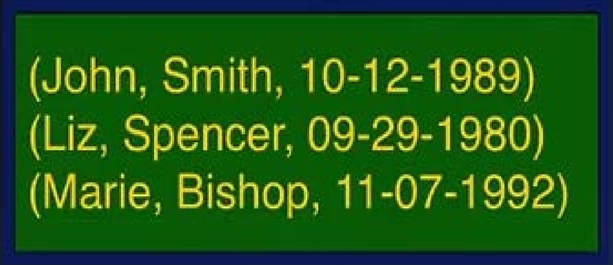
Big Data (MHI222956/MHI225101)

3.1 NoSQL Database

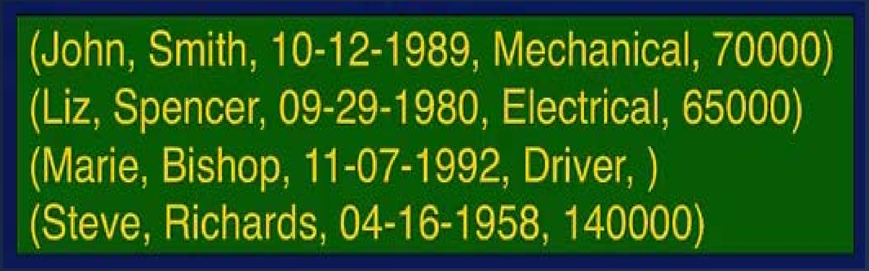
Structured and Unstructured data

Structured



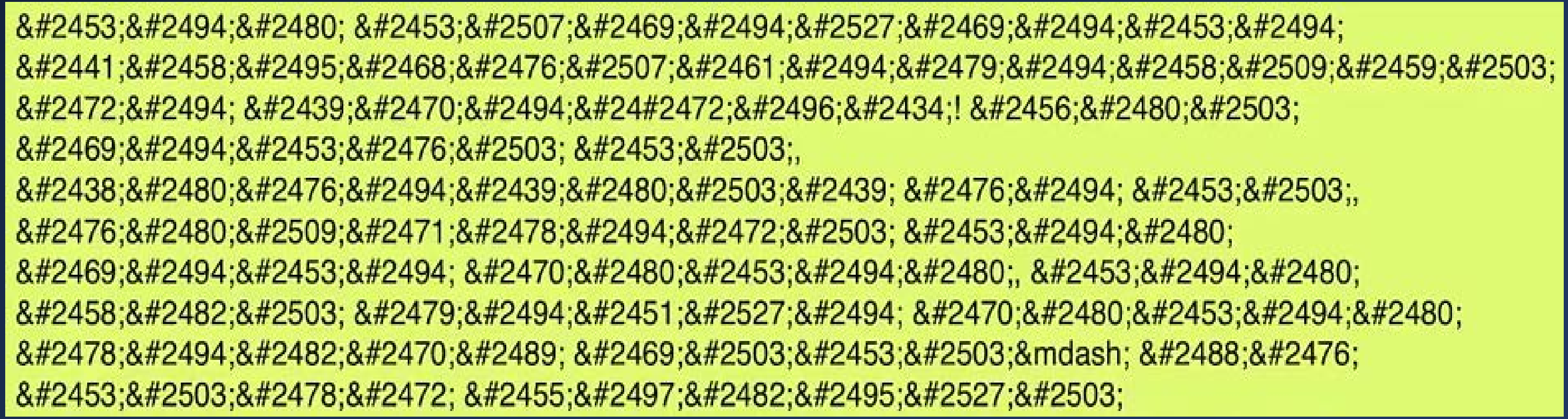
(John, Smith, 10-12-1989)
(Liz, Spencer, 09-29-1980)
(Marie, Bishop, 11-07-1992)

Semi-structured



(John, Smith, 10-12-1989, Mechanical, 70000)
(Liz, Spencer, 09-29-1980, Electrical, 65000)
(Marie, Bishop, 11-07-1992, Driver,)
(Steve, Richards, 04-16-1958, 140000)

Unstructured



```
&#2453;&#2494;&#2480; &#2453;&#2507;&#2469;&#2494;&#2527;&#2469;&#2494;&#2453;&#2494;
&#2441;&#2458;&#2495;&#2468;&#2476;&#2507;&#2461;&#2494;&#2479;&#2494;&#2458;&#2509;&#2459;&#2503;
&#2472;&#2494; &#2439;&#2470;&#2494;&#24#2472;&#2496;&#2434;! &#2456;&#2480;&#2503;
&#2469;&#2494;&#2453;&#2476;&#2503; &#2453;&#2503;;,
&#2438;&#2480;&#2476;&#2494;&#2439;&#2480;&#2503;&#2439; &#2476;&#2494; &#2453;&#2503;;,
&#2476;&#2480;&#2509;&#2471;&#2478;&#2494;&#2472;&#2503; &#2453;&#2494;&#2480;
&#2469;&#2494;&#2453;&#2494; &#2470;&#2480;&#2453;&#2494;&#2480;;, &#2453;&#2494;&#2480;
&#2458;&#2482;&#2503; &#2479;&#2494;&#2451;&#2527;&#2494; &#2470;&#2480;&#2453;&#2494;&#2480;
&#2478;&#2494;&#2482;&#2470;&#2489; &#2469;&#2503;&#2453;&#2503;&mdash; &#2488;&#2476;
&#2453;&#2503;&#2478;&#2472; &#2455;&#2497;&#2482;&#2495;&#2527;&#2503;
```

