INFO20003 Database Systems

Xiuge Chen

Tutorial 12
2020.06.01

Agenda

1. Understand the concepts of NoSQL databases - 15min
2. Choosing appropriate NoSQL database types for scenarios - 10min
3. CAP theorem with respect to NoSQL databases - 10min
4. Lab/Revision - 50 min

Understand the concepts of NoSQL databases

1. NoSQL database

- non-relational database
- Help store and retrieve data in formats **other than tabular form**
- Not depend on any particular structure such as tables, rows, columns or schemas to organize data

Understand the concepts of NoSQL databases

1. NoSQL database

- Why need it?

- **Intensive but flexible** data analysis using distributed systems, cloud computing and high-performance computing (HPC)

- Traditional relational databases are unable to meet *performance*, *scalability* and *flexibility* requirements.

- Examples: chat data, messaging, large objects such as videos and images and many types of business documents.

Understand the concepts of NoSQL databases

1. Type of NoSQL database

a. **Graph databases:**

- Based on **graph theory** and utilize the concept of a *graph* to store, connect and query data.

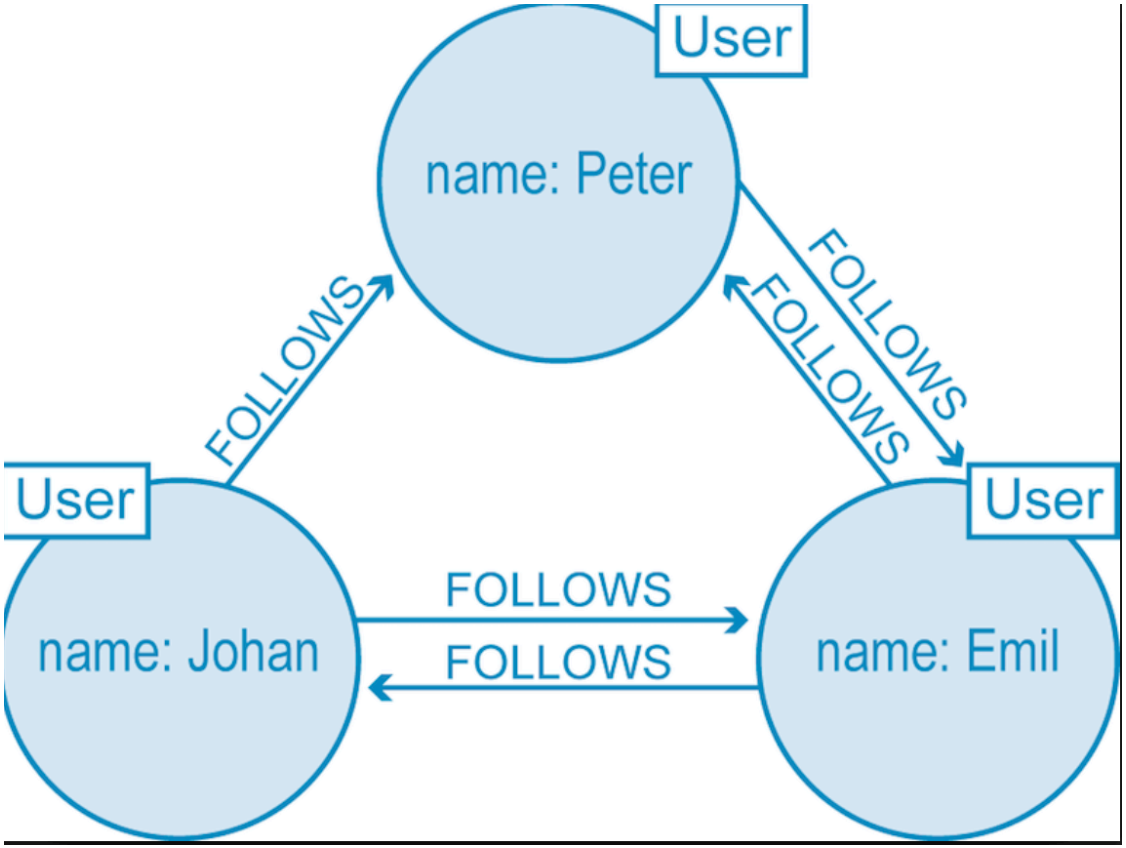
- ***Nodes****:* rows or records in the relational database and represent entities such as accounts, people, items etc.
- **Edges:** connect nodes and resemble the relational relationship between tables. Both nodes and edges can have properties associated with them.

