

# Technologie Zasilania i Odświeżania Hurtowni Danych

laboratorium

część IV

v20170324

© Paweł Boiński, Krzysztof Jankiewicz

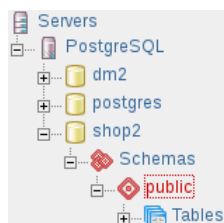
## I. Zasilanie wymiaru z wykorzystaniem zastępczego klucza. Metoda Trigger Based Change Detection.

W poprzednio zdefiniowanych transformacjach mieliśmy w obu źródłowych bazach danych spójne informacje o identyfikatorach klientów i pracowników. Oznacza to, że jeżeli dane klienta lub pracownika znajdują się w obu bazach danych, to w każdej z nich taki klient bądź pracownik ma przypisany ten sam identyfikator. Do tej pory używaliśmy tego identyfikatora jako klucza również w hurtowni danych. Takiego mechanizmu nie możemy jednak zastosować w przypadku filmów, które nie są w pełni replikowane pomiędzy systemami i mogą posiadać różne identyfikatory w źródłowych bazach danych. Wiemy jednak, że unikalne są tytuły filmów i lata ich wydania na płytach DVD.

Dodatkowo przypomnijmy sobie, że podczas analizy źródeł danych zauważyliśmy brak atrybutu zawierającego datę ostatniej modyfikacji (lub dodania) filmu. Dotychczas wykorzystywaliśmy taką informację do określenia, które krotki będą uczestniczyły w odświeżaniu hurtowni danych. Jednym z możliwych rozwiązań jest modyfikacja istniejących źródeł danych poprzez dodanie wyzwalaczy (procedur wyzwalanych), które będą dbały o aktualizowanie znaczników czasowych podczas operacji modyfikacji i wstawiania danych.

1. Na początku zmodyfikujemy źródło danych shop2.

- a) Zaloguj się przy pomocy programu *phpPgAdmin* do bazy danych shop2 (przypomnienie: nazwa i hasło użytkownika to shop2). Przejdź do schematu *public*.



- b) Skorzystaj z odnośnika *SQL* w prawym górnym rogu ekranu. Efektem będzie ukazanie się okna przeglądarki, w którym możemy wprowadzać instrukcje języka SQL.
- c) Wprowadź następujące instrukcje i wykonaj je poprzez wybranie przycisku *Execute*. Pierwsza instrukcja dodaje nowy atrybut do relacji *filmy*. Druga instrukcja przypisuje istniejącym krotkom w bazie danych wartość aktualnego znacznika czasowego.

**Uwaga: wyniki działania operacji będą wyświetlone w oryginalnym oknie przeglądarki (a nie w oknie z poleceniami)!**

```
alter table filmy add ostatnia_modyfikacja timestamp;  
update filmy set ostatnia_modyfikacja = current_timestamp;
```

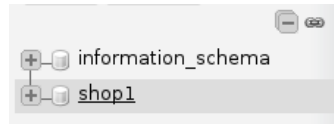
- a. Kolejny zbiór instrukcji jaki należy wykonać to polecenia tworzące procedurę wyzwalaną. Skopiuj i wykonaj następujące instrukcje:

```
CREATE OR REPLACE FUNCTION aktualizuj_czas()  
  RETURNS TRIGGER AS $$  
  BEGIN  
    NEW.ostatnia_modyfikacja = now();  
    RETURN NEW;  
  END;  
  $$ language 'plpgsql';
```

```
CREATE TRIGGER trg_nowy_film BEFORE UPDATE OR INSERT
```

```
ON filmy FOR EACH ROW EXECUTE PROCEDURE
aktualizuj_czas();
```

2. Podobne działania jak w poprzednim punkcie wykonane muszą zostać w bazie danych `shop1`. Tym razem skorzystamy z możliwości przypisania specjalnego atrybutu do nowej kolumny.
  - a) Zaloguj się do bazy danych `shop1` wykorzystując narzędzie *phpMyAdmin*. Wybierz z lewego menu bazę `shop1`.

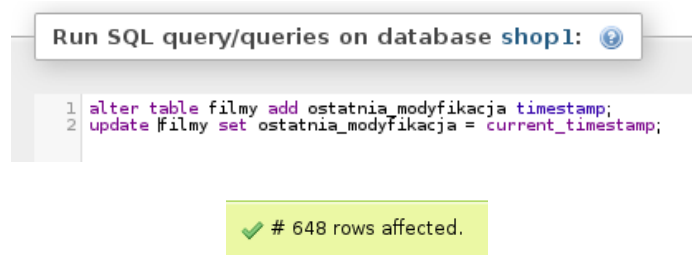


- b) Przejdź do zakładki z możliwością wprowadzania poleceń języka SQL.



- c) Wykonaj następujący zbiór poleceń:

```
alter table filmy add ostatnia_modyfikacja timestamp;
update filmy set ostatnia_modyfikacja = current_timestamp;
```



- d) Analogicznie jak w przypadku bazy danych `shop2` relacja `filmy` zyskała nowy atrybut `ostatnia_modyfikacja`. Dla istniejących w bazie danych krotek została przypisana wartość aktualnego znacznika czasowego. Baza danych `shop1` zarządzana jest przez *SZBD MySQL*, w którym dodanie pojedynczego atrybutu o typie *timestamp* powoduje ustawienie dla takiej kolumny domyślnej wartości *CURRENT\_TIMESTAMP* oraz właściwości *ON UPDATE CURRENT\_TIMESTAMP*. Oznacza to, że dla naszych zastosowań nie musimy tworzyć własnej procedury wyzwalanej.



3. Mając odpowiednio skonfigurowane źródła danych możemy przejść do utworzenia transformacji zasilającej wymiar `filmy`.
  - a) Dodaj nową transformację.
  - b) Dodaj komponent *Table Input* z katalogu *Input*.
    - Ustaw jego nazwę na `Okresl znacznik czasowy`.
    - Wskaż połączenie do bazy danych `dwh`.
    - Wprowadź następujące polecenie SQL:

```
SELECT COALESCE (
    MAX(FI_ostatnia_modyfikacja) ,
    TIMESTAMP '2001-01-01 00:00:00') as
maks_data_wymiar_filmy
```

## FROM FILMY

- Zatwierdź zmiany przyciskiem **OK**.
- c) Dodaj drugi komponent **Table Input** i połącz go z poprzednim tak aby komponent **Okresl znacznik czasowy** stanowił źródło danych. W nowo dodanym komponencie:
- Wprowadź nazwę **Wczytaj filmy z shop1**.
  - Wygeneruj (i uzupełnij) polecenie SQL, które z bazy danych **shop1** odczyta wszystkie informacje o filmach ale tylko takich, które zostały zmodyfikowane po znaczniku czasowym wyznaczonym przez komponent **Okresl znacznik czasowy**.

