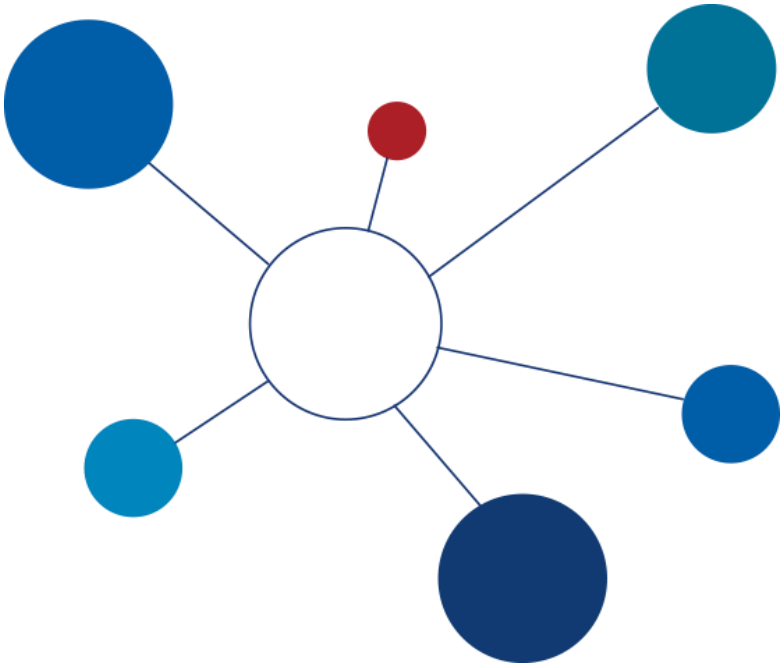


Technologie Zasilania i Odświeżania Hurtowni Danych

laboratorium

część III

v20170324

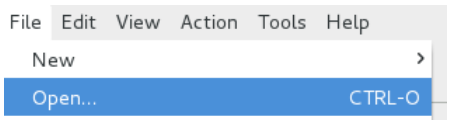
I. Dodanie drugiego źródła danych

W poprzednim zestawie ćwiczeń wykonaliśmy transformację pracowników z jednego źródła danych jakim była baza `shop1`. W modelowanym systemie mamy jednak również drugą bazę danych – `shop2`. Pomimo, że pracownicy są zatrudnieni w konkretnym sklepie, ich dane są co jakiś czas replikowane do drugiej bazy danych (pracowników z pierwszego sklepu do bazy danych `shop2` a pracowników ze sklepu drugiego do bazy danych `shop1`). Mamy zatem dwa źródła, w których mogą występować takie same lub podobne (ze względu na zachodzące zmiany) zbiory danych.

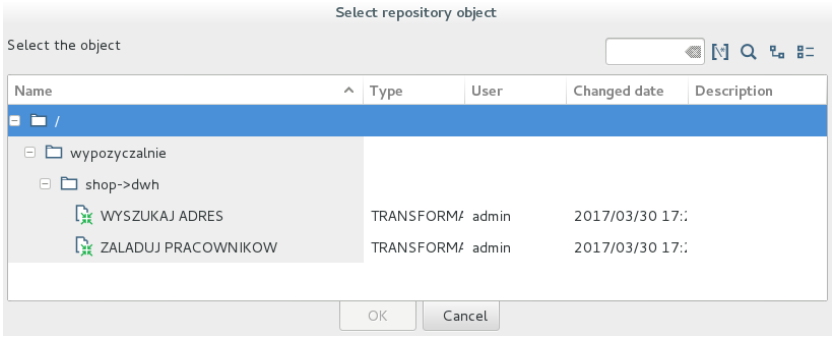
W przyjętym rozwiązaniu problemu integracji tych danych założyliśmy, że dla konkretnego pracownika decydujące znaczenie będzie miała baza danych sklepu, w którym jest on zatrudniony (bez względu na datę modyfikacji). Ponieważ również adresy pracowników są replikowane między systemami, problem integracji dwóch źródeł będzie także obejmował transformację dla wyszukiwania adresów.

1. Dodanie drugiego źródła danych transformacji `ZALADUJ PRACOWNIKOW`.

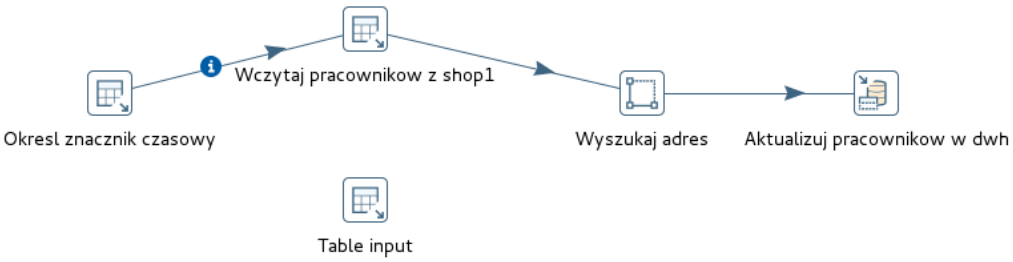
   a) Jeżeli nie masz otwartego diagramu transformacji `ZALADUJ PRACOWNIKOW` z głównego menu wybierz opcję *File->Open*.



   b) Rozwiń katalog `wypozyczalnie/shop->dwh` i wskaż element `ZALADUJ PRACOWNIKOW`. Zatwierdź wybór przyciskiem *OK*.

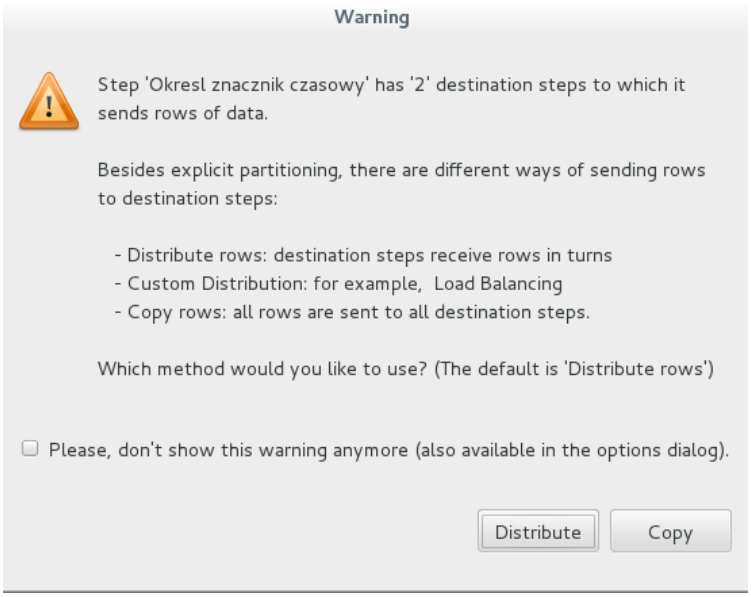


   c) W obecnej chwili mamy na diagramie tylko jeden węzeł odczytujący dane z relacji pracownicy ze sklepu `shop1`. Dodaj kolejny komponent *Table Input* (katalog *Input*).

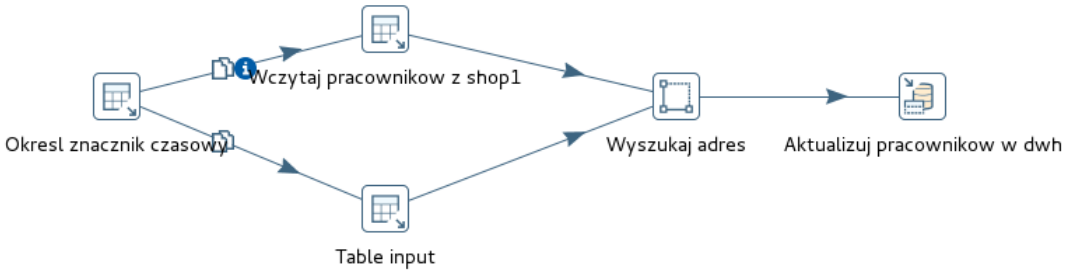


   d) Nowo dodany komponent, podobnie jak komponent `Wczytaj pracownikow z shop1`, będzie także wykorzystywał wartość wyliczoną przez komponent `Okresl znacznik czasowy`. W związku z tym, wynik pochodzący z komponentu `Okresl znacznik czasowy` musi zostać przekazany w postaci dwóch kopii. Wykonaj połączenie w następujący sposób:

      - Wejściem do nowo dodanego komponentu ma być wynik działania komponentu `Okresl znacznik czasowy`. Podczas wykonywania połączenia zostaniesz zapytany o sposób przekazywania wyniku. Wybierz opcję *Copy*.

