

Grafowy model bazy danych na przykładzie GOOD

(Graph-Oriented Object Database Model)

Marcin Jakubek

Plan prezentacji

- ➔ Przykłady modeli danych
- ➔ Model grafowy GOOD
- ➔ Zastosowania
- ➔ Inne modele grafowe
- ➔ Wizualizacja grafów
- ➔ Podsumowanie

Model danych:

1. Struktura danych
2. Operacje na danych
3. Więzy integralności

Przykłady modeli danych

- Hierarchiczny
 1. rekordy, związki nadrzędny-podrzędny
 2. proceduralne języki zapytań, nawigacja po rekordach
 3. powiązania nadrzędny-podrzędny
- Sieciowy
 1. rekordy, kolekcje (właściciel + członkowie)
 2. j.w.
 3. wymagania dot. członkostwa

Przykłady modeli danych

➤ Relacyjny

1. relacje - zbiory krotek
2. operacje algebry relacyjnej: selekcja, rzut, złączenie...
3. unikalność krotek, klucze...

➤ Obiektowy

1. abstrakcyjne typy danych i relacje między nimi - obiekty, hierarchia klas, metody, dziedziczenie...
2. operacje poprzez metody obiektów
3. definiowanie atrybutów, integralność klasa-klasa, klasa-obiekt...

Przykłady modeli danych

- Półstrukturalny, XML
- Przestrzenny (*Spatial*)


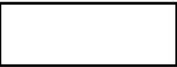
Model grafowy GOOD

GOOD - Graph-Oriented Object Database Model
M. Gyssens, J. Paredaens, J. Van den Bussche,
D. Van Gucht (1990)

Model grafowy GOOD

Struktura danych

Skierowany graf o dwóch typach wierzchołków:

- ➔ „atomowych” (*printable*), reprezentujących dane liczbowe, stringi, daty, ale także bitmapy oraz pliki dźwiękowe, etykietowanych typem danych, ozn.: 
- ➔ „nieatomowych” (*non-printable*), reprezentujących obiekty abstrakcyjne, etykietowanych klasą obiektu, ozn.: 

Model grafowy GOOD

Struktura danych

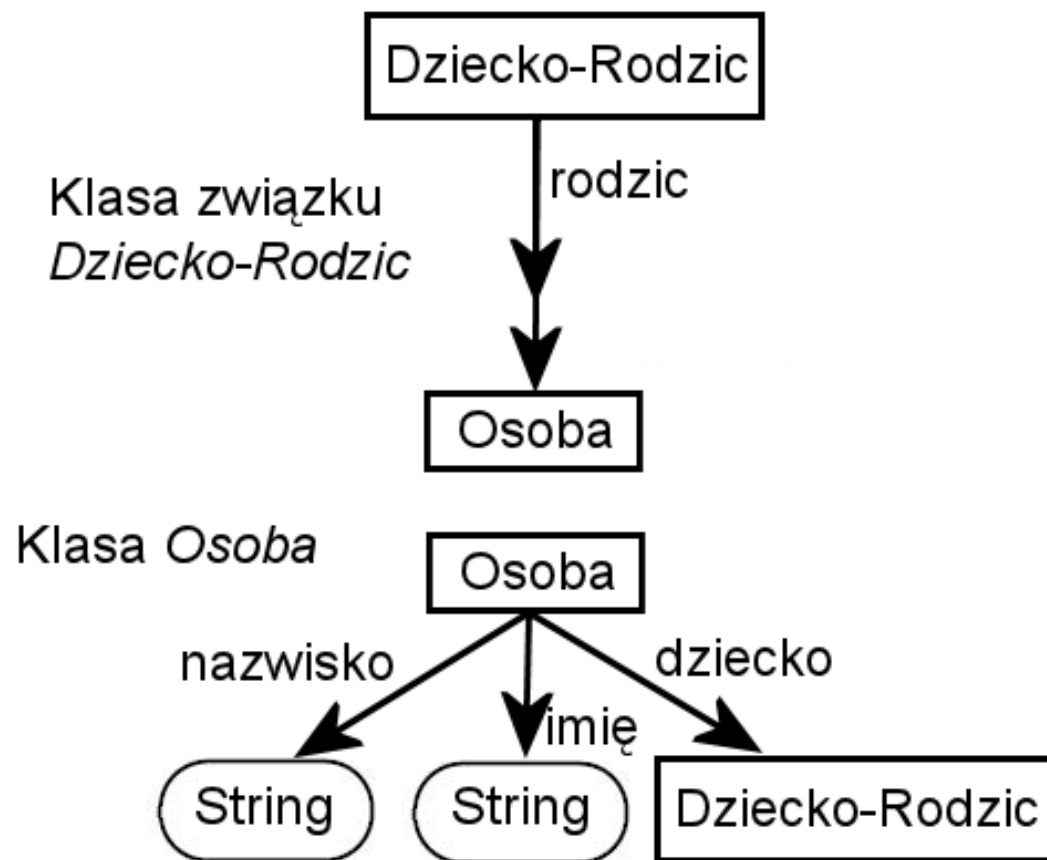
oraz dwóch typach krawędzi, etykietowanych nazwą związku lub atrybutu:

- reprezentujących związki funkcyjne,
ozn.: \longrightarrow
- reprezentujących związki jeden-do-wielu,
ozn: $\longrightarrow\!\!\!\rightarrow$

Model grafowy GOOD

Struktura danych

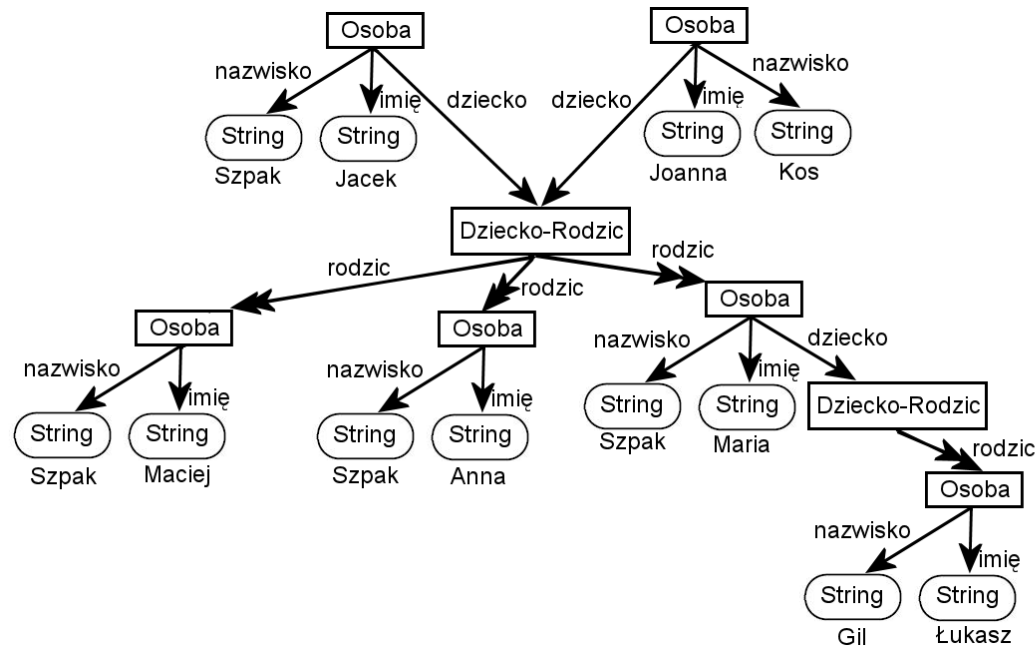
Przykładowy schemat bazy danych:



Model grafowy GOOD

Struktura danych

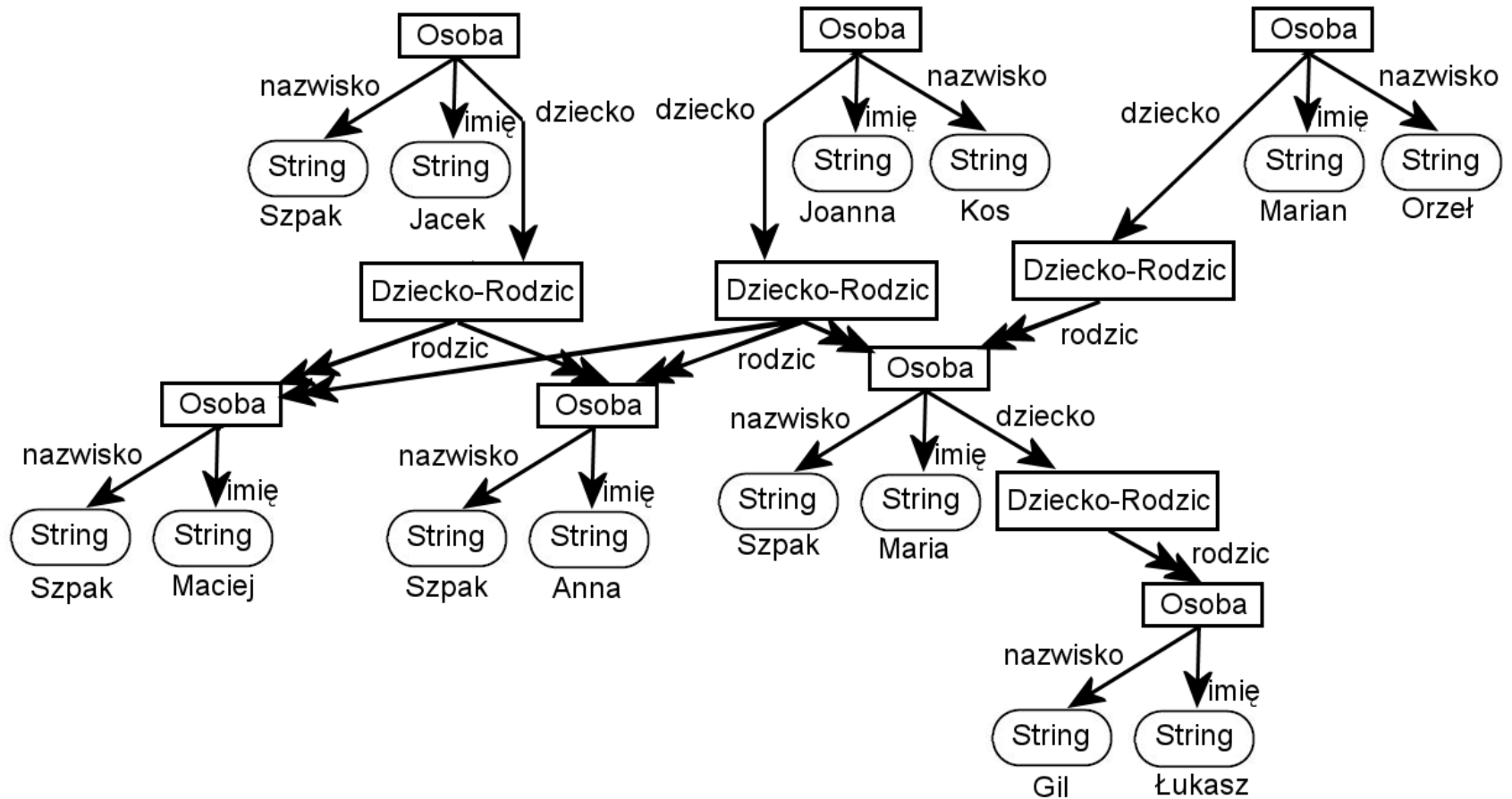
Przykładowy egzemplarz bazy danych:



Model grafowy GOOD

Struktura danych

Przykładowy egzemplarz bazy danych:

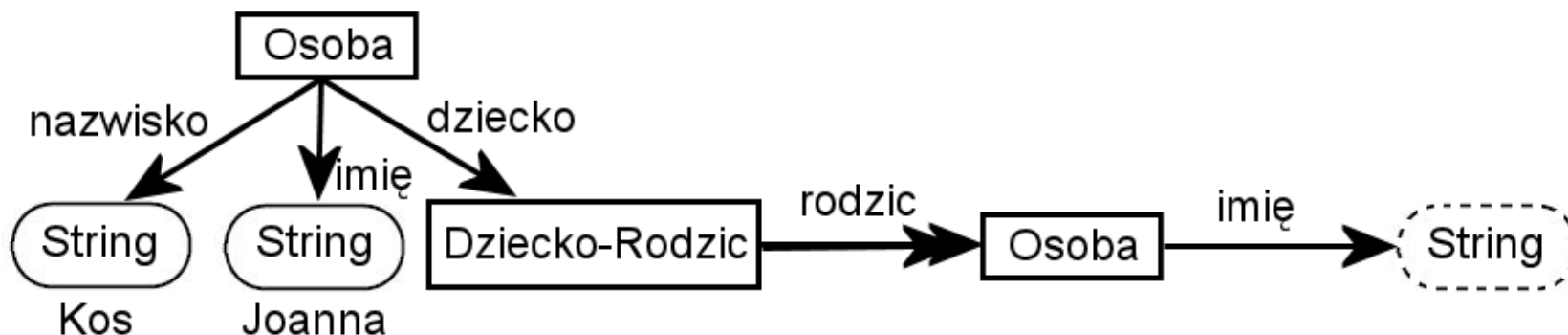


Model grafowy GOOD

Operacje na danych

„Grafowe” operacje na danych - dopasowywanie wzorca (*pattern matching*) podgrafu.

Przykład: zapytanie o imiona dzieci Joanny Kos:



Model grafowy GOOD

Operacje na danych

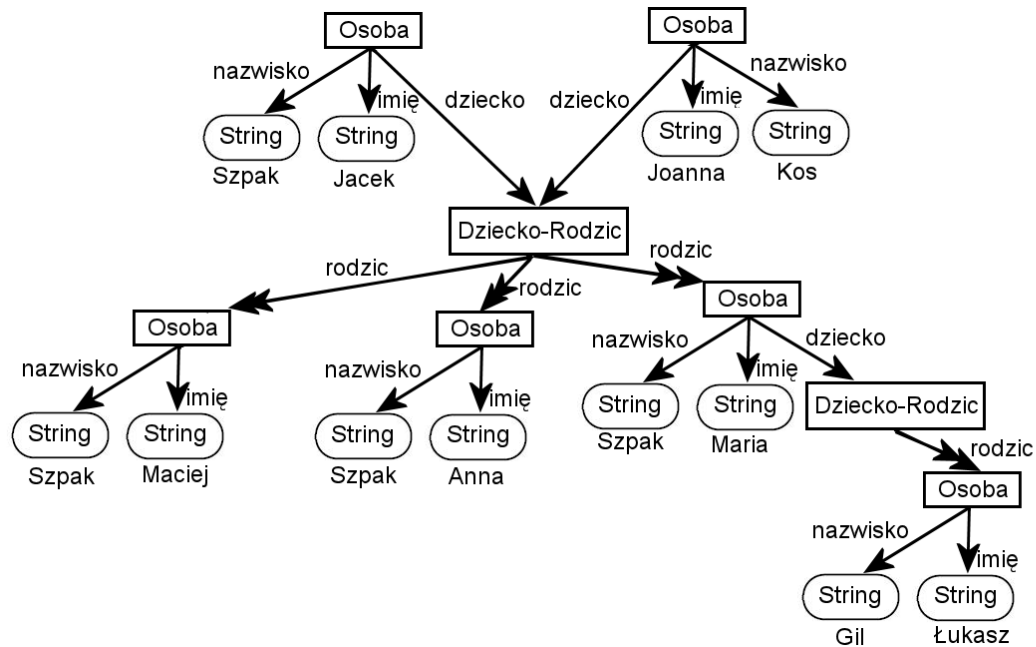
Modyfikacje danych oparte o *pattern matching*:

- ➔ Dodawanie i kasowanie wierzchołków
- ➔ Dodawanie i kasowanie krawędzi
- ➔ Abstrakcja

Model grafowy GOOD

Operacje na danych

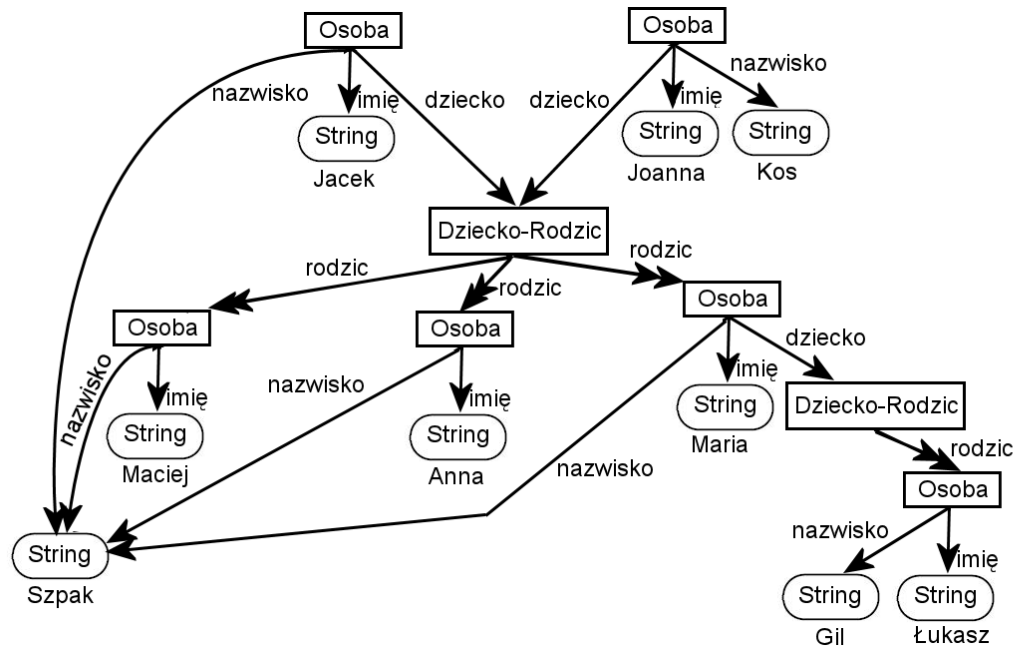
Abstrakcja - grupowanie duplikatów



Model grafowy GOOD

Operacje na danych

Abstrakcja - grupowanie duplikatów



Model grafowy GOOD

Więzy integralności

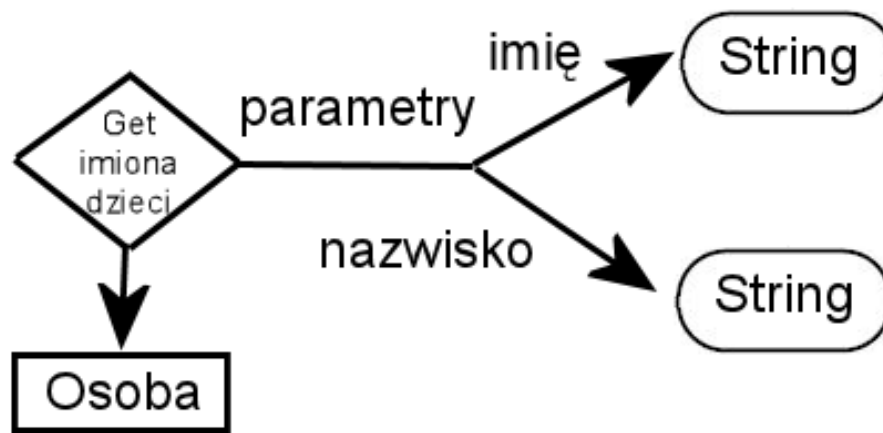
- ➔ Obiekty muszą być zgodne z ich definicjami w schemacie bazy
- ➔ Unikalność etykiet krawędzi i wierzchołków

Model grafowy GOOD

Obiektość GOOD

➔ Metody

Definiowanie operacji na danych dla poszczególnych klas, np. zapytanie o imiona dzieci obiektu klasy osoba:

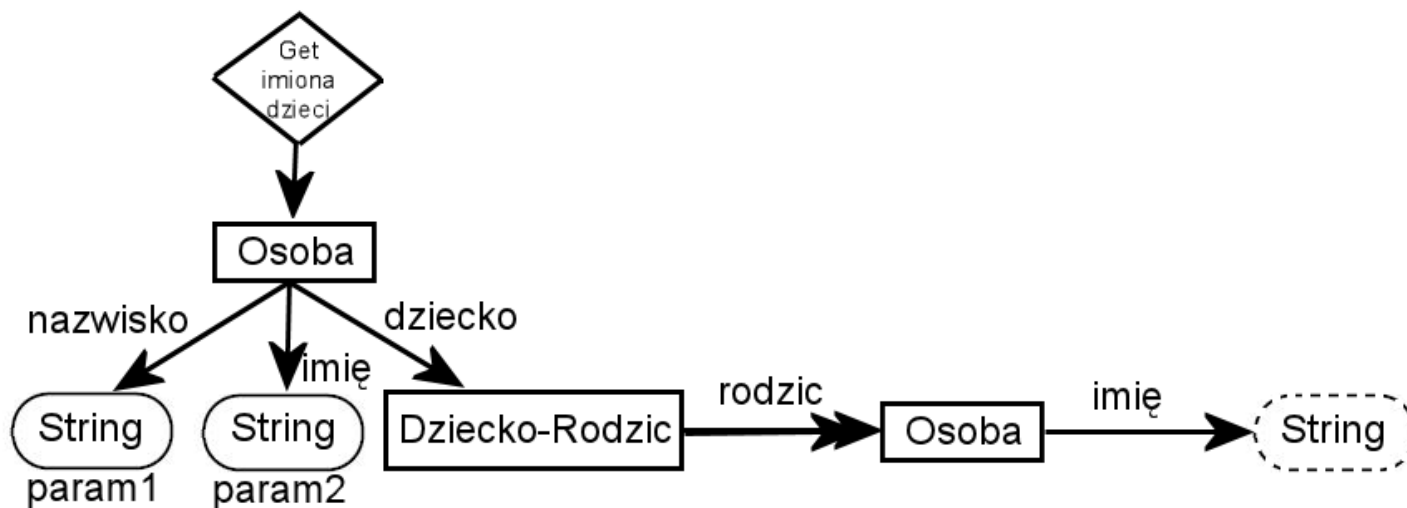


Model grafowy GOOD

Obiektość GOOD

➔ Metody

Definiowanie operacji na danych dla poszczególnych klas, np. zapytanie o imiona dzieci obiektu klasy osoba:



Model grafowy GOOD

Obiektowość GOOD

- ➔ Hierarchia klas i dziedziczenie

Szczałkowe mechanizmy:

Hierarchia wymieszana ze schematem bazy i przez niego definiowana.

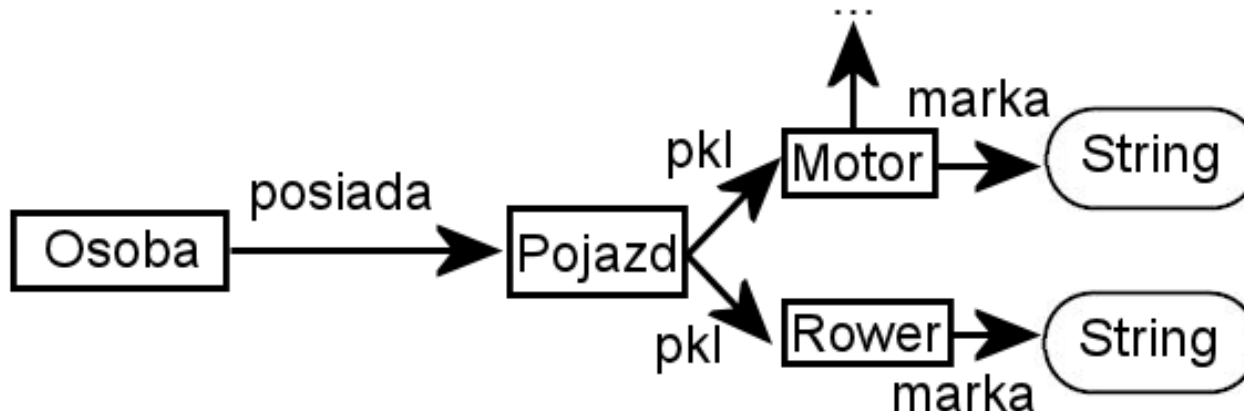
Dziedziczenie - możliwość odwoływania się do atrybutów podklasy w zapytaniu do nadklasy.

Model grafowy GOOD

Obiektość GOOD

- ➔ Hierarchia klas i dziedziczenie

Dziedziczenie - przykład:

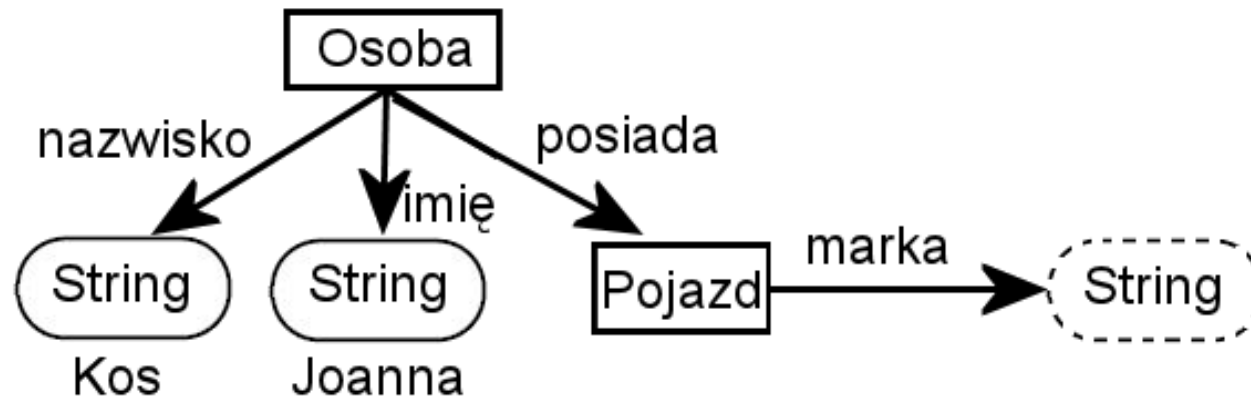


Model grafowy GOOD

Obiektość GOOD

- ➔ Hierarchia klas i dziedziczenie

Dziedziczenie - przykład:



Zastosowania modelu grafowego

- Prymitywna obiektowość
- Relacyjność a grafowość
- Interfejs użytkownika
- Związki między danymi „obywatelami pierwszej kategorii”
- Sąsiedztwa, ścieżki, odległość, grupowanie...

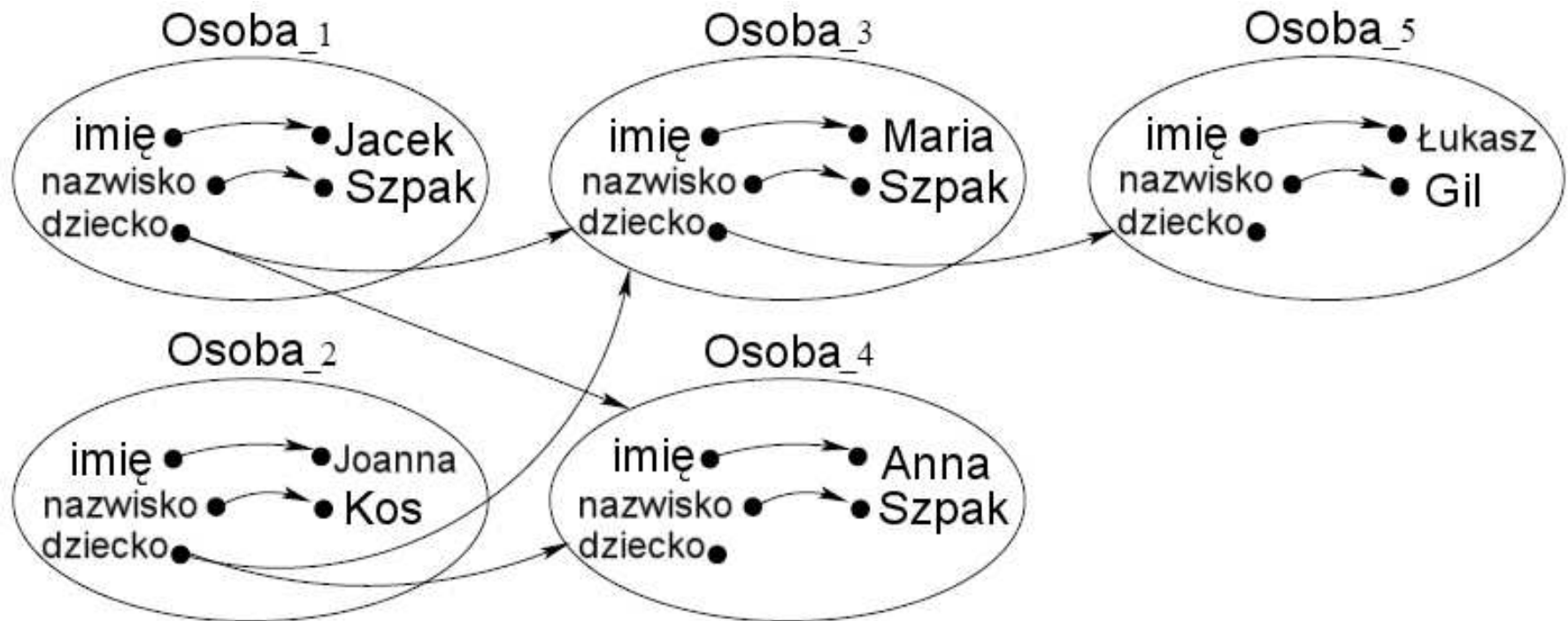
Zastosowania modelu grafowego

Modele danych o charakterze „sieciowym”:

- Socjologia - sieci znajomych, kontaktów biznesowych, współpracy...
- Przepływ informacji - cytowania artykułów naukowych.
- Technika - sieci elektryczne, telefoniczne, autostrady, koleje, rzeki, internet...
- Chemia/Biologia - struktury cząsteczek, funkcje regulujące genów...

Inne modele grafowe

Model *Hypernode* (Levene, Poulouvassilis 1990)

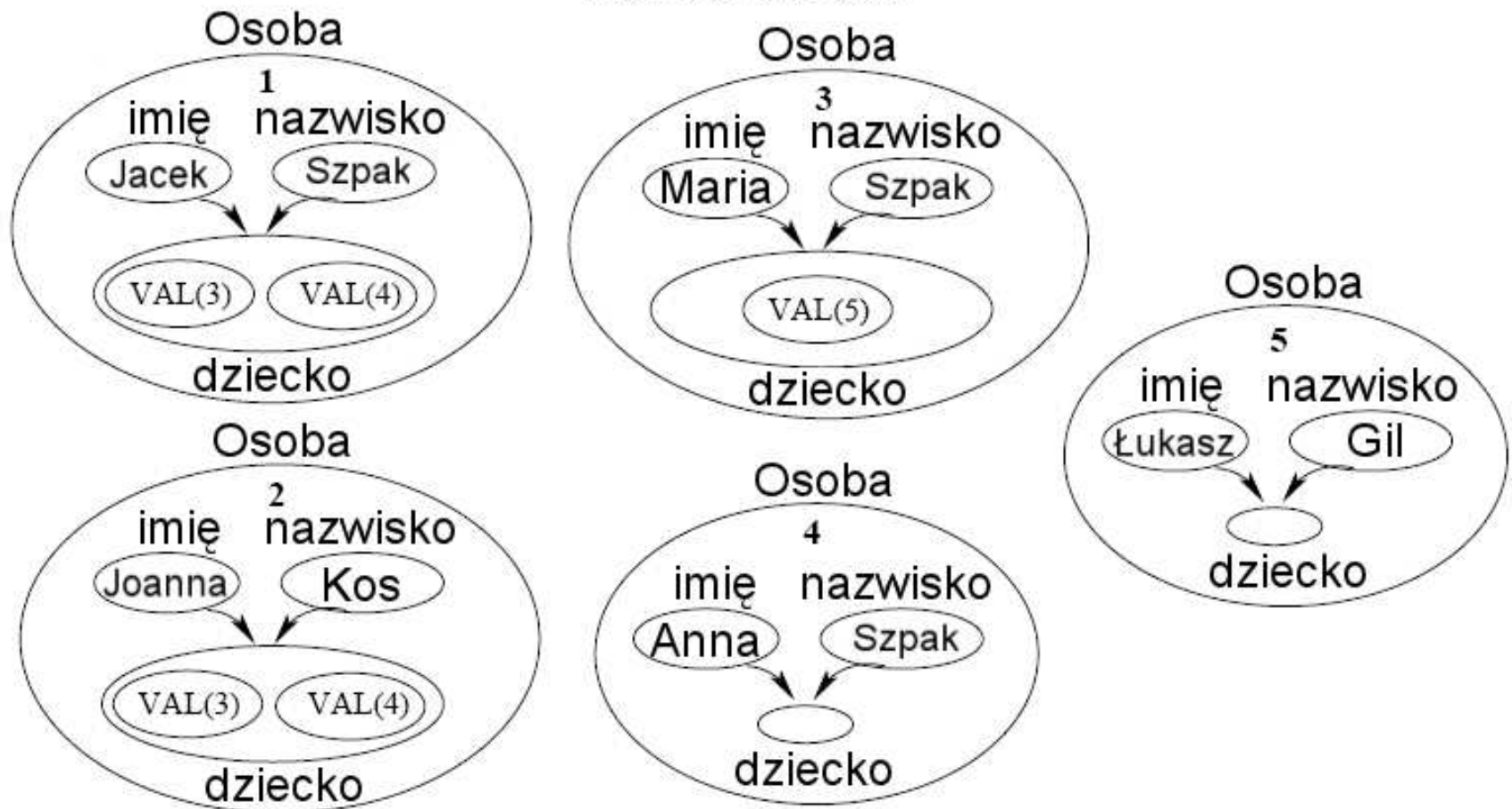


Inne modele grafowe

GROOVY (Levene, Poulouvassilis 1991)

(Graphically Represented Object-Oriented data model with Values)

Dziecko-Rodzic



Wizualizacja grafów

- Algorytmy „energetyczne” lub „sprężynowe” (*spring embedding*)
- Algorytm „geometryczny” - składowe główne (*High-dimensional embedding*)

Wizualizacja grafów

Algorytm sprężynowy

1. Rozpięcie „sprężynek” między każdą parą sąsiadujących wierzchołków (v, w) o „naturalnej długości” $d_{v,w}$.
2. Minimalizacja całkowitej energii układu:

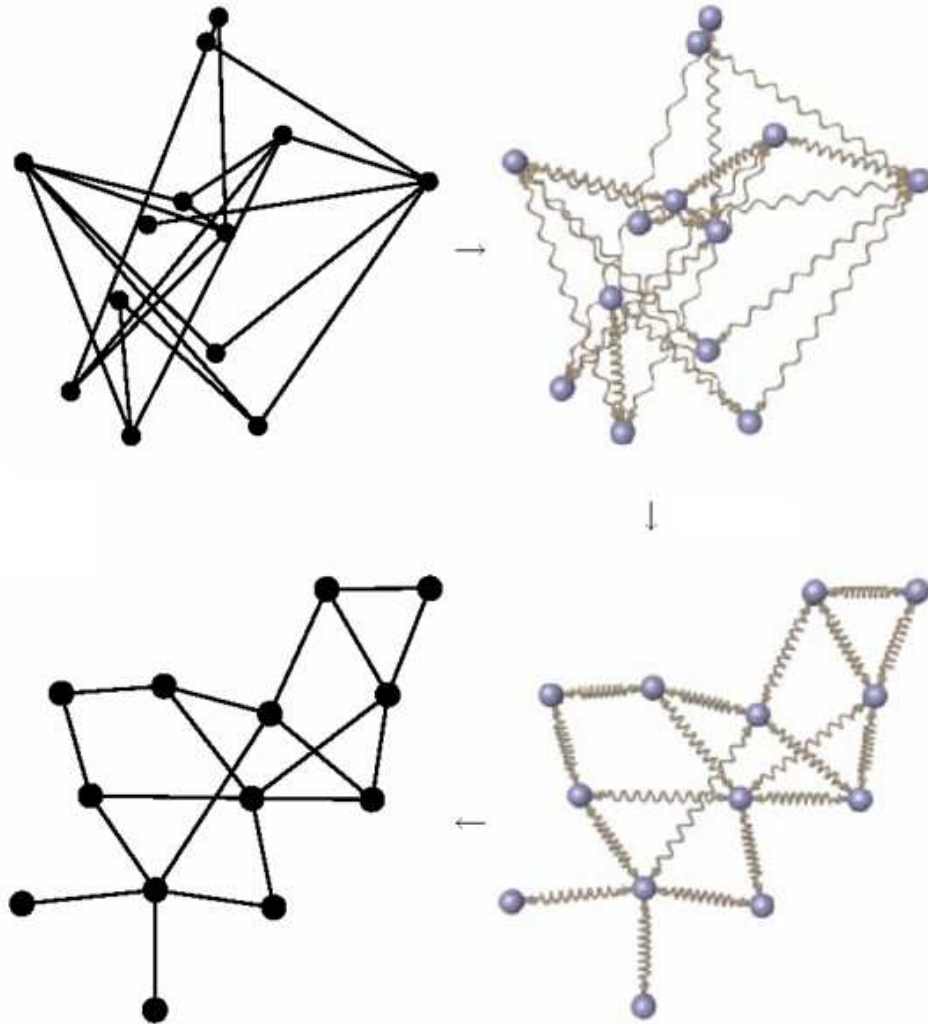
$$\min_{x_w, w \in W} \sum_{w \in W} \sum_{v \in W, v \leftrightarrow w} E_s(x_w, x_v)$$

gdzie

$$E_s(x_w, x_v) = (|x_w - x_v| - d_{w,v})^2$$

Wizualizacja grafów

Algorytm sprężynowy



Wizualizacja grafów

Algorytm geometryczny

„Rozpinanie” wierzchołków grafu w wielowymiarowej przestrzeni:

n - liczba wszystkich wierzchołków

m - wymiar przestrzeni ($m \ll n$)

1. Wybór m „wiodących” wierzchołków p_1, \dots, p_m stanowiących osie przestrzeni
2. Rozmieszczenie pozostałych $n - m$ wierzchołków według ich odległości w grafie od p_1, \dots, p_m
3. Obcięcie wymiaru przestrzeni do 2 (lub 3) używając metody składowych głównych

Podsumowanie

- ➔ Model grafowy - „ślepa uliczka” ewolucji baz danych?

Bibliografia

M. Gyssens, J. Paredaens, J. Van den Bussche, D. Van Gucht (1990) - *A Graph-Oriented Object Database Model*, Tech. Rep. 91-27, University of Antwerp (UIA), Belgium

R. Angeles, C. Gutierrez (2005) - *Survey of Graph Database Models*, Technical Report TR/DCC-2005-10, Universidad de Chile

M. Levene, A. Poulouvasilis (1990) - *The Hypernode Model and its Associated Query Language*, Proc. of the 5th Jerusalem Conf. on Information technology. IEEE Computer Society Press, 520–530.

M. Levene, A. Poulouvasilis (1990) - *An Object-Oriented Data Model Formalised Through Hypergraphs*, Data & Knowledge Engineering (DKE) 6, 3, 205–224.



➔ **Dziękuję za uwagę!**