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

**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**: **‘sairam’** or **‘88’**.

**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: (sairam, 88).**

**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)**}**

**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]**

**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).

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

Grunt shell is the Pig’s interactive shell. It enables users to enter Pig Latin interactively and provides a shell for users to interact with HDFS.

To enter Grunt, invoke Pig with no script or command to run. Typing:

Pig –x local;

will result in the prompt:

grunt>

This gives you a Grunt shell to interact with your local filesystem. If you omit the -x local and have a cluster configuration set in PIG\_CLASSPATH, this will put you in a Grunt shell that will interact with HDFS on your cluster.

To exit Grunt you can type quit or enter Ctrl-D.

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

|  |  |
| --- | --- |
| **Apache Pig** | **MapReduce** |
| Apache Pig is a data flow language. | MapReduce is a data processing paradigm. |
| It is a high level language. | MapReduce is low level and rigid. |
|  |  |
| Performing a Join operation in Apache Pig is pretty simple. | It is quite difficult in MapReduce to perform a Join operation between datasets. |
| Any novice programmer with a basic knowledge of SQL can work conveniently with Apache Pig. | Exposure to Java is must to work with MapReduce. |
| Apache Pig uses multi-query approach, thereby reducing the length of the codes to a great extent. | MapReduce will require almost 20 times more the number of lines to perform the same task. |
| There is no need for compilation. On execution, every Apache Pig operator is converted internally into a MapReduce job. | MapReduce jobs have a long compilation process. |

1. **Explain how pig differs from sql.**

|  |  |
| --- | --- |
| **Pig** | **SQL** |
| Pig Latin is a **procedural** language. | SQL is a **declarative** language. |
| In Apache Pig, **schema** is optional. We can store data without designing a schema (values are stored as $01, $02 etc.) | Schema is mandatory in SQL. |
| The data model in Apache Pig is **nested relational**. | The data model used in SQL **is flat relational**. |
| Apache Pig provides limited opportunity for **Query optimization**. | There is more opportunity for query optimization in SQL. |

1. **Explain the scalar data types in pig.**
2. **int** : It is signed 32 bit integer. This is similar to the Integer in java.

**Eg**: 10.

1. **long**: It is a 64 bit signed integer. This is similar to the Long in java.

**Eg**: 10L or 10l

1. **float** : It is a 32 bit floating point. This data type is similar to the Float in java.

**Eg:** 10.5F or 10.5f or 10.5e2f or 10.5E2F

1. **double** : It is a 63 bit floating pint. This data type is similar to the Double in java.

**Eg**: 10.5 or 10.5e2 or 10.5E2

1. **chararray** : It is character array in unicode UTF-8 format. This corresponds to java's String object.

**Eg**: hello world

1. **bytearray** : Used to represent bytes. It is the default data type. If you don't specify a data type for a filed, then bytearray datatype is assigned for the field.
2. **boolean** : to represent true/false values.