

## **Unit 1: Hadoop Setup & HDFS**

### **1. Q: What does the command ls do in HDFS, and how is it different from Linux?**

A: In HDFS, ls lists files in the distributed file system, not your local machine. In your practical, you used hdfs dfs -ls to view files and ls -la to check hidden files and permissions.

### **2. Q: How do you create a directory in the HDFS system?**

A: You use the command hdfs dfs -mkdir <directory\_name>. In your journal, you used this command to create a folder named "FirstDir".

### **3. Q: What is the command to copy a file from your local Linux machine to HDFS?**

A: The command is hdfs dfs -put. You used hdfs dfs -put Firstfile1.txt /user/cloudera/FirstDir to upload your text file into the Hadoop system.

### **4. Q: You used the touchz command in your practical. How is it different from a standard touch command?**

A: The standard Linux touch updates the timestamp of a file. In HDFS, hadoop fs -touchz creates a file of zero length. I used it to create Firstfile.txt in the HDFS directory.

### **5. Q: How can you read the content of a file stored in HDFS?**

A: You use the cat command. Specifically, hdfs dfs -cat /path/to/file. You used this to verify the contents of Firstfile1.txt.

### **6. Q: What is the difference between rm and rmr?**

A: rm removes a specific file. rmr removes a directory and all its contents recursively. In my practical, the system warned that rmr is deprecated and suggested using rm -r instead.

### **7. Q: What did the output "66 66" mean when you ran the du command?**

A: The du command stands for Disk Usage. The first number is the actual file size, and the second is the total space consumed including replication. I observed this when checking the .Trash directory.

### **8. Q: Why did you get a "Permission denied" error when trying to cat a file?**

A: I tried to run cat on a path (/user/cloudera/FirstDir) that was either not a file or I didn't have the execute permission for it. The error message explicitly stated cat: Permission denied.

### **9. Q: What happens to a file when you delete it using rm?**

A: It isn't erased immediately; it is moved to a .Trash directory. My logs showed: "Moved to trash at: hdfs://.../.Trash/Current".

### **10. Q: How do you check the specific syntax or options for a command like put?**

A: You use the -usage or -help flag. My journal shows hdfs dfs -usage put revealed options like -f (overwrite) and -p (preserve permissions).

### **11. Q: What is the NameNode and what happens if it goes down?**

A: The NameNode is the master node that stores the metadata (file names, permissions, and location of blocks). If the NameNode goes down, the entire file system becomes inaccessible because the DataNodes don't know where the file blocks are.

**12. Q: Why is the default block size in HDFS 128 MB instead of a small size like 4 KB?**

A: HDFS is designed for huge files. If the block size were small (like 4 KB), the NameNode would have to store millions of metadata entries for a single large file, which would overload its RAM.

**13. Q: What is "Safe Mode" in Hadoop?**

A: Safe Mode is a maintenance state that the NameNode enters automatically on startup. In this mode, the file system is Read-Only; you cannot create or delete files until the DataNodes report their blocks are available.

**14. Q: How does HDFS ensure fault tolerance?**

A: Through Replication. By default, every block of data is replicated 3 times across different DataNodes. If one node fails, HDFS automatically reads from a replica on another node.

**15. Q: What is the fsck command?**

A: File System Check (hdfs fsck /). It is used to check the health of the HDFS system, identifying missing blocks or under-replicated blocks without actually reading the file data.

---

## Unit 2: MapReduce Programming

**1. Q: What are the two main phases of a MapReduce job?**

A: The Map phase, which splits and processes the input data, and the Reduce phase, which aggregates and summarizes the results.

**2. Q: In your WordCount program, what types did you use for the output Key and Value?**

A: I used Text for the key (representing the word) and IntWritable for the value (representing the count, usually 1).

**3. Q: Why do we use IntWritable instead of a regular Java int?**

A: Because data needs to travel over the network between different nodes in a cluster. IntWritable allows the integer to be serialized (converted into bytes) for this transfer.

