

# Programowanie obiektowe

Laboratorium 9 - krótkie przypomnienie diagramu związków  
encji i praktyczne porady do tworzenia relacyjnych baz  
danych

mgr inż. Krzysztof Szwarc

[krzysztof@szwarc.net.pl](mailto:krzysztof@szwarc.net.pl)

Sosnowiec, 10 maja 2017

# Czym jest diagram związków encji?

## Diagram związków encji

**Diagram związków encji**, nazywany również diagramem ERD (od ang. Entity-Relationship Diagram), umożliwia graficzne przedstawienie związków między poszczególnymi encjami. Składa się on z trzech elementów: encji, ich atrybutów oraz związków.

# Elementy ERD

## Encja

**Encja** to reprezentacja zbioru obiektów, które zostały opisane takimi samymi cechami (atributami).

## Atrybuty encji

**Atrybut encji** to własność encji określonego typu, reprezentowana pewną wartością (np. liczbą rzeczywistą).

## Związek

**Związek** to uporządkowana lista encji ( $Z(E_1, \dots, E_n)$ , gdzie  $E_1, \dots, E_n$  to encje, a  $Z$  to związek) charakteryzująca się trzema cechami: stopniem (unarny, binarny, ternarny oraz  $n$ -arny), istnieniem (opcjonalny, obowiązkowy) oraz kardynalnością (1:1, 1:N, N:M).

# Szczegółowe pojęcia

## Klucz kandydujący

**Klucz kandydujący** jest najmniejszym zbiorem atrybutów jednoznacznie identyfikującym każde wystąpienie encji w ich zbiorze. Encja może zawierać więcej niż jeden klucz kandydujący.

## Klucz główny

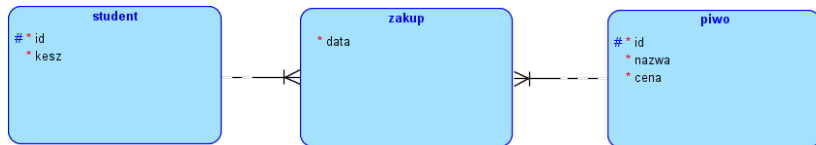
**Klucz główny** jest kluczem kandydującym, wytypowanym do jednoznacznej identyfikacji każdego z wystąpień encji w ich zbiorze. Encja może zawierać tylko jeden klucz główny.

## Normalizacja

**Normalizacja bazy danych** polega na wyeliminowaniu redundandnych danych w relacyjnej bazie danych.

## Słabe encje

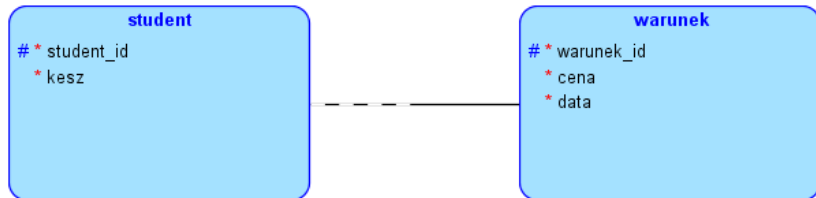
**Słaba encja** to encja, której istnienie zależy od innych encji. Nie posiada ona swojego identyfikatora. Dla podanego przykładu zakup nie istnieje bez konkretnego piwa i studenta (w praktyce podana struktura nie przyniesie oczekiwanego rezultatu - nie ma możliwości kupienia wielu piw podczas jednego zakupu. Jak możemy wyeliminować to niedopatrzenie?).



# Kardynalność 1:1

## Interpretacja przykładu

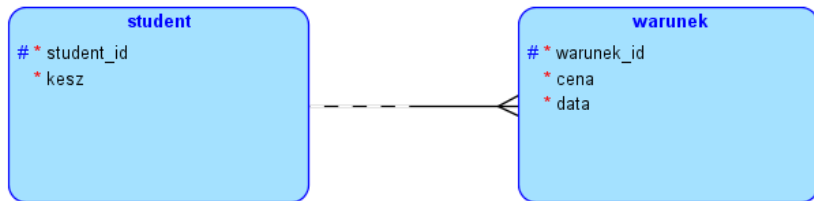
Każdy student może mieć maksymalnie jeden warunek, lecz warunek nie ma sensu bez przypisania go do konkretnego studenta (brak ekonomicznej zasadności).



# Kardynalność 1:N

## Interpretacja przykładu

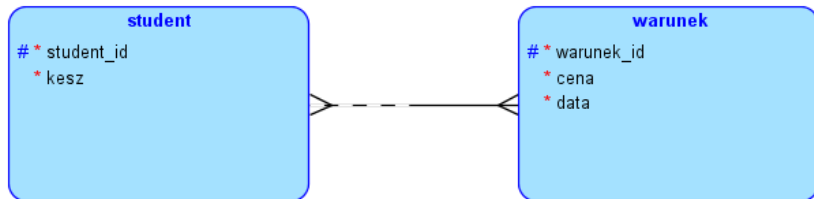
W praktyce oczywiście zdarzają się sytuacje wyjątkowe (PTC, SWI, SE, PO,...), które wymuszają wzięcie kilku warunków przez jednego studenta.



# Kardynalność N:M

## Interpretacja przykładu

W idealnych warunkach moglibyśmy przypisać jeden warunek do kilku studentów, żeby najlepszy z nich zdał przedmiot za innych, jednocześnie dopuszczając możliwość wzięcia kilku dodatkowych szans zaliczenia. W praktycznej realizacji, dla kardynalności N:M, tworzy się dodatkową encję pośrednią (asocjacyjną) i korzysta z dwóch związków 1:N.

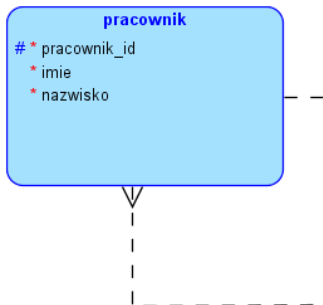




# Związek rekurencyjny

## Związek rekurencyjny

**Związek rekurencyjny** jest związkiem, w którym ta sama encja występuje więcej niż raz w różnych rolach. W przykładzie pracownik może zawierać identyfikator innego pracownika, który jest jego przełożonym (*Zawsze dwóch ich jest, nie więcej, nie mniej. Mistrz i jego uczeń*).



# Wybrane notacje zapisu ERD

Notation	Information Engineering	Barker Notation	IDEF1X	UML
<u>Multiplicities:</u>				
- Zero or one				
- One only				
- Zero or more				
- One or more				
- Specific range	N/A	N/A	N/A	

<http://www.agiledata.org>

## Uwaga

Zamieszczone porady są subiektywne i należy traktować je z rozsądkiem - są one jedynie zbiorem obserwacji, które sprawdzają się dla wielu przypadków, jednakże należy zawsze rozważyć alternatywne rozwiązania (analogicznie do *No Free Lunch* i metaheurystyk nie są one panaceum na całe zło), wybierając najlepsze z dostępnych dla danej sytuacji.

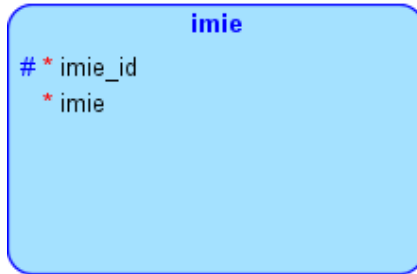
„Wyższa niż trzecia postać normalna to science-fiction.”

— Łukasz Ś., handlowiec i wdrożeniowiec

# Czemu nie należy przesadzać z normalizacją?

- 1 Długi czas złączeń (i w rezultacie zapytań).
- 2 Trudności w napisaniu zapytania.

# Encja słownikowa imion - przesadzona normalizacja?



# Klucz główny jako surogat zamiast naturalnego, tekstowego o większym rozmiarze

- ❶ Wydajniejsza część zapytań (obejmujących złączenia).
- ❷ Mniejsza zajętość pamięci (dla wielu wystąpień FK).
- ❸ Brak problemów z białymi znakami.

# Benchmark - klucz główny numeryczny

```
create table benchmark (  
numeryczny int not null primary key,  
tekstowy varchar(10) not null)  
go  
declare @i int  
set @i = 0  
while @i < 30000  
begin  
set @i = @i + 1  
insert into benchmark values(@i, @i)  
end
```



# Zapytanie - klucz główny numeryczny

```
select count(*) from benchmark tmp1  
    inner join benchmark tmp2  
on tmp1.numeryczny = tmp2.numeryczny
```

Record Count: 1; Execution Time: 8ms

# Benchmark - klucz główny tekstowy

```
create table benchmark (  
tekstowy varchar(10) not null primary  
    key,  
numeryczny int not null)  
go  
declare @i int  
set @i = 0  
while @i < 30000  
begin  
set @i = @i + 1  
insert into benchmark values(@i, @i)  
end
```

# Zapytanie - klucz główny tekstowy

```
select count(*) from benchmark tmp1  
    inner join benchmark tmp2  
on tmp1.tekstowy = tmp2.tekstowy
```

Record Count: 1; Execution Time: 12ms


# Zakładanie kluczy obcych

- 1 Zapewniają spójność danych.
- 2 Automatycznie dokonują zmian w powiązanych tabelach lub uniemożliwiają dokonania zmian naruszających zadane ograniczenia.
- 3 Spowalniają działanie bazy.
- 4 Przed MySQL 5.5.5 domyślnym silnikiem bazy był MyISAM, który nie wspierał kluczy obcych. Aby je dodać należy zmienić silnik na InnoDB.


- ➊ Gdy indeksy nie są założone wyszukanie pojedynczej wartości prowadzi do odczytania całej tabeli.
- ➋ Indeksy warto zakładać na kolumnach, które często występują w zapytaniach jako warunek.
- ➌ Powinniśmy je zakładać na kolumnach reprezentujących klucze obce.

# Korzystajmy z odpowiedniej funkcji hashującej dla haseł

  
Automated Security Analyser for ASP.NET Websites



Password

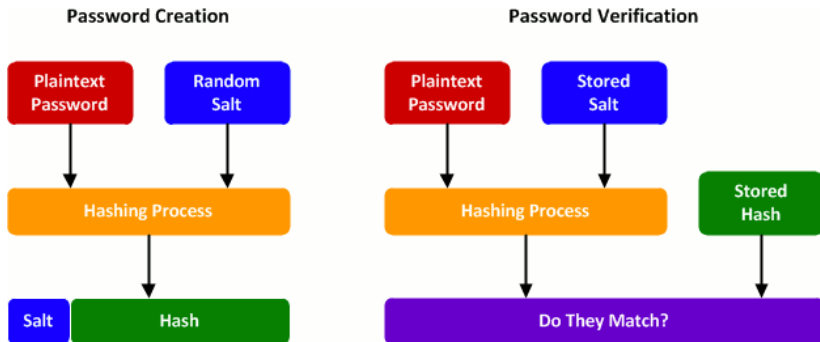
Generate 

Algorithm	Iterations/100ms	Ticks/hash	Hashes/s
MD5	6,732	148	67,568
SHA1	6,860	145	68,966
SHA256	1,329	2,671	3,744
SHA512	5,046	198	50,505
bcrypt (4 rounds)	43	92,093	109
bcrypt (5 rounds)	23	44,347	225

<https://asafaweb.com/HashSpeed>

<https://sekurak.pl/>

# Używajmy soli dynamicznej



<http://3.bp.blogspot.com/>

# Stosujemy takie same nazwy kluczy głównych i obcych

```
SELECT ... FROM student JOIN pracownik  
    USING (student_id)
```

```
SELECT ... FROM student JOIN pracownik  
    ON student.id=pracownik.student_id
```



## Widok

**Widok** (perspektywa) jest logicznym bytem znajdującym się na serwerze baz danych. Pozwala on na uzyskanie dostępu do wybranych kolumn wchodzących w skład różnych tabel, na podstawie zapytania SQL, które zostało wykorzystane do zdefiniowania obiektu (zamiast każdorazowo wysyłać skomplikowane zapytanie tworzymy widok i odwołujemy się do pożądaných danych za pomocą prostych zapytań).

# Skomplikowane zapytanie

```
SELECT
  (SELECT count(`pn2017_sze`.`Uzytkownik`.`id`)
   FROM `pn2017_sze`.`Uzytkownik`
   WHERE (`pn2017_sze`.`Uzytkownik`.`czy_surogat` = 0)) AS `uzytkownicyWWW`,

  (SELECT count(`pn2017_sze`.`Uzytkownik`.`id`)
   FROM `pn2017_sze`.`Uzytkownik`
   WHERE ((`pn2017_sze`.`Uzytkownik`.`czy_surogat` = 1)
          AND (`pn2017_sze`.`Uzytkownik`.`uprawnienia_id` <> 252))) AS `uzytkownicyTelefoniczni`,

  (SELECT `pn2017_sze`.`Sprzet_material`.`nazwa`
   FROM `pn2017_sze`.`Sprzet_material`
   WHERE (`pn2017_sze`.`Sprzet_material`.`id` =
          (SELECT `pn2017_sze`.`Oddawany_sprzet`.`sprzet_material_id`
           FROM `pn2017_sze`.`Oddawany_sprzet`
           GROUP BY `pn2017_sze`.`Oddawany_sprzet`.`sprzet_material_id`
           ORDER BY count(0) DESC
           LIMIT 1))) AS `najczesciejOddawany`,

  (SELECT round(avg(`pn2017_sze`.`Oddawany_sprzet`.`waga`),2)
   FROM `pn2017_sze`.`Oddawany_sprzet`) AS `sredniaWaga`,

  (SELECT round(avg(`pn2017_sze`.`Oddawany_sprzet`.`ilosc`),2)
   FROM `pn2017_sze`.`Oddawany_sprzet`) AS `sredniaIlosc`,
```

# Odpytanie widoku

```
SELECT 'liczbaTras' FROM 'stat_widok'
```

- Unikajmy umiejscawiania logiki biznesowej po stronie SZBD (m.in. z powodu trudności w debugowaniu).
- Mimo braku ogólnej konwencji nazewnictwa zaleca się unikanie białych i regionalnych znaków oraz - dla niektórych SZBD - korzystanie wyłącznie z małych liter.

- 1 Zaprojektujcie ERD dla zadania projektowego.

# Dziękuję za uwagę