

# BAZY DANYCH

## Relacyjne Bazy Danych

kierunek: Informatyka Medyczna

**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/BDIM/>

# 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

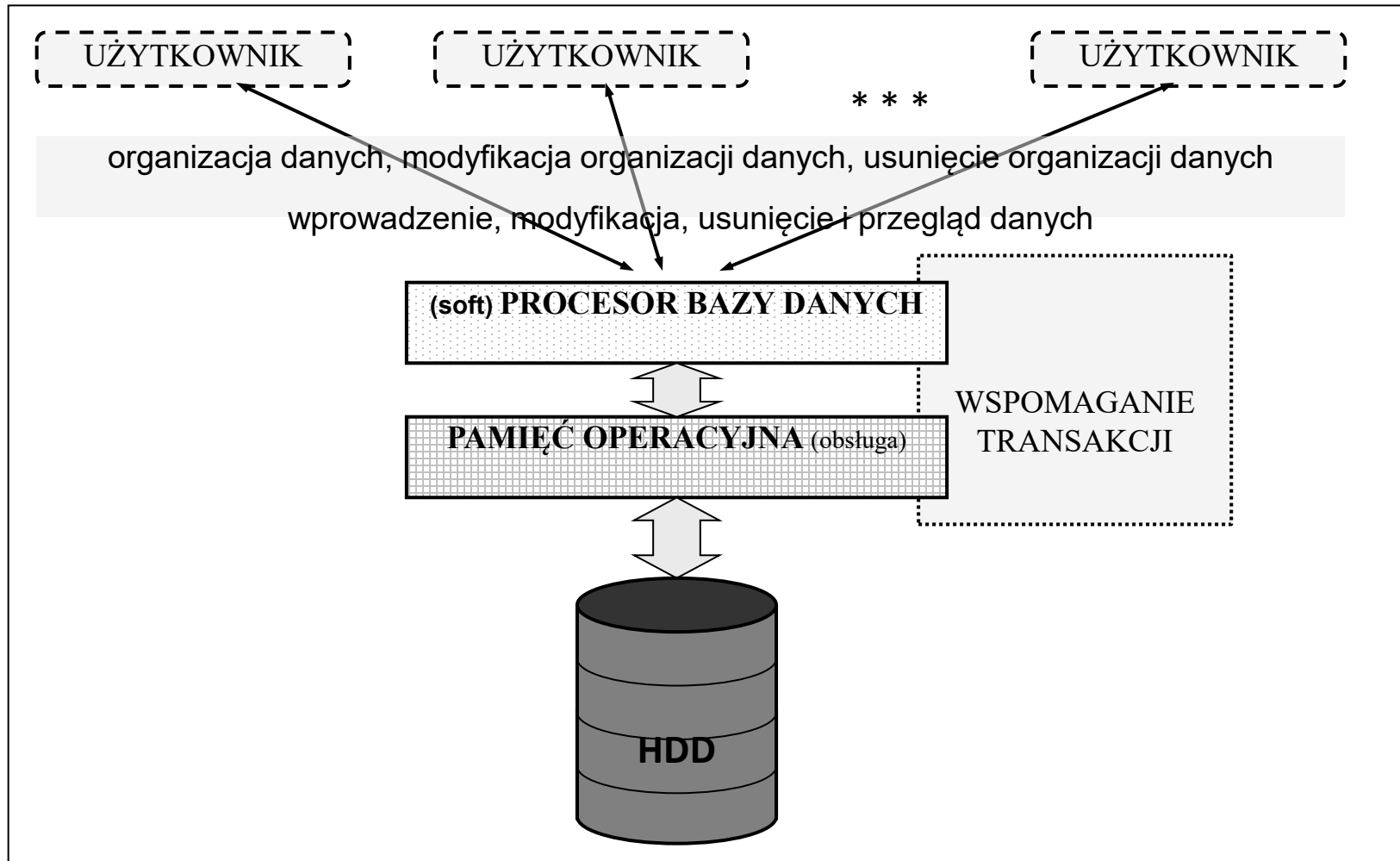
**Bank danych** – baza danych służąca do przechowywania danych.

**Repozytorium danych** – baza archiwizowanych danych.

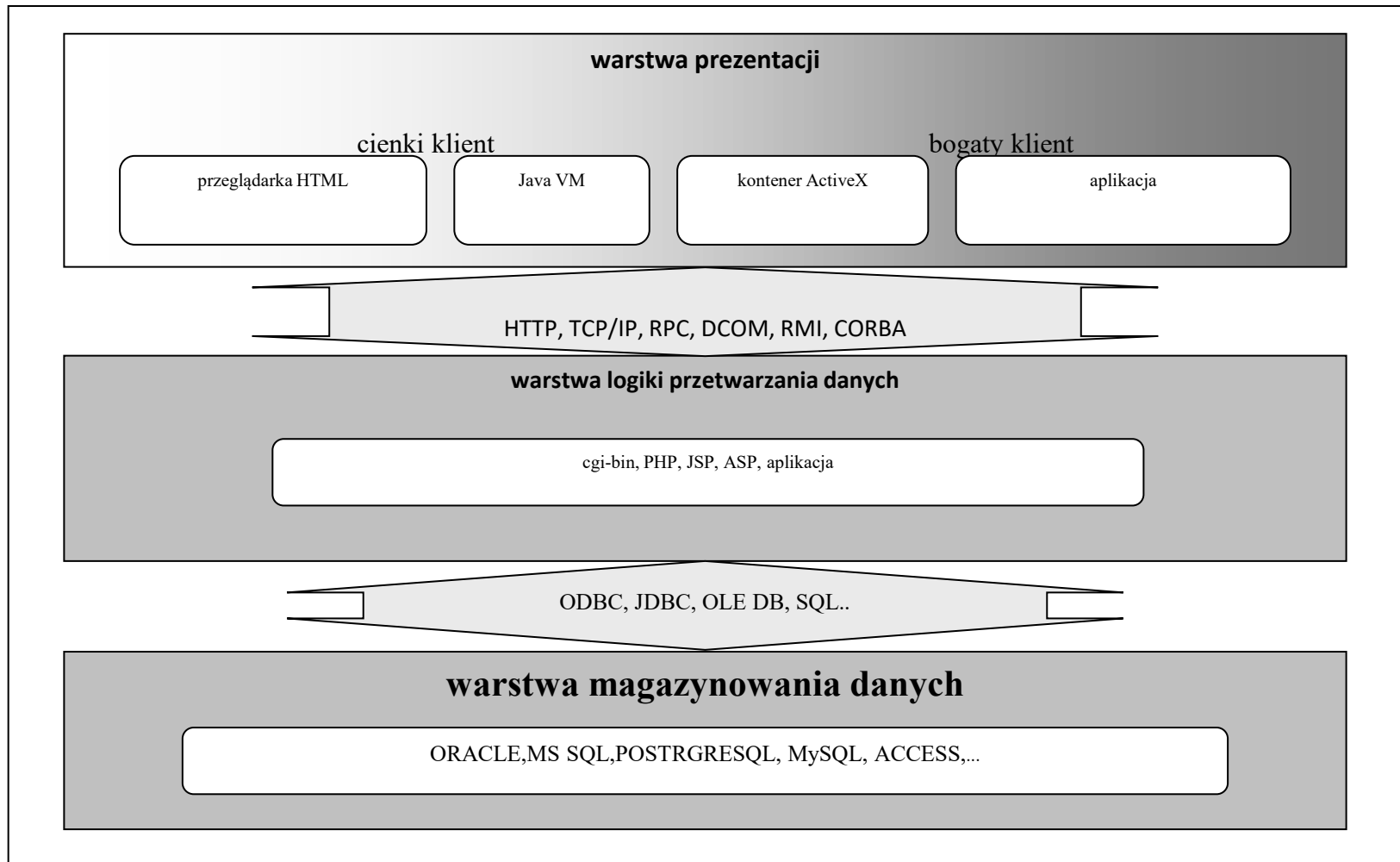
**Hurtownia danych** – baza nadrzędna, ściągająca okresowo dane z podrzędnych baz operacyjnych, zawierająca dane zweryfikowane, opcjonalnie kumulowane i selekcjonowane. Przeznaczona do szybkiego wyszukiwania tematycznego. Często nie jest bazą relacyjną.

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

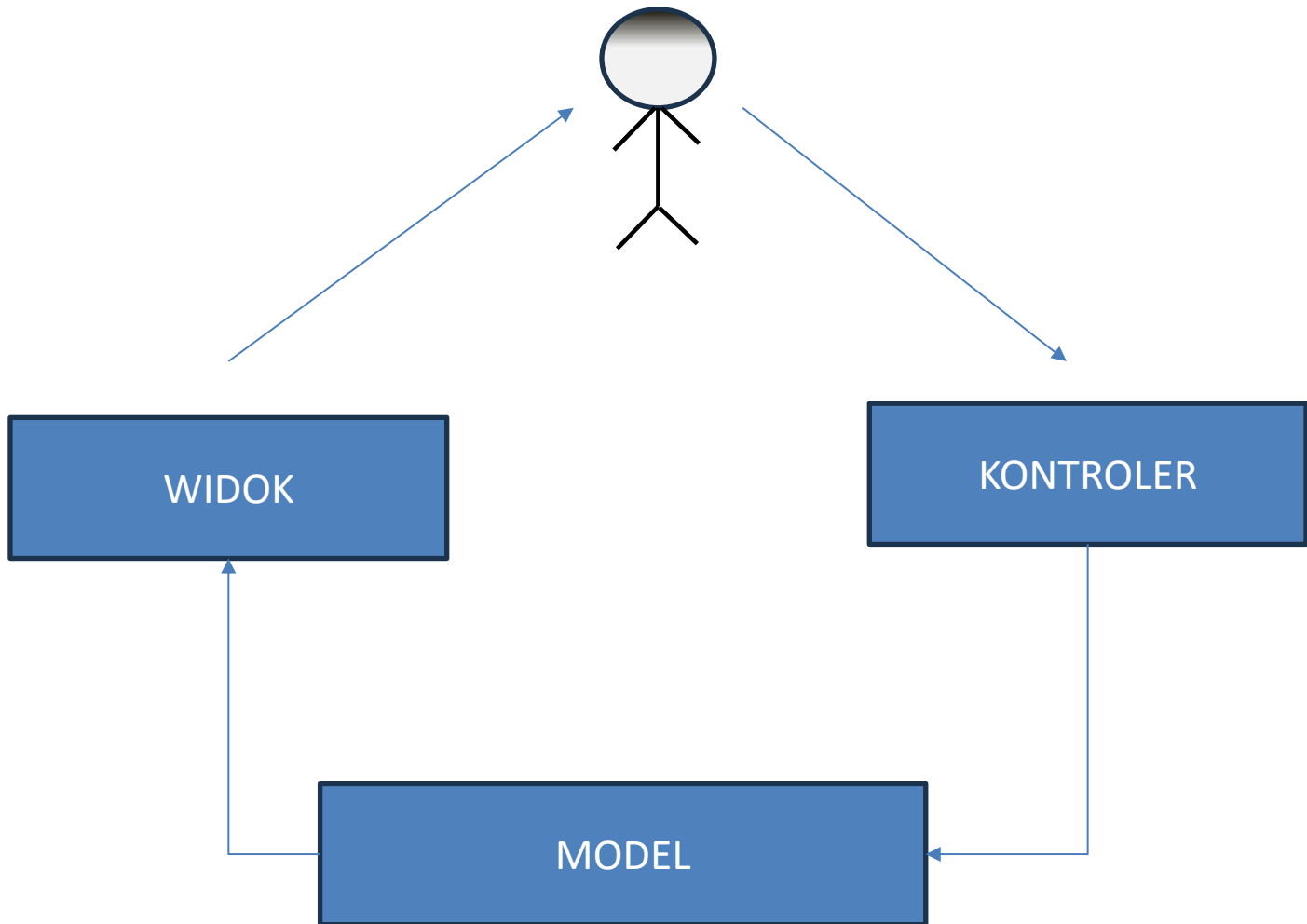
# Składowe Systemu Zarządzania Bazą Danych



# Model trójwarstwowy



# Model-View-Controller



# 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 wyzwała 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.

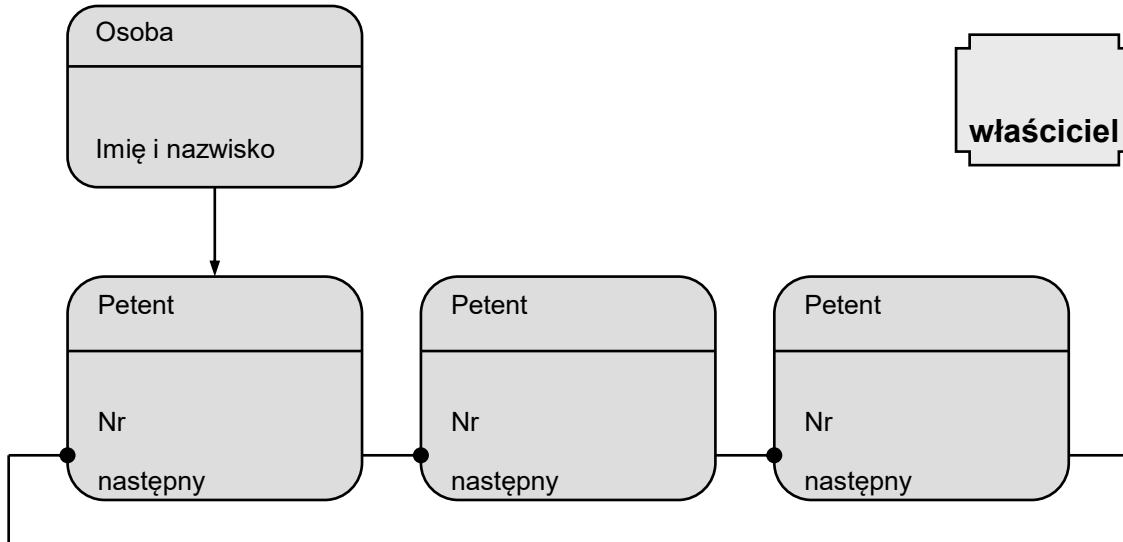
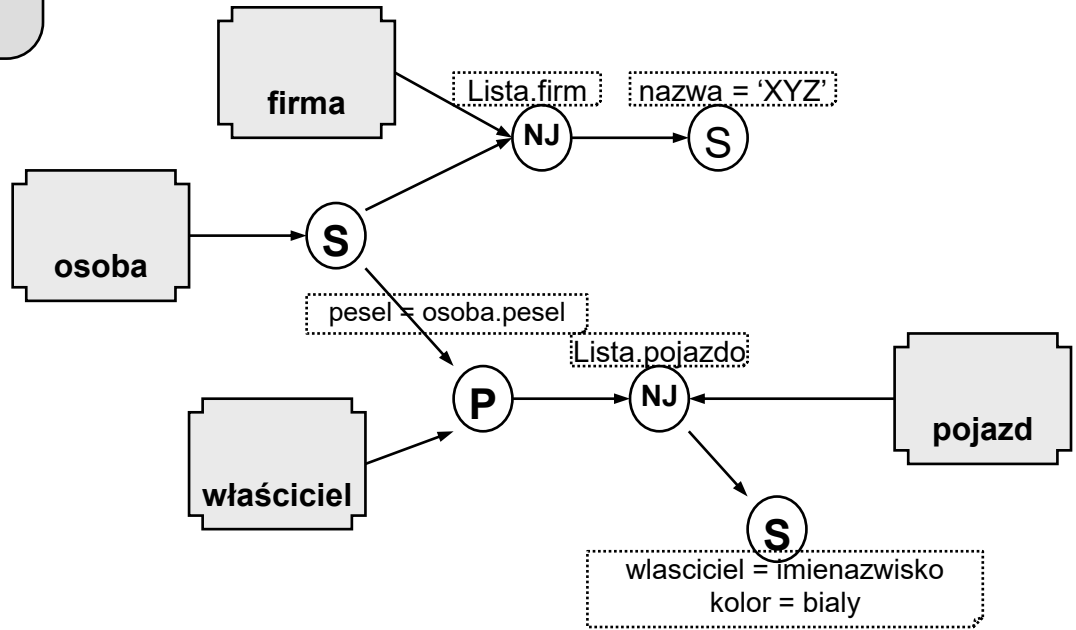
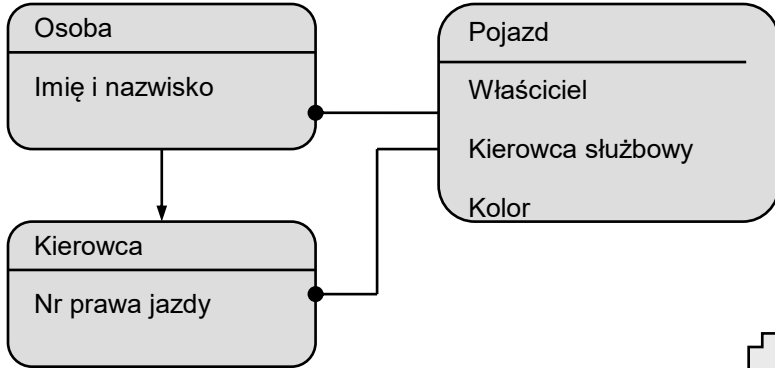
# Rodzaje baz danych – organizacja danych

- **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).

# Obiektowe bazy danych



# 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.

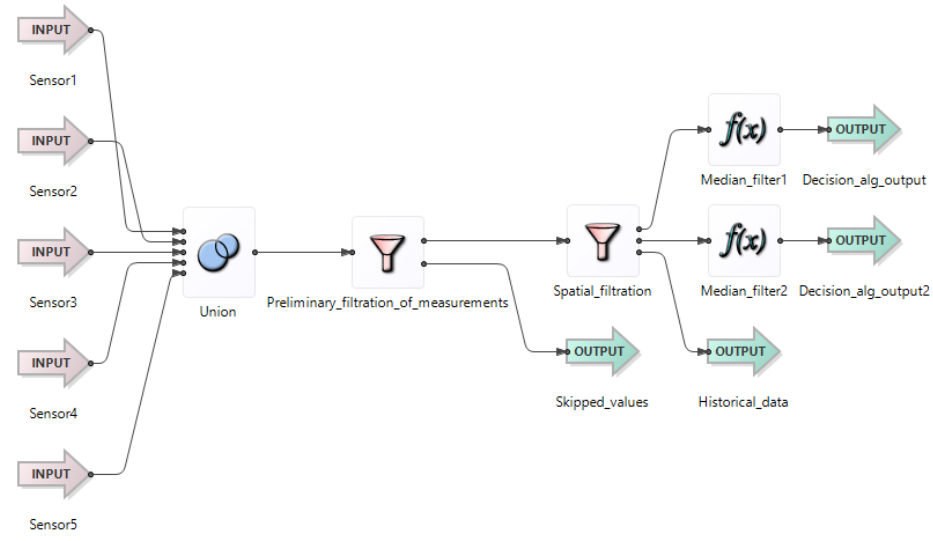
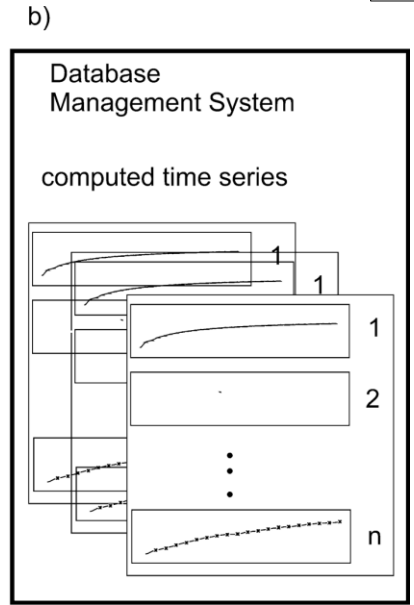
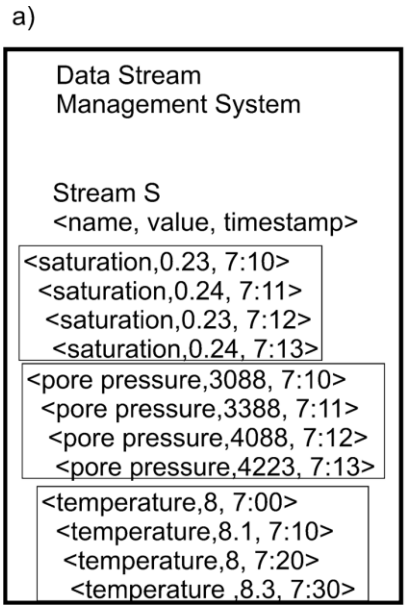
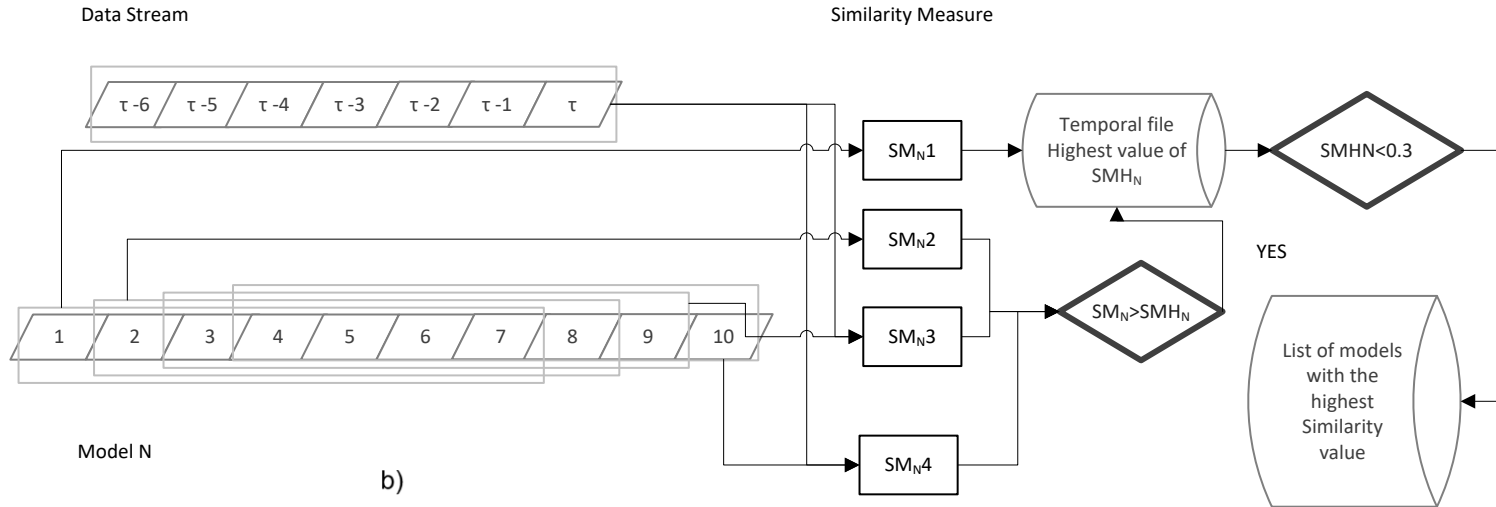
# Rodzaje baz danych – organizacja danych

- **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



Chuchro, M., Lupa, M., Pięta, A., Piórkowski, A., & Leśniak, 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).

Pięta, A., Lupa, M., Chuchro, M., Piórkowski, A., & Leśniak, 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).

# 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 standaryzowany, 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 standaryzowanym językiem do operacji na dokumentach XML, zawierającym (rozszerzającym) język XPath.

# XML-owe bazy danych

```
<?xml version="1.1" encoding="ISO-8859-2"?>
<tag_glowny>
<nazwatagu>dane</nazwatagu>
<tagpustybezdanychiparametrow/>
<tagzparametrami parametr="pierwszy" opcja="druga">dane</tagzparametrami>
<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>
```

# Format JSON (JavaScript Object Notation)

```
{  
  "Opis": "Dane podstawowe",  
  "Wyswietlanie wlaczone": 1,  
  "Rozmiar bloku": [512,512,288],  
  "LiniaX": [158, 158, 158, 158, 158, 155, 155, 155, 155],  
  "LiniaY": [417, 417, 417, 417, 417, 417, 417, 417, 417],  
  "Projekcje": [ {"nazwa": "4 komory", "a": -40, "b": 0, "c": 5},  
  {"nazwa": "2 komory", "a": -55, "b": 0, "c": 90},  
  {"nazwa": „1 komora", "a": -60, "b": 0, "c": 70} ],  
  "status": "2",  
  "Bledy": ""  
}
```

# Rodzaje baz danych – organizacja danych

- **‘Przestrzenne bazy danych’ (Bazy danych przestrzennych)**

– bazy, które posiadają specyfikowany 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))'));

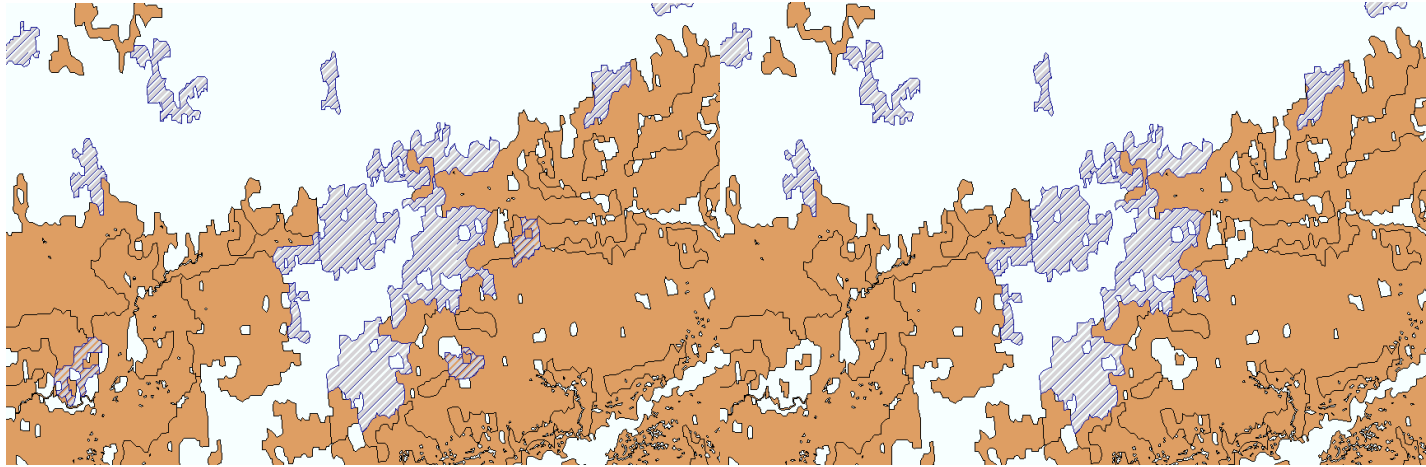
INSERT INTO Drogi(id_drogi, przebieg)
VALUES (1, GeomFromText('LINESTRING(0 10, 15 10, 25 13)', -1));
```

```
mysql> SELECT Nazwa_gminy, Area(obszar) FROM Gminy;
```

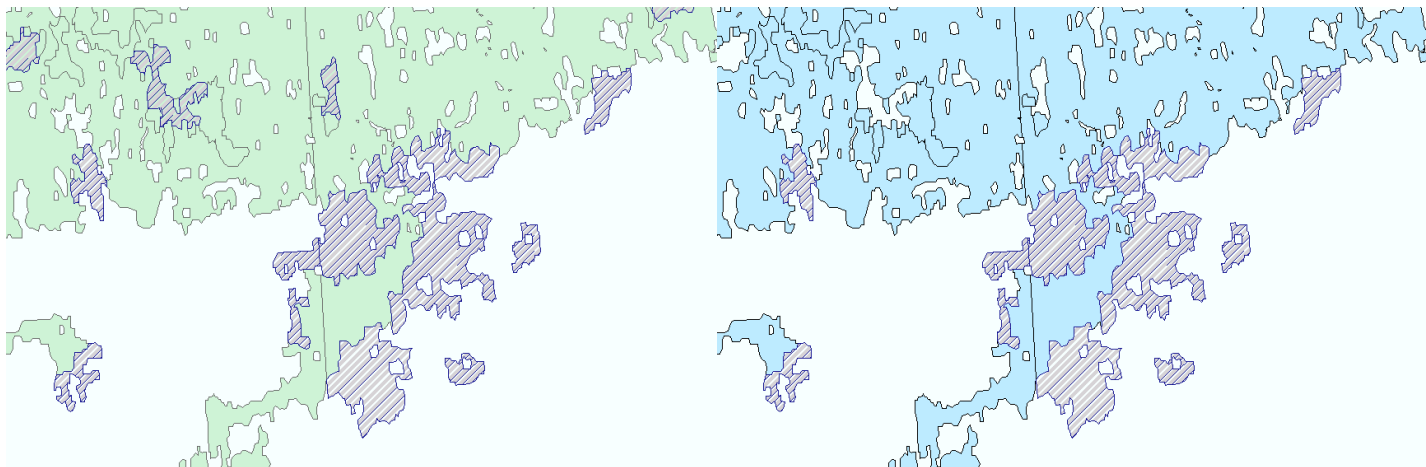
```
+-----+-----+
| Nazwa_gminy | Area(obszar) |
+-----+-----+
| Staropolska |          240 |
+-----+-----+
1 row in set (0.00 sec)
```

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

# Bazy danych przestrzennych



Input layers: trees and swamp (L), disjoint of input layers (R)



Input layers: tundra and swamp (L), disjoint of input layers (R)

# Bazy danych przestrzennych

-- czesc wspolna

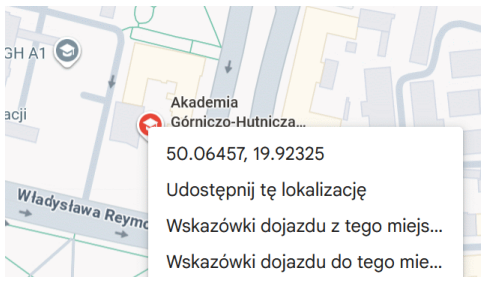
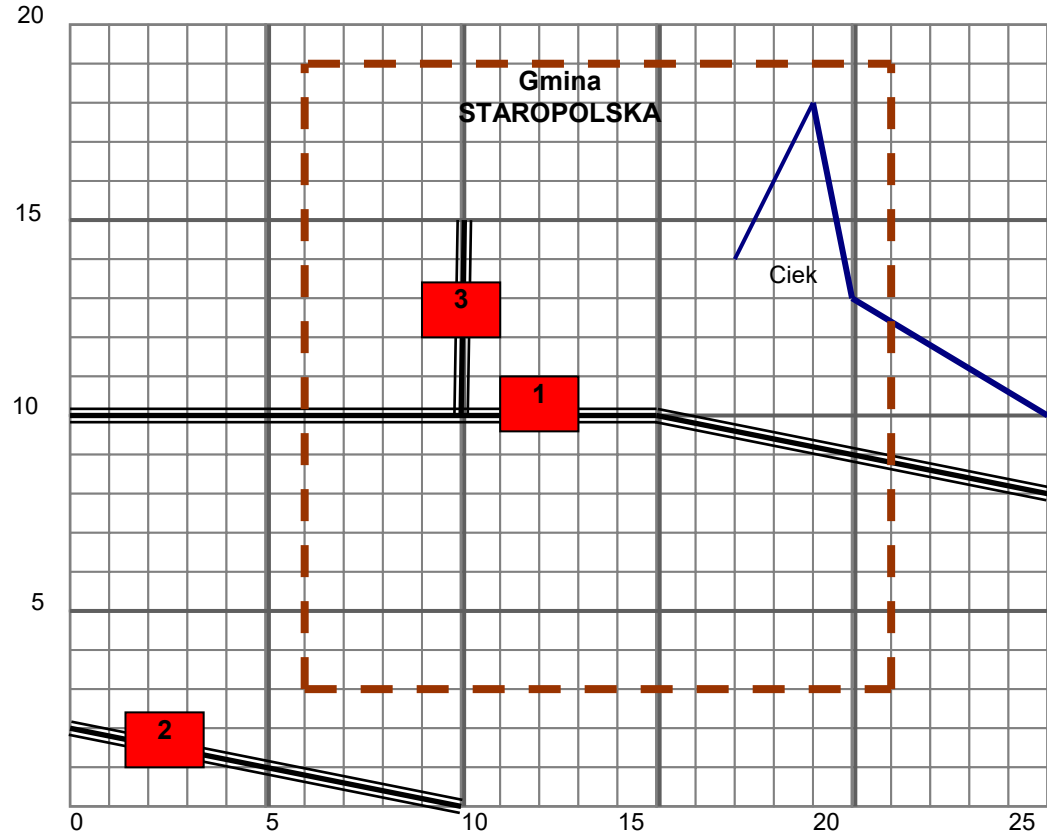
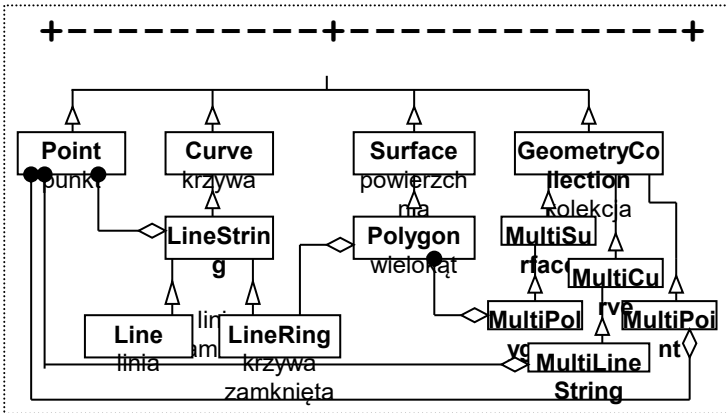
```
SELECT id_drogi, Nazwa_gminy FROM Drogi, Gminy
```

```
WHERE Intersects(obszar, przebieg);
```

```

+-----+-----+
| id_drogi | Nazwa_gminy |
+-----+-----+
|          |              |
|          |              |
|          |              |
|          |              |
+-----+-----+

```



# Systemy zarządzania bazami danych

- **SQL-owe bazy danych – bazy relacyjne**
  - Darmowe: MySQL, PostgreSQL, SQLite
  - Komercyjne: Oracle, MS SQL Server, IBM DB2
- **Bazy do niezaawansowanych 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

# 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 zabezpieczane mechanizmami transakcji (dla różnych poziomów izolacji). Dostępne są blokady.

# 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 kartezyjański danych z tabel.

# 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 bytecode)
- **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:

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

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

Relacje - przykład:

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

$$R_1 = \{ (a,c), (b,c), (b,d) \}$$

$$R_2 = \{ (a,c) \}$$

$$R_3 = \{ (a,d), (b,c) \}$$

$$R_4 = \{ (a,c), (b,c), (a, d), (b,d) \}$$

...

# RELACYJNE BAZY DANYCH

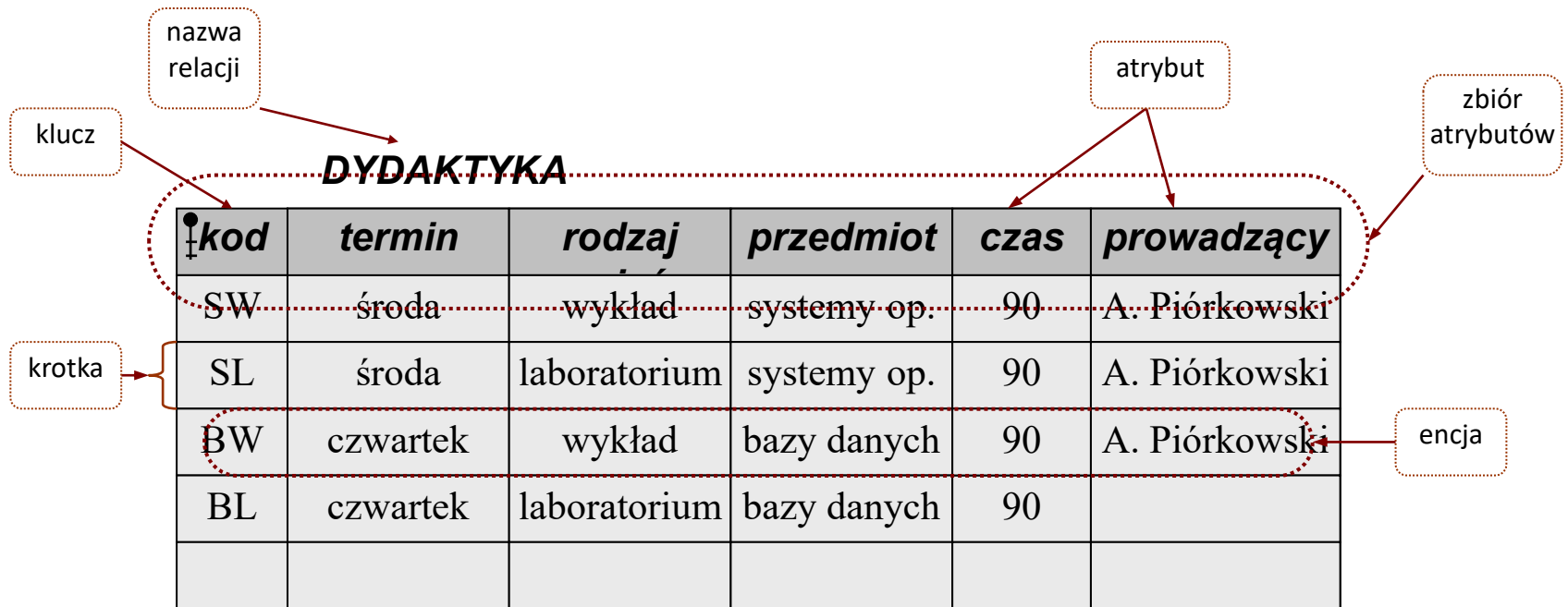
Iloczyn kartezjański - przykład:

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

Relacje - przykład:

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

# Koncepcja relacyjnej bazy danych



# Koncepcja relacyjnej bazy danych

**Encja** – elementarna porcja danych, opisujących jakiś byt, posiadający pewne własności.

**Atrybut** – cecha wspólna encji z danego zbioru encji.  
Każda relacja posiada zbiór atrybutów – każdy atrybut stanowi kolumnę w tabeli danej relacji.

**Schemat relacji** = nazwa relacji + zbiór atrybutów.  
Każdy wiersz dowolnej tabeli odpowiada dokładnie jednemu **rekordowi**.

**Krotka** – wiersz tabeli zawierający dane – odpowiada jednemu rekordowi.

Dane relacji – zbiór wszystkich krotek.

Krotki nie muszą mieć unikalnych danych (opcjonalne).

Kolejność krotek jest przypadkowa, najczęściej jest kolejnością wpisu.

**Klucz** – jest to atrybut (lub zestaw atrybutów), którego wartości są unikalne dla wszystkich krotek.

schemat  
relacji

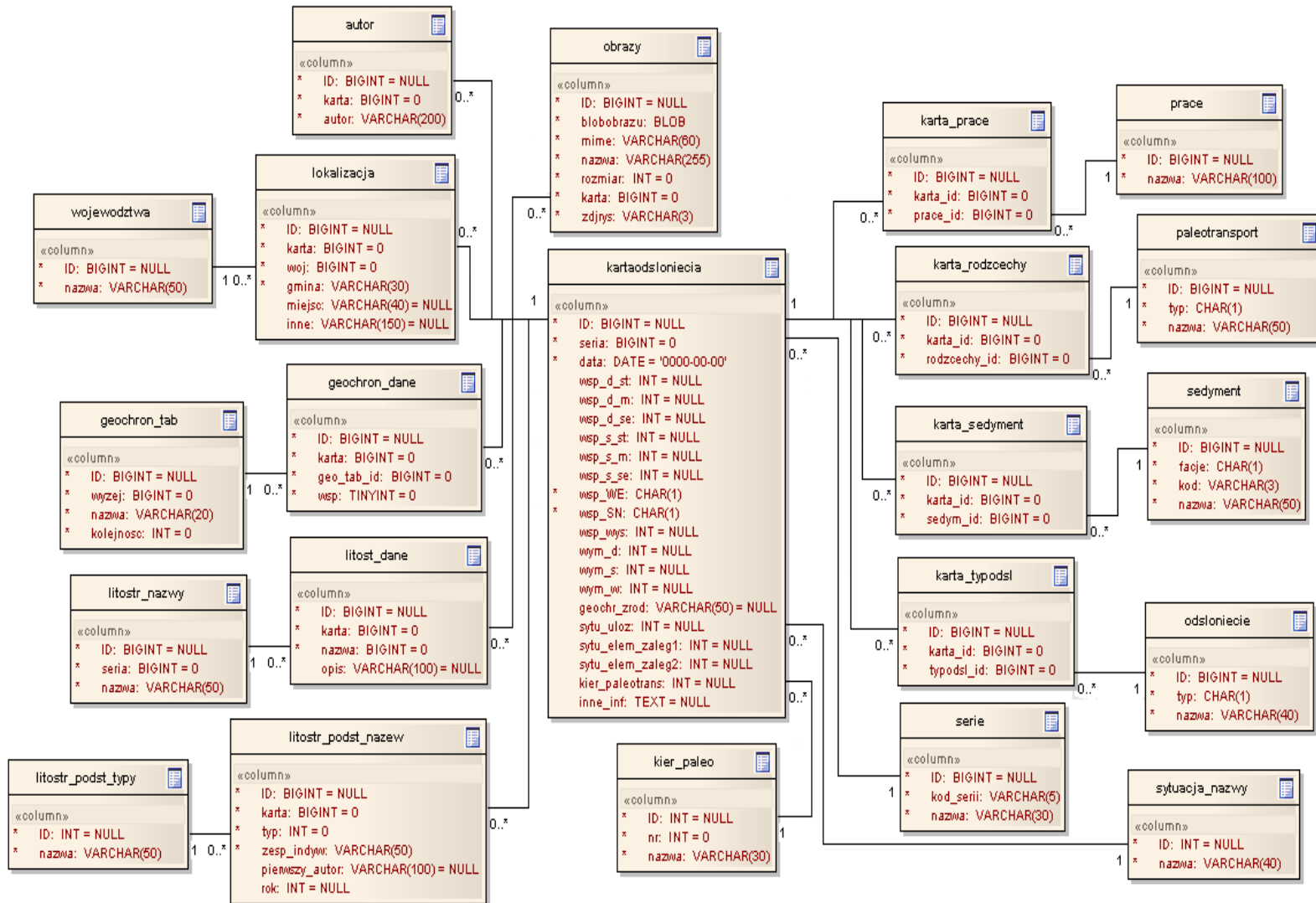
DYDAKTYKA
<i>kod</i>
<i>termin</i>
<i>rodzaj zajęć</i>
<i>przedmiot</i>
<i>czas</i>
<i>prowadzący</i>

# PRZYKŁADY



Exposition		
<b>IDExposition</b>	LiAddNotes	SeErosionDitch
ExTerrainNo	LiNameBase	SeAtherL
ExUnitName	LiBibliogNote	SeWhirlHole
ExDescDate	CHRPrecision	SeDiagonalLaye
ExNotes	CHRRange	SeFlowRound
ExCommune	CHRBase	SeRipplemarkAs
ExCity	SeC	SeAtherA
ExAuthor1	SeCS	Se_N_S
ExAuthor2	SeSC	Se_W_E
ExLongitudeDeg	SeS	Se_NE_SW
ExLongitudeMin	SeSM	Se_NW_SE
ExLongitudeSec	SeMS	Se_N
ExLatitudeDeg	SeM	Se_S
ExLatitudeMin	SeCL	Se_E
ExLatitudeSec	SeMC	Se_W
ExHeightAboveSeaLvl	SeLG	Se_NE
ExLength	SeLP	Se_SE
ExWidth	SeLS	Se_SW
ExHigh	SeDO	Se_NW
ExNaturalKind	SeSD	TeDetailedProfil
ExArtificialKind	SeML	TeFotoDoc
ExLayerArrangement	SeF	TeExposPic
ExSinkAzimuth	SeST	TeTectMeasure
ExFallAngle	SeTraceDrag	TeTest
LiUnits	SeGrainOrient	TeAther
LiAddUnits	SeTrail	TeAtherNotes

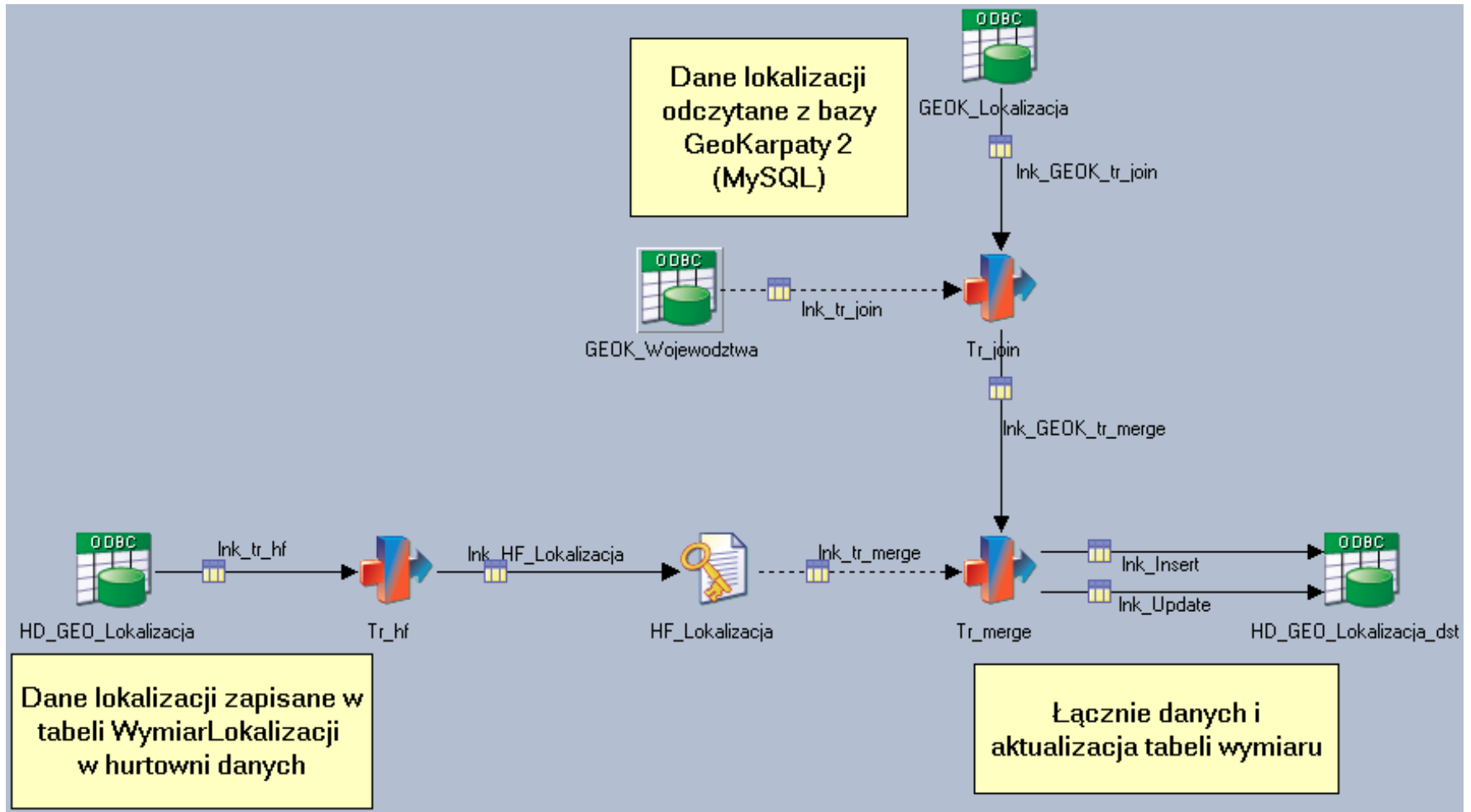
# PRZYKŁADY



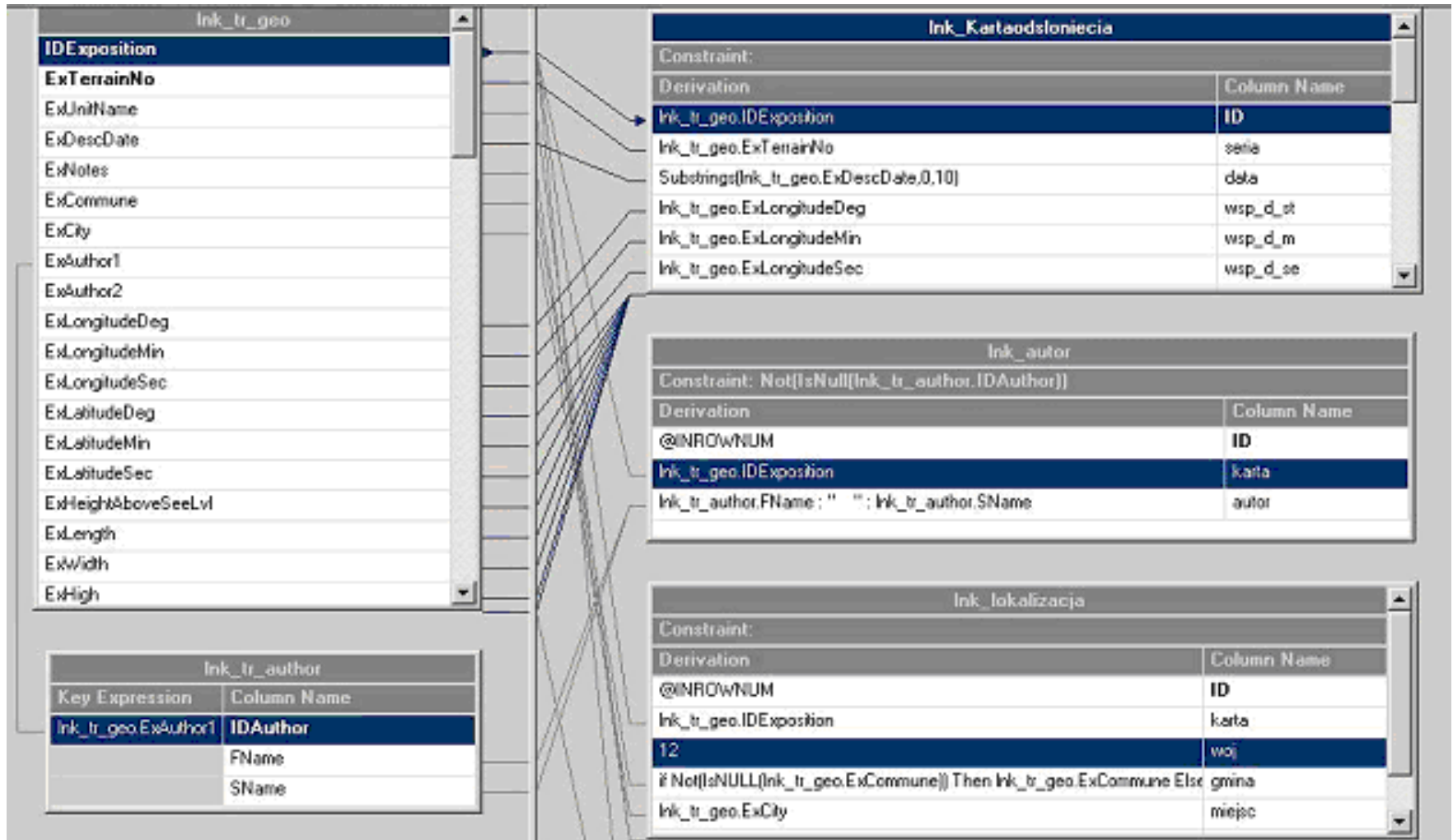
Małec O., Kyc M., Piórkowski A.: Internetowa baza danych odsłoneń GeoKarpaty II. Materiały kongresowe Pierwszego Polskiego Kongresu Geologicznego, Kraków 26–28 czerwca 2008. Polskie Towarzystwo Geologiczne, 2008.

Gajda G., Piórkowski A.: Możliwości konstrukcji hurtowni danych geologicznych. Bazy danych - rozwój metod i technologii - bezpieczeństwo, wybrane technologie i zastosowania. WKŁ, Warszawa 2008.

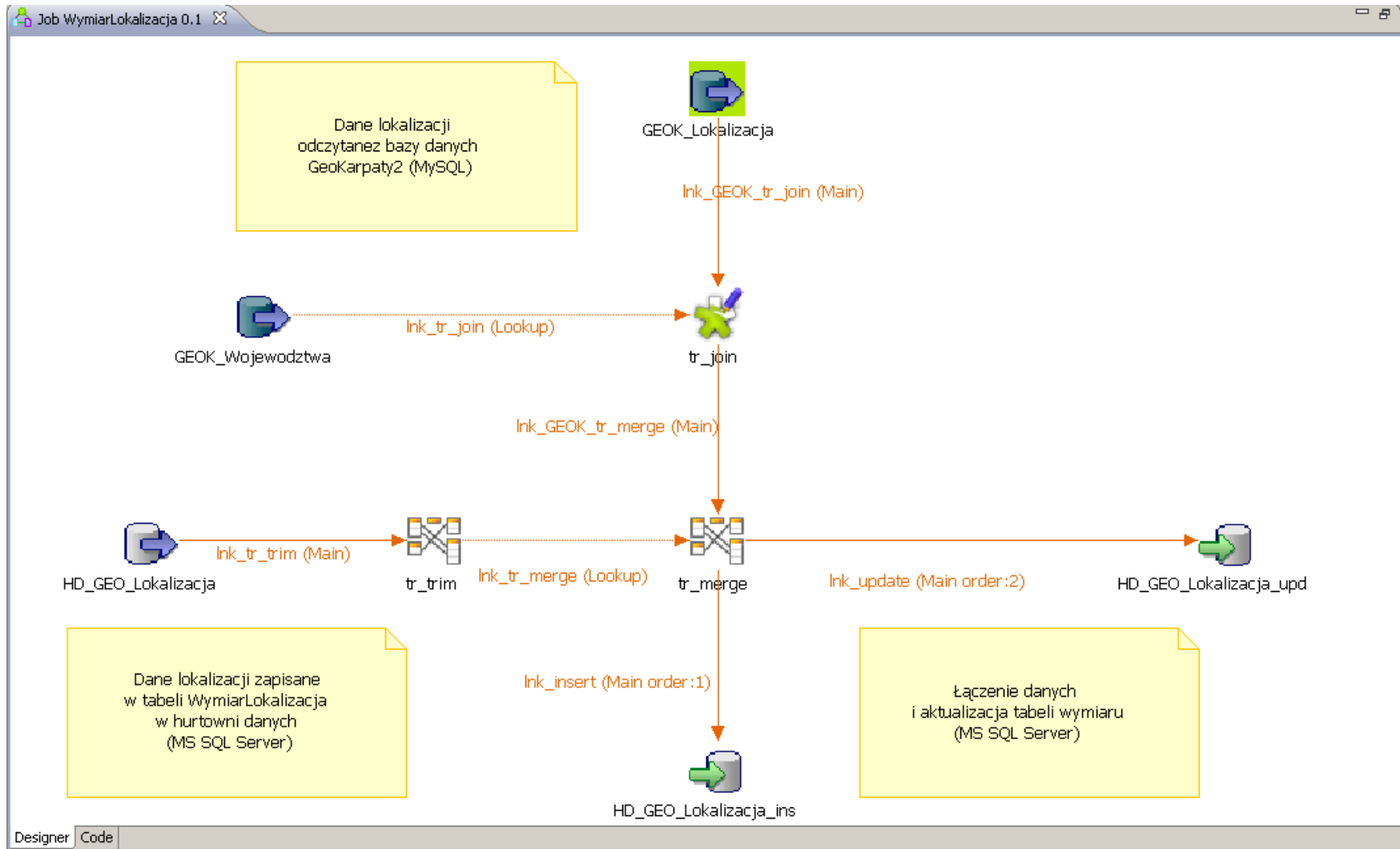
# PRZYKŁADY



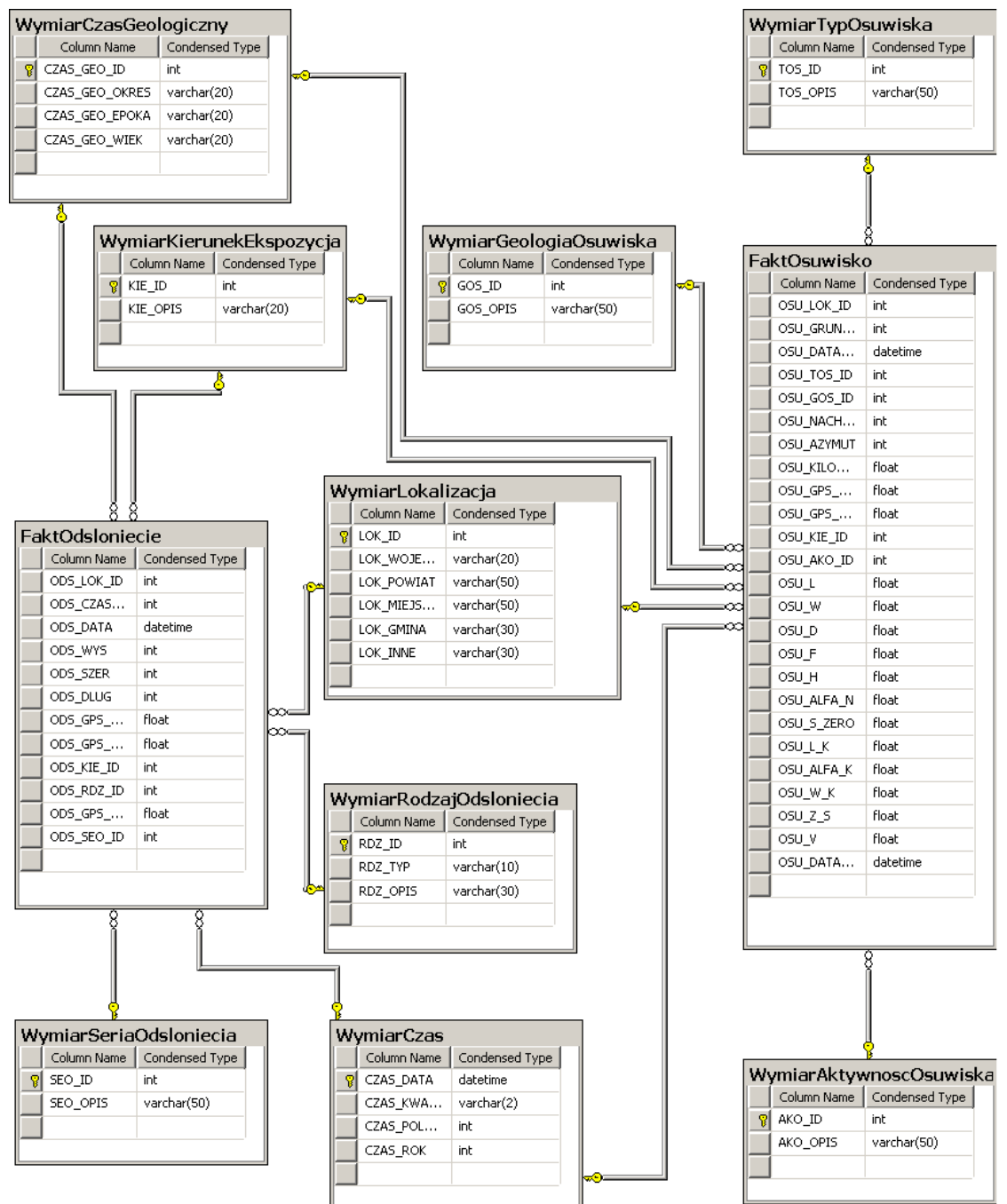
# PRZYKŁADY



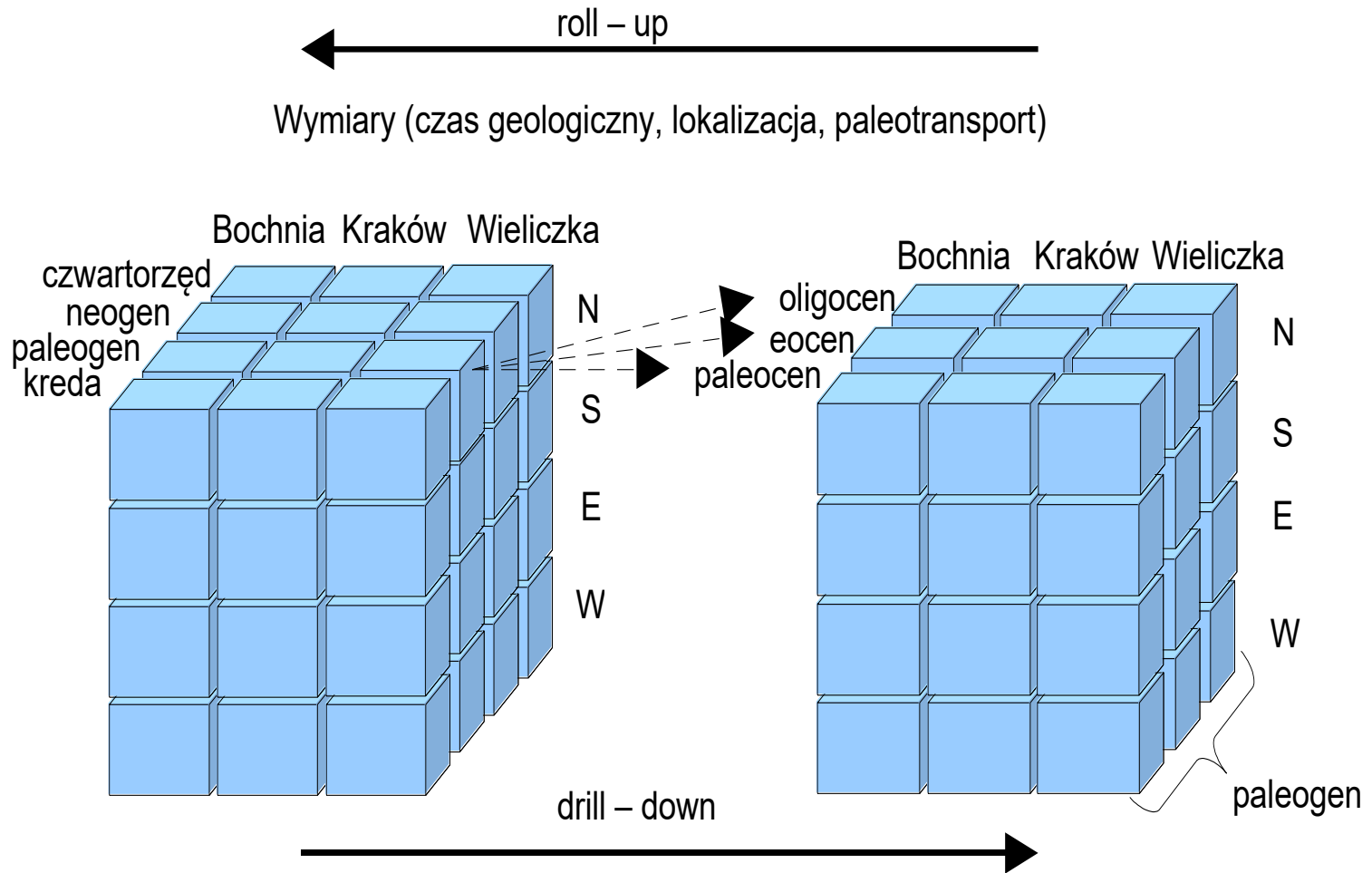
# PRZYKŁADY



# PRZYKŁADY

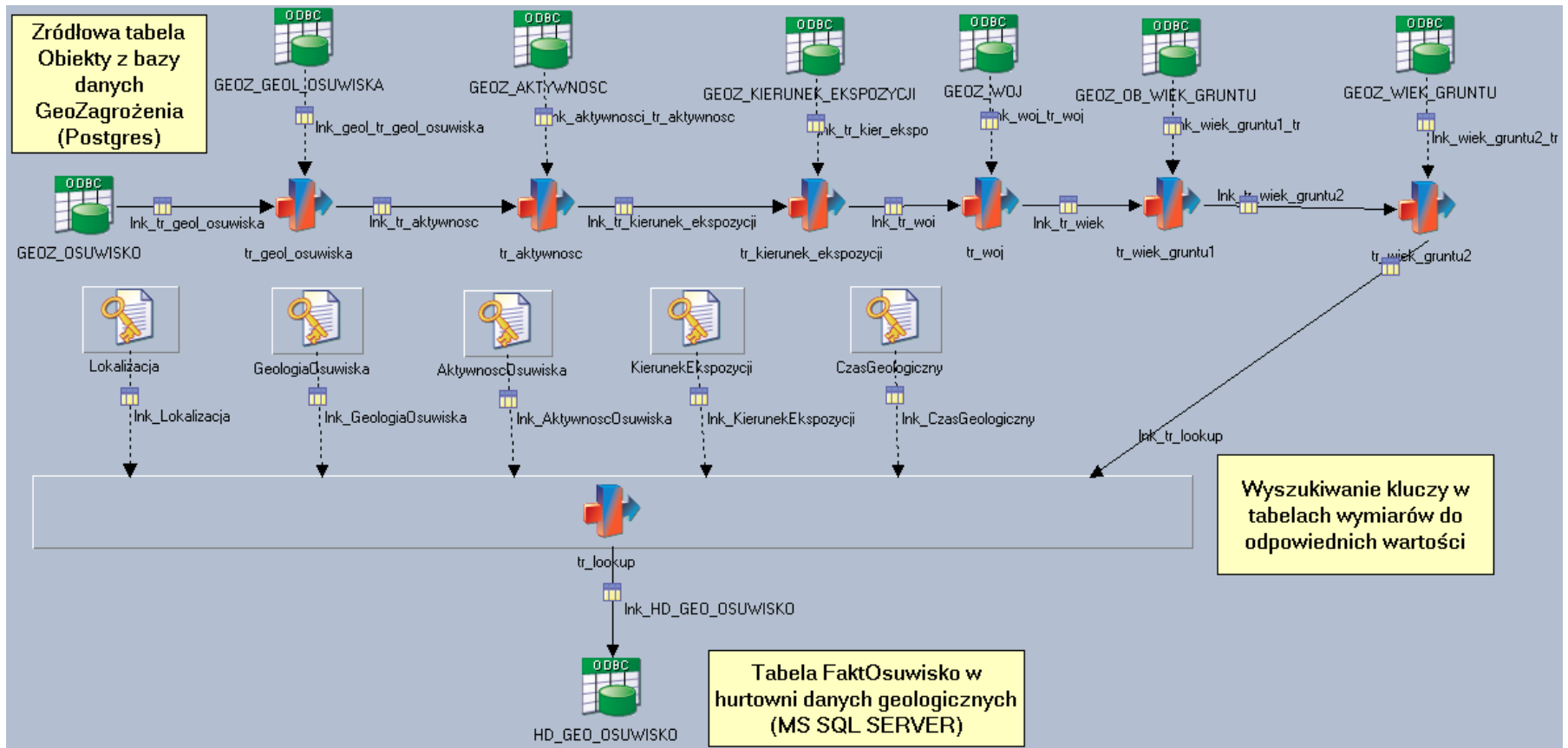


# PRZYKŁADY

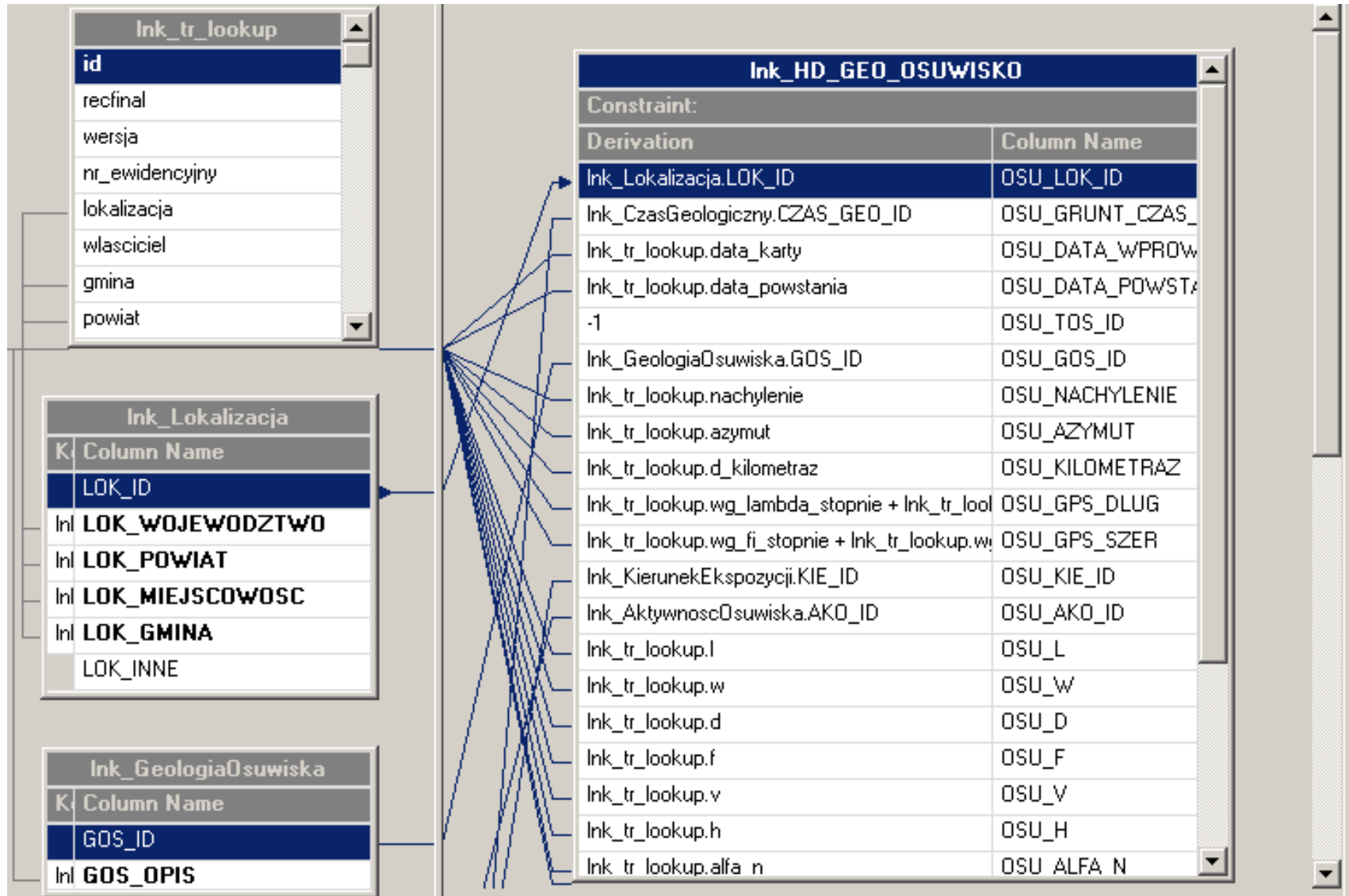


Hierarchia atrybutu WymiarCzasGeologiczny: Okres -> Epoka ->Wiek

# PRZYKŁADY



# PRZYKŁADY

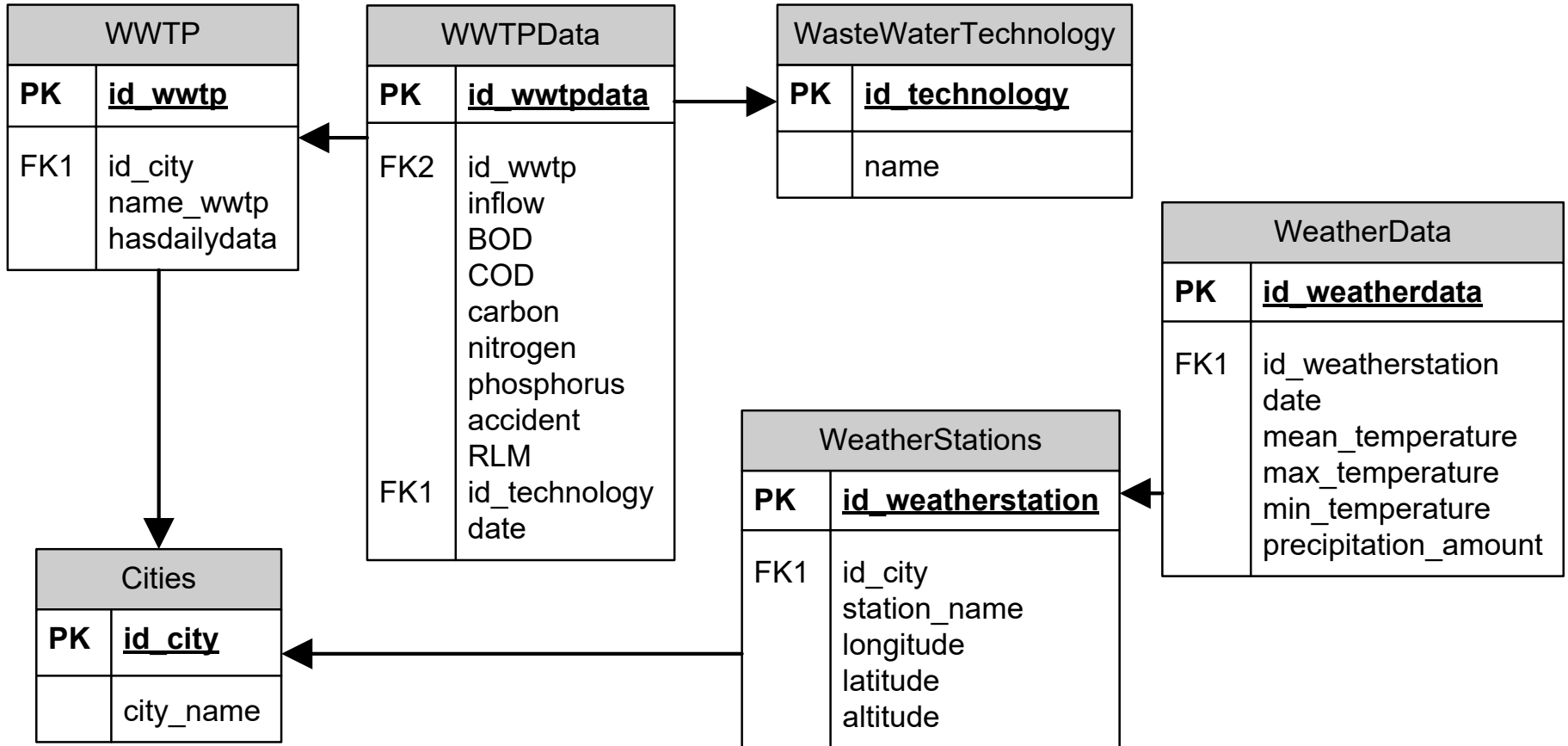


# PRZYKŁADY

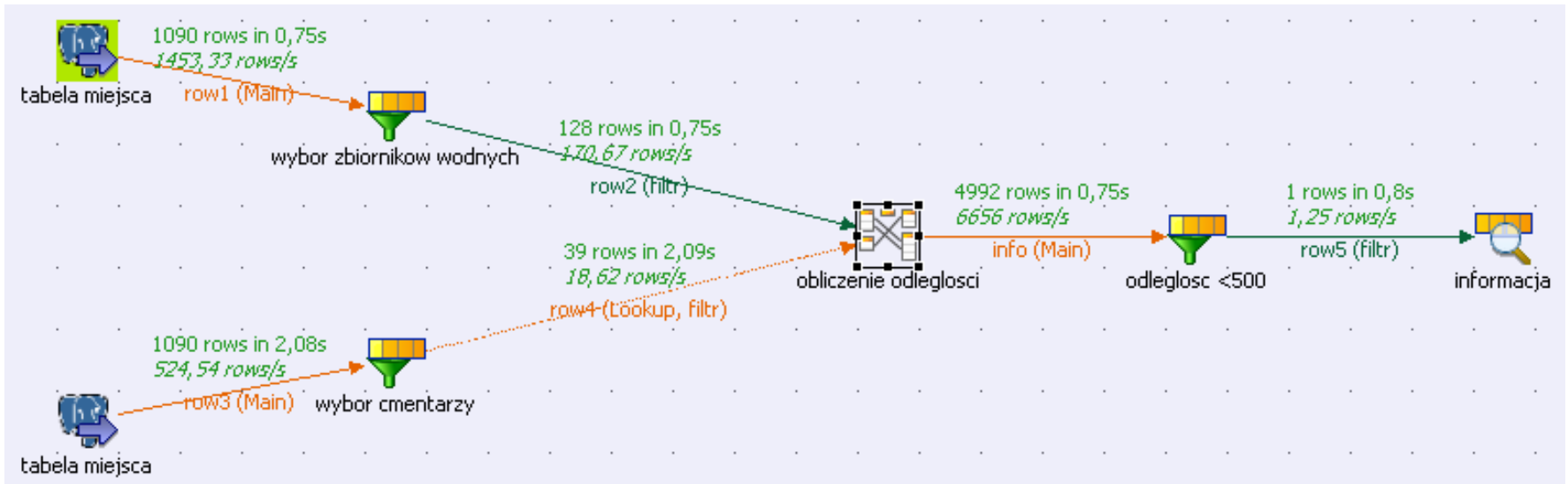


TabGeochronologiczna	
<b>PK</b>	<u><b>id_pietro</b></u>
	nazwa_pietro nazwa_epoka nazwa_okres nazwa_era nazwa_eon

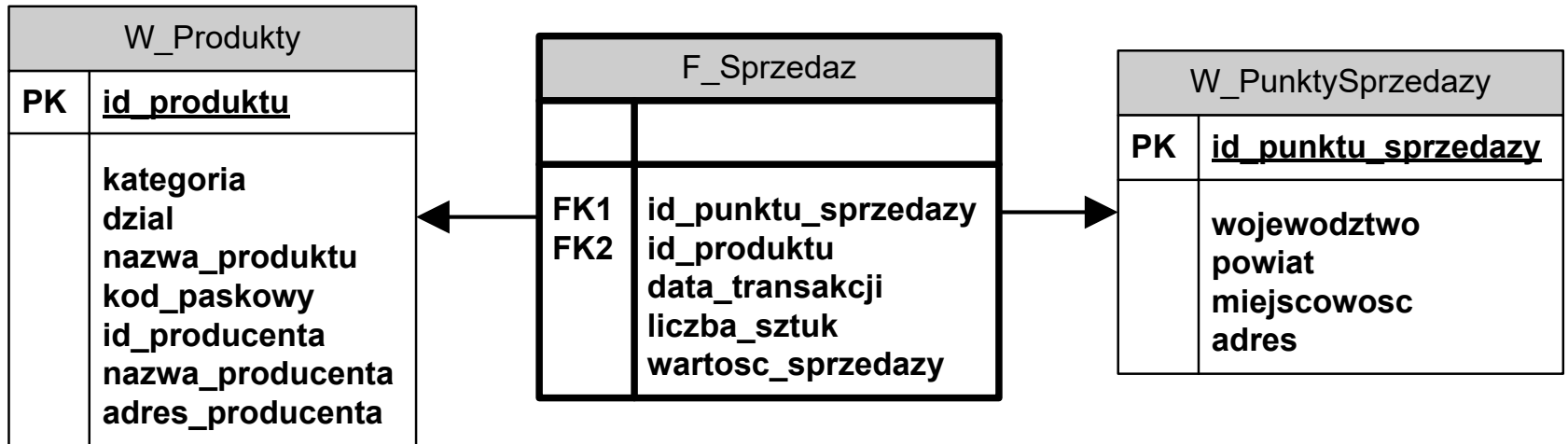
# PRZYKŁADY



# PRZYKŁADY



# PRZYKŁADY



# PRZYKŁADY

W_Producenci	
PK	<u>id_producenta</u>
	nazwa_producenta adres_producenta

W_Produkty	
PK	<u>id_produktu</u>
FK1 FK2	id_producenta id_dzialu nazwa_produktu kod_paskowy nazwa_producenta adres_producenta

W_Dzialy	
PK	<u>id_dzialu</u>
	nazwa_dzialu kategoria

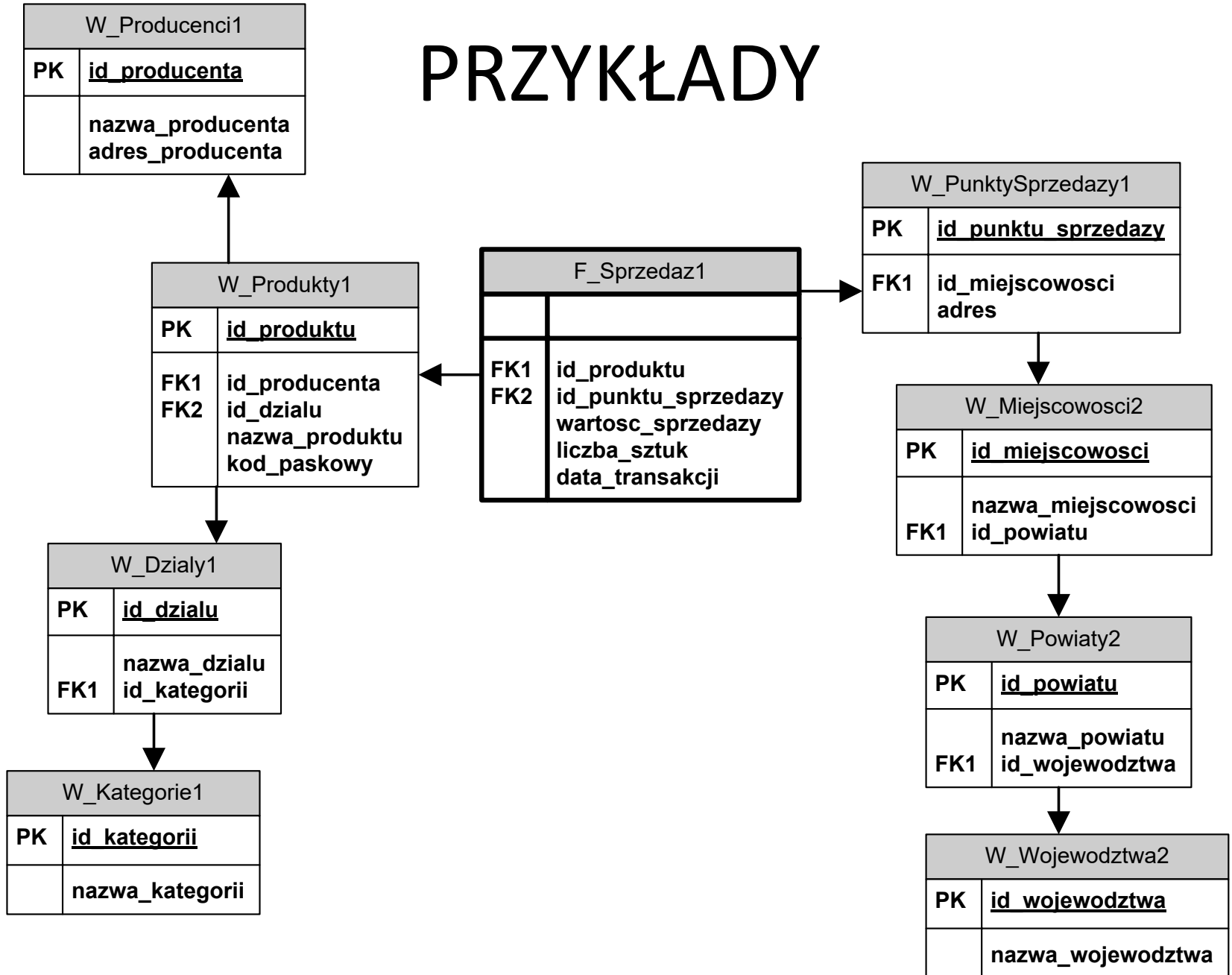
F_Sprzedaz	
FK1 FK2	id_produktu id_punktu_sprzedazy wartosc_sprzedazy liczba_sztuk data_transakcji

W_PunktySprzedazy	
PK	<u>id_punktu_sprzedazy</u>
FK1	id_miejscowosci adres

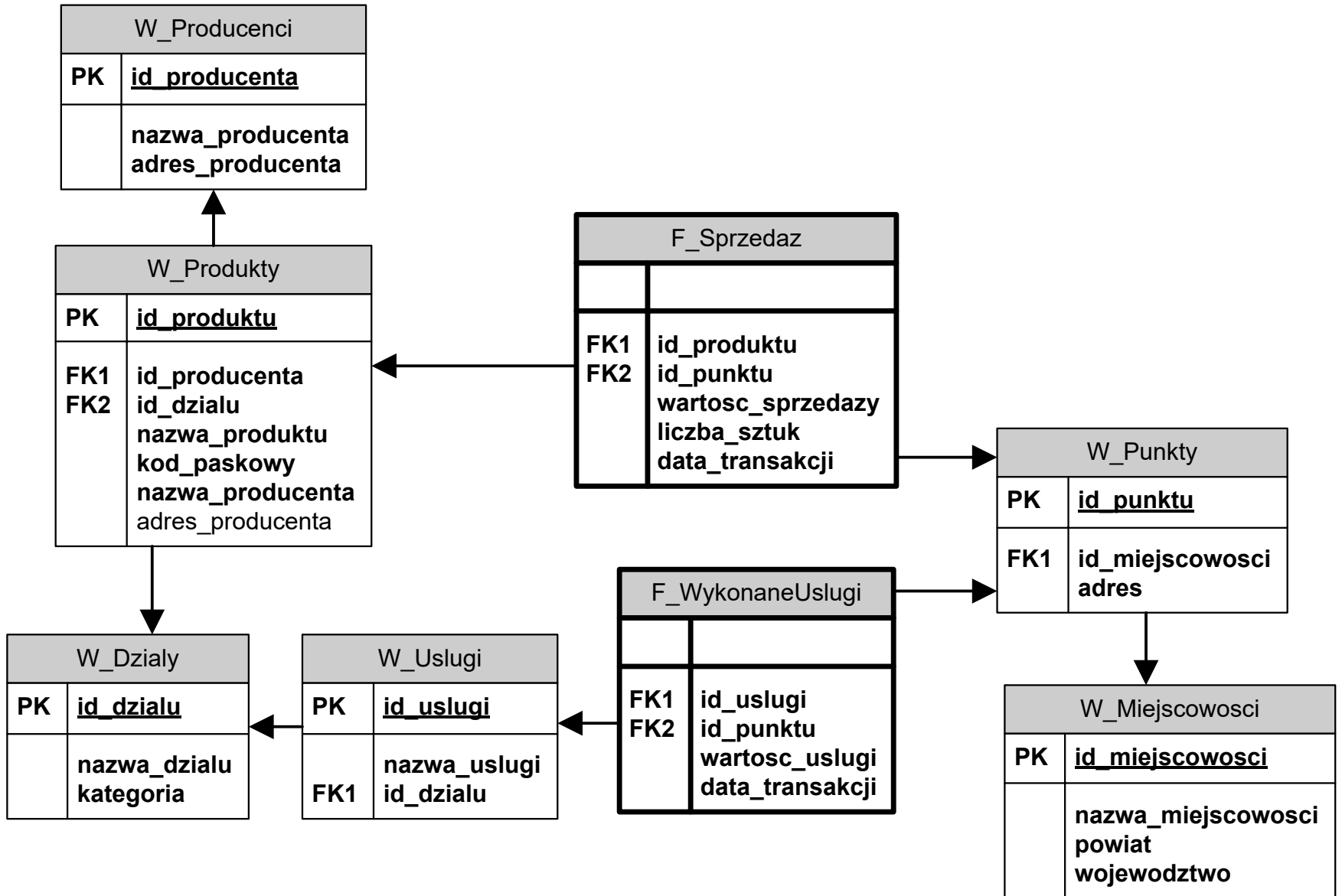
W_Miejscowosci	
PK	<u>id_miejscowosci</u>
	nazwa_miejscowosci powiat wojewodztwo



# PRZYKŁADY



# PRZYKŁADY



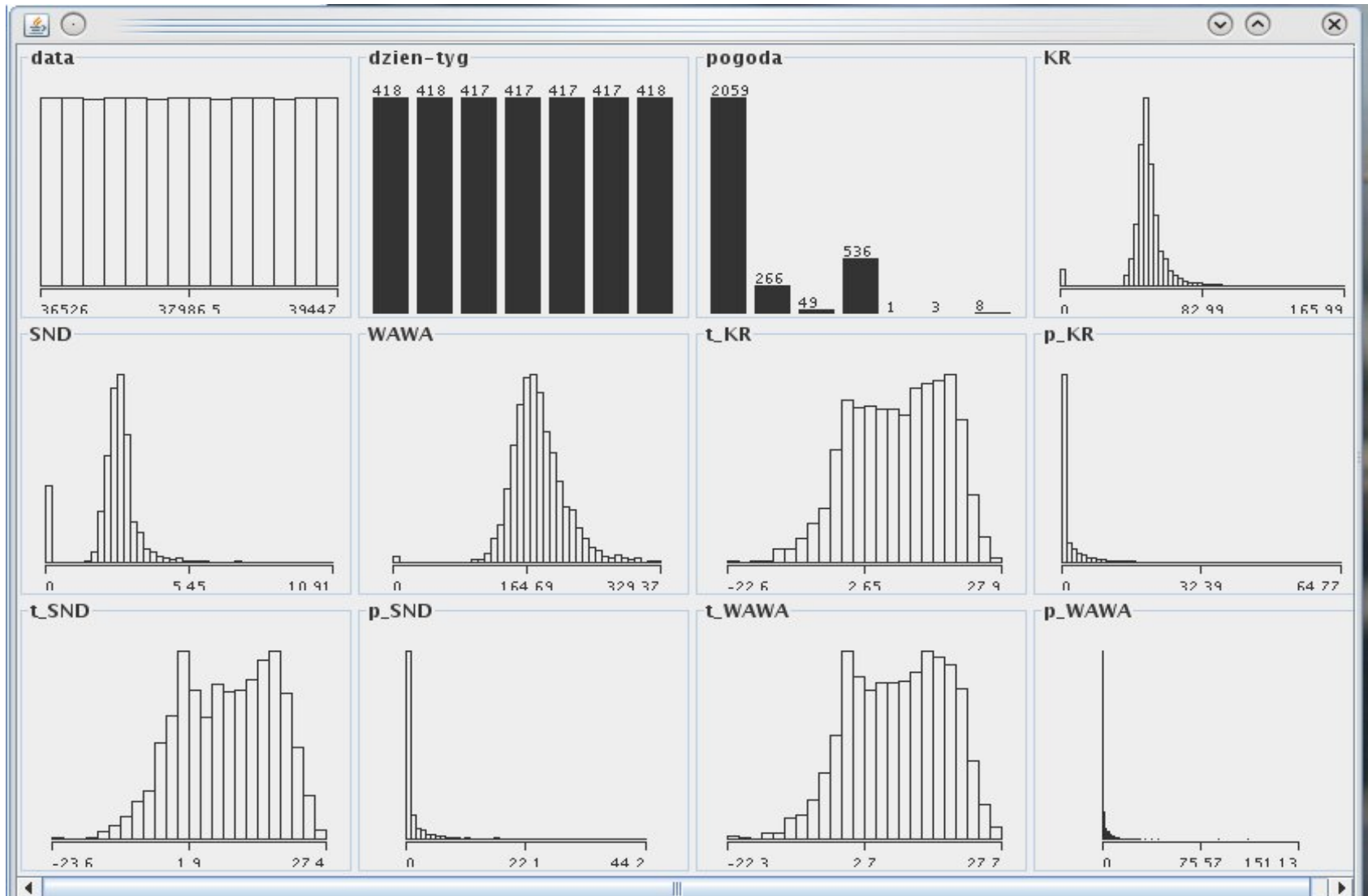
# PRZYKŁADY

The screenshot shows the Explorer software interface with the following components:

- Navigation tabs:** Preprocess, Classify, Cluster, Associate, Select attributes, Visualize.
- Buttons:** Open file..., Open URL..., Open DB..., Generate..., Undo, Edit..., Save...
- Filter:** Choose None [Apply]
- Current relation:** Relation: wwtp, Instances: 2922, Attributes: 12, Sum of weights: 2922
- Attributes:** All, None, Invert, Pattern. A list of 12 attributes with checkboxes, where the last 8 are checked.
- Selected attribute:** Name: p\_WAWA, Missing: 0 (0%), Distinct: 84, Type: Numeric, Unique: 24 (1%).
- Statistics table:**

Statistic	Value
Minimum	0
Maximum	151.13
Mean	1.631
StdDev	5.122
- Class:** p\_WAWA (Num) [Visualize All]
- Histogram:** A histogram showing the distribution of p\_WAWA values, with a peak near 0 and a long tail extending to 151.13.
- Status:** OK [Log] x 0

# PRZYKŁADY



# PRZYKŁADY

STATISTICA - data mining

Plik Edycja Widok Uruchom Wezły Statystyka Data Mining Wykresy Narzędzia Okno Pomoc

data mining

Źródła danych    Przygotowanie, czyszczenie, transformacja danych    Analiza danych (modelowanie, klasyfikacja, prognozowanie)    Raporty

	1	2	3	4	5	6	7	8	9	10	11	
	Data	Data	Pogoda	Kraków - Kujawy	Sandomierz	Warszawa - Czajka	T Kr	PP Kr	T Snd	PP Snd	T Wawa	PF
1	36526	7 s		43	2,587	167,215	-8,6	0	-3,8	0	-1,1	
2	36527	1 s		43,6	2,621	171,455	-5,9	0	-8	0	-5,7	
3	36528	2 s		40,8	2,83	187,2	1,2	0,76	-0,3	0,76	1,2	
4	36529	3 od		58	2,9	223,55	2,3	0,76	0,8	1,02	2,3	
5	36530	4 od		58,2	2,95	225,34	2,8	2,03	1,2	2,54	2,7	
6	36531	5 od		51,3	3,05	193,5	0,4	0	0,7	0,25	0,8	
7	36532	6 s		46,7	2,816	193,85	-1,7	0	-0,8	0	0,4	
8	36533	7 s		54,2	3,05	187,99	0,3	0	-0,2	0	1,8	
9	36534	1 s		50,2	2,86	179,8	0,9	0	-0,3	0	1,2	
10	36535	2 s		51,1	3,001	183,91	0,9	0,76	0,6	0	1,6	
11	36536	3 s		50,8	2,99	190,37	0,2	0,76	0,2	0	0,6	

Gotowe



# 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 + zbior atrybutow

`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 $2^{31} - 1$ -2,147,483,648 - 2,147,483,647	Liczby całkowite
BIGINT, SMALLINT, TINYINT	8,2,1		
DECIMAL (precyzja, skala) NUMERIC*	~ precyzji	max liczba cyfr ~SZBD	Liczby dziesiętne, np. waluta: 100.15
FLOAT,	Precyzja 1-24 / 25-53	-1.79E+308 :-2.23E-308, 2.23E-308 to 1.79E+308	Liczby zmiennoprzecinkowe
REAL	4	- 3.40E + 38 to -1.18E – 38 1.18E - 38 to 3.40E + 38.	

REAL w SQL odpowiada typowi float w rozumieniu procesora Intel (4 bajty), a FLOAT – double (precision)

NUMERIC ma stałą precyzję, DECIMAL – może mieć większą, niż potrzebuje projektant (związane z wydajnością szbd, w części szbd te typy są równoważne) - zależy także od S.Z.B.D.

Standard SQL

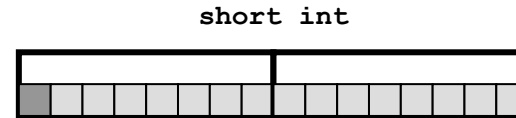
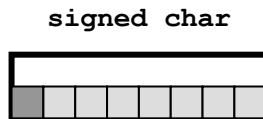
# Typy danych

TYP	[B]	ZAKRES	OPIS
CHAR CHAR(20)	1 20	Jeden znak ASCII Tablica 20 znaków ASCII	Jeden znak Dokładnie 20 znaków, dopełniane spacjami
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 CLOB	Np. do 2GB	Blok binarny Blok tekstowy	Np. na zdjęcia Np. na strony

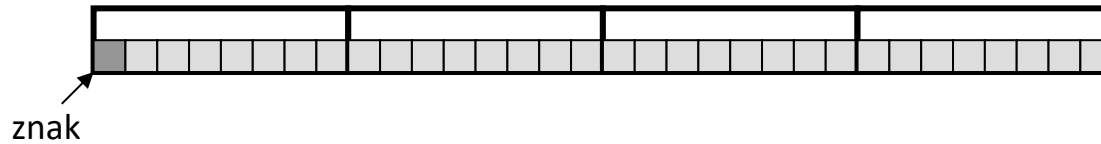
# Typy danych

całkowitoliczbowe

ze znakiem

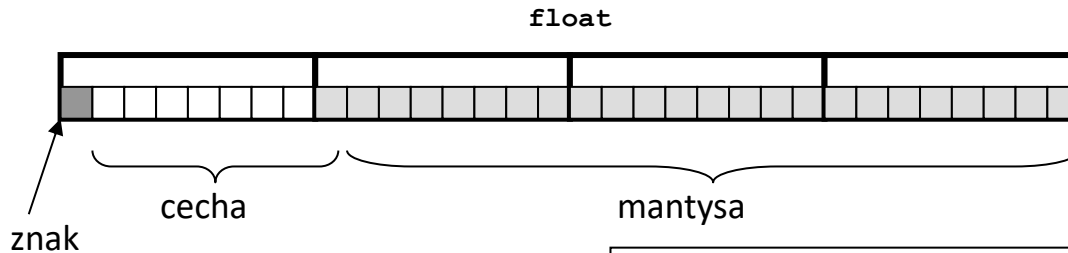


long int



zmiennoprzecinkowe

ze znakiem



$$\text{liczba} = \text{mantysa} * 2^{\text{cecha}}$$

dziesiętne  
(stała precyzja)



# Liczby zmiennoprzecinkowe

```
CREATE TABLE liczby(i INT, b BIGINT, f FLOAT, r REAL);
```

```
INSERT INTO liczby VALUES
```

```
( 1234567891, 1234567891234567891, 123456789123456789, 123456789123456789 );
```

```
SELECT * FROM liczby;
```

i	b	f	r
1234567891	1234567891234567891	1.23456789123457e+017	1.23457e+017

```
DELETE FROM liczby;
```

```
INSERT INTO liczby VALUES( 0, 0, 100000, 100000 );
```

```
INSERT INTO liczby VALUES( 0, 0, 1000000, 1000000 );
```

```
SELECT * FROM liczby;
```

i	b	f	r
0	0	100000	100000
0	0	1000000	1e+006

```
UPDATE liczby SET f = f + 0.007;
```

```
UPDATE liczby SET r = r + 0.007;
```

```
SELECT * FROM liczby;
```

i	b	f	r
0	0	100000.007	100000
0	0	1000000.007	1e+006

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

# Projektowanie tabeli

The image shows two overlapping windows from LibreOffice Base. The top window, titled 'Przykladowa.odt : Indeksy - LibreOffice Base: Projekt tabeli', displays the design view for a table named 'Indeksy'. The table has five fields: 'nr\_indeksu' (Integer [INTEGER]), 'imie' (Text [VARCHAR]), 'nazwisko' (Text [VARCHAR]), 'pesel' (Text [VARCHAR]), and 'kierunek' (Text [VARCHAR]). Below the table, the 'AutoValue' property is set to 'Yes', 'Length' is 10, and 'Format example' is 0.

The bottom window, titled 'Przykladowa.odt : Studenci - LibreOffice Base: Projekt tabeli', displays the design view for a table named 'Studenci'. The table has nine fields: 'pesel' (Text [VARCHAR]), 'imie' (Text [VARCHAR]), 'nazwisko' (Text [VARCHAR]), 'data\_ur' (Date [DATE]), 'kraj' (Text [VARCHAR]), 'miejscowosc' (Text [VARCHAR]), 'ulica' (Text [VARCHAR]), 'nr\_domu' (Text [VARCHAR]), 'nr\_mieszkania' (Text [VARCHAR]), and 'kod\_pocztowy' (Text [VARCHAR]). The 'pesel' field has a tooltip that reads 'Text, INT ma za maly zakres...'. The 'AutoValue' property is set to 'Yes', 'Length' is 10, and 'Format example' is 0.


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

Field Name	Field Type	Notes
pesel	Text [ VARCHAR ]	Text, INT ma za maly zakres...
imie	Text [ VARCHAR ]	
nazwisko	Text [ VARCHAR ]	
data_ur	Date [ DATE ]	
kraj	Text [ VARCHAR ]	
miejscowosc	Text [ VARCHAR ]	
ulica	Text [ VARCHAR ]	
nr_domu	Text [ VARCHAR ]	
nr_mieszkania	Text [ VARCHAR ]	
kod_pocztowy	Text [ VARCHAR ]	

# Wpisywanie wartości


Studenci - Przykładowa - LibreOffice Base: Widok danych tabeli

File Edit View Insert Tools Window Help



	imie	nazwisko	pesel	nr_indeksu
	Adam	Nowak	12345	100
	Ewa	Kowalska	12346	101
	Anna	Nowaczyk	12346	101

LibreOffice Base

 Error inserting the new record

Violation of unique constraint SYS\_PK\_47: duplicate value(s) for column(s) "pesel" in statement [INSERT INTO "Studenci" ("imie","nazwisko","nr\_indeksu","pesel") VALUES (?, ?, ?, ?)]

OK More



# 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)

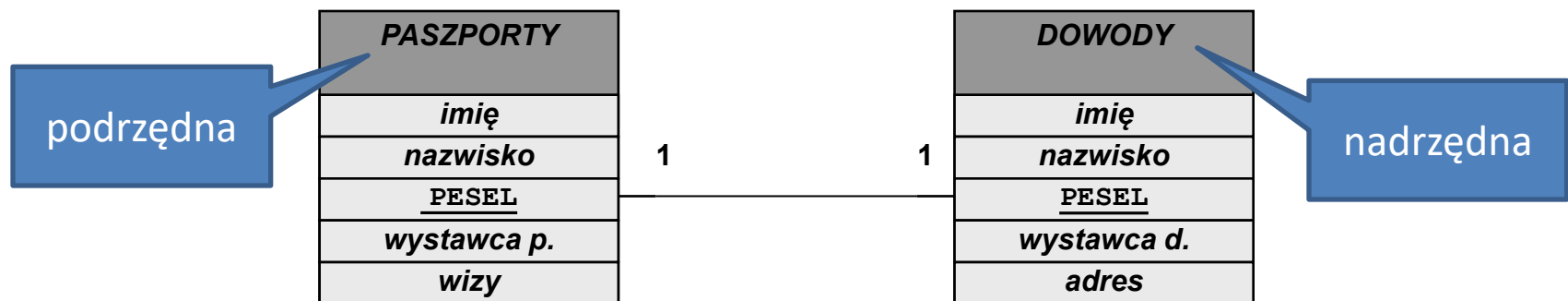
Dowody\_osobiste( imię, nazwisko, PESEL , osoba\_wyst\_dowod, adres\_zamieszkania).

Założenia:

a)każdy musi mieć dowód osobisty, ale tylko wybrani mają paszport (nie odpowiada to sytuacji np. studentów obcokrajowców)

b) można mieć tylko jeden dowód osobisty i jeden paszport,

c) wniosek: paszport można wyrobić tylko na **istniejący** dowód osobisty



# 1:1 – jeden do jednego

paszporty.odt : Dowody - LibreOffice Base: Projekt

File Edit View Tools Window Help

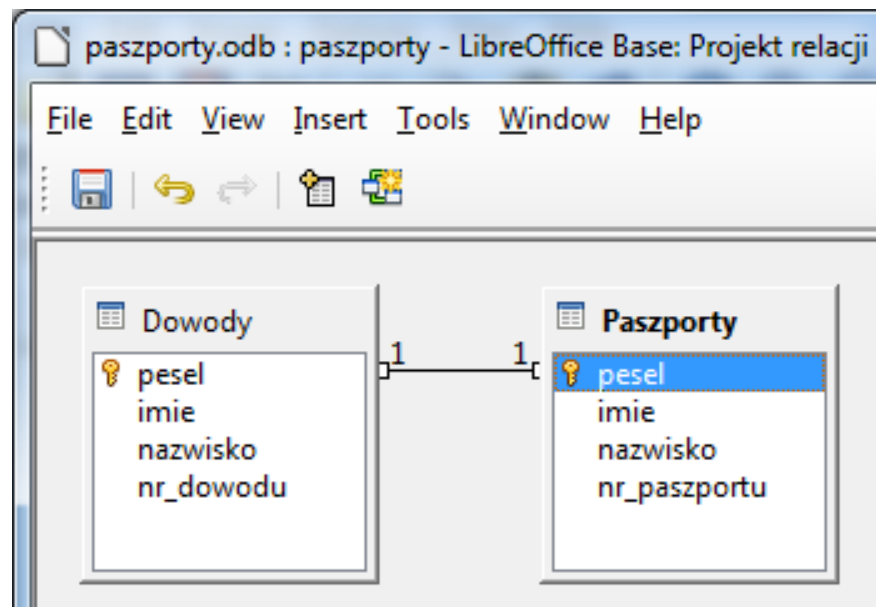
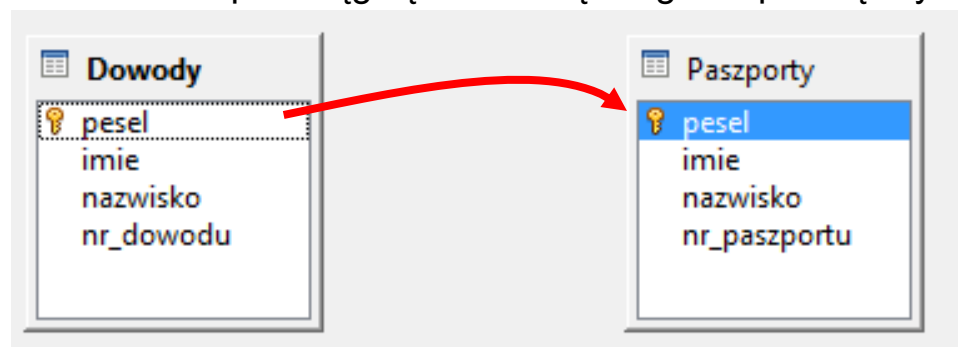
	Field Name	Field Type
🔑	pesel	BigInt [ BIGINT ]
	imie	Text [ VARCHAR ]
	nazwisko	Text [ VARCHAR ]
	nr_dowodu	Text [ VARCHAR ]

paszporty.odt : Paszporty - LibreOffice Base: Projekt

File Edit View Tools Window Help

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


LibreOffice: przeciągnięcie nadrzędnego na podrzędny



# 1:1 – jeden do jednego

Dowody - paszporty - LibreOffice Base: Widok danych


File Edit View Insert Tools Window Help



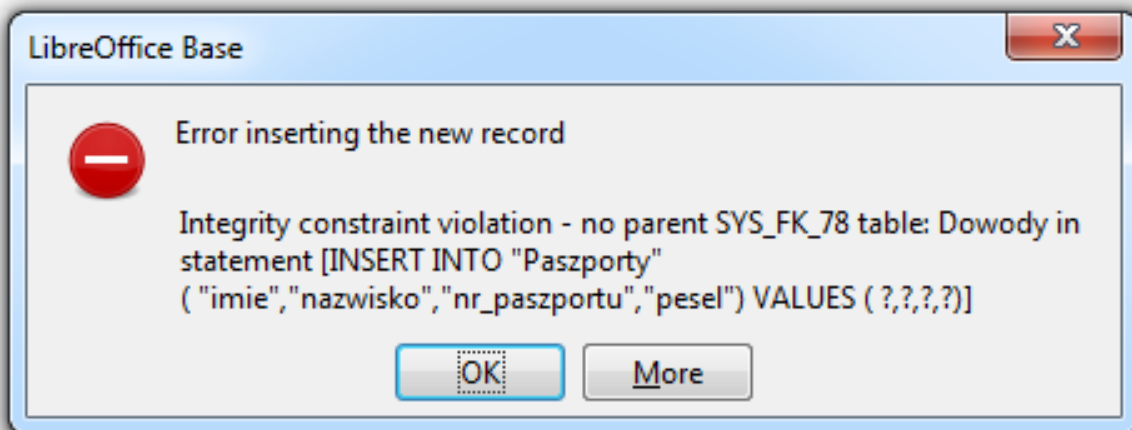
	pesel	imie	nazwisko	nr_dowodu
	1000	Adam	Nowak	1
	1001	Ewa	Nowak	2

Paszporty - paszporty - LibreOffice Base: Widok danych tabeli

File Edit View Insert Tools Window Help



	pesel	imie	nazwisko	nr_paszportu
	1000	Andrzej	Nowy	111
	1002	Ewa	Inna	1233



# 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:

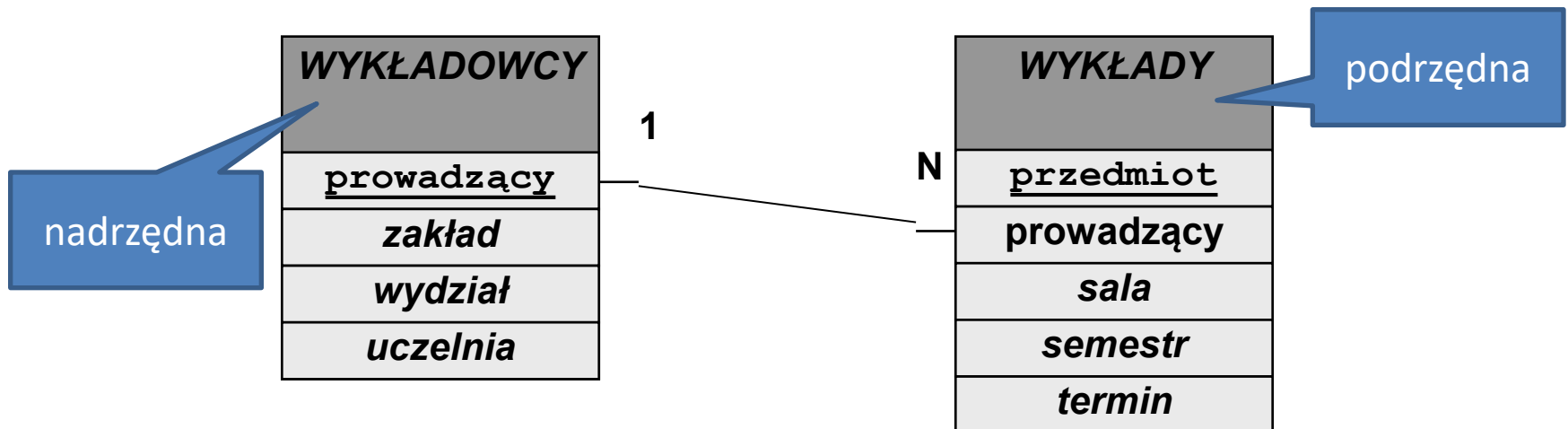
wykładowcy: prowadzący, zakład, wydział, uczelnia

wykłady: przedmiot, prowadzący, sala, semestr, termin

Założenia:

a) każdy wykład musi być poprowadzony przez jednego, istniejącego wykładowcę

b) każdy wykładowca może prowadzić 0 – N wykładów



# 1:N – jeden do wielu

wyklady.odb : Wykladowcy - LibreOffice

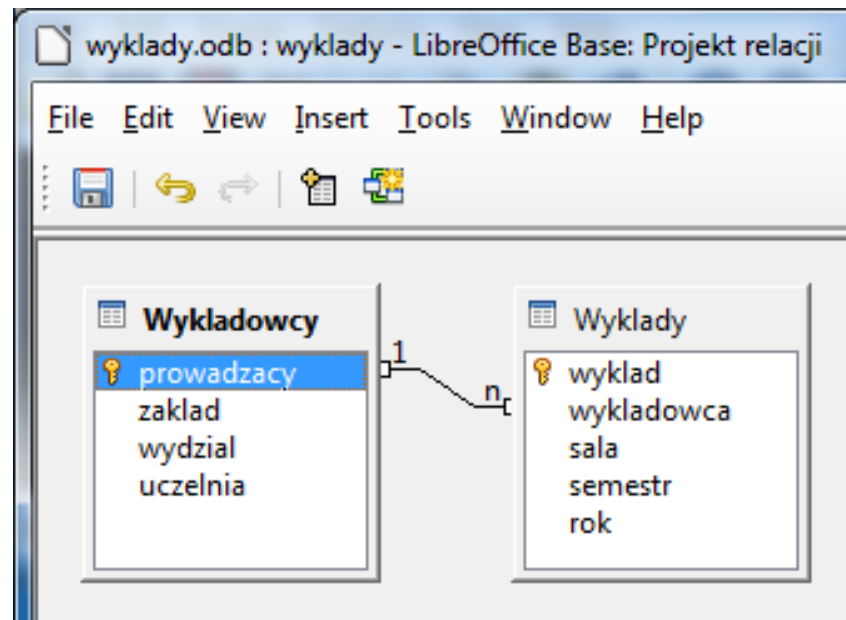
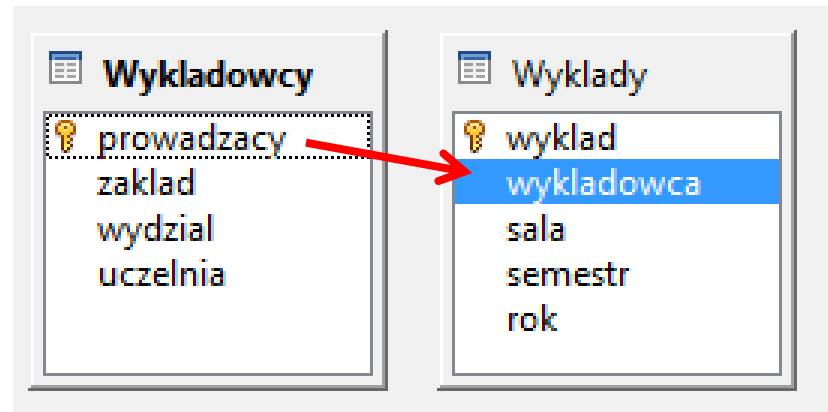
File Edit View Tools Window Help

	Field Name	Field Type
🔑	prowadzacy	Text [ VARCHAR ]
	zaklad	Text [ VARCHAR ]
	wydzial	Text [ VARCHAR ]
	uczelnia	Text [ VARCHAR ]

wyklady.odb : Wyklady - LibreOffice B

File Edit View Tools Window Help

	Field Name	Field Type
🔑	wyklad	Text [ VARCHAR ]
	wykladowca	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

The screenshot shows two database windows in LibreOffice Base. The top window, titled 'Wykladowcy - wykłady - LibreOffice Base: Widok danych', displays a table with the following data:

	prowadzacy	zaklad	wydzial	uczelnia
	Adam Nowak	KGIS	WGGIOS	AGH
	Ewa Kowalczyk	KGIS	WGGIOS	AGH

The bottom window, titled 'wyklady - LibreOffice Base: Widok danych tabeli', displays a table with the following data:

wyklad	wykladowca	sala	semestr	rok
Bazy Danych	Adam Nowak	1	1	1
Systemy Op.	Ewa Kowalczyk	2	2	2
JFiA	Adam Nowak	3	3	3
KiK	Adam Nowak	4	4	4
ZPI	Ewa Nowak	5	5	5

An error dialog box is shown in the foreground with the following text:

LibreOffice Base

Error inserting the new record

Integrity constraint violation - no parent SYS\_FK\_57 table:  
Wykladowcy in statement [INSERT INTO "Wyklady"  
("rok","sala","semestr","wyklad","wykladowca") VALUES (?,?,?,?)]

Buttons: OK, More

# M:N – wiele do wielu

W obrębie klucza danej wartości w jednej tabeli odpowiada podzbiór wartości w drugiej tabeli – i odwrotnie.

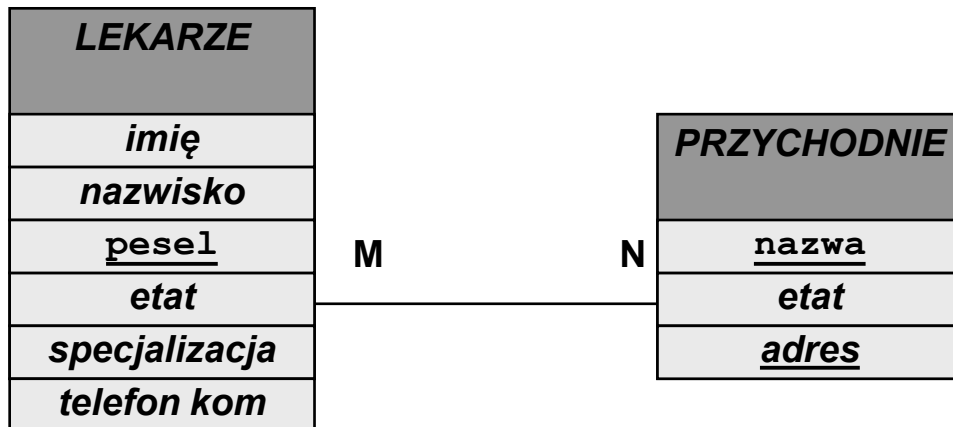
Takie zależności nie są możliwe do bezpośredniej implementacji w r.b.d i wymagana jest dekompozycja M:N do dwóch związków M:1 i 1:N

przykład:

lekarze: imię, nazwisko, specjalizacja, etat

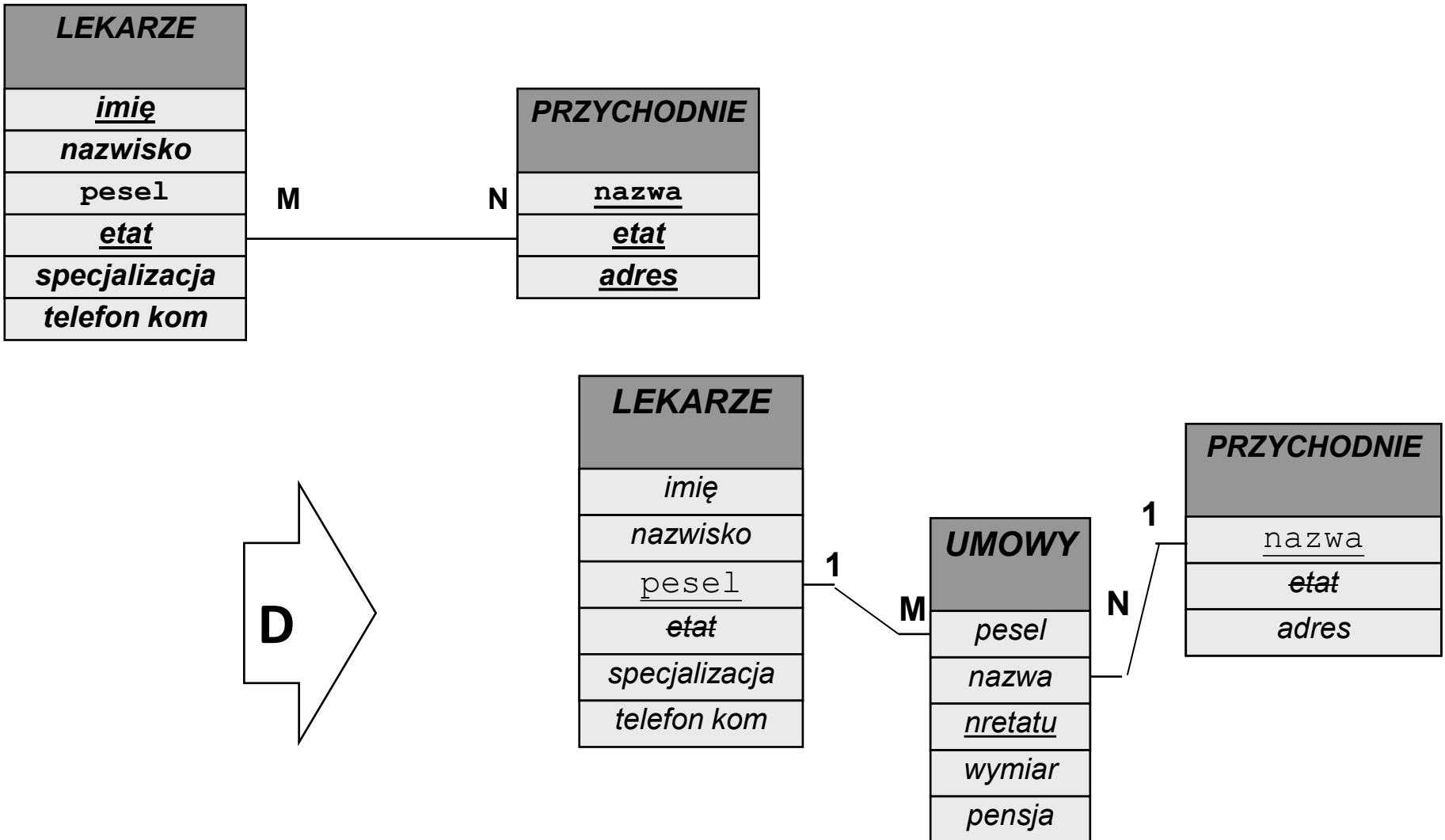
przychodnie: nazwa, adres, etat

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

Wybrane województwo: **małopolskie**

Miejscowość: **Kraków**

Wybrany typ szkoły: **Szkoła wyższa (zweryfikowane)**

<< poprzednie 1 2 następne >>

## Nazwa szkoły

Uniwersytet Jagielloński

Uniwersytet Jagielloński - Wydział Polonistyki

Uniwersytet Papieski Jana Pawła II

Uniwersytet Papieski Jana Pawła II

Uniwersytet Papieski Jana Pawła II

Uniwersytet Pedagogiczny im. Komisji Edukacji Narodowej (dawniej Akademia Pedagogiki)

Uniwersytet Rolniczy

Wydział Hodowli i Biologii Zwierząt Akademii Rolniczej

Wydział Pedagogiczny - Uniwersytet Pedagogiczny

Wydział Prawa i Administracji - Studia Podyplomowe-Uniwersytet Jagielloński

## WSZYSTKIE KIERUNKI

<< poprzednie 1 2 3 4 5 ... 166 następne >>

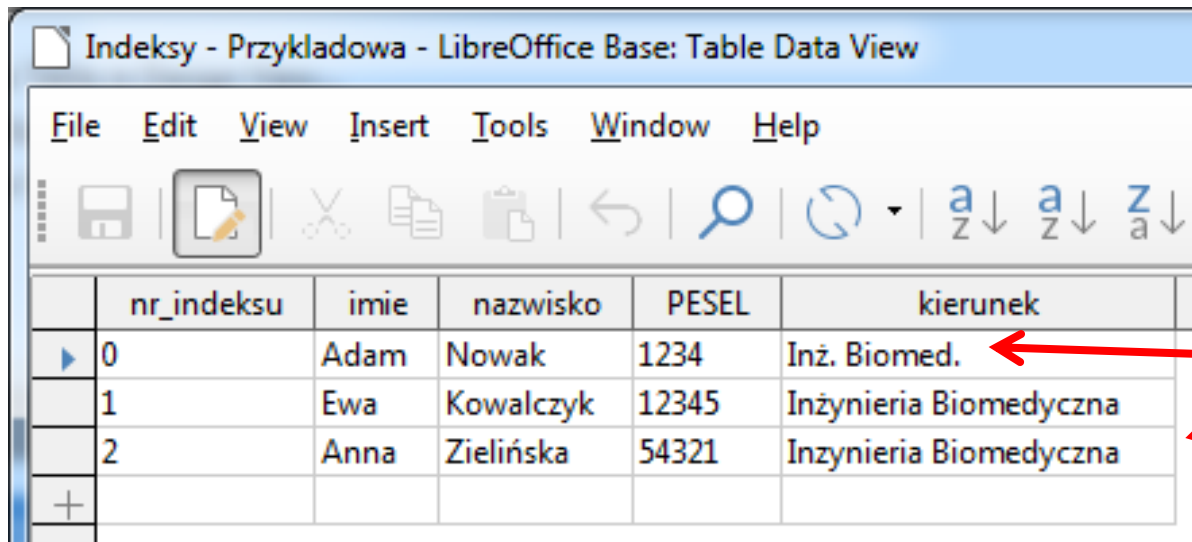
21. Administracja	dzienne lic, 3-letnie	W. Prawa i Administracji
22. Administracja	uzupełniające zaoczne, 2-letnie	WPIA
23. Administracja	zaoczne, 5-letnie	WPIA
24. Administracja	zaoczne, 2-letnie	Prawa i Administracji
25. Administracja	niestacjonarne, mgr, 5-letnie	Wydział Prawa i Administracji
26. Administracja	Niestacjonarne-uzupełniające, 2-letnie	Prawa i Administracji
27. Administracja	zaoczne, 3-letnie	Prawa i Administracji
28. Administracja	zaoczne mgr, 2-letnie	Prawo i Administracja
29. Administracja	zaoczne, 5-letnie	Prawo i Administracja
30. Administracja	niestacjonarne licencjackie, 3-letnie	Prawa i Administracji
31. Administracja	zaoczne, 5-letnie	Prawa i Administracji
32. administracja	zaoczne, 5-letnie	Prawo i administracja
33. Administracja	licencjackie, 3-letnie	Prawa i Administracji
34. Administracja	dzienne mgr, 4-letnie	Prawo i Administracja
35. Administracja	dzienne lic, 3-letnie	Prawa i Administracji
36. Administracja	jednolite magisterskie, 5-letnie	Prawa i Administracji
37. Administracja	zaoczne, 5-letnie	Wydział Prawa i Administracji
38. Administracja	niestacjonarne lic, 3-letnie	Prawa i Administracji
39. Administracja	SUM, 2-letnie	Prawa i Administracji
40. Administracja	zaoczne, 2-letnie	Prawa i Administracji

<< poprzednie 1 2 3 4 5 ... 166 następne >>

# Więzy integralności danych

Indeksy - Przykładowa - LibreOffice Base: Table Data View

File Edit View Insert Tools Window Help



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

Przykładowa.odt : Kierunki - LibreOffice Ba

File Edit View Tools Window Help

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

Kierunki - Przykładowa - LibreOffice Base: V

File Edit View Insert Tools Window H

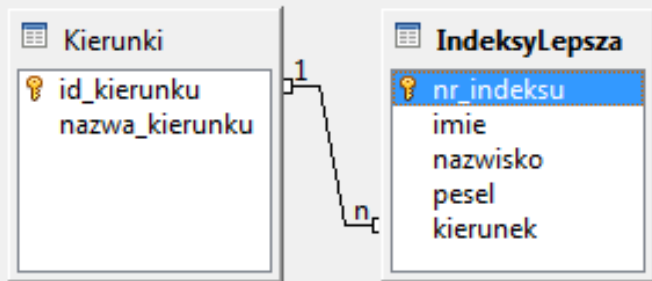
	id_kierunku	nazwa_kierunku
	1	Ochrona Środowiska
	2	Informatyka Stosowana
▶		

Przykładowa.odt : IndeksyLepsza - LibreOffice Bas

File Edit View Tools Window Help

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

# Więzy integralności danych



IndeksyLepsza - Przykładowa - LibreOffice Base: Widok danych tabeli

File Edit View Insert Tools Window Help

	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

Przykładowa.odt : Indeksy - LibreOffice

File Edit View Tools Window

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

Indeksy - Przykładowa - LibreOffice Base: Table Data View

File Edit View Insert Tools Window Help

	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\_sdbc error:  
\*violation of PRIMARY or UNIQUE KEY constraint "INTEG\_6" on table "Indeksy"  
\*Problematic key value is ("nr\_indeksu" = 2)  
caused by  
'isc\_dsqli\_execute'

OK More

# Klucze – klucz złożony

Klucze.odt : Uczniowie - LibreOffice Base

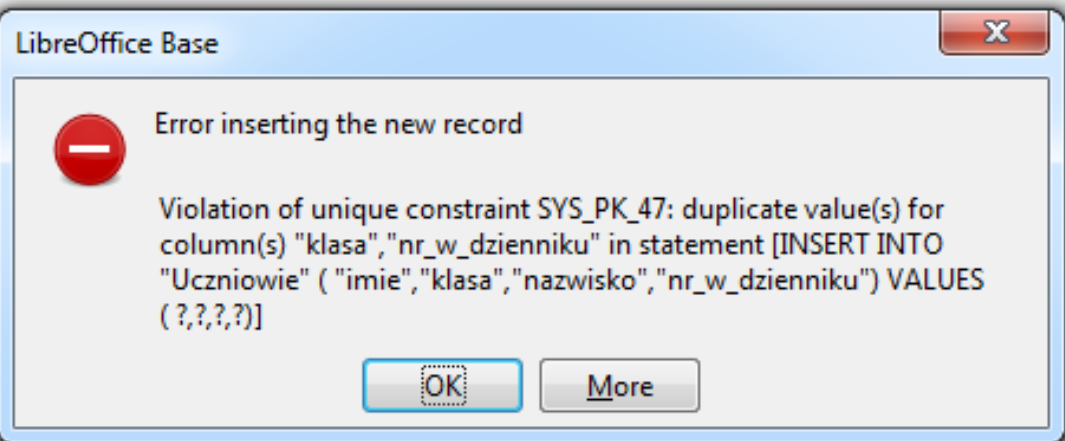
File Edit View Tools Window Help

	Field Name	Field Type
🔑	klasa	Text [ VARCHAR ]
🔑	nr_w_dzienniku	Integer [ INTEGER ]
	imie	Text [ VARCHAR ]
	nazwisko	Text [ VARCHAR ]

Uczniowie - Klucze - LibreOffice Base: Widok danych tabeli

File Edit View Insert Tools Window Help

	klasa	nr_w_dzienniku	imie	nazwisko
	1a	7	Adam	Nowak
	1a	8	Ewa	Nowaczek
	1b	7	Anna	Kowalczyk
	1b	10	Olga	Szmit
✏️	1a	7	Ula	Wróć
☀️				



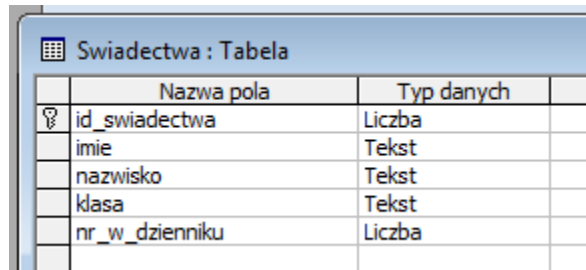
# Klucze

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 czcionką pogrubioną.

# 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



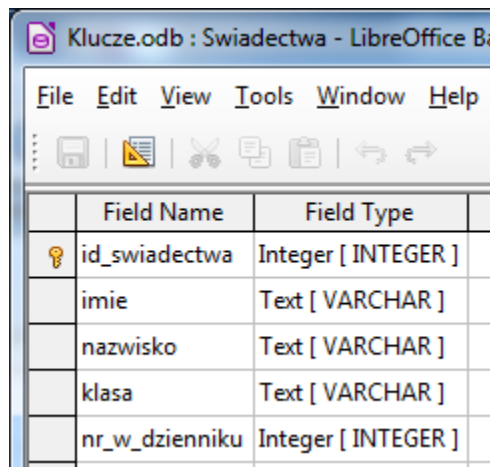
Swiadcetwa : Tabela

	Nazwa pola	Typ danych
🔑	id_swiaectwa	Liczba
	imie	Tekst
	nazwisko	Tekst
	klasa	Tekst
	nr_w_dzienniku	Liczba



Uczniowie : Tabela

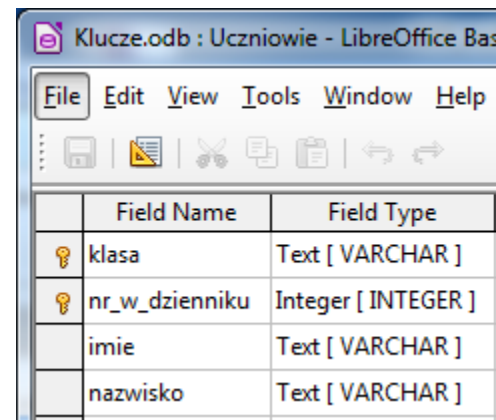
	Nazwa pola	Typ danych
🔑	klasa	Tekst
🔑	nr_w_dzienniku	Liczba
	imie	Tekst
	nazwisko	Tekst



Klucze.odb : Swiadcetwa - LibreOffice Base

File Edit View Tools Window Help

	Field Name	Field Type
🔑	id_swiaectwa	Integer [ INTEGER ]
	imie	Text [ VARCHAR ]
	nazwisko	Text [ VARCHAR ]
	klasa	Text [ VARCHAR ]
	nr_w_dzienniku	Integer [ INTEGER ]

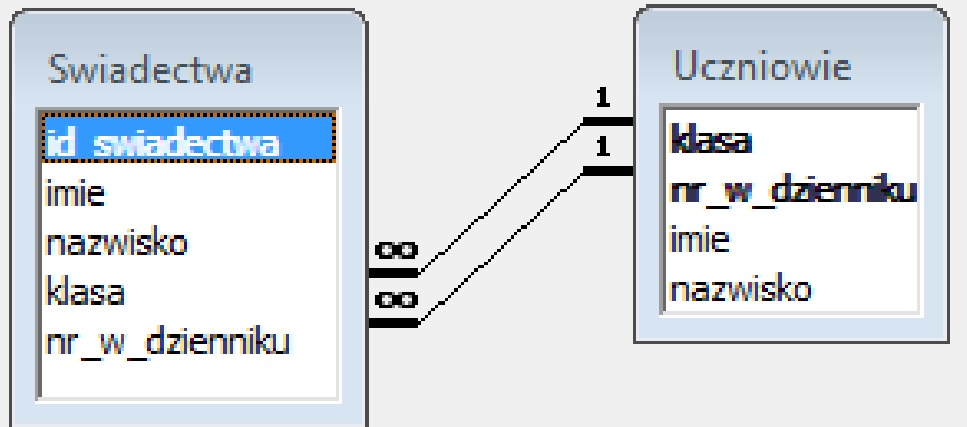


Klucze.odb : Uczniowie - LibreOffice Base

File Edit View Tools Window Help

	Field Name	Field Type
🔑	klasa	Text [ VARCHAR ]
🔑	nr_w_dzienniku	Integer [ INTEGER ]
	imie	Text [ VARCHAR ]
	nazwisko	Text [ VARCHAR ]

# Klucze obce złożone



MS ACCESS

Edytowanie relacji

Tabela/Kwerenda:	Pokrewna tabela/kwerenda:
Uczniowie	Swiadcetwa
klasa	klasa
nr_w_dzienniku	nr_w_dzienniku

Wymuszaj więzy integralności

Kaskadowo aktualizuj pola pokrewne

Kaskadowo usuń rekordy pokrewne

Typ relacji: Jeden-do-wielu

Utwórz

Anuluj

Typ sprzężenia..

Utwórz nowe..

# 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.

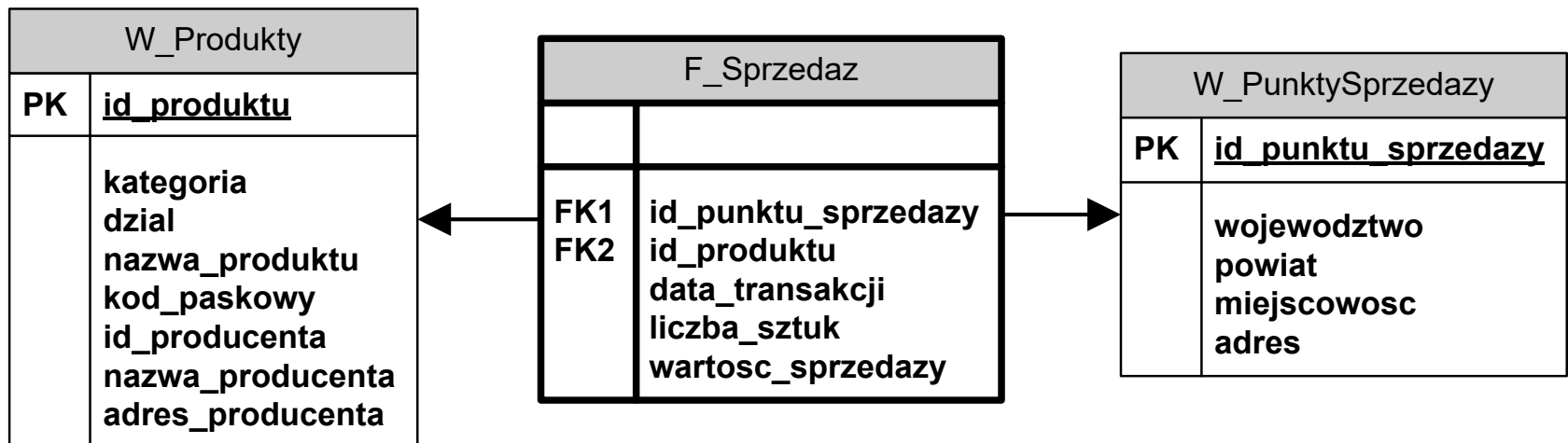
### Miary

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

# Model schematów danych

## Gwiazda

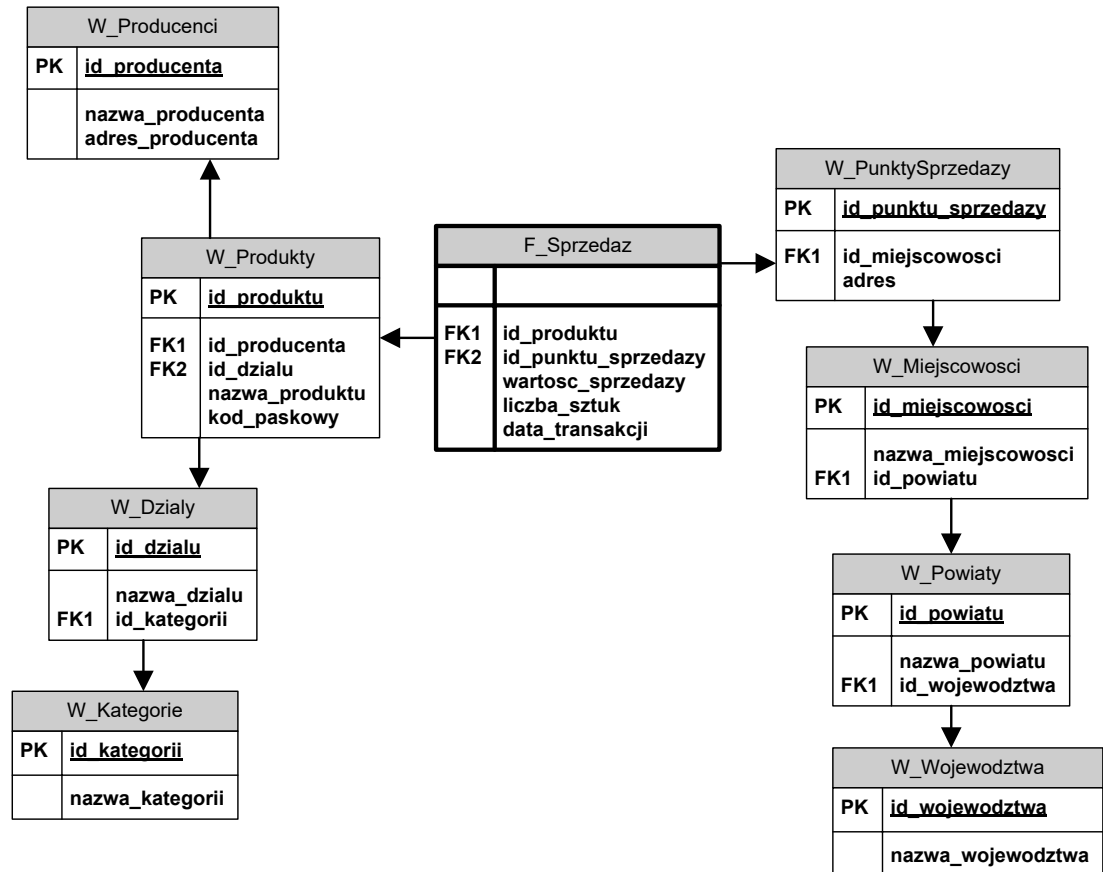
- jedna tabela faktów w postaci znormalizowanej,
- wiele tabel wymiarów w postaci znormalizowanej lub zdenormalizowanej – każdemu wymiarowi odpowiada tylko jedna tabela (nie ma hierarchii),
- prosta struktura,
- niewielka liczba połączeń (→ złączeń),
- klucz tabeli faktów obejmuje wszystkie kolumny oprócz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



# Model schematów danych

## Płatek śniegu

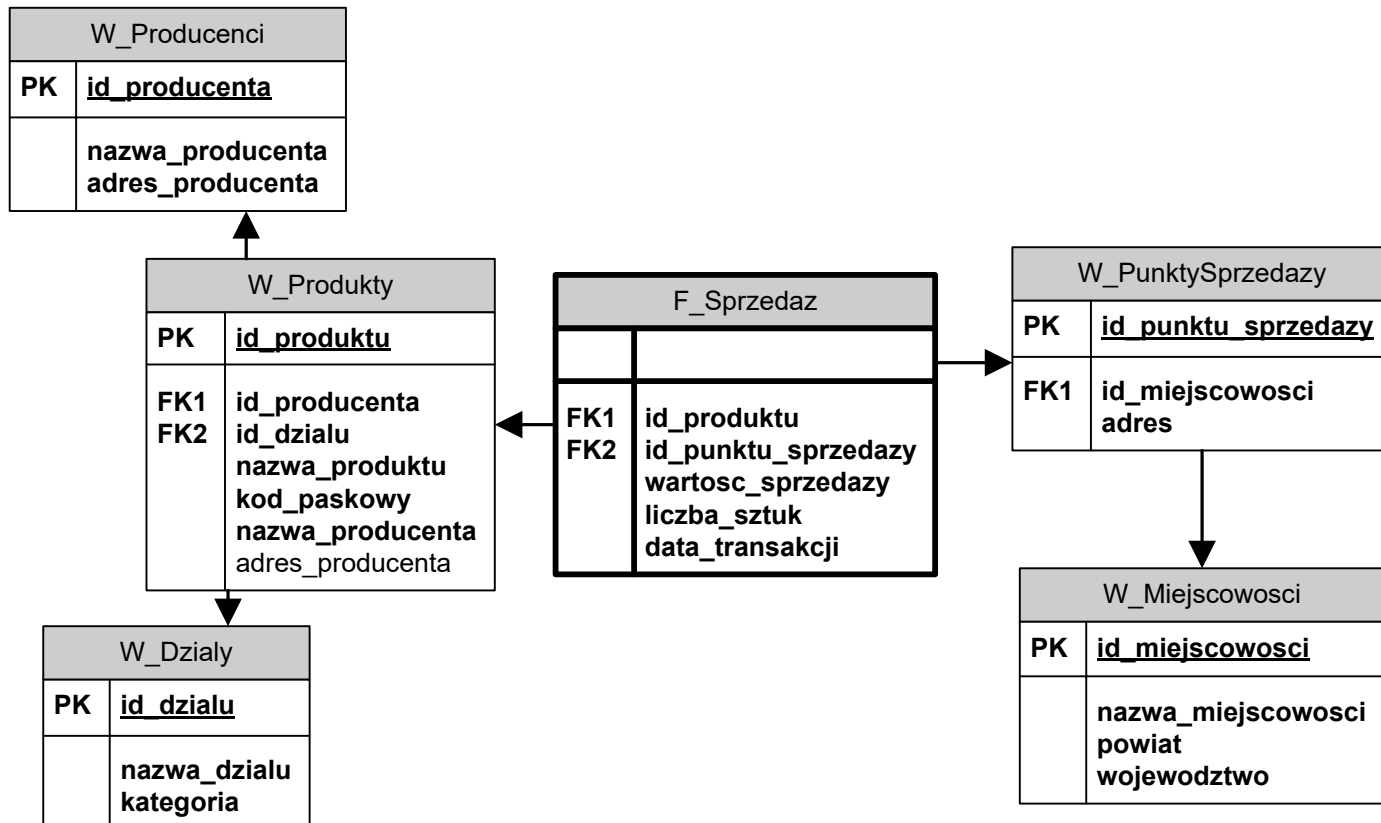
- jedna tabela faktów (podobnie jak w schemacie gwiazdy),
- tabele wymiarów znormalizowane (tworzą hierarchię – płatek śniegu),
- schemat łatwiejszy do modyfikacji
- większa liczba połączeń – większa liczba złączeń – spadek efektywności zapytań ?\*



# Model schematów danych

## Gwiazda-płatek

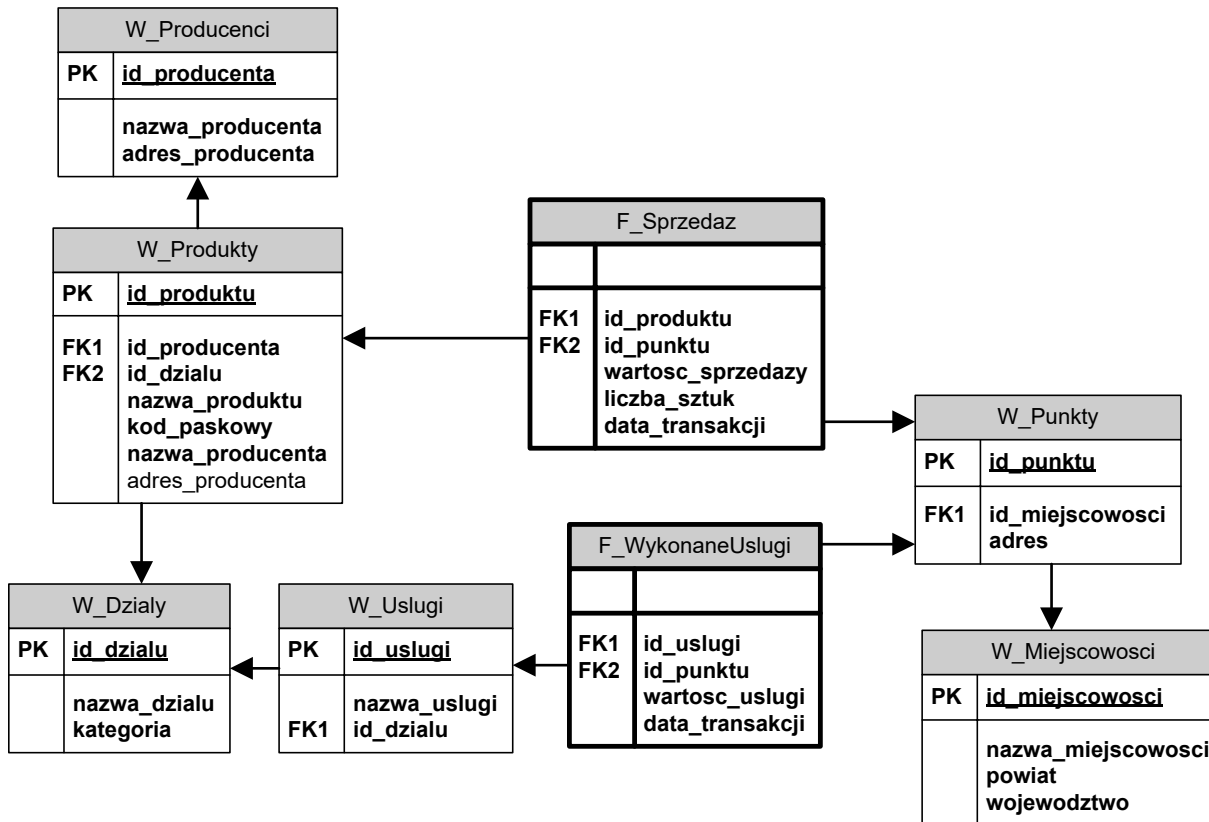
- schemat posiadający część wymiarów znormalizowanych, a część nie – optymalizacja wydajności, wybrane tabele wymiarów nie będą modyfikowane ani rozwijane.



# Model schematów danych

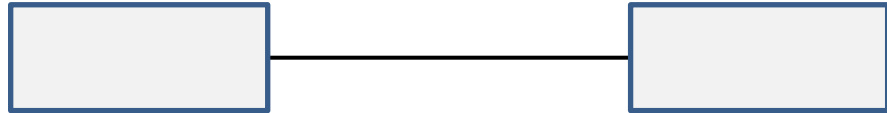
## Konstelacja

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

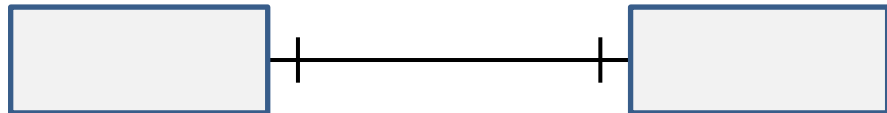


# Notacja Martina

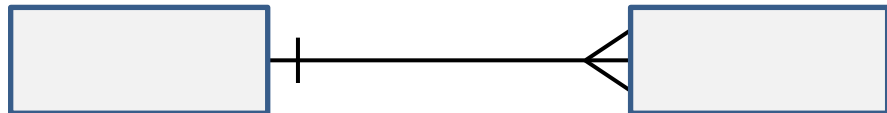
brak liczebności



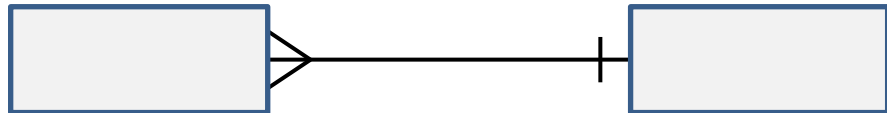
jeden do jednego



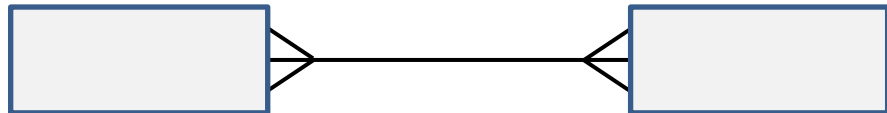
jeden do wielu



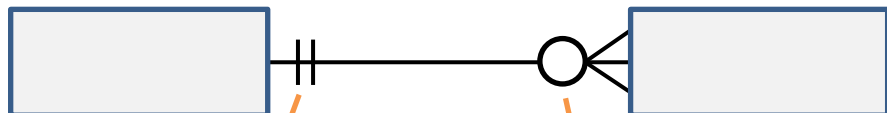
wiele do jednego



wiele do wielu



jeden do wielu



wymagane

opcjonalne

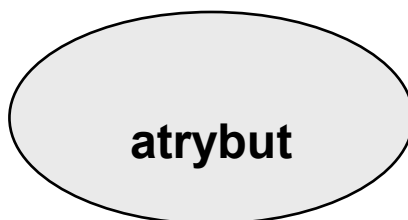
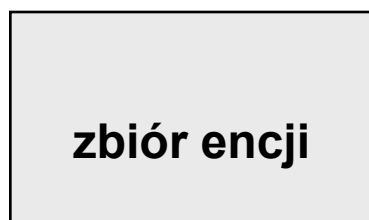
# Diagramy związków encji (ERD)

## notacja Chena

Zbiory encji – prostokąty

Atrybuty – owale

Związki encji - romby



Nazwy zbiorów encji – liczba mnoga

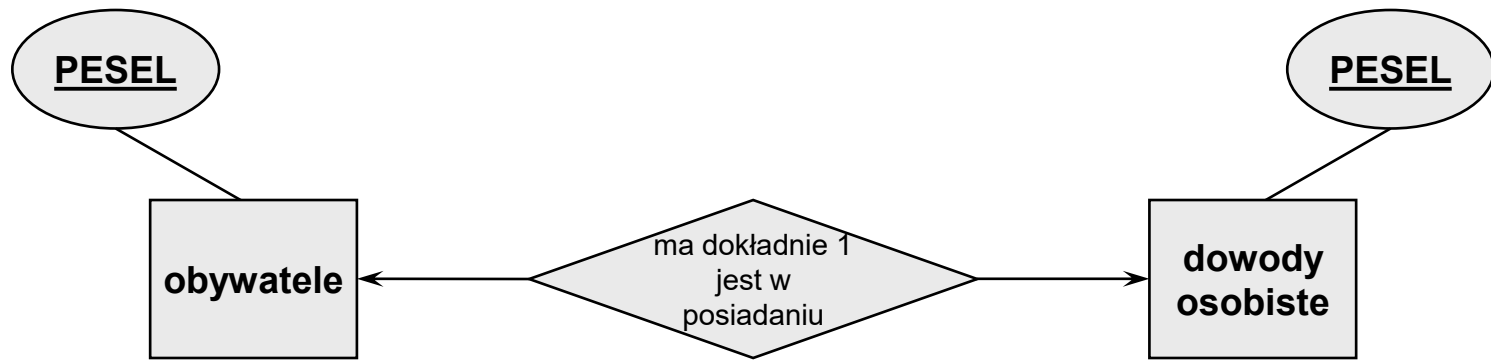
Połączenie *atrybut* – *zbiór encji* jest nieorientowane

Połączenie *zbiór encji* – *związek* - *zbiór encji* może być lub nie być skierowane.

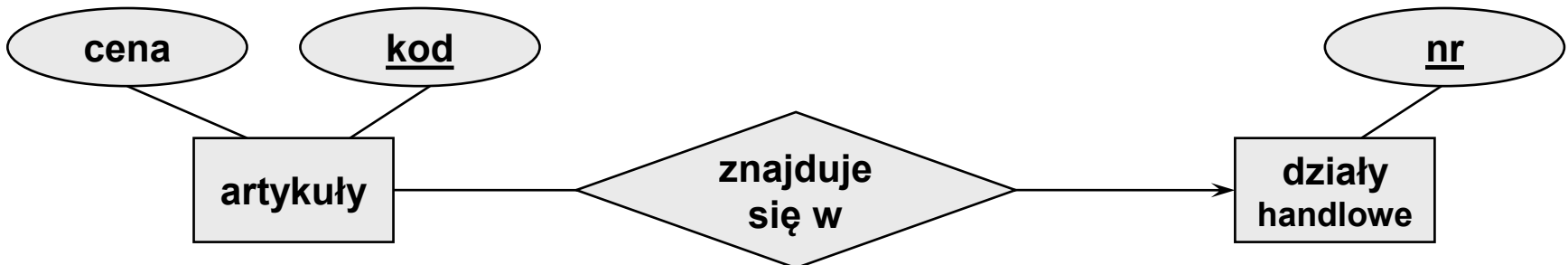
Strzałka wskazuje encję pojedynczą.

# Diagramy związków encji (ERD) notacja Chena

## Związek jeden-do-jednego



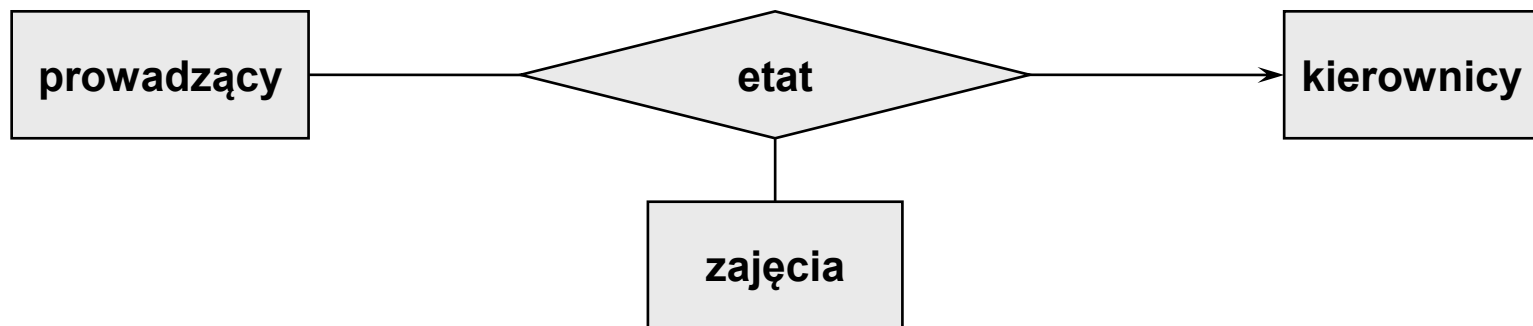
## Związek wiele-do-jednego



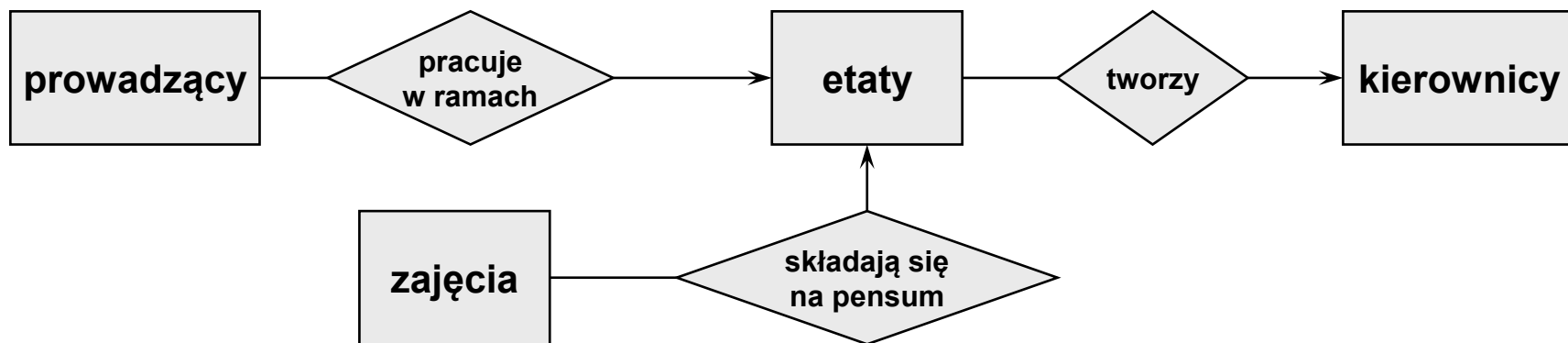
# Diagramy związków encji (ERD)

## notacja Chena

### Związki wieloargumentowe

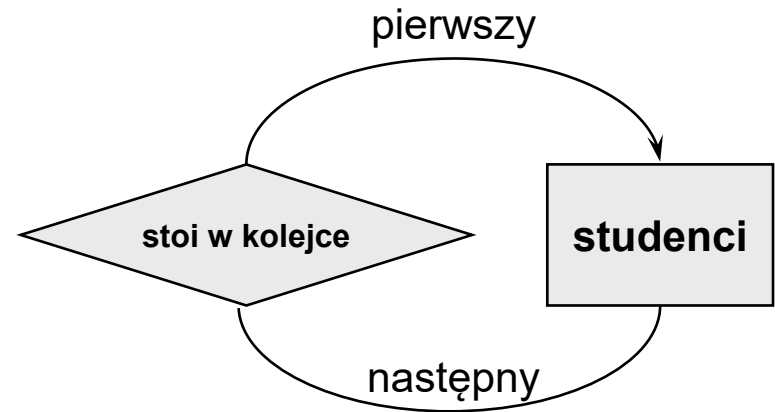


### Dekompozycja związku wieloargumentowego w związki dwuargumentowe

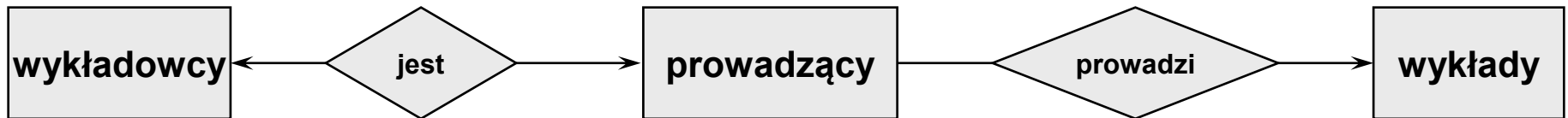


# Diagramy związków encji (ERD) notacja Chena

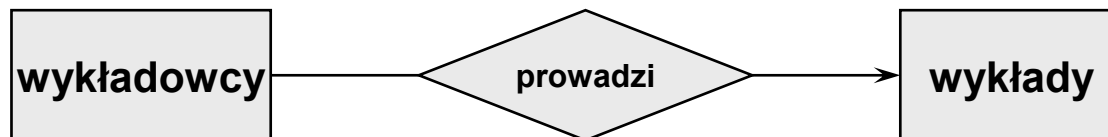
Wielokrotne wykorzystanie zbioru encji



Redundancja



redukcja



# ZBIORY ENCJI SŁABYCH

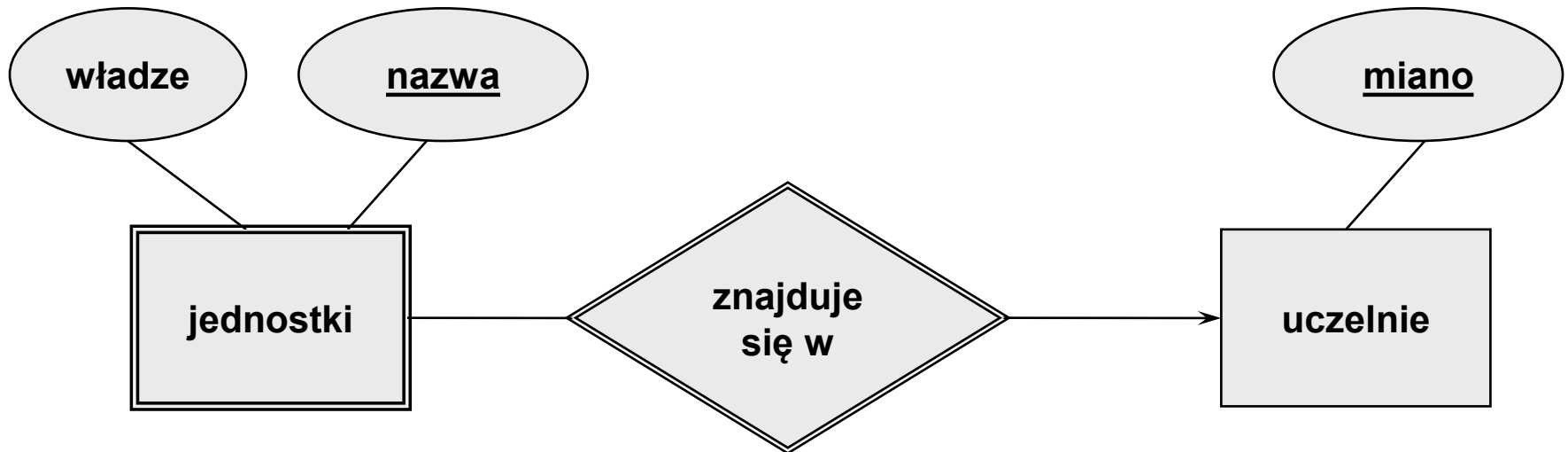
Encja słaba:

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,...})

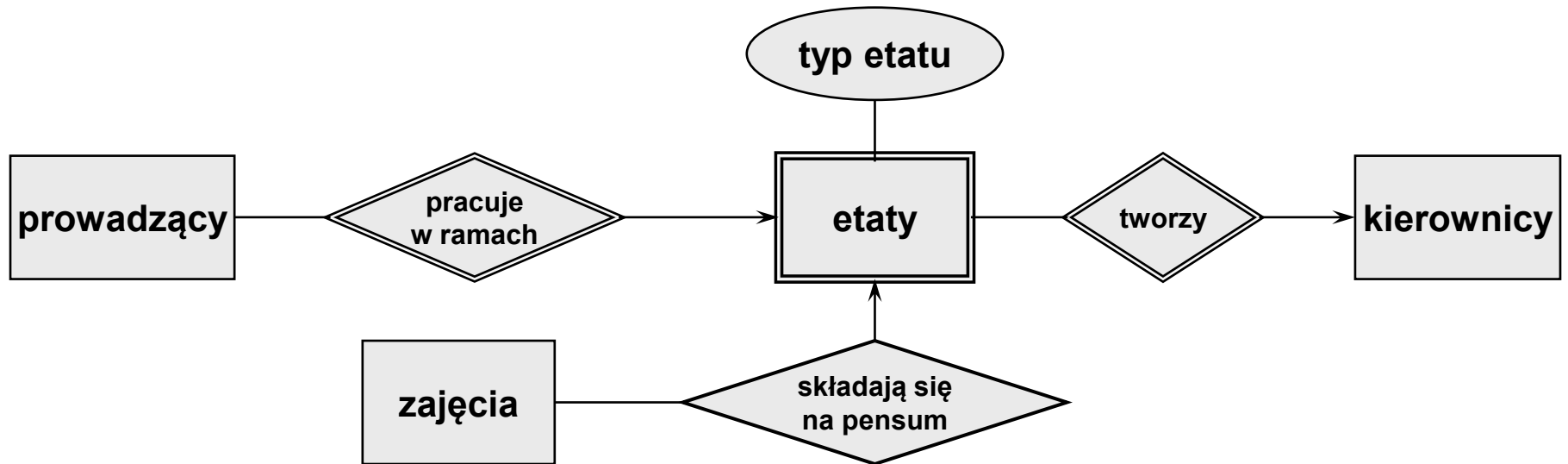
uczelnia: miano (np. {AGH, UE, UP, UJ,..})



? 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

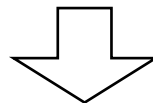
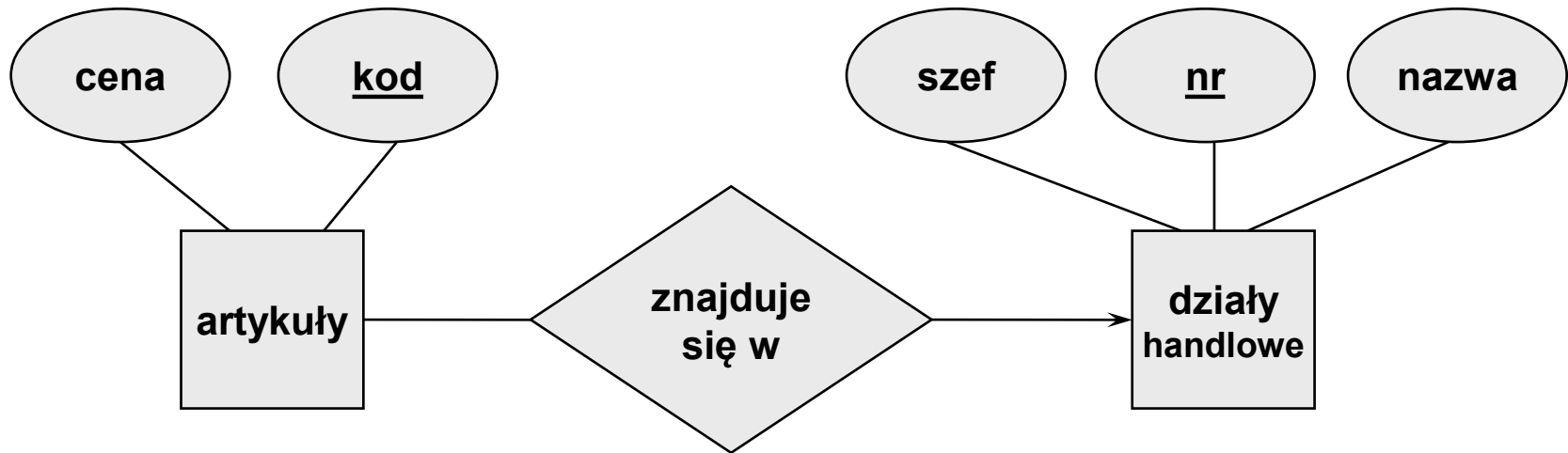
Dla każdego zbioru encji tworzy się tabelę o nazwie odpowiadającej nazwie zbioru encji i takim samym zbiorze atrybutów.

Dla każdego związku encji tworzy się tabelę o zbiorze atrybutów, w którym są 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



*artykuły*

<u>kod</u>	cena
123	100,50
124	25,00

*położenie*

<u>kod</u>	nr
123	1
124	2

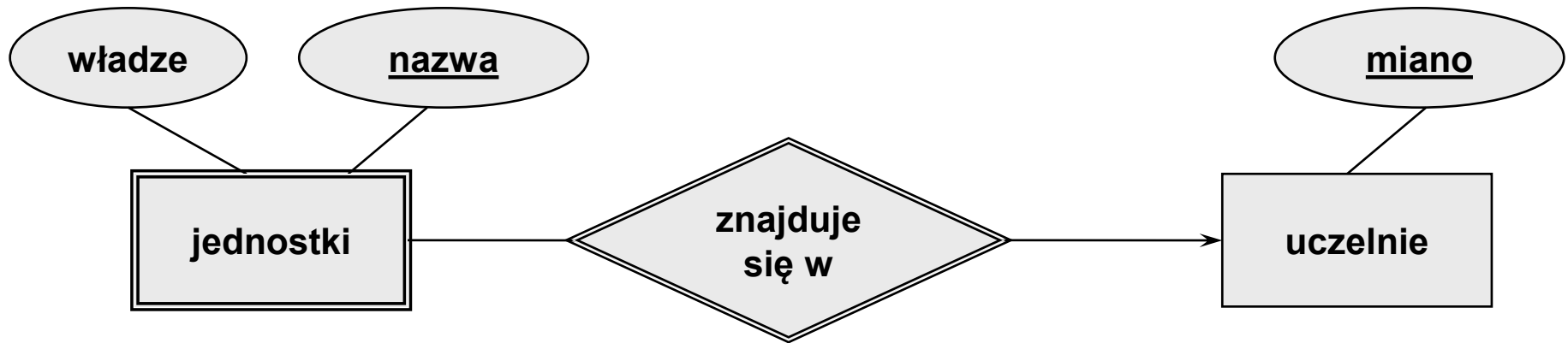
*działy handlowe*

<u>nr</u>	nazwa	szef
1	chemia	J. Nowak
2	prasa	L. Kot

# Przekształcanie diagramów związków encji do postaci relacyjnego modelu danych

Postępowanie w przypadku związków słabych:

- relacja odpowiadająca słabemu zbiorowi encji musi zawierać także atrybuty kluczy 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



*jednostki*

<u>nazwa</u>	władze	<u>miano</u>
Wydz Inf. St.	dziekan	AGH
Wydz Inf. St.	dziekan	UJ
Wydz Mat. St.	dziekan	AGH

*uczelnie*

<u>miano</u>
AGH
UJ

# 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

<u>typ_silnika</u>	pojemność	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  $A_1 \dots A_x$ ,
- nietrywialna – jeśli istnieje  $B_u B_w$  zawierające się w  $A$ ,
- całkowicie nietrywialna – 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

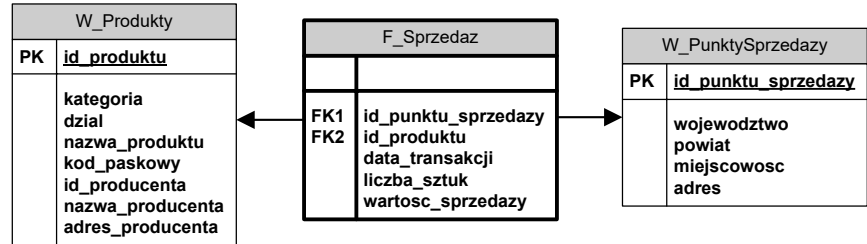
Zależność wielowartościowa:  $A \rightarrow \rightarrow B$

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

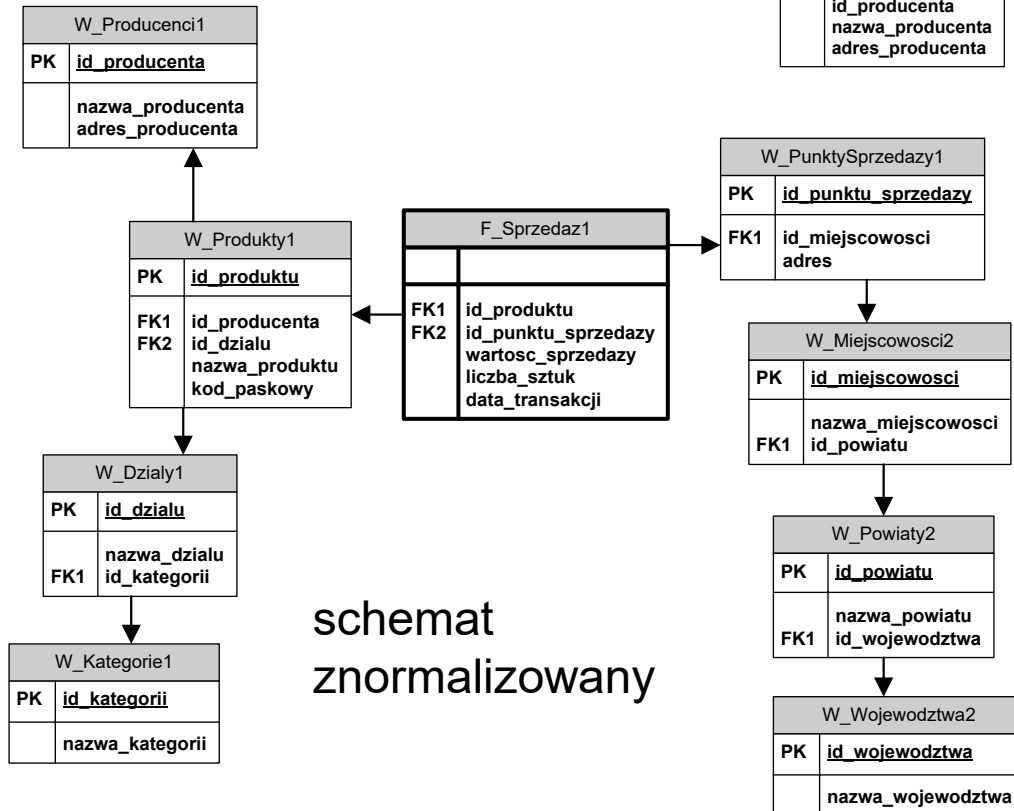
# NORMALIZACJA SCHEMATU RELACYJNEJ BAZY DANYCH

Cel normalizacji:

- zapobieganie anomaliom,
- doprowadzenie do relacji elementarnych.



schemat zdenormalizowany



schemat znormalizowany

# PIERWSZA POSTAĆ NORMALNA (1NF)

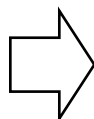
Relacja jest w pierwszej postaci normalnej (**1NF**), jeśli wartości atrybutów są pojedynczymi wartościami atomowymi.

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

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

<u>klasa</u>	sala						
1a	<table border="1"><thead><tr><th>adres</th><th>liczba miejsc</th></tr></thead><tbody><tr><td>Krótka 2a</td><td>40</td></tr><tr><td>Długa 3b</td><td>35</td></tr></tbody></table>	adres	liczba miejsc	Krótka 2a	40	Długa 3b	35
adres	liczba miejsc						
Krótka 2a	40						
Długa 3b	35						
1b	315						

<u>klasa</u>	sala
1a	314, 316
1b	315



<u>klasa</u>	sala1	sala2
1a	314	316
1b	315	

# 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 )

<u>imię</u>	<u>nazwisko</u>	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

# TRZECIA POSTAĆ NORMALNA (3NF)

Relacja  $X$  jest w postaci **3NF**  $\Leftrightarrow$

jeśli zachodzi dla dowolnej nietrywialnej zależności  $A_1 \dots A_x \rightarrow B$ ,  
to  $A_1 \dots A_x$  jest nadkluczem **albo**  $B$  jest podkluczem.

Uwagi:

- Nie są możliwe zależności funkcyjne między atrybutami niekluczowymi.

Dla przykładu: **Notes( imię , nazwisko , imieniny , urodziny , znak zodiaku )**  
 $urodziny \rightarrow \text{znak zodiaku}$  - {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)

przykład 0: pesel / nr dowodu / nr paszportu

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 }  $\rightarrow$  kod,

kod  $\rightarrow$  miasto (kod jest podkluczem, spełniony jest warunek 3NF, ale nie BCNF)

# POSTAĆ NORMALNA Boyce-Codda (BCNF)

Relacja  $X$  jest w postaci **BCNF**  $\Leftrightarrow$

jeśli zachodzi dla dowolnej nietrywialnej zależności  $A_1 \dots A_x \rightarrow B$ ,  
to  $A_1 \dots A_x$  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.

# NORMALIZACJA - przykład

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

<u>nr_dz</u>	imię	nazwisko	<u>klasa</u>	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  $K = \{\text{nr\_dz}, \text{klasa}\}$

Zależności:

$\text{nr\_dz}, \text{klasa} \rightarrow \text{nazwisko}$

$\{\text{nr\_dz}, \text{klasa}\} - \text{nadklucz } K$

$\text{nr\_dz}, \text{klasa}, \text{nazwisko} \rightarrow \text{sala}$

$\{\text{nr\_dz}, \text{klasa}, \text{nazwisko}\} - \text{nadklucz } K$

$\text{klasa} \rightarrow \text{sala}$

$\{\text{klasa}\} - \text{jest podkluczem } K, (\text{a nie nadkluczem})$

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

# DEKOMPOZYCJA DO POSTACI BCNF

Dekompozycję przeprowadza się wyznaczając pewną zależność  $A_1 \dots A_x \rightarrow B_1 \dots B_y$  naruszającą warunek BCNF, następnie rozbudowuje się zbiór  $B_1 \dots B_y$  o kolejne atrybuty, zależne funkcyjnie od  $A_1 \dots A_x$  i wydziela się nową relację, zawierającą  $A_1 \dots A_x$  i  $B_1 \dots B_y$ , przy czym  $B_1 \dots B_y$  usuwa się z dekomponowanej relacji.

Dla podanego przykładu:

uczeń (nr\_dz, imię, nazwisko, klasa)

klasa (klasa, sala)

<u>nr_dz</u>	imię	nazwisko	<u>klasa</u>
1	Andrzej	Nowak	1a
2	Gustaw	Kot	1a
3	Ala	Wójcik	1a
1	Ola	Nowak	1b
2	Gustaw	Werner	1b
3	Ala	Wójcik	1b

<u>klasa</u>	sala
1a	314
1b	315

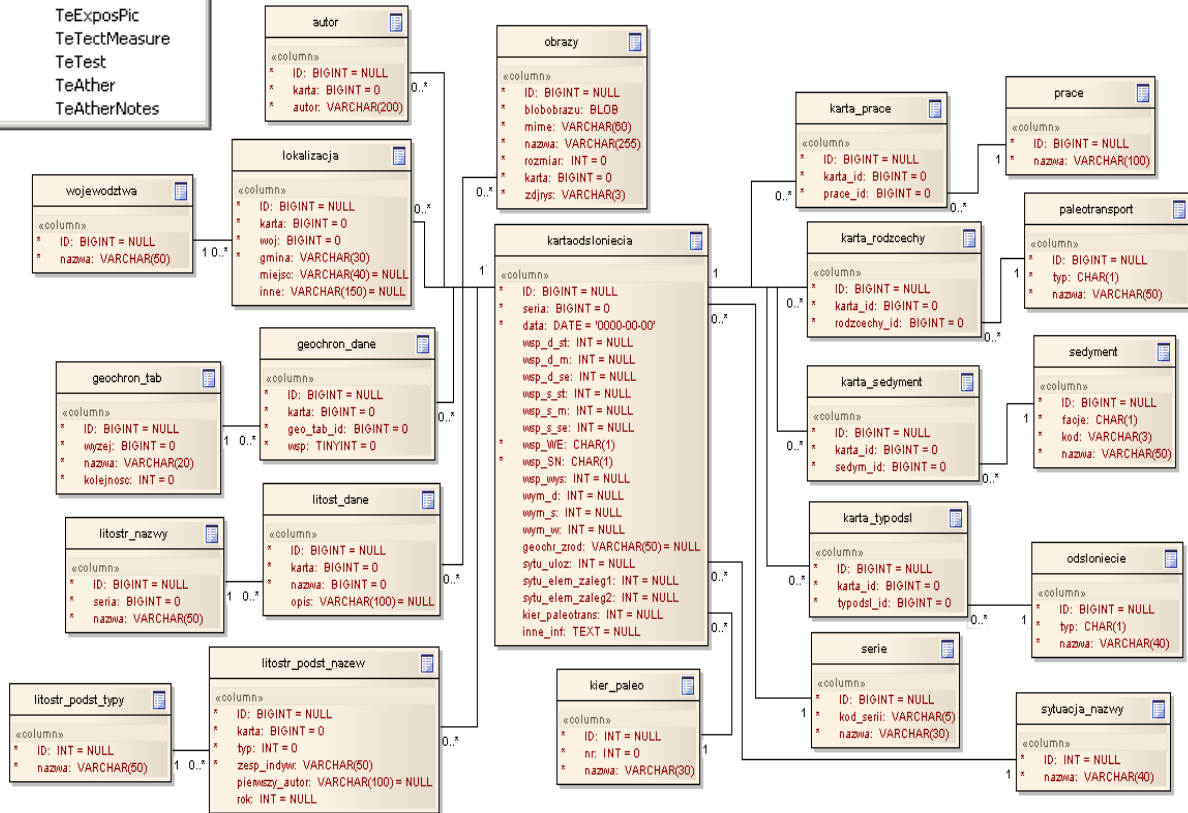
## Rozdzielanie danych po dekompozycji do postaci BCNF

Przy dekompozycji do postaci BCNF (w oparciu o zależności funkcyjne) dane z dekomponowanej relacji przenosi się do nowych relacji tak, że jest możliwe odzyskanie tych danych poprzez wykonanie wszystkich możliwych połączeń krotek nowych relacji (złączenie naturalne).

# zdenormalizowana

Exposition		
LiAddNotes	SeErosionDitch	SeAtherL
ExTerrainNo	LiNameBase	SeAtherHole
ExUnitName	LiBibliogNote	SeDiagonalLaye
ExDescDate	ChrPrecision	SeFlowRound
ExNotes	ChrRange	SeRipplemarkAs
ExCommune	ChrBase	SeAtherA
ExCity	SeC	Se_N_S
ExAuthor1	SeCS	Se_W_E
ExAuthor2	SeSC	Se_NE_SW
ExLongitudeDeg	SeS	Se_NW_SE
ExLongitudeMin	SeSM	Se_N
ExLongitudeSec	SeMS	Se_S
ExLatitudeDeg	SeM	Se_E
ExLatitudeMin	SeCL	Se_W
ExLatitudeSec	SeMC	Se_NE
ExHeightAboveSeelvl	SeLG	Se_SE
ExLength	SeLP	Se_SW
ExWidth	SeL5	Se_NW
ExHigh	SeDO	TeDetailedProfil
ExNaturalKind	SeSD	TeFotoDoc
ExArtificialKind	SeML	TeExposPic
ExLayerArrangement	SeF	TeTectMeasure
ExSinkAzimuth	SeST	TeTest
ExFallAngle	SeTraceDrag	TeAther
LiUnits	SeGrainOrient	TeAtherNotes
LiAddUnits	SeTrail	

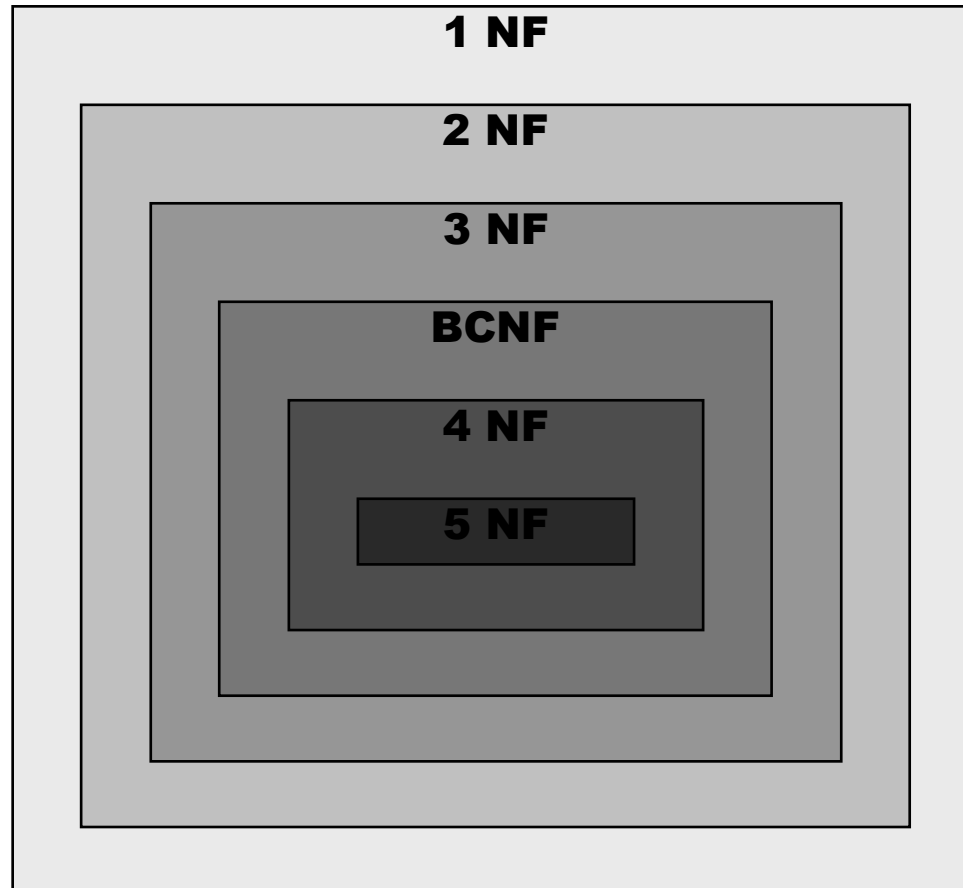
Author
IDAuthor
FName
SName



# znormalizowana

# INNE POSTACI NORMALNE

- czwarta postać normalna (4NF), (co najwyżej jedna zależność wielowartościowa)
- piąta postać normalna (5NF, postać złączenie-rzut),
- postać normalna domena-klucz.



# OPERACJE NA DANYCH

## Proste działania teoriomnogościowe

- suma
- różnica
- iloczyn

## Działania złożone

## Wybieranie danych

- rzut (projekcja)
- selekcja

## Złączenia tabel

- iloczyn kartezjański (CROSS JOIN)
- złączenie  $\theta$  (teta)
- złączenie naturalne (NATURAL JOIN)
- równozłączenie / złączenie wewnętrzne (INNER JOIN)
- złączenia zewnętrzne (lewostronne, prawostronne, pełne – OUTER JOIN)
- autozłączenie (auto-join)
- złączenie częściowe (semi-join)
- iloraz

# 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

## ILOCZYN

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

## ILOCZYN

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  $\{b_1, \dots, b_k\}$  jest podzbiorem atrybutów  $A \{a_1, \dots, a_n\}$

$A$

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

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

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  $A$ .

$A$

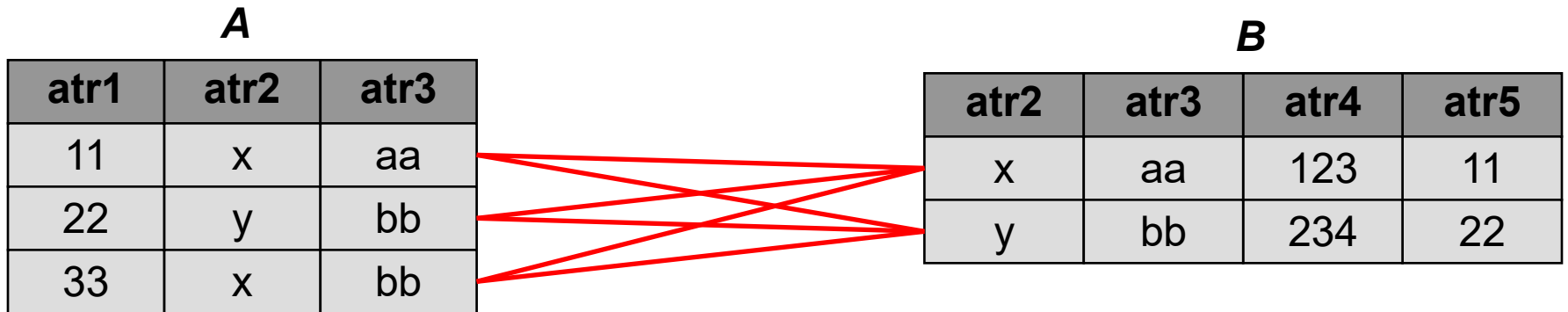
atr1	atr2	atr3	atr4
1	x	aa	123
2	y	bb	234
3	x	aa	576
4	y	bb	121
3	x	aa	260

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

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

# OPERACJE NA DANYCH

Iloczyn kartezyński



A  $\blacktriangleright$   $\blacktriangleleft$  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 ów 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 ÷ B**

atr1	atr2
1	x
2	y

3	z
---	---

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

# OPERACJE NA DANYCH

## ZŁĄCZENIE NATURALNE

$A(\text{atr1}, \text{atr2}, \text{atr3}, \dots)$ ,  $B(\text{atr2}, \text{atr3}, \text{atr4}, \text{atr5}, \dots)$

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:

1. wyznacza się  $A \times B$ ,
2. z wyznaczonego iloczynu kartezjańskiego wybiera się tylko te krotki, które mają te same wartości dla atrybutów wspólnych dla  $A$  i  $B$  ( $A.\text{atr2} \leftrightarrow B.\text{atr2}$ ,  $A.\text{atr3} \leftrightarrow B.\text{atr3}$ )
3. dokonuje się rzutu usuwającego zdublowane kolumny ( $A.\text{atr2}, B.\text{atr2} \rightarrow \text{atr2}, \dots$ )

**A**

atr1	atr2	atr3
1	x	aa
2	y	bb
3	x	bb
4	y	bb

**B**

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

# OPERACJE NA DANYCH

## ZŁĄCZENIE NATURALNE

$A \times B$

atr1	A.ATR2	A.ATR3	B.ATR2	B.ATR3	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

1

atr1	A.ATR2	A.ATR3	B.ATR2	B.ATR3	atr4	atr5
1	x	aa	x	aa	123	11
2	y	bb	y	bb	234	22
4	y	bb	y	bb	234	22

2

$A \blacktriangleright \blacktriangleleft B$

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

3

# 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

<b>A</b>		<b>B</b>	
atr1	atr2	atr3	atr4
1	x	x	AA
2	y	y	BB
3	x	v	CC

**A**  $\blacktriangleright$   $\blacktriangleleft$  **B**  
 $A.attr2 = B.attr3$

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 zadany operator porównania ( =, <>, <, >, <=, >= )

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

**A**

atr1	atr2
1	x
2	y
3	x

**B**

atr3	atr4
x	AA
y	BB
v	CC

**A**  $\blacktriangleright$   $\blacktriangleleft$   $A.attr2 > B.attr3$  **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 (*atrybutyzlaczienia*)

są wszystkie krotki relacji A, zawierające wartości atrybutów

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

**A**

atr1	atr2
1	x
2	x
3	z

**B**

atr3	atr4
x	AA
y	BB
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		

A

Imiona	
imie	Mazurko
Janusz	Kowalski
Adam	Wieloch
Kamil	Zelazki

B

imie	Pesel
Janusz	111
Tomasz	222
Zuzanna	333
Kamil	444
WAWA	555

left join  
A ~~to~~ B

imie	Mazurko	imie	pesel
Janusz	Kowalski	Janusz	111
Kamil	Zelazki	Kamil	444

# OPERACJE NA DANYCH

## Złączenie zewnętrzne prawostronne (RIGHT OUTER JOIN)

Wynikiem złączenia zewnętrznego prawostronnego  
 $A \text{ LEFT OUTER JOIN } B \text{ ON } (a_{trybuty} = b_{trybuty})$   
są wszystkie krotki relacji B, zawierające wartości atrybutów  
z relacji A dla uzgodnionych warunkiem złączenia ( $a_{trybuty} = b_{trybuty}$ ) krotek z relacji A  
lub wartości puste.

**A**

atr1	atr2
1	x
2	x
3	z

**B**

atr3	atr4
x	AA
y	BB
u	CC

**$A \text{ RIGHT OUTER JOIN } B \text{ ON } (A.a_{tr2} = B.a_{tr3})$**

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 (*atrybutyzlaczzenia*) są wszystkie krotki relacji B, zawierające wartości atrybutów z relacji A dla uzgodnionych warunkiem złączenia (*atrybutyzlaczzenia*) krotek z relacji A lub wartości puste.

**A**

atr1	atr2
1	x
2	x
3	z

**B**

atr3	atr4
x	AA
y	BB
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



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.

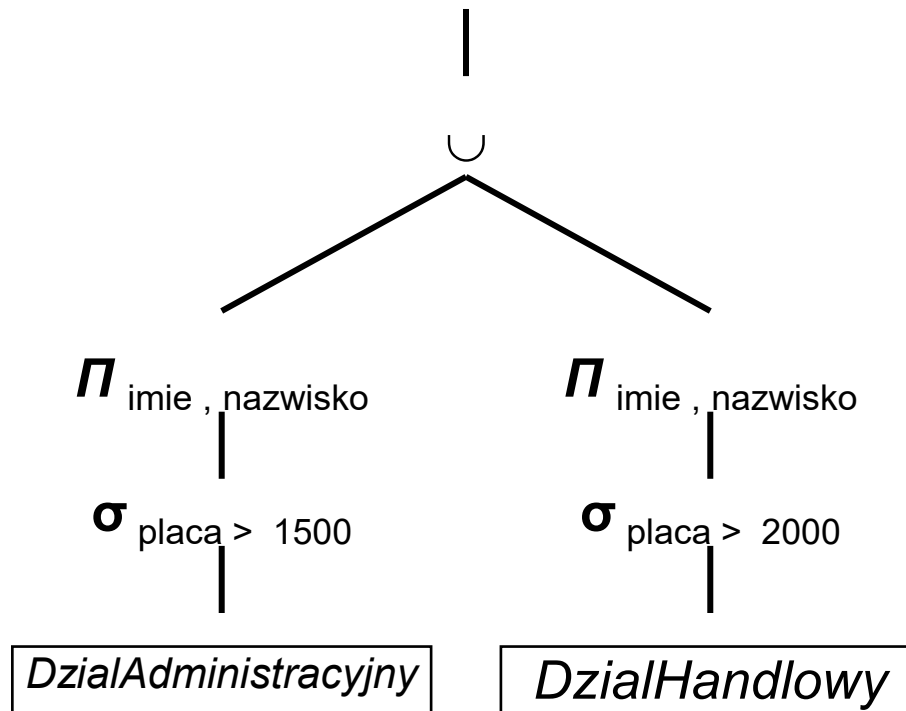
# OPERACJE NA DANYCH

## ZŁOŻENIA

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

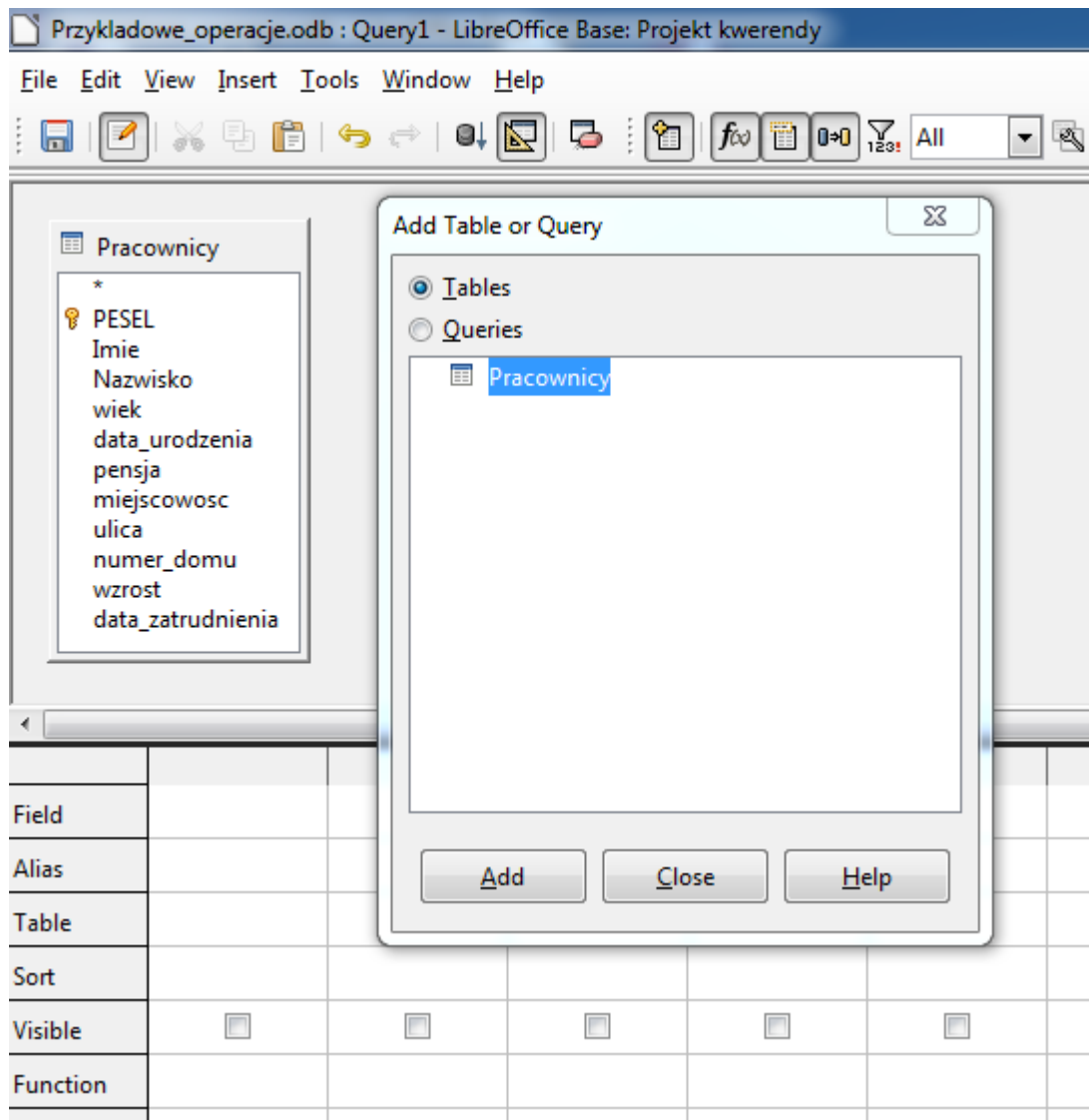
$(\pi_{\text{IMIE}, \text{NAZWISKO}} (\sigma_{\text{PLACA} > 1500} (\text{DzialAdministracyjny}))) \cup (\pi_{\text{IMIE}, \text{NAZWISKO}} (\sigma_{\text{PLACA} > 2000} (\text{DzialHandlowy})))$

Powyższe złożenie można przedstawić przy pomocy drzewa operacji:





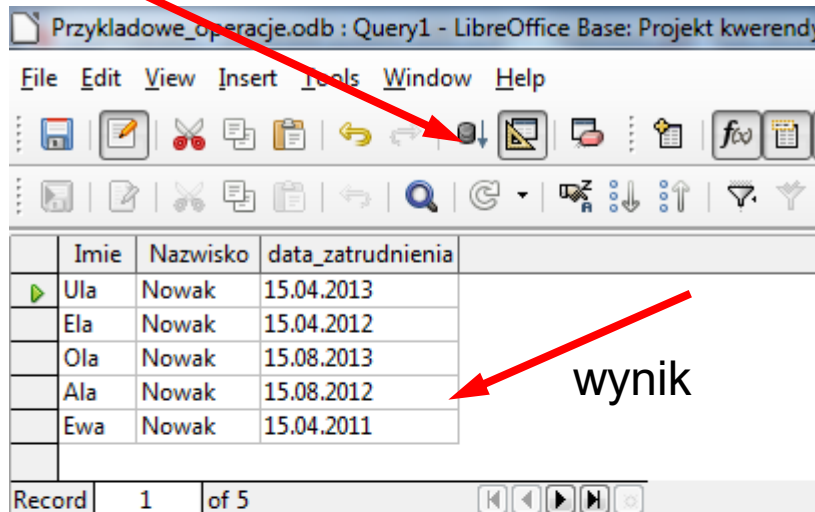
# Przykładowe kwerendy w LibreOffice



Wybór tabel biorących udział w kwerendzie – na początek – jedna

# Przykładowe kwerendy w LibreOffice

uruchomienie zapytania



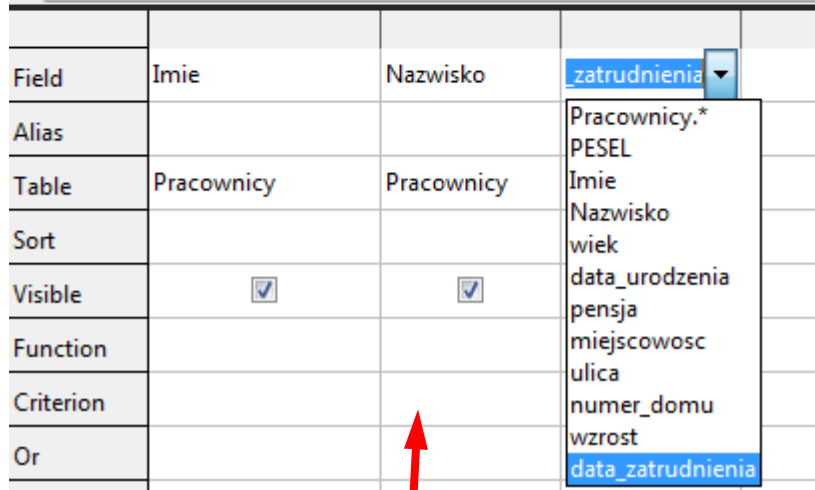
Przykładowe\_operacje.odt : Query1 - LibreOffice Base: Projekt kwerendy

File Edit View Insert Tools Window Help

	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

Record 1 of 5

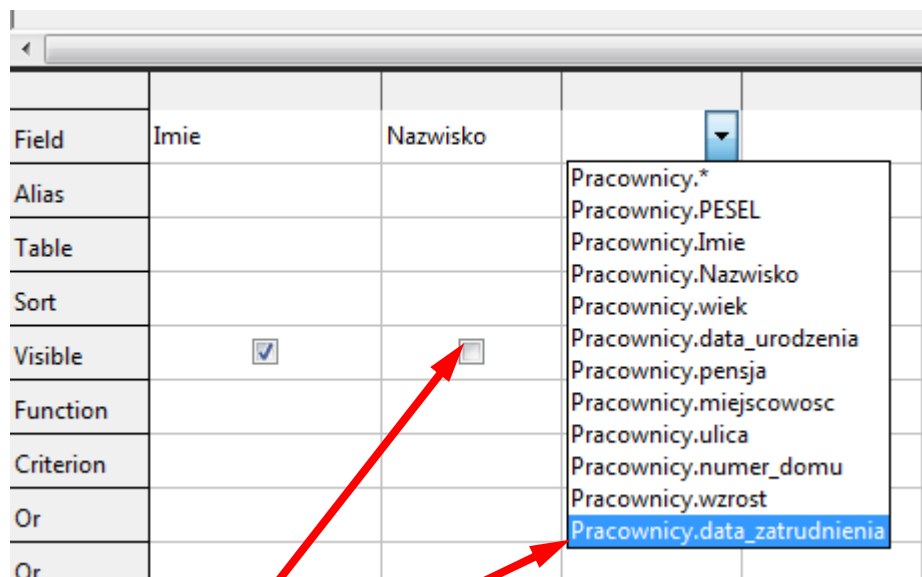
wynik



Field	Imie	Nazwisko	zatrudnienia
Alias			Pracownicy.* PESEL
Table	Pracownicy	Pracownicy	Imie Nazwisko wiek
Sort			data_urodzenia pensja miejscowosc ulica numer_domu wzrost
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Function			
Criterion			
Or			data_zatrudnienia

Projekt kwerendy

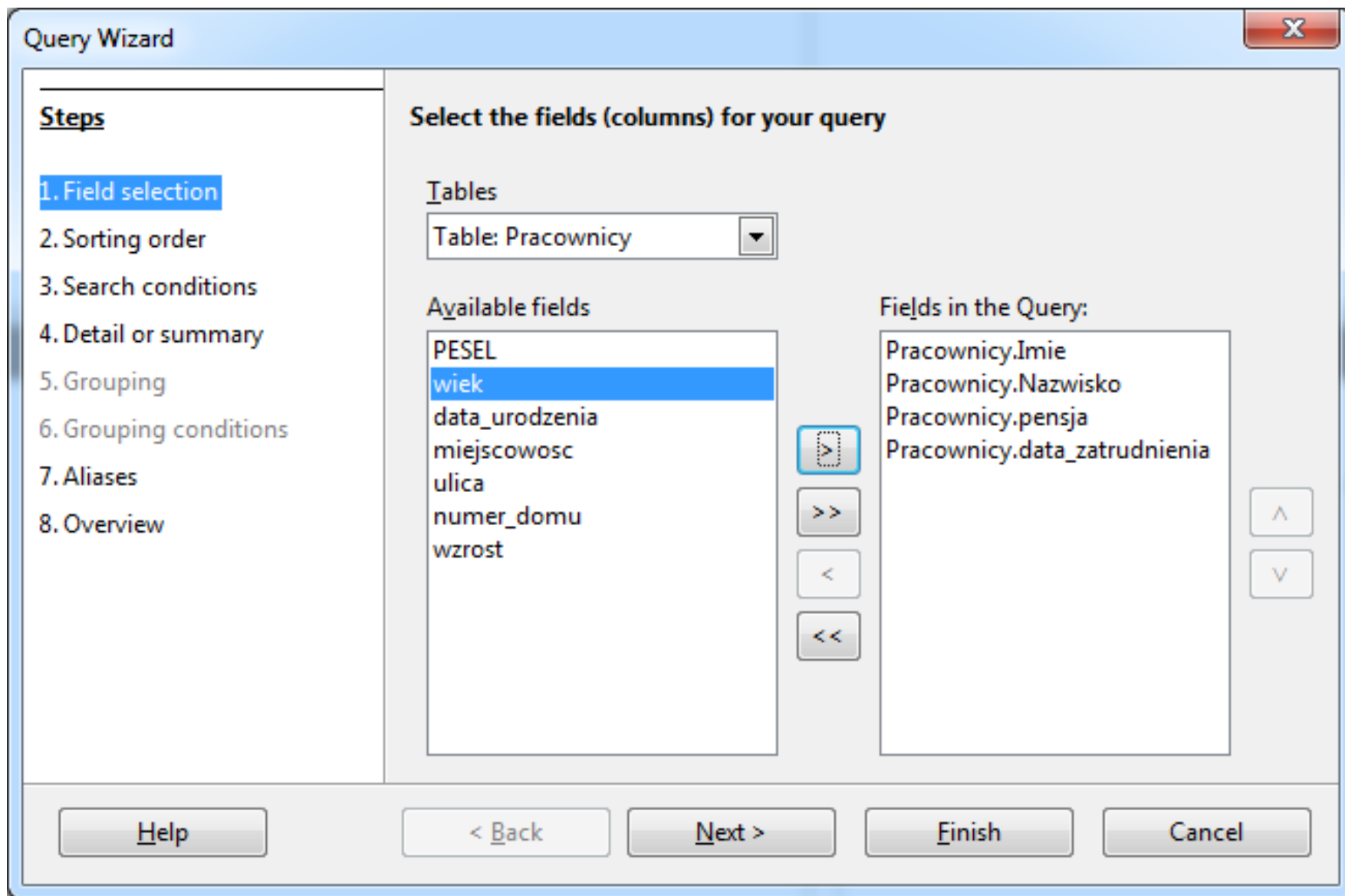
PROJEKCJA



Field	Imie	Nazwisko	
Alias			Pracownicy.* Pracownicy.PESEL Pracownicy.Imie Pracownicy.Nazwisko Pracownicy.wiek Pracownicy.data_urodzenia Pracownicy.pensja Pracownicy.miejscowosc Pracownicy.ulica Pracownicy.numer_domu Pracownicy.wzrost Pracownicy.data_zatrudnienia
Table			
Sort			
Visible	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Function			
Criterion			
Or			
Or			

wybor pól

# Przykładowe kwerendy w LibreOffice



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

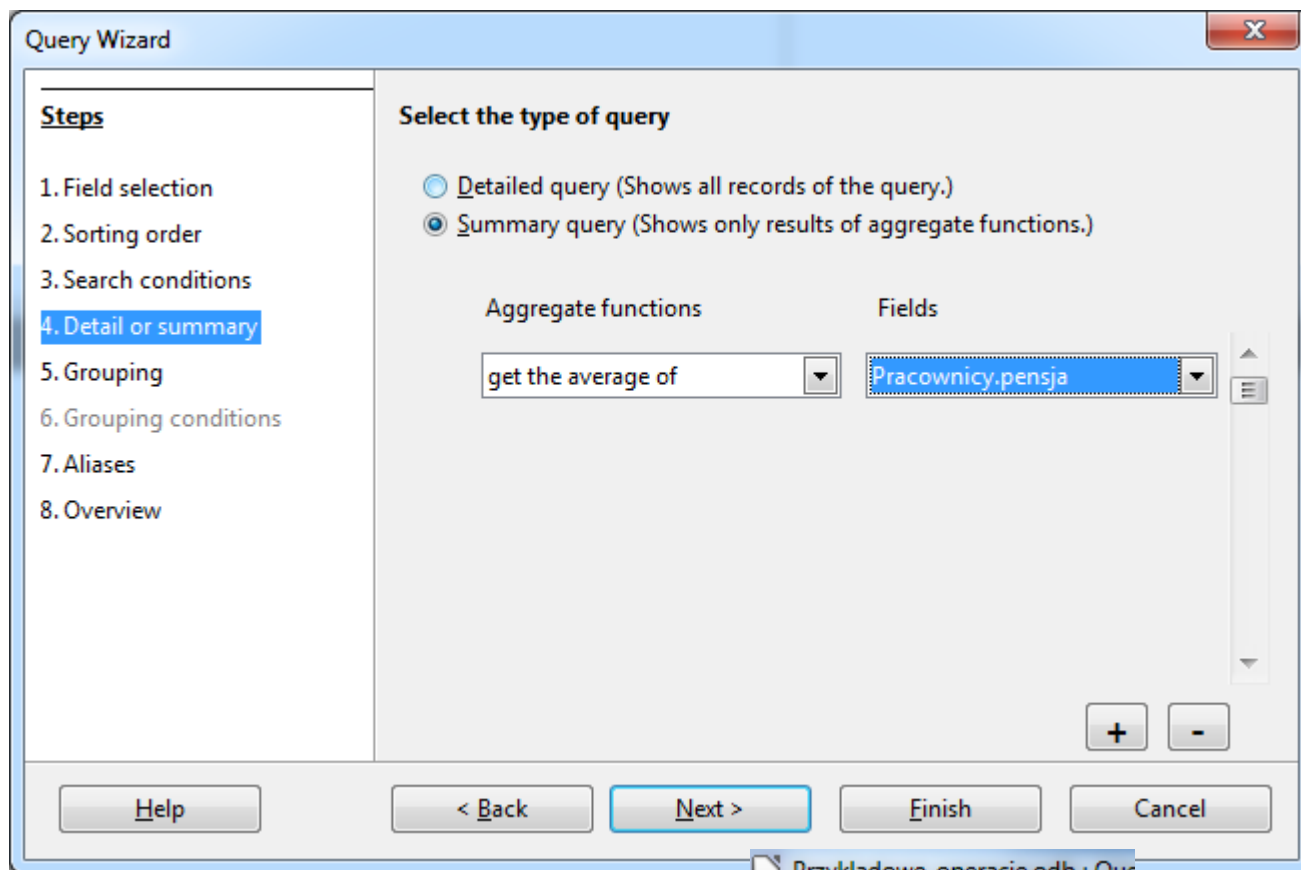
Fields	Condition	Value
Pracownicy.pensja	is greater than	3000
Pracownicy.Imie	like	E*
	is equal to	

Help < Back Next > Finish Cancel

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

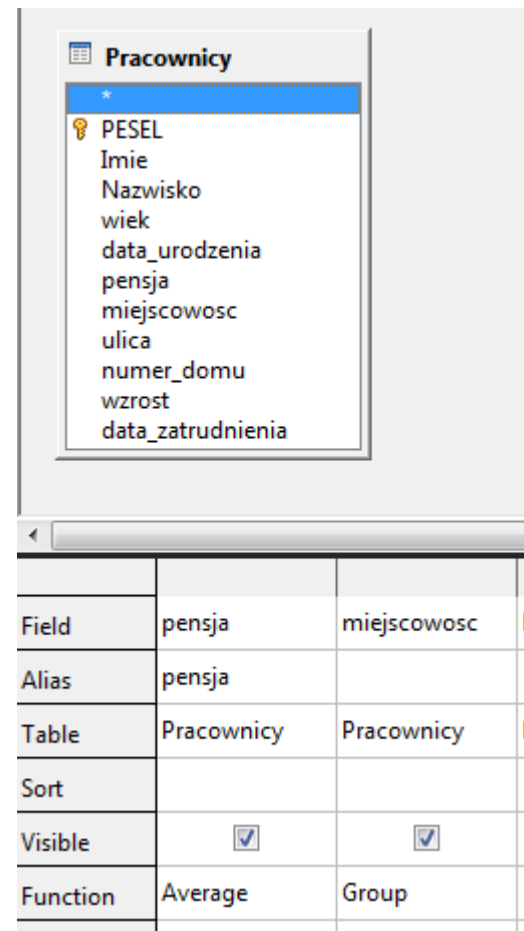
	Imie	Nazwisko	pensja	data_zatrudnienia
▶	Ela	Nowak	5500,00	15.04.2012
	Ewa	Nowak	4000,00	15.04.2011

# Przykładowe kwerendy w LibreOffice



wizzard

## GRUPOWANIE



projekt

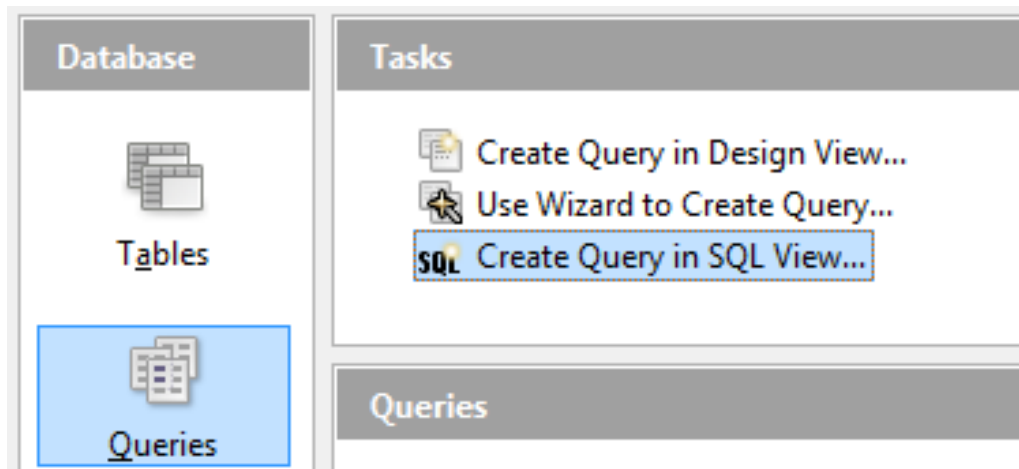
Przykladowe\_operacje.odt : Que

	pensja	miejscowosc
▶	3500	Rzeszów
	4750	Kraków
	5500	Warszawa

wynik

# Przykładowe kwerendy w LibreOffice

## GRUPOWANIE



Trzecia metoda tworzenia kwerend

