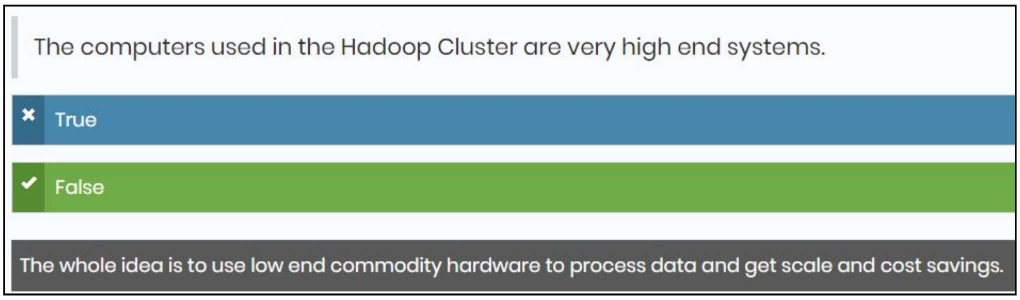
http://www.xpathtester.com/xpath
https://www.xmlvalidation.com/

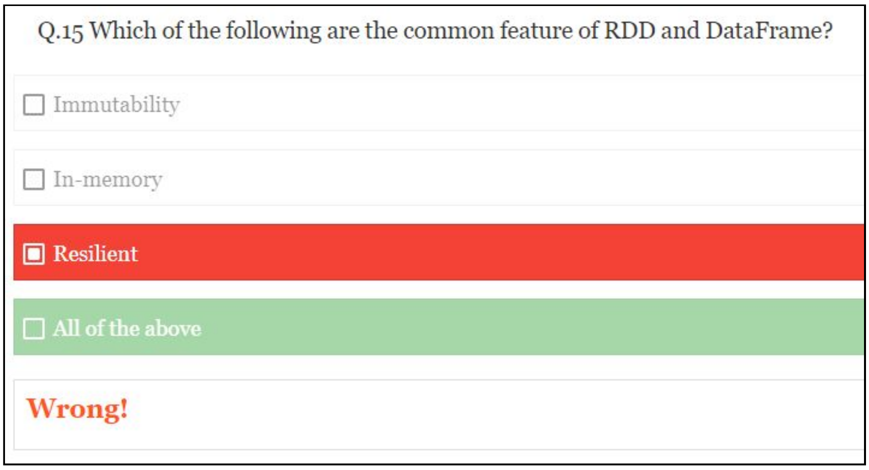
XML is an example of Semi Structured Data.



The computers used in the Hadoop Cluster are very high end systems.

- ✖ True
- ✓ False

The whole idea is to use low end commodity hardware to process data and get scale and cost savings.

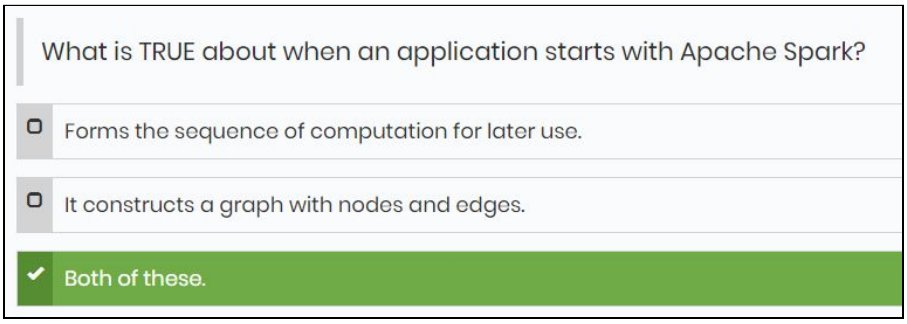


Q.15 Which of the following are the common feature of RDD and DataFrame?

- □ Immutability
- □ In-memory
- ▣ Resilient
- □ All of the above

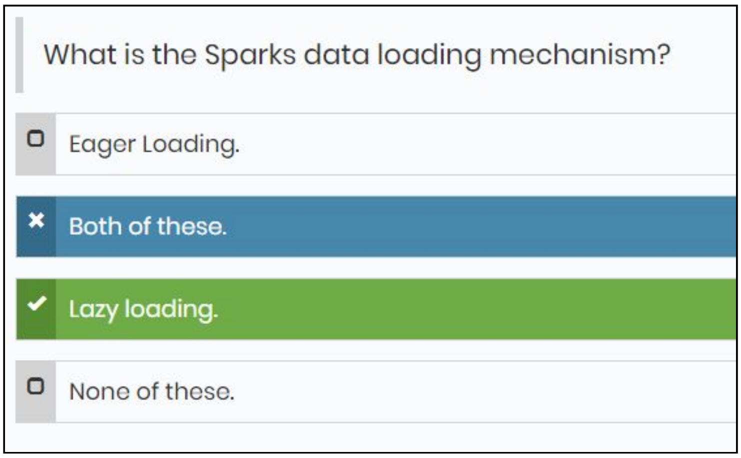
Wrong!

https://data-flair.training/blogs/apache-spark-multiple-choice-questions/



What is TRUE about when an application starts with Apache Spark?

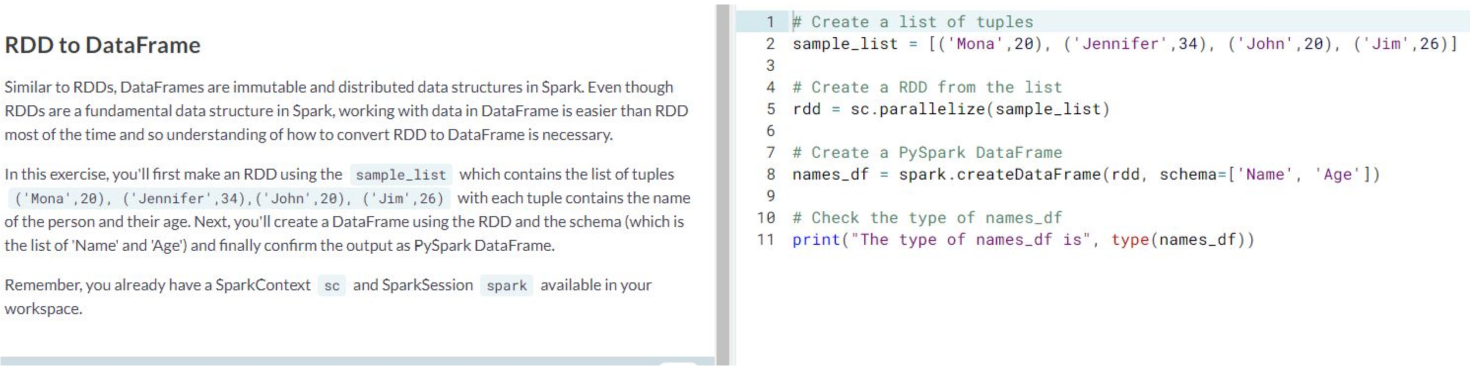
- □ Forms the sequence of computation for later use.
- □ It constructs a graph with nodes and edges.
- ✓ Both of these.



What is the Sparks data loading mechanism?