Relational databases

- The best example of structured data
- The primary data structure for a relational model is a table, which is also called a relation.
- A table actually represents a set of tuples (rows). A relational tuple implies that unless otherwise stated, the elements of it are atomic: represent one unit of information and cannot be decomposed further.
- Relational Algebra - providing a theoretical foundation for relational databases, particularly query languages for such databases, chief among which is SQL.

Features of relational databases

- Persistence
  - Database stores large amounts of data in a way that allows an application program to get at small bits of that information quickly and easily
- Concurrency
  - Allow many concurrent accesses, controlled using transactions
- Referential integrity
  - Uses primary and foreign keys to maintain relationships between tables and ensure data is consistent
- Integration
  - Traditional enterprise approach to integration of different applications within the organisation is to use shared database with concurrency control
- Standard model
  - Relational databases provide these capabilities using a (mostly) standard data model and SQL

Limitations of RDBMs

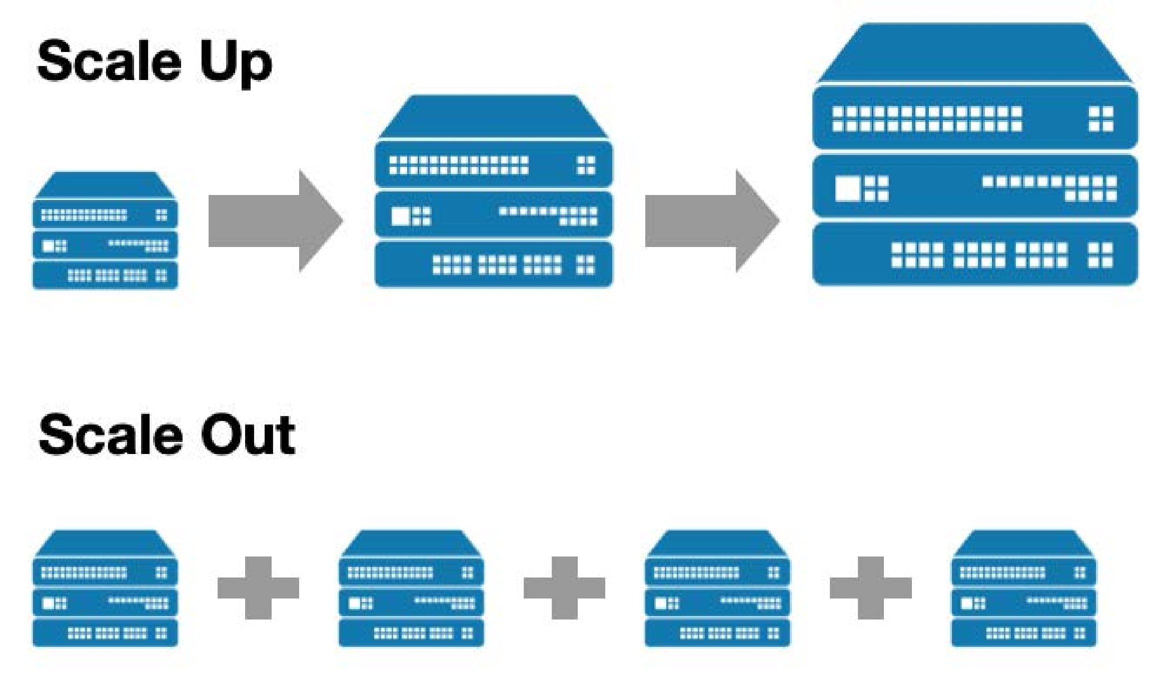
- Unstructured data may have its own internal structure, but does not conform neatly into a spreadsheet or relational database.

- For RDBMs it becomes a real challenge to provide the cost effective and fast Create, Read, Update and Delete (CRUD) operation as it has to deal with the overhead of joins and maintaining relationships amongst various data.

