CASSANDRA SUMMIT 2016

Scalable data modelling by example

Carlos Alonso (@calonso)

Carlos Alonso

- Ex-Londoner
- MSc Salamanca University, Spain
- Software Engineer @ Jobandtalent
- Cassandra certified developer
- Datastax Cassandra MVP 2015 & 2016
- @calonso /

CASSANDRA SUMMIT 2016

Jobandtalent

- Revolutionising how people find jobs and how businesses hire employees.
- Leveraging data to produce a unique job matching technology.
- 10M+ users and 150K+ companies worldwide
- @jobandtalentEng / http://jobandtalent.com
- We are hiring!!

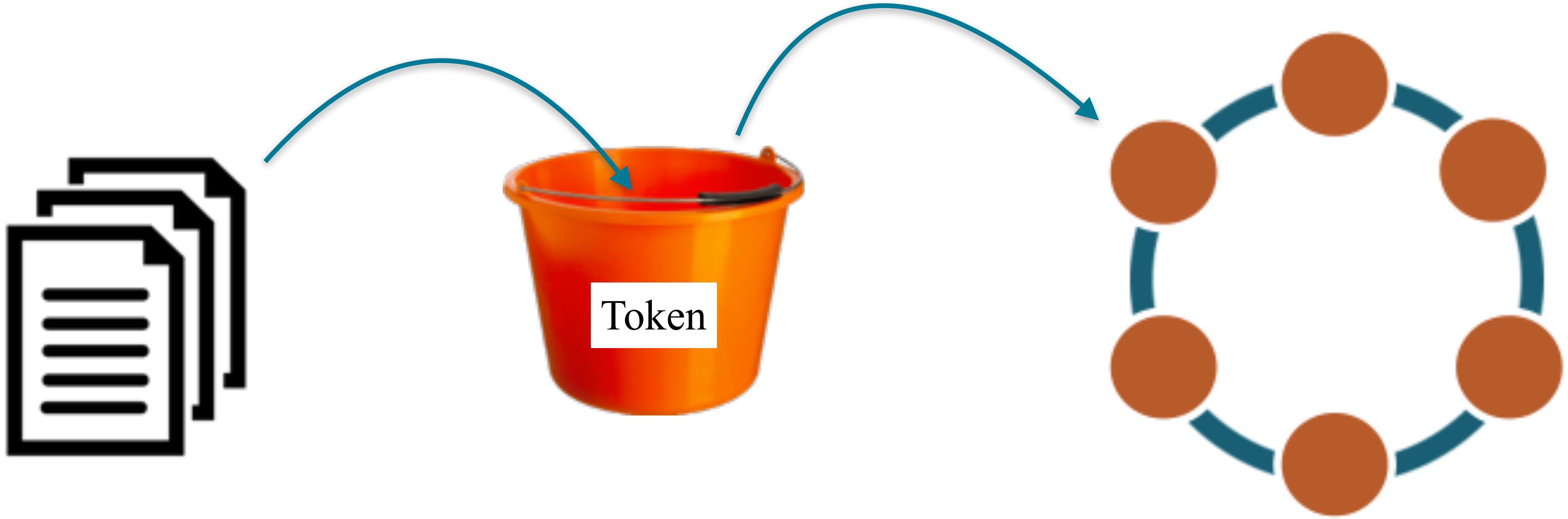
CASSANDRA SUMMIT 2016

Cassandra Concepts

CASSANDRA SUMMIT 2016

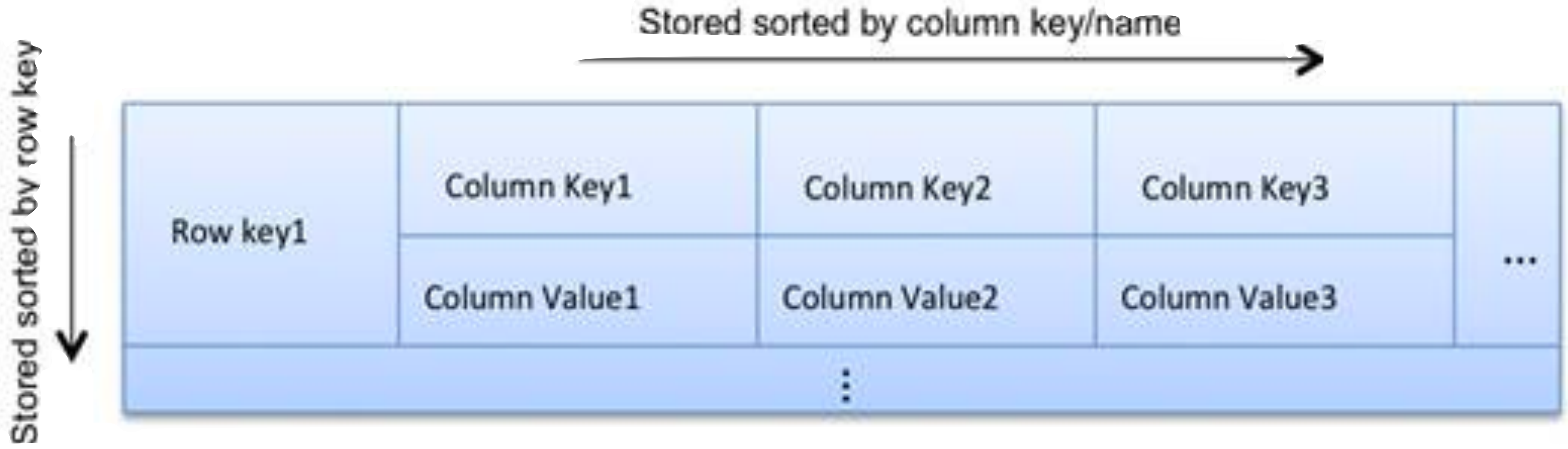
The data model is the only thing you can’t change once in production.

