# CSCI235 – Database Systems

## NoSQL Database Systems

Sionggo Japit

sjapit@uow.edu.au

3 October 2021

## Acknowledgements

The following presentation were adapted from the lecture slides of:

CSCI235 – Database Systems,

15MongoDBDatabasesCollectionsDocuments

By Dr Janusz R. Getta,

School of Computing and Information Technology

University of Wollongong, Australia

## Outline

Properties of NoSQL database systems

What is in the name “NoSQL”?

NoSQL database system – What is it?

Why NoSQL database systems?

# NoSQL database system – What is it?

- NoSQL database systems were developed in early 2000 in response to demands for processing of vast amounts of data produced by increasing Internet usage and mobile geo-location technologies.
- Traditional solutions were either too expensive, not scalable, or required too much time to process data.
- Modern NoSQL database systems borrowed some solutions from the earlier systems and made significant advances in scalability and efficient processing of diverse types of data such as text audio, video, image, and geo-location.
- NoSQL database systems include the following implementation types: key-value stores, document stores, graph stores, column stores, in-memory databases.

# Outline

Properties of NoSQL database systems

NoSQL database system – What is it?

What is in the name "NoSQL"?

Why NoSQL database systems?

## What is in the name "NoSQL"?

- A term "NoSQL" has been used for the first time in late 1990 by C. Strozzi as a name of open-source relational database system that did not use SQL as a query language.
- The usage of "NoSQL" as it is recognized today has been used by J. Oskarsson in 2009 for the projects experimenting with alternate data storage, like BigTable (Google) and Dynamo (Amazon).
- "NoSQL" database systems do not use SQL run on the clusters of computers and provide different options for consistency and distribution.
- Despite its confrontational nature some people say that "NoSQL" means "Not Only SQL", but then it should be written as "NOSQL" and … it is not.

## Outline

Properties of NoSQL database systems

Why NoSQL database systems?

What is in the name “NoSQL”?

NoSQL database system – What is it?

## Why NoSQL database systems?

- Impedance mismatch problem: the difference between persistent data structures (relational model) and the transient data structures (object-oriented programming language).
- Application and integration databases: a structure that integrates many applications is more complex than single application requires, e.g., index required by one application may cause performance problems for another application.
- Cluster of small machines: relational database that operate on shared disk subsystems are not designed to work on clusters.
- Distribution and consistency: ACID transaction protocol is too strict for distributed transactions.

