

Working with Basic Data Transformations

Overview

Work with the foreach and generate commands to project useful bits of data

Understand the different kinds of functions available in Pig e.g. evaluate and filter functions

Work with basic transformation such as distinct, sort, limit and split

The Foreach and Generate Commands

Foreach... Generate...

ID	Product ID	Quantity	Amount

Relation

Foreach... Generate...

ID	Product ID	Quantity	Amount

Field names

Foreach... Generate...

ID	Product ID	Quantity	Amount

Tuple

Foreach... Generate...

ID	Product ID	Quantity	Amount

**Foreach iterates through every tuple in
a relation (or inner bag)**

Foreach... Generate...

ID	Product ID	Quantity	Amount

**And projects the fields that we're
interested in**

Foreach... Generate...

ID	Product ID	Quantity	Amount

**The projected fields can also be part
of expressions**

Demo

The foreach and generate statements:

- using column indexes**
- using column names**

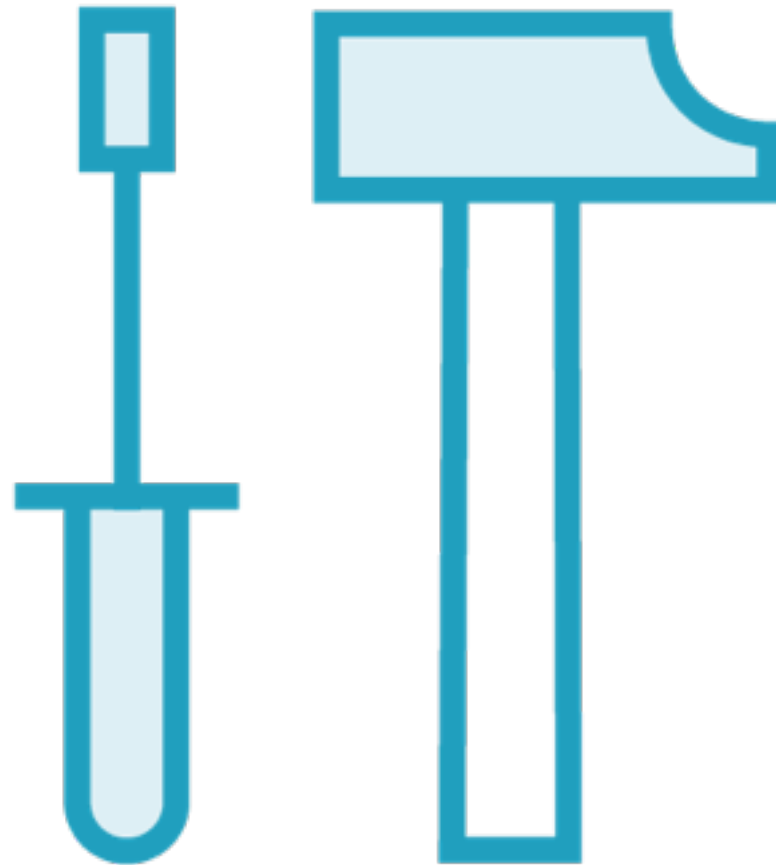
Demo

**The foreach and generate statements
with complex data types such as:**

- tuple**
- map**
- bag**

Applying Functions Using Foreach

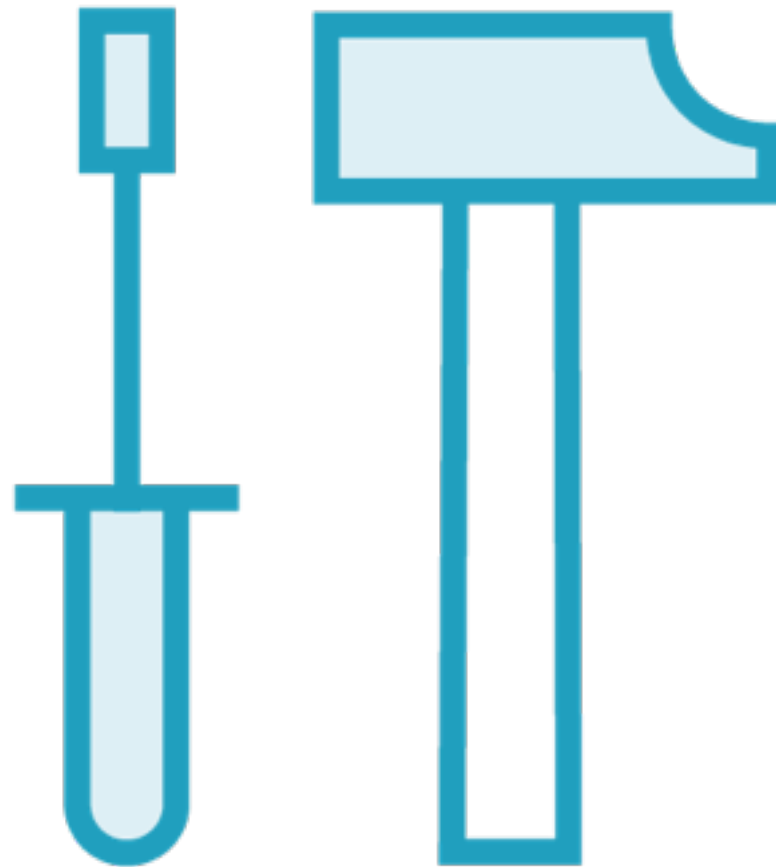
Functions in Pig



UDF - User Defined Functions

Pig allows developers to write their own custom functions which operates on data

Functions in Pig



UDF - User Defined Functions

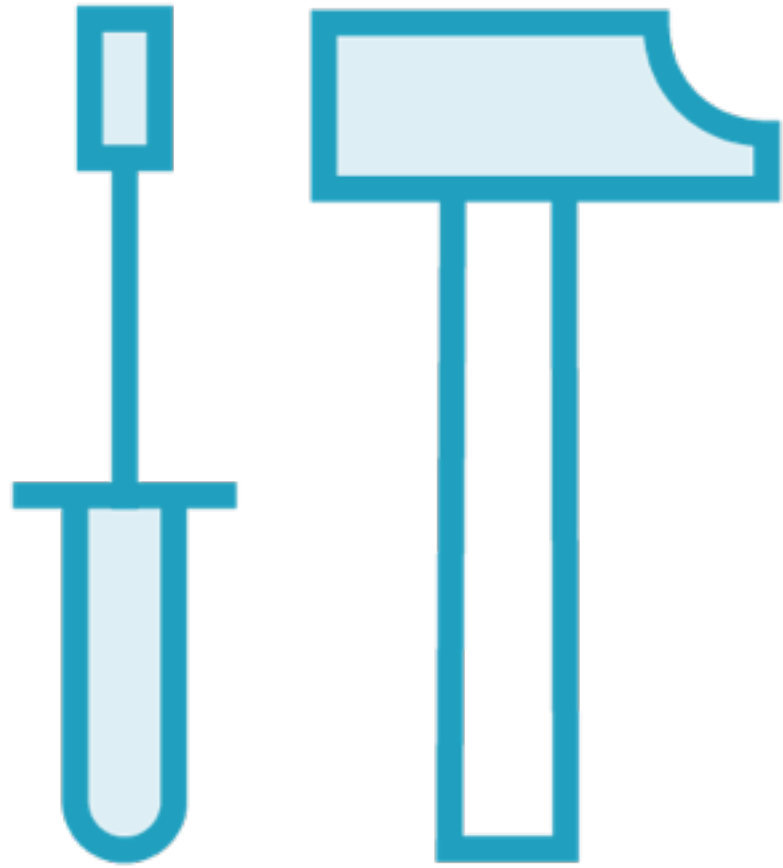
Pig supports UDFs in a number of programming languages such as Java, Python, Ruby, JavaScript

Built-in UDFs



Pig comes prepackaged with a number of UDFs that can be directly used

Built-in UDFs



These make Pig very powerful right out of the box

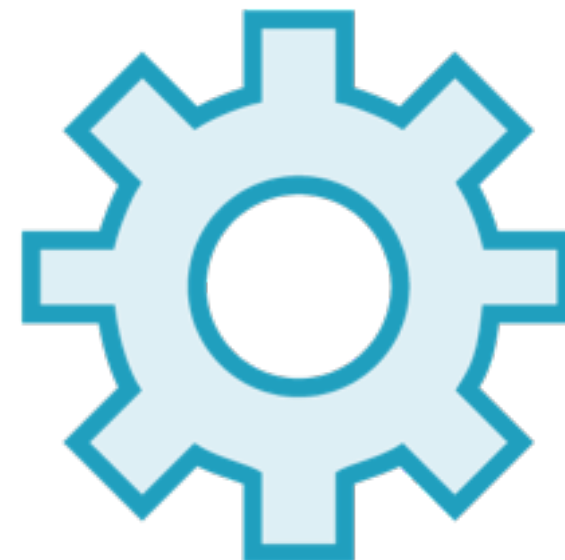
Built-in UDFs



Load



Store



Evaluate



Filter

Load and Store

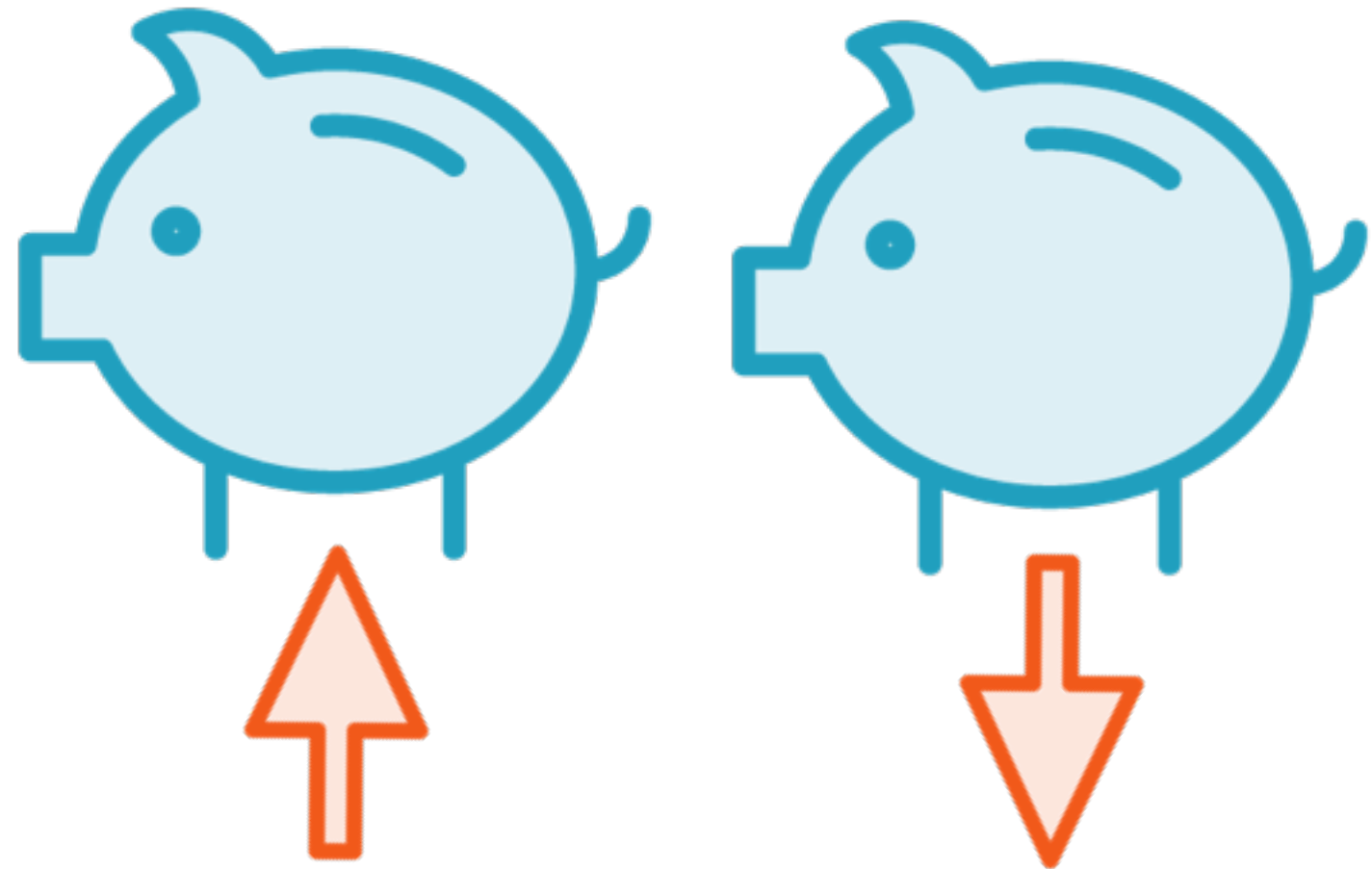
PigStorage()

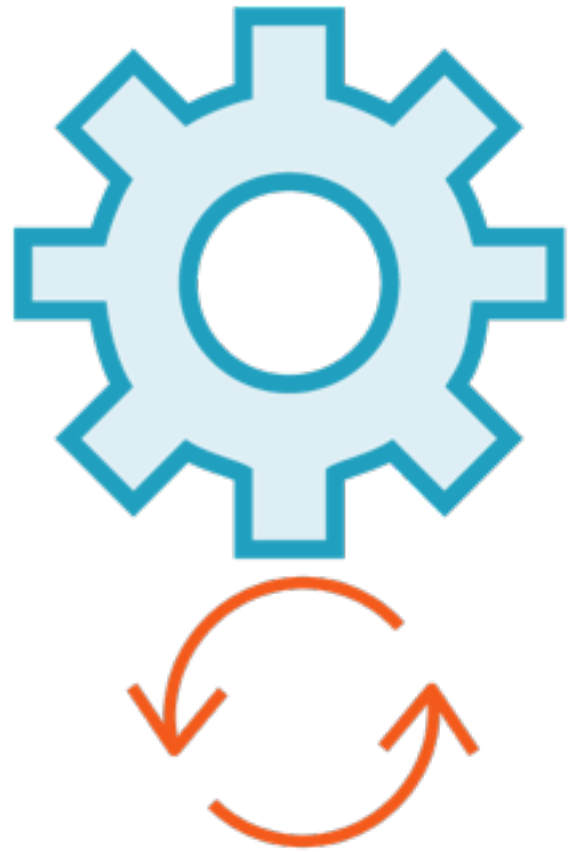
HBaseStorage()

JsonLoader()

AvroStorage()

CSVExcelStorage()





Evaluate Functions

Math: Works with numeric data types

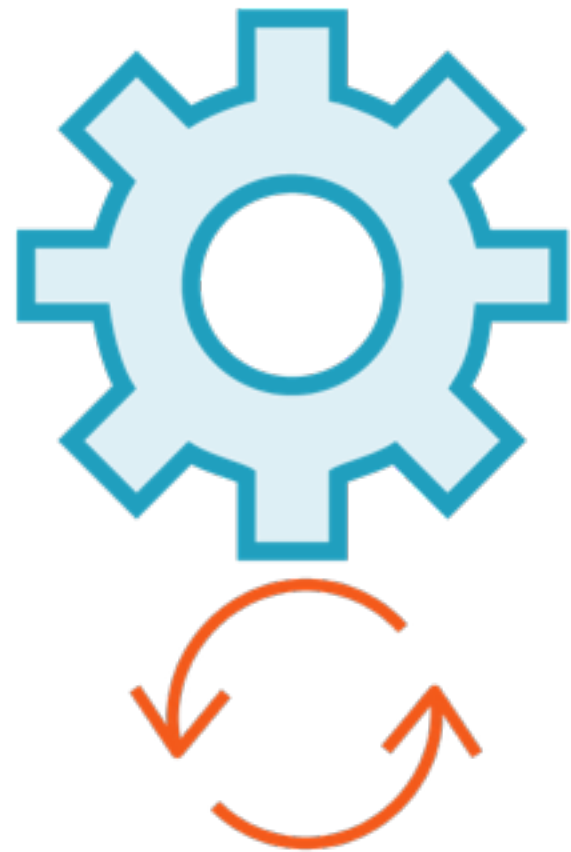
String: Works with the chararray

Date: Works with datetime

Complex data types: Used with the tuple, bag and map type

Aggregate: Different functions take different kinds of input

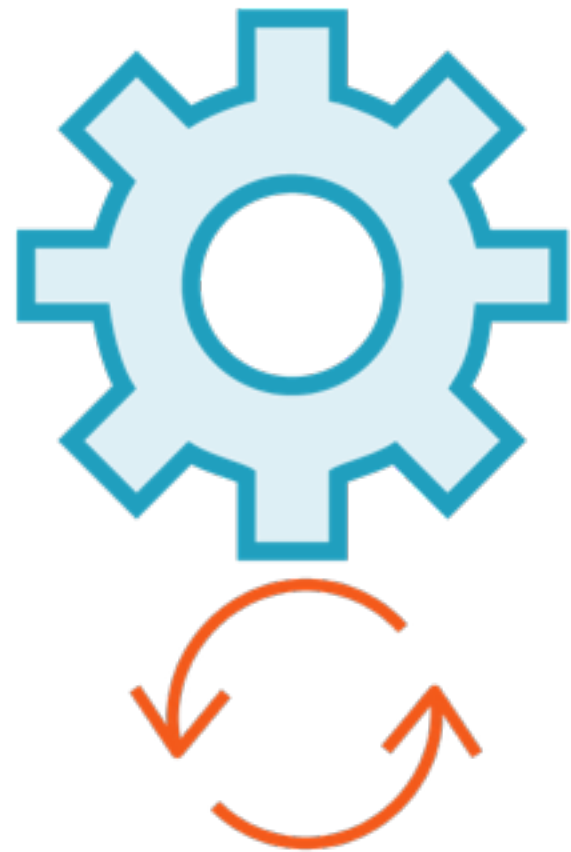
Complex Data Types



TOTUPLE(), TOBAG(), TOMAP()

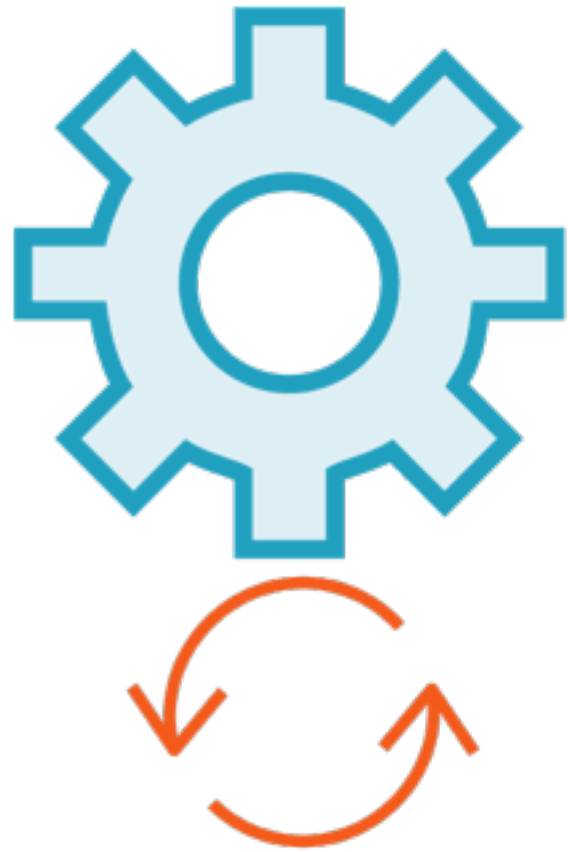
Familiar with these

Aggregate



`SUM()`, `COUNT()` etc

Will cover these once we've understood how **group by** works



Evaluate Functions

Math: Works with numeric data types

String: Works with the chararray

Date: Works with datetime

Complex data types: Used with the tuple, bag and map type

Aggregate: Different functions take different kinds of input

Demo

Use function to evaluate fields:

- math functions**
- string functions**
- date functions**

Built-in UDFs



Load



Store



Evaluate



Filter



Filter

A condition determines which record is in the result dataset

- Condition = true: Include record
- Condition = false: Leave record out

Equivalent of the **where** clause in SQL



Conditional Operators for Filters

`==, >, <,`

`!=, >=, <=`

Filter functions



Conditional Operators for Filters

`==, >, <,
!=, >=, <=`

All work with
scalar data types



Conditional Operators for Filters

`==, >, <,
!=, >=, <=`

**Apply to maps
and tuples**



Conditional Operators for Filters

~~==, >, <, !=, >=, <=~~

**None work with
bags**

Conditional Operators for Filters



Filter functions

**UDFs meant to be used
with the filter command**

Filter Functions



matches

IsEmpty



Filter Functions

matches

`order_id matches '0D01'`

`order_id matches '0D01.*'`



Filter Functions

IsEmpty

Used to check
whether a bag or
map is empty

They cannot be null

Demo

Use the filter keyword to filter records by predicate using:

- conditional operators
- the matches filter function

The Distinct, Limit and Sort Commands

Distinct, Limit and Sort

Distinct

Remove duplicate tuples from a relation

Limit

Choose a specified number of tuples from a relation

Order By

Sort tuples in ascending or descending order based on a column value

Distinct

Remove duplicate tuples
from a relation

Distinct

Acts on entire records in a relation

- tuples where all fields have the same value are duplicates

Does not act on individual fields

Limit

Choose a specified
number of tuples from a
relation

Limit

**Specify N, the number of records
we're interested in**

Limit chooses the first N records

Order By

Sort tuples in ascending
or descending order
based on a column value

Order By

Sorting is done based on a particular column

Ascending as well as descending order

Default is ascending order

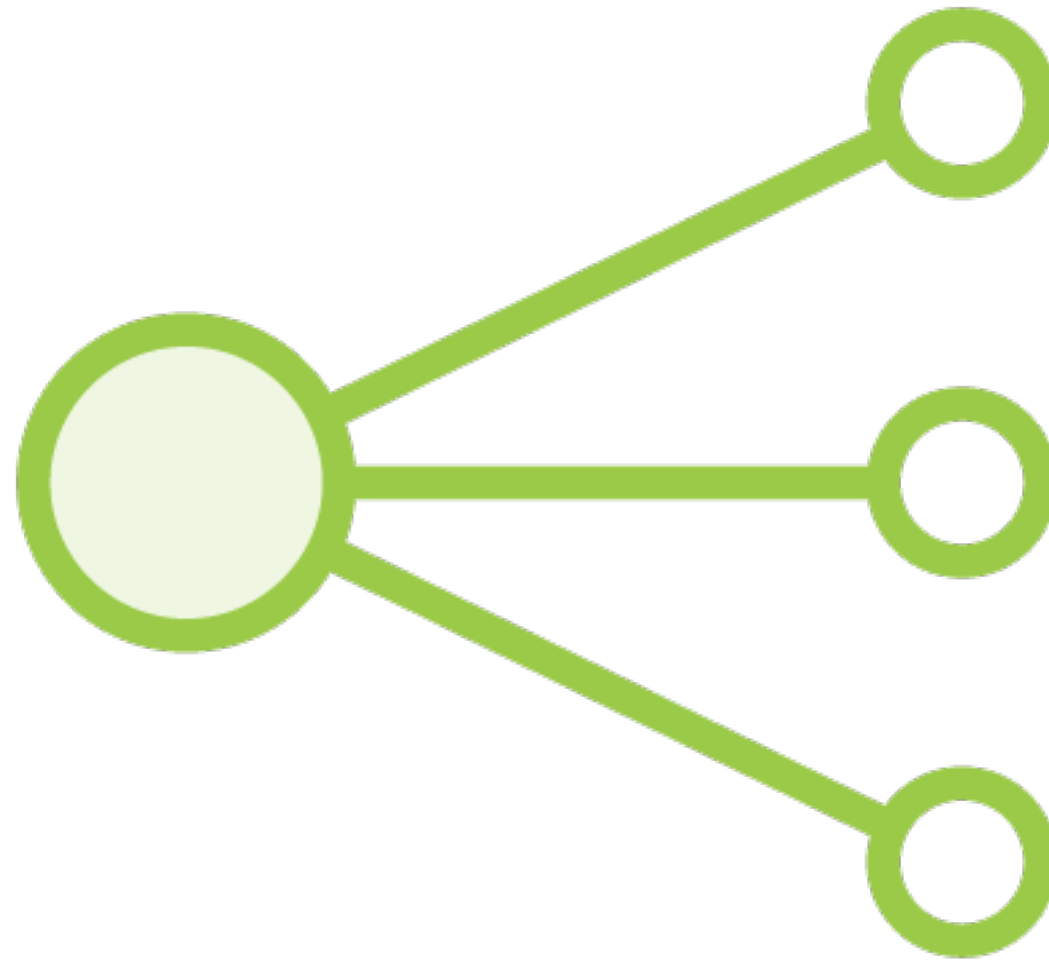
Demo

Implement commands in Pig using the following keywords

- distinct**
- limit**
- order by**

The Split Command

Split



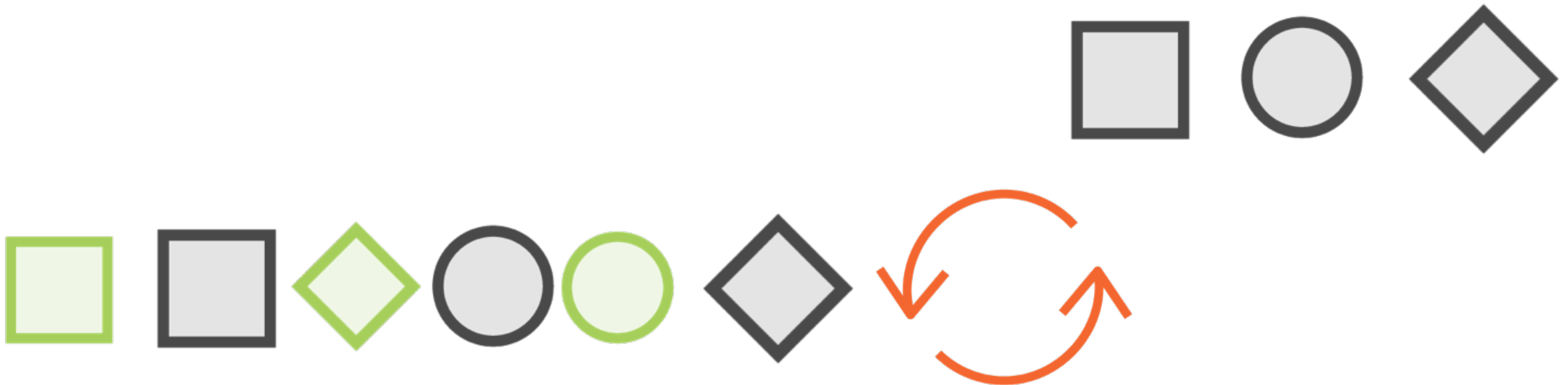
Split

**Explicitly split data into two or more
data flows based on conditions**

Split



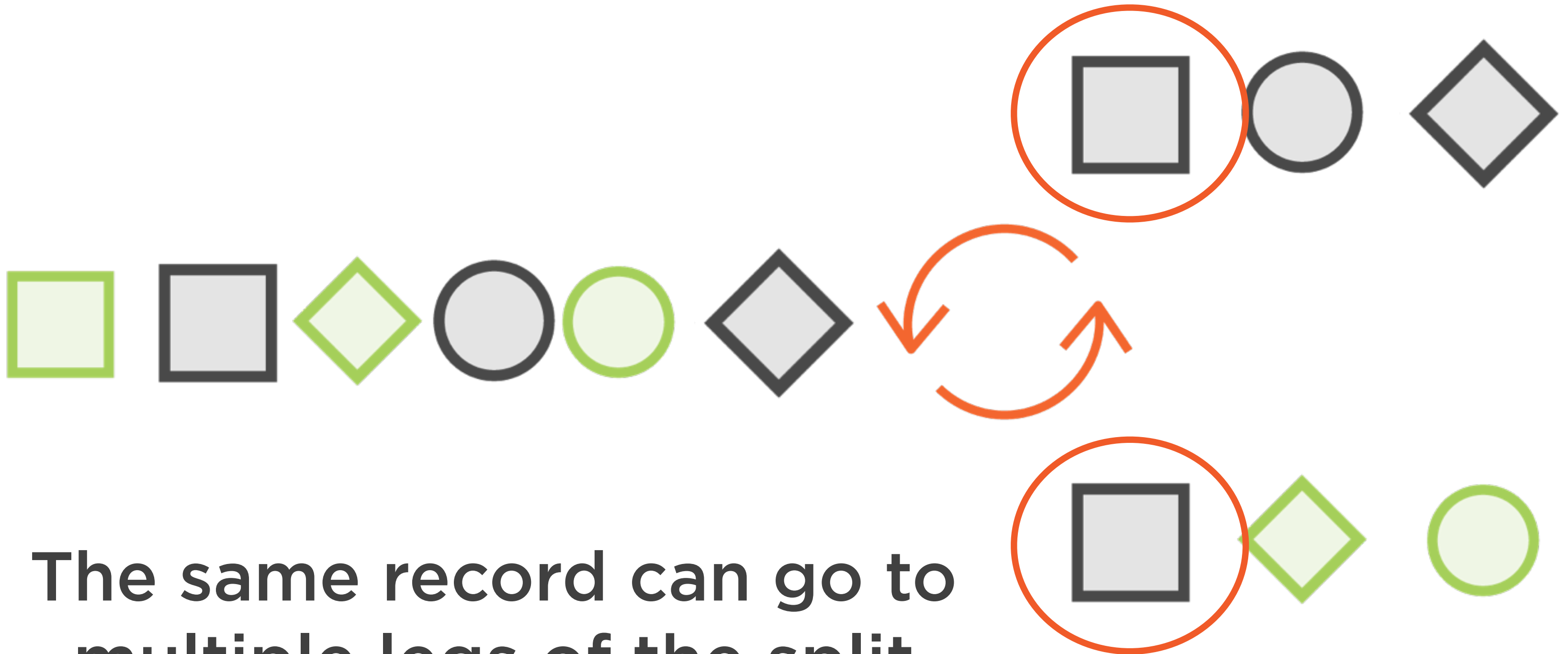
Split



**2 or more data flows
can be created**



Split



**The same record can go to
multiple legs of the split**

Demo

Use the split command to separate records into multiple logical relations

Summary

Implemented the foreach-generate commands to project useful information from relations

Understood and worked with different types of Pig functions such as evaluate and filter.

Implemented basic transformations such as distinct, sort, limit and split