**Wrocławska Wyższa Szkoła Informatyki Stosowanej**

# Normalizacja baz danych

Dr hab. inż. Krzysztof Pieczarka

Email: krzysztof.pieczarka@gmail.com

Normalizacja relacji ma na celu takie jej przekształcenie, by nie posiadała ona cech niepożądanych. Normalizacja polega na dekompozycji relacji na „mniejsze” schematy relacji.

Proces normalizacji musi posiadać następujące własności:

- żaden atrybut (składowa definicyjna encji) nie może ulec zagubieniu w trakcie tego procesu,
- dekompozycja relacji nie może prowadzić do utraty informacji – wszystkie (pożądane) zależności funkcyjne (głównie semantyczne) są reprezentowane w otrzymanych mniejszych schematach relacji.

Weźmy dla przykładu relację opisującą zajęcia odbywające się na uczelni w jednym semestrze. Relacja ta może zawierać następujące atrybuty:

- nazwa przedmiotu,
- imię,
- nazwisko,
- adres prowadzącego

Przykładowa krotka takiej relacji mogłaby mieć postać:

(język angielski, Lucyna, Nowak, ul. Królewska 30/3 Kraków)

Jednak z tak zaprojektowaną relacją związane są następujące anomalia:

- adres składający się z kilku części nie został podzielony w związku z tym nie jest możliwe sprawdzenie ile osób mieszka w Krakowie i nie potrzebuje hotelu;
- jeden prowadzący może mieć zajęcia z kilku przedmiotów, w związku z tym występuje redundancja danych;
- zmiana jednej z informacji o prowadzącym (np. adresu) powoduje konieczność zmiany wszystkich krotek zawierających te dane w celu zachowania integralności;
- nie jest możliwe wprowadzenie informacji o prowadzącym, który w aktualnym semestrze nie ma żadnych zajęć;
- usunięcie przedmiotu może spowodować również usunięcie wszelkich informacji o osobie, która go prowadziła.

Utrzymanie integralności takiej bazy nie jest więc proste.

Jednak opisaną relację można zamienić na dwie inne, które nie będą posiadały tych wad:

- relacja Prowadzący:

(identyfikator, imię, nazwisko, kod pocztowy, miejscowość, ulica, nr_domu, nr_mieszkania)

- relacja Zajęcia

(nazwa, id_prowadzącego)

Te dwie relacje nie posiadają opisanych wcześniej cech niepożądanych ponieważ:

- adres jest zdekomponowany na części składowe, w związku z czym możliwe jest wyszukiwanie danych np. według miejscowości zamieszkania prowadzącego;
- zmiana informacji o prowadzącym (np. adresu) nie powoduje konieczności zmian danych w relacji "Zajęcia". Zmiana ta odbywa się tylko w jednym miejscu;
- możliwe jest wprowadzenie informacji o osobie, która nie ma zajęć w aktualnym semestrze, ale być może będzie je miała w semestrze następnym;
- usunięcie przedmiotu nie powoduje usunięcia informacji o osobie, która go prowadziła.

Jednak taka reprezentacja danych posiada wady podobne do opisanych wcześniej, ale dotyczące przedmiotów. Dlatego w dobrze zaprojektowanej bazie danych konieczne jest wydzielenie trzeciej tabeli, która będzie zawierała spis przedmiotów.

Postaci normalne - określają stopień dekompozycji bazy danych.

W odróżnieniu od schematu procesu projektowania bazy danych „z góry do dołu” (ang. top – down – od ogółu do szczegółów), normalizacja jest uznawana niekiedy za odrębną metodologię projektowania typu „z dołu do góry” (ang. bottom – up, tzn. od szczegółów do uogólnień). W swojej pracy na temat relacyjnego modelu danych E.F.Codd sformułował reguły projektowania relacyjnych baz danych. Reguły te zostały pierwotnie nazwane postaciami normalnymi. Codd opisał trzy postacie normalne oznaczane często symbolami 1NF, 2NF, 3NF. Proces kolejnego przekształcania projektu bazy danych przez te trzy postacie normalne jest znany jako normalizacja bazy danych.

W połowie lat siedemdziesiątych spostrzeżono pewne niedostatki w trzeciej postaci normalnej Codd’a i zdefiniowano mocniejszą postać normalną, znaną jako postać normalna Boyce'a-Codda. Później Fagin przedstawił czwartą postać normalną, i piątą postać normalną.

Jest pięć (ale jedna z nich dodatkowo ma dwa warianty!) postaci normalnych:

(I, II, III, III B-C, IV, V).

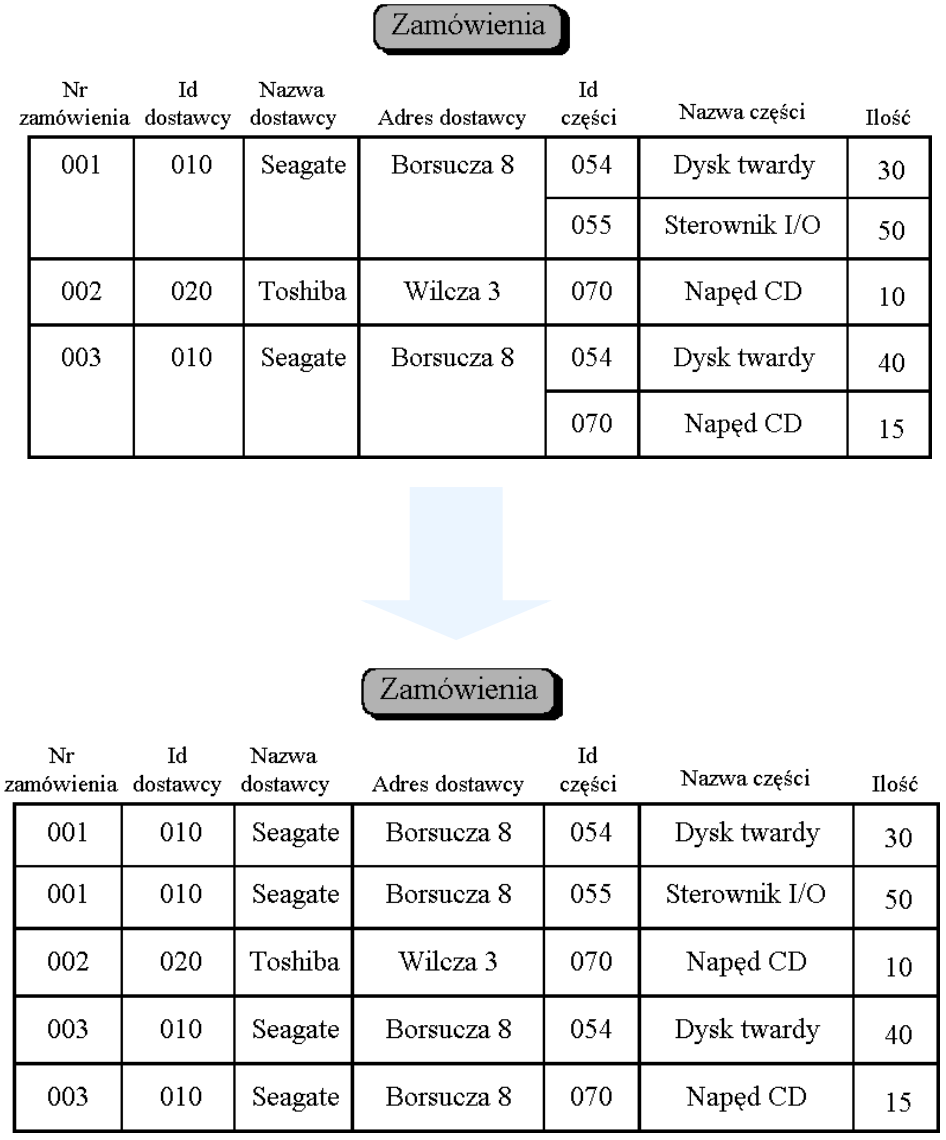
Zwykle doprowadzenie bazy danych do trzeciej postaci normalnej wystarcza do uznania schematów relacji za „dobre” do implementacji.

## Pierwsza postać normalna

Relacja jest w pierwszej postaci normalnej, jeśli wartości atrybutów są elementarne tzn. są to pojedyncze wartości określonego typu, a nie zbiory wartości.

Pierwsza postać normalna jest konieczna aby, tabelę można było nazwać relacją.
Większość systemów baz danych nie ma możliwości zbudowania tabel nie będących w pierwszej postaci normalnej.

Przekształcenie z postaci nie znormalizowanej do pierwszej postaci normalnej ilustruje rysunek:



## Druga postać normalna

Relacja jest w drugiej postaci normalnej, jeżeli każdy atrybut wtórny (tzn. nie wchodzący w skład żadnego klucza potencjalnego) tej relacji jest w pełni funkcjonalnie zależny od wszystkich kluczy potencjalnych tej relacji.

Można zauważyć, że relacja "Zamówienia" nie jest w drugiej postaci normalnej, ponieważ atrybuty "Id dostawcy", "Nazwa dostawcy", "Adres dostawcy" i "Nazwa części" nie są w pełni funkcjonalnie zależne od jedynego klucza potencjalnego - pary ("Nr zamówienia", "Id części").

Zamówienia

| Nr zamówienia | Id dostawcy | Nazwa dostawcy | Adres dostawcy | Id części | Nazwa części | Ilość |
|---|---|---|---|---|---|---|
| 001 | 010 | Seagate | Borsucza 8 | 054 | Dysk twardy | 30 |
| 001 | 010 | Seagate | Borsucza 8 | 055 | Sterownik I/O | 50 |
| 002 | 020 | Toshiba | Wilcza 3 | 070 | Napęd CD | 10 |
| 003 | 010 | Seagate | Borsucza 8 | 054 | Dysk twardy | 40 |
| 003 | 010 | Seagate | Borsucza 8 | 070 | Napęd CD | 15 |

W celu sprowadzenia relacji do drugiej postaci normalnej, należy podzielić ją na takie relacje, których wszystkie atrybuty będą w pełni funkcjonalnie zależne od kluczy. W tym celu przykładową relację "Zamówienia" należy podzielić na trzy relacje: "Dostawca na zamówieniu", "Zamówione dostawy", "Części" w następujący sposób:

Dostawca na zamówieniu

| Nr zamówienia | Id dostawcy | Nazwa dostawcy | Adres dostawcy |
|---|---|---|---|
| 001 | 010 | Seagate | Borsucza 8 |
| 002 | 020 | Toshiba | Wilcza 3 |
| 003 | 010 | Seagate | Borsucza 8 |

Zamówione dostawy

| Nr zamówienia | Id części | Ilość |
|---|---|---|
| 001 | 054 | 30 |
| 001 | 055 | 50 |
| 002 | 070 | 10 |
| 003 | 054 | 40 |
| 003 | 070 | 15 |

Części

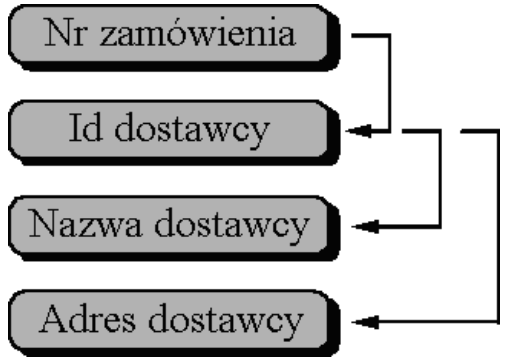
| Id części | Nazwa części |
|---|---|
| 054 | Dysk twardy |
| 055 | Sterownik I/O |
| 070 | Napęd CD |

Jak widać wszystkie te trzy relacje są w drugiej postaci normalnej, ponieważ klucze relacji "Dostawca na zamówieniu" oraz "Części" są kluczami prostymi, natomiast atrybut "Ilość" w relacji "Zamówione dostawy" jest w pełni funkcjonalnie zależny od klucza złożonego ("Nr zamówienia", "Id części").

Należy zauważyć, że relacja będąca w pierwszej postaci normalnej, jest równocześnie w drugiej postaci normalnej, jeśli wszystkie jej klucze potencjalne są kluczami prostymi.

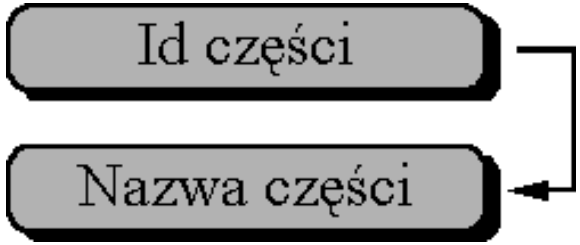
Po przekształceniu relacji "Zamówienia" do drugiej postaci normalnej otrzymujemy następujące zależności funkcjonalne:

- Dostawca na zamówieniu




- Zamówione dostawy

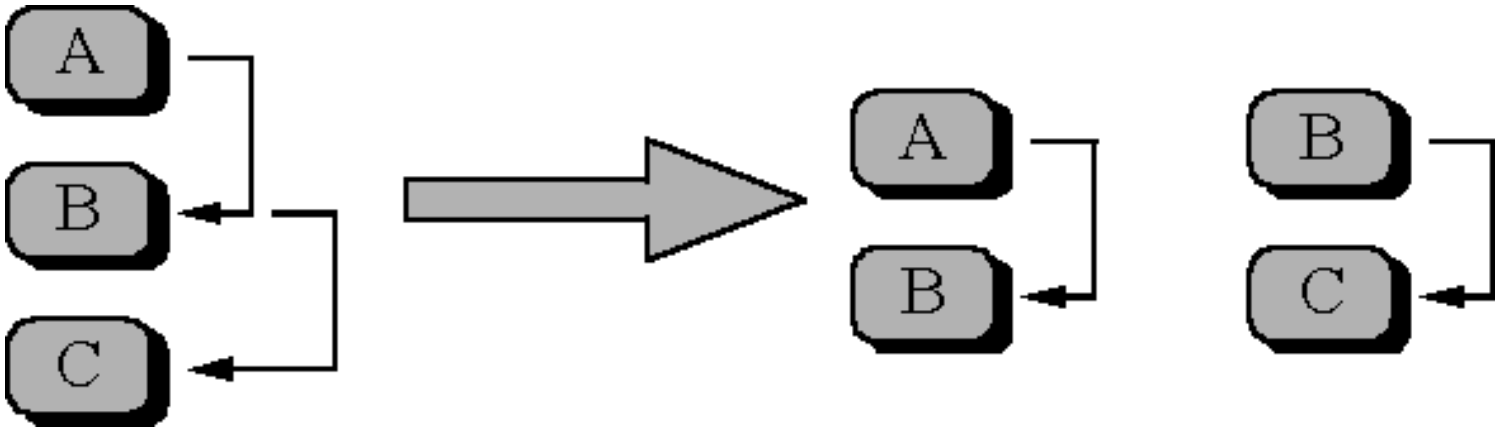

- Części



## Trzecia postać normalna

Dana relacja jest w trzeciej postaci normalnej, jeśli jest ona w drugiej postaci normalnej i każdy jej atrybut nie wchodzący w skład żadnego klucza potencjalnego nie jest przechodnio funkcjonalnie zależny od żadnego klucza potencjalnego tej relacji.

Aby doprowadzić relację, której atrybuty pozostają w przechodniej zależności funkcjonalnej, należy podzielić ją na relacje zawierające tylko zależność funkcjonalną. Podział relacji ilustruje rysunek:



W opisywanym przykładzie przechodnia zależność funkcjonalna występuje pomiędzy atrybutami "Nazwa dostawcy" i "Adres dostawcy" a atrybutem "Nr zamówienia" w relacji "Dostawca na zamówieniu". W związku z tym konieczne jest dokonanie podziału relacji "Dostawca na zamówieniu" na dwie relacje: "Zamówienia" i "Dostawcy" w następujący sposób:

Zamówienia

| Nr zamówienia | Id dostawcy |
|---|---|
| 001 | 010 |
| 002 | 020 |
| 003 | 010 |

Dostawcy

| Id dostawcy | Nazwa dostawcy | Adres dostawcy |
|---|---|---|
| 010 | Seagate | Borsucza 8 |
| 020 | Toshiba | Wilcza 3 |
| 030 | Sony | Ptasia 15 |

## Podsumowanie

Przekształcenie relacji do kolejnych postaci normalnych wiąże się najczęściej ze zmniejszeniem ilości pamięci potrzebnej do przechowania informacji.

Proces normalizacji ma na celu takie przekształcenie relacji, by uniknąć dublowania informacji. Unikanie powtórzeń pozwala na łatwiejsze i w wielu przypadkach szybsze posługiwanie się bazą danych.