Table input

Step name: Wczytaj filmy z shop1

Connection: shop1

SQL: `SELECT  
film_id  
, tytul  
, data_wydania  
, ostatnia_modyfikacja  
FROM filmy  
where ostatnia_modyfikacja > ?`

Line 7 Column 30

Enable lazy conversion

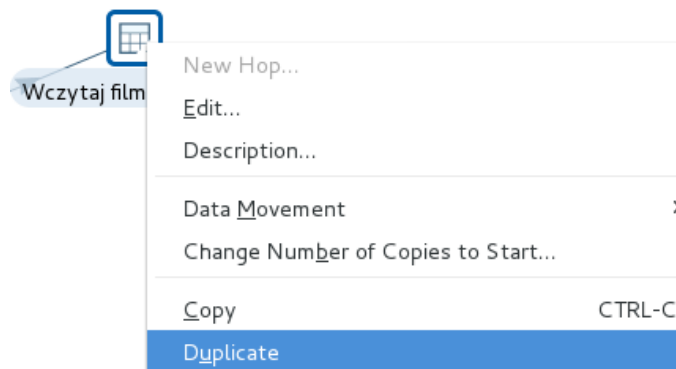
Replace variables in scrip

Insert data from step: Okresl znacznik czasowy

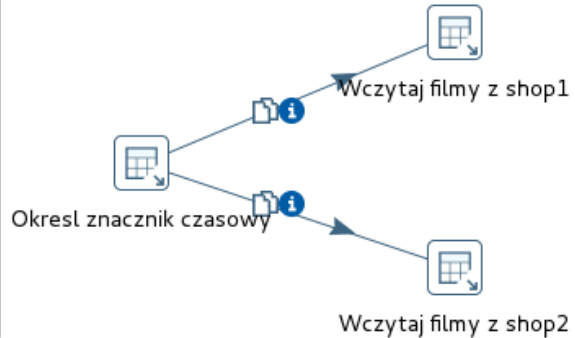
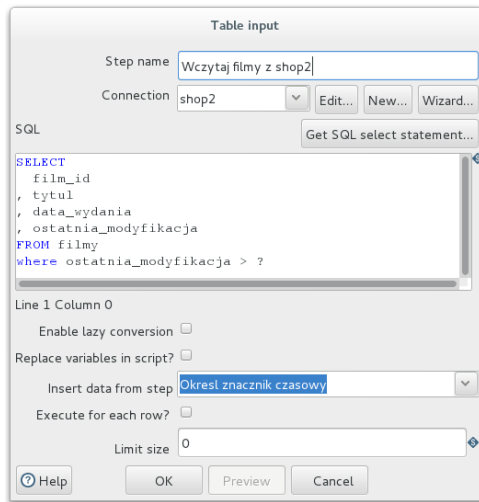
Execute for each row?

Limit size: 0

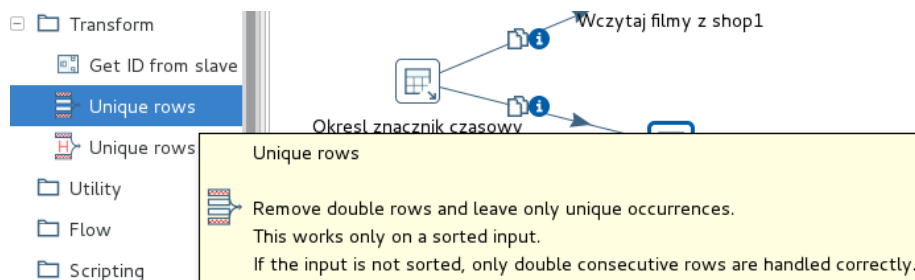
- d) Z menu kontekstowego komponentu **Wczytaj filmy z shop1** wybierz polecenie **Duplicate**.



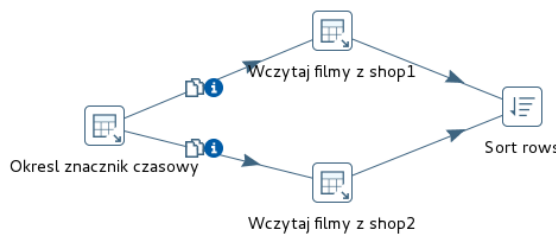
- e) Zmień właściwości utworzonego w ten sposób komponentu tak aby odczytywał klientów z bazy danych **shop2**. Zmień jego nazwę na **Wczytaj filmy z shop2** i połącz go z komponentem **Okresl znacznik czasowy** (wykorzystaj metodę kopiowania przy definiowaniu połączenia).



- f) Ponieważ dane o tych samych filmach mogą występować w obu źródłach danych konieczne jest wyeliminowanie duplikatów. Wykorzystamy do tego komponent *Unique rows* z katalogu *Transform*. Przesuń wskaźnik myszy nad nazwą komponentu. Po chwili zostanie wyświetlony krótki opis jego działania. Dodatkowo jesteśmy informowani, że komponent ten działa tylko dla posortowanego zbioru wierszy.



- g) Zauważ, że wykorzystane w poprzednim zestawie ćwiczeń, sortowanie w źródłach nie zapewni nam wymaganego globalnego posortowania. Dodaj zatem komponent *Sort rows* z katalogu *Transform*. Połącz go z komponentami odczytującymi dane o filmach w taki sposób, że jest on konsumentem wierszy przez nie zwracanych.



- Przejdź do edycji nowo dodanego komponentu. Ustaw nazwę na *Posortuj filmy*.
- Zdefiniuj warunek sortowania oparty na atrybutach *data\_wydania* i *tytul*. Kierunek sortowania może być dowolny. Dla atrybutu *tytul* ustaw opcję *Case sensitive compare?* na wartość *Y*, co oznacza, że wielkość liter będzie miała znaczenie przy sortowaniu.

Fields :		
^ : Fieldname	Ascending	Case sensitive compare?
1 data_wydania	Y	N
2 tytul	Y	Y

- h) Dodaj komponent *Unique rows* z katalogu *Transform* i połącz go z komponentem *Posortuj filmy*.
- Ustaw jego nazwę na *Wybierz unikalne filmy*.

- Jako atrybuty, które mają uczestniczyć w badaniu unikalności wierszy wskaż atrybuty `data_wydania` i `tytul`.
- Potwierdź podczas zapisywania ustawień komponentu zrozumienie potrzeby dostarczenia posortowanych wierszy.

Fields to compare on (no entries means: compare complete row)	
Fieldname	Ignore case
1 data_wydania	
2 tytul	

- Ostatnim krokiem transformacji jest zapisanie danych do hurtowni danych. Ponieważ filmy (lub ich podzbiór) mogą już istnieć w relacji wymiaru konieczne jest ich zaktualizowanie. Dla nowych filmów należy wygenerować ich sztuczny identyfikator.
- Dodaj komponent *Combination lookup/update* z katalogu *Data Warehouse*. Połącz go z komponentem Wybierz unikalne filmy. Przejdź do edycji komponentu.
  - Ustaw nazwę na Zaktualizuj filmy w dwh.
  - Komponent ma korzystać z połączenia dwh i przeprowadzać działania na relacji FILMY.

