



Relacyjne Bazy Danych

Andrzej M. Borzyszkowski
PJATK/ Gdańsk

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

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

Program wykładu

- Wstęp: historia, systemy zarządzania bazą danych
- Modelowanie danych: encje, związki, atrybuty
- Model relacyjny i algebra relacyjna
- Projektowanie baz danych
 - postaci normalne
- Język SQL – realizacja algebry relacyjnej
 - definiowanie danych
 - operowanie na danych: dostęp, aktualizacja
- Fizyczna organizacja plików, indeksy, optymalizacja
- Współbieżność, blokady, transakcje
- Integracja ze środowiskiem programistycznym

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

2

Bibliografia, oprogramowanie

- R. Elmasari, S. B. Navathe, *Wprowadzenie do systemów baz danych*, Helion, 2005
- R. Stones, N. Matthew, *Bazy danych i PostgreSQL*, Helion, 2002 (seria Wrox)
- Jeffrey D. Ullman, Jennifer Widom, *Podstawowy wykład z systemów baz danych*, WNT, 2001 (seria Klasyka Informatyki)
- Praca w laboratorium:
 - system PostgreSQL (open source, dostępny na systemy Linux i Windows)
 - każdy student ma swoją bazę danych na serwerze szuflandia
 - dostęp do bazy w trybie tekstowym (terminal) i graficznym (dedykowany program, przeglądarka)

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

3

Dane

- Przechowywanie danych:
 - bank: wszystkie transakcje
 - sprzedaż: klienci, towary
 - produkcja: części, dostawcy, proces produkcyjny
 - administracja państwowa: dane osobowe, miejsce zamieszkania, samochody
 - urząd skarbowy: dochody, podatki
 - szkoła wyższa: studenci, pracownicy, proces dydaktyczny
- Komputery służą (były zaprojektowane) do obliczeń
 - ale chcemy je użyć do przechowywania i przetwarzania danych

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

4

Technologia

- Technologia komputerowa
 - plik: sekwencyjny zapis danych, dobry np. dla muzyki/filmu
 - albo trochę struktury: wiersze z polami, znaczniki

```
hplip:x:107:7:HPLIP system user,,,:/var/run/hplip:/bin/false
gdm:x:108:113:Gnome Display Manager:/var/lib/gdm:/bin/false
amb:x:1000:1000:Andrzej Borzyszkowski,,,:/home/amb:/bin/bash
postgres:x:111:115:PostgreSQL
administrator,,,:/var/lib/postgresql:/bin/bash
```
 - rekordy i pola
- indeksy: dodatkowy plik zawierający adresy rekordów wyszukiwanych wg klucza
- tzw. hasze (skrót): sam klucz wyznacza adres
 - gdy zachodzi kolizja adresów, to wyszukiwanie sekwencyjne

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

5

Baza danych

- Zbiór powiązanych ze sobą informacji
 - opisujących *fakty i zdarzenia*
 - zachodzące w pewnym wycinku rzeczywistości
 - przechowywanych w sposób *trwały*
 - zorganizowanych w *strukturę* pozwalającą na ich szybkie wyszukiwanie i analizę
- Baza danych jest projektowana, konstruowana i wypełniana danymi
 - w określonym *celu*, ma określone zastosowania,
 - ma określonych *użytkowników*.

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

7

Technologia – problemy

- Problemy:
 - nieregularność danych, różna liczba pól w rekordach, różna wielkość rekordów
 - wielkość danych (tzn. liczba rekordów), wydajność
 - warunki spójności
 - nie są wyrażone bezpośrednio
 - nie są też gwarantowane
 - problem dostępu współbieżnego

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

6

System Zarządzania Bazą Danych

- Zestaw programów umożliwiających *definiowanie, konstruowanie* baz danych, *manipulowanie i udostępnianie* zawartych w nich danych oraz *ochronę i konserwację*
 - SZBD (DBMS – *database management system*)
- *System bazy danych* = baza danych + system zarządzania bazą danych

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

8

Znane i lubiane implementacje SZBD

- Oracle
 - MySQL
 - MicroSoft SQL Server
 - PostgreSQL
 - DB2
 - Sybase
 - Informix
 - dBASE
 - Ingres
 - Adabas
 - Paradox (starość)
 - SQLite (niezupelnie)
 - MS Access (czy to w ogóle SZBD?)
- <http://is.gd/buzdrE> wikipedia:
Lista_systemów_zarządzania_relacyjnymi_bazami_danych
- http://db-engines.com/en/ranking_trend lista rankingowa

© Andrzej M. Borzyszkowski
Relacyjne Bazy Danych

9

Otoczenie programistyczne

- Aplikacje zewnętrzne
- Programy do budowy aplikacji
- Programy narzędziowe (np. kopie zapasowe)
- Arkusze kalkulacyjne,
 - pakiety statystyczne,
 - inne programy do analiz wykorzystujących zgromadzone dane,
 - programy do grafiki,
 - edytory raportów, etc

© Andrzej M. Borzyszkowski
Relacyjne Bazy Danych

10

Architektura klient-serwer

- Aplikacja bazodanowa
 - na serwerze zawierającym SZBD
 - na komputerze użytkownika (klient)
- Funkcje serwera
 - zarządzanie bazą danych (w tym dbałość o spójność danych)
 - zarządzanie kontami użytkowników
 - wykonywanie poleceń przekazanych przez klienta
- Funkcje klienta
 - kontakt z użytkownikiem (interfejs, np.. graficzny)
 - wykonywanie lokalne obliczeń
 - komunikacja z serwerem
 - prezentacja danych otrzymanych od serwera

© Andrzej M. Borzyszkowski
Relacyjne Bazy Danych

11

Języki zapytań

- Języki zapytań (*query*)
 - dawniej przewidywane zapytania sterowały projektem bazy
 - inne zapytania były bardzo nieefektywne
 - bazy relacyjne są neutralne
- Pierwsze języki zapytań
 - QBE (*query by example*) – zapytanie przez przykład
 - SQL – standardowy język zapytań
 - dziś rozwinął się do powszechnego standardu
- SQL
 - manipulowanie danymi – wstawianie, usuwanie, wyszukiwanie
 - definiowanie danych – tworzenie tabel
 - sterowanie danymi – np. prawa dostępu w bazie danych

© Andrzej M. Borzyszkowski
Relacyjne Bazy Danych

12

Przykład

- Baza danych „uniwersytet”:
 - student (imię i nazwisko, rok i kierunek studiów, indeks)
 - przedmiot (nazwa i skrót, kierunek, l.godzin)
 - obsada (przedmiot, prowadzący, semestr)
 - oceny (student, przedmiot, ocena)
 - wymagania (przedmiot, co wymaga)
- Dane podzielone są na wiele plików,
 - plik składa się z rekordów (wiersze/krotki, *tuple*)
 - kolumny/pola/attributy, stała liczba i format
 - komórki są atomowe
 - mają one ustalone z góry typy (napis/liczba/data ...)

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

13

Cechy systemów baz danych

- Opis struktury
 - SZBD przechowuje *katalog* czyli informację o strukturze wszystkich plików bazy danych
 - również informacje o użytkownikach i ich uprawnieniach
 - SZBD jest przystosowany do obsługi dowolnej bazy, struktura nie jest częścią aplikacji
- Abstrakcja danych
 - aplikacja jest niezależna od struktury bazy danych
 - istnieje możliwość dodania pól, połączenia tabel, zmiany organizacji wewnętrznej, etc.
 - istnieje możliwość zmiany sposobu dostępu do danych
 - w podejściu obiektowym aplikacja może być niezależna od operacji: wywołuje metody, implementacja operacji w metodzie

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

15

Baza danych vs. dane rozproszone

- Integracja danych
 - baza może być traktowana jako połączenie informacji zawartych w odrębnych plikach danych
 - przykład: dziekanat posługuje się tabelą studentów i ocen by analizować zaliczenia, inny dział oblicza wynagrodzenie pracowników korzystając z tabel obsady danych
- Gdyby każdy z działów miał swoje dane:
 - nadmiarowość
 - niepotrzebnie zajęte miejsce
 - niebezpieczeństwo niespójności danych
 - odmienny format danych w każdym dziale
 - być może utrudniłoby to wymianę

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

14

Cechy systemów baz danych, c.d.

- Spójność (integralność) danych
 - system pozwala zdefiniować własności wymagane od danych
 - system sprawdza te własności
 - zmniejsza ryzyko zapisania błędnych danych w bazie (błędy użytkowników)
 - zmniejsza/niweluje ryzyko błędów sprzętowych/awarii
 - zmniejsza/niweluje ryzyko błędów wskutek dostępu współbieżnego

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

16

Cechy systemów baz danych, c.d.

- Poufność danych, obsługa perspektyw
 - różni użytkownicy mają dostęp do różnych danych
 - nie każdy użytkownik bazy powinien mieć dostęp do wszystkich danych,
 - dane zbiorcze/indywidualne, czytanie/zapis
 - może nawet nie wiedzieć o istnieniu niektórych danych
- Autoryzacja dostępu: system kont z hasłami, różne systemy identyfikacyjne
- Szyfrowanie danych

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

17

Zalety rozwiązań bazodanowych

- Ograniczanie nadmiarowości (redundancji)
 - większy wysiłek przy wprowadzaniu danych
 - większa zajętość miejsca (dziś mniej ważny argument)
 - niespójność danych (błędy w niektórych kopiach, rozbieżność wprowadzanych danych)
- Ale: nadmiarowość może być pożyteczna
 - dane bliżej użytkownika końcowego
 - dane wynikowe przechowywane w celu dalszego użycia
 - wniosek: nadmiarowość trzeba kontrolować

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

19

Cechy systemów baz danych, c.d.

- Współdzielenie danych
 - fragmenty danych mogą być używane przez wielu użytkowników jednocześnie (współbieżność)
 - problem czytelników i pisarzy
 - *transakcja* - niepodzielna operacja dokonana przez jednego z użytkowników, izolowana od innych operacji
- Niezawodność
 - możliwość odtworzenia bazy sprzed awarii sprzętowej czy programowej
- Wydajność
 - struktura odpowiednia do wyszukiwania danych (indeksy)

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

18

Zalety rozwiązań bazodanowych, c.d.

- Różnorodność interfejsów
 - baza danych jest jedna, ale różni użytkownicy mogą różnie ją widzieć
 - narzędzia graficzne, formatki do wprowadzania danych, graficzne przedstawienie danych
 - dostęp poprzez witrynę internetową
 - dostęp poprzez SQL
- Definiowanie reguł
 - automatyczne wnioskowanie na podstawie danych
 - automatyczne podejmowanie odpowiednich działań
 - również dbałość o spójność (wartości czy zależności pomiędzy danymi, *business rules*)

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

20

System BD – użytkownicy

- Użytkownik końcowy
 - komunikuje się z bazą, np. ze stacji roboczej
 - realizuje swoje cele za pomocą udostępnionego mu interfejsu (system formularzy, procesora zapytań SQL)
- Rodzaje użytkowników końcowych:
 - dorywczy: inne potrzeby za każdym podejściem
 - naiwny użytkownik: standardowe i powtarzalne czynności, używa formularzy
 - doświadczony użytkownik: wykonuje niestandardowe operacje, używa SQL

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

21

Modelowanie danych (model związków encji)

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

23

System BD – użytkownicy

- Programista aplikacji
 - określa wymagania użytkowników końcowych (naiwnych)
 - tworzy programy umożliwiające użytkownikom końcowym dostęp do bazy
- Administrator
 - specjalista z dziedziny IT (*Information Technology*)
 - zakłada bazę danych, implementuje kontrolę dostępu do bazy, monitoruje wykorzystanie, odpowiada za wydajność systemu i za bezpieczeństwo danych
- Projektant bazy
 - identyfikuje dane do przechowania, projektuje struktury, przewiduje perspektywy dla różnych użytkowników

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

22

Modelowanie rzeczywistości

- Model semantyczny: „rozumiemy” modelowaną rzeczywistość
 - potem planujemy jej reprezentację

Projektowanie bazy danych: analiza wymagań

- wymagania funkcjonalne (planowane operacje)
 - diagramy przepływu danych, diagramy sekwencji, scenariusze (inżynieria oprogramowania)
 - stosowane są diagramy UML (*unified modelling language*)
- wymagania danych
 - schemat koncepcyjny: decyzje biznesowe (*bussiness logic*) – co chcemy przechowywać?
 - jakie operacje chcemy wykonywać
 - warunki spójności narzucane na dane

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

24

Modelowanie rzeczywistości, c.d.

- Modele historyczne
 - model hierarchiczny (np. drzewo katalogów systemu operacyjnego)
 - model sieciowy
- Model relacyjny (Peter Chen 1976)
 - dane tworzą relację/wiele relacji
 - relacja \approx tabela
 - diagramy związków encji – *entity relationship diagrams*
- Modele przyszłości ?
 - model obiektowo-relacyjny
 - model semistrukturalny
 - itd.

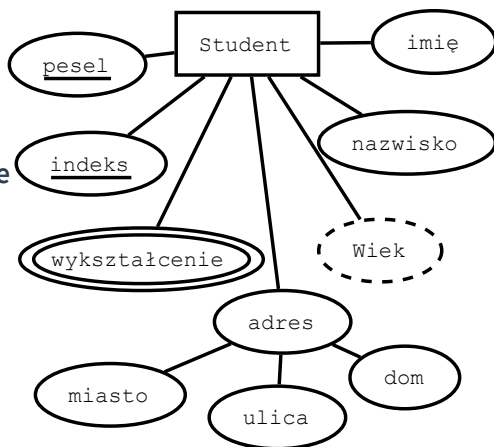
© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

25

Encje

- Encja (jednostka) jest opisywana atrybutami
 - np. imię, nazwisko, pesel (atrybuty proste)
 - mogą być atrybuty złożone (np. adres)
 - pochodne (np. wiek)
 - wielowartościowe (np. wykształcenie)



© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

27

Encje i związki

- Encja (*entity*): realny byt, jednostkowy i odróżnialny od innych podobnych encji, np. człowiek, przedmiot, organizacja
 - baza danych zawiera właśnie informacje o encjach
 - encje pewnego typu stanowią zbiór, ma on swoją nazwę
 - encje charakteryzują się własnościami.
- Własność (atrybut): cecha encji przechowywana w bazie danych
 - ma wartość w pewnym zbiorze właściwym dla tej własności
 - a priori może być złożona, wielowartościowa, pochodna.
- Klucz (*key*): jedna lub więcej własności jednoznacznie identyfikujących encję w bazie danych.
- Związek (*relationship*): zależność pomiędzy zbiorami encji w bazie danych, ma swoją nazwę.

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

26

Encje, c.d.

- Typ encji definiuje zbiór możliwych encji o tych samych atrybutach – schemat, intensja
- Ekstensja – chwilowy stan bazy danych, zbiór encji przechowywanych w danej chwili
- Atrybut kluczowy – dla każdej ekstensji atrybut jest niepowtarzalny
 - tzn. nigdy nie będą przechowywane dwie encje o tej samej wartości klucza
 - oznaczany jest jako podkreślenie nazwy
 - najczęściej jest to atrybut atomowy
 - może być kilka atrybutów kluczowych
- Dziedzina wartości atrybutu *nie jest* na diagramie reprezentowana
 - ani typ danych, ani dodatkowe ograniczenia

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

28

Związki

- Typ związku określa typy encji, pomiędzy którymi zachodzi związek oraz dopuszczalną liczbę elementów encji będących w związku
 - bieżący stan bazy danych określa istniejące powiązania dla danego związku
- Np. w bazie danych przechowywane są informacje o studentach, przedmiotach i zaliczeniach
 - zaliczenie jest związkiem pomiędzy encjami przedmiotów i studentów, związkiem wieloznacznym
 - w bazie danych przechowywane są bieżące informacje na powyższy temat, zmieniają się one w czasie
 - ale istnienie i typ związku jest niezmienny
- Prawie zawsze związki są binarne (pomiędzy dwiema encjami)

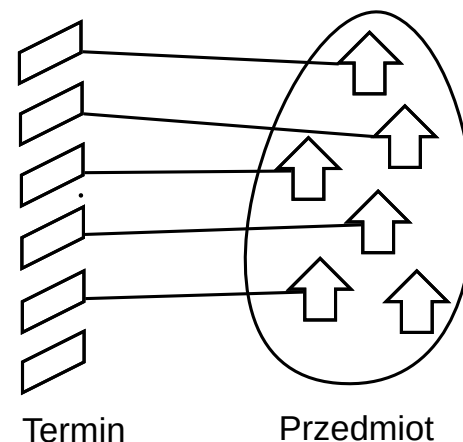
© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

29

Klasyfikacja (binarnych) związków encji

- 1-1 (jednoznaczny)
 - każda encja z jednego zbioru może być skojarzona z co najwyżej jednym elementem z drugiego zbioru
 - pewne encje mogą pozostać bez skojarzenia
 - czasami wyraźnie chcemy uniknąć takiej sytuacji
 - np. przedmiot ma pełen udział w związku oznacza, że każdy przedmiot ma przypisany termin – wymóg istnienia

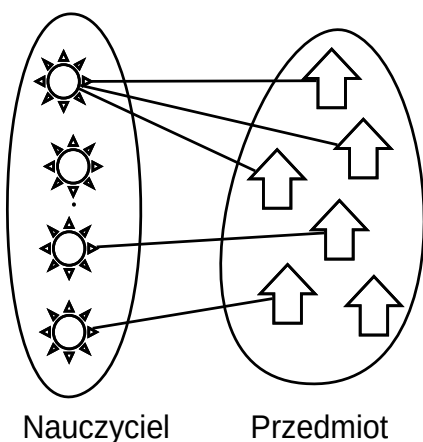


© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

30

Klasyfikacja (binarnych) związków encji



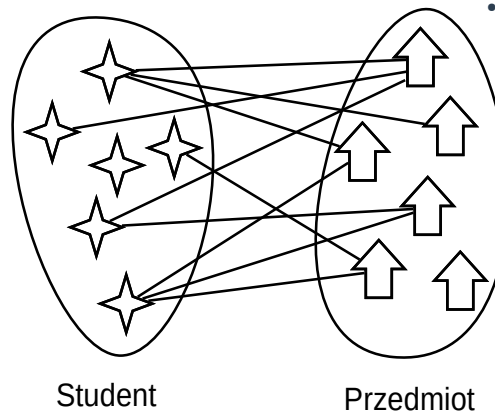
- 1-wielu, 1-N (jednoznaczny)
 - każda encja ze jednego zbioru może być skojarzona z pewną ich liczbą z drugiego zbioru
 - jednakże encja z drugiego zbioru najwyżej z jedną encją z pierwszego zbioru
 - i znowu mogą pozostać encje bez skojarzenia
 - ale czasami wyraźnie chcemy uniknąć takiej sytuacji
 - np. zapewnić, że przedmioty mają obsadę

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

31

Klasyfikacja (binarnych) związków encji



- wieloznacznym
 - dowolna liczba encji z jednego zbioru może być skojarzona z dowolną liczbą encji z drugiego zbioru
 - nadal aktualne uwagi o encjach niezwiązanych

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

32

Przykład

- Szkoła Wyższa organizuje bazę danych zawierającą informacje o *nauczycielach akademickich* (nazwisko, imię, nr legitymacji), *studentach* (nazwisko, imię, nr indeksu), wykładanych *przedmiotach* (nazwa, rodzaj, liczba godzin w tygodniu, kod) i ich *terminach* (dzień tygodnia, godzina, sala).
- Rozważamy też następujące związki między encjami:
 - odbywa się*: każdy przedmiot posiada określony termin/salę
 - związek jednojednoznaczny
 - jest prowadzony*: każdy przedmiot jest prowadzony przez nauczyciela, który prowadzi wiele przedmiotów
 - związek jednoznaczny
 - zalicza*: każdy student zalicza kilka przedmiotów, każdy z nich gromadzi wielu studentów, zaliczenia są na ocenę
 - związek wieloznaczny

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

33

Cechy związków

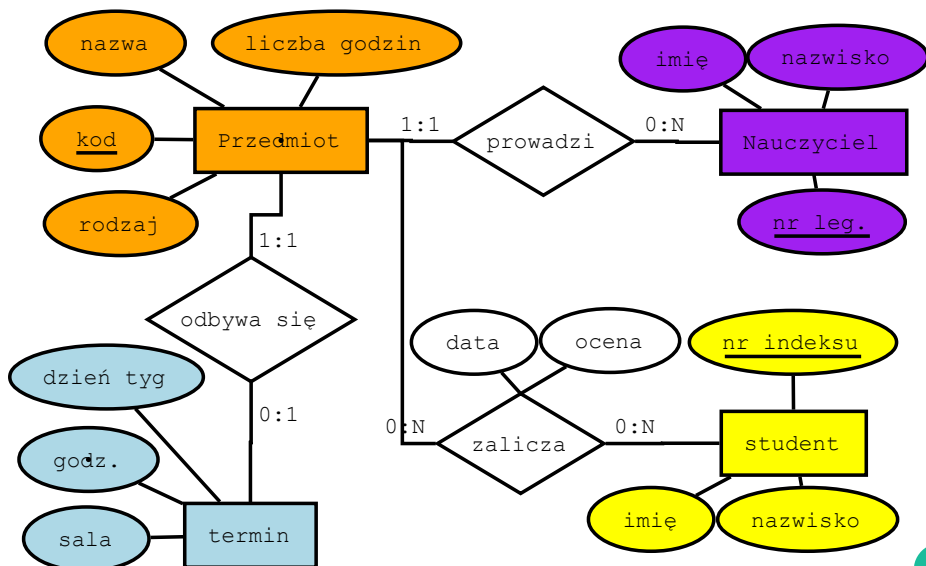
- Dla związku binarnego mamy dwie możliwe nazwy
 - student zalicza przedmiot – przedmiot jest uczęszczany
 - przedmiot odbywa się w terminie – termin jest zajęty przez
 - nauczyciel wykłada przedmiot – przedmiot jest wykładany
- Technicznie nie ma znaczenia jaką nazwę przyjmie
 - ale musi być jasna w przypadku związku rekursywnego
 - np. pracownik jest kierownikiem innego pracownika
- Na diagramie reprezentuje się dokładniej możliwe liczebności encji w związku, np. 1:∞, 0:∞, 2:10
 - albo podaje się tylko maksymalne ograniczenie
- Związki mogą posiadać swoje atrybuty
 - np. student nie tylko uczęszcza na wykład, ale i zalicza na ocenę

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

34

Diagram ER (notacja ISO)

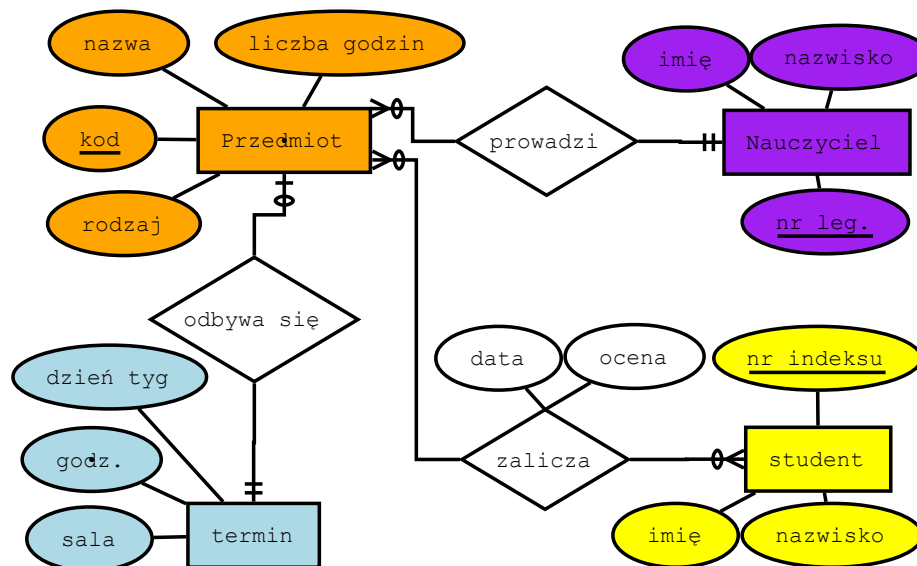


© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

35

Diagram ER (notacja Martina)

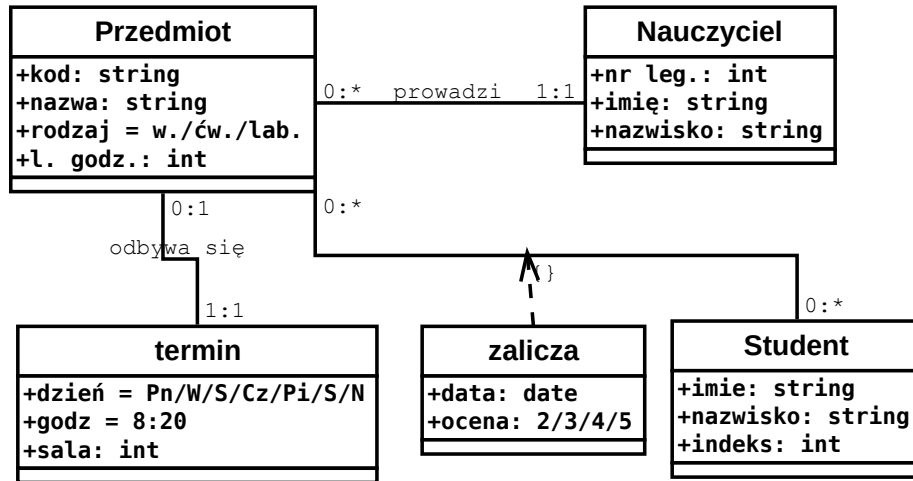


© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

36

Diagram ER w notacji UML



© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych