wszystko po trochu:
https://www.dropbox.com/sh/qw2qcu8wrkwrtxd/vojh4lmtRy/Multimedialne_Bazy_danych_niestacjonarne.ppt

23. Rodzaje klauzul i mechanizmy sprawdzenia spójności

Źródło: notatki z DBD, książka z której korzystał Wilczek na wykładach „Deductive Database and their applications" oraz http://csc.csudh.edu/suchenek/CSC321/_NEGATIVE_INFO_DDB.pdf

Klauzule pozwalają wyrazić wypowiedzi bardziej złożone niż literały, zachowując względną prostotę, tak pożądaną przez potencjalnych użytkowników dedukcyjnych baz danych. Używając ich, można reprezentować zależności i ograniczenia różnych rodzajów.

Program w języku Prolog jest zbiorem klauzul Horna. **Klauzula Horna** to klauzula z co najwyżej jednym literałem pozytywnym (bez negacji). Literał pozytywny nazywamy głowa klauzuli, a literały negatywne tworzą ciało klauzuli.

Typ klauzuli	**Przykład**
Klauzula Określona	Sub - component(X, Z) component(X, Y) ∧ component(Y, Z)
Klauzula Nieokreślona	father(X, Y) ∨ Mother(X, Y) ← parent(X, Y)
Klauzula Pozytywna	supplies(X, P1) ∨ supplies(X, P2)
Klauzula Ustalona (Ground Clause)	father(Gary, Tom) ←Son(Tom, Gary)
Klauzula Pusta	□

Klauzule składają się z: **głowy** i **ciała** (zawierającego pod cele). Mogą być zapisywane jako implikacja z pozytywnymi literałami w poprzedniku (głowie) i negatywnymi w następniku (ciele). Dzielą się na **określone** (bez negacji - klauzula zawierająca dokładnie jeden literał pozytywny i dowolną liczbę literałów negatywnych) i **nieokreślone** (z negacją- klauzula o co najmniej dwóch literałach pozytywnych).

Klauzula ustalona to taka klauzula, w której nie występują zmienne.

Klauzula pozytywna to taka klauzula, która składa się z samych pozytywnych literałów.

Pusta klauzula zawiera nie literały, oznaczane jako □. Klauzula ta reprezentuje sprzeczność.

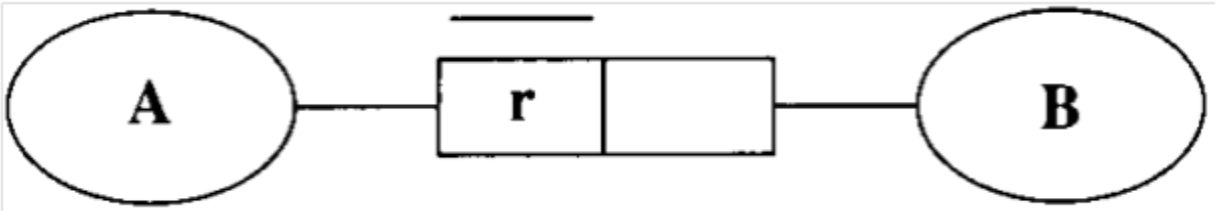
Zbiór klauzul o tym samym predykacie w głowie to **procedura**.
Pełną postać klauzuli nazywa się **regułą**.
Klauzula bez literałów negatywnych to **fakt – klauzula pozytywna**.
Klauzula bez literałów pozytywnych to **zapytanie**.

WIĘZY INTEGRALNOŚCI

Ograniczenia integralności to opis dozwolonych stanów baz danych. Jeżeli szczególna transakcja *T* mapuje szczególny stan *s* w *S* poza *S*, to *T* nie jest dopuszczona do bazy danych w stanie *s*. Ograniczenia integralności są często określane w fazie koncepcyjnej modelowania rozwoju systemów informacyjnych, a niektóre rodzaje ograniczeń można modelować za pomocą metody modelowania systemów informatycznych. Szczególnie interesujące dla nas jest to, że większość ograniczeń integralności napotkanych w praktyce może być wyrażona jako formuły w pierwszej kolejności rachunku predykatów z agregacji. Ograniczenia integralności, mogą być modelowane za pomocą koncepcyjnej metody modelowania NIAM.

Przykłady ograniczeń integralności:

- **Ograniczenie unikatowości**

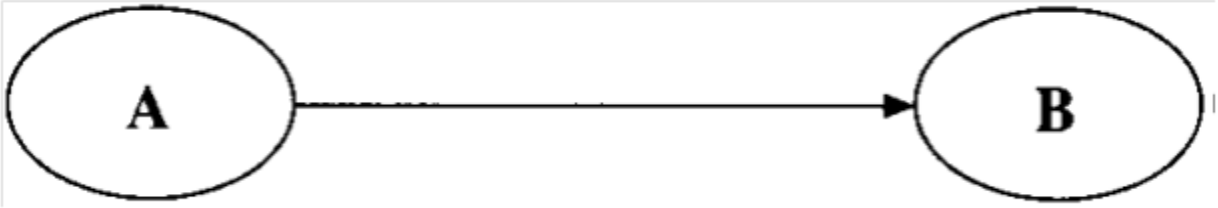


Ograniczenie to oznacza, że każda jednostka w populacji *A* uczestniczy w relacji *r* z co najwyżej jedną jednostką z populacji *B*.

- **Ograniczenie obowiązku** (obowiązkowe – mandatory constraint)

Oznacza, że każda jednostka w populacji *A* uczestniczy w relacji *r* z co najmniej jedną jednostką z populacji *B*.

- **Ograniczenie podtypu**



Oznacza, że każdy członek populacji *A* jest także członkiem populacji *B*.

Istnieje wiele rodzajów ograniczeń, które nie są zwykle wyrażone w diagramach NIAM. Ponadto, ograniczenia integralności mogą być wyrażone w predykatach IDB oraz predykatach EDB, np. ograniczenie mówiące, że graf musi być acykliczny. Jest także możliwość wyrażenia ograniczenia integralności jako formuły rachunku predykatów pierwszego rzędu – klauzula Horna. Przez limity stosowane wobec ograniczeń integralności do wykrywania naruszeń, rezygnując z części mocy logiki pierwszego rzędu, można wyrazić je jako klauzule Horna. Interesuje nas, w przypadku braku naruszenia, deklaracja dwóch specjalnych predykatów, *zły (Bad)* i *nie_naruszony (no_violations)*. Predykat *zły* jest prawdziwy, gdy przynajmniej jedno z ograniczeń integralności jest naruszone, *nie_naruszony* jest definiowany jako: no_violations :- not bad.

24. Mechanizmy wnioskowania dla klauzul określonych

Źródło: notatki z DBD, książka z której korzystał Wilczek na wykładach „Deductive Databases and their applications" oraz http://csc.csudh.edu/suchenek/CSC321/_NEGATIVE_INFO_DDB.pdf

Klauzula określona (bez negacji) – klauzula zawierająca dokładnie jeden literał pozytywny i dowolną liczbę literałów negatywnych.

Klauzule określone tworzą określone dedukcyjne bazy danych, w których wszystkie odpowiedzi na zapytania są określone, co powoduje, że proces ich obliczania jest prosty. Istnieją dwa podejścia szacowania zapytań w określonej dedukcyjnej bazie danych: **top-down** – podejście używające SLD rezolucji i **bottom-up** – używający algebry relacyjnej.

Top-down – metoda konkurencyjna dla podejścia bottom-up, w metodzie tej nie przegląda się modelu doskonałego, jest to metoda wyboru (wnioskowanie poprzez SLD rezolucję).
Rezolucja SLD – Selection- rule-driven Linear resolution for Definite clauses – polega na upraszczaniu zapytania krok po kroku, aż dotrzemy do TRUE. W każdym kroku literał z zapytania i głowa reguły są unifikowane (podstawienie, które doprowadza do stwierdzenia, że dwa elementy są tożsame). Następnie głowa reguły jest zastępowana jej ciałem (fakty – reguły bez ciała).
W top-down, dla zapytania, buduje się jeden raz drzewo SLD. Następuje przeglądanie modeli – porównywanie wartości stałych (matching). Istnieje tutaj problem unifikacji: co krok wartościujemy, co krok przeglądamy ekstensjonalną bazę danych (EDB). W ramach top-down wyróżniamy dwa algorytmy: depth-first i breadth-first. **Depth-first** jest silnikiem wnioskowania Prologu, gdzie przeszukuje się drzewo SLD w głąb. Jego wadą jest wrażliwość na uporządkowanie podceli w IDB. Niewłaściwa kolejność podceli może prowadzić do nieskończonych pętli, choć rozwiązanie istnieje i jest skończone. Algorytm **breadth-first** polega na przeszukiwania drzewa wszerz. Jest on podobny do algorytmu bottom-up.

Bottom-up – metodę tę dzielimy ze względu na algorytm na: podejście naiwne i semi-naiwne. Podejście **naiwne** polega na wyznaczaniu modelu doskonałego. Następnie z modelu doskonałego, poprzez nawigację wyznaczane są zapytania. Większość zapytań zwraca jeden wiersz – istnieje jedno rozwiązanie. Istnieją zapytania, które weryfikują stwierdzenia, zwracają: True lub False. **Algorytm naiwny** – pewna transformacja ponad ekstensjonalną bazą danych wyznaczająca views zmaterializowany (lepiej wykonywać zapytania na viewsie zmaterializowanym). Viewsy zmaterializowane stosuje się do otrzymania modelu perfekcyjnego. Materializacja viewsa polega na obliczeniu formuł, faktów z intensjonalnej bazy danych (IDB). Przyrost faktów następuje do momentu, aż nie będziemy mieli co dodawać, wtedy zostaje osiągnięty stały punkt. Osiągnięty zostaje model perfekcyjny. W przeciwieństwie do top-down tutaj przegląda się model tylko raz. W celu zredukowania obliczeń powstał **algorytm semi-naiwny** podejścia bottom-up. Tutaj realizujemy przekształcenie materializujące wyniki, przepisujemy fakty do wyników. Następnie wartościujemy reguły z IDB , których wszystkie podcele są faktami. Następnie te reguły, które oprócz faktów mają coś innego w podlecach, o ile pojawiły się w poprzedniej iteracji. Iterujemy, aż przyrost będzie zerowy, czyli nie będziemy mogli już dodać nic nowego do wyników.
Podejście bottom-up jest lepsze, gdy wymagany jest cały model perfekcyjny. Algorytm semi-naiwny jest podstawą skutecznego obliczenia większości kwerend, gdzie wymagany jest zestaw odpowiedzi. Z drugiej strony top-down jest lepszy, gdy wymagana jest pojedyncza krotka.

25. Mechanizmy wnioskowania dla klauzul nieokreślonych

Źródło: notatki z DBD, książka z której korzystał Wilczek na wykładach „Deductive Database and their applications" oraz http://csc.csudh.edu/suchenek/CSC321/_NEGATIVE_INFO_DDB.pdf

Klauzula nieokreślona (z negacją) – klauzula o co najmniej dwóch literałach pozytywnych.

Problem z negacją jest następujący: wyniki zastosowania algorytmów naiwnego i semi-naiwnego (bottom-up) do zapytań zawierających w podcelach negację, nie są w pełni zadawalające. Kiedy w intensjonalnej bazie danych znajduje się formuła zawierająca negację głowy innej reguły, to istnieje zasada mówiąca, że negacje oblicza się przed jej użyciem. Niestety nie jest to możliwe dla całej intensjonalnej bazy danych. Mówi się, że pogram jest **stratyfikowalny**, kiedy powyższa reguła jest spełniona na dla całej intensjonalnej bazy danych. W datalogowym programie ze stratyfikacją negacji, model obliczany na podstawie powyższej reguły jest modelem doskonałym. Istnieje prosty algorytm determinujący czy IDB jest stratyfikowalna. Używa on grafu, którego węzły są regułami IDB. Posiada łuk biegnący od reguły źródłowej do reguły docelowej, jeśli reguła źródłowa jest częścią zdefiniowanego predykatu, który jest podcelem reguły docelowej. Łuk jest nazywany pozytywnym, gdy podcel reguły docelowej jest pozytywny, i negatywnym, gdy podcel reguły docelowej jest negatywny. Taki graf nazywa się **grafem zależności** IDB.

Twierdzenie stratyfikacji: program jest stratyfikowalny, jeżeli jego graf zależności nie zawiera cyklów z negatywnym łukiem.

Przy obliczeniach wyniku dla IBD, w której znajdują się klauzule nieokreślone najpierw dzielimy program na warstwy, aby określić w jakiej kolejności obliczać reguły. Mechanizmy wnioskowania są takie same jak dla klauzul określonych. – tak myślę wysłałam maila do Wilczka czy rzeczywiście tak jest bo w dostępnych materiałach tego wprost nie napisali i taki był mój indywidualny wniosek. Poczekam aż Wilczek mi odpisze i wtedy w razie jak to jest źle to poprawię

Opracowanie jest ok, Wilczek napisał:
Tak, ale...
proszę pamiętać, że metody top-down mają wyłącznie szczególne zastosowania w przypadku Datalogu, pochodzą z metod stosowanych w programowaniu w logice (Prolog), gdzie dowodzimy spełnienia twierdzeń.
W Datalogu interesuja nas wszystkie możliwe zastąpienia zmiennych, które uzyskujemy metodami bottom-up.
Oprócz negacji problemem w metodzie top-down depth-first może być rekurencja.
Dodatkowe materiały na ten temat z prośba o przekazanie pozostałym osobom:
http://www.cs.uni-paderborn.de/fileadmin/Informatik/AG-Boettcher/Lehre/WS_07_08/dbis1/stanford/dbis1k4-ddb-JU-datalog_stratified-negation.pdf

26. Zasady zamkniętego świata w dedukcyjnych bazach danych

źródło: notatki z DBD

Dedukcyjną bazą danych określamy system bazy danych zintegrowany z deklaratywnym językiem programowania w logice. Deklaratywną bazę danych dzielimy na :

· EDB (ekstensjonalna/jawna baza danych) – najczęściej relacyjna BD, przechowuje definicje relacji/predykatów (krotki relacji).

· IBD (intensjonalna baza danych) – program w logice – odpowiednik widoków z modelu relacyjnego.

Interpretator programu PROLOG, często używanego w dedukcyjnych BD, przeprowadza wnioskowanie zgodnie z poniższymi punktami:

1. Próba odnalezienia klauzuli w programie, której głowa może być unifikowana z zapytaniem. Jeżeli brak takiej klauzuli to zwracamy ‘false’.
2. Jeżeli zapytanie to klauzula jednostkowa (bez zmiennych) to zwracamy ‘true’ i wynikowe podstawienie.
3. Powtarzamy próbę unifikacji dla wszystkich podceli (predykatów ciała naszego zapytania), zwracamy ‘true’, jeżeli uda się spełnić wszystkie podlece, ‘false’ w przeciwnym przypadku.

PROLOG, w przeciwieństwie do DATALOGA, nie poszukuje wszystkich możliwych podstawień, bo realizacja programu polega na dowodzeniu twierdzenia. Problem pojawia się, gdy w wyrażeniach mamy negację. Aby powiedzieć, że nie istnieje x musimy zbadać wszystkie elementy. Stąd pojawia się pojęcie zamkniętego świata (closed word assumption), czyli założenie, że w programie mamy wszystkie niezbędne dane do wyliczenia programu. Jeżeli w ekstensjonalnej bazie danych brakuje faktu X, to oznacza, że X nie istnieje.
NOT jest operatorem modalnym (realizowanym na predykatach). Negacja zmienia sposób wyliczania programu. W przypadku negacji kolejność predykatów ma znaczenie. W DATALOGu wprowadza się pojęcie 'stratyfikacji', Stratyfikacja programu P to taki podział jego reguł IBD w zbiory P1, P2, … takie, że w i-tej warstwie występują jakiekolwiek zaprzeczone podcele to wystąpiły one (w postaci niezaprzeczonej) najpóźniej w warstwie P(i-1). Przy stratyfikacji wyliczamy modele doskonałe poszczególnych warstw, modelem wynikowym jest suma modeli poszczególnych warstw.

27. Modele bezpieczeństwa danych

Modele bezpieczeństwa danych: model macierzowy, model Wooda, model Bella-LaPaduli, model Biby, Model Diona, Model Sea View, Model Jajodii-Sandhu, Model Smitha-Winsletta, Model bezpieczeństwa dla przepływu danych (z opisu kursu Bezpieczeństwo Baz Danych);

od BINSI (maszter):
http://nob.cs.ucdavis.edu/book/book-intro/slides/04.pdf
http://nob.cs.ucdavis.edu/book/book-intro/slides/05.pdf
http://nob.cs.ucdavis.edu/book/book-intro/slides/06.pdf

Formalne modele bezpieczeństwa informacji w systemach komputerowych

Model formalny – model matematyczny opisujący stan początkowy systemu, sposób przejścia do innego stanu oraz definicję „bezpiecznego" stanu systemu.

Model może obejmować następujące mechanizmy ochrony:

- uwierzytelnienie użytkownika,
- szyfrowanie danych,
- kontrolę dostępu,
- kontrolę przepływu informacji,
- kontrolę możliwości wnioskowania,
- audyt.

¨ Krata jako model zarządzania prawami dostępu do informacji,
¨ Model Bell-La Padula dostępu do informacji
¨ Modele ochrony informacji w bazach danych

§ Model Wooda
§ Model Sea View

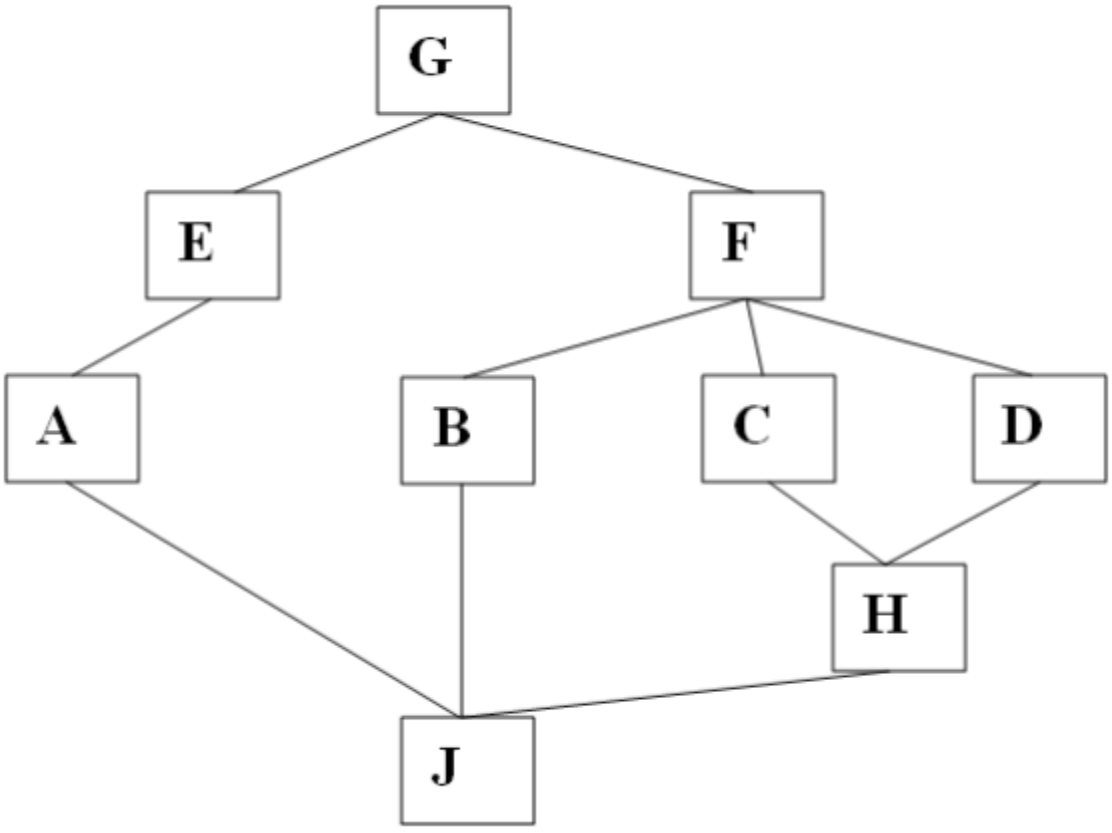
¨ Modele teoretycznych ograniczeń systemów bezpieczeństwa

§ Model Grahama- Denninga
§ Model Harrisona-Ruzzo-Ullmana (HRU)
§ Model „take-grant”

Ad. krata:

uogólniony model zarządzania prawami dostępu do informacji zgodnie z zasadami *wojskowej polityki bezpieczeństwa.* Każda informacja (plik) musi być przyporządkowana do jakiegoś poziomu tajności oraz muszą być dla niej określone prawa dostępu opisane znaną regułą że *informacja jest ujawniana tylko tym osobom, które jej potrzebują do wykonania swojej pracy.* Te prawa dostępu wynikają z kategorii tajności określających zakres informacji w ramach pewnej tematyki.

Krata jest strukturą matematyczną której elementy są uporządkowane za pomocą operatora relacji częściowego porządku ³ nazywanego też operatorem dominacji. W kracie nie wszystkie elementy muszą być porównywalne. Każda para posiada posiada kres górny oraz kres dolny.



Ad.Bell-La Padul

Umożliwia on kontrolę dostępu na podstawie tzw. macierzy dostępu, ponadto umożliwia kontrolę przepływu informacji. Jest to implementacja modelu kratowego. Macierz dostępu zawiera wiersze reprezentujące użytkowników i kolumny reprezentujące obiekty. Macierz zawiera podzbiory zbioru dopuszczalnych w systemie *praw dostępu*, np. odczyt, zapis, itp. Na

przecięciu wiersza *j* i kolumny *k* znajduje się zbiór praw dostępu, jakie posiada użytkownik **j** względem obiektu **k**.

Bieżący stan modelowanego systemu opisuje trójka:

(S, O, A)

gdzie: **S** – jest zbiorem użytkowników, **O** – zbiorem obiektów, **A** – macierzą dostępu.

Mechanizm autoryzacji sprawdza, czy dostęp **r** który użytkownik **s** próbuje zastosować do obiektu **o** należy do podzbioru praw dostępu określonego **s** i **o** w macierzy **A**, czyli czy na przecięciu kolumny odpowiadającej **o** i wiersza odpowiadającego **s**, znajduje się taki zbiór $\mathbf{R_{so}}$, że $\mathbf{r \hat{I} R_{so}}$

Każdy użytkownik posiada przypisany poziom ochrony określony przez parę:

(C, G)

gdzie: C – jest klasyfikacją poziomów tajności (poufne, tajne,...),

G – podzbiorem kategorii tajności.

Zbiór podzbiorów tajności jest uporządkowany liniowo ze względu na relację ³.

Przepływ danych sterowany jest przez poniższe reguły:

1. Każdemu użytkownikowi przypisany jest maksymalny poziom ochrony MPO.
2. Użytkownik nie może czytać danych z obiektu, gdy POO>MPO, gdzie POO jest poziomem ochrony obiektu.
3. Użytkownik o bieżącym poziomie ochrony Ln może zapisywać dane tylko do tych obiektów, których poziom ochrony POO spełnia warunek POO³Ln, a czytać tylko z tych obiektów, dla których POO<=Ln. Operacja *odczyt-i-zapis* jest dozwolona w przypadku, gdy Ln=POO.
4. Poziomy chrony obiektów nie mogą być zmienane przez użytkowników – poziomy ochrony nadawane są odgórnie, np. przez administratora.
5. Każdy dostęp do obiektu musi być autoryzowany poprzez mechanizmy polityki dyskrecjonalnej.
6. Obiekty nie posiadające nadanego poziomu ochrony (nie występujące w hierarchii) nie są dostępne.
7. Jeżeli obiekt jest aktywowany to nadawany mu poziom ochrony jest niezależny od poziomu ochrony, który posiadał wcześniej (o ile posiadał taki przed dezaktywacją).

28. Bezpieczeństwo w SQL-owych systemach baz danych

Dane przechowywane w bazach danych mogą być narażone na wiele rodzajów zagrożeń. Do najważniejszych zaliczamy:

- nielegalny odczyt danych przez nieuprawnionych użytkowników
- niepoprawne operacje modyfikacji danych, które mogą być skutkiem:
- umyślnego działania nieuprawnionych użytkowników
- przypadkowych omyłek użytkowników
- braku właściwej kontroli przy współbieżnym dostępie do danych wielu użytkowników

- błędów oprogramowania lub awarii systemów komputerowych
- zniszczenie danych w przypadku poważnych awarii sprzętu komputerowego.

Środki ochrony baz danych rozpatruje się zazwyczaj w następujących obszarach:

- Autentyfikacja i autoryzacja użytkowników, określane też jako kontrola dostępu. Kontrola dostępu realizowana jest według określonych reguł, nazywanych polityką bezpieczeństwa.
- Ochrona integralności danych. Mechanizmy zapewniające integralność baz danych można podzielić na mechanizmy integralności semantycznej oraz mechanizmy integralności transakcyjnej. Semantyczne więzy integralności definiują poprawny stan baz danych pomiędzy kolejnymi operacjami na bazie, stąd też umożliwiają ochronę, w pewnym zakresie, przed niepoprawną(umyślną lub przypadkową) modyfikacją danych. Mechanizmy integralności transakcyjnej chronią spójność bazy danych w warunkach współbieżnie realizowanych operacji na bazie przez wielu użytkowników, a także w przypadku błędów oprogramowania i awarii sprzętowych. Zapewnienie ochrony integralności należy do klasycznych zadań każdego systemu zarządzania bazą danych i nie będzie rozpatrywane w niniejszej pracy.
- Monitorowanie operacji na bazie danych. Wszystkie działania użytkowników istotne z punktu widzenia bezpieczeństwa systemu winny być monitorowane i rejestrowane. Analiza wyników takiej rejestracji może służyć do oceny poprawności przyjętej polityki bezpieczeństwa.
- Szyfrowanie zawartości bazy danych.

W systemach zarządzania bazami danych (w skrócie nazywanych dalej serwerami baz danych) stosowane są dwa różne podejścia w zakresie polityki bezpieczeństwa:

- Uznaniowy model bezpieczeństwa (ang. discretionary security). W modelu tym dla każdego podmiotu (użytkownika) definiuje się reguły dostępu określające uprawnienia podmiotu do obiektów systemu. Ważną właściwością modelu uznaniowego jest możliwość nadawania przez określonego użytkownika posiadanych przez niego uprawnień innym użytkownikom.
- Obowiązkowy model bezpieczeństwa (ang. mandatory security). Model ten wymusza obowiązkową kontrolę dostępu do danych, realizowaną według ścisłej, hierarchicznej klasyfikacji podmiotów (użytkowników) i obiektów bazy danych. Każdemu obiektowi przypisuje się pewien poziom tajności, zaś podmiotowi – przepustkę. Porównanie rodzaju przepustki z poziomem tajności określa możliwości odczytu bądź zapisu danych. W komercyjnych systemach baz danych jest powszechnie implementowany uznaniowy model bezpieczeństwa.

Każdy użytkownik zdefiniowany w systemie zarządzania bazą danych musi mieć określony identyfikator (nazwa użytkownika/konto) oraz charakterystyczne cechy tożsamości Autentykacja użytkownika rozpoczyna się od jego identyfikacji (login), a następnie obejmuje ona weryfikację jego tożsamości (hasło, podpis cyfrowy). Autoryzacja obejmuje nadawanie użytkownikom uprawnień, które są zwykle dzielone na dwie kategorie:

1. uprawnienia systemowe, o charakterze ogólnym, odnoszące się do całej bazy danych (wszystkich jej obiektów),
2. uprawnienie szczegółowe (obiektowe), definiujące prawo do wykonywania określonej operacji na określonym obiekcie bazy danych.

Nadawanie uprawnień systemowych jest różnie realizowane w różnych Systemach Zarządzania Bazami Danych:

- W niektórych systemach (np. Oracle, SQLServer) określony jest duży zbiór uprawnień systemowych (kilkadziesiąt) nadawanych indywidualnie lub grupowanych w tzw. role.
- W innych systemach (np. Centura, Informix) użytkownicy dzieleni są na klasy, którym są przyporządkowane różne zakresy uprawnień (DBA – wszelkie prawa w bazie danych, RESOURCE – prawo tworzenia tablic i dysponowania nimi, CONNECT – tylko możliwość korzystania z przyznanych uprawnień). Nadawanie uprawnień obiektowych wiąże się z przyznaniem danemu użytkownikowi prawa wykonywania określonej operacji języka SQL (SELECT, INSERT, DELETE, INDEX, ALTER, UPDATE) na określonym obiekcie (tablicy lub perspektywie). Jest ono realizowane podobnie we wszystkich wymienionych systemach baz danych (polecenie GRANT). Specyficznym aspektem rozważanych zagadnień jest umożliwienie propagacji uprawnień, tzn. przekazanie użytkownikowi prawa do przekazania nadanych mu uprawnień innym użytkownikom (opcja WITH GRANT OPTION). Wykorzystanie tej opcji może w sposób mało kontrolowany rozszerzać grono użytkowników operujących na zasobach bazy danych. W serwerach baz danych wykorzystujących prezentowany model bezpieczeństwa stosowane są zwykle dodatkowe mechanizmy kontroli i zabezpieczeń.

Perspektywy

Do szczególnych obiektów, objętych nadawaniem uprawnień, należą perspektywy (VIEWS). Mechanizm perspektyw pozwala tworzyć wirtualne tablice udostępniające użytkownikowi fragmenty zasobów jednej lub wielu tablic rzeczywistych, w formie prostej lub przetworzonej. Przy wykorzystaniu tego mechanizmu mogą być przyznawane użytkownikom prawa dostępu nie do tablic rzeczywistych, a tylko do perspektyw. Pozwala to na wprowadzenie różnorodnych ograniczeń, w tym np. udostępnienie tylko wybranych kolumn bądź też wybranych wierszy tablic.

Limity zasobów – profile użytkowników

Poza ograniczeniami dostępu do danych (wynikającymi z przyznanych uprawnień) na użytkowników mogą być nakładane ograniczenia dotyczące zasobów systemowych, kontrolowanych przez System Zarządzania Bazą Danych. Należą do nich np.:

- ilość czasu procesora przeznaczonego na obsługę zadań określonego użytkownika,
- liczba równoczesnych sesji otwartych przez użytkownika,
- liczba odczytów (logicznych) z dysku przez użytkownika,
- dopuszczalny czas bez wykonywania operacji na bazie danych.

Zestaw nałożonych ograniczeń tworzy tzw. profil użytkownika.

Monitorowanie bazy danych

Z uznaniową kontrolą dostępu powiązana jest dostępna w niektórych serwerach baz danych możliwość tzw. audytu, czyli obserwacji i rejestrowania informacji o działaniach użytkowników. Taka funkcja, którą określamy jako monitorowanie, może np. obejmować:

- monitorowanie operacji: śledzenie wskazanych operacji (instrukcji) języka SQL (wykonanych lub odrzuconych),
- monitorowanie uprawnień: śledzenie wykorzystania uprawnień systemowych przez wskazanych użytkowników,
- monitorowanie obiektów: śledzenie wykonania wskazanych operacji (instrukcji) SQL na wskazanych obiektach. Monitorowanie może być wykorzystane do podniesienia bezpieczeństwa systemu, przede wszystkim poprzez kontrolę działań użytkowników, którzy próbują przekraczać przyznane im uprawnienia. Poza tym monitorowanie jest wykorzystywane do strojenia serwera bazy danych.

Szyfrowanie w bazach danych

Dodatkowym mechanizmem zabezpieczeń dla baz danych może być szyfrowanie. Szyfrowanie może dotyczyć różnych informacji związanych z bazą danych. Najczęściej stosowanym zabiegiem jest szyfrowanie haseł użytkowników. Hasła przechowywane na dysku w postaci jawnej mogą się bowiem stać dość łatwym łupem włamywaczy penetrujących przestrzeń dyskową. Z podobnych powodów szyfruje się w pierwszej kolejności niektóre obiekty bazy danych takie jak procedury pamiętane, funkcje itp. Ewentualna nieuprawniona ingerencja w ich treść mogłaby bowiem doprowadzić do uszkodzenia zawartości bazy danych. Ostatnim krokiem w stosowaniu omawianych mechanizmów jest szyfrowanie danych. Może ono dotyczyć szyfrowania zawartości plików jako całości, bądź też szyfrowanie poszczególnych rekordów. Celem szyfrowania danych jest przede wszystkim zabezpieczenie przed odczytaniem danych przy nieuprawnionym dostępie do fizycznych struktur danych (np. z poziomu systemu operacyjnego lub nawet poza nim, np. kradzież dysku).

Opracowanie numer 2 - trochę krótkie

I. **Definiowanie typów kolumn** – np. decimal, character itd… w celu zapewnienia spójności danych

II. **Zarządzanie transakcjami** - podstawowym zadaniem systemu zarządzania transakcjami jest takie ich wykonywanie, które zapewni spójność (poprawność, niesprzeczność danych) bazy danych. Transakcja w SQL posiada cechy ASOT, poprawne zakończenie transakcji kończy się operacją COMMIT (gdy się wykona poprawnie) i ROLLBACK (gdy się nie powiedzie). Standard SQL zawiera środki do definiowania warunków spójności, których naruszenie skutkuje niepowodzeniem operacji, która w/w warunek próbuje naruszyć. Są to:

A. **UNIQUE** – Każda wartość w kolumnie musi być unikalna
B. **PRIMARY KEY** – specjalny przypadek UNIQUE. Każda wartość w kolumnie musi być unikalna, wartość nie może być pusta (równa NULL)
C. **FOREIGN KEY** – służy do zdefiniowania referencyjnych warunków spójności
D. **CHECK** – służy do określenia czy kolumna/zestaw kolumn (z jednej/wielu tabel) mają mieć wartości spełniające określony warunek

III. **Tworzenie widoków** – jeden z mechanizmów służących zapewnieniu poufności danych, służy do ukrywania informacji przed nieupoważnionymi użytkownikami. Wybieramy kolumny z jednej lub kilku tabel i uzyskujemy możliwość udostępnienia ich określonej klasie użytkowników. Tworzone za pomocą operacji CREATE VIEW. Można użyć podczas jego tworzenia operacji WITH CHECK OPTION, która definiuje warunki modyfikacji widoku.
IV. **Nadawanie uprawnień** – możemy przyznawać (operacja GRANT) jak i odbierać (operacja REVOKE) oraz (DENY zabraniać jakieś operacji) uprawnienia w SQL bazach danych. Każdy użytkownik aby wykonać określoną operacje (CRUD) musi posiadać do niej uprawnienia. Użytkownicy mogą sobie przekazywać uprawnienia (włącznie z opcją przekazywania uprawnień), autor tabeli - posiada wszystkie prawa do tabeli jak i obiektów zdefiniowanych w jej obrębie. Uprawnienia możemy nadawać w przypadku UPDATE oraz INSERT do kolumny/ tabeli. Ponadto istnieje funkcja USAGE, która uprawnia do użycia m. in. określonej dziedziny, zbioru znaków, uporządkowania.
V. **Identyfikator autoryzacji** – podczas łączenia się do serwera bazy danych, podawany jest jawnie lub pośrednio np. za pomocą funkcji CURRENT_USER identyfikator autoryzacji.

Ponadto na przykładzie MS SQL Server mogą być zdefiniowane dodatkowe mechanizmy bezpieczeństwa jak:

- **Uwierzytelnienie użytkowników –** możemy się uwierzytelnić za pomocą trybu uwierzytelnienia systemu Windows jak i również trybu mieszanego, w którym dopuszczona jest możliwość uwierzytelnienia za pomocą loginu i hasła przechowywanego w bazie MS SQL Server
- **Szyfrowanie** – możemy włączyć szyfrowanie definicji widoków, procedur pamiętanych, wyzwalaczy.
- **Wykorzystanie protokołu SSL** - możliwość tworzenia bezpiecznego połączenia między MS SQL Server, a aplikacją kliencką za pomocą protokołu SSL (Secure Sockets Layer) http://www.greensql.com/content/10-must-do-sql-server-2012-security-tasks
- **Plus to co pojawiło się wyżej…**

29. Bezpieczeństwo statystycznych baz danych

źródło: materiały, niestety papierowe, które dał mi Liber - miałam taki temat na projekcie

Statystyczna baza danych zawiera informacje o pojedynczych obiektach, ale pozwala tylko na wykonywanie zapytań sumarycznych przy założeniu, że jedno zapytanie dotyczy dużego zbioru wierszy. Nie zabezpiecza to jednak w ogólności przed sięganiem przy użyciu zapytań sumarycznych do informacji o pojedynczych obiektach. Potrzebne są dodatkowe zabezpieczenia uniemożliwiające tego typu wnioskowania - np. ograniczenie liczby zapytań, jakie może skierować do bazy danych jeden użytkownik.
Stan informacyjny statystycznej bazy danych składa się z dwóch elementów:
- danych (przechowywanych w bazie danych)
- wiedzy zewnętrznej.
Wiedza zewnętrzna to wiedza użytkowników o bazie danych i dzieli się na :

- wiedzę roboczą : wiedza o atrybutach występujących w bazie danych i o rodzajach dostępnych statystyk
- wiedzę dodatkową : obejmuje informacje, które zwykle nie są udostępniane przez bazę danych. Informacje te mogą być poufne (data urodzenia) lub jawne (płeć).

Rodzaje statystyk udostępnianych przez statystyczne bazy danych:
- ilość (count)
- suma (sum)
- średnia (avg)
- minimum (min)
- maksimum (max)

Statystyka jest wrażliwa, gdy ujawnia zbyt dużo poufnych informacji o pewnych osobach, organizacjach, firmach itp. Statystyka obliczana na podstawie poufnych informacji ze zbioru odpowiedzi o liczności 1 jest zawsze wrażliwa. Statystyka obliczana ze zbioru odpowiedzi o liczności 2 też jest uznawana za wrażliwą, bo użytkownik posiadając pewną wiedzę dodatkową może wydedukować pojedynczą wartość.

Mówimy, że baza zapewnia poufność doskonałą, gdy nie ujawnia żadnych statystyk wrażliwych (stan nie do osiągnięcia w rzeczywistych systemach, aby zapewnić poufność doskonałą należy nie ujawniać żadnych informacji J).

Istnieją mechanizmy zabezpieczania statystycznych baz danych. Najprostszy atak na statystyczną bazę danych to taki, który używa małe i duże zbiory odpowiedzi. Jeżeli atakujący ma pewną wiedzę na temat cechy, która może identyfikować daną osobę i zapytanie o liczbę elementów posiadających tą cechę zwraca 1, to jednoznacznie wyznaczyliśmy osobę. Można teraz pytać o cechę, którą osoba ma plus dodatkową cechę, którą chcemy zbadać. Np. „liczba osób, które mieszkają w Wołowie i mają między 40 a 45 lat” à odpowiedź systemu 1. „liczba osób, które mieszkają w Wołowie i mają między 40 a 45 lat i pensję większą niż 8000” i mamy odpowiedź, czy nasza osoba ma pensję większą niż 8000. Podobnie można zadawać pytania o bardzo duże zbiory odpowiedzi, używając dopełnień (dopełnienie bardzo dużego zbioru jest bardzo małe :-)). Aby uchronić się przed takimi atakami nie można udostępniać statystyk bazujących na bardzo małych lub bardzo dużych zbiorach odpowiedzi. Czyli statystyka C jest dozwolona, gdy jest wyliczana dla liczby elementów większych niż n i mniejszych niż N-n, gdzie N to cały możliwy zbiór odpowiedzi, a n to wybrana liczba całkowita.

Bazy danych, które umożliwiają użytkownikom wstawianie i usuwanie rekordów są narażone na dodatkowe ataki – ograniczenie rozmiaru zbioru odpowiedzi można ograniczyć za pomocą wstawiania nowych danych. Zabezpieczeniem w tym przypadku jest zablokowanie dodawania/ usuwania rekordów dla użytkowników statystycznych.

Kolejnym mechanizmem zabezpieczającym statystyczne bazy danych to sterowanie maksymalnym rzędem statystki, gdzie przez rząd statystyki rozumie się liczbę atrybutów, na podstawie których liczy się statystykę (po informatycznemu: liczba składowych w klauzuli WHERE). Sterowanie maksymalnym rzędem statystyki ogranicza użycie statystyk używających zbyt wielu atrybutów. Może to być bardzo ograniczające, bo często statystyki wysokich rzędów są bezpieczne, a będą niedozwolone.

Bardziej skuteczne są mechanizmy wprowadzające szumy.

· Zniekształcanie odpowiedzi polega na zmienia statystyki przed jej udostępnieniem zgodnie z przyjętą funkcją zniekształcania (np. zaokrąglenie do najbliższej wielokrotności pewnej podstawy b).
· Losowanie zbioru odpowiedzi uniemożliwia intruzowi precyzyjne sterowanie zawartością zbiorów odpowiedzi. Metoda używa 80 – 90% pierwotnego zbioru rekordów, każdorazowo losując inny podzbiór, tworzący próbę, z której obliczana jest statystyka.
· Zniekształcanie danych polega na chwilowym zniekształceniu danych w momencie obliczania statystyki poprzez zmianę każdej wartości używanej do obliczenia statystyki, z godnie z przyjętą funkcją zniekształcania. Zniekształcenie odbywa się w taki sposób, żeby odzwierciedlały statystyczną rzeczywistość, ale nie podawały dokładnych danych.

30. Metodyka tworzenia bezpiecznych baz danych

http://www.roz6.woiz.polsl.pl/bazydanych/pliki/BD5.pdf ?

Wersja 1 (tylko na podstawie rozdziału w źródle [3])
Wersja 2 trochę w inny sposób, źródła są podane.

I. ***Analiza wstępna*** – celem tej fazy jest zbadanie wykonywalności systemów zabezpieczeń (ocena ryzyka, koszty projektu oraz systemu, określają jakie aplikacje muszą zostać zaprojektowane oraz nadanie im priorytetów). Wynikiem tej analizy jest lista zagrożeń wraz z priorytetami oraz analiza stosowalności i spójności produktów komercyjnych z istniejącymi mechanizmami (czy użyć obecnych czy tworzyć własne od podstaw). Ponadto określa ona wykonalność techniczno-ekonomiczną. W tym celu rozważa się:

A. **Ryzyko związane z systemem** – określa się najpoważniejsze zagrożenia dla bazy danych oraz wynikające z nich straty za pomocą analizy ryzyka jak np. nieautoryzowany odczyt i modyfikacje (zarówno poprzez dostęp do komputera z bazą danych jak i również do schematów samej bazy danych)
B. **Cechy środowiska bazy danych** – cechy te wpływają na wymagania dotyczące ochrony i na związane z nimi mechanizmy (czy wykorzystać wielopoziomowe systemy zabezpieczeń).
C. **Stosowalność istniejących produktów** – rozważa się wady i zalety dostępnych produktów komercyjnych oraz celowość budowy systemu zabezpieczeń od podstaw.
D. **Spójność produktów –** sprawdza się możliwość zintegrowania mechanizmów zabezpieczeń z właściwymi mechanizmami sprzętowymi i programowymi.
E. **Wydajność powstałego produktu** – porównuje się wydajność systemy z zabezpieczeniami jak i bez nich.

II. ***Analiza bezpieczeństwa i wybór polityki bezpieczeństwa*** – każda z baz danych ma inne potrzeby dotyczące ochrony przed różnymi zagrożeniami. Definiujemy w tej fazie tryby dostępów podmiotów do obiektów bazy. Wynikiem tej fazy jest zestaw zdań określający wymagania dotyczące bezpieczeństwa jak np. Autoryzacja jest scentralizowana, Dostęp użytkowników to transakcji musi być kontrolowany itd.… Wymaganie dotyczące bezpieczeństwa można przedstawić jako czwórkę (podmiot, akcja, obiekt, warunek). System

może być sklasyfikowany jak system małego lub dużego ryzyka na podstawie poniższych elementów:

A. **Korelacja** – występuje gdy dane są ze sobą powiązane i zmiana jednej wpływa na drugą. (lub gdy odczyt jednych powoduje ujawnienie wartości drugiej)
B. **Współdzielenie** – występuje gdy dana jest używana przez wielu użytkowników/aplikacji. W bazie danych powoduje to problem ze współbieżnością (jednoczesny dostęp do tych samych danych). Współdzielenie wiąże się z poufnością i spójnością danych. Najlepiej tworzyć systemy o danych w dużym stopniu rozdzielonych i procesach zdecentralizowanych.
C. **Dostęp do danych** – określamy kto i do jakich danych może uzyskać dostęp oraz w jakim celu. Bierzemy pod uwagę prywatność, bezpieczeństwo, poufność, implikacje prawne oraz uwierzytelnienie. Nadajemy tak uprawnienia aby móc kontrolować poczynania użytkowników (dzienniki zdarzeń). Najbardziej zagrożone są systemy czasu rzeczywistego.
D. **Fachowość użytkowników** – wprawni użytkownicy gwarantują lepszą pewność działania systemu, jednakże często to oni nadużywają swoich uprawnień (istotna jest solidność i lojalność personelu)
E. **Sprzęt i oprogramowanie** – wykorzystanie certyfikowanego oprogramowania i sprzętu od certyfikowanych dostawców
F. **Analiza wrażliwości** – ustala się wrażliwość danych i aplikacji. Im większa kontrola tym większa wrażliwość danych.
G. **Identyfikacja zagrożeń** – identyfikacja: zagrożeń, trybów pracy aplikacji, technik penetracji.
H. **Analiza odporności** – identyfikacja słabych punktów systemu w oparciu o wcześniej zidentyfikowane zagrożenia.
I. **Analiza ryzyka** – porównywanie naruszeń bezpieczeństwa systemu z możliwymi zagrożeniami, słabościami systemu i trybami penetracji.
J. **Oszacowanie ryzyka** – szacuje się prawdopodobieństwo wystąpienia zdarzeń oraz reakcje systemu na nie.
K. **Definiowanie wymagań** – na podstawie zagrożeń i niepożądanych zdarzeń oraz prawdopodobieństwie ich zajścia definiowane są wymagania dotyczące bezpieczeństwa.

III. ***Wybór polityki bezpieczeństwa*** – celem polityki bezpieczeństwa jest zdefiniowanie uwierzytelnionego dostępu do obiektów w systemie dla każdego z podmiotów. Wybierane są strategie najlepiej pasujące do zdefiniowanych wymagań. Kryteria doboru strategii są następujące:

A. **Poufność a spójność i niezawodność** – szukamy równowagi pomiędzy tymi cechami.
B. **Maksymalizacja współdziałania a minimalizacja uprawnień** – przyznajemy dostęp do informacji ściśle związanej z realizowanymi przez podmioty zadaniami.
C. **Granulacja kontroli** – odnosi się do kontroli względem kontrolowanych podmiotów i obiektów.
D. **Atrybuty używane przy kontroli dostępu** – decyzje dotyczące ochrony bazują na atrybutach podmiotu/obiektów i/lub kontekście żądania obiektu.
E. Spójność
F. **Priorytety** – stosujemy priorytety jeśli pojawiają się konflikty między regułami wyrażającymi strategie bezpieczeństwa.

G. **Uprawnienia** – definiujemy uprawnienia domyślne jak i w jaki sposób mogą one być modyfikowane dla użytkowników/aplikacje.
H. **Role** – definiujemy typy ról (np. właściciel, administrator) określających uprawnienia oraz odpowiedzialność w ramach systemu.
I. **Dziedziczenie** – propagacja uprawnień dostępu z obiektów na ich kopie.

IV. ***Projektowanie konceptualne*** – w tej fazie następuje formalizacja koncepcji dotyczących wymagań i polityki bezpieczeństwa z poprzedniej fazy, a wynikiem jest konceptualny model bezpieczeństwa. W modelu tym:

A. Bezpieczeństwo bazy danych jest niezależne od szczegółów implementacyjnych
B. Przeprowadza się testy kompletności i poprawności projektu (czy zestaw wymagań jest minimalny, kompletny, i nie zawiera w sobie konfliktów)
[3: str 332 jest przykład]

V. ***Projektowanie logiczne*** – w tej fazie przekształcamy model konceptualny na model logiczny obsługiwany przez konkretny SZBD. Np. w relacyjnych bazach danych mamy komendy GRANT/REVOKE/DENY itd…. W modelu tym wykorzystujemy mechanizmy zabezpieczeń systemów operacyjnych, dodatkowych pakietów zabezpieczeń. Jeśli wymagania jakiekolwiek z modelu konceptualnego nie mogą być zapewnione przy wykorzystaniu dostępnych mechanizmów, to muszą one zostać utworzone od podstaw przez projektantów.
VI. ***Implementacja mechanizmów bezpieczeństwa*** – w fazie tej ustala się szczegóły dotyczące nośników pamięci oraz wymagane metody implementacji i integracji mechanizmów bezpieczeństwa. Dobrym kompromisem jest implementacja mechanizmów po części programowo a po części sprzętowo, równoważąca potrzeby systemu i użytkownika. Implementacja mechanizmów bezpieczeństwa musi spełniać szereg warunków ([3 str 335] jest ich tam mnóstwo, tu wymienię ich kilka)
A. **Ekonomia mechanizmów** – powinny być jak najbardziej proste co redukuje koszty jak i ułatwia korekty błędów.
B. **Wydajne** – szczególne gdy często są wywoływane.
C. **Rozdzielenie uprawnień** – tam gdzie to możliwe powinno ze sobą współpracować wiele mechanizmów.
D. **Minimalne uprawniania** – uprawnienia podmiotów powinny być ograniczone do minimum.
E. **Znany projekt** - wykorzystane techniki zabezpieczeń powinny być dobrze znane. (musi być założone że potencjalni intruzi znają techniki projektowania)
F. **Weryfikowalność** – należy dowieźć że mechanizm bezpieczeństwa spełnia wymagania strategii bezpieczeństwa.

VII. ***Weryfikacja i testowanie a inżynieria oprogramowania*** – celem tej fazy jest zweryfikowanie czy wymagania i strategie bezpieczeństwa zostały poprawnie zaimplementowane. W tym celu wykorzystujemy metody formalne (np. kontrola zgodności wymagań w kodzie, analiza zachowania programu w zależności od parametrów) i nieformalne (np. podanie atakowi systemu ze strony wyspecjalizowanych hakerów/użytkowników).

Źródła

[1] – notatki **atm** z przedmiotu Bezpieczeństwo Baz Danych
[2] - https://docs.google.com/document/d/1sMLCBOmLh09CQpi7p0y93Un8MHfK8k0Ed9TpUiR5rxs/edit
[3] – **Stokłosa, Bilski, Pankowski** *„Bezpieczeństwo danych w systemach informatycznych”*

Parę definicji na początek:

Bezpieczeństwo – jest to stan, w którym nie występują zagrożenia znajdujące się na liście zagrożeń. [1]
Ponadto według [3] bezpieczeństwo baz danych to:

- Ochrona poufności i integralności danych (ochrona przed nieupoważnionym dostępem i modyfikacją zawartości baz danych)
- Utrzymywanie spójności baz danych (przechowanie warunków spójności [niesprzeczności] bazy danych)
- Zapewnienie dostępności (zawsze dostępne, bez względu na awarię sprzętu jak i łączy komunikacyjnych)

Polityka bezpieczeństwa[2:str 5][3:str. 15] – jest to szeroko rozumiany plan lub sposób działania przyjęty w celu ochrony danych oraz wysokiego poziomy bezpieczeństwa systemu informatycznego. Podczas jej tworzenia ustalamy między innymi ***listę zagrożeń***. Określa ona zbiór reguł i warunków, w jaki zarządza się informacją, chroni ją i rozdziela. Wyróżniamy 2 podejścia:

- To co nie jest zabronione, jest dozwolone
- To co nie jest dozwolone, jest zabronione (bardziej restrykcyjne, większy poziom bezpieczeństwa)

Lista zagrożeń[2:str 2] – zbiór zagrożeń przed którą chronimy naszą bazę (zarówno na poziomie oprogramowania jak i na sprzętowym.

- ustalane przez administratora bezpieczeństwa (oficera bezpieczeństwa)
- listy są różne dla różnych użytkowników/baz
- może być wiele dla jednego systemu

Omówienie zagadnienia

Celem metodyki jest podanie sposobów realizacji poniższych zadań poprzez dostarczenie narzędzi do modelowania, implementacji i weryfikacji (najtrudniejszy etap, brak uniwersalnych metod i narzędzi wspomagających tę operację) zamierzonego przedsięwzięcia:

- zdefiniowanie semantyki bezpieczeństwa baz danych – opisanie wymaganych właściwości bezpieczeństwa z wykorzystaniem pojęć bazy danych
- osadzenie tej semantyki w systemie bazy danych – zaimplementowanie jej w systemie zarządzania bazą danych oraz w zbiorze danych, którymi system operuje
- zweryfikowanie czy zaimplementowany system zapewnia wymagane właściwości zabezpieczeń

Do utworzenia bezpiecznej bazy danych potrzebujemy ustalić politykę bezpieczeństwa. W ramach jej tworzenia definiujemy przede wszystkim listę zagrożeń, przed którymi chronimy naszą bazę danych. Polityka bezpieczeństwa powinna zatem zawierać sposoby ochrony bazy danych na wszystkich poziomach bezpieczeństwa bazy danych, które są następujące:

1. **Poziom infrastruktury**

Baza danych znajduje się na fizycznym komputerze, który przechowuje w całości lub częściowo dane z bazy. Każde wykorzystywane pomieszczenie przez w/w komputery znajduje się w budynku, a ten z kolei posiada instalacje takie jak elektryczna, gazowa, wodnokanalizacyjna itd…., które stanowią potencjalne źródła awarii (zatopienie, spalenie) w/w komputerów, a co za tym idzie częściowe lub całkowite utracenie danych. Przy wyborze lokalizacji dla w/w komputerów, bierzemy pod uwagę takie czynniki jak:

a) *Lokalizacja budynku* – dbamy o to, aby budynki zawierające komputery znajdowały się na terenach niezagrożonych klęskami żywiołowymi (aby zmniejszyć ryzyko uszkodzenia sprzętu komputerowego, bądź np. zaprzestania na dłuższy czas dostaw energii eklektycznej)

b) *Lokalizacja oraz budowa pomieszczenia* – pomieszczenia zawierające komputery powinny znajdować się na środkowych kondygnacjach budynku, powinny posiadać klimatyzację (ochrona przed awarią na skutek przegrzania, komputery wydzielają dużo ciepła), dodatkowe mechanizmy autoryzacji jak czytniki kart elektronicznych (metoda identyfikacji oraz uwierzytelnienia na podstawie identyfikatorów materialnych, zabezpieczenia przed nie uprawnionym dostępem do fizycznych komputerów i np. zmianą uprawnień grupy użytkowników, umieszczeniu na komputerze wirusów komputerowych[np. przechwytywanie wpisywanych znaków na klawiaturze], bakterii[zużywają zasoby jak dysk, procesor i pamięć RAM co powoduje blokadę systemu] bądź koni trojańskich, skopiowaniem zawartości bazy danych na nośnik zewnętrzny), z kolei nie wskazane jest aby takie pomieszczenia posiadały okna co zwiększa ryzyko włamania.

c) *Sprzęt komputerowy* - dzięki postępowi technologicznemu awaryjność sprzętu komputerowego maleje, natomiast trwałość nośników danych wzrasta[3:str 23]. Wybierając sprzęt kierujemy się technologią produkcji (typ nośnika), jakością nośnika, warunkami pracy oraz przechowywania tegoż sprzętu(najlepsza dla nośników magnetycznych obecnie najczęściej wykorzystywanych to 20 stopni Celsjusza przy wilgotności na poziomie 30%, pozwala to zmniejszyć tempo degradacji warstwy magnetycznej nośnika). Tutaj szczególny nacisk kładziemy na dobór nośników ze względu na ich trwałość (deklarowaną żywotność, wykorzystane tworzywa do ich budowy) takich jak magnetyczne (np. dyski twarde) jak i również nośniki optyczne (wszelkiego rodzaju płyty oraz urządzenia odczytujące). Ponadto do serwery bazodanowe powinny być zabezpieczone przed zakłóceniami parametrów napięcia zasilającego (zaniki, obniżenia napięcia, przepięcia, wahania częstotliwości itd…) poprzez zastosowanie zasilania awaryjnego takiego jak akumulatorowe zasilacze awaryjne (UPS) czy spalinowe generatory prądotwórcze (banki, wojsko itd….)

2. **Poziom proceduralny**

Na poziomie tym definiujemy procedury bezpieczeństwa, które pozwalają m. in. małym kosztem uzyskać wysoki stopień bezpieczeństwa bazy danych. Przykładem takiej procedury jest składanie podpisu wraz z odnotowaniem daty pobrania klucza przez osobę fizyczną, co pozwala w łatwy sposób na identyfikację osób podejrzanych za kradzież, uszkodzenia, które

mogą zagrozić bezpieczeństwu danych w bazie danych.

a) *Szkolenia użytkowników* – aby zachować jak największe bezpieczeństwo bazy danych, należy sprawdzać wiedzę użytkowników, osobistą odpowiedzialność za zaniedbania, prowadzić systematyczne szkolenia. Jest to spowodowane faktem, iż naruszenia bezpieczeństwa bazy danych są nieświadome działania uprawnionych użytkowników jak i zaniedbania.

3. Poziom zabezpieczeń algorytmicznych i elektronicznych

Są to zabezpieczenia na poziomie oprogramowania. Ponieważ baza danych stanowi składową systemu informatycznego, dlatego wszystkie metody związane z bezpieczeństwem systemów informatycznych mają zastosowanie do systemu, w skład, którego wchodzi baza danych.

a) *Systemy kontroli dostępu (element SZBD)* – elementami takiego systemu są: podmioty (użytkownicy, procesy), obiekty (dane, programy), operacje (zapis, odczyt, tworzenie, modyfikacja, usuwanie…).

I. Jedną z metod jest tworzenie indywidualnych uprawnień (dla indywidualnych loginów) w systemie (nie tworzyć ich dla grupy użytkowników, pomimo puli wspólnych uprawnień, a indywidualnie) co pozwala na nadzorowanie aktywności każdego użytkownika osobno, ustalenie indywidualnej odpowiedzialności.

II. Nadajemy użytkownikowi tylko minimum niezbędne(uprawnień) do wykonywania określonych obowiązków lub jedynie w okresach, w których wymagane są te uprawnienia do wykonania obowiązków. Wykorzystujemy tutaj SQL lub inne języki czwartej generacji (QUEL, QBE) Dodatkowa def. - (systemy zamknięte – co nie jest dozwolone, jest zabronione Ad. Przeciwieństwem systemu zamkniętego jest otwarty co nie jest zabronione – jest dozwolone)

b) *Metody uwierzytelniania użytkowników* – sprawdzamy tożsamość użytkownika. Podstawowymi metodami weryfikacji tożsamości użytkownika jest:

I. Metoda oparta o wiedzę użytkownika – tutaj przykładem jest stosowanie poufnych haseł. Ustalamy reguły dotyczące tworzenia haseł, a następnie każdy z użytkowników tworzy własne hasło, którym autoryzuje swój dostęp do systemu (a więc i bazy danych). Hasła te mogą być czasowe, jak i jednorazowe. Tutaj przesyłane hasła jak i również przechowywane w bazie danych w postaci przekształconej np. za pomocą funkcji skrótu jednokierunkowego)

II. Metody biometryczne – wykorzystywanie faktu unikatowości pewnych cech człowieka i jego zachowania. Możemy w ten sposób ograniczyć dostęp do komputerów hostujących naszą bazę. Do metod biometrycznych zaliczamy czytniki linii papilarnych, kształt twarzy, dłoni…

c) *Utrzymywanie spójności bazy danych* - właściwe zarządzanie transakcjami współbieżnymi, odrzucanie oraz zatwierdzanie transakcji
d) *Zastosowanie oprogramowania antywirusowego* – zagrożenia takie jak wirusy komputerowe [np. przechwytywanie wpisywanych znaków na klawiaturze], bakterie[zużywają zasoby jak dysk, procesor i pamięć RAM co powoduje blokadę systemu], konie trojańskie mogą zostać ograniczone w istotny sposób przez profilaktykę taką jak zastosowanie oprogramowania antywirusowego.
e) *Tworzenie macierzy RAID (def. Kilka dysków magnetycznych, widzianych przez system operacyjny jako jeden dysk logiczny)* – zwielokrotniamy dane z naszej bazy na wiele nośników danych (np. dysków) co pozwala nam na znaczące zmniejszenie ryzyka utraty danych. Zawartość dysku uszkodzonego jest automatycznie odtwarzana na dysku zapasowym. Na temat macierzy RAID można poczytać na wykładzie 1 str. 9 z ISBD z 2 semestru.
f) *Kopiowanie danych z naszej bazy* – regularne tworzenie kopii zapasowych bazy danych, okresowe składowanie i dzienniki pracy.

31. Temporalne bazy danych

Opracowanie nr 1.

(źródło: notatki Kamili z wykładu ZSBD)

Temporalną bazą danych nazywamy system zarządzania bazą danych udostępniający wbudowane mechanizmy przechowywania i przetwarzania danych zmieniających się w czasie.

Dane zmienne w czasie to np. dane o zatrudnieniu pracownika, polisy ubezpieczeniowe, notowania akcji itp. Wprowadzenie jakiejkolwiek daty nie musi oznaczać operowania na danych zmiennych w czasie, np. data urodzenia nie jest taką daną. Z punktu widzenia baz temporalnych interesująca jest historia zmian.

Z danymi zmiennymi w czasie możemy powiązać dwa rodzaje znaczników czasowych:

- valid time
- transcation time

Bazę danych wykorzystującą oba wymienione znaczniki nazywamy bitemporalną.

"**Valid time**" to znacznik czasowy wskazujący kiedy zdarzenie miało miejsce lub okres, w którym dana była zgodna z prawdą, np. kiedy przyjęto pracownika lub kiedy obowiązywało ubezpieczenie.

"**Transaction time**" wskazuje, kiedy dana została zapisana do bazy danych. Czas transakcji ma wartość dyskretną, jest to punkt w czasie powiązany z konkretną transakcją.

Aktualizację w bazach bitemporalnych możemy wykonywać w dwojaki sposób:

- **aktualizacja proaktywna** - zmiany, które wprowadzamy są ważne od momentu wprowadzenia ich do bazy danych, np. wprowadzenie pracownika 19. maja znaczy, że jest zatrudniony od 20. maja.
- **aktualizacja retroaktywna** - zmiany wprowadzane są po fakcie

W 1996 roku zaproponowano rozszerzenie standardu SQL, nazwane **TSQL2**, wprowadzające funkcje wspomagające tworzenie baz z danymi zmiennymi w czasie. Rozszerzenie to wprowadziło do standardu operatory działające na przedziałach czasu, takie jak:

- BEFORE (czy jeden przedział jest wcześniejszy niż drugi)
- MEETS (czy koniec jednego przedziału jest początkiem drugiego)
- EQUALS (czy przedziały są równe)
- OVRELAPS (czy przedziały nachodzą na siebie)
- DURING (czy jeden przedział zawiera sie w drugim)
- MERGES (czy przedziały się łączą)
- CONTAINS (czy jeden przedział zawiera inny przedział)

Kolejnym rozwinięciem standardu SQL było rozszerzenie o nazwie **SQL/Temporal**, zaproponowane w 1999 roku. Rozszerzenie to, oprócz zmian w zapytaniach, wprowadzało zmiany w języku definiowania danych (DDM - Data Definition Language) oraz modyfikacji danych (DML - Data Modification Language).

W definicji tabel można wykorzystać słowa kluczowe VALIDTIME oraz TRANSACTIONTIME, realizujące wspomiane wcześniej znaczniki czasowe. Słowa kluczowe te można także stosować w zapytaniach, w celu wskazania czy chcemy operować na danych historycznych lub aktualnych.

W ramach znacznika TRANSACTIONTIME dostępne są tryby działania:

- **current** - operowanie na aktualnych danych
- **sequenced** - należy wyznaczyć kontekst, w którym chcemy zadwać zapytania/ modyfikować dane (przedział czasu)
- **nonsequenced** - operowanie na danych historycznych

Warto dodać, że nie spotyka się implementacji zaproponowanych rozszerzeń i zaproponowane funkcje trzeba implementować samodzielnie.

Opracowanie nr 2.

(z 2. wykładu wilczka: http://izinf.info/index.php?topic=7982.0)

Temporalna baza danych - baza danych

- posiadająca informację o czasie wprowadzenia lub czasie ważności zawartych w niej danych.
- SZBD oferują wbudowane mechanizmy przechowywania i przetwarzania danych zmieniających się w czasie.

- Wiele baz danych różnych dziedzin przechowuje informacje zmienne w czasie, np. dane o zatrudnieniu pracowników, notowania akcji, polisy ubezpieczeniowe, itp…
- Trudno jest w języku SQL wyrazić zapytania dotyczące tego rodzaju danych.

Temporalna BD	Nietemporalna BD
Zawiera dane historyczne. Modyfikacje rekordów polegają na wstawianiu nowych. Usuwanie może polegać na domknięciu ostatniego okresu ważności danych	Zwiera dane aktualne. Modyfikacje poprzez nadpisanie wartości. Usuwanie powoduje trwałe usuniecie danych.

Czas w temporalnych bazach danych jest traktowany jako uporządkowany ciąg punktów o pewnej określonej granulacji (dyskretna liniowa struktura czasu).

- Valid time (czas rzeczywisty) - znacznik czasowy jest momentem, w którym zdarzenie miało miejsce; okresem, w którym fakt był zgodny z prawdą.
- Transaction time (czas transakcji) – znacznik oznacza kiedy informacja została zapisana w bazie danych
- Bitemporal – użycie obu wymiarów równocześnie.

Języki temporalne

- TSQL2 – propozycja temporalnego rozszerzenie języka SQL92 (SQL2) – 1993.
- SQL/Temporal: Standard SQL:1999 Part 7 Temporal.
    - Wprowadza znaczniki czasu rzeczywistego i transakcyjnego
    - znaczniki czasowe: period, date
    - operacje na znacznikach czasu (contains, meet, overlaps)
- SQL:2003, Pakiet SQL/Temporal nie znalazł się w standardzie.

Prace nad temporalnymi bazami danych doprowadziły do rozszerzenia standardu języka SQL. Brak pakietu SQL/Temporal w SQL2003 ma swoją wymowę.

Więcej: załączniki http://izinf.info/index.php?topic=7982.0

32. Systemy aktywnych baz danych

z wykładu ZSBD (w1: http://izinf.info/index.php?topic=7982.0)

W klasycznych, nieaktywnych bazach danych

- wszelkie działania wykonywane przez system bazy danych,
- działania związane z realizacją procesów informacyjnych użytkownika są uaktywniane przez aplikacje bazy danych.
- Autonomiczne w stosunku do aplikacji są jedynie procesy realizujące działania systemowe, na przykład takie jak obsługa logów, wykrywanie zakleszczeń, przydział i zwalnianie zasobów, itp.

Aktywne systemy baz danych

- potrafią same uruchamiać zadania związane z realizacją procesów informacyjnych użytkownika, w sposób niezależny od aplikacji bazy danych.
- Wymaga to rozszerzenia funkcjonalności klasycznych systemów baz danych o trzy dodatkowe funkcje:
    - monitorowania przez system zarządzania bazą danych zdarzeń zachodzących w bazie danych,
    - ewaluacji warunków przypisanych tym zdarzeniom
    - autonomicznego „odpalania" akcji, czyli uruchamiania kodu specjalnych procedur składowanych w bazie danych.

Rozpoznanie predefiniowanych zdarzeń à uruchomienie predefiniowanych akcji (operacje na danych, wykonianie programu)

Aktywny System Zarządzania Bazą Danych (ASZBD)

- SZBD +definicja zdarzeń +reakcja na zdarzenia

Zastosowania ASZBD

- wymuszenie ograniczeń , reguł biznesowych
    - reakcja na złamianie ograniczenia (np. rollback)
- utrzymanie danych wyliczanych
    - materializacja widoków (np. aktualizacja widoku agregującego salda kont w okresie po rejestracji nowej operacji księgowej)
- automatyzacja procesów biznesowych
    - np. zlecenie przelewu bankowego po rejestracji płatności w systemie

Ogólnie przyjętym modelem definiowania aktywnych reguł jest tak zwany model ECA(Event, Condition, Action). Definicja obejmuje reguł przez trzy elementy:

- wystąpienie zdarzenia
- weryfikacja warunku
- odpalenie akcji

Przykład SQL :
CREATE TRIGGER trig_name
ON {table|view} {FOR | AFTER | INSTEAD OF} {INSERT, DELETE, UPDATE} <- **event**
AS
IF [SQL Condition] <- **condition**

BEGIN [SQL Statement] **<- action**
end…

33. Rozmyte bazy danych

źródło: http://zti.polsl.pl/bd3/rozmyte_bd_2011.pdf - wysłany przez Wilczka

Aproksymacyjne wyszukiwanie informacji w bazach danych

Wyróżniamy 3 rodzaje zapytań:
- precyzyjne
- zakresowe
- aproksymacyjne

Rodzaje zapytań aproksymacyjnych:
- Wektorowe – wyszukiwanie na podstawie podobieństwa dwóch wektorów, z których jeden reprezentuje wyszukiwany obiekt, a drugi kryteria pytania
- Oparte na prawdopodobieństwie – wyszukiwanie na podstawie prawdopodobieństwa spełnienia przez obiekt warunków pytania
- Rozmyte – wyszukiwanie na podstawie stopnia zgodności obiektu z kryteriami pytania
- W języku naturalnym – semantyka języka określa sposób interpretacji pytania

Twórca teorii zbiorów rozmytych: Lotfi Zadeh (1965r.)
Badania nad zastosowaniem teorii zbiorów rozmytych w bazach danych
- Zadawanie rozmytych pytań do bazy danych:
    a. Translator języka naturalnego wykorzystujący teorię zbiorów rozmytych PRUF - (Zadeh - 1978)
    b. Pierwszy rozmyty języka zapytań (Takahashi - 1991) (X jest bardzo niski)
    c. Fuzzy Query – tworzony od połowy 1997r. Przez Sonalysts
    d. Fquery – IBS PAN Warszawa (J. Kacprzyk, S. Zadrożny) dla MS Access
    e. SQLf – P. Bosc, O. Pivert itd.
- Zapamiętywanie rozmytych informacji w bazie danych:
    a. rozmyte modele danych (B. P. Buckles, F. E. Petry)
    b. rozmyty relacyjny model GEFRED – Uniwersytet w Granadzie (J. Cubero, M. Vila, K. Pons, J. Medina)
    c. rozmyte modele obiektowych baz danych – Belgia (R. De Caluwe itd.)
    d. zbiory rozmyte w bazach danych systemów geograficznych (M. Cobb, A. Yazici, K. Akkaya, V. Robinson)

FuzzySQL - rozszerzenie języka SQL, opracowany w 2003 r. w Instytucie Informatyki, Politechniki Śląskiej w Gliwicach - rozszerzany cały czas
Aproksymacyjny charakter wyszukiwania - poprzez wprowadzenie elementów teorii zbiorów rozmytych do klasycznej składni języka SQL, tym samym do umożliwienia konstruowania zapytań zawierających wyrażenia rozmyte
Rodzaje pytań oraz przechowywanych danych
- Precyzyjne pytania - dokładne dane w BD

- Rozmyte pytania - dokładne dane w BD
- Precyzyjne pytania - rozmyte dane w BD
- Rozmyte pytania - rozmyte dane w BD

Miejsca występowania wartości rozmytych w pytaniach niezagnieżdżonych

- Warunki filtrujące we frazie WHERE
- Warunki filtrujące we frazie HAVING
- Grupowanie wg wyrażeń zawierających
- wartości rozmyte – GROUP BY
- Porządkowanie wg kolumn zawierających
- wartości rozmyte – ORDER BY
- Wartości rozmyte we frazie SELECT

Teoria zbiorów rozmytych - podstawowe pojęcia

Zbiór rozmyty – zbiór par, w pewnej numerycznej przestrzeni rozważań X $A = \{(\mu_A(x), x)\}$, dla każdego $x \in X$, gdzie:

μ_A – funkcja przynależności zbioru rozmytego A – każdemu elementowi zbioru $x \in X$ przypisuje stopień przynależności $\mu_A(x)$ do zbioru A, przy czym $\mu_A(x) \in [0, 1]$

Wyrażenia rozmyte we frazie WHERE

SELECT A1, ..., Ak
FROM T
WHERE W;

W - warunek filtrujący postaci: X jest A
X – kolumna tablicy T bazy danych
A – zadana wartość

Ze względu na rodzaj wartości X i A warunki można podzielić na:

- klasyczne, gdzie X i A przyjmują wartości ostre (dokładne)
- X przyjmuje wartości ostre a A wartości rozmyte – reprezentowane przez funkcje przynależności
- X przyjmuje wartości rozmyte a A wartości ostre
- X i A przyjmują wartości rozmyte (reprezentowane przez funkcje przynależności)

Wartości rozmyte w pytaniach zagnieżdżonych

- w podzapytaniu wewnętrznym
- w podzapytaniu zewnętrznym
- w warunku wiążącym oba podzapytania

Wprowadzenie do języka zapytań elementów teorii zbiorów rozmytych pozwala na:

- Formułowanie rozmytych, nieprecyzyjnych warunków wyszukiwania (np. średni wzrost, szczupła sylwetka itd.)
- Określenie w odpowiedzi częściowej przynależności elementu do zdefiniowanego zbioru

Implementacja w SZBD PostgreSQL

Dlaczego SZBD PostgreSQL?

- dostępny kod
- możliwość tworzenia własnych typów, funkcji i operatorów

34. Wielowersyjne bazy danych

źródła:
http://sirius.cs.put.poznan.pl/~inf75975/ZSBD/Wielowersyjne%20metody%20synchronizacji.pdf
http://gdr.geekhood.net/gdrwpl/heavy/studia/sbd_podstawowe_pojecia.pdf

WIELOWERSYJNE BD
np. proces projektowania samochodu przez kilka zespołów, w pewnym momencie projekt musi być widoczny dla innych zespołów. Muszą występować warianty historyczne, w algorytmie podejmowania decyzji ważną rolę grają stare wartości. Są dwa typy takich baz:
•z wersjami obiektów – historyczne
•z wersjami obiektów – wariantowe
W systemie baz danych istnieje wiele wersji obiektów, nie muszą być ze sobą spójne. Mogą istnieć elementy nie tworzące kompletnej wersji(wariantowe), wielowersyjne zaś:
•z wersjami obiektów
•z wersjami baz danych

Z wersjami baz danych występuje problem redundantności, z reguły występują systemy z wersjami obiektów. Wśród obiektów wyróżnia się obiekt generyczny – ostatni, aktualny. Do hierarchii kompozycji, dziedziczenia dochodzi hierarchia wariantów (wersji). Może to być postać rozbudowanego grafu skierowanego (drzewo wywodu wersji - dla każdego obiektu). W drzewie wywodu wersji wszystkie muszą być blokowane w wypadku blokady. Powinno się też eliminować powtarzanie informacji w wersjach.

Każdy obiekt musi mieć identyfikator. W przypadku wersji – także jej identyfikator. Nie wszystkie obiekty muszą mieć wersje. Wersje z bazami danych. Blokujemy obiekty w całej wersji – jest to o wiele prostsze jeśli chodzi o redundancję. Przy blokowaniu wykorzystujemy drzewo hierarchii wersji. Transakcje są realizowane w jednej wersji, a także w innych wersjach: obiektowe– wewnątrz jednej wersji BD a bazowe– na wielu wersjach.

W wielowersyjnych bazach danych operacja modyfikacji polega na utworzeniu nowej wersji danej i zachowaniu wersji poprzedniej. Poszczególne dane zawierają wartość aktualną danej i listę historycznych wartości danej:

$$x = (x0, x1, x2, ..., xK).$$

dla dociekliwych:
http://dbs.mathematik.uni-marburg.de/publications/mypapers/1996/bs96.pdf
http://www.cs.bu.edu/fac/gkollios/cs591/tsbtree.pdf

35. Struktura zapisu plików rekordów w pamięci zewnętrznej

źródłó: materiały z ISBD - nie wiem czy nie wjachałam trochę w temat 36 :-)

Nośniki pamięci możemy podzielić **ulotne** (utrata zawartości przy braku zasilania) oraz **trwałe** (zawartość nie jest tracona przy braku zasilania). Do nośników ulotnych należy DRAM (dynamic RAM), natomiast trwałych dyski twarde, dyski optyczne, taśmy magnetyczne.
Najlepszym rozwiązaniem jest zapis całej bazy danych w pamięci operacyjnej, jednak znacznym ograniczeniem jest rozmiar pamięci operacyjnej i jej cena. W wielkich bazach danych jest konieczny zapis informacji w pamięci zewnętrznej (na dyskach twardych). Dane na dyskach są zapisywane i odczytywane w jednostkach zwanych blokami bądź stronami, a czas dostępu się różni, w zależności od miejsca na dysku (inaczej niż w RAM). Odczyt polega na transferze danych z dysku do pamięci operacyjnej (wewnętrznej), zapis to transfer danych z pamięci operacyjnej na dysk. Rozmiar bloku to kompromis pomiędzy amortyzacją kosztu I/O, a koniecznością odczytu nadmiarowych (niepotrzebnych) danych.
Nośniki pamięci możemy pogrupować według ich rodzajów (wraz ze skróceniem czasu dostępu do pamięci idzie także zwiększanie jej kosztu). Najszybszą i najdroższą pamięcią jest **cache**. Kolejna jest **pamięć operacyjna**, tańsza, ale o dłuższym czasie dostępu. Wolniejsza, ale tańsza jest pamięć **flash** (w tym dyski **SSD** – solid state drive). Kolejne w hierarchii są dyski magnetyczne, dyski optyczne i taśmy magnetyczne.
Czas potrzebny na odczyt/zapis bloku danych na dysku składa się z następujących elementów:

- **Seek time** – przeniesienie ramienia głowicy (1-10 ms)
- **Rotational delay** – oczekiwanie na odpowiedni blok (0-10 ms)
- **Transfer time** – odczyt/zapis danych z nośnika (1 ms)

W dyskach magnetycznych, w przeciwieństwie do np. pamięci operacyjnej, czas odczytu i zapisu zależy od lokalizacji na dysku.
Wyróżnia się następujące metody optymalizacji dostępu do pamięci zewnętrznej:

1. **Szeregowanie żądań** (np. algorytm windy). Za szeregowanie odpowiada kontroler dysku.
2. **Buforowanie**. Pamięć pośrednicząca przechowuje dane. Opłacalne, gdy czas przetwarzania jest większy bądź równy czasowi odczytu.
3. Wstępne **ładowanie bloków** (pre-fetch). Czytanie nadmiarowe.
4. **Wiele dysków** (macierze).
5. **Kopie dysków** (mirror).

Główne techniki zapobiegania awariom oraz zwiększaniu wydajności w pamięci zewnętrznej są dwa mechanizmy RAID w dwóch kombinacjach 1+0 oraz 0+1. Poziom 0 RAID odpowiada za podział bloków lub sekwencji bloków danych (ang. striping) pomiędzy poszczególne dyski. 1- mirroring, czyli replikacja (kopie dysków).
RAID 1+0 to RAID 0, którego elementami są macierze RAID 1.
RAID 0+1 to RAID 1, którego elementami są macierze RAID 0.
Reprezentacja fizyczna danych na dysku zależy od typu danych. Wyróżniamy pola o stałej i zmiennej długości. W przypadku pól o zmiennej długości zwykle długość poprzedza zawartość pola.
Rekord – zbiór powiązanych ze sobą elementów danych nazywanych polami. Wśród rekordów wyróżniamy rekordy o :

a) Stałym / zmiennym rozmiarze
b) Stałym / zmiennym formacie.

W przypadku **stałego formatu** posiadany schemat, który zawiera informacje o liczbie pól, typie każdego pola, porządku pól w rekordzie, przeznaczeniu każdego pola.

Przy **zmiennym formacie** rekord zawiera przyjęty format danych (rekordy samoopisujące). Zmienny format jest przydatny, gdy mamy „rzadkie" rekordy (dużo nullów), pola powtarzalne (uwaga! pogwałcenie postaci normalanej), założenie zmian w formatach danych (łatwa realizacja dodawania pól do schematu).
Zapis rekordu w blokach:

· **Separacja rekordów**

- nie ma potrzeby separować rekordów o stałym rozmiarze
- możliwe opcje separacji: zastosowanie markerów, każdy rekord ma swój rozmiar, offset rekordu w nagłówku bloku

· **spanned i unspanned**

Unspanned: rekord musi się mieścić w jednym bloku, spanned: rekord może być rozpięty pomiędzy blokami

· **clustering (grupowanie)**

różne rekordy, różnych typów przechowywane w jednym bloku (np. wydział razem ze swoimi pracownikami), przydatne przy częstym użyciu określonych kwerend.

· **Rekordy podzielone**

Typowe dla formatu hybrydowego, część stała w bloku, część zmienna w innym bloku.

· **Uporządkowanie**

Uporządkowanie rekordów w pliku (i bloku) po wartości klucza w celu efektywnego czytania rekordów w przyjętym porządku. Metody utrzymania kolejności:

- Kolejny rekord przylega fizycznie
- Powiązania
- Nadmiarowy obszar (namiary na obszar przechowywane w nagłówku)

· **Dostęp niebezpośredni**

W dostępie fizycznym podajemy adres rekordu (składający się z m.in. ID napędu, numeru cylindra, numeru ścieżki, numeru bloku i offsetu wewnątrz bloku). W dostępnie niebezpośrednim następuje odwzorowanie przyjętego identyfikatora rekordu na adres fizyczny. Dostęp niebezpośredni wewnątrz bloku realizowany jest poprzez wskazywanie rekordów z nagłówka. Nagłówek jest najczęściej zapisywany od początku bloku, rekordy od końca bloku.

· Bloki logiczne systemu plików – podajemy identyfikator pliku, numer bloku i numer rekordu lub offset, co jest tłumaczone przez system plików na fizyczny identyfikator bloku.

Operacje na rekordach w pamięci zewnętrznej:

1. Usunięcie

a. Natychmiastowe zwolnienie miejsca (niepraktyczne)

b. Oznaczenie usunięcia (łatwiejsze, ale tracimy pamięć. Dodatkowo trzeba przechowywać adresu usuniętych rekordów, do ponownego użycia)

2. Wstawianie

a. W rekordach nieuporządkowanych łatwo – wstawienie rekordu na końcu lub w wolnym miejscu. Problematyczne, gdy rekordy są zmiennej długości.

b. Rekordy uporządkowane – trudniej. Łatwo, gdy jest wolne miejsce, gdy brak – pamięć nadmiarowa.

Plik – zbór bloków (stron) z których każdy zawiera zbiór rekordów. Musi pozwalać na :

· Modyfikacje, wstawianie, usunięcie rekordu

· Odczytanie wskazanego rekordu

· Przeszukanie wszystkich rekordów

Techniki alokacji plików:

· **Alokacja ciągła** – ciągły zapis blok po bloku, czytanie całego pliku bardzo efektywne, ale rozszerzanie problematyczne.

· **Alokacja powiązana** – każdy blok zawiera wskaźnik do kolejnego bloku, czytanie całego pliku kłopotliwe, rozszerzanie efektywne

· **Alokacja ciągła-powiązana** – obszary ciągłego zapisu bloków (**segmenty**, **ekstenty**) zawierają wskaźnik do następnego obszaru. Czytanie całego pliku mniej kłopotliwe, rozszerzanie efektywne.

· **Alokacja indeksowa** – dostęp do bloków z rekordami przez bloki indeksowe. Czytanie wskazanego zakresu rekordów efektywne, rozszerzanie pliku efektywne.

Nagłówek pliku zawiera informacje umożliwiające SZBD dostęp do rekordów pliku (adres pierwszego bloku, opis formatu rekordu przy rekordach o stałym formacie, kody typów pól i separatorów gdy rekordy o zmiennym formacie).

Organizacja pliku – przyjęta metoda reprezentacji danych, organizacji rekordów, bloków i struktur dostępu (np. powiązań). Dostępne organizacje pliku:

· **Pliki stogowe** – najprostszy sposób, nowe rekordy zapisywane są w blokach w kolejności w jakiej napływają (pliki nieuporządkowane). Nowy rekord zapisywany jest na końcu pliku. Ze wzrostem bądź kurczeniem się pliku kolejne obszary pamięci są alokowane bądź zwalniane. Aby umożliwić realizację operacji na rekordach należy znać liczbę bloków w pliku, wolny obszar w blokach, liczbę rekordów w blokach. Implementacja plików stogowych to **listy** bądź **katalog bloków**.

· **Pliki uporządkowane** – rekordy zapisane w pamięci zewnętrznej mogą być uporządkowane na podstawie wartości wybranego pola rekordu, zwykle jest to klucz rekordu, który zapewnia unikalną wartość dla każdego rekordu. Pliki uporządkowane zapewniają szybki dostęp do rekordów, o ile wyszukiwanie odbywa się w oparciu o pole, po którym nastąpiło uporządkowanie, szybki dostęp do rekordów przy listowaniu w kolejności, znalezienie kolejnego rekordu często nie wymaga czytania kolejnego bloku (kolejny rekord w tym samym bloku). Możemy szukać : binarnie, interpolacyjnie, liniowo, blokowo. Usuwanie i wstawianie skomplikowane. Modyfikacja prosta, o ile nie zmienia pola kluczowego. Praktyczne rozwiązania wykorzystują pliki uporządkowane wraz z plikami indeksowymi.

· **Pliki wymieszane** – rekordy zapisane w pamięci zewnętrznej mogą być wyszukiwane na podstawie wartości wybranego pola rekordu, zwykle jest to klucz rekordu, który zapewnia unikalną wartość dla każdego rekordu. Przy wykorzystaniu funkcji mieszającej wyznaczany jest adres bloku przechowującego rekord o danej wartości klucza. Techniki mieszające w pamięci zewnętrznej zakładają adresowanie komór przy czym komory składają się z jednego lub więcej bloków pamięci przechowującej rekordy. Funkcja mieszająca przekształca pole mieszające ze względu na numer komory a nie bezwzględny adres bloku (co dzieje się w mieszaniu w pamięci wewnętrznej). Odpowiedni katalog komór przechowywany w nagłówku pliku przekształca numery komór na adresy bloków pamięci. Kolizja następuje wtedy, gdy funkcja mieszająca wskazuje komorę, która jest przepełniona. Wykorzystanie technik mieszających w pamięci zewnętrznej można podzielić na:

- Statyczną alokację komór – zakłada stałą liczbę komór.

o Dynamiczna alokacja komór – zakłada zmienną liczbę komór w pliku. Stosuje się tu dwie techniki:

§ Komory rozszerzalne- bierzemy k znaczących bitów wyniku funkcji mieszającej i wg nich dzielimy – k zwiększa się z czasem, gdy dodajemy nowe rekordy.

§ Komory liniowe – bierzemy k nieznaczących bitów h(K) – brak adresowania pośredniego, ale wykorzystywane są rekordy nadmiarowe.

36. Podstawowe techniki organizacji pamięci zewnętrznej

źródło: wykłady 3, 4 i 5 z ISBD

Plikiem nazywamy zbiór bloków (stron pamięci), zawierających rekordy. Plik musi udostępniać takie operacje jak wstawienie, usunięcie, modyfikacja rekordu, odczytanie wskazanego rekordu oraz przeszukanie wszystkich rekordów przy określonych warunkach.

Podstawowe techniki alokacji bloków:

- alokacja ciągła
- alokacja powiązana
- alokacja ciągła-powiązana
- alokacja indeksowa

W alokacji ciągłej bloki zapisywane są jeden po drugim - odczyt całego pliku jest bardzo efektywny, rozszerzanie go jest nieefektywne. Innym podejściem jest alokacja powiązana - blok zawiera wskaźnik do następnego bloku. W ten sposób uzyskuje się efektywne rozszerzanie pliku, ale utrudnia to odczyt całego pliku. Połączeniem tych dwóch koncepcji jest alokacja ciągła-powiązana - obszary ciągłego zapisu bloków (segmenty lub extenty) powiązane są ze sobą wskaźnikami - ułatwia to odczyt całego pliku i wciąż jest łatwo rozszerzalne. Alokacja indeksowa wykorzystuje bloki indeksowe, wskazujące na miejsce rekordów w blokach.

Nagłówek pliku składa się z adresu pierwszego bloku, formatu rekordu oraz kodów pól i separatorów w przypadku rekordów o zmiennym formacie. Dane te umożliwiają uzyskanie dostępu do rekordów zapisanych w pliku.

Operacje na plikach

- czytanie rekordow (retrieval) - np. operacje Find, Read, FindNext
- modyfikacja rekordów (update) - np. operacje Insert, Modify, Delete

Organizacja plików - metoda reprezentacji danych i organizacji rekodów, bloków i struktur dostępu

- pliki stogowe (heap)
- pliki uporządkowane (sorted)
- pliki wymieszane (hashed)

Pliki stogowe to najprostsza organizacja plików, nowe rekordy są zapisane na końcu pliku (pliki takie nie są uporządkowane). Wstawianie rekordów jest bardzo efektywne, przeszukiwanie musi być wykonywane liniowo (złożoność O(n))

Pliki uporządkowane zapewniają uporządkowanie rekordów wg wybranego pola (ordering field), najczęściej jest to klucz główny rekordu. Zapewnia to szybki dostęp do rekordów na podstawie tego pola oraz szybki dostęp do rekordów w kolejności. Ze względu na konieczność utrzymania uporządkowania wstawianie jest skomplikowane - wymaga odszukania miejsca, w które należy wstawić rekord. Z tego samego powodu usuwanie też nie jest łatwe do zrealizowania.
Dzięki uporządkowaniu możliwe jest zastosowanie technik wyszukiwania: liniowego, binarnego, interpolacyjnego i blokowego.

Przeszukiwanie blokowe zakłada sprawdzanie np. co setnego rekordu i w momencie natrafienia na rekord o kluczu większym niż poszukiwany dokładne przeszukanie poprzednich 99 rekordów - mało przydatne przy przeszukiwaniu plików w pamięci zewnętrznej ze względu na odczyt dużej ilości bloków. (Reszty nie opisuje bo są znane ;)

W plikach wymieszanych adres bloku, w którym przechowywany jest rekord o danym kluczu wyliczany jest na podstawie funkcji haszującej (mieszającej), której argumentem jest tzw. pole mieszające.

Techniki mieszające możemy podzielić na "internal hashing" (działające w pamięci wewnętrznej) oraz "external hashing" (działające w pamięci zewnętrznej).
W podejściu "external hashing" zakłada się adresowanie komór, zawierających bloki przechowujące rekordy. W takim podejściu funkcja mieszająca przekształca wartość pola mieszającego (np. klucza) na względny adres komory, a nie na adres bloku. Adres bloku uzyskuje się na podstawie katalogu komór, przechowywanego w nagłówku pliku.
W ramach "external hashing" wyróżnia się dwa sposoby zarządzania komorami - statyczną alokację komór (static hashing) oraz dynamiczną alokację komór (dynamic hashing). Pierwsza z nich zakłada stałą liczbę komór, w których przechowuje się bloki. Jeżeli w komorach zabraknie miejsca na bloki, to konieczne jest wykorzystanie list nadmiarowych rekordów - powoduje to obniżenie wydajności przeszukiwania.
Dynamiczna alokacja komór zakłada zmienną liczbę komór - ma to na celu zniwelowanie problemów wynikających z zastosowania stałej liczby komór. Alokację tą można zrealizować na dwa sposoby - w sposób rozszerzalny (extendable hashing) oraz liniowy (linear hashing).

Funkcje mieszające nie zapewniają, że różne wartości pola mieszającego zostaną przekształcone na różne adresy. Sytuację taką nazywamy kolizją.
Istnieje szereg technik **rozwiązywania kolizji**:

- adresowanie jawne (open addressing)
- łączenie (chaining)
- mieszanie wielokrotne (multiple hashing)

Adresowanie jawne polega na sprawdzaniu kolejnych adresów w poszukiwaniu pierwszego wolnego - pod tym adresem zostaje zapisany rekord.
Łączenie zakłada istnienie obszaru nadmiarowego powiązanego z każdym adresem - kolejne kolidujące rekordy zapisywane są w obszarze nadmiarowym. Powoduje to powstanie listy kolidujących rekordów.
Mieszanie wielokrotne to technika polegająca na wykorzystaniu dodatkowej funkcji mieszającej w wypadku wystąpienia kolizji. Jeśli po jej zastosowaniu wciąż występuje kolizja to stosuje się inną technikę (adresowanie jawne bądź mieszanie wielokrotne)

Indeks to dodatkowa struktura danych (wraz z algorytmem), która wspomaga lokalizację rekordów w pliku danych. Indeks nie ingeruje w położenie rekordów na dysku (niektore typy indeksów wymagają specjalnej organizacji pamięci zewnętrznej) i może być budowany i usuwany wedle potrzeb.
Użycie indeksu wymaga dostępu do niego oraz dostępu do bloku/bloków wskazanych przez indeks, przechowujących rekordy.

Klasyfikacja indeksów:
- jednopoziomowe lub wielopoziomowe
- podstawowe lub pochodne
- grupujące lub niegrupujące
- gęste lub rzadkie

Indeks możemy potraktować jako plik danych i na nim zbudować indeks - uzyskujemy wtedy indeks wielopoziomowy. Zaletą takiego rozwiązania jest coraz mniejszy rozmiar kolejnych indeksów, aż do rozmiaru, który można przechowywać bezpośrednio w pamięci wewnętrznej. Indeksy wielopoziomowe mogą być statyczne (ISAM) lub dynamiczne (B+ Tree).

Indeks podstawowy zakladany jest na polu porządkującym, posiadającym unikalne wartości. Najczęściej jest to klucz główny. Indeks pochodny zakładany jest na innych kolumnach tabeli.

Indeks grupujący budowany jest na polu porzadkującym o unikalnych wartościach, w pliku uporządkowanym (!). Indeks niegrupujący budowany jest na innych polach.

Indeks rzadki to taki, który nie zawiera wszystkich wartości pola indeksującego, natomiast indeks gęsty zawiera wszystkie wartości takiego pola - najniższy poziom indeksu wielopoziomowego musi być indeksem gęstym!

Indeks wielopoziomowy buduje się przez potraktowanie istniejącego indeksu jako pliku danych i założenie na nim kolejnego indeksu - podejście to znacząco ogranicza ilość dostępów do pamięci zewnętrznej potrzebnych do odszukania rekordu. Wyróżniamy 2 typy indeksów wielopoziomowych, wykorzystujące drzewa wyszukiwania: statyczne (ISAM) oraz dynamiczne (B+ drzewa).

Własności drzewa wyszukiwania:

- wartości w węzłach drzewa są wartością pola indeksowego
- każda wartość w węźle skojarzona jest ze wskaźnikiem do rekordu lub bloku w pliku danych
- każdy węzeł jest zapisywany jako blok pliku indeksowego
- algorytmy wstawiania i usuwania z drzewa muszą utrzymywać zrównoważenie drzewa

ISAM - wyszukiwanie zaczyna się od korzenia, schodząc "w dół" po kolejnych poziomach drzewa aż do liści (bloki danych). Wstawianie odbywa się przez wyszukanie odpowiedniego liścia i wstawienie wartości do tego liścia albo do powiązanego z nim bloku nadmiarowego. Usuwanie polega na odnalezieniu bloku danych w którym zapisany jest rekord, usunięcie go i ewentualnej dealokacji bloków nadmiarowych.
Drzewa te są nazywane statycznymi, ponieważ zmianom ulegają wyłącznie bloki danych - bloki indeksu nie są modyfikowane w czasie jego wykorzystania.

B-drzewa to zrównoważone drzewa przeszukiwań (balanced tree) wykorzystywane do budowania dynamicznych wielopoziomowych indeksów. B-drzewa wprowadzają dodatkowe ograniczenia zapewniające utrzymanie zrównoważenia drzewa i zapewnienia utrzymania oczekiwanego wypełnienia węzłów drzewa.
Najczęściej wykorzystywaną odmianą B-drzew są B+ drzewa. Zakładają one, że wskaźniki do danych znajdują się tylko w liściach.

37. Przetwarzanie i optymalizacja zapytań

// galaktyczne opracowanie na forum w .pdf

Zapytanie wyrażone w wysokopoziomowym języku zapytań, takim jak SQL, musi najpierw zostać odczytane, poddane analizie składniowej i zweryfikowane.
• Czytnik (ang. Scanner) - identyfikuje elementy języka (słowa kluczowe SQL, nazwy atrybutów, nazwy relacji) w tekście zapytania
• Analizator składniowy (ang. Parser) - sprawdza składnię zapytania w celu określenia czy sformułowano je zgodnie z regułami gramatyki języka zapytań.
• Drzewo zapytania - wewnętrzna reprezentacja zapytania, w postaci drzewiastej struktury
• Strategia wykonania zapytania - określana jest przez SZBD i określa strategię pobrania wyników zapytania z plików bazy danych. Proces wyboru najlepszej strategii określa się mianem optymalizacji zapytania

Strategie optymalizacji zapytań
• Reguły heurystyczne – sprawdzają się z większości sytuacji, ale nie gwarantują poprawnego działania w każdym przypadku
• Reguły z systematycznym szacowaniem - szacowany jest koszt różnych strategii wykonania zapytania. Wybierany jest plan o najniższym szacowanym koszcie

Wykorzystanie indeksów
INDEKSY – wskazują(ułatwiają) odnalezienie poszukiwanego elementu
• umiejętne definiowanie (jeśli wszystko poindeksujemy – znaczne spowolnienie bazy danych),
• umiejętne korzystanie (trzeba wiedzieć, że indeks jest i z niego skorzystać),
• właściwy dobór rodzaju indeksu do planowanych zastosowań

Wybrane rodzaje indekósw:
• BITMAPOWE – zastępowanie danych informacją o jej występowaniu 1 lub nie (np. mamy nazwę państwa, z którego pochodzi firma, możemy utworzyć indeks z atrybutem/kolumną Polska i wypełniać wiersze 1 lub 0) - przydatne w określonych typach zapytań np. count
• DRZEWIASTE (B-drzewa, R-drzewa, aP-drzewa, itp.)
• INNE (np. indeksy oparte o funkcję, bitmapowe z dołączeniem).

Wykorzystanie partycji
Partycje umożliwiają m. in. rozłożenie obiektów bazy danych na różnych dyskach w celu zrównoleglenia operacji wejścia/wyjścia, a także dodatkową (automatyczną) optymalizację niektórych zapytań przez działanie tylko na wybranych partycjach. W bardziej zaawansowanych SZBD (np. Oracle) dotyczy to zarówno tabel, indeksów jak i perspektyw zmaterializowanych.

http://www.equus.wroc.pl/studia/HD/HD_PS6.pdf
http://aragorn.pb.bialystok.pl/~gkret/BazyD/Dzienne/Wyklad_cz2/BD_w12.pdf

38. Zarządzanie transakcjami i ochrona przed awariami

Własności transakcji:
- Atomowość (atomicity) - w ramach jednej transakcji wykonują się albo wszystkie operacje, albo żadna
- Spójność (consistency) - o ile transakcja zastała bazę danych w spójnym stanie, po jej zakończeniu stan jest również spójny. (W międzyczasie stan może być chwilowo niespójny)
- Izolacja (isolation) - transakcja nie wie nic o innych transakcjach i nie musi uwzględniać ich działania.
- Trwałość (durability) - po zakończeniu transakcji jej skutki są na trwale zapamiętane (na dysku).

Wykrywanie zakleszczeń
- Konstruowanie grafu czekania (wait-for graph).
- Rozrywanie pętli zakleszczenia polega na wybraniu transakcji “ofiary” uczestniczącej w zakleszczeniu i następnie, jej zerwaniu i uruchomieniu od nowa.
- Wybór “ofiary” podlega różnym kryteriom - np. najmłodsza, najmniej pracochłonna, o niskim priorytecie, itd.

Unikanie zakleszczeń

- Wstępne żądanie zasobów (preclaiming): przed uruchomieniem każda transakcja określa potrzebne jej zasoby. Nie może później nic więcej żądać. Wada: zgrubne oszacowanie żądanych zasobów prowadzi do zmniejszenia stopnia współbieżności.
- Czekasz-umieraj (wait-die): jeżeli transakcja próbuje dostać się do zasobu, który jest zablokowany, to natychmiast jest zrywana i powtarzana od nowa. Metoda może być nieskuteczna dla systemów interakcyjnych (użytkownik może się denerwować koniecznością dwukrotnego wprowadzania tych samych danych) oraz prowadzi do spadku efektywności (sporo pracy idzie na marne).

Dzienniki operacji:

- UNDO:
  - Dla każdej operacji generowany jest rekord wycofania (zawierający poprzednią wartość)
  - Przed modyfikacją x na dysku, rekord wycofania odpowiadający x musi być na dysku (najpierw zapis dziennika)
  - Przed zapisem zatwierdzenia do dziennika wszystkie zmiany muszą być zapisane na dysk
- REDO
  - Dla każdej operacji generowany jest rekord powtórzenia (zawierający nową wartość)
  - Przed modyfikacją x na dysku, wszystkie rekordy powtórzeń transakcji zmieniającej x (commit także) muszą być na dysku

Implementacja blokad

- Menadżer blokad można implementować jako odrębny proces do którego transakcje przekazują żądania blokad i zwolnień.
- Menadżer blokad odpowiada na żądanie potwierdzeniem założenia blokady (bądź komunikatem przerwania transakcji w przypadku deadlock'a).
- Transakcja oczekuje na odpowiedź menadżera.
- Menadżer blokad utrzymuje tablice blokad przeznaczoną do zapisywania założonych blokad.
- Tablica blokad jest zwykle implementowana jako tablica asocjacyjna (hash table).

39. Modele systemów wyszukiwania informacji

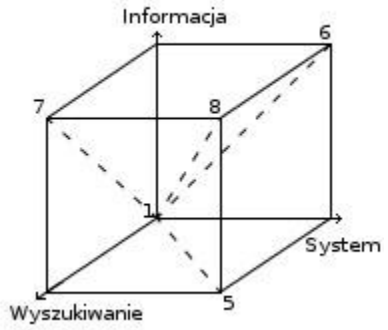
Model boolowski

W modelu tym dopuszcza się stosowanie alternatyw, koniunkcji i negacji. Należy jednak zwrócić uwagę na błędne zapytania, które są niepoprawne z punktu widzenia logiki matematycznej. Przykładem takiego zapytania może być „osoby starsze i dzieci do lat trzech są zwolnione z opłaty". Na pierwszy rzut oka nie razi nas żaden błąd w tym zdaniu. Jednakże z punktu widzenia logiki matematycznej zostaną wyszukane

osoby, które są starsze i są dziećmi do lat trzech, co prowadzi do sprzeczności. Prawidłowa forma zapisu takiego zapytania powinna brzmieć „osoby starsze lub dzieci do lat trzech są zwolnione z opłaty”.
Przykładowo – posiadając zbiór dokumentów:
d1={system, wyszukiwanie, informacja}
d2={system, dokument}
d3={system, informacja, dokument}
d4={wyszukiwanie, dokument}
odpowiedzią na zapytanie f(system, ¬informacja) będzie jedynie dokument numer 2. Dokument numer 1 i 3 zawierają deskryptory „informacja”, dlatego nie mogą zostać zwrócone w wynikach zapytania.

Model wektorowy

W przypadku tego modelu, każdy dokument można przedstawić za pomocą wektora.



Dla przykładowych deskryptorów „system”, „wyszukiwanie” i „informacja”, każdy dokument będzie znajdował się w jednym z wierzchołków wyżej przedstawionego sześcianu. Możliwe jest zatem osiem pozycji:

1. dokument nie zawiera, żadnego deskryptora,
2. zawiera tylko „system”,
3. zawiera tylko „wyszukiwanie”,
4. zawiera tylko „informacja”,
5. „system” i „wyszukiwanie”,
6. „system” i „informacja”,
7. „wyszukiwanie” i „informacja”,
8. „system”, „wyszukiwanie” i „informacja”.

Model ten zakłada, że każdy deskryptor jest jednakowo ważny. Nie zawiera on żadnych wag, dlatego niemożliwe jest umiejscowienie dokumentu w innym punkcie niż w wierzchołku. W takim wypadku jedynymi możliwymi wartościami w macierzy będącej charakterystyką dokumentów są wartości 0 albo 1. Oznacza to, że sprawdzana jest wyłącznie obecność termu w danym dokumencie.
System z tej rodziny analizuje kolekcję dokumentów, a następnie zapisuje w postaci macierzy ich charakterystykę. Charakterystyka ta jest macierzą m x n, gdzie:

- m – jest liczbą dokumentów w kolekcji systemu,
- n – jest łączną liczbą różnych termów ze wszystkich dokumentów.

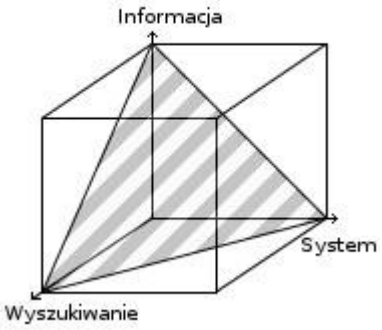
Zakładając, że termy „system”, „wyszukiwanie” i „informacja” są jedynymi termami w całej kolekcji dokumentów, powyższy przykład można przedstawić następująco:

	system	wyszukiwanie	informacja
d1	1	1	0
d2	1	0	0
d3	0	1	0
d4	1	1	1
…	...	…	...
dn	0	0	1

Jednak macierz ta może osiągać ogromne rozmiary. W przypadku kilku tysięcy dokumentów, z każdym kolejnym terminem przybywa dodatkowa kolumna z taką samą ilością wierszy.
Wyszukiwanie dokumentów odbywa się kolumnowo. Wybieramy jedynie te, które opisują występowanie termu w dokumencie, a następnie szukane są wiersze, które dla zadanych termów przyjmują wartość 1.

Model wagowy

Jest to szczególny przypadek modelu wektorowego. Różni się on tym, że poszczególnym termom z dokumentu możemy przypisywać dodatkowe wagi.



W odróżnieniu od klasycznego modelu wektorowego, dokumenty mogą znajdować się nie tylko na wierzchołkach sześcianu, ale także na powierzchni między nimi, będącej zaznaczonym trójkątem wewnątrz sześcianu. Przestrzeń tą nazywamy rozkładem procentowym termów.
Aby przejść z modelu wektorowego na wagowy, należy przekształcić jego charakterystykę dokumentów. Suma wag wszystkich termów dla każdego dokumentu musi zawsze wynosić 1. Dlatego najłatwiejszym sposobem jest podzielenie wartości z macierzy modelu wektorowego przez sumę wszystkich termów w danym dokumencie. Dla wcześniejszego przykładu, model wagowy będzie wyglądał następująco:

	system	wyszukiwanie	informacja
d1	1/2	1/2	0

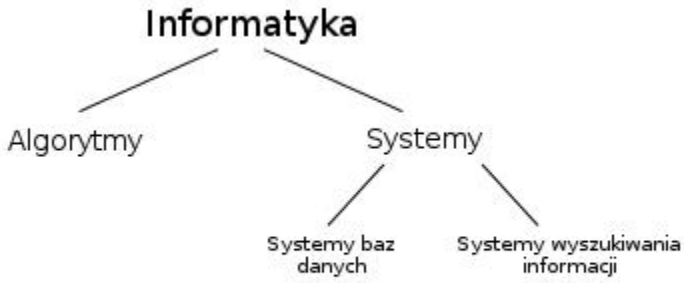
d2	1	0	0
d3	0	1	0
d4	1/3	1/3	1/3
…	...	…	...
dn	0	0	1

Zarówno deskryptory jak i wagi danego dokumentu mogą się zmieniać. Mają na to wpływ między innymi trzy czynniki:

- zmieniła się treść dokumentu,
- zmienia się nasza wiedza na na temat dziedziny poruszanej przez dany dokument,
- dokument może być oceniany różnie w zależności od osoby, która go ocenia.

Model hierarchiczny

Model ten, w odróżnieniu od pozostałych, wprowadza zależności pomiędzy deskryptorami. Jak przedstawia rysunek poniżej, można go przedstawić za pomocą drzewa, gdzie wierzchołek nadrzędny jest deskryptorem bardziej ogólnym, a wierzchołek podrzędny jest bardziej szczegółowy.



Charakterystyczną cechą takiej budowy systemu wyszukiwania informacji jest to, że pytając o deskryptor bardziej ogólny, zostaną zwrócone w wynikach również dokumenty odpowiadające na termy bardziej szczegółowe. Odnosząc się do przykładu pokazanego powyżej, dla zapytania f(informatyka), zostaną zwrócone wszystkie dokumenty. Analogicznie dla zapytania f(systemy) zostaną zwrócone dokumenty zawierające deskryptory „systemy", „system wyszukiwania informacji" i „systemy baz danych".
W modelu tym można również wprowadzać wagi, podobnie jak w modelu wektorowym. Kolejną zaletą jest to, że możemy wprowadzać relacje pomiędzy wyrazami, dzięki czemu będziemy mogli rozpatrywać synonimy, przykładowo: droga, ulica, jezdnia.

40. Metody wyszukiwania informacji

Metoda pełnego przeglądu

Posiadając opis dokumentu jesteśmy w stanie przeprowadzać wyszukiwanie poszczególnych termów. Może ona być wykorzystywana w momencie, gdy posiadamy macierz term-dokument. Podając zapytanie, wybieramy jedynie te kolumny, które odpowiadają danym termom, a następnie przeszukujemy wszystkie wiersze w poszukiwaniu dokumentów, które odpowiadają na wszystkie z nich.
Metoda ta nie sprawdza się w praktyce, gdyż za każdym razem podając nowe zapytanie, musimy przeszukiwać całą bazę dokumentów na nowo, co w przypadku macierzy posiadających po kilka tysięcy dokumentów potrafi być bardzo czasochłonne.
Aby przyspieszyć wyszukiwanie metodą pełnego przeglądu, należy przygotować system tak, żeby nie trzeba było za każdym razem analizować posiadanej bazy. Przykładowo posiadając macierz dokumentów:

	system	**wyszukiwanie**	**informacja**
d1	1	1	0
d2	1	0	0
d3	0	1	0
d4	1	1	1
d5	0	0	1

Można ją zapisać jako zbiór dokumentów zawierających odpowiednie termy. W tym przypadku będą to:
d1={system, wyszukiwanie}
d2={system}
d3={wyszukiwanie}
d4={system, wyszukiwanie, informacja}
d5={informacja}

Posiadając bazę dokumentów zapisanych w ten sposób, system jest w stanie sprawdzić od razu, czy dany dokument odpowiada na zapytanie. Jednakże takie rozwiązanie dalej wymaga przeglądania wszystkich dokumentów. Rozwiązaniem tego problemu jest zastosowanie list inwersyjnych.

Listy inwersyjne

Zastosowanie list inwersyjnych pozwala nam na wyeliminowanie konieczności każdorazowego przeglądania wszystkich dokumentów w poszukiwaniu zadanego termu. Jak sama nazwa wskazuje, listy inwersyjne są odwrotnością zapisu stosowanego w metodzie pełnego przeglądu. W przypadku tym, nie są przechowywane listy termów w danym dokumencie. Zamiast tego system tworzy listę wszystkich termów w danym systemie, a w każdym elemencie tejże listy przechowuje kolekcję dokumentów, które odpowiadają na zadany term.
Zastosowanie list inwersyjnych, do powyższego przykładu, będzie wyglądało następująco:
T={system, wyszukiwanie, informacja}

L(t1)={d1, d2, d4}
L(t2)={d1, d3, d4}

L(t3)={d4, d5}

Najdłuższa z list może zawierać maksymalnie tyle elementów, ile jest dokumentów w systemie. Dzieje się tak ponieważ dany dokument, w obrębie jednego elementu listy, jest przechowywany jednokrotnie. Aktualizacja list przy dodawaniu nowego dokumentu do kolekcji polega wyłącznie na przeanalizowaniu danego dokumentu i dopisaniu go do elementów listy, które zawierają charakterystyki termów w nim zawartych.

Podstawowym problemem list inwersyjnych jest redundancja. Objawia się to poprzez konieczność przechowywania tego samego dokumentu na każdej liście termu znajdującego się w danym dokumencie. Jednakże nadmiarowość ta jest rekompensowana przez szybkość dostępu do tych dokumentów, przy wykonywaniu zapytania.

Listy inwersyjne umożliwiają stosowanie wag. Wtedy elementy listy mają postać par (dokument, waga).

Metoda łańcuchowa

Innym rozwiązaniem problemu konieczności przeglądania wszystkich dokumentów jest metoda łańcuchowa. Jest ona zapisem, który umożliwia przechowywanie odnośnika do kolejnego dokumentu zawierającego dany term. Charakterystyka dokumentu składa się z elementów (tx | Oy), gdzie:

- tx – jest kolejnym termem,
- Oy – jest odnośnikiem do kolejnego dokumentu, w który dany term się znajduje.

W momencie, gdy dokument jest ostatnim w kolekcji, zawierającym dany term, odnośnik do kolejnego dokumentu oznacza się symbolem pustego zbioru.

Przedstawiając za pomocą tej metody ten sam przykład otrzymamy:

d1={(system | d2), (wyszukiwanie | d3)}

d2={(system | d4)}

d3={(wyszukiwanie | d4)}

d4={(system | ∅), (wyszukiwanie | ∅), (informacja | d5)}

d5={(informacja | ∅)}

Metoda ta wymaga przechowywania dwóch dodatkowych informacji:

- adres początkowy dla danego termu,
- długość łańcuchów.

Dla termów „system” oraz „wyszukiwanie” adresem początkowym jest dokument d_1, a długość łańcucha wynosi 3 (występują w trzech dokumentach), z kolei dla termu „informacja” dokumentem początkowym jest dokument d_4, a długość łańcucha wynosi 2. Elementy te są wykorzystywane w algorytmie wyszukiwania informacji, który działa następująco:

Dla przykładowego zapytania: r={system, informacja}

1. Porównywane są ze sobą długości łańcuchów obu termów i wybierany jest krótszy. W tym przypadku L(system)>L(informacja) dlatego zostanie wybrany drugi z nich.
2. Adresem początkowym termu „informacja” jest dokument d_4 i od niego rozpocznie się wyszukiwanie.
3. Sprawdzane jest czy w dokumencie znajduje się term „system”?
   - Tak – dokument jest dopisywany do listy wyników.
   - Nie – wykonywany jest krok 4.
4. Sprawdzane jest czy istnieje odnośnik do kolejnego dokumentu?
   - Tak – przejdź do dokumentu, na który wskazuje odnośnik i wykonaj krok 3.

- Nie – zakończ.

Wynikiem dla tego zapytania będzie jednoelementowa lista dokumentów składająca się z dokumentu d_4.

Metoda Wonga - Chonga

metoda iloczynów kanonicznych.
Dokumenty : d1 = {t1,t2,t3} - adres 10, d2 ={t2,t3,t4} - adres 20, d3={t1}- adres 30, d4={t1,t4} - adres 40, d5={t1,t2,t3}- adres 50, d6={t3,t4} - adres 60, d7={t1,t3} - adres 70
t - terminy opisujące dokument
T = {t1,t2,t3,t4} - słownik terminów
r = {t1,t3} - kwerenda
t1 ∧t3 - iloczyn kanoniczny, np. t1 ∧¬ t2∧t3∧¬ t4
2 ∧n - 1 - iloczyn z n-negacjami nie ma sensu rozważań, 2 ∧n - tyle klas
Każdy dokument należy dokładnie do jednej klasy.
Klasa nr 1 , iloczyn kanoniczny : t1 ∧ t2∧t3∧ t4, adres klasy: 50
Klasa nr 2, iloczyn kanoniczny : t1 ∧ t2∧t3∧¬ t4, adres klasy: 10
Klasa nr 3, iloczyn kanoniczny : t1 ∧ t2∧¬ t3∧ t4, adres klasy: zbiór pusty
Klasa nr 4, iloczyn kanoniczny : t1 ∧ t2∧¬ t3∧¬ t4, adres klasy: zbiór pusty
Klasa nr 5, iloczyn kanoniczny : t1 ∧ ¬ t2∧t3∧ t4, adres klasy: zbiór pusty
Klasa nr 6, iloczyn kanoniczny : t1 ∧¬ t2∧t3∧¬ t4, adres klasy: 70
….
Klasa nr 15, iloczyn kanoniczny : t1 ∧ t2∧t3∧¬ t4, adres klasy: 10

r (t1 ∧t2∧¬ t4) ⇔ r (t1 ∧¬t2∧t3∧¬ t4) ∨ r (t1 ∧t2∧¬ t3∧¬ t4) = 2. (numer klasy) ∨zbiór pusty ⇔ 2. = 10 (adres klasy)

Jeśli dwa dokumenty mają te same opisy to w ramach jednej klasy przechowujemy więcej adresów w klasie.
Zalety metody: brak redundancji
Wady: może być dużo pustych klas - macierz rzadka

Metoda Chowa

później wkleje skany z wykladu

Metoda Luma

później wkleje skany z wykladu

41. Sieci neuronowe w systemach wyszukiwania informacji

Od kiedy zaczęto stosować w praktyce sztuczne sieci neuronowe, okazało się że ich zdolności do klasyfikacji i aproksymacji zdecydowanie wyprzedzają inne metody. Poza tym sieci neuronowe mają zadziwiającą zdolność do ignorowania większości błędów.
Naturalnym więc było, że sieci neuronowe znajdą zastosowanie w dziedzinie wyszukiwania informacji. Wykorzystuje się tutaj głównie sieci klasyfikujące jedno bądź dwuwarstwowe. Na wejście sieci neuronowej kieruje się zapytanie, natomiast wyjściem są dokumenty, do których zapytanie zostanie zaklasyfikowane.
W warunkach laboratoryjnych, czyli dla niewielkiej ilości dokumentów, metoda wyszukiwania wykorzystująca sieci neuronowe sprawdza się bez zarzutu, dając świetne, zarówno ilościowo jak i jakościowo wyniki.
Jednak stosowanie sieci neuronowych do klasyfikacji kilku milionów dokumentów, czyli rzeczywistej liczby spotykanej w Internecie, jest aktualnie fizycznie niemożliwe, przy wykorzystaniu aktualnie znanych rozwiązań. Jednak spojrzenie na wyszukiwanie poprzez sieci neuronowe pozwala znacznie rozszerzyć perspektywy na przyszłość, tym bardziej, że alternatywne metody przeszukiwania są coraz bardziej pożądane.

