Database Management Systems in Social Networking

By Andrew, Graeme and Juliana

Introduction to Social networks as a Dataset

- Users have profile information, posts, images and data collected such as phone locations.
- Users connect and interact. This includes friending, commenting and more.
- Often very graph-like.

Challenges of Social Media Database Systems

-Big: 155 million tweets per day

-Real time responses

-Required data to store always changing

Requirements of Databases

- Ability to modify schema at low cost
- Fast access to related data
- Recovery for failed nodes with no down time

Modern databases used in social networks

- Relational, Col-store, Document Store, Key-value and Graphs databases
- MySQL, HBase, Cassandra, MongoDB, CouchDB, DynamoDB, Riak, Voldemort, Neo4j, FlockDB, InfoGrid, OrientDB, AllegroGraph

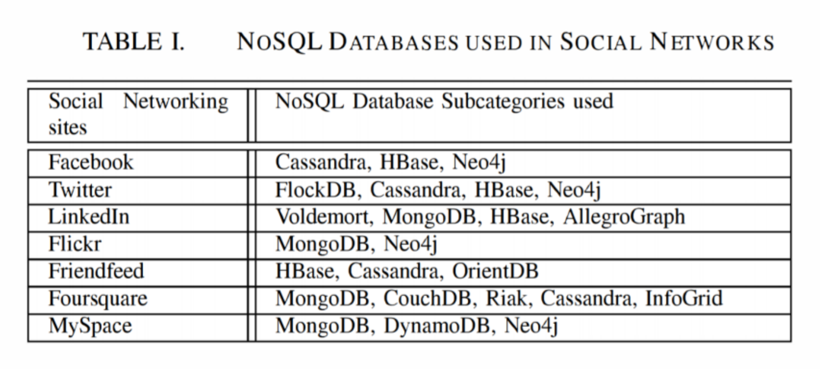


TABLE I. NoSQL DATABASES USED IN SOCIAL NETWORKS

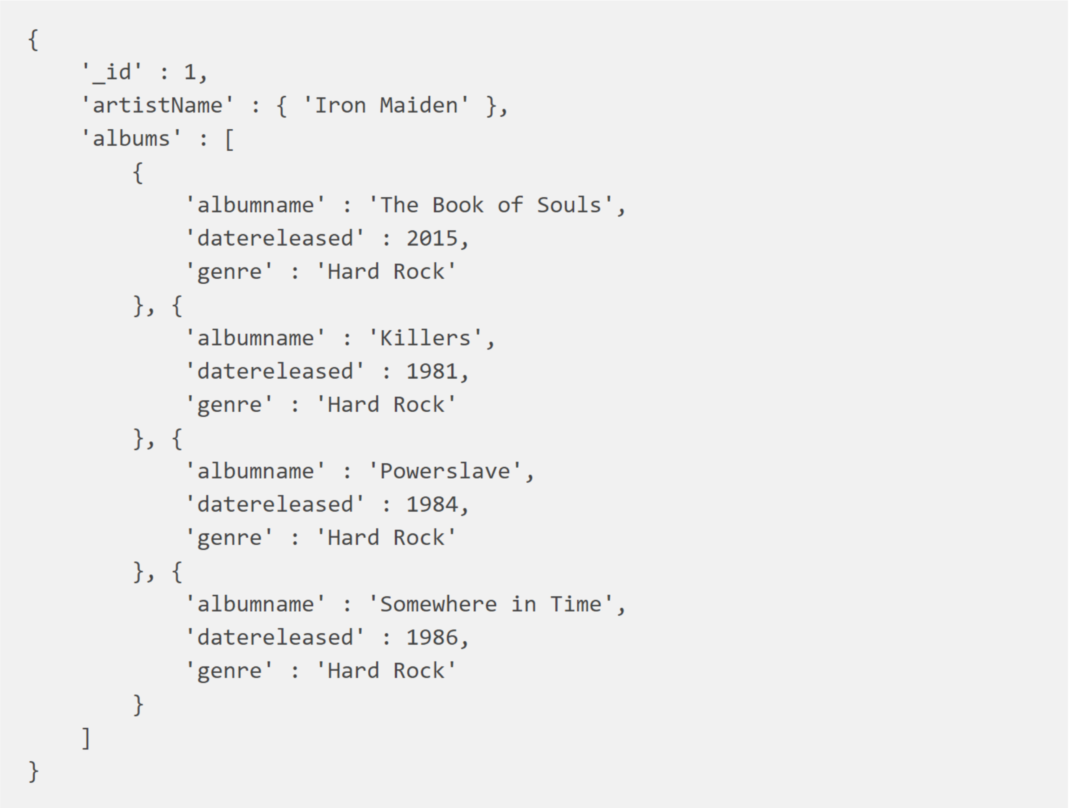
Social Networking sites	NoSQL Database Subcategories used
Facebook	Cassandra, HBase, Neo4j
Twitter	FlockDB, Cassandra, HBase, Neo4j
LinkedIn	Voldemort, MongoDB, HBase, AllegroGraph
Flickr	MongoDB, Neo4j
Friendfeed	HBase, Cassandra, OrientDB
Foursquare	MongoDB, CouchDB, Riak, Cassandra, InfoGrid
MySpace	MongoDB, DynamoDB, Neo4j

Source: Nicolas Ruflin, Helmar Burkhart, and Sven Rizzotti. 2011. “Social-data storage-systems. In Databases and Social Networks”

Document store

Store record (and additional information) in a document.

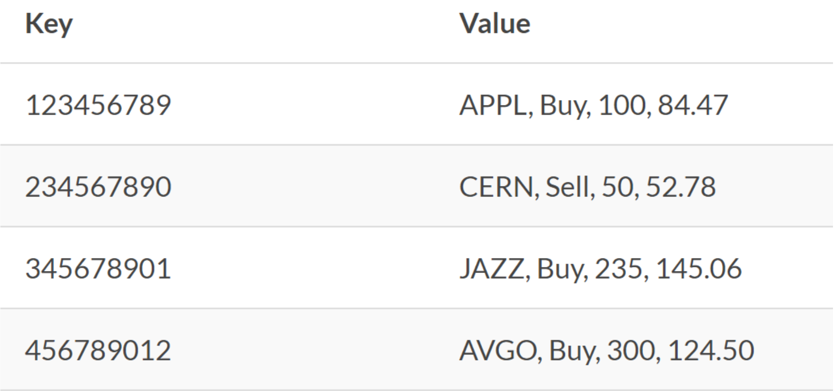
Adept for storing, retrieving and manipulating document-oriented information



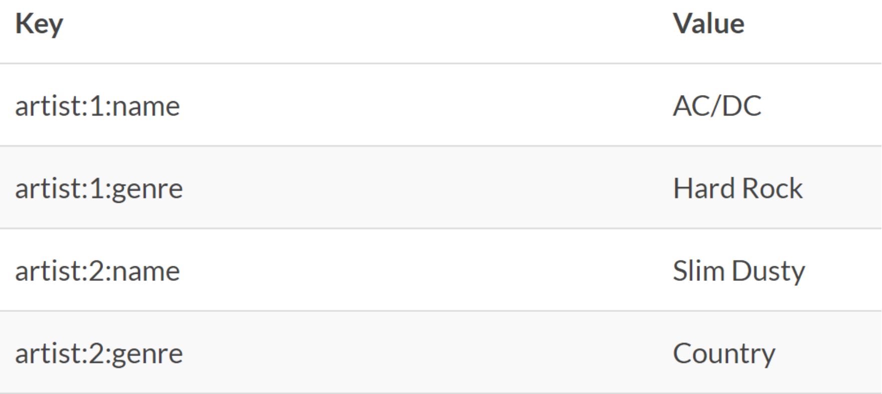
```
{
    '_id' : 1,
    'artistName' : { 'Iron Maiden' },
    'albums' : [
        {
            'albumname' : 'The Book of Souls',
            'datereleased' : 2015,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Killers',
            'datereleased' : 1981,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Powerslave',
            'datereleased' : 1984,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Somewhere in Time',
            'datereleased' : 1986,
            'genre' : 'Hard Rock'
        }
    ]
}
```