- Examples: *neo4j*, used by Airbnb, Microsoft, IBM, eBay and Walmart.

Understand the concepts of NoSQL databases

1. Type of NoSQL database

a. **Graph databases:**



https://www.google.com/search?q=graph+database&rlz=1C5CHFA_enAU783AU812&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi7hIa_0rviAhUF_XMBHaYZAQoQ_AUIDigB&biw=1440&bih=716#imgrc=MKmkZXUR5NIEDM:

Understand the concepts of NoSQL databases

1. Type of NoSQL database

a. **Graph databases:**

https://www.google.com/search?q=graph+database&rlz=1C5CHFA_enAU783AU812&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi7hIa_0rviAhUF_XMBHaYZAQoQ_AUIDigB&biw=1440&bih=716#imgrc=ydEvFNq3bjBTQM:

Understand the concepts of NoSQL databases

1. Type of NoSQL database

b. **Key-value stores**:

- Most **flexible** NoSQL databases
- **Least structured**

- Use a simple **key-value structure** to organize data
- No schema and the data values can be of any data type
- ***Key***: unique identifier to allow retrieval of the associated value
- ***Value***: anything (images, text, videos, binary data, lists, JSON)

- Examples: Berkeley DB, Aerospike and Redis. Key-value stores are highly flexible and support massive scalability

Understand the concepts of NoSQL databases

1. Type of NoSQL database

b. **Key-value stores**:

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

https://www.google.com/search?q=key+value+store&rlz=1C5CHFA_enAU783AU812&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjEyrSg07viAhUkjOYKHYknCwgQ_AUIDigB&biw=1440&bih=716#imgrc=21g0eFT_9EdOmM:

Understand the concepts of NoSQL databases

1. Type of NoSQL database

c. **Column-family stores:**

- Wide-column stores or extensible record stores
- A type of **key-value** database

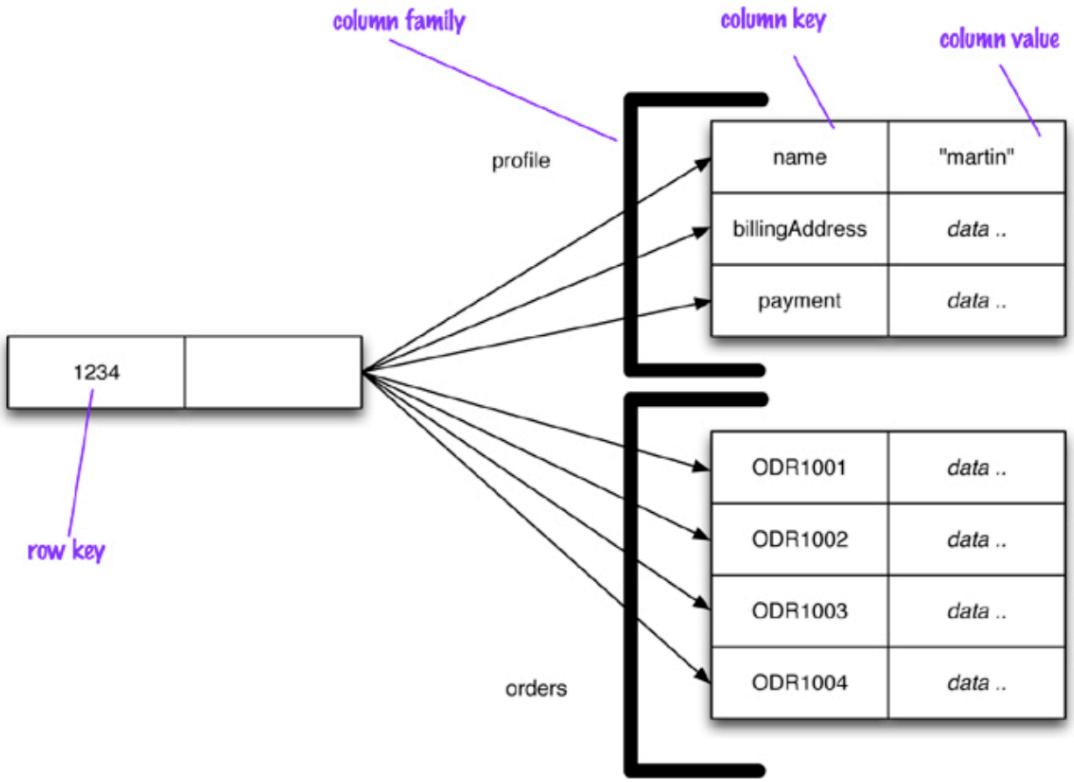
- Use tables, rows and columns but for each record the column names, their format and record keys can greatly vary
- **Semi-structured** data
- ***Columns***: created for each row instead of being predefined

- Examples: Cassandra and Hbase used by Facebook in the past to handle messaging and inbox search. Other enterprises using wide column stores are Netflix, Twitter and Reddit

Understand the concepts of NoSQL databases

1. Type of NoSQL database

c. Column-family stores:

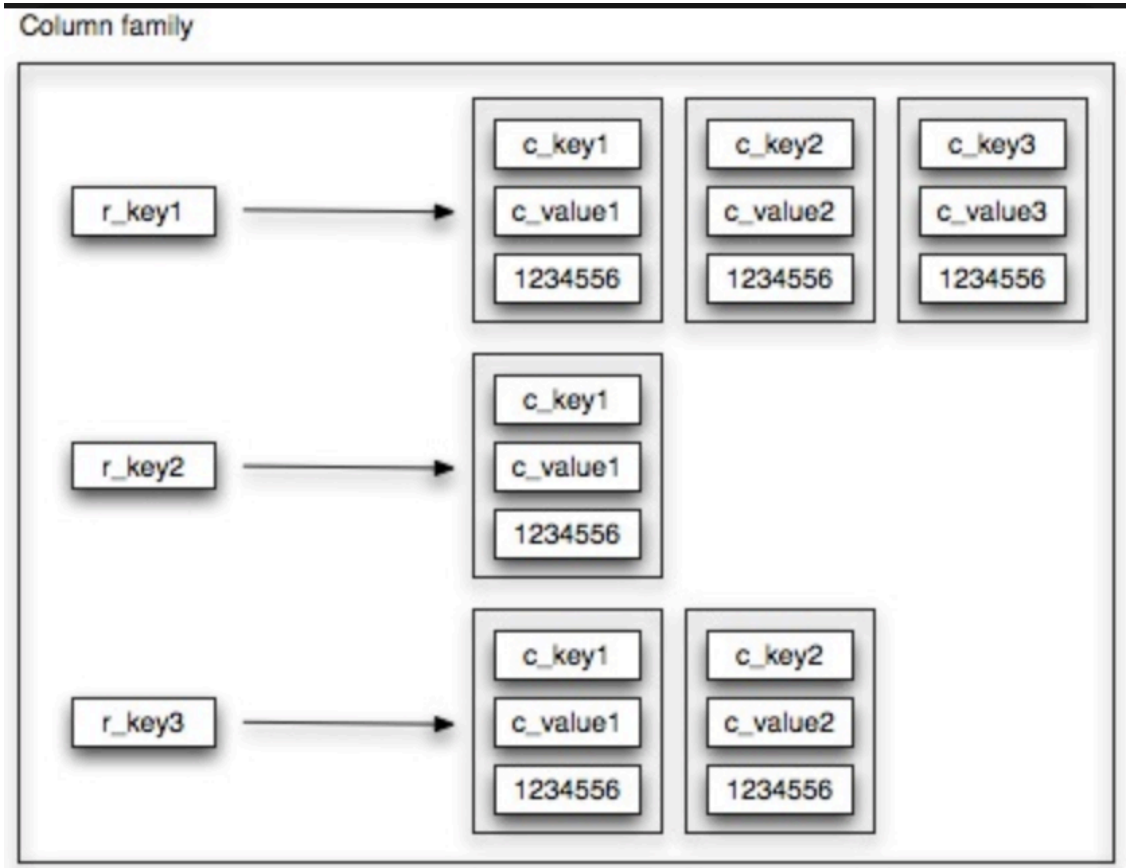


https://www.google.com/search?q=column+family+store&rlz=1C5CHFA_enAU783AU812&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiIg4i707viAhXx8XMBHWuCAY8Q_AUIDigB&biw=1440&bih=716#imgrc=vktlPkW8D7CQeM:

Understand the concepts of NoSQL databases

1. Type of NoSQL database

c. **Column-family stores:**



https://www.google.com/search?q=column+family+store&rlz=1C5CHFA_enAU783AU812&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiIg4i707viAhXx8XMBHWuCAY8Q_AUIDigB&biw=1440&bih=716#imgrc=hR5i4LaD5DW0SM:

Understand the concepts of NoSQL databases

1. Type of NoSQL database

d. **Document stores**

- Store data in JSON, XML or BSON documents (where JSON is by far the most dominant).
- Resulting documents are independent components which can be distributed more easily
- Each document can have its own structure and schema
- Still possible to create indexes within documents
- Examples: MongoDB

Understand the concepts of NoSQL databases

1. Type of NoSQL database

d. **Document stores**

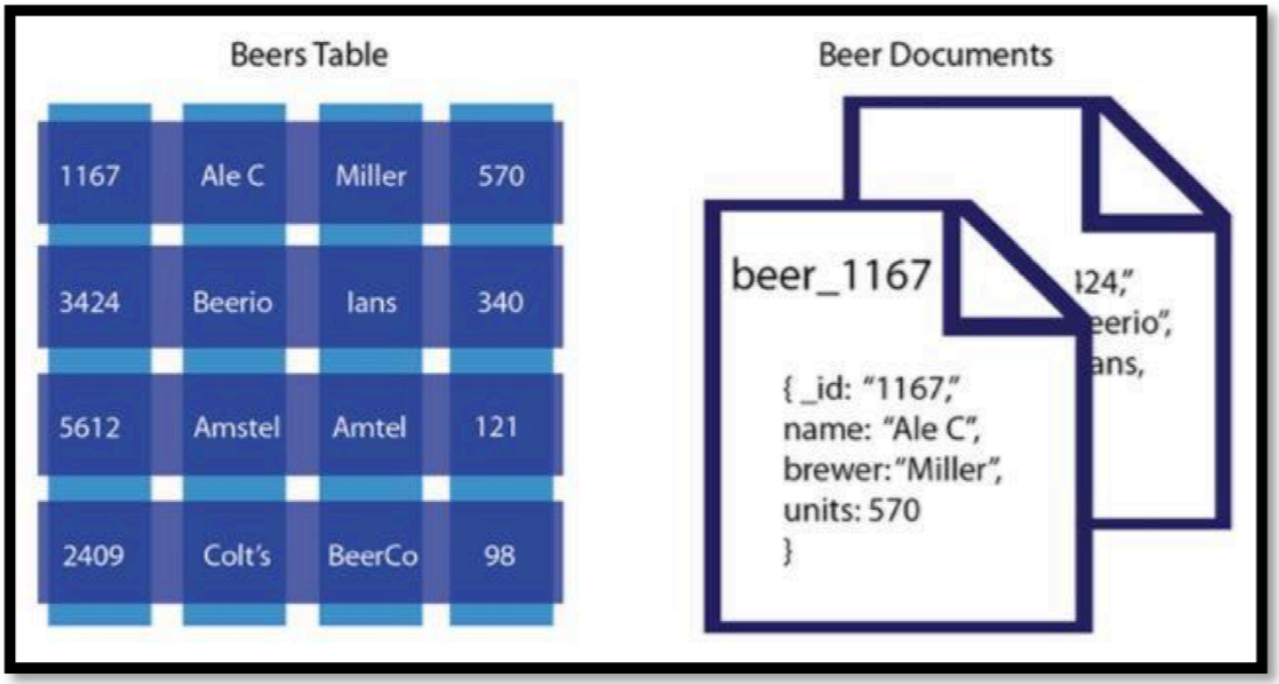
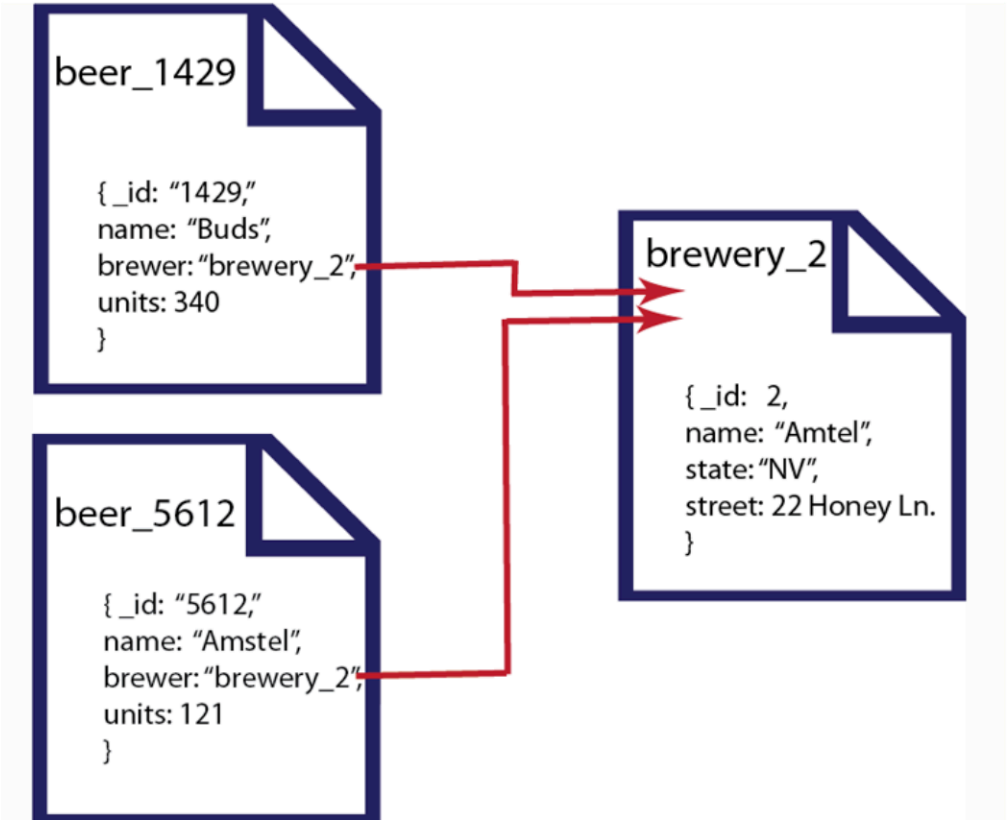


