

2019.01.17 (PIG)

2019년 1월 17일 목요일 오전 9:23

Pig - 데이터 분석

- 대규모 데이터 세트에서 샘플용 데이터 추출
- ETL(Extract, Transform, Load) 작업 추출
- 데이터를 탐색하는 데이터 흐름언어와 실행환경
- 피그 라틴이라는 데이터세트 플로우 제어 언어사용
- 내부 인터프린터에 의해 맵리듀스 작업으로 변환

=====

- **int** : 32비트 부호화 정수 42
- **long** : 64비트 부호화 정수 42L
- **float** : 32비트 부동 소수점 수 3.1415f
- **double** : 64비트 부동 소수점 수 2.7182818
- **chararray** : Unicode UTF-8 형식의 문자 배열 (String)
- **bytearray** : 바이트 배열(바이너리 객체)

1. 실행

2. 명령

Loading and Storing

LOAD It loads the data from a file system into a relation.

STORE It stores a relation to the file system (local/HDFS).

Filtering

FILTER There is a removal of unwanted rows from a relation.

DISTINCT We can remove duplicate rows from a relation by this operator.

FOREACH, GENERATE It transforms the data based on the columns of data.

STREAM To transform a relation using an external program.

Grouping and Joining

JOIN We can join two or more relations.

COGROUP There is a grouping of the data into two or more relations.

GROUP It groups the data in a single relation.

CROSS We can create the cross product of two or more relations.

Sorting

ORDER It arranges a relation in an order based on one or more fields.

LIMIT We can get a particular number of tuples from a relation.

Combining and Splitting

UNION We can combine two or more relations into one relation.

SPLIT To split a single relation into more relations.

Diagnostic Operators

DUMP It prints the content of a relationship through the console.

DESCRIBE It describes the schema of a relation.

EXPLAIN We can view the logical, physical execution plans to evaluate a relation.

ILLUSTRATE It displays all the execution steps as the series of statements.

3. 스키마

```
xa = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XA.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
xb = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XB.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
xc = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XC.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
xd = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XD.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
xe = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XE.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
```

```
xf = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XF.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
xg = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XG.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
xh = load '/user/hadoop/innisfreeData/Innisfree_pdt_info_XH.csv' using PigStorage(',') as (pdtName : chararray, category : chararray, line : chararray, Volume : chararray, price : int, userGrade : int, userid : chararray , userSex : chararray, userAge : int, userDate : datetime, reviewPoint : int, reviewTxt : chararray);
```

Hive..

(0:pdtName, 1:category, 2:line, 3:Volume, 4:price, 5:userGrade, 6:userid ,7: userSex, 8:userAge , 9:userDate , 10:reviewPoint ,11: reviewTxt)
(XA : 수분, XB : 보습, XF : 진정, XH : 각질, XE : 모공, XC : 트러블, XG : 화이트닝, XD : 안티에이징)

- 30대 이상 고객 수

```
grunt> xa_3 = FILTER xa BY userAge > 30;
grunt> xa_4 = group xa_3 by userAge ;
grunt> xa_5 = foreach xa_4 generate $0 as age , COUNT($1) ;
grunt> describe xa_5;
xa_5: {age: int,long}
```

(40,1338)
(50,326)
(60,54)
(70,10)
(80,1)

- Union 파일 합치기

```
grunt> student = UNION student1, student2;
xall = union xa , xb , xc , xd , xe , xf , xg , xh ;
```

- 상품 리스트 받아오기

```
grunt> productAll = foreach xall generate $0;  
grunt> productlist = distinct productAll;  
grunt> dump product  
productAll  productlist  
grunt> dump productlist;
```