Source : https://database.guide/what-is-a-document-store-database/

Key-value databases



Key	Value
123456789	APPL, Buy, 100, 84.47
234567890	CERN, Sell, 50, 52.78
345678901	JAZZ, Buy, 235, 145.06
456789012	AVGO, Buy, 300, 124.50



Key	Value
artist:1:name	AC/DC
artist:1:genre	Hard Rock
artist:2:name	Slim Dusty
artist:2:genre	Country

Unstructured, scalability.

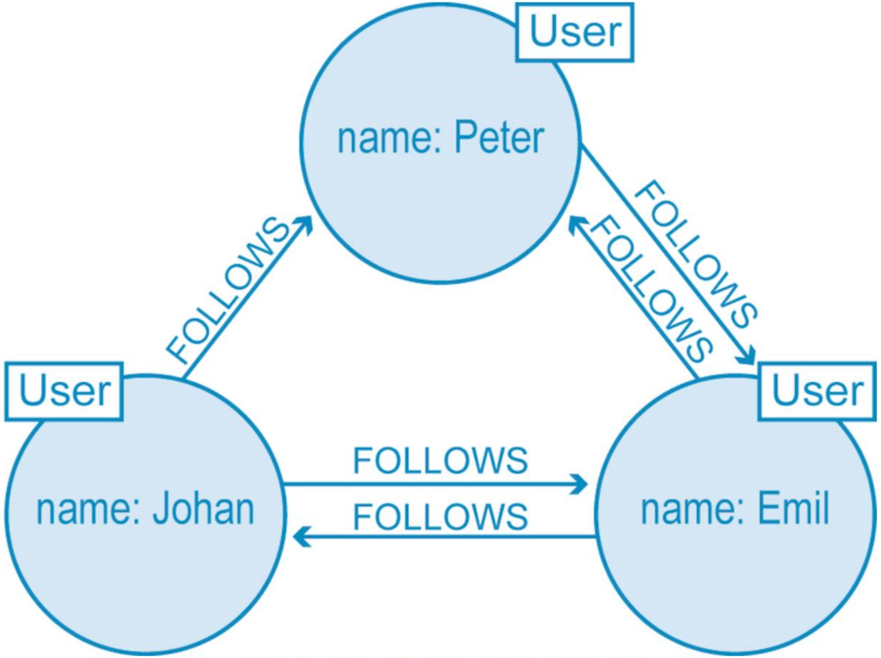
Retrieval in real time applications.