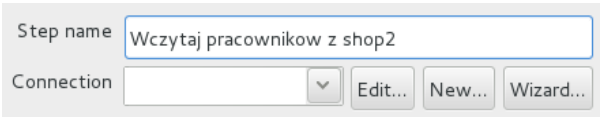


- Wynik działania nowo dodanego komponentu ma być przekazywany do komponentu `Wyszukaj adres`.



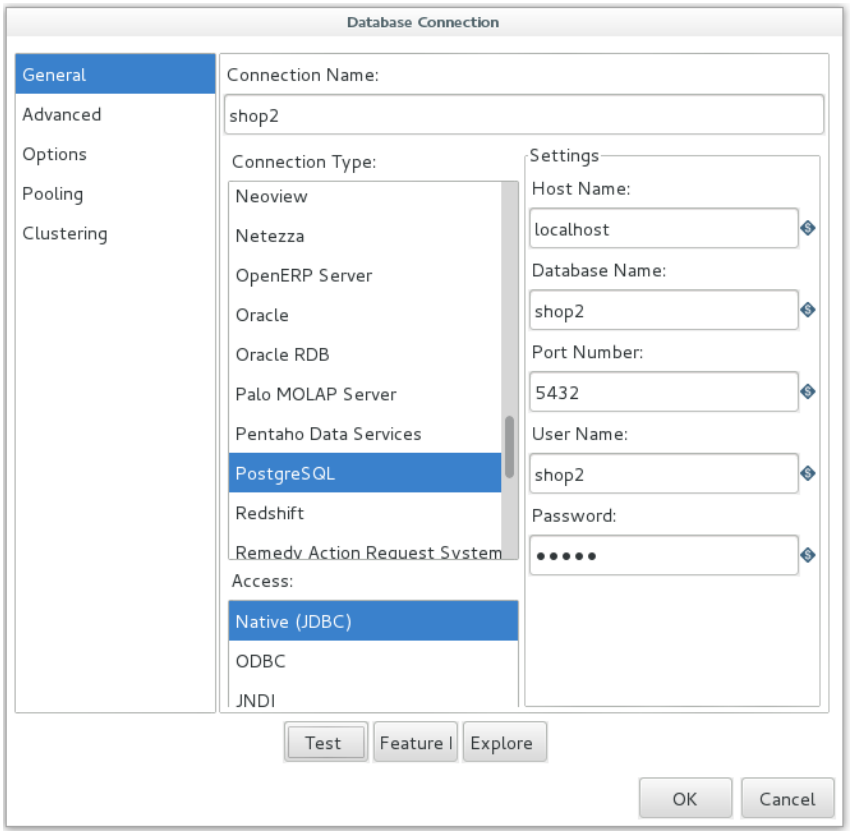
e) Przejdź do edycji nowo dodanego komponentu.

- Ustaw nazwę na wartość `Wczytaj pracownikow z shop2`.
- Ponieważ nie ma jeszcze skonfigurowanego połączenia do bazy danych `shop2`, dodaj je wykorzystując przycisk *New*.

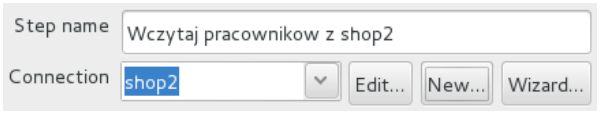


f) Ponieważ drugim źródłem danych jest baza danych zarządzana przez *PostgreSQL*, wybierz jako *Connection Type* pozycję *PostgreSQL* i wprowadź następujące dane:

- *Connection Name*: `shop2`
- *Host*: `localhost`
- *Database Name*: `shop2`
- *Port*: `5432`
- *User Name*: `shop2`
- *Password*: `shop2`



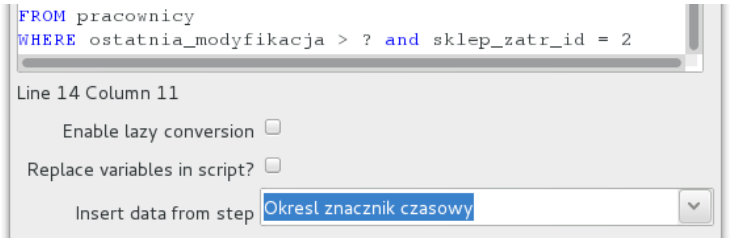
g) Przetestuj połączenie korzystając z przycisku *Test*. Zatwierdź dane korzystając z przycisku *OK*. Po powrocie do edycji kroku *Table input*, połączenie `shop2` będzie już automatycznie wybrane.



- Wygeneruj zapytanie SQL przy pomocy przycisku *Get SQL select statement…* wskazując relację `pracownicy`.
- Dodaj następujący warunek selekcji do polecenia SQL:

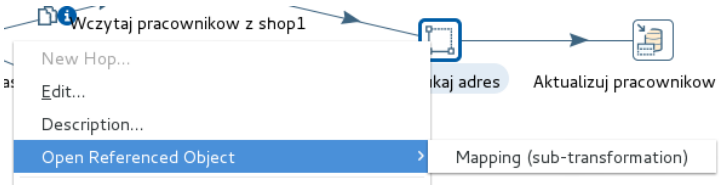
```
WHERE ostatnia_modyfikacja > ? and sklep_zatr_id = 2
```

- W polu *Insert data from step* wybierz z listy element `Okresl znacznik czasowy`.



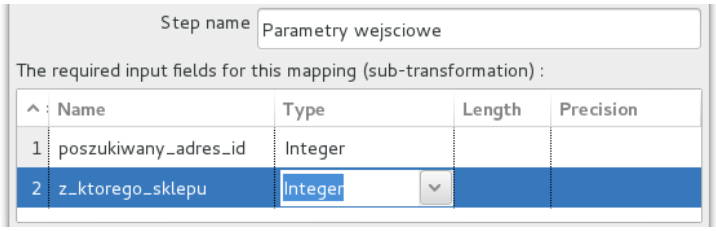
- Zatwierdź zmiany przyciskiem *OK*.

h) W obecnej sytuacji dane o pracownikach będą odczytywane z baz danych sklepów, w których są oni zatrudnieni. Podobne działanie należy wykonać podczas wyszukiwania adresów. Pracownicy ze sklepu `shop1` powinni mieć ustalane adresy na podstawie zapisów w bazie danych tego sklepu i analogicznie adresy pracowników drugiego sklepu powinny pochodzić z bazy danych `shop2`. Przejdźmy zatem do edycji podtransformacji odpowiedzialnej za wyszukiwanie adresów. Z menu kontekstowego dla komponentu `Wyszukaj adres` wybierz opcję *Open Referenced Object -> Mapping (sub-transformation)*.



i) Przejdź do edycji komponentu `Parametry wejsciowe`.

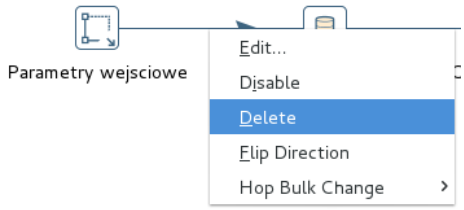
- Dodaj nowy parametr o nazwie `z_ktorego_sklepu`.
- Ustaw typ parametru na `Integer`.



- Zatwierdź zmiany przyciskiem *OK*.

j) Kolejnym elementem przetwarzania, który dodamy do transformacji jest komponent *Switch/Case*. Będzie on odpowiadał za wybór jednego z dwóch kierunków przepływu danych w zależności od wartości parametru `z_ktorego_sklepu`. Ponieważ nowy komponent musi zostać dodany do istniejącego łańcucha przetwarzania konieczne jest wprowadzenie drobnych modyfikacji.

- Usuń połączenie pomiędzy komponentami `Parametry wejsciowe` i `Wczytaj adres z shop1`. Możesz to wykonać wybierając polecenie *Delete* z menu kontekstowego strzałki reprezentującej przepływ.



