Elmélet

MongoDB

NoSQL:

- Non-relational model
- Horizontally scalability
- Flexible schema
- Document-oriented
- General purpose
- Free and open-source:
  - Enterprise version exists
  - Managed: MongoDB Atlas

Compnaies using MongoDB:

- EA
- Bosch IoT
- Cisco
- eBay
- HSBC

Data model:

- BSON = binary JSON

- Single entity

- Embedded struture

- No schema

- Support for unstructured data

- Embedded structure avoids need for JOINS

  - Horizontal scalability

- Flexible schema

  - Ease of development

- Map well to language structures like objects

  - No need for awkward mapping
  - Supports polymorphism

Querry Samples:

- Query filter by field (value)
- Query on embedded field
- Query on array element
- Add to array

Data types:

BSON – binary JSON

- Converts to Extended JSON

Primitive types

- bool
- int, long
- double, decimal
- string
- regex
- date, timestamp
- binData

Embedded structures

- object
- array

Special types

- objectId
- javascript
- null
- minKey
- maxKey

In MongoDB:

- database = database
- table = collection
- row = document
- column = field
- index = index
- table joins = embedded documents
- primary key = primary key (_id field)
- unique key = unique index
- foreign key = foreign key
- stored procedure = user-defined function
- trigger = user-defined function
- aggregate function = aggregation pipeline

Atomicity and transactions

- Atomicity on document level
- Multi-document operations are not atomic by default

Multi document transactions:

- Provide distributed ACID
- Transaction level consistency settings
- Has significant performance impact
- Has limitations
- Not as mature as in relational DBs

Data modelling

Denormalized:

- Avoids JOIN
- Atomic

Normalized:

- Avoids duplication
- Flexible
- Can JOIN via $lookup stage
- Needs transaction to be atomic

Architecture:

Replication:

Provides:

- High availability
- Scalability
- Redundancy

Replica set:

- 3 or more nodes
- Single primary
- Several secondaries
- All writes go to primary
- All changes are replicated to secondaries

Consistency:

- Availability:
  - The cluster returns a responsible response within a reasonable time
- Consistency:
  - The cluster returns a response that reflects the current state of the cluster
- Partition tolerance:

    - The cluster continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

Automatic Failover:

- Election takes ~12 seconds (median with default settings)
- No writes until new primary elected

Read preference:

- Primary (default)
- Allow non-primary reads:
    - primaryPreferred
    - secondary
    - secondaryPreferred
    - nearest

Main goal is better availability, not throughput scaling

Read concern:

- local (default)
- majority
- available
- linearizable
- snapshot

Write concern:

- w: 1 (default)
- w: majority
- w: 0
- w: N

Durability:

- Single machine:
    - Works on documents in memory
    - Periodic checkpoint
    - Journal
- Replica set:
    - Majority of nodes must acknowledge write
    - Majority of nodes must write to disk
    - Majority of nodes must acknowledge write to journal

Sharding:

- Shard:
    - Stores a subset of sharded data

- No direct client connection
- Config server:
  - Stores metadata
  - Direct client connection
  - Holds metadata for sharded cluster
- Router:
  - Routes client requests to shards
  - Direct client connection
  - Caches metadata from config server
  - Stateless

Choosing a shard key:

- Data distribution:

  - Evenly distributed
  - Cardinality
  - Frequency
  - Monotonicity

- Sharding strategy:

  - Hashed
  - Range
  - Compound