- Jako elementy klucza służące do wyszukania istniejącego w hurtowni filmu wskaż atrybuty `tytul` i `data_wydania` odpowiadające kolumnom `FI_TYTUL` oraz `FI_ROK_WYDANIA`.

Key fields (to look up row in table):	
Dimension field	Field in stream
1 FI_TYTUL	tytul
2 FI_ROK_WYDANIA	data_wydania

- Wskaż pole klucza technicznego/zastępczego `FI_FILM_ID`.

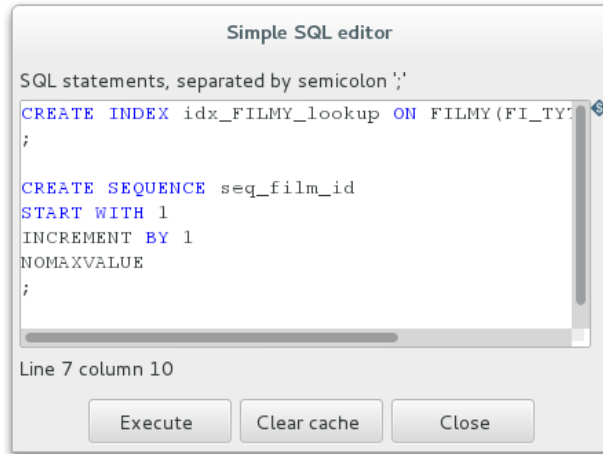
Technical key field

- Jako metodę generowania nowych wartości klucza wybierz sekwencję o nazwie `seq_film_id`.

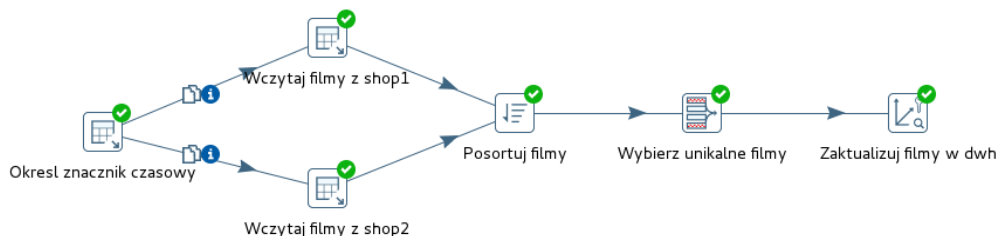
- Jako parametr *Date of last update* field podaj FI\_ostatnia\_modyfikacja.

Date of last update field (optional)

- Użyj przycisku *SQL* aby wygenerować sugerowane polecenia DDL. Ponieważ sekwencja seq\_film\_id nie istnieje, *Pentaho DI* zaproponuje jej utworzenie. *Pentaho DI* zaproponuje również utworzenie indeksu – struktury wspomagającej wyszukiwanie. Wybierz przycisk *Execute*.



- Zamknij okna związane z wykonaniem polecenia SQL i zatwierdź zmiany w komponencie przyciskiem *OK*.
- Nasza transformacja jest gotowa. Komponent Zaktualizuj filmy w dwh wstawi brakujące wiersze do relacji wymiaru jeżeli nie istniały one tam poprzednio. W przeciwnym wypadku zaktualizowane zostaną znaczniki czasowe (FI\_ostatnia\_modyfikacja).
  - Zapisz transformację pod nazwą ZALADUJ\_FILMY w katalogu wypożyczalnie/shop->dwh.
  - Uruchom transformację i sprawdź wynik jej działania.



## Execution Results

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



Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected
1 Okresl znacznik czasowy	0	0	2	1	0	0	0
2 Wczytaj filmy z shop1	0	1	648	648	0	0	0
3 Wczytaj filmy z shop2	0	1	663	663	0	0	0
4 Posortuj filmy	0	1311	1311	0	0	0	0
5 Wybierz unikalne filmy	0	1311	900	0	0	0	0
6 Zaktualizuj filmy w dwh	0	900	900	900	900	0	0

## II. Utworzenie wymiaru z obliczonymi wcześniej wartościami

Jedną z technik obsługi czasu w ramach zdefiniowanego w hurtowni wymiaru jest możliwość wcześniejszego

wygenerowania wszystkich możliwych/potrzebnych wpisów. Alternatywne rozwiązanie polega na uzupełnianiu wymiaru czasu podczas wykonywania zasilania tabeli faktów. Każde z tych rozwiązań ma wady i zalety. W ramach tego ćwiczenia wypełnimy wymiar czasu wszystkimi wartościami jaki mogą zostać wykorzystane podczas kolejnych zasileń hurtowni danych.

Interesują nas daty z dokładnością do dni. Daty wypożyczeń (czyli faktów) w naszych źródłach obejmują zakres od maja 2005 roku do czerwca roku 2006. Wystarczy zatem wygenerować wpisy dotyczące takiego zakresu. W praktyce najczęściej generuje się określony zbiór danych, który po upływie pewnego czasu uzupełnia się kolejnymi wpisami. W ramach ćwiczenia wygenerujemy sekwencję 2000 kolejnych dat począwszy od dnia 01.01.2005.

## 1. Generowanie sekwencji dat.

- Utwórz nową transformację poprzez wybranie z głównego menu polecenia **File->New->Transformation**.
- Dodaj komponent **Generate Rows** z katalogu **Input**.
- Przejdź do edycji nowo dodanego komponentu.
  - Ustaw nazwę komponentu na **Wygeneruj wiersze**.
  - Określ liczbę generowanych wierszy poprzez ustawienie wartości parametru **Limit** na 2000.

Step name	Wygeneruj wiersze
Limit	2000

- Generowane wiersze mają zawierać tylko jeden atrybut typu data.
- W tabeli **Fields** dodaj w kolumnie **Name** wartość **data\_początkowa**.
- Ustaw jej typ na **Date**.
- Jako format daty wybierz z listy rozwijanej wartość **yyyy/MM/dd**.
- W kolumnie **Value** wprowadź **2005/01/01**.

^	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value
1	data_początkowa	Date	yyyy/MM/dd						2005/01/01

- Użyj przycisku **Preview** do podglądu wygenerowanych danych. Ogranicz liczbę wyświetlanych krotek do pięciu sztuk.

Enter preview size

Enter the number of rows you would like to preview:

OK Cancel

Examine preview data

Rows of step: Wygeneruj wiersze

^	data_początkowa
1	2005/01/01
2	2005/01/01
3	2005/01/01
4	2005/01/01
5	2005/01/01

Close Show Log



- Zauważ, że we wszystkich wygenerowanych krotkach występuje dokładnie taka sama data.
  - Zamknij podgląd wybierając przycisk *Close*. Zatwierdź komponent wybierając przycisk *OK*.
- d) W kolejnym kroku dodamy komponent, który dla każdej krotki wejściowej doda numer sekwencyjny. Odpowiednio dobierając parametry taką wartość można potraktować jako liczbę dni, które upłynęły od daty przechowywanej w atrybucie `data_początkowa`. Tym samym kolejne daty mogą być dynamicznie wyznaczone jako suma daty początkowej i numeru sekwencyjnego krotki.
- e) Dodaj do diagramu transformacji komponent *Add sequence* z katalogu *Transform* i połącz go z komponentem *Wygeneruj wiersze*.



- f) Przejdź do edycji nowo dodanego elementu.
- Ustaw nazwę elementu na *Ponumeruj wiersze*.
  - Określ nazwę nowego atrybutu wiersza z wartością sekwencji na `numer_dnia_od_poczatku`.

Step name	Ponumeruj wiersze
Name of value	numer_dnia_od_poczatku

- Numerowanie powinno zaczynać się od 0. Ustaw w tym celu wartość parametru *Start at value* na równą 0.
- g) Pozostaw bez zmian wartości dla *Increment by* oraz *Maximum Value*.

Use a transformation counter to generate the sequence	
Use counter to calculate sequence?	<input checked="" type="checkbox"/>
Counter name (optional)	
Start at value	0
Increment by	1
Maximum value	99999999

- Zatwierdź zmiany używając przycisku *OK*.
- h) Z menu kontekstowego dla komponentu *Ponumeruj wiersze* wybierz opcję *Preview*.
- Wprowadź liczbę 5 jako maksymalną liczbę wierszy podglądu.

Number of rows to retrieve	5
<input checked="" type="checkbox"/> Retrieve first rows (preview)	

- Wybierz przycisk *Quick Launch* i przeglądaj wyniki. Zgodnie z oczekiwaniami data początkowa jest stała, natomiast zwiększa się wartość atrybutu `numer_dnia_od_poczatku`.

Rows of step: Ponumeruj wiersze (5 rows)		
	data_początkowa	numer_dnia_od_poczatku
1	2005/01/01	0
2	2005/01/01	1
3	2005/01/01	2
4	2005/01/01	3
5	2005/01/01	4

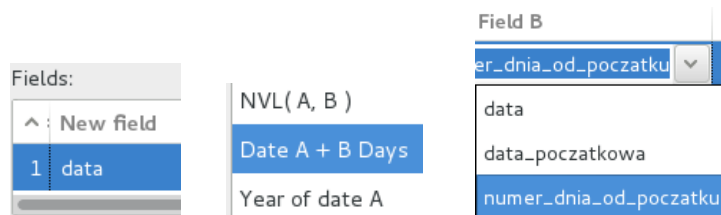
- Zamknij podgląd używając przycisku *Stop*.
- i) W kolejnym kroku obliczymy właściwą datę dla wiersza oraz wyznaczmy niektóre z elementów daty, np. numer miesiąca. Dodaj do diagramu transformacji komponent *Calculator* z katalogu *Transform* i

połącz go z komponentem Ponumeruj wiersze.



j) Przejdź do edycji nowo dodanego komponentu.

- Wprowadź nazwę **Oblicz podstawowe elementy daty**.
- Zdefiniuj elementy do obliczenia. W kolumnie **New field** należy podać nazwę atrybutu (np. data). Następnie w kolumnie **Calculation** z listy wybieramy interesujące nas działanie (np. **Date A + B Days**). Parametry do działania definiujemy w kolumnach **Field A** i **Field B**. Dla opisywanego przykładu dla obliczania daty dla **Field A** należy przypisać atrybut `data_początkowa` natomiast dla **Field B** atrybut `numer_dnia_od_początku`. Jako typ wartości (**Value type**) należy wskazać **Date**.



- Zdefiniuj kolejno atrybuty: `numer_miesiaca`, `rok`, `numer_dnia_tygodnia` i `numer_dnia_miesiaca` zgodnie z przedstawionym poniżej zrzutem ekranu. Nie zapomnij o określeniu typu danych (kolumna **Value type**). Zauważ, że tylko jeden z wykorzystanych elementów obliczeniowych wymaga podania dwóch parametrów. Pozostałe wymagają wyspecyfikowania jednego atrybutu. Zauważ również, że taki atrybut może być jednym z tych, które wcześniej zostały utworzone (w tym samym komponentcie). W naszym przykładzie pierwszym obliczonym elementem jest `data`, która stanowi podstawę do wyliczeń kolejnych elementów.

Fields:					
^ : New field	Calculation	Field A	Field B	Field C	Value type
1 data	Date A + B Days	data_początkowa	numer_dnia_od_początku		Date
2 numer_miesiaca	Month of date A	data			Integer
3 rok	Year of date A	data			Integer
4 numer_dnia_tygodnia	Day of week of date	data			Integer
5 numer_dnia_miesiaca	Day of month of date	data			Integer

- Zatwierdź zmiany przyciskiem **OK**.
- Z menu kontekstowego komponentu **Oblicz podstawowe elementy daty** wybierz polecenie **Preview....** Określi liczbę krotek podglądu na 5. Wybierz przycisk **Quick Launch**.

Examine preview data							
Rows of step: Oblicz podstawowe elementy daty (5 rows)							
^ :	data_początkowa	numer_dnia_od_początku	data	numer_miesiaca	rok	numer_dnia_tygodnia	numer_dnia_miesiaca
1	2005/01/01		0 2005/01/01 00:00:00.000	1	2005	7	1
2	2005/01/01		1 2005/01/02 00:00:00.000	1	2005	1	2
3	2005/01/01		2 2005/01/03 00:00:00.000	1	2005	2	3
4	2005/01/01		3 2005/01/04 00:00:00.000	1	2005	3	4
5	2005/01/01		4 2005/01/05 00:00:00.000	1	2005	4	5

- Zweryfikuj poprawność wyliczonych elementów. Zamknij okno używając przycisku **Stop**.

k) Pomimo bogactwa opcji komponentu **Calculate** nie wszystkie elementy daty mogą zostać wyznaczone przy jego użyciu. Rozwiązaniem jest wykorzystanie języka **JavaScript**.

l) Dodaj komponent *Modified Java Script Value* z katalogu *Scripting*. Połącz go z komponentem Wyznacz podstawowe element daty i przejdź do edycji nowo dodanego komponentu.

- Ustaw nazwę na Wyznacz pozostałe elementy.

- Wprowadź następujące polecenia języka *JavaScript* w panelu *Script1*:

```
var locale = new java.util.Locale("pl", "pl");
var dateFormat = java.text.SimpleDateFormat("EEEE", locale);
var nazwa_dnia = dateFormat.format(data);
dateFormat.applyPattern("MMMM");
var nazwa_miesiaca = dateFormat.format(data);

var ktory_kwartal = "1";
switch (parseInt(numer_miesiaca))
{
    case 4: case 5: case 6: ktory_kwartal = "2"; break;
    case 7: case 8: case 9: ktory_kwartal = "3"; break;
    case 10: case 11: case 12: ktory_kwartal = "4"; break;
}

var temp = rok.toString();
temp += numer_miesiaca < 10 ? "0" : "";
temp += numer_miesiaca.toString();
temp += numer_dnia_miesiaca < 10 ? "0" : "";
temp += numer_dnia_miesiaca.toString();
var klucz_podstawowy = parseInt(temp);
```

- Powyższy kod w języku *JavaScript* służy do wyznaczenia nazwy miesiąca, nazwy dnia oraz wyliczenia, na które kwartały roku przypadają przetwarzane daty. Dodatkowo wyliczony jest klucz podstawowy na podstawie numeru roku, miesiąca i dnia.

