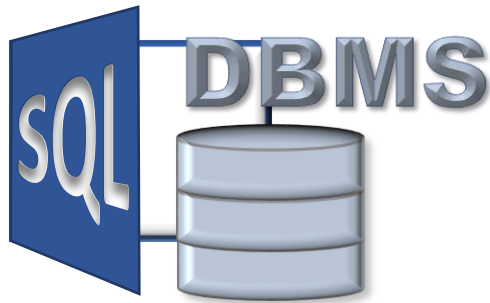


# Bazy danych



**Bazy danych – wprowadzenie**  
pojęcia podstawowe  
modelowanie i modele danych  
architektura systemów baz danych

**Przeszukiwanie danych**  
algebra relacji  
podstawy formułowania zapytań w SQL


## Język SQL

- ❑ L. Beighley, *SQL*, Helion, Gliwice 2011.
- ❑ M.J. Hernandez, *Zapytania SQL dla zwykłych śmiertelników : praktyka obróbki danych w języku SQL*, Mikom, 2001.
- ❑ L. Rockoff, *Język SQL : przyjazny podręcznik*, Helion, 2017.
- ❑ R.K. Stephens, R.R. Plew, B. Morgan, J. Perkins, *SQL w 3 tygodnie*, LT&P, 1999.
- ❑ R.K. Stephens, A. Jones, R.R. Plew, *SQL w 24 godziny*, Helion, 2016.


## Bazy danych

- ❑ P. Beynon-Davies, *Systemy baz danych*, WNT, 2003.
- ❑ M.J. Hernandez, *Bazy danych dla zwykłych śmiertelników*, Mikom, 2000.
- ❑ J.D.Ullman, J.Widom, *Podstawowy wykład z systemów baz danych*, WNT, 2001.
- ❑ M. Whitehorn, *Relacyjne bazy danych : teoria i praktyka projektowania relacyjnego baz danych*, Helion, 2011.

Dostęp do zasobów Springer'a z wyszukiwarki EBSCO (<http://www.bu.uz.zgora.pl>)




BIBLIOTEKA UNIWERSYTETU ZIELONOGÓRSKIEGO



Multiwyszukiwarka E-czasopisma **E-książki** Przejdź do listy A-Z

Przeszukuj tytuły książek elektronicznych

 Full Text Finder

Szukaj

Instrukcja konfiguracji przeglądarki: <https://www.ck.uz.zgora.pl/index.php?w3cacheproxy>

- ❑ J. R. Russo, *SQL by Example*, New York : Momentum Press. 2018.
- ❑ A. Kriegel, *Discovering SQL : A Hands-On Guide for Beginners*, Ind: Wrox., 2011.
- ❑ J.L. Harrington, *SQL Clearly Explained*, 2nd ed., Morgan Kaufmann, 2003.
- ❑ J.L. Harrington, *Relational Database Design and Implementation: Clearly Explained*, 3rd ed. Amsterdam : Morgan Kaufmann. 2009.

# Bazy danych – pojęcia podstawowe

---

**System informatyczny** – zespół systemów komputerowych (komputery, urządzenia peryferyjne) i oprogramowania służący do przetwarzania informacji.

**Baza danych** – uporządkowany zbiór danych, dostępnych dla licznych użytkowników, w którym można przeprowadzić efektywne wyszukiwanie i aktualizowanie informacji.

**System zarządzania bazą danych (SZBD, *ang.* DBMS)** – program zawierający narzędzia umożliwiające budowę i przetwarzanie bazy danych o dowolnej strukturze.

**System bazy danych** – baza danych wraz z jej systemem zarządzania.

## Architektura systemu baz danych

- Poziom fizyczny – pliki dyskowe i algorytmy zapewniające dostęp do danych,
- Poziom logiczny – konceptualny model bazy danych, obejmuje całą zawartość bazy,
- Poziom zewnętrzny (użytkowy) – zbiór widoków poprzez które użytkownicy uzyskują dostęp do odpowiednich fragmentów bazy danych.

## Przykład – analiza zadania

---

**Zadanie:** Należy zaprojektować system informatyczny (bazę danych) dla potrzeb placówki medycznej, który umożliwi gromadzenie i przetwarzanie danych skierowań na badania wystawianych przez lekarzy rodzinnych swoim pacjentom.

**Przeznaczenie:** Aplikacja przetwarza dane skierowań na badania.

### Funkcje:

- Przechowuje dane osobowe pacjentów.
- Przechowuje dane lekarzy.
- Przechowuje dane skierowań na badania.
- Tworzy statystyki (np. skierowania pacjenta w określonym przedziale czasowym).

**Schemat opisowy:** Lekarz jest identyfikowany przez numer prawa wykonywania zawodu, ma przypisane imię i nazwisko. Pacjent jest identyfikowany przez numer PESEL, ma imię i nazwisko, jest przypisany do jednego lekarza POZ. Każde badanie ma unikalny kod (ICD-9) nazwę kategorii oraz cenę. Skierowanie zawiera informacje o dacie wystawienia, kodzie badania oraz identyfikator pacjenta i lekarza kierującego.

**Modelowanie danych** – proces tworzenia odpowiedniego odwzorowania danych dla potrzeb systemu informatycznego, polega na przejściu od swobodnego opisu fragmentu rzeczywistości (modelowanego systemu) do jego formalnej reprezentacji.

## Elementy świata rzeczywistego

- ❑ **obiekt** – składnik rzeczywistego systemu postrzegany jako istotny przez jednostkę lub grupę, przyszłych użytkowników bazy danych,
- ❑ **powiązanie** – opis stanu, w którym znalazły się co najmniej dwa obiekty.

Obiekty i powiązania mogą być dodatkowo opisane za pomocą atrybutów.

## Przykład

- **obiekt**: pacjent, **atrybuty**: PESEL, nazwisko, imię,
- **obiekt**: lekarz, **atrybuty**: PWZ (nr prawa wykonywania zawodu), nazwisko, imię,
- **obiekt**: badanie, **atrybuty**: ICD9 (kod badania), nazwa kategorii, cena,
- **powiązanie**: pacjent-lekarz,
- **powiązanie**: pacjent-lekarz-badanie, **atrybut**: data wystawienia.

**Model danych** – dobrze zdefiniowany sposób opisu świata rzeczywistego.

## Najważniejsze modele danych

- ❑ **kartotekowy** – dane zapisywane są w pojedynczej tablicy, każda tablica jest samodzielnym dokumentem i nie może współpracować z innymi tablicami,
- ❑ **sieciowy** – oparty na strukturze grafu, obiekty są reprezentowane przez wierzchołki grafu, powiązania przez jego krawędzie,
- ❑ **hierarchiczny** – modyfikacja modelu sieciowego, oparty na strukturze drzewa (graf, który nie zawiera cykli),
- ❑ **relacyjny** – wykorzystuje matematyczne pojęcie relacji, przy jej pomocy reprezentuje zarówno obiekty jak i powiązania,
- ❑ **obiektyowy** – wprowadza rozszerzone pojęcie obiektu (zgodne z podejściem stosowanym w programowaniu obiektowym), w którym uwzględniane są atrybuty (reprezentujące cechy) oraz metody (reprezentujące zachowania/umiejętności) rzeczywistych obiektów występujących w modelowanym systemie.

