NAME: SHAIK ABDUL KHADAR JILANI ROLLNO: DXC262AB12038

BATCH: DXC-262-Analytics-B12-Azure SUBMISSION: 6-6-2022

COMPANY: DXC TECHNOLOGY

DAY 6

1)Explain what is in-Memory computation in details?

In-memory computation is the technique of running computer calculations entirely in computer memory (e.g., in RAM). This term typically implies large-scale, complex calculations which requires specialized systems software to run the calculations on computers working together in a cluster.

- Processing in memory is one approach to overcoming the von Neumann bottleneck, which is a limitation on throughput caused by the latency inherent in the standard computer architecture.
- In-memory computing primarily relies on keeping data in a server's RAM as a means of processing at faster speeds.
- In-memory computing especially applies to processing problems that require extensive access to data-analytics, reporting or data warehousing and big data applications.

In-memory computation works by eliminating all slow data accesses and relying exclusively on data in RAM. Overall computation performance is greatly improved by remoting the latency commonly seen when accessing the hard dick drives or SSDs. Software running on one or more computers manages the computation as well as the data in memory, and in the case of multiple computers, the software divides into smaller tasks which are distributed out of each computer to run in parallel.

Example Use Cases

One use case of in-memory computing is the modeling of complex systems using a large set of related data points. For example, the modeling of weather data is used to help understand the current trends around events such as storms. Millions or even billions of data points are captured to represent various weather-related measurements, upon which calculations are run to identify the likely patterns the storms will take. With so much data to process, eliminating slow disk accesses will help run the calculations in a much shorter time window.

2)Explain advantages of Spark framework?

Advantages of Spark framework

1. Speed:

When comes to Big Data, processing speed always matters. Spark is wildly popular with data scientists because of its speed. Spark is 100x faster than Hadoop for large scale data processing.

Apache Spark uses in-memory (RAM) computing system whereas Hadoop uses local memory space to store data. Spark can handle multiple petabytes of clustered data of more than 8000 nodes at a time.

2. Ease of Use

Apache Spark carries easy-to-use APIs for operating on large datasets. It offers over 80 high-level operators that make it easy to build parallel apps.

3. Advanced Analytics

Spark not only supports 'MAP' and 'reduce'. It also supports Machine Learning (ML), Graph algorithms, Streaming data, SQL queries, etc.

4. Dynamic in Nature

With Apache Spark, you can easily develop parallel applications. Spark offers you over 80 high-level operators.

5. Multilingual

Apache Spark supports many languages for code writing such as Python, Java, Scala, etc.

6. Powerful

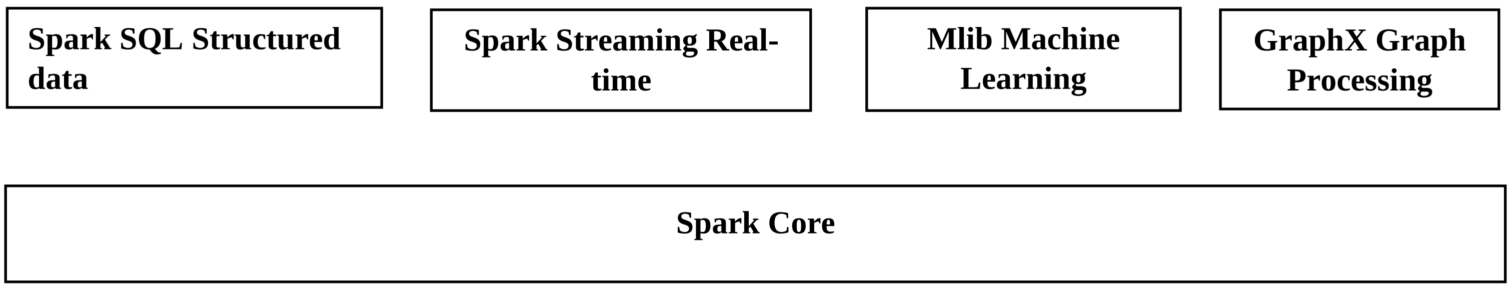
Apache Spark can handle many analytics challenges because of its low-latency in-memory data processing capability. It has well-built libraries for graph analytics algorithms and machine learning.

7. Increased access to Big data

Apache Spark is opening up various opportunities for big data

3)Explain components of Spark with block diagram?

Components of Apache Spark



4) Explain benifits of in-Memory computation ?

