Spark - Mondial

Keto rezultate janë marrë nga ekzekutimi i Spark_Mondial.ipynb - Colab

---

1) Instalimi i spark

```
# Instalimi i spark. Behet vetem nje here.

# Step 1: Install Java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Step 2: Download Spark 3.4.1 (latest confirmed working version)
!wget -q https://archive.apache.org/dist/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz

# Step 3: Extract Spark
!tar -xzf spark-3.4.1-bin-hadoop3.tgz

# Step 4: Install findspark
!pip install -q findspark

# Step 5: Set environment variables
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.4.1-bin-hadoop3"

import findspark
findspark.init()
```

2) Krijimi i Spark Session

```
# Krijon nje spark Session.
  # Gjendja pass mbylljes se session nuk ruhet

from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("BigDataProject") \
    .getOrCreate()

spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.4.1
Master
local[*]
AppName
BigDataProject

3) Ngarkimi i Mondial

```python
#Kjo skripte lexon nga folder mondial ku jane te gjitha tabelat me csv dhe i konverton ne DataFrame (tabela te Spark)
  # df.createOrReplaceTempView i vendos keto dataframe ne memorie
  # kjo na lejon qe te therrasim tabelat sikurse te ishin ne databaze sql


import os

# Required for downloading files
import urllib.request

# GitHub raw base path
base_url = "https://raw.githubusercontent.com/JonKuqi/BigData_Projects/main/Project%203/Resources/Datasets/mondial"
data_path = "mondial"

# Make local folder to save them
os.makedirs(data_path, exist_ok=True)

table_names = [
    "borders", "city", "continent", "country", "desert", "economy", "encompasses", "ethnicgroup",
    "geo_desert", "geo_estuary", "geo_island", "geo_lake", "geo_mountain", "geo_river",
    "geo_sea", "geo_source", "island", "islandin", "ismember", "lake", "language", "located",
    "locatedon", "mergeswith", "mountain", "mountainonisland", "organization", "politics",
    "population", "province", "religion", "river", "sea"
]

# Download CSVs
for table in table_names:
    file_url = f"{base_url}/{table}.csv"
    file_path = os.path.join(data_path, f"{table}.csv")
    print(f"⬇️ Downloading {table}.csv")
    urllib.request.urlretrieve(file_url, file_path)

# Load into Spark
mondial = {}

for table in table_names:
    file_path = os.path.join(data_path, f"{table}.csv")
    df = spark.read.csv(file_path, header=True, inferSchema=True)
    df.createOrReplaceTempView(table)
    mondial[table] = df
    print(f"✅ Loaded '{table}' with {df.count()} rows.")
```

```
⬇️ Downloading mountainonisland.csv
⬇️ Downloading organization.csv
⬇️ Downloading politics.csv
⬇️ Downloading population.csv
⬇️ Downloading province.csv
⬇️ Downloading religion.csv
⬇️ Downloading river.csv
⬇️ Downloading sea.csv
✅ Loaded 'borders' with 320 rows.
✅ Loaded 'city' with 3111 rows.
✅ Loaded 'continent' with 5 rows.
✅ Loaded 'country' with 238 rows.
✅ Loaded 'desert' with 63 rows.
✅ Loaded 'economy' with 238 rows.
✅ Loaded 'encompasses' with 242 rows.
```

4) Query 5 nga faza 1

```
# Query 5: Te listohen te gjithe lumenjte te cilet kalojne neper vendet antare te NATO-s dhe EU-se perjashtuar Suedinte dhe Francen

query5 = spark.sql("""
    SELECT DISTINCT r.Name AS Lumi,
                    c.Name AS Shteti
    FROM   river r
    JOIN   geo_river gr ON r.Name = gr.River      -- lidh lumi-shtet
    JOIN   country   c  ON gr.Country = c.Code
    WHERE  c.Code IN (                            -- anëtar i NATO-s DHE BE-s
              SELECT Country
              FROM   ismember
              WHERE  Organization IN ('NATO','EU')
              GROUP  BY Country
              HAVING COUNT(DISTINCT Organization)=2
           )
      AND  c.Name NOT IN ('Sweden','France')        -- perjashto
    ORDER  BY Lumi, Shteti
""")

query5.show(100, truncate=False)
```

```
+------------+---------------+
|Lumi        |Shteti         |
+------------+---------------+
|Adda        |Italy          |
|Aller       |Germany        |
|Alz         |Germany        |
|Ammer       |Germany        |
|Arno        |Italy          |
|Breg        |Germany        |
|Brigach     |Germany        |
|Donau       |Germany        |
|Douro       |Portugal       |
|Douro       |Spain          |
|Drau        |Italy          |
|Ebro        |Spain          |
|Elbe        |Germany        |
|Etsch       |Italy          |
|Euphrat     |Turkey         |
|Fulda       |Germany        |
|Garonne     |Spain          |
|Guadalquivir|Spain          |
|Guadiana    |Portugal       |
|Guadiana    |Spain          |
|Iller       |Germany        |
|Inn         |Germany        |
|Isar        |Germany        |
|Karasu      |Turkey         |
|Kura        |Turkey         |
```

5) Query 3 nga faza 1

```
# Query 3: Te listohen te gjitha kryeqytetet e shteteve anetare te NATO-s ne te cilat kalon te pakten nje lum

query3 = spark.sql("""
    SELECT DISTINCT c.Capital AS Kryeqyteti
    FROM country c
    INNER JOIN ismember m ON c.Code = m.Country
    INNER JOIN located l ON c.Capital = l.City AND c.Code = l.Country
    WHERE m.Organization = 'NATO'
      AND l.River IS NOT NULL
""")

query3.show(100, truncate=False)
```

```
+----------+
|Kryeqyteti|
+----------+
|Lisbon    |
|London    |
|Paris     |
|Rome      |
+----------+
```