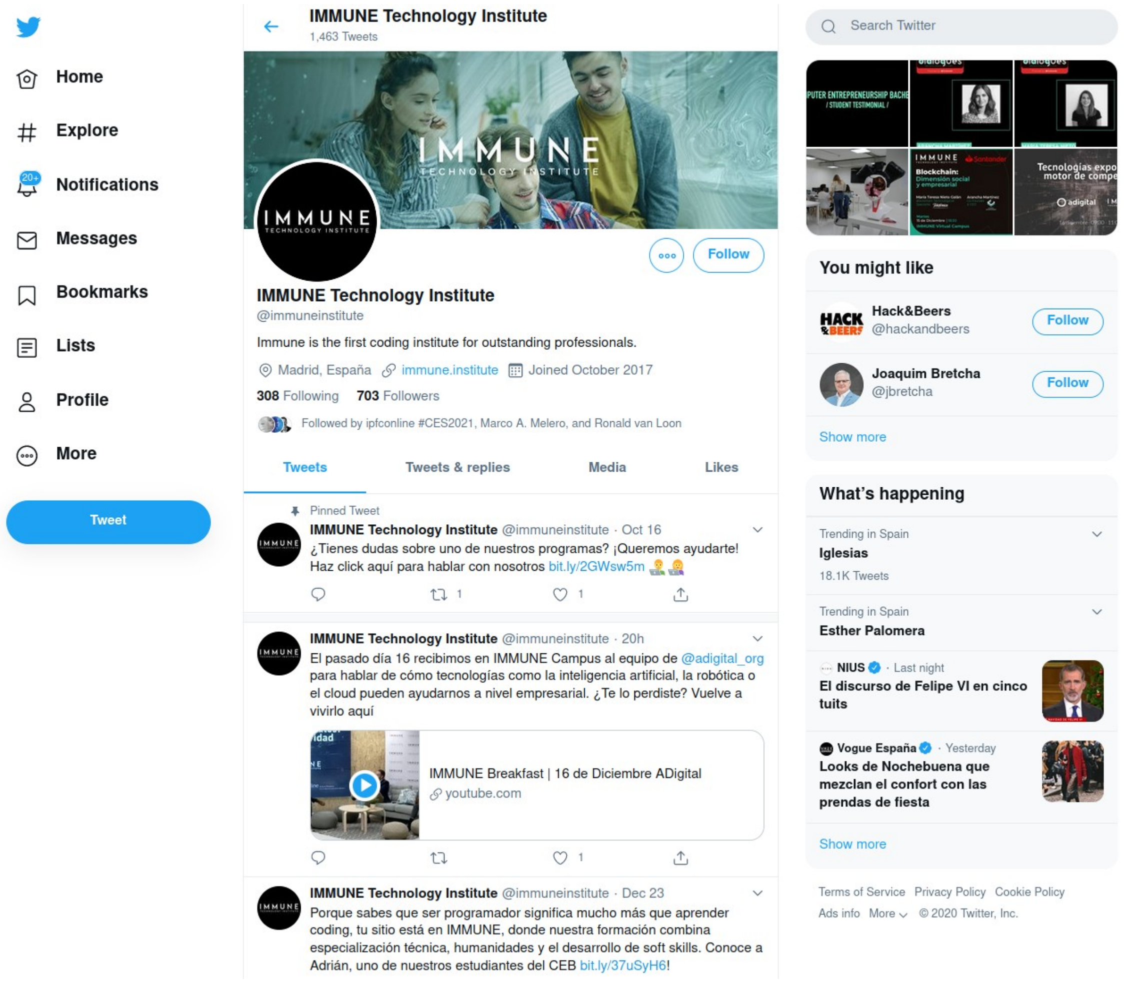
"A database is a **storehouse** that allows us to store **large amounts of information in an organized** manner so that we can easily **find** and **use it."**

Agenda

- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- MongoDB
- Apache Cassandra
- Neo4j

IMMUNE CODING INSTITUTE

Case Study - Twitter


Home
Explore
Notifications
Messages
Bookmarks
Lists
Profile
More
Tweet

IMMUNE Technology Institute
1,463 Tweets

Follow

IMMUNE Technology Institute
@immuneinstitute

Immune is the first coding institute for outstanding professionals.

Madrid, España immune.institute Joined October 2017

308 Following 703 Followers

Followed by ipfconline #CES2021, Marco A. Melero, and Ronald van Loon

Tweets | Tweets & replies | Media | Likes

Pinned Tweet
IMMUNE Technology Institute @immuneinstitute · Oct 16
¿Tienes dudas sobre uno de nuestros programas? ¡Queremos ayudarte! Haz click aquí para hablar con nosotros bit.ly/2GWsw5m
1 1

IMMUNE Technology Institute @immuneinstitute · 20h
El pasado día 16 recibimos en IMMUNE Campus al equipo de @adigital_org para hablar de cómo tecnologías como la inteligencia artificial, la robótica o el cloud pueden ayudarnos a nivel empresarial. ¿Te lo perdiste? Vuelve a vivirlo aquí

IMMUNE Breakfast | 16 de Diciembre ADigital
youtube.com
1

IMMUNE Technology Institute @immuneinstitute · Dec 23
Porque sabes que ser programador significa mucho más que aprender coding, tu sitio está en IMMUNE, donde nuestra formación combina especialización técnica, humanidades y el desarrollo de soft skills. Conoce a Adrián, uno de nuestros estudiantes del CEB bit.ly/37uSyH6!

Search Twitter

You might like
Hack&Beers @hackandbeers Follow
Joaquim Bretcha @jbretcha Follow
Show more

What's happening
Trending in Spain
Iglesias
18.1K Tweets

Trending in Spain
Esther Palomera

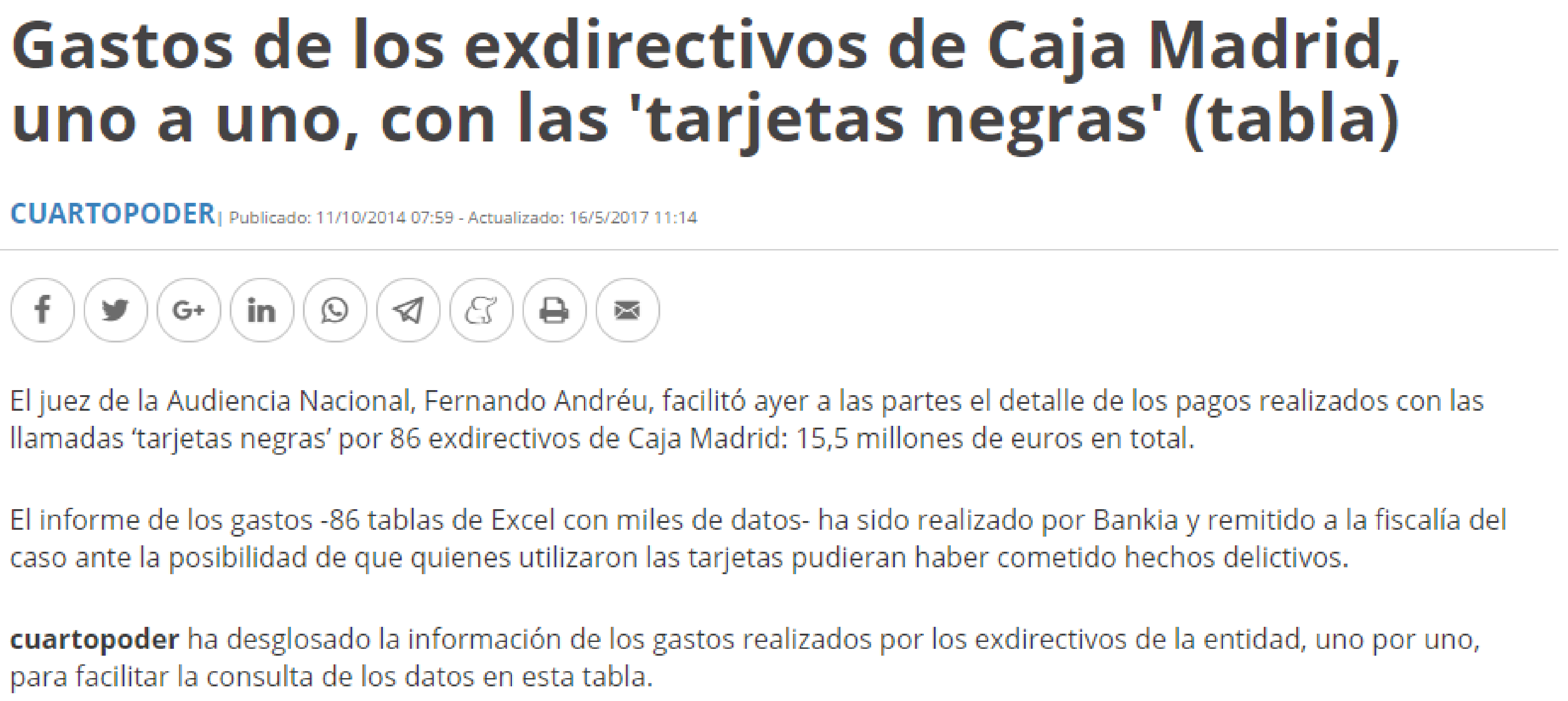
NIUS · Last night
El discurso de Felipe VI en cinco tuits

Vogue España · Yesterday
Looks de Nochebuena que mezclan el confort con las prendas de fiesta

Show more

Terms of Service Privacy Policy Cookie Policy
Ads info More © 2020 Twitter, Inc.

Case Study – Tarjetas Black



Gastos de los exdirectivos de Caja Madrid, uno a uno, con las 'tarjetas negras' (tabla)

CUARTOPODER | Publicado: 11/10/2014 07:59 - Actualizado: 16/5/2017 11:14

El juez de la Audiencia Nacional, Fernando Andréu, facilitó ayer a las partes el detalle de los pagos realizados con las llamadas 'tarjetas negras' por 86 exdirectivos de Caja Madrid: 15,5 millones de euros en total.

El informe de los gastos -86 tablas de Excel con miles de datos- ha sido realizado por Bankia y remitido a la fiscalía del caso ante la posibilidad de que quienes utilizaron las tarjetas pudieran haber cometido hechos delictivos.

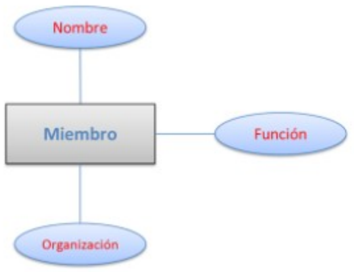
cuartopoder ha desglosado la información de los gastos realizados por los exdirectivos de la entidad, uno por uno, para facilitar la consulta de los datos en esta tabla.

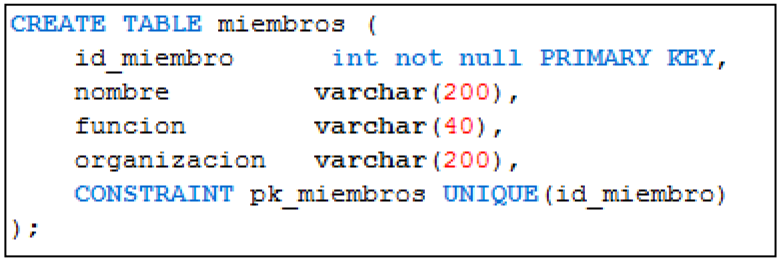
Agenda

- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- MongoDB
- Apache Cassandra
- Neo4j

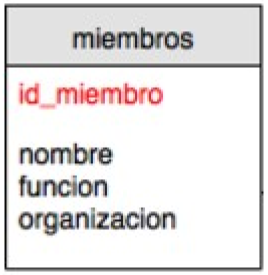
Elements of a relational database

- Tables
- Fields (or Columns)



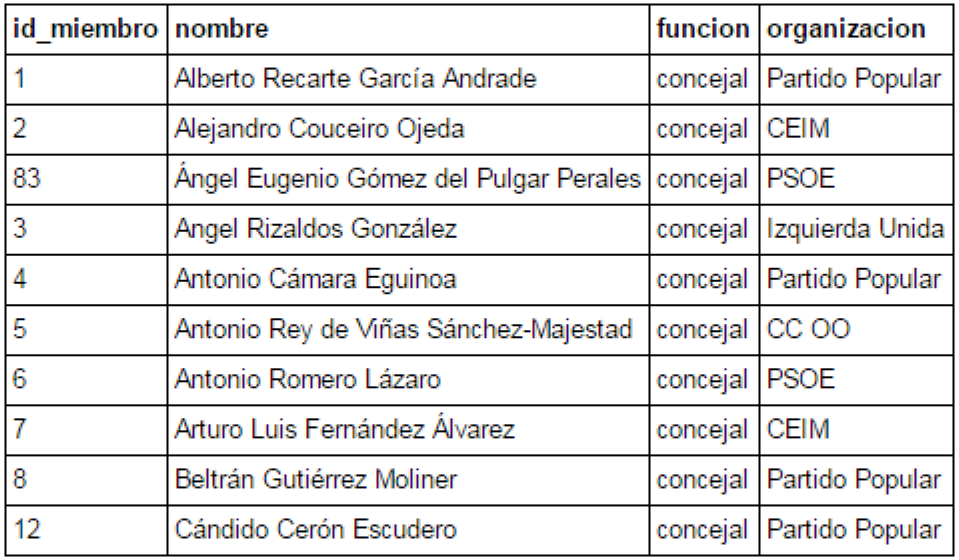


```
CREATE TABLE miembros (
    id_miembro       int not null PRIMARY KEY,
    nombre          varchar(200),
    funcion         varchar(40),
    organizacion    varchar(200),
    CONSTRAINT pk_miembros UNIQUE(id_miembro)
);
```



Elements of a relational database

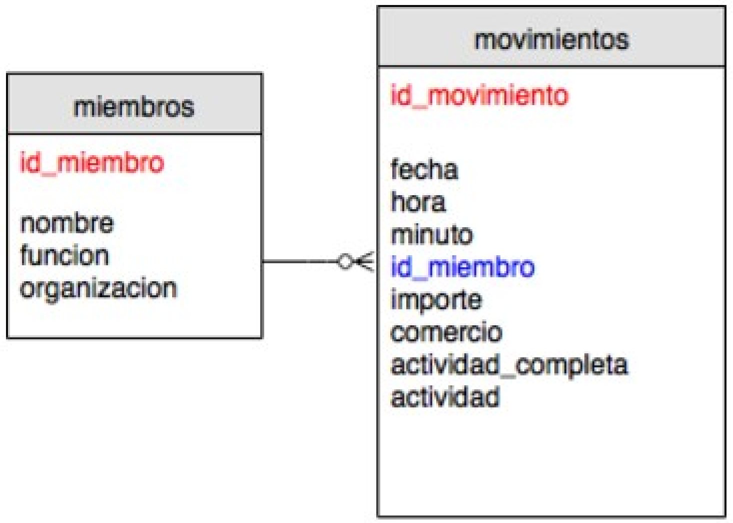
- Records (or Rows)



id_miembro	nombre	funcion	organizacion
1	Alberto Recarte García Andrade	concejal	Partido Popular
2	Alejandro Couceiro Ojeda	concejal	CEIM
83	Ángel Eugenio Gómez del Pulgar Perales	concejal	PSOE
3	Angel Rizaldos González	concejal	Izquierda Unida
4	Antonio Cámara Eguinoa	concejal	Partido Popular
5	Antonio Rey de Viñas Sánchez-Majestad	concejal	CC OO
6	Antonio Romero Lázaro	concejal	PSOE
7	Arturo Luis Fernández Álvarez	concejal	CEIM
8	Beltrán Gutiérrez Moliner	concejal	Partido Popular
12	Cándido Cerón Escudero	concejal	Partido Popular

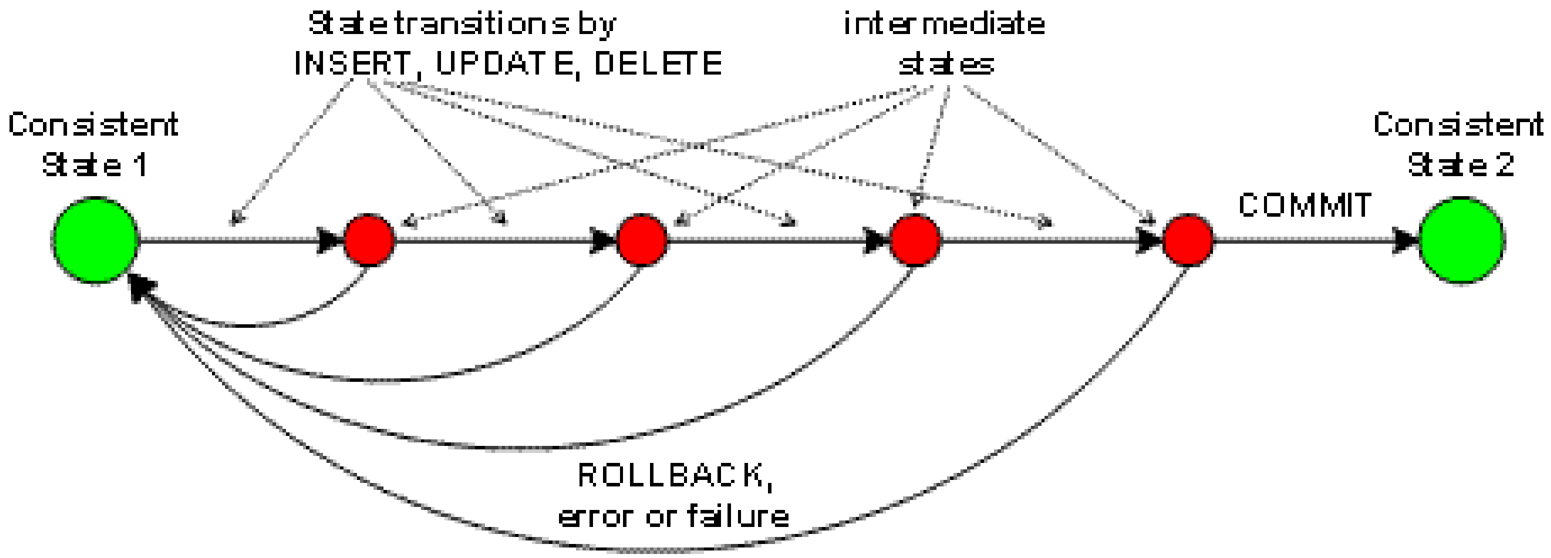
Elements of a relational database

- Relationships between tables
- Primary Keys
- Foreign Keys



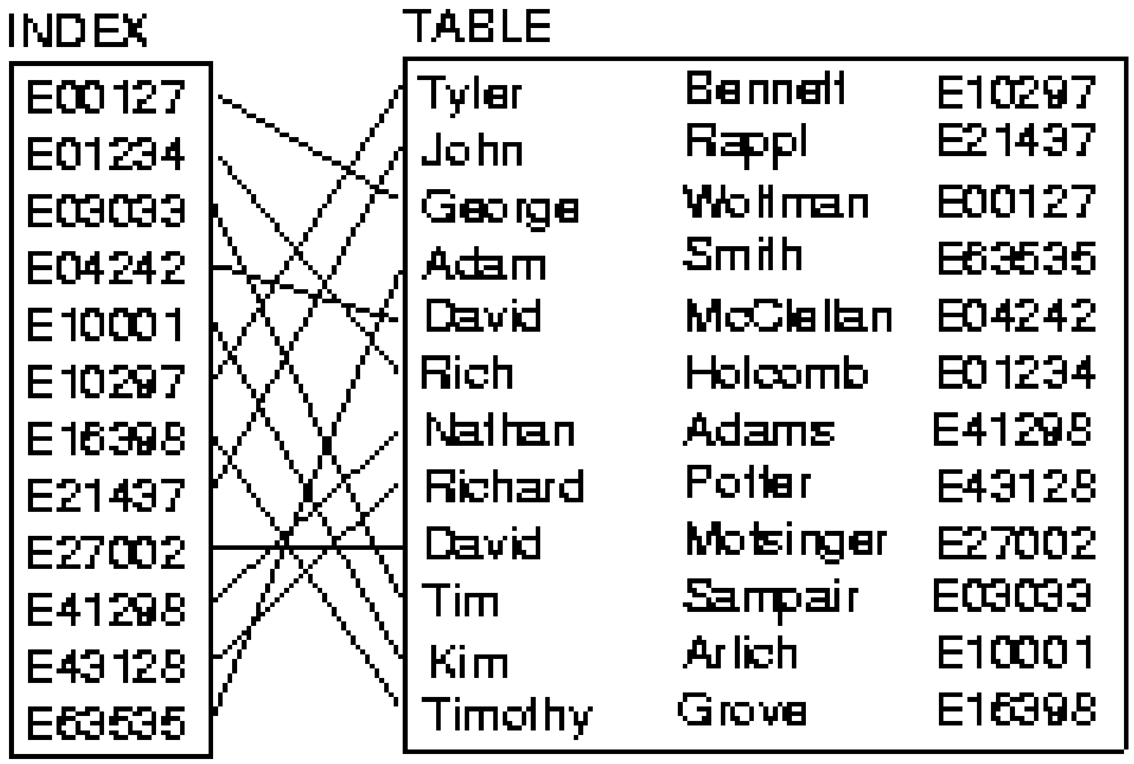
Elements of a relational database

- Views
- Transactions



Elements of a relational database

- Indexes



INDEX

E00127
E01234
E03033
E04242
E10001
E10297
E16398
E21437
E27002
E41298
E43128
E63535

TABLE

Tyler	Bennett	E10297
John	Rappl	E21437
George	Wolfman	E00127
Adam	Smith	E63535
David	McClellan	E04242
Rich	Holcomb	E01234
Nathan	Adams	E41298
Richard	Potter	E43128
David	Motsinger	E27002
Tim	Sampair	E03033
Kim	Arlich	E10001
Timothy	Grove	E16398

IMMUNE CODING INSTITUTE

Elements of a relational database

- SQL Language

```
CREATE TABLE users (
    user_id          bigint not null PRIMARY KEY,
    screen_name      varchar(50) not null,
    name             varchar(50) null,
    created_at       timestamp with time zone null,
    description      varchar(200) null,
    retweet_count    int null,
    favorite_count   int null,
    friends_count    int null,
    followers_count  int null,
    statuses_count   int null,
    geo_enabled      boolean null,
    time_zone        varchar(50) null,
    profile_image_url varchar(300) null
);
```

```
select user_id, count(*) as count
from tweet_usermention
group by 1
having count(*) > 0
order by 2 desc
```

```
CREATE INDEX id_user_screenname
ON users (screen_name)
```

```
SELECT * FROM tweets
WHERE geo_type is not null
LIMIT 10
```

```
DELETE FROM users
WHERE user_id = 2012312
```

ACID properties associated to a Relational database

Atomicity

Each transaction is “all or nothing”

Consistency

Data should be valid according to all defined rules

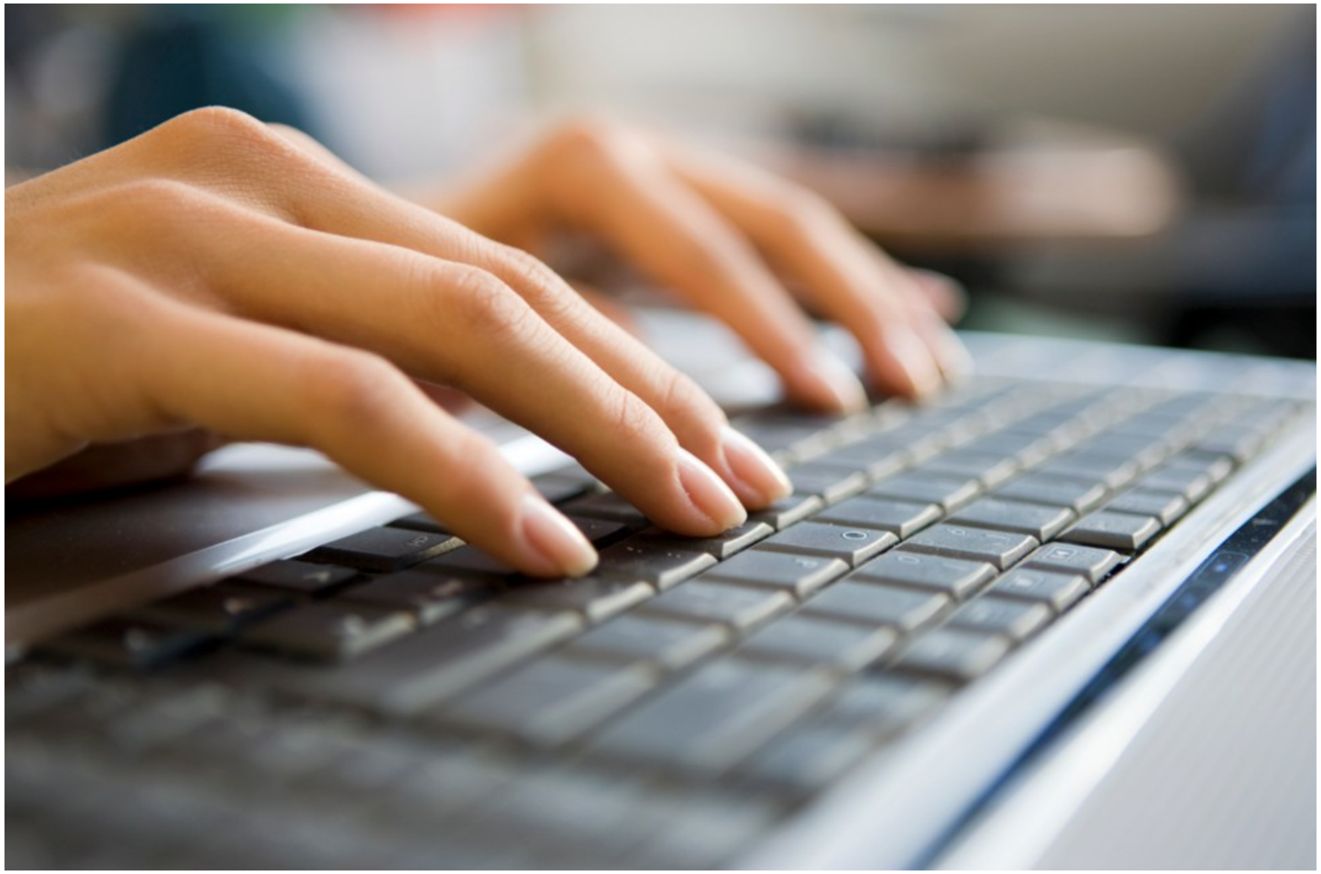
Isolation

Transactions do not affect each other

Durability

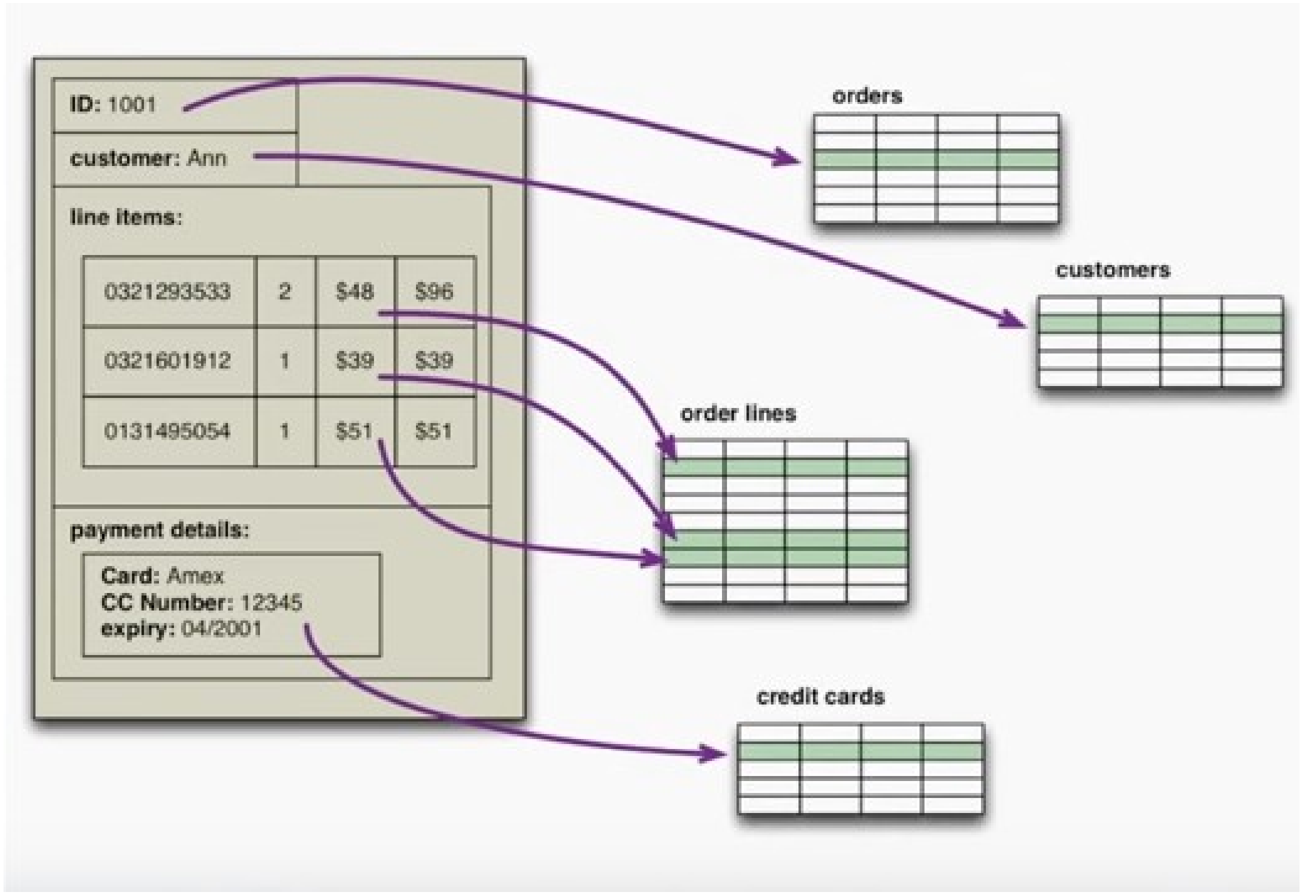
Committed data would not be lost, even after power failure.

PostgreSQL - HandsOn



Problems?

Impedance Mismatch



Rigid schemes

```
CREATE TABLE Customers (
    Customer_Id Int,
    Name Varchar(100),
    ...
)
```

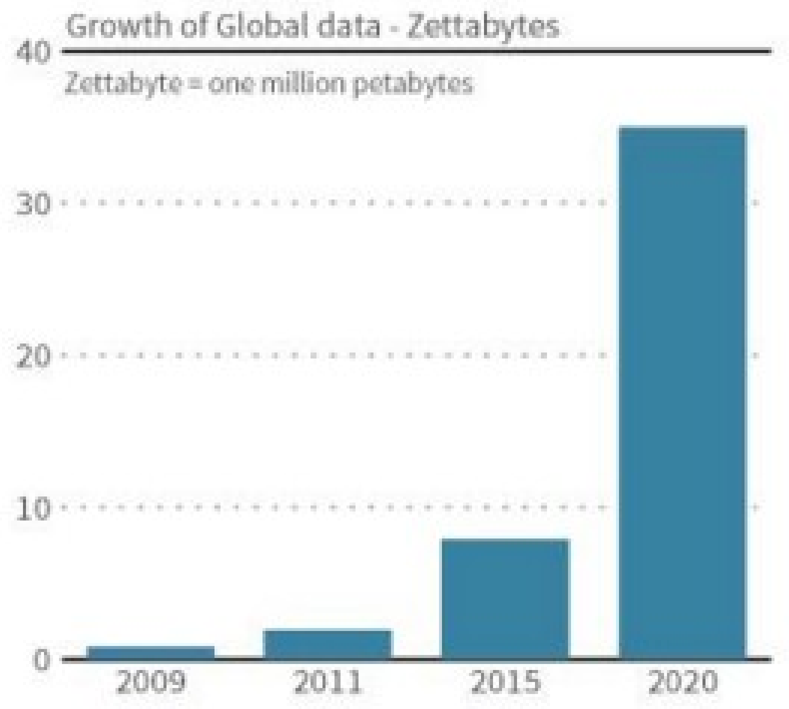
```
BULK INSERT Customers
FROM 'CustFile.txt'
WITH FIELDTERMINATOR = ';'
```

```
SELECT Customer_Id, Name
FROM Customers
```

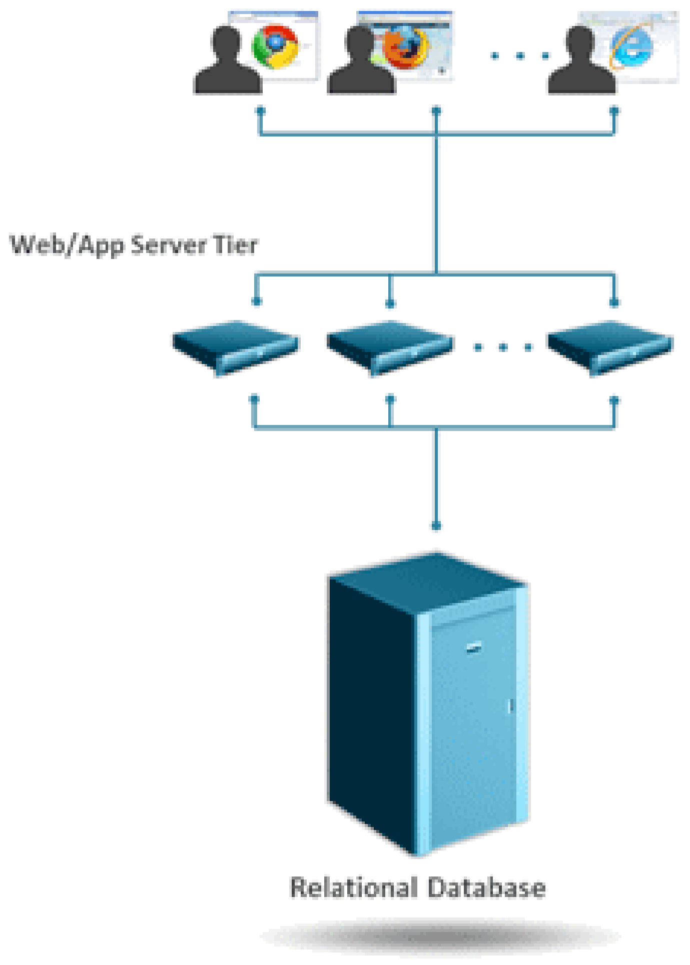
Rigid schemes

- You cannot load the information until you create the structure in the database
- You cannot create the structure until you understand the schema to be stored in the table
- What happens if the data changes?

