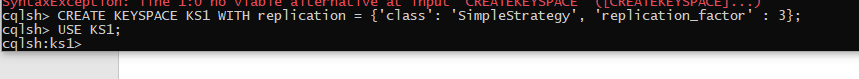
**Atividade Cassandra**

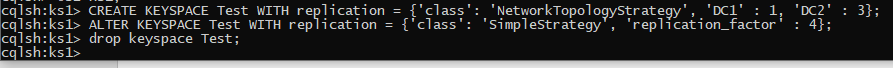
**Keyspace**

Ao criar um Keyspace tem duas configurações necessárias:

* Class: estratégia de cópia, o 'SimpleStrategy' é a para data center simples, único, e o 'NetworkTopologyStrategy' para mais de 1 data center
* Replication factor – fator de replicação, no caso de data center complexos, definir por data center

Exemplos:





***Hands on:***

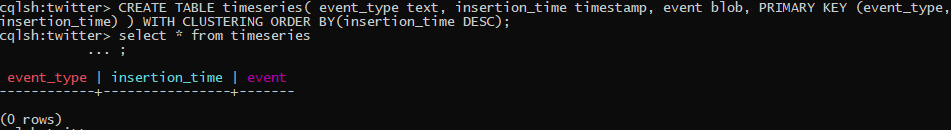
CREATE KEYSPACE twitter WITH replication = {'class': 'SimpleStrategy', 'replication\_factor' : 1};



**Tabelas**

O primary key são os campos de colunas que concatenados farão a partição da tabela, e ela será ordenada pelo insertion\_time.

Exemplos:



Armazenamento compacto



Cashing: Tamanho de cash que vai manter otimizado, e seta por quantidade de linhas por partição

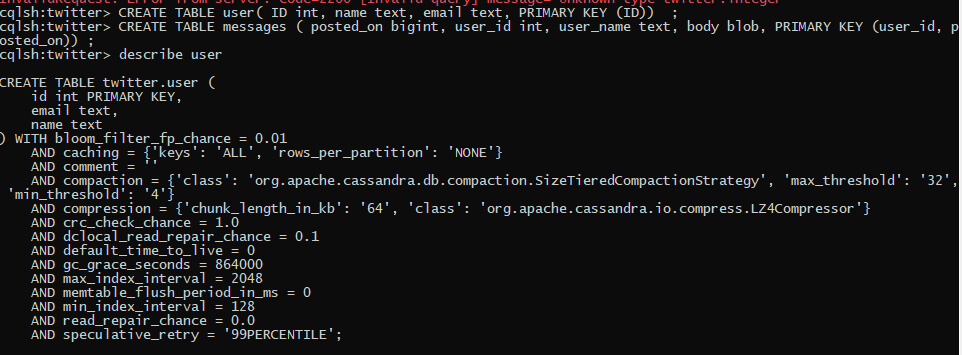


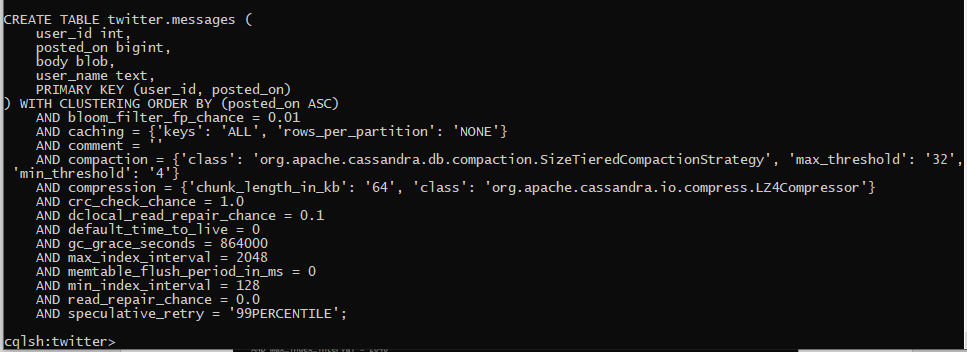
***Hands on:***

Sem o order by ele automaticamente ordena pela primary key de forma ascendente

CREATE TABLE user( ID integer, name text, email text, PRIMARY KEY (ID))

