Zaawansowane Technologie Baz Danych

laboratorium

Paweł Boiński
Politechnika Poznańska, Instytut Informatyki

Część I

WPROWADZENIE

Literatura

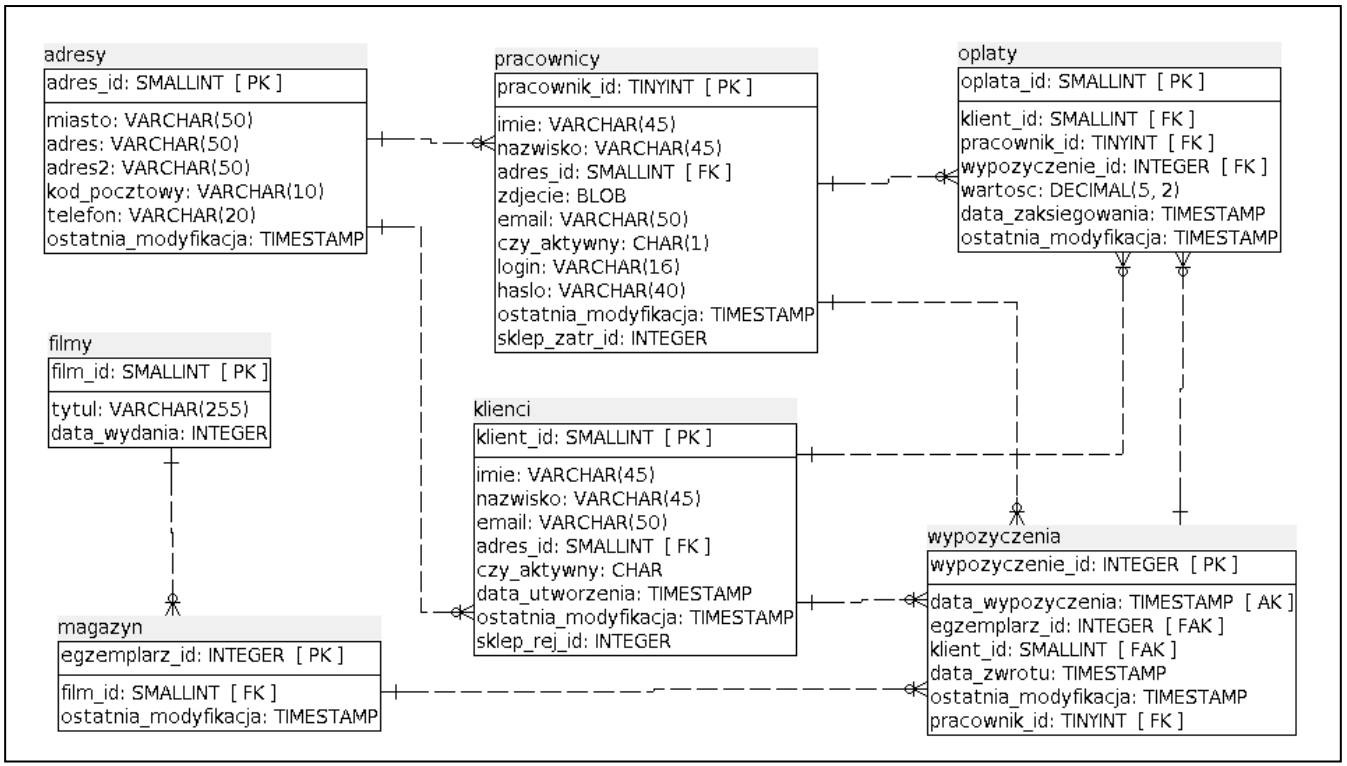
- **Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration**
  - Matt Casters, Roland Bouman, Jos van Dongen
  - 2010, ISBN-10: 0470635177
- **Pentaho Data Integration 4 Cookbook**
  - Pulvirenti Adrián Sergio, Roldán María Carina
  - 2011, ISBN-10: 1849515247
- **Learning Pentaho Data Integration 8 CE - Third Edition: An end-to-end guide to exploring, transforming, and integrating your data across multiple sources**
  - Roldán María Carina
  - 2017

Studium przypadku

- Rozwijająca się sieć wypożyczalni filmów
  - Dwa sklepy
  - Wdrożony system informatyczny
  - Dwie bazy danych (różne systemy zarządzania bazami danych)
- Cel: przeprowadzenie analizy danych o wypożyczeniach w celu opracowania nowej strategii marketingowej
- Źródło danych: Sakila DB

http://dev.mysql.com/doc/sakila/en/

Źródła danych (1)



Sklep nr 1 – SZBD MySQL, sklep nr 2 – SZBD PostgreSQL

Źródła danych (2)

Plik **CSV** - dane adresowe i geograficzne z serwisu geonames.org

```
US      34050   FPO
US      34034   APO
US      99553   Akutan  Alaska
US      99571   Cold Bay
US      99583   False Pass
US      99612   King Cove
US      99661   Sand Point
US      99546   Adak    Alaska
US      99547   Atka    Alaska
```