Benefits of in-memory computing

-Batter, faster, decision making

-Ability to reduce cost

-Identify competitive opportunities

-Grow revenue

-More efficient application

-reduce risk

-Its best suited for performing real-time analytics and developing and deploying

real-time applications

-In-memory Computing Imperative

Avoid movement of detailed data

Calculate first, then move the results

5) Explain major difference between Hadoop & Spark ?

Hadoop	Spark
1. Hadoop is an open-source framework which uses a MapReduce algorithm	Spark is lightning-fast cluster computing technology, which extends the MapReduce model to efficiently use with more type of computations.
2. Hadoop's MapReduce model reads and writes from a disk, thus slow down the processing speed	Spark reduces the number of read/write cycles to disk and store intermediate data in-memory, hence faster processing speed
3. Hadoop is designed to handle batch processing efficiently	Spark is designed to handle real-time data efficiently
4. Hadoop is a high latency computing framework, which does not have an interactive mode	Spark is a low latency computing and can process data interactively.
5. With Hadoop MapReduce, a developer can only process data in batch mode only	Spark can process real-time data, from real time events like twitter, Facebook
6. Hadoop is a cheaper option available while comparing it in terms of cost.	Spark requires a lot of RAM to run in-memory, thus increasing the cluster and hence cost

6) Explain features of Spark?

Features of Apache Spark

-Fast: It provides high performance for both batch and streaming data, using a state of-the-art DAG scheduler, a query optimizer and a physical execution engine

-Easy to Use: It supports various languages like Java, Python, Scale, Sql, R .It faclilitates to write the application in Java, Scala, Python, R and SQL. It also provides more than 80 high-level operators.

-Supports Various Libraries: It provides a collection of libraries including SQL and DataFrames.MLlib for machine learning, GraphX and Spark Streaming

-Supports Realtime Streaming

-Lightweight: It is a light unified analytics engine which is used for large scale data processing. Runs Everywhere - It can easily run-on Hadoop, Apache Mesos, Kubernetes, Standalone or in he cloud.

7) Write a Py-Spark program to create Dataframe from RDD & explain with screenshots & steps?

Go to

colab.research.google.com

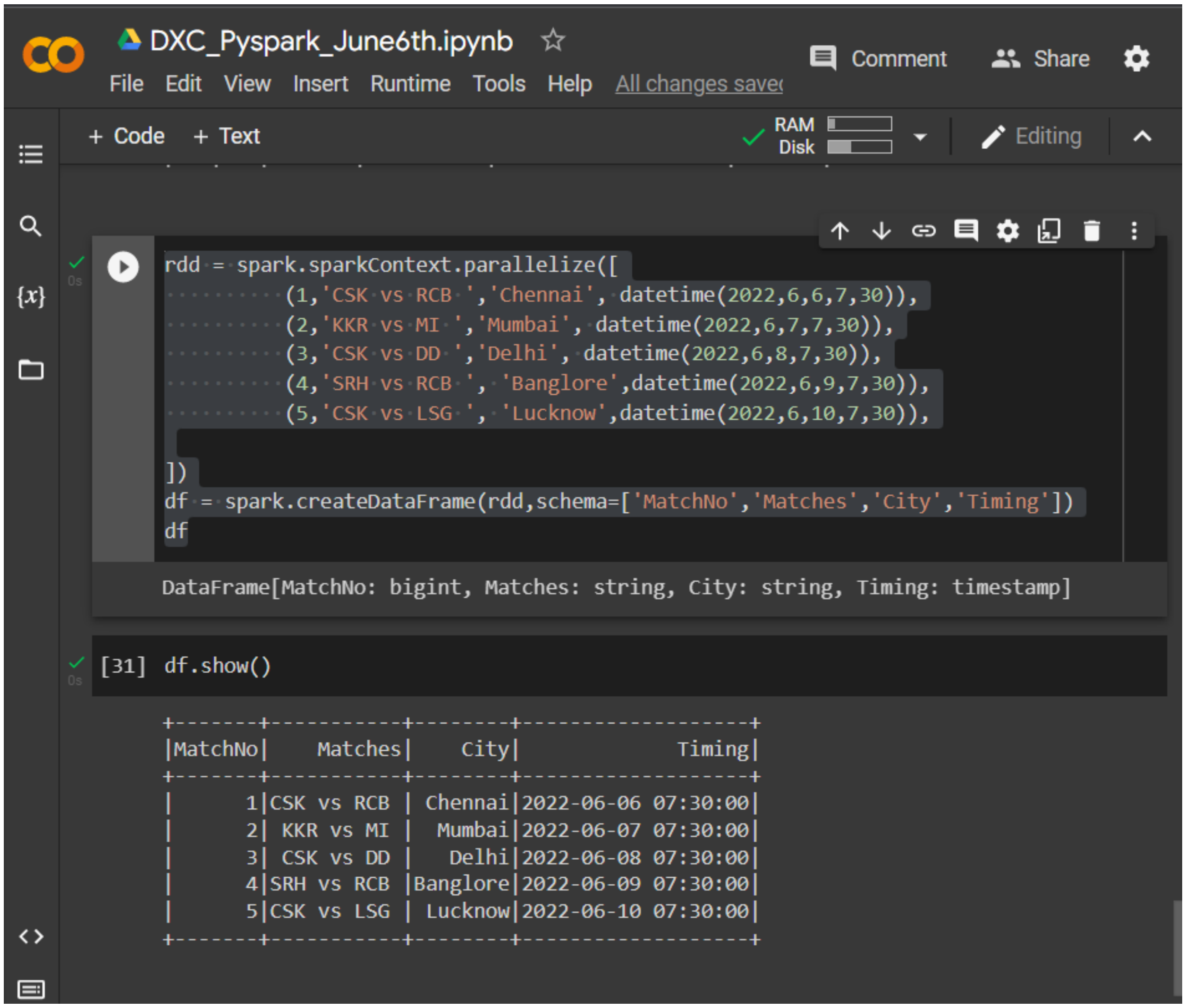
-recent->new notebook

```
>pip install pyspark (and run)
>from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
rdd = spark.sparkContext.parallelize([
        (1,'CSK vs RCB ','Chennai', datetime(2022,6,6,7,30)),
        (2,'KKR vs MI ','Mumbai', datetime(2022,6,7,7,30)),
        (3,'CSK vs DD ','Delhi', datetime(2022,6,8,7,30)),
        (4,'SRH vs RCB ', 'Banglore',datetime(2022,6,9,7,30)),
        (5,'CSK vs LSG ', 'Lucknow',datetime(2022,6,10,7,30)),
```