(한 란 스 킨)
(비 자 시 카 밤)
(비 자 시 카 젤)
(아 토 수 딩 젤)
(한 란 젤 크 림)
(그 린 티 미 스 트)
(진 저 허 니 크 림)
(한 란 아 이 크 림)
(한 란 를 루 이 드)
(화 산 송 이 코 팩)
(더 미 니 멈 토 너)
(아 토 수 딩 크 림)
(퀵 톤업 마 스크)
(노 세 범 프 라 이 머)
(퍼 품 드 핸 드 크 림)
(그 린 티 씨 드 세 럼)
(그 린 티 씨 드 스 킨)
(그 린 티 씨 드 크 림)
(그 린 티 품 클 렌저)
(블 랙 그 린 티 세 럼)
(블 랙 그 린 티 크 림)
(비 자 시 카 마 스크)
(비 자 트 러 블 로 션)
(비 자 트 러 블 스 킨)
(한 란 인 텐 스 크 림)
(아 토 수 딩 5.5 로 션)
(그 린 티 3분 스 킨 팩)
(탠저린 비 타 C 세 럼)
(탠저린 비 타 C 스 킨)
(그 린 티 모 닝 클 렌저)
(그 린 티 벨 런 싱 로 션)
(그 린 티 벨 런 싱 스 킨)
(그 린 티 벨 런 싱 크 림)
(그 린 티 클 렌 징 오 일)
(그 린 티 클 렌 징 워 터)
(블 랙 그 린 티 마 스크)
(진 저 허 니 앰 플 스 킨)
(청 보 리 버 블 클 렌저)
(청 보 리 클 렌 징 크 림)
(한 란 슬 리 핑 마 스크)
(한 란 인 리 치 드 크 림)
(화 이 트 닝 포 어 스 킨)

- Sorting

```
grunt> Relation_name2 = ORDER Relatin_name1 BY (ASC|DESC);  
grunt> pdtlobname = ORDER productlist BY pdtName ASC;
```

(2018 에 코 손 수 건 그 린 티 별 런 싱 스 칸)
(그 린 티 3분 스 칸 팩)
(그 린 티 5분 찻 잎 마 스크 [소 용 량])
(그 린 티 모 닝 클 렌 저)
(그 린 티 미 스 트)
(그 린 티 미 스 트 [안 개 분 사])
(그 린 티 별 런 싱 로 선)
(그 린 티 별 런 싱 스 칸)
(그 린 티 별 런 싱 스 칸 7DAYS)
(그 린 티 별 런 싱 스 칸 케 어 2종 세 트)
(그 린 티 별 런 싱 스 칸 케 어 스페 셜 세 트)
(그 린 티 별 런 싱 크 림)
(그 린 티 슬 리 풍 마 스크)
(그 린 티 씨 드 듀 오 세 트)
(그 린 티 씨 드 딥 크 림)
(그 린 티 씨 드 세 릴)
(그 린 티 씨 드 스 칸)
(그 린 티 씨 드 아 이 크 림)
(그 린 티 씨 드 에 센 스 인 로 선)
(그 린 티 씨 드 크 림)
(그 린 티 클 렌 징 오 일)
(그 린 티 클 렌 징 워 터)
(그 린 티 클 렌 징 젤 투 품)
(그 린 티 품 클 렌 저)
(꽃 송 이 버섯 바 이 탈 로 선)
(꽃 송 이 버섯 바 이 탈 립 앤 아 이 크 림)
(꽃 송 이 버섯 바 이 탈 스 칸)
(꽃 송 이 버섯 바 이 탈 스 칸 케 어 2종 세 트)
(꽃 송 이 버섯 바 이 탈 크 림 기 획 세 트)
(꽃 송 이 버섯 바 이 탈 크 림 라 이 트 스페 셜 세 트)
(노 세 범 세 팅 스 프 레 이)
(노 세 범 프 라 이 머)
(더 미 니 멈 모 이 스 트 크 림)
(더 미 니 멈 앰 플 에 센 스)
(더 미 니 멈 클 렌 징 로 선)
(더 미 니 멈 토 너)
(더 미 니 멈 페 이 셜 클 렌 저)
(더 마 포 쿨 러 필 링 크 림 런 칭 세 트 [스 칸 베 리 어])

```
(더 미니멈 토너 )
(더 미니멈 페이셜 클렌저 )
(더 마포를러 필링 크림 런칭 세트 [스킨베리어])
(더 마포를러 필링 크림 런칭 세트 [토닝 세럼])
(더 마포를러 세범 블린젤 )
```

- Total count

```
grunt> one = foreach xall generate 1 as one;
grunt> grouping = group one all;
grunt> counting = foreach grouping generate SUM(one.one);
grunt> dump counting;
```

> **dump grouping;**

```
(all,{(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),
(1),(1),(1),(1),(1),(1),(1),(1),(1),(1),(1))
```

