



Bazy Danych

Dr inż. Przemysław Falkowski-Gilski

Wydział Elektroniki, Telekomunikacji i Informatyki



- Wykład opracowany na podstawie materiałów pomocniczych z Baz Danych autorstwa: K. Goczyły, A. Landowskiej, M. Piechówka oraz A. Ramsa
- Lektura materiału nie zastępuje obecności na wykładzie
- Wykorzystanie materiałów w innym celu oraz ich rozpowszechnianie jest zabronione



- Banachowski L., Chądryńska A., Matewski K., *Relacyjne bazy danych. Wykłady i ćwiczenia*, PJWSTK, 2004.
- Beynon-Davies P., *Systemy baz danych*, WNT, 2000.
- Connolly T., Begg C., *Systemy baz danych*, RM, 2004
- Date C. J., *Wprowadzenie do systemów baz danych*, WNT, 2000.
- Delobel C., Adiba M., *Relacyjne bazy danych*, WNT, 1988.
- Dybowska-Dyk A., Bartnik M., *Ćwiczenia z SQL*, Mikom, 1999.
- Elmasri R., Navathe S. B., *Wprowadzenie do systemów baz danych*, Helion, 2005.



- Gruber M., *SQL*, Helion, 2000.
- Muller R. J., *Bazy danych. Język UML w modelowaniu danych*, Mikom, 2000.
- Riordan R. M., *Projektowanie systemów relacyjnych baz danych*, Microsoft Press, 2000.
- Tsichritzis D., Lochovsky F., *Modele danych*, WNT, 1990.
- Ullman J. D., *Systemy baz danych*, WNT, 1988.
- Ullman J. D., Widom J., *Podstawowy wykład z systemów baz danych*, WNT, 1999.



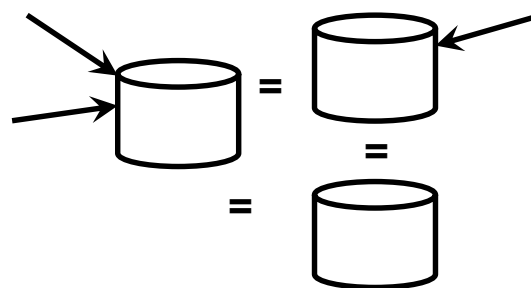
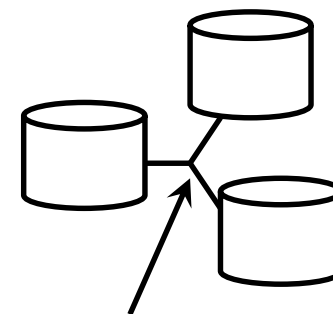
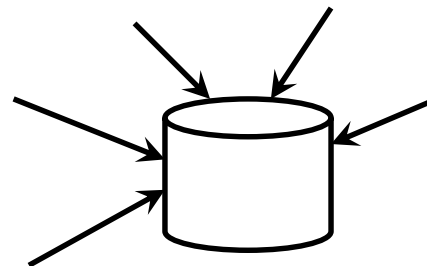
- Zbiór powiązanych ze sobą logicznie danych, zorganizowany zgodnie z określonym modelem danych
- Model danych – wyznacza logiczne reguły konstrukcji bazy danych, bez względu na jej fizyczną reprezentację
- System baz danych – wiele baz danych zarządzanych przez jeden system komputerowy
- System zarządzania bazami danych (DBMS – *Database Management System*) – oprogramowanie zarządzające systemem baz danych

