

Projekcje, przekształcenia

Zapis z tabeli A do tabeli B kolumn col1 i col2	B = foreach A generate col1, col2;
Zapis z tabeli A do tabeli B kolumn col1 i col2 (odniesienie po numerze kolumny)	B = foreach A generate \$0,\$1;
Zapis z tabeli A do tabeli B kolumn od col1 do col2 (odniesienie po numerze kolumny)	B = foreach A generate \$0..\$1;
Zapis z tabeli A do tabeli B kolumn do col2 (odniesienie po numerze kolumny)	B = foreach A generate ..\$1;
A = load 'input.tsv' as (col1:chararray, col2:long, col3:int); B = ...	

Ograniczanie liczby wierszy

Próbka 50% z tabeli A	B = sample A 0.5;
10 wierszy z tabeli A	B = limit A 10;

Filtrowanie wierszy

Odfiltrowanie wierszy z wartościami większymi niż zadana	B = filter A by year>2000;
j.w., rozbudowany przykład	B = filter A by (year > 2000 and year < 2010);
Odfiltrowanie wierszy z wartościami niepustymi	B = filter A by descri- ption is not null;
Odfiltrowanie wierszy z tekstami spełni- ającymi zadane wyrażenie regularne	B = filter A by text MATCHES '.Miami.';

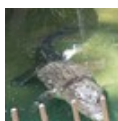
Operacje masowe

Grupowanie wierszy mających tę samą wartość kolumny col1	B = group A by col1;
Grupowanie wierszy mających te same wartości kolumn col1 i col2	B = group A by (col1,col2);
Grupowanie wszystkich wierszy	B = group A all;
Sortowanie wierszy rosnąco wg. wartości kolumny col1	B = sort A by col1 asc;
Sortowanie wierszy malejąco wg. wartości kolumny col1	B = sort A by col1 desc;
Rankingowanie wierszy rosnąco wg. wartości col1	B = rank A by col1 asc;

Operacje na kolumnach zawierających tekst

Podział tekstu na słowa	B = foreach A generate id, TOKENIZE(txt) as tokens;	bag:{{chararray}} TOKENIZE(char- array)
Podział tekstu na określoną liczbę fragmentów względem znaku podziału	B = foreach A generate id, STRSPLIT(txt,',',2) as tup;	tuple STRSPLIT(- chararray,chara- rray,int)
Usunięcie wiodących i kończących białych znaków	B = foreach A generate TRIM(txt) as txt;	chararray TRIM(c- hararray)
Wyciągnięcie tekstu spełniającego zadane wyrażenie regularne	B = foreach A generate REGEX_ EXTRACT(txt,'([0- 9]+)',1) as txt;	chararray REGEX_EXTRAC- T(chararray, regex, group num)

Operacje przekształcające kolumny tekstowe



Splaszczanie kolekcji

```
B = group A by col1;
-- B:{group, A:{ (co ll)}}
C = foreach B generate group, flatte n(A);
-- C:{group, col1}
D = foreach B generate A as bag1, A as bag2;
E = foreach D generate FLATTE N(bag1) as t1,
FLATTE N(bag1) as t2;
-- E:{t1,t2}
-- wiersze w tabeli E reprez entują wszystkie
możliwe pary obiektów z bag1 oraz bag2 (produkt
kartez jański)
```

Import skryptow z makrami

```
import 'macros.pig';
A = load ...
B = MACRO_ FRO M_ O THE R_ F ILE(A);
```

Pig umożliwia zaimportowanie do skryptu Pigowego innych skryptów Pigowych, które np. zawierają makra. Skrypty importowane NIE MOGĄ ZAWIERAĆ operacji REGISTER.

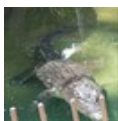
Operacja na grupach wierszy z kolumnami liczbowymi

```
A = load 'input.tsv' as (id1:long,val:long);
B = group A all;
-- wartość średnia, maksymalna, minimalna dla
kolumny val
C = foreach B generate AVG(A.v al);
D = foreach B generate MAX(A.v al);
E = foreach B generate MIN(A.v al);
-- liczba wierszy w tabeli A.
-- wszystkie wiersze z tabeli A zostały umiesz -
czone w jednym bag-u o nazwie A umiesz czonym w
wierszu w tabeli B.
-- Wynik zliczenia elementów z bag-a zostanie
umiesz czony w tabeli C
C = foreach B generate COUNT(A) as cnt;
```

Makra

```
DEFINE COUNTALL(intable) RETURNS out{
  grpd = group $intable all;
  $out = foreach grpd generate COUNT( $in table);
}
A = load ...
B = COUNTA LL(A);
```

Pig umożliwia zdefiniowanie części kodu jako makra i odwoływanie się do niego przez nazwę



By **pdendek**
cheatography.com/pdendek/

Not published yet.
 Last updated 12th May, 2016.
 Page 2 of 2.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>