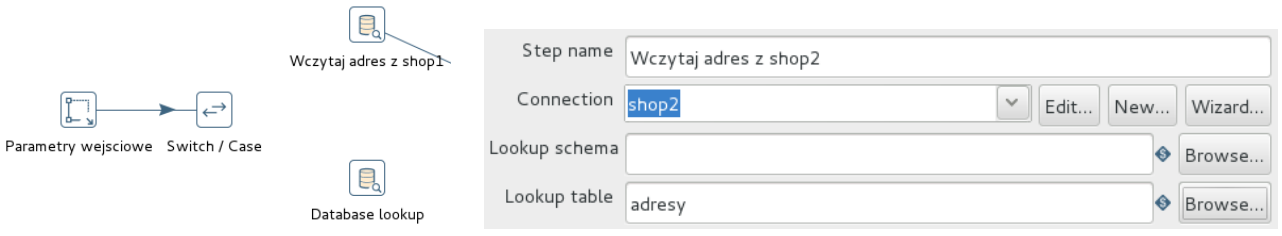
- Dodaj komponent *Switch/Case* z katalogu *Flow*.

k) Zdefiniuj połączenie od komponentu `Parametry wejsciowe` do nowo dodanego komponentu.

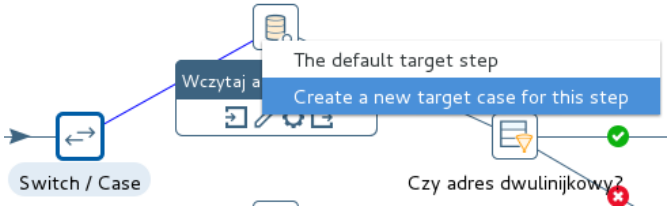
- Do całości przetwarzania brakuje nam jeszcze jednego komponentu – takiego, który będzie wyszukiwał adresy w bazie danych `shop2`. Umieść na diagramie transformacji komponent *Database Lookup* z katalogu *Lookup*.
- Przejdź do jego właściwości i ustaw nazwę na `Wczytaj adres z shop2`.
- Wskaż połączenie `shop2`, ustaw parametr *Lookup table* na `adresy`.

UWAGA! Jeżeli na liście wyboru połączeń nie ma pozycji shop2, zapisz obie transformacje, tj. WYSZUKAJ ADRES i ZALADUJ PRACOWNIKOW. Zamknij je i otwórz ponownie.

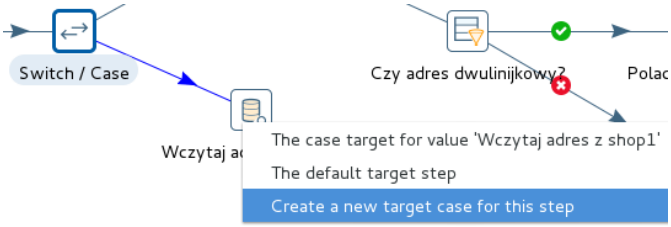
- Zatwierdź przyciskiem *OK*.



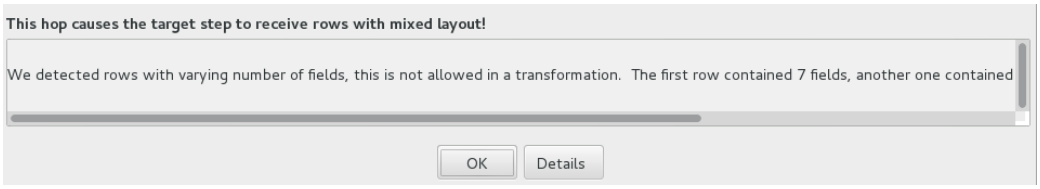
- Wykonaj połączenie od komponentu *Switch/Case* do komponentu `Wczytaj adres z shop1`. Podczas definiowania tego połączenia zostaniesz zapytany o warunek przekazania danych. Wybierz opcję *Create a new target case for this step*.



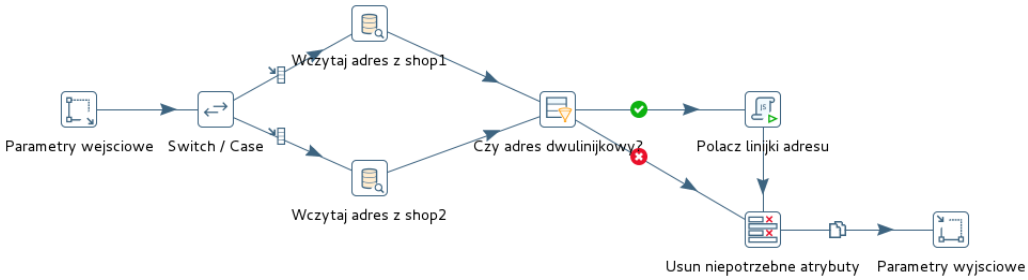
- Wykonaj analogiczne działanie dla połączenia komponentu *Switch/Case* z komponentem `Wczytaj adres z shop2`.



- Połącz komponent `Wczytaj adres z shop2` z komponentem `Czy adres dwulinijkowy?`. Wskaż opcję *Main output of step* przy definiowaniu połączenia. Na ekranie pojawi się ostrzeżenie dotyczące niezgodności danych wejściowych dla elementu `Czy adres dwulinijkowy?`. Wymagane jest aby dane z obu źródłowych komponentów miały taki sam format (liczbę i typy atrybutów). Zaakceptuj informację wybierając przycisk *OK*.

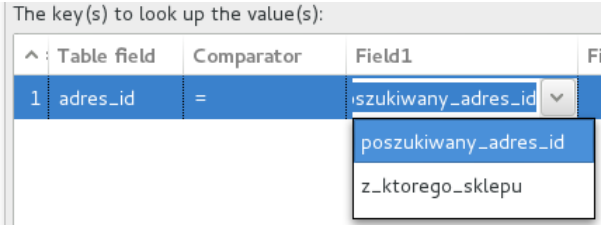


- Diagram transformacji powinien wyglądać następująco:

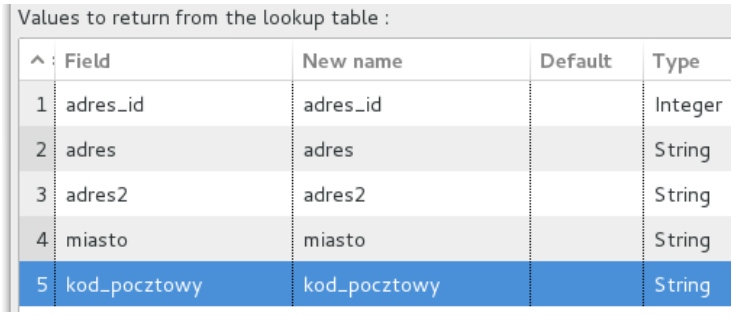


l) Przyczyną wystąpienia ostrzeżenia jest niedokończona definicja komponentu `Wczytaj adres z shop2`. Przejdź do edycji tego komponentu.

- W sekcji definiującej klucz jako *Table field* wskaż `adres_id`.
- Użyj metody porównania (*Comparator*) opartej na relacji równości.
- W kolumnie *Field1* wybierz parametr `poszukiwany_adres_id`.



- Jako wartości pobierane z bazy danych wybierz wszystkie elementy (przycisk *Get lookup fields*).
- W tabeli *Values to return from the lookup table* usuń wiersze z atrybutami `telefon` oraz `ostatnia_modyfikacja`.
- W razie konieczności przesuń pozycje tabeli tak, aby były zgodnie z poniższym rysunkiem.

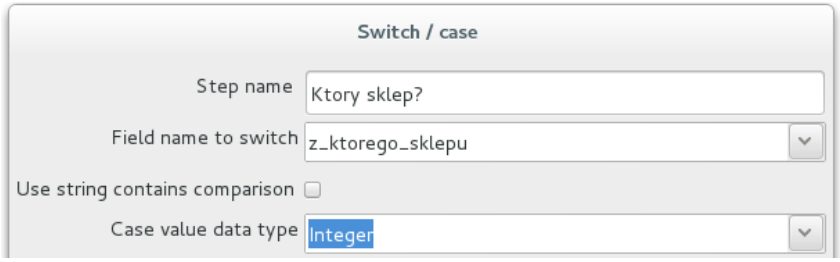

Values to return from the lookup table :

	Field	New name	Default	Type
1	adres_id	adres_id		Integer
2	adres	adres		String
3	adres2	adres2		String
4	miasto	miasto		String
5	kod_pocztowy	kod_pocztowy		String

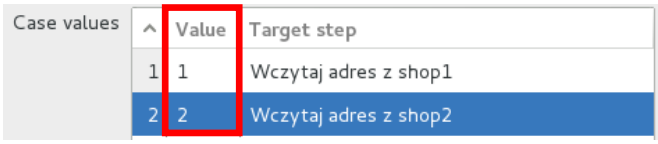
- Zatwierdź zmiany przyciskiem *OK*.

m) Do tej pory nie skonfigurowaliśmy komponentu odpowiedzialnego za podział przepływu. Przejdź do właściwości komponentu *Switch/Case*.

- Ustaw nazwę na wartość `Ktory sklep?`.
- Wskaż atrybut `z_ktorego_sklepu` jako ten, którego wartość będzie decydowała o kierunku przepływu danych (*Field name to switch*).
- Ustaw typ danych dla tego pola (*Case value data type*) na `Integer`.



- W tabeli dla parametru `Case Values` definiujemy konkretne wartości decydujące o kierunku przepływu. Dla wartości (kolumna *Value*) równej `1` powinien to być komponent (kolumna *Target step*) `Wczytaj adres z shop1`, natomiast dla wartości `2` komponent `Wczytaj adres z shop2`. **Zwróć uwagę na zawartość kolumny *Value***.


Case values

	Value	Target step
1	1	Wczytaj adres z shop1
2	2	Wczytaj adres z shop2

- Zatwierdź zmiany przyciskiem *OK*.

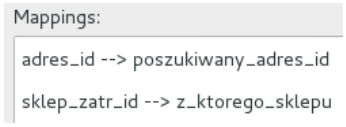
n) Wybierz opcję *File->Save* aby zapisać zmiany w transformacji podrzędnej `WYSZUKAJ ADRES`.

o) Ostatnią czynnością, którą musimy wykonać jest przekazanie numeru sklepu do transformacji podrzędnej `WYSZUKAJ ADRES`.

- Wróć do diagramu transformacji `ZALADUJ PRACOWNIKOW`.
- Przejdź do edycji komponentu `Wyszukaj adres`.

- W zakładce *Input* wybierz przycisk *Mapping…*.
- Wprowadź powiązanie dla par `adres_id` i `poszukiwany_adres_id` oraz

`sklep_zatr_id` i `z_ktorego_sklepu`. Wykorzystaj przycisk *Add* do dodawania kolejnych mapowań.



- Zaakceptuj zmiany dwukrotnie wybierając przycisk *OK*.

p) Zapisz zmiany w transformacji pracowników (*File->Save*).

q) Uruchom transformację tym razem obejmującą oba źródła danych.

	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active
1	Okresl znacznik czasowy	0	0	2	1	0	0	0	0	Finished
2	Wczytaj pracownikow z shop1	0	1	0	0	0	0	0	0	Finished
3	Wczytaj pracownikow z shop2	0	1	1	1	0	0	0	0	Finished
4	Wyszukaj adres	0	1	1	0	0	0	0	0	Finished
5	Aktualizuj pracownikow w dwh	0	1	1	1	1	0	0	0	Finished

II. Transformacja i integracja informacji o klientach na podstawie dwóch źródeł danych

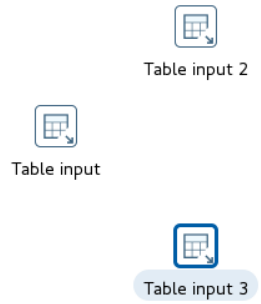
Dane o klientach przechowywane są w obu źródłowych bazach danych. Replikacja zapewnia, że ich numery są spójne. Klienci mogą jednak zmieniać swoje dane w dowolnym ze sklepów. Każda taka zmiana prowadzi do aktualizacji znacznika czasowego wskazującego moment ostatniej modyfikacji danych określonego klienta. Podczas integracji zasilania hurtowni, jeżeli dane klienta będą obecne w obu źródłach, do hurtowni trafią te bardziej aktualne tzn. z większym znacznikiem czasowym.

W odróżnieniu od transformacji pracowników tym razem uwzględnimy oba źródła danych już od początku tworzenia transformacji. Dodatkowo wykorzystamy wcześniej zdefiniowaną transformację podrzędną dla wyszukiwania adresu.

1. Na początku odczytamy klientów z dwóch źródeł danych.

a) Utwórz nową transformację wybierając z głównego menu *File->New->Transformation*.

b) Umieść na diagramie trzy komponenty *Table Input* z katalogu *Input*. Jeden z tych komponentów będzie wyznaczał znacznik czasowy, dzięki któremu będzie możliwe pobranie tylko takich klientów ze źródłowych baz danych, którzy ulegli modyfikacji od poprzedniego zasilenia hurtowni danych.



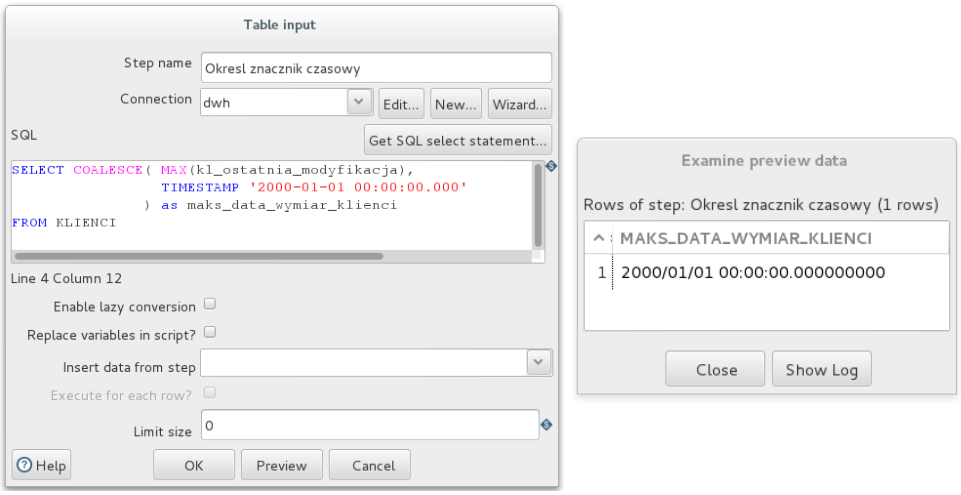
c) Przejdź do edycji komponentu *Table Input* (dowolnego z trójki obecnej na diagramie)

- Ustaw nazwę komponentu na wartość `Okresl znacznik czasowy`.
- Wskaż połączenie do bazy danych `dwh`.
- Wprowadź następujące polecenie SQL:

```
SELECT COALESCE( MAX(kl_ostatnia_modyfikacja) ,
```

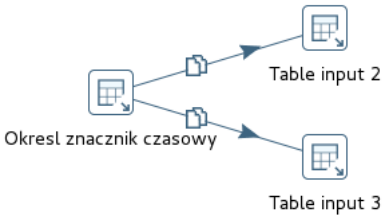
```
                    TIMESTAMP '2000-01-01 00:00:00.000'
                ) as  maks_data_wymiar_klienci
        FROM KLIENCI
```

- Sprawdź działanie komponentu używając przycisku *Preview*. Podobnie jak w przypadku transformacji dla pracowników, w przypadku braku danych w hurtowni danych zwracany jest znacznik przypadający na początek roku 2000.



- Zatwierdź zmiany wybierając przycisk *OK*.

d) Połącz edytowany przed chwilą komponent z pozostałą dwójką w ten sposób, że komponent `Okresl znacznik czasowy` będzie elementem dostarczającym dane do tych komponentów. Przy wykonywaniu drugiego połączenia wybierz metodę *Copy* w celu wysłania takiego samego zbioru danych w obu kierunkach.

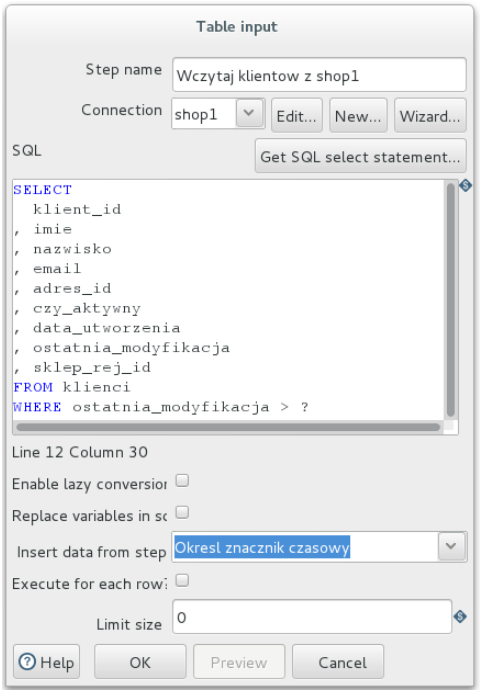


e) Przejdź do definiowania jednego z komponentów *Table Input*, który będzie odpowiedzialny za odczyt danych z bazy `shop1`.

