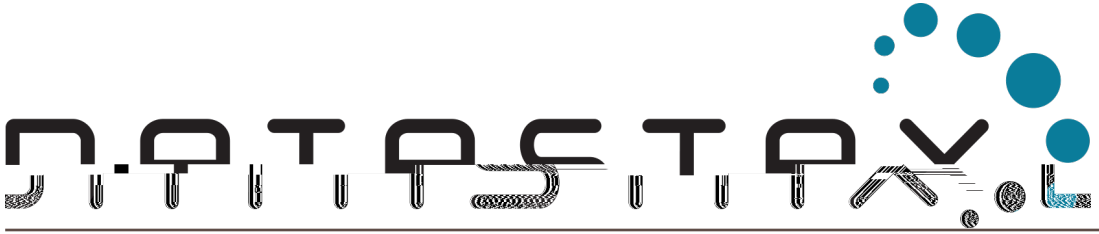


DuyHai DOAN, Technical Advocate

Agenda

DATASTAX

Architecture

-
-

Data model

- ( ),
- (, , ,)
-

Who Am I ?

DATASTAX

Duy Hai DOAN

- , ,
- - (Achilles,)
-
  - ☞ **duy_hai.doan@datastax.com**
  - ☞ **@doanduyhai**

Datastax

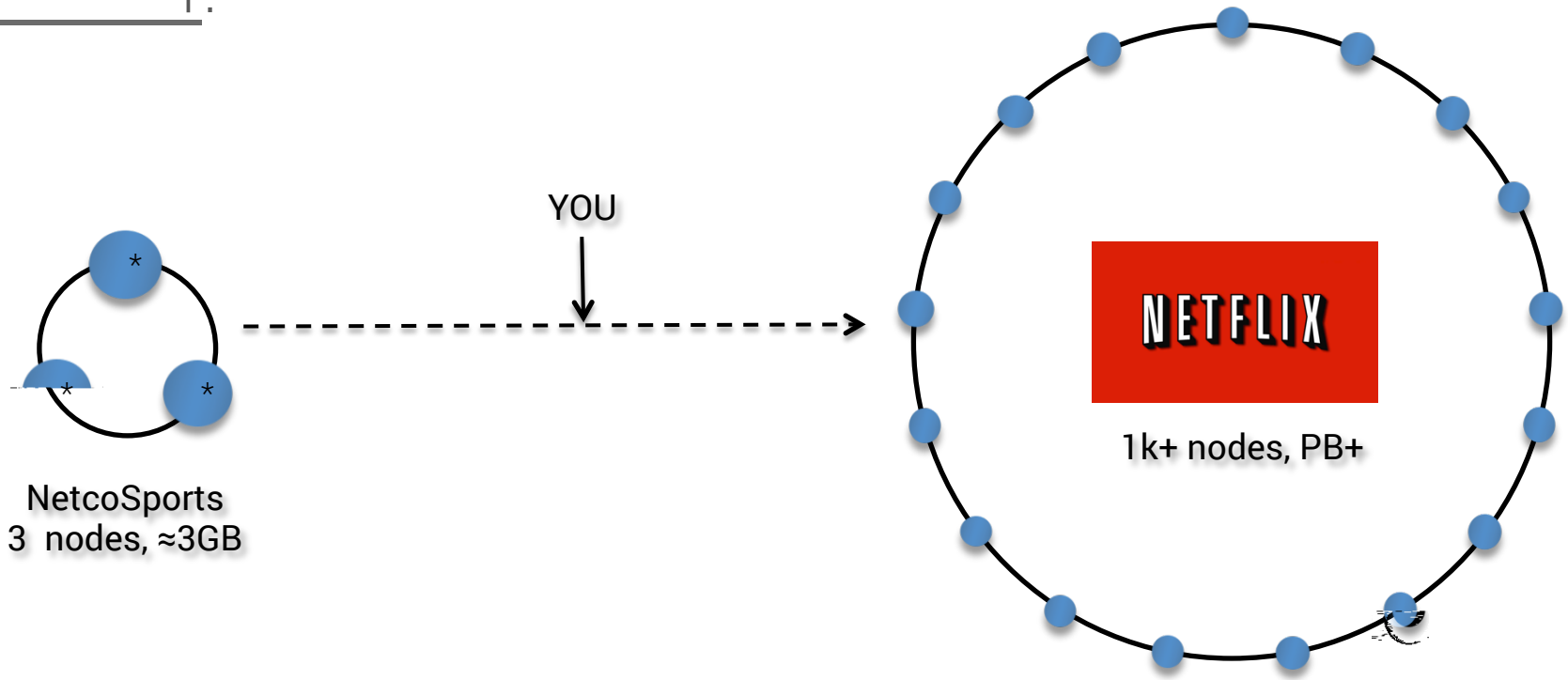
DATASTAX

- April 2010
- a lot
- 400+ (25 100), 200+
-
- London, France Germany
- Datastax Enterprise = + extra features

Cassandra 5 key facts

DATASTAX

1:



Cassandra 5 key facts

DATASTAX

2: (≈100% -)

- ( )

Cassandra 5 key facts

DATASTAX

3: -

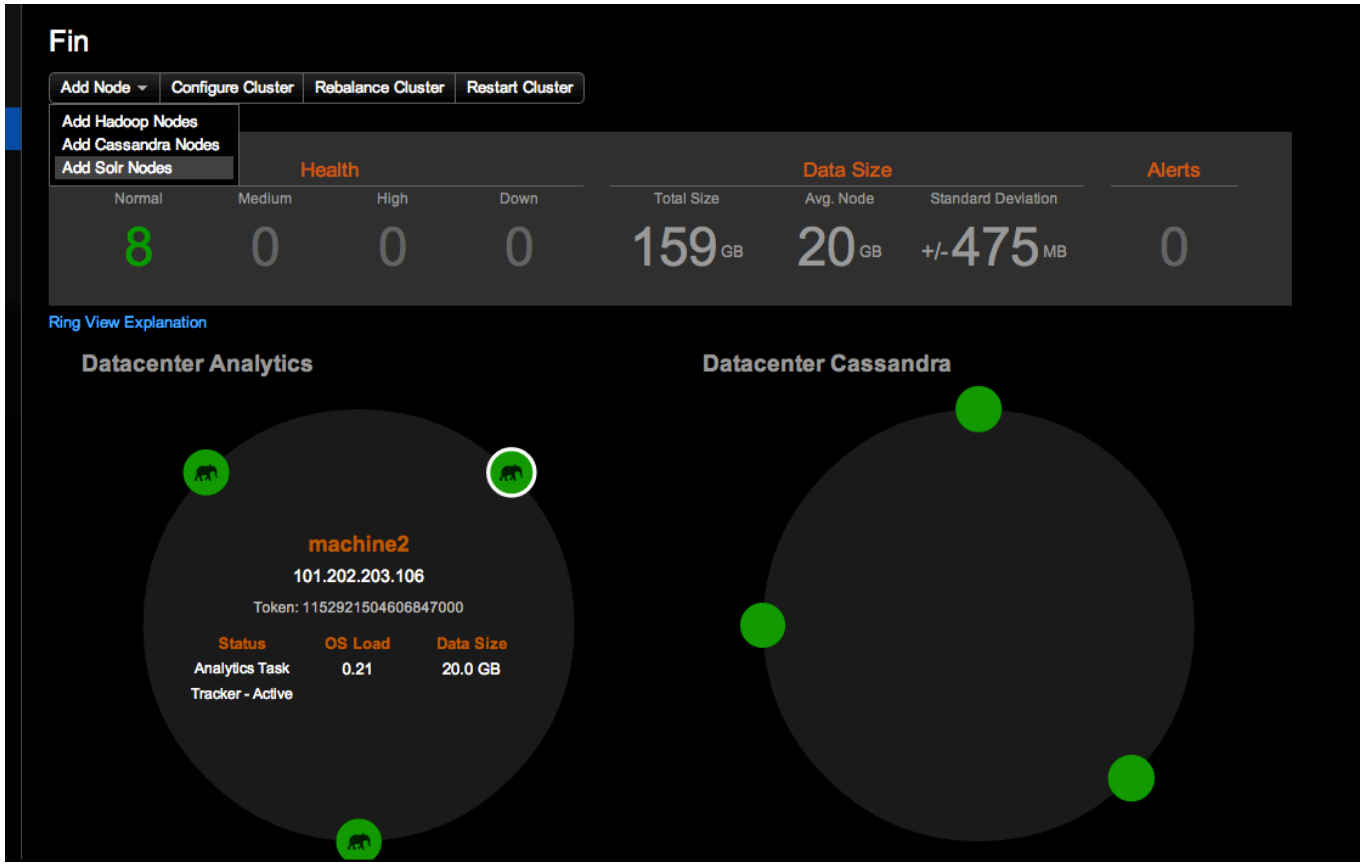
- - - - ( )
- -
- /
-

Cassandra 5 key facts

4:

- 1 = 1 + 2 (+)
-
-

Cassandra 5 key facts



Cassandra 5 key facts

DATASTAX

5:

- Cassandra + Spark = awesome !
- realtime streaming/

DATASTAX

Cassandra architecture

Cluster

Replication

Cassandra architecture

DATASTAX

- DynamoDB
- masterless

-

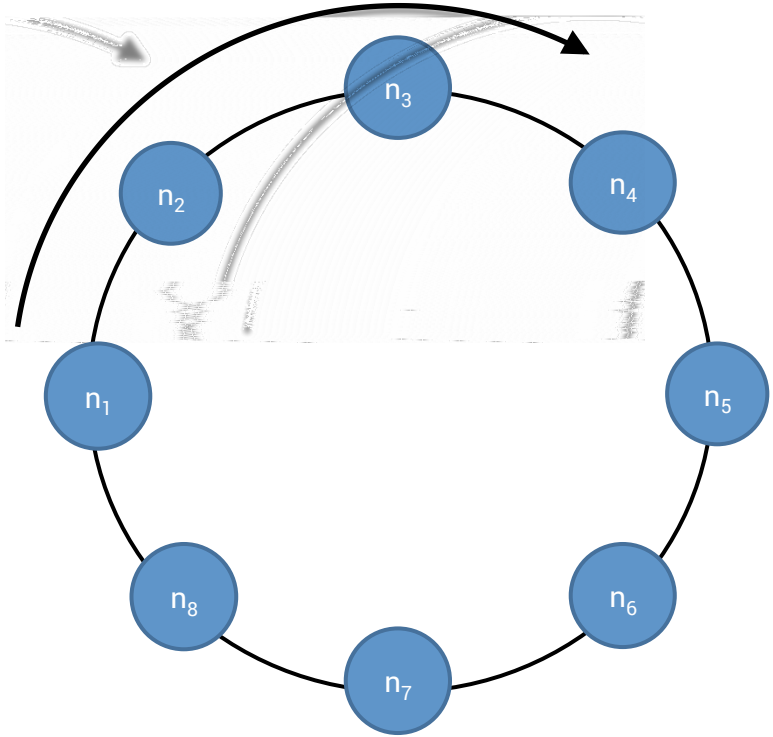
- Big Table
- /

Data distribution

: #partition → token = (#p)

