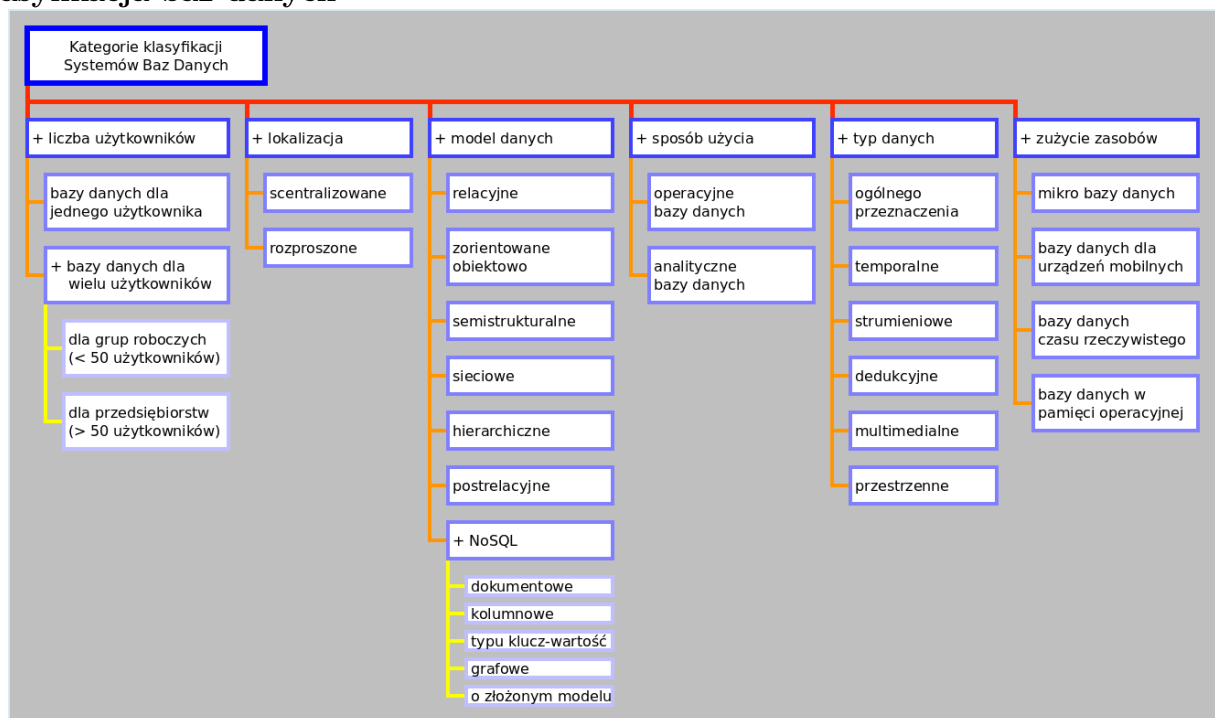


Spis treści

1	Klasyfikacja systemów baz danych ze względu na:	1
1.1	liczbę użytkowników	1
1.2	lokalizację	2
1.3	model danych	5
1.4	sposób użycia	13
1.5	typ danych	14
1.6	zużycie zasobów	17
2	Źródła	19

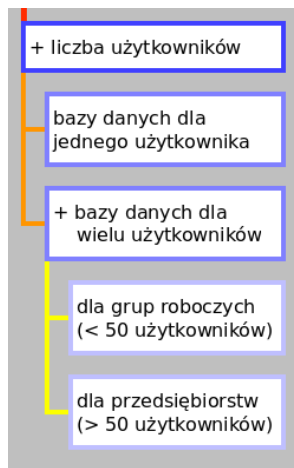
1 Klasyfikacja systemów baz danych ze względu na:

Klasyfikacja baz danych



1.1 liczbę użytkowników

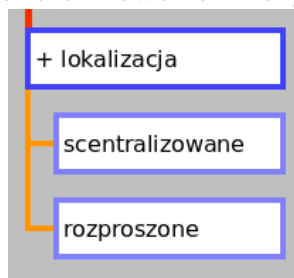
Bazy danych dla jednego lub wielu użytkowników



- Bazy danych dla jednego użytkownika dopuszczają tylko jedno połączenie do bazy danych w tym samym czasie.
- Bazy danych dla wielu użytkowników pozwalają na równoczesny dostęp wielu użytkownikom.

1.2 lokalizację

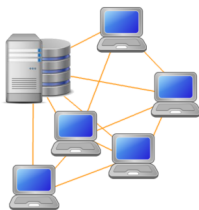
Scentralizowane i rozproszone systemy baz danych



- Scentralizowane bazy danych znajdują się w jednym miejscu, w jednej lokalizacji.
- Rozproszone bazy danych są zlokalizowane w wielu różnych miejscach.

Scentralizowane bazy danych

Scentralizowana baza danych



- baza zlokalizowana, przechowywana i zarządzana w jednej lokalizacji
- jeden komputer, serwer bazodanowy
- najczęściej używany przez organizacje albo instytucje
- dostęp dla użytkowników poprzez sieć

