Unstructured Data Analysis and NoSQL

MSA 8040: Data Management for Analytics

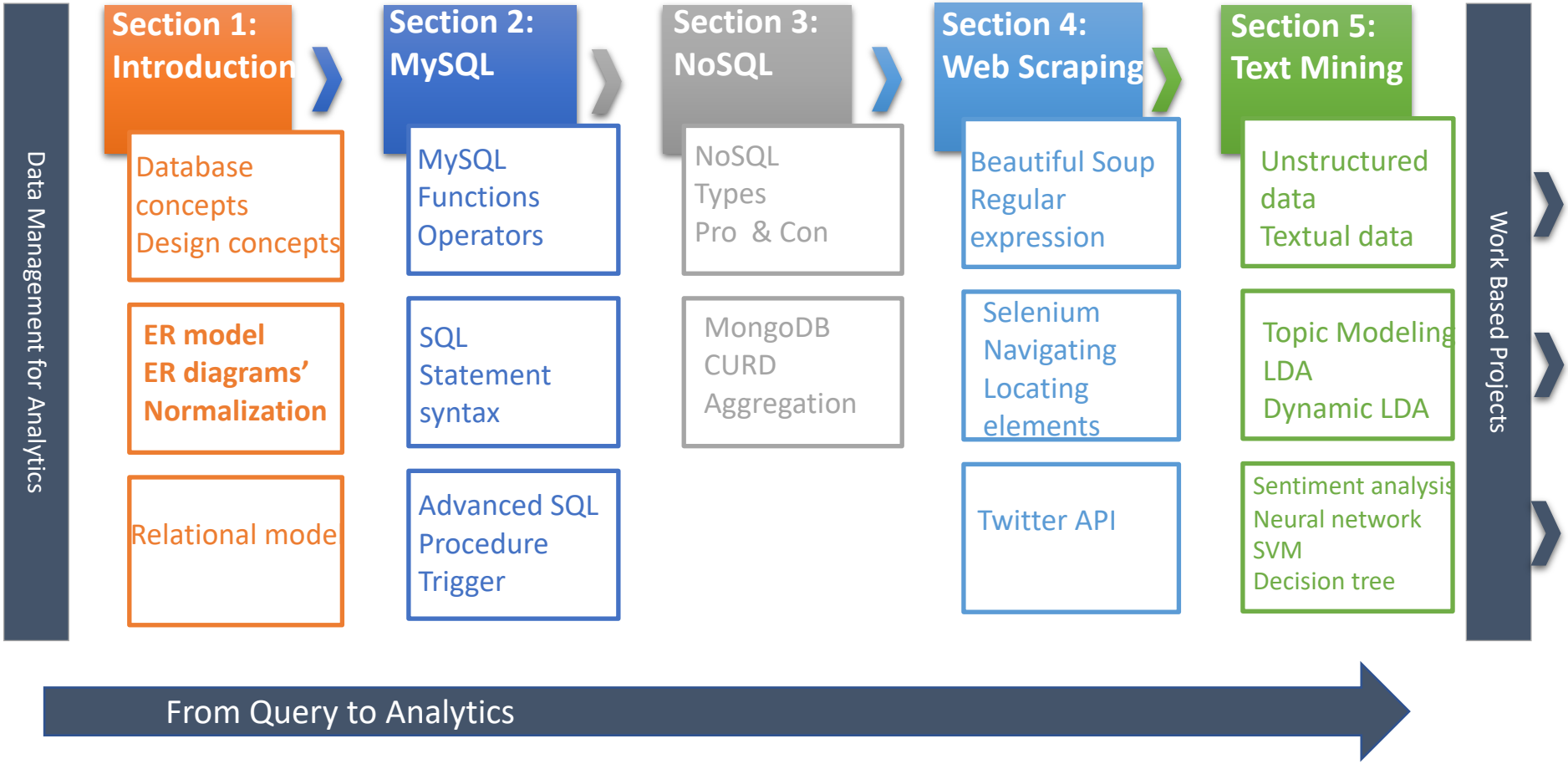
Houping Xiao

hxiao@gsu.edu

Georgia State University | J. Mack Robinson College of Business

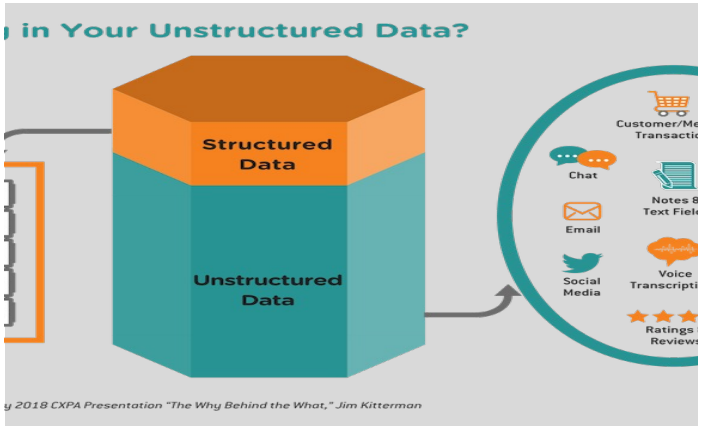
Robinson

Overview


Data Management for Analytics
Section 1: Introduction
Database concepts
Design concepts
ER model
ER diagrams'
Normalization
Relational mode
Section 2: MySQL
MySQL
Functions
Operators
SQL
Statement
syntax
Advanced SQL
Procedure
Trigger
Section 3: NoSQL
NoSQL
Types
Pro & Con
MongoDB
CURD
Aggregation
Section 4: Web Scraping
Beautiful Soup
Regular expression
Selenium
Navigating
Locating elements
Twitter API
Section 5: Text Mining
Unstructured data
Textual data
Topic Modeling
LDA
Dynamic LDA
Sentiment analysis
Neural network
SVM
Decision tree
Work Based Projects
From Query to Analytics

Learning objectives

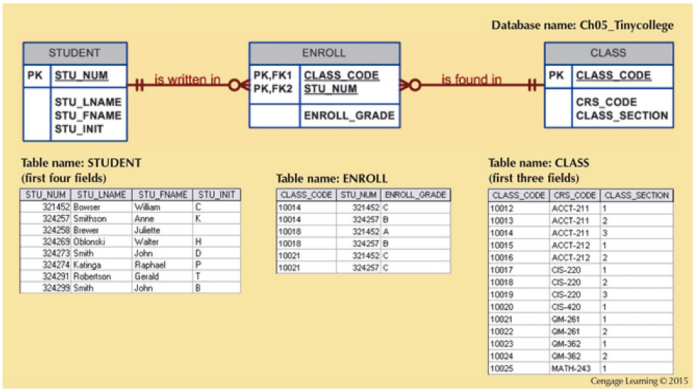
- Unstructured data
  - What is unstructured data?
  - How can we use unstructured data?
- NoSQL
  - What is NoSQL?
  - Types of NoSQL databases



What is structured data?

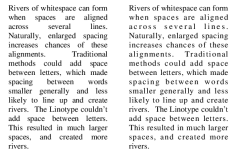
- The result of formatting disorganized data in order to facilitate storage, use and generation of information
- Well-defined entities, relationships between entities
- Clearly defined data types
  - Easily searchable

RDMBS



From structured to unstructured

- EE –"Everything Else”
- Data that exists in its original (raw) state—that is, in the format in which it was collected
- Entities and relationships are ambiguous
- Unstructured data is not as easily searchable
- Stored whin a non-relational database like NoSQL



Text

Images

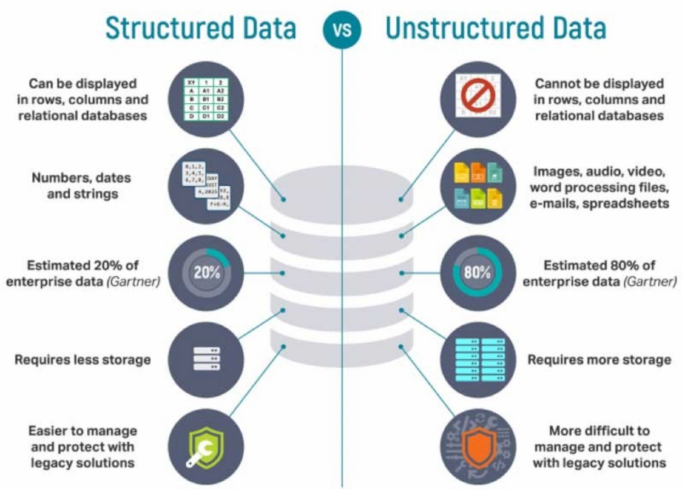
Videos

Audio

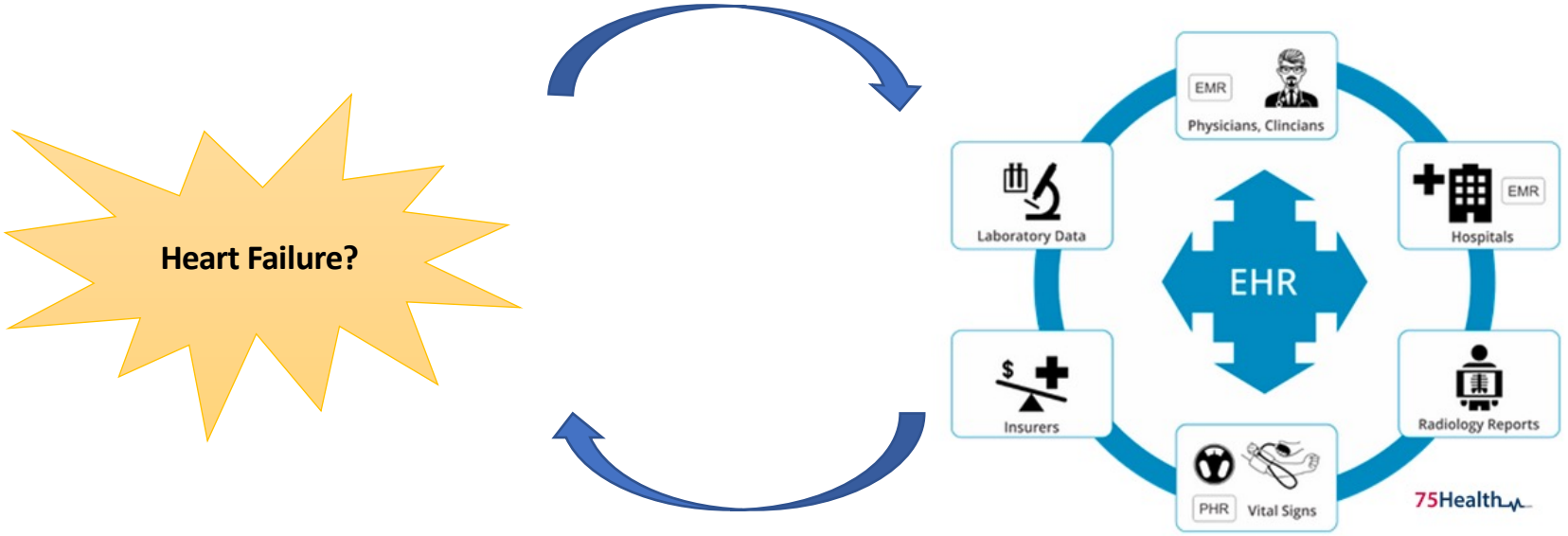
Heterogeneous data usually used together for making decisions!

Difference between structured and unstructured data

	Structured Data	Unstructured Data
characteristics	❑ Pre-defined data models ❑ Usually, number, simple characters only ❑ Easily searchable	❑ No pre-defined data models ❑ Could be text, images, video, voice or other formats ❑ Difficult to search
Resides in	RDMBS (SQL)	NoSQL
Generated by	Humans or machines	Humans or machines
Applications	❑ Airline reservation systems ❑ Transactions ❑ Inventory control	❑ Speech translations ❑ Face recognition ❑ Word processing
Examples	❑ Dates ❑ Phone numbers ❑ SSN ❑ Credit card numbers ❑ Customer names ❑ Address	❑ Text files ❑ Email message ❑ Videos ❑ Audios ❑ Images ❑ Surveillance imagery

