**CRIAÇÃO DO BANCO DE DADOS EM CASSANDRA**

Fase 5 - Exercício Avaliativo

Grupo 51

Nesse passo a passo, iremos cobrir as operacōes necessárias para:

1. [Acessar um serviço de servidor Cassandra para o host do banco de dados](#_g4mzvb1e6nc9)
2. [Criação do banco de dados e *keyspace*](#_vdi0y352c5mt)
3. [Criação da tabela](#_rshm42o2jurt)
4. [Inserção de dados](#_ldi2nsn6h459)
5. [Consulta dos dados](#_nehveyehops)

#### **Acesso ao Serviço de Servidor Cassandra**

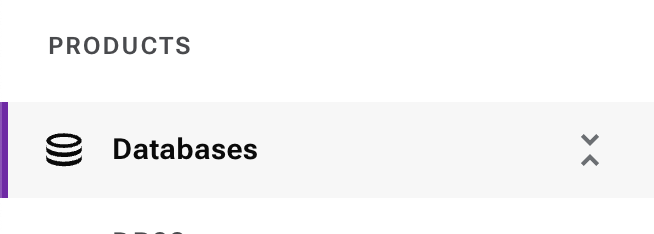
Neste exemplo utilizaremos o DataStax pela sua facilidade de ser acessado e praticidade na criação do espaço necessário.

1. Acesse o DataStax e cria uma conta por [este link](https://auth.cloud.datastax.com/auth/realms/CloudUsers/login-actions/registration?client_id=auth-proxy&tab_id=bx43RGNAKm4&redirect_uri=https://astra.datastax.com/welcome);
2. Caso o site volte para sua página inicial, clique em *Sign In* no canto superior direito e insira as credenciais da conta criada. Caso forneça uma o site já solicite o email e senha da conta, os insira.
3. Clique em *Login*.

#### **Criação do banco de dados e *keyspace***

Neste passo iremos definir o nome do banco de dados, *keyspace* e tabela que os nossos dados serão registrados.

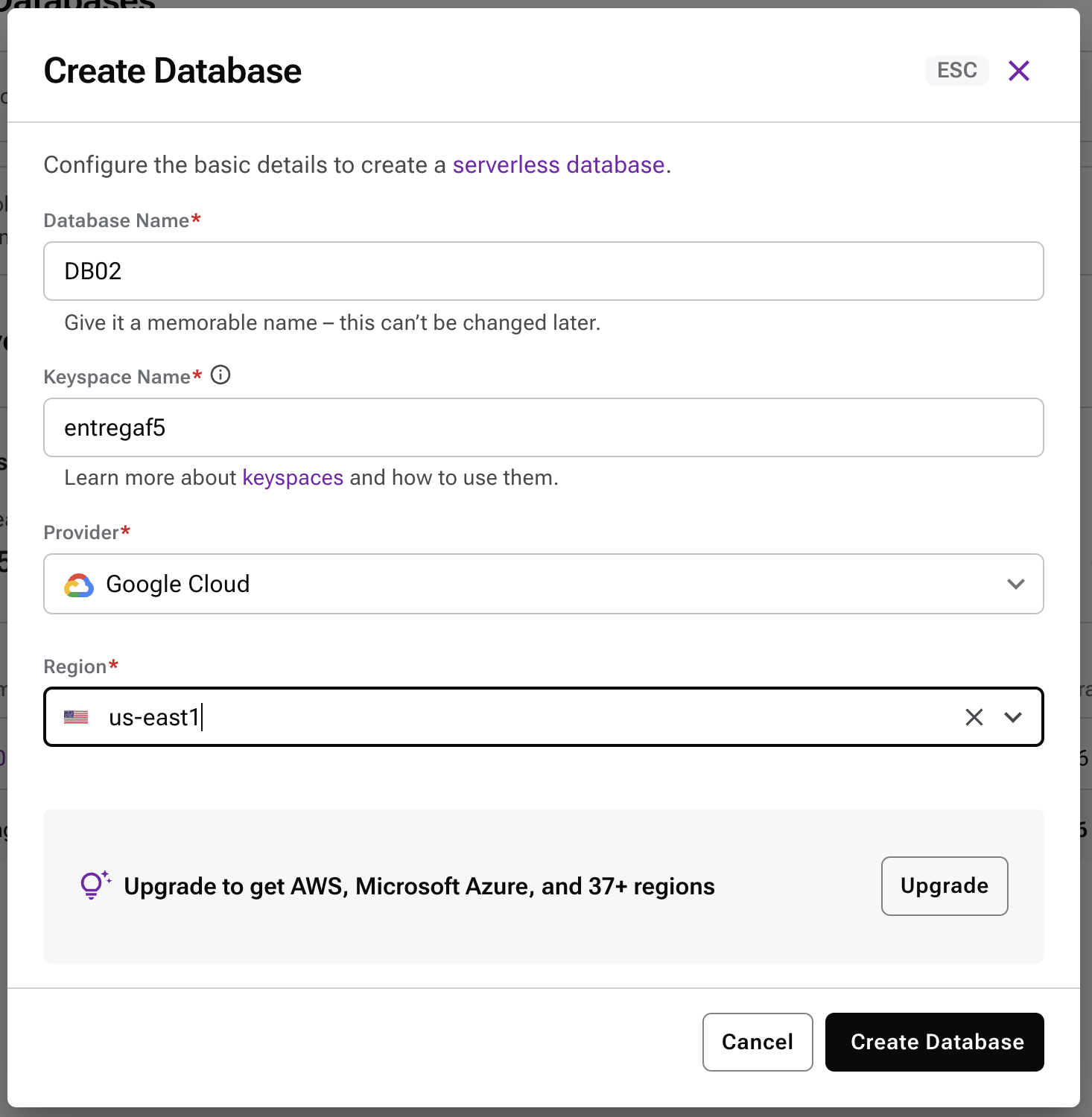
1. Ao acessar o *Astra*, clique em *Databases* no menu esquerdo do site.



