

Systemy rozproszonych baz danych – 1

Problematyka rozproszonych baz danych

Wykład przygotował:

Robert Wrembel



Plan wykładu

- Wprowadzenie do problematyki
- Definicja rozproszonej bazy danych
- 12 reguł Date
- Podstawowa architektura systemu rozproszonej bazy danych

ZSBD – wykład 1 (2)

Celem pierwszego wykładu jest wprowadzenie do problematyki rozproszonych baz danych. Zostaną tu omówione:

- definicja rozproszonej bazy danych,
- dwanaście uznanych reguł zdefiniowanych przez C.J.Date, które powinien spełniać system rozproszonej bazy danych,
- podstawową architekturę sfederowanej bazy danych, jak przykład architektury implementacyjnej.



Wprowadzenie

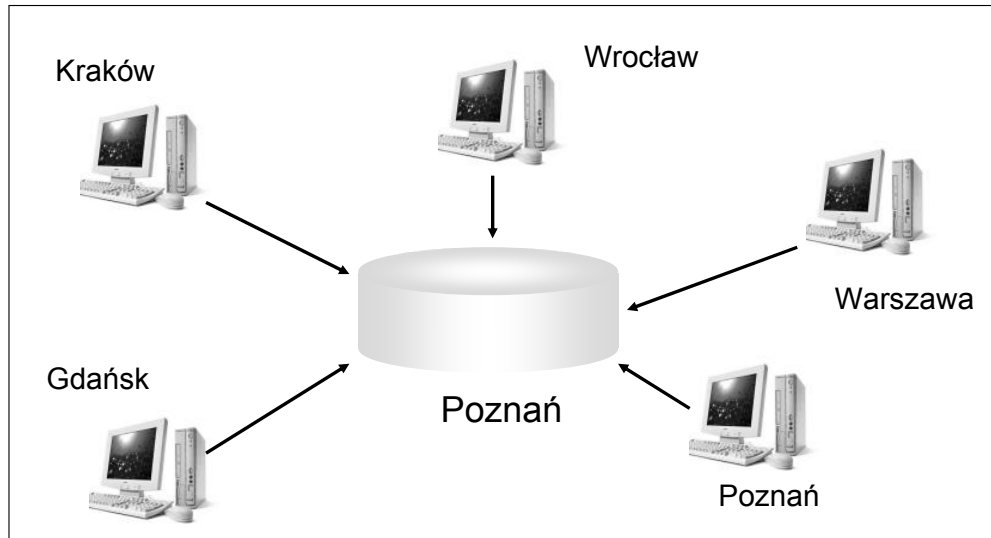
- Typowa architektura systemu informatycznego
 - scentralizowana baza danych
 - aplikacje klient-serwer lub w architekturze 3-warstwowej
- Scentralizowane bazy danych nie zawsze oferują wymaganą funkcjonalność i zadowalającą efektywność
⇒ rozproszone bazy danych (RBD)

ZSBD – wykład 1 (3)

W typowych zastosowaniach systemów baz danych wykorzystuje się architekturę scentralizowaną, w której system zarządzania bazą danych (SZBD) i wszystkie dane znajdują się w tym samym węźle sieci informatycznej. Dostęp do takiej bazy danych jest realizowany albo za pomocą aplikacji pracujących w architekturze klient-serwer albo pracujących w architekturze 3-warstwowej. Istnieje jednak wiele zastosowań, w których scentralizowane bazy danych nie zapewniają wymaganej funkcjonalności i efektywności pracy. W takich przypadkach, stosuje się tzw. rozproszone bazy danych.



Przykład - scentralizowana BD

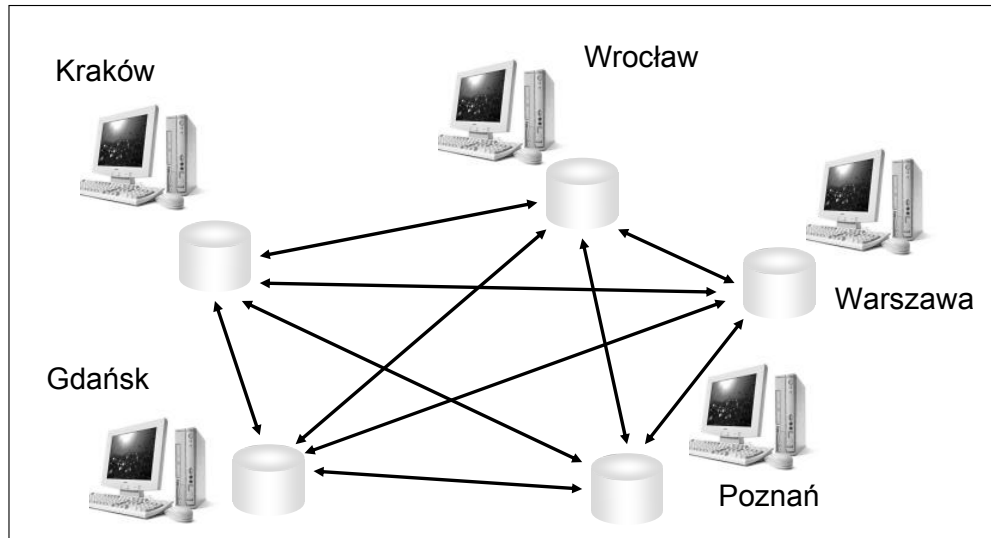


ZSBD – wykład 1 (4)

Przykładowo, rozważmy sieć dużych warsztatów samochodowych w Polsce, z kilkoma oddziałami w każdym dużym mieście. Gdyby zbudować system informatyczny dla tej sieci oparty o scentralizowaną bazę danych umieszczoną np. w Poznaniu, wówczas każde odwołanie do tej bazy z innego miasta wymagałoby transmisji sieciowej. Przy sieci o niskiej przepustowości i dużej częstotliwości odwołań, poprawne wykorzystywanie systemu stałoby się niemożliwe. Dodatkowo, taki system byłby znacznie bardziej podatny na awarie niż system rozproszony. Awaria serwera scentralizowanej bazy danych powodowałaby niemożliwość korzystania z systemu we wszystkich oddziałach firmy!



Przykład - rozproszona BD



ZSBD – wykład 1 (5)

Alternatywnym rozwiązaniem do przedstawionego poprzednio jest zastosowanie wielu lokalnych baz danych, np. po jednej w każdym dużym mieście, czyli tzw. systemu rozproszonych baz danych. Każda z tych baz lokalnych przechowywałaby informacje o klientach z danego regionu.



Rozproszona baza danych (1)

- Zbiór współpracujących z sobą baz danych (lokalne bd)
- Każda z baz lokalnych znajduje się na innym serwerze
- Z punktu widzenia użytkownika bazy lokalne logicznie stanowią jedną bd

ZSBD – wykład 1 (6)

Rozproszona baza danych jest zbiorem współpracujących z sobą baz danych, z których każda znajduje się na innym serwerze. Bazy te dalej będziemy nazywali lokalnymi. Z punktu widzenia użytkownika wszystkie te bazy logicznie stanowią jedną rozproszoną bazę danych.



Rozproszona baza danych (2)

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

ZSBD – wykład 1 (7)

Dzięki umieszczeniu danych „blisko” ich użytkowników, skraca się opóźnienia transmisji sieciowej ponieważ dane specyficzne dla węzła są składowane i przetwarzane lokalnie. Dodatkowo, zmniejsza się ryzyko utraty wszystkich danych na skutek awarii systemu i wzrasta niezawodność całego systemu, ponieważ awaria jednej bazy danych np. w Krakowie nie ma wpływu na bazy danych w pozostałych miastach tak długo, dopóki zapytania nie są kierowane do bazy w Krakowie.



Rozproszona baza danych (3)

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

ZSBD – wykład 1 (8)

Architektura rozproszonych baz danych ma dwie podstawowe wady. Po pierwsze, rozproszenie danych utrudnia dostęp do pełnego-zintegrowanego zbioru danych pochodzących z różnych baz i ich analizę. Przykładowo, zarząd sieci warsztatów samochodowych będzie zainteresowany zestawieniami ilości sprzedaży i usług zrealizowanych w poszczególnych warsztatach. Uzyskanie takich informacji wymaga zintegrowania danych pochodzących ze wszystkich baz danych firmy.

