Cassandra Query Language

NIMMY PRABHA

Keyspace

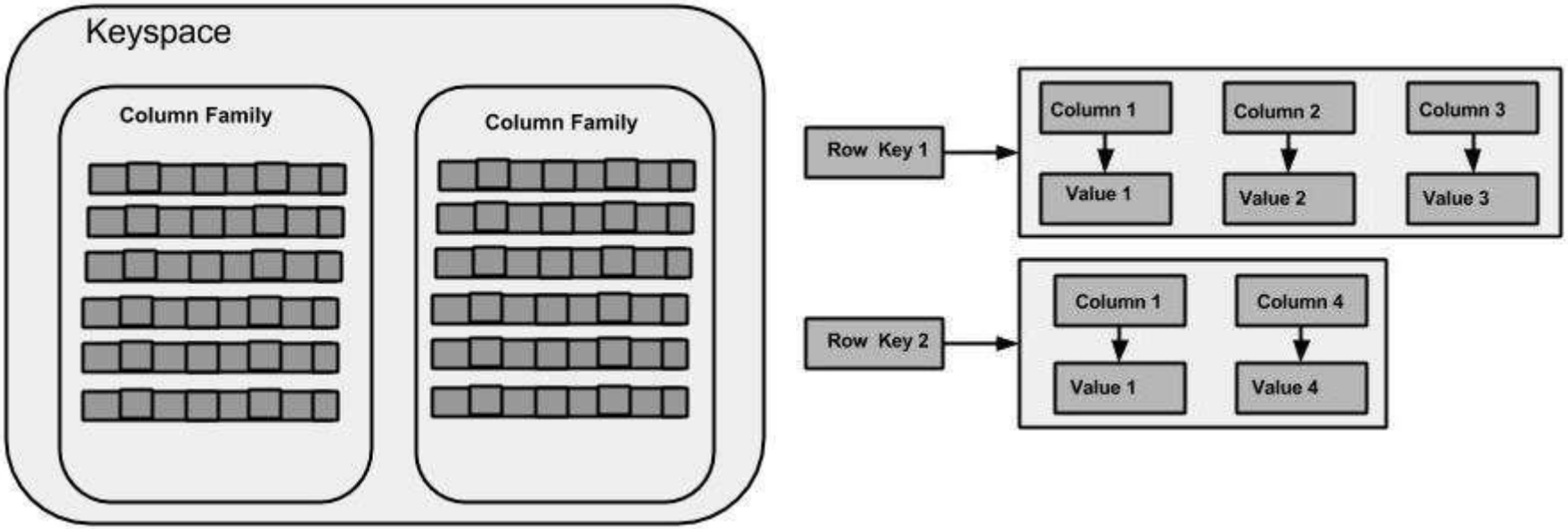
- **Replication factor** − It is the number of machines in the cluster that will receive copies of the same data.
- **Replica placement strategy** − It is nothing but the strategy to place replicas in the ring. We have strategies such as simple strategy(rack-aware strategy), old network topology strategy (rack-aware strategy), and network topology strategy (datacenter-shared strategy).

- CREATE KEYSPACE Keyspace name WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 1};

```
cqlsh> CREATE KEYSPACE employee WITH REPLICATION={'class':'SimpleStrategy','repl
ication_factor':1};
```

Column Family

- A Cassandra column family has the following attributes −
- **keys_cached** − It represents the number of locations to keep cached per SSTable.
- **rows_cached** − It represents the number of rows whose entire contents will be cached in memory.
- **preload_row_cache** − It specifies whether you want to pre-populate the row cache.



Column & Super Column

- A **column** is the basic data structure of Cassandra with three values, namely key or column name, value, and a time stamp. Given below is the structure of a column**.**

Column		
name : byte[]	value : byte[]	clock : clock[]

- A **super column** is a special column, therefore, it is also a key-value pair. But a super column stores a map of sub-columns.