# Outline

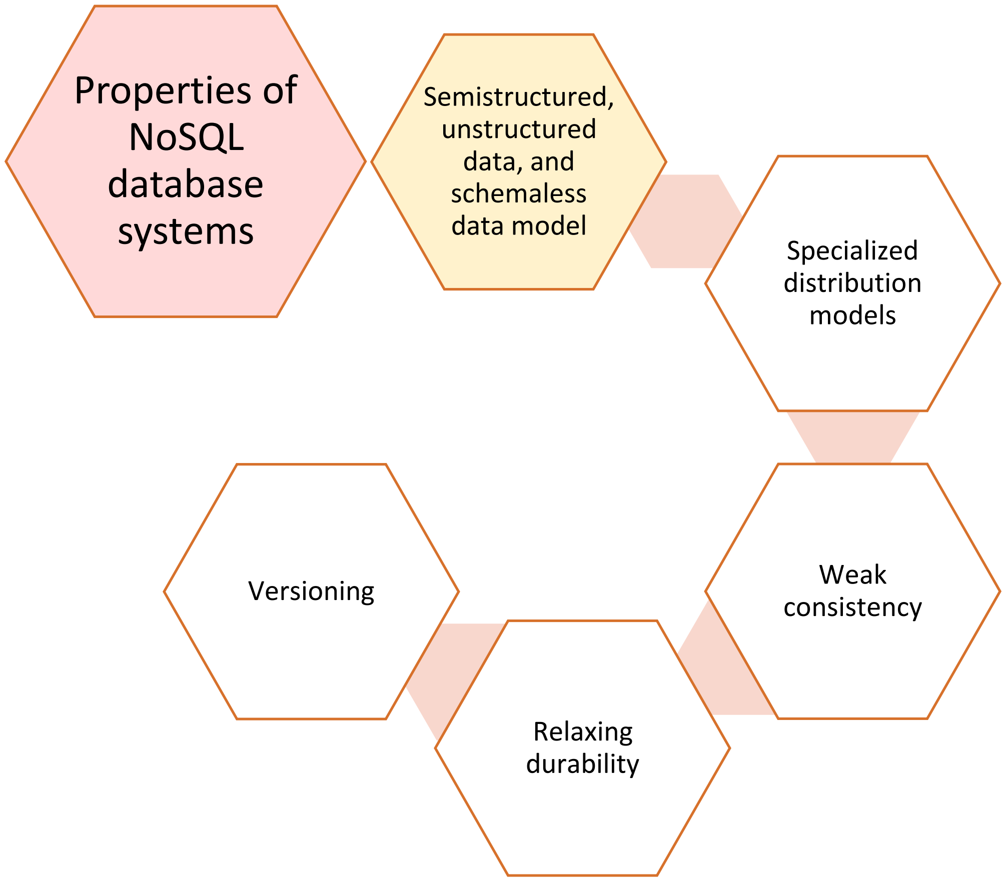
Properties of NoSQL database systems

Why NoSQL database systems?

What is in the name "NoSQL"?

NoSQL database system – What is it?

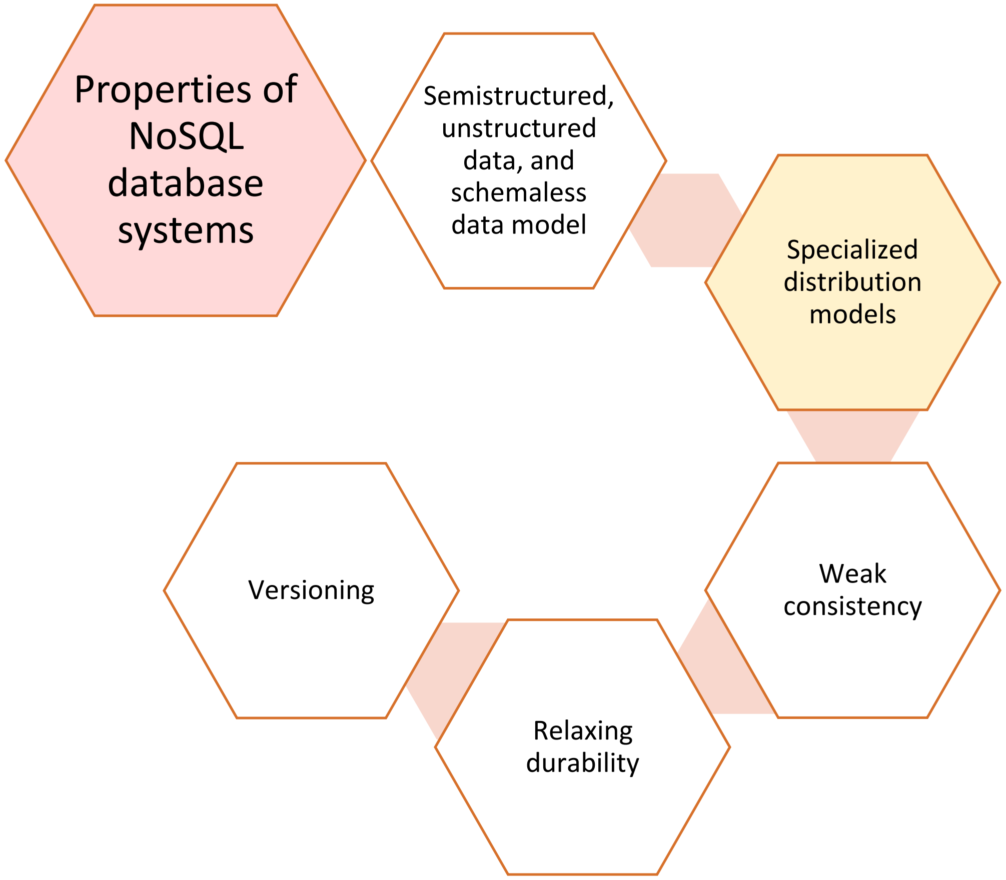
# Outline


Properties of NoSQL database systems
Semistructured, unstructured data, and schemaless data model
Specialized distribution models
Weak consistency
Relaxing durability
Versioning

## Semistructured, schemaless data model

- Semistructured data is a form of structured data that does not conform to the formal structure of data models associated with relational databases or other forms of data tables.
- Semistructured data contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data.
- Schemaless data model means that no particular data structure used to store data in a database.
- Schemaless database does not require consistency with a rigid schema, e.g., database schema, relational schema, data type, table, etc.
- Schemaless database does not enforce data type limitations on individual values.
- Schemaless database can store structured and unstructured data.

