NoSQL | Meerza Ahmed

Tuesday, November 7, 2023 1:00 PM

Lecture 12: Link | Syllabus

**Class Week 12/15*

Database Systems by Coronel & Morris

Chapter 14: Big Data Analytics & NoSQL

*** 14.3** *NoSQL*

NoSQL is the unfortunate name given to a broad array of nonrelational database technologies that have developed to address the challenges represented by Big Data. The name is unfortunate in that it does not describe what the NoSQL technologies are, but rather what they are not. There are literally hundreds of products that can be considered as being under the broadly defined term NoSQL. Most of these fit roughly into one of four categories : key value data stores, document databases, column-oriented databases, and graph databases.

ACID	BASE
Strong consistency	Weak consistency – stale data OK
Isolation	Availability first
Focus on "commit"	Best effort
Nested transactions	Approximate answers OK
Availability?	Aggressive (optimistic)
Conservative (pessimistic)	Simpler!
Difficult evolution (e.g. schema)	Faster
	Easier evolution

* The NoSQL DBs are characterized by their 'BASE' property, in contrast with RDBMS' 'ACID' property.
BASE stands for BAsic availability (db is up most of the time), Soft state (of consistency - consistency is not guaranteed while data is written to a node, or between replicas), Eventual consistency (at a later point in time, by push or pull, data will become consistent across nodes and replicas).

*** 14.3-a** *Key-Value Databases*

Key-value (KV) databases are conceptually the simplest of the NoSQL data models. A KV database is a NoSQL database that stores data as a collection of key-value pairs. The key acts as an identifier for the value. The value can be anything such as text, an XML document, or an image. Key-value pairs are typically organized into "buckets." A bucket can roughly be thought of as the KV database equivalent of a table. A bucket is a logical grouping of keys. Key values must be unique within a bucket, but they can be duplicated across buckets. All data operations are based on the bucket plus the key

FIGURE 14.7 KEY-VALUE DATABASE STORAGE

Bucket = Customer

Key	Value
10010	"LName Ramas FName Alfred Initial A Areacode 615 Phone 844-2573 Balance 0"
10011	"LName Dunne FName Leona Initial K Areacode 713 Phone 894-1238 Balance 0"
10014	"LName Orlando FName Myron Areacode 615 Phone 222-1672 Balance 0"

*** 14.3-b** *Document Databases*

Document databases are conceptually similar to key-value databases, and they can almost be considered a subtype of KV databases. A document database is a NoSQL database that stores data in tagged documents in key-value pairs. Unlike a KV database where the value component can contain any type of data, a document database always stores a document in the value component. The document can be in any encoded format, such as XML, JSON (JavaScript Object Notation), or BSON (Binary JSON). Another important difference is that while KV databases do not attempt to understand the content of the value component, document databases do. Tags are named portions of a document. For example, a document may have tags to identify which text in the document represents the title, author, and body of the document. Within the body of the document, there may be additional tags to indicate chapters and sections. Despite the use of tags in documents, document databases are considered schema-less, that is, they do not impose a predefined structure on the data that is stored.

* Just as KV databases group key-value pairs into logical groups called buckets, document databases group documents into logical groups called collections. While a document may be retrieved by specifying the collection and key, it is also possible to query based on the contents of tags

FIGURE 14.8 DOCUMENT DATABASE TAGGED FORMAT

Collection = Customer

Key	Document
10010	(LName: "Ramas", FName: "Alfred", Initial: "A", Areacode: "615", Phone: "844-2573", Balance: "0")
10011	(LName: "Dunne", FName: "Leona", Initial: "K", Areacode: "713", Phone: "894-1238", Balance: "0")
10014	(LName: "Orlando", FName: "Myron", Areacode: "615", Phone: "222-1672", Balance: "0")

*** 14.3-c** *Column-Oriented Databases*

The term Column-oriented database can refer to two different sets of technologies that are often confused with each other. In one sense, column-oriented database or columnar database can refer to traditional, relational database technologies that use column-centric storage instead of row-centric storage. Relational databases present data in logical tables; however, the data is actually stored in data blocks containing rows of data. All of the data for a given row is stored together in sequence with many rows in the same data block.

* The other use of the term *column-oriented database*, also called column family database, is to describe a type of NoSQL database that takes the concept of column-centric storage beyond the confines of the relational model. As NoSQL databases, these products do not require the data to conform to predefined structures nor do they support SQL for queries.
* As more columns are added, it becomes clear that some columns form natural groups , these groupings are used to create super columns. A super column is a group of columns that are logically related.

FIGURE 14.9 COMPARISON OF ROW-CENTRIC AND COLUMN-CENTRIC STORAGE

CUSTOMER relational table

Cus_Code	Cus_LName	Cus_FName	Cus_City	Cus_State
10010	Ramas	Alfred	Nashville	TN
10011	Dunne	Leona	Miami	FL
10012	Smith	Kathy	Boston	MA
10013	Olowski	Paul	Nashville	TN
10014	Orlando	Myron		
10015	O'Brian	Amy	Miami	FL
10016	Brown	James		
10017	Williams	George	Mobile	AL
10018	Farriss	Anne	Opp	AL
10019	Smith	Olette	Nashville	TN

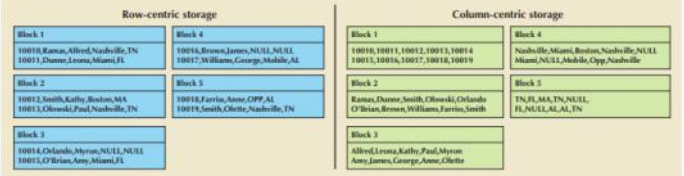

FIGURE 14.10 COLUMN FAMILY DATABASE

Column Family Name	CUSTOMERS	
Key	Rowkey 1	
Columns	City	Nashville
	Fname	Alfred
	Lname	Ramas
	State	TN
Key	Rowkey 2	
Columns	Balance	345.86
	Fname	Kathy
	Lname	Smith
Key	Rowkey 3	
Columns	Company	Local Markets, Inc.
	Lname	Dunne

* 14.3-c *Graph Databases*

Graph database is a NoSQL database based on graph theory to store data about relationship-rich environments. Graph theory is a mathematical and computer science field that models relationships, or edges, between objects called nodes. Modeling and storing data about relationships is the focus of graph databases. The concept of graph theory immediately provides a rich source for algorithms and applications that have helped graph databases gain in sophistication very quickly.

* The primary components of graph databases are nodes, edges, and properties; A node corresponds to the idea of a relational entity instance. The node is a specific instance of something we want to keep data about. Properties are like attributes; they are the data that we need to store about the node. All agent nodes might have properties like first name and last name, but all nodes are not required to have the same properties. An edge is a relationship between nodes. Edges (shown as arrows in Figure 14.10) can be in one direction, or they can be bidirectional. Note that edges can also have properties. A query in a graph database is called a traversal. Instead of querying the database, the correct terminology would be traversing the graph.

FIGURE 14.11 GRAPH DATABASE REPRESENTATION