BAZY DANYCH I SYSTEMY INFORMATYCZNE ORAZ ICH WPŁYW NA ROZWÓJ INFORMATYKI W POLSCE

Witold Andrzejewski, Zbyszko Królikowski, Tadeusz Morzy

Instytut Informatyki
Politechnika Poznańska
e-mail: witold.andrzejewski@cs.put.poznan.pl
e-mail: zbyszko.krolikowski@cs.put.poznan.pl
e-mail: tadeusz.morzy@cs.put.poznan.pl

1. Wprowadzenie

W ciągu ostatnich 60 lat bazy danych wyewoluowały z prostych systemów plików i stały się zaawansowanymi strukturami danych składującymi dane olbrzymiej liczby użytkowników dla wielu różnych aplikacji. Bazy danych znajdują zastosowania w każdej dziedzinie życia, która podlega informatyzacji. W bazach danych składowane są: dane pracowników firm, dane klientów banku, dane o sprzedaży towarów, dane firm ubezpieczeniowych, dane pacjentów szpitali itp. Prostymi bazami danych są: listy utworów muzycznych w odtwarzaczach MP3 i książki telefoniczne w telefonach komórkowych. W systemach nawigacji konwencjonalne bazy danych wykorzystuje się do przechowywania informacji o szpitalach, bankach, bankomatach, zabytkach, stacjach benzynowych itp., a mapy składowane są w przestrzennych bazach danych. Bazy danych stanowią podstawę większości serwisów internetowych, w ramach których przechowują m.in. informacje o użytkownikach, ich preferencjach i historii korzystania z serwisu. Przykładem powszechnie wykorzystywanej bazy danych jest np. serwis Google. Systemy plików w komputerach również są prostymi bazami danych. W bazach danych przechowuje się wyniki obserwacji astronomicznych, wyniki sekwencjonowania DNA, wyniki eksperymentów fizycznych, chemicznych i biologicznych. Potencjalny zakres zastosowań baz danych jest zatem olbrzymi i ciągle się powiększa. Pojawiają się ciągle nowe dziedziny zastosowań, w których zachodzi konieczność składowania, przeszukiwania, zarządzania i analizowania

danych. Wraz z nowymi dziedzinami zastosowań pojawiają się nowe problemy naukowe związane z zapewnieniem efektywnej pracy systemów baz danych w nowych zastosowaniach, jak również zapewnienie nowych metod przeszukiwania i analizy danych. Przykładami mogą być tutaj ewolucja baz danych od systemów przechowujących proste dane alfanumeryczne do systemów przechowujących dane o złożonych typach, w tym, dane multimedialne i semistrukturalne. Innym przykładem może być ewolucja złożoności wyszukiwania i przetwarzania danych, począwszy od prostego wyszukiwania rekordów spełniających określone warunki, poprzez systemy obliczające złożone podsumowania na podstawie olbrzymich zbiorów danych, a kończąc na tak zwanej eksploracji danych, której celem jest odkrywanie wiedzy (to jest, zależności, które nie są explicite zapisane w bazie danych) na podstawie zgromadzonych danych. Mimo niewątpliwego sukcesu komercyjnego technologii systemów baz danych, hurtowni danych czy eksploracji danych, będącego wynikiem olbrzymiej ilości środków i pracy, jaką włożyło środowisko naukowe i firmy komercyjne w rozwój tych technologii, nadal rozwiązanie szeregu problemów badawczych i implementacyjnych warunkuje dalszy rozwój tych technologii i ich wykorzystanie w nowych dziedzinach zastosowań. Należy wśród nich wymienić analizę danych sekwencyjnych, analizę informacji zawartych w blogach, forach i serwisach społecznościowych, transakcyjne przetwarzanie danych w architekturze zorientowanej na usługi, optymalizację przetwarzania danych w hurtowniach danych przy wykorzystaniu nowych rozwiązań sprzętowych, itd. Lista ta oczywiście nie wyczerpuje wszystkich potencjalnych tematów rozwoju technologii baz danych. W niniejszej pracy chcielibyśmy przybliżyć dotychczasowe osiągnięcia w dziedzinie technologii baz danych.

W rozdziale 2 przedstawiono historię rozwoju baz danych. W rozdziale 3 scharakteryzowano krótko typy baz danych, które powstały w ciągu ostatnich 60 lat rozwoju technologii baz danych. W ostatnich dwóch rozdziałach omówiono wybrane, nowe, interesujące zastosowania technologii baz danych, które wychodzą poza standardowe podejście do baz danych, jako repozytorium składowania danych.

2. Historia rozwoju baz danych

Zanim pojawiły się pierwsze urządzenia pozwalające na składowanie i automatyczne przeszukiwanie oraz przetwarzanie danych, dane były zapisywane i przetwarzane ręcznie. Dane gromadzono na glinianych tabliczkach w Sumerii, papirusach w Egipcie, czy też na papierze, najpierw w Chinach, a później na całym świecie. Powstało wiele różnych systemów pisma, w tym supełkowe, piktograficzne, ideograficzne i fonetyczne, z których każde charakteryzowało się pewną swobodą wyrażania koncepcji mniej lub bardziej abstrakcyjnych oraz stopniem skomplikowania, a przez to wymaganymi kwalifikacjami koniecznymi do ich skutecznego wykorzystywania. Dane zapisane w ten sposób były przechowywane w archiwach, bibliotekach i katalogach. Wynalazek druku upowszechnił dostęp do danych, jednak mimo zautomatyzowania kopiowania danych, przetwarzanie ich było wciąż ręczne. Czytelnicy korzystali z książek w bibliotekach, urzędnicy przetwarzali dane handlowe, dane o zebranych podatkach oraz dane statystyczne i demograficzne w celu opracowywania podsumowań, które pozwalały kupcom i właścicielom ziemskim dobrze zarządzać swoim majątkiem. Wraz z rozwojem handlu oraz przyrostem ludności, coraz trudniej było ogarnąć wszystkie dostępne dane. Zmieniło się to wraz z pojawieniem się nowych wynalazków w dziewiętnastym wieku, a w szczególności kart perforowanych oraz urządzeń mechanicznych i mechaniczno-elektrycznych, które z tych kart korzystały. Pierwszymi wynalazkami tego typu było np. krosno Josepha Jacquarda, sterowane za pomocą zapisu na karcie perforowanej, pianole, które odtwarzały utwory zakodowane na takich kartach, czy maszyna analityczna Charlesa Babbage'a, która miała być sterowana za pomocą instrukcji odczytywanych z kart perforowanych[1]. Wykorzystanie kart perforowanych do przechowywania i przetwarzania danych zostało zapoczątkowane przez Hermana Holleritha [Gray96]. Hollerith opracował maszynę nazywaną „tabulatorem", której zadaniem było odczytywanie danych demograficznych zapisanych na kartach perforowanych,

[1] Maszyny analitycznej nigdy nie zbudowano. Istniała tylko w postaci projektów

sumowanie ich i drukowanie raportów. Tabulator został wykorzystany po raz pierwszy do opracowania spisu ludności w Stanach Zjednoczonych w roku 1890 i przyspieszył jego opracowanie z 8 lat (potrzebnych na opracowanie poprzedniego spisu z roku 1880) do jednego roku. Pierwszy tabulator był na stałe skonfigurowany do obliczeń związanych ze spisem ludności. Kolejne tabulatory były programowane poprzez rekonfigurację niektórych obwodów urządzenia na specjalnym panelu zarządzającym rejestrami zliczającymi, których wartości były następnie drukowane w postaci tabel na papierze, bądź zapisywane na innych kartach perforowanych.

W 1896 Hollerith założył firmę *Tabulating Machine Company*, której celem była budowa i sprzedaż urządzeń służących do zapisu danych na kartach perforowanych, późniejszego ich przetwarzania i drukowania w postaci tabelarycznej [Shur84]. W roku 1911 cztery korporacje, w tym firma Holleritha połączyły się w *Computing Tabulating Recording Corporation* (CTR). CTR, w okresie zarządzania przez Thomasa J. Watsona, w roku 1924 przyjęła nazwę *International Business Machines* (IBM). W latach 1915-1960 CTR, a później IBM rozwijał się pomyślnie, dzięki dużemu popytowi na urządzenia przetwarzające dane zapisane na kartach perforowanych zarówno w kręgach rządowych, jak i w biznesie[2]. Urządzenia Holleritha przyjęły się bardzo szybko i w rezultacie, już w roku 1955 wiele firm posiadało całe piętra przeznaczone na przechowywanie kart perforowanych oraz ich katalogowanie i przetwarzanie. Większe firmy wykorzystywały swoje urządzenia do przetwarzania nawet milionów rekordów każdej nocy, co jak łatwo zauważyć nie było możliwe do wykonania ręcznie [Gray96].

Pierwsze komputery, które przechowywały zarówno dane, jak i program w pamięci operacyjnej (zgodnie z architekturą von Neumanna) zostały opracowane w latach czterdziestych i wczesnych pięćdziesiątych dwudziestego wieku i były wykorzystywane początkowo dla obliczeń naukowych i numerycznych. Mniej więcej w tym samym czasie przedsiębiorstwo *Univac*

[2] Thomas J. Watson znany jest między innymi ze swojego twierdzenia z roku 1943: „Zapotrzebowanie na komputery na świecie szacuję na około pięć sztuk.”. Jak łatwo zauważyć, pomylił się on w swoich oszacowaniach o kilka rzędów wielkości.

opracowało metodę składowania danych na taśmie magnetycznej. Taśma magnetyczna mogła pomieścić tyle samo danych, co dziesięć tysięcy kart perforowanych. Firma ta opracowała również komputer UNIVAC1, który jest uznawany za pierwszy komercyjny, elektroniczny komputer ogólnego przeznaczenia. Pierwszy egzemplarz tego komputera został dostarczony w roku 1951 do amerykańskiego biura spisowego, tego samego, w którym debiutował tabulator Holleritha sześćdziesiąt lat wcześniej [Gray96]. Piąty egzemplarz UNIVAC1, zbudowany dla Komisji Energii Atomowej Stanów Zjednoczonych (*U.S. Atomic Energy Commission*) został wykorzystany przez sieć telewizyjną CBS w celu wykonania predykcji wyników wyborów prezydenckich w roku 1952. Na podstawie danych zebranych od około 1% uprawnionych do głosowania, maszyna poprawnie przewidziała sukces Dwighta Eisenhowera w tych wyborach. Nowe komputery potrafiły przetwarzać dane z szybkością kilkuset rekordów na sekundę i zajmowały jedynie niewielką część miejsca wymaganą przez maszyny poprzedniej generacji. Dodatkową zaletą tych komputerów był fakt, iż mogły być znacznie łatwiej dostosowywane do nowych zadań. W przeciwieństwie do tabulatorów, które były konfigurowane za pomocą fizycznej zmiany obwodów urządzenia, komputery UNIVAC mogły być programowane, co czyniło je znacznie wygodniejszymi w użyciu. Oprogramowanie stanowiło istotny trzon tej technologii. Powstał szereg języków programowania, za pomocą których można było sortować, analizować i przetwarzać dane. Wśród takich języków można wymienić: FORTRAN, LISP, COBOL i RPG. Zaczęły się również pojawiać pierwsze pakiety programów pozwalających na wykonywanie standardowych operacji takich jak: prowadzenie księgi głównej, zarządzanie płacami i magazynem, zarządzanie abonamentami, obsługa ogólnie pojętej bankowości i zarządzanie bibliotekami dokumentów. W oprogramowaniu, które pojawiło się w tamtych czasach, wprowadzono również koncepcję przetwarzania zorientowanego na pliki. Każdy plik był rozumiany jako sekwencja rekordów różnych typów. Typowe programy działały wówczas według następującego schematu. Na początku program odczytywał zawartość kilku plików wejściowych, następnie wykonywał obliczenia i zapisywał wyniki obliczeń w nowym pliku. Ówczesne

języki programowania, np. COBOL były przystosowane do tego typu przetwarzania. Wprowadzenie plików spowodowało konieczność zarządzania dostępem do nich, co przyczyniło się do powstania koncepcji systemów plików zarządzanych przez system operacyjny komputera. Ponieważ dane w tamtych czasach były składowane na nośnikach umożliwiających jedynie sekwencyjny dostęp (taśmach magnetycznych), to ówczesne systemy plików również pozwalały jedynie na sekwencyjny dostęp do danych. Należy tutaj jednak zwrócić uwagę na jeden fakt. Otóż, system plików stanowi zorganizowany zbiór danych, a zatem na mocy definicji bazy danych można taki system plików uznać za prostą bazę danych. Tak też jest w rzeczywistości. Proste systemy plików są prekursorami dzisiejszych baz danych.

Oprócz wymuszenia sekwencyjnego odczytu i zapisu danych, komputery stosowane w tamtych czasach miały jeszcze jedną wadę – nie były one urządzeniami interaktywnymi. Dane przetwarzane były wsadowo. Wsadowe przetwarzanie polegało na wstępnym przygotowaniu zestawu danych wejściowych i późniejszym uruchomieniu przetwarzania zadania. Z reguły przetwarzanie takie uruchamiane było w nocy, przez co o błędach wykonania, bądź o niespójności danych wejściowych z danymi już składowanymi w plikach na taśmie, można było przekonać się dopiero następnego dnia. Wymagany był zatem kolejny krok w ewolucji komputerów i metod przetwarzania danych, który zapewniłby przetwarzanie interaktywne.

Obsługa takich obszarów zastosowań jak rynek akcji i papierów wartościowych oraz biura podróży wymagały znajomości danych aktualnych. Przetwarzanie wsadowe, które zapewniało jedynie dostęp do danych z ostatniego wsadu nie było w stanie zaspokoić tego wymagania. Postępujący rozwój sprzętu komputerowego, w tym opracowanie monitorów CRT, dysków magnetycznych oraz komputerów interaktywnych umożliwiły dalszy rozwój technologii pozwalających na składowanie, przeszukiwanie i przetwarzanie danych. W szczególności istotny był tutaj rozwój dysków magnetycznych, które pozwalały na swobodny (w przeciwieństwie do sekwencyjnego) dostęp do danych. Na nośnikach tego typu możliwy był dostęp bezpośrednio do wybranych fragmentów plików, bez konieczności sekwencyjnego odczytania

wszystkich danych fizycznie umieszczonych przed nimi. Dostęp taki możliwy był oczywiście dzięki opracowaniu bardziej zaawansowanych systemów plików, które potrafiły przetransformować adres logiczny (pozycję informacji w pliku) na adres fizyczny (sektor dyskowy), za pomocą którego wykonywany był dostęp do odpowiedniego fragmentu nośnika. Komputery interaktywne z wieloma terminalami oraz nośniki danych pozwalające na dostęp w ułamku sekundy do wybranego rekordu w pliku, umożliwiły utworzenie systemów przetwarzających dane w trybie on-line. Był to początek systemów baz danych.

Pierwszy, tak zwany hierarchiczny, system bazy danych, nazywany IMS (*Information Management System*), został opracowany przez IBM dla środowiska MVS (*Multiple Virtual Storage*). Jego pierwsza wersja (IMS 360 Version 1) została dopuszczona do sprzedaży w roku 1968. IMS szybko stał się jednym z trzech produktów o największej liczbie instalacji i użytkowników. IMS zarządzał danymi zorganizowanymi w postaci hierarchii, to jest wykorzystywał hierarchiczny model danych. Główną przyczyną motywującą taki a nie inny model danych, było umożliwienie wykorzystywania urządzeń pozwalających jedynie na sekwencyjne składowanie i odczytywanie danych, np. z taśm magnetycznych. Należy tutaj zwrócić uwagę, iż pomimo tego, że były już dostępne twarde dyski, miały one niewielką pojemność i zajmowały dużo miejsca. Nie nadawały się zatem do pełnienia roli głównego repozytorium danych. Rolę tę nadal pełniły taśmy. Jednym z najważniejszych pomysłów wprowadzonych przez IMS było oddzielenie zarządzania danymi od aplikacji wykorzystujących te dane. Poprzednio, każda aplikacja posiadała własne pliki z danymi, we własnych formatach, często duplikujące te same dane, gdy były one wykorzystywane w więcej niż jednej aplikacji. Wprowadzenie centralnego mechanizmu składowania i zarządzania danymi uwolniło programistów od konieczności samodzielnego oprogramowania takich zadań jak odczyt danych, wyszukiwanie danych, modyfikacja, dodawanie i usuwanie danych (współbieżnie wykonywane przez wielu użytkowników), odtwarzania danych po awarii i zapobieganie niespójności w danych. Należy pamiętać, że tego typu zadania nie są łatwe w implementacji i wymagają istotnych nakładów pracy.

Wprowadzenie niezależnego systemu, który realizował centralne zarządzanie danymi, zmniejszyło koszty i czas implementacji nowych aplikacji, jak również zwiększyło ich niezawodność i bezpieczeństwo przetwarzanych w nich danych. Wadą hierarchicznego modelu danych była konieczność przechowywania wielu redundantnych kopii tego samego rekordu w różnych hierarchiach, co utrudniało operacje modyfikacji, dodawania i usuwania danych.

Hierarchiczny model danych był prekursorem kolejnego modelu danych, tak zwanego sieciowego modelu danych, zaimplementowanego w opracowanym w General Electric, przez Charlesa Bachmana, systemie IDS (*Integrated Data Store)*. W modelu sieciowym, w przeciwieństwie do modelu hierarchicznego, każdy rekord mógł mieć kilku rodziców i kilku potomków, efektywnie zamieniając hierarchię danych w graf (sieć). Model sieciowy pozwalał na reprezentację bardziej skomplikowanych związków pomiędzy rekordami, niż tylko hierarchia, Pozwalało to na uniknięcie konieczności przechowywania redundantnych kopii rekordów. Przeglądanie danych w modelu sieciowym polegało na nawigacji po łączach pomiędzy rekordami. Konieczność przetwarzania danych opisanych w postaci powiązanych ze sobą zbiorów rekordów występowała w tak wielu zastosowaniach, że konsorcjum CODASYL (*Conference on Data System Languages*) w połowie lat 60-tych utworzyło *List Processing Task Force*. Była to grupa informatyków, których zadaniem było opracowanie standaryzowanych rozszerzeń dla języka COBOL, które pozwoliłyby wykorzystywać funkcjonalność zaimplementowaną przez Bachmana w systemie IDS. Należałoby w tym miejscu wspomnieć, iż w połowie lat 60-tych, pomimo tego, że termin „baza danych" był już w użyciu, termin „przetwarzanie list" (*list processing*) był znacznie popularniejszy, stąd też nazwa *List Processing Task Force*. W roku 1967 nazwa *List Processing Task Force* została zmieniona na *Database Task Group* (DBTG) co znacznie przyczyniło się do popularyzacji powszechnie dzisiaj używanego terminu „baza danych" [Olle06]. Grupa DBTG opracowała dwa języki: język definicji danych (*data definition language* – DDL), pozwalający na specyfikację struktur logicznych, w których dane są przechowywane oraz język manipulacji danymi (*data manipulation language*

– DML), pozwalający na modyfikację danych składowanych w bazie danych. Ponadto, DBTG opracowało koncepcję schematów. Wyróżniono 3 rodzaje schematów: *schemat logiczny*, który opisuje logiczną organizację całej bazy danych, *podschemat* określający fragment bazy danych widziany przez użytkownika (lub aplikację użytkownika) i *schemat fizyczny*, który opisywał fizyczne umieszczenie rekordów bazy danych na nośniku. Mechanizm oparty na schematach logicznych, fizycznych i podschematach, opracowany przez DBTG, wprowadził koncepcję niezależności danych. Podschematy udostępniały jedynie fragment bazy danych przeznaczony dla danej aplikacji lub użytkownika, co zapewniało zwiększone bezpieczeństwo danych przed nieautoryzowanym dostępem i niezamierzoną modyfikacją. Wykorzystanie niezależnego schematu fizycznego i logicznego pozwoliło na oddzielenie ewolucji aplikacji od ewolucji systemu bazy danych, w którym dane aplikacji są składowane. Słuszność tego podejścia potwierdza fakt, iż do dzisiaj działają aplikacje oparte na tym samym podschemacie, pomimo tego, iż leżące poniżej schematy logiczny i fizyczny dawno uległy zmianie [Gray96].

