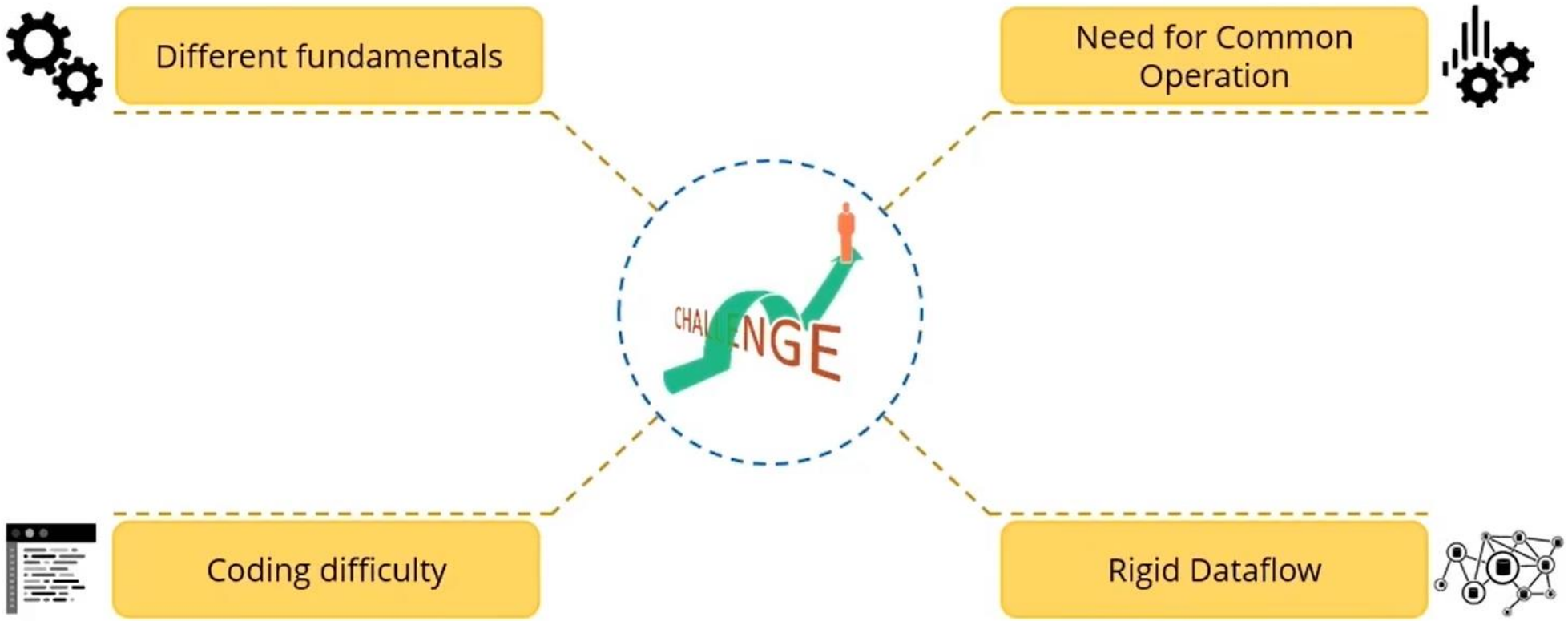
Introduction to Pig

Prior to 2006, programs were written only on MapReduce using Java.



What is Pig

- Extensible
- Self-Optimising
- Easy programmed

Pig is a scripting platform designed to process and analyze large data sets, and it runs on Hadoop clusters. Pig is extensible, self-optimizing, and easily programmed.

Pig—Example

Yahoo has scientists who use grid tools to scan through petabytes of data.

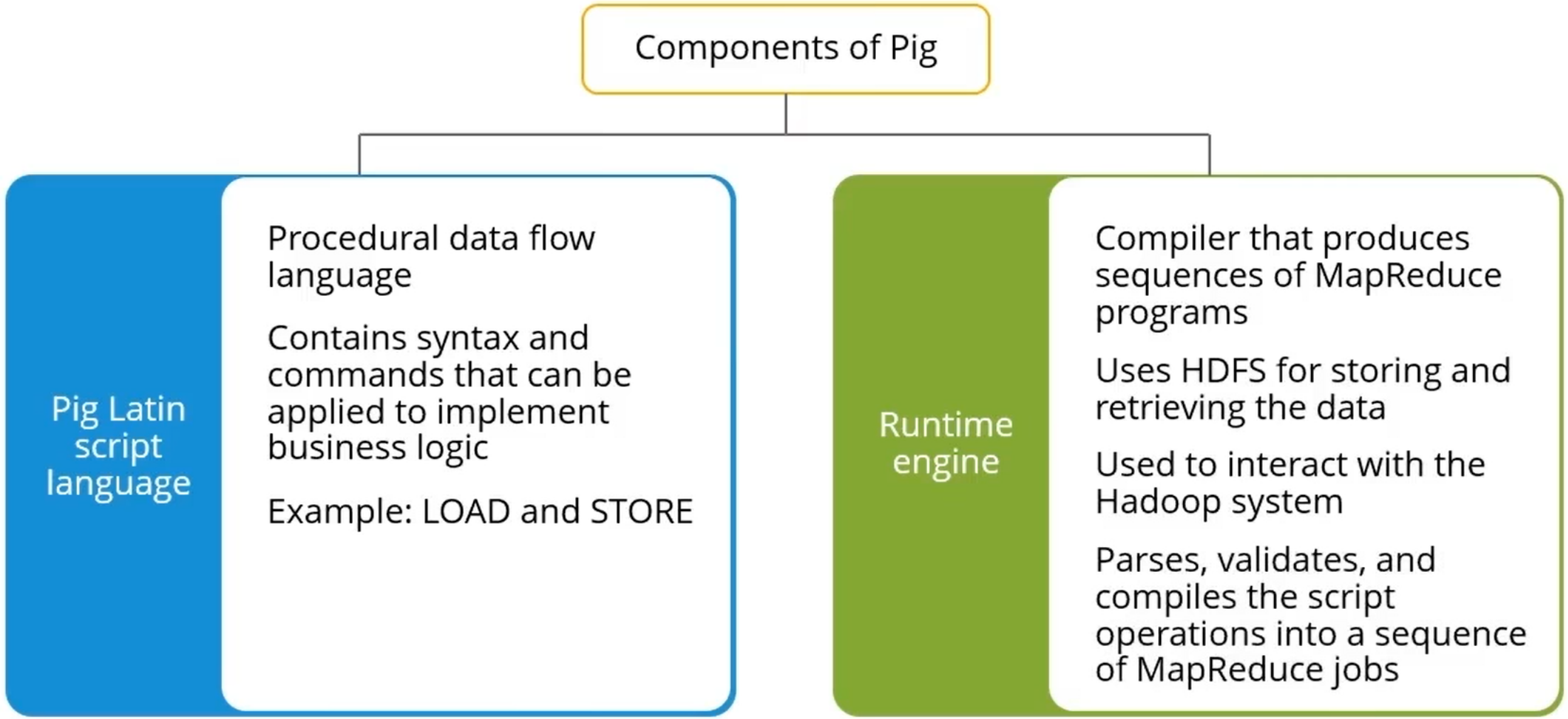
Write scripts to test a theory

In the data factory, data may not be in a standardized state

Pig supports data with partial or unknown schemas, and semi-structured or unstructured data

Components of Pig

Following are the components of Pig:



How Pig Works

Pig's operation can be explained in the following 3 stages:

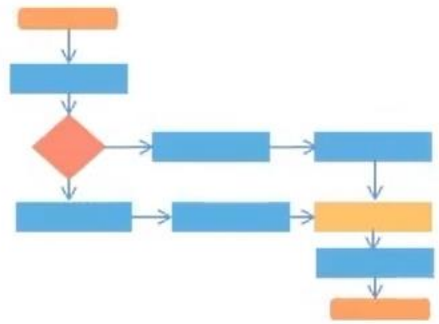

```
A = LOAD 'myfile'
AS (x, y, z);
B = FILTER A by x > 0;
C = GROUP B BY x;
D = FOREACH A GENERATE
x, COUNT(B);
STORE D INTO 'output';
```

- Parses and checks the script
- Optimizes the scripts
- Plans execution
- Submits to Hadoop
- Monitors job progress

- Results are dumped on screen or stored in HDFS

Salient Features

Developer and analysts like to use Pig as it offers many features.



Step-by-Step
Procedural Control

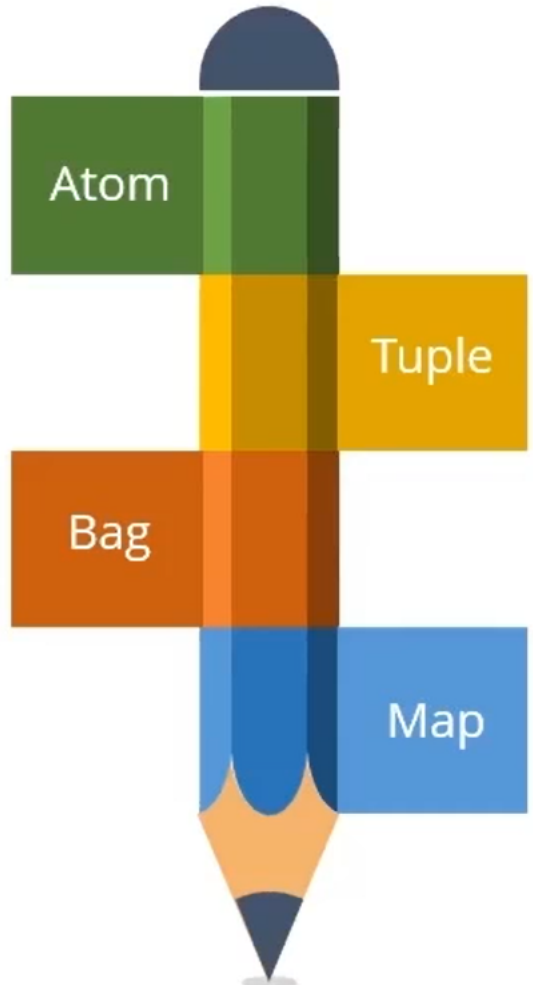
Schemas Assigned
Dynamically

Supports UDFs and
Data types

Data Model

As part of its data model, Pig supports four basic types:

A simple atomic value
Example: ‘Mike’



A sequence of fields that can be any of the data types
Example: (‘Mike’,43)

A collection of tuples of potentially varying structures; can contain duplicates
Example: {(‘Mike’), (‘Doug’, (43, 45))}

An associative array; the key must be a chararray, but the value can be any type
Example: [name#Mike,phone#5551212]

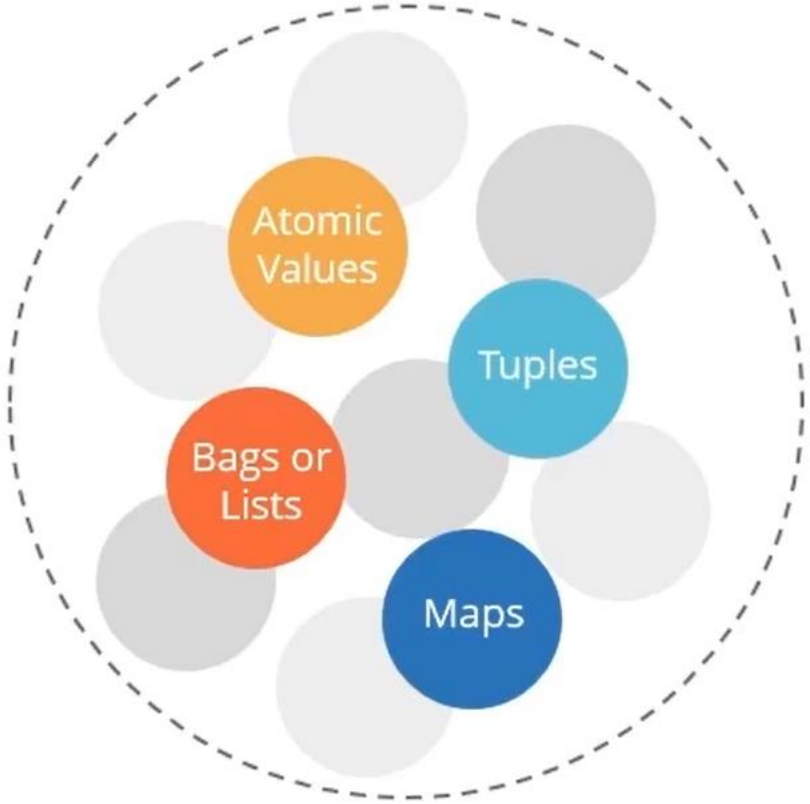
Data Model

- By default, Pig treats undeclared fields as ByteArrays.
- Pig can infer a field's type based on:
  - use of operators that expect a certain type of field,
  - User Defined Functions (UDFs) with a known or explicitly set return type, and
  - schema information provided by a LOAD function or explicitly declared using an AS clause.

Type conversion is lazy which means the data type is enforced at execution only.

Nested Data Model

Pig Latin has a fully-nestable data model.



Advantages of nested data model

- More natural to programmers than flat tuples
- Avoids expensive joins

Pig Execution Modes

Pig works in two execution modes:

Pig execution modes

Local mode: Pig depends on the OS file system.

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig -x local myscript.pig
```

MapReduce mode: Pig relies on HDFS.

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig
```

Pig Interactive Modes

Pig Latin program can be written in two interactive modes:

Pig Interactive Modes

Interactive mode: a line by line code is written and executed.

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig
grunt> customer = LOAD '/data/customer.dat' AS (c_id, city, name);
```

Batch mode: a file containing Pig scripts is created and executed in a batch.

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig myscript.pig
```

Pig vs SQL

The differences between Pig and SQL are given below:

Difference	 Pig	 SQL
Definition	Scripting language used to interact with HDFS	Query language used to interact with databases
Query Style	Step-by-step	Single block
Evaluation	Lazy evaluation	Immediate evaluation
Pipeline Splits	Pipeline splits are supported	Requires the join to be run twice or materialized as an intermediate result

Pig vs. SQL - Example

Track customers in Texas who spend more than $2,000.

