1. Introdução ao Spark e DataFrames

Objetivo: Entender como carregar e manipular dados.

- Crie um DataFrame a partir de um arquivo CSV com dados simples (por exemplo, nome, idade, cidade).
- Exiba as primeiras 5 linhas usando `.show()`.
- Filtre os dados para mostrar apenas pessoas com idade acima de 25.

```
from pyspark.sql import SparkSession

from pyspark.sql.functions import col

# Criar uma SparkSession

spark = SparkSession.builder.appName("ExercicioSpark").getOrCreate()


# Ler o arquivo CSV e criar um DataFrame

# Certifique-se de ajustar o caminho do arquivo para o local correto

df = spark.read.csv("dados.csv", header=True, inferSchema=True)


# Exibir as primeiras 5 linhas do DataFrame

print("Primeiras 5 linhas do DataFrame:")


df.show(5)

# Filtrar as pessoas com idade acima de 25

print("Pessoas com idade acima de 25:")

df.filter(col("idade") > 25).show()
```

2. Transformações e Ações

Objetivo: Explorar operações básicas.

- Use `.select()` para selecionar apenas as colunas "nome" e "cidade".
- Ordene os dados pela coluna "idade" em ordem decrescente usando `.orderBy()`.
- Converta todas as letras dos nomes para maiúsculas usando `.withColumn()` e funções de Spark SQL.

```
from pyspark.sql.functions import col, upper

# Selecionar apenas as colunas "nome" e "cidade"
df_selecionado = df.select("nome", "cidade")
print("Colunas selecionadas:")
df_selecionado.show()

# Ordenar os dados pela coluna "idade" em ordem decrescente
df_ordenado = df.orderBy(col("idade").desc())
print("Dados ordenados pela idade (decrescente):")
df_ordenado.show()

# Converter os nomes para letras maiúsculas
df_maiusculo = df.withColumn("nome", upper(col("nome")))
print("Nomes em letras maiúsculas:")
df_maiusculo.show()
```

3. Agregações Simples

Objetivo: Aprender a realizar cálculos com dados.

- Calcule a idade média das pessoas no DataFrame usando `.agg()` e funções como avg.
- Conte o número de pessoas por cidade usando `.groupBy()` e `.count()`.

from pyspark.sql.functions import avg

Calcular a idade média das pessoas no DataFrame usando .agg() e avg

idade_media = df.agg(avg("idade").alias("idade_media")) print("Idade média das pessoas:")

idade_media.show()

Contar o número de pessoas por cidade usando .groupBy() e .count()

pessoas_por_cidade = df.groupBy("cidade").count().alias("numero_pessoas")

print("Número de pessoas por cidade:")

pessoas_por_cidade.show()

4. Spark SQL

Objetivo: Trabalhar com consultas SQL dentro do Spark.

- Registre o DataFrame como uma tabela temporária usando `.createOrReplaceTempView()`.
- Escreva uma consulta SQL para selecionar todas as pessoas que moram em uma cidade específica.
- Use uma consulta SQL para calcular a soma das idades.

Registrar o DataFrame como uma tabela temporária

df.createOrReplaceTempView("tabela_pessoas")

Escrever uma consulta SQL para selecionar todas as pessoas que moram em uma cidade específica

cidade_especifica = "São Paulo" consulta_cidade = f""" SELECT * FROM tabela_pessoas WHERE cidade = '{cidade_especifica}' """ pessoas_na_cidade = spark.sql(consulta_cidade) print(f"Pessoas que moram na cidade '{cidade_especifica}':") pessoas_na_cidade.show()

Escrever uma consulta SQL para calcular a soma das idades

consulta_soma_idades = """ SELECT SUM(idade) AS soma_das_idades FROM tabela_pessoas """ soma_idades = spark.sql(consulta_soma_idades) print("Soma das idades:") soma_idades.show()

5. Leitura e Escrita de Dados

Objetivo: Entender como salvar e carregar dados.

- Carregue dados de um arquivo JSON e transforme-os em um DataFrame.
- Salve um DataFrame filtrado como arquivo Parquet.

6. Desafio Final

Objetivo: Combinar tudo o que foi aprendido. Dado um conjunto de dados fictício sobre vendas (por exemplo, id_cliente, valor_compra, data_compra):

- Identifique os clientes com maior valor de compra.
- Agrupe as compras por ano e calcule o total de vendas anuais.

- Salve os resultados em um formato de sua escolha (CSV, JSON, etc.).

Esses exercícios oferecem uma base sólida para trabalhar com Spark. Caso queira adicionar algo mais ou deixar as tarefas mais interativas, posso ajudar! 😊