Aluna: Samantha Pinheiro Gomes

RM: 362698

Turma: 29ABD

Matéria: Distributed Data Processing & Storage

Este notebook realiza a análise do arquivo "results", que contém dados de diversos jogos realizados em várias cidades e países ao redor do mundo, incluindo as datas dos eventos e seus respectivos placares.

Import de bibliotecas

```
✓ 07:52 PM (<1s)                                                                3

from pyspark.sql import SparkSession
import pandas as pd
import requests
from pyspark.sql.types import StructType, StructField, StringType, IntegerType, DoubleType, DateType, BooleanType
from io import StringIO
from pyspark.sql.functions import to_date, col, max
```

```
✓ 07:52 PM (<1s)                                                                4

spark =SparkSession.builder \
    .appName("Analise arquivo results") \
    .getOrCreate()
```

Baixa arquivo armazenado no GitHub

```
## URL do arquivo no GitHub

url = "https://raw.githubusercontent.com/samypinheiro/fiap_distributed_data_processing_-_storage/main/results.csv"

## Armazena os dados
response = requests.get(url)
csv_content = response.content.decode("utf-8")

## Lê com pandas
pandas_df = pd.read_csv(StringIO(csv_content))
```

▸ pandas_df: pandas.core.frame.DataFrame = [date: object, home_teamName: object ... 7 more fields]

Load do dados em um DF do spark

```
## Define schema do dataframe

schema = StructType([
    StructField("date", StringType(), True),
    StructField("home_team", StringType(), True),
    StructField("away_team", StringType(), True),
    StructField("home_score", IntegerType(), True),
    StructField("away_score", IntegerType(), True),
    StructField("tournament", StringType(), True),
    StructField("city", StringType(), True),
    StructField("country", StringType(), True),
    StructField("neutral", BooleanType(), True)
])

## Converte para Spark DataFrame
df_results = spark.createDataFrame(pandas_df, schema=schema)

## Converte o formato da coluna "date"
df_results = df_results.withColumn("date", to_date("date", "yyyy-MM-dd"))
```

▸ df_results: pyspark.sql.connect.dataframe.DataFrame = [date: date, home_team: string ... 7 more fields]

Define dataframe para iniciar a análise

```
df_results_consol = (
    df_results
    .withColumnRenamed("date", "dt_torneio")
    .withColumnRenamed("home_team", "nm_time_casa")
    .withColumnRenamed("away_team", "nm_time_visitante")
    .withColumnRenamed("home_score", "nr_placar_casa")
    .withColumnRenamed("away_score", "nr_placar_visitante")
    .withColumnRenamed("tournament", "nm_torneio")
    .withColumnRenamed("city", "nm_cidade")
    .withColumnRenamed("country", "nm_pais")
).drop("neutral")
```

▸ df_results_consol: pyspark.sql.connect.dataframe.DataFrame = [dt_torneio: date, nm_time_casa: string ... 6 more fields]

Análises realizadas no arquivo

```
## Quantos registros existem na base??

nm_registro = df_results_consol.count()

display(f"Existem {nm_registro} registros na base")
```

▸ See performance (1) Optimize

```
'Existem 40839 registros na base'
```

```
## Quantas equipes únicas mandantes existem na base?

nm_equipes_mandantes = df_results_consol.select("nm_time_casa").distinct().count()

display(f"Existem {nm_equipes_mandantes} equipes mandantes na base")
```

▸ See performance (1) Optimize

```
'Existem 309 equipes mandantes na base'
```

```
## Quantas vezes as equipes mandantes saíram vitoriosas?

nm_vitorias_mandantes = (df_results_consol
                         .filter(col("nr_placar_casa") > col("nr_placar_visitante"))
                         .count())

display(f"As equipes mandantes saíram vitoriosas {nm_vitorias_mandantes} vezes")
```

▸ See performance (1) Optimize

```
'As equipes mandantes saíram vitoriosas 19864 vezes'
```

```
## Quantas vezes as equipes visitantes saíram vitoriosas?

nm_vitorias_visitantes = (df_results_consol
                         .filter(col("nr_placar_visitante") > col("nr_placar_casa"))
                         .count())

display(f"As equipes visitantes saíram vitoriosas {nm_vitorias_visitantes} vezes")
```