CREATE TABLE messages ( posted\_on bigint, user\_id int, user\_name text, body text, PRIMARY KEY (user\_id, posted\_on))

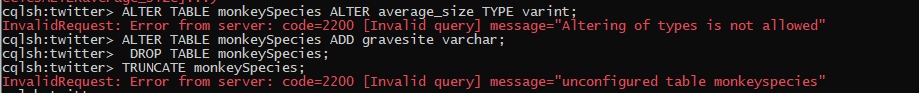




Comandos de alteração de tabela para alterar tipo, adicionar coluna, deletar tabela e apagando dados de uma tabela.

DROP TABLE monkeySpecies;

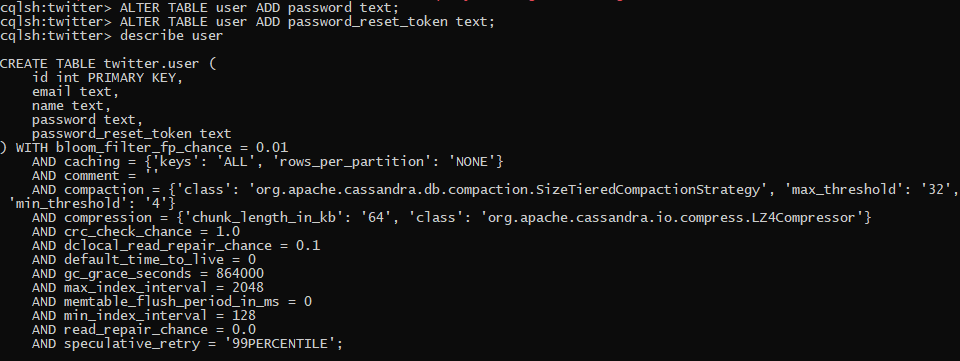
TRUNCATE monkeySpecies;



***Hands On***

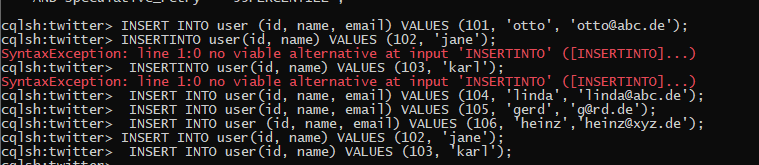
ALTER TABLE user ADD password text;

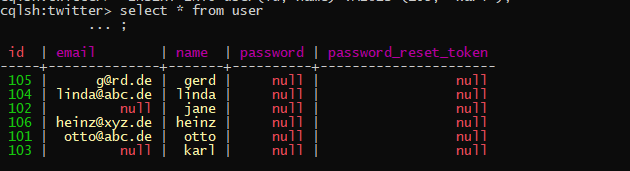
ALTER TABLE user ADD password\_reset\_token text;



O insert é igual a do sql:

Exemplos aula:





***Hands on:***

INSERT INTO messages(user\_id, posted\_on, user\_name, body) VALUES (101, 1384895178, 'otto', 'Hello World!');

INSERT INTO messages (user\_id, posted\_on, user\_name, body) VALUES (101, 1384895319, 'otto', 'Hello again... ');

INSERT INTO messages(user\_id, posted\_on, user\_name, body) VALUES (104, 1384895222, 'linda', 'Hi, Otto');

