

Principles of Data Management

Alvin Lin

August 2018 - December 2018

NoSQL

The term NoSQL refers to any non-relational non-row oriented database.

- With column oriented databases, you can only query on certain columns (indexed), and you do not receive the data in the entire row. There are many implementations of column oriented databases that behave slightly differently, some may return a pointer to the row. Column oriented databases make it easy to retrieve attributes in a single column without the need to load unnecessary attributes.
- Another type of database is the key-value database, where values are stored based on a key. You can only query on the key, and operations may be extremely expensive.
- Document based databases are similar to key-value databases except that you can query on any attribute. You can also index on any attribute, thus adding performance to any query. Every document can also have different attributes, so it is up to the application inserting data to enforce attribute limitations. The most popular one is MongoDB.

MongoDB

MongoDB is a document based database whose objects are stored in BSON format, or Binary JavaScript Object Notation. It's object hierarchy involves databases which include collections of documents, which may have differing attributes and values between them. In terms of security, MongoDB has no username or password by

default. Security is intended to be performed by extrinsic firewalls so that databases on a local area network can be easily used to replicate data for distributed systems.

MongoDB has built in fault tolerance and horizontal scalability. In a distributed MongoDB environment, a MongoDB client connects to multiple MongoDB instances to handle data replication and distribution (sharding). In the event that any client goes down, an arbitrary connected server can be elected to take over as the client to handle data replication and distribution.

Collections in MongoDB by default have a required `_id` field which acts as a primary key. It can be changed and determines how documents are organized. The seven most common aggregation specifiers are: `$project`, `$skip`, `$sort`, `$match`, `$group`, `$count`, and `$limit`.

Example Usage

```
db.dog.find()
db.dog.insert({ name: "spot", age: 7 })
db.dog.insertMany([
  { name: "blue", age: 20 },
  { name: "green", age: 30 }
])
db.dog.replaceOne({ name: "spot" }, { name: "greg", age: 10 })
db.dog.deleteOne({ name: "spot" })
db.dog.updateOne({ name: "blue" }, { age: 31, size: "large" })
db.dog.updateMany({ age: { $gt : 19 }}, { status: "old" })
db.dog.aggregate([
  $match: {
    age: {
      $lt: 20
    }
  },
  $sort: {
    name: 1
  }
])
```

You can find all my notes at <http://omgimanerd.tech/notes>. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech