USN

Internal Assessment Test 3 – January 2022

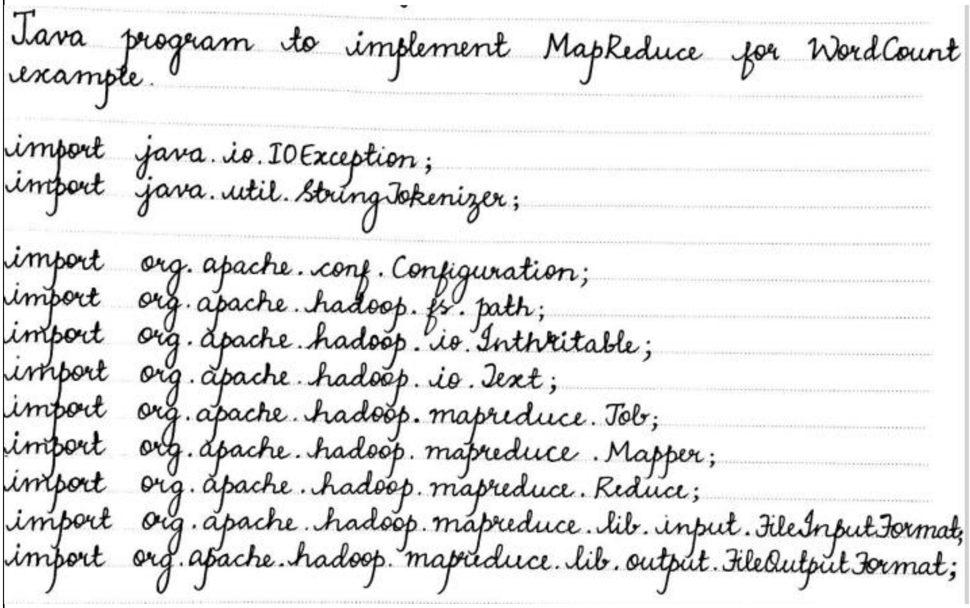
Sub:	BIG DATA AND ANALYTICS					Sub Code:	18CS72	Branch:	ISE
Date:	24/01/2022	Duration:	90 min's	Max Marks:	50	Sem / Sec:	VII / A, B & C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	**What is MapReduce and write the java program for Map phase and Reduce phase of Word Count Problem.**	[10]	CO4	L3

Scheme: MapReduce + Program = 2 + 8 =10M

Solution:

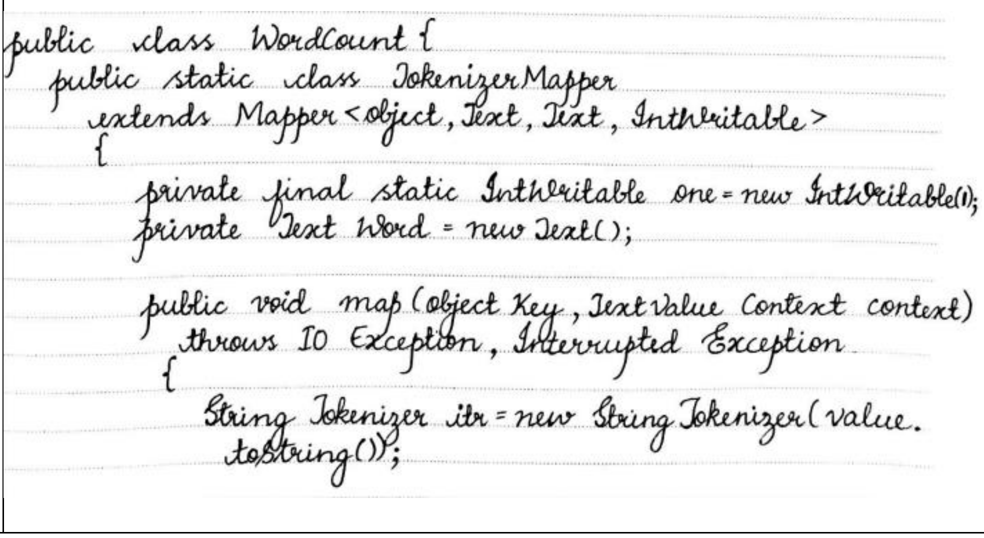
MapReduce programming model refers to a programming paradigm for processing Big Data sets with a parallel and distributed environment using map and reduce tasks.



Java program to implement MapReduce for WordCount example.

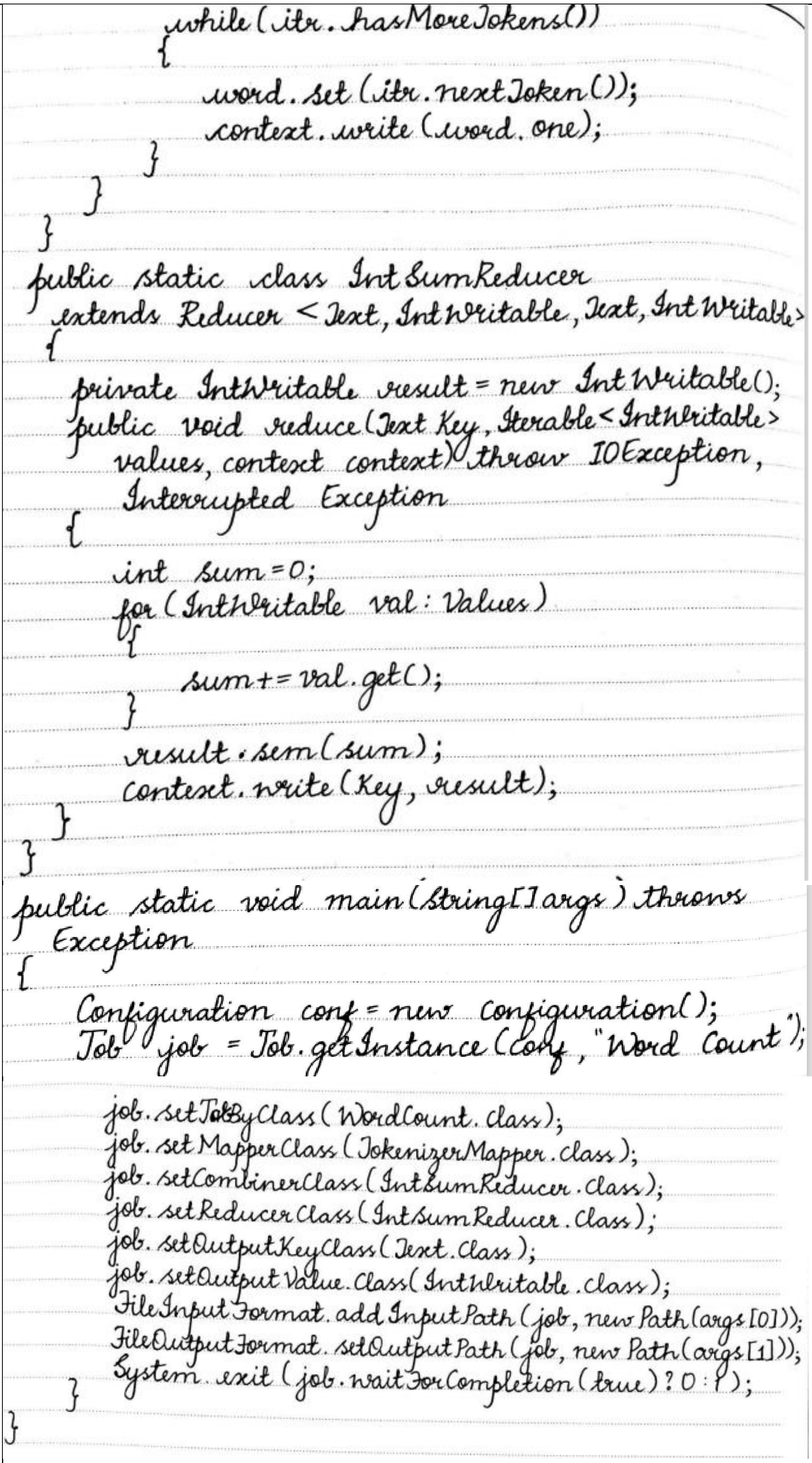
```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.conf.Configuration;
import org.apache.hadoop.fs.path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reduce;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```



```
public class WordCount {
  public static class TokenizerMapper
    extends Mapper<object, Text, Text, IntWritable>
    {
        private final static IntWritable one = new IntWritable(1);
        private Text Word = new Text();

        public void map(object Key, Text value Context context)
          throws IO Exception, Interrupted Exception
          {
             String Tokenizer itr = new String Tokenizer(value.
              toString());
```



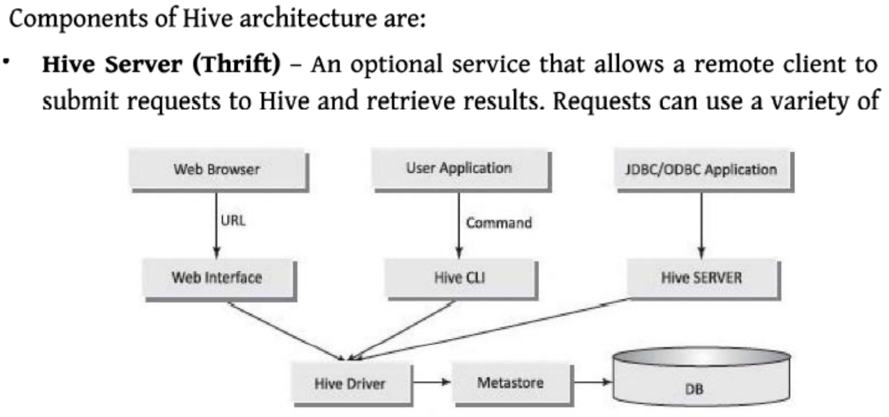
```
            while (itr.hasMoreTokens())
            {
                word.set (itr.nextToken());
                context.write (word.one);
            }
        }
    }
public static class IntSumReducer
  extends Reducer <Text, IntWritable, Text, IntWritable>
  {
    private IntWritable result = new IntWritable();
    public void reduce(Text Key, Iterable<IntWritable>
       values, context context) throw IOException,
       Interrupted Exception
    {
       int sum=0;
       for (IntWritable val: Values)
       {
           sum+= val.get();
       }
       result.sem(sum);
       context.write(Key, result);
    }
  }
public static void main(String[]args) throws
  Exception
{
    Configuration conf = new configuration();
    Job job = Job.getInstance (Conf, "Word Count");

       job.setJobByClass( WordCount.class);
       job.set MapperClass (TokenizerMapper.class);
       job.setCombinerClass (IntSumReducer.Class);
       job.setReducerClass (IntSumReducer.Class);
       job.setOutputKeyClass( Text.Class);
       job.setOutput Value.Class( IntWritable.class);
       FileInput Format.add InputPath (job, new Path(args[0]));
       FileOutputFormat.setOutputPath (job, new Path(args[1]));
       System.exit (job.waitForCompletion (true)? 0 : 1);
    }
}
```

2	**Explain in detail about Hive Architecture with neat diagram, data types and file formats of Hive.**	[10]	CO4	L2

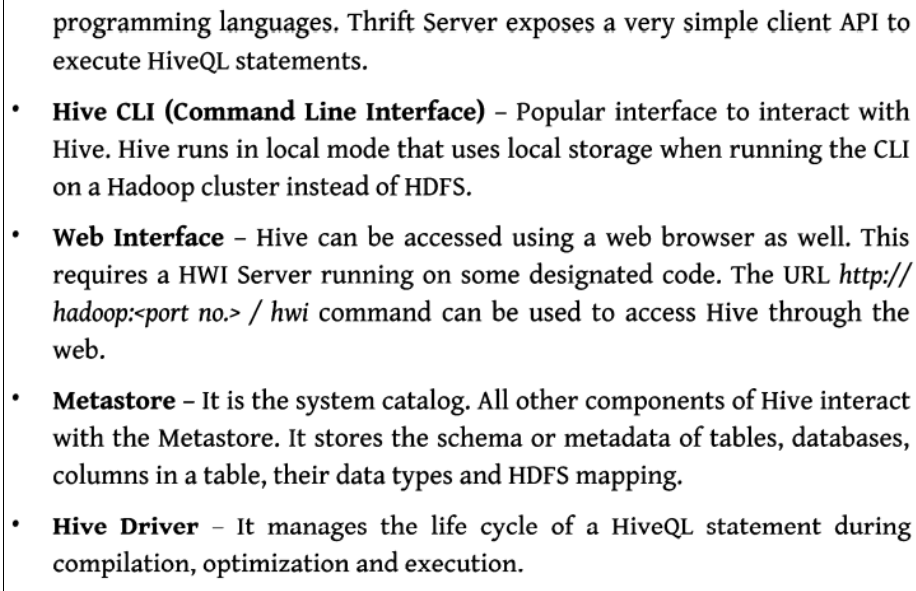
Scheme: Architecture + Data types + File Formats = 6 + 2 + 2 =10M

Solution:



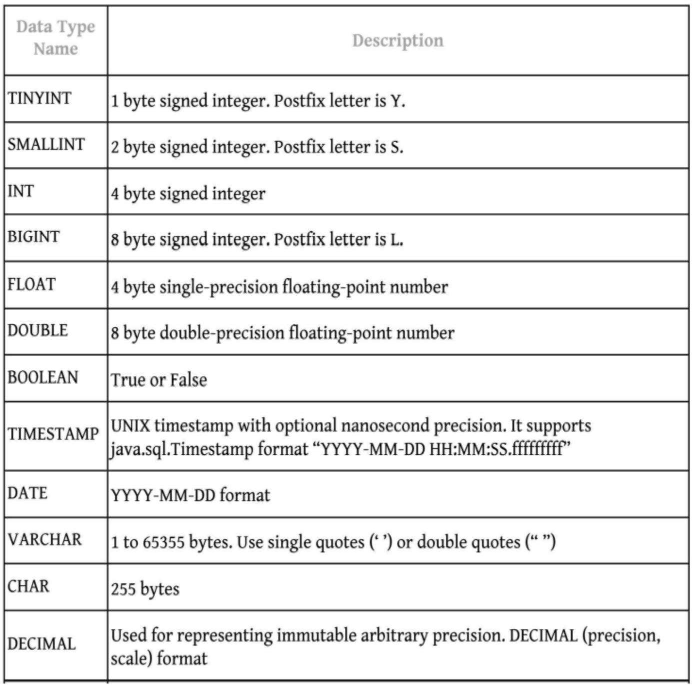
Components of Hive architecture are:

- **Hive Server (Thrift)** – An optional service that allows a remote client to submit requests to Hive and retrieve results. Requests can use a variety of



programming languages. Thrift Server exposes a very simple client API to execute HiveQL statements.

- **Hive CLI (Command Line Interface)** – Popular interface to interact with Hive. Hive runs in local mode that uses local storage when running the CLI on a Hadoop cluster instead of HDFS.
- **Web Interface** – Hive can be accessed using a web browser as well. This requires a HWI Server running on some designated code. The URL *http:// hadoop:<port no.> / hwi* command can be used to access Hive through the web.
- **Metastore** – It is the system catalog. All other components of Hive interact with the Metastore. It stores the schema or metadata of tables, databases, columns in a table, their data types and HDFS mapping.
- **Hive Driver** – It manages the life cycle of a HiveQL statement during compilation, optimization and execution.



Data Type Name	Description
TINYINT	1 byte signed integer. Postfix letter is Y.
SMALLINT	2 byte signed integer. Postfix letter is S.
INT	4 byte signed integer
BIGINT	8 byte signed integer. Postfix letter is L.
FLOAT	4 byte single-precision floating-point number
DOUBLE	8 byte double-precision floating-point number
BOOLEAN	True or False
TIMESTAMP	UNIX timestamp with optional nanosecond precision. It supports java.sql.Timestamp format “YYYY-MM-DD HH:MM:SS.fffffffff”
DATE	YYYY-MM-DD format
VARCHAR	1 to 65355 bytes. Use single quotes (‘ ’) or double quotes (“ ”)
CHAR	255 bytes
DECIMAL	Used for representing immutable arbitrary precision. DECIMAL (precision, scale) format

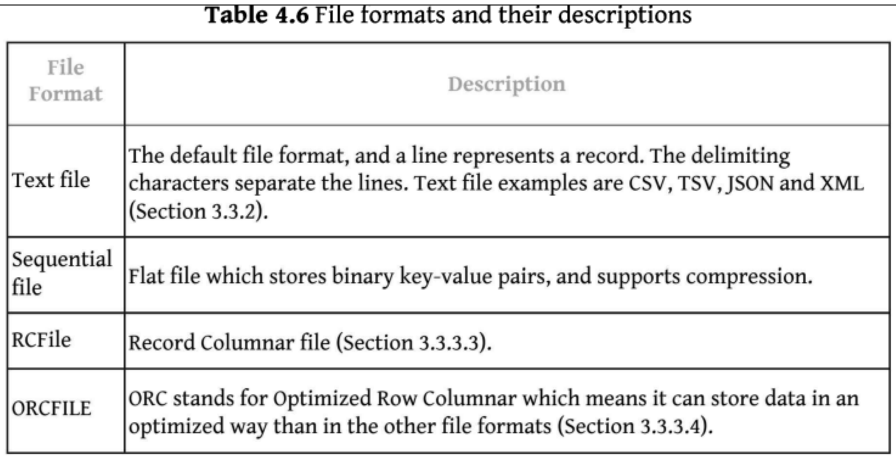


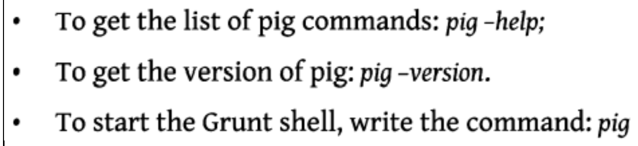
Table 4.6 File formats and their descriptions

File Format	Description
Text file	The default file format, and a line represents a record. The delimiting characters separate the lines. Text file examples are CSV, TSV, JSON and XML (Section 3.3.2).
Sequential file	Flat file which stores binary key-value pairs, and supports compression.
RCFile	Record Columnar file (Section 3.3.3.3).
ORCFILE	ORC stands for Optimized Row Columnar which means it can store data in an optimized way than in the other file formats (Section 3.3.3.4).

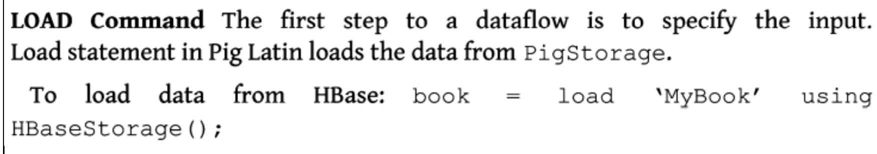
3 **Explain Pig Latin Scripting Commands with example**. [10] CO4 L2

Scheme: Commands = 10M

Solution:

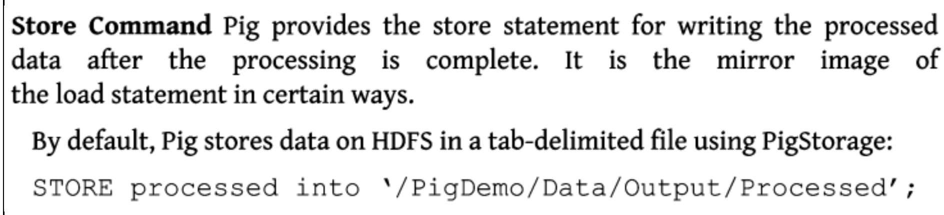


- To get the list of pig commands: *pig –help;*
- To get the version of pig: *pig –version.*
- To start the Grunt shell, write the command: *pig*



LOAD Command The first step to a dataflow is to specify the input. Load statement in Pig Latin loads the data from `PigStorage`.

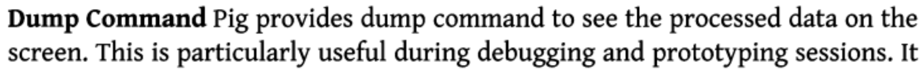
To load data from HBase: `book = load 'MyBook' using HBaseStorage();`



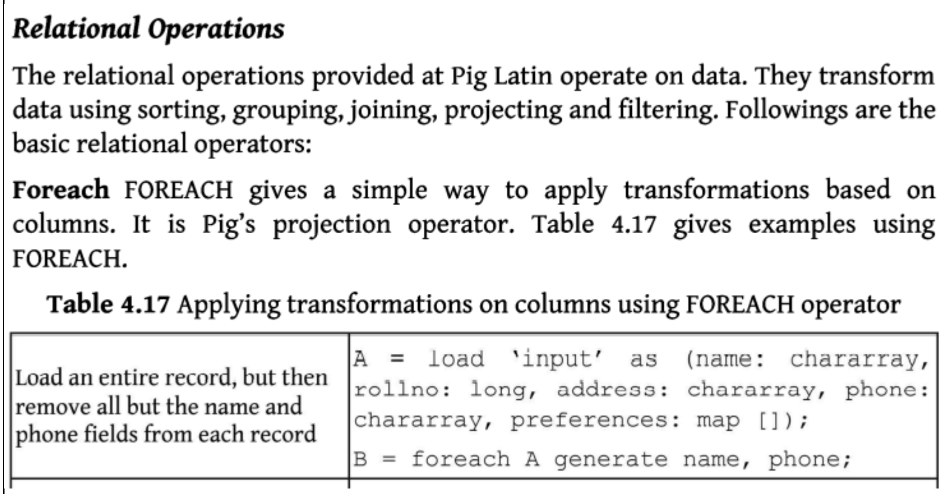
Store Command Pig provides the store statement for writing the processed data after the processing is complete. It is the mirror image of the load statement in certain ways.

By default, Pig stores data on HDFS in a tab-delimited file using PigStorage:

```
STORE processed into '/PigDemo/Data/Output/Processed';
```



Dump Command Pig provides dump command to see the processed data on the screen. This is particularly useful during debugging and prototyping sessions. It



Relational Operations

The relational operations provided at Pig Latin operate on data. They transform data using sorting, grouping, joining, projecting and filtering. Followings are the basic relational operators:

Foreach FOREACH gives a simple way to apply transformations based on columns. It is Pig's projection operator. Table 4.17 gives examples using FOREACH.

Table 4.17 Applying transformations on columns using FOREACH operator

Load an entire record, but then remove all but the name and phone fields from each record	`A = load 'input' as (name: chararray, rollno: long, address: chararray, phone: chararray, preferences: map []);`
	`B = foreach A generate name, phone;`

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

Loads an entire record, then selects the tuples with marks more than 75 from each record	`A = load ‘input’ as (name:chararray, rollno:long, marks:float);`
	`B = filter A by marks > 75.0;`

Group GROUP statement collects records with the same key. There is no direct connection between group and aggregate functions in Pig Latin unlike SQL.

Collects all records with the same value for the provided key into a bag. Then it can pass to aggregate function, if required or do other things with that.	`A = load ‘input’ as (name: chararray, rollno:long, marks: float);`
	`grpd = group A by marks;`
	`B = foreach grpd generate name, COUNT(A);`

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

The syntax of order is similar to group.
Key indicates by which the data sort.
	`A = load ‘input’ as (name: chararray, rollno: long, marks: float);`
	`B = order A by name;`

Distinct DISTINCT removes duplicate tuples. It works only on entire tuples, not on individual fields:

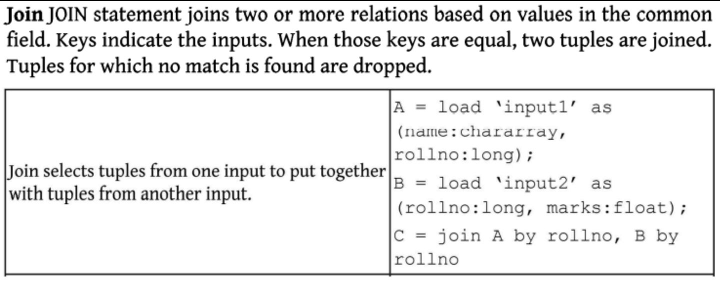
Removes the tuples having the same name and city.	`A = load ‘input’ as (name: chararray, city: chararray);`
	`B = distinct A;`

Limit LIMIT gets the limited number of results.

Outputs only first five tuples from the relation.	`A = load ‘input’ as (name: chararray, city: chararray);`
	`B = Limit A 5;`

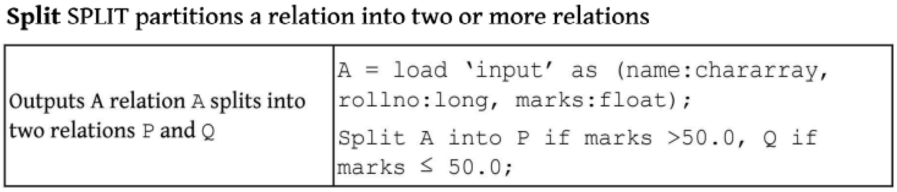
Sample SAMPLE offers to get a sample of the entire data. It reads through all of the data but returns only a percentage of rows on random basis. Thus, results of a script with sample will vary with every execution. The percentage it will return is expressed as a double value, between 0 and 1. For example, 0.2 indicates 20%.

Outputs only 10% tuples from the relation	`A = load ‘input’ as (name:chararray, city: chararray);`
	`B = sample A 0.1;`



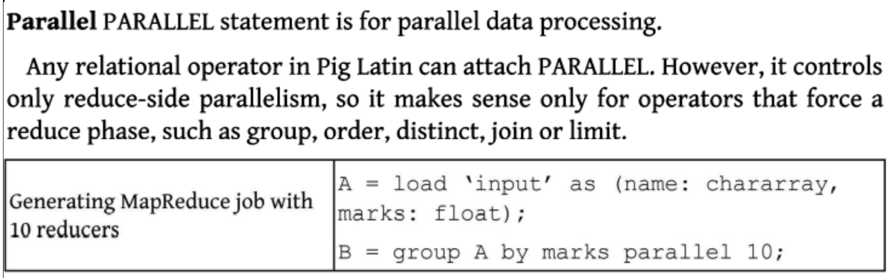
Join JOIN statement joins two or more relations based on values in the common field. Keys indicate the inputs. When those keys are equal, two tuples are joined. Tuples for which no match is found are dropped.

Join selects tuples from one input to put together with tuples from another input.	`A = load 'input1' as (name:chararray, rollno:long);`
	`B = load 'input2' as (rollno:long, marks:float);`
	`C = join A by rollno, B by rollno`



Split SPLIT partitions a relation into two or more relations

Outputs A relation A splits into two relations P and Q	`A = load 'input' as (name:chararray, rollno:long, marks:float);`
	`Split A into P if marks >50.0, Q if marks ≤ 50.0;`



Parallel PARALLEL statement is for parallel data processing.

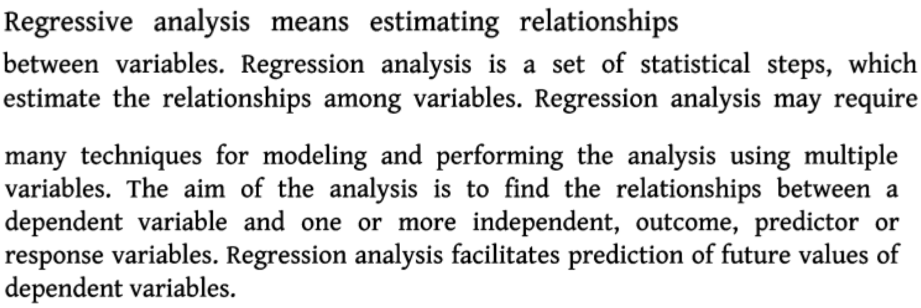
Any relational operator in Pig Latin can attach PARALLEL. However, it controls only reduce-side parallelism, so it makes sense only for operators that force a reduce phase, such as group, order, distinct, join or limit.

Generating MapReduce job with 10 reducers	`A = load 'input' as (name: chararray, marks: float);`
	`B = group A by marks parallel 10;`

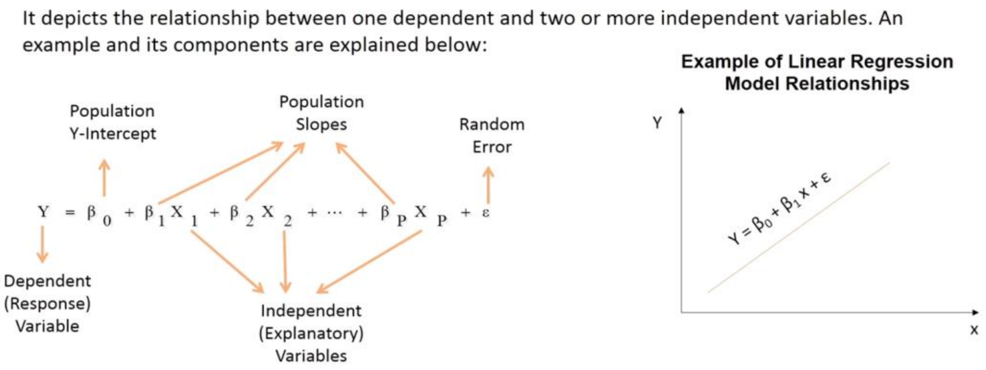
4 **Write in detail about Regression Analysis.** [10] CO5 L2

Scheme: Regression + types = 6 + 4 =10M

Solution:



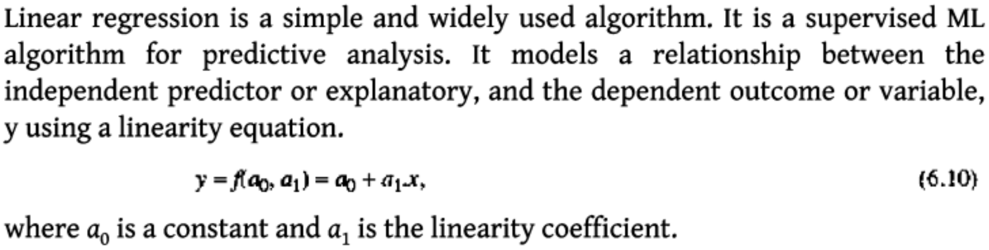
Regressive analysis means estimating relationships between variables. Regression analysis is a set of statistical steps, which estimate the relationships among variables. Regression analysis may require many techniques for modeling and performing the analysis using multiple variables. The aim of the analysis is to find the relationships between a dependent variable and one or more independent, outcome, predictor or response variables. Regression analysis facilitates prediction of future values of dependent variables.



It depicts the relationship between one dependent and two or more independent variables. An example and its components are explained below:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_P X_P + \varepsilon$$

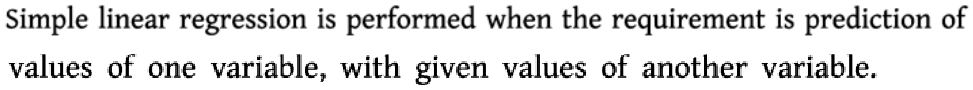
Simple Linear Regression:



Linear regression is a simple and widely used algorithm. It is a supervised ML algorithm for predictive analysis. It models a relationship between the independent predictor or explanatory, and the dependent outcome or variable, y using a linearity equation.

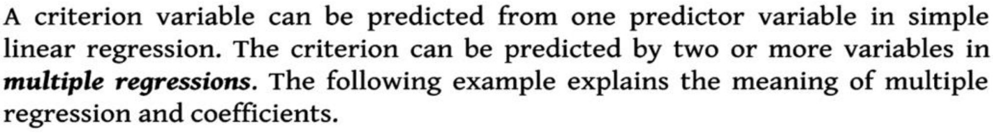
$$y = f(a_0, a_1) = a_0 + a_1.x, \tag{6.10}$$

where a_0 is a constant and a_1 is the linearity coefficient.

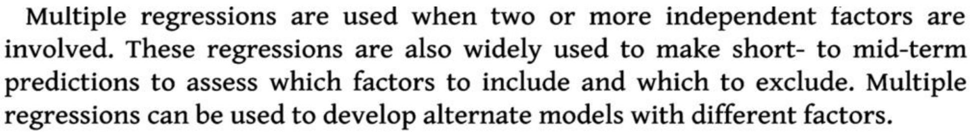


Simple linear regression is performed when the requirement is prediction of values of one variable, with given values of another variable.

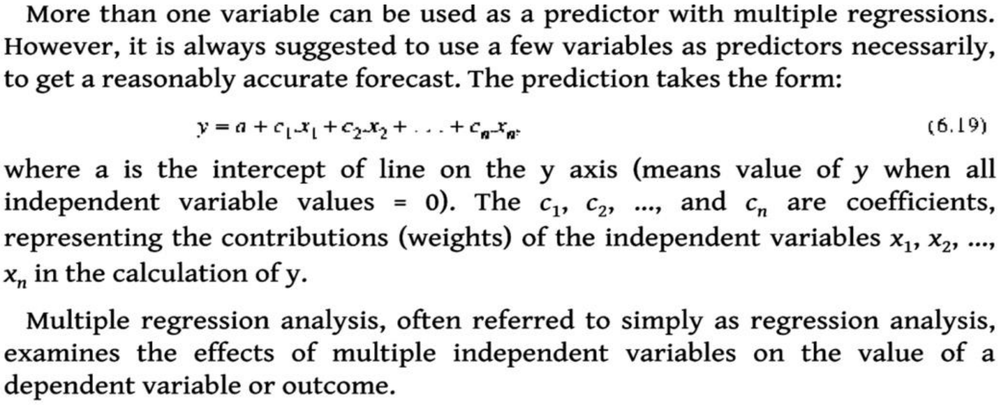
Multiple Regressions:



A criterion variable can be predicted from one predictor variable in simple linear regression. The criterion can be predicted by two or more variables in ***multiple regressions***. The following example explains the meaning of multiple regression and coefficients.



Multiple regressions are used when two or more independent factors are involved. These regressions are also widely used to make short- to mid-term predictions to assess which factors to include and which to exclude. Multiple regressions can be used to develop alternate models with different factors.



More than one variable can be used as a predictor with multiple regressions. However, it is always suggested to use a few variables as predictors necessarily, to get a reasonably accurate forecast. The prediction takes the form:

$$y = a + c_1.x_1 + c_2.x_2 + \ldots + c_n.x_n. \tag{6.19}$$

where a is the intercept of line on the y axis (means value of y when all independent variable values = 0). The c_1, c_2, ..., and c_n are coefficients, representing the contributions (weights) of the independent variables x_1, x_2, ..., x_n in the calculation of y.

Multiple regression analysis, often referred to simply as regression analysis, examines the effects of multiple independent variables on the value of a dependent variable or outcome.

5

For the following table describe the different steps of forming Association rule using Apriori algorithm.
Scheme: Apriori Algorithm + Association Rule = 6 + 2 =10M

Solution:

S. No	TRANSACTION LIST			
1	MILK	EGG	BREAD	BUTTER
2	MILK	BUTTER	EGG	KETCHUP
3	BREAD	BUTTER	KETCHUP	
4	MILK	BREAD	BUTTER	
5	BREAD	BUTTER	COOKIES	
6	MILK	BREAD	BUTTER	COOKIES
7	MILK	COOKIES		
8	MILK	BREAD	BUTTER	
9	BREAD	BUTTER	EGG	COOKIES
10	MILK	BUTTER	BREAD	
11	MILK	BREAD	BUTTER	
12	MILK	BREAD	COOKIES	KETCHUP

Consider Support Count: 4

S. No	List	Support
1	MILK	9
2	EGG	3
3	BREAD	10
4	BUTTER	10
5	KETCHUP	3
6	COOKIES	5

After pruning itemset those who have less than support count value:

S. No	List	Support
1	MILK	9
2	BREAD	10
3	BUTTER	10
4	COOKIES	5

Join step to form candidate 2 itemset:

S. No	List	Support
1	Milk, EGG	2
2	MILK, BREAD	7
3	MILK, BUTTER	7

[10]

CO5

L3

4	Milk, KETCHUP	2
5	Milk, COOKIES	2
6	EGG, BREAD	2
7	EGG, BUTTER	3
8	EGG, KETCHUP	1
9	EGG, COOKIES	0
10	BREAD, BUTTER	9
11	BREAD, KETCHUP	1
12	BREAD, COOKIES	2
13	BUTTER, KETCHUP	2
14	BUTTER, COOKIES	3
15	KETCHUP, COOKIES	1

After pruning itemset those who have less than support count value:

S. No	List	Support
1	MILK, BREAD	7
2	MILK, BUTTER	7
3	BREAD, BUTTER	9

Join step to form candidate 3 itemset:

S. No	List	Support
1	MILK, BREAD, BUTTER	6

Final selected frequent itemset is **{MILK, BREAD, BUTTER}**

{MILK,BREAD}->{BUTTER} = support(Milk, BREAD, BUTTER) / support(MILK,BREAD) = 6/7*100=86%

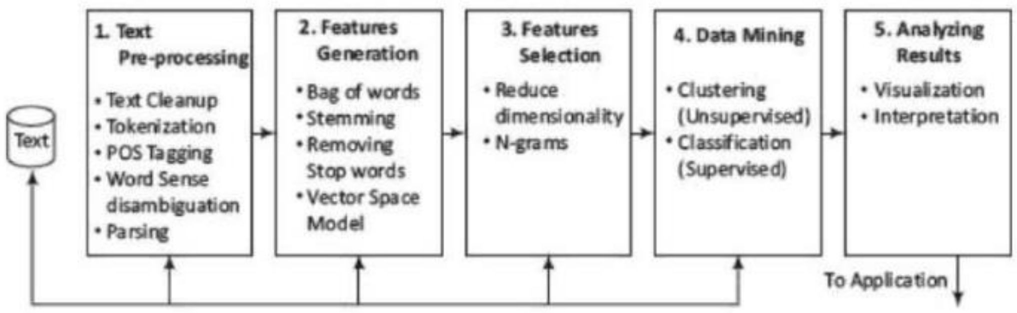
{MILK, BUTTER}->{BREAD} = support(Milk, BREAD, BUTTER) / support(MILK,BUTTER} = 6/7*100 = 86%

{BREAD, BUTTER}->{MILK} = support(Milk, BREAD, BUTTER) / support(BREAD,BUTTER} = 6/9*100 = 67%

{MILK}->{BREAD,BUTTER} = support(Milk, BREAD, BUTTER) / support(MILK) = 6/9*100=67%

{BREAD}->{MILK,BUTTER} = support(Milk, BREAD, BUTTER) / support(BREAD) = 6/10*100=60%

{BUTTER}->{MILK,BREAD} = support(Milk, BREAD, BUTTER) / support(BUTTER) = 6/10*100=60%

6	**Write short note on Text Mining and explain in detail Process Phases of Text Mining.** **Scheme: Phases + Diagram = 7 + 3 =10M** **Solution:** Text Mining is the process of deriving high quality information from text.  Text 1. Text Pre-processing • Text Cleanup • Tokenization • POS Tagging • Word Sense disambiguation • Parsing 2. Features Generation • Bag of words • Stemming • Removing Stop words • Vector Space Model 3. Features Selection • Reduce dimensionality • N-grams 4. Data Mining • Clustering (Unsupervised) • Classification (Supervised) 5. Analyzing Results • Visualization • Interpretation To Application • The five phases for processing text are as follows: • Phase 1: Text pre-processing enables Syntactic/Semantic text-analysis and does the followings: • Text *cleanup* is a process of removing unnecessary or unwanted information. Text cleanup converts the raw data by filling up the missing values, identifies and removes outliers, and resolves the inconsistencies. For example, removing comments, removing or escaping "%20" from URL for the web pages or cleanup the typing error, such as teh (the), don't (do not) [%20 specifies space in a URL]. • *Tokenization* is a process of splitting the cleanup text into tokens (words) using white spaces and punctuation marks as delimiters. • *Part of Speech (POS) tagging* is a method that attempts labeling of each token (word) with an appropriate POS. Tagging helps in recognizing names of people, places, organizations and titles. English language set includes the noun, verb, adverb, adjective, prepositions and conjunctions. Part of Speech encoded in the annotation system of the Penn Treebank Project has 36 POS tags.4 • *Word sense disambiguation* is a method, which identifies the sense of a word used in a sentence; that gives meaning in case the word has multiple meanings. The methods, which resolve the ambiguity of words can be context or proximity based. Some examples of such words are bear, bank, cell and bass. • *Parsing* is a method, which generates a parse-tree for each sentence. Parsing attempts and infers the precise grammatical relationships between different words in a given sentence.	[10]	CO5	L2

Phase 2: Features Generation is a process which first defines features (variables, predictors). Some of the ways of feature generations are:

1. *Bag of* words-Order of words is not that important for certain applications. Text document is represented by the words it contains (and their occurrences). Document classification methods commonly use the bag-of-words model. Document classification methods then use the occurrence (frequency) of each word as a feature for training a classifier. Algorithms do not directly apply on the bag of words, but use the frequencies.
2. Stemming-identifies a word by its root.
    - Normalizes or unifies variations of the same concept, such as *speak* for three variations, i.e., speaking, speaks, speakers denoted by [speaking, speaks, speaker -+ speak]
    - Removes plurals, normalizes verb tenses and remove affixes.
3. Stemming reduces the word to its most basic element. For example, impurification -+ pure.

4. *Removing stop words* from the feature space-they are the common words, unlikely to help text mining. The search program tries to ignore stop words. For example, ignores *a, at, for,* it, *in* and *are.*
5. *Vector Space Model* (VSM)-is an algebraic model for representing text documents as vector of identifiers, word frequencies or terms in the document index. VSM uses the method of term frequency-inverse document frequency (TF-IDF) and evaluates how important is a word in a document.
6. When used in document classification, VSM also refers to the bag-of-words model. This bag of words is required to be converted into a term-vector in VSM. The term vector provides the numeric values corresponding to each term appearing in a document. The term vector is very helpful in feature generation and selection.

Phase 3: Features Selection is the process that selects a subset of features by rejecting irrelevant and/or redundant features (variables, predictors or dimension) according to defined criteria. Feature selection process does the following:

1. *Dimensionality* reduction-Feature selection is one of the methods of division and therefore, dimension reduction. The basic objective is to eliminate irrelevant and redundant data. Redundant features are those, which provide no extra information. Irrelevant features provide no useful or relevant information in any context.
2. Principal Component Analysis (PCA) and Linear Discriminate Analysis (LDA) are dimension reduction methods. Discrimination ability of a feature measures relevancy of features. Correlation helps in finding the redundancy of the feature. Two features are redundant to each other if their values correlate with each other.
3. *N-gram evaluation-finding* the number of consecutive words of interest and extract them. For example, 2-gram is a two words sequence, ["tasty food", "Good one"]. 3-gram is a three words sequence, ["Crime Investigation Department"].
4. *Noise detection and evaluation of outliers* methods do the identification of unusual or suspicious items, events or observations from the data set. This step helps in cleaning the data.

The feature selection algorithm reduces dimensionality that not only improves the performance of learning algorithm but also reduces the storage requirement for a dataset. The process enhances data understanding and its visualization.

- Phase 4: Data mining techniques enable insights about the structured database that resulted from the previous phases. Examples of techniques are:

1. Unsupervised learning (for example, clustering)
    1. The class labels (categories) of training data are unknown
    2. Establish the existence of groups or clusters in the data

Good clustering methods use high intra-cluster similarity and low inter-cluster similarity. Examples of uses - blogs, patterns and trends.

2. *Supervised learning (for example, classification)*
    1. The training data is labeled indicating the class
    2. New data is classified based on the training set

Classification is correct when the known label of test sample is identical with the resulting class computed from the classification model.

Examples of uses are *news filtering application,* where it is required to automatically assign incoming documents to pre-defined categories; *email spam filtering,* where it is identified whether incoming email messages are spam or not.

Example of text classification methods are *Naive Bayes Classifier* and *SVMs.*

3. Identifying evolutionary patterns in temporal text streams-the method is useful in a wide range of applications, such as summarizing of events in news articles and extracting the research trends in the scientific literature.

Tf/Idf

Phase 5: Analysing results

(i) Evaluate the outcome of the complete process.

(ii) Interpretation of Result- If acceptable then results obtained can be used as an input for next set of sequences. Else, the result can be discarded, and try to understand what and why the process failed.

(iii) Visualization - Prepare visuals from data, and build a prototype.

(iv) Use the results for further improvement in activities at the enterprise, industry or institution.

Faculty Signature CCI Signature HOD Signature