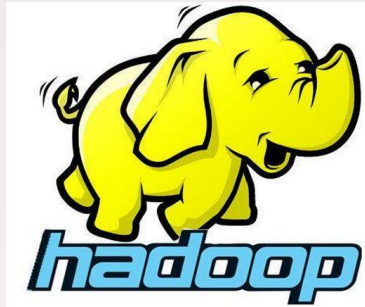




mongoDB



Büyük Veri & NoSQL Veritabanları

Özal YILDIRIM

Debanjan Mahata

Department of Information Science

University of Arkansas

İçerik

- Büyük Veri
- Hadoop
- Map Reduce
- NoSQL ?
- MongoDB
- Neo4j

Olası Ders Çıktıları

Hiç bir şey anlamadım ya!



Anladım!!!

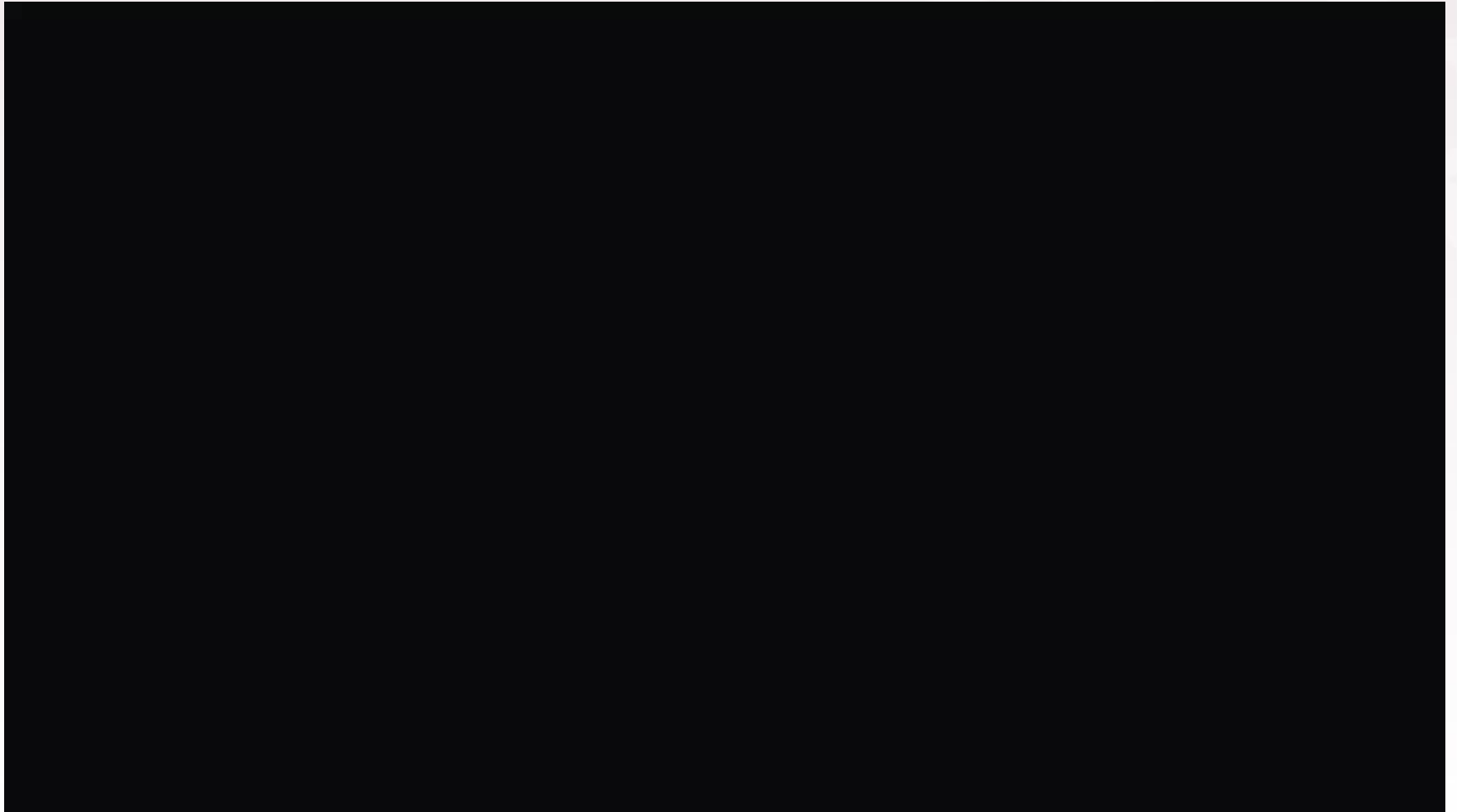


veya

Dersi kaçıranlar...



Big Data



Hadoop

Hadoop, yalın tabiriyle, sıradan sunucularda büyük verileri işlemek amacıyla kullanılan açık kaynak kodlu bir kütüphanedir.

Her türlü veri için devasa depolama, çok yüksek işlem gücü ve neredeyse sınırsız sayıda eşzamanlı görevleri yönetme yeteneği sağlar. Dağınık bir bilgi işlem ortamında büyük verileri verimli bir şekilde yönetmenizi ve işlemenizi mümkün kılar.

Apache projesidir...

Java ile geliştirilmiş açık kaynaklı bir kütüphanedir



Hadoop dört ana modülden oluşur.

- **Hadoop Common**
- **HDFS**
- **MapReduce**
- **YARN**

Hadoop şu anda Yahoo, Amazon, eBay, Facebook, LinkedIn gibi birçok lider firmada büyük verileri analiz etmek amacıyla kullanılıyor

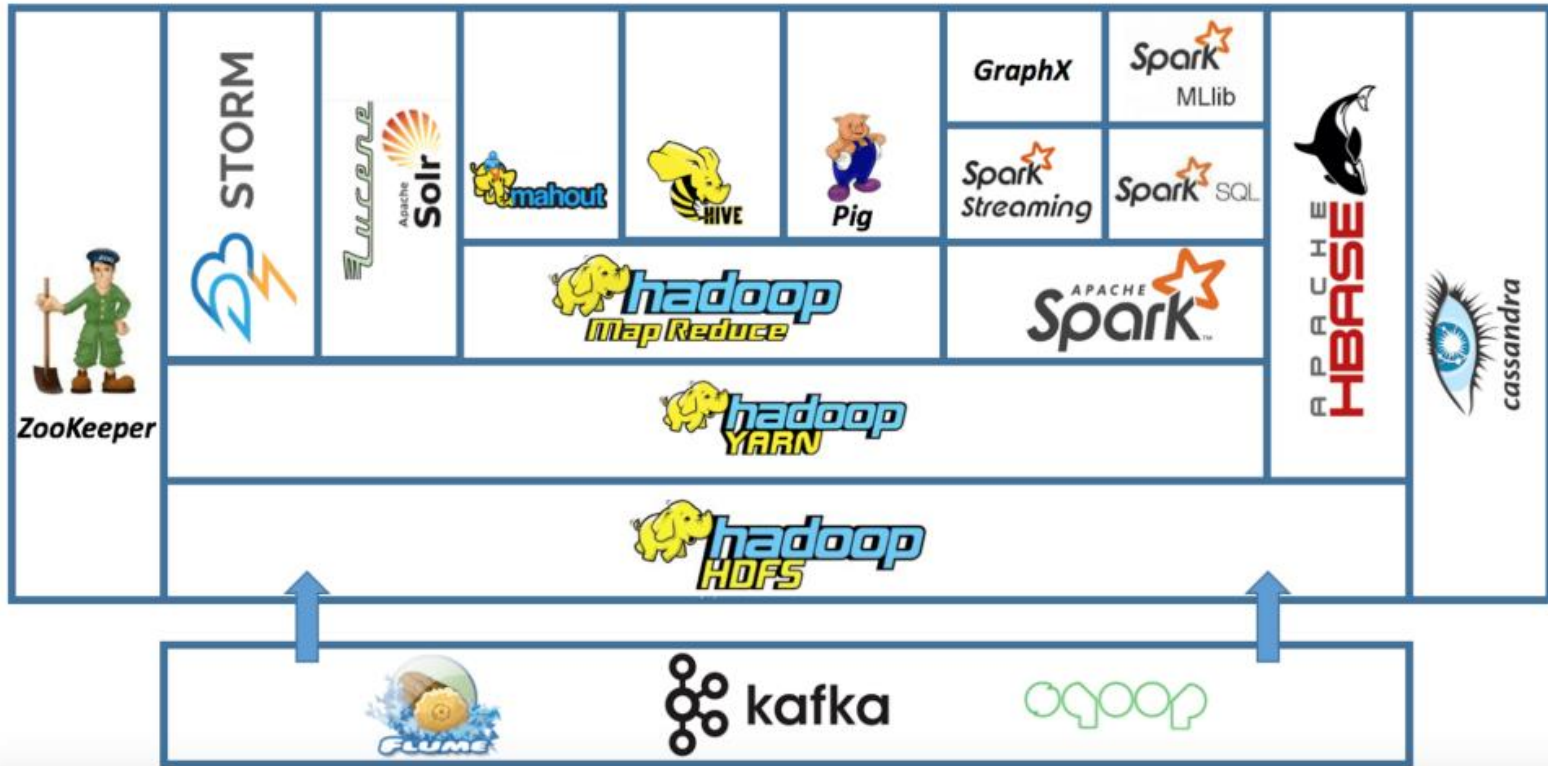
- [A9.com](#) - Amazon*
 - Akış API'sini ve önceden var olan C++, Perl ve Python araçlarını kullanarak Amazon'un ürün arama dizinlerini oluşturuyoruz.
 - Hem Java hem de akış API'lerini kullanarak analitik için günlük olarak milyonlarca oturum işliyoruz.
 - Kümelerimiz 1 ila 100 düğüm arasında değişir

[ARA.COM.TR](#) - Ara Com Tr - Türkiye'nin ilk ve tek arama motoru

- [Ara.com.tr](#) arama motorunu Python araçlarını kullanarak oluşturuyoruz.
- Analitik için Apache Hadoop kullanıyoruz.
- Ayda yaklaşık 400 TB işliyoruz
- Kümelerimiz 10 ila 100 düğüm arasında değişir

- [Facebook](#)
 - Dahili günlük ve boyut veri kaynaklarının kopyalarını depolamak için Apache Hadoop kullanıyoruz ve bunu raporlama/analitik ve makine öğrenimi için bir kaynak olarak kullanıyoruz.
 - Şu anda 2 ana kümemiz var:
 - 8800 çekirdeğe ve yaklaşık 12 PB ham depolamaya sahip 1100 makinelik bir küme.
 - 2400 çekirdeğe ve yaklaşık 3 PB ham depolamaya sahip 300 makinelik bir küme.
 - Her (emtia) düğümün 8 çekirdeği ve 12 TB depolama alanı vardır.
 - Hem akışın hem de Java API'lerinin yoğun kullanıcılarıyız. Hive adı verilen bu özellikleri kullanarak daha üst düzey bir veri ambarı çerçevesi oluşturduk (bkz. <http://hadoop.apache.org/hive/>). Ayrıca HDFS üzerinden bir FUSE uygulaması geliştirdik.

EKOSİSTEM



Avro (veri dizileştirme (serialization) sistemi), **Cassandra** (yüksek erişilebilir, ölçeklenebilir NoSQL veritabanı), **HBase** (Hadoop üzerinde çalışan, büyük veriler için ölçeklenebilir, dağıtık NoSQL veritabanı), **Hive** (büyük veriler üzerinde iş zekası sistemi), **Mahout** (ölçeklenebilir yapay öğrenme (machine learning) ve veri madenciliği kütüphanesi), **Pig** (paralel hesaplamalar için yüksek düzeyli bir veri akışı dil ve yürütme kütüphanesi), **ZooKeeper** (dağıtık uygulamalar için yüksek ölçekli koordinasyon uygulaması) projeleri geliştiriliyor.

