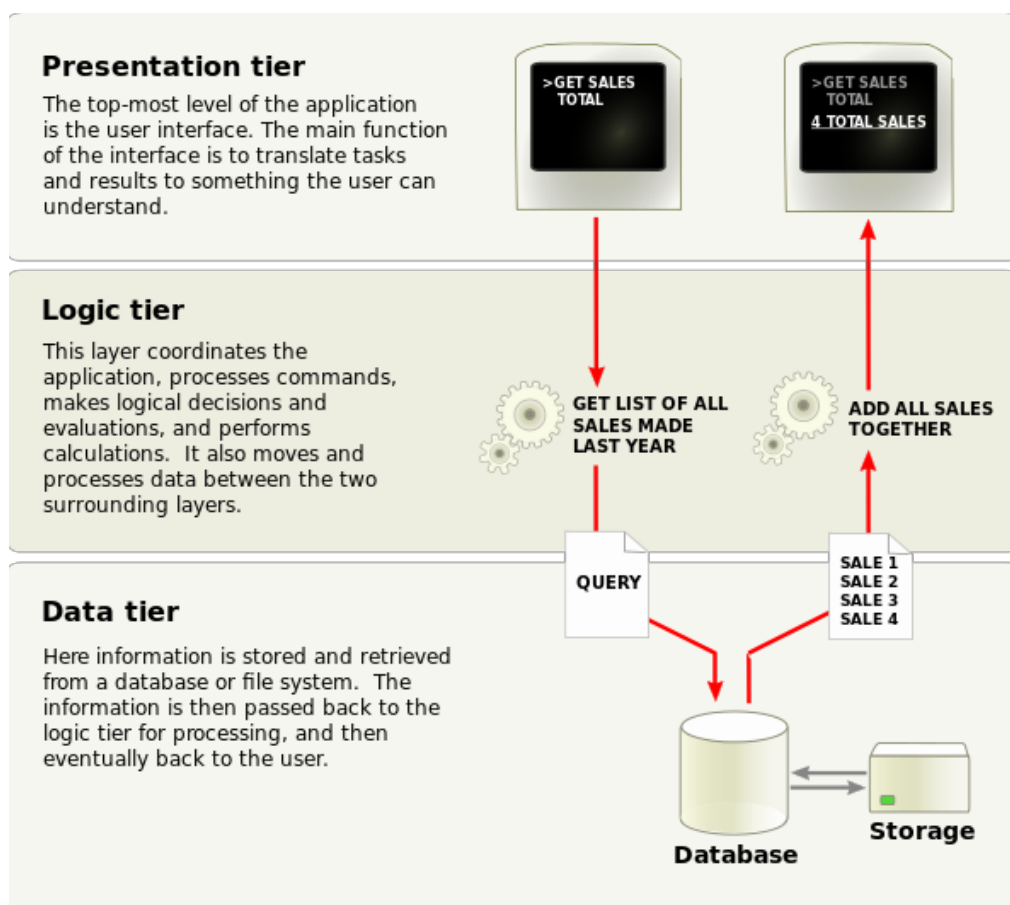
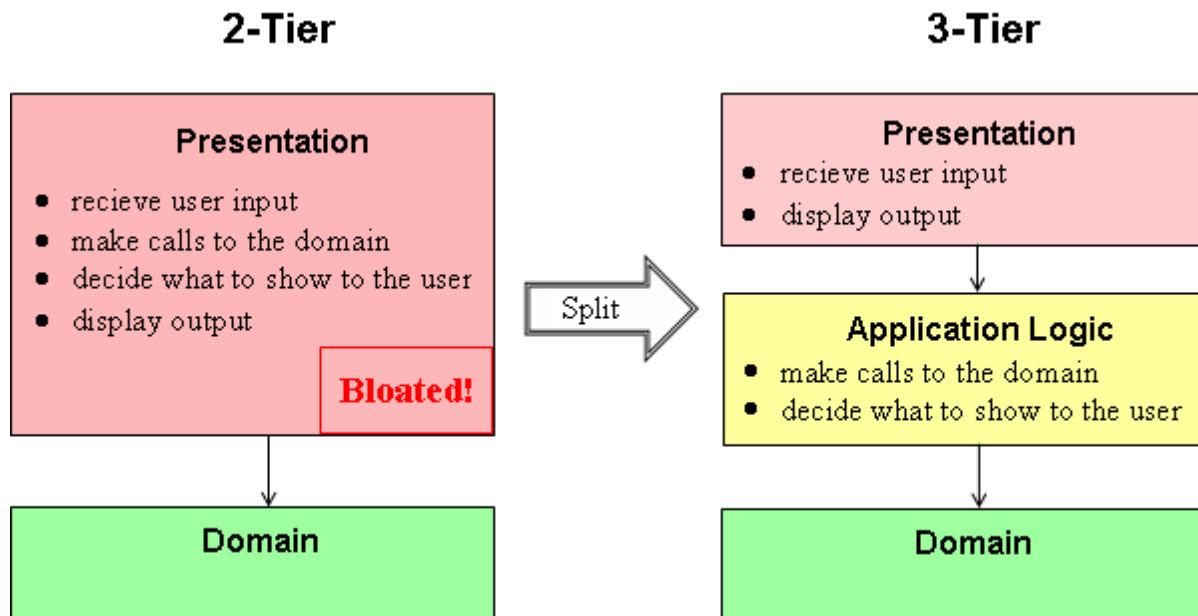


Bazy - opracowanie:

1. Wyjaśnić pojęcia i zasady tworzenia wielopoziomowych architektur klient/serwer.

Ogólnie: architektury 2/3 warstwowe, czyli modele z ciekim i grubym klientem.



2. Podać znaczenie i miejsce pojęć 'model gwiazdy' i 'model płatka śniegu' w systemach gromadzenia danych.

To nie o to chodzi! Mamy hurtownię danych, teraz chcemy użytkownikowi umożliwić zadawanie jej pytań tak, żeby mógł sobie znajdować interesujące go informacje. Powiedzmy, że damy mu 3 wartości do wyboru. Na ich podstawie otrzyma zestaw wyników [jakichś wielkości] związanych z ustawionymi przez niego danymi. Trzeba mu to jakoś zobrazować, żeby zrozumiał. Zauważmy, że to wygląda jak duża kostka, gdzie on ustala współrzędne i w efekcie otrzymuje małą kosteczkę "w której znajdują się obliczone wielkości". Ale można przecież dać mu więcej wymiarów -> wtedy wielowymiarową kostkę trzeba jakoś zobrazować. Np. w postaci gwiazdy, gdzie w centrum będą obliczone wielkości, a 'promień gwiazdy' to osie współrzędnych. Ale do tych ustawianych danych wejściowych można zażądać dodatkowych właściwości/atributów. Wtedy osie współrzędnych się rozgałęziają [odchodzą od nich takie mniejsze osie]. Otrzymujemy płatek śniegu.

http://en.wikipedia.org/wiki/Snowflake_schema http://en.wikipedia.org/wiki/Star_schema
http://www.diffen.com/difference/Snowflake_Schema_vs_Star_Schema - ODP: ogarnij opracowania na wiki, tam to jest fajnie zobrazowane, w tym Burczyka oraz w tym z Worda.

