cassandra

Sunbeam Infotech

Features

- Peer to peer architecture (no master-slave → no single point of failure).
- Linear scale performance Capacity ∝ nodes
- High performance → high speed read/write
- Simplified deployment and maintenance → on linux
- Less expensive → horizontal scalability
- Supports multiple programming languages → Java, Python, ... & REST services
- Operational and Development simplicity
- Cloud availability → avail on all leading cloud vendors AWS, Azure, GCP, ...
- Ability to deploy across data-centers
- Fault tolerance
- Configurable consistency (tight or eventual) → tradeoff with speed
- Flexible data model → columnar storage. Number of columns can be added dynamically (super column)
- Column family store
- Schema-free

Limitations

- Aggregation operations not supported
- Range queries on partition key are not supported
- Not good for too many joins (Denormalized data preferred)
- Not suitable for transactional data - "C" poor from ACID → eventual consistency
- During compaction performance/throughput slow down
- Not designed for update-delete (is possible)

Performance

- Performance measures
  - Throughput (Operations per second) → number of reads/writes per second (high)
  - Latency → time required for single operation (low)

- Cassandra vs MySQL
  - MySQL (more than 50 GB data)
    - Write speed: 300 ms
    - Read speed: 350 ms
  - Cassandra (more than 50 GB data)
    - Write speed: 0.12 ms
    - Read speed: 15 ms

Applications

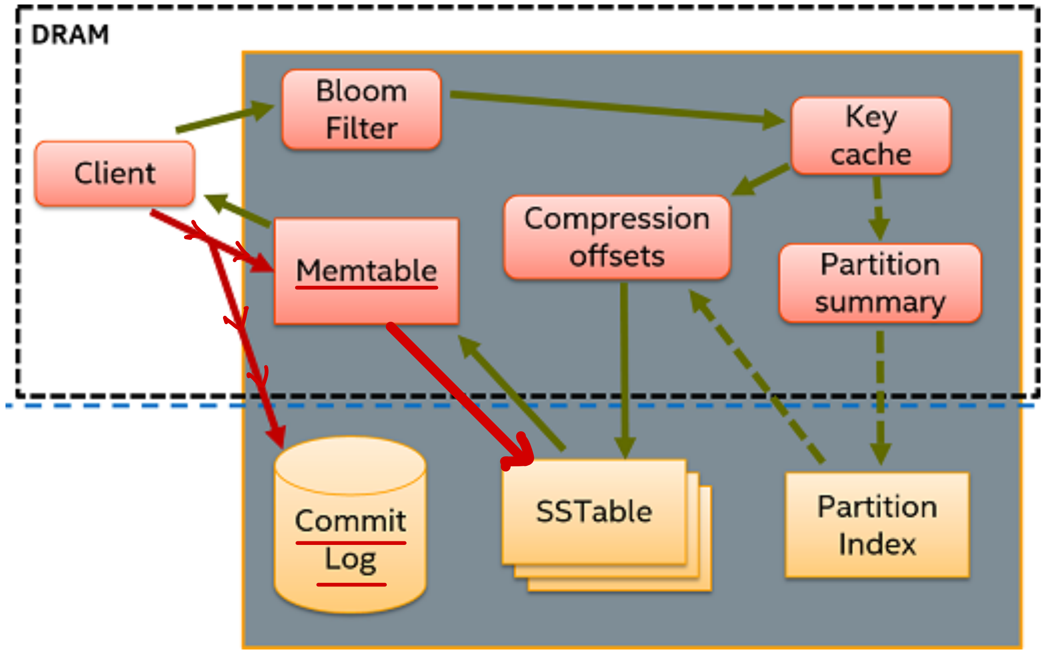
- Applications
  - Product catalog/Playlist
  - Recommendation/Personalization engine
  - Sensor/IoT data
  - Messaging/Time-series data
  - Fraud detection
- Customers
  - Facebook, Netflix, eBay, Apple, Walmart, GoDaddy
- Application requirements
  - Store and handle time-series data
  - Store and handle large volume of data
  - Scale predictably (linear Scaling)
  - High availability

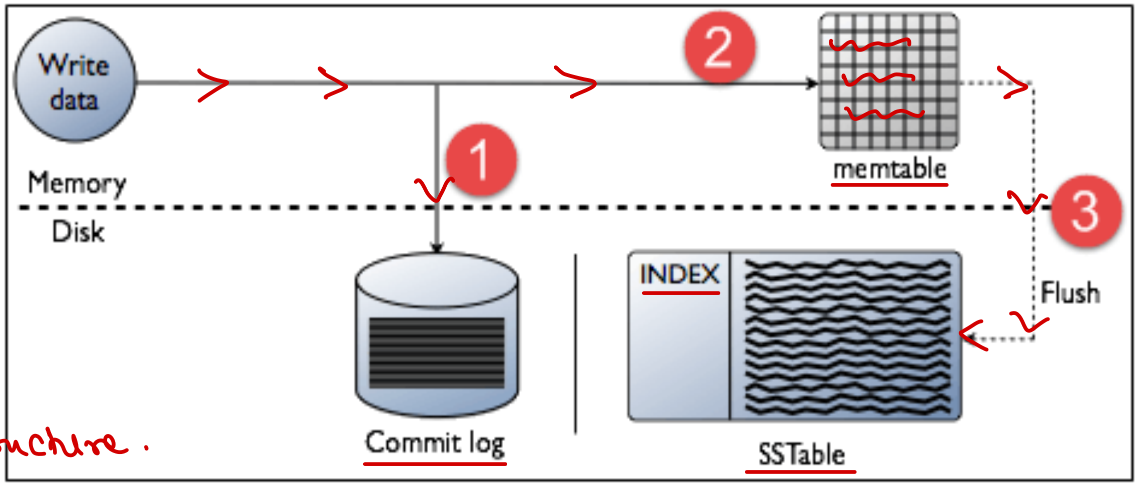
uuid - universally unique identifier - 128 bit

timeuuid - uuid + timestamp (helpful in sorting on time field)

Architecture

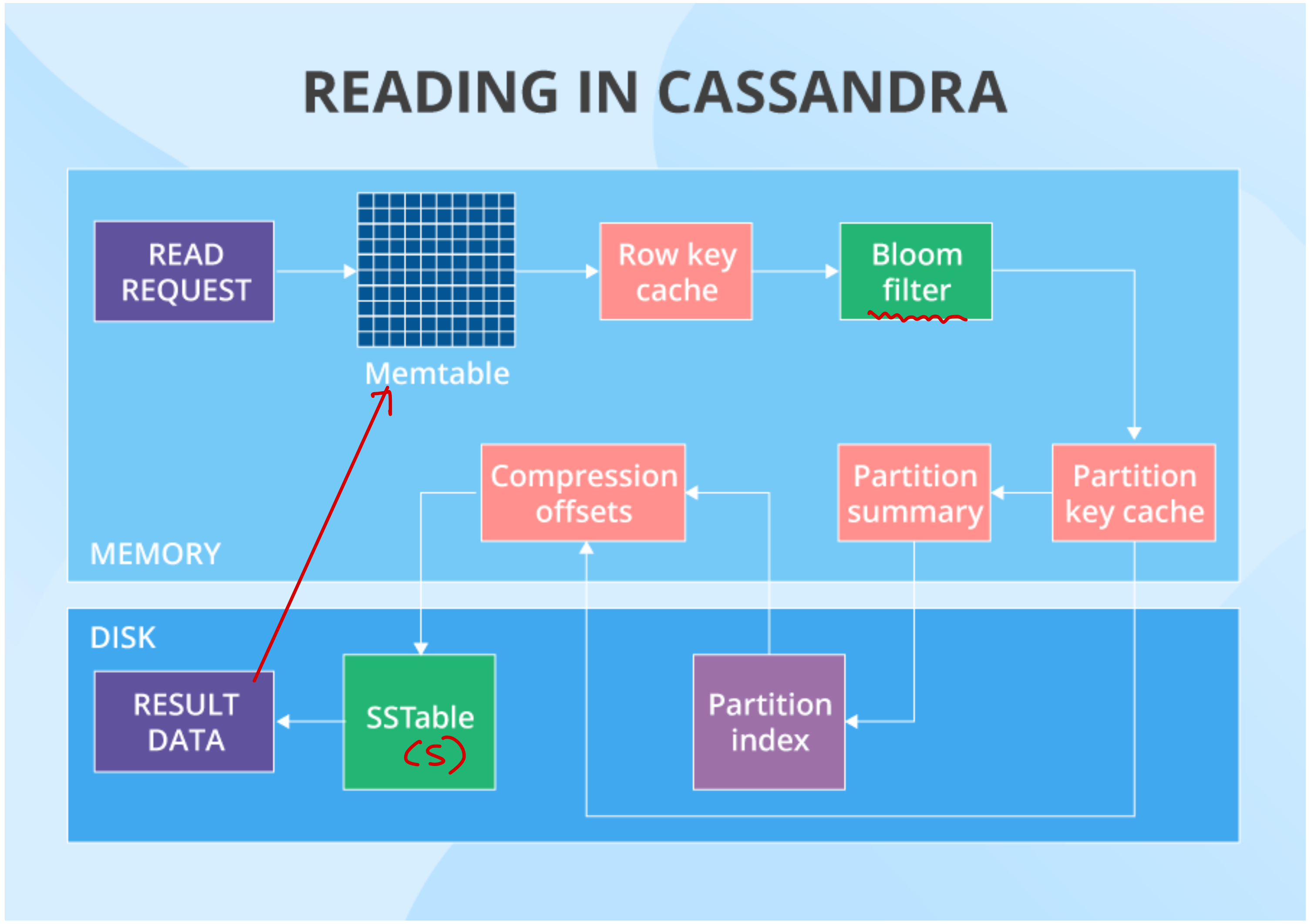
- Commit Log
  - Append only log of all mutations local to a node.
  - Client data → commit log → memtable.
  - Durability in the case of unexpected shutdown. On startup, any changes in log will be applied to tables.
- Memtable
  - In-memory structures to write Cassandra buffers.
  - One active memtable per table.
- Sorted String Table
  - Immutable data files for persisting data on disk.
  - Multiple memtables merged into single SSTable.
- LSM Tree (Log structure Merge Tree)
  - Disk based data structure to provide low-cost indexing for a file, in which records are to be inserted at very high rate. also efficient in-mem structure.





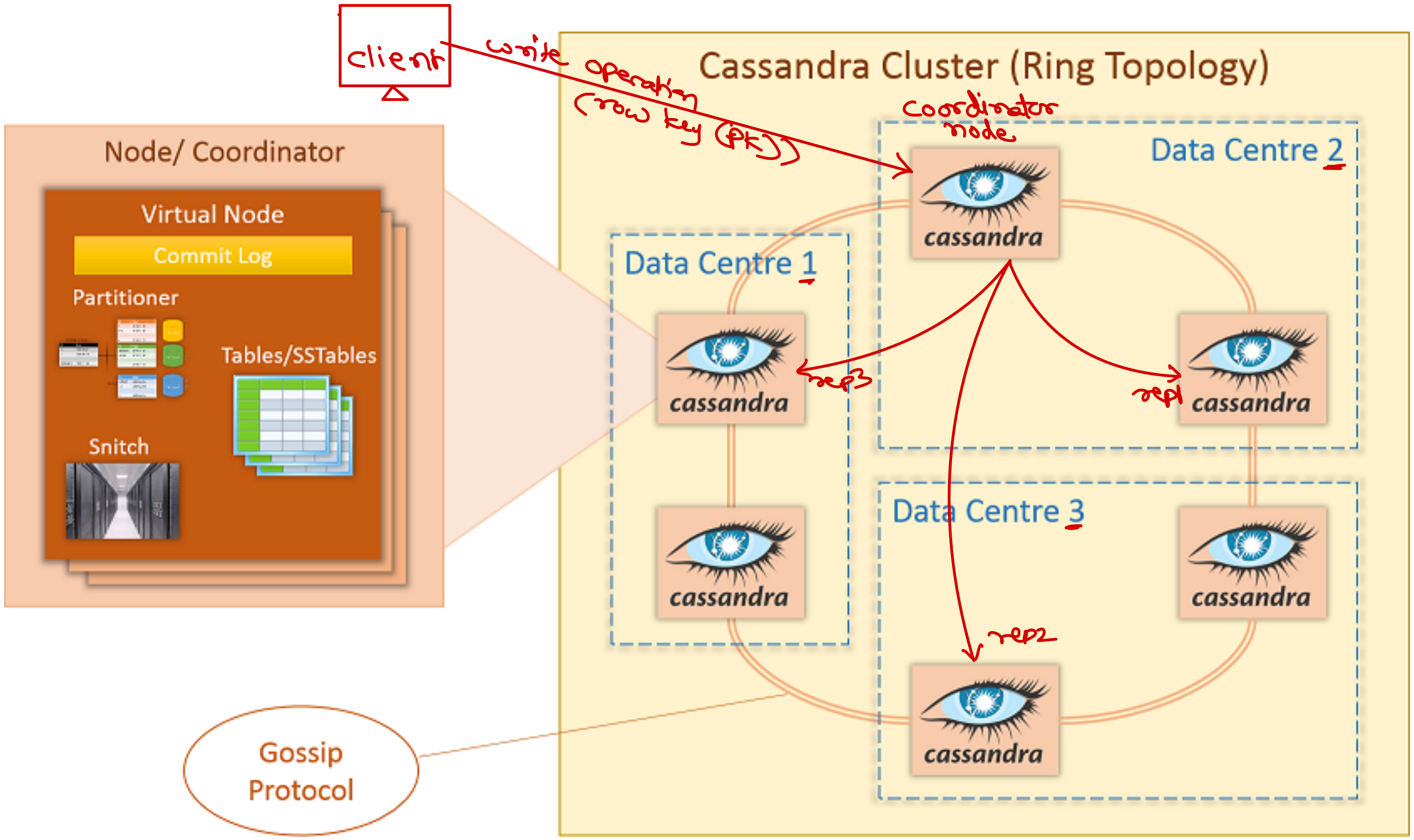
Architecture

- Bloom filter
  - In the read path, Cassandra merges data on disk (in SSTables) with data in RAM (in memtables).
  - To avoid checking every SSTable data file for the partition being requested, Cassandra employs a data structure known as a bloom filter.
  - Bloom filters are a probabilistic data structure that allows Cassandra to determine one of two possible states
    - The data definitely does not exist in the given file
    - The data probably exists in the given file.



Architecture

- Gossip protocol
  - Each node learn about cluster topology.
  - Communicate among nodes.
  - Detection of faulty nodes.
- Snitch
  - Snitch helps map IPs to racks and data centers.
  - This info is used for replica location and other tasks.



Data Model

Table/Column Family

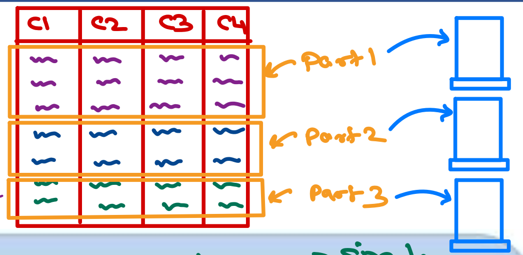


Table → like hash table

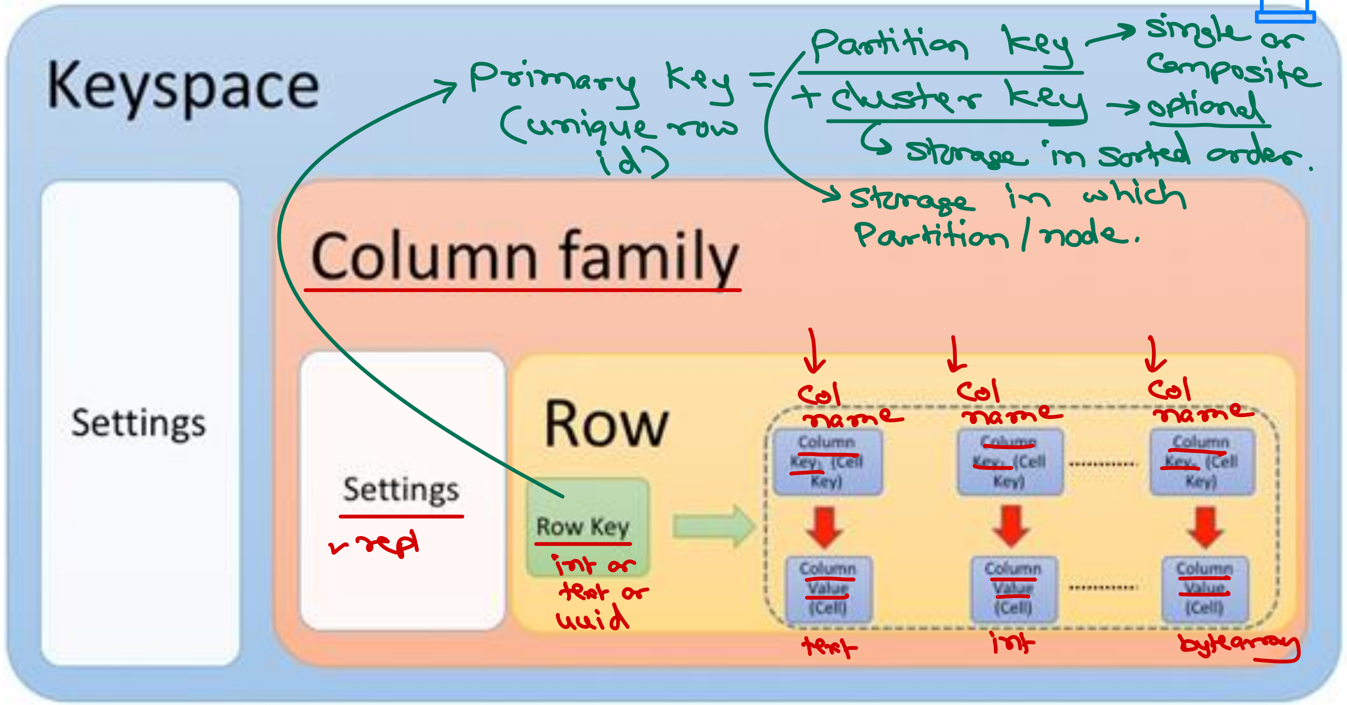
key = pri key (row id)

value = hash table

key = col name

value = cell value

- Basic data model is Rows and Columns and distributed across the nodes. → partitions
  - Rows are distributed across nodes by Sharding on primary key. (Partition key)
  - Columns are distributed across nodes column groups.
- Each row is identified by unique key (primary key).
- One or more column families are contained inside Key-space (Table in CQL 3.0). → db schema
- A column family contain super-columns or columns.



Primary key (unique row id) = Partition key → single or composite
+ cluster key → optional
↳ storage in sorted order.
→ storage in which Partition / node.

people

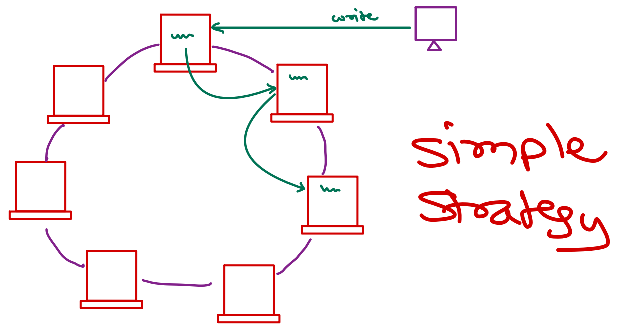
	name	contact
r1	sal: mr. fname: nilesh lname: ghule	mobile: 9527331338 email: nilesh@sun.com
r2	fname: nitin lname: kudale	email: nitin@sun.com fax: 24260308
r3	cname: sunbeam Infotech	site: sunbeaminfo.com

Super column → set of columns that can be added dynamically.

$$\begin{tikzcd} & \text{Super column} \arrow[dl] \arrow[dr] & \\ \text{name} \arrow[d] & & \text{contact} \arrow[d] \\ \text{sal, fname, lname} & & \text{mobile, email, fax} \end{tikzcd}$$

Cassandra Keyspace

- DESCRIBE KEYSPACES;
- CREATE KEYSPACE dbda WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 3};
- USE dbda;
- DESCRIBE TABLES;



- Strategy
  - Simple Strategy: Simple strategy is used in the case of one data center. In this strategy, the first replica is placed on the selected node and the remaining nodes are placed in clockwise direction in the ring without considering rack or node location.
  - Network Topology Strategy: This strategy is used in the case of more than one data centers. In this strategy, you have to provide replication factor for each data center separately.

CQL Data types

- ascii - US-ascii character string
- bigint - 64-bit signed long ints
- blob - Arbitrary bytes in hexadecimal
- boolean - True or False
- counter - Distributed counter values 64 bit
- decimal - Variable precision decimal
- double - 64-bit floating point
- float - 32-bit floating point
- frozen - Tuples, collections, UDT containing CQL types
- inet - IP address in ipv4 or ipv6 string format
- int - 32 bit signed integer
- list - Collection of elements
- map - JSON style collection of elements
- set - Sorted collection of elements
- text - UTF-8 encoded strings
- timestamp - ID generated with date+time as int/string
- timeuuid - Type 1 uuid
- tuple - A group of 2,3 fields
- uuid - Standard uuid (128-bit)
- varchar - UTF-8 encoded string
- varint - Arbitrary precision integer

RDBMS	Cassandra
✓ Database/schema	✓ keyspace
✓ Table	✓ Table or Column Family
✓ Row	✓ Row
✓ Primary key	✓ Row key or Primary key
✓ Column name	✓ Column name
✓ Column value	✓ column or cell value
✓ Foreign keys, joins	✓ Collections
✓ Indexes	✓ Indexes

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>