Po drugie, wszystkie warsztaty korzystają z pewnego wspólnego zbioru informacji, tzw. słowników. Przykładem takiego słownika jest wykaz części znajdujących się w sprzedaży wraz z ich aktualnymi cenami. Gdyby dane słownikowe były przechowywane centralnie, tj. w jednej bazie danych, wówczas powstawałyby omówione wcześniej problemy architektury scentralizowanej. W związku z tym, informacje słownikowe są najczęściej powielane w każdej bazie danych firmy. Są to tzw. repliki.

W przypadku replik, występuje problem utrzymywania ich aktualnej zawartości, w przypadku, gdy oryginalne dane słownikowe ulegają modyfikacjom. Przykładowo, zmiana ceny opony w centrali firmy musi być propagowana do wszystkich oddziałów.



Komponenty architektury

- Sprzętowe komponenty rozproszonej bazy danych (RBD)
 - węzły - komputery, na których działa lokalna bd
 - sieć komputerowa
- Programowe komponenty RBD
 - protokoły sieciowe, np. TCP/IP, IPX/SP, LU6.2, DEC Net
 - dedykowane oprogramowanie realizujące dostęp z jednej bazy danych do drugiej

ZSBD – wykład 1 (9)

W skład systemu rozproszonej bazy danych wchodzi komponenty sprzętowe i programowe. Do pierwszej grupy zalicza się:

- tzw. węzły, czyli komputery na których działają lokalne bazy danych
- sieć komputerowa, dzięki której poszczególne węzły mogą się z sobą komunikować, a użytkownik z dowolnego węzła może sięgnąć do dowolnych innych węzłów systemu.

Do grupy programowych komponentów zalicza się:

- protokoły sieciowe np. TCP/IP, IPX/SPX, LU6.2, DEC Net,
- dedykowane oprogramowanie umożliwiające dostęp z jednej bazy danych do innej i przetwarzanie danych z innej bazy tak, jakby dane te były przechowywane lokalnie.



Reguły Date (1)

1. Lokalna autonomia
2. Uniezależnienie od centralnego miejsca
3. Działanie ciągłe
4. Niezależność lokalizacji
5. Niezależność fragmentacji
6. Replikacja
7. Niezależność sprzętowa

ZSBD – wykład 1 (10)

C.J.Date zaproponował 12 reguł jakie powinien spełniać system rozproszonej bazy danych. W dalszej części wykładu zostaną omówione te reguły.



Reguły Date (2)

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



1. Lokalna autonomia

- Każdy węzeł RBD jest zarządzany niezależnie od pozostałych węzłów systemu
- Wszystkie operacje na danych w węźle są kontrolowane przez ten węzeł
- Działanie węzła X nie powinno zależeć od działania węzła Y
- Na każdym węźle działa niezależny system zarządzania bazą danych

ZSBD – wykład 1 (12)

Lokalna autonomia oznacza, że:

- każdy węzeł należący do rozproszonej bazy danych jest zarządzany (administrowany) niezależnie od pozostałych węzłów,
- wszystkie operacje na danych w węźle są kontrolowane przez ten węzeł,
- działanie węzła X nie powinno zależeć od działania lub niedziałania innych węzłów,
- na każdym węźle działa niezależny system zarządzania bazą danych.



2. Uniezależnienie od centralnego węzła

- Wszystkie węzły są traktowane jednakowo
- Nie ma wyróżnionego centralnego węzła usługi (np. przetwarzanie zapytań)
 - mógłby stanowić wąskie gardło całego systemu

ZSBD – wykład 1 (13)

Uniezależnienie od centralnego węzła oznacza, że: wszystkie węzły systemu rozproszonej bazy danych są traktowane jednakowo i nie ma wyróżnionego węzła oferującego usługi dla pozostałych węzłów, np. przetwarzania zapytań. Taki wyróżniony węzeł mógłby stanowić tzw. "wąskie gardło" całego systemu.