Plik **XML** - dane na temat filmów

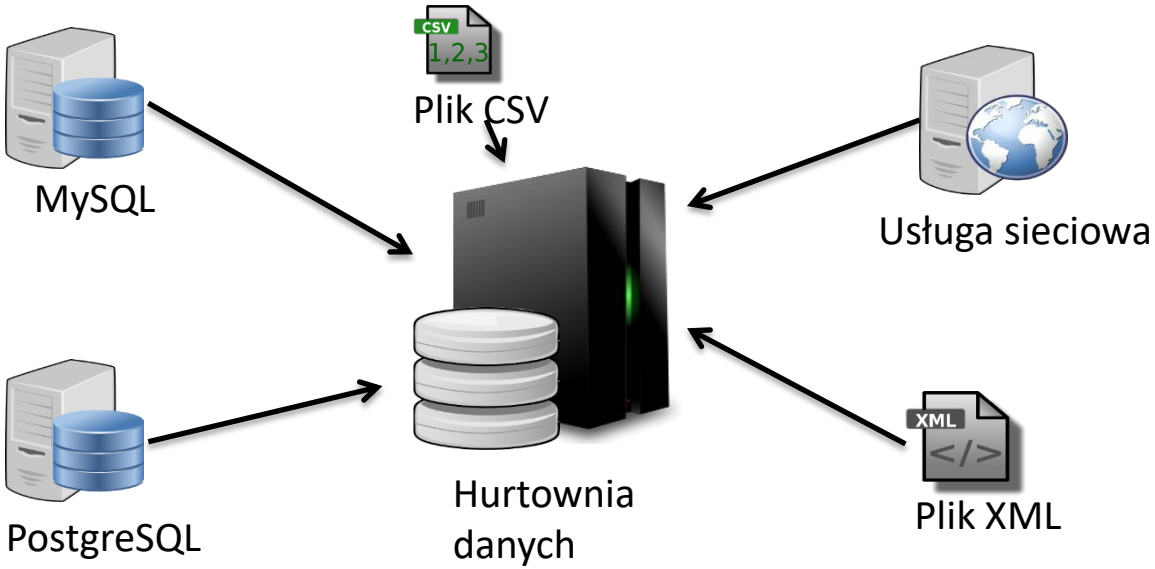
```
−<FILMY>
 −<FILM id="1">
    <TYTUL>ACADEMY DINOS
   −<OPIS>
      A Epic Drama of a Feminis
    </OPIS>
    <DATA_PRODUKCJI>2006
    <JEZYK>English</JEZYK>
```

Usługa sieciowa - dane na temat aktorów występujących we filmach

```
{"row":
[{"ACTOR_ID":23,"FIRST_NAME":"SANDRA"
{"ACTOR_ID":37,"FIRST_NAME":"VAL","LA
{"ACTOR_ID":124,"FIRST_NAME":"SCARLET
{"ACTOR_ID":155,"FIRST_NAME":"IAN","L
{"ACTOR_ID":198,"FIRST_NAME":"MARY","
```

Założenia

- Dane pracowników i klientów są replikowane
- Dane pracowników są uaktualniane tylko w sklepie, w którym jest zatrudniona dana osoba
- Dane klientów mogą być uaktualniane w dowolnym sklepie
- Filmy identyfikowane są przez nazwę i rok wydania (produkcji)



