

## Bazy danych

**Jacek Rumiński**



**Adam Bujnowski**

wykład 3

## Plan wykładu

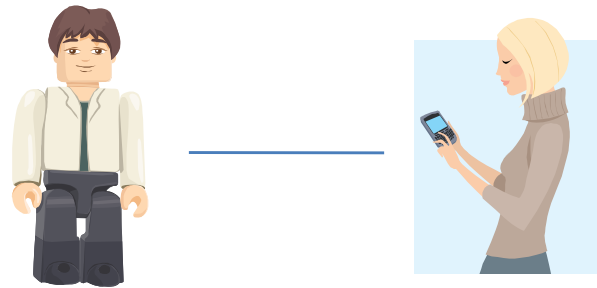
1. Związki pomiędzy encjami – rekordami
2. Model hierarchiczny
3. Model sieciowy

W czasie poprzedniego wykładu poznaliśmy **model płaski** (nie ma związków pomiędzy rekordami). Czy są inne modele danych? Oczywiście tak.

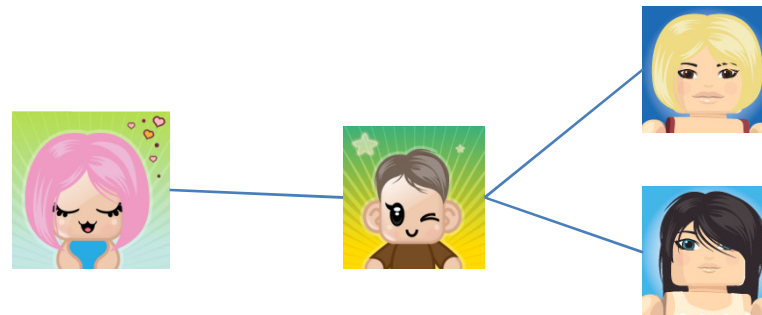
Zanim je jednak przedstawimy rozważmy podstawowe rodzaje związków jakie mogą zachodzić pomiędzy encjami, są to:

- związek jeden do jednego,
- związek jeden do wielu lub wielu do jednego,
- związek wiele do wielu.

Związek jeden do jednego oznacza, że w świecie realnym dokładnie jedna encja jest w związku z dokładnie jedną, inną encją.



W przypadku związku jeden do wielu (analogiczne wiele do jednego) jedna encja jest powiązana z wieloma, innymi encjami.



Dla związku wiele do wielu istnieje sytuacja, w której wiele encji jest powiązanych z wieloma, innymi encjami.

## Co oznaczają związki?

Rozpatrzmy następujący **przykład**. Niech będą dane dwie kolekcje rekordów: kolekcja mężczyzn i kolekcja kobiet.

W tradycyjnym modelu małżeństwa występuje związek jeden do jednego, czyli dokładnie jedna kobieta  $X_1$ , jest w związku z dokładnie jednym mężczyzną  $Y_1$ . Inaczej nie ma w kolekcji kobiet, takiego innego rekordu poza  $X_1$ , który wskazywałby na powiązanie z  $Y_1$ .

W niektórych jednak kulturach (historycznie i obecnie) możliwy jest związek małżeński pomiędzy jednym mężczyzną i wieloma kobietami. Zatem jeden rekord  $Y_1$  jest powiązany z wieloma rekordami kolekcji kobiet ( $X_1, X_2, X_3, \dots$ ).

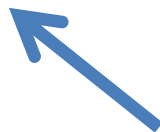
Oczywiście może być również odwrotnie (wielu mężczyzn związanych jest z jedną kobietą).

Związek wiele do wielu jest stosownym rozwinięciem przykładu....

W kolekcji danych (jako odzwierciedleniu świata realnego) występują związki pomiędzy rekordami. Stąd grupie encji odpowiada grupa rekordów, a grupie związków pomiędzy encjami grupa związków pomiędzy rekordami.

Jak zamodelować takiego związku. Rozpatrzmy uproszczone struktury dla rekordów mężczyzn i kobiet:

```
Male {  
    int maleID;  
    char[40] lName ;  
    char[20] fName;  
    int femaleIDasWife;  
}
```



```
Female {  
    int femaleID;  
    char[40] lName;  
    char[20] fName;  
}
```

Utwórzmy teraz przykładowe rekordy dla tych struktur:

Kobiety:

*{1, 'Kowalska', 'Janina'}*

*{2, 'Nowak', 'Krystyna'}*

Mężczyźni:

*{1, 'Wypych', 'Andrzej', 0}*

*{2, 'Czwartek', 'Aleksander', 2}*

*{3, 'Uniwersytecki', 'Kazimierz', 1}*

W przykładzie wykorzystaną wartość **0** jako oznaczenie braku związku oraz założenie, że wartość identyfikatora kobiety w polu ***femaleIDasWife*** nie może się powtarzać (inaczej byłby to związek wiele do jednego: wiele kobiet ma tego samego męża).

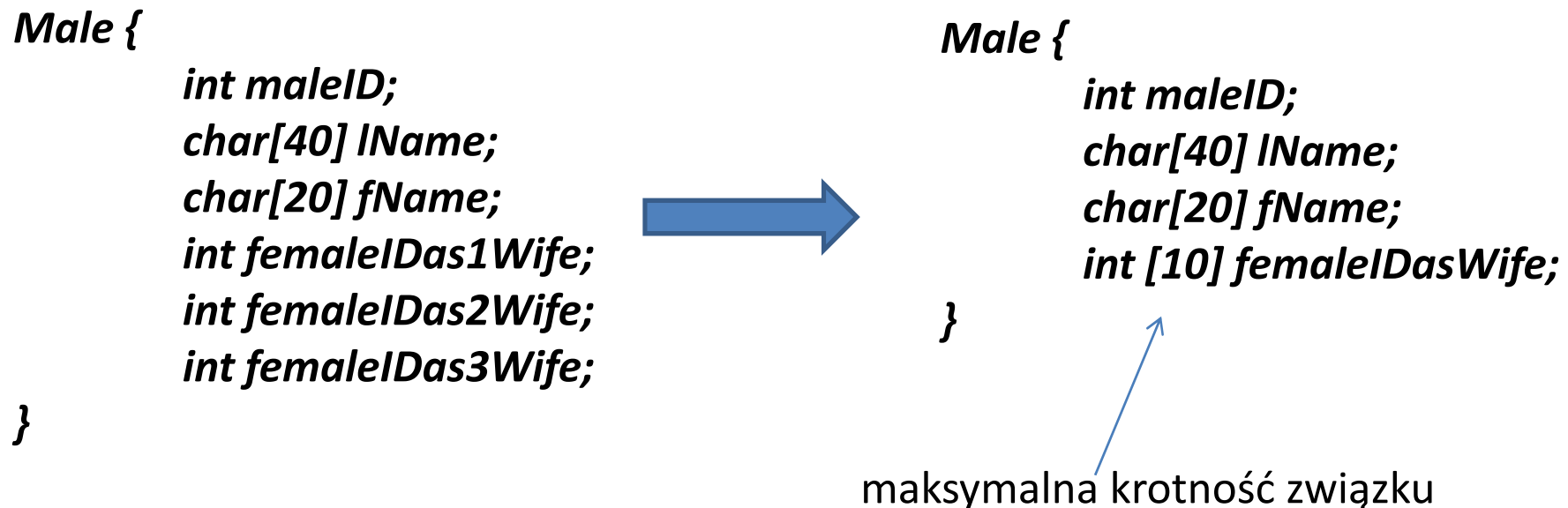
# Bazy danych – związki



Przedmiot: *Bazy danych*

Politechnika Gdańska, *Inżynieria Biomedyczna*

Jak zmienić powyższe struktury, aby utrzymać związek jeden do wielu?  
Wystarczy dodać kolejne atrybuty, które wskazują kobiety jako żony:



Przykładowy mężczyzna będący w związku małżeńskim z dwoma kobietami może być reprezentowany przez rekord:

**{2, 'Czwartek', 'Aleksander', 2, 1, 0}.**



Jeśli nie można zastosować tablicy do przechowywania związku, można zmienić definicje struktur do postaci:

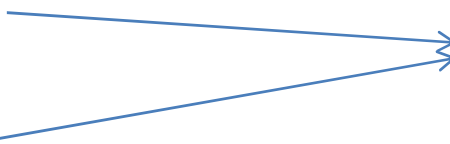
```
Male {  
    int maleID;  
    char[40] lName ;  
    char[20] fName;  
}
```

```
Female {  
    int femaleID;  
    char[40] lName;  
    char[20] fName;  
    int maleIDasHusband;  
}
```

Wówczas, rekordy reprezentujące kobiety będą przechowywać informacje wskazującą partnera w związku małżeńskim, np.:

```
{1, 'Kowalska', 'Janina', 1}  
{2, 'Nowak', 'Krystyna', 2}  
{2, 'Pietrzak', 'Anna', 1}
```

Wiele do jednego



# Bazy danych – związki



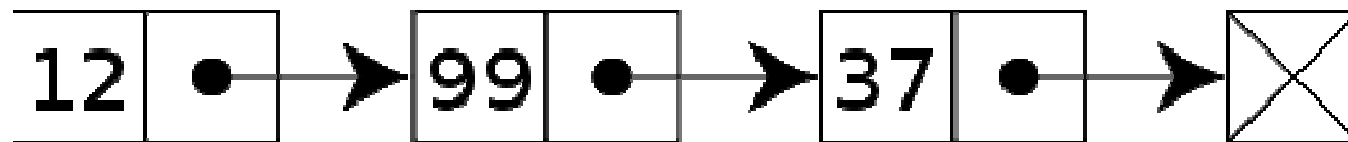
Przedmiot: *Bazy danych*

Politechnika Gdańska, *Inżynieria Biomedyczna*

Związki pomiędzy rekordami są definiowane na poziomie struktur danych. Tworzą złożone struktury danych, do których zaliczamy między innymi listy, drzewa, itd.

Struktury takie w ujęciu wewnętrznej organizacji danych nie przechowują jednak identyfikatorów wskazywanych rekordów, lecz wskaźnik (adres pamięci) na powiązany rekord. Wówczas z samej definicji struktury wynika sposób wykorzystania pamięci (przemieszczania się w niej pomiędzy rekordami), który może być dynamiczny (dodawanie kolejnych rekordów).

Przykładowo lista będzie organizowała kolejne rekordy w taki sposób, że jeden rekord będzie posiadał informacje o adresie pamięci, pod którym zapisano kolejny rekord, itd.



Źródło: Wikipedia

Wykorzystanie złożonych (dynamicznych) struktur danych oraz dodatkowych kryteriów doprowadziło do utworzenia wielu modeli danych (z wyłączeniem modelu płaskiego):

- hierarchiczny model danych,
- sieciowy model danych,
- relacyjny model danych,
- obiektowy model danych,
- obiektowo-relacyjny model danych i inne.

