

# **Transformacja modelu ER do modelu relacyjnego**

**Wykład przygotował:  
Robert Wrembel**



## Plan wykładu

- Transformacja encji
- Transformacja związków
- Transformacja hierarchii encji

Celem wykładu jest omówienie technik transformacji modelu związków-encji do modelu relacyjnego. W ramach wykładu zostaną omówione podstawowe techniki transformacji encji do modelu relacyjnego, transformacji związków i transformacji hierarchii encji.



## Pojęcia podstawowe (1)

- Schemat bazy danych
  - zbiór schematów relacji
- Relacja (tabela)
  - dwu-wymiarowa tablica
  - kolumny  $\Leftrightarrow$  atrybuty
  - wiersze  $\Leftrightarrow$  krotki, rekordy
    - każda krotka reprezentuje wystąpienie encji

Tytułem przypomnienia podamy podstawowe definicje, do których odwołuje się wykład. Schemat bazy danych jest zbiorem schematów relacji. Relacja, zwana tabelą, jest postrzegana jako dwu-wymiarowa tablica. Kolumny tej tablicy są nazywane atrybutami, a wiersze - krotkami lub rekordami. W modelu relacyjnym każda krotka reprezentuje wystąpienie encji.



## Pojęcia podstawowe (2)

- Klucz podstawowy
  - atrybut lub zbiór atrybutów - wybrany spośród kluczy potencjalnych
- Klucz obcy
  - atrybut lub zbiór atrybutów wskazujący na klucz podstawowy innej relacji
    - atrybut lub zbiór atrybutów w relacji B, będący jednocześnie kluczem podstawowym w relacji A
  - należy zaznaczyć, że klucz obcy może odnosić się do klucza podstawowego samej relacji, w której został on zdefiniowany

Kluczem podstawowym relacji nazywamy atrybut lub zbiór atrybutów jednoznacznie identyfikujący krotkę relacji.

Kluczem obcym nazywamy atrybut lub zbiór atrybutów wskazujący na klucz podstawowy innej relacji. Innymi słowy jest to atrybut lub zbiór atrybutów w relacji B, będący jednocześnie kluczem podstawowym w relacji A. Należy zaznaczyć, że klucz obcy może odnosić się do klucza podstawowego samej relacji, w której został on zdefiniowany.



## Transformacja

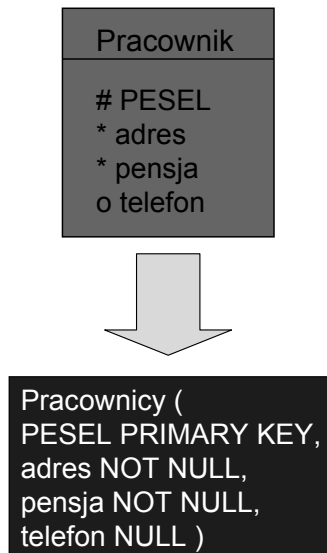
- Model ER  $\Leftrightarrow$  schemat relacyjny
- Transformacja
  - encji z atrybutami
  - związków
  - hierachii encji

Transformacja modelu ER jest konieczna, ponieważ jak pamiętamy jest on modelem abstrakcyjnym niezależnym od implementacji. Modelem do którego transformujemy jest model relacyjny. Jest on modelem implementacyjnym baz danych. Transformacji podlegają wszystkie obiekty modelu ER, czyli encje z atrybutami, związki i hierarchie encji.

Omawianie technik transformacji rozpoczniemy od przypadku najprostszego, czyli transformacji encji.



## Reguły transformacji encji



- Encja ⇔ relacja
- Atrybut encji ⇔ atrybut relacji
- Typ danych atrybutu encji ⇔ typ danych atrybutu relacji
- Identyfikator encji ⇔ klucz podstawowy relacji
- Obowiązkowość atrybutów encji ⇔ ograniczenie NOT NULL
- Opcjonalność atrybutów encji ⇔ ograniczenie NULL
- Pozostałe ograniczenia integr. atrybutów encji ⇔ ograniczenia integr. atrybutów relacji

BD – wykład 4 (6)

Reguły transformacji encji są następujące.

1. Encja jest odwzorowywana w relację. Nazwa encji jest odwzorowywana w nazwę relacji. Uwaga: przyjmuje się, że nazwy relacji są rzeczownikami w liczbie mnogiej.
2. Atrybut encji jest odwzorowywany w atrybut relacji. Nazwy atrybutów encji są odwzorowywane w nazwy atrybutów relacji.
3. Typ danych atrybutu encji jest odwzorowywany w odpowiadający mu typ danych atrybutu relacji.
4. Unikalny identyfikator encji jest transformowany w klucz podstawowy relacji.
5. Obowiązkowość atrybutów encji jest reprezentowana w relacji w postaci ograniczenia NOT NULL zdefiniowanego na atrybucie relacji odpowiadającym atrybutowi encji.
6. Opcjonalność atrybutów encji jest reprezentowana w relacji w postaci ograniczenia NULL zdefiniowanego na atrybucie relacji odpowiadającym atrybutowi encji.
7. Ograniczenia integralnościowe dla atrybutów encji (unikalność, zawężenie dziedziny) są transformowane do odpowiadających im ograniczeń integralnościowych relacji. (Ograniczenia integralnościowe atrybutów encji zostały omówione w wykładzie 3, a ograniczenia integralnościowe atrybutów relacji - w wykładzie 2).



## Reguły transformacji związków

- Związek binarny 1:1  $\Rightarrow$  klucz obcy we wskazanej tabeli
- Związek unarny 1:1  $\Rightarrow$  klucz obcy w tej samej tabeli
- Związek binarny 1:M  $\Rightarrow$  klucz obcy w tabeli po stronie "wiele"
- Związek binarny M:N  $\Rightarrow$  tabela
- Związek unarny M:N  $\Rightarrow$  tabela

Przypadkiem bardziej złożonym jest transformacja związków.

Związek binarny 1:1 transformuje się do klucza obcego we wskazanej tabeli.

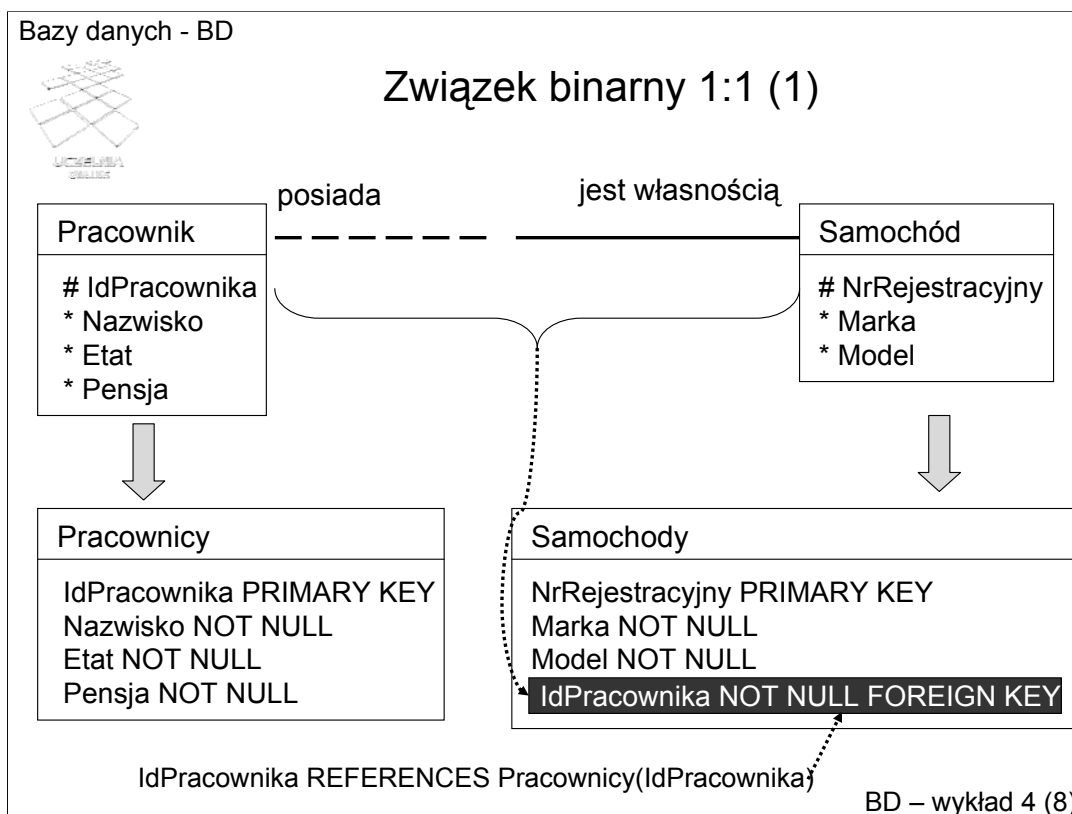
Związek unarny 1:1 transformuje się do klucza obcego w tej samej tabeli.

Związek binarny 1:M transformuje się do klucza obcego w tabeli po stronie "wiele".

Związek binarny M:N transformuje się do tabeli.

Związek unarny M:N transformuje się do tabeli.

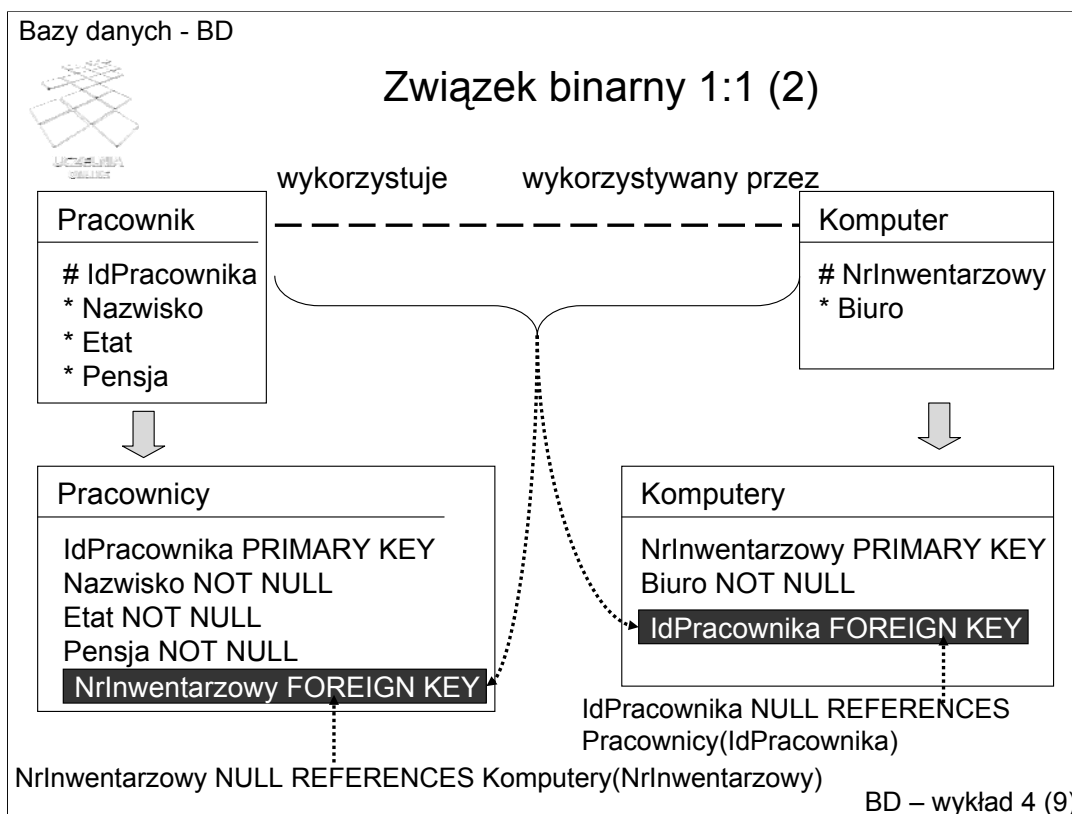
Wymienione przypadki transformacji związków zostaną dokładnie omówione i zilustrowane przykładami w dalszej części wykładu.



Związek binarny 1:1 jednostronnie obowiązkowy transformuje się do klucza obcego w tabeli po stronie związku obowiązkowego. Ograniczenie integralnościowe jest definiowane dla atrybutu klucza obcego. Klucz ten nie może przyjmować wartości pustych.

W przykładzie ze slajdu, z encji Pracownik powstaje tabela Pracownicy, a z encji Samochód - tabela Samochody. Związek pomiędzy tymi dwoma encjami jest transformowany do klucza obcego IdPracownika w tabeli Samochody. Klucz ten wskazuje na klucz podstawowy w tabeli Pracownicy, czyli IdPracownika. Wartość atrybutu klucza obcego musi być zawsze określona ponieważ związek jest obowiązkowy od strony encji Samochód.





Związek binarny 1:1 obustronnie opcjonalny transformuje się do klucza obcego w tabeli o mniejszym rozmiarze. Ograniczenie integralnościowe jest definiowane dla atrybutu klucza obcego. Atrybut ten może przyjmować wartości puste, ponieważ związek jest opcjonalny.

W przykładzie ze slajdu, z encji Pracownik powstaje tabela Pracownicy, a z encji Komputer - tabela Komputery. Związek jest transformowany do klucza obcego IdPracownika w tabeli Komputery. Klucz ten wskazuje na klucz podstawowy w tabeli Pracownicy, czyli IdPracownika. Klucz obcy może przyjmować wartości puste.



## Związek binarny 1:1 (3)

1.	<table border="1"> <thead> <tr> <th>Pracownicy</th> </tr> </thead> <tbody> <tr> <td>IdPracownika PRIMARY KEY</td> </tr> <tr> <td>...</td> </tr> <tr> <td><b>NrInwentarzowy FOREIGN KEY</b></td> </tr> </tbody> </table>	Pracownicy	IdPracownika PRIMARY KEY	...	<b>NrInwentarzowy FOREIGN KEY</b>	<table border="1"> <thead> <tr> <th>Komputery</th> </tr> </thead> <tbody> <tr> <td>NrInwentarzowy PRIMARY KEY</td> </tr> <tr> <td>Biuro NOT NULL</td> </tr> <tr> <td><b>IdPracownika FOREIGN KEY</b></td> </tr> </tbody> </table>	Komputery	NrInwentarzowy PRIMARY KEY	Biuro NOT NULL	<b>IdPracownika FOREIGN KEY</b>
Pracownicy										
IdPracownika PRIMARY KEY										
...										
<b>NrInwentarzowy FOREIGN KEY</b>										
Komputery										
NrInwentarzowy PRIMARY KEY										
Biuro NOT NULL										
<b>IdPracownika FOREIGN KEY</b>										
2.	<table border="1"> <thead> <tr> <th>Pracownicy</th> </tr> </thead> <tbody> <tr> <td>IdPracownika PRIMARY KEY</td> </tr> <tr> <td>...</td> </tr> </tbody> </table>	Pracownicy	IdPracownika PRIMARY KEY	...	<table border="1"> <thead> <tr> <th>Komputery</th> </tr> </thead> <tbody> <tr> <td>NrInwentarzowy PRIMARY KEY</td> </tr> <tr> <td>Biuro NOT NULL</td> </tr> <tr> <td><b>IdPracownika FOREIGN KEY</b></td> </tr> </tbody> </table>	Komputery	NrInwentarzowy PRIMARY KEY	Biuro NOT NULL	<b>IdPracownika FOREIGN KEY</b>	
Pracownicy										
IdPracownika PRIMARY KEY										
...										
Komputery										
NrInwentarzowy PRIMARY KEY										
Biuro NOT NULL										
<b>IdPracownika FOREIGN KEY</b>										
3.	<table border="1"> <thead> <tr> <th>Pracownicy</th> </tr> </thead> <tbody> <tr> <td>IdPracownika PRIMARY KEY</td> </tr> <tr> <td>...</td> </tr> <tr> <td><b>NrInwentarzowy FOREIGN KEY</b></td> </tr> </tbody> </table>	Pracownicy	IdPracownika PRIMARY KEY	...	<b>NrInwentarzowy FOREIGN KEY</b>	<table border="1"> <thead> <tr> <th>Komputery</th> </tr> </thead> <tbody> <tr> <td>NrInwentarzowy PRIMARY KEY</td> </tr> <tr> <td>Biuro NOT NULL</td> </tr> </tbody> </table>	Komputery	NrInwentarzowy PRIMARY KEY	Biuro NOT NULL	
Pracownicy										
IdPracownika PRIMARY KEY										
...										
<b>NrInwentarzowy FOREIGN KEY</b>										
Komputery										
NrInwentarzowy PRIMARY KEY										
Biuro NOT NULL										

BD – wykład 4 (10)

Możliwe przypadki transformacji związku 1:1 obustronnie opcjonalnego przedstawia slajd. Przypadek 1 jest stosowany niezwykle rzadko. Polega on na umieszczeniu kluczy obcych w obu tabelach wynikowych. Przypadek 2 został omówiony na poprzednim slajdzie. Przypadek trzeci polega na umieszczeniu klucza obcego w tabeli Pracownicy.