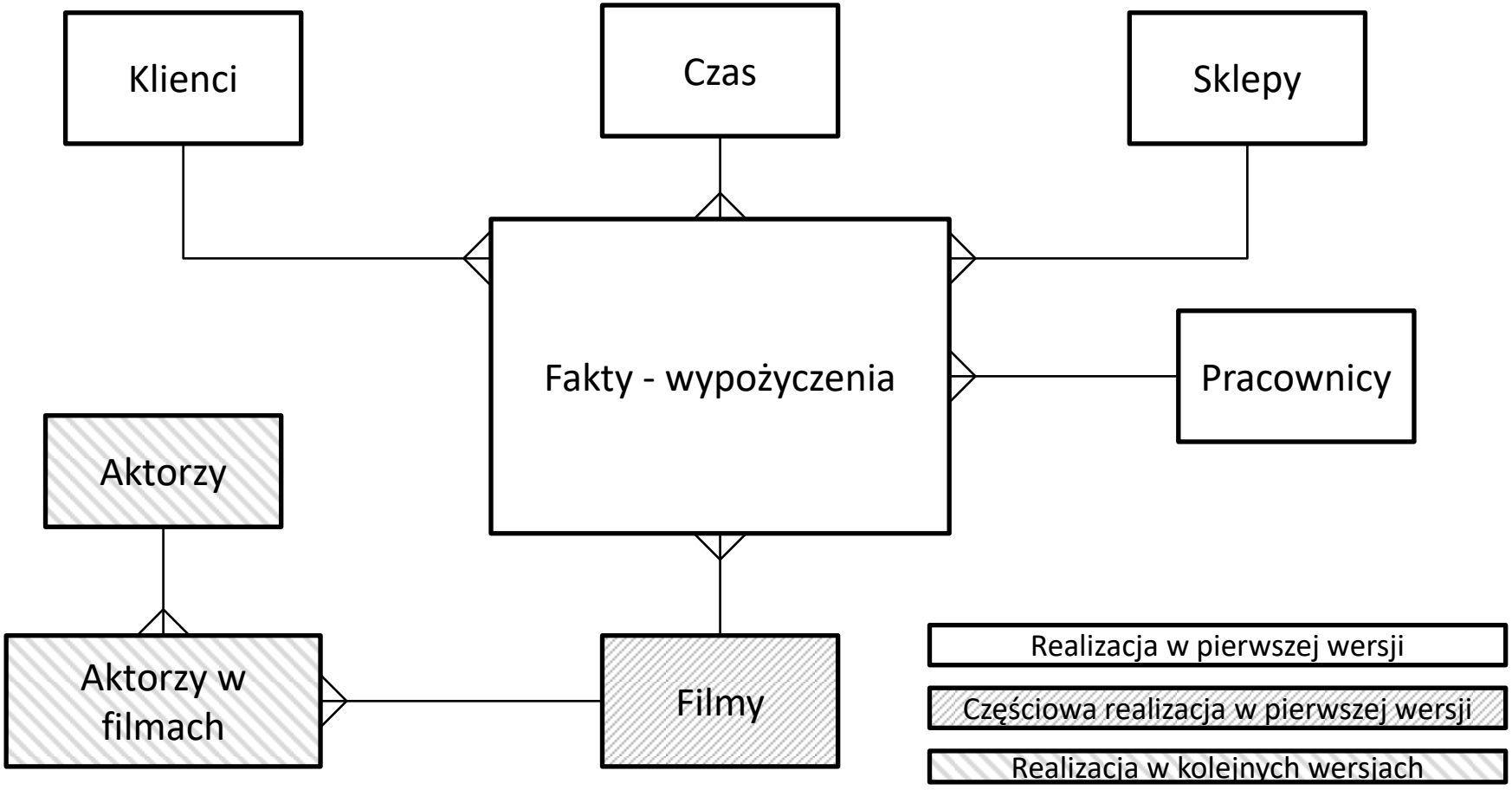
ETL/ELT

- **ETL** (Extract, Transform, Load)
  - Ogólna definicja: proces lub zbiór procesów zasilania hurtowni danych danymi ze źródeł
- **Ekstrakcja** – przyłączenie się i pobranie danych ze źródeł, w taki sposób, że możliwe jest ich dalsze przetwarzanie
- **Transformacja** – Zbiór operacji wykonywanych na pobranych ze źródeł danych np. sprawdzanie poprawności, konwersja, obliczanie agregatów
- **Ładowanie** – Wstawianie danych do hurtowni danych. Obejmuje takie elementy jak zarządzanie kluczami, utrzymywanie historii dla wymiarów etc.
- Inne podejście: ELT

