

# Technologie Zasilania i Odświeżania Hurtowni Danych

laboratorium

część II

v20170324

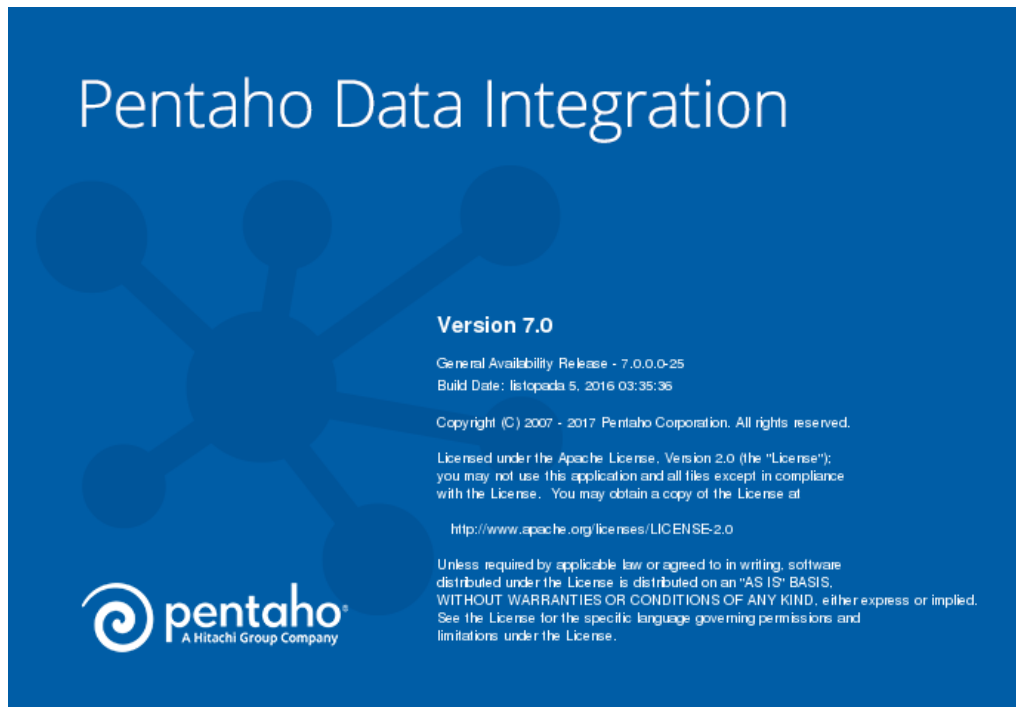
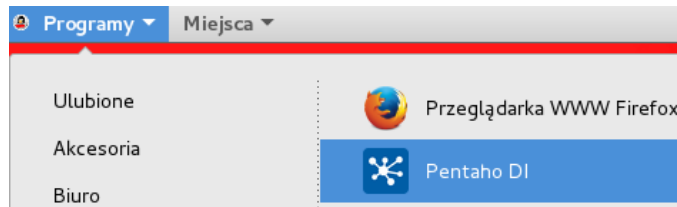
© Paweł Boiński, Krzysztof Jankiewicz

## I. Przygotowanie repozytorium i zdefiniowanie połączeń do źródłowych baz danych

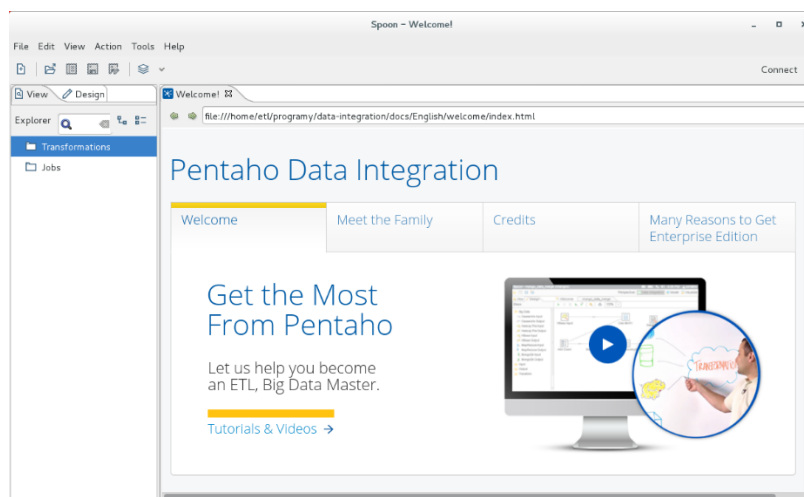
W poprzednim zestawie ćwiczeń utworzyliśmy bazę danych na potrzeby przechowywania danych repozytorium. W kolejnych punktach utworzymy nowe repozytorium, zdefiniujemy w nim strukturę katalogów oraz dodamy wymagane połączenia do źródłowych baz danych oraz hurtowni danych.

### 1. Utworzenie repozytorium

- a) Uruchom program *Pentaho Data Integration*. Możesz w tym celu wybrać z menu systemowego Programy wybrać zakładkę *Ulubione* -> *Pentaho DI*. Poczekaj na uruchomienie programu

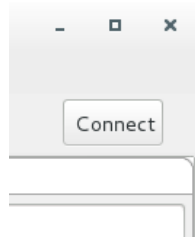


- b) Po uruchomieniu programu domyślnie zostanie wyświetlona strona powitalna.

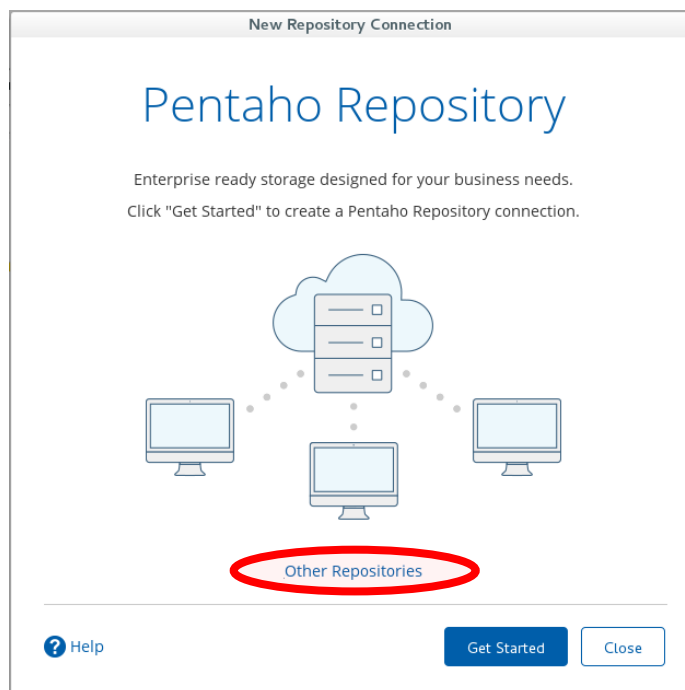


- c) Pierwszym naszym krokiem będzie podłączenie się do repozytorium. W tym celu użyj przycisku *Connect* w

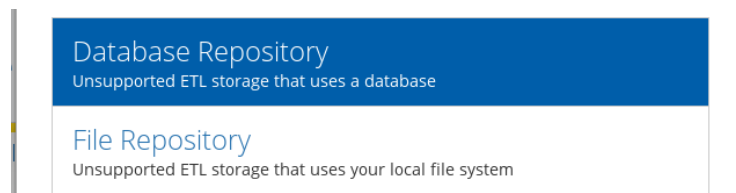
prawym, górnym rogu okna programu *Pentaho DI*.