: - ,

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



Token Ranges

A: 0, /8

B: /8, 2 /8

C: 2 /8, 3 /8

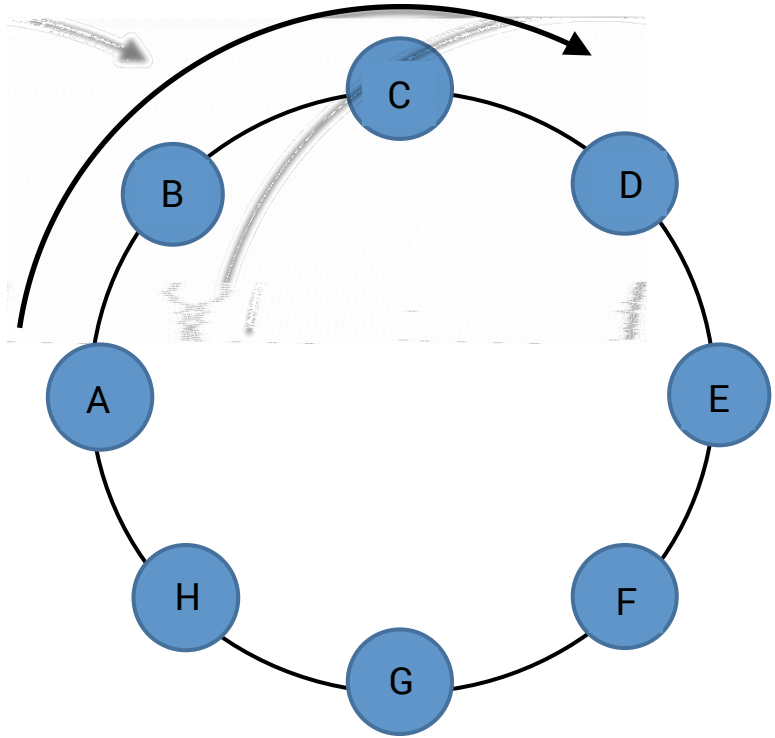
D: 3 /8, 4 /8

E: 4 /8, 5 /8

F: 5 /8, 6 /8

G: 6 /8, 7 /8

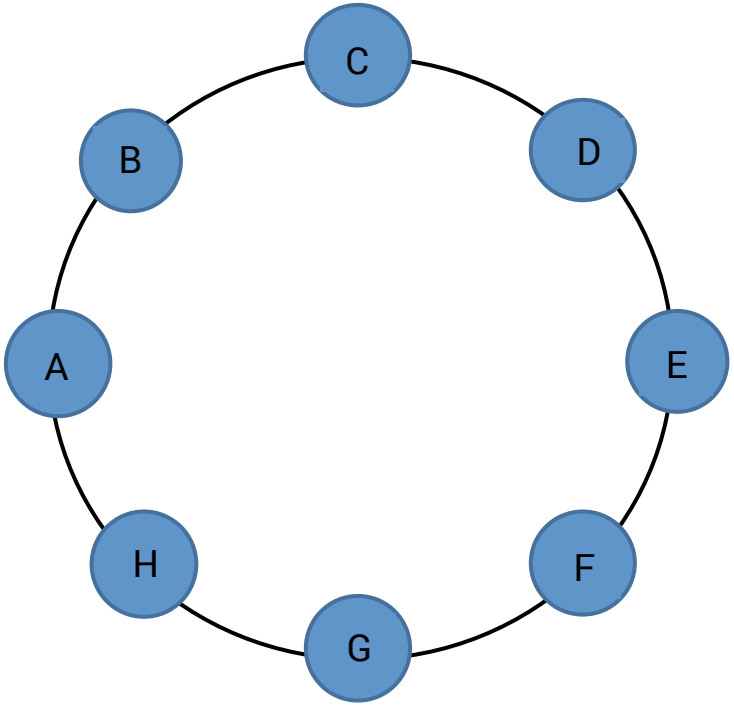
H: 7 /8,



Distributed Table

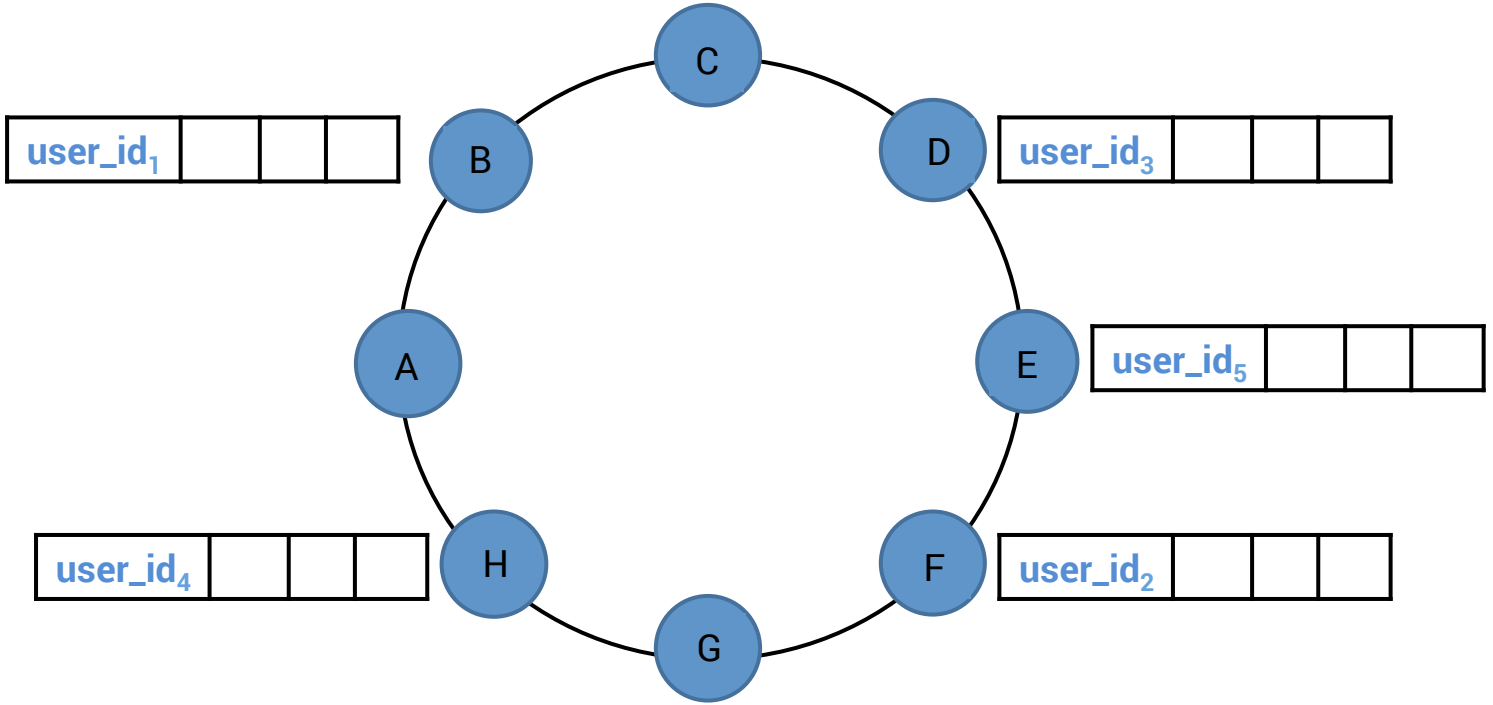
DATASTAX

$user\_id_1$			
$user\_id_2$			
$user\_id_3$			
$user\_id_4$			
$user\_id_5$			



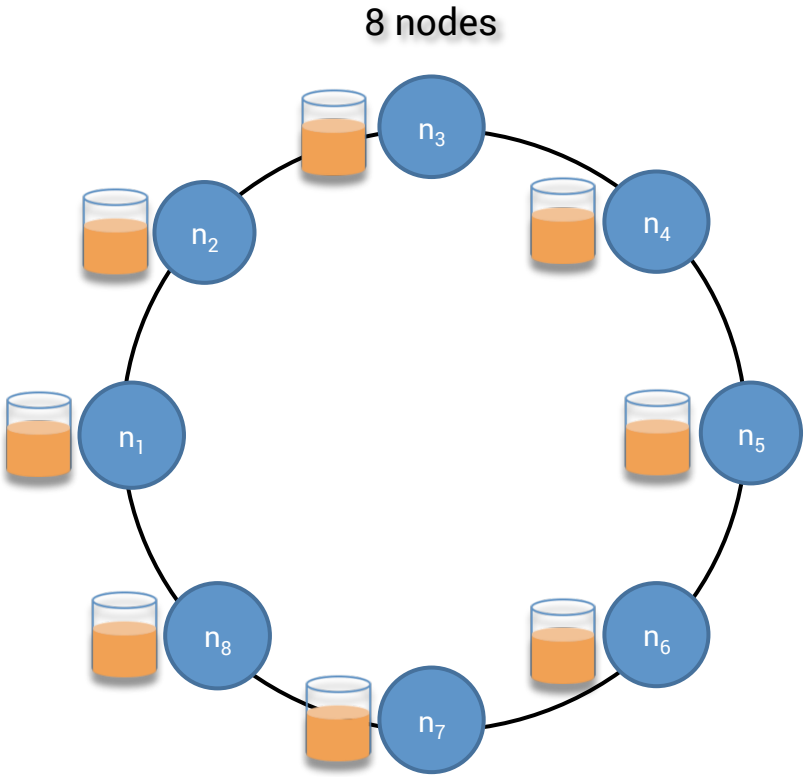
Distributed Table

DATASTAX



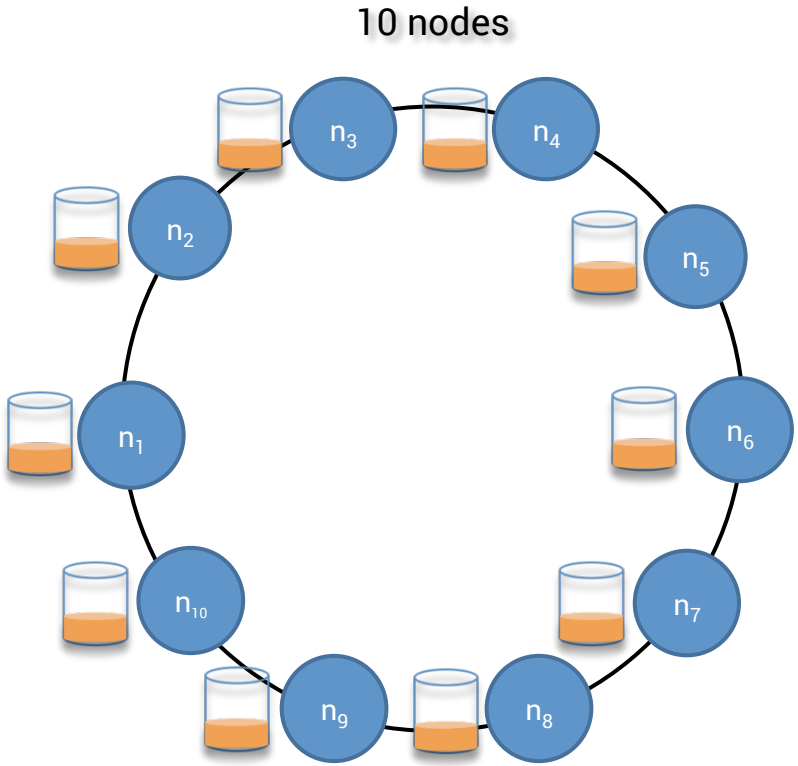
Linear scalability

DATASTAX



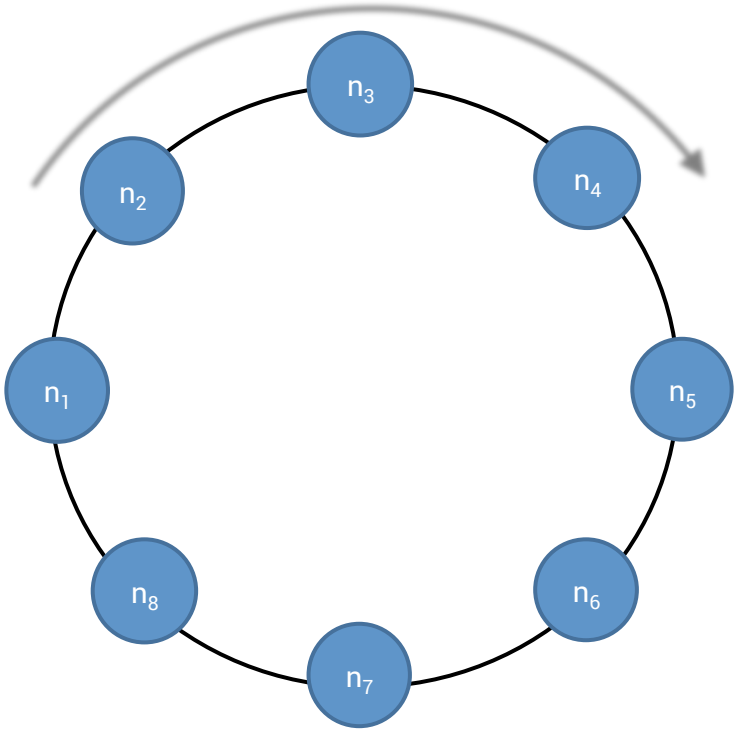
Linear scalability

DATASTAX



Failure tolerance

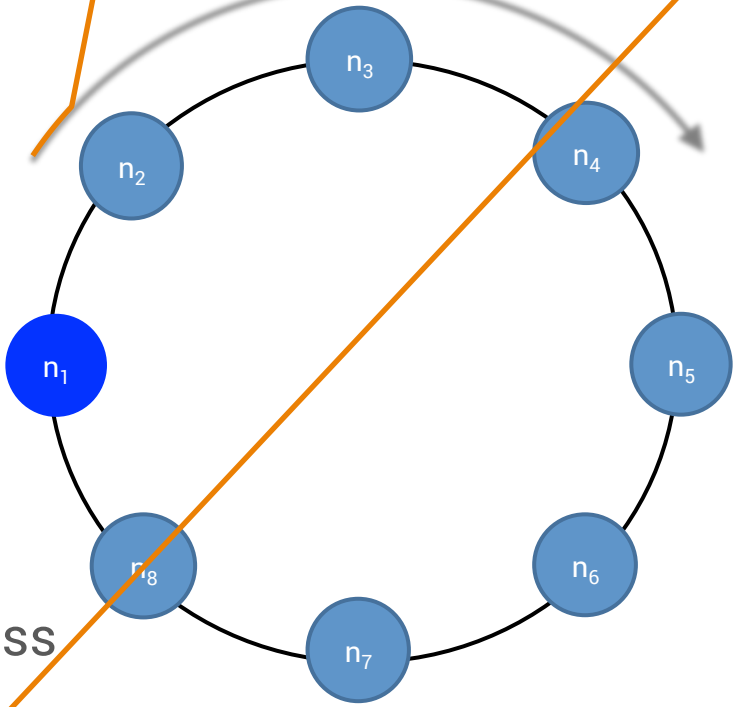
(RF) = 3



Coordinator node

(/)

