

# BAZY DANYCH w Biologii i Medycynie

## Relacyjne Bazy Danych

Adam Piórkowski

[pioro@agh.edu.pl](mailto:pioro@agh.edu.pl)

<http://home.agh.edu.pl/~pioro/dyd/>

<http://home.agh.edu.pl/~pioro/dyd/BDwBIM/>

© AP, wszelkie prawa zastrzeżone

Kraków, 2019-

## Wprowadzenie

**Baza danych** – zbiór danych, obsługiwany przez **system zarządzania bazą danych**

**System zarządzania bazą danych (S.Z.B.D)** – oprogramowanie udostępniające następujące usługi:

- organizacji i klasyfikacji danych,
- magazynowania dużej (teoretycznie nieograniczonej) ilości danych w sposób trwały,
- przeglądania, dodawania, usuwania i modyfikacji danych,
- ochrony danych przed nieuprawnionymi odczytami/modyfikacjami,
- kontroli współbieżnego dostępu do zasobów,
- ochrony danych przed błędnymi operacjami,
- odzyskiwania danych po sytuacjach awaryjnych.

## Wprowadzenie

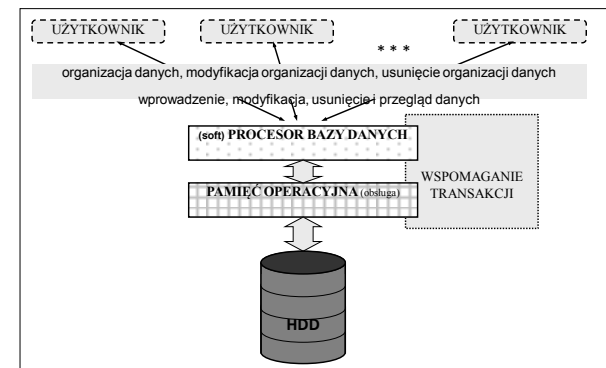
**Bank danych** – baza danych służąca do przechowywania danych.

**Repozytorium danych** – baza archiwizowanych danych.

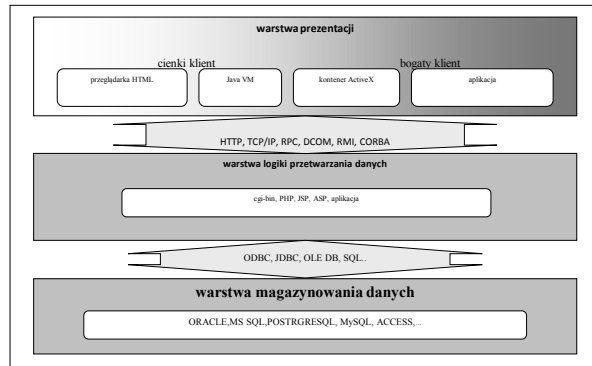
**Hurtownia danych** – baza nadrzędna, ściągająca periodycznie dane z podrzędnych baz operacyjnych, zawierająca dane zweryfikowane, opcjonalnie kumulowane i selekcyjonowane. Przeznaczona do szybkiego wyszukiwania tematycznego. Często nie jest bazą relacyjną.

**Data mining (eksploracja danych)** – proces analizy danych, mający na celu wykrycie (ukrytych przed człowiekiem) zależności między danymi.

## Składowe Systemu Zarządzania Bazą Danych



## Model trójwarstwowy



## Rodzaje baz danych – podejście ogólne

- **Bazy proste** – dane są przechowywane w tablicach, niepowiązanych ze sobą żadną więzią. Najczęściej jest to rozwiązanie dedykowane.
- **Analityczne bazy danych** – gromadzą dane (do przeglądania).
- **Operacyjne bazy danych** – gromadzą dane i pozwalają na ich modyfikację w dowolnym momencie.
- **Pasywne bazy danych** – bazy danych, w których akcje wywołuje użytkownik (HADP - Human Active Database Passive).
- **Aktywne bazy danych** – bazy danych, które mogą wyzwać akcje (DAHP - Database Active Human Passive).

## Rodzaje baz danych – organizacja danych

- **Hierarchiczne bazy danych** – koncepcja baz danych o strukturze drzewa, każdy węzeł zawiera pewien **rekord** (zestaw danych) oraz odnośniki do skończonej liczby innych rekordów, powiązanych z owym pewną hierarchią (rodzic – potomni). Powiązania między rekordami są jednokierunkowe i mają znaczenie logiczne.
- **Zalety:**
  - prosta struktura,
  - szybki dostęp do danych.
- **Wady:**
  - ograniczone możliwości realizacji relacji (1:1, 1:n)
  - utrudniona modyfikacja bazy danych.

## Rodzaje baz danych – organizacja danych

- **Relacyjne bazy danych** – najpopularniejsze obecnie bazy danych, zaproponowane w 1970 przez E.F. Codd. Model r.b.d. polega na gromadzeniu danych w tabelach, zwanych relacjami. Każda relacja składa się ze zbioru atrybutów. Wpisy w tabeli (krotki) dokonywane są w wierszu – max. po jednej wartości dla danego atrybutu. Dostępne są następujące związki między relacjami: 1:1, 1:n, n:1, m:n.

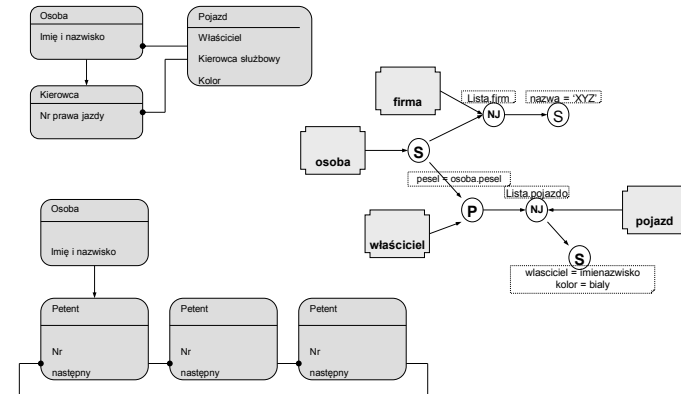
## Rodzaje baz danych – organizacja danych

- **Obiektowe bazy danych** – ten rodzaj baz danych nie jest ściśle zdefiniowany. Opiera się on o paradygmaty oprogramowania obiektowego - klasyfikuje dane jako obiekty pewnych klas, uwzględniając odpowiednie mechanizmy enkapsulacji, hermetyzacji i dziedziczenia. Zaletą o.b.d. są możliwości łatwego i szybkiego wyszukiwania danych, wadą jest uzależnienie od konkretnej platformy, na której są rozwijane (próba standaryzacji – język ODL).

## Rodzaje baz danych – organizacja danych

- **Relacyjno - obiektowe bazy danych** – hybryda obiektowych i relacyjnych baz danych – daje możliwości zamieszczania danych niestandardowych (multimedialnych, abstrakcyjnych, itp.) przy zachowaniu łatwości obsługi relacyjnej bazy danych.
- **Temporalne bazy danych** – są to bazy danych, w których dodatkowo łączy się dane z pewnymi atrybutami czasowymi. Dane mogą być aktualne dla pewnych chwil czasowych, przedziałów czasowych lub zbiorów chwil i przedziałów czasowych.

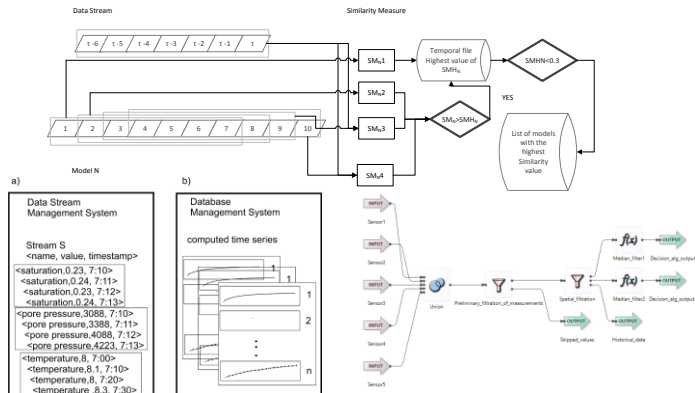
## Obiektowe bazy danych



## Rodzaje baz danych – organizacja danych

- **Strumieniowe bazy danych** – bazy aktywne, zorientowane na dane w postaci strumienia.
- Charakterystyczne cechy s.b.d.:
- użytkownik nie sprawuje nadzoru nad rejestracją danych,
  - dane mogą napływać nieustannie, przetwarzanie dotyczy głównie najnowszych danych, które mogą być archiwizowane lub odrzucone,
  - strumienie danych nie posiadają teoretycznego ograniczenia rozmiaru,
  - wbudowany moduł rejestracji danych
  - strumieniowy język zapytań (deklaratywny, obiektowy lub proceduralny), uwzględniający operacje na strumieniu danych, np. efektem zapytania może być seria odpowiedzi (teoretycznie nieskończona pętla).

## Strumieniowe bazy danych



Chudro, M., Lupa, M., Peja, A., Piotrowski, A., & Lesniak, A. (2015). A concept of time windows length selection in stream databases in the context of sensor networks monitoring. In *New Trends in Database and Information Systems II* (pp. 173-183).  
 Peja, A., Lupa, M., Chudro, M., Piotrowski, A., & Lesniak, A. (2014). A Model of a System for Stream Data Storage and Analysis Dedicated to Sensor Networks of Embankment Monitoring. In *Computer Information Systems and Industrial Management* (pp. 514-525).

## XML-owe bazy danych

```
<?xml version="1.1" encoding="ISO-8859-2"?>
<tag_glowny>
<nazwatagu>dane</nazwatagu>
<tagpustybezdanochiparametrow/>
<tagparametrami parametr="pierwszy" opcja="druga">dane</tagparametrami>
<danebinarne dt:dt="binary.base64">Fs54Dh6745GgU764s</danebinarne>
<danebinarnenasze>fs54Dh6745GgU764s</danebinarnenasze>
</tag_glowny>
```

```
<?xml version="1.0" encoding="ISO-8859-2" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="custom-date-type">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{2}/[0-9]{2}/[0-9]{4}" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="id_firmy" type="xs:int">
    <xs:annotation>
      <xs:documentation>numer firmy</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="wojewodztwo">
    <xs:annotation>
      <xs:documentation>informacje skategoryzowane</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="malopolskie"/>
        <xs:enumeration value="podkarpackie"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

## Rodzaje baz danych – organizacja danych

### • XML-owe bazy danych – ....

Są to albo samodzielne bazy danych, przechowujące dane w formacie XML, albo rozszerzenia np. relacyjnych baz danych, pozwalające na przetwarzanie danych umieszczonych w kolumnie jako tekst/blok binarny, a interpretowanych przez rozszerzenia języka odpytywania.

### • XML (eXtensible Markup Language) – to uniwersalny, tekstowy standardyzowany, sprzętowo niezależny format organizacji i zapisu danych. Podstawowym elementem przechowywania i organizacji danych jest tzw. Tag. Budowa dokumentu XML:

- nagłówek określający wersję XML i kodowanie,
- tag główny, zagnieżdżający cały dokument.

- XML Schema Definition (xsd) to sposób dołączania do dokumentu informacji o strukturze danych.
- XPath jest językiem służącym do nawigacji i operacji na danych XML.
- XQuery jest standardyzowanym językiem do operacji na dokumentach XML, zawierającym (rozszerzającym) język XPath.

## Rodzaje baz danych – organizacja danych

### • ‘Przestrzenne bazy danych’ (Bazy danych przestrzennych)

– bazy, które posiadają specyficzny system do składowania i przetwarzania danych przestrzennych.

- Główne zastosowanie – magazynowanie danych przestrzennych
- Rozszerzenie języka SQL do przetwarzania tychże danych

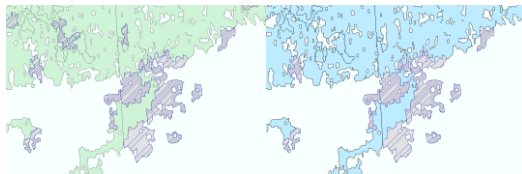
```
INSERT INTO Gminy (id_gminy, Nazwa_gminy, obszar)
VALUES (1, 'Staropolska', GeomFromText('POLYGON((6 3, 6 19,
21 19, 21 3, 6 3))));
```

```
INSERT INTO Drogi(id_drogi, przebieg)
VALUES (1, GeomFromText('LINESTRING(0 10, 15 10, 25 13)', -1));
```

```
mysql> SELECT Nazwa_gminy, Area(obszar) FROM Gminy;
+-----+-----+
| Nazwa_gminy | Area(obszar) |
+-----+-----+
| Staropolska |          240 |
+-----+-----+
1 row in set (0.00 sec)
```

```
psql=# SELECT Nazwa_gminy, Area(obszar) FROM Gminy;
 nazwa_gminy | area
+-----+-----+
 Staropolska | 240
(1 row)
```

## Bazy danych przestrzennych

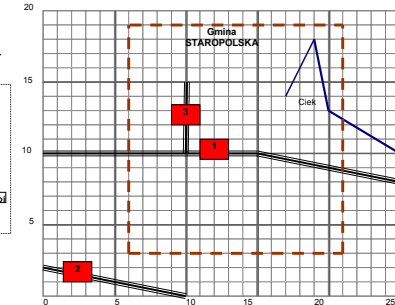
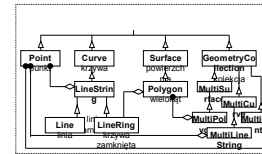


Lupa, M., & Piórkowski, A. (2014). Spatial query optimization based on transformation of constraints. In Man-Machine Interactions 3 (pp. 621-629).

## Bazy danych przestrzennych

```
-- czesc wspolna
SELECT id_drogi, Nazwa_gminy FROM Drogi, Gminy
WHERE Intersects(obszar, przebieg);
```

id_drogi	Nazwa_gminy
1	Staropolska
3	Staropolska



Piórkowski, A. (2011). MySQL spatial and postgis-implemations of spatial data standards. Electronic Journal of Polish Agricultural Universities, 14(1), 1-8.

## Systemy zarządzania bazami danych

- **SQL-owe bazy danych – bazy relacyjne**
  - Darmowe: MySQL, PostgreSQL, SQLite
  - Komercyjne: Oracle, MS SQL Server, IBM DB2
- **Bazy do niezawansowanych zastosowań biurowych**
  - MS Access – komercyjna, zawiera także SQL
  - OpenOffice Base / LibreOffice Base – darmowe, konkurencyjne
- **NoSQL** – bazy przeznaczone dla projektów o ogromnej ilości danych (ang. Big data), nastawione na wydajność, realizacja często na zasadzie <klucz, wartość>

## NOSQL-owe bazy danych

- Nowy rodzaj baz danych, zrywających ze schematem relacyjnym jako podstawowym,
- Tym samym rezygnuje się ze standardu języka SQL
- Każdy system ma własny język
- Różne modele danych – np.:
  - bazy typu key-value,
  - bazy dokumentowe (np. oparte o JSON),

## Unifikacja typów danych

### • Typy danych

- dane numeryczne (całkowite, dziesiętne, zmiennoprzecinkowe);
- operacje na danych numerycznych - konwersja: rzutowanie, zaokrąglanie, obcinanie; operacje arytmetyczne
- dane znakowe, unicode
- miary czasu (data, godzina)
- NULL
- BLOB
- dane multimedialne (obrazy, dźwięki, video, strumienie)
- dane w specjalnych formatach, np. XML

## Wybrane funkcjonalności systemów baz danych

- **Raporty** - służą do uporządkowanego prezentowania zawartości bazy danych, jej fragmentu lub wyników wykonania pewnego zapytania do bazy danych. Nie wprowadzają modyfikacji.
- **Formularze** – służą do wprowadzania danych do bazy (lub ich modyfikowania) w wyniku interakcji z użytkownikiem. Czuwają nad poprawnością powierzanych danych.
- **Kwerendy (zapytania)** - stanowią realizację zapytania użytkownika do bazy danych – mogą być wykonywane w interakcji z nim. Kwerendy pozwalają na wyświetlanie, analizę, wyszukiwanie i modyfikowanie danych. Ich efektem jest zbiór rekordów spełniających wyrażenie zapytania.
- **Kwerendy modyfikujące:**
  - usuwające – usuwanie zbioru krotek w tabeli (tabelach),
  - dołączające – dodaje zbiór krotek do tabeli (tabel),
  - aktualizujące – wykonuje operacje na wybranych atrybutach wszystkich krotek w wybranych tabelach,
  - tworzące tabele - np. tworzy kopię zapasową tabeli.
- **Kwerendy krzyżowe** – operują na co najmniej dwóch tabelach – ich efektem jest iloczyn kartezjański danych z tabel.

## Wielodostępowość

### • Dostęp do danych (użytkownicy)

- Systemy baz danych oferują różne poziomy dostępu do danych. Najczęściej spotykane role użytkowników (w relacyjnych bazach danych) to:
  - administrator – zazwyczaj posiada przywileje tworzenia, podłączania i usuwania baz danych, tworzenia i usuwania tabel oraz dowolnych operacji na danych,
  - użytkownik – zazwyczaj posiada możliwości dostępu w trybie odczytu do bazy danych, opcjonalne możliwości tworzenia i kasowania tabel (np. nadane przez administratora) oraz pełne lub ograniczone prawa manipulacjami w tabelach.
- **Współbieżność w bazach danych**
- Systemy baz danych oferują wielodostęp do danych. Pojedyncze operacje są atomowe. Sekwencje operacji mogą być zabezpieczone mechanizmami transakcji (dla różnych poziomów izolacji). Dostępne są blokady.

## Wybrane funkcjonalności systemów baz danych

- **Triggery (wyzwalacze)** – pozwalają na powiadomienie użytkownika (aplikacji) / wykonaniu akcji w wyniku zaistniałego zdarzenia w bazie danych.
- **Procedury składowane / wbudowane (stored procedures)** – procedury przechowywane w serwerze i wykonywane przez serwer, w języku SQL albo w języku wewnętrznym danego serwera (wówczas mogą być prekompilowane do postaci bytecodeu)
- **Archiwizacja (kopie zapasowe)** – umożliwia stworzenie obrazu aktualnego stanu danych, potrzebnych do odtworzenia lub przeniesienia.

## RELACYJNE BAZY DANYCH

E.F. Codd, 1970r

-relacyjne struktury danych  
-operatory algebry relacyjnej  
-ograniczenia integralności danych (więzy)

- **Zapis formalny modelu relacyjnego**
- **dziedzina** = atrybut = zbiór elementów danego rodzaju (liczby całkowite albo liczby zmiennoprzecinkowe albo tekst albo ....)
- Przyjmijmy zbiór atrybutów  $\{D_1, D_2, \dots, D_N\}$
- Iloczyn kartezjański atrybutów:  $D_1 \times D_2 \times \dots \times D_N$
- **relacja** – dowolny podzbiór iloczynu kartezjańskiego utworzonego z listy dziedzin.

## RELACYJNE BAZY DANYCH

Iloczyn kartezjański - przykład:

$D_{\text{imie}} = \{\text{Adam, Ewa}\}, D_{\text{nazwisko}} = \{\text{Nowak, Kowal}\}$   
 $D_{\text{imie}} \times D_{\text{nazwisko}} = \{(\text{Adam, Nowak}), (\text{Adam, Kowal}),$   
 $(\text{Ewa, Nowak}), (\text{Ewa, Kowal})\}$

Relacje - przykład:

$R_{\text{kobiety}} = \{(\text{Ewa, Nowak}), (\text{Ewa, Kowal})\}$   
 $R_{\text{RodzinaNowakow}} = \{(\text{Ewa, Nowak}), (\text{Adam, Nowak}), \}$

## RELACYJNE BAZY DANYCH

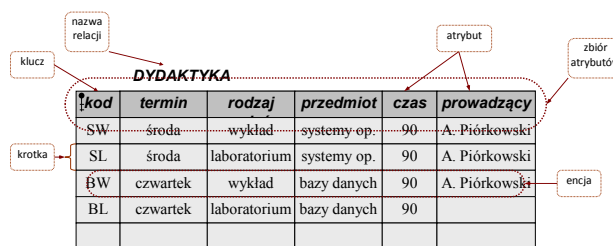
Iloczyn kartezjański - przykład:

$D_1 = \{a, b\}, D_2 = \{c, d\}$   
 $D_1 \times D_2 = \{(a,c), (b,c), (a, d), (b,d)\}$

Relacje - przykład:

$D_1 = \{a, b\}, D_2 = \{c, d\}$   
 $R_1 = \{(a,c), (b,c), (b,d)\}$   
 $R_2 = \{(a,c)\}$   
 $R_3 = \{(a,d), (b,c)\}$   
 $R_4 = \{(a,c), (b,c), (a, d), (b,d)\}$   
 ...

## Koncepcja relacyjnej bazy danych

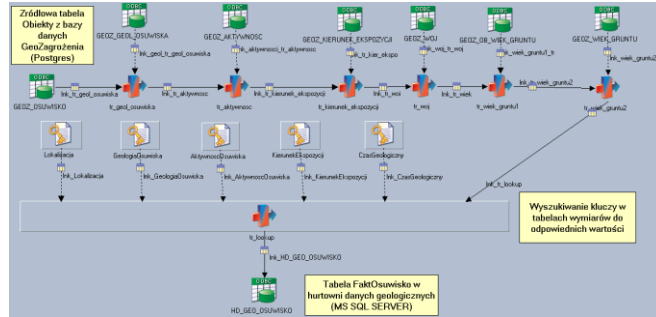






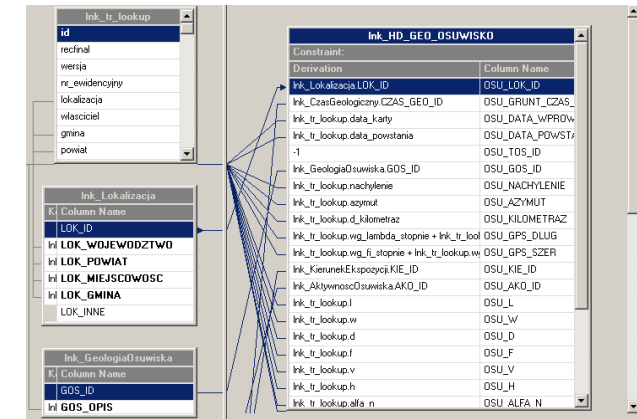


## PRZYKŁADY



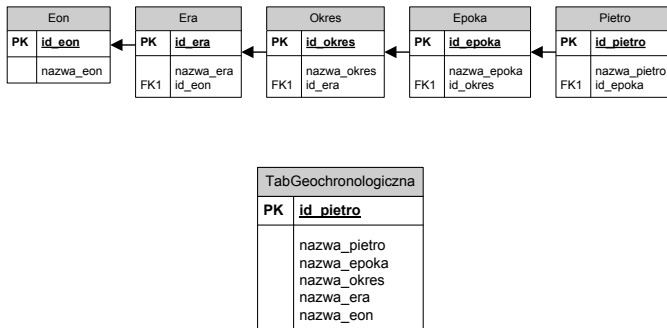
Pińkowski A., Gajda G.: Konstrukcja Wielowymiarowej Bazy Danych Geologicznych. Studia Informatica 2009, Vol. 30 No 26 (84)

## PRZYKŁADY



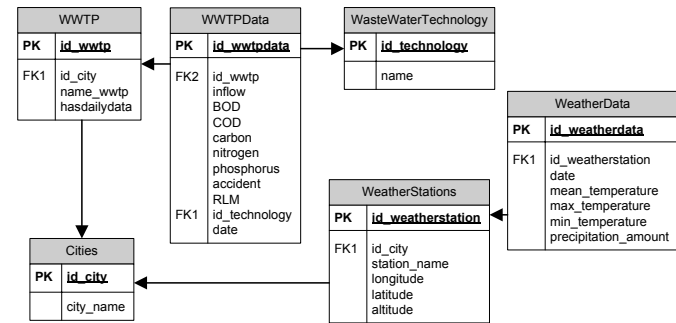
Pińkowski A., Gajda G.: Konstrukcja Wielowymiarowej Bazy Danych Geologicznych. Studia Informatica 2009, Vol. 30 No 26 (84)

## PRZYKŁADY



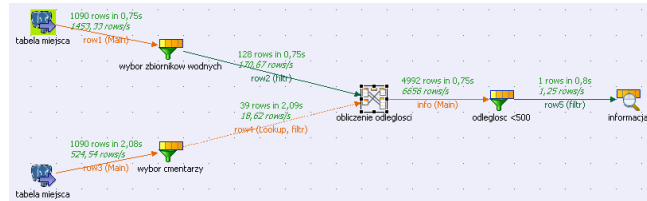
Jajęcina L., Pińkowski A.: Wydajność Złączeń i Zagęszczeń Dla Schematów Znormalizowanych i Zdenormalizowanych. Studia Informatica 2010, Vol. 31

## PRZYKŁADY

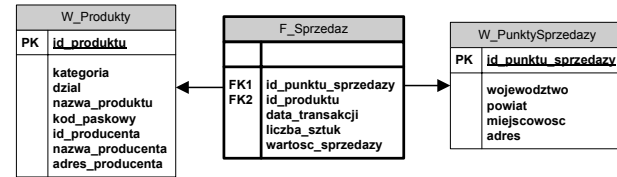


M Chudro, Pińkowski A.: Wykorzystanie Metod i Narzędzi Eksploracji Danych Do Analizy Zmniejszenia Natężenia Dopyływu Do Komunalnych Oczyszczalni Ścieków. Studia Informatica 2010, Vol. 31

## PRZYKŁADY

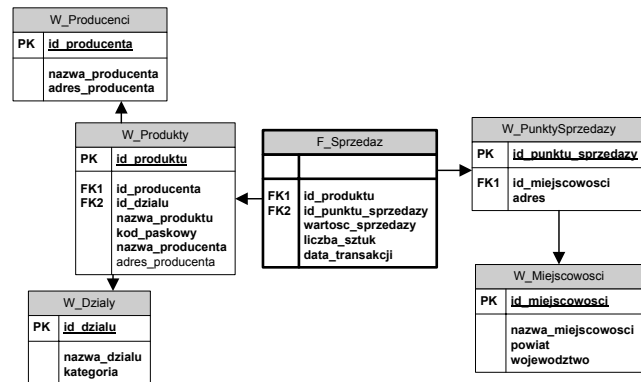


## PRZYKŁADY

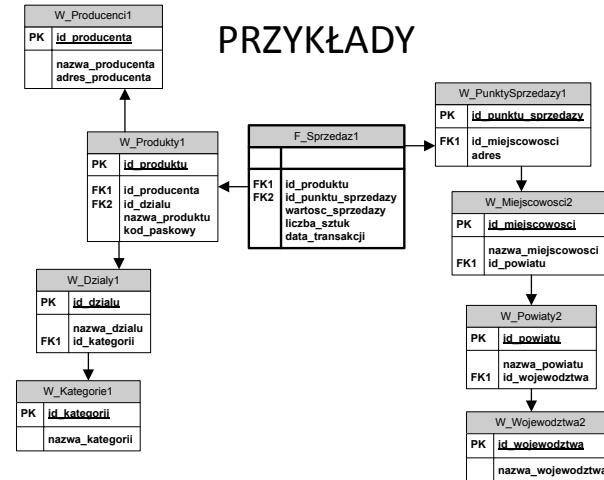


Kisilewicz, J., Piórkowski, A., Porzycka, S. (2011). Konstrukcja procesu ETL dla danych przestrzynnych. Studia Informatica, 32(2B), 131-142.

## PRZYKŁADY

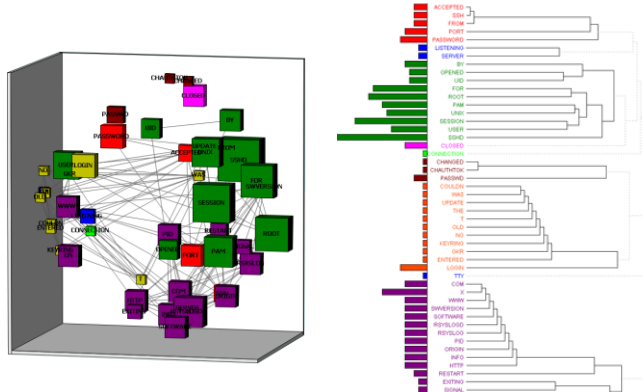


## PRZYKŁADY





## PRZYKŁADY



Chuzho, M., Szeszek, K., Piórkowski, A., & Darzek, T. (2013). Using Data Mining Techniques for Diagnostic of Virtual Systems Under Control of KVM. In *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering* (pp. 1011-1022). Springer New York.

## PROJEKTOWANIE (relacyjnych) BAZ DANYCH

Reprezentacja relacji (tabeli)

Więzy relacji (relacje)

Diagramy ERD

\* notacja Chena

\* notacja Martina

Normalizacja schematów

\* anomalie w r.b.d.

\* postaci normalne

### Relacja (tabela)

Schemat relacji = nazwa relacji + zbiór atrybutów

**Studenci**(*imie, nazwisko, pesel, nr\_indeksu*)

**Akademiki**(*nr\_ds, uczelnia, miasto, ulica, numer*)

**SaleWykladowe**(*budynek, nr\_Sali, pietro, liczba\_miejsc, rzutnik*)

### Projektowanie tabeli

- Nazwa tabeli – zbiór - liczba mnoga?
- Klucz główny
- Zbiór atrybutów – typy danych

## Typy danych

TYP	[B]	ZAKRES	OPIS
INTEGER, INT,	4	$-2^{31}$ do $2^{31}-1$	Liczby całkowite
BIGINT, SMALLINT, TINYINT	8,2,1	-2,147,483,648 - 2,147,483,647	
DECIMAL (precyzja, skala) NUMERIC*	~ precy zji		Liczby dziesiętne, np. waluta: 100.15
FLOAT, REAL	Precy zja 4	-1.79E+308 : -2.23E-308, 2.23E-308 to 1.79E+308 - 3.40E + 38 to -1.18E – 38 1.18E - 38 to 3.40E + 38.	Liczby zmiennoprzecinkowe

NUMERIC ma stałą precyzję, DECIMAL – może mieć większą, niż potrzebuje projektant (związane z wydajnością szbd, w części szbd te typy są równoważne)

Standard SQL

## Typy danych

TYP	[B]	ZAKRES	OPIS
CHAR CHAR(20)	1 20	Jeden znak ASCII Tablica 20 znaków ASCII	Jeden znak Dokładnie 20 znaków
VARCHAR(_X_)	1B - 2GB	Tablica znaków ASCII	Dowolna liczba znaków, ograniczona przez _X_
NVARCHAR(_X_)	2B - 2GB	Tablica znaków UNICODE	Można składować tekst w znakach narodowych
DATE, TIME, DATETIME	10B,..	Np. tekstowo YYYY-MM-DD	czas
BLOB	Np. do 2GB	Blok binarny	Np. na zdjęcia

Standard SQL

## Typy danych

całkowitoliczbowe

ze znakiem

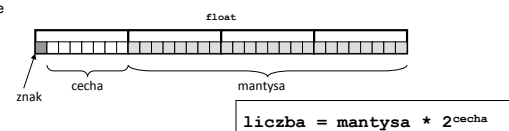


long int

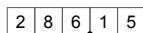


zmiennoprzecinkowe

ze znakiem



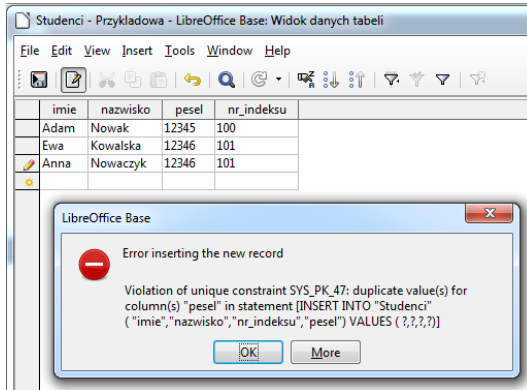
dziesiętne  
(stała precyzja)



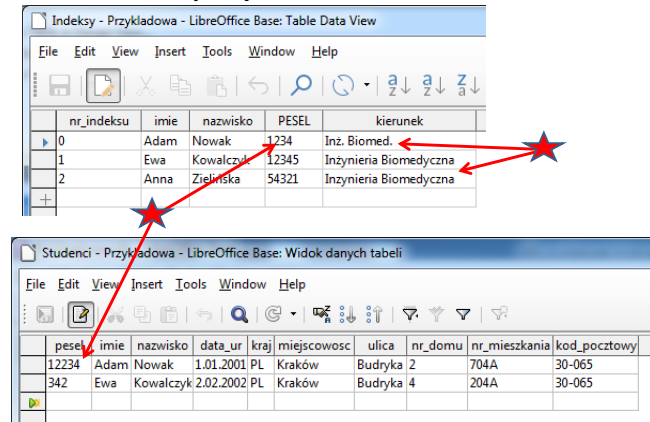
Indeksy(nr\_indeksu, imie, nazwisko, pesel, kierunek)

## Projektowanie tabeli

## Wpisywanie wartości



## Wpisywanie wartości



## ZWIĄZKI ENCJI, WIĘZY RELACJI (relacje;)

- Wymuszenie integralności danych między dwoma tabelami.

Między dwoma zbiorami encji A i B mogą istnieć powiązania. Przyjmijmy, że zbiór A jest zbiorem źródłowym a zbiór B jest zbiorem docelowym.

- Powiązania są **jednokierunkowe** (ale mogą istnieć odwrotne).
- Odwzorowanie jest **całkowite**, jeśli każda encja zbioru źródłowego ma swój obraz.
- Odwzorowanie jest **częściowe**, jeśli istnieją z zbiorze źródłowym encje bez obrazu.
- Liczebność związków: 1:1, 1:N oraz M:N (!)

## 1:1 – jeden do jednego

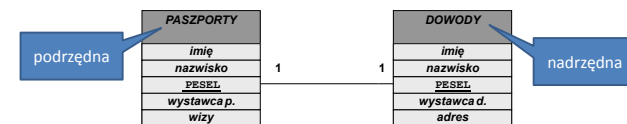
W tabeli podrzędnej, w kolumnie klucza obcego (FK) mogą być tylko wartości [i nie mogą się powtarzać] ze wskazanej kolumny (np. klucza głównego) tabeli nadrzędnej [tu też nie mogą się powtarzać].

przykład:

Paszporty( imię, nazwisko, PESEL, osoba wystawiająca paszport, wizy)  
Dowody\_osobiste( imię, nazwisko, PESEL, osoba\_wyst\_dowod, adres\_zamieszkania).

Założenia:

- każdy musi mieć dowód osobisty, ale tylko wybrani mają paszport (nie odpowiada to sytuacji np. studentów obcokrajowców)
- można mieć tylko jeden dowód osobisty i jeden paszport,
- wniosek: paszport można wyrobić tylko na **istniejący** dowód osobisty



## 1:1 – jeden do jednego

LibreOffice: przeciągnięcie nadrzędnego na podrzędny

LibreOffice: projekt relacji

Field Name	Field Type
pesel	BigInt [ BIGINT ]
imie	Text [ VARCHAR ]
nazwisko	Text [ VARCHAR ]
nr_dowodu	Text [ VARCHAR ]

Field Name	Field Type
pesel	BigInt [ BIGINT ]
imie	Text [ VARCHAR ]
nazwisko	Text [ VARCHAR ]
nr_paszportu	Text [ VARCHAR ]

## 1:1 – jeden do jednego

LibreOffice Base

Error inserting the new record

Integrity constraint violation - no parent SYS\_FK\_78 table: Dowody in statement [INSERT INTO "Paszporty" ("imie","nazwisko","nr\_paszportu","pesel") VALUES (?, ?, ?, ?)]

OK More

## 1:N – jeden do wielu

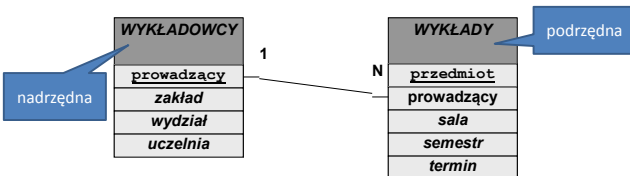
W tabeli podrzędnej, w kolumnie klucza obcego (FK – Foreign Key) mogą być tylko wartości [ale mogą się powtarzać - N] ze wskazanej kolumny (np. klucza głównego PK – Primary Key, wart. unikalne) tabeli nadrzędnej [tu nie mogą się powtarzać].

przykład:

wykładowcy: rowadzący, zakład, wydział, uczelnia  
 wykłady: przedmiot, prowadzący, sala, semestr, termin

Założenia:

- każdy wykład musi być poprowadzony przez jednego, istniejącego wykładowcę
- każdy wykładowca może prowadzić 0 – N wykładów



## 1:N – jeden do wielu

Field Name	Field Type
prowadzący	Text [ VARCHAR ]
zakład	Text [ VARCHAR ]
wydział	Text [ VARCHAR ]
uczelnia	Text [ VARCHAR ]

Field Name	Field Type
wykład	Text [ VARCHAR ]
wykładowca	Text [ VARCHAR ]
sala	Text [ VARCHAR ]
semestr	Text [ VARCHAR ]
rok	Text [ VARCHAR ]

Nazwy atrybutów, o które związek się opiera, mogą, ale nie muszą być identyczne



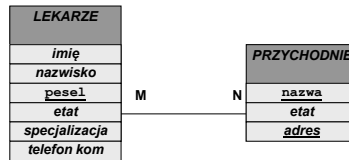
## 1:N – jeden do wielu

## M:N – wiele do wielu

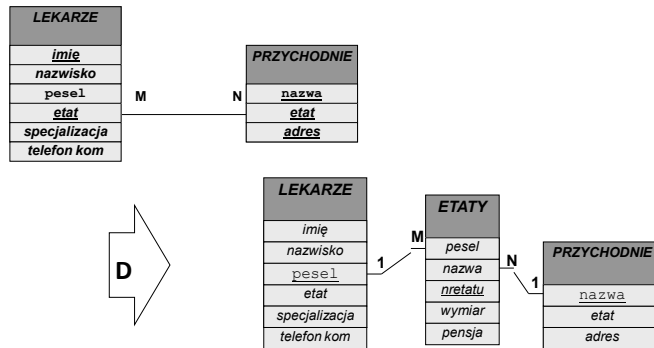
W obrębie klucza danej wartości w jednej tabeli odpowiada podzbiór wartości w drugiej tabeli – i odwrotnie.  
 Takie zależności nie są możliwe do bezpośredniej implementacji w r.b.d i wymagana jest dekompozycja M:N do dwóch związków M:1 i 1:N

**przykład:**  
 lekarze: imię, nazwisko, specjalizacja, etat  
 przychodnie: nazwa, adres, etat

Założenie: każdy lekarz może pracować w różnych przychodniach, przychodnie mogą zatrudniać różnych lekarzy.



## M:N – wiele do wielu dekompozycja M:N do 1:M i N:1



## Więzy integralności danych

## Więzy integralności danych

nr_indeksu	imie	nazwisko	PESEL	kierunek
0	Adam	Nowak	1234	Inż. Biomed.
1	Ewa	Kowalczyk	12345	Inżynieria Biomedyczna
2	Anna	Zielińska	54321	Inżynieria Biomedyczna



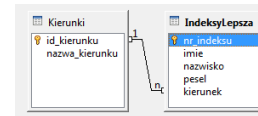
Ż vs Z

Field Name	Field Type
id_kierunku	Integer [ INTEGER ]
nazwa_kierunku	Text [ VARCHAR ]

Field Name	Field Type
id_kierunku	Integer [ INTEGER ]
nazwa_kierunku	Text [ VARCHAR ]

Field Name	Field Type
nr_indeksu	Integer [ INTEGER ]
imie	Text [ VARCHAR ]
nazwisko	Text [ VARCHAR ]
pesel	Text [ VARCHAR ]
kierunek	Integer [ INTEGER ]

## Więzy integralności danych



nr_indeksu	imie	nazwisko	pesel	kierunek
0	Adam	Nowak	4324	2
1	Ewa	Nowaczyk	4343	1
<AutoField>	Anna	Xyz	322	3
<AutoField>				

LibreOffice Base

**Error inserting the new record**

Integrity constraint violation - no parent SYS\_FK\_64 table: Kierunki in statement [INSERT INTO "IndeksyLepsza" ("imie", "kierunek", "nazwisko", "pesel") VALUES (?, ?, ?, ?)]

OK More

## Klucze

- Klucz prosty – klucz utworzony z jednego atrybutu
- Klucz złożony – klucz utworzony z dwóch lub więcej atrybutów
- Klucz minimalny (kandydujący) – minimalny zestaw atrybutów, tworzący klucz.
- Nadklucz – zbiór atrybutów, zawierający klucz, a niekoniecznie będący kluczem minimalnym.
- Podklucz – podzbiór właściwy atrybutów klucza.
- Klucz podstawowy (główny) – jeden ze zbioru dostępnych kluczy minimalnych, wybrany przez projektanta bazy.
- Klucz obcy – zbiór atrybutów pewnej relacji, stanowiący klucz podstawowy dla innej relacji.

PK – PRIMARY KEY

FK – FOREIGN KEY

## Klucze – klucz prosty

Field Name	Field Type
nr_indeksu	Integer [ INTEGER ]
imie	Text [ VARCHAR ]
nazwisko	Text [ VARCHAR ]
PESEL	Text [ VARCHAR ]
kierunek	Text [ VARCHAR ]

nr_indeksu	imie	nazwisko	PESEL	kierunek
0	Adam	Nowak	1234	Inż. Biomed.
1	Ewa	Kowalczyk	12345	Inżynieria Biomedyczna
2	Anna	Zielińska	54321	Inżynieria Biomedyczna
2	Olga	Zajac	2222	IB

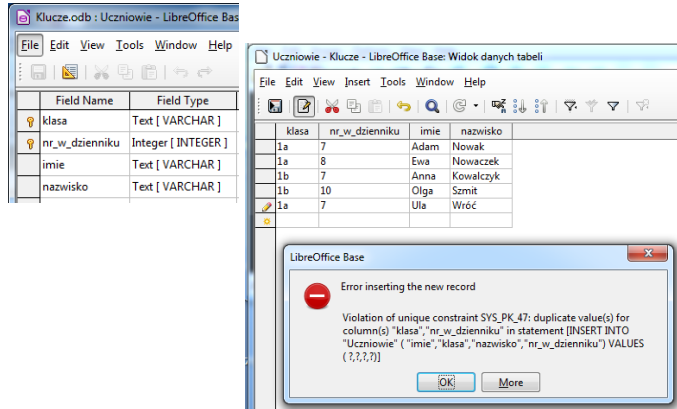
LibreOffice Base

**Error inserting the new record**

firebird\_dbc: error:  
 "validation of PRIMARY or UNIQUE KEY constraint "INTEG\_6" on table "Indeksy"  
 "Problematic key value is ("nr\_indeksu" = 2)  
 caused by  
 'isc\_dsql\_execute'"

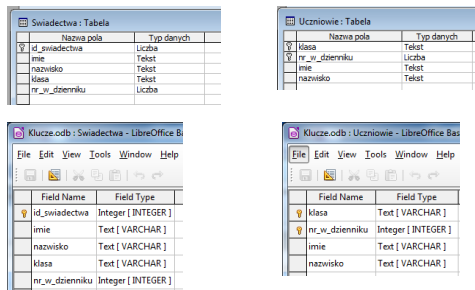
OK More

## Klucze – klucz złożony



## Klucze obce

- LibreOffice nie chciał mi tutaj służyć w przypadku kluczy obcych złożonych
- W MS Access takie klucze są
- Podobieństwa i różnice LibreOffice i MS Access

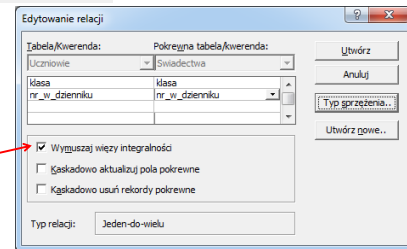
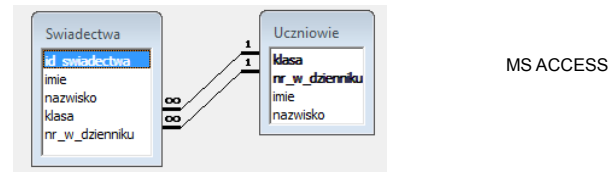


## Klucze

Jaki atrybut wybrać na klucz główny:

- Wartość unikana, np. nr PESEL:
  - jeśli to tabela opisująca osoby, to PESEL może być kluczem,
  - PESEL może nie być unikalny, np. w księdze odwiedzin pesel odwiedzających może się powtarzać,
- atrybut typu integer – jest szybciej porównywany przez s.z.b.d. niż np. łańcuch tekstowy (string), także zapis peselu w postaci stringu może zawierać spacje, których nie widać, a rozróżniają dwie wartości,
- w przypadku wartości słownikowych (a nawet ogólnie) każda tabela może mieć klucz główny jako dodatkowy atrybut (najlepiej liczba), stworzony od nazwy tej relacji, np.:
  - Powiaty(id\_powiatu, nazwa\_powiatu)
  - Studenci(id\_studenta, PESEL, nr\_indeksu, imie, nazwisko)
- klucze w opisie należy wyróżnić, najlepiej czcionką podkreśloną, jeśli są znaki ' ' , to dla czytelności może to być czcionka pogrubiona.

## Klucze obce złożone



## Tabele a rodzaje danych

### Model logiczny dla hurtowni danych

#### Wymiary

- stabelaryzowane dane określające atrybuty danego pojęcia,
- tabele wymiarów posiadają klucze główne – przy pomocy klucza identyfikowany jest zestaw atrybutów,
- wymiary mogą tworzyć hierarchię, np. województwo → powiat → gmina → miejscowość → dzielnica,
- dane zgromadzone w tabeli wymiaru zazwyczaj nie zmieniają się (częściej: dochodzą nowe),
- danych w tabeli wymiaru nie musi być wiele.

#### Fakty

- gromadzone dane, opisujące wybrane zjawisko,
- zazwyczaj duże (ogromne) tabele,
- posiadają klucze obce do tabel wymiarów, które to wymiary zawierają zestawy atrybutów opisujących fakty.

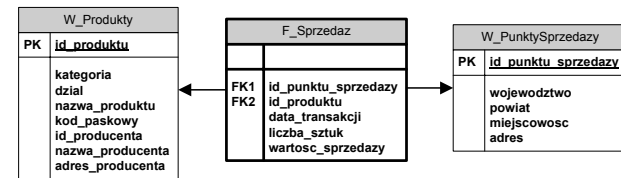
#### Miary

- wartości numeryczne opisujące fakt, czyli np. cena, waga, długość.

## Model schematów danych

### Gwiazda

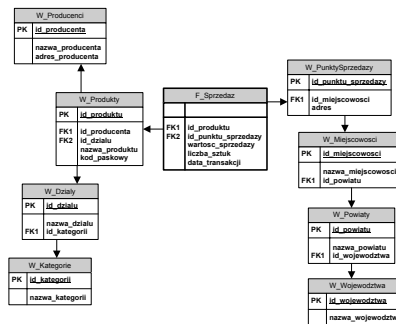
- jedna tabela faktów w postaci znormalizowanej,
- wiele tabel wymiarów w postaci znormalizowanej lub zdenormalizowanej – każdemu wymiarowi odpowiada tylko jedna tabela (nie ma hierarchii),
- prosta struktura,
- niewielka liczba połączeń (→ złączeń),
- klucz tabeli faktów obejmuje wszystkie kolumny oprócz miar.



## Model schematów danych

### Płatek śniegu

- jedna tabela faktów (podobnie jak w schemacie gwiazdy),
- tabele wymiarów znormalizowane (tworzą hierarchię – płatek śniegu),
- schemat łatwiejszy do modyfikacji
- większa liczba połączeń – większa liczba złączeń – spadek efektywności zapytań ?\*

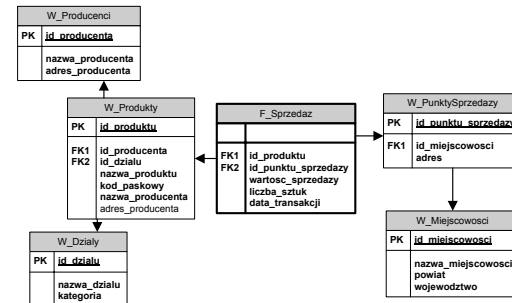


\* Jajędnica L., Piórkowski A.: Wydajność złączeń i zagęszczeń dla schematów znormalizowanych i zdenormalizowanych. Studia Informatica. 2010. vol. 31 no. 2A s. 445-456.

## Model schematów danych

### Gwiazda-płatek

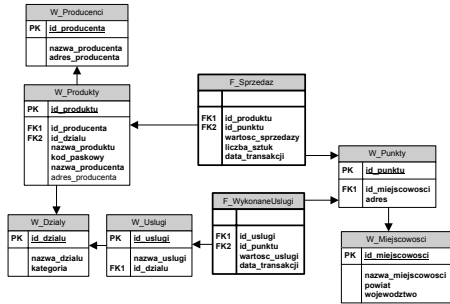
- schemat posiadający część wymiarów znormalizowanych, a część nie – optymalizacja wydajności, wybrane tabele wymiarów nie będą modyfikowane ani rozwijane.



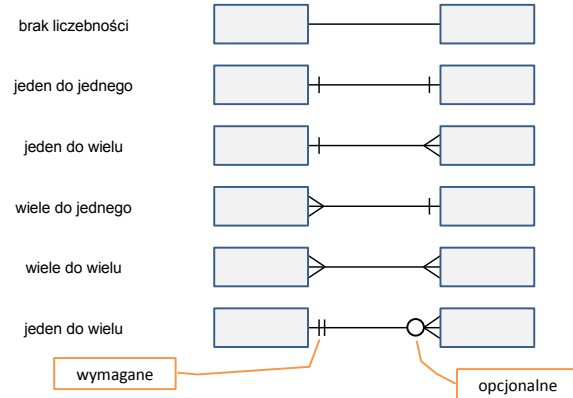
# Model schematów danych

## Konstelacja

- występują tu co najmniej dwie tabele faktów, mogą być one połączone,
- tabele faktów mogą współdzielić tabele wymiarów,
- rozwiązanie elastyczne, trudniejsza implementacja, trudniejszy rozwój.



# Notacja Martina



# Diagramy związków encji (ERD)

notacja Chena

Zbiory encji – prostokąty  
 Atrybuty – owale  
 Związki encji - romby



Nazwy zbiorów encji – liczba mnoga  
 Połączenie *atrybut* – *zbiór encji* jest niezorientowane  
 Połączenie *zbiór encji* – *związek* - *zbiór encji* może być lub nie być skierowane.  
 Strzałka wskazuje encję pojedynczą.

# Diagramy związków encji (ERD)

notacja Chena

## Związek jeden-do-jednego

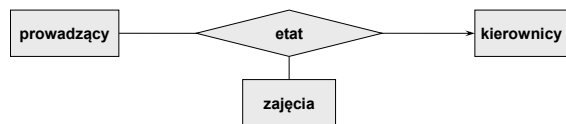


## Związek wiele-do-jednego

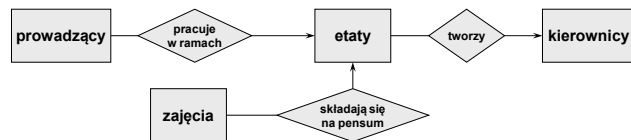


## Diagramy związków encji (ERD) notacja Chena

### Związki wieloargumentowe

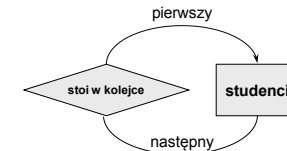


### Dekompozycja związku wieloargumentowego w związki dwuargumentowe

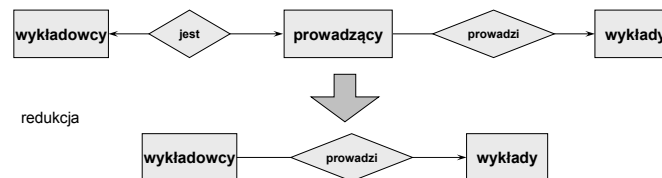


## Diagramy związków encji (ERD) notacja Chena

### Wielokrotne wykorzystanie zbioru encji



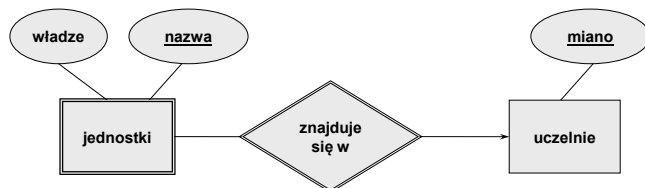
### Redundancja



## ZBIORY ENCJI SŁABYCH

Encja słaba:  
może istnieć tylko powiązana z innymi encjami lub nie posiada własnego klucza.

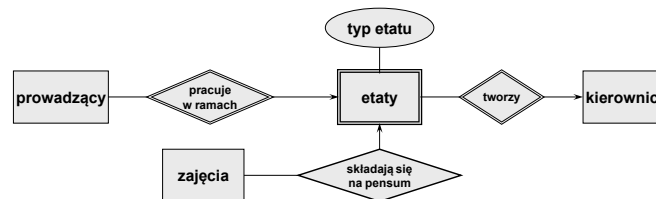
**Przypadek 1** – zbiór encji A jest podzbiorem encji zbioru B  
jednostka: władze (np. {dziekan, kierownik, ...}), nazwa (np. {biblioteka, wydział informatyki,...})  
uczelnia: miano (np. {AGH, UE, UP, UJ,...})



? klucz {np. funkcja: kierownik biblioteki - w UP, w UE}

## ZBIORY ENCJI SŁABYCH

### Przypadek 2 – brak własnego klucza



Jan Nowak pracuje 210h w ramach etatu **adiunkta** stworzonego przez kierownika jednostki Marcina Kowalskiego  
Jan Nowak pracuje 210h w ramach etatu **adiunkta** stworzonego przez kierownika (innej) jednostki Jerzego Wójcika  
 Justyna Krawczyk pracuje 60h w ramach etatu **adiunkta** stworzonego przez kierownika jednostki Jerzego Wójcika.  
 Alicja Kowalska pracuje 120h w ramach etatu **asystenta** stworzonego przez kierownika jednostki Jerzego Wójcika.

## Przekształcanie diagramów związków encji do postaci relacyjnego modelu danych

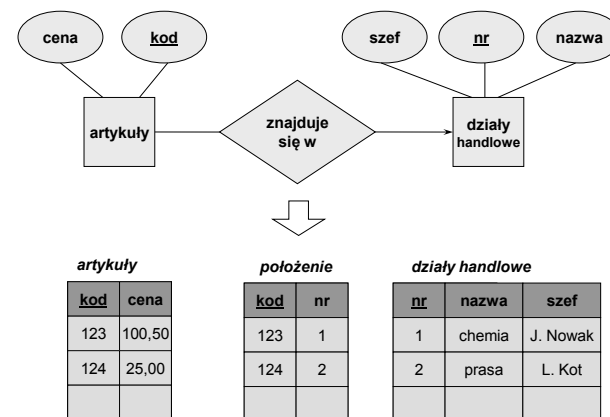
Dla każdego zbioru encji tworzy się tabelę o nazwie odpowiadającej nazwie zbioru encji i takim samym zbiorze atrybutów.

Dla każdego związku encji tworzy się tabelę o zbiorze atrybutów, w którym klucze powiązanych zbiorów encji i ewentualne klucze związku.

W zależności od liczebności związku:

- 1:1 – nowa tabela przejmuje klucze po jednym ze zbiorów encji,
- N:1 – nowa tabela przyjmuje klucze po zbiorze encji w związku po stronie N,
- M:N – nowa tabela przyjmuje klucze obydwu zbiorów encji.

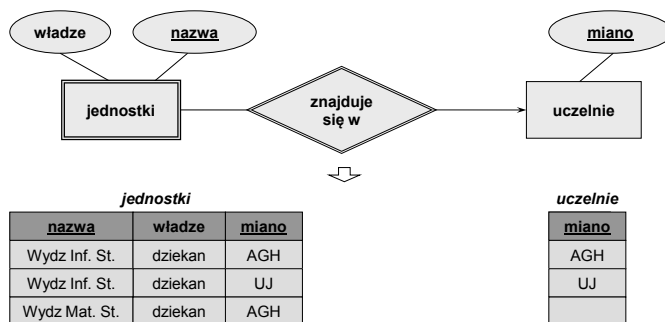
## Przekształcanie diagramów związków encji do postaci relacyjnego modelu danych



## Przekształcanie diagramów związków encji do postaci relacyjnego modelu danych

Postępowanie w przypadku związków słabych:

- relacja odpowiadająca słabemu zbiorowi encji musi zawierać także atrybuty klucze powiązanych relacji
- związki obejmujące słaby zbiór encji muszą wykorzystywać klucze powiązanych zbiorów encji
- nie zawsze jest konieczne dekomponowanie związków słabych zbiorów encji przy pomocy dodatkowych relacji



## ANOMALIE W SCHEMATACH R.B.D.

Tworzenie relacji z dużą liczbą atrybutów (np. poprzez łączenie innych relacji) prowadzi do następujących anomalii:

- **redundancja** – nadmiar danych w tabeli prowadzi do nadmiernego zużycia pamięci i spowalnia działanie systemu,
- **błąd wstawiania** – niemożność wstawienia krotki bez pełnych danych,
- **błąd modyfikacji** – pominięcie aktualizacji części krotek,
- **błąd usunięcia** – usunięcie wartości atrybutu jednej krotki powoduje utratę danych całej krotki.

marka	model	kolor	karos.	linia	typ_silnika	pojemność	wtrysk	paliwo
VW	golf	czarny	ocynk.	3drz.	DCV200D	1300	SI	ON
VW	golf	czarny	ocynk.	3drz.	DCV200	1300	SI	U95
VW	golf	czarny	ocynk.	5drz.	DCV200D	1300	SI	ON
VW	golf	czarny	ocynk.	5drz.	DCV200	1300	SI	U95
VW	golf	zielony	ocynk.	3drz.	DCV200D	1300	SI	ON
VW	golf	zielony	ocynk.	3drz.	DCV200	1300	SI	U95

## DEKOMPOZYCJA RELACJI

Relacja  $A: \{A_1, \dots, A_x\}$  jest dekomponowana na dwie relacje  $B: \{B_1, \dots, B_y\}$ , i  $C: \{C_1, \dots, C_z\}$ , takie, że:

$$1. \{B_1, \dots, B_y\} \cup \{C_1, \dots, C_z\} = \{A_1, \dots, A_x\}$$

2. krotki relacji  $B$  i  $C$  powstają przez zrzutowanie wszystkich krotek relacji  $A$ , przy czym kopie zostają usunięte

marka	model	kolor	karos.	linia	typ_silnika
VW	golf	czarny	ocynk.	3drz.	DCV200D
VW	golf	czarny	ocynk.	3drz.	DCV200
VW	golf	czarny	ocynk.	5drz.	DCV200D
VW	golf	czarny	ocynk.	5drz.	DCV200
VW	golf	zielony	ocynk.	3drz.	DCV200D
VW	golf	zielony	ocynk.	3drz.	DCV200

typ_silnika	pojemnosc	wtrysk	paliwo
DCV200D	1300	SI	ON
DCV200	1300	SI	U95

## ZALEŻNOŚCI FUNKCYJNE

Zależność funkcyjna:  $A_1, \dots, A_x \rightarrow B$

jeżeli dwie krotki danej relacji są zgodne dla atrybutów  $A_1, \dots, A_x$ , to muszą być zgodne dla pewnego atrybutu  $B$ .

W przypadku zależności funkcyjnej określającej zgodność dla większej liczby atrybutów, zapis jest następujący:

$$A_1, \dots, A_x \rightarrow B_1, \dots, B_y$$

Zależność funkcyjna  $A_1, \dots, A_x \rightarrow B_1, \dots, B_y$  jest:

- trywialna – jeśli  $B_1, \dots, B_y$  zawiera się w  $A_1, \dots, A_x$ ,
- nietywialna – jeśli istnieją  $B_i, B_j$  zawierające się w  $A$ ,
- całkowicie nietywialna – żaden atrybut z  $B$  nie należy do  $A$ .

- np.: imię, nazwisko, pesel  $\rightarrow$  nazwisko
- np.: imię, nazwisko, pesel  $\rightarrow$  pesel, płaca
- np.: imię, nazwisko, pesel  $\rightarrow$  płaca

*Pełna zależność funkcyjna* – pewien klucz jednoznacznie identyfikuje pewne wartości niekluczowe.

*Częściowa zależność funkcyjna* – pewien podklucz jednoznacznie identyfikuje pewne wartości niekluczowe.

### Zależności wielowartościowe

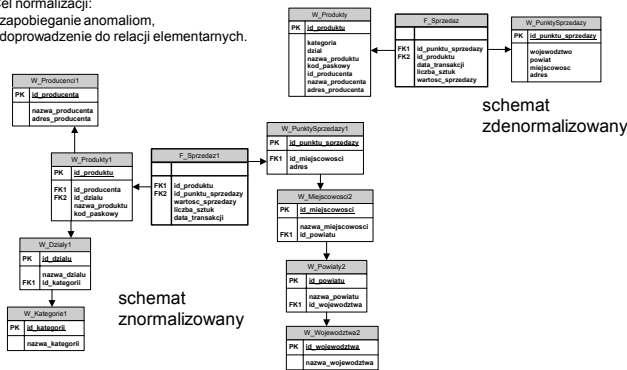
Zależność wielowartościowa:  $A \rightarrow B$

Znając wartość atrybutu  $A$  możemy określić zbiór możliwych wartości atrybutu  $B$ .

## NORMALIZACJA SCHEMATU RELACYJNEJ BAZY DANYCH

Cel normalizacji:

- zapobieganie anomalom,
- doprowadzenie do relacji elementarnych.

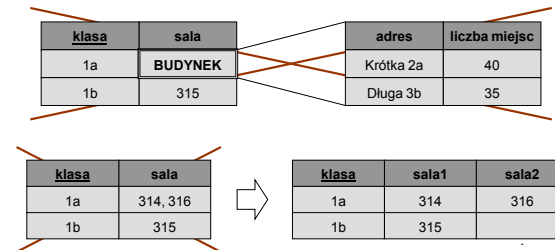


## PIERWSZA POSTAĆ NORMALNA (1NF)

Relacja jest w pierwszej postaci normalnej (1NF), jeśli wartości atrybutów są pojedynczymi wartościami atomowymi.

Tzn. wartość atrybutu pojedynczej krotki jest wartością atomową, o zdefiniowanym typie, nie występują tutaj agregaty danych (listy, tabele, etc)

Uwaga! W relacyjno-objektowych danych może zostać zdefiniowana klasa obiektu, która może stać się atrybutem.





## DRUGA POSTAĆ NORMALNA (2NF)

*Pełna zależność funkcyjna* – pewien klucz jednoznacznie identyfikuje pewne wartości niekluczowe.  
*Częściowa zależność funkcyjna* – pewien podklucz jednoznacznie identyfikuje pewne wartości niekluczowe.

Relacja  $X$  jest w postaci **2NF** jeśli jest **1NF** i nie zawiera częściowych zależności funkcyjnych (klucz określa wszystkie niekluczowe atrybuty).

- Dopuszczalne są zależności funkcyjne między atrybutami niekluczowymi.
- Mogą wystąpić zależności przechodnie.
- Niedopuszczalne są nietrywialne zależności funkcyjne, w których lewa strona jest podkluczem.

Notes( imię , nazwisko , imieniny , urodziny , znak zodiaku )

imię	nazwisko	imieniny*	urodziny	znak zodiaku
Adam	Nowak	24.12	8.08	lew
Agnieszka	Kowalska	21.01	2.06	bliźnięta

urodziny → znak zodiaku - {urodziny} nie wchodzi w skład klucza, więc zależność nie wyklucza postaci 2NF.  
 imię → imieniny - {imię} jest podkluczem, więc zależność wyklucza postać 2NF.

\* zakładamy imieniny dla danego imienia raz w roku (np. Adam), inaczej byłby to przykład zależności funkcyjnej wielowartościowej

## POSTAĆ NORMALNA Boyce-Codda (BCNF)

Relacja  $X$  jest w postaci **BCNF** <=>  
 jeśli zachodzi dla dowolnej nietrywialnej zależności  $A_1...A_x \rightarrow B$ ,  
 to  $A_1...A_x$  jest nadkluczem.

Uwagi:

- Nie ma żadnych zależności funkcyjnych:
  - przechodnich,
  - między atrybutami niekluczowymi,
  - między atrybutami niekluczowymi a podkluczem,
- Nie ma dodatkowych kluczy kandydujących.
- Jedyna zależność funkcyjna związana jest z kluczem głównym.

BCNF jest bardzo silnym ograniczeniem i czasem trudnym do wprowadzenia, zakłada się, że 3NF jest wystarczająca np. do tworzenia schematów hurtowni danych.

## TRZECIA POSTAĆ NORMALNA (3NF)

Relacja  $X$  jest w postaci **3NF** <=>  
 jeśli zachodzi dla dowolnej nietrywialnej zależności  $A_1...A_x \rightarrow B$ ,  
 to  $A_1...A_x$  jest nadkluczem **albo**  $B$  jest podkluczem.

Uwagi:

- Nie są możliwe zależności funkcyjne między atrybutami niekluczowymi.

Dla przykładu: Notes( imię , nazwisko , imieniny , urodziny , znak zodiaku )  
 urodziny → znak zodiaku - {urodziny} nie wchodzi w skład klucza, więc warunek 2NF jest spełniony, ale 3NF już nie.

- $B$  może być częścią innego klucza (w relacji mogą być różne zbiory atrybutów, tworzące klucze)

miasto	ulica	kod
Kraków	Mickiewicza	30-059
Kraków	Tokarskiego.	30-065
Gdańsk	Mickiewicza	80-405

Adresy ( miasto , ulica , kod )

Klucze: { miasto , ulica }, { ulica , kod } (zakładamy dla uproszczenia, że dany kod obowiązuje wzdłuż całej ulicy)

Zależności funkcyjne: { miasto , ulica } → kod,  
 kod → miasto (kod jest podkluczem, spełniony jest warunek 3NF, ale nie BCNF)

## NORMALIZACJA - przykład

uczniowie( nr\_dz , imię , nazwisko , klasa , sala )

nr_dz	imię	nazwisko	klasa	sala
1	Andrzej	Nowak	1a	314
2	Gustaw	Kot	1a	314.
3	Ala	Wójcik	1a	314.
1	Ola	Nowak	1b	315.
2	Gustaw	Werner	1b	315.
3	Ala	Wójcik	1b	315.

Klucz  $K = \{nr\_dz, klasa\}$

Zależności:

$nr\_dz, klasa \rightarrow nazwisko$

$nr\_dz, klasa, nazwisko \rightarrow sala$

$klasa \rightarrow sala$

{ nr\_dz, klasa } – nadklucz K

{ nr\_dz, klasa, nazwisko } – nadklucz K

{ klasa } - **jest podkluczem K, (a nie nadkluczem)**

Zatem przedstawiony model bazy danych nie spełnia warunków **2NF**, **3NF**, **BCNF**.



## OPERACJE NA DANYCH

## SUMA

Dane są relacje A i B:

 $A(\text{atr1}, \text{atr2}), B(\text{atr1}, \text{atr2})$ 

A	
ATR1	ATR2
1	x
2	y
3	z

B	
ATR1	ATR2
1	u
2	y

## Suma

Sumą relacji A i B ( $A \cup B$ )

jest zbiór wszystkich krotek należących do relacji A lub B lub ich części wspólnej.

-duplikaty są odrzucane  
-- lub nie

$A \cup B$	
ATR1	ATR2
1	x
2	y
3	z
1	u
2	y

$A \cup B$	
ATR1	ATR2
1	x
2	y
3	z
1	u

## OPERACJE NA DANYCH

## RÓŻNICA

Dane są relacje A i B:

 $A(\text{atr1}, \text{atr2}), B(\text{atr1}, \text{atr2})$ 

A	
ATR1	ATR2
1	x
2	y
3	z

B	
ATR1	ATR2
1	u
2	y

## Różnica

Różnicą relacji A i B ( $A - B$ ) jest

zbiór wszystkich krotek należących do relacji A i nienależących do relacji B ani części wspólnej A i B.

$A - B$	
atr1	atr2
1	x
3	z

## OPERACJE NA DANYCH

## ILO CZYŃ

Dane są relacje A i B:

 $A(\text{atr1}, \text{atr2}), B(\text{atr1}, \text{atr2})$ 

A	
ATR1	ATR2
1	x
2	y
3	z

B	
ATR1	ATR2
1	u
2	y

## ILO CZYŃ

Iloczynem relacji A i B ( $A \cap B$ ) jest

zbiór wszystkich krotek należących zarówno do relacji A jak i do relacji B (część wspólna).

Iloczyn można wyrazić przy pomocy różnicy:  $A \cap B = A - (A - B)$ 

$A \cap B$	
ATR1	ATR2
2	y

## OPERACJE NA DANYCH

## PROJEKCJA

## Rzut (projekcja)

Operacja rzutu  $\Pi$  relacji A przekazuje dane relacji A do innej (nowej) relacji B, której zbiór atrybutów  $\{b_1, \dots, b_n\}$  jest podzbiorem atrybutów A  $\{a_1, \dots, a_n\}$ 

A			
atr1	atr2	atr3	atr4
1	x	aa	dx
2	y	bb	dy

 $\Pi_{\text{atr1}, \text{atr2}, \text{atr4}}(A)$ 

atr1	atr2	atr4
1	x	dx
2	y	dy

liczba krotek się nie zmienia, ale mogą powstać duplikaty

## OPERACJE NA DANYCH

## SELEKCJA

## Selekcja

Operacja selekcji  $\sigma$  na relacji A tworzy podzbiór krotek, spełniających kryterium F, operujące na atrybutach A.

A

atr1	atr2	atr3	atr4
1	x	aa	123
2	y	bb	234
3	x	aa	576
4	y	bb	121
3	x	aa	260

$\sigma_{atr1 < 4, atr4 < 250} (A)$

atr1	atr2	atr3	atr4
1	x	aa	123
2	y	bb	234

## OPERACJE NA DANYCH

## Iloczyn kartezjański

A

atr1	atr2	atr3
11	x	aa
22	y	bb
33	x	bb

B

atr2	atr3	atr4	atr5
x	aa	123	11
y	bb	234	22

A  $\bowtie$  B

atr1	A.atr2	A.atr3	B.atr2	B.atr3	atr4	atr5
11	x	aa	x	aa	123	11
11	x	aa	y	bb	234	22
22	y	bb	x	aa	123	11
22	y	bb	y	bb	234	22
33	x	bb	x	aa	123	11
33	x	bb	y	bb	234	22

## OPERACJE NA DANYCH

Złączenie  $\theta$  (teta)

Złączenie  $\theta$  dwóch relacji A i B ( $A \bowtie_{\theta} B$ ) powstaje poprzez selekcję wyników  $A \times B$  przy użyciu pewnego warunku  $\theta$ . Warunek  $\theta$  dotyczy najczęściej wybranych atrybutów relacji A i B.

A

atr1	atr2	atr3
11	x	aa
22	y	bb
33	x	bb
44	y	bb

B

atr2	atr3	atr4	atr5
x	aa	123	11
y	bb	234	22

$A \bowtie_{\theta} B$

$\theta: atr1 > atr5$

atr1	A.atr2	A.atr3	B.atr2	B.atr3	atr4	atr5
22	y	bb	x	aa	123	11
33	x	bb	x	aa	123	11
44	y	bb	x	aa	123	11
33	x	bb	y	bb	234	22
44	y	bb	y	bb	234	22

## OPERACJE NA DANYCH

## ILORAZ

Dane są relacje  $A = \{a_1, a_2, \dots, a_m\}$  i  $B = \{b_1, b_2, \dots, b_k\}$ .  $B \neq \emptyset$ ,  $m > k$

Ilorazem relacji A i B ( $A \div B$ ) jest nowa relacja C, która zawiera te krotki A, które zawierają wszystkie kombinacje krotek B.

Zbiór atrybutów relacji C zawiera jedynie atrybuty należące do relacji A i nienależące do relacji B.

A

atr1	atr2	atr3	atr4
1	x	aa	12
2	y	aa	12
3	z	aa	12
1	x	bb	34
2	y	bb	34
3	z	bb	34
1	x	cc	56
2	y	cc	56

B

atr3	atr4
aa	12
bb	34
cc	56

A  $\div$  B

atr1	atr2
1	x
2	y
3	z

C

3	z	cc	56
---	---	----	----

## OPERACJE NA DANYCH

### ZŁĄCZENIE NATURALNE

A (atr1, atr2, atr3,...), B (atr2, atr3, atrbis1, atrbis2,...)

Złączenie naturalne relacji A i B ( $A \bowtie B$ ) kreuje nową relację, której zbiór atrybutów powstaje jako suma teoriomnogościowa zbiorów atrybutów A i B. Operacja przebiega następująco:

- wyznacza się  $A \times B$ ,
- z wyznaczonego iloczynu kartezjańskiego wybiera się tylko te krotki, które mają te same wartości dla atrybutów wspólnych dla A i B ( $A.attr2 \leftrightarrow B.attr2$ ,  $A.attr3 \leftrightarrow B.attr3$ )
- dokonuje się rzutu usuwającego zdublowane kolumny ( $A.attr2, B.attr2 \rightarrow attr2, \dots$ )

A			B			
atr1	atr2	atr3	atr2	atr3	atr4	atr5
1	x	aa	x	aa	123	11
2	y	bb	y	bb	234	22
3	x	bb				
4	y	bb				

## OPERACJE NA DANYCH

### ZŁĄCZENIE NATURALNE

1

A x B						
atr1	A.attr2	A.attr3	B.attr2	B.attr3	atr4	atr5
1	x	aa	x	aa	123	11
2	y	bb	x	aa	123	11
3	x	bb	x	aa	123	11
4	y	bb	x	aa	123	11
1	x	aa	y	bb	234	22
2	y	bb	y	bb	234	22
3	x	bb	y	bb	234	22
4	y	bb	y	bb	234	22

2

atr1	A.attr2	A.attr3	B.attr2	B.attr3	atr4	atr5
1	x	aa	x	aa	123	11
2	y	bb	y	bb	234	22
4	y	bb	y	bb	234	22

3

A $\bowtie$ B					
atr1	atr2	atr3	atr4	atr5	
1	x	aa	123	11	
2	y	bb	234	22	
4	y	bb	234	22	

## OPERACJE NA DANYCH

### RÓWNOZŁĄCZENIE

Szczególny przypadek złączenia  $\theta$ , w którym warunek  $\theta$  łączy krotki o tej samej wartości dla wybranych atrybutów

$\theta$ :  $A.attr2 = B.attr3$

! Iloczyn kartezjański

A		B	
atr1	atr2	atr3	atr4
1	x	x	AA
2	y	y	BB
3	x	v	CC

A $\bowtie_{A.attr2 = B.attr3}$ B			
atr1	atr2	atr3	atr4
1	x	x	AA
2	y	y	BB
3	x	x	AA

## OPERACJE NA DANYCH

### Złączenie wewnętrzne (INNER JOIN)

Złączenie wewnętrzne to szczególny przypadek złączenia  $\theta$ , w którym warunek  $\theta$  łączy krotki dla wybranych atrybutów zadanym operatorem porównania ( $=, <, >, <=, >=$ )

$\theta$ :  $A.attr2 > B.attr3$

A		B	
atr1	atr2	atr3	atr4
1	x	x	AA
2	y	y	BB
3	x	v	CC

A $\bowtie_{A.attr2 > B.attr3}$ B			
atr1	atr2	atr3	atr4
1	x	v	CC
2	y	x	AA
2	y	v	CC
3	x	v	CC

## OPERACJE NA DANYCH

### Złączenie zewnętrzne lewostronne (LEFT OUTER JOIN)

Wynikiem złączenia zewnętrznego lewostronnego

A LEFT OUTER JOIN B ON (atrybutyzłączenia)

są wszystkie krotki relacji A, zawierające wartości atrybutów

z relacji B dla uzgodnionych warunkiem złączenia (atrybutyzłączenia) krotek z relacji B lub wartości puste.

A		B	
atr1	atr2	atr3	atr4
1	x	x	AA
2	x	y	BB
3	z	v	CC

A LEFT OUTER JOIN B ON (A.atr2 = B.atr3)

atr1	atr2	atr3	atr4
1	x	x	AA
2	x	x	AA
3	z		

Handwritten notes on a whiteboard illustrating a LEFT OUTER JOIN. It shows two tables, A and B, and the result of joining them. Table A has columns 'imie' and 'nazwisko' with rows: (Janina, Kocjan), (Adam, Wodan), (Kamil, Edan). Table B has columns 'imie' and 'Peseł' with rows: (Janina, 111), (Kamir, 222), (Zukure, 333), (Kamil, 444), (V.MWL, 555). A circle contains the text 'left join A na B'. The resulting table has columns 'imie', 'nazwisko', 'imie', 'Peseł' with rows: (Janina, Kocjan, Janina, 111), (Kamil, Edan, Kamil, 444), (Zukure, , Zukure, 333), (V.MWL, , V.MWL, 555).

Studenci Inżynierii Biomedycznej słowo 'join' rozumieją jako 'staw'.

Etap szkolenia przyszłej kadry programistycznej – Informatyka Stosowana

## OPERACJE NA DANYCH

### Złączenie zewnętrzne prawostronne (RIGHT OUTER JOIN)

Wynikiem złączenia zewnętrznego prawostronnego

A LEFT OUTER JOIN B ON (atrybutyzłączenia)

są wszystkie krotki relacji B, zawierające wartości atrybutów

z relacji A dla uzgodnionych warunkiem złączenia (atrybutyzłączenia) krotek z relacji A lub wartości puste.

A		B	
atr1	atr2	atr3	atr4
1	x	x	AA
2	x	y	BB
3	z	u	CC

A RIGHT OUTER JOIN B ON (A.atr2 = B.atr3)

atr1	atr2	atr3	atr4
1	x	x	AA
2	x	x	AA
		y	BB
		u	CC

## OPERACJE NA DANYCH

### Złączenie zewnętrzne pełne (FULL OUTER JOIN)

Wynikiem złączenia zewnętrznego prawostronnego

A LEFT OUTER JOIN B ON (atrybutyzłączenia)

są wszystkie krotki relacji B, zawierające wartości atrybutów

z relacji A dla uzgodnionych warunkiem złączenia (atrybutyzłączenia) krotek z relacji A lub wartości puste.

A		B	
atr1	atr2	atr3	atr4
1	x	x	AA
2	x	y	BB
3	z	u	CC

A FULL OUTER JOIN B ON (A.atr2 = B.atr3)

atr1	atr2	atr3	atr4
1	x	x	AA
2	x	x	AA
		y	BB
		u	CC
3	z		

## OPERACJE NA DANYCH

### Autozłączenie (self-join)

Autozłączenie to operacja łączenia danych z tej samej tabeli.

Nr	Imię	Nazwisko	Nr szefa
1	Jerzy	Nowak	
2	Anna	Wójcik	1
3	Aneta	Mazur	2

→ s-j

Nr	Imię	Nazwisko	Sz.Imię	Sz.Nazw
1	Jerzy	Nowak		
2	Anna	Wójcik	Jerzy	Nowak
3	Aneta	Mazur	Anna	Wójcik

### Złączenie częściowe (semi join)

Złączenie częściowe to takie złączenie dwóch tabel, w którym wyniki pochodzą tylko z jednej tabeli. Stosowane przy dekompozycji z zapytania zagnieżdżonego lub przy ograniczaniu rozwiązań.

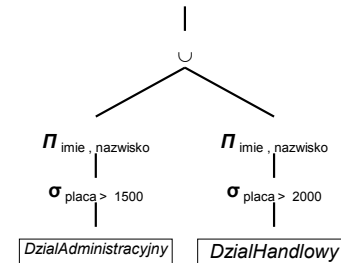
## OPERACJE NA DANYCH

### ZŁOŻENIA

Algebra w modelu relacyjnym dopuszcza tworzenie złożonych operacji. Kolejność wykonywania można określić operatorem ().

$(\pi_{\text{IMIE, NAZWISKO}}(\sigma_{\text{PLACA} > 1500}(\text{DzialAdministracyjny}))) \cup (\pi_{\text{IMIE, NAZWISKO}}(\sigma_{\text{PLACA} > 2000}(\text{DzialHandlowy})))$

Powyższe złożenie można przedstawić przy pomocy drzewa operacji:



## Przykładowe kwerendy w LibreOffice

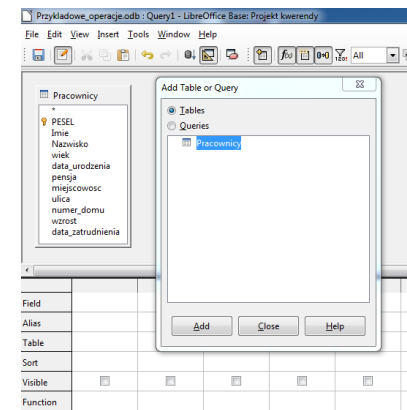
Przykładowa tabela z danymi

Field Name	Field Type
PESEL	BigInt [ BIGINT ]
Imie	Text [ VARCHAR ]
Nazwisko	Text [ VARCHAR ]
wiek	Small Integer [ SMALLINT ]
data_urodzenia	Date [ DATE ]
penja	Decimal [ DECIMAL ]
miiejscowosc	Text [ VARCHAR ]
ulica	Text [ VARCHAR ]
numer_domu	Text [ VARCHAR ]
wzrost	Decimal [ DECIMAL ]
data_zatrudnienia	Date [ DATE ]

PESEL	Imie	Nazwisko	wiek	data_urodzenia	penja	miiejscowosc	ulica	numer_domu	wzrost	data_zatrudnienia
222222	Ula	Nowak	29	1.05.1986	3500,00	Rzeszów	Ciepłinski 1		170,0	15.04.2013
333333	Ela	Nowak	31	2.06.1984	5500,00	Kraków	Nawojki 1		180,0	15.04.2012
11111111	Ola	Nowak	32	1.03.1983	6000,00	Warszawa	Piękna 1		160,5	15.08.2013
123456567	Ala	Nowak	30	1.01.1985	5000,00	Warszawa	Słiczna 1		175,0	15.08.2012
987654321	Ewa	Nowak	28	1.02.1987	4000,00	Kraków	Wiedowi 1		165,0	15.04.2011

## Przykładowe kwerendy w LibreOffice



Wybór tabel biorących udział w kwerendzie – na początek – jedna

## Przykładowe kwerendy w LibreOffice

uruchomienie zapytania

Przykładowe operacje.odt : Query1 - LibreOffice Base: Projekt kwerend

File Edit View Insert Tools Window Help

Record 1 of 5

Imie	Nazwisko	data_zatrudnienia
Ula	Nowak	15.04.2013
Ela	Nowak	15.04.2012
Ola	Nowak	15.08.2013
Ala	Nowak	15.08.2012
Ewa	Nowak	15.04.2011

Field Imie Nazwisko data\_zatrudnienia

Alias Pracownicy PESEL

Table Pracownicy Pracownicy

Sort Imie Nazwisko wiek

Visible

Function data\_urozdenia pensja miejscowosc ulica numer\_domu wzrost

Criterion

Or

wynik

Projekt kwerendy

PROJEKCJA

Field Imie Nazwisko

Alias Pracownicy PESEL

Table Pracownicy Pracownicy

Sort Imie Nazwisko wiek

Visible

Function data\_urozdenia pensja miejscowosc ulica numer\_domu wzrost

Criterion

Or

Pracownicy PESEL  
Pracownicy.Imie  
Pracownicy.Nazwisko  
Pracownicy.wiek  
Pracownicy.data\_urozdenia  
Pracownicy.pensja  
Pracownicy.miejscowosc  
Pracownicy.ulica  
Pracownicy.numer\_domu  
Pracownicy.wzrost

Pracownicy.Imie  
Pracownicy.Nazwisko  
Pracownicy.pensja  
Pracownicy.data\_zatrudnienia

wybor pól

## Przykładowe kwerendy w LibreOffice

Query Wizard

Steps

1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

Select the fields (columns) for your query

Tables  
Table: Pracownicy

Available fields

PESEL  
wiek  
data\_urozdenia  
miejscowosc  
ulica  
numer\_domu  
wzrost

Fields in the Query:

Pracownicy.Imie  
Pracownicy.Nazwisko  
Pracownicy.pensja  
Pracownicy.data\_zatrudnienia

Help < Back Next > Finish Cancel

Inna metoda: użycie tzw. wizzarda

## Przykładowe kwerendy w LibreOffice

Query Wizard

Steps

1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

Select the sorting order

Sort by  
Pracownicy.Nazwisko  Ascending  Descending

Then by  
Pracownicy.Imie  Ascending  Descending

Then by  
- undefined -  Ascending  Descending

Then by  
- undefined -  Ascending  Descending

Help < Back Next > Finish Cancel

sortowanie danych

## Przykładowe kwerendy w LibreOffice

Query Wizard

Steps

1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

Select the search conditions

Match all of the following  
 Match any of the following

Fields Condition Value

Pracownicy.pensja is greater than 3000

Fields Condition Value

Pracownicy.Imie like E|

Fields Condition Value

is equal to

Help < Back Next > Finish Cancel

Przykładowe operacje - LibreOffice

Window Help

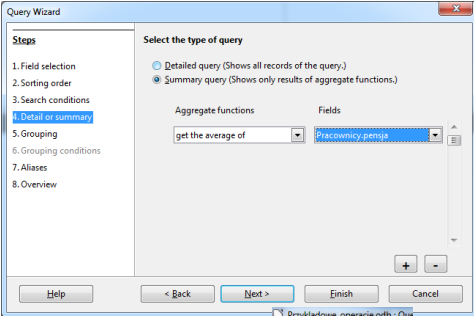
Imie	Nazwisko	pensja	data_zatrudnienia
Ela	Nowak	5500,00	15.04.2012
Ewa	Nowak	4000,00	15.04.2011

Wprowadzenie przykładowych warunków dla liczb (pensja) i tekstów (Imie)



## Przykładowe kwerendy w LibreOffice

**GRUPOWANIE**



wizzard

Field	pensja	miescowosc
Alias	pensja	
Table	Pracownicy	Pracownicy
Sort		
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function	Average	Group

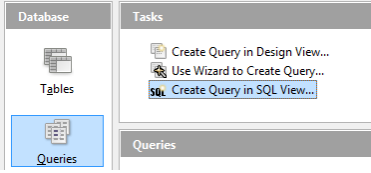
projekt

wynik

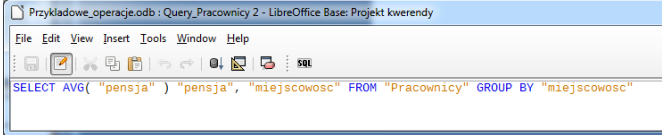
pensja	miescowosc
3500	Rzeszów
4750	Kraków
5500	Warszawa

## Przykładowe kwerendy w LibreOffice

**GRUPOWANIE**



Trzecia metoda tworzenia kwerend



```
SELECT AVG( "pensja" ) "pensja", "miescowosc" FROM "Pracownicy" GROUP BY "miescowosc"
```