Pig Latin is the scripting language used by Apache Pig to process and analyze large datasets.

It differs significantly from traditional programming languages like Java or Python, offering a more declarative and user-friendly approach to data manipulation.

Key Features

- Declarative: You specify what you want to achieve with your data, and Pig handles the how.
- High-level abstraction: Focus on the logic of your data processing without worrying about low-level details.
- Data flow-oriented: Describes a series of operations applied to data flowing from one step to the next.
- Expressive: Supports various data structures like bags, tuples, and maps, enabling complex data manipulations.
- Extensible: Users can define their own functions (UDFs) to extend the language's capabilities.

Structure of a Pig Latin Script

A Pig Latin script typically consists of three sections:

- DEFINE: Defines functions used within the script, including user-defined functions (UDFs).
- REGISTER: Registers data sources (e.g., files, tables) and assigns aliases for easier reference.
- OPERATIONS: Describes the data processing steps using Pig Latin operators like LOAD, FILTER, JOIN, GROUP, etc.

Basic Operators

- LOAD: Loads data from different sources (e.g., files, HDFS) into Pig.
- STORE: Stores the results of your data processing into different destinations.
- FILTER: Filters data based on specific conditions.
- JOIN: Combines data from two or more relations based on common attributes.
- GROUP: Groups data based on specific keys.
- FOREACH: Applies an operation to each element in a relation.

Benefits of Pig Latin

- Simplified data processing: Makes big data analysis easier and more accessible for a wider audience.
- Increased productivity: Reduces the time and effort required to write complex data processing pipelines.
- Improved code readability: Declarative nature makes it easier to understand the logic of the script.
- Scalability: Leverages the power of Hadoop to handle massive amounts of data efficiently.
- Integration with other tools: Seamlessly integrates with other big data tools, allowing for smooth data flow.

Example

Filter tweets by hashtag:

```
tweets = LOAD 'tweets.txt' AS (tweet:chararray, hashtag:chararray);
```

```
filtered_tweets = FILTER tweets BY hashtag == '#bigdata';
```