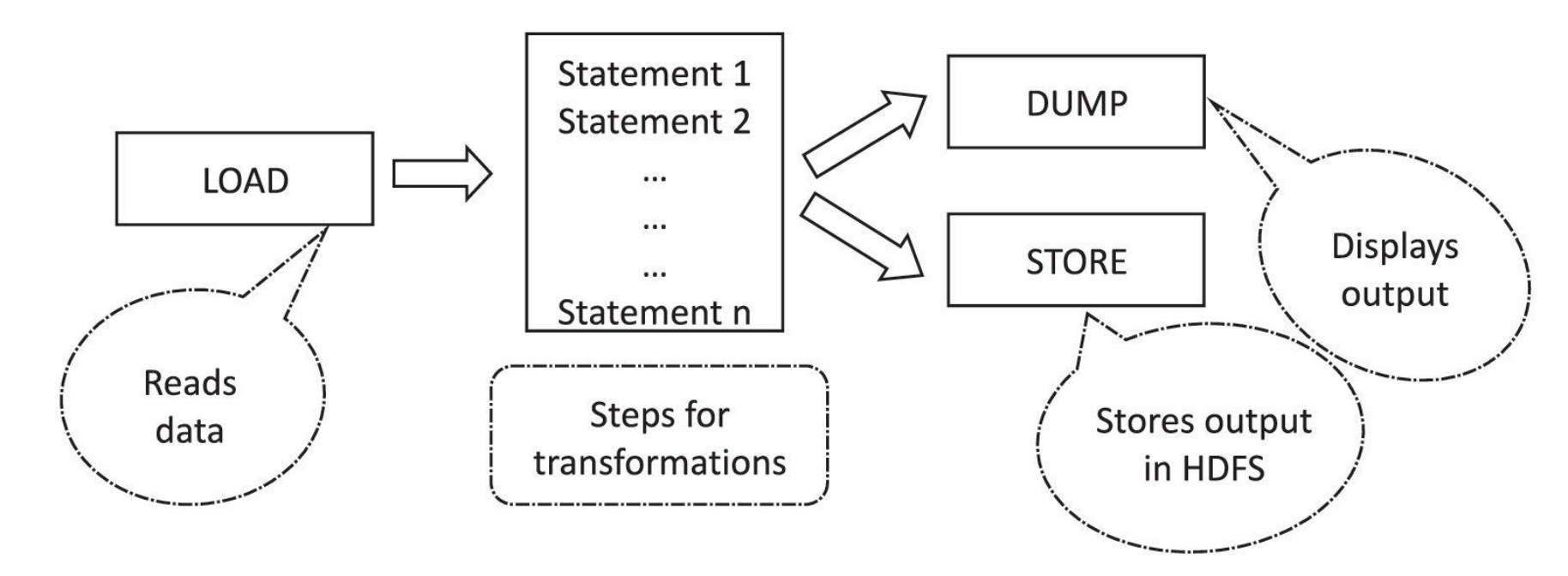
Lesson 8

PIG Programming

PIG Commands

- LOAD
- STORE
- SUMP

Figure 4.15: Order of processing Pig statements– load, dump, and store



PIG Latin Data Model

- Supports primitive data types, which are atomic or Scalar data types.
- Atomic data types─ int, float, long, double, char[], byte[]
- Complex data types─ tuple, bag, and map.
- Table 4.17 ─ Data types and examples

Tuple

- A record of an ordered set of fields
- Similar to a row in a table of RDBMS
- The elements inside a tuple do not necessarily need to have a schema associated to it
- Represents by '()' symbol
- For example, (1, Oreo, 10, Cadbury)

