



Technology Series

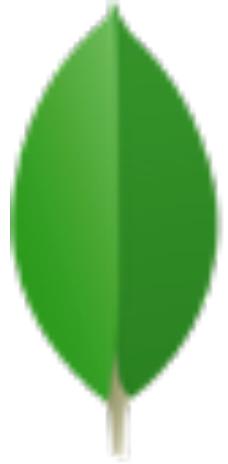
MongoDB

03

Presented by  
Subhash EP



Subhash



# mongoDB

{ name: mongo, type: DB }

# \_id and ObjectIds

- MongoDB document must have an "\_id" key.
- The "\_id" key's value can be any type,
  - but it defaults to an **ObjectId**.
- In a single collection,
  - every document must have a unique "\_id"
- That is, if you had two collections,
  - each one can have a document "\_id" of 123.
  - no collection can have document with "\_id" of 123.

# ObjectId

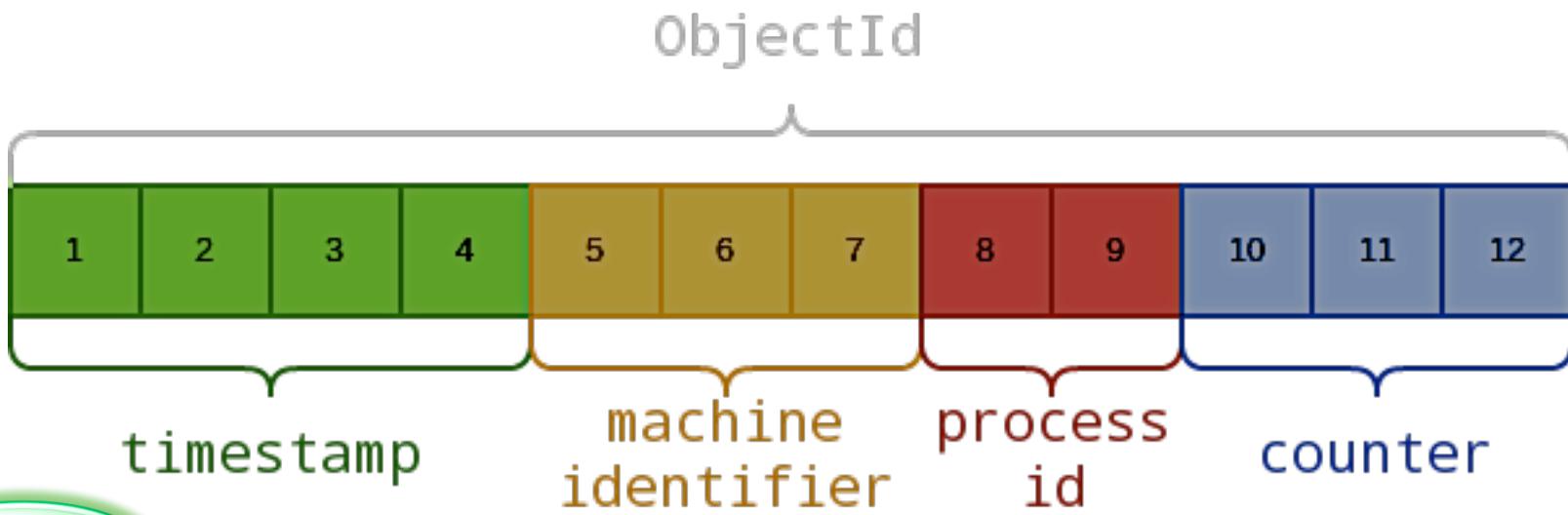
- default type for "\_id".
- lightweight and generate in a global UID across different machines.
- Why ObjectIds?
- Why not auto incrementing primary keys?
- MongoDB is **distributed** database
- important to generate unique identifiers in a **sharded** environment.

# ObjectIds

- use 12 bytes of storage
- a hexadecimal string of 24 digits
  - 2 digits for each byte
  - causes them to appear larger than they are

# creating multiple new ObjectIds

- you can see that only the last few digits change each time.
- The 12 bytes of an ObjectId are generated as follows:



# Bulk Insert

```
db.foo.insert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])
```

```
db.foo.find()
```

# NoSQL – what's the big deal?

- if the basic structure of the document is JSON
  - something so simple and textual,
  - what can be so complex about the creation of a NoSQL database?

# Simple or Complex

- NoSQL databases can be very simple.
- But, this does not mean that their structure will be less complex than that of a relational database.
- It will be different, though!
- How about relationships?

# Relationships in MongoDB

- MongoDB has its own way
- There are two ways:
  - Embedded documents
  - References

# Type# 1: embedded documents

- subdocuments

```
{  
  a: "b",  
  {  
    c: "d"  
    e: "f"  
  }  
}
```

See the example embedded document (0105.txt)

# Advantage of Embedded Document

- with a single query, we have all the data we need to present to the user.
- same applies to updates: with just one query, we can modify the content of this document.
- But the document should not exceed the BSON size limit of 16 MB.

## Type# 2: Reference Documents

```
{  
    reference_id: 12  
}  
{  
    reference_id: 12  
}  
{  
    _id: 12  
}
```

# Reference and Normalization!

- creating a reference in MongoDB is the way we have to "normalize" our model.
- why we are considering relationships in a non-relational universe?
  - We often use the concepts of relational modeling to solve common problems.
  - As stated before, to eliminate redundancy, documents can refer to each other.

# MongoDB and Joins

- MongoDB does not support joins.
- This means that,
  - even with the reference to another document,
  - you must perform at least two queries
  - to get the complete information you need.



# Repeating Data – an example

- See an example (0106.txt)
- We have a products collection
- Same Supplier Key repeats

# Example: Refine with 2 Collections

- Instead of this repetition of data,
- have two collections:
  - products and
  - suppliers,
- see the example:
  - 0107.txt – not so refined
  - 0108.txt – refined better

# Data Modeling Considerations

- Whether consistency is the priority
- Whether read is the priority
- Whether write is the priority
- What update queries we will make
- Document growth

# One-to-one

- One-to-one relationships are simpler than the others.
- Most of the time,
  - we will map this relationship with embedded documents,
  - especially if it is a "contains" relationship.
- Example:
  - 0109.txt – referenced data
  - 0110.txt – embedded data

# One-to-many

- more complex than one-to-one
- to decide to embed or to references
  - consider the "many" side of the relationship.
  - If many side displayed with its parents,
    - then embed the data;
    - otherwise, references on the parents
- Example:
  - 0111

# Many-to-many

- NOT something trivial,
  - in a relational world often represented as a join table
  - in the non-relational one, it can be represented in many different ways
- Example:
  - 0113.txt
  - 0114.txt
  - 0115.txt
  - 0116.txt

# That's all



## End of Session

©

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