- Ostatnim elementem definicji komponentu *Modified Java Script Value* jest wskazanie, które wartości zmiennych mają zostać zwrócone na zewnątrz komponentu (i tym samym przesłane do kolejnych kroków transformacji). Przejdź do sekcji *Fields* i wprowadź definicje atrybutów: *nazwa\_dnia*, *nazwa\_miesiaca*, *ktory\_kwartal*, *klucz\_podstawowy* zgodnie z poniższą tabelką. Zwróć uwagę na typ (kolumna *Type*) atrybutów. Możesz pomóc sobie przyciskiem *Get variables*.

Fields					
Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1 nazwa_dnia		String			N
2 nazwa_miesiaca		String			N
3 ktory_kwartal		String			N
4 klucz_podstawowy		Integer			N

- Zakończ edycję komponentu wybierając przycisk *OK*.

m) Ostatnim krokiem transformacji będzie zapisanie wygenerowanych wierszy do relacji w hurtowni danych. W tym celu wykorzystamy komponent *Table Output*.

n) Dodaj komponent *Table Output* z katalogu *Output*. Połącz go z komponentem Wyznacz pozostałe elementy (wybierz opcję *Main output step*).

o) Przejdź do edycji nowo dodanego komponentu.

- Ustaw nazwę na Zapisz dane.

- Wskaż połączenie dwh oraz relację DATY.

- Zaznacz opcję *Specify database fields*.
- Przejdź do zakładki *Database fields*.

ignore insert errors   
Specify database fields   
ns Database fields

- Wykorzystaj przycisk *Enter field mapping* i przejdź do powiązania wygenerowanych atrybutów z atrybutami relacji w bazie danych. Wynik powinien wyglądać następująco:

```
Mappings:  
klucz_podstawowy --> DA_DATA_ID  
nazwa_miesiaca --> DA_NAZWA_MIESIACA  
numer_dnia_miesiaca --> DA_NUMER_DNIA_MIESIACA  
ktory_kwartal --> DA_KWARTAL  
nazwa_dnia --> DA_NAZWA_DNIA  
numer_dnia_tygodnia --> DA_NUMER_DNIA_TYGODNIA  
rok --> DA_ROK  
numer_miesiaca --> DA_NUMER_MIESIACA  
data --> DA_PELNA_DATA
```

- Zatwierdź zdefiniowane powiązania wybierając przycisk *OK*.
- Tabela w zakładce *Database fields* zostanie wypełniona automatycznie.

Fields to insert:		
	Table field	Stream field
1	DA_DATA_ID	klucz_podstawowy
2	DA_NAZWA_MIESIACA	nazwa_miesiaca
3	DA_NUMER_DNIA_MIESIACA	numer_dnia_miesiaca
4	DA_KWARTAL	ktory_kwartal
5	DA_NAZWA_DNIA	nazwa_dnia
6	DA_NUMER_DNIA_TYGODNIA	numer_dnia_tygodnia
7	DA_ROK	rok
8	DA_NUMER_MIESIACA	numer_miesiaca
9	DA_PELNA_DATA	data

- Zakończ edycję komponentu wybierając przycisk *OK*.
- p) Nasza transformacja powinna teraz wyglądać następująco



- Zapisz transformację w repozytorium (*File->Save as...*).
  - Jako nazwę transformacji podaj GENEROWANIE WYMIARU DATY.
  - Wskaż folder *wypożyczalnie/shop->dwh*.
- q) Uruchom transformację korzystając wybierając z głównego menu *Action->Run*. Wybierz przycisk *Run*. Uzyskany wynik powinien być identyczny z poniższym:

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active
1 Wygeneruj wiersze	0	0	2000	0	0	0	0	0	Finished
2 Ponumeruj wiersze	0	2000	2000	0	0	0	0	0	Finished
3 Oblicz podstawowe elementy daty	0	2000	2000	0	0	0	0	0	Finished
4 Wyznacz pozostałe elementy	0	2000	2000	0	0	0	0	0	Finished
5 Zapisz dane	0	2000	2000	0	2000	0	0	0	Finished

### III. Zasilanie tabeli faktów

Ostatnim etapem zasilania hurtowni danych jest zasilanie relacji faktów. W tym celu konieczne będzie nie tylko wstawienie wierszy do tabeli faktów, ale także określenie w nich wartości odpowiednich kluczy wskazujących na relacje wymiarów. Część kluczy jest zgodna z tymi które są obecne w źródłach (np. dla klientów i pracowników). Jednak filmy posiadają w hurtowni klucze zastępcze, które są generowane w czasie zasilania wymiaru. Dla dat występuje natomiast klucz, który można wyliczyć na podstawie elementów daty.

#### 1. Ekstrakcja danych o faktach.

Tak jak w poprzednich transformacjach dane będą pobierane z dwóch źródeł `shop1` i `shop2`. Tym razem zdefiniujemy nieco bardziej skomplikowane polecenia SQL do odczytu danych o wypożyczeniach.

a) Dodaj nową transformację. Umieść na niej komponent *Table Input* z katalogu *Input* i przejdź do jego edycji.

- Ustaw nazwę komponentu na `Okresl znacznik czasowy`.
- Wskaż bazę danych `dwh`.
- Wprowadź następujące polecenie SQL:

```
SELECT COALESCE( MAX(wy_ostatnia_modyfikacja) ,
                TIMESTAMP '2000-01-01 00:00:00.000') as
maks_data_f_wypozyczenia
FROM WYPOZYCZENIA
```

The screenshot shows the 'Table input' configuration window. The 'Step name' is 'Okresl znacznik czasowy'. The 'Connection' is 'dwh'. The 'SQL' field contains the query: `SELECT COALESCE( MAX(wy_ostatnia_modyfikacja), TIMESTAMP '2000-01-01 00:00:00.000') as maks_data_f_wypozyczenia FROM WYPOZYCZENIA`. The 'Limit size' is 0. The 'OK' button is highlighted.

- Zatwierdź zmiany wybierając przycisk *OK*.

b) Dodaj następujący komponent *Table Input*. Połącz go z komponentem Okresl znacznik czasowy. Przejdź do edycji komponentu.

