



# Bazy Danych

---

## Model Relacyjny

Krzysztof Regulski

WIMiP, KISiM,  
regulski@agh.edu.pl  
B5, pok. 408

---

# Relacyjny model danych

---

- Relacyjny model danych jest obecnie najbardziej popularnym modelem używanym w systemach baz danych. Podstawą tego modelu stała się praca opublikowana przez E.F. Codd'a w 1970r.
- Zauważył on, że zastosowanie struktur i procesów matematycznych w zarządzaniu danymi mogłoby rozwiązać wiele problemów trapiących ówczesne modele.
- W swej pracy ***A relational model for large shared data banks*** Codd zaprezentował założenia relacyjnego modelu baz danych, model ten oparł na teorii mnogości i rachunku predykatów pierwszego rzędu.

- W roku 1990 Codd opublikował artykuł „Relacyjny model zarządzania bazami danych: wersja 2”, rozszerzający poprzednie prace
- RMD oparty jest o algebrę relacji
- Podstawowe elementy modelu to
  - » relacje
  - » więzi

# Podstawowe pojęcia w bazach danych:

---

- **encja** – relacja – klasa – tabela
  - » zbiór podobnych obiektów opisanych w jednolity sposób
- **krotka** – obiekt (instancja klasy) – rekord – wiersz
  - » zestaw wartości atrybutów opisujących jeden obiekt identyfikowany przez wyróżnione atrybuty lub nazwę
- **związek** – więź – asocjacja
  - » związek pomiędzy dwoma encjami (klasami) pokazujący jakie rekordy (obiekty) z jednej encji odpowiadają rekordom z drugiej i jaki jest charakter tej odpowiedniości
- **atrybut** – kolumna – pole
  - » pojedyncza dana wchodząca w skład krotki np. nazwisko studenta, nr ewidencyjny pracownika, wielkość zapasu czy rodzaj filmu.

# Relacyjny model danych

- Podstawową strukturą danych jest **relacja** będąca podzbiorem iloczynu kartezyjskiego dwóch wybranych zbiorów reprezentujących dopuszczalne wartości.

Niech  $A_1 = [a,b,c]$ ,  $A_2 = [x,y]$

Wtedy  $A_1 \times A_2 = \{(a,x), (a,y), (b,x), (b,y), (c,x), (c,y)\}$

Przykłady relacji, które są podzbiorem iloczynu kartezyjskiego  $A_1 \times$

$A_2$  :

$X = \{(a,x), (b,x), (c,x)\}$

$Y = \{(a,x), (a,y), (b,y)\}$

Analogicznie jak dla iloczynu kartezyjskiego elementy relacji są nazywane krotkami

- Relacja jest zbiorem krotek posiadających taką samą strukturę, lecz różne wartości.
- Każda krotka posiada co najmniej jeden atrybut.

- Każda relacja posiada następujące własności
  - » krotki są unikalne
  - » atrybuty są unikalne
  - » kolejność krotek nie ma znaczenia
  - » kolejność atrybutów nie ma znaczenia
  - » wartości atrybutów są atomowe
  
- Tworzenie modeli relacyjnych nazywane jest **modelowaniem związków encji**.

# Postulaty Codd'a

## Postulaty Codd'a (1)

---

1. **Postulat informacyjny** – dane są reprezentowane jedynie poprzez wartości atrybutów w wierszach tabel,
2. **Postulat (gwarantowanego) dostępu** – każda wartość w bazie danych jest dostępna poprzez podanie nazwy tabeli, atrybutu oraz wartości klucza podstawowego,
3. **Postulat dotyczący wartości NULL** – dostępna jest specjalna wartość NULL dla reprezentacji wartości nieokreślonej jak i nieadekwatnej, inna od wszystkich i podlegająca przetwarzaniu,



## Postulaty Codd'a (2)

---

4. **Postulat dotyczący katalogu** – informacje o obiektach bazy danych tworzących schemat bazy danych są na poziomie logicznym zgrupowane w tabele i dostępne w taki sam sposób jak każde inne dane.
  
