DuyHai DOAN, Technical Advocate

Shameless self-promotion

Duy Hai DOAN

- , ,
- - (Achilles,) :
-

☞ **duy_hai.doan@datastax.com**

- production

Datastax

-

Agenda

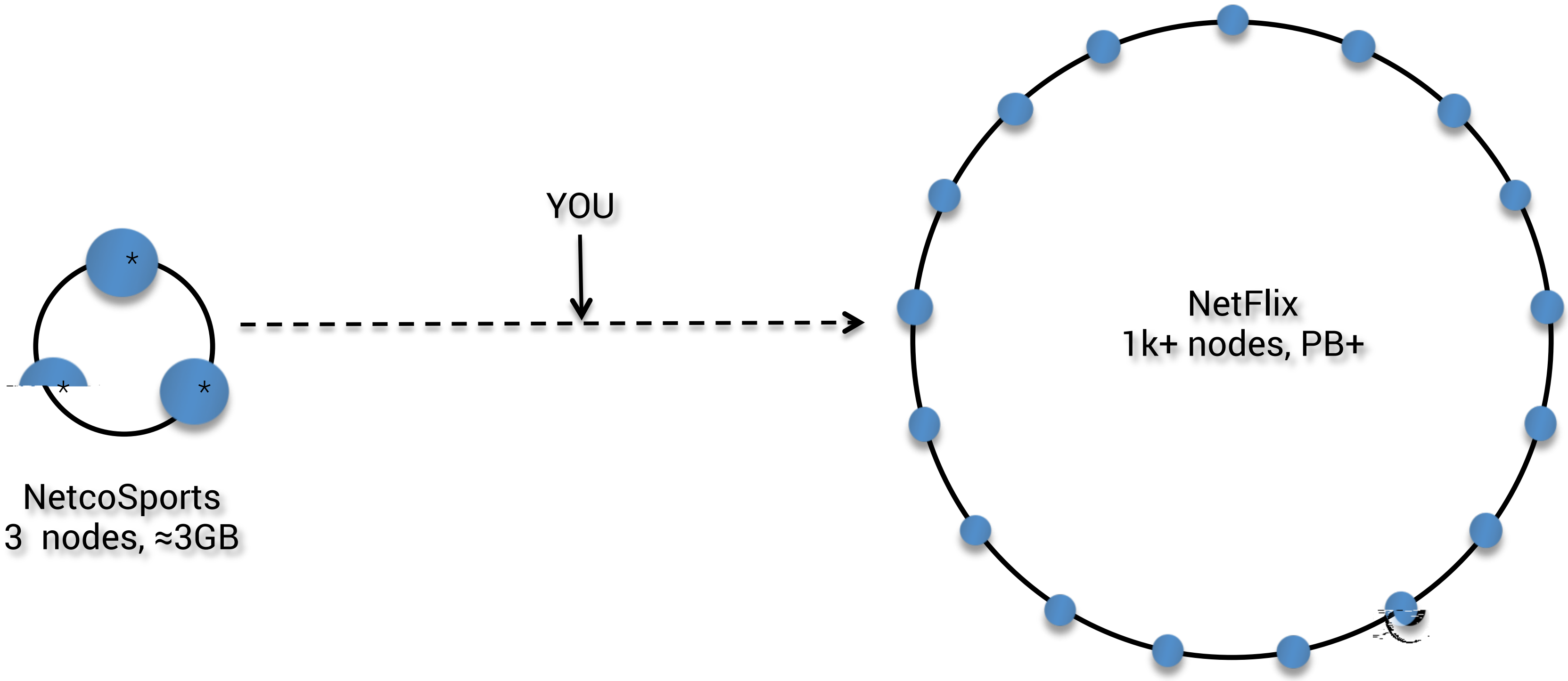
Architecture

-
-

Data model

- ( ),
- (, , ,)
-

Cassandra 5 key facts



Cassandra 5 key facts

(≈100% -)

- ( )

Cassandra 5 key facts

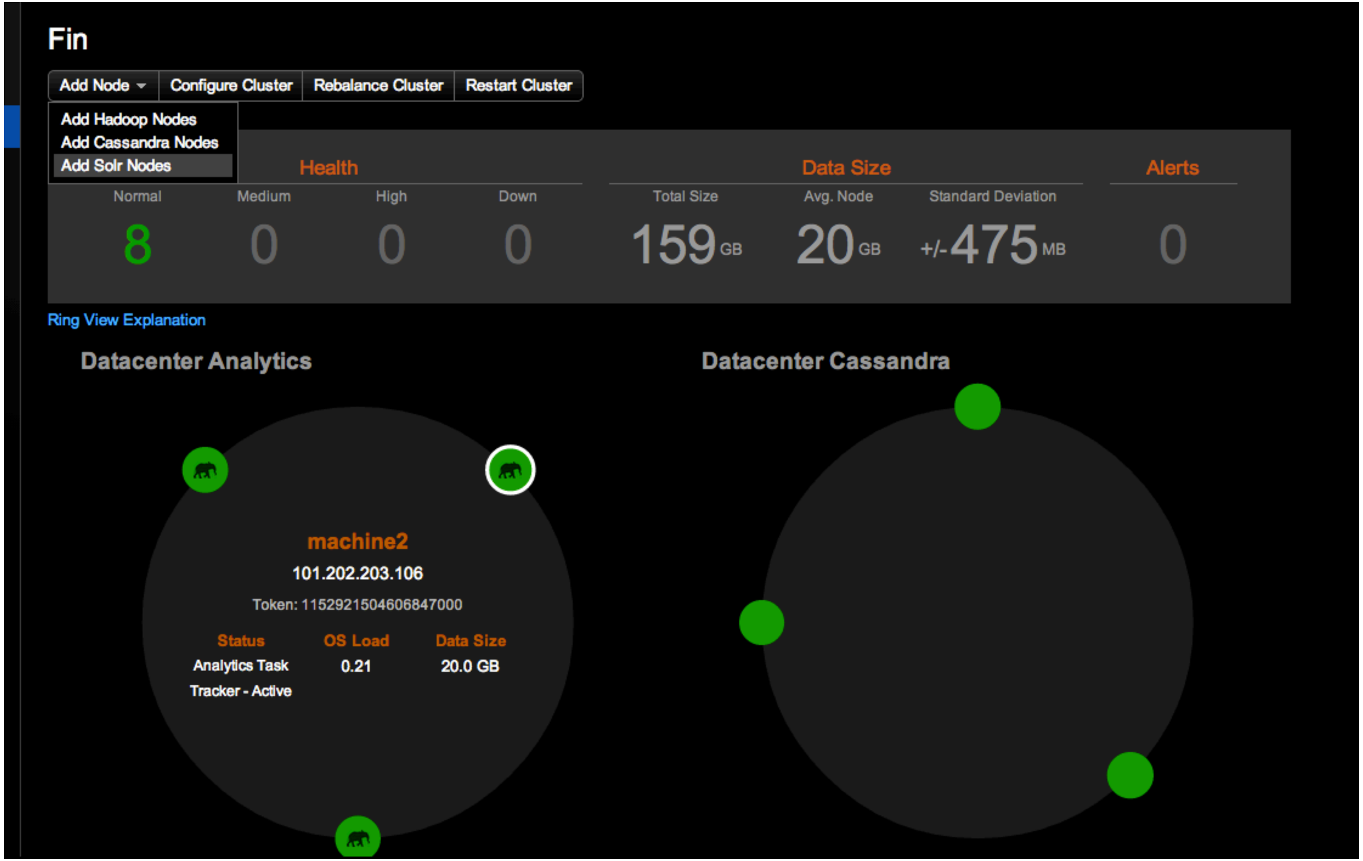
-

- - - - ( )
- -
- /

Cassandra 5 key facts

- 1 = 1 + 1
-
-

Cassandra 5 key facts



Fin

Health				Data Size			Alerts
Normal	Medium	High	Down	Total Size	Avg. Node	Standard Deviation	
8	0	0	0	159 GB	20 GB	+/-475 MB	0

machine2

101.202.203.106

Token: 1152921504606847000

Status	OS Load	Data Size
Analytics Task Tracker - Active	0.21	20.0 GB

Cassandra 5 key facts

- Cassandra + Spark = awesome !
- realtime streaming

Cassandra architecture

Cluster

Replication

Cassandra architecture

- DynamoDB
- masterless

  –

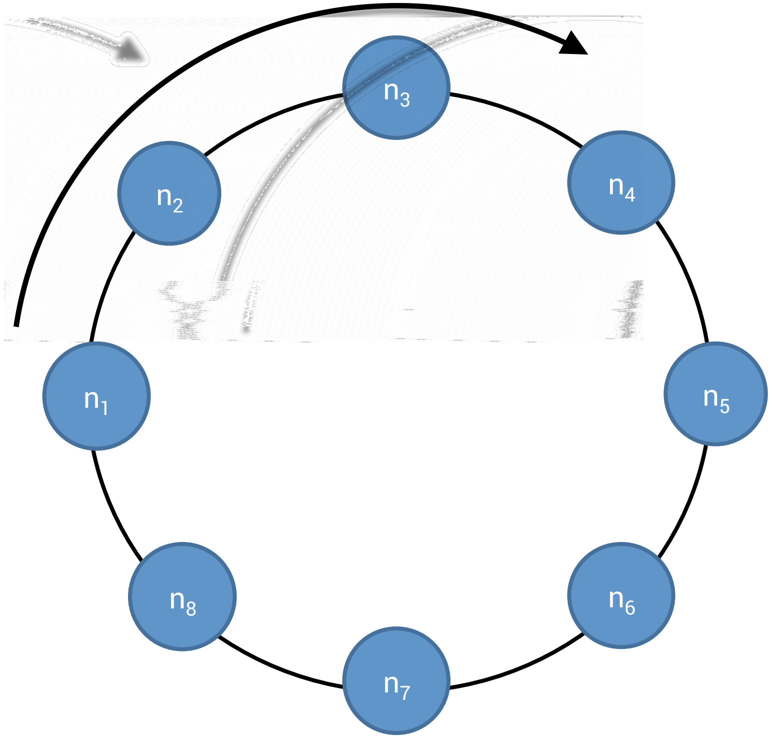
- Big Table
- /

Data distribution

: #partition → token = (#p)

