




 Home	Blogs <a href="https://www.edureka.co/blog/all/">(https://www.edureka.co/blog/all/)</a>	Videos <a href="https://www.edureka.co/blog/videos/">(https://www.edureka.co/blog/videos/)</a>	Interview Questions <a href="https://www.edureka.co/blog/interview-questions/">(https://www.edureka.co/blog/interview-questions/)</a>
<h1>Pig Tutorial: Apache Pig Architecture &amp; Twitter Case Study</h1> <p>📌 Recommended by 297 users</p> <div> <span>Shubham Sinha (<a href="https://www.edureka.co/blog/author/shubham-sinha/">https://www.edureka.co/blog/author/shubham-sinha/</a>)</span> <span>  Jul 16, 2018</span> <span>  <a href="https://www.edureka.co/blog/pig-tutorial/">https://www.edureka.co/blog/pig-tutorial/</a></span> </div> <div>     </div> <div> <a href="#">How can I share this article?</a>  <a href="https://www.edureka.co/blog/pig-tutorial/">u=https://www.edureka.co/blog/pig-tutorial/</a> </div>			



As we mentioned in our **Hadoop Ecosystem** (<https://www.edureka.co/blog/hadoop-ecosystem>) blog, Apache Pig is an essential part of our Hadoop ecosystem. So, I would like to take you through this Apache Pig tutorial, which is a part of our **Hadoop Tutorial Series** (<https://www.edureka.co/blog/hadoop-tutorial/>). In this Apache Pig Tutorial blog, I will talk about:

- Before starting with the Apache Pig tutorial, I would like you to ask yourself a question – “*while MapReduce was there for Big Data Analytics why Apache Pig came into picture?*”

The sweet and simple answer to this is:

***approximately 10 lines of Pig code is equal to 200 lines of MapReduce code.***

Writing MapReduce jobs in Java is not an easy task for everyone. If you want a taste of MapReduce Java code, [click here \(https://www.edureka.co/blog/mapreduce-example-reduce-side-join/\)](https://www.edureka.co/blog/mapreduce-example-reduce-side-join/) and you will understand the complexities. Thus, Apache Pig emerged as a boon for programmers who were not good with Java or Python. Even if someone who knows Java and is good with MapReduce, they will also prefer Apache Pig due to the ease working with Pig. Let us take a look now.

Programmers face difficulty writing MapReduce tasks as it requires Java or Python programming knowledge. For them, Apache Pig is a savior.

- Pig Latin is a high-level data flow language, whereas MapReduce is a low-level data processing paradigm.
- Without writing complex Java implementations in MapReduce, programmers can achieve the same implementations very easily using Pig Latin.
- Apache Pig uses multi-query approach (i.e. using a single query of Pig Latin we can accomplish multiple MapReduce tasks), which reduces the length of the code by 20 times. Hence, this reduces the development period by almost 16 times.
- Pig provides many built-in operators to support data operations like joins, filters, ordering, sorting etc. Whereas to perform the same function in MapReduce is a humongous task.
- Performing a Join operation in Apache Pig is simple. Whereas it is difficult in MapReduce to perform a Join operation between the data sets, as it requires multiple MapReduce tasks to be executed sequentially to fulfill the job.
- In addition, it also provides nested data types like tuples, bags, and maps that are missing from MapReduce. I will explain you these data types in a while.

Now that we know why Apache Pig came into the picture, you would be curious to know what is Apache Pig? Let us move ahead in this Apache Pig tutorial blog and go through the introduction and features of Apache Pig.

Apache Pig is a platform, used to analyze large data sets representing them as data flows. It is designed to provide an abstraction over MapReduce, reducing the complexities of writing a MapReduce program. We can perform data manipulation operations very easily in Hadoop using Apache Pig.

The features of Apache pig are:

- Pig enables programmers to write complex data transformations without knowing Java.
- Apache Pig has two main components – the *Pig Latin language* and the *Pig Run-time Environment*, in which Pig Latin programs are executed.
- For Big Data Analytics, Pig gives a simple data flow language known as **Pig Latin** which has functionalities similar to SQL like join, filter, limit etc.
- Developers who are working with scripting languages and SQL, leverages Pig Latin. This gives developers **ease of programming** with Apache Pig. Pig Latin provides various built-in operators like join, sort, filter, etc to read, write, and process large data sets. Thus it is evident, Pig has a **rich set of operators**.
- Programmers write scripts using Pig Latin to analyze data and these scripts are internally converted to Map and Reduce tasks by Pig MapReduce Engine. Before Pig, writing MapReduce tasks was the only way to process the data stored in HDFS.
- If a programmer wants to write custom functions which is unavailable in Pig, Pig allows them to write User Defined Functions (**UDF**) in any language of their choice.

