



Wykład 2

Wprowadzenie do Baz Danych

Warstwa zarządzania danymi,



Relacyjny model danych



Jacek Bartman

jbartman@univ.rzeszow.pl

3. Bazy danych i systemy informacyjne

★ TEMATYKA:

- określenie różnic pomiędzy danymi, informacją i wiedzą
- umiejscowienie systemów baz danych w systemach informacyjnych
- modele baz danych i ich związek z architekturą systemu informacyjnego
- współczesne obszary zastosowań systemów baz danych

3.1. Dane, informacja, wiedza

★ Dane to fakty (zarejestrowane w bazie danych)

- Dane są zapisywane w postaci symboli, które coś reprezentują

★ Informacja jest to przyrost wiedzy, który może być uzyskany na podstawie danych (Tsitchizris i Lochovsky).

- pojęcie informacji jest płynne i różnie interpretowane
- Informacja ma charakter subiektywny
- Informacje uzyskujemy poprzez zinterpretowanie danych

★ Wiedza jest otrzymywana z informacji przez zintegrowanie z istniejącą wiedzą

3.2. System informacyjny i systemy technologii informacyjnej

- ★ **System informacyjny to system, który dostarcza informacje do przedsiębiorstwa lub jego części (IS)**

- Przedstawia się je modelem wejście-proces-wyjście istniejącego w danym środowisku



- Środowisko systemu to wszystko co ma na niego wpływ i jest poza nim

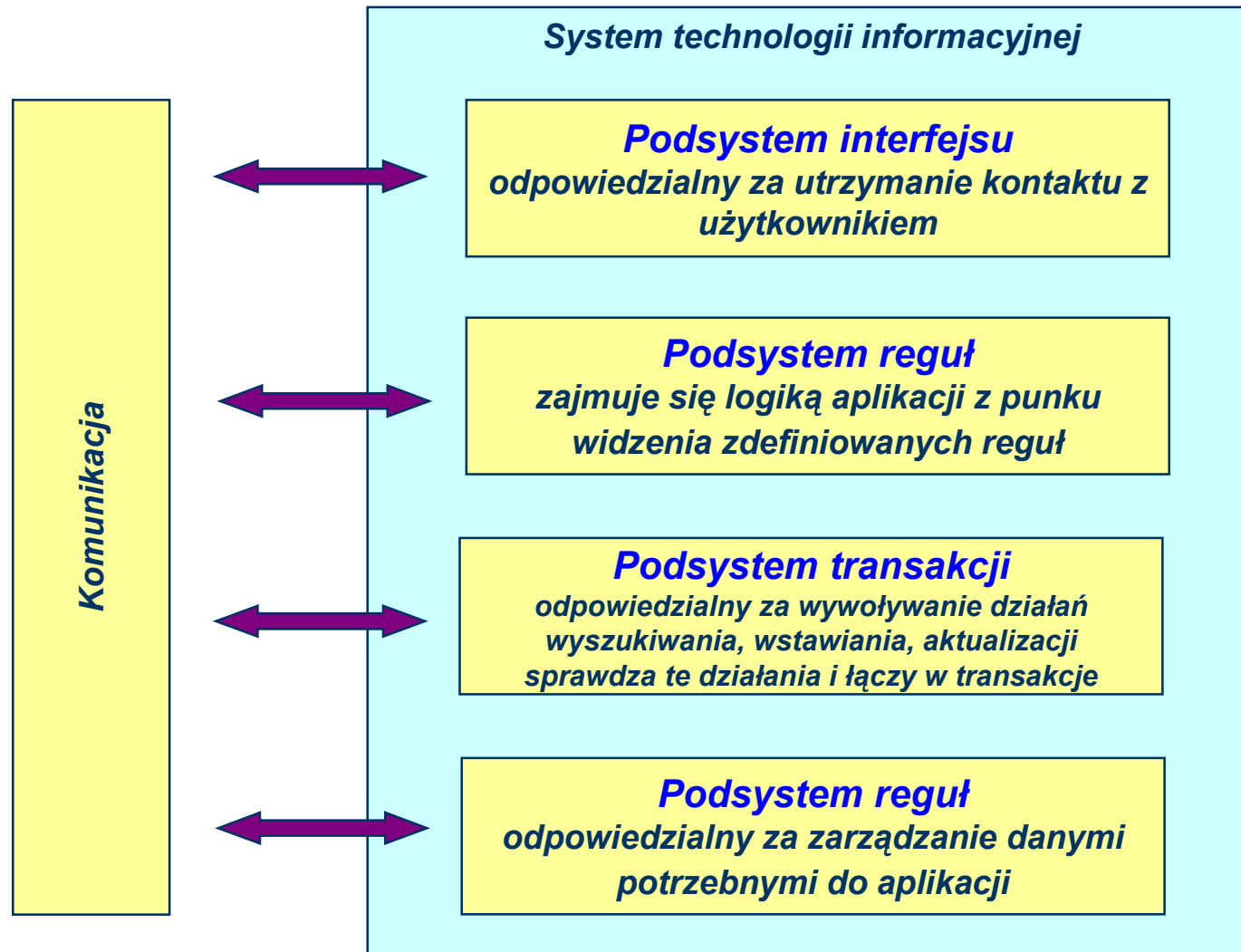
- ★ **System informacyjny obsługują systemy związane z działalnością człowieka.**

- Przedsiębiorstwa najczęściej potrzebują kilku systemów informacyjnych bo posiadają kilka systemów związanych z działalnością człowieka.


- ★ **Technologia informacyjna (IT) obejmuje komputery wraz z oprogramowaniem oraz komunikację**

- Technologia informacyjna zapewnia środki do konstruowania nowoczesnych systemów informacyjnych
- Systemy informacyjne mogą istnieć bez technologii informacyjnej (ale wtedy nie są to systemy technologii informacyjnej)

3.3. Warstwy systemu technologii informacyjnej

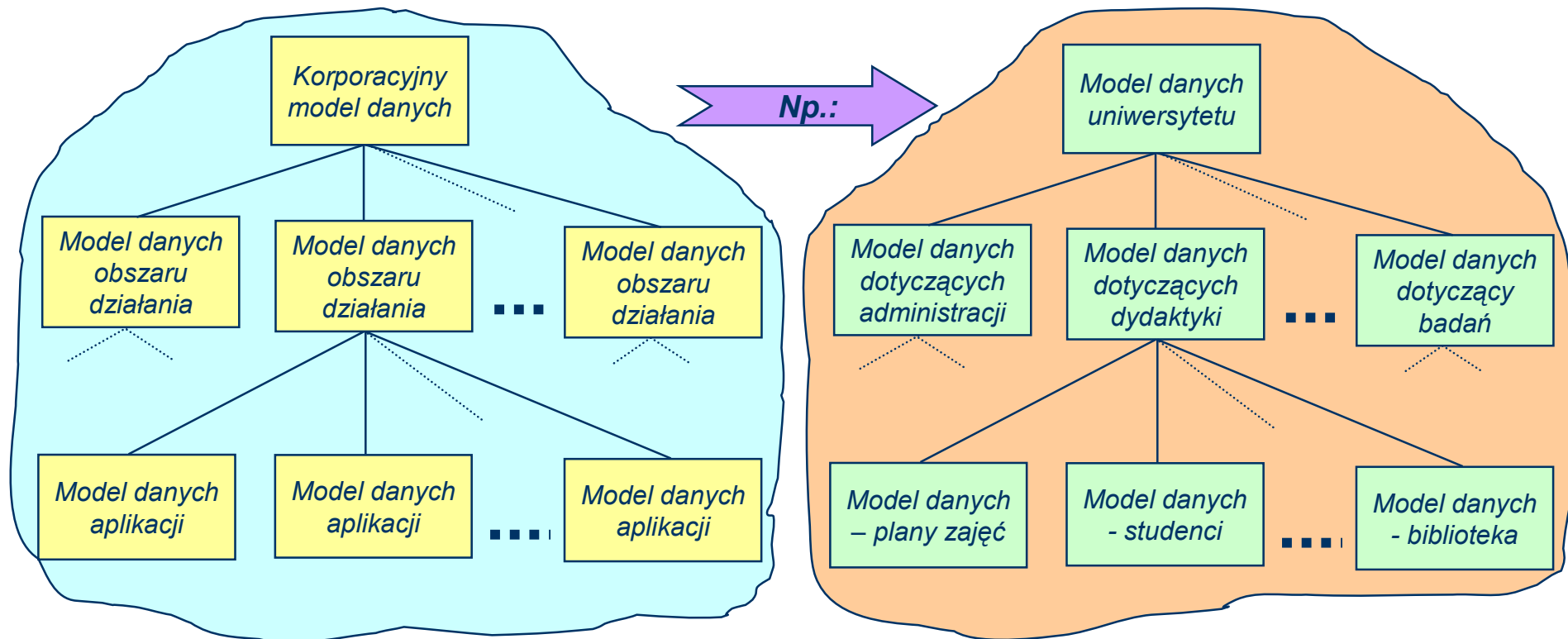


- ★ We współczesnych systemach podsystemy składowe mogą być rozproszone na kilku maszynach znajdujących się w różnych miejscach

- 
- ★ **W starszych aplikacjach wszystkie warstwy były tworzone przy użyciu jednego narzędzia – języka programowania wysokiego poziomu najczęściej trzeciej generacji (np.. Cobol, Clipper).**
 - ★ **Od kilku (kilkunastu) lat poszczególne warstwy systemu tworzone są przez różne narzędzia:**
 - narzędzia graficznego interfejsu użytkownika,
 - języki czwartej generacji (stosowane do kodowania reguł i logiki aplikacji),
 - systemy przetwarzania transakcji,
 - systemy zarządzania bazą danych.
 - oprogramowanie wspierające komunikację.

3.4. Modele danych aplikacji

- ★ Dane powinny być planowane i zarządzane – podstawowym narzędziem temu służącym są model danych
- ★ Model mogą być tworzone co najmniej na trzech poziomach:
 - korporacyjne modele danych (określają wymagania dla całego przedsiębiorstwa)
 - modele danych obszarów działania (biznesowych)
 - modele danych aplikacji.



3.4.1. Korporacyjne modelowanie danych i planowanie systemu informacyjnego

- ★ Odpowiednie zaplanowanie modelu danych dla każdego poziomu jest bardzo ważne
- ★ Architektura systemu informacyjnego może być traktowana jako złożona z trzech warstw wynikających z rozróżnienia pomiędzy informacją, systemem informacyjnym i technologią informacyjną:
 - architektura informacyjna składa się z działań związanych ze zbieraniem, przechowywaniem, rozdzielaniem i wykorzystywaniem informacji,
 - architektura systemów informacyjnych składa się z systemów niezbędnych do obsługi działań na elementach architektury informacyjnej,
 - architektura technologii informacyjnej składa się ze sprzętu, oprogramowania, urządzeń komunikacyjnych, wiedzy oraz umiejętności dostępnych w przedsiębiorstwie.
- ★ Istnieje związek pomiędzy poziomami modeli baz danych i poziomami architektury technologii informacyjnej.
 - Strategia informacyjna – model korporacyjny danych
 - Strategia systemów informacyjnych – model danych obszarów działania
 - Strategia technologii informacyjnej – model danych aplikacji

3.5 Typy współczesnych baz danych

- ★ **Produkcyjne bazy danych** – obsługują standardowe funkcje przedsiębiorstwa. Oferują tworzenie, odczytywanie, modyfikowanie i kasowanie danych.
 - Np.: baza o postępach studentów
- ★ **Bazy danych wspomagające decyzje** – służą do wyszukiwania informacji wspomagających decyzje w przedsiębiorstwie. Oferują tylko odczyt danych.
 - Np.: baza dotycząca rekrutacji na uniwersytecie
- ★ **Informacyjne bazy danych** – są to narzędzia dla użytkowników, mogą być aktualizowane, korzystają z baz produkcyjnych i wspomagających decyzje
 - Np.: wykładowca powinien mieć informacyjną bazę danych dotyczącą uczęszczania na zajęcia studentów.

3.7 Rozwój zastosowań baz danych

- ★ **Hurtownie danych**

- Składnice danych (małe hurtownie)

- ★ **Internetowe bazy danych**

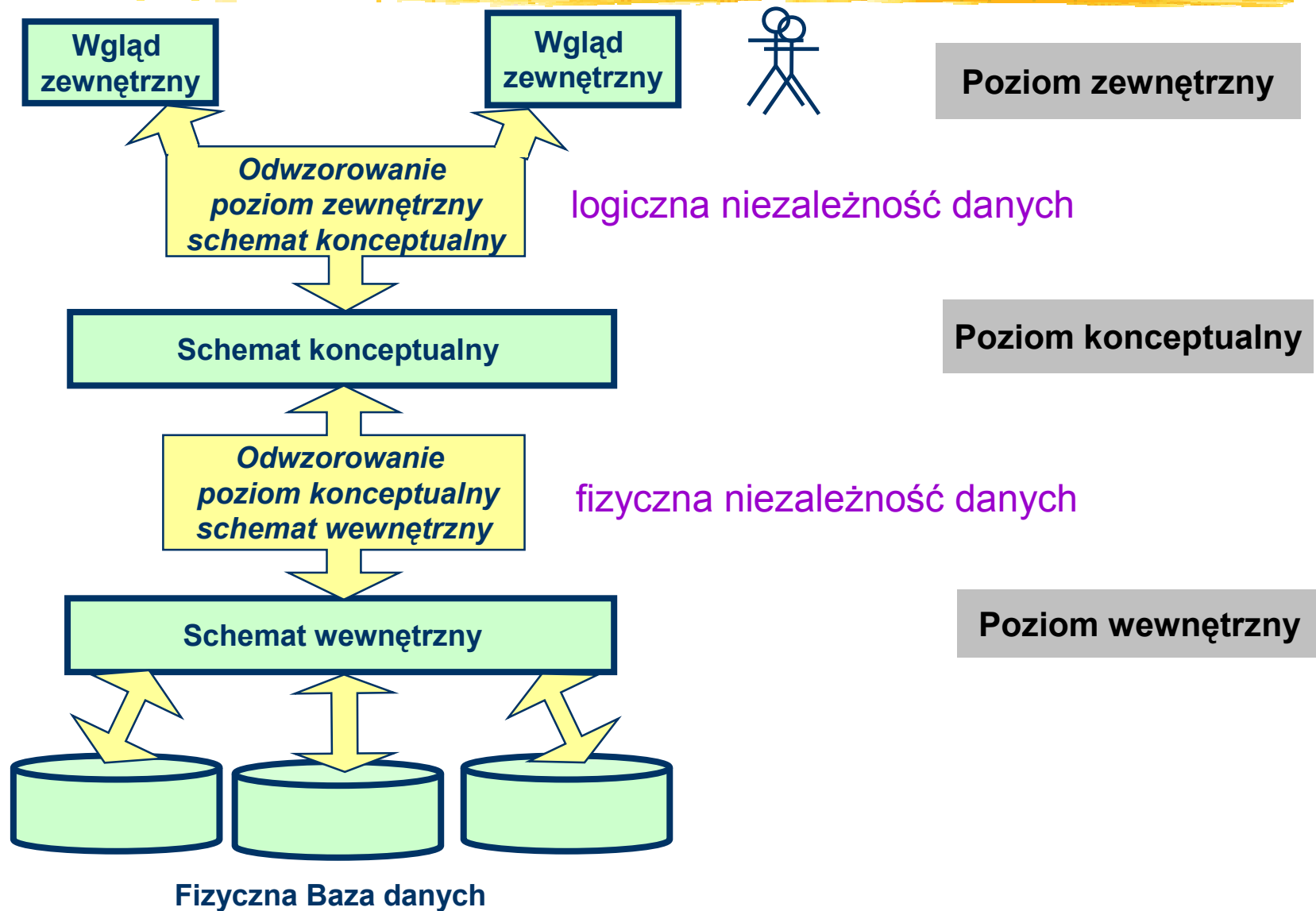
4. Warstwa zarządzania danymi

★ TEMATYKA

- Trzywarstwowa architektura SZBD (DBMS)
- Kluczowe funkcje SZBD
- Interfejsu do SZBD
- Jądro SZBD

4.1 Trzy warstwowa architektura SZBD

- ★ System zarządzania bazą danych jest pośrednikiem (buforem) pomiędzy programami użytkowymi, użytkownikiem końcowym i bazą danych.
- ★ W 1975 (ANSI-SPARC) zaproponował trzypoziomową architekturę SZBD:
 - **poziom zewnętrzny (użytkownika)** – opisuje jak użytkownicy widzą dane,
 - **poziom koncepcyjny (pojęciowy)** – opisuje widok wszystkich danych w bazie. Poziom ten opisuje logiczny widok baz danych, bez szczegółów dotyczących realizacji,
 - **poziom wewnętrzny (fizyczny)** – opisuje sposób przechowywania danych oraz metody dostępu do nich.
- ★ Pomiędzy warstwami istnieją dwa poziomy odwzorowania przekładające się na dwa poziomy niezależności danych:
 - **logiczna niezależność danych** – oznacza niewrażliwość schematów zewnętrznych na zmiany w schemacie koncepcyjnym,
 - **fizyczna niezależność danych** – oznacza niewrażliwość schematu koncepcyjnego na zmiany w schemacie fizycznym.



★ W SZBD można wyróżnić: **jądro, interfejs i zestaw narzędzi**

4.2. Funkcje Systemu Zarządzania Bazą Danych

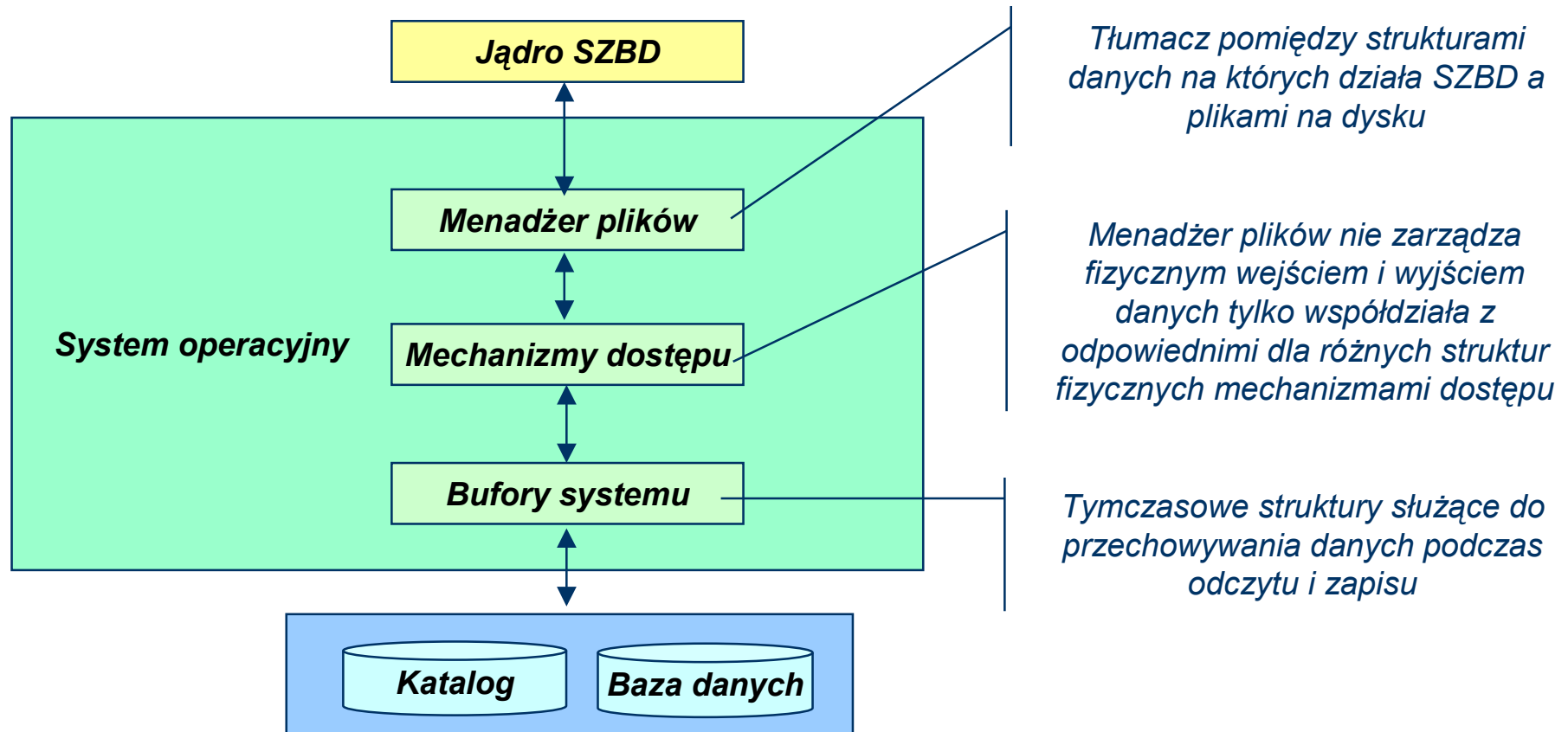
- ★ Funkcje CRUD (Crate Read Update Delete),
- ★ Obsługa słownika danych – przechowywanie metadanych (danych o danych),
- ★ Zarządzanie transakcjami
- ★ Sterowanie współbieżnością
- ★ Odtwarzanie po awarii
- ★ Kontrola uprawnień użytkownika
- ★ Komunikacja danych (wymiana danych w systemie technologii informacyjnej)
- ★ Wymuszanie więzów integralności
- ★ Udostępnianie narzędzi do administrowania bazą danych (do importowania, eksportowania, monitorowania operacji, monitorowania wydajności)

4.3. Interfejs SZBD

- ★ Służy do powiązania jądra systemu z zestawem narzędzi
- ★ Składa się z subjęzyka bazy danych przeznaczonego do inicjowania funkcji SZBD. Składa się on z:
 - **języka definiowania danych (DDL)** służącego do tworzenia, usuwania i uzupełniania struktur danych oraz aktualizacji metadanych,
 - **języka operowania danymi (DML)** służącego do wykonywania operacji CRUD,
 - **języka integralności danych (DIL)** wykorzystywanego do określania więzów integralności,
 - **języka kontroli danych (DCL)** oferuje operacje przeznaczone dla administratora bazy
- ★ Przykładem takiego subjęzyka jest **SQL**

4.4. Jądro SZBD

- ★ Realizuje podstawowe funkcje zarządzania danymi (pkt 4.2.)

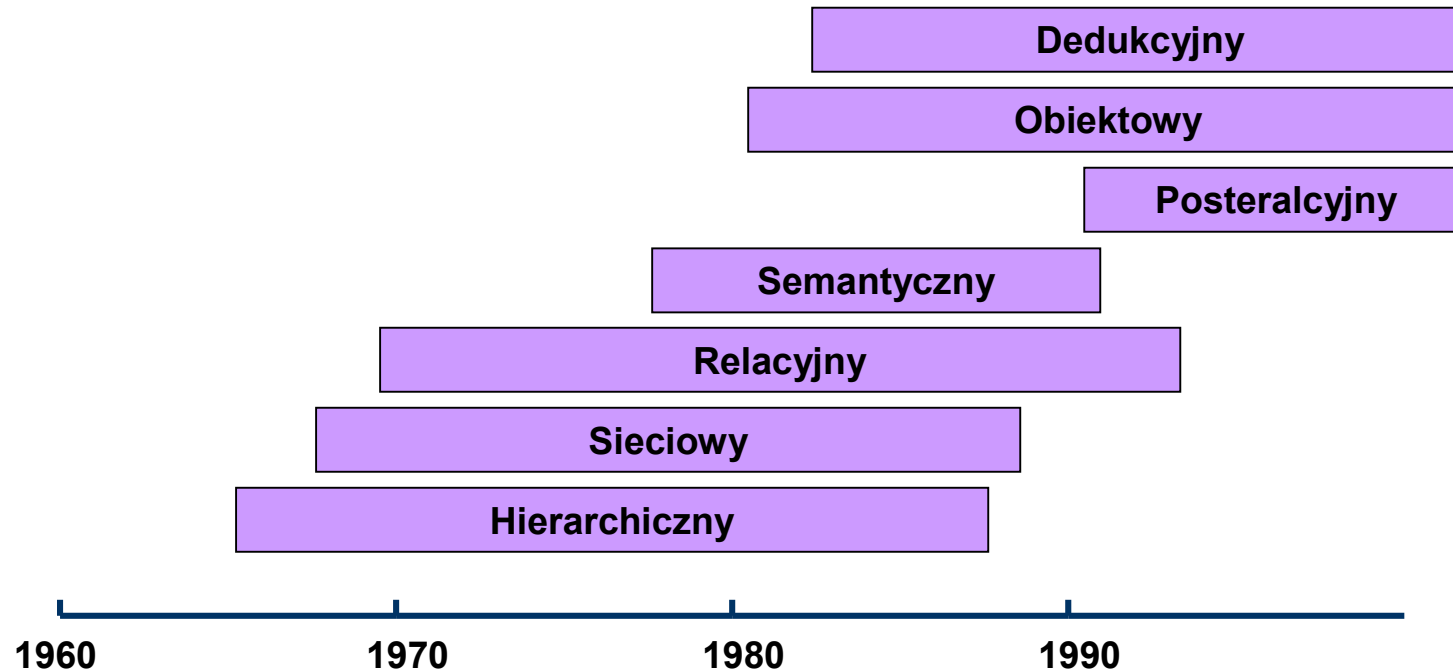


5. Model relacyjny

★ TEMATYKA:

- Modele danych – chronologiczny układ
- Definicja danych wg modelu relacyjnego
- Operowanie danymi
- Integralność danych w modelu relacyjnym

5.1. Chronologia opracowania modeli danych



- ★ Chronologia powstawania modeli danych nie odpowiada chronologii powstawania SZBD opartych na tych modelach

5.2. Ogólnie o modelu relacyjnym

- ★ Relacyjny model bazy danych został opublikowany w 1970 roku przez E.F. Codd'a. Model ten jest oparty jest na gałęziach matematyki zwanych teorią zbiorów i teorią predykatów.
- ★ Zasadą na której opiera się model relacyjny jest to, że typowa baza danych składa się z szeregu nieuporządkowanych tablic (relacji), którymi można manipulować używając nieproceduralnych operacji zwracających całe tablice.
- ★ Słowo „relacyjny” w modelu relacyjnym pochodzi od określenia relacja rozumianego jako specyficzna tablica.
- ★ Codd i inni teoretycy relacyjnych baz danych używają terminów:
„relacja”, „atrybut”, „krotka” tam gdzie większość stosuje pojęcia, odpowiednio:
„tablice”, „kolumny” i „wiersze” ale ma to swoje uzasadnienie.

5.3. Tworzenie definicji danych

- ★ Baza danych jest zbiorem struktur danych służącym do organizowania i przechowywania danych
- ★ W każdym modelu danych i w każdym SZBD musimy dysponować zbiorem reguł określających wykorzystanie struktur danych w aplikacjach bazy danych
- ★ Tworząc definicję danych używamy wewnętrznych struktur danych danego modelu danych w kontekście konkretnego zadania.
- ★ W modelu relacyjnym jest tylko jedna struktura danych – relacja (tablica)

5.3.1. Relacje (tablice)

★ **Relacja jest tabelą spełniającą następujące warunki:**

- każda relacja w bazie danych ma jednoznaczną nazwę,
- każda kolumna ma jednoznaczną nazwę w ramach relacji
- wszystkie wartości w kolumnie są tego samego rodzaju (mają tę samą dziedzinę)
- porządek kolumn w relacji nie jest istotny
- wiersze mają unikalne wartości (są różne)
- kolejność wierszy nie jest istotna
- każde pole relacji winno zawierać elementarne wartości

★ **Tablice w modelu relacyjnym stosowane są do reprezentacji „różnych rzeczy” pochodzących ze świata rzeczywistego.**

★ **Każda tablica powinna reprezentować tylko jedną taką rzecz.**

★ **Rzeczy (jednostki) mogą być rzeczywistymi obiektami bądź zdarzeniami.
Np. rzeczywistym obiektem może być klient, przedmiot z inwentarza lub faktura.
Przykładami zdarzeń mogą być wizyty pacjentów, zlecenia, czy rozmowy telefoniczne.**

Relacje – przykłady

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz danych	1	CSD	244
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445
Roproszone bazy danych	2	CSD	247

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

Kursy	
Kod	NazwaModułu
CSD	Systemy relacyjnych baz danych
	Projektowanie relacyjnych baz danych
	Dedukcyjne bazy danych
	Obiektowe bazy danych
	Rozproszone bazy danych
BSD	Wprowadzenie do biznesu
	Podstawy księgowości

Relacje – przykłady

Studenci			
Nazwisko_Imie	Identyfikator	Rok	RokUr
Kozłowski Zbigniew	ED1034	1	1985
Szubart Joanna	ED1162	1	1986
Kluz Daniel	121342	3	1982
Balawender Robert	MD8956	2	1982
Legień Krystyna	B77880	3	1983

krotka, **wiersz**, **rekord**

atrybut
kolumna

pole

kolumny = atrybuty

wiersze = krotki

liczba kolumn = stopień tabeli

liczba wierszy = liczebność tabeli

5.3.2. Klucz główny (pierwotny)

- ★ Model relacyjny nakazuje, żeby każdy wiersz w tablicy był unikalny. Jeśli pozwoli się na powtarzające się wiersze, wtedy dla programu bazy danych nie będzie istniał żaden sposób na jednoznaczne określenie pozycji danego wiersza.
- ★ Unikalność w tablicy zapewnia się wyznaczając klucz główny.
- ★ **Klucz główny to kolumna (lub grupa kolumn), która jednoznacznie identyfikuje każdy wiersz tabeli.**
- ★ **Każda tablica musi mieć dokładnie jeden klucz główny.**
- ★ Wszystkie kolumny zawierające unikalne wartości nazywa się kluczami potencjalnymi (kandydującymi).
- ★ Spośród nich wybiera się jeden klucz główny, a pozostałe klucze nazywa się kluczami alternatywnymi.
- ★ Klucz prosty składa się tylko z jednej kolumny, natomiast klucz złożony zawiera dwie lub więcej kolumn.

Wybór klucza głównego

★ Klucz kandydujący musi mieć dwie cechy:

- jednoznaczność
- nie może być mieć wartości null

★ Decyzja o wyborze klucza głównego powinna opierać się na:

- **zasadzie minimalności** (klucz powinien zawierać najmniejszą niezbędną ilość kolumn),
- **stabilności** (klucz powinien rzadko ulegać zmianom)
- **prostoty / intuicyjności** (klucz powinien być zarówno prosty jak i łatwy do zrozumienia przez użytkowników)

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

Zawartość tabeli sugeruje unikalność każdego z atrybutów. Jednak z doświadczenia wiemy, iż Nazwiska pracowników mogą się powtórzyć. Podobnie status (typ pracownika) może być jednakowy dla kilku osób. Dlatego jedynie NrPrac nadaje się na klucz główny

★ W wielu sytuacjach jako klucz główny najlepiej obrać jakiś arbitralny, statyczny numer (np.. numer pracownika, numer zlecenia, indeks, itp..), a unikać obierania kolumn zawierających opis tekstowy.

★ Do kluczy głównych, nie należy stosować liczb rzeczywistych, gdyż są one niedokładne

5.3.3. Dziedzina



- ★ **Zbiór z którego kolumna może czerpać swoje wartości.**
- ★ **Dziedziny można traktować jako zdefiniowane przez użytkownika typy kolumn, nakładające na kolumnę reguły, do których muszą się stosować wartości zawarte w kolumnie, a także definiujące operacje, które można na kolumnach wykonywać.**
- ★ **Np.: dziedziną numerów pracownika jest zbiór możliwych numerów pracownika, dziedziną kolumny data urodzenia będą możliwe daty urodzenia.**

5.3.4. Klucze obce (zewnętrzne)

- ✱ **Klucz obcy to taka kolumna (lub grupa kolumn), która zawiera odnośniki do klucza głównego z innej tablicy.**
- ✱ **Klucze główne, chociaż są specyfiką pojedynczych tablic, stanowią niezbędny element w definiowaniu powiązań między tablicami.**
- ✱ **Ważne jest, żeby klucz główny i klucz obcy miały to samo znaczenie i posiadały tę samą dziedzinę.**

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz danych	1	CSD	244
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445
Roproszone bazy danych	2	CSD	247

KZ

KP, KZ

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

5.3.5. Składnia definicji danych

★ Nie istnieje uzgodniona składnia wyrażania struktury relacji

★ Np.: dla bazy akademickiej

Dziedziny

NazwyModułów: CHARACTER(30)
Poziomy: INTEGER: {1, 2, 4}
KodyKursów: CHARACTER (3)
NryPrac: INTEGER
Satusy: CHARACTER: {P, SW, A, L, As, W}
NazwiskaPrac: CHARACTER(20)

Relacja MODUŁY

Atrybuty

NazwaModułu: NazwaModułu
Poziom: Poziomy
KodKursu: KodyKursów
NrPrac: NryPrac

Klucz główny NazwaModułu

Klucz obcy NrPrac do Wykładowcy

Relacja WYKŁADOWCY

Atrybuty

NrPrac: NryPrac
NazwiskoPrac: NazwiskaPrac
Status: Statusy

Klucz główny NrPrac

★ Jak opisuje się w MS ACCESS?

Przykład relacyjnej bazy danych

KLIENCI	Id_klienta	Nazwa_firmy	Adres	Miasto	
ZAMÓWIENIA	Nr_zam	Data_zam	Id_klienta	Wartość_zam	
POZYCJE_ZAMÓWIEŃ	Nr_zam	Nr_katalog_towaru	Ilość_zam		
TOWARY	Nr_katalog_towaru	Rodzaj_towaru	Nazwa_producenta	Model	Cena_sprzed
					Ilość_w_mag
DOSTAWCY	Id_dostawcy	Nazwa_firmy	Adres	Miasto	Kraj
OFERTY_DOSTAWCÓW	Id_dostawcy	Nr_katalog_towaru	Cena		
DOSTAWY	Nr_katalog_towaru	Id_dostawcy	Data_zam	Data_przyjęcia	Ilość_zam

★ Dla każdego atrybutu (kolumny) powinna być określona dziedzina

5.4. Operowanie danymi

★ Przetwarzanie danych wymaga:

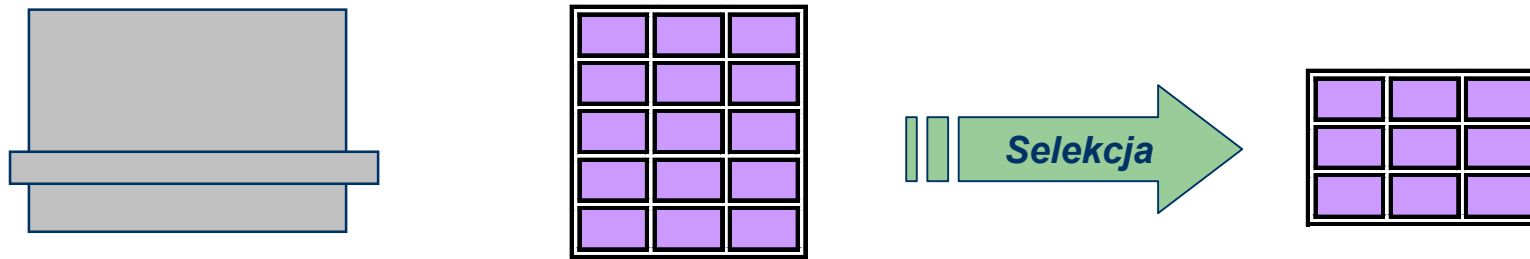
- wstawiania danych do relacji
- usuwania danych z relacji
- poprawiania danych w relacji
- wyszukiwania danych w relacji (służy do tego algebra relacyjna)

5.4.1. Algebra relacyjna

- ★ Algebra relacyjna jest zbiorem ośmiu operatorów.
- ★ Podstawowe operacje algebry relacji to:
 - wybór (selekcja, ograniczenie, restrykcja),
 - rzut (projekcja),
 - złączenie,
 - iloczyn,
 - suma,
 - przecięcie,
 - różnica,
 - iloraz.
- ★ Każdy operator bierze jedną lub więcej relacji jako argument i produkuje jedną relację jako wynik.
- ★ Algebra relacji jest proceduralnym językiem zapytań

Wybór (selekcja, restrykcja, filtrowanie)

- ★ Selekcja jest operatorem, który bierze jedną relację jako swój argument i produkuje w wyniku jedną relację w której umieszcza tylko wiersze spełniające zadany warunek (warunki).



- ★ Składnia operatora selekcji jest następująca:

RESTRICT <nazwa tabeli> [**WHERE** <warunek>] → <tabela wynikowa>

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz danych	1	CSD	244
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445
Roproszone bazy danych	2	CSD	247

RESTRICT Moduły **WHERE** Poziom=4 → Poziom4

Poziom 4			
NazwaModułu	Poziom	KodKursu	NrPrac
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445

Przykład:

drużyna	zawodnik	ile strzelił
Lechia	Kowalski	2
Lechia	Nowak	0
Arka	Aliński	0
Lechia	Kowalski	1
Lechia	Nowak	1
Pogoń	Wielgus	1

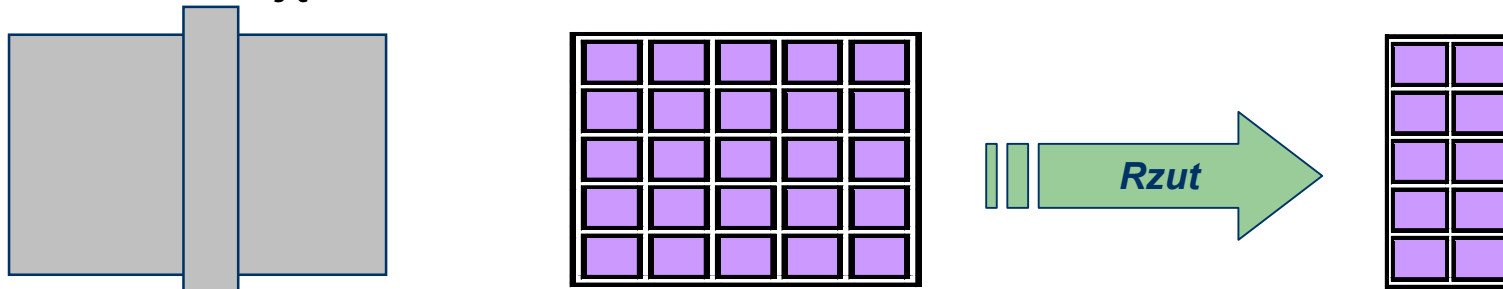
ile-strzelił > 0
→

drużyna	zawodnik	ile strzelił
Lechia	Kowalski	2
Lechia	Kowalski	1
Lechia	Nowak	1
Pogoń	Wielgus	1

Wybór zachowuje ilość kolumn a zmniejsza ilość wierszy.

Rzut

- ★ Operator rzutu wyciąga żądane atrybuty (kolumny) z jednej relacji i tworzy z nich nową relację



- ★ Składnia operatora rzutu jest następująca:

PROJECT <nazwa tabeli> (<lista kolumn>) → <tabela wynikowa>

PROJECT Moduły (NazwaModułu) → NazwMod

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz danych	1	CSD	244
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445
Roproszone bazy danych	2	CSD	247



NazwMod	
NazwaModułu	
Systemy relacyjnych baz danych	
Projektowanie relacyjnych baz danych	
Dedukcyjne bazy danych	
Obiektowe bazy danych	
Roproszone bazy danych	

Przykład:

drużyna	zawodnik	ile strzelił
Lechia	Kowalski	2
Lechia	Nowak	0
Arka	Aliński	0
Lechia	Kowalski	1
Lechia	Nowak	1
Pogoń	Wielgus	1

[zawodnik, ile-strzelił]



zawodnik	ile strzelił
Kowalski	2
Nowak	0
Aliński	0
Kowalski	1
Nowak	1
Wielgus	1

Wybór zachowuje ilość wierszy a zmniejsza ilość kolumn.

Iloczyn

- ★ Operator iloczynu tworzy z dwóch relacji jedną która składa się ze wszystkich możliwych kombinacji wierszy relacji wejściowych.
- ★ Iloczyn w praktyce jest używany bardzo rzadko.
- ★ Operator iloczynu wykorzystany jest do zdefiniowania operatora złączenia.
- ★ Składnia operatora iloczynu jest następująca:

`PRODUCT <tabela1> WITH <tabela2> → <tabela wynikowa>`

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Roproszone bazy danych	2	CSD	247

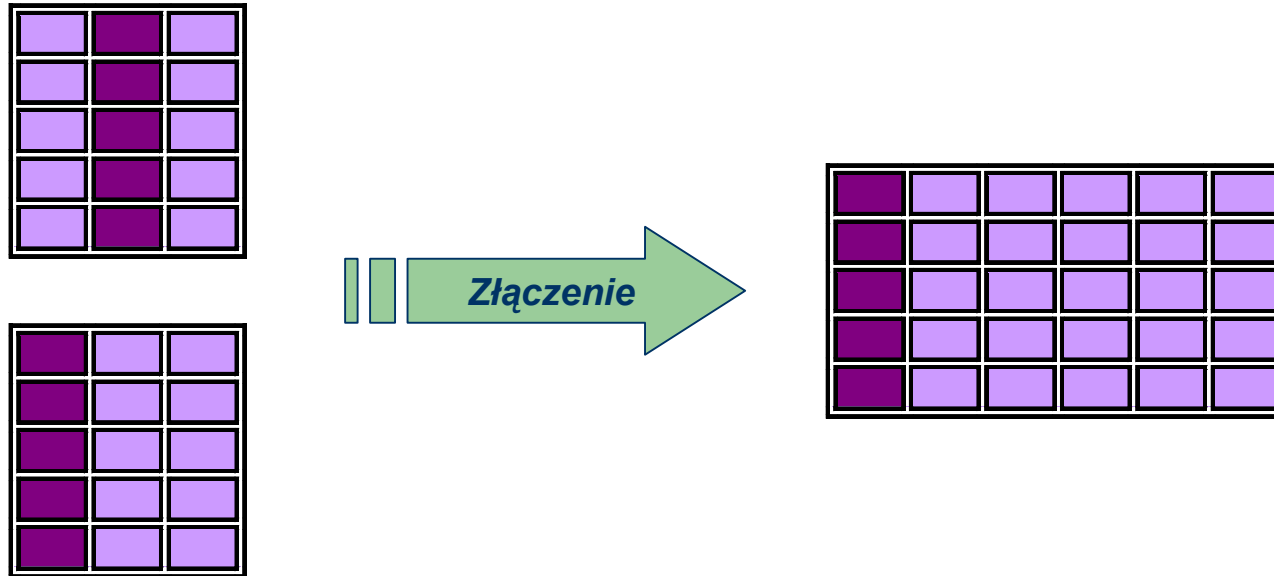
Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

PRODUCT Moduł WITH Wykładowcy → ModWyk

ModWyk						
NazwaModułu	Poziom	KodKursu	NrPrac	NrPrac	NazwiskoPrac	Status
Systemy relacyjnych baz danych	1	CSD	244	244	Buczek Jan	P
Systemy relacyjnych baz danych	1	CSD	244	247	Wysocki Edward	SW
Systemy relacyjnych baz danych	1	CSD	244	445	Kalita Henryk	A
Roproszone bazy danych	2	CSD	247	244	Buczek Jan	P
Roproszone bazy danych	2	CSD	247	247	Wysocki Edward	SW
Roproszone bazy danych	2	CSD	247	445	Kalita Henryk	A

Złączenie

- ★ Operator złączenia bierze dwie relacje i łączy je w jedną.



- ★ Wyróżnia się:
 - równozłączenie
 - złączenie naturalne
 - Złączenie zewnętrzne

Równozłączenie

★ Operator równozłączenia jest iloczynem kartezjańskim, po którym jest wykonywana selekcja (wybór) wybierająca tylko te wiersze, w których wartości w kolumnach złączenia są takie same.

★ Składnia operatora równozłączenia jest następująca:

EQUIJOIN <tabela1> **WITH** <tabela2> → <tabela wynikowa>

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz danych	1	CSD	244
Roproszone bazy danych	2	CSD	247

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

EQUIJOIN Wykładowcy **WITH** Moduł → ModWykR

ModWykR						
NazwaModułu	Poziom	KodKursu	NrPrac	NrPrac	NazwiskoPrac	Status
Systemy relacyjnych baz danych	1	CSD	244	244	Buczek Jan	P
Projektowanie relacyjnych baz danych	1	CSD	244	244	Buczek Jan	P
Rozproszone bazy danych	2	CSD	247	247	Wysocki Edward	SW

Złączenie naturalne

- ★ Operator złączenia naturalnego jest iloczynem kartezjańskim, po którym jest wykonywana selekcja (wybór) wybierająca tylko te wiersze, w których wartości w kolumnach złączenia są takie same, oraz rzut usuwający powtarzające się kolumny
- ★ Składnia operatora złączenia naturalnego jest następująca:

JOIN <tabela1> **WITH** <tabela2> → <tabela wynikowa>

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz danych	1	CSD	244
Roproszone bazy danych	2	CSD	247

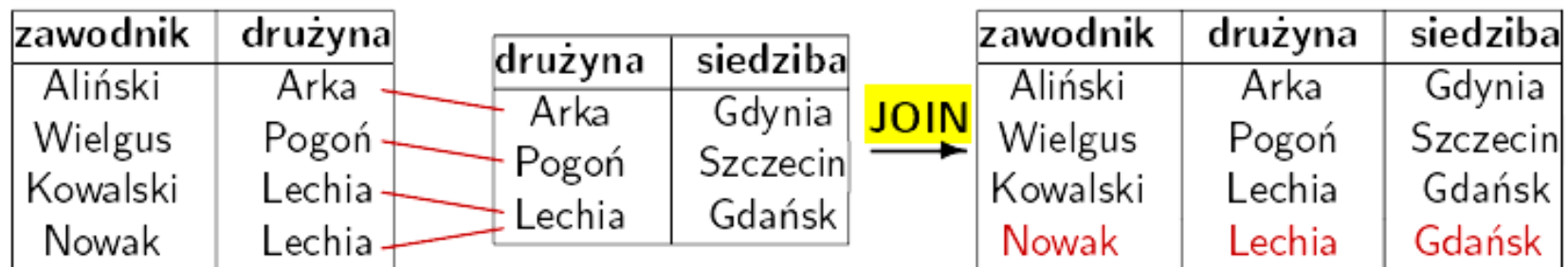
Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

JOIN Wykładowcy **WITH** Moduł → ModWykZN

ModWykZN					
NazwaModułu	Poziom	KodKursu	NrPrac	NazwiskoPrac	Status
Systemy relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Projektowanie relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Rozproszone bazy danych	2	CSD	247	Wysocki Edward	SW

Złączenie tabel

Przykład:



Atrybuty tabelki wynikowej = atrybuty obu tabel argumentowych
(bez powtórzeń)

Encje tabelki wynikowej = każda połączona z każdą
(bez sprzeczności na wspólnych atrybutach)

Złączenie zewnętrzne

- ★ Podobne do złączenia naturalnego, różnica polega na tym, iż pozostawiane są w relacji wynikowej wiersze nie posiadające odpowiedników w obu relacjach wyjściowych
- ★ Wyróżnia się złączenia zewnętrzne:
 - Lewostronne – zachowuje nie pasujące wiersze z tabeli będącej pierwszym argumentem,
 - Prawostronne – zachowuje nie pasujące wiersze z tabeli będącej drugim argumentem
 - Dwustronne – zachowuje nie pasujące wiersze z obu tabeli

Przykład złączenia zewnętrznego lewostronnego

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz da	1	CSD	244
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445
Rozproszone bazy danych	2	CSD	247
Opracowanie baz danych	2	CSD	null
Administrowanie danymi	2	CSD	null

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A
145	Zaborowski Jan	SW
447	Fusiarz Kamila	L



ModWykZZL					
NazwaModułu	Poziom	KodKursu	NrPrac	NazwiskoPrac	Status
Systemy relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Projektowanie relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Dedukcyjne bazy danych	4	CSD	445	Kalita Henryk	A
Obiektowe bazy danych	4	CSD	445	Kalita Henryk	A
Rozproszone bazy danych	2	CSD	247	Wysocki Edward	SW
Opracowanie baz danych	2	CSD	null	null	null
Administrowanie danymi	2	CSD	null	null	null

Przykład złączenia zewnętrznego prawostronnego

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz da	1	CSD	244
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445
Rozproszone bazy danych	2	CSD	247
Opracowanie baz danych	2	CSD	null
Administrowanie danymi	2	CSD	null

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A
145	Zaborowski Jan	SW
447	Fusiarz Kamila	L



ModWykZZP					
NazwaModułu	Poziom	KodKursu	NrPrac	NazwiskoPrac	Status
Systemy relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Projektowanie relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Dedukcyjne bazy danych	4	CSD	445	Kalita Henryk	A
Obiektowe bazy danych	4	CSD	445	Kalita Henryk	A
Rozproszone bazy danych	2	CSD	247	Wysocki Edward	SW
null	null	null	145	Zaborowski Jan	SW
null	null	null	447	Fusiarz Kamila	L

Przykład złączenia zewnętrznego dwustronnego

Moduły			
NazwaModułu	Poziom	KodKursu	NrPrac
Systemy relacyjnych baz danych	1	CSD	244
Projektowanie relacyjnych baz da	1	CSD	244
Dedukcyjne bazy danych	4	CSD	445
Obiektowe bazy danych	4	CSD	445
Rozproszone bazy danych	2	CSD	247
Opracowanie baz danych	2	CSD	null
Administrowanie danymi	2	CSD	null

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A
145	Zaborowski Jan	SW
447	Fusiarz Kamila	L



ModWykZZD					
NazwaModułu	Poziom	KodKursu	NrPrac	NazwiskoPrac	Status
Systemy relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Projektowanie relacyjnych baz danych	1	CSD	244	Buczek Jan	P
Dedukcyjne bazy danych	4	CSD	445	Kalita Henryk	A
Obiektowe bazy danych	4	CSD	445	Kalita Henryk	A
Rozproszone bazy danych	2	CSD	247	Wysocki Edward	SW
Opracowanie baz danych	2	CSD	null	null	null
Administrowanie danymi	2	CSD	null	null	null
null	null	null	145	Zaborowski Jan	SW
null	null	null	447	Fusiarz Kamila	L

Suma

- ★ Suma jest operatorem, który jako argumentów używa dwóch zgodnych relacji i produkuje jedną relację wyjściową, w której uwzględnia wszystkie różne wiersze z obu relacji.
- ★ Relacje zgodne to takie które mają identyczną strukturę i każda kolumna określona jest na tej samej dziedzinie
- ★ Składnia operatora sumy jest następująca:

`<tabela1> UNION <tabela2> → <tabela wynikowa>`

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

Administracja		
NrPrac	NazwiskoPrac	Status
1010	Pawłowicz Maria	U
247	Wysocki Edward	SW

Wykładowcy UNION Administracja → WykAdmSum

WykAdmSum		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A
1010	Pawłowicz Maria	U

Przecięcie

★ Przecięcie jest operatorem, który jako argumentów używa dwóch zgodnych relacji i produkuje jedną relację wyjściową w której uwzględnia tylko identyczne wiersze z obu relacji.

★ Składnia operatora przecięcia jest następująca:

`<tabela1> INTERSECTION <tabela2> → <tabela wynikowa>`

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

Administracja		
NrPrac	NazwiskoPrac	Status
1010	Pawłowicz Maria	U
247	Wysocki Edward	SW

Wykładowcy INTERSECTION Administracja → WykAdmPrz

WykAdmPrz		
NrPrac	NazwiskoPrac	Status
247	Wysocki Edward	SW

Różnica

- ★ Różnica jest operatorem, który jako argumentów używa dwóch zgodnych relacji i produkuje jedną relację wyjściową w której uwzględnia tylko te wiersze które występują jedynie w pierwszej relacji.

- ★ Składnia operatora przecięcia jest następująca:

`<tabela1> DIFFERENCE <tabela2> → <tabela wynikowa>`

Wykładowcy		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
247	Wysocki Edward	SW
445	Kalita Henryk	A

Administracja		
NrPrac	NazwiskoPrac	Status
1010	Pawłowicz Maria	U
247	Wysocki Edward	SW

Wykładowcy DIFFERENCE Administracja → WykAdm

WykAdm		
NrPrac	NazwiskoPrac	Status
244	Buczek Jan	P
445	Kalita Henryk	A

Iloraz

- ★ Iloraz jest operatorem, który jako argumentów używa dwóch relacji i produkuje jedną relację wyjściową. Jedna z tabel wejściowych musi być tabelą binarną (dwukolumnową) a druga unarną (jednokolumnową). Tabela unarna musi mieć dziedzinę zgodną z jedną z kolumn tabeli binarnej
- ★ Idea ilorazu polega na wybieraniu wartości z tabeli unarnej i porównywaniu z odpowiednią kolumną tabeli binarnej. Jeśli wszystkie wartości w tabeli unarnej są zgodne z taką samą wartością w tabeli binarnej to wartość jest wyprowadzana do tabeli wynikowej.

Chcemy znaleźć wspólny dzień w którym wykładane są moduły: Systemy relacyjnych baz danych oraz Projektowanie relacyjnych baz danych

DniModułu	
NazwaModułu	Dzień
Systemy relacyjnych baz danych	04.02.2005
Projektowanie relacyjnych baz danych	02.02.2005
Projektowanie relacyjnych baz danych	04.02.2005
Obiektowe bazy danych	02.02.2005

ParyModułów	
NazwaModułu	
Systemy relacyjnych baz danych	
Projektowanie relacyjnych baz danych	



DniWspolne	
Dzień	
04.02.2005	

5.4.2. Algebra relacyjna - proceduralny język zapytań

- ★ Algebra relacyjna stanowi proceduralny język zapytań (ma właściwość domknięcia) – w celu wydobywania informacji z bazy danych określamy ciąg operatorów, w którym wynik z każdego kroku może być użyty jako argument wejściowy w następnych krokach.

Wypisz wszystkie moduły prowadzone przez Buczka Jana

1 `RESTRICT Wykładowcy WHERE NazwiskoPrac = 'Buczek Jan' → Tab1`
`JOIN Tab1 WITH Moduły ON NrPrac → Tab2`
`PROJECT Tab2 (NazwaModułu) → Tab3`

Wybiera z relacji *Wykładowcy* rekordy zawierające w polu *NazwiskoPrac* wyrażenie *Buczek Jan* i zapisuje w relacji *Tab1*

Wybiera z relacji *Tab2* kolumnę *NazwaModułu* i zapisuje w relacji *Tab3*

Łączy relacje *Tab1* oraz *Moduły* zapisując w tworzonej relacji *Tab2* tylko te wiersze w których pola *NrPrac* w obu relacjach wejściowych są identyczne

2 `JOIN Wykładowcy WITH Moduły ON NrPrac → Tab1`
`RESTRICT Tab1 WHERE NazwiskoPrac = 'Buczek Jan' → Tab2`
`PROJECT Tab2 (NazwaModułu) → Tab3`

1a `PROJECT (JOIN Moduły WITH`
`(RESTRICT Wykładowcy WHERE NazwiskoPrac = 'Buczek Jan') ON NrPrac) (NazwaModułu) → Tab1`

Zagnieżdżona wersja zapisu 1. Wynik jednej operacji jest argumentem następnej

5.4.3. Rachunek relacyjny

- ★ **Rachunek relacyjny stanowi alternatywę dla algebry relacyjnej**
- ★ **Rachunek relacyjny ma charakter nieproceduralny lub deklaracyjny (algebra proceduralny).**
 - W rachunku formułujemy wyrażenie, które określa co ma być wyszukane, a nie jak to wyszukać
- ★ **Rachunek relacyjny jest oparty na rachunku predykatów.**
- ★ **Istnieją dwa warianty rachunku relacyjnego:**
 - rachunek relacyjny oparty na rekordach (podstawa SQL)
 - rachunek relacyjny oparty na dziedzinach (podstawa QBE – Query By Example)

Rachunek relacyjny na krotkach

- ★ Wyrażenie w tym rachunku ma postać:

```
RANGE OF <zmienna krotkowa> IS <nazwa tabeli>
GET <zmienna krotkowa>.<nazwa atrybutu>
INTO <tabela wynikowa>
WHERE <wyrażenie warunkowe>
```

Wybierz wykładowców p statusie SW

```
RANGE OF Kr1 IS Wykładowcy
GET Kr1.Status
INTO Tab1
WHERE Kr1.Status = SW
```

- ★ Instrukcja relacyjnego rachunku na krotkach może zawierać:
 - spójniki logiczne *and, or, not*,
 - kwantyfikatory egzystencjalne *Forsome* (istnieje)
 - kwantyfikatory uniwersalne *Forall* (dla każdego)

```
RANGE OF Kr1 IS Wykładowcy
GET Kr1.Status
INTO Tab1
WHERE Forsome Kr1 (Kr1.Status = SW)
```

Efekt jak wyżej

```
RANGE OF Kr2 IS Moduły
GET Kr2.Poziom
INTO Tab1
WHERE Forall Kr1 (Kr1.Status = SW)
```

Wydobędzie z bazy wszystkie poziomy związane z modułami

5.4.4. Operacje dynamiczne na relacjach

★ Istnieją trzy operacje dynamiczne (zarówno w algebrze relacyjnej jak i w rachunku):

- wstawianie

`INSERT (<wartość>,<wartość>,...) INTO <nazwa tabeli>`

- usuwanie

`DELETE <nazwa tabeli> WITH <warunek>`

- modyfikowanie

`UPDATE <nazwa tabeli> WHERE <warunek> SET <nazwa kolumny>=<wartość>`

```
INSERT ('Systemy rozproszonych baz danych',2,CSD,247) INTO Moduły
DELETE Moduły WITH Poziom = 1
UPDATE Moduły WHERE Poziom = 2 SET Poziom = 1
```

5.5. Integralność danych

- ★ Integralność danych oznacza, iż baza danych stanowi odzwierciedlenie rzeczywistości.
- ★ Integralność zapewniają reguły integralności
- ★ W relacyjnym modelu danych istnieją dwa rodzaje wewnętrznych reguł integralności:
 - integralność encji (jednostek)
 - integralność referencyjna (odniesień)
- ★ Aby wyrazić wszystkie aspekty integralności stosuje się także tzw. integralności dodatkowe

5.5.1. Integralność encji (jednostek)

Zasada integralność encji mówi, że klucze główne nie mogą zawierać pól pustych

- ★ Uzasadnieni zasady jest oczywiste – nie da się jednoznacznie zidentyfikować wiersza w tablicy (krotki w relacji) jeśli klucze będą mogły być nieokreślone.
- ★ W przypadku kluczy złożonych żadna z indywidualnych kolumn nie może zawierać nieokreślonych (pustych) pól

5.5.2. Integralność referencyjna (odniesień)

Zasada integralności referencyjnej mówi, że klucz obcy nie może odnosić się do nieistniejących rekordów

★ **Z reguły wynika, że:**

- do tablicy nie można dodać wiersza, którego wartość klucza obcego nie ma odpowiednika w tablicy odniesień,
- zmiana wartości klucza głównego w tablicy (lub usunięcie całego rekordu) nie może prowadzić do „osierocania” wierszy w innej tablicy (która poprzez klucz obcy wskazuje na modyfikowany klucz główny).

★ **Utrzymanie integralności referencyjnej wymaga określenia więzów propagacji (zdefiniowania co zrobić w przypadku modyfikacji klucza głównego z wierszami powiązanymi z nim). Możliwe są trzy podejścia:**

- zakazać dokonywania zmian klucza głównego w tablicy odniesień jeżeli istnieją do niego powiązania – metoda restrykcyjna
- propagować zmiany do wszystkich tablic zawierających referencje do zmienianych wierszy (w przypadku usuwania rekordu usuwać rekordy powiązane) – podejście ufne
- Wyzerować wszystkie wartości kluczy obcych odnoszących się do usuwanych rekordów – podejście wyważone.

5.5.3. Integralność dodatkowa

- ★ Dodatkowe więzy integralności pochodzą ze środowiska modelowanego przez bazę danych