Data organisation



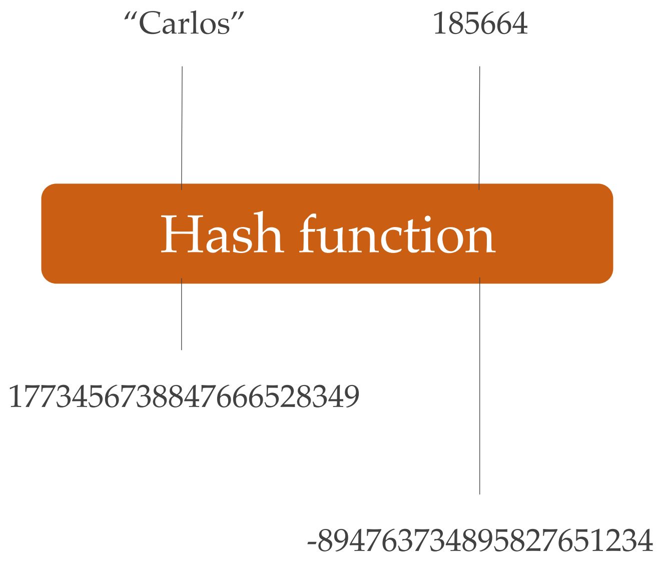
CASSANDRA SUMMIT 2016

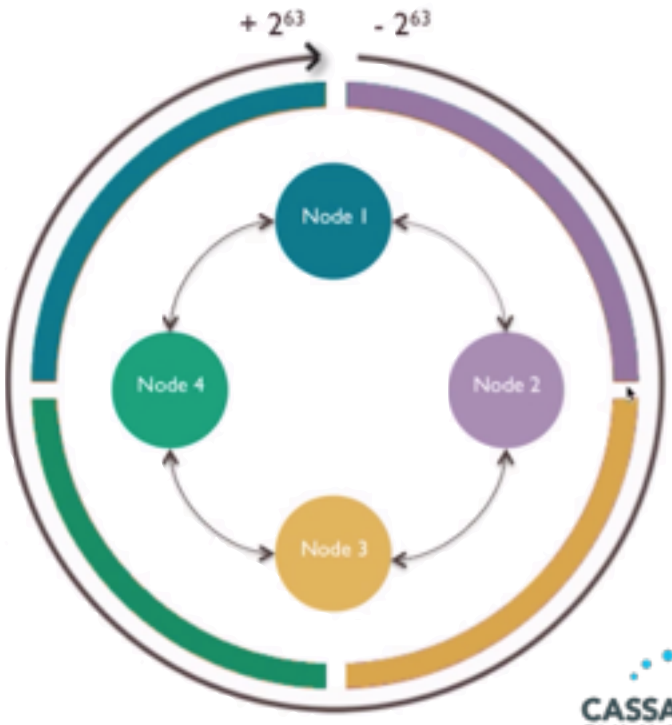
Physical Data Layout



CASSANDRA SUMMIT 2016

Consistent Hashing

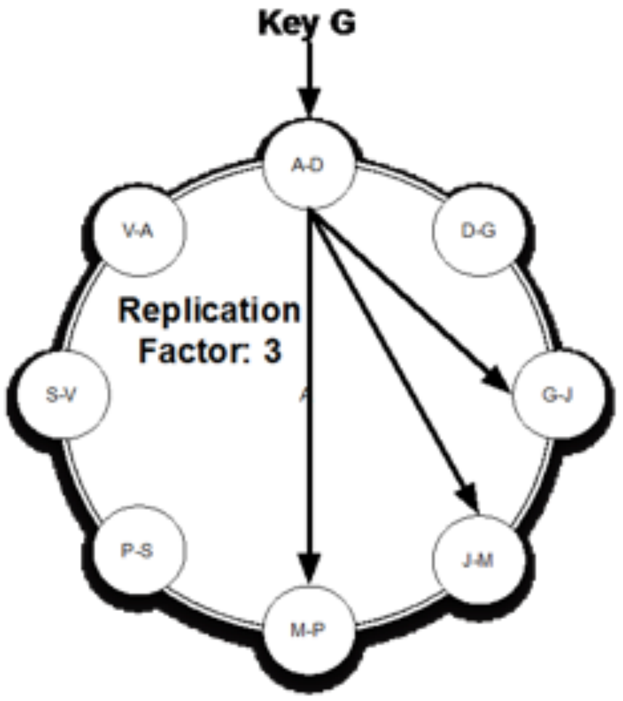




CASSANDRA SUMMIT 2016

Replication factor

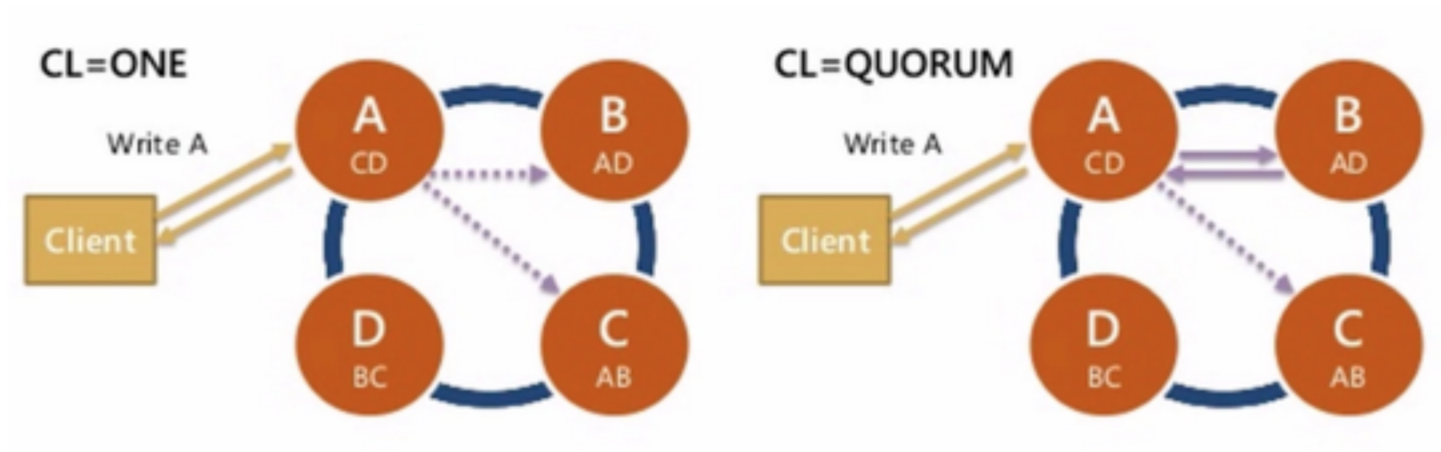
How many copies (replicas) for your data



