MapReduce w Sparku III

3 grudnia 2018

Opis pliku z zadaniami

Wszystkie zadania na zajęciach będą przekazywane w postaci plików `.pdf`, sformatowanych podobnie do tego dokumentu. Zadania będą różnego rodzaju. Za każdym razem będą one odpowiednio oznaczone:

- Zadania do wykonania na zajęciach oznaczone są symbolem $\triangle$ – nie są one punktowane, ale należy je wykonać w czasie zajęć.
- Punktowane zadania do wykonania na zajęciach oznaczone są symbolem $\diamond$ – należy je wykonać na zajęciach i zaprezentować prowadzącemu, w wypadku nie wykonania zadania w czasie zajęć lub nieobecności, zadania staje się zadaniem do wykonania w domu ($\star$).
- Zadania do wykonania w domu oznaczone są symbolem $\star$ – są one punktowane, należy je dostarczyć w sposób podany przez prowadzącego i w wyznaczonym terminie (zwykle przed kolejnymi zajęciami).
- Zadania programistyczne można wykonywać w dowolnym języku programowania, używając jedynie biblioteki standardowej dostępnej dla tego języka.

1 Zapytania do danych MSDC ⋆

Treść

Wykorzystując pliki `.csv`, przygotowane na zajęciach dotyczących przetwarzania danych MSDC w powłoce bash, zapisz w Sparku poniższe zapytania (są to te same zapytania, jak na wcześniejszych zajęciach dotyczących danych MSDC):

- Ranking popularności piosenek,
- Ranking użytkowników ze względu na największą liczbę odsłuchanych unikalnych utworów,
- Artysta z największą liczbą odsłuchań,
- Sumaryczna liczba odsłuchań w podziale na poszczególne miesiące,
- Wszyscy użytkownicy, którzy odsłuchali wszystkie trzy najbardziej popularne piosenki zespołu Queen.

Można wykorzystać następujące instrukcje: `spark.read.csv`, `groupBy`, `count`, `join`, `select`, `orderBy`,`show`, `agg`, `countDistinct`, `col`, `toDF`, `limit`, `filter`.

Dla ułatwienia wykonania zadania poniżej przedstawiony jest kod dla pierwszego zapytania:

```
//Read data
val songs = spark.read.
                  option("delimiter", ",").
                  csv("songs").
                  toDF("song_id", "track_long_id", "
                      song_long_id", "artist", "song")
val facts = spark.read.
                  option("delimiter", ",").
                  csv("facts").
                  toDF("id", "user_id", "song_id", "
                      date_id")

//The most popular songs
facts.groupBy("song_id").
      count.
      join(songs, facts("song_id")===songs("song_id")).
      select("song", "count").
      orderBy(desc("count")).
      show(10)
```

spark–msdc–1.scala

Za powyższe zadanie można otrzymać punkty bonusowe, po 1 punkcie za każde poprawnie wykonane zapytanie.