Q1Explain about the different complex data types in pig**?**

A1

1. **Atom**
   * Any single value in Pig Latin, irrespective of their data or type is known as an Atom.
   * It is stored as bytearray by default and can be used as string or number like int, long, float, double, chararray, and bytearray are the atomic values of Pig.
   * A piece of data or a simple atomic value is known as a field.

Example − ‘raja’ or ‘30’

1. **Tuple**
   * A record that is formed by an ordered set of fields is known as a tuple, the fields can be of any type.
   * A tuple is similar to a row in a table of RDBMS.

Example − (Raja, 30)

1. **Bag**
   * A bag is an unordered set of tuples.
   * In other words, a collection of tuples (non-unique) is known as a bag.
   * Each tuple can have any number of fields (flexible schema).
   * A bag is represented by ‘{}’.
   * It is similar to a table in RDBMS, but unlike a table in RDBMS, it is not necessary that every tuple contain the same number of fields or that the fields in the same position (column) have the same type. Example − {(Raja, 30), (Mohammad, 45)}
   * A bag can be a field in a relation; in that context, it is known as inner bag.

Example − {Raja, 30, {9848022338, [raja@gmail.com,}](mailto:raja@gmail.com,%7d)}

1. **Map** 
   * A map (or data map) is a set of key-value pairs.
   * The key needs to be of type chararray and should be unique.
   * The value might be of any type. It is represented by ‘[]’

Example: [name#Raja, age#30]

1. **Relation** 
   * A relation is an outer bag of tuples.
   * The relations in Pig Latin are unordered (there is no guarantee that tuples are processed in any particular order).

Q2. How can you interact with the shell in Apache pig ?

A2

**With Grunt Shell**

• Interactive Shell for executing Pig Commands

• Used when script file is not provided

• Can execute scripts from Grunt via run or exec commands

**Using Script Files**

• Executes Commands in a file

• pig ScriptFile.pig

• Pig commands are executed using script files as batch Jobs

Q3. Explain how pig differs from Map reduce?

A3

PIG is a data flow language, the key focus of Pig is manage the flow of data from input source to output store. As part of managing this data flow it moves **data** feeding it to process1, taking output and feeding it to process2. The core features are preventing execution of subsequent stages if previous stage fails, manages temporary storage of data and most importantly compresses and rearranges processing steps for faster processing. While this can be done for any kind of processing tasks Pig is written specifically for managing data flow of Map reduce type of jobs. Most if not all jobs in a Pig are map reduce jobs or data movement jobs. Pig allows for custom functions to be added which can be used for processing in Pig, some default ones are like ordering, grouping, distinct, count etc.

    Map reduce on the other hand is a data processing paradigm, it is a framework for application developers to write code in so that its easily scaled to PB of tasks, this creates a separation between the developer that writes the application vs the developer that scales the application. Not all applications can be migrated to Map reduce but good few can be including complex ones like k-means to simple ones like counting uniques in a dataset.

Q 4. Explain how pig differs from sql?

A4

* In SQL, when users want to do several data operations together, they must either write separate queries, storing the intermediate data into temporary tables, or write it in one query using subqueries inside that query to do the earlier steps of the processing.
* Pig, however, is designed with a long series of data operations in mind, so there is no need to write the data pipeline in an inverted set of subqueries or to worry about storing data in temporary tables.
* SQL is designed for the RDBMS environment, where data is normalized and schemas and proper constraints are enforced (that is, there are no nulls in places they do not belong, etc.).
* Pig is designed for the Hadoop data-processing environment, where schemas are sometimes unknown or inconsistent. Pig does not require data to be loaded into tables first. It can operate on data as soon as it is copied into HDFS.
* Pig Latin is the native language of parallel data-processing systems.

Q 5. Explain the scalar data types in pig?

A5