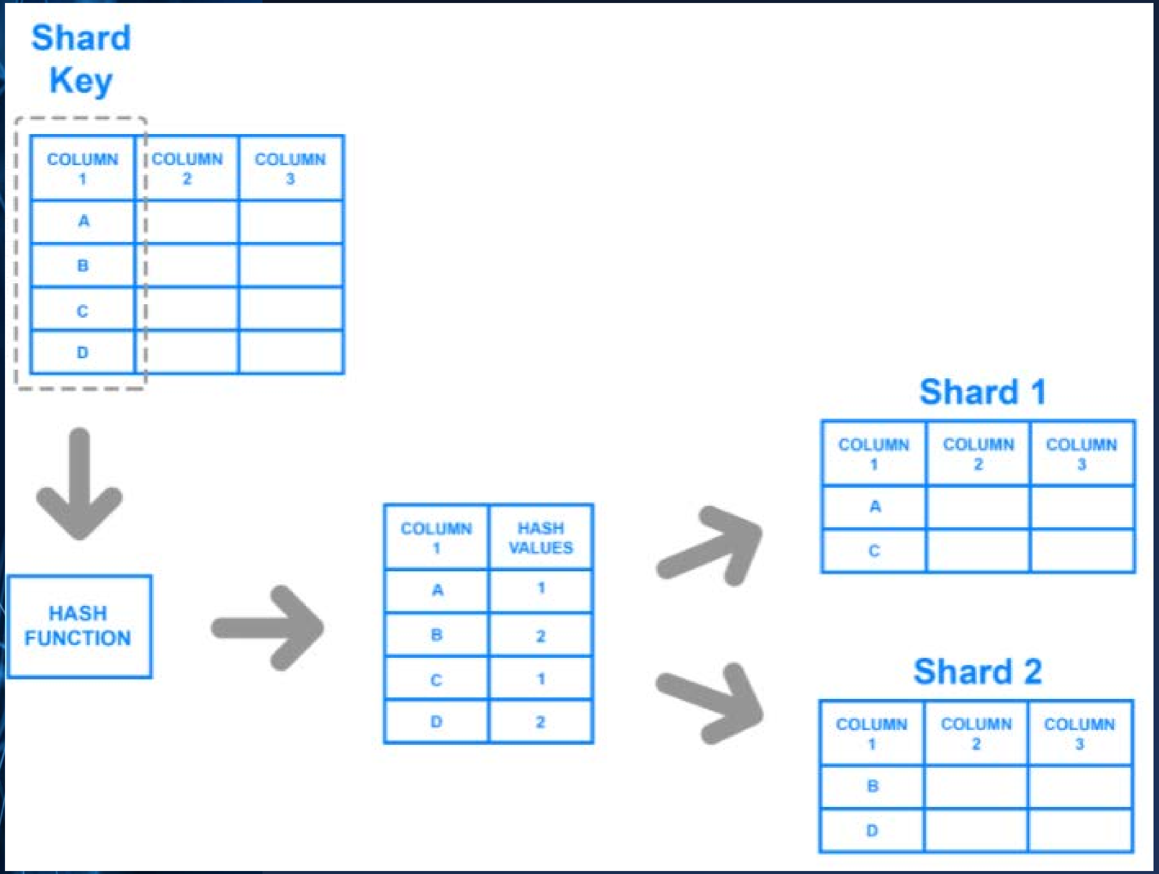
Scaling

- In the 2000s web businesses with massive user bases found relational databases don’t scale easily to match their new requirements in data volume and throughput.

- Coping with the increase required more computing resources - two choices:
  - Scaling up – Vertical scaling
  - Scaling out – Horizontal scaling



Sharding



Shard Key

COLUMN 1	COLUMN 2	COLUMN 3
A		
B		
C		
D		

HASH FUNCTION

COLUMN 1	HASH VALUES
A	1
B	2
C	1
D	2

Shard 1

COLUMN 1	COLUMN 2	COLUMN 3
A		
C		

Shard 2

COLUMN 1	COLUMN 2	COLUMN 3
B		
D		

- Sharding is a database architecture pattern related to horizontal partitioning - the practice of separating one table’s rows into multiple different tables, known as partitions

- Different data on different nodes in a cluster, each of which does its own reads and writes

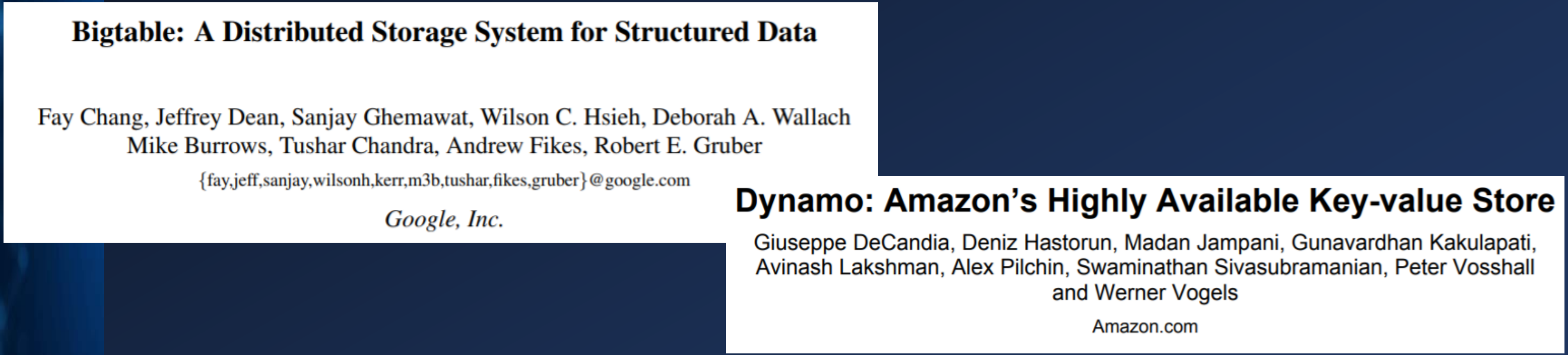
- Try to keep data that belongs together on same node

