IT 775
Database Technology

Data Stores

No SQL

NoSQL DATABASES

NoSQL databases

- A database which provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational
- Does not use SQL as a query language
- Data does not have to be stored in structures described by an organized database schema
- “Non-SQL”; “Not Only SQL”; “Non-Relational”

NoSQL DATABASES

NoSQL databases

With various advancements in the field of computing, scalability, resource utilization and power savings is being given higher priorities. NoSQL database and cloud computing goes hand in hand when compared with SQL databases.

NoSQL DATABASES

NoSQL databases

The main advantage of NoSQL databases is agility. The demand for NoSQL database is growing because most of the applications built, demands high availability, speed, failover options, fault tolerance and consistency which a traditional relational database fails to offer modern web applications.

NoSQL DATABASES

NoSQL databases

Many NoSQL stores compromise consistency in favor of availability, partition tolerance, and speed. Barriers to the greater adoption of NoSQL stores include the use of low-level query languages (instead of SQL, for instance), lack of ability to perform ad hoc joins across tables, lack of standardized interfaces, and huge previous investments in existing relational databases.

NoSQL DATABASES

NoSQL databases

Most NoSQL stores lack true ACID (atomicity, consistency, isolation, durability) transactions, although a few databases have made them central to their designs.

Most offer a concept of "eventual consistency", in which database changes are propagated to all nodes "eventually" (typically within milliseconds), queries for data might not return updated data immediately or might result in reading data that is not accurate, a problem known as stale reads.

NoSQL DATABASES

NoSQL databases

Additionally, some NoSQL systems may exhibit lost writes and other forms of data loss. Some NoSQL systems provide concepts such as write-ahead logging to avoid data loss. For distributed transaction processing across multiple databases, data consistency is an even bigger challenge that is difficult for both NoSQL and relational databases. Relational databases "do not allow referential integrity constraints to span databases". Few systems maintain both ACID transactions and X/Open XA standards for distributed transactions.

NoSQL DATABASES

Key–value store:

Key–value (KV) stores use the associative array (also called a map or dictionary) as their fundamental data model. In this model, data is represented as a collection of key–value pairs, such that each possible key appears at most once in the collection.

NoSQL DATABASES

Key–value store:

This model is one of the simplest non-trivial data models, and richer data models are often implemented as an extension of it. The key–value model can be extended to a discretely ordered model that maintains keys in lexicographic order. This extension is computationally powerful, in that it can efficiently retrieve selective key ranges.

NoSQL DATABASES

Key–value store:

Key–value stores can use consistency models ranging from eventual consistency to serializability. Some databases support ordering of keys. There are various hardware implementations, and some users store data in memory (RAM), while others on solid-state drives (SSD) or rotating disks (aka hard disk drive (HDD)).

NoSQL DATABASES

Document store:

The central concept of a document store is that of a "document". While the details of this definition differ among document-oriented databases, they all assume that documents encapsulate and encode data (or information) in some standard formats or encodings. Encodings in use include XML, YAML, and JSON and binary forms like BSON. Documents are addressed in the database via a unique key that represents that document. Another characteristic is an API or query language to retrieve documents based on their contents.

NoSQL DATABASES

Document store:

Different implementations offer different ways of organizing and/or grouping documents:

- Collections
- Tags
- Non-visible metadata
- Directory hierarchies

Collections could be considered analogous to tables and documents analogous to records. But they are different: every record in a table has the same sequence of fields, while documents don't.

NoSQL DATABASES

Graph:

Graph databases are designed for data whose relations are well represented as a graph consisting of elements connected by a finite number of relations. Examples of data include social relations, public transport links, road maps, network topologies, etc.

NoSQL DATABASES

NoSQL database example: MongoDB

A **document-oriented** NoSQL database

Organized around collections of documents

- Collection – equivalent to a table
- Documents – equivalent to a table row

TinyDB is another example

MongoDB Example

Code for document creation

```
h1 = {name: "Metro Blu", address: "Chicago, IL", rating: 3.5}
db.hotels.save(h1)
h2 = {name: "Experiential", address: "New York, NY", rating: 4}
db.hotels.save(h2)
h3 = {name: "Zazu Hotel", address: "San Francisco, CA", rating: 4.5}
db.hotels.save(h3)

h4 = {name: "Solace", address: "Los Angeles, CA"}
db.hotels.save(h4)
```

MongoDB Example
Queries

```
db.hotels.find()

db.hotels.find( { address : { $regex : "CA" } } )
```

MongoDB Example
Updates

```
db.hotels.update( { name:“Zazu Hotel” }, { $set : {wifi: “free”} } )

db.hotels.update( { name:“Zazu Hotel” }, { $set : {parking: 45} } )
```

NoSQL DATABASES

NoSQL databases vs. relational databases

	NoSQL	Relational
Fixed Schema	*No*	*Yes*
Scalability	*Horizontal*	*Vertical*
Transaction Properties	*BASE*	*ACID*
Standard Language	*none*	*SQL*

ACID – atomicity, consistency, isolation, durability

BASE – basically available soft-state eventual consistency