**4. Q: What logic did you use inside the Mapper for the "Union" operation?**

A: I checked if the line was not empty, then wrote the line itself as the Key and a generic text value "1" to the context. This essentially passes all valid data from both files to the Reducer.

**5. Q: How did your "Intersection" program identify common records?**

A: In the Mapper, I tagged data: "A" for file1 and "B" for file2. In the Reducer, I used boolean flags (fromA and fromB) and only outputted records where both flags were true.

**6. Q: What does StringTokenizer do in your code?**

A: It is a Java utility I used in the Mapper to break a line of text into individual words (tokens) so they could be counted one by one.

**7. Q: In Matrix Multiplication, what was the format of your input data?**

A: The input was comma-separated in the format MatrixName, Row, Col, Value. For example, A,0,0,1 meant Matrix A, row 0, column 0 has value 1.

**8. Q: Why is the Mapper logic for Matrix Multiplication complex?**

A: For every element in Matrix A, I had to emit it multiple times (once for every column in Matrix B) so the Reducer would have all the necessary data to calculate the dot product.

**9. Q: What is the Driver class used for?**

A: It is the main class that configures the job. It sets the Mapper class, Reducer class, and Input/Output paths so Hadoop knows how to run the program.

**10. Q: What error happens if you run a job but the output folder already exists?**

A: Hadoop will throw an error. In my practical, I had to delete the output directory using hdfs dfs -rm -r /output\_path before re-running the job.

**11. Q: What is a Combiner in MapReduce?**

A: A Combiner is essentially a "Mini-Reducer" that runs locally on the Mapper node. It aggregates data before sending it over the network to the Reducer (e.g., creating local sums), reducing network traffic.

**12. Q: What is the difference between an InputSplit and an HDFS Block?**

A: An HDFS Block is a chunk of physical data stored on disk. An InputSplit is a logical chunk of data assigned to be processed by a single Mapper. Usually, split size equals block size, but they are conceptually different.

**13. Q: What is the "Shuffle and Sort" phase?**

A: It is the intermediate phase between Map and Reduce. It transfers the mapped data (Shuffle) to the appropriate Reducers and sorts the data by Key so the Reducer receives values in a contiguous list.

**14. Q: What is a Partitioner?**

A: A Partitioner decides which Key-Value pair goes to which Reducer. The default is HashPartitioner, which distributes keys evenly, but custom partitioners can send specific keys to specific Reducers.

**15. Q: What is Speculative Execution?**

A: If a task is running slower than expected, Hadoop launches a duplicate "speculative" task on another node. Whichever task finishes first is accepted, and the slower one is killed. This prevents slow nodes from delaying the job.

### **Unit 3: MongoDB**

**1. Q: What is a Collection in MongoDB?**

A: It is the equivalent of a Table in a relational database. In my practical, I created a collection named students.

**2. Q: How do you switch to a specific database?**

A: Using the command use <database\_name>. I used use sampleDB to switch to my database.

**3. Q: What command inserts multiple documents at once?**

A: db.collection.insertMany([...]). I passed an array of student records (Name, Age, Course, City) inside the brackets.

**4. Q: What is the \_id field seen in your output?**

A: It is a unique primary key automatically assigned by MongoDB. My logs showed values like ObjectId("69015dd...").

**5. Q: How do you filter records? (e.g., find students in Mumbai)**

A: I pass a query object to the find command: db.students.find({ city: "Mumbai" }).

**6. Q: How did you sort the students by age?**

A: I used .sort({ age: 1 }). The 1 indicates ascending order; -1 would be descending.

**7. Q: What does the \$set operator do in an update?**

A: It updates the value of a specific field without removing the rest of the document. I used it to change the city of the student named "Aakash" to "Panvel".

**8. Q: What does upsertedCount: 0 mean in your update result?**

A: "Upsert" means "Update or Insert". A count of 0 means the record I tried to update already existed, so it was updated rather than a new one being inserted.

**9. Q: How did you join two collections (Sales and Sales\_Profile)?**

A: I used the aggregate framework with the \$lookup operator. I matched the ID field from both collections to combine their data.