5. **Postulat języka danych** – system musi dostarczać pełnego języka przetwarzania danych, który:
  - » charakteryzuje się liniową składnią,
  - » może być używany zarówno w trybie interaktywnym, jak i w obrębie programów aplikacyjnych,
  - » obsługuje operacje definiowania danych (łącznie z definiowaniem perspektyw), operacje manipulowania danymi (aktualizację, także wyszukiwanie), ograniczenia związane z bezpieczeństwem i integralnością oraz operacje zarządzania transakcjami (rozpoczynanie, zapis zmian i ponowny przebieg)

## Postulaty Codd'a (3)

---

6. **Postulat modyfikowalności perspektyw** – system musi umożliwiać modyfikowanie perspektyw, o ile jest ono (modyfikowanie) semantycznie realizowane,
7. **Postulat modyfikowalności danych** – system musi umożliwiać operacje modyfikacji danych, musi obsługiwać operatory INSERT, UPDATE oraz DELETE,
8. **Postulat fizycznej niezależności danych** – zmiany fizycznej reprezentacji danych i organizacji dostępu nie wpływają na aplikacje,

9. **Postulat logicznej niezależności danych** – zmiany wartości w tabelach nie wpływają na aplikacje,
10. **Postulat niezależności więzów spójności** (*niezależność integralnościowa*) – więzy spójności są definiowane w bazie i nie zależą od aplikacji,
11. **Postulat niezależności dystrybucyjnej** – działanie aplikacji nie zależy od modyfikacji i dystrybucji bazy,

## Postulaty Codd'a (5)

---

11. **Postulat bezpieczeństwa względem operacji niskiego poziomu** – operacje niskiego poziomu nie mogą naruszać modelu relacyjnego i więzów spójności.
12. **Reguła nieprowadzenia "działalności wywrotowej"**: jeśli system jest wyposażony w interfejs niskiego poziomu (operacje na pojedynczych rekordach), nie może być użyty do prowadzenia działalności wywrotowej (np. omijania zabezpieczeń relacyjnych lub ograniczeń integralnościowych)

## Postulaty Codd'a (6)

---

- Codd określił również 9 cech strukturalnych, 3 cechy integralnościowe oraz 18 cech manipulacyjnych, które także są wymagane. Codd swą listę 12 reguł rozszerzył do 333 w drugiej wersji modelu relacyjnego.

Rozważmy relację, której **atrybutami** są nazwisko, imię, wiek.  
Relację tę można zapisać następująco:

PRAC <nazwisko, imię, wiek>,

gdzie PRAC jest nazwą danej **relacji**.

A oto trzy **krotki** relacji PRAC:

<Kowalski, Jan, 36>

<Tomaszewski, Wojciech, 40>

<Wiśniewski, Marek, 50>.

$$R = \{A_1, A_2, \dots, A_n\}$$

zbiór atrybutów

$$S \subseteq R$$

który jednoznacznie identyfikuje wszystkie krotki w relacji  $R$   
w żadnej relacji o schemacie  $R$  **nie mogą** istnieć dwie krotki  
 $t_1$  i  $t_2$  takie, że

$$t_1[S] = t_2[S]$$

- Minimalny **zbiór identyfikujący**
- Taki zbiór atrybutów relacji, których kombinacje wartości **jednoznacznie identyfikują** każdą krotkę tej relacji a **żaden podzbiór** tego zbioru nie posiada tej własności
- W kluczu nie może zawierać się wartość Null



- Klucz jest **kluczem prostym**, jeżeli powyżej opisany zbiór jest jednoelementowy - w przeciwnym razie mówimy o **kluczu złożonym**
- W ogólności, w relacji można wyróżnić wiele kluczy, które nazywamy **kluczami potencjalnymi**. Wybrany klucz spośród kluczy potencjalnych nazywamy **kluczem głównym (Primary Key PK)**

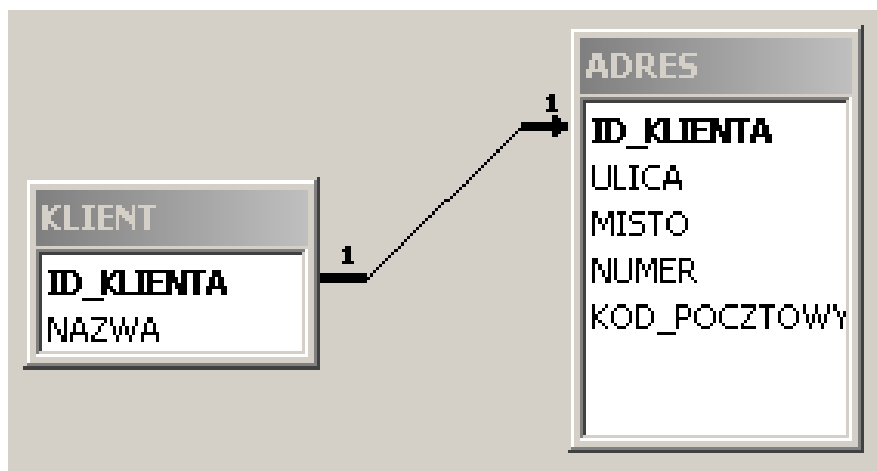
- Atrybut ***B*** relacji ***R*** jest funkcjonalnie zależny od atrybutu ***A*** jeżeli dowolnej wartości ***a*** atrybutu ***A*** odpowiada nie więcej niż jedna wartość ***b*** atrybutu ***B***