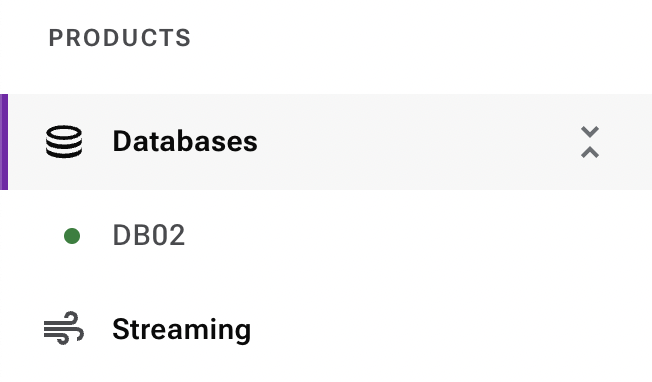
1. A nova página irá trazer todos os bancos de dados já criados e neste momento provavelmente você não possui nenhum listado. Para isso, clique no botão *Create Database* no canto superior direito.



1. Um *pop-up* abrirá na própria página e pedirá as informações de seu primeiro banco de dados. Deveremos informar o nome do banco de dados, *keysapce* e região do servidor disponível. Siga as informações na imagem abaixo.

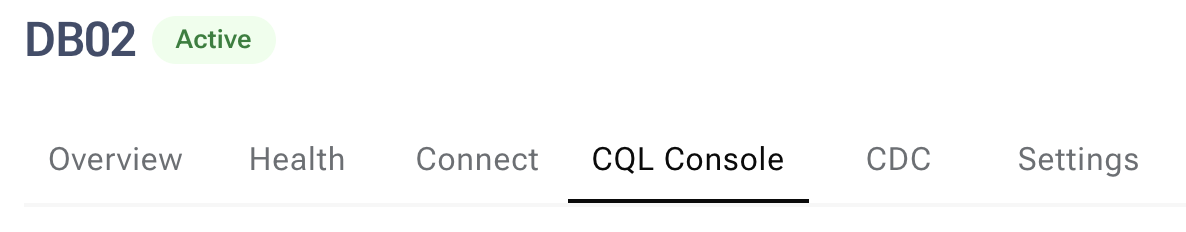


1. Clique em *Create Database*. Ele estará disponível para uso em cerca de 5 a 10 minutos para utilização, listado abaixo de *Databases.*



#### **Criação de uma tabela**

Todos os comandos para gerenciamento do banco de dados, *keyspace* e tabelas são realizados por um terminal *bash* (comando e enter). Você pode acessá-lo clicando no nome do banco de dados e na aba *CQL Console* (imagem abaixo). O Cassandra responde por sua própria linguagem chamada CQL, muito parecida com o SQL. Sua documentação pode ser acessada [neste link](https://docs.datastax.com/en/cql-oss/3.3/cql/cqlIntro.html).