42. Sieci semantyczne w wyszukiwaniu informacji

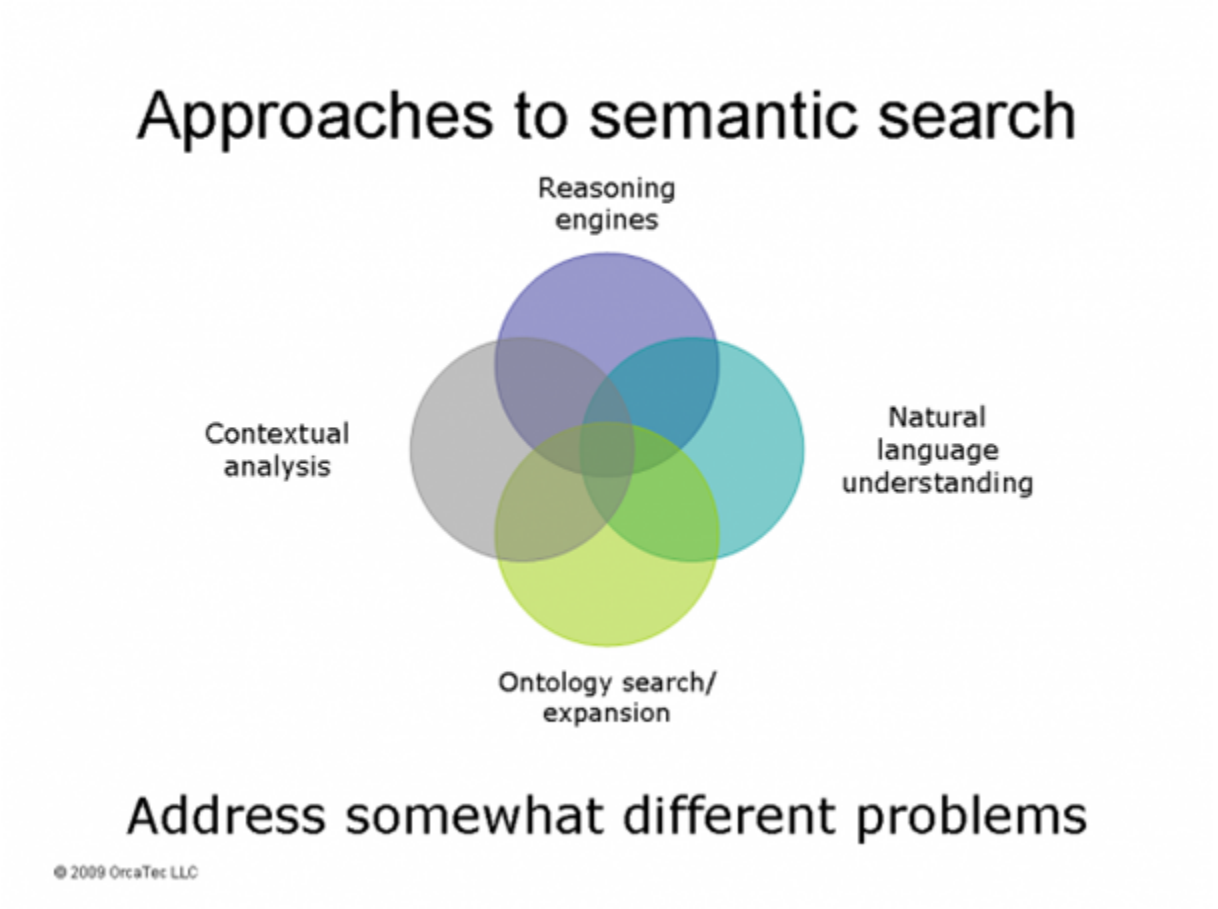
źródła:
http://ittechblog.pl/2012/03/18/krok-w-przyszlosc-dla-wyszukiwarki-google-zalazki-wyszukiwania-semantycznego/
http://4programmers.net/Artyku%C5%82y_do_poprawy/Sieci_Semantyczne_-_WWW_nast%C4%99pnej_generacji
http://www.intelligence.ideo.pl/nasza-oferta/wyszukiwanie-semantyczne/

Koncepcja Sieci Semantycznej

• *Semantic Web* rozszerza *WWW* o nowa funkcjonalnosc oparta na semantycznej analizie danych
• Zrezygnowanie z prób przetwarzania wiedzy dostepnej w sieci w oparciu o narzedzia Sztucznej Inteligencji na rzecz rozwoju *jezyków opisu* dostepnej w sieci wiedzy.
• Idea maszynowego przetwarzania wiedzy
• Zastapienie mechanizmu wyszukiwania słów kluczowych mechanizmem udzielania odpowiedzi
• Poszukiwanie informacji umozliwiajace analize wielu dokumentów jednoczesnie
• W perspektywie, umozliwienie działania inteligentnym agentom działajacym w sieci.

Sieć semantyczna – w dużym uproszczeniu – to sieć, w której danym elementom nie jest nadawana tylko nazwa (etykieta), ale także znaczenie, jak także znaczenie w aspekcie położenia danej informacji. Dla przykładu słowa “tanie zamki” nic wyszukiwarce nie mówią. Natomiast jeśli w sieci semantycznej takiemu stwierdzeniu nadamy znaczenie w odniesieniu do innych elementów i intencji poszukującego, może się okazać, że zwrócenie wyników z tanimi budowlami nie będzie najlepszym rozwiązaniem. Intencją szukającego jest bowiem znalezienie pasmanterii w obrębie 200 metrów. Upraszczając – semantyka to próba zrozumienia intencji, a także specyficzne umiejscowienie elementu i nadanie mu dodatkowych znaczeń. To sieć

inteligentna.



Charatkerystyka Sieci Semantycznych

Tak samo jak Internet, Sieć Semantyczna (SS) będzie się charakteryzować silną decentralizacją. Rozproszone systemy operujące w SS, działające dla potężnych korporacji jak i zwykłych użytkowników będą dostarczały szereg inteligentnych usług, niemożliwych do zrealizowania w dzisiejszym WWW. Sieci Semantyczne umożliwią automatyczny dostęp do informacji oparty na semantycznych metadanych, umożliwiających przetwarzanie i rozumienie (wykorzystując techniki z dziedziny AI) informacji przez maszyny. Udostępnienie maszynom możliwości operowania na poziomie semantyki umożliwi stworzenie sieci, która dostarczy nowy i niespotykany poziom wyrafinowanych usług. Powstanie ogromna sieć wiedzy, wzbogacona o możliwość inteligentnego przetwarzania tych informacji przez maszyny. Różne zautomatyzowane usługi będą w stanie pomóc użytkownikowi wykonać skomplikowane zadania poprzez dostęp do informacji rozumianej przez maszyny. Ten proces w fazie końcowej stworzy ogromny system wiedzy, z różnego rodzaju wyspecjalizowanymi usługami wnioskującymi (reasoning services) – systemami, które będą nam pomagały niemalże w każdym aspekcie naszego życia i staną się tak niezastąpione, jak dostęp do elektryczności dzisiaj. Autorem i pomysłodawcą SS, jako sukcesora dzisiejszego internetu jest sam twórca WWW – Tim Berners-Lee, aktualnie dyrektor konsorcjum WWW (W3C), które z pomocą

wielu firm softwarowych oraz ośrodków badawczych pracuje nad przyszłością internetu. Od kiedy pojęcie oraz idea Sieci Semantycznych została zaproponowana przez Tima w 1999, została ona określona jako najbardziej obiecująca technologia internetowa, która pozwoli zrewolucjonizować informatykę.

Wyszukiwanie semantyczne

Jak działa?
Dzięki zastosowanym algorytmom oprogramowanie wykorzystywane do przeszukiwania bazy danych będzie nie tylko odnajdywało istotne słowa i frazy, ale także lepiej je rozumiało, podając wyniki w konkretnym, zadanym przez operatora kontekście. W tym znaczeniu wyszukiwanie semantyczne zmierza do zautomatyzowanego analizowania i przetwarzania zawartości znaczeniowych informacji znajdujących się w bazie danych.

Zbieranie informacji w oparciu o wyszukiwanie semantyczne odbywa się na zasadzie analizy i rozumienia kontekstu wynikającego z szeregu fraz składających się na zapytanie, a podstawowym założeniem jest precyzyjne dopasowanie wyników do danego zapytania.

Zaawansowane algorytmy stosowane w wyszukiwaniu semantycznym to połączenie inteligentnych rozwiązań w zakresie filtrowania i nawigacji danych. Podczas gdy standardowe mechanizmy przeszukujące bazy danych jedynie rozpoznawały ciąg znaków, teraz mogą także analizować ich kontekst. Ze względu na umiejętność rozpoznawania i określania zależności między danymi znajdującymi się w bazie mechanizmy wykorzystywane w tej technologii dokonują precyzyjnego wnioskowania.

Wyszukiwanie semantyczne zyskuje coraz większe znaczenie:
- stanowi podstawę sieci inteligentnych, które poprzez nadanie znaczenia elementom znajdującym się w bazach danych pozwalają na przeprowadzanie precyzyjnych analiz
- jest podstawą narzędzi służących do wspomagania procesów analitycznych
- umożliwia nieograniczony przepływ danych i ujednolicenie informacji zgromadzonych w bazie danych

Wśród wyszukiwarek semantycznych rozróżnia się przede wszystkim dwa rodzaje: te, które odnajdują informacje i te, które je analizują.

Zastosowania
Innowacyjna technologia ma wspomóc procesy analityczne zmierzające do pełnego sprecyzowania określonych baz danych istotnych w wielu dziedzinach gospodarczych. Rozwiązania inteligentne wykorzystywane w biznesie wspomagają rozwój i ułatwiają interakcję pomiędzy firmą a odbiorcą usług, głównie dzięki tworzeniu pełnego obrazu marki i prezentowaniu usystematyzowanego obrazu branży.

Wyszukiwanie semantyczne znalazło już szerokie zastosowanie w obszarach biznesowych:

- jest podstawą prac analitycznych wspomagających procesy badawcze, rozwojowe i wdrożeniowe
- pomaga w tworzeniu baz danych opartych o znaczenia wyrazów
- ułatwia poszukiwanie informacji i przeszukiwanie dokumentów
- pozwala na systematyzację informacji związanych z ofertą produktowo - usługową
- stanowi pomoc w zakresie merchandisingu

Wyszukiwanie semantyczne związane jest z opracowaniem wydajnych narzędzi poszukiwania i przetwarzania informacji, wyróżniających się efektywnością i ekonomicznością.

43. Modele danych dla multimedialnych baz danych, standard MPEG-7

źródło: http://wazniak.mimuw.edu.pl/images/2/2d/ZSBD-2st-1.2-w7.tresc-1.4.pdf

MPEG-7 standaryzuje sposób reprezentacji wszelkiego rodzaju metadanych o wszelkiego typu danych multimedialnych i jest oparty o XML. Nie standaryzuje metod ekstrakcji i wykorzystywania metadanych. Na razie MPEG-7 jest słabo wspierany w systemach zarządzania bazą danych. Wsparcie dla opisów MPEG-7 nie wykracza w nich poza ogólne wsparcie dla XML.

- MPEG-7 = Multimedia Content Description Interface
- Standard normujący opisy zawartości multimedialnej
- Adresatami opisu mają być ludzie lub programy komputerowe
- Przewidywane zastosowania:
  – wyszukiwanie w multimedialnych bazach danych
  – filtracja treści multimedialnych

W przeciwieństwie do standardów MPEG-1, MPEG-2 i MPEG-4 dotyczących reprezentacji (sposobu kodowania) zawartości obiektów multimedialnych, standard MPEG-7 (Multimedia Content Description Interface) dotyczy opisu zawartości multimedialnej. Standard MPEG-7 normuje jedynie samą formę opisu, nie standaryzując technik generacji i konsumpcji opisów. Przyczyny wyłączenia generacji opisów, ekstrakcji właściwości oraz metod wykorzystywania opisów z zakresu standardu były następujące:

- mechanizmy te mają stanowić obszar swobodnej konkurencji, sprzyjającej rozwojowi technologicznemu;
- standaryzacja algorytmów nie jest konieczna dla możliwości współdziałania różnych systemów;
- charakter i funkcjonalność wyszukiwarek zwykle zależy od zastosowania, a format MPEG-7 ma z założenia być niezależny od zastosowań.

Zakłada się, że odbiorcami opisów MPEG-7 będą zarówno ludzie jak i programy komputerowe oraz inteligentne urządzenia elektroniczne. Opisy MPEG-7 powinny towarzyszyć danym multimedialnym składowanym w bazach danych, ale również udostępnianym przez radio, telewizję, media. Przewidywane zastosowania MPEG-7 to:

- wyszukiwanie informacji w multimedialnych bazach danych;
- filtracja treści multimedialnych odbieranych za pomocą telewizji, radia itp. w oparciu o opisy MPEG-7 przesyłane wraz z zawartością i ustawienia preferencji użytkownika.

Założenia standardu MPEG-7

- Standaryzacja opisu metadanych
- Niezależność od sposobu kodowania medium
- Elastyczność
- Rozszerzalność
- Oparcie o język XML
- Binarny format BiM do kompresji i transmisji opisów

Podstawowym założeniem przyświecającym twórcom standardu MPEG-7 była standaryzacja opisu metadanych. Wcześniej pojawiało się wiele niezgodnych ze sobą propozycji formatów opisów multimediów, MPEG-7 korzysta z ich doświadczeń i normuje format opisu jako powszechnie uznany i oficjalny standard.

MPEG-7 jest całkowicie niezależny od sposobu kodowania i przechowywania medium i może dotyczyć np. obrazów narysowanych na papierze. Jeśli jednak dany format kodowania może zawierać informacje o strukturalnej dekompozycji obiektu multimedialnego (jak np. MPEG-4), informacje te mogą być wykorzystane przy tworzeniu opisów MPEG-7. MPEG-7 jest standardem elastycznym i rozszerzalnym. Jako efekt analizy szerokiego spektrum potencjalnych zastosowań, MPEG-7 jest formatem ogólnego przeznaczenia. Jednocześnie daje on możliwość rozszerzenia składni języka metadanych o elementy przydatne w specyficznych zastosowaniach.
MPEG-7 jest oparty o język XML. Wybór formatu do tekstowych opisów był w zasadzie oczywisty, biorąc pod uwagę elastyczność, prostotę i pozycję na rynku języka XML. Dzięki temu, że twórcy MPEG-7 postanowili nie opracowywać od podstaw nowego formatu, ale oparli się o XML, automatycznie dostępnych jest szereg narzędzi do parsowania i przeszukiwania opisów. Język definicji składni opisów MPEG-7 (Description Definition Language - DDL) został zdefiniowany w oparciu o XML Schema, który jest silnym narzędziem opisu struktury (gramatyki) dokumentów XML.

Mimo niekwestionowanych zalet XML, ma on też pewne wady. XML jest nieodpowiedni dla transmisji strumieniowej, choćby ze względu na nieoszczędny sposób reprezentacji informacji. Dlatego też, w ramach standardu MPEG-7 z myślą o transmisji (szczególnie strumieniowej) opisów opracowany został binarny format opisów MPEG-7 o nazwie BiM. BiM umożliwia bardzo dobrą kompresję opisów XML-owych z zachowaniem pewnych możliwości nawigacji po strukturze opisu na poziomie binarnym.

Zawartość opisów MPEG-7

- Zbudowane z deskryptorów i schematów opisu
- Dekompozycja przestrzenna i czasowa – segmenty, regiony
- Opisy znaczenia zawartości (metadane semantyczne) – opisy zdarzeń, osób, obiektów, czynności
- Niskopoziomowe opisy zawartości (metadane sygnałowe) – kształt, tekstura, kolor, lokalizacja, trajektoria ruchu, melodia
- Informacje uzupełniające (metadane zewnętrzne) – forma, dostępność materiału, klasyfikacja, kontekst

Od strony technicznej opisy MPEG-7 są tworzone za pomocą deskryptorów (reprezentujących poszczególne właściwości obiektów multimedialnych) i schematów opisu (specyfikujących strukturę i semantykę związków między elementami opisu tj. deskryptorami i innymi schematami opisu). Treść dokumentów MPEG-7 oprócz ogólnego opisu całego obiektu multimedialnego, zawiera typowo opisy poszczególnych elementów składowych obiektu, wyodrębnionych w drodze dekompozycji przestrzennej i/lub czasowej. Przykładem dekompozycji czasowej jest podział sekwencji wideo na segmenty. Z kolei dekompozycja przestrzenna może dotyczyć samodzielnych obrazów nieruchomych lub klatek filmu wideo i polega na wskazaniu na obrazie regionów zawierających interesujące obiekty.
Opisy MPEG-7 mogą obejmować wszystkie rodzaje metadanych. Za pomocą MPEG-7 można opisać semantykę obiektu i jego elementów (opisy zdarzeń, osób, obiektów, czynności) oraz własności niskopoziomowe (zależnie od typu danych: kształt, rozmiar, tekstura, kolor, lokalizacja, trajektoria ruchu, tempo, melodia). Opisy typowo obejmują również informacje uzupełniające na temat formy (format, rozmiar, rozdzielczość), dostępności materiału (cena, ochrona praw autorskich), klasyfikacji treści (kategoria tematyczna, kategoria wiekowa odbiorców) i kontekstu (czas i miejsce nagrania, wydarzenie).

44. TODO_Natywne multimedialne bazy danych

Natywne MMBD

- Bardziej dostosowane do specyfiki materiału multimedialnego
- Lepsza wydajność
- Konieczność implementacji mechanizmów relacyjnych
- Transakcyjność, bezpieczeństwo
- SCORE, TVQL, SMDS, CSQL, MOQL

45. Techniki wyszukiwania, składowania i prezentacji danych multimedialnych

Źródło: http://wazniak.mimuw.edu.pl/images/2/2d/ZSBD-2st-1.2-w7.tresc-1.4.pdf

Wyszukiwanie
Zapytania do multimedialnych baz danych odwołują się do:

- metadanych zewnętrznych opisujących obiekty multimedialne
- rzeczywistej zawartości multimediów, reprezentowanej przez metadane semantyczne i sygnałowe.

Kluczowe znaczenie mają algorytmy wyszukiwania w oparciu o zawartość (ang. content-based retrieval) operujące na metadanych sygnałowych.

Ważną rolę w przeszukiwaniu multimedialnych baz danych pełnią interfejsy wizualne, za pomocą których zapytania tekstowe są uzupełnione lub zastąpione przez wybranie, dostarczenie lub naszkicowanie wzorca do przeszukiwania kolekcji danych.

Charakterystyczne dla zapytań do multimedialnych baz danych jest podejście w stylu **Information Retrieval** (IR). Po pierwsze, użytkownik powinien mieć możliwość przypisywania wag poszczególnym kryteriom selekcji np. odwołującym się do koloru i tekstury. Po drugie, zakłada się, że wyszukiwanie ma charakter procesu iteracyjnego, w którym użytkownik modyfikuje parametry zapytania aż do uzyskania satysfakcjonującego rezultatu. Jest to uzasadnione, gdyż np. przy wyszukiwaniu obiektów podobnych do zadanego, trudno jest za pierwszym razem dobrać odpowiednie wagi i próg dla miary podobieństwa. Po trzecie, tolerowane są niekompletne wyniki, w których nie znalazły się niektóre obiekty spełniające zadane kryteria, a być może pojawiły się obiekty tych kryteriów nie spełniające. Takie podejście jest znacząco odmienne od klasycznych zapytań do baz danych, ale w sytuacji gdy zapytania odwołują się niekiedy do subiektywnego odczucia podobieństwa, trudno o obiektywną kompletność wyników wyszukiwania.

Wyszukiwanie w oparciu o zawartość (ang. content-based retrieval) polega na wyszukiwaniu w oparciu o automatycznie wyekstrahowane metadane sygnałowe. Są to metadane reprezentujące takie właściwości jak:
(a) dla obrazów: średni kolor, udział i lokalizacja kolorów, tekstura;
(b) dla danych audio: melodia, rytm;
(c) dla danych wideo: dynamika ruchu, właściwości obrazów stanowiących klatki filmu.
Wyszukiwanie w oparciu o zawartość jest najlepiej dopracowane dla obrazów.

Wyszukiwanie w oparciu o zawartość może stanowić alternatywę lub uzupełnienie dla wyszukiwania w oparciu o alfanumeryczne opisy wprowadzone „ręcznie". Zaletą wyszukiwania w oparciu o zawartość jest to, że tworzenie alfanumerycznych opisów zawartości danych multimedialnych jest czasochłonne, a na razie nie może być wykonane bez udziału człowieka. Ponadto, nie wszystkie właściwości można w sposób naturalny opisać słownie, np. tekstury i kształty na obrazach. Wyszukiwanie w oparciu o zawartość ma szereg zastosowań. Przede wszystkim są to wszelkie interfejsy do zapytań poprzez przykład, a z zastosowań ściśle praktycznych – wykrywanie plagiatów utworów muzycznych, zastrzeżonych znaków towarowych, itp.

Zapytania do multimedialnych baz danych można pogrupować wg typu metadanych, do których się odwołują.

Przykłady zapytań odwołujących się do metadanych zewnętrznych to:

(a) „Znajdź wszystkie utwory wykonawcy X dostępne w formacie MP3”;
(b) „Znajdź wszystkie komedie reżysera X o czasie trwania poniżej 1,5h”.

Przykłady zapytań odwołujących się do znaczenia, realizowanych w oparciu o metadane semantyczne to:
(a) „Znajdź wszystkie przemówienia na temat bezrobocia”;
(b) „Znajdź wszystkie fotografie przedstawiające prezydenta z premierem”.

Przykłady zapytań odwołujących się do zawartości, realizowanych w oparciu o metadane sygnałowe to:
(a) „Znajdź wszystkie obrazy podobne do narysowanego (naszkicowanego)”;
(b) „Znajdź wszystkie utwory podobne do zanuconego do mikrofonu”;
(c) „Znajdź wszystkie filmy z aktorem przedstawionym na zdjęciu”;
(d) „Znajdź wszystkie obrazy, w których dominuje kolor czerwony”.

Jak wspomniano wcześniej, technika wyszukiwania w oparciu o zawartość w multimedialnych bazach danych jest w chwili obecnej najlepiej dopracowana i dostępna dla obrazów. Typowe właściwości obrazów wykorzystywane przy wyszukiwaniu obrazów w oparciu o zawartość to:
Średni kolor – wyznaczany w oparciu o podział obrazu na n próbek i niezależne uśrednianie komponentów RGB koloru;
Histogram kolorów – reprezentujący udział procentowy kolorów w obrazie, najczęściej wyznaczany po zredukowaniu liczby kolorów np. do 256;
Kształty - odkrywane w procesie tzw. segmentacji obrazu, gdzie segmenty są rozumiane jako spójne regiony o tym samym kolorze;
Tekstury - opisujące chropowatość, kontrast, kierunkowość wzorców wypełniających kształty i tło obrazu; reprezentowane w postaci wektorów czy macierzy współczynników liczbowych, wyznaczanych specjalistycznymi algorytmami;
Lokalizacja kolorów, kształtów i tekstur na obrazie – wyznaczana poprzez podział obrazu na siatkę regionów i następnie wyznaczenie dominujących kolorów, kształtów i tekstur dla każdego regionu.

Kryteria wyszukiwania obrazów ze względu na zawartość typowo są podawane w formie przykładowego obrazka. Wyszukiwanie sprowadza się do znalezienia obrazów podobnych do podanego przykładu. Często użytkownik ma możliwość wskazania, które właściwości wizualne mają być uwzględnione w testach podobieństwa z opcją przypisania im wag. Dostarczenie systemowi wzorcowego obrazu może polegać na:
(a) wyborze z predefiniowanego zbioru obrazków reprezentujących poszczególne kategorie,
(b) załadowaniu do systemu obrazka dostarczonego przez użytkownika,
(c) naszkicowaniu obrazu za pomocą prostego, wbudowanego w interfejs edytora graficznego.
Edytor ten może być zorientowany na konkretną dziedzinę zastosowań i udostępniać pewne predefiniowane kształty.

Składowanie

Składowanie danych stanowi istotny problem dla danych multimedialnych o szczególnie dużym rozmiarze, czyli przede wszystkim filmów wideo. Pierwszym ograniczeniem jakie może napotkać system składowania danych wideo jest ograniczona pamięć dyskowa. Tradycyjnie problem ten był rozwiązywany przez wykorzystanie hierarchicznych struktur składowania. Idea polegała na przechowywaniu najczęściej żądanych filmów na dyskach magnetycznych (dostępnych on-line), a wszystkich filmów na nośnikach typu dyski optyczne czy taśmy (dostępnych off-line), udostępnianych przez urządzenia typu jukebox albo archiwa obsługiwane przez roboty. Dane zawsze były serwowane z pamięci dyskowej, co pociągało za sobą konieczność migracji danych z pamięci off-line do on-line w przypadku żądania mało popularnego

filmu.

Drugim problemem, który miał znaczenie przede wszystkim w przeszłości, była zbyt mała szybkość transmisji danych z dysków. Rozwiązania tego problemu to:

- **replikacja -** polega na duplikowaniu zawartości poszczególnych dysków, dzięki czemu kolejne fragmenty filmu mogą być równolegle odczytywane z kolejnych dysków.
- **striping -** technika umożliwiająca osiągnięcie tego samego efektu, ale bez duplikowania danych i marnowania przestrzeni dyskowej. W tym wypadku dane rozmieszczane są na kolejnych dyskach w formie tzw. pasków (ang. stripe). Fragmenty jednego paska mogą być odczytywane równolegle, zwiększając przepustowość transmisji danych filmu z macierzy dyskowej maksymalnie tyle razy, ile dysków obejmuje pasek.

Prezentacja

Optymalizacja strumieniowej transmisji dla wielu klientów sprowadza się przede wszystkim do redukcji liczby strumieni wykorzystywanych do transmisji tego samego filmu.
Zaproponowano w tym zakresie następujące techniki:

- **Batching** – polega na odsunięciu w czasie rozpoczęcia transmisji – być może nadejdą identyczne żądania i grupę żądań obsłuży jeden strumień;
- **Bridging** – dla strumieni dotyczących tego samego filmu, nieznacznie przesuniętych w czasie, polega na przechowywaniu danych z okna czasowego między strumieniami w pamięci cache;
- **Piggybacking** – dla strumieni dotyczących tego samego filmu, nieznacznie przesuniętych w czasie, polega na przyspieszeniu drugiego filmu (poprzez pominięcie niektórych klatek) lub spowolnieniu pierwszego (poprzez wstawienie klatek interpolowanych) tak aby oba żądania były od pewnego momentu obsługiwane przez jeden strumień (zaobserwowano, że wahania prędkości transmisji poniżej 5% są niezauważalne dla człowieka).

46. TODO_Manipulacja multimedialnymi danymi

47. Standardy mapowania relacyjno-obiektowego

http://zeszyty-naukowe.wwsi.edu.pl/zeszyty/zeszyt4/Mapowanie_Obiektowo-Relacyjne_ORM_-_Czy_Tylko_Dobra_Idea.pdf

Do niedawna rozwiazania ORM nie tworzyły spójnej grupy narzedzi. Jedyne uznawanym standardem była warstwa, na której one bazuja, czyli interfejs dostepu do relacyjnych baz danych JDBC. Zaproponowany przez firme Sun standard JDO (ang. Java Data Objects) nie został zaakceptowany przez rynek oprogramowania komercyjnego. Istniejacy w ramach Java Enterprise standard Enterprise Java Beans 2 z obiektami entity beans, ze wzgledu na niezwykle skomplikowany sposób wykorzystania, nie osiagna dominujacej pozycji. Dopiero wersja 3.0 przyniosła znaczaca zmiane. EJB wydzieliło zagadnienie utrwalania obiektów jako osobna specyfikacje — JPA.
Na rynku istniały juz narzedzia o ugruntowanej pozycji takie jak iBatis czy Cayenne. Jednakze jest cos co sprawia iz framework bedacy tematem niniejszej pracy jest wyjatkowy. Specyfikacja EJB w duzej mierze czerpie z doswiadczen Hibernate. Przy jej tworzeniu brał udział twórca Hibernate — Gavin King. JPA udostepnia podobne metody konfiguracji jak Hibernate a takze w obu rozwiazaniach zastosowano zwykłe

obiekty jezyka Java — POJO.
Hibernate był równoczesnie jedna z pierwszych implementacji standardu JPA. Pomimo tak bliskich relacji omawiany framework do ORM zachował swoja unikalnosc.
W przeciwienstwie do JPA, osoba posługujaca sie Hibernate ma pełna kontrole nad momentem odłaczenia oraz połaczenia obiektu z baza danych. Rózni sie tez interfejs podstawowych obiektów udostepniajacych API do persystencji danych. Pod wieloma wzgledami Hibernate stanowi takze rozszerzenie standardowych opcji dostepnych w JPA.
Aby zachowac zgodnosc ze standardem wchodzacym w skład EJB 3.0 a takze nie utracic swojego indywidualnego podejscia do ORM, Hibernate umozliwia wykorzystanie jego funkcjonalnosci na dwa sposoby. Pierwszy polega na wykorzystaniu klasycznych obiektów Configuration, SessionFactory oraz Sessions. Konfiguracja moze odbywac sie za pomoca dokumentów XML9 lub opcjonalnie adnotacji.
Drugie rozwiazanie preferuje umieszczane w kodzie metadanych opisujacych szczegóły mapowania. Jako podstawowy obiekt udostepniajacy usługi utrwalania korzysta sie tu z EntityManagera. W takiej sytuacji niezbedne jest, poza główna czescia Hibernate czyli Hibernate Core, uzycie modułów Hibernate Adnotation oraz Hibernate EntityManager.
W niniejszej pracy omówiono podejscie oparte na klasycznym interfejsie. Daje ono najwieksze mozliwosci a do tego nie wymaga dodawania kolejnych bibliotek.

48. Obiektowe rozszerzenia relacyjnych baz danych

(nie mój temat, teoretycznie dalej TODO przez odpowiedzialną osobę)
Zakładając, że posiadamy OODB i wykorzystujemy ją do utrwalania obiektów z warstwy aplikacji (napisanej w obiektowym języku programowania), wówczas takie połączenie ma następujące zalety i możliwości:

- Bogate środki modelowania danych.
- **Rozszerzalność**.
- **Brak niezgodności impedancji.**
- Język zapytań o większej ekspresyjności.
- Wsparcie dla zmian schematu.
- Wsparcie długich transakcji.
- Złożone aplikacje.

Niestety OODB nie są pozbawione **wad**, przykładowo:

- Brak powszechnego, jednolitego modelu danych jak w RDB.
- Brak standaryzacji.
- Często niższa wydajność.
- Większa złożoność.
- Brak perspektyw (view).
- **Trudne utrzymanie integralności danych. – największy problem.**

Obiektowe rozszerzenia relacyjnych baz danych są pewnym kompromisem pomiędzy bazami obiektowymi a relacyjnymi – jest to rozszerzenie modelu relacyjnego o własności obiektowe. Obecnie obiektowe są tylko niszowo wykorzystywane, w praktyce używa się RDB z

rozszerzeniami obiektowymi.

SQL:1999 (SQL-99, podzbiór SQL3) - wprowadzenie własności obiektowych:

- Konstruktory typów pozwalające na definiowanie obiektów złożonych: struktur i kolekcji,
- User Defined Types (atrybuty, metody, enkapsulacja, dziedziczenie – wszystko może podać programista, są definicjami klas) – można wykorzystać jako typ relacji lub typ atrybutu w relacji lub innym obiekcie,
- Tożsamość obiektów: typ REF i związki pomiędzy obiektami.

W SQL-99 dostęp do wartości atrybutu jest w notacji kropkowej. Dla atrybutów typu referencyjnego operator dereferencji. W Oracle jest to operator DEREF.

Każdy UDT może definiować dwie funkcje porównujące EQUAL i LESSTHAN. Zwracają TRUE lub FALSE. Pozwala na porządkowanie np. w ORDER BY.

Oracle jest jednym z SZBD, który posiada obiektowe rozszerzenia. W Oracle typ deklaruje się CREATE TYPE T AS OBJECT (lista atrybutów i metod); Implementacja metod w CREATE TYPE BODY. Do porządkowania można przesłonić metodę ORDER. Dziedziczenie odbywa się za pomocą słowa kluczowego UNDER. Można przeciążać metody (polimorfizm).

Oracle umożliwia tworzenie zagnieżdżonych tabel, wartością atrybutu dla danego wiersza może być relacja. Zagnieżdżanie to jest możliwe przy tworzeniu relacji poleceniem CREATE TABLE. Wówczas operatorem NESTED TABLE wskazuje się, który atrybut będzie przechowywał relację.

Dodatkowo Oracle definiuje dwa operatory, operator TABLE, który umożliwia spłaszczanie kolekcji relacji oraz operator MULTISET, który wiersze zwrócone przez operator SELECT umieszcza w pojedynczej kolekcji – obiekcie.

49. Obiektowe bazy danych

Obiektowa baza danych jest zbiorem obiektów, których zachowanie się i stan oraz związki są określone zgodnie z obiektowym modelem danych. Obiektowy system zarządzania bazą danych (OSZBD) jest systemem wspomagającym definiowanie, zarządzanie, utrzymywanie, zabezpieczanie i udostępnianie obiektowej bazy danych.

Systemy obiektowych baz danych były rozwijane w celu dostarczenia elastycznego modelu danych bazującego na tym samym paradygmacie, co obiektowe języki programowania. Obiektowe bazy danych dają możliwość silniejszego powiązania z aplikacjami obiektowymi, niż było to w przypadku relacyjnych baz danych. Dzięki temu można zminimalizować ilość operacji związanych z przechowywaniem i dostępem do danych zorientowanych obiektowo. Zaleta ta staje się szczególnie cenna w przypadku, gdy obiektowy model danych jest naprawdę skomplikowany.

Obiektowe systemy zarządzania bazą danych zapewniają tradycyjną funkcjonalność baz danych, lecz bazują na modelu obiektowym.

Do klasycznych funkcji obsługiwanych przez OSZBD można zaliczyć:

- zarządzanie pamięcią zewnętrzną,

- zarządzanie schematem,
- sterowanie współbieżnością,
- zarządzanie transakcjami,
- odtwarzalność,
- przetwarzanie zapytań,
- kontrolę dostępu.

Do przedstawionych powyżej funkcji, OSZBD dokładają dodatkowo:

- złożone obiekty,
- typy definiowane przez użytkownika,
- tożsamość obiektów,
- hermetyzację,
- typy i/lub klasy oraz ich hierarchię,
- przesłanianie, przeciążanie, późne wiązanie,
- kompletność obliczeniową (pragmatyczną).

Poza bezsprzecznymi zaletami OSZBD, istnieje wiele powodów powstrzymujących ich dynamiczny rozwój i powodujących, iż systemy relacyjnych baz danych wciąż posiadają znacznie większy udział w rynku. Do powodów tych zaliczyć można:

- *Niedojrzałość technologii* - wiele organizacji wykorzystuje obecnie OSZBD do aplikacji o mniejszym znaczeniu, z uwagi na ryzyko wiążące się z wykorzystaniem nowej technologii.
- *Niestabilność producentów* - producenci OSZBD wydają się być relatywnie małymi firmami, co powstrzymuje niektóre organizacje przed inwestowaniem w te systemy z obawy, że w niedalekiej przyszłości firmy te mogą zniknąć z rynku.
- *Brak wykwalifikowanego personelu* - niestety wciąż brakuje informatyków wykwalifikowanych w obsłudze obiektowych systemów zarządzania bazami danych.
- *Koszty konwersji* - niebagatelną przeszkodę stanowią koszty konwersji na nowy system. Należy tu wziąć pod uwagę koszty nowego oprogramowania i jego instalacji, a także w przypadku firm już istniejących na rynku, inwestycje poczynione dotychczas w systemy relacyjnych baz danych.

Trwałość obiektu

Trwałość obiektu (ang. *object persistence*) jest jedną z najważniejszych usług dostarczanych przez obiektowe systemy zarządzania bazami danych. Przez trwałość rozumie się zachowanie polegające na tym, że obiekt jest stale dostępny, a jego stan pozostanie niezmieniony pomiędzy kolejnymi wywołaniami.

Obiekty, które nie są stale dostępne nazywane są ulotnymi. Zależnie od aplikacji, niektóre obiekty muszą zmieniać swój stan z ulotnego na trwały, a zadaniem OSZBD jest zarządzanie tą konwersją. OSZBD przyporządkowuje każdemu obiektowi unikalny identyfikator, który jest wykorzystywany głównie do ustalenia powiązań pomiędzy trwałymi obiektami. OSZBD posiada procedury odzyskiwania zapewniające, że trwałe obiekty przetrwają wszelkie awarie systemu.

Istnieją dwie metody wykorzystywane przez OSZBD w celu uzyskania dostępu do trwałych obiektów. Są to wirtualne wskaźniki adresu pamięci oraz tablice z kodowaniem mieszającym.

Trwałe obiekty są stale gotowe do wywołania. Są one przechowywane na dysku, natomiast obiekty ulotne istnieją w pamięci RAM. W OSZBD obiekty mogą przechodzić z jednego stanu w drugi, mając swoją reprezantację w pamięci RAM, a także na dysku.

Poniżej przedstawiono niektóre z działań, które OSZBD musi wykonywać w związku z zarządzaniem trwałością obiektów:

- wyszukiwanie obiektów na dysku i przyporządkowywanie im nowych identyfikatorów w pamięci RAM,
- przyporządkowywanie i alokowanie przestrzeni w składzie trwałych obiektów,
- automatyczne zwiększanie przestrzeni w razie potrzeby,
- zarządzanie wolną przestrzenią w składzie trwałych obiektów,
- interfejs pomiędzy systemem wejścia-wyjścia a składem trwałych obiektów,
- zapewnienie, że trwałe obiekty mogą być odzyskane,
- zarządzanie buforami bazy danych,
- zarządzanie mapowaniem pomiędzy fizycznymi adresami a identyfikatorami obiektów.

Architektura

Istnieje kilka koncepcji odnośnie architektury obiektowego systemu zarządzania bazą danych. Najbardziej abstrakcyjną jest architektura zaproponowana przez komitet ANSI/SPARC. Wyróżnia ona trzy poziomy:

- *poziom pojęciowy* systemu, wspólny dla wszystkich jego użytkowników,
- *poziom zewnętrzny*, specyficzny dla konkretnego użytkownika,
- *poziom fizyczny*, który odnosi się do implementacji bazy danych.

Kolejnym rodzajem jest architektura klient-serwer, gdzie występuje podział na dwie części: serwer bazy danych, wykonujący przykładowo wyrażenia SQL wysyłane przez klientów oraz druga część, którą stanowi jeden lub kilku klientów wysyłających żądania do serwera.

Bardziej zaawansowane są architektury trzywarstwowa oraz wielowarstwowa. Jak sama nazwa wskazuje, architektura trzywarstwowa charakteryzuje się podziałem na trzy warstwy: interfejs użytkownika, logikę przetwarzania oraz bazę danych. Warstwy te są zaprojektowane i istnieją niezależnie, co ma duże znaczenie dla utrzymania całego systemu ze względu na możliwość zmian w dowolnej warstwie bez konieczności interwencji w pozostałych warstwach. Warstwa środkowa może być złożona z kilku warstw i mamy wówczas do czynienia z architekturą wielowarstwową.

Obiektowo-relacyjne bazy danych

Obiektowo-relacyjne bazy danych są najnowszym osiągnięciem w rozwoju hybrydowych architektur baz danych. Systemy te pojawiły się między innymi z powodu ogromnych inwestycji poczynionych przez różne organizacje w systemy relacyjnych baz danych. Bezpośrednie przekwalifikowanie z relacyjnych na obiektowe systemy baz danych wiązałoby się z dodatkowymi kosztami, natomiast obiektowo-relacyjne bazy danych stanowią swoisty kompromis.

Obiektowe bazy danych nie posiadają niektórych cech, do których przywykli użytkownicy poprzednich systemów. Dodatkowo, w chwili obecnej obiektowe systemy baz danych nie mają odpowiedniej infrastruktury, aby przejąć rynek relacyjnych baz danych, podobnie jak bazy relacyjne uczyniły to z systemami hierarchicznymi i sieciowymi.

Rozwiązaniem dla firm takich jak Oracle, Informix, Sybase, czy IBM jest rozwój ich systemów polegający na przekształcaniu ich w systemy obiektowo-relacyjnych baz danych.

Systemy obiektowo-relacyjne są wyposażane w wiele cech umożliwiających efektywną produkcję aplikacji. Wśród nich można wymienić przystosowanie do multimediów (duże obiekty BLOB, CLOB i pliki binarne), dane przestrzenne, abstrakcyjne typy danych (ADT), metody (funkcje i procedury) definiowane przez użytkownika w różnych językach, kolekcje (zbiory, wielozbiory, sekwencje, zagnieżdżone tablice, tablice o zmiennej długości), typy referencyjne, przeciążanie funkcji, późne wiązanie i inne. Systemy te zachowują jednocześnie wiele technologii, które sprawdziły się w systemach relacyjnych (takie jak architektura klient/serwer, mechanizmy buforowania i indeksowania, przetwarzanie transakcji, optymalizacja zapytań)
Obiektowo-relacyjne bazy danych zdobywają uznanie, gdyż wiele organizacji dostrzega, że relacyjne bazy danych nie są wystarczające do obsługi ich złożonych wymagań. Zamiast więc bezpośredniego przejścia na systemy czysto obiektowe, podejście obiektowo-relacyjne pozwala organizacjom na zapoznanie się z technologią obiektową w rozsądnym tempie. Dodatkową zaletą jest uniknięcie konwersji aktualnych baz danych do nowego, obiektowego formatu danych, co oszczędza czas i pieniądze. Obiektowo-relacyjne bazy danych stanowią zatem pomost pomiędzy relacyjnymi a obiektowymi bazami danych.

50. TODO_Języki opisu i manipulacji danymi oraz ich zastosowania dla wybranych modeli danych

Nic poza (jakaś wersja WWW tej prezentacji lub na odwrót) tą prezentacją (do 10 slajdu nie znalazłem). Nie ma tego jak streścić - Maciej
http://chomikuj.pl/m.szpaner/Dokumenty/Politechnika+wip/Semestr+I/Bazy+danych/J*c4*99zyki+manipulacji+danymi.+J*c4*99zyki+zapyta*c5*84.+Optymalizacja+zapyta*c5*84,1613773186.pptx