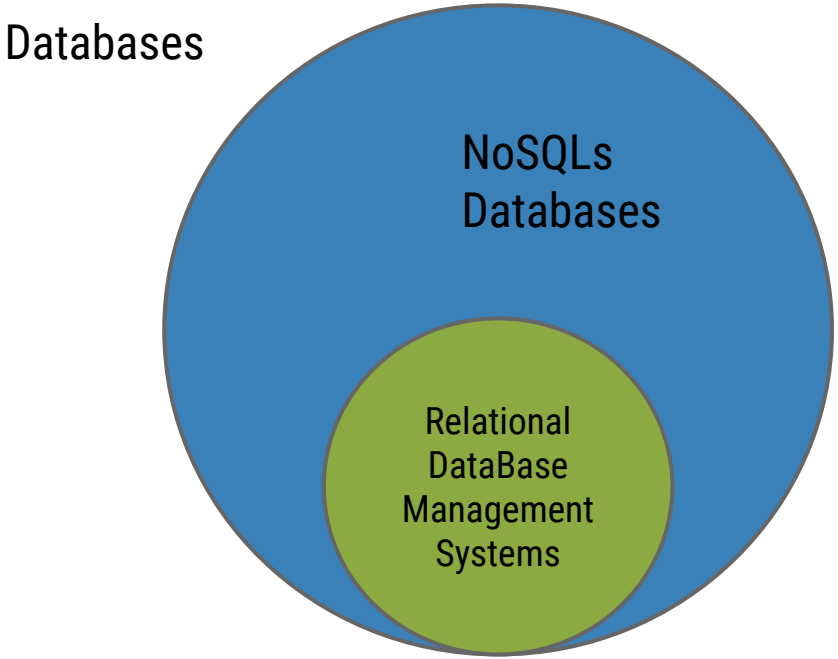
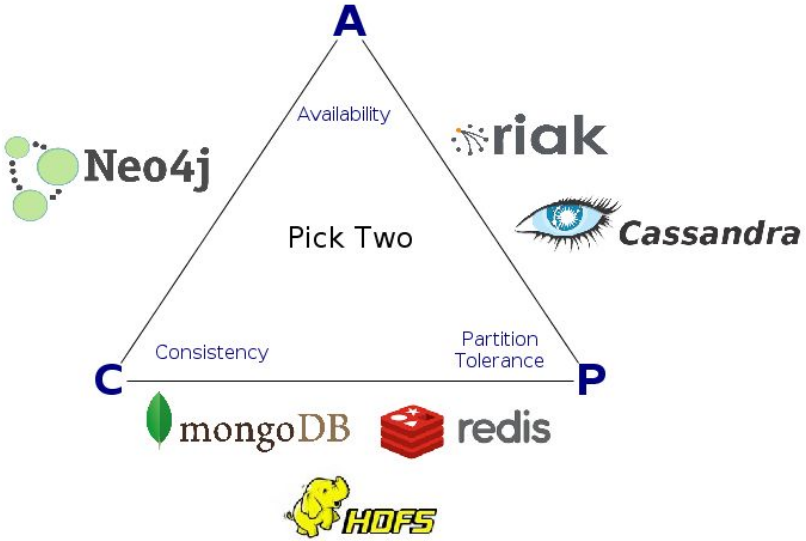
Module II: Introduction to NoSQL

NoSQL?



We will define NoSQL databases as all of the modern databases that cannot or currently are not used through a relational schema.