: - ,

= $(2^{64}/2)$



Token Ranges

A: 0, /8

B: /8, 2 /8

C: 2 /8, 3 /8

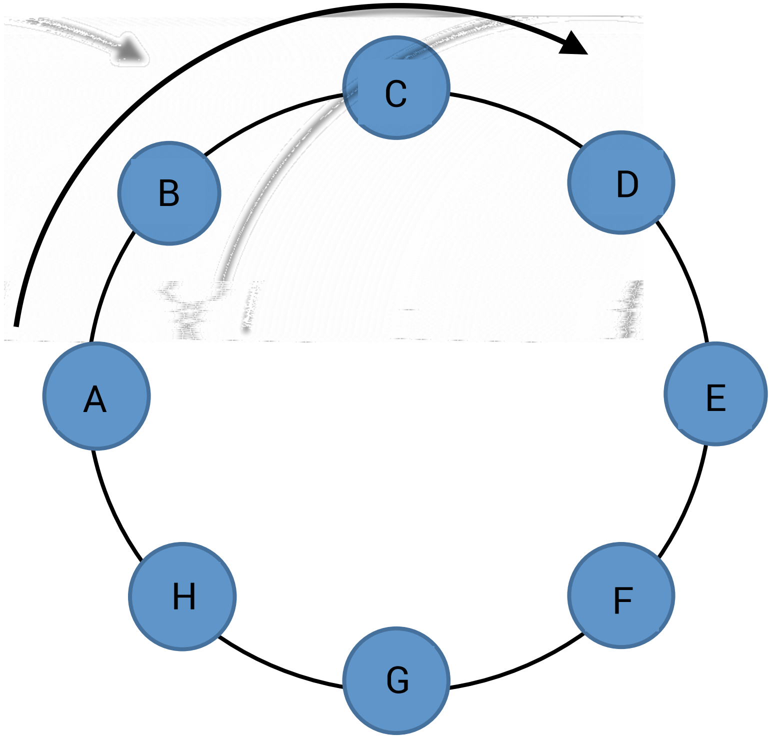
D: 3 /8, 4 /8

E: 4 /8, 5 /8

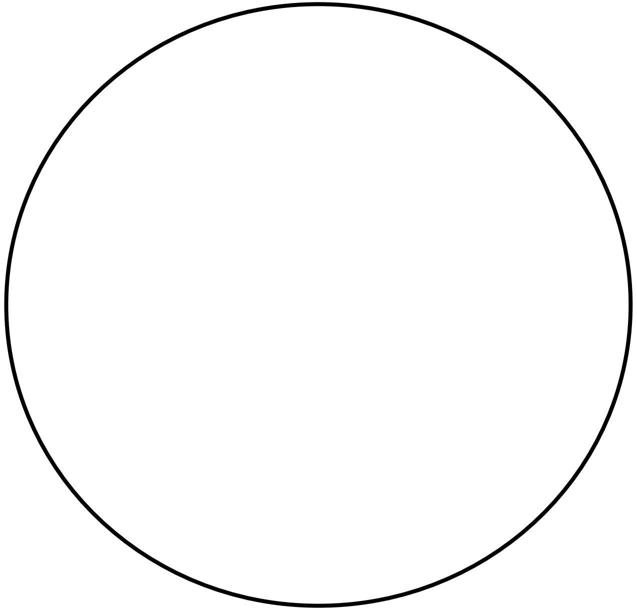
F: 5 /8, 6 /8

G: 6 /8, 7 /8

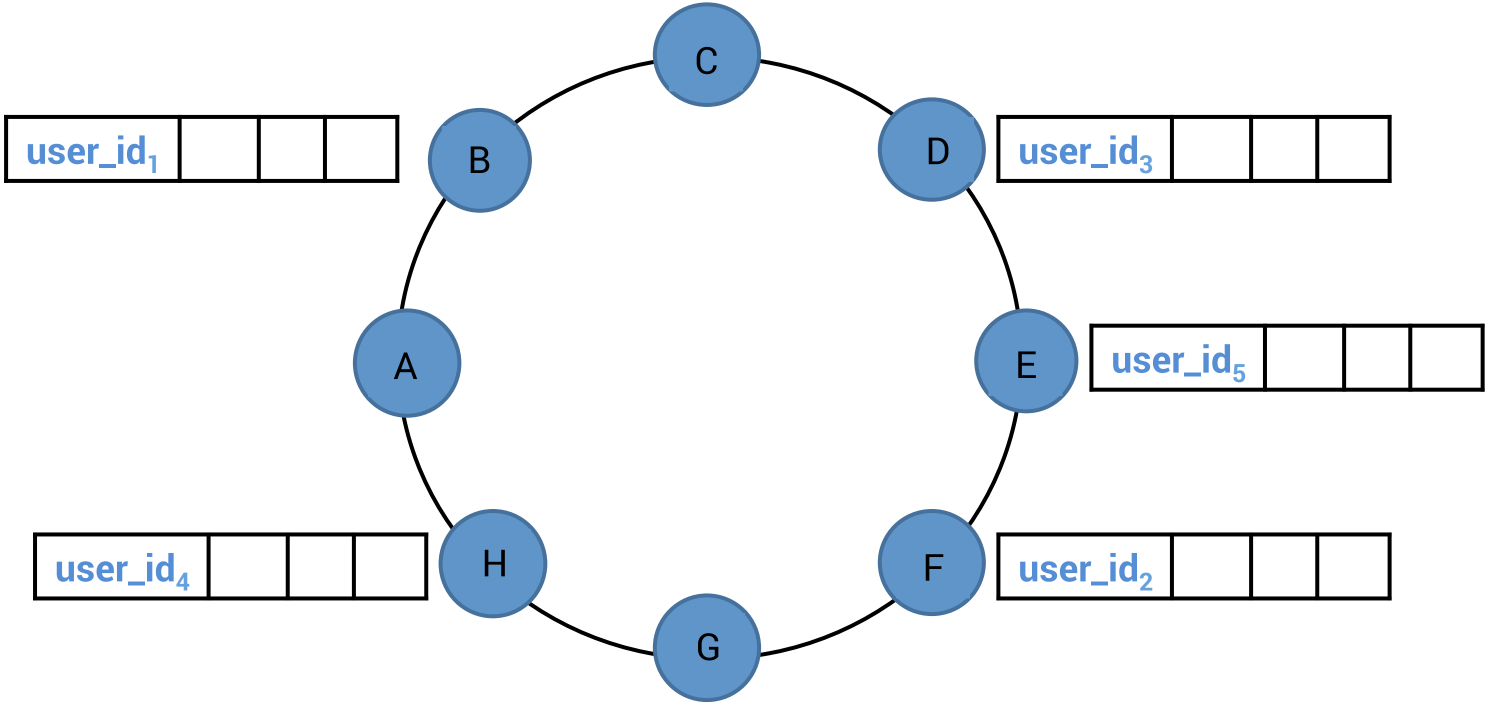
H: 7 /8,



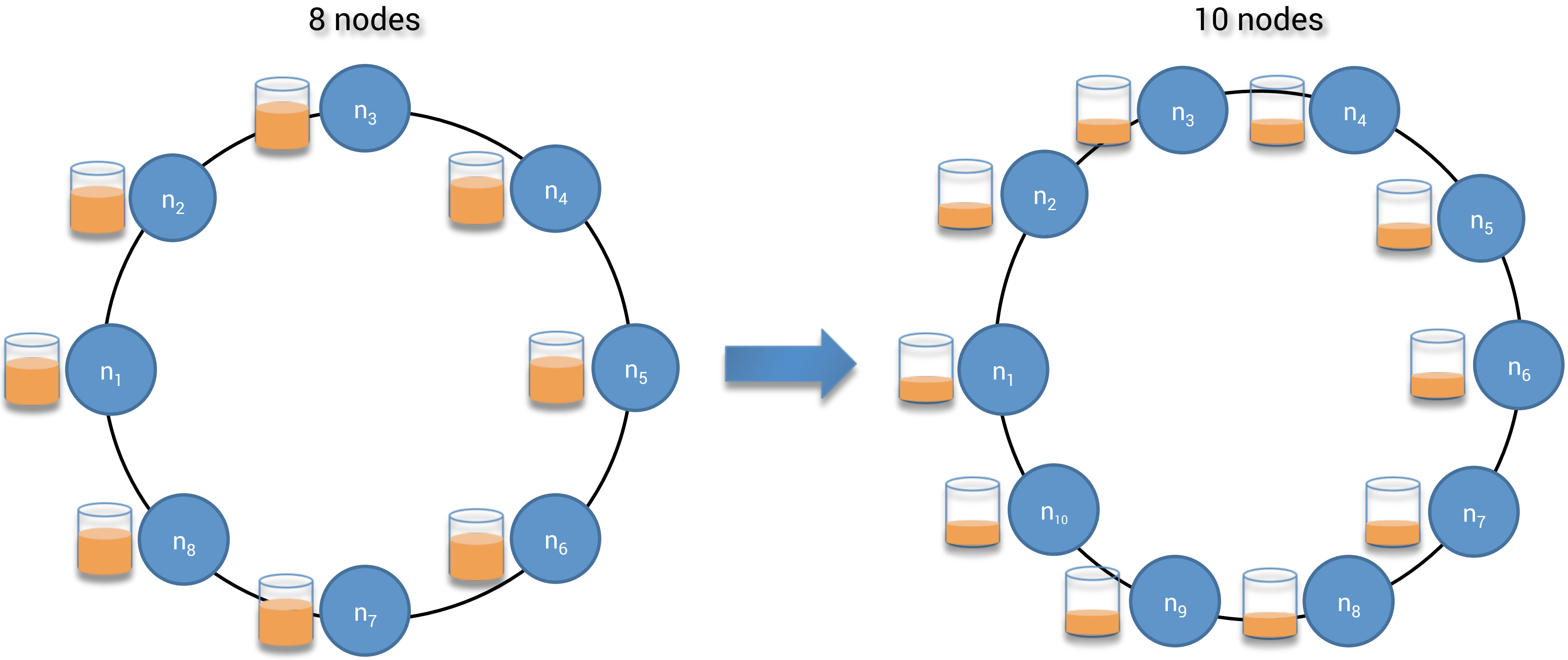
Distributed Table



Distributed Table

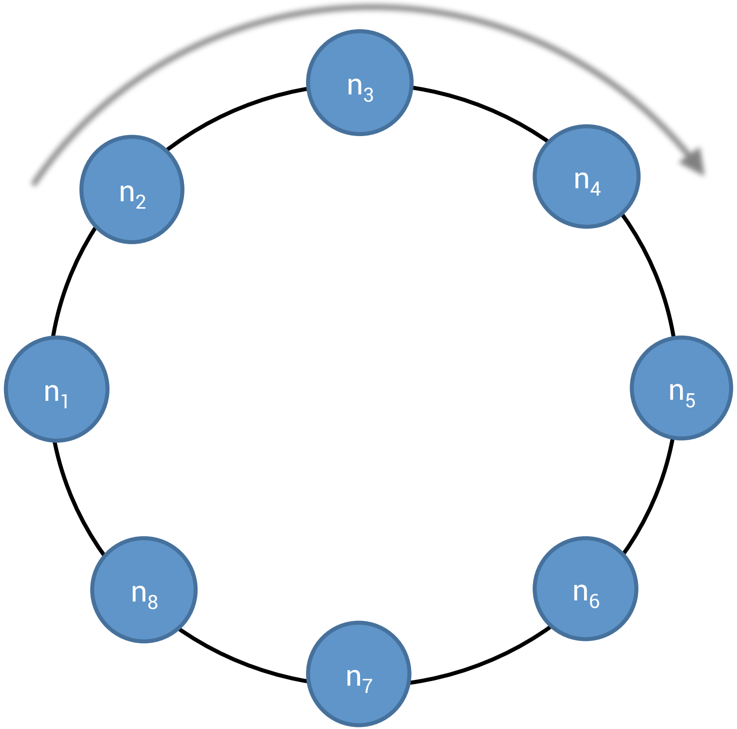


Linear scalability