Coordinator



coordinator →masterless

Consistency

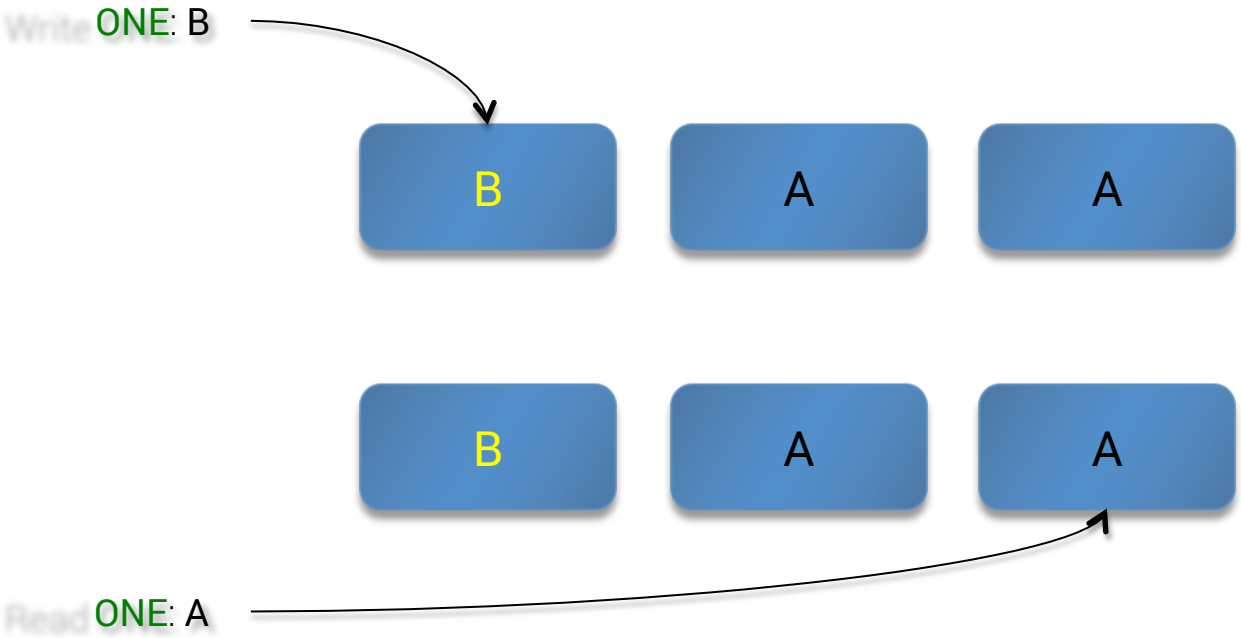
DATASTAX

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

read & write

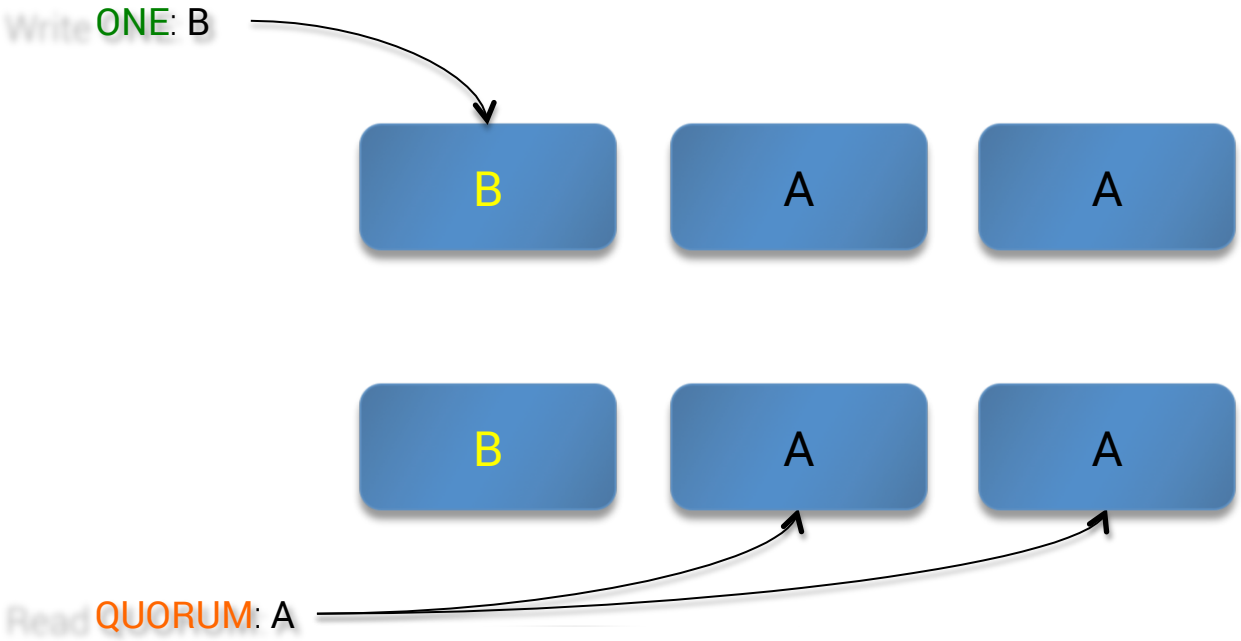
Consistency in action

= 3, ONE, ONE



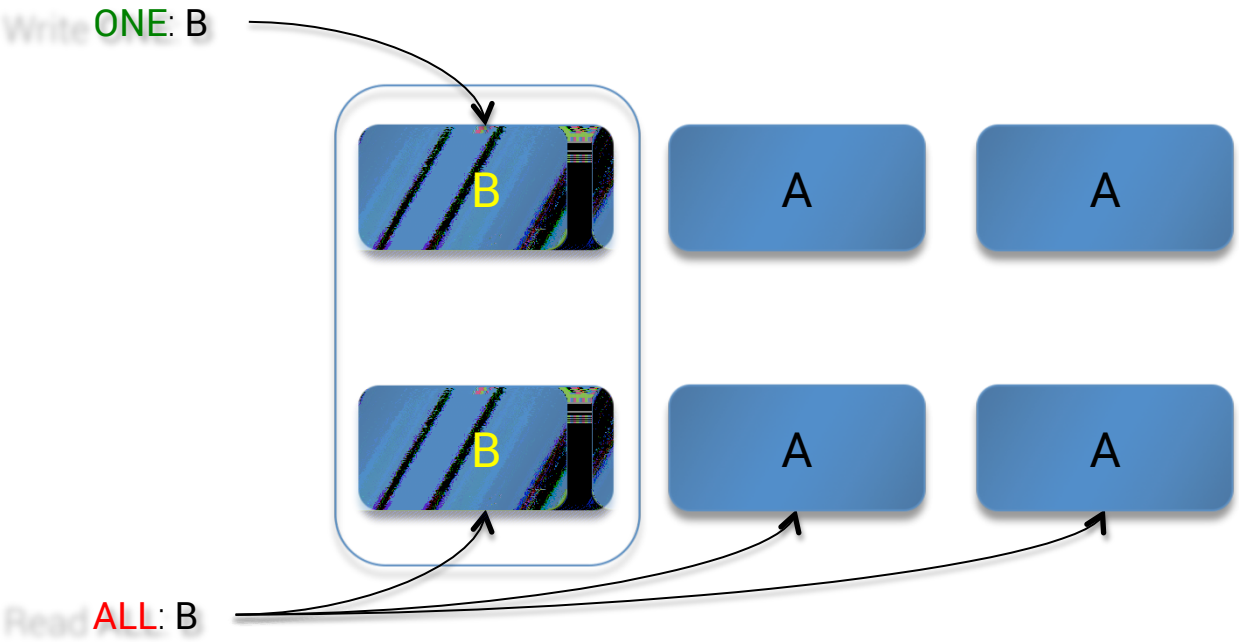
Consistency in action

= 3, ONE, QUORUM



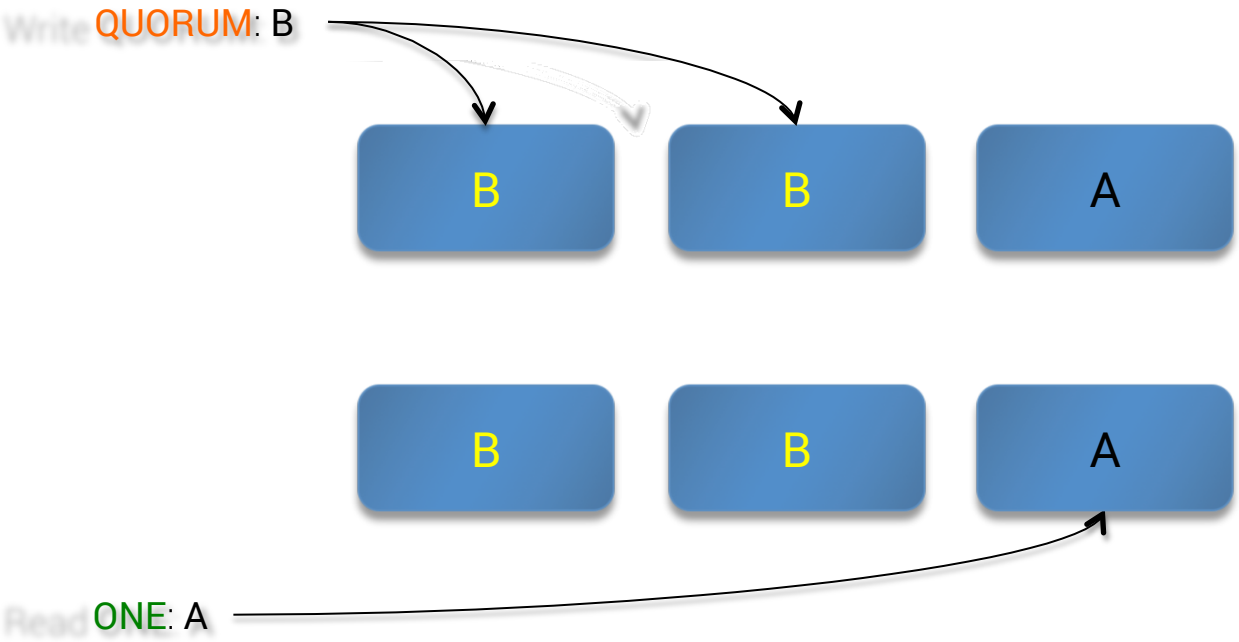
Consistency in action

= 3, ONE, ALL



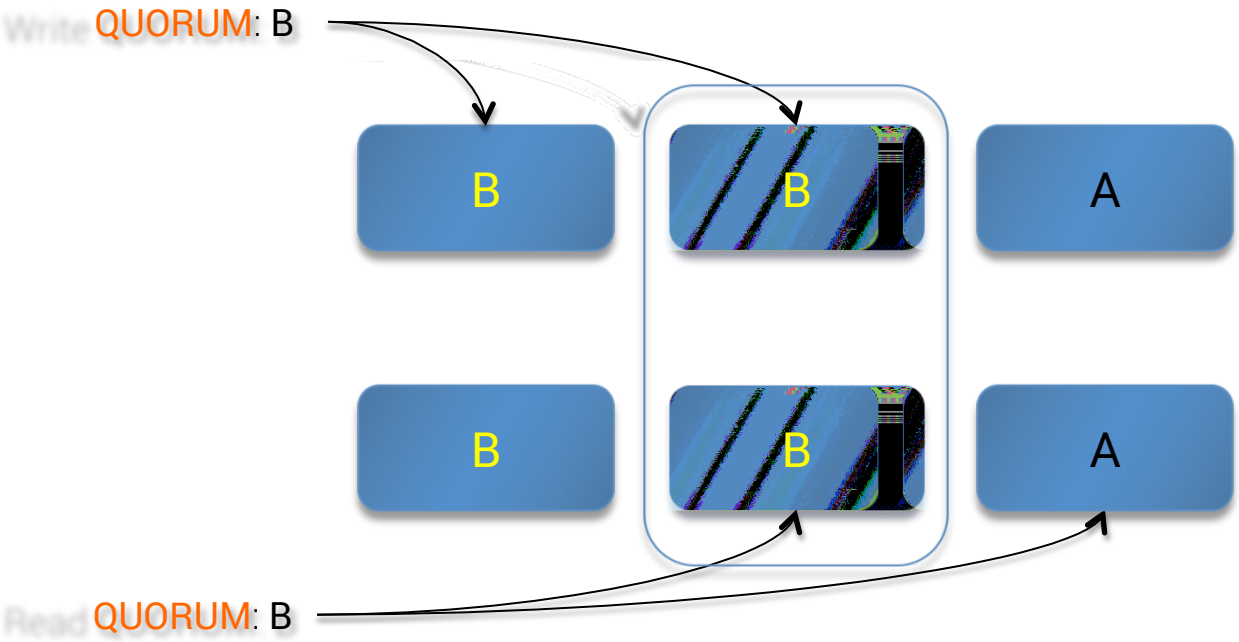
Consistency in action

= 3, QUORUM, ONE



Consistency in action

= 3, QUORUM, QUORUM



Consistency trade-off

DATASTAX

Consistency level