Systemy on-line musiały rozwiązać wiele problemów występujących przy współbieżnym dostępie do danych, które nie występowały w systemach wsadowych. Wczesne systemy on-line wprowadziły koncepcję transakcji jako zbioru operacji wykonywanego atomowo. Transakcje blokowały dostęp do danych, na których wykonywały swoje operacje, co pozwoliło uniknąć anomalii wynikających ze współbieżnej modyfikacji danych. Dodatkowo, systemy utrzymywały dziennik zmian (plik logu) wykonywanych w ramach transakcji. Jeżeli transakcja nie powiodła się, informacje z dziennika były wykorzystywane do wycofania zmian wykonanych przez niezakończoną transakcję. Jeżeli uszkodzony został nośnik danych, dziennik zmian pozwalał na odtworzenie najnowszego stanu bazy danych z kopii zapasowej, poprzez zaaplikowanie zmian, które zostały wykonane już po wykonaniu kopii [Gray96].

Do końca dekady lat 80-tych sieciowe systemy baz danych cieszyły się dużym powodzeniem. Wielu projektantów aplikacji uważało jednak, iż przeglądanie bazy danych za pomocą nawigacji po łączach rekordów jest realizowane na zbyt niskim

poziomie, co w znaczny sposób utrudniało pisanie aplikacji wykorzystujących bazy danych tego typu [Gray96]. Przykładowo, aby odpowiedzieć na konkretne zapytanie skierowane do bazy danych, programista musiał napisać skomplikowany program nawigujący po bazie danych, uwzględniając przy tym wszystkie ograniczenia wynikające z optymalizacji dostępu do dysku twardego. Odpowiadanie na zapytania wymyślane ad-hoc było raczej niemożliwe. Co więcej, ponieważ zapytania były dostosowane do struktury bazy danych, jakiekolwiek zmiany w tej strukturze mogły wymagać przebudowy aplikacji z niej korzystających [DBS90]. Rozwiązaniem tych problemów okazał się kolejny model danych nazwany relacyjnym modelem danych. Podstawy tego modelu zostały opublikowane w roku 1970 przez E. F Codda w artykule „A Relational Model of Data for Large Shared Databanks" [Codd70]. Model relacyjny opierał się na kilku rewolucyjnych pomysłach. Pierwszym z nich była reprezentacja danych. Zgodnie z modelem relacyjnym dane są przechowywane jako zbiory rekordów tego samego typu (tzw. relacje), a związki pomiędzy rekordami są reprezentowane niejawnie poprzez wartości atrybutów. Zdefiniowanie danych jako zbioru rekordów pozwoliło na wykorzystanie algebry relacji i jej operatorów do przetwarzania zapytań w bazie danych. Wynikiem działania operatorów algebry na relacjach są relacje. Pozwala to na łączenie (konkatenację) operatorów. Zapytania napisane z wykorzystaniem algebry relacji były znacznie prostsze od niskopoziomowych zapytań modelu sieciowego, a co najważniejsze były to zapytania *deklaratywne*. Innymi słowy, zapytania nie były procedurami określającymi jak odnaleźć poszukiwane rekordy w bazie danych, a były jedynie specyfikacją poszukiwanego zbioru rekordów. Znalezieniem optymalnego algorytmu realizacji takich zapytań zajmować się miał moduł systemu zarządzania bazą danych nazywany optymalizatorem zapytań. Takie podejście zwalniało programistów z czasochłonnego przygotowywania i optymalizacji zapytań oraz pozwalało na tworzenie zapytań ad hoc. Ponieważ powiązania pomiędzy rekordami nie były definiowane w sposób jawny, zapytania zawarte w aplikacjach były bardziej odporne na zmiany w strukturze bazy danych, przez co konieczność ich aktualizacji występowała rzadziej. Artykuł Codda okazał się przełomowy i stanowił inspirację dla świata nauki zajmującego się bazami danych. W krótkim

okresie po publikacji Codd'a osiągnięto wiele istotnych rezultatów. Po pierwsze, w celu ułatwienia tworzenia zapytań stworzono języki wysokiego poziomu pozwalające przygotowywać takie zapytania. Najpopularniejszym z nich okazał się język SQL (*Structured Query Language*), który pomimo swojej nazwy nie był jedynie językiem zapytań, ale integrował w sobie język zapytań, język definicji danych, język manipulowania danymi oraz język kontroli danych, który pozwalał na definiowanie reguł uwierzytelniania i autoryzacji dostępu do danych. Opracowano również teorię języków zapytań wysokiego poziomu, dzięki której możliwa była ocena zdolności języka do określania poszukiwanego zbioru danych (siły ekspresji języka). Opracowano teorię „normalizacji" pozwalającą na eliminację redundancji danych i anomalii występujących podczas eksploatacji bazy danych. Opracowano szereg technik pozwalających na przyspieszenie realizacji zapytań, w tym reguł i algorytmów stosowanych przez optymalizator zapytań, algorytmów rozmieszczenia danych na dysku, algorytmów buforowania danych zmniejszających konieczność wykonywania dostępów do pamięci dyskowej do minimum, algorytmów równoległej realizacji zapytań oraz technik indeksowania danych eliminujących konieczność przeszukiwania całych relacji w poszukiwaniu wybranych rekordów [DBS90]. Zaimplementowano wiele eksperymentalnych relacyjnych systemów zarządzania bazą danych, a pierwsze komercyjne systemy oparte na tym modelu pojawiły się początku lat 80-tych. Dodatkową zaletą relacyjnego modelu danych jest fakt, iż dane relacyjne bardzo dobrze pasują do intuicyjnego sposobu postrzegania zbiorów rekordów o identycznej strukturze w postaci dwuwymiarowych tabel. Użytkownicy mogą w łatwy sposób manipulować takimi danymi w intuicyjny, wizualny sposób za pomocą graficznego interfejsu użytkownika (ang. *Graphical User Interface* – GUI). Systemy relacyjne, obsługiwane za pomocą aplikacji z GUI pozwalają na łatwą konstrukcję skomplikowanych zapytań ludziom nieposiadającym wykształcenia informatycznego. [Gray96].

Model relacyjny oferował wiele udogodnień, przez co już w roku 1990 systemy relacyjne stały się bardziej popularne niż systemy hierarchiczne i sieciowe. Zaczęły się jednak ujawniać pewne wady modelu relacyjnego. Okazało się, że model relacyjny nie jest wystarczający do konstrukcji baz danych dla

nowych dziedzin zastosowań. Jednym z podstawowym zarzutów była ograniczona liczba typów danych, które mogły być przechowywane w relacjach. Po drugie, istniała wyraźna separacja pomiędzy danymi składowanymi w bazie danych a działaniami wykonywanymi na tych danych zdefiniowanymi w aplikacji. Próba rozszerzenia języka SQL o nowe typy danych takie jak: czas, interwał czasowy, znacznik czasowy, data, waluta oraz wiele różnych wariantów liczb i łańcuchów okazała się niewystarczająca. Zaistniała konieczność, ze względu na potrzeby nowych aplikacji, zapewnienia możliwości przechowywania danych o złożonych typach, takich jak np. obrazy, dźwięki albo mapy. Przewidzenie wszystkich możliwych zastosowań i implementacja wszystkich możliwych typów danych okazało się niewykonalne. Zaproponowano rozwiązanie polegające na rozszerzeniu systemów zarządzania bazą danych o możliwość definiowania przez programistę własnych typów danych, wraz ze skojarzonymi z nimi operacjami, które mogłyby być przechowywane następnie w bazie danych. Stanowiło to przyczynę, dla której w roku 1985 społeczność akademicka zaczęła prace nad nowymi modelami danych. W wyniku tych prac powstał obiektowy model danych, który zakładał powiązanie danych z operacjami na nich wykonywanymi. Wiele eksperymentalnych i komercyjnych zorientowanych obiektowo systemów zarządzania bazą danych pojawiło się w późnych latach 80-tych i wczesnych 90-tych. Niestety, model obiektowy nie przyjął się zbyt dobrze na rynku i niewiele przedsiębiorstw zdecydowało się na powierzenie swoich danych nowym systemom. W tym czasie producenci tradycyjnych relacyjnych systemów zarządzania bazą danych w odpowiedzi na wspomniane potrzeby oraz próbując ominąć ograniczenia modelu relacyjnego, rozszerzyli swoje produkty o podstawową funkcjonalność charakterystyczną dla modelu obiektowego. Wśród rozszerzeń znajdowały się: hermetyzacja (połączenie danych i operacji na nich wykonywanych w obiekty), tożsamość obiektów (unikalne identyfikatory obiektów), dziedziczenie, typy abstrakcyjne i zagnieżdżone.