Volume

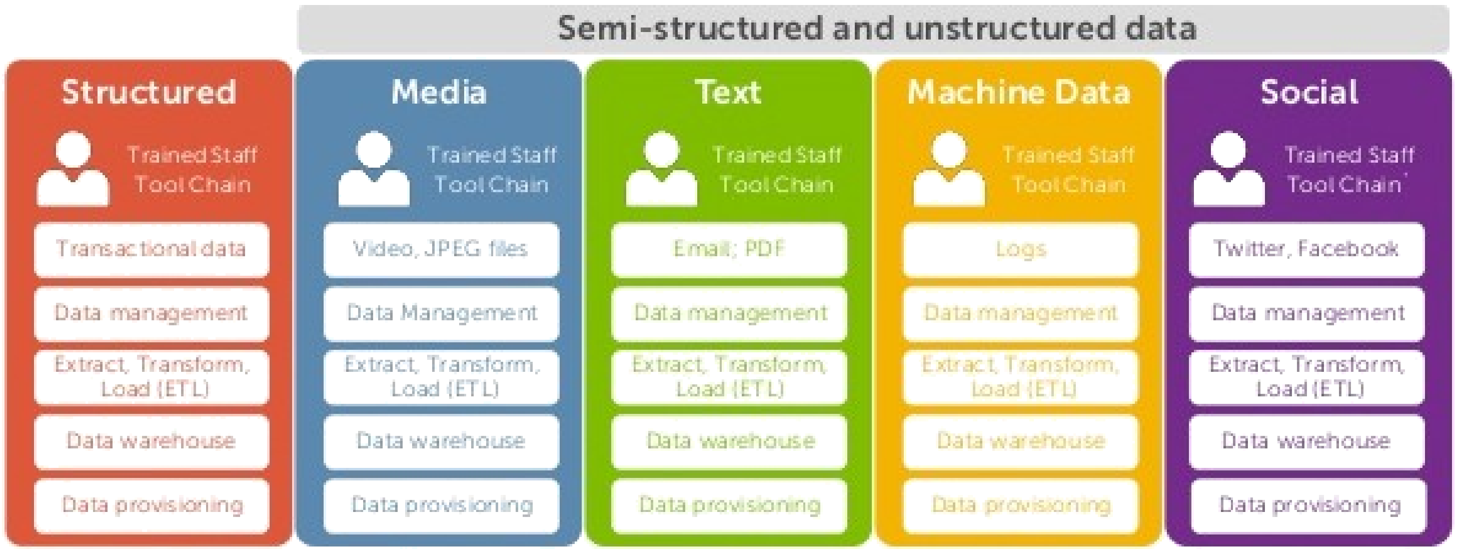

Growth of Global data - Zettabytes
Zettabyte = one million petabytes
40
30
20
10
0
2009
2011
2015
2020

Scalability



Scalability

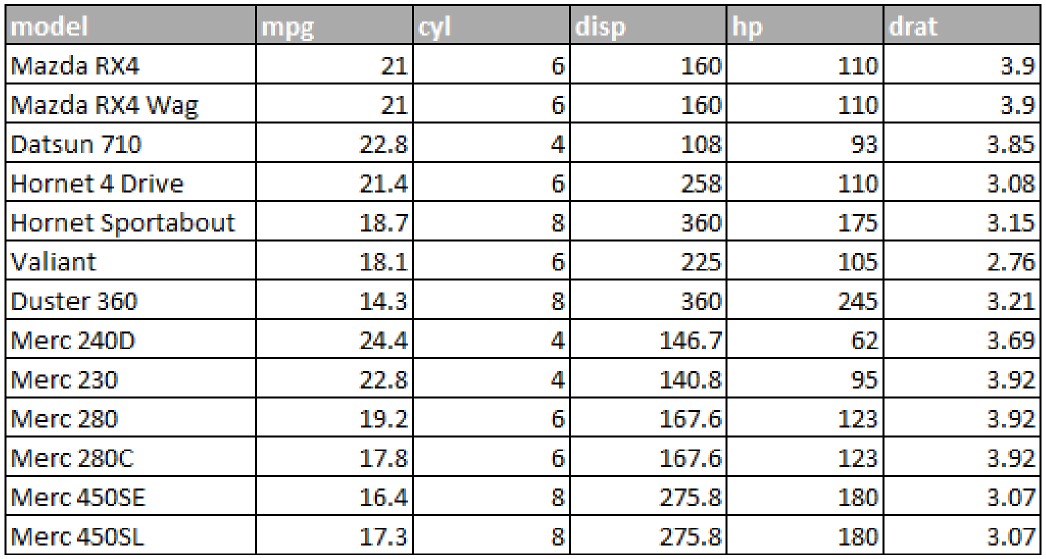
Variety of information



	Semi-structured and unstructured data			
Structured	Media	Text	Machine Data	Social
Trained Staff Tool Chain	Trained Staff Tool Chain	Trained Staff Tool Chain	Trained Staff Tool Chain	Trained Staff Tool Chain
Transactional data	Video, JPEG files	Email; PDF	Logs	Twitter, Facebook
Data management	Data Management	Data management	Data management	Data management
Extract, Transform, Load (ETL)	Extract, Transform, Load (ETL)	Extract, Transform, Load (ETL)	Extract, Transform, Load (ETL)	Extract, Transform, Load (ETL)
Data warehouse	Data warehouse	Data warehouse	Data warehouse	Data warehouse
Data provisioning	Data provisioning	Data provisioning	Data provisioning	Data provisioning

IMMUNE CODING INSTITUTE

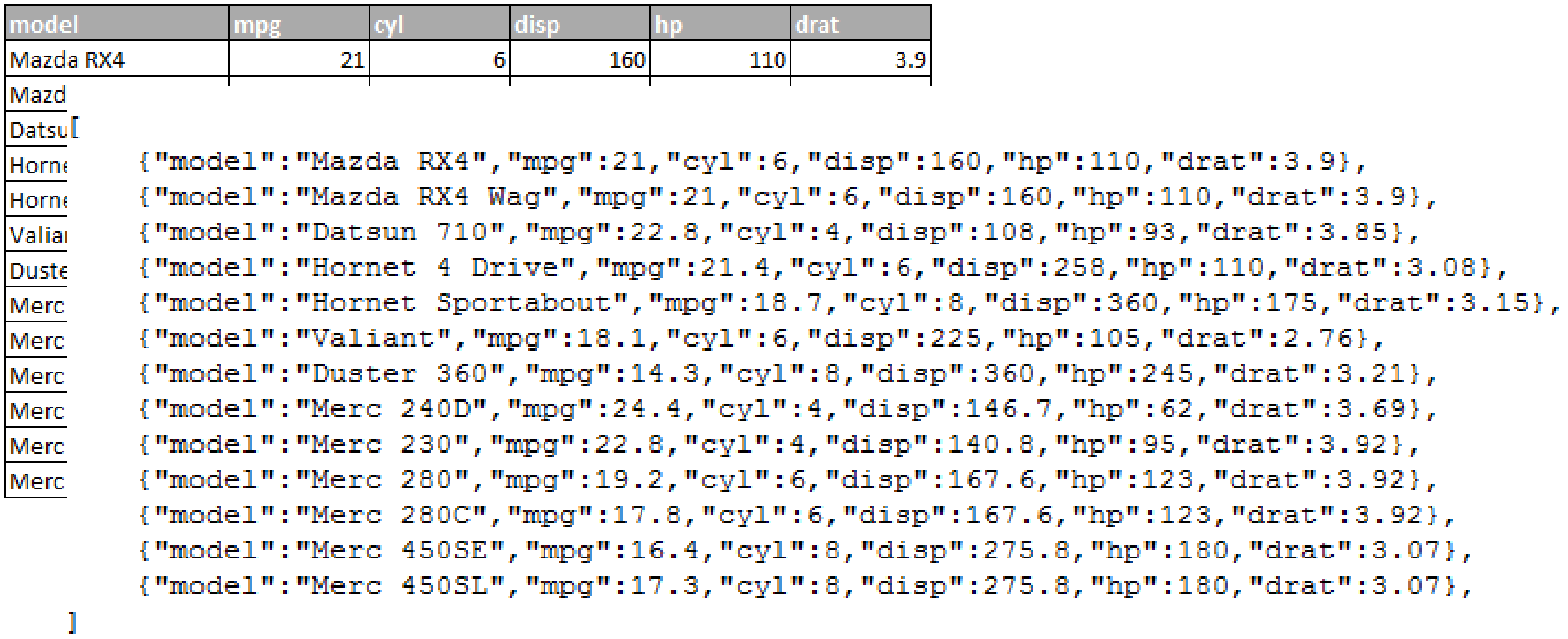
Structured Data



model	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.85
Hornet 4 Drive	21.4	6	258	110	3.08
Hornet Sportabout	18.7	8	360	175	3.15
Valiant	18.1	6	225	105	2.76
Duster 360	14.3	8	360	245	3.21
Merc 240D	24.4	4	146.7	62	3.69
Merc 230	22.8	4	140.8	95	3.92
Merc 280	19.2	6	167.6	123	3.92
Merc 280C	17.8	6	167.6	123	3.92
Merc 450SE	16.4	8	275.8	180	3.07
Merc 450SL	17.3	8	275.8	180	3.07

IMMUNE
CODING INSTITUTE

Semi-Structured Data

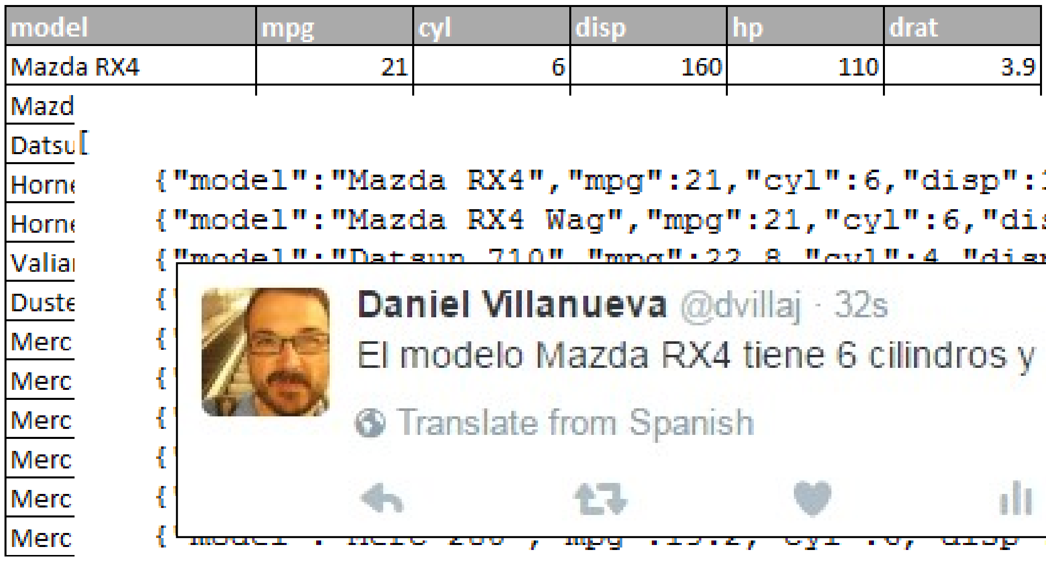


model	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazd					
Datsu					
Horn					
Horn					
Valia					
Duste					
Merc					
Merc					
Merc					
Merc					
Merc					
Merc					

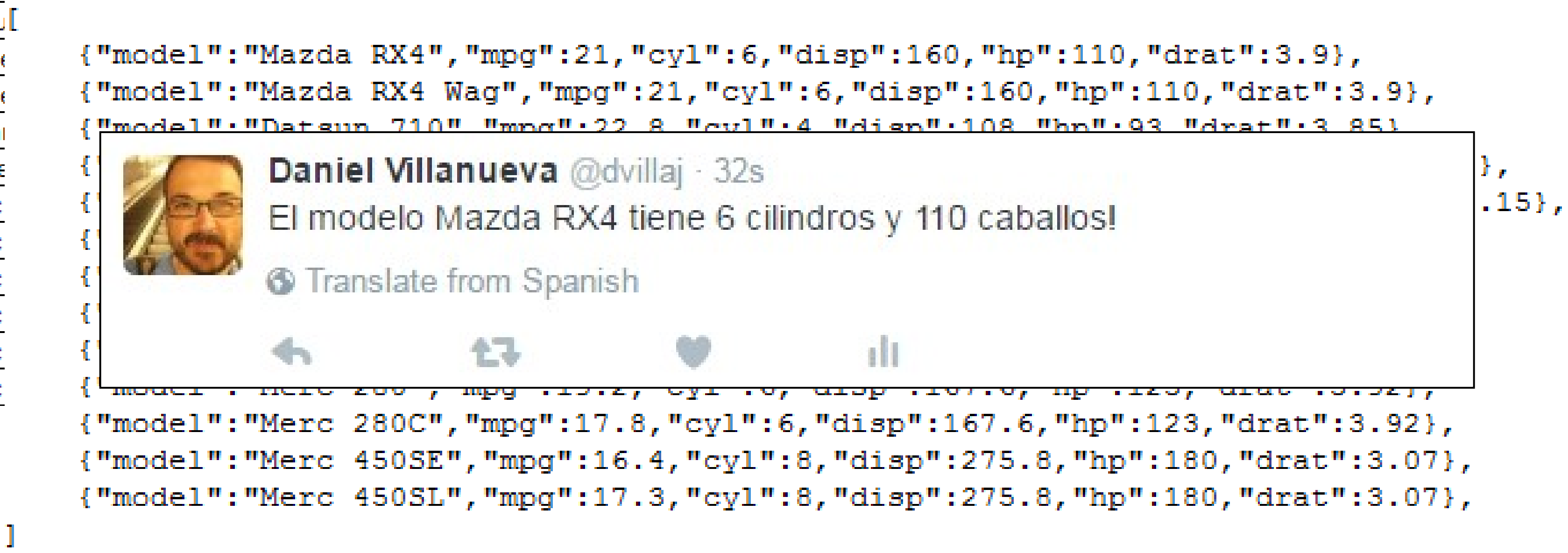
```
[
    {"model":"Mazda RX4","mpg":21,"cyl":6,"disp":160,"hp":110,"drat":3.9},
    {"model":"Mazda RX4 Wag","mpg":21,"cyl":6,"disp":160,"hp":110,"drat":3.9},
    {"model":"Datsun 710","mpg":22.8,"cyl":4,"disp":108,"hp":93,"drat":3.85},
    {"model":"Hornet 4 Drive","mpg":21.4,"cyl":6,"disp":258,"hp":110,"drat":3.08},
    {"model":"Hornet Sportabout","mpg":18.7,"cyl":8,"disp":360,"hp":175,"drat":3.15},
    {"model":"Valiant","mpg":18.1,"cyl":6,"disp":225,"hp":105,"drat":2.76},
    {"model":"Duster 360","mpg":14.3,"cyl":8,"disp":360,"hp":245,"drat":3.21},
    {"model":"Merc 240D","mpg":24.4,"cyl":4,"disp":146.7,"hp":62,"drat":3.69},
    {"model":"Merc 230","mpg":22.8,"cyl":4,"disp":140.8,"hp":95,"drat":3.92},
    {"model":"Merc 280","mpg":19.2,"cyl":6,"disp":167.6,"hp":123,"drat":3.92},
    {"model":"Merc 280C","mpg":17.8,"cyl":6,"disp":167.6,"hp":123,"drat":3.92},
    {"model":"Merc 450SE","mpg":16.4,"cyl":8,"disp":275.8,"hp":180,"drat":3.07},
    {"model":"Merc 450SL","mpg":17.3,"cyl":8,"disp":275.8,"hp":180,"drat":3.07},
]
```

IMMUNE CODING INSTITUTE

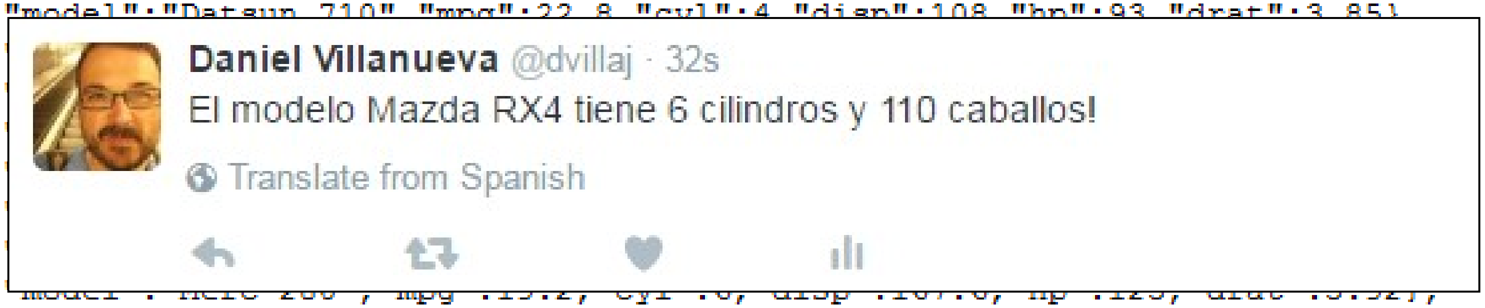
Unstructured Data



model	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazd					
Datsu					
Horne					
Horne					
Valiar					
Duste					
Merc					
Merc					
Merc					
Merc					
Merc					
Merc					



```
[
    {"model":"Mazda RX4","mpg":21,"cyl":6,"disp":160,"hp":110,"drat":3.9},
    {"model":"Mazda RX4 Wag","mpg":21,"cyl":6,"disp":160,"hp":110,"drat":3.9},
    {"model":"Datsun 710","mpg":22.8,"cyl":4,"disp":108,"hp":93,"drat":3.85},
    {                                                                       },
    {                                                                    .15},
    {
    {
    {
    {
    {"model":"Merc 280","mpg":19.2,"cyl":6,"disp":167.6,"hp":123,"drat":3.92},
    {"model":"Merc 280C","mpg":17.8,"cyl":6,"disp":167.6,"hp":123,"drat":3.92},
    {"model":"Merc 450SE","mpg":16.4,"cyl":8,"disp":275.8,"hp":180,"drat":3.07},
    {"model":"Merc 450SL","mpg":17.3,"cyl":8,"disp":275.8,"hp":180,"drat":3.07},
]
```

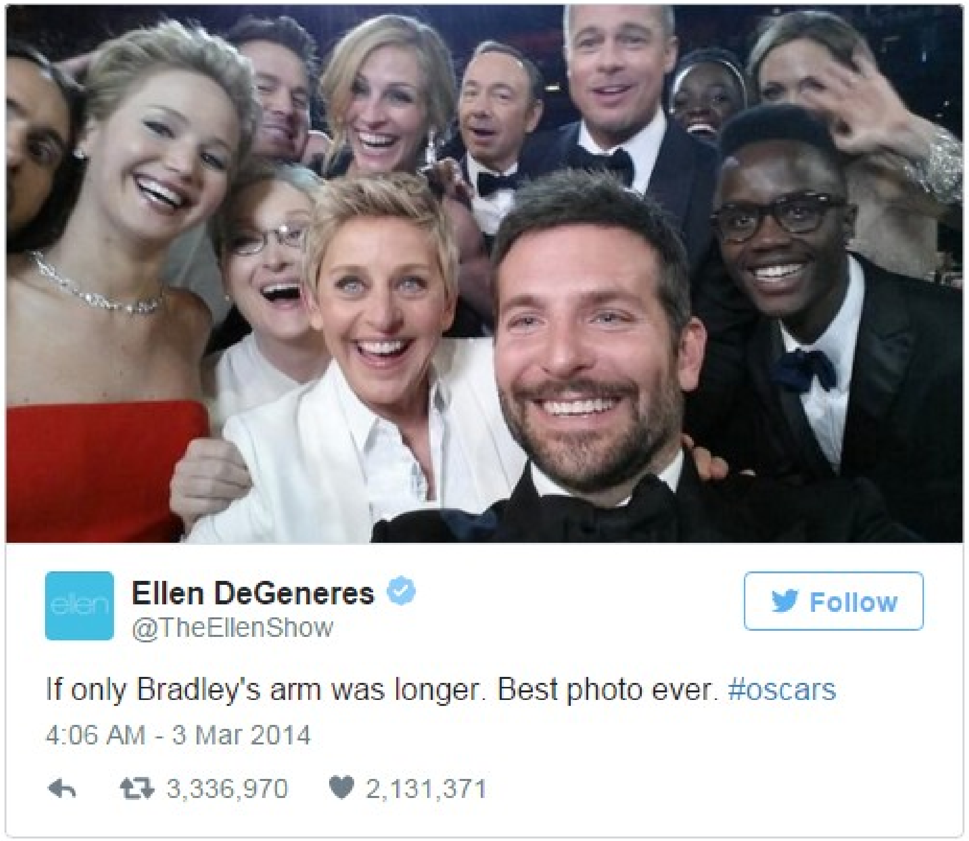


Daniel Villanueva @dvillaj · 32s

El modelo Mazda RX4 tiene 6 cilindros y 110 caballos!

IMMUNE CODING INSTITUTE

Velocity



Ellen DeGeneres
@TheEllenShow

Follow

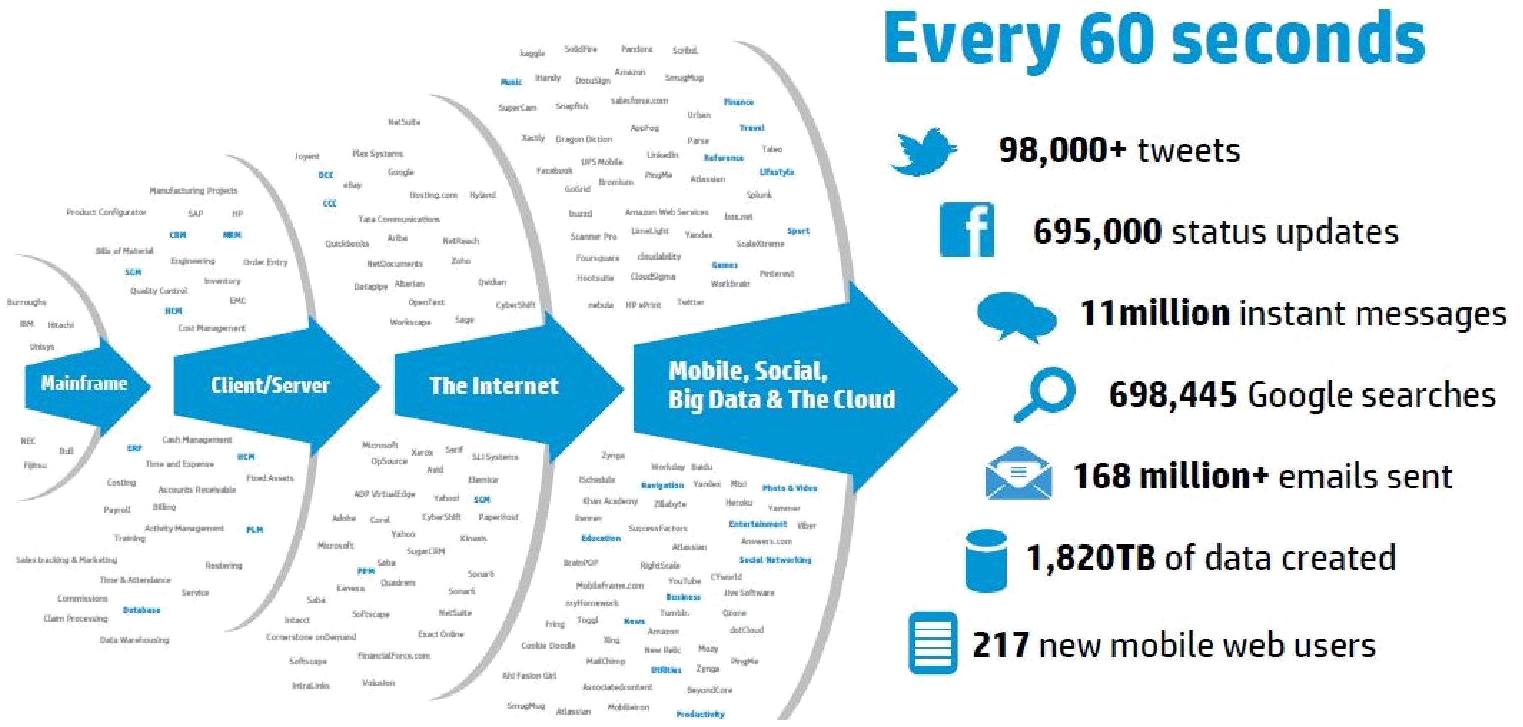
If only Bradley's arm was longer. Best photo ever. #oscars

4:06 AM - 3 Mar 2014

3,336,970 2,131,371

Tweeting

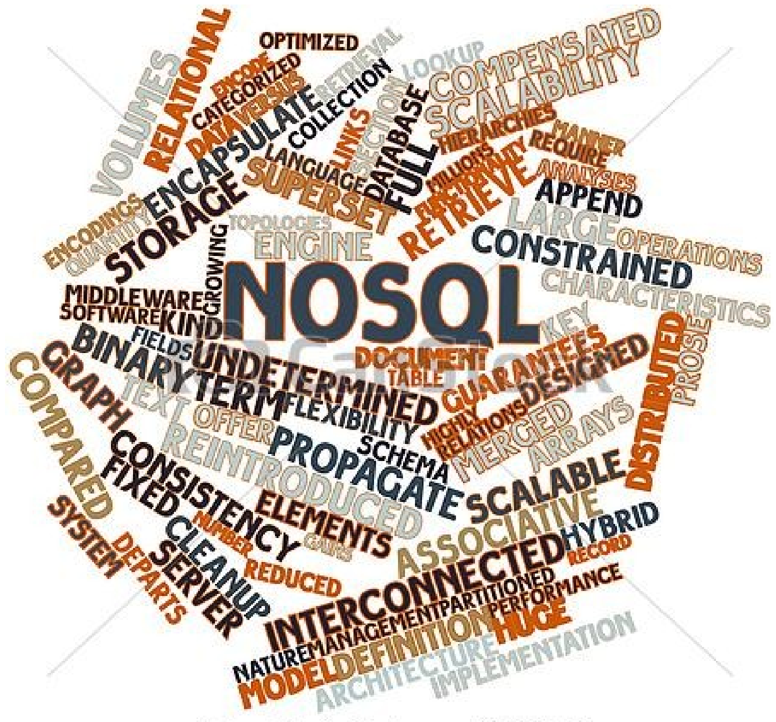
Velocity


Mainframe
Client/Server
The Internet
Mobile, Social, Big Data & The Cloud
Every 60 seconds
98,000+ tweets
695,000 status updates
11million instant messages
698,445 Google searches
168 million+ emails sent
1,820TB of data created
217 new mobile web users

IMMUNE CODING INSTITUTE

Agenda

- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- Apache Cassandra
- MongoDB
- Neo4j



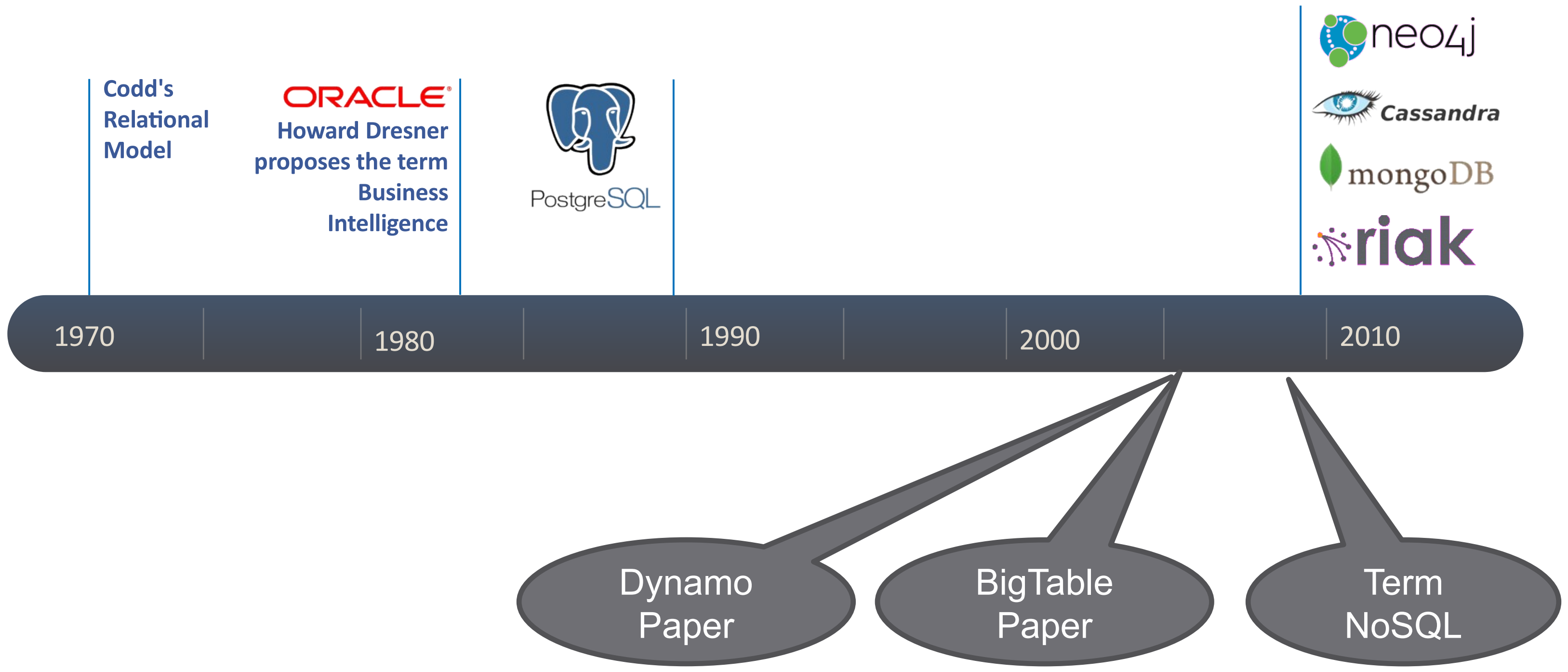
IMMUNE CODING INSTITUTE

NoSQL

NoSQL is a broad class of **database management systems** that differs from the classic model of the relational database management system

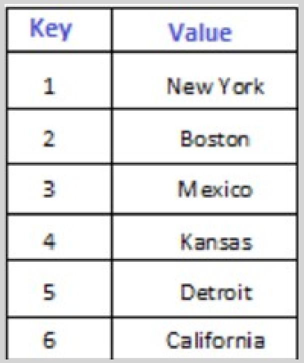
- They usually **scale** well horizontally
- Do not use **SQL** as the main query language
- Stored data does not require **fixed structures** such as tables
- Normally do not support **JOIN** operations
- Not fully guaranteed by **ACID**
- Many of them are **Open Source**

A little bit of history



Types of NoSQL databases

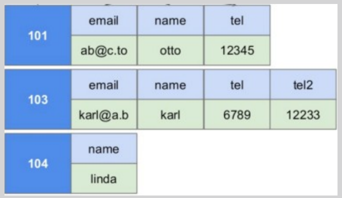
Key / Value



Key	Value
1	New York
2	Boston
3	Mexico
4	Kansas
5	Detroit
6	California

riak

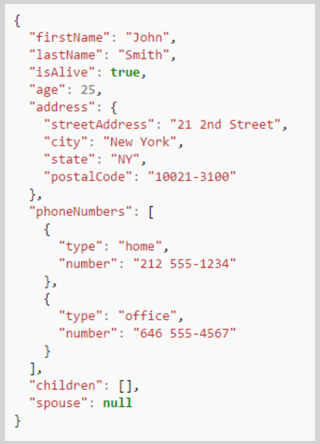
Columnar



101	email	name	tel	
	ab@c.to	otto	12345	
103	email	name	tel	tel2
	karl@a.b	karl	6789	12233
104	name			
	linda			

Cassandra

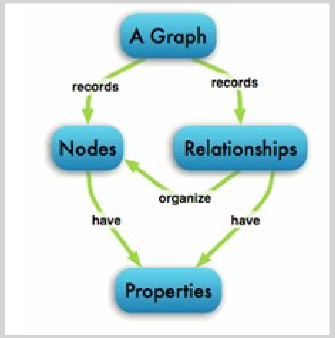
Documents



```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

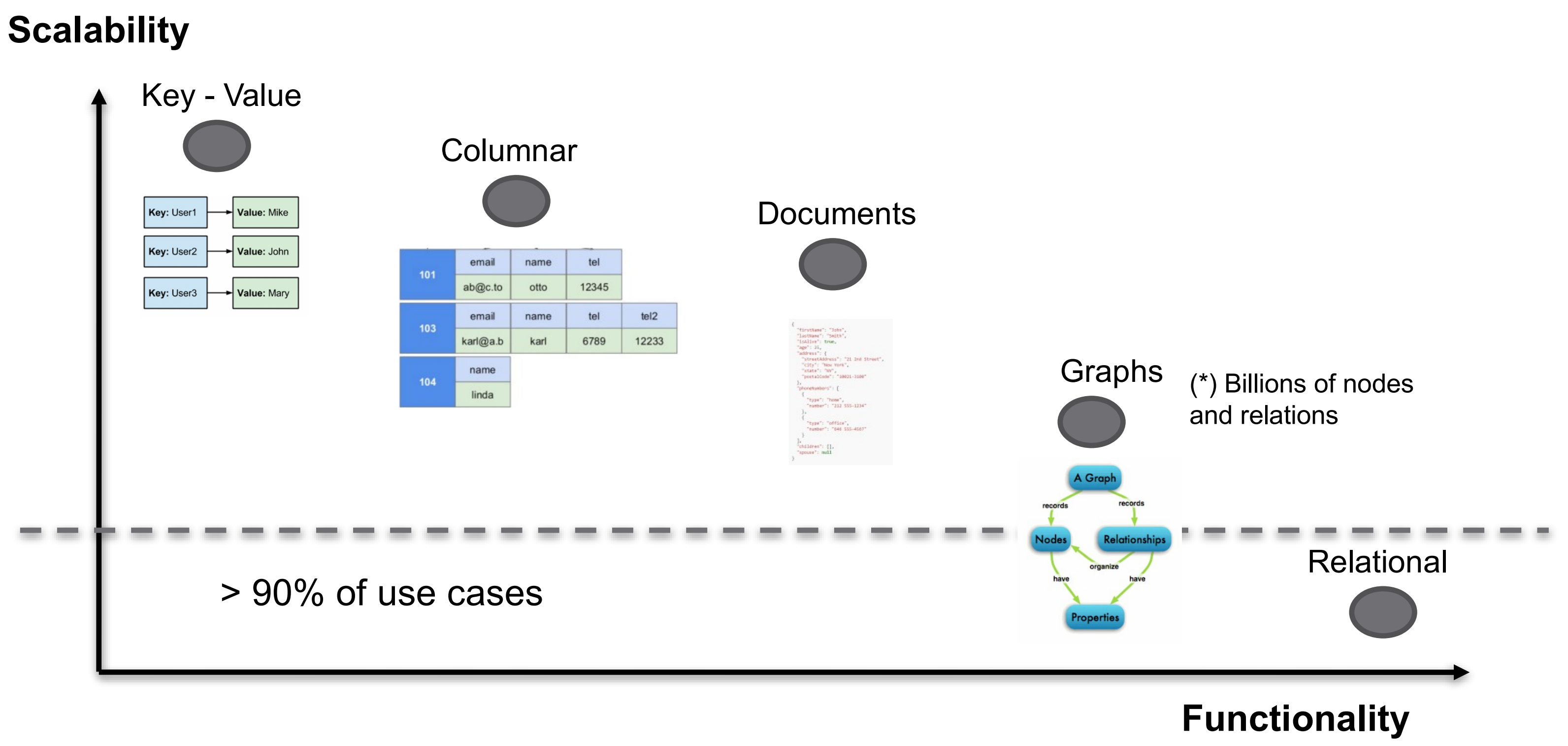
mongoDB

Graphs



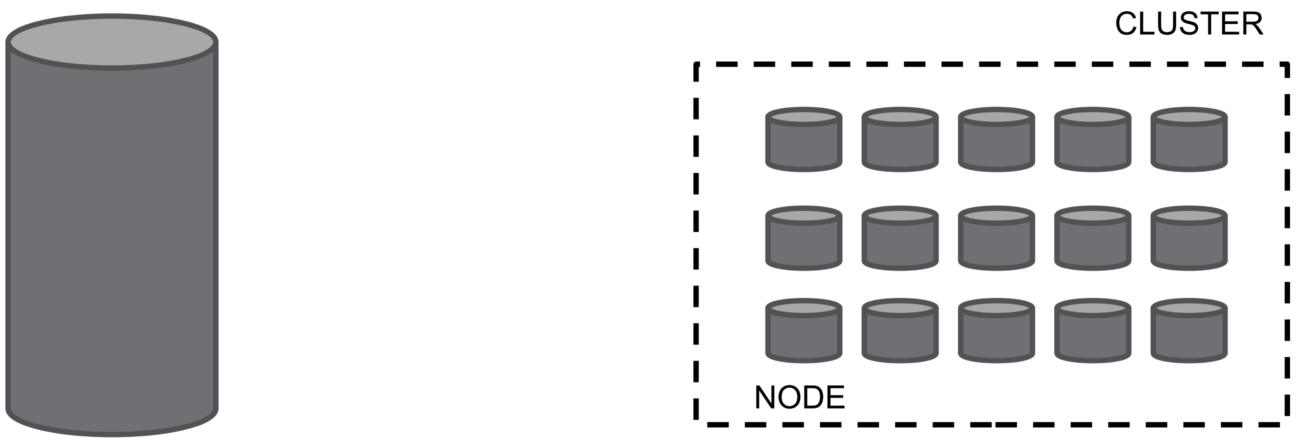
neo4j

Size vs. Functionality



IMMUNE CODING INSTITUTE

Vertical and horizontal scaling

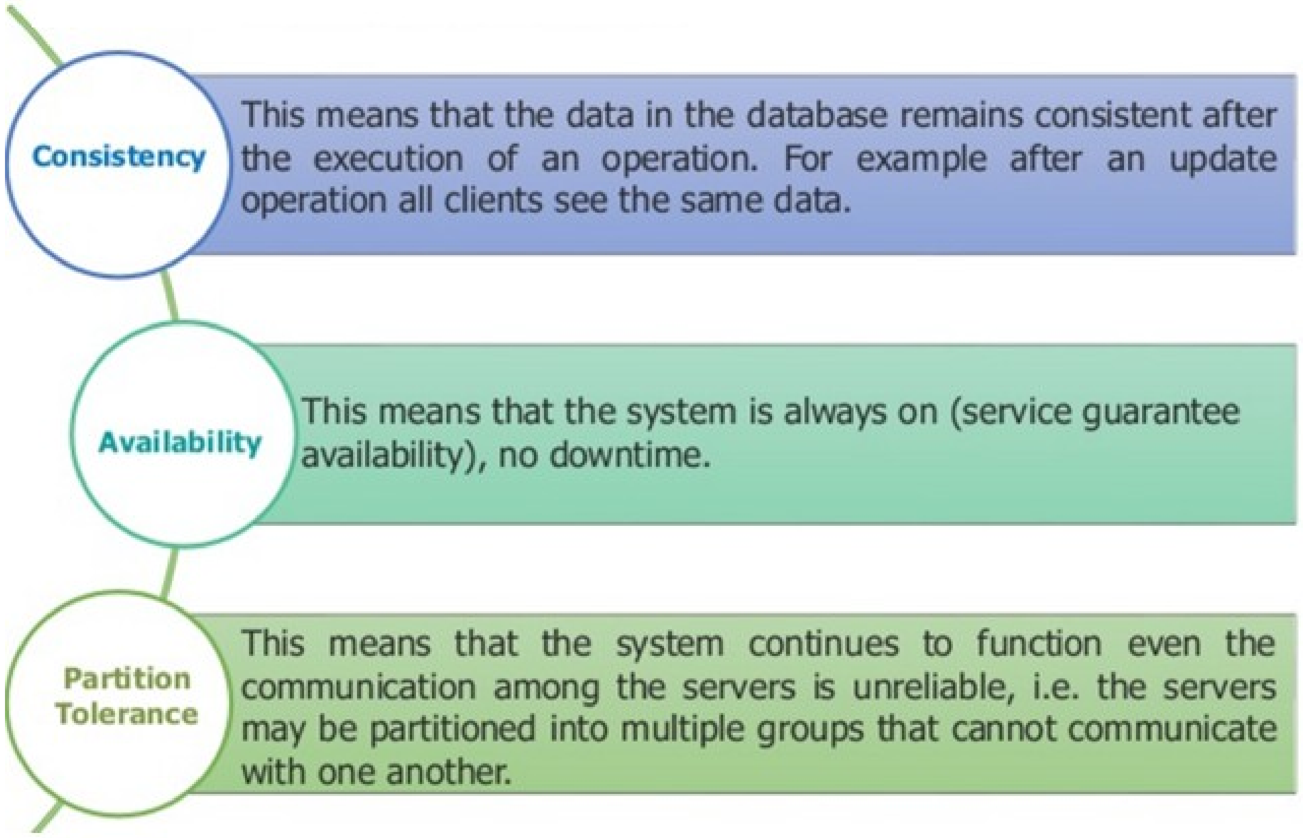


VERTICAL SCALING

HORIZONTAL SCALING

CAP Theorem

- Requirements for distributed databases

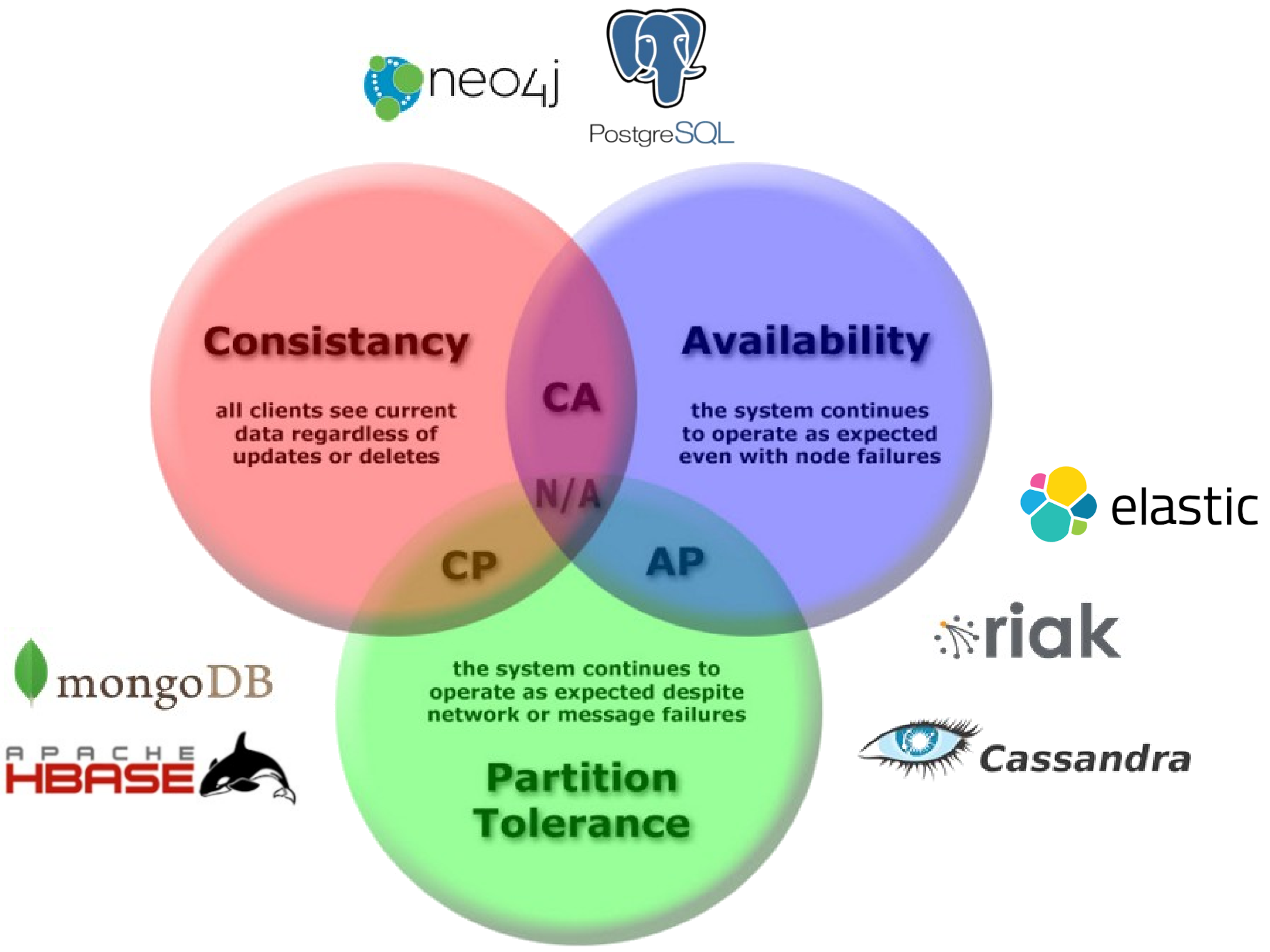


This means that the data in the database remains consistent after the execution of an operation. For example after an update operation all clients see the same data.

This means that the system is always on (service guarantee availability), no downtime.

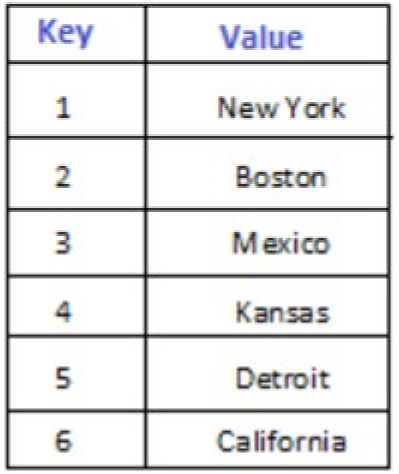
This means that the system continues to function even the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

CAP Theorem


neo4j
PostgreSQL
Consistancy
all clients see current data regardless of updates or deletes
Availability
the system continues to operate as expected even with node failures
CA
N/A
CP
AP
the system continues to operate as expected despite network or message failures
Partition Tolerance
elastic
riak
Cassandra
mongoDB
APACHE HBASE

Agenda

- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- Apache Cassandra
- MongoDB
- Neo4j

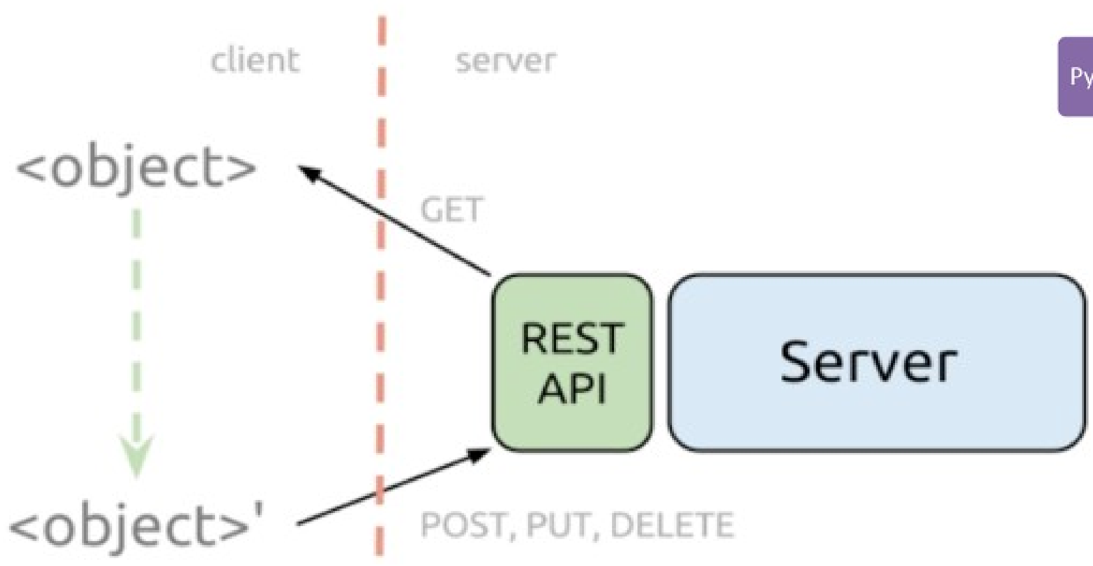


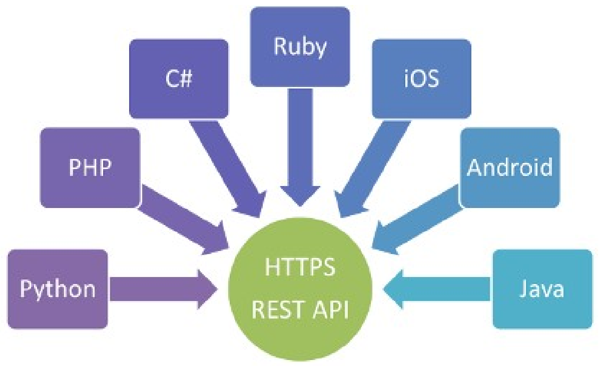
Key	Value
1	New York
2	Boston
3	Mexico
4	Kansas
5	Detroit
6	California

Riak

- Developed by Basho Technologies in Erlang
- Inspired by Amazon Dynamo
- Horizontal Distribution - Fault Tolerant
- Prioritizes availability - Tunable consistency
- No master node - No single point of failure
- Querys - Provides a REST API over HTTP
- Drivers in multiple languages - Java, Python, Ruby, etc.
- Storage options - Memory, disk or both.

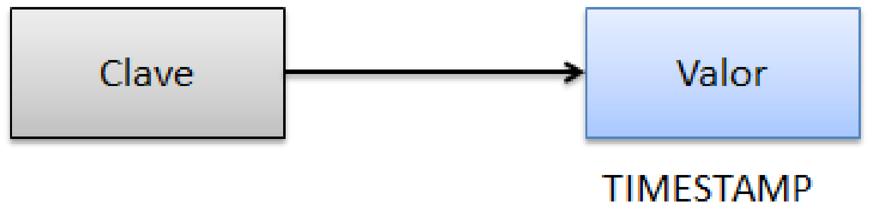
API REST


client
server
<object>
GET
REST API
Server
<object>'
POST, PUT, DELETE


Ruby
C#
iOS
PHP
Android
Python
HTTPS REST API
Java

Key and Value

- It's the most basic structure

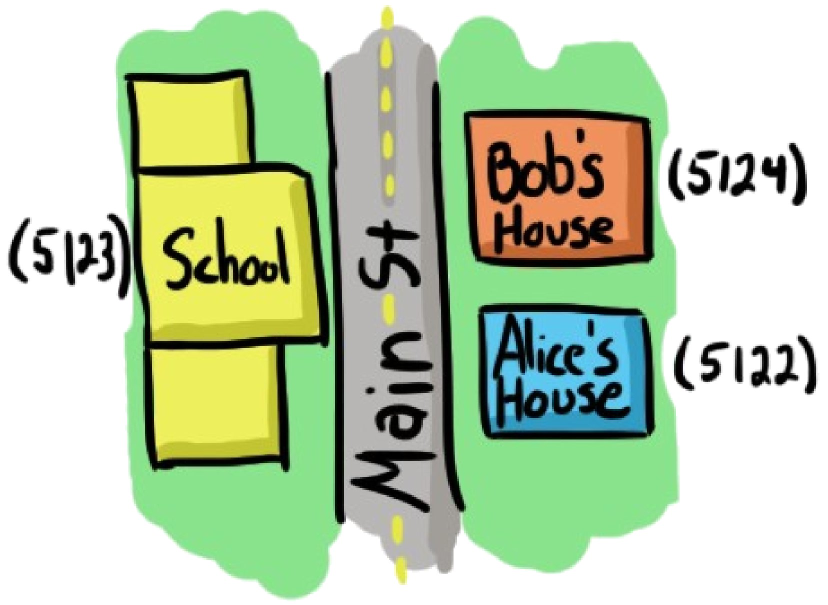


Key and Value

- For example:
  - Key: Address
  - Value: Tenant

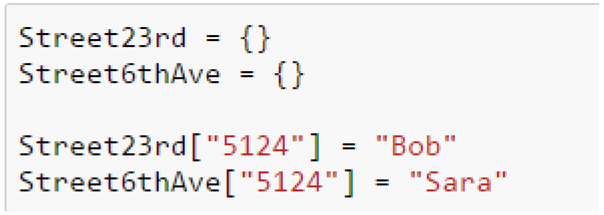
```
hashtable = {}
hashtable["5124"] = "Bob"