])

df = spark.createDataFrame(rdd,schema=['MatchNo','Matches','City','Timing'])

df,show() – it will show the entire data frame


DXC_Pyspark_June6th.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing

```
rdd = spark.sparkContext.parallelize([
          (1,'CSK vs RCB ','Chennai', datetime(2022,6,6,7,30)),
          (2,'KKR vs MI ','Mumbai', datetime(2022,6,7,7,30)),
          (3,'CSK vs DD ','Delhi', datetime(2022,6,8,7,30)),
          (4,'SRH vs RCB ', 'Banglore',datetime(2022,6,9,7,30)),
          (5,'CSK vs LSG ', 'Lucknow',datetime(2022,6,10,7,30)),

])
df = spark.createDataFrame(rdd,schema=['MatchNo','Matches','City','Timing'])
df

DataFrame[MatchNo: bigint, Matches: string, City: string, Timing: timestamp]
```

```
[31] df.show()

+-------+-----------+--------+-------------------+
|MatchNo|    Matches|    City|             Timing|
+-------+-----------+--------+-------------------+
|      1|CSK vs RCB | Chennai|2022-06-06 07:30:00|
|      2| KKR vs MI |  Mumbai|2022-06-07 07:30:00|
|      3| CSK vs DD |   Delhi|2022-06-08 07:30:00|
|      4|SRH vs RCB |Banglore|2022-06-09 07:30:00|
|      5|CSK vs LSG | Lucknow|2022-06-10 07:30:00|
+-------+-----------+--------+-------------------+
```

8)Explain what is RDD & why it is needed ?

RDD- Resilient Distributed Dataset:

-It is basis building block of Spark

-The RDD is the spark's core abstraction

-It is a collection of elements, partitioned across the nodes of the cluster so that we can execute various parallel operations on it.

-There are 2 ways to create RDDs:

-Parallelizing an existing data in the driver program

-Referencing a dataset in an external storage system, such as a shared filesystem,HDFS,

HBase or any data source offering a Hadoop InputFormat.

-The key idea of spark is RDD, it supports in-memory processing computation. This means, it stores the state of memory as an object across the jobs and the objects is sharable between those jobs. Data sharing in memory is 10 to 100 times faster than network and Disk.

9)Write a Py-Spark program to make the column in Upper case & explain with screenshots & steps ?

First run this

from pyspark.sql import Column

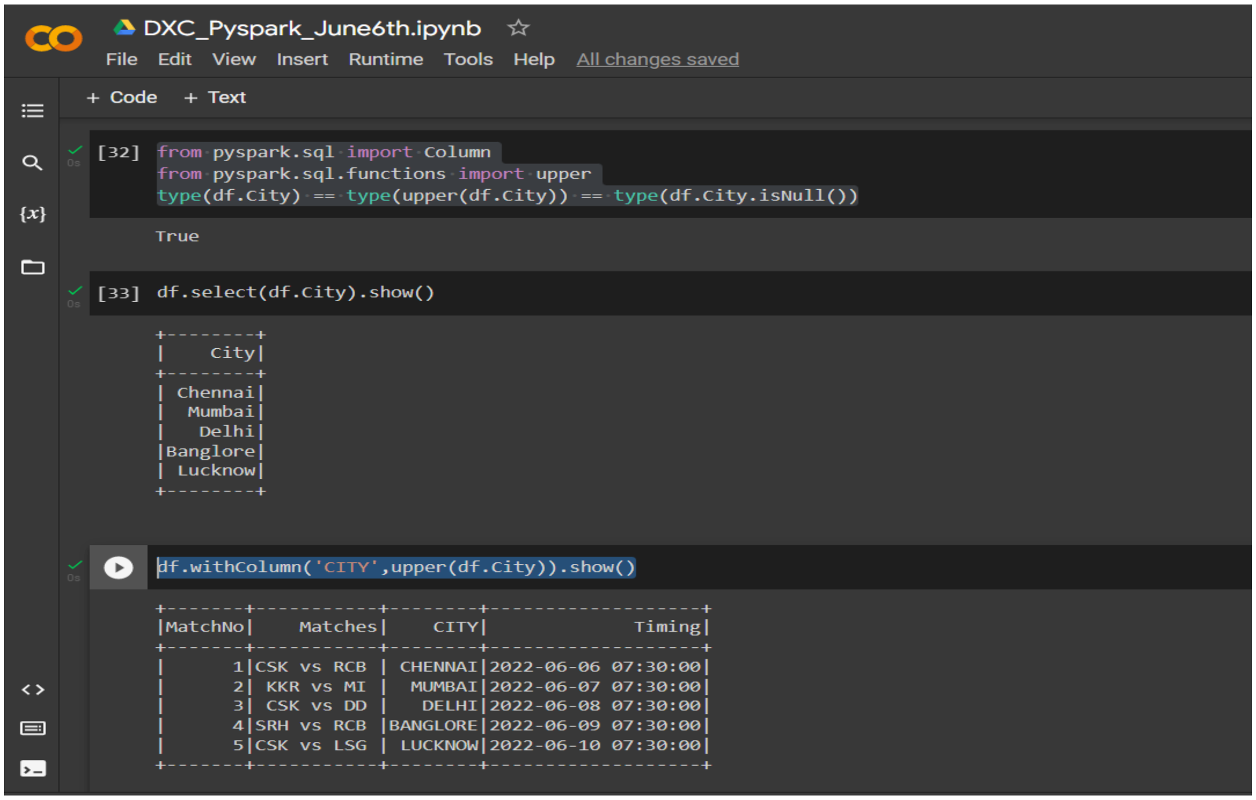
from pyspark.sql.functions import upper

type(df.City) == type(upper(df.City)) == type(df.City.isNull())

df.select(df.City).show() this will show particular column

df.withColumn('CITY',upper(df.City)).show()

-the above command will change the column City into upper case


DXC_Pyspark_June6th.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
[32] from pyspark.sql import Column
     from pyspark.sql.functions import upper
     type(df.City) == type(upper(df.City)) == type(df.City.isNull())

     True

[33] df.select(df.City).show()

     +--------+
     |    City|
     +--------+
     | Chennai|
     |  Mumbai|
     |   Delhi|
     |Banglore|
     | Lucknow|
     +--------+

     df.withColumn('CITY',upper(df.City)).show()

     +-------+-----------+--------+-------------------+
     |MatchNo|    Matches|    CITY|             Timing|
     +-------+-----------+--------+-------------------+
     |      1|CSK vs RCB | CHENNAI|2022-06-06 07:30:00|
     |      2| KKR vs MI |  MUMBAI|2022-06-07 07:30:00|
     |      3| CSK vs DD |   DELHI|2022-06-08 07:30:00|
     |      4|SRH vs RCB |BANGLORE|2022-06-09 07:30:00|
     |      5|CSK vs LSG | LUCKNOW|2022-06-10 07:30:00|
     +-------+-----------+--------+-------------------+
```