Warto również wiedzieć, że model gwiazdy i płatka śniegu przekłada się również na formy normalne bazy: w modelu gwiazdy tabele znajdują się drugiej, natomiast w płatku śniegu w trzeciej formie normalnej.

Model gwiazdy

Fact tables record measurements or metrics for a specific event. Fact tables generally consist of numeric values, and foreign keys to dimensional data where descriptive information is kept. Fact tables are designed to a low level of uniform detail (referred to as "granularity" or "grain"), meaning facts can record events at a very atomic level. This can result in the accumulation of a large number of records in a fact table over time.

Dimension tables usually have a relatively small number of records compared to fact tables, but each record may have a very large number of attributes to describe the fact data. Dimensions can define a wide variety of characteristics, but some of the most common attributes defined by dimension tables include:

- Time dimension tables describe time at the lowest level of time granularity for which events are recorded in the star schema
- Geography dimension tables describe location data, such as country, state, or city
- Product dimension tables describe products
- Employee dimension tables describe employees, such as sales people
- Range dimension tables describe ranges of time, dollar values, or other measurable quantities to simplify reporting
- Dimension tables are generally assigned a surrogate primary key, usually a single-column integer data type, mapped to the combination of dimension attributes that form the natural key.

Model płatka śniegu

In computing, a snowflake schema is a logical arrangement of tables in a multidimensional database such that the entity relationship diagram resembles a snowflake shape. The snowflake schema is represented by centralized fact tables which are connected to multiple dimensions.

The snowflake schema is similar to the star schema. However, in the snowflake schema, dimensions are normalized into multiple related tables, whereas the star schema's dimensions are denormalized with each dimension represented by a single table.

Star and snowflake schemas are most commonly found in dimensional data warehouses and data marts where speed of data retrieval is more important than the efficiency of data manipulations. As such, the tables in these schemas are not normalized much, and are frequently designed at a level of normalization short of third normal form.

3. Różnica między systemami hierarchicznymi a sieciowymi (codasyłowymi).

Hierarchiczne zabraniają cykli, możesz mieć jedynie strukturę drzewiastą pomiędzy rekordami. W sieciowych możesz mieć dowolne powiązania pomiędzy rekordami.

While the hierarchical database model structures data as a tree of records, with each record having one parent record and many children, the network model allows each record to have multiple parent and child records, forming a generalized graph structure. This property applies at two levels: the schema is a generalized graph of record types connected by relationship types (called "set types" in CODASYL), and the database itself is a generalized graph of record occurrences connected by relationships (CODASYL "sets"). Cycles are permitted at both levels.

The chief argument in favour of the network model, in comparison to the hierarchic model, was that it allowed a more natural modeling of relationships between entities. It was eventually displaced by the relational model, which offered a higher-level, more declarative interface.

4. Co to jest MOLAP, ROLAP, HOLAP?

Rodzaje OLAP (Online Analytical Processing).

MOLAP (multidimensional)

MOLAP = tradycyjne systemy OLAP, które przekładają transakcje na wielowymiarowe widoki. Dane są organizowane w postaci wielowymiarowych kostek. Kostka OLAP (OLAP cube) = struktura, przypomina wielowymiarowe arkusze kalkulacyjne. Może mieć więcej niż 3 wymiary (hypercube). Przykładowe wymiary: produkt, czas, miasto, rodzaj dochodu itp.

While both ROLAP and MOLAP analytic tools are designed to allow analysis of data through the use of a multidimensional data model, MOLAP differs significantly in that (in some software) it requires the pre-computation and storage of information in the cube — the operation known as processing. Most MOLAP solutions store these data in an optimized multidimensional array storage, rather than in a relational database (i.e. in ROLAP).

Zalety:

- 1) duża wydajność zapytań dzięki bardziej optymalnej technice przechowywania danych, wielowymiarowemu indeksowaniu oraz cache'owaniu
- 2) mniejszy rozmiar danych na dysku w porównaniu z rozwiązaniami relacyjnymi

3) automatyczne przetwarzanie agregatów wyższego rzędu

Wady:

- 1) problemy z wybieraniem danych o dużej liczbie wymiarów (np. kilka milionów elementów)
- 2) redundancja danych w niektórych systemach tego typu

ROLAP (relational)

Przechowują dane (często w postaci źródłowej) oraz tabele wymiarów w relacyjnych bazach danych. W tym rozwiązaniu również można wykorzystać dodatkowe tabele do przechowywania zagregowanych informacji. Wymaga to jednak większego nakładu na proces ETL.

While both ROLAP and MOLAP analytic tools are designed to allow analysis of data through the use of a multidimensional data model, ROLAP differs significantly in that it does not require the pre-computation and storage of information. Instead, ROLAP tools access the data in a relational database and generate SQL queries to calculate information at the appropriate level when an end user requests it. With ROLAP, it is possible to create additional database tables (summary tables or aggregations) which summarize the data at any desired combination of dimensions.

While ROLAP uses a relational database source, generally the database must be carefully designed for ROLAP use. A database which was designed for OLTP will not function well as a ROLAP database.

Zalety:

- 1) ROLAP is considered to be more scalable in handling large data volumes, especially models with dimensions with very high cardinality (i.e., millions of members).
- 2) The data are stored in a standard relational database and can be accessed by any SQL reporting tool (the tool does not have to be an OLAP tool).
- 3) By decoupling the data storage from the multi-dimensional model, it is possible to successfully model data that would not otherwise fit into a strict dimensional model.
- 4) The ROLAP approach can leverage database authorization controls such as row-level security, whereby the query results are filtered depending on preset criteria applied, for example, to a given user or group of users (SQL WHERE clause).