Failure tolerance

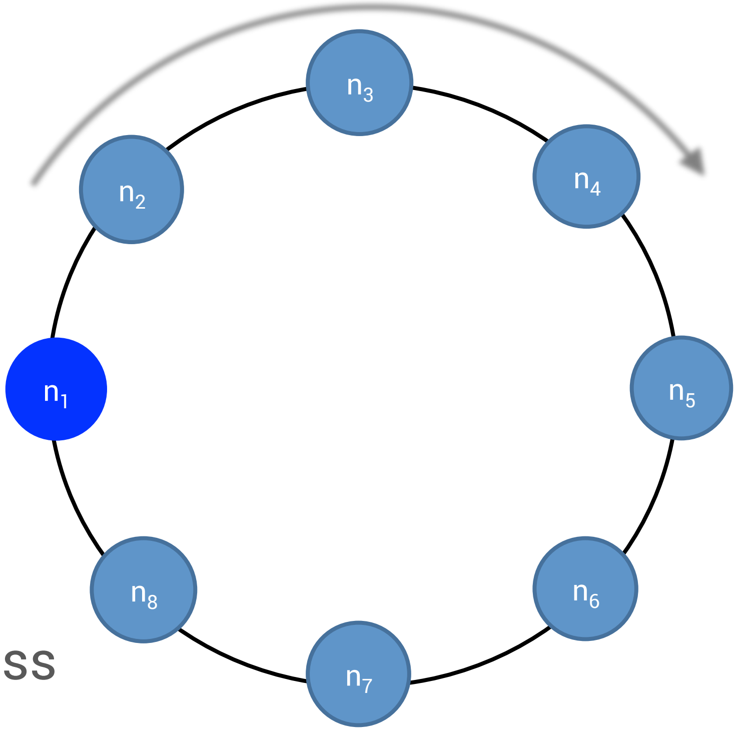
(RF) = 3



Coordinator node

(/)

Coordinator



coordinator →masterless

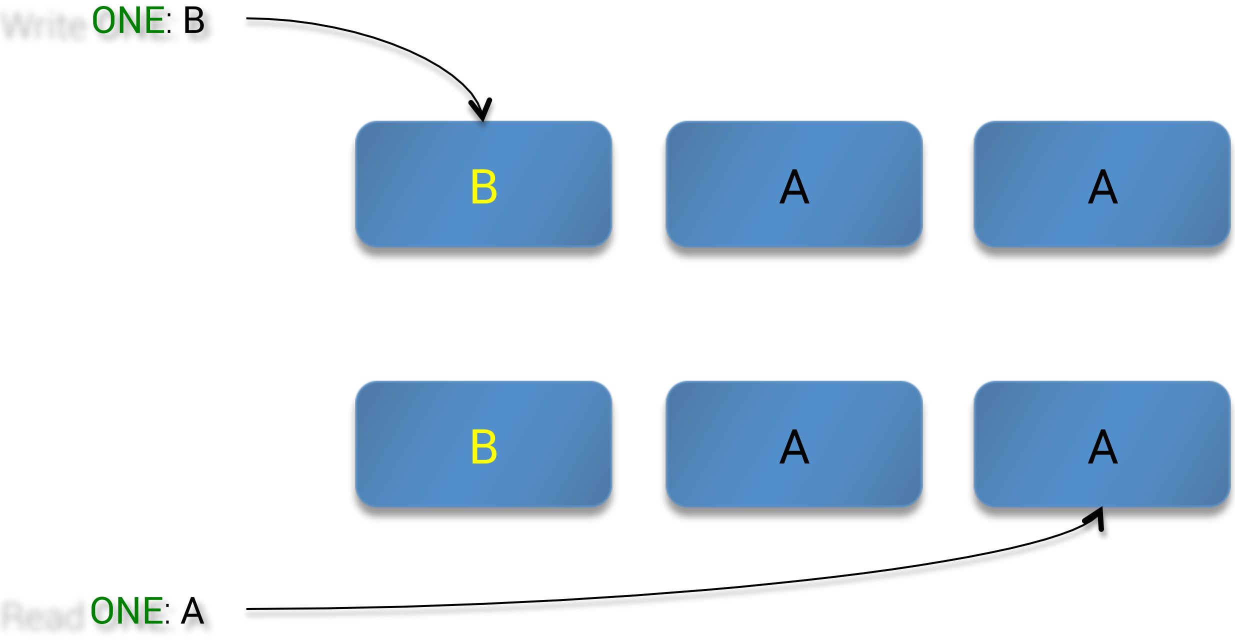
Consistency

- ONE
- QUORUM (strict majority . . . RF)
- ALL

read & write

Consistency in action

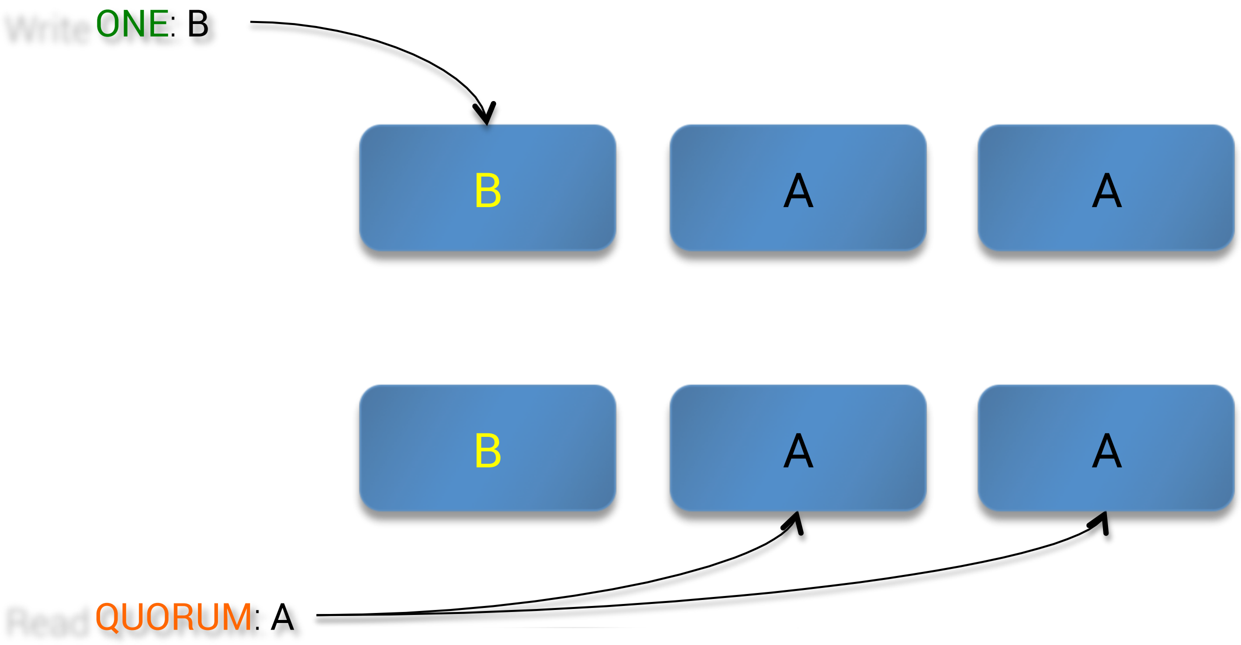
= 3, ONE, ONE



:

Consistency in action

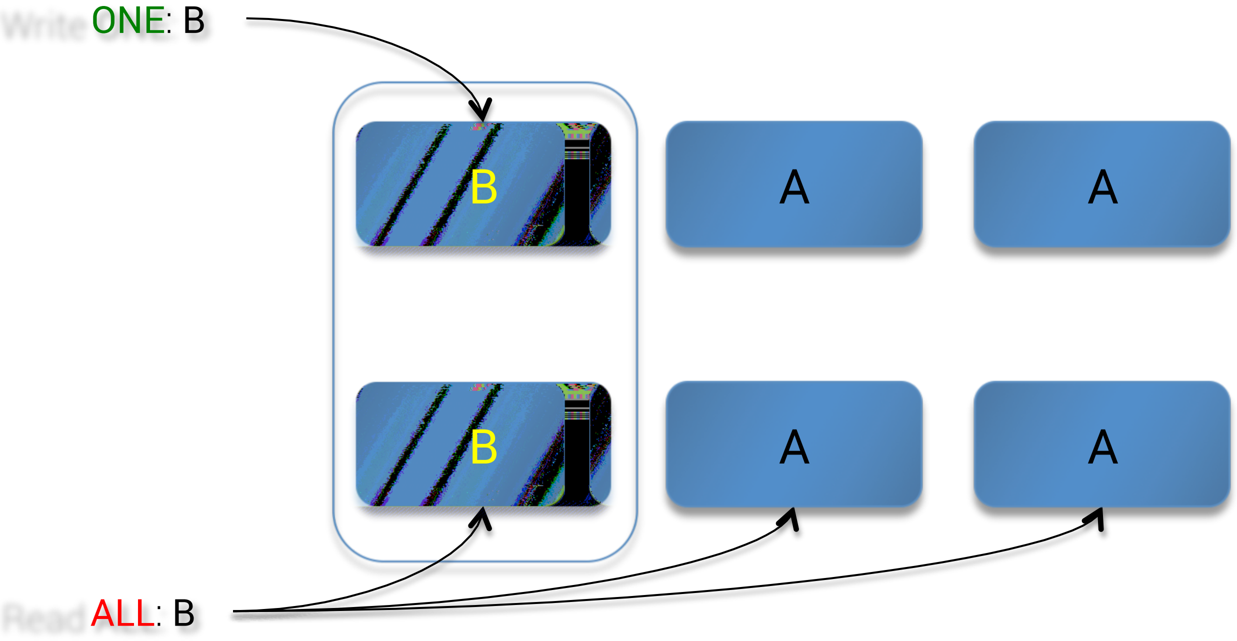
= 3, ONE, QUORUM



:

Consistency in action

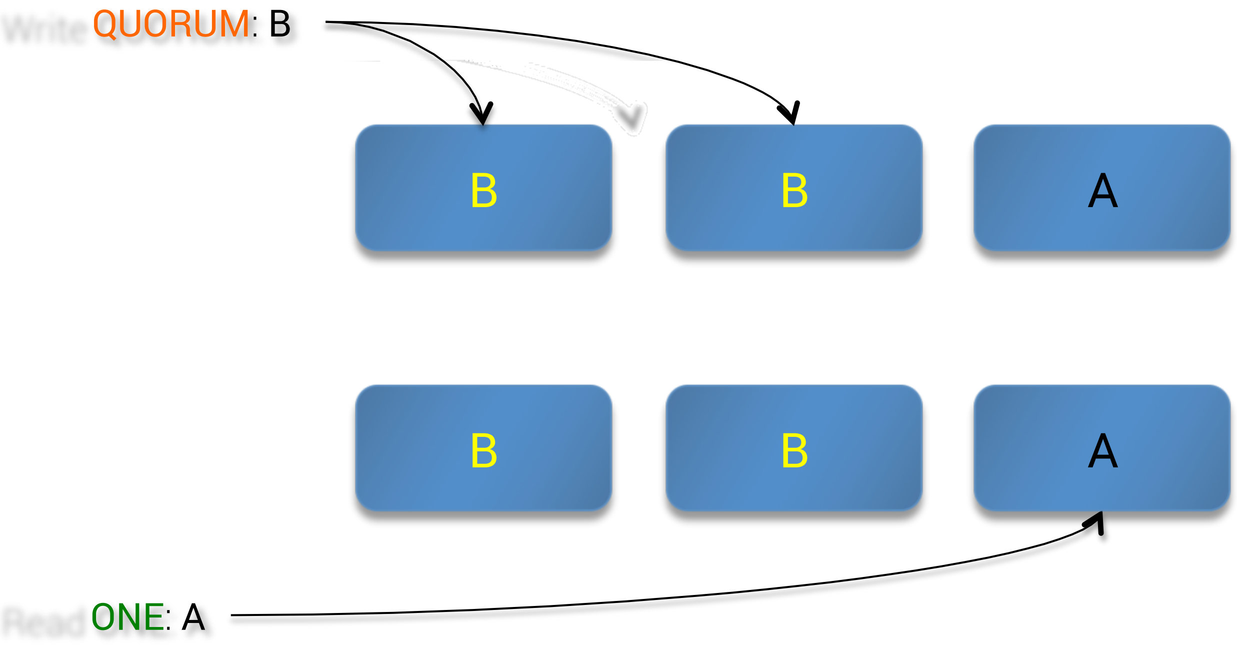
= 3, ONE, ALL



:

Consistency in action

= 3, QUORUM, ONE



Write QUORUM: B

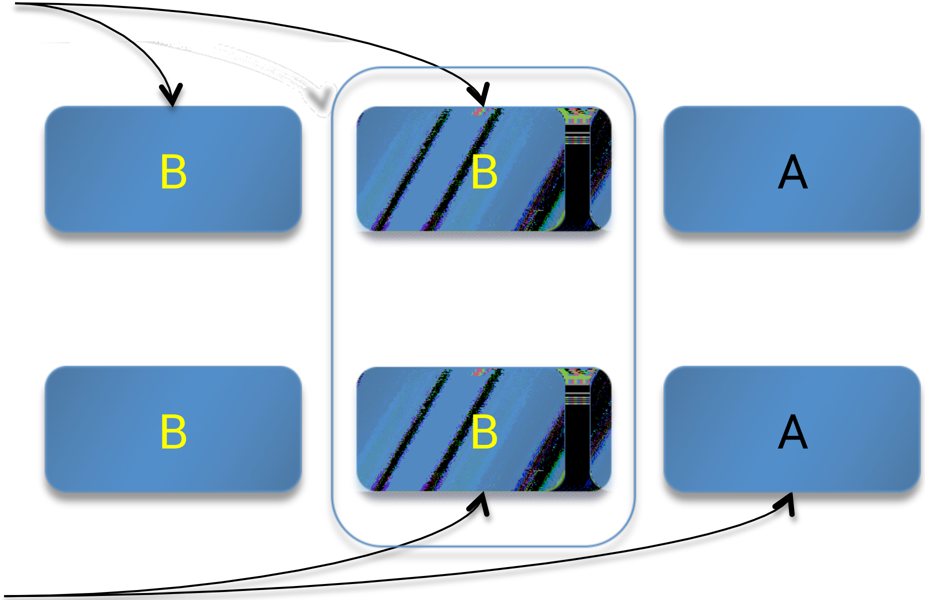
Read ONE: A

:

Consistency in action

= 3, QUORUM, QUORUM

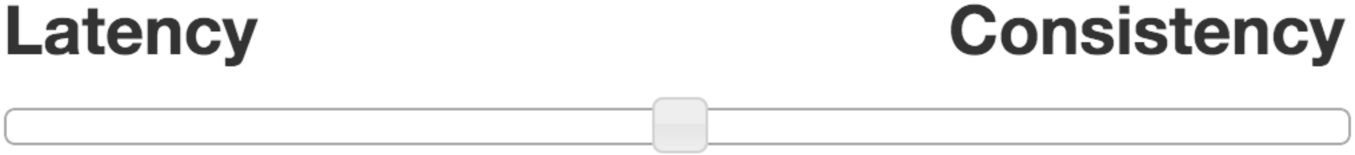
Write QUORUM: B



Read QUORUM: B

:

Consistency trade-off



Consistency level

ONE

Fast, may not read latest written value

Consistency level

QUORUM

Strict majority w.r.t. **R**eplication **F**actor

Good balance

Consistency level

ALL

Paranoid

Slow, no high availability

Consistency summary

$\text{ONE}_{\text{Read}} + \text{ONE}_{\text{Write}}$

☞ available / (N-1)

$\text{QUORUM}_{\text{Read}} + \text{QUORUM}_{\text{Write}}$

☞ available / 1+

Q & R

Data model

Last Write Win

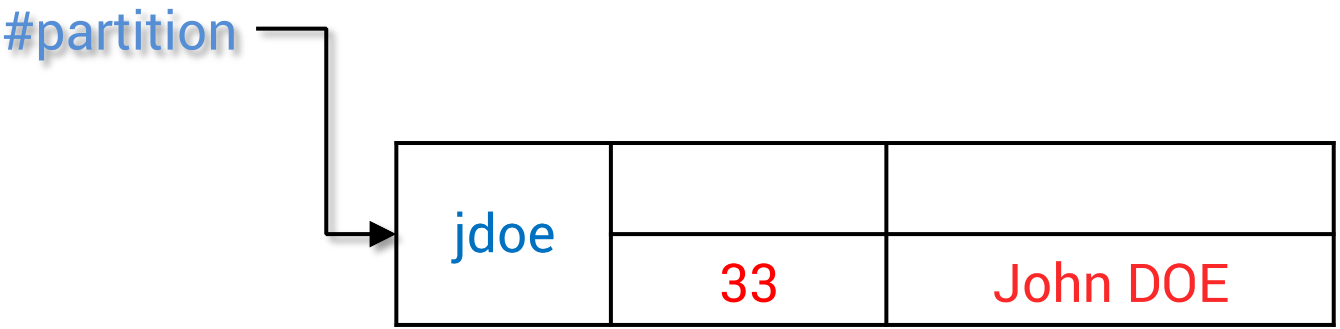
CQL basics

Clustered tables

Lightweight transactions

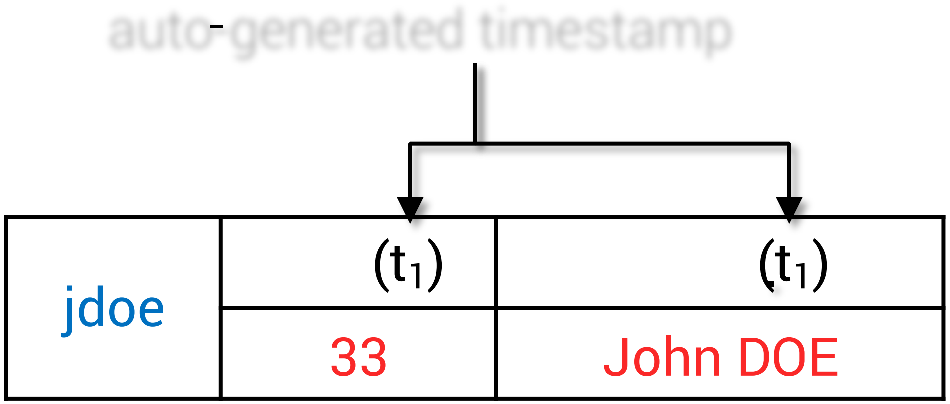
Last Write Win (LWW)

```
(login, , ) (jdoe, John DOE, 33);
```



Last Write Win (LWW)

```
(login, , ) (jdoe, John DOE, 33);
```



jdoe	(t_1)	(t_1)
	33	John DOE

Last Write Win (LWW)

= 34 login = jdoe;

SSTable$_1$

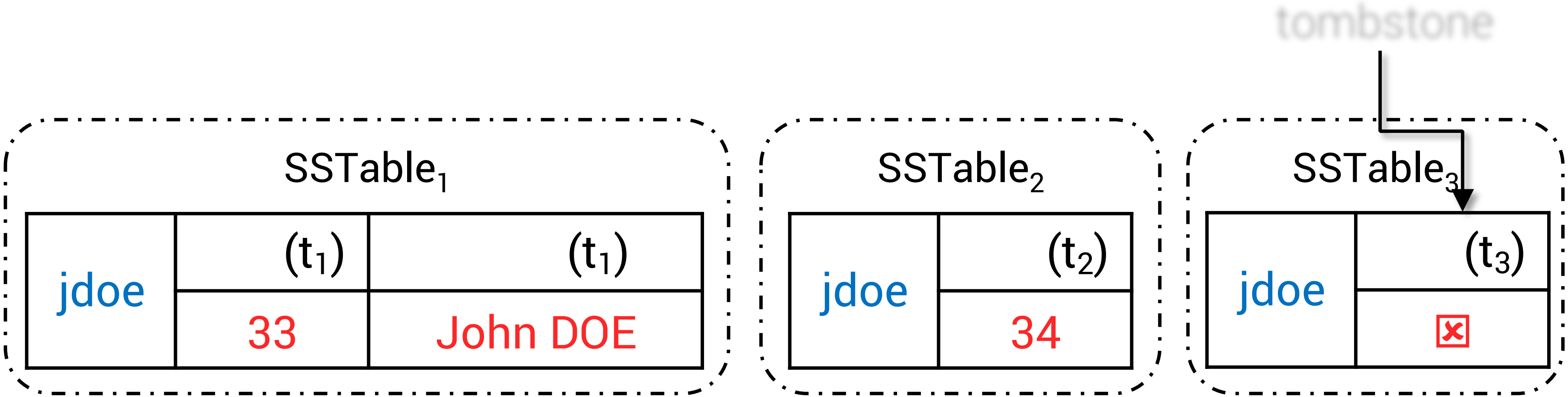
	(t_1)	(t_1)
jdoe	33	John DOE

SSTable$_2$

	(t_2)
jdoe	34

Last Write Win (LWW)

```
login = jdoe;
```



tombstone

SSTable$_1$

jdoe	(t_1)	(t_1)
	33	John DOE

SSTable$_2$

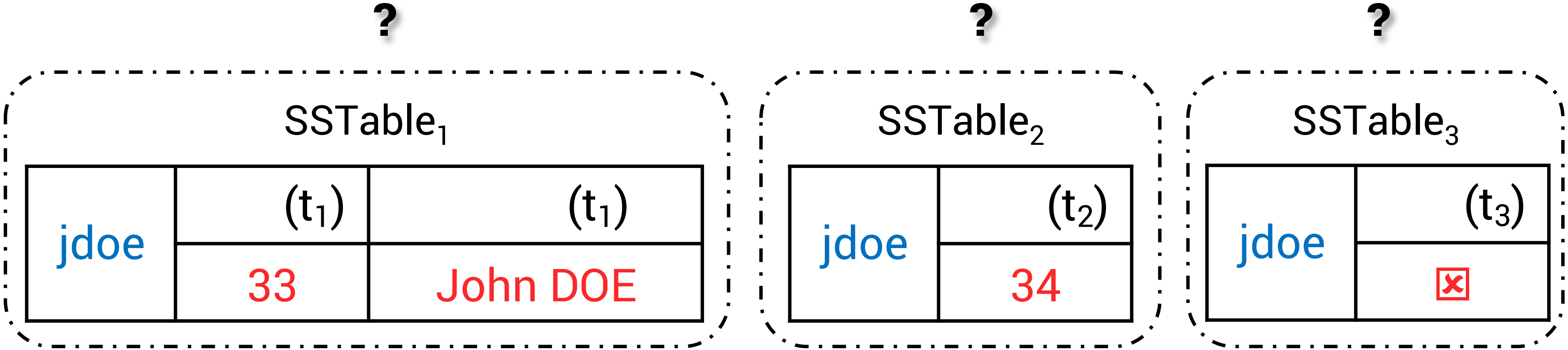
jdoe	(t_2)
	34

SSTable$_3$

jdoe	(t_3)
	☒

Last Write Win (LWW)

login = jdoe;



?

SSTable$_1$

jdoe	(t_1)	(t_1)
	33	John DOE

?

SSTable$_2$

jdoe	(t_2)
	34

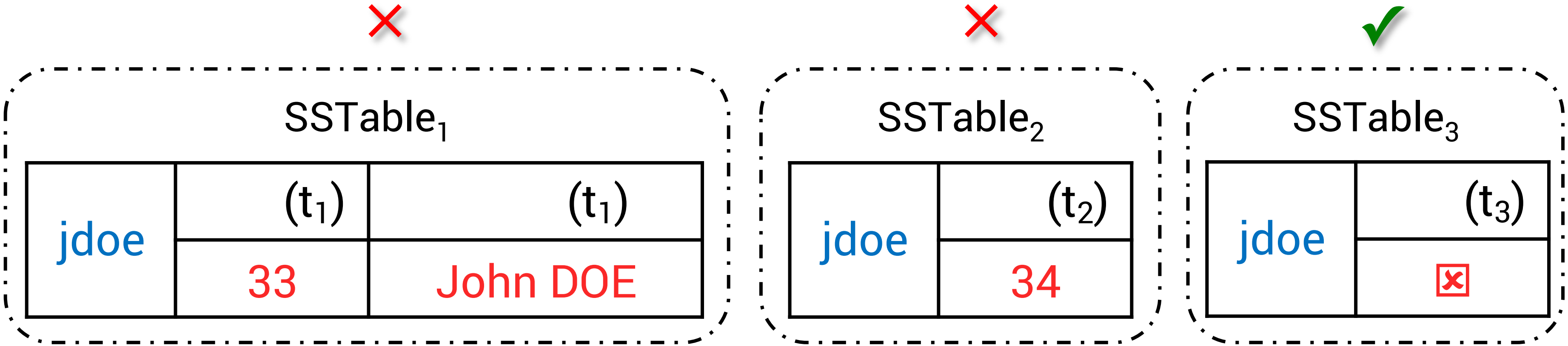
?

SSTable$_3$

jdoe	(t_3)
	☒

Last Write Win (LWW)

login = jdoe;



✗ SSTable$_1$

jdoe	(t$_1$)	(t$_1$)
	33	John DOE

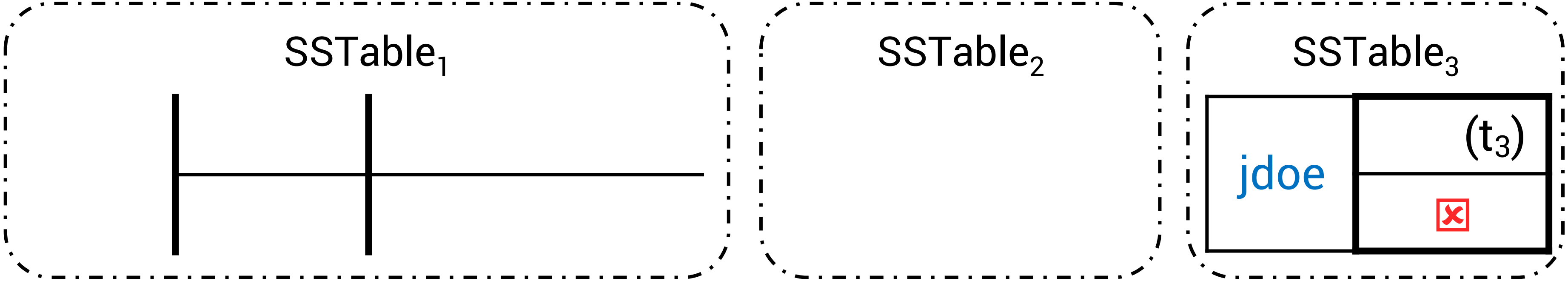
✗ SSTable$_2$

jdoe	(t$_2$)
	34

✓ SSTable$_3$

jdoe	(t$_3$)
	☒

Compaction



SSTable$_1$

SSTable$_2$

SSTable$_3$

jdoe

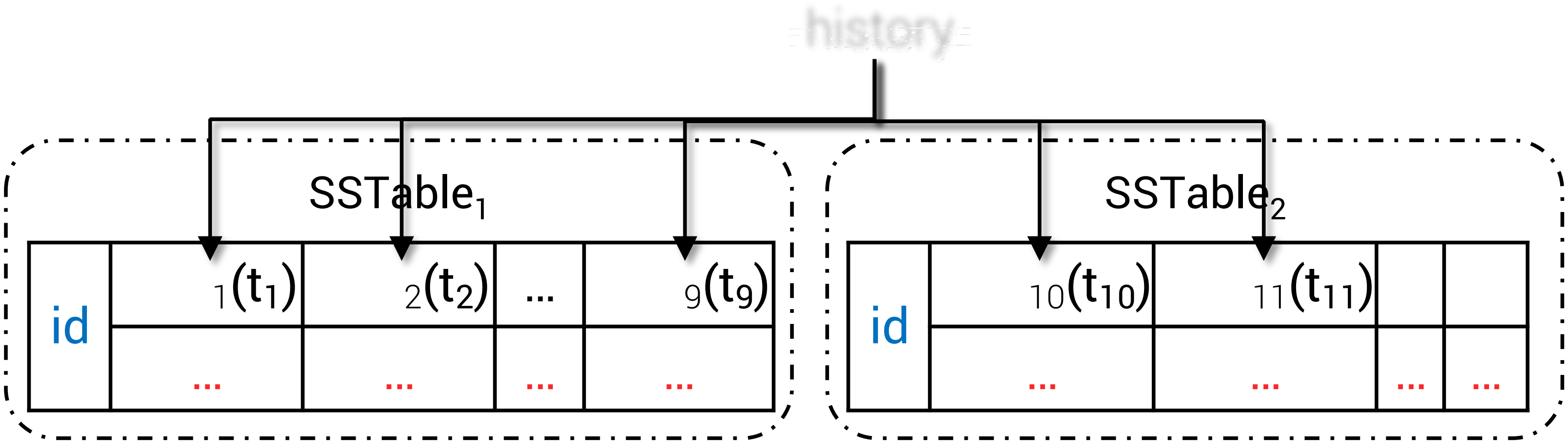
(t$_3$)

☒

Historical data

?

- do not
- ☞ time-series



CRUD operations

```
(login,          ,      )          ( jdoe , John DOE , 33);
```

```
= 34          login = jdoe;
```

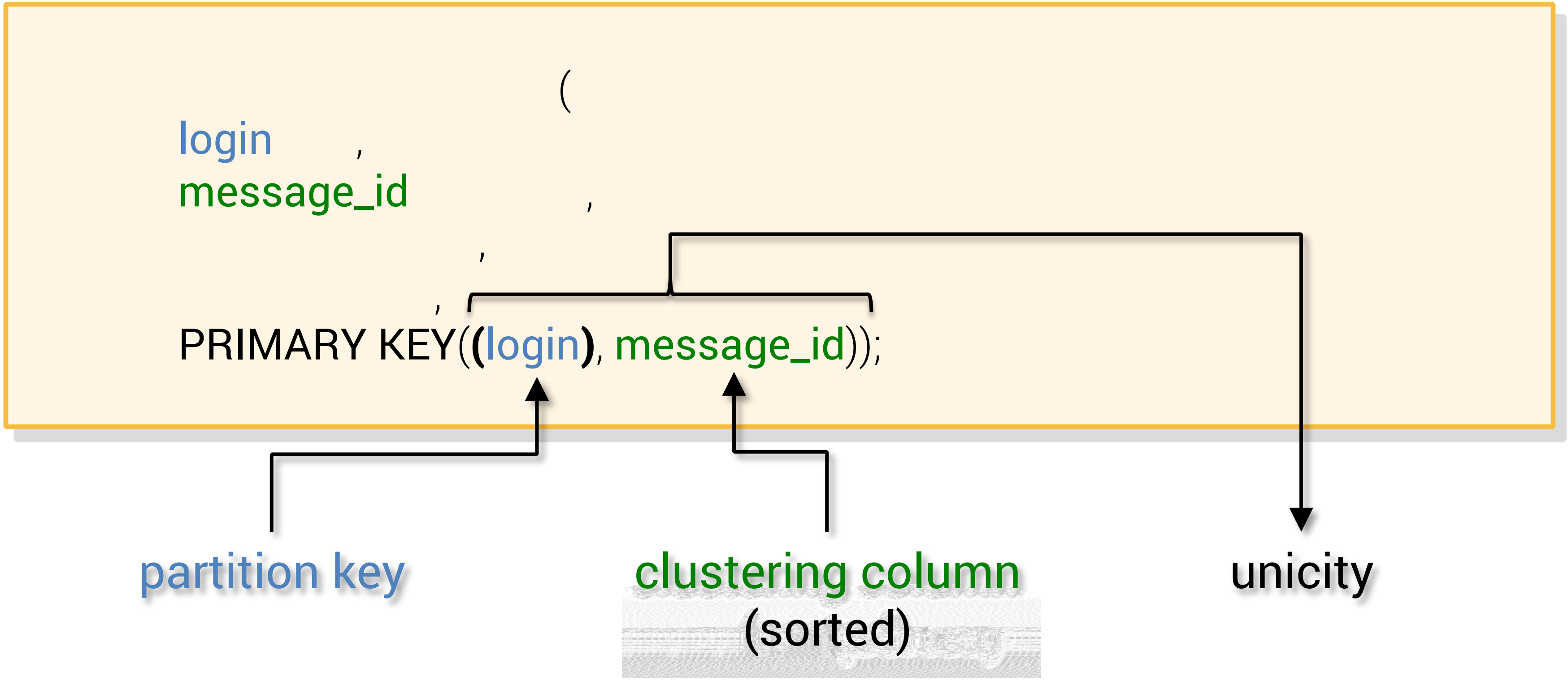
```
login = jdoe;
```

```
login = jdoe;
```

Simple Table

```
                    (
      login
```

Clustered table (1 – N)



```
                        (
login      ,
message_id                 ,
                   ,
                 ,
PRIMARY KEY((login), message_id));
```

partition key

clustering column
(sorted)

unicity

Queries

(   )

```
      *                        login = jdoe
message_id =  2014-09-25 16:00:00 ;
```

```
      *                        login = jdoe
message_id <=  2014-09-25 16:00:00
message_id >=  2014-09-20 16:00:00 ;
```

Queries

(#partition)

```
*                 message_id =  2014-09-25 16:00:00 ; ❌
```

(#partition)

```
*
message_id <=  2014-09-25 16:00:00    ❌
message_id >=  2014-09-20 16:00:00 ;
```

Queries

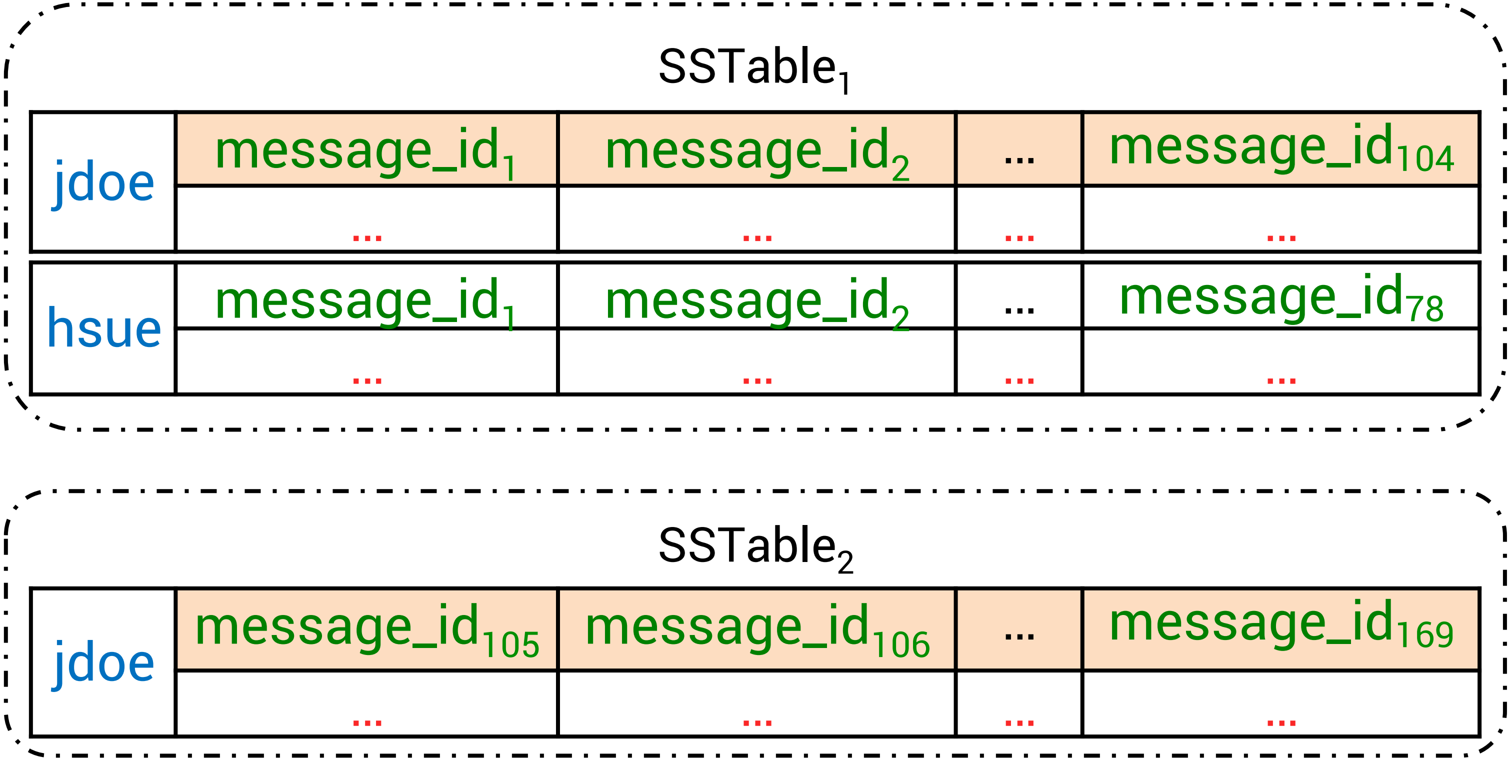
(#partition)

* login >= hsue login <= jdoe; ✗

(#partition)

* login %doe% ; ✗

On disk layout



SSTable$_1$

jdoe	message_id$_1$	message_id$_2$	...	message_id$_{104}$
	...	...	...	...
hsue	message_id$_1$	message_id$_2$	...	message_id$_{78}$
	...	...	...	...