HDFS (Hadoop Distributed File System- Hadoop Dağıtılmış Dosya Sistemi)

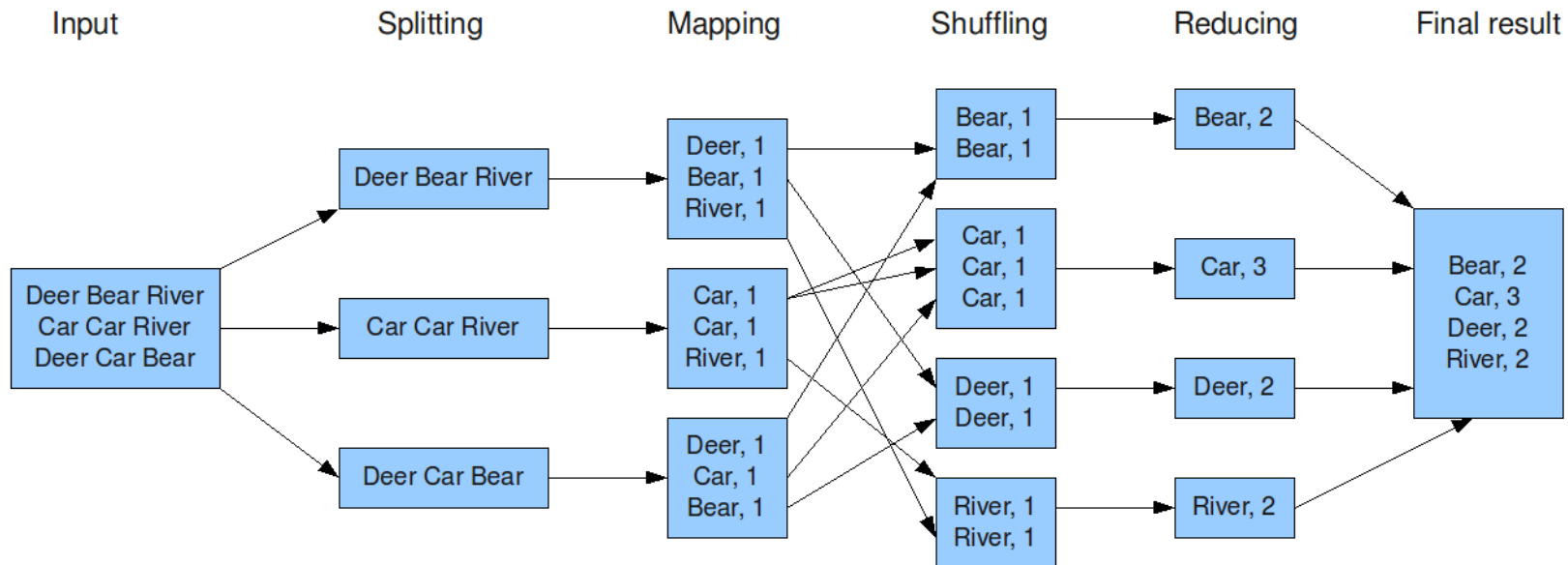
Hataya dayanıklı, güvenilir ve ölçeklenebilir veri depolama yapabilen, kendi kendini onaran dağıtılmış bir dosya sistemi. Basit sunucuların disklerini bir araya getirerek bile büyük bir disk alanı oluşturabilmekte.

Map Reduce

HDFS üzerindeki büyük dosyaları verileri işleyebilmek amacıyla kullanılan bir yöntemdir. İstedığınız verileri filtrelemek için kullanılan Map fonksiyonu ve bu verilerden sonuç elde etmenizi sağlayan Reduce fonksiyonlarından oluşan program yazıldıktan sonra Hadoop üzerinde çalıştırılır.

Kelime Sayma (Map/Reduce)

The overall MapReduce word count process



Veritabanı Savaşları

Veritabanı savaşlarının olduğu bir dönemdeyiz. Yeni veritabanlarıyla dikkat çeken NoSQL mühendisleri, kudretli SQL İmparatorluğu'na karşı ilk zaferlerini kazandılar. Savaş sırasında NoSQL mühendisleri, SQL imparatorluğunu şaşırtacak şekilde web ölçeğindeki verileri işleyebilecek açık kaynaklı altyapılar oluşturmayı başardılar.

SQL imparatorluğunun boş gururundan yararlanan NoSQL mühendisleri, **Neo4j**, **MongoDB**, **Hbase**, **Cassandra**.... gibi çok sayıda NoSQL veri tabanı oluşturarak teknoloji alanında yarıştı ve bunları Google, Facebook, Twitter gibi endüstri devlerinin arkasında destekleyici altyapı olarak kullandı...

Ve savaş başlar...



SQL veritabanı

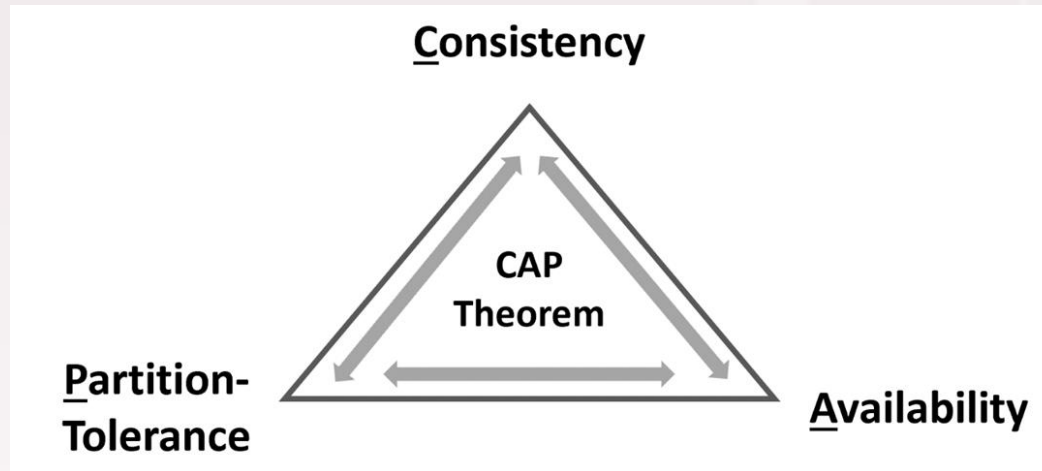
- İlişkisel veritabanı yönetim sistemi
- Ana odak noktası ACID
 - **Atomicity** – Her işlem atomiktir. Bir kısmı başarısız olursa, tüm işlem başarısız olur (ve geri alınır)
 - **Consistency** (tutarlılık)– Her işlem tutarlı bir dizi kurala tabidir (kısıtlamalar, tetikleyiciler, basamaklar)
 - **Isolation** – Hiçbir işlem başka bir işleme müdahale etmemelidir
 - **Durability** (dayanıklılık)– Bir işlem taahhüt edildikten sonra, taahhüt edilmiş olarak kalır.



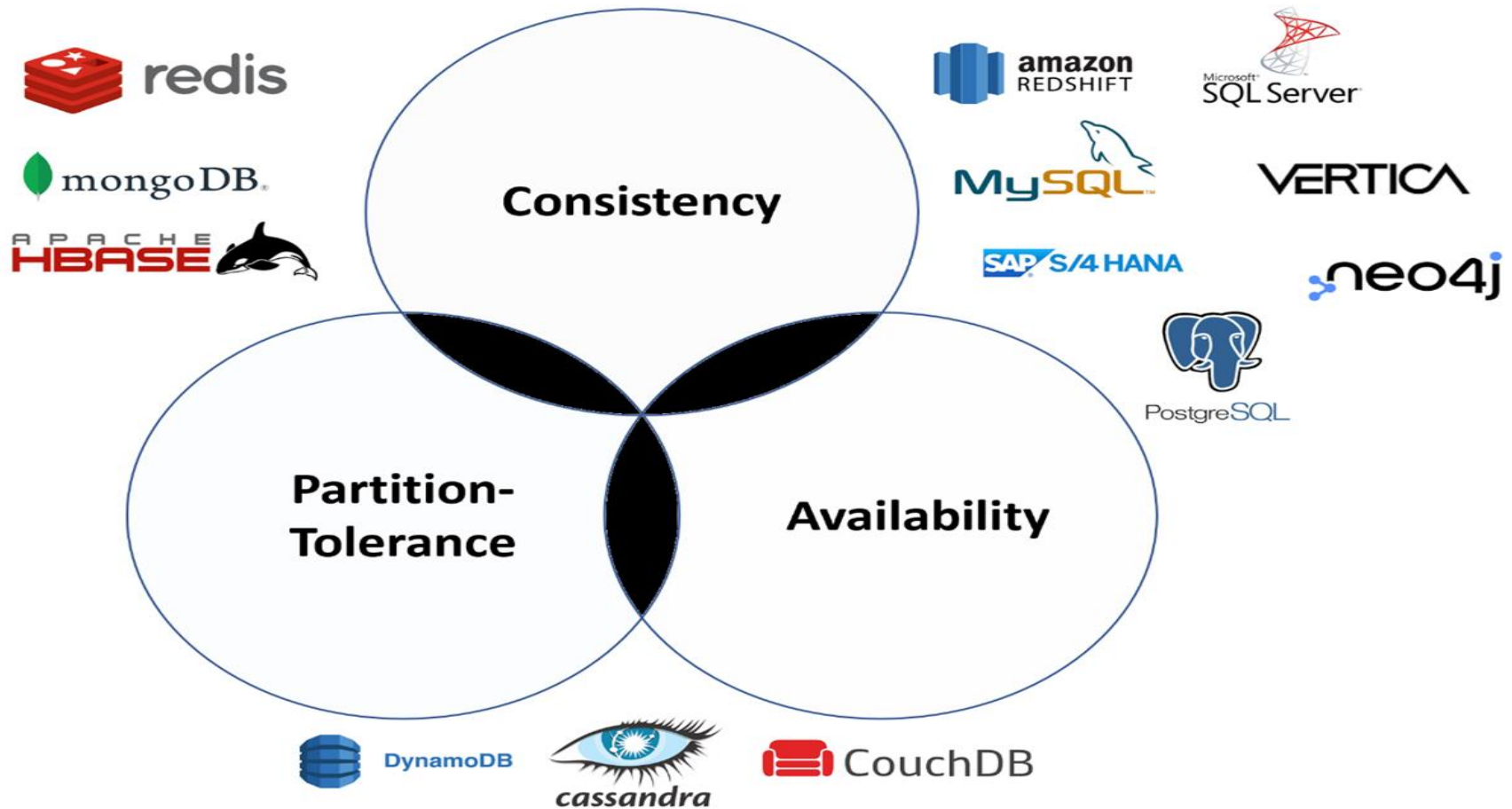
SQL veritabanı

- ACID önemli
 - Ama sadece önemli olduğunda...
 - Bankacılık, finans, güvenlik sistemleri, vb.
 - İnsanların 30 yıl önce bilgisayarlarla inşa ettiği sistem türleri (ve günümüzde)
- ACID ek yük getirir
 - Atomiklik, yalıtım gibi özellikler temel olarak veritabanı sunucularını sıralı değerlendirme yapmaya zorlar.

