# COLUMN FAMILY DATABASE

## CASSANDRA

**Yunghans Irawan (yirawan@nus.edu.sg)**

# Agenda

- Column Family Database
- Cassandra
- Terminology
- Data Types
- Data Modeling
- Architecture
- Summary

# Column Family Database

- Store data in column families
  - Many column associated with a row key
- Column families are groups of related data that is often accessed together
- Arguably more similar to RDBMS compared to other types of NoSQL databases

# Column Family Databases

| Name | Initial Release | Latest Version | License |
|---|---|---|---|
| HBase | 2008 | 1.4.3 April 2018 | Open Source |
| Cassandra | 2008 | 3.11.2 Feb 2018 | Open Source |

# Cassandra

- Open source Apache Project
- Originated at Facebook in 2007 to solve its inbox search problem
  - Huge volumes of data
  - Many random reads
  - Many simultaneous random writes
- Paper on Cassandra is published in 2009
- Project is led and supported by DataStax
  - Provide enterprise versions, integrations with other technologies and product support

# Strength

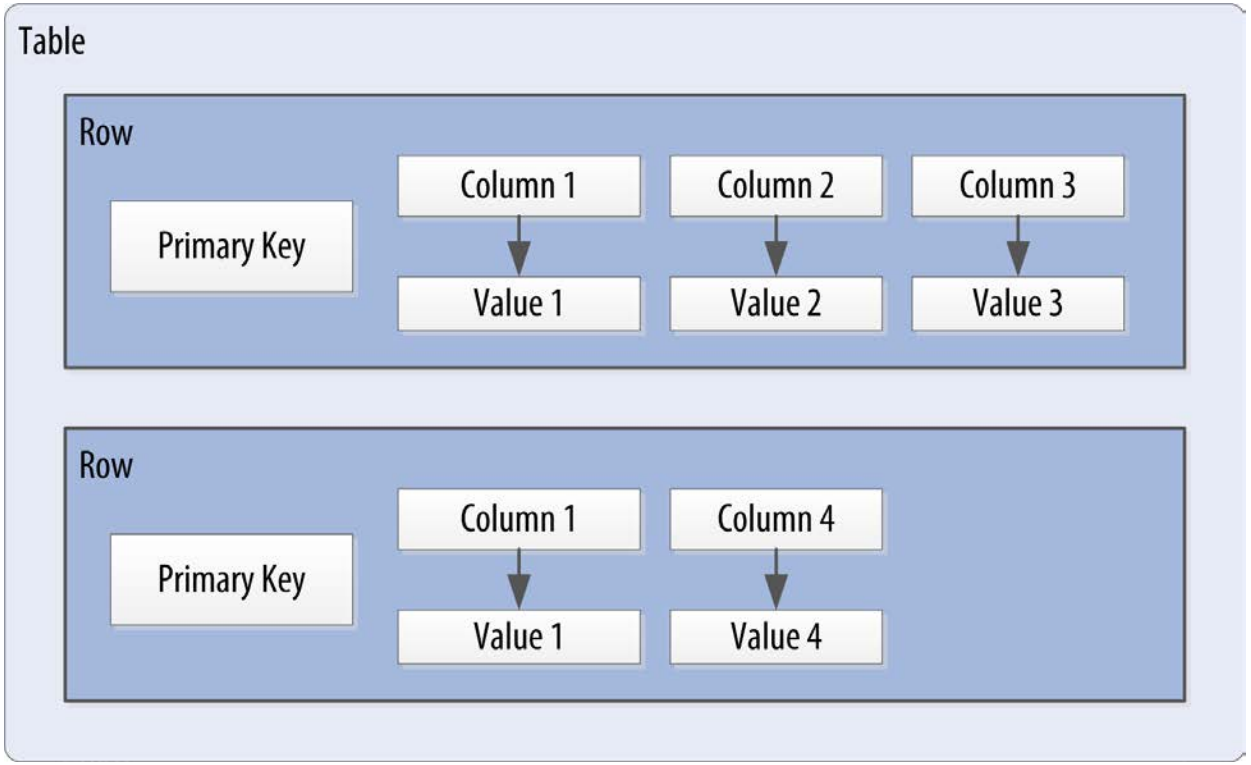
- Can support high rate of write
- Scalable write and read throughput, suitable for large deployment
- Similarity with RDBMS concepts with richer data types
- Many configurable parameters depending to customize the behavior based on needs
  - E.g. replication and consistency requirements
- Can support very large number of “columns”
  - 2 billions variable columns is the limit
  - Quote mark because the behavior is kind of different from traditional RDBMS column we are familiar with

# Weakness

- Weaker ACID transactions property
  - Starts with no transaction support
  - More scenarios are being supported as the product matures
- Requires more design effort
  - Similar to RDBMS but not quite the same
- No joins
- No referential integrity enforcement