## Plan wykładu

1. Związki pomiędzy encjami – rekordami
2. Model hierarchiczny
3. Model sieciowy

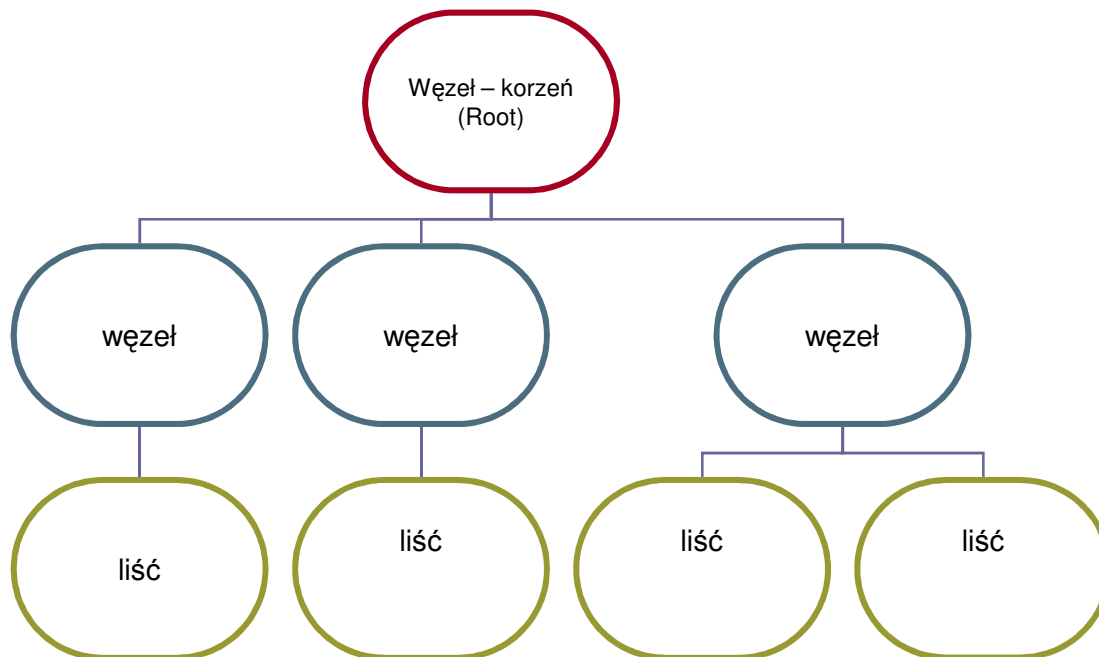
# Bazy danych – model hierarchiczny



Przedmiot: *Bazy danych*

Politechnika Gdańska, *Inżynieria Biomedyczna*

Podstawą hierarchicznego modelu danych jest struktura danych określana jako drzewo. Począwszy od węzła (wierzchołka) podstawowego, będącego korzeniem drzewa (np. "c:\" w systemie plików) poprzez krawędzie - rozgałęzienia (np. foldery) dochodzimy do ostatnich zbiorów danych - liści (np. pliki). System hierarchiczny jest często przedstawiany poprzez związek „rodzic-potomek”. Rodzic posiada pewną liczbę potomków, potomkowie ci mogą mieć swoich potomków, ci z kolei ...



Jacek Rumiński



Źródło: <http://cordis.europa.eu/esprit/icons/tree.jpg>

Struktura danych – drzewo – posiada wiele realizacji związanych z szeregiem dodatkowych warunków. Przykładowo drzewo binarne dopuszcza podział węzła tylko na dwa podwężły (inaczej podział na dwa podrzewa lub dwie gałęzie). Wówczas taką strukturę możemy zaprojektować w sposób następujący

```
struct drzewoBinarne {  
    struct drzewo *prawe;  
    struct drzewo *lewe;  
    int wartosc;  
    //inne dane  
};
```

Rekord zgodny ze strukturą drzewa binarnego będzie przechowywał własny zestaw danych (atrybuty opisowe, np. nazwisko, imię, itp., w przykładzie struktury reprezentowane przez „wartosc”) oraz wskaźniki na rekordy stanowiące lewe i prawe rozgałęzienie (poddrzewa).

Oczywiście do obsługi drzewa nie wystarczy sama struktura danych.  
Potrzebny jest kod oprogramowania

**(struktury+algorytmy=programy),**

który będzie kontrolował, czy w wyniku kolejnej operacji na danych (czyli na rekordach) zachowane są wymagania dotyczące danych w strukturze drzewa (np. wartości wskaźników właściwie odnoszą się do istniejących rekordów).

Zasady zgodności danych z wymaganiami dotyczącymi struktury wchodzą w skład tzw. **warunków integralności danych**.

Określenie zasad definiowania i obsługi warunków integralności stanowi część definicji modelu danych (tutaj modelu hierarchicznego).