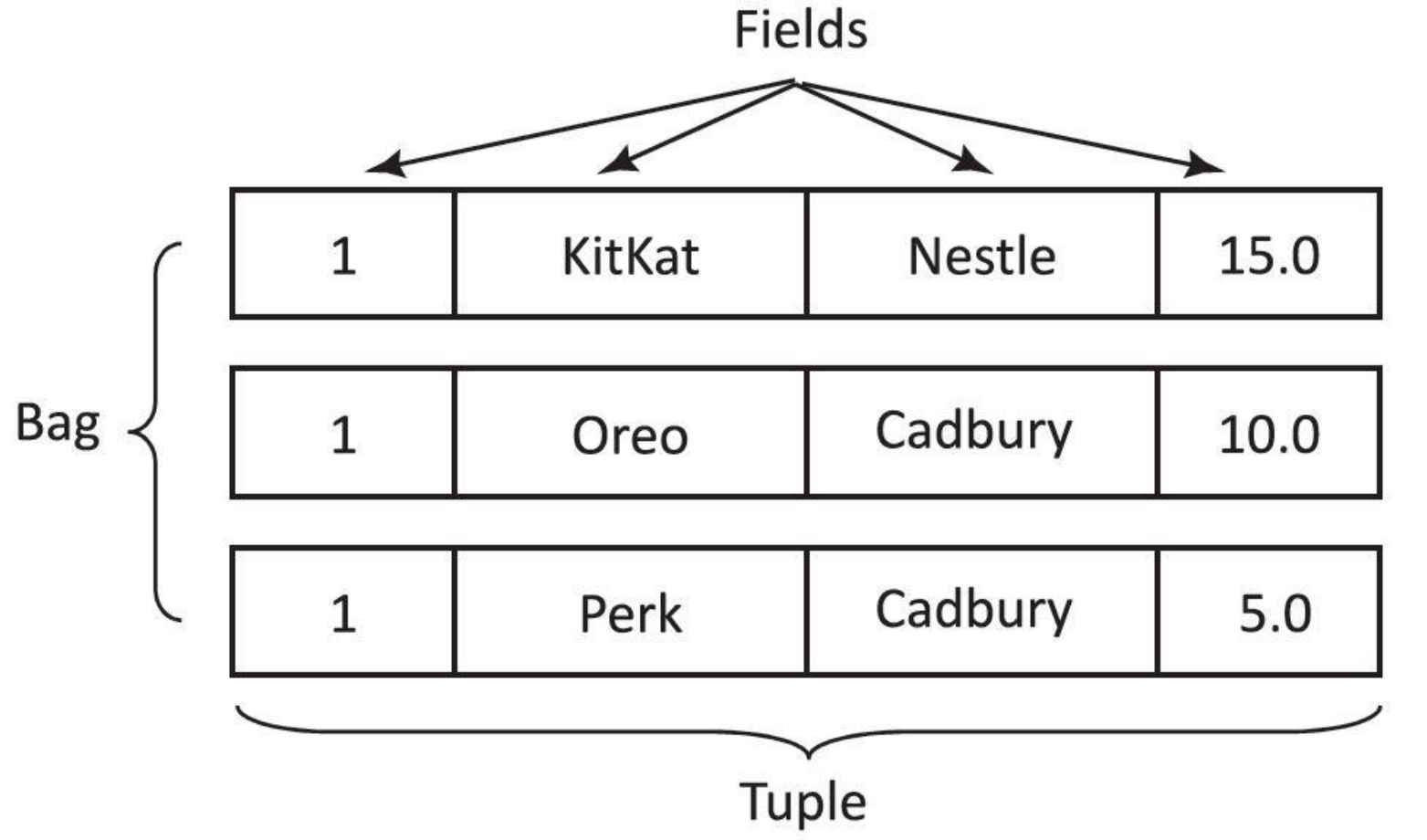
Bag

- An unordered set of tuples
- Can contain duplicate tuples as it is not mandatory that they need to be unique
- Each tuple can have any number of fields (flexible schema).
- Can also have tuples with different data types

Example

- Bag of tuples: (Oreo, 10) and (KitKat, 15, Cadbury)
- Represented by {(Oreo, 10), (KitKat, 15, Cadbury)}

Figure 4.13 Pig Data Model with fields, Tuple and Bag



Two Types of Bags

- Outer bag or relations
- Inner bag.

Pig Latin Relation (Outer Bag)

- A bag of tuples
- Similar as relations in relational databases
- But unordered (there is no guarantee that tuples are processed in any particular order)

Outer bag or relation

- Example:{(Oreo, Cadbury), (KitKat, Nestle), (Perk, Cadbury)}
- Bag explains the relation a between the Chocolate brand and their branding company

Inner Bag

- A field in a relation; in that context, it is known as inner bag
- Thus, an inner bag contains a bag inside a tuple

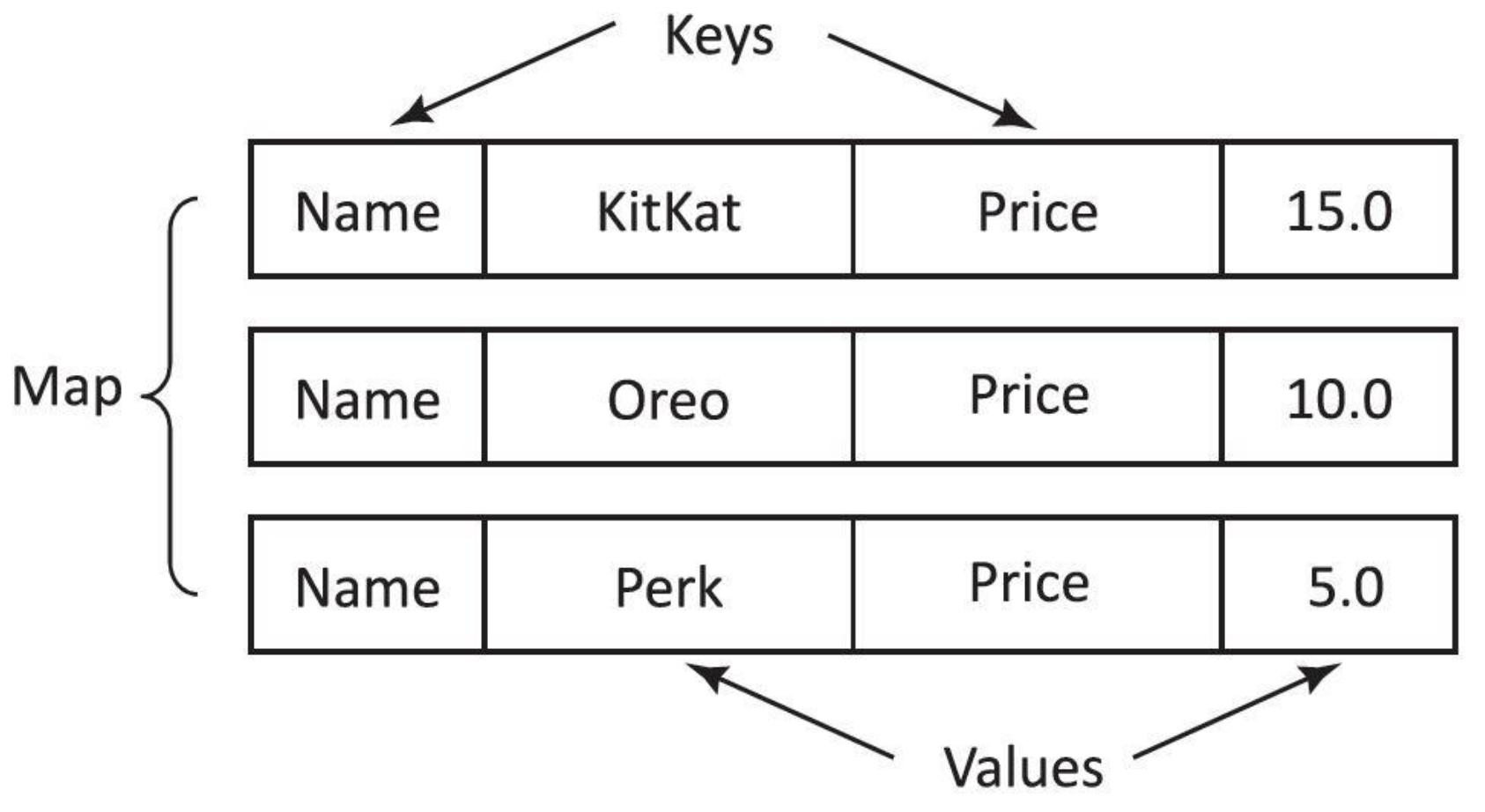
Map (Data Map)

- A set of key-value pairs
- The key needs to be of type chararray and should be unique (similar to a column name)

Map (Data Map)

- Can be indexed and associated to value with it
- Value accesses from the keys
- The value might be of any type.
- [] symbol represents Map and the key and value separate by '#'
- Example, [type#Oreo, price#10]

Figure 4.14 Relation and corresponding keys and their values (key-value pairs)



PIG Latin Features

- Programs (scripts) execute in the Pig run-time environment
- Enables developing the scripts for data-analysis
- Consists of a number of operators

Operators in Pig Latin

- Arithmetic Operators: +, -, *, /, %
- Comparison Operators: ==, !=, <, >,≤, ≥
- Boolean Operators: AND, OR, NOT

PIG Latin Operators

- Help in developing their own functions for reading, writing, and processing data

foreach

- A simple way to apply transformations based on columns
- A projection operator

Operations

- Load an entire record, but then remove all but the name and phone fields from each record:
- Tuple projection using dot operator.
- Bag projection
- Bag projection
- Add all integer values

Pig Latin Script Execution Modes

- Interactive Mode– using the Grunt shell.
- Batch Mode– Writing the Pig Latin script in a single file with .pig extension
- Embedded Mode– defining UDFs in programming languages such as Java and using them in the script.

PIG User defined functions (UDFs)

- Custom functions, not available in Pig
- UDF can be in other programming languages such as, Java, Python, Ruby, Jython, JRuby
- UDF codes easily embed into Pig scripts

Pig UDF

- Should extend either a Filter function or Eval function
- Must contain a core method called *exec*, which contains a Tuple

EVAL Functions

- AVG, SUM, MAX, MIN, COUNT, COUNT_STAR, CONCAT, DIFF, IsEmpty, Size, Tokenize

EVAL Functions

- AVG, SUM, MAX, MIN, COUNT, COUNT_STAR, CONCAT, DIFF, IsEmpty, Size, Tokenize

Filter Function

- FILTER gives a simple way to select tuples from a relation based on some specified conditions (predicate)
- It is Pig's *select* command

Group Function

- GROUP statement collects together records with the same key
- No direct connection between group and aggregate functions in Pig Latin unlike SQL

OrderBy Function

- ORDER statement sorts the data based on a specific field value, producing a total order of output data

Distinct Function

- DISTINCT removes duplicate tuples
- Works only on entire tuples, not on individual fields

JOIN Function

- JOIN statement joins two or more relations based on values in the common field.
- Keys indicate the inputs
- When those keys are equal, two tuples are joined
- Tuples for which no match is found are dropped

Limit

- LIMIT gets the limited number of results
- Example: Outputs only first 5 tuples from the relation

Sample

- SAMPLE offers to get a sample of the entire data
- Reads through all of the data but returns only a percentage of rows on the random basis
- Example: sample A 0.1; returns 10% of tuples from the relation A

Split and Parallel

- SPLIT partitions a relation into two or more relations
- Parallel statements for parallel data processing

Piggy Bank

- Pig users share their functions from Piggy Bank
- *Register* is keyword for using Piggy bank functions

Summary

We learnt PIG:

- load, dump, and store statements
- Operators: +, -, *, /, %, ==, !=, <, >,≤, ≥ , AND, OR, NOT
- Data Model
- Bag, Tuple, Map
- UDF Creation
-

Summary

We learnt :

- Group
- Distinct
- Limit
- Sample

Summary

We learnt PIG Latin:

- foreach
- EVAL
- FILTER
- Group By
- OrderBy
- Join

End of Lesson 8 on PIG Programming