CASSANDRA SUMMIT 2016

Consistency Level

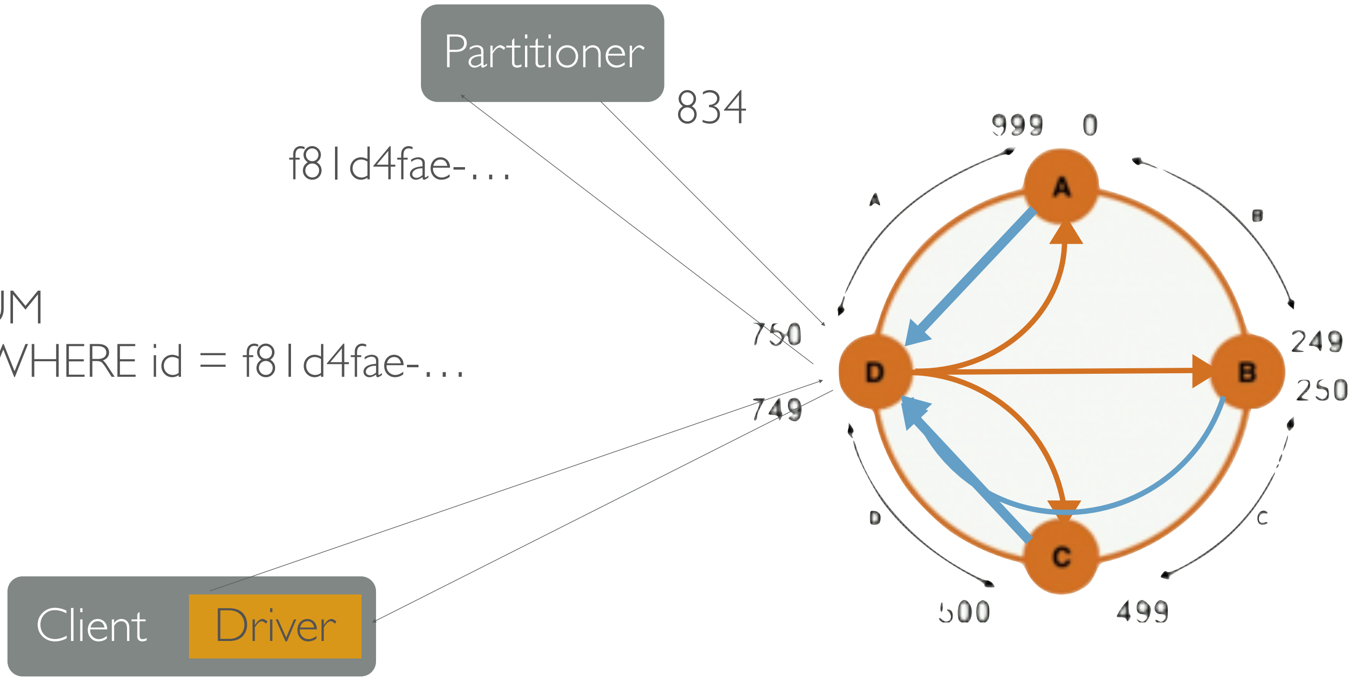
How many replicas of your data must acknowledge?



