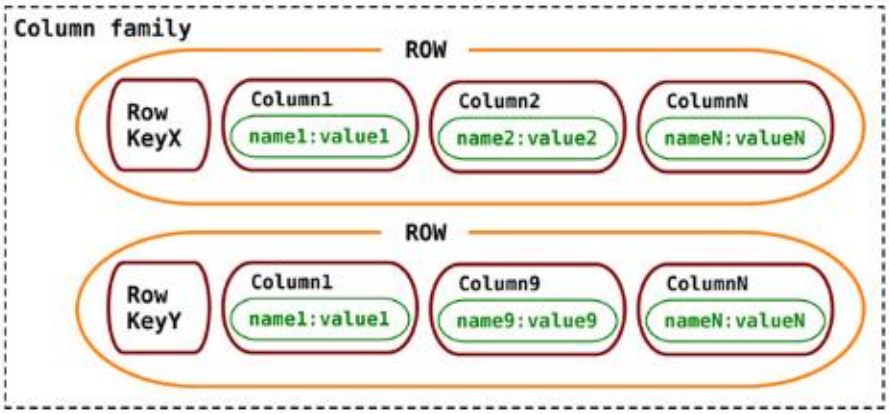
Modelo Família de Colunas

- A unidade de informação é a coluna (chave-valor) e um "registro" é uma família de colunas.
- Coluna:
  - Par chave-valor
- Super coluna:
  - *Array* de colunas
- Família de colunas:
  - Um *container* para colunas ordenadas por seus nomes. Família de colunas são referenciadas e classificadas por *rowkeys*.
- Família de super colunas:
  - Um *container* para super colunas ordenados por seus nomes. Famílias como Coluna, Famílias de super coluna são referenciados e ordenados por *rowkeys*.

Fonte: http://www.sinbadsoft.com/blog/cassandra-data-model-cheat-sheet

Modelo Família de Colunas



Fonte: Sadalage e Fowler 2012

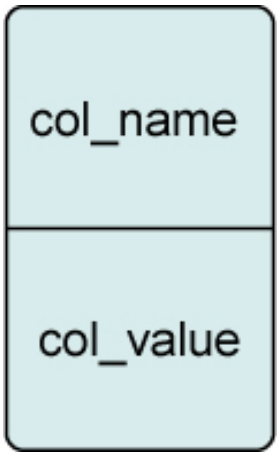
Tabela relacional

Id	CPF	Nome	DTNasc	Cidade	Estado
1	11111111	Ivon Canedo	1947-05-04	Goiânia	GO
2	22222222	Marcio Kanutto	1950-10-30	Goiânia	GO
3	33333333	Maria Ribeiro	1943-03-10	Rio de Janeiro	RJ
4	44444444	Carla Lemos	1960-06-30	Goiânia	GO
5	55555555	Marcos Cintra	1952-07-30	São Paulo	SP

Tabela Família de Colunas

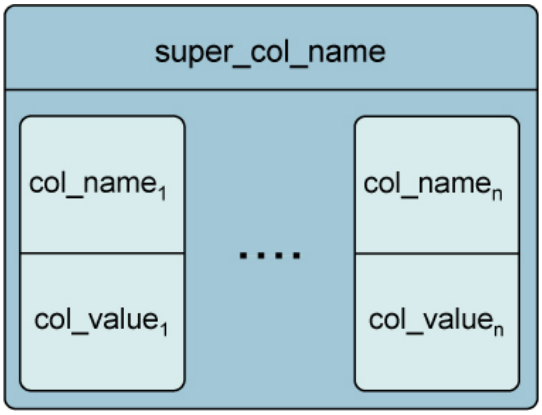
ID	Colunas
1	Cpf=1111111 nome=Ivon Canedo DtNasc=1947-05-04
2	Cpf=2222222 nome=Marcio Kanutto DtNasc=1950-10-30
3	Cpf=3333333 nome=Maria Ribeiro Cidade=Rio de Janeiro
4	Cpf=4444444 nome=Carla Lemos Cidade=Goiânia
5	Cpf=5555555 nome=Marcos Cintra

Coluna



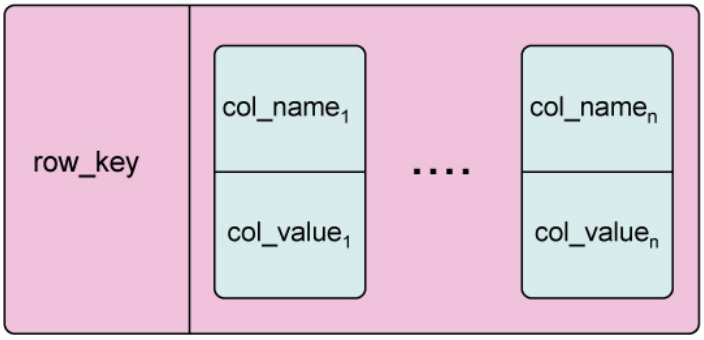
Fonte: http://www.sinbadsoft.com/blog/cassandra-data-model-cheat-sheet

