

Technologie bazodanowe

Rozproszony system baz danych

jest zbiorem współpracujących ze sobą systemów baz danych, z których każdy znajduje się w innym węźle sieci. Z punktu widzenia użytkownika wszystkie te bazy logicznie stanowią jedną rozproszoną bazę danych.

Zalety:

- dane umieszczone „ blisko „ użytkownika – skrócenie czasu transmisji sieciowej
- mniejsze ryzyko utraty wszystkich danych na skutek awarii systemu
- wzrost niezawodności całego systemu

Wady:

- utrudniony dostęp do pełnego (zintegrowanego) zbioru danych
- konieczność utrzymywania kopii danych (replik) i ich odświeżania.

C.J.Date zaproponował 12 reguł jakie powinien spełniać system rozproszonej bazy danych

1. Lokalna autonomia
2. Uniezależnienie od centralnego miejsca
3. Działanie ciągle
4. Niezależność lokalizacji
5. Niezależność fragmentacji

Wykład VI

6. Replikacja
7. Niezależność sprzętowa
8. Niezależność od systemu operacyjnego
9. Niezależność od systemu zarządzania bazą danych
10. Niezależność od sieci
11. Rozproszone zarządzanie transakcjami
12. Rozproszone przetwarzanie zapytań

Aktywny system baz danych

- system, który jest aktywny nawet wtedy, gdy nie są do niego jawnie kierowane żadne transakcje czy zapytania.

Zmiana stanu bazy następuje na skutek:

- zajścia określonego zdarzenia zewnętrznego
- zakończenia realizacji określonego zbioru transakcji kierowanych do SZBD
- upływu określonego kwantu czasu
- kombinacji dwóch powyższych przypadków

Aktywne bazy danych wykorzystują:

- model definiowania aktywnych reguł: Event – Condition – Action (*wystąpienie zdarzenia, weryfikacja warunku, odpalenie akcji*)
- modele aktywności – zależności czasowe i przyczynowo skutkowe między zdarzeniami i akcjami
- operatory zdarzeniowe – zbiór operatorów umożliwiających specyfikację złożonych wyrażeń zdarzeniowych

Semistrukuralny system baz danych

- zapewnia trwałe przechowywanie danych w strukturach XML, wykorzystuje języki zapytań takie jak: XQUERY, Xpath, XQL, XML_QL, posiada bardzo dobrą przenaszalność. Umożliwia zagnieżdżanie i hierarchizację danych.

przykłady: DbXML, eXist, BaseX, Sedna

Podejście dano – centryczne:

- XML wykorzystywane jako medium transportowe
- regularna struktura dokumentów
- dane pochodzą z bazy danych lub będą wprowadzane do bazy danych

Podejście dokumento – centryczne

- dokumenty najczęściej tworzone ręcznie
- nieregularna struktura dokumentów
- stosunkowo duża ziarnistość danych

Multimedialny system baz danych

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

Wykład VI

Klasyfikacja danych:

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

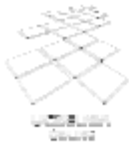
Wyzwania:

- 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

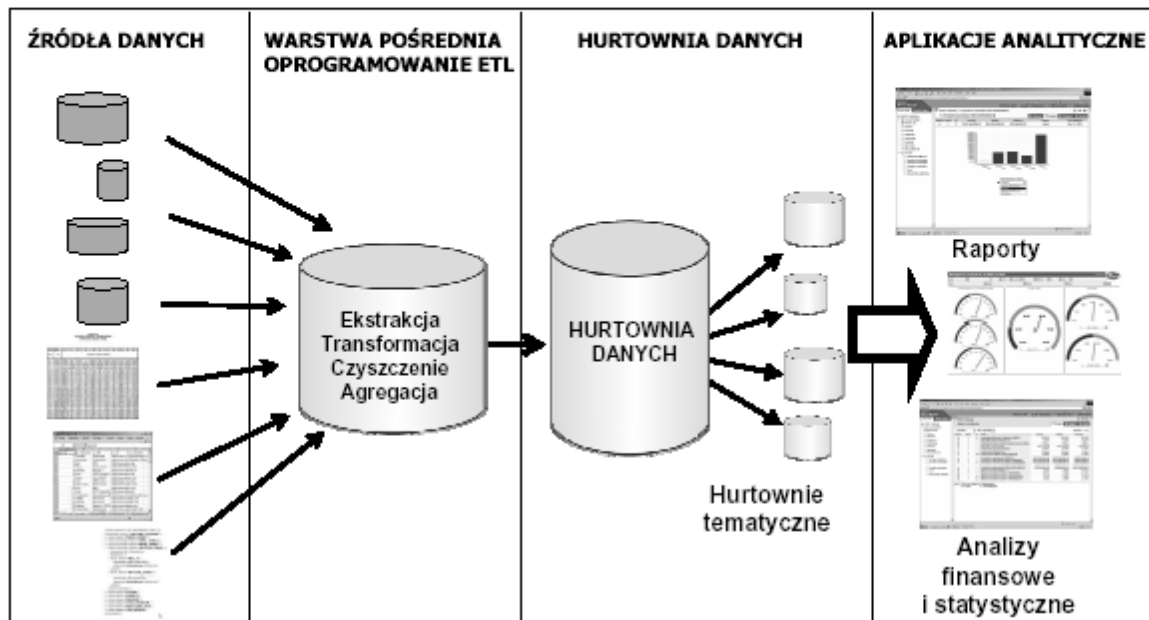
Hurtownia danych

- jest bardzo dużą bazą danych posiadającą następujące cechy:
- dane w niej składowane nie są modyfikowane przez użytkowników, są jedynie przez nich odczytywane
- zawiera wszystkie dane historyczne i bieżące
- zawiera dane zagregowane na wielu poziomach szczegółowości
- zawartość hurtowni jest zorientowana tematycznie

przykłady: Oracle, DB2 UDB, SybaseIQ, SAS Enterprise BI Server, MS SQL Server,



Podstawowa architektura HD



ETL

Oprogramowanie ETL (*Extraction, Translation, Loading*) realizuje tzw. procesy ETL, składające się z trzech następujących faz:

- odczytu danych ze źródeł (*Extraction*)
- transformacji ich do wspólnego modelu wykorzystywanego w magazynie wraz z usunięciem wszelkich niespójności (*Translation*)
- wczytywanie danych do magazynu (*Loading*)

OLAP

- (*ang. On – Line Analytical Processing*) - aplikacje analityczne zorientowane na wspieranie procesów decyzyjnych, przetwarzanie danych historycznych i zagregowanych.

Aplikacje analityczne wspomagają decyzje i służą do eksploracji danych, wykonują zaawansowane analizy, wspomagają zarządzanie przedsiębiorstwem, np.:

- analiza trendów sprzedaży
- analiza nakładów reklamowych i zysków
- analiza ruchu telefonicznego

Wielowymiarowy model danych

- dane zorganizowane w postaci wielowymiarowego modelu danych (*ang. Multidimensional Data Model*) - fakty i wymiary
- fakty – informacje podlegające analizie charakteryzowane ilościowo za pomocą miar
- wymiary – ustalają kontekst analizy, składają się z poziomów, które tworzą hierarchię, zależności hierarchiczne między poziomami tworzą tzw. strukturę wymiaru

Strumieniowy system baz danych

- dane są przedstawione w postaci zbioru strumieni danych, umożliwiające analizę serii czasowych i przetwarzanie danych napływających z dużą szybkością

Cechy

- ciągle 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
- język zapytań oparty o zmodyfikowaną/rozszerzoną/alternatywną algebrę opisującą operacje na strumieniach danych

Zastosowania strumieniowych systemów baz danych

- medyczne systemy monitorujące
- kontrola sieci czujników
- monitorowanie ruchu danych w sieciach telekomunikacyjnych, serwerów WWW
- wspomaganie bieżącej analizy finansowej: giełdy, sieci bankomatów
- bieżąca analiza transakcji
- zastosowania militarne

przykłady:

STREAM (*The Stanford Data Stream Management System*) - najważniejszą ideą wypracowaną przez zespół projektu STREAM jest model strumieni danych, jak również język CQL (*continuous query language*), w którym rejestruje się ciągle zapytania w systemie. Telegraph projekt Berkeley, NiagaraCQ,

Temporalny system baz danych

- dla każdej danej przechowuje jej czas ważności (okres, w którym dany fakt jest prawdziwy względem świata rzeczywistego) oraz czas transakcji (okres, w którym dany fakt jest zapisany w bazie danych)

Cechy

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

przykłady systemów – TimeDB, Tiger

Mikro system baz danych

- okrojony, uproszczony system baz danych

Cechy

- prostota
- funkcjonalność zaspokajająca potrzeby wielu aplikacji
- mikro – rozmiar
- kompilacja poleceń SQL do wirtualnego kodu maszynowego

Przykłady: SQLite, HSQLDB, tinySQL, picoSQL, Axion

Zalety i wady modelu relacyjnego

Zalety:

- intuicyjność modelu
- łatwość przejścia z systemów plików
- wykorzystanie wieloprocesowości
- wieloplatformowość
- zapytania równoległe
- zaawansowane metody optymalizacji, metody bezpieczeństwa danych
- rozbudowany system transakcji

Wady:

- ubogi zestaw typów
- brak funkcji i procedur użytkownika
- problemy z hierarchizacją
- brak danych zagnieżdżonych
- brak dziedziczenia
- niedopasowanie pomiędzy językiem operowania danymi a językami tworzącymi aplikacje – **niezgodność impedancji**
- brak uwzględnienia wersji i czasu

Niezgodność impedancji (w zakresie)

- składni
- systemu typów
- semantyki i paradygmatów języków
- pragmatyki użycia
- faz i mechanizmów wiązania
- przestrzeni nazw i reguł zakresu
- schematów iteracyjnych
- traktowania cechy trwałości danych
- środków programowania ogólnego

Najczęściej stosowane rozwiązania najczęściej wśród SZBD

- Obiektowe bazy danych
- Postrelacyjne bazy danych – model relacyjny poszerzony o

Wykład VI

- elementy obiektowości
- obsługę XML
- rozwiązania analityczne
- zapytania historyczne,

Model obiektowy

- obiekt jako podstawowa składowa
- cechy obiektów: atrybuty i związki
- funkcjonalność obiektu: metody
- tożsamość obiektu
- hermetyczność obiektów
- dziedziczenie; współdzielenie implementacji i relacja podtypu
- przeciążenie i dynamiczne wiązanie funkcjonalności obiektów

Zalety

- odzwierciedlenie dowolnych obiektów świata rzeczywistego w postaci złożonych typów
- możliwość wprowadzenia agregacji i dziedziczenia
- zmniejszenie dystansu pomiędzy fazami analizy, projektowania i implementacji
- dopasowanie, bądź osadzenie wewnątrz języków programowania aplikacji
- uwzględnienie informacji proceduralnej (metody)
- zwiększenie poziomu abstrakcji w myśleniu programistów i użytkowników
- duża elastyczność

Wady

- brak standaryzacji
- problemy ze skalowalnością, niezawodnością i efektywnością
- słabo rozwinięta optymalizacja i bezpieczeństwo danych

Manifest baz obiektowych trzeciej generacji – 1989, 1990

Poniższy tekst z Kazimierz Subieta: **Słownik terminów z zakresu obiektowości**. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1999

„ Istotną rolę w rozwoju obiektowości odegrał manifest obiektowych systemów baz danych autorów: Atkinson, Bancilhon, DeWitt, Dittrich, Maier, Zdonik, który zrywając z ideologią modelu relacyjnego ustalił ramowe założenia obiektowości w bazach danych. Manifest postulował zachowanie klasycznych cech baz danych (trwałość, zarządzanie pamięcią pomocniczą, współbieżność, transakcje, odtwarzanie, rozproszenie, języki zapytań i inne), przystosowując je do cech obiektowości (złożone obiekty, tożsamość obiektów, hermetyzacja, typy lub klasy, dziedziczenie, przesłanianie połączone z dynamicznym wiązaniem, rozszerzalność). Reakcją na ten manifest był tzw. „manifest systemów baz danych trzeciej generacji” grupy konserwatywnych zwolenników obecnej filozofii systemów relacyjnych (Stonebrakera i innych), postulujący zachowanie wszystkich potwierdzonych w praktyce cech modelu relacyjnego (w szczególności SQL jako „intergalaktycznego języka danych”) i wzmocnienie go o nowe własności, m.in. obiektowe.

Manifesty w dziedzinie baz danych są odpowiedzią na brak naukowych i technologicznych kryteriów, które mogłyby obiektywnie wyznaczyć kierunki rozwoju tej dyscypliny. W tej sytuacji ekspert lub grupa ekspertów stawia się w roli proroka, próbującego *a priori* dyktować światu, jak ma budować swoją przyszłość. Niezbywalną cechą wszystkich manifestów - od komunistycznego do obiektowego - jest to, że większość wymienianych w nich postulatów jest rozsądna i zgodna z powszechnym odczuciem dobrego. Jest to psychologiczna baza, na której mogą być przemycane cechy utopijne, kontrowersyjne, lub nieakceptowalne. Zatem w stosunku do wszelkich manifestów należy zachować sporo rezerwy i sceptycyzmu. To właśnie mówi ostatnia doktryna manifestu obiektowych baz danych, nakazująca obowiązek kwestionowania wszystkich poprzednich doktryn tego manifestu.

Wykład VI

manifest obiektowych systemów baz danych (*Object-Oriented Database System Manifesto*) Dokument opracowany przez autorów: Atkinson, Bancilhon, DeWitt, Dittrich, Maier, Zdonik (1989) formułujący podstawowe założenia obiektowych baz danych w postaci charakterystyk, które są podzielone na trzy grupy:

- **Obowiązkowe**, czyli takie, które musi posiadać każdy system zarządzania bazą danych określany mianem „obiektowy”. Do nich należą: złożone obiekty, tożsamość obiektów, hermetyzacja, typy lub klasy, dziedziczenie, przesłanianie wraz z późnym wiązaniem, rozszerzalność, kompletność obliczeniowa, trwałość, zarządzanie pamięcią pomocniczą, współbieżność, odtwarzanie oraz udogodnienia dla zapytań *ad hoc*.
- **Opcyjne**, czyli takie, które nie są obowiązkowe, ale które mogą podnieść jakość systemu. Do nich zaliczono: wielokrotne dziedziczenie, kontrolę typów i wnioskowanie o typie, rozproszenie bazy danych, transakcje projektowe (długie lub zagnieżdżone) oraz wersje.
- **Otwarte**, czyli takie, gdzie projektanci systemów mają pewną dowolność co do ich wyboru. Do nich zaliczono: paradygmat programowania, system typów, system reprezentacji oraz zuniformizowanie.

manifest systemów baz danych trzeciej generacji (*Third-Generation Database System Manifesto*) Manifest będący reakcją na pojawienie się manifestu obiektowych baz danych zrywającego z założeniami modelu relacyjnego (co, jak można się domyśleć, wprowadziło pewną nerwowość w kręgach dostawców systemów relacyjnych). Został on opracowany przez następujących autorów: Stonebraker, Rowe, Lindsay, Gray, Carey, Brodie, Bernstein, Beach (1990). Manifest systemów baz danych trzeciej generacji (3GDB) zawiera trzy doktryny:

- 3GDB muszą wspomagać bogatsze struktury danych i reguły.
- 3GDB muszą posiadać wszystkie pozytywne cechy baz danych drugiej generacji.
- 3GDB muszą być otwarte dla innych systemów.

oraz 13 postulatów:

- 3GDB muszą posiadać bogaty system typów.
- Dziedziczenie jest dobrym pomysłem.
- Funkcje (w tym zapamiętane procedury i metody) plus hermetyzacja są dobrymi pomysłami.
- Unikalne identyfikatory powinny być stosowane wtedy, gdy nie są dostępne klucze główne.
- Reguły (wyzwalacze, więzy integralności) staną się główną cechą przyszłych systemów.
- Cały dostęp do bazy danych powinien odbywać się poprzez nieproceduralny język wysokiego poziomu.
- Kolekcje powinny być specyfikowane zarówno przez ich nazwę, jak i poprzez zapytanie. Chodzi o to, aby istniała możliwość przetwarzania kolekcji wyprodukowanych przez zapytanie, a nie tylko kolekcji zapamiętanych w bazie danych.
- Aktualizowalne perspektywy (*views*) są istotne.
- Własności fizyczne nie mają związku z modelem danych i powinny być z niego usunięte.
- 3GDB muszą być dostępne z wielu języków wysokiego poziomu.
- Języki te powinny być wyposażone w cechę trwałości i zintegrowane z językiem zapytań.
- Na lepsze i gorsze, SQL jest intergalaktycznym językiem danych.
- Zapytania i odpowiedzi na zapytania powinny być dolnym poziomem komunikacji klient-serwer (chodzi o to, aby ten poziom nie ograniczał się do przesyłania fizycznych stron). „

Wykorzystano

Wykłady dr inż. Olga Siedlecka-Lamch – Systemy baz danych z roku 2012

http://wazniak.mimuw.edu.pl/index.php?title=Zaawansowane_systemy_baz_danych