$$A \rightarrow B$$

- **Więź** (*relationship*) to powiązanie pomiędzy parą tabel (relacji).
- Istnieje ona wtedy, gdy dwie tabele są połączone przez **klucz podstawowy i klucz obcy**. Każda więź jest opisywana przez typ więzi istniejący między dwoma tabelami, typ uczestnictwa oraz stopień uczestnictwa tych tabel.

## Typy więzi

- **jeden-do-jednego** (jeżeli pojedynczemu rekordowi z pierwszej tabeli przyporządkowany jest najwyżej jeden rekord z drugiej tabeli i na odwrót)



**Edytowanie relacji** [?] [X]

Tabela/Kwerenda:  Pokrewna tabela/kwerenda:

<input type="text" value="ID_KLIENTA"/>	<input type="text" value="ID_KLIENTA"/>

Wymuszaj więzy integralności  
 Kaskadowo aktualizuj pola pokrewne  
 Kaskadowo usuń rekordy pokrewne

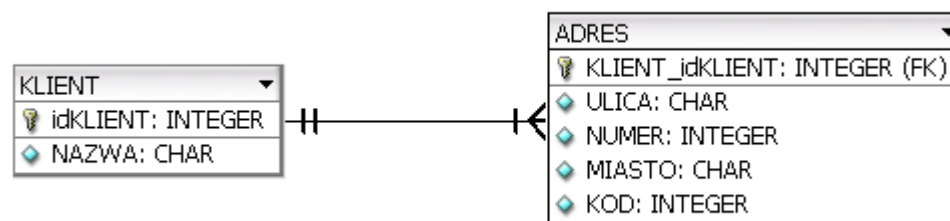
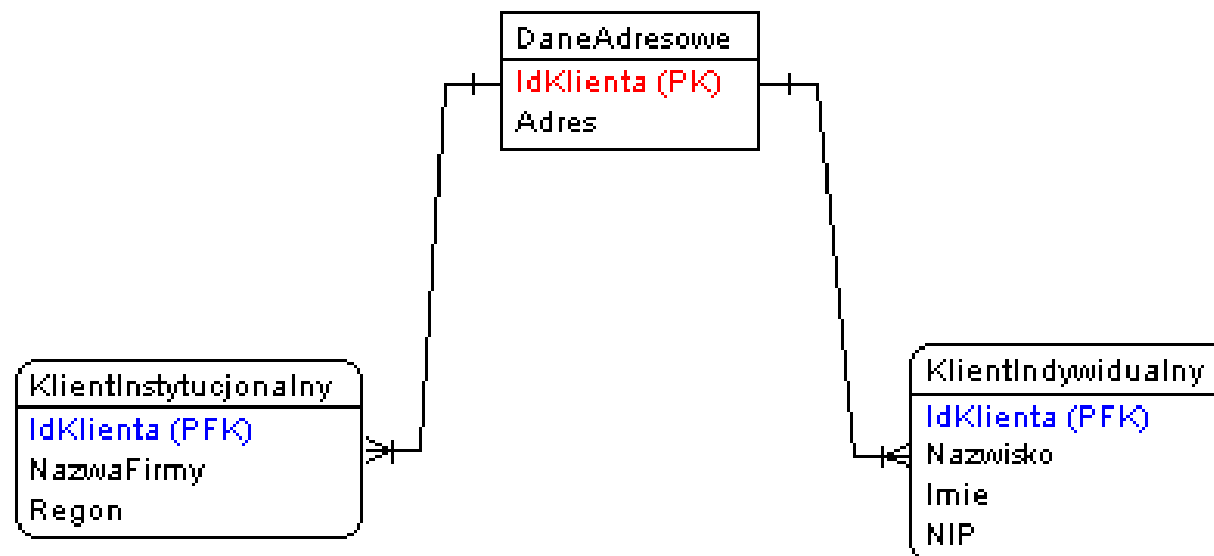
Typ relacji:

**Właściwości sprzężenia** [?] [X]

1: Zawiera tylko te wiersze, w których sprzężone pola z obu tabel są równe.  
 2: Uwzględnia **WSZYSTKIE** rekordy z 'KLIENT' i tylko te rekordy z 'ADRES', dla których sprzężone pola są równe.  
 3: Uwzględnia **WSZYSTKIE** rekordy z 'ADRES' i tylko te rekordy z 'KLIENT', dla których sprzężone pola są równe.

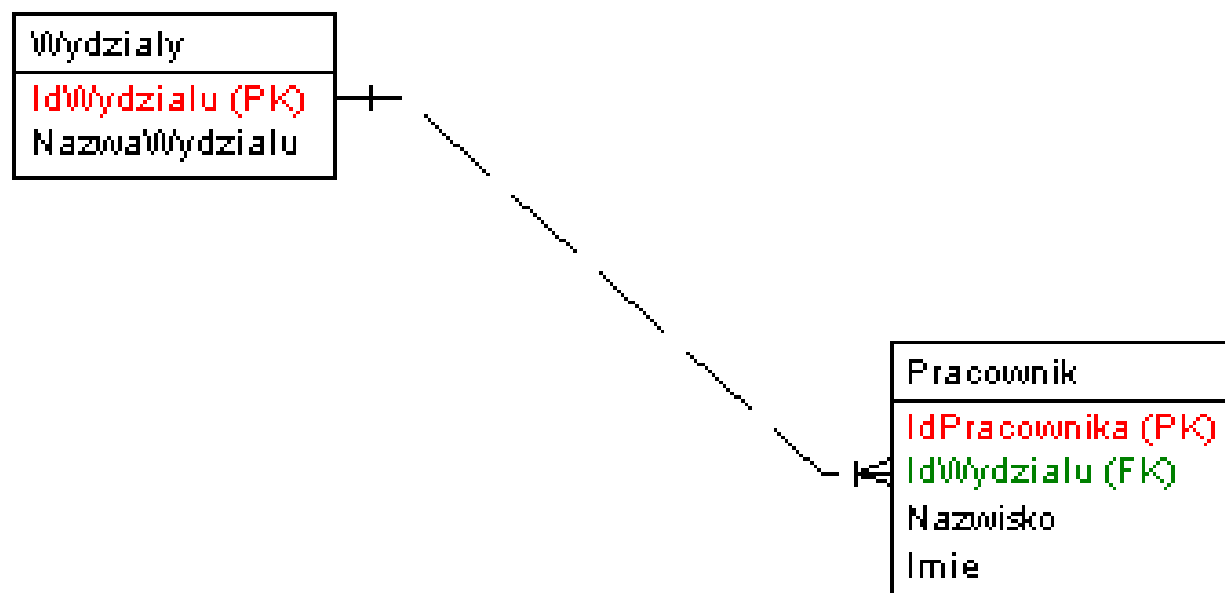
# Więź jeden-do-jednego

[1.1]



## Typy więzi

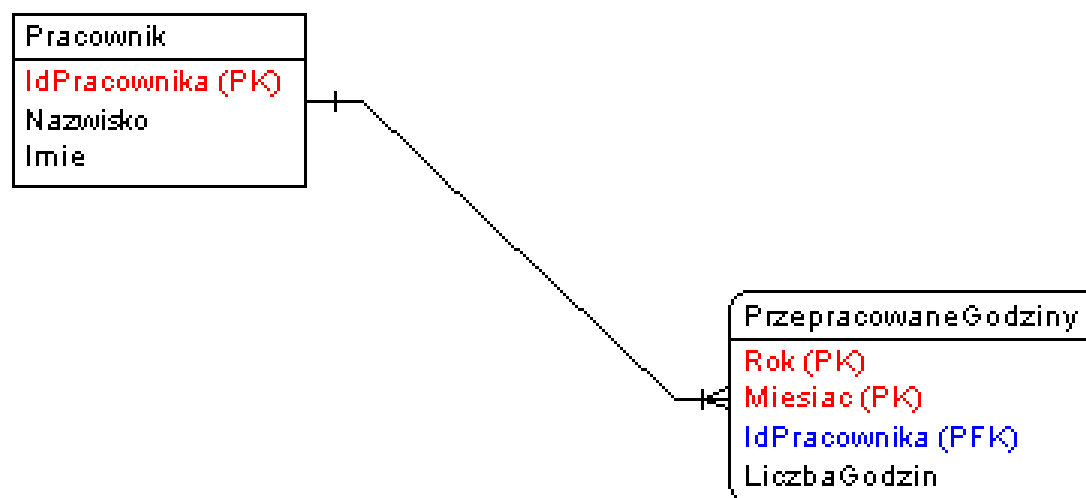
- **jeden-do-wielu** (jeżeli pojedynczemu rekordowi z pierwszej tabeli może odpowiadać jeden lub więcej rekordów z drugiej, ale pojedynczemu rekordowi z drugiej tabeli odpowiada najwyżej jeden rekord z tabeli pierwszej)



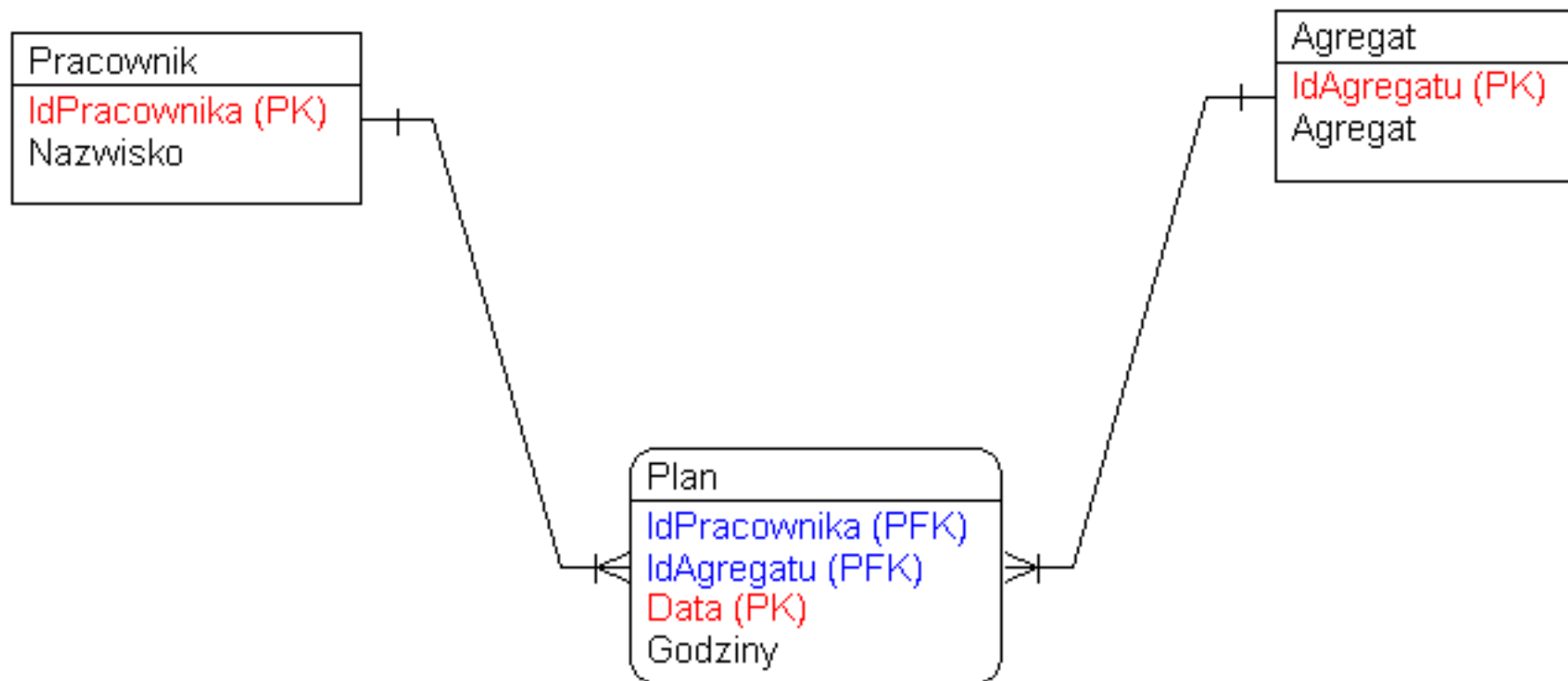
## Więzi identyfikujące

- **Klucz obcy**, który jest składnikiem złożonego klucza głównego w relacji zależnej określany jest mianem **klucza obcego głównego (Primary Foreign Key)** a tak zbudowana więź jest **więzią identyfikującą**

[1.1]



# Więź wiele-do-wielu





# Podstawowe zagadnienia algebry relacji

## Relacja (przypomnienie):

---

- Dane są zbiory  $A_1, A_2, \dots, A_n$ , relacją  $r$  nazywamy dowolny podzbiór  $A_1 \times A_2 \times \dots \times A_n$
- Relacja jest zbiorem krotek  $(a_1, a_2, \dots, a_n)$ , gdzie każde  $a_i \in A_i$

» Np.

*customer-name* = {Jones, Smith, Curry, Lindsay}

*customer-street* = {Main, North, Park}

*customer-city* = {Harrison, Rye, Pittsfield}

$r = \{(Jones, Main, Harrison), (Smith, North, Rye),$   
 $(Curry, North, Rye), (Lindsay, Park, Pittsfield)\}$

jest relacją określoną na

*customer-name*  $\times$  *customer-street*  $\times$  *customer-city*

## Model relacyjny danych:

---

- $A_1, A_2, \dots, A_n$  oznaczają atrybuty.
- $R = (A_1, A_2, \dots, A_n)$  jest schematem relacji  $R$ .
  - » Np.  
$$\text{Customer-schema} = (\text{customer-name}, \text{customer-street}, \text{customer-city})$$
- $r(R)$  oznacza instancję  $r$  relacji o schemacie  $R$ .
  - » Np.  
$$\text{customer} (\text{Customer-schema})$$
- $t = (a_1, a_2, \dots, a_n), t \in r$  oznacza krotkę relacji  $r(R)$ .

## Model relacyjny danych (c.d.):

- Aktualna wartość relacji (**instancja relacji**) może być przedstawiona w formie tabeli, której:
  - » kolumny odpowiadają **atrybutom**,
  - » nagłówek odpowiada **schematowi relacji**.
- Elementy relacji - **krotki** są reprezentowane przez wiersze tabeli.

<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
<i>Jones</i>	Main	Harrison
<i>Smith</i>	North	Rye
<i>Curry</i>	North	Rye
<i>Lindsay</i>	Park	Pittsfield

*customer*

## Kategorie w algebrze relacji:

---

- Zwyczajne działania algebry zbiorów: **suma**, **przecięcie** i **różnica**
- Operacje zawężenia: **selekcja** eliminuje pewne wiersze, a **rzutowanie** pewne kolumny
- Operacje komponowania krotek z różnych relacji: np. **iloczyn kartezyjski**
- Operacje **przemianowania** nie zmieniające krotek ale schemat ich relacji

## Iloczyn kartezjański:

- (inaczej produkt) relacji  $R$  i  $S$  to relacja wszystkich uporządkowanych par krotek, z których pierwszy element pary należy do relacji  $R$  a drugi do  $S$
- Schemat relacji  $R \times S$  jest sumą schematów relacji  $R$  i  $S$ , w której powtarzające się atrybuty (kolumny) traktowane są jako odrębne elementy schematu, np.  $R.A$  i  $S.A$

**R:**

<i>Student</i>	<i>Język</i>
Adam Kot	angielski
Adam Kot	niemiecki

**S:**

<i>Student</i>	<i>Język</i>
Adam Kot	matematyka
Adam Kot	fizyka

**$R \times S$ :**

<i>R.Student</i>	<i>Język</i>	<i>S.Student</i>	<i>Przedmiot</i>
Adam Kot	angielski	Adam Kot	matematyka
Adam Kot	angielski	Adam Kot	fizyka
Adam Kot	niemiecki	Adam Kot	matematyka
Adam Kot	niemiecki	Adam Kot	fizyka

## Złączenie naturalne:

- polega na połączeniu w pary tych krotek z relacji **R** i **S**, które mają identyczne wartości dla wszystkich wspólnych atrybutów i jest oznaczane  $R \bowtie S$
- w rezultacie powstaje relacja, której schemat zawiera atrybuty relacji **R** i relacji **S**, przy czym wspólna część uwzględniana jest tylko raz

<i>Student</i>	<i>Przedmiot</i>	<i>Semestr</i>	<i>Ocena</i>
Adam Kot	Matematyka	I	3,0
Adam Kot	Fizyka	II	4,0
Jan Pies	Matematyka	I	2,0

<i>Przedmiot</i>	<i>Semestr</i>	<i>Prowadzący</i>
Matematyka	I	Prof. Wilk
Fizyka	II	Prof. Zajac
Matematyka	II	Prof. Kos

<i>Student</i>	<i>Przedmiot</i>	<i>Semestr</i>	<i>Ocena</i>	<i>Prowadzący</i>
Adam Kot	Matematyka	I	3,0	Prof. Wilk
Adam Kot	Fizyka	II	4,0	Prof. Zajac
Jan Pies	Matematyka	I	2,0	Prof. Wilk

## Typy złączeń:

---

- **złączenie wewnętrzne** (inner join) – w relacji wynikowej występują wyłącznie te krotki, które spełniają warunek złączenia
- **złączenie lewostronne zewnętrzne** (left outer join) – zawiera wszystkie krotki **R** uzupełnione krotkami **S** spełniającymi warunek
- **złączenie prawostronne zewnętrzne** (right outer join) - zawiera wszystkie krotki **S** uzupełnione krotkami **R** spełniającymi warunek
- **złączenie zewnętrzne pełne** (full outer join) – zawiera wszystkie krotki **R** oraz **S** uzupełnione wartościami typu **NULL** gdy do danej krotki nie pasuje żadna krotka z drugiej relacji
- **złączenie zewnętrzne typu union** - zawiera wszystkie krotki **R** nie pasujące do żadnej krotki **S** uzupełnione krotkami **S** nie pasującymi do żadnej krotki **R**



- polega na złączeniu dwóch relacji  $R$  i  $S$  w iloczyn kartezyjański i wyborze z niego tych krotek, które spełniają wyrażenie warunkowe na parze lub zbiorze par atrybutów z  $R$  i  $S$  i jest oznaczane symbolem  $R \bowtie_{\Theta} R$  lub  $R \bowtie_C S$ , gdzie  $\Theta$  lub  $C$  to wyrażenia logiczne

## Złączenie teta

***R***

Towar	Data_Od	Data_do	Cena
mąka	1.01.2004	31.01.2004	2,00
mąka	1.02.2004	31.03.2004	2,10
mąka	1.04.2004		2,05

***S***

Towar	Data	Ilość
mąka	15.03.2004	10

***R***  $\bowtie_C$  ***S***

R.Towar	Data_Od	Data_do	Cena	S.Towar	Data	Ilość
mąka	1.02.2004	31.03.2004	2,10	mąka	15.03.2004	10

***C = (R.Towar = S.Towar AND Data >= Data\_Od AND Data <= Data\_Do )***

- to szczególny przypadek złączenia teta, w którym warunek ma charakter równości wybranych atrybutów obu relacji
- powtarzające się kolumny opisujące atrybuty z warunku złączenia są pomijane

# Równozłączenie

***R***

Towar	Klient
stal	Exbud
cegła	PBS
złom	

***S***

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

***R*** ⋈ *R.Klient=S.Kontrahent* ***S***

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków

# Złączenie lewostronne zewnętrzne

***R***

Towar	Klient
stal	Exbud
cegła	PBS
złom	

***S***

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

***R*** ⋈ *R.Klient=S.Kontrahent* ***S***

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków
złom		

# Złączenie prawostronne zewnętrzne

***R***

Towar	Klient
stal	Exbud
cegła	PBS
złom	

***S***

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

***R*** ⋈ *R.Klient=S.Kontrahent* ***S***

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków
	PHS	Tarnów

# Złączenie zewnętrzne pełne

***R***

Towar	Klient
stal	Exbud
cegła	PBS
żłom	

***S***

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

***R*** ⋈ ***R.Klient=S.Kontrahent*** ***S***

Towar	Klient	Miasto
stal	Exbud	Kielce
cegła	PBS	Kraków
żłom		
	PHS	Tarnów

# Złączenie zewnętrzne typu union

***R***

Towar	Klient
stal	Exbud
cegła	PBS
złom	

***S***

Kontrahent	Miasto
Exbud	Kielce
PBS	Kraków
PHS	Tarnów

***R*** ⋈ *R.Klient=S.Kontrahent* ***S***

Towar	Klient	Miasto
złom		
	PHS	Tarnów