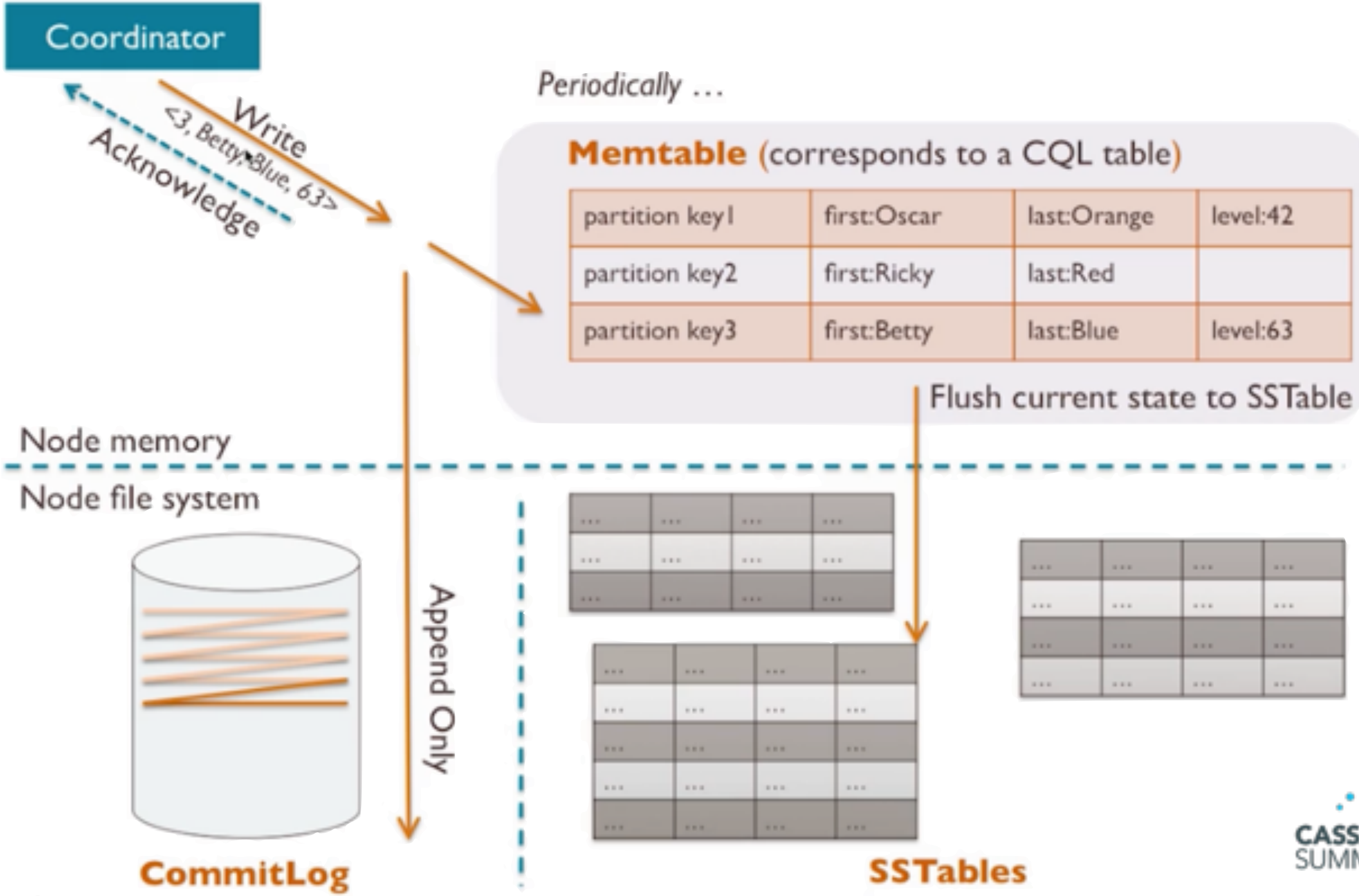
CASSANDRA SUMMIT 2016

A complete read/write example

- RF = 3
- CL = QUORUM
- SELECT * … WHERE id = f81d4fae-…



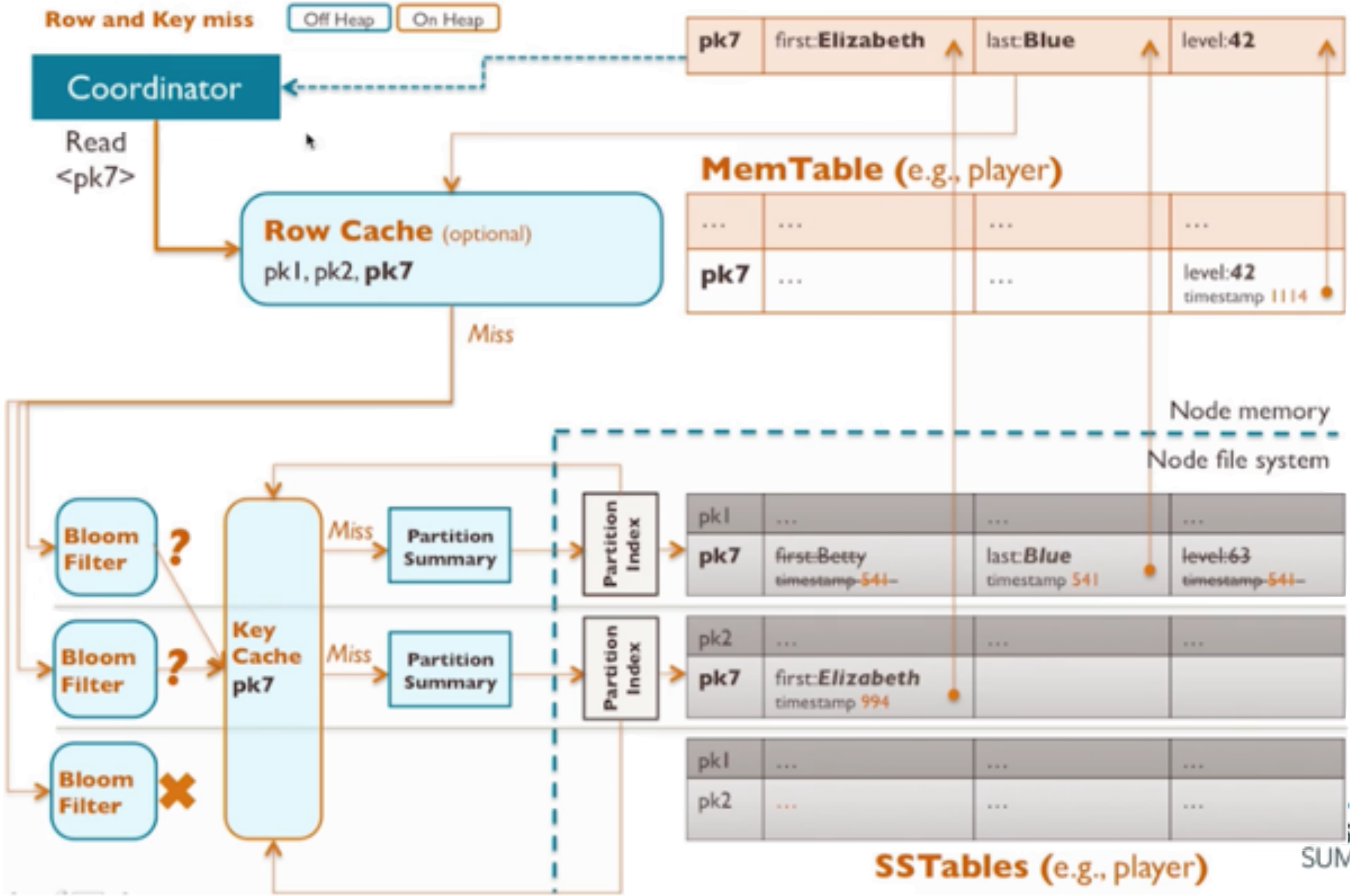
Cassandra Summit 2016



partition key1	first:Oscar	last:Orange	level:42
partition key2	first:Ricky	last:Red	
partition key3	first:Betty	last:Blue	level:63

