Laboratório 2 – PIG

Objetivos do laboratório

Neste laboratório você ira praticar o conteúdo apresentados nesta lição, mais especificamente, você irá praticar os comandos apresentados na lição na linha de comando do PIG.

Instruções para o laboratório

Este laboratório foi criado apenas como um tutorial. Simplesmente execute o comando apresentado e observe o resultado.

Comandos básicos do sistema de arquivos do Hadoop

1. Como arquivo de entrada usaremos o arquivo **`/tmp/population.csv`**. Abaixo uma amostra do arquivo.

1850	0	1	1483789
1850	0	2	1450376
1850	5	1	1411067
1850	5	2	1359668
1850	10	1	1260099

   A primeira indica o ano do censo. A segunda coluna indica a idade do cidadão, a terceira o seu sexo e a última indica o número de pessoas. Por exemplo, a linha um indica que, em 1850, existiam 1.483.789 pessoas do sexo indicado com idade inferior a cinco anos.

2. Neste exemplo iremos criar os scripts PIG no diretório **`/tmp.`**

```
>  cd /tmp
>  vi seunome1.pig
```

3. Aperte a letra “i” (sem as aspas) para começar a editar o seu script. Inclua os seguintes comandos em seu script.

```
population = LOAD '/tmp/population.csv' USING PigStorage(',') AS
(year:int, age:int, gender:int, popsize:int);

DUMP population
```

Quando terminar de digitar aperte a tecla “Esc” (sem as aspas) e, em seguida digite “:wq” (sem as aspas).

Você pode executar o seu script localmente digitando:

```
> pig -x local seunome1.pig
```

Uma amostra da saída é listada abaixo:

```
(1850,0,1,1483789)
(1850,0,2,1450376)
(1850,5,1,1411067)
(1850,5,2,1359668)
(1850,10,1,1260099)
(1850,10,2,1216114)
(1850,15,1,1077133)
...........
```

Note que essa é apenas uma saída parcial. Neste caso só estamos listando o conteúdo do arquivo **/tmp/population.csv**.

4. Agora você irá agrupar cada linha retornada por ano. Você irá usar o operador GROUP BY.

```
>  vi seunome2.pig
```

Novamente aperte a letra “i” (sem as aspas) para começar a editar o seu script. Inclua os seguintes comandos em seu script.

```
population = LOAD '/tmp/population.csv' USING PigStorage(',') AS
(year:int, age:int, gender:int, popsize:int);

year = GROUP population BY year;

DUMP year;
```

Quando terminar de digitar aperte a tecla “Esc” (sem as aspas) e, em seguida digite “:wq” (sem as aspas).

Você pode executar o seu script localmente digitando:

```
> pig -x local seunome2.pig
```

Uma amostra da saída é listada abaixo:

```
(1850,{(1850,5,2,1359668),(1850,10,1,1260099),(1850,10,2,1216114),(1850
,15,1,1077133),(1850,15,2,1110619),(1850,20,1,1017281),(1850,20,2,10038
41),(1850,25,1,862547),(1850,25,2,799482),(1850,30,1,730638),(1850,30,2
,639636),(1850,35,1,588487),(1850,35,2,505012),(1850,40,1,475911),(1850
,40,2,428185),(1850,45,1,384211),(1850,45,2,341254),(1850,0,1,1483789),
(1850,0,2,1450376),(1850,5,1,1411067),(1850,50,1,321343),(1850,50,2,286
580),(1850,55,1,194080),(1850,55,2,187208),(1850,60,1,174976),(1850,60,
2,162236),(1850,65,1,106827),(1850,65,2,105534),(1850,70,1,73677),(1850
,70,2,71762),(1850,75,1,40834),(1850,75,2,40229),(1850,80,1,23449),(185
0,80,2,22949),(1850,85,1,8186),(1850,85,2,10511),(1850,90,1,5259),(1850
,90,2,6569)})
.........................
```

Note que essa é apenas uma saída parcial. Agora temos os campos agrupados por ano.

5. Agora iremos calcular a média da população por ano. Usaremos os operadores FOREACH e GENERATE para efetuar a transformação dos dados baseados nos dados das colunas.

   > vi seunome3.pig

   Novamente aperte a letra “i” (sem as aspas) para começar a editar o seu script. Inclua os seguintes comandos em seu script.

```
population = LOAD '/tmp/population.csv' USING PigStorage(',') AS
(year:int, age:int, gender:int, popsize:int);

year = GROUP population BY year;

results = FOREACH year GENERATE population.year,
AVG(population.popsize);

DUMP results;
```

   Quando terminar de digitar aperte a tecla “Esc” (sem as aspas) e, em seguida digite “:wq” (sem as aspas).

   Você pode executar o seu script localmente digitando:

   > pig -x local seunome3.pig

   Uma amostra da saída é listada abaixo:

```
({(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850)
,(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),
(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850),(
```

```
1850),(1850),(1850),(1850),(1850),(1850),(1850),(1850)},525988.39473684
21)
.....................................
```

Nesse caso podemos observar que a media populacional de 1850 é de 525.988,39473684 indivíduos. O número é apresentado como uma fração porque é uma media.

6. Agora veremos uma forma simples de gerarmos listarmos apenas duas colunas de nosso arquivo. Iremos exibir apenas as colunas YEAR e POPSIZE.

   > vi seunome4.pig

   Novamente aperte a letra “i” (sem as aspas) para começar a editar o seu script. Inclua os seguintes comandos em seu script.

   ```
   raw = LOAD '/tmp/population.csv' USING PigStorage(',') AS
   (year:int, age:int, gender: int, popsize:int);

   final = FOREACH raw GENERATE year, popsize;

   DUMP final;
   ```

   Quando terminar de digitar aperte a tecla “Esc” (sem as aspas) e, em seguida digite “:wq” (sem as aspas).

   Você pode executar o seu script localmente digitando:

   > pig -x local seunome4.pig

   Uma amostra da saída é listada abaixo:

   ```
   (1850,1483789)
   (1850,1450376)
   (1850,1411067)
   (1850,1359668)
   (1850,1260099)
   (1850,1216114)
   (1850,1077133)
   (1850,1110619)
   (1850,1017281)
   (1850,1003841)
   (1850,862547)
   (1850,799482)
   (1850,730638)
   (1850,639636)
   ```

```
(1850,588487)
.....................................
```

7. Agora iremos filtrar os dados dos anos anteriores a 1890.

   > vi seunome5.pig

   Novamente aperte a letra “i” (sem as aspas) para começar a editar o seu script. Inclua os seguintes comandos em seu script.

```
raw = LOAD '/tmp/population.csv' USING PigStorage(',') AS
(year:int, age:int, gender: int, popsize:int);

final = FILTER raw by year < 1890;

DUMP final;
```

   Quando terminar de digitar aperte a tecla “Esc” (sem as aspas) e, em seguida digite “:wq” (sem as aspas).

   Você pode executar o seu script localmente digitando:

```
> pig -x local seunome5.pig
```

   Outros operadores que podem ser usados são:

```
== - Igual
!= - Diferente
>  - Maior
<  - Menor
>= - Maior ou igual
<= - Menor ou igual
```

   Caso queira, pode classificar o resultado usando o comando

   finalorder = ORDER final BY popsize;

------ Fim deste laboratório ------