## Announcing the Ridiculously Committed Mentor Award

[Nominate Now](#)

- Pig can process any kind of data, not structured, semi structured or unstructured data, coming from various sources. Apache Pig handles all kinds of data.
- Approximately, *10 lines of pig code is equal to 200 lines of MapReduce code.*
- It can handle inconsistent schema (in case of unstructured data).
- Apache Pig extracts the data, performs operations on that data and dumps the data in the required format in HDFS i.e. **ETL (Extract Transform Load)**.
- Apache Pig automatically optimizes the tasks before execution, i.e. **automatic optimization**.
- It allows programmers and developers to concentrate upon the whole operation irrespective of creating mapper and reducer functions separately.

After knowing what is Apache Pig, now let us understand where we can use Apache Pig and what are the use cases which suits Apache Pig the most?

### Apache Pig Tutorial: Where to use Apache Pig?

Apache Pig is used for analyzing and performing tasks involving ad-hoc processing. Apache Pig is used:

- Where we need to process, huge data sets like Web logs, streaming online data, etc.
- Where we need Data processing for search platforms (different types of data needs to be processed) like **Yahoo uses Pig for 40% of their jobs including news feeds and search engine.**
- Where we need to process time sensitive data loads. Here, data needs to be extracted and analyzed quickly. E.g. machine learning algorithms requires time sensitive data loads, like twitter needs to quickly extract data of customer activities (i.e. tweets, re-tweets and likes) and analyze the data to find patterns in customer behaviors, and make recommendations immediately like trending tweets.

Now, in our Apache Pig Tutorial, let us go through the **Twitter case study** to better understand how Apache Pig helps in analyzing data and makes business understanding easier.

Learn Apache Pig from Experts

(<https://www.edureka.co/big-data-and-hadoop>)

### Apache Pig Tutorial: Twitter Case Study

I will take you through a case study of Twitter where Twitter adopted Apache Pig.

Twitter's data was growing at an accelerating rate (i.e. 10 TB data/day). Thus, Twitter decided to move the archived data to HDFS and adopt Hadoop for extracting the business values out of it.

Their major aim was to analyse data stored in Hadoop to come up with the following insights on a daily, weekly or monthly basis.

#### Counting operations:

- How many requests twitter serve in a day?
- What is the average latency of the requests?
- How many searches happens each day on Twitter?
- How many unique queries are received?
- How many unique users come to visit?
- What is the geographic distribution of the users?

#### Correlating Big Data:

- How usage differs for mobile users?
- Cohort analysis: analyzing data by categorizing user, based on their behavior.
- What goes wrong while site problem occurs?
- Which features user often uses?
- Search correction and search suggestions.

#### Research on Big Data & produce better outcomes like:

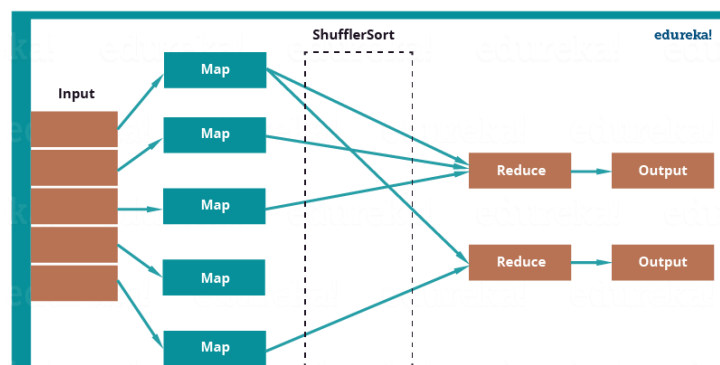
- What can Twitter analysis about users from their tweets?
- Who follows whom and on what basis?
- What is the ratio of the follower to following?
- What is the reputation of the user?

and many more...

So, for analyzing data, Twitter used MapReduce initially, which is a parallel computing over HDFS (i.e. Hadoop Distributed File system).

For example, they wanted to analyse how many tweets are stored per user, in the given tweet table?

