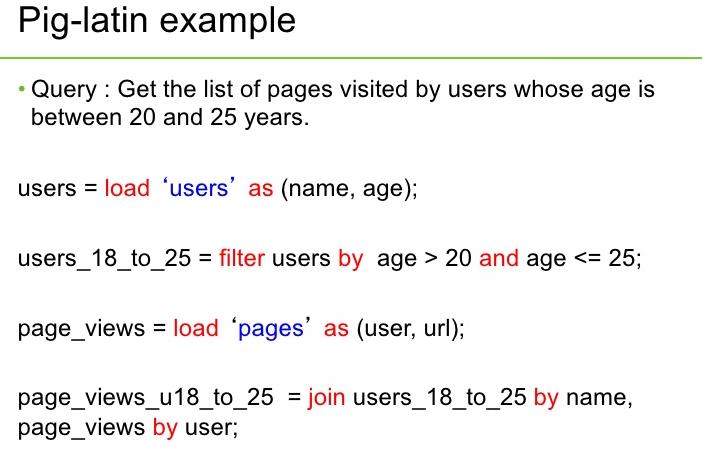
# Apache Pig Nedir ?

MapReduce ile büyük verileri dağıtık sistemlerde analiz edebiliriz. MapReduce programı yazmak için alttaki kodlama yöntemlerinden bir tanesini izlememiz gerekir

* Java MapReduce
* Apache Pig
* Apache Hive



Aslında Apache Pig ve Apache Hive yazmış olduğumuz kodları arka planda Java MapReduce kodlarına çevirir.Pig ve Hive kullanmamızın en önemli sebebi Java MapReduce ile kod geliştirmek zahmetlidir , bu yüzden Pig ve Hive ile daha hızlı MapReduce uygulamaları geliştirebiliriz

## Apache Pig Mimarisi

Geliştirmiş olduğumuz Pig kodları arka planda şu adımları izleyerek MapReduce kodlarına çevrilir

### Parse :

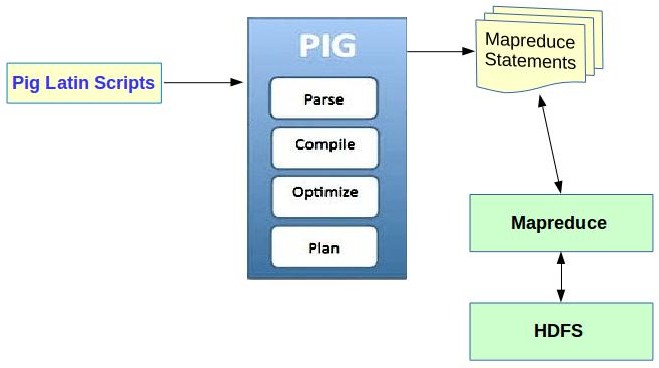
Bu adımda geliştirmiş olduğumuz Pig kodları syntax olarak kontrol edilir.Yanlış bir yazım varsa hata verilir

### Compile :

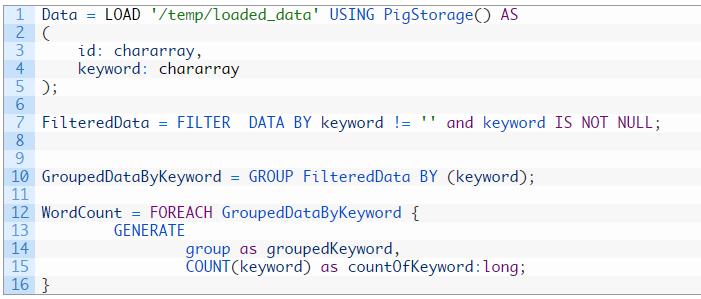
Bu adımda yazmış olduğumuz kodlar MapReduce job larına çevrilir

### Optimize ve Plan :

 Bu adımda yazmış olduğumuz kodları Apache Pig bizim için optimize eder. Çünkü geliştirmiş olduğumuz kodların daha performanslı çalışması için Pig çeşitli optimizasyonlar yapar



Alttaki şekilde Apache Pig ile yazılmış örnek bir kod bloğu bulunmaktadır. Burada örnek bir word-count uygulaması yapılmıştır. Benzer örneği Java MapReduce ile yazmak istersek yüzlerce satır kod geliştmemiz gerekir



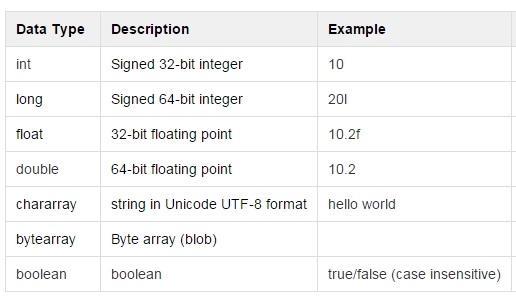
## Apache Pig’in avantajları

* Öğrenmesi ve geliştirmesi kolaydır
* Büyük verileri üzerinde kolaylıkla analizler yapılabilir
* Geliştirmiş olduğumuz MapReduce job’larını optimize eder
* Veri üzerinde analizler yapabileceğimiz metodlar sunar (filter, group, join, ordering vs..)
* İhtiyaç halinde java,javascript ve python ile kütüphaneler yazıp Apache pig içerisinde kullanabiliriz (udf)
* Hadoop içerisindeki ayarlamalar Apache Pig ile yapılabilir (mapreduce.reduce.memory.mb ..)

## Veri Tipleri

HDFS üzerinden yüklediğimiz verileri Apache Pig ile işlemek için öncelikle LOAD komutu ile verileri yüklememiz gerekir.

Apache Pig içerisinde tanımlanmış veri tipleri altta listelenmiştir



## Aritmetik Operatörler

Apache pig kütüphanesi içerisindeki aritmetik operatörler altta listelenmiştir

|  |  |  |
| --- | --- | --- |
| Operatör | Sembol | Örnek |
| toplama | + | 3 + 6 = 9 |
| çıkarma | – | 4 – 1 = 3 |
| çarpma | \* | 2 \* 6 = 12 |
| bölme | / | 9 / 2 = 4 |
| modülo | % | 9 % 2 = 1 |
| bincond | ?: | 4 == 3 ? ‘eşit’ : ‘eşit değil’ |

## Karşılaştırma operatörleri

|  |  |  |
| --- | --- | --- |
| Operatör | Sembol | Örnek |
| eşit | == | X = FILTER A BY (id == 8); |
| eşit değil | != | X = FILTER A BY (country != ‘US’); |
| küçüktür | < | X = FILTER A BY (durationTime < 15) |
| büyüktür | > | X = FILTER A BY (durationTime > 5) |
| küçük eşit | <= | X = FILTER A BY (durationTime <= 15) |
| büyük eşit | >= | X = FILTER A BY (durationTime >= 7) |
| regex | matches | X = FILTER A BY (url matches ‘.\*google.\*’) |

## Mantıksal Operatörler

|  |  |  |
| --- | --- | --- |
| Operatör | Sembol | Örnek |
| AND | and | X = FILTER A BY (id==8) AND (country == ‘US’); |
| OR | or | X = FILTER A BY (id==8) OR (country == ‘US’); |
| NOT | not | X = FILTER A BY (f1==8) OR (NOT (f2+f3 > f1)); |

## Null Operatörler

|  |  |  |
| --- | --- | --- |
| Operatör | Sembol | Örnek |
| is null | is null | X = FILTER A BY (url is null); |
| is not null | is not null | X = FILTER A BY country is not null |

## Verilerin Veritabanına Yazdırılması

Bu bölümde Apache Pig ile analiz ettiğimiz verileri, veritabanı veya hdfs’e nasıl yazdırıldığını inceleyeceğiz

Apache Pig kütüphanesinde bulunan STORE komutu ile verileri diğer sistemlere yazabiliriz

**Örnek log**

*1,John,US*

*2,Sarah,TR*

*3,John,US*

### Verilerin HDFS’e kaydedilmesi

*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*customerId:  int,*

*name:        chararray,*

*country:     chararray*

*);*

*RESULT = FOREACH CUSTOMER  {*

*GENERATE*

*name ,*

*country;*

*}*

*STORE RESULT INTO '/user/data/customeroutput' using PigStorage(',');*

### Verilerin Mysql’e kaydedilmesi

*%declare PARAM\_DB\_URL 'jdbc:mysql://192.168.1.1:3306/my\_db'*

*%declare PARAM\_DB\_USERNAME 'username1'*

*%declare PARAM\_DB\_PASSWORD 'password1'*

*register /user/lib/mysql-connector-java-5.1.21.jar*

*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*customerId:            int,*

*name:        chararray,*

*country:     chararray*

*);*

*RESULT = FOREACH CUSTOMER  {*

*GENERATE*

*name ,*

*country;*

*}*

*STORE RESULT INTO into 'my\_table' using org.apache.pig.piggybank.storage.DBStorage(*