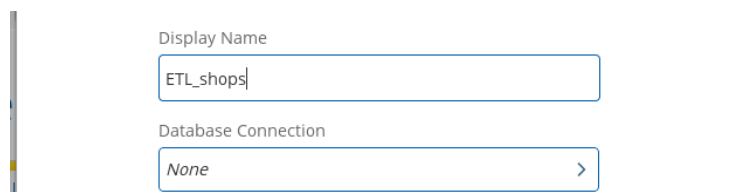
- d) Wyświetlone zostanie okno kreatora połączenia do repozytorium. W rzeczywistych zastosowaniach zalecany środowiskiem jest specjalny, osobny serwer repozytorium (*Pentaho Repository*). My nie potrzebujemy jednak jego funkcji, a dodatkowo mamy bardziej ograniczone zasoby sprzętowe. Użyjemy repozytorium przechowywanego w bazie danych. Jest ono dobrym kompromisem pomiędzy zaawansowanym serwerem repozytorium a niewygodnym przechowywaniem procesów ETL w plikach XML. Użyj opcji *Other Repositories*.



- e) Zaznacz opcję *Database Repository*. Potwierdź wybór przyciskiem *Get Started*.



- f) Ustaw nazwę repozytorium na `ETL_shops`. Następnie kliknij na pozycję `None` w polu *Database Connection*.

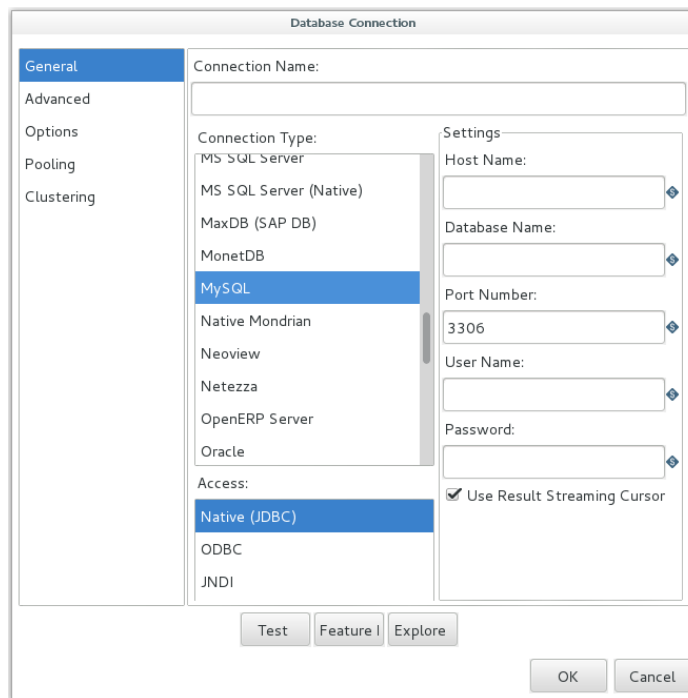


- g) Otwarte zostanie okno menadżera połączeń. Ponieważ jest to nasze pierwsze uruchomienie programu, nie ma jeszcze żadnego zdefiniowanego połączenia. Wybierz przycisk *Create New Connection*.

You do not have a database connection for this repository. Please create one.

Create New Connection

- h) W oknie definiowania połączenia do bazy danych wybierz z listy *Connection Type* opcję *MySQL*. Zauważ, że lista parametrów (*Settings*) dla połączenia uległa dostosowaniu do wybranego rodzaju bazy danych.



- Wprowadź w polu *Connection Name* łańcuch repo.
- Jako *Host Name* wprowadź wartość localhost.
- Właśność *Database Name* ustaw na repodb.
- Pozostaw domyślną wartość *Port Number* (3306).
- Jako nazwę użytkownika (*User Name*) podaj repouser z hasłem (*Password*) repouser.
- Upewnij się że wskazaną metodą komunikacji (*Access*) jest Native (JDBC).

Database Connection

General

Advanced

Options

Pooling

Clustering

Connection Name:  
repo

Connection Type:  
MS SQL Server  
MS SQL Server (Native)  
MaxDB (SAP DB)  
MonetDB  
MySQL  
Native Mondrian  
Neoview  
Netezza  
OpenERP Server  
Oracle

Settings:  
Host Name:  
localhost  
Database Name:  
repodb  
Port Number:  
3306  
User Name:  
repouser  
Password:  
.....  
 Use Result Streaming Cursor

Access:  
Native (JDBC)  
ODBC  
JNDI

Test Feature I Explore

OK Cancel

- i) Przetestuj działanie połączenia do bazy danych wybierając przycisk Test.

Database Connection Test

Connection to database [repo] is OK.  
Hostname :localhost  
Port :3306  
Database name :repodb

OK

- j) Zamknij okno *Database Connection Test* i zatwierdź nowe połączenie do bazy danych poprzez wybranie przycisku *OK*.
- k) W oknie wyboru połączenia do bazy danych automatycznie zostanie zaznaczone połączenie `repo`, które przed chwilą zostało utworzone. Wybierz przycisk *Back*.

New Repository Connection

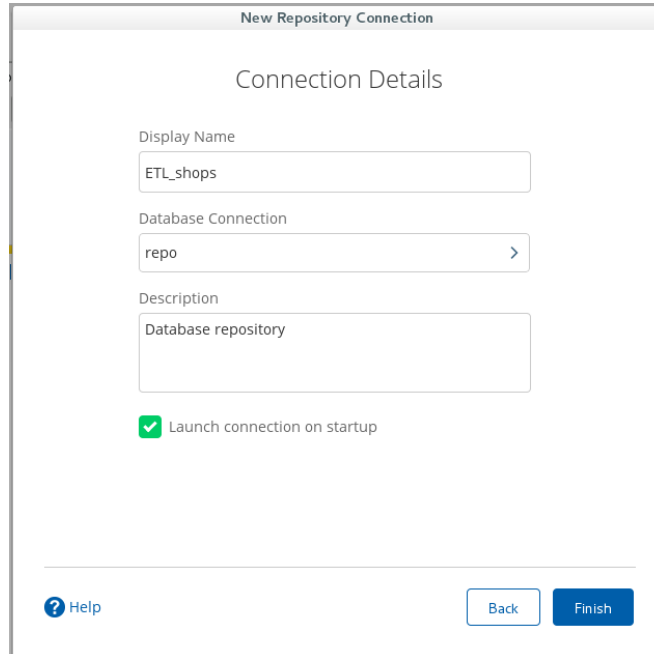
Select a database connection

repo

New Edit Delete

Help Back

- l) W oknie tworzenia połączenia do repozytorium upewnij się, że dla pola *Database Connection* wybrane jest połączenie *repo*. Jeżeli jest *None*, ponownie kliknij na nie przyciskiem i na kolejnym ekranie kliknij na połączenie *repo*, a następnie wybierz przycisk *Back*.
- m) Ponieważ będziemy korzystać tylko z jednego połączenia do repozytorium, zaznacz opcję *Launch connection on startup*. Zakończ działanie kreatora wybierając przycisk *Finish*.