CASSANDRA SUMMIT 2016

Row and Key miss



SSANDRA SUMMIT2016

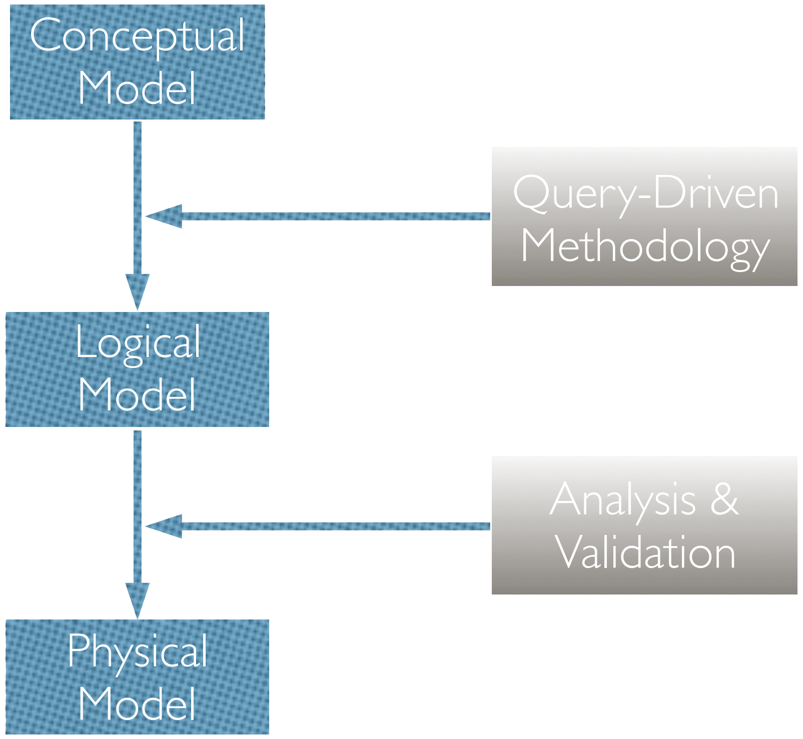
CASSANDRA SUMMIT 2016

Data Modelling

Data Modelling

- Understand your data
- Decide (know) how you'll query the data
- Define column families to satisfy those queries
- Implement and optimise

Data Modelling



Query Driven Methodology: goals

- Spread data evenly around the cluster
- Minimise the number of partitions read
- Keep partitions manageable

Query Driven Methodology: process

- Entities and relationships: map to tables
- Key attributes: map to primary key columns
- Equality search attributes: must be at the beginning of the primary key
- Inequality search attributes: become clustering columns
- Ordering attributes: become clustering columns

The Primary Key

PARTITION KEY + CLUSTERING COLUMN(S)

```
CREATE TABLE . . .(
  fields . . .
  PRIMARY KEY (part_key, clust1, . . .)
);
```

Analysis & Validation

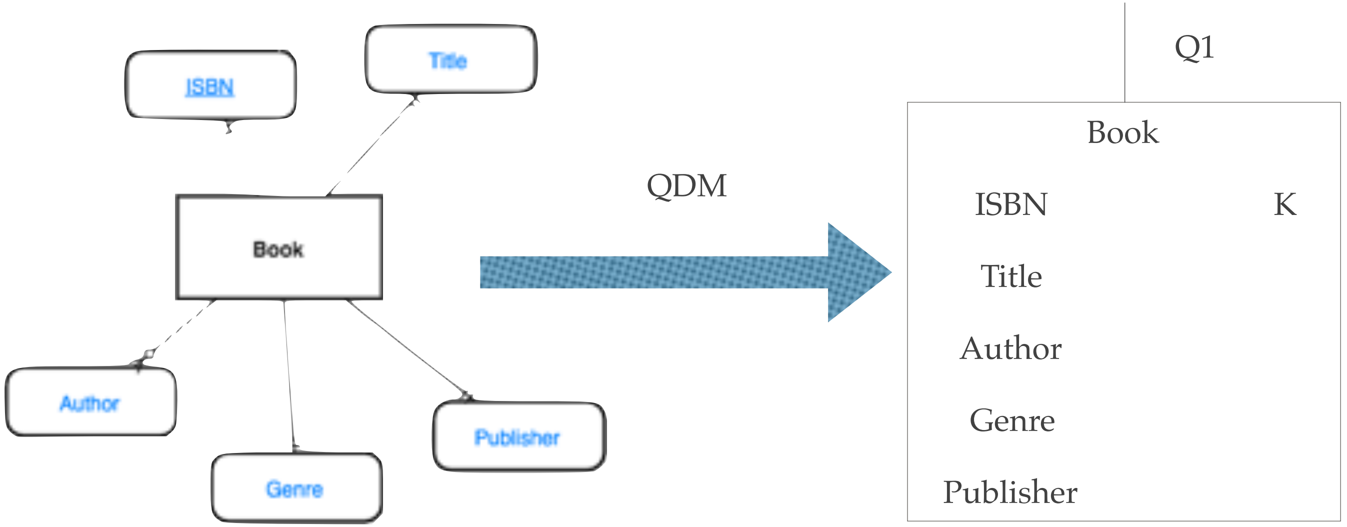
- Data evenly spread?
- 1 Partition per read?
- Are write conflicts (overwrites) possible?
- How large are partitions?
  - Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
- How much data duplication? (batches)

CASSANDRA SUMMIT 2016

An E-Library project.

Requirement: 1

Books can be uniquely identified and accessed by ISBN, we also need a title, genre, author and publisher.

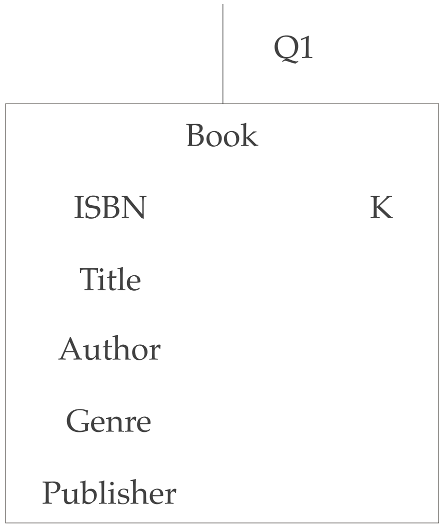


Q1: Find books by ISBN

Analysis & Validation

- Data evenly spread? ✓
- 1 Partition per read? ✓
- Are write conflicts (overwrites) possible? ✓
- How large are partitions? ✓
  - Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
  - 1 X (5 - 1 - 0) + 0 < 1M
- How much data duplication? 0 ✓

Q1: Find books by ISBN

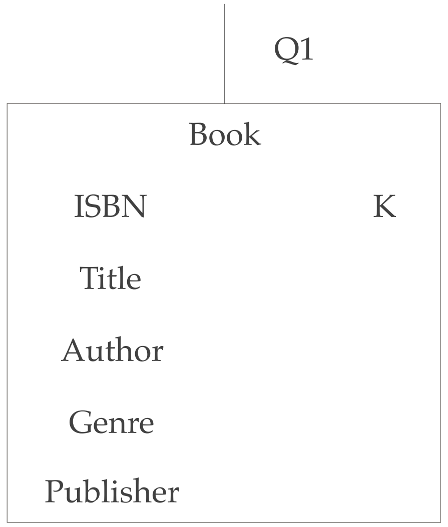


Physical data model

```
CREATE TABLE books (
  ISBN VARCHAR PRIMARY KEY,
  title VARCHAR,
  author VARCHAR,
  genre VARCHAR,
  publisher VARCHAR
);
```