*'com.mysql.jdbc.Driver','$PARAM\_DB\_URL',*

*'$PARAM\_DB\_USERNAME','$PARAM\_DB\_PASSWORD','insert into my\_table (name,country) values (?,?)');*

## Dosya Komutları

**Örnek log**

*1,John,US*

*2,Sarah,TR*

*3,John,US*

*4,David,EN*

*5,Peter,PL*

**Pig Kodu**

*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*customerId:  int,*

*name:        chararray,*

*country:     chararray*

*);*

*RESULT = FOREACH CUSTOMER  {*

*GENERATE*

*name ,*

*country;*

*}*

*fs -mkdir /user/data/temp*

*fs -cp /user/data/customer\* /user/data/temp*

**Örnek komutlar**

|  |  |  |
| --- | --- | --- |
| komut | Açıklama | Parametreler |
| fs -mv p\_from p\_to | Veriyi hdfs üzerinde taşımak için kullanılır | p\_from : kaynak dosya p\_to : hedef dosya |
| fs -rmr  p\_removed | Veriyi silmek için kullanılır | p\_removed : silinecek dosya |
| fs -copyFromLocal p\_from p\_to | Local makineden veri taşımak için | p\_from : local makinedeki dosya yolu p\_to : hdfs üzerindeki dosya yolu |
| fs -copyToLocal p\_from p\_to | hdfs üzerindeki dosyayı local makineye taşıyabiliriz | p\_from : hdfs dosya yolu p\_to : local dosya yolu |
| fs -touchz p\_file | boş dosya oluşturur | p\_file : dosyanın oluşturulacağı klasör yolu |

## Fonksiyonlar

**Örnek Veri**

*1,US,5,https://www.google.com.tr/#q=apache+pig,apache,pig*

*2,US,25,https://www.google.com.tr/#q=apache+hive,apache,hive*

*3,TR,20,https://www.google.com.tr/#q=apache+hadoop,apache,hadoop*

*4,EN,10,https://www.google.com.tr/#q=apache+oozie,apache,oozie*

*5,PL,16,https://www.google.com.tr/#q=apache%20flume,apache,flume*

*6,US,24,https://www.google.com.tr/#q=apache+spark,apache,spark*

*7,US,36,https://www.google.com.tr/#q=apache+kafka,apache,kafka*

*8,EN,48,https://www.google.com.tr/#q=apache+storm,apache storm in hadoop ,storm*

### AVG fonksiyonu

AVG fonksiyonu ile sayısal verilerin ortalamasını alabiliriz

*DATA = LOAD '/user/data/samplelogfile' USING PigStorage(',') AS*

*(*

*id:      int,*

*countryCode:   chararray,*

*durationTime:    int,*

*url:      chararray,*

*keyword1:    chararray,*

*keyword2:    chararray*

*);*

*GROUPED\_DATA = GROUP DATA BY countryCode;*

*RESULT = FOREACH GROUPED\_DATA  {*

*GENERATE*

*group ,*

*AVG(DATA.durationTime);*

*}*

*DUMP RESULT;*

Diğer fonksiyonlarında kullanımı bire bir aynı olduğu için sadece isimlerini ve ne işe yaradıklarını yazıyorum.

