1 Definicje

W potocznym rozumieniu termin baza danych [1] odnosi się do zbioru danych zorganizowanych przez system zarządzania bazą danych, który nazywa się również skrótowo systemem DBMS (ang. database management system) lub po prostu systemem baz danych.

Baza danych [3] - dostępny dla wielu użytkowników zbiór powiązanych logicznie danych wraz z definicją ich struktury, zaprojektowany dla zaspokojenia potrzeb przetwarzania danych przez instytucje.

Oczekiwania wobec systemu DBMS [1]:

- Umożliwienie użytkownikowi utworzenie nowej bazy danych i określenie jej schematu (logicznej struktury danych) za pomocą specjalnego języka definiowania danych (ang. data definition language).
- Udostępnienie użytkownikowi możliwości tworzenia zapytań (ang. query) o dane oraz aktualizowanie danych za pomocą odpowiedniego języka nazywanego językiem zapytań (ang. query language) lub językiem manipulowania danymi (ang. data manipulation language).
- Zapewnienie możliwości przechowywania ogromnych ilości danych przez długi czas, chroniąc je przed przypadkowym, lub niepowołanym dostępem, a także zapewniając efektywny dostęp do danych z poziomu języka zapytań i operacji na danych.
- Zapewnienie trwałości, odzyskiwania bazy danych w przypadku awarii, różnego rodzaju błędów lub niewłaściwego użytkowania.
- Sterowanie jednoczesnym dostępem do danych przez wielu użytkowników, z zapewnieniem bezkolizyjności oraz ochrony danych przed przypadkowym uszkodzeniem.

Zalety SZBD [3]:

- kontrola redundancji danych,
- spójność danych,
- więcej informacji w oparciu o te same dane,
- wspólny dostęp do danych,
- poprawa integralności danych,
- poprawa bezpieczeństwa,
- przestrzeganie standardów,
- oszczędność,
- godzenie sprzecznych wymagań,
- poprawa zakresu i czasu dostępu do danych,
- wzrost wydajności,
- łatwość utrzymania wynikająca z niezależności danych,
- zwiększone możliwości wielodostępu,
- rozszerzona obsługa składowania i odtwarzania bazy danych po awarii.

Wady SZBD [3]:

- złożoność,
- rozmiar,
- koszt SZBD,
- dodatkowe koszty sprzętu,
- koszty przeniesienia,
- większy zasięg awarii.

2 Modele danych

Model danych [3] - To spójny zestaw pojęć służący do opisywania danych i związków między nimi oraz do manipulowania danymi ich związkami, a także do wyrażania więzów nałożonych na dane w pewnej instytucji.

W modelu danych [3] można wydzielić trzy następujące składniki:

- część strukturalną zawierającą zbiór zasad, zgodnie z którymi powinny być konstruowane bazy danych;
- cześć wykonawczą opisującą typy dopuszczalnych operacji na danych;
- część opcjonalną zawierającą zbiór zasad integralności, które gwarantują spójność danych.

Kategorie modeli danych [3]:

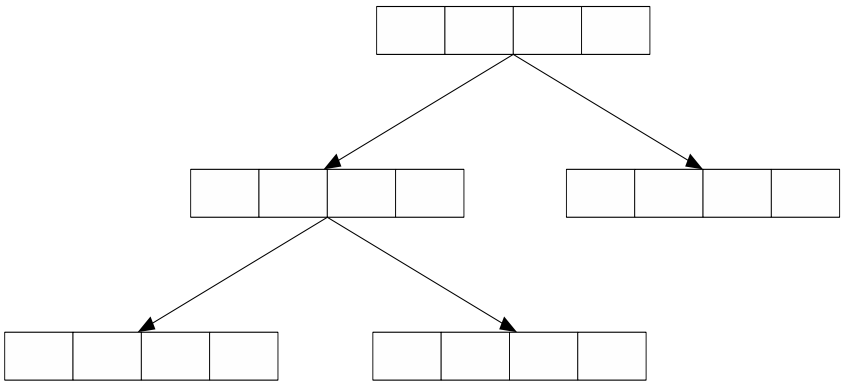
- bazujące na obiektach,
- rekordowe,
- fizyczne.

W rekordowym modelu danych [3] baza składa się z pewnej liczby rekordów o ustalonych formatach, które mogą być różnych typów. Każdy z typów rekordów definiuje ustaloną liczbę pól o zazwyczaj stałych długościach.

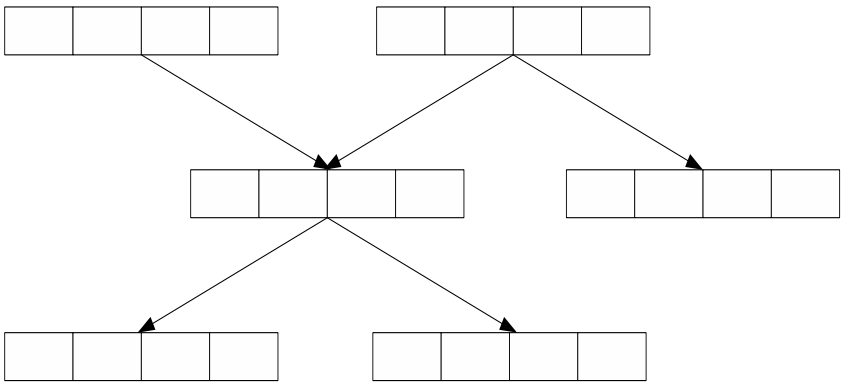
Rekordowe modele danych [3]:

- sieciowy model danych,
- hierarchiczny model danych,
- relacyjny model danych.