print hashtable["5124"]
```

```
Bob
```



Buckets

- They allow to separate the same key according to a context
- Example: Streets



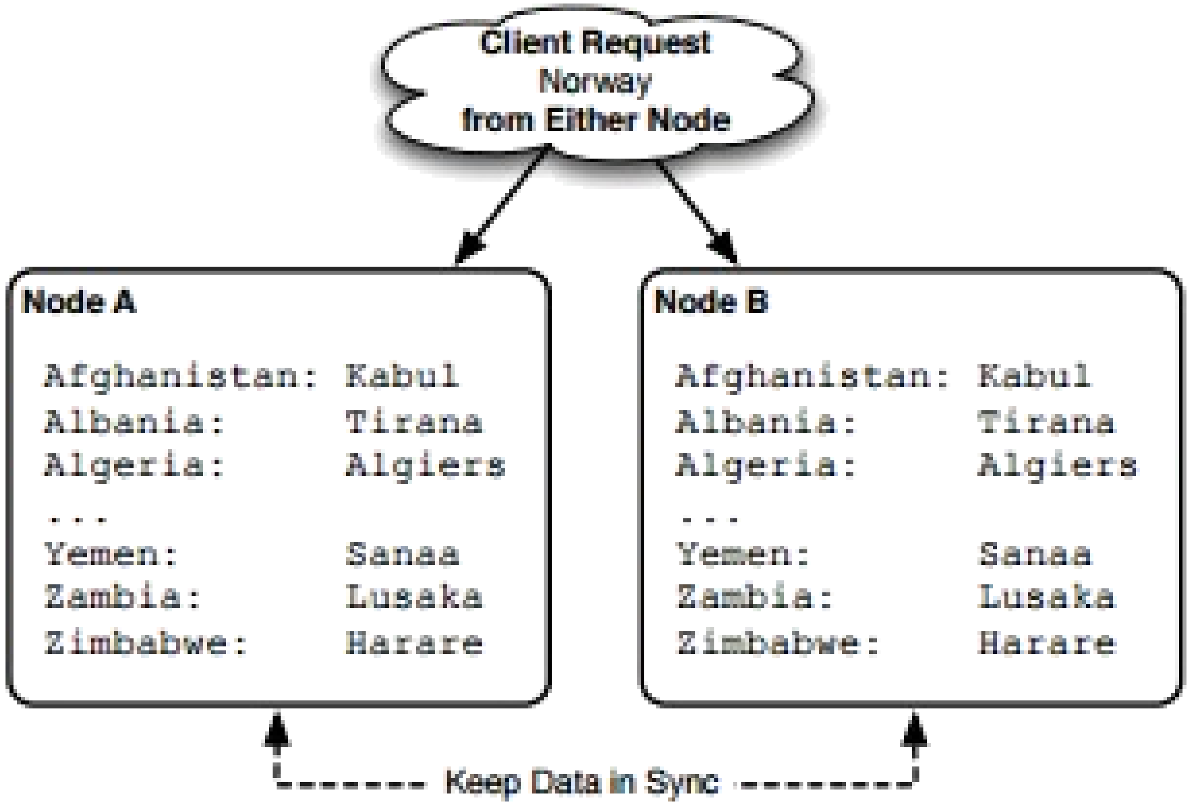
```
Street23rd = {}
Street6thAve = {}

Street23rd["5124"] = "Bob"
Street6thAve["5124"] = "Sara"
```

Data distribution in Riak

- Riak is kept available by distributing the data between different nodes
- There are 2 styles of layout ...

Data distribution in Riak

- Replication

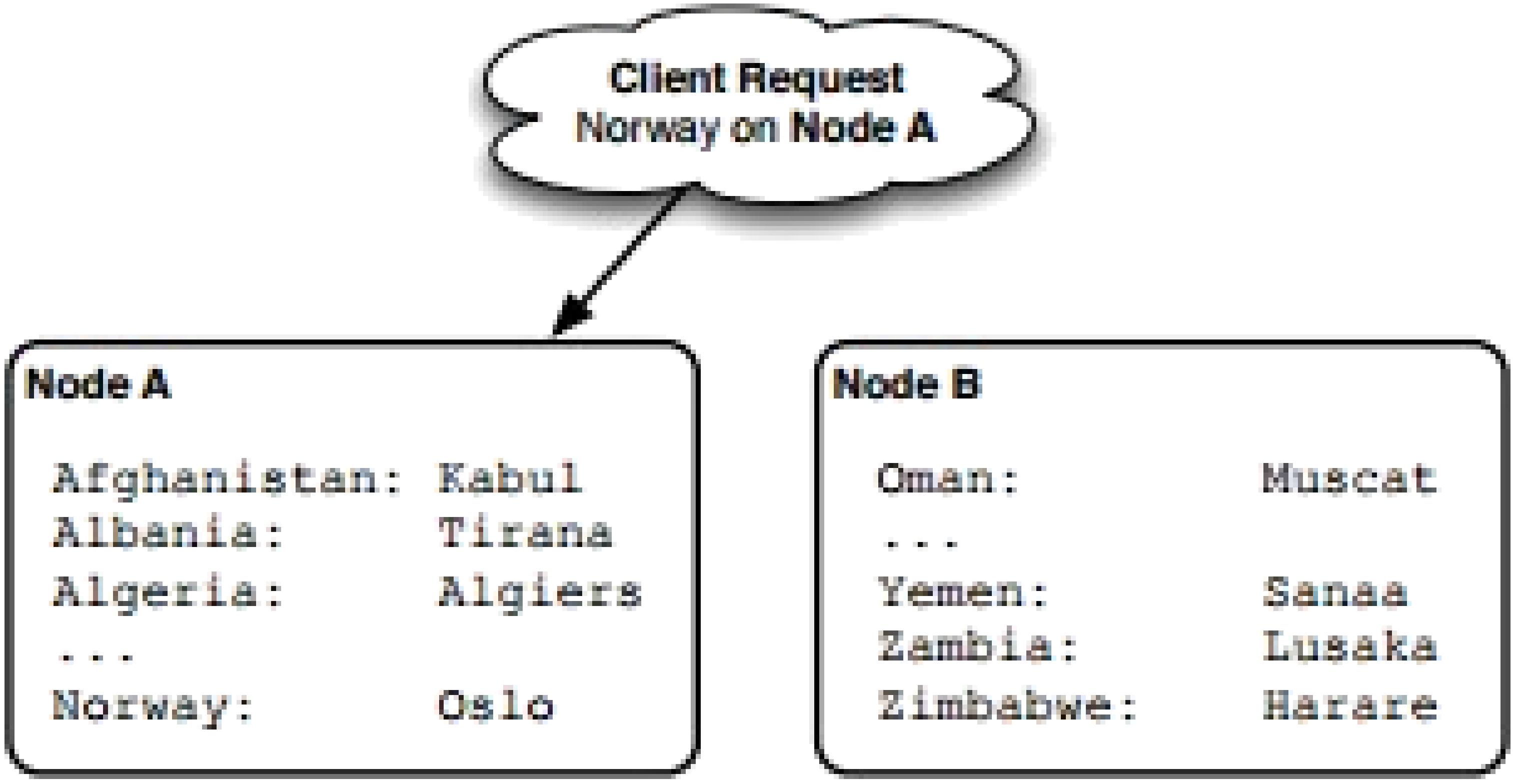


```
Afghanistan: Kabul
Albania:     Tirana
Algeria:     Algiers
...
Yemen:       Sanaa
Zambia:      Lusaka
Zimbabwe:    Harare
```

```
Afghanistan: Kabul
Albania:     Tirana
Algeria:     Algiers
...
Yemen:       Sanaa
Zambia:      Lusaka
Zimbabwe:    Harare
```

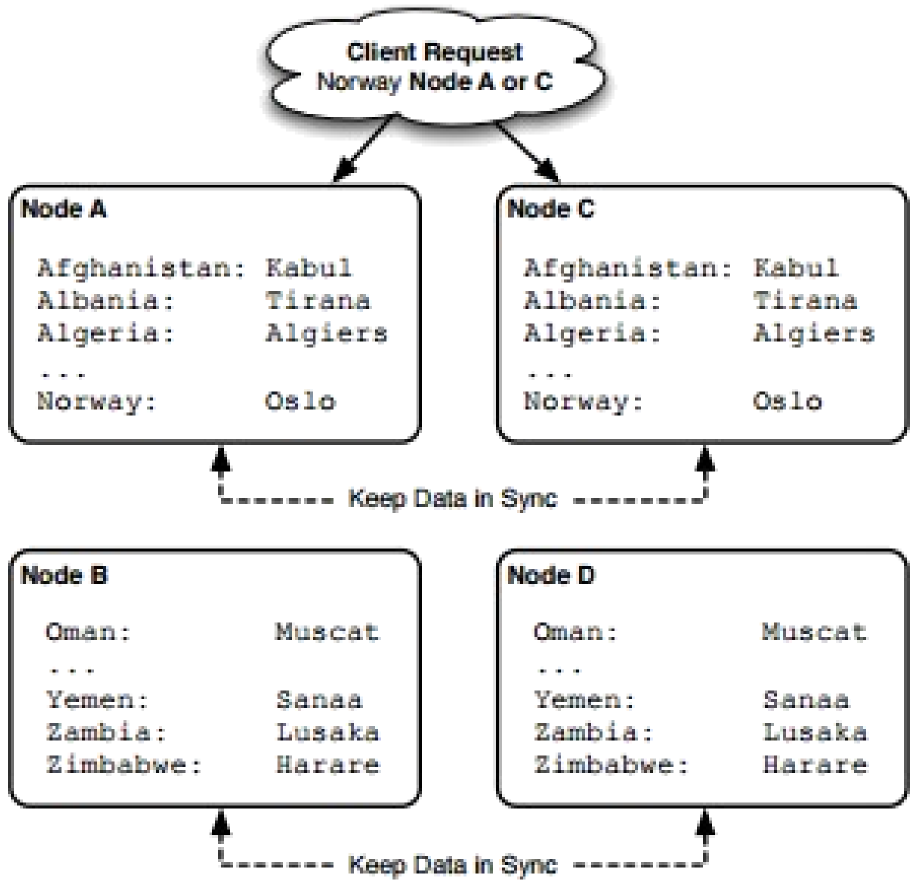
Data distribution in Riak

- Partitioned

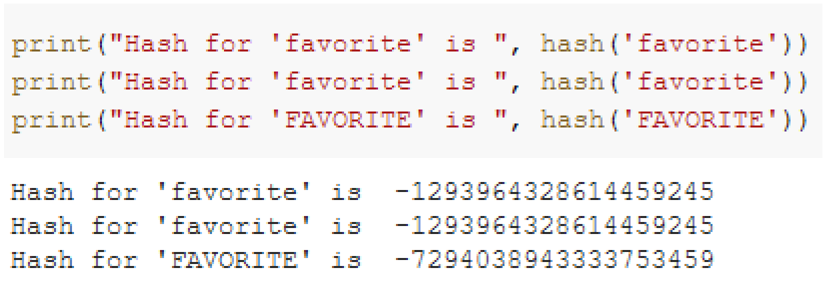


Data distribution in Riak

- Riak uses Replication + Partitioning



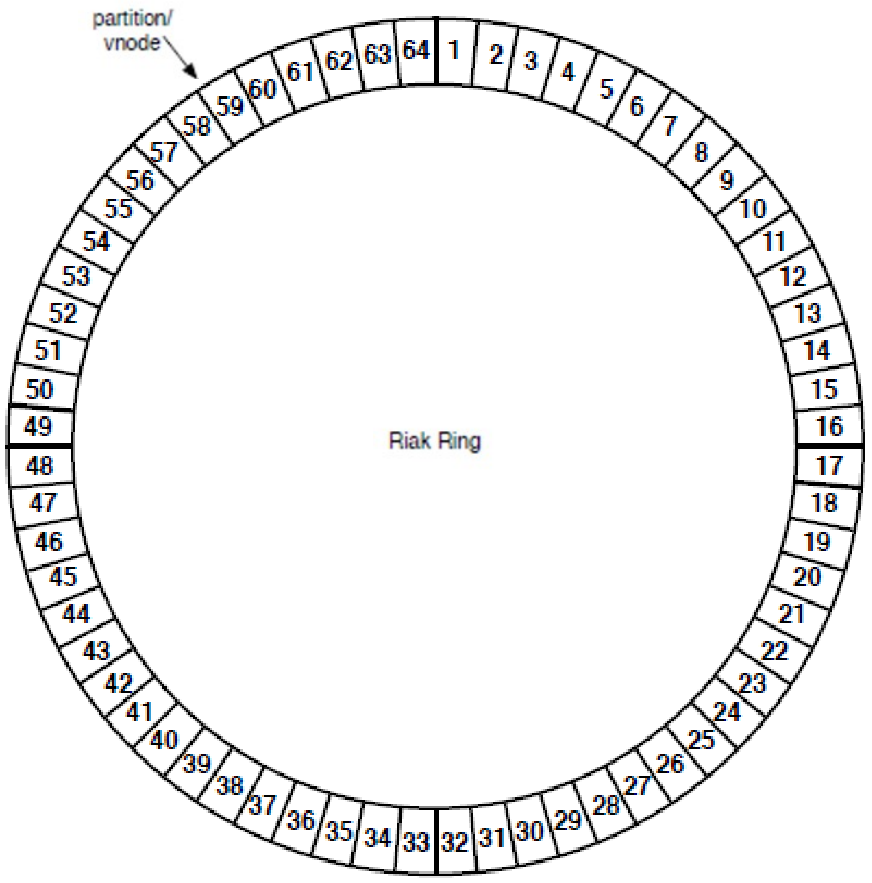
Hash function



```python
print("Hash for 'favorite' is ", hash('favorite'))
print("Hash for 'favorite' is ", hash('favorite'))
print("Hash for 'FAVORITE' is ", hash('FAVORITE'))
```

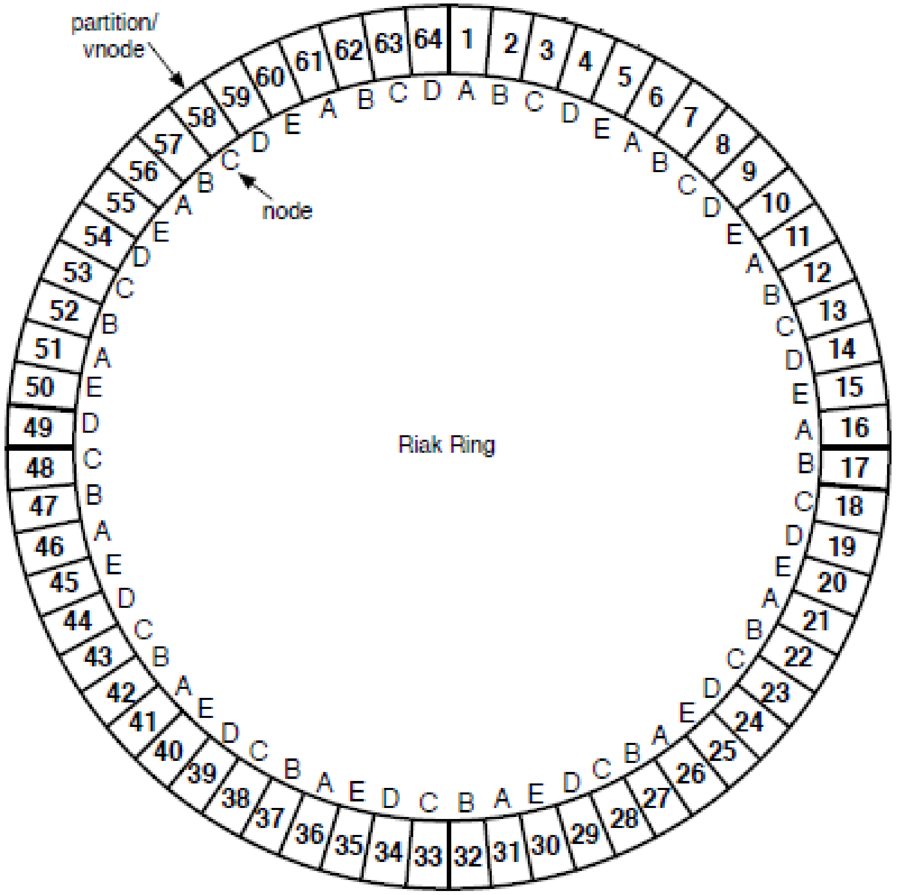
```
Hash for 'favorite' is  -1293964328614459245
Hash for 'favorite' is  -1293964328614459245
Hash for 'FAVORITE' is  -7294038943333753459
```

The Riak Ring – The Cluster

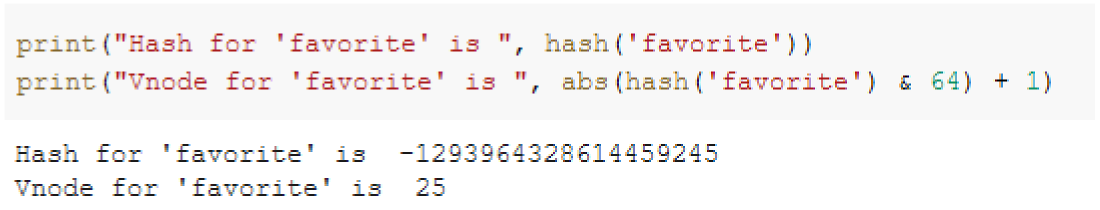


IMMUNE CODING INSTITUTE

The Riak Ring – The Cluster



Hash function

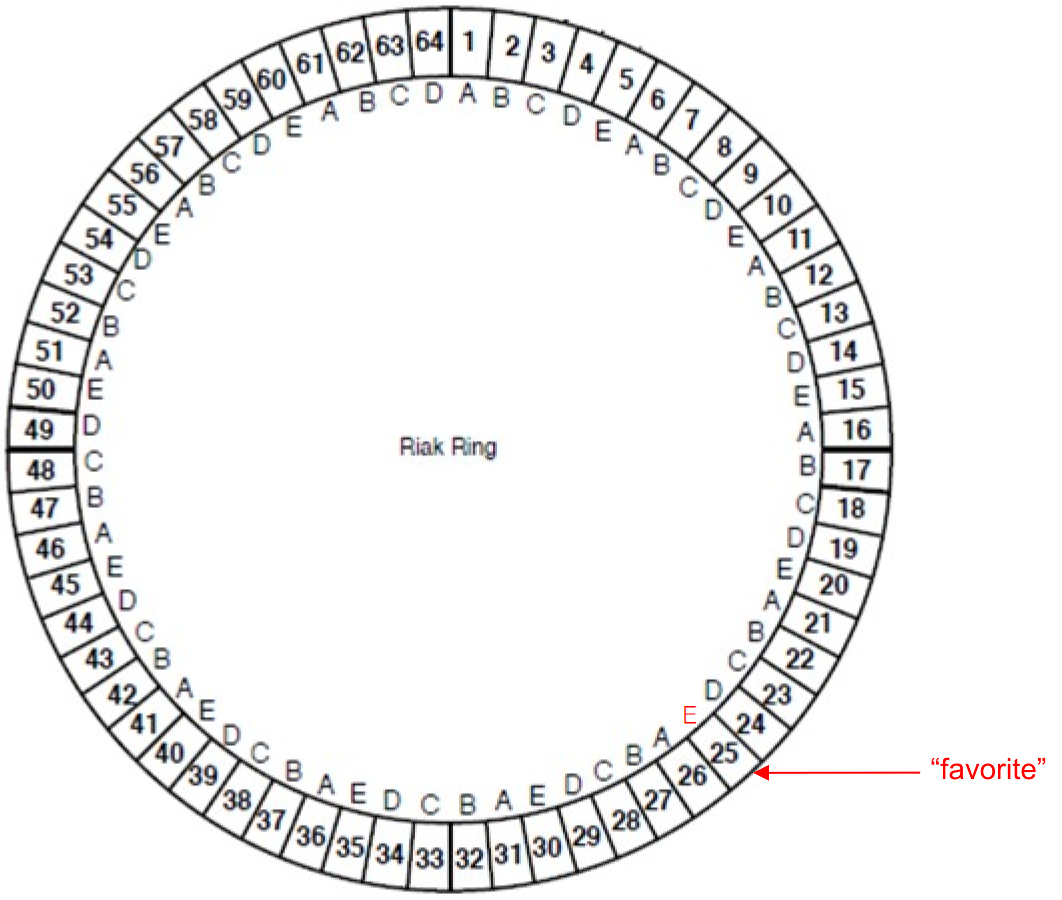


