



### **1. What is pig?**

Pig is a Apache open source project which is run on hadoop,provides engine for data flow in parallel on hadoop.It includes language called pig latin,which is for expressing these data flow.It includes different operations like joins,sort,filter ..etc and also ability to write UserDefine Functions(UDF) for proceesing and reaing and writing.pig uses both HDFS and MapReduce i,e storing and processing.

### **2. What is differnce between pig and sql?**

Pig latin is procedural version of SQL.pig has certainly similarities,more difference from sql.sql is a query language for user asking question in query form.sql makes answer for given but dont tell how to answer the given question.suppose ,if user want to do multiple operations on tables,we have write maultiple queries and also use temporary table for storing,sql is support for subqueries but intermediate we have to use temporary tables,SQL users find subqueries confusing and difficult to form properly.using sub-queries creates an inside-out design where the first step in the data pipeline is the innermost query .pig 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.

### **3. How Pig differs from MapReduce?**

In mapreduce,groupby operation performed at reducer side and filter,projection can be implemented in the map phase.pig latin also provides standard-operation similar to mapreduce like orderby and filters,groupby..etc.we can analyze pig script and know data flows ans also early to find the error checking.pig Latin is much lower cost to write and maintain thanJava code for MapReduce.



#### 4. How is Pig Useful For?

In three categories, we can use pig. They are 1) ETL data pipeline 2) Research on raw data 3) Iterative processing

Most common use case for pig is data pipeline. Let us take one example, web-based companies get the weblogs, so before storing data into warehouse, they do some operations on data like cleaning and aggregation operations...etc. i.e. transformations on data.

#### 5. What are the scalar datatypes in pig?

scalar datatype

int -4bytes,

float -4bytes,

double -8bytes,

long -8bytes,

chararray,

bytearray

#### 6. What are the complex datatypes in pig?

**map:**

map in pig is chararray to data element mapping where element has pig data type including complex data type.

example of map ['city'#:hyd', 'pin'#:500086]

the above example city and pin are data elements (key) mapping to values

**tuple:**

tuple has fixed length and it has collection datatypes. tuple containing multiple fields and also tuples are ordered.

example, (hyd,500086) which contains two fields.



**bag:**

A bag containing collection of tuples which are unordered, Bag constants are constructed using braces, with tuples in the bag separated by commas. For example, {('hyd', 500086), ('chennai', 510071), ('bombay', 500185)}

**7. Whether pig latin language is case-sensitive or not?**

pig latin is sometimes not a case sensitive. let us see example, Load is equivalent to load.

A=load 'b' is not equivalent to a=load 'b'

UDF are also case sensitive, count is not equivalent to COUNT.

**8. How should 'load' keyword is useful in pig scripts?**

first step in dataflow language we need to specify the input, which is done by using 'load' keyword. load looks for your data on HDFS in a tab-delimited file using the default load function 'PigStorage'. suppose if we want to load data from hbase, we would use the loader for hbase

'HBaseStorage'.

example of pigstorage loader

A = LOAD '/home/ravi/work/flight.tsv' using PigStorage ('t') AS

(origincode:chararray, destinationcode:chararray, origincity:chararray, destinationcity:chararray, passengers:int, seats:int, flights:int, distance:int, year:int, originpopulation:int, destpopulation:int);

example of hbasestorage loader

x= load 'a' using HBaseStorage();

if dont specify any loader function, it will takes built in function is 'PigStorage' the 'load' statement can also have 'as' keyword for creating schema, which allows you to specify the schema of the data you are loading.



PigStorage and TextLoader, the two built-in Pig load functions that operate on HDFS files.

### 9. How should 'store' keyword is useful in pig scripts?

After we have completed process, then result should write into somewhere, Pig provides the store statement for this purpose

store processed into '/data/ex/process';

If you do not specify a store function, PigStorage will be used. You can specify a different store function with a using clause:

store processed into '?processed' using HBaseStorage();

we can also pass argument to store function, example, store processed into 'processed' using PigStorage(',');

### 10. What is the purpose of 'dump' keyword in pig?

dump display the output on the screen

dump 'processed'

### 11. what are relational operations in pig latin?

they are

a) for each

b) order by

c) filters

d) group

e) distinct

f) join

g) limit

### 12. How to use 'foreach' operation in pig scripts?



foreach takes a set of expressions and applies them to every record in the data pipeline

```
A = load 'input' as (user:chararray, id:long, address:chararray,  
phone:chararray,preferences:map[]);  
B = foreach A generate user, id;
```

positional references are preceded by a \$ (dollar sign) and start from 0:  
c= load d generate \$2-\$1

### **13. How to write 'foreach' statement for map datatype in pig scripts?**

```
for map we can use hash('#')  
bball = load 'baseball' as (name:chararray,  
team:chararray,position:bag{t:(p:chararray)}, bat:map[]);  
avg = foreach bball generate bat#'batting_average';
```

### **14. How to write 'foreach' statement for tuple datatype in pig scripts?**

```
for tuple we can use dot('.')  
A = load 'input' as (t:tuple(x:int, y:int));  
B = foreach A generate t.x, t.$1;
```

### **15. How to write 'foreach' statement for bag datatype in pig scripts?**

when you project fields in a bag, you are creating a new bag with only those fields:

```
A = load 'input' as (b:bag{t:(x:int, y:int)});  
B = foreach A generate b.x;
```

we can also project multiple field in bag

```
A = load 'input' as (b:bag{t:(x:int, y:int)});  
B = foreach A generate b.(x, y);
```



## 16. Why should we use 'filters' in pig scripts?

Filters are similar to where clause in SQL.filter which contain predicate.If that predicate evaluates to true for a given record, that record will be passed down the pipeline. Otherwise, it will not.predicate contain different operators like ==,>=,<=,!=.so,== and != can be applied to maps and tuples.  
A= load 'inputs' as(name,address)

B=filter A by symbol matches 'CM.\*';

## 17. Why should we use 'group' keyword in pig scripts?

The group statement collects together records with the same key.In SQL the group by clause creates a group that must feed directly into one or more aggregate functions. In Pig Latin there is no direct connection between group and aggregate functions.

input2 = load 'daily' as (exchanges, stocks);

grpds = group input2 by stocks;

## 18. Why should we use 'orderby' keyword in pig scripts?

The order statement sorts your data for you, producing a total order of your output data.The syntax of order is similar to group. You indicate a key or set of keys by which you wish to order your data

input2 = load 'daily' as (exchanges, stocks);

grpds = order input2 by exchanges;

## 19. Why should we use 'distinct' keyword in pig scripts?

The distinct statement is very simple. It removes duplicate records. It works only on entire records, not on individual fields:

input2 = load 'daily' as (exchanges, stocks);

grpds = distinct exchanges;



## 20. Is it possible to join multiple fields in pig scripts?

yes, Join select records from one input and join with another input. This is done by indicating keys for each input. When those keys are equal, the two rows are joined.

```
input2 = load 'daily' as (exchanges, stocks);
```

```
input3 = load 'week' as (exchanges, stocks);
```

```
grpds = join input2 by stocks, input3 by stocks;
```

we can also join multiple keys

example:

```
input2 = load 'daily' as (exchanges, stocks);
```

```
input3 = load 'week' as (exchanges, stocks);
```

```
grpds = join input2 by (exchanges, stocks), input3 by (exchanges, stocks);
```

## 21. Is it possible to display the limited no of results?

yes, Sometimes you want to see only a limited number of results. 'limit' allows you do this:

```
input2 = load 'daily' as (exchanges, stocks);
```

```
first10 = limit input2 10;
```

## 22. What do you mean by a bag in Pig?

Collection of tuples is referred as a bag in Apache Pig

## 23. Does Pig support multi-line commands?

Yes

## 24. What are different modes of execution in Apache Pig?

Apache Pig runs in 2 modes- one is the "Pig (Local Mode) Command Mode" and the other is the "Hadoop MapReduce (Java) Command Mode". Local



Mode requires access to only a single machine where all files are installed and executed on a local host whereas MapReduce requires accessing the Hadoop cluster.

### **25. 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 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.

### **26. Explain about co-group in Pig.**

COGROUP operator in Pig is used to work with multiple tuples. COGROUP operator is applied on statements that contain or involve two or more relations. The COGROUP operator can be applied on up to 127 relations at a time. When using the COGROUP operator on two tables at once-Pig first groups both the tables and after that joins the two tables on the grouped columns.

### **27. 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.

### **28. 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.





### **29. 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

### **30. 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.

### **31. What does Flatten do in Pig?**

Sometimes there is data in a tuple or 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.

### **32. What is Dataflow language?**

To access the external data, every language must follow many rules and regulations. The instructions are flowing through data by executing different control statements, but data doesn't get moved. Dataflow language can get a stream of data which passes from one instruction to another instruction to be processed. Pig can easily process those conditions, jumps, loops and process the data in efficient manner.

### **33. Can you define Pig in 2 lines?**



Pig is a platform to analyze large data sets that should either structured or unstructured data by using Pig latin scripting. Intentionally done for streaming data, unstructured data in parallel.

#### **34. What are the main difference between local mode and MapReduce mode?**

Local mode: No need to start or install Hadoop. The pig scripts run in the local system. By default Pig store data in File system. 100% MapReduce and Local mode commands everything same, no need to change anything.

MapReduce Mode: It's mandatory to start Hadoop. Pig scripts run and stored in in HDFS. in Both modes, Java and Pig installation is mandatory.

#### **35. What is the difference between Store and dump commands?**

Dump command after process the data displayed on the terminal, but it's not stored anywhere. Where as Store stored in local file system or HDFS and output execute in a folder. In the protection environment most often hadoop developer used 'store' command to store data in in the HDFS.

#### **36. What is the relation between map, tuple and bag?**

Bag: collection of touples is called bag. It hold entire touples and maps data, we represent bags with {}

Tuple: collection of map called fields. It's fixed length and have multiple fields in (tuple). The fields in a tuple can be any data type, including the complex data types: bags, tuples, and maps.

map: collection of data element that mapping where element have pig data types. Most often map can ease unstructured data's datatype.

#### **37. What are the relational operations in Pig?**

for each — to iterate and loop all date into an object.

order by — sort the data in ascending order or descending order.



filters – It's similar to where command in SQL. It filter the data to process.

group: grouping the data to get desired output.

distinct: Displays only unique values, but it's works on entire records, but not individual fields.

join: logically join many tables and get desired output.

limit: It not use MapReduce, just filter and display limited data info only.

### **38. What is Pig Engine importance?**

It's acts as interpreter between Pig Latin script and MapReduce Jobs. It creating environment to execute Pig scripts into series of mapreduce jobs in parallel manner.

### **39. Why Pig instead of Mapreduce?**

Compare with MapReduce many features available in Apache Pig.

In Mapreduce it's too difficult to join multiple data sets. Development cycle is very long.

Depends on the task, Pig automatically converts code into Map or Reduces. Easy to join multiple tables and run many sql queries like Join, filter, group by, order by , union and many more.

### **40. Can you tell me little bit about Hive and Pig?**

Pig internally use Pig Latin, it's procedural language. Schema is optional, no meta store concept. where as Hive use a database to store meta store.

Hive internally use special language called HQL, it's subset of SQL. Schema is mandatory to process. Hive intentionally done for Queries.

But both Pig and Hive run on top of MapReduce and convert internal commands into MapReduce jobs. Both used to analyze the data and eventually generate same output.

### **41. What is Flatten does in Pig?**



Syntactically flatten similar to UDF, but it's powerful than UDFs. The main aim of Flatten is change the structure of tuple and bags, UDFs can't do it. Flatten can unnest the Tuple and bags, it's opposite to "Tobag" and "ToTuple".

#### **42. Can we process vast amount of data in local mode? Why?**

No, System has limited fixed amount of storage, where as Hadoop can handle vast amount of data. So Pig -x

Mapreduce mode is the best choice to process vast amount of data.

