

Indexing



Nuri Halperin

@nurih | www.plusnconsulting.com

```
> db.people.find({Country: "US"})
```

How ?

{_id: 1, Country: **"US"**}

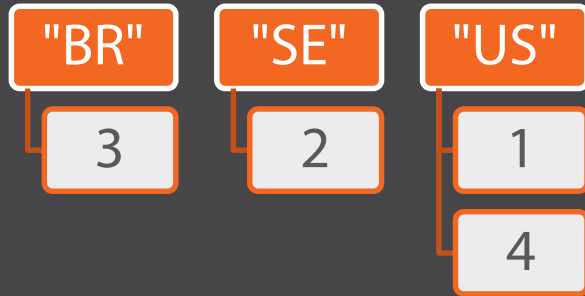
{_id: 2, Country: **"SE"**}

{_id: 3, Country: **"BR"**}

{_id: 4, Country: **"US"**}

{_id: 5, Country: **"US"**}

Index



Documents



```
> db.people.find({Country: "US"})
```



"US"?

"US"?

"US"?

"US"?

{_id: 1, Country: "**US**" }

{_id: 2, Country: "**SE**" }

{_id: 3, Country: "**BR**" }

{_id: 4, Country: "**US**" }

{_id: 5, Cou
}

```
> db.collectionName.ensureIndex({field: direction})
```

Create and index on a collection

Creates index if it doesn't exist

Creates collection if it doesn't exist

Index Key Matchup



{ 'from.country' }



{ 'from.country' : ..., 'from.number' : ... }



{ 'from.country' : ..., 'from.number' : ..., 'time' : ... }



{ ~~'from.country' : ...~~, 'from.number' : ... }



*{ 'from.country' : ..., ~~'from.number' : ...~~, 'time' : ... }

Index
{ Country:1 }

Documents

"BR"
3

"SE"
2

NULL
1
4

{ _id: 1, Country: NULL }

{ _id: 2, Country: "SE" }

{ _id: 3, Country: "BR" }

{ _id: 4 }

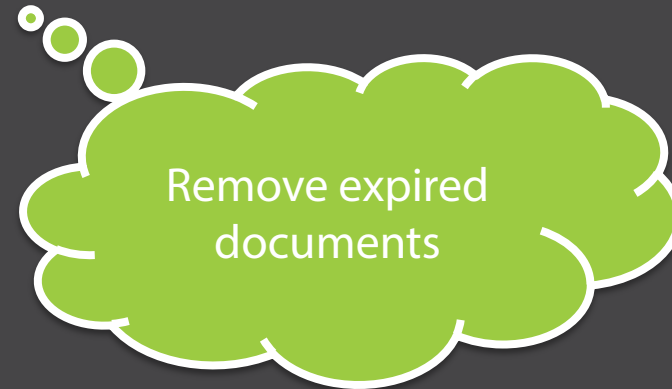
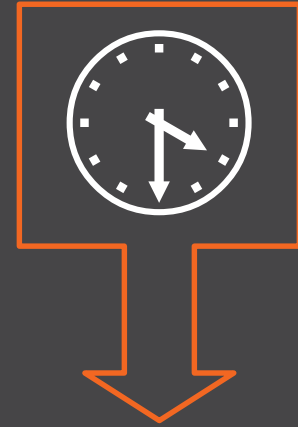
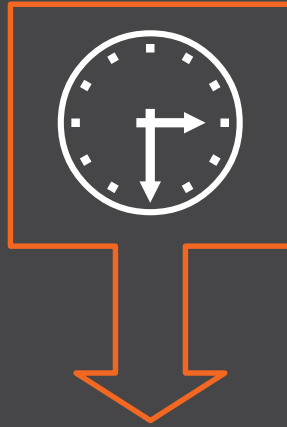
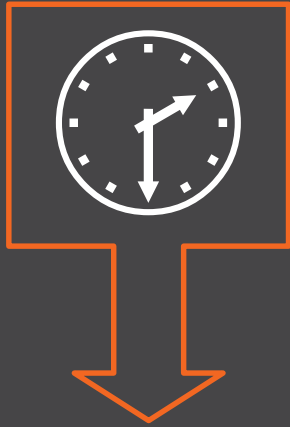
TTL Index

