



# Relacyjne Bazy Danych

Andrzej M. Borzyszkowski  
PJATK/ Gdańsk

materiały dostępne elektronicznie  
<http://szuflandia.pjwstk.edu.pl/~amb>

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych



## Dwie metodologie

- Relacyjna baza danych:
  - dane przechowywane w tabelach
  - w tabeli klucz główny plus inne atrybuty
- Diagramy encji i związków
  - encje odpowiadają realnym bytom, które modelujemy
  - naturalny podział na tabele
  - techniczne szczegóły: tabele dla związków wieloznacznych
- Inne podejście: normalizacja
  - zaczynamy od jednej tabeli dla wszystkich danych
    - tzn. integracja danych
  - wydzielamy tabele dla fragmentów danych

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych



# Projekt bazy danych – normalizacja

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych



## Formalne zasady projektowe

- Diagramy związków i encji
  - jedynie intuicyjny podział danych na tabele
  - jasna semantyka atrybutów i łączenia zestawów atrybutów w tabele
- Normalizacja
  - precyzyjna definicja warunków koniecznych/pożądanych
  - identyfikacja anomalii
  - pojęcie determinowania (atrybutów przez inne atrybuty)
  - warunki na możliwość/konieczność podziału danych pomiędzy tabelami

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych



## Pierwsza postać normalna

- 1 postać normalna: komórki tabeli są atomowymi wartościami
  - atrybut wielowartościowy zostaje zamieniony na powtórzenie krotek
  - atrybut złożony zostaje zamieniony na kilka atrybutów
- Przykład: w relacji (Klient join Zamowienie join Pozycja join Towar) [ klient.nr, nazwisko, zamowienie.nr, koszt, towar.nr, opis, ilość ]
  - atrybuty towar.nr i opis odpowiadają jednemu towarowi
  - w jednym zamówieniu może być wiele towarów, w tabeli będą powtórzenia wierszy
  - uwaga: jeśli krotność powtórzeń atrybutu wielokrotnego jest ograniczona i niewielka, można zaproponować kilka odrębnych atrybutów (np. pierwsze i drugie imię), dopuszczając wartość NULL

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

5

## Anomalia wstawiania

- Chcemy wprowadzić do systemu nowego klienta
  - nie ma tej możliwości bez jednoczesnego zamówienia
  - a jeśli z zamówieniem, to może dojść do wstawienia niedokładnej kopii istniejącego towaru
  - a jeśli dopuszczamy wartości NULL dla danych o zamówieniu i towarze, to konieczność ta zniknie po dalszych wstawieniach
- Teraz wprowadzamy nowy towar
  - znowu wymaga to istnienia klienta i zamówienia
  - a jeśli dopuścimy możliwość NULL dla tych danych, to nie będzie w ogóle klucza głównego
  - będzie możliwość wstawienia całkowicie pustej krotki

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

7

## Tabela w 1NF

- Fragment ogólnej tabeli:

K.nr	nazwisko	miasto	Z.nr	koszt	T.nr	opis
3	Szczęsna	Gdynia	1	1,99	4	Linux CD
3	Szczęsna	Gdynia	1	7,45	7	wentylator
3	Szczęsna	Gdynia	12	26,43	19	zegarek męski
4	Łukowski	Gdynia	9	7,45	7	wentylator
4	Łukowski	Gdynia	10	7,45	7	wentylator
8	Koćak	Kielno	2	1,99	4	Linux CD
8	Koćak	Kielno	5	7,45	3	kostka Rubika
13	Soroczyński	Tczew	8	8,18	13	nożyczki drewni

- Trzy anomalie przy zmianie zawartości tabeli: wstawianiu, usuwaniu, aktualizacji

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

6

## Anomalie usuwania i aktualizacji

- Anomalia usuwania
  - usuwamy dane o nożyczkach – zniknie informacja o Soroczyńskim
  - usuwamy dane o Koćak – zniknie informacja o kostce Rubika
  - rozwiązaniem może być wstawianie NULL przy usuwaniu ostatniej krotki
  - dopuszcza to możliwość krotki całej równej NULL
- Anomalia aktualizacji
  - poprawiamy literówkę w nazwisku „Szczęśna”
  - albo zmieniamy miejsce jej zamieszkania
  - może się okazać, że nie wszystkie wystąpienia zostaną zaktualizowane