Super Coluna



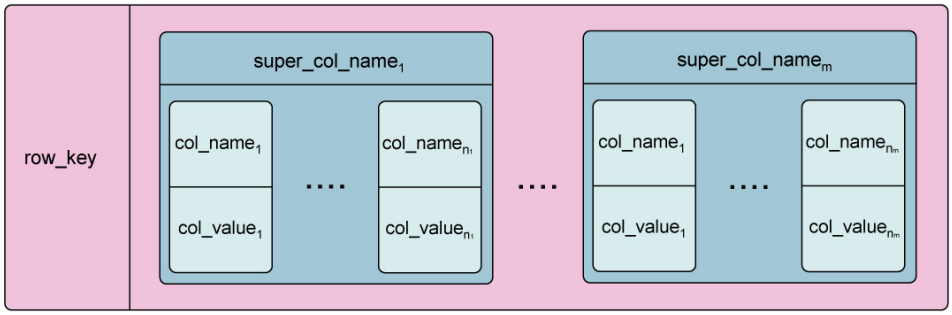
Fonte: http://www.sinbadsoft.com/blog/cassandra-data-model-cheat-sheet

Família de Colunas



Fonte: http://www.sinbadsoft.com/blog/cassandra-data-model-cheat-sheet

Família de Super Colunas



Fonte: http://www.sinbadsoft.com/blog/cassandra-data-model-cheat-sheet

Ranking Família de Colunas

5 systems in ranking, October 2014

Rank	Last Month	DBMS	Database Model	Score	Changes
1.	1.	Cassandra	Wide column store	85.70	-2.16
2.	2.	HBase	Wide column store	47.10	+2.08
3.	3.	Accumulo	Wide column store	2.71	+0.00
4.	4.	Hypertable	Wide column store	0.61	-0.04
5.	5.	Sqrrl	Multi-model	0.18	+0.04

Fonte: db-engines.com

Apache Cassandra

- Também chamado de C*
- É um banco de dados de fonte aberto, distribuído, orientado a colunas, descentralizado, elasticamente escalável e tolerante a falhas, com consistência configurável.
- Baseia a sua concepção de distribuição no Dynamo, da Amazon e o seu modelo de dados no Bigtable do Google.
- Suportado comercialmente pela DataStax.

- http://cassandra.apache.org

Executáveis

- Cassandra
  - Bat que executa o cassandra.
- Cqlsh
  - Bat que executa o shell do cassandra.
  - Feito em python 2.
- Stop-server –p <pidfile>
  - Para parar o servidor.

Cassandra vs. Relacional

Relacional	Cassandra
Structured Data, Fixed Schema	Unstructured Data, Flexible Schema
"Array of Arrays" 2D: ROW x COLUMN	"Nested Key-Value Pairs" 3D: ROW Key x COLUMN key x COLUMN values
DATABASE	KEYSPACE
TABLE	TABLE a.k.a COLUMN FAMILY
ROW	ROW a.k.a PARTITION. Unit of replication.
COLUMN	COLUMN [Name, Value, Timestamp]. a.k.a CLUSTER. Unit of storage. Up to 2 billion columns per row.
FOREIGN KEYS, JOINS, ACID Consistency	Referential Integrity not enforced, so ACID. BUT relationships may represented by using COLLECTIONS.

Cassandra não suporta Joins nem SubQueries

Tipos de dados

Internal Type	CQL Name	Description
BytesType	Blob	Arbitrary hexadecimal bytes (no validation)
AsciiType	Ascii	US-ASCII character string
UTF8Type	Text, varchar	UTF-8 encoded string
IntegerType	Varint	Arbitrary-precision integer
Int32Type	Int	4-byte integer
LongType	Bigint	8-byte long
UUIDType	Uuid	Type 1 or type 4 UUID
TimeUUIDType	Timeuuid	Type 1 UUID only (CQL3)
DateType	Timestamp	Date plus time, encoded as 8 bytes since epoch
BooleanType	Boolean	True or false
FloatType	Float	4-byte floating point
DoubleType	Double	8-byte floating point
DecimalType	Decimal	Variable-precision decimal
CounterColumnType	Counter	Distributed counter value (8-byte long)

Thrift RPC

```
// Your Column
Column col = new Column(ByteBuffer.wrap("name".getBytes()));
col.setValue(ByteBuffer.wrap("value".getBytes()));
col.setTimestamp(System.currentTimeMillis());
// Don't ask
ColumnOrSuperColumn cosc = new ColumnOrSuperColumn();
cosc.setColumn(col);
// Prepare to be amazed
Mutation mutation = new Mutation();
mutation.setColumnOrSuperColumn(cosc);
List<Mutation> mutations = new ArrayList<Mutation>();
mutations.add(mutation);
Map mutations_map = new HashMap<ByteBuffer, Map<String,List<Mutation>>>();
Map cf_map = new HashMap<String, List<Mutation>>();
cf_map.set("Standard1", mutations);
mutations_map.put(ByteBuffer.wrap("key".getBytes()),cf_map);
cassandra.batch_mutate(mutations_map,consistency_level);
```