Ta ewolucja relacyjnych systemów zarządzania bazą danych doprowadziła do powstania nowych, hybrydowych systemów nazywanych obiektowo-relacyjnymi systemami zarządzania bazą danych (ORSZBD) [Kim95]. Istnieje ogólna zgoda co do faktu, iż podstawą nowych, post-relacyjnych technologii baz danych jest

technologia oparta o połączone modele danych: relacyjny i obiektowy. Nowa technologia powinna mieć wszystkie zalety współczesnych, relacyjnych systemów baz danych (m.in. optymalizację zapytań, transakcje, autoryzację dostępu do danych), ale również zalety modelu obiektowego, w tym dziedziczenie, polimorfizm oraz możliwość definiowania własnych typów łączących dane z operacjami na nich wykonywanymi (hermetyzacja).

Jak zauważono w wielu raportach publikowanych przez środowisko akademickie zajmujące się problematyką przetwarzania danych [SSU91, SSU96, SZ96,BBC+98], istnieje coraz bardziej wyraźny trend w przemyśle komputerowym, mający na celu ułatwienie zarządzania danymi nienumerycznymi. Powszechna dostępność tanich urządzeń pozwalających na pozyskiwanie danych (aparaty cyfrowe, sensory, skanery, dyktafony cyfrowe, telefony komórkowe, itp.), w połączeniu z dostępnością tanich nośników danych o dużej pojemności i wyświetlaczy o wysokiej rozdzielczości, spowodowała powstanie nowych klas aplikacji, wymagających baz danych pozwalających na składowanie i zarządzanie danymi multimedialnymi. Nowa generacja multimedialnych systemów zarządzania bazami danych będzie najprawdopodobniej oparta o ORSZBD, rozszerzone o wsparcie dla multimedialnych typów danych. Na takie wsparcie będzie się składać: zdolność reprezentacji arbitralnych typów danych i procedur, które współdziałają z arbitralnymi źródłami danych, zdolność wstawiania, modyfikacji, usuwania i wykonywania zapytań do danych multimedialnych, zdolność do definiowania i wykonywania abstrakcyjnych operacji na danych multimedialnych oraz zdolność do obsługi heterogenicznych źródeł danych w jednolity sposób.

Powyższy krótki opis historii baz danych pokazuje wysiłek, jaki został włożony przez świat nauki w dziedzinie zarządzania danymi. Pokazuje on również, że sukces komercyjny zawsze musi być poprzedzony badaniami podstawowymi. Wszystkie wyżej opisane systemy baz danych wywodzą się z badań naukowych prowadzonych zarówno w środowiskach akademickich, jak i przemysłowych. Najpierw powstawały one jako prototypowe implementacje pewnych koncepcji, które następnie ewoluowały w systemy komercyjne.

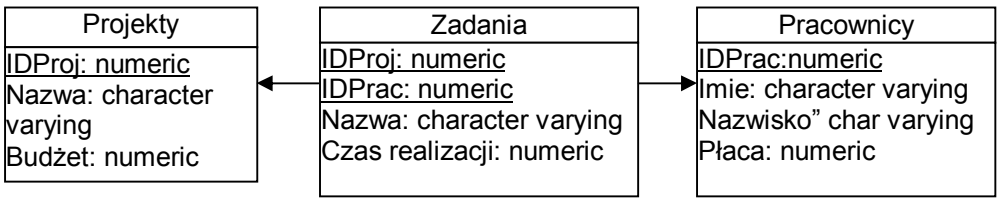
3. Charakterystyka wybranych typów baz danych

W poprzednim rozdziale przedstawiono, krótko, historię baz danych. W międzyczasie powstało wiele różnych systemów baz danych wykorzystywanych w specyficznych dziedzinach zastosowań. W niniejszym rozdziale przedstawiono ważniejsze typy baz danych: relacyjne, obiektowe, obiektowo-relacyjne oraz multimedialne bazy danych, jak również ich warianty: temporalne, przestrzenne, przestrzenno-czasowe, czasu rzeczywistego, rozproszone, semistrukturalne, mobilne oraz hurtownie danych.

3.1. Relacyjne bazy danych

W bazach relacyjnych dane są zorganizowane w postaci zbiorów rekordów nazywanych relacjami. Relacje można w łatwy sposób przedstawić w postaci dwuwymiarowych tabel. Każda relacja opisywana jest przez zbiór atrybutów. Każdy atrybut odpowiada jednej kolumnie tabeli. Każdy atrybut jest powiązany z dziedziną dopuszczalnych wartości (tak zwanym *typem danych*). Każdy wiersz tabeli nazywany jest krotką. Wymagane jest, aby wartość zapisana w pojedynczym atrybucie, w pojedynczym wierszu była atomowa (nie była zbiorem wartości). Wymagane jest również, aby dla każdej relacji istniał atrybut, bądź zbiór atrybutów, których wartość (bądź kombinacja wartości) jednoznacznie identyfikuje każdą krotkę. Taki zbiór atrybutów nazywany jest nadkluczem. Minimalny naklucz, to znaczy taki nadkluczem, dla którego nie można usunąć ani jednego atrybutu bez utraty własności identyfikacji, nazywany jest kluczem. Jeżeli relacja posiada więcej niż jeden klucz, wyróżnia się jeden z nich jako tak zwany klucz główny. Wyróżniamy w relacji również tak zwane klucze obce. Kluczem obcym nazywamy atrybut (bądź zbiór atrybutów), który przyjmuje wartości klucza innej relacji. Klucze obce reprezentują powiązania pomiędzy krotkami w relacjach, ale możliwe jest definiowanie ad-hoc innych warunków łączących krotki, w ramach wykonywania określonego zapytania skierowanego do bazy danych. Przykładowy

schemat relacyjnej bazy danych przedstawiono na rys. 1, a przykładową bazę danych na rys. 2.



Rys. 1. Schemat relacyjnej bazy danych

Projekty

IDProj	Nazwa	Budżet
1	CMS	50000
2	DB	10000

Zadania

IDProj	IDPrac	Nazwa	Czas realizacji
1	10	GUI	20 dni
1	10	Testy	30 dni
1	20	Kod	10 dni
2	10	Testy	15 dni

Pracownicy

IDPrac	Imie	Nazwisko	Płaca
10	Jan	Kowalski	2000
20	John	Smith	2500

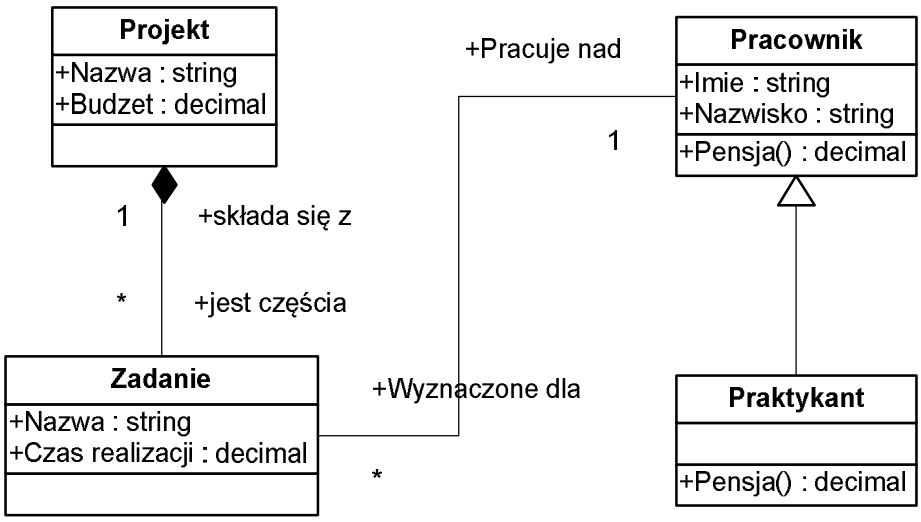
Rys. 2. Przykładowa relacyjna baza danych

Jak wynika z rys. 1, każda relacja, oprócz atrybutów opisujących projekty, zadania lub pracowników, posiada dodatkowe atrybuty: IDPrac w przypadku relacji Pracownicy, IDProj w przypadku relacji Projekty oraz IDPrac i IDProj w przypadku relacji Zadania. IDPrac i IDProj w relacjach Projekty i Pracownicy pełnią funkcję kluczy głównych. IDPrac i IDProj w relacji Zadania pełnią dwojaką rolę. Razem stanowią klucz główny relacji Zadania, a z osobna, każdy z nich stanowi klucz obcy zawierający wartości relacji Projekty (IDProj) lub relacji Pracownicy (IDPrac). Klucz obcy w reprezentuje powiązanie pomiędzy rekordami. Przykładowo, pierwsza krotka w relacji Zadania (patrz rys. 2) posiada wartości atrybutów IDProj i IDPrac odpowiednio 1 i 10. Oznacza to, że zadanie reprezentowane przez ten wiersz jest częścią projektu o numerze 1 i zostało przydzielone pracownikowi o numerze 10.