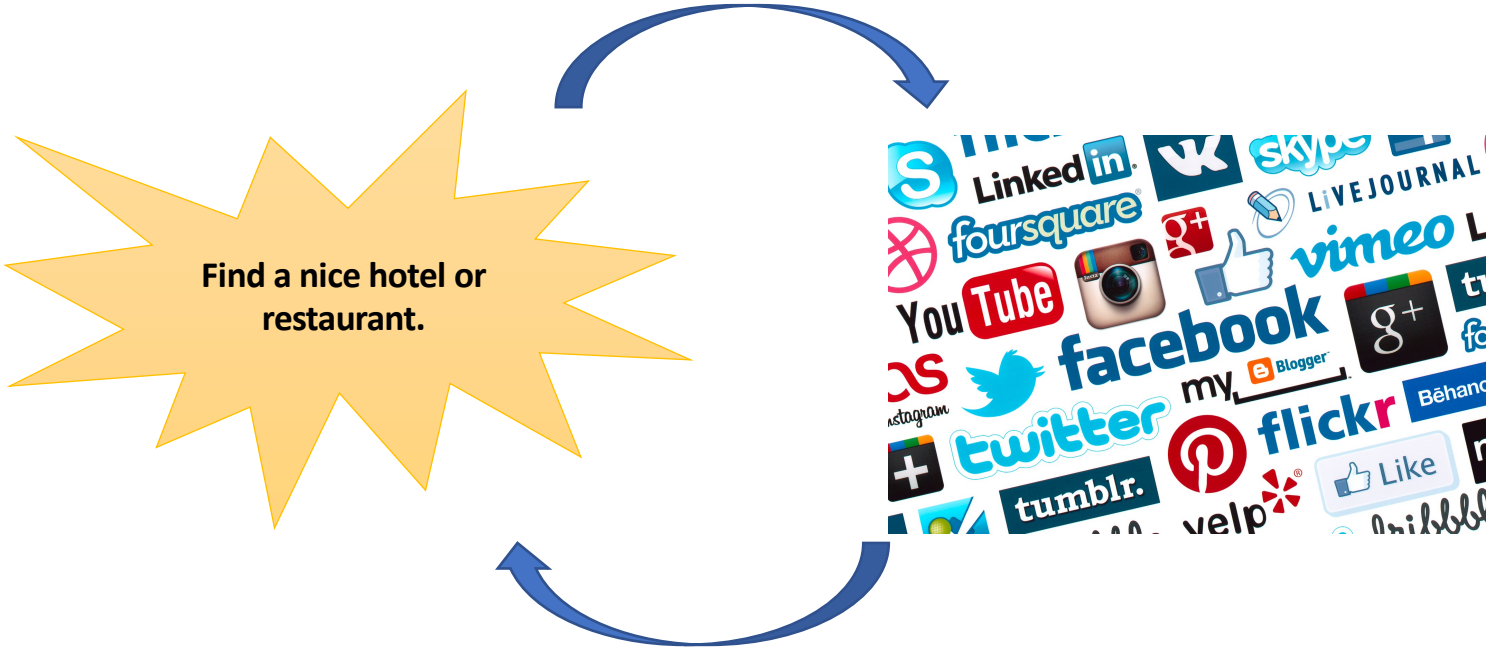


An example of healthcare application



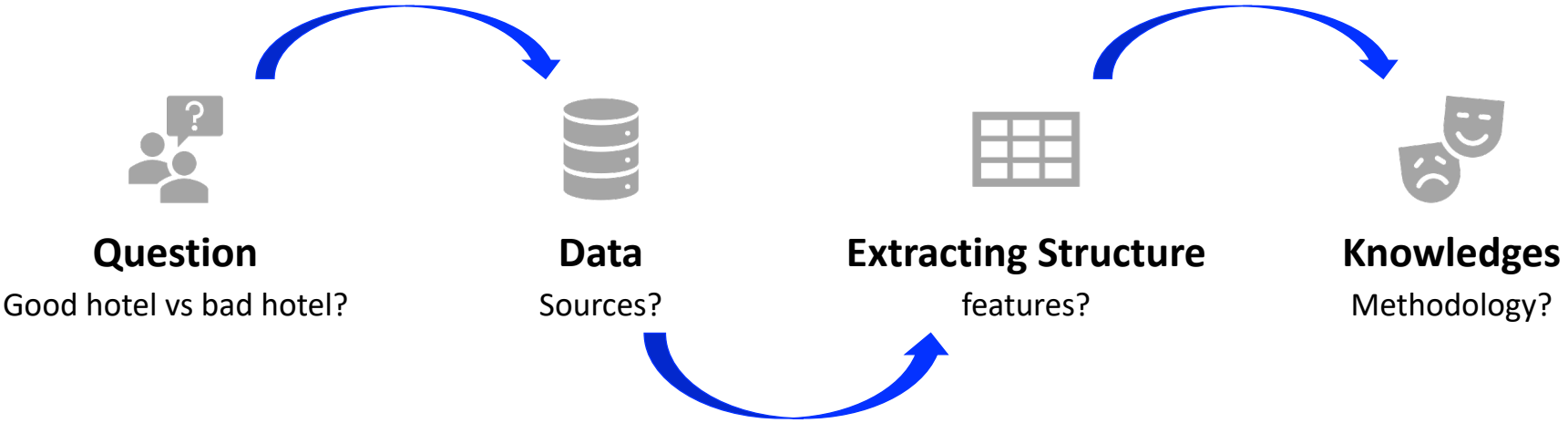
- ❑Chart measures, e.g. weight, blood pressure, etc.
- ❑Lab tests, e.g. blood tests, screening tests, etc.
- ❑Medical images
- ❑Doctor's notes
- ❑Patient's medical records
- ❑Family medical history

An example of social media application



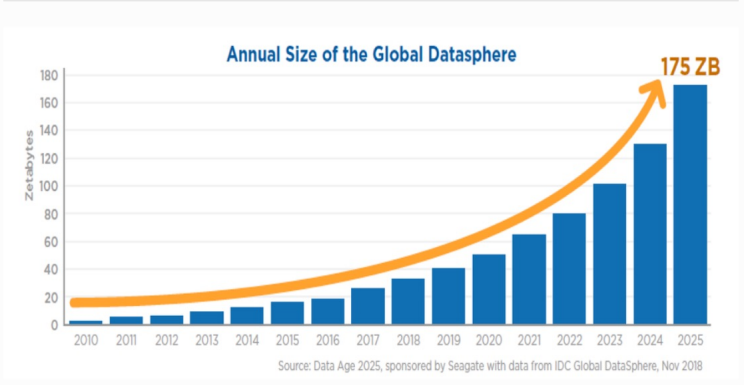
- ❑ Review score (usually 1 ~ 5)
- ❑ Review context
- ❑ Images
- ❑ News

Procedure of unstructured data analysis



The value of unstructured data

- Unstructured data make up 80% and more of enterprise data, and is growing fast
- Provide a rich source of information about people, households, and economies
- May enable more accurate and timely measurement of a range of demographic, social, economic and environmental phenomena
  - As a replacement for traditional data sources
  - **Combined with traditional data sources (semi-structured data)**
- Presents unprecedented opportunities for official statistics to
  - Improve delivery of current statistical outputs
  - Create new information products not possible with traditional data sources



https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/?sh=171a7d075459

The scale of unstructured data

How many pages of text do you read every day?