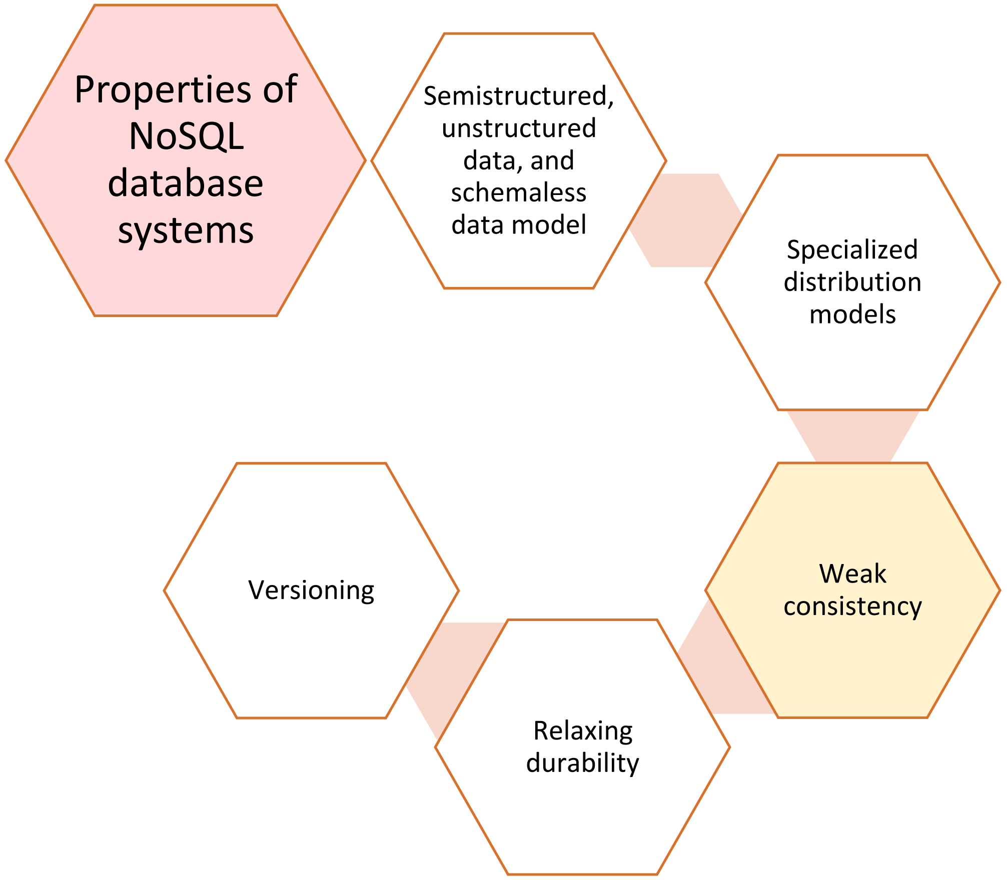
# Outline


Properties of NoSQL database systems
Semistructured, unstructured data, and schemaless data model
Specialized distribution models
Weak consistency
Relaxing durability
Versioning

## Specialized distribution models

- Single server: means no distribution at all.
- Sharding: to support horizontal scalability we put different parts of data onto different servers (shards).
- Master-slave replication: data are replicated across multiple nodes and one node is designated as master node, the others are slaves; all updates are made to the master and later on propagated to slaves.
- Peer-to-peer replication: data replicated across multiple nodes; all replicas have the same weight, no master mode; nodes communicate their writes; all nodes read and write all data.
- Combining sharding and replication: use both master-slave application and sharding.

# Outline


Properties of NoSQL database systems
Semistructured, unstructured data, and schemaless data model
Specialized distribution models
Weak consistency
Relaxing durability
Versioning

## Weak consistency

- A typical read consistency principle where update is performed over two or more data items blocks access to all data items affected by the update, e.g., 2PL protocol.
- NoSQL database systems relax the transactional consistency to some extent.
- Data items can be left inconsistent over certain period of time called as inconsistency window.
- A concept of eventual consistency is used to enforce replication consistency over distributed and replicated data items.
- Eventual consistency means that the copies of data items can be inconsistent in inconsistency window and all copies will have the same value later on.

## Weak consistency

- It is possible tolerate long inconsistency window under a condition that read-your-write consistency is enforced.
- The leads to a concept of session consistency, which means that within a user session read-your-write consistency is enforced.
- To provide session consistency it is possible to implement sticky session, i.e. a session that is attached to only one node in a cluster (it is also called as session affinity).
- Another solution to provide session consistency is to use version timestamps where every interaction with a data item is performed on a data item with the highest timestamp.

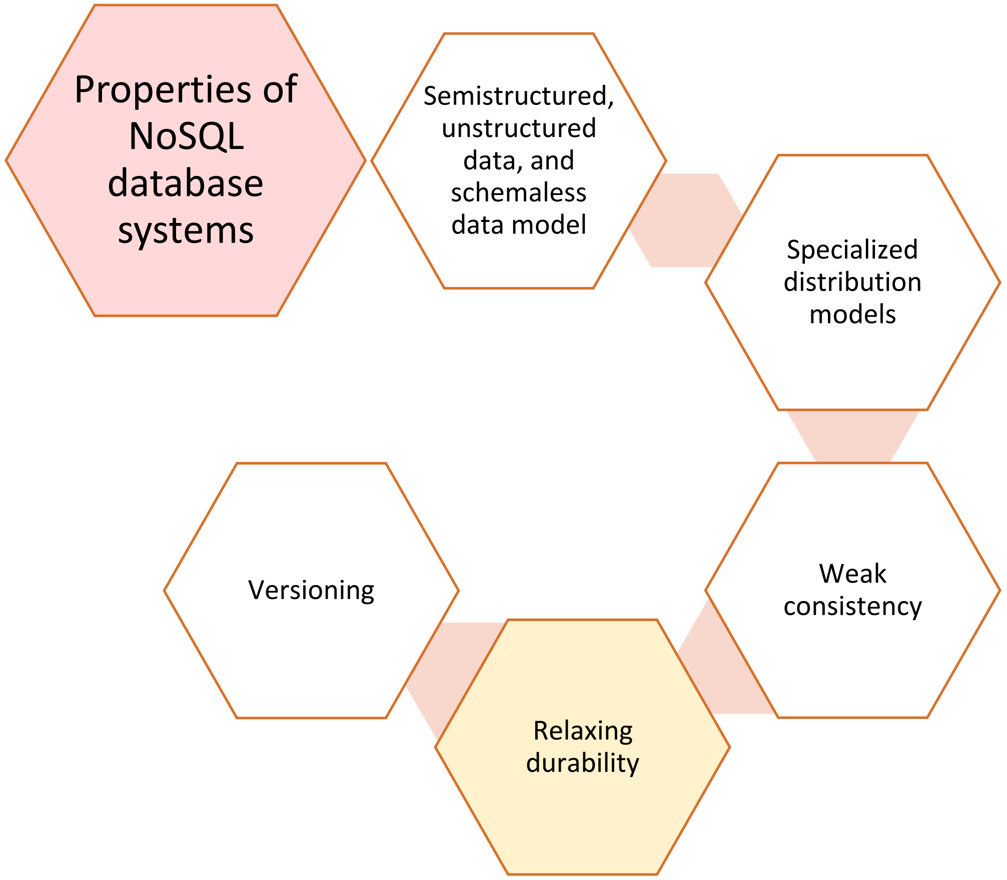
## Weak consistency

- It is always possible to design a system that avoids inconsistencies but sometimes we have to trade consistency for some other characteristics of a system.
- Different domains have different tolerances for inconsistencies and we have to take this tolerance into account as make our decisions.
- Even in traditional relational database systems it is possible to relax consistency from the highest isolation level (serializable) to the lowest levels to get better performance.
- CAP theorem: given three properties of Consistency, Availability, and Partition tolerance, you can get only two.

## Weak consistency

- Consistency: A state of a database satisfies the given consistency constraints at any moment in time.
- Availability: If you can talk to a node in a cluster then it can read and write data.
- Partition: Cluster can survive communication breakages that separate the cluster into multiple partitions unable to communicate with each other (split brain).
- A single server system is CA because it has Consistency and Availability and not Partition tolerance.
- If clusters must be tolerant of network partitions, then we have to trade consistency for availability.