- Ustaw nazwę komponentu na `Wczytaj klientow z shop1`.
- Wskaż połączenie do bazy danych `shop1`.
- Wygeneruj polecenie SQL używając przycisku *Get SQL select statement*. Wskaż relację `klienci` i zgódź się na zawarcie nazw atrybutów w klauzuli SELECT.
- Dodaj do polecenia SQL następujący warunek:

```
WHERE ostatnia_modyfikacja > ?
```

- Jako źródło danych dla parametru zapytania (*Insert data from step*) ustaw `Okresl znacznik czasowy`.



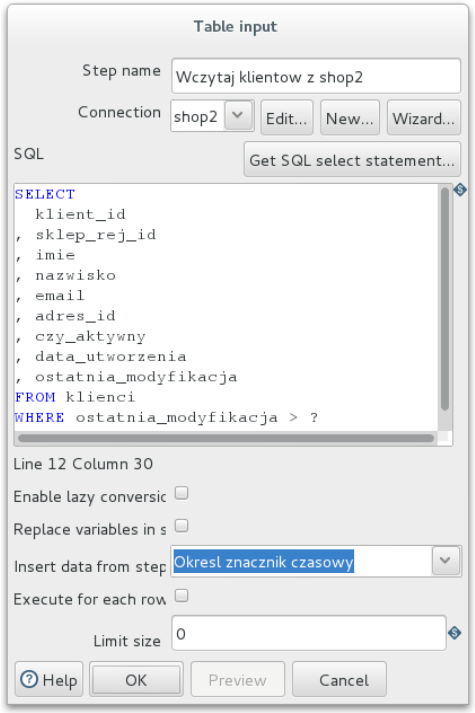
- Zatwierdź zmiany przyciskiem *OK*.

f) Wykonaj poprzedni punkt dla ostatniego z trójki elementów *Table Input* przy uwzględnieniu następujących zmian:

- Użyj nazwy `Wczytaj klientow z shop2`.
- Wskaż połączenie o nazwie `shop2`.
- Dodaj do polecenia SQL odczytującego dane o klientach następujący warunek:

```
WHERE ostatnia_modyfikacja > ?
```

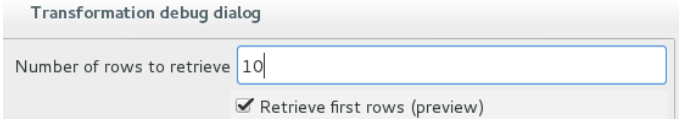
- Pamiętaj o dodaniu informacji o źródle wartości w/w parametru.



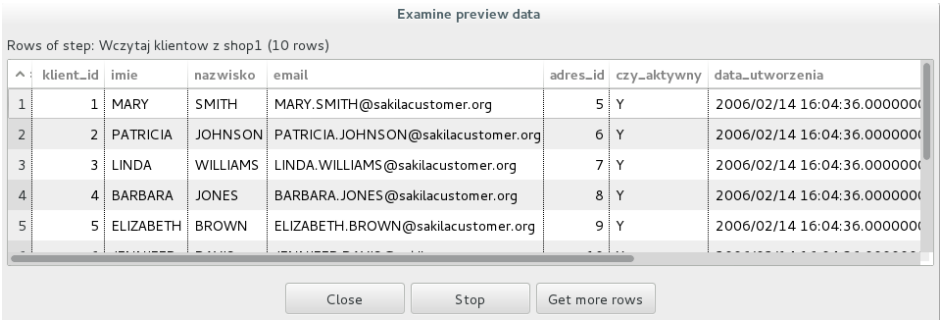
g) Przetestuj działanie transformacji. Przed możliwością uruchomienia transformacji konieczne jest jej zapisanie w repozytorium.

- Wybierz z głównego menu *File-Save As…*.

- Użyj nazwy `ZALADUJ KLIENTOW`.
- Jako miejsce zapisu wskaż katalog `wypozyczalnie/shop->dwh`.
- Kliknij lewym przyciskiem na komponent `Wczytaj klientow z shop1`. Następnie korzystając z prawego przycisku wywołaj menu kontekstowe i wybierz polecenie *Preview*.
- Wprowadź liczbę `10` określającą ile maksymalnie może zostać zaprezentowanych krotek danych przetwarzanych przez wybrany komponent .



- Wybierz przycisk *Quick launch*. Wynik powinien być podobny do następującego:



- Zamknij okno z wynikami używając przycisku *Stop*.

Uwaga: jeśli zamknąłeś okno używając przycisku *Close* prawdopodobnie proces transformacji nadal będzie uruchomiony. Możesz go zatrzymać używając ikony kwadratu umieszczonej tuż nad diagramem transformacji.

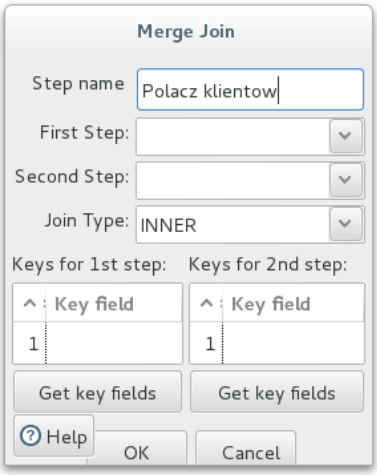
2. Integracja danych o klientach

Po wykonaniu poprzedniego ćwiczenia dysponujemy krotkami klientów z obu źródeł danych. Ze względu na ograniczoną częstotliwość aktualizacji, część klientów może występować tylko w jednym źródle. Może również dojść do sytuacji, w której ten sam klient posiada różne dane w obu źródłach. W kolejnych punktach zdefiniujemy kroki transformacji pozwalające uspójnić dane o klientach przez wprowadzeniem ich do hurtowni danych.

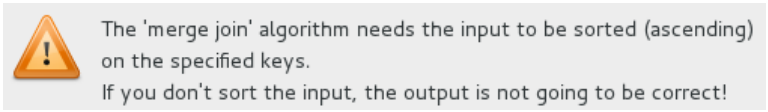
a) Z katalogu *Joins* wybierz i umieść na diagramie komponent *Merge Join*. Działanie tego komponentu przypomina operację złączenia relacyjnego w relacyjnych bazach danych.

b) Przejdź do określenia właściwości tego komponentu. Jak łatwo zauważyć do działania tego komponentu wymagane jest zdefiniowanie dokładnie dwóch źródeł dostarczających dane oraz podanie warunków i typu połączenia.

- Ustaw nazwę na `Polacz klientow`.



- Zatwierdź zmiany używając przycisku *OK*. Zostanie wyświetlony komunikat o konieczności dostarczenia do edytowanego komponentu danych posortowanych zgodnie z kluczem, który zostanie wykorzystany do porównywania danych.



- Potwierdź przyjęcie do wiadomości komunikatu poprzez wybranie przycisku *I understand*.

c) Zastanówmy się jeszcze chwilę nad wynikiem połączenia. Wejście stanowią dwa zbiory krotek z danymi klientów, wyjściem jest jeden zbiór krotek, w którym każdy rekord zawiera dane pochodzące z obu źródeł. Ponieważ nazwy atrybutów są takie same w źródłowych bazach danych podczas kolejnych kroków transformacji może dojść do konfliktu nazw, który środowisko *Pentaho DI* rozwiązuje poprzez automatyczne dopisywanie sufiksu do konfliktowych nazw. Takie nazwy nie zawsze są czytelne i mogą prowadzić do nieprawidłowych ich interpretacji. Wrócimy zatem do zapytań SQL służących do ekstrakcji danych i wprowadźmy własne, unikalne nazwy. Wykorzystamy możliwość definiowania aliasów na poziomie SQL i dodatkowo uporządkujemy klauzulę SELECT usuwając niepotrzebne atrybuty. Zauważmy również, że żaden z atrybutów krotki nie wskazuje źródła jej pochodzenia. Żeby zapewnić możliwość określenia, z której bazy danych odczytano wybraną krotkę dodajmy także stałą wartość odpowiednio 1 dla `shop1` i 2 dla `shop2` dostępną pod aliasami `ktory_sklep_shop1` oraz `ktory_sklep_shop2`.

