



Bazy Danych

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytetu Gdańskiego

materiały dostępne elektronicznie
<http://inf.ug.edu.pl/~amb>

© Andrzej M. Borzyszkowski

Bazy Danych

Model relacyjny

© Andrzej M. Borzyszkowski

Bazy Danych

2/23

Dane w jednej tabeli

I Oddział	Sopot	Anna	Drygas	referent	4200
I Oddział	Sopot	Kazimierz	Moskal	asystent	4750
I Oddział	Sopot	Patrycja	Songin	referent	4200
I Oddział	Gdynia	Mariusz	Rumak	referent	4200
I Oddział	Gdynia	Bartosz	Ostrowski	kierownik	6000
I Oddział	Gdynia	Anna	Nehrebecka	dyrektor	9500

- Nadmiarowość danych
 - powtarzalna pełna nazwa oddziału
 - pensja zależna tylko od stanowiska
- Wady
 - niespójność danych (różne pensje dla różnych referentów)
 - nie ma gdzie wpisać pensji prezesa (aktualny brak w/w)
 - nie ma gdzie wpisać danych nowego oddziału (jeszcze bez pracowników)

© Andrzej M. Borzyszkowski

Bazy Danych

3/23

Dane w strukturze hierarchicznej

I Oddział	Sopot		
Anna	Drygas	referent	4200
Kazimierz	Moskal	asystent	4750
Patrycja	Songin	referent	4200

Nadmiarowość danych

- pensja zależna tylko od stanowiska

Wady

- niespójność danych
- nie ma gdzie wpisać pensji prezesa
- wyszukiwanie wg stanowiska wymaga przeczesania całości

I Oddział	Gdynia		
Mariusz	Rumak	referent	4200
Bartosz	Ostrowski	kierownik	6000
Anna	Nehrebecka	dyrektor	9500

II Oddział	Gdynia		
------------	--------	--	--

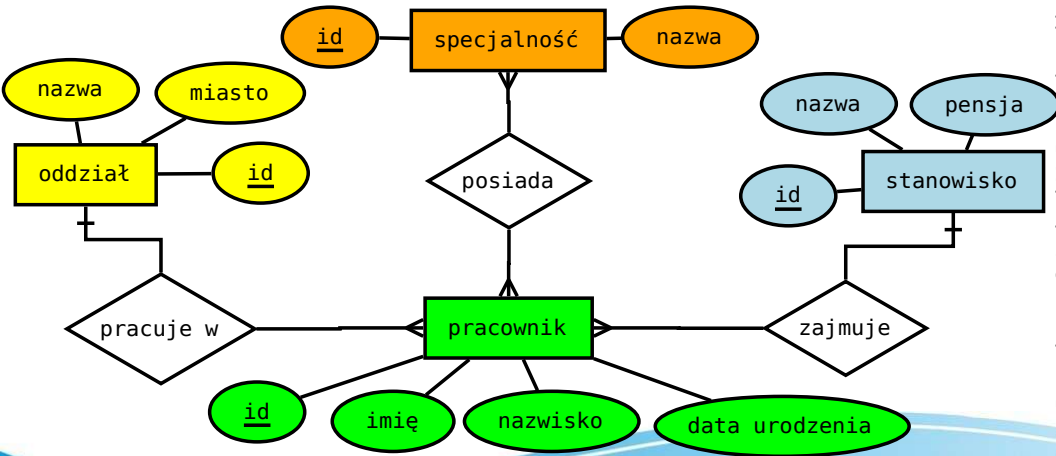
© Andrzej M. Borzyszkowski

Bazy Danych

4/23

Pracownicy w banku – ERD

Relacyjne bazy danych



© Andrzej M. Borzyszkowski

Bazy Danych

5/23

- Dane przechowywane są w tabelach dla każdego rodzaju encji
 - wyszukiwanie wg kluczy głównych (indeksy)
 - powiązanie za pomocą kluczy obcych
- Zalety
 - brak redundancji – każda informacja jest w jednej kopii
 - wyszukiwanie wg różnych kryteriów tak samo łatwe

I Oddział	Sopot	1
I Oddział	Gdynia	2
II Oddział	Gdynia	3

15	Anna	Drygas	1
12	Kazimierz	Moskal	1
15	Patrycja	Songin	1
15	Mariusz	Rumak	2
11	Bartosz	Ostrowski	2
13	Anna	Nehrebecka	2

11	kierownik	6000
12	asystent	4750
13	dyrektor	9500
14	prezes	25000
15	referent	4200

© Andrzej M. Borzyszkowski

Bazy Danych

23

Idea: relacyjna baza danych = tabele

- Relacyjna baza danych: tabele + operacje na tabelach
- Pojedyncza tabela, np. arkusz Excela:

nr	tytuł	nazwisko	imie	kod_poc	miasto	ulica_dom	telefon
1	Pani	Kuśmerek	Małgorzata	81-124	Gdynia	NULL	058 6252840
2	Pan	Chodkiewicz	Jan	81-737	Gdynia	Chwarznińska 33/5	058 6240860
3	Pani	Szczęsna	Jadwiga	81-444	Gdynia	Bema 41a/12	058 6243741
4	Pan	Łukowski	Bernard	81-620	Gdynia	Górnicza 29	058 6230799
5	Pan	Soroczyński	Jan	80-230	Gdańsk	Al. Hallera	058 3090788
6	Pani	Niezbittowska-Na	Marzena	80-619	Gdańsk	Focha 39-41 m.66	058 3099102
7	Pani	Kolak	Agnieszka	80-832	Gdańsk	Wawóz 4	NULL
8	NULL	Hałasa	Ewa	80-511	Gdańsk	Dywigjonu 303/303	058 3483240

© Andrzej M. Borzyszkowski

Bazy Danych

7/23

- wiersz jest rekordem, dane jednej osoby
- każda kolumna jest przeznaczona na pewną cechę
 - numer służy głównie do identyfikacji osoby (jest kluczem)
- Terminologia matematyczna: tabela ≈ relacja

Arkusz kalkulacyjny, ograniczenia

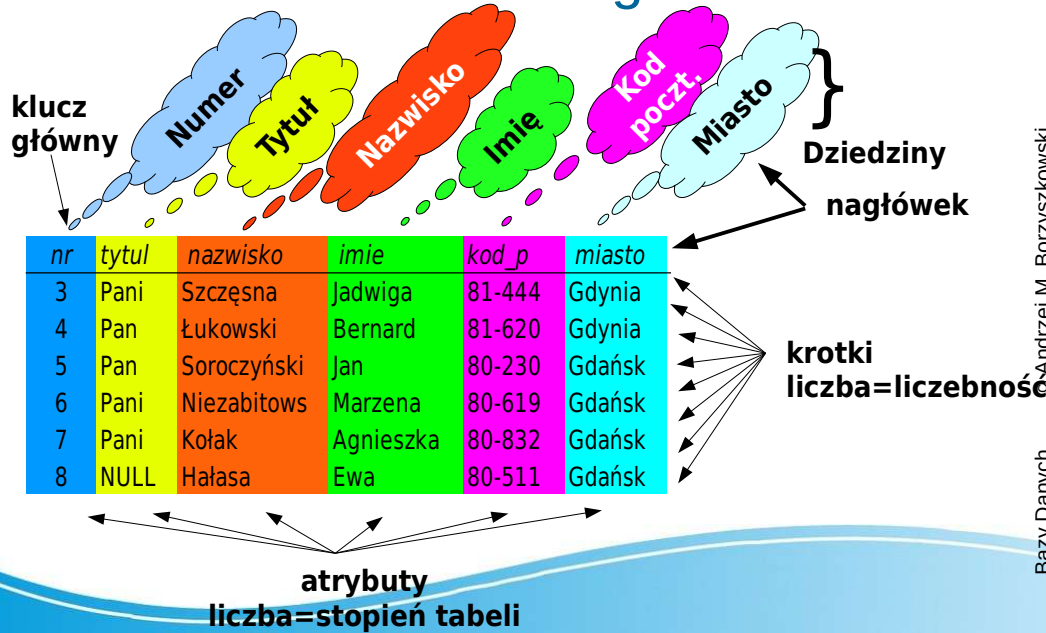
- Problem ze współbieżnością:
 - wielu użytkowników korzysta z tych samych danych
 - jeden plik nie może być równocześnie edytowany przez kilku użytkowników
- Problem z wydajnością:
 - proste operacje wyszukiwania, sortowania i inne stają się zbyt złożone jeśli danych jest bardzo dużo
- Problem z nadmiarem danych:
 - niektóre dane powtarzają się, zajmują miejsce, kopie mogą być niedokładne
- Problem z powtarzalnością pól:
 - struktura tabeli nie przewiduje powtórzeń, np. wiele imion
 - albo zaliczonych przedmiotów