# Model relacyjny – definicje (1)

**Relacją**  $R$  na zbiorach  $D_1, D_2, \dots, D_n$  nazywamy dowolny podzbiór iloczynu kartezyjskiego tych zbiorów i zapisujemy:

$$R(D_1, D_2, \dots, D_n), R \subset D_1 \times D_2 \times \dots \times D_n.$$

Zapis postaci  $R(D_1, D_2, \dots, D_n)$  jest **schematem relacji**,  $R$  nazwą relacji, a elementy  $D_1, D_2, \dots, D_n$  **atrybutami** lub **składnikami relacji**.

**Krotka relacji** – ciąg wartości atrybutów danego schematu relacji.

## Alternatywna definicja relacji

Relacją  $R$  na zbiorach  $D_1, D_2, \dots, D_n$  nazywamy dowolny zbiór krotek postaci:

$$\langle d_1, d_2, \dots, d_n \rangle$$

takich, że:  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$ .

## Przykład

Schemat relacji *Pacjent*:

*Pacjent(PESEL, Nazwisko, Imię, Adres)*

Krotka relacji *Pacjent*:

*<70052000234, Kowalski, Jan, 65-516 Zielona Góra Podgórna 50>*



## Model relacyjny – definicje (2)

---

**Identyfikator relacji** – atrybut lub ciąg atrybutów, których wartości określają w sposób jednoznaczny krotkę relacji.

**Identyfikator kluczowy (klucz) relacji** – jeden, dowolnie wybrany identyfikator relacji (zazwyczaj kryterium wyboru jest długość). Klucz w schemacie relacji jest zaznaczany przez podkreślenie odpowiednich atrybutów.

**Klucz naturalny** – klucz złożony z atrybutów relacji, których obecność wynika z przeprowadzonej analizy problemu.

**Klucz sztuczny** – sztucznie wprowadzony atrybut relacji, którego wartości gwarantują jednoznaczną identyfikację krotek. W roli klucza sztucznego najczęściej występuje liczba porządkowa.

### Przykład

Klucz naturalny: *Pacjent(PESEL, Nazwisko, Imię, Adres)*

Klucz sztuczny: *Pacjent(IDP, PESEL, Nazwisko, Imię, Adres)*

*IDP* – identyfikator pacjenta, unikalna wartość liczbowa.

**Powiązanie (związek)** – opis stanu, który dotyczy dwóch lub więcej obiektów.

### Typy powiązań

- jeden-do-jeden – krotce z relacji pierwszej odpowiada dokładnie jedna krotka z relacji drugiej i odwrotnie (zazwyczaj eliminowane z modelu danych),
- jeden-do-wielu – krotce z relacji pierwszej odpowiada wiele krotek z relacji drugiej, ale jednej krotce z relacji drugiej odpowiada jedna krotka w relacji pierwszej,
- wiele-do-wielu – krotce z relacji pierwszej odpowiada wiele krotek z relacji drugiej i krotce z relacji drugiej odpowiada wiele krotek z relacji pierwszej.

**Klucz obcy** – atrybut lub ciąg atrybutów, który występuje w pewnej relacji i jednocześnie jest identyfikatorem kluczowym innej relacji. Tworzy powiązanie pomiędzy relacjami.

### Przykład

*Lekarz(PWZ, NazwiskoL, ImięL)*

*Pacjent(PESEL, NazwiskoP, ImięP, PWZ)*

*PWZ* jest kluczem obcym w relacji *Pacjent*, tworzy powiązanie typu jeden-do-wielu pomiędzy *Pacjentem* i *Lekarzem* (np. lekarz rodzinny danego pacjenta).

# Przykład – relacyjny model danych

---

**Obiekty:** pacjent, lekarz, badanie.

**Powiązania:** pacjent-lekarz, pacjent-lekarz-badanie.

## Relacje reprezentujące obiekty

$P(\underline{PESEL}, \text{NazwiskoP}, \text{ImięP})$

$L(\underline{PWZ}, \text{NazwiskoL}, \text{ImięL})$

$B(\underline{ICD9}, \text{Kategoria}, \text{Cena})$

## Relacje reprezentujące powiązania

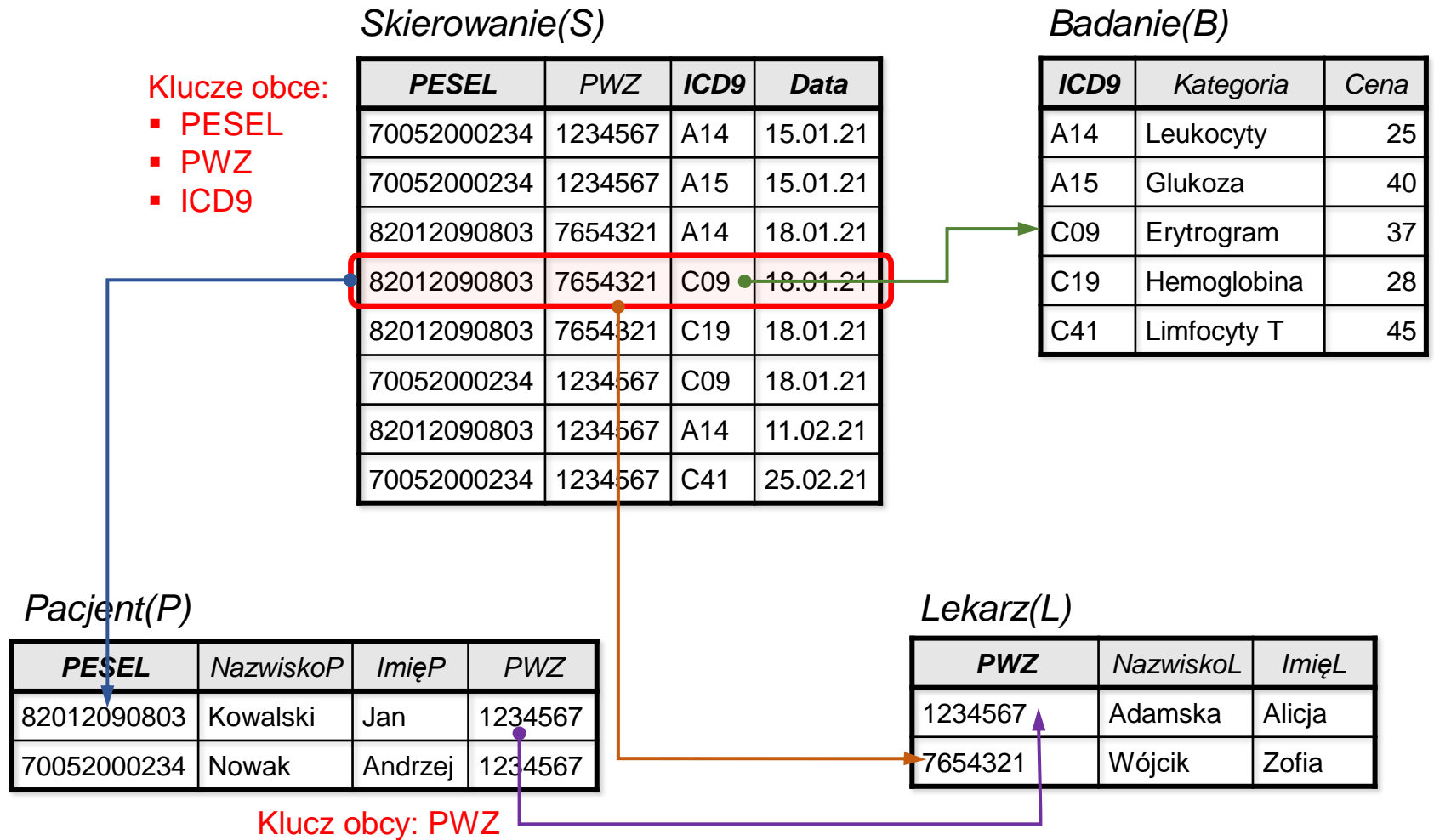
- pacjent-lekarz – jeden pacjent ma dokładnie jednego lekarza, dodatkowa relacja nie jest wymagana, powiązanie jest realizowane przez uzupełnienie danych pacjenta o identyfikator lekarza:

$P(\underline{PESEL}, \text{NazwiskoP}, \text{ImięP}, \text{PWZ})$

- pacjent-lekarz-badanie – jeden pacjent ma wiele badań, jeden lekarz wystawia wiele skierowań, to samo badanie występuje na wielu skierowaniach (dotyczy wielu pacjentów, jest zlecane przez wielu lekarzy), konieczna dodatkowa relacja:

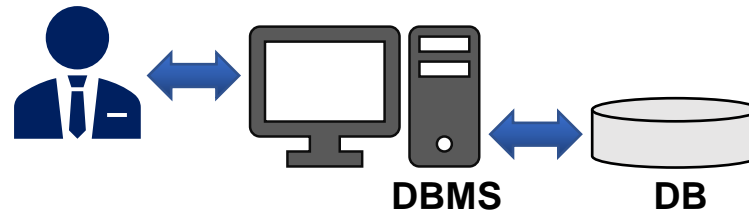
$S(\underline{PESEL}, \text{PWZ}, \underline{ICD9}, \underline{Data})$

# Przykładowy zestaw danych

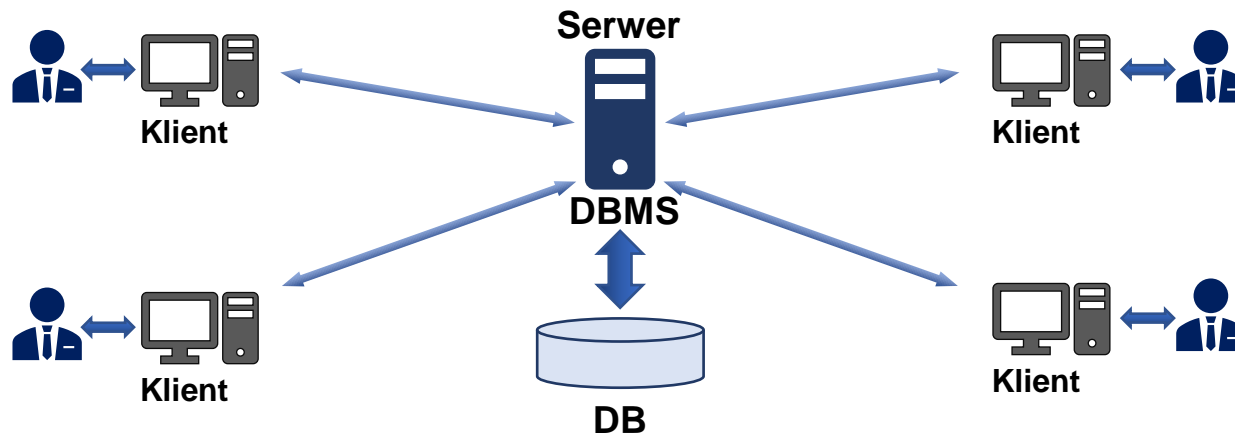


# Architektura systemów baz danych

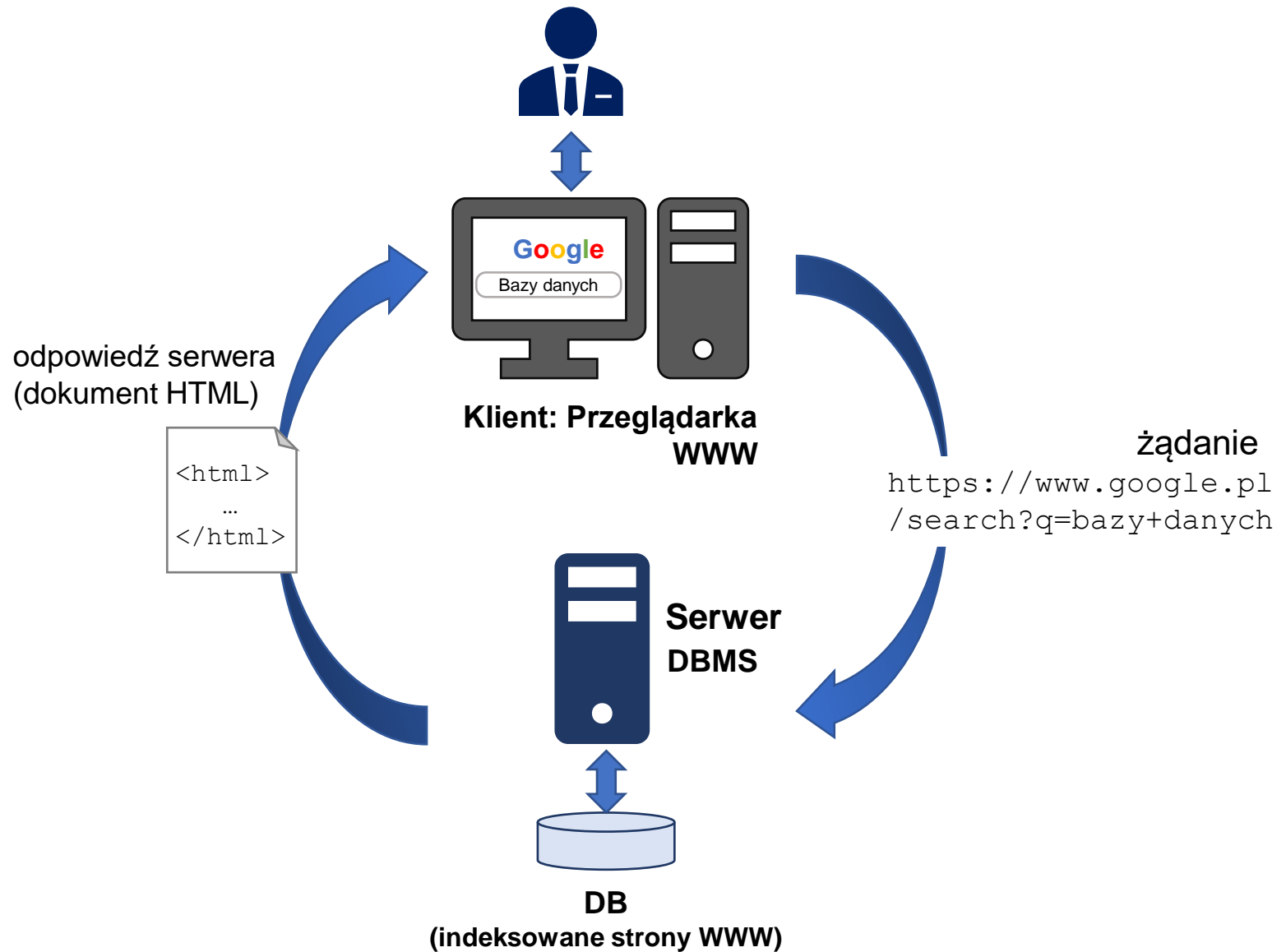
**Lokalna baza danych** – rodzaj bazy danych do użytku indywidualnego, system zarządzania bazą danych oraz zbiór danych, który przetwarza znajdują się na jednym komputerze, dane nie są dostępne z innych jednostek.



**Baza klient-serwer** – składa się z wyróżnionego węzła nazywanego serwerem i szeregu podłączonych do niego węzłów nazywanych klientami. Serwer zawiera bazę danych i system zarządzania, i wykonuje usługi zlecane przez klientów (przetwarzanie danych). Rolą klienta jest przesłanie żądania i oczekiwanie na odpowiedź serwera. Architektura klient-serwer pozwala na współużytkowanie bazy przez wielu użytkowników (klientów).



# Przykład – wyszukiwarka Google



The screenshot displays the phpMyAdmin interface for a database named 'skierowania' on a local server. The top navigation bar includes options like 'Struktura', 'SQL', 'Szukaj', 'Zapytanie', 'Eksport', 'Import', 'Operacje', 'Procedury i funkcje', 'Zdarzenia', and 'Więcej'. The left sidebar shows a tree view of the database structure, including 'Nowa', 'information\_schema', and 'skierowania' with sub-tables 'B', 'L', 'P', and 'S'.

The main area shows a table list with the following data:

Tabela	Działanie	Rekordy	Typ	Metoda porównywania napisów	Rozmiar	Nadmiar
<input type="checkbox"/> B	Przeglądaj	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
<input type="checkbox"/> L	Przeglądaj	2	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
<input type="checkbox"/> P	Przeglądaj	4	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
<input type="checkbox"/> S	Przeglądaj	8	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
<b>4 tabel</b>	<b>Suma</b>	<b>19</b>	<b>InnoDB</b>	<b>utf8mb4_0900_ai_ci</b>	<b>64.0 KB</b>	<b>0 B</b>

The bottom window shows the SQL query editor for table 'P'. The query is:

```

1 SELECT *
2 FROM P INNER JOIN S ON P.PESEL=S.PESEL
3 WHERE S.ICD9 = 'A14'
    
```

The column list on the right includes: PESEL, NazwiskoP, ImieP, and PWZ. The interface also features buttons for 'SELECT \*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Wyczyść', and 'Format', along with a 'Wykonaj' (Execute) button.

*phpMyAdmin w trybie podglądu struktury bazy i wprowadzania kwerendy SQL*

**Język manipulowania danymi** (*ang. DML, Data Manipulation Language*) – język udostępniający operacje niezbędne do zarządzania zbiorem danych (tworzenie, aktualizacja, usuwanie, przeszukiwanie, itp.).

**Przeszukiwanie danych** – podstawowa operacja wykonywana w bazie danych, poszukiwanie informacji spełniających określone przez użytkownika kryteria.

**Język zapytań** (*ang. query language*) – język programowania umożliwiający formułowanie zapytań, na podstawie których system zarządzania bazą danych wykonuje przeszukiwanie danych.

**Abstrakcyjne języki zapytań** definiujące zestaw podstawowych operacji dla relacyjnego modelu danych:

- język algebraiczny – notacja wykorzystująca operacje teoriomnogościowe,
- datalog (*ang. database logic*) – notacja oparta na rachunku predykatów.

Obydwie notacje są równoważne i definiują pełny zestaw operacji dla języka zapytań (gwarantują znalezienie dowolnej informacji o ile znajduje się ona w bazie danych).



**Algebra relacji** – zestaw operacji, które umożliwiają tworzenie nowych (zazwyczaj bardziej skomplikowanych) relacji na podstawie relacji istniejących w bazie danych.

## Kategorie operatorów

- operacje teoriomnogościowe: suma, przecięcie, różnica zbiorów (używane w szczególnych przypadkach, nie będą omawiane),
- operacje zawężające: selekcja – eliminacja wybranych krotek (wierszy) relacji, rzutowanie – eliminacja wybranych kolumn relacji,
- operacje komponowania krotek z różnych relacji: iloczyn kartezjański – wszystkie możliwe kombinacje par krotek z dwóch relacji, złączenie – selektywne łączenie par krotek na podstawie pewnego warunku,
- operacje przemianowania – zmiana schematu relacji, czyli jej nazwy lub nazw atrybutów, bez wpływu na krotki relacji.

*Uwaga:* Zestaw operatorów algebry relacji nie dostarcza wszystkich możliwych działań niezbędnych do stworzenia pełnego języka manipulowania danymi, jest jednak pełnym zestawem operatorów dla języka zapytań.

# Operator rzutowania (projekcji)

**Operacja rzutowania** tworzy nową relację, która powstaje z relacji pierwotnej poprzez wybór atrybutów (kolumn):

$$\Pi_{A_1 \dots A_n} R(A_1, \dots, A_m) = R'(A_1, \dots, A_n),$$

gdzie  $A_1, \dots, A_n$  jest pewnym podzbiorem atrybutów relacji  $R$ .