Scaling and relational databases

- Traditional cluster-aware RDBMS use a shared disk subsystem
- Although data is distributed and highly available there is a single point of failure
- Concept of sharding does not sit well with relational databases
- Lose any querying, referential integrity, transactions, or consistency controls that cross shards, so lose many of the benefits of the relational model
- Most RDBMS don’t have automatic sharding built-in

Origins of “NoSQL”

- Google (BigTable) and Amazon (Dynamo) developed databases to scale out, not up, and published papers on this



Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber

{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com

Google, Inc.

Dynamo: Amazon’s Highly Available Key-value Store

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati,
Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall
and Werner Vogels

Amazon.com

- Others recognised that they had similar needs and a range of other databases emerged along similar lines – NoSQL databases

NoSQL database

- The name “NoSQL” was in fact first used by Carlo Strozzi in 1998 as the name of file-based database he was developing.

- The term re-surfaced in 2009 when Eric Evans used it to name the current surge in non-relational databases.

- NoSQL stands for “Not Only SQL”
  - do not use SQL as their primary query language
  - providing access by means of Application Programming Interfaces (APIs).

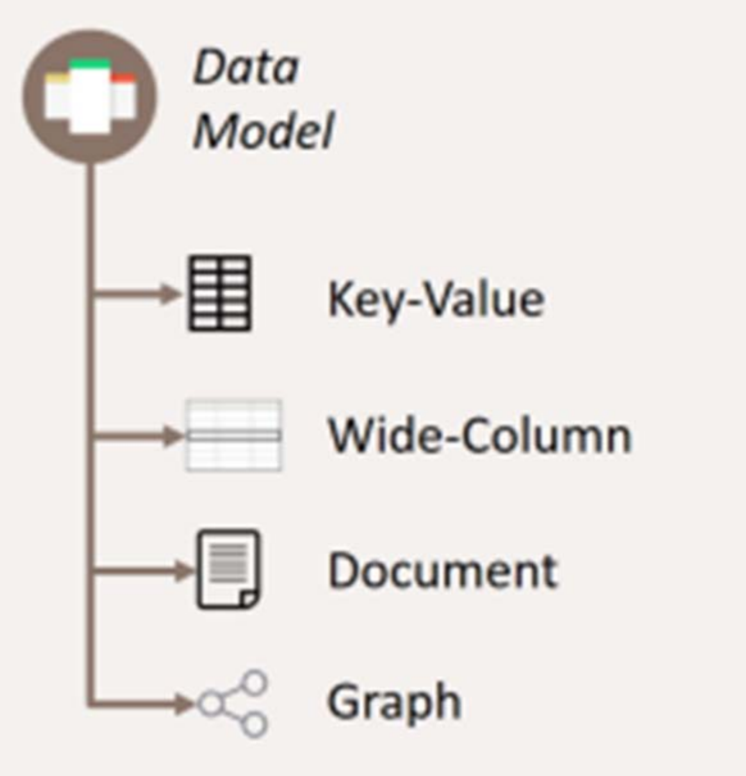
- Many different database systems exist or have existed

  > 225 currently listed on http://nosql-database.org/

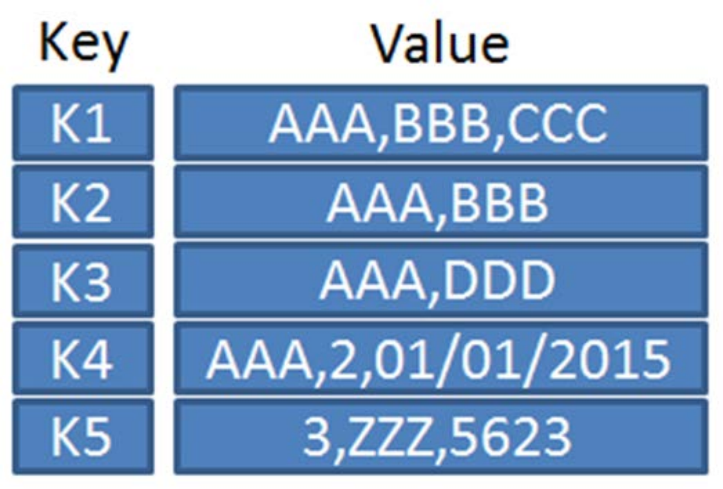
Types of NoSQL database

- Significant variety among NoSQL databases, particularly in terms of the data model used.