CQL

- `INSERT INTO (id, name) VALUES ('key', 'value');`

CQL

- Cassandra Query Language
- Provê uma interface bem mais simples que o Thrift
- Adiciona uma camada de abstração que esconde os detalhes de implementação e fornece sintaxes nativas para coleções e outras codificações comuns

Criando um KeySpace

```
CREATE KEYSPACE cursodb WITH REPLICATION =
{'class': 'SimpleStrategy',
'replication_factor' : 1};

USE cursodb;
```

Criando uma Família de colunas

```
CREATE COLUMNFAMILY songs (
     id uuid PRIMARY KEY,
     title text,
     album text,
     artist text,
     data blob
);
```

Criando uma Família de colunas

```
CREATE TABLE playlists(
     id uuid,
     song_order int,
     song_id uuid,
     title text,
     album text,
     artist text,
     PRIMARY KEY(id, song_order)
);
```

Inserindo registros

```
INSERT INTO playlists (
      id,
      song_order,
      song_id,
      title,
      artist,
      album
) VALUES (
      62c36092-82a1-3a00-93d1-46196ee77204,
      4,
      7db1a490-5878-11e2-bcfd-0800200c9a66,
      'Ojo Rojo',
      'Fu Manchu',
      'No One Rides for Free'
);
```

Alterando registros

```
UPDATE playlists SET album = 'No One Rides for Free Remastered' WHERE id = 62c36092-82a1-3a00-93d1-46196ee77204;
```

```
UPDATE users
SET email = 'janedoe@abc.com'
WHERE login = 'jdoe'
IF email = 'jdoe@abc.com';
```

Query

```
SELECT * FROM playlists
WHERE id=62c36092-82a1-3a00-93d1-46196ee77204
ORDER BY song_order DESC LIMIT 50;
```

id	song_order	album	artist	song_id	title
62c36092...	4	No One Rides for Free	Fu Manchu	7db1a490...	Ojo Rojo
62c36092...	3	Roll Away	Back Door Slam	2b09185b...	Outside Woman Blues
62c36092...	2	We Must Obey	Fu Manchu	8a172618...	Moving in Stereo
62c36092...	1	Tres Hombres	ZZ Top	a3e64f8f...	La Grange

Usando coleções

- Existem 3 tipos de coleções:
  - Set
  - List
  - Map

Usando coleções Set

```
CREATE TABLE users (user_id text PRIMARY KEY,
first_name text, last_name text, emails set<text>);

INSERT INTO users (user_id, first_name, last_name,
emails) VALUES('frodo', 'Frodo', 'Baggins',
{'f@baggins.com', 'baggins@gmail.com'});

UPDATE users SET emails = emails +
{'fb@friendsofmordor.org'} WHERE user_id = 'frodo';

UPDATE users SET emails = emails -
{'fb@friendsofmordor.org'} WHERE user_id = 'frodo';

UPDATE users SET emails = {} WHERE user_id = 'frodo';
```

Usando coleções List

```
ALTER TABLE users ADD top_places list<text>;

UPDATE users SET top_places = ['rivendell',
'rohan'] WHERE user_id = 'frodo';

UPDATE users SET top_places = ['the shire'] +
top_places WHERE user_id = 'frodo';

UPDATE users SET top_places[2] = 'riddermark'
WHERE user_id = 'frodo';
```

Usando coleções Map

```
ALTER TABLE users ADD todo map<timestamp, text>;

UPDATE users SET todo = { '2012-9-24' : 'enter mordor', '2012-10-2 12:00' : 'throw ring into mount doom'} WHERE user_id = 'frodo';

UPDATE users SET todo['2012-10-2 12:00'] = 'throw my precious into mount doom' WHERE user_id = 'frodo';

DELETE todo['2012-9-24'] FROM users WHERE user_id = 'frodo';
```

Criando Índices

- `CREATE INDEX album_name ON playlists (album);`
- `CREATE INDEX title_name ON playlists (title);`

Cassandra em Java

- Use o driver da Datastax
  - http://www.datastax.com/download#dl-datastax-drivers

Cassandra em Java

```
Cluster cluster = new
Cluster.Builder().addContactPoint("127.0.0.1"
).build();
Session session = cluster.connect();
ResultSet rs = session.execute("select id,
des from cursodb.teste");
for (Row row : rs) {
    System.out.println(row.getInt(0) + ":" +
row.getString(1));
}
session.close();
cluster.close();
```