Using MapReduce, this problem will be solved sequentially as shown in the below image:



## Announcing the Ridiculously Committed Mentor Award

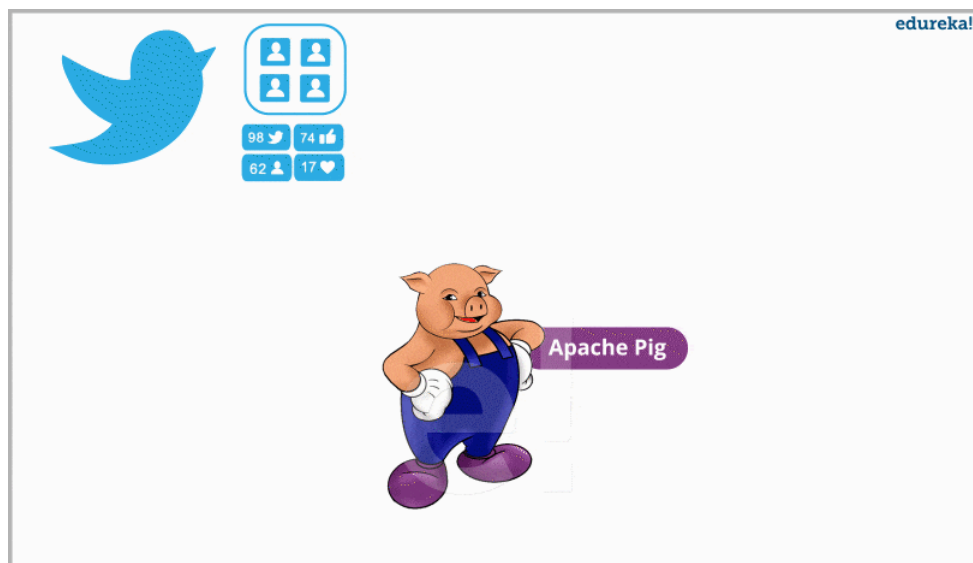
Nominate Now

MapReduce program first inputs the key as rows and sends the tweet table information to mapper function. Then the Mapper function will select the user id and associate unit value (i.e. 1) to every user id. The Shuffle function will sort same user ids together. At last, Reduce function will add all the number of tweets together belonging to same user. The output will be user id, combined with user name and the number of tweets per user.

But while using MapReduce, they faced some limitations:

- Analysis needs to be typically done in Java.
- Joins, that are performed, needs to be written in Java, which makes it longer and more error-prone.
- For projection and filters, custom code needs to be written which makes the whole process slower.
- The job is divided into many stages while using MapReduce, which makes it difficult to manage.

So, Twitter moved to Apache Pig for analysis. Now, joining data sets, grouping them, sorting them and retrieving data becomes easier and simpler. You can see in the below image how twitter used Apache Pig to analyse their large data set.



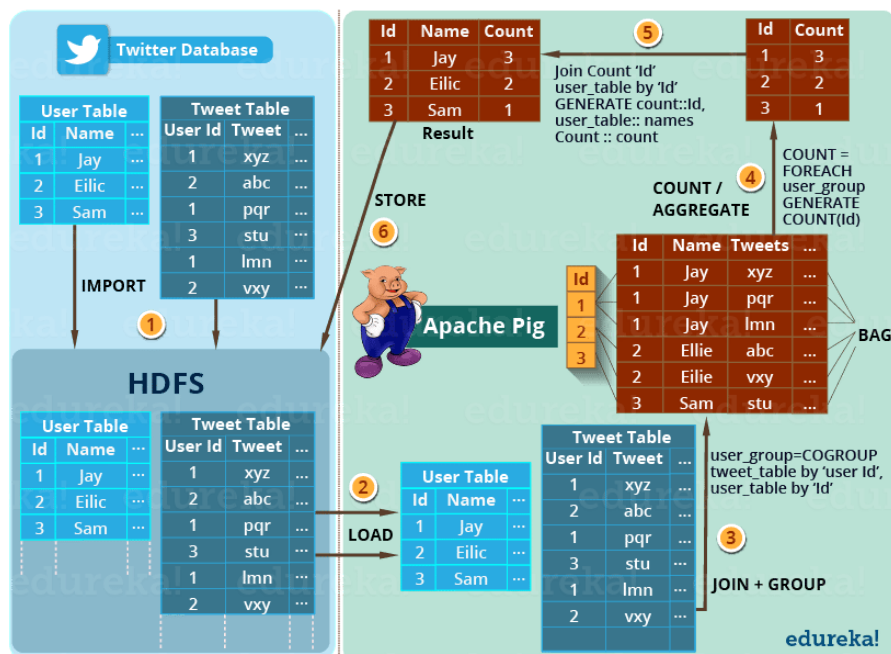
Twitter had both semi-structured data like Twitter Apache logs, Twitter search logs, Twitter MySQL query logs, application logs and structured data like tweets, users, block notifications, phones, favorites, saved searches, re-tweets, authentications, SMS usage, user followings, etc. which can be easily processed by Apache Pig.