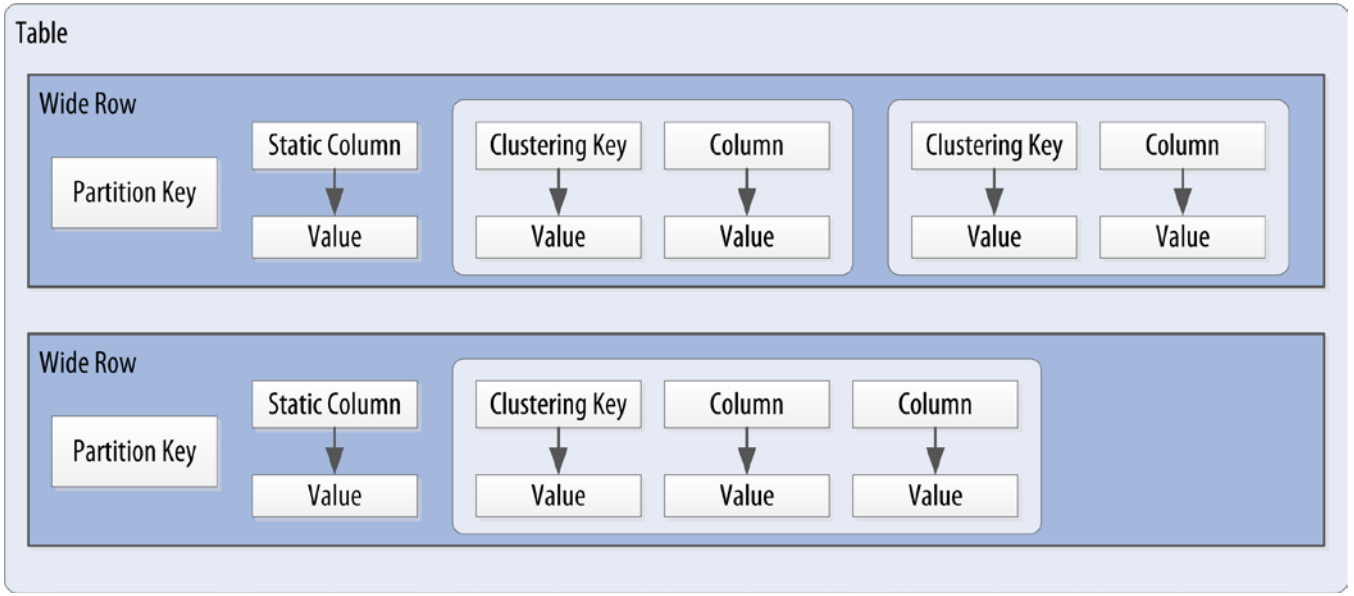
# Cassandra Table

- A logical collection of similar rows



# Compound Key

- Compound key is made of:
  - 1 partition key
  - 1 or more clustering key
- Partition key determines which node stores the data



# Static column

- Can be quite a complicated concept looking from RDBMS perspective
- Behave like ‘static’ keyword in C# or Java
  - If you’re from developer background
- It means that there’s only one of such cell for one partition key, regardless of the number of clustering key
  - Refer to the previous diagram
- Example – Attendance taking
  - Compound key: ClassID (P) + Date (C)+ Student ID (C)
  - Static column: Instructor
    - Assuming instructor depends only on the class and not on the date/student

# Cassandra Data Structures

- Column
  - Name and value pair
- Row
  - Container for columns referenced by a primary key
- Table
  - Container for rows
- Keyspace
  - Container for tables
- Cluster
  - Container for keyspaces that span one or more nodes

# Comparison of terms

| RDBMS | Cassandra |
|---|---|
| Cluster | Cluster |
| Database | Keyspace |
| Table | Table |
| Row | Row |
| Column | Column |

# Cassandra Data Types

| CQL Type | Constants | Description |
|---|---|---|
| ascii | strings | US-ASCII character string |
| bigint | integers | 64-bit signed long |
| blob | blobs | Arbitrary bytes (no validation), expressed as hexadecimal |
| boolean | booleans | true or false |
| counter | integers | Distributed counter value (64-bit long) |
| date | strings | Date string, such as 2015-05-03 |

# Cassandra Data Types

| CQL Type | Constants | Description |
|---|---|---|
| decimal | integers, floats | Variable-precision decimalJava type **Note:** When dealing with currency, it is a best practice to have a currency class that serializes to and from an int or use the Decimal form. |
| double | integers, floats | 64-bit IEEE-754 floating pointJava type |
| float | integers, floats | 32-bit IEEE-754 floating pointJava type |

# Cassandra Data Types

| CQL Type | Constants | Description |
|---|---|---|
| frozen | user-defined types, collections, tuples | A frozen value serializes multiple components into a single value. Non-frozen types allow updates to individual fields. Cassandra treats the value of a frozen type as a blob. The entire value must be overwritten.**Note:** Cassandra no longer requires the use of **frozen** for **tuples**:frozen <tuple <**int**, tuple<text, **double**>>> |
| inet | strings | IP address string in IPv4 or IPv6 format, used by the python-cql driver and CQL native protocols |
| int | integers | 32-bit signed integer |
| list | n/a | A collection of one or more ordered elements: [ literal, literal, literal ] |

# Cassandra Data Types

| CQL Type | Constants | Description |
|---|---|---|
| list | n/a | A collection of one or more ordered elements: [ literal, literal, literal ] |
| map | n/a | A JSON-style array of literals: { literal : literal, literal : literal ... } |
| set | n/a | A collection of one or more elements: { literal, literal, literal } |
| smallint | integers | 2 byte integer |
| text | strings | UTF-8 encoded string |
| time | strings | Time string, such as 13:30:54.234 |
| timestamp | integers, strings | Date plus time, encoded as 8 bytes since epoch |
| timeuuid | uuids | Version 1 UUID only |

# Cassandra Data Types

| CQL Type | Constants | Description |
|---|---|---|
| tinyint | integers | 1 byte integer |
| tuple | n/a | Cassandra 2.1 and later. A group of 2-3 fields. |
| uuid | uuids | A UUID in standard UUID format |
| varchar | strings | UTF-8 encoded string |
| varint | integers | Arbitrary-precision integerJava type |

# Cassandra compared to RDBMS

- Denormalization is normal
  - Logical side effect of the first two characteristic
- Cannot use column that is not indexed as part of the query criteria
- Query-first design

# Secondary Index

- Allow query to use the column as criteria
- Can be applied to simple type column as well as collection columns

```
> SELECT * FROM user WHERE last_name = 'Nguyen';
InvalidRequest: code=2200 [Invalid query] message="No supported
  secondary index found for the non primary key columns restrictions“

> CREATE INDEX ON user ( last_name );

> SELECT * FROM user WHERE last_name = 'Nguyen';
 first_name | last_name
------------+-----------
       Bill |    Nguyen
(1 rows)

> CREATE INDEX ON user ( emails );
> CREATE INDEX ON user ( phone_numbers );
```