- Ustaw nazwę na Wczytaj wypożyczenia z shop1.
- Wskaż połączenie shop1.
- Wykorzystaj następujące polecenie SQL:

```
SELECT
    w.wypozyczenie_id
  , w.data_wypozyczenia
  , w.data_zwrotu
  , w.klient_id
  , w.pracownik_id
  , w.ostatnia_modyfikacja
  , f.tytul
  , f.data_wydania
  , l as sklep_id
  , o.wartosc
FROM oplaty o RIGHT JOIN wypozyczenia w ON w.wypozyczenie_id =
o.wypozyczenie_id
JOIN magazyn m on w.egzemplarz_id = m.egzemplarz_id JOIN filmy
f on f.film_id = m.film_id
where w.ostatnia_modyfikacja > ?
```

- Zwróć uwagę na fakt wykorzystania połączenia wielu tabel. Celem jest uzyskanie tytułu i daty wydania filmu, którego egzemplarz (z magazynu) został wypożyczony klientowi . Dodatkowo, podobnie jak w transformacji klientów, w klauzuli SELECT wprowadzony został identyfikator sklepu, z którego pochodzi krotki.
- Pamiętaj o ustawieniu źródła danych dla parametru zapytania.

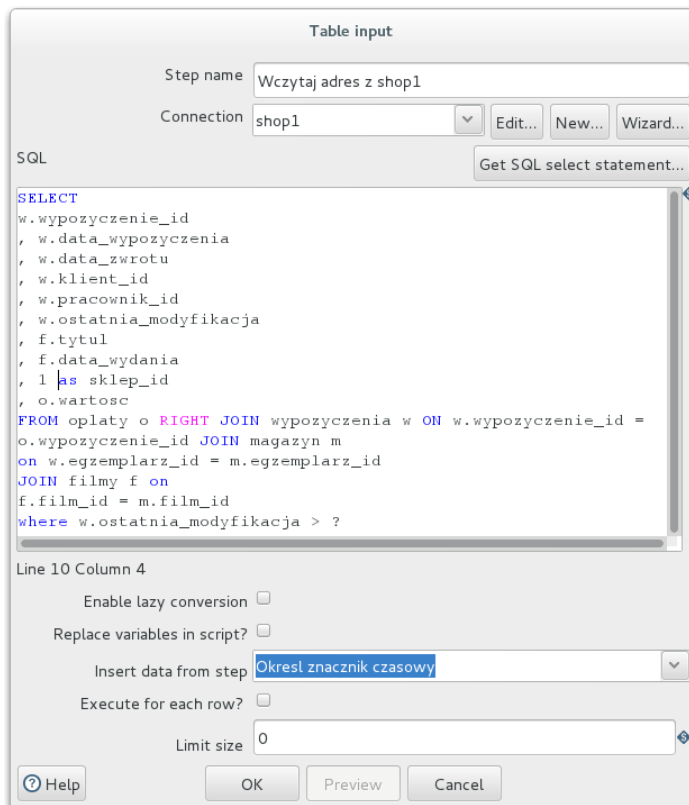


Table input

Step name: Wczytaj adres z shop1

Connection: shop1

SQL

```
SELECT
w.wypozyczenie_id
, w.data_wypozyczenia
, w.data_zwrotu
, w.klient_id
, w.pracownik_id
, w.ostatnia_modyfikacja
, f.tytul
, f.data_wydania
, l as sklep_id
, o.wartosc
FROM oplaty o RIGHT JOIN wypozyczenia w ON w.wypozyczenie_id =
o.wypozyczenie_id
JOIN magazyn m on w.egzemplarz_id = m.egzemplarz_id JOIN filmy
f on f.film_id = m.film_id
where w.ostatnia_modyfikacja > ?
```

Line 10 Column 4

Enable lazy conversion

Replace variables in script?

Insert data from step: Okresl znacznik czasowy

Execute for each row?

Limit size: 0

Buttons: Help, OK, Preview, Cancel

- Zapisz zmiany wybierając przycisk **OK**.
- a. Wykonaj kopię komponentu Wczytaj wypożyczenia z shop1. Możesz w tym celu

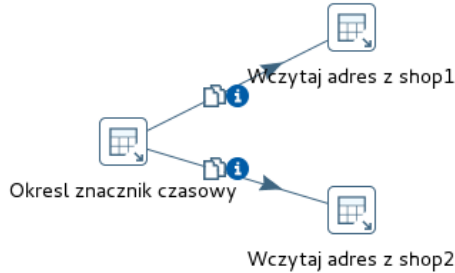
wykorzystać polecenie *Duplicate step* z menu kontekstowego komponentu.

- Zmień jego nazwę na *Wczytaj wypożyczenia z shop2*.
- Zmień źródłową bazę danych na *shop2* oraz identyfikator sklepu w klauzuli *SELECT*.

```
, f.data_wydania
, 2 as sklep_id
, o.wartosc
```

- Połącz komponent *Wczytaj wypożyczenia z shop2* z komponentem *Okresl znacznik czasowy* (użyj metody *COPY* dla przekazywania krotek).

**Uwaga:** tutaj wkradł się błąd na obrazkach – tym i kolejnych. *Wczytaj wypożyczenia z shop1* jest błędnie pokazane jako *Wczytaj adres z shop1*. Podobnie dla *shop2*.

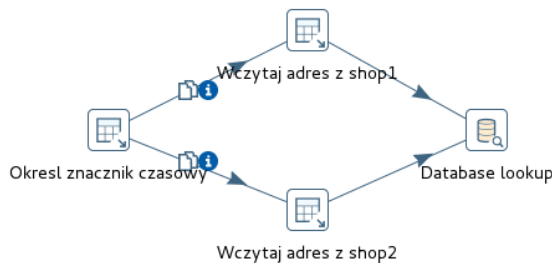


## 2. Ustalenie klucza dla filmu

W tym kroku zajmiemy się wyznaczeniem klucza wskazującego na wypożyczony film. Klucz ten zostanie odczytany z wymiaru *FILMY*.

- Dodaj komponent *Database lookup* z katalogu *Lookup*.

- Połącz go z komponentami odczytującymi dane ze źródłowych baz danych w sposób jaki został pokazany na rysunku poniżej.



- Przejdź do edycji nowo dodanego komponentu.
- Ustaw nazwę komponentu na *Okresl id filmu*. Komponent powinien odczytać dane z relacji *FILMY* w bazie danych *dwh*.
- Zdefiniuj warunek jaki musi spełniać poszukiwany wpis w relacji *FILMY* (unikalne są pary tytuł i rok w którym wydano film).

The key(s) to look up the value(s):

^	Table field	Comparator	Field1	Field2
1	FI_TYTUL	=	tytuł	
2	FI_ROK_WYDANIA	=	data_wydania	

- Wskaż atrybut *FI\_FILM\_ID* jako poszukiwaną wartość. Ustaw typ na *Integer*.

Values to return from the lookup table :

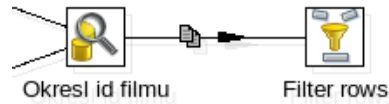
^	Field	New name	Default	Type
1	FI_FILM_ID			Integer

- Zatwierdź wprowadzone zmiany przyciskiem *OK*.

## 3. Kolejnym krokiem będzie wyznaczenie kluczy dla atrybutów zawierających daty. Dodatkowo wyliczone

zostaną miary.