```python
print("Hash for 'favorite' is ", hash('favorite'))
print("Vnode for 'favorite' is ", abs(hash('favorite') & 64) + 1)
```

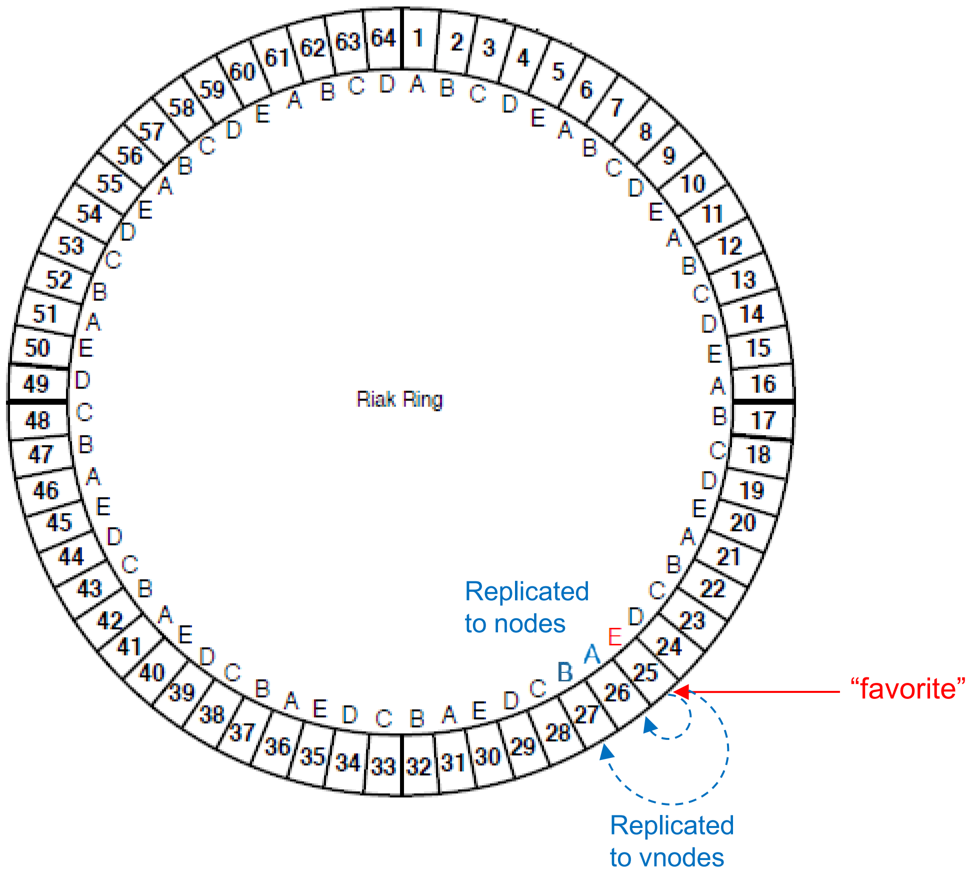
```
Hash for 'favorite' is  -1293964328614459245
Vnode for 'favorite' is  25
```

IMMUNE CODING INSTITUTE

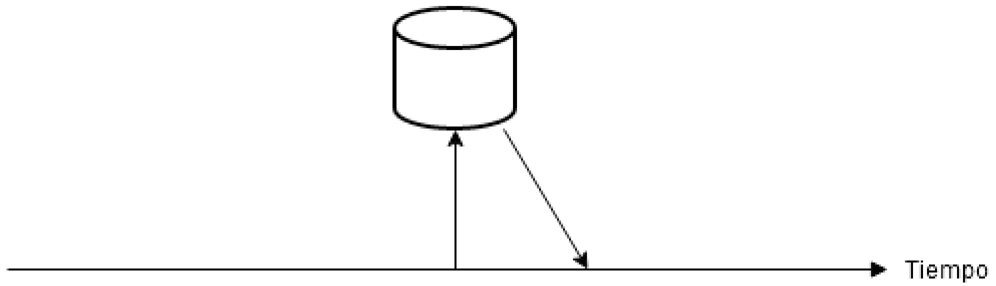
The Riak Ring – Replication



The Riak Ring – Replication

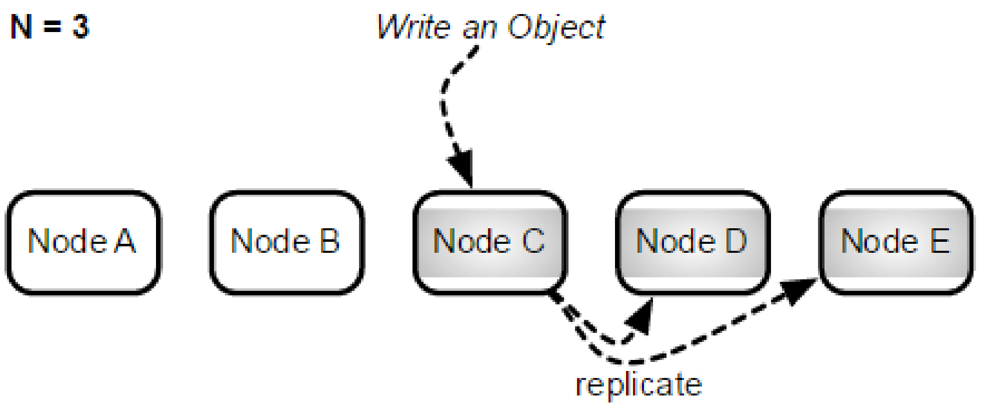


Consistence vs Disponibility



N/R/W

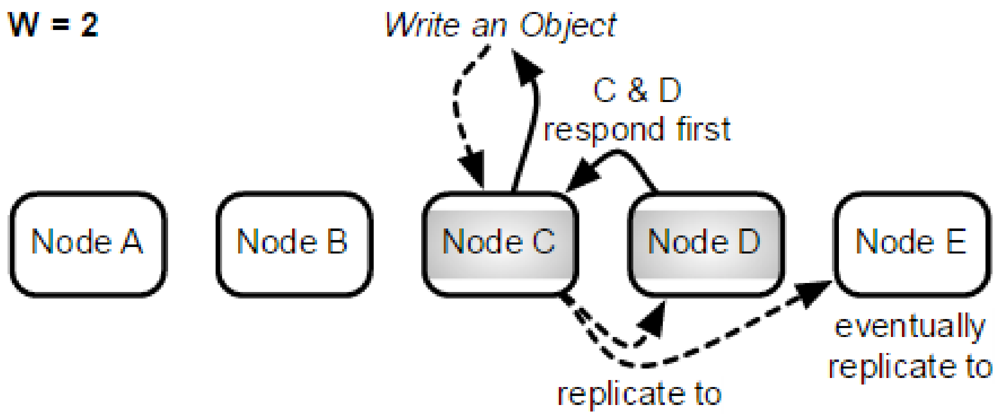
- N - Number of nodes in which the information is replicated



IMMUNE CODING INSTITUTE

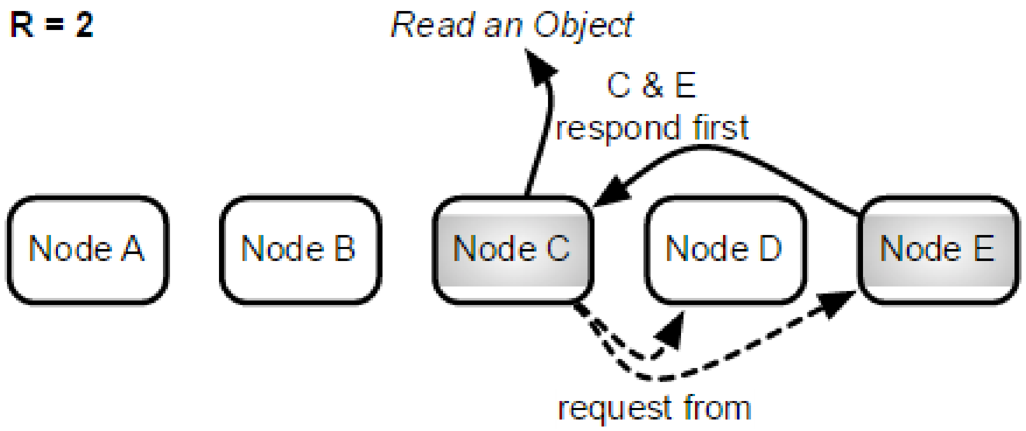
N/R/W

- W - Number of nodes to be written to before the operation is considered successful

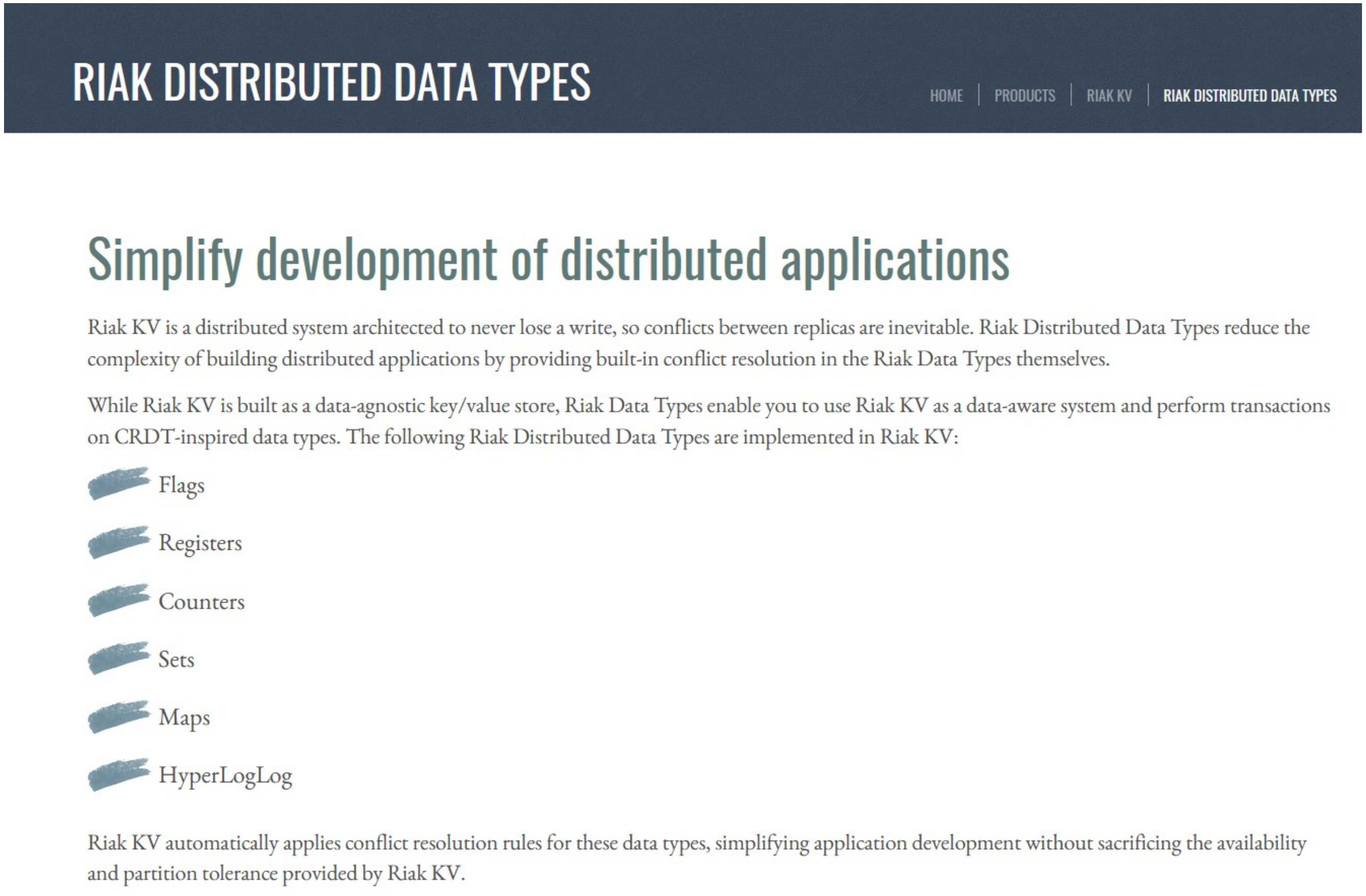


N/R/W

- R - Number of nodes to be read from before returning the value



Riak Data Types



RIAK DISTRIBUTED DATA TYPES

HOME | PRODUCTS | RIAK KV | RIAK DISTRIBUTED DATA TYPES

Simplify development of distributed applications

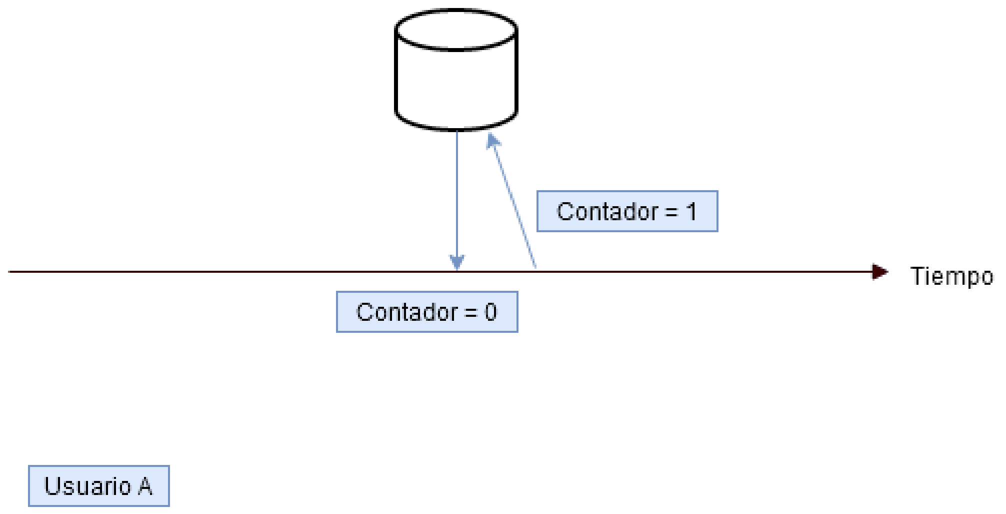
Riak KV is a distributed system architected to never lose a write, so conflicts between replicas are inevitable. Riak Distributed Data Types reduce the complexity of building distributed applications by providing built-in conflict resolution in the Riak Data Types themselves.

While Riak KV is built as a data-agnostic key/value store, Riak Data Types enable you to use Riak KV as a data-aware system and perform transactions on CRDT-inspired data types. The following Riak Distributed Data Types are implemented in Riak KV:

- Flags
- Registers
- Counters
- Sets
- Maps
- HyperLogLog

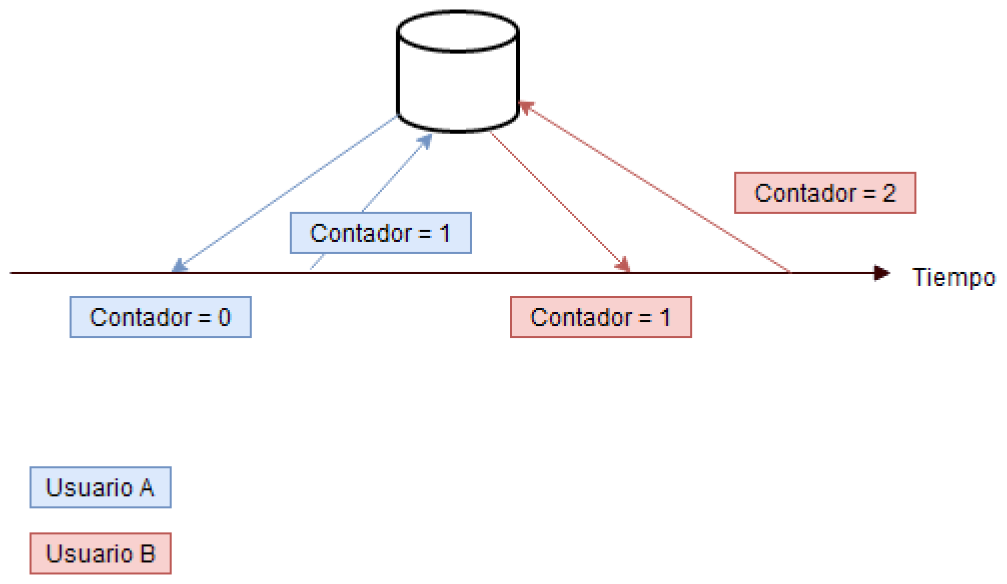
Riak KV automatically applies conflict resolution rules for these data types, simplifying application development without sacrificing the availability and partition tolerance provided by Riak KV.

Handling conflicts can be complex

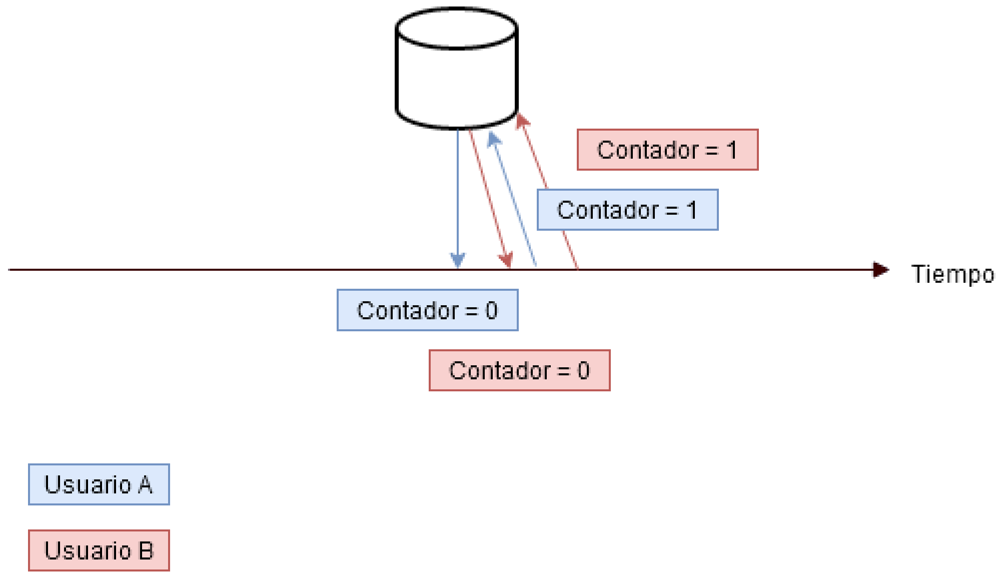


https://riak.com/products/riak-kv/riak-distributed-data-types/index.html?p=10947.html

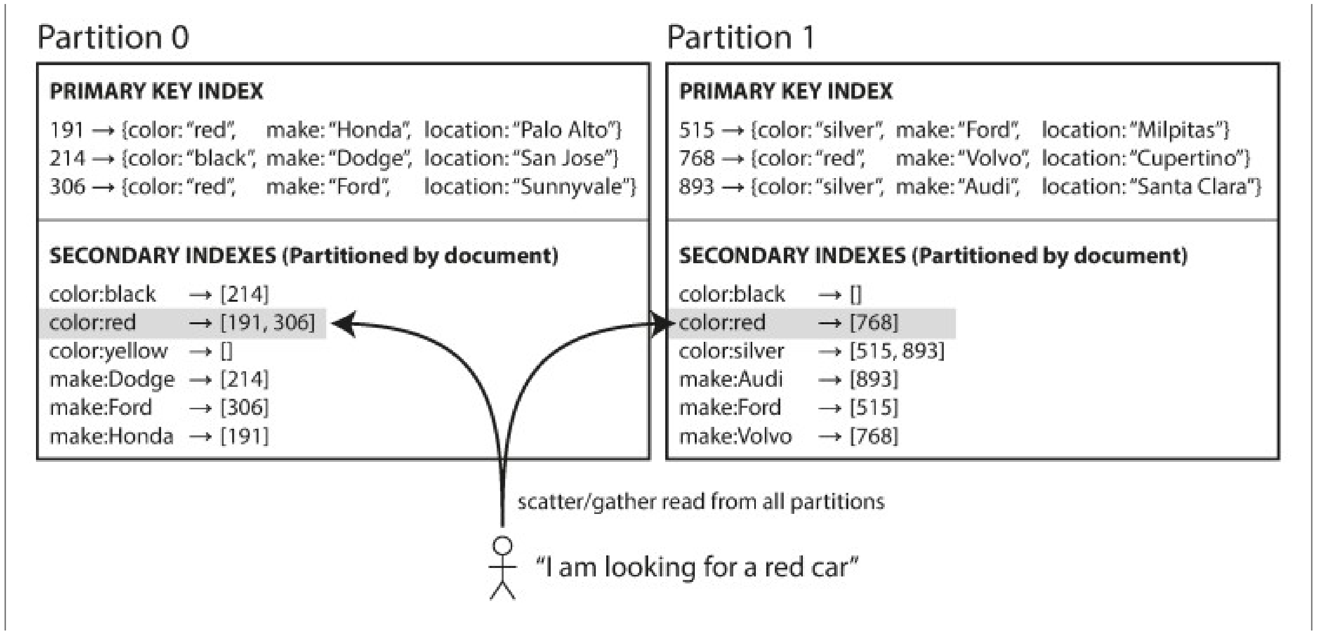
Handling conflicts can be complex



Handling conflicts can be complex



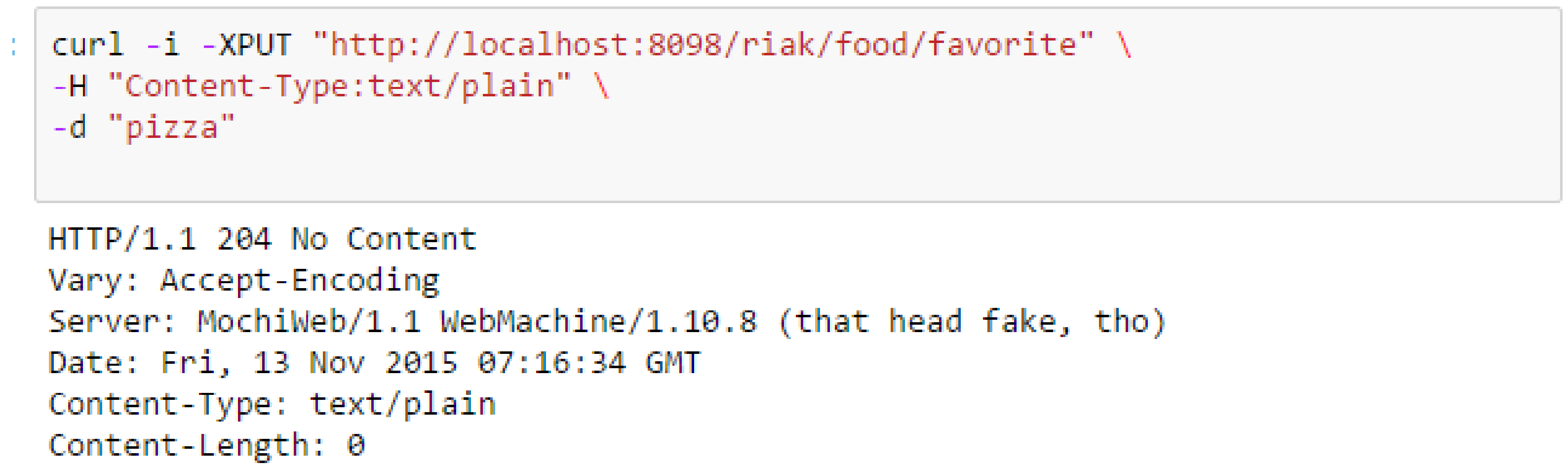
Riak Secundary Index



Partition 0
PRIMARY KEY INDEX
191 → {color: "red", make: "Honda", location: "Palo Alto"}
214 → {color: "black", make: "Dodge", location: "San Jose"}
306 → {color: "red", make: "Ford", location: "Sunnyvale"}
SECONDARY INDEXES (Partitioned by document)
color:black → [214]
color:red → [191, 306]
color:yellow → []
make:Dodge → [214]
make:Ford → [306]
make:Honda → [191]
Partition 1
PRIMARY KEY INDEX
515 → {color: "silver", make: "Ford", location: "Milpitas"}
768 → {color: "red", make: "Volvo", location: "Cupertino"}
893 → {color: "silver", make: "Audi", location: "Santa Clara"}
SECONDARY INDEXES (Partitioned by document)
color:black → []
color:red → [768]
color:silver → [515, 893]
make:Audi → [893]
make:Ford → [515]
make:Volvo → [768]
scatter/gather read from all partitions
"I am looking for a red car"

CRUD operations in Riak

- Inserting / Changing a key



```
curl -i -XPUT "http://localhost:8098/riak/food/favorite" \
-H "Content-Type:text/plain" \
-d "pizza"
```

```
HTTP/1.1 204 No Content
Vary: Accept-Encoding
Server: MochiWeb/1.1 WebMachine/1.10.8 (that head fake, tho)
Date: Fri, 13 Nov 2015 07:16:34 GMT
Content-Type: text/plain
Content-Length: 0
```

IMMUNE
CODING INSTITUTE

CRUD operations in Riak

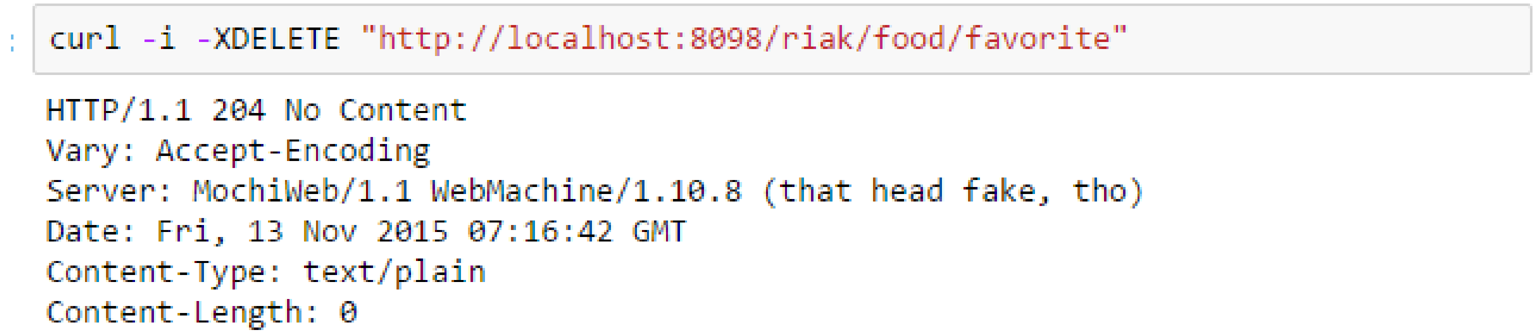
- Reading a key

```
curl -i -XGET "http://localhost:8098/riak/food/favorite"
```

```
HTTP/1.1 200 OK
X-Riak-Vclock: a85hYGBgzGDKBVI8ypz/fkY4RxxgYPx9KYMpkTGPlSGIa9d5viwA
Vary: Accept-Encoding
Server: MochiWeb/1.1 WebMachine/1.10.8 (that head fake, tho)
Link: </riak/food>; rel="up"
Last-Modified: Fri, 13 Nov 2015 07:16:34 GMT
ETag: "5eBXvxQJPMoirlTo6QeXV5"
Date: Fri, 13 Nov 2015 07:16:35 GMT
Content-Type: text/plain
Content-Length: 5

pizza
```

CRUD operations in Riak

- Deleting a key

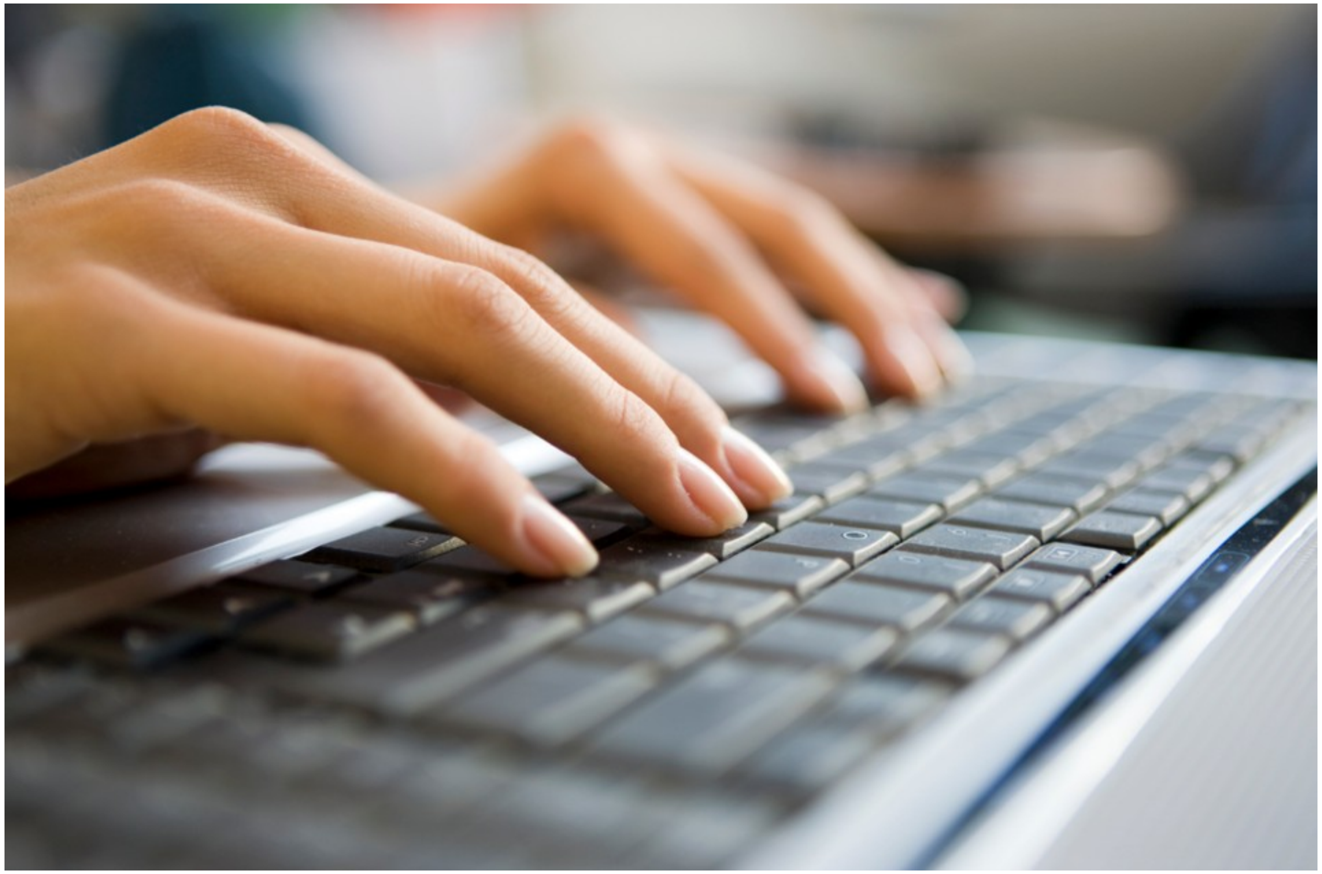


```
curl -i -XDELETE "http://localhost:8098/riak/food/favorite"
```

```
HTTP/1.1 204 No Content
Vary: Accept-Encoding
Server: MochiWeb/1.1 WebMachine/1.10.8 (that head fake, tho)
Date: Fri, 13 Nov 2015 07:16:42 GMT
Content-Type: text/plain
Content-Length: 0
```

IMMUNE CODING INSTITUTE

Riak - HandsOn

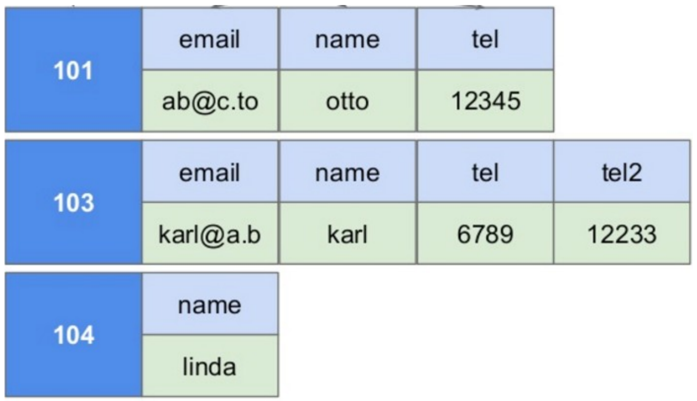


Use Cases

- Simple applications requiring high performance in read/write operations
- Applications that need the database to be always available
- Serving ads to web / mobile applications
- User preferences
- Session Storage

Agenda

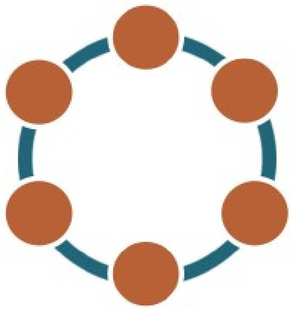
- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- Apache Cassandra
- MongoDB
- Neo4j



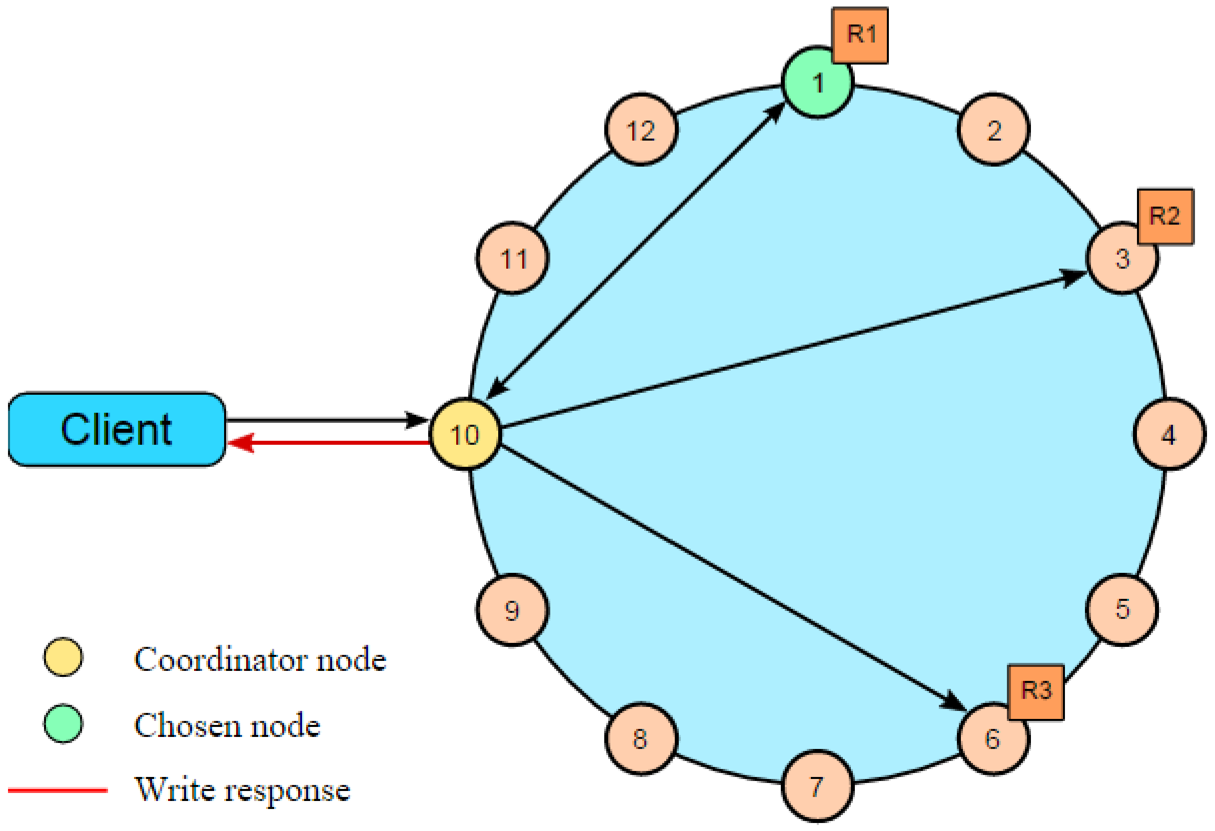
101	email	name	tel	
	ab@c.to	otto	12345	
103	email	name	tel	tel2
	karl@a.b	karl	6789	12233
104	name			
	linda			

Apache Cassandra

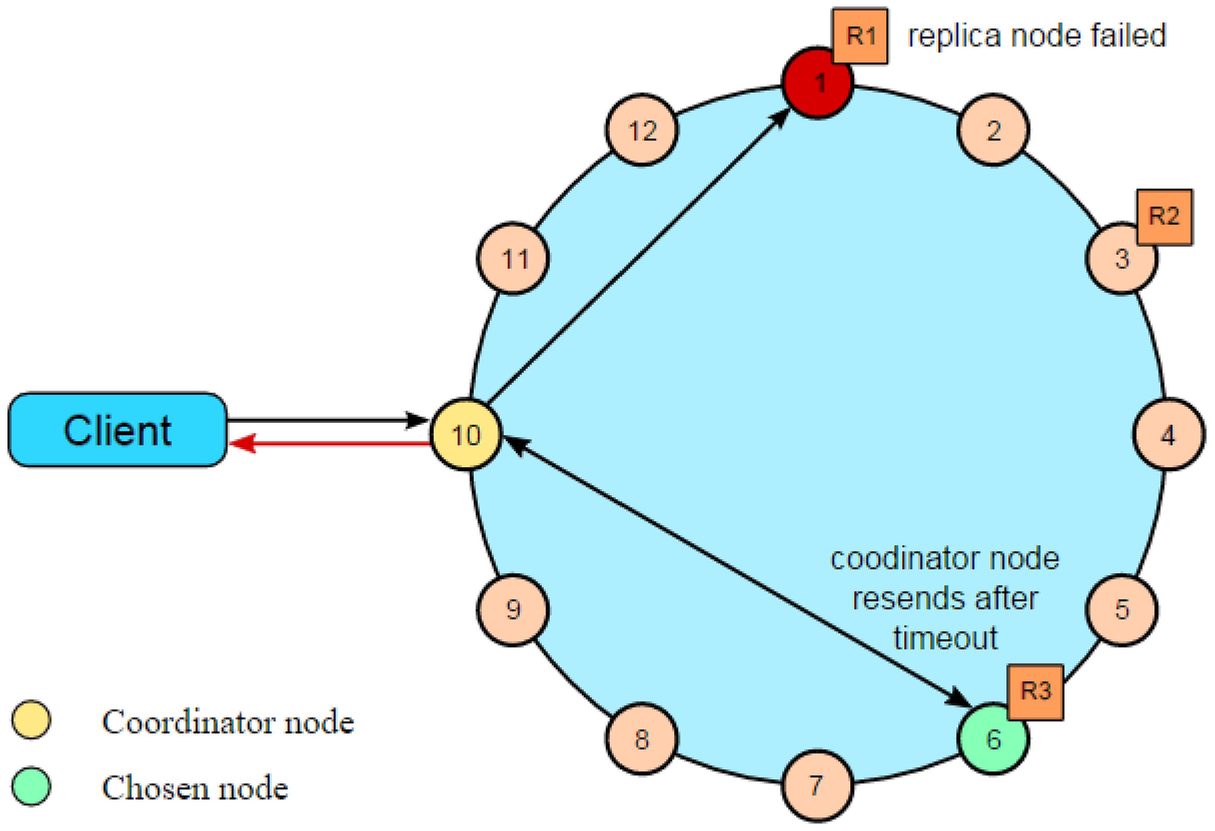
- Developed in java by Facebook and donated to the Apache Foundation in 2008
- It is based on a key/value model by storing several columns per key.
- Inspired by Amazon's Dynamo (Same as Riak) and Google's BigTable (Column families)
- There's no central controller. No single point of failure
- Querys: CQL Language - Similar to SQL
- Compatible with Hadoop and Spark
- Supports multiple data centers
- Linear scalability



Architecture - Writing (W = 3)

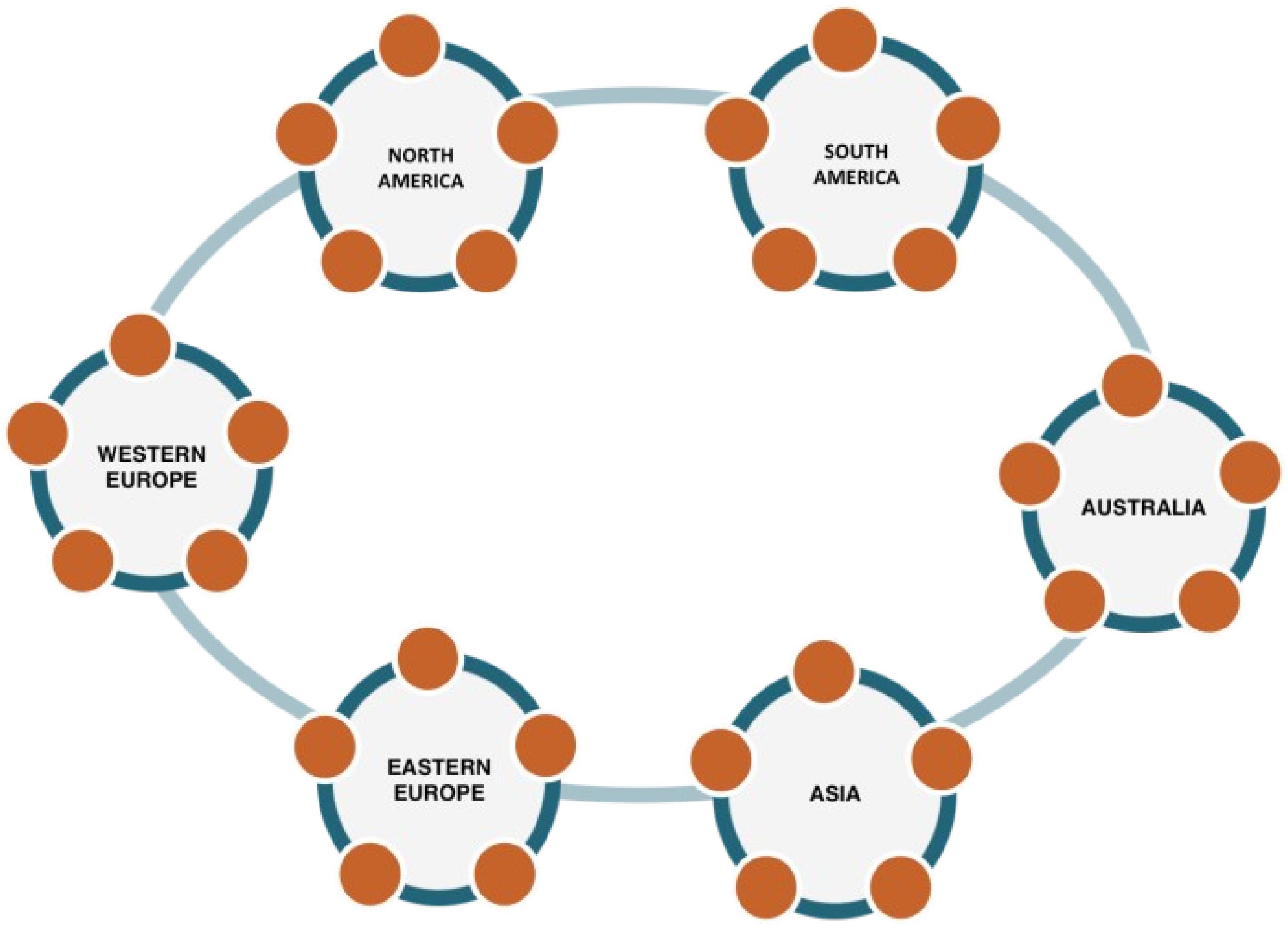


Architecture - Reading

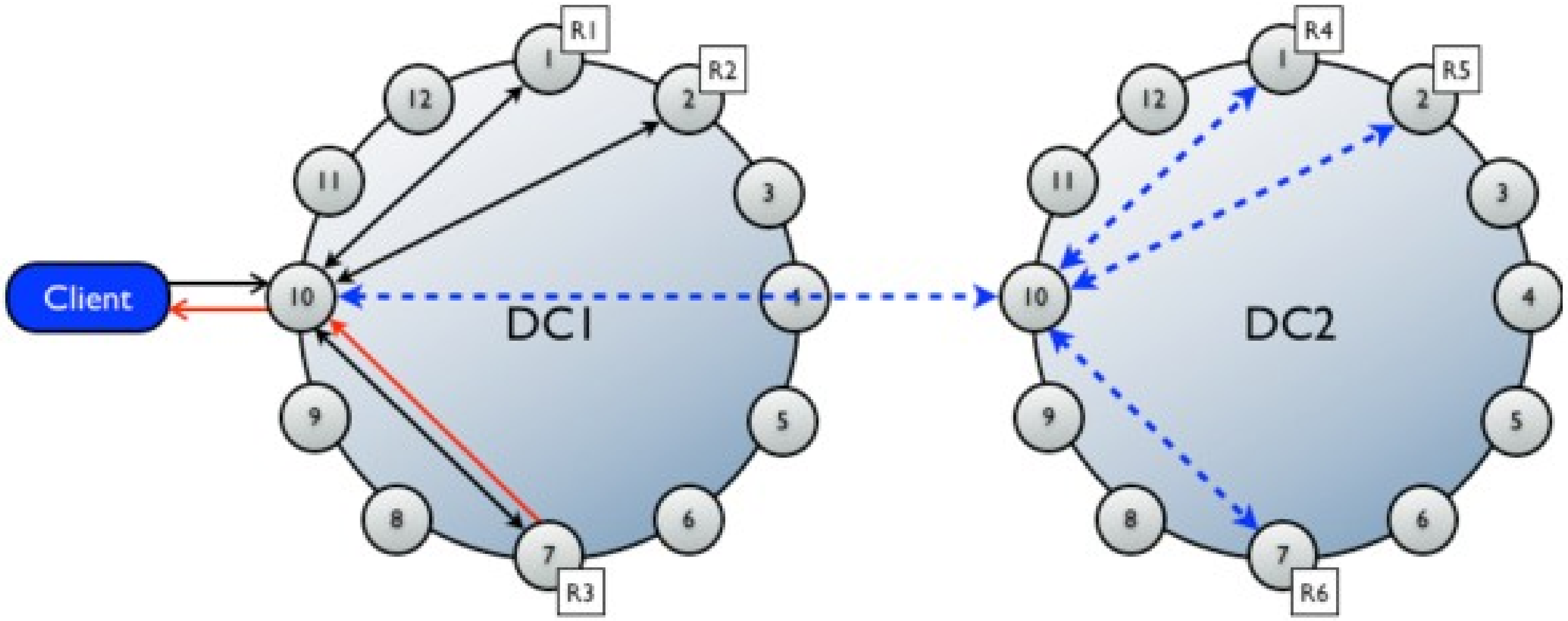


IMMUNE CODING INSTITUTE

Cassandra Cluster Multiple DataCenters



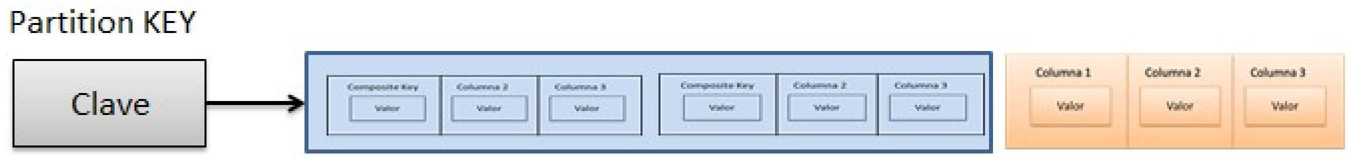
Cassandra Cluster Multiple DataCenters



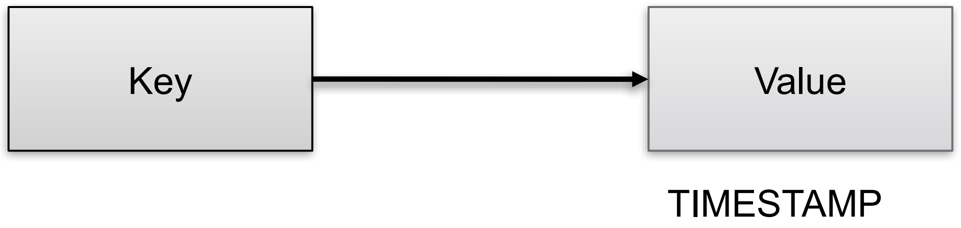
IMMUNE
CODING INSTITUTE

BigTable

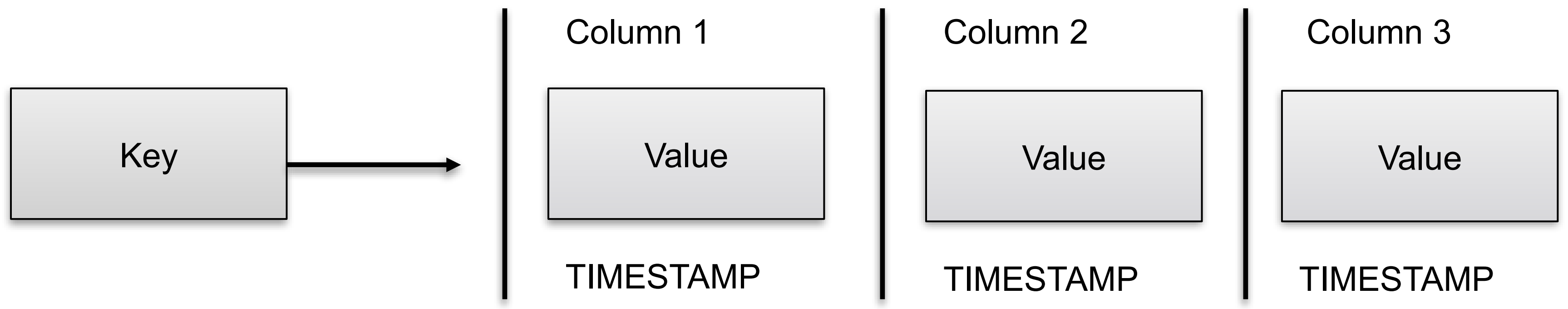
- Data is stored in tables (Column Families)
- Tables are stored in separate databases (Keyspaces)
- Every table must have a primary key (Partition Key)
- Additionally a table can have composite keys



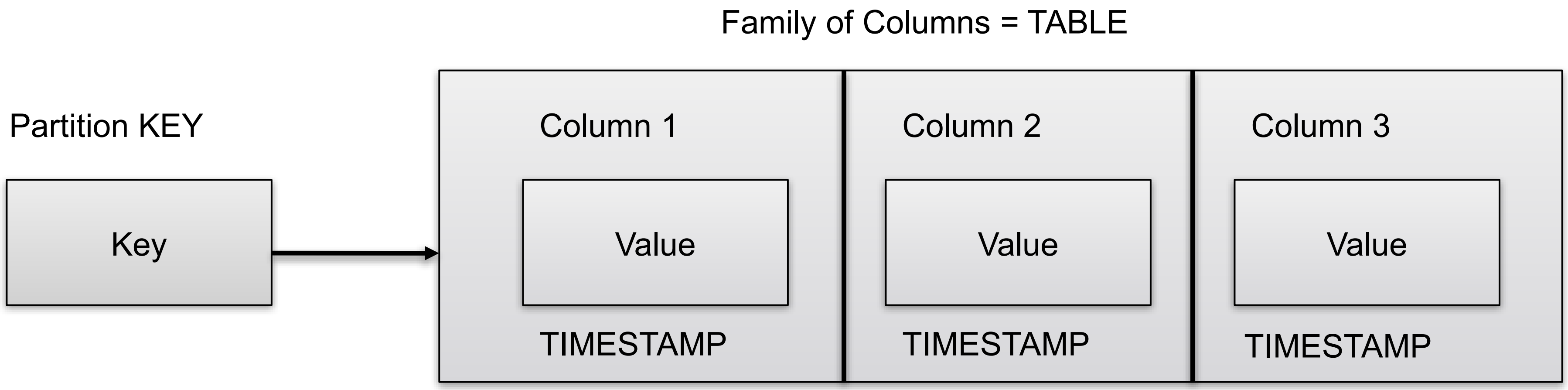
Remembering the Key - Value model



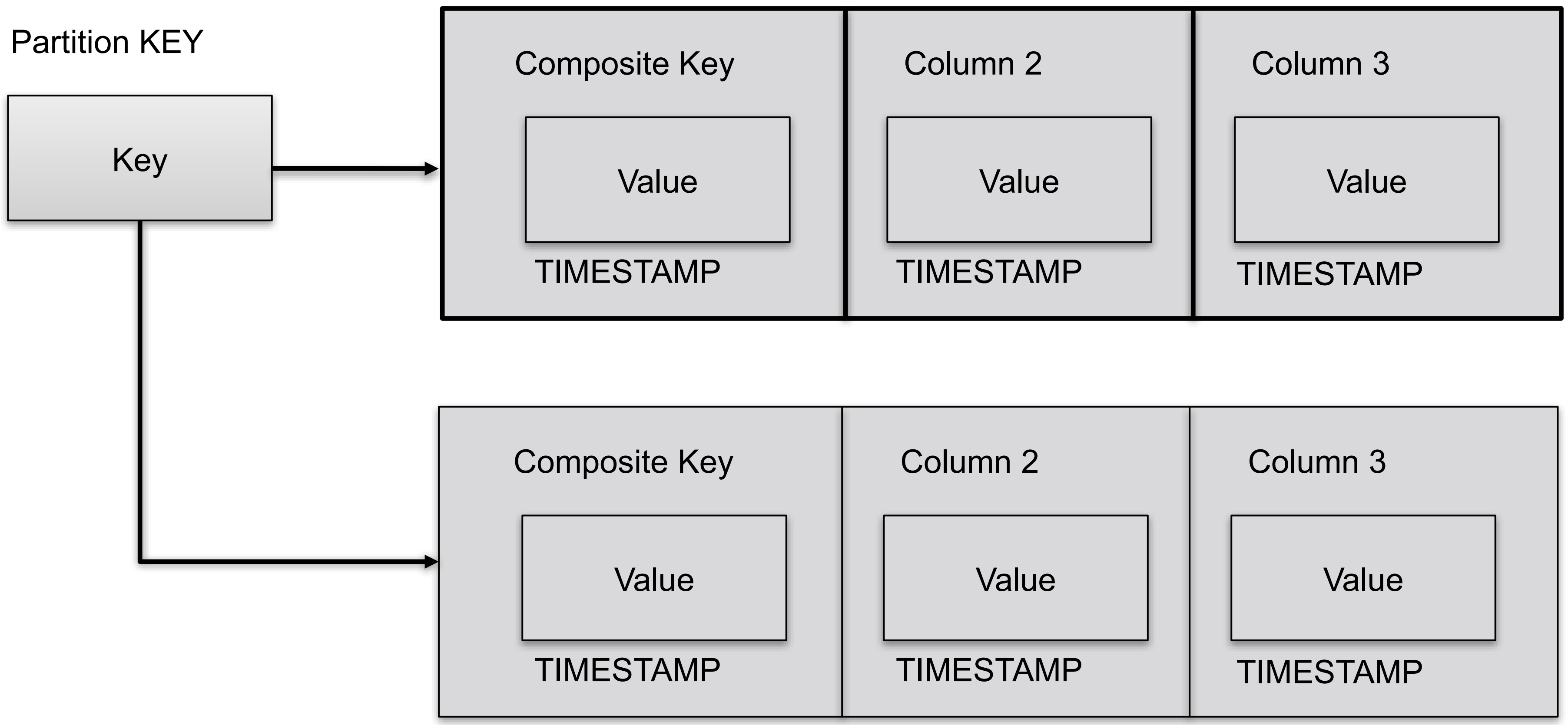
BigTable


Key
Column 1
Value
TIMESTAMP
Column 2
Value
TIMESTAMP
Column 3
Value
TIMESTAMP

BigTable

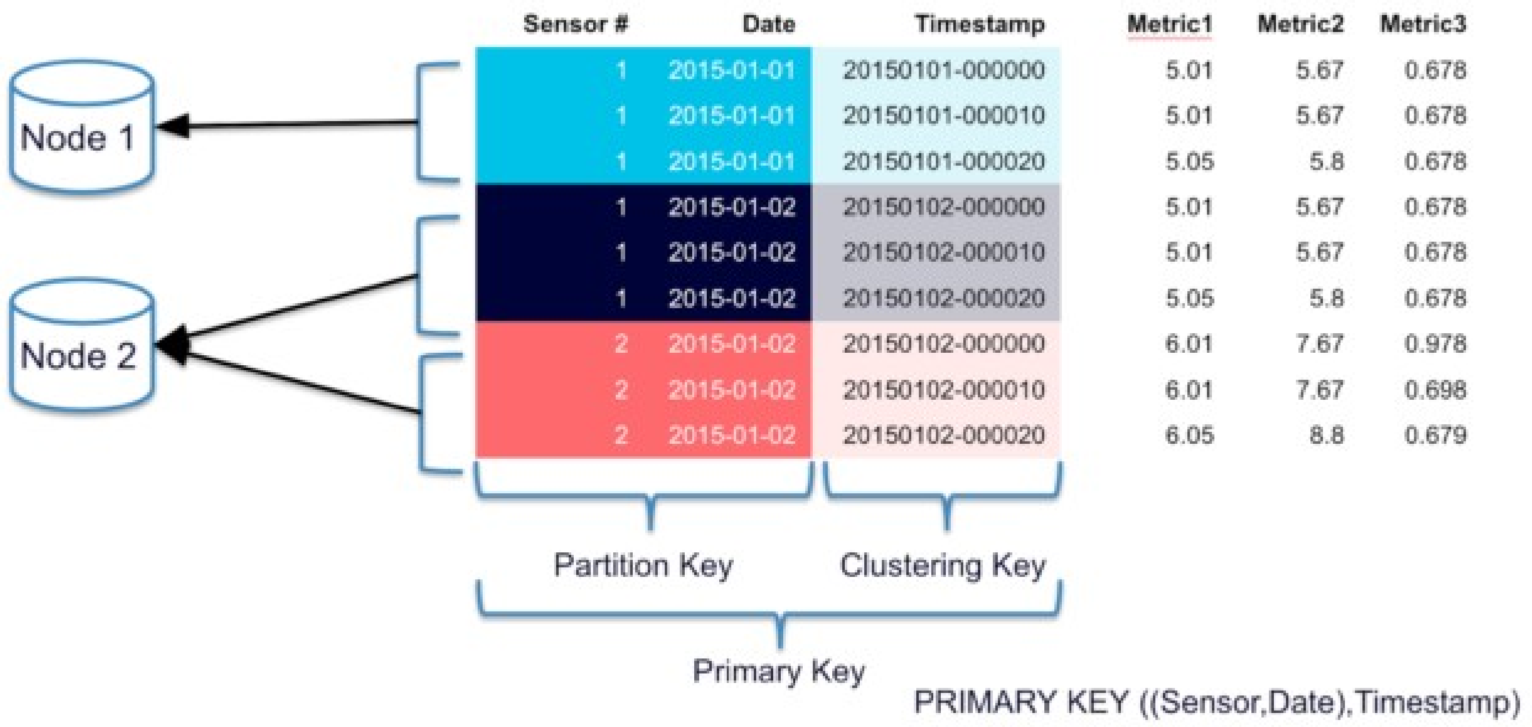

Family of Columns = TABLE
Partition KEY
Key
Column 1
Value
TIMESTAMP
Column 2
Value
TIMESTAMP
Column 3
Value
TIMESTAMP

BigTable


Partition KEY
Key
Composite Key
Value
TIMESTAMP
Column 2
Value
TIMESTAMP
Column 3
Value
TIMESTAMP
Composite Key
Value
TIMESTAMP
Column 2
Value
TIMESTAMP
Column 3
Value
TIMESTAMP

IMMUNE CODING INSTITUTE

Partition Key vs Composite Key



Sensor #	Date	Timestamp	Metric1	Metric2	Metric3
1	2015-01-01	20150101-000000	5.01	5.67	0.678
1	2015-01-01	20150101-000010	5.01	5.67	0.678
1	2015-01-01	20150101-000020	5.05	5.8	0.678
1	2015-01-02	20150102-000000	5.01	5.67	0.678
1	2015-01-02	20150102-000010	5.01	5.67	0.678
1	2015-01-02	20150102-000020	5.05	5.8	0.678
2	2015-01-02	20150102-000000	6.01	7.67	0.978
2	2015-01-02	20150102-000010	6.01	7.67	0.698
2	2015-01-02	20150102-000020	6.05	8.8	0.679

PRIMARY KEY ((Sensor,Date),Timestamp)

Cassandra Query Language CQL

- It looks like SQL but is much more limited
  - No JOINS
  - No GROUP BY
  - No ORDER BY