Source: https://database.guide/what-is-a-document-store-database/

Graph

Relationships between entities modelled with Graph

Nodes and adjacency lists

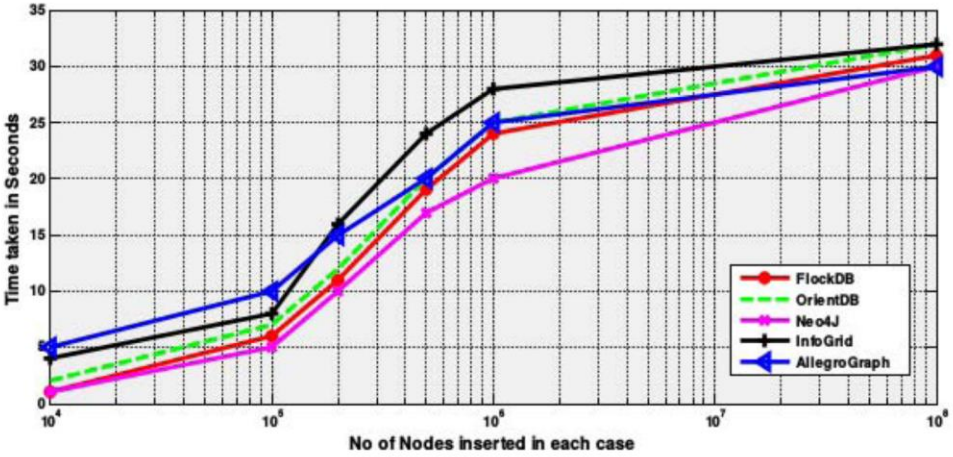


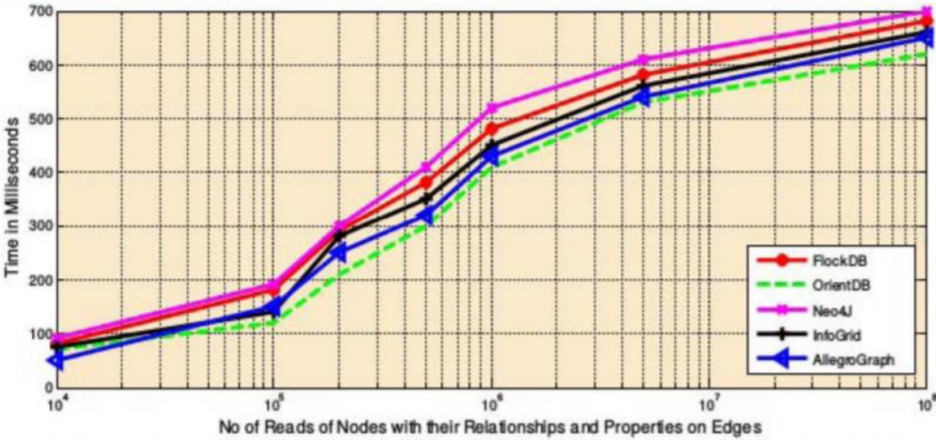
Twitter users represented in a graph database model.

Source: https://neo4j.com/blog/why-graph-databases-are-the-future/

Graph Databases Compared in Second Paper

Tradeoff between writes and reads





Source: A. B. Mathew and S. D. Madhu Kumar, "Analysis of data management and query handling in social networks using NoSQL databases,"

Graph Databases: A good idea, not common in practice

- Graph DB’s model the data better than other options
- Not used very often
- Scalability issues
- Trouble retrieving data efficiently

When to use which database

Depends on the context

Structure vs scalability tradeoff

Database Model	Examples	Uses
Relational	MySQL	Common when data entities have fixed-sized relations e.g. transactions between entities.
Column	Hadoop Hbase Cassandra	Well suited for applications like data warehousing, customer association management systems.
Document	MongoDB CouchDB	Adept for storing, retrieving and manipulating document-oriented information
Key Value Store	DynamoDB Riak Voldemort	Prefered for single query retrieval in real time applications.
Graph DB	Neo4j FlockDB InfoGrid OrientDB AllegroGraph	Supports many applications like Facebook in finding friends, other relations through graph search, etc...

Database Performance by type

- Less structured data is easier to scale, but cannot model entities as well.