**10. Q: How did you export your database to a JSON file?**

A: I used the mongoexport command tool from the system terminal (not the mongo shell), specifying the database, collection, and the output filename sales.json.

**11. Q: What is the difference between MongoDB and RDBMS (SQL)?**

A: SQL databases are relational, table-based, and have a rigid schema. MongoDB is NoSQL, document-based (BSON), and has a dynamic schema (documents in the same collection can have different fields).

**12. Q: What is Sharding in MongoDB?**

A: Sharding is the process of distributing data across multiple machines. Replication handles High Availability (copies), while Sharding handles Scalability (splitting huge datasets).

**13. Q: What is BSON, and why does MongoDB use it instead of pure JSON?**

A: BSON stands for Binary JSON. MongoDB uses it because it supports more data types (like Date and Binary data) and is faster for machines to parse than text-based JSON.

#### **14. Q: What is the Aggregation Pipeline?**

A: It is a framework for data aggregation where documents pass through "stages" (like \$match, \$group, \$sort). The output of one stage is the input for the next.

#### **15. Q: Explain the CAP Theorem in the context of MongoDB.**

A: The CAP theorem states a distributed system can only have 2 of 3 properties: Consistency, Availability, Partition Tolerance. MongoDB prioritizes Consistency and Partition Tolerance (CP).

---

### **Unit 4: Hive**

#### **1. Q: What command lists all databases in Hive?**

A: SHOW DATABASES;;

#### **2. Q: How did you load data from a text file into a Hive table?**

A: I used LOAD DATA LOCAL INPATH 'path' INTO TABLE tablename. This copies the file from my local system into the Hive warehouse.

#### **3. Q: What does ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' mean?**

A: It tells Hive that the source file (like employees.txt) is a CSV file, so it knows to use commas to separate the column values.

#### **4. Q: What error did you face when creating the company\_db database?**

A: I got "Database company\_db already exists". This happened because I tried to create a database that I had already created in a previous session.

#### **5. Q: What is Partitioning in Hive?**

A: It splits the table storage into different folders based on a column value. I used it to split sales data by year and month.

#### **6. Q: How did you load data into a specific partition?**

A: I added the PARTITION clause to the load command: LOAD DATA ... INTO TABLE sales PARTITION (year=2024, month=1).

#### **7. Q: What is the difference between ORDER BY and SORT BY?**

A: ORDER BY sorts all data globally (using one reducer), while SORT BY sorts data only within each reducer. I used ORDER BY salary DESC to find the highest earners.

#### **8. Q: What is a Hive View?**

A: It is a virtual table saved as a query. I created a view named high\_salary to quickly access employees earning more than 50,000.

#### **9. Q: What is a Map-side Join?**

A: My logs showed "Starting to launch local task to process map join". This is an optimization where the smaller table (departments) is loaded into memory, making the join much faster than a standard Reduce-side join.

**10. Q: How does Hive store metadata?**

A: Hive stores the table definitions (schema) in a separate database called the Metastore (typically MySQL), while the actual data resides in HDFS.

**11. Q: What is the difference between an Internal (Managed) Table and an External Table?**

A: If you drop an Internal Table, Hive deletes both the metadata and the actual data. If you drop an External Table, Hive deletes only the metadata; the actual data remains safe in HDFS.

**12. Q: What is Bucketing and how is it different from Partitioning?**

A: Partitioning creates sub-directories based on column values. Bucketing creates fixed-size files based on the Hash of a column. Bucketing is better when a column has too many unique values to partition.

**13. Q: What is the Hive Metastore?**

A: It is the central repository where Hive stores structure information (schema, locations, types). Without it, Hive wouldn't know how to map files in HDFS to tables.

**14. Q: What is SerDe?**

A: It stands for Serializer/Deserializer. It tells Hive how to interpret data. For example, a CSV SerDe reads comma-separated files, while a JSON SerDe reads JSON files.