Wynikiem operacji rzutowania jest relacja zawierająca wymienione kolumny, występujące w zadanej kolejności i wszystkie krotki relacji pierwotnej.

**Przykład 1.** Imiona i nazwiska wszystkich pacjentów

$$\Pi_{ImięP, NazwiskoP} P(\underline{PESEL}, NazwiskoP, ImięP, PWZ) = P'(ImięP, NazwiskoP)$$

**Przykład 2.** Kategorie wszystkich badań


$$\Pi_{Kategoria} B(\underline{ICD9}, Kategoria, Cena) = B'(Kategoria)$$

# Operator rzutowania – przykłady

**Przykład 1.** Imiona i nazwiska wszystkich pacjentów

$\Pi_{\text{ImięP, NazwiskoP}} P(\text{PESEL}, \text{NazwiskoP}, \text{ImięP}, \text{PWZ}) = P'(\text{ImięP}, \text{NazwiskoP})$

PESEL	NazwiskoP	ImięP	PWZ
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321




NazwiskoP	ImięP
Kowalski	Jan
Nowak	Andrzej
Wiśniewska	Anna
Kowalczyk	Julia

**Przykład 2.** Kategorie wszystkich badań

$\Pi_{\text{Kategoria}} B(\text{ICD9}, \text{Kategoria}, \text{Cena}) = B'(\text{Kategoria})$

ICD9	Kategoria	Cena
A14	Leukocyty	25
A15	Glukoza	40
C09	Erytrogram	37
C19	Hemoglobina	28
C41	Limfocyty T	45



Kategoria
Leukocyty
Glukoza
Erytrogram
Hemoglobina
Limfocyty T

**Operacja selekcji** tworzy nową relację, która powstaje z relacji pierwotnej poprzez wybór wierszy spełniających określone kryterium:

$$\Sigma_w R(A_1, \dots, A_n) = R'(A_1, \dots, A_n),$$

gdzie  $w$  to kryterium utworzone przy pomocy operatorów relacyjnych (=, >, >=, <, <=, <>) oraz logicznych (AND, OR, NOT); relacja  $R'$  zawiera krotki spełniające kryterium  $w$ .

Wynikiem operacji selekcji jest relacja o identycznym schemacie, ale zawierająca tylko pewien podzbiór krotek relacji pierwotnej.

**Przykład 1.** Pacjenci lekarza 1234567

$$\Sigma_{PWZ = 1234567} P(\underline{PESEL}, \underline{NazwiskoP}, \underline{ImięP}, PWZ)$$

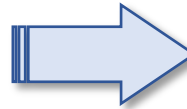
**Przykład 2.** Skierowania na badanie A14

$$\Sigma_{ICD9 = A14} S(\underline{PESEL}, \underline{ICD9}, \underline{Data}, PWZ)$$

# Operator selekcji – przykłady

**Przykład 1.** Pacjenci lekarza 1234567:  $\Sigma_{PWZ = 1234567} P(\underline{PESEL}, \underline{NazwiskoP}, \underline{ImięP}, \underline{PWZ})$

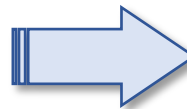
PESEL	NazwiskoP	ImięP	PWZ
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321



PESEL	NazwiskoP	ImięP	PWZ
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567

**Przykład 2.** Skierowania na badanie A14:  $\Sigma_{ICD9 = A14} S(\underline{PESEL}, \underline{ICD9}, \underline{Data}, \underline{PWZ})$

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
70052000234	A15	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	C09	18.01.21	7654321
82012090803	C19	18.01.21	7654321
70052000234	C09	18.01.21	1234567
82012090803	A14	11.02.21	1234567
70052000234	C41	25.02.21	1234567



PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	A14	11.02.21	1234567

**Iloczyn kartezjański** relacji  $R$  i  $S$  tworzy nową relację  $T$  zawierającą wszystkie uporządkowane pary krotek, z których pierwsza należy do  $R$ , a druga do  $S$ :

$$R(A_1, \dots, A_n) \times S(B_1, \dots, B_m) = T(A_1, \dots, A_n, B_1, \dots, B_m)$$

Wynikiem iloczynu skalarnego jest nowa relacja, której schemat jest sumą schematów relacji pierwotnych, zawierająca wszystkie możliwe pary krotek (zawiera  $x \cdot y$  krotek, gdzie  $x$  i  $y$  to liczba krotek odpowiednio relacji  $R$  i  $S$ ).

*Uwaga 1:* jeżeli w  $R$  i  $S$  występują atrybuty o tej samej nazwie, to w  $T$  zostaną poprzedzone nazwą relacji. Np. jeżeli w  $R$  i  $S$  istnieje atrybut  $A$ , to  $T$  postanie  $R.A$  i  $S.A$ .

*Uwaga 2:* iloczyn kartezjański należy do operacji, które nie są używane w praktyce, stanowi jednak podstawę do realizacji operatorów złączenia (s. 24 i 26).

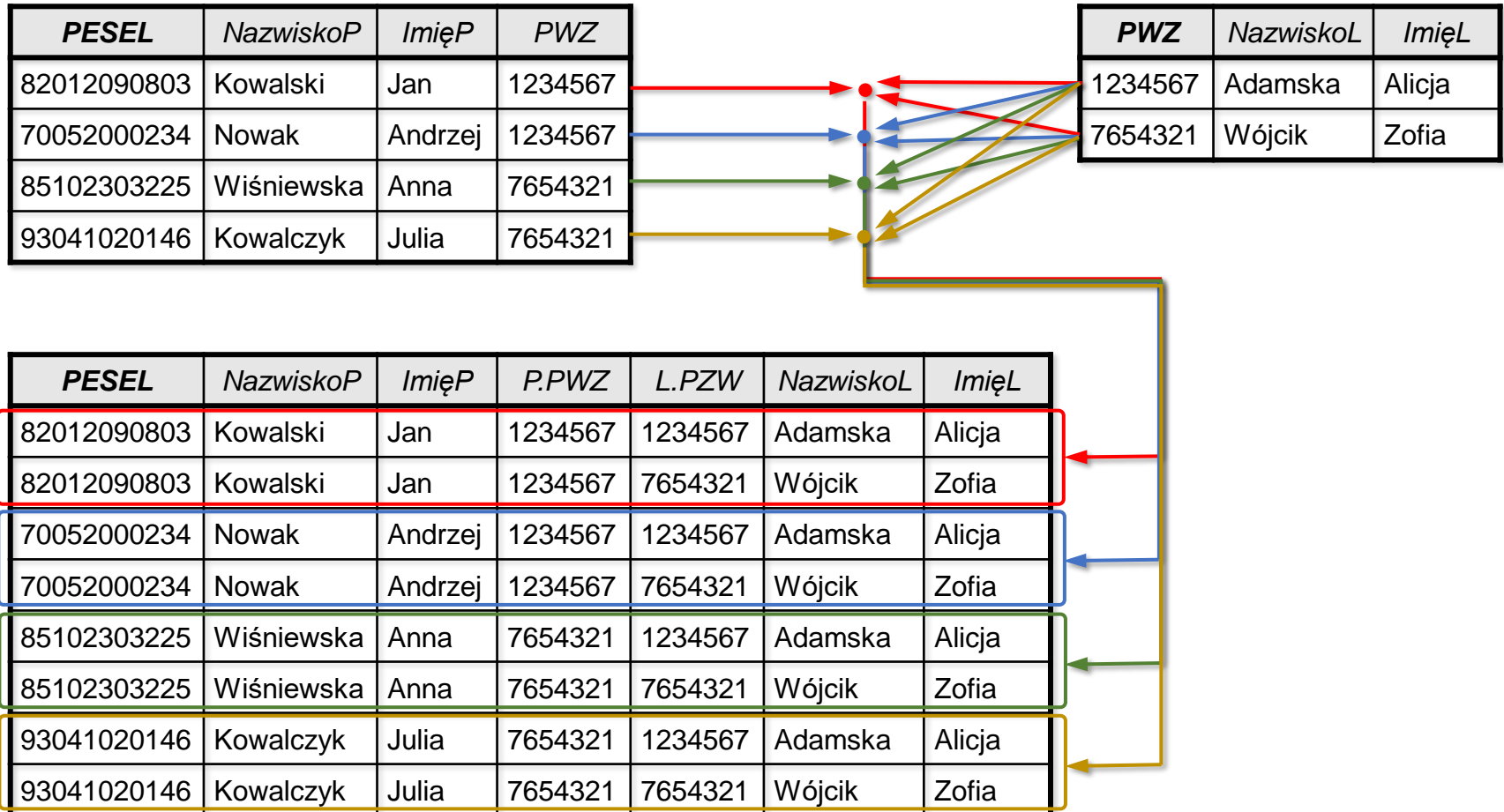
**Przykład** Iloczyn kartezjański relacji Pacjent i Lekarz

$$P(\underline{PESEL}, \text{NazwiskoP}, \text{ImięP}, \text{PWZ}) \times L(\underline{PWZ}, \text{NazwiskoL}, \text{ImięL})$$

# Iloczyn kartezyński – przykład

**Przykład** Iloczyn kartezyński relacji Pacjent i Lekarz

$P(\underline{PESEL}, \text{NazwiskoP}, \text{ImięP}, \text{PWZ}) \times L(\underline{PWZ}, \text{NazwiskoL}, \text{ImięL})$



**Złączenie naturalne** relacji  $R$  i  $S$  tworzy nową relację  $T$  zawierającą pary krotek, które mają identyczne wartości atrybutów wspólnych dla relacji  $R$  i  $S$ :

$$R(A_1, \dots, A_n, A_{n+1}, \dots, A_{n+m}) \otimes S(A_1, \dots, A_n, B_1, \dots, B_p) = T(A_1, \dots, A_{n+m}, B_1, \dots, B_p),$$

gdzie  $A_1, \dots, A_n$  to atrybuty występujące jednocześnie w  $R$  i  $S$ .

Wynikiem złączenia naturalnego jest nowa relacja, której schemat jest sumą schematów relacji pierwotnych (bez powtórzeń), zawierająca tylko te pary krotek, dla których atrybuty występujące jednocześnie w relacjach  $R$  i  $S$  mają takie same wartości.

*Uwaga 1:* relacja wynikowa ma takie same wartości dla atrybutów wspólnych w relacjach pierwotnych, więc jej schemat nie zawiera powtórzeń tych atrybutów.

*Uwaga 2:* złączenie naturalne można interpretować jako rezultat selekcji wykonanej na iloczynie kartezjańskim relacji  $R$  i  $S$  z warunkiem określającym równość atrybutów wspólnych.

**Przykład** Złączenie naturalne relacji Pacjent i Lekarz

$$P(\underline{PESEL}, \text{Nazwisko}_P, \text{Imię}_P, \text{PWZ}) \otimes L(\underline{PWZ}, \text{Nazwisko}_L, \text{Imię}_L)$$



# Złączenie naturalne – przykład

**Przykład** Złączenie naturalne relacji Pacjent i Lekarz

$P(\underline{PESEL}, \text{Nazwisko}_P, \text{Imię}_P, \text{PWZ}) \otimes L(\underline{PWZ}, \text{Nazwisko}_L, \text{Imię}_L)$

<i>PESEL</i>	<i>Nazwisko<sub>P</sub></i>	<i>Imię<sub>P</sub></i>	<i>PWZ</i>
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321

<i>PWZ</i>	<i>Nazwisko<sub>L</sub></i>	<i>Imię<sub>L</sub></i>
1234567	Adamska	Alicja
7654321	Wójcik	Zofia

<i>PESEL</i>	<i>Nazwisko<sub>P</sub></i>	<i>Imię<sub>P</sub></i>	<i>PWZ</i>	<i>Nazwisko<sub>L</sub></i>	<i>Imię<sub>L</sub></i>
82012090803	Kowalski	Jan	1234567	Adamska	Alicja
70052000234	Nowak	Andrzej	1234567	Adamska	Alicja
85102303225	Wiśniewska	Anna	7654321	Wójcik	Zofia
93041020146	Kowalczyk	Julia	7654321	Wójcik	Zofia

**Złączenie teta** relacji  $R$  i  $S$  tworzy nową relację  $T$ , zawierającą pary krotek z relacji  $R$  i  $S$ , dla których spełnione jest pewne kryterium:

$$R(A_1, \dots, A_n) \otimes_w S(B_1, \dots, B_m) = T(A_1, \dots, A_n, B_1, \dots, B_m),$$

gdzie  $w$  jest kryterium konstruowanym analogicznie do operatora selekcji.

Wynikiem złączenia teta jest nowa relacja, której schemat jest sumą schematów relacji pierwotnych, zawierająca tylko te pary krotek, dla których spełniony jest warunek  $w$ .

*Uwaga 1:* wartości łączonych atrybutów mogą być różne, stąd w schemacie relacji wynikowej występują wszystkie atrybuty relacji pierwotnych.

*Uwaga 2:* złączenie teta można interpretować jako rezultat selekcji wykonanej na iloczynie kartezyjskim relacji  $R$  i  $S$  z dowolnie określonym kryterium.

**Przykład** Złączenie teta relacji Pacjent i Lekarz z wyborem pacjentów lekarza 1234567

$$P(\underline{PESEL}, \text{NazwiskoP}, \text{ImięP}, \text{PWZ}) \otimes L(\underline{PWZ}, \text{NazwiskoL}, \text{ImięL})$$

$$P.PWZ = L.PWZ \text{ AND } P.PWZ = 1234567$$

# Złączenie teta – przykład

**Przykład** Złączenie teta relacji Pacjent i Lekarz

$P(\underline{PESEL}, \text{NazwiskoP}, \text{ImięP}, \text{PWZ}) \otimes L(\underline{PWZ}, \text{NazwiskoL}, \text{ImięL})$

$P.PWZ = L.PWZ$  AND  $P.PWZ = 1234567$

<i>PESEL</i>	<i>NazwiskoP</i>	<i>ImięP</i>	<i>PWZ</i>
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321

<i>PWZ</i>	<i>NazwiskoL</i>	<i>ImięL</i>
1234567	Adamska	Alicja
7654321	Wójcik	Zofia



<i>PESEL</i>	<i>NazwiskoP</i>	<i>ImięP</i>	<i>P.PWZ</i>	<i>L.PZW</i>	<i>NazwiskoL</i>	<i>ImięL</i>
82012090803	Kowalski	Jan	1234567	1234567	Adamska	Alicja
70052000234	Nowak	Andrzej	1234567	1234567	Adamska	Alicja

**Operacja przemianowania** zmienia nazwę relacji i/lub jej atrybutów:

$$P_{S(B_1 \dots B_n)} R(A_1, \dots, A_n) = S(B_1, \dots, B_n)$$

$$P_S R(A_1, \dots, A_n) = S(A_1, \dots, A_n)$$

Wynikiem operacji przemianowania jest nowa relacja o takich samych krotkach i takiej samej liczbie atrybutów jak relacja pierwotna, ale o innej nazwie i/lub zmienionych nazwach atrybutów.

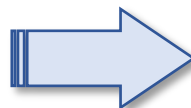
Operator przemianowania pozwala na zwiększenie czytelności wyniku kwerendy.

**Przykład** Przemianowanie relacji *B*

$$P_{Badania(Kod\ ICD9,\ Kategorie,\ Cena)}(B(\underline{ICD9},\ Kategorie,\ Cena)) = Badania(Kod\ ICD9,\ Kategorie,\ Cena)$$

*B(ICD9, Kategorie, Cena)*

<i>ICD9</i>	<i>Kategoria</i>	<i>Cena</i>
A14	Leukocyty	25
A15	Glukoza	40
C09	Erytrogram	37
C19	Hemoglobina	28
C41	Limfocyty T	45



*Badania(Kod ICD9, Kategorie, Cena)*

<i>Kod ICD9</i>	<i>Kategoria</i>	<i>Cena</i>
A14	Leukocyty	25
A15	Glukoza	40
C09	Erytrogram	37
C19	Hemoglobina	28
C41	Limfocyty T	45

# Tworzenie zapytań złożonych

Algebra relacji umożliwia tworzenie dowolnie złożonych wyrażeń poprzez zastosowanie operatorów do relacji będących wynikiem działania innych operatorów (złożenie operatorów). Kolejność działań jest ustalana przez zastosowanie nawiasów.

Wyrażenia zbudowane z pełnego zestawu operatorów pozwalają na skonstruowanie zapytania gwarantującego znalezienie dowolnej informacji o ile znajduje się ona w bazie danych.

## Przykład 1.

Daty badania A14 (*PESEL*, *PWZ*, *Data*)

$$\Pi_{PESEL, PWZ, Data} (\Sigma_{ICD9=A14} S)$$

## Przykład 2.

Dane pacjentów, który mieli wykonane badania 18.01.2021 (*ImięP*, *NazwiskoP*, *ICD9*)

$$\Pi_{ImięP, NazwiskoP, ICD9} (\Sigma_{Data=18.01.2021} (P \otimes S))$$

$$\Pi_{ImięP, NazwiskoP, ICD9} (P \otimes \Sigma_{Data=18.01.2021} (S))$$

**SQL** (*ang. Structured Query Language*, strukturalny język zapytań) – język manipulowania danymi utworzony przez firmę IBM. Obowiązuje jako standardowe narzędzie w systemach zarządzania relacyjnymi bazami danych. Zestaw poleceń związanych z przeszukiwaniem danych jest oparty na algebrze relacji.

SQL jest podjęzykiem danych. Może być wykorzystany wyłącznie do komunikacji z bazą danych, nie ma cech pozwalających na tworzenie kompletnych programów.

## Standardy SQL

- SQL-86 – pierwszy oficjalny standard języka wspierany przez Międzynarodową Organizację Normalizacyjną (ISO) oraz Amerykański Narodowy Instytut Normalizacji (ANSI). Określał wspólną płaszczyznę łączącą produkty różnych producentów, pozostawiał dużą swobodę twórcom implementacji.
- SQL-89 – drobna (minor) modyfikacja standardu 86, brak pełnej zgodności pomiędzy produktami różnych producentów.
- SQL-92 – precyzyjna specyfikacja składni, zapewnienie zgodności pomiędzy różnymi produktami, standard obowiązujący do dziś.
- Kolejne standardy: SQL:1999 (SQL3-przestarzały), SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016 – nowe typy danych, funkcje, itp.

**DQL** (*Data Query Language*) – formułowanie zapytań

- **SELECT** – operacje rzutowania selekcji i złączenia.

**DDL** (*Data Definition Language*) – operacje na strukturach danych

- **CREATE** – tworzenie struktury (bazy danych, tabeli),
- **DROP** – usunięcie struktury,
- **ALTER** – zmiana struktury danych.

**DML** (*Data Manipulation Language*) – operacje na danych

- **INSERT** – umieszcza dane w bazie,
- **UPDATE** – aktualizuje dane,
- **DELETE** – usuwa dane z bazy.

**DCL** (*Data Control Language*) – kontrola uprawnień

- **GRANT** – nadawanie uprawnień do obiektów bazy danych,
- **REVOKE** – odbieranie uprawnień,
- **DENY** – blokowanie możliwości wykonania określonych operacji.

## Podstawowa składnia zapytania w języku SQL

```
SELECT atrybuty
FROM   źródło_danych
WHERE  warunek;
```

- *atrybuty* to rozdzielona przecinkami lista atrybutów, których wartości są zwracane jako wynik zapytania;
- *źródło\_danych* to rozdzielona przecinkami lista nazw tabel, widoków lub podzapytań z których pobierane są dane;
- *warunek* jest opcjonalny, może być dowolnym wyrażeniem warunkowym zbudowanym przy pomocy operatorów dostępnych w języku SQL.

*Uwaga:* Odwołania do *atrybutów* można poprzedzić nazwą *źródła* z kropką (ważne gdy nazwy *atrybutów* powtarzają się w kilku źródłach danych), domyślnie SQL przyjmuje, że odwołanie dotyczy *atrybutu* z pierwszego *źródła* w którym wystąpił.



Operator rzutowania tworzy nową relację, która powstaje z relacji pierwotnej przez wybór wskazanych atrybutów. W SQL rzutowaniu odpowiada lista atrybutów w umieszczona w klauzuli `SELECT`. Symbol `*` oznacza wszystkie atrybuty relacji.

## Przykłady

$\Pi_{ImięP, NazwiskoP} P(\underline{PESEL}, NazwiskoP, ImięP, PWZ)$

**SELECT** ImięP, NazwiskoP **FROM** P;

<i>PESEL</i>	<i>NazwiskoP</i>	<i>ImięP</i>	<i>PWZ</i>
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321

$\Pi_{Kategoria} B(\underline{ICD9}, Kategoria)$

**SELECT** Kategoria **FROM** B;

<i>ICD9</i>	<i>Kategoria</i>	<i>Cena</i>
A14	Leukocyty	25
A15	Glukoza	40
C09	Erytrogram	37
C19	Hemoglobina	28
C41	Limfocyty T	45

$\Pi_{PWZ, NazwiskoL, ImięL} L(\underline{PWZ}, NazwiskoL, ImięL)$

**SELECT** \* **FROM** L;

<i>PWZ</i>	<i>NazwiskoL</i>	<i>ImięL</i>
1234567	Adamska	Alicja
7654321	Wójcik	Zofia

Operacja selekcji tworzy nową relację, która powstaje z relacji pierwotnej poprzez wybór wierszy spełniających określone kryterium. W SQL kryterium selekcji określa wyrażenie warunkowe umieszczone w klauzuli `WHERE`.

*Uwaga: dane tekstowe oraz daty należy umieszczać w apostrofach (').*

## Przykłady

$\sum_{PWZ=1234567} P(\underline{PESEL}, \underline{NazwiskoP}, \underline{ImięP}, PWZ)$

```
SELECT * FROM P
WHERE PWZ='1234567';
```

$\sum_{ICD9=A14} S(\underline{PESEL}, \underline{ICD9}, \underline{Data}, PWZ)$

```
SELECT * FROM S
WHERE ICD9='A14';
```

PESEL	NazwiskoP	ImięP	PWZ
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
70052000234	A15	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	C09	18.01.21	7654321
82012090803	C19	18.01.21	7654321
70052000234	C09	18.01.21	1234567
82012090803	A14	11.02.21	1234567
70052000234	C41	25.02.21	1234567

# Wyrażenia w klauzuli WHERE

Klauzula WHERE dopuszcza następujące typy predykatów:

- porównania logiczne (operatory =, >, >=, <, <=, <>),
- **IN** (*zbiór*) – porównanie wartości ze zbiorem,
- **BETWEN** *przedział* – sprawdzenie przynależności do przedziału,
- **LIKE** – porównanie tekstu z wzorcem,
- **IS NULL** – porównanie z wartością pustą,
- operatory logiczne (OR, AND, NOT).

## Przykłady

Lista skierowań na badania o kodach A14 i A15

```
SELECT * FROM S WHERE ICD9='A14' OR ICD9='A15';
```

```
SELECT * FROM S WHERE ICD9 IN('A14', 'A15');
```

Lista skierowań wystawionych w styczniu 2021 roku

```
SELECT * FROM S WHERE Data>='2021-01-01' AND Data<='2021-01-31';
```

```
SELECT * FROM S WHERE Data BETWEEN '2021-01-01' AND '2021-01-31';
```

# Wyrażenia w klauzuli WHERE – predykat LIKE

Predykat LIKE umożliwia porównywanie wartości tekstowych z podanym wzorcem.

```
nazwa_atrybutu LIKE 'wzorzec'
```

Znaki specjalne:

- `_` (podkreślenie) oznacza dowolny znak,
- `%` (procent) oznacza dowolny ciąg znaków w tym pusty.

## Przykłady

Lista badań, których kod rozpoczyna się od litery „A”

```
SELECT * FROM B WHERE ICD9 LIKE 'A%';
```

Lista badań, których kod kończy się na „19”

```
SELECT * FROM B WHERE ICD9 LIKE '%19';
```

Lista badań, których kod zawiera „19” (w dowolnym miejscu)

```
SELECT * FROM B WHERE ICD9 LIKE '%19%';
```

Lista badań, których kod jest trzycyfrowy i kończy się na „19”

```
SELECT * FROM B WHERE ICD9 LIKE '_19';
```

Złączenie tworzy nową relację zawierającą pary krotek z relacji podlegających złączeniu, które mają spełniają określony warunek. Złączenie naturalne łączy krotki na podstawie wartości atrybutów wspólnych (występujących w obydwu łączonych relacjach), złączenie teta umożliwia zdefiniowanie dowolnego warunku.

Standard SQL-89 realizuje złączenie poprzez wyrażenie warunkowe umieszczone w klauzuli WHERE (nie wykonuje automatycznie złączenia naturalnego).

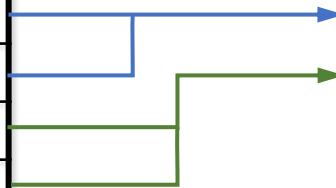
## Przykład

$P(\underline{PESEL}, \text{NazwiskoP}, \text{ImięP}, \text{PWZ}) \otimes L(\underline{PWZ}, \text{NazwiskoL}, \text{ImięL})$

```
SELECT * FROM P, L WHERE P.PWZ=L.PWZ;
```

<i>PESEL</i>	<i>NazwiskoP</i>	<i>ImięP</i>	<i>PWZ</i>
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321

<i>PWZ</i>	<i>NazwiskoL</i>	<i>ImięL</i>
1234567	Adamska	Alicja
7654321	Wójcik	Zofia



Operacja przemianowania w SQL pozwala na nadawanie alternatywnych nazw (*aliasów*) atrybutom oraz tabelom w poleceniu SELECT. Nowe nazwy mogą być używane w zapytaniu, dodatkowo alternatywne nazwy atrybutów opisują kolumny wyniku.

## Nadawanie aliasów atrybutom

```
SELECT nazwa_atrybutu AS alias_atrybutu
FROM ...
```

## Nadawanie aliasów tabelom

```
SELECT alias_tabeli.atrybut
FROM nazwa_tabeli AS alias_tabeli
```

*Uwaga: nadawanie aliasów tabelom pozwalana na skrócenie zapisu kwerendy SQL.*

## Przykład

```
PBadania(Kod ICD9, Kategoria, Cena)(B(ICD9, Kategoria, Cena))
SELECT ICD9 AS 'Kod ICD9', Kategoria, Cena
FROM B AS Badania;
```