Zalety

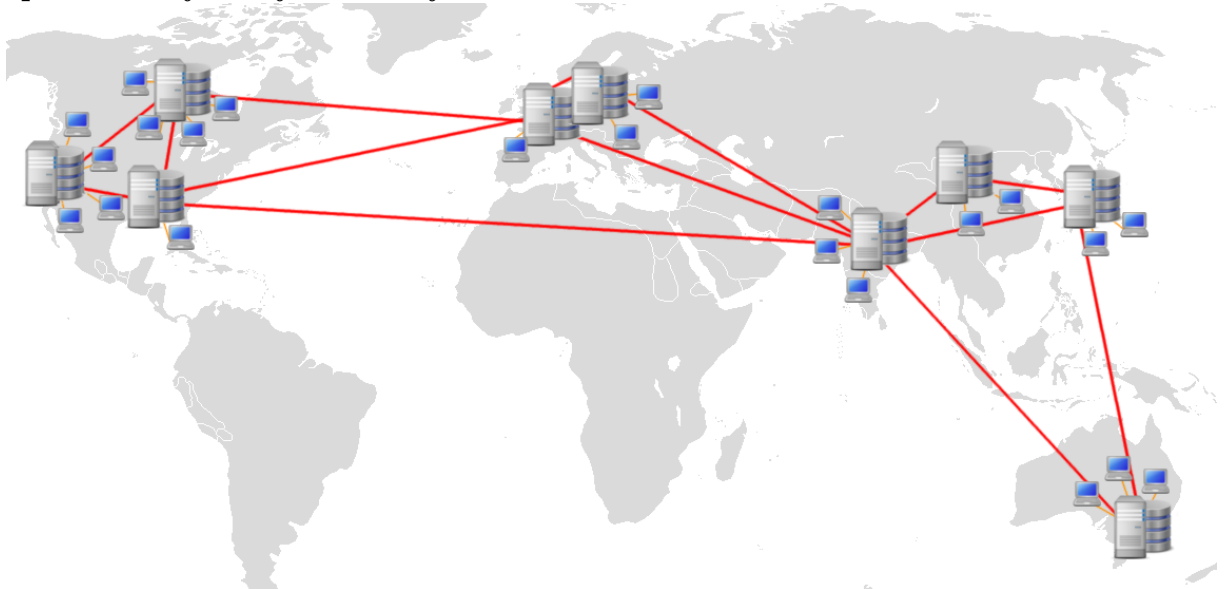
- maksymalna integralność i minimalna redundancja
- jedno miejsce dla wszystkich danych - każdy zbiór danych jest tylko w jednym miejscu
- łatwiejsze zarządzanie danymi, możliwa pełna ich spójność
- większe bezpieczeństwo - dane można wykraść tylko z jednego miejsca
- lepsze zabezpieczenie danych, bo baza w jednym miejscu - łatwiej o konfigurację odporną na uszkodzenia
- łatwiejsza w użyciu dla użytkownika, bo mamy tylko jedną konstrukcję bazy danych
- łatwiej administrować bazą i przenosić dane

- dostęp do danych w tym samym czasie - dane są w jednej lokalizacji
- aktualizacja danych natychmiast widoczna dla użytkownika końcowego
- opłacalna ze względu na koszty - mniej pracowników, zasilania, mniejsze koszty utrzymania bazy

Wady

- w dużym stopniu zależne od połączeń sieciowych,
- duży ruch sieciowy może powodować opóźnienia w dostępie
- jedno miejsce przechowywania danych to ograniczony dostęp - spadek wydajności
- ewentualna awaria, uszkodzenie - wszystkie dane przepadają
- ze względu na normalizację danych, w przypadku utraty danych, ciężko je odzyskać

Rozproszone systemy baz danych



Rozproszone systemy baz danych

Systemy rozproszonych baz danych

to bazy danych, które mogą być składowane na wielu komputerach, zlokalizowanych w tym samym miejscu; albo mogą być rozproszone na wiele komputerów połączonych siecią.

- podział kolekcji danych na wiele lokalizacji fizycznych (serwery internetowe, komputery w sieciach intranetowych czy extranetowych)
- poprawa wydajności poprzez przetwarzanie transakcji na wielu komputerach

Replikacja i powielanie danych

Dwa procesy zapewniające aktualność danych w rozproszonej bazie.

Replikacja

- wymaga specjalnego oprogramowania synchronizującego dane
- prowadzi do ujednoczenia danych w rozproszonej bazie

Powielanie

- to duplikacja bazy, np. z serwera nadrzędnego na podrzędny
- ma mniejszą złożoność
- odbywa się często w ustalonych godzinach - jest harmonogram
- zmiana danych tylko za pośrednictwem komputera nadrzędnego

Architektura

- **Homogeniczne rozproszone DBMS** mają takie samo oprogramowanie i sprzęt, na którym działają wszystkie instancje baz danych i mogą być dostępne poprzez jeden interfejs tak, jakby były jedną bazą danych.
- **Heterogeniczne rozproszone DBMS** mogą działać na różnym sprzęcie, na innych systemach operacyjnych, z wykorzystaniem innych systemów zarządzania bazą danych, a nawet przy różnych modelach danych dla różnych baz.

Zalety

- Zwiększona wydajność.
- Zwiększenie niezawodności i dostępności.
- Łatwiejsza rozbudowa - pozioma rozbudowa (w bok) zakłada tylko dodawanie kolejnych węzłów.
- Możliwość odzwierciedlenia struktury organizacji; wiele oddziałów na świecie - wiele serwerów.
- Lokalna autonomia - oddziały same kontrolują swoje dane.
- Większe bezpieczeństwo - dane w wielu miejscach bardzo trudno wszystkie zniszczyć.
- Awaria jednego serwera nie wpływa na wydajność - ciągłość pracy.
- Koszt - taniej zbudować klaster tańszych serwerów niż kupować bardzo wydajny serwer.
- Niezależność od sprzętu, systemu operacyjnego, sieci, fragmentacji, SZBD, lokalizacji.
- Modułowość - łatwość zmian, dodawania czy usuwania modułów.

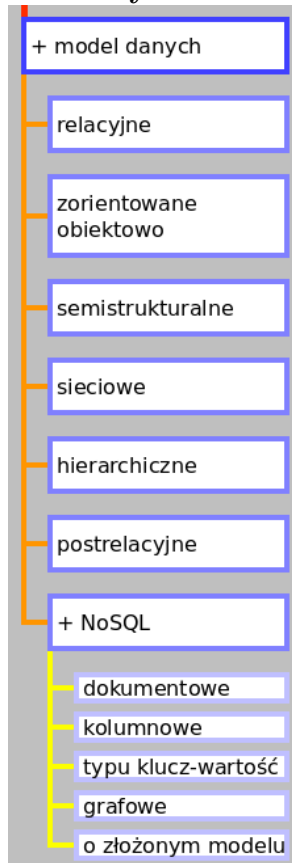
Wady

- Złożoność - trudniej o przezroczystość dla użytkownika końcowego, trudniej taką bazą zarządzać.
- Koszty - wiele serwerów w wielu miejscach - więcej budynków, pracowników, prądu, ...
- Bezpieczeństwo - wiele oddalonych serwerów - każdy trzeba zabezpieczyć, szyfrowanie połączeń między serwerami.
- Trudność w utrzymaniu spójności.

- Braki w doświadczeniu ...
- Brak standardów - nie istnieją jeszcze żadne narzędzia czy metody, które pomogą użytkownikom przekształcać scentralizowany SZBD do rozproszonego SZBD.
- Potrzebne jest dodatkowe oprogramowanie, np. do replikacji.
- Poważnym problemem jest kontrola współbieżności - blokady i punkty czasowe.
- Analiza rozproszonych danych.