Twitter dumps all its archived data on HDFS. It has two tables i.e. user data and tweets data. User data contains information about the users like username, followers, followings, number of tweets etc. While Tweet data contains tweet, its owner, number of re-tweets, number of likes etc. Now, twitter uses this data to analyse their customer's behaviors and improve their past experiences.

We will see how Apache Pig solves the same problem which was solved by MapReduce:

**Question:** Analyzing how many tweets are stored per user, in the given tweet tables?

The below image shows the approach of Apache Pig to solve the problem:



The step by step solution of this problem is shown in the above image.

## Announcing the Ridiculously Committed Mentor Award

[Nominate Now](#)

**STEP 2**– Then Apache Pig loads (**LOAD**) the tables into Apache Pig framework.

**STEP 3**– Then it joins and groups the tweet tables and user table using **COGROUP** command as shown in the above image.

This results in the inner Bag Data type, which we will discuss later in this blog.

Example of Inner bags produced (refer to the above image) –

(1,{(1,Jay,xyz),(1,Jay,pqr),(1,Jay,lmn)})

(2,{(2,Elle,abc),(2,Elle,vxy)})

(3, {(3,Sam,stu)})

**STEP 4**– Then the tweets are counted according to the users using **COUNT** command. So, that the total number of tweets per user can be easily calculated.

Example of tuple produced as (id, tweet count) (refer to the above image) –

(1, 3)

(2, 2)

(3, 1)

**STEP 5**– At last the result is joined with user table to extract the user name with produced result.

Example of tuple produced as (id, name, tweet count) (refer to the above image) –

(1, Jay, 3)

(2, Elle, 2)

(3, Sam, 1)

**STEP 6**– Finally, this result is stored back in the HDFS.

Pig is not only limited to this operation. It can perform various other operations which I mentioned earlier in this use case.

These insights helps Twitter to perform sentiment analysis and develop machine learning algorithms based on the user behaviors and patterns.

Master Apache Pig in our Hadoop Course

(<https://www.edureka.co/big-data-and-hadoop>)

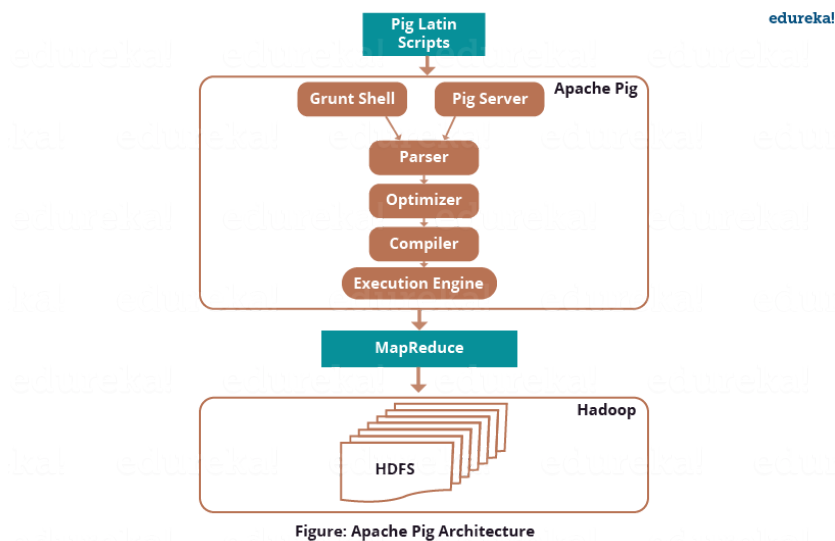
Now, after knowing the the Twitter case study, in this Apache Pig tutorial, let us take a deep dive and understand the architecture of Apache Pig and Pig Latin's data model. This will help us understand how pig works internally. Apache Pig draws its strength from its architecture.

Meanwhile you may check out the video given below where all the concepts related to Pig has been discussed:

**Pig Tutorial | Apache Pig Script | Hadoop Pig Tutorial | Edureka**

## Apache Pig Tutorial: Architecture

For writing a Pig script, we need Pig Latin language and to execute them, we need an execution environment. The architecture of Apache Pig is shown in the below image.



### Pig Latin Scripts

Initially as illustrated in the above image, we submit Pig scripts to the Apache Pig execution environment which can be written in Pig Latin using built-in operators.

[https://www.edureka.co/blog/pig-tutorial/?utm\\_source=hd&utm\\_campaign=lms\\_hd\\_120616&utm\\_medium=lms](https://www.edureka.co/blog/pig-tutorial/?utm_source=hd&utm_campaign=lms_hd_120616&utm_medium=lms)

## Announcing the Ridiculously Committed Mentor Award

[Nominate Now](#)

- **Grunt Shell:** This is Pig's interactive shell provided to execute all Pig Scripts.
- **Script File:** Write all the Pig commands in a script file and execute the Pig script file. This is executed by the Pig Server.
- **Embedded Script:** If some functions are unavailable in built-in operators, we can programmatically create User Defined Functions to bring that functionalities using other languages like Java, Python, Ruby, etc. and embed it in Pig Latin Script file. Then, execute that script file.

**Parser**

From the above image you can see, after passing through Grunt or Pig Server, Pig Scripts are passed to the Parser. The Parser does type checking and checks the syntax of the script. The parser outputs a DAG (directed acyclic graph). DAG represents the Pig Latin statements and logical operators. The logical operators are represented as the nodes and the data flows are represented as edges.

**Optimizer**

Then the DAG is submitted to the optimizer. The Optimizer performs the optimization activities like split, merge, transform, and reorder operators etc. This optimizer provides the automatic optimization feature to Apache Pig. The optimizer basically aims to reduce the amount of data in the pipeline at any instance of time while processing the extracted data, and for that it performs functions like:

- **PushUpFilter:** If there are multiple conditions in the filter and the filter can be split, Pig splits the conditions and pushes up each condition separately. Selecting these conditions earlier, helps in reducing the number of records remaining in the pipeline.
- **PushDownForEachFlatten:** Applying flatten, which produces a cross product between a complex type such as a tuple or a bag and the other fields in the record, as late as possible in the plan. This keeps the number of records low in the pipeline.
- **ColumnPruner:** Omitting columns that are never used or no longer needed, reducing the size of the record. This can be applied after each operator, so that fields can be pruned as aggressively as possible.
- **MapKeyPruner:** Omitting map keys that are never used, reducing the size of the record.
- **LimitOptimizer:** If the limit operator is immediately applied after a load or sort operator, Pig converts the load or sort operator into a limit-sensitive implementation, which does not require processing the whole data set. Applying the limit earlier, reduces the number of records.

This is just a flavor of the optimization process. Over that it also performs **Join**, **Order By** and **Group By** functions.

To shutdown, automatic optimization, you can execute this command:

```
pig -optimizer_off [opt_rule | all ]
```

**Compiler**

After the optimization process, the compiler compiles the optimized code into a series of MapReduce jobs. The compiler is the one who is responsible for converting Pig jobs automatically into MapReduce jobs.

**Execution engine**

Finally, as shown in the figure, these MapReduce jobs are submitted for execution to the execution engine. Then the MapReduce jobs are executed and gives the required result. The result can be displayed on the screen using "**DUMP**" statement and can be stored in the HDFS using "**STORE**" statement.

After understanding the Architecture, now in this Apache Pig tutorial, I will explain you the Pig Latin's Data Model.

**Apache Pig Tutorial: Pig Latin Data Model**

The data model of Pig Latin enables Pig to handle all types of data. Pig Latin can handle both atomic data types like int, float, long, double etc. and complex data types like tuple, bag and map. I will explain them individually. The below image shows the data types and their corresponding classes using which we can implement them:

Pig Data Type	Implementing Class
<b>Bag</b>	org.apache.pig.data.DataBag
<b>Tuple</b>	org.apache.pig.data.Tuple
<b>Map</b>	java.util.Map<Object, Object>
<b>Integer</b>	java.lang.Integer
<b>Long</b>	java.lang.Long
<b>Float</b>	java.lang.Float
<b>Double</b>	java.lang.Double
<b>Chararray</b>	java.lang.String
<b>Bytearray</b>	byte[]

**Atomic /Scalar Data type**

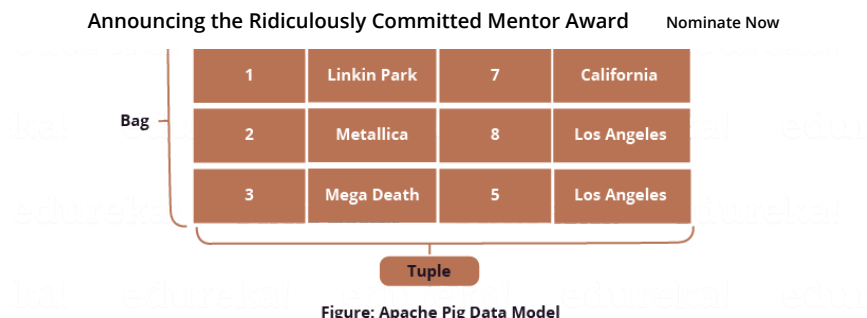
Atomic or scalar data types are the basic data types which are used in all the languages like string, int, float, long, double, char[], byte[]. These are also called the primitive data types. The value of each cell in a field (column) is an atomic data type as shown in the below image.