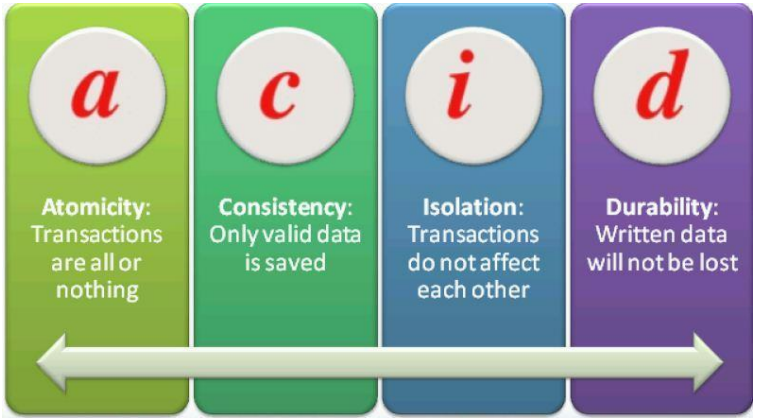
The CAP Theorem/Brewers

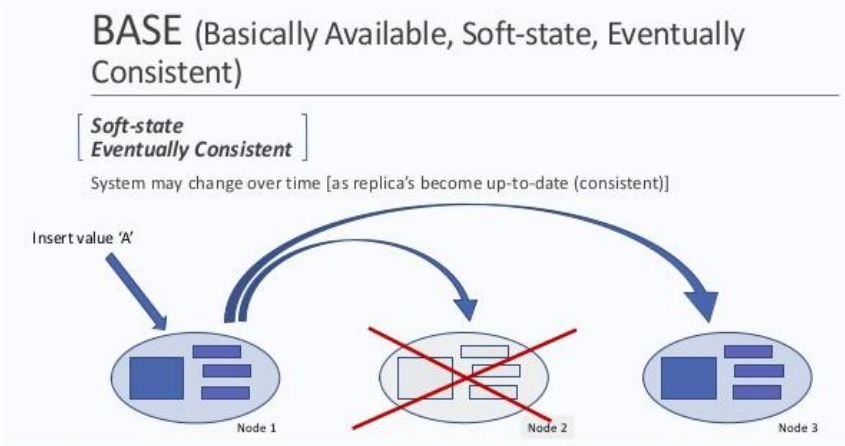


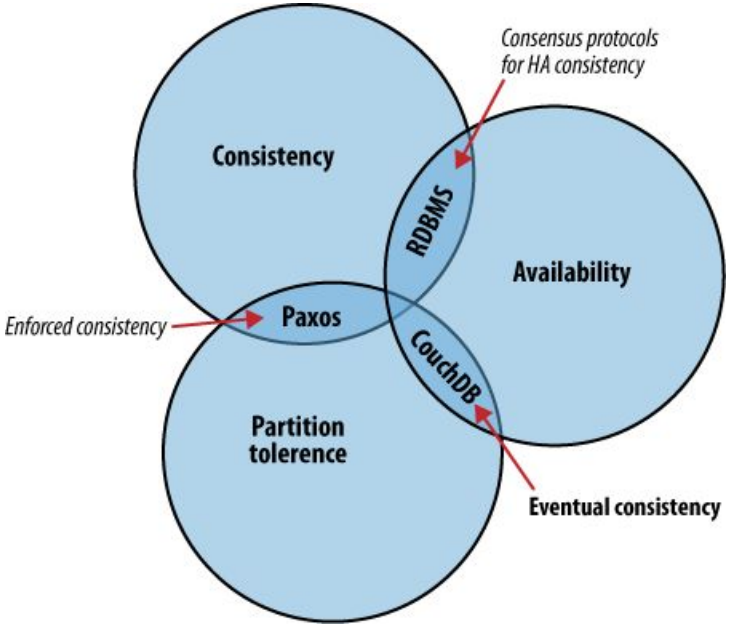
- *Consistency* (all nodes see the same data at the same time)
- *Availability* (a guarantee that every request receives a response about whether it succeeded or failed)
- *Partition tolerance* (the system continues to operate despite arbitrary message loss or failure of part of the system)

In theoretical computer science, the **CAP theorem**, also known as **Brewer's theorem**, states that it is impossible for a distributed computer system to **simultaneously** provide all three

ACID vs BASE



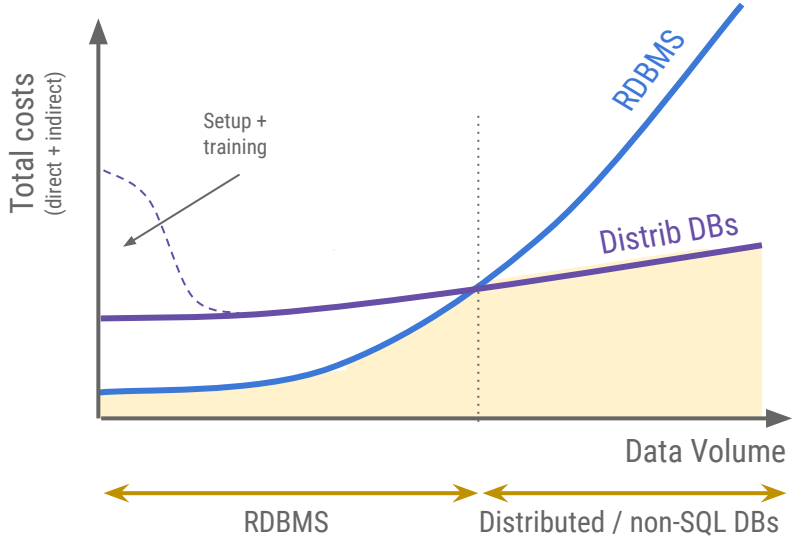




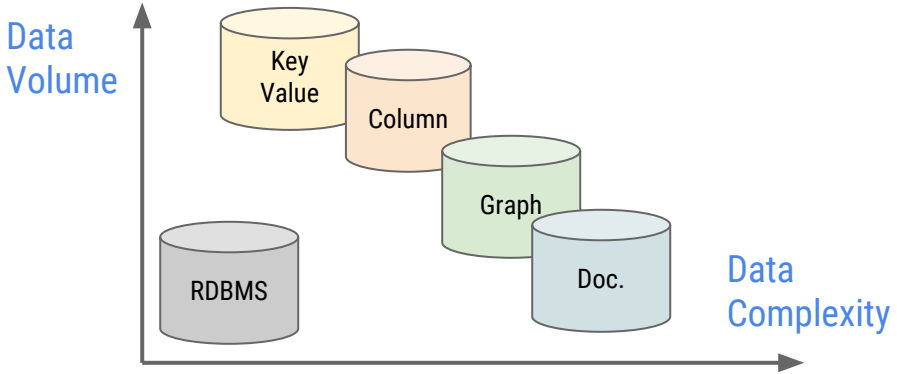
Paxos is a family of protocols for solving consensus in a network of unreliable processors. Consensus is the process of agreeing on one result among a group of participants.

Main characteristics

- Distributed databases become cheaper when the volume of data increases.
- Initial costs are determined by licences and implementation.



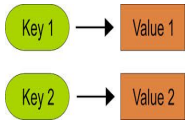
- Data complexity is given by relationship and subentities that the records have.
- Data volume is given by the amount and size of the records.



Categories

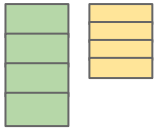
Unstructured
(not really a DB)

General file storage
Text files
Log files



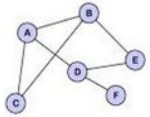
Key Value

Complex models
Flexible business logic
Semi-structured data
High volumes



Column

OLAP
Analytics
NOT FOR UPDATES



Graph

Relations between
entities (social graphs)

