
DATABASE DESIGN AND MANAGEMENT(IT 2140)

LECTURE 09 – INTRODUCTION TO NON- RELATIONAL DATABASES



LECTURE CONTENT

- Introduction to Non-Relational Databases
- Why NoSQL?
- NoSQL Databases
- Who uses NoSQL ?
- RDBMS vs NoSQL
- CAP Theorem
- BASE Concept
- Types of Non-Relational Databases
- Choosing NoSQL databases
- Current Trends in NoSQL Databases
- Challenges in Non-Relational Data Modelling

LEARNING OUTCOMES

- At the end of this lecture students should be able to
 - Differentiate between relational and non-relational data models.
 - Explain the need for NoSQL and CAP concepts.
 - Identify and describe key types of non-relational databases.
 - Discuss current trends and challenges in data modeling for NoSQL systems.

INTRODUCTION TO NON-RELATIONAL DATABASES

- Relational databases use tables, rows, and columns; perfect for structured data.
- Modern systems need to handle massive, fast, and varied data (Big Data, IoT, social media).
- Non-Relational Databases (NoSQL) emerged to handle:
 - High scalability & availability
 - Flexible schema for unstructured data
 - Distributed storage across clusters
- Examples: MongoDB, Cassandra, Redis, Neo4j

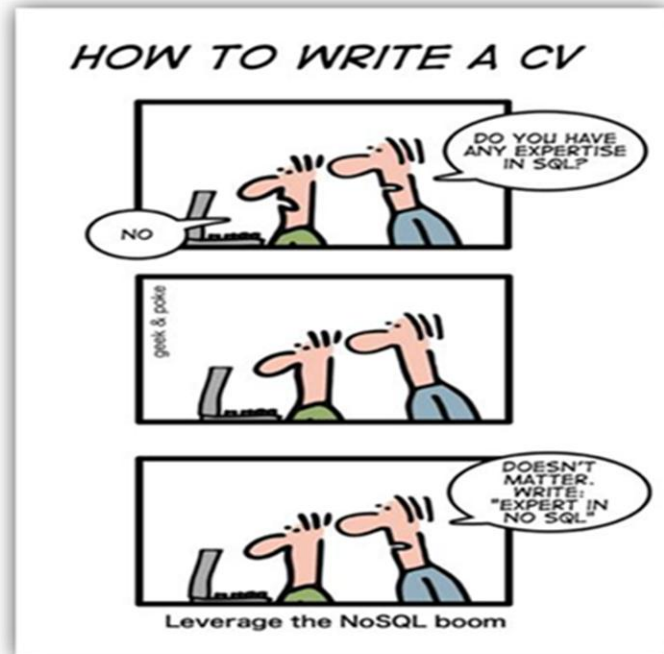
WHY NOSQL?

- Traditional RDBMS struggles with:
 - Large, distributed datasets
 - Real-time and high-velocity data
 - Dynamic or unstructured data models
- Web companies (Google, Amazon, Facebook) built internal scalable systems (e.g., BigTable, DynamoDB).
- NoSQL provides horizontal scaling, replication, and eventual consistency

NOSQL DATABASES

“NoSQL is an accidental term with no precise definition”

- **first used** at an informal meetup in **2009** in San Francisco (presentations from Voldemort, Cassandra, Dynamite, HBase, Hypertable, CouchDB, and MongoDB)



[Sadalage & Fowler: NoSQL Distilled, 2012]

NOSQL DATABASES

- NoSQL: Database technologies that are (mostly):
 - **Not using** the **relational** model (nor the SQL language)
 - Designed to run on **large clusters** (horizontally scalable)
 - **No schema** - fields can be freely added to any record
 - Open source
 - Based on the needs of 21st century web estates

[Sadalage & Fowler: NoSQL Distilled, 2012]

- Other characteristics (often true):
 - easy **replication** support (fault-tolerance, query efficiency)
 - **simple** API
 - **eventually** consistent (not ACID)

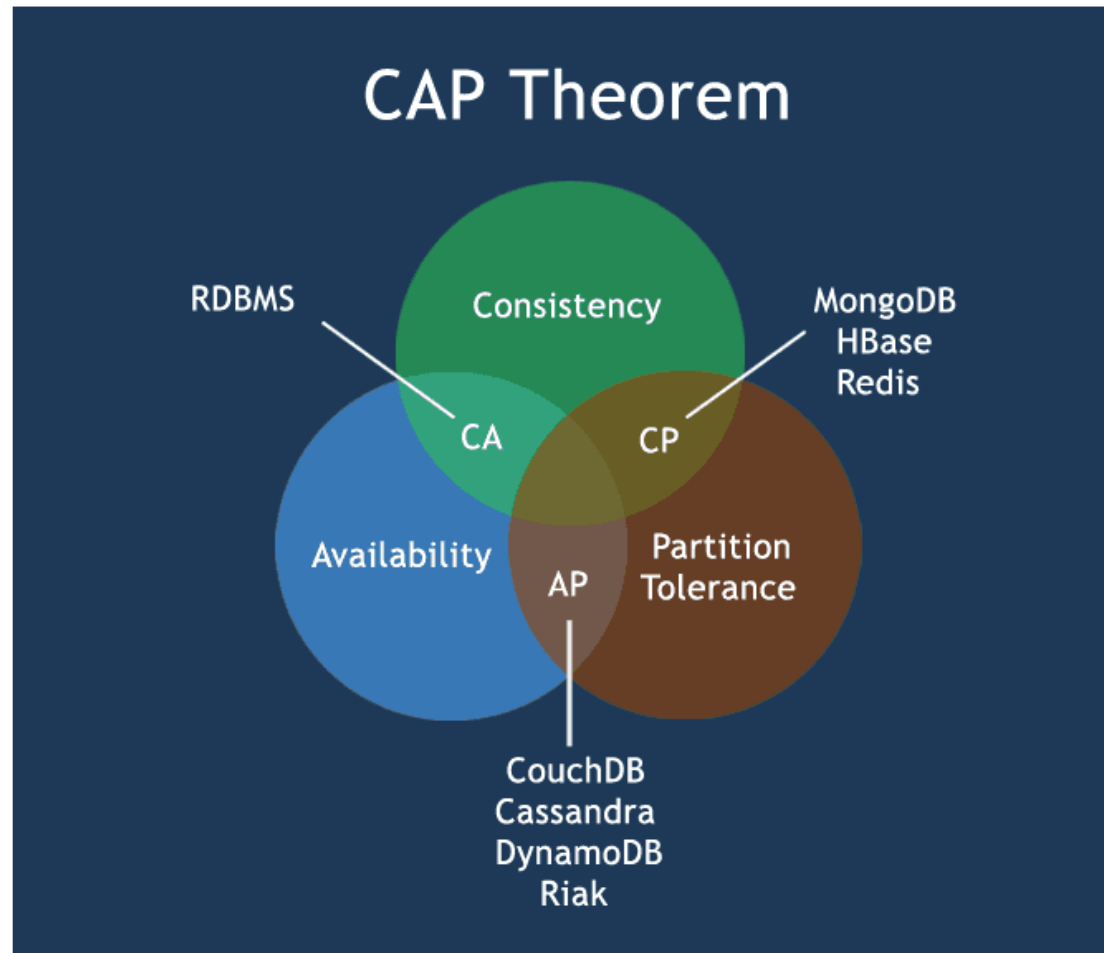
WHO USES NOSQL ?