Wady:

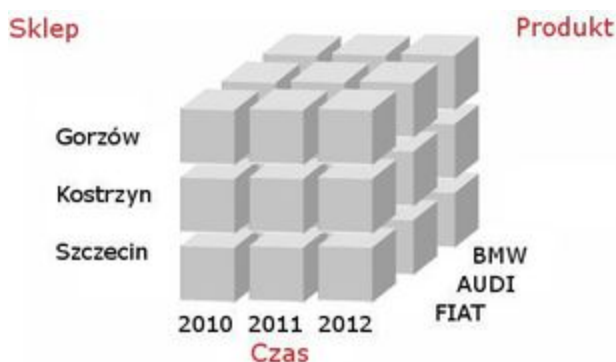
- 1) There is a consensus in the industry that ROLAP tools have slower performance than MOLAP tools. However, see the discussion below about ROLAP performance.
- 2) The loading of aggregate tables must be managed by custom ETL code. The ROLAP tools do not help with this task. This means additional development time and more code to support.

HOLAP (hybrid)

Wśród dostawców rozwiązań nie ma jednoznacznych ustaleń definicyjnych, określających co kwalifikuje się jako hybrydowy OLAP – poza tym, że baza danych rozdziela dane pomiędzy podsystem relacyjny i specjalizowany, łącząc tym samym cechy OLAP wielowymiarowego i relacyjnego.

Wykorzystane są tutaj jednocześnie wielowymiarowe (dane zagregowane) oraz relacyjne (dane elementarne, bez dublowania) bazy danych. Zapytania szczegółowe są wykonywane wolniej, ponieważ działają na strukturze relacyjnej (ROLAP). Dane w HOLAP mają zwykle mniejszy rozmiar niż dane w MOLAP. Struktury HOLAP są przeznaczone dla potrzeb szybkiego dostępu do agregacji bazujących na dużych zbiorach danych.

http://pl.wikipedia.org/wiki/Kostka_OLAP



5. Opisać ETL - co to jest, i jaki ma związek z hurtowniami?

ETL (Extract, Translate lub Transform, Load) - moduł pozyskiwania danych (np. do hurtowni), pobiera dane ze źródeł, przekształca je i ładuje do bazy (hurtowni)
http://en.wikipedia.org/wiki/Extract,_transform,_load

The load phase loads the data into the end target, usually the data warehouse. Depending on the requirements of the organization, this process varies widely. Some data warehouses may overwrite existing information with cumulative information; frequently, updating extracted data they add new data in a historical form at regular intervals—for example, hourly.

6. Wyjaśnić różnice pomiędzy realizacją architektury klient - serwer z grubym i cienkim klientem.

W cienkim kliencie jest tylko pobieranie i wyświetlanie informacji a w grubym jest także logika, co wyświetlić klientowi, jakie zapytania i gdzie wysłać itd.

(To review) http://pl.wikipedia.org/wiki/Cienki_klient

Cienki klient (ang. thin client) – komputer bądź specjalizowane urządzenie (terminal komputerowy) wraz z odpowiednim oprogramowaniem typu klient, umożliwiające obsługę aplikacji stworzonej w architekturze klient-serwer. Cechą szczególną cienkiego klienta jest niezależność od obsługiwanej aplikacji serwerowej (jej zmiana nie pociąga za sobą konieczności wymiany oprogramowania klienta).

7. Jakie zasadnicze problemy i w jaki sposób rozwiązuje technologia realizacji systemów typu Hurtownie Danych?

Hurtownia danych jest agregatem danych potrzebnych do dalszych analiz. Rozwiązuje problemy:

- rozproszenia danych na różnych nośnikach, środowiskach, ... - teraz wszystko jest w hurtowni
- różnej ziarnistości danych - przeliczenie danych
- różnej reprezentacji danych - przeliczenie danych
- niedostępności danych historycznych - w hurtowni przechowywane są wszystkie dane od początku
- przeprowadzanie obliczeń na serwerze, na którym działają użytkownicy systemu może im przeszkadzać (obciążenie itp.) - teraz analizujemy kopię, sporządzaną co jakiś czas

Ponadto hurtownia danych jest zoptymalizowana pod kątem wyszukiwania danych do analiz, nie wykonuje się tam transakcji.

,

rerf

8. Model z „grubym klientem” - wyjaśnić co to jest i do czego jest wykorzystywany?

Gruby klient posiada logikę przetwarzania danych. http://pl.wikipedia.org/wiki/Gruby_klient

Programy użytkowe wykonywane są bezpośrednio i autonomicznie na stacji, dokonującej przetwarzania danych oraz wymiany danych z użytkownikiem i innymi komputerami w sieci. Dane są przechowywane po stronie serwera. Przykładem oprogramowania typu gruby klient są aplikacje Swing napisane w języku Java a także kontrolki ActiveX w C/C++.

Architektura dwuwarstwowa (serwer/klient gruby):

Zalety: odciążenie serwera, zwiększenie skalowalności

Wady: konieczność instalacji aplikacji na wielu klientach, problem ze zmianą wersji aplikacji

9. Model pojęciowy/konceptualny

Model konceptualny występuje między światem rzeczywistym, a modelem logicznym i ma na celu zamodelowanie pojęć z rzeczywistości i lepsze zrozumienie problemu - modelujemy tu funkcje użytkowe (DFD, usecase'y) oraz dane (ERD, UML).

Generalnie model konceptualny ma być zrozumiały dla człowieka, a logiczny dla SZBD.

Konceptualne projektowanie bazy danych – to proces konstrukcji modelu danych, który jest niezależny od wszelkich aspektów fizycznych (specyficzny model danych, docelowy DBMS, programy użytkowe, języki programowania, platforma sprzętowa).

(Notatki Łukasza Szymańskiego)

Fragment świata rzeczywistego - byty/związki logiczne, zdarzenia => (analiza) =>

=> Model pojęciowy (konceptualny) - zdarzenia: DFD/use case, dane: ERD/diagramy klas =>

=> Model logiczny - projekt aplikacji (SQL), schemat BD; dostępność, uprawnienia, warunki integralności =>

=> Aplikacja

10. Wyjaśnij na czym polegają nowe zadania SZBD związane z wprowadzeniem koncepcji rozproszenia bazy danych i stosowania replikacji.

Two Phase Commit - http://en.wikipedia.org/wiki/Two-phase_commit_protocol

In a "normal execution" of any single distributed transaction, i.e., when no failure occurs, which is typically the most frequent situation, the protocol consists of two phases:

- 1) The commit-request phase (or voting phase), in which a coordinator process attempts to prepare all the transaction's participating processes (named participants, cohorts, or workers) to take the necessary steps for either committing or aborting the transaction and to vote, either "Yes": commit (if the transaction participant's local portion execution has ended properly), or "No": abort (if a problem has been detected with the local portion), and
- 2) The commit phase, in which, based on voting of the cohorts, the coordinator decides whether to commit (only if all have voted "Yes") or abort the transaction (otherwise), and notifies the result to all the cohorts. The cohorts then follow with the needed actions (commit or abort) with their local transactional resources (also called recoverable resources; e.g., database data) and their respective portions in the transaction's other output (if applicable).