1.3 model danych

Model danych



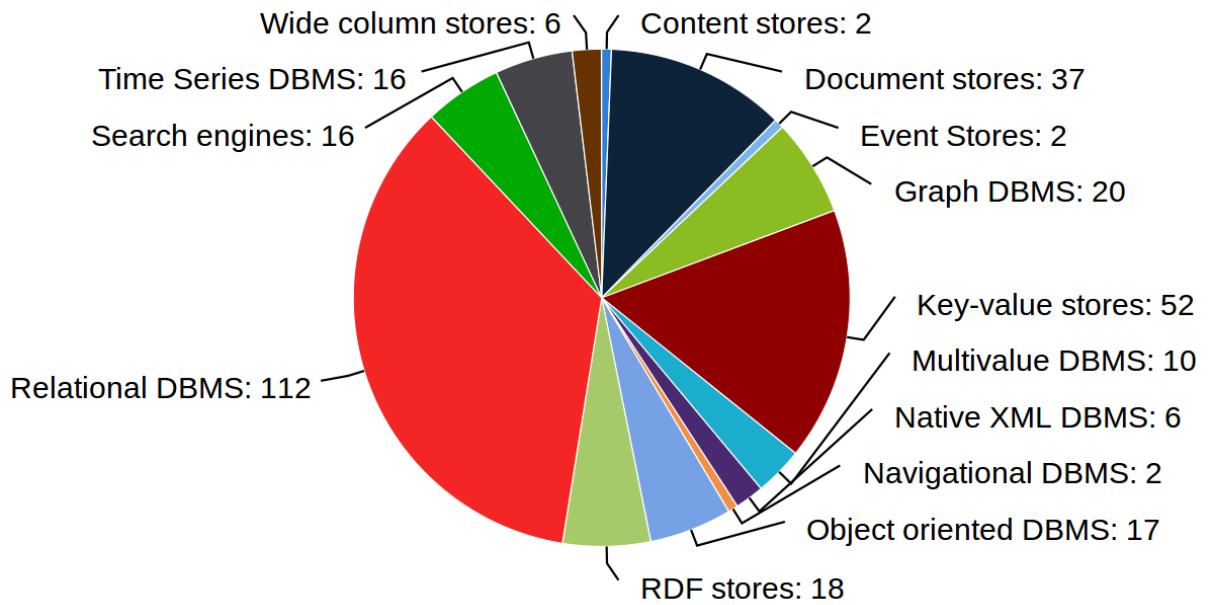
Logiczny model danych

reprezentuje abstrakcyjną strukturę w dziedzinie informacji. Jest niezależny od używanej technologii.

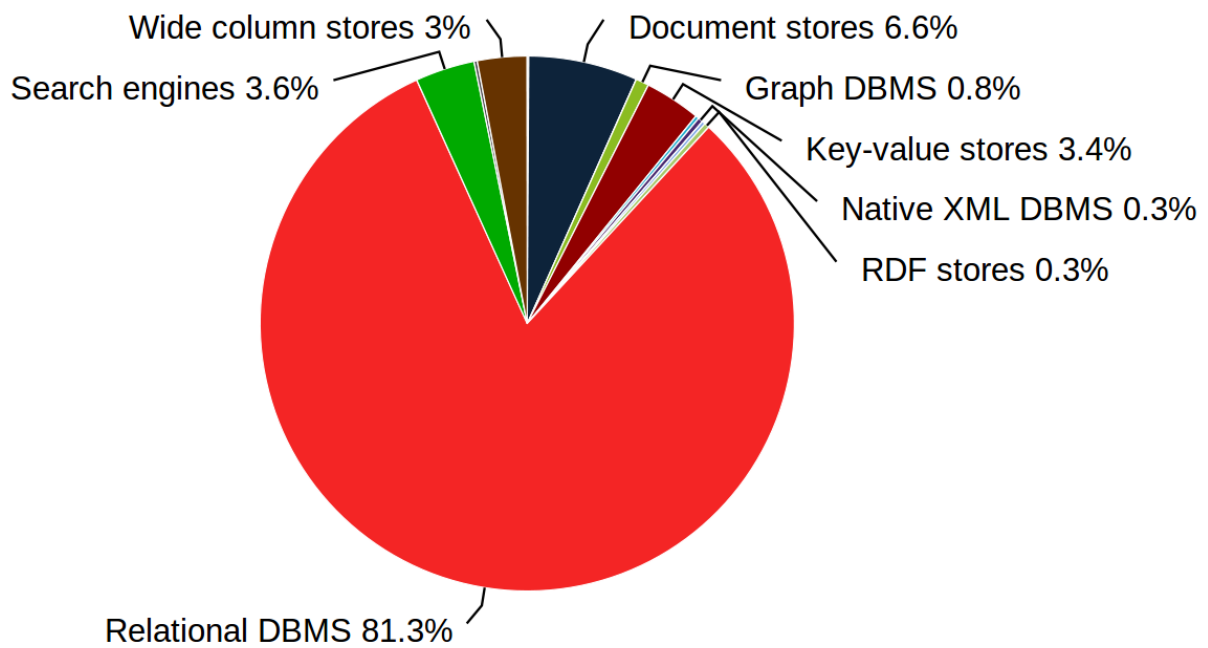
Fizyczny model danych

jest reprezentacją zaprojektowanych danych, który uwzględnia udogodnienia i ograniczenia danego systemu zarządzania bazą danych. W cyklu życia projektu pochodzi on z logicznego modelu danych.

Statystyki dla modeli danych - liczba DBMS

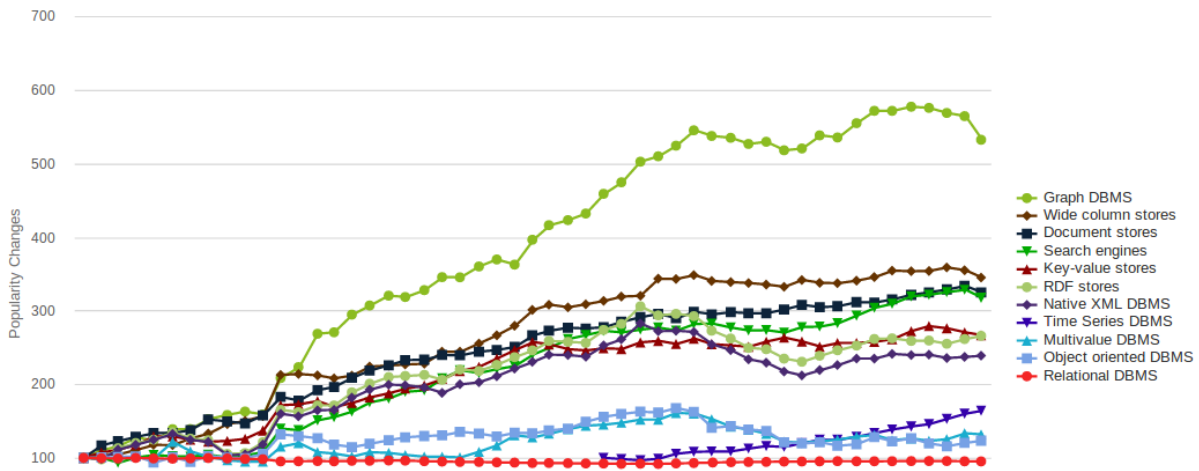


Statystyki dla modeli danych - udział w rynku

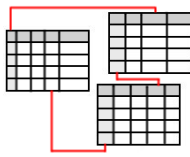


Statystyki dla modeli danych - zmiany popularności

Dane od stycznia 2013 do marca 2017



Relacyjne bazy danych



Relacyjne bazy danych

są oparte na relacyjnym modelu danych zaproponowanym przez E.F. Codd'a w 1970 r. Model ten organizuje dane w jedną lub więcej tabel (albo "relacji") składających się z wierszy i kolumn zawierających unikalny klucz dla każdego wiersza.

Relacja jest zdefiniowana jako zbiór krotek, które mają takie same atrybuty. Krotka zwykle reprezentuje obiekt i związane z nim informacje. Obiekty są zwykle obiektami fizycznymi/materialnymi albo pojęciami. Relację przedstawia się zwykle jako tabelę, która jest zorganizowana w wiersze i kolumny. Wszystkie dane związane z konkretnym atrybutem mają tę samą dziedzinę i spełniają te same ograniczenia.

RDBMS

Relacyjny system zarządzania bazą danych

jest zbudowany w oparciu o model relacyjny.

Zgodnie z badaniami firmy Gartner z roku 2011, pięć przodujących dostawców komercyjnych systemów baz danych ze względu na dochody to Oracle (48.8%), IBM (20.2%), Microsoft (17.0%), SAP z Sybase (4.6%), Teradata (3.7%).

Trzy najważniejsze implementacje open source to: MySQL, PostgreSQL i SQLite. Ważnym rozgałęzieniem od MySQL jest MariaDB, która powstała po przejęciu MySQL AB przez Oracle.

Zalety

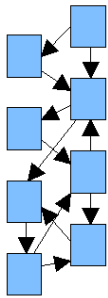
- łatwość użycia: aktualizacja jakiegokolwiek informacji jest precyzyjna i łatwa w zrozumieniu,
- elastyczność: możliwość uzyskania informacji w dokładnie takiej formie jak tego oczekujemy (odpowiednie zapytanie SQL),
- bezpieczeństwo: kontrola bezpieczeństwa i autoryzacji może być zaimplementowana znacznie łatwiej poprzez przeniesienie wrażliwych danych z wybranej tabeli do osobnej tabeli z jej własną kontrolą autoryzacji,
- niezależność danych: łatwiej uzyskać niezależne dane z wykorzystaniem znormalizowanych danych relacyjnych,

- DML (język manipulacji danymi): prosta struktura oznacza prosty język operujący na niej, dla innych struktur danych język musi być bardziej skomplikowany albo musi mieć ograniczone możliwości,
- dojrzałość i możliwość korzystania z nowoczesnych technik: wieloprocessorowość, działanie na wielu platformach, przetwarzanie równoległe, zaawansowane metody optymalizacji, zaawansowane metody zabezpieczeń, rozszerzony system transakcji.

Wady

- niezgodność pomiędzy językiem manipulacji danymi a językami programowania,
- wydajność: przy dużej liczbie tabel i złączeniach między nimi wydajność bardzo spada,
- wolny proces odkrywania znaczenia danych: jeśli dane są naturalnie zorganizowane w sposób hierarchiczny i w taki sposób przechowywane, pojęcie hierarchiczne może skutkować szybkim znalezieniem znaczenia danych,
- mały zbiór typów i struktur: brak funkcji użytkownika, procedur, typów, brak możliwości hierarchizacji danych, brak zagnieżdżonych danych, brak dziedziczenia, nie uwzględnienie czasu (punktu danych w czasie, czy wersjonowania).

Bazy zorientowane obiektowo



Obiektowa baza danych

to system zarządzania bazą danych, w którym informacje są reprezentowane w postaci obiektów jak ma to miejsce w programowaniu obiektowym.

- zapisanie wyniku pracy programistów w bazie danych jako obiektów,
- możliwość powielania lub zmiany istniejących obiektów wewnątrz bazy danych,
- pozwalają zachować spójność między językiem programowania i danymi przechowywanymi w bazie - ten sam model reprezentacji danych,

Przykłady: Caché, Db4o, Versant Object Database, ObjectStore, Objectivity/DB, Matisse, ...

Cechy techniczne

- udostępniają pewien rodzaj języka zapytań, który pozwala na wyszukiwanie obiektów z wykorzystaniem deklaratywnego podejścia programistycznego,
- integracja języka zapytań oraz interfejsy nawigacyjne, to obszary, w których są największe różnice pomiędzy producentami,
- była próba standaryzacji przez ODMG ale nieudana (Object Query Language, OQL),
- szybszy dostęp do danych - często nie potrzebne są złączenia,
- operowanie na całych obiektach,
- wiele obiektowych baz danych udostępnia wsparcie dla wersjonowania.

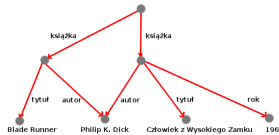
Zalety

- obiekty nie wymagają składania i rozkładania co zaoszczędza czas podczas pisania kodu oraz czas wykonywania,
- lepsza kontrola współbieżności - można zablokować całą hierarchię obiektów.
- model danych oparty na świecie rzeczywistym (albo przynajmniej lepiej go oddający).
- dobrze działają w architekturze rozproszonej,
- aplikacje zorientowane obiektowo wymagają mniej kodu.

Wady

- mniejsza efektywność przy prostych, nieskomplikowanych danych i prostych relacjach,
- tabele relacyjne są prostsze,
- istnieje więcej narzędzi dla użytkowników RSZBD,
- standardy przyjęte dla RSZBD są dużo stabilniejsze,
- wsparcie dla RSZBD jest dużo pewniejsze i ewentualna konieczność zmian jest dużo mniej prawdopodobna.

Semistrukturalne bazy danych



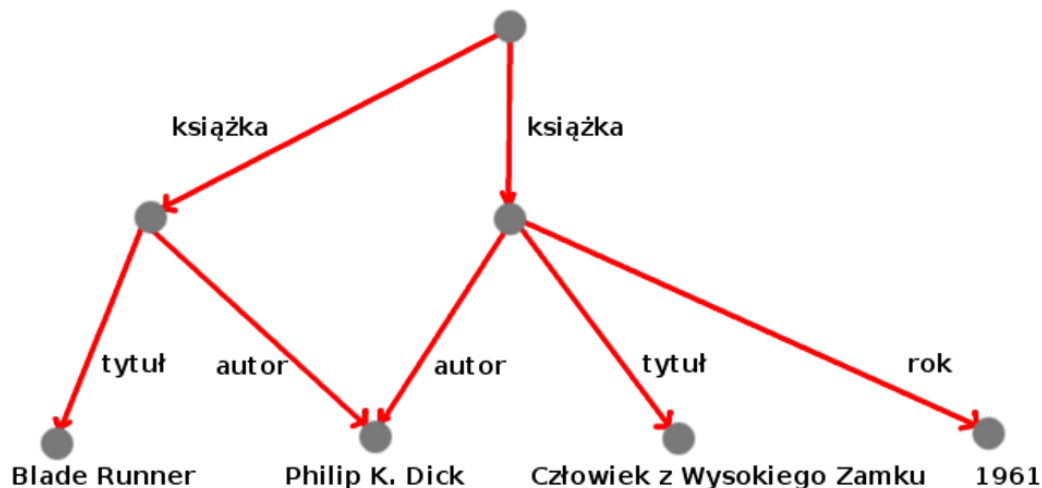
Baza danych semistrukturalnych

to baza, w której nie ma podziału na dane i ich schemat. W takich bazach liczba stosowanych struktur jest zależna od potrzeb.

Zalety takiego modelu są następujące:

- Mogą reprezentować informacje ze źródeł danych, które nie mogą być określone/ograniczone żadnym schematem.
- Dostarczają elastyczny format wymiany danych pomiędzy różnymi typami baz danych.
- Duża łatwość w zmianie schematu danych.
- Mogą służyć jako przenośny format transmisji danych.

Semistrukturalne bazy danych



Bazy danych XML

Bazy danych XML

pozwalają na określenie danych i czasem ich przechowywanie w formacie XML. Dane takie mogą być odpytywane, transformowane, eksportowane czy zwracane do systemu z którego wywołano zapytanie.

W różnych publikacjach dane takie są traktowane jako semistrukturalne albo NoSQL (zorientowane na dokumenty). Przykładami natywnych baz XML'owych są:

- dbXML
- eXist
- Sedna
- BaseX

Bazy danych XML

- Dano-centriczne pobieranie informacji XML:
 - XML jest używany jako medium transmisyjne
 - regularna struktura dokumentów
 - dane na odpowiednim poziomie szczegółowości z ograniczoną liczbą detali
 - dane pochodzą z bazy danych albo będą zapisane w bazie danych
- Podejście dokumento-centriczne
 - dokumenty w większości przypadków są tworzone ręcznie
 - nieregularna struktura dokumentów
 - relatywnie duża granulacja danych

Post-relacyjny bazy danych

Post-relacyjne bazy danych

są umowną nazwą oznaczającą bazy danych oferujące hybrydowe podejście: kombinację relacyjnych, zorientowanych obiektowo, semistrukturalnych i innych baz danych.

Niektóre z rozszerzeń do modelu relacyjnego integrują pojęcia, które poprzedziły model relacyjny. Na przykład pozwalają na reprezentację grafu skierowanego.

Baza obiektowo-relacyjna

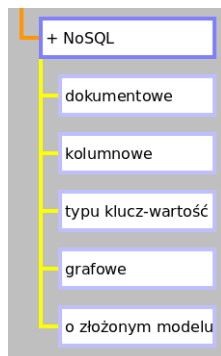
ORSZBD

jest systemem zarządzania bazą danych (SZBD) podobnym do baz relacyjnych, ale z modelami danych zorientowanymi obiektowo: obiekty, klasy, dziedziczenie są bezpośrednio wspierane w schematach bazy danych jak również w języku zapytań. Dodatkowo, tak jak surowe systemy relacyjne, wspierają rozszerzanie modelu danych o dowolnie zdefiniowane, niestandardowe typy i metody.

ORSZBD oferują:

- dane multimedialne i przestrzenne,
- typy złożone, abstrakcyjne i referencyjne, kolekcje, rekordy, zbiory, listy, perspektywy obiektowe, tabele obiektowe, tabele zagnieżdżone, tablice o zmiennej długości, i inne.
- klasy, metody, funkcje i procedury zdefiniowane w innym języku, dziedziczenie, agregacje, przeciążanie,
- język zapytań SQL3 nazywany również językiem Obiektowym SQL.

Bazy danych NoSQL



Bazy danych NoSQL

(np. Not only SQL) dostarczają mechanizmy składowania i dostępu do danych, które są opisane modelem innym niż relacje zapisane w tabelach z modelu relacyjnego. Motywacja:

- uproszczenie projektowania,
- łatwa skalowalność pozioma,
- lepsza kontrola dostępności.

Używane struktury danych (inne niż w bazach relacyjnych): klucz-wartość, rodzina kolumn, dokument i graf.

Typy baz NoSQL I

- Klucz-wartość (dane są reprezentowane jako kolekcje par klucz-wartość; każdy klucz pojawia się najwyżej raz w całej kolekcji). Przykłady: Redis, Memcached, Amazon DynamoDB, RiakKV, Ehcache, Hazelcast, OrientDB.
- Rodziny kolumn (dwu wymiarowa mapa, w której każdy klucz ma jedną lub więcej par klucz/wartość; możliwe jest składowanie i używanie bardzo dużych danych, nie ma jasno określonej struktury. Przykłady: Cassandra, HBase, Accumulo, Hypertable, Sqrrl, Google Cloud Bigtable, ScyllaDB.

Typy baz NoSQL II

- Dokumentów (przechowują dokumenty w jednym ze standardowych formatów - XML, YAML, JSON, również BSON). Przykłady: MongoDB, Couchbase, CouchDB, RavenDB, Cloudant, RethinkDB.
- Grafowe (dla danych, w których relacje dobrze przedstawić za pomocą grafu - elementy są połączone nieokreśloną liczbą relacji pomiędzy nimi; (sieci społecznościowe, transport publiczny, mapy drogowe,...) Przykłady: Neo4j, Titan, Giraph, Allegro-Graph, InfiniteGraph, OrientDB.
- O złożonym modelu: Amazon DynamoDB, MarkLogic, OrientDB, Virtuoso, ArangoDB, Kdb+, AllegroGraph, Stardog, ...

Najnowszy ranking systemów baz danych

323 systems in ranking, April 2017

Rank			DBMS	Database Model	Score		
Apr 2017	Mar 2017	Apr 2016			Apr 2017	Mar 2017	Apr 2016
1.	1.	1.	Oracle +	Relational DBMS	1402.00	+2.50	-65.54
2.	2.	2.	MySQL +	Relational DBMS	1364.62	-11.46	-5.49
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1204.77	-2.72	+69.72
4.	4.	↑5.	PostgreSQL +	Relational DBMS	361.77	+4.14	+58.05
5.	5.	↓4.	MongoDB +	Document store	325.43	-1.51	+12.98
6.	6.	6.	DB2 +	Relational DBMS	186.66	+1.74	+2.57
7.	7.	7.	Microsoft Access	Relational DBMS	128.18	-4.76	-3.79
8.	8.	8.	Cassandra +	Wide column store	126.18	-3.01	-3.49
9.	↑10.	9.	Redis +	Key-value store	114.36	+1.35	+3.12
10.	↓9.	10.	SQLite	Relational DBMS	113.80	-2.39	+5.83
. . .							
21.	21.	↓20.	Neo4j +	Graph DBMS	34.91	+0.59	+3.00

<http://db-engines.com/en/ranking>

39 spośród 100 najważniejszych SZBD, to bazy NoSQL.

Bardzo dynamiczna branża przemysłu informatycznego:

- nowe bazy NoSQL powstają co roku,
- powstają też nowe funkcjonalności.

Dane sprzed 6 m-cy

316 systems in ranking, October 2016

Rank			DBMS	Database Model	Score		
Oct 2016	Sep 2016	Oct 2015			Oct 2016	Sep 2016	Oct 2015
1.	1.	1.	Oracle	Relational DBMS	1417.10	-8.46	-49.85
2.	2.	2.	MySQL +	Relational DBMS	1362.65	+8.62	+83.69
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1214.18	+2.62	+90.95
4.	↑5.	4.	MongoDB +	Document store	318.80	+2.81	+25.54
5.	↓4.	5.	PostgreSQL	Relational DBMS	318.69	+2.34	+36.56
6.	6.	6.	DB2	Relational DBMS	180.56	-0.62	-26.25
7.	7.	↑8.	Cassandra +	Wide column store	135.06	+4.57	+6.05
8.	8.	↓7.	Microsoft Access	Relational DBMS	124.68	+1.36	-17.16
9.	↑10.	↑10.	Redis	Key-value store	109.54	+1.75	+10.75
10.	↓9.	↓9.	SQLite	Relational DBMS	108.57	-0.05	+5.90
. . .							
21.	21.	21.	Neo4j +	Graph DBMS	36.45	+0.08	+3.04

<http://db-engines.com/en/ranking>

Zalety

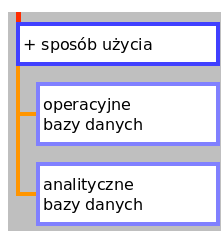
- prostsze skalowanie dla bardzo dużych zbiorów danych,
- bazy NoSQL są zwykle szybsze - czasem ekstremalnie szybsze - jeśli chodzi o zapis,
- odczyt również może być bardzo szybki zależnie od typu bazy NoSQL i danych, które są pobierane,
- brak określonej struktury danych - duża elastyczność,
- bardzo szybko się rozwijają i ciągle są aktywnie tworzone,
- możliwość wyboru z różnych modeli danych - można wybrać to co nam najbardziej odpowiada.

Wady

- większość baz NoSQL jest jeszcze w fazie rozwojowej, braki w funkcjonalności,
- niski poziom wsparcia,
- mało udogodnień dla zapytań i przeprowadzania analiz - nie ma języka SQL,
- wymagają wysokich umiejętności i znacznego wysiłku aby je zainstalować i nimi zarządzać,
- mało specjalistów, wielu z nich ciągle się jeszcze uczy ...

1.4 sposób użycia

Analityczne i operacyjne bazy danych



- **Operacyjne bazy danych** nazywane również OLTP (ang. *On Line Transaction Processing*), są używane do zarządzania (dynamicznymi) danymi w czasie rzeczywistym, kładące nacisk na efektywność transakcji, dostarczające narzędzia do codziennych operacji na danych.
- **Analityczne bazy danych** nazywane również OLAP (ang. *On Line Analytical Processing*) pozwalają użytkownikom na interaktywną analizę wielowymiarowych danych z różnych perspektyw.

Operacyjne bazy danych

- używane dla informacji biznesowych napływających w czasie rzeczywistym,
- pozwalają przedsiębiorstwom wprowadzać, gromadzić i pobierać duże ilości dokładnie określonych informacji, jak dane o firmie, dane finansowe, ...
- wykorzystane do zarządzania danymi biznesowymi o znaczeniu krytycznym:
 - do monitorowania czynności,
 - do badania podejrzanych transakcji
 - do przejrzania historii kontaktów z konkretnym klientem,
- mogą być one częścią rzeczywistego procesu podejmowania i finalizowania zakupu, np. w e-commerce,
- zoptymalizowane do szybszego przetwarzania transakcji (operacje CRUD).

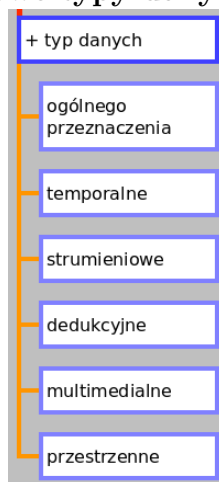
Analityczne bazy danych

- wspierają podejmowanie decyzji,
- składają się z narzędzi OLAP i hurtowni danych,
- duży nacisk na integrację różnych elementów danych - spójny obraz całości,
- u podstaw każdego systemu OLAP leży kostka OLAP (wielowymiarowa kostka danych, hiperkostka),

- użycie wielowymiarowego modelu danych - fakty i wymiary,
- skomplikowane zapytania analityczne,
- fakty - informacje podlegające analizie charakteryzowane ilościowo za pomocą miar,
- wymiary - ustalają kontekst analizy, mogą składać się z poziomów, które tworzą hierarchię,
- analizowane wartości są rozmieszczone na przecięciach hipersześcianu według wymiarów.

1.5 typ danych

Nowe typy danych - nowe wyzwania



- nowe typy danych pochodzące z różnych źródeł (np. dane multimedialne, z kamer przemysłowych, uwzględniające czas,...)
- duża szybkość napływania danych, przetwarzanie w czasie rzeczywistym.

Temporalne bazy danych

Temporalna baza danych

- baza danych z wbudowanym wsparciem do obsługi danych uwzględniających czas. Chodzi tutaj o dane temporalne przechowywane w bazie jak również język zapytań SQL rozszerzony do algebry relacyjnej uwzględniającej czas.

Cechy:

- temporalny język definicji, modyfikacji i wyszukiwania danych,
- wsparcie temporalnych ograniczeń integralnościowych,
- algebra relacyjna rozszerzona o operacje uwzględniające czas.

Przykłady: TimeDB, Tiger.

Cechy I

- Dla każdej danej przechowuje się:
 - czas ważności (okres, w którym dany fakt jest prawdziwy względem świata rzeczywistego)
 - czas transakcji (okres, w którym dany fakt jest zapisany w bazie danych).
- Czas transakcyjności zarządzany przez system.

- Typ danych przedziału czasowego, uwzględniający również nieskończone przedziały czasowe,
- Temporalne klucze główne, uwzględniające również nie nakładanie się przedziałów czasowych.
- Nakładanie temporalnych ograniczeń, np. unikalność przedziałów czasowych czy integralność referencyjna.

Cechy II

- Aktualizacja i usuwanie danych temporalnych z automatycznym dzieleniem i łączeniem okresów czasowych w spójną całość.
- Zapytania czasowe uwzględniające obecne dane, dane z punktów czasowych w przeszłości, w przyszłości albo dotyczące przedziału czasowego.
- Predykaty dla zapytań uwzględniających przedziały czasowe, często bazujące na relacjach interwałowych Allena.

Strumieniowe systemy baz danych

Strumieniowy system baz danych(ang. data stream management system - DSMS)

to program komputerowy zarządzający strumieniami danych. Jest podobny do systemu zarządzania bazą danych. Dane są przedstawione w postaci zbioru strumieni danych. System umożliwia analizę serii czasowych i przetwarzanie danych napływających z dużą szybkością.

Cechy

- ciągłe zapytania (plan realizacji jest zamknięty w martwej pętli),
- strumień danych ma charakter nieskończony (odmienne operatory złączenia i funkcje agregujące),
- zapytania predefiniowane i zadawane ad-hoc,
- algebry strumieni danych.

Zastosowania

Zastosowania strumieniowych systemów baz danych:

- medyczne systemy monitorujące,
- kontrola sieci czujników,
- monitorowanie ruchu danych w sieciach,
- wspomaganie bieżącej analizy finansowej,
- bieżąca analiza transakcji,
- zastosowania militarne, itp...

Przykłady: Stream, Borealis, TelegraphCQ, NiagaraCQ, Cougar.

Dedukcyjne bazy danych

Dedukcyjna baza danych

to system bazodanowy, który potrafi dokonywać wnioskowania (dedukować dodatkowe fakty) bazując na zasadach i faktach zapisanych w bazie. Do specyfikowania faktów, zasad i zapytań zwykle używa się języka Datalog.

Dedukcyjne bazy danych wyrosły z chęci połączenia programowania logicznego z relacyjnymi bazami danych w celu budowy systemów bardzo mocno wspierających formalizm, które nadal są szybkie i zdolne poradzić sobie z bardzo dużymi zbiorami danych. Dedukcyjne bazy danych są bardziej wyraziste niż relacyjne bazy danych, ale mniej wyraziste niż systemy programowania logiki. W ostatnich latach, dedukcyjne bazy danych, takie jak Datalog znalazły nowe zastosowania w integracji danych, ekstrakcji informacji, tworzeniu sieci, analizie programów, bezpieczeństwie czy obliczeniach w chmurze.

Multimedialne systemy baz danych I

Multimedialne systemy baz danych

- ich przeznaczeniem jest przechowywanie i wyszukiwanie danych dotyczących zawartości multimedialnych (nagrań muzycznych, filmów, grafiki, itp.).

Klasyfikacja danych:

- ciągłe (dźwięk, mowa, animacja, video) - uwzględnienie wymiaru czasu
- dyskretne (tekst, obraz, grafika komputerowa, typy konwencjonalne) - bez uwzględniania czasu.

Multimedialne systemy baz danych II

Wyzwania:

- duży rozmiar danych,
- brak standardu przechowywania informacji multimedialnej,
- sposób przesyłania danych multimedialnych nie jest ujednolicony,
- kompresja,
- synchronizacja w odtwarzaniu różnych elementów przekazu,
- język zapytań dostosowany do danych multimedialnych.

Bazy danych przestrzennych

Baza danych przestrzennych (danych geograficznych)

to baza, która jest zoptymalizowana pod kątem przechowywania i odpytywania danych reprezentujących obiekty zdefiniowane w przestrzeni geometrycznej.

Rodzaje danych:

- proste obiekty geometryczne: punkty, linie, wielokąty,
- bardziej skomplikowane struktury: obiekty 3D, relacje topologiczne, sieci liniowe, obiekty TIN (Triangulated Irregular Network).

Typowe bazy danych są projektowane do zarządzania różnymi danymi numerycznymi czy napisowymi. Do skutecznej obsługi danych przestrzennych wymagane jest dodanie do systemu bazodanowego specjalnej funkcjonalności. Konsorcjum OGC (ang. *Open Geospatial Consortium*) odpowiada za tworzenie i rozwój standardów dla danych i usług przestrzennych, systemów informacji geograficznej (GIS), funkcjonalności implementowanej w bazach danych.

Cechy

- Pomiar przestrzenny: obliczanie długości linii, powierzchni wielokątów, odległości między obiektami geometrycznymi, itp.
- Funkcje przestrzenne: modyfikowanie istniejących cech w celu tworzenia nowych, na przykład tworzeni bufora wokół jakiegoś obiektu, sprawdzania punktów przecięcia dwóch obiektów itp.
- Predykaty przestrzenne: pozwalają na zapytania zwracające prawdę/fałsz odnośnie relacji przestrzennych między geometriami. Przykłady zapytań:
 - "czy dwa wielokąty na siebie nachodzą"
 - "czy w odległości 10km od miejsca, w którym chcemy utworzyć wysypisko znajduje się jakiś dom"?
- Konstruktory geometryczne: tworzenie nowych geometrii, zwykle określając wierzchołki, które określają kształt.
- Funkcje obserwatorów: zapytania, które zwracają konkretne informacje o jakiejś własności, np. lokalizacja środka koła.

1.6 zużycie zasobów

Różnice w zapotrzebowaniu na zasoby



W nowoczesnych bazach danych trzeba wziąć pod uwagę ilość zasobów pamięciowych, które są używane, np. na małych urządzeniach mobilnych.

Mikro bazy danych

Systemy mikro baz danych

- maksymalnie okrojone i uproszczone systemy baz danych.

Cechy:

- prostota,
- funkcjonalność wystarczająca na potrzeby wielu aplikacji,
- przenośność,
- cała baza danych bardzo często w jednym pliku,
- mikroskopijny rozmiar,
- kompilacja poleceń SQL to kodu maszyny wirtualnej (SQLite).

Przykłady: SQLite, HSQLDB, tinySQL, picoSQL, Mckoi SQL, Axion etc...

Bazy dla urządzeń mobilnych

Bazy danych dla urządzeń mobilnych

- są zarówno bazami, z którymi można połączyć się za pomocą mobilnego urządzenia komputerowego (np. smartfonu czy tabletu) poprzez sieć komórkową jak również bazami danych, które rzeczywiście znajdują się na urządzeniu przenośnym. Może być to lista kontaktów, informacje o cenach, przebyta odległość, lub wszelkie inne informacje.

Przykłady: SQL Anywhere, SQL Server Compact, SQL Server Express, Oracle Database Lite, SQLite, SQLBase, Sparksee 5 mobile, Couchbase Lite, iBoxDB, Realm, CryptonorDB

Rozważania

- Użytkownicy urządzeń mobilnych muszą być w stanie pracować bez stałego podłączenia do sieci (awaria, brak zasięgu). Pamięć podręczna może być wykorzystana do przechowania ostatnio używanych danych czy transakcji aby nie zostały utracone w wypadku awarii połączenia.
- Mobilne urządzenia komputerowe mają zwykle wolniejsze procesory i ograniczoną żywotność.
- Użytkownicy korzystający z wielu urządzeń (np. smartfon i tablet) muszą synchronizować dane w jednej centralnej bazie. Może to wymagać pewnej automatycznej funkcjonalności specyficznej dla używanej aplikacji.
- Użytkownicy mogą zmienić położenie geograficzne czy położenie w sieci. Zwykle zarządzaniem połączeniami sieci bezprzewodowych zajmuje się system operacyjny.

Bazy danych czasu rzeczywistego

Baza danych czasu rzeczywistego

- to system bazodanowy, który używa przetwarzania w czasie rzeczywistym do obsługi obciążeń, których stan ciągle się zmienia. Klasyczne bazy zawierają trwałe dane, zwykle niezależne od czasu.

Przetwarzanie w czasie rzeczywistym:

- transakcje są przetwarzane na tyle szybko aby wynik mógł być natychmiast zwrócony i wykorzystany,
- używane w rachunkowości, bankowości, prawie, dokumentacji medycznej, kontroli procesów, systemach rezerwacji czy naukowej analizie danych.

Bazy danych w pamięci operacyjnej I

Main Memory Database System

- baza danych w pamięci operacyjnej, gdzie plik bazy danych jest odwzorowywany w wirtualną przestrzeń pamięci przydzielonej do danej aplikacji.

Przykłady: DataBlitz, FastDB, Oracle TimesTen, Polyhedra, eXtremeDB, Monet, Tachyon.

Bazy danych w pamięci operacyjnej II

Cechy:

- bardzo szybkie wykonywanie zapytań (użycie specyficznych algorytmów),
- pionowa fragmentacja - podzbiory kolumn,
- natychmiastowe odtwarzanie po awarii,
- stosowanie pobierania wyprzedzającego,
- konieczność stosowania tradycyjnych baz danych jako back-office,
- sortowanie indeksów złączeniowych.

2 Źródła

Źródła

- <http://en.wikipedia.org>
- S. Kozielski, B. Małyśiak, P. Kasprowski, D. Mrożek, Bazy Danych: Modele, Technologie, Narzędzia, WKŁ 2005
- <http://www.cs.put.poznan.pl/mmorzy/>
- <http://www.ipipan.waw.pl/~subieta/prezentacje/Obiektowe\%20bazy\%20danych\%20kontra\%20relacyjne\%201997.ppt>
- C.Zaniolo, S.Ceri, Ch.Faloutsos, R.T. Snodgrass, V. S. Subrahmanian, R.Zicari, Advanced Database Systems, Morgan Kaufmann, 1997
- http://wazniak.mimuw.edu.pl/index.php?title=Zaawansowane_systemy_baz_danych
- Materiały do wykładów Pani mgr. Wandy Kik
- "O modelach danych i innych rzeczach, które utrudniają życie informatykom" Krzysztof Goczyła, VI Konferencja PLOUG 2000
- M. Lentner, Oracle 9i Kompletny podręcznik użytkownika, PJWSTK - W-wa, 2003
- <http://www.odbms.org/>
- <http://www.objectivity.com/products/objectivitydb/>
- http://db-engines.com/en/ranking_categories