- □ Eager Loading.
- ✖ Both of these.
- ✓ Lazy loading.
- □ None of these.

https://www.codercrunch.com/quiz/take/1826511478/hands-on-hadoop
https://www.codercrunch.com/quiz/take/796592572/intro-to-hadoop



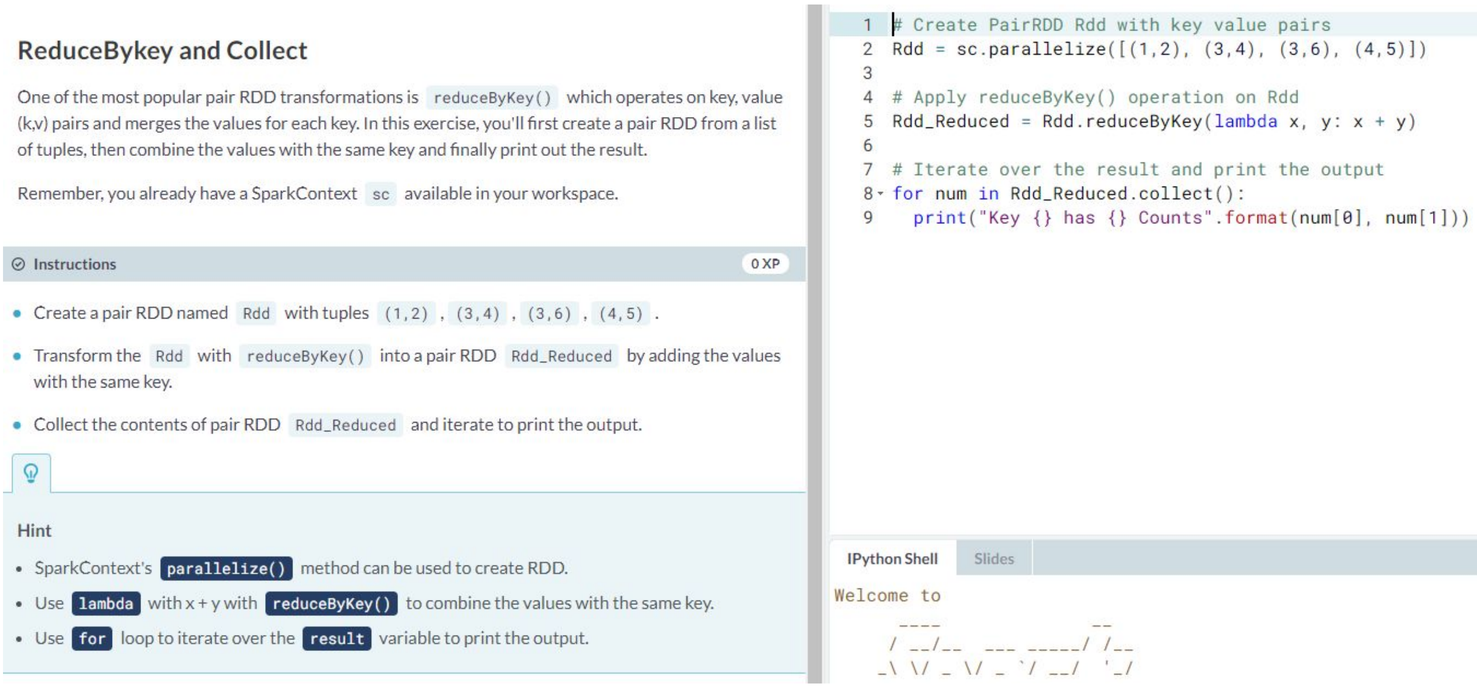
RDD to DataFrame

Similar to RDDs, DataFrames are immutable and distributed data structures in Spark. Even though RDDs are a fundamental data structure in Spark, working with data in DataFrame is easier than RDD most of the time and so understanding of how to convert RDD to DataFrame is necessary.

In this exercise, you'll first make an RDD using the `sample_list` which contains the list of tuples `('Mona',20), ('Jennifer',34),('John',20), ('Jim',26)` with each tuple contains the name of the person and their age. Next, you'll create a DataFrame using the RDD and the schema (which is the list of 'Name' and 'Age') and finally confirm the output as PySpark DataFrame.

Remember, you already have a SparkContext `sc` and SparkSession `spark` available in your workspace.

```
# Create a list of tuples
sample_list = [('Mona',20), ('Jennifer',34), ('John',20), ('Jim',26)]

# Create a RDD from the list
rdd = sc.parallelize(sample_list)

# Create a PySpark DataFrame
names_df = spark.createDataFrame(rdd, schema=['Name', 'Age'])

# Check the type of names_df
print("The type of names_df is", type(names_df))
```



ReduceBykey and Collect

One of the most popular pair RDD transformations is `reduceByKey()` which operates on key, value (k,v) pairs and merges the values for each key. In this exercise, you'll first create a pair RDD from a list of tuples, then combine the values with the same key and finally print out the result.

Remember, you already have a SparkContext `sc` available in your workspace.

Instructions

- Create a pair RDD named `Rdd` with tuples `(1,2)`, `(3,4)`, `(3,6)`, `(4,5)`.
- Transform the `Rdd` with `reduceByKey()` into a pair RDD `Rdd_Reduced` by adding the values with the same key.
- Collect the contents of pair RDD `Rdd_Reduced` and iterate to print the output.

Hint

- SparkContext's `parallelize()` method can be used to create RDD.
- Use `lambda` with x + y with `reduceByKey()` to combine the values with the same key.
- Use `for` loop to iterate over the `result` variable to print the output.

```
# Create PairRDD Rdd with key value pairs
Rdd = sc.parallelize([(1,2), (3,4), (3,6), (4,5)])

# Apply reduceByKey() operation on Rdd
Rdd_Reduced = Rdd.reduceByKey(lambda x, y: x + y)

# Iterate over the result and print the output
for num in Rdd_Reduced.collect():
  print("Key {} has {} Counts".format(num[0], num[1]))
```

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
```

Slayttaki bilgilerin toplu hali:
https://github.com/Shoklan/datacamp/tree/master/BigDataWithPySpark

1. Which of these kinds of data motivated the Map/Reduce framework?

Large number of patient records that are updated immediately after each patient visit.

Large number of customer internet transactions that are often retrieved by a billing id.

Large number of internet documents that need to be indexed for searching by words.

2. What is the organizing data structure for map/reduce programs?

A dictionary of words and their semantic value

A list of identification keys and some value associated with that identifier

A set of indices for a table of data values

3.In map/reduce framework, which of these logistics does Map/Reduce do with the map function?

Gather data distributed across a cluster to the user's computer and run map

Distribute map to cluster nodes, run map on the data partitions at the same time

Distribute map to cluster nodes, run map at one node, wait for it to finish, then run map at the next node, etc,

4. Map/Reduce performs a ‘shuffle’ and grouping. That means it…

Shuffles <key,value> pairs into random bins and then within a bin it groups keys.

Shuffles <key,value> pairs into different partitions according to the key value, and then aggregates all pairs in 1 partition into 1 group.

Shuffles <key,value> pairs into different partitions according to the key value, and sorts within the partitions by key.

5.In the word count example, what is the key?

The line number that contains the word.

The document id that contains the word.

The word itself.

6.Streaming map/reduce allows mappers and reducers to be written in what languages:

java

python

Unix shell commands

R

All of the above

7.The assignment asked you to run with 2 reducers. When you use 2 reducers instead of 1 reducer, what is the difference in global sort order?

With 1 reducer, but not 2 reducers, the word counts are in global sort order by word.

With 2 reducers, but not 1 reducer, the word counts are in global sort order by word.

With 1 reducer or 2 reducers, the word counts are in global sort order by word.

With 1 reducer or 2 reducers, the word counts are NOT in global sort order by word.

8.Apache Spark was developed in order to provide solutions to shortcomings of another project, and eventually replace it. What is the name of this project?

MapReduce

Pig

HDFS

Hadoop

9. Why is Hadoop MapReduce slow for iterative algorithms?
The Java Virtual Machine uses too much memory
Communication is a bottleneck
It needs to read off disk for every iteration
Iterative algorithms do not scale well

10.What is the most important feature of Apache Spark to speedup iterative algorithms?
Resiliency to data loss
Caching datasets in memory
Python interface
Caching datasets on disk

11.Which other Hadoop project can Spark rely to provision and manage the cluster of nodes?
YARN
HDFS
PIG
MAPREDUCE

What does XML stand for? eXtensible Markup Language

There is a way of describing XML data, how?
XML uses a DTD to describe the data
XML uses XSL to describe data
XML uses a description node to describe data

XML's goal is to replace HTML -> false

What is the correct syntax of the declaration which defines the XML version?
<?xml version="1.0"?>
<xml version="1.0" />
<?xml version="1.0" />

What does DTD stand for?
Document Type Definition
Dynamic Type Definition
Direct Type Definition
Do The Dance

Is this a "well formed" XML document? yes