Replication involves using specialized software that looks for changes in the distributive database. Once the changes have been identified, the replication process makes all the databases look the same. The replication process can be complex and time-consuming depending on the size and number of the distributed databases. This process can also require a lot of time and computer resources.

Duplication, on the other hand, has less complexity. It basically identifies one database as a master and then duplicates that database. The duplication process is normally done at a set time after hours. This is to ensure that each distributed location has the same data. In the duplication process, users may change only the master database. This ensures that local data will not be overwritten.

http://en.wikipedia.org/wiki/Distributed_database

<http://wazniak.mimuw.edu.pl/images/2/26/ZSBD-2st-1.2-w01.tresc-1.1.pdf>

Jeszcze inaczej o replikacji

If at any time one master replica is designated to process all the requests, then we are talking about the primary-backup scheme (master-slave scheme) predominant in high-availability clusters - *prawdopodobnie chodzi o to, co zostało nazwane duplikacją powyżej*

On the other side, if any replica processes a request and then distributes a new state, then this is a multi-primary scheme (called multi-master in the database field). In the multi-primary scheme, some form of distributed concurrency control must be used, such as distributed lock manager - *prawdopodobnie chodzi o to, co zostało nazwane replikacją powyżej*

Rodzaje/tryby replikacji

Migawkowa (snapshot)

Dane rozprowadzane są w stanie z pewnego określonego momentu. Taki rodzaj replikacji znajduje głównie zastosowanie przy danych, które nie są często modyfikowane. Jednak takie modyfikacje mogą być znaczne. Wszelkie zmiany pomiędzy migawkami nie są monitorowane.

Transakcyjna/przyrostowa - ma sens przy master-slave

Dane rozprowadzane są na podstawie logów transakcji. Umożliwia zachowanie zasady ACID, ponieważ dane zmieniane są tylko na głównym serwerze.

Dwukierunkowa/łącząca (merge) - ma sens przy multi-master

Dwukierunkowe rozprowadzanie danych, zarówno od serwera, jak i od klientów, które mogły być również przeprowadzone bez połączenia pomiędzy serwerami. W czasie synchronizacji może dojść do konfliktu, który musi być rozwiązany przez osobę przeprowadzającą aktualizację.

11. Wyjaśnij pojęcie niezgodności impedancji, która może wystąpić w aplikacjach systemów baz danych.

Niezgodność impedancji - zbiór problemów występujących przy łączeniu relacyjnych baz danych z aplikacjami obiektowymi, m. in. przy mapowaniu obiektów na encje i relacje między nimi.

<http://edu.pjwstk.edu.pl/wyklady/jps/scb/wyklad2/section1.html>

Składnia

Programista musi w jednym tekście programu używać dwóch stylów językowych i przestrzegać reguł dwóch różnych gramatyk.

System typów

Język zapytań operuje na typach zdefiniowanych w schemacie bazy danych, natomiast język programowania posiada zwykle odmienny system typów.

Semantyka i paradygmaty języków

Język zapytań bazuje na stylu deklaratywnym, podczas gdy języki programowania bazują na stylu imperatywnym/obektowym/funkcyjnym itp.

Fazy i mechanizmy wiązania

Języki zapytań są oparte o późne wiązanie (interpretowane), podczas gdy wiele języków programowania często zakłada wczesne wiązanie (podczas kompilacji i konsolidacji).

Rozwiązanie: Criteria w Hibernate albo Slick w Scali - można pisać zapytania w całości w obiektowym języku programowania, zapytania są sprawdzane na etapie kompilacji.

Traktowanie wartości zerowych (null)

Rozwiązanie: np. Option w Scali itp.

Schematy iteracyjne

W języku zapytań iteracje są wtopione w semantykę operatorów takich jak selekcja, projekcja i złączenie. W języku programowania iteracje muszą być organizowane explicite przy pomocy pętli for, while, repeat lub innych.

Rozwiązanie: przetwarzanie wyników zapytań przy pomocy języka programowania wymaga specjalnych udogodnień takich jak kursory i iteratory.

Traktowanie cechy trwałości danych

Języki zapytań przetwarzają wyłącznie trwałe dane (znajdujące się na dysku), podczas gdy języki programowania przetwarzają wyłącznie dane nietrwałe znajdujące się w pamięci operacyjnej.

12. Podaj i scharakteryzuj przykłady zastosowania specjalnych systemów baz danych.

NoSQL - skalowanie horyzontalne = ang. scale out (!= scale up - skalowanie wertykalne):

1) sharding (partycjonowanie poziome) - horizontal partitioning is a database design principle whereby rows of a database table are held separately, rather than being split into columns (which is what normalization and vertical partitioning do, to differing extents). Each partition forms part of a **shard**, which may in turn be located on a separate database server or physical location.

2) partycjonowanie pionowe

3) repliki do odczytu

SQL/MM - SQL/MM is the effort to standardize extension for multi-media and application-specific packages in SQL. SQL, as defined in [ISO99] is extended to manage data like text, images, spatial data, or to perform data mining. The standard is grouped into several parts.

GIS - designed to capture, store, manipulate, analyze, manage, and present all types of geographical data

Systemy 3D - ? http://en.wikipedia.org/wiki/Spatial_database

A **spatial database** is a database that is optimized to store and query data that represents objects defined in a geometric space. Most spatial databases allow representing simple geometric objects such as points, lines and polygons. Some spatial databases handle more complex structures such as 3D objects, topological coverages, linear networks

Długie teksty - ?

Bazy strumieniowe

W modelu strumieni danych zakłada się, że niektóre lub wszystkie napływające do systemu dane nie są dostępne w dowolnej chwili. Możliwy czas, w jakim można je zarejestrować jest skończony. Dane te pojawiają się w źródle danych i przyjmują postać strumienia danych.

Bazy temporalne

Jest odmianą bazy relacyjnej, w której każdy rekord posiada stempel czasowy, określający czas w jakim wartość jest prawdziwa. Posiada także operatory algebry relacyjnej, które pozwalają operować na danych temporalnych (wyciągać historię).

13. Opisz różnice między ERD a schematem bazy danych.

ERD jest schematem na poziomie modelu pojęciowego (konceptualnego) i odwzorowuje fragment rzeczywistości w postaci zrozumiałej dla człowieka. Natomiast schemat bazy danych definiujemy już na poziomie modelu logicznego i musi być on dostosowany do konkretnej implementacji bazy przez SZBD (skutkuje to między innymi innym modelowaniem związków wiele do wielu).