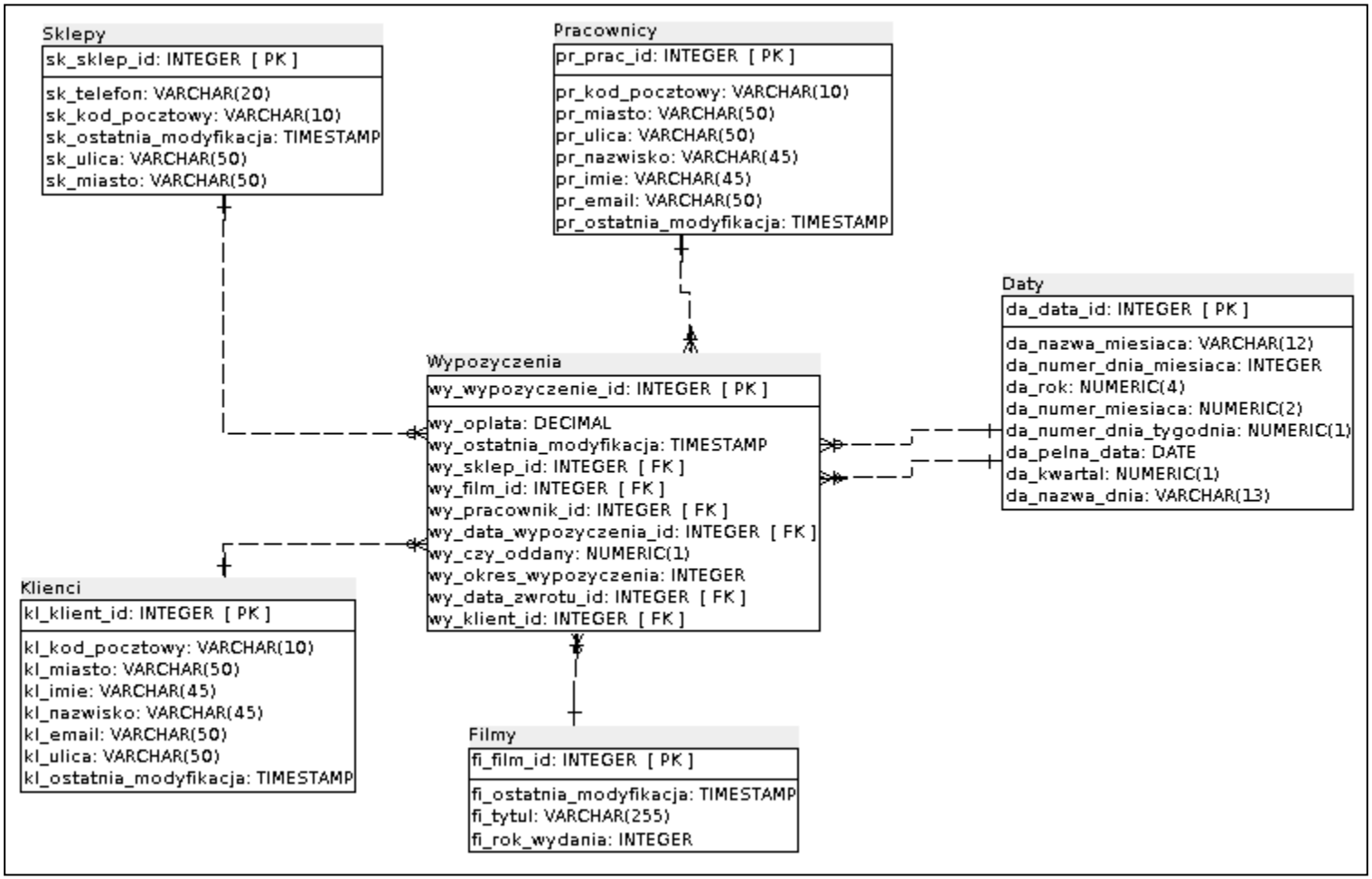
Agile Business Intelligence

- Rozwijanie i utrzymywanie procesów ETL można porównać do rozwijania i utrzymywania oprogramowania
- Zyskują na popularności tzw. zwinne metodyki
- Przykładem jest metodyka **Agile BI**, której celem jest szybka adaptacja organizacji do zmieniających się warunków biznesowych
- Wybrane cechy Agile BI:
  - Realizacja przyrostowa
  - Szybkie tworzenie podstawowych elementów oprogramowania (procesów ETL)
  - Możliwość łatwego i szybkiego wprowadzania zmian
- Środowisko do budowy procesów ETL powinno wspierać metodykę Agile BI

Hurtownia danych – ogólny schemat

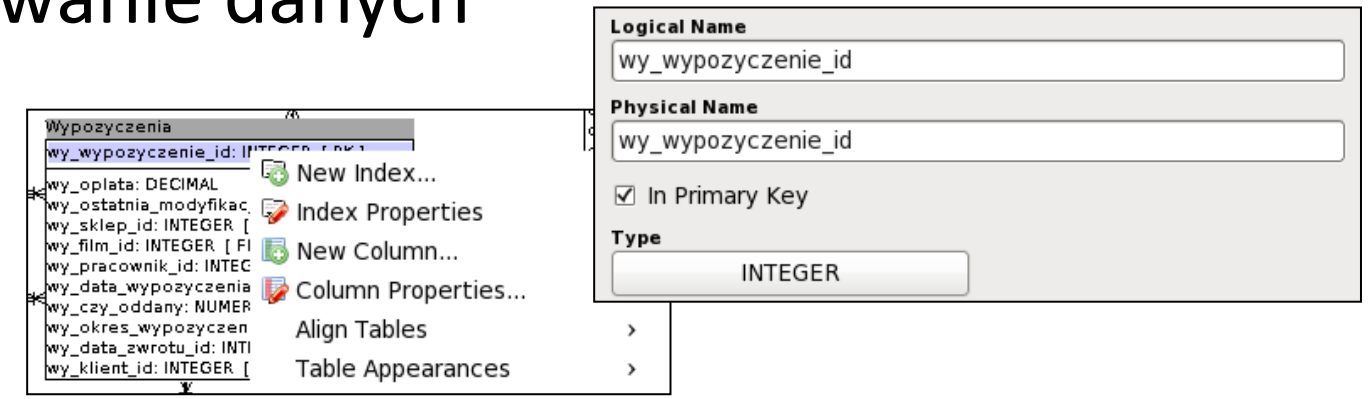


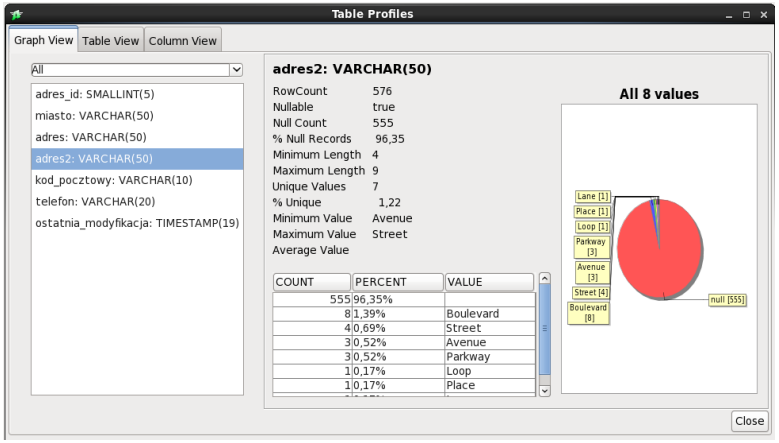
Hurtownia danych – pierwsza wersja



Power Architect

- Modelowanie danych





- Profilowanie danych

Część II

PENTAHO DATA INTEGRATION
PROSTA TRANSFORMACJA
TRANSFORMACJA PODRZĘDNA

Pentaho Data Integration

- *Kettle* – wersja „community edition”
  - K
  - E (*Extraction*)
  - T (*Transformation*)
  - T (*Transportation*)
  - L (*Loading*)
  - E
- Implementacja w języku Java
  - Windows, Unix/Linux
- Podstawowe składniki Kettle:
  - *Spoon* – graficzny interfejs użytkownika
  - *Kitchen* – program odpowiedzialny za wykonywanie zaprojektowanych procesów ETL (zadań)
  - *Pan* – program odpowiedzialny za wykonywanie zaprojektowanych transformacji

Puk puk.
- Kto tam?
… (mijają 3s)
- Java.