© Andrzej M. Borzyszkowski

Bazy Danych

8/23

Terminologia



© Andrzej M. Borzyszkowski

Bazy Danych

9/23

Podstawy teoretyczne

- Matematyka: relacja r to podzbiór iloczynu kartezjańskiego
 - czyli zbiór "krotek" (*tuple*)
 - czyli zbiór rekordów mających (nazwane) pola o wartościach w pewnych zbiorach możliwych wartości
 - uwaga: dopuszczamy również wartość „NULL”
 - $r \subset D1 \times D2 \times D3 \times \dots \times Dn$
 - pola mają (różne) nazwy: np. $A1, A2, \dots, An$, gdzie $A1:D1, A2:D2, \dots, An:Dn$, kolejność jest nieważna,

$D1 \times D2 \times \dots \times Dn =$

$\{x \mid x.A1 \in D1, x.A2 \in D2, \dots, x.An \in Dn\} =$

$\{\{ \langle A1, x1 \rangle, \langle A2, x2 \rangle, \dots, \langle An, xn \rangle \} \mid x1 \in D1, x2 \in D2, \dots, xn \in Dn\}$

- zbiór nie ma powtórzeń
- kolejność elementów w zbiorze jest nieokreślona

© Andrzej M. Borzyszkowski

Bazy Danych

10/23

Relacje a tabele

- Matematyczne własności relacji:
 - nie ma podwójnych krotek
 - krotki są nieuporządkowane
 - atrybuty są nieuporządkowane
- Zawartość tabeli jest uporządkowana, i wiersze i kolumny
 - może zawierać powtórzenia wierszy
 - formalnie nie jest relacją
 - ale można ją uważać za przedstawienie relacji

© Andrzej M. Borzyszkowski

Bazy Danych

11/23

Relacje a schematy relacji

- Schematem relacji (*schema*) jest jej „typ”
 - $R(A1, A2, \dots, An)$
 - każdy atrybut A ma przypisaną dziedzinę $dom(A)$
 - dziedziny można wymienić w schemacie relacji (w praktyce typ dziedziny)
 - $Klient(nr:integer, tytuł:string, nazwisko:string, imię:string, kod_poczt:integer, miasto:string)$
 - inna nazwa: R - zmienna relacyjna, intensja relacji
- Relacja to konkretny zbiór krotek
 - $r(R) \subset dom(A1) \times dom(A2) \times dom(A3) \times \dots \times dom(An)$
 - inna nazwa: r - stan relacji, bieżący stan relacji, ekstensja relacji
- Podobnie nagłówek tabeli vs treść tabeli

© Andrzej M. Borzyszkowski

Bazy Danych

12/23

Ograniczenia modelu relacyjnego a schematy

- Ograniczenia wynikające z modelu
 - wartości atrybutów są atomowe
 - nie ma powtórzeń wierszy (dwóch krotek o identycznych atrybutach)
- Ograniczenie wyrażane explicite w schemacie
 - wartości atrybutów muszą należeć do dziedzin
 - pewne atrybuty nie mogą przyjmować NULL
 - wartości kluczowe: nie dwóch krotek o identycznych wartościach pewnych atrybutów
 - inne ograniczenia
- Ograniczenia wymuszane przez aplikacje zewnętrzne
 - niektóre mogą być alternatywnie wyrażalne w schemacie, ale nie muszą

© Andrzej M. Borzyszkowski

Bazy Danych

13/23

Relacyjne bazy danych i ich schematy

- Schemat relacyjnej bazy danych
 - zbiór schematów relacji
 - oraz zbiór więzów integralności
- Stan relacyjnej bazy danych
 - zbiór stanów (relacji) dla każdego ze schematów relacji bazy danych
 - spełniających więzy integralności
- Zbiór stanów *nie* spełniających więzów integralności nazywamy stanem niespójnym bazy danych
 - rozważamy tylko teoretycznie, nie powinien istnieć
- Język definiowania danych: *Data Definition Language*, służy do definiowania schematów baz danych

© Andrzej M. Borzyszkowski

Bazy Danych

14/23

Więzy integralności (t.j. spójności)

- Warunki, które musi spełniać każdy stan bazy danych
 - SZBD sprawdza więzy integralności przy każdej operacji na bazie danych
 - w przypadku niespełnienia warunków następuje odrzucenie operacji
 - albo inne rozwiązanie
- Rodzaje warunków:
 - wartości w odpowiedniej dziedzinie
 - w tym różne od NULL
 - jednoznaczność (klucz kandydujący)
 - istnienie (integralność referencyjna)
 - inne warunki określone w schemacie

© Andrzej M. Borzyszkowski

Bazy Danych

15/23

Klucze kandydujące

- Klucz kandydujący schematu relacji R jest zbiorem K atrybutów relacji takim, że
 - jednoznaczność: żadne dwie różne krotki relacji $r(R)$ nie mają tej samej wartości dla K
 - nieredukowalność: żaden podzbiór właściwy K nie posiada powyższej własności
- Przykłady:
 - tabela **Pierwiastki_chemiczne** dla układu okresowego pierwiastków posiada pola *nazwa*, *symbol*, *liczba atomowa*
 - każde pole jednoznacznie identyfikuje pierwiastek, każde pole jest kluczem kandydującym
 - tabela **Pozycja** ma klucz kandydujący złożony z dwu atrybutów, numer zamówienia i numer towaru, żaden pojedynczy atrybut nie jest kluczem kandydującym

© Andrzej M. Borzyszkowski

Bazy Danych

16/23

Klucz główny, klucze alternatywne

- Kluczem głównym schematu relacji nazywamy jeden wybrany klucz kandydujący
 - pozostałe klucze kandydujące nazywamy kluczami alternatywnymi
- Integralność encji – wartość klucza głównego nie może być NULL
- Pojęcia klucza można stosować do schematu relacji ale również do relacji (tj. bieżącego stanu)
 - np. numer indeksu jest kluczem głównym dla schematu relacji Student w bazie danych Szkoła Wyższa
 - imię i nazwisko nie jest kluczem kandydującym
 - jest (prawdopodobnie) kluczem dla bieżącej wartości relacji student na wydziale MFI

© Andrzej M. Borzyszkowski

Bazy Danych

17/23

Klucz obcy

- Kluczem obcym schematu relacji R2 odwołującym się do schematu relacji R1 (być może = R2) nazywamy zbiór atrybutów FK taki, że
 - wartości atrybutów FK należą do tej samej dziedziny co wartości wskazywane CK
 - CK jest kluczem kandydującym w R1
 - dla każdej krotki w R2 wartość klucza FK jest równa wartości klucza CK pewnej krotki w R1
 - tzn. klucz obcy R2 „wskazuje” krotkę w R1, krotka jest jednoznaczna, skoro CK jest kluczem kandydującym
 - najczęściej klucze obce składają się z jednego atrybutu
 - dopuszcza się, by klucz obcy miał wartość NULL (wówczas niczego nie musi wskazywać)

© Andrzej M. Borzyszkowski

Bazy Danych

18/23

Klucz obcy, przykłady

- W schemacie zamówienie będzie klucz obcy wskazujący na klienta
 - pozycja zamówienia wskazuje na zamówienie oraz na towar (dwa klucze obce)
 - schematy zapas i kod_kreskowy również zawierają klucze wskazujące na towar
- W schemacie szkoły wyższej będzie klucz obcy w tabeli przedmiotów wskazujący na prowadzącego
 - będzie NULL przed dokonaniem obsady zajęć
- Klucz obcy może wskazywać na klucz kandydujący we własnej tabeli
 - np. w bazie danych pracowników można zapisywać bezpośredniego przełożonego
 - wówczas, oczywiście, co najmniej jedna krotka musi mieć wartość NULL tego klucza

© Andrzej M. Borzyszkowski

Bazy Danych

19/23

Klucz obcy, wymagania

- Wartość klucza obcego występująca w relacji musi pojawić się jako wartość odpowiadającego klucza kandydującego
 - ale odwrotna zależność nie jest wymagana
 - np. istnieją towary niezamawiane, klienci, którzy nie złożyli żadnego zamówienia, a nawet zamówienia bez pozycji
- Wartość klucza obcego stanowi odwołanie (*reference*) do krotki zawierającej wartość odpowiadającego mu klucza kandydującego (adresat odwołania)
- Integralność referencyjna: warunek by baza danych w żadnym stanie nie zawierała wartości klucza obcego nieobecnych we wskazywanej tabeli (*dangling references*)

© Andrzej M. Borzyszkowski

Bazy Danych

20/23

Operacje na bazie danych – obsługa naruszenia więzów integralności

- Operacje na relacjach
 - wstawianie krotki
 - usuwanie krotki
 - modyfikacja krotki (zmiana wartości atrybutów danej krotki)
- Wstawianie
 - niewłaściwa wartość atrybutu lub niedozwolona wartość NULL → odrzucenie operacji
 - powtórzona wartość klucza kandydującego → odrzucenie operacji
 - klucz obcy odwołujący się do nieistniejącej krotki → odrzucenie operacji

© Andrzej M. Borzyszkowski

Bazy Danych

21/23

Obsługa naruszenia więzów integralności 2

- Usuwanie
 - może naruszyć tylko integralność referencyjną (usuwanie adresata odwołania klucza obcego)
 - opcja 1: → odrzucenie operacji
 - opcja 2: operacja usuwania jest propagowana (*cascade*) do powiązanych krotek
 - opcja 3: wartości klucza obcego ustawiane są na NULL (o ile schemat to dopuszcza)
- PostgreSQL, domyślne zachowanie: nie wolno usuwać adresata istniejącego odwołania: ON DELETE NO ACTION
 - opcja usuwania kaskadowego: ON DELETE CASCADE
 - lub ustawiania wartości NULL: ON DELETE SET NULL

© Andrzej M. Borzyszkowski

Bazy Danych

22/23

Obsługa naruszenia więzów integralności 3

- Modyfikacja
 - niewłaściwa wartość atrybutu lub niedozwolona wartość NULL → odrzucenie operacji
 - powtórzona wartość klucza kandydującego → odrzucenie operacji
 - modyfikowany klucz obcy odwołujący się do nieistniejącej krotki → odrzucenie operacji
 - modyfikowany klucz kandydujący będący adresatem odwołania pewnego klucza obcego → odrzucenie operacji,
 - albo modyfikacja klucza obcego tak by wskazywał na tę samą krotkę
- PostgreSQL, domyślne zachowanie: nie wolno modyfikować adresata istniejącego odwołania
 - specjalna opcja do modyfikacji kaskadowej

© Andrzej M. Borzyszkowski

Bazy Danych

23/23