CAP Teoremi

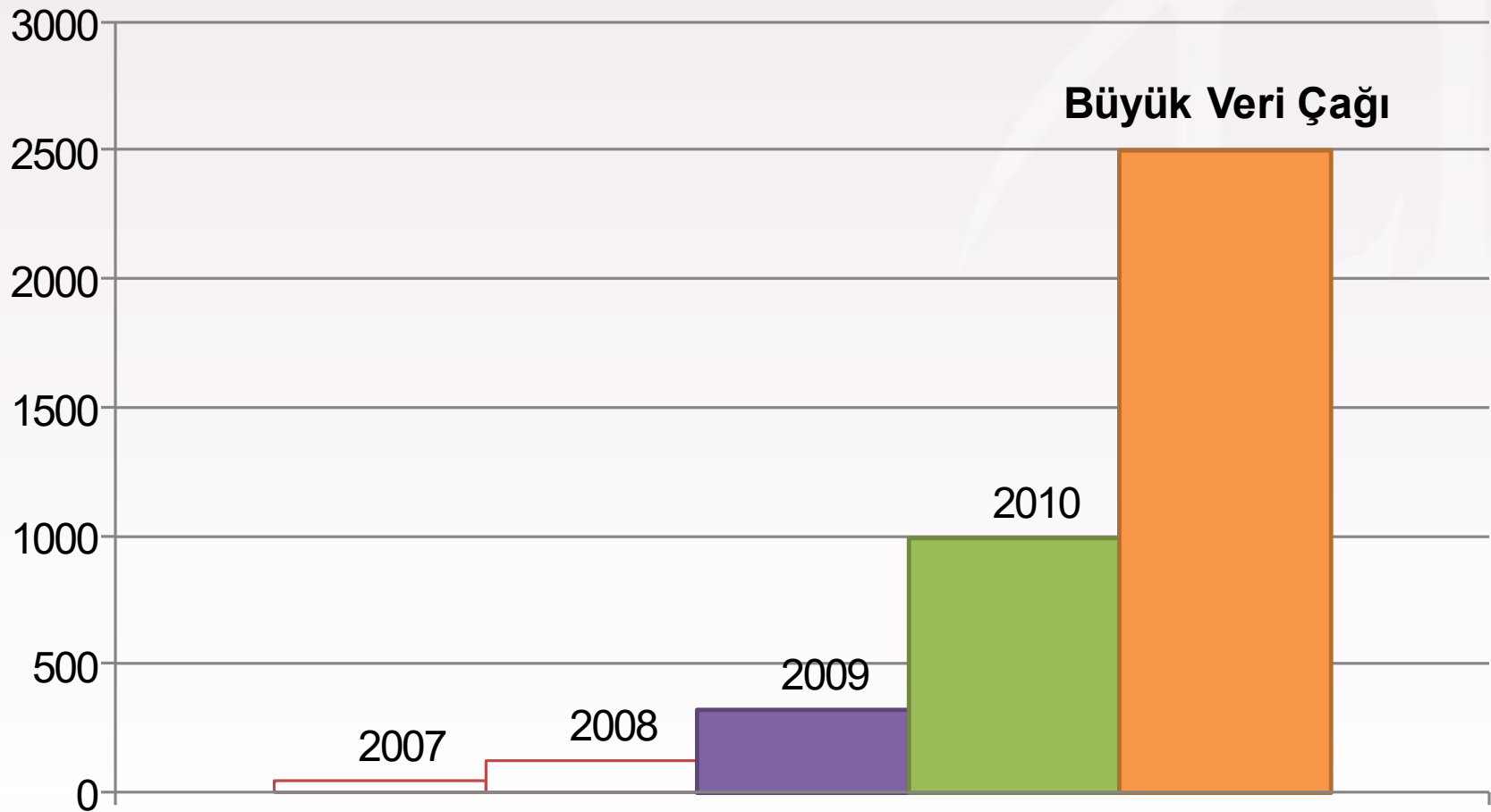


- Herhangi bir paylaşılan veri sisteminin aşağıdaki üç özelliğin tümünü aynı anda garanti etmesi imkansızdır.:
 - Consistency (Tutarlılık) - veriler yazıldıktan sonra, gelecekteki tüm okuma istekleri bu verileri içerecektir.
 - Availability (Kullanılabilirlik) – veritabanı her zaman kullanılabilir ve cevap verebilir.
 - Partition Tolerance (Bölme Toleransı) - veritabanının bir kısmı kullanılamıyorsa, diğer kısımlar etkilenmez.

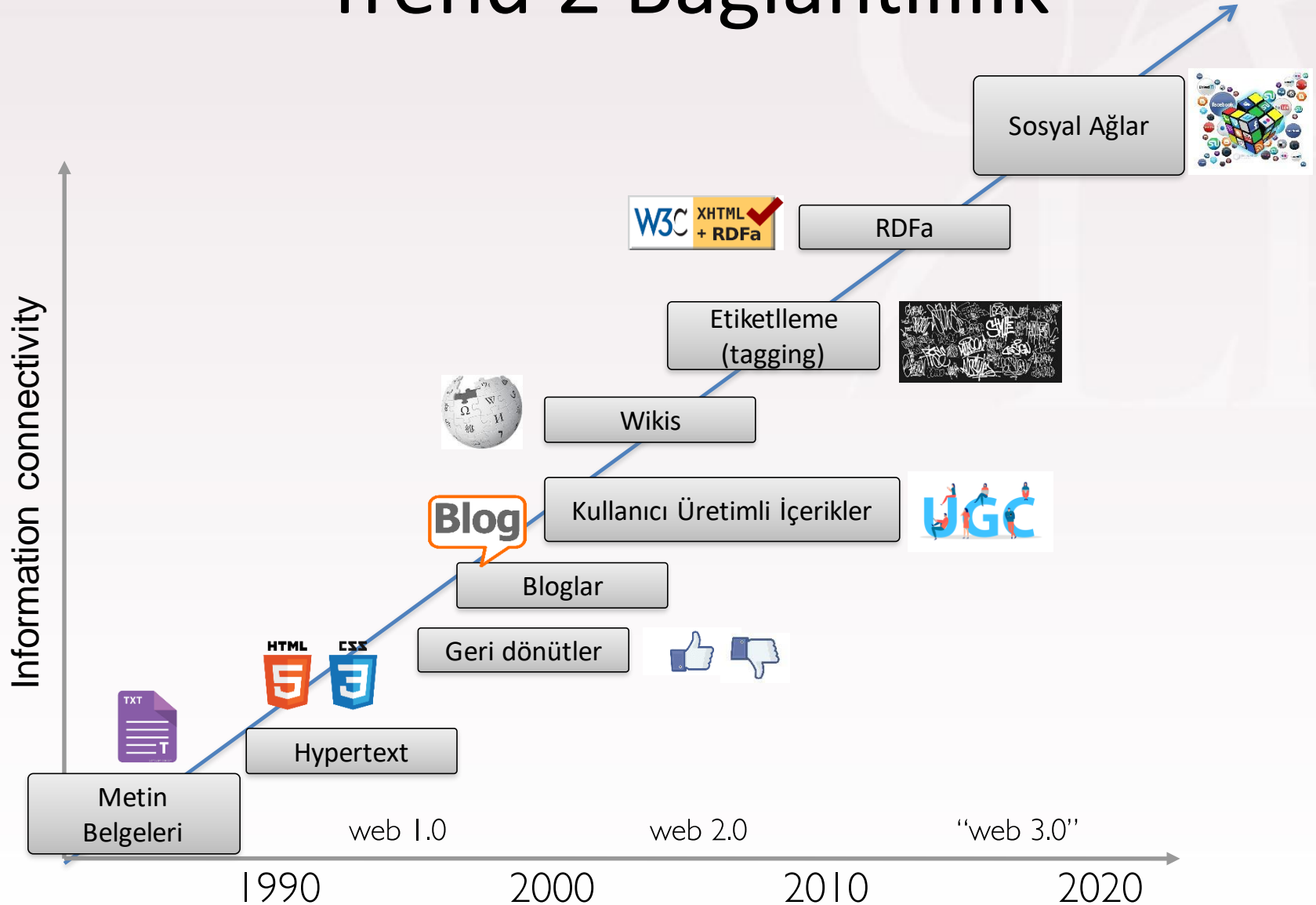


- RDBMS ACID garantisi için kullanılabilirliğe göre tutarlılığı tercih ederken, NoSQL veritabanları çoğunlukla tutarlılığa göre kullanılabilirliği tercih eden BASE felsefesine atıfta bulunarak tasarlanmış sistemlerdir.

Trend 1 Veri Boyutu



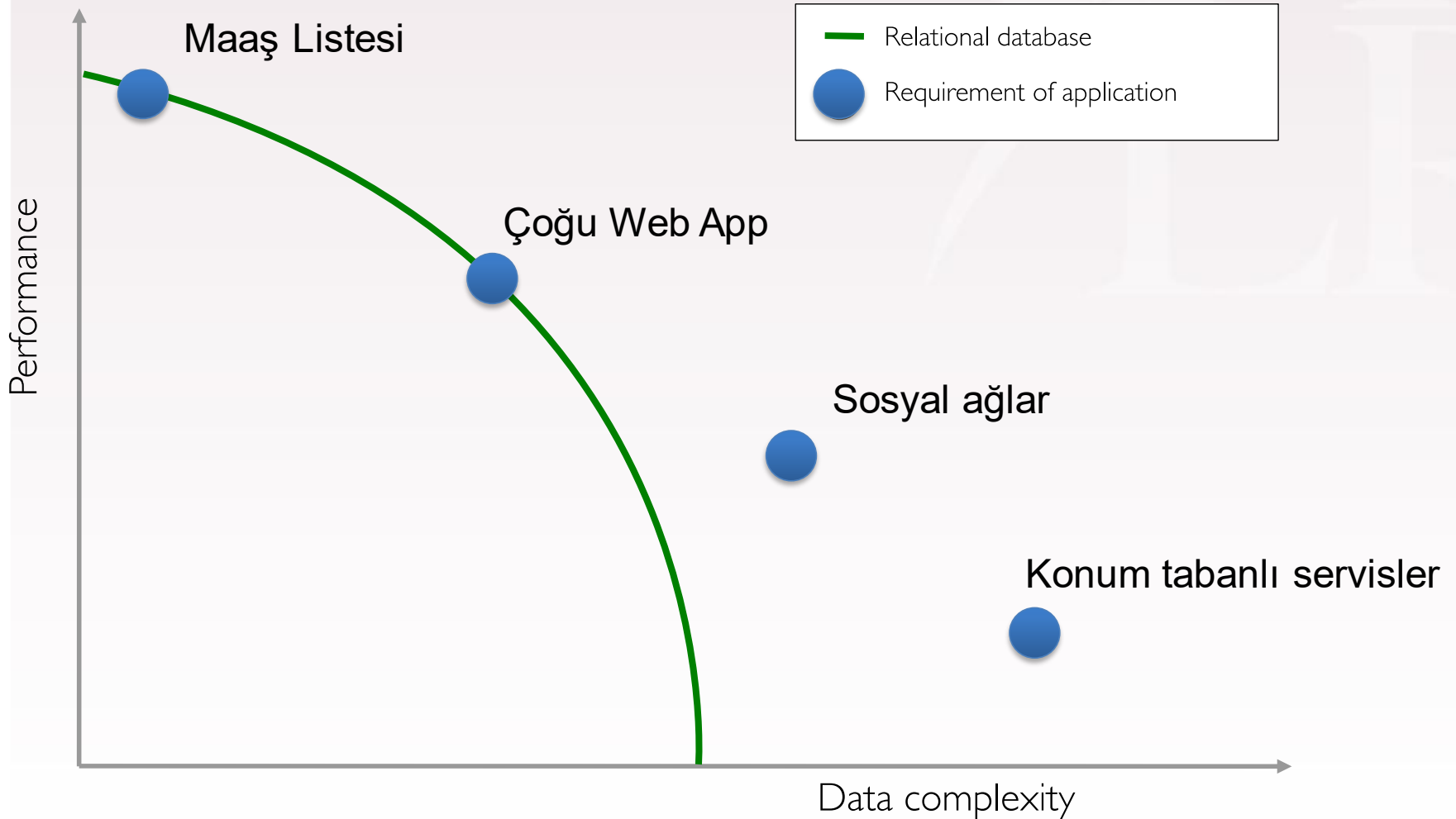
Trend 2 Bağlantılılık



Trend 3: Yarı-Yapılandırılmış Bilgi

- İçeriğin kişiselleştirilmesi
 - 1970'lerin maaş listeleri, tüm unsurlar tam olarak tek iş
 - 2000'li yılların maaş listeleri, birçok iş sütununa ihtiyacımız var!
- Her varlık için daha fazla veri depolamak
- İçerik üretiminin merkezden uzaklaştırılmasıyla hızlanan trend
 - Katılım çağı (“web 2.0”)

RDBMS Performansı



NoSQL'e Giriş

- RDBMS *bazı* gereksinimlere tam olarak uymuyor.
- Google ve Amazon, kendi benzersiz ihtiyaçlarını karşılamak için kendi ürünlerini (diğer adıyla BigTable ve S3 Storage) yapmaya karar verdiler.
 - Dağıtık
 - Ölçeklenebilir
 - Performans karakteristiği üzerinde kontrol
 - Yüksek erişebilirlik
 - Düşük gecikme
 - Ucuz



Google Bigtable

Şubat 2005



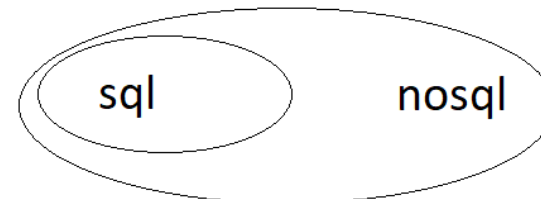
Mart 2006

No SQL

- SQL'e hayır!
- SQL'den bıktım!
- Basitçe SQL'e hayır anladınız mı?
- Sadece SQL değil?