DATASTAX

ONE

Fast, may not read latest written value

Consistency level

DATASTAX

QUORUM

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

Good balance

Consistency level

DATASTAX

ALL

Paranoid

Slow, no high availability

Consistency summary

DATASTAX

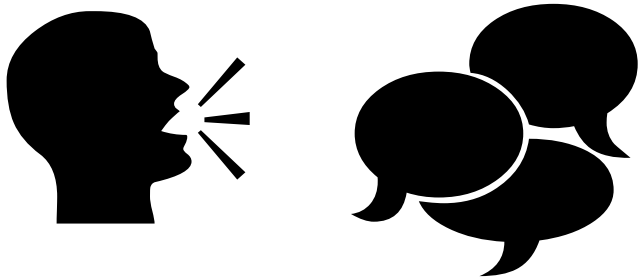
$ONE_{Read} + ONE_{Write}$

☞ available / (N-1)

$QUORUM_{Read} + QUORUM_{Write}$

☞ available / 1+

DATASTAX



Q & R

DATASTAX

Data model

Last Write Win

CQL basics

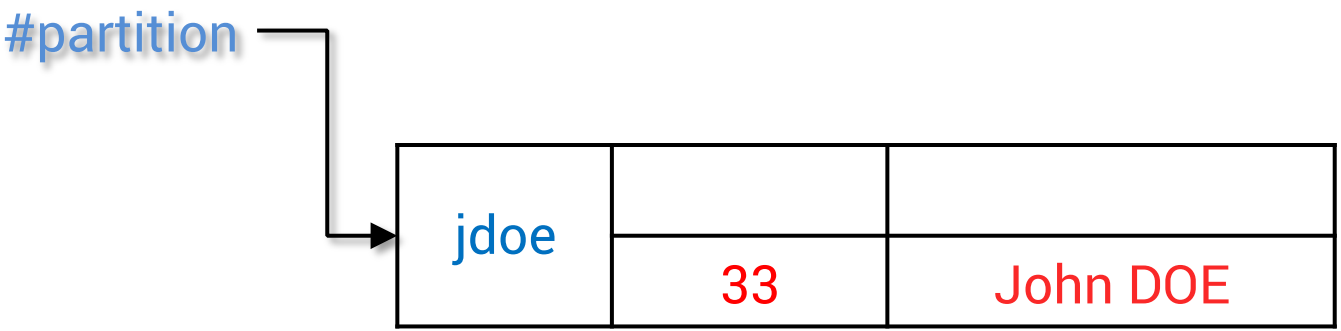
Clustered tables

Lightweight transactions

Last Write Win (LWW)

DATASTAX

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



Last Write Win (LWW)

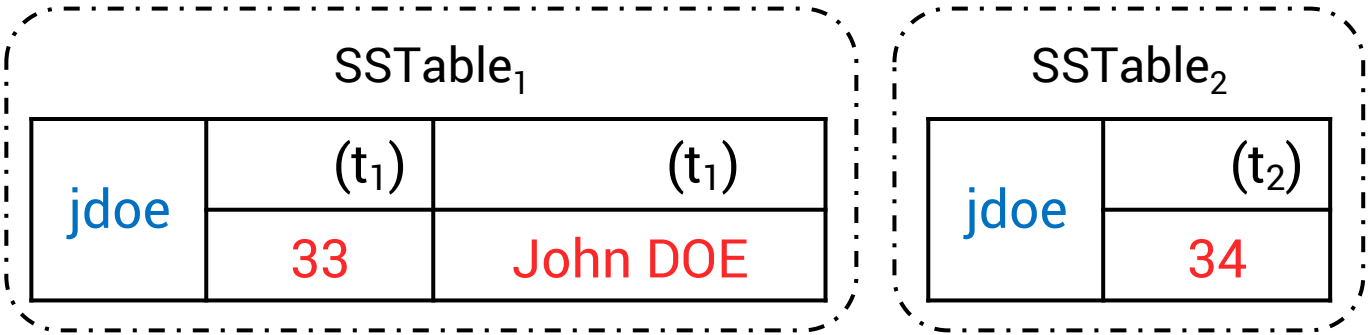
DATASTAX

(login, ,) (jdoe, +

jdoe	(t_1)	(t_1)
	33	John DOE

Last Write Win (LWW)

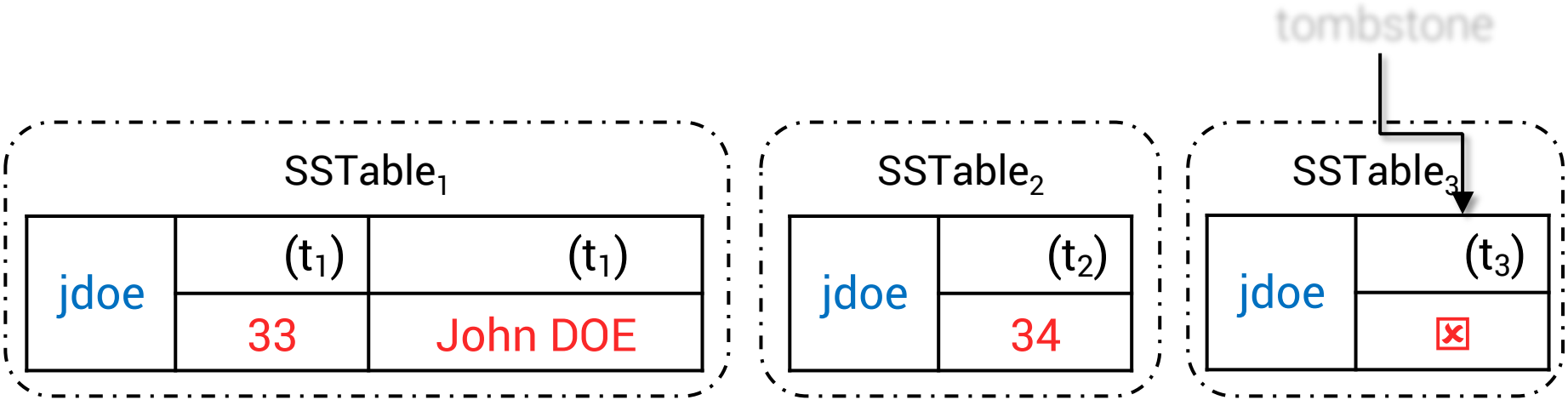
= 34 login = jdoe;



Last Write Win (LWW)

DATASTAX

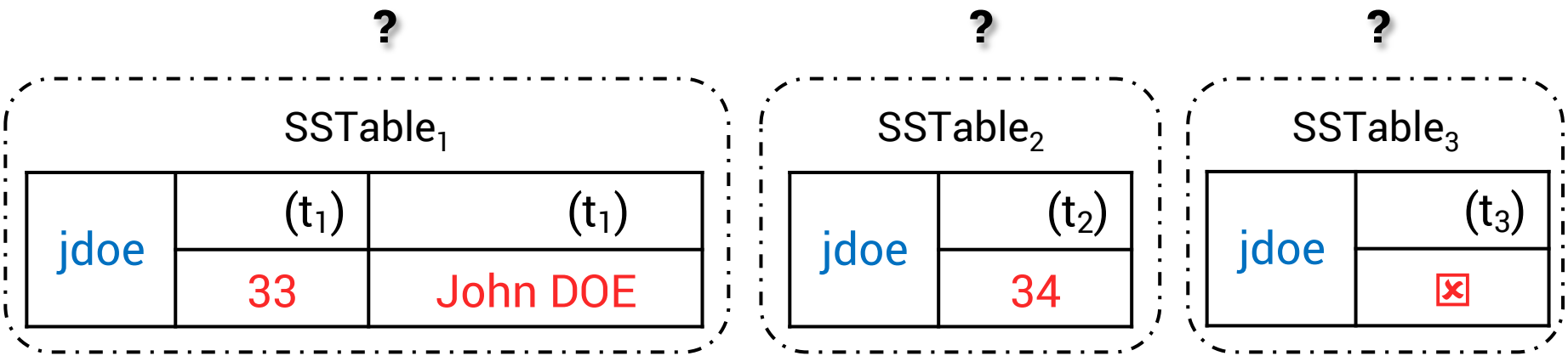
login = jdoe;



Last Write Win (LWW)

DATASTAX

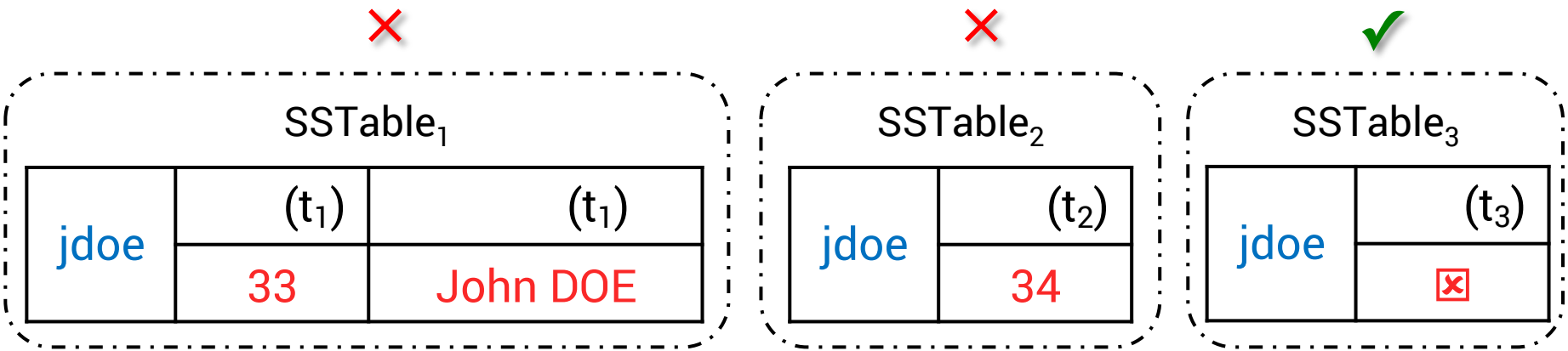
login = jdoe;



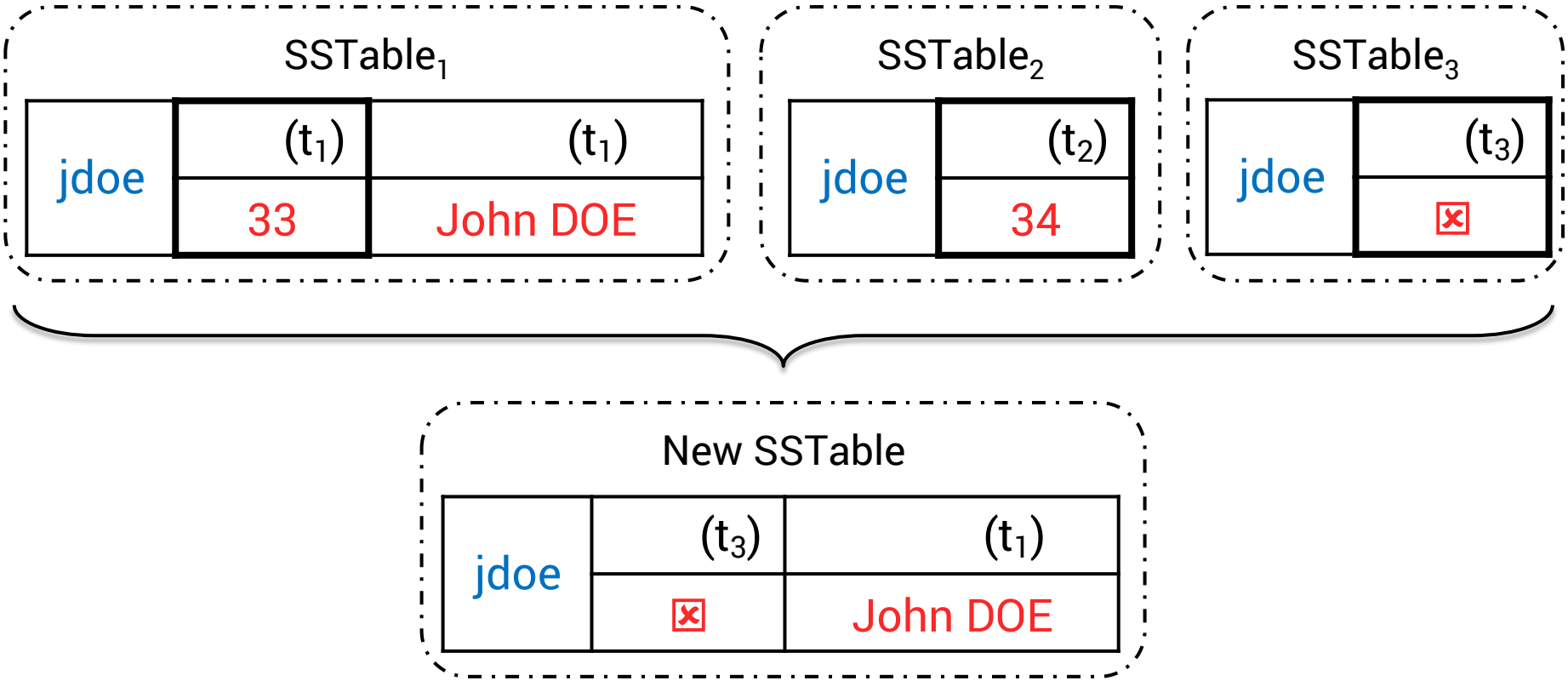
Last Write Win (LWW)

DATASTAX

login = jdoe;



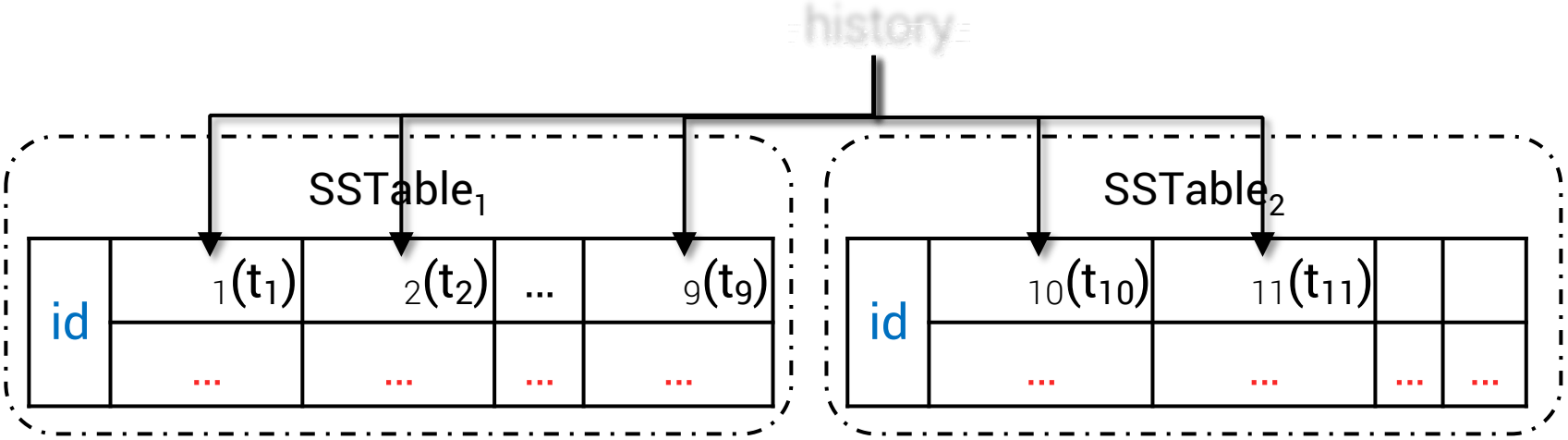
Compaction


SSTable1
jdoe
(t1)
33
(t1)
John DOE
SSTable2
jdoe
(t2)
34
SSTable3
jdoe
(t3)
New SSTable
jdoe
(t3)
(t1)
John DOE

Historical data

?

- do not
- ☞ time-series



CRUD operations

DATASTAX

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

= 34 login = jdoe;

login = jdoe;

login = jdoe;

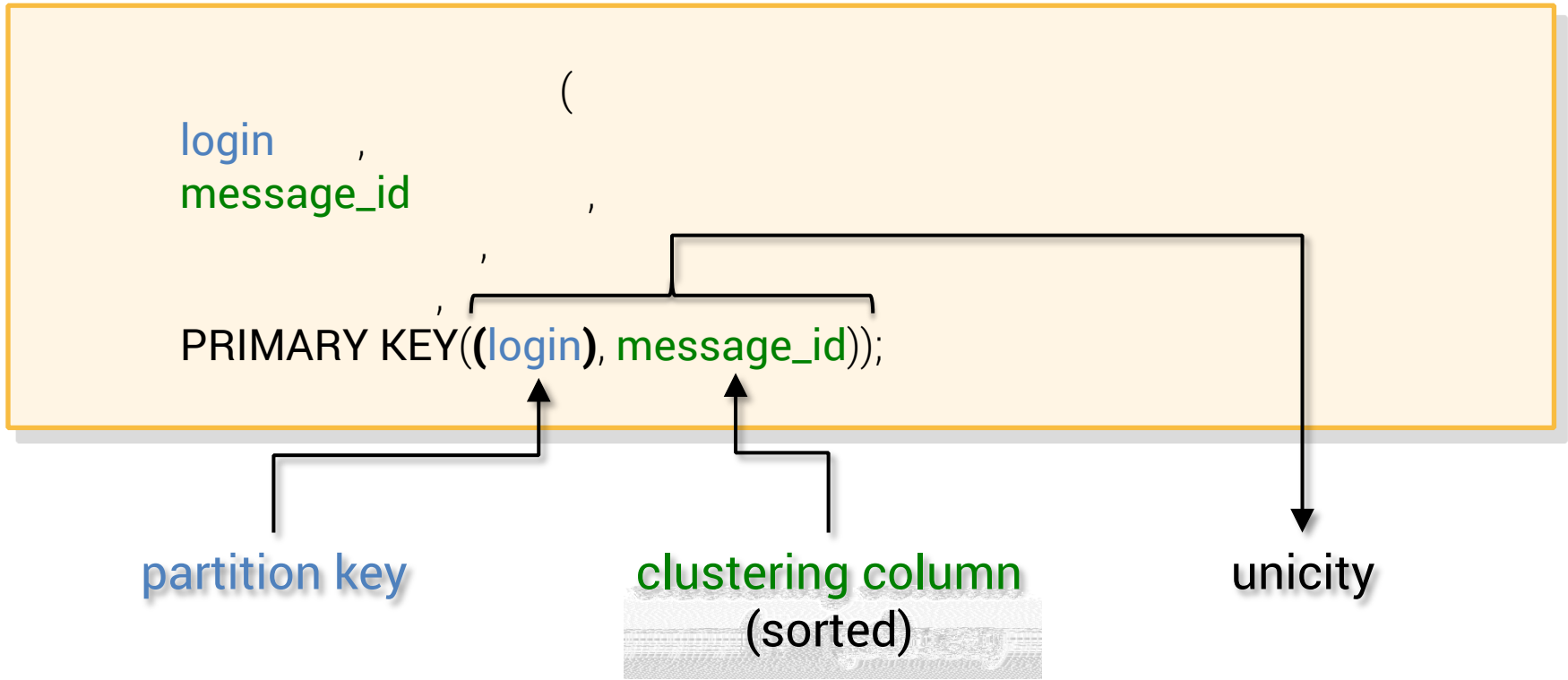
Simple Table

```
                    (
login       ,
             ,
          ,

PRIMARY KEY(login));
```

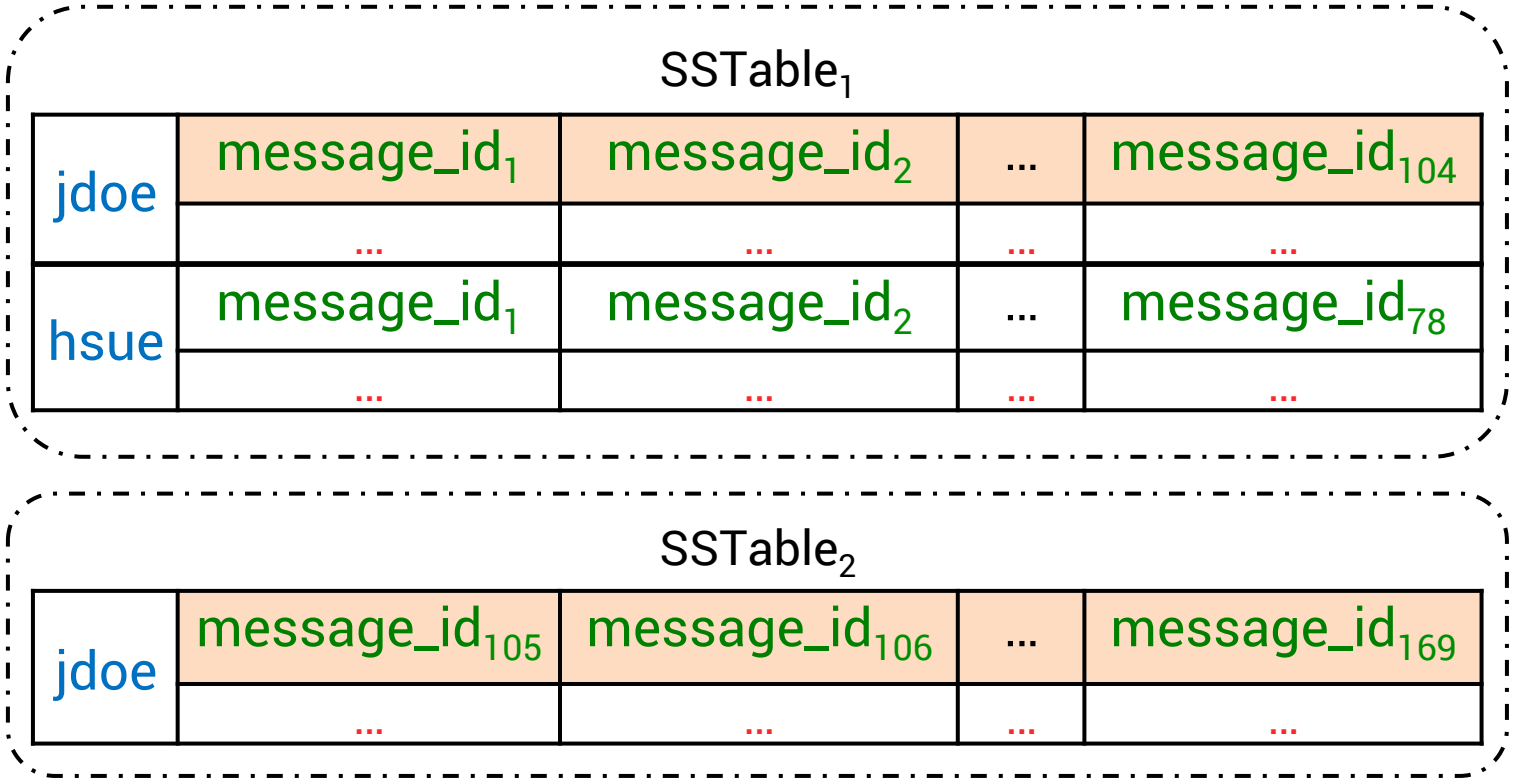
partition key (#partition)

Clustered table (1 – N)



On disk layout

DATASTAX



Queries

( )

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 ; +
```

Without #partition

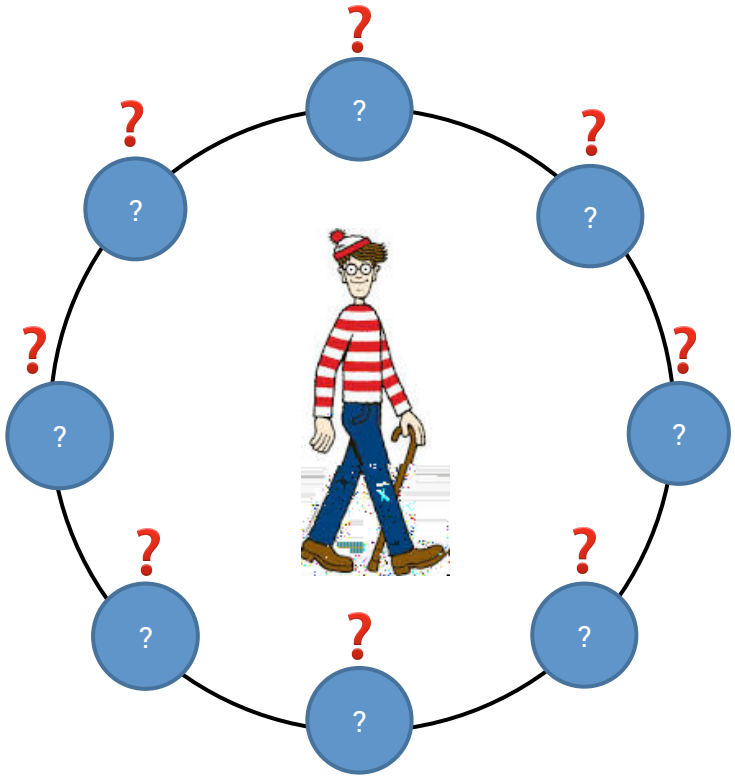
DATASTAX

#

☞

☞

?



The importance of #partition

DATASTAX

In RDBMS, no primary key

☞ full table scan

The importance of #partition

DATASTAX

With Cassandra, no partition key

☞ full CLUSTER scan

Queries

(#partition)

```
*                    login >= hsue      login <= jdoe;
```

(#partition)

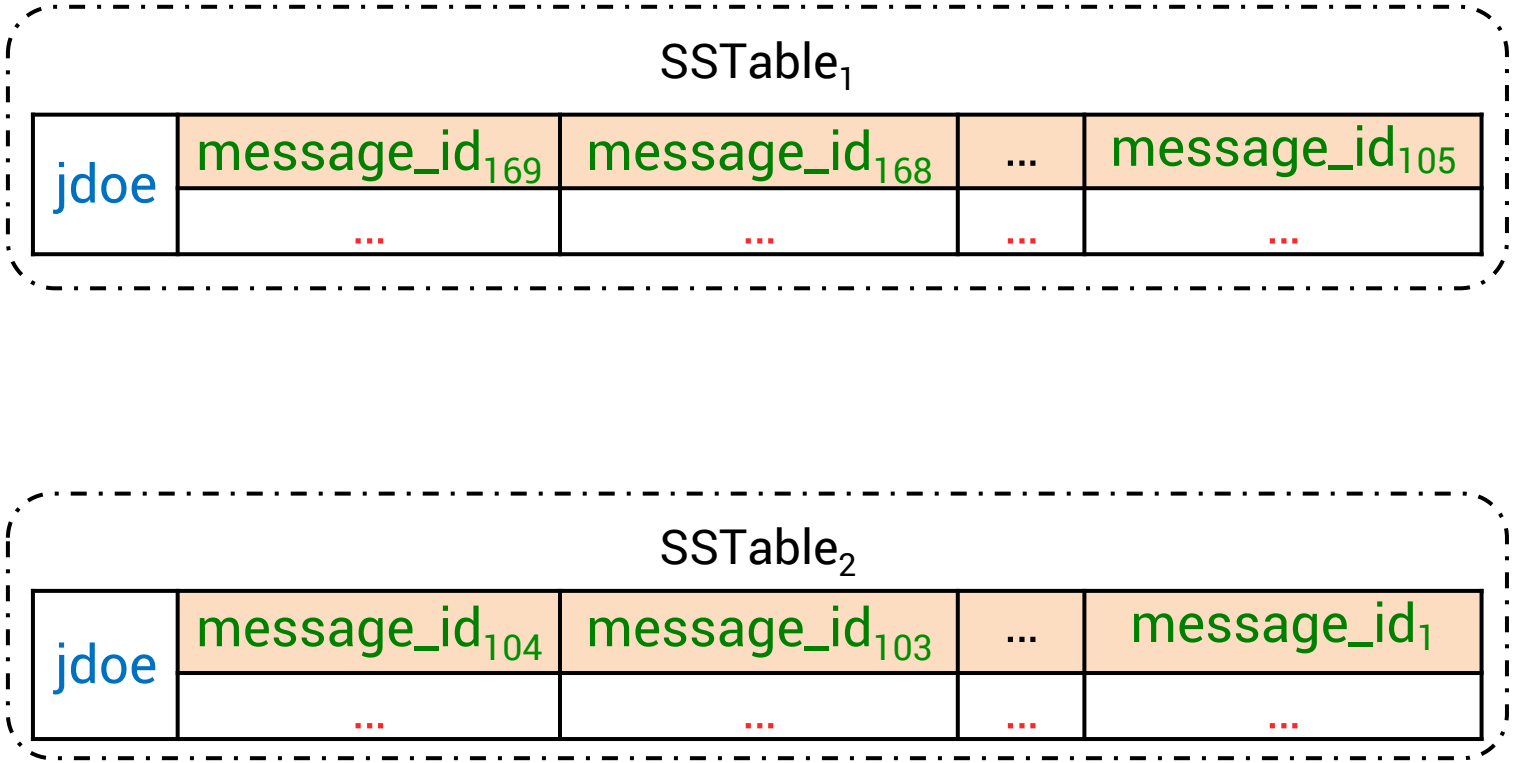
```
*                    login      %doe% ;
```

Clustering order

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

Reverse on disk layout

DATASTAX



WHERE clause restrictions

DATASTAX

(/ / /) #partition

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

- ☞ full cluster scan

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

WHERE

-

WHERE clause restrictions

DATASTAX

?

- ,

WHERE clause restrictions

?

- ,

-

☞ Apache Solr ( ) (Datastax Enterprise)

☞ , 1- -2- (&)

WHERE clause restrictions

DATASTAX

?

- ,

-

☞ Apache Solr ( ) (Datastax Enterprise)

☞ , 1- -2- (&)

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

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

Collections & maps

DATASTAX

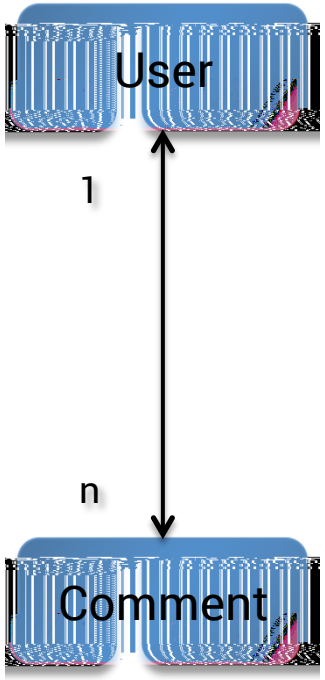
```
                    (
login       ,
              ,
          ,
           set<text>,
            list<text>,
               map<int, text>,