Document

Agile development
Flexible data-models
Too many types. Eg:
Corporate areas

amazon DynamoDB
amazon web services S3
hadoop

amazon DynamoDB
cassandra
APACHE HBASE
Google BigTable
redis

cloudera IMPALA
Google BigQuery

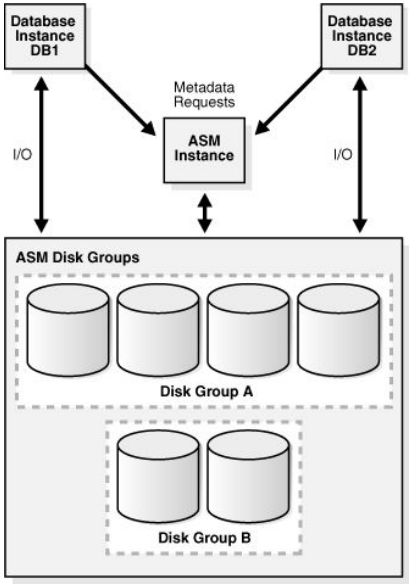
Neo4j
InfiniGraph

amazon DynamoDB
mongoDB
CouchDB
riak

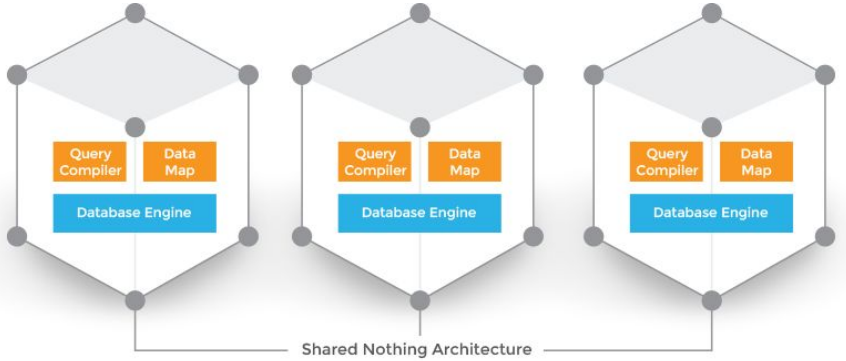
Main usage

	Unstructured (not really a DB)	Key Value	Column	Graph	Document
Summary	Stores large datasets as raw files. Its the base for other structured models. Provides scalability and High Availability.	Data as key-value pairs. Can store duplicate values No rigid structure	Store data as columns Slow write Fast read	Models information as entities and relations. Transactional.	Models data as Trees of semi-structured data. No rigid models
Good for	General file storage Text / Log files Custom structure Images Full text search	Entities Complex models Flexible business logic Semi-structured data High volumes	OLAP Analytics	Relations between entities	Agile development Flexible data-models Too many types. Eg: Corporate areas
Bad for	Structured data Fast queries Near real time data	Large inserts Consistency Transactions	Updated data	Non relationship data	Large datasets Fast queries
Sample usage	Log files	Large corporate data	Historical Marketing data	Social Graph	Project database