Not
Only SQL

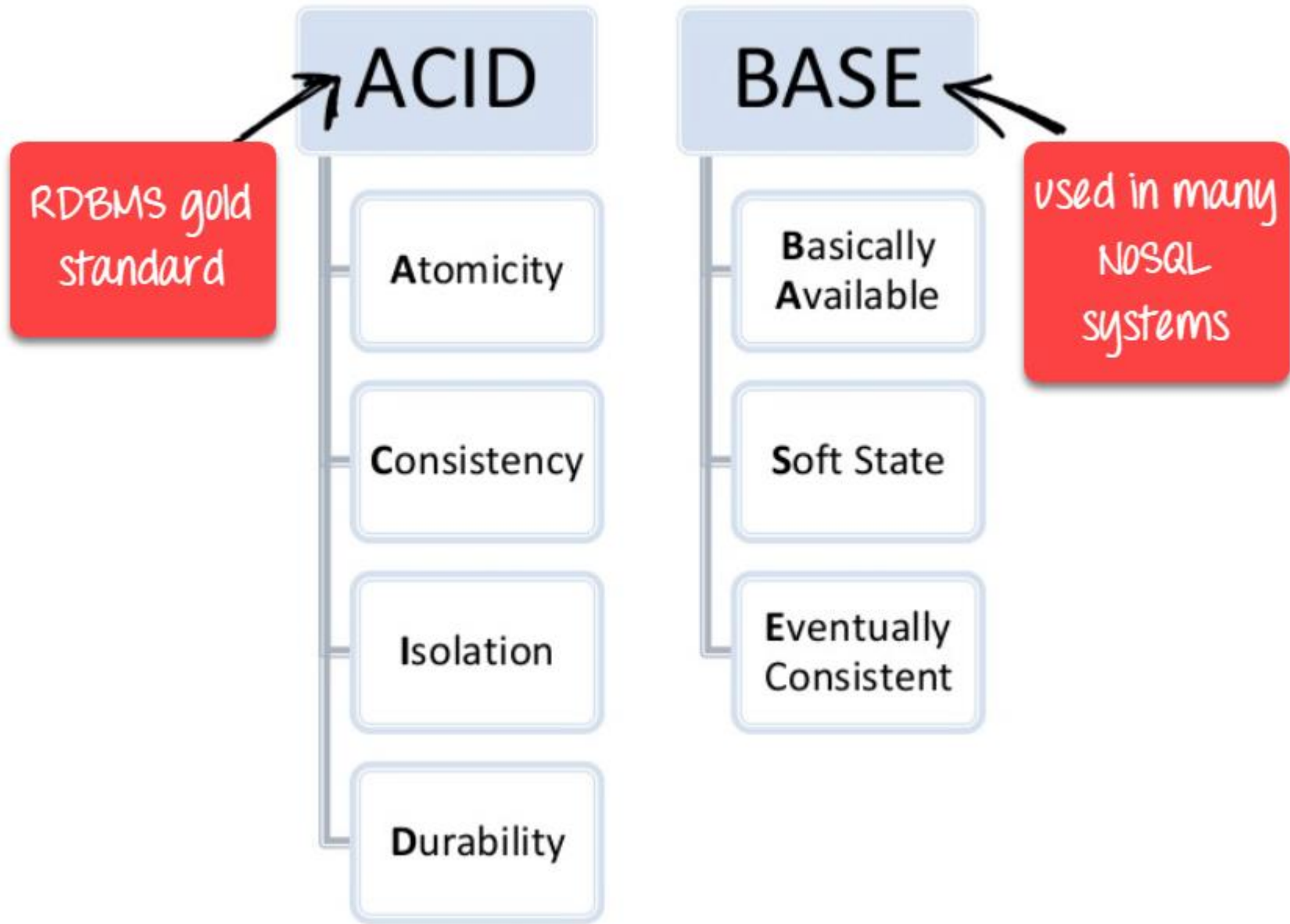


NoSQL Nedir ?

- Temel olarak büyük bir seri hale getirilmiş nesne deposu
 - (çoğunlukla) tanımlanan ID'ye göre nesneleri getir.
- Genel olarak, karmaşık sorguları desteklemez
- Yapılandırılmış (veya herhangi bir!) şeması yok
 - denormalizasyon önerir
- Dağıtık tasarlandı (bulut ölçekli)
- Bu nedenle, ACID gereksinimlerini zayıflatır
 - Herhangi bir veritabanı herhangi bir sorguyu cevaplayabilir
 - Herhangi bir yazma sorgusu, herhangi bir veritabanına karşı çalışabilir ve "sonunda" diğer dağıtılmış sunuculara yayılır.

En Popüler NoSQL Veritabanları

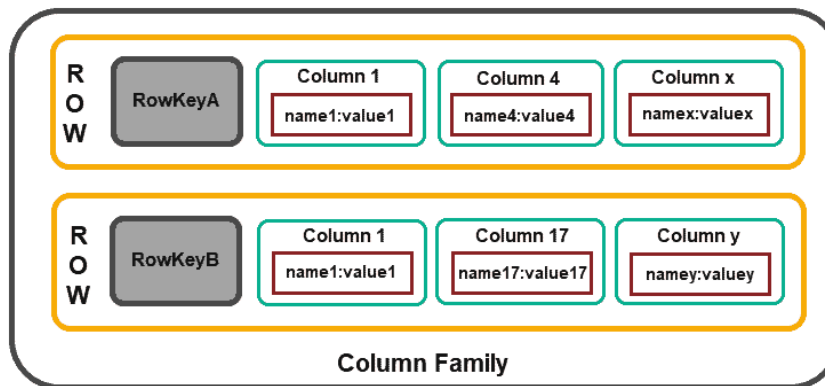




Farklı NoSQL Türleri

- **Sütun Deposu**

- İlişkisel veritabanlarından farklı olarak sütun veritabanları, verilerini satırlar yerine sütunlar halinde saklar.
- Veri analitiği için süper!
- Hadoop, Cassandra, Hypertable



- İlişkisel veritabanlarıyla karşılaşanlar, ilişkisel veritabanının her sütununun aynı sayıda satıra sahip olduğunu bilirler, ancak bazı alanların boş bir değere sahip olduğu veya boş görüldüğü görülür. Geniş sütunlu veritabanlarında, boş olmak yerine, bu satırlar belirli bir sütun için mevcut değildir.

Column-oriented vs. row-oriented databases

ID NUMBER	LAST NAME	FIRST NAME	BONUS
513001	Jones	Joanna	8000
502333	Smith	Jamie	4000
455332	Beck	Samuel	1000

Row-oriented database

513001	Jones	Joanna	8000
502333	Smith	Jamie	4000
455332	Beck	Samuel	1000

Column-oriented database

513001	502333	455332
Jones	Smith	Beck
Joanna	Jamie	Samuel
8000	4000	1000

Farklı NoSQL Türleri

Anahtar-Değer (Key-Value) Deposu

Veriye atıfta bulunan bir anahtar

MemcacheDB, Azure Table Storage, Redis

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Document / XML / Object Store

Key (and possibly other indexes) point at a serialized object

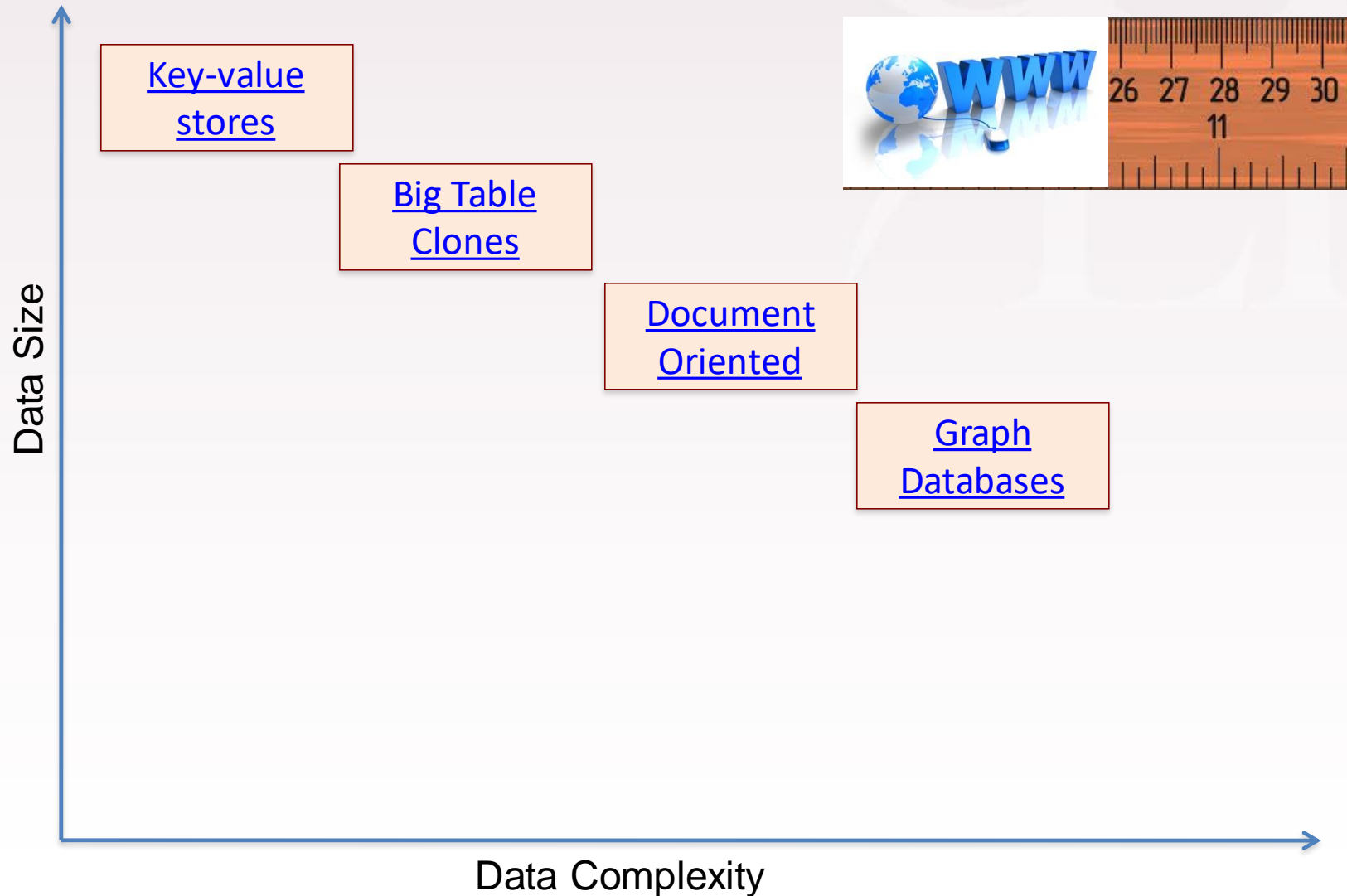
DB can operate against values in document

MongoDB, CouchDB, RavenDB

Graph Store

Nodes are stored independently, and the relationship between nodes (edges) are stored with data

NoSQL Data Models



Also Important – MapReduce

- One of the most common large queries (and hard to optimize) is Map Reduce
 - Map: Select a subset of data (into a key, value pair)
 - Reduce: Perform some sort of operation on that data
 - Used primarily for data analysis (reporting)
 - Example: Average revenue on all invoices in 2010
- In RDBMS, this is usually pretty expensive
 - Involves a full table scan with a select
 - If it's not a built-in aggregate function, involves custom code on database server (sproc/udf)
- In (some) NoSQL, it's automagically distributed
 - MapReduce functions are automatically distributed across all nodes to process, and the result is automatically gathered and returned as part of the vendor's framework
- In Google's environment, can run a MapReduce operation across all their nodes in about 30 minutes.

That being said...

NoSQL-type databases power:

- Google reader
- Google maps
- Blogger.com
- Youtube
- Gmail
- Amazon
- Sourceforge
- Github
- CollegeHumor
- Justin.tv
- Grooveshark
- BMW
- Cisco
- Honda
- Mozilla
- Adobe
- Ebay
- Facebook
- Hulu
- Last.Fm
- LinkedIn
- New York Times
- Twitter
- Yahoo!
- Disney
- Craigslist
- Foursquare
- Forbes
- Bit.ly
- Intuit
- Boeing
- US Army
- Seagate
- Hertz
- IBM
- Intel
- Oxford University Press
- United Airways
- University of Toronto
- XQ ☺

MongoDB Session Starts Now



MongoDB is a _____ database

- Document
- Open source
- High performance
- Horizontally scalable
- Full featured

Document Database

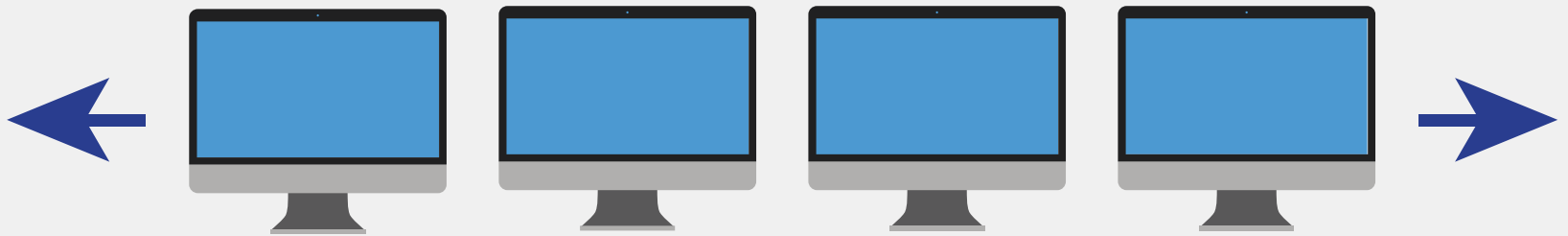
- Not for .PDF & .DOC files
- A document is essentially an associative array
- Document == JSON object
- Document == PHP Array
- Document == Python Dictionary
- Document == Ruby Hash
- etc

