O método query() do Pandas permite filtrar dados de um DataFrame utilizando uma **expressão de consulta** semelhante à linguagem SQL. Ele permite que você selecione linhas com base em condições específicas definidas na expressão de consulta, facilitando a filtragem dos dados. Isso é bastante útil ao lidar com grandes conjuntos de dados e condições de filtragem simples e complexas.

Ele aceita um texto (string) contendo a expressão de consulta e retorna um DataFrame contendo apenas as linhas que atendem às condições especificadas. A expressão de consulta pode incluir operadores lógicos como and, or e not, bem como operadores de comparação como ==, <, >, <= e >=.

Vamos ver alguns exemplos?

### **Query para filtrar dados em uma coluna**

Vamos utilizar aqui, em nossos exemplos, a base de dados da Serenatto, que você pode baixar por meio deste [link](https://cdn3.gnarususercontent.com.br/3675-data-science-analise-exploratoria/Projeto/Base_Atividades/serenatto_2sem_2023.csv). Primeiro, vamos trazer o método query() aplicado a filtragem de apenas uma coluna:

import pandas as pd

vendas = pd.read\_csv("serenatto\_2sem\_2023.csv")

vendas.query('produto == "Tiramisù" ').head()

Neste exemplo, estamos filtrando o DataFrame para selecionar apenas as linhas em que o produto seja “Tiramisù”. Para efeito de visualização colocamos o head() para aparecer apenas os 5 primeiros valores.

### **Query para filtrar dados de duas ou mais colunas**

Este método também pode ser utilizado para filtrar dados em duas ou mais colunas, combinando as condições com operadores lógicos.

vendas.query('valor > 10 and metodo\_pagamento != "Dinheiro" ')

Neste outro exemplo, estamos filtrando o DataFrame para selecionar apenas as linhas em que o valor do produto é maior que 10 e o método de pagamento é qualquer um, exceto “Dinheiro”.

### **Query com o operador @**

Além disso, o método query() permite referenciar variáveis externas dentro da expressão de consulta usando o operador @. Sendo bastante útil para criar consultas dinâmicas com base em variáveis definidas previamente.

# Definindo uma variável externa

produtos = ['Café au lait', 'Espresso', 'Cappuccino']

vendas.query('produto in @produtos and metodo\_pagamento == "PIX" ')

Neste exemplo, estamos filtrando o DataFrame para selecionar apenas as linhas em que tenham os produtos passados pela lista produtos e que o método de pagamento seja em PIX.