

Wprowadzenie do problematyki baz danych

**Wykład przygotował:
Robert Wrembel**

Niniejszy cykl 13 wykładów będzie poświęcony bazom danych.

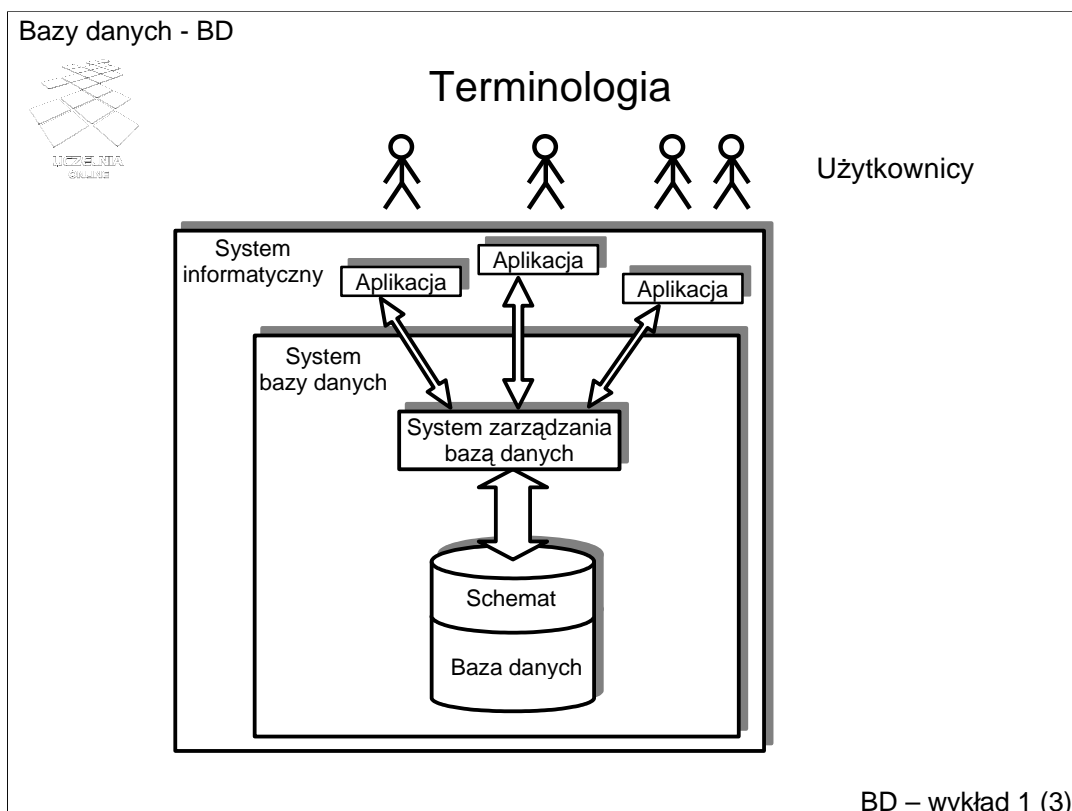


Plan wykładu

- Podstawowa terminologia
- Charakterystyka baz danych
- Modele danych
- Użytkownicy baz danych
- System zarządzania bazą danych (SZBD)
- Klasyfikacja baz danych

Celem pierwszego wykładu jest ogólne wprowadzenie do problematyki baz danych. Zostaną tu omówione:

- podstawowa terminologia
- charakterystyka baz danych
- wymagania stawiane bazom danych
- cechy technologii baz danych
- cechy systemu zarządzania bazą danych
- wprowadzenie do modeli danych
- charakterystyka użytkowników baz danych
- charakterystyka sposobów korzystania z bazy danych
- architektury: wewnętrzna i komunikacyjna baz danych
- ogólny podział baz danych



Zbiór danych opisujący pewien wybrany fragment rzeczywistości będziemy nazywać bazą danych. Przykładowo, bazą danych może być zbiór danych banku na temat klientów, ich rachunków, operacji na rachunkach, udzielanych kredytach.

Dane w bazie danych posiadają dwie podstawowe cechy. Po pierwsze, odzwierciedlają rzeczywistość w sposób z nią zgodny (prawidłowy). Po drugie, są zorganizowane w specyficzny sposób, zgodnie z tzw. modelem danych (model danych zostanie omówiony w dalszej części kursu).

Struktura danych i powiązania między nimi są opisane przez tzw. schemat bazy danych.

Baza danych jest zarządzana przez tzw. system zarządzania bazą danych, w skrócie SZBD. Funkcje oferowane przez SZBD zostaną omówione w dalszej części wykładu.

SZBD i bazę danych będziemy dalej nazywać systemem bazy danych.

Z systemem bazy danych współpracują programy użytkowników, zwane aplikacjami. Zadaniem tych programów jest przetwarzanie danych, tj. wstawianie nowych danych, modyfikowanie danych już istniejących, usuwanie danych nieaktualnych, wyszukiwanie danych.

Wszystkie omówione wyżej komponenty (tj. baza danych, SZBD i aplikacje) wchodzi w skład tzw. systemu informatycznego.



Charakterystyka baz danych (1)

1. Trwałość danych
 - Długi czas życia – kilka, kilkadziesiąt, kilkaset lat
 - Niezależność od działania aplikacji
2. Rozmiar wolumenu danych
 - Dane nie mieszczą się w pamięci operacyjnej – wymagana pamięć zewnętrzna
 - Danych jest zbyt dużo dla ich liniowego przeglądania przez użytkowników

Do głównych cech charakteryzujących bazę danych zalicza się: trwałość danych, rozmiar wolumenu danych i złożoność danych.