How many images you see?

How many videos you watch?

How many music/radio you listen?

How many hours of conversation you participate in?

- Some estimates suggest that we will generate 90 zettabytes of digital information in 2020.
- How do you find information available from the unstructured data?

  - "According to a McKinsey report, employees spend 1.8 hours every day—9.3 hours per week, on average—searching and gathering information. Put another way, businesses hire 5 employees but only 4 show up to work; the fifth is off searching for answers, but not contributing any value." Source: Time Searching for Information.

Text mining tasks

- Analysis of text: I have a bunch of text I am interested in, tell me something about it
  - sentiment analysis
  - topic modeling
- Retrieval: there is a large corpus of text documents, and I want the one closest to a specified query
  - web search
  - legal and medical precedent studies

Text mining: Analysis

- Which words are most present?
- Which words are most surprising?
- Which words help define the document?
- What are the interesting text phrases?
- What sentiments are being expressed?
- How are the various words clustered and how different clusters related?

Challenges of text mining

- Calculating similarity is not obvious – what is the distance between two sentences or queries?
- Evaluating retrieval is hard: what is the “right” answer? (no ground truth)
- User can query things you have not seen before, e.g., misspelled, foreign, and new terms.
- Goal (score function) is different than in classification/regression: not looking into model all of the data, just get best results for a given user
- Words can hide semantic content
  - **Synonymy**: A keyword T does not appear anywhere in the document, even though the document is closely related to T, e.g., data mining, word embedding
  - **Polysemy**: the same keyword may mean different things in different contexts, e.g. mining
  - Abbreviations
  - Slangs and domain-specific terms

Text analytics related tasks in This Class

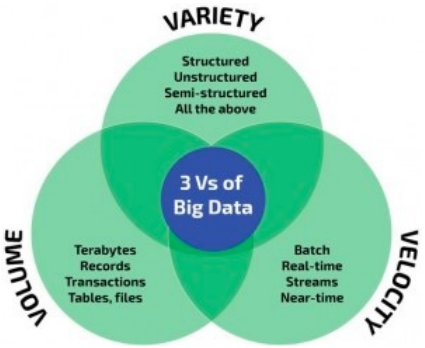
- ❑Store and Manipulate data
- ❑Topic modeling (and Sentiment Analysis)
- ❑Word Embedding

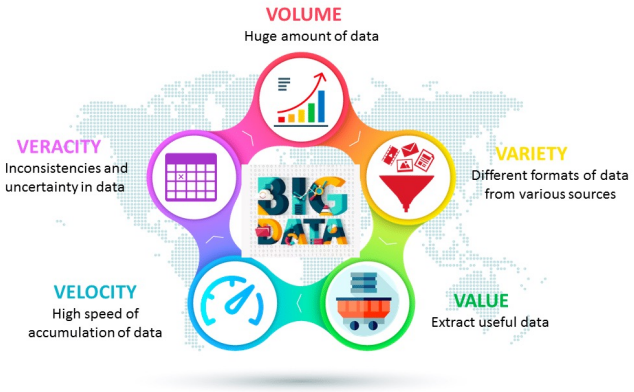
- ❑Much Much More Next Year, *Text Mining*

What NoSQL is

- Not Only SQL
- Non-relational database (not tables)
  - Store data in a format other than relational tables
  - Can store relationship data, but in a different way
  - Sometimes easy to model relationship in NoSQL, no need to split between tables
- Flexible database used for big data
- Multiple types of NoSQL databases

What is big data?



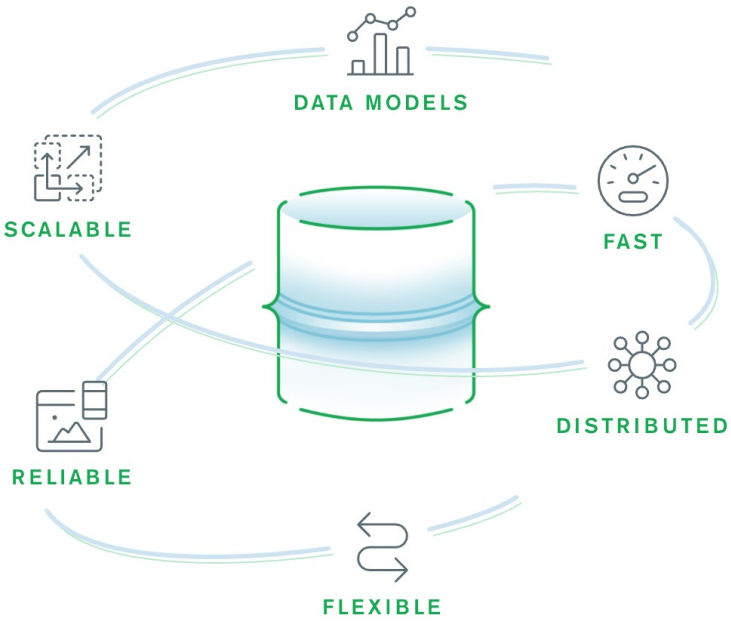


A term for datasets that are so large that traditional methods are inadequate in storage, processing, etc.

Social networks, search engines, etc.

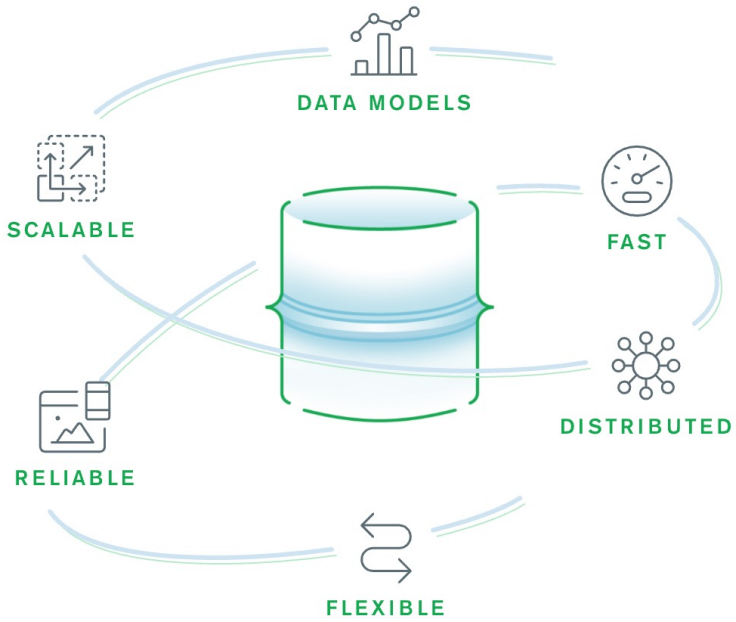
Advantages of NoSQL over SQL (RDBMS)