Super Column	
name : byte[]	cols : map<byte[], column>

- **HELP** - Displays help topics for all cqlsh commands.
- **CAPTURE** - Captures the output of a command and adds it to a file.
- **CONSISTENCY** - Shows the current consistency level, or sets a new consistency level.
- **COPY** - Copies data to and from Cassandra.
- **DESCRIBE** - Describes the current cluster of Cassandra and its objects.
- **EXPAND** - Expands the output of a query vertically.
- **EXIT** - Using this command, you can terminate cqlsh.
- **PAGING** - Enables or disables query paging.
- **SHOW** - Displays the details of current cqlsh session such as Cassandra version, host,
- **SOURCE** - Executes a file that contains CQL statements.
- **TRACING** - Enables or disables request tracing.
- **CQL Data Definition Commands**
- **CREATE KEYSPACE** - Creates a KeySpace in Cassandra.
- **USE** - Connects to a created KeySpace.
- **ALTER KEYSPACE** - Changes the properties of a KeySpace.
- **DROP KEYSPACE** - Removes a KeySpace
- **CREATE TABLE** - Creates a table in a KeySpace.
- **ALTER TABLE** - Modifies the column properties of a table.
- **DROP TABLE** - Removes a table.
- **TRUNCATE** - Removes all the data from a table.
- **CREATE INDEX** - Defines a new index on a single column of a table.
- **DROP INDEX** - Deletes a named index.

CQL Data Manipulation Commands

- **INSERT** - Adds columns for a row in a table.
- **UPDATE** - Updates a column of a row.
- **DELETE** - Deletes data from a table.
- **BATCH** - Executes multiple DML statements at once.
- CQL Clauses
- **SELECT** - This clause reads data from a table
- **WHERE** - The where clause is used along with select to read a specific data.
- **ORDERBY** - The orderby clause is used along with select to read a specific data in a specific order.

CQLSH Commands

- HELP
- CONSISTENCY

```
cqlsh> CONSISTENCY
Current consistency level is ONE.
```

- Describe Keyspaces

```
cqlsh> HELP

Documented shell commands:
==========================
CAPTURE  CLS          COPY  DESCRIBE  EXPAND  LOGIN    SERIAL  SOURCE   UNICODE
CLEAR    CONSISTENCY  DESC  EXIT      HELP    PAGING   SHOW    TRACING

CQL help topics:
================
AGGREGATES               CREATE_KEYSPACE           DROP_TRIGGER      TEXT
ALTER_KEYSPACE           CREATE_MATERIALIZED_VIEW  DROP_TYPE         TIME
ALTER_MATERIALIZED_VIEW  CREATE_ROLE               DROP_USER         TIMESTAMP
ALTER_TABLE              CREATE_TABLE              FUNCTIONS         TRUNCATE
ALTER_TYPE               CREATE_TRIGGER            GRANT             TYPES
ALTER_USER               CREATE_TYPE               INSERT            UPDATE
APPLY                    CREATE_USER               INSERT_JSON       USE
ASCII                    DATE                      INT               UUID
BATCH                    DELETE                    JSON
BEGIN                    DROP_AGGREGATE            KEYWORDS
BLOB                     DROP_COLUMNFAMILY         LIST_PERMISSIONS
BOOLEAN                  DROP_FUNCTION             LIST_ROLES
COUNTER                  DROP_INDEX                LIST_USERS
CREATE_AGGREGATE         DROP_KEYSPACE             PERMISSIONS
CREATE_COLUMNFAMILY      DROP_MATERIALIZED_VIEW    REVOKE
CREATE_FUNCTION          DROP_ROLE                 SELECT
CREATE_INDEX             DROP_TABLE                SELECT_JSON
```

```
cqlsh> Describe Keyspaces

people          system_auth  emp                   employee
system_schema   system       system_distributed    system_traces
```

- cqlsh> CAPTURE 'C:\Users\Admin\Desktop\Cassandra-1'
- Already capturing output to C:\Users\Admin\Desktop\Cassandra-1. Use CAPTURE OFF to disable.
- cqlsh> capture off

- Use employee;

```
cqlsh> use employee
   ... ;
cqlsh:employee> use emp
           ... ;
cqlsh:emp>
```