```
<?xml version="1.0"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Is this a "well formed" XML document? No

```
<?xml version="1.0"?>
<to>Tove</to>
<from>Jani</from>
```

```
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

Which statement is true?
All XML elements must be properly closed
All XML documents must have a DTD
All the statements are true
All XML elements must be lower case

Which statement is NOT true?
White-space is not preserved in XML
XML tags are case sensitive
XML documents must have a root tag
XML elements must be properly nested

XML preserves white spaces True

Is this a "well formed" XML document? yes

```
<?xml version="1.0"?>
<note>
<to age="29">Tove</to>
<from>Jani</from>
</note>
```

Is this a "well formed" XML document? No

```
<?xml version="1.0"?>
<note>
<to age=29>Tove</to>
<from>Jani</from>
</note>
```

XML elements cannot be empty
False
True

Which is not a correct name for an XML element?
<1dollar>
<h1>
<Note>
All 3 names are incorrect

Which is not a correct name for an XML element?
<first name>
<age>
All 3 names are incorrect
<NAME>

Which is a correct name for an XML element?
<Name>
<phone number>
<7eleven>
<x mldocument>

XML attribute values must always be enclosed in quotes

True
False

What does XSL stand for?
eXtensible Stylesheet Language
eXtensible Style Listing
eXtra Style Language
eXpandable Style Language

What is a correct way of referring to a stylesheet called "mystyle.xsl" ?
<?xml-stylesheet type="text/xsl" href="mystyle.xsl" ?>
<stylesheet type="text/xsl" href="mystyle.xsl" />
<link type="text/xsl" href="mystyle.xsl" />

For the XML parser to ignore a certain section of your XML document, which syntax is correct?
<![CDATA[Text to be ignored]]>
<PCDATA> Text to be ignored </PCDATA>
<xml:CDATA[Text to be ignored]>
<CDATA> Text to be ignored </CDATA>

Which statement is true?
Attributes must always be present
None of the other two statements are true
Attributes must occur in defined order

What are XML entities used for?
Entities define shortcuts to standard text or special characters
Entities define shortcuts to standard attributes
Entities define shortcuts to standard elements

Which of the following XML fragments is well-formed?

<customer id="3456"><address/><zip code="3456"/></customer>

<customer id=3456><name>John Smith</name></customer>

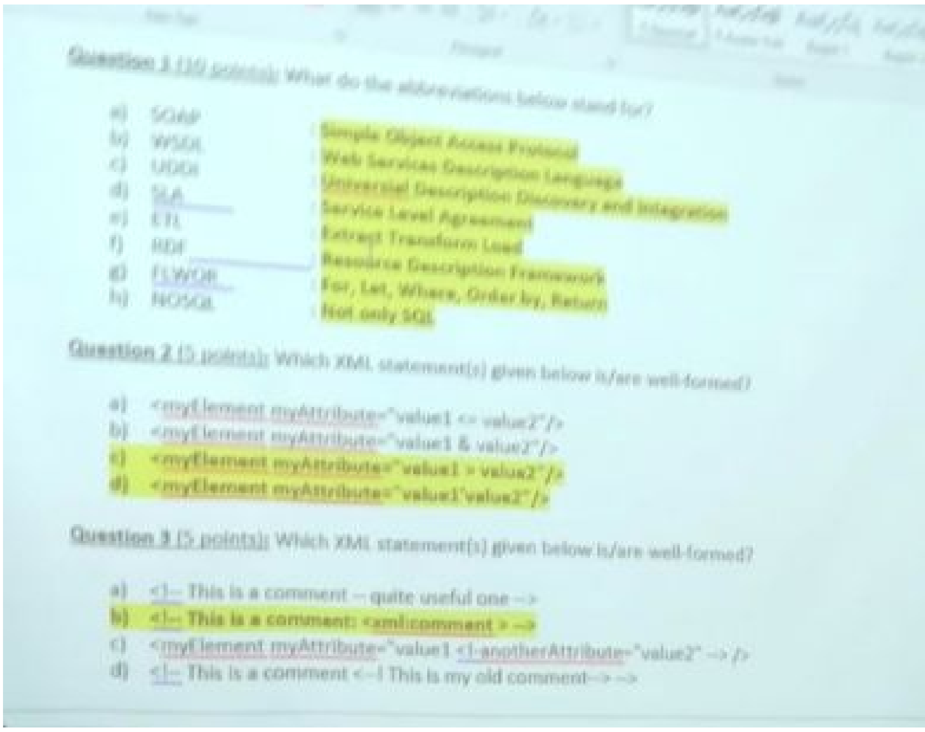
What is an XML instance?

An XML document

An XML attribute

An XML element

Which XML DOM object represents a node in the node tree?
The node object
The document object
The nodeList object



Question 1 (10 points): What do the abbreviations below stand for?

a)	SOAP	Simple Object Access Protocol
b)	WSDL	Web Services Description Language
c)	UDDI	Universial Description Discovery and Integration
d)	SLA	Service Level Agreement
e)	ETL	Extract Transform Load
f)	RDF	Resource Description Framework