Hierarchiczny model danych zakłada podstawowe **warunki integralności** danych:

- *każdy rekord (z wyjątkiem pierwszego rodzica - korzenia drzewa) musi posiadać własnego, jednego rodzica,*
- *jeżeli dany rekord posiada więcej rodziców niż jeden musi być skopiowany dla każdego rodzica oddzielnie tak, aby prawdziwa była powyższa zasada,*
- *jeżeli usunięty zostaje dany rekord oznacza to, że usunięte zostają również wszystkie wywodzące się z niego rekordy – potomkowie (węzły i liście).*

Podstawowy związek jaki występuje pomiędzy rekordami to związek jeden do wielu ( korzeń – wierzchołki/liście) i wielu do jednego (wierzchołki – korzeń).



Aby w pełni odnieść się do definicji modelu danych pozostaje tylko wskazać operacje na danych hierarchicznych:

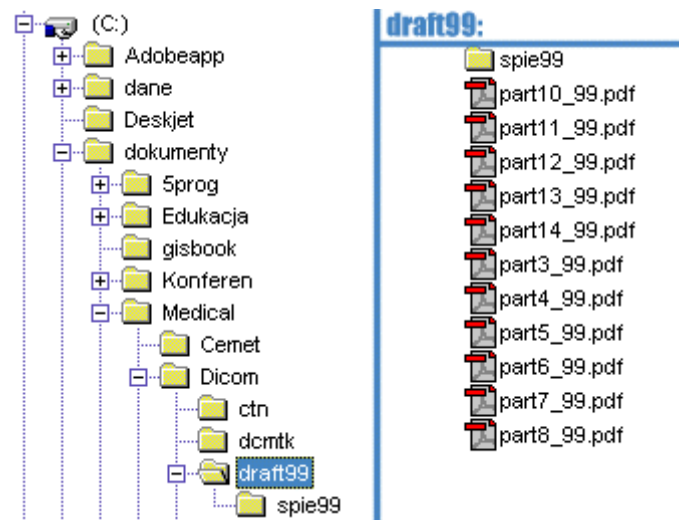
- ***zapis/odczyt drzewa/elementów drzewa (operacja związana z utrwaleniem drzewa w pamięci),***
- ***wyszukanie elementu drzewa,***
- ***dodanie elementu drzewa,***
- ***usunięcie elementu drzewa.***

Elementem drzewa jest rekord reprezentujący wierzchołek (węzeł) lub liść. Należy pamiętać, że rekord taki posiada również wartości atrybutów (poza wskaźnikami określającymi związki z innymi rekordami).

Operacja na elemencie drzewa będzie związana z przetwarzaniem (odczyt, zapis, modyfikacja) wartości atrybutów opisowych danego elementu jak również atrybutów opisujących związek z innymi rekordami.

## DEMO 3

# System plików



Czy oprócz systemów plików znane są inne zastosowania modelu hierarchicznego? Oczywiście tak.

## Języki znaczników!

Podstawowy język XML (ang. eXtensible Markup Language, rozszerzalny język znaczników) to typowe zastosowanie modelu hierarchicznego, w który istnieje tylko jeden główny element (root), zawierający kolejne subelementy, te natomiast mogą obejmować inne subelementy, itd.

Język XML bezpośrednio lub poprzez swoje dialekty (np. HTML, SVG, GML) jest powszechnie stosowany do przechowywania danych/informacji oraz reprezentowania wiedzy (ontologie).

typ	wersja	strona kodowa	rozłączność
<?xml	version="1.0"	encoding="ISO-8859-2"	standalone="no"?

```
<!DOCTYPE salon SYSTEM "salon.dtd">
<salon wlasciciel="&kontakt;">
<!--Stan salonu-->
  <samochod VIN="121241242d32432" nrsilnika="23124432">
    <marka> Renault</marka>
    <model>Clio 1.2RN</model>
    <kolor> czerwony</kolor>
    <silnik miara="ccm">1159</silnik>
  </samochod>
  <samochod VIN="8785654343242" nrsilnika="542435324">
    <marka> BMW</marka>
    <model>730</model>
    <kolor> czarny</kolor>
    <silnik miara="ccm">3234</silnik>
  </samochod>
  <samochod VIN="4343567565475643" nrsilnika="98996796">
    <marka> Ford</marka>
    <model>Mondeo</model>
    <kolor> zielony</kolor>
    <silnik miara="ccm">1567</silnik>
  </samochod>
</salon>
```

Kolejny wykład poświęcony będzie realizacji modelu hierarchicznego poprzez XML.

Czy oprócz systemów plików znane są inne zastosowania modelu hierarchicznego? Oczywiście tak.

## **Indeksowanie danych – warianty struktury danych typu drzewo**

Omawiając strukturę danych drzewa posłużyliśmy się drzewem binarnym.

Wśród typowych kategorii drzew można jeszcze wyróżnić: B-drzewa, B+-drzewa, B\*-drzewa, R-drzewa, R+-drzewa, R\*-drzewa, kd-drzewa, VP-drzewa, M-drzewa, UB-drzewa, X-drzewa, itd.

Poszczególne struktury danych (i związane z nimi algorytmy) mają istotne zastosowania jako wewnętrzne organizacje danych w pamięci oraz jako specjalne struktury danych ułatwiające wyszukiwanie danych (indeksowanie).

## PRZYKŁAD

Rozpatrzmy przykład wykorzystania drzewa binarnego (którego uogólnieniem jest B-drzewo) do poprawy szybkości wyszukiwania rekordów.

Drzewo binarne to takie drzewo, którego węzły posiadają co najwyżej dwóch potomków.

W celu zilustrowania problemu wyszukiwania danych posłużmy się zbiorem rekordów nauczycieli. Jeśli pragniemy znaleźć rekord (rekordy), które w polu (atrybut) „IName” przechowują wartość „Kowalski” to ile trzeba wykonać operacji odczytu? Stosując typową metodę (odczyt sekwencyjny, rekord po rekordzie) może się zdarzyć, że poszukiwany rekord odczytamy jako ostatni (zakładając, że rekordy nie są uporządkowane według wartości atrybutu „IName”).

Czy można przyspieszyć proces wyszukiwania? Oczywiście tak.

## PRZYKŁAD

Istnieje wiele sposobów, ale my posłużymy się następującym zastosowaniem drzewa binarnego. Utwórzmy dodatkową strukturę drzewa binarnego, o postaci:

```
struct drzewoBinarneIndeksy {  
    struct drzewo *mniejsze;  
    struct drzewo *wieksze;  
    int *wskaznikNaRekord;  
};
```

Pole **wskaznikNaRekord** jest wskaźnikiem na rzeczywisty rekord danych. Zakładając, że wartości „lName” (w rozpatrywanym przykładzie), są unikalne możemy zbudować drzewo binarne sortując wartości atrybutu „lName”. Rekord o takiej wartości atrybutu „lName”, która przypada w środku zbioru po sortowaniu będzie wskazywany przez węzeł główny (Root) drzewa binarnego. Podzbiór rekordów o wartości pola „lName”, mniejszej niż wartość środkowa będzie tworzył drzewo wskazywane przez wskaźnik *\*mniejsze*, analogicznie utworzone będzie drzewo wskazywane przez wskaźnik *\*wieksze* dla rekordów o wartości pola większej niż wartość środkowa. Każde utworzone drzewo będzie znowu dzielone na dwa, itd.

# Bazy danych – model hierarchiczny



Przedmiot: *Bazy danych*

Politechnika Gdańska, *Inżynieria Biomedyczna*

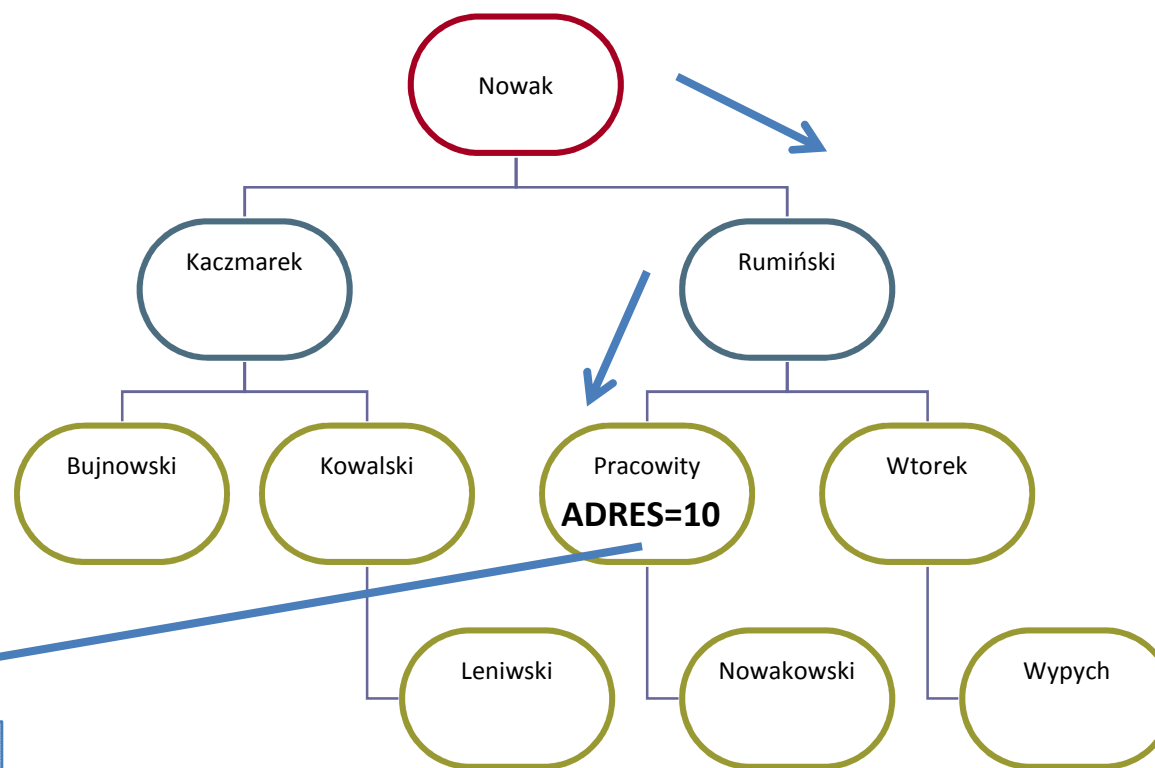
## PRZYKŁAD

Chcemy odnaleźć informacje („inne dane”) o osobie o nazwisku „Pracowity”.

### Dane źródłowe

Adres	IName	Inne dane
1	Nowak	
2	Kowalski	
3	Rumiński	
4	Bujnowski	
5	Wtorek	
6	Kaczmarek	
7	Nowakowski	
8	Wypych	
9	Leniowski	
10	Pracowity	

### Dane indeksowane (IName)



## Podsumowanie

Hierarchiczny model danych bazuje na strukturze drzewa. Istnieje wiele realizacji struktury drzewa. Najbardziej popularne to drzewa binarne i B-drzewa.

W modelu hierarchicznym wykorzystywane są związki jeden do jednego oraz jeden do wielu (wiele do jednego).

Model ten jest najbardziej popularny w praktycznych zastosowaniach składowania danych, do których można zaliczyć: systemy plików, dokumenty XML i pochodne oraz metody indeksowania danych.





## Plan wykładu

1. Związki pomiędzy encjami – rekordami
2. Model hierarchiczny
3. Model sieciowy

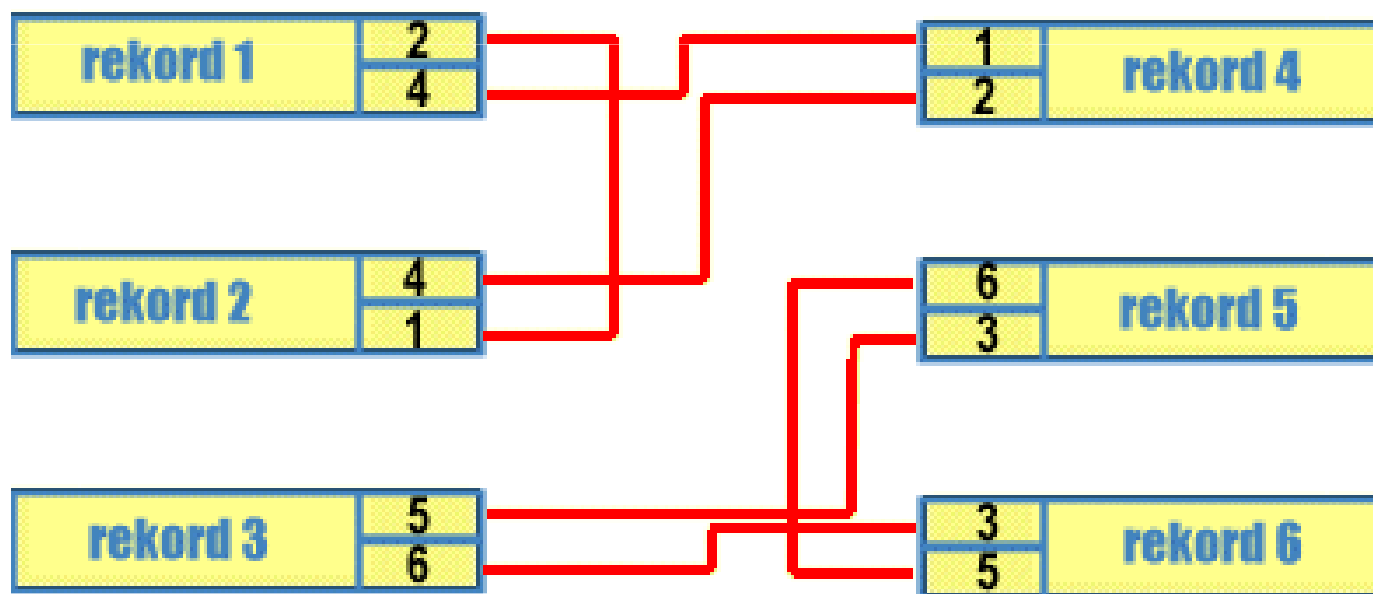
# Bazy danych – model sieciowy



Przedmiot: *Bazy danych*

Politechnika Gdańska, *Inżynieria Biomedyczna*

W modelu tym możliwe jest powiązanie dowolnego rekordu z wieloma innymi i szerzej możliwe są związki wiele do wielu. Przykładowo rekord A wskazuje na zbiory B i C, zbiór C na zbiór D, zbiór D na zbiór B, itp. Tworzona jest więc sieć powiązań i dlatego model nazywany jest modelem sieciowym.



W modelu sieciowym możliwe jest wykorzystanie praktycznie wszystkich typów związków (jeden do jednego, jeden do wielu, wiele do jednego i wiele do wielu).

W przypadku małych baz danych liczba wskaźników jest również niewielka co umożliwia szybkie wyszukiwanie informacji. W przypadku dużych baz danych wielka liczba wskaźników znacznie ogranicza szybkość wyszukiwania danych co jest istotną wadą tego modelu.

Model sieciowy jest obecnie bardzo rzadko stosowany. W zakresie wielu powiązań (bez ograniczenia potomek ma tylko jednego rodzica) zastąpił go model relacyjny (dostępny poprzez o wiele prostszy interfejs). W innym przypadku można zastosować model hierarchiczny.

# Bazy danych – koniec wykładu 3



Przedmiot: *Bazy danych*

Politechnika Gdańska, *Inżynieria Biomedyczna*

## Co dalej?

Zapoznaliśmy się właśnie z podstawowymi rodzajem związków pomiędzy encjami oraz z modelem hierarchicznym i sieciowym. W czasie kolejnego wykładu przedstawiony zostanie XML jako nowoczesny format składowania i wymiany danych realizujący model hierarchiczny. W kolejnych wykładach poznamy model relacyjny.

# ZAPRASZAMY NA WYKŁAD 4