PIG Commands

Following files are used for the given examples:

brand.csv

```
B7,Reebok,1
B8,Samsung,1
B9,Apple,1
B10,Motorola,1
B11,XIOMI,1
B12,Zara,0
B13,H&M,0
B14,HP,1
B15,Sony,0
B16,Canon,1
B17,Lenovo,0
```

category.csv

```
C4,C0,Electronics,1
C5,C4,Cameras,0
C6,C4,Laptops,1
C7,C4,Mobile Phone,1
C9,C0,Clothing,1
C10,C9,Men Clothing,1
C12,C9,Women Clothing,1
C13,C4,Sound,1
C15,C9,Top wear,0
```

products.csv

```
P4,C7,B9,Apple iPhone 7,57000,1
P5,C7,B10,Motorola G5 Plus,16999,1
P6,C10,B7,Solid Polo Neck Grey T-Shirt,699,1
P12,C7,B11,Redmi Note 8 Pro,14999,1
P14,C12,B12,Mohnish Fashion,1500,1
P15,C12,B12,Multicolor Dress,800,1
P16,C12,B13,silk Saree,8000,1
P18,C13,B15,Webster,439,1
P19,C6,B14,Laptop,54000,1
P21,C13,B15,Headset,1500,1
P22,C6,B14,HP Pavilion,45000,1
P26,C15,B13,Saree,7899,0
```

PIG LOAD

Syntax:

```
relation = LOAD 'Input file path' USING function as schema;
```

E.g.

```
products = LOAD 'product.csv' using PigStorage(',') as
(pid:chararray,cid:chararray,bid:chararray,name:chararray,price:double
,isactive:int);

category = LOAD 'category.csv' using PigStorage(',') as
(cid:chararray, cpid:chararray, name:chararray, isactive:int);

brands = LOAD 'brand.csv' using PigStorage(',') as (bid:chararray,
name:chararray, isactive:int);
```

PIG DUMP

Syntax:

```
DUMP relation;
```

Example(s):

```
DUMP products;
DUMP categories;
DUMP brands;
```

PIG STORAGE

Syntax:

```
STORE relation INTO 'Output File Path' USING function;
```

Example(s)

```
STORE brands INTO 'brand_copy.csv' USING PigStorage('\t');
```

PIG FOREACH..GENERATE

Syntax:

```
relation = FOREACH relation GENERATE (required data);
```

Example:

```
p1 = FOREACH products GENERATE (pid, name, price);
p1 = FOREACH products GENERATE ($0, $3, $4);
```

PIG FILTER

Syntax:

```
relation = FILTER relation BY (condition);
```

Example:

```
p2 = FILTER products by price>50000.0;
p2 = FILTER products by isactive==1;
```

PIG DISTINCT

Syntax:

```
relation = DISTINCT relation;
```

Example:

```
p2 = DISTINCT products;
```

PIG SORTING

Syntax:

```
relation = ORDER relation BY column (ASC|DESC);
```

Example:

```
p2 = ORDER products BY price;
p2 = ORDER products by cid, bid;
```

PIG LIMIT

Syntax:

```
relation = LIMIT relation #number_of_records;
```

Example:

```
p2 = ORDER products BY price;
p2 = ORDER products by cid, bid;
```

PIG GROUP

Syntax:

```
relation = GROUP relation BY column;
```

Example:

```
p2 = GROUP products BY bid;
p2 = GROUP products BY cid, bid;
```

PIG JOIN

Syntax (Equi Join):

```
relation = JOIN relation BY column, relation BY column;
```

Example:

```
p2 = JOIN products BY cid, categories BY cid;
```

Syntax (Left Out Join)

```
relation = JOIN relation BY column LEFT OUTER, relation BY column;
```

Example:

```
p2 = JOIN products BY cid LEFT OUTER, categories BY cid;
```

Syntax (Right Out Join)

```
relation = JOIN relation BY column RIGHT OUTER, relation BY column;
```

Example:

```
p2 = JOIN products BY cid RIGHT OUTER, categories BY cid;
```

Syntax (Left Out Join)

```
relation = JOIN relation BY column FULL OUTER, relation BY column;
```

Example:

```
p2 = JOIN products BY cid FULL OUTER, categories BY cid;
```

PIG FUNCTIONS

1. AGGREGATE

A. COUNT
B. MAX
C. MIN
D. AVG
E. SUM
F. COUNT_STAR

Syntax:

```
COUNT(expression)
MAX(expression)
MIN(expression)
AVG(expression)
SUM(expression)
COUNT_STAR(relation)
```

Example:

```
g1 = GROUP products BY bid;
c1 = foreach g1  Generate COUNT(products.price), MAX(products.price),
MIN(products.price), AVG(products.price);

g1 = GROUP products ALL;
c1 = foreach g1  Generate COUNT(products.price), MAX(products.price),
MIN(products.price), AVG(products.price);
```

2. MATH

A. ABS
B. CEIL
C. FLOOR
D. ROUND
E. SQRT
F. RANDOM

Syntax:

```
ABS(expression)
CEIL(expression)
FLOOR(expression)
ROUND(expression)
SQRT(expression)
RANDOM()
```

Example:

```
g1 = GROUP products BY bid;
c1 = foreach products Generate ABS(products.price),
CEIL(products.price), FLOOR(products.price), ROUND(products.price),
RANDOM()
```

3. STRING

A. LCFIRST
B. UCFIRST
C. UPPER
D. LOWER

E. TRIM
F. LTRIM
G. RTRIM
H. INDEXOF
I. LASTINDEXOF
J. SUBSTRING
K. STARTSWITH
L. ENDSWITH
M. TOKENIZE

Syntax:

```
LCFIRST(string)
UCFIRST(string)
UPPER(string)
LOWER(string)
TRIM(string)
LTRIM(string)
RTRIM(string)
INDEXOF(string,value,start)
LASTINDEXOF(string,value)
SUBSTRING(string,start,stop)
STARTSWITH(string, value)
ENDSWITH(string, value)
TOKENIZE(string)
```

Example:

```
c1 = foreach products  Generate LCFIRST(name), UCFIRST(name),
UPPER(name)

t1 = FOREACH products GENERATE TOKENIZE(name);
```

4. DATETIME

A. TODATE()
B. CURRENTTIME()
C. GETDAY()
D. GETMONTH()
E. GETYEAR()
F. GETHOUR()
G. GETMINUE()
H. ADDDURATION()
I. SUBTRACTDURATION()

J. DAYSBETWEEN()
K. MONTHSBETWEEN()
L. YEARSBETWEEN()
M. HOURSBETWEEN()
N. MINUTESBETWEEN()

Syntax:

```
TODATE(string,formate)
Currenttime()
GETDAY(datetime)
GETMONTH(datetime)
GETYEAR(datetime)
GETHOUR(datetime)
GETMINUTE(datetime)
ADDDURATION(datetime, duration)
SUBTRACTDURATION(datetime, duration)
DAYSBETWEEN(datetime, datetime)
MONTHSBETWEEN(datetime, datetime)
YEARSBETWEEN(datetime, datetime)
HOURSBETWEEN(datetime, datetime)
MINUTESBETWEEN(datetime, datetime)
```

5 MISCELLENEOUS FUNCTIONS

A. TOTUPLE
B. TOBAG
C. TOMAP

Syntax:

```
TOTUPLE(val1,val2,val3,…)
TOBAG(val1,val2,val3,…)
TOMAP(key1,val1,key2,val2,…)
```

Example:

```
e1 = FOREACH products GENERATE TOTUPLE(id,name,price)
e2 = FOREACH products GENERATE TOBAG(id,name,price)
e3 = FOREACH products GENERATE TOMAP(name,price)
```

6. MATH

A. PIGSTORAGE
B. TEXTLOADER
C. BINSTORAGE

Syntax:

```
PIGSTORAGE(delimiter)
TEXTLOADER()
BINSTORAGE()
```

Example:

```
r1 = products = LOAD 'product.csv' using PigStorage(',') as
(pid:chararray,cid:chararray,bid:chararray,name:chararray,price:double
,isactive:int);

r2 = LOAD 'product.csv' using TextLoader() as (record:chararray);

r3 = LOAD 'product.csv' using BinStorage();
```