**Trabalho Cassandra**

**334409 - Charles Guimarães Cavalcante**

Nossa plataforma de *marketplace*, deseja realizar algumas análises e relatórios sobre as vendas, produtos e lojas em tempo real (*Near Real Time*). Dado o grande volume de informações que serão armazenadas, o volume de escrita que será superior ao de leitura e forte necessidade de baixo tempo de resposta, como responsáveis pela arquitetura dessa estrutura, vocês definiram o Cassandra como solução de armazenamento de dados.

**1) Pedidos realizados na última hora por loja:**

Criação da tabela **relatorio1**, com uma partição primária composta pelo id da loja, data e hora. A clusterização mantém os dados em ordem inversa de pedido para retornar rapidamente os últimos pedidos.

CREATE TABLE relatorio1 (

    id\_loja INT,

    data TIMESTAMP,

    hora INT,

    id\_pedido INT,

    cliente\_nome TEXT,

    cidade TEXT,

    uf TEXT,

    id\_produto set<int>,

    valor\_total DECIMAL,

    PRIMARY KEY ((id\_loja, data, hora), id\_pedido)

) WITH CLUSTERING ORDER BY (id\_pedido DESC);

**Índices para atender às premissas:**

CREATE INDEX relatorio1\_uf ON relatorio1(uf);

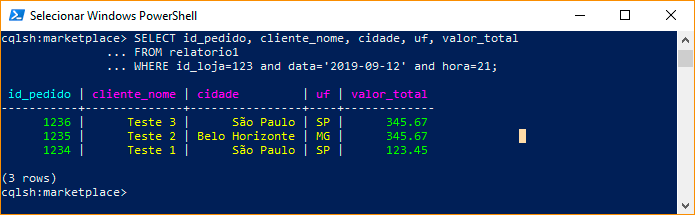
CREATE INDEX relatorio1\_produto ON relatorio1(id\_produto);

**Query para retornar os pedidos da loja 123 no dia '2019-08-29' e hora 21:**

SELECT id\_pedido, cliente\_nome, cidade, uf, valor\_total

FROM relatorio1

WHERE id\_loja=123 and data='2019-09-12' and hora=21;

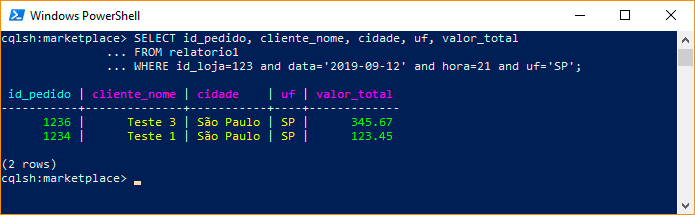
[](https://github.com/charlesartbr/fiap-mba-big-data-data-science/blob/master/Banco%20de%20Dados%20NoSQL/Aula%204%20-%20Cassandra/aula4-trabalho-relatorio1-query1.png)

**Query para retornar os pedidos da loja 123 no dia '2019-08-29', hora 21 e uf 'SP':**

SELECT id\_pedido, cliente\_nome, cidade, uf, valor\_total

FROM relatorio1

WHERE id\_loja=123 and data='2019-09-12' and hora=21 and uf='SP';

[](https://github.com/charlesartbr/fiap-mba-big-data-data-science/blob/master/Banco%20de%20Dados%20NoSQL/Aula%204%20-%20Cassandra/aula4-trabalho-relatorio1-query2.png)

**2) Produtos vendidos na última hora**

Criação da tabela **relatorio2**, com uma partição primária com o id do pedido. A clusterização mantém os dados em ordem inversa de data e hora para retornar rapidamente os últimos pedidos.