>**dump counting;**

```
(74500)
```

- Replace Date

```
grunt> REPLACE(string, 'RegExp', 'newChar');
```

- Get Time F

GetMinute(datetime)

GetMonth(datetime)

GetWeek(datetime)

```
GetSecond(datetime)
```

- 계절별 화장품 판매 수

```
grunt> xall = union xa , xb , xc , xd , xe , xf , xg , xh ; #스키마 합치기
grunt> month = foreach xall generate $0, $9 as date ; #제품명, 구매날짜 투플화
grunt> describe month;
grunt> dump month ;
grunt> pdt_date = foreach month generate $0 , GetMonth(date);
month: {pdtName: chararray,date: datetime}
```

```
(화 이 트 닝 포 어 스 킨 ,2015-03-31T00:00:00.000Z)
(화 이 트 닝 포 어 스 킨 ,2015-03-12T00:00:00.000Z)
(화 이 트 닝 포 어 스 킨 ,2015-03-11T00:00:00.000Z)
(퀵 톤 업 마 스 크 ,2019-01-16T00:00:00.000Z)
(퀵 톤 업 마 스 크 ,2019-01-15T00:00:00.000Z)
(퀵 톤 업 마 스 크 ,2019-01-13T00:00:00.000Z)
```

```
(꽃 송 이 버섯 바 이 탈 스 킨 ,1)
(꽃 송 이 버섯 바 이 탈 스 킨 ,1)
(꽃 송 이 버섯 바 이 탈 스 킨 ,1)
(꽃 송 이 버섯 바 이 탈 스 킨 ,1)
(꽃 송 이 버섯 바 이 탈 스 킨 ,1)
(꽃 송 이 버섯 바 이 탈 스 킨 ,1)
(꽃 송 이 버섯 바 이 탈 스 킨 ,1)
(꽃 송 이 버섯 바 이 탈 스 킨 ,12)
(꽃 송 이 버섯 바 이 탈 스 킨 ,12)
(꽃 송 이 버섯 바 이 탈 스 킨 ,12)
(꽃 송 이 버섯 바 이 탈 스 킨 ,12)
(꽃 송 이 버섯 바 이 탈 스 킨 ,12)
(꽃 송 이 버섯 바 이 탈 스 킨 ,12)
(꽃 송 이 버섯 바 이 탈 스 킨 ,11)
(꽃 송 이 버섯 바 이 탈 스 킨 ,11)
(꽃 송 이 버섯 바 이 탈 스 킨 ,11)
(꽃 송 이 버섯 바 이 탈 스 킨 ,11)
(꽃 송 이 버섯 바 이 탈 스 킨 ,11)
(꽃 송 이 버섯 바 이 탈 스 킨 ,11)
(꽃 송 이 버섯 바 이 탈 스 킨 ,10)
(꽃 송 이 버섯 바 이 탈 스 킨 ,10)
(꽃 송 이 버섯 바 이 탈 스 킨 ,10)
(꽃 송 이 버섯 바 이 탈 스 킨 ,9)
(꽃 송 이 버섯 바 이 탈 스 킨 ,9)
(꽃 송 이 버섯 바 이 탈 스 킨 ,9)
(꽃 송 이 버섯 바 이 탈 스 킨 ,9)
(꽃 송 이 버섯 바 이 탈 스 킨 ,9)
(꽃 송 이 버섯 바 이 탈 스 킨 ,9)
```

```
grunt> describe pdt_date;
grunt> xx = group pdt_date by ($0,$1);
grunt> describe xx;
xx: {group: (pdtName: chararray,org.apache.pig.builtin.getmonth_date_20: int),pdt_date: {(
  pdtName: chararray,org.apache.pig.builtin.getmonth_date_20: int)}}
```

```
grunt> dump xx ;
pdt_date: {pdtName: chararray,org.apache.pig.builtin.getmonth_date_7: int}
```

```
((더 마 포풀러 필링 크림 런칭 세트 [스킨베리어],12),{((더 마 포풀러 필링 크림 런칭 세트 [스킨베리어],12),(더 마 포풀러 필링 크림 런칭 세트 [스킨베리어],12)})  
((리프팅 사이언스 안티에이징 밴드 트라이얼 세트,1),{((리프팅 사이언스 안티에이징 밴드 트라이얼 세트,1),(리프팅 사이언스 안티에이징 밴드 트라이얼 세트,1)})
```

```
grunt> xxx = foreach xx generate $0 , COUNT($1);  
grunt> describe xxx;  
xxx: {group: (pdtName: chararray,org.apache.pig.builtin.getmonth_date_59: int),long}  
grunt> dump xxx
```

```
((한 란 스 킨 ,5),18)
((한 란 스 킨 ,6),28)
((한 란 스 킨 ,7),63)
((한 란 스 킨 ,8),61)
((한 란 스 킨 ,9),47)
((한 란 스 킨 ,10),53)
((한 란 스 킨 ,11),74)
((한 란 스 킨 ,12),106)
((비 자 시 카 밤 ,1),273)
((비 자 시 카 밤 ,2),275)
((비 자 시 카 밤 ,3),428)
((비 자 시 카 밤 ,4),162)
((비 자 시 카 밤 ,5),145)
((비 자 시 카 밤 ,6),219)
((비 자 시 카 밤 ,7),565)
((비 자 시 카 밤 ,8),350)
((비 자 시 카 밤 ,9),317)
((비 자 시 카 밤 ,10),382)
((비 자 시 카 밤 ,11),395)
((비 자 시 카 밤 ,12),458)
((비 자 시 카 젤 ,1),68)
((비 자 시 카 젤 ,2),6)
((비 자 시 카 젤 ,3),5)
((비 자 시 카 젤 ,4),26)
((비 자 시 카 젤 ,5),40)
((비 자 시 카 젤 ,6),50)
((비 자 시 카 젤 ,7),160)
((비 자 시 카 젤 ,8),84)
((비 자 시 카 젤 ,9),84)
((비 자 시 카 젤 ,10),79)
```

```
grunt> xxxx = foreach xxx generate group.pdtName as pdtname , group.$1 as month , $1 as pdtcount;
grunt> describe xxxx
xxxx: {pdtname: chararray,month: int,pdtcount: long}
grunt> STORE xxxx INTO '/user/hadoop/innisfreeOutput1' USING PigStorage (',');
```

포 레 스 트	포 맨	오 일	컨 트 를	을 인 원	에 센 스	,7,59
포 레 스 트	포 맨	오 일	컨 트 를	을 인 원	에 센 스	,8,29
포 레 스 트	포 맨	오 일	컨 트 를	을 인 원	에 센 스	,9,21
포 레 스 트	포 맨	오 일	컨 트 를	을 인 원	에 센 스	,10,24
포 레 스 트	포 맨	오 일	컨 트 를	을 인 원	에 센 스	,11,25
포 레 스 트	포 맨	오 일	컨 트 를	을 인 원	에 센 스	,12,38
꽃 송	이 버	섯	바 이 탈	크 림	라 이 트	스 페 설 세 트 ,1,2
꽃 송	이 버	섯	바 이 탈	크 림	라 이 트	스 페 설 세 트 ,6,1
꽃 송	이 버	섯	바 이 탈	크 림	라 이 트	스 페 설 세 트 ,7,4
꽃 송	이 버	섯	바 이 탈	크 림	라 이 트	스 페 설 세 트 ,10,5
꽃 송	이 버	섯	바 이 탈	크 림	라 이 트	스 페 설 세 트 ,11,5
꽃 송	이 버	섯	바 이 탈	크 림	라 이 트	스 페 설 세 트 ,12,5
더 마	포 르 러	필 링	크 림	런 칭	세 트 [토 닝 세 럼]	,1,17
더 마	포 르 러	필 링	크 림	런 칭	세 트 [토 닝 세 럼]	,2,3
더 마	포 르 러	필 링	크 림	런 칭	세 트 [토 닝 세 럼]	,3,2
더 마	포 르 러	필 링	크 림	런 칭	세 트 [토 닝 세 럼]	,4,2
더 마	포 르 러	필 링	크 림	런 칭	세 트 [토 닝 세 럼]	,5,1

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hadoop	supergroup	0 B	2019. 1. 10. 오전 10:39:57	1	128 MB	_SUCCESS
-rw-r--r--	hadoop	supergroup	67.61 KB	2019. 1. 10. 오전 10:39:57	1	128 MB	part-r-00000