```
CREATE KEYSPACE demo
WITH replication = {'class':'SimpleStrategy', 'replication_factor': 1};
```

```
CREATE TABLE users (
    firstname text,
    lastname text,
    age int,
    email text,
    city text,
    PRIMARY KEY (lastname)
);
```

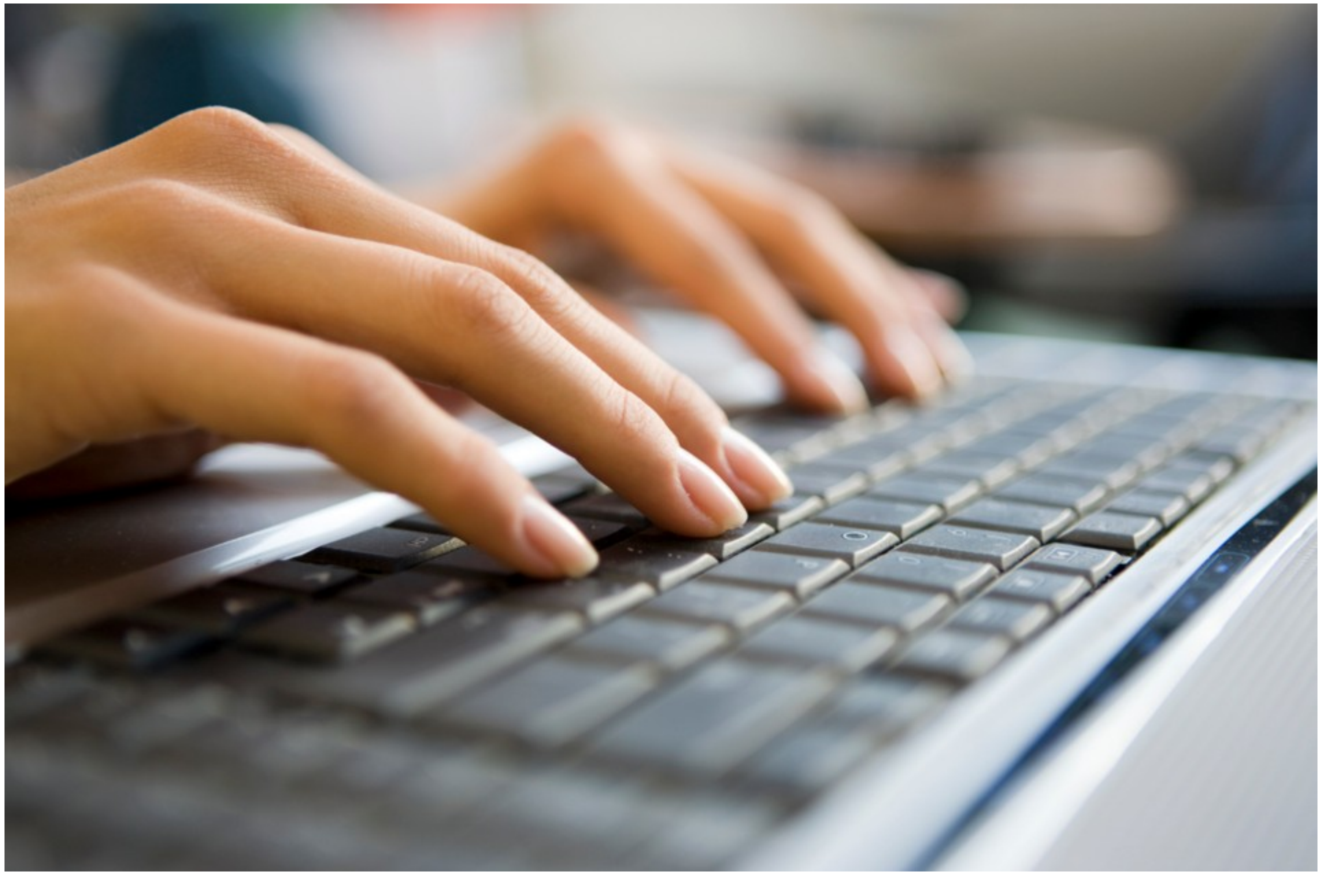
```
SELECT *
FROM users
WHERE lastname= 'Doe'
LIMIT 1;
```

```
UPDATE users
    SET city= 'San Jose'
WHERE lastname= 'Doe';
```

```
DELETE from users WHERE lastname = 'Doe';
```

```
INSERT INTO users (firstname, lastname, age, email, city)
VALUES ('John', 'Smith', 46, 'johnsmith@email.com', 'Sacramento');
```

Cassandra - HandsOn

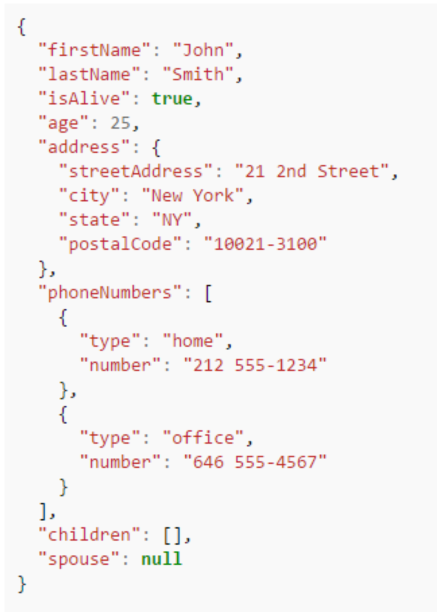


Use Cases

- Applications requiring very high real-time writing capabilities
- Business Intelligence systems that require a very fast database reading
- Decentralized applications that need to store large amounts of information
- Smart cities. Sensors and monitoring
- Content Delivery Network (CDN) - Highly distributed static content servers

Agenda

- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- Apache Cassandra
- MongoDB
- Neo4j

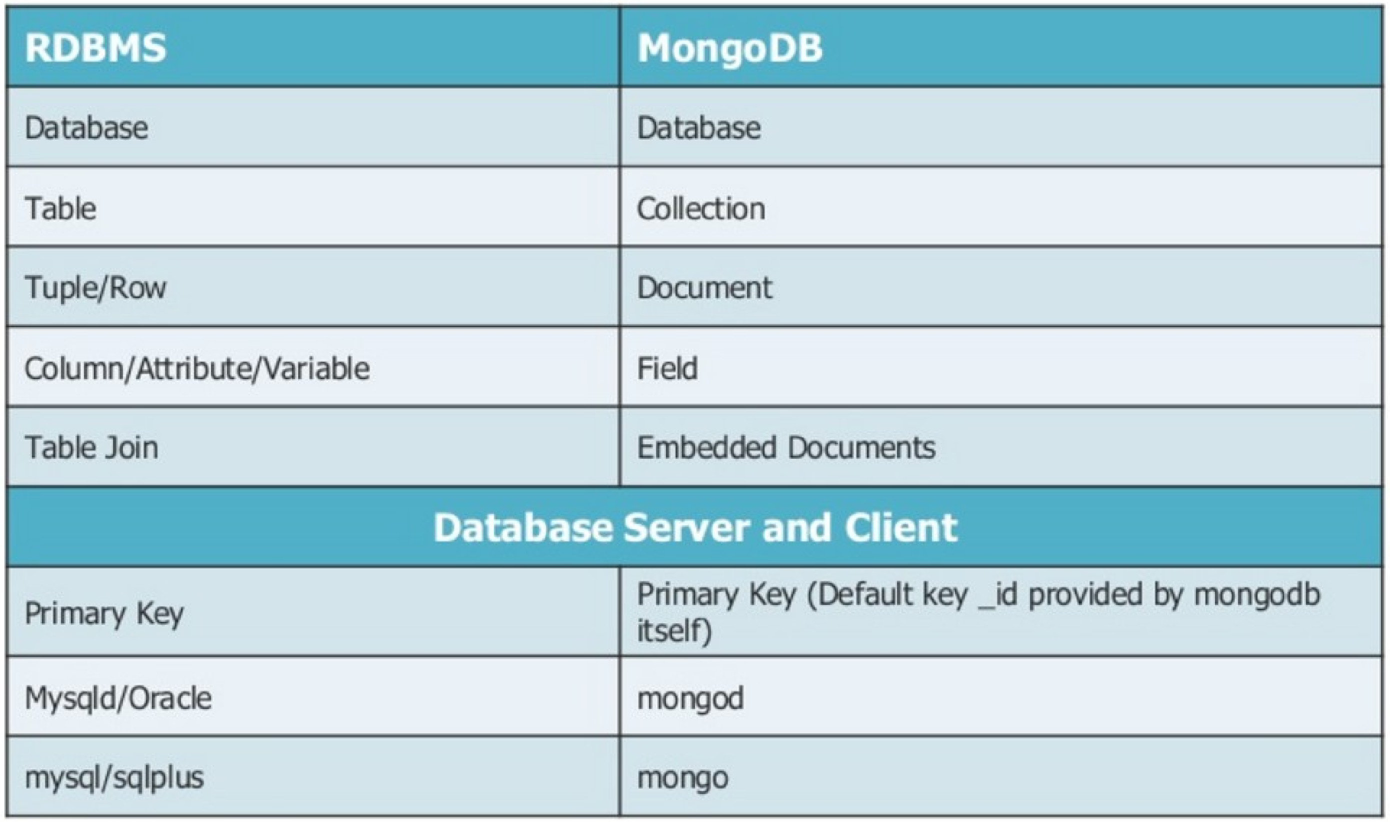


```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

MongoDB

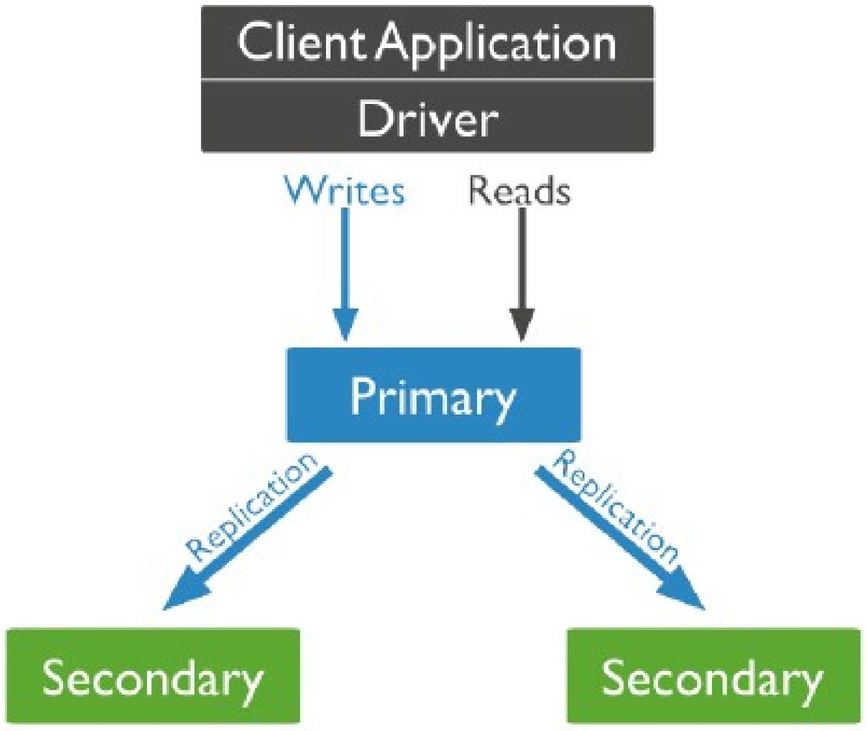
- It is a document-oriented database (Key Collections / value)
- It is very flexible in structuring the data
- Querys: Javascript with its own API based on high capabilities for information querying
- It has geospatial characteristics
- Prioritizes consistency over availability
- Master / Slave type replication
- Scale horizontally thanks to Sharding
- There is a connector for BI tools
- Compatible with Hadoop and Spark

Nomenclature

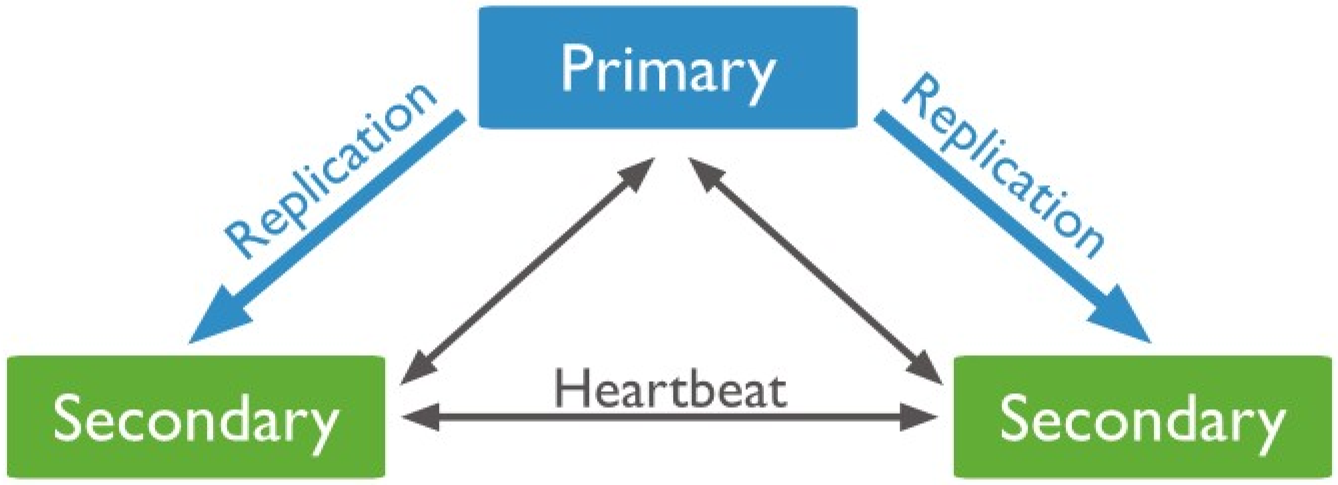


RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
Column/Attribute/Variable	Field
Table Join	Embedded Documents
Database Server and Client	
Primary Key	Primary Key (Default key _id provided by mongodb itself)
Mysqld/Oracle	mongod
mysql/sqlplus	mongo