### COUNT fonksiyonu

COUNT fonksiyonu ile kayıt sayılarını bulabiliriz

### MAX,MIN fonksiyonları

Verilerin minimum ve maksimum değerlerini bulabiliriz

### SIZE fonksiyonu

SIZE fonksiyonu ile verilerin kaç karakter olduğunu bulabiliriz

### SUM fonksiyonu

SUM fonksiyonu ile sayısal verilerin toplamını bulabiliriz

### TOKENIZE fonksiyonu

TOKENIZE fonksiyonu verileri boşluklara göre ayırır

### **CONCAT fonksiyonu**

CONCAT fonksiyonu ile 2 alanı birleştirebiliriz

*DATA = LOAD '/user/data/samplelogfile' USING PigStorage(',') AS*

*(*

*id:      int,*

*countryCode:   chararray,*

*durationTime:    int,*

*url:      chararray,*

*keyword1:    chararray,*

*keyword2:    chararray*

*);*

*RESULT = FOREACH DATA  {*

*GENERATE*

*url ,*

*CONCAT(keyword1,keyword2) as combinedKeywords;*

*}*

*DUMP RESULT;*

## İlişkisel Operatörler

### **DISTINCT fonksiyonu**

Aynı bilgilere sahip kayıtları silmeye yarar

*DATA = LOAD '/user/data/sampledata3' USING PigStorage(',') AS*

*(*

*timestamp  :long,*

*computerID   :chararray,*

*countryCode  :chararray*

*);*

*DUMP DATA;*

*DISTINCT\_DATA = DISTINCT DATA;*

*DUMP DATA;*

### FILTER **fonksiyonu**

FILTER fonksiyonu ile filtreleme işlemleri yapabiliriz

*DATA = LOAD '/user/data/sampledata3' USING PigStorage(',') AS*

*(*

*timestamp  :long,*

*computerID   :chararray,*

*countryCode  :chararray*

*);*

*DUMP DATA;*

*FILTERED\_DATA = FILTER DATA BY countryCode == 'US' ;*

*DUMP FILTERED\_DATA ;*

### **GROUP fonksiyonu**

Apache Pig içerisinde bir alan üzerinde maksimum değer,ortalama değer gibi sonuçlar üretmek için bir alana göre gruplama yapmamız gereklidir

*DATA = LOAD '/user/data/sampledata4' USING PigStorage(',') AS*

*(*

*id:  int,*

*countryCode: chararray,*

*durationTime:  int,*

*url:  chararray*

*);*

*DATA\_GROUPED = GROUP DATA BY countryCode;*

*DUMP DATA\_GROUPED;*

### **FOREACH fonksiyonu**

Bu fonksiyon verileri üzerinde bir döngü ile dolaşmamızı sağlar

*DATA = LOAD '/user/data/sampledata3' USING PigStorage(',') AS*

*(*

*timestamp: long,*

*computerID: chararray,*

*countryCode: chararray*

*);*

*DATA\_GROUPED = GROUP DATA BY countryCode;*

*RESULT = FOREACH DATA\_GROUPED  {*

*GENERATE*

*group as countryCode,*

*COUNT(DATA);*

*}*

*DUMP RESULT;*

## Join Komutları

### JOIN

JOIN ile iki yada daha fazla log verisini belirli alanlara göre birleştirebiliriz

*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*name:            chararray,*

*age:       int,*

*dept\_id:         int*

*);*

*DEPARTMENT = LOAD '/user/data/departmentlog' USING PigStorage(',') AS*

*(*

*dept\_id:            int,*

*dept\_name:    chararray*

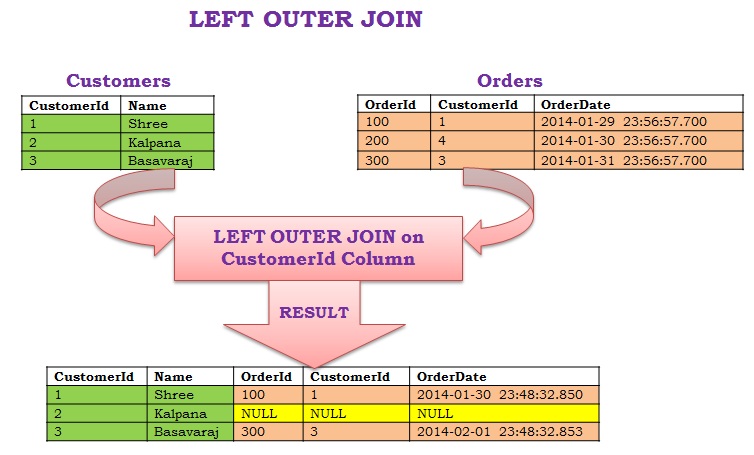
*);*

*JOINED\_DATA = JOIN CUSTOMER BY dept\_id , DEPARTMENT BY dept\_id;*

*DUMP JOINED\_DATA;*

### **LEFT JOIN**

Bu join türü ile soldaki tablodan tüm kayıtlar gelir. Sağdaki tablodan ise yalnızca eşleşen kayıtlar gelir



*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*customerId:            int,*