- Przejdź do edycji kroku `Wczytaj klientow z shop1`.
- Zmodyfikuj klauzulę SELECT tak aby wyglądała w następujący sposób:

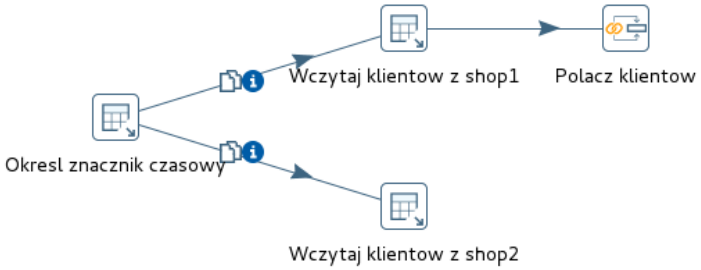
```
SELECT
  klient_id as klient_id_shop1
, imie as imie_shop1
, nazwisko as nazwisko_shop1
, email as email_shop1
, adres_id as adres_id_shop1
, ostatnia_modyfikacja as ostatnia_modyfikacja_shop1
, 1 as ktory_sklep_shop1
```

- Zatwierdź zmiany przyciskiem *OK*.
- Wykonaj analogiczne działanie dla transformacji `Wczytaj klientow z shop2`. Tym razem klauzula SELECT powinna wyglądać następująco:

```
SELECT
  klient_id as klient_id_shop2
, imie as imie_shop2
```

```
, nazwisko as nazwisko_shop2
, email as email_shop2
, adres_id as adres_id_shop2
, ostatnia_modyfikacja as ostatnia_modyfikacja_shop2
, 2 as ktory_sklep_shop2
```

d) Zdefiniuj połączenie od komponentu `Wczytaj klientów z shop1` do nowo dodanego komponentu.



e) Zadbajmy o spełnienie warunku komponentu `Polacz klientow` dotyczącego posortowanej listy krotek na wejściu. Zauważ, że warunek ten możemy spełnić na etapie ekstrakcji danych z bazy danych. Przejdź do edycji komponentu `Wczytaj klientow z shop1`.

- Na końcu polecenia SQL dopisz następującą klauzulę:

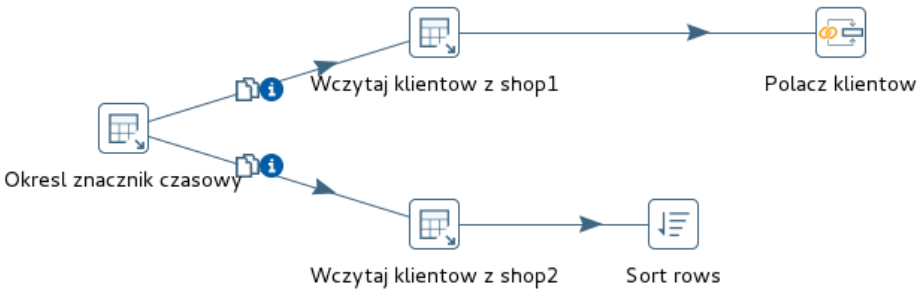
```
ORDER BY klient_id
```

```
, ostatnia_modyfikacja as ostatnia_modyfikacja_shop1
, 1 as ktory_sklep_shop1
FROM klienci
WHERE ostatnia_modyfikacja > ?
ORDER BY klient_id
```

- Zatwierdź zmianę przyciskiem *OK*.

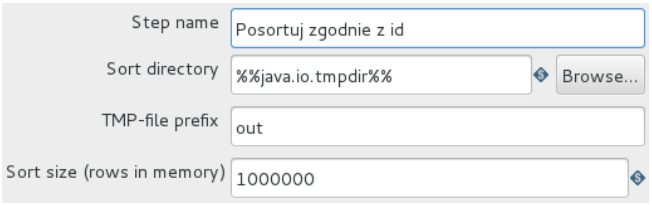
f) Podobne działanie moglibyśmy przeprowadzić dla komponentu `Wczytaj klientow z shop2`, jednak nie zawsze jest to pożądane. Przykładem może być chęć uniknięcia generowania dodatkowego obciążenia w źródłowej bazie danych. Dla wielu innych źródeł danych, na przykład plikowych, nie mamy możliwości sortowania bezpośrednio w komponencie ekstrakcji danych. W takich sytuacjach możemy wykorzystać stworzony specjalnie do tego celu komponent.

- Dodaj do diagramu transformacji komponent *Sort rows* z katalogu *Transform*.
- Zdefiniuj połączenie od komponentu `Wczytaj klientow z shop2` do nowo dodanego komponentu.

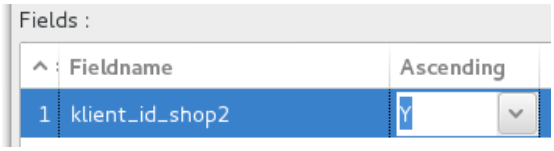


g) Przejdź do edycji komponentu *Sort rows*.

- Nadaj mu nazwę `Posortuj zgodnie z id`.
- Parametry odpowiedzialne za określenie rozmiaru pamięci przeznaczonej na sortowanie oraz lokalizacji plików sortowania pozostaw bez zmian.

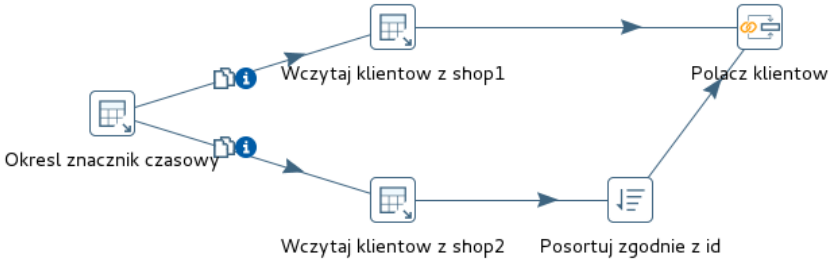


- W sekcji *Fields* określ porządek sortowania. Krotki powinny zostać posortowanie zgodnie z rosnącym identyfikatorem klienta. Wybierz atrybut `klient_id_shop2` dla kolumny *Fieldname*. W kolumnie *Ascending* wybierz opcję `Y`.

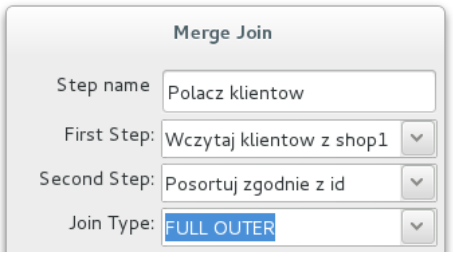


- Zatwierdź zmiany przyciskiem *OK*.

h) Wprowadź połączenie pomiędzy komponentami `Posortuj zgodnie z id` oraz `Polacz klientow`. Przejdź do edycji własności komponentu `Polacz klientow`.



- Jako *First Step* wskaż na komponent `Wczytaj klientow z shop1`.
- Jako *Second Step* wskaż na komponent `Posortuj zgodnie z id`.
- Typ połączenia (*Join Type*) to FULL OUTER, co oznacza, że wszystkie krotki z obu relacji pochodzących z dwóch baz danych zostaną zachowane w wyniku połączenia.



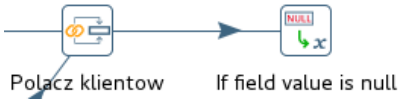
- Ostatnim elementem definicji połączenia jest określenie warunku połączeniowego . W odróżnieniu od poprzednio definiowanych elementów w tym przypadku nie mamy możliwości posłużenia się wyborem z listy. Konieczne jest pobranie wszystkich atrybutów i usunięcie niepotrzebnych lub ręczne wpisanie nazw. Skorzystajmy z tej drugiej opcji.
- W sekcji *Keys for 1st step* pozostaw tylko klucz `klient_id_shop1`. W sekcji *Keys for 2nd step* pozostaw klucz `klient_id_shop2`.

- Zatwierdź zmiany przyciskiem *OK* (potwierdź również zrozumienie komunikatu dotyczącego sortowania).