New Repository Connection

Connection Details

Display Name  
ETL\_shops

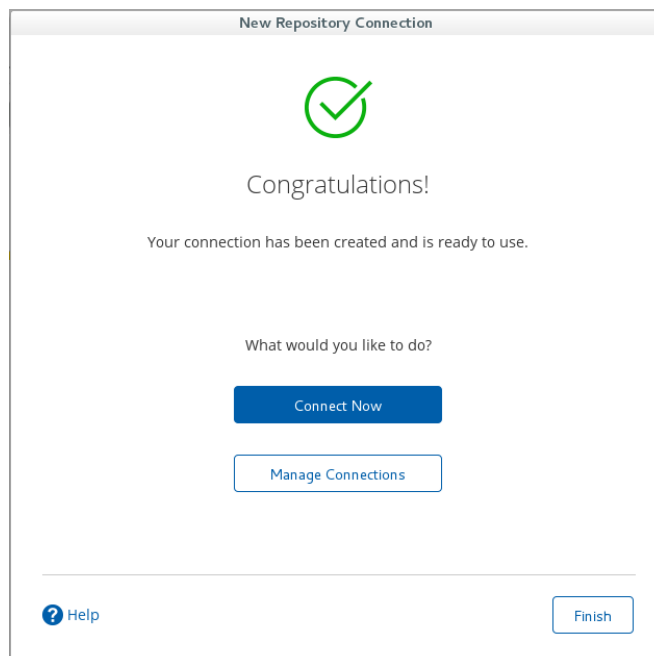
Database Connection  
repo

Description  
Database repository


Launch connection on startup

Help Back Finish

- n) Poczekaj na utworzenie połączenia. Automatycznie zostaną utworzone odpowiednie obiekty w bazie danych wskazanej przez połączenie bazodanowe. Wybierz przycisk *Connect Now*.



New Repository Connection



Congratulations!

Your connection has been created and is ready to use.

What would you like to do?

Connect Now

Manage Connections

Help Finish

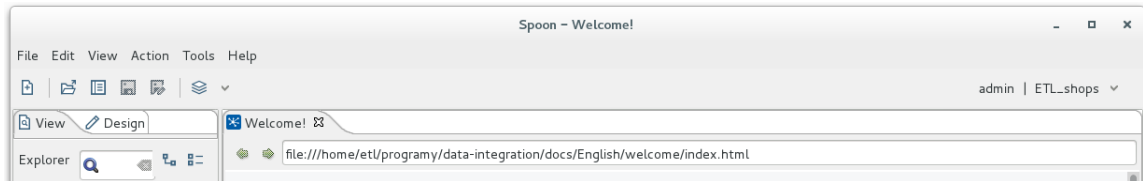
- o) Podaj domyślne hasło *admin* dla domyślnego użytkownika, również o nazwie *admin*.

Connect to  
ETL\_shops

User name

Password

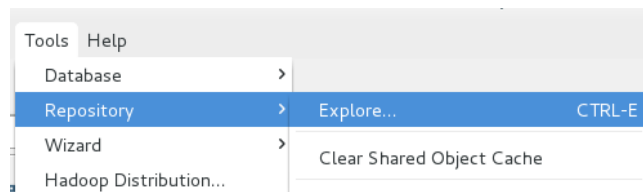
- p) Powróciliśmy do głównego ekranu. Zauważ, że w miejscu przycisku *Connect* pojawiła się nazwa użytkownika i repozytorium, do którego jesteśmy przyłączeni. Środowisko *Pentaho DI* jest gotowe do pracy.



## 2. Zdefiniowanie połączeń do źródłowych baz danych oraz hurtowni danych

Podczas realizacji ćwiczenia wykorzystywać będziemy dwie źródłowe bazy danych oraz hurtownie danych. W kolejnych krokach zdefiniujemy te połączenia w utworzonym przez nas repozytorium *repo*.

- a) Z głównego menu narzędziowego wybierz opcję *Tools->Repository->Explore*.



- b) Przejdź do zakładki *Connections*. Wybierz przycisk dodawania nowego połączenia (znaczek plus).



- c) Wybierz bazę danych *MySQL* i wprowadź poniższe dane:

- *Connection Name*: shop1
- *Host*: localhost
- *Database Name*: shop1
- *Port*: 3306
- *User Name*: shop1
- *Password*: shop1

d) Przetestuj połączenie korzystając z przycisku **Test**. Zatwierdź dodanie połączenia korzystając z przycisku **OK**.

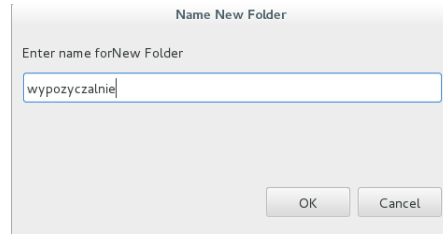
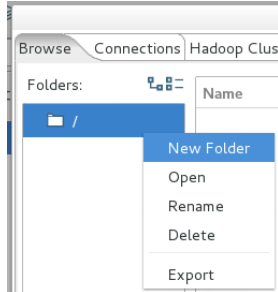
**Uwaga:** ze względu na błąd w *Pentaho DI* nie zostanie wyświetlone zdefiniowane połączenie na liście dostępnych połączeń. Zignorujmy to w tym momencie. W przyszłości i tak będziemy mogli skorzystać z połączenia shop1.

e) Dodajmy jeszcze połączenie do hurtowni danych. Wskaż jako **Connection Type** pozycję **Oracle** i wprowadź następujące dane:

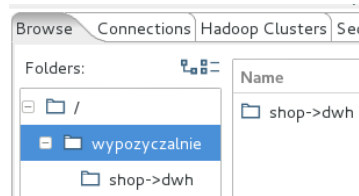
- **Connection Name:** dwh
- **Host:** localhost
- **Database Name:** XE
- **Port:** 1521
- **User Name:** dwh
- **Password:** dwh
- Pola **Tablespace for Data** i **Tablespace for Indices** pozostaw puste.



- f) Przetestuj połączenie korzystając z przycisku *Test*. Zatwierdź dane korzystając z przycisku *OK*.
- g) Zakończymy konfigurowanie repozytorium poprzez utworzenie w nim katalogu, w którym będziemy zapisywać elementy realizowanego projektu. Przejdź do zakładki *Browse*.
- h) Z menu kontekstowego wywołanego na głównym folderze repozytorium (/) wybierz opcję *New Folder*. Wprowadź nazwę *wypożyczalnie* i zatwierdź wybierając przycisk *OK*.



- i) Następnie wykorzystując menu kontekstowe dla katalogu *wypożyczalnie* utwórz podkatalog *shop->dwh*. Będzie on zawierał wszystkie transformacje potrzebne do zasilania hurtowni danych.

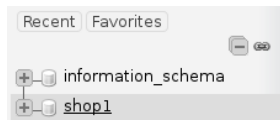


- j) Zamknij okno używając przycisku *Close*.

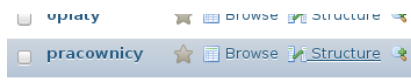
## II. Transformacja informacji o pracownikach na podstawie jednego źródła danych

Założmy na początku, że informacje o pracownikach będą pochodziły z jednego źródła danych (*shop1*). Problemem uwzględnienia drugiego źródła danych zajmiemy się w kolejnym ćwiczeniu.