Figure S1: A comparison between a relational Beers table and a document store containing the same data. Adapted from *https://developer.couchbase.com/documentation/server/3.x/developer/dev-guide-3.0/compare-docs-vs-relational.html*

Understand the concepts of NoSQL databases

1. Type of NoSQL database

d. **Document stores**



https://www.google.com/search?q=document+store&rlz=1C5CHFA_enAU783AU812&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjZgsb807viAhURU30KHZHRCskQ_AUIDigB&biw=1440&bih=716&dpr=2#imgrc=rFDJBRFS9hyDGM:

Understand the concepts of NoSQL databases

Any questions?

1. Choosing a NoSQL database

Libraries store information about their collections in their catalogue.

Match each of the following statements to the type of NoSQL database that would be best for storing that library's data. Select from the four types of NoSQL database discussed previously.

1. Choosing a NoSQL database

Libraries store information about their collections in their catalogue.

Match each of the following statements to the type of NoSQL database that would be best for storing that library's data. Select from the four types of NoSQL database discussed previously.

a. In one library, items are catalogued by author, title and publisher, as well as any number of other fields chosen by the cataloguer, such as physical description, subject codes and notes.

A *column-family database* would be the best choice. Each row in a column-family table may have a different set of columns associated with it.

1. Choosing a NoSQL database

Libraries store information about their collections in their catalogue.

Match each of the following statements to the type of NoSQL database that would be best for storing that library's data. Select from the four types of NoSQL database discussed previously.

b. In another library, each catalogue record is stored in the MARC format (Figure 1), a coded text format that contains all the catalogue information for a particular item.

1. Choosing a NoSQL database

```
LEADER 00000nam  22000001  4500
008    730220s1955    ilu      b    00000 eng
019    55007351
050 0  QA276.5|b.R3
082    311.22
110 20 Rand Corporation.
245 12 A million random digits|bwith 100,000 normal deviates.
260 0  Glencoe, Ill.,|bFree Press|c[1955]
300    xxv, 400, 200 p.|c28 cm.
504    Bibliography: p. xxiv-xxv.
650  0 Numbers, Random.
984    |cMS T 519 R152
```

Figure 1: An example of a MARC record. MARC is a very old format that predates NoSQL, JSON and even XML by several decades, yet it remains the industry standard in library data systems.