# Data Modeling for Cassandra

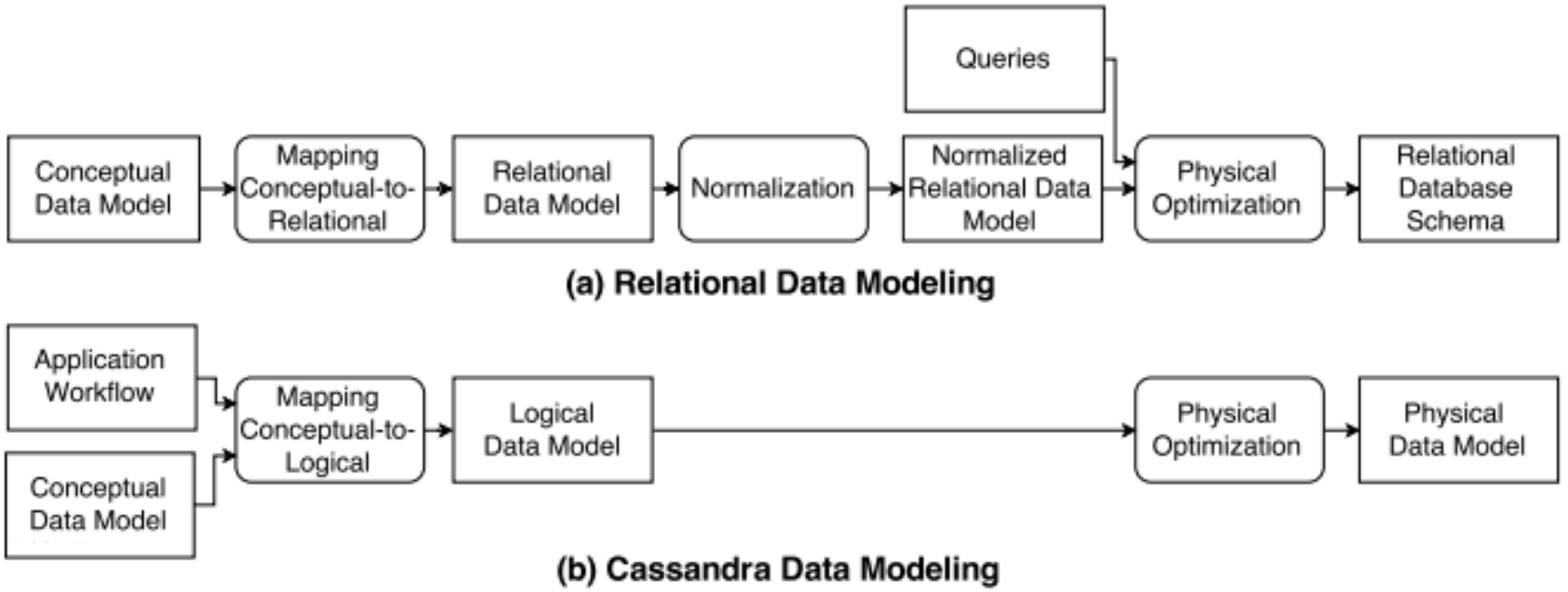
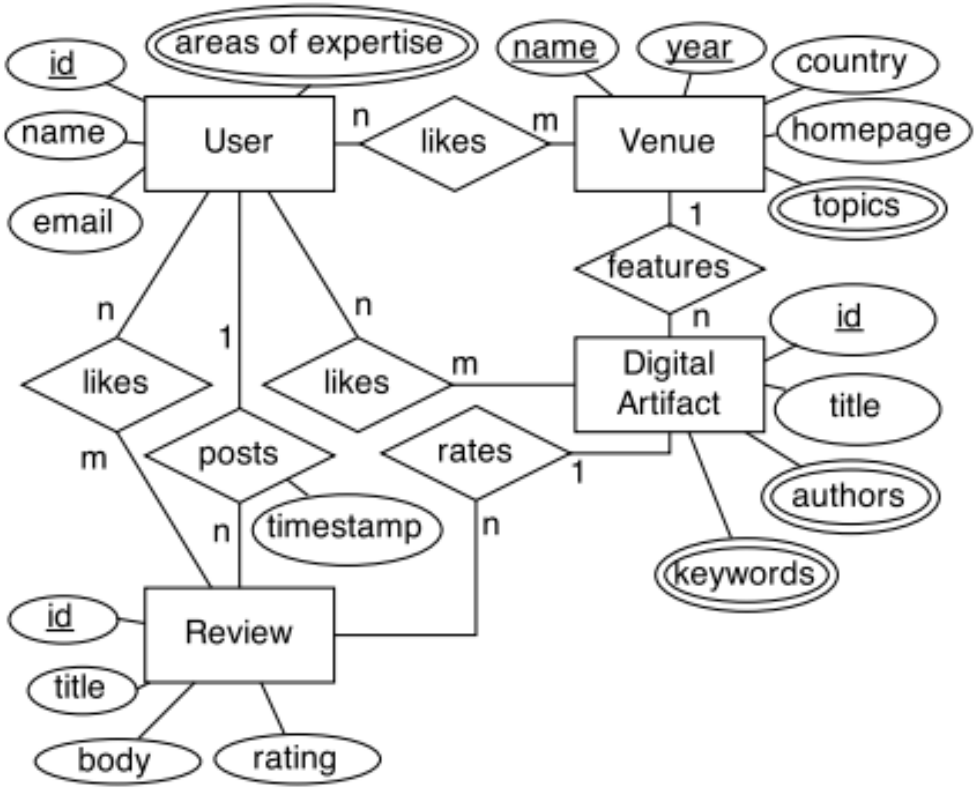


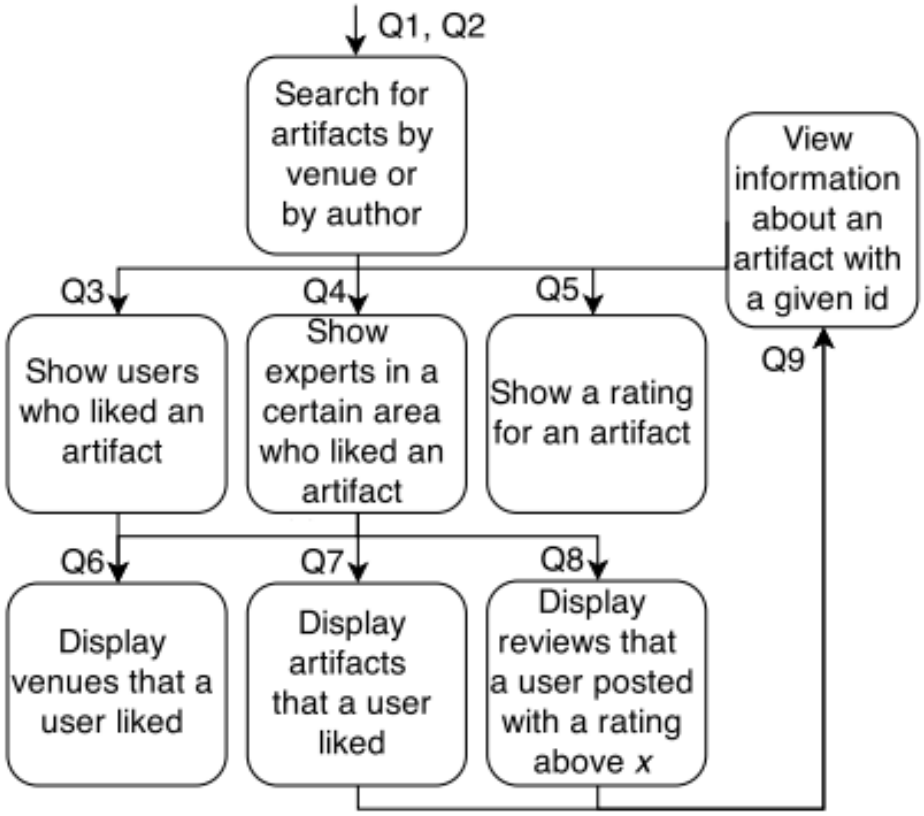

Fig. 1: Traditional data modeling compared with our proposed methodology for Cassandra.