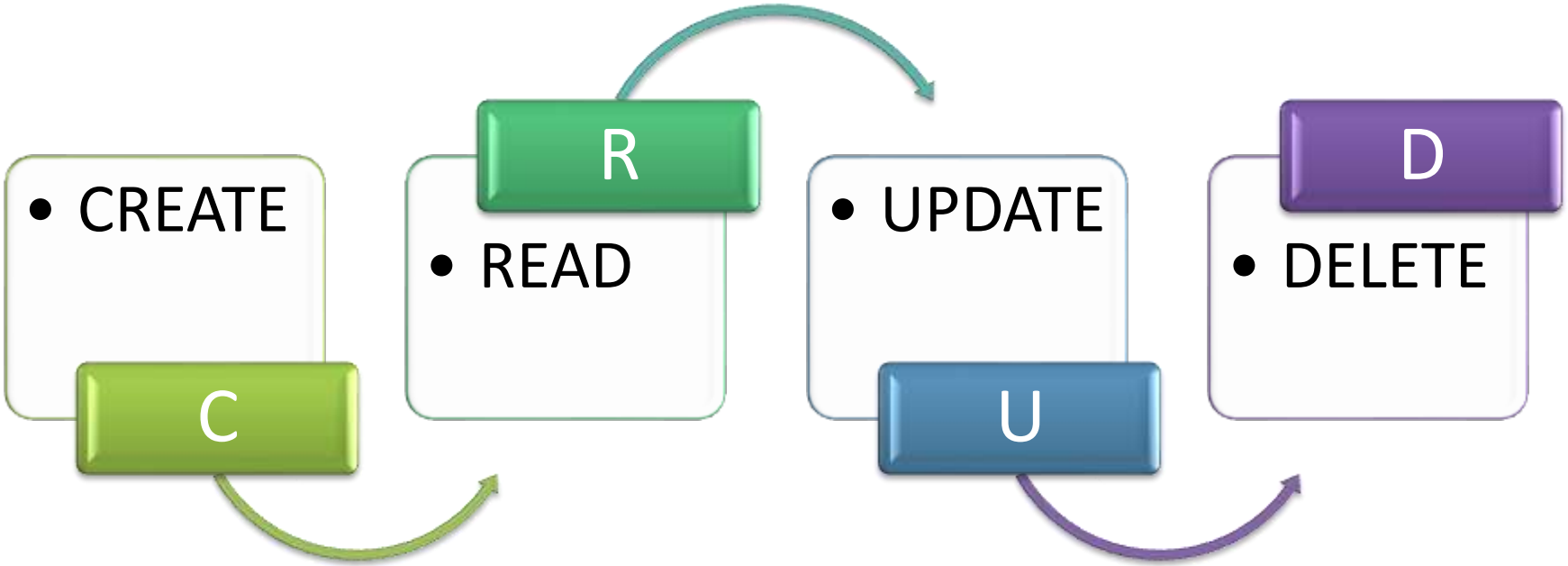
- CREATE TABLE EMP_INFO (empid int PRIMARY KEY, emp_name text, DOJ timestamp, lastsalary double);
- Describe tables

```
cqlsh:emp> CREATE TABLE EMP_INFO(empid int PRIMARY KEY,empname text, DOJ timesta
mp, lastsalary double);
cqlsh:emp> describe tables;

emp_info
```

- Describe tables EMP_INFO

```
cqlsh:emp> describe table EMP_INFO

CREATE TABLE emp.emp_info (
    empid int PRIMARY KEY,
    doj timestamp,
    empname text,
    lastsalary double
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCom
pactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandr
a.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

CRUD OPERTION



INSERT INTO cycling.cyclist_name (id, lastname, firstname) VALUES (6ab09bec-e68e-48d9-a5f8-97e6fb4c9b47, 'KRUIKSWIJK','Steven') **USING TTL** 86400 **AND TIMESTAMP** 123456789;