▸ See performance (1) Optimize

```
'As equipes visitantes saíram vitoriosas 13544 vezes'
```

```
## Quantas partidas resultaram em empate?
nm_empate = (df_results_consol
                         .filter(col("nr_placar_casa") == col("nr_placar_visitante"))
                         .count())

display(f'{nm_empate} partidas resultaram em empate')
```

▸ See performance (1) Optimize

```
'9431 partidas resultaram em empate'
```

```
## Quantas partidas foram realizadas em cada país?

df_pais = (df_results_consol
           .groupBy("nm_pais")
           .count()
           .orderBy(col("count").desc())
           .withColumnRenamed("count", "qt_partidas"))

df_pais.show()
```

▸ See performance (1) Optimize

▸ df_pais: pyspark.sql.connect.dataframe.DataFrame = [nm_pais: string, qt_partidas: long]

```
|              France|       801|
|             England|       687|
|            Malaysia|       644|
|              Sweden|       637|
|             Germany|       581|
|              Brazil|       529|
|               Spain|       517|
|            Thailand|       483|
|               Italy|       480|
|         Switzerland|       477|
|             Austria|       475|
|[nited Arab Emirates]|       472|
|        South Africa|       470|
|               Qatar|       467|
|         South Korea|       453|
|           Argentina|       449|
|             Hungary|       431|
|               Chile|       405|
|             Belgium|       396|
+--------------------+----------+
only showing top 20 rows
```

```
## Qual país teve mais partidas?

df_partidas = df_pais.orderBy(col("qt_partidas").desc()).limit(1)
df_partidas.show()
```

▸ See performance (1) Optimize

▸ df_partidas: pyspark.sql.connect.dataframe.DataFrame = [nm_pais: string, qt_partidas: long]

```
+-------------+-----------+
|      nm_pais|qt_partidas|
+-------------+-----------+
|United States|       1144|
+-------------+-----------+
```

```
## Qual a partida com maior número de gols?

max_nr_gols = (df_results_consol
               .withColumn("nr_maior_placar", col("nr_placar_casa") + col("nr_placar_visitante"))
               .orderBy(col("nr_maior_placar").desc())
               .select("dt_torneio", "nm_time_casa", "nm_time_visitante", "nr_maior_placar")
               .limit(1)
)
max_nr_gols.show()
```

▸ See performance (1) Optimize

▸ max_nr_gols: pyspark.sql.connect.dataframe.DataFrame = [dt_torneio: date, nm_time_casa: string ... 2 more fields]

```
+----------+------------+-----------------+---------------+
|dt_torneio|nm_time_casa|nm_time_visitante|nr_maior_placar|
+----------+------------+-----------------+---------------+
|2001-04-11|   Australia|   American Samoa|             31|
+----------+------------+-----------------+---------------+
```

```
## Qual a maior goleada?
max_diff_gols = (df_results_consol
               .withColumn("nr_maior_diff_gols", col("nr_placar_casa") - col("nr_placar_visitante"))
               .orderBy(col("nr_maior_diff_gols").desc())
               .select("*")
               .limit(1)
)
max_diff_gols.show()
```

▸ See performance (1) Optimize

▸ max_diff_gols: pyspark.sql.connect.dataframe.DataFrame = [dt_torneio: date, nm_time_casa: string ... 7 more fields]

```
+----------+------------+-----------------+---------------+-----------------+-----------+---------+-----------------+
|dt_torneio|nm_time_casa|nm_time_visitante|nr_placar_casa|nr_placar_visitante|nm_torneio|nm_cidade|nm_pais|nr_maior_diff_gols|
+----------+------------+-----------------+---------------+-----------------+-----------+---------+-----------------+
|2001-04-11|   Australia|   American Samoa|             31|                0|FIFA World Cup qu...|Coffs Harbour|Australia|              31|
+----------+------------+-----------------+---------------+-----------------+-----------+---------+-----------------+
```

```
## Quantos jogos ocorreram no Brasil?

df_brasil = (df_results_consol
             .filter(col("nm_pais") == "Brazil")
             .count()
             )

display(f"{df_brasil} jogos foram realizados no Brasil")
```

▸ See performance (1) Optimize

```
'529 jogos foram realizados no Brasil'
```

+ Code + Text ✦ Assistant

[Shift+Enter] to run and move to next cell
[Ctrl+Shift+P] to open the command palette
[Esc H] to see all keyboard shortcuts