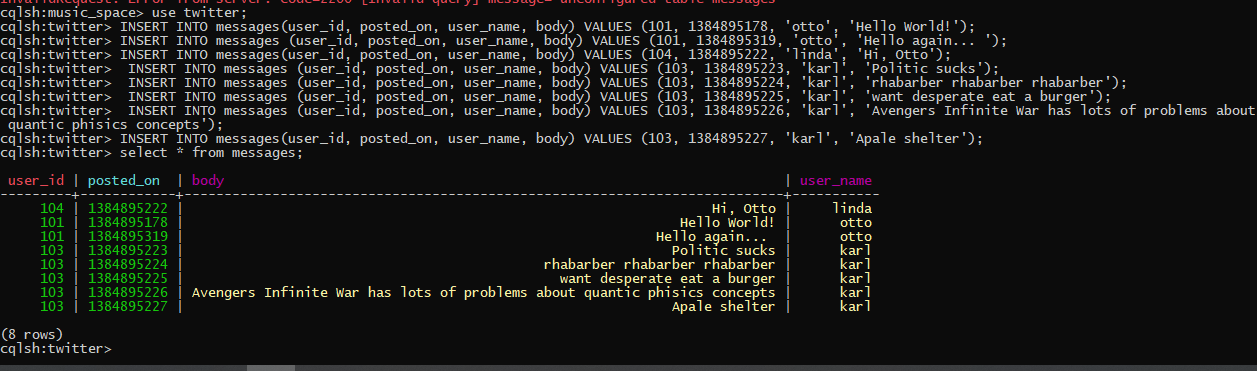
INSERT INTO messages (user\_id, posted\_on, user\_name, body) VALUES (103, 1384895223, 'karl', 'Politic sucks');

INSERT INTO messages (user\_id, posted\_on, user\_name, body) VALUES (103, 1384895224, 'karl', 'rhabarber rhabarber rhabarber');

INSERT INTO messages (user\_id, posted\_on, user\_name, body) VALUES (103, 1384895225, 'karl', 'want desperate eat a burger');

INSERT INTO messages (user\_id, posted\_on, user\_name, body) VALUES (103, 1384895226, 'karl', 'Avengers Infinite War has lots of problems about quantic phisics concepts');

INSERT INTO messages(user\_id, posted\_on, user\_name, body) VALUES (103, 1384895227, 'karl', 'Apale shelter');



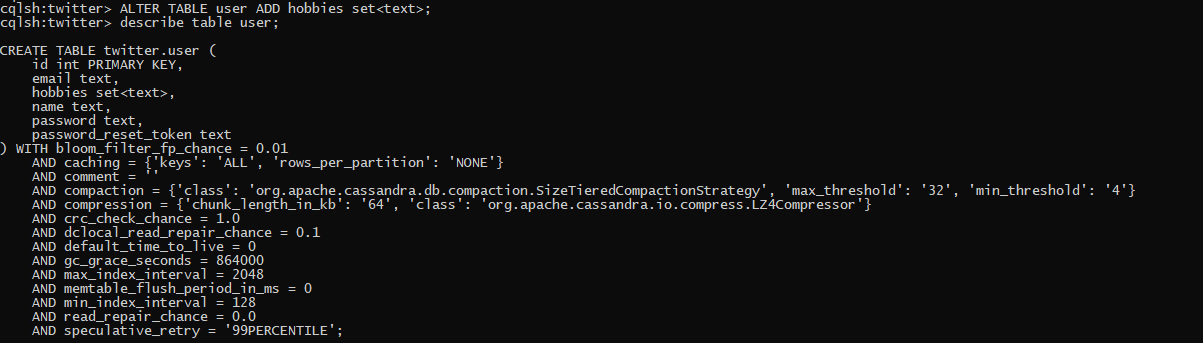
**Colunas**

Existem 4 tipos de colunas multivaloradas

* Set – chave valor simples
* List – array
* Map – chave valor complexo
* Tuples – uma composição de colunas dentro da linha

***Hands on***

***ALTER TABLE user ADD hobbies set<text>;***

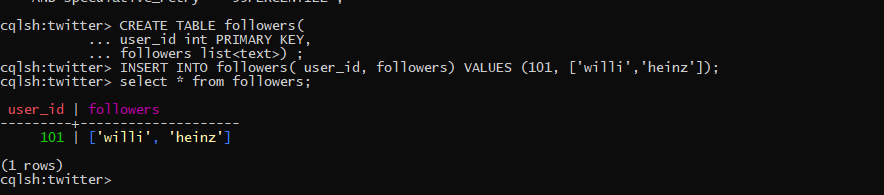


CREATE TABLE followers(

user\_id int PRIMARY KEY,

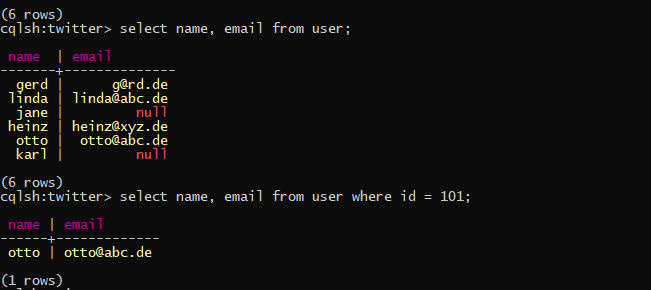
followers list<text>) ;

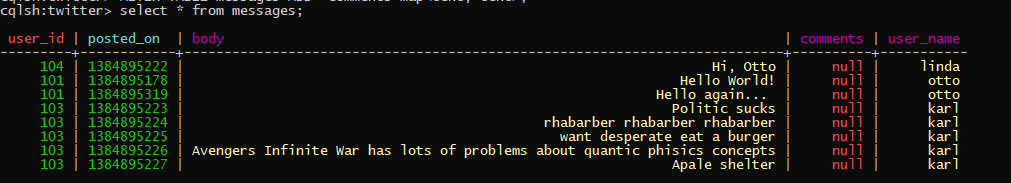
INSERT INTO followers( user\_id, followers) VALUES (101, ['willi','heinz']);



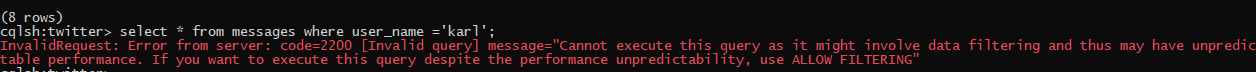


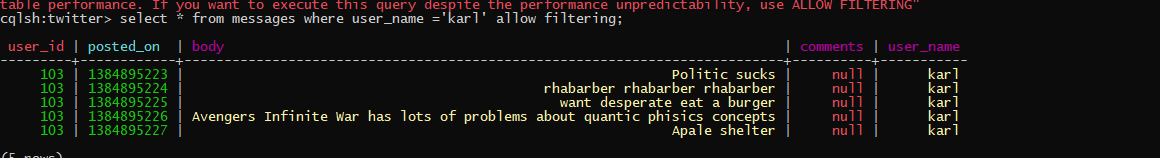
**Select**





Ao tentar selecionar por um campo que não é a primary key, o Cassandra da erro falando que a performance pode ser ruim e que consegue continuar usando o allow filtering ( que não é recomendado utilizar). O recomendado é criar outra tabela com esse campo como primary key.





**Índices**

Os índices são utilizados para conseguir filtrar um campo sem ser primary key, não é recomendado.