g)	FLWOR	For, Let, Where, Order by, Return
h)	NOSQL	Not only SQL

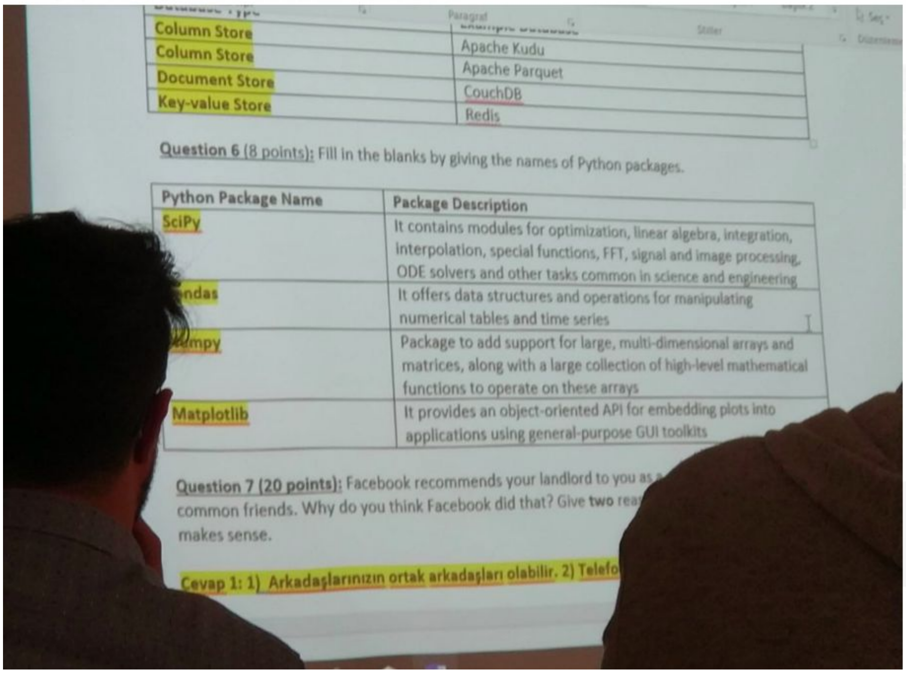
Question 2 (5 points): Which XML statement(s) given below is/are well-formed?

a) <myElement myAttribute="value1 <= value2"/>
b) <myElement myAttribute="value1 & value2"/>
c) <myElement myAttribute="value1 > value2"/>
d) <myElement myAttribute="value1'value2"/>

Question 3 (5 points): Which XML statement(s) given below is/are well-formed?

a) <!-- This is a comment -- quite useful one -->
b) <!-- This is a comment: <xml:comment > -->
c) <myElement myAttribute="value1 <!-anotherAttribute="value2" --> />
d) <!-- This is a comment <--! This is my old comment--> -->

SOAP Simple Object Access Protocol
WSDL Web Services Description Language
UDDI Universal Description, Discovery, and Integration
SLA Service Level Agreement
ETL Extract Transform Load
RDF Resource Description Framework
FLWOR For, Let, Where, Order by, Return
NOSQL Not only SQL



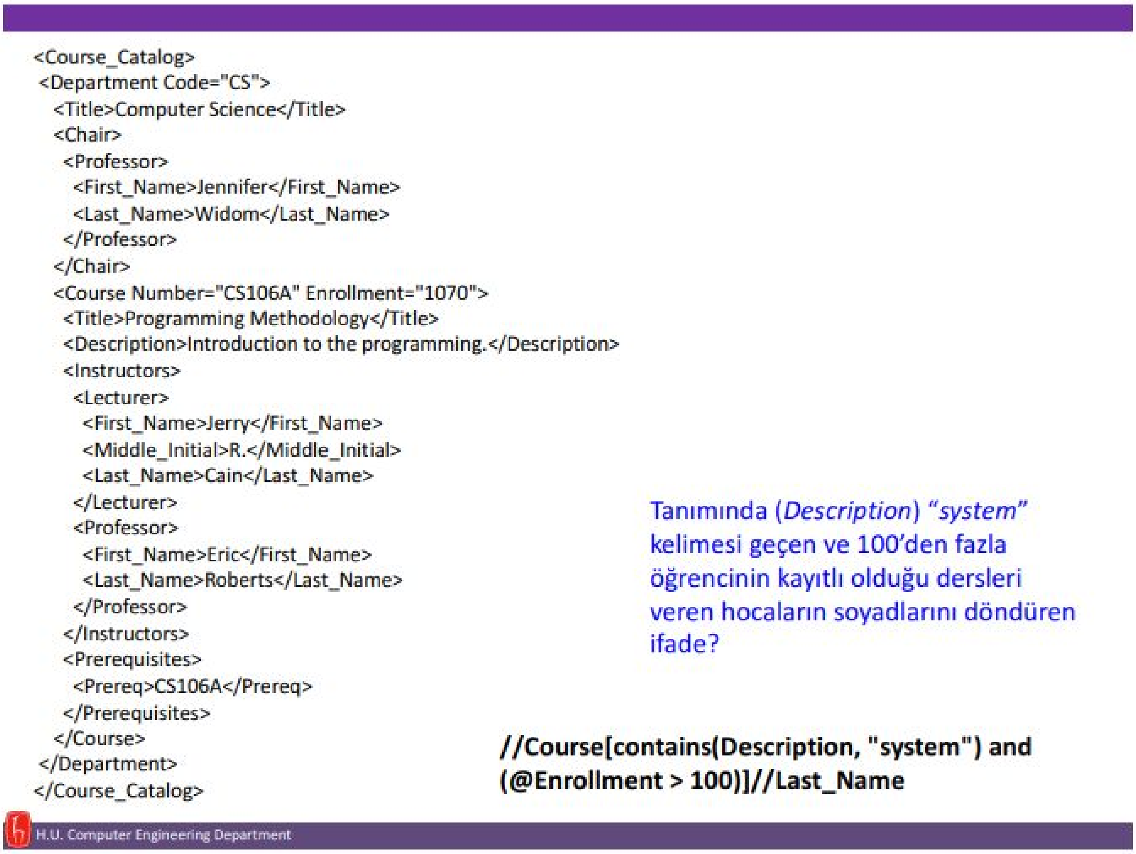
Database Type	Example Database
Column Store	Apache Kudu
Column Store	Apache Parquet
Document Store	CouchDB
Key-value Store	Redis

Question 6 (8 points): Fill in the blanks by giving the names of Python packages.

Python Package Name	Package Description
SciPy	It contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering
ndas	It offers data structures and operations for manipulating numerical tables and time series
mpy	Package to add support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays
Matplotlib	It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

Question 7 (20 points): Facebook recommends your landlord to you as a
common friends. Why do you think Facebook did that? Give **two** rea
makes sense.

Cevap 1: 1) Arkadaşlarınızın ortak arkadaşları olabilir. 2) Telefo



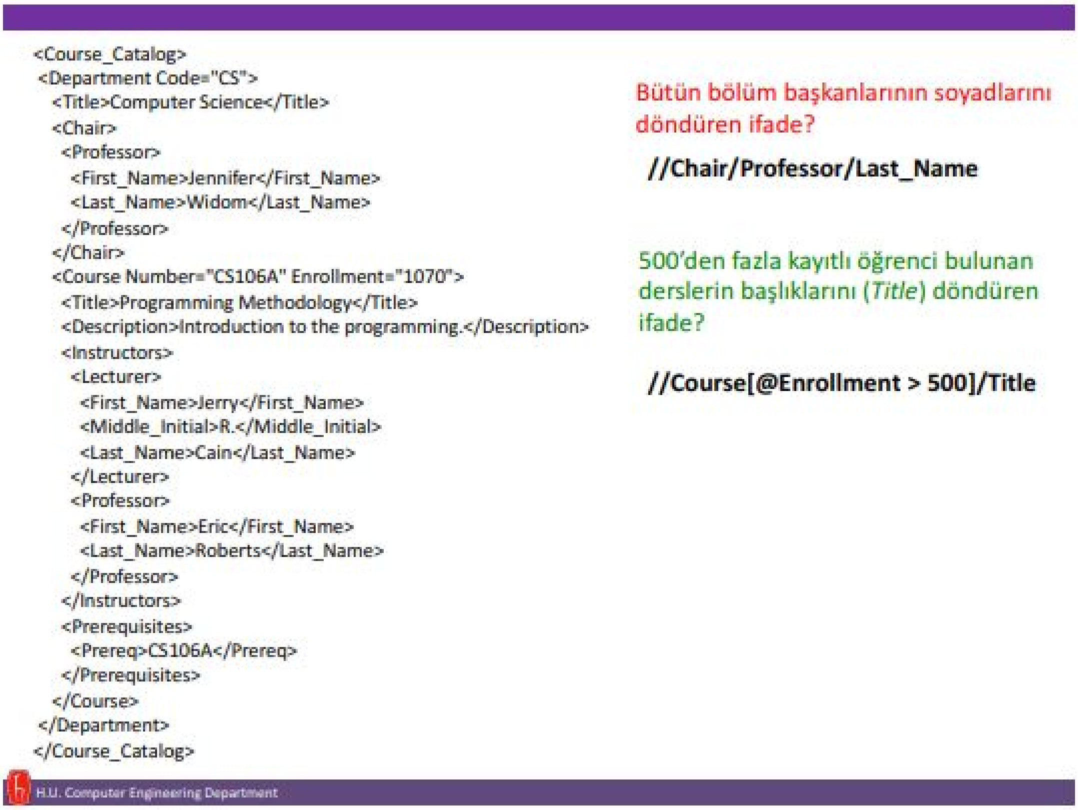
```
<Course_Catalog>
 <Department Code="CS">
   <Title>Computer Science</Title>
   <Chair>
    <Professor>
      <First_Name>Jennifer</First_Name>
      <Last_Name>Widom</Last_Name>
    </Professor>
   </Chair>
   <Course Number="CS106A" Enrollment="1070">
    <Title>Programming Methodology</Title>
    <Description>Introduction to the programming.</Description>
    <Instructors>
      <Lecturer>
        <First_Name>Jerry</First_Name>
        <Middle_Initial>R.</Middle_Initial>
        <Last_Name>Cain</Last_Name>
      </Lecturer>
      <Professor>
        <First_Name>Eric</First_Name>
        <Last_Name>Roberts</Last_Name>
      </Professor>
    </Instructors>
    <Prerequisites>
      <Prereq>CS106A</Prereq>
    </Prerequisites>
   </Course>
 </Department>
</Course_Catalog>
```

Tanımında (*Description*) *“system”* kelimesi geçen ve 100’den fazla öğrencinin kayıtlı olduğu dersleri veren hocaların soyadlarını döndüren ifade?

//Course[contains(Description, "system") and (@Enrollment > 100)]//Last_Name

H.U. Computer Engineering Department

```
<!-- 1. Return all Title elements (of both departments and courses).-->
//Title

<!-- 2. Return last names of all department chairs. →
//Chair//Last_Name
//Chair/*/Last_Name
//Chair/Professor/Last_Name

<!-- 3. Return titles of courses with enrollment greater than 500. -->
//Course[@Enrollment > 500]/Title

<!-- 4. Return titles of departments that have some course that takes "CS106B" as a
prerequisite. →
//Department[Course/Prerequisites/Prereq = 'CS106B']/Title

<!-- 5. Return last names of all professors or lecturers who use a middle initial.
Don't worry about eliminating duplicates. →
//(Professor|Lecturer)[Middle_Initial]/Last_Name

<!-- 6. Return the count of courses that have a cross-listed course (i.e., that
have "Cross-listed" in their description). -->;
count(//Course[contains(Description, 'Cross-listed')])
```



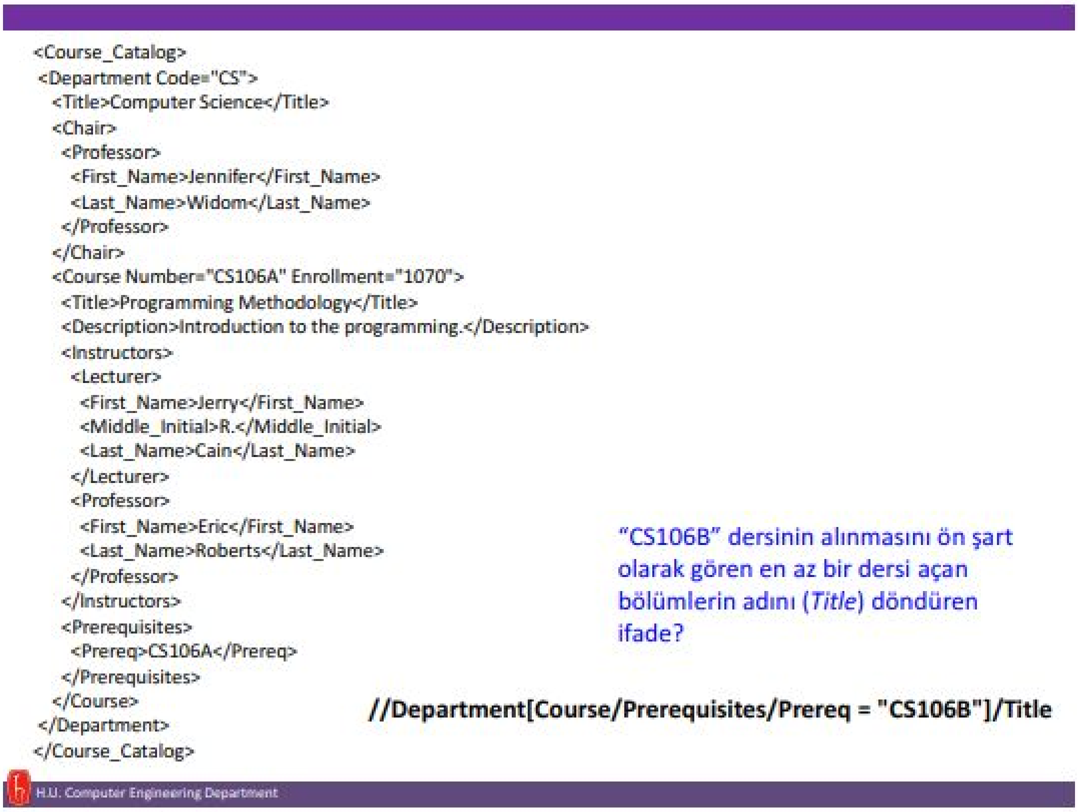
```
<Course_Catalog>
 <Department Code="CS">
   <Title>Computer Science</Title>
   <Chair>
     <Professor>
       <First_Name>Jennifer</First_Name>
       <Last_Name>Widom</Last_Name>
     </Professor>
   </Chair>
   <Course Number="CS106A" Enrollment="1070">
     <Title>Programming Methodology</Title>
     <Description>Introduction to the programming.</Description>
     <Instructors>
       <Lecturer>
         <First_Name>Jerry</First_Name>
         <Middle_Initial>R.</Middle_Initial>
         <Last_Name>Cain</Last_Name>
       </Lecturer>
       <Professor>
         <First_Name>Eric</First_Name>
         <Last_Name>Roberts</Last_Name>
       </Professor>
     </Instructors>
     <Prerequisites>
       <Prereq>CS106A</Prereq>
     </Prerequisites>
   </Course>
 </Department>
</Course_Catalog>
```

Bütün bölüm başkanlarının soyadlarını döndüren ifade?

//Chair/Professor/Last_Name

500'den fazla kayıtlı öğrenci bulunan derslerin başlıklarını (*Title*) döndüren ifade?

//Course[@Enrollment > 500]/Title



```
<Course_Catalog>
 <Department Code="CS">
   <Title>Computer Science</Title>
   <Chair>
     <Professor>
       <First_Name>Jennifer</First_Name>
       <Last_Name>Widom</Last_Name>
     </Professor>
   </Chair>
   <Course Number="CS106A" Enrollment="1070">
     <Title>Programming Methodology</Title>
     <Description>Introduction to the programming.</Description>
     <Instructors>
       <Lecturer>
         <First_Name>Jerry</First_Name>
         <Middle_Initial>R.</Middle_Initial>
         <Last_Name>Cain</Last_Name>
       </Lecturer>
       <Professor>
         <First_Name>Eric</First_Name>
         <Last_Name>Roberts</Last_Name>
       </Professor>
     </Instructors>
     <Prerequisites>
       <Prereq>CS106A</Prereq>
     </Prerequisites>
   </Course>
 </Department>
</Course_Catalog>
```

"CS106B" dersinin alınmasını ön şart olarak gören en az bir dersi açan bölümlerin adını (*Title*) döndüren ifade?

//Department[Course/Prerequisites/Prereq = "CS106B"]/Title

Predicates

Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element

Predicates

//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='en']	Selects all the title elements that have an attribute named lang with a value of 'en'
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

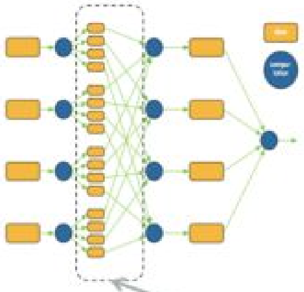
Selecting Nodes

Expression	**Description**
nodename	Selects all child nodes with this name
/	Selects from the root node
//	Selects nodes in the document from the current node down that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

Path Expression	**Result**
bookstore	Selects all the bookstore elements
/bookstore	Selects the root element bookstore **Note:** If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

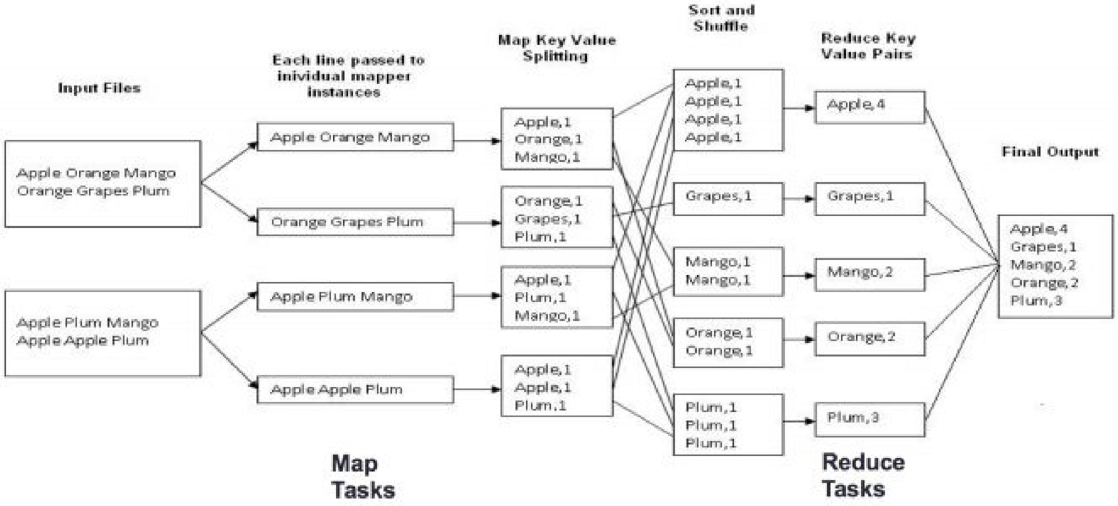
Fault Tolerance

- Intermediate data between mappers and reducers are *materialized* to simple & straightforward fault tolerance
- What if a task fails (map or reduce)?
  - Tasktracker detects the failure
  - Sends message to the jobtracker
  - Jobtracker re-schedules the task
- What if a datanode fails?
  - Both namenode and jobtracker detect the failure
  - All tasks on the failed node are re-scheduled
  - Namenode replicates the users' data to another node
- What if a namenode or jobtracker fails?
  - The entire cluster is down



Intermediate data
(materialized)

Word Count Example



Input Files

Apple Orange Mango
Orange Grapes Plum

Apple Plum Mango
Apple Apple Plum

Each line passed to inividual mapper instances

Apple Orange Mango

Orange Grapes Plum

Apple Plum Mango

Apple Apple Plum

Map Key Value Splitting

Apple,1
Orange,1
Mango,1

Orange,1
Grapes,1
Plum,1

Apple,1
Plum,1
Mango,1

Apple,1
Apple,1
Plum,1

Sort and Shuffle

Apple,1
Apple,1
Apple,1
Apple,1

Grapes,1

Mango,1
Mango,1

Orange,1
Orange,1

Plum,1
Plum,1
Plum,1

Reduce Key Value Pairs

Apple,4

Grapes,1

Mango,2

Orange,2

Plum,3

Final Output

Apple,4
Grapes,1
Mango,2
Orange,2
Plum,3

Map Tasks

Reduce Tasks

The Workflow

- Load data into the Cluster (HDFS writes)
- Analyze the data (MapReduce)
- Store results in the Cluster (HDFS)
- Read the results from the Cluster (HDFS reads)

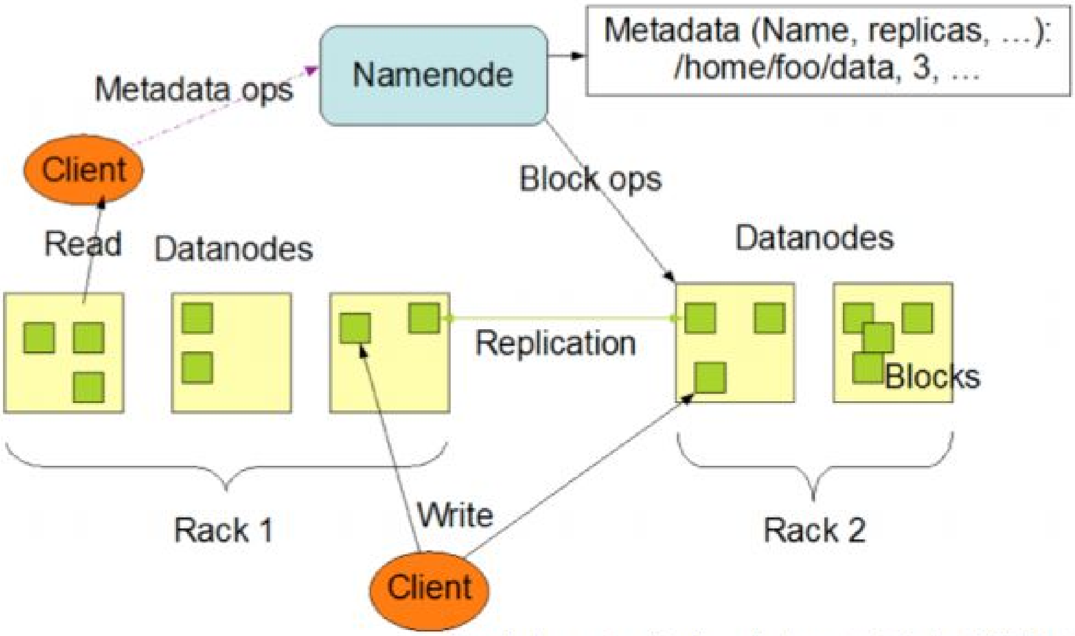
Disk Failures

- Each DataNode sends a Heartbeat message to the NameNode periodically.
- The NameNode marks DataNodes without recent Heartbeats as dead and does not forward any new IO requests to them.
- Any data that was registered to a dead DataNode is not available to HDFS any more.
- DataNode death may cause the replication factor of some blocks to fall below their specified value.
- The NameNode constantly tracks which blocks need to be replicated and initiates replication whenever necessary.

Block Placement

- Files are split into fixed sized blocks and stored on data nodes (Default 64MB)
- Data blocks are replicated for fault tolerance and fast access (Default is 3)
- Where to put a given block by default?
  - **First copy** is written to the node creating the file (write affinity)