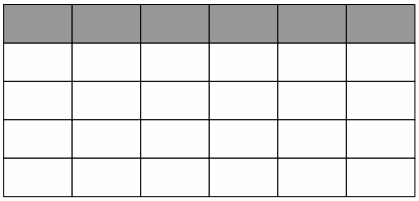
W hierarchiczny modelu danych [3] dane reprezentowane są jako kolekcje rekordów, a związki jako zbiory. W tym modelu węzeł może jednak posiadać tylko jednego przodka. Reprezentujący go graf jest drzewem, w którym rekordy są węzłami (nazywanymi odcinkami), a zbiory krawędziami.



W sieciowym modelu danych [3] dane są reprezentowane jako kolekcje rekordów, a związki pomiędzy nimi są przedstawiane w postaci zbiorów. Rekordy są zorganizowane w postaci uogólnionych struktur grafowych, w których rekordy występują jako węzły, a zbiory jako krawędzie grafu.



Relacyjny model danych [3] bazuje na matematycznym pojęciu relacji. W tym modelu dane i związki są reprezentowane w postaci tabel, z których każda zawiera pewną liczbę kolumn o unikalnych identyfikatorach.



Model relacyjny jest oparty o pojęcie relacji, zaproponowany przez E.F. Codd'a z firmy IBM w 1970 roku. Język SQL jest standardem umożliwiającym definiowanie danych i operacji na nich. W 1985 roku Codd'a zaproponował zbiór zasad dla relacyjnych baz danych [3].

Zasada 0 - Zasada fundamentalna
"Każdy system reklamowany jako relacyjny SZBD bądź uznawany za taki powienien posiadać możliwość zarządzania bazą danych wyłącznie w oparciu o relacyjne cechy danych."

Zasada 12 - Zasada nienaruszalności
"Jeżeli system udostępnia niskopoziomy język umożliwiwiający manipulowanie danymi (działania na pojedynczych rekordach), język ten nie powinien naruszania bądź omijania reguł integralności i więzów wyrażonych w wysokopoziomowym języki relacyjnym systemu (definiującym zależności między wieloma rekordami)."

Zasada 1 - Reprezentacja informacji
"Wszystkie informacje są w relacyjnej bazie danych jawnie reprezentowane na poziomie logicznym tylko w jeden sposób - jako wartości w tabelach."

Zasada 6 - Aktualizacja poprzez perspektywy
"Wszystkie teoretycznie modyfikowalne perspektywy powinny być również modyfikowalne w systemie."

Zasada 3 - Jednorodne traktowanie wartości pustych
"System umożliwia niezależną od typu danych reprezntację wartości pustych (...). Wartości te oznaczają brak odpowiedniej informacji bądź fakt, że określona kolumna nie dotyczy danego elementu."

Zasada 10 - Niezależność więzów integralności
"Węzły integralności dotyczące konkretnej bazy danych powinny definiowalne w relacyjnym podjęzyku danych oraz reprezentowane w katalogu systemu, a nie w programach aplikacji."

Zasada 2 Gwarancja dostępu (do danych)
"Do każdej informacji (wartości atomowej) w relacyjnej bazie danych można uzyskać logiczny dostęp poprzez wskazanie nazwy tabeli, wartości klucza głównego i nazwy kolumny."

Zasada 4 - Katalog systemu jest dostępny bezpośrednio i ma strukturę opartą na modelu relacyjnym - "Opis bazy danych jest na poziomie logicznym reprezentowany tak samo jak 'normalne' dane w bazie,

dzięki czemu uprawnieniu użytkownicy mogą uzyskać dostęp do tych informacji (i modyfikować je) za pomocą tego samego języka relacyjnego, który jest stosowany w przypadku standardowych danych w bazie."

Zasada 5 - Wszechstonny podjęzyk danych
"System relacyjny może wspierać wiele języków oraz różnorodne tryby dostępu do danych (na przykład tryb pozostawiania nieustalonych wartości). Jednak co najmniej jeden z języków dostępnych w systemie powinine umożlwiaić realizację następujących funkcji: (1) definiowanie danych; (2) definiowanie perspektyw; (3) manipulacja danymi (interaktywne i za pomocą programu); (4) definiowanie więzów integralności; (5) autoryzacja dostępu; (6) obsługa transakcji (...)."

Zasada 7 - Wysokopoziomowy dostęp do operacji wstawiania, modyfikacji i usuwania danych
"Możliwość operowania na relacjach bazowych i relacjach pochodnych (czyli na perspektywach) za pomoca pojedynczych poleceń dotyczyć powinna nie tylko wyszukiwania, lecz również wstawiania, modyfikacji i usuwania danych."

Zasada 8 - Fizyczna niezależność danych
"Zmiany fizycznej reprezntacji danych bądź metod dostępu do nich nie wspływają na funkcjonalność programów aplikacji i operacji dostępnych z poziomu terminala."

Zasada 9 - Logiczna niezależność danych
"Zmiany tabel bazowych zachowujące umieszczone w nich informacje nie wpływają na funkcjonalność programów aplikacji i operacji dostępnych z poziomu terminala (o ile istnieje teoretyczna możliwość, że zmiany te nie zaburzą funkcjonlaności operacji na danych).