Open Source

- MongoDB is an open source project
- On GitHub
- Licensed under the AGPL
- Commercial licenses available
- Started & sponsored by 10gen
- Contributions welcome

High Performance

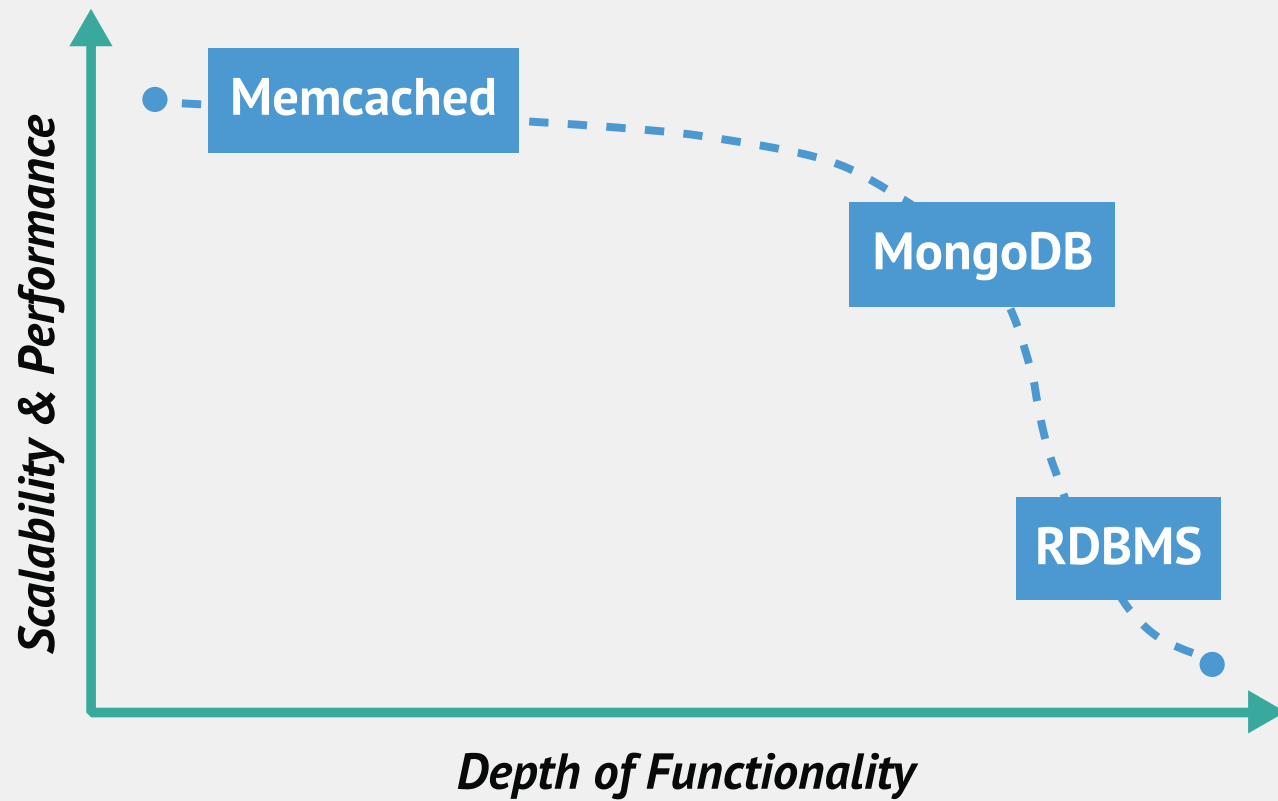
- Written in C++
- Extensive use of memory-mapped files i.e. read-through write-through memory caching.
- Runs nearly everywhere
- Data serialized as BSON (fast parsing)
- Full support for primary & secondary indexes
- Document model = less work



Horizontally Scalable

Full Featured

- Ad Hoc queries
- Real time aggregation
- Rich query capabilities
- Traditionally consistent
- Geospatial features
- Support for most programming languages
- Flexible schema



Database Landscape



{name: "mongo", type: "DB"}

[Forums](#) | [Blog](#) | [Twitter](#) | [Facebook](#) | [Events](#) | [International](#)



MONGO DOCS

[MongoDB Documentation »](#)



TRY IT OUT

[Try The Online Shell »](#)



DOWNLOADS

[Download MongoDB »](#)



DRIVERS

[Get The Latest Drivers »](#)

MongoDB Downloads

This table lists MongoDB distributions by platform and version. We recommend using these binary distributions, but there are also [packages](#) available for various package managers.

	OS X 32-bit note	OS X 64-bit	Linux 32-bit note	Linux 64-bit	Windows 32-bit note	Windows 64-bit	Solaris 64-bit	Source
Production Release (Recommended)								
2.2.0 8/29/2012 Changelog Release Notes	download	download	download *legacy-static	download *legacy-static	download	download *2008+	download	tgz zip

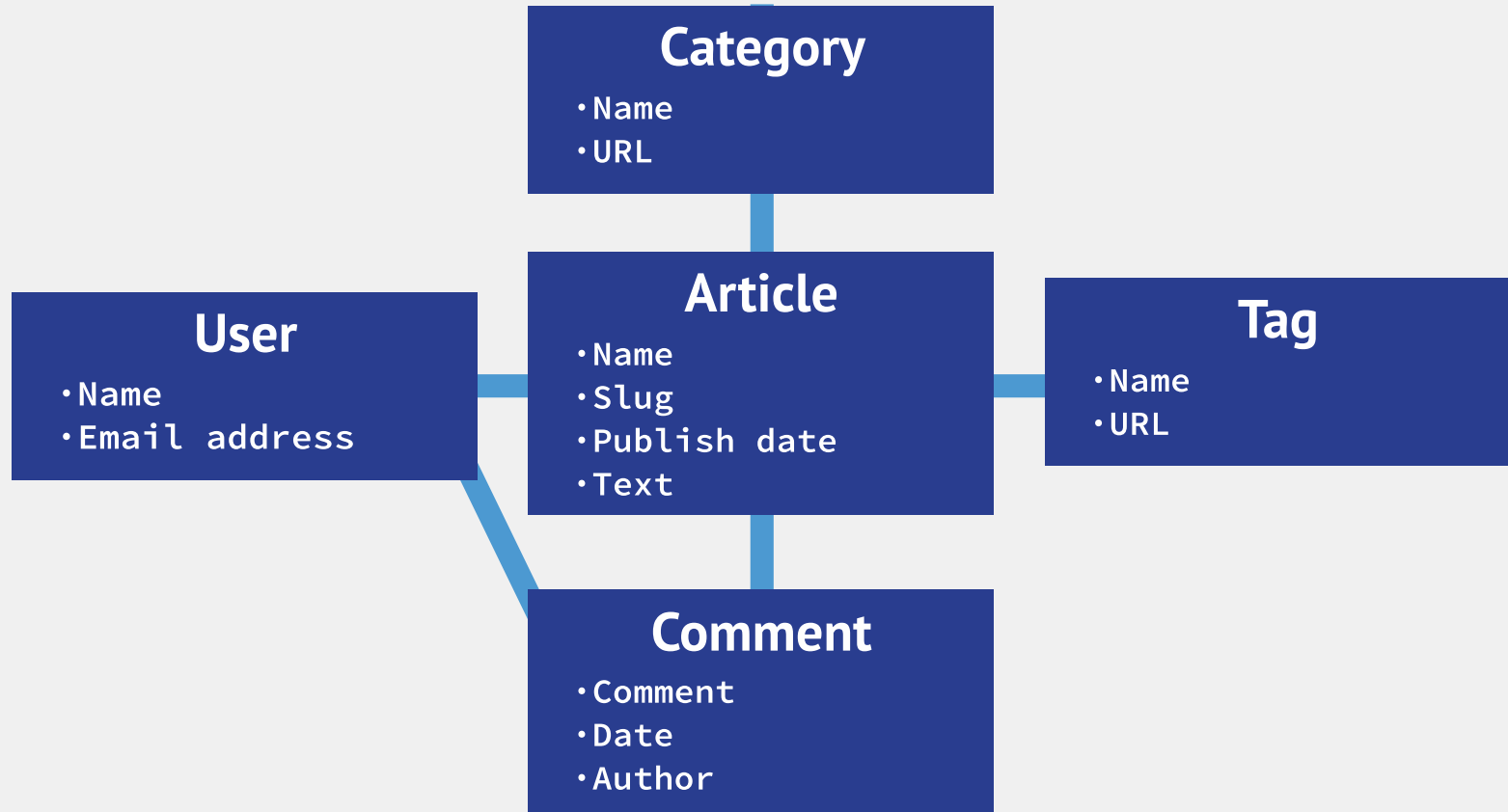
<http://www.mongodb.org/downloads>

```
~ | ⇒ mongo
MongoDB shell version: 2.2.0
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
> db.test.insert({text: 'Welcome to MongoDB!'})
> db.test.find().pretty()
{
    "_id" : ObjectId("507f07fbd94ccab2da652596"),
    "text" : "Welcome to MongoDB!"
}
> |
```

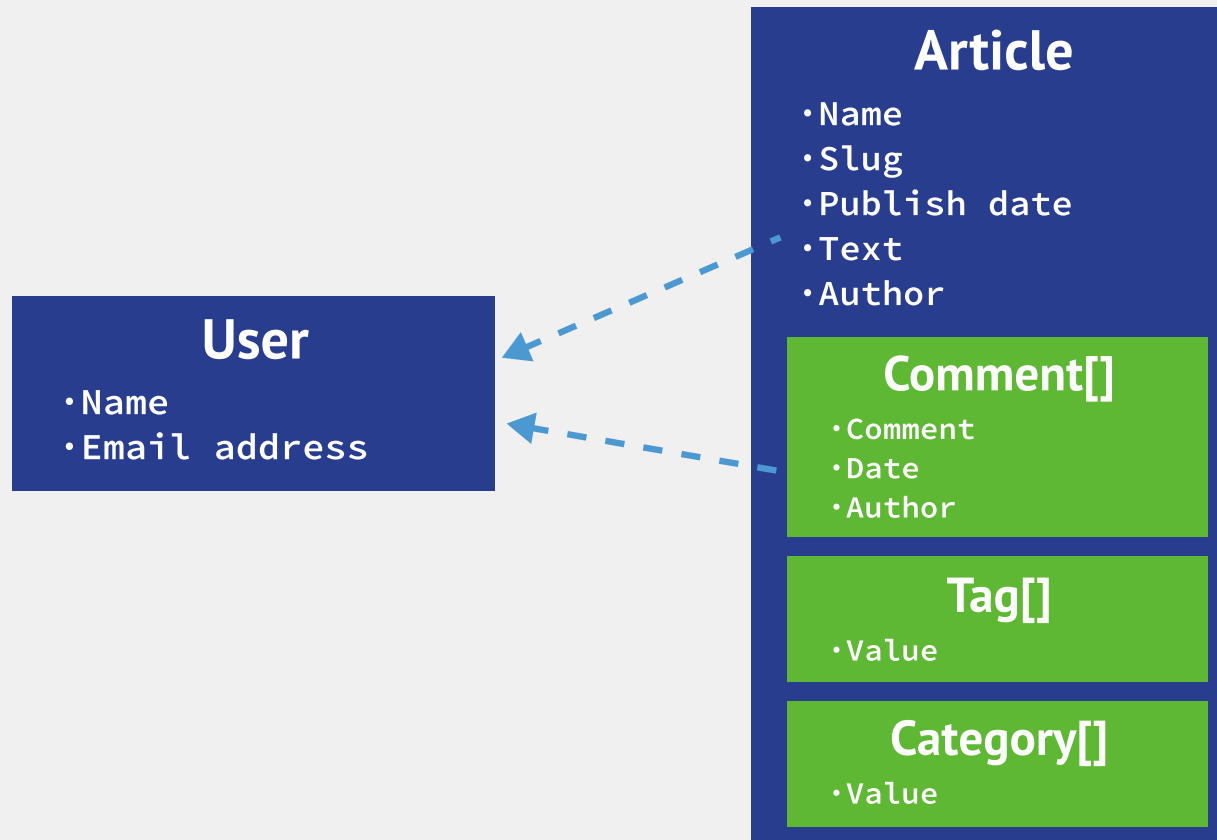
Mongo Shell

RDBMS		MongoDB
Table, View	→	Collection
Row	→	Document
Index	→	Index
Join	→	Embedded Document
Foreign Key	→	Reference
Partition	→	Shard

Terminology



Typical (relational) ERD

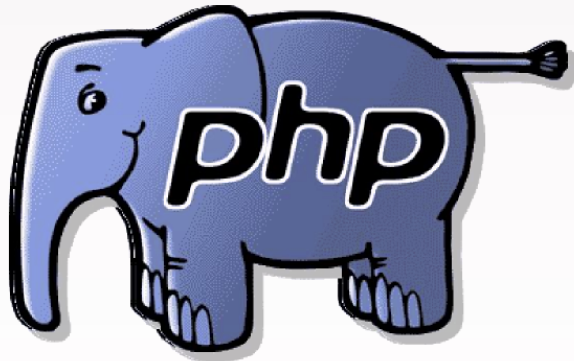


MongoDB ERD

MongoDB has native bindings for over 12 languages



C++



C

nodeJS

MongoDB drivers

- Official Support for 12 languages
- Community drivers for tons more
- Drivers connect to mongo servers
- Drivers translate BSON into native types
- mongo shell is not a driver, but works like one in some ways
- Installed using typical means (npm, pecl, gem, pip)

Neo4j Session Starts Now



Why Graph Databases?