RDBMS VS NOSQL

RDBMS	NoSQL
Structured and organized data	Unstructured or semi structured data
Structured Query Language (SQL)	No declarative query language
Schema - centric	Schema - free
Data and its relationships are stored in separate tables	Distributed storage
Complex data relationships	Very simple relationships
Joins	Avoid joins
Database-centric	Application-centric or developer-centric
Well defined standards	Standards not yet evolved

CAP THEOREM



CAP THEOREM

- **Consistency** : data in the database remains consistent after the execution of an operation
e.g : after an update operation, all clients see the same data
- **Availability** : system is always on
- **Partition Tolerance** : system continues to function even the communication among the servers is unreliable

BASE CONCEPT

BASE is vague term often used as contrast to ACID

- **Basically Available**

The system works basically all the time

Partial features can occur, but without total system failure

- **Soft state**

The system is in flux, non-deterministic state

Changes occur all the time

- **Eventual consistency**

The system will be in some consistent state

At some time in future

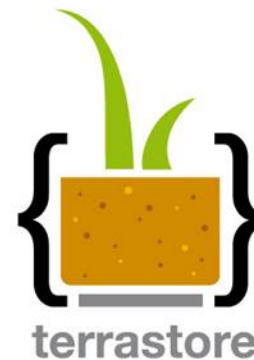
TYPES OF NON-RELATIONAL DATABASES

Type	Description	Examples
Key-value Stores	Simplest NoSQL databases. Every single item in the database is stored as an attribute name (or "key"), together with its value	Riak Voldemort Redis
Document Databases	Pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents	MongoDB ClusterPoint CouchDB MarkLogic
Graph Stores	Used to store information about networks, such as social connections	Neo4J Hyper, GraphDB OrientDB
Wide-column Stores	Optimized for queries over large datasets, and store columns of data together, instead of rows	Cassandra Hbase BigTable

KEY - VALUE STORES: REPRESENTATIVES



DOCUMENT DATABASES: REPRESENTATIVES

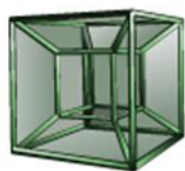


MS Azure
DocumentDB

GRAPH DATABASES: REPRESENTATIVES



COLUMN-FAMILY STORES: REPRESENTATIVES



HYPERTABLE



REASONS TO CHOOSE NOSQL DATABASES

- To improve programmer productivity by using a database that better matches an application's needs.
- To improve data access performance via some combination of handling larger data volumes, reducing latency and improving throughput.

CHOOSING NOSQL DATABASES

It depends on the system requirements and here are some general guidelines.

- Key-value databases are generally useful for storing session information, user profiles, preferences, shopping cart data.
- Document databases are generally useful for content management systems, blogging platforms, web analytics, real-time analytics, ecommerce-applications.
- Column family databases are generally useful for content management systems, blogging platforms, maintaining counters, expiring usage, heavy write volume.
- Graph databases are very well suited to problem spaces where we have connected data such as social networks, special data, routing information for goods and money, recommendation engines.

CURRENT TRENDS IN NOSQL DATABASES

- Multi-Model Databases – support multiple data types (e.g., ArangoDB).
- Serverless Databases – auto-scaling NoSQL in the cloud (Firebase, DynamoDB).
- Integration with AI & Analytics – real-time data pipelines.
- Polyglot Persistence – using both SQL and NoSQL in hybrid systems.
- Increased focus on data governance and consistency.

CHALLENGES IN NON-RELATIONAL DATA MODELLING

- Lack of standard query languages.
- Difficulty enforcing relationships.
- Eventual consistency complicates real-time data.
- Migration from relational to NoSQL requires re-designing schema logic.
- Limited trained expertise and tooling compared to RDBMS.

REFERENCES

- Sadalage, P. J., & Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley Professional.
- Pramod Sadalage, NoSQL Databases : An Overview, <http://www.thoughtworks.com/insights/blog/nosql--databases--overview>.

WHAT YOU HAVE TO DO BY NEXT WEEK

- Try out the self-test questions on the course web.
- Complete the tutorial.