Zasada 11 - Niezależność od rozmieszczenia (rozproszenia) danych
"Podjęzyk manipulacji danymi relacyjnego SZBD powinien zapewniać, że program aplikacji i operacji dostępu do danych mogą pozostawać niezmienione niezależnie od tego, czy dane są fizycznie scentralizowane, czy rozproszone."

Modele danych bazujące na obiektach [3]:

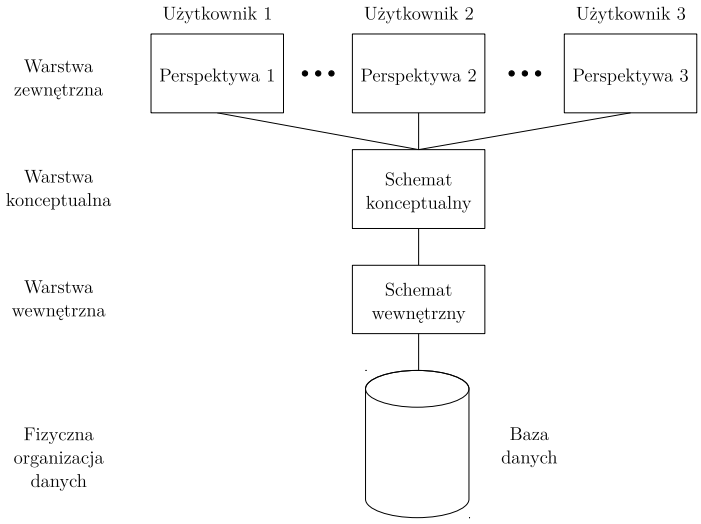
- model związków encji ER (ang. Entity-Relationship),
- model semantyczny,
- model funkcjonalny,
- model obiektowy.

Model obiektowy [3] (ang. object model, object-oriented model) - model danych, którego podstawą są pojęcia obiektu, klasy, dziedziczenia, polimorfizmu oraz innych cech obiektowości. Dość często model obiektowy jest rozumiany jako wariant modelu encja-związek, gdzie istotnym uzupełnieniem jest przypisanie metod do encji (klas obiektów).

Fizyczne modele danych [3] przedstawiają sposób przechowywania danych w komputerze. Dostarczają one takich informacji, jak: struktura rekordów, uporządkowanie rekordów i ścieżki dostępu.

3 Architektury DBMS

3.1 Trójwarstwowa architektura ANSI-SPARC

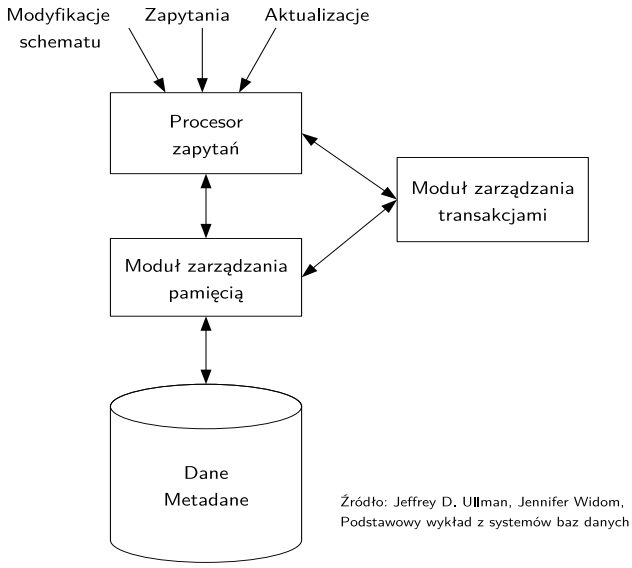


Źródło: T. Connolly, C. Begg, Systemy baz danych, RM, 2004

Trójwarstwowa architektura ANSI-SPARC

Trójwarstwowa architektura ANSI-SPARC składa się z następujących warstw [3]: zewnętrznej, wewnętrznej i konceptualnej. Warstwa zewnętrzna przedstawia sposób w jaki baza danych jest widziana przez użytkowników. Natomiast warstwa wewnętrzna zawiera opis sposobu widzenia tych danych przez system operacyjny oraz SZBD. Warstwa konceptualna odpowiada za odwzorowanie pomiędzy tym warstwami oraz zapewnia ich niezależność.

3.2 Główne elementy systemu DBMS



Architektura DBMS

Moduł zarządzania transakcjami [1] odpowiada za spójność systemu. Musi on gwarantować, że kilka jednocześnie przetwarzanych zapytań nie będzie sobie wzajemnie przeszkadzać oraz że żadne dane nie zostaną utracone, nawet jeśli nastąpi awaria systemu.

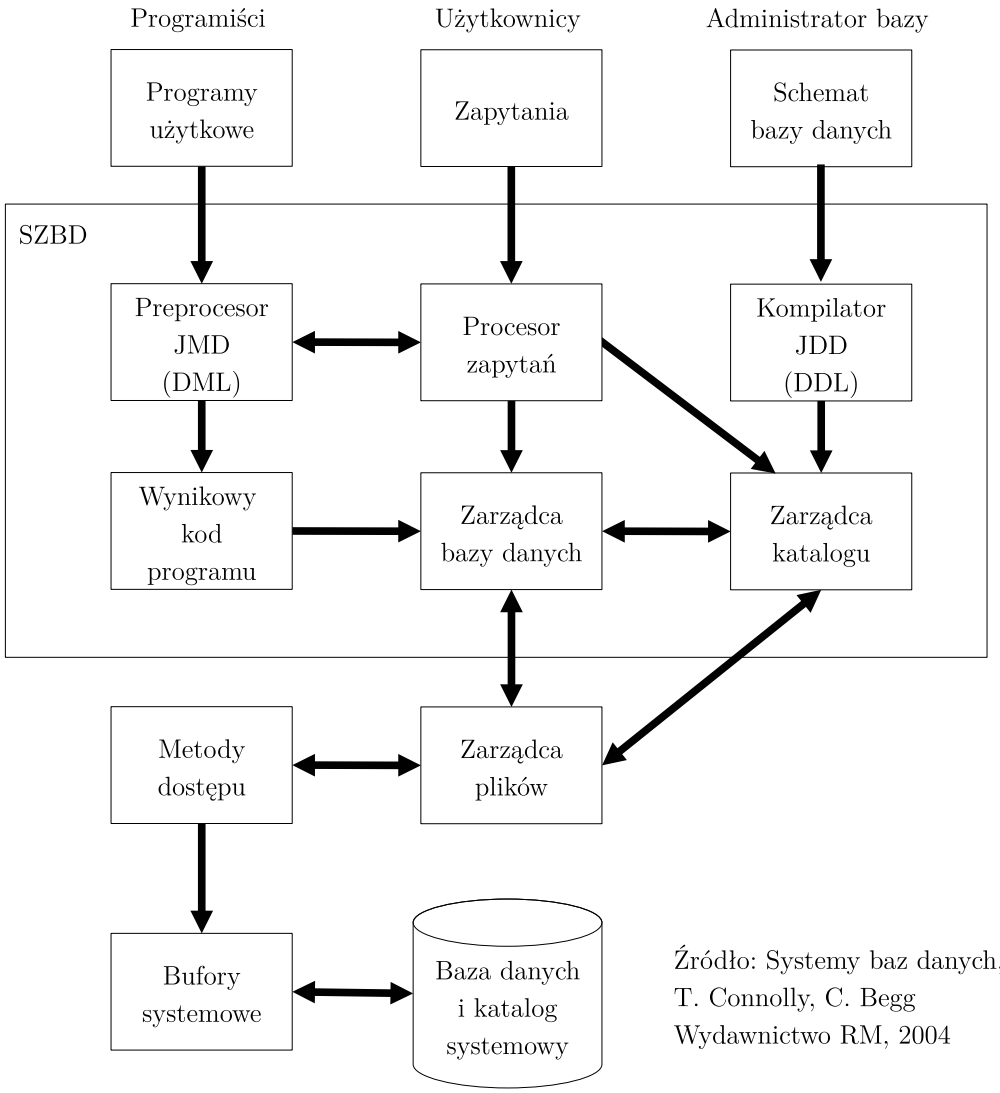
Transakcje ACID:

– niepodzielność (ang. atomicity) – wszystkie części transakcji powinny zostać przeprowadzone albo żadna,
– spójność (ang. consistency) – przed i po zatwierdzeniu transakcji wszystkie dane w bazie są spójne, spójność oznacza, że dane nie naruszają ograniczeń nałożonych na bazę danych,
– izolacja (ang. isolation) – gdy dwie transakcje są przetwarzane jednocześnie, ich działania nie mogą na siebie wzajemnie wpływać,
– trwałość (ang. durability) – jeśli transakcja się zakończy, to jej wynik nie może zostać utracony z powodu awarii systemu.

Moduł zarządzania pamięcią [1] składa się z dwóch części:

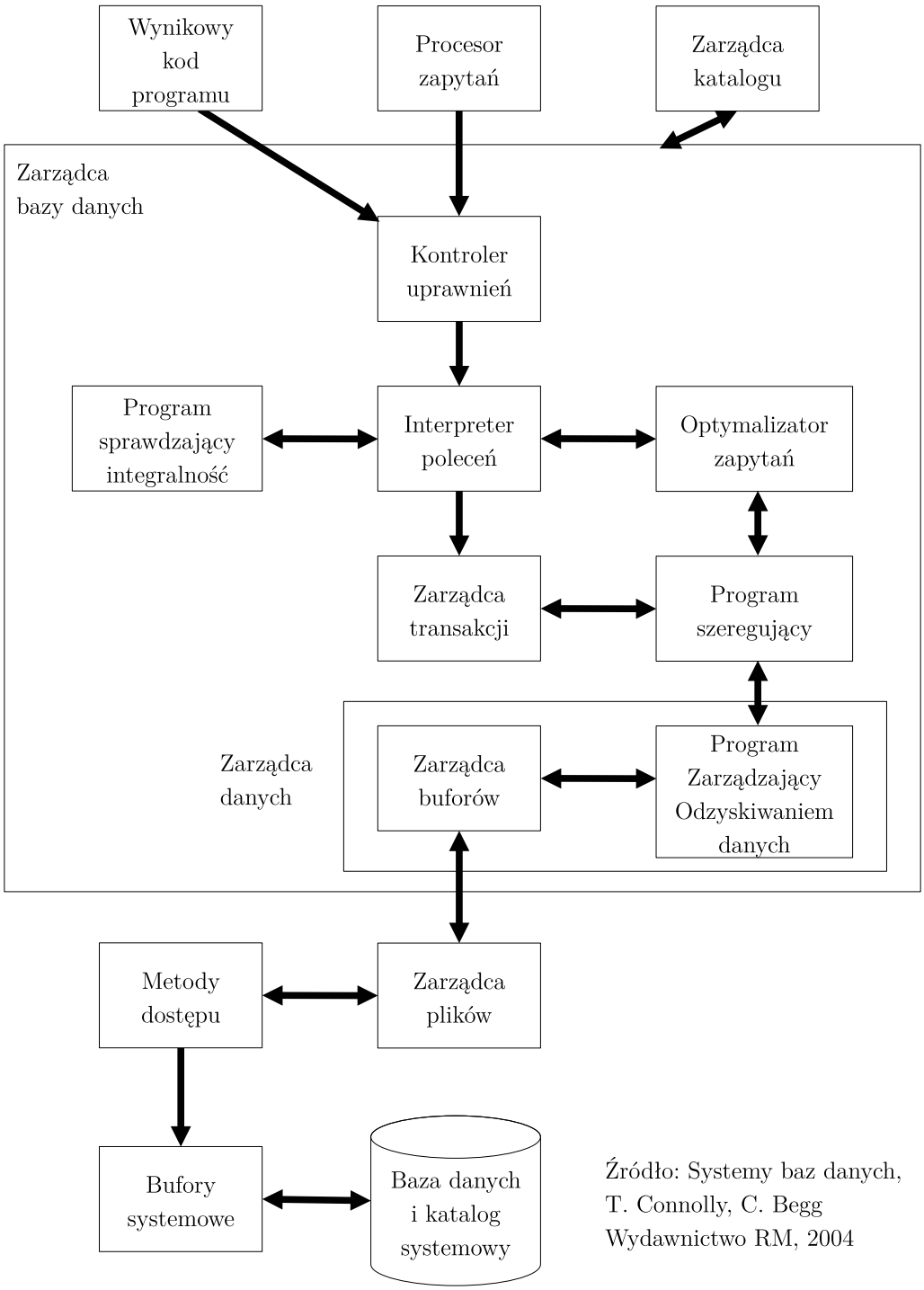
1. Moduł zarządzania plikami przechowuje dane o miejscu zapisania plików na dysku i na polecenie modułu zarządzania buforami przesyłam zawartość bloku lub bloków, gdzie jest zapamiętany żądany plik.
2. Moduł zarządzania buforami obsługuje pamięć operacyjną. Moduł zarządzania plikami przekazuje bloki danych z dysku, a moduł zarządzania buforami wybiera w pamięci operacyjnej strony, które zostaną przydzielone dla wybranych bloków.

Zadaniem moduł zarządzania zapytaniami [1] jest przekształcenie zapytania lub operacji na bazie danych, wyrażanych zazwyczaj w języku bardzo wysokiego poziomu (np. SQL), w ciąg poleceń żądających dostarczenie odkreślonych danych.



Źródło: Systemy baz danych,
T. Connolly, C. Begg
Wydawnictwo RM, 2004

- "Procesor zapytań (ang. query processor) Jest to główny składniki SZBD. którego zadaniem jest tłumaczenie zapytań na ciąg niskopoziomowych instrukcji przesyłanych do zarządcy bazy danych."
- "Zarządca bazy danych (DM od ang. Database Manager) DM współpracuje z dostarczonymi przez użytkownika programami oraz z zapytaniami. DM przyjmuje zapytania i bada schematy zewnętrzne i schemat konceptualny, dążąc do wyznaczenia tych rekordów konceptualnych, które są niezbędne do udzielenia odpowiedzi. Następnie DM przekazuje wywołanie do zarządcy pliku."
- "Zarządca plików (ang. file manager) Zarządca plików operuje na plikach przechowujących dane oraz zajmuje się przydziałem pamięci na dyskach. Tworzy i przetwarza listę struktur i indeksów zdefiniowaną w schemacie wewnętrznym. (...) Nie realizuje on jednak bezpośrednio fizycznego odczytu i zapisu danych, lecz raczej przekazuje te żądania do odpowiedniej metody dostępu, która odczytuje dane z bufora systemowego (lub pamięci podręcznej) lub je tam zapisuje."
- "Preprocesor JMD Ten moduł przekształca instrukcje języka manipulowania danymi (JMD), zawarte w programach użytkowych, na wywołania funkcji w macierzystym języku."
- "Kompilator JDD Kompilator języka definicji danych (JDD) przekształca instrukcje tego języka na zestaw tabel zawierających metadane. Te tabele są następnie zapamiętywane w katalogu systemowych, a informacje sterujące są umieszczane w nagłówkach plików z danymi."
- Zarządca katalogu (ang. catalog manager) Zarządca katalogu zarządza dostępem i administruje katalogiem systemowym."



Źródło: Systemy baz danych,
T. Connolly, C. Begg
Wydawnictwo RM, 2004

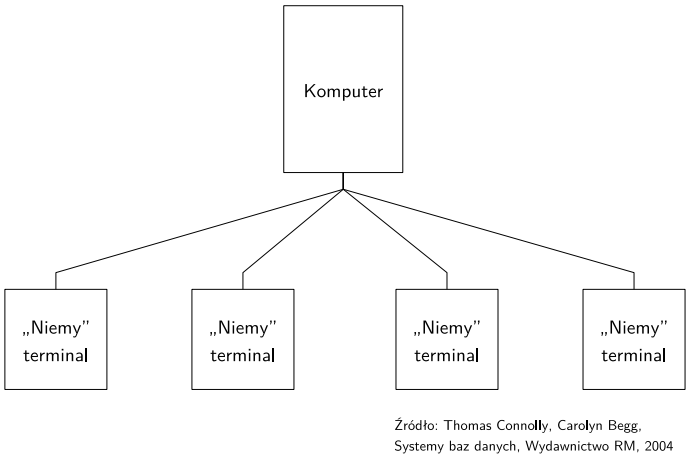
- "Kontroler uprawnień (ang. authorization control) Ten moduł sprawdza, czy użytkownik posiada uprawnienia konieczne do przeprowadzenia żądanej operacji."
- "Interpreter poleceń (ang. command processor) Kiedy system sprawdzi, czy użytkownik posiada uprawnienia pozwalające na przeprowadzenie danej operacji, sterowanie jest przekazywane do interpretera poleceń."
- "Program sprawdzający integralność (ang. integrity checker) Dla każdej operacji modyfikującej bazę danych, program sprawdza, czy żądana operacja spełnia wszystkie wymagane więzy integralności (takie jak np. więzy klucza).

- "Optymalizator zapytań (ang. query optimizer) Ten moduł optymalny sposób wykonania zapytania."
- "Zarządca transakcji (ang. transaction manager) Ten moduł wykonuje niezbędne przetwarzanie operacji pobieranych z transakcji."
- "Program szeregujący (ang. scheduler) Ten moduł jest odpowiedzialny za bezkonfliktowe wykonywanie się współbieżnych operacji na bazie danych. Steruje on względnym porządkiem, w jakim operacje są wykonywane."
- "Program zarządzający odzyskiwaniem danych (ang. recovery manager) Ten moduł zapewnia, że baza danych pozostaje w niesprzecznym stanie nawet po awarii. Jest odpowiedzialny za commit i abort transakcji."
- "Zarządca buforów (ang. buffer manager) Ten moduł jest odpowiedzialny za przesyłanie danych pomiędzy pamięcią operacyjną a pamięciami pomocniczymi, takimi jak dyski czy taśmy."

3.3 Architektury wielodostępnego SZBD:

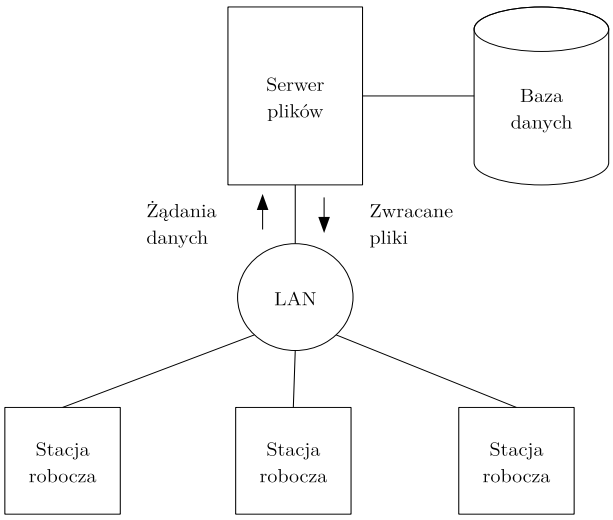
Architektury wielodostępnego SZBD:

- przetwarzanie zdalne,
- serwer plików,
- klient - serwer,
- inne np. trójwarstwowa architektura klient-serwer.



Przetwarzanie zdalne

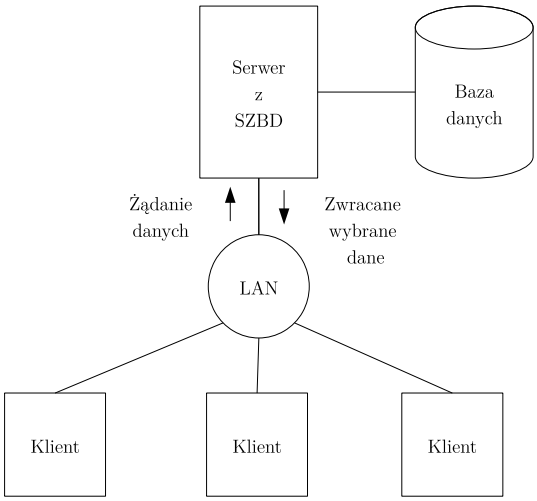
Terminale użytkowników [3] były niezdolne do samodzielnego funkcjonowanie. Służyły do przesyłania komunikatów do aplikacji i z powrotem poprzez podsystem systemu operacyjnego odpowiedzialny za sterowanie komunikacją. Aplikacje korzystały z usług SZBD.



Źródło: Thomas Connolly, Carolyn Begg,
Systemy baz danych, Wydawnictwo RM, 2004

Serwer plików

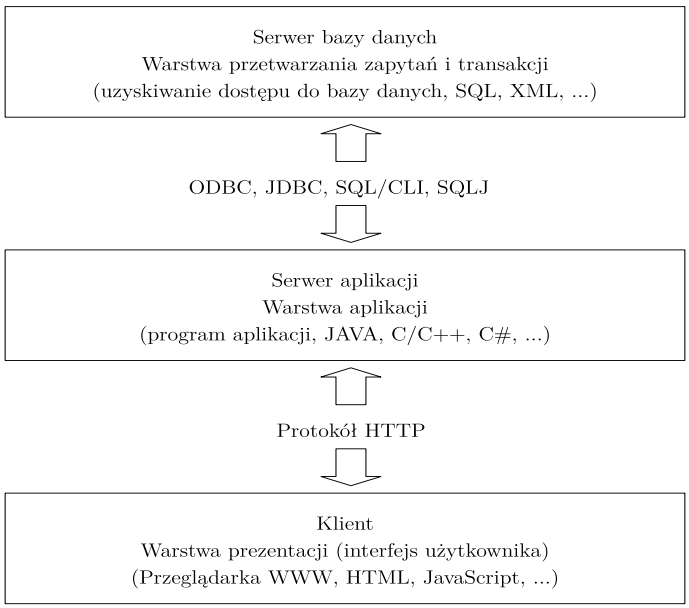
Zadaniem serwera plików [3] jest przechowywanie plików używanych przez aplikacje i SZBD. Same aplikacje i SZBD są wykonywane na każdej stacji roboczej i w razie potrzeby odwołują się do plików na serwerze plików.



Źródło: Thomas Connolly, Carolyn Begg,
Systemy baz danych, Wydawnictwo RM, 2004

Klient - serwer

Klient [3] obsługuje interfejs użytkownika i sterowanie w aplikacji. Klient przyjmuje żądania użytkowników, sprawdza ich poprawność składniową i tworzy żądanie do bazy danych. Następnie przesyła je jako komunikat do serwera, czeka na odpowiedź i formatuje tę odpowiedź zgodnie z wymaganiami użytkownika. Serwer [3] przyjmuje i przetwarza żądanie do bazy danych, a następnie przesyła wyniki z powrotem do klienta.

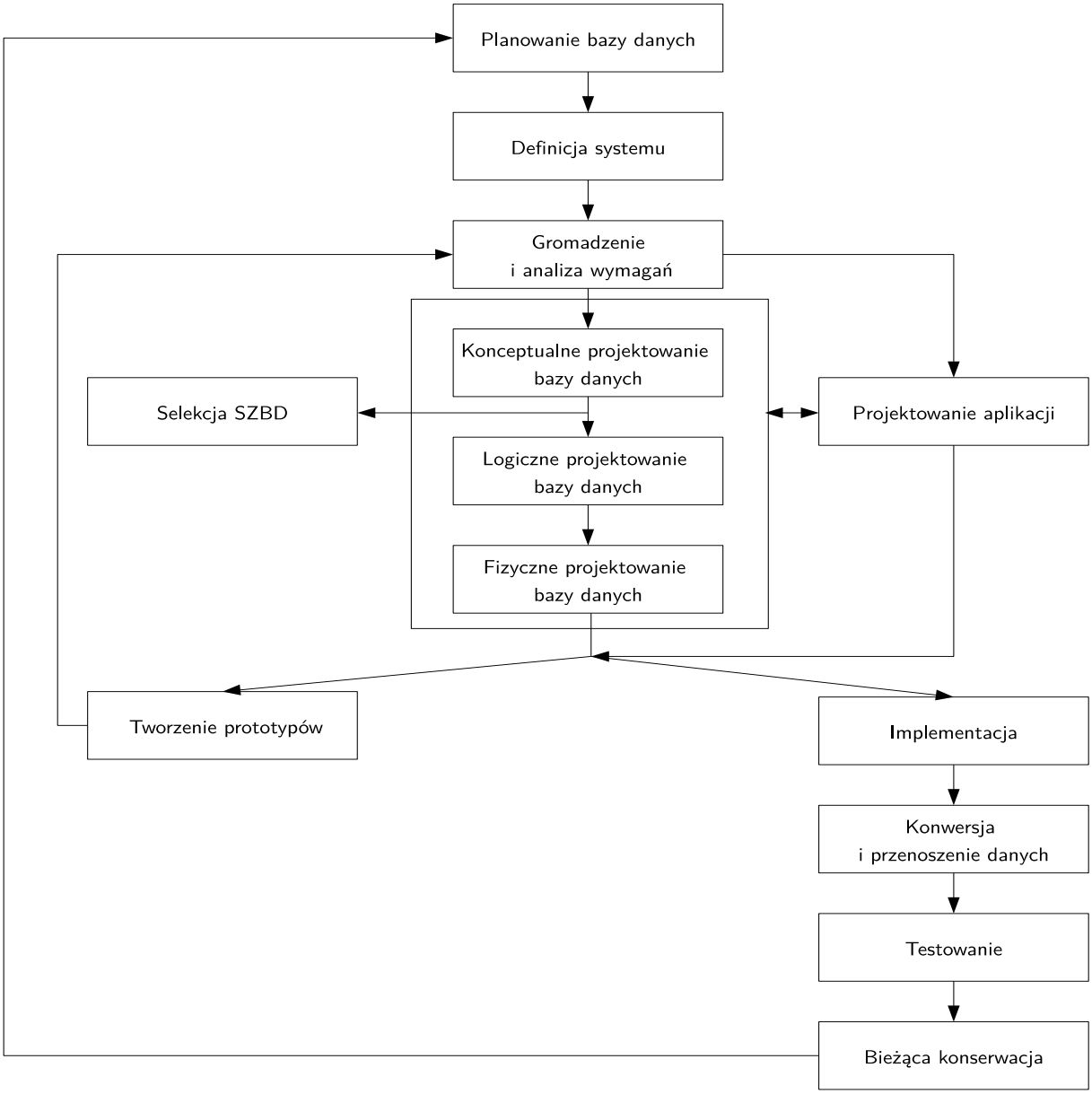


Source: Ramez Elmasri, Shamkant B. Navathe,
Wprowadzeni do systemów baz danych, Helion, 2005

Trójwarstwowa architektura klient-serwer

Warstwa prezentacji (klient) [8] odpowiada za skonstruowanie interfejsu i za interakcję z użytkownikiem. Warstwa aplikacji odpowiada za logikę aplikacji. Serwer baz danych obsługuje zapytania i żądania aktualizacji warstwy danych z warstwy aplikacji, przetwarza zapytania i wysyła ich wyniki.

4 Cykl życia aplikacji bazy danych



Źródło:
Thomas Connolly, Carolyn Begg, Systemy baz danych, Wydawnictwo RM, 2004

Zestawienie głównych czynności związanych z każdą fazą cyklu życia aplikacji bazy danych [3]:

- Planowanie bazy danych - planowanie najbardziej skutecznych metod realizacji faz cyklu życia.
- Definicja systemu - określenie zakresu i granic stosowania danej aplikacji bazy danych.
- Gromadzenie i analiza wymagań - zbieranie i analiza wymagań pochodzących od użytkowników i wynikających z obszarów zastosowań.
- Projektowanie bazy danych - projektowanie konceptualne, logiczne i fizyczne bazy danych.
- Selekcja SZBD - wybór SZBD odpowiedniego dla aplikacji bazy danych.
- Projektowanie aplikacji - projektowanie interfejsów użytkownika i programów użytkowych, które będę przetwarzać bazę danych.

- Tworzenie prototypów (opcjonalne) - budowanie działającego modelu aplikacji bazy danych, który pozwala projektantom i użytkownikom zobrazować i ocenić sposób działania i wygląd końcowego systemu.
- Implementacja - tworzenie zewnętrznych, konceptualnych i wewnętrznych definicji bazy danych i programów użytkowych.
- Konwersja i przenoszenie danych - przenoszenie danych ze starego systemu do nowego.
- Testowanie - testowanie i usuwanie błędów z aplikacji bazy danych oraz sprawdzanie zgodności z wymaganiami użytkowników.
- Bieżąca konserwacja - aplikacja bazy danych jest w pełni zaimplementowana. System jest na bieżąco monitorowany i konserwowany. W razie potrzeby do aplikacji baz danych są wprowadzane nowe wymagania poprzez ponowne przejście przez powyższe fazy.

Metodologia projektowania - Kompleksowe podejście wykorzystujące procedury, techniki, narzędzia oraz metody tworzenia dokumentacji służącej realizacji i uproszczeniu procesu projektowania.

Konceptualne projektowanie bazy danych [3] - proces tworzenia modelu dla informacji wykorzystywanych w przedsiębiorstwie, niezależnego od wszelkich szczegółów reprezentacji fizycznej.

Logiczne projektowanie bazy danych [3] - proces tworzenia modelu dla informacji wykorzystywanych w przedsiębiorstwie, przystosowanego do konkretnego modelu danych, lecz niezależnego od konkretnego SZBD i innych szczegółów implementacji fizycznej.

Fizyczne projektowanie bazy danych [3] - proces tworzenia opisu implementacji bazy danych w pamięci zewnętrznej; opis ten zawiera relacje bazowe, organizację plików, indeksy stosowane dla uzyskania efektywnego dostępu do danych, a także wszystkie więzy integralności i zagadnienia bezpieczeństwa.

Literatura

[1] Jeffrey D. Ullman, Jennifer Widom, "Podstawowy wykład z systemów baz danych", WNT, 2001

[2] Jeffrey D. Ullman, Jennifer Widom, "Systemy baz danych, Pełny wykład", WNT, 2006

[3] Thomas Connolly, Carolyn Begg, "Systemy baz danych", Wydawnictwo RM, 2004

[4] Thomas Connolly, Carolyn Begg, "Database systems", Peason, 2015

[5] Kevin Loney, Bob Bryla, Oracle Database 10g, Podręcznik administratora baz danych. Helion 2008

[6] Scott Urman, Ron Hardman, Michael McLaughlin, "Oracle Database 10g, Programowanie w języku PL/SQL", Helion, 2008

[7] Price J, "Oracle Database 11g i SQL Programowanie", Helion, 2009

[8] Ramez Elmasri, Shamkant B. Navathe, "Wprowadzeni do systemów baz danych", Helion, 2005

[9] Sharon Allen, "Modelowanie danych", Helion, 2006

[10] Jose A. Ramalho, Oracle 8i, Mikom, 2001

[11] Dokumentacja do bazy danych Oracle dostępna na stronie www.oracle.com

[12] Sadalage P. J., Fowler M., "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012

[13] Michael Stonebraker,"New Sql: An Alternative to Nosql and Old Sql For New Oltp Apps", June 16, 2011

[14] Oracle Database PL/SQL Language Reference 11g Release 1 (11.1) B28370-02 (B28370.pdf)

Materiały do przedmiotu dostępne są na stronie: http://weaii-moodle.tu.kielce.pl/