#### **43. How Pig integrate with Mapreduce to process data?**

Pig can easier to execute. When programmer wrote a script to analyze the data sets, Here Pig compiler will convert the programs into MapReduce understandable format. Pig engine execute the query on the MR Jobs. The Mapreduce process the data and generate output report. Here Mapreduce doesn't return output to Pig, directly stored in the HDFS.

#### **44. How to debugging in Pig?**

Describe: Review the schema.

Explain: logical, Physical and MapReduce execution plans.

Illustrate: Step by step execution of the each step execute in this operator.

These commands used to debugging the pig latin script.

#### **45. Tell me few important operators while working with Data in Pig.**

Filter: Working with Tuples and rows to filter the data.

Foreach: Working with Columns of data to load data into columns.

Group: Group the data in single relation.

Cogroup & Join: To group/Join data in multiple relations.

Union: Merge the data of multiple relations.

Split: partition the content into multiple relations.



#### 46. What is Topology Script?

Topology scripts are used by Hadoop to determine the rack location of nodes. Its trigger to replicate the data. As a part of rack awareness, Hadoop by default configured in `topology.script.file.name`. If not set, the rack id is returned for any passed IP address.

#### 47. Hive doesn't support multi-line commands, what about Pig?

Pig can support single and multiple line commands.

Single line comments:

Dump B; — It execute the data, but not store in the file system.

Multiple Line comments:

Store B into '/output'; /\* it can store/persists the data in Hdfs or Local File System.

In protection level most often used Store command \*/

#### 48. Can you tell me important data types in Pig?

Primitive datatypes: Int, Long, float, double, arrays, chararray, byte array.

Complex datatypes: Touple, bag, map

#### 49. What is BloomMapFile used for?

The *BloomMapFile* is a class that extends *MapFile*. So its functionality is similar to *MapFile*. *BloomMapFile* uses dynamic Bloom filters to provide quick membership test for the keys. It is used in Hbase table format.

#### 50. What is the difference between logical and physical plans?

Pig undergoes some steps when a Pig Latin Script is converted into MapReduce jobs. After performing the basic parsing and semantic checking, it produces a logical plan. The *logical plan* describes the logical operators that have to be executed by Pig during execution. After this, Pig produces a physical plan. The *physical plan* describes the physical operators that are needed to execute the script.



### **51. Does 'ILLUSTRATE' run MR job?**

No, illustrate will not pull any MR, it will pull the internal data. On the console, illustrate will not do any job. It just shows output of each stage and not the final output.

### **52. Is the keyword 'DEFINE' like a function name?**

Yes, the keyword 'DEFINE' is like a function name. Once you have registered, you have to define it. Whatever logic you have written in Java program, you have an exported jar and also a jar registered by you. Now the compiler will check the function in exported jar. When the function is not present in the library, it looks into your jar.

### **53. Is the keyword 'FUNCTIONAL' a User Defined Function (UDF)?**

No, the keyword 'FUNCTIONAL' is not a User Defined Function (UDF). While using UDF, we have to override some functions. Certainly you have to do your job with the help of these functions only. But the keyword 'FUNCTIONAL' is a built-in function i.e a pre-defined function, therefore it does not work as a UDF.

### **54. Why do we need MapReduce during Pig programming?**

Pig is a high-level platform that makes many Hadoop data analysis issues easier to execute. The language we use for this platform is: *Pig Latin*. A program written in *Pig Latin* is like a query written in SQL, where we need an execution engine to execute the query. So, when a program is written in Pig Latin, Pig compiler will convert the program into MapReduce jobs. Here, *MapReduce* acts as the execution engine.