CREATE TABLE relatorio2 (  
    status TEXT,   
    data TIMESTAMP,   
    hora INT,   
    id\_pedido INT,   
    cliente\_nome TEXT,   
    cidade TEXT,   
    uf TEXT,   
    id\_produto set<int>,  
    valor\_total DECIMAL,   
    PRIMARY KEY ((status, data, hora), id\_pedido)  
) WITH CLUSTERING ORDER BY (id\_pedido DESC);

**Índices para atender às premissas:**

CREATE INDEX relatorio2\_uf ON relatorio2(uf);

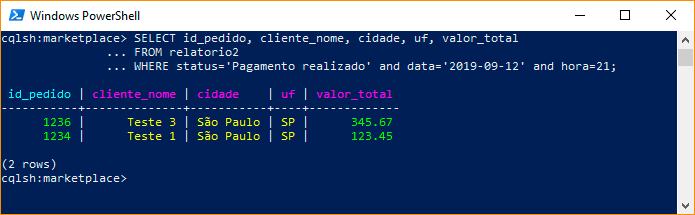
CREATE INDEX relatorio2\_produto ON relatorio2(id\_produto);

**Query para retornar os pedidos com status 'Pagamento realizado' no dia '2019-08-29' e hora 21:**

SELECT id\_pedido, cliente\_nome, cidade, uf, valor\_total

FROM relatorio2

WHERE status='Pagamento realizado' and data='2019-09-12' and hora=21;

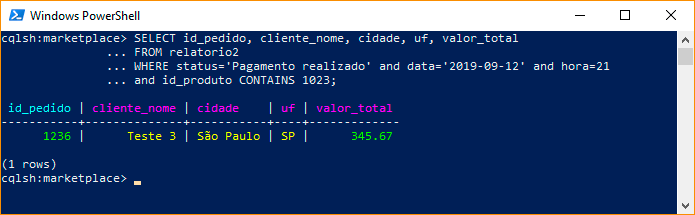
[](https://github.com/charlesartbr/fiap-mba-big-data-data-science/blob/master/Banco%20de%20Dados%20NoSQL/Aula%204%20-%20Cassandra/aula4-trabalho-relatorio2-query1.png)

**Query para retornar os pedidos com status 'Pagamento realizado' no dia '2019-08-29', hora 21 e contém o produto id 1234:**

SELECT id\_pedido, cliente\_nome, cidade, uf, valor\_total

FROM relatorio2

WHERE status='Pagamento realizado' and data='2019-09-12' and hora=21   
 and id\_produto CONTAINS 1023;

[](https://github.com/charlesartbr/fiap-mba-big-data-data-science/blob/master/Banco%20de%20Dados%20NoSQL/Aula%204%20-%20Cassandra/aula4-trabalho-relatorio2-query2.png)

**3) Produtos mais vendidos por Região**

Criação da tabela **relatorio3**, com uma partição primária com o id e nome do produto. A clusterização será pela UF e quantidade.

CREATE TABLE relatorio3 (

    id\_pedido INT,

    id\_produto int,

    produto\_nome TEXT,

    quantidade int,

    valor DECIMAL,

    uf TEXT,

    PRIMARY KEY ((id\_produto, produto\_nome), uf, quantidade)

) WITH CLUSTERING ORDER BY (uf ASC, quantidade DESC);

**Índices para filtrar por UF:**

CREATE INDEX relatorio3\_uf ON relatorio3(uf);

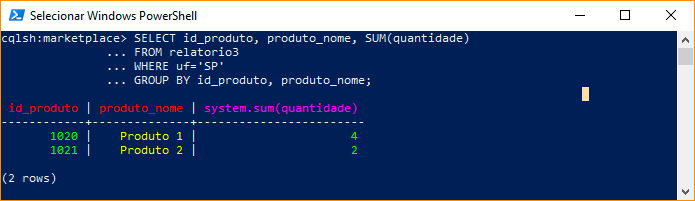
**Query para retornar os produtos mais vendidos na UF 'SP':**

SELECT id\_produto, produto\_nome, SUM(quantidade)

FROM relatorio3

WHERE uf='SP'

GROUP BY id\_produto, produto\_nome;

[](https://github.com/charlesartbr/fiap-mba-big-data-data-science/blob/master/Banco%20de%20Dados%20NoSQL/Aula%204%20-%20Cassandra/aula4-trabalho-relatorio3-query1.png)