Puk puk.
- Kto tam?
- C++.

Puk puk.
- Assembler.

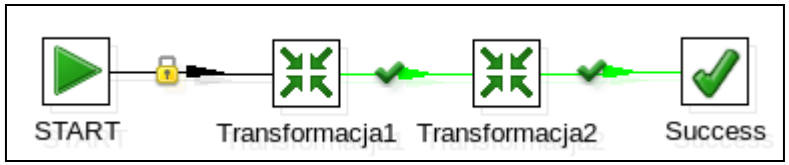
Pojęcia podstawowe

- Transformacja (*transformation*)
  - Szeroko rozumiane wykonywanie działań na wierszach danych
  - Może składać się z wielu kroków (*steps*) np. odczyt, filtrowanie
  - Kroki transformacji są wykonywane równolegle*
- Zadanie(*job*)
  - Składa się z wielu transformacji oraz innych elementów
  - Wykonanie sekwencyjne*
- Połączenie (*hop*)
  - Reprezentuje skierowany przepływ danych pomiędzy krokami transformacji lub elementami zadania
  - Ma ograniczony rozmiar bufora dla danych (FIFO)

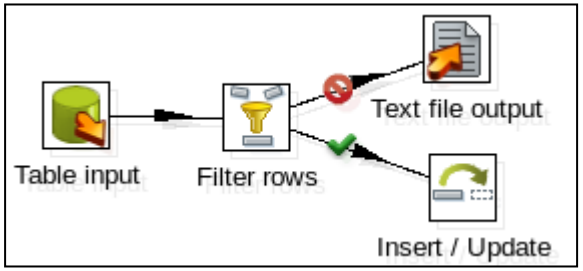
* w większości przypadków

Pojęcia podstawowe

- Krok (*step*)
  - Nazwany, elementarny składnik transformacji
  - Może być połączony z innymi krokami poprzez przepływy wchodzące i wychodzące
  - Nie ma kroku początkowego, wszystkie kroki wykonywane są równolegle
  - Krotki ze wszystkich przepływów wchodzących muszą mieć taką samą strukturę



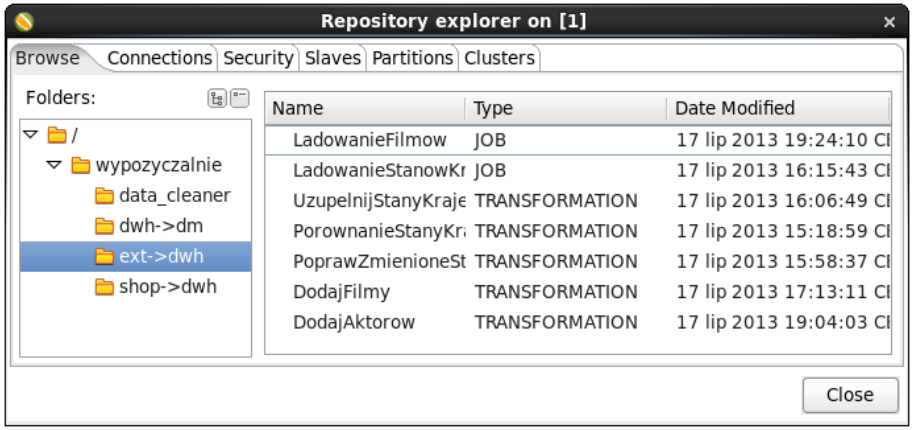
Przykład zadania



Przykład transformacji

Repozytorium

- Zawiera wszystkie elementy związane z procesem transformacji: zadania, transformacje, połączenia etc.
- Rodzaje repozytorium
  - Serwer Pentaho – utrzymanie repozytorium przez dedykowany system
  - Bezpośrednio w bazie danych – łatwe współdzielenie, bezpieczeństwo zapewniane przez SZBD
  - Plikowe – wykorzystuje *Virtual File System* - repozytorium można umieścić zarówno w katalogu jak i w pliku zip czy też na zdalnym serwerze (np. FTP).
  - Zadania i transformacje można również zapisać poza repozytorium w postaci plików XML (*.kjb dla zadań i *.ktr dla transformacji)