*Botão para o Terminal*

**

*Área do terminal*

Com o terminal aberto, seguiremos estes passos para a criação do banco de dados:

1. Neste momento, precisamos definir a *keyspace* a ser utilizada para nossos comandos. Ela é o *container* que particionará nossos dados, parecido com o conceito de *schema* em SQL e os SGBDS. Definiremos ela digitando o comando ‘**use exerciciof5;**’ e aperte enter.

Perceba que o início da próxima linha mudou para ‘**token@cqlsh:exerciciof5>**’ e indica que estamos na partição correta.

1. Muito parecido ao SQL, criaremos a tabela para esse exercício com o comando abaixo. Nele indicamos as duas chaves primárias da tabela, os dados a serem registrados e seus respectivos tipos, que neste caso serão o código do centro de distribuição e o código do produto oferecido por ele. A função *clustering* indica como os dados devem ser organizados no processo de particionamento, facilitando a busca de dados mais recentes de forma mais eficiente.

Create table estoque\_distribuicao (

id\_centro text,

id\_produto text,

id\_pedido text,

id\_cliente text,

ts\_venda timestamp,

st\_concluido text,

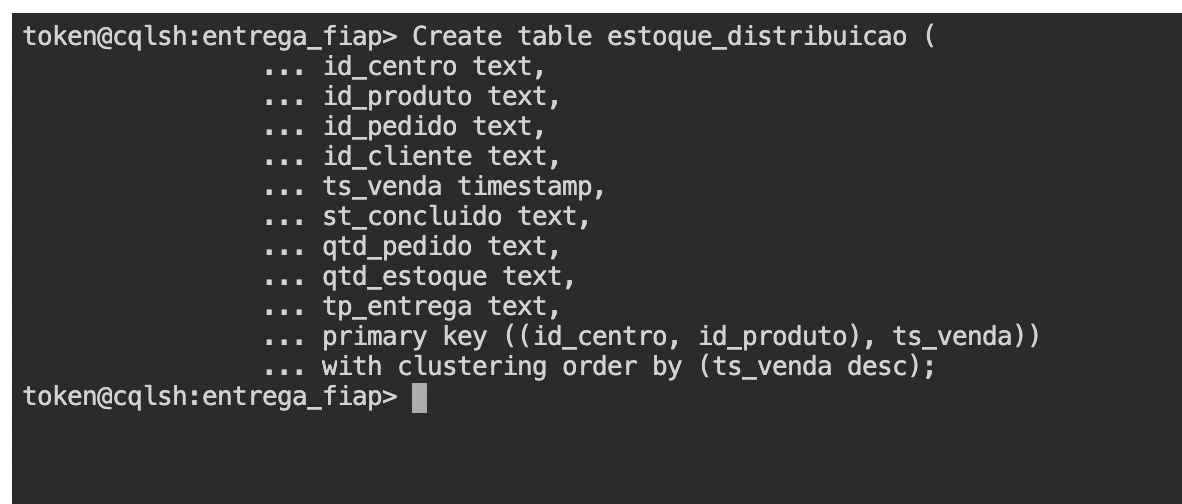
qtd\_pedido text,

qtd\_estoque text,

tp\_entrega text,

primary key ((id\_centro, id\_produto), ts\_venda))

with clustering order by (ts\_venda desc);

Caso não haja erros, o terminal deve retornar a próxima linha de comando para ser escrita (imagem abaixo). Podemos excluir a tabela se for necessário pelo comando ‘drop table estoque\_distribuicao;’.