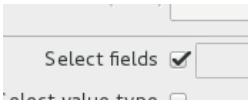
i) W wyniku połączenia dla klientów, którzy występowali tylko w jednej ze źródłowych baz danych data modyfikacji pochodząca z drugiej bazy danych posiada wartość nieokreśloną (NULL). Ponieważ takie

wartości nie mogą być w żaden sposób porównywane należy je zamienić na wartość zdefiniowaną przez użytkownika. W naszym przypadku powinna to być data, która jest mniejsza od wszystkich dat występujących w atrybucie `ostatnia_modyfikacja`. Ze względu na okres działania modelowanych wypożyczalni niech będzie to początek 2000 roku.

- Dodaj do diagramu komponent *If field value is null* z katalogu *Utility* i połącz go z komponentem `Polacz klientow`.



- Przejdź do ustawień komponentu
- Wprowadź nazwę komponentu `Zamien puste daty`.
- Zaznacz opcję *Select fields*, która umożliwi nam wyspecyfikowanie określonych atrybutów, które mają być przetwarzane przez ten krok transformacji.



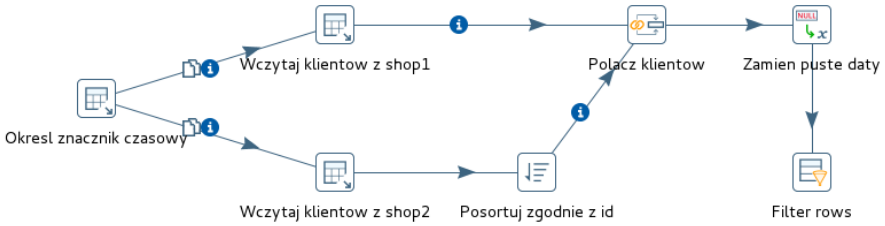
- W sekcji *Fields* wprowadź dane dla ustawienia znacznika czasowego równego `2000/01/01 00:00:00.000`. Wykonaj działanie dla obu dostępnych pól ze znacznikami czasowymi ostatniej modyfikacji.

Fields

	Field	Replace by value	Conversion mask (Date)
1	ostatnia_modyfikacja_shop1	2000/01/01 00:00:00.000	yyyy/MM/dd HH:mm:ss.SSS
2	ostatnia_modyfikacja_shop2	2000/01/01 00:00:00.000	yyyy/MM/dd HH:mm:ss.SSS

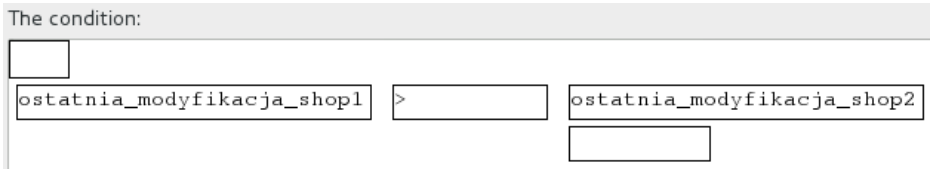
- Zatwierdź zmiany przyciskiem *OK*.

j) Mając dostępne daty modyfikacji krotek pochodzące z różnych źródeł wybierzmy to źródło, które będzie decydowało o zawartości wymiaru klienci w hurtowni danych. Dodaj znany nam już komponent *Filter rows* z katalogu *Flow* i połącz go z komponentem `Zamien puste daty`. Wybierz opcję *Main output of step* przy definiowaniu tego połączenia.



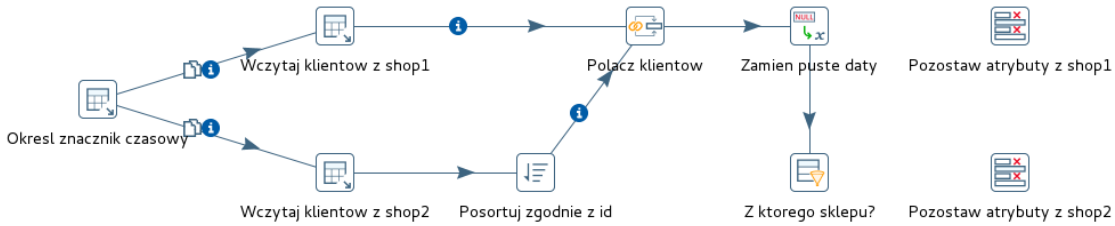
k) Przejdź do edycji nowo dodanego komponentu.

- Ustaw nazwę na `Z ktorego sklepu?`.
- Pozostaw na razie puste kierunki przesyłania krotek (*Send 'true' data to step* i *Send 'false' data to step*).
- Zdefiniuj warunek filtrowania krotek. Jako operator wybierz relację „większe od”. Po lewej stronie operatora umieść atrybut `ostatnia_modyfikacja_shop1` natomiast po prawej stronie atrybut `ostatnia_modyfikacja_shop2`.

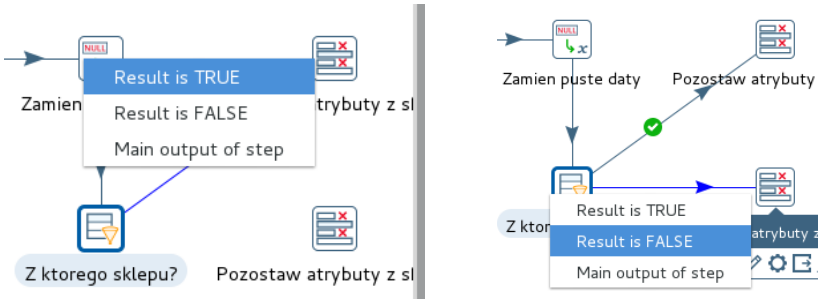


- Zatwierdź zmiany przyciskiem *OK*.

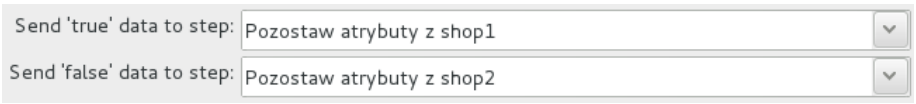
l) Komponent `Z ktorego sklepu?` zwróci dla każdej krotki wartość *true* albo *false* w zależności od tego, w której z baz danych informacje o klientach zostały wprowadzone później (i są dla nas ważniejsze). Wartość *true* oznacza, że to w bazie danych `shop1` są bardziej aktualne dane o wybranym kliencie podczas gdy wartość *false* wskazuje na bazę danych `shop2`. Zadaniem kolejnych dwóch kroków jest usunięcie atrybutów pochodzących z „nieaktualnej" bazy danych.

- Dodaj do diagramu dwa komponenty *Select values* z katalogu *Transform*.
- Nadaj komponentom nazwy `Pozostaw atrybuty z shop1` oraz `Pozostaw atrybuty z shop2`.
- 

m) Zdefiniuj połączenia do nowych komponentów wychodzące z komponentu `Z ktorego sklepu?`. Przy połączeniu do komponentu `Pozostaw atrybuty z shop1` wybierz opcję *Result is TRUE*. Przy połączeniu do komponentu `Pozostaw atrybuty z shop2` wybierz opcję *Result is FALSE*.



- Zweryfikuj jak zmieniła się definicja komponentu `Z ktorego sklepu?`.



n) Pozostaje nam zdefiniować, które atrybuty będą usuwane przez komponenty `Pozostaw atrybuty z shop1` i `Pozostaw atrybuty z shop2`.

- Przejdź do edycji komponentu `Pozostaw atrybuty z shop1`. Na zakładce *Select & Alter* wprowadź do tabeli informacje zawarte na poniższym zrzucie ekranu. Zauważ, że wybieramy tylko elementy pochodzące z bazy danych `shop1` i zamieniamy ich nazwy na oryginalne (bez przyrostka `_shop1`).

Select & Alter | Remove | Meta-data

Fields :

	Fieldname	Rename to	Length	Precision
1	klient_id_shop1	klient_id		
2	imie_shop1	imie		
3	nazwisko_shop1	nazwisko		
4	email_shop1	email		
5	adres_id_shop1	adres_id		
6	ostatnia_modyfikacja_shop1	ostatnia_modyfikacja		
7	ktory_sklep_shop1	ktory_sklep		

- Zatwierdź zmiany przyciskiem *OK*.

o) Przejdź do edycji komponentu `Pozostaw atrybuty z shop2` i wypełnij tabelę pod zakładką *Select & Alter* zgodnie z poniższym zrzutem ekranu.

Select & Alter | Remove | Meta-data

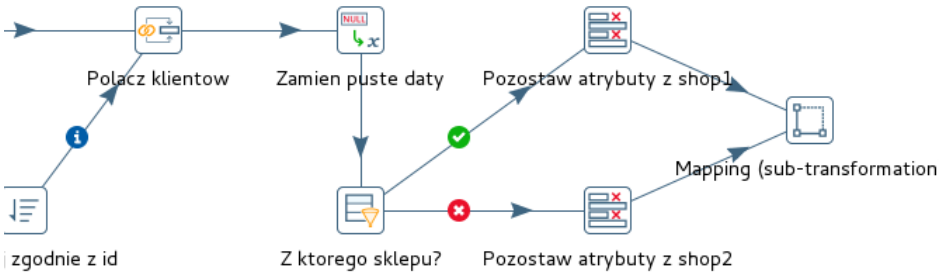
Fields :

	Fieldname	Rename to	Length	Precision
1	klient_id_shop2	klient_id		
2	imie_shop2	imie		
3	nazwisko_shop2	nazwisko		
4	email_shop2	email		
5	adres_id_shop2	adres_id		
6	ostatnia_modyfikacja_shop2	ostatnia_modyfikacja		
7	ktory_sklep_shop2	ktory_sklep		

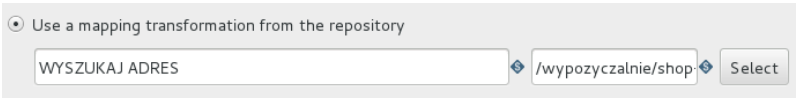
- Zatwierdź zmiany przyciskiem *OK*.

p) Przypomnijmy teraz, że klienci powinni być w hurtowni danych uzupełnieni o dane adresowe. Podobne działanie wykonywaliśmy już dla pracowników. Zdefiniowaliśmy wówczas transformację podrzędną do wyszukiwania adresu, którą zapisaliśmy pod nazwą `WYSZUKAJ ADRES` w katalogu `wypozyczalnie`. Wykorzystamy ją do określenia adresów zamieszkania klientów.

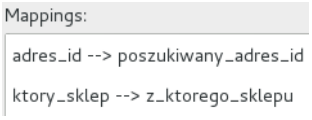
- Wstaw komponent *Mapping (sub-transformation)* z katalogu *Mapping*.
- Połącz go z komponentami `Pozostaw atrybuty z shop1` i `Pozostaw atrybuty z shop2`. W obu przypadkach wybierz opcję *Main output of step* jako typ połączenia.



- Przejdź do edycji nowo dodanego komponentu i wprowadź dla niego nazwę `Wyszukaj adres`.
- Jako źródło transformacji (*Mapping transformation*) wybierz opcję *Use a mapping transformation from the repository* i wskaż transformację `WYSZUKAJ ADRES` zapisaną w repozytorium w katalogu `wypozyczalnie/shop->dwh`.



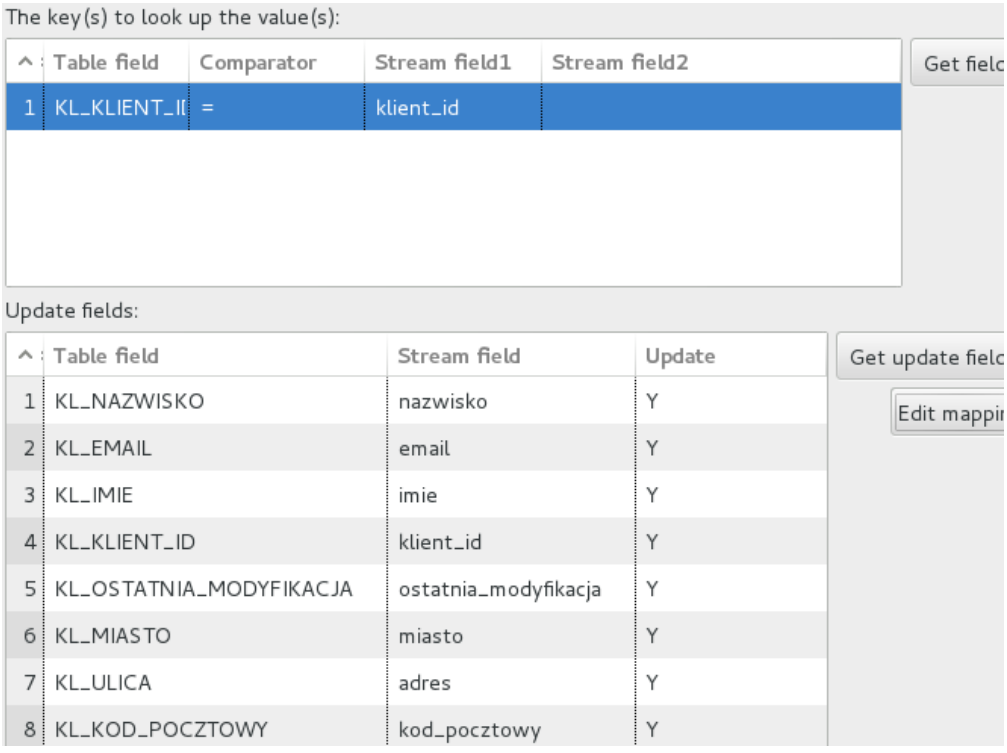
- Przejdź do zakładki *Input* i wybierz przycisk *Mapping…*.
- Powiąż ze sobą atrybuty `adres_id` i `poszukiwany_adres_id` oraz `ktory_sklep` i `z_ktorego _sklepu`.



- Dwukrotnie wybierz przycisk *OK* aby zatwierdzić wprowadzone zmiany.

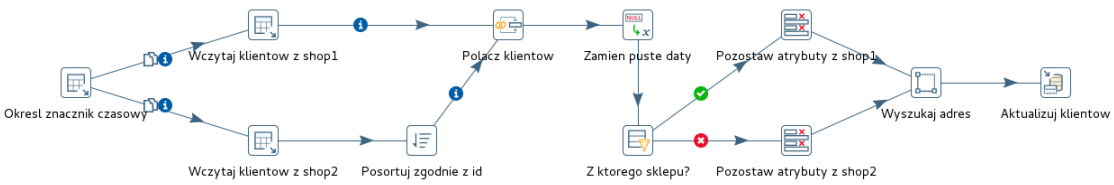
q) Ostatnim elementem transformacji jest, znany nam już z poprzedniego ćwiczenia, komponent *Insert/Update* odpowiedzialny za wprowadzenie zmian lub dodanie krotek do hurtowni danych.

- Dodaj komponent *Insert/Update* z katalogu *Output*.
- Połącz go z komponentem `Wyszukaj adres`.
- Przejdź do edycji nowo dodanego komponentu.
- Wprowadź nazwę `Aktualizuj klientow`.
- Użyj połączenia `dwh`.
- Jako relację do modyfikacji (*Target table*) wybierz relację `KLIENCI` ze schematu `DWH`.
- Ustaw mapowanie klucza i atrybutów do wstawienia/modyfikacji zgodnie z informacjami zawartymi na poniższym zrzucie ekranu.



- Zatwierdź zmiany wybierając przycisk *OK*.

r) Zapisz transformację w repozytorium wybierając opcję *File->Save*. Powinna ona teraz wyglądać następująco:



s) Na zakończenie uruchom transformację celem jej weryfikacji:

Execution Results

Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active
1	Okresl znacznik czasowy	0	0	2	1	0	0	0	0	Finished
2	Wczytaj klientow z shop2	0	1	566	566	0	0	0	0	Finished
3	Posortuj zgodnie z id	0	566	566	0	0	0	0	0	Finished
4	Wczytaj klientow z shop1	0	1	574	574	0	0	0	0	Finished
5	Polacz klientow	0	1140	598	0	0	0	0	0	Finished
6	Zamien puste daty	0	598	598	0	0	0	0	0	Finished
7	Z ktorego sklepu?	0	598	598	0	0	0	0	0	Finished
8	Pozostaw atrybuty z shop1	0	32	32	0	0	0	0	0	Finished
9	Pozostaw atrybuty z shop2	0	566	566	0	0	0	0	0	Finished
10	Wyszukaj adres	0	598	598	0	0	0	0	0	Finished
11	Aktualizuj klientow	0	598	598	598	598	0	0	0	Finished