Trwałość danych oznacza, że dane przechowywane w bazie danych nie są ulotne. W konsekwencji, okres przechowywania danych jest ograniczony wyłącznie okresem żywotności nośnika danych. "Czas życia" danych, po ich zapisaniu do bazy danych jest niezależny od istnienia i działania bądź niedziałania aplikacji. Trwałość danych jest również niezależna od platformy sprzętowo-programowej.

W ogromnej większości zastosowań, dane zgromadzone w bazie danych nie mieszczą się w pamięci operacyjnej, więc do ich składowania jest wymagana pamięć zewnętrzna (dyskowa, optyczna, taśmowa). Tak duże ilości danych nie mogą być przeglądane liniowo ze względu na niewielką efektywność tej techniki. W konsekwencji konieczne jest wykorzystanie innych zaawansowanych mechanizmów efektywnego dostępu do danych.



Charakterystyka baz danych (2)

3. Złożoność danych

- Złożoność strukturalna i złożoność zależności pomiędzy danymi
- Złożoność semantyczna
- Ograniczenia integralnościowe

Dane gromadzone w bazie danych często są złożone ze względu na:

- złożoność ich struktur i zależności pomiędzy danymi (np. projekt samochodu, złożony z tysięcy elementów),
- złożoność semantyczną danych (np. fakt przyznania kredytu mieszkaniowego jest uzależniony od spełnienia lub niespełnienia wielu wymagań przez petenta).

Ponadto, na dane są często nakładane tzw. ograniczenia integralnościowe gwarantujące, że w bazie danych pojawią się wyłącznie dane spełniające te ograniczenia. Ograniczenia takie mogą być również bardzo złożone.



Wymagania (1)

- Spójność bazy danych
- Efektywne przetwarzanie danych
- Poprawne modelowanie świata rzeczywistego
- Autoryzacja dostępu do danych
- Współbieżność dostępu do danych
- Metadane

Bazie danych stawia się 6 podstawowych wymagań.

Po pierwsze, musi ona gwarantować spójność danych.

Po drugie, musi zapewniać efektywne przetwarzanie danych.

Po trzecie, musi poprawnie odzwierciedlać zależności w świecie rzeczywistym, który baza danych reprezentuje.

Po czwarte, musi chronić przed nieautoryzowanym dostępem.

Po piąte, musi zapewniać współbieżny dostęp do danych wielu użytkownikom.

Po szóste, musi udostępniać tzw. metadane.



Wymagania (2)

1. Spójność bazy danych

- Poprawność danych z punktu widzenia przyjętych kryteriów
 - wierne odzwierciedlenie danych rzeczywistych
 - spełnienie ograniczeń nałożonych przez użytkowników

Spójność bazy danych jest definiowana jako poprawność danych z punktu widzenia pewnych przyjętych kryteriów. Definiując spójność wymienia się trzy takie kryteria, tj.:

- wierne odzwierciedlenie danych rzeczywistych,
- spełnienie ograniczeń nałożonych przez użytkowników,
- brak anomalii wynikających ze współbieżnego dostępu do danych.

Kryterium pierwsze należy interpretować następująco: dane przechowywane w bazie danych są takie jak w świecie rzeczywistym, który ta baza reprezentuje. Przykładowo, w bazie danych dziekanatu Wydziału Informatyki i Zarządzania PP są przechowywane dane tylko tych studentów, którzy kiedykolwiek ukończyli studia i dane tylko studentów aktualnie studiujących. Innymi słowy, baza ta nie zawiera danych studentów nieistniejących. Ponadto, zależności między danymi wiernie odzwierciedlają zależności pomiędzy obiektami świata rzeczywistego. Przykładowo, grupa studencka G1 w bazie danych dziekanatu składa się ze studentów należących do tej grupy w świecie rzeczywistym.

Kryterium drugie oznacza, że wszystkie dane w bazie, na które nałożono pewne ograniczenia integralnościowe muszą te ograniczenia spełniać. Przykładowo, w bazie danych dziekanatu zdefiniowano ograniczenie na możliwe wartości oceny i określono, zbiór dozwolonych ocen jako {2, 3, 4, 5}. Oznacza to, że w bazie danych nie pojawi się żadna inna ocena niż dozwolona.



Wymagania (3)

- Spójność bazy danych cd.
 - Odporność na anomalie będące wynikiem współbieżności dostępu do baz danych
 - Odporność na błędy, awarie i inne anormalne sytuacje wynikające z zawodności środowiska sprzętowo-programowego
 - Odporność na błędy użytkowników

W przypadku baz danych, z których korzysta przynajmniej dwóch użytkowników mogą powstać dane niepoprawne na skutek równoczesnego modyfikowania tego samego zbioru danych. Baza danych musi być odporna na takie sytuacje niepoprawne.

Spójność to również poprawność danych w przypadku awarii sprzętowo-programowych. W sytuacji wystąpienia awarii, w bazie danych nie mogą powstać dane niepoprawne. Ponadto, żadne dane nie mogą zostać utracone.

Baza danych powinna być również odporna na przypadkowe błędy użytkowników, np. usunięcie danych. W takiej sytuacji musi istnieć mechanizm naprawienia błędu, wycofania akcji użytkownika.



Wymagania (4)

2. Efektywne przetwarzanie danych

- Efektywne metody dostępu do danych
- Optymalizacja metod dostępu do danych
- Niezależność aplikacji od fizycznych metod dostępu

Drugim wymaganiem stawianym bazie danych jest zapewnienie efektywnego przetwarzania danych, tj. wstawiania nowych danych, modyfikowania istniejących, usuwania i wyszukiwania danych. W tym celu koniecznej jest wykorzystywanie efektywnych metod dostępu do danych z wykorzystaniem specjalizowanych struktur i optymalizacja metod dostępu. Ponadto, program, czy użytkownik korzystający z bazy danych nie zna fizycznej organizacji danych na nośniku. W związku z tym, optymalizacja dostępu powinna być realizowana przez wyspecjalizowany moduł programowy i powinna być niewidoczna dla użytkownika.