- Four main types, each with its own data model, although boundaries are blurred:
  - Key-value
  - Document
  - Wide column stores
  - Graph



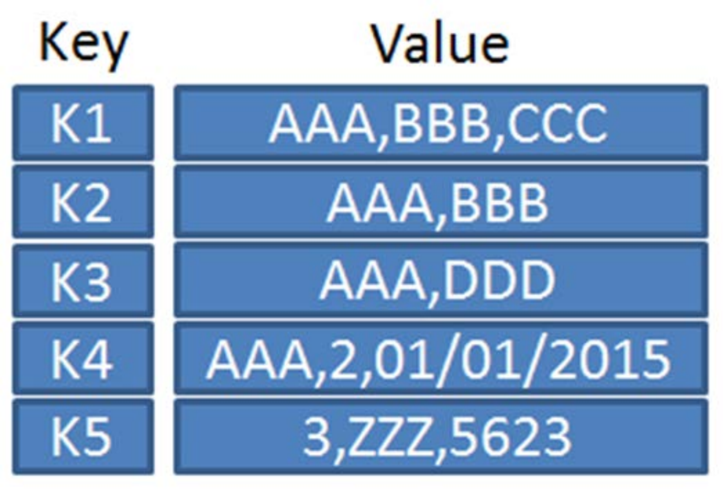
Key-value Database



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

- A data storage paradigm designed for storing, retrieving, and managing associative arrays, a data structure more commonly known today as a dictionary or hash.
- These manage a simple value or row, indexed by a key. There is no schema and the value of the data is opaque.
- Simplest and most flexible type of NoSQL database.
- The client can either get the value for the key, put a value for a key, or delete a key from the data store.
- The value is a blob (*binary large object*) that the data store just stores, without caring or knowing what’s inside.

Key-value Database



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

- Responsibility of the application to understand what was stored.
- Generally have good performance and can be easily scaled.

  e.g. DynamoDB, Voldemort, Riak, Redis, Azure Table Storage

- Suitable use cases:
  - Session information
  - User profiles and preferences
  - Shopping cart data
- Don’t use for:
  - Data with relationships
  - Transactions with multiple operations
  - Querying based on content of data
  - Operations on sets of data

Document-oriented Database

- Multi-field indexable / queryable documents
- Documents are self-describing, hierarchical tree data structures which can consist of lists/arrays (sequential collections of values), maps (key/value collections), and scalar values

Documents stored are generally similar to each other but do not have to be exactly the same – schema-less data

- Document format usually JSON (JavaScript Object Notation) , its binary equivalent BSON, or XML
- Documents stored are generally similar to each other but do not have to be exactly the same – schema-less data

```
{
        _id: ObjectId(5fd976a0902d),
        title: 'MongoDB in Action',
        description: 'MongoDB is NoSQL database',
        tags: [‘MongoDB', 'database', 'NoSQL'],
        likes: 100
}
```

Document-oriented Database

- Suitable use cases
  - Event logs
  - Content management systems/blogging platforms
  - Analytics
  - eCommerce applications
- Don’t use for
  - Complex cross-document transactions
  - Queries that rely on fixed schema

- Related to key/value store:

  essentially databases store documents in the value part of the key-value store, but can examine/query the contents of the documents
- Example

  MongoDB, RavenDB, CouchDB

```
{
        _id: ObjectId(5fd976a0902d),
        title: 'MongoDB in Action',
        description: 'MongoDB is NoSQL database',
        tags: [‘MongoDB', 'database', 'NoSQL'],
        likes: 100
}
```

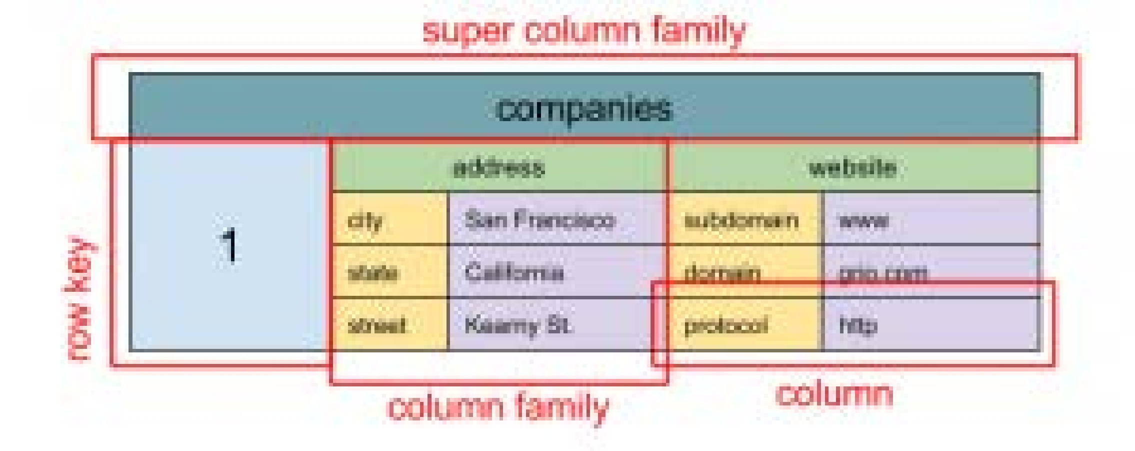
Wide column stores/column families

- Inspired by the Google Big table paper (https://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf).

