1. Problem

Pracujesz dla Firmy, która dała ci zadanie wyznaczania najbardziej cenionych klientów ze zbiorów danych pochodzących z trzech źródeł - trzech różnych systemów. Systemy te dzielą klientów, tzn. istnieją klienci, którzy występują w dwóch lub nawet trzech systemach – w takich wypadkach klient posiada ten sam adres, imię i nazwisko. Czasem ten sam klient może występować więcej niż raz w tym samym systemie.

1.1. Najbardziej ceniony klient, to taki który:

A. ma dochód powyżej wartości INCOME_THRESHOLD (przyjąć wartość 200000),
B. lub sumaryczna suma pieniędzy wszystkich transakcji dla danego klienta jest większa niż TRANSACTION_THRESHOLD (przyjąć wartość 500).

1.2. Zwroty produktów powinny być traktowane jak transakcje o ujemnej wartości transakcji, tj. jeżeli klient ma dwie transakcje na sumę 100 i 200 oraz zwrot w wysokości 50, to sumaryczna wartość transakcji dla takiego klienta jest obliczana następująco: 100 + 200 – 50 = 250.

1.3. Nie bierzemy pod uwagę klientów, których dochód jest większy niż parametr VIP_INCOME (użyć wartości 10900), ale możemy odrzucić w ten sposób nie więcej niż 200 rekordów.

2. Formaty zbiorów danych

Dane zostały przekazane w następującej formie:

PLIKI A:

1. Plik a-customers.txt – plik używa kodowania ISO-8859-2

Nazwa pola	Type pola	Długość pola	Opis
custid	Number	8	Identyfikator
fname	String	20	Imię
lname	String	25	Nazwisko
street_address	String	90	Ulica
district	String	30	Powiat
voivodship	String	20	Województwo
postcode	Number	5	Kod pocztowy (format 99999)
preferred	Number	1	1 - klient VIP, 0 nie klient VIP
newline	String	1	pole techniczne

2. Plik a-transactions.txt – pola są w kodowaniu ISO-8859-1

Nazwa pola	Type pola	Długość pola	Opis
Transid	number	9	identyfikator transakcji
transtype	string	3	PUR - zakup, RET - zwrot
transdate	date	6	data transakcji
custid	number	8	id klienta robiącego transakcję, ten sam co pole custid w pliku a-customers.
prodid	string	8	identyfikator produktu
quantity	number	3	ilość produktów

Price	number	7	cena pojedynczego produktu
discount	number	3	zniżka w ułamku, tj. .10 oznacza 10% zniżki
returnid	number	9	identyfikator zwrotu
reason	String	30	powód zwrotu
newline	String	1	pole techniczne

* Sumaryczna transakcja dla pojedynczego rekordu z tej tabeli to quantity * price * (1-discount)

PLIKI B:

1. Plik b-customers.dat

Jest plik pipe-delimited z kodowaniem Windows-1250

Nazwa pola	**Type pola**	**Długość pola**	**Opis**
custid	number	N/A	Identyfikator klienta
firstname	string	N/A	Imię
lastname	string	N/A	Nazwisko
street_address	String	N/A	Ulica
district	String	N/A	Powiat
voivodship	String	N/A	Województwo
postcode	Number	N/A	Kod pocztowy (format 99-999)

2. Plik b-transactions.dat

Jest to plik CSV z kodowaniem ISO-8859-1.

Nazwa pola	**Type pola**	**Długość pola**	**Opis**
transid	number	N/A	Identyfikator transakcji
prodid	string	N/A	Kod zakupionego produktu
price	number	N/A	Cena jednostkowa produktu
quantity	number	N/A	Ilość produktów
transdate	data (DD.MM.YYYY)	N/A	Data transakcji
custid	number	N/A	identyfikator klienta, ten sam co w polu custid pliku b-customers

* Sumaryczna transakcja dla pojedynczego rekordu z tej tabeli to quantity * price.

PLIK C:

Plik customer-info.dat

Plik pochodzi z serwera mainframe i ma kodowanie EBCDIC (strona kodowa IBM037). Pola te po konwersji z EBCDIC proszę zinterpretować jako ISO-8859-2 - już bez konwersji. Wynika to z faktu, że w większości systemów mainframe nie ma natywnych polskich znaków.