1. Przypomnijmy strukturę relacji zawierającej dane o pracownikach.
  - a) W tym celu otwórz za pomocą menu *Przeglądarki WWW Firefox* program *phpMyAdmin* i zaloguj się podając nazwę użytkownika *shop1* i hasło *shop1*. Wybierz z lewego menu bazę danych *shop1*.



- b) Odszukaj na liście relację *pracownicy* i skorzystaj z odnośnika *Structure*.

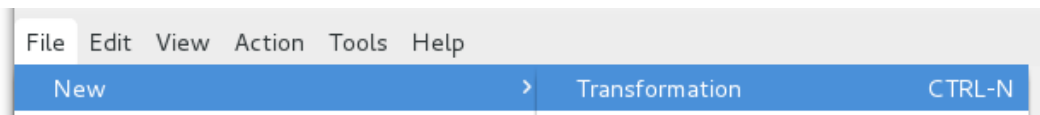


- c) Nie wszystkie atrybuty relacji *pracownicy* są dla nas interesujące. Przykładem mogą być dane dotyczące logowania do systemu informatycznego (*login* i *hasło*). Na wyróżnienie zasługuje także atrybut *adres\_id*. Przypomnijmy, że jest to klucz obcy wskazujący na relację *adresy*. W odróżnieniu od źródłowej bazy danych, w hurtowni danych chcemy, aby szczegółowe dane adresowe były przechowywane wraz z krotkami pracowników w relacji reprezentującej wymiar *pracownicy*. Będzie to oznaczało konieczność sięgnięcia do dwóch relacji źródłowych: *pracownicy* i *adresy*.

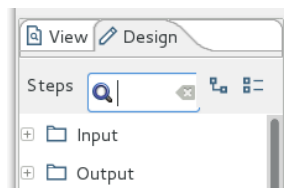
#	Name
1	pracownik_id
2	imie
3	nazwisko
4	adres_id
5	zdjecie
6	email
7	czy_aktywny
8	login
9	haslo
10	ostatnia_modyfikacja
11	sklep_zatr_id

2. Pobieranie danych ze źródła z uwzględnieniem znacznika czasu.

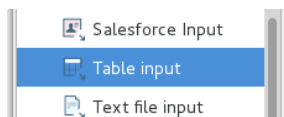
- a) W programie *Pentaho DI* utwórz nową transformację. W tym celu z głównego menu programu wybierz *File* -> *New* -> *Transformation*.



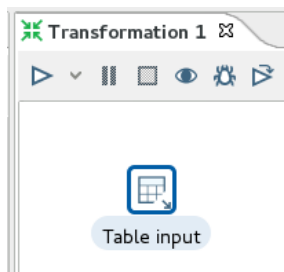
- b) W panelu bocznym przejdź do zakładki *Design*.



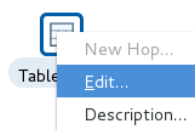
- c) Rozwiń katalog *Input* i odszukaj element *Table Input*. Kliknij na nim lewym przyciskiem. Uwaga: być może będziesz musiał za pierwszym razem poczekać kilka, kilkanaście sekund na reakcję środowiska (tj. zaznaczenie)



- d) Stosując technikę „Przeciągnij i upuść” umieść w/w element na obszarze zawartym w głównym panelu programu (nazwanym tymczasowo *Transformation 1*);



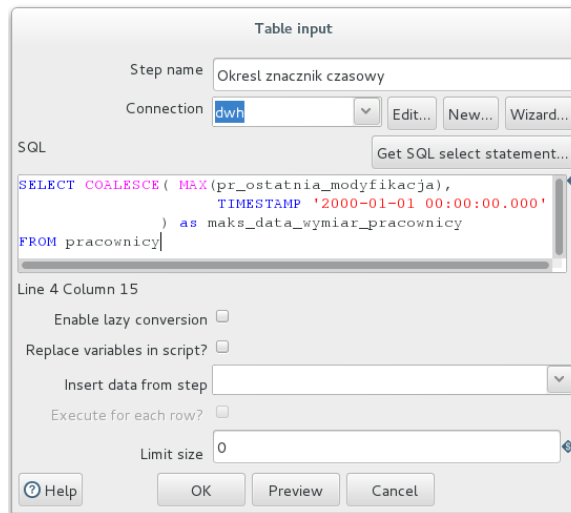
- e) Element *Table Input* stanowi pojedynczy krok transformacji. Przejdź do edycji tego kroku poprzez dwukrotne kliknięcie na nim lewym przyciskiem myszy lub wybierając opcję *Edit...* z menu kontekstowego dla tego elementu.



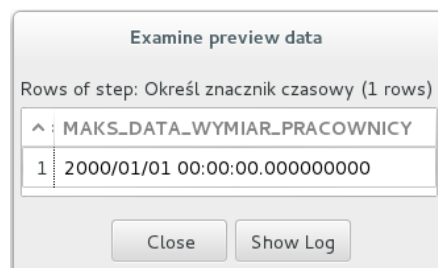
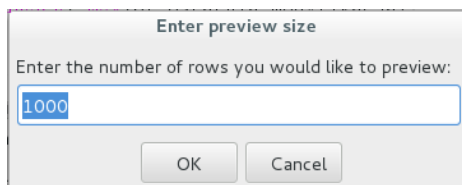
f) Zadaniem elementu *Table Input* jest odczyt krotek ze wskazanego źródła bazodanowego. Celem naszej transformacji jest pozyskanie danych o pracownikach, którzy ulegli zmianie (lub zostali dodani) od ostatniego momentu odświeżania hurtowni danych. W praktyce oznacza konieczność określenia znacznika czasowego (najbliższego aktualnemu), który znajduje się w hurtowni danych. W przypadku gdy jest to pierwsze zasilanie, czyli nie ma jeszcze danych w hurtowni, polecenie powinno zwrócić znacznik czasowy mniejszy od każdego ze znaczników czasowych obecnych w źródłowej relacji. Ponieważ w naszym przypadku będziemy rozpatrywali działalność wypożyczalni od 2005 roku, takim domyślnym znacznikiem może być początek roku 2000.

- Jako nazwę kroku (*Step Name*) ustaw `Okresl znacznik czasowy`.
- Wskaż połączenie `dwh`.
- Wprowadź następujące polecenie języka SQL:

```
SELECT COALESCE( MAX(pr_ostatnia_modyfikacja),
                TIMESTAMP '2000-01-01 00:00:00.000'
                ) as maks_data_wymiar_pracownicy
FROM pracownicy
```

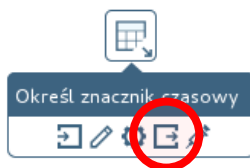


g) Wybierz przycisk *Preview* aby zobaczyć podgląd wyniku wykonania polecenia. Zostaniesz zapytany o liczbę krotek w podglądzie. Zaakceptuj podaną wartość. Otrzymany wynik jest zgodny z naszymi przypuszczeniami. Ponieważ jeszcze nie zasilaliśmy hurtowni danych, zwrócony został zdefiniowany przez nas „domyślny” znacznik czasowy.



