Bazy Danych

Andrzej M. Borzyszkowski
Instytut Informatyki
Uniwersytetu Gdańskiego
materiały dostępne elektronicznie
http://inf.ug.edu.pl/~amb

Projekt bazy danych – normalizacja

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

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

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

Tabela w 1NF

- Fragment ogólnej tabeli:

K_nr	***nazwisko***	***Z_nr***	***koszt***	***T_nr***	***opis***	***ilosc***
3	Szczęsna	1	2,99	4	Linux CD	1
3	Szczęsna	1	2,99	7	wentylator	5
3	Szczęsna	12	0,99	19	zegarek	1
4	Łukowski	9	6,99	7	wentylator	5
4	Łukowski	10	0,99	7	wentylator	1
8	Kołak	2	0	4	Linux CD	2
8	Kołak	5	0	3	kostka Rubika	4
13	Soroczyński	8	5,99	13	nożyczki	3

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

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

Anomalie usuwania i aktualizacji

- Anomalia usuwania
  - usuwamy dane o nożyczkach – zniknie informacja o Soroczyńskim
  - usuwamy dane o Kołak – 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 „Szczęsna”
  - albo zmieniamy miejsce jej zamieszkania
  - może się okazać, że nie wszystkie wystąpienia zostaną zaktualizowane

Wartości NULL

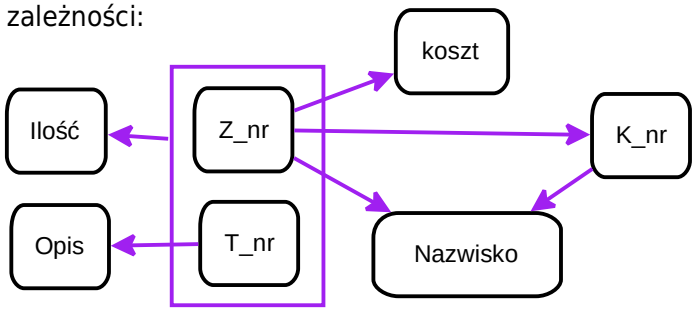
- Semantyka NULL jest niejednoznaczna
  - nieznajomość danych
  - dane jeszcze nie wprowadzone
  - dane nie mają sensu w kontekście
- Problemy z NULL
  - wydajność – zajmują miejsce w tabeli
  - nieoczywista semantyka dla funkcji agregujących
  - nieoczywista semantyka dla wartości NULL klucza obcego
  - klucz główny nie może mieć wartości NULL
- Zasada projektowa: unikać, o ile to możliwe, dopuszczania wartości NULL

Zależności atrybutów

- Pojęcie funkcyjnej zależności (determinowania)
  - X funkcyjnie determinuje Y (oznaczenie X→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→Y oraz X **nie jest** kluczem kandydującym

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

Zależności funkcyjne wynikowe

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

Rozkład odwracalny (bezstratny)

- Relacje R1,...,Rn nazywamy rozkładem odwracalnym relacji R wtedy i tylko wtedy, gdy złączenie naturalne relacji R1,...,Rn jest równe wyjściowej relacji R
  - uwaga: oczywiście relacje R1,...,Rn są wówczas rzutami relacji R
  - oraz w sumie obejmują wszystkie atrybuty relacji R
  - prawo zachowania atrybutów
- Założenie: R1 i R2 są rzutami pewnej relacji R oraz obejmują wszystkie atrybuty R
  - oczywiście złączenie naturalne R1 i R2 będzie zawierać R
    - dlaczego?
  - pytanie: jakie warunki gwarantują, że złączenie naturalne R1 i R2 będzie równe wyjściowej relacji R, tzn nie będzie większe?

Rozkłady, przykład

- Fragment tabeli klient [nr, nazwisko, miasto]

nr	nazwisko	miasto
3	Szczęsna	Gdynia
4	Łukowski	Gdynia

  - rozkład odwracalny (bez utraty informacji)

nr	nazwisko
3	Szczęsna
4	Łukowski

nr	miasto
3	Gdynia
4	Gdynia

  - rozkład nieodwracalny (z utratą informacji)

nazwisko	miasto
Szczęsna	Gdynia
Łukowski	Gdynia

nr	miasto
3	Gdynia
4	Gdynia

Rozkład odwracalny, tw. Heatha

- Tw. Heatha: Niech R będzie relacją, zaś A, B i C zbiorami atrybutów. Jeżeli R spełnia zależność funkcyjną A→B, wówczas relacja R jest równa złączeniu naturalnemu swoich rzutów na { A,B } i { A,C }
- Twierdzenie jest zasadniczo używane gdy A↛C
  - wówczas w R występuje redundancja i rozkład jest uzasadniony
- Teza twierdzenia nie zachodzi, gdy żadna z zależności funkcyjnych nie jest spełniona:

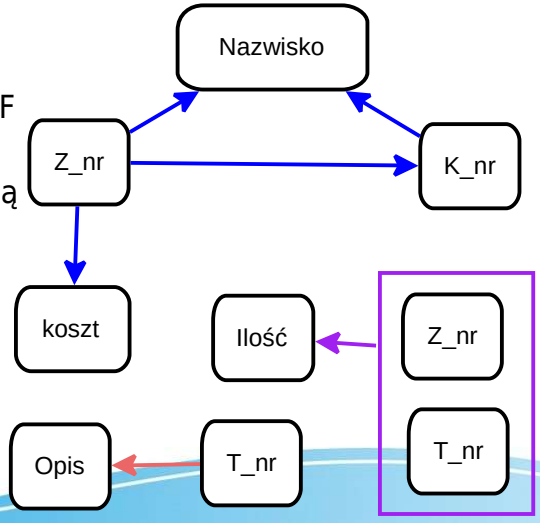
  { MIASTO }↛{ NR } oraz { MIASTO }↛{ NAZWISKO }
  - rozkład nieodwracalny

Rozkład odwracalny, tw. Heatha, c.d.

- Tw. Heatha: Niech R będzie relacją, zaś A, B i C zbiorami atrybutów. Jeżeli R spełnia zależność funkcyjną A→B, wówczas relacja R jest równa złączeniu naturalnemu swoich rzutów na { A,B } i { A,C }
- Twierdzenie jest prawdziwe gdy również A→C
  - wówczas A zawiera klucz relacji R
  - rozkład nie jest konieczny, prowadzi do związku 1-1, relacje mogły być scalone
- { NR }→{ MIASTO } oraz { NR }→{ NAZWISKO }
  - rozkład nie jest konieczny, mogła być jedna relacja, NR jest kluczem

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

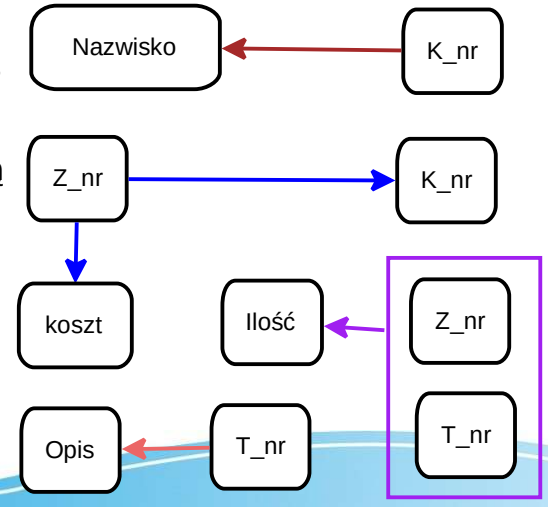


Druga postać normalna, c.d.

- 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

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



Trzecia postać normalna, c.d.

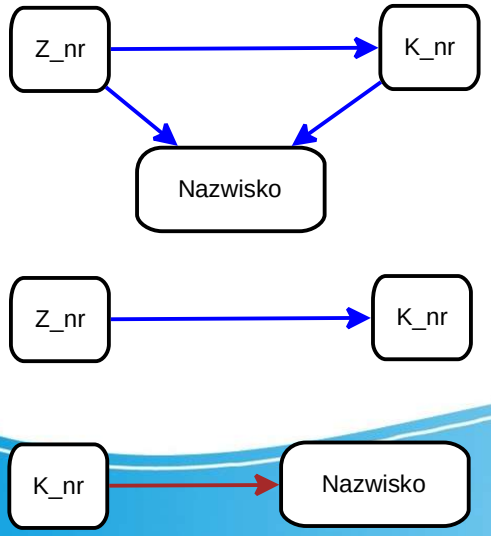
- Anomalia aktualizacji
  - dane o klientach występują tylko jeden raz
  - nie ma problemu z nieprawidłową aktualizacją
- Anomalia usuwania
  - dane o kliencie są niezależne od zamówień, można usunąć zamówienia pozostawiając dane klienta
- Anomalia wstawiania
  - również nie ma przeszkód w niezależnym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

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
  - kopiowane danych, by ułatwić dostęp
  - utrzymywanie danych zbiorczych (też pewien sposób kopiowania)
  - są narzędzia by uniknąć anomalii (procedury wyzwalane, reguły Postgresa)

Trzecia postać normalna – 3NF, przykład

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



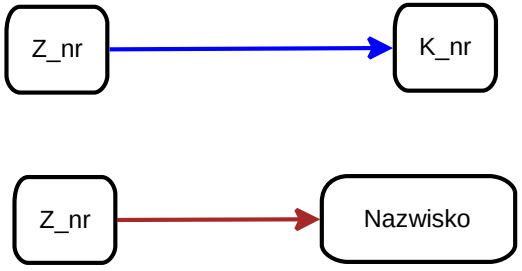
z_nr	k_nr	nazwisko
12	3	Szczęsna
9	4	Łukowski

z_nr	k_nr
12	3
9	4

k_nr	nazwisko
3	Szczęsna
4	Łukowski

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

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

z_nr	k_nr
10	4
9	4

z_nr	nazwisko
10	Szczęsna
9	Łukowski

z_nr	k_nr	nazwisko
10	4	Szczęsna
9	4	Łukowski

Normalizacja

- Rozkład do 2NF

  R (A, B, C, D)
  PRIMARY KEY (A, B)
  A → D

(D zależy od części klucza)

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

(zależność tranzytywna A → B → C)

rozkładamy następująco:

R1 (B, C)
PRIMARY KEY (B)

R2 (A, B)
PRIMARY KEY (A)
FOREIGN KEY (B)
REFERENCES R1

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

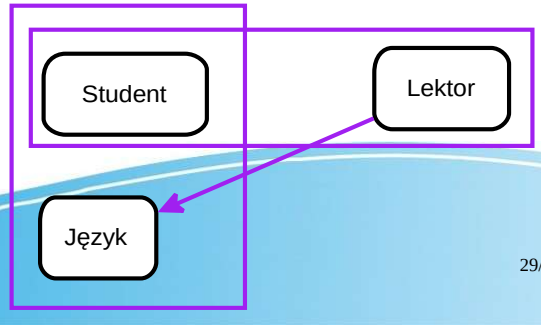
Postać normalna Boyce'a-Codda – BCNF

- 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

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



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 zachowanie zależności funkcyjnych
  - ale można zdefiniować procedurę wyzwalaną zapewniającą zachowanie brakującej zależności funkcyjnej

Czwarta postać normalna

- Pojęcie determinowania wielowartościowego
  - 1NF wymusza powtórzenia wierszy, gdy wartością atrybutu ma być zbiór wartości atomowych
  - X determinuje Y wielowartościowo:
    dla każdych dwóch krotek t1 i t2 takich, że t1[X]=t2[X] istnieją krotki t3 i t4 takie, że
    - t3[X]=t4[X]=t1[X]
    - t3[Y]=t1[Y], t4[Y]=t2[Y]
    - dla pozostałych atrybutów Z zachodzi
    - t3[Z]=t2[Z], t4[Z]=t1[Z]
  - oznaczenie: X⏭Y
    - ponieważ Z gra tę samą rolę, można pisać X⏭Y|Z
  - fakt: jeśli X→Y, to X⏭Y (dlaczego?)

Czwarta postać normalna, przykład

- Chcemy zapisywać dane o studentach, zapisach na lektoraty i zapisach na fakultety
  - lektoraty i fakultety są niezależne
  - typowa tabelka

	nazwisko	lektorat	fakultet
t1	Szczęsna	angielski	logika
t2	Szczęsna	niemiecki	kryptografia
	Szczęsna	francuski	logika
t3	Szczęsna	angielski	kryptografia
t4	Szczęsna	niemiecki	logika
	Szczęsna	francuski	kryptografia

  - każda wartość lektoratu musi być skombinowana z każdą wartością fakultetu

Czwarta postać normalna,c.d.

- Anomalie
  - wstawianie, usuwanie, aktualizacja:
  - można naruszyć warunek, że każda wartość jest do pary z każdą, można niejednolicie aktualizować wartości
  - w tym przykładzie 3NF i wcześniejsze nie są naruszone
  - bo nie ma w ogóle zależności funkcyjnych
- Rozwiązanie
  - jeśli X⏭Y|Z, gdzie X,Y i Z są rozłącznymi zbiorami atrybutów, to relację R(X,Y,Z) należy podzielić na R1(X,Y) oraz R2(X,Z)
- Innymi słowy: zależność wielowartościowa (nietrywialna) oznacza, że relacja musi być złączeniem naturalnym dwóch relacji
  - 4NF: nie ma potrzeby podziału na złączenie dwóch relacji

Piąta postać normalna

- Tabela jest w 5NF, jeśli nie jest złączeniem innych tabel
  - praktyczne znaczenie 5NF jest bliskie zera
  - jeśli wiemy z góry, że tabela jest złączeniem, to otrzymujemy radę, by ją potraktować jako złączenie

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

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