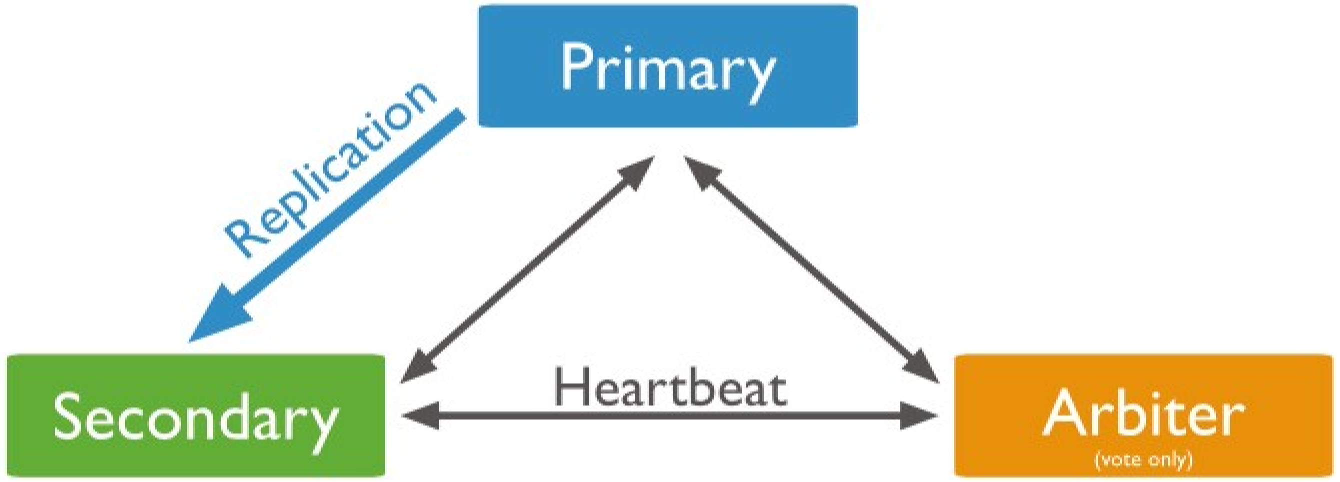
Replication - Scale the readings



Replication

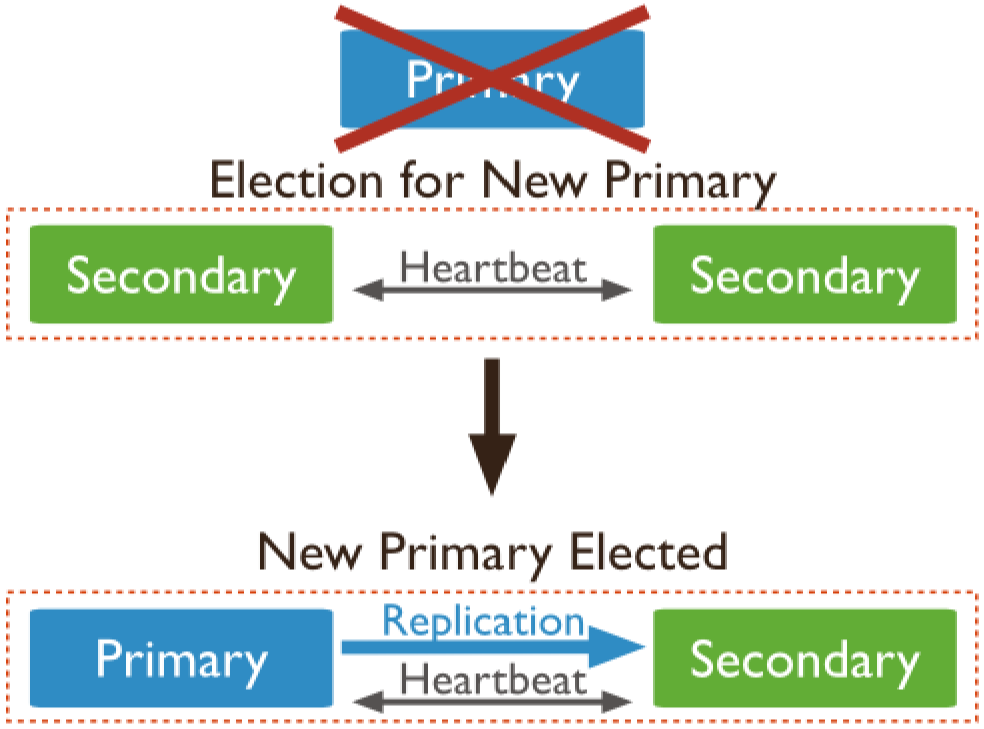


Replication

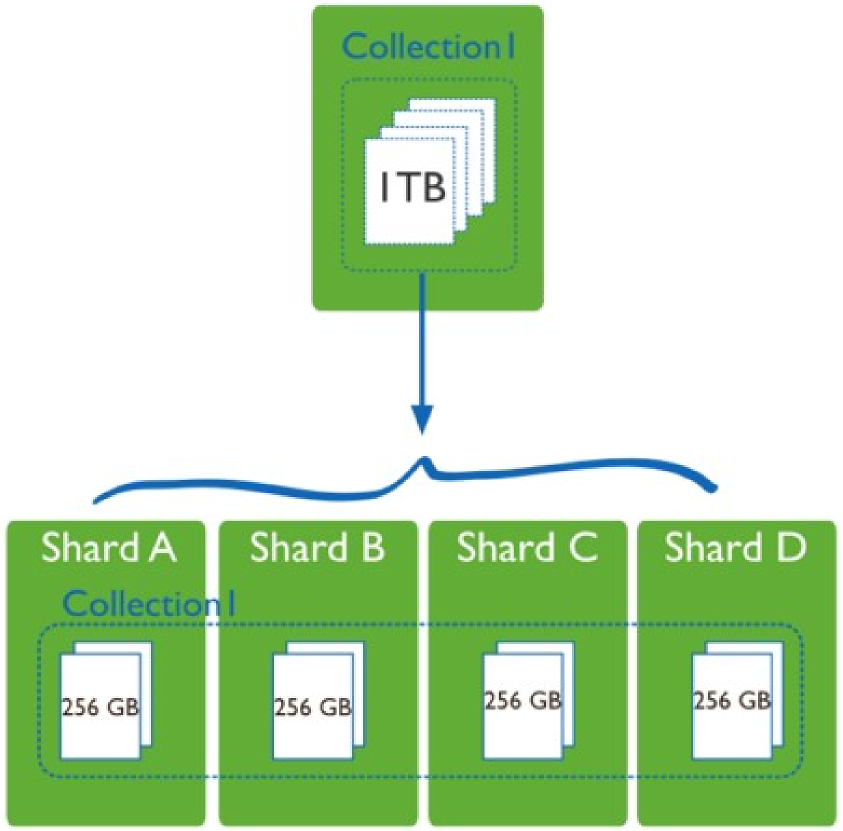


IMMUNE
CODING INSTITUTE

Replication

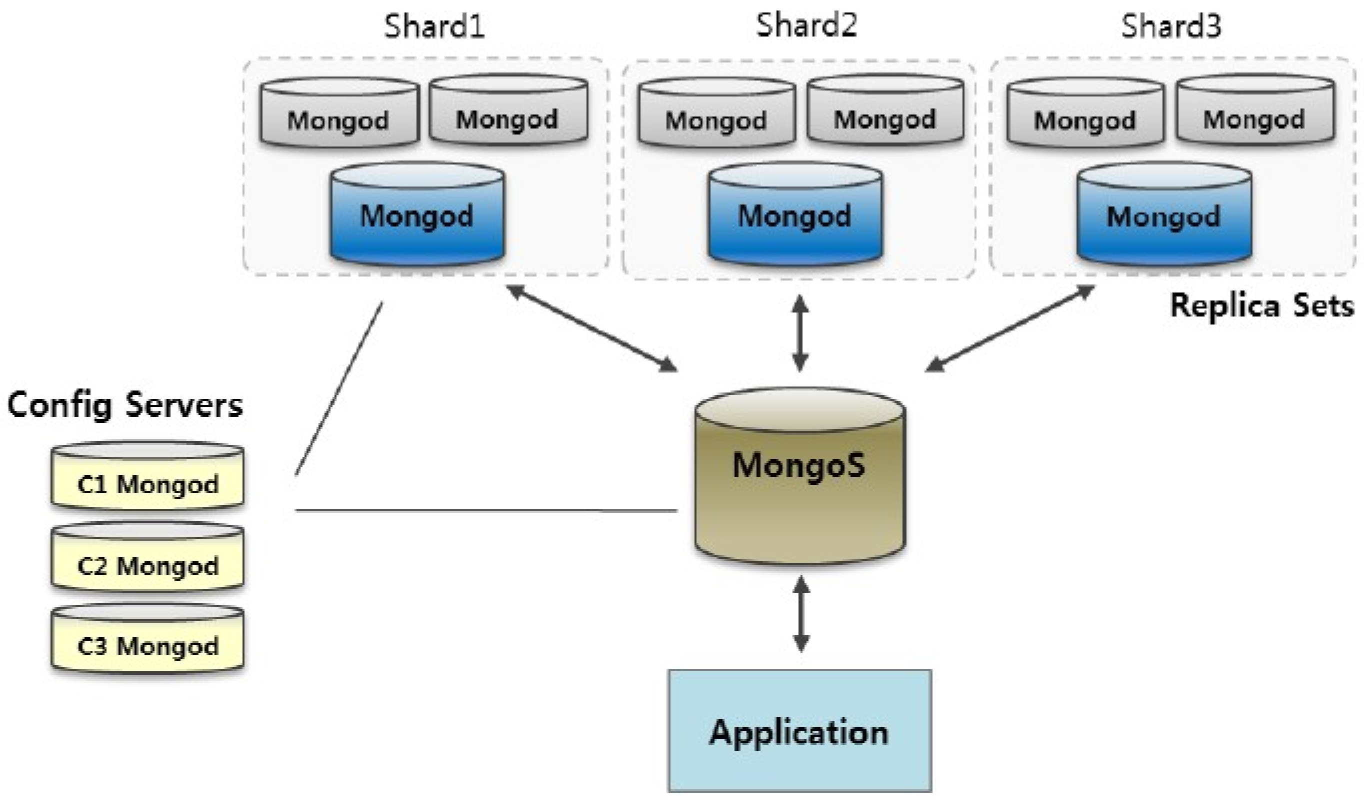


Sharding- Scale the scriptures

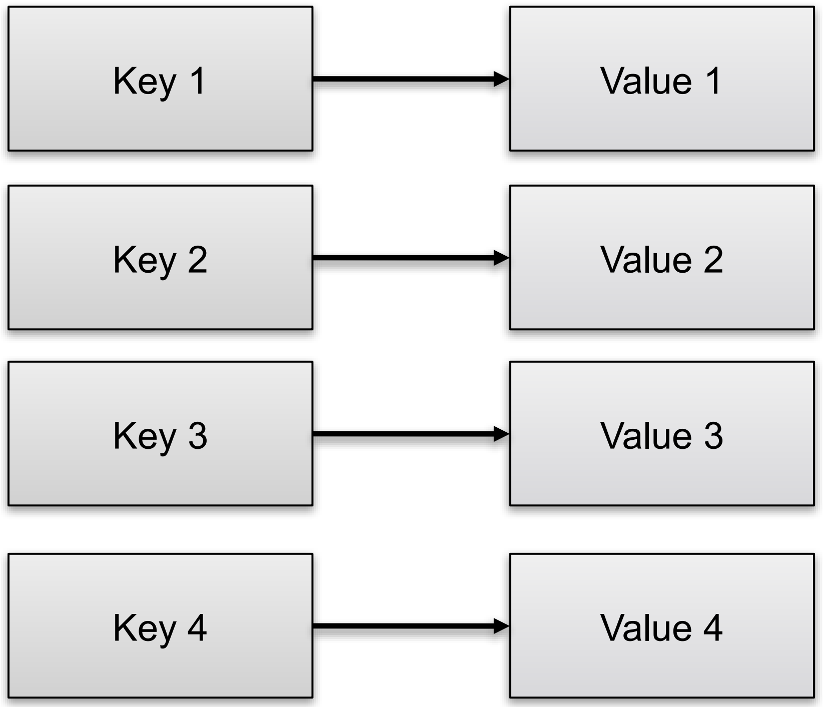


IMMUNE CODING INSTITUTE

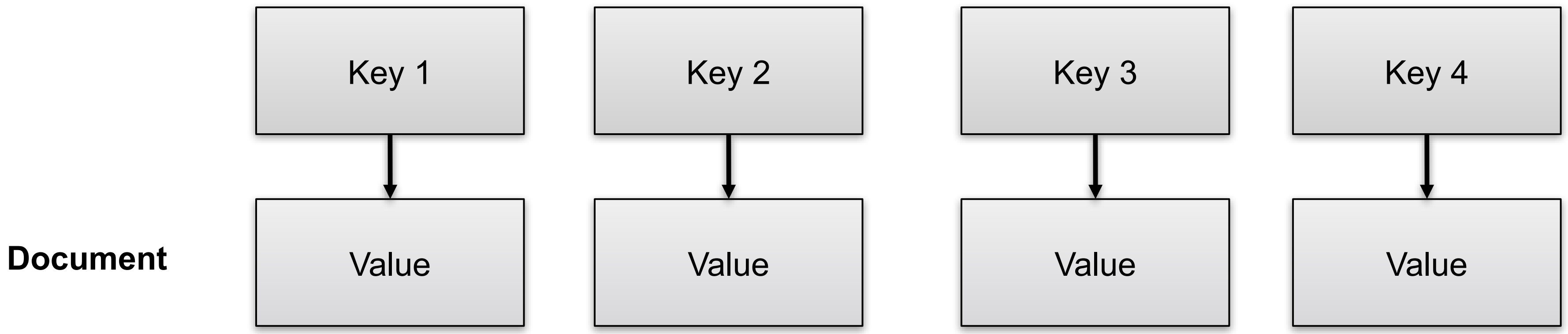
Complete map



Document

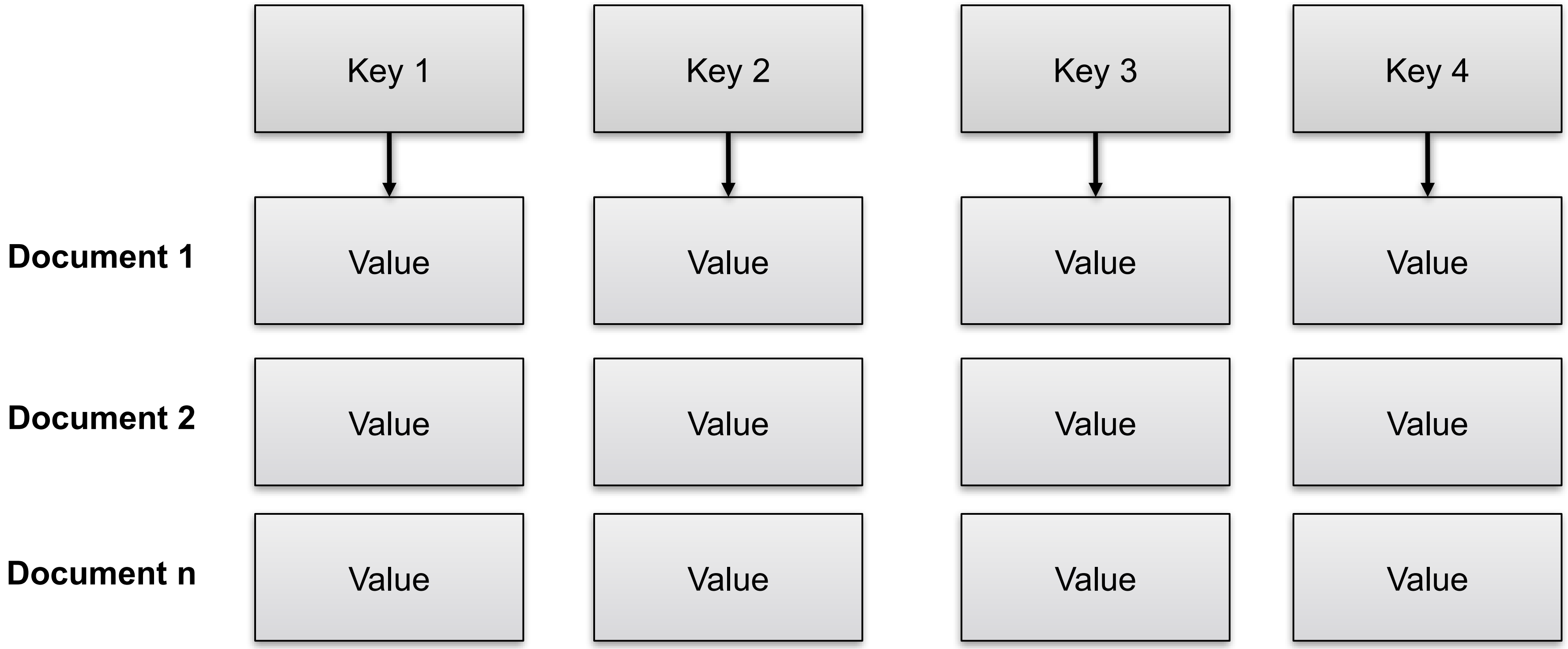


Document

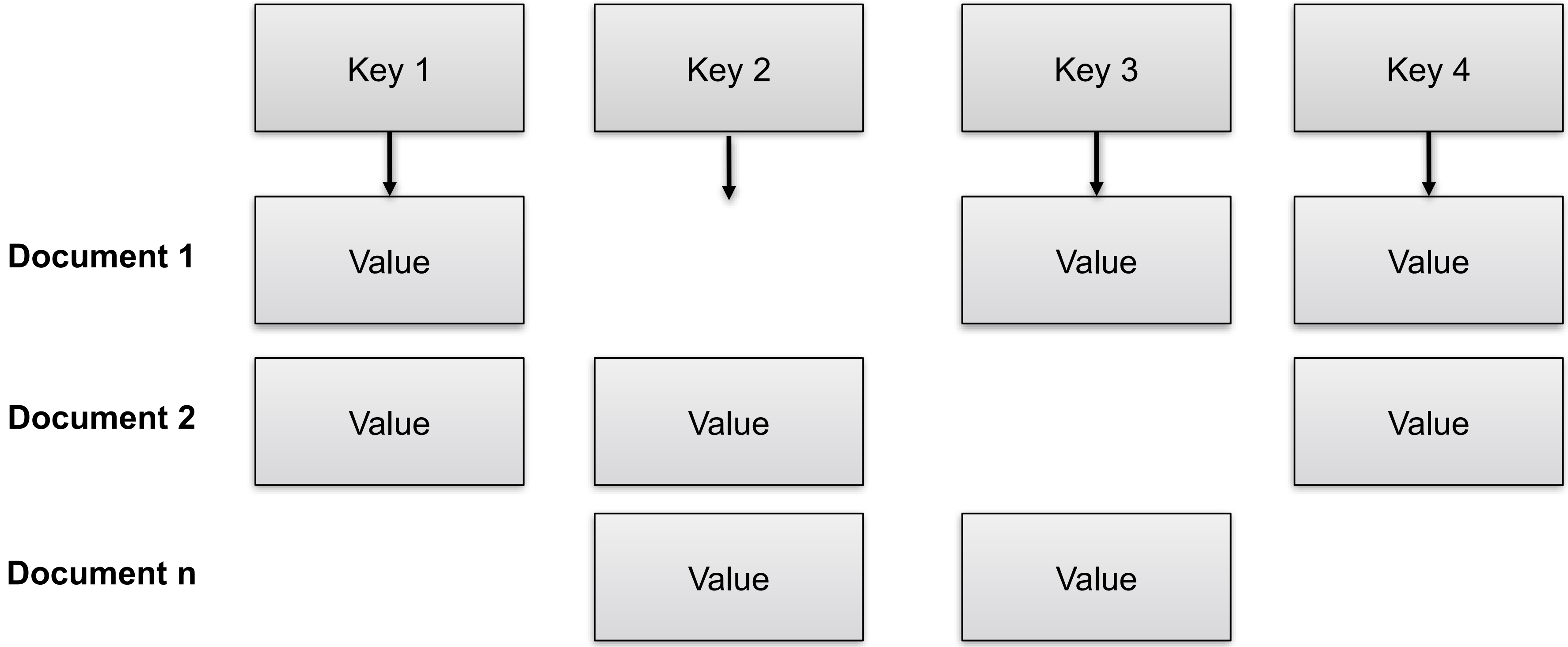


IMMUNE CODING INSTITUTE

Collection



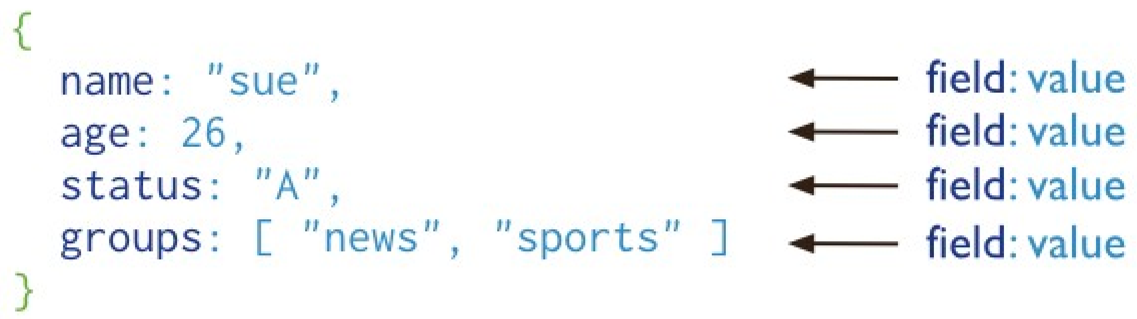
Collection



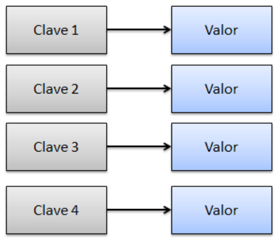
Key 1
Key 2
Key 3
Key 4
Document 1
Value
Value
Value
Document 2
Value
Value
Value
Document n
Value
Value

Documents

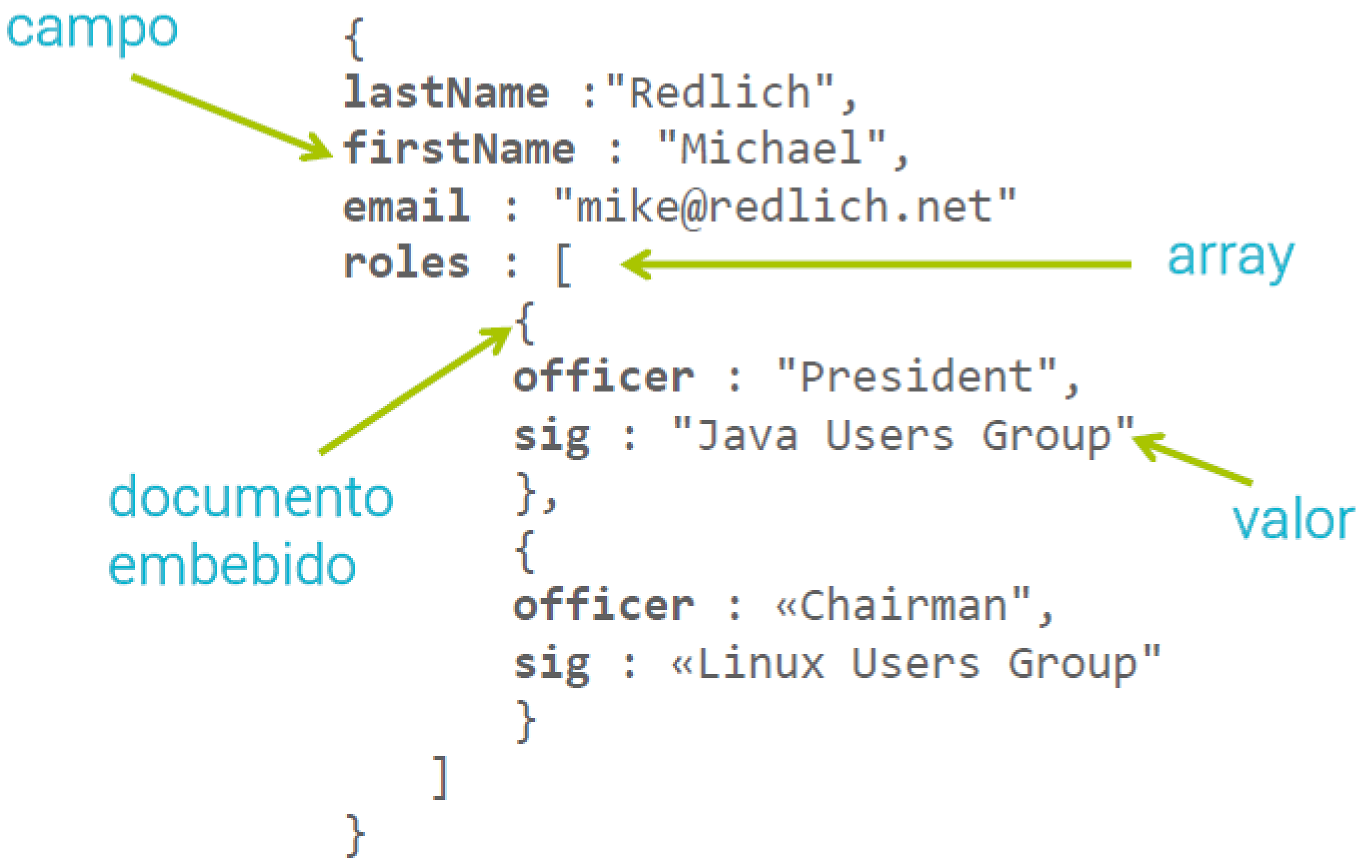
- The documents correspond to native data types in most programming languages.
- The ability to include other documents and arrays within the documents reduces the need for joins.
- Dynamic schemes allow support for any data structure in a collection



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



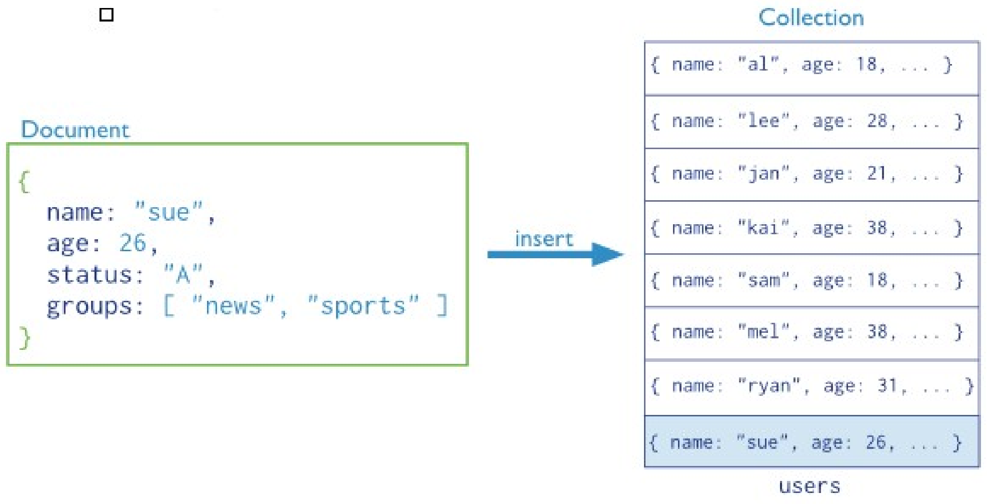
Document



```
{
lastName :"Redlich",
firstName : "Michael",
email : "mike@redlich.net"
roles : [
        {
        officer : "President",
        sig : "Java Users Group"
        },
        {
        officer : «Chairman",
        sig : «Linux Users Group"
        }
    ]
}
```

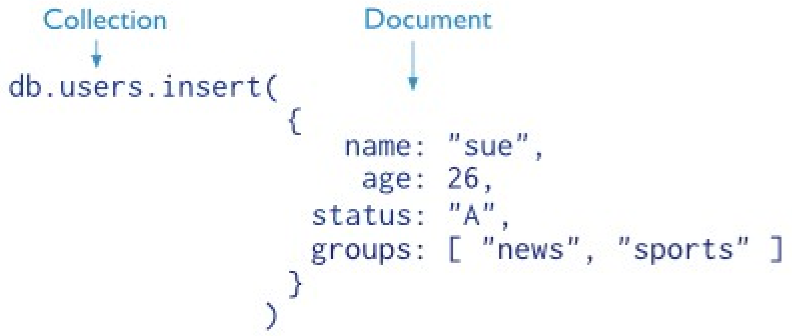
CRUD operations

- Data insertion



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

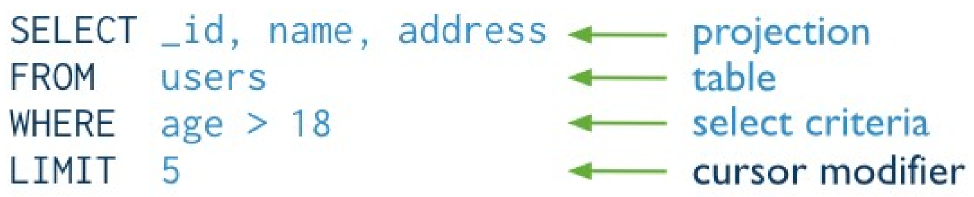
```
{ name: "al", age: 18, ... }
{ name: "lee", age: 28, ... }
{ name: "jan", age: 21, ... }
{ name: "kai", age: 38, ... }
{ name: "sam", age: 18, ... }
{ name: "mel", age: 38, ... }
{ name: "ryan", age: 31, ... }
{ name: "sue", age: 26, ... }
```



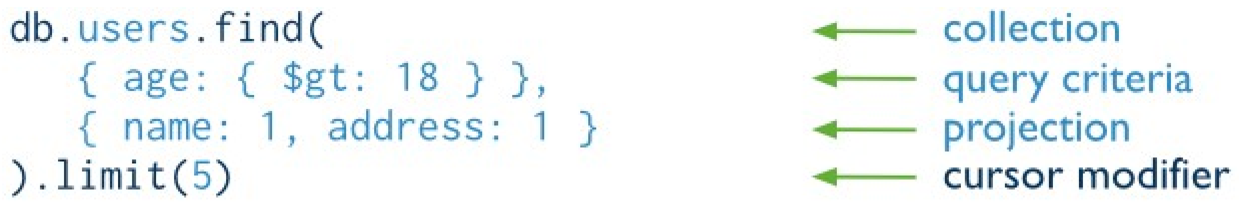
```
db.users.insert(
                {
                   name: "sue",
                    age: 26,
                 status: "A",
                 groups: [ "news", "sports" ]
                }
              )
```

CRUD operations

- Querys



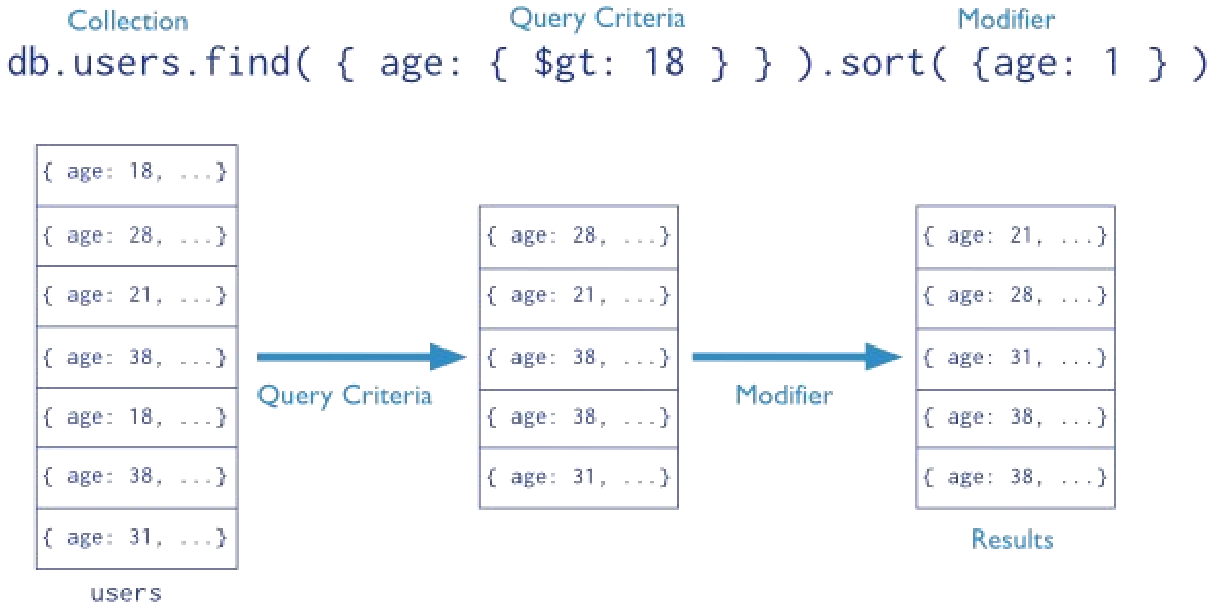
```
SELECT _id, name, address
FROM   users
WHERE  age > 18
LIMIT  5
```



```
db.users.find(
    { age: { $gt: 18 } },
    { name: 1, address: 1 }
).limit(5)
```

CRUD operations

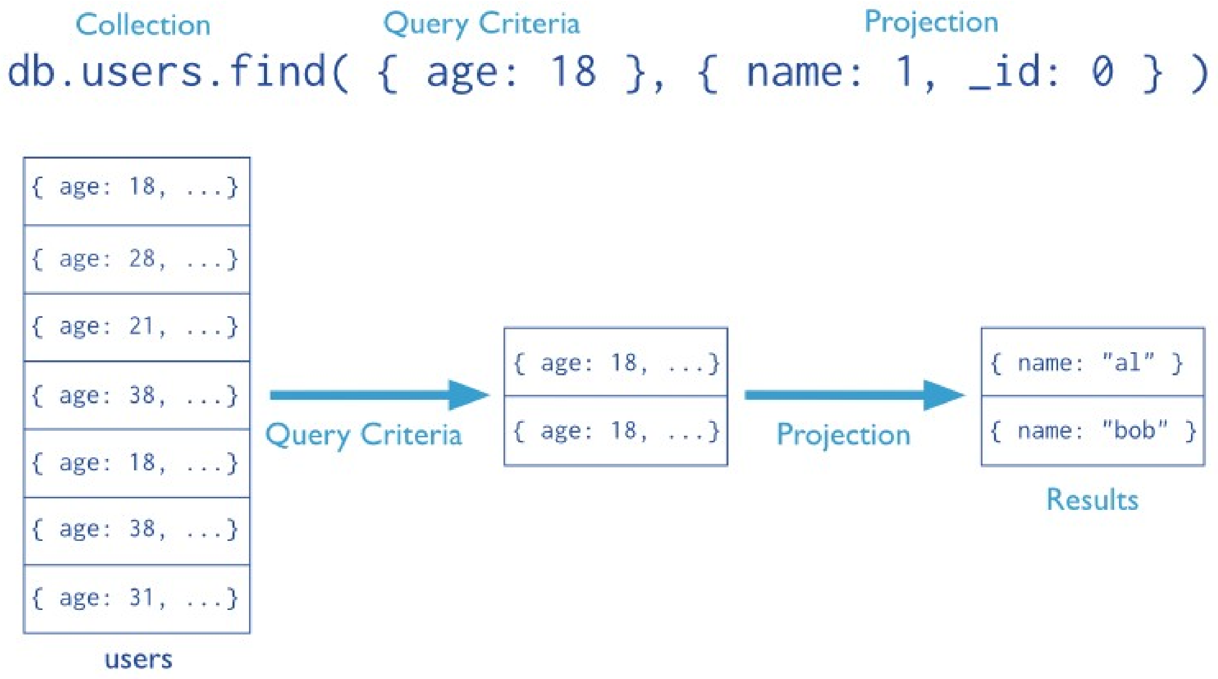
- Ordered queries



IMMUNE CODING INSTITUTE

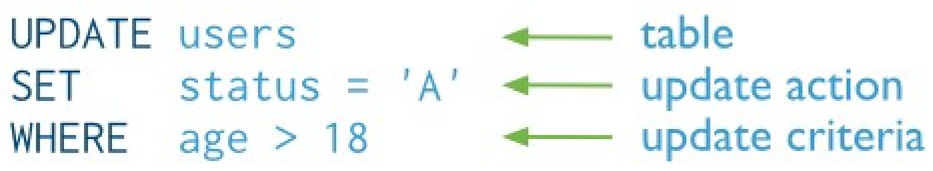
CRUD operations

- Queries with projection

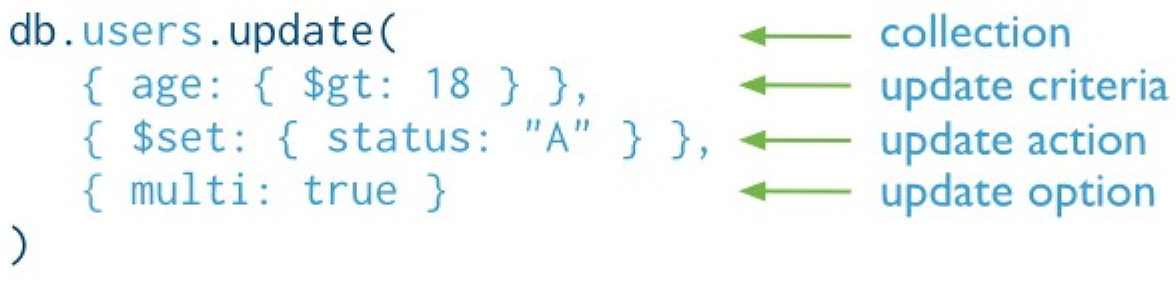


CRUD operations

- Update



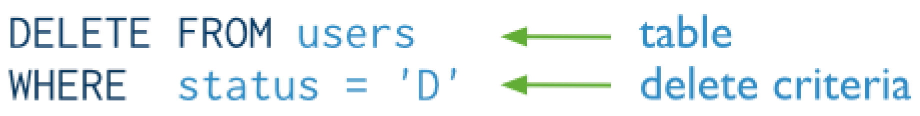
```
UPDATE users
SET    status = 'A'
WHERE  age > 18
```



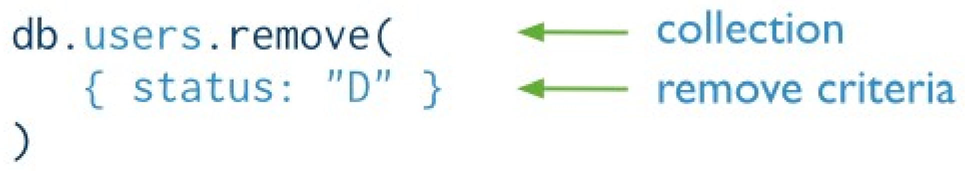
```
db.users.update(
   { age: { $gt: 18 } },
   { $set: { status: "A" } },
   { multi: true }
)
```

CRUD operations

- Deletion



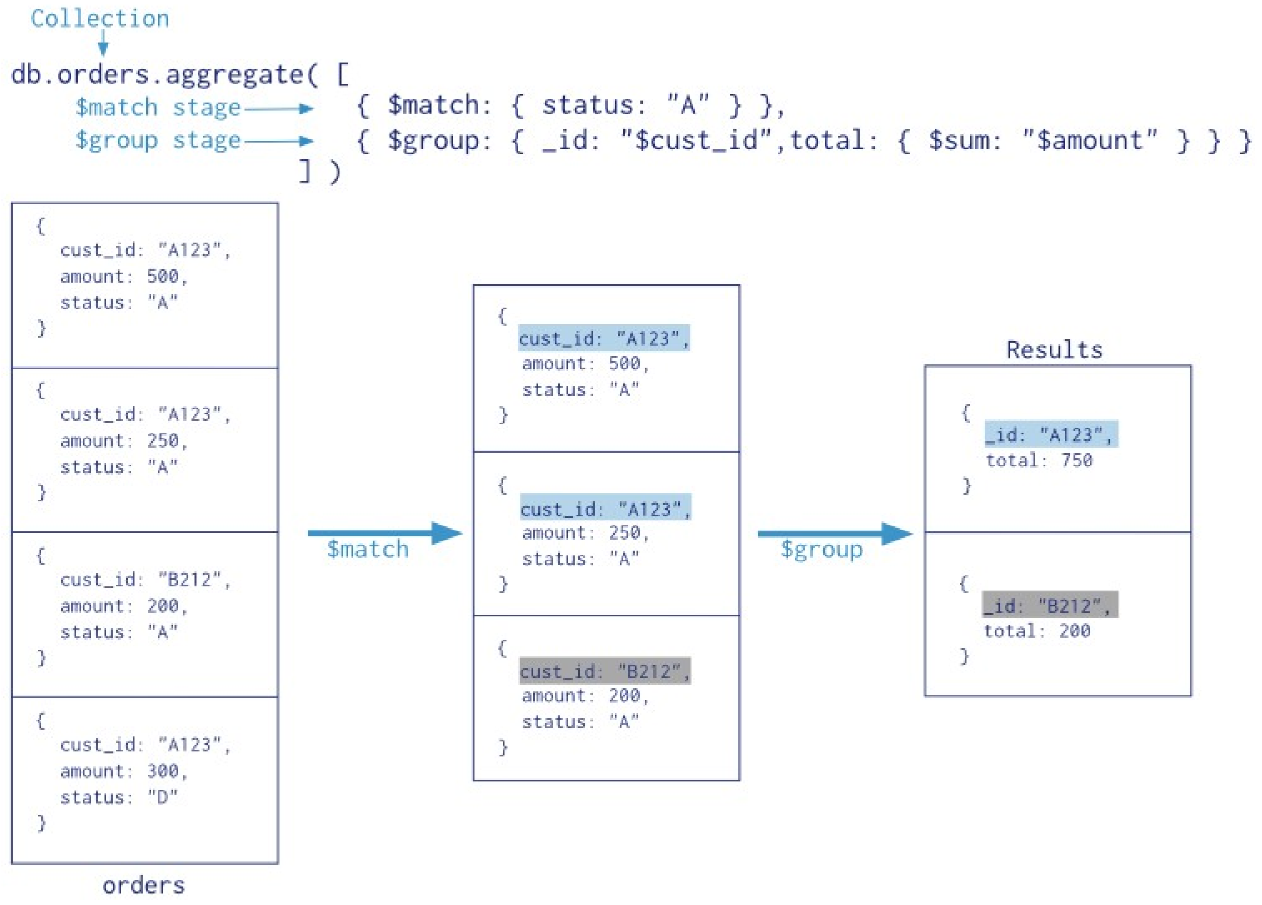
```
DELETE FROM users
WHERE  status = 'D'
```



```
db.users.remove(
   { status: "D" }
)
```

CRUD operations

- Aggregation - Equivalent to GROUP BY



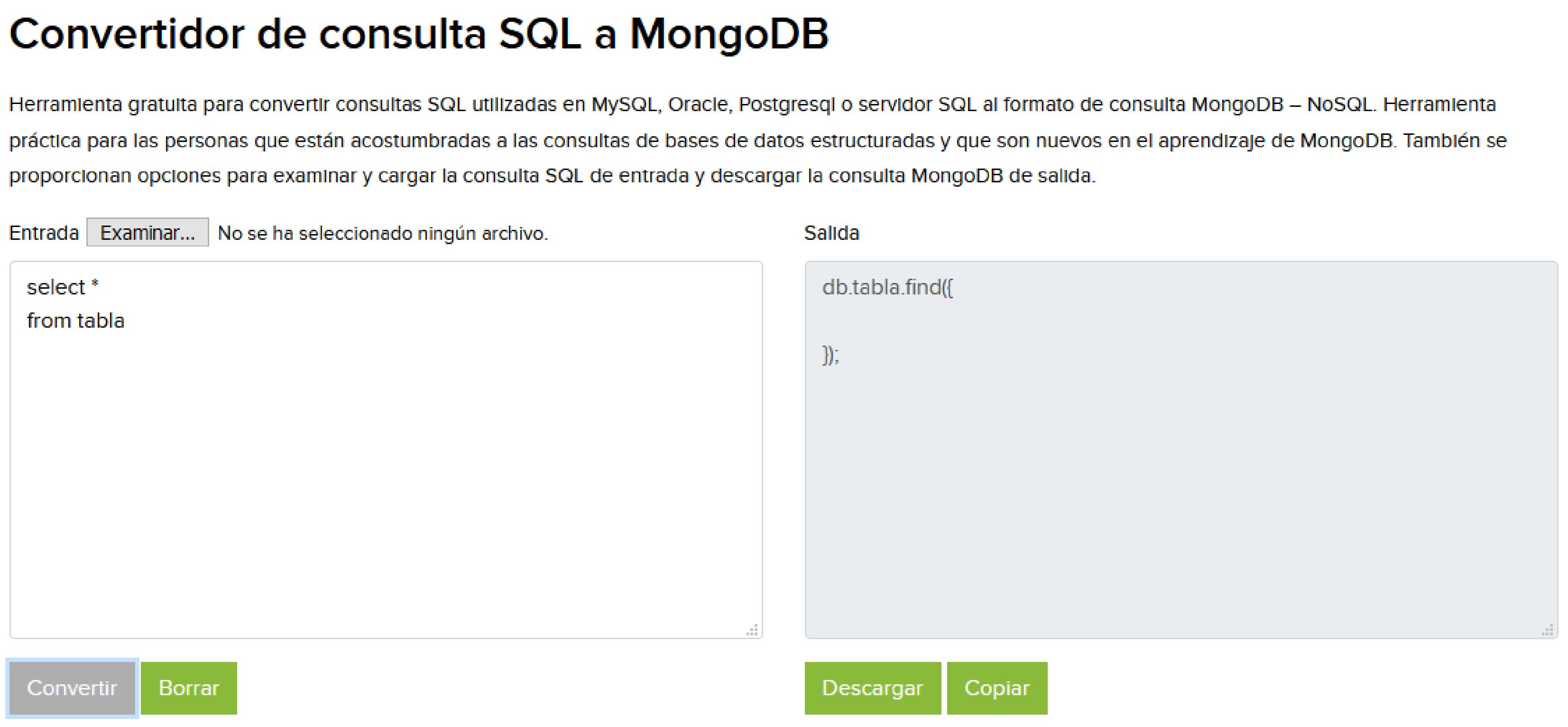
```
db.orders.aggregate( [
    { $match: { status: "A" } },
    { $group: { _id: "$cust_id",total: { $sum: "$amount" } } }
] )
```

```
{
  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"
}
```

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

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

Converting from SQL to MongoDB



Convertidor de consulta SQL a MongoDB

Herramienta gratuita para convertir consultas SQL utilizadas en MySQL, Oracle, Postgresql o servidor SQL al formato de consulta MongoDB – NoSQL. Herramienta práctica para las personas que están acostumbradas a las consultas de bases de datos estructuradas y que son nuevos en el aprendizaje de MongoDB. También se proporcionan opciones para examinar y cargar la consulta SQL de entrada y descargar la consulta MongoDB de salida.

Entrada Examinar... No se ha seleccionado ningún archivo.

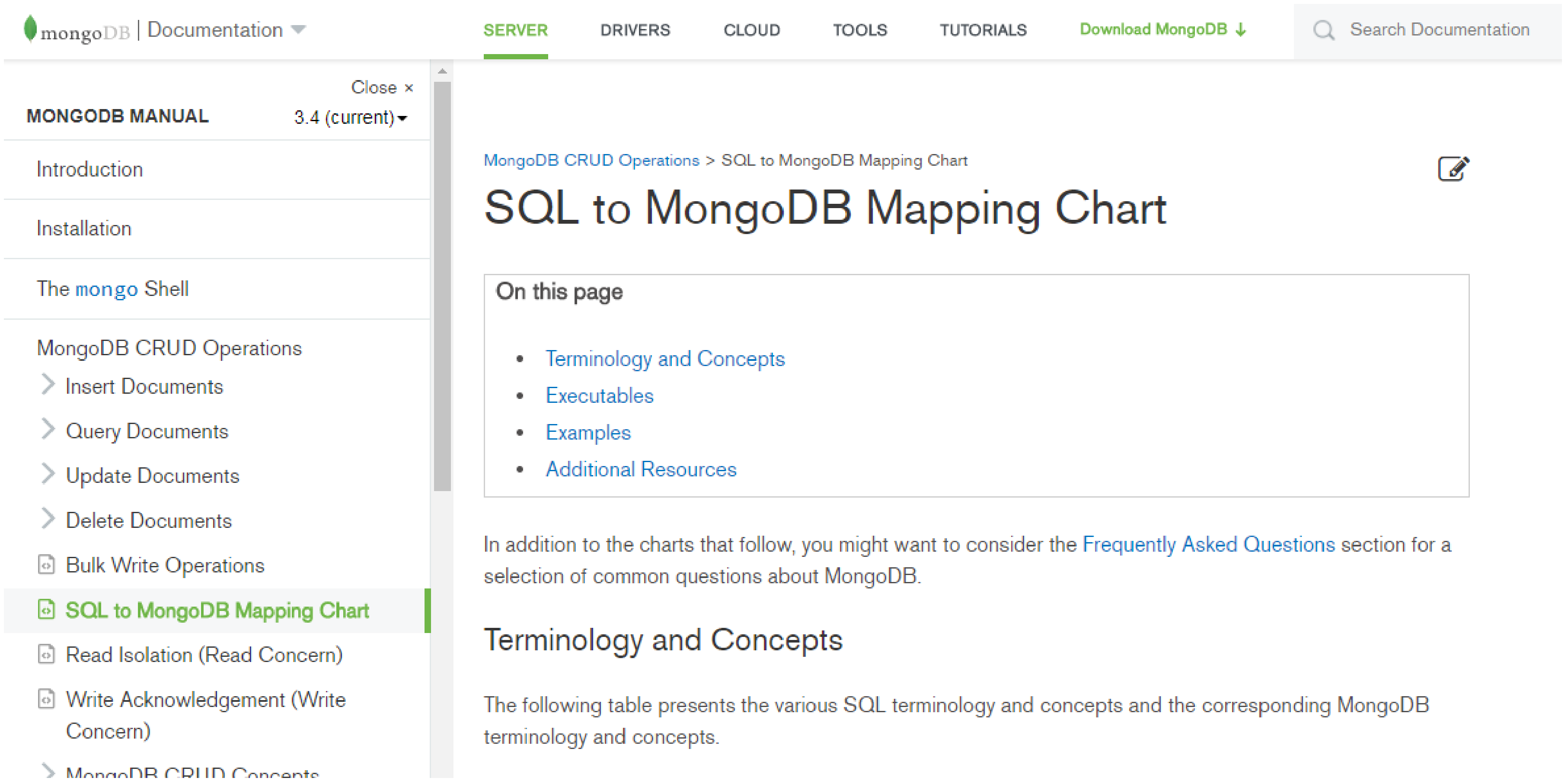
```
select *
from tabla
```

Salida

```
db.tabla.find({

});
```

https://www.google.com/search?q=sql+to+mongodb

Converting from SQL to MongoDB



MongoDB CRUD Operations > SQL to MongoDB Mapping Chart

SQL to MongoDB Mapping Chart

On this page

- Terminology and Concepts
- Executables
- Examples
- Additional Resources

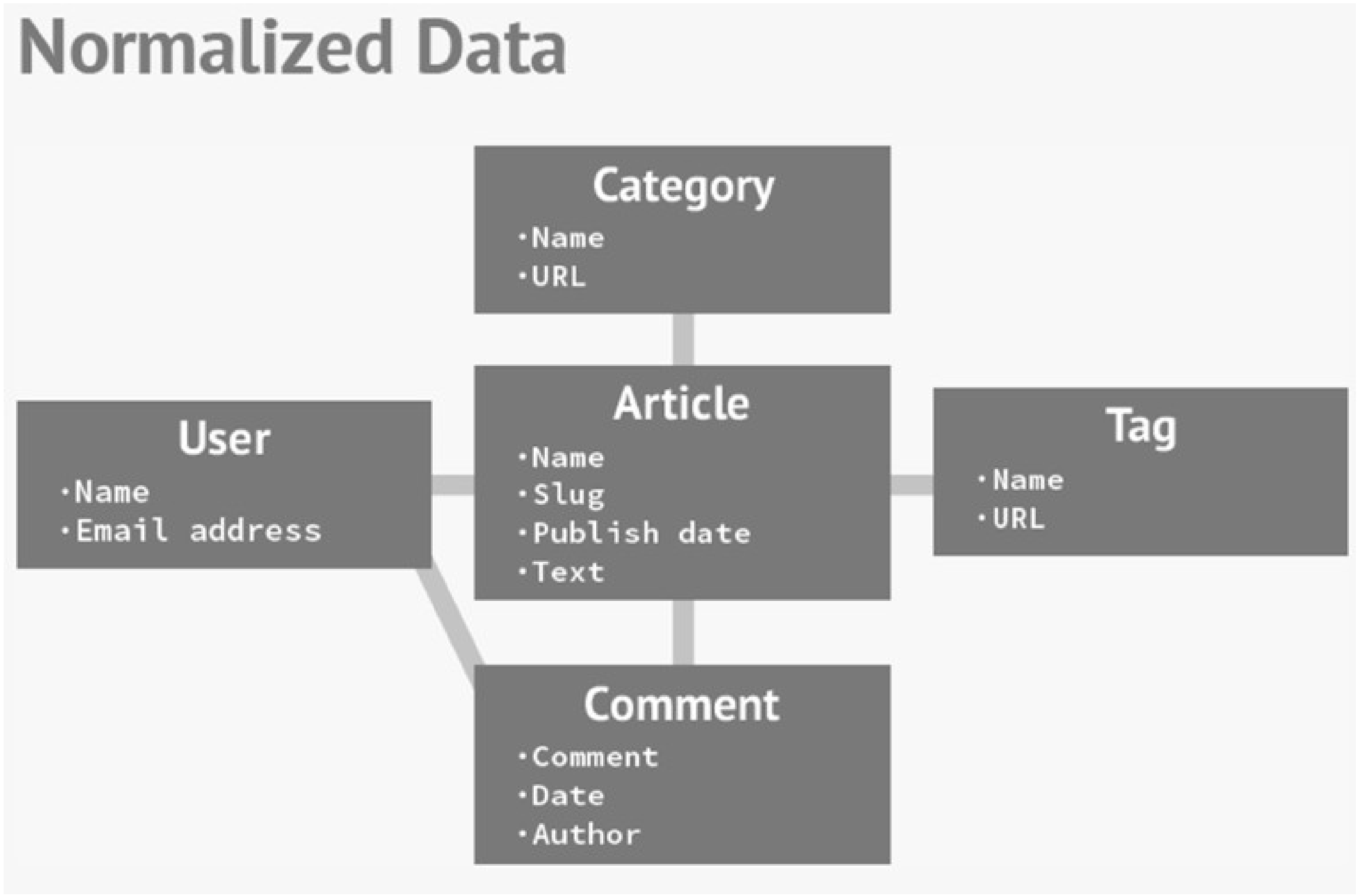
In addition to the charts that follow, you might want to consider the Frequently Asked Questions section for a selection of common questions about MongoDB.

Terminology and Concepts

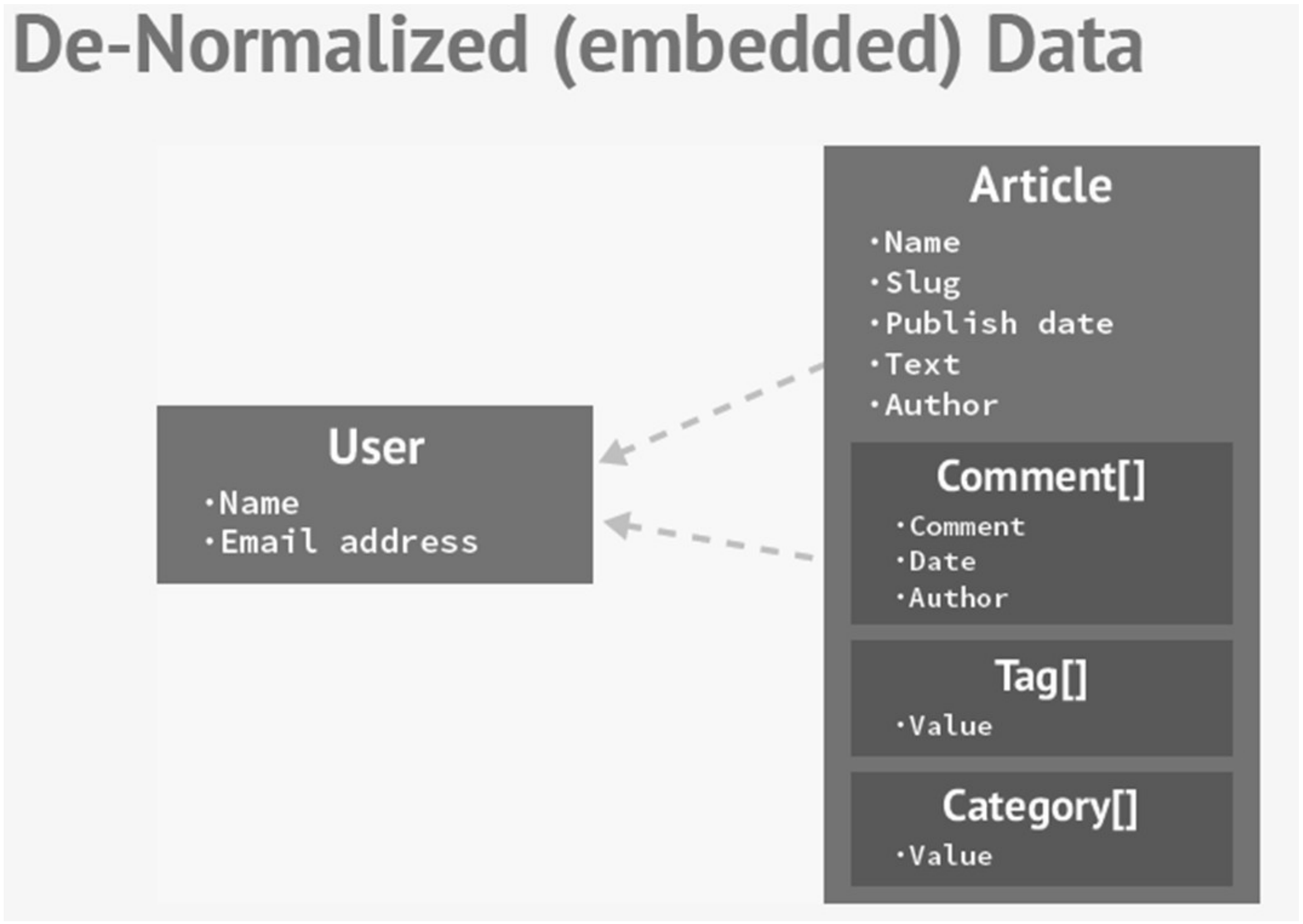
The following table presents the various SQL terminology and concepts and the corresponding MongoDB terminology and concepts.

https://docs.mongodb.com/manual/reference/sql-comparison/

Schematic design

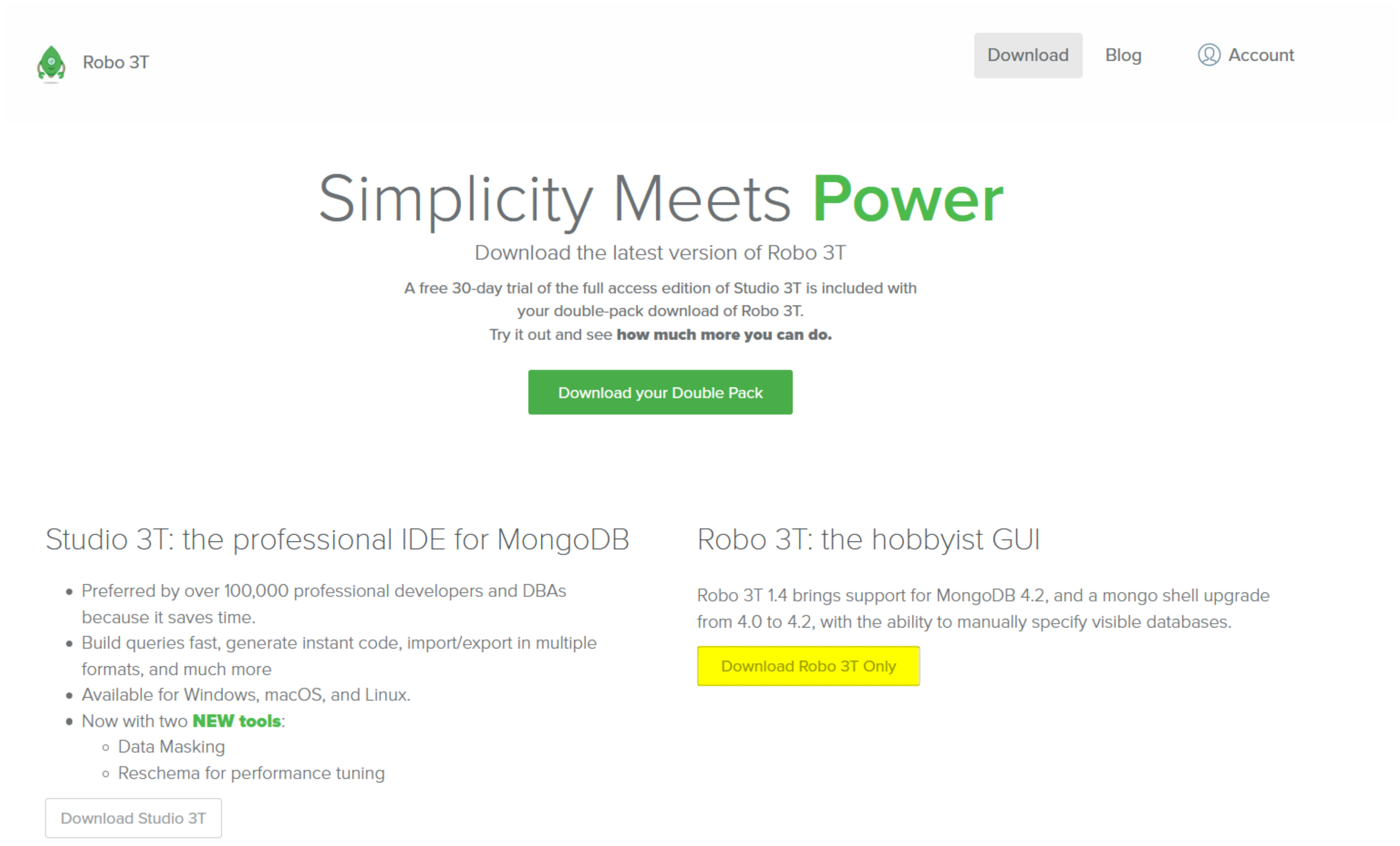


Scheme design



IMMUNE
CODING INSTITUTE

MongoDB Client



Simplicity Meets **Power**

Download the latest version of Robo 3T

A free 30-day trial of the full access edition of Studio 3T is included with your double-pack download of Robo 3T.
Try it out and see **how much more you can do.**

Studio 3T: the professional IDE for MongoDB

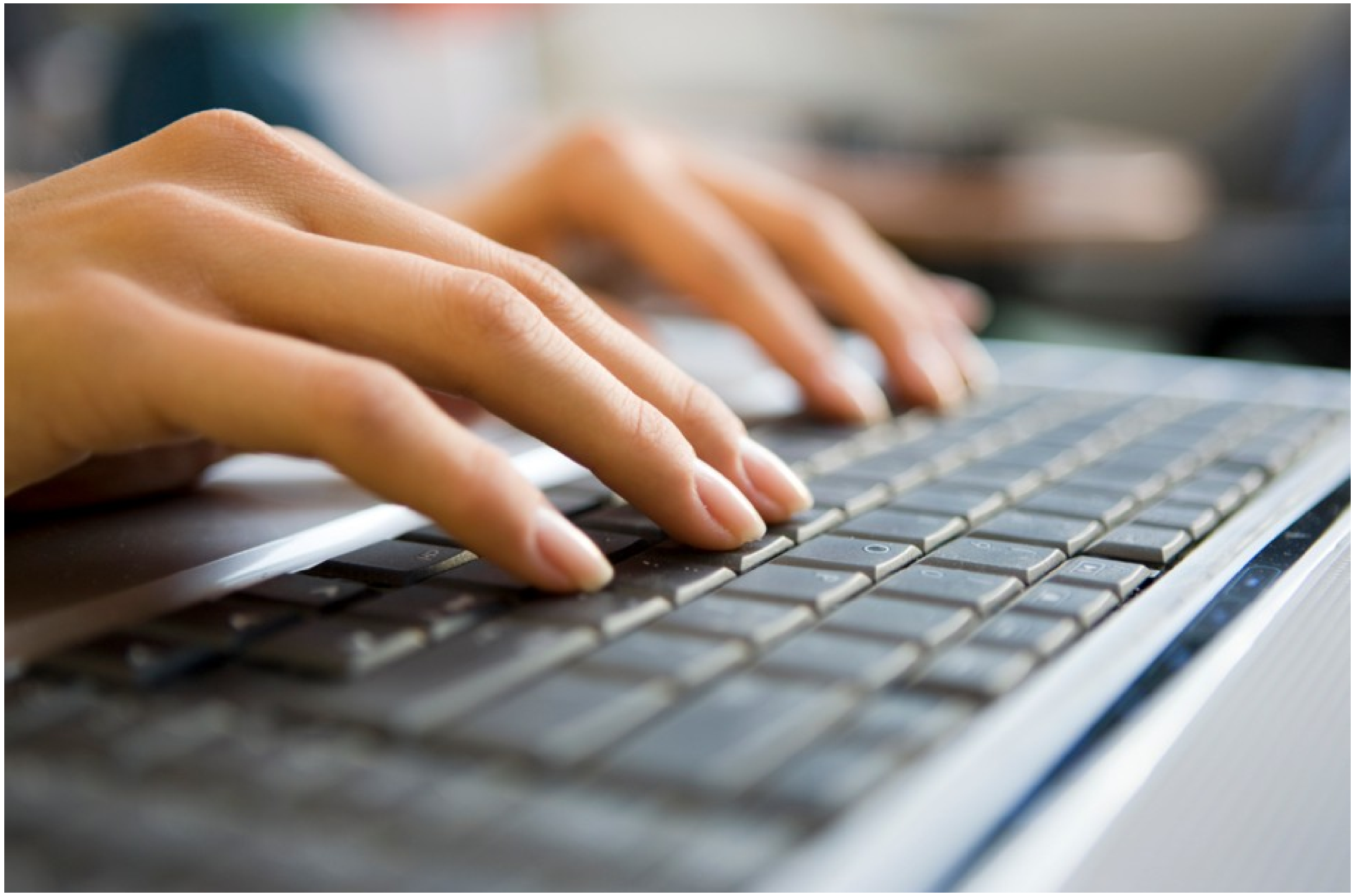
- Preferred by over 100,000 professional developers and DBAs because it saves time.
- Build queries fast, generate instant code, import/export in multiple formats, and much more
- Available for Windows, macOS, and Linux.
- Now with two **NEW tools**:
  - Data Masking
  - Reschema for performance tuning

Robo 3T: the hobbyist GUI

Robo 3T 1.4 brings support for MongoDB 4.2, and a mongo shell upgrade from 4.0 to 4.2, with the ability to manually specify visible databases.

https://robomongo.org/download

MongoDB - HandsOn

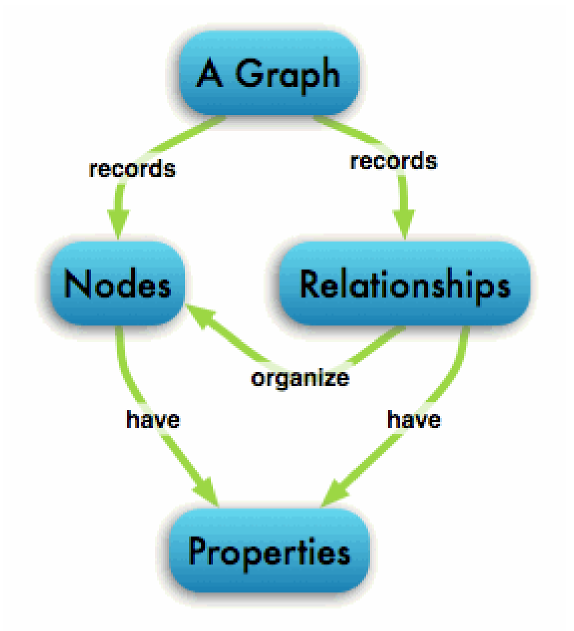


Use Cases

- Any application that needs to use semi-structured data
- Applications with high volume of information
- Document and Content Management Systems
- Rapid development / Agile methodologies
- Machine-generated data (logs, sensors, etc.)
- It is not appropriate when there is more than one data center

Agenda

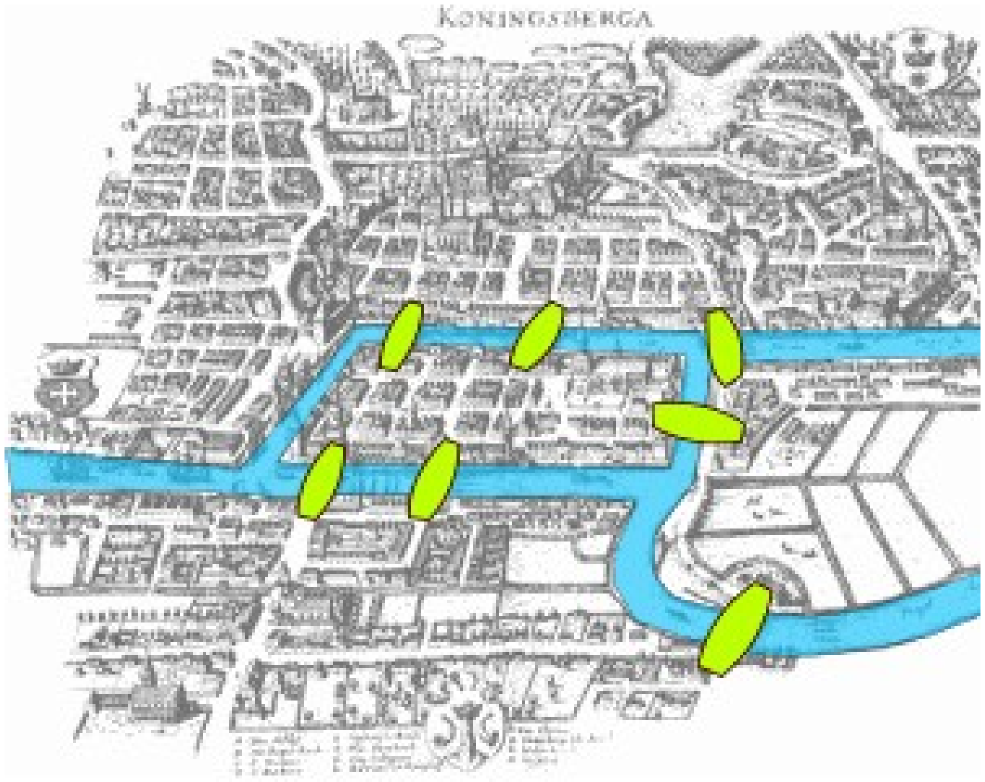
- Material
- Use case
- Relational Databases
- NoSQL
- Riak
- Apache Cassandra
- MongoDB
- Neo4j



Neo4j

- It is a network oriented database (stores information as nodes and relationships)
- Implemented in java in 2010
- Querys: Proprietary language called Cypher that allows you to explore connections between information
- REST Interface
- It is not necessary to declare a scheme
- Prioritizes consistency and availability
- ACID

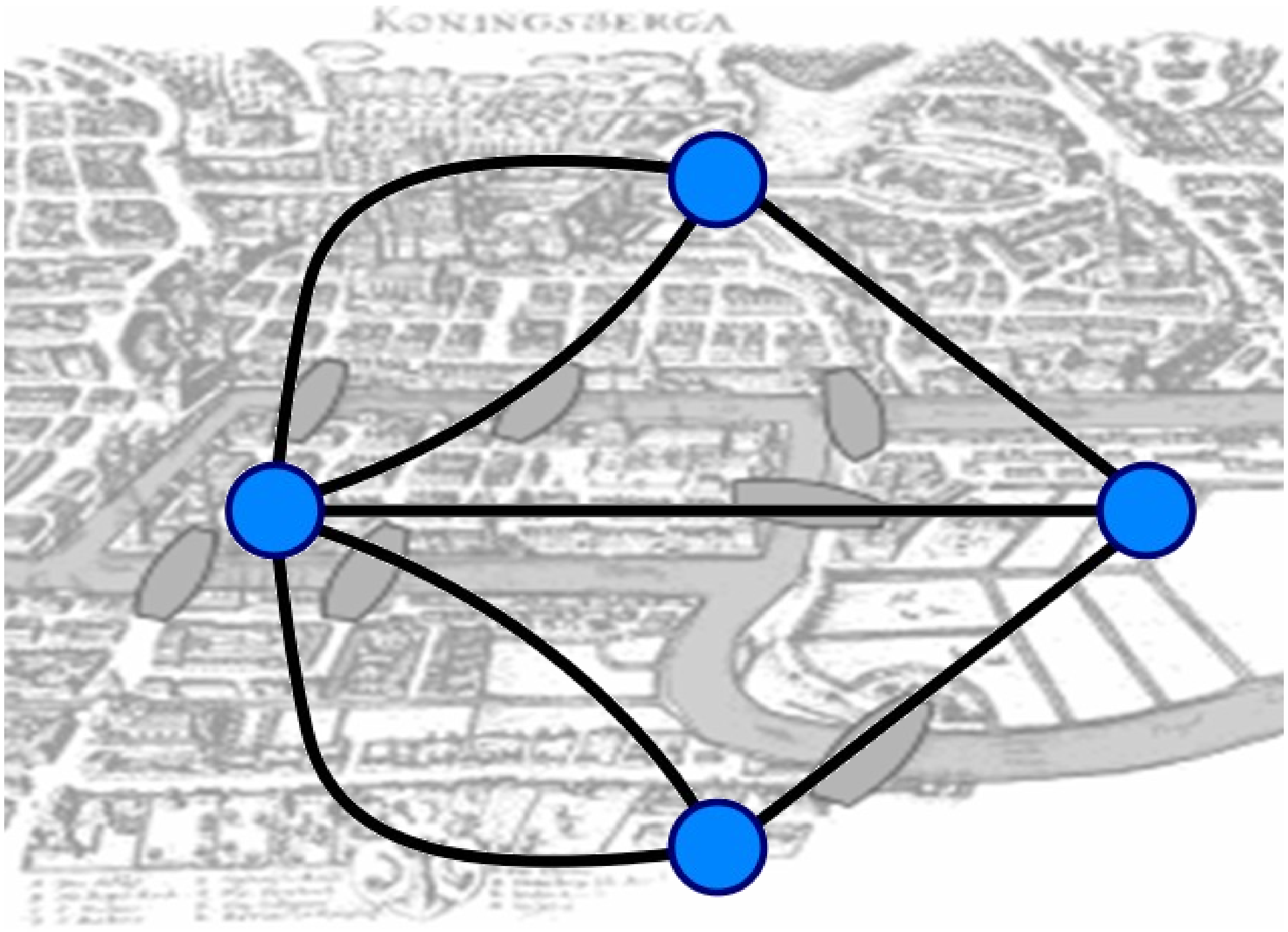
Graphite Theory - Königsberg Bridges



Leonhard Euler
1707-1783

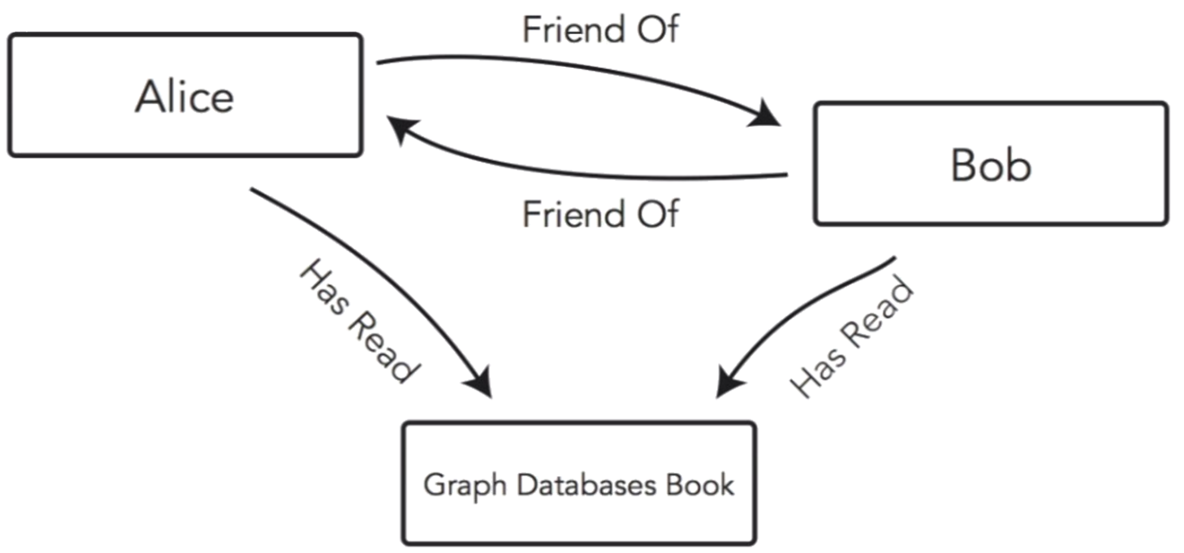
IMMUNE CODING INSTITUTE

Graphite Theory - Königsberg Bridges

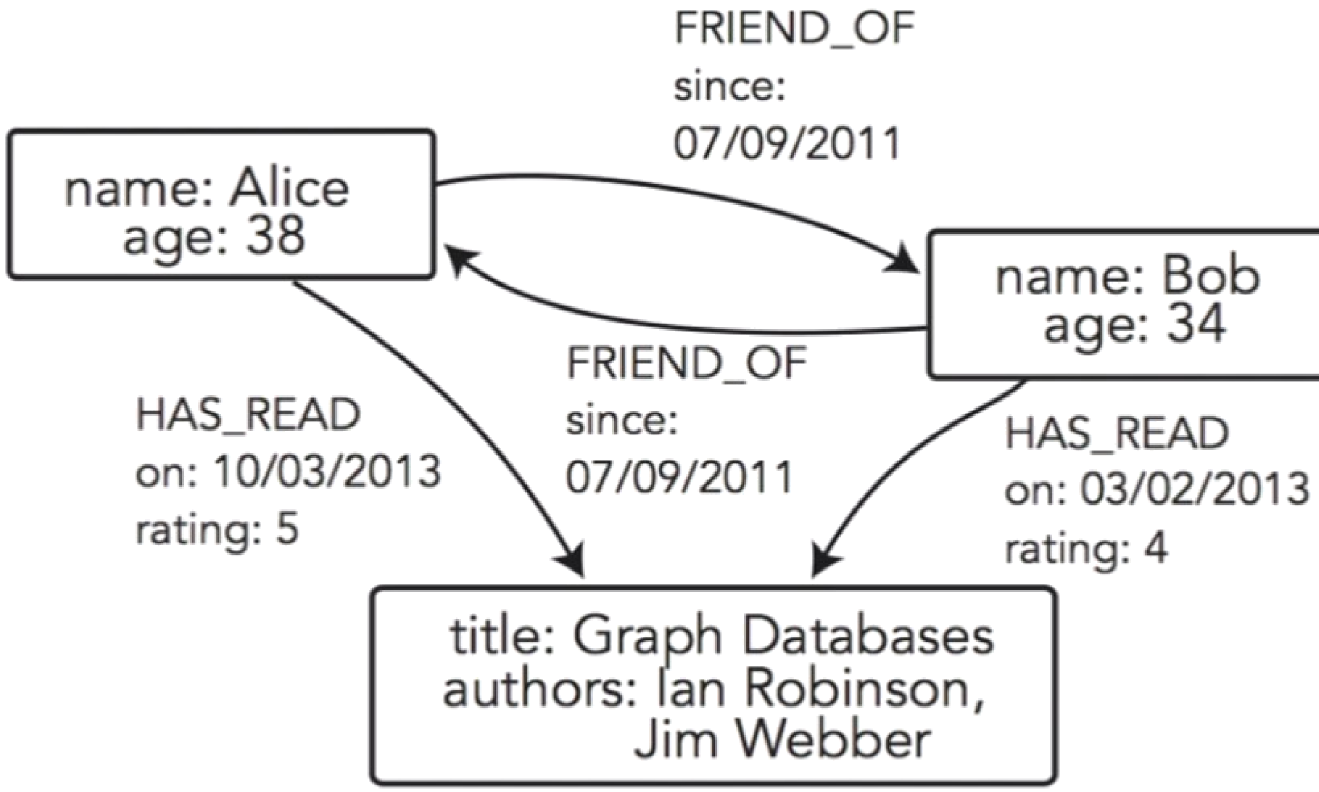


Nodes and Relations

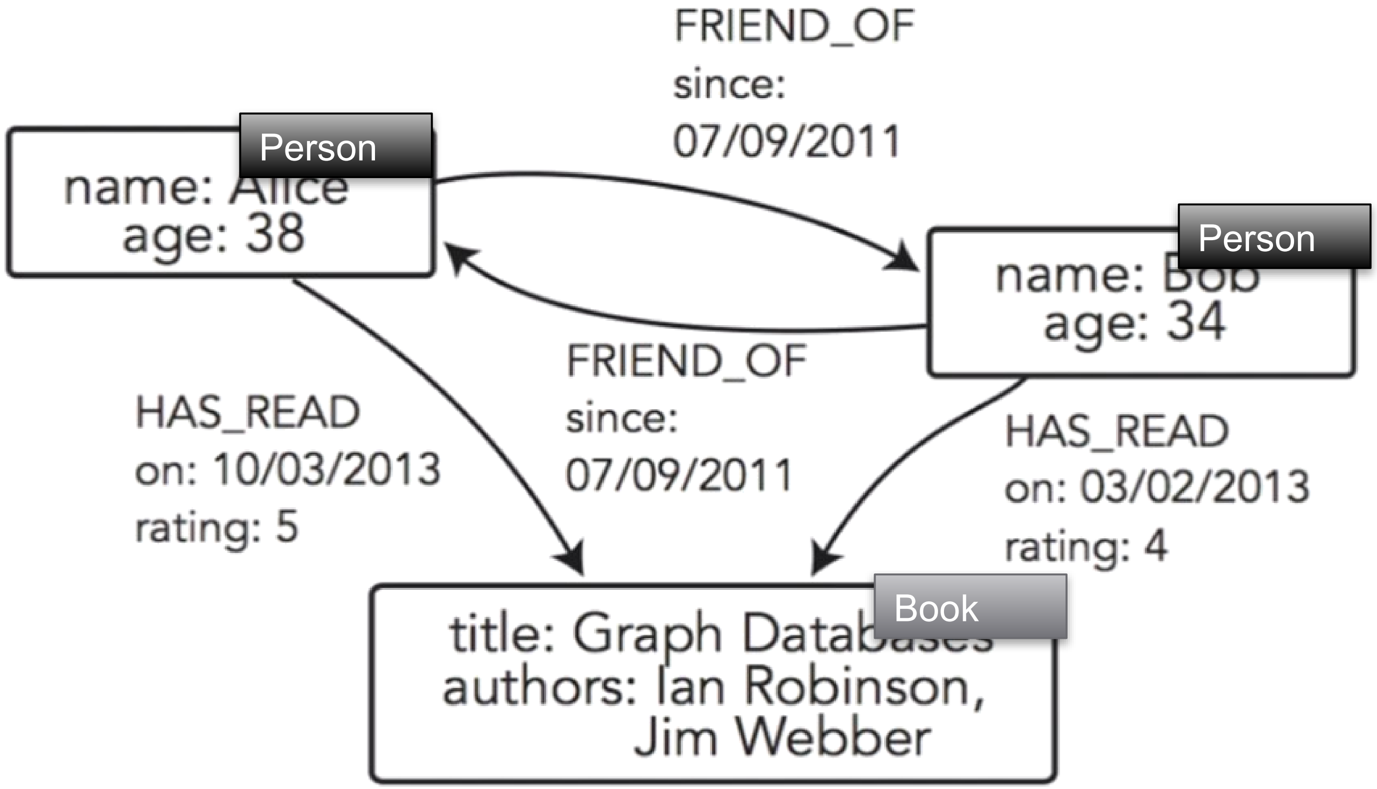
- A network is built the way people really think
- The nodes or vertices represent entities
- The edges represent relationships



Properties

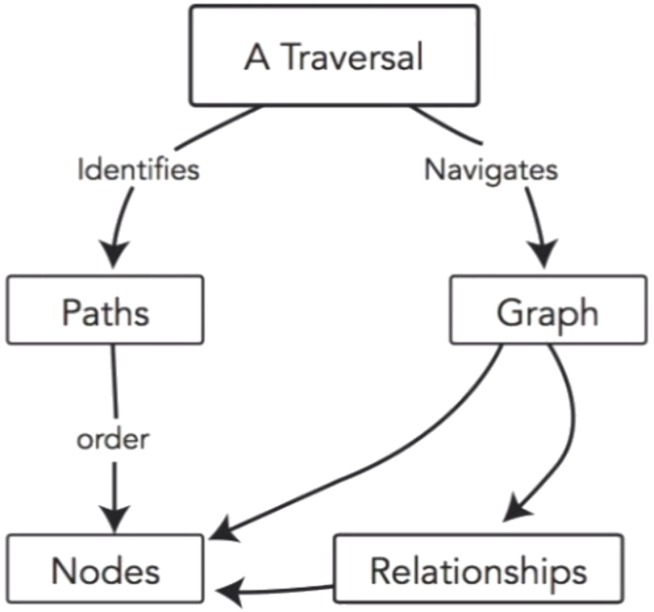

FRIEND_OF
since:
07/09/2011
name: Alice
age: 38
name: Bob
age: 34
FRIEND_OF
since:
07/09/2011
HAS_READ
on: 10/03/2013
rating: 5
HAS_READ
on: 03/02/2013
rating: 4
title: Graph Databases
authors: Ian Robinson,
Jim Webber

Tags


FRIEND_OF
since:
07/09/2011
Person
name: Alice
age: 38
Person
name: Bob
age: 34
FRIEND_OF
since:
07/09/2011
HAS_READ
on: 10/03/2013
rating: 5
HAS_READ
on: 03/02/2013
rating: 4
Book
title: Graph Databases
authors: Ian Robinson,
Jim Webber

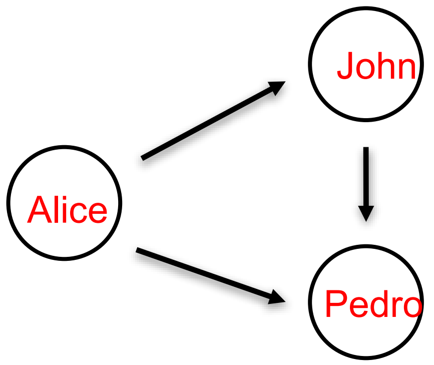
Searching for information

- Locate a node and explore other nodes through their relationships

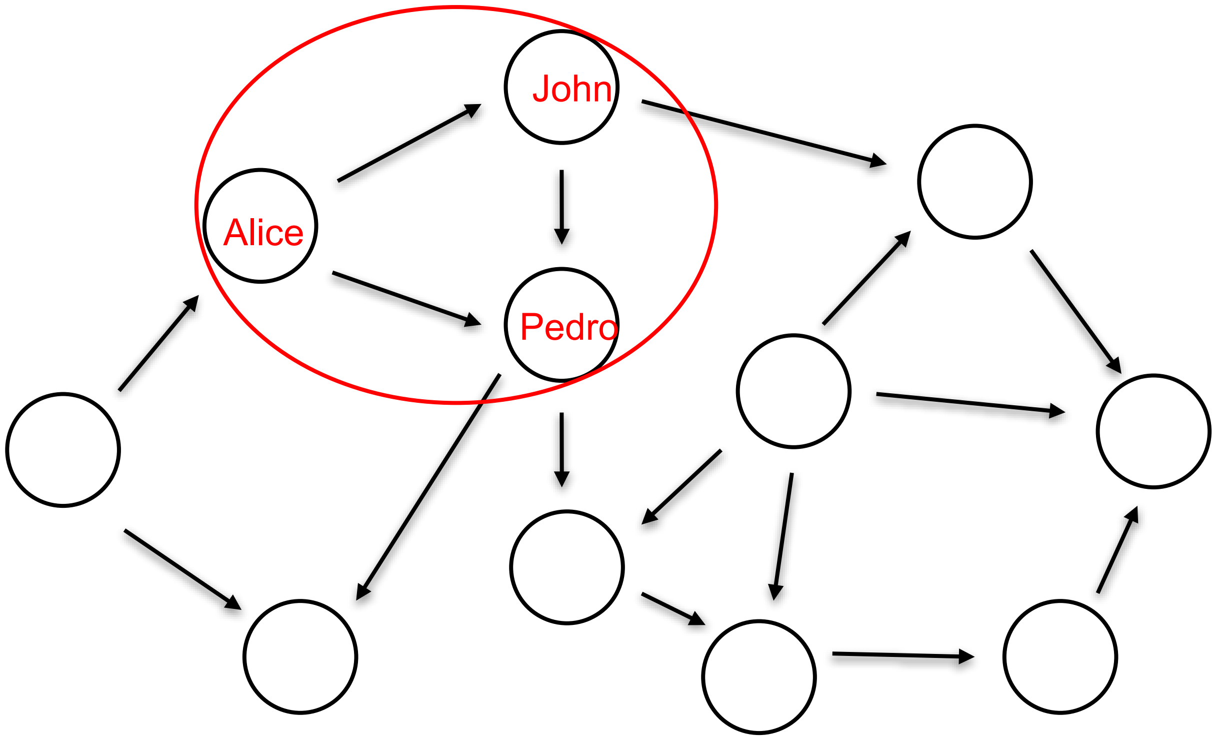


Searching for information

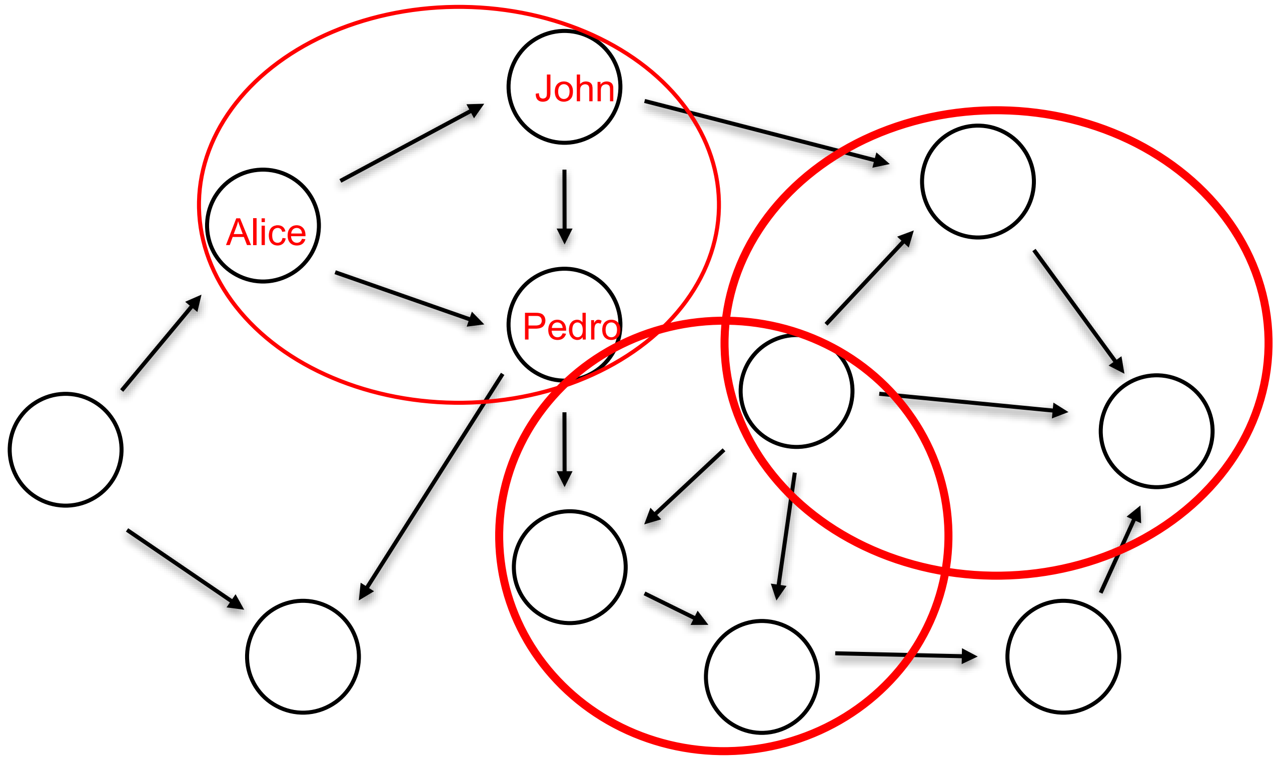
- Locate a node and explore other nodes through their relationships
- or we can identify patterns