```
cqlsh:emp> INSERT INTO emp.emp_info(empid,DOJ,empname,lastsalary) VALUES(1,'2015
-05-25','arul',45698.25);
```

- BEGIN BATCH INSERT INTO cycling.cyclist_name (id, lastname, firstname) VALUES (6ab09bec-e68e-48d9-a5f8-97e6fb4c9b47, 'KRUIKSWIJK','Steven') INSERT INTO cycling.cyclist_name (id, lastname, firstname) VALUES (6ab09bec-e68e-48d9-a5f8-97e6fb4c9b47, 'KRUIKSWIJK','Steven') APPLY BATCH

```
cqlsh:emp> BEGIN BATCH INSERT INTO emp.EMP_INFO(empid,DOJ,empname,lastsalary) VA
LUES (5,'2010-12-15','VIDHYA',145452.25) INSERT INTO emp.EMP_INFO(empid,DOJ,empn
ame,lastsalary) VALUES(6,'2012-11-20','MARY',74256.85) APPLY BATCH;
```

- SELECT * FROM EMP_INFO;

empid	doj	empname	lastsalary
5	2010-12-14 18:30:00+0000	VIDHYA	1.4545e+05
1	2015-05-24 18:30:00+0000	arul	45698.25
2	2016-04-04 18:30:00+0000	ANU	48698.2
4	2014-11-19 18:30:00+0000	MICHAEL	47256.85
6	2012-11-19 18:30:00+0000	MARY	74256.85
3	2015-12-24 18:30:00+0000	PREM	11452.25

select * from emp_info where empid IN(1,2,3,4);

```
cqlsh:emp> select * from emp_info where empid IN(1,2,3,4);

 empid | doj                         | empname | lastsalary
-------+-----------------------------+---------+------------
     1 | 2015-05-24 18:30:00+0000 |    arul |   45698.25
     2 | 2016-04-04 18:30:00+0000 |     ANU |    48698.2
     3 | 2015-12-24 18:30:00+0000 |    PREM |   11452.25
     4 | 2014-11-19 18:30:00+0000 | MICHAEL |   47256.85

(4 rows)
```

- CREATE INDEX ON EMP_INFO (empname);
- SELECT * FROM EMP_INFO WHERE empname='MICHAEL';

```
cqlsh:emp> SELECT * FROM EMP_INFO WHERE empname='MICHAEL';

 empid | doj                         | empname | lastsalary
-------+-----------------------------+---------+------------
     4 | 2014-11-19 18:30:00+0000 | MICHAEL |   47256.85
```

- select * from emp_info LIMIT 2;

```
cqlsh:emp> SELECT * FROM EMP_INFO LIMIT 2;

 empid | doj                         | empname | lastsalary
-------+-----------------------------+---------+------------
     5 | 2010-12-14 18:30:00+0000 |  VIDHYA | 1.4545e+05
     1 | 2015-05-24 18:30:00+0000 |    arul |   45698.25
```

- UPDATE EMP_INFO SET empname='stephen' WHERE empid=4;

```
cqlsh:emp> UPDATE emp.EMP_INFO SET empname='STEPHEN' WHERE empid=4;
cqlsh:emp> SELECT * FROM EMP_INFO WHERE EMPID=4
       ... ;

 empid | doj                         | empname | lastsalary
-------+-----------------------------+---------+------------
     4 | 2014-11-19 18:30:00+0000 | STEPHEN |   47256.85
```

Delete from emp_info where empid=1;

```
cqlsh:emp> delete from emp_info where empid=1;
cqlsh:emp> select * from emp_info
       ... ;

 empid | doj                             | empname | lastsalary
-------+---------------------------------+---------+------------
     5 | 2010-12-14 18:30:00+0000 |  VIDHYA | 1.4545e+05
     2 | 2016-04-04 18:30:00+0000 |     ANU |    48698.2
     4 | 2014-11-19 18:30:00+0000 | STEPHEN |   47256.85
     6 | 2012-11-19 18:30:00+0000 |    MARY |   74256.85
     3 | 2015-12-24 18:30:00+0000 |    PREM |   11452.25
```