*Resposta do terminal*

#### **Inserção de Dados**

A inserção de dados é muito parecida com o SQL, funcionando basicamente com pelo comando INPUT (nomes\_das\_colunas) seguidos de VALUES (‘valores a serem inseridos’). Podemos inseri-los de duas formas:

* INPUT Simples

1. Copie e cole o código abaixo para popular o banco de dados criado com dez registros. Perceba a formatação do INSERT e VALUES mencionados anteriormente.

Insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('sp1', 'b9wdpzdrkgbx', 'ped-il1wcoaatg', 'cli-xiql8kb5', '2023-02-24 09:30:00.456', '', '8', '49992', 'brasil');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('abc', 'b9wdpzdrkgbx', 'ped-13izqlx7lh', 'cli-o6392e7e', '2023-02-24 09:30:00.821', 's', '5', '49987', 'abc ja');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('abc', 'b9wdpzdrkgbx', 'ped-ufyugbghcy', 'cli-ys8xuc69', '2023-02-24 09:30:01.186', 's', '4', '49983', 'abc ja');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('sp4', 'b9wdpzdrkgbx', 'ped-hxaud665h1', 'cli-l4or06ur', '2023-02-24 09:30:01.551', 's', '7', '49976', 'agora');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('sp2', 'b9wdpzdrkgbx', 'ped-ml8e5tdou3', 'cli-284ht3ea', '2023-02-24 09:30:01.916', '', '4', '49972', 'via mao');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('sp1', 'b9wdpzdrkgbx', 'ped-x6qk58znqx', 'cli-kiyollva', '2023-02-24 09:30:02.281', 's', '1', '49971', 'via mao');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('abc', 'b9wdpzdrkgbx', 'ped-4hjbt5ic55', 'cli-so1yv6f1', '2023-02-24 09:30:02.646', 's', '4', '49967', 'abc ja');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('sp1', 'b9wdpzdrkgbx', 'ped-4bmbi089oz', 'cli-dptdqinn', '2023-02-24 09:30:03.011', 's', '5', '49962', 'brasil');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('sp1', 'b9wdpzdrkgbx', 'ped-euhom2oje4', 'cli-70ryt0b7', '2023-02-24 09:30:03.376', 's', '5', '49957', 'brasil');

insert into estoque\_distribuicao (id\_centro, id\_produto, id\_pedido, id\_cliente, ts\_venda, st\_concluido, qtd\_pedido, qtd\_estoque, tp\_entrega) values ('sp1', 'b9wdpzdrkgbx', 'ped-z63b4134di', 'cli-onpeoouq', '2023-02-24 09:30:03.741', '', '3', '49954', 'brasil');I

* Carregamento de grande volume de dados

O carregamento de um grande volume de dados pode ser realizado com a função **SOURCE**. É necessário ter criado um arquivo com a extensão *.CQL* anteriormente para utilizá-lo como fonte.

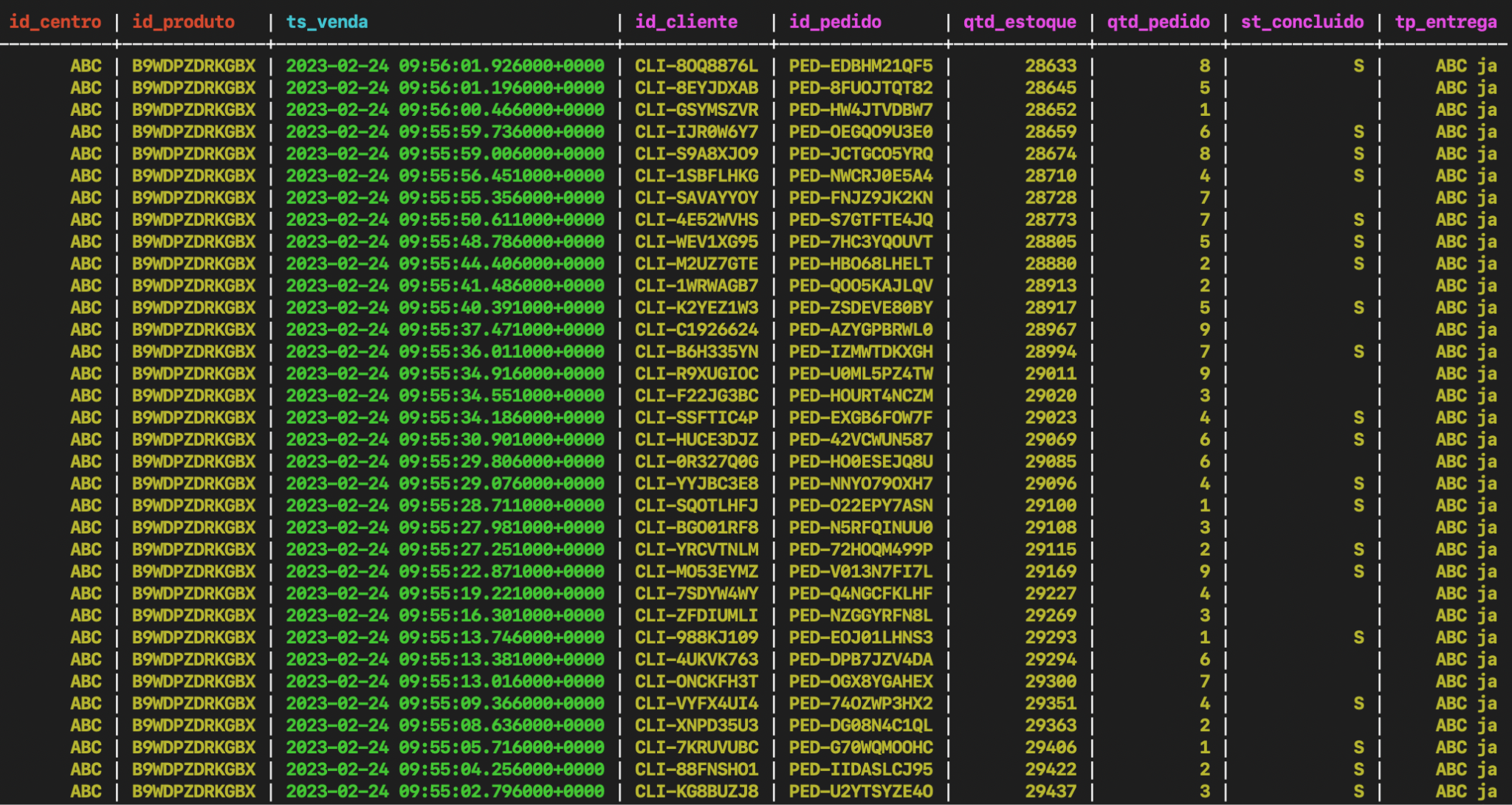
No terminal utilizamos esta função indicando o caminho em que o arquivo se encontra localmente ou em algum site, como o exemplo **source ‘/Users/nome\_do\_usuario/Desktop/Cassandra - Input.cql’**.

Caso tenha interesse em utilizar essa função, disponibilizamos um arquivo com um grande volume de inserts [aqui](https://drive.google.com/file/d/1HIsjsIJsA0u4wvQJtmR7euObfBP6nJY_/view?usp=share_link).

#### **Consulta dos Dados**

As consultas no Cassandra utilizam uma semântica parecida ao SQL, como mencionamos antes, mas é necessário estarmos atentos a algumas necessidades do código.

Pela quantidade dos dados manejados, é essencial levarmos em consideração a performance do código DQL. É interessante adicionarmos as duas chaves primárias da tabela se for necessário fazer pesquisas sobre registros específicos, que neste caso são id\_centro e id\_produto. Dependendo do DQL é também necessário utilizar o *statement* ALLOW FILTERING no final do código. Essa função previne que códigos possam impactar no processamento do banco de dados - use com parcimônia.



*Exemplo de resultado DQL*

Seguimos para o SELECT:

* Select das últimas 100 linhas:

SELECT \* FROM estoque\_distribuicao LIMIT 100;

* Select do centro de distribuição SP1 e item ‘b9wdpzdrkgbx’:

SELECT \* FROM estoque\_distribuicao WHERE id\_centro = 'SP1' AND id\_produto = 'B9WDPZDRKGBX';

* Select do centro de distribuição SP2 e venda maior ou igual ‘2023-02-24 09:54:20’:

SELECT \* FROM estoque\_distribuicao WHERE id\_centro = 'SP2' AND ts\_venda >= '2023-02-24 09:54:20' ALLOW FILTERING;