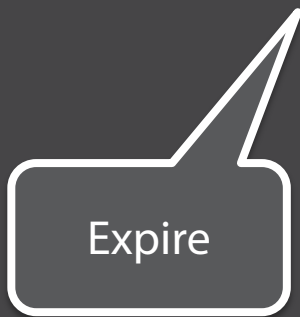
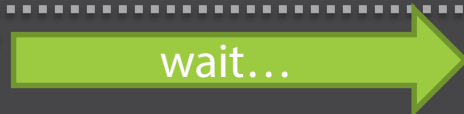
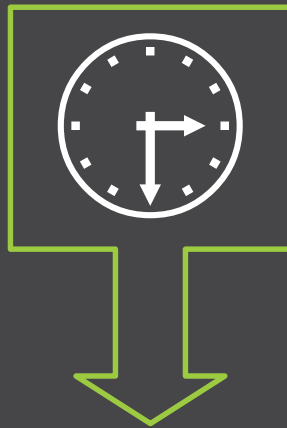
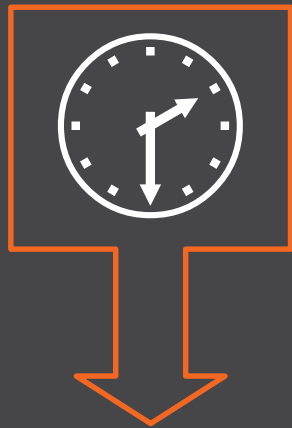
~~"Forever"~~


```
db.myCollection.ensureIndex(  
  { field: [-1|1] },  
  { expireAfterSeconds: 86400 }  
);
```

TTL Index

- Removes documents after they expire
- Define on date-time field
- One TTL index per collection







```
db.myCollection.ensureIndex({...})
```

```
if ( index-exists )  
  done  
else  
  build-index
```

`db.myCollection.ensureIndex({...})`

```
if ( index-exists )  
  done  
else  
  build-index
```

foreground
(blocking)

```
db.myCollection.ensureIndex({...},  
  { background: true } )
```

```
if ( index-exists )  
  done  
else  
  build-index
```

background
(slow-create)

```
db.myCollection.reIndex()
```

Re-Index

Rebuilds all indexes on a collection

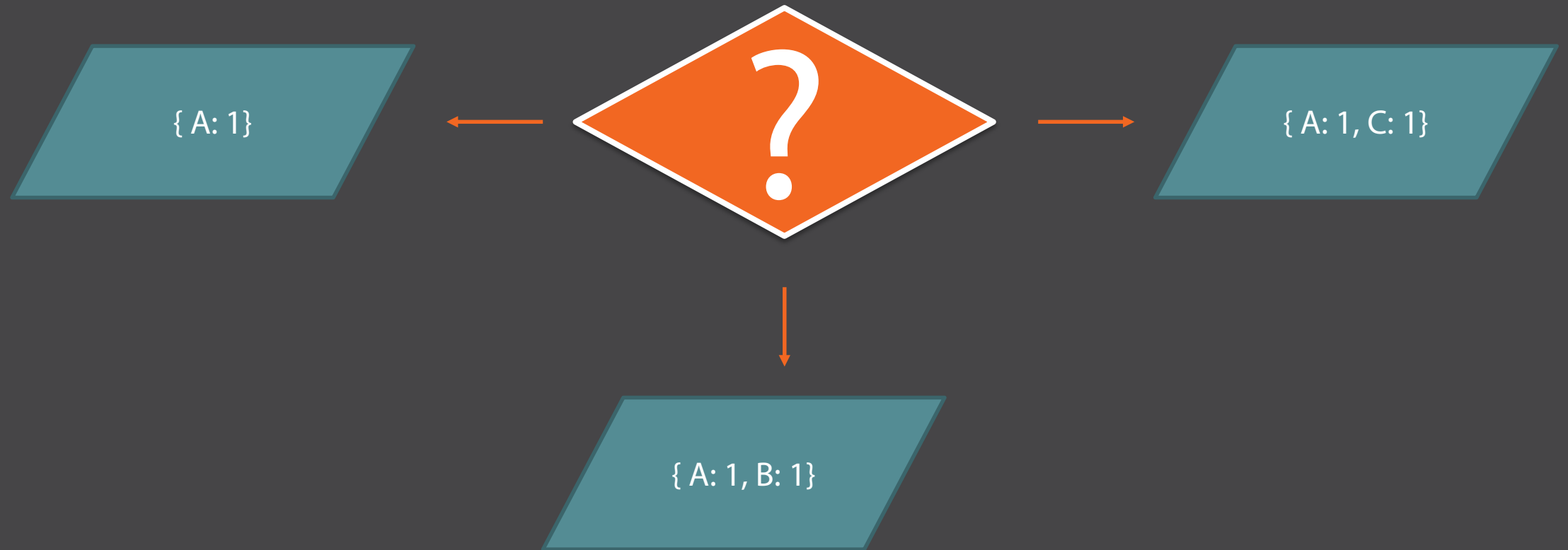

```
db.myCollection.runCommand({ compact: 'myCollection' })
```

Compact

Rebuilds collection

- Defragments documents
- Rebuilds all indexes

```
db.myCollection.find({...})
```



```
db.messages.find( { Country: '01', Time: { $gte: x, $lt: y } } )
```

Country index

?

Time index

$1\text{BN} \div 200$
 $\sim 5\text{M}$

```
db.messages.find( { Country: '01', Time: { $gte: x, $lt: y } } )
```

Country index



Time index

$1\text{BN} \div 10 \div 365$
 $\sim 300\text{K}$

```
db.messages.find( { Country: '01', Time: { $gte: x, $lt: y } } )
```

Country index



Time +
Country
index

$1\text{BN} \div 10 \div 365 \div 200$
 ~ 1500

Index Key selectivity

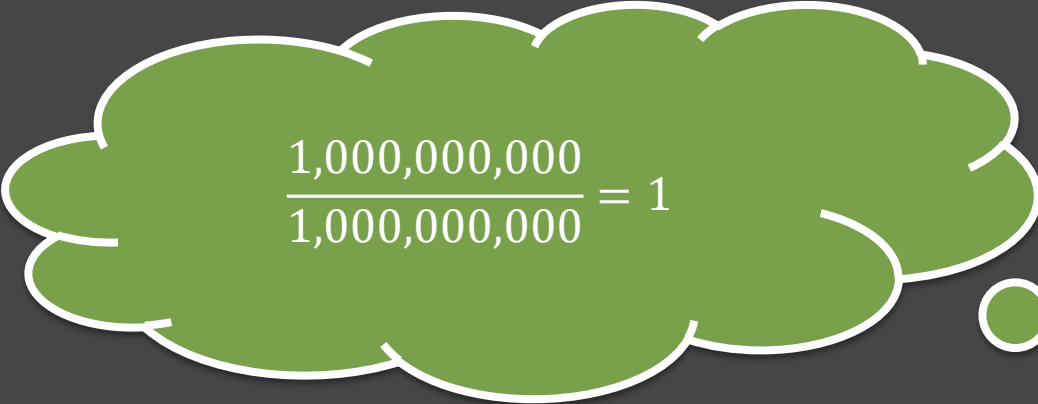
$$= \frac{(\text{key value count})}{(\text{document count})}$$

Low key value count

$$\frac{1,000,000,000}{2} = 500,000,000$$

$$= \frac{(\text{document count})}{(\text{key value count})}$$

High (or unique) key values

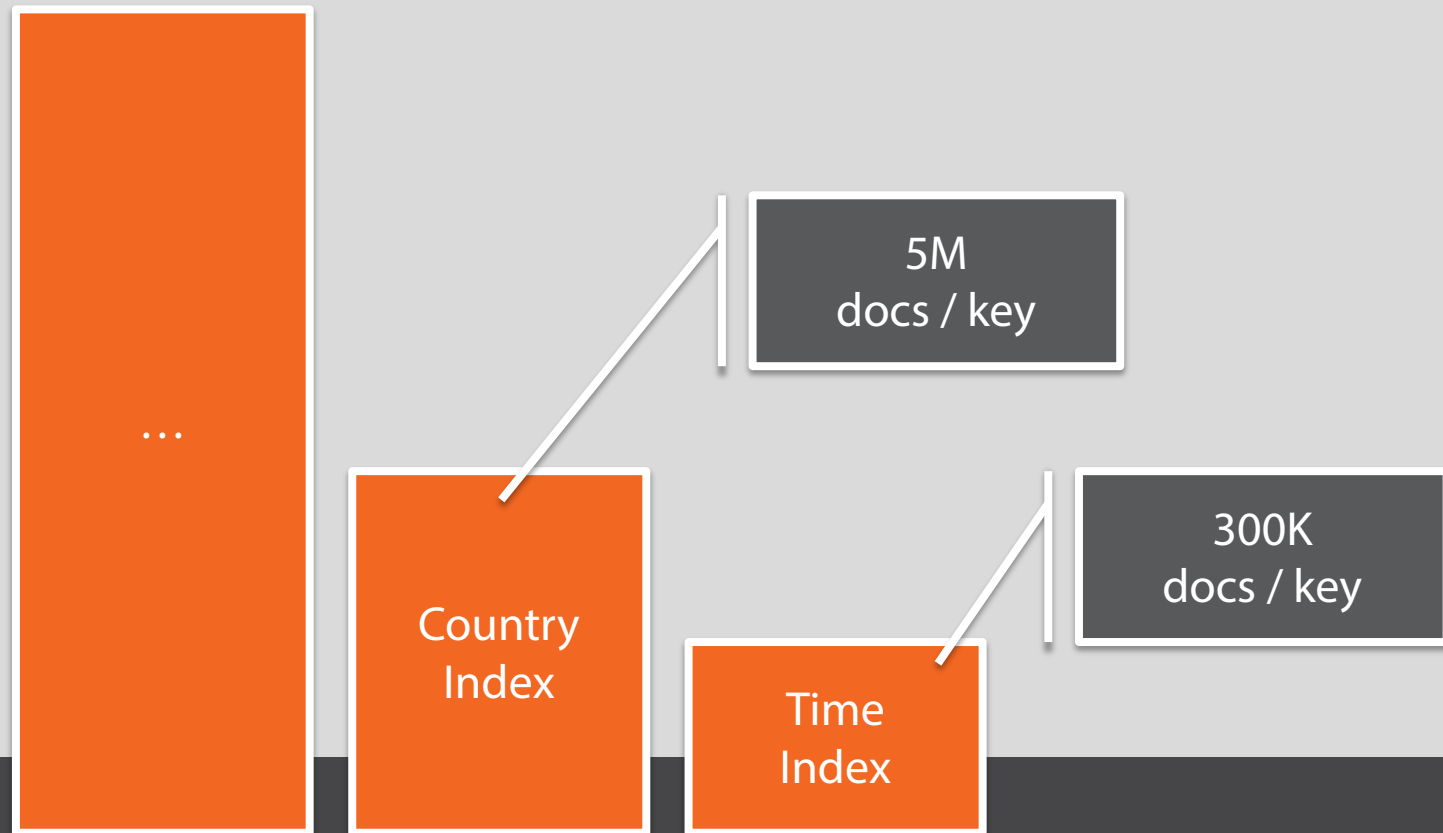

$$\frac{1,000,000,000}{1,000,000,000} = 1$$




$$= \frac{(\text{document count})}{(\text{key value count})}$$

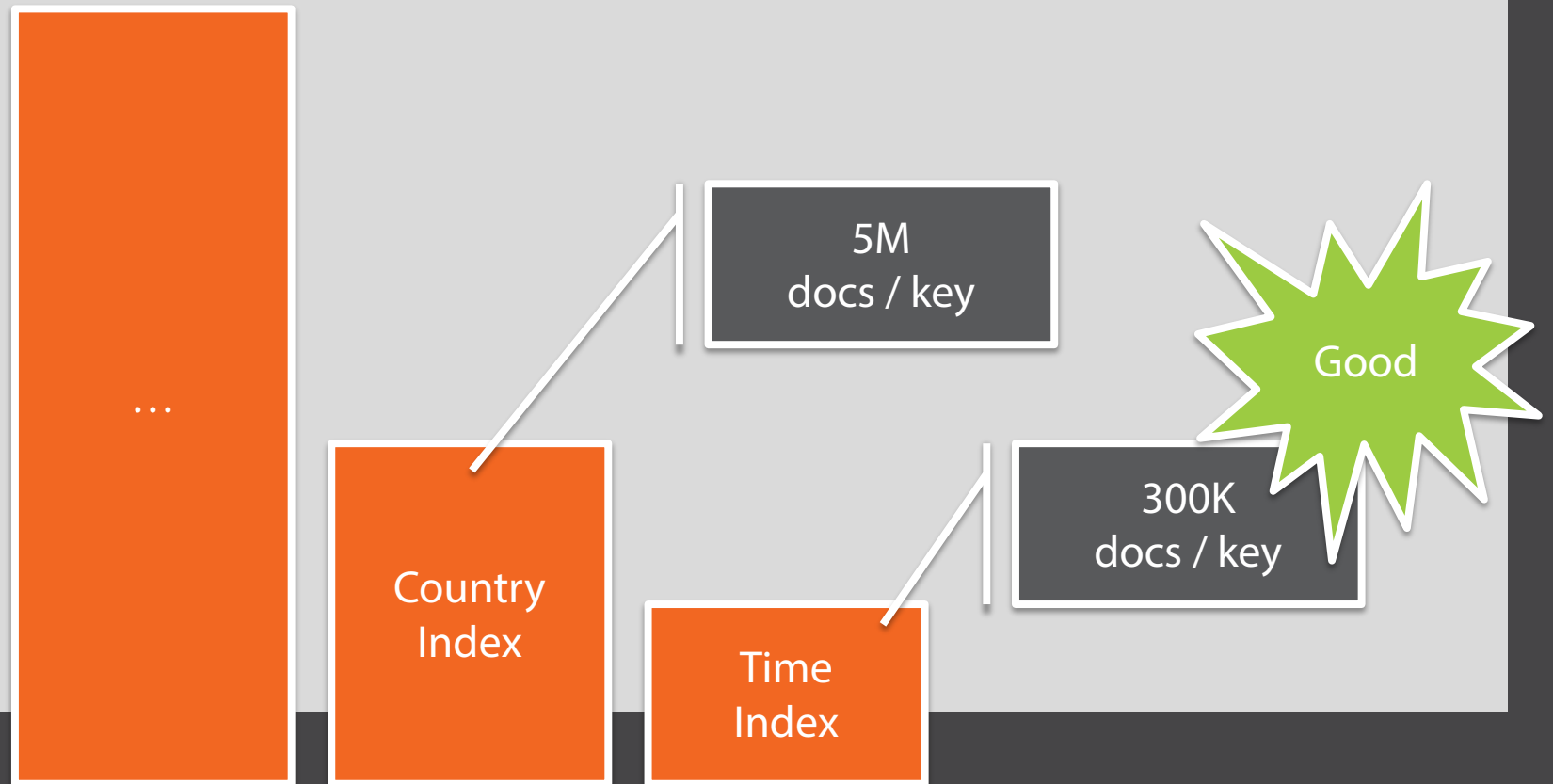
Statistics?

```
db.message.find({Country: c1, Time: {$gt: d1, $lt: d2 }})
```



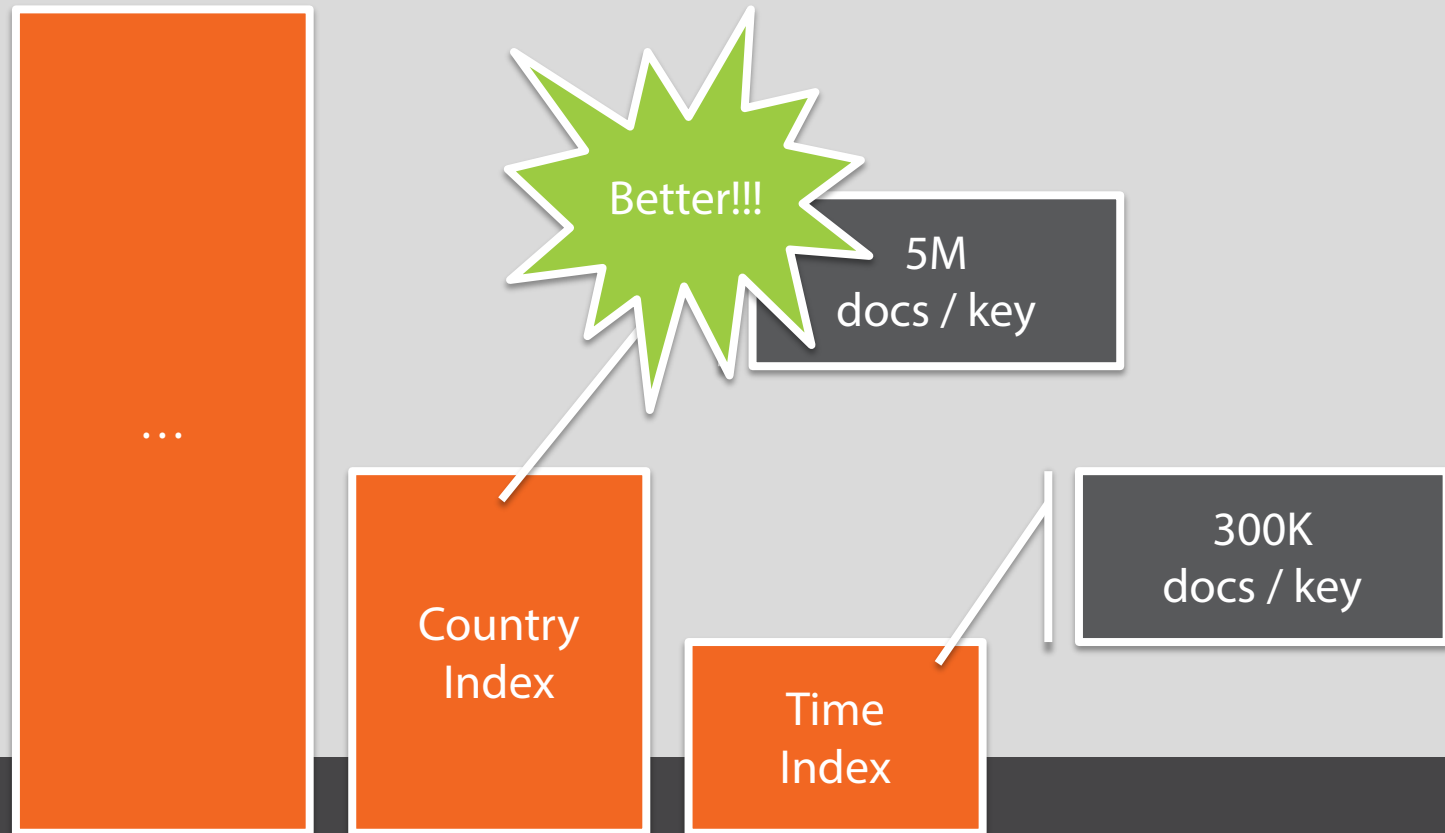
Statistics?

```
db.message.find({Country: c1, Time: {$gt: d1, $lt: d2 }})
```



Statistics?

```
db.message.find({Country: c1, Time: {$gt: d1, $lt: d2 }})
```



Index Considerations

Speeds-up
Queries

Minimize
Index Size

Minimize index
count

Maximize
Selectivity



Under
Indexing

Over
Indexing

```
{ f1: 'abcdefg', f2: [1,2,3,4,...752]}
```



```
db.myCollection.ensureIndex({ f1: 1, f2: 1 ...})
```

Key Value Size Limit

- 1024 bytes
- Combined value data size of all fields in index

```
> db.system.indexes.find()  
{ ..., "name" : "operationTime_1_countryOfOrigin_2",... }
```

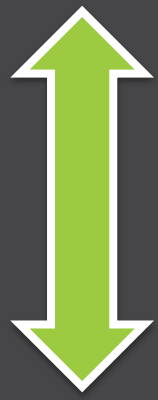
```
db.myCollection.ensureIndex({  
  operationTime: 1,  
  countryOfOrigin: 1 ...})
```



Key Name Size Limit

- 128 bytes

```
db.myCollection.ensureIndex({
```



```
  f1: 1,  
  f2: 1,  
  ...  
  f31: 1})
```

Keys Per Index

- 31


```
db.myCollection.ensureIndex({ ...1 })
```

```
db.myCollection.ensureIndex({ ...2 })
```

```
...
```

```
db.myCollection.ensureIndex({ ...64 })
```



Indexes Per Collection

- 31 indexes per collection
- (Am I over-indexing?)

marv

USE

INDEXES!

Summary