- h) Zamknij okno z podglądem danych i zatwierdź zmiany w edycji komponentu przyciskiem *OK*.
- i) Dodaj kolejny krok transformacji. Podobnie jak poprzednio będzie to komponent *Table Input* z katalogu *Input*. Zadaniem tego kroku będzie odczyt pracowników z bazy danych `shop1`, którzy zostali zmodyfikowani po znaczniku czasowym wyznaczonym w poprzednio zdefiniowanym komponente. Ponieważ wyniki pierwszego kroku mają być elementem wejściowym dla kroku drugiego konieczne jest zdefiniowanie odpowiedniego przepływu. Istnieje kilka metod na wykonanie tej czynności. Jedną z nich wygląda następująco:

- Umieść kursor myszy nad elementem – źródłem danych. Po chwili zostanie narysowany zbiór ikon reprezentujących akcje związane z tym elementem.

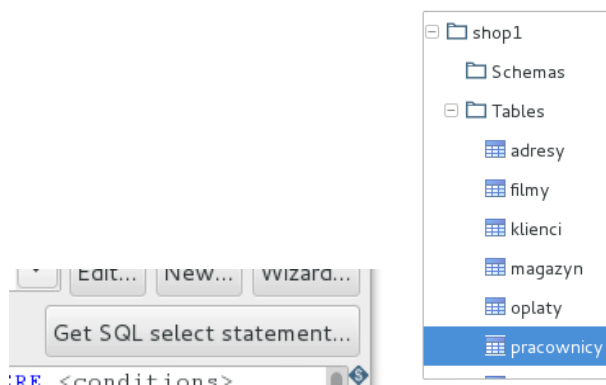


- Naciśnij lewy przycisk myszy na ikonie symbolizującej kartkę (prostokąt) ze strzałką „wychodzącą” i cały czas trzymając przycisk przesunij kursor na drugi z łączonych elementów. Puść lewy przycisk myszy. Połączenie wraz z kierunkiem przepływu danych zostanie ustanowione.

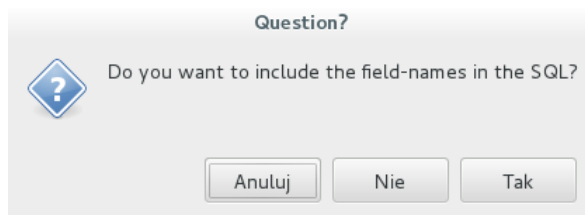


j) Przejdź do edycji drugiego z kroków transformacji.

- Jako nazwę wprowadź `Wczytaj pracowników z shop1`.
- Wybierz połączenie `shop1`.
- Wybierz przycisk *Get SQL select statement...*. Zostanie wyświetlone okno, w którym możesz wybrać relację będącą źródłem krotek. Wskaż tabelę `pracownicy` znajdującą się na ścieżce `shop1->Tables`.

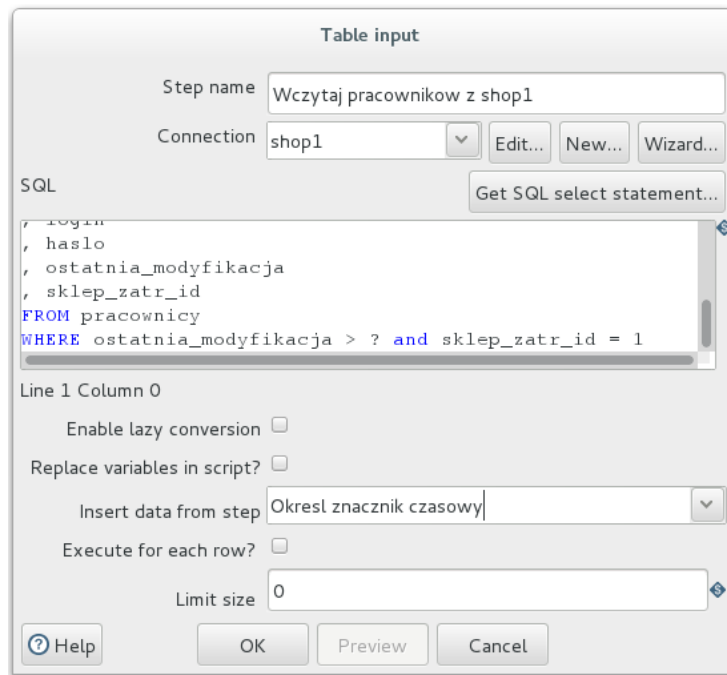


k) Zostaniesz zapytany o zawarcie w poleceniu SQL nazw wybieranych atrybutów. Zaakceptuj to działanie wybierając przycisk *Tak*.



- Zmodyfikuj wygenerowane polecenie SQL poprzez dodanie następującego warunku selekcji krotek:  

```
WHERE ostatnia_modyfikacja > ? and sklep_zatr_id = 1
```
- Parametr `?` zostanie zastąpiony przez wartość wyliczoną w kroku *Określ znacznik czasowy*. Wybierz tę nazwę z listy rozwijanej dla *Insert data from step*.

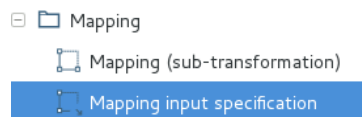


- Zatwierdź wprowadzone dane przyciskiem **OK**.

3. Kolejnym elementem transformacji jest uzupełnienie danych o pobranych pracownikach o elementy związane z adresem ich zamieszkania. Przypomnijmy, że adresy pracowników i klientów przechowywane są w osobnej relacji o nazwie `adresy` natomiast w relacjach `pracownicy` i `klienci` zostały zdefiniowane odpowiednie klucze obce do relacji `adresy`. Ponieważ takie samo działanie będzie wymagane dla klientów możemy wprowadzić transformację podrzędną dla wyszukiwania adresów. Będzie ona wykorzystywana zarówno przez transformację dla pracowników, jak również klientów.
  - a) Z bocznego panelu wybierz element **Mapping (sub-transformation)** z katalogu **Mapping**. Umieść go na diagramie i połącz z elementem `Wczytaj pracownikow z shop1` tak aby nowy komponent stanowił element przetwarzający krotki pochodzące z tabeli .

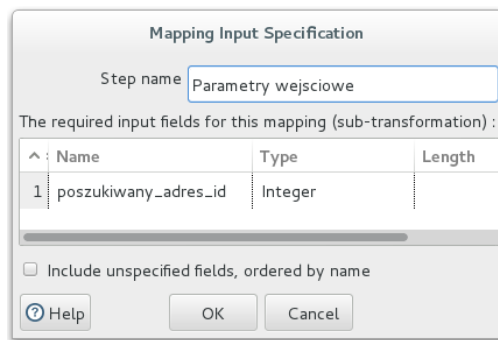


- b) Komponent **Mapping (sub-transformation)** wymaga określenia transformacji, która zostanie wywołana. Utwórzmy zatem nową transformację. Z głównego menu wybierz opcję **File->New->Transformation**.
- c) Z bocznego panelu wybierz i umieść w nowej transformacji komponent **Mapping input specification** z katalogu **Mapping**.



- d) Przejdź do edycji właściwości komponentu i wykonaj następujące działania:

- Ustaw wartość pola **Step Name** na Parametry wejściowe.
- Wypełnij jeden wiersz tabelki wpisując w polu **Name** łańcuch `poszukiwany_adres_id`.
- Ustaw dla tego pola typ **Integer**.
- Zatwierdź zmiany wybierając przycisk **OK**.



e) Dodaj drugi element transformacji – *Database lookup* z katalogu *Lookup*. Połącz ze sobą oba elementy.



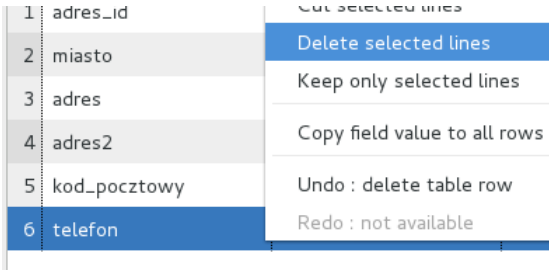
f) Przejdź do edycji nowo dodanego komponentu.

- Ustaw nazwę (*Step Name*) na *Wczytaj adres z shop1*.
- Wskaż połączenie *shop1*.
- Jako przeszukiwaną relację (*Lookup table*) wskaż relację *adresy*. Możesz wpisać nazwę ręcznie lub posłużyć się przyciskiem *Browse* i wybrać nazwę relacji z listy.

- Przejdź następnie do zdefiniowania warunku wyszukiwania (warunku jaki musi spełniać poszukiwana krotka). W naszym przypadku poszukujemy krotki relacji *adresy* o zadanym identyfikatorze.
- W sekcji *The key(s) to look up the value(s)* ustaw *Table field* na *adres\_id* (kreator podpowiada w postaci listy rozwijanej wartości dostępne w bazie danych).
- Użyj metody porównania (*Comparator*) opartej na relacji równości.
- W kolumnie *Field1* wybierz parametr *poszukiwany\_adres\_id*.

The key(s) to look up the value(s):				
^	Table field	Comparator	Field1	Field2
1	adres_id	=	poszukiwany_adres_id	

- Ostatnim zadaniem jest określenie, które atrybuty należy pobrać z bazy danych.
- W tym celu wybierz przycisk *Get lookup fields* umieszczony w dolnej części okna. Automatycznie zostanie wypełniona tabelka *Values to return from the lookup table*. Ponieważ domyślnie wybrane zostaną wszystkie atrybuty, należy usunąć te, które nie będą nam potrzebne.
- **!Uwaga!** Jeżeli po wybraniu przycisku *Get lookup fields* widzisz tylko jeden wiersz, jest to spowodowane zbyt małą wysokością okienka z właściwościami elementu transformacji. Zauważ, że nadal jest możliwe przeglądanie kolejnych wierszy przy użyciu niewielkiego paska przewijania po prawej stronie.
- Usuń wiersze z atrybutami *telefon* oraz *ostatnia modyfikacja*. Możesz tego dokonać posługując się menu kontekstowym lub klikając lewym przyciskiem myszy na numerze wiersza, a następnie posługując się klawiszami strzałek w górę i w dół aby wskazać interesujący nas wiersz. Naciśnięcie klawisza *delete* spowoduje usunięcie wiersza.



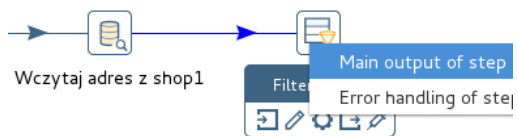
- **Upewnij się, że kolejność zwracanych elementów jest zgodna z poniższą tabelą.**

Jeżeli istnieje konieczność zmiany kolejności możesz tego dokonać wywołując prawym przyciskiem myszy menu kontekstowe na numerze wiersza i wybierając opcję *Move up* lub *Move down*.

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

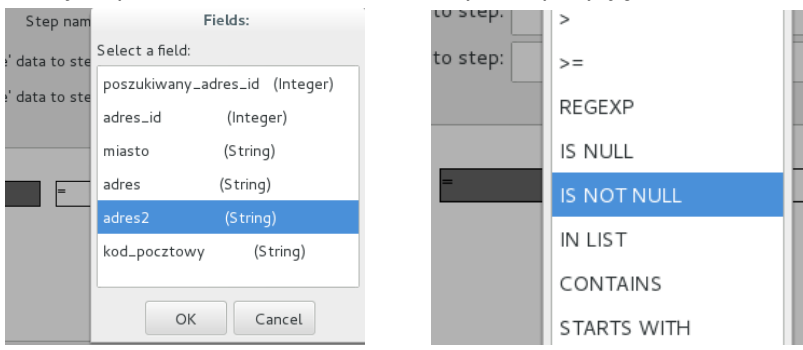
- Zatwierdź zmiany przyciskiem *OK*.

- g) W poprzednim zestawie ćwiczeń (podczas wykonywania profilowania danych) zauważyliśmy, że niektóre adresy w danych źródłowych mają wypełnione pola *adres* i *adres2* co sugeruje, że być może adres klienta się nie zmieścił w jednym polu formularza. W hurtowni danych chcemy przetwarzać tylko jedno pole z adresem. W związku z tym dokonamy konkatencji tych dwóch atrybutów dla krotek, w których występuje niepusta wartość atrybutu *adres2*.
- h) Umieść na diagramie transformacji element *Filter rows* z katalogu *Flow* i połącz go z komponentem *Wczytaj adres z shop1*. Podczas wykonywania połączenia zostaniesz poproszony o wybranie jednej z dwóch opcji: *Main output of step* wskazującej na prawidłowe przetwarzanie lub *Error handling of step* gdy komponent docelowy ma zostać aktywowany przy wystąpieniu błędu podczas przetwarzania komponentu źródłowego. Wybierz opcję *Main output of step*.



- i) Przejdź do edycji nowo dodanego komponentu.

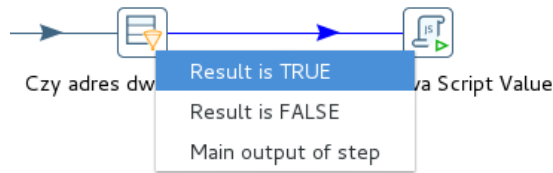
- Ustaw nazwę na *Czy adres dwulinijkowy?*
- W sekcji *The condition* kliknij na pole *<field>* po lewej stronie. Z listy wybierz pozycję *adres2*.
- Następnie kliknij na pole ze znakiem równości i wybierz pozycję *IS NOT NULL*.



- Zatwierdź zmiany przyciskiem *OK*.

- j) Krotki, dla których spełniony będzie warunek w zdefiniowanym filtrze będą wymagały wykonania konkatencji

dwóch łańcuchów znaków. W tym celu wykorzystamy komponent *Modified Java Script Value* z katalogu *Scripting*. Umieść go na diagramie transformacji i wykonaj połączenie z komponentem filtrowania. Przy połączeniu wybierz opcję *Result is TRUE*, które spowoduje, że do tego komponentu trafią tylko krotki, dla których warunek filtrowania jest spełniony.



k) Nowo wstawiony komponent umożliwia wykorzystanie podzbioru funkcji języka *JavaScript* do przeprowadzania działań na danych wejściowych. Przejdź do edycji komponentu.

- Ustaw nazwę na *Połącz linijki adresu*.
- W polu *Script 1* wprowadź następujący kod odpowiedzialny za konkatencję dwóch łańcuchów znaków:

```
var adres = adres + " " + adres2;
```



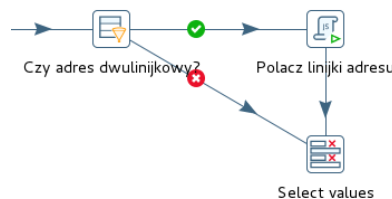
- Aby zmiany zostały wprowadzone w przetwarzanym wierszu konieczne jest zdefiniowanie wyjściowych pól w sekcji *Fields*. Można to zrobić dla naszego prostego skryptu automatycznie poprzez wybranie przycisku *Get variables*. Po wybraniu przycisku w tabeli zostanie umieszczona definicja atrybutu *adres*.

Fields						
	Fieldname	Rename to	Type	Length	Precision	Replace value
1	adres		String			Y

- Zatwierdź zmiany przyciskiem *OK*.

l) Kolejnym elementem przetwarzania będzie komponent pozwalający na wyeliminowanie niepotrzebnych atrybutów (takich jak atrybut *adres2*) z dalszego przetwarzania. Dodatkowo będą do niego przesyłane krotki nie posiadające wypełnionego drugiego pola adresu.

m) Umieść na diagramie transformacji komponent *Select values* z katalogu *Transform*. Połącz go zarówno z komponentem *Połącz linijki adresu* jak również z komponentem *Czy adres dwulinijkowy?*. W pierwszym przypadku wybierz opcję połączenia *Main output of step*. W drugim przypadku określ typ połączenia na *Result is FALSE*. W ten sposób krotki, które nie spełniały warunku filtrowania bezpośrednio trafią do nowo wstawionego komponentu, natomiast pozostałe krotki zostaną przetworzone przez komponent *Połącz linijki adresu*.

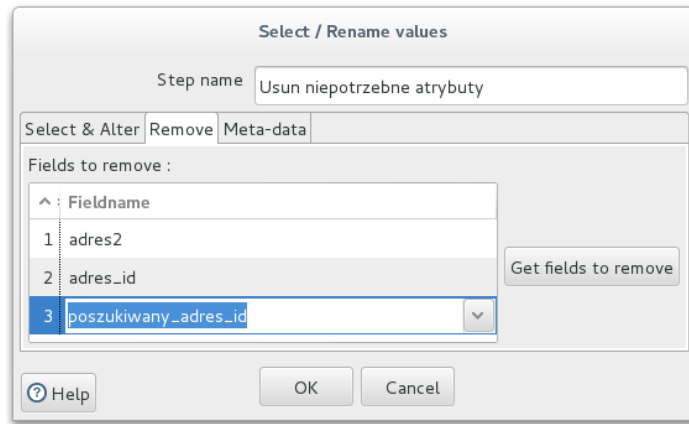


n) Przejdź do edycji nowo wstawionego komponentu.

- Ustaw jego nazwę na *Usun niepotrzebne atrybuty*.



- Przejdź do zakładki **Remove** (!) i dodaj do listy atrybuty `adres2`, `adres_id`, `poszukiwany_adres_id`.

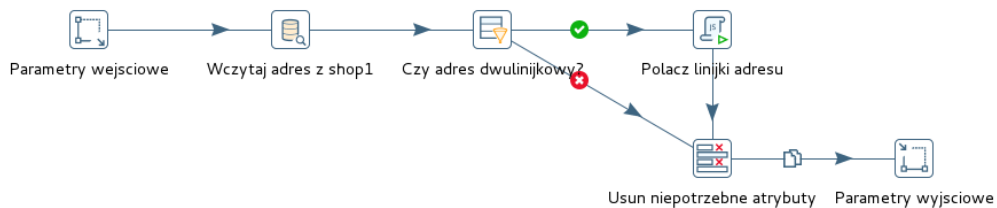


- Zatwierdź zmiany przyciskiem **OK**.

o) Ostatnim elementem transformacji podrzędnej będzie komponent odpowiedzialny za przekazanie danych do transformacji wywołującej.

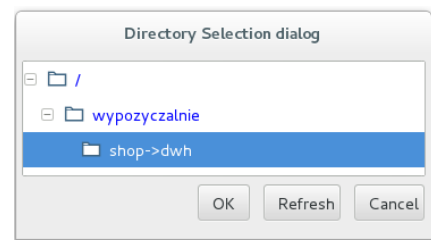
- Dodaj do diagramu transformacji komponent **Mapping output specification** z katalogu **Mapping**.
- Połącz nowo dodany komponent z komponentem **Usun niepotrzebne atrybuty** (wybierz opcję **Main output of step**).
- Przejdź do edycji komponentu. Jedynym parametrem, który możemy zmienić jest nazwa. Ustaw ją na **Parametry wyjsciowe**.

p) Ostatecznie transformacja podrzędna wyszukująca i przetwarzająca adres powinna wyglądać następująco:



q) Z głównego menu wybierz **File->Save As...**

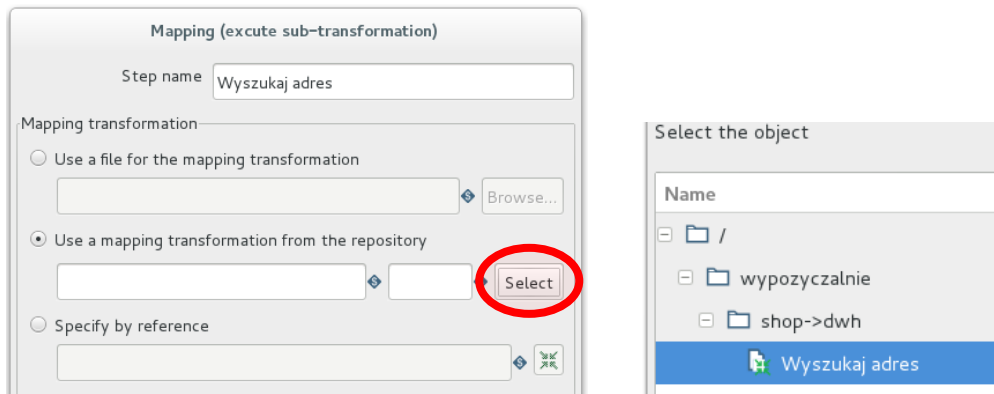
- Jako nazwę transformacji (**Transformation Name**) podaj wartość **Wyszukaj adres**.
- Wybierz ikonę folderu przy polu **Directory**. Wskaż folder **wypożyczalnie/shop->dwh**.



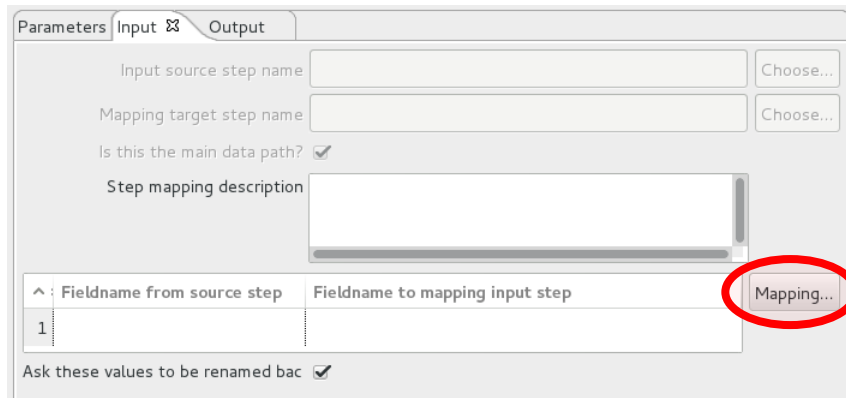
r) Wybierz przycisk **OK**. Transformacja zostanie zapisana.

s) Powróć do edycji transformacji dotyczącej pracowników (obecnie to **Transformation 1**) i przejdź do edycji komponentu **Mapping(sub-transformation)**.

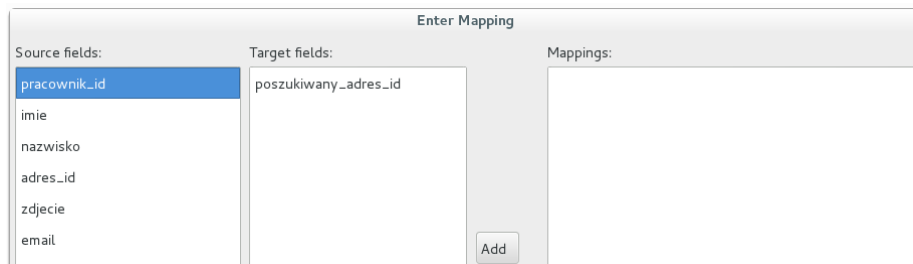
- Ustaw nazwę kroku na **Wyszukaj adres**.
- Jako źródło transformacji wybierz **Use a mapping transformation from the repository** a następnie wybierz przycisk **Select**. Otwarte okienko to przeglądarka repozytorium. Wskaż transformację **Wyszukaj adres** zapisaną w katalogu **wypożyczalnie/shop->dwh**. Zatwierdź wybór przyciskiem **OK**.



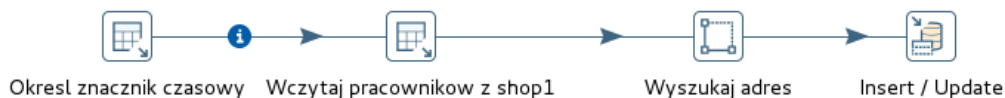
- t) Przejdź do zakładki *Input*. Wybierz przycisk *Mapping...* (uwaga: może być konieczne powiększenie okna by go zobaczyć).



- u) Na liście *Source fields* wybierz pozycję `adres_id`. Automatycznie zostanie wybrana jedyna pozycja z listy *Target fields* (`poszukiwany_adres_id`). Użyj przycisku *Add*.



- v) Zatwierdź dwukrotnie wybierając przycisk *OK*.
- w) Ostatnim elementem przetwarzania będzie komponent zapisujący dane do hurtowni danych. W tym celu dodaj do diagramu transformacji komponent *Insert/Update* z katalogu *Output*. Połącz nowo dodany element z komponentem *Wyszukaj adres*.



- x) W wykorzystywanej na zajęciach wersji środowiska *Pentaho DI* jest błąd związany z odświeżaniem komponentu *Insert/Update*. Rozwiązaniem jest zapisanie transformacji, zamknięcie jej i ponowne jej otwarcie.
- y) Zapisanie transformacji. Wybierz z głównego menu opcję *File->Save As...*

- Wprowadź nazwę `Zaladuj pracownikow`.
- Jako miejsce składowania transformacji wybierz folder `wypozyczalnie/shop->dwh`.

- Zatwierdź zapis wybierając przycisk **OK**.
- z) Zamknij transformację (np. poprzez naciśnięcie ikony krzyżyka w pasku zakładek).
- aa) Otwórz transformację Zaladuj pracowników.
- bb) Przejdź do edycji komponentu **Insert / Update**.
- cc) Ustaw nazwę na Aktualizuj pracowników w dwh.
- Wskaż połączenie dwh.
  - Jako tabelę docelową (**Target table**) wskaż tabelę PRACOWNICY. Możesz użyć przycisku **Browse** i wybrać tabelę z listy.

- Ponieważ krotki pracowników mają być aktualizowane lub wstawiane do hurtowni danych, konieczne jest wyspecyfikowanie, które atrybuty (lub atrybut) danych źródłowych stanowią klucz podstawowy relacji w hurtowni danych. Dla opisywanej transformacji należy wskazać, że atrybut **pracownik\_id** odpowiada wartościom atrybutu PR\_PRAC\_ID (klucz podstawowy w hurtowni danych dla relacji PRACOWNICY).
- W sekcji **The key() to look up the value(s)** wprowadź wiersz, w którym atrybut **Table field** ustawiony jest na PR\_PRAC\_ID, **Comparator** to relacja równości a **Stream field** to pracownik\_id.

The key(s) to look up the value(s):					
^	Table field	Comparator	Stream field1	Stream field2	Get fields
1	PR_PRAC_ID	=	pracownik_id		

- Ostatnim elementem definicji komponentu **Insert/Update** jest wskazanie, które pola w docelowej relacji będą modyfikowane (kolumna **Table field**) i jakie zostaną przypisane im wartości (kolumna **Stream field**) w tabelce **Update fields**. Wprowadź w niej dane zgodne z poniższym zrzutem ekranu. Wykorzystaj w tym celu przycisk **Edit mapping** i zdefiniuj odpowiednie powiązania.

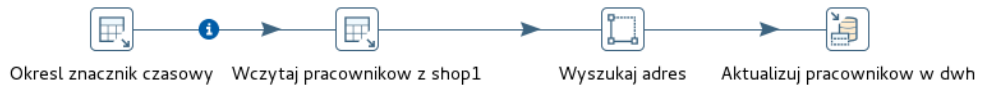
Update fields:

Table field	Stream field	Update
1 PR_PRAC_ID	pracownik_id	Y
2 PR_IMIE	imie	Y
3 PR_NAZWISKO	nazwisko	Y
4 PR_EMAIL	email	Y
5 PR_OSTATNIA_MODYFIKACJA	ostatnia_modyfikacja	Y
6 PR_MIASTO	miasto	Y
7 PR_ULICA	adres	Y
8 PR_KOD_POCZTOWY	kod_pocztowy	Y

Get update fields  
Edit mapping

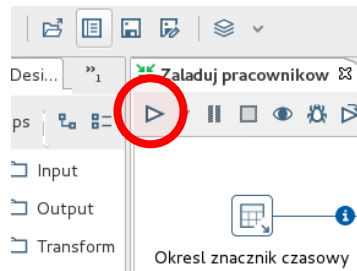
- Zatwierdź zmiany wybierając przycisk **OK**.

dd) Zapisz transformację (*File->Save*)

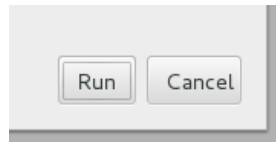


ee) Na zakończenie uruchom tak zdefiniowaną transformację w celu sprawdzenia jej poprawności.

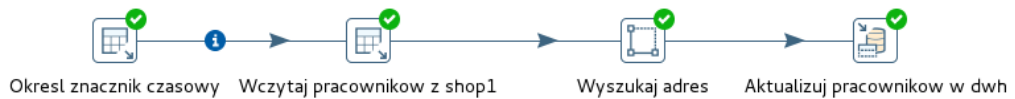
- Użyj przycisku uruchamiania transformacji.



- W oknie specyfikowania parametrów wykonania wybierz przycisk **Run**.



- Zwróć uwagę na kolumnę **Errors** (nie powinno być żadnego błędu).



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