



Technology Series

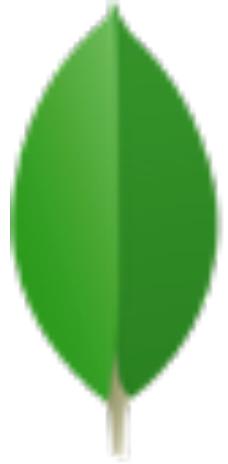
MongoDB

05

Presented by  
Subhash EP



Subhash



# mongoDB

{ name: mongo, type: DB }

# Indexing

- Indexing documents
- Index types
- Special index properties

# Indexing Document

## Index

### A

- About cordless telephones 51
- Advanced operation 17
- Answer an external call during an intercom call 15
- Answering system operation 27

### B

- Basic operation 14
- Battery 9, 38

### C

- Call log 22, 37
- Call waiting 14
- Chart of characters 18

### D

- Date and time 8
- Delete from redial 26
- Delete from the call log 24
- Delete from the directory 20
- Delete your announcement 32
- Desk/table bracket installation 4
- Dial a number from redial 26

### Dial type 4, 12

- Directory 17
- DSL filter 5

### E

- Edit an entry in the directory 20
- Edit handset name 11

### F

- FCC, ACTA and IC regulations 53
- Find handset 16

### H

- Handset display screen messages 36
- Handset layout 6

### I

- Important safety instructions 39
- Index 56-57
- Installation 1
- Install handset battery 2
- Intercom call 15
- Internet 4

# Indexing

- indexes are data structures that hold part of the data from our main data source.
- In relational databases,
  - indexes hold parts of a table
- while in MongoDB,
  - since indexes are on a collection level,
  - these will hold part of a document

# Example

- 0301.txt

# Standalone

- So far, we've been using a standalone server, a single mongod server
- what if your server crashes or becomes unavailable?

# Replication

- Replication is a way of keeping identical copies of your data on multiple servers
  - and is recommended for all production deployments.
- Replication keeps your application running and your data safe,
  - even if something happens to one or more of your servers.

# MonogDB Replication

- With MongoDB, you set up replication by creating a replica set.
- A replica set is a group of servers
  - with one primary, the server taking client requests, and
  - multiple secondaries, servers that keep copies of the primary's data.
  - If the primary crashes, the secondaries can elect a new primary from amongst themselves.

# Create a MongoDB Replica Set

- setting up a three-member replica set on your local machine
- not suitable for production, but it's a nice way to familiarize yourself with replication

# Step # 1

```
$ mongo --nodb
```

## Step # 2

```
replicaSet = new ReplSetTest({"nodes" : 3})
```

## Step # 3

```
// starts three mongod processes
```

```
> replicaSet.startSet()
```

```
>
```

```
> // configures replication
```

```
> replicaSet.initiate()
```

# Step # 4

- Open another shell

mongo --nodb

# Step # 5

```
conn1 = new Mongo("localhost:31000")
```

```
primaryDB = conn1.getDB("test")
```

```
primaryDB.isMaster()
```

# Step # 6

```
for (i=0; i<1000; i++)  
{ primaryDB.coll.insert({count: i}) }
```

```
primaryDB.coll.count()
```

# Step # 7

```
conn2 = new Mongo("localhost:31001")
```

```
secondaryDB = conn2.getDB("test")
```

## Step # 8

secondaryDB.coll.find()

error: { "\$err" : "not master and slaveok=false", "code" : 13435 }

(Error! This is to protect your application from accidentally connecting to a secondary and reading stale data)

# Step # 9

- To allow queries on the secondary, we set an “I’m okay with reading from secondaries” flag, like so:

```
conn2.setSlaveOk()
```

```
secondaryDB.coll.find()
```

# Introduction to Sharding

- Sharding refers to the process of splitting data up across machines;
- the term partitioning is also sometimes used to describe this concept.
- By putting a subset of data on each machine,
- it becomes possible to store more data and handle more load without requiring larger or more powerful machines, just a larger quantity of less-powerful machines

# Introduction to Sharding

- Manual sharding can be done with almost any database software.
- Manual sharding is when an application maintains connections to several different database servers, each of which are completely independent.
- The application manages storing different data on different servers and querying against the appropriate server to get data back.
- This approach can work well but becomes difficult to maintain when adding or removing nodes from the cluster or in the face of changing data distributions or load patterns.

# Introduction to Sharding

- MongoDB supports autosharding, which tries to both abstract the architecture away from the application and simplify the administration of such a system.
- MongoDB allows your application to ignore the fact that it isn't talking to a standalone MongoDB server, to some extent.
- On the operations side, MongoDB automates balancing data across shards and makes it easier to add and remove capacity.

# That's all



## End of Session

©

[subhash.ep@gmail.com](mailto:subhash.ep@gmail.com)  
[linkedin.com/in/subhashep](https://linkedin.com/in/subhashep)