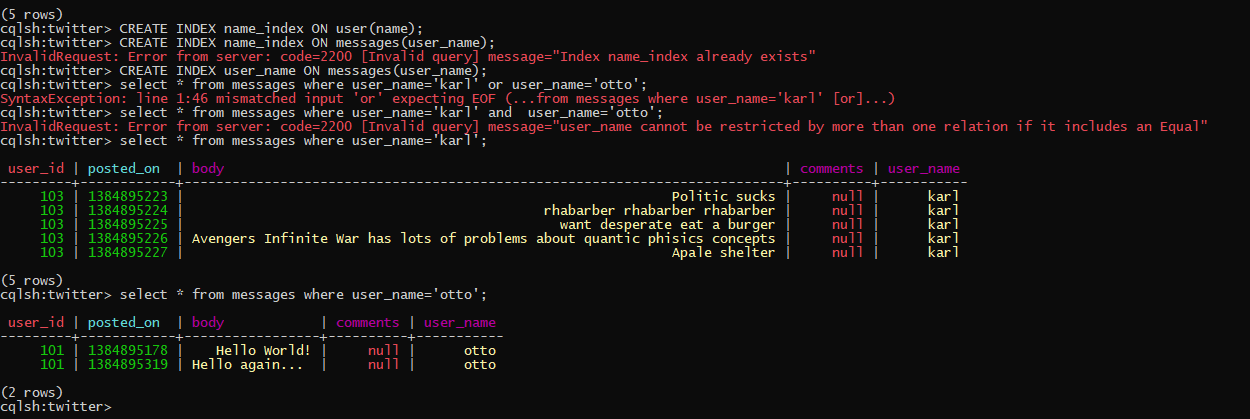
* Observação não é utilizado or no Cassandra, e o IN não é suportado com índices

CREATE INDEX name\_index ON user(name);

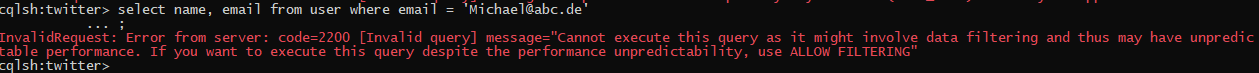
CREATE INDEX name\_index ON messages(user\_name);

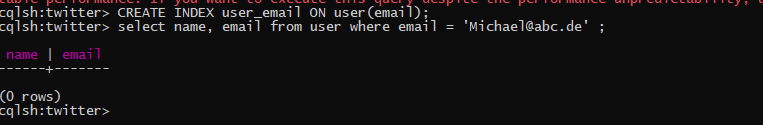
select \* from messages where user\_name='otto';

select \* from messages where user\_name='karl';



* Não permite procurar por email porque não é chave primaria nem tem índice , como solução criei um índice, mas o recomendado é criar uma nova tabela.

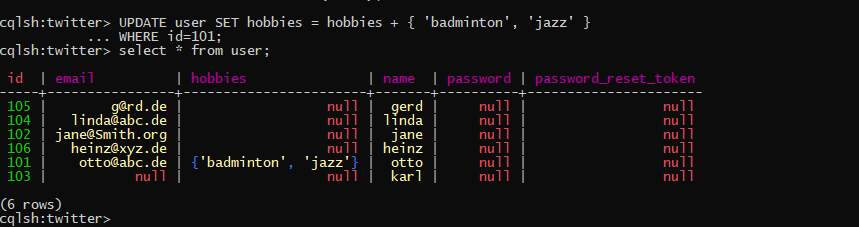


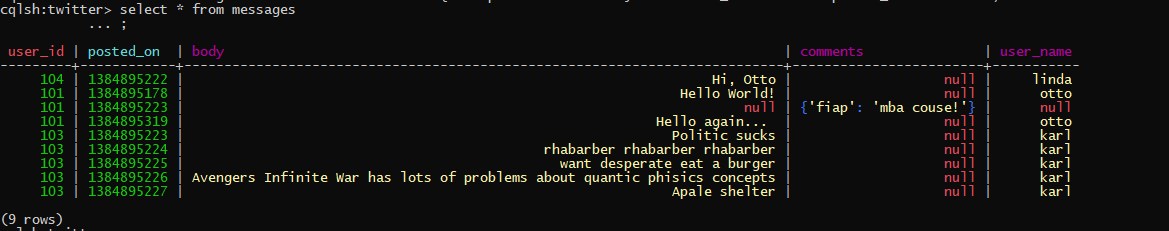


**Update**

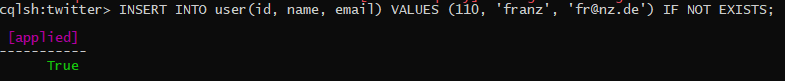
Update não é recomendado no Cassandra e se for necessário utilizar o campo de primary key.



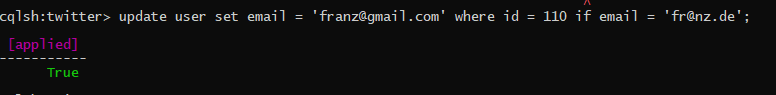




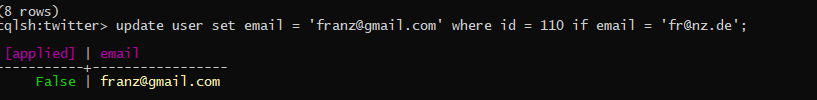
Insere caso não exista a linha, e retorna true se existir



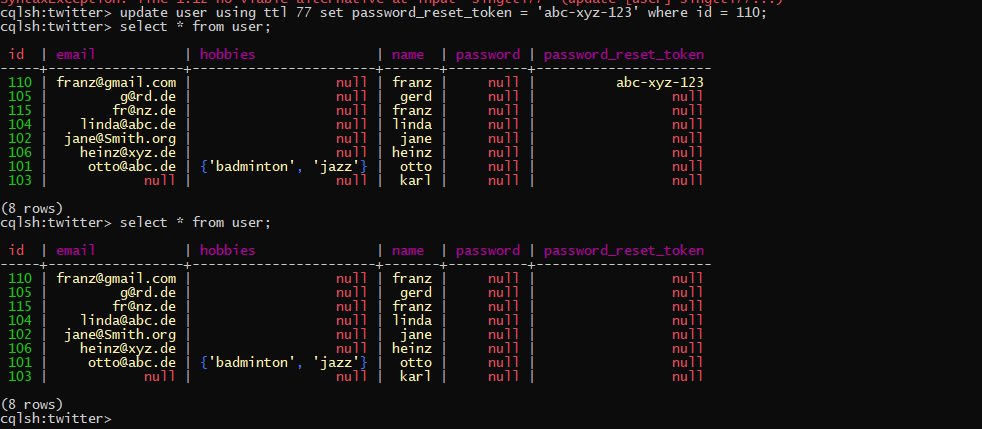
Altera o valor do email, se o email for o especificado no if.



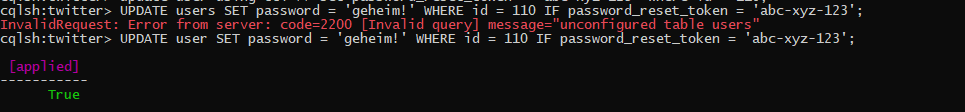
Retorna falso, porque o email não é o especificado no if.



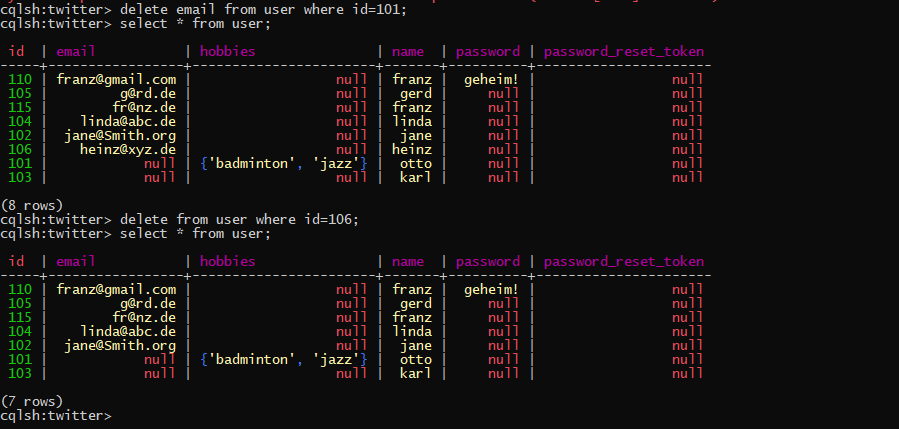
Ele some após 77 segundos

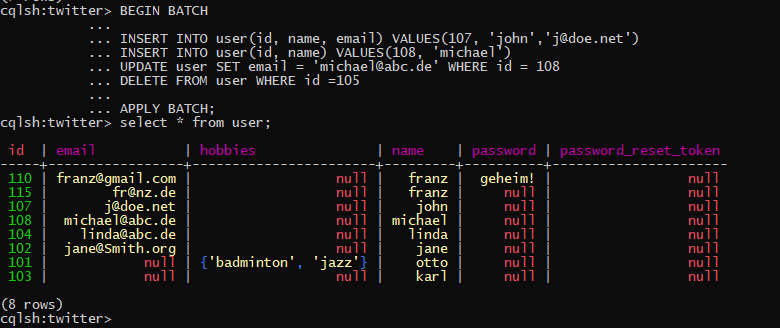


Ele altera se o código for igual ao especificado



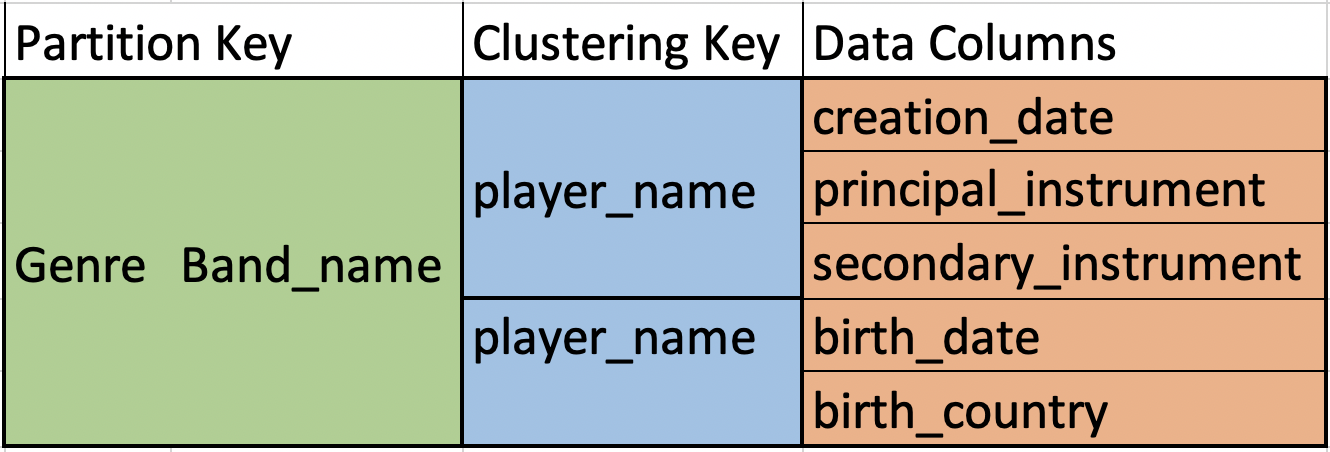
Delete





Conceitos aula que não estão no hands-on

Tabela particionado, entre os primeiros parênteses é a partição, parênteses externos cluster key.



Melhor forma de utilizar o banco é sempre que precisar procurar por outro campo sem ser clustering key, é criar uma nova tabela.

Sempre tem que utilizar a partição no where, e pode adicionar o clustering key.

Uma outra forma sem ser a criação de tabela para utilizar filtros é criar uma view materalizada.

Regras: não pode mudar a partição, so pode tirar a ultima clustering key, caso mais comum de uso é mudar a clustering key