

# BD2 – Bazy Danych 2

dr inż. Tomasz Traczyk

## 1. Systemy informacyjne z bazami danych

Copyright © Tomasz Traczyk  
Instytut Automatyki i Informatyki Stosowanej Politechniki Warszawskiej  
Materiały dydaktyczne przeznaczone są wyłącznie do indywidualnego użytku studiujących.  
Rozpowszechnianie kopii bez pisemnej zgody autora jest zabronione.

### Pojęcie bazy danych

---

#### Czym jest baza danych

- Pamięć trwała danych (*persistent data*)
- Określona struktura i reguły integralności

#### Cele użytkowania b.d.

- Niezawodność zapisu
- Integralność danych
- Sprawność zapytań
- Wygodne interfejsy
- Wielodostęp
- Zabezpieczenia dostępu

## System zarządzania bazą danych (DBMS)

---

### System zarządzania bazą danych (*Database Management System*)

- Program lub zbiór programów działający na „serwerze bazy danych”
- Pośredniczy (koniecznie!) w uzyskaniu dostępu do danych w bazie

### Rola DBMS

- Izolowanie programów korzystających z danych od reprezentacji fizycznej tych danych
  - model danych
  - słownik danych
  - język opisu danych
- Mechanizmy dostępu do danych
  - języki zapytań i manipulacji danymi
  - optymalizacja dostępu
- Ochrona danych
  - autoryzacja dostępu
  - ochrona spójności
  - mechanizmy odtwarzania po awarii
- Wielodostęp
  - zarządzanie transakcjami
- Dostęp przez sieć
  - różne architektury i interfejsy dostępu
  - mechanizmy dla rozproszonych b.d.

## Systemy informacyjne z bazami danych

---

Współczesne profesjonalne systemy informacyjne z reguły wykorzystują bazy danych. W ogromnej większości przypadków stosowane są systemy relacyjne (RDBMS).

### Dlaczego baza danych stanowi dobry fundament s.i.?

- Bezpieczeństwo
  - zabezpieczenia dostępu
  - niezawodność
  - ograniczenia integralności
  - możliwość scentralizowanego administrowania danymi
  - przetwarzanie transakcyjne
- Wydajność
  - wydajne przetwarzanie transakcji
  - szybkie wyszukiwanie
- Wielodostęp
- Otwartość
  - elastyczność dostępu do danych
  - współdziałanie z różnymi źródłami danych
- Możliwości rozwoju
  - skalowalność
  - przenośność
  - możliwości pracy rozproszonej

## Składniki systemów informacyjnych z bazami danych

---

### Baza danych

- DBMS
- Metadane (słownik)
  - struktura danych
  - reguły integralności
  - prawa dostępu
- Dane

### Języki dostępu do danych

- Budowa
  - proceduralne
  - nieproceduralne
- Typ dostępu
  - nawigacyjne (rekord bieżący)
  - zwracające struktury (np. tabele)

### Aplikacja

- Warstwa sieciowa
- Środowiska wykonania aplikacji (DBMS, *runtime*, serwery aplikacyjne, kontenery itp.)
- Moduły aplikacji

## Główne typy s.i. z bazami danych

### Systemy transakcyjne

#### (OLTP – *On Line Transaction Processing*)

- Główne zadanie: gromadzenie danych
- Typowe środowisko: operacyjne bazy danych
- Typowe operacje
  - wielka liczba niewielkich transakcji modyfikujących dane
  - operacje zapisu i odczytu – interaktywne i wsadowe
- Główne problemy
  - wielodostęp
  - konieczność stałego utrzymania spójności danych
  - maksymalizacja średniej wydajności
- Tradycyjna domena relacyjnych baz danych

### Systemy analityczne

#### (OLAP – *On Line Analytical Processing*)

- Główne zadanie: analiza danych do celów wspomagania decyzji
- Typowe środowisko: hurtownie danych
- Typowe operacje
  - niewielka liczba wielkich „transakcji” odczytujących dane
  - operacje odczytu – głównie interaktywne
  - operacje zapisu – jedynie wsadowe
- Główne problemy
  - wielka ilość danych (VLDB – *Very Large Databases*)
  - analiza wielowymiarowa
- Nowoczesne RDBMS zawierają mechanizmy wspomagające VLDB i przetwarzanie analityczne

## Architektury s.i. z bazami danych

### Architektura terminalowa

- Terminale bez żadnej inteligencji
- Całe przetwarzanie w bazie centralnej
- Architektura spotykana głównie w środowisku *mainframe*, obecnie rzadko stosowana

### Architektura wielokomputerowa ze współdzieleniem plików

- Pełne przetwarzanie na każdym z komputerów
- Współdzielenie plików z danymi
- Model „dane do zapytania”
- Architektura ta nie powinna być stosowana w systemach profesjonalnych

## Architektury s.i. z bazami danych, c.d.

### Architektura klient-serwer

- Podział zadań
  - serwer
    - \* DBMS + dane
    - \* realizacja zapytań
    - \* realizacja ograniczeń
    - \* przetwarzanie danych
  - klient
    - \* obsługa prezentacji (GUI)
    - \* wykonywanie logiki aplikacji
    - \* przetwarzanie danych
- Zalety
  - bezpieczeństwo serwera
  - minimalizacja ruchu w sieci
  - model „zapytanie do danych”
  - możliwość przetwarzania danych bezpośrednio na serwerze
  - odciążenie centralnego komputera od obsługi interfejsu (ważne dla GUI)
- możliwość budowy sporych systemów bez użycia wielkich komputerów
- Wady
  - konieczne administrowanie wieloma PC
  - trudne administrowanie aplikacjami (np. zmiany wersji)
  - zmiany oprogramowania klientów, wymuszające częstą wymianę sprzętu
  - problemy z bezpieczeństwem (lokalne kopie danych, wirusy)
  - brak kontroli nad działaniami pracowników
  - brak kontroli licencji
  - bardzo duże koszty eksploatacji PC (USA: 40 k\$/rok)
- Do niedawna najpopularniejsza architektura, nadal powszechnie stosowana

## Architektury s.i. z bazami danych, c.d.

### Architektura wielowarstwowa

- Podział zadań
  - serwer danych
    - \* jak w architekturze klient-serwer
  - serwer aplikacyjny (lub wiele serwerów)
    - \* logika aplikacji
    - \* przetwarzanie danych
    - \* funkcje serwera HTTP (opcjonalnie)
  - serwer HTTP (opcjonalny)
    - \* nasłuchiwanie żądań HTTP
    - \* przekazywanie ich do odpowiedniego serwera aplikacyjnego
    - \* przesyłanie odpowiedzi
  - klient
    - \* obsługa prezentacji
    - \* wykorzystanie przeglądarki HTML – cienki klient
    - \* wykonywanie części logiki aplikacji (aplety) – gruby klient
- Zalety
  - większość zalet architektury klient-serwer
  - bezpieczeństwo aplikacji (centralizacja)
  - łatwość administrowania aplikacjami
  - brak konieczności administrowania konfiguracją klientów
- Wady
  - potrzebny silny sprzęt na serwery aplikacyjne
  - trudniejsze technologie
  - dla cienkiego klienta – ograniczenia w funkcjonalności interfejsu użytkownika
  - większy ruch sieciowy
- Architektura powszechnie stosowana w systemach inter- i intranetowych (najczęściej trójwarstwowa)
- Wypiera architekturę klient-serwer w zastosowaniach korporacyjnych

## Architektury s.i. z bazami danych, c.d.

### Rozproszona

- Podział zadań
  - serwer(y) danych – jak w poprzednich architekturach
  - aplikacja – współdziałanie obiektów rozproszonych w sieci
  - klienci – jak w architekturze wielowarstwowej
- Zalety
  - możliwość wykorzystania gotowych komponentów istniejących w sieci
  - dostosowanie do obliczeń rozproszonych
  - niemal nieograniczona skalowalność
- Wady
  - trudne technologie
  - nadmiarowość w stosunku do typowych współczesnych potrzeb
- Technologie
  - CORBA
  - DCOM
  - EJB
  - *Web Services*
  - .Net
- Architektura rzadko stosowana, ale uważana za przyszłościową