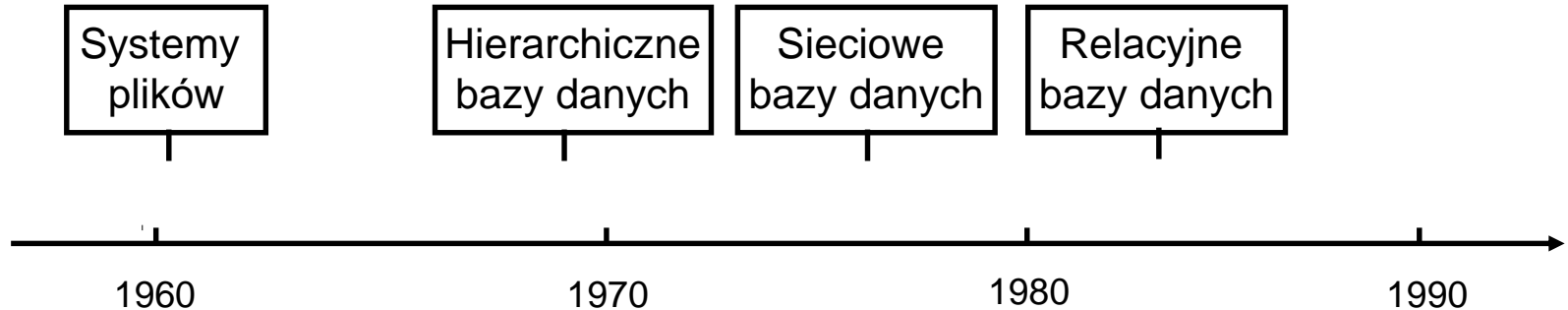


- Baza danych jest zarządzana przez system komputerowy (sprzęt i oprogramowanie), który powinien wypełniać następujące funkcje:
 - trwałe i niezawodne przechowywanie dużych ilości informacji
 - efektywny dostęp do żądanych informacji
 - możliwość modyfikowania informacji
 - zagwarantowanie logicznej spójności (integralności) informacji
 - ochrona informacji

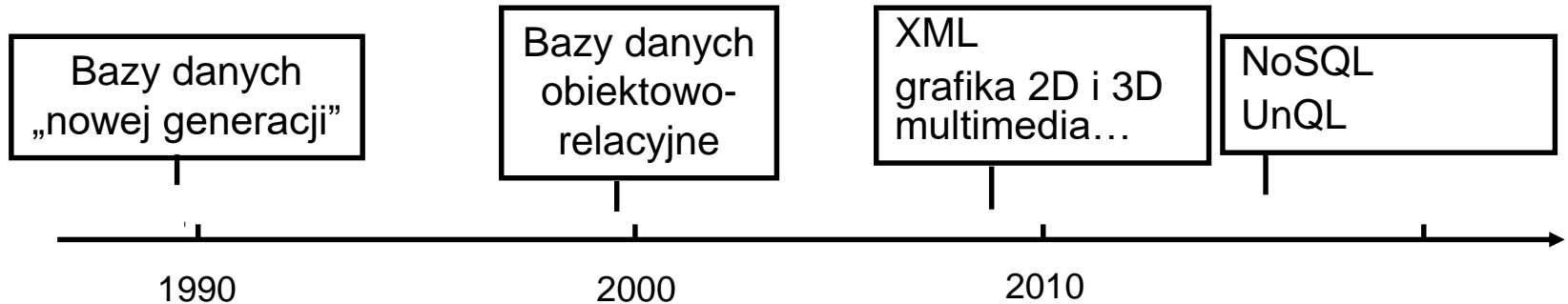


- Model danych:
 - relacyjny
 - obiektowy
 - obiektowo-relacyjny
 - semistrukturalny (XML)
 - sieciowy, hierarchiczny (dawniej)
- Liczba węzłów:
 - scentralizowana baza danych
 - rozproszona baza danych
 - replikowana baza danych





- Systemy plików:
 - realizacja operacji plikowych; stanowią poziom fizyczny współczesnych baz danych, ISAM, CSAM, VSAM, WiSS, inne specjalizowane
- Hierarchiczne bazy danych:
 - dane organizowane są w struktury drzewiaste, IMS, System 2000
- Sieciowe bazy danych
 - dane organizowane są w dowolne struktury sieciowe (Codasy1), IDS, Total, Adabas, IDMS
- Relacyjne bazy danych
 - dane organizowane są w relacje (zbiory krotek o ustalonej, prostej strukturze), Oracle, Informix, SyBase, DB2, Ingres, Progress, MSSQL