### **55. Are there any problems which can only be solved by MapReduce and cannot be solved by PIG? In which kind of scenarios MR jobs will be more useful than PIG?**

Let us take a scenario where we want to count the population in two cities. I have a data set and sensor list of different cities. I want to count the population by using one mapreduce for two cities. Let us assume that one is Bangalore and the other is Noida. So I need to consider key of Bangalore city similar to Noida through which I can bring the population data of these two cities to one reducer. The idea behind this is some how I have to instruct map reducer program – whenever you find city with the name '*Bangalore*' and city



with the name 'Noida', you create the alias name which will be the common name for these two cities so that you create a common key for both the cities and it get passed to the same reducer. For this, we have to write *custom partitioner*.

In mapreduce when you create a 'key' for city, you have to consider 'city' as the key. So, whenever the framework comes across a different city, it considers it as a different key. Hence, we need to use customized partitioner. There is a provision in mapreduce only, where you can write your custom partitioner and mention if city = bangalore or noida then pass similar hashcode. However, we cannot create custom partitioner in Pig. As Pig is not a framework, we cannot direct execution engine to customize the partitioner. In such scenarios, MapReduce works better than Pig.

**56. Does Pig give any warning when there is a type mismatch or missing field?**

No, Pig will not show any warning if there is no matching field or a mismatch. If you assume that Pig gives such a warning, then it is difficult to find in log file. If any mismatch is found, it assumes a null value in Pig.

**57. Can we say cogroup is a group of more than 1 data set?**

Cogroup is a group of one data set. But in the case of more than one data sets, cogroup will group all the data sets and join them based on the common field. Hence, we can say that cogroup is a *group* of more than one data set and *join* of that data set as well.

**58. What does FOREACH do?**

FOREACH is used to apply transformations to the data and to generate new data items. The name itself is indicating that for each element of a data bag, the respective action will be performed.

*Syntax* : FOREACH bagname GENERATE expression1, expression2, .....  
The meaning of this statement is that the expressions mentioned after GENERATE will be applied to the current record of the data bag.

**59. What is bag?**

A bag is one of the data models present in Pig. It is an unordered collection of tuples with possible duplicates. Bags are used to store collections while



grouping. The size of bag is the size of the local disk, this means that the size of the bag is limited. When the bag is full, then Pig will spill this bag into local disk and keep only some parts of the bag in memory. There is no necessity that the complete bag should fit into memory. We represent bags with “{}”.

## 60. Why Pig ?

- i) Ease of programming
- ii) Optimization opportunities.
- iii) Extensibility

### **i) Ease of programming :-**

It is trivial to achieve parallel execution of simple, “embarrassingly parallel” data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.

### **ii) Optimization opportunities :-**

The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.

### **iii) Extensibility :-**

Users can create their own functions to do special-purpose processing.

## 61. Advantages of Using Pig ?

- i) Pig can be treated as a higher level language
  - a) Increases Programming Productivity
  - b) Decreases duplication of Effort
  - c) Opens the M/R Programming system to more uses





ii) Pig Insulates against hadoop complexity

a) Hadoop version Upgrades

b) Job configuration Tuning

## **62. Pig Features ?**

i) Data Flow Language

User Specifies a Sequence of Steps where each step specifies only a single high-level data transformation.

ii) User Defined Functions (UDF)

iii) Debugging Environment

iv) Nested data Model

## **63. What is Pig Latin?**

Pig Latin is a data flow Scripting Language like Perl for exploring large data sets. A Pig Latin program is made up of a series of operations, or transformations, that are applied to the input data to produce output.

## **64. What is Pig Engine?**

Pig Engine is an execution environment to run Pig Latin programs. It converts these Pig Latin operators or transformations into a series of MapReduce jobs.

## **65. What are the modes of Pig Execution?**

Pig execution can be done in two modes.

Local Mode: Local execution in a single JVM, all files are installed and run using local host and file system.

Mapreduce Mode: Distributed execution on a Hadoop cluster, it is the default mode.

## **66. What are the Pig Latin Features?**



Pig Latin script is made up of a series of operations, or transformations, that are applied to the input data to produce output

Pig Latin programs can be executed either in Interactive mode through Grunt shellor in Batch mode via Pig Latin Scripts.

Pig Latin includes operators for many of the traditional data operations (join, sort, filter, etc.)