For fields, positional indexes are generated by the system automatically (also known as **positional notation**), which is represented by '\$' and it starts from \$0, and grows \$1, \$2, so on... As compared with the below image \$0 = S.No., \$1 = Bands, \$2 = Members, \$3 = Origin.

Scalar data types are – '1', 'Linkin Park', '7', 'California' etc.



edureka!



Now we will talk about **complex data types** in Pig Latin i.e. Tuple, Bag and Map.

### Tuple

Tuple is an ordered set of fields which may contain different data types for each field. You can understand it as the records stored in a row in a relational database. A Tuple is a set of cells from a single row as shown in the above image. The elements inside a tuple does not necessarily need to have a schema attached to it.

A tuple is represented by '()' symbol.

Example of tuple – (1, Linkin Park, 7, California)

Since tuples are ordered, we can access fields in each tuple using indexes of the fields, like \$1 from above tuple will return a value 'Linkin Park'. You can notice that above tuple doesn't have any schema attached to it.

### Bag

A bag is a collection of a set of tuples and these tuples are subset of rows or entire rows of a table. A bag can contain duplicate tuples, and it is not mandatory that they need to be unique.

The bag has a flexible schema i.e. tuples within the bag can have different number of fields. A bag can also have tuples with different data types.

A bag is represented by '{}' symbol.

Example of a bag – {(Linkin Park, 7, California), (Metallica, 8), (Mega Death, Los Angeles)}

But for Apache Pig to effectively process bags, the fields and their respective data types need to be in the same sequence.

Set of bags –

{(Linkin Park, 7, California), (Metallica, 8), (Mega Death, Los Angeles)},

{(Metallica, 8, Los Angeles), (Mega Death, 8), (Linkin Park, California)}

There are two types of Bag, i.e. **Outer Bag** or relations and **Inner Bag**.

Outer bag or relation is nothing but a bag of tuples. Here relations are similar as relations in relational databases. To understand it better let us take an example:

{(Linkin Park, California), (Metallica, Los Angeles), (Mega Death, Los Angeles)}

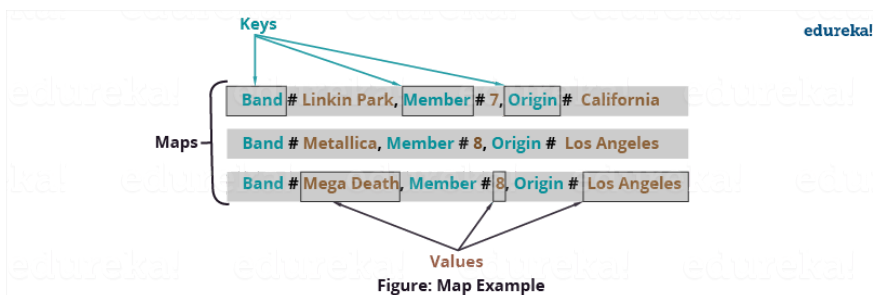
This above bag explains the relation between the *Band* and their place of *Origin*.

On the other hand, an inner bag contains a bag inside a tuple. For Example, if we sort *Band* tuples based on *Band's Origin*, we will get:

(Los Angeles, {(Metallica, Los Angeles), (Mega Death, Los Angeles)})

(California, {(Linkin Park, California)})

Here, first field type is a string while the second field type is a bag, which is an inner bag within a tuple.



### Map

A map is key-value pairs used to represent data elements. The key must be a chararray [] and should be unique like column name, so it can be indexed and value associated with it can be accessed on basis of the keys. The value can be of any data type.

Maps are represented by '[]' symbol and key-value are separated by '#' symbol, as you can see in the above image.

Example of maps– [band#Linkin Park, members#7 ], [band#Metallica, members#8 ]

## Announcing the Ridiculously Committed Mentor Award

[Nominate Now](#)

## Apache Pig Tutorial: Schema

Schema assigns name to the field and declares data type of the field. Schema is optional in Pig Latin but Pig encourage you to use them whenever possible, as the error checking becomes efficient while parsing the script which results in efficient execution of program. Schema can be declared as both simple and complex data types. During LOAD function, if the schema is declared it is also attached with the data.

