



Intro to Cassandra for Developers

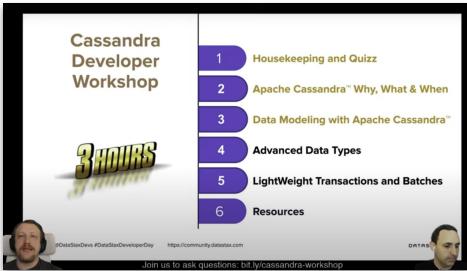


LEVEL
UP
with the

DataStax
Developers

Housekeeping

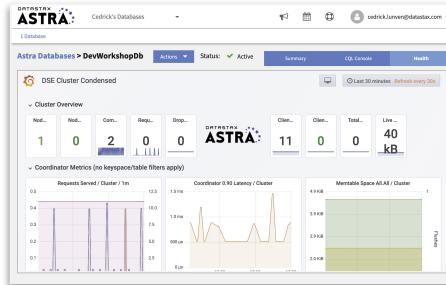
Courses: youtube.com/DataStaxDevs



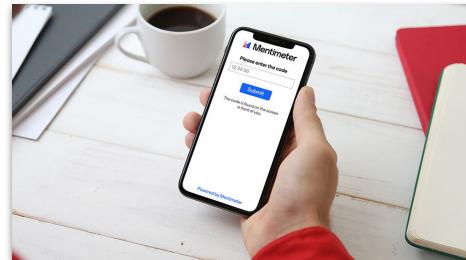
Questions: bit.ly/cassandra-workshop



Runtime: dtsx.io/workshop



Quizz: menti.com



Achievement Unlocked! – “Introduction to Cassandra”



K8ssandra Workshop

Awarded to **Sylwester Lachiewicz** • slachiewicz@gmail.com
Issued on Mar 5, 2021

Offered By
DataStax Developers

Upgrade Complete! This badge is to certify successful completion of the DataStax K8ssandra Workshop: "Running Apache Cassandra on Kubernetes".

 **Verified**
Last verified by Badgr on Mar 31, 2021

[Re-verify Badge](#)

EARNING CRITERIA
Recipients must complete the earning criteria to earn this badge

To earn this badge, individuals must complete the following steps during the **K8ssandra Workshop**:

- Attend the lecture
- Complete the practical steps by doing all required exercises

[View External Criteria](#)

TAGS

kubernetes
cassandra



Homework



Cassandra Query Language (CQL)

Learn the essential CQL commands you need to get started with Apache Cassandra™

[START CODING](#)



Cassandra Data Modeling

So, you want to create a Cassandra schema? Cassandra's schema development methodology is differen...

[START CODING](#)



==



Fully managed Cassandra
Without the ops!

DataStax Astra



Global Scale

Put your data where you need it without compromising performance, availability, or accessibility.



No Operations

Eliminate the overhead to install, operate, and scale Cassandra.



25 Gig Free Tier

Launch a database in the cloud with a few clicks, no credit card required.

menti.com



Go to www.menti.com and use the code 3491 9972

Inequality predicates are allowed on ...

A bar chart titled "Inequality predicates are allowed on ...". The y-axis represents the number of inequality predicates allowed, ranging from 1 to 15. The x-axis categories are "All table columns", "Partition key columns", "clustering key columns", and "No inequality predicates are allowed".

Column Type	Number of inequality predicates allowed
All table columns	4
Partition key columns	3
clustering key columns	15
No inequality predicates are allowed	1

Below the chart, there is a video player interface showing a video of a person speaking. The video player includes controls like play/pause, volume, and a progress bar.

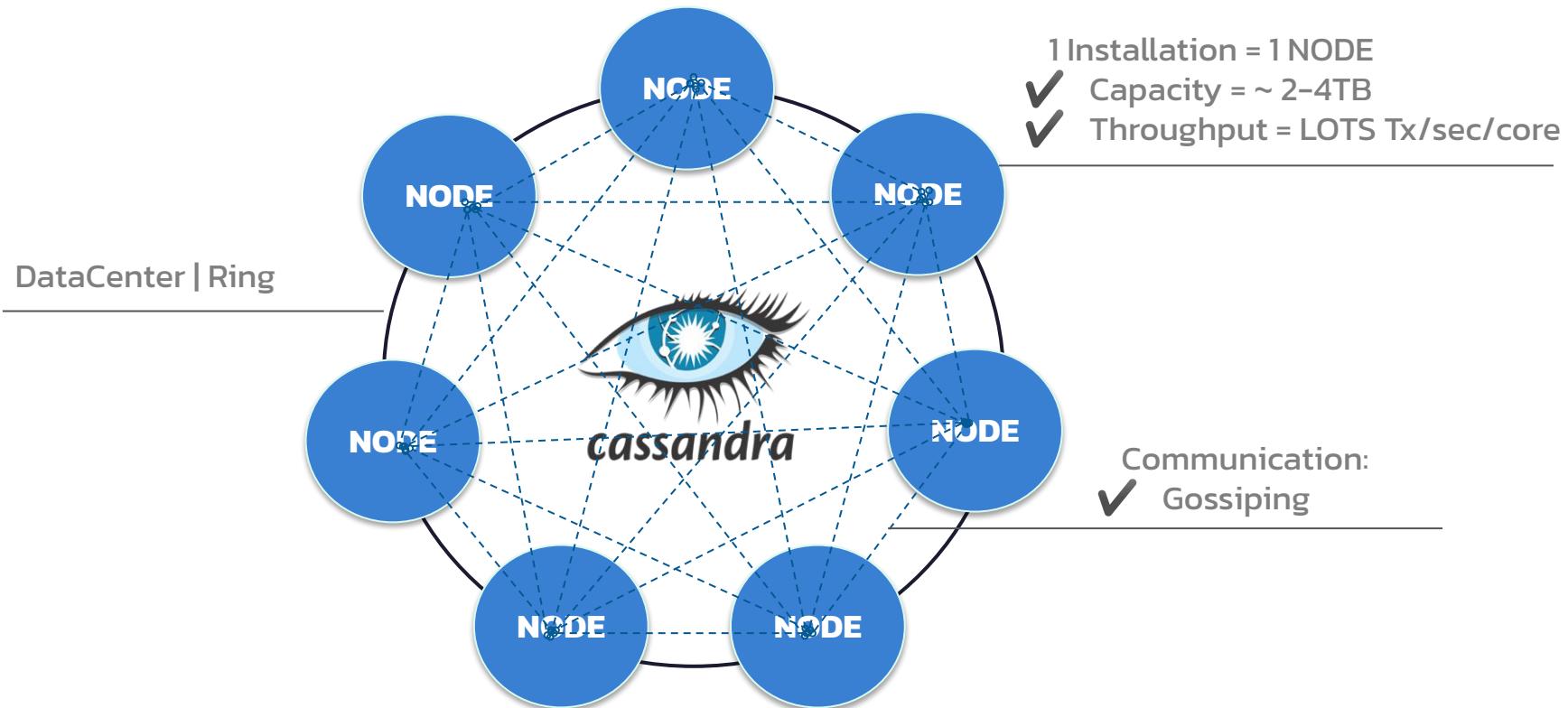
Go to www.menti.com and use the code 3491 9972

Leaderboard

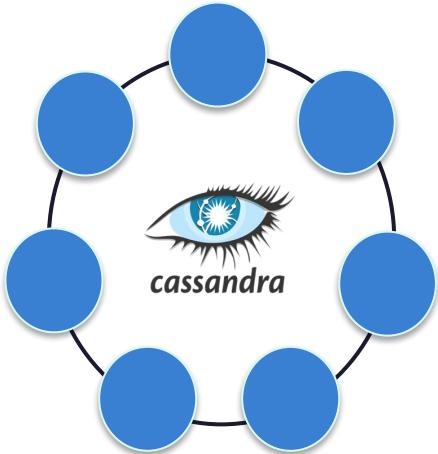
User ID	User Name	Profile Icon
4821 p	spanda	Avatar icon
4820 p	Agent X9	Avatar icon
4775 p	Sam	Avatar icon
4711 p	CCedrickThePresenter	Avatar icon
4468 p	shubham	Avatar icon
4371 p	aaa	Avatar icon
3895 p	vignesh	Avatar icon
3877 p	adry	Avatar icon
3861 p	Millie	Avatar icon
3812 p	Puggie	Avatar icon

Below the leaderboard, there is a video player interface showing a video of a person speaking. The video player includes controls like play/pause, volume, and a progress bar.

Apache Cassandra™ = NoSQL Distributed Database

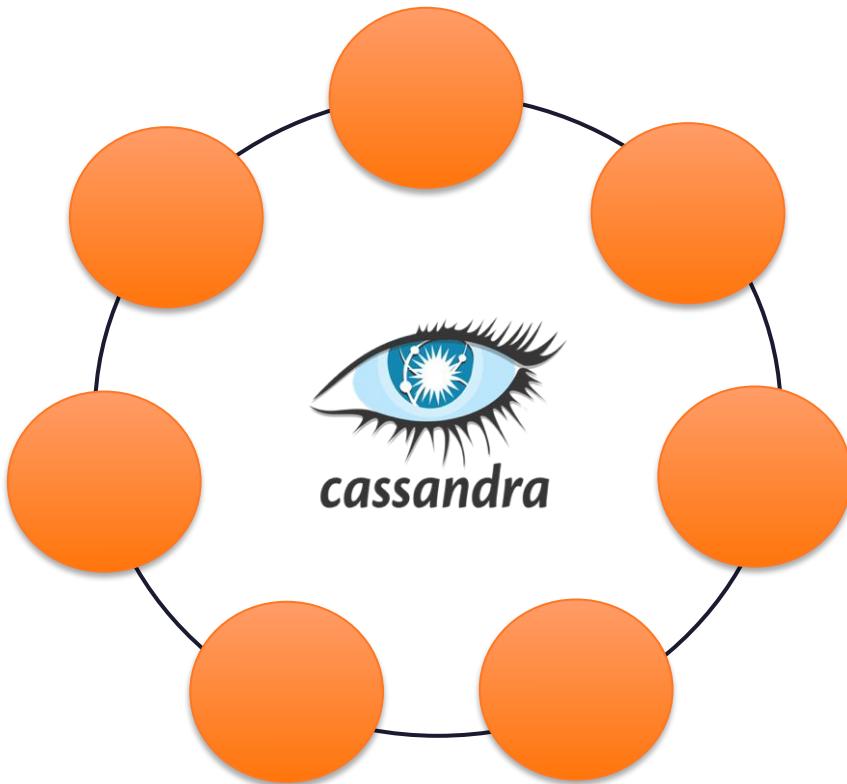


Apache Cassandra™ = NoSQL Distributed Database



- Big Data Ready
- Highest Availability
- Geographical Distribution
- Read/Write Performance
- Vendor Independent

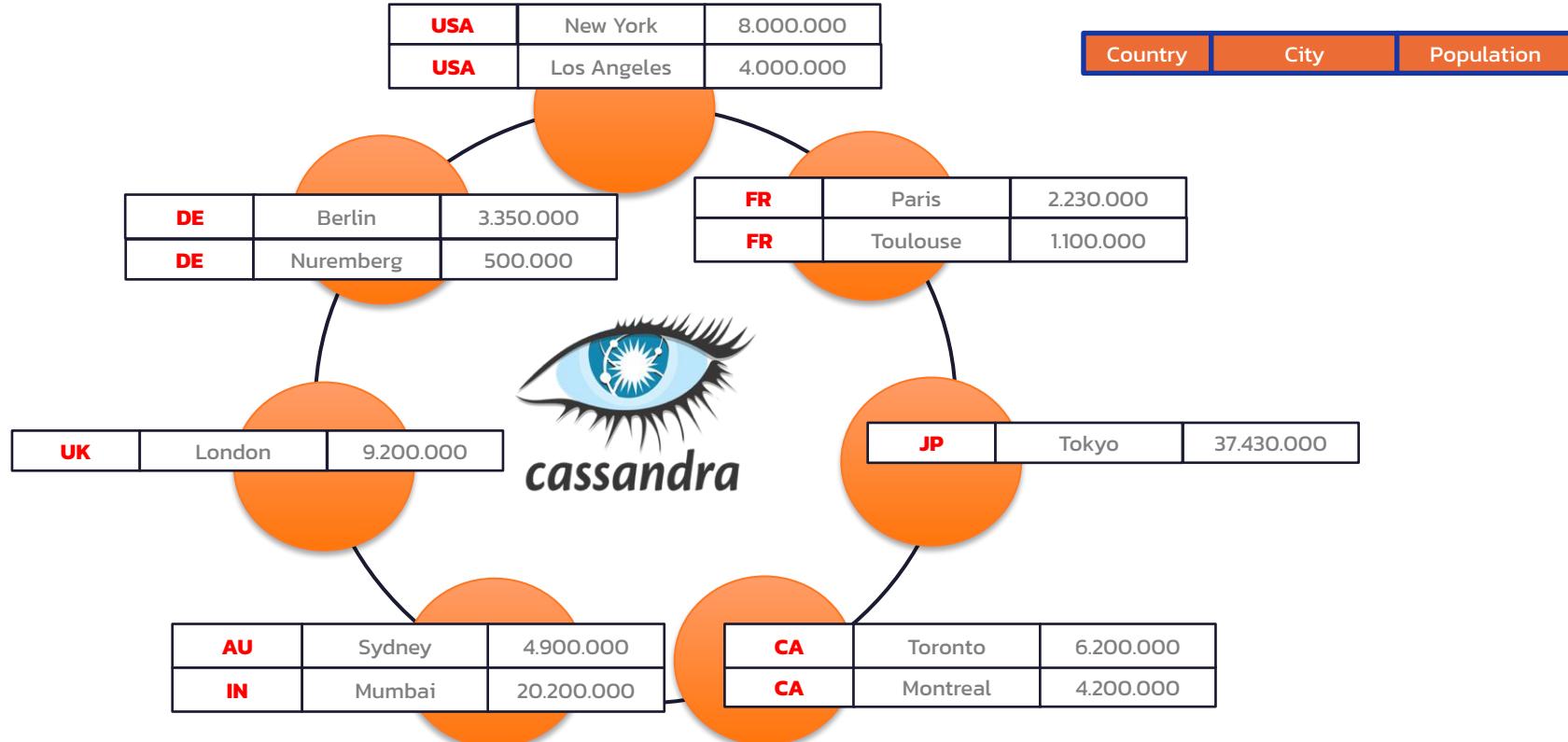
Data is Distributed



Country	City	Population
USA	New York	8.000.000
USA	Los Angeles	4.000.000
FR	Paris	2.230.000
DE	Berlin	3.350.000
UK	London	9.200.000
AU	Sydney	4.900.000
DE	Nuremberg	500.000
CA	Toronto	6.200.000
CA	Montreal	4.200.000
FR	Toulouse	1.100.000
JP	Tokyo	37.430.000
IN	Mumbai	20.200.000

Partition Key

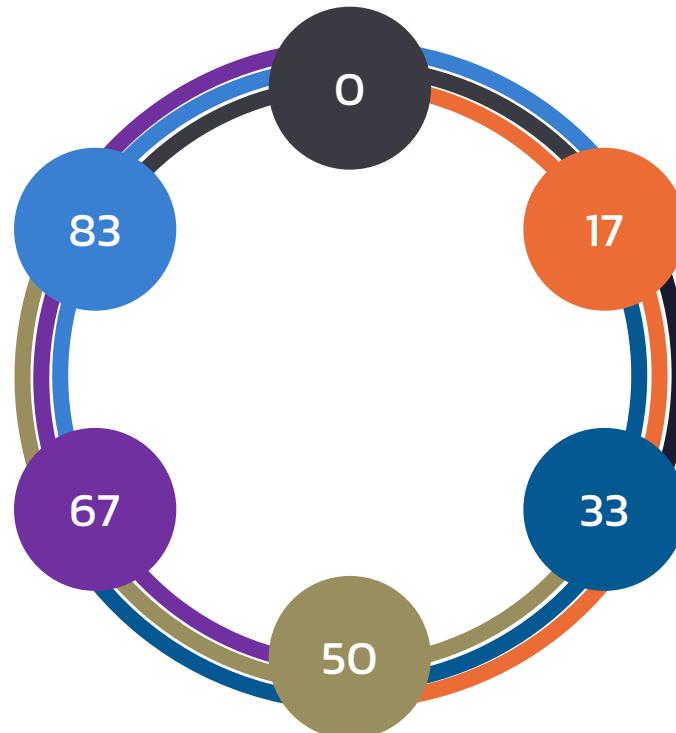
Data is Distributed



Data is Replicated

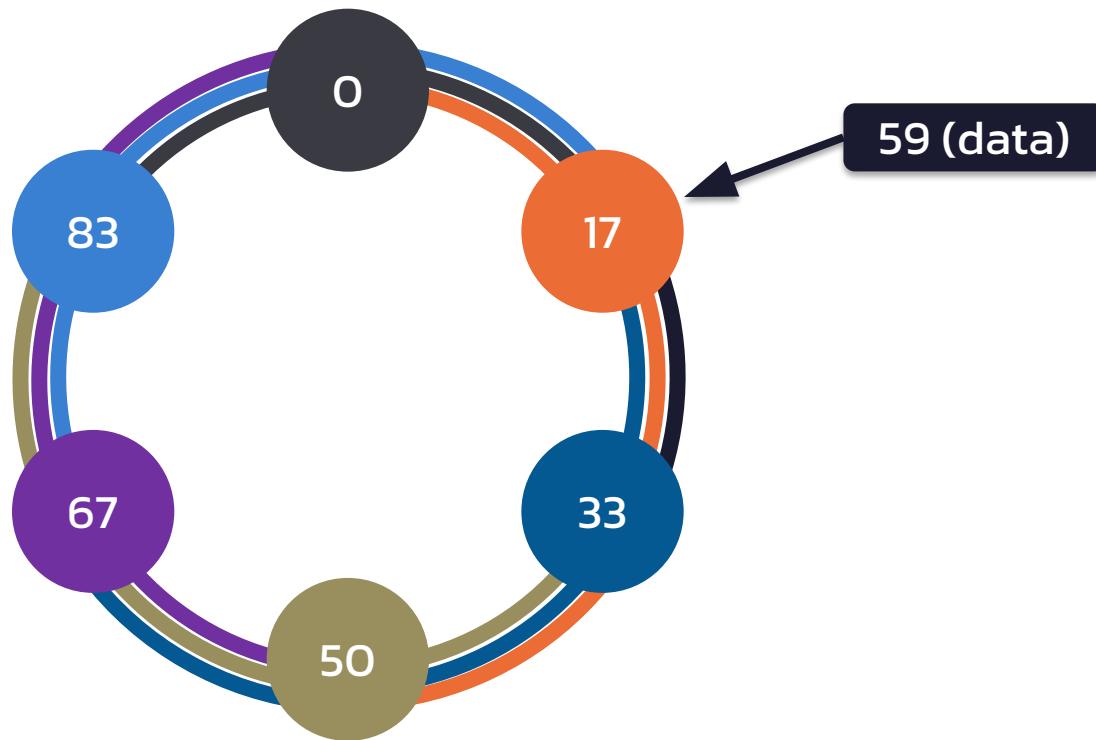
RF = 3

Replication Factor 3
means that every
row is stored on 3
different nodes



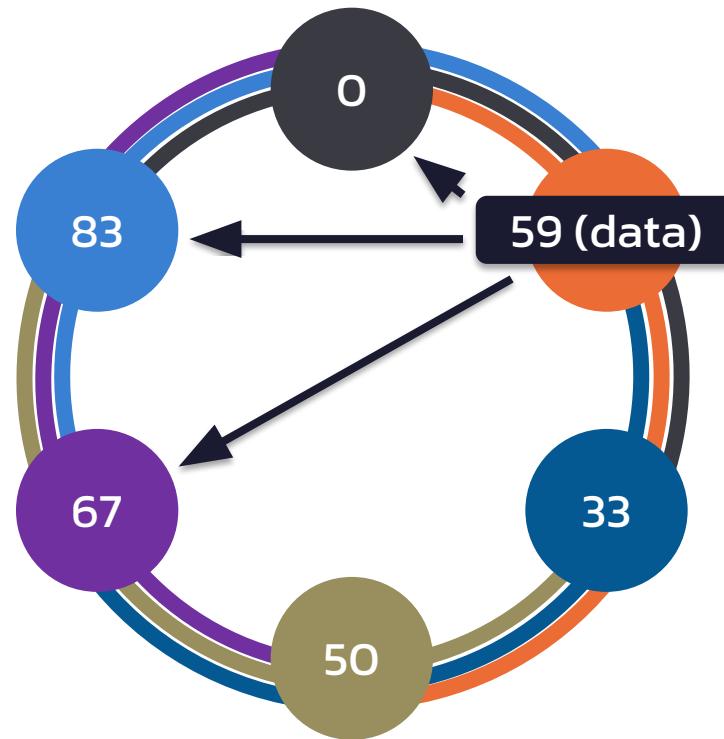
Replication within the Ring

RF = 3

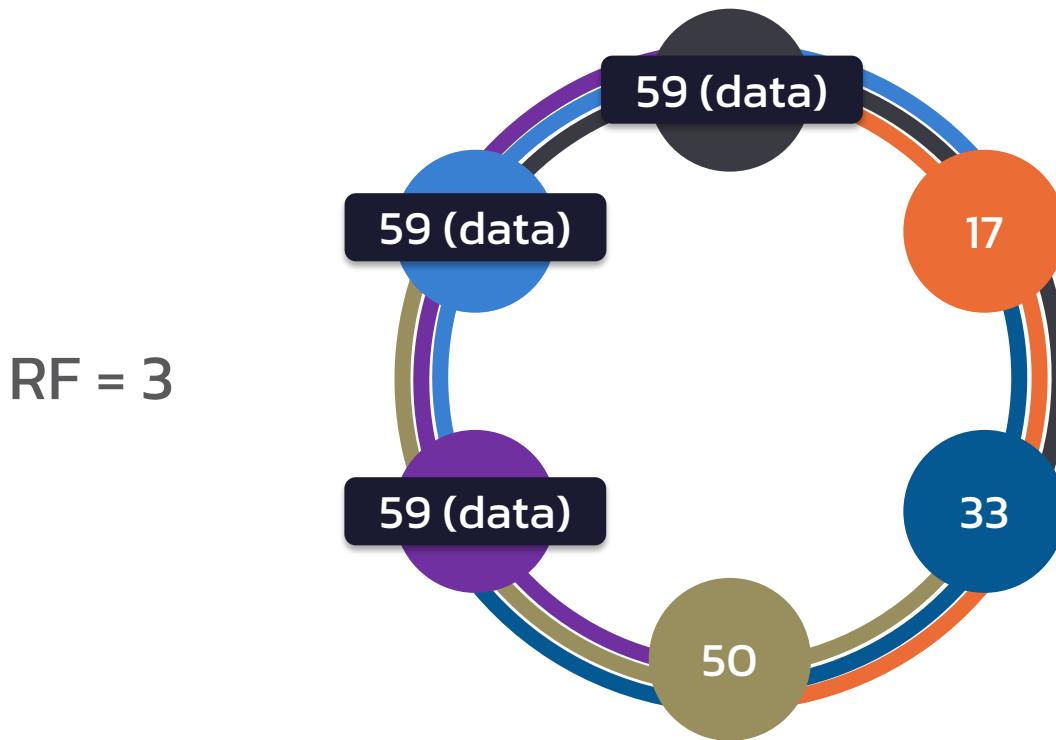


Replication within the Ring

RF = 3

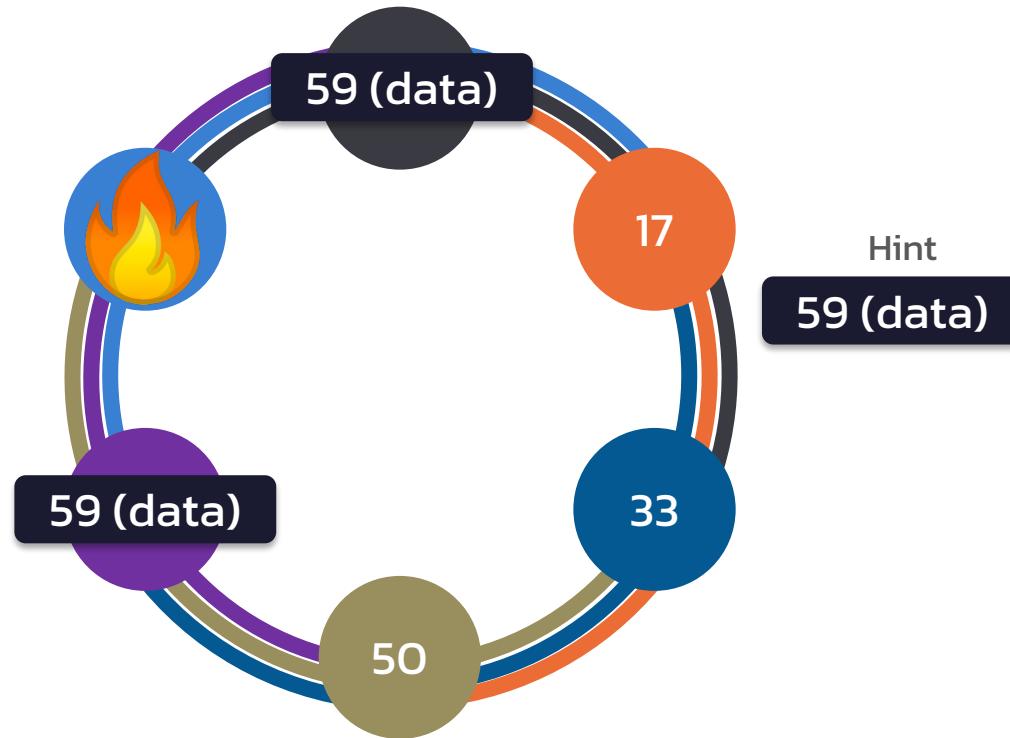


Replication within the Ring

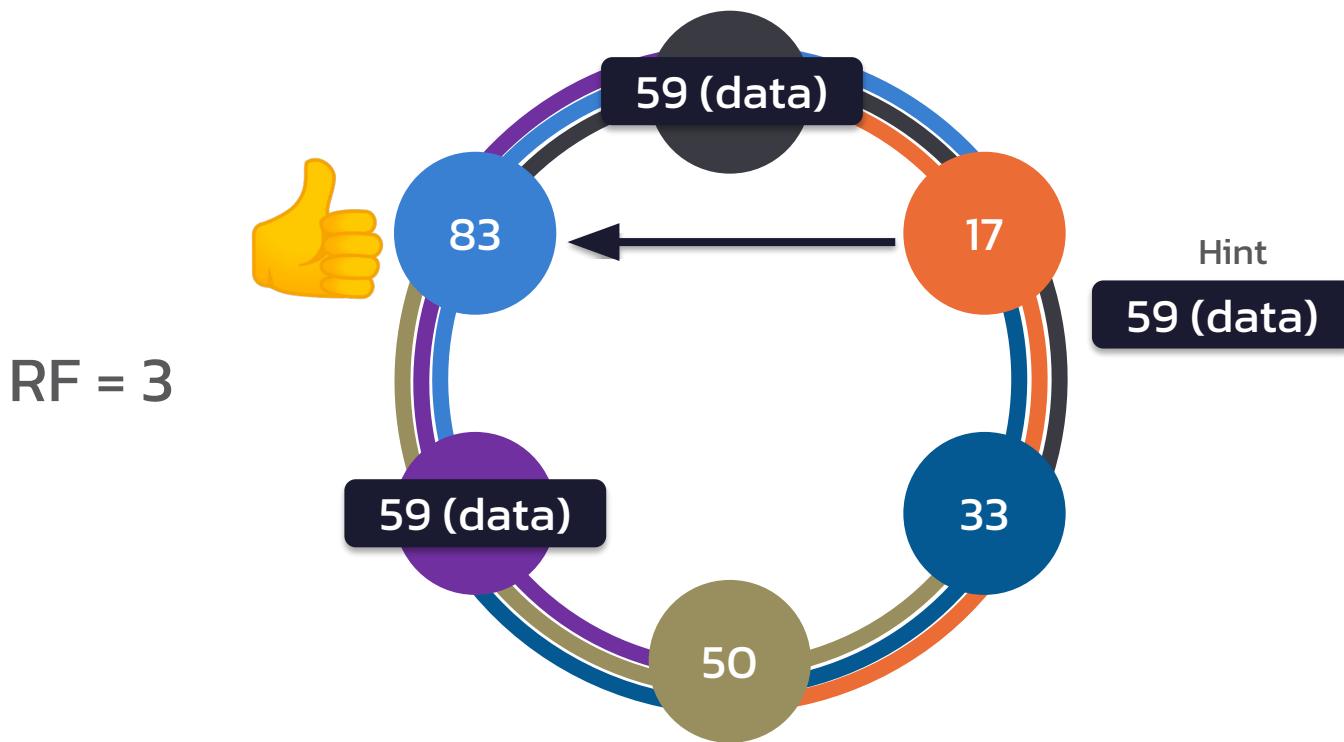


Node Failure

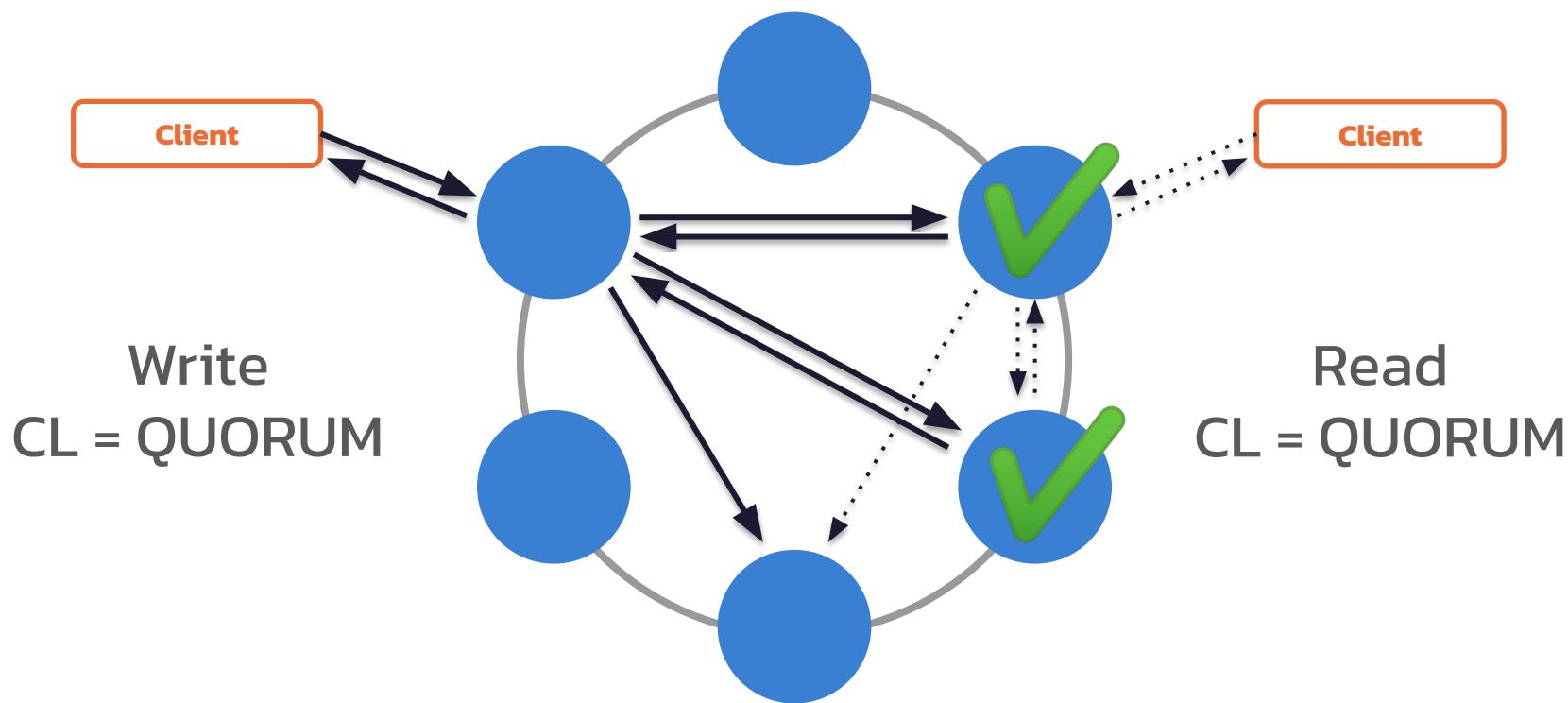
RF = 3



Node Failure Recovered

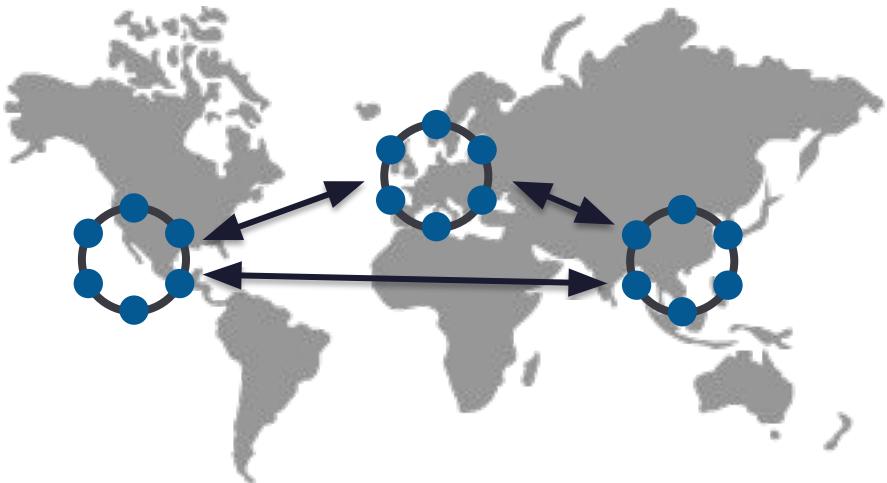


Immediate Consistency – A Better Way

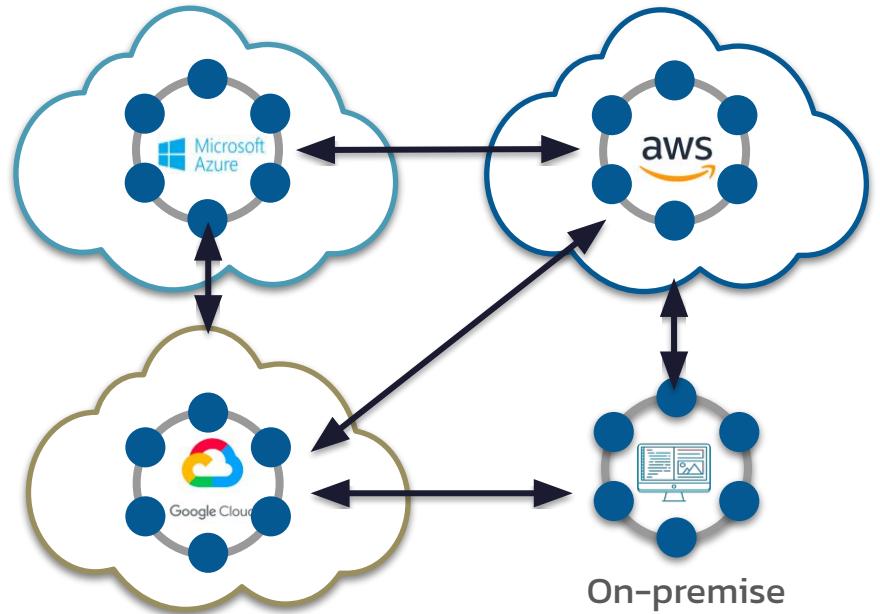


Data Distributed Everywhere

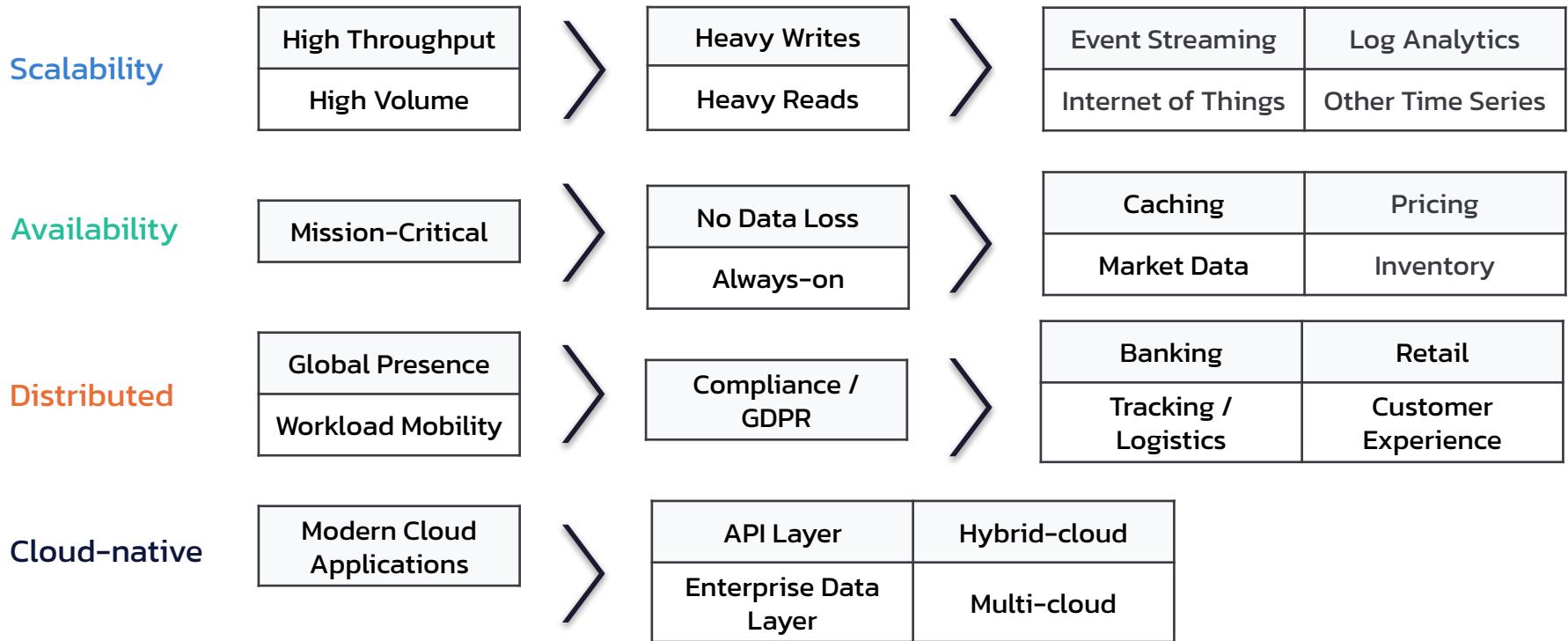
- Geographic Distribution



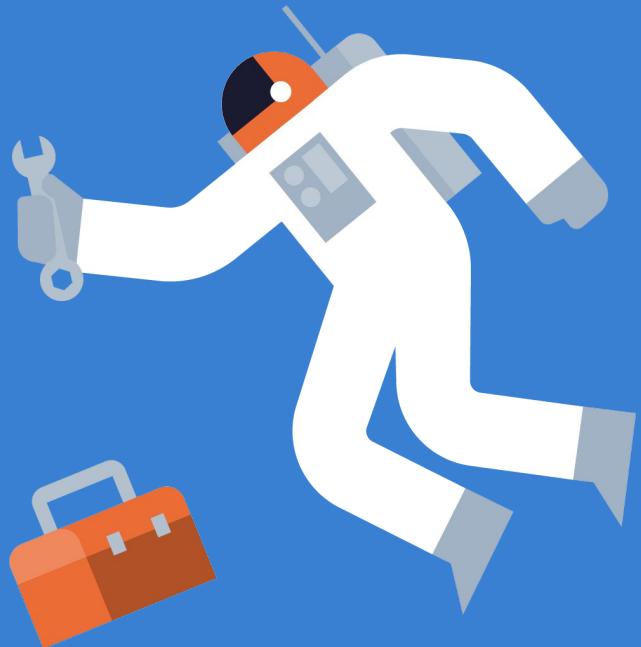
- Hybrid-Cloud and Multi-Cloud



Understanding Use Cases

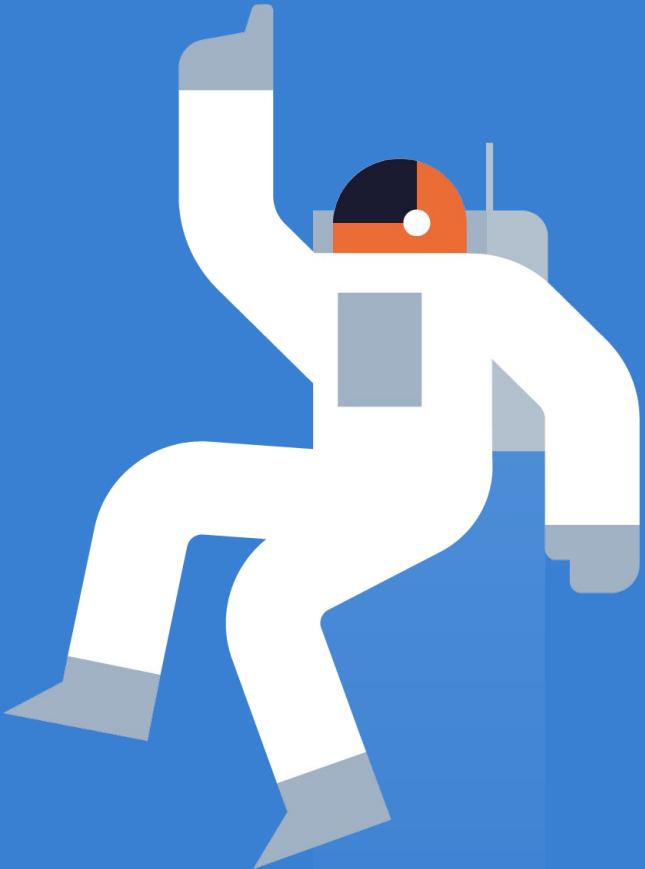


[https://github.com/DataStax-Academy
/Intro-to-Cassandra-for-Developers](https://github.com/DataStax-Academy/Intro-to-Cassandra-for-Developers)



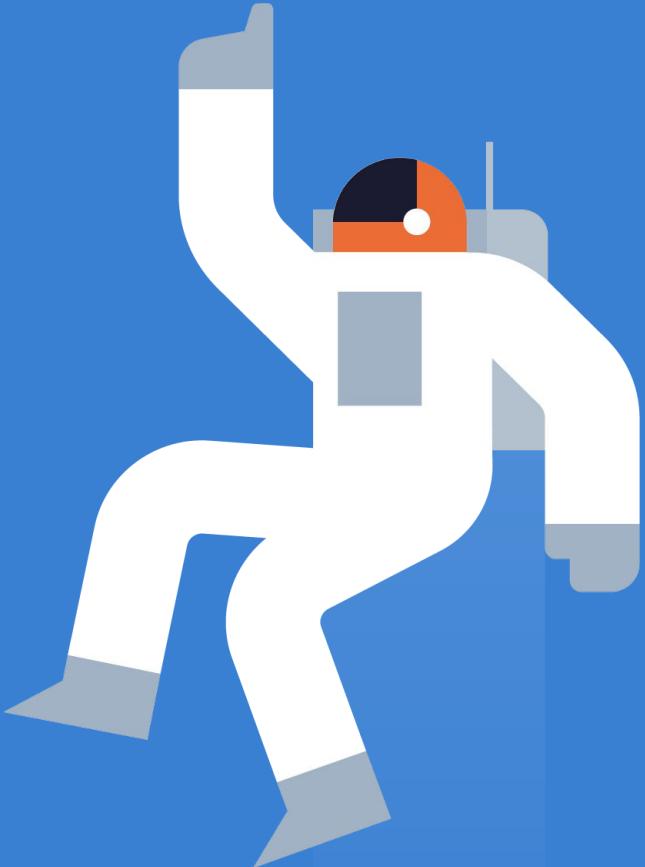
Intro to Cassandra for Developers

1. Tables, Partitions
2. The Art of Data Modelling
3. What's NEXT?



Intro to Cassandra for Developers

1. Tables, Partitions
2. The Art of Data Modelling
3. What's NEXT?



Data Structure: a Cell



An intersection of a row
and a column, stores data.



Data Structure: a Row



A single, structured
data item in a table.

1	John	Doe	Wizardry
---	------	-----	----------

Data Structure: a Partition



A group of rows having the same partition token, a base unit of access in Cassandra.

IMPORTANT: stored together, all the rows are guaranteed to be neighbors.

ID	First Name	Last Name	Department
1	John	Doe	Wizardry
399	Marisha	Chavez	Wizardry
415	Maximus	Flavius	Wizardry

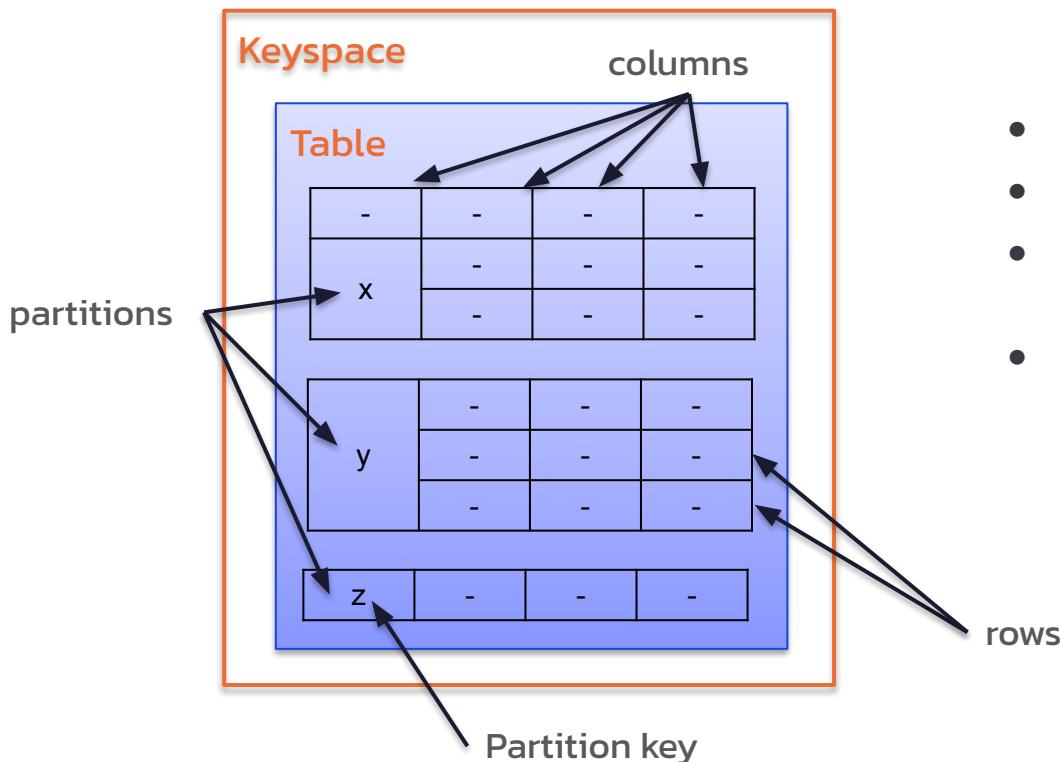
Data Structure: a Table



A group of columns and rows storing partitions.

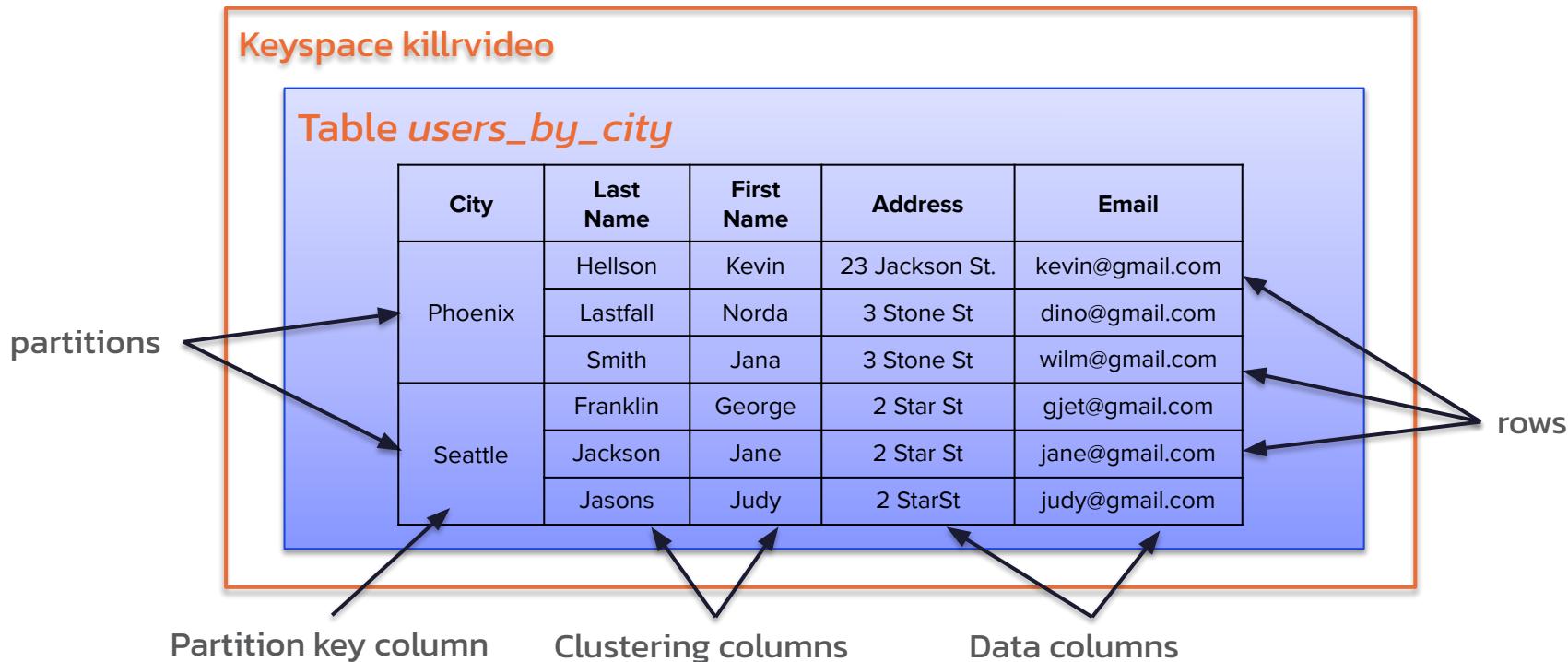
ID	First Name	Last Name	Department
1	John	Doe	Wizardry
2	Mary	Smith	Dark Magic
3	Patrick	McFadin	DevRel

Data Structure: Overall



- Tabular data model, with one twist
- *Tables* are organized in *rows* and *columns*
- Groups of related rows called *partitions* are stored together on the same node (or nodes)
- Each row contains a *partition key*
 - One or more columns that are hashed to determine which node(s) store that data

Example Data: Users organized by city



Creating a Table in CQL



Primary Key

An identifier for a row. Consists of at least one Partition Key and zero or more Clustering Columns.

MUST ENSURE UNIQUENESS.
MAY DEFINE SORTING.

```
CREATE TABLE killrvideo.users_by_city (
    city text,
    last_name text,
    first_name text,
    address text,
    email text,
    PRIMARY KEY ((city), last_name, first_name, email));
```



Good Examples:

```
PRIMARY KEY ((city), last_name, first_name, email);
```

```
PRIMARY KEY (user_id);
```

Bad Example:

```
PRIMARY KEY ((city), last_name, first_name);
```

Partition Key

An identifier for a partition.

Consists of at least one column,
may have more if needed

PARTITIONS ROWS.

```
CREATE TABLE killrvideo.users_by_city (
    city text,
    last_name text,
    first_name text,
    address text,
    email text,
    PRIMARY KEY ((city), last_name, first_name, email));
```



Good Examples:

```
PRIMARY KEY (user_id);
```

```
PRIMARY KEY ((video_id), comment_id);
```

Bad Example:

```
PRIMARY KEY ((sensor_id), logged_at);
```

Clustering Column(s)

Used to ensure uniqueness and sorting order. Optional.

```
CREATE TABLE killrvideo.users_by_city (
    city text,
    last_name text,
    first_name text,
    address text,
    email text,
    PRIMARY KEY ((city), last_name, first_name, email));
```

Partition key

Clustering columns

PRIMARY KEY ((city), last_name, first_name);



Not Unique

PRIMARY KEY ((city), last_name, first_name, email);



PRIMARY KEY ((video_id), comment_id);



Not Sorted

PRIMARY KEY ((video_id), created_at, comment_id);



Rules of a Good Partition

- **Store together what you retrieve together**
- Avoid big partitions
- Avoid hot partitions

Example: open a video? Get the comments in a single query!

```
PRIMARY KEY ((video_id), created_at, comment_id);
```



```
PRIMARY KEY ((comment_id), created_at);
```



Rules of a Good Partition

- Store together what you retrieve together
- **Avoid big partitions**
- Avoid hot partitions

```
PRIMARY KEY ((video_id), created_at, comment_id);
```



```
PRIMARY KEY ((country), user_id);
```



- Up to 2 billion cells per partition
- Up to ~100k rows in a partition
- Up to ~100MB in a Partition

Rules of a Good Partition

- Store together what you retrieve together
- **Avoid big and constantly growing partitions!**
- Avoid hot partitions

Example: a huge IoT infrastructure, hardware all over the world, different sensors reporting their state every 10 seconds. Every sensor reports its UUID, timestamp of the report, sensor's value.

- Sensor ID: UUID
- Timestamp: Timestamp
- Value: float

```
PRIMARY KEY ((sensor_id), reported_at);
```



Rules of a Good Partition

- Store together what you retrieve together
- **Avoid big and constantly growing partitions!**
- Avoid hot partitions

Example: a huge IoT infrastructure, hardware all over the world, different sensors reporting their state every 10 seconds. Every sensor reports its UUID, timestamp of the report, sensor's value.

```
PRIMARY KEY ((sensor_id), reported_at);
```



```
PRIMARY KEY ((sensor_id, month_year), reported_at);
```



BUCKETING

- Sensor ID: UUID
- **MonthYear:** Integer or String
- Timestamp: Timestamp
- Value: float

Rules of a Good Partition

- Store together what you retrieve together
- Avoid big partitions
- **Avoid hot partitions**

```
PRIMARY KEY (user_id);
```



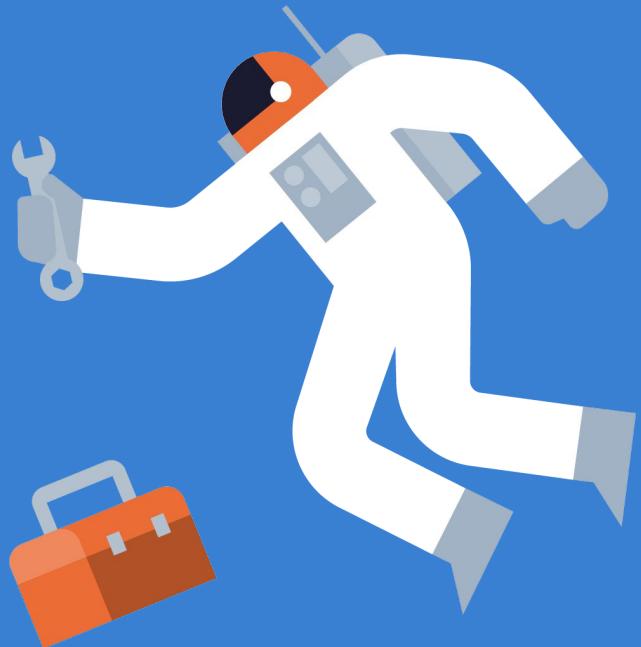
```
PRIMARY KEY ((video_id), created_at, comment_id);
```



```
PRIMARY KEY ((country), user_id);
```

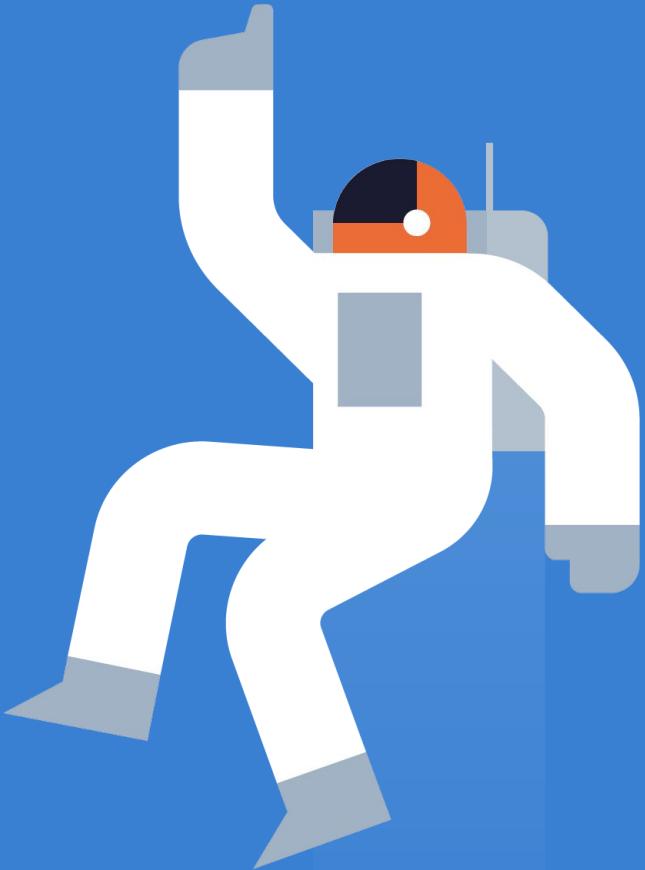


<https://github.com/DataStax-Academy/Intro-to-Cassandra-for-Developers#2-create-a-table>



Intro to Cassandra for Developers

1. Tables, Partitions
2. The Art of Data Modelling
3. What's NEXT?



Normalization

"Database normalization is the process of structuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. It was first proposed by Edgar F. Codd as part of his relational model."

PROS: Simple write, Data Integrity
CONS: Slow read, Complex Queries



Employees			
userId	deptId	firstName	lastName
1	1	Edgar	Codd
2	1	Raymond	Boyce

Departments	
departmentId	department
1	Engineering
2	Math

Denormalization

"Denormalization is a strategy used on a database to increase performance. In computing, denormalization is the process of trying to improve the read performance of a database, at the expense of losing some write performance, by adding redundant copies of data"

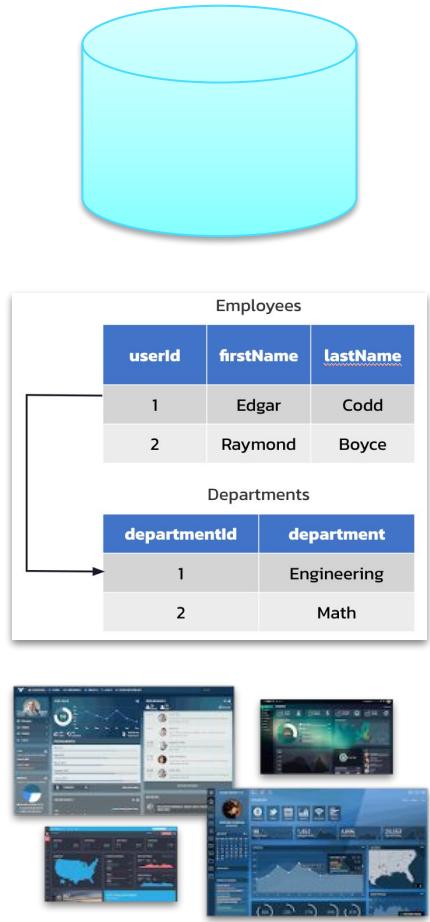
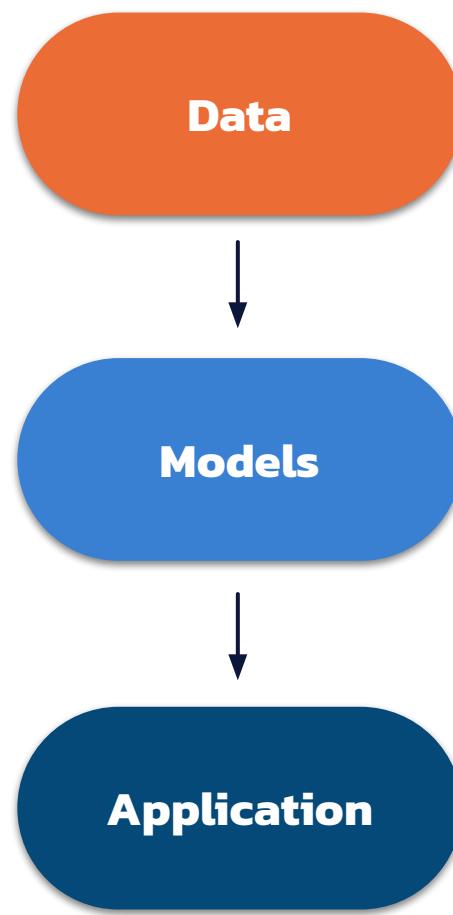
PROS: Quick Read, Simple Queries

CONS: Multiple Writes, Manual Integrity

Employees			
userId	firstName	lastName	department
1	Edgar	Codd	Engineering
2	Raymond	Boyce	Engineering
3	Sage	Lahja	Math
4	Juniper	Jones	Botany

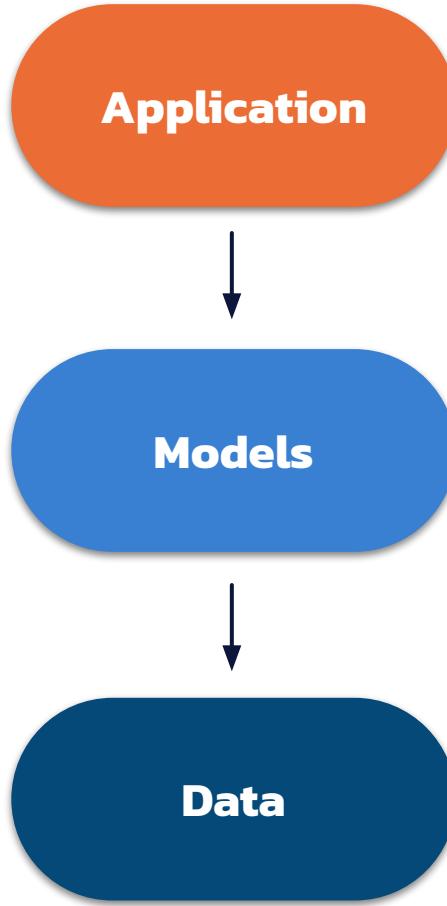
Relational Data Modelling

1. Analyze raw data
2. Identify entities, their properties and relations
3. Design tables, using **normalization** and foreign keys.
4. Use JOIN when doing queries to join normalized data from multiple tables

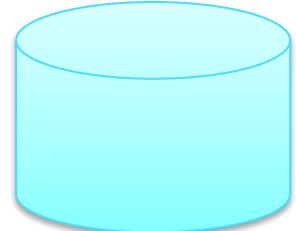


NoSQL Data Modelling

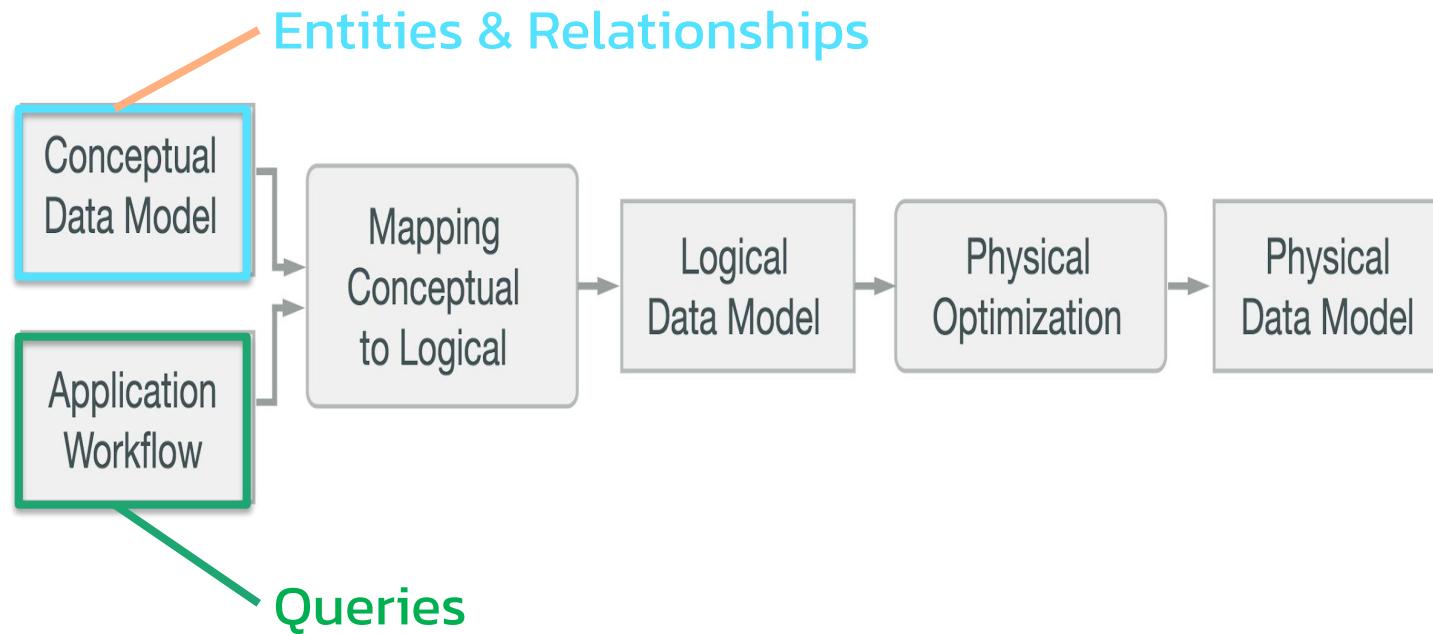
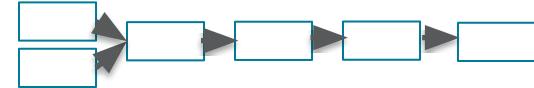
1. Analyze user behaviour
(customer first!)
2. Identify workflows, their dependencies and needs
3. Define Queries to fulfill these workflows
4. Knowing the queries, design tables, using **denormalization**.
5. Use BATCH when inserting or updating denormalized data of multiple tables



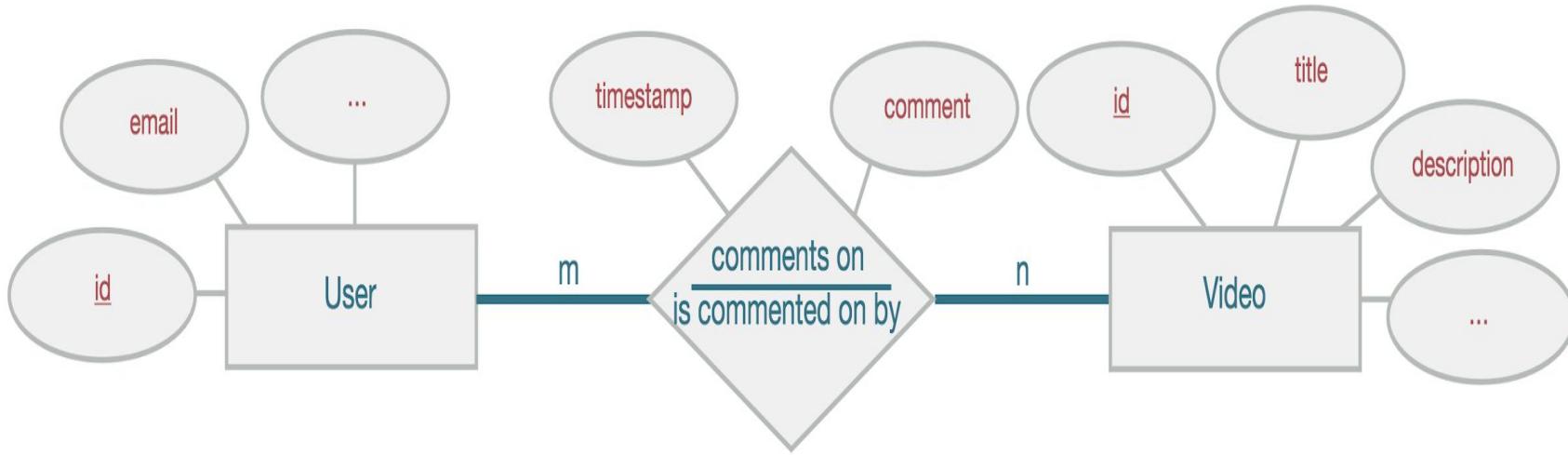
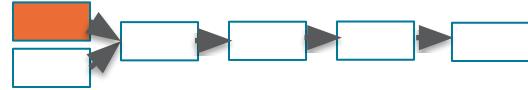
Employees			
userId	firstName	lastName	department
1	Edgar	Codd	Engineering
2	Raymond	Boyce	Math
3	Sage	Lahja	Math
4	Juniper	Jones	Botany



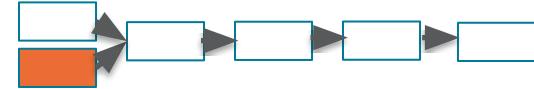
Designing Process: Step by Step



Designing Process: Conceptual Data Model



Designing Process: Application Workflow



Use-Case I:

- A User opens a Profile

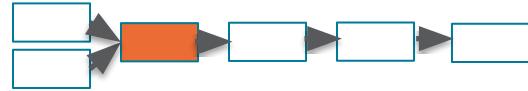
WF2: Find **comments** related to target **user** using its identifier, get most recent first

Use-Case II:

- A User opens a Video Page

WF1: Find **comments** related to target **video** using its identifier, most recent first

Designing Process: Mapping



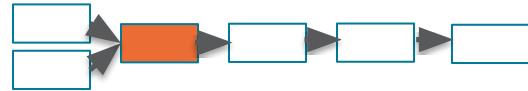
Query I: Find comments posted for a user with a known id (show most recent first)



Query II: Find comments for a video with a known id (show most recent first)



Designing Process: Mapping



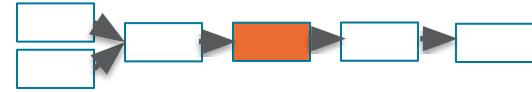
```
SELECT * FROM comments_by_user  
WHERE userid = <some UUID>
```



```
SELECT * FROM comments_by_video  
WHERE videoid = <some UUID>
```



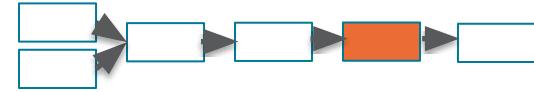
Designing Process: Logical Data Model



comments_by_user	
userid	K
creationdate	C ↓
commentid	C ↑
videoid	
comment	

comments_by_video	
videoid	K
creationdate	C ↓
commentid	C ↑
userid	
comment	

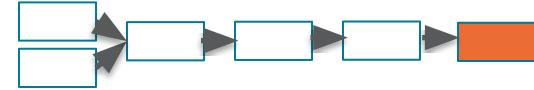
Designing Process: Physical Data Model



comments_by_user		
userid	UUID	K
commentid	TIMEUUID	C↓
videoid	UUID	
comment	TEXT	

comments_by_video		
videoid	UUID	K
commentid	TIMEUUID	C↓
userid	UUID	
comment	TEXT	

Designing Process: Schema DDL



```
CREATE TABLE IF NOT EXISTS comments_by_user (
    userid uuid,
    commentid timeuuid,
    videoid uuid,
    comment text,
    PRIMARY KEY ((userid), commentid)
) WITH CLUSTERING ORDER BY (commentid DESC);
```

```
CREATE TABLE IF NOT EXISTS comments_by_video (
    videoid uuid,
    commentid timeuuid,
    userid uuid,
    comment text,
    PRIMARY KEY ((videoid), commentid)
) WITH CLUSTERING ORDER BY (commentid DESC);
```

<https://github.com/DataStax-Academy/Intro-to-Cassandra-for-Developers#3-execute-crud-operations>



menti.com



Go to www.menti.com and use the code 3491 9972

Inequality predicates are allowed on ...

A bar chart titled "Inequality predicates are allowed on ...". The y-axis represents the number of inequality predicates allowed, ranging from 1 to 15. The x-axis categories are "All table columns", "Partition key columns", "clustering key columns", and "No inequality predicates are allowed".

Column Type	Number of inequality predicates allowed
All table columns	4
Partition key columns	3
clustering key columns	15
No inequality predicates are allowed	1

Below the chart, there is a video player interface showing a video of a man speaking. The video player includes controls like play/pause, volume, and a progress bar indicating 2:10:19 / 2:26:05. The title of the video is "Big paycheck".

Go to www.menti.com and use the code 3491 9972

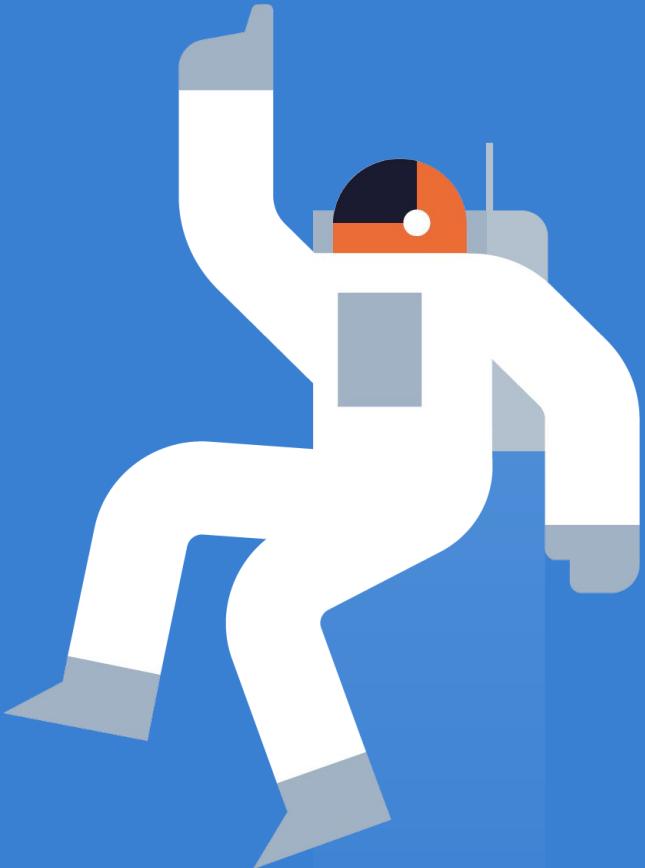
Leaderboard

User ID	User Name	Profile Icon
4821 p	spanda	Avatar of a green sphere
4820 p	Agent X9	Avatar of a black mask
4775 p	Sam	Avatar of a cat-like face
4711 p	CCedrickThePresenter	Avatar of a green leaf
4468 p	shubham	Avatar of a yellow bird
4371 p	aaa	Avatar of a yellow cat
3895 p	vignesh	Avatar of a red crown
3877 p	adry	Avatar of a smiling face
3861 p	Millie	Avatar of a brown bear
3812 p	Puggie	Avatar of a brown dog

Below the leaderboard, there is a video player interface showing a video of a man speaking. The video player includes controls like play/pause, volume, and a progress bar indicating 2:11:07 / 2:26:05. The title of the video is "DataStax".

Intro to Cassandra for Developers

1. Tables, Partitions
2. The Art of Data Modelling
3. What's NEXT?



Homework



Cassandra Query Language (CQL)

Learn the essential CQL commands you need to get started with Apache Cassandra™

[START CODING](#)

Cassandra Data Modeling

So, you want to create a Cassandra schema? Cassandra's schema development methodology is differen...

[START CODING](#)

MORE LEARNING!!!!

Developer site: [datastax.com/dev](https://www.datastax.com/dev)

- **Developer Stories**
- New hands-on learning scenarios with Katacoda
 - Try it Out
 - Cassandra Fundamentals
 - <https://www.datastax.com/learn/cassandra-fundamentals>
 - New Data Modeling course
<https://www.datastax.com/dev/modeling>

Classic courses available at [DataStax Academy](#)



Katacoda

Developers

The screenshot shows the DataStax Developers YouTube channel page. At the top, there's a banner with the text "LEVEL UP with the DataStax Developers". Below the banner, the channel name "DataStax Developers" is displayed with 8.47K subscribers. There are buttons for "CUSTOMIZE CHANNEL" and "MANAGE VIDEOS". The main content area features a "Leaderboard" section with a list of names and profile pictures, and a "Level up with the DataStax Developers!" video thumbnail. Below this, there's a "Sign up to our Event Alert mailing list" section with a link. The bottom part of the screenshot shows "Upcoming live streams" with four items: "Cassandra + Quarkus!", "Implement & Consume APIs", "Bootiful Cassandra!", and "Build Microservices with Cassandra and Quarkus!". Each stream has a thumbnail, a title, a "LIVE" indicator, and a "BEGGINER TOPIC" label.

- ✓ Academy.datastax.com
- ✓ datastax.com/dev
- ✓ community.datastax.com
- ✓ Datastax Developers
YouTube Channel

Weekly Workshops

<https://www.datastax.com/workshops>

The image shows a composite view of the DataStax Developers YouTube channel and its website.

YouTube Channel: The left side displays the official YouTube channel page for "DataStax Developers". It features a banner with the "LEVEL UP with the DataStax Developers" logo and a white astronaut icon. Below the banner, there's a video thumbnail for "Microservices with Cassandra + Spring" and a grid of smaller video thumbnails for various workshops like "Advanced Data Modeling" and "Apache Cassandra™ Certification Preparation". The channel has 8,1k subscribers.

Website: The right side shows the "Upcoming Live Events" section of the DataStax Developers website. It lists five live workshops with details and registration links:

- Apache Cassandra™ Certification Preparation**: Multiple Dates | NoSQL | Beginner. [Register Now](#)
- Build Microservices with Apache Cassandra™!**: Feb 17 or Feb 18 | NoSQL | Beginner. [Register Now](#)
- Certification Exam Preparation Workshop**: MULTIPLE DATES. [Register Now](#)
- Cloud-Native Workshop: Build Spring Microservices with Apache Cassandra™**: MULTIPLE DATES. [Register Now](#)
- Learn how to build a Serverless Game!**: Feb 24 or Feb 25 | Game Development | Beginner. [START](#) button. [Register Now](#)
- Build Microservices with Cassandra & Quarkus**: March 11 | Microservices | Beginner. [Register Now](#)

Join our 10k Discord Community

The Fellowship of the RINGS

<https://bit.ly/cassandra-workshop>

thank you everyone! we hope to see you again in a future workshop! 🎉

Jack Fryer Aujourd'hui à 14:30
NEW WORKSHOP ALERT

Cassandra meets Kubernetes!

Come and meet K8ssandra! A cloud-native distribution of Apache Cassandra™ built for running on Kubernetes

<https://www.eventbrite.co.uk/e/cloud-native-workshop-connecting-cassandra-and-kubernetes-tickets-142078180663>

Eventbrite

Cloud-Native Workshop: Connecting Cassandra and Kubernetes!

Come and meet K8ssandra! A cloud-native distribution of Apache Cassandra™ built for running on Kubernetes

LIVE hands-on workshop </>

Apache Cassandra™
meets Kubernetes!

March 3 or March 4 | Cloud-native | Beginner

LEVEL UP Developers

Cedrick Lunven Aujourd'hui à 16:18
Hey Community I will be live tonight for 50 min talk on Spring + Cassandra to the Virtual Java User Group

<https://www.youtube.com/watch?v=nuyPKDQn1gl> come and say hi ?

Envoyer un message à #main-chat-room

PRESENTER—3

- Aleks Volochnev
- David Jones-Gilardi
- jscarp

HELPER—1

- John Sanda

EN LIGNE—222

- Abhiprada
- Absurdism
- hiya
- Adalberto
- aditya_dhunna
- adnaneCord
- Adrigunz
- Aemilius Gaurav
- Aemilius gaurav
- Aguvas
- ajscilingo
- akashTaxvisor

DataStax Developers

Thank you!



Subscribe