Shared Everything vs Shared Nothing Storage



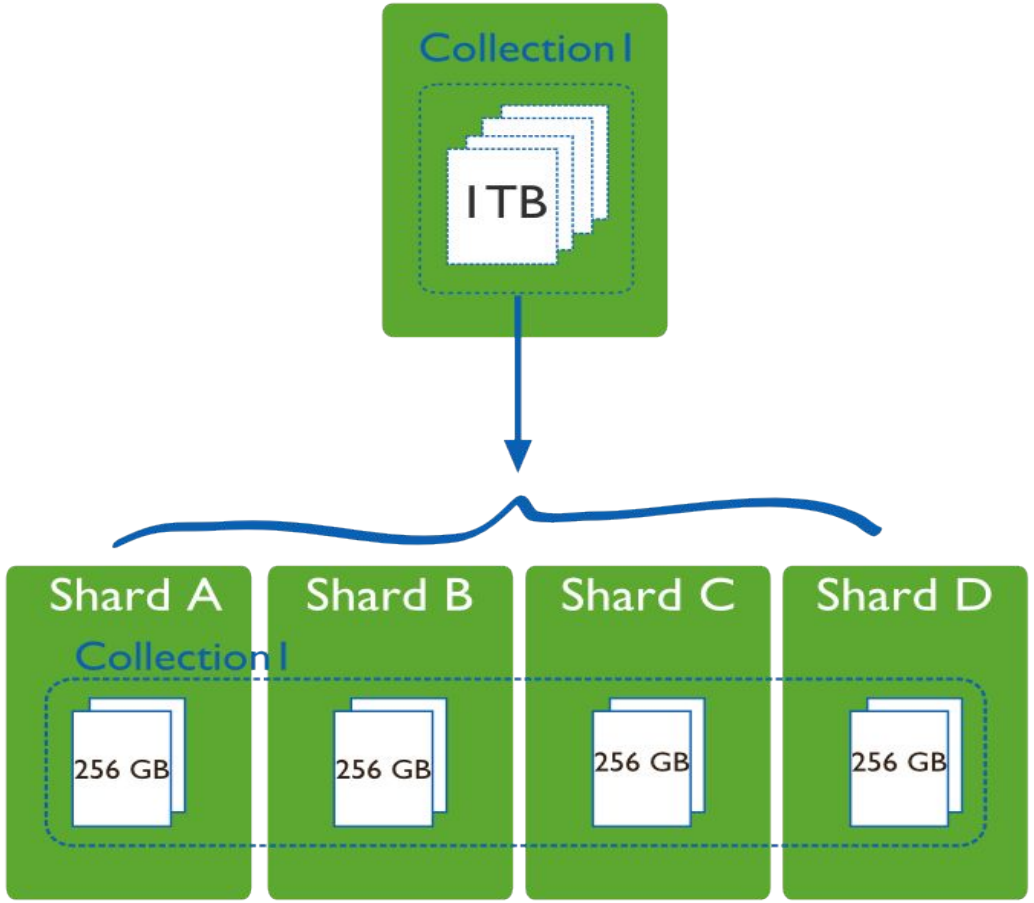
Traditional Oracle/Microsoft approach



Under the hood of NoSQLs database

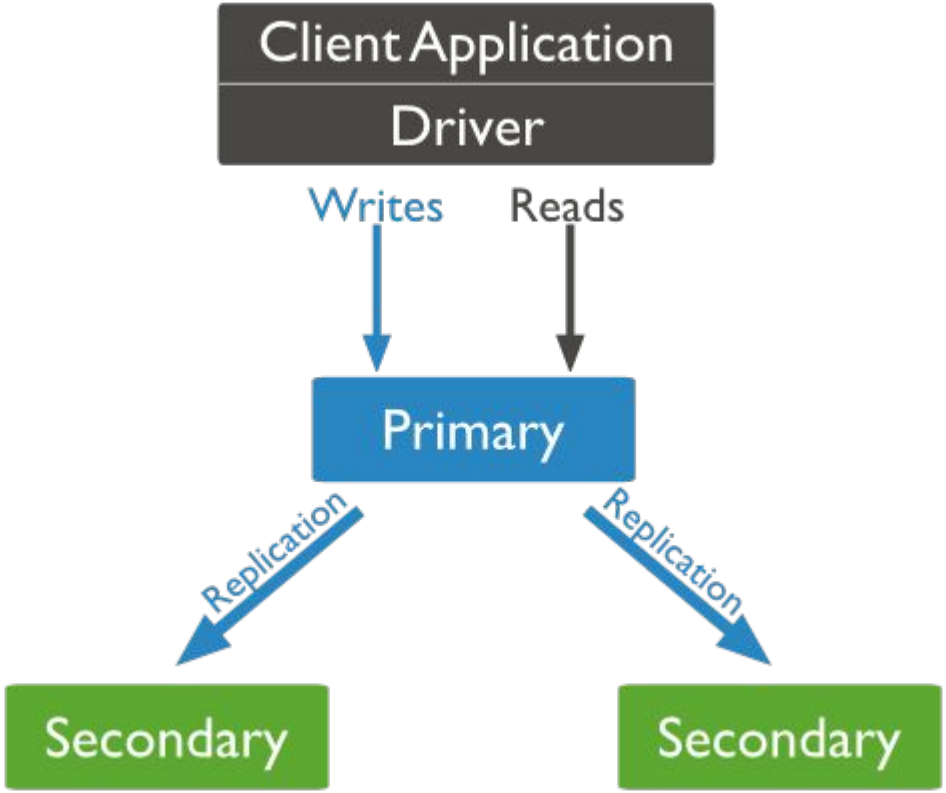
Sharding

- The concept of sharding is comparable to the RDBMs concept of partitioning.
- NoSQL approaches sharding since its conception and does not add additional latency to it's implementation. It's designed to work this way.



