

Introduction to NoSQL Database Systems

Professor Xiannong Meng
Bucknell University
Spring 2018

Information adopted from Wikipedia
<https://en.wikipedia.org/wiki/NoSQL>

History

- So far the databases we discussed are all SQL based (Structured Query Languages).
- SQLs work on relational databases, each of which consists of a collection of tables (relations).
- However, there are huge collection of information, especially on the web that do not fit into this model, e.g., documents, free texts, images, videos, and others.
- This is where NoSQL database comes to play a important role.
- The term was originated in the 60s, gaining wide popularity in the early 21st century as the needs of web 2.0 companies rise.

What is a NoSQL database?

- A **NoSQL** (originally referring to “non SQL” or “non relational”) database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases (SQL databases).

<https://en.wikipedia.org/wiki/NoSQL>

Why NoSQL databases?

- Motivations
 - Simplicity of design
 - Simpler horizontal scaling to clusters of machines (distributed systems)
 - Finer control over availability
 - Structural flexibility tailoring to different problems

Barriers for NoSQL databases

- Barriers for greater adoption of NoSQL databases
 - Compromising consistency in favor of availability, partition tolerance, and speed;
 - Use of low-level query languages instead of SQL;
 - Lack of standard interfaces;
 - Huge previous investments in existing relational databases.

Types and Examples of NoSQL Databases

- There are a variety approaches to classify NoSQL databases. What follows is a basic classification by data model.
- **Column** : A column of a distributed data store is a NoSQL object of the lowest level in a keyspace. It is a tuple consisting of three elements

- Unique name
- Value
- Timestamp



[https://en.wikipedia.org/wiki/Column_\(data_store\)](https://en.wikipedia.org/wiki/Column_(data_store))

- Examples include Accumulo, Cassandra, Druid, HBase, Vertica.

Types and Examples

- **Document database** : A document-oriented database, or document store, is a computer program designed for storing, retrieving and managing document-oriented information.
 - Document-oriented databases are one of the main categories of NoSQL databases, and the popularity of the term "document-oriented database" has grown with the use of the term NoSQL itself. XML databases are a subclass of document-oriented databases that are optimized to work with XML documents. Graph databases are similar, but add another layer, the relationship, which allows them to link documents for rapid traversal.
 - Examples include Apache CouchDB, ArangoDB, BaseX, Clusterpoint, Couchbase, Cosmos DB, IBM Domino, MarkLogic, **MongoDB**, OrientDB, Qizx, RethinkDB

Types and Examples

- **Key-value database** : A data storage paradigm designed for storing, retrieving, and managing associative arrays, a data structure more commonly known today as a dictionary or hash.
 - Examples include Aerospike, Apache Ignite, ArangoDB, Couchbase, Dynamo, FairCom c-treeACE, FoundationDB, InfinityDB, MemcacheDB, MUMPS, Oracle NoSQL Database, OrientDB, Redis, Riak, Berkeley DB, SDBM/Flat File dbm, ZooKeeper.

Types and Examples

- **Graph database** : A database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data.
 - Examples include AllegroGraph, ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J, OrientDB, Virtuoso.

Types and Examples

- **Multi-model database** : A database model that supports multiple data models against a single, integrated backend.
 - Examples include Apache Ignite, ArangoDB, Couchbase, FoundationDB, InfinityDB, MarkLogic, OrientDB.

MongoDB

- We will concentrate on one such example, MongoDB, a document-based database.
- We'll discuss the basic ideas of MongoDB
- We'll implement a MongoDB to support some basic information needs
- We'll also learn how to program MongoDB through Python