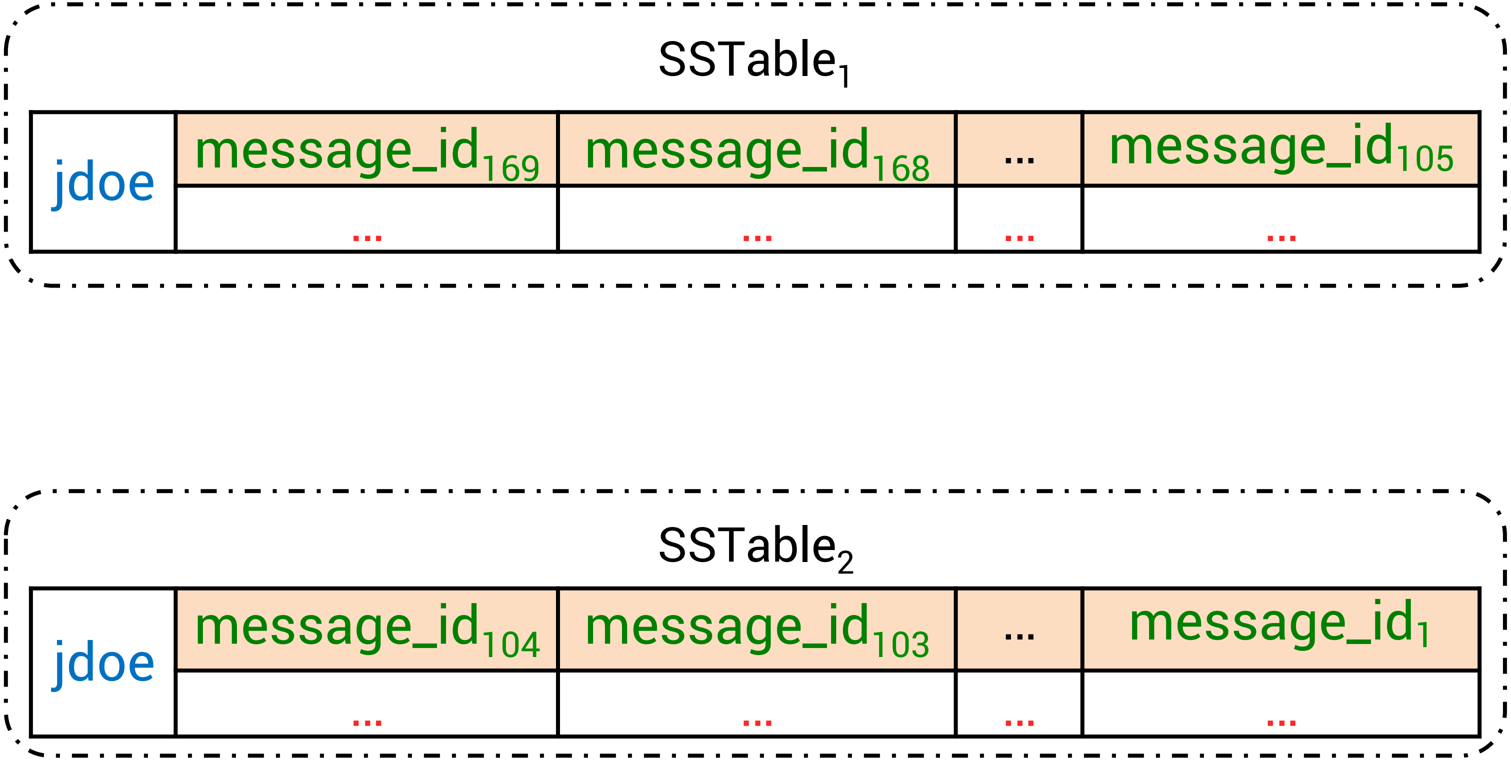
SSTable$_2$

jdoe	message_id$_{105}$	message_id$_{106}$	...	message_id$_{169}$
	...	...	...	...

Clustering order

```
                           (
    login      ,
    message_id              ,
                      ,
                    ,
                     ((login), message_id))
     CLUSTERING ORDER BY (message_id        ) ;
```

Reverse on disk layout



SSTable$_1$

jdoe	message_id$_{169}$	message_id$_{168}$	…	message_id$_{105}$
	…	…	…	…

SSTable$_2$

jdoe	message_id$_{104}$	message_id$_{103}$	…	message_id$_1$
	…	…	…	…

WHERE clause restrictions

(/ / /) #partition

exact match (=) #partition, (<, ≤, >, ≥)

- ☞ full cluster scan

clustering columns, (<, ≤, >, ≥) exact match

WHERE PRIMARY KEY

Dynamic search

?

- ,

Dynamic search

?

- ,

☞ Apache Solr ( ) (Datastax Enterprise)

```
*          solr_query = age:[33 TO *] AND gender:male ;

*          solr_query = lastname:*schwei?er ;
```

Collections & maps

```
                    (
    login      ,
                ,
          ,
            set<text>,
             list<text>,
                map<int, text>,
                      :                        (
    PRIMARY KEY(login));
```

(≈ 1000)

From SQL to CQL

: (

From SQL to CQL

: (

CQL not SQL

From SQL to CQL

: (

no join

(do you want to scale ?)

From SQL to CQL

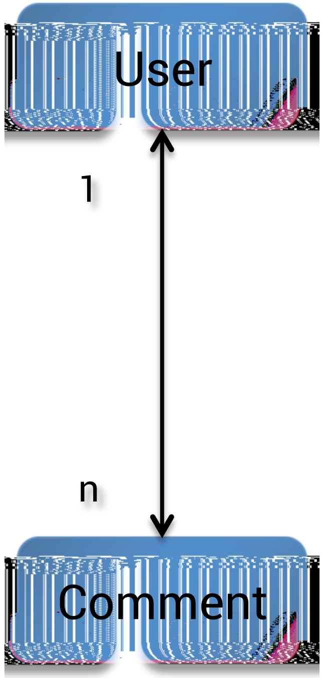
: (

no integrity constraint

(do you want to read-before-write ?)

From SQL to CQL

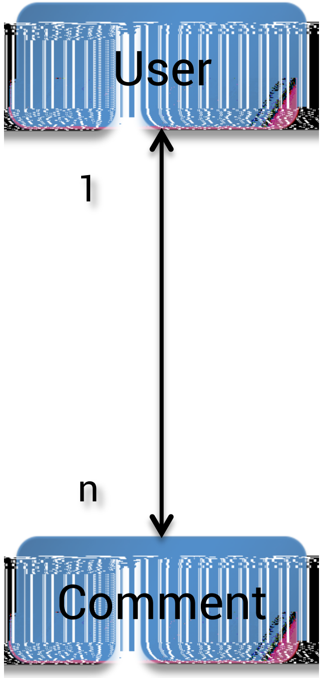
```
                              (
    article_id       ,
    comment_id              ,
    author_id text,  // typical join id
                  ,
                     ((article_id), comment_id));
```



From SQL to CQL

-

```
                                   (
    article_id         ,
    comment_id                ,
    author_json text,  // de-normalize
                    ,
                      ((article_id), comment_id));
```



