

# Bazy danych

## Wykład 2\_2

**Temat:** Integralność relacyjnej bazy danych

**Sławomir Świętoniowski**

slawomir-swietoniowski@wp.pl

# Plan wykładu

---

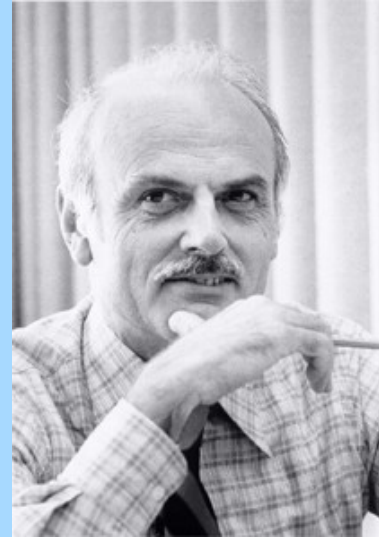
- 1. Założenia relacyjnego modelu danych - przypomnienie.**
2. Definicja podstawowych pojęć.
3. Normalizacja bazy danych: 1NF, 2NF, 3NF.

# Model relacyjny - historia

Główny twórca:

**Edgar F. Codd**

(23 VIII 1923 - 18 IV 2003)



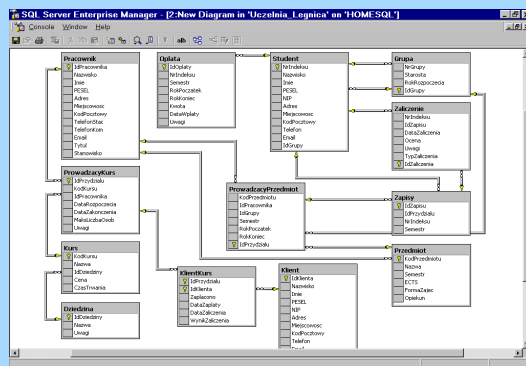
Źródło: Wikipedia (<http://en.wikipedia.org>).

Ważniejsze publikacje:

- "A Relational Model of Data for Large Shared Data Banks", 1970, CACM 13, No. 6.
- „Extending the Relational Database Model to Capture More Meaning”, ACM Transactions on Database Systems, 1979, Vol. 4, No. 4, pp. 397-434.
- „Relational Database: A Practical Foundation for Productivity”, Communications of ACM, 1982, Vol. 25, No. 2.
- „The Relational Model for Database Management: Version 2”, Reading, Mass., Addison-Wesley, 1990.

# Model relacyjny – podstawowe założenia

- Każda tabela w bazie danych ma jednoznaczną nazwę.
- Każda kolumna ma jednoznaczną nazwę w ramach jednej tabeli.
- Wszystkie wartości kolumny muszą być tego samego typu – zdefiniowane na tej samej dziedzinie
- Porządek kolumn w tabeli nie jest istotny.
- Każdy wiersz w tabeli musi być różny – powtórzenia nie są dozwolone.
- Porządek wierszy nie jest istotny.
- Każda wartość pola tabeli (na przecięciu kolumna/wiersz) powinna być atomowa – nie może być ciągiem, ani zbiorem.



2:Data in Table 'Pracownik' in 'Uczelnia\_Wroclaw' on 'HOMESQL'

	IdPracownika	Nazwisko	Imie	NIP	PESEL	Adres	Miejscowosc
1		Nowakowski	Andrzej	612-412-54-64	57121943212	ul. Świerkowa 6	Wrocław
2		Kowalski	Jan	663-654-76-87	72013142337	ul. Strzelecka 15/8	Wrocław
3		Janicki	Bogdan	432-543-654-6	49042343259	ul. Pastelowa 58/9	Wrocław
4		Marcinkowski	Piotr	789-098-23-45	76110309878	ul. Lakiernicza 98A	Wrocław
5		Andrzejewski	Grzegorz	980-432-23-12	52082898152	ul. Nobla 8/3	Wrocław
6		Piotrowski	Bartłomiej	780-678-66-11	65102189131	ul. Nadrzeczna 16/	Legnica
7		Bogdanska	Ewa	430-543-55-22	79031487924	ul. Wrocławska 23/	Legnica
8		Grzegorzewski	Paweł	250-532-99-90	76061898778	ul. Karkonoska 58	Wrocław
9		Styczen	Tomasz	610-220-96-52	68071678156	ul. Janowska 43/63	Legnica
10		Romanowski	Janusz	616-420-90-19	74090209835	ul. Góralska 43	Legnica

# Plan wykładu

---

1. Założenia relacyjnego modelu danych - przypomnienie.

**2. Definicja podstawowych pojęć.**

3. Normalizacja bazy danych: 1NF, 2NF, 3NF.

# Def. 1 Relacja matematyczna

Niech dane będą zbiory  $D_1, D_2, \dots, D_n$ .

**Relacją matematyczną**  $R$  nad tymi zbiorami nazywamy dowolny podzbiór iloczynu kartezyjskiego nad tymi zbiorami, tzn.

$$R \subseteq D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) : d_i \in D_i, i=1, 2, \dots, n\}.$$

Przykład:

$R: >$  (relacja większości)

$$D_1 = \{3, 4\}$$

$$D_2 = \{1, 2, 3\}$$

$$D_1 \times D_2 = \{(3, 1), (3, 2), (3, 3), (4, 1), (4, 2), (4, 3)\}$$

$$R \subseteq D_1 \times D_2 = \{(3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$$

Na podstawie: PANKOWSKI T., *Podstawy baz danych*, Wydawnictwo Naukowe PWN, Warszawa, 1992.

# Def. 2. Wiersz

---

Zbiór atrybutów (kolumn):  $U = \{A_1, A_2, \dots\}$ . Dla każdego  $A \in U$ ,  $DOM(A)$  jest zbiorem wartości **dziedzina** (domeną) atrybutu  $A$ .

**Wierszem** typu  $U$  nazywamy dowolną funkcję:

$$f: U \rightarrow \cup \{DOM(A): A \in U\}$$

taką, że dla dowolnego  $A \in U$ ,  $f(A) \in DOM(A)$ .

Zbiór wszystkich wierszy typu  $U$  oznaczamy jako:  $WIERSZ(U)$ .

Na podstawie: PANKOWSKI T., *Podstawy baz danych*, Wydawnictwo Naukowe PWN, Warszawa, 1992.

# Wiersz - przykład

---

Zbiór atrybutów:  $U = \{imię, nazwisko, wiek\}$ .

$DOM(imię) = \{Jan, Andrzej\}$

$DOM(nazwisko) = \{Nowak, Kowalski, Jabłoński\}$

$DOM(wiek) = \mathbb{N} \cap [1; 130]$

Przykładowy wiersz typu  $U$ :

$r(U) = \{(imię, Andrzej), (nazwisko, Nowak), (wiek, 22)\}$

$WIERSZ(U) = \{$

$\{(imię, Jan), (nazwisko, Nowak), (wiek, 1)\},$

$\{(imię, Jan), (nazwisko, Nowak), (wiek, 2)\},$

$\dots,$

$\{(imię, Andrzej), (nazwisko, Jabłoński), (wiek, 130)\} \}$



# Def. 3. Relacja (tabela)

**Relacją** (ang. *relation*) typu  $U$  nazywamy dowolny, skończony podzbiór zbioru  $WIERSZ(U)$ .

Zbiór wszystkich relacji (tabel) typu  $U$  oznaczamy:  $REL(U)$ .

Oznaczenia:

- Relacje typu  $U$ :  $R(U)$ ,  $S(U)$ ,  $T(U)$ ,... lub  $R$ ,  $S$ ,  $T$ , ... .
- Wiersze typu  $U$ :  $r(U)$ ,  $s(U)$ ,  $t(U)$ ,... lub  $r$ ,  $s$ ,  $t$ , ... .
- Podzbiory  $U$ :  $X$ ,  $Y$ ,  $Z$ , ... .
- Wiersz  $r(U)$ :  $r(U) = \{(A_1, a_1), (A_2, a_2), \dots, (A_n, a_n)\}$   
lub w uproszczeniu:  $r(U) = (a_1, a_2, \dots, a_n)$   
np.  $r(U) = (Andrzej, Nowak, 22)$

# Relacja (tabela) – przykład

Tabela: Osoby

Imię	Nazwisko	Wiek
Jan	Kowalski	18
Andrzej	Jabłoński	37
Andrzej	Nowak	25

$WIERSZ(U) = \{$   
 $\{(imię, Jan), (nazwisko, Nowak), (wiek, 1)\},$   
 $\{(imię, Jan), (nazwisko, Nowak), (wiek, 2)\},$   
 $\dots,$   
 $\{(imię, Andrzej), (nazwisko, Jabłoński), (wiek, 130)\} \}$

# Def. 4. Zależność funkcyjna

Niech dana będzie tabela  $R(U)$  i niech  $X, Y \subseteq U$  będą zbiorami atrybutów. Mówimy, że w  $R$  spełniona jest zależność funkcyjna  $X \rightarrow Y$ , jeśli dla wszystkich wierszy w relacji  $R$  wartości atrybutów ze zbioru  $Y$  zależą od wartości atrybutów ze zbioru  $X$ . Mówimy wówczas, że  $Y$  zależy funkcyjnie od  $X$  lub, że  $X$  determinuje funkcyjnie  $Y$ .

Tabela  $R$

Nr_indeksu	Nazwisko	Przedmiot	Ocena
1000	Kowalski	Bazy danych	4.5
1000	Kowalski	Akademia CISCO	4.0
1003	Morawski	Bazy danych	5.0
1006	Nowak	Bazy danych	3.0
1006	Nowak	Akademia CISCO	4.5

W tabeli  $R$  występują poniższe zależności funkcyjne:

$Nr\_indeksu \rightarrow Nazwisko$  oraz  $\{Nr\_indeksu, Przedmiot\} \rightarrow Ocena$

# Klucze główne i obce

---

- **Klucz główny** (ang. *primary key*, PK):
  - zbiór atrybutów, który identyfikuje jednoznacznie wiersze tabeli;
  - w tabeli może być kilka kluczy kandydujących, spośród których wybieramy jeden klucz główny (np. w tabeli [Osoba] kluczami kandydującymi mogą być kolumny [NIP], [PESEL], [IdOsoby]).
- **Klucz obcy** (ang. *foreign key*, FK):
  - pozwala na łączenie danych z różnych tabel;
  - zbiór atrybutów w tabeli, który czerpie swoje wartości z tej samej dziedziny, co klucz główny tabeli powiązanej.

# Integralność encji

---

- Każda tabela musi mieć klucz główny, który jednoznacznie identyfikuje wiersze tej tabeli.
- Klucz główny nie może zawierać wartości pustych (null).
- Zabronione są powtórzenia wierszy w ramach jednej tabeli.

# Integralność referencyjna

---

- Klucz obcy może przyjmować jedną z dwóch wartości:
  - klucz główny z tabeli powiązanej;
  - wartość NULL (jeżeli nie koliduje to z innymi regułami integralności).
- Niedozwolone są wskazania poprzez klucz obcy na wiersz, który nie istnieje.
- Kaskadowa aktualizacja i usuwanie zależą od konkretnego zastosowania (np. jeśli usuwamy fakturę VAT z tabeli [Faktura], to usuwamy także wszystkie jej pozycje z powiązanej tabeli [Pozycja]; natomiast jeśli usuwamy grupę studentów z tabeli [Grupa], to raczej nie usuwamy z bazy wszystkich studentów z tej grupy, zapisanych w powiązanej tabeli [Student]).

# Plan wykładu

---

1. Założenia relacyjnego modelu danych - przypomnienie.
2. Definicja podstawowych pojęć.
- 3. Normalizacja bazy danych: 1NF, 2NF, 3NF.**

# Normalizacja bazy danych

- Normalizacja bazy danych jest oparta na teorii modelu relacyjnego. Jest to technika projektowa, która polega na odpowiednim porządkowaniu schematu bazy tak, aby spełniał on poniższe wymagania;
  - dane przechowywane w bazie są zawsze w stanie spójności, a ich modyfikacja nie powoduje błędów;
  - wyszukiwanie i modyfikowanie danych jest wydajne;
  - struktura bazy jest przejrzysta i zrozumiała dla projektantów oraz programistów, co ułatwia administrowanie, utrzymanie i rozwijanie.
- Istnieją następujące postaci normalne: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF. W praktyce zazwyczaj normalizujemy bazę do 2–3NF, wyższe postaci w większości przypadków nie są stosowane.
- Na ogólnym poziomie normalizacja sprowadza się do poniższych zasad:
  - każda tabela powinna mieć klucz, które jednoznacznie identyfikuje wiersze;
  - jedna tabela powinna przechowywać dane na temat jednej encji (klasy obiektów);
  - pola tabeli nie powinny przyjmować wartości NULL;
  - tabele nie powinny zawierać zduplikowanych kolumn, ani wartości danych.



# Zalety i wady normalizacji

---

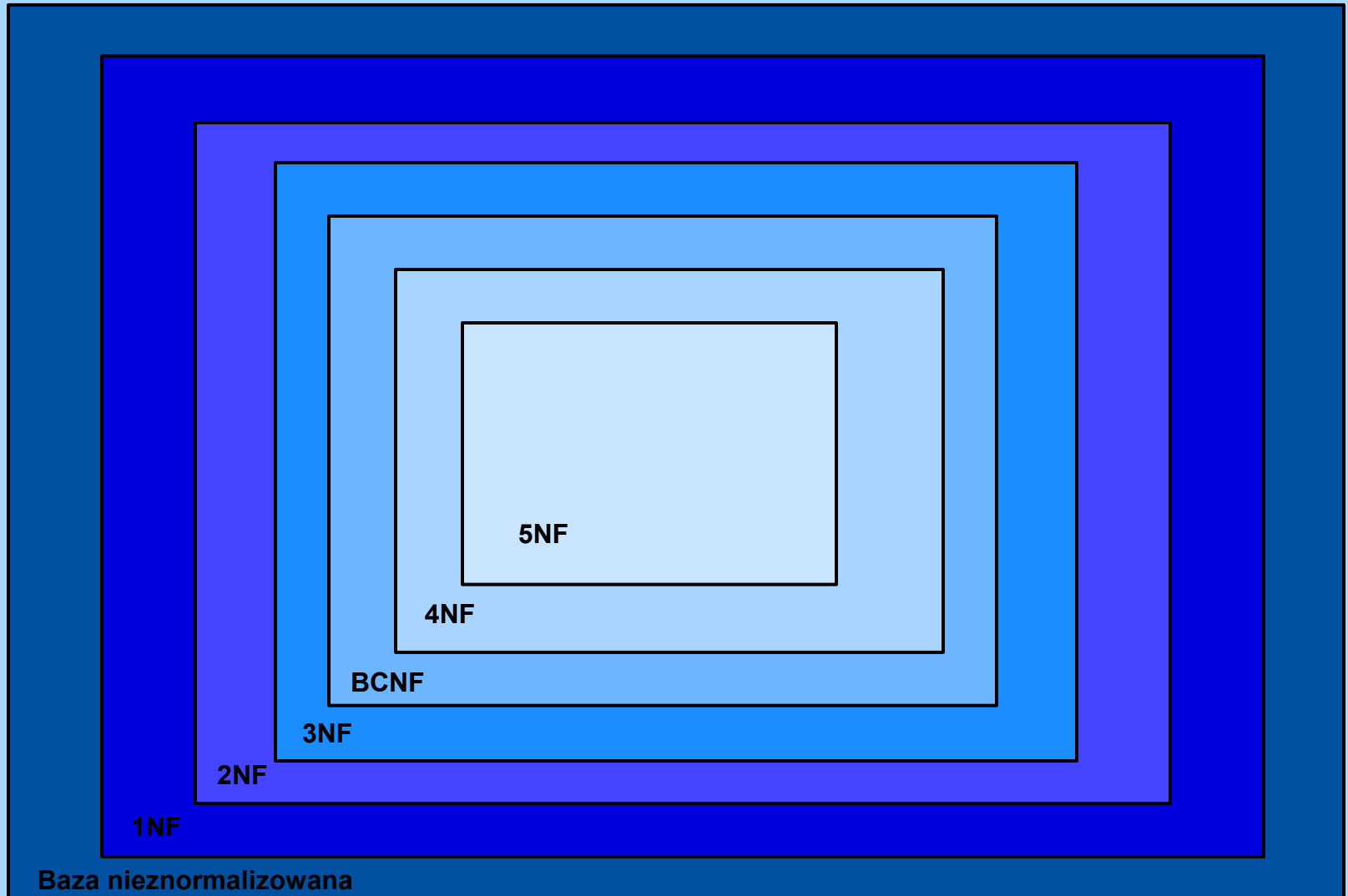
## ■ Zalety normalizacji:

- eliminuje anomalie przy modyfikacji danych, zapewniając ich spójność;
- redukuje nadmiar danych (redundancję), przez co osiągnięta równowaga pomiędzy zajętością przestrzeni dyskowej i wydajnością;
- przyspiesza tworzenie indeksów, sortowanie i przeszukiwanie tabel na skutek zmniejszenia ich rozmiaru;
- ułatwia zachowanie integralności referencyjnej;
- prowadzi do struktury relacyjnej, która jest przejrzysta i zrozumiała;
- zmniejsza liczbę kolumn w tabeli, dzięki czemu maleje rozmiar wierszy, więcej wierszy wchodzi na jedną stronę danych – a w konsekwencji przyspieszane są operacje wejścia-wyjścia.

## ■ Wady normalizacji:

- zwiększa liczbę tabel w bazie, przez co wiele z nich musi być złączanych przy wyszukiwaniu lub modyfikacji danych; zmniejsza to wydajność bazy, gdyż złączenia są operacjami bardzo kosztownymi (rozwiązanie → denormalizacja).

# Poziomy normalizacji bazy danych



# Nieznormalizowany zbiór danych

Kursy\_Oceny

KodKursu	NrPrac	NazwiskoPrac	ImiePrac	IndeksStud	NazwiskoStud	ImieStud	Ocena	TypOceny
INF407	234	Dudek	Damian	34698, 37798, 34888	Kowalski, Nowak, Bracki	Jan, Artur, Krzysztof	3.5; 4.5; 4.0	L1, L2, L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	4.0; 3.5	L1, L2
INF517	345	Choroś	Kazimierz	34668	Morawski	Andrzej	5.0	Egz

**Problem:** w kolumnach [IndeksStud], [NazwiskoStud], [ImieStud], [Ocena], [TypOceny] występuje wiele wartości, które sobie odpowiadają (np. student Jan Kowalski, numer indeksu 34698 za L1 z przedmiotu „INF407” uzyskał ocenę 3.5). Z tego powodu zapis w tabeli może nie być jednoznaczny, a wyszukiwanie danych jest bardzo utrudnione i nieefektywne.

# Pierwsza postać normalna (1NF)

---

Tabela  $R(U)$  jest w **pierwszej postaci normalnej** (ang. *first normal form*, 1NF), jeżeli dziedziny  $DOM(A)$  wszystkich kolumn  $A \in U$  są zbiorami wartości prostych, to znaczy nie są ani zbiorami, ani ciągami elementów należących do zbioru  $\cup \{DOM(A): A \in U\}$ .

Definicje przytoczone są na podstawie książki:

PANKOWSKI T., *Podstawy baz danych*, Wydawnictwo Naukowe PWN, Warszawa, 1992, str. 162 - 192.

# Baza w 1NF – przykład

Kursy\_Oceny (tabela nieznormalizowana)

KodKursu	NrPrac	NazwiskoPrac	ImiePrac	IndeksStud	NazwiskoStud	ImieStud	Ocena	TypOceny
INF407	234	Dudek	Damian	34698, 37798, 34888	Kowalski, Nowak, Bracki	Jan, Artur, Krzysztof	3.5; 4.5; 4.0	L1, L2, L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	4.0; 3.5	L1, L2
INF517	345	Choroś	Kazimierz	34668	Morawski	Andrzej	5.0	Egz

Kursy\_Oceny (pierwsza postać normalna – 1NF)

KodKursu	NrPrac	NazwiskoPrac	ImiePrac	IndeksStud	NazwiskoStud	ImieStud	Ocena	TypOceny
INF407	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L1
INF407	234	Dudek	Damian	37798	Nowak	Artur	4.5	L2
INF407	234	Dudek	Damian	34888	Bracki	Krzysztof	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L2
INF517	345	Choroś	Kazimierz	34668	Morawski	Andrzej	5.0	Egz

# Problemy z bazą w 1NF

Kursy\_Oceny (pierwsza postać normalna – 1NF)

KodKursu	NrPrac	NazwiskoPrac	ImiePrac	IndeksStud	NazwiskoStud	ImieStud	Ocena	TypOceny
INF407	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L1
INF407	234	Dudek	Damian	37798	Nowak	Artur	4.5	L2
INF407	234	Dudek	Damian	34888	Bracki	Krzysztof	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L2
INF517	345	Choroś	Kazimierz	34668	Morawski	Andrzej	5.0	Egz

- **Anomalia usuwania:** jeśli usuniemy studenta o numerze indeksu 34668, utracimy informację o kursie "INF517" i jego wykładowcy.
- **Anomalia modyfikacji** (aktualizacji): jeśli zmienimy wykładowcę przedmiotu "INF407", to: (1) musimy aktualizować jednocześnie trzy kolumny [NrPrac], [NazwiskoPrac] i [ImiePrac]; (2) musimy modyfikować wiele wierszy, co może prowadzić do okresowego zablokowania tabeli albo do sprzeczności danych.
- **Anomalia wstawiania** (dołączania): nie możemy zapisać nowego studenta na dany przedmiot, dopóki nie będzie on miał przynajmniej jednej oceny z tego przedmiotu (chyba że wstawimy wartość NULL w kolumnach [Ocena] i [TypOceny]).

# [Kursy\_Oceny] w 1NF

## – zależności funkcyjne

Kursy\_Oceny (pierwsza postać normalna – 1NF)

KodKursu	NrPrac	NazwiskoPrac	ImiePrac	IndeksStud	NazwiskoStud	ImieStud	Ocena	TypOceny
INF407	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L1
INF407	234	Dudek	Damian	37798	Nowak	Artur	4.5	L2
INF407	234	Dudek	Damian	34888	Bracki	Krzysztof	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L2
INF517	345	Choroś	Kazimierz	34668	Morawski	Andrzej	5.0	Egz

**Klucz główny:**  $K = \{\text{KodKursu}, \text{IndeksStud}, \text{TypOceny}\}$  (drugi kadydat:  $\{\text{NrPrac}, \text{IndeksStud}, \text{TypOceny}\}$ )

**Zależności funkcyjne:**

$K \rightarrow \text{NrPrac}$ ,  $K \rightarrow \text{NazwiskoPrac}$ ,  $K \rightarrow \text{ImiePrac}$ ,  $K \rightarrow \text{NazwiskoStud}$ ,  $K \rightarrow \text{ImieStud}$ ,  $K \rightarrow \text{Ocena}$ ,

$\text{KodKursu} \rightarrow \text{NrPrac}$ ,  $\text{KodKursu} \rightarrow \text{NazwiskoPrac}$ ,  $\text{KodKursu} \rightarrow \text{ImiePrac}$ ,

$\text{NrPrac} \rightarrow \text{NazwiskoPrac}$ ,  $\text{NrPrac} \rightarrow \text{ImiePrac}$ ,  $\text{IndeksStud} \rightarrow \text{NazwiskoStud}$ ,  $\text{IndeksStud} \rightarrow \text{ImieStud}$

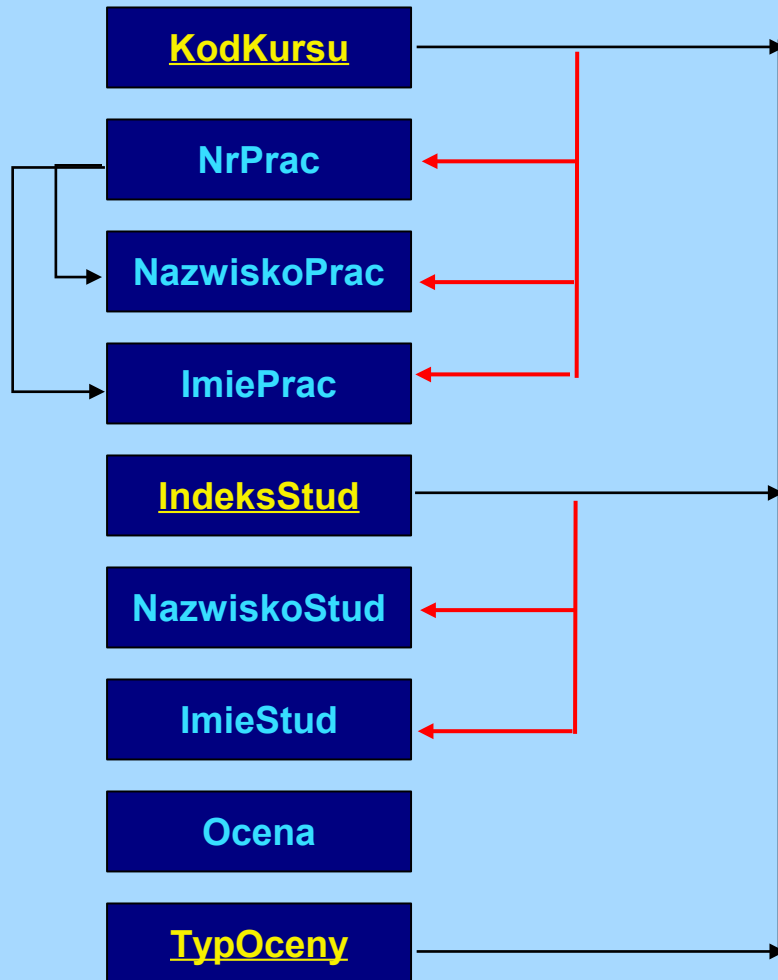
**Źródła problemów:**

kolumny: NrPrac, NazwiskoPrac i ImiePrac są zależne od części klucza - kolumny KodKursu;

kolumny NazwiskoStud i ImieStud są zależne od części klucza - kolumny IndeksStud.

# [Kursy\_Oceny] w 1NF

## – zależności funkcyjne



**Cel:** chcemy, aby każda kolumna niekluczowa był w pełni zależna od klucza głównego.



# Druga postać normalna (2NF)

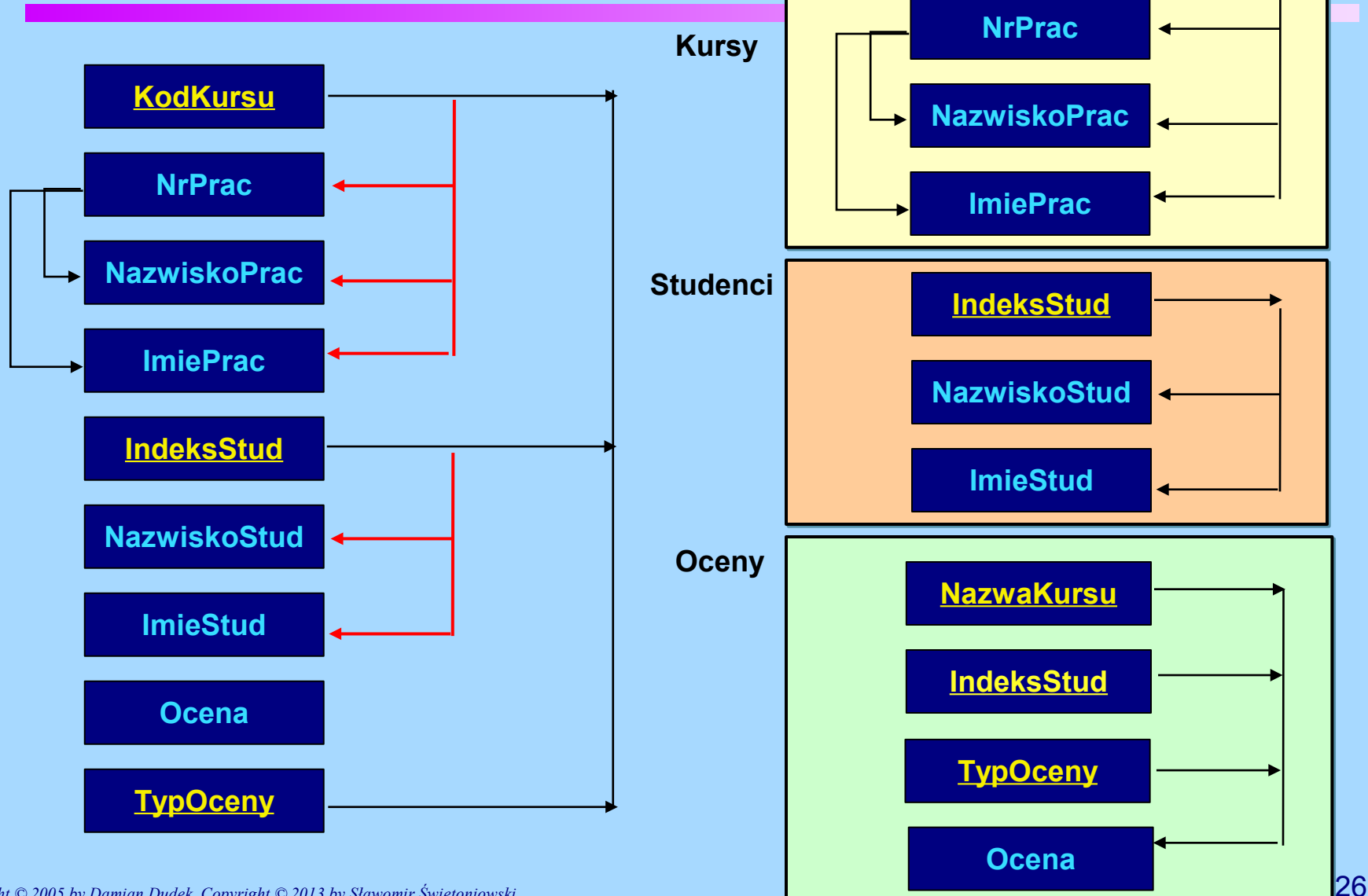
---

Tabela  $R$  jest w **drugiej postaci normalnej** (ang. *second normal form*, 2NF), jeżeli jest w 1NF i każda niekluczowa kolumna  $A \in U$  jest w pełni zależna funkcyjnie od klucza głównego tej tabeli (i od każdego innego klucza kandydującego).

Normalizacja tabeli do 2NF:

- Przeprowadzamy rozkład (dekompozycję) tabeli względem niepełnych zależności funkcyjnych. W ten sposób dążymy, aby każda tabela odnosiła się do jednej klasy obiektów.
- Tabela jest już w 2NF, jeśli klucz główny jest jednoelementowy.
- Przeprowadzanie tabeli do 2NF nie jest procesem jednoznacznym, tzn. dla jednej tabeli może istnieć wiele równoważnych informacyjnie rozkładów w 2NF.

# Normalizacja tabeli [Kursy\_Oceny] do 2NF



# Normalizacja tabeli [Kursy\_Oceny] do 2NF

Kursy\_Oceny (pierwsza postać normalna – 1NF)

KodKursu	NrPrac	NazwiskoPrac	ImiePrac	IndeksStud	NazwiskoStud	ImieStud	Ocena	TypOceny
INF407	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L1
INF407	234	Dudek	Damian	37798	Nowak	Artur	4.5	L2
INF407	234	Dudek	Damian	34888	Bracki	Krzysztof	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	4.0	L1
INF507	234	Dudek	Damian	34698	Kowalski	Jan	3.5	L2
INF517	345	Choroś	Kazimierz	34668	Morawski	Andrzej	5.0	Egz

Kursy (2 NF)

KodKursu	NrPrac	NazwiskoPrac	ImiePrac
INF407	234	Dudek	Damian
INF507	234	Dudek	Damian
INF517	345	Choroś	Kazimierz

Studenci (2NF i od razu 3NF)

IndeksStud	NazwiskoStud	ImieStud
34698	Kowalski	Jan
37798	Nowak	Artur
34888	Bracki	Krzysztof
34668	Morawski	Andrzej

Oceny (2NF i od razu 3NF)

KodKursu	IndeksStud	TypOceny	Ocena
INF407	34698	L1	3.5
INF407	37798	L2	4.5
INF407	34888	L1	4.0
INF507	34698	L1	4.0
INF507	34698	L2	3.5
INF517	34668	Egz	5.0

# Problemy z bazą w 2NF

Kursy (2 NF)

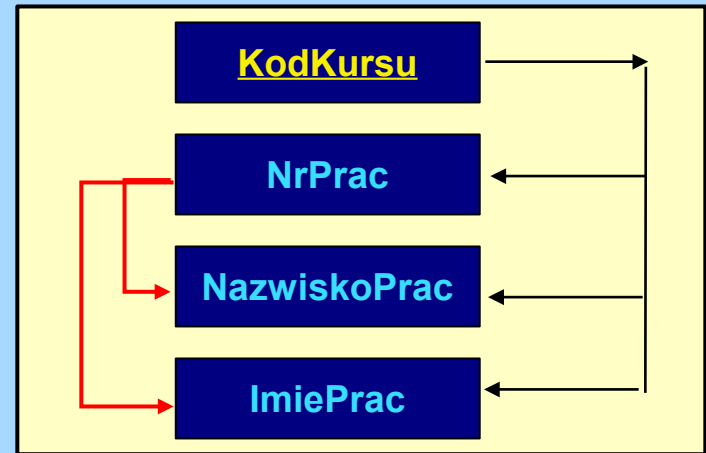
KodKursu	NrPrac	NazwiskoPrac	ImiePrac
INF407	234	Dudek	Damian
INF507	234	Dudek	Damian
INF517	345	Choroś	Kazimierz

- **Anomalia usuwania:** jeśli usuniemy przedmiot "INF517", utracimy informację o wykładowcy; podobnie, usunięcie wierszy dotyczących wykładowcy o numerze "234" powoduje utratę informacji o kursach "INF407" i "INF507".
- **Anomalia modyfikacji** (aktualizacji): zmiana numeru pracownika (pola "NrPrac") lub jego danych osobowych (pola "NazwiskoPrac" i "ImiePrac") wymaga dokonania aktualizacji w wielu wierszach; może to prowadzić do okresowego zablokowania tabeli albo do sprzeczności danych.
- **Anomalia wstawiania** (dołączania): nie można zapisać informacji o zatrudnionym pracowniku, jeśli nie prowadzi on przynajmniej jednego kursu.

# Kursy 2NF – zależności funkcyjne

Kursy (2 NF)

KodKursu	NrPrac	NazwiskoPrac	ImiePrac
INF407	234	Dudek	Damian
INF507	234	Dudek	Damian
INF517	345	Choroś	Kazimierz



**Klucz główny:**  $K = \{KodKursu\}$

**Zależności funkcyjne:**

$KodKursu \rightarrow NrPrac$ ,  $KodKursu \rightarrow NazwiskoPrac$ ,  $KodKursu \rightarrow ImiePrac$ ,

$NrPrac \rightarrow NazwiskoPrac$ ,  $NrPrac \rightarrow ImiePrac$

**Źródło problemów:**

kolumny: NazwiskoPrac i ImiePrac są zależne funkcyjnie od atrybutu niekluczowego NrPrac (zależność tranzytywna - inaczej przechodnia).

# Trzecia postać normalna (3NF)

---

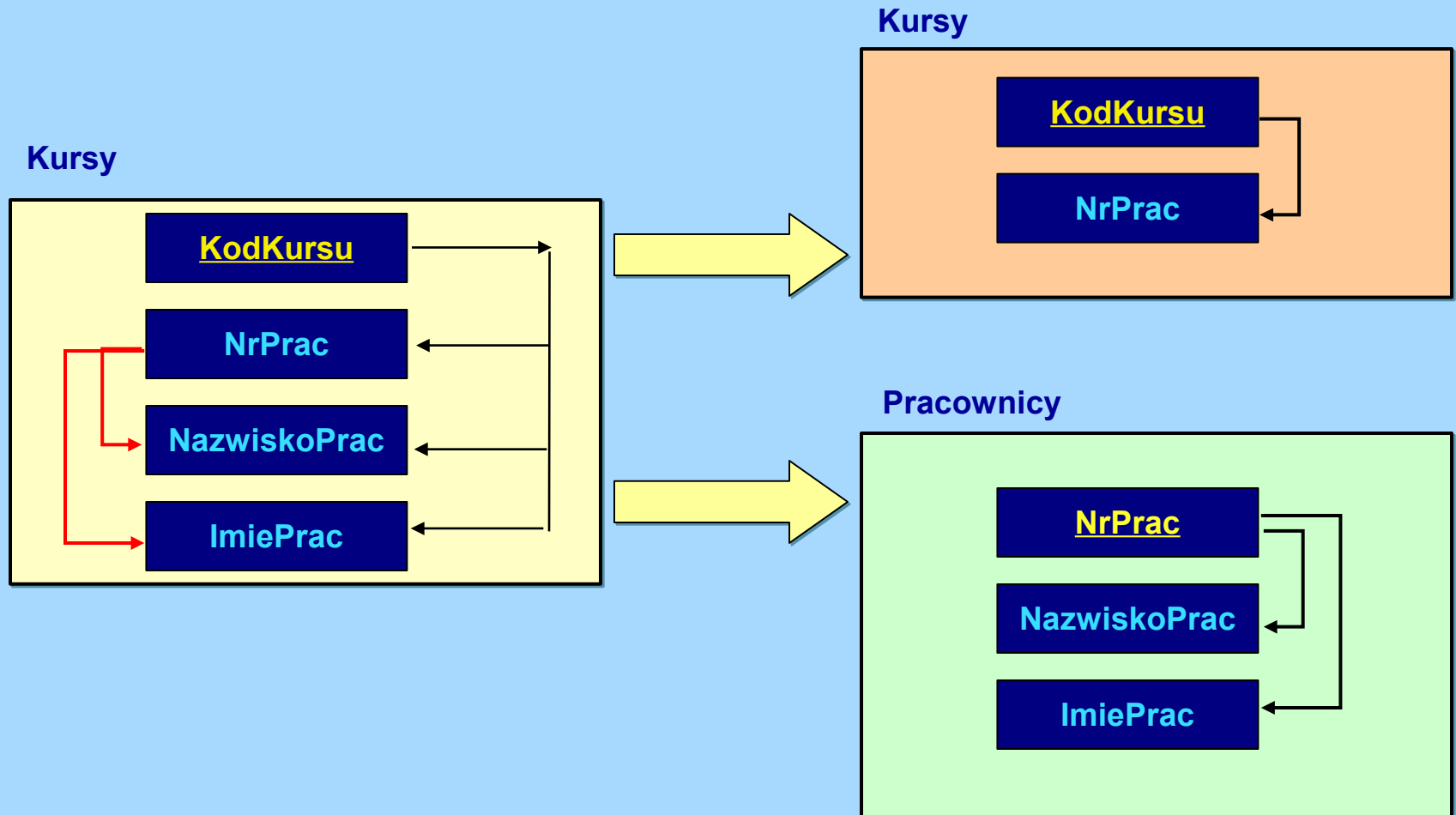
Tabela  $R$  jest w **trzeciej postaci normalnej** (ang. *third normal form*, 3NF), jeżeli jest w 2NF i żaden zbiór kolumn niekluczowych nie jest tranzytywnie (przechodnio) zależny od klucza głównego tabeli  $R$  (ani żadnego innego klucza kandydującego).

Normalizacja tabeli do 3NF:

- Przeprowadzamy rozkład (dekompozycję) relacji względem zależności tranzytywnych.
- Przeprowadzanie relacji z 2NF do 3NF nie jest procesem jednoznacznym, tzn. dla jednej relacji może istnieć wiele równoważnych informacyjnie rozkładów w 3NF.

# Normalizacja relacji [Kursy] do 3NF

## – rozkład (dekompozycja)



# Normalizacja relacji [Kursy] do 3NF

## – rozkład (dekompozycja)

**Kursy (2 NF)**

<b>KodKursu</b>	<b>NrPrac</b>	<b>NazwiskoPrac</b>	<b>ImiePrac</b>
INF407	234	Dudek	Damian
INF507	234	Dudek	Damian
INF517	345	Choroś	Kazimierz

**Kursy (3 NF)**

<b>KodKursu</b>	<b>NrPrac</b>
INF407	234
INF507	234
INF517	345

**Kursy (3 NF)**

<b>NrPrac</b>	<b>NazwiskoPrac</b>	<b>ImiePrac</b>
234	Dudek	Damian
345	Choroś	Kazimierz



# Literatura

---

1. BEYNON-DAVIES P., *Systemy baz danych – nowe wydanie*, WNT, Warszawa 2003.
2. MICROSOFT, *Books On-Line* – dokumentacja systemu *MS SQL Server*, Microsoft Corp. 1988 – 2000.
3. PANKOWSKI T., *Podstawy baz danych*, Wydawnictwo Naukowe PWN, Warszawa, 1992.
4. RANKINS R., JENSEN P., BERTUCCI P., *Microsoft SQL Server 2000. Księga eksperta.*, HELION, Gliwice 2003 (książka dostępna w bibliotece WSIZ „Copernicus”).

# Bazy danych

Wykład 2\_2

Dziękuję za uwagę !