Given an ER diagram, we can look for a relational schema that closely approximates the ER design. The translation is approximate because it is not always feasible to capture all the constraints in the ER design within the relational schema. (In SQL, certain types of constraint, for example, are inefficient to enforce, and so usually not implemented.)

W skrócie: tabele to prostokąty, kolumny to kółka (i chyba też właściwości relacji), relacje to romby; mogą być strzałki przy relacjach (1:N/N:N itp.)

14. Opisz model obiektowy.

http://en.wikipedia.org/wiki/Object_database

Z notatek Łukasza:

- warunki integralności utrzymane w aplikacji
- obiekty w bazie identyfikujemy za pomocą ID

Because the database is integrated with the programming language, the programmer can maintain consistency within one environment, in that both the OODBMS and the programming language will use the same model of representation. The programming language and the database schema use the same type definitions. Relational DBMS projects, by way of contrast, maintain a clearer division between the database model and the application.

Most object databases also offer some kind of query language, allowing objects to be found using a declarative programming approach. An attempt at standardization was made by the ODMG with the Object Query Language, OQL.

Access to data can be faster because joins are often not needed (as in a tabular implementation of a relational database). This is because an object can be retrieved directly without a search, by following pointers.

The efficiency of such a database is also greatly improved in areas which demand massive amounts of data about one item. For example, a banking institution could get the user's

account information and provide them efficiently with extensive information such as transactions, account information entries etc.

An object database stores complex data and relationships between data directly, without mapping to relational rows and columns, and this makes them suitable for applications dealing with very complex data.

15. 2009/2010 Grupa A pytanie 2: Co to jest Mainframe wymień wady i zalety tego rozwiązania.

http://en.wikipedia.org/wiki/Mainframe_computer ?

Różnica pomiędzy superkomputerem i mainframe nie jest zbyt oczywista. Można powiedzieć, że superkomputery stosuje się do rozwiązywania problemów które wymagają dużej szybkości obliczeń, podczas gdy mainframe'y stosuje się, gdy potrzebna jest wysoka wydajność I/O, niezawodność oraz jednoczesna obsługa różnorodnych procesów biznesowych (ang. mixed workload).

http://docs7.chomikuj.pl/1341616015,PL,0,0,podstaw_tech_inf.pdf

Zalety: Centralne zarządzanie, Wysoki poziom bezpieczeństwa

Wady: Brak standardów systemów operacyjnych i współpracy w środowisku wielu dostawców, Potencjalny pojedynczy punkt uszkodzenia (konfiguracja nieodporna na uszkodzenia)

fa16. 2009/2010 Grupa B pytanie 2: Jaki był powód wprowadzenia Kostek Wielowymiarowych? Opisz ich podstawowe składniki.

http://pl.wikipedia.org/wiki/Kostka_OLAP

17. (O bezpieczeństwie SZBD - z notatek wrzuconych na wiki)

Warunki bezpieczeństwa:

- dostępność dla użytkowników – usługi i dane mają być zawsze dostępne
- ochrona przed nieupoważnionym dostępem – zarówno do danych, jak i do funkcji
- integralność, spójność, pewność i poprawność danych – czyli pełne zaufanie do danych

Z punktu widzenia funkcjonalnego te 3 warstwy bezpieczeństwa wystarczą. m

4 klasy bezpieczeństwa (A, B, C, D).

D – na systemy tej klasy nie stawia się w zasadzie żadnych wymogów bezpieczeństwa.

C – kwestia wyjmowalności dysku, dostęp dopiero po rozkręceniu itp. – PCty są tej klasy.

A1 – od momentu powstania konceptu architektury (!) tego komputera wszystkie dane (dokumentacje, rysunki, schematy) go dotyczące objęte są tajemnicą państwową, a każdy do nich dostęp jest rejestrowany.

Bezpieczeństwo w warstwie systemu operacyjnego

Niedopuszczenie obcej aplikacji do nieupoważnionego dostępu do plików fizycznych BD, każda aplikacja powinna mieć dostęp pośrednio, za udziałem SZBD

Bezpieczeństwo w warstwie SZBD

- zarządzanie transakcjami, w szczególności tymi złożonymi i rozproszonymi. Po ponownym uruchomieniu, jeśli SZBD stwierdzi, że ma niezakończone transakcje, to ma je albo poprawnie zakończyć albo też cofnąć do stanu sprzed uruchomienia transakcji
- zarządzanie wielodostępem – ochrona systemu przed zakleszczeniem spowodowanym wielodostępem i niepoprawnością danych
- warunki integralnościowe danych – ochrona przed brakiem integralności (dopuszczalne wartości, triggery, funkcje składowe); funkcje integralności: o integralność dziedziny (poprawność typów), o integralność tabeli – klucz główny – unikalny, żadna jego część nie może być NULLem o integralność referencyjna – klucz obcy – musi odwoływać się do istniejącego elementu, może być NULLem
- kontrola dostępu – zdefiniowanie upoważnień poszczególnych użytkowników

Bezpieczeństwo w warstwie aplikacji

Aplikacja odpowiada za wszystkie elementy bezpieczeństwa, których nie da się zrealizować na niższych warstwach.

18. (12 postulatów Codda - j.w.)

12 postulatów Codda – dotyczą wszystkich aspektów pomysłów realizacji SZBD, definiują, jakie warunki musi spełniać relacyjny SZBD. Najważniejsze z nich to:

- rozbicie na model logiczny i fizyczny (użytkownik definiuje model logiczny i na nim operuje np. za pomocą SQL; model fizyczny jest w pełni zarządzany przez SZBD i nie jest dostępny z zewnątrz za pomocą innej aplikacji; dla użytkownika BD fizyczny element BD jest całkowicie przezroczysty;
- w BD nie przechowujemy tylko danych, ale też związki między danymi (do których też możemy się odwoływać, a więc istnieją również fizycznie);
- dobrze by było, gdyby definicja przechowywana była w z góry zdefiniowanej strukturze BD, zarządzana tym samym SZBD i obsługiwana za pomocą tego samego języka.

0.

System musi być kwalifikowany jako relacyjny, jako baza danych i jako system zarządzania

1.

Postulat informacyjny - dane są reprezentowane jedynie poprzez wartości atrybutów w wierszach tabel,

2.

Postulat dostępu - każda wartość w bazie danych jest dostępna poprzez podanie nazwy tabeli, atrybutu oraz wartości klucza podstawowego.

3.

Postulat dotyczący wartości NULL - dostępna jest specjalna wartość NULL dla reprezentacji wartości nieokreślonej jak i nieadekwatnej, inna od wszystkich i podlegająca przetwarzaniu

4.

Postulat dotyczący katalogu - wymaga się, aby system obsługiwał wbudowany katalog relacyjny z bieżącym dostępem dla uprawnionych użytkowników używających języka zapytań,

5.

Postulat języka danych - system musi dostarczać pełnego języka przetwarzania danych, który może być używany w trybie interaktywnym jak i w obrębie programów aplikacyjnych, obsługuje operacje definiowania danych, operacje manipulowania danymi, ograniczenia związane z bezpieczeństwem i integralnością oraz operacje zarządzania transakcjami,

6.

Postulat modyfikowalności perspektyw - system musi umożliwiać modyfikowanie perspektyw, o ile jest ono (modyfikowanie) semantycznie realizowalne,

7.

Postulat modyfikowalności danych - system musi umożliwiać operacje modyfikacji danych, musi obsługiwać operatory INSERT, UPDATE oraz DELETE,

8.

Postulat fizycznej niezależności danych - zmiany fizycznej reprezentacji danych i organizacji dostępu nie wpływają na aplikacje,

9.

Postulat logicznej niezależności danych - zmiany wartości w tabelach nie wpływają na aplikacje,

10.

Postulat niezależności więzów spójności - więzy spójności są definiowane w bazie i nie zależą od aplikacji,

11.

Postulat niezależności dystrybucyjnej - działanie aplikacji nie zależy od modyfikacji i dystrybucji bazy,

12.

Postulat bezpieczeństwa względem operacji niskiego poziomu - operacje niskiego poziomu nie mogą naruszać modelu relacyjnego i więzów spójności.

19. Hierarchiczny model bazy danych

A hierarchical database model is a data model in which the data is organized into a tree-like structure. The structure allows representing information using parent/child relationships: each

parent can have many children, but each child has only one parent (also known as a 1-to-many relationship). All attributes of a specific record are listed under an entity type.

The hierarchical structure is used primarily today for storing geographic information and file systems. Currently the most widely used hierarchical databases are IMS and Windows Registry by Microsoft.

20. RAID

RAID (ang. *Redundant Array of Independent Disks*, Nadmiarowa macierz niezależnych dysków) – polega na współpracy dwóch lub więcej dysków twardych w taki sposób, aby zapewnić dodatkowe możliwości, nieosiągalne przy użyciu jednego dysku jak i kilku dysków podłączonych jako oddzielne.

RAID używa się w następujących celach:

- zwiększenie niezawodności (odporność na awarie),
- zwiększenie wydajności transmisji danych,
- powiększenie przestrzeni dostępnej jako jedna całość.

RAID 1 - mirroring

RAID 5 - odporność na awarię 1ego dysku

RAID 6 - odporność na awarię 2óch dysków

21. Poziomy izolacji

| Isolation level | Dirty reads | Non-repeatable reads | Phantoms |
|------------------|-------------|----------------------|-----------|
| Read Uncommitted | may occur | may occur | may occur |
| Read Committed | - | may occur | may occur |
| Repeatable Read | - | - | may occur |
| Serializable | - | - | - |

A *dirty read* (aka *uncommitted dependency*) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.

A *non-repeatable read* occurs, when during the course of a transaction, a row is retrieved twice and the values within the row differ between reads.

A *phantom read* occurs when, in the course of a transaction, two identical queries are executed, and the collection of rows returned by the second query is different from the first.

This can occur when *range locks* are not acquired on performing a *SELECT ... WHERE* operation.

The *phantom reads* anomaly is a special case of *Non-repeatable reads* when Transaction 1 repeats a ranged *SELECT ... WHERE* query and, between both operations, Transaction 2 creates (i.e. *INSERT*) new rows (in the target table) which fulfill that *WHERE* clause.

Rok akad. 2012/2013

Grupa A

1. Przedstawić i opisać elementy charakterystyczne dla obiektowych baz danych

[Pytanie 14.](#)

2. Coś z modułem ETL

3. ACID jest to...

ACID – zbiór właściwości, które gwarantują poprawne przetwarzanie *transakcji* w *bazach danych*. **ACID** jest skrótem od angielskich słów: **atomicity** – *atomowość*, **consistency** – *spójność*, **isolation** – *izolacja*, **durability** – *trwałość*.

<http://pl.wikipedia.org/wiki/ACID>

Atomicity - każda transakcja albo wykonana się w całości, albo w ogóle

Consistency (spójność) - po wykonaniu transakcji nie zostaną naruszone zasady integralności (constraints, triggers, cascades itp.)

Isolation - jeśli 2 transakcje wykonują się współbieżnie, to nie widzą nawzajem wprowadzanych przez siebie zmian

Durability (trwałość) - system potrafi udostępnić dane zapisane w ramach zatwierdzonych transakcji (np. po awarii zasilania)

4. Największą wadą „obsługi” BLOBów przez SZBD jest...

[Notatki Łukasza](#)

Ich użycie spowalnia funkcjonowanie relacyjnej bazy danych. DBMS może takie obiekty tylko wstawiać i usuwać z bazy. Korzystanie z bloba jest utrudnione – można je tylko ładować i pobierać, kwestia obsługi takiej danej przechodzi do aplikacji (baza danych nie potrafi nic z tym zrobić, bo nie zna jej typu) – dopiero na poziomie aplikacji wiadomo, co można zrobić z taką daną.

5. d/r

6. Podział baz danych typu NoSQL

- Column family stores (These were created to store and process very large amounts of data distributed over many machines. There are still keys but they point to multiple columns. The columns are arranged by column family.)
- Document Databases
- Graph Databases
- Key-values stores

<http://rebelic.nl/2011/05/28/the-four-categories-of-nosql-databases/>

Grupa B

1. Podać i opisać charakterystyczne cechy systemów typu Hurtownia Danych.
7. Podać i scharakteryzować sposoby realizacji kostki OLAP
8. W jaki sposób można rozwiązać problem niezgodności impedancji w systemach relacyjnych baz danych?
9. Replikację stosuje się, aby...
10. Co to jest ePUAP?

Elektroniczna Platforma Usług Administracji Publicznej

Ogólnopolska platforma teleinformatyczna służąca do komunikacji obywateli z jednostkami administracji publicznej w ujednolicony, standardowy sposób.

11. Coś o tworzeniu modelu koncepcyjnego / pojęciowego dla systemów obiektowych (nie pamiętam dokładnie).

Grupa D

1. Jakie mechanizmy wykorzystuje się do tworzenia i przechowywania replik?

Pytanie 10.

2. Czym różni się i do czego służy model logiczny Gwiazdy od Płatka Śniegu?

Pytanie 2.

3. Czego dotyczy prawo Brewera?

CAP theorem

<http://www.julianbrowne.com/article/viewer/brewers-cap-theorem>

Żaden system rozproszony nie może naraz spełnić wszystkich trzech z poniższych warunków (max 2):

Consistency (spójność) = wszystkie węzły systemu widzą te same dane jednocześnie

A service that is consistent operates fully or not at all. Gilbert and Lynch use the word "atomic" instead of consistent in their proof, which makes more sense technically because, strictly speaking, consistent is the C in ACID as applied to the ideal properties of database transactions and means that data will never be persisted that breaks certain pre-set constraints.

Availability (dostępność)

Availability most often deserts you when you need it most - sites tend to go down at busy periods precisely because they are busy.

Partition tolerance

A partition happens when, say, a network cable gets chopped, and Node A can no longer communicate with Node B.

4. Do czego służy normalizacja relacyjnej bazy danych?

Normalizacja bazy danych jest to proces mający na celu eliminację powtarzających się danych w relacyjnej bazie danych. Główna idea polega na trzymaniu danych w jednym miejscu, a w razie potrzeby linkowania do danych. Taki sposób tworzenia bazy danych zwiększa bezpieczeństwo danych i zmniejsza ryzyko powstania niespójności

Normalizacja bazy danych do konkretnej postaci może wymagać rozbicia dużych tabel na mniejsze i przy każdym wykonywaniu zapytania do bazy danych ponownego ich łączenia. Zmniejsza to wydajność, więc w niektórych przypadkach świadoma denormalizacja (stan bez normalizacji) jest lepsza - zwłaszcza w systemach niekorzystających z modelu relacyjnego (np. OLAP).

5. Jakie są ew. minusy stosowania SBD na mainframe'ach?

?

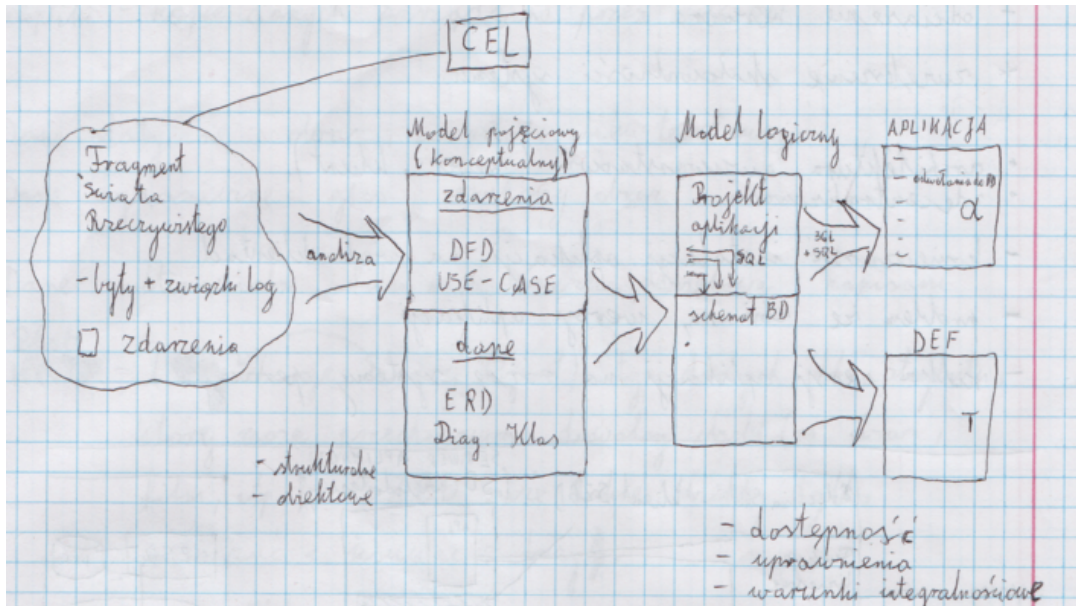
6. Data mart vs data warehouse - czym się różnią (przy czym nazewnictwo było po polsku)

Data Warehouse - is a [database](#) used for [reporting](#) and [data analysis](#):

- Holds multiple subject areas
- Holds very detailed information
- Works to integrate all data sources
- Does not necessarily use a dimensional model but feeds dimensional models.

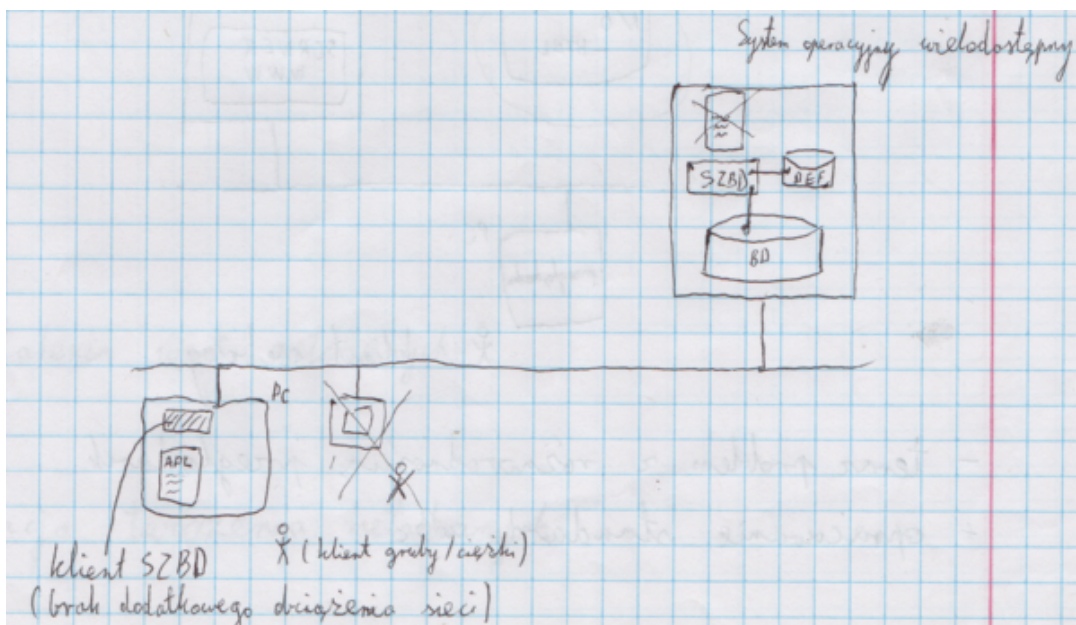
Data Mart - is the access layer of the [data warehouse](#) environment that is used to get data out to the users:

- Often holds only one subject area - for example, Finance, or Sales
- May hold more summarized data (although many hold full detail)
- Concentrates on integrating information from a given subject area or set of source systems
- Is built focused on a dimensional model using a star schema.

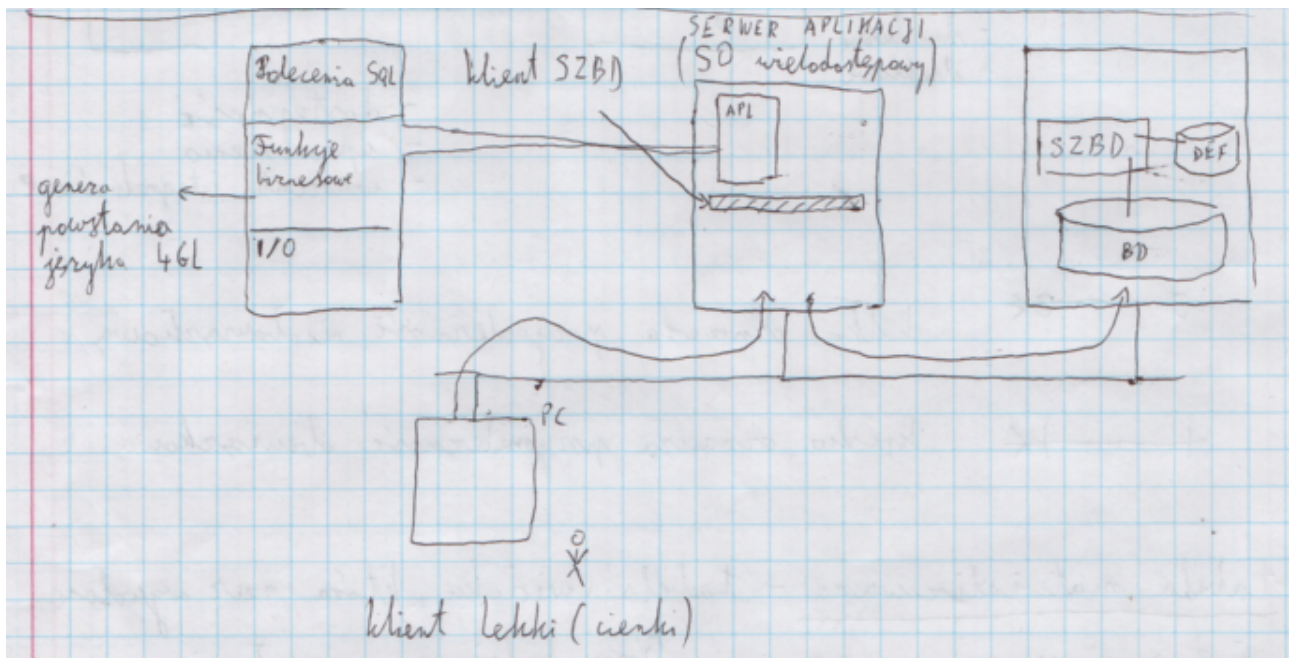


- tabela materializowana – tabela widoku, która raz wywołana jest wczytywana z pamięci

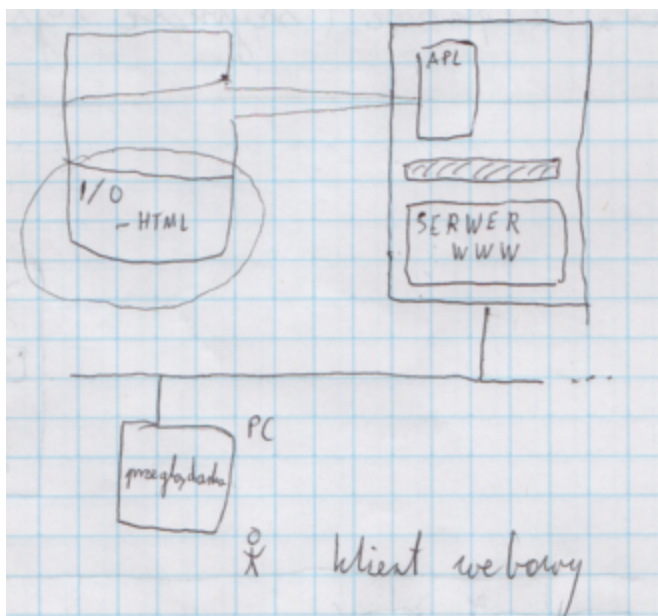
O co w tym obrazku chodzi? - zamiana klienta cienkiego na grubego:



- + obciążenie serwera
- + zwiększenie skalowalności serwera
- architektura dwuwarstwowa (klient, serwer)
- zdecentralizowana
- konieczność instalacji aplikacji na wielu klientach
- problem ze zmianą wersji aplikacji
- kilka wersji aplikacji na różne systemy operacyjne



O co tu chodzi?



- teraz problem z różnorodnością przeglądarek
- + opracowanie standardu W3C

Repliki – kopie danych zarządzane przez SZBD

Dane statyczne – obraz, zawartość pliku tekstowego

Dane dynamiczne – głos, ruchomy obraz

Dane multimedialne - film ze ścieżką dźwiękową

BLOB – (Binary Large Object) typ danych dla atrybutu, który może przechowywać dowolny obiekt (np. obraz, film, itp.) SZBD może takie obiekty tylko wstawiać i usuwać z bazy.

Bazy danych działają wolniej przez obiekty BLOB. Aby przyspieszyć np. wyszukiwanie zdjęć, osób w okularach, trzeba dodać dodatkowy atrybut „czy_ma_okulary”.

Metadane semantyczne – odnoszą się bezpośrednio do zawartości BLOB (do treści samego zapisu)

Metadane sygnałowe – atrybuty opisujące techniczne realia tego zapisu (np. format pliku)

Metadane zewnętrzne - Informacje nieodnoszące się bezpośrednio do zawartości medium, tylko do sposobu jego tworzenia, przechowywania, udostępniania, klasyfikacji (Np. reżyser filmu, nazwa pliku źródłowego, typ kompresji pliku, tytuł zdjęcia, czas i miejsce wykonania zdjęcia)

http://www.cs.put.poznan.pl/mwojciechowski/slides/mbd/mbd_short.pdf

Koszty:

- pamięć
- problem ustalenia wartości metadanych

Rozwiązanie:

- automatyzacja tworzenia metadanych

Obiektowe bazy danych:

- warunki integralności utrzymane w aplikacji
- obiekty w bazie identyfikujemy za pomocą ID