- Schema Less and Efficient storage of Semi Structured Information
- No O/R mismatch – very natural to map a graph to an Object Oriented language like Ruby.
- Express Queries as Traversals. Fast deep traversal instead of slow SQL queries that span many table joins.
- Very natural to express graph related problem with traversals (recommendation engine, find shortest path etc..)
- Seamless integration with various existing programming languages.
- ACID Transaction with rollbacks support.
- Whiteboard friendly – you use the language of node, properties and relationship to describe your domain (instead of e.g. UML) and there is no need to have a complicated O/R mapping tool to implement it in your database. You can say that Neo4j is “Whiteboard friendly” ! (<http://video.neo4j.org/JHU6F/live-graph-session-how-allison-knows-james/>)

Social Network “path exists” Performance

- Experiment:
 - ~1k persons
 - Average 50 friends per person
 - `pathExists(a, b)`
limited to depth 4

	# persons	query time
Relational database	1000	2000ms

Social Network “path exists” Performance

- Experiment:
 - ~1k persons
 - Average 50 friends per person
 - `pathExists(a, b)`
limited to depth 4

	# persons	query time
Relational database	1000	2000ms
Neo4j	1000	2ms

Social Network “path exists” Performance

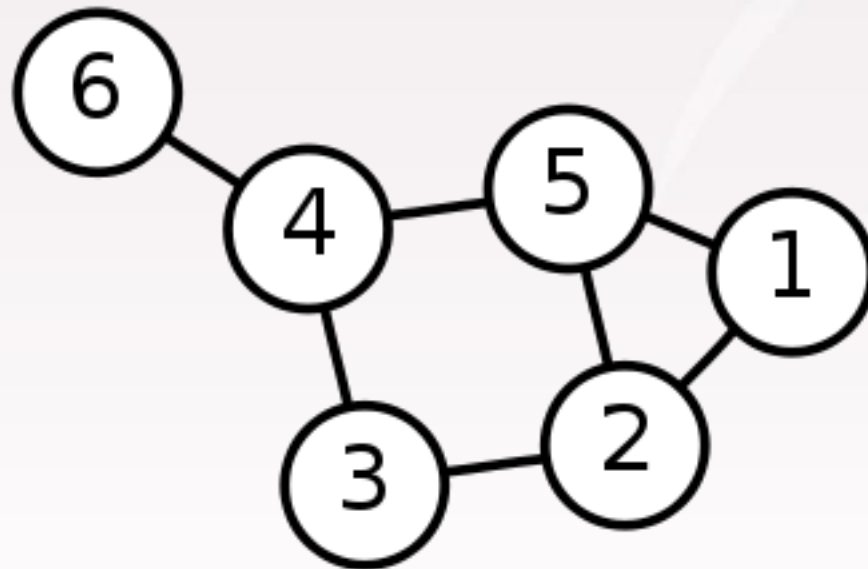
- Experiment:
 - ~1k persons
 - Average 50 friends per person
 - `pathExists(a, b)` limited to depth 4

	# persons	query time
Relational database	1000	2000ms
Neo4j	1000	2ms
Neo4j	1000000	2ms

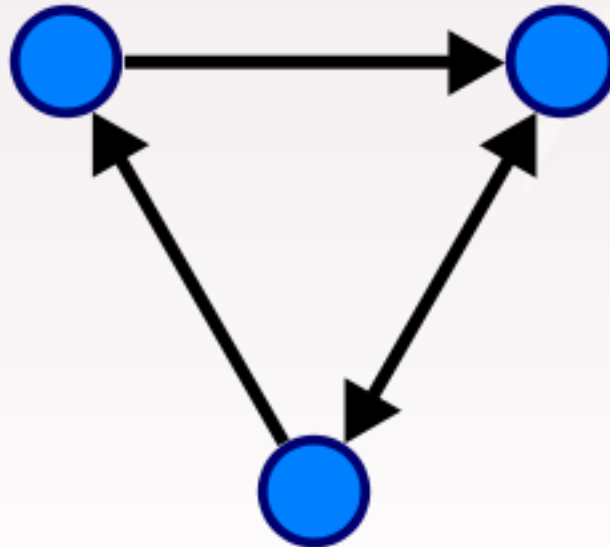
What are graphs good for?

- Recommendations
- Business intelligence
- Social computing
- Geospatial
- Systems management
- Web of things
- Genealogy
- Time series data
- Product catalogue
- Web analytics
- Scientific computing (especially bioinformatics)
- Indexing your *slow* RDBMS
- And much more!

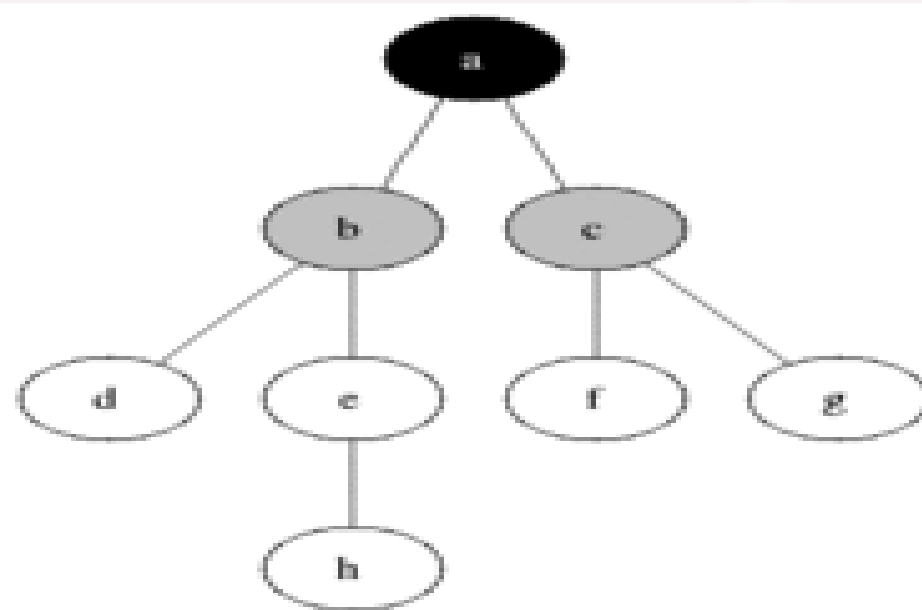
Graphs



Directed Graphs



Breadth First Search



Depth First Search

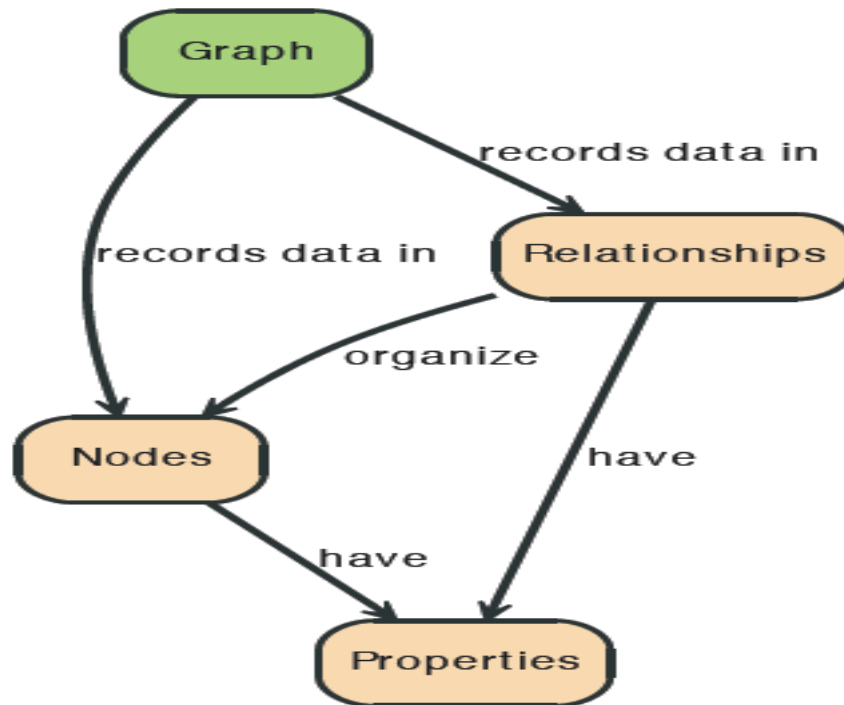
????????????????

Graph Databases

- A graph database stores data in a graph, the most generic of data structures, capable of elegantly representing any kind of data in a highly accessible way.

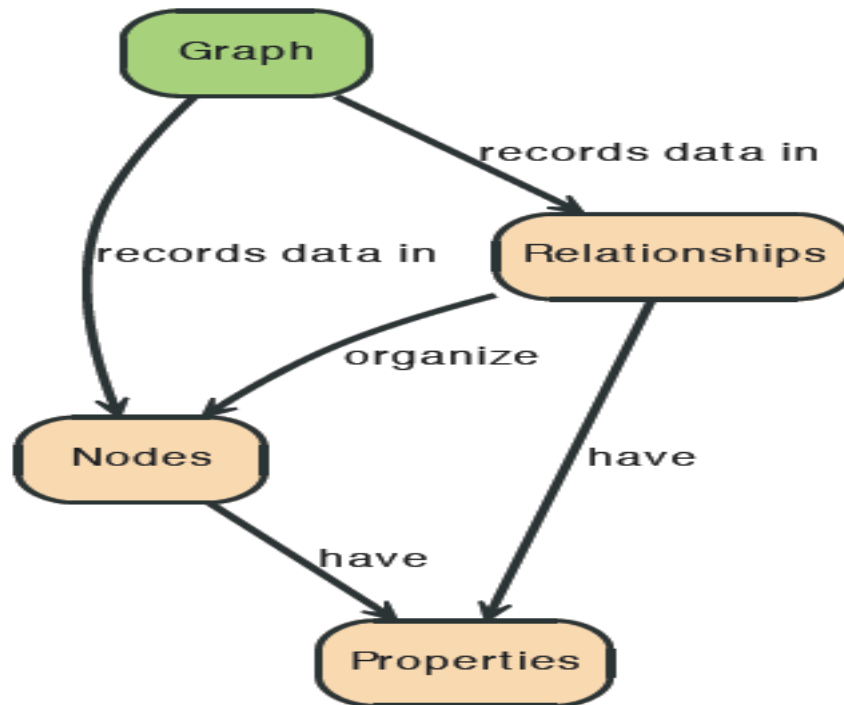
Graphs

- “A Graph —records data in→ Nodes —which have→ Properties”



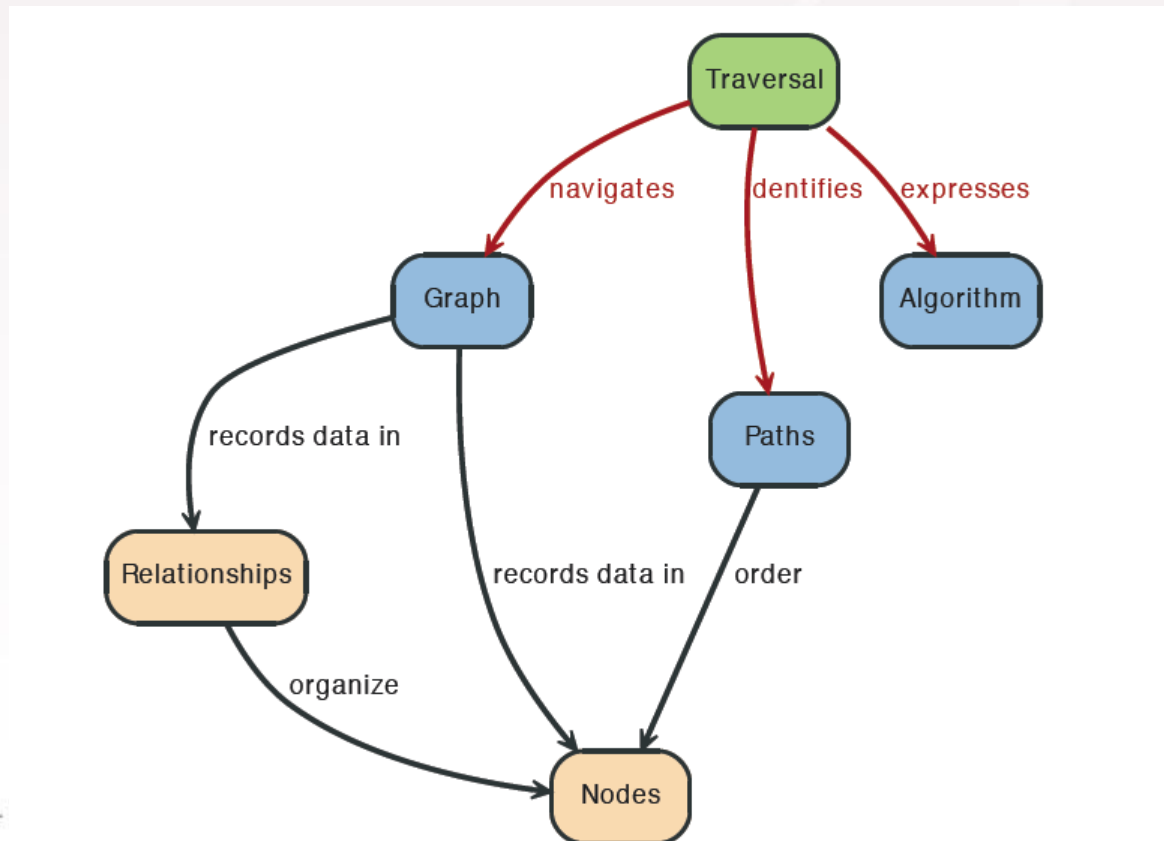
Graphs

- “Nodes —are organized by→ Relationships — which also have→ Properties”



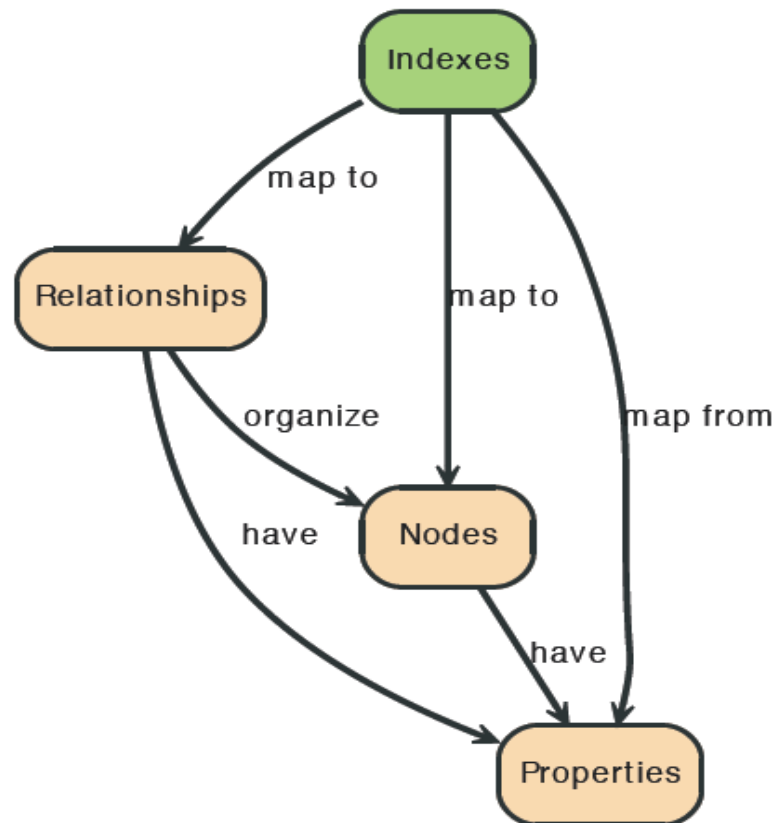
Query a graph with Traversal

- “A Traversal —navigates→ a Graph; it —identifies→ Paths —which order→ Nodes”



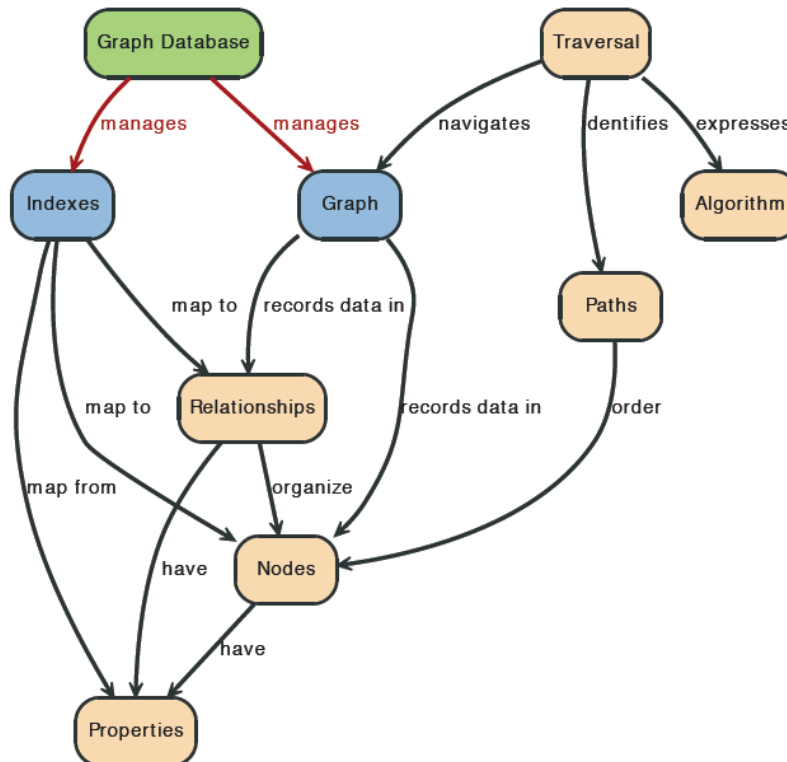
Indexes

- “An Index —maps from→ Properties —to either→ Nodes or Relationships”



Neo4j is a Graph Database

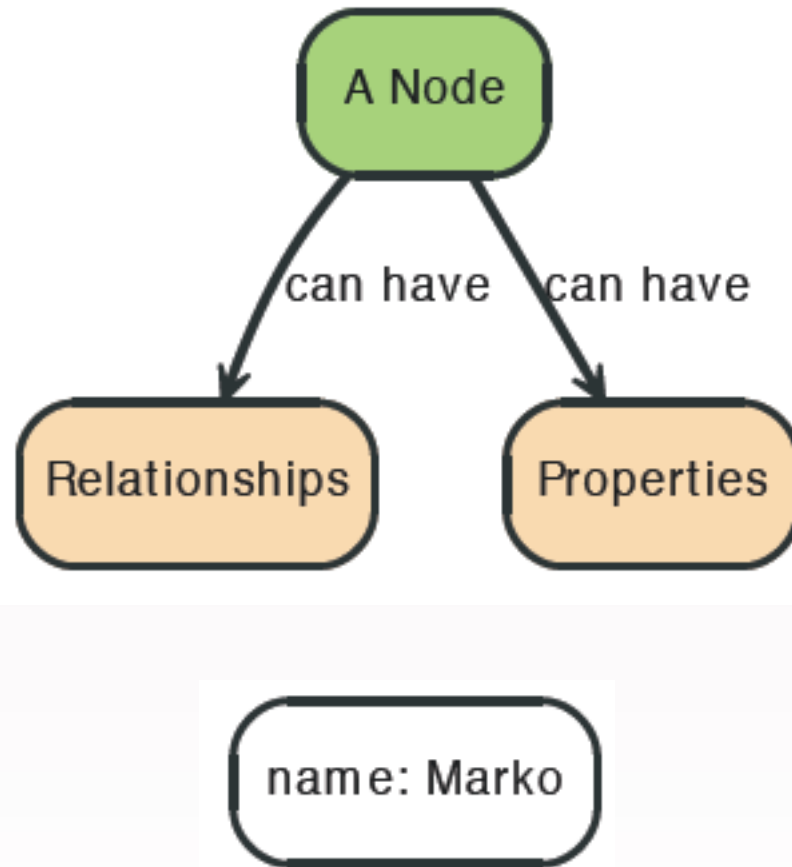
- “A Graph Database —manages a \rightarrow Graph and —also manages related \rightarrow Indexes”



Neo4j – Hey! This is why I am a Graph Database.

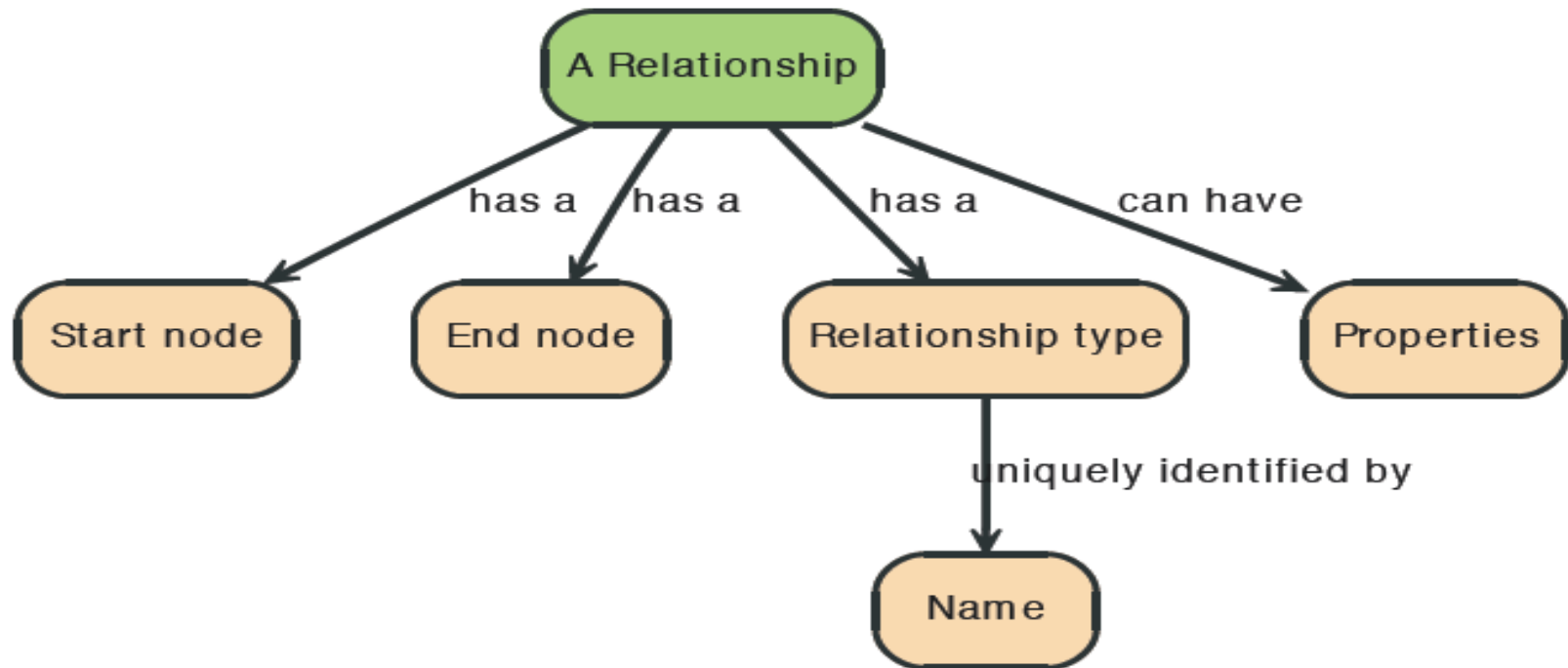
- The fundamental units that form a graph are nodes and relationships.
- In Neo4j, both nodes and relationships can contain properties.
- Nodes are often used to represent entities, but depending on the domain relationships may be used for that purpose as well.

Node in Neo4j

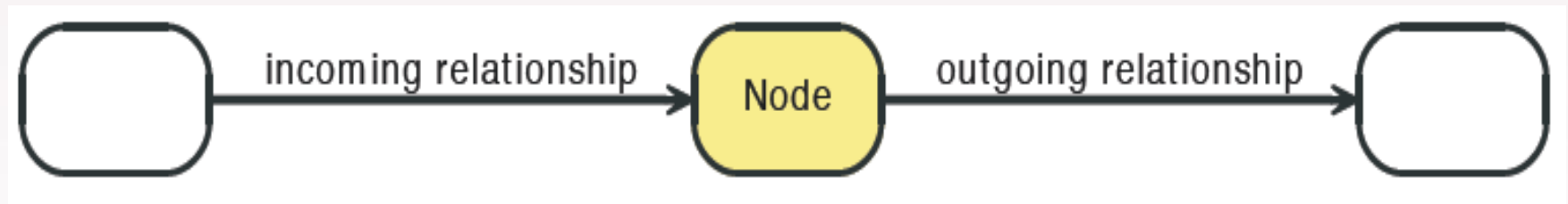
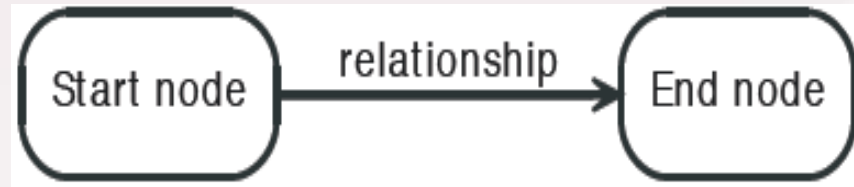


Relationships in Neo4j

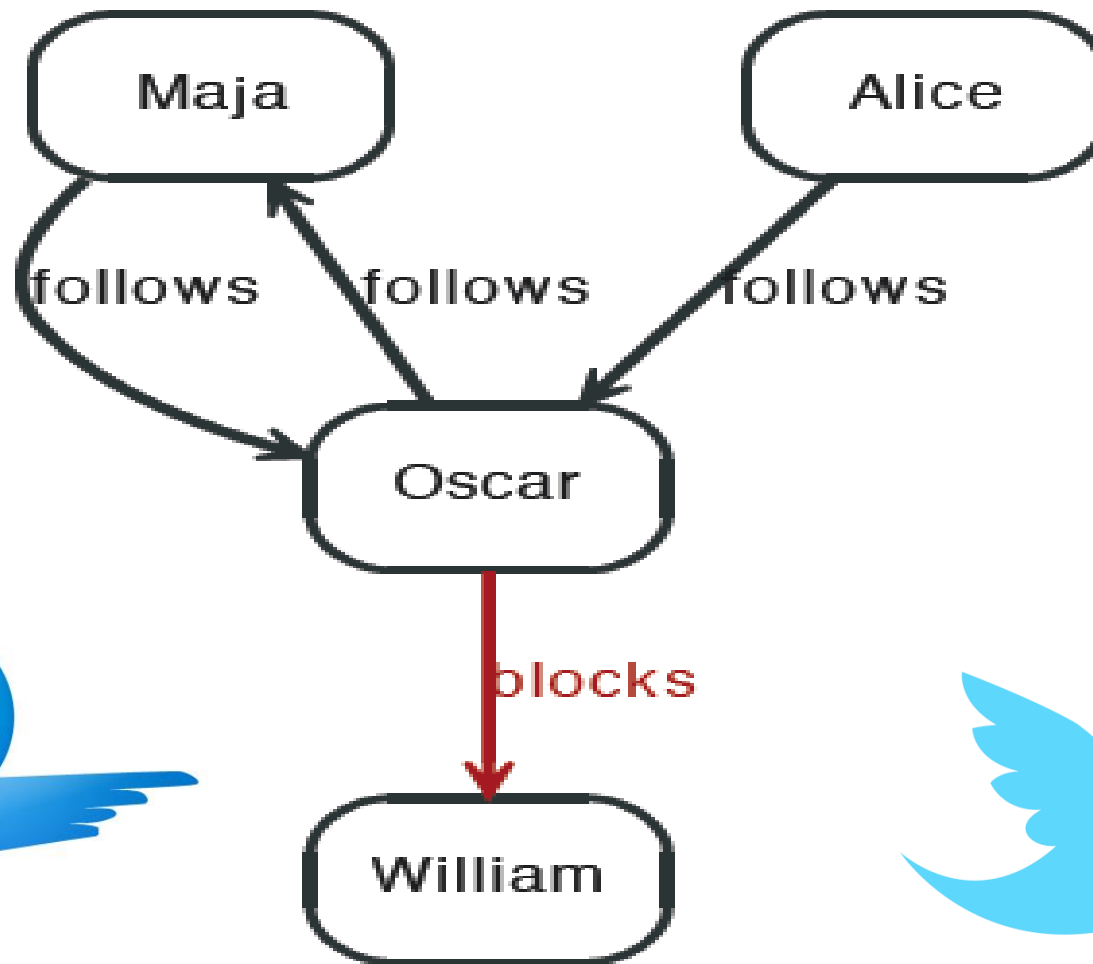
- Relationships between nodes are a key part of Neo4j.



Relationships in Neo4j



Twitter and relationships

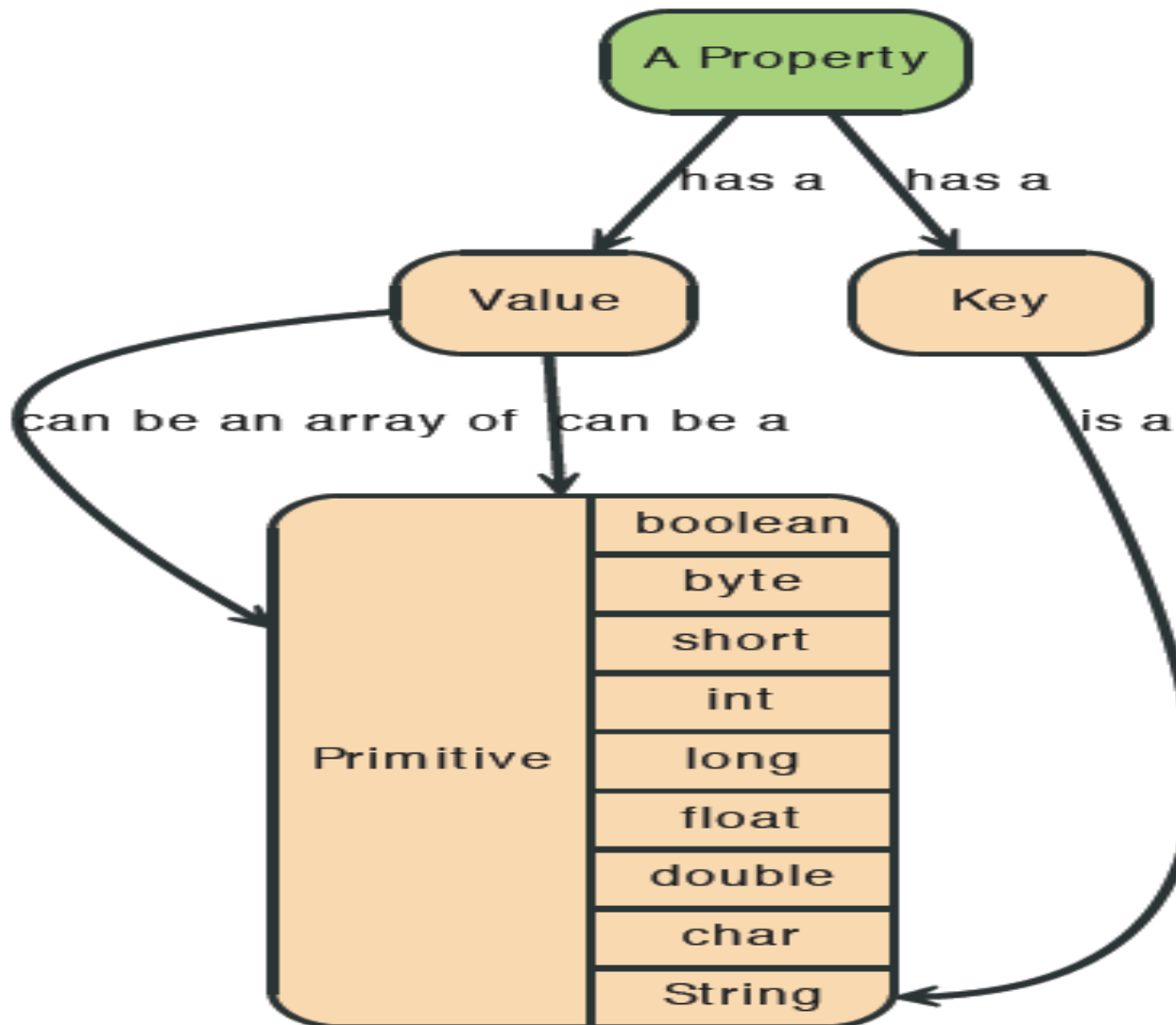


Properties

- Both nodes and relationships can have properties.
- Properties are key-value pairs where the key is a string.
- Property values can be either a primitive or an array of one primitive type.

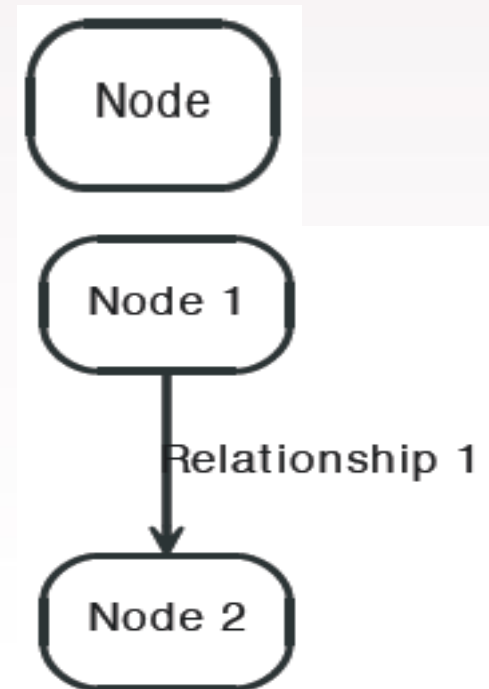
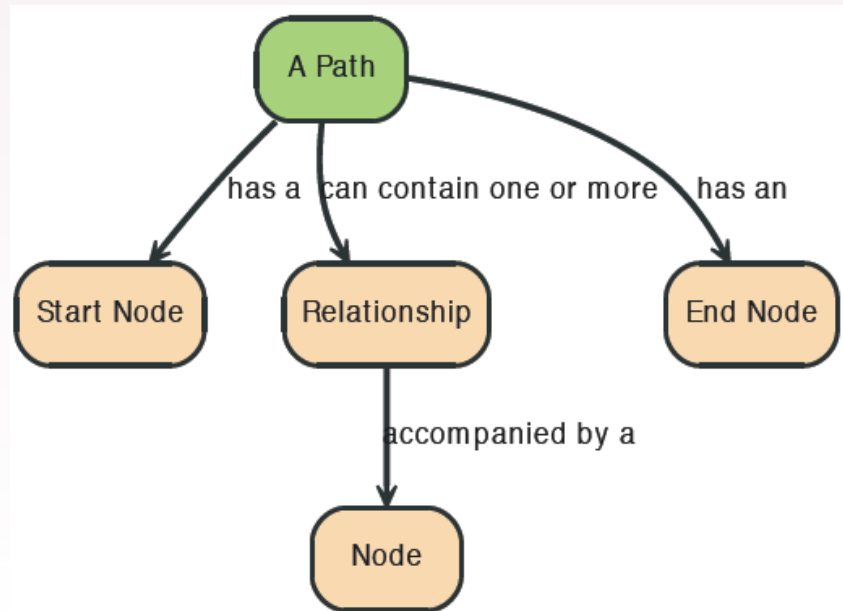
For example String, int and int[] values are valid for properties.

Properties



Paths in Neo4j

- A path is one or more nodes with connecting relationships, typically retrieved as a query or traversal result.



Traversals in Neo4j

- Traversing a graph means visiting its nodes, following relationships according to some rules.
- In most cases only a subgraph is visited, as you already know where in the graph the interesting nodes and relationships are found.
- Traversal API
- Depth first and Breadth first.

Thanks!

Resources & References

- Neo4j website : <http://neo4j.org/>
- Neo4j learning resources:
<http://neo4j.org/resources/>
- Videos about Neo4j: <http://video.neo4j.org/>
- Neo4j tutorial:
<http://docs.neo4j.org/chunked/snapshot/tutorials.html>
- Neo4j Java API documentation:
<http://api.neo4j.org/current/>

Resources and References

- MongoDB manual:
<http://docs.mongodb.org/manual/>
- Big Data:
<http://www.youtube.com/watch?v=eEpxN0htRKI>
- Hadoop:
<http://www.youtube.com/watch?v=xJHv5t8jcM8>