3. Działanie ciągłe

- System RBD jest bardziej odporny na awarie
 - awaria jednego węzła nie wpływa na pracę innych (autonomia)
 - dzięki replikacji danych inny węzeł może udostępnić dane węzła uszkodzonego

ZSBD – wykład 1 (14)

Działanie ciągłe oznacza, że:

- awaria jednego węzła nie wpływa na pracę innych węzłów (jest to zagwarantowane przez autonomię węzłów),
- dzięki zastosowaniu mechanizmu replikowania danych do wielu węzłów, inny węzeł może udostępnić replikę oryginalnych danych w przypadku awarii węzła przechowującego dane oryginalne.

Uwaga: mechanizm replikacji zostanie omówiony w wykładzie drugim.



4. Niezależność lokalizacji

- Sposób dostępu do danych powinien być jednakowy
 - niezależny od fizycznego umiejscowienia sposobu składowania danych
- Użytkownik nie powinien być świadomym fizycznego umiejscowienia danych (przezroczystość lokalizacji)
 - powinien mieć wrażenie, że dane są przechowywane lokalnie

ZSBD – wykład 1 (15)

Niezależność lokalizacji oznacza, że sposób dostępu do danych przechowywanych w węzłach systemu RBD powinien być jednakowy, niezależny od fizycznego umiejscowienia danych i niezależny od ich fizycznego sposobu składowania. Ponadto, system powinien zapewniać tzw. przezroczystość lokalizacji (ang. location transparency), czyli ukrywać przed użytkownikiem fizyczne miejsce składowania danych. Innymi słowy, przezroczystość lokalizacji gwarantuje, że użytkownik nie musi znać fizycznego umiejscowienia danych a dostęp do danych jest realizowany w taki sposób, jak gdyby dane były przechowywane lokalnie.



5. Niezależność fragmentacji

- Dane można dzielić na fragmenty
- Każdy fragment można umieścić w dowolnym węźle
- Użytkownik nie powinien być świadomym istnienia fragmentów i ich lokalizacji
- Dostęp do fragmentu jest jednakowy i nie zależy od lokalizacji

ZSBD – wykład 1 (16)

Niezależność fragmentacji oznacza, że:

- dane, np. tabelę lub indeks, można dzielić na fragmenty,
- każdy fragment można niezależnie umieścić w innym węźle systemu RBD,
- użytkownik nie powinien być świadomym fizycznej lokalizacji fragmentów, często nie powinien też być świadomym istnienia fragmentów,
- dostęp do fragmentów powinien być jednakowy, niezależny od lokalizacji.

Uwaga: mechanizm fragmentacji zostanie omówiony w wykładzie drugim.



6. Replikacja

- Mechanizm tworzenia kopii danych pochodzących z jednego węzła w innym węźle
- Użytkownik może operować zarówno na danych oryginalnych, jak i na ich kopii w taki sam sposób (bez ograniczeń)

ZSBD – wykład 1 (17)

Replikacja jest mechanizmem polegającym na tworzeniu kopii danych pochodzących z jednego węzła w innym węźle. Użytkownik może operować w taki sam sposób na danych oryginalnych (źródłowych), jak i na kopii danych. Podstawowym obiektem bazy danych, który się replikuje jest tabela.



7. Niezależność sprzętowa

- Możliwość korzystania z tego samego SZBD na różnych platformach sprzętowych i ich współpraca w jednym systemie rozproszonym

ZSBD – wykład 1 (18)

Niezależność sprzętowa oznacza, że ten sam system zarządzania bazą danych (pochodzący od jednego producenta, w jednej wersji) może zostać zainstalowany na różnych platformach sprzętowych, z których każda stanowi osobny węzeł. SZBD działający na różnych platformach sprzętowych może wejść w skład tego samego systemu RBD.



8. Niezależność od systemu operacyjnego

- Możliwość korzystania z tego samego SZBD w różnych systemach operacyjnych
- Przykład
 - Oracle10g Release 2
 - z/Linux
 - z/OS
 - Solaris (x86-64)
 - HP-UX Itanium
 - Linux Itanium
 - Microsoft Windows
 - AIX5L

ZSBD – wykład 1 (19)

Niezależność od systemu operacyjnego oznacza, że ten sam system zarządzania bazą danych (pochodzący od jednego producenta, w jednej wersji) może zostać zainstalowany w różnych systemach operacyjnych i może wejść w skład tego samego systemu RBD.

Przykładowo, SZBD Oracle10g Release 2 może zostać zainstalowany m.in. w systemie operacyjnym z/Linux, z/OS, Solaris (x86-64), HP-UX Itanium, Linux Itanium, Microsoft Windows, AIX5L. Każda z tych instalacji może wchodzić w skład tego samego systemu RBD.



9. Niezależność od SZBD

- W skład systemu RBD mogą wchodzić bazy danych zarządzane przez różne SZBD
 - np. Oracle10g, IBM DB2, MS SQLServer2005, Sybase Adaptive Server Enterprise
- Dostęp do tych baz danych powinien być jednolity
 - jednolity/ustandaryzowany interfejs dostępu

ZSBD – wykład 1 (20)

Niezależność od systemu zarządzania bazą danych oznacza, że po pierwsze, w skład systemu RBD mogą wchodzić bazy danych zarządzane przez różne systemy zarządzania bazami danych, np. Oracle10g, IBM DB2, MS SQLServer2005, Sybase Adaptive Server Enterprise. Po drugie, sposób dostępu do tych baz danych powinien być jednolity. Oznacza to, że każdy z SZBD powinien dostarczać jednolity i ustandaryzowany interfejs dostępu do bazy danych.



10. Niezależność od sieci

- System RBD powinien pracować w różnych architekturach sieciowych i z różnymi protokołami sieciowymi
- Dostęp do poszczególnych węzłów powinien być jednolity, niezależnie od architektury i protokołów sieciowych

ZSBD – wykład 1 (21)

Niezależność od sieci oznacza, że system RBD powinien pracować również w różnych architekturach sieciowych i z różnymi protokołami sieciowymi. W takim przypadku, dostęp do poszczególnych węzłów powinien być jednolity i niezależny od architektury sieciowej i niezależny od wykorzystywanych protokołów sieciowych.



11. Rozproszone zarządzanie transakcjami

- W systemie RBD można realizować transakcję, która odwołuje się do wielu węzłów \Rightarrow transakcję rozproszoną
- Należy zagwarantować trwałość, spójność, atomowość i izolację transakcji rozproszonych

ZSBD – wykład 1 (22)

Rozproszone zarządzanie transakcjami oznacza, że w systemie RBD można realizować transakcje rozproszone. Transakcja rozproszona odwołuje się do wielu węzłów systemu. W przypadku tego typu transakcji system RBD powinien zagwarantować cztery cechy transakcji rozproszonej, tzn. jej trwałość, spójność, atomowość i izolację, podobnie jak w przypadku standardowych transakcji scentralizowanych.

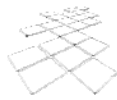


12. Rozproszone przetwarzanie zapytań

- Jedno zapytanie użytkownika może adresować wiele węzłów systemu RBD
- Należy zagwarantować optymalny sposób wykonania takiego zapytania

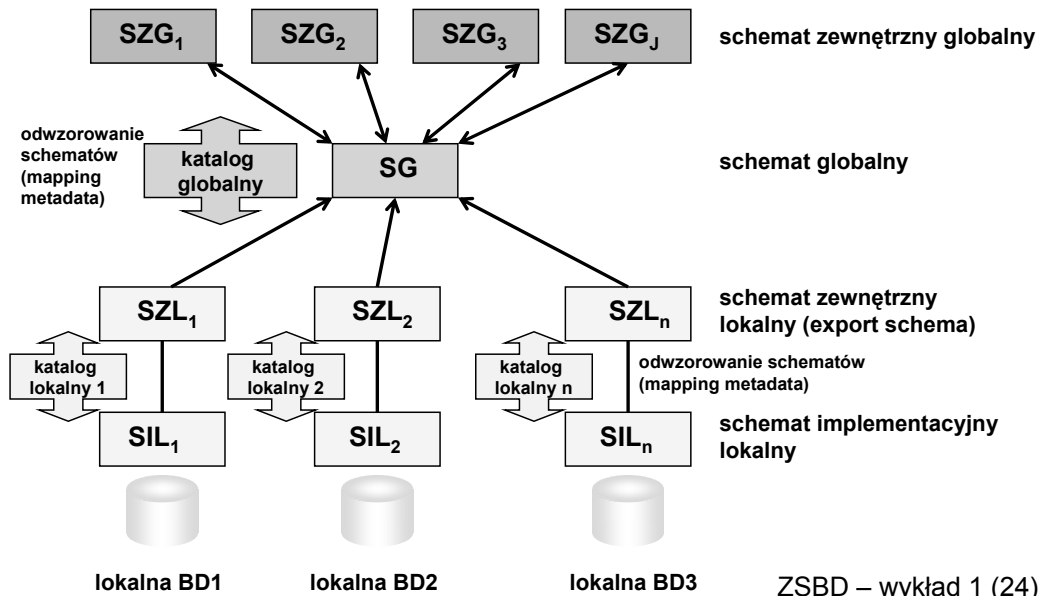
ZSBD – wykład 1 (23)

Rozproszone przetwarzanie zapytań gwarantuje możliwość wykonania zapytania, które adresuje jednocześnie wiele węzłów systemu RBD. W takim przypadku, system powinien zagwarantować optymalny lub suboptymalny sposób wykonania takiego zapytania, zgodnie z przyjętym kryterium kosztu wykonania.



Podstawowa architektura systemu RBD

- Architektura odniesienia ANSI ⇨ system sfederowanych BD



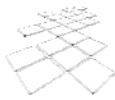
Jedną z podstawowych architektur implementacyjnych systemu RBD jest tzw. system sfederowanych baz danych. Jego ogólną architekturę przedstawia slajd.

Każda z lokalnych baz danych BD1, BD2 i BD3 posiada swój schemat implementacyjny lokalny (SIL1, SIL2, SIL3). Schemat implementacyjny określa w jakim implementacyjnym modelu danych są reprezentowane dane.

Przykładami takich modeli są: model relacyjny, obiektowy, obiektowo-relacyjny, semistrukturalny.

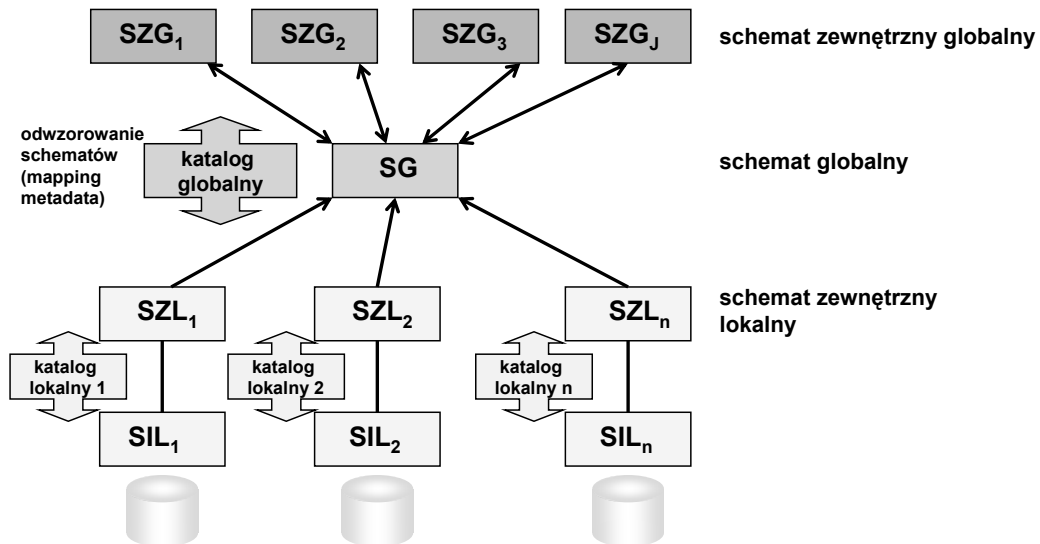
Schemat implementacyjny lokalny jest niedostępny na zewnątrz węzła. Z tego względu, każdy węzeł udostępnia tzw. schemat zewnętrzny lokalny (SZL1, SZL2, SZL3). Schemat ten, pełni dwie zasadnicze funkcje. Po pierwsze, umożliwia on dokonanie konwersji schematu implementacyjnego lokalnego do wspólnego modelu danych wykorzystywanego w systemie sfederowanych BD. Najczęściej jest to model relacyjny. Po drugie, umożliwia udostępnienie nie całego schematu implementacyjnego lokalnego, ale jego fragmentu.

Schemat implementacyjny lokalny jest odwzorowywany w schemat zewnętrzny lokalny z wykorzystaniem katalogu lokalnego. Przechowuje on m.in. odwzorowania nazw obiektów, informacje o prawach dostępu, procedury konwersji.



Podstawowa architektura systemu RBD cd.

- Architektura odniesienia ANSI ⇔ system sfederowanych BD



ZSBD – wykład 1 (25)

Schematy zewnętrzne lokalne są integrowane w jeden schemat globalny (SG). Schemat ten zapewnia, że wszystkie lokalne bazy danych są widziane jako jedna, spójna baza. Schematy zewnętrzne lokalne są odwzorowywane w SG z wykorzystaniem katalogu globalnego. Podobnie, jak katalog lokalny, katalog globalny przechowuje m.in. odwzorowania nazw obiektów, informacje o prawach dostępu, procedury konwersji.

Na podstawie schematu globalnego, użytkownicy systemu sfederowanych BD tworzą własne schematy, tzw. schematy zewnętrzne globalne. Umożliwiają one zawężenie danych udostępnianych przez SG do wycinka interesującego użytkowników. Użytkownicy pracują z systemem poprzez swoje schematy zewnętrzne globalne.



System sfederowanych DB

- Dwa niezależne systemy BD + mechanizm konsolidujący
- Każdy system BD jest autonomiczny i ma swoich użytkowników

ZSBD – wykład 1 (26)

System sfederowanych baz danych to system składający się z co najmniej dwóch niezależnych, różnych systemów baz danych oraz odpowiedniego mechanizmu konsolidującego wszystkie ich komponenty. Ponadto, każdy system BD jest niezależnym i autonomicznym scentralizowanym SZBD, który ma swoich własnych lokalnych użytkowników.



System sfederowanych BD - przykład

- System kontroli opłat abonamentowych TVP
- Komponenty - autonomiczne bazy danych:
 - Urzędu Miasta
 - dane meldunkowe obywateli
 - sieci MediaMarkt
 - dane o sprzedaży odbiorników RTV
 - Urzędu Radiofonii i TV
 - dane o płaconych abonamentach

ZSBD – wykład 1 (27)

Jako przykład systemu sfederowanych BD rozważmy system kontroli opłat abonamentowych TVP. Załóżmy, że w jego skład wchodzi trzy autonomiczne bazy danych należące do: Urzędu Miasta, sieci sklepów MediaMarkt, Urzędu Radiofonii i TV.

Pierwsza z nich udostępnia dane meldunkowe obywateli, a jej zawartość jest niezbędna do wysyłania kar i wezwań do uiszczenia opłat abonamentowych. Druga baza udostępnia dane o sprzedaży odbiorników RTV, tj. rachunki lub faktury wystawiane kupującym sprzęt RTV. Jej zawartość jest niezbędna do zidentyfikowania kupujących sprzęt RTV. Trzecia baza udostępnia dane abonentów, którzy zarejestrowali odbiorniki RTV.

W takim systemie sfederowanych baz danych, odpowiednio uprzywilejowany użytkownik mógłby wydać zapytanie o dane meldunkowe obywateli (baza Urzędu Miasta), którzy w ostatnim roku zakupili odbiorniki RTV (baza sieci MediaMarkt), ale którzy nie płacą abonamentu, tj. nie zostali zarejestrowani w bazie danych Urzędu Radiofonii i TV.