- a) W danych o wypożyczeniach mogą istnieć wpisy dotyczące filmów, które nie zostały jeszcze oddane a zatem mają nieokreślony atrybut `data_zwrotu`. Przetwarzanie takich wypożyczeń musi zatem uwzględniać taką sytuację. Dodaj do diagramu transformacji component *Filter rows* z katalogu *Flow* i połącz go z componentem Okresl id filmu (użyj opcji *Main output of step*).



- b) Przejdź do edycji nowo dodanego componentu.

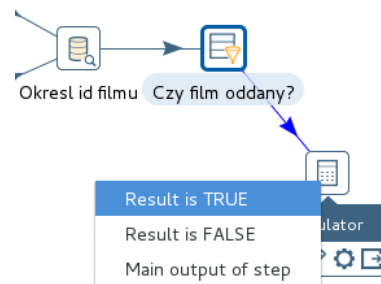
- Ustaw jego nazwę na `Czy film oddany?`.
- Wprowadź warunek sprawdzający czy data zwrotu filmu (atrybut `data_zwrotu`) jest określona.

The condition:

<input type="text"/>		<input type="text"/>
<code>data_zwrotu</code>	IS NOT NULL	<input type="text"/>
		<input type="text"/>

- Zatwierdź zmiany przyciskiem *OK*.

- c) Dodaj component *Calculator* z katalogu *Transform*. Połącz go z componentem `Czy film oddany?` wskazując opcję obsługi krotek dla których warunek filtrowania jest spełniony (są to zatem krotki, które mają ustawioną datę zwrotu).



- Przejdź do edycji nowo dodanego componentu i ustaw jego nazwę na `Oblicz dni wypożyczenia`.
- Dodaj dwa atrybuty: `czy_oddany`, którego wartość 1 będzie wskazywać na to, że film został oddany oraz `liczba_dni_wypożyczenia` przechowujący informację ile pełnych dni trwało wypożyczenie.

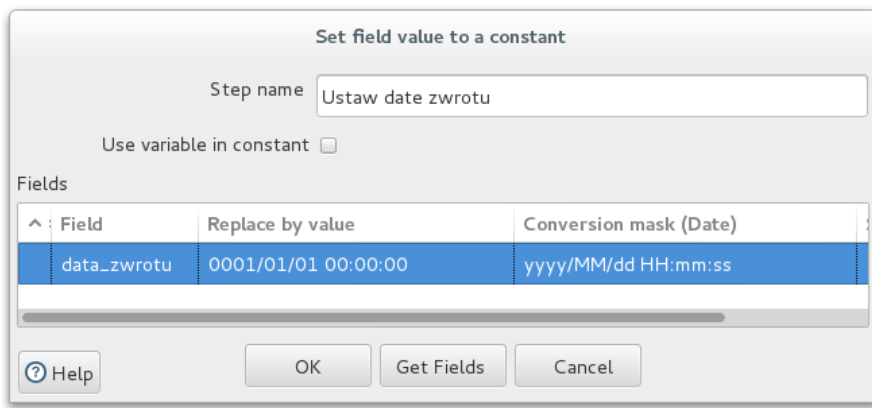
Fields:						
^	New field	Calculation	Field A	Field B	Field C	Value type
1	<code>czy_oddany</code>	Set field to constant value A	1			Integer
2	<code>liczba_dni_wypożyczenia</code>	Date A - Date B (in days)	<code>data_zwrotu</code>	<code>data_wypożyczenia</code>		Integer

- Zatwierdź zmiany przyciskiem *OK*.

- d) Zajmijmy się teraz krotkami reprezentującymi wypożyczenia, dla których nie zwrócono filmów. Dodaj component *Set field value to a constant* z katalogu *Transform*. Jego zadaniem będzie ustawienie brakującej daty zwrotu na wybraną przez nas wartość. Niech to będzie `0001/01/01 00:00:00`.

- Ustaw nazwę componentu na `Ustaw date zwrotu`.
- Ustaw wartość pola `data_zwrotu` zgodnie z podanym niżej przykładem.



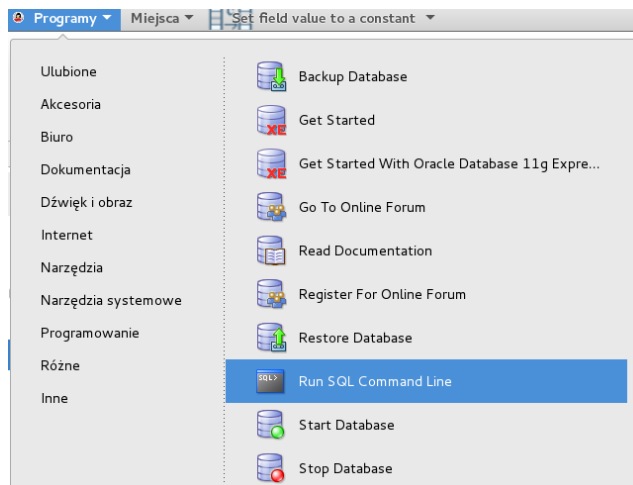


- Zamknij okno edycji ustawień dla komponentu.
- W tym miejscu musimy jeszcze przypomnieć sobie jakie daty i klucze znajdują się w wymiarze daty. Wygenerowaliśmy za pomocą odpowiedniej transformacji 2000 wpisów dla kolejnych dni począwszy od 01.01.2005. Nie ma jednak w relacji DATY wpisu reprezentującego specjalną datę 01.01.0001. Wykonaj następujące polecenie w bazie danych dwh:

```
insert into dwh.daty(DA_DATA_ID, DA_NAZWA_MIESIACA,
DA_NUMER_DNIA_MIESIACA, DA_ROK,
DA_NUMER_MIESIACA, DA_NUMER_DNIA_TYGODNIA, DA_PELNA_DATA,
DA_KWARTAL, DA_NAZWA_DNIA)
values(10101, ' ',0,0,0,0,DATE '0001-01-01',0,' ');

commit;
```

- Możesz do tego celu wykorzystać narzędzie *SQLPlus* dostępne w menu systemowym *Programy->Inne->Run SQL Command Line*.



```
Use "connect username/password@XE" to connect to the database.
SQL> connect dwh@xe/dwh
Connected.
SQL> insert into dwh.daty(DA_DATA_ID, DA_NAZWA_MIESIACA, DA_NUMER_DNIA_MIESIACA,
DA_ROK, DA_NUMER_MIESIACA, DA_NUMER_DNIA_TYGODNIA, DA_PELNA_DATA, DA_KWARTAL, DA
_NAZWA_DNIA) values(10101, ' ',0,0,0,0,DATE '0001-01-01',0,' ');

1 row created.

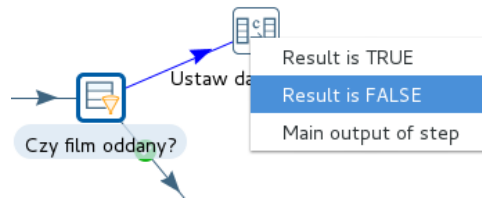
SQL> █
```

```
SQL> commit;

Commit complete.

SQL> █
```

- Możesz zamknąć program *SQLPlus* wykonując w nim polecenie `exit`;
- Połącz komponenty *Czy film dodany?* i *Ustaw date zwrotu* wybierając opcję *Result is FALSE*.



- W porównaniu z krotkami posiadającymi datę zwrotu w krotkach bez daty zwrotu brakuje jeszcze dwóch atrybutów: `czy_oddany` oraz `liczba_dni_wypozyczenia`. Dodaj komponent *Add constants* z katalogu *Transform*.
- Przejdź do edycji komponentu, ustaw jego nazwę na *Uzupelnij dane* i wprowadź omawiane atrybuty z przypisanymi wartościami równymi 0.

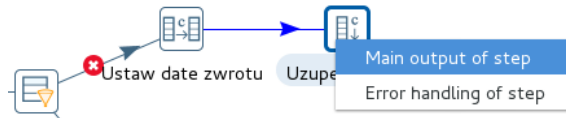
Add constant values

Step name:

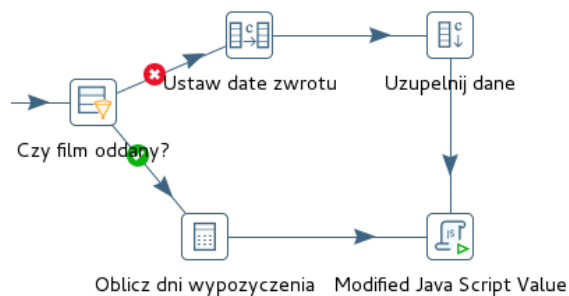
Fields:

^	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty str
1	czy_oddany	Integer							0	
2	liczba_dni_wypozyczenia	Integer							0	

- Zatwierdź zmiany w komponencie i połącz go z komponentem *Ustaw date zwrotu*.



- e) Dla wszystkich dat należy obliczyć ich klucze wskazujące na odpowiedni wiersz w relacji *DATY*. Podobnie jak przy wypełnianiu tego wymiaru, również tym razem posłużymy się komponentem *Modified Java Script Value* z katalogu *Scripting*. Dodaj ten komponent do diagramu transformacji i połącz go w pokazany niżej sposób z komponentami *Uzupelnij dane* oraz *Oblicz dni wypozyczenia*.



- Przejdź do edycji komponentu i ustaw jego nazwę na *Okresl id dla dat*.
- Wprowadź następujący kod JavaScript:

```
var locale = new java.util.Locale("pl", "pl");
var dateFormat =
java.text.SimpleDateFormat("yyyyMMdd", locale);
var data_wypozyczenia_klucz =
parseInt(dateFormat.format(data_wypozyczenia).toString(), 10);
var data_zwrotu_klucz =
parseInt(dateFormat.format(data_zwrotu).toString(), 10);
```

- Pamiętaj również o wyspecyfikowaniu atrybutów `data_wypozyczenia_klucz` i `data_zwrotu_klucz` zwracanych przez skrypt.

Fields						
Fieldname	Rename to	Type	Length	Precision	Replace value	
1 data_wypozyczenia_klucz		Integer	10	0	N	
2 data_zwrotu_klucz		Integer	10	0	N	

- Zatwierdź zmiany przyciskiem **OK**.

#### 4. Zapisanie krotek do relacji faktów.

a) Ostatnim krokiem transformacji jest zapisanie wyliczonych wartości do bazy danych. W tym celu użyjemy komponentu **Table Output** z katalogu **Output**. Połącz komponent z komponentem **Okresl id dla dat**. Przejdź do edycji nowo dodanego komponentu.

- Ustaw nazwę na `Zapisz fakty`.
- Wskaż połączenie `dwh` oraz relację `WYPOZYCZENIA`.
- Zaznacz opcję **Specify database fields**.
- Przejdź do zakładki **Database fields**.

Specify database fields

Database fields

- Wykorzystaj przycisk **Enter field mapping** i przejdź do powiązania wygenerowanych atrybutów z atrybutami relacji w bazie danych. Wynik powinien wyglądać następująco:

Mappings:

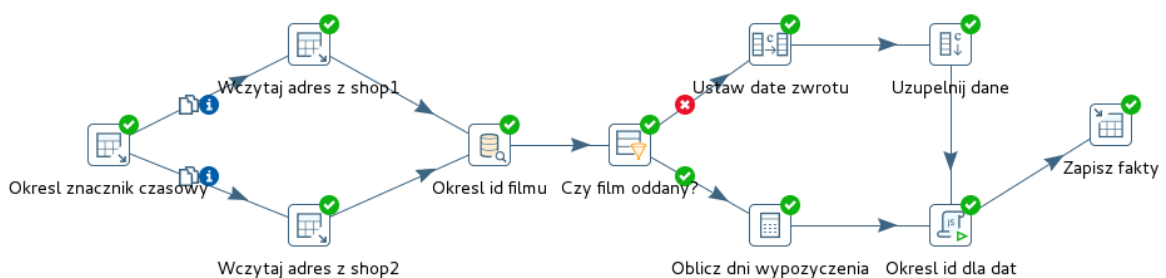
```

czy_oddany --> WY_CZY_ODDANY
data_wypozyczenia_klucz --> WY_DATA_WYPOZYCZENIA_ID
data_zwrotu_klucz --> WY_DATA_ZWROTU_ID
FI_FILM_ID --> WY_FILM_ID
klient_id --> WY_KLIENT_ID
liczba_dni_wypozyczenia --> WY_OKRES_WYPOZYCZENIA
wartosc --> WY_OPLATA
ostatnia_modyfikacja --> WY_ostatnia_modyfikacja
pracownik_id --> WY_PRACOWNIK_ID
sklep_id --> WY_SKLEP_ID
wypozyczenie_id --> WY_WYPOZYCZENIE_ID
    
```

- Zatwierdź zdefiniowane powiązania wybierając przycisk **OK**.
- Zakończ edycję komponentu wybierając przycisk **OK**.

#### 5. Zapisz transformację pod nazwą `ZALADUJ FAKTY` w katalogu `wypożyczalnie/shop->dwh`.

#### 6. Uruchom transformację. Wynik powinien wyglądać następująco:





# Zasilanie i Odświeżanie Hurtowni Danych - część IV

© Paweł Boiński, Krzysztof Jankiewicz - Instytut Informatyki, Politechnika Poznańska

[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 adres z shop2	0	1	1771	1771	0	0	0	0	Finished
3 Wczytaj adres z shop1	0	1	1696	1696	0	0	0	0	Finished
4 Okresl id filmu	0	3467	3467	3467	0	0	0	0	Finished
5 Czy film oddany?	0	3467	3467	0	0	0	0	0	Finished
6 Ustaw date zwrotu	0	0	0	0	0	0	0	0	Finished
7 Oblicz dni wypozyczenia	0	3467	3467	0	0	0	0	0	Finished
8 Uzupelnij dane	0	0	0	0	0	0	0	0	Finished
9 Okresl id dla dat	0	3467	3467	0	0	0	0	0	Finished
10 Zapisz fakty	0	3467	3467	0	3467	0	0	0	Finished