Aula 4 - Spark - Prática

Lucio Monteiro

Spark - Prática 1 - RDD

1) Entrar no site https://databricks.com//
2) Crie um login - Versão gratuita - Community Edition
3) Clique em criar -> Cluster -> Escolher um nome e esperar executar
4) Criar um Notebook - Create Notebook - trocar o nome e a linguagem para Python e selecionar o cluster
5) Create -> Table -> Upload File
6) logDF = sc.textFile("/FileStore/tables/Log_exemplo.txt")
7) logDF.count()
8) logDF.first()
9) palavras = logDF.flatMap(lambda linha: linha.split(" "))
10) palavras.count()
11) palavras.first()
12) minuscula = palavras.map(lambda palavra: palavra.lower())
13) minuscula.first()
14) minuscula_maior2 = minuscula.filter(lambda palavra: len(palavra) >2)
15) minuscula_maior2.count()
16) palavra_1 = minuscula_maior2.map(lambda palavra: (palavra,1))
17) palavra_1.first()
18) palavras_reduce = palavra_1.reduceByKey(lambda chave1, chave2 : chave1 + chave2)

Spark - Prática 1 - RDD

1) palavras_reduce.count()

2) palavras_reduce.collect()

Spark - Prática 2 - Dataframe

1) Entrar no site https://databricks.com//
2) Crie um login - Versão gratuita - Community Edition
3) Clique em criar -> Cluster -> Escolher um nome e esperar executar
4) Criar um Notebook - Create Notebook - trocar o nome e a linguagem para Python e selecionar o cluster
5) Create -> Table -> Upload File
6) jurosDF = spark.read.json("/FileStore/tables/juros_selic.json")
7) jurosDF.printSchema()
8) jurosDF.show(5)
9) jurosDF.count()
10) jurosDF10 = jurosDF.where("valor > 10")
11) jurosDF10.show(10)
12) Gravar no Hive: jurosDF10.write.saveAsTable("tab_juros_selic")
13) Ler do Hive: jurosHiveDF = spark.read.table("tab_juros_selic")
14) jurosHiveDF.printSchema()
15) jurosHiveDF.show(5)
16) jurosHiveDF.write.save("/FileStore/save_juros")
17) display(dbutils.fs.ls("/FileStore/save_juros")

Spark - Prática 3 - Dataframe

1) Entrar no site https://databricks.com//
2) Crie um login - Versão gratuita - Community Edition
3) Clique em criar -> Cluster -> Escolher um nome e esperar executar
4) Criar um Notebook - Create Notebook - trocar o nome e a linguagem para Python e selecionar o cluster
5) Create -> Table -> Upload File
6) alunosDF = spark.read.csv("/FileStore/tables/alunos_1.csv")
7) alunosDF.printSchema()
8) alunosDF = spark.read.option("header","true").csv("/FileStore/tables/alunos_1.csv")
9) alunosDF.printSchema()
10) alunosDF.show(3)
11) alunosDF = spark.read.option("header","true").option("inferSchema","true").csv("/FileStore/tables/alunos_1.csv")
12) alunosDF.write.saveAsTable("tab_alunos")
13) cursosDF = spark.read.option("header","true").option("inferSchema","true").csv("/FileStore/tables/cursos.csv")
14) alunos_cursosDF = alunosDF.join(cursosDF,"id_curso")
15) alunos_cursosDF.show(10)

Spark - Prática 4 - Dataframe

1) Entrar no site https://databricks.com//
2) Crie um login - Versão gratuita - Community Edition
3) Clique em criar -> Cluster -> Escolher um nome e esperar executar
4) Criar um Notebook - Create Notebook - trocar o nome e a linguagem para Python e selecionar o cluster
5) Create -> Table -> Upload File
6) populacaoLA =

   spark.read.option("header","true").option("inferSchema","true").csv("/FileStore/tables/populacaoLA.csv")
7) populacaoLA.printSchema()
8) populacaoLA.show(5)
9) populacaoLA.count()
10) populacaoLA.select("Median Age").groupBy().max().show()
11) populacaoLA.select("Total Females").groupBy().mean().show()

Spark - Prática 5 - Dataframe

1) Entrar no site https://databricks.com//
2) Crie um login - Versão gratuita - Community Edition
3) Clique em criar -> Cluster -> Escolher um nome e esperar executar
4) Criar um Notebook - Create Notebook - trocar o nome e a linguagem para Python e selecionar o cluster
5) Create -> Table -> Upload File (alunos_1.csv)
6) spark.catalog.listDatabases()
7) spark.catalog.listTables("default")
8) spark.catalog.setCurrentDatabase("default")
9) spark.catalog.listTables()
10) spark.catalog.listColumns("tab_alunos_2")
11) spark.read.table("tab_alunos_2").show(10)
12) spark.sql("select * from tab_alunos_2 limit 10").show()
13) alunosHiveDF = spark.read.table("tab_alunos_2")
14) alunosHiveDF.select("id_discente","nome").limit(5).show()
15) spark.sql("select id_discente, nome, ano_ingresso from tab_alunos_2 where ano_ingresso >= 2018").show()
16) alunosHiveDF.select("id_discente","nome","ano_ingresso").where("ano_ingresso >= 2018").show()
17) spark.sql("select id_discente, nome, ano_ingresso from tab_alunos_2 where ano_ingresso >= 2018 order by nome desc").show()
18) spark.sql("select id_discente, nome, ano_ingresso from tab_alunos_2 where ano_ingresso >= 2018").orderBy("nome").show()

Spark - Prática 6 - Dataframe

1) Entrar no site https://databricks.com//
2) Crie um login - Versão gratuita - Community Edition
3) Clique em criar -> Cluster -> Escolher um nome e esperar executar
4) Criar um Notebook - Create Notebook - trocar o nome e a linguagem para Python e selecionar o cluster
5) Create -> Table -> Upload File (alunos_1.csv, cursos.csv)
6) alunos = spark.read.option("header","true").option("inferSchema","true").csv("/FileStore/tables/alunos_1.csv")
7) cursos = spark.read.option("header","true").option("inferSchema","true").csv("/FileStore/tables/cursos.csv")
8) alunos.show(3)
9) cursos.show(3)
10) alunos_join = alunos.join(cursos, on=alunos.id_curso == cursos.id_curso, how='left')
11) alunos_join.show(1)

Obrig.ada