Wymagania (5)

3. Poprawne modelowanie świata rzeczywistego
 - Wspomaganie procesu projektowania i utrzymania bazy danych
 - Różne poziomy modelowania danych
 - Transformacje między modelami danych
4. Autoryzacja dostępu do danych
 - użytkownicy z hasłami dostępu
 - użytkownicy i ich uprawnienia

BD – wykład 1 (10)

Trzecim wymaganiem stawianym bazie danych jest poprawne modelowanie świata rzeczywistego. Oznacza to, że struktura bazy danych musi odzwierciedlać we właściwy/poprawny sposób obiekty świata rzeczywistego i powiązania pomiędzy tymi obiektami. Przykładowo, jeżeli dealer samochodowy sprzedaje samochody osobowe i dostawcze w różnych konfiguracjach, to baza danych dla tego dealera musi umożliwiać przechowywanie danych na temat samochodów i osobowych i dostawczych, oraz konfiguracji poszczególnych modeli.

Producenci systemów zarządzania bazami danych oferują narzędzia wspomagające procesy modelowania danych, projektowania bazy danych i transformacje pomiędzy różnymi modelami.

Czwartym wymaganiem jest autoryzacja dostępu do danych. Oznacza to, że dostęp do bazy danych mają tylko jej użytkownicy identyfikowani unikalną nazwą i hasłem. Ponadto, każdy użytkownik posiada określone uprawnienia w bazie danych.



Wymagania (6)

5. Współbieżność dostępu do danych
 - równoczesny dostęp do tych samych danych przez wielu użytkowników
 - konflikt odczyt-zapis, zapis-zapis
6. Metadane
 - dane o danych, strukturach dostępu, użytkownikach i ich prawach

Piątym wymaganiem jest zagwarantowanie możliwości równoczesnej pracy wielu użytkowników tej samej bazy danych. Co więcej, użytkownicy ci mogą jednocześnie pracować z tym samym zbiorem danych. W takim przypadku mogą powstać konflikty w dostępie do danych, gdy jeden użytkownik modyfikuje zbiór danych, a drugi próbuje ten sam zbiór odczytać lub zmodyfikować. Baza danych musi zapewnić poprawne rozwiązanie tego typu konfliktów.

Szóstym wymaganiem jest wsparcie dla tzw. metadanych. Metadane to najprościej mówiąc dane o bazie danych. Dane te opisują m.in.: dane przechowywane w bazie, struktury danych, użytkowników i ich uprawnienia.



Technologia baz danych (1)

1. Fizyczne struktury danych i metody dostępu
 - Pliki uporządkowane, haszowe, zgrupowane, indeksy drzewiaste i bitmapowe
 - Metoda połowienia binarnego, haszowanie statyczne i dynamiczne, metody połączenia, sortowanie, grupowanie
 - Składniowe i kosztowe metody optymalizacji dostępu
 - Fizyczna niezależność danych

Omówione wymagania odnośnie baz danych są zapewniane w ramach tzw. technologii baz danych. Oferuje ona m.in.

fizyczne struktury i metody dostępu. Do fizycznych struktur wykorzystywanych w bazach danych zalicza się pliki uporządkowane, pliki haszowe, pliki zgrupowane, indeksy drzewiaste i indeksy bitmapowe. Do metod dostępu zalicza się: połowienie binarne, haszowanie statyczne i dynamiczne, algorytmy łączenia, sortowania i grupowania.

Dostęp do danych z wykorzystaniem struktur fizycznych i metod dostępu jest optymalizowany za pomocą zaawansowanych technik optymalizacji składniowej i kosztowej.

Ponadto, fizyczna organizacja danych na dysku nie ma wpływu na działanie aplikacji/programów użytkowników korzystających z bazy danych. Oznacza to, że zmiana fizycznej organizacji danych np. o klientach banku, po pierwsze, jest niewidoczna dla użytkownika i po drugie, nie wymaga zmiany kodu aplikacji. Innymi słowy aplikacja działa tak samo dobrze jak poprzednio.



Technologia baz danych (2)

2. Przetwarzanie transakcyjne (spójność baz danych)
 - Dostęp do bazy danych za pomocą transakcji o własnościach ACID
 - Metody synchronizacji transakcji (2PL, znaczniki czasowe, wielowersyjność danych)
 - Metody odtwarzania spójności bazy danych (plik logu, odtwarzanie i wycofywanie operacji, Write Ahead Log, punkty kontrolne)
 - Archiwizacja bazy danych i odtwarzanie po awarii

Technologia baz danych oferuje wsparcie dla tzw. przetwarzania transakcyjnego, zapewniającego spójność całej bazy danych.

W ramach tego przetwarzania każdy dostęp do bazy danych jest realizowany w ramach pewnej jednostki interakcji, zwanej transakcją. Posiada ona cechy atomowości, spójności, izolacji i trwałości (problematyka transakcji zostanie omówiona w osobnym wykładzie). Transakcje działające równocześnie w systemie muszą być synchronizowane za pomocą specjalizowanych algorytmów (2PL, znaczników czasowych) i stosowania wersji danych.

Zapewnienie spójności danych, np. w przypadku konfliktu transakcji lub awarii sprzętowo-programowych, często wymaga wycofania zmian w bazie danych. Do tego celu konieczne są dodatkowe struktury danych, algorytmy i mechanizmy systemowe.