  “A sparse, distributed, persistent multidimensional sorted map”

  - Google’s description of their BigTable

- Examples - Google BigTable, Hbase, HyperTable, Cassandra, Amazon SimpleDB (Following description mostly based on Cassandra, but others have similar concepts)



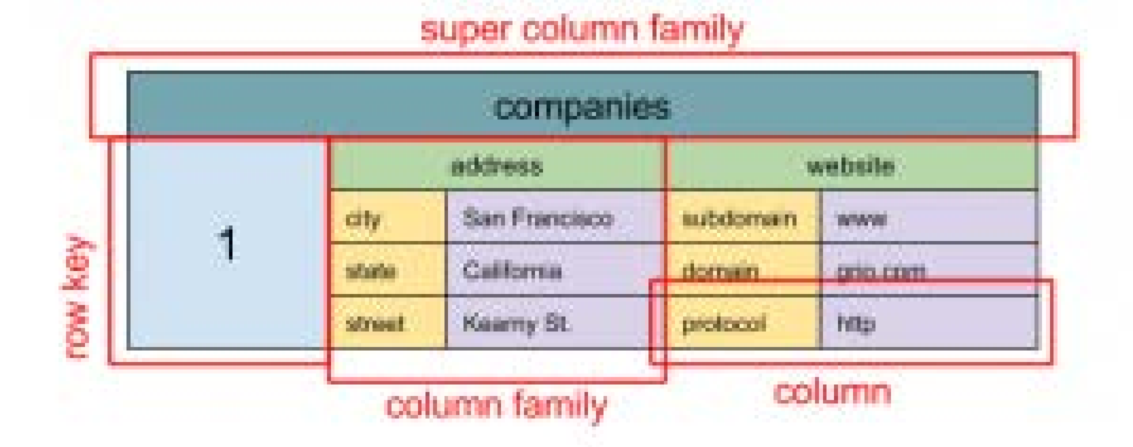
	companies			
	address		website	
1	city	San Francisco	subdomain	www
	state	California	domain	gnip.com
	street	Kearny St.	protocol	http

Wide column stores/column families

- Column consists of a name-value pair where the name also behaves as the key - each of these key-value pairs is a single column and is always stored with a timestamp value.
- A row is a collection of columns attached or linked to a key - a collection of similar rows makes a column family.
- Each column family can be compared to a container of rows in an RDBMS table where the key identifies the row and the row consists of multiple columns, except that various rows do not have to have the same columns

  - columns can be added to any row at any time without having to add it to other rows

- A column can consists of a map of columns - super column

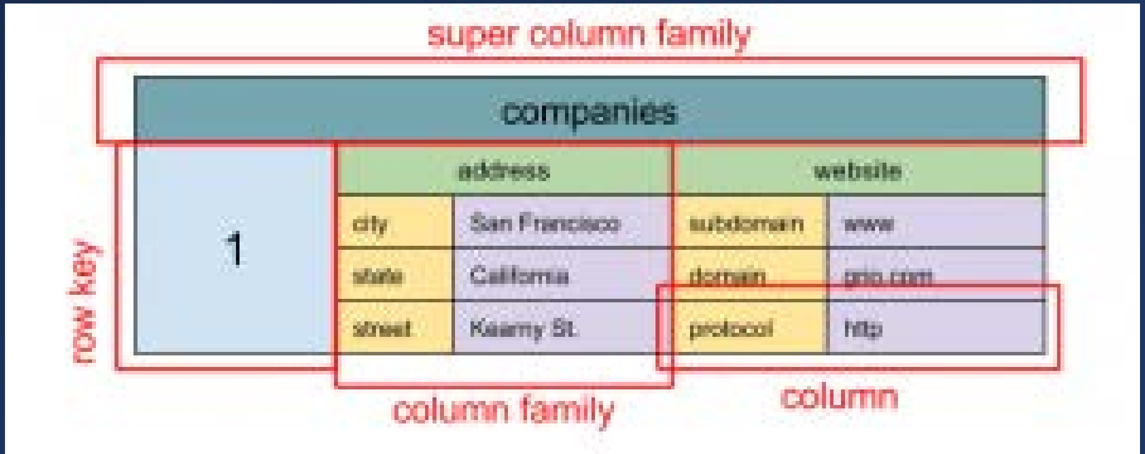


companies				
	address		website	
1	city	San Francisco	subdomain	www
	state	California	domain	grio.com
	street	Kearny St.	protocol	http

Wide column stores/column families

- Suitable use cases
  - Event logs
  - Content management systems/blogging platforms
  - Social network messaging systems (e.g. Facebook uses Hbase)
  - Analytics on large data sets (e.g. Hbase & Hadoop)

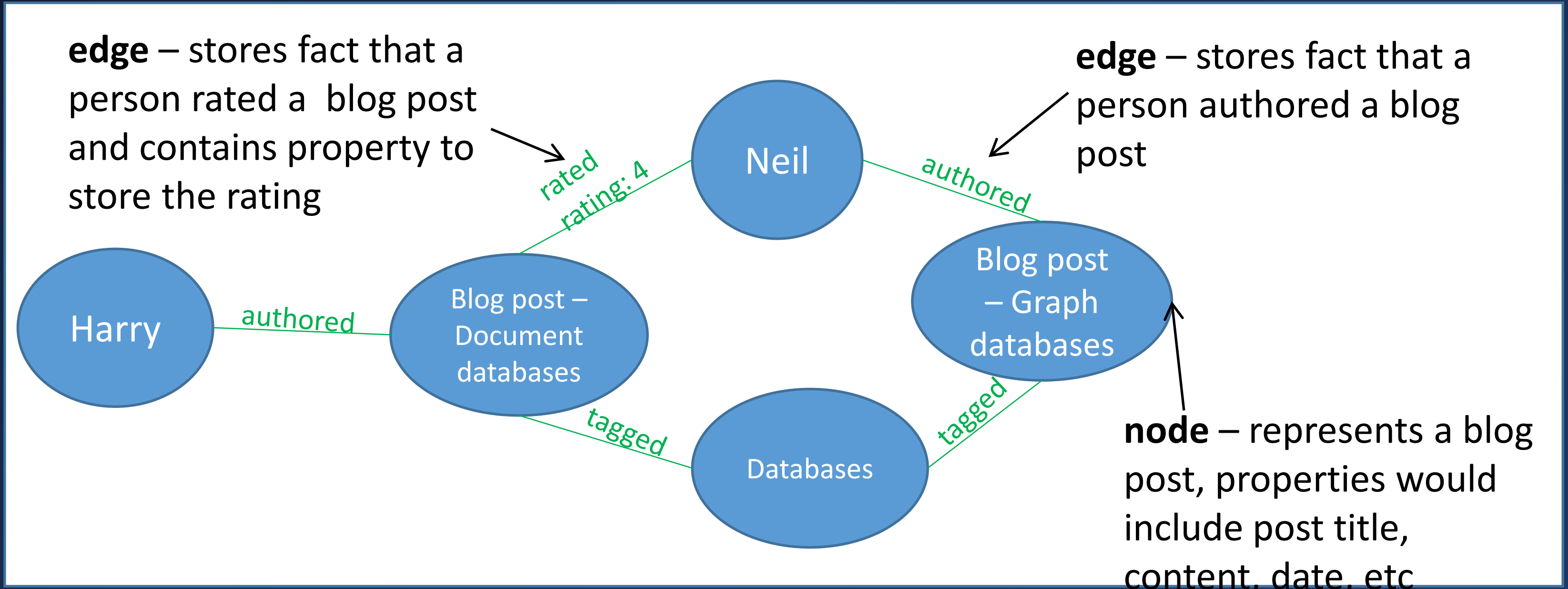
- Don’t use for
  - Complex cross-document transactions
  - Queries that rely on fixed schema



companies				
	address		website	
1	city	San Francisco	subdomain	www
	state	California	domain	gnip.com
	street	Kearny St.	protocol	http

Graph databases

- A graph is composed of two elements:
  - Node - represents an entity (a person, place, thing, category or other piece of data),
  - Relationship (edge) - represents how two nodes are associated.



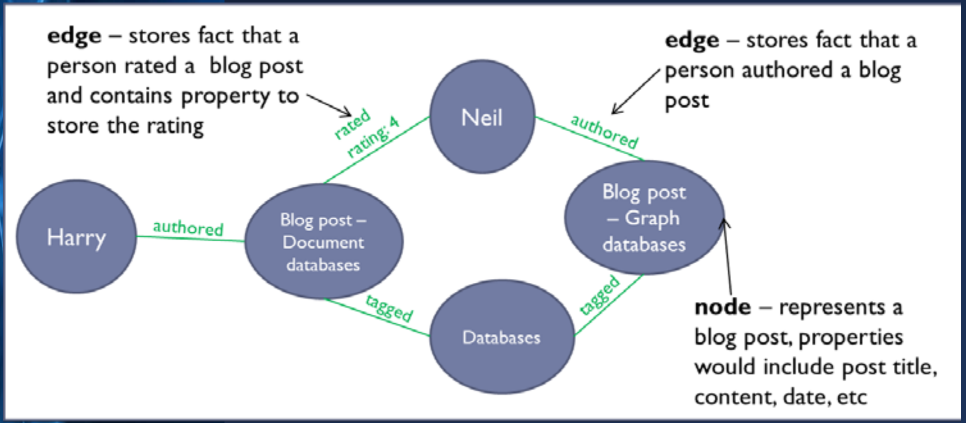
edge – stores fact that a person rated a blog post and contains property to store the rating

edge – stores fact that a person authored a blog post

node – represents a blog post, properties would include post title, content, date, etc

Graph databases

- A graph database is an online database management system with Create, Read, Update and Delete (CRUD) operations working on a graph data model.
  - Relationships take first priority in graph databases
  - The data model for a graph database is simpler and expressive