Few Points on Schema in Pig:

- If the schema only includes the field name, the data type of field is considered as byte array.
- If you assign a name to the field you can access the field by both, the field name and the positional notation. Whereas if field name is missing we can only access it by the positional notation i.e. \$ followed by the index number.
- If you perform any operation which is a combination of relations (like JOIN, COGROUP, etc.) and if any of the relation is missing schema, the resulting relation will have null schema.
- If the schema is null, Pig will consider it as byte array and the real data type of field will be determined dynamically.

I hope this Apache Pig tutorial blog is informative and you liked it. In this blog, you got to know the basics of Apache Pig, its data model and its architecture. The Twitter case study would have helped you to connect better. In my next blog of **Hadoop Tutorial Series** (<https://www.edureka.co/blog/hadoop-tutorial/>), we will be covering the **installation of Apache Pig** (<https://www.edureka.co/blog/apache-pig-installation>), so that you can get your hands dirty while working practically on Pig and executing Pig Latin commands.

Master Apache Pig in our Hadoop Course

(<https://www.edureka.co/big-data-and-hadoop>)

Now that you have understood the Apache Pig Tutorial, check out the **Hadoop training** (<https://www.edureka.co/big-data-and-hadoop>) by Edureka, a trusted online learning company with a network of more than 250,000 satisfied learners spread across the globe. The Edureka Big Data Hadoop Certification Training course helps learners become expert in HDFS, Yarn, MapReduce, Pig, Hive, HBase, Oozie, Flume and Sqoop using real-time use cases on Retail, Social Media, Aviation, Tourism, Finance domain.

Got a question for us? Please mention it in the comments section and we will get back to you.



About Shubham Sinha (26 Posts (<https://www.edureka.co/blog/author/shubham-sinha/>))

Shubham Sinha is a Big Data and Hadoop expert working as a Research Analyst at Edureka. He is keen to work with Big Data related technologies such as Hadoop, Spark, Flink and Storm and web development technologies including Angular, Node.js & PHP.



Share on (<https://www.edureka.co/blog/hadoop-tutorial/>) (<https://www.edureka.co/blog/hadoop-tutorial/>) (<https://www.edureka.co/blog/hadoop-tutorial/>) (<https://www.edureka.co/blog/hadoop-tutorial/>)

◀ PREVIOUS

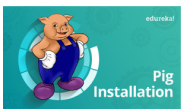
NEXT ▶

Got your brain cells running?  
Stay tuned to latest technology updates

Enter your Email Address

SUBSCRIBE

### Related Posts



Apache Pig Installation on Linux  
👁 18.3K

(<https://www.edureka.co/blog/apache-pig-installation>)



Hadoop Tutorial: All you need to know about Hadoop!  
👁 92.4K

(<https://www.edureka.co/blog/hadoop-tutorial/>)



Steps to Create UDF in Apache Pig  
👁 11.2K

(<https://www.edureka.co/blog/creating-udf-in-apache-pig/>)



Pig Programming: Create Your First Apache Pig Script  
👁 34K

(<https://www.edureka.co/blog/pig-programming-create-your-first-apache-pig-script/>)



Announcing the Ridiculously Committed Mentor Award

Nominate Now

- Big Data NoSQL (<https://www.edureka.co/blog/category/big-data-nosql/>)Blockchain (<https://www.edureka.co/blog/category/blockchain/>)
- Business Intelligence (<https://www.edureka.co/blog/category/business-intelligence/>)Cloud Computing (<https://www.edureka.co/blog/category/cloud-computing/>)
- Cyber Security (<https://www.edureka.co/blog/category/cyber-security/>)Deep Learning (<https://www.edureka.co/blog/category/deep-learning/>)
- Finance (<https://www.edureka.co/blog/category/finance/>)Frameworks (<https://www.edureka.co/blog/category/frameworks/>)
- Marketing (<https://www.edureka.co/blog/category/marketing/>)Mobile Development (<https://www.edureka.co/blog/category/mobile-development/>)
- Operations (<https://www.edureka.co/blog/category/operations/>)Programming (<https://www.edureka.co/blog/category/programming/>)
- Project Management (<https://www.edureka.co/blog/category/project-management/>)Robotic Process Automation (<https://www.edureka.co/blog/category/robotic-process-automation/>)
- Success Story (<https://www.edureka.co/blog/category/success-story/>)Systems & Architecture (<https://www.edureka.co/blog/category/systems-architecture/>)
- Systems Engineering (<https://www.edureka.co/blog/category/systems-engineering/>)Testing (<https://www.edureka.co/blog/category/testing/>)

Comments

2 Comments

2 Comments


<https://www.edureka.co/blog/>

 Rajiv Chaudhuri


 Recommend

 Share



Sort by Best





Join the discussion...

