

**Wstęp**

Jednym z głównych elementów naszej pracy jest przetwarzanie surowych danych do postaci, która przynosi wartość biznesową i jest czytelna dla odbiorców. Integrujemy się z wieloma systemami zewnętrznymi jak i aplikacjami wewnętrznymi, dlatego nie ograniczamy się do jednego formatu plików.

**Opis danych**

Plik “*statuses.json*” zawiera sztucznie wygenerowane informacje na temat obsługi klientów.

*kontakt\_id* - unikalny identyfikator zdarzenia

*klient\_id* - identyfikator klienta

*pracownik\_id* - identyfikator pracownika, który nadał status danemu klientowi

*status* - status jaki został nadany dla naszego klienta

*kontakt\_ts* - timestamp kontaktu z klientem w formacie

**Zadania**

1. Za pomocą języka Java, wyklucz rekordy z datą sprzed lipca 2017, posortuj dane w dostarczonym pliku po kolumnach “klient\_id”, “kontakt\_ts”, a następnie zapisz wynik w formacie .**csv** o strukturze:

*kontakt\_id | klient\_id | pracownik\_id | status | kontakt\_ts*

1. Zaproponuj zapytanie SQL zwracające **ostatni** status każdego klienta, z którym były co najmniej 3 próby kontaktu. Potraktuj podesłany plik jako tabelę.
2. Zaproponuj zapytanie SQL, które zwróci dane do faktu *f\_docieralnosc* pokazującego globalną docieralność do klientów. Jego struktura wygląda następująco: *data* | *sukcesy* | *utraty* | *do\_ponowienia* | *zainteresowani\_utraty* | *niezainteresowani\_sukcesy*

Fakt zawiera 1 wymiar:

* *data* (nie timestamp!)

oraz miary:

* *sukcesy -* liczba klientów, których ostatnim statusem jest “zainteresowany”
* *utraty* - liczba klientów, których ostatnim statusem jest “niezainteresowany”
* *do\_ponowienia* - liczba klientów, których ostatni status to “poczta\_głosowa” lub “nie\_ma\_w\_domu”
* ***(Bonus)*** *zainteresowani\_utraty* - liczba klientów, których ostatnim statusem jest “niezainteresowany”, a poprzednio wystąpił status “zainteresowany”
* ***(Bonus)*** *niezainteresowani\_sukcesy* - liczba klientów, których ostatnim statusem jest “zainteresowany”, a poprzednio wystąpił status “niezainteresowany”

1. **(Bonus)** Rozwiąż zadanie 1. w Talend Open Studio for Data Integration bez pisania kodu w czystej javie.

**Podsumowanie**

Do każdego z powyższych przygotuj krótki komentarz/notatkę w której wyjaśnisz nam swoje podejście do problemu. Zwróć uwagę na czytelność swojego kodu - ułatwi nam to weryfikację. W razie wątpliwości - pisz śmiało.

Przygotowane przez Ciebie rozwiązania prześlij nam w postaci archiwum **.zip** lub jeśli posiadasz konto na **github/bitbucket**- prześlij w odpowiedzi link do repozytorium.

Jeśli nie poradzisz sobie ze wszystkimi zadaniami, nie rezygnuj i prześlij nam stan swojej pracy. Pozwoli nam to dopasować stopień stanowiska (Junior/Regular/Senior) do Twojej wiedzy i umiejętności.

POWODZENIA !