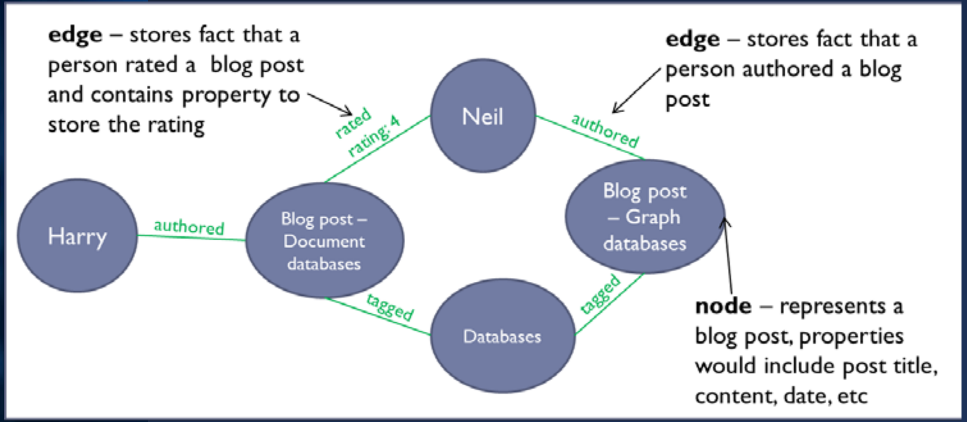

edge – stores fact that a person rated a blog post and contains property to store the rating
edge – stores fact that a person authored a blog post
node – represents a blog post, properties would include post title, content, date, etc
Neil
Harry
Blog post – Document databases
Blog post – Graph databases
Databases
rated
rating: 4
authored
authored
tagged
tagged

- Edges have directional significance and allow you to find interesting patterns between the nodes
- Query using specialised graph traversal languages (Gremlin, Cypher)

Graph databases

- Different from a relational database which stores entities with relationships defined by values of their properties – in graph databases, the relationships are stored explicitly as database objects

- Example: Neo4j, InfiniteGraph, Titan



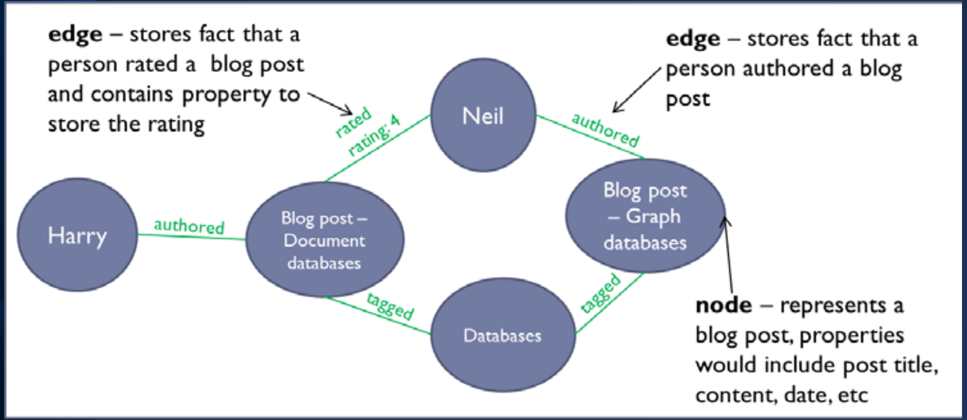
edge – stores fact that a person rated a blog post and contains property to store the rating
edge – stores fact that a person authored a blog post
Neil
rated
rating: 4
authored
Harry
authored
Blog post – Document databases
Blog post – Graph databases
Databases
tagged
tagged
node – represents a blog post, properties would include post title, content, date, etc

- Some graph databases use other NoSQL types as backed storage

  e.g Titan DB can be configured to use Cassandra or HBase

Graph databases

- Suitable use cases
  - Social networks or similar connected data scenario
  - Recommender systems
  - Security analysis
  - Routing and location-based services



edge – stores fact that a person rated a blog post and contains property to store the rating

edge – stores fact that a person authored a blog post

node – represents a blog post, properties would include post title, content, date, etc

- Don’t use for
  - Operations on sets of data

Other types…

- Multi-model databases
  - aim to simplify this by combining different NoSQL types in a single database with unified query language and API

- NewSQL database – a class of modern relational database management systems that seek to:
  - provide the same scalable performance of NoSQL systems for read-write workloads
  - maintain the consistency guarantees of a traditional relational database system

Schema-less databases

- Most NoSQL databases are considered to be schema-less
- No fixed schema defined at database level
    - e.g. documents stored in a document store can have varying sets of fields and field types
- Contrast to fixed schemas defined in relational databases
- In practice, there needs to be some kind of schema, as the application needs to understand what is in the database in order to use it
- “Schema” is defined in the application, not the database

Summary

- Limitations of relational databases
- Origins of “NoSQL”
- Key-value database
- Document-oriented database
- Wide column stores/column families
- Graph databases