Replica Sets

- Replica sets are the active-active solution to high availability that NoSQL offers.
- MongoDB for example, allows one set to be marked as primary allowing writes and reads on it and the rest to be marked as secondary, only allowing reads.
- Data replication is done through an oplog that traces each operation done in the primary set to be passed forward to the replicas.
- oplog is comparable to Oracles REDO logs.



Database normalization

Database normalization is the process of organizing the attributes and tables of a relational database to minimize data redundancy.

Normalization involves refactoring a table into smaller (and less redundant) tables but without losing information; defining foreign keys in the old table referencing the primary keys of the new ones. The objective is to isolate data so that additions, deletions, and modifications of an attribute can be made in just one table and then propagated through the rest of the database using the defined foreign keys.

Database **DE**normalization

```
db.createCollection("mycollection")
```

=

USER INFO

KEY	First	Last	ZIP_id
1	Frank	Weigel	2
2	Ali	Bob	2
3	Mark	Azad	2
4	Steve	Yen	3

+

ADDRESS INFO

ZIP_id	City	State	ZIP
1	DEN	CO	30303
2	MV	CA	94040
3	CHI	IL	60609
4	NY	NY	10010

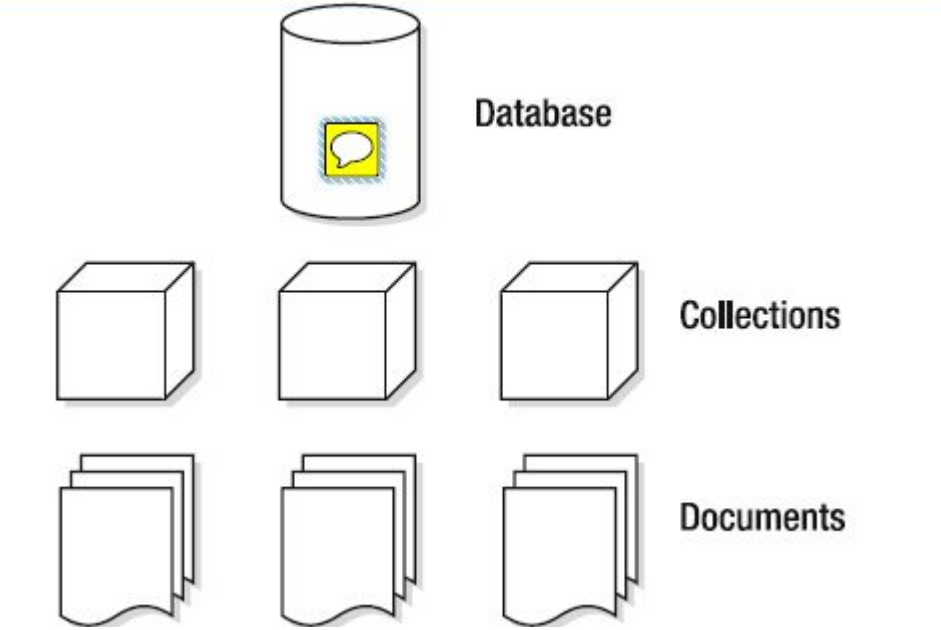
```
db.students.insert(...)
```

```
db.students.update(
  { _id: 1, first: "frank" },
  { $set: { "city" : "NY" } }
)
```

```
db.students.remove( {"_id" : 1 } )
```

Internal structure

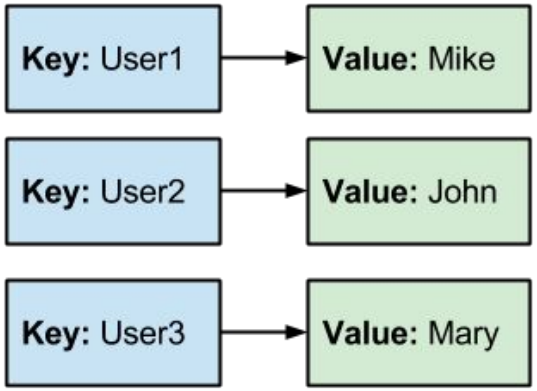
Document Namespace



Embedded Documents

```
{
  person: {
    first_name: "Peter",
    last_name: "Peterson",
    addresses: [
      {street: "123 Peter St"},
      {street: "504 Not Peter St"}
    ],
  }
}
```

Key-Value NoSQLs



Key-Value databases act as in-memory, stand alone/clusterable Map data structures.
They usually expose functionality to:

- Add pairs
- Get Values from keys
- Delete pairs with the key.