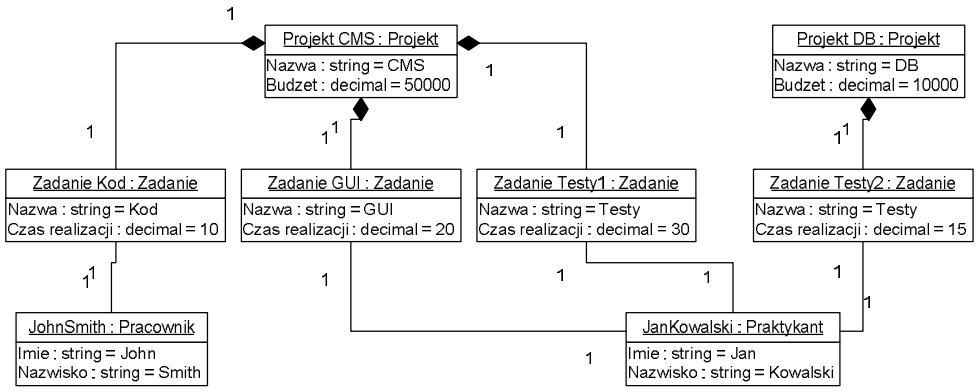
Wśród najbardziej popularnych producentów systemów zarządzania bazami danych (SZBD), zgodnych z modelem relacyjnym (lub post-relacyjnym), są Oracle, IBM i Microsoft, a najpopularniejszymi, darmowymi SZBD są PostgreSQL i MySQL.

3.2 Obiektowe i obiektowo-relacyjne bazy danych

Obiektowe bazy danych przechowują dane w postaci obiektów. Charakterystyczną cechą obiektów jest hermetyzacja, czyli połączenie w jednym obiekcie danych i operacji na nich wykonywanych. Jedną z podstawowych zalet obiektowych baz danych jest możliwość zapisywania i odczytywania obiektów aplikacji bez żadnych faz pośrednich transformujących dane pomiędzy modelem obiektowym i relacyjnym. Na rys. 3 przedstawiono przykładowy diagram klas obiektowej bazy danych, natomiast rys. 4 przedstawia przykładową bazę danych. *Klasą* nazywamy typy danych, które zawierają również definicje operacji wykonywanych na tych danych. W dużym uproszczeniu można je traktować jak typy rekordów, do których dodano operacje, które mogą być wykonywane na rekordach tych typów. Jak łatwo zauważyć na rysunku 3, w tym przykładowym modelu obiektowym zdefiniowano cztery klasy (*typy obiektów*): Projekt, Zadanie, Pracownik i jego specjalizację Praktykant. Pomiędzy klasą Projekt i klasą zadanie zdefiniowano relację kompozycji, co oznacza, że obiekty klasy Zadanie stanowią część obiektów klasy Projekt i nie mogą istnieć samodzielnie. Pomiędzy klasą Zadanie, a klasą Pracownik, zdefiniowano zwykłe powiązanie, które oznacza, że Pracownik może być powiązany z wieloma zadaniami. Ostatecznie, zdefiniowano również klasę Praktykant stanowiącą specjalizację klasy Pracownik. Specjalizacja klasy polega na dziedziczeniu wszystkich pól i operacji klasy nadrzędnej (w tym wypadku klasy Pracownik) i dodaniu własnych istniejących pól, i operacji (tzw. *metod*) i ewentualnej redefinicji operacji odziedziczonych. Praktykant różni się od Pracownika jedynie sposobem obliczania pensji (redefiniuje jedynie metodę Pensja). Na rys. 4 przedstawiono przykładową obiektową bazę danych. Projekt CMS składa się z 3 zadań, z których dwa wykonuje praktykant Jan Kowalski, a jedno zadanie pracownik John Smith. Praktykant bierze również udział w drugim projekcie, w którym wykonuje jedno zadanie.



Rys. 3. Diagram klas obiektowej bazy danych



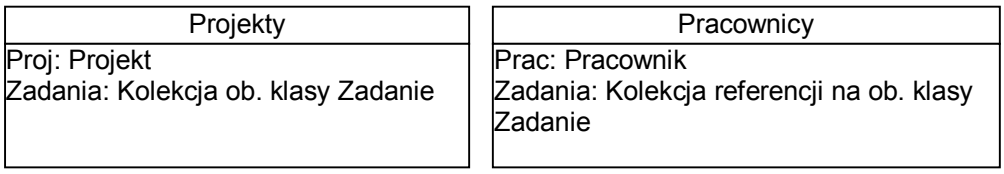
Rys. 4. Przykładowa obiektowa baza danych

Można zauważyć, że obiektowa baza danych przypomina pod pewnymi względami sieciową bazę danych (łącza pomiędzy obiektami tworzą sieć).

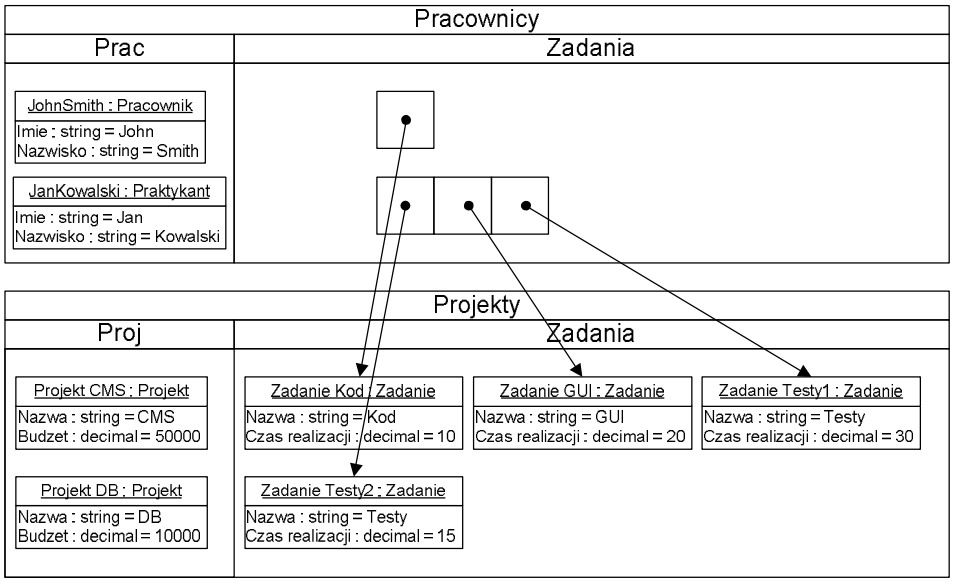
Opracowano kilka prototypów obiektowych systemów zarządzania bazą danych oraz kilka komercyjnych SZBD. Wśród prototypowych systemów można wyróżnić: Exodus (Wisconsin University) [Vos91], Ode (AT&T), Orion (MCC), Zeitgeist (Texas Instruments). Wśród komercyjnych systemów zarządzania bazą danych można wymienić Itasca (MCC), GemStone (Servio Corporation),

ObjectStore (Object Design, Inc.), Ontos (Ontologic), Objectivity/DB (Objectivity, Inc.), Poet (Poet Software), Versant (Versant Corporation), O2 (Ardent Software), db4objects (Versant), JADE (Jade Software Corporation), FastObjects (Versant).

W obiektowo-relacyjnych bazach danych połączono zalety obydwu podejść poprzez zezwolenie w relacyjnym modelu danych na tworzenie własnych, nieatomowych typów danych, w tym typów obiektowych i kolekcji, oraz na tworzenie tabel, których atrybuty mogą mieć być typu zdefiniowanego przez użytkownika. Rozważmy nieznacznie uproszczony schemat obiektowy w stosunku do schematu przedstawionego na rys. 3, w którym nie istnieje relacja kompozycji pomiędzy Zadaniem a Projektem, ani asocjacja pomiędzy Pracownikiem a Zadaniem. Na rys. 5 przedstawiono schemat obiektowo-relacyjnej bazy danych, wykorzystujący ten uproszczony schemat obiektowy, natomiast rys. 6 przedstawia przykładową bazę danych.



Rys. 5. Schemat obiektowo-relacyjnej bazy danych



Rys. 6. Przykładowa obiektowo-relacyjna baza danych

Na schemacie można zauważyć, iż zdefiniowano dwie relacje, których kolumny są niestandardowych typów. W relacji Projekty pierwszy atrybut jest typu Projekt, a zatem relacja Projekty przechowuje obiekty opisujące kolejne Projekty. Obiekty tej klasy posiadają tylko dwa pola: Nazwa i Budżet, ale nie są zdefiniowane żadne powiązania tej klasy z innymi. Rolę usuniętych powiązań przejmuje drugi atrybut relacji Pracownicy, który reprezentuje kolekcje obiektów klasy Zadanie. Pojedyncza krotka w relacji Pracownicy wiąże zatem Projekt z kolekcją Zadań, które należy w ramach tego projektu wykonać. W relacji Pracownicy mamy do czynienia z analogiczną sytuacją. Pierwszy atrybut jest typu Pracownik, co pozwala mu przechowywać wszystkie obiekty typu Pracownik, bądź jego specjalizacje, np. obiekty typu Praktykant. Drugi atrybut relacji Pracownicy reprezentuje kolekcje referencji do obiektów typu Zadanie, co zastępuje usunięte asocjacje. Należy zwrócić uwagę, iż związki pomiędzy klasami zostały tutaj usunięte tylko ze względu na przykład (wcale nie musiały być usuwane). Dodatkowo, pomimo iż w przykładzie nie są wykorzystywane klucze obce, nie znaczy to, że nie można ich używać w dalszym ciągu. Na rys. 6 przedstawiono przykładową obiektowo-relacyjną bazę danych. Można zauważyć, analizując przedstawioną bazę danych, znany już dobrze zbiór danych - tym razem w postaci obiektów składowanych w relacjach. Składowanie obiektów w relacjach jest w modelu obiektowo-relacyjnym obowiązkowe (nie można ich przechowywać poza relacją tak jak ma to miejsce w czysto obiektowych bazach danych). Obiektowo-relacyjnymi systemami zarządzania bazą danych są, między innymi, następujące systemy zarządzania bazami danych: Oracle, PostgreSQL, OpenLink Virtuoso.

