Przetwarzanie masywnych danych

Sprawozdanie

MSD – Transformacja danych z linii komend

Marcin Mrugas

21 kwietnia 2018

1 Proces transformacji

1. zmiana nazw plików i zamiana $< SEP >$ na ;

```
cat unique_tracks.txt | sed 's/<SEP>/;/g' > songs.csv
rm unique_tracks.txt
cat triplets_sample_20p.txt | sed 's/<SEP>/;/g' > listen.csv
rm unique_tracks.txt
```

2. Wydzielenie użytkowników do osobnej tabeli

```
cat listen.csv | cut -d';' -f1 | sort | uniq | awk '{print NR
    ";" $0}' > users.csv
```

3. Podmiana klucza użytkowników na klucz sztuczny

```
awk -F';' 'BEGIN{OFS=";"} NR==FNR{a[$2]=$1; next} {print a[$1],
     $2, $3}' users.csv listen.csv > listen_user_replaced.csv
rm listen.csv
mv listen_user_replaced.csv listen.csv
```

4. Dodanie klucza sztucznego do pilku songs

```
cat songs.csv | awk -F';' 'BEGIN{OFS=";"} {print NR, $0}' >
    songs_with_id.csv
rm songs.csv
mv songs_with_id.csv songs.cdv
```

5. Podmiana klucza piosenki na klucz sztuczny

```
awk -F';' 'BEGIN{OFS=";"} NR==FNR{a[$3]=$1; next} {print $1, a[
    $2], $3}' song.csv listen.csv > listen_song_replaced.csv
rm listen.csv
mv listen_song_replaced.csv listen.csv
```

6. Znalezienie mimimalnej i maksymalnej daty odtworzenia piosenki

```
cat listen.csv | cut -d";" -f3 | awk 'NR==1{max=$0; min=$0;} {
    if($0>max) max=$0; else if(min>$0) min=$0; } END{print
    strftime("%Y-%m",max),strftime("%Y-%m",min)}'
```

7. Stworzenie tabeli z datami

```
echo '2001;2010' | awk -F";" 'BEGIN{OFS=";"}END{id=1; for(y=$1;
    y<=$2;y++) for(m=1;m<=12;m++) printf("%d;%02d;%d\n", id++,m
    ,y);}' > date.csv
```

8. Podmiana timestampu na klucz obcy

```
awk -F';' 'BEGIN{OFS=";"} NR==FNR{a[$2 $3]=$1; next} {print $1,
     $2, a[strftime("%m%Y",$3)]}' date.csv listen.csv >
    listen_date_replaced.csv

rm listen.csv
mv listen_date_replaced.csv listen.csv
```

9. Wydzielenie artystów do osobnej tabeli

```
cat songs.csv | cut -d";" -f4 | sort | uniq | awk 'BEGIN{OFS
    =";"}{print NR, $0}' > artist.csv
```

10. Stworzenie tabeli łączącej artystów i piosenki

```
awk -F';' 'BEGIN{OFS=";"}NR==FNR{ a[$2]=$1; next} { print $1, a
    [$4] }' artist.csv songs.csv > song_artist.csv
```

11. Dodanie artystów do tabeli odsłuchów

```
 awk -F';' 'BEGIN{OFS=";"}NR==FNR{ a[$1]=$2; next } { print $0,
      a[$2] }' song_artist.csv listen.csv > listen_with_artist.
     csv
rm listen.csv
mv listen_wiht_artist.csv listen.csv
```

12. Usunięcie artystów z tabeli songs

```
cut -d';' -f1,2,3,5 songs.csv > songs_without_artist.csv
rm songs.csv
mv songs_without_artist.csv songs.csv
```

2 Zapytania i czas ich wykonania

1. Najpopularniejsze piosenki

```
cut -d';' -f2 listen.csv | sort | uniq -c | sort -nr | sed 's/^
     *//;s/ /;/' | awk -F';' 'BEGIN{OFS=";"}NR==FNR{ a[$1]=$4;
    next } {print $1,a[$2]}' songs.csv - | head -n 15
```

Czas zapytania 1m48.579s

2. Najbardziej aktywni użytkownicy - słuchacze którzy odtworzyli najwięcej unikalnych piosenek

```
cut -d';' -f1-2 listen.csv | sort -u | cut -d';' -f1 | sort |
    uniq -c | sort -nr | sed 's/^ *//;s/ /;/' | awk -F';' '
    BEGIN{OFS=";"}NR==FNR{ a[$1]=$1 OFS $2; next} {print $1,a[
    $2]}' users.csv - | head -n 15
```

Czas zapytania 3m11.550s

3. Najpopularniejsi artyści wg ilości odsłuchań

```
cut -d';' -f4 listen.csv | sort | uniq -c | sort -nr | sed -e '
    s/^ *//;s/ /;/' | awk -F';' 'BEGIN{OFS=";"} NR==FNR{ a[$1]=
    $2; next } {print $1, a[$2]}' artist.csv - | head -n 15 >
    results/popular_artists.csv
```

Czas zapytania 1m31.583s

4. Sumaryczna liczba odsłuchań w podziale na poszczególne miesiące

```
 cut -d';' -f3 listen.csv | awk -F';' 'BEGIN{OFS=";"}NR==FNR{a[
     $1]=$2;next} {print a[$1]}' date.csv - | sort | uniq -c
```

Czas wykonania polecenia 1m1.655s

5. Wszyscy użytkownicy, którzy odsłuchali wszystkie trzy najbardziej popularne piosenki zespołu Queeni

```
cat artist.csv | grep ';Queen$' | cut -d';' -f1 | awk '{print
    ";" $0 "$"}' | xargs -I{} grep {} listen.csv | cut -d';' -
    f2 | sort | uniq -c | sort -nr | head -n 3 | sed 's/^ *//'
    | cut -d' ' -f2 | awk -F';' 'BEGIN{OFS=";"}NR==FNR{a[$1]=1;
     next} a[$2]==1{ print $0 }' - listen.csv | cut -d';' -f1-2
     | sort -u | cut -d';' -f1 | sort | uniq -c | sed -e 's/^
    *//;s/ /;/' | awk -F';' '$1==3 {print $0}' | awk -F';' 'NR
    ==FNR{a[$2]=1; next}a[$1]==1{print $2 }' - users.csv | sort
```

Cza swykonania polecenia 44.657s