

1 Zadanie 1

1.1 Rozwiązanie

Rozwiązanie korzysta z mavena, aby ułatwić dodawanie zależności do projektu.

Zadanie składa się z dwóch części: Odczyt pliku JSON oraz zapis do pliku CSV. W obu przypadkach wykorzystałem biblioteki napisane w tym celu, Gson oraz OpenCSV. Aby ułatwić pracę i poprawić czytelność kodu, stworzyłem klasę Record, której obiekty przechowują odczytane pola. Rekordy z datą sprzed lipca 2017 filtruję między odczytem rekordu z pliku, a zapisem do listy, wykorzystując klasę Timestamp.

Sortowanie rekordów zapewnia interfejs Comparator, który pozwala na czytelne określenie pól, po których ma się odbyć sortowanie. Obie części zadania są od siebie niezależne. Rekordy przechowywanie są jako lista, która w klasie głównej App łączy oba komponenty w jedno rozwiązanie.

Rozwiązanie znajduje się w repozytorium, Java/recruitment

2 Zadanie 2

2.1 Rozwiązanie

Zaproponowane zapytanie SQL składa się z dwóch istotnych części. Pierwsze podzapytanie zlicza ilość wystąpień ID klienta w tabeli. Drugie podzapytanie zdobywa największą (czyli ostatnią) datę dla danego ID klienta. Gdy będzie kilka takich samych dat, wybierana jest ta, która została dodana później (ma większe kontakt_id).

Łącząc oba podzapytania w warunki, SELECT wyciąga najnowsze statusy klientów, z którymi były przynajmniej trzy próby kontaktu.

Rozwiązanie znajduje się w pliku repozytorium SQL/one-two.sql pod numerem 1

3 Zadanie 3

3.1 Rozwiązanie (bez bonusów)

INSERT INTO wkłada do tabeli faktów f_docieralnosc wszystkie elementy z zapytania SELECT. Aby wyciągnąć potrzebne informacje, stworzona została tabela posiadająca: status, datę oraz ilość wystąpień statusu tego dnia. Mając te pola, pozostało tylko wyciągnąć je z tabeli do konkretnych pól. Podzapytanie CASE pozwala łatwo wydobywać potrzebne informacje, przyrównując kolejne wartości kolumny status.

Rozwiązanie znajduje się w pliku repozytorium SQL/one-two.sql pod numerem pod numerem 2

4 Zadanie 4

4.1 Rozwiązanie

Na rozwiązanie składają się cztery komponenty: *tFileInputJSON*, *tFilterRow*, *tSortRow*, *tFileOutput-Delimited*. I/O komponentów są połączone w tej samej kolejności. Odczyt zadanego pliku JSON odbywa się wyłącznie po polach *\$.records[*]*, gdzie pola każdego rekordu mapowane są do kolejnych kolumn tabeli. *tFilterRow* pozbywa się rekordów z datą sprzed 1 lipca 2017, *tSortRow* sortuje rosnąco po polach *klient_id* oraz *kontakt_ts*, ostatni komponent zapisuje plik w odpowiednim formacie.

Rozwiązanie zostało wyeksportowane i spakowane do Talend project/packed.zip