User Defined Functions (UDF)

Debugging Environment

### **67. What are the advantages of using Pig over Mapreduce?**

In Mapreduce,

Development cycle is very long. Writing mappers and reducers, compiling and packaging the code, submitting jobs, and retrieving the results is a time Consuming process.

Performing Data set joins is very difficult

Low level and rigid, and leads to a great deal of custom user code that is hard to maintain and reuse is complex.

In pig,

No need of compiling or packaging of code. Pig operators will be converted into map or reduce tasks internally.

Pig Latin provides all of the standard data-processing operations, such as join, filter, group by, order by, union, etc

high level of abstraction for processing large data sets

### **68. What is the difference between Pig Latin and HiveQL ?**

Pig Latin:

Pig Latin is a Procedural language



Nested relational data model

Schema is optional

HiveQL:

HiveQL is Declarative

HiveQL flat relational

Schema is required

### **69. What are the common features in Pig and Hive?**

Both provide high level abstraction on top of Mapreduce

Both convert their commands internally into Mapreduce jobs

Both doesn't support low-latency queries and thus OLAP or OLTP are not supported

### **70. What is the difference between logical and physical plans?**

Pig undergoes some steps when a Pig Latin Script is converted into MapReduce jobs. After performing the basic parsing and semantic checking, it produces a logical plan. The logical plan describes the logical operators that have to be executed by Pig during execution. After this, Pig produces a physical plan. The physical plan describes the physical operators that are needed to execute the script.

### **71. Why do we need MapReduce during Pig programming?**

Pig is a high-level platform that makes many Hadoop data analysis issues easier to execute. A program written in Pig Latin is like a query written in SQL, where we need an execution engine to execute the query. So, Pig engine will convert the program into MapReduce jobs. Here, MapReduce acts as the execution engine.

### **72. How many ways we can run Pig programs?**

Pig programs or commands can be executed in three ways.



Script – Batch Method

Grunt Shell – Interactive Method

Embedded mode

All these ways can be applied to both Local and Mapreduce modes of execution.

### **73. What is Grunt in Pig?**

Grunt is an Interactive Shell in Pig, and below are its major features:

Ctrl-E key combination will move the cursor to the end of the line.

Grunt remembers command history, and can recall lines in the history buffer using up or down cursor keys.

Grunt supports Auto completion mechanism, which will try to complete

Pig Latin keywords and functions when you press the Tab key

### **74. Is Pig Latin Case Sensitive?**

The names (aliases) of relations and fields are case sensitive. The names of Pig Latin functions are case sensitive. The names of parameters and all other Pig Latin keywords are case insensitive.

### **75. What is the command for to enter into local mode?**

Pig -x local

### **76. What is the default load function to store data?**

Pigstorage

### **77. Where the intermediate data will store in Pig?**

HDFS

### **78. What is the command for getting help about Pig?**

Pig -help



**79. How to validate the pig.jar by running a unit test?**

Ant Test

**80. What is the shell name to write Pig commands?**

Grunt shell

**81. What is the shortcuts for DUMP and ILLUSTRATE operators?**

\d – DUMP

\I – ILLUSTRATE

**82. Where do we store tuples on Pig?**

()

**83. How to refer fields in Pig?**

By both names and positional notations

**84. How to specify positional notation in Pig?**

By \$

**85. When should you use Pig Latin and when should you use Hive?**

When data is purely structured, we should go for Hive.

In case of semi-structured data, we can consider Pig.

**86. What do you mean by unordered collection in a bag or in a relation?**

Relations are unordered means there is no guarantee that tuples are processed in any particular order. Furthermore, processing may be parallelized in which cases are not processed according to any total ordering.

**87. What is a relation in Pig?**

A Pig relation is a bag of tuples. A Pig relation is similar to a table in a relational database, where the tuples in the bag corresponds to the rows in a table. Unlike a relational table, however, Pig relations don't require that every



tuple contain the same number of fields or that the fields in the same position (column) have the same type.

**88. What is a tuple?**

A tuple is an ordered set of fields and a field is a piece of data.