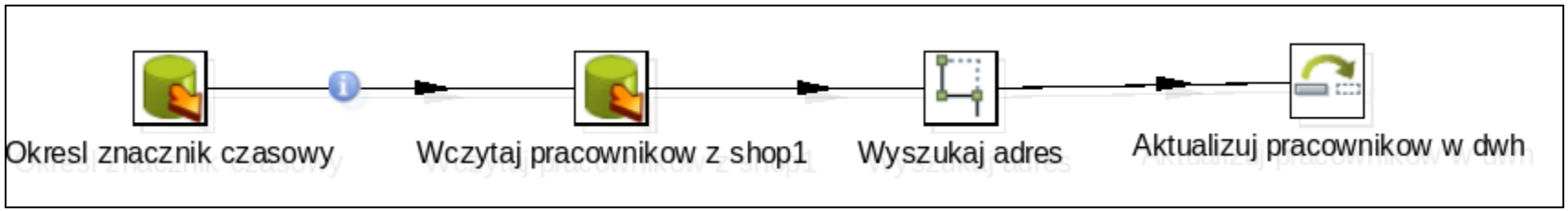


Agile BI w Pentaho DI

- Narzędzie *Pentaho DI* wspomaga metodykę *Agile BI*:
  - Łatwość instalacji
  - Minimalna liczba wymaganych parametrów dla definiowanych zadań (np. dla połączeń nie trzeba specyfikować nazwy klasy sterownika JDBC)
  - Predefiniowane najpopularniejsze, parametryzowane kroki transformacji
  - Prawie cała funkcjonalność dostępna z poziomu interfejsu graficznego
  - Minimalizacja liczby wyświetlanych informacji (np. brak mapowań dla każdego pola)
  - Całkowicie dowolne nazewnictwo dla komponentów wykorzystywanych w transformacjach (poprawia czytelność)
  - Możliwość podglądu przetwarzanych danych

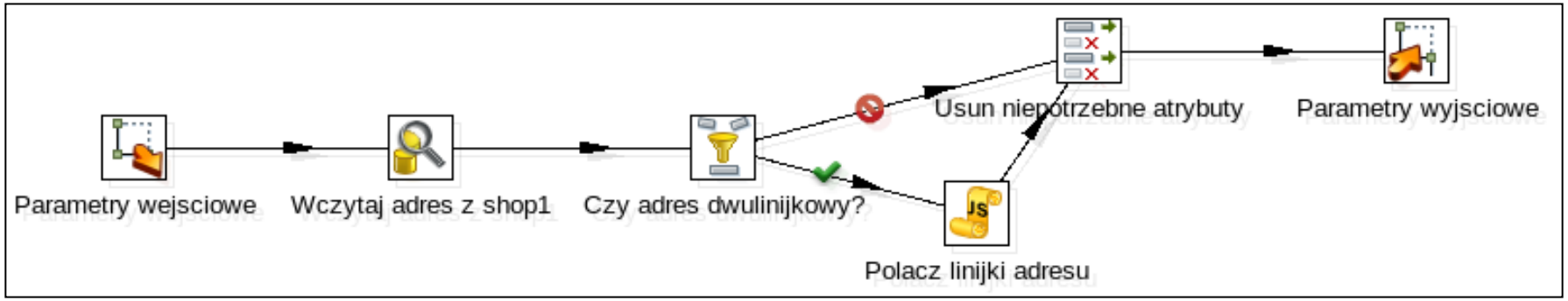
Transformacja pracowników z jednego źródła danych

- Odczytuje tylko dane, które uległy zmianie od ostatniego uruchomienia
- Odczytuje dane tylko z jednego sklepu
- Dla każdego pracownika określa adres jego zamieszkania wykorzystując transformację podrzędną
- Wprowadza zmiany w wymiarze PRACOWNICY w hurtowni danych



Transformacja podrzędna

- Wywoływana z innej transformacji
- Może pobierać wartości parametrów (identyfikatory pracowników)
- Może zwracać wynik w postaci strumienia krotek (dane adresowe pracowników)
- Odczytuje dane adresowe z bazy danych
- Zamienia dwukolumnowe adresy na jeden atrybut



Część III

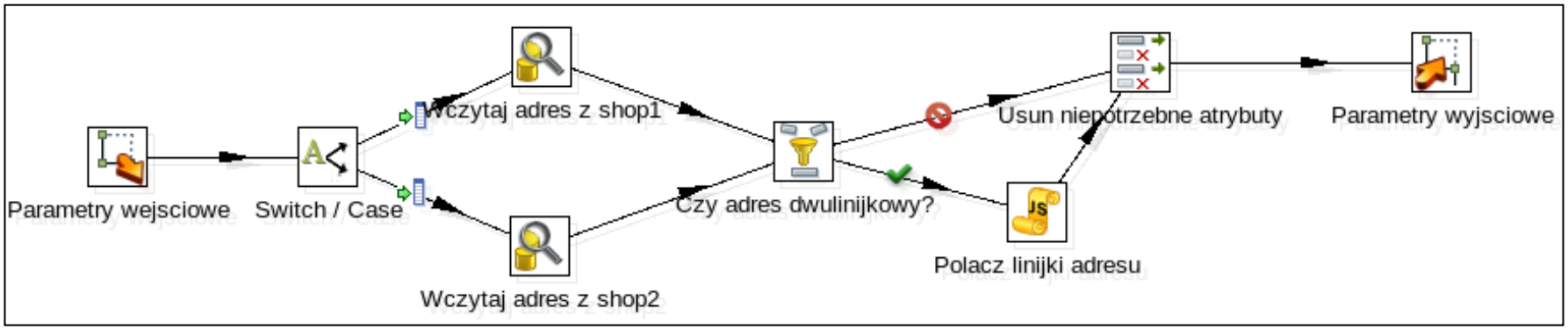
DRUGIE ŹRÓDŁO DANYCH TRANSFORMACJA KLIENTÓW

Drugie źródło danych

- Pierwsza wersja transformacji pracowników zakłada odczyt tylko z jednego źródła danych
- Odczyt drugiego źródła (sklepu nr 2) wygląda analogicznie, konieczne jest jednak zduplikowanie znacznika czasowego wskazującego, którzy pracownicy mają być transformowani

Modyfikacja transformacji podrzędnej

- Pierwotnie wyszukiwany adres tylko ze sklepu nr 1
- Dodanie parametru – numeru sklepu – do transformacji podrzędnej
- Na podstawie wartości tego parametru decyzja, z którego sklepu odczytać dane adresowe
- Krotki z dwóch źródeł są łączone w jeden strumień a następnie przetwarzane w taki sam sposób jak poprzednio



Transformacja klientów - założenia

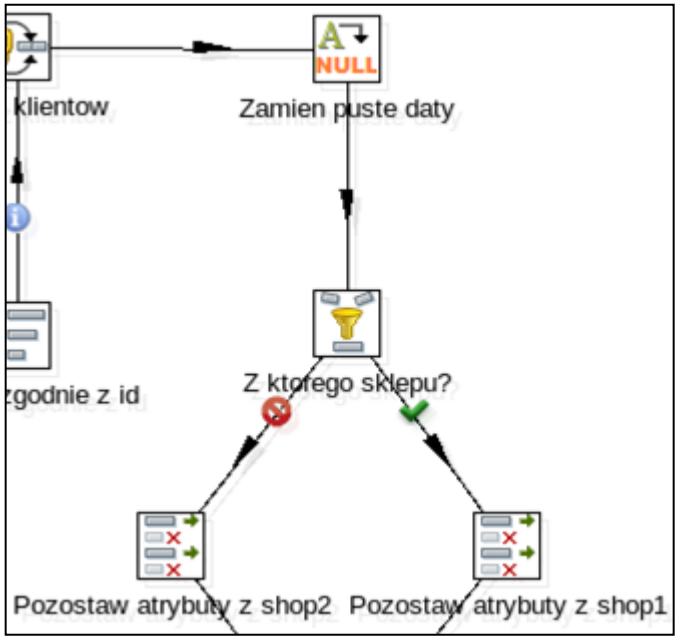
- Klienci są replikowani pomiędzy bazami danych
- Ten sam klient może mieć różne dane w różnych sklepach ale ma zawsze ten sam identyfikator
- Podczas modyfikacji/wstawiania klientów uaktualniany/wstawiany jest znacznik czasowy
- Jeżeli dane klienta są tylko w jednym źródle (replikacja odbywa się w określonych odstępach czasu) to są one wstawiane do hurtowni danych
- Jeżeli dane klienta są w dwóch źródłach danych to wybierane są te dane klienta, które oznaczone są późniejszym znacznikiem czasowym

Łączenie klientów

- Komponent **Merge Join** działa jak operacja *sort merge* w bazie danych
- Dane wejściowe muszą być posortowane (w źródle lub poza nim) zgodnie z warunkiem połączeniowym
- Ponieważ nie wszystkie dane klientów muszą być zreplikowane konieczne jest wykorzystanie połączenia zewnętrznego (FULL JOIN) aby zachować krotki z obu źródeł
- W krotkach, dla których zabrakło połączenia parametry z drugiego źródła pozostają puste

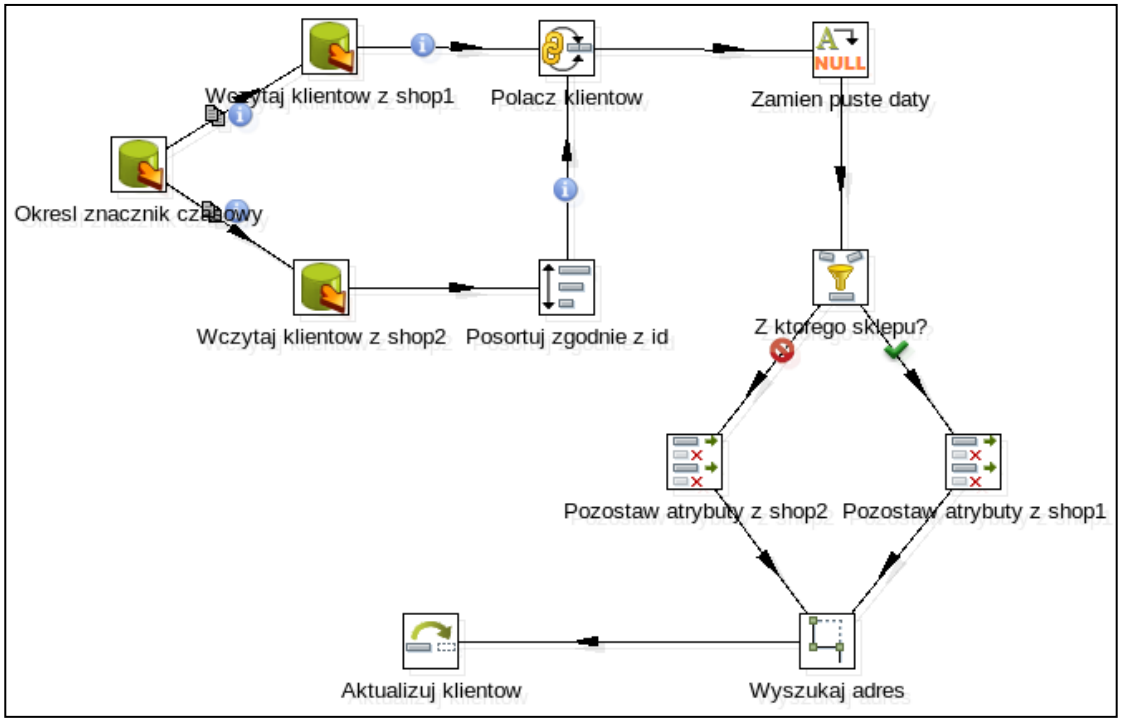
Wybór źródła danych

- Określenie, w którym źródle dane są bardziej aktualne na podstawie znaczników czasowych
- Dla danych z połączenia zewnętrznego trzeba zamienić puste daty na daty, które muszą być starsze niż te w aktualnym źródle.
- Wynik porównania znaczników czasowych wyznacza ścieżkę, w której nastąpi wybór danych z właściwego źródła



Pełna transformacja klientów

- Ostatnim elementem jest pobranie adresu klienta (transformacja podrzędna) i aktualizacja danych klienta w hurtowni danych

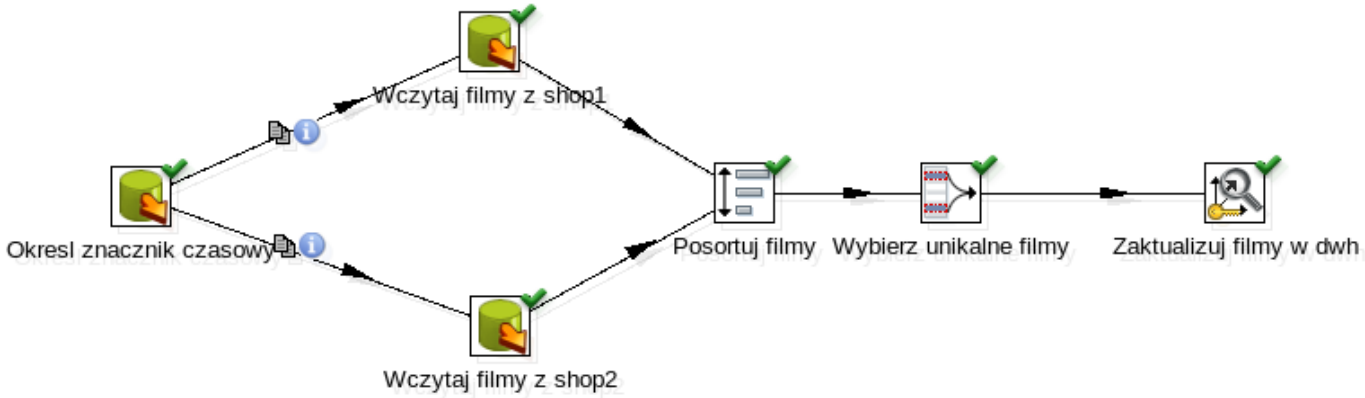


Część IV

GENEROWANIE DANYCH
ZASILANIE RELACJI FAKTÓW

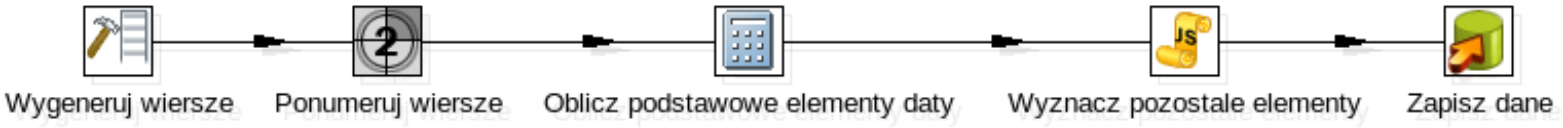
Transformacja filmów

- Dane o filmach mogą być zduplikowane
- Komponent *Unique rows* eliminuje duplikaty ale wymaga posortowanego zbioru krotek
- Nowe filmy (lub zmiany) są zapisywane w wymiarze FILMY.



Generowanie danych

- Wymiar przechowujący informacje o datach może być wstępnie wypełniony danymi
- Komponent *Generate rows* generuje określoną liczbę wierszy natomiast komponent *Add sequence* może zostać wykorzystany do ich ponumerowania
- Na podstawie numeru wiersza, komponenty *Calculate* i *Modified Java Script Value* wyznaczają konkretne dane (datę, nazwę dnia, miesiąca etc.) dla przetwarzanej krotki



Zasilanie relacji faktów

- Przed zapisaniem nowej krotki w relacji faktów konieczne jest
  - wyznaczenie identyfikatora filmu
  - wyliczenie czasu wypożyczenia dla oddanych filmów
  - wyznaczenie identyfikatorów dat (wypożyczenia i opcjonalnie zwrotu)