PRIMARY KEY(login));
```

(≈ 1000)

From SQL to CQL

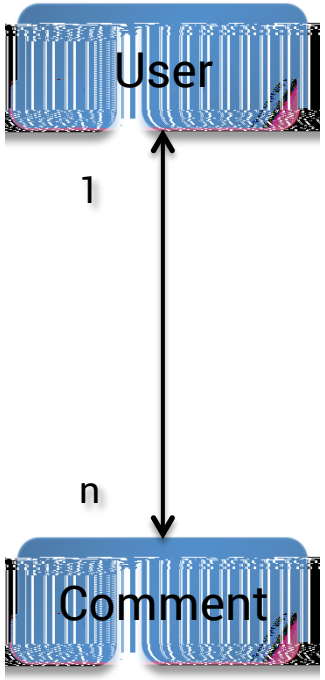
```
                                    (
article_id          ,
comment_id                  ,
author_id text,  // typical join id
                    ,
                        ((article_id), comment_id));
```



From SQL to CQL

-

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



Data modeling best practices

DATASTAX

-
- 1 ≈ 1

Data modeling best practices

DATASTAX

-
- 1 ≈ 1

- , necessary & immutable data
- / trade-off

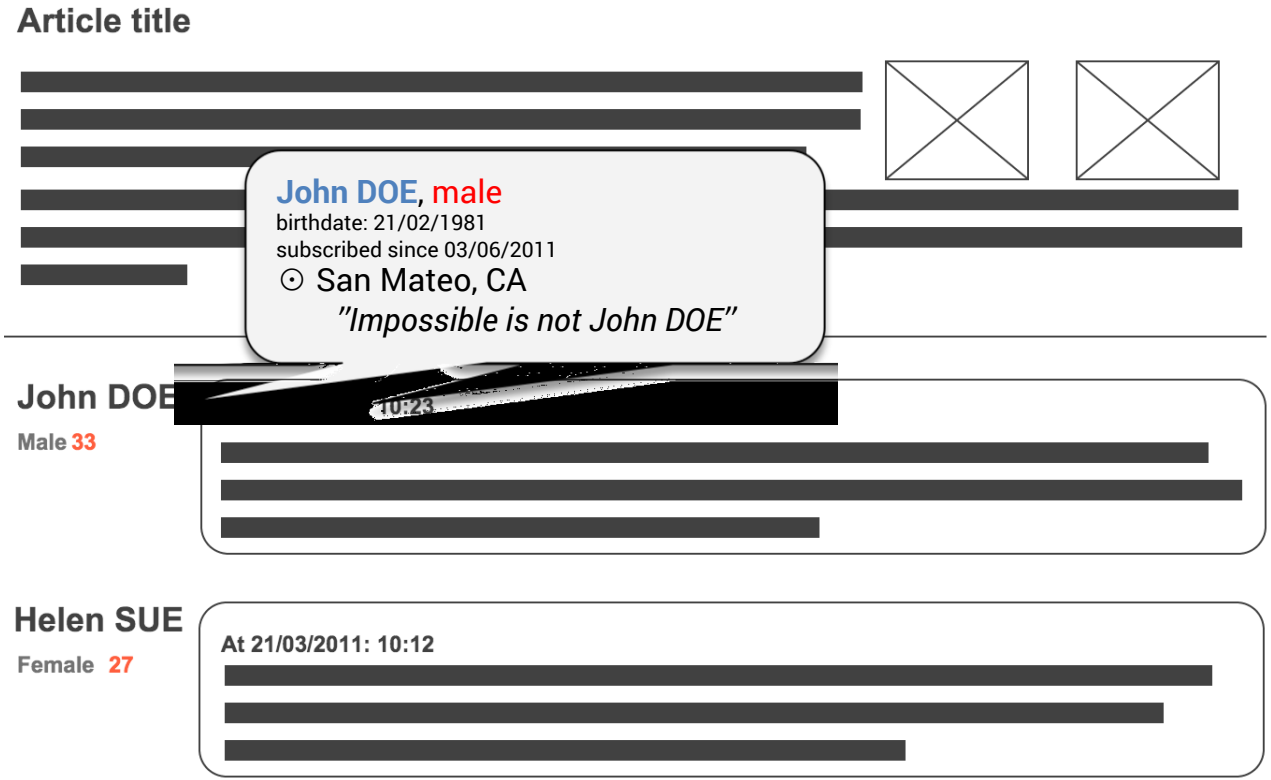
Data modeling best practices

DATASTAX

Person **JSON**
- firstname/lastname
- date of birth
- gender
- ~~mood~~
- ~~location~~

Data modeling best practices

Full detail read from **User** table on click



Lightweight Transaction (LWT)

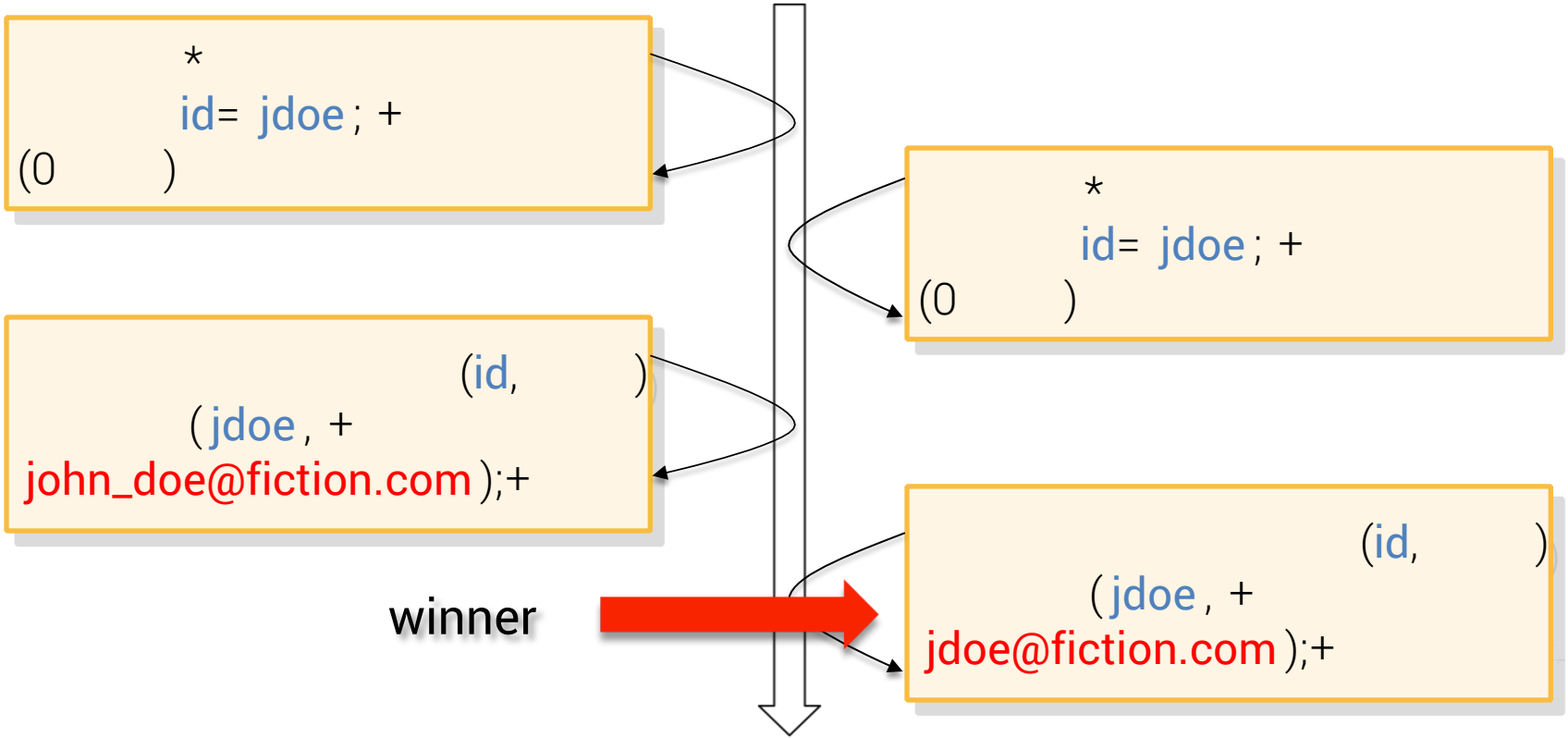
DATASTAX

? ☞ linearizable

? ☞

Lightweight Transaction (LWT)

DATASTAX



Lightweight Transaction (LWT)

DATASTAX

? ☞ Paxos

?

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

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

Lightweight Transaction (LWT)

DATASTAX

- ☞ must

IF NOT EXISTS

☞ IF EXISTS

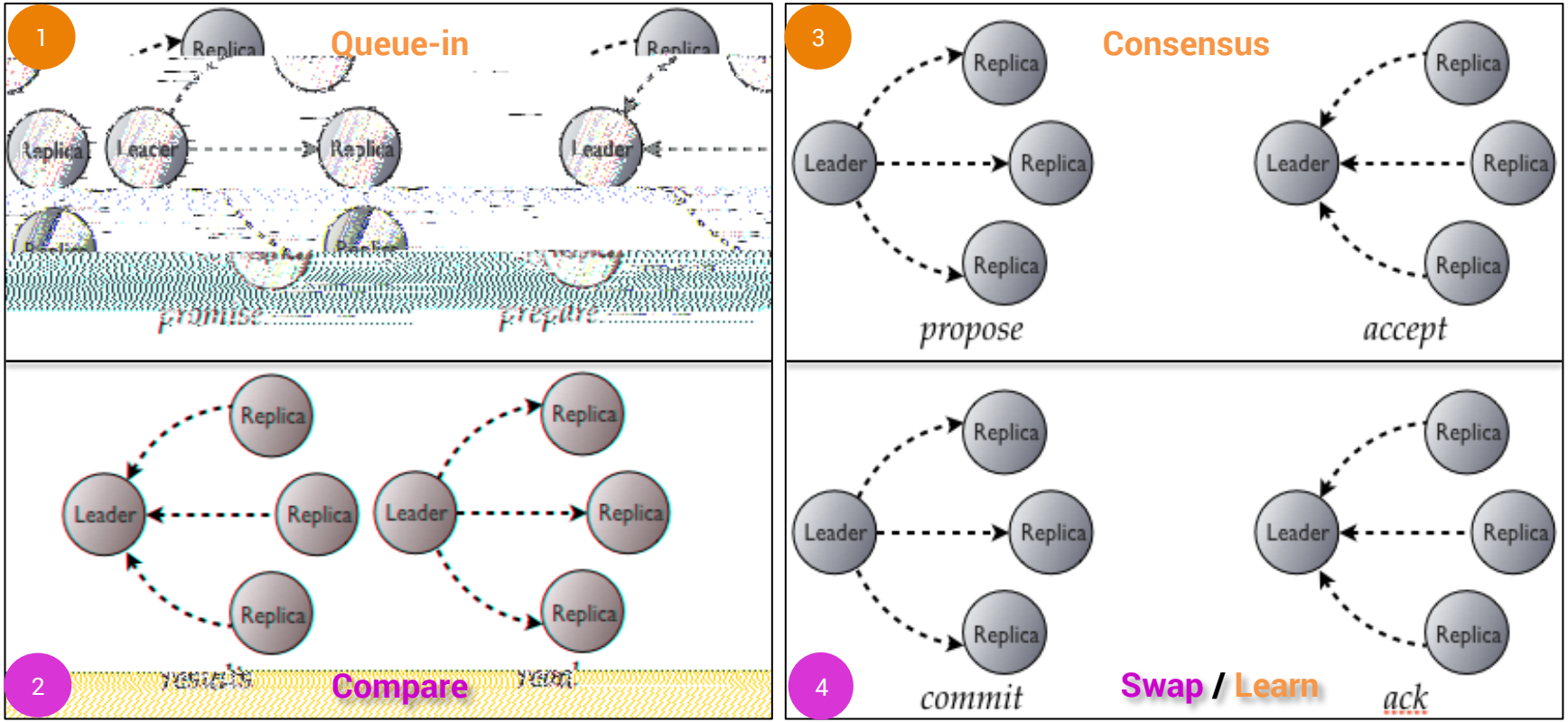
Lightweight Transaction (LWT)

DATASTAX

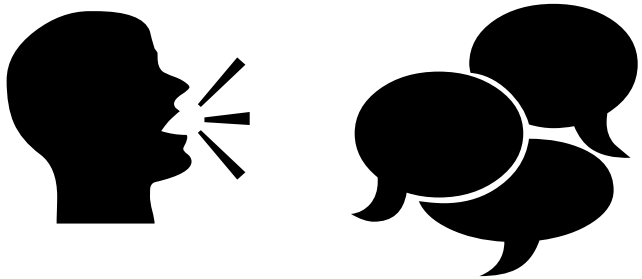
- (4 -), do not abuse
- 1% – 5%

Lightweight Transaction (LWT)

DATASTAX



DATASTAX



Q & R

Thank You

@doanduyhai

duy_hai.doan@datastax.com

https://academy.datastax.com/