  - **Second copy** is written to a data node within the same rack
  - **Third copy** is written to a data node in a different rack

HDFS Architecture

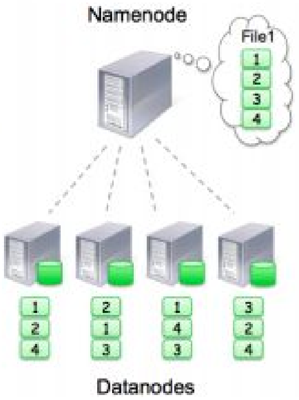


http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

HDFS Pursues a master-slave Model

- NameNode
  - Executes file system namespace operations like opening, closing, and renaming files and directories.
  - Determines the mapping of blocks to DataNodes.

- DataNodes
  - Manage attached storage.
  - Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes.
  - The DataNodes are responsible for serving read and write requests from the file system's clients.
  - The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.



Hadoop Components

- Hadoop Distributed file system (HDFS)
  - Single namespace for entire cluster
  - Replicates data for fault-tolerance

- MapReduce framework
  - Executes user jobs specified as “map” and “reduce” functions
  - Manages work distribution & fault-tolerance

Hadoop Ecosystem

Ambari™: A web-based tool for provisioning, managing, and monitoring Apache Hadoop.

Avro™: A data serialization system.

Cassandra™: A scalable multi-master database with no single points of failure.

Chukwa™: A data collection system for managing large distributed systems.

HBase™: A scalable, distributed database that supports structured data storage for large tables.

Hive™: A data warehouse infrastructure that provides data summarization and ad hoc querying.

Mahout™: A Scalable machine learning and data mining library.

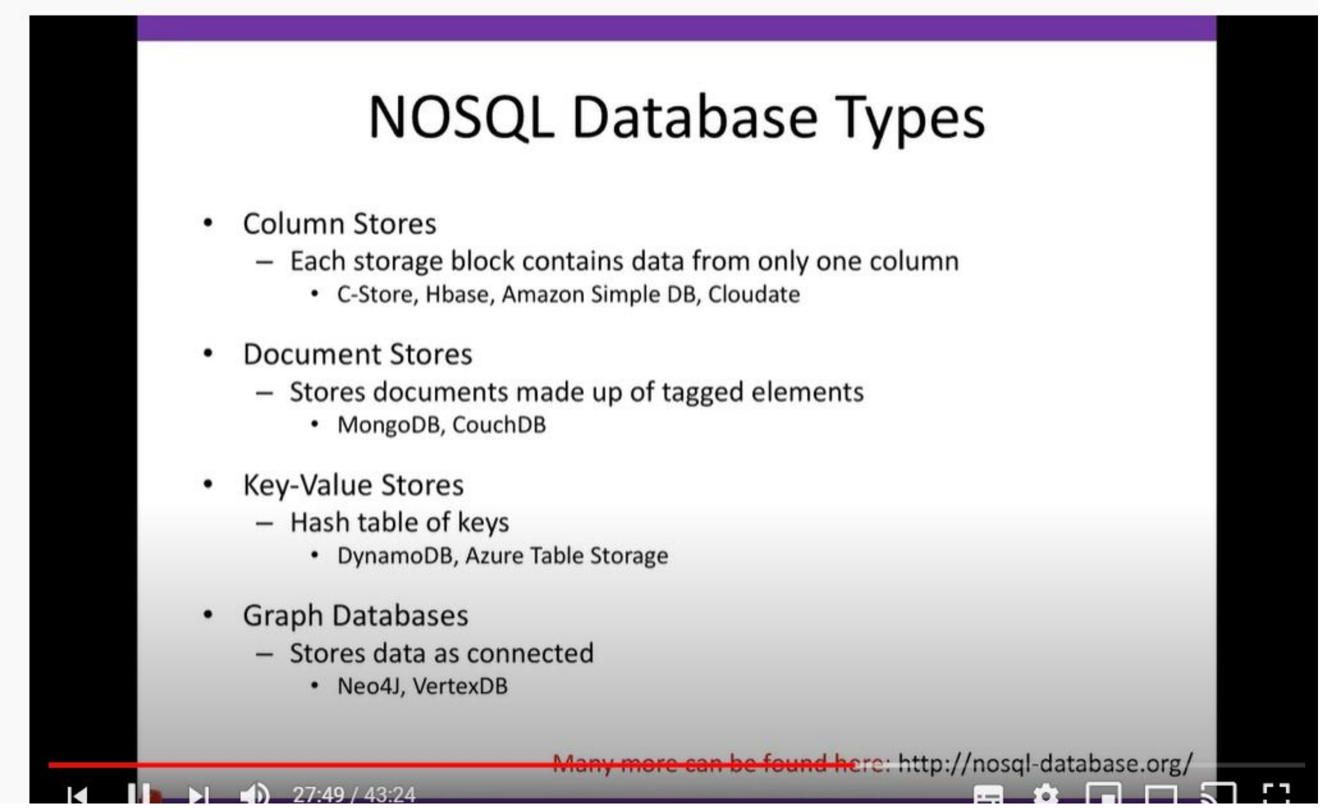
Pig™: A high-level data-flow language and execution framework for parallel computation.

ZooKeeper™: A high-performance coordination service for distributed applications.

Hadoop

- Based on work done by Google in the early 2000s
  - "The Google File System" in 2003
  - "MapReduce: Simplified Data Processing on Large Clusters" in 2004

- The core idea was to distribute the data as it is initially stored
  - Each node can then perform computation on the data it stores without moving the data for the initial processing



NOSQL Database Types

- Column Stores
  - Each storage block contains data from only one column
    - C-Store, Hbase, Amazon Simple DB, Cloudate
- Document Stores
  - Stores documents made up of tagged elements
    - MongoDB, CouchDB
- Key-Value Stores
  - Hash table of keys
    - DynamoDB, Azure Table Storage
- Graph Databases
  - Stores data as connected
    - Neo4J, VertexDB

Many more can be found here: http://nosql-database.org/

1.How can you create an RDD? Mark all that apply

- Reading from a local file available both on the driver and on the workers
- Reading from HDFS
- Apply a transformation to an existing RDD

2.How does Spark make RDDs resilient in case a partition is lost?

- Tracks the history of each partition and reruns what is needed to restore it

3.Which of the following sentences about flatMap and map are true?

- flatMap accepts a function that returns multiple elements, those elements are then flattened out into a continuous RDD.
- map transforms elements with a 1 to 1 relationship, 1 input - 1 output

4.Check all wide transformations

- Repartition, even if it triggers a shuffle, can improve performance of your pipeline by balancing the data distribution after a heavy filtering operation

1.What does SQOOP stand for?

- SQL to Hadoop

2.What is not part of the basic Hadoop Stack 'Zoo'?

- Horse

3.What is considered to be part of the Apache Basic Hadoop Modules?

- HDFS

4.What are the two major components of the MapReduce layer?

- TaskManager
- JobTracker

5.What does HDFS stand for?

- Hadoop Distributed File System

6.What are the two majority types of nodes in HDFS?

- DataNode
- NameNode

7.What is Yarn used as an alternative to in Hadoop 2.0 and higher versions of Hadoop?

- MapReduce

8.Could you run an existing MapReduce application using Yarn?

- Yes

9.What are the two basic layers comprising the Hadoop Architecture?

- MapReduce and HDFS

10.What are Hadoop advantages over a traditional platform?

- Scalability
- Reliability
- Flexibility
- Cost

1.Apache Spark was developed in order to provide solutions to shortcomings of another project, and eventually replace it. What is the name of this project?

- MapReduce

2.Why is Hadoop MapReduce slow for iterative algorithms?

- It needs to read off disk for every iteration

3.What is the most important feature of Apache Spark to speedup iterative algorithms?

- Caching datasets in memory

4.Which other Hadoop project can Spark rely to provision and manage the cluster of nodes?

- YARN

5.When Spark reads data out of HDFS, what is the process that interfaces directly with HDFS?

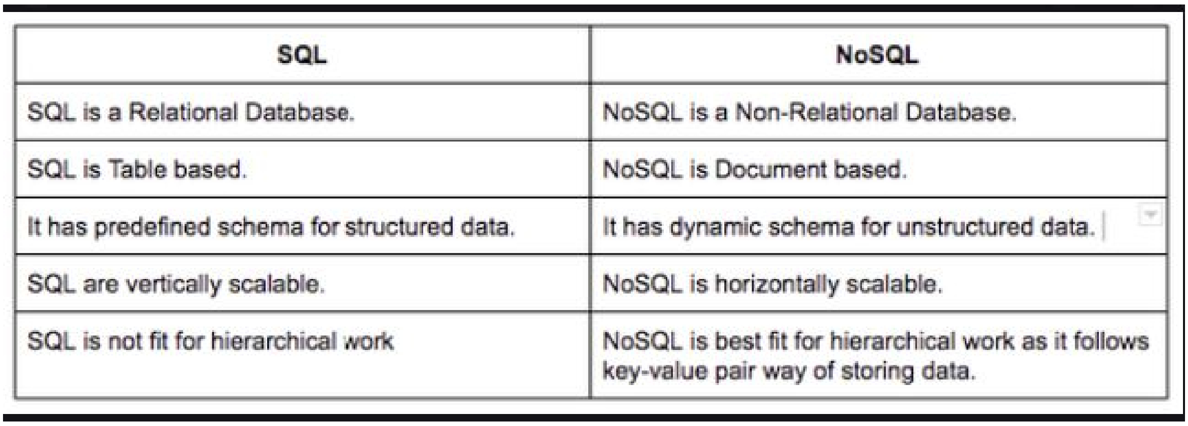
- Executor

6.Under which circumstances is preferable to run Spark in Standalone mode instead of relying on YARN?

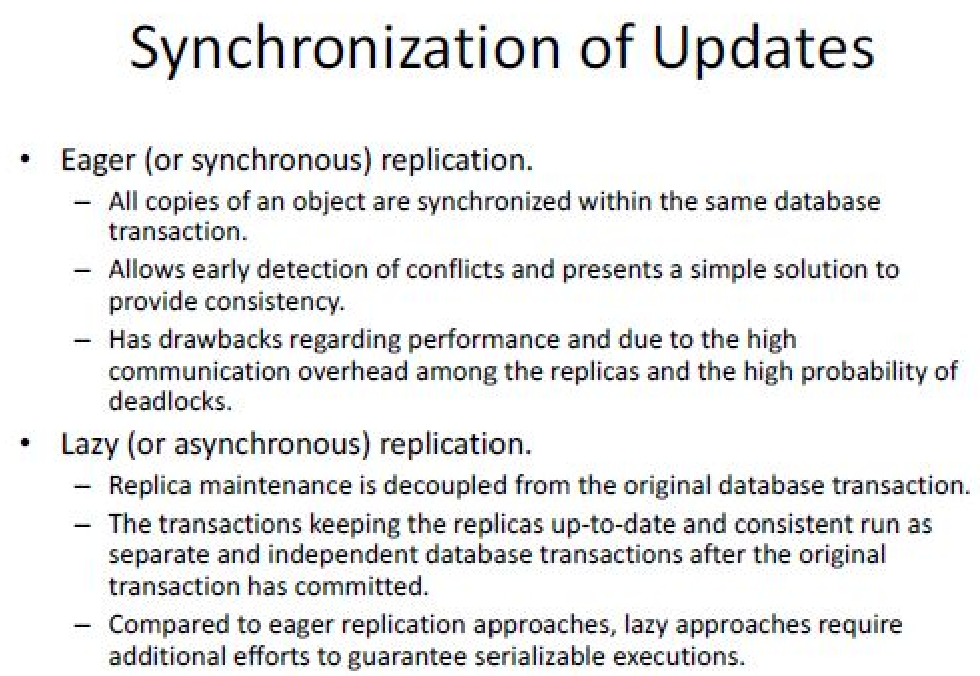
- When you only plan on running Spark jobs

Big Data implies a high volume, high velocity and extensible data (Structured, Semi Structured and even Unstructured data).

- It is faster than Hadoop.
- Intended to handle large scale data.
- Spark is build on top of Scala programming language.
- Spark SQL construct structured data into CSV or JSON.
- Hadoop is entirely dependent on MapReduce.
- When DAG is created it does not compute or perform any actions unless an Action occurs.
- Spark Context is the entry point of main function and allows to communicate with other nodes within cluster.
- Worker node have many number of executors & Tasks.
- Apache Mesos is a third party cluster manager.
- Apache Spark is a Fast and general-purpose cluster computing System.

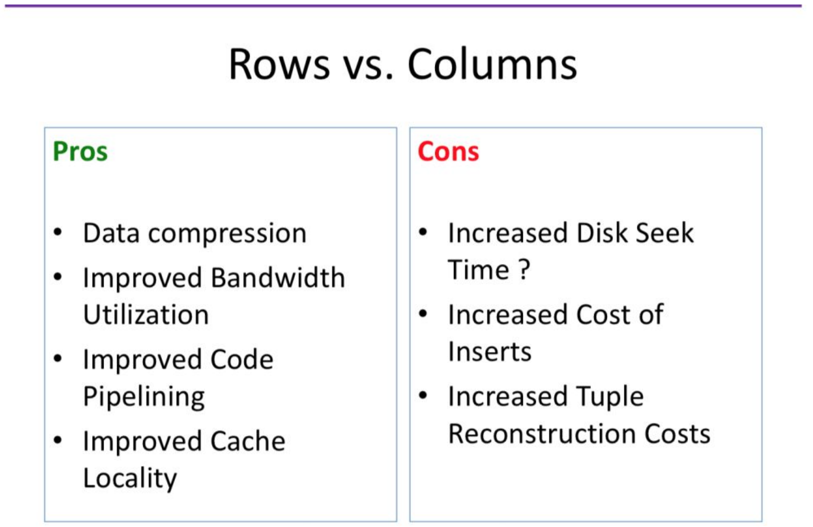


SQL	NoSQL
SQL is a Relational Database.	NoSQL is a Non-Relational Database.
SQL is Table based.	NoSQL is Document based.
It has predefined schema for structured data.	It has dynamic schema for unstructured data.
SQL are vertically scalable.	NoSQL is horizontally scalable.
SQL is not fit for hierarchical work	NoSQL is best fit for hierarchical work as it follows key-value pair way of storing data.



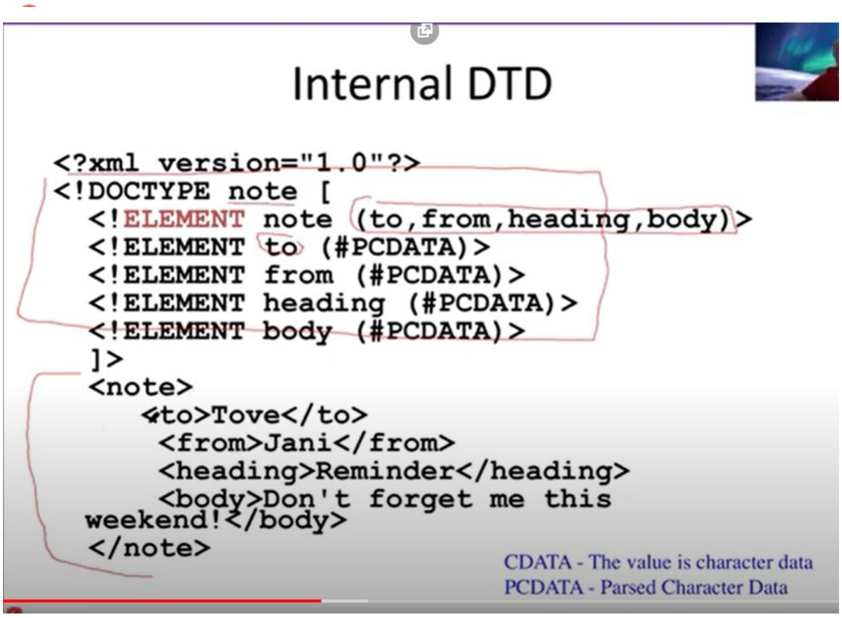
Synchronization of Updates

- Eager (or synchronous) replication.
  - All copies of an object are synchronized within the same database transaction.
  - Allows early detection of conflicts and presents a simple solution to provide consistency.
  - Has drawbacks regarding performance and due to the high communication overhead among the replicas and the high probability of deadlocks.
- Lazy (or asynchronous) replication.
  - Replica maintenance is decoupled from the original database transaction.
  - The transactions keeping the replicas up-to-date and consistent run as separate and independent database transactions after the original transaction has committed.
  - Compared to eager replication approaches, lazy approaches require additional efforts to guarantee serializable executions.



Rows vs. Columns

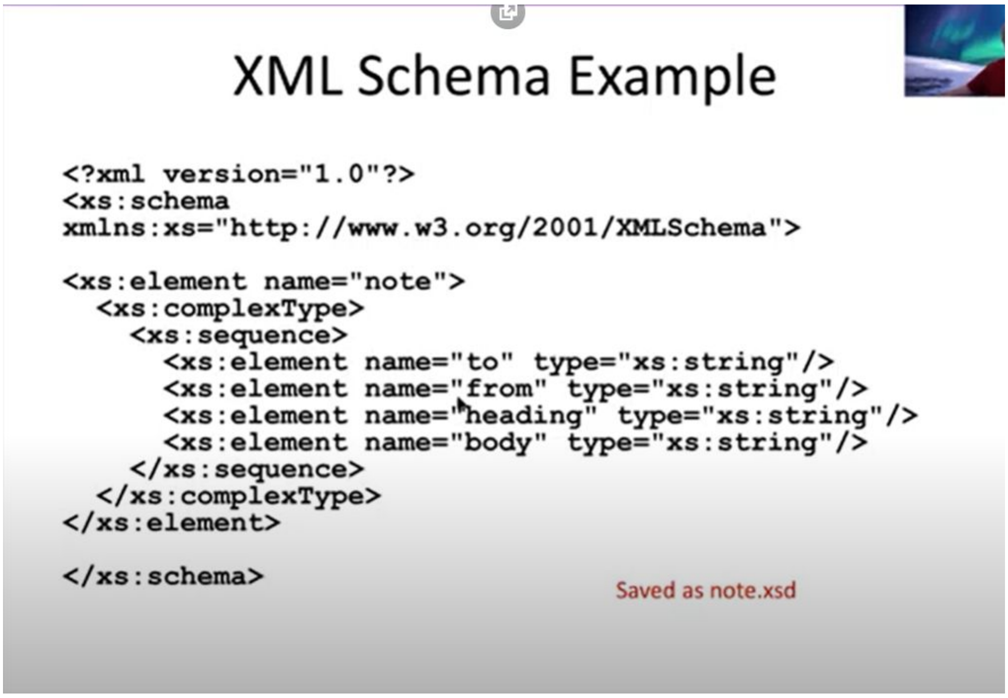
Pros	Cons
• Data compression • Improved Bandwidth Utilization • Improved Code Pipelining • Improved Cache Locality	• Increased Disk Seek Time ? • Increased Cost of Inserts • Increased Tuple Reconstruction Costs



Internal DTD

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
  ]>
  <note>
     <to>Tove</to>
      <from>Jani</from>
      <heading>Reminder</heading>
      <body>Don't forget me this
 weekend!</body>
  </note>
```

CDATA - The value is character data
PCDATA - Parsed Character Data



XML Schema Example

```
<?xml version="1.0"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Saved as note.xsd

Graphs Embrace Relationships

- SQL/NOSQL examples have dealt with *implicitly* connected data.
  - Users infer semantic dependencies between entities, but the data models—and the databases themselves—are blind to these connections.

- We want a cohesive picture of the whole, including the connections between elements.

- In contrast to the SQL/NOSQL data stores we looked at before, in the graph world, connected data is *stored as connected data*.