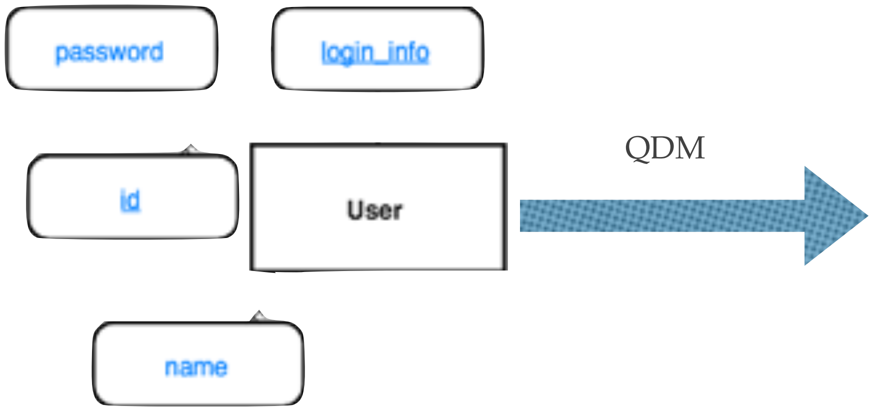
```
SELECT * FROM books WHERE ISBN = '…';
```

Q1: Find books by ISBN



Requirement 2

Users register into the system uniquely identified by an email and a password. We also want their full name. They will be accessed by email and password or internal unique ID.



Q1

Users_by_ID	
ID	K
full_name	

Q2 | Q3

Users_by_login_info	
email	K
password	C
full_name	
ID	

Q1: Find users by ID

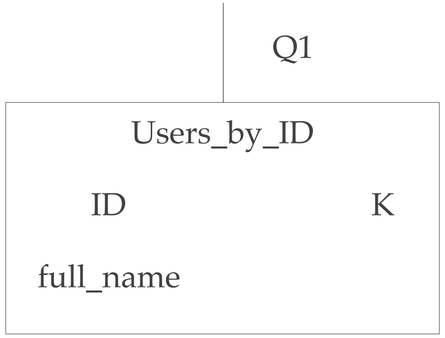
Q2: Find users by login info

Q3: Find users by email
(to guarantee uniqueness)

Analysis & Validation

- Data evenly spread? ✓
- 1 Partition per read? ✓
- Are write conflicts (overwrites) possible? ✓
- How large are partitions? ✓
  - Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
  - 1 X (2 - 1 - 0) + 0 < 1M
- How much data duplication? 0 ✓

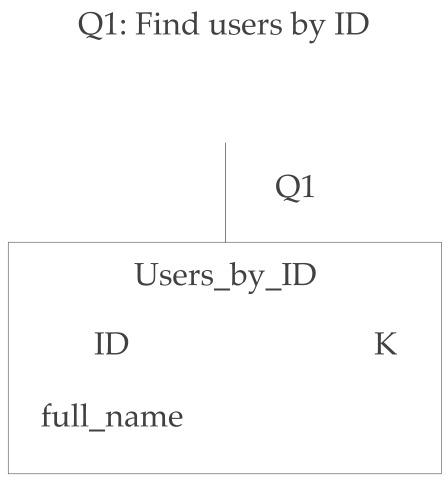
Q1: Find users by ID



Physical Data Model

```
CREATE TABLE users_by_id (
   ID TIMEUUID PRIMARY KEY,
   full_name VARCHAR
);
```

```
SELECT * FROM users_by_id WHERE ID = …;
```



Analysis & Validation

- Data evenly spread? ✓
- 1 Partition per read? ✓
- Are write conflicts (overwrites) possible? ✓
- How large are partitions? ✓
  - Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
  - 1 X (4 - 1 - 0) + 0 < 1M
- How much data duplication? 1 ✓

Q2: Find users by login info

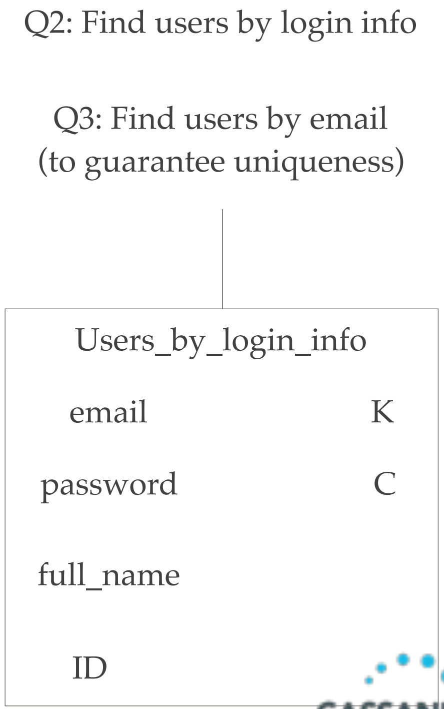
Q3: Find users by email
(to guarantee uniqueness)

Users_by_login_info	
email	K
password	C
full_name	
ID	

Physical Data Model

```
CREATE TABLE users_by_login_info (
   email VARCHAR,
   password VARCHAR,
   full_name VARCHAR,
   ID TIMEUUID,
   PRIMARY KEY (email, password)
);
```

```
SELECT * FROM users_by_login_info
WHERE email = '…' [AND password = '…'];
```



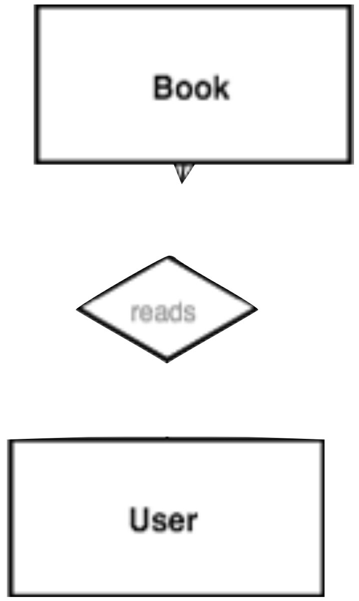
Physical Data Model

```
BEGIN BATCH
  INSERT INTO users_by_id (ID, full_name) VALUES (…) IF NOT EXISTS;
  INSERT INTO users_by_login_info (email, password, full_name, ID) VALUES (…);
APPLY BATCH;
```

Requirement 3

Users read books.
We want to know which books has a user read and show them sorted by title and author

Q1: Find all books a logged user has read



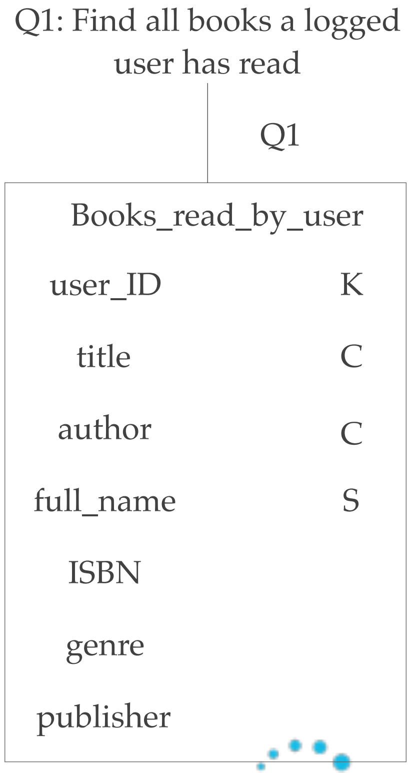
QDM

Q1

Books_read_by_user	
user_ID	K
title	C
author	C
full_name	S
ISBN	
genre	
publisher	

Analysis & Validation

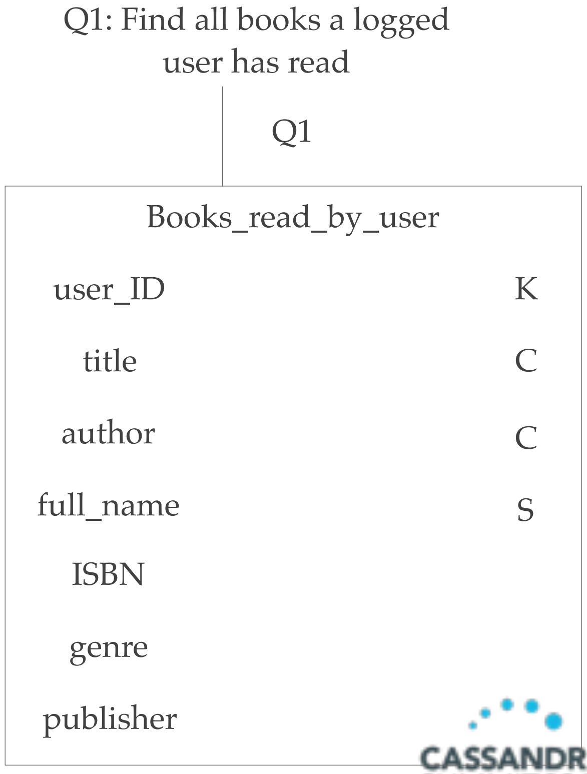
- Data evenly spread? ✓
- 1 Partition per read? ✓
- Are write conflicts (overwrites) possible? ✓
- How large are partitions? ✓
  - Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
  - Books X (7 - 1 - 1) + 1 < 1M => 200,000 books per user
- How much data duplication? 0 ✓



Physical Data Model

```
CREATE TABLE books_read_by_user (
  user_id TIMEUUID,
  title VARCHAR,
  author VARCHAR,
  full_name VARCHAR STATIC,
  ISBN VARCHAR,
  genre VARCHAR,
  publisher VARCHAR,
  PRIMARY KEY (user_id, title, author)
);
```

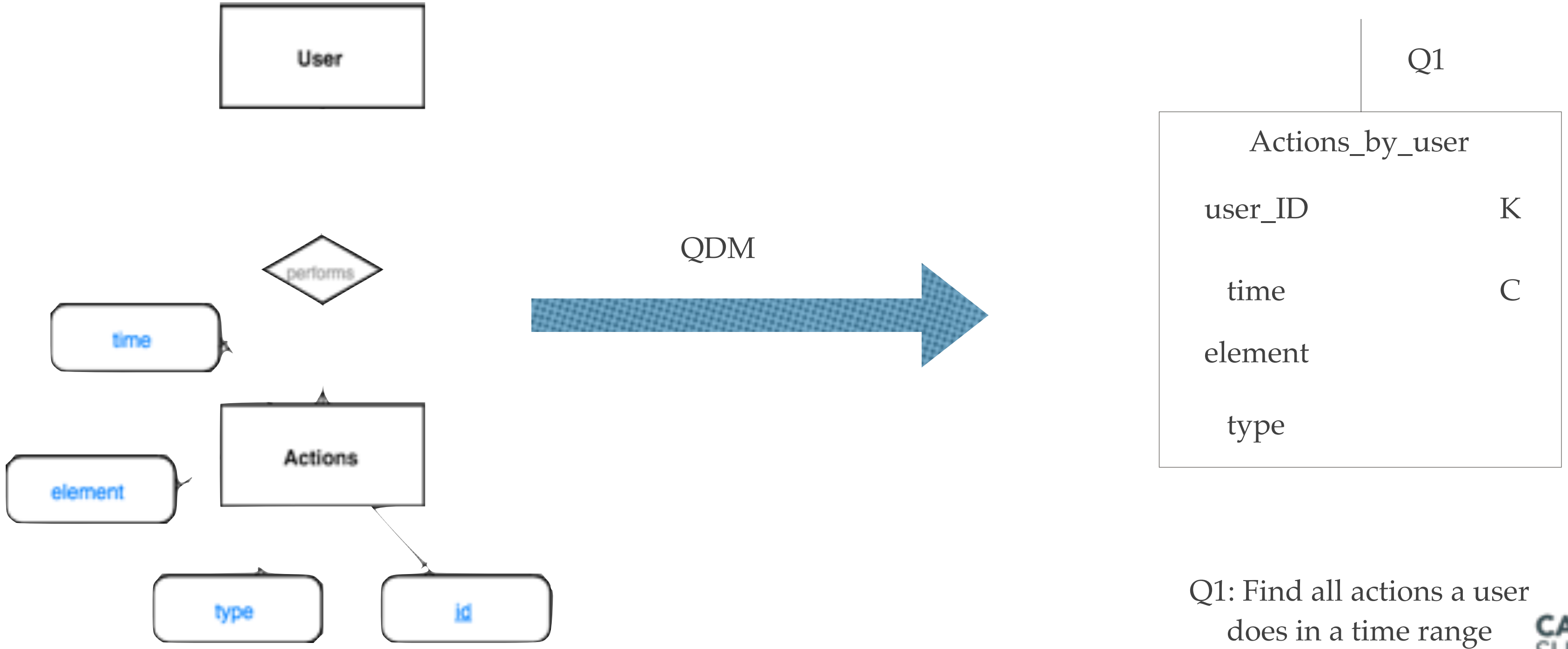
```
SELECT * FROM books_read_by_user
WHERE user_ID = …;
```



CASSANDRA SUMMIT 2016

Requirement 4

In order to improve our site's usability we need to understand how our users use it by tracking every interaction they have with our site.

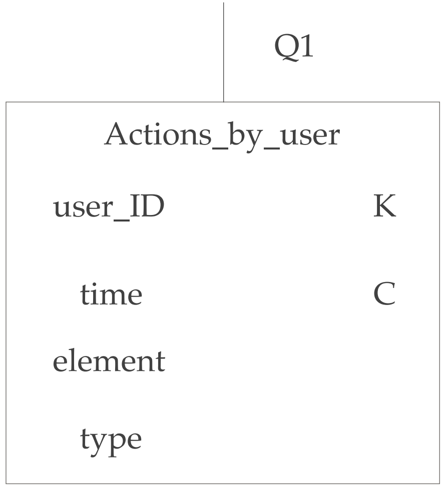


Q1: Find all actions a user does in a time range

Analysis & Validation

- Data evenly spread? ✓
- 1 Partition per read? ✓
- Are write conflicts (overwrites) possible? ✓
- How large are partitions? ✗
  - Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
  - Actions X (4 - 1 - 0) + 0 < 1M => 333.333
- How much data duplication? 0

Q1: Find all actions a user does in a time range



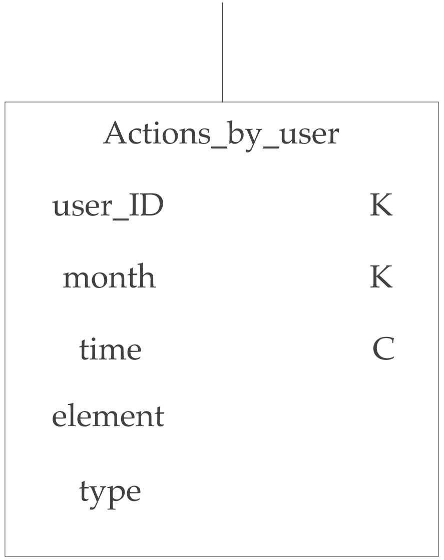
Requirement 4: Bucketing

- Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
- Actions X (5 - 2 - 0) + 0 < 1M => 333.333
  per user every <bucket_size>

bucket_size = 1 year => 38 actions / h

bucket_size = 1 month => 462 actions / h

bucket_size = 1 week => 1984 actions / h

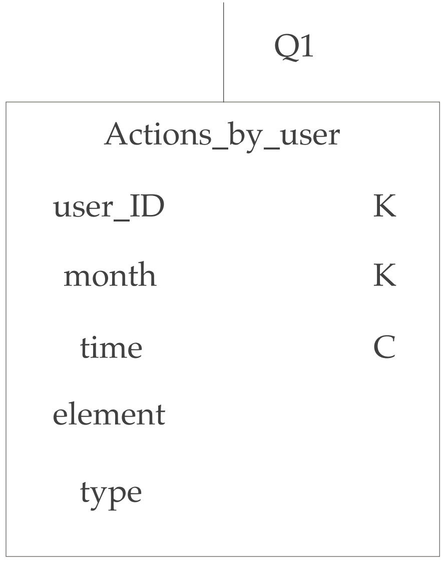


Actions_by_user	
user_ID	K
month	K
time	C
element	
type	

Analysis & Validation

- Data evenly spread? ✓
- 1 Partition per read? ✓
- Are write conflicts (overwrites) possible? ✓
- How large are partitions? ✓
  - Ncells = Nrow X (Ncols – Npk – Nstatic) + Nstatic < 1M
  - Actions X (5 - 2 - 0) + 0 < 1M => 333.333 / month
- How much data duplication? 0 ✓

Q1: Find all actions a user does in a time range

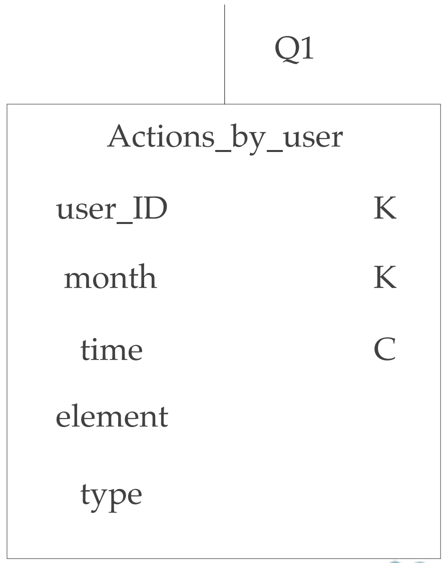


Physical Data Model

```
CREATE TABLE actions_by_user (
  user_ID TIMEUUID,
  month INT,
  time TIMESTAMP,
  element VARCHAR,
  type VARCHAR,
  PRIMARY KEY ((user_ID, month), time)
);
```

```
SELECT * FROM actions_by_user
WHERE user_ID = … AND month = … AND time < … AND time > …;
```

Q1: Find all actions a user does in a time range



Further validation

$$\sum sizeOf(pk) + \sum sizeOf(sc) + Nr \times \sum(sizeOf(rc) + \sum sizeOf(clc)) + 8 \times Nv < 200\ MB$$

- pk = Partition Key column
- sc = Static column
- Nr = Number of rows
- rc = Regular column
- clc = Clustering column
- Nv = Number of values

Next Steps

- Test your models against your hardware setup
  - cassandra-stress
  - http://www.sestevez.com/sestevez/CassandraDataModeler/ (kudos Sebastian Estevez)
- Monitor everything
  - DataStax OpsCenter
  - Graphite
  - Datadog
  - . . .

CASSANDRA SUMMIT 2016

Thanks!

Carlos Alonso
Software Engineer
@calonso