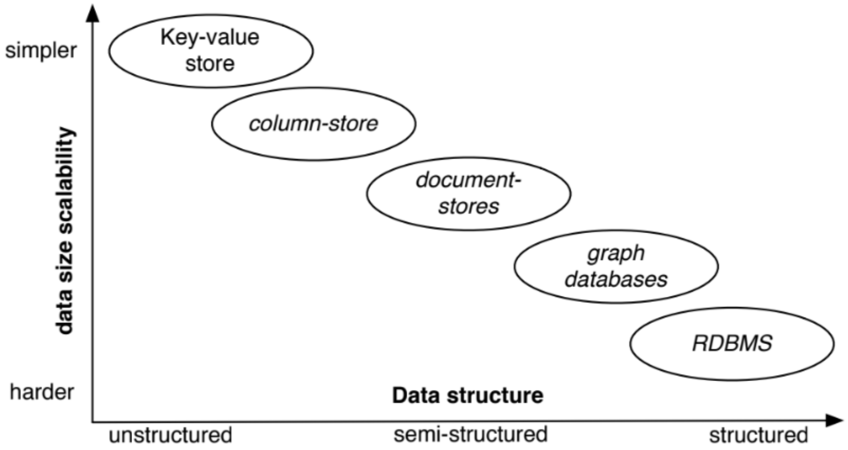


Figure 1: Data size scalability vs data structure

Source: Nicolas Ruflin, Helmar Burkhart, and Sven Rizzotti. 2011. “Social-data storage-systems. In Databases and Social Networks”

UseKit

-What is useKit? 50,000 nodes and 100,000 edges to simulate a social networks db.

-Challenges using it as a benchmark.

- How to convert it to different models.

Performance of Databases on useKit

- Query: Retrieve a node and its associated data
  - SQL, Neo4j, Redis: 3 requests
  - Hbase, Casandra, MongoDB, Couch DB, Riak: 1 request

- Critique of the paper: want numerical results

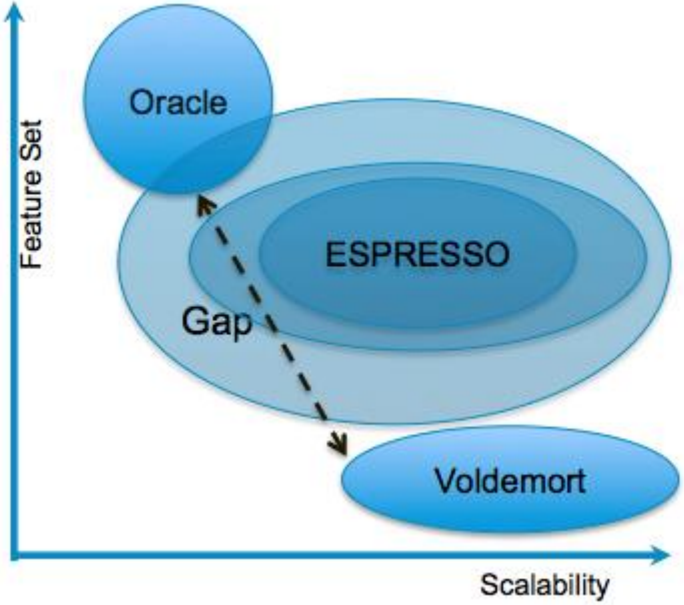
Industry Solutions

- Facebook adds a cache to sql to save on joins
- LinkedIn discussed later on
- Augmentation works well in practice.

Case Study: LinkedIn Espresso

RDBMS has many limitations in its ability to support LinkedIn’s data model

- Schema updates are costly
- Data model doesn’t cleanly fit into common relational schema patterns
- Costly cache support to meet scale and latency requirements



Case study: LinkedIn Espresso

- Data grouped into partitions with a common partition key (e.g mailbox ID)
- Multiple operations on different documents in the same partition can occur within the same transaction
- Use prefixed-based indexing including partition key for faster lookup
- On-the-fly schema evolution with backwards compatibility
- Sharding with the master-slave model: allow different master and slave partitions within a node

Espresso Tests

- Full text-based indexing: Lucene vs. prefix-indexing
    - Conclusion: prefix indexing performs better for many mailboxes with few messages since Lucene is log-based and immutable, so documents must be remade on every update

- Distribution: Espresso vs. Sharded MySQL
    - Conclusion: Espresso has a faster response time upon failure as failed partitions get distributed across cluster and reads can be performed on slaves, whereas MySQL contains all master partitions in a master node, so all requests are blocked until a slave node promotion

Takeaways

- Social Networks require large scale, fast responses and flexible schemas
- Different available DBMS systems are capable of doing a subset of these
- Augmentation tends to increase performance
- Future idea: Is there a maximum network size? Can this be used to improve performance?