# Conceptual Data Model




**(a) Conceptual data model**

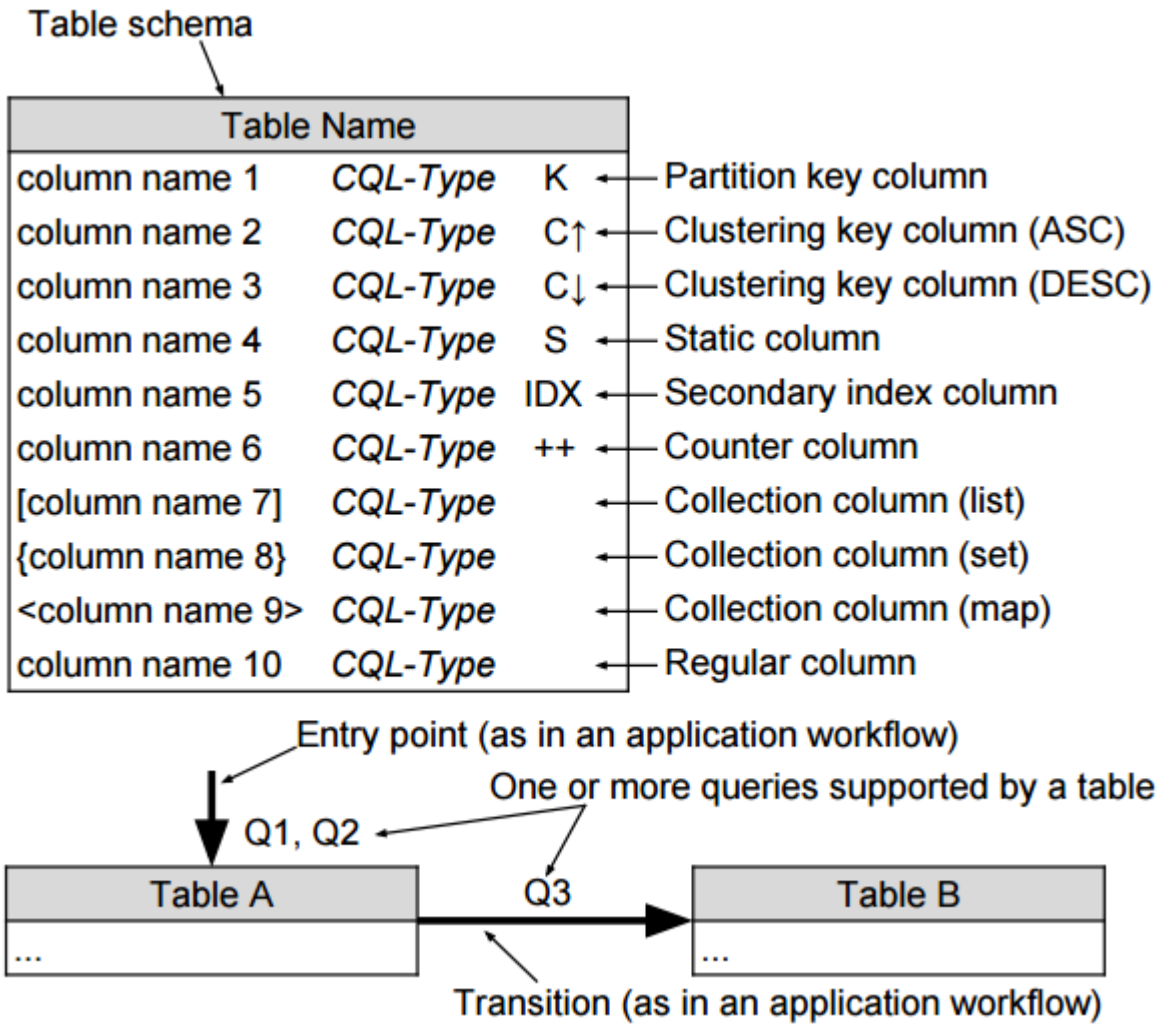
# Application Workflow



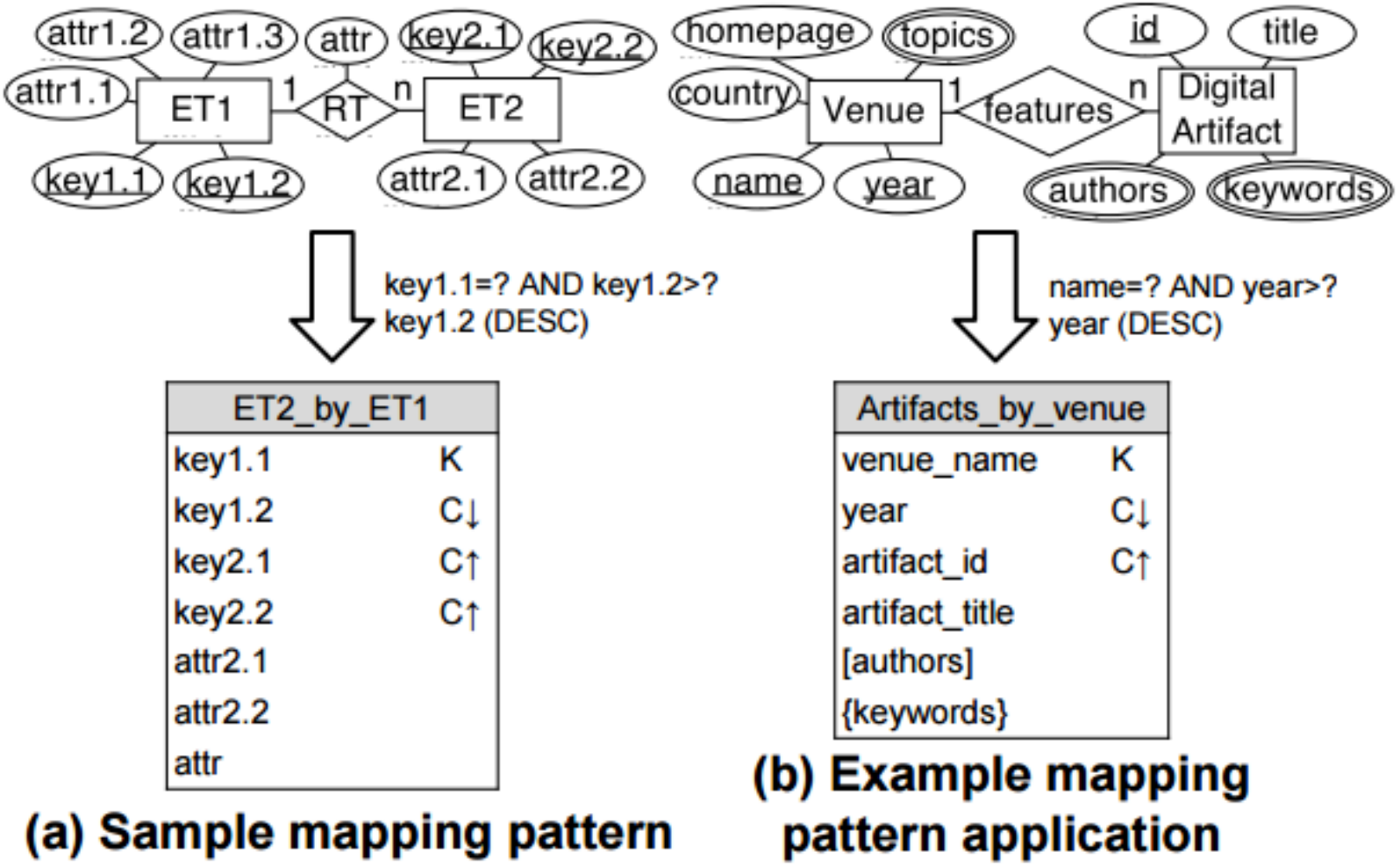

(b) Application workflow

Q1: Find artifacts published in a venue with a given name after a given year. Order results by year (DESC).
Q2: Find artifacts published by a given author. Order results by year (DESC).
Q3: Find users who liked a given artifact.
Q4: Find users who liked a given artifact and who have expertise in a certain area.
Q5: Find an average rating of a given artifact.
Q6: Find venues that a given user liked.
Q7: Find artifacts published after a certain year that a given user liked. Order results by year (DESC).
Q8: Find reviews posted by a given user with a rating >=*x*. Order results by rating (DESC).
Q9: Find information about an artifact with a given id.

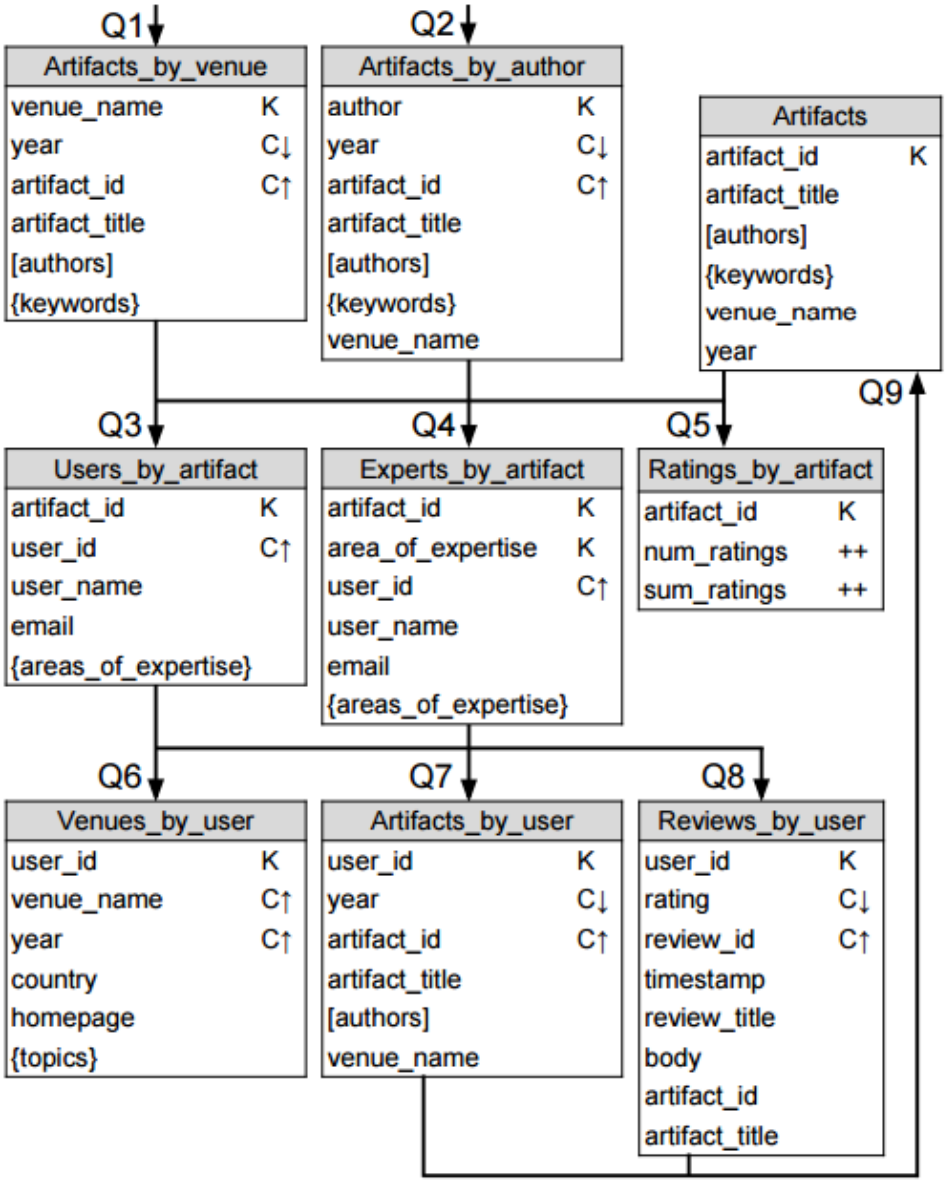
# Chebotko Diagram Notation



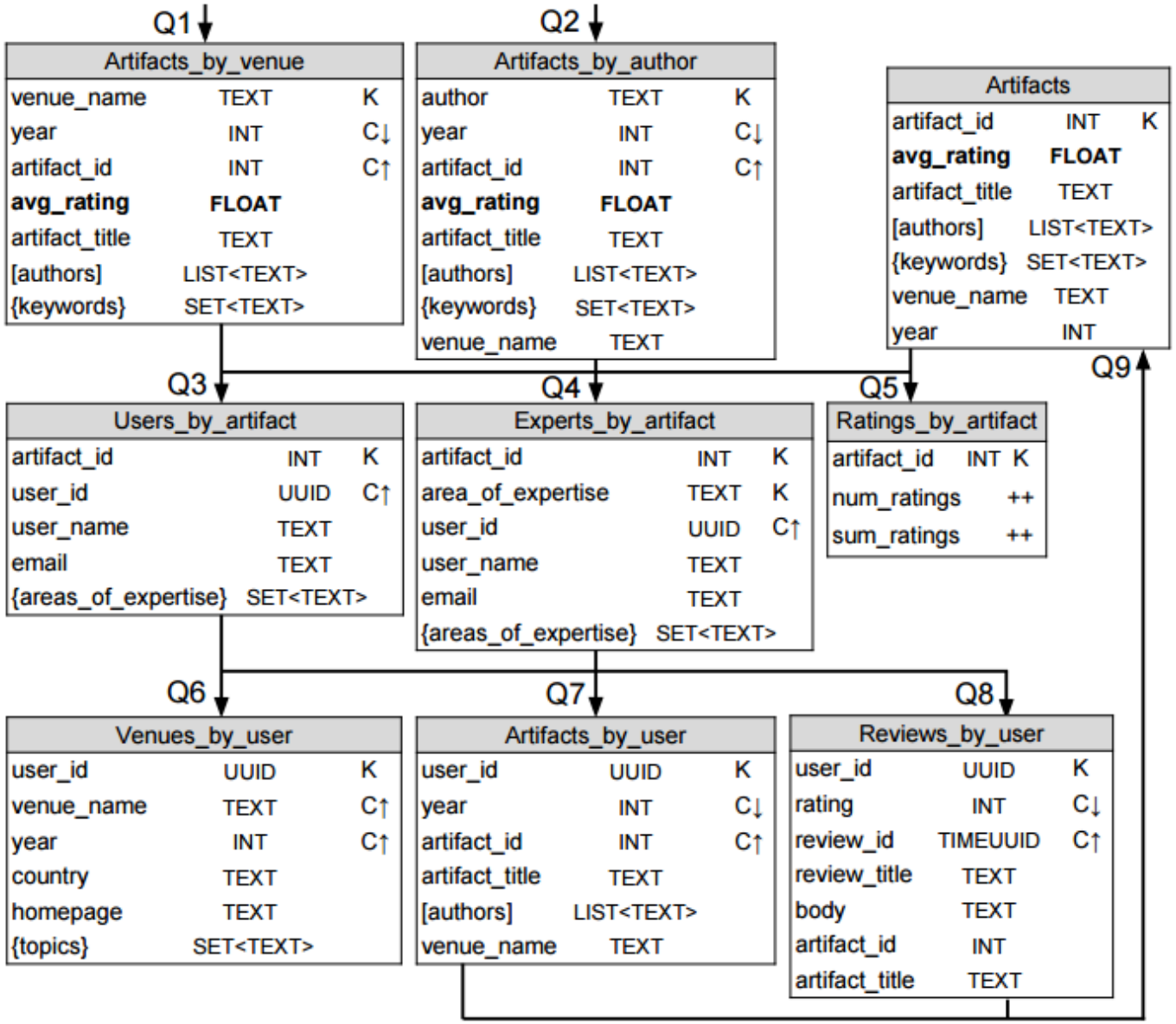
# Mapping into Logical



# Logical Diagram

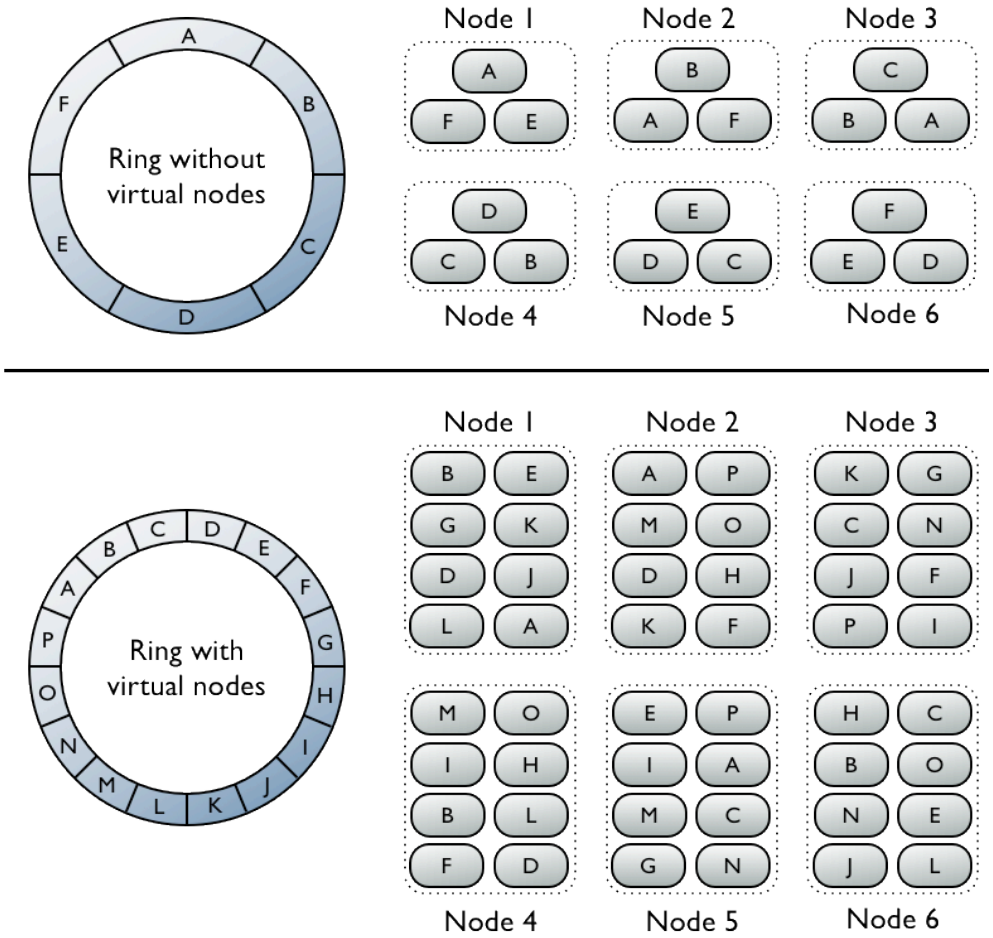


# Physical Diagram



# Cassandra Architecture

- Ring based replication
- No single point of failure
- Key hashes to a location in the ring
- Virtual nodes
  - Allow non-contiguous assignment of ranges
  - Data is split into more ranges
- Cluster rebalancing is automatically done when adding or removing nodes

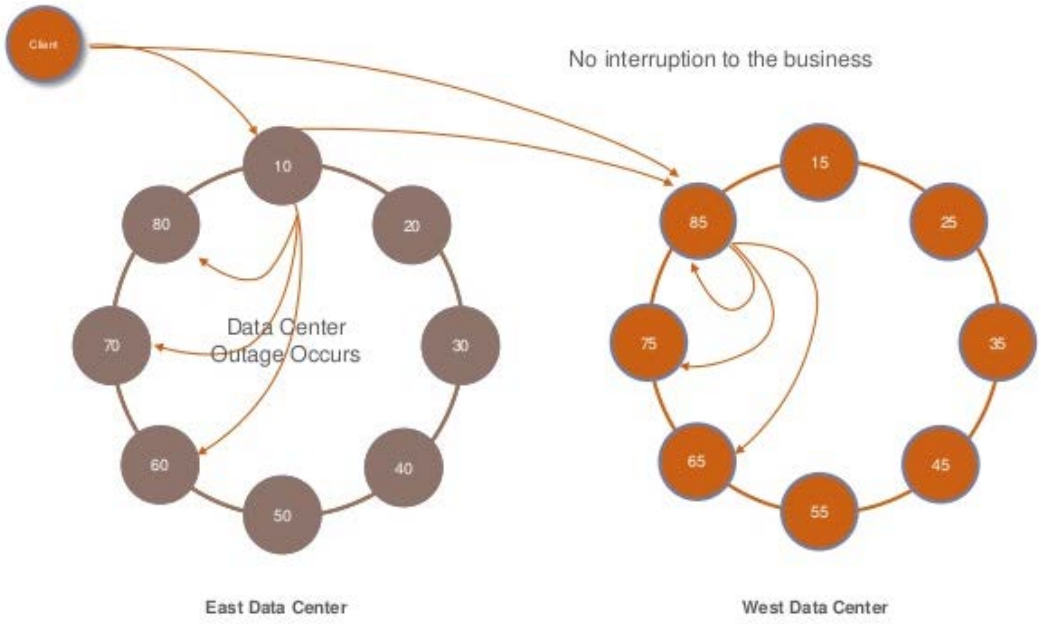


# Consistency Level

- Consistency level for read and write is tunable
- Write consistency levels
  - ALL, EACH_QUORUM, QUORUM, LOCAL_QUORUM, ONE, TWO, THREE, LOCAL_ONE, ANY
- Read consistency levels
  - ALL, EACH_QUORUM, QUORUM, LOCAL_QUORUM, ONE, TWO, THREE, LOCAL_ONE, SERIAL, LOCAL_SERIAL
- https://docs.datastax.com/en/cassandra/3.0/cassandra/dml/dmlConfigConsistency.html

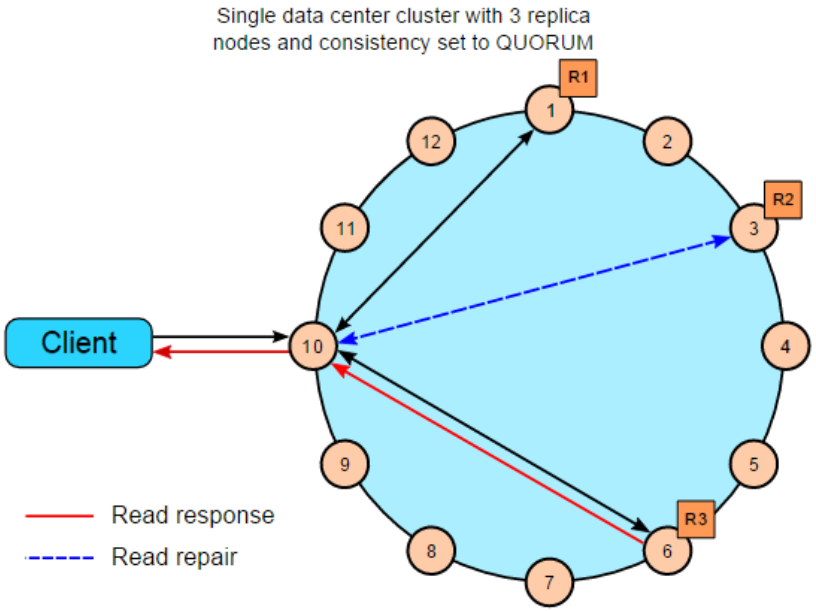
# Data Center Aware Replication

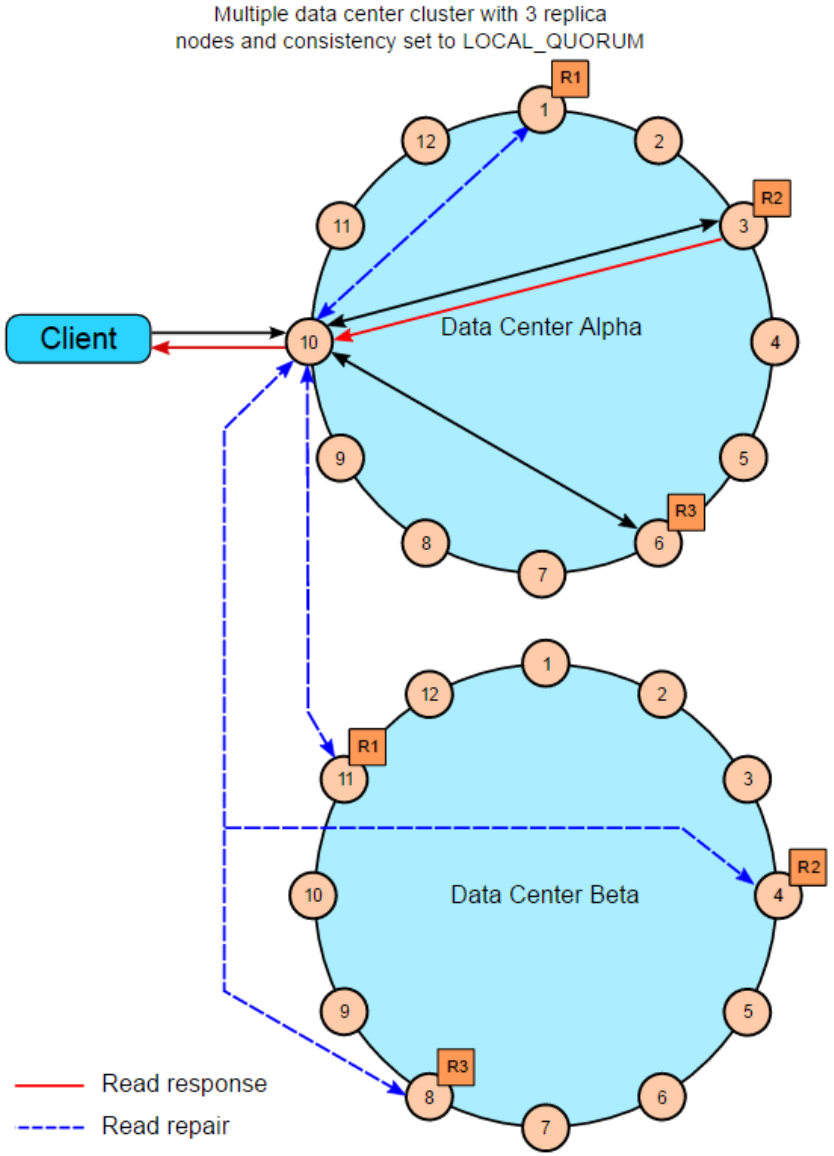
- Instances can be tagged with the datacenter ID to create multi data center deployment
- Consistency levels are data center aware




Source: http://image.slidesharecdn.com/nycjavameetupdsepresentation-140701094225-phpapp01/95/datastax-nyc-java-meetup-cassandra-with-java-8-638.jpg?cb=1404207863

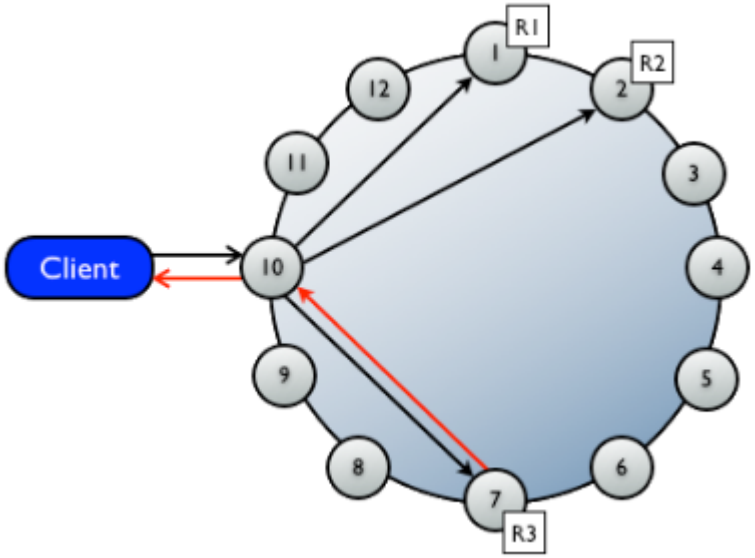
# Example: Read with QUORUM

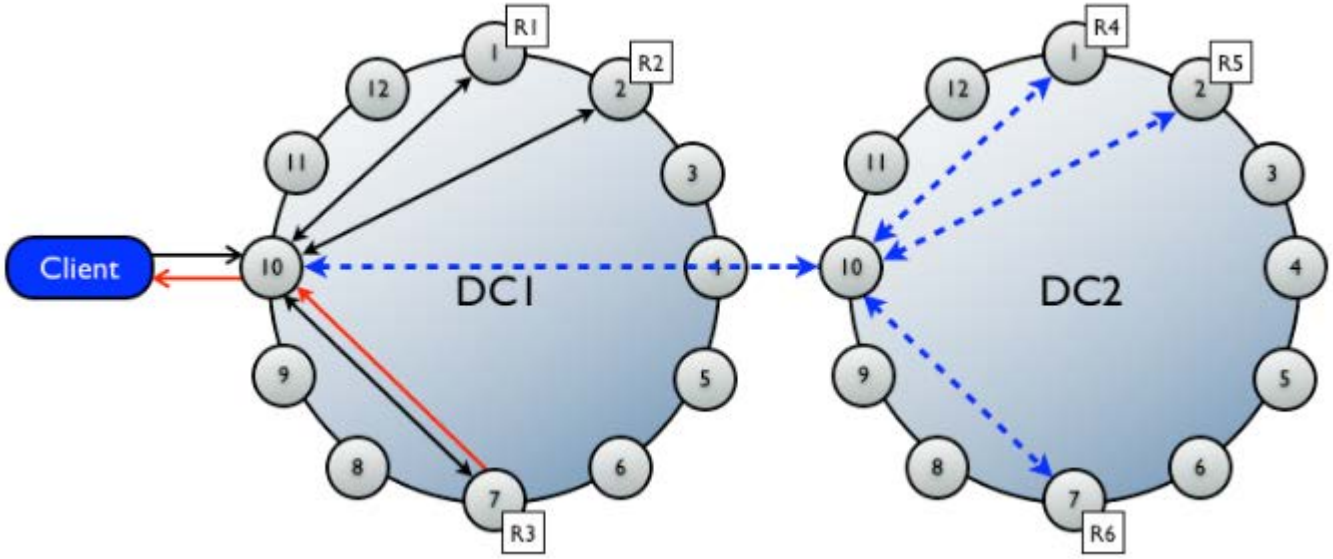



Source: https://teddyma.gitbooks.io/learncassandra/content/client/read_requests.html

# Example: Write



# Summary

- Column-family database offers performance and scalability beyond what traditional RDBMS offers
- The trade-off is extra complexity at design and application development
- Sophisticated replication technique
  - Data center aware
  - Seamless partition rebalancing

# References

- Cassandra Documentation
  - http://docs.datastax.com/
- Original Cassandra Paper
  - http://www.cs.cornell.edu/projects/ladis2009/papers/lakshman-ladis2009.pdf
- Cassandra Reference Card
  - http://www.datastax.com/wp-content/uploads/2013/03/cql_3_ref_card.pdf
- Cassandra: The Definitive Guide, 2nd Edition
  - Jeff Carpenter, Eben Hewitt, O'Reilly Media, 2016
- A Big Data Modeling Methodology for Apache Cassandra
  - http://www.cs.wayne.edu/andrey/papers/TR-BIGDATA-05-2015-CKL.pdf