- Be able to handle big data
- Flexible data models, no predefined schema need
- Be able to handle unstructured data
- Cheaper to manage
- Be able to both vertical and horizontal scaling



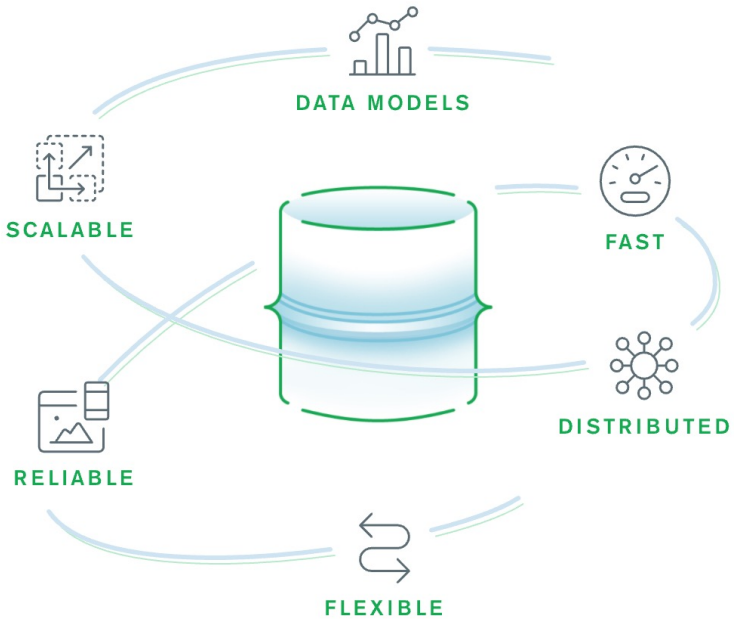
Advantages of NoSQL over SQL (RDBMS)

- Data model
  - More flexible, tailored data model to specific application
  - Key-value support simple queries very efficiently
  - Graph-based best for queries involving identifying complex relationships between separate pieces of data
- Performance
  - Often performance better than SQL
- Scalability
  - Designed to scale-out vertical and horizontal



Advantages of NoSQL over SQL (RDBMS)

- Data distribution
  - NoSQL databases are designed as distributed systems
- Reliability
  - High availability and uptime with native replications
- Flexibility
  - Flexible for end-users to test new ides and update data structure



Advantages of SQL over NoSQL

- Better for relational data
- Normalization
- Better for minimizing data redundancy (for storage purpose)
- Well defined structure query language (SQL)
- Data integrity

Four major types of NoSQL databases

- ❑Key-value databases
- ❑Graph databases
- ❑Document databases
- ❑Column family

Key-value databases

- Each item contains (key, value)
- Very useful when
  - Store large amounts of data with no need of complex queries to retrieve it
- Redis, DynanoDB
- Provide the key and get the data

FIGURE 14.7 KEY-VALUE DATABASE STORAGE

Bucket = Customer

Key	Value
10010	"LName Ramas FName Alfred Initial A Areacode 615 Phone 844-2573 Balance 0"
10011	"LName Dunne FName Leona Initial K Areacode 713 Phone 894-1238 Balance 0"
10014	"LName Orlando FName Myron Areacode 615 Phone 222-1672 Balance 0"

Document-oriented database

- Store data in documents similar to JSON objects
- Each document contains pairs of (filed, value)
- The values can be:
  - Strings
  - Numbers
  - Booleans
  - Arrays
  - Objects
- MongoDB

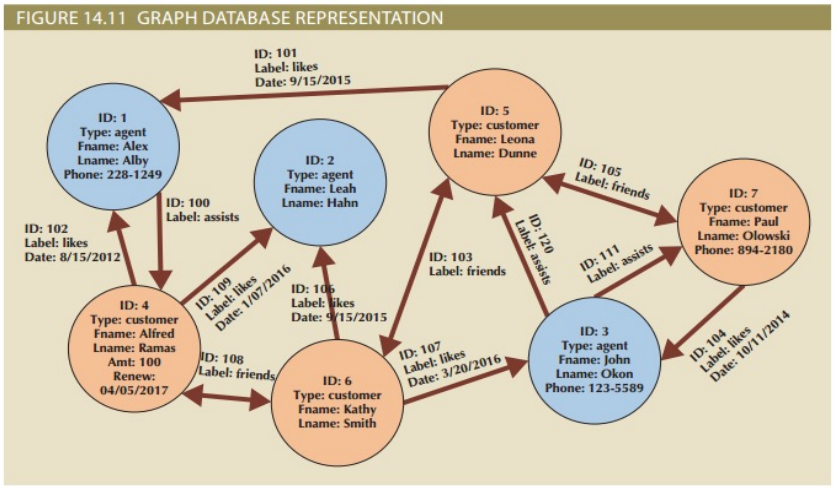
FIGURE 14.8 DOCUMENT DATABASE TAGGED FORMAT

Collection = Customer

Key	Document
10010	{LName: "Ramas", FName: "Alfred", Initial: "A", Areacode: "615", Phone: "844-2573", Balance: "0"}
10011	{LName: "Dunne", FName: "Leona", Initial: "K", Areacode: "713", Phone: "894-1238", Balance: "0"}
10014	{LName: "Orlando", FName: "Myron", Areacode: "615", Phone: "222-1672", Balance: "0"}

Graph-based database

- Store data in nodes and edges
  - Nodes—information about people, places, and things
  - Edges-–information about relationships between nodes
- Useful when
  - Traverse relationships to look for patterns such as social networks, fraud detection and recommendation
- Neo4j and JanusGraph



Column-oriented database

- Store data in tables, rows and dynamic columns
  - Each row is not required to have the same columns
  - Row: consists of (key, one or more related columns (column families))
  - Fast aggregation queries
- Useful when
  - Store large amounts of data
  - Can predict the query patterns
  - Commonly used for IoT and user profile data
  - E.g., quickly aggregate the value of a given column
- Cassandra and HBase

FIGURE 14.9 COMPARISON OF ROW-CENTRIC AND COLUMN-CENTRIC STORAGE

CUSTOMER relational table

Cus_Code	Cus_LName	Cus_FName	Cus_City	Cus_State
10010	Ramas	Alfred	Nashville	TN
10011	Dunne	Leona	Miami	FL
10012	Smith	Kathy	Boston	MA
10013	Olowski	Paul	Nashville	TN
10014	Orlando	Myron		
10015	O'Brian	Amy	Miami	FL
10016	Brown	James		
10017	Williams	George	Mobile	AL
10018	Farriss	Anne	Opp	AL
10019	Smith	Olette	Nashville	TN

Row-centric storage

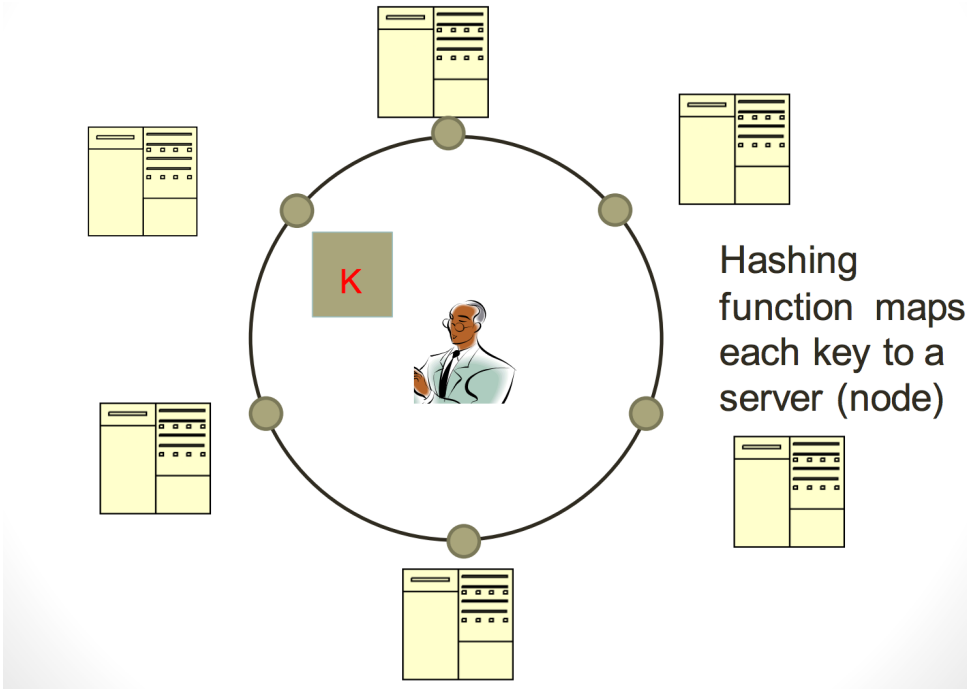
Block 1	Block 4
10010,Ramas,Alfred,Nashville,TN 10011,Dunne,Leona,Miami,FL	10016,Brown,James,NULL,NULL 10017,Williams,George,Mobile,AL
Block 2	**Block 5**
10012,Smith,Kathy,Boston,MA 10013,Olowski,Paul,Nashville,TN	10018,Farriss,Anne,OPP,AL 10019,Smith,Olette,Nashville,TN
Block 3	
10014,Orlando,Myron,NULL,NULL 10015,O'Brian,Amy,Miami,FL	

Column-centric storage

Block 1	Block 4
10010,10011,10012,10013,10014 10015,10016,10017,10018,10019	Nashville,Miami,Boston,Nashville,NULL Miami,NULL,Mobile,Opp,Nashville
Block 2	**Block 5**
Ramas,Dunne,Smith,Olowski,Orlando O'Brian,Brown,Williams,Farriss,Smith	TN,FL,MA,TN,NULL, FL,NULL,AL,AL,TN
Block 3	
Alfred,Leona,Kathy,Paul,Myron Amy,James,George,Anne,Olette	

Typical NoSQL architecture

Hashing function maps each key to a server (node)
h(K) = location of Server



CAP Theorem for NoSQL

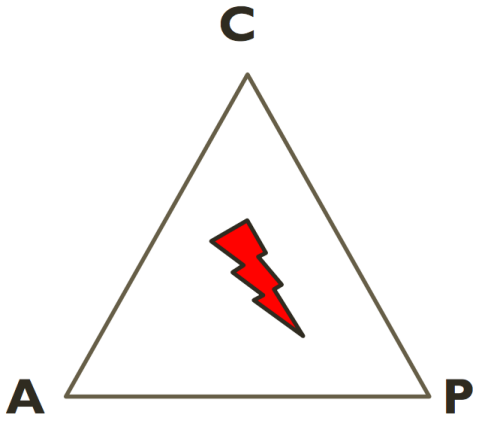
- What the CAP theorem really says:
  - **If you cannot limit the number of faults and requests can be directed to any server and you insist on serving every request you receive then you cannot possible be consistent**

- How it is interpreted:
  - **You must always give something up: Consistency, Availability or Tolerance to failure and reconfiguration**

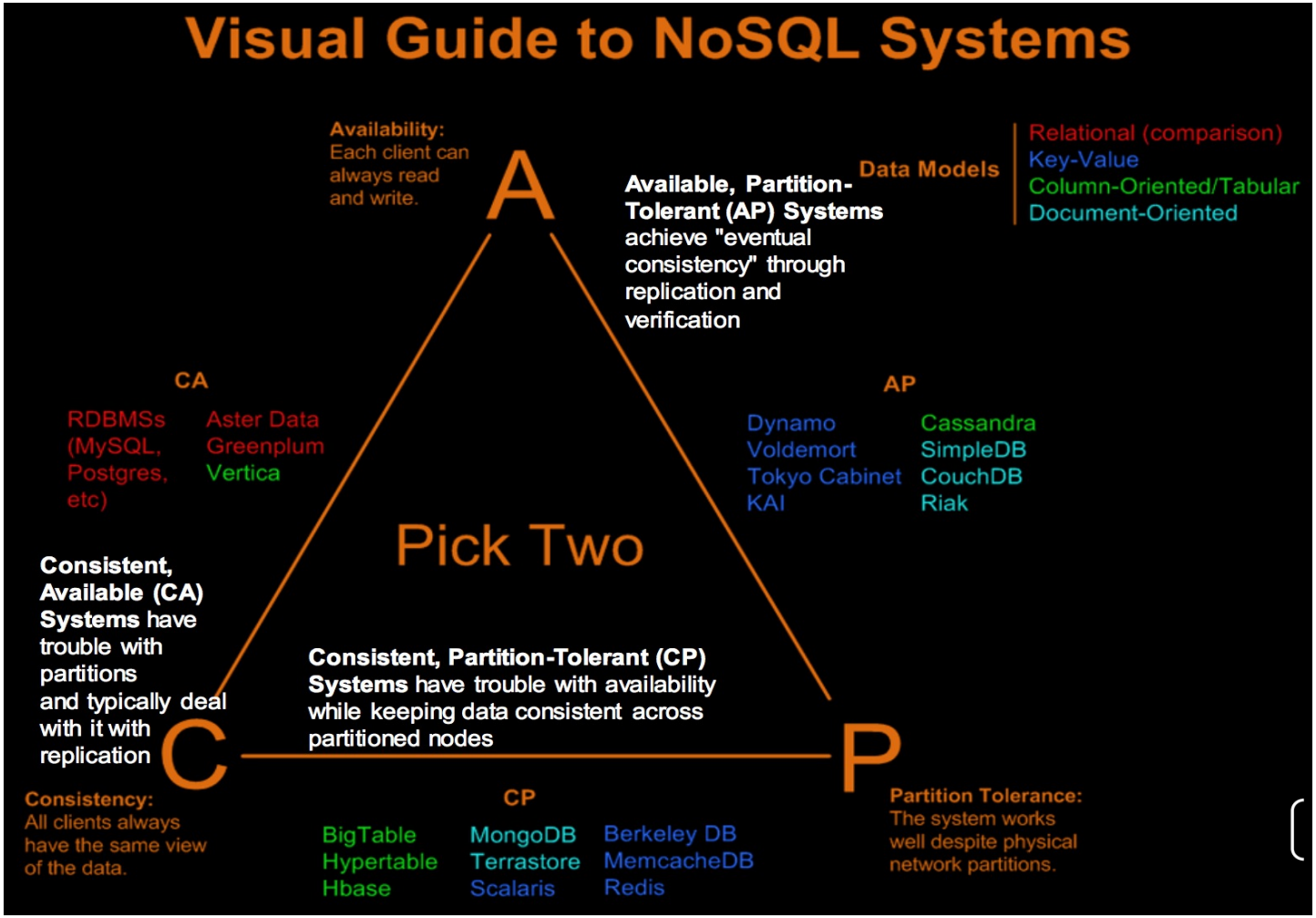
According to Eric Brewer 2001

Theory of NoSQL: CAP

- Given:
  - Many nodes
  - Nodes contain replicas of partitions of the data
- **C**onsistency
  - All replicas contain the same version of data
  - Client always has the same view of the data (no matter what node)
- **A**vailability
  - System remains operational on failing nodes
  - All clients can always read and write
- **P**artition tolerance
  - Multiple entry points
  - System remains operational on system split (communication malfunction)
    - System works well across physical network partitions



CAP Theorem: satisfying all three at the same time is impossible



http://blog.nahurst.com/visual-guide-to-nosql-systems

Sharding of data

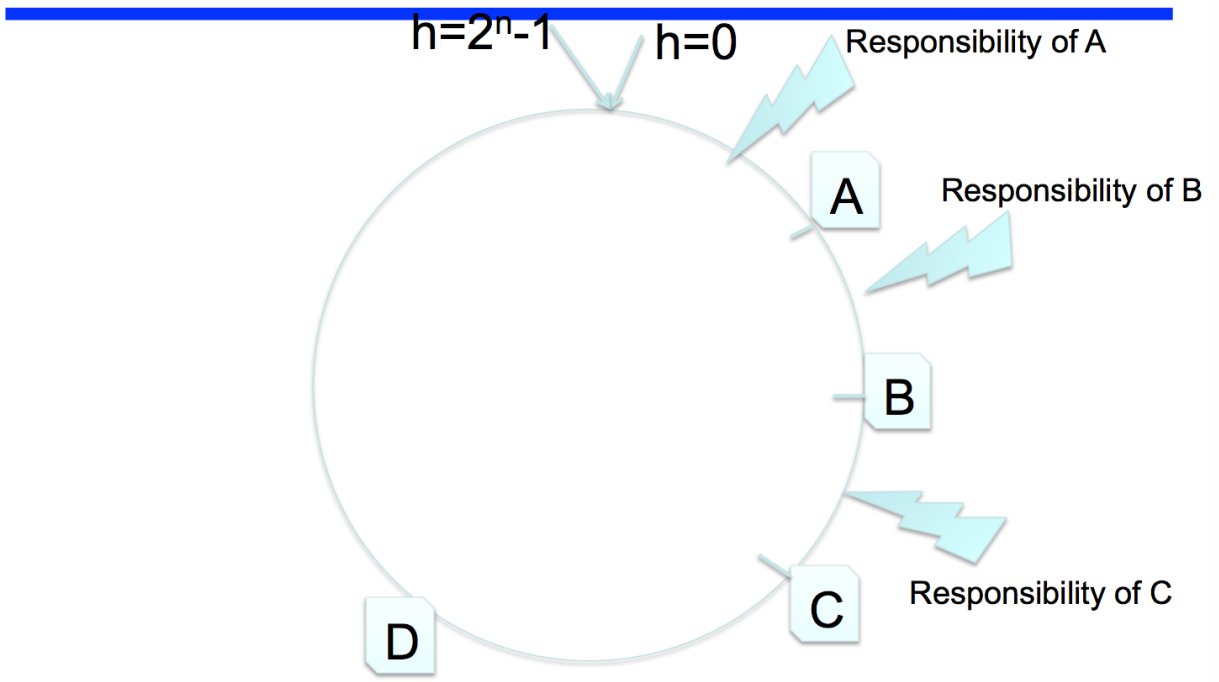
- Distributes a single logical database system across a cluster of machines
- Uses range-based partitioning to distribute documents based on a specific shard key
- Automatically balances the data associated with each shard
- Can be turned on and off per collection (table)

Storage: distributed hash table

- Implements a distributed storage
- Each key-value pair (k,v) is stored at some server h(k)
- API: write(k,v); read(k)

- Use standard hash function: service key k by server h(k)

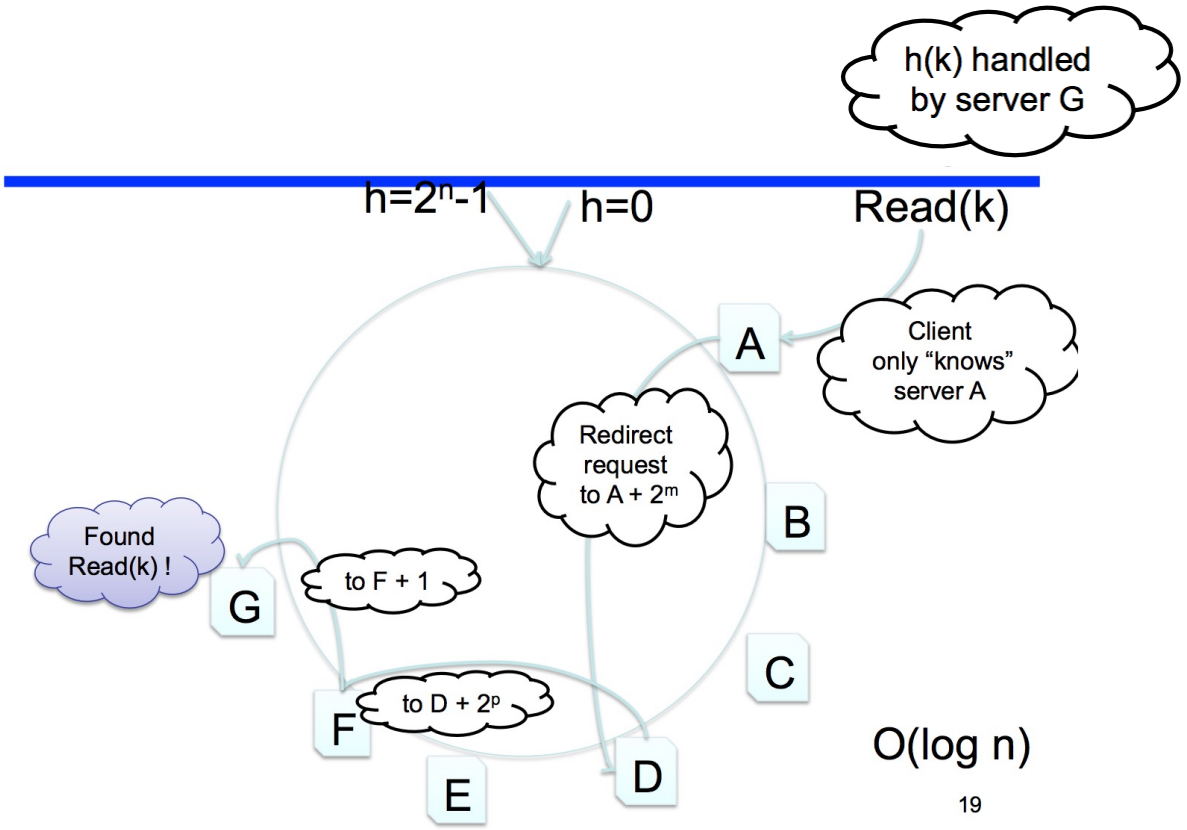
Distributed hash table



Routing

- A client doesn't know server h(k), but some other server
- Naïve routing algorithm:
  - Each node knows its neighbors
  - Send message to nearest neighbor
  - Hop-by-hop from there
  - Obviously, this is O(n)
- Better algorithm: "finger table"
  - Memorize locations of other nodes in the ring
  - a, a+2, a+4, …, a+2**n-1
  - Send message to closest node to destination
  - Hop-by-hop again: this log(n)

Routing

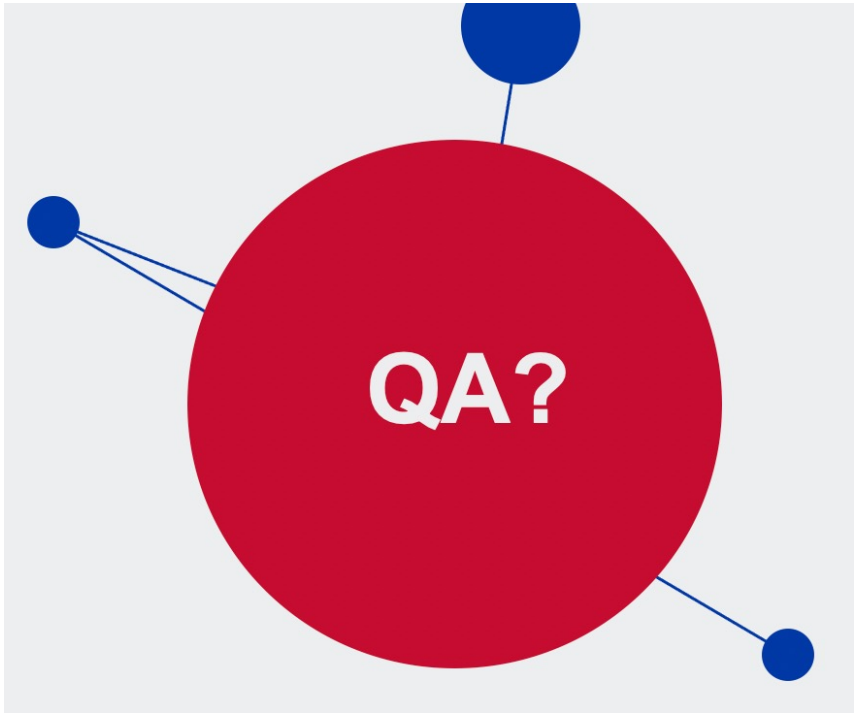

h(k) handled by server G
h=2^n-1
h=0
Read(k)
A
Client only "knows" server A
Redirect request to A + 2^m
B
Found Read(k) !
to F + 1
G
C
to D + 2^p
F
E
D
O(log n)
19

Replication

- Need to have some degree of replication to cope with node failures
- N Replication
  - N=degree of replication,
  - Assign key k to h(k), h(k)+1, …, h(k)+N-1

Comparison between SQL and NoSQL

	SQL	NoSQL
Characteristics	❑ Data uses schemes ❑ Relations ❑ Data is distributed across multiple tables ❑ Horizontal scaling is difficult or impossible; vertical scaling is possible ❑ Limitations for lots of (thousands) read & write queries per second	❑ Schema-less ❑ No (or very few) Relations ❑ Data is typically merged/nested in a few collections ❑ Both horizontal and vertical scaling is possible ❑ Greater performance for mass (simple) read & write requests


QA?