Awaria sprzętowo-programowa nie może spowodować utraty żadnych danych. W celu zapewnienia tego wymagania stosuje się techniki i systemowe mechanizmy archiwizowania bazy danych i jej odtwarzania po awarii.



Technologia baz danych (3)

3. Modele danych
 - Modele pojęciowe (model związków-encji, UML)
 - Modele logiczne (relacyjny, obiektowy, obiektowo-relacyjny, semistrukturalny, hierarchiczny, sieciowy)
4. Narzędzia programistyczne
 - Języki budowy aplikacji
 - Narzędzia modelowania i projektowania
 - Metodyki projektowania

Technologia baz danych oferuje wsparcie dla wielu modeli danych, czyli wielu sposobów reprezentowania danych. Wyróżnia się tu:

- modele pojęciowe (np. związków-encji, UML),
- modele logiczne (np. relacyjny, obiektowy, obiektowo-relacyjny, semistrukturalny, hierarchiczny, sieciowy).

Oprócz technik związanych z zarządzaniem danymi, technologia baz danych oferuje narzędzia programistyczne do budowania aplikacji, modelowania i projektowania bazy danych. Narzędzie te wspierają uznane metodyki projektowania.



System Zarządzania Bazą Danych (SZBD)

- Oprogramowanie zarządzające całą bazą danych
- Funkcjonalność
 - Język bazy danych - tworzenie, definiowanie, wyszukiwanie i pielęgnacja danych w bazie danych
 - Struktury danych - efektywne składowanie i przetwarzanie dużych wolumenów danych
 - Optymalizacja dostępu do danych
 - Współbieżny dostęp do danych
 - Zapewnienie bezpieczeństwa danych zagrożonego awaryjnością środowiska sprzętowo-programowego
 - Autoryzacja dostępu do danych
 - Wielość interfejsów dostępu do bazy danych

BD – wykład 1 (15)

Jak wspomniano przy okazji omawiania slajdu nr 3, jednym z komponentów systemu bazy danych jest tzw. System Zarządzania Bazą Danych (SZBD). Z technologicznego punktu widzenia jest to moduł programowy, którego zadaniem jest zarządzanie całą bazą danych oraz realizowanie żądań aplikacji użytkowników. Podstawowa funkcjonalność SZBD obejmuje:

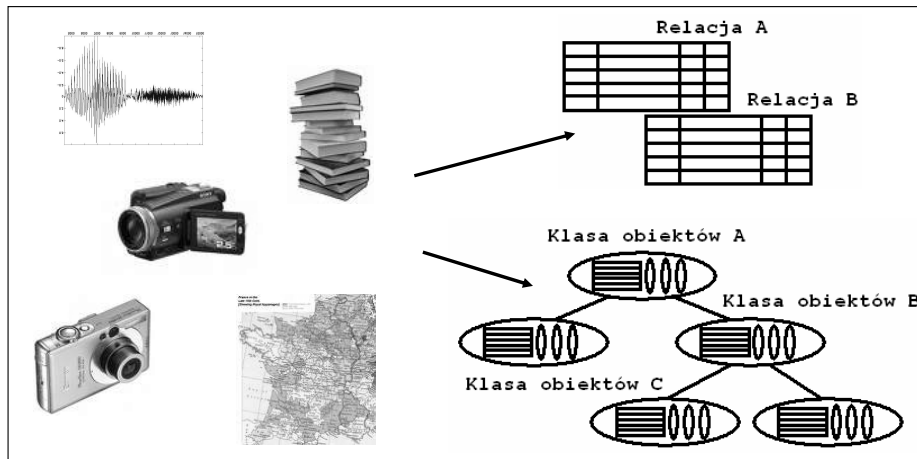
- po pierwsze, wsparcie dla języka bazy danych, który umożliwia m.in. wstawianie, modyfikowanie, usuwanie i wyszukiwanie danych oraz tworzenie, modyfikowanie i usuwanie struktur danych;
- po drugie, wsparcie dla struktur danych zapewniających efektywne składowanie i przetwarzanie dużych wolumenów danych;
- po trzecie, optymalizację dostępu do danych;
- po czwarte, synchronizację współbieżnego dostępu do danych;
- po piąte, zapewnienie bezpieczeństwa danych w przypadku awarii sprzętowo-programowej;
- po szóste, autoryzację dostępu do danych;
- po siódme, wielość interfejsów dostępu do bazy danych.



Model danych (1)

Obiekty świata rzeczywistego

Obiekty modelu danych



BD – wykład 1 (16)

Obiekty ze świata rzeczywistego są reprezentowane w bazie danych za pomocą tzw. modelu danych.

Wyróżnia się następujące modele danych: hierarchiczny, sieciowy, relacyjny, obiektowy, obiektowo-relacyjny, semistrukturalny.

Model hierarchiczny i sieciowy nie są już stosowane w nowo-budowanych systemach. Obecnie w bazach danych najczęściej stosuje się model relacyjny, obiektowo-relacyjny lub semistrukturalny.



Model danych (2)

1. Struktury danych



JAN	NOWAK	47
TADEUSZ	KOWALSKI	34
MACIEJ	NOWAK	26
JANINA	RZEPA	19
KUBA	TARZAN	31
JÓZEF	MALINIAK	29
JAN	NOWAK	56

Każdy model danych definiuje trzy podstawowe elementy, tj. struktury danych, operacje na danych i ograniczenia integralnościowe nakładane na dane.

Struktura danych służy do reprezentowania w bazie danych obiektów ze świata rzeczywistego. Przykładowo, grupa pracowników firmy może być reprezentowana w modelu obiektowym jako klasa, lub w modelu relacyjnym jako relacja. Poszczególni pracownicy są reprezentowani odpowiednio jako wystąpienia klasy (w modelu obiektowym) lub krotki relacji (w modelu relacyjnym).



Model danych (3)

2. Operacje (operatory modelu danych)

3. Ograniczenia integralnościowe

Nazwa projektu	Budżet	Data rozpoczęcia	Data zakończenia
Indeksy w BD	500 000,-	1.07.2002	30.06.2005
Magazyny danych	700 000,-	1.09.2002	31.08.2001

Każdy model danych posiada zbiór predefiniowanych operacji na danych. Przykładowo, w modelu relacyjnym operacje na danych oferowane przez model to: selekcja, projekcja, połączenie i operacje na zbiorach.

Ponadto, model danych umożliwia nałożenie ograniczeń integralnościowych na dane reprezentowane w nim dane. Przykładowo, dla relacji ze slajdu można zdefiniować ograniczenie integralnościowe zapewniające, że data rozpoczęcia projektu będzie zawsze mniejsza niż data jego zakończenia.



Przykładowa baza danych

Pracownicy

IdPrac	Nazwisko	Etat	Szef	DataZat	Płaca	Premia	IdZesp
7340	Kowalski	referent	7548	17.12.90	1800		20
7341	Nowak	asystent	7340	20.02.91	1600	300	30
7342	Tarzan	asystent	7340	22.02.91	1500	500	20
7544	Colargol	kierownik	7600	2.04.91	2500	100	20
7548	Król	księgowy	7600	28.09.91	3500		10
7600	Dziuba	dyrektor		17.10.89	5000		10
7880	Buba	referent	7544	23.05.95	2100		30
7900	Misiek	kierownik	7600	29.10.01	2700		30

Etaty

Nazwa	PłacaMin	PłacaMax
dyrektor	4000	9999
kierownik	2000	5000
referent	1500	2500
asystent	1100	1800
księgowy	2500	4500

Zespoły

IdZesp	Nazwa	Adres
10	Administracyjny	Poznań
20	Sprzedaży	Poznań
30	Reklamy	Gniezno
40	Badań	Oborniki

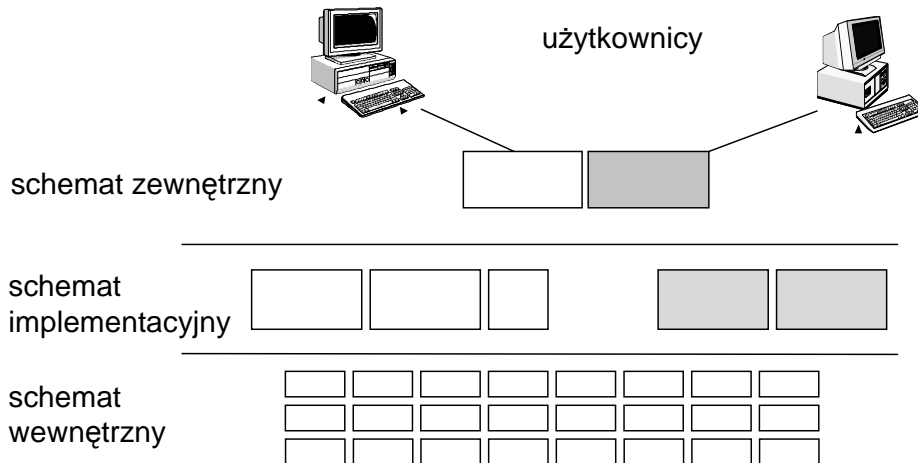
BD – wykład 1 (19)

Przykład prostej bazy danych zaimplementowanej w modelu relacyjnym przedstawiono na slajdzie. Strukturami danych modelu są w tym przypadku trzy relacje: *Pracownicy*, *Zespoły*, *Etaty*. Pierwsza z nich przechowuje dane o pracownikach, druga - o zespołach, w których ci pracownicy są zatrudnieni, a trzecia - zawiera katalog widełek płacowych.



Architektura systemu bazy danych

- 3-warstwowa architektura wg standardu ANSI/SPARC



BD – wykład 1 (20)

Podstawowa architektura systemu bazy danych została zdefiniowana w standardzie ANSI/SPARC. Wyróżnia się w niej 3 następujące tzw. schematy: wewnętrzny, implementacyjny, zewnętrzny.

Schemat wewnętrzny opisuje fizyczny sposób składowania danych na nośnikach. Schemat implementacyjny odwzorowuje schemat wewnętrzny w struktury modelu danych wykorzystywanego w bazie danych. W modelu relacyjnym schemat wewnętrzny jest odwzorowywany w schemat relacyjny. Wreszcie, schemat zewnętrzny stanowi interfejs użytkownika do bazy danych. Schemat ten odwzorowuje schemat implementacyjny w schemat poprzez, który użytkownik widzi bazę danych i pracuje z nią. Należy podkreślić, że schemat zewnętrzny nie zawsze jest stosowany.



Użytkownicy SBD (1)

- Aktorzy na scenie
 - Użytkownicy końcowi
 - Programiści aplikacji
 - Projektanci baz danych
 - Analitycy systemowi
 - Administratorzy systemów baz danych

Z użytkowaniem bazy danych na różnych etapach są związane różne grupy użytkowników. Wyróżnia się tu: użytkowników końcowych, programistów aplikacji, projektantów baz danych, analityków systemowych i administratorów.

Użytkownicy końcowy charakteryzują się tym, że korzystają z bazy danych głównie poprzez gotowe aplikacje/programy. Ich wiedza zwykle obejmuje sposób obsługi aplikacji i znajomość zagadnień z zakresu obowiązków służbowych.

Zadaniem programistów jest implementowanie aplikacji dla użytkowników końcowych. Funkcjonalność tych aplikacji wynika z wymagań użytkowników.

Projektanci baz danych zajmują się projektowaniem struktury logicznej bazy danych, czyli struktur modelu danych i projektowaniem struktury fizycznej bazy danych, czyli doбором parametrów fizycznego składowania danych na nośnikach. Ponadto, ich zadaniem jest przygotowanie działającej bazy danych.

Analitycy systemowi zajmują się analizą wymagań systemu bazy danych i aplikacji. Wynik ich pracy jest podstawą opracowania struktury logicznej (a często również fizycznej) bazy danych i podstawą dla programistów aplikacji.

Administratorzy systemu bazy danych są odpowiedzialni m.in. za: przygotowanie systemu do pracy produkcyjnej, zagwarantowanie ciągłości pracy systemu, zarządzanie użytkownikami i instalowanie nowych wersji systemu.



Użytkownicy SBD (2)

- Aktorzy poza sceną
 - Administratorzy serwerów, sieci komputerowych
 - Projektanci i programiści SZBD
 - Projektanci narzędzi deweloperskich

Ponadto, istnieją jeszcze trzy inne grupy użytkowników, których praca nie dotyczy bezpośrednio samej bazy danych, ale bez których system bazy danych nie będzie działał. Do grup tych zaliczamy: administratorów serwerów i sieci komputerowych, projektantów i programistów SZBD, projektantów narzędzi deweloperskich baz danych i aplikacji.



Interakcja z bazą danych (1)

- Język SQL
 - jedyny sposób interakcji z bazą danych
 - język deklaratywny
 - ustandaryzowany
 - producenci systemów komercyjnych i niekomercyjnych starają się implementować ten standard

```
select nazwisko, etat, płaca
from pracownicy
where idzesp=30
and etat='kierownik'
```

Jakakolwiek interakcja programu użytkowego (aplikacji) z bazą danych odbywa się za pomocą języka SQL. Jest to jedyny sposób komunikowania się aplikacji z bazą danych. SQL jest językiem deklaratywnym. Oznacza to, że posługując się nim specyfikujemy tylko co chcemy otrzymać. Nie specyfikujemy sposobu (algorytmu) w jaki ma być zrealizowane zadanie. Przykładem polecenia SQL może być zapytanie do bazy danych poszukujące informacje o klientach banku z Poznania, którzy w ciągu ostatniego miesiąca wypłacili z bankomatu łącznie powyżej 8000 PLN. W tym zapytaniu specyfikujemy tylko jakie dane nas interesują. Sposób ich wyszukania jest automatycznie dobierany przez SZBD.

SQL jest językiem ustandaryzowanym. Jego standaryzacją zajmuje się specjalny międzynarodowy komitet, w skład którego wchodzi przedstawiciele największych producentów SZBD (IBM, Microsoft, Oracle). Dotychczas opracowano trzy standardy języka SQL, kolejno rozszerzające jego funkcjonalność. Standardy te to: SQL-92, SQL-99, SQL-2003.

Producenci systemów komercyjnych i niekomercyjnych starają się implementować przynajmniej standard SQL-92. Należy jednak pamiętać, że nie ma 100% zgodności implementacji.

Przykład prostego polecenia SQL będącego zapytaniem do bazy danych przedstawiono na slajdzie. Zapytanie to wyszukuje pracowników (nazwisko, etat, płaca) zatrudnionych w zespole o numerze 30 na etacie kierownika.



Interakcja z bazą danych (2)

- Aplikacje
 - formularze
 - elektroniczne formularze z polami, listami, elementami wyboru
 - umożliwiają wstawianie, modyfikowanie, usuwanie, wyszukiwanie danych
 - raporty
 - umożliwiają prezentowanie zawartości bazy danych
 - teksty
 - wykresy
 - grafika

Język SQL jest narzędziem dostępu do bazy danych stosowanym głównie przez projektantów aplikacji, projektantów baz danych i administratorów baz danych. Standardowym sposobem korzystania z bazy danych przez użytkowników końcowych są aplikacje. Należy jednak pamiętać, że na poziomie programistycznym aplikacje również komunikują się z bazą danych za pomocą poleceń SQL.

Ze względu na funkcjonalność, wyróżnia się dwa rodzaje aplikacji, tj. formularze i raporty. Aplikację pierwszego rodzaju należy postrzegać jako elektroniczny formularz (z polami, listami, elementami wyboru) wypełniany przez użytkownika. Formularze umożliwiają pełną obsługę danych, tj. wstawianie, modyfikowanie, usuwanie i wyszukiwanie.

Raporty umożliwiają wyłącznie odczytywanie danych z bazy i prezentowanie ich w różnej postaci, głównie tekstu lub wykresu.



Formularz - przykład

Oracle Forms Runtime - [Zespoły i pracownicy]

Action Edit Query Block Record Field Window Help

Zespoły

Nazwa: SYSTEMY EKSPERCKIE Adres: STRZELECKA 14

Pracowników w zespole: 4

Pracownicy

Nazwisko	Etat	Szef	Data zatrud.	Płaca pod.	Płaca dod.	Zarobki razem
BINOS	STAZYSTA	SKRABEK	2093.10.15	250.00	170.60	420.60
FARMON	ASYSTENT	MIDAS	2092.09.01	480.00	90.00	570.00
MIDAS	PROFESOR	DRABEK	2077.09.01	1,070.00		1,070.00
TELMAR	STAZYSTA	MIZEK	2094.07.15	208.00		208.00

Sumaryczne zarobki w zespole: 2,268.60

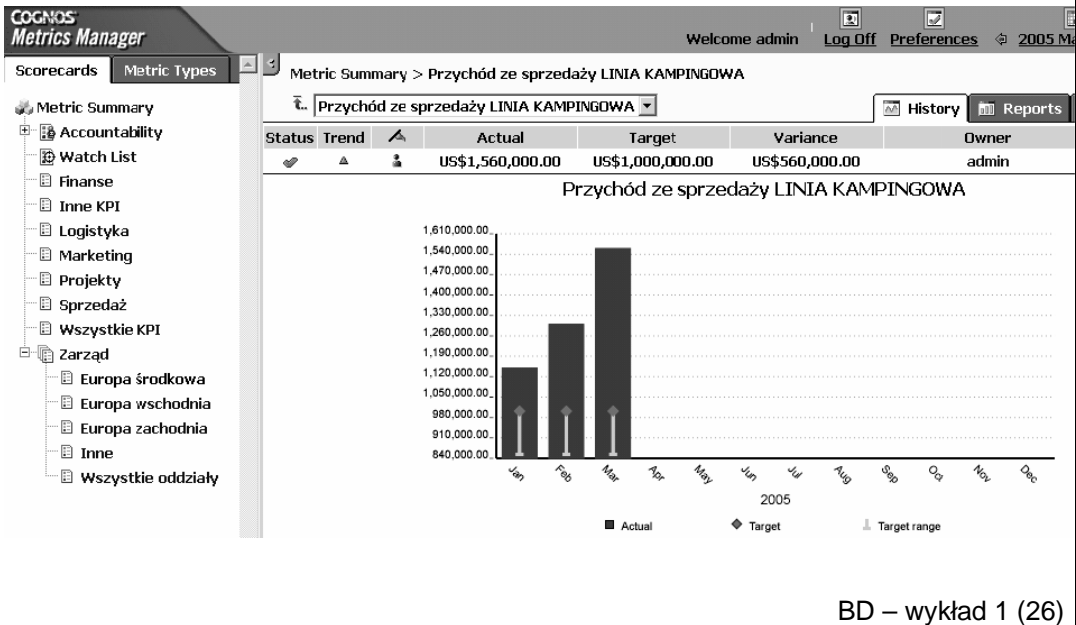
Nazwisko pracownika

Record: 4/4

BD – wykład 1 (25)

Przykład prostego formularza przestawiono na slajdzie.

Raport - przykład



BD – wykład 1 (26)

Przykład raportu przestawiono na slajdzie.



Technologie implementacyjne aplikacji

- Języki 3GL
 - np. C, C++, Visual Basic, Visual C++
 - biblioteki umożliwiające zagnieżdżanie poleceń SQL w kodzie
- Języki 4GL
 - np. SAS 4GL, Oracle Forms
 - umożliwiają bezpośrednio umieszczanie poleceń SQL w kodzie aplikacji i bezpośrednią obsługę wyników poleceń SQL
- Java, PHP, Perl
 - stosowane w aplikacjach web'owych pracujących w architekturze 3-warstwowej

BD – wykład 1 (27)

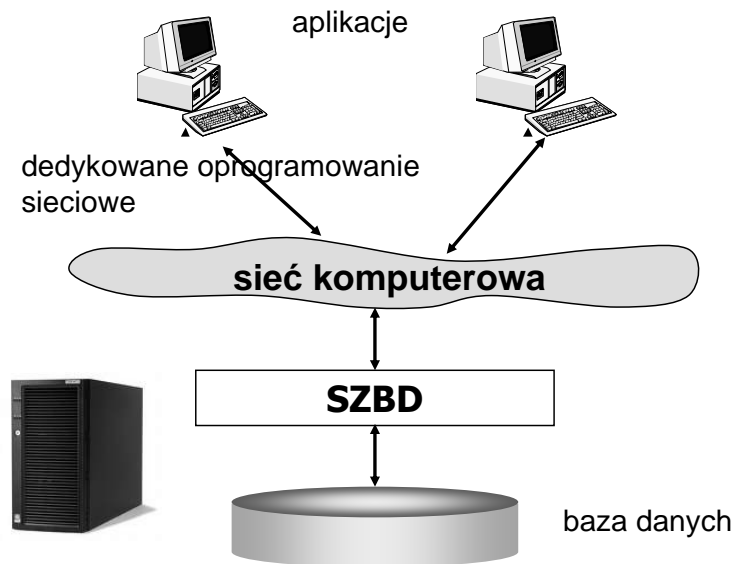
Aplikacje baz danych można implementować w językach trzeciej generacji (3GL) takich jak np. C, C++, Visual Basic, Visual C++. Komunikacja z bazą danych i wykonywanie poleceń SQL i odbiór ich wyników z programów napisanych w tych językach wymaga stosowania specjalizowanych bibliotek, a kod który powstaje jest kodem niskiego poziomu. Z tego względu, w praktyce najczęściej stosuje się albo języki czwartej generacji (4GL) albo języki programowania aplikacji web'owych.

Języki czwartej generacji takie jak np. SAS 4GL lub Oracle Forms, umożliwiają bezpośrednio umieszczanie poleceń SQL w kodzie aplikacji i bezpośrednią obsługę wyników poleceń SQL.

Języki programowania aplikacji web'owych takie jak np. Java, PHP, Perl są stosowane w aplikacjach baz danych pracujących w architekturze 3-warstwowej (omówionej dalej).



Architektura komunikacyjna - klient-serwer



BD – wykład 1 (28)

Obecnie, w praktyce stosuje się dwie podstawowe architektury komunikacyjne z systemem bazy danych, tj. architekturę klient-serwer i architekturę 3-warstwową. W pierwszej z nich, aplikacje użytkowe są zainstalowane na stacjach roboczych i komunikują się z SZBD z wykorzystaniem oprogramowania sieciowego dedykowanego do komunikacji z SZBD. Baza danych znajduje się na dedykowanym serwerze.



Architektura komunikacyjna - 3-warstwowa



BD – wykład 1 (29)

W architekturze 3-warstwowej, pomiędzy stacjami użytkowników, a serwerem bazy danych znajduje się tzw. serwer aplikacji. Jego zadaniem jest udostępnianie umieszczonych na nim aplikacji. Jest to typowa architektura dla aplikacji web'owych. Użytkownik na swojej stacji roboczej posiada tylko przeglądarkę stron www. Aplikacje są udostępniane przez serwer aplikacji w postaci czystych stron html lub w postaci applet'ów Java. W odpowiedzi na polecenia użytkowników realizowane w aplikacjach, serwer aplikacji wysyła odpowiednie żądania do SZBD. SZBD wykonuje polecenia i ich wyniki przesyła do serwera aplikacji, który z kolei przesyła je do aplikacji użytkowników.



Podział systemów baz danych (1)

- Kryteria podziału
 - wykorzystywany model danych
 - liczba węzłów / baz danych
 - cel stosowania
- Model danych
 - relacyjny
 - obiektowy
 - obiektowo-relacyjny
 - semistrukturalny (XML)
 - hierarchiczny
 - sieciowy
- Liczba węzłów / baz danych
 - bazy scentralizowane
 - bazy rozproszone

Podziału systemów BD można dokonać w oparciu o kilka kryteriów. Najważniejszymi są: wykorzystywany model danych, liczba węzłów, czyli liczba baz danych wchodzących w skład systemu i cel stosowania systemu bd.

Ze względu na model danych SBD dzieli się na: relacyjne, obiektowe, obiektowo-relacyjne, semistrukturalne, hierarchiczne, sieciowe.

Ze względu na liczbę wykorzystywanych BD, wyróżnia się systemy scentralizowane z jedną bazą danych i systemy rozproszone z więcej niż jedną bazą wchodzącą w skład systemu.



Podział systemów baz danych (2)

- Cel stosowania
 - przetwarzanie transakcyjne (On-Line Transaction Processing - OLTP)
 - wszelkiego rodzaju systemy ewidencyjne
 - przetwarzanie analityczne (On-Line Analytical Processing - OLAP)
 - hurtownie danych
 - wspomaganie projektowania (Computer Aided Design - CAD)
 - konstrukcje, budynki, urządzenia

Ze względu na cel stosowania wyróżnia się bazy danych przetwarzania transakcyjnego, BD przetwarzania analitycznego, wspomaganie projektowania, informacji geograficznej, wytwarzania oprogramowania.

Bazy danych przetwarzania transakcyjnego (OLTP) stosuje się w typowych zastosowaniach ewidencyjnych, np. w rezerwacji i sprzedaży biletów, w bibliotekach i wypożyczalniach, w systemach ewidencji ludności, pojazdów, mienia nieruchomości, w bankowości w obsłudze bieżącej, w systemach handlu internetowego i bankowości elektronicznej. Zastosowania tego typu charakteryzują się ogromną liczbą jednocześnie działających użytkowników (tysiące, dziesiątki tysięcy). Interakcja pojedynczego użytkownika z bazą danych jest krótka - kilka kilkanaście sekund.

Bazy danych przetwarzania analitycznego (OLAP) stosuje się w systemach wspomaganie zarządzania. Zastosowania tego typu charakteryzują się niewielką liczbą użytkowników (kilku, kilkunastu) ale czas interakcji użytkownika z bazą danych jest długi (godziny, dziesiątki godzin).

Bazy danych dla wspomaganie projektowania umożliwiają przechowywanie projektów złożonych obiektów, np. konstrukcji mostów, budynków, schematy urządzeń.



Podział systemów baz danych (3)

- Cel stosowania cd.
 - systemy informacji geograficznej (Geographical Information Systems - GIS)
 - wytwarzanie oprogramowania (Computer Aided Software Engineering - CASE)

W bazach danych informacji geograficznej przechowuje się zarówno dane tekstowe (np. dane triangulacyjne, opisy terenu) jak i dane przestrzenne (mapy). Tego typu systemy wymagają zaawansowanych technik przeszukiwania map i operacji na nich.

Bazy danych służące do wspomagania wytwarzania oprogramowania przechowują wyniki poszczególnych faz realizacji projektów. Wyniki te są najczęściej reprezentowane w postaci specjalizowanych modeli (diagramów), obiektów i ich własności, projektów i kodów oprogramowania. Tego typu systemy, oprócz standardowej funkcjonalności, wspierają wyszukiwanie zależności pomiędzy obiektami oraz wywodzenie wersji obiektów (np. oprogramowania) i zarządzanie tymi wersjami.



Dostępne SZBD (1)

- Komercyjne
 - Oracle
 - wersja 9i, 10g
 - IBM
 - DB2 UDB
 - Informix(R) Dynamic Server
 - Microsoft
 - SQL Server2000, SQL Server2005
 - Sybase
 - Adaptive Server Enterprise, Adaptive Server Anywhere

Na rynku istnieje wiele komercyjnych systemów BD. Do najpopularniejszych producentów zalicza się Oracle, IBM, Microsoft i Sybase. Oracle oferuje SZBD o nazwie Oracle9i, Oracle10g. IBM oferuje systemy DB2 i Informix(R) Dynamic Server. Microsoft oferuje popularny SQL Server w wersjach 2000 i 2005. Sybase jest producentem systemu Adaptive Server Enterprise i Adaptive Server Anywhere.



Dostępne SZBD (2)

- Niekomercyjne
 - MySQL
 - PostgreSQL
 - FireBird

Ponadto, dostępne są rozwiązania niekomercyjne, spośród których najpopularniejszymi są MySQL, PostgreSQL i FireBird.