 Narayan Sallakonda • 10 months ago

Hello Shubham Sinha, I have a doubt. I need your assistance to get it fixed. How to run multiple pig scripts in oozie using single `job.properties` file and single workflow.xml file.

  • Reply • Share

 EdurekaSupport Mod  Narayan Sallakonda • 8 months ago

Workflow.xml file :


<workflow-app xmlns="uri:oozie:workflow:0.1" name="consolidation-wf">  
 <start to="pig-node"/>  
 <action name="pig-node">  
 <pig>  
 <job-tracker>\${jobtracker}</job-tracker>  
 <name-node>\${namenode}</name-node>  
 <prepare>  
 <delete path="\${outputdir1}"/>  
 <delete path="\${outputdir2}"/>  
 </prepare>  
 <configuration>  
 <property>  
 <name>mapred.job.queue.name</name>  
 <value>default</value>  
 </property>  
 <property>  
 <name>mapred.compress.map.output</name>

see more


1   • Reply • Share

- ALSO ON [HTTPS://WWW.EDUREKA.CO/BLOG/](https://www.edureka.co/blog/)
- Cybersecurity Tools You Must Know – Tools for Cyber Threats


1 comment • 2 months ago

 Gvsm Chaithanya — what are the industrial applications for information security
- AWS DevOps: Introduction to DevOps on AWS


4 comments • 7 months ago

 NAVEEN SAI — i am looking to learn Devops +Aws. So should i learn Aws Developer
- Real Time Big Data Applications in Various Domains

2 comments • 7 months ago

 Amber Anen — Yes, I totally agree with this. Industry influencers, academicians, and
- Capsule Neural Networks – Set of Nested Neural Layers

1 comment • 9 months ago

 Saurabh Jain — Hi Saurabh ,Would like to know about ..How to compare two images



Subscribe  
to our newsletter

Enter your Email Address

**SUBSCRIBE**

#### Related Blogs



Apache Pig Installation on Linux (<https://www.edureka.co/blog/apache-pig-installation>)

(<https://www.edureka.co/blog/apache-pig-installation>)



Hadoop Tutorial: All you need to know about Hadoop! (<https://www.edureka.co/blog/hadoop-tutorial/>)

(<https://www.edureka.co/blog/hadoop-tutorial/>)



Steps to Create UDF in Apache Pig (<https://www.edureka.co/blog/creating-udf-in-apache-pig/>)

(<https://www.edureka.co/blog/creating-udf-in-apache-pig/>)



Pig Programming: Create Your First Apache Pig Script (<https://www.edureka.co/blog/pig-programming-create-your-first-apache-pig-script/>)

(<https://www.edureka.co/blog/pig-programming-create-your-first-apache-pig-script/>)

# Get Live Training & Certification!



## Big Data HADOOP

**LEARN MORE**

(<http://www.edureka.co/big-data-and-hadoop>)

#### Big Data Analytics Courses



Big Data Hadoop Certification Training (</big-data-and-hadoop>)

(</big-data-and-hadoop>)

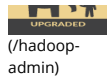


Apache Spark and Scala Certification Training (</apache-spark-scala-training>)

(</apache-spark-scala-training>)

Announcing the Ridiculously Committed Mentor Award

Nominate Now



(/hadoop-admin)

Scala, DevOps, Hadoop, AWS, Java, Certification, Training, (edureka)

Edureka	Work with us	Usefull Links	Follow us on
About us (https://www.edureka.co/about-us)	Careers (https://www.edureka.co/careers)	Reviews (https://www.edureka.co/reviews)	f t in v (https://facebook.com/edurekaIN)
News & Media (https://www.edureka.co/allmedia)	Become an Instructor (https://www.edureka.co/instructor)	Terms & conditions (https://www.edureka.co/terms-and-conditions)	
Contact us (https://www.edureka.co/contact-us)	Become an Affiliate (https://www.edureka.co/affiliate-program)	Privacy policy (https://www.edureka.co/privacy-policy)	<b>Learn on the GO!</b> <div></div> (https://itunes.apple.com/in/app/edureka/id1033145415?mt=8) <div></div> (https://play.google.com/store/apps/details?id=co.edureka.app)
Blog (https://www.edureka.co/blog/)	Hire from Edureka (https://www.edureka.co/hire-from-edureka)	Sitemap (https://www.edureka.co/sitemap)	
Community (https://www.edureka.co/community)			