*name:      chararray*

*);*

*ORDERS = LOAD '/user/data/orderlog' USING PigStorage(',') AS*

*(*

*orderId:          int,*

*customerId:       int,*

*orderDate:     chararray*

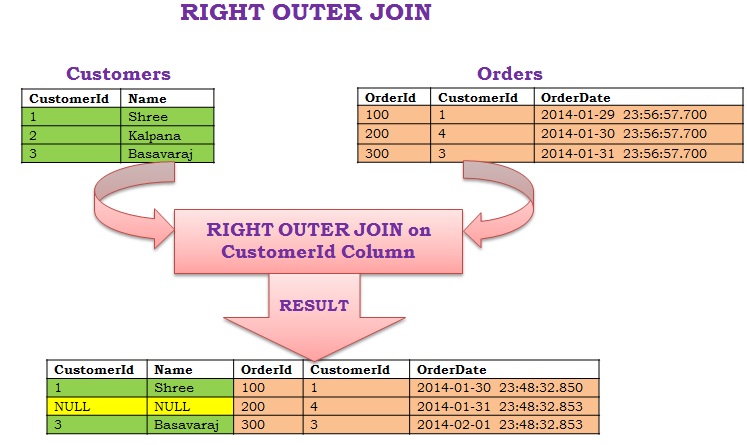
*);*

*JOINED\_DATA = JOIN CUSTOMER BY customerId LEFT OUTER , ORDERS BY customerId;*

*DUMP JOINED\_DATA;*

### **RIGHT JOIN**

Bu join türü ile sağdaki tablodan tüm kayıtlar gelir. Soldaki tablodan ise yalnızca eşleşen kayıtlar gelir



*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*customerId:            int,*

*name:      chararray*

*);*

*ORDERS = LOAD '/user/data/orderlog' USING PigStorage(',') AS*

*(*

*orderId:          int,*

*customerId:       int,*

*orderDate:     chararray*

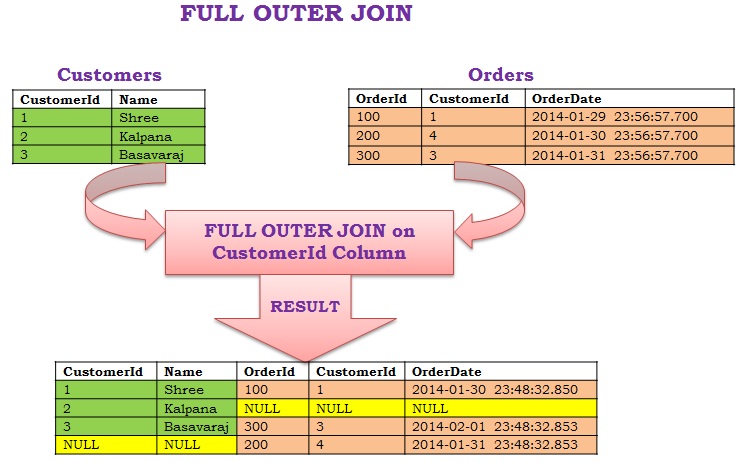
*);*

*JOINED\_DATA = JOIN CUSTOMER BY customerId RIGHT OUTER , ORDERS BY customerId;*

*DUMP JOINED\_DATA;*

### **FULL OUTER JOIN**

Bu join türü ile eşleşen kayıtlara ek olarak sağdaki ve soldaki tüm kayıtlar listelenir



*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*customerId:            int,*

*name:      chararray*

*);*

*ORDERS = LOAD '/user/data/orderlog' USING PigStorage(',') AS*

*(*

*orderId:          int,*

*customerId:       int,*

*orderDate:     chararray*

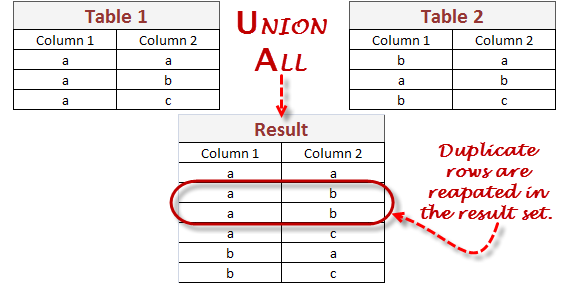
*);*

*JOINED\_DATA = JOIN CUSTOMER BY customerId FULL OUTER , ORDERS BY customerId;*

*DUMP JOINED\_DATA;*

### **UNION**

Union ile iki farklı log’u birleştirebiliriz



*CUSTOMER = LOAD '/user/data/customerlog' USING PigStorage(',') AS*

*(*

*customerId:            int,*

*name:      chararray*

*);*

*EMPLOYEE = LOAD '/user/data/employeelog' USING PigStorage(',') AS*

*(*

*employeeId:          int,*

*employeeName:          chararray*

*);*

*MERGED\_DATA = UNION CUSTOMER, EMPLOYEE;*

*DUMP MERGED\_DATA;*