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

8

## Zależności atrybutów

- Pojęcie funkcyjnej zależności (determinowania)
  - X funkcyjnie determinuje Y (oznaczenie  $X \rightarrow Y$ ): wszystkie krotki o pewnych wartościach atrybutów X mają te same wartości atrybutów Y
  - w szczególności: klucz funkcyjnie determinuje wszystkie pozostałe atrybuty
  - np. numer indeksu studenta identyfikuje studenta
  - imię i nazwisko nie identyfikuje studenta
  - ale samo imię determinuje płeć
  - a kod pocztowy determinuje województwo/powiat/gminę ?
- Redundancja
  - gdy w relacji R występuje zależność funkcyjna  $X \rightarrow Y$  oraz X nie jest kluczem kandydującym

10

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Zależności funkcyjne wynikowe

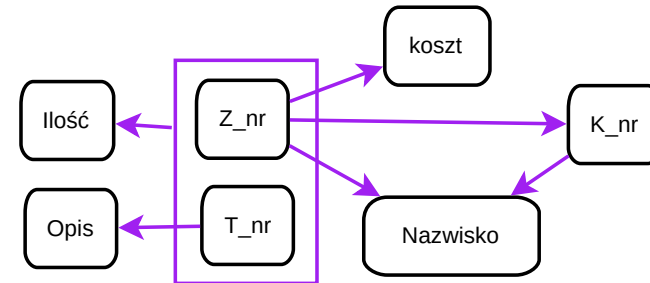
- Pewne zależności funkcyjne powodują zachodzenie innych zależności
  - można formalnie wywnioskować te zależności pochodne
- Reguły wnioskowania dla zależności funkcyjnych (Armstrong)
  - zwrotność:  $X \rightarrow X$
  - uzupełnienie:  $X \rightarrow Y$  pociąga  $XZ \rightarrow Y$
  - rzut:  $X \rightarrow YZ$  pociąga  $X \rightarrow Y$
  - suma:  $X \rightarrow Y$  oraz  $X \rightarrow Z$  pociąga  $X \rightarrow YZ$
  - przechodniość:  $X \rightarrow Y$  oraz  $Y \rightarrow Z$  pociąga  $X \rightarrow Z$
- Zależności trywialne i nietrywialne
  - zawsze  $X \twoheadrightarrow Y$  pociąga  $X \rightarrow Y$
  - inne zależności trzeba postulować

12

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Redundancja, przykład

- Relacja (Klient join Zamowienie join Pozycja join Towar)  
[ klient.nr, nazwisko, zamowienie.nr, koszt, towar.nr, opis, ilość ] spełnia zależności:



- niektóre ze strzałek wychodzą z podzbioru klucza
- inne wychodzą z innych (zbiorów) atrybutów
- Redundancja
  - niepotrzebnie powtarzamy nazwisko klienta dla różnych towarów z tego samego zamówienie
  - nie można zapisać nazwiska klienta, który nie zamówił żadnego towaru

11

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Rozkład odwracalny (bezstratny)

- Relacje  $R_1, \dots, R_n$  nazywamy rozkładem odwracalnym relacji R wtedy i tylko wtedy, gdy złączenie naturalne relacji  $R_1, \dots, R_n$  jest równe wyjściowej relacji R
  - uwaga: oczywiście relacje  $R_1, \dots, R_n$  są wówczas rzutami relacji R
  - oraz w sumie obejmują wszystkie atrybuty relacji R
  - prawo zachowania atrybutów
- Założenie:  $R_1$  i  $R_2$  są rzutami pewnej relacji R oraz obejmują wszystkie atrybuty R
  - oczywiście złączenie naturalne  $R_1$  i  $R_2$  będzie zawierać R
    - dlaczego?
  - pytanie: jakie warunki gwarantują, że złączenie naturalne  $R_1$  i  $R_2$  będzie równe dokładnie wyjściowej relacji R?

13

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Rozkłady, przykład

- Fragment tabeli klient [ nr, nazwisko, miasto ]

nr	nazwisko	miasto
3	Szczęсна	Gdynia
4	Łukowski	Gdynia

- rozkłady odwracalne (bez utraty informacji)

nr	nazwisko	nr	miasto
3	Szczęсна	3	Gdynia
4	Łukowski	4	Gdynia

- rozkład nieodwracalny (z utratą informacji)

nr	miasto
3	Gdynia
4	Gdynia

nr	miasto
3	Gdynia
4	Gdynia

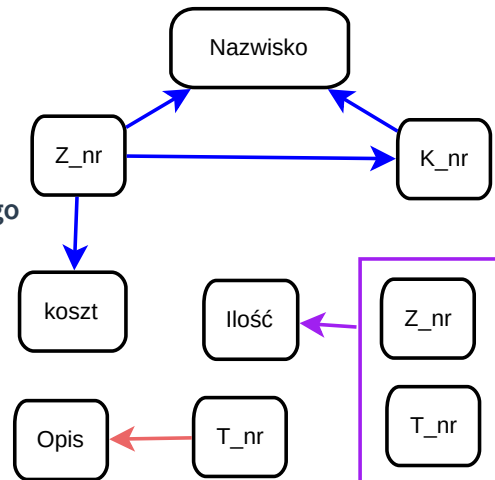
© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

15

## Druga postać normalna

- Relacja R jest w drugiej postaci normalnej wtedy i tylko wtedy, gdy jest w 1NF i wszystkie atrybuty nienależące do klucza zależą od całego klucza, a nie od jego części



© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

17

## Rozkład odwracalny, tw. Heatha

- Tw. Heatha: Niech R będzie relacją, zaś A, B i C zbiorami atrybutów. Jeżeli R spełnia zależność funkcyjną  $A \rightarrow B$ , wówczas relacja R jest równa złączeniu naturalnemu swoich rzutów na  $\{A, B\}$  i  $\{A, C\}$
- Twierdzenie jest prawdziwe gdy  $A \rightarrow C$ 
  - wówczas A zawiera klucz relacji R
  - rozkład jest niepotrzebny, prowadzi do związku 1-1, relacje mogły być scalone
- oraz gdy  $A \nrightarrow C$ , wówczas w R występowała redundancja
  - rozkład jest uzasadniony
- $\{NR\} \rightarrow \{MIASTO\}$  oraz  $\{NR\} \rightarrow \{NAZWISKO\}$ 
  - rozkład niepotrzebny, NR jest kluczem w relacji scalonej
- $\{MIASTO\} \nrightarrow \{NR\}$  oraz  $\{MIASTO\} \nrightarrow \{NAZWISKO\}$ 
  - rozkład nieodwracalny

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

16

## Druga postać normalna, c.d.

- Anomalia aktualizacji
  - dane o towarach występują tylko jeden raz
  - nie ma problemu z nieprawidłową aktualizacją
  - dane klienta z wieloma zamówieniami nadal są powtarzane
- Anomalia usuwania
  - dane o kliencie związane są z jakimś zamówieniem
  - anomalia usuwania nadal jest obecna
- Anomalia wstawiania
  - analogicznie do anomalii usuwania – obecna

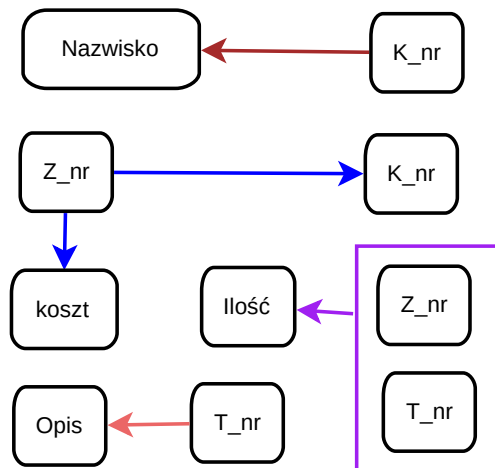
© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

18

## Trzecia postać normalna

- Relacja R jest w trzeciej postaci normalnej wtedy i tylko wtedy, gdy jest w 2NF i wszystkie atrybuty nienależące do klucza zależą bezpośrednio od klucza
  - innymi słowy: krotka składa się z klucza głównego i pewnej liczby atrybutów niezależnych; atrybuty te można aktualizować niezależnie od siebie



© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

19

## Postaci normalne, druga i trzecia

- Każdy projekt można doprowadzić do 3 postaci normalnej
  - i powinno się doprowadzić
- W zaawansowanych zastosowaniach są powody by robić inaczej
  - kopiowanie danych, by ułatwić dostęp
  - utrzymywanie danych zbiorczych (też pewien sposób kopiowania)
  - są narzędzia by uniknąć anomalii (procedury wyzwalane, reguły Postgresa)

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

21

## Trzecia postać normalna, c.d.

- Anomalia aktualizacji
  - dane o klientach występują tylko jeden raz
  - nie ma problemu z nieprawidłową aktualizacją
- Anomalia usuwania
  - dane o kliencie są niezależne od zamówień, można usunąć zamówienia pozostawiając dane klienta
- Anomalia wstawiania
  - również nie ma przeszkód w niezależnym wstawianiu danych o klientach czy towarach

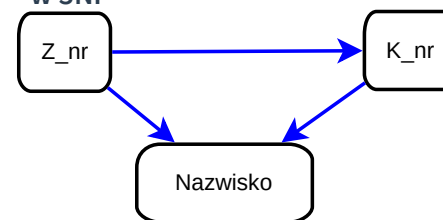
© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

20

## Trzecia postać normalna – 3NF, przykład

- [Z.nr,K.nr,nazwisko] nie jest w 3NF, ma rozkład ma dwie relacje w 3NF



z_nr	k_nr	nazwisko
12	3	Szczęсна
9	4	Łukowski



z_nr	k_nr
12	3
9	4



k_nr	nazwisko
3	Szczęсна
4	Łukowski

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

22

## Trzecia postać normalna – kontrprzykład

- [Z.nr,K.nr,nazwisko] ma też inny rozkład na dwie relacje w 3NF:



z_nr	k_nr
12	3
9	4



z_nr	nazwisko
12	Szczęsna
9	Łukowski

- każda relacja [Z.nr,K.nr,nazwisko] jest złożeniem swoich rzutów

23

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Normalizacja

- Rozkład do 2NF

R ( A, B, C, D )

PRIMARY KEY ( A, B )

A → D

rozkładamy następująco:

R1 ( A, D )

PRIMARY KEY ( A )

R2 ( A, B, C )

PRIMARY KEY ( A, B )

FOREIGN KEY ( A )

REFERENCES R1

- Rozkład do 3NF

R ( A, B, C )

PRIMARY KEY ( A )

B → C

rozkładamy następująco:

R1 ( B, C )

PRIMARY KEY ( B )

R2 ( A, B )

PRIMARY KEY ( A )

FOREIGN KEY ( B )

REFERENCES R1

25

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Trzecia postać normalna – kontrprzykład

- Nie jest to pożyteczny rozkład

- nie każde złożenie relacji [Z.nr,K.nr] oraz [Z.nr,nazwisko] spełnia zależność funkcyjną K.nr → nazwisko

z_nr	k_nr
10	4
9	4

z_nr	nazwisko
10	Szczęsna
9	Łukowski

- Każdy projekt można doprowadzić do 3 postaci normalnej bez utraty zależności

z_nr	k_nr	nazwisko
10	4	Szczęsna
9	4	Łukowski

24

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Normalizacja – przykład konkretny

- Rozkład do 2NF

R ( t.nr, z.nr, ilość, opis )

PRIMARY KEY ( t.nr, z.nr )

t.nr → opis

rozkładamy następująco:

towar ( t.nr, opis )

PRIMARY KEY ( t.nr )

pozycja ( t.nr, z.nr, ilość )

PRIMARY KEY ( t.nr, z.nr )

FOREIGN KEY ( t.nr )

REFERENCES towar

- Rozkład do 3NF

R ( z.nr, k.nr, nazwisko )

PRIMARY KEY ( z.nr )

k.nr → nazwisko

rozkładamy następująco:

klient ( k.nr, nazwisko )

PRIMARY KEY ( k.nr )

zamowienie ( z.nr, k.nr )

PRIMARY KEY ( z.nr )

FOREIGN KEY ( k.nr )

REFERENCES klient

26

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Postać normalna Boyce'a-Codda – BCNF

- Relacja R jest w postaci normalnej Boyce'a/Codda (BCNF) gdy elementem determinującym każdej nietrywialnej zależności funkcyjnej jest klucz kandydujący
  - tzn. relacja R jest w BCNF gdy na diagramie zależności funkcyjnych jedynymi strzałkami wychodzącymi są strzałki wychodzące z kluczy kandydujących
  - dla 3NF nakłada się warunek jedynie dla atrybutów niebędących częścią klucza
- Okazuje się, że nie każdą relację można rozłożyć na relacje w postaci Boyce'a-Codda nie tracąc zależności funkcyjnych
  - ale można zdefiniować procedurę wyzwalaną zapewniającą zachowanie brakującej zależności funkcyjnej

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

27

## BCNF, próba rozkładu

- Istnieje rozkład odwracalny relacji SZKOŁA na
  - Lektor ( LEKTOR, JĘZYK )**
  - PRIMARY KEY ( LEKTOR )**
  - Zapis ( STUDENT, LEKTOR )**
  - jedyna zależność funkcyjna to { LEKTOR } → { JĘZYK }
  - brakuje zależności { STUDENT, JĘZYK } → { LEKTOR }
  - nie można aktualizować obu relacji i zagwarantować zachowania brakującej zależności funkcyjnej
- Wniosek: nie zawsze jest możliwy rozkład odwracalny na relacje spełniające BCNF z zachowaniem zależności funkcyjnych
  - ale można zdefiniować procedurę wyzwalaną zapewniającą zachowanie brakującej zależności funkcyjnej

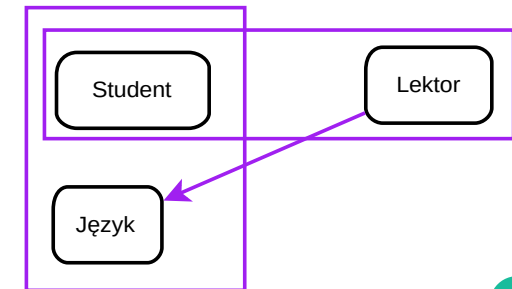
© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

29

## BCNF, (kontr)przykład

- Załóżmy, że relacja SZKOŁA ma definicję
  - SZKOŁA ( STUDENT, JĘZYK, LEKTOR )**
  - UNIQUE ( STUDENT, JĘZYK )**
  - UNIQUE ( STUDENT, LEKTOR )**
  - załóżmy dodatkowo, że każdy lektor prowadzi tylko jeden język
  - tzn. diagram zależności funkcyjnych wygląda następująco:
- SZKOŁA nie jest w BCNF



© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

28

## Przykłady, gdy normalizacja nie wystarcza

- Dane zagregowane:
  - jest to pewien rodzaj kopiowania danych
  - zaleca się (w zasadzie) nie zapisywać atrybutów wynikowych
  - teoria normalizacji nie wypowiada się na ten temat
- Determinowanie bezwarunkowe
  - np. pesel determinuje datę urodzenia
  - a więc nie warto w ogóle zapisywać daty urodzenia, gdy zapisuje się pesel
  - teoria normalizacji mówi jedynie o determinowaniu atrybutów zapisanych w tabeli

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

34

## Przykłady, c.d.

- Tabele słownikowe
  - czasami problem z powtarzalnością ma charakter pragmatyczny
  - np. zapisujemy dane studentów razem z nazwą wydziału, nazwa może być długa, wielokrotne powtarzanie nazwy umożliwia błędy zapisu
  - jeśli zaplanujemy kolumny: album, nazwa, skrót
    - gdzie nazwa i skrót nazwy determinują się wzajemnie,
    - to teoria normalizacji wskaże rozkład z odrębną tabelą [nazwa, skrót\_nazwy]
    - ale nie wymusi by kluczem obcym był właśnie skrót nazwy