SQL	Pig

SELECT c_id , **SUM**(amount) AS CTotal
FROM customers c
JOIN sales s ON c.c_id = s.c_id
WHERE c.city = 'Texas'
GROUP BY c_id
HAVING SUM(amount) > 2000
ORDER BY CTotal **DESC**
	customer = **LOAD** '/data/customer.dat' **AS** (c_id,name,city);
	sales = **LOAD** '/data/sales.dat' **AS** (s_id,c_id,date,amount);
	salesBLR = **FILTER** customer **BY** city == 'Texas';
	joined= **JOIN** customer **BY** c_id, salesTX BY c_id;
	grouped = **GROUP** joined **BY** c_id;
	summed= **FOREACH** grouped **GENERATE GROUP**, **SUM**(joined.salesTX::amount);
	spenders= **FILTER** summed **BY** $1 > 2000;
	sorted = **ORDER** spenders **BY** $1 **DESC**;
	DUMP sorted;

Getting Datasets for Pig Development

Use the following URLs to download different database for Pig development

Datasets	URL
Books	www.gutenberg.org (war_and_peace.txt)
Wikipedia database	http://dumps.wikimedia.org/enwiki/
Variant datasets	www.infochimps.com/datasets
Open data base from Amazon S3 data	http://aws.amazom.com/datasets
Open database from National climate data	http://cdo.ncdc.noaa.gov/qclcd_ascii/

Loading and Storing Methods

Loading refers to loading relations from files in the Pig buffer.
Storing refers to writing output to the file system.

Load Data

Keyword: LOAD

```
Applications  Places  System
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig
grunt> warandpeace = LOAD '/pig/war_and_peace.txt';
grunt>
```

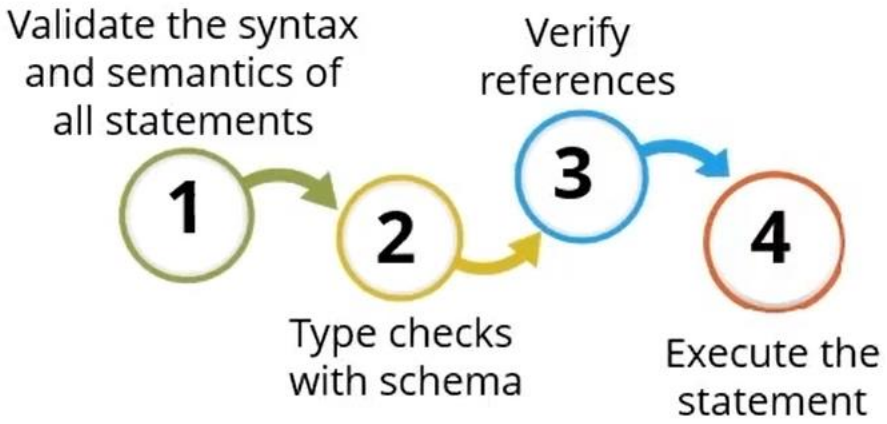
Store Data

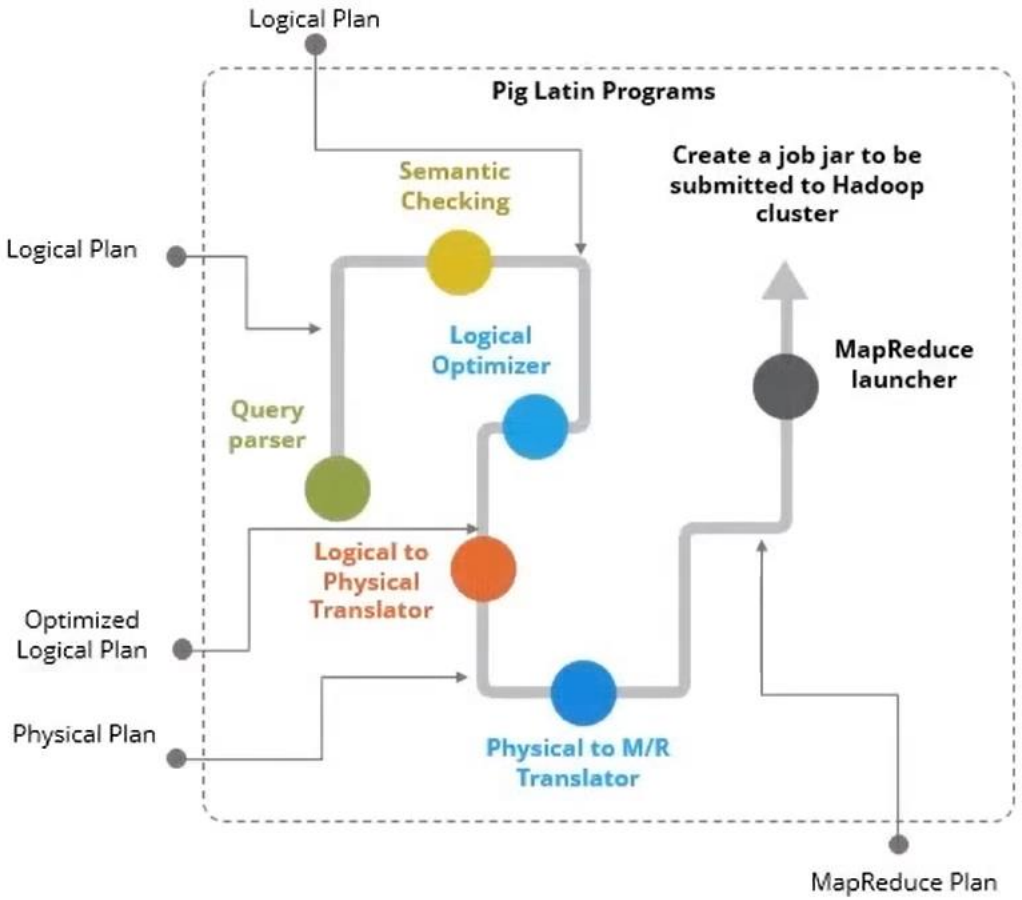
Keyword: STORE

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig
grunt> warandpeace = LOAD '/pig/war_and_peace.txt';
grunt> STORE warandpeace INTO '/pig/pigresult1' USING PigStorage();
```

Script Interpretation

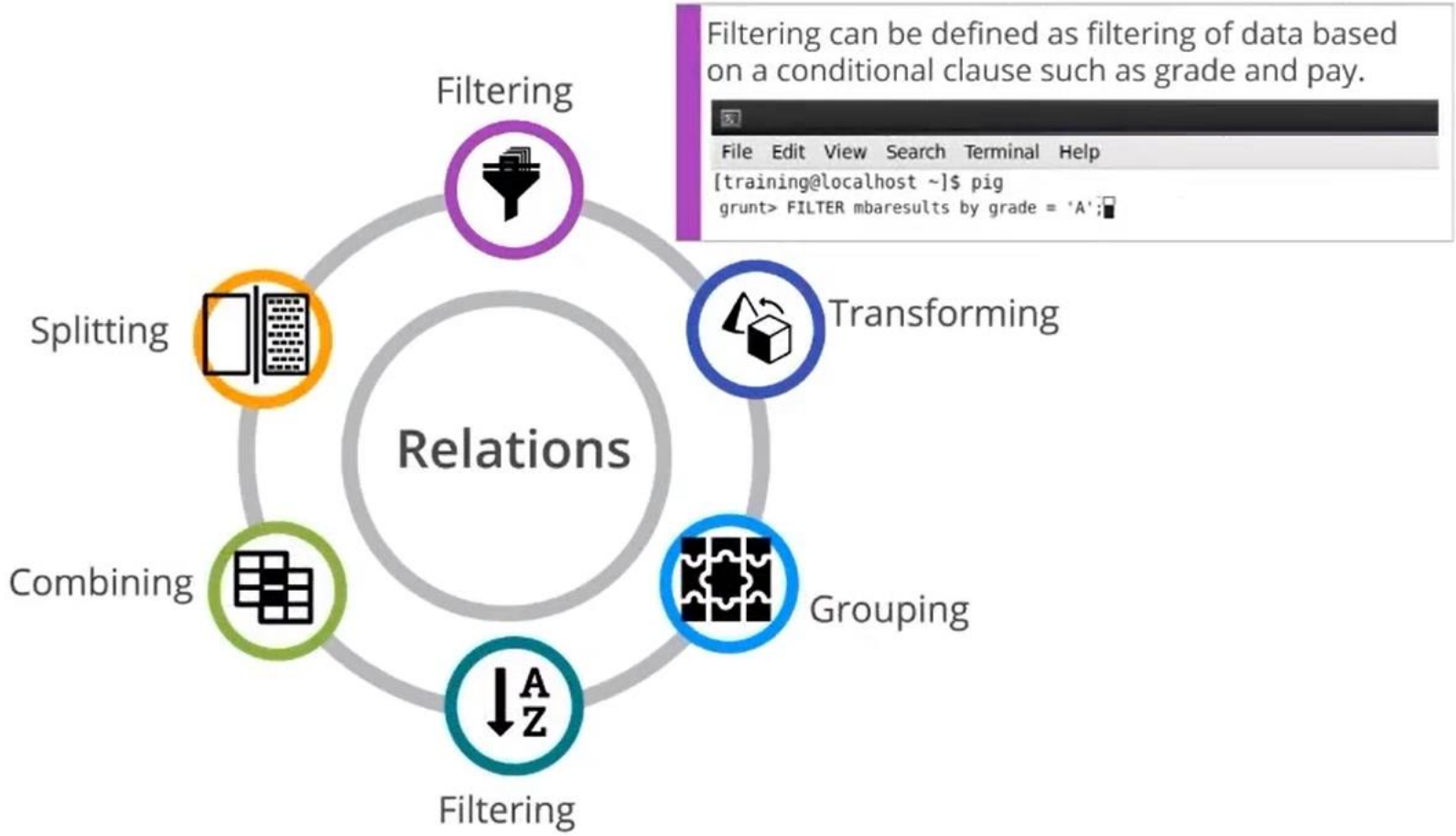
Pig processes Pig Latin statements in the following manner:





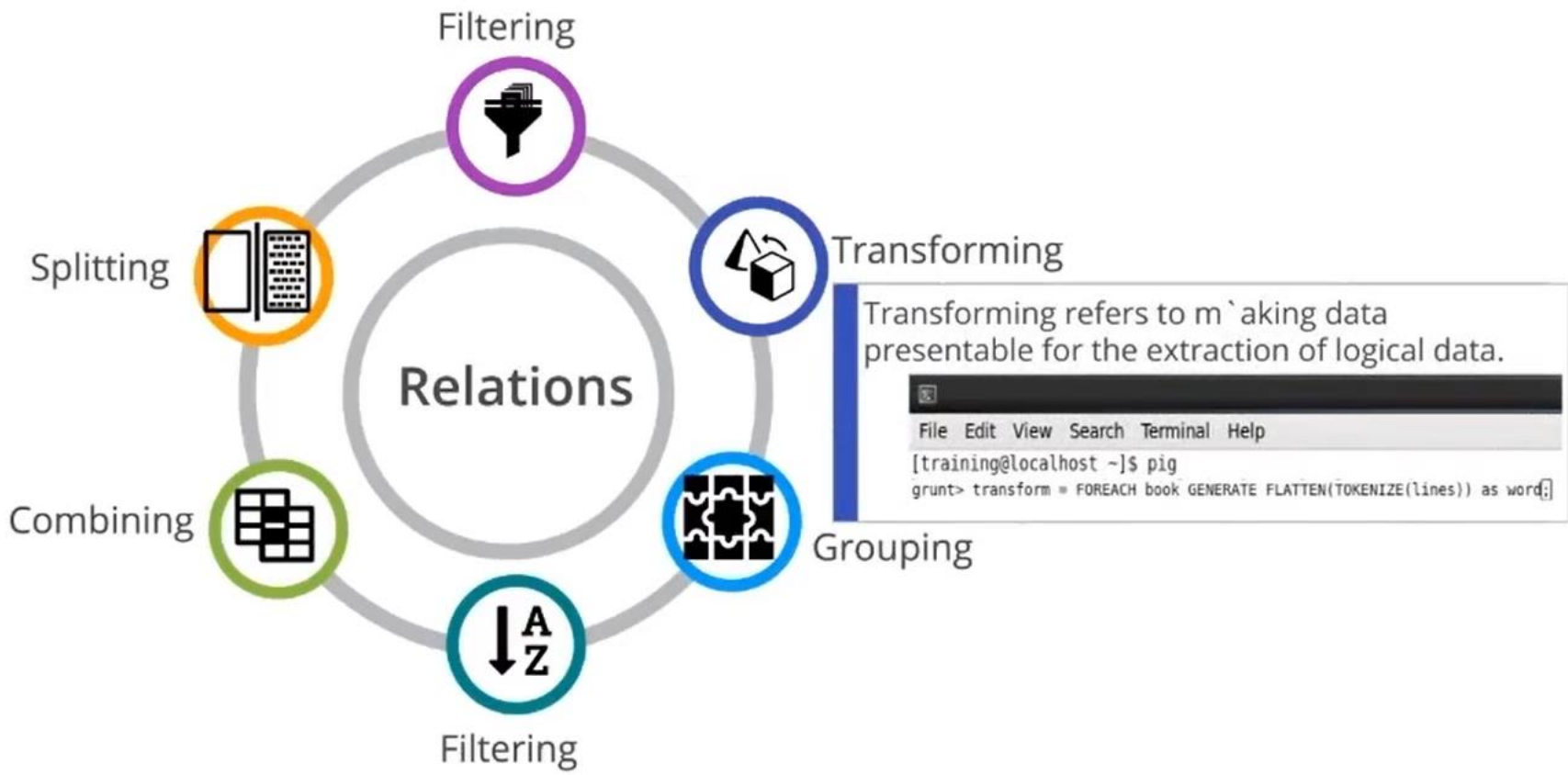
Various Relations Performed by Developers

Some of the relations performed by Big Data and Hadoop Developer are as follows:



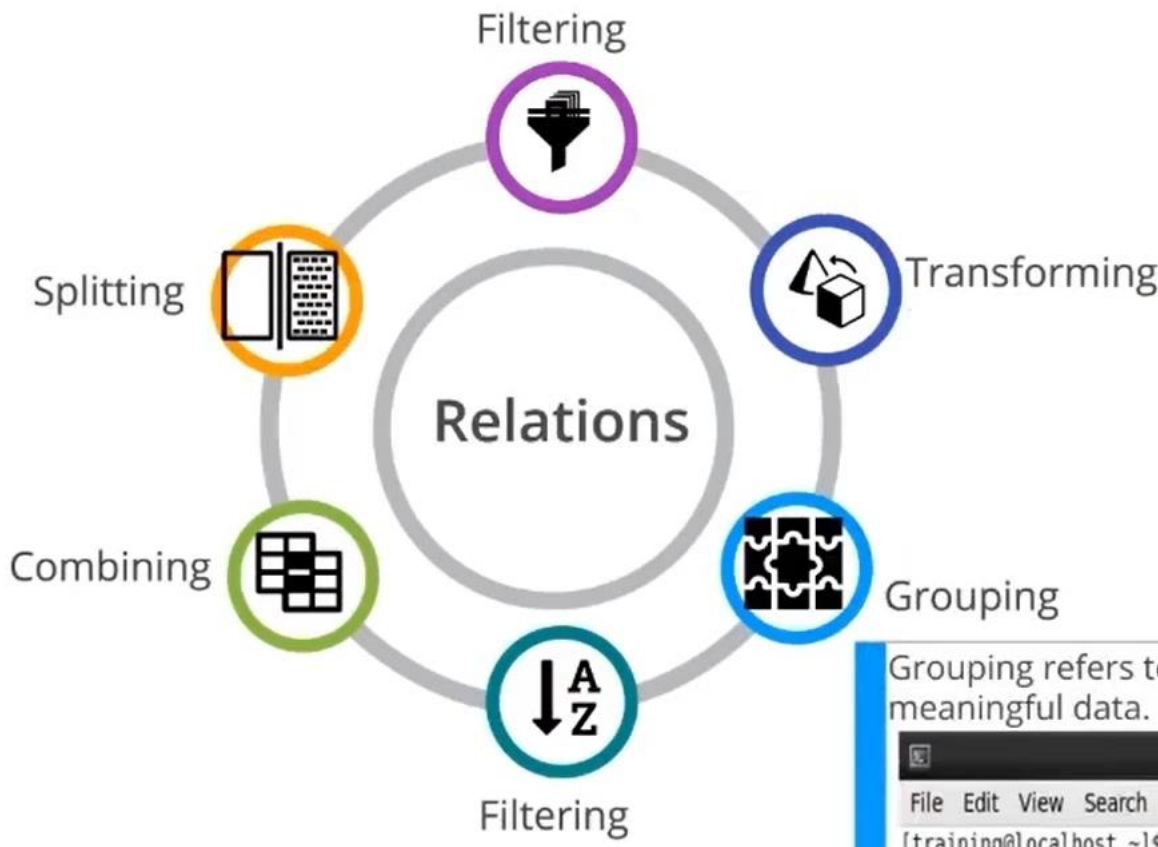
Various Relations Performed by Developers

Some of the relations performed by Big Data and Hadoop Developer are as follows:



Various Relations Performed by Developers

Some of the relations performed by Big Data and Hadoop Developer are as follows:

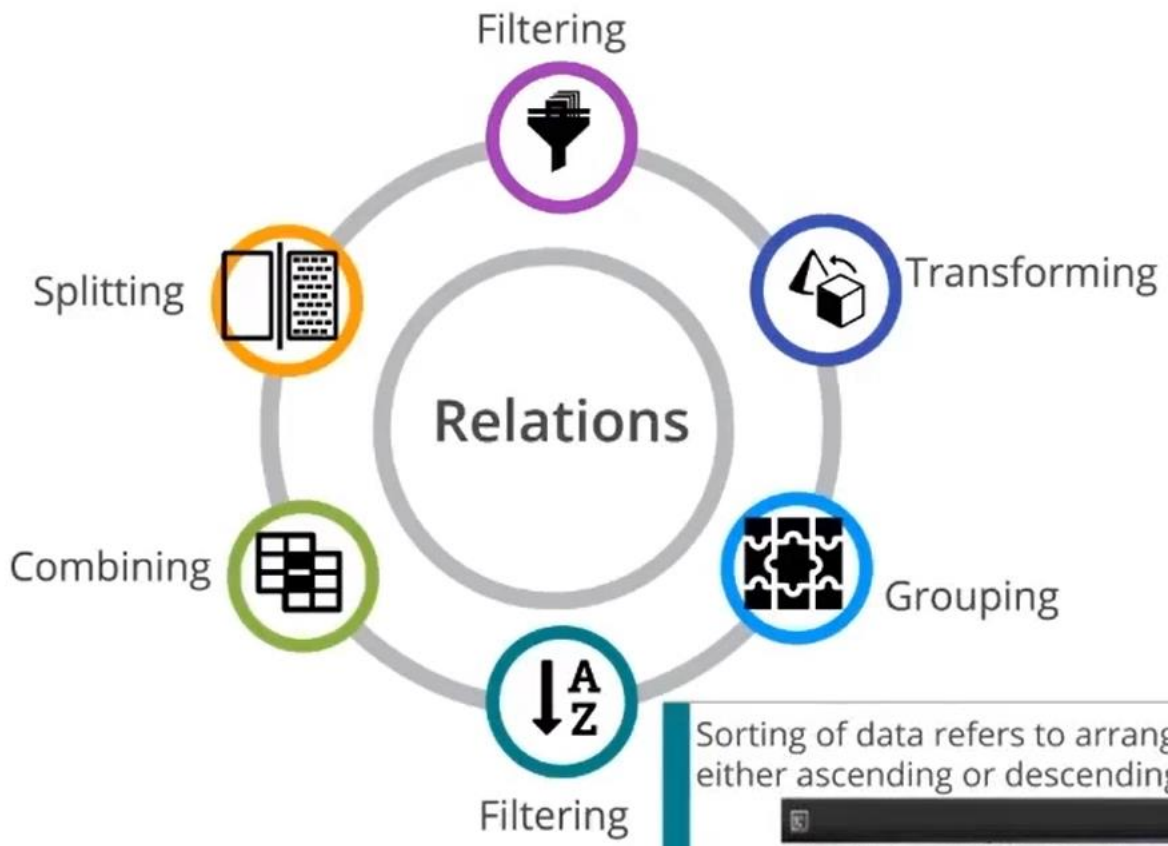


Grouping refers to generating a group of meaningful data.

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig
grunt> grouped = GROUP words by words;
```

Various Relations Performed by Developers

Some of the relations performed by Big Data and Hadoop Developer are as follows:



Sorting of data refers to arranging the data in either ascending or descending order.

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig
grunt> sorted = ORDER words by $1 DESC;
```

Various Relations Performed by Developers

Some of the relations performed by Big Data and Hadoop Developer are as follows:

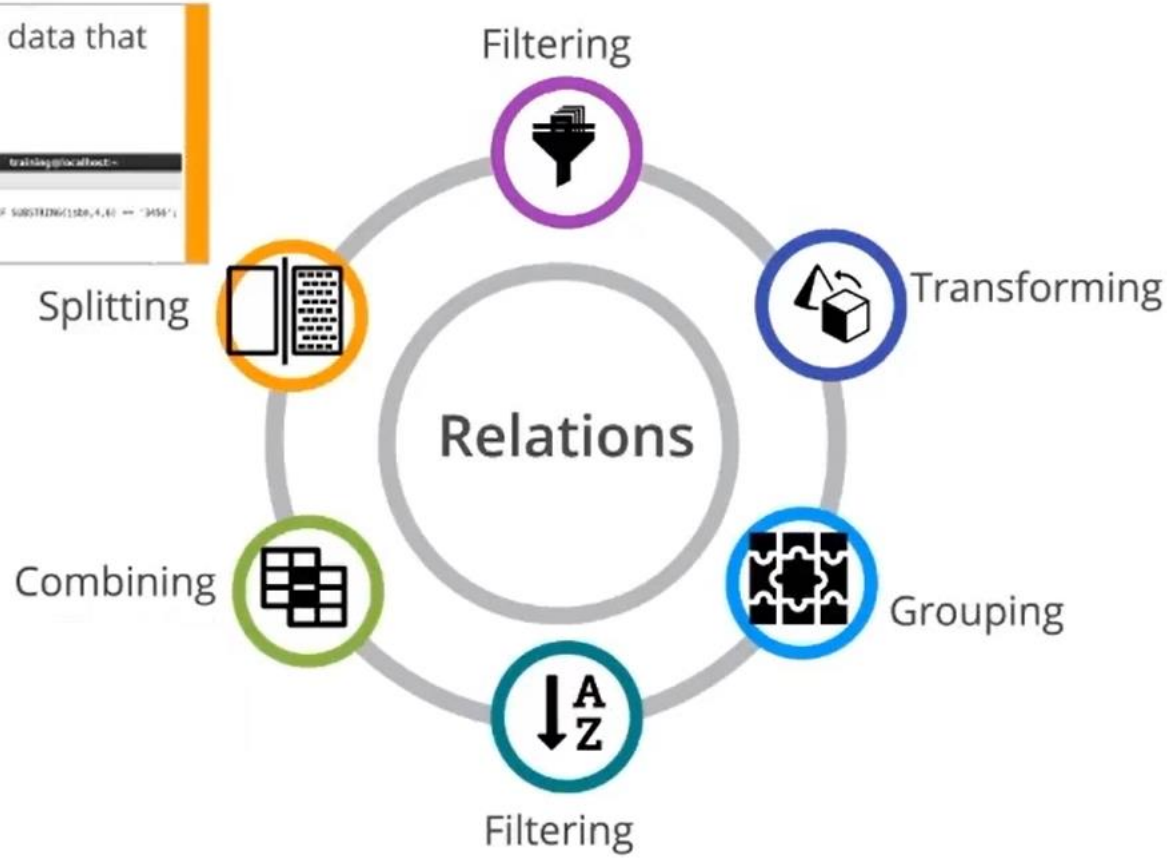
Combining refers to performing a union operation of the data stored in the variable.

```
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ pig
grunt> bookscombined = UNION book1, book2;
```

Various Relations Performed by Developers

Some of the relations performed by Big Data and Hadoop Developer are as follows:

Splitting refers to separating data that has logical meaning.



Various Pig Commands

Following are some of the Pig commands:

Pig command	What it does
load	Reads data from system
Store	Writes data to file system
foreach	Applies expressions each record and outputs one or more records
filter	Applies predicate and removes records that do not return true
Group/cogroup	Collect records with the same key from one or more inputs
join	Joins two or more inputs based on a key
order	Sorts records based on a key
distinct	Removes duplicate records
union	Merges data sets
split	Split data 2 or more sets, based on filter conditions
stream	Sends all records through a user-provided binary
dump	Writes output to stdout
limit	Limits the number of records