**15. Q: How does Hive actually execute queries?**

A: Hive is an abstraction layer. When you write HiveQL, the Hive driver converts that SQL query into a series of MapReduce (or Tez/Spark) jobs which run on the cluster.

---

**Unit 5: Pig****1. Q: What is the command to run the Pig shell?**

A: Simply typing pig opens the Grunt shell. I used pig -x local to run it in local mode.

**2. Q: How do you print the results of a relation to the screen?**

A: Using the DUMP operator. For example: DUMP student\_info;.

**3. Q: How do you save results to a file?**

A: Using the STORE operator. Example: STORE student\_info INTO 'path' USING PigStorage('|').

**4. Q: What does the FILTER operator do?**

A: It selects rows that match a condition. I used FILTER students BY marks > 80 to find the top students.

**5. Q: What is the SPLIT operator?**

A: It divides one dataset into two or more datasets. I used it to split students into younger\_students (age < 23) and older\_students.

**6. Q: What does FOREACH ... GENERATE do?**

A: It iterates through every row to transform data or select columns. I used it to calculate the average marks of students: GENERATE AVG(students.marks).

**7. Q: What is the ILLUSTRATE command?**

A: It runs a simulation on a small sample of data to show step-by-step how the data is transformed. I used ILLUSTRATE foreachstudent to verify my logic.

**8. Q: What is COGROUP?**

A: It groups two different datasets by a common key. The result contains the key and two separate "bags" of tuples (one bag from each dataset).

**9. Q: What is a "Bag" in Pig?**

A: A Bag is a collection of tuples (rows), represented by curly braces {}. In my output, grouped data appeared as {(Ashok, 22), (Sudha, 23)}.

**10. Q: In your Pig WordCount, what did FLATTEN(TOKENIZE(line)) do?**

A: TOKENIZE splits a line into a bag of words. FLATTEN un-nests that bag, creating a separate row for each word so they can be counted individually.

**11. Q: Why would you use Pig over MapReduce Java code?**

A: Pig Latin is a high-level language. 10 lines of Pig Latin can replace 200+ lines of Java MapReduce code. It is easier to maintain, and the compiler automatically optimizes the execution.

**12. Q: What is "Schema on Read"?**

A: Pig/Hadoop doesn't verify data structure when loading (like SQL does). It only applies the schema when you read or process the data. If data doesn't match, it is handled as a runtime error.

**13. Q: Does Pig execute commands line-by-line?**

A: No, Pig uses Lazy Evaluation. It checks syntax but doesn't actually run any jobs until you call an output command like DUMP or STORE.

**14. Q: What are the Complex Data Types in Pig?**

A: Tuple (an ordered set of fields), Bag (a collection of tuples), and Map (a set of key-value pairs).

**15. Q: What is a UDF in Pig?**

A: User Defined Function. If Pig's built-in functions aren't enough, you can write custom logic in Java or Python and register it in your script.

---

## Unit 6: Spark

**1. Q: What is the "sc" variable in the Spark Shell?**

A: It stands for Spark Context. It is the main entry point to Spark functionality. My logs showed Spark context available as sc.

**2. Q: How do you load a text file into Spark?**

A: Using `val textFile = sc.textFile("path")`. This creates an RDD (Resilient Distributed Dataset).

**3. Q: What does the collect() command do?**

A: It brings all the data from the distributed cluster back to the driver program so you can see it. I used `words.collect()` to display the array of words.

**4. Q: What is the difference between map and flatMap?**

A: map is one-to-one. flatMap is one-to-many. I used flatMap to split one line into multiple words.

**5. Q: What does reduceByKey(\_ + \_) mean?**

A: It groups pairs by key (the word) and adds the values together. The `_ + _` is Scala shorthand for "add the first value to the second value".

**6. Q: Why did you map every word to (word, 1)?**

A: To create a Key-Value pair where every word counts as "1". This allows `reduceByKey` to simply sum these 1s to get the total count.

**7. Q: What is "Lazy Evaluation" in Spark?**

A: Spark doesn't execute transformations like `map` immediately. It waits until an Action (like `collect`) is called to actually run the computation.

**8. Q: What language did you use in the Spark practical?**

A: I used Scala, which is why the variable declarations start with `val` and the logs mention Using Scala version 2.10.5.

**9. Q: What is an RDD?**

A: Resilient Distributed Dataset. It is the main data structure in Spark—an immutable, distributed collection of objects.

**10. Q: What is a Lineage Graph (DAG)?**

A: It is the history of transformations applied to the data. If data is lost, Spark uses this graph to recompute it, which makes the dataset "Resilient".

**11. Q: What is the difference between a Transformation and an Action?**

A: Transformations (like `map`, `filter`) are lazy and create a new RDD. Actions (like `collect`, `count`) trigger the actual computation and return results.

**12. Q: What is the difference between Narrow and Wide Dependencies?**

A: Narrow dependencies (like `map`) happen locally without moving data. Wide dependencies (like `reduceByKey`) require data shuffling across the network.

**13. Q: What is DAG (Directed Acyclic Graph)?**

A: It is the execution plan Spark builds. Before running, Spark looks at all transformations and creates a graph to optimize the execution path.

#### **14. Q: What is Caching/Persisting in Spark?**

A: If you reuse an RDD multiple times, you can cache() it. This keeps the data in RAM so Spark doesn't have to re-compute it from the beginning every time.

#### **15. Q: What are Broadcast Variables?**

A: If you have a large read-only variable (like a lookup table) that every node needs, a Broadcast Variable sends it to each machine once, rather than sending it with every single task.

---

### **Unit 7: Visualization (Tableau)**

#### **1. Q: What are "Dimensions" in Tableau?**

A: They are qualitative data fields (like Name, Date, City). They usually appear in Blue in the sidebar.

#### **2. Q: What are "Measures"?**

A: They are quantitative data fields (numbers like Sales, Profit). They usually appear in Green.

#### **3. Q: How did you create a Geographic Map?**

A: I double-clicked the "State" field. Tableau recognized the geographic role (globe icon) and automatically plotted the states.

#### **4. Q: What chart type did you use to compare Marketing vs Sales?**

A: A Scatter Plot. It placed dots to show the correlation between marketing spend (x-axis) and sales (y-axis).

#### **5. Q: What is a Treemap?**

A: A chart using nested rectangles. I used it to show Profit by Product—size represented Sales volume and color represented Profit.

#### **6. Q: What is a Dashboard?**

A: A single view that displays multiple worksheets simultaneously. I combined "Sales by Product" and "Profit by State" onto one dashboard.

#### **7. Q: What is a Story in Tableau?**

A: It is like a presentation built into Tableau. It uses "Story Points" to walk the audience through insights step-by-step.

#### **8. Q: What does the "Show Me" panel do?**

A: It highlights the chart types that are possible based on the data fields I have currently selected.

#### **9. Q: What is a Calculated Field?**

A: It is a new field created by applying a formula to existing data. This allows for custom metrics not present in the original dataset.

**10. Q: What is the difference between a Live connection and an Extract?**

A: A Live connection updates automatically when the source data changes. An Extract is a static snapshot of the data, which usually improves performance.

**11. Q: What is the difference between Joining and Data Blending?**

A: Joining combines tables from the same source row-by-row. Data Blending combines data from different sources by aggregating data first and then linking them.

**12. Q: What is an LOD Expression?**

A: Level of Detail Expression. It allows you to compute values at the data source level and the visualization level independently (e.g., average sales per country while viewing sales per city).

**13. Q: What are Context Filters?**

A: Filters that run first and create a temporary subset of data. All other filters apply only to that subset, which improves performance.

**14. Q: Explain the difference between discrete (Blue) and continuous (Green) fields.**

A: Discrete fields create headers and break the view into distinct buckets. Continuous fields create an axis and show a range of values.

**15. Q: How does a Tableau Extract improve performance?**

A: An Extract is a snapshot stored in Tableau's proprietary "Hyper" format. It is columnar and compressed, allowing Tableau to query it much faster than connecting to a slow excel file or database.