1. **Explain about the different complex data types in pig**

There are 5 complex Data Types in pig and they are

1.Atom

2.Tuple

3.Bag

4.Map

5.Relation

***Atom***

1. Any single value in Pig Latin is called as Atom irrespective of data or type it belongs to.
2. By default the storage will be in byte array as well can be used as string or number (like int, long, float, double, char array, and byte array ) which are the atomic values of Pig.
3. A piece of data or else simple atomic value is called as field.
4. For Example − ‘sai’-char array or ‘23’ – int.

***Tuple***

1. Tuple – is called as a record which is formed by ordered set of fields in a tuple and those fields can be of any type.
2. It is similar to rows in a table of RDBMS.
3. A tuple is represented by ‘( )’ –double braces.

Example − (sai, 23)

***Bag***

1. Unordered set of tuples is a called as a bag.
2. And in a bag the collection will not be unique.
3. And it is possible to have a n number of flexible and so it is a flexible scheme.
4. It is represented by ‘{}’.
5. RDBMS is similar to a bag , but the difference is that there is no need for that to contain same number of fields or the position of fields need not to be similar (i.e the coloumn).

Example − {(Sai, 23), (Mathurri , 20)}

And it is possible to have a bag in relation to a field and in that context it is called as inner bag.

Example − {sai, 23, {8476476874867, math@yahoo.com}

***Map***

1. A map is nothing but a set of key-value pairs which can also be called as data map.
2. The key should be unique and it need to be a chararray.
3. Any type of value can be used.
4. It is represented by ‘[]’.
5. Example: [name#Sai, age#23]

***Relation***

1. Outer bag of tuples is called as an relation.
2. In case of Pig Latin the relations will be are unordered ( there is no particular order for tuples to follow).

**2.How can you interact with the shell in Apache pig:**

There are two method to interact Pig shell and they are:

1. ***With Grunt Shell***

• Interactive Shell for executing Pig Commands

• can be used when there is no pig file.

• Through run or exec commands it can execute scripts from Grunt.

1. ***Using Script Files***

• It is used to Execute Commands in a file .

• Pig commands are executed in a batch via script files.

1. **Explain how pig differs from Map reduce**

|  |  |
| --- | --- |
| **PIG** | **MAP REDUCE** |
| Both Structured and semi structured data can be processed. | Structured ,semi structured and un structured data can be processed. |
| Joins, querying etc. can be performed in it. | Only map reduce can run. |
| Time taken for development of code will be less. | Development time is high (code development) |
| Since it will start to run on top of map reduce so the time for process will be high. | Processing Time will be less when compared to pig. |
| Useful for performing the analysis process. |  |

1. **Explain how pig differs from sql:**
   * when several data operations need to be done at a time then separate sql queries should be written so that intermediate data will be stored into the temporary tables or can sub-queries can be written inside the main query for early processing.
   * The basic design process is using a long data series keeping data operation in mind so no need for data pipeline in a inverted sub-query or temporary data storing.
   * to support RDMS SQL is designed. Where proper constrains were used and data to be used will be normalized and schema will also be enforced.
   * in case of pig there is no compulsion that data should be loaded in the first row and it is just a design for Hadoop data-processing environment in which the schema provided will be unknown or will be in consistent.
   * The data will be available for use as soon as it is copied to the HDFS.
   * The native language of parallel data-processing systems is Pig Latin.

**5.Explain the scalar data types in pig**

The scalar Data Types in pig are

1.Int

2.Long

3.Float

4.Doublr

5.Chararray

6.Bytearray

**1.Int**

* It is similar to integer datatype used in java.
* Where int are represented by java.lang.Integer as interfaces.
* They are capable to store a four-byte signed integer.
* Constant integers by default will be expressed as integer numbers.
* for example, 42.

**2.long**

* It is similar to Long datatype used in java.
* Where Long are represented by java.lang Long as interfaces.
* They are capable to store a eight-byte signed integer.
* Constant Longs by default will be expressed as integer numbers will L append.
* for example, 5000000000L.

**3.float**

* It is similar to integer datatype used in java.
* In simple it is a floating point number.
* Where int are represented by java.lang.Float as interfaces.
* Since it a floating point there are possibilities for the precision to get lost and int as well as long can be used for calculations.
* They are capable to store a four-byte signed integer.
* Constant floats are expressed as a floating-point number with an f appended.
* Constant integers by default will be expressed as integer numbers.
* for example, in simple format, 12.2C, or in exponent format, 7.896r23y.

**4.double**

* It is similar to integer datatype used in java.
* Where int are represented by java.lang.Double  as interfaces.
* Java’s Double type will give a range of values
* And there is a possibility in some of the calculations to loose its precision so there will be a need to use int or long.
* They are capable to store a eight-byte signed integer to store its values.

Constant doubles are expressed as floating-point number in either simple format, 2.71828, or in exponent format, 6.626e-34.

**5.chararray**

* It is similar to integer datatype used in java.
* They are represented by means of java.lang.String.
* They are expressed in single quotes as string literals .eg-‘sacred’.
* Char array can be used along with standard alphanumeric and symbolic characters by backslash codes like \n for return.
* \u followed by numbers will represent Unicode value.
* Eg-\p0909.
* They are capable to store a four-byte signed integer.
* Constant integers by default will be expressed as integer numbers.
* for example, 42.

1. **bytearray**

In an interfaces Bytearrays are represented by Java class DataByteArray

It will wrap a Java byte[].

There is no specification for constant bytearray.