Data modeling best practices

-
- 1 ≈ 1

Data modeling best practices

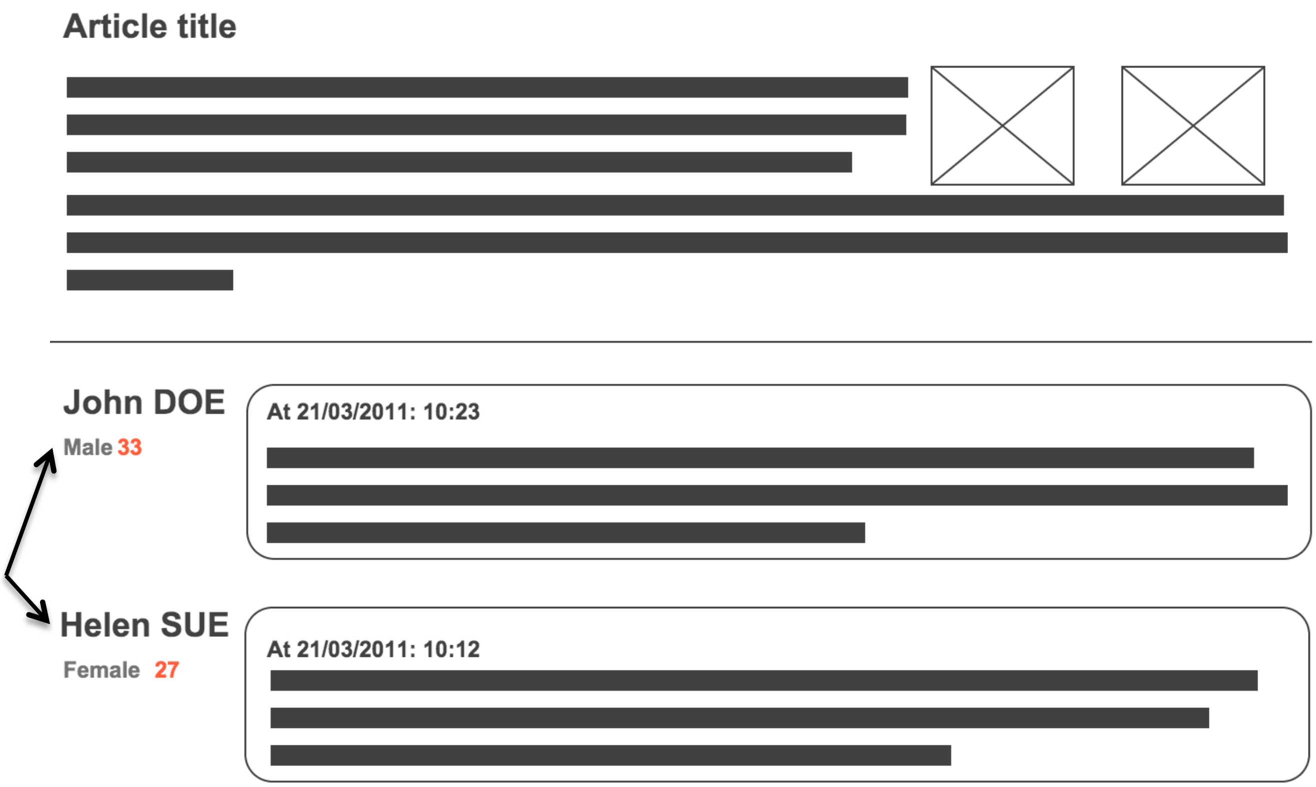
-
- 1 ≈ 1

- , necessary & immutable data
- / trade-off

Data modeling best practices

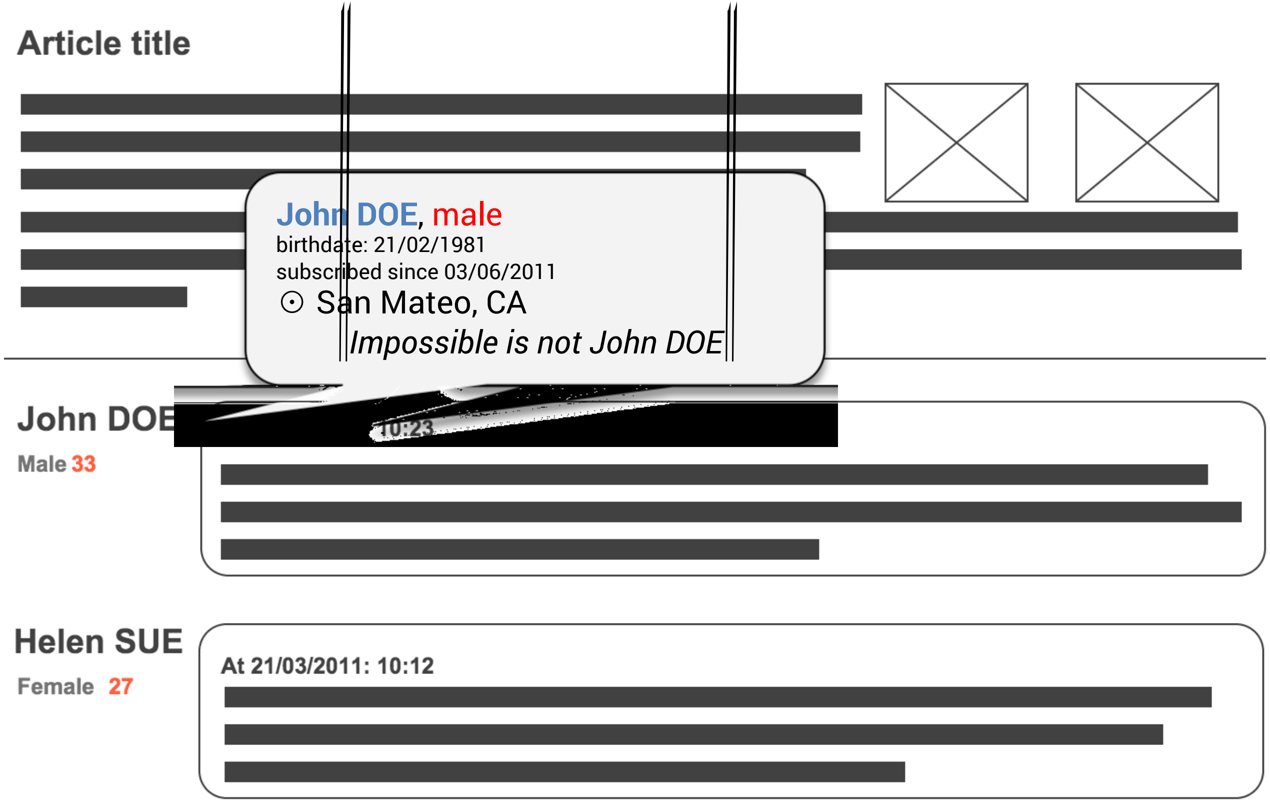
Person **JSON**

- firstname/lastname
- date of birth
- gender
- ~~mood~~
- ~~location~~



Data modeling best practices

Full detail read from
User table on click



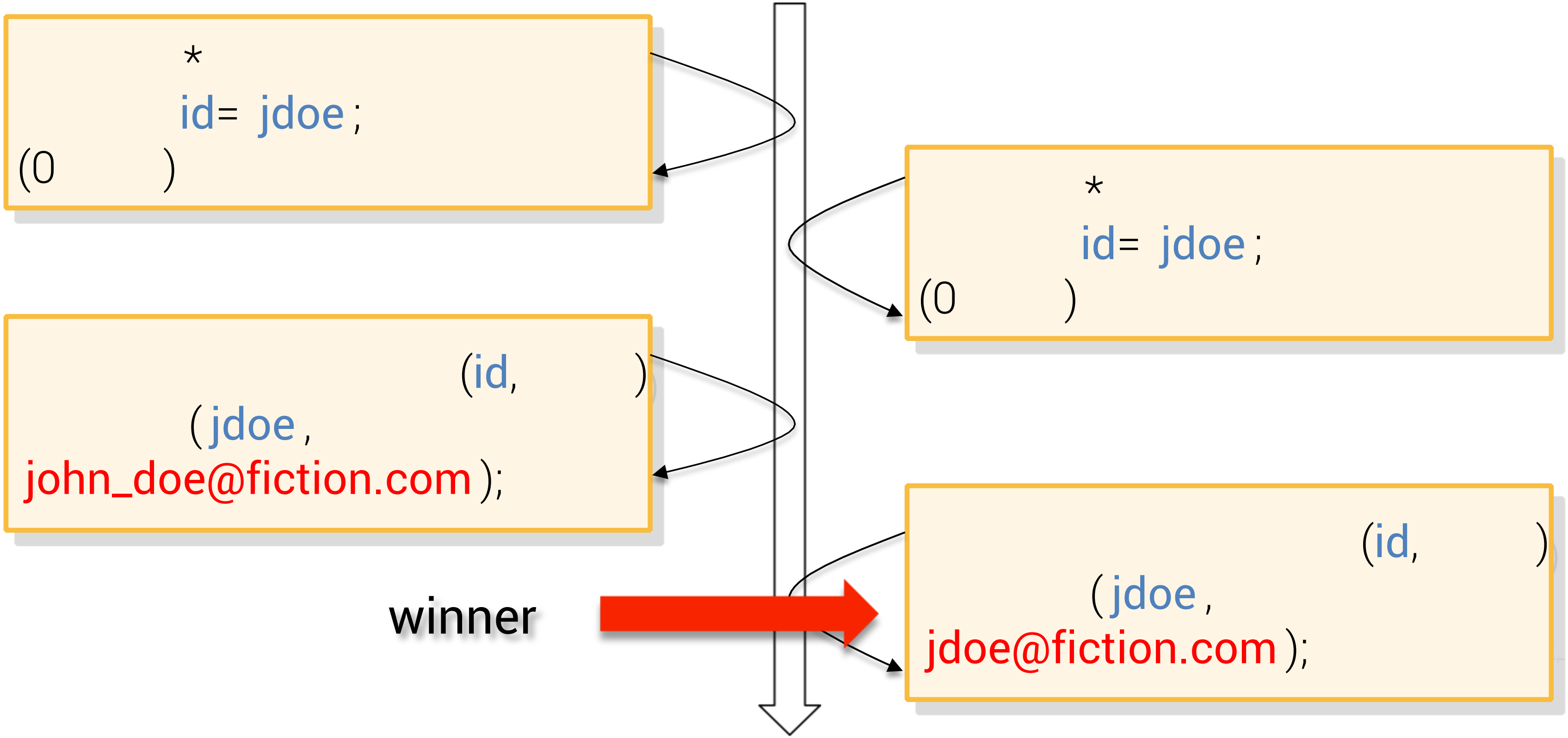
John DOE, male
birthdate: 21/02/1981
subscribed since 03/06/2011
☉ San Mateo, CA
Impossible is not John DOE

Lightweight Transaction (LWT)

? ☞ linearizable

? ☞

Lightweight Transaction (LWT)



*
id= jdoe ;
(0)

*
id= jdoe ;
(0)

(id,)
(jdoe ,
john_doe@fiction.com);

(id,)
(jdoe ,
jdoe@fiction.com);

winner

Lightweight Transaction (LWT)

? ☞ Paxos

?

```
( , ) ( , . )
IF NOT EXISTS;

= .
IF email = ‘john_doe@fiction.com’ = ;
```

Lightweight Transaction (LWT)

- ☞ must

IF NOT EXISTS : (

☞ IF EXISTS : (

Lightweight Transaction (LWT)

- ☞

must

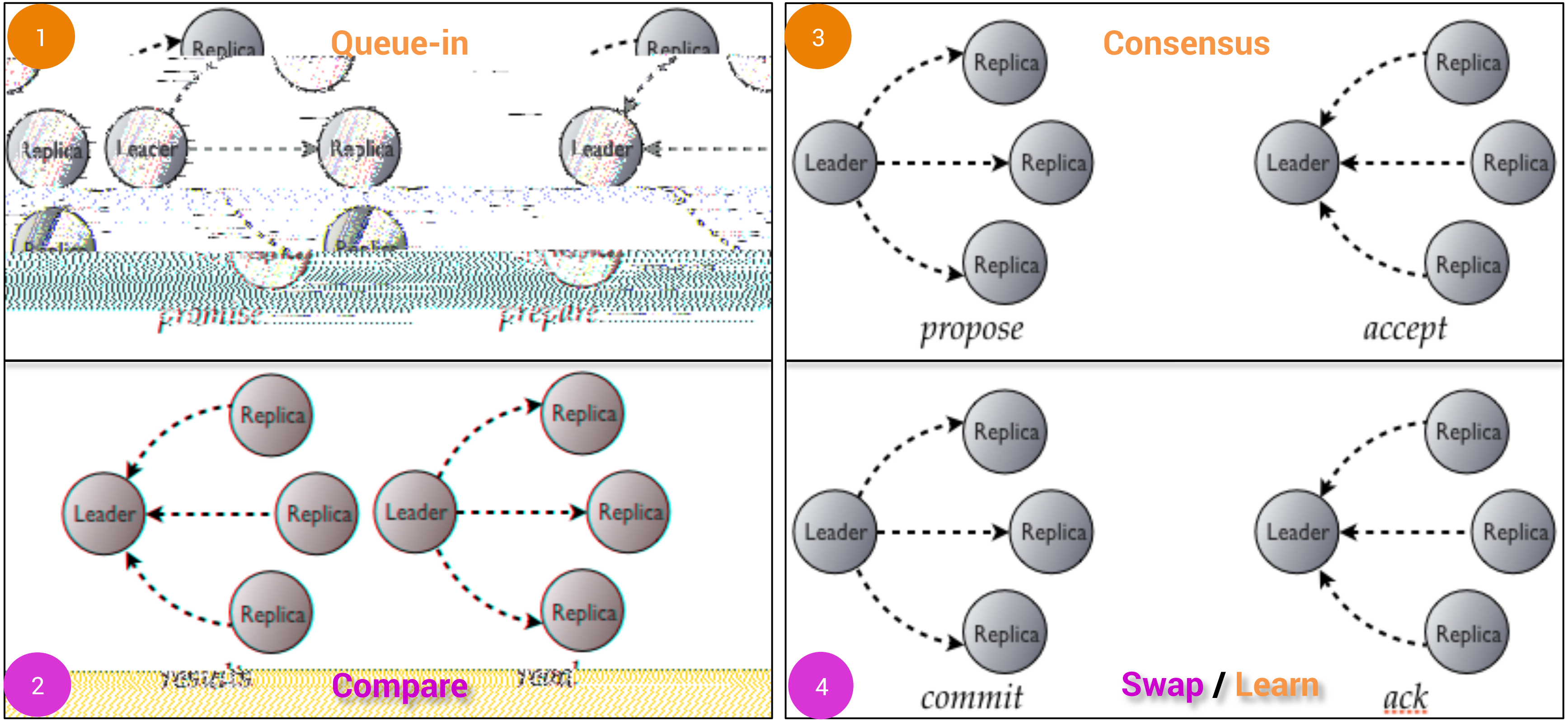
= IF condition_column = yyy

☞ condition_column = IF :

Lightweight Transaction (LWT)

- (4 -), do not abuse
- 1% – 5%

Lightweight Transaction (LWT)



1 Queue-in

2 Compare

3 Consensus

4 Swap / Learn

Q & R

Thank You

@doanduyhai

duy_hai.doan@datastax.com

https://academy.datastax.com/