**1)  What do you mean by a bag in Pig?**

Collection of tuples is referred as a bag in Apache Pig

2) Does Pig support multi-line commands?

Yes

**3) Explain the need for MapReduce while programming in Apache Pig.**

Apache Pig programs are written in a query language known as Pig Latin that is similar to the SQL query language. To execute the query, there is a need for an execution engine. The Pig engine converts the queries into MapReduce jobs and thus MapReduce acts as the execution engine and is needed to run the programs.

**4) Explain about the BloomMapFile.**

BloomMapFile is a class, that extends the MapFile class. It is used in HBase table format to provide quick membership test for the keys using dynamic bloom filters.

**5) Differentiate between Hadoop MapReduce and Pig**

* Pig provides higher level of abstraction whereas MapReduce provides low level of abstraction.
* MapReduce requires the developers to write more lines of code when compared to Apache Pig.
* Pig coding approach is comparatively slower than the fully tuned MapReduce coding approach.

Read More in Detail- <http://www.dezyre.com/article/-mapreduce-vs-pig-vs-hive/163>

**6) What is the usage of foreach operation in Pig scripts?**

FOREACH operation in Apache Pig is used to apply transformation to each element in the data bag, so that respective action is performed to generate new data items.

Syntax- FOREACH data\_bagname GENERATE exp1, exp2

**7) Explain about the different complex data types in Pig.**

Apache Pig supports 3 complex data types-

* Maps- These are key, value stores joined together using #.
* Tuples- Just similar to the row in a table, where different items are separated by a comma. Tuples can have multiple attributes.
* Bags- Unordered collection of tuples. Bag allows multiple duplicate tuples.

**8) What does Flatten do in Pig?**

Sometimes there is data in a tuple or a bag and if we want to remove the level of nesting from that data, then Flatten modifier in Pig can be used. Flatten un-nests bags and tuples. For tuples, the Flatten operator will substitute the fields of a tuple in place of a tuple, whereas un-nesting bags is a little complex because it requires creating new tuples.

**9) How do users interact with the shell in Apache Pig?**

Using Grunt i.e. Apache Pig’s interactive shell, users can interact with HDFS or the local file system. To start Grunt, users should invoke Apache Pig with no command –

Executing the command “pig –x local” will result in the prompt -

grunt >

This is where PigLatin scripts can be run either in local mode or in cluster mode by setting the configuration in PIG\_CLASSPATH.

To exit from grunt shell, press CTRL+D or just type exit.

**10) What are the debugging tools used for Apache Pig scripts?**

describe and explain are the important debugging utilities in Apache Pig.

* explain utility is helpful for Hadoop developers, when trying to debug error or optimize PigLatin scripts. explain can be applied on a particular alias in the script or it can be applied to the entire script in the grunt interactive shell. explain utility produces several graphs in text format which can be printed to a file.
* describe debugging utility is helpful to developers when writing Pig scripts as it shows the schema of a relation in the script. For beginners who are trying to learn Apache Pig can use the describe utility to understand how each operator makes alterations to data. A pig script can have multiple describes.

**11) What is illustrate used for in Apache Pig?**

Executing pig scripts on large data sets, usually takes a long time. To tackle this, developers run pig scripts on sample data but there is possibility that the sample data selected, might not execute your pig script properly. For instance, if the script has a join operator there should be at least a few records in the sample data that have the same key, otherwise the join operation will not return any results. To tackle these kind of issues, illustrate is used. illustrate takes a sample from the data and whenever it comes across operators like join or filter that remove data, it ensures that only some records pass through and some do not, by making modifications to the records such that they meet the condition. illustrate just shows the output of each stage but does not run any MapReduce task.

**12) Explain about the execution plans of a Pig Script**

**Or**

**Differentiate between the logical and physical plan of an Apache Pig script**

Logical and Physical plans are created during the execution of a pig script. Pig scripts are based on interpreter checking. Logical plan is produced after semantic checking and basic parsing and no data processing takes place during the creation of a logical plan. For each line in the Pig script, syntax check is performed for operators and a logical plan is created. Whenever an error is encountered within the script, an exception is thrown and the program execution ends, else for each statement in the script has its own logical plan.

A logical plan contains collection of operators in the script but does not contain the edges between the operators.

After the logical plan is generated, the script execution moves to the physical plan where there is a description about the physical operators, Apache Pig will use, to execute the Pig script. A physical plan is more or less like a series of MapReduce jobs but then the plan does not have any reference on how it will be executed in MapReduce. During the creation of physical plan, cogroup logical operator is converted into 3 physical operators namely –Local Rearrange, Global Rearrange and Package. Load and store functions usually get resolved in the physical plan.

**13) What do you know about the case sensitivity of Apache Pig?**

It is difficult to say whether Apache Pig is case sensitive or case insensitive. For instance, user defined functions, relations and field names in pig are case sensitive i.e. the function  COUNT is not the same as function count or X=load ‘foo’ is not same as x=load ‘foo’. On the other hand, keywords in Apache Pig are case insensitive i.e. LOAD is same as load.

**14) What are some of the Apache Pig use cases you can think of?**

Apache Pig big data tools, is used in particular for iterative processing, research on raw data and for traditional ETL data pipelines. As Pig can operate in circumstances where the schema is not known, inconsistent or incomplete- it is widely used by researchers who want to make use of the data before it is cleaned and loaded into the data warehouse.

To build behaviour prediction models, for instance, it can be used by a website to track the response of the visitors to various types of ads, images, articles, etc.

**15) Differentiate between PigLatin and HiveQL**

* It is necessary to specify the schema in HiveQL, whereas it is optional in PigLatin.
* HiveQL is a declarative language, whereas PigLatin is procedural.
* HiveQL follows a flat relational data model, whereas PigLatin has nested relational data model.

Read More about [Pig vs. Hive](https://www.dezyre.com/article/difference-between-pig-and-hive-the-two-key-components-of-hadoop-ecosystem/79)

**16) Is PigLatin a strongly typed language? If yes, then how did you come to the conclusion?**

In a strongly typed language, the user has to declare the type of all variables upfront. In Apache Pig, when you describe the schema of the data, it expects the data to come in the same format you mentioned. However, when the schema is not known, the script will adapt to actually data types at runtime. So, it can be said that PigLatin is strongly typed in most cases but in rare cases it is gently typed, i.e. it continues to work with data that does not live up to its expectations.

**17) What do you understand by an inner bag and outer bag in Pig?**

A relation inside a bag is referred to as inner bag and outer bag is just a relation in Pig

**18) Differentiate between GROUP and COGROUP operators.**

Both GROUP and COGROUP operators are identical and can work with one or more relations. GROUP operator is generally used to group the data in a single relation for better readability, whereas COGROUP can be used to group the data in 2 or more relations. COGROUP is more like a combination of GROUP and JOIN, i.e., it groups the tables based on a column and then joins them on the grouped columns. It is possible to cogroup up to 127 relations at a time.

**19) Explain the difference between COUNT\_STAR and COUNT functions in Apache Pig?**

COUNT function does not include the NULL value when counting the number of elements in a bag, whereas COUNT\_STAR (0 function includes NULL values while counting.

**20) What are the various diagnostic operators available in Apache Pig?**

1. **Dump Operator-** It is used to display the output of pig Latin statements on the screen, so that developers can debug the code.
2. **Describe Operator-**Explained in apache pig interview question no- 10
3. **Explain Operator-**Explained in apache pig interview question no -10
4. **Illustrate Operator-** Explained in apache pig interview question no -11

**21) How will you merge the contents of two or more relations and divide a single relation into two or more relations?**

This can be accomplished using the UNION and SPLIT operators.

**22) I have a relation R. How can I get the top 10 tuples from the relation R.?**

TOP () function returns the top N tuples from a bag of tuples or a relation. N is passed as a parameter to the function top () along with the column whose values are to be compared and the relation R.

**23) What are the commonalities between Pig and Hive?**

* HiveQL and PigLatin both convert the commands into MapReduce jobs.
* They cannot be used for OLAP transactions as it is difficult to execute low latency queries.

**24)  What are the different types of UDF’s in Java supported by Apache Pig?**

Algebraic, Eval and Filter functions are the various types of UDF’s supported in Pig.

**25) You have a file employee.txt in the HDFS directory with 100 records. You want to see only the first 10 records from the employee.txt file. How will you do this?**

The first step would be to load the file employee.txt into with the relation name as Employee.

The first 10 records of the employee data can be obtained using the limit operator -

Result= limit employee 10.