Интеллектуальные информационные системы

Apache Pig – решение задач

Кафедра управления и интеллектуальных технологий НИУ «МЭИ»

2023 г.

WordCount

```
textLines = LOAD `file.txt` AS (line: chararray);

tokens = FOREACH textLines GENERATE FLATTEN(TOKENIZE(line)) AS token;

filtered_words = FILTER tokens BY token  MATCHES `\\w+`;

group_words = GROUP filtered_words BY token;

word_count = FOREACH group_words  GENERATE COUNT(filtered_words) AS

count, group AS word;

ordered_word_count = ORDER word_count BY count DESC;

STORE  ordered_word_count INTO `word_count_pig`;
```

Regular Expressions

Регулярные выражения (*regular expressions, RegExp*) — формальный язык, используемый в компьютерных программах, работающих с текстом, для поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании шаблона (*pattern),* состоящего из символов и метасимволов и задающего правило поиска.

Подробнее:

https://ru.wikipedia.org/wiki/Регулярные_выражения

https://habr.com/ru/articles/545150/

Regular Expressions

Символы в шаблоне представляют сами себя за исключением спец.символов (метасимволов):
[ ] \ ^ $. | ? * + ( ) { }

\d	Цифровой символ
\D	Нецифровой символ
\s	Пробельный символ
\S	Непробельный символ
\w	Буквенный или цифровой символ или знак подчёркивания; буквы ограничены латиницей
\W	Любой символ, кроме буквенного или цифрового символа или знака подчёркивания

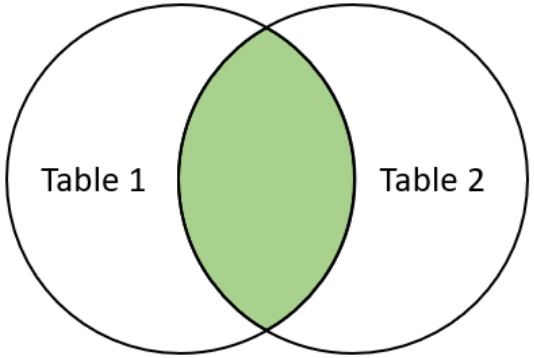
Примеры:
Выражение вида ^\S.* будет находить строки, начинающиеся с непробельного символа

Выражение вида \w+ будет находить и выделять отдельные слова на английском языке

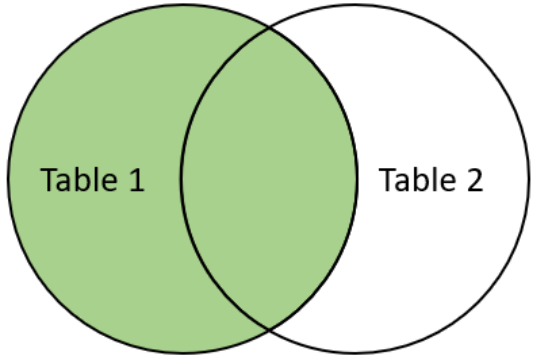
JOIN

- Pig поддерживает
  - Inner Joins (по-умолчанию)
  - Outer Joins
  - Full Joins
- Фазы join
  - Загрузить записи в bag из input #1
  - Загрузить записи в bag из input #2
  - Сделать join для двух массивов данных (bags) по заданному ключу

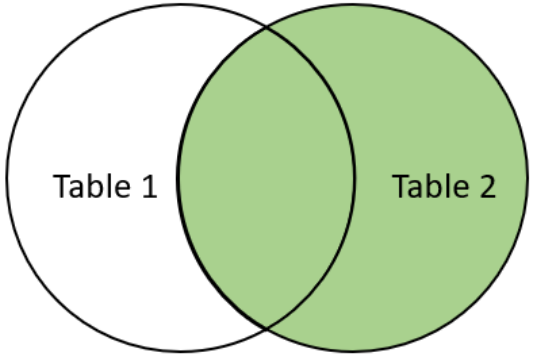
JOIN



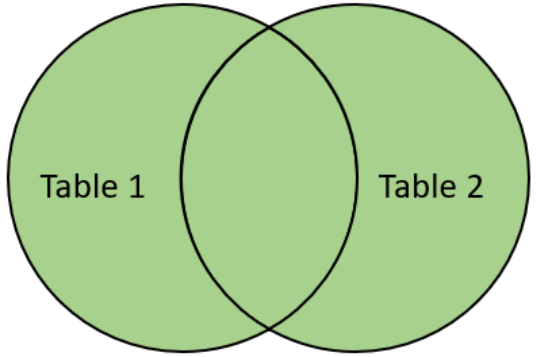
INNER JOIN



LEFT JOIN



RIGHT JOIN



FULL JOIN

Inner join

Users

1	Mathew21
2	Julia49
3	Stue55
4	Jonny87

Visits

1	yandex.ru
2	yandex.ru
1	vk.com
3	mail.ru
2	stackoverflow.com
5	stackoverflow.com

```
users = LOAD 'users.txt' AS (userid: long, nickname: chararray);
visits = LOAD 'visits.txt' AS (userid: long, site: chararray);
visitInfo = JOIN users BY userid, visits BY userid;
DUMP visitInfo;
```

(1,Mathew21,1,vk.com)
(1,Mathew21,1,yandex.ru)
(2,Julia49,2,stackoverflow.com)
(2,Julia49,2,yandex.ru)
(3,Stue55,3,mail.ru)

Inner join

(1,Mathew21,1,vk.com)
(1,Mathew21,1,yandex.ru)
(2,Julia49,2,stackoverflow.com)
(2,Julia49,2,yandex.ru)
(3,Stue55,3,mail.ru)

```
DESCRIBE visitInfo;
visitInfo: {users::userid: long,
        users::nickname: chararray,
        visits::userid: long,
        visits::site: chararray}
```

Outer join

JOIN <bag #1> BY <join field>
LEFT OUTER, <bag #2> **BY** <join field>;

JOIN <bag #1> BY <join field>
RIGHT OUTER, <bag #2> **BY** <join field>;

JOIN <bag #1> BY <join field>
FULL OUTER, <bag #2> **BY** <join field>;

Left outer join

```
users = LOAD 'users.txt' AS (userid: long, nickname: chararray);
visits = LOAD 'visits.txt' AS (userid: long, site: chararray);
visitInfo = JOIN users BY userid LEFT OUTER, visits BY userid;
DUMP visitInfo;
```

(1,Mathew21,1,vk.com)
(1,Mathew21,1,yandex.ru)
(2,Julia49,2,stackoverflow.com)
(2,Julia49,2,yandex.ru)
(3,Stue55,3,mail.ru)
(4,Jonny87,,)

Right outer join

```
users = LOAD 'users.txt' AS (userid: long, nickname: chararray);
visits = LOAD 'visits.txt' AS (userid: long, site: chararray);
visitInfo = JOIN users BY userid RIGHT OUTER, visits BY userid;
DUMP visitInfo;
```

(1,Mathew21,1,vk.com)
(1,Mathew21,1,yandex.ru)
(2,Julia49,2,stackoverflow.com)
(2,Julia49,2,yandex.ru)
(3,Stue55,3,mail.ru)
(,,5,stackoverflow.com)