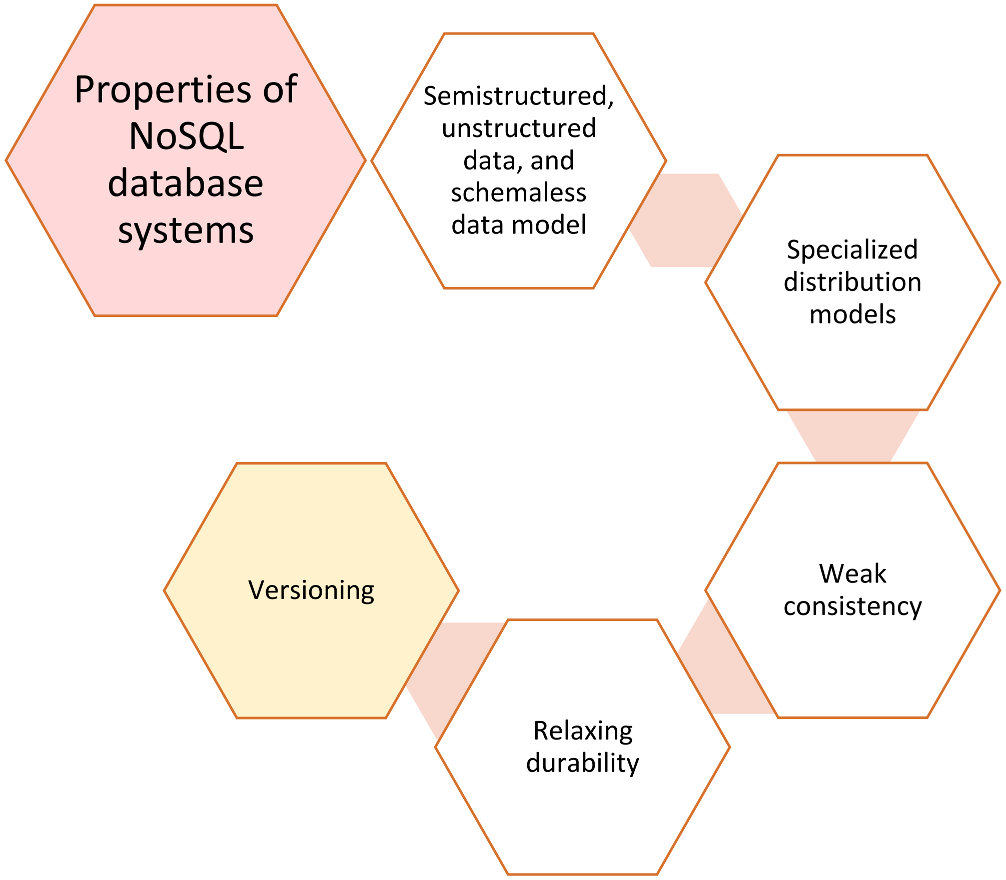
# Outline


Properties of NoSQL database systems
Semistructured, unstructured data, and schemaless data model
Specialized distribution models
Weak consistency
Relaxing durability
Versioning

# Relaxing durability

- If SQL systems follow ACID properties then NoSQL systems follow BASE properties: Basically Available, Soft state, Eventually consistent
- Relaxing durability means that we trade of durability for higher performance, e.g., apply updates to in-memory representation of a database and periodically flush changes to disk.
- Another class of durability tradeoffs comes with replicated data, e.g., when a node processes and update but fails before that updates is replicated to other nodes.

# Outline


Properties of NoSQL database systems
Semistructured, unstructured data, and schemaless data model
Specialized distribution models
Weak consistency
Relaxing durability
Versioning

## Versioning

- A version is a particular form of something differing in certain respects from an earlier form or other forms of the same type of thing.
- Versioning is a database systems means that all modifications of data items are stored in a database together with timestamps when such modifications occurred.
- In practice versioning is performed to a predefined depth, i.e. a total number of versions of data item is determined when a data item is created.
- Versioning allows for representation of historical information.
- Numbering of data versions through timestamps allows to track when a data item has changed and if a new version is available allows to determine specifically which version is the most current one.

## References

- Harrison G., Next Generation Databases NoSQL Big Data, Apress, 2015