Add:

- TTL (Time to live)
- Appropriate protocol

You end up having a Memcached implementation.

Couchbase

redis

Documental NoSQLs with embedded Key/Value

```
{id: 1,
 name: Joe,
 url: '...',
 stream:
 [{
   user: 2,
   title: 'today',
   body: 'go fly a kite',
   likes: [
    {user: 3},
    {user: 1},
   ],
 }]
}
```

Primary Key

Nested Document within an Array

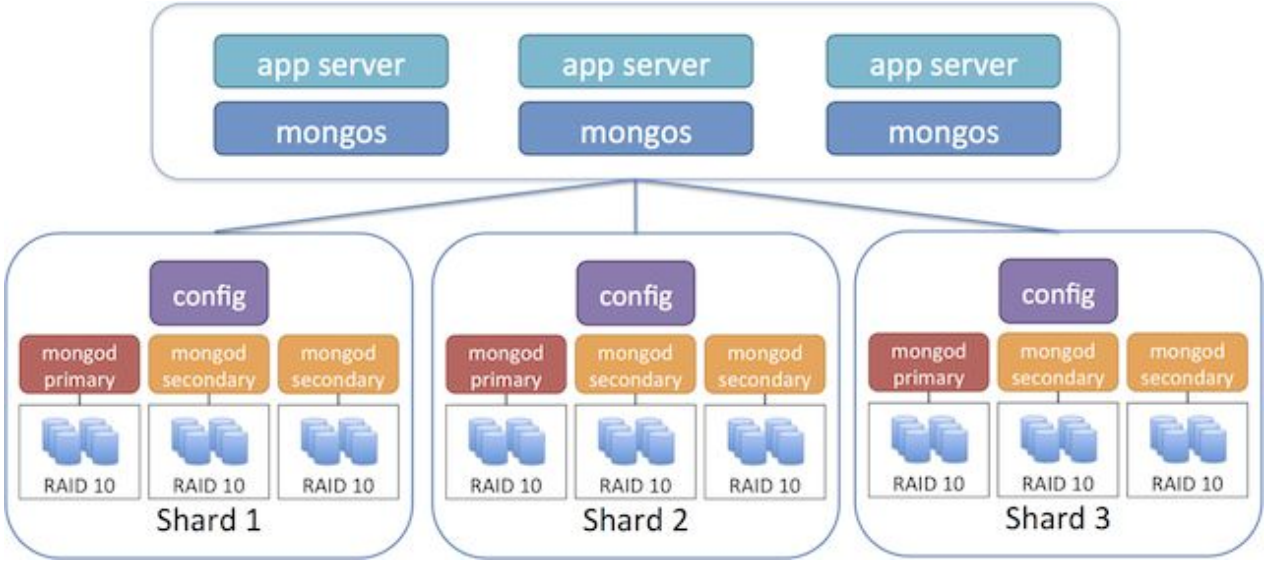
mongoDB

elasticsearch.