Nazwa pola	**Typ pola**	**Długość pola**	**Opis**
Id	number	9	Identyfikator
firstname	string	42	Imię

lastname	string	32	Nazwisko
Street_address	string	110	Ulica
District	string	40	Powiat
Voivodship	string	50	Województwo
postcode	number	5	Kod pocztowy
est_income	number	8	Dochód –zmniejszony ma być o połowę gdy own_or_rent = 'R'
own_or_rent	string	1	O – posiada dom/mieszkanie, R – wynajmuje dom/mieszkanie
Date	"MM/DD/YYYY"	10	Data
newline	string	1	Pole techniczne

3. Wyjście

Struktura wyjściowa powinna być w formie pliku pipe-delimited z kodowaniem UTF-8:

Nazwa pola	**Type pola**	**Długość pola**	**Opis**
Id	number	N/A	first_defined (custid z (a), custid z (b), id (c))
Source	string	N/A	A - dla rekordu z pliku a-customers.txt, B- dla b-customers.dat, C gdy rekord jest z customer-info.
Fname	string	N/A	Imię
Lname	string	N/A	Nazwisko
Street_address	string	110	Ulica
District	string	40	Powiat
Voivodship	string	50	Województwo
postcode	number	5	Kod pocztowy
Preferred	string	N/A	1 = klient VIP, 0 = nie VIP, 2 = nieokreślony
est_income	Numer	N/A	dochód, pusty jak nieznany
own_or_rent	String	N/A	O - posiada dom lub hipotekę, R - wynajmuje, U – nieokreślony
Purchases	Numer	N/A	Sumaryczna wartość transakcji – puste gdy nie określona.
Newline	String	N/A	pole techniczne zawsze wartość '\n'

4. Warunki rozwiązania zadania

Zadanie będzie punktowane od 0 do **30 punktów**.

Rozwiązanie może być zrealizowane przez **2 lub 3-osobowe grupy**. W rozwiązaniu dopuszczalne jest używanie Pythona, R, SQL, lub javy – całość ma działać na platformie Linux. Rozwiązanie powinno być sparametryzowane dla INCOME_THRESHOLD, TRANSACTION_THRESHOLD i VIP_INCOME.

Proszę o zanotowanie, kto był odpowiedzialny z grupy za każdą część zadania oraz ile każdy etap rozwiązania zajął: np. 4 godziny na zbudowanie modułu do przeczytania plików, 2 godziny na wyznaczenie połączeń, 6 godzin na testowanie. Jaka część zajęła najdłużej?

Pytania (odpowiedzi proszę zawrzeć w rozwiązaniu):

1. Jakiego typu połączenia zbiorów musimy wykonać: inner-join, outer-join?
2. Czy każda transakcja posiada znanego klienta? Jak to sprawdzić?
3. Czy każdy znany klient z plików A i B posiada co najmniej jedną transakcję?
4. Czy każda transakcja jest związana z istniejącym klientem?
5. Czy występują problemy z jakością danych w plikach źródłowych? Jeżeli tak, to jakie? Jak to wpływa na połączenia zbiorów?
6. W jaki sposób przeczytać dane z serwerów mainframe?
7. Który rekord (imię i nazwisko) ma największy dochód, a który największą wartość transakcji?
8. Ile rekordów spełnia kryteria z punktu 1?
9. Ile rekordów odrzucimy w punkcie (1.2)?
10. Czy zmieni się kod jeżeli wolumen danych wzrośnie 100 krotnie (dla każdego pliku)?

Rozwiązaniem zadania jest także **ZWIĘZŁA dokumentacja techniczna** o następującej zawartości:

- opis koncepcji rozwiązania problemu;
- architektura oprogramowania rozwiązującego problem, ze wskazaniem konkretnych technologii programistycznych (np. język programowania, wykorzystane biblioteki);
- diagram przepływów zadań (ang. data processing workflow / ELT workflow / data processing pipeline).

Wskazówki:

1. W pierwszej kolejności należy skoncentrować się na poprawnym przeczytaniu plików.
2. Po odczytaniu plików polskie znaki powinny być prawidłowo interpretowane.
3. Po odczytaniu plików, obliczyć sumaryczne transakcje dla każdego z klientów dla plików A i B lub wyznaczyć dochód dla pliku C.
4. W ostatniej fazie wyznaczyć najbardziej cenionych klientów.