3.3. Multimedialne bazy danych

Multimedia to, formalnie, obiekty integrujące w sobie przekaz informacji za pomocą kilku mediów (tekst, obraz, dźwięk, wideo). Termin „multimedia” jest jednak powszechnie, w tym również w kontekście systemów baz danych, używany do opisu wszelkich danych niealfanumerycznych, czyli obrazów, dźwięków (w tym mowy i muzyki) oraz sekwencji wideo, nawet gdy dany obiekt opiera się o jedno medium. Mimo, że otaczająca nas rzeczywistość ma charakter multimedialny, a zadaniem systemów informatycznych jest reprezentacja bądź

imitacja fragmentów tej rzeczywistości, pierwsze systemy informatyczne ograniczały się do przetwarzania danych wyłącznie alfanumerycznych (ze względu na ograniczenia technologiczne). Dopiero w latach dziewięćdziesiątych dwudziestego wieku nastąpił gwałtowny rozwój technologii umożliwiających powszechne składowanie treści multimedialnych w postaci cyfrowej. Z jednej strony o kilka rzędów wielkości zwiększyła się pojemność pamięci masowych przy jednoczesnym spadku ich cen, z drugiej, opracowane zostały efektywne algorytmy kompresji danych multimedialnych, co nie tylko ułatwiło ich przechowywanie, ale i udostępnianie. Równolegle następował rozwój i wzrost przepustowości sieci komputerowych i telekomunikacyjnych, w tym również mobilnych, oraz spadek cen cyfrowych kamer i aparatów fotograficznych. Konsekwencją powyższych procesów jest powszechność danych multimedialnych, czyli obrazów, danych audio i filmów wideo, w postaci cyfrowej i duży rozmiar zgromadzonych kolekcji danych multimedialnych. Pojawienie się potrzeby efektywnego zarządzania dużymi kolekcjami danych multimedialnych stanowiło bezpośrednią motywację dla rozwoju systemów multimedialnych baz danych [Subr98]. Systemy te w naturalny sposób bazują na osiągnięciach technologii baz danych, stawiając sobie za cel udostępnienie mechanizmów zweryfikowanych w praktyce w kontekście tradycyjnych danych alfanumerycznych, takich jak składowanie, przetwarzanie i wyszukiwanie informacji, ochrona przed niepowołanym dostępem, zarządzanie współbieżnością, odtwarzanie po awarii itd., również w stosunku do danych multimedialnych.

Niemniej, ze względu na charakter danych multimedialnych, tworzenie systemów zarządzania multimedialnymi bazami danych wymaga opracowania nowych rozwiązań w zakresie składowania danych, indeksowania danych, prezentacji danych i wyszukiwania danych. Implementacja tych mechanizmów w kontekście multimediów nie jest zadaniem prostym, gdyż wymaga integracji technologii baz danych z innymi dziedzinami, takimi jak rozpoznawanie obrazów, przetwarzanie sygnałów czy transmisja danych.

Składowanie obiektów multimedialnych stanowi wyzwanie przede wszystkim ze względu na ich rozmiar i brak logicznej struktury. Model relacyjny, który po rozszerzeniach jest nadal dominującym modelem danych używanym przez systemy

baz danych, w początkach swojego istnienia nie obsługiwał danych nie posiadających dobrze zdefiniowanej struktury. Standard języka SQL, związanego z relacyjnym modelem danych, dopiero od wersji SQL99 [ME99], wprowadza typ danych BINARY LARGE OBJECT (BLOB), który może być użyty do przechowywania w bazie danych obiektów multimedialnych. Alternatywnie proponuje się składowanie danych w systemie plików serwera bazy danych lub na dedykowanych serwerach, które mogą wykorzystywać hierarchiczne struktury składowania czy specjalne sposoby organizacji danych, na przykład, rozmieszczanie fragmentów pliku na kilku dyskach (ang. disk striping).

Omawiając problematykę składowania danych multimedialnych należy zwrócić uwagę, że zazwyczaj zawartości multimedialnej, zakodowanej binarnie, towarzyszą różnego rodzaju metadane. Metadane te mają różny charakter: począwszy od informacji na temat pochodzenia i formatu multimediów (np. rozdzielczość obrazu, liczba kanałów audio itp.), poprzez właściwości wizualne lub sygnałowe automatycznie z nich ekstrahowane (np. średni kolor, melodia itp.), aż po metadane semantyczne opisujące treści zawarte w przekazie multimedialnym (tematyka, obiekty, ich cechy i związki między nimi). Ponieważ multimedia wraz z opisującymi je metadanymi stanowią złożone obiekty, do ich przechowywania bardziej niż „czysty" model relacyjny odpowiednie są modele obiektowy i obiektowo-relacyjny. Prezentacja danych rozumiana jako udostępnienie wyników zapytań użytkownikowi nie stanowi problemu w tradycyjnych systemach baz danych, ale staje się kluczowym zagadnieniem, gdy udostępnianą zawartością są dane audio lub wideo. Utwory muzyczne i filmy wideo ze względu na duży rozmiar często dostarczane są klientowi strumieniowo. W takim wypadku konieczne jest zapewnianie klientom odpowiedniej jakości usług i wydajne zarządzanie współbieżnie transmitowanymi strumieniami.

Największym wyzwaniem w kontekście multimedialnych baz danych jest jednak wyszukiwanie informacji. Zapytania do multimedialnych baz danych w zasadzie nie odnoszą się bezpośrednio do binarnej zawartości multimediów, ale do metadanych opisujących multimedia. Nie jest problemem obsługa zapytań dotyczących metadanych o pochodzeniu i formacie obiektów multimedialnych, gdyż metadane tego rodzaju są zwykłymi danymi alfanumerycznymi. Badania w

zakresie obsługi zapytań do multimedialnych baz danych skupiają się więc na metadanych semantycznych i danych ekstrahowanych z zawartości. Te pierwsze wymagają opracowania modelu do reprezentacji znaczenia obiektów multimedialnych i treści w nich przedstawionych. Z kolei dla tych drugich konieczne są specjalistyczne algorytmy ich ekstrakcji. Większość prowadzonych badań skupia się na wyszukiwaniu obiektów podobnych względem automatycznie ekstrahowanych cech wizualnych i sygnałowych. Stąd, pojawienie się nowej dziedziny badawczej zajmującej się wyszukiwaniem w oparciu o zawartość (ang. content-based multimedia retrieval) [Lew06].

Z wyszukiwaniem danych multimedialnych w oparciu o zawartość ściśle wiąże się problematyka indeksowania kolekcji multimediów w celu przyspieszenia realizacji zapytań. Multimedialne bazy danych potrzebują podobnych struktur indeksowych do tych używanych w przestrzennych bazach danych, ze względu na fakt, że właściwości obiektu multimedialnego można zamodelować w formie punktu lub wektora w wielowymiarowej przestrzeni. Specyfika multimedialnych baz danych polega na znacznie większej liczbie wymiarów – mogą ich być setki w porównaniu z kilkoma, w przypadku baz przestrzennych. Jednakże, struktury indeksowe efektywne dla przestrzennych baz danych (np. struktury R-drzewa), okazały się niewydajne w przypadku dużej liczby wymiarów. Dla potrzeb multimedialnych baz danych zaproponowano szereg nowych struktur indeksowych takich jak: SS-drzewa, TV-drzewa czy SR-drzewa. Ich zaletą jest skalowalność dla przestrzeni wielowymiarowych, a ponadto, wspieranie zapytań typu „najbliższy sąsiad", charakterystycznych dla multimedialnych baz danych.

Obecnie obowiązują dwa standardy odnoszące się do multimedialnych baz danych. Pierwszy z nich to SQL/MM [ME01] – standard, który uzupełnia SQL o obsługę zaawansowanych typów danych i wsparcie dla zaawansowanych zastosowań. SQL/MM opiera się na obiektowo-relacyjnych mechanizmach języka SQL, które pojawiły się w standardzie SQL99. Mimo, że według pierwotnych założeń SQL/MM miał być poświęcony głównie multimediom, to obecnie, spośród wielu typów danych multimedialnych, SQL/MM obejmuje jedynie obrazy. Pozostałe części standardu SQL/MM dotyczą danych przestrzennych, tekstowych i eksploracji danych. W zakresie obsługi obrazów SQL/MM proponuje obiektowe

typy danych do: (1) przechowywania obrazów, wraz z podstawowymi metadanymi, oraz (2) reprezentacji automatycznie ekstrahowanych własności wizualnych (średniego koloru, histogramu kolorów, lokalizacji kolorów i tekstury). Standard SQL/MM specyfikuje też metody przeprowadzania testów podobieństwa obrazów względem czterech przewidzianych w standardzie właściwości wizualnych.

Drugim standardem ważnym z punktu widzenia multimedialnych baz danych jest MPEG-7 [Mart04]. Jest to standard opracowany przez Moving Pictures Expert Group, która wcześniej zajmowała się metodami kodowania audio i wideo. MPEG-7 dotyczy metadanych, standaryzując format opisu zawartości multimediów na bazie języka XML. W przeciwieństwie do SQL/MM, standard MPEG-7 standaryzuje wszystkie typy multimediów. W kontekście przechowywania i przetwarzania obrazów jest znacznie bardziej szczegółowy niż SQL/MM. MPEG-7 nie normuje jednak ani metod ekstrakcji właściwości wizualnych czy sygnałowych, ani technik konsumpcji opisów (w tym wyszukiwania w oparciu o opisy) zakładając, że standaryzacja wymagana jest jedynie dla formatu opisów, a ich generacja i przeszukiwanie może być polem do konkurencji między systemami implementującymi ten standard.

Wsparcie dla danych multimedialnych we współczesnych systemach zarządzania bazami danych pozostawia jeszcze wiele do życzenia. O ile wszystkie liczące się systemy (Oracle, IBM DB2, Informix, Microsoft SQL Server, Sybase, MySQL, PostgreSQL) oferują możliwość przechowywania dużych obiektów binarnych (różniąc się maksymalnym dopuszczalnym rozmiarem pojedynczego obiektu), to wsparcie dla przetwarzania i wyszukiwania multimediów można znaleźć jedynie w systemach Oracle, IBM DB2 i Informix. Ciągle niewielkie jest znaczenie obowiązujących, wspomnianych wyżej standardów SQL/MM i MPEG-7. Część SQL/MM poświęcona obrazom została zaimplementowana tylko w Oracle (od wersji 10g). Z kolei MPEG-7 nie doczekał się żadnego szczególnego wsparcia ze strony producentów systemów zarządzania bazami danych. Trzeba jednak przyznać, że być może nie jest ono potrzebne, gdyż wystarczą ogólne mechanizmy obsługi danych XML. Z pewnością niepokojący może być natomiast fakt, że systemy będące liderami w zakresie obsługi multimediów, czyli Oracle,

IBM DB2 i Informix, w najnowszych wersjach zostały pozbawione wcześniej dostępnych firmowych mechanizmów wyszukiwania obrazów w oparciu o zawartość. Mechanizm ten pozostał jedynie w produkcie Oracle, ze względu na wsparcie dla standardu SQL/MM. Trend ten można uzasadnić przekonaniem, że mechanizmy wyszukiwania w oparciu o zawartość wymagają dostosowania do konkretnych zastosowań i dlatego powinny być implementowane na poziomie aplikacji, a nie dostarczane w wersji uniwersalnej przez systemy zarządzania bazami danych.

3.4. Semistrukturalne bazy danych

Od 1996 roku, gdy grupa *XML Working Group,* działająca pod auspicjami *World Wide Web Consortium* (*W3C*), stworzyła standard XML (*Extensible Markup Language*), jego znaczenie nieustannie wzrasta. Cechy standardu XML takie jak prostota formatu, czytelność, łatwość tworzenia aplikacji przetwarzających dokumenty XML, możliwość wykorzystania w środowisku Internetu, spowodowały, że w wielu gałęziach gospodarki dokumenty XML stały się jednym z podstawowych sposobów integracji, przechowywania i przetwarzania danych, a liczba stworzonych dokumentów jest w chwili obecnej trudna do oszacowania. W efekcie koniecznością stało się opracowanie rozwiązań, które wspomagałyby użytkowników nie tylko w przechowywaniu dużych ilości dokumentów XML, ale także w wydajnym ich przeszukiwaniu i przetwarzaniu. Rozwiązania te przyjęły różną postać. Po pierwsze, dostosowywano systemy już istniejące, w szczególności relacyjne bazy danych, do możliwości operowania na dokumentach XML. Zaangażowanie poszczególnych producentów oprogramowania było na tyle duże, że zaowocowało stworzeniem specyfikacji SQL/XML, która została stosunkowo szybko zaimplementowana w wielu produktach (DB2, Oracle), i która ostatecznie w roku 2003 stała się fragmentem standardu języka SQL [ISO/IEC 9075-14:2003]. Zadaniem specyfikacji SQL/XML jest wyznaczenie sposobu przetwarzania dokumentów XML w relacyjnych bazach danych. Zakres specyfikacji obejmuje: mapowanie świata baz danych relacyjnych na dokumenty XML, mapowanie schematów relacji na schematy XML, definicję specjalizowanego typu XML przeznaczonego do przechowywania dokumentów XML, a także szereg funkcji

SQL, które, z jednej strony, potrafią tworzyć z danych relacyjnych dowolne dokumenty XML i ich fragmenty, z drugiej, potrafią przetwarzać dokumenty XML przechowywane w specjalizowanym typie XML. Drugim ze sposobów mających na celu umożliwienie przechowywania i przetwarzania dużej liczby dokumentów XML było stworzenie specjalizowanych rozwiązań zorientowanych w pełni na powyższym zadaniu. Rozwiązania te spowodowały powstanie nowego typu baz danych – baz danych dokumentów XML. Początkowo powstawały one całkowicie niezależnie od siebie, tworzone bez uznanych standardów dotyczących sposobów przechowywania, języków zapytań, języków modyfikacji, bez wspólnego interfejsu programistycznego. W konsekwencji, na rynku produktów komercyjnych, a także rynku otwartego oprogramowania, powstało kilkadziesiąt różnych rozwiązań, o różnej funkcjonalności, różnych możliwościach, nie posiadających wielu cech wspólnych. Dopiero prace nieformalnej grupy XML:DB zainicjowały porządkowanie środowiska baz danych dokumentów XML. Zaowocowało to stworzeniem interfejsu programistycznego dla baz danych dokumentów XML (XML:DB API), stworzeniem i wyborem języków zapytań (XPath, XML-Query), języków modyfikacji (XUpdate - XML Update Language), a także opracowaniem definicji baz danych dokumentów XML. Zgodnie z zaproponowaną definicją baza danych dokumentów XML:

- definiuje model dla dokumentów XML (w przeciwieństwie do danych zawartych wewnątrz tego dokumentu) i składuje oraz udostępnia dokumenty wg. tego modelu,
- dokumenty XML są jej podstawową jednostką składowania, analogicznie do tego jaką jest krotka w bazach danych relacyjnych,
- nie wymaga się stosowania określonego fizycznego modelu składowania; bazy danych dokumentów XML można zatem budować w oparciu o bazy danych relacyjne, obiektowe, hierarchiczne, lub wykorzystywać indeksowane, skompresowane pliki przechowywane na poziomie systemu operacyjnego.

Definicja ta wprowadziła jednoznaczny podział pomiędzy rozwiązaniami, które miały na celu adaptację istniejących systemów do przechowania dokumentów XML a rozwiązaniami dedykowanymi. Te pierwsze (*XML-enabled Database Systems*) korzystały z własnego modelu danych (np. relacyjnego lub

obiektowego), te drugie (*Native XML Database Systems*) definiowały swój własny model dla dokumentów XML. Rozwój baz danych dokumentów XML, a także działania standaryzacyjne spowodowały, że ustabilizowały się ich cechy. Zdecydowana większość baz danych pozwala na grupowanie dokumentów XML w tak zwanych kolekcjach dokumentów XML. Kolekcje pełnią różną rolę, część rozwiązań traktuje kolekcje w sposób podobny do katalogów w systemie plików. Dokumenty w nich przechowywane mogą posiadać różną strukturę, kolekcje mogą być zagnieżdżone, itp. W innych rozwiązaniach kolekcje bardziej przypominają tabele w systemach relacyjnych – dokumenty w nich zawarte posiadają z góry narzuconą strukturę. Wydajne przetwarzanie dokumentów XML wymaga wykorzystywania specjalizowanych typów indeksów. Większość istniejących rozwiązań pozwala na wykorzystywanie trzech rodzajów indeksów:

- strukturalnych – pozwalających na wyszukiwanie dokumentów posiadających określoną strukturę,
- opartych na wartościach – pozwalających na wyszukiwanie dokumentów zawierających w określonym elemencie (atrybucie) określoną wartość,
- pełno-tekstowych – przeznaczonych do wydajnego wyszukiwania dokumentów zawierających np. określone słowo, lub sekwencję słów.

Praktycznie wszystkie bazy danych dokumentów XML pozwalają na przetwarzanie dokumentów XML z wykorzystaniem języków zapytań (XPath, XQuery), a także języków modyfikacji (XQuery Update Extension, XUpdate). Większość baz danych dokumentów XML wspiera interfejs programistyczny XML:DB API i koncepcję transakcji oraz kontroluje współbieżny dostęp do dokumentów. Istotną cechą wielu z baz danych dokumentów XML jest możliwość dostępu do różnorodnych danych zewnętrznych i mapowania tych danych na dokumenty XML.

W chwili obecnej bazy danych dokumentów XML, mimo iż nie tak popularne jak relacyjne bazy danych, posiadają ugruntowaną pozycję na rynku. Wykorzystywane są w wielu różnych dziedzinach. Przykładowo, Schiphol Airport w Amsterdamie wykorzystuje bazę danych Tamino do integracji danych dotyczących obsługiwanych lotów pochodzących z ponad 38 różnych systemów informatycznych, Renault wykorzystuje bazę danych X-Hive/DB do integracji

danych pochodzących z czujników montowanych w bolidach Formuły 1, RTL korzysta z bazy danych Tamino przy produkcji programów informacyjnych, a firma HP wykorzystuje bazę danych Ipedo do integracji wewnętrznych danych finansowych.

3.5. Przestrzenne i przestrzenno-czasowe bazy danych

Przestrzenne bazy danych wspierają aplikacje, które składują i przetwarzają dane przestrzenne (wielowymiarowe). Tego typu systemy baz danych są używane w takich zastosowaniach jak: pogodowe serwisy informacyjne, środowiskowe serwisy informacyjne czy też kartograficzne systemy informacyjne. Przykładowo, kartograficzne systemy informacyjne przechowują mapy, wraz z dwu- lub trójwymiarowymi opisami przestrzennymi ich obiektów (krajów, rzek, miast, dróg itp.). Specjalny rodzaj systemów przestrzennych wykorzystywany jest do tworzenia geograficznych systemów informacji (*Geographic Information Systems* – GIS). Pozwalają one na składowanie danych pochodzących ze zdjęć satelitarnych, jak również danych o drogach, sieciach transportowych, itp. Budowa systemu GIS wymaga zaawansowanych metod przechowywania, zarządzania i wizualizacji danych, które nie są dostępne w standardowych SZBD. Systemy GIS pozwalają, między innymi, na realizację zapytań typu: „jaką trasą można dojechać z Warszawy do Poznania, tak, żeby ominąć najbardziej dziurawe drogi", „znajdź najbliższe 5 restauracji z jedzeniem chińskim" lub też „na jakim obszarze widoczny będzie billboard umiejscowiony w zadanym miejscu, na zadanej wysokości i dla zadanej orientacji, oraz czy będzie widoczny z samochodu (uwzględniając kierunek jazdy)". Bardzo często nowe aplikacje GIS przetwarzają dane, które są zarówno przestrzenne, jak i temporalne. Wśród przykładowych danych tego typu można wymienić dane o ruchu drogowym, śledzeniu ruchu zwierząt, dotyczące analiz przemieszczania się ludzi (w tym planowanie połączeń drogowych/transportu, analiza wydarzeń sportowych, np. poruszanie się zawodników) oraz badań rozprzestrzeniania się epidemii, dotyczące analiz danych sieci sensorów (np. organizacja i dynamiczna rekonfiguracja sieci czujników), dane dotyczące planowania ruchu karetek na podstawie analizy pory dnia i miejsc gdzie dochodzi do wypadków/napaści, czy też dotyczące analiz przestępczości

pozwalające na lepsze planowanie rozmieszczenia patroli. Około roku 1990 powstały pierwsze prototypy przestrzenno-czasowych baz danych łączących własności temporalnych i przestrzennych baz danych.

3.6. Bazy danych czasu rzeczywistego

Systemy baz danych czasu rzeczywistego są wykorzystywane do wykonywania zadań, które podlegają ograniczeniom związanym z czasem wykonania oraz wykonującym dostęp do danych, których wartości i ważność zmienia się w czasie. Ograniczenia te są najczęściej wyrażone w postaci ostatecznego terminu wykonania zadania i wynikają z konieczności udostępnienia wyników zadania aplikacji, która musi podjąć odpowiednią decyzję na czas. Systemy baz danych czasu rzeczywistego odgrywają w dzisiejszych czasach bardzo ważną rolę, gdyż istnieje wiele dziedzin zastosowań wymagających przetwarzania danych w czasie rzeczywistym. Wśród przykładowych zastosowań można wymienić komputerowo zintegrowane wytwarzanie, automatyzacja fabryk, robotyka, systemy przepływu pracy, systemy lotnicze, zastosowania militarne, medycyna, kontrola ruchu itp. Systemy baz danych czasy rzeczywistego powstały jako wynik integracji systemów czasu rzeczywistego z tradycyjnymi systemami baz danych.

3.7. Aktywne bazy danych

Aktywne systemy baz danych są wykorzystywane w aplikacjach wymagających, aby pewne operacje były wykonywane w bazie danych, kiedy tylko zajdzie taka konieczność. Aktywne bazy danych posiadają dodatkową funkcjonalność, która pozwala na definiowanie tak zwanych reguł ECA (od ang. *Event-Condition-Action* – zdarzenie-warunek-akcja), które pozwalają określić jakie zdarzenia, przy spełnieniu dodatkowych warunków, powodują wykonanie określonej akcji. Aktywne bazy danych mogą być wykorzystywane w kontrolowaniu procesów przemysłowych i produkcji, systemach monitorowania medycznego, systemach papierów wartościowych, itp. Przykładowo, aktywny system bazy danych może być wykorzystywany do monitorowania ciśnienia krwi u

pacjenta w systemie monitoringu medycznego. Aplikacja może okresowo przekazywać do systemu bazy danych ciśnienie krwi odczytane za pomocą sensorów na ciele pacjenta. Reguła ECA w takim systemie może specyfikować akcję, że w przypadku przekroczenia zadanego przez lekarza progu, należy zapisać datę, godzinę oraz wartość ciśnienia w bazie danych.

3.8. Temporalne bazy danych

Temporalne bazy danych są wykorzystywane w aplikacjach, które organizują dane w zależności od wybranego aspektu czasu. Dla każdej przechowywanej danej w bazie danych mogą być przechowywane również dodatkowe informacje, takie jak: okres czasu w świecie rzeczywistym, w którym dana informacja była prawdziwa, bądź znaczniki czasowe wprowadzenia informacji do bazy danych i zastąpienia jej przez nową wersję tej informacji. Przykładowo, w bazie danych osobowych w relacji Obywatel, może znajdować się atrybut „stan cywilny". Niech będzie dana krotka relacji Obywatel, w której, w atrybucie „stan cywilny", zapisana jest wartość „kawaler". Niech na tej przykładowej bazie danych zostanie wykonana operacja wprowadzenia nowej wartości atrybutu „stan cywilny" - „żonaty". W klasycznym systemie bazy danych, wprowadzenie nowej wersji informacji nadpisuje starą wartość atrybutu, usuwając ją bez śladu. W temporalnej bazie danych zapisanie nowego stanu cywilnego „żonaty" spowoduje wprowadzenie nowej wartości atrybutu „stan cywilny" oraz zapisanie daty wprowadzenia tej informacji. Poprzednia wartość jednak nie zostanie usunięta i będzie możliwe wykonywanie zapytań w tej bazie danych uwzględniających fakt, iż atrybut „stan cywilny" mógł mieć kilka wartości. Bazy danych, które przechowują zarówno daty wprowadzania i zmian informacji, jak i okresy czasu, kiedy informacje były prawdziwe w świecie rzeczywistym, nazywane są *bitemporalnymi* bazami danych**.** Niektóre modele temporalnych baz danych pozwalają nawet na zapisywanie informacji przypuszczalnych, które faktycznie będą znane dopiero w przyszłości. Wykorzystanie baz danych tego typu pozwala użytkownikowi na wykonywanie zapytań uwzględniających zarówno obecny jak i przeszły stan bazy danych. Do przykładowych zapytań, na które potrafią „odpowiedzieć" temporalne bazy danych należą: „znajdź produkt, którego cena

wzrosła w ciągu ostatniego miesiąca trzykrotnie", „znajdź pracowników, którzy przez pięć ostatnie dni wyszli z pracy przed czasem", „gdzie mieszkał Jan Kowalski w ciągu ostatnich dwóch lat, dłużej niż 2 miesiące". Do wyrażania zapytań do temporalnych baz danych powstał wariant języka SQL, oparty o standard SQL92, o nazwie TSQL2. Do przykładowych współczesnych implementacji temporalnych baz danych należą między innymi: Oracle Workspace Manager, która pozwala na zarządzania przeszłymi, obecnymi i proponowanymi wersjami danych w tej samej bazie danych, oraz TimeDB, będąca nakładką na bazę danych firmy Oracle, która przyjmuje zapytania w TSQL2 i transformuje je na zapytania SQL92 wykonywane następnie w bazie danych przez SZBD Oracle.

3.9. Dedukcyjne bazy danych i dedukcyjne, obiektowe bazy danych.

Dedukcyjne bazy danych powstały w wyniku integracji rozwiązań pochodzących z programowania w logice z technologią baz danych. Dedukcyjne systemy baz danych pozwalają na definiowanie tak zwanych reguł dedukcyjnych. Reguły dedukcyjne umożliwiają wywodzenie nowych faktów z danych składowanych w bazie danych. Systemy baz danych tego typu znajdują zastosowanie w kilku różnych dziedzinach zastosowań takich jak: modelowanie procesów gospodarczych, testowanie hipotez, ponowne wykorzystanie istniejących fragmentów kodu, czy też handel elektroniczny. Dedukcyjne-obiektowe systemy baz danych, jak sama nazwa wskazuje, powstały w wyniku integracji systemów obiektowych i dedukcyjnych. Motywacją stojącą za tym połączeniem jest obserwacja, iż te dwa rodzaje systemów posiadają znoszące się wzajemne wady i zalety.

3.10. Mobilne bazy danych

Najnowsze osiągnięcia w dziedzinach technologii bezprzewodowych doprowadziły do powstania mobilnych systemów baz danych, które pozwalają użytkownikom utrzymywać łączność z pozostałymi użytkownikami, bądź głównymi repozytoriami danych nawet w przypadku braku dostępu do komputera lub terminalu. Te cecha mobilnych baz danych jest szczególnie istotna dla

rozproszonych geograficznie organizacji, których pracownicy są mobilni, ale potrzebują okresowego dostępu do danych organizacji. Typowymi przykładami są tutaj: policja drogowa, dyspozytorzy taksówek albo akwizytorzy.

3.11. Równoległe, rozproszone i mediacyjne bazy danych oraz hurtownie danych

Wczesne systemy baz danych były ściśle scentralizowane. Działały na dedykowanych komputerach jednoprocesorowych. Większość dzisiejszych systemów działa w środowisku, w którym wiele procesorów współpracuje równolegle w celu realizacji zadań systemu zarządzania bazą danych. Systemy baz danych, których CPU są fizycznie blisko siebie i komunikują się za pomocą wydajnych łączy (np. magistrali na płycie głównej) nazywane są równoległymi systemami baz danych. Bazy danych, które wykorzystują wiele procesorów, które są rozproszone geograficznie i komunikują się poprzez sieć nazwano bazami rozproszonymi. Opracowanie równoległych i rozproszonych systemów baz danych możliwe było dzięki nowym osiągnięciom w dziedