Repository: Apache Pig

General Pig commands

1) LOAD – It loads the data in HDFS

   LOAD 'Input file path' USING function as schema;

2) STORE - It is used to store data into a file

   STORE Relation_name INTO ' required_directory_path ' [USING function];

3) FILTER - It is used to select the required tuples from a relation based on a condition

   Relation2_name = FILTER Relation1_name BY (condition);

4) DISTINCT - It is used to remove redundant (duplicate) tuples from a relation.

   Relation_name2 = DISTINCT Relatin_name1;

5) FOREACH – It is used to generate specified data transformations based on the column data.

   Relation_name2 = FOREACH Relatin_name1 GENERATE (required data);

6) MIN, MAX – They give the minimum and maximum value of the column in a Bag

MIN(expression)
MAX(expression)

7) SUM – It returns the sum of the columns in a Bag. For a global sum, a GROUP ALL should be done

   SUM(expression)

8) COUNT – It counts the no of elements in Bag. It doesn`t count tuples having null in first field. For a global count, GROUP ALL needs t be done

   COUNT(expression)

9) ORDER BY – This is used to display sorted data

   ORDER relation_a BY (ASC|DESC);

10) Tokenize – It splits a string into a tuple and returns Bag of Tuples

    TOKENIZE(string [, 'delimiter'])

11) GROUP – It groups the data in one or more relations. The key should be same for the relations to be grouped

    GROUP = GROUP relation_name BY age;
    GROUP_Multiple = GROUP relation BY (column1, column2)

12) TOP – it fetches the top N tuples fron the Bag

    TOP(topN, column, relation)

13) COGROUP – It groups the data in two or more relations.

COGROUP_DATA = COGROUP relation_1 BY column_name, relation_2 BY column_name;

14) LIMIT – This is used to put limited on the output returned.

LIMIT relation_a x;
X is a Number

Diagnose operators

DUMP – This is used to run the script and display output to the console
DUMP relation_a;

DESCRIBE – This shows the schema of a relation
DESCRIBE relation_a;

Illustrate – This shows how a data is transformed at different steps
Illustrate relation_a;

EXPLAIN – shows the query New logical plan, Physical plan and Execution plan
EXPLAIN relation_a;

STRING functions

ENDSWITH – Compare two strings and to find if the string ends with the second string.
ENDSWITH(first_string, test_string);

STARTSWITH – Compare two strings and to find if the first stirng starts with second string.
STARTSWITH(first_string, test_string);

SUBSTRING – Returns a substring from string. It takes string, startIndex and stopIndex as parameters.
SUBSTRING(string, start_index, stop_index);

UPPER – Converts all strings into uppercase
UPPER(string);

LOWER – Converts all strings into lowercase
LOWER(string);

REPLACE – Replaces characters in a string with new characters. It accepts string, regular expression and new characters as arguments.
REPLACE(string, regex, newChraracters);

STRSPLIT – Splits the string as per the delimiter

STRSPLIT(string, regex_expression, limit);

JOIN

Joins are used to combine the records from relations.

Inner Join – It returns row when there is a match in both the tables

Joined_table = JOIN relation_1 BY column_name, relation2 BY column_name

Full Outer Join – This join returns rows when there is a match in one of the relations

Full_Outer_Join = JOIN relation_1 BY column_name FULL OUTER, relation_2 BY column_name_2

Left Outer Join – This join returns all rows from the left table and matching rows from the right table

Left_Outer_Join = JOIN relation_1 BY column_name LEFT OUTER, relation_2 BY column_name_2

Right Outer Join – This join returns all rows from the right table and matching rows from the left table

Right_Outer_Join = JOIN relation_1 BY column_name RIGHT OUTER, relation_2 BY column_name_2

UDFs in Python

Create a UDF in local storage in cloudxlab. Name it udfs.py

```
@outputSchemaFunction("word:chararray")
def hello():
    return 'Hello, world'
```

check the content using 'cat udfs.py'

Start grunt shell and then type to register the UDF

```
register './udfs.py' using
org.apache.pig.scripting.jython.JythonScriptEngine as myudf;
```

Now you can use this UDF on your dataset

Reference

https://pig.apache.org/docs/r0.9.1/udf.html#python-udfs