Patterns

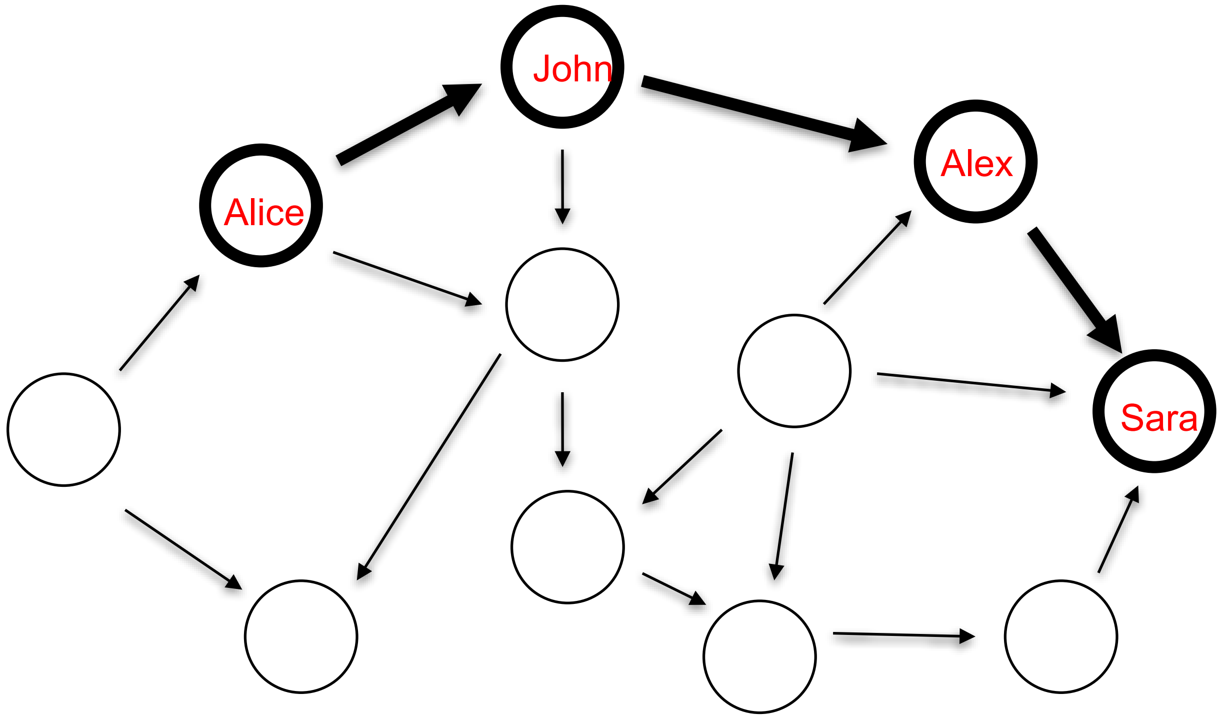


Patterns



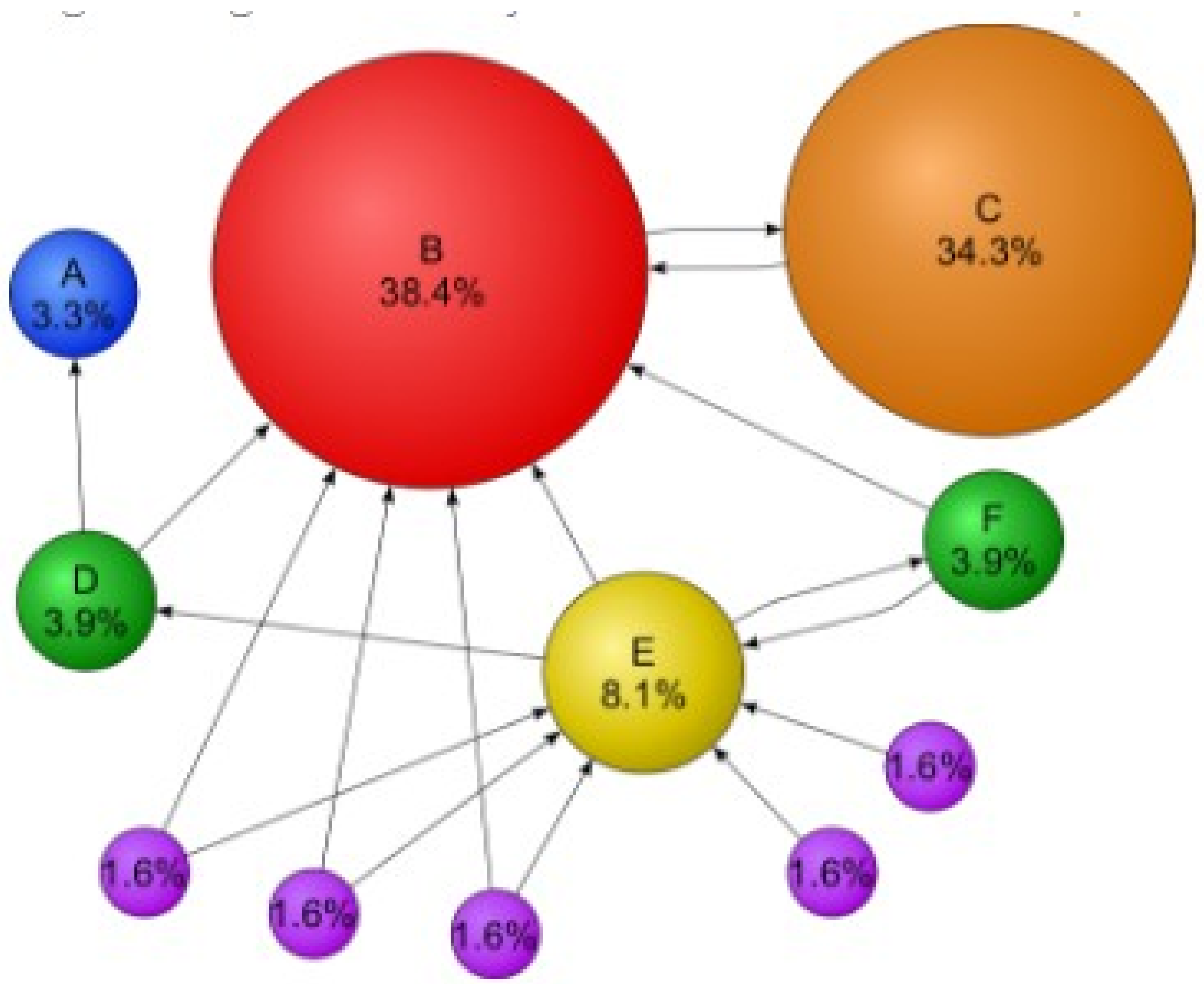
IMMUNE CODING INSTITUTE

Algorithms - The shortest path



IMMUNE CODING INSTITUTE

Algorithms - PageRank



Cypher

- Language with a philosophy similar to SQL
- Allows you to create nodes and/or relationships, maintain them or delete them
- Finding patterns
- Execute algorithms implemented in the DB

```
CREATE (ann:Person { name: 'Ann' })
RETURN ann
```

```
MATCH p=shortestPath(
  (a:Person { name: 'Ann' })-[:KNOWS]-(b:Person { name: 'Dan'})
)
RETURN p
```

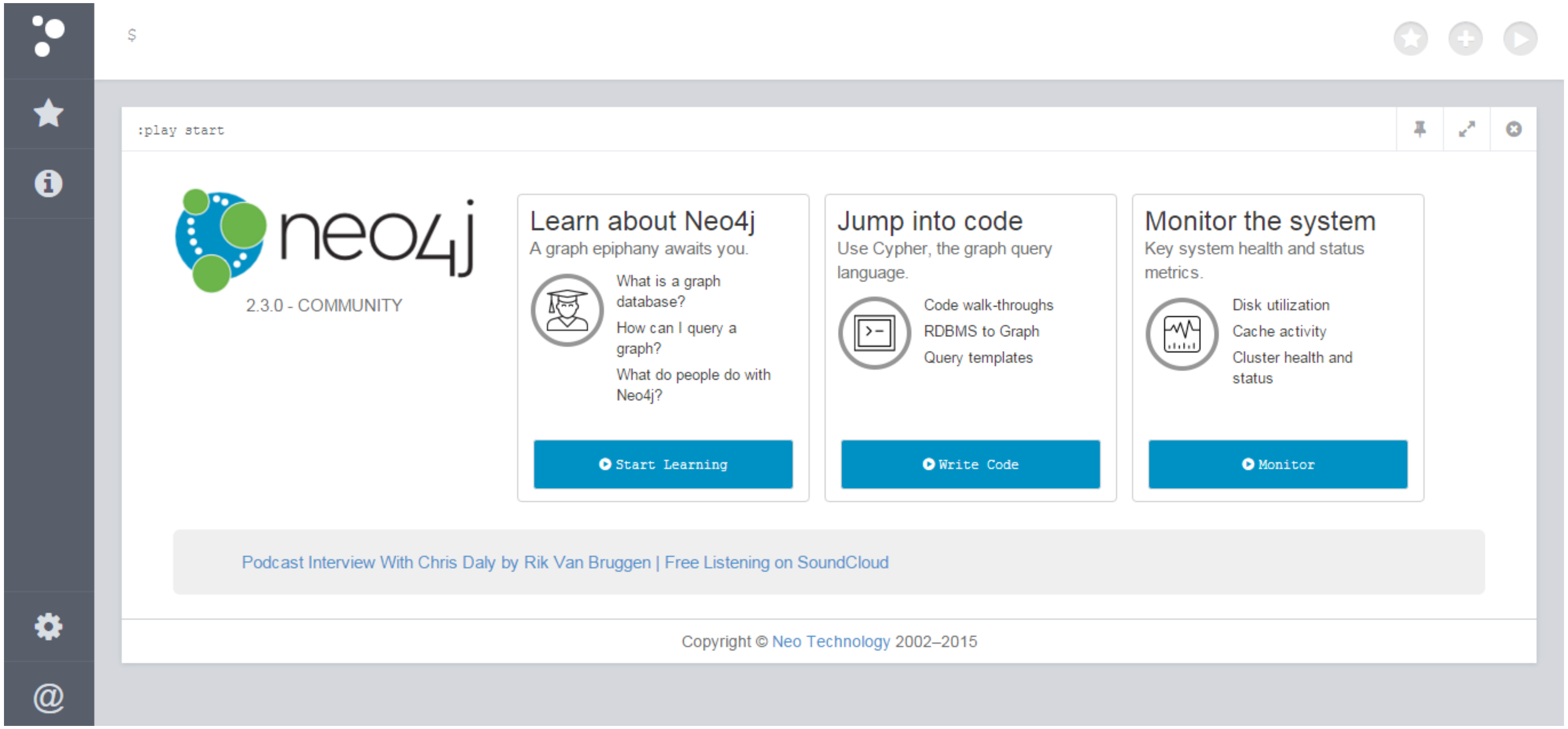
```
MATCH (a:Person { name: 'Ann' }),
     (b:Person { name: 'Dan' })
CREATE (a)-[:KNOWS]->(b)
```

```
MATCH (Alex:Person {name:"Alex"})
DELETE Alex
```

```
MATCH (n:Person { name: 'Ann' })
RETURN n
```

```
MATCH (n:Person {name : "Ann"})
SET n.hair = "Brown"
```

Neo4j


$

:play start

2.3.0 - COMMUNITY

Learn about Neo4j
A graph epiphany awaits you.

What is a graph database?
How can I query a graph?
What do people do with Neo4j?

Jump into code
Use Cypher, the graph query language.

Code walk-throughs
RDBMS to Graph
Query templates

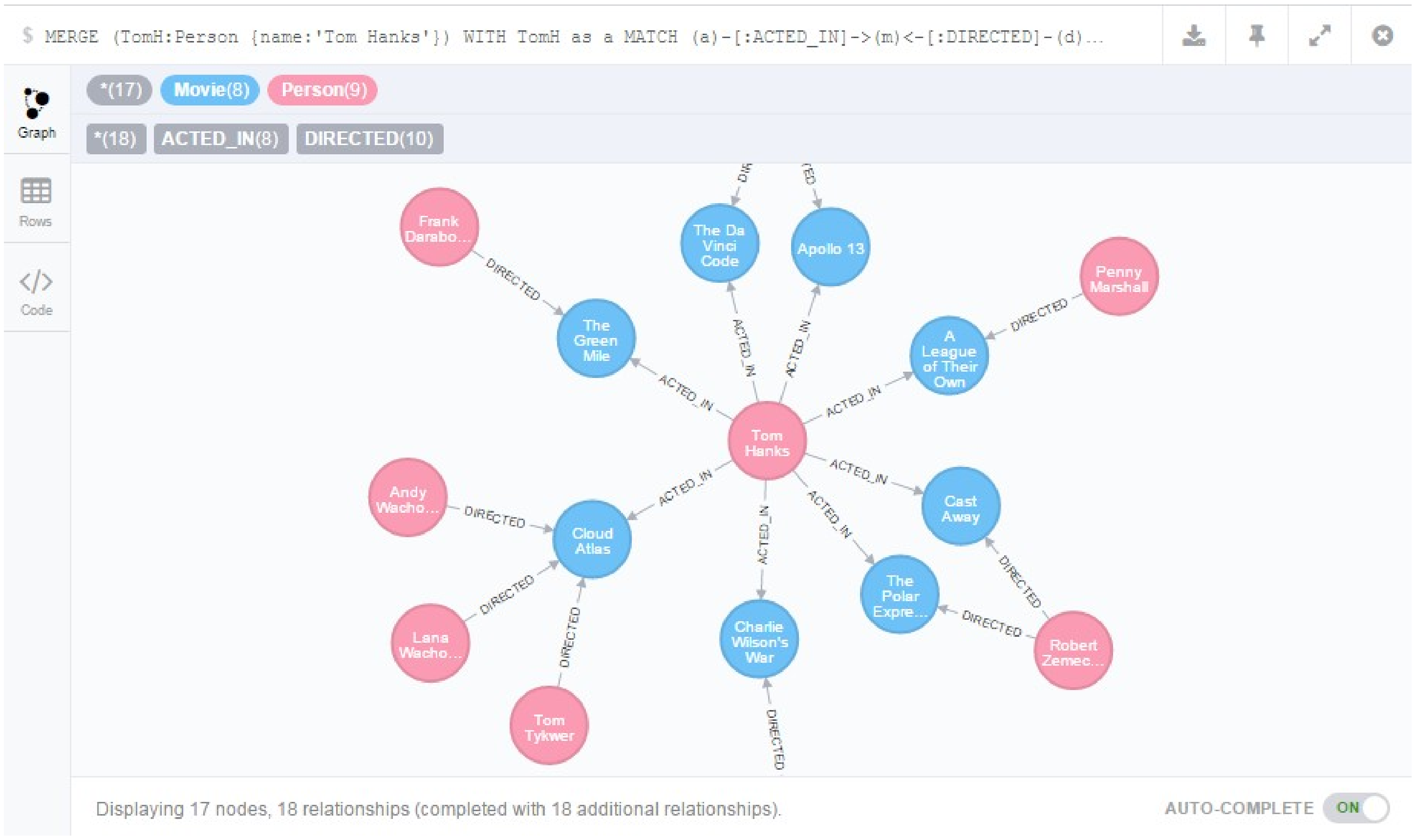
Monitor the system
Key system health and status metrics.

Disk utilization
Cache activity
Cluster health and status

Podcast Interview With Chris Daly by Rik Van Bruggen | Free Listening on SoundCloud

Copyright © Neo Technology 2002–2015

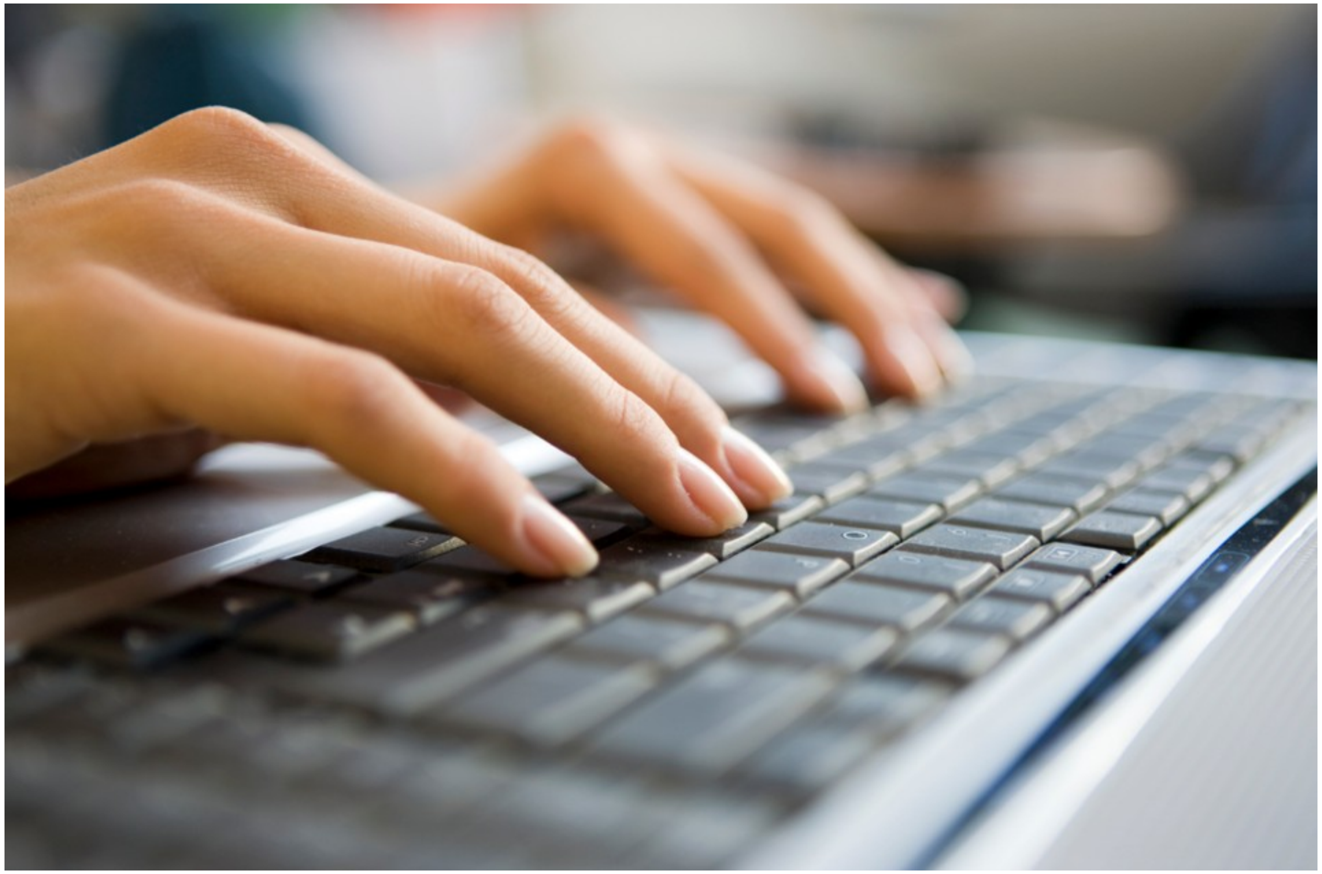
Neo4j - Visualization



```
$ MERGE (TomH:Person {name:'Tom Hanks'}) WITH TomH as a MATCH (a)-[:ACTED_IN]->(m)<-[:DIRECTED]-(d)...
```

Displaying 17 nodes, 18 relationships (completed with 18 additional relationships).

Neo4j - HandsOn



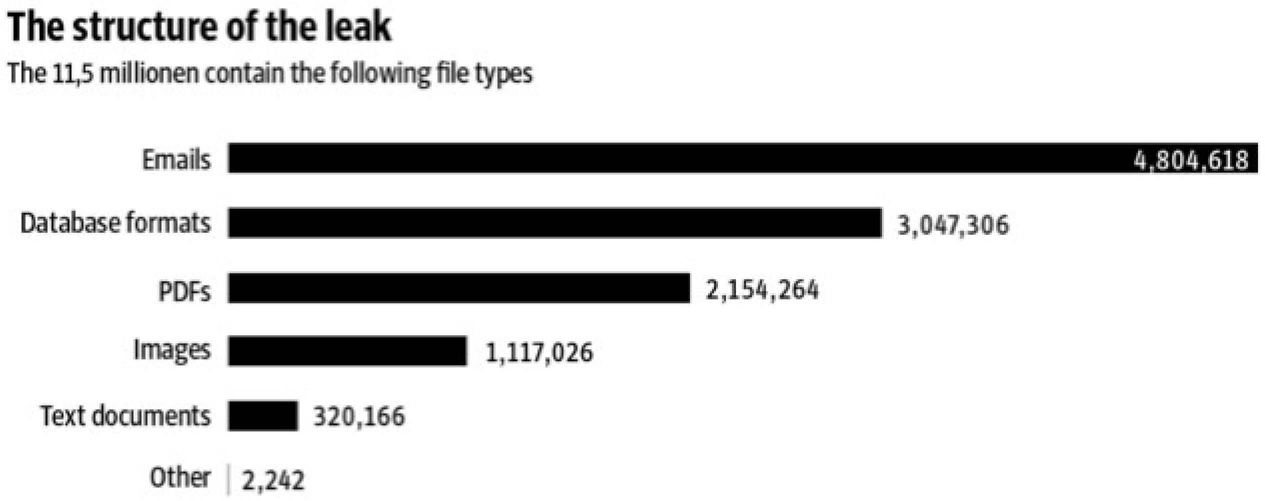
Use Cases

- Optimal for applications that need to look for relationships in information
- Social networking
- Fraud detection by identifying patterns
- Real-time recommendations
- Data center management - devices, users, etc.
- Master Data Systems Management
- Identity and access management

IMMUNE CODING INSTITUTE

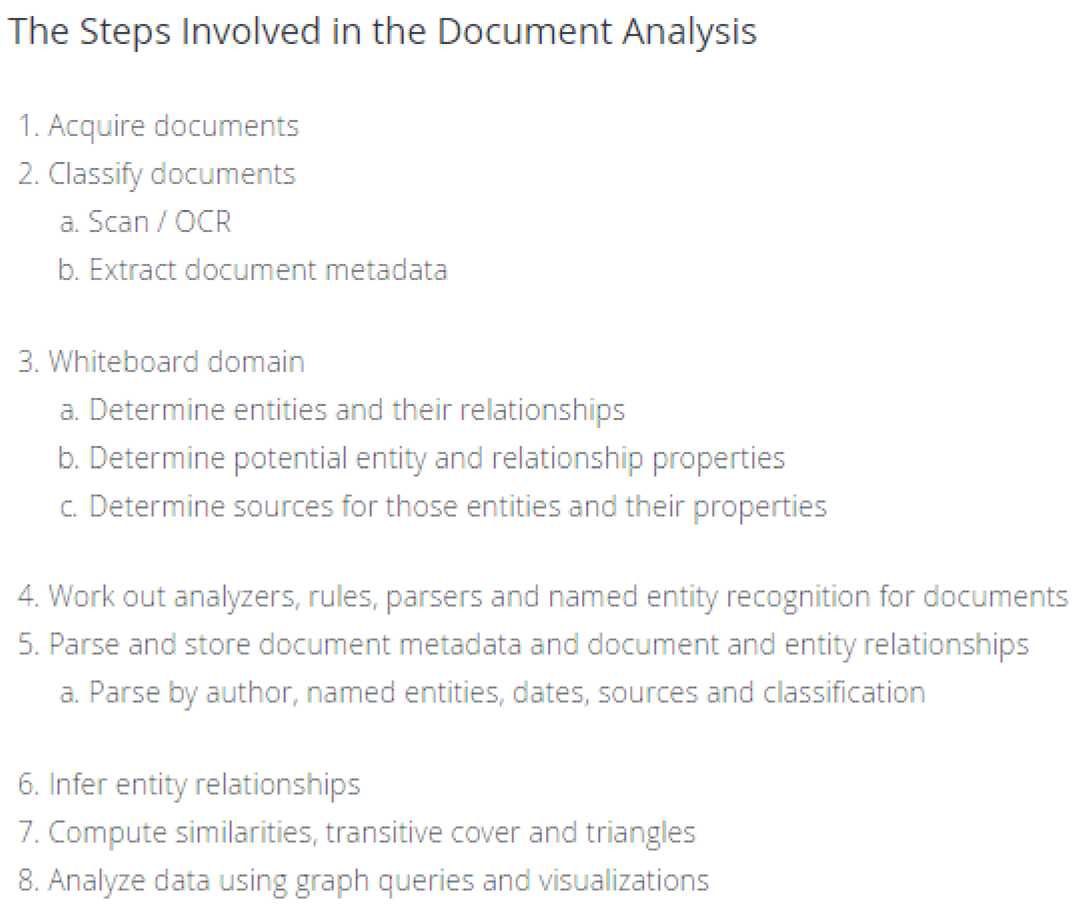
Panama's papers

Panama's papers



IMMUNE
CODING INSTITUTE

Panama's papers



The Steps Involved in the Document Analysis

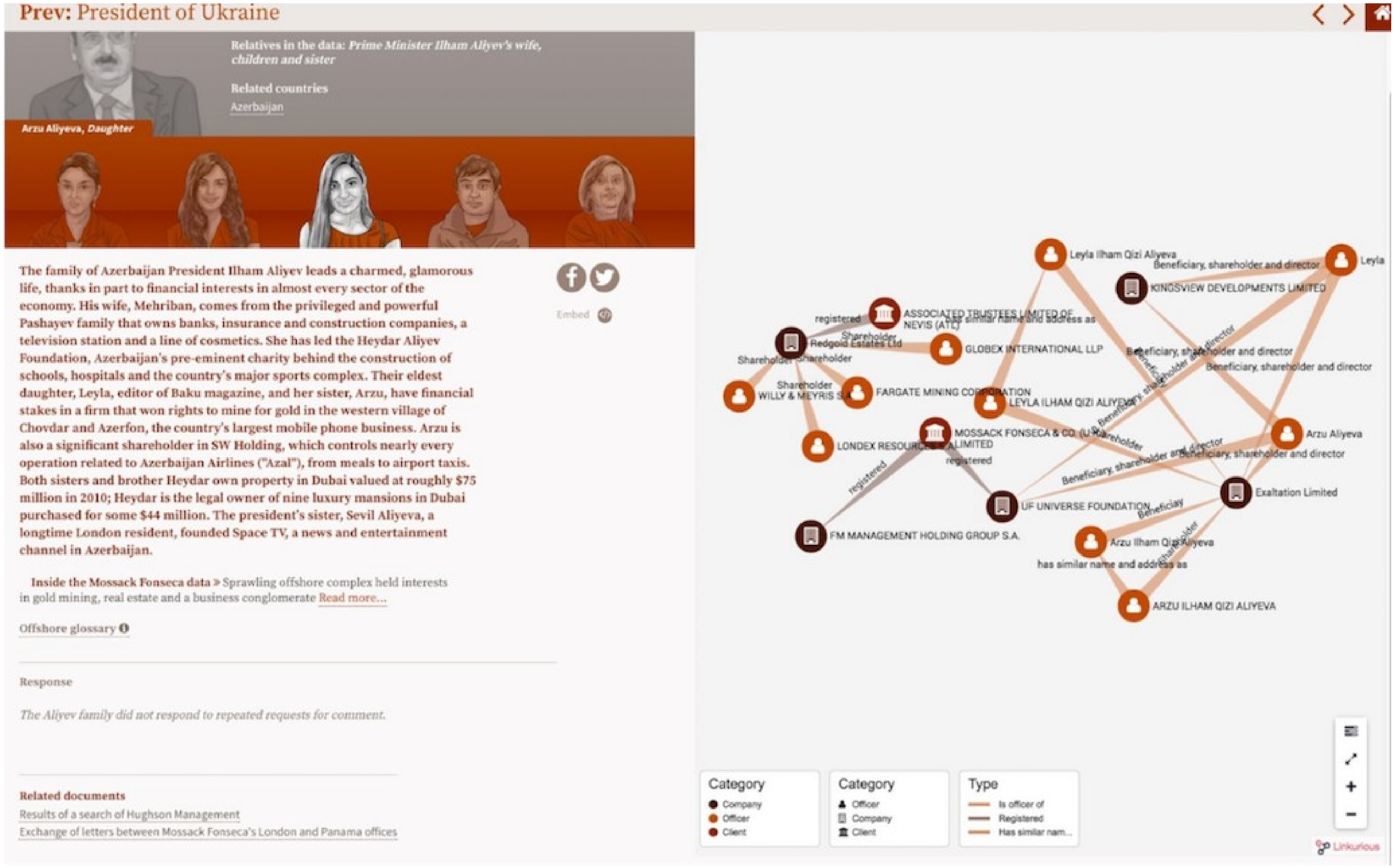
1. Acquire documents
2. Classify documents
    a. Scan / OCR
    b. Extract document metadata

3. Whiteboard domain
    a. Determine entities and their relationships
    b. Determine potential entity and relationship properties
    c. Determine sources for those entities and their properties

4. Work out analyzers, rules, parsers and named entity recognition for documents
5. Parse and store document metadata and document and entity relationships
    a. Parse by author, named entities, dates, sources and classification

6. Infer entity relationships
7. Compute similarities, transitive cover and triangles
8. Analyze data using graph queries and visualizations

Panama's papers



Prev: President of Ukraine

Relatives in the data: *Prime Minister Ilham Aliyev's wife, children and sister*

Related countries

Azerbaijan

Arzu Aliyeva, Daughter

The family of Azerbaijan President Ilham Aliyev leads a charmed, glamorous life, thanks in part to financial interests in almost every sector of the economy. His wife, Mehriban, comes from the privileged and powerful Pashayev family that owns banks, insurance and construction companies, a television station and a line of cosmetics. She has led the Heydar Aliyev Foundation, Azerbaijan's pre-eminent charity behind the construction of schools, hospitals and the country's major sports complex. Their eldest daughter, Leyla, editor of Baku magazine, and her sister, Arzu, have financial stakes in a firm that won rights to mine for gold in the western village of Chovdar and Azerfon, the country's largest mobile phone business. Arzu is also a significant shareholder in SW Holding, which controls nearly every operation related to Azerbaijan Airlines ("Azal"), from meals to airport taxis. Both sisters and brother Heydar own property in Dubai valued at roughly $75 million in 2010; Heydar is the legal owner of nine luxury mansions in Dubai purchased for some $44 million. The president's sister, Sevil Aliyeva, a longtime London resident, founded Space TV, a news and entertainment channel in Azerbaijan.

Inside the Mossack Fonseca data > Sprawling offshore complex held interests in gold mining, real estate and a business conglomerate Read more...

Offshore glossary

Embed

Response

The Aliyev family did not respond to repeated requests for comment.

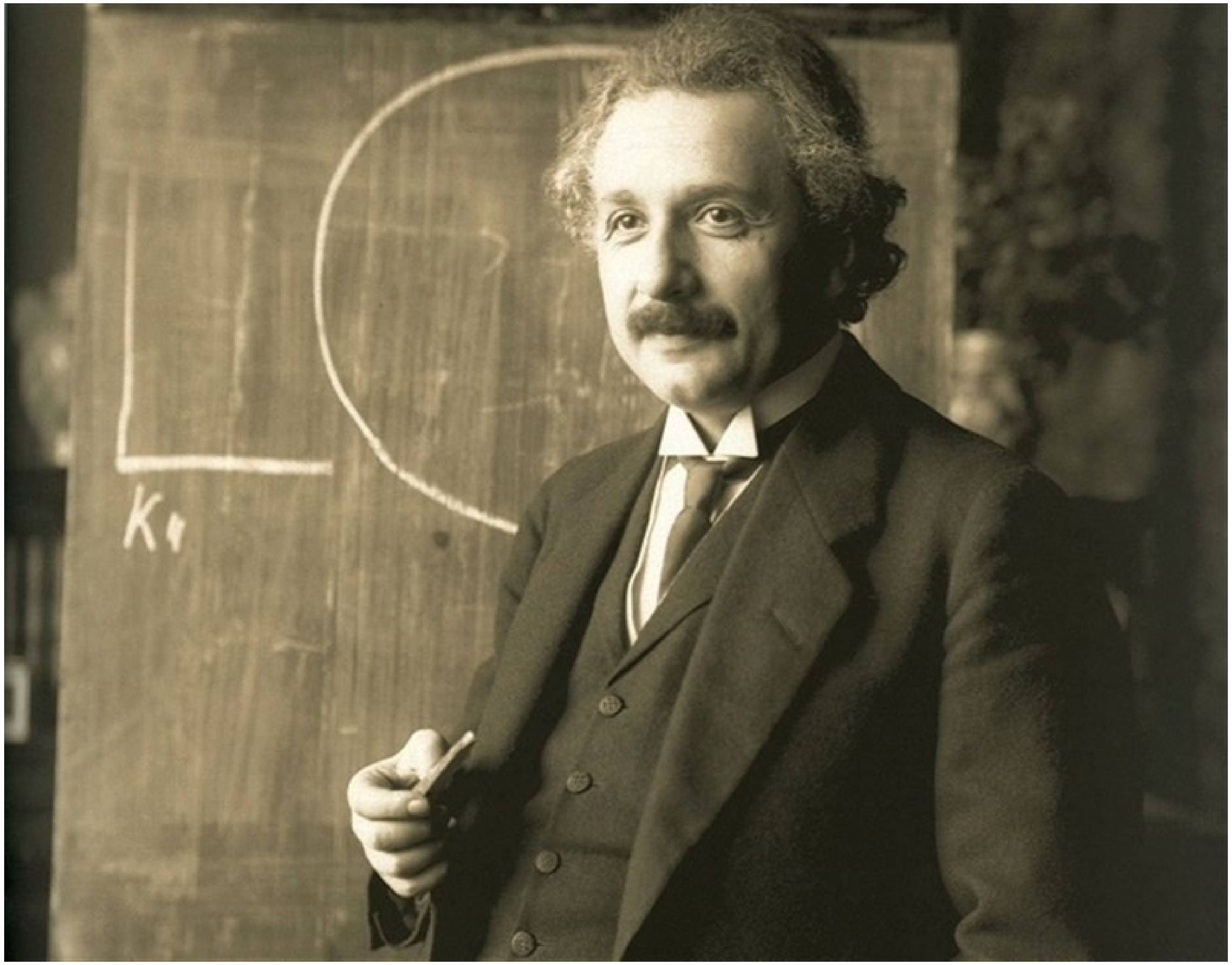
Related documents

Results of a search of Hughson Management

Exchange of letters between Mossack Fonseca's London and Panama offices

Category	Category	Type
● Company	Officer	Is officer of
● Officer	Company	Registered
● Client	Client	Has similar nam...

Einstein's Riddle



Einstein's Riddle

- This seemingly simple Einstein's riddle is based on a number of considerations and one question.
- These are about a group of five people of different nationalities, with five different pets, consuming a certain brand of tobacco, drinking a certain drink and living in a different house entirely in each case.

Who owns the fish?

IMMUNE CODING INSTITUTE

Einstein's Riddle

- The Englishman lives in the red house.
- The Swede has a dog.
- The Dane drinks tea.
- The Norwegian lives in the first house.
- The German smokes Prince.
- The green house is immediately to the left of the white one.
- The owner of the green house drinks coffee.

Einstein's Riddle

- The person who smokes Pall Mall breeds birds.
- The owner of the yellow house smokes Durnhill.
- The man who lives in the house downtown drinks milk.
- The man who smokes Blends lives next door to the man who has a cat.
- The man who has a horse lives next to the man who smokes Dunhill.
- The man who smokes Bluemaster drinks beer.
- The man who smokes Blends is a neighbor of the man who drinks water.
- The Norwegian lives next door to the blue house.

THANKS FOR YOUR ATTENTION

Daniel Villanueva Jiménez
daniel.villanueva@immune.institute
@dvillaj