Couchbase

```
collection.get(1);
collection.find('name','Joe');
```

Distributed MongoDB Architecture



mongos for “MongoDB Shard,” is a routing service for MongoDB shard configurations that processes queries from the application layer, and determines the location of this data in the sharded cluster

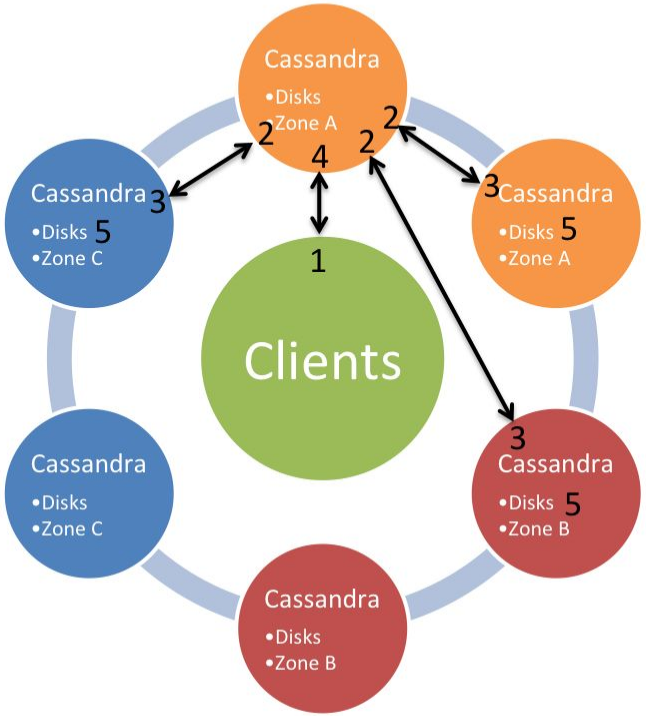
Tabular NoSQLs

Tabular databases include the following key properties:

- They share the same set of properties per record. Each column is usually assigned with metadata as its header title. If one of the rows lacks data for a specific column, a missing value that is related to that column's metadata will be stored in that cell.
- They access records through identifiers. Each table in a tabular database contains a particular set of related information that is connected with the database subject through key fields, which describe each record.

This type of database has a virtually infinite range for mass data storage.

Cassandra architecture

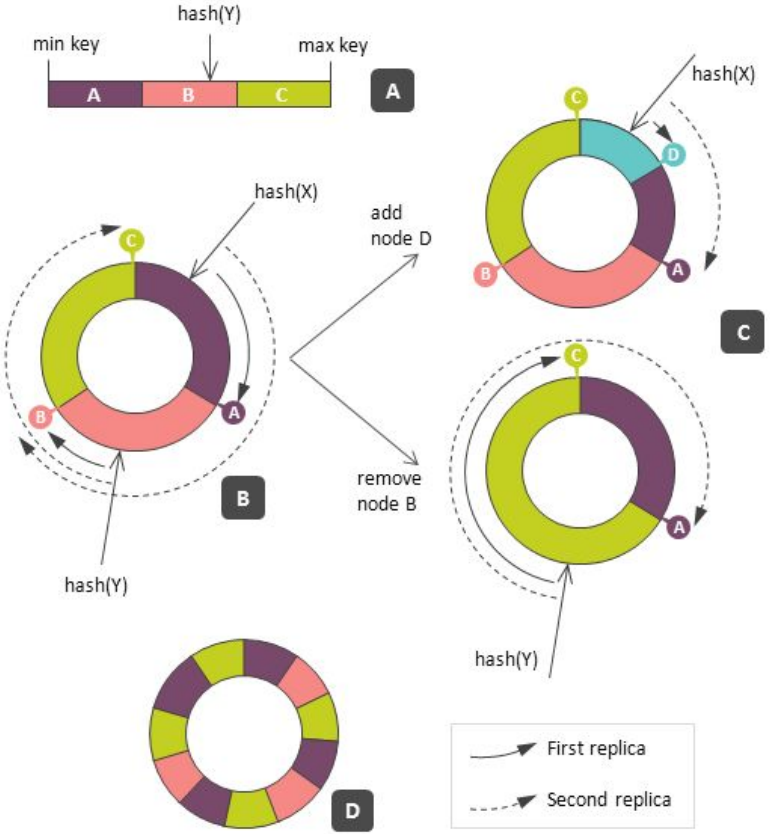


Cassandra

elasticsearch.

Numbers show a strictly consistent write/read.

Cassandra's Hash Ring



Cassandra

Virtual Nodes