1. Choosing a NoSQL database

Libraries store information about their collections in their catalogue.

Match each of the following statements to the type of NoSQL database that would be best for storing that library's data. Select from the four types of NoSQL database discussed previously.

b. In another library, each catalogue record is stored in the MARC format (Figure 1), a coded text format that contains all the catalogue information for a particular item.

A *document store* would be best suited to this task. Normally, document stores use a modern data interchange format such as JSON or XML, but industry-specific structured data formats such as MARC can be used with specialised document store systems.

1. Choosing a NoSQL database

Libraries store information about their collections in their catalogue.

Match each of the following statements to the type of NoSQL database that would be best for storing that library's data. Select from the four types of NoSQL database discussed previously.

c. A public library wishes to store cover photos of all its items, which might be in JPEG, PNG or PDF format, or stored as a URL.

Key-value stores can store any kind of data. Each document in a document store should be made up of structured data – images are not structured data in the same way as JSON, so a document store is a poor choice.

1. Choosing a NoSQL database

Libraries store information about their collections in their catalogue.

Match each of the following statements to the type of NoSQL database that would be best for storing that library's data. Select from the four types of NoSQL database discussed previously.

d. A university library wishes to keep track of which published academic papers reference each other in order to help researchers measure their metrics.

By storing papers as nodes and references as edges joining the nodes, a *graph database* can efficiently capture, and answer complex queries about, the relationships between papers.

Any questions?

CAP theorem with respect to NoSQL databases

1. Advantages of NoSQL

a. Flexible modelling

Flexible data models
More suited to coping with less structured data sources.

b. Scalability

Capacity in a NoSQL database can be added and removed quickly using a **horizontal scale-out methodology** (adding inexpensive servers and connecting them to a database cluster).

As a result, the cost and complexity associated with scaling up a relational database into a distributed database are avoided.

1. Advantages of NoSQL

c. Performance

Horizontal scale-out methodology -> manage efficient reads, writes and storage of the data items when handling big data.

LinkedIn, Facebook and Google deploy data centres in different parts of the world and partition their users so that all of their users experience the fewest possible hops by being routed to the closest data centre.

d. High availability

Constant availability (24/7) is a challenge for relational databases, since they are physically implemented on a single server or on a cluster with a shared storage.
NoSQL databases are typically stored in **partitions** and they divide data across multiple database instances without any shared resources.

2. The CAP theorem

a. **Consistency**

All the servers hosting the database will have the **same** data, so that anyone accessing the data will get the same copy regardless of which server is answering the query.
(This is a different thing to “consistency” in the context of the ACID principles, which refers to data integrity.)

b. **Availability**

The system will always respond to a request even if it is not the latest data or consistent across the system.

2. The CAP theorem

c. Partition tolerance

A “partition” in the context of the CAP theorem refers to a disruption in network access so that some servers are unable to access other servers.

The system continues to operate **as a whole** even if individual servers fail or can’t be reached.

2. The CAP theorem

What is it used for?

The CAP theorem states that, at any given point in time, a system can achieve only *two* out of three principles, while it is theoretically impossible to achieve three at the same time.

In case of NoSQL databases, the choice is between AP or CP, as the biggest advantage of NoSQL databases is partition tolerance when compared to relational DBMSs:

2. The CAP theorem

a. **AP**

The database always answers, but possibly with outdated or wrong data, hence ensuring *availability* instead of consistency.

On systems that allow reads before updating all the nodes, high availability is achieved. Such systems eventually achieve consistency as well.

For example, Google and Facebook enforce eventual consistency such that different servers might have inconsistent views depending on how many servers are updated at a given time.

2. The CAP theorem

b. **CP:**

The database stops all the operations until the latest copy of data is available on all nodes.

On systems that lock all the nodes before allowing reads, high consistency is achieved. Such systems become available after the consistency is achieved.

Most NoSQL databases choose AP over CP to ensure continuous availability and eventual consistency.

Any questions?

Final Exam

Thank you

Good luck on exams!