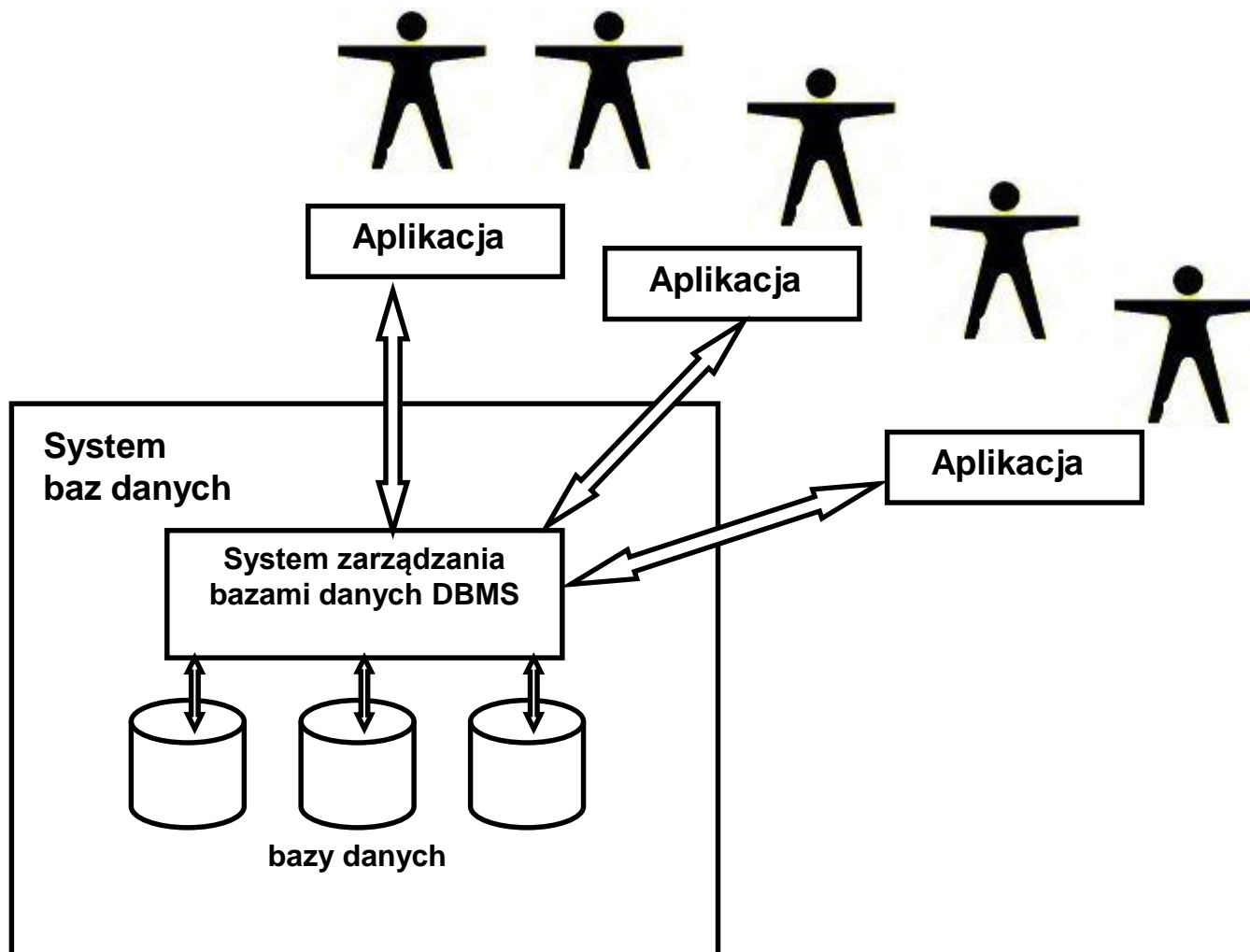
- Bazy danych nowej generacji – motywacja
- Nowe aplikacje baz danych:
 - CAD, CAM, CAP, CASE, DSS, automatyzacja prac biurowych, obiekty 2D (kartografia), 3D (geometryczne) i wielowymiarowe (analiza danych), multimedia, sztuczna inteligencja, bazy NoSQL (kolekcje dokumentów z luźno określonymi polami, model klucz-wartość)



- Niedostatki systemów tradycyjnych:
 - zbyt prosty model danych, nieodpowiedni do modelowania złożonych struktur
 - ograniczony, nierozszerzalny zestaw typów danych
 - ograniczone możliwości obliczeniowe
 - zbyt prosty model transakcji
 - brak pojęcia wersji
 - mała elastyczność w operowaniu schematem



- Elementy tworzące system muszą być trwałe:
 - ręczne programowanie trwałości jest kłopotliwe, zawodne, nieelastyczne
 - możliwe tylko dla niewielkich zbiorów danych, w środowisku jednego użytkownika
- Potrzeba efektywnego operowania na dużych wolumenach danych:
 - obszerne składnice (magazyny) ustrukturalizowanych danych, zarządzane przez gotowe oprogramowanie
- Złożoność danych:
 - złożoność strukturalna i złożoność powiązań między danymi, a także złożoność ograniczeń integralnościowych
- Narzędzie implementacyjne:
 - środowiska wspomagające projektowanie i wdrażanie aplikacji opartych na bazach danych, np. języki typu 4GL lub środowiska dla języków 3GL z bibliotekami do korzystania z interfejsów ODBC/JDBC; łączą one cechy typowych języków programowania z funkcjami dostępu do bazy danych

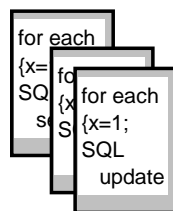
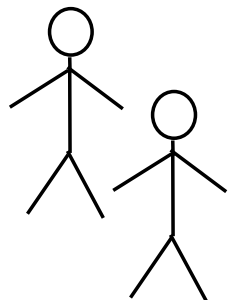




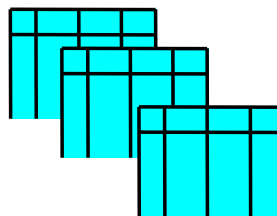
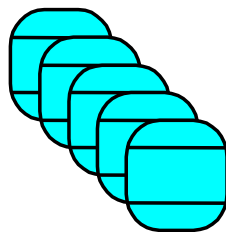
- SYSTEM BAZ DANYCH = Trwała pamięć zewnętrzna + System Zarządzania Bazami Danych (DBMS) + Język zapytań
- Trwała pamięć zewnętrzna:
 - urządzenia sprzętowe przechowujące dane
- System Zarządzania Bazami Danych (*Data Base Management System*):
 - oprogramowanie zarządzające dostępem do baz danych



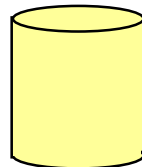
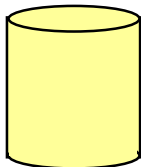
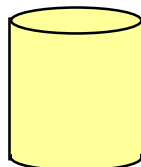
Poziom
zewnętrzny
(użytkownika)



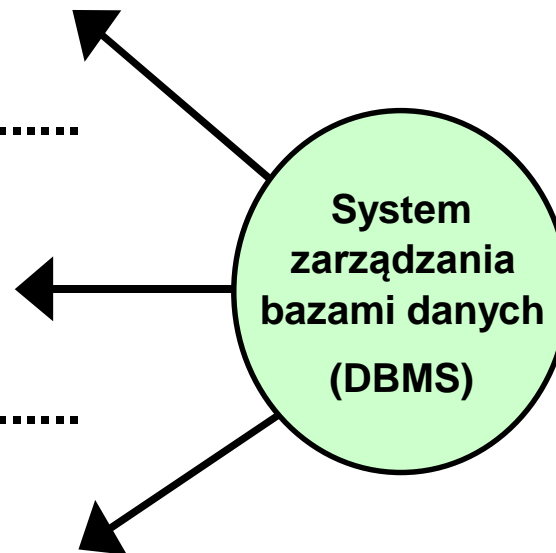
Poziom
logiczny
(modelu
danych)



Poziom
wewnętrzny
(fizyczny)



System
zarządzania
bazami danych
(DBMS)





- Poziom zewnętrzny (użytkownika):
 - zbiór zewnętrznych widoków, poprzez które użytkownicy widzą zawartość bazy danych. Widok jest wyabstrahowanym fragmentem bazy danych, zdefiniowanym w terminach języka zapytań
- Poziom logiczny (modelu danych):
 - Zawiera model konceptualny bazy danych. Model ten obejmuje całą zawartość bazy danych, tak jak ją widzi administrator lub właściciel bazy danych
- Poziom wewnętrzny (fizyczny):
 - Fizyczna reprezentacja bazy danych w postaci plików dyskowych i algorytmów dostępu do nich. Jest to reprezentacja niskiego poziomu (poziomu jednostek pamięciowych, np. bajtów).



- Role użytkowników baz danych:
 - użytkownicy interakcyjni
 - programiści aplikacji
 - administratorzy danych
 - administratorzy systemów baz danych (DBA)
 - projektanci baz danych
 - analitycy systemowi
- Role powiązane:
 - administratorzy systemów komputerowych i administratorzy sieci
 - twórcy systemów SZBD
 - projektanci narzędzi programistycznych



- Administrator bazy danych (DBA – *Data Base Administrator*) – osoba sprawująca kontrolę nad dostępem do bazy danych i nad jej funkcjonowaniem
- Zadania DBA:
 - udzielanie użytkownikom i grupom użytkowników uprawnień dostępu do bazy danych (uprawnienia do tworzenia obiektów bazy danych, ich pobierania i modyfikowania)
 - monitorowanie wydajności bazy danych
 - definiowanie strategii archiwizowania i odtwarzania danych po awariach
 - uruchamianie nowych strategii dostępu do danych

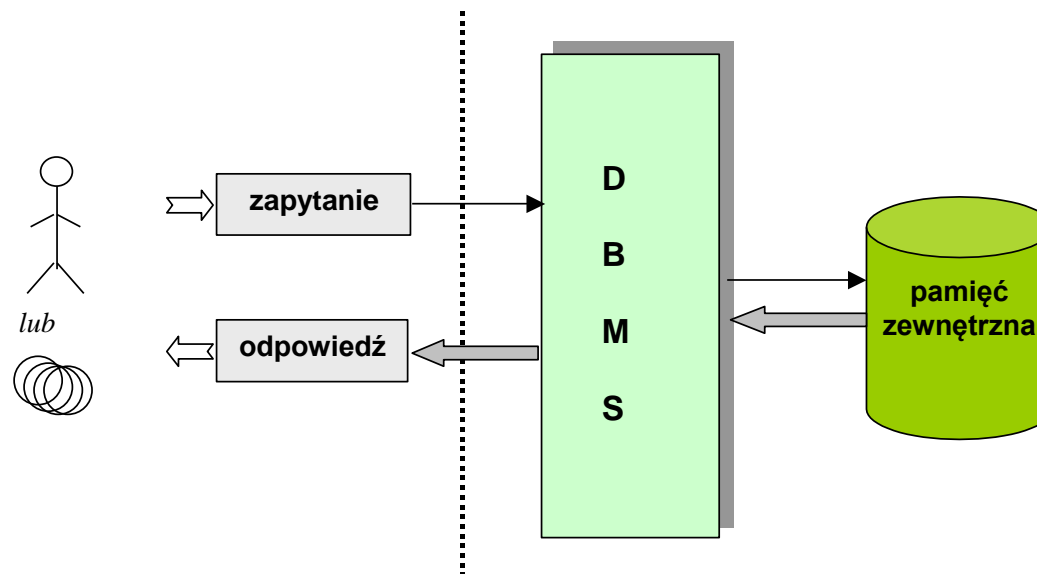


- Narzędzia do dyspozycji DBA:
 - program do ładowania bazy danych z plików zewnętrznych
 - programy do archiwizacji danych
 - programy do przywracania danych z archiwum
 - programy do zbierania statystyk i ich analizowania
 - procedury definiowania użytkowników i grup użytkowników
 - programy do operowania na słowniku danych
 - programy do analizowania dziennika bazy danych (logu)



- Użytkownik:

- człowiek przy terminalu, który interakcyjnie operuje na bazie danych, formułując żądania w języku zapytań ad hoc
- program, który oprócz innych instrukcji zawiera instrukcje operujące na danych z bazy danych. Instrukcje te formułowane są w języku zapytań zanurzone w języku macierzystym





- Logiczna niezależność danych – odporność schematów zewnętrznych (widoków) na zmiany dokonywane w schemacie pojęciowym
- Fizyczna niezależność danych – odporność schematu pojęciowego na zmiany dokonywane w schemacie wewnętrznym
- Korzyści:
 - można wprowadzać nowe urządzenia pamięciowe
 - można wprowadzać nowe struktury danych na dysku i nowe metody dostępu
 - można zmieniać fizyczną reprezentację danych

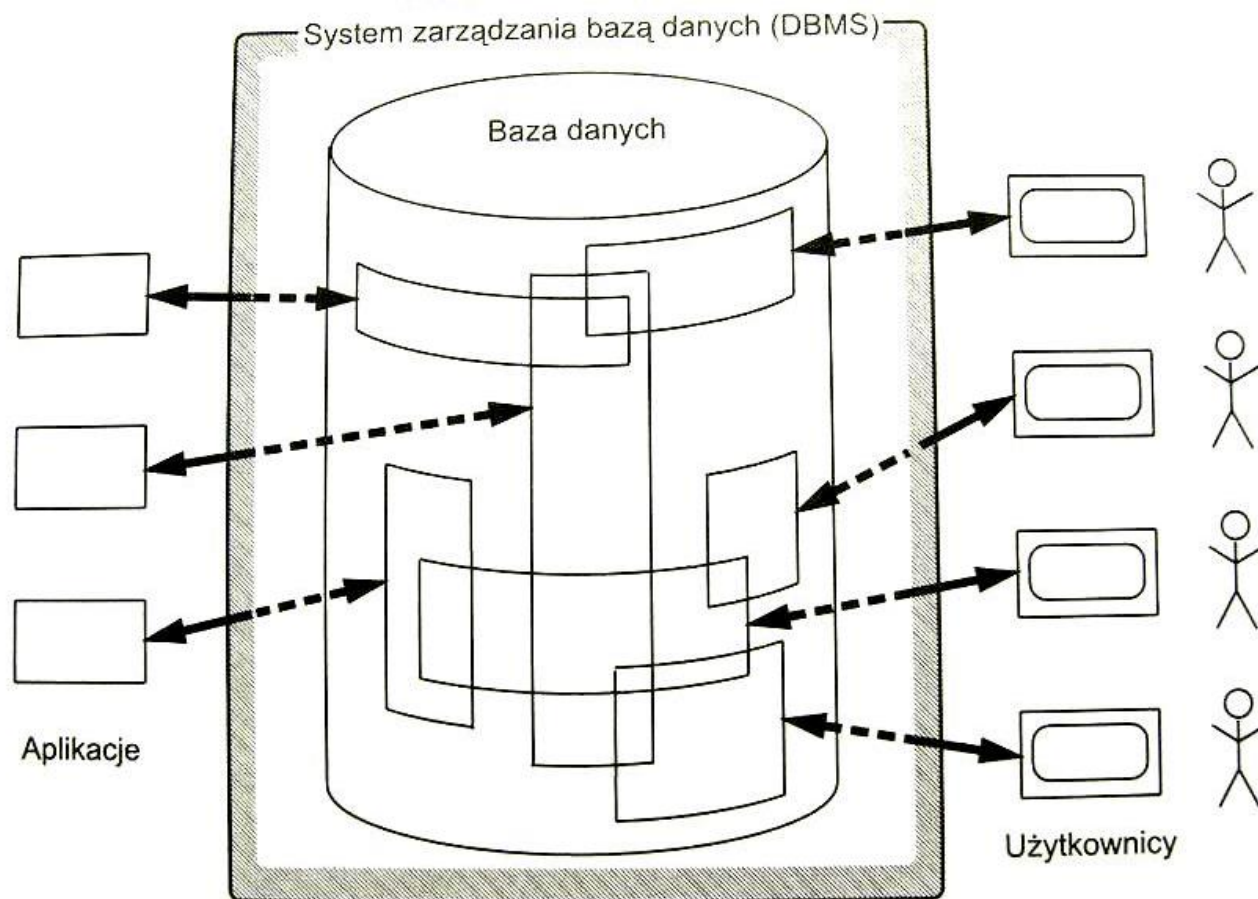
bez konieczności dokonywania zmian w kodzie aplikacji bazodanowych (może być konieczna tylko rekompilacja)



- Wyraźne odizolowanie poziomów architektury systemu baz danych (szczególnie poziomu logicznego od fizycznego) pozwala osiągnąć niezależność programu od danych
- Program jest zależny od danych, jeśli wiedza na temat struktur danych i metod dostępu do nich jest wbudowana w program, tzn. nie jest możliwa zmiana (bez dokonania zmian w programie):
 - struktur danych (czyli sposobu przechowywania danych)
 - metod dostępu do danych
- Niezależność programu od danych oznacza sytuację odwrotną:
 - logika programu nie zależy od struktur danych i metod dostępu stosowanych w systemie baz danych



- Współczesne aplikacje wymagają znacznie bogatszego zestawu typów danych niż oferowany przez klasyczny model relacyjny
- Geneza:
 - zastosowania komputerów w dziedzinach związanych z projektowaniem i przetwarzaniem danych o nieregularnej, często nie znanej z góry strukturze
- Dziedziny:
 - inżynieria oprogramowania (CASE)
 - wspomaganie projektowania (CAD)
 - obsługa produkcji (CAM)
 - wspomaganie podejmowania decyzji (DSS)
 - wspomaganie prac wydawniczych (CAP)
 - automatyzacja prac biurowych
 - kartografia
 - obróbka danych multimedialnych
 - sztuczna inteligencja



Źródło: Date C. J., *Wprowadzenie do systemów baz danych*, WNT, 2000.



- Realizacja operacji dostępu do danych na poziomie fizycznym (algorytmy dostępu, struktury danych, itp.)
- Zapewnienie integralności danych (przetwarzanie transakcyjne, kontrola ograniczeń nałożonych na dane, itp.)
- Obsługa współbieżności (rozstrzyganie konfliktów przy próbach jednoczesnego dostępu do tych samych danych)
- Ochrona danych (kontrola uprawnień do wykonywania określonych operacji przez określonych użytkowników)



- Odtwarzanie po awariach (restart, przywracanie danych z kopii zapasowych)
- Śledzenie operacji dokonywanych na bazie danych (efektywność, bezpieczeństwo, niezawodność)
- Obsługa rozproszenia (jeśli jedna baza danych fizycznie jest umieszczona na różnych systemach komputerowych)
- Obsługa przetwarzania równoległego (jeśli zapytanie użytkownika skierowane do bazy danych może być realizowane równocześnie przez kilka procesorów)



- Komercyjne:
 - Oracle – wersje 9i, 10g, 11g
 - Microsoft – Access, SQL Server 2000, SQL Server 2005, SQL Server 2008
 - IBM - DB2, Informix
 - Sybase – Adaptive Server
 - PSC – Progress (4GL)
- Niekomercyjne:
 - mySQL, postgresQL, FireBird, Derby, Ingres, SQLite
- Obiektowe:
 - eXcelon – ObjectStore, Objectivity, Versant



- Rola:
 - przechowywanie dużej ilości danych w sposób trwały i niezawodny
- Trwałość:
 - dane powinny być przechowywane w pamięci tak długo, jak tego wymagają użytkownicy systemu baz danych
- Urządzenia pamięci zewnętrznej:
 - dyski magnetyczne, dyski optyczne, urządzenia do archiwizacji

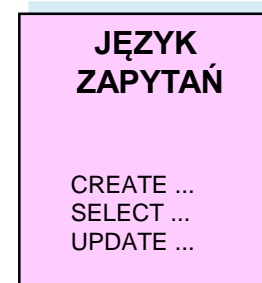
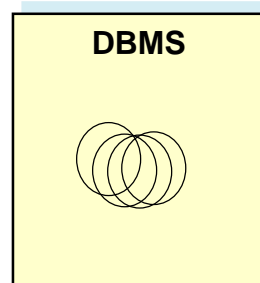


- Niezawodność:
 - częstotliwość występowania błędów powinna być zanedbywalnie mała
- Środki służące zwiększeniu niezawodności:
 - dublowanie urządzeń pamięci zewnętrznych
 - kontrola poprawności zapisu
 - kody detekcji/korekcji błędów



- Język wysokiego poziomu, za pomocą którego użytkownik komunikuje się z bazą danych

SYSTEM BAZ DANYCH



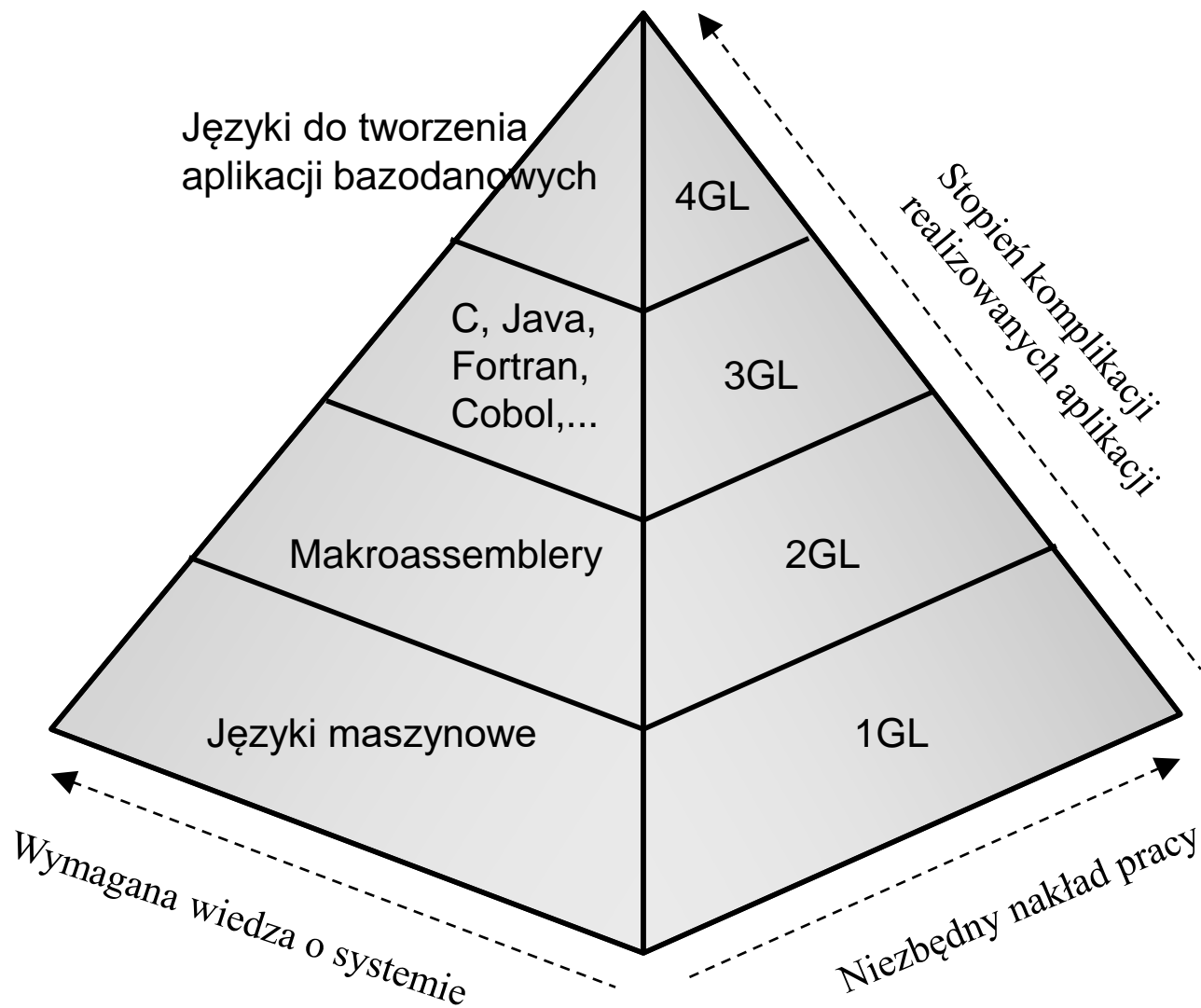


- Dla użytkowników interakcyjnych:
 - język zapytań ad hoc, z poleceniami wybieranymi z menu lub wprowadzanymi w wierszu poleceń
- Dla programistów aplikacyjnych:
 - kompletny język programowania (język czwartej generacji, 4GL) zawierający:
 - instrukcje definiowania i operowania na danych z bazy danych,
 - instrukcje wejścia/wyjścia,
 - instrukcje i struktury danych typowe dla języków programowania

Typowy język programowania (PL/1, Cobol, C, Java, ...) rozszerzony o instrukcje operujące na bazie danych (instrukcje języka zapytań zanurzonego w języku macierzystym). Instrukcje te mogą mieć postać wywołań interfejsu ODBC/JDBC, w których przekazywana jest tekstowo treść zapytania.



- Język zapytań = Język definiowania danych (*Data Definition Language*) + Język manipulowania danymi (*Data Manipulation Language*) + Język kontrolowania danych (*Data Control Language*)
- DDL – definiuje struktury danych, na których operują instrukcje DML
- DML – wykonuje operacje na danych zdefiniowanych przez DDL (pobieranie danych i ich aktualizowanie)
- DCL – kontroluje uprawnienia dostępu do danych





- Słownik danych – specjalna baza danych zawierająca informacje o danych przechowywanych w innych bazach danych (metadane)
- Zawartość słownika danych:
 - opisy struktury obiektów bazy danych (np. schematów tablic, widoków)
 - opisy użytkowników (identyfikatory, uprawnienia)
 - definicje formatów danych stosowanych w bazie danych
 - dane statystyczne o wartościach danych w bazie danych
- Dostęp do słownika danych:
 - zwykły użytkownik ma prawa tylko do odczytu, i to tylko pewnych fragmentów słownika danych
 - DBA może mieć pełniejszy dostęp do słownika, jednak jego modyfikowanie może wykonywać tylko poprzez procedury systemowe
 - DBMS ma pełen dostęp do słownika danych (np. modyfikuje go przy tworzeniu nowego obiektu w bazie danych)

Dziękuję za uwagę