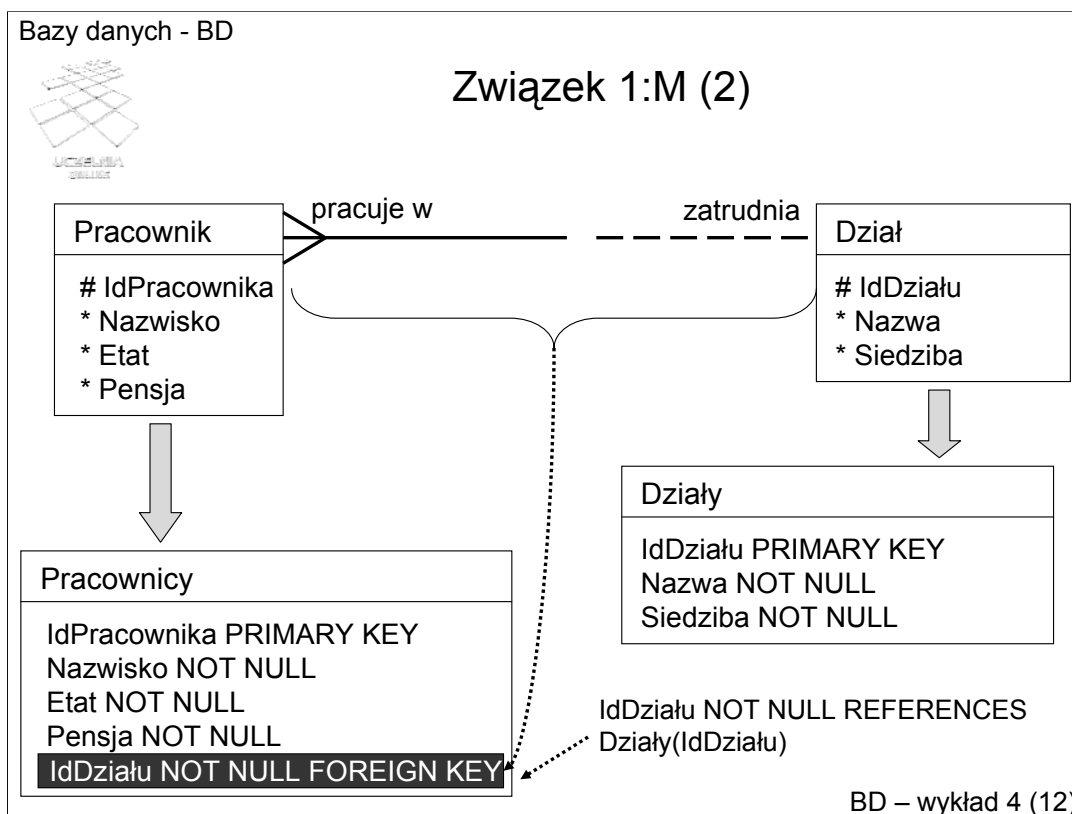


## Związek 1:M (1)

- Klucz obcy dodawany do relacji po stronie "wiele"
- Ograniczenia referencyjne definiowane dla klucza obcego
- Obowiązkowość związku po stronie "wiele"  $\Leftrightarrow$  ograniczenie NOT NULL definiowane na kluczu obcym
- Opcjonalność związku po stronie "wiele"  $\Leftrightarrow$  ograniczenie NULL definiowaną na kluczu obcym relacji
- Opcjonalność lub obowiązkowość związku po stronie "jeden" nie jest odwzorowywana w modelu relacyjnym

Reguły transformacji związku 1:M są następujące.

1. Klucz obcy jest dodawany do relacji po stronie "wiele" niezależnie od opcjonalności, czy obowiązkowości tego związku.
2. Ograniczenia integralnościowe referencyjne (tj. definiujące klucz obcy) są definiowane dla atrybutu reprezentującego klucz obcy.
3. Obowiązkowość związku po stronie "wiele" jest reprezentowana przez ograniczenie integralnościowe NOT NULL definiowane na kluczu obcym relacji.
4. Opcjonalność związku po stronie "wiele" jest reprezentowana przez ograniczenie integralnościowe NULL definiowaną na kluczu obcym relacji.
5. Opcjonalność lub obowiązkowość związku po stronie "jeden" nie jest odwzorowywana w modelu relacyjnym.



Przykład ze slajdu ilustruje sposób transformacji binarnego związku 1:M jednostronnie obowiązkowego. Z encji Pracownik powstaje tabela Pracownicy, a z encji Dział - tabela Działy. Klucz obcy jest dodawany do tabeli Pracownicy (strona "wiele") i wskazuje on na klucz podstawowy tabeli Działy, czyli IdDziału. Należy zwrócić uwagę, że klucz obcy IdDziału posiada ograniczenie NOT NULL ponieważ związek jest obowiązkowy od strony "wiele".

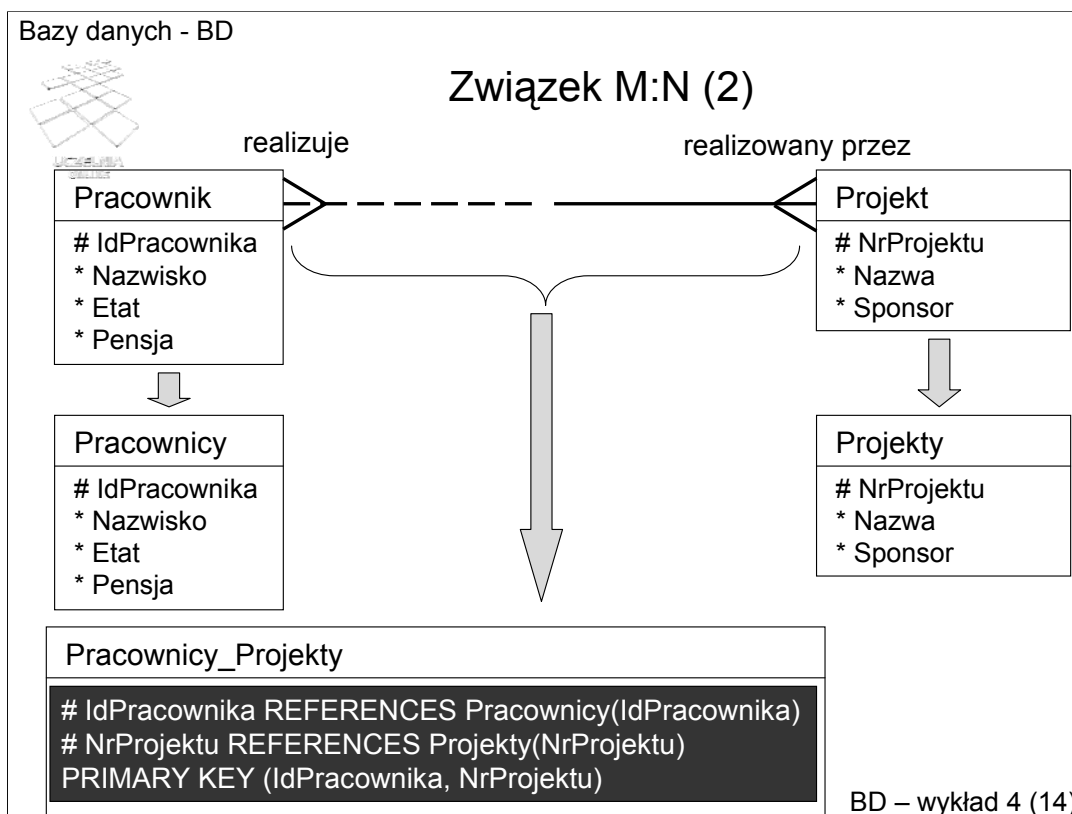


## Związek M:N (1)

- Reprezentowany poprzez dodatkową relację
- Nazwa relacji jest połączeniem nazw relacji powstałych z encji
- Relacja zawiera klucze obce wskazujące na klucze podstawowe relacji powstałych z powiązanych encji
- Ograniczenia referencyjne definiowane dla kluczy obcych
- Klucze obce tworzą klucz podstawowy relacji

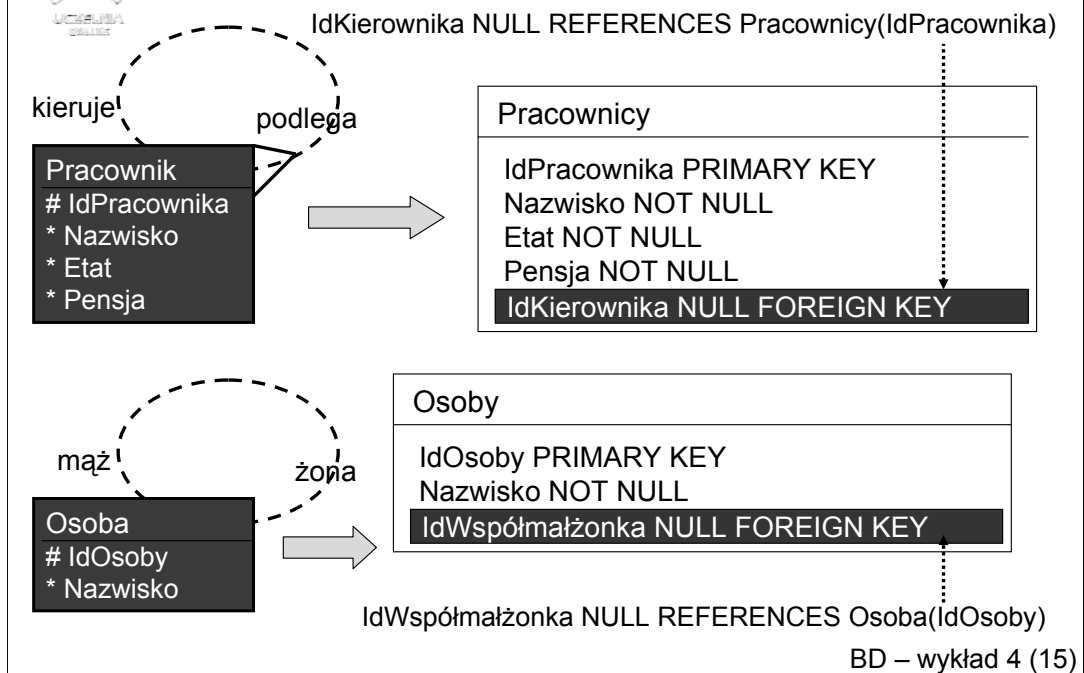
Reguły transformacji związku M:N są identyczne zarówno dla związków jednostronnie obowiązkowych, jak i obustronnie opcjonalnych. Reguły te są następujące.

1. Związek M:N jest reprezentowany w modelu relacyjnym poprzez dodatkową relację.
2. Nazwa relacji reprezentującej związek M:N jest połączeniem nazw relacji powstałych z encji połączonych tym związkiem.
3. Relacja dodatkowa zawiera klucze obce wskazujące na klucze podstawowe relacji powstałych z powiązanych encji.
4. Ograniczenia referencyjne są definiowane dla kluczy obcych.
5. Klucze obce tworzą klucz podstawowy relacji. W konsekwencji, ich wartości nigdy nie będą puste.



Poniższy przykład ilustruje sposób transformacji binarnego związku M:N jednostronnie obowiązkowego. Z encji Pracownik powstaje tabela Pracownicy, a z encji Projekt - tabela Projekty. Ze związku powstaje tabela pośrednia o nazwie Pracownicy\_Projekty. Zawiera ona dwa klucze obce. Jeden wskazuje na klucz podstawowy tabeli Pracownicy, czyli IdPracownika, a drugi - na klucz podstawowy tabeli Projekty, czyli NrProjektu. Oba klucze obce stanowią klucz podstawowy tabeli Pracownicy\_Projekty. Oznacza to, że ich wartości nigdy nie mogą być puste.

## Związek unarny (1)

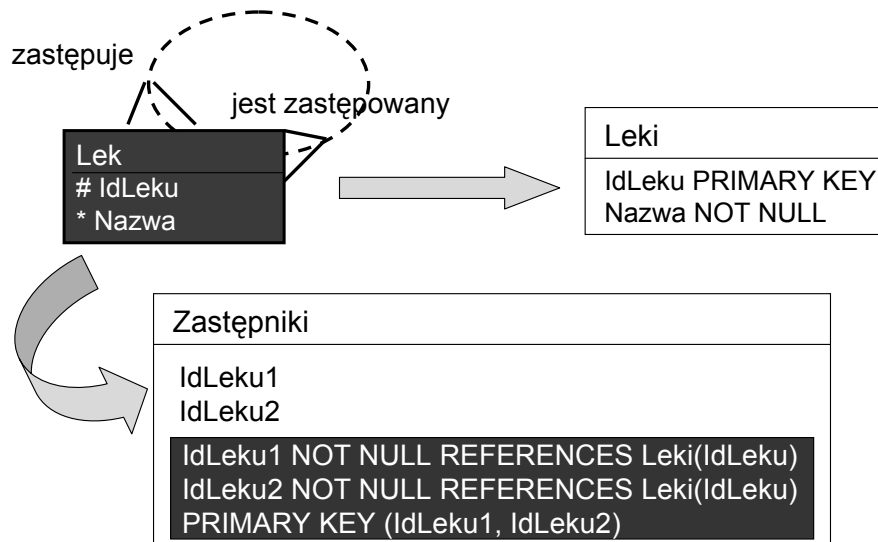


Związek unarny 1:1 lub 1:M obustronnie opcjonalny transformuje się do klucza obcego w tej samej tabeli. W pierwszym przykładzie ze slajdu, z encji Pracownik powstaje tabela Pracownicy. Zawiera ona klucz obcy IdKierownika wskazujący na IdPracownika w tej samej tabeli. Należy zwrócić uwagę, że wartość klucza obcego może być pusta ponieważ związek jest opcjonalny od strony "wiele".

W drugim przykładzie, w tabeli Osoby powstaje klucz obcy IdWspółmałżonka wskazujący na IdOsoby w tej samej tabeli. Ponieważ związek jest opcjonalny, więc klucz obcy może przyjmować wartości puste.



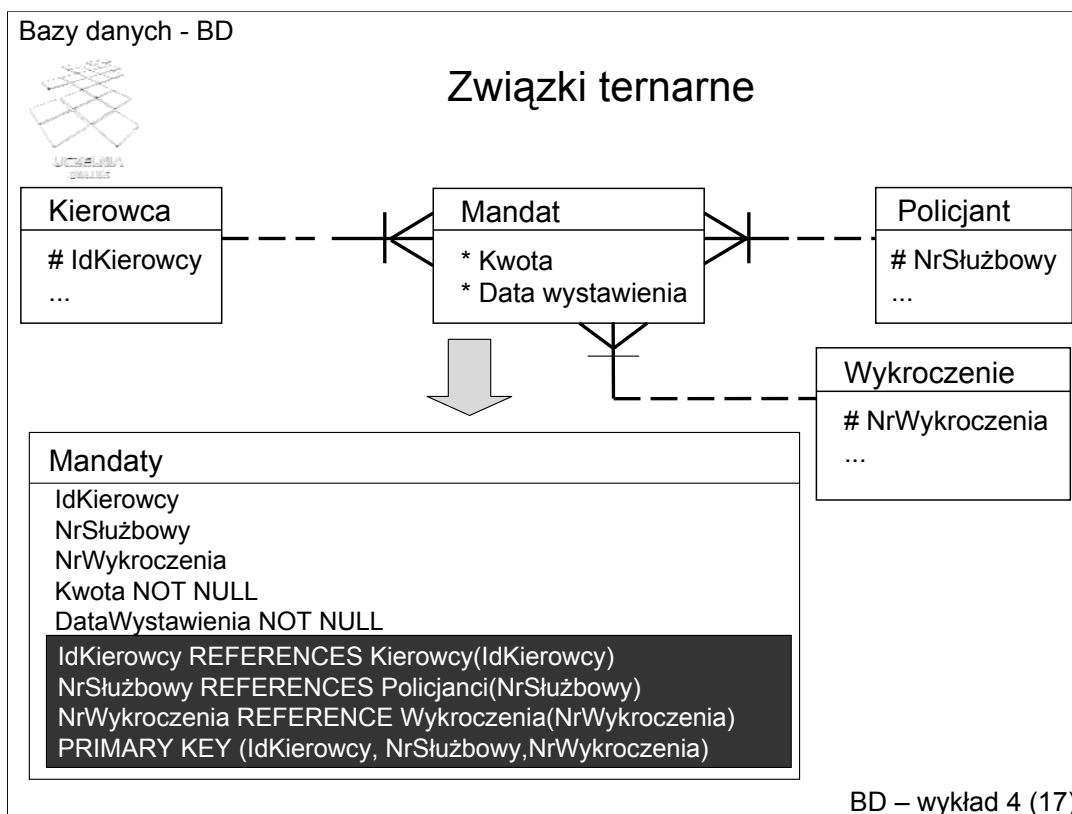
## Związek unarny (2)



BD – wykład 4 (16)

Związek unarny M:N obustronnie opcjonalny jest transformowany do tabeli pośredniej. W przykładzie ze slajdu, z encji **Lek** powstaje tabela **Leki**, a związek jest transformowany do tabeli o nazwie **Zastępniki**. Tabela ta posiada dwa klucze obce (atrybut **IdLeku1** i **IdLeku2**), oba wskazują na klucz podstawowy tabeli **Leki**, czyli na atrybut **IdLeku**. Oba klucze obce wchodzą w skład klucza podstawowego tabeli **Zastępniki**.





Związek ternarny transformuje się w sposób identyczny jak związek 1:M. W przykładzie ze slajdu, z encji Mandat powstaje tabela Mandaty. Zawiera ona 3 klucze obce: IdKierowcy, NrSłużbowy, NrWykroczenia wskazujące odpowiednio na: IdKierowcy w tabeli Kierowcy, NrSłużbowy w tabeli Policjanci, NrWykroczenia w tabeli Wykroczenia. Te trzy klucze obce wchodzą w skład klucza podstawowego tabeli Mandaty, ponieważ transformowana encja Mandat była słaba.



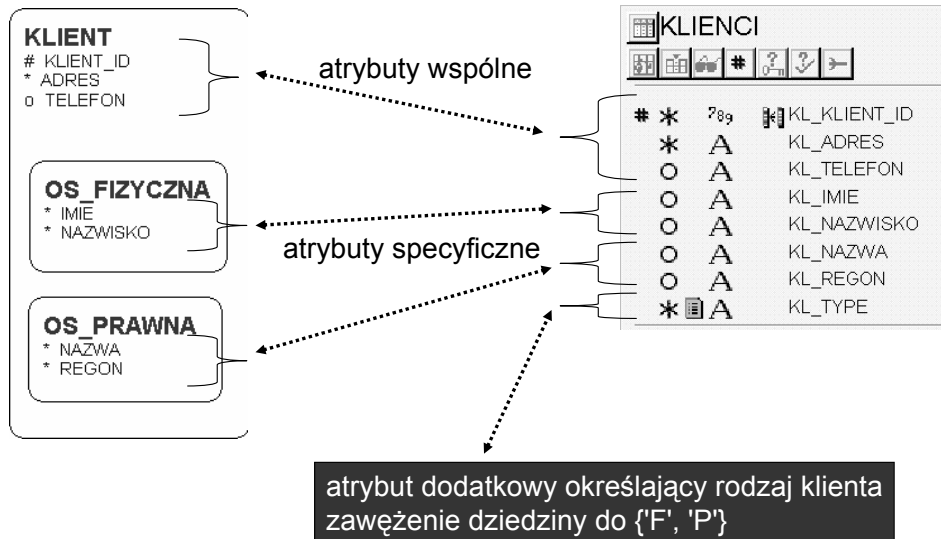
## Hierarchia encji

- Schemat 1: jedna wspólna tabela
- Schemat 2: dla każdej podencji tworzona tabela zawierająca atrybuty wspólne i specyficzne
- Schemat 3:
  - dla atrybutów wspólnych tworzona tabela wspólna
  - dla każdej podencji tworzona osobna tabela z kluczem i atrybutami specyficznymi
  - tabela wspólna i tabele powstałe z podencji powiązane ograniczeniami referencyjnymi

Hierarchię encji do modelu relacyjnego można przetransformować na 3 sposoby. Sposób (schemat) 1 polega na utworzeniu jednej tabeli ze wszystkimi atrybutami i kluczami obcymi, tj. wspólnymi i specyficznymi dla podencji. Sposób (schemat) 2 polega na utworzeniu osobnej tabeli dla każdej podencji. Każda z tabel zawiera atrybuty wspólne i specyficzne dla określonej encji. Sposób (schemat) 3 polega na utworzeniu osobnej tabeli na atrybuty wspólne i osobnej tabeli dla każdej podencji. Tabele powstałe z podencji zawierają klucz podstawowy i atrybuty specyficzne. Tabela wspólna i tabele powstałe z podencji są połączone ograniczeniami referencyjnymi.



## Transformacja hierarchii generalizacji - schemat 1

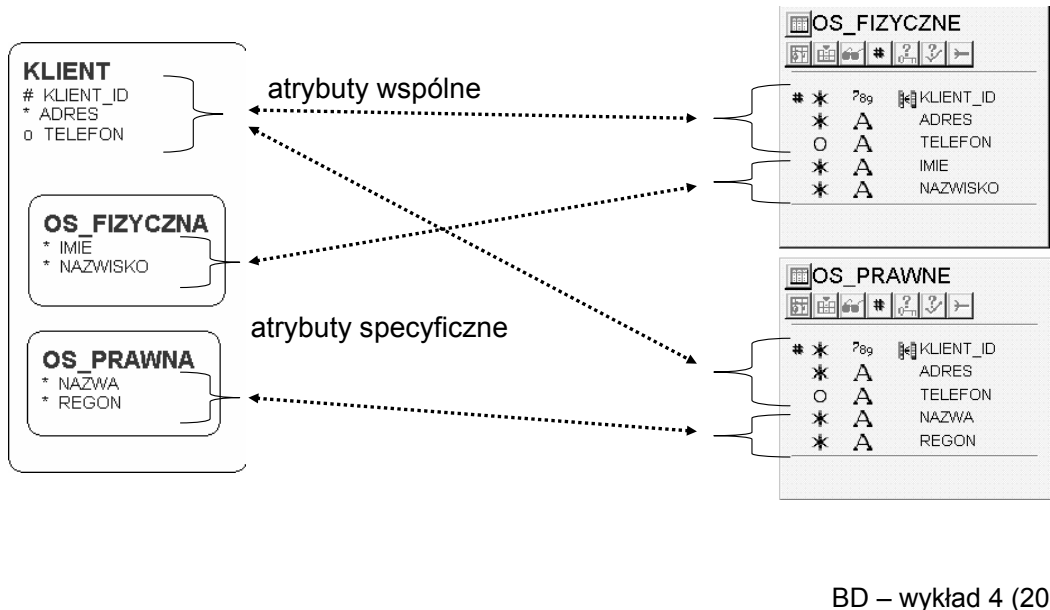


BD – wykład 4 (19)

Przykład ze slajdu ilustruje sposób transformacji hierarchii encji według schematu 1. Powstała tabela Klienci zawiera wszystkie atrybuty wspólne i wszystkie atrybuty specyficzne dla wszystkich podencji. Atrybuty specyficzne w tabeli mogą przyjmować wartości puste zawsze, niezależnie od ich definicji w pod-encjach. Wynika to z faktu, że rekord w tabeli klienci opisuje albo klienta fizycznego albo prawnego. Jeśli rekord opisuje klienta fizycznego, to atrybuty klienta prawnego pozostają puste i odwrotnie. Dodatkowo, w tabeli Klienci jest tworzony atrybut z wartością obowiązkową (KL\_TYPE), którego wartościami mogą być albo 'F' albo 'P'. Wartość tego atrybutu określa czy rekord opisuje klienta fizycznego ('F'), czy prawnego ('P'). Atrybut ten jest przydatny przy wyszukiwaniu klientów określonego typu.

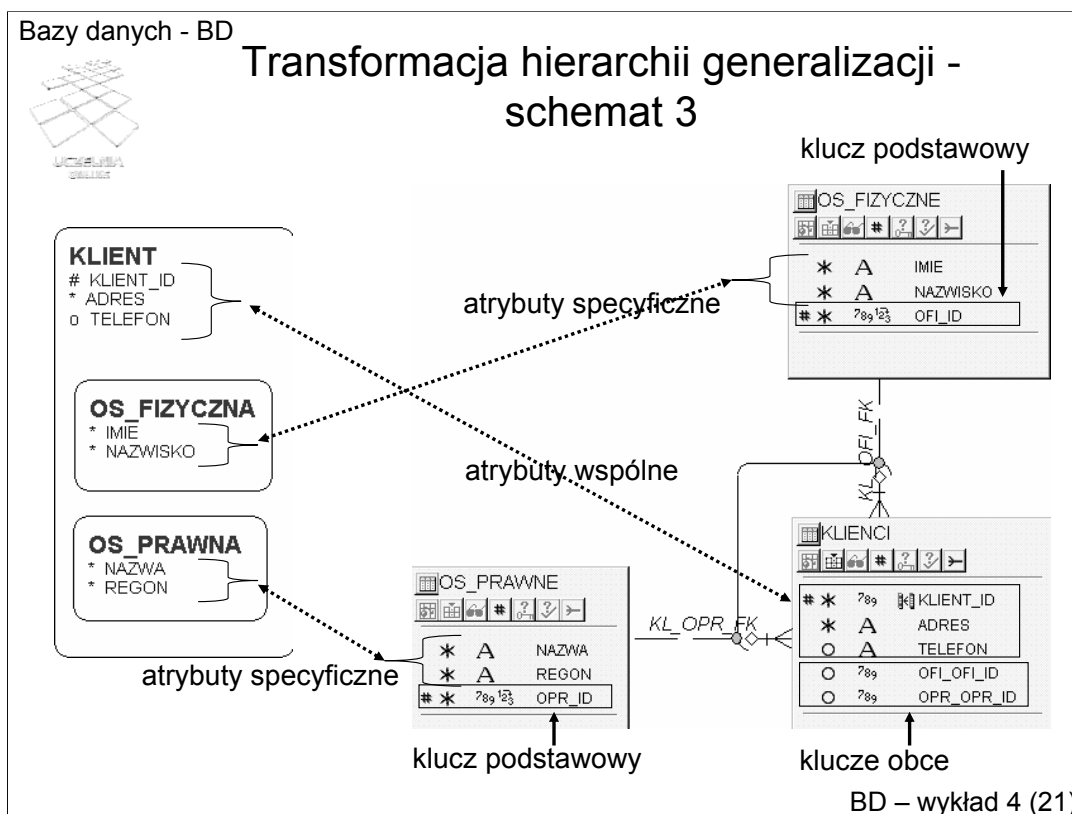


## Transformacja hierarchii generalizacji - schemat 2



BD – wykład 4 (20)

Przykład ze slajdu ilustruje sposób transformacji hierarchii encji według schematu 2. Z encji OS\_FIZYCZNA powstaje tabela OS\_FIZYCZNE. Zawiera ona wszystkie atrybuty wspólne encji Klient i wszystkie atrybuty specyficzne encji OS\_FIZYCZNA. Opcjonalność/obowiązkowość wartości atrybutów encji przenosi się bezpośrednio na opcjonalność/obowiązkowość atrybutów tabeli OS\_FIZYCZNE. W podobny sposób jest transformowana encja OS\_PRAWNA.

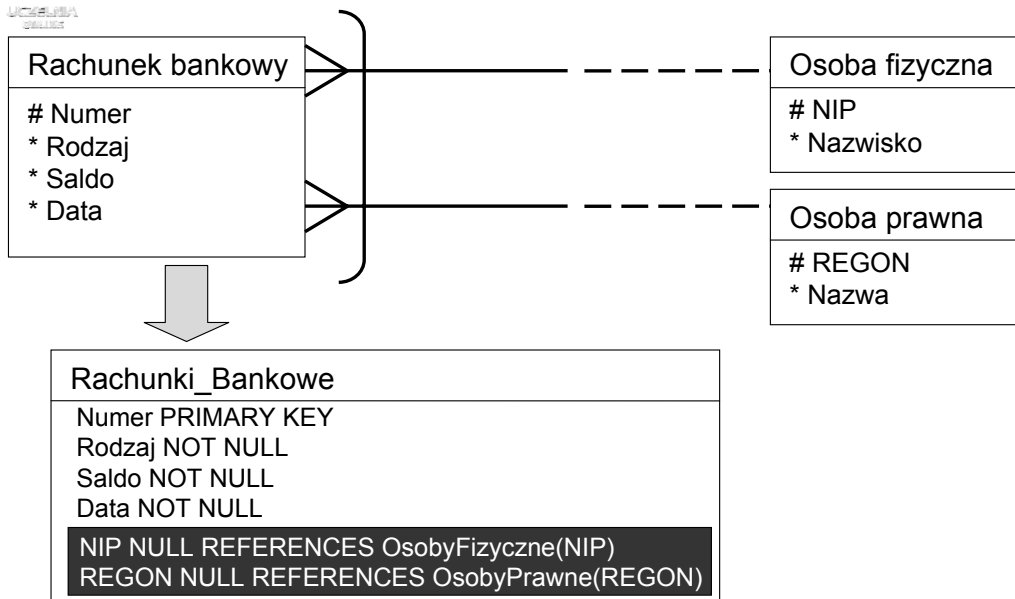


Przykład ze slajdu ilustruje sposób transformacji hierarchii encji według schematu 3. Z encji OS\_FIZYCZNA powstaje tabela OS\_FIZYCZNE, która zawiera wszystkie atrybuty specyficzne ze swojej encji i atrybut OFI\_ID, który jest kluczem podstawowym tabeli OS\_FIZYCZNE. Z encji OS\_PRAWNA powstaje tabela OS\_PRAWNE, która zawiera wszystkie atrybuty specyficzne ze swojej encji i atrybut OPR\_ID, który jest kluczem podstawowym tabeli OS\_PRAWNE.

Z atrybutów wspólnych encji Klient powstaje tabela KLIENCI. Dodatkowo, tabela ta posiada dwa klucze obce OFI\_OFI\_ID i OPR\_OPR\_ID, z wartościami opcjonalnymi. Pierwszy z nich wskazuje na klucz podstawowy tabeli OS\_FIZYCZNE, a drugi - na klucz podstawowy tabeli OS\_PRAWNE. Dla danego rekordu w tabeli KLIENCI, tylko jeden klucz obcy może przyjąć wartość, ponieważ rekord w tabeli KLIENCI opisuje albo osobę prawną albo fizyczną.



## Transformacja związków wyłącznych - schemat 1



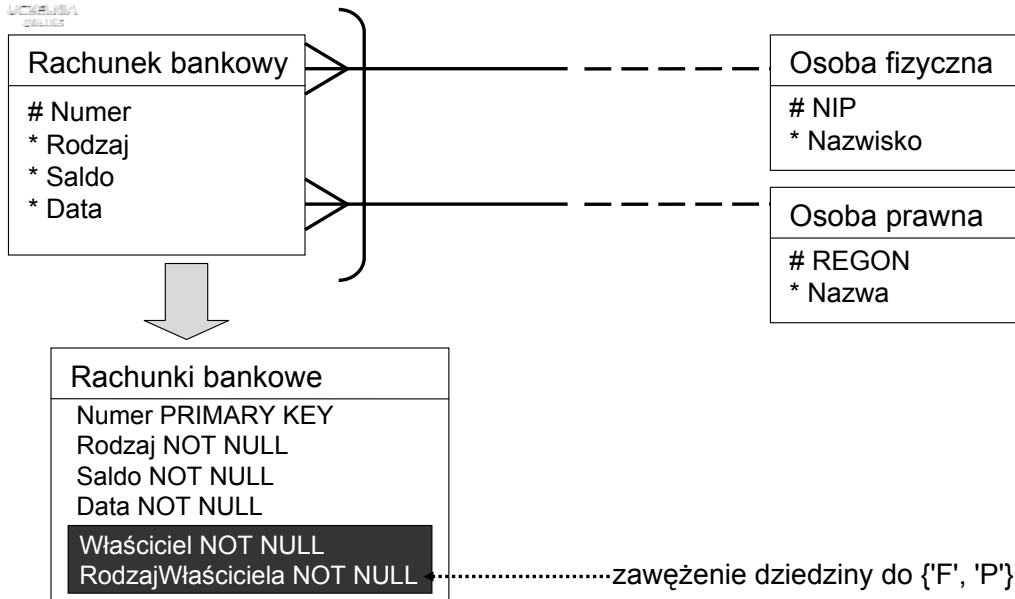
BD – wykład 4 (22)

Transformacja związków wyłącznych jest podobna do transformacji związku 1:M. Z tą tylko różnicą, że klucze obce mogą przyjmować wartości puste.

Jako przykład rozważmy encję Rachunek bankowy powiązaną związkami wyłącznymi z encją Osoba fizyczna i Osoba prawna. Z encji Rachunek bankowy powstaje tabela Rachunki\_Bankowe, która posiada dwa klucze obce, jeden wskazuje na klucz podstawowy tabeli Osoby\_Fizyczne, a drugi - na klucz podstawowy tabeli Osoby\_Prawne. Oba klucze obce mogą przyjmować wartości puste, pomimo, że związki z których powstały są obowiązkowe od strony "wiele". Dzieje się tak dla tego, że dany rekord rachunku bankowego jest albo związany z osobą fizyczną albo z osobą prawną, więc tylko jeden klucz obcy przyjmie wartość.



## Transformacja związków wyłącznych - schemat 2



Alternatywny sposób transformacji związków wyłącznych przedstawiono na slajdzie. Różni się on od poprzedniego tym, że w tabeli *Rachunki\_Bankowe* jest atrybut *Właściciel*, który może przyjąć wartość klucza podstawowego rekordu albo z tabeli *Osoby\_Ficzyzne* albo z tabeli *Osoby\_Prawne*; dla atrybutu tego zdefiniowanie ograniczenia referencyjnego jest niemożliwe. Ponadto, w tabeli *Rachunki\_Bankowe* jest atrybut *RodzajWłaściciela*, który może przyjąć jedną z dwóch wartości 'F' lub 'P'. Służy on do określania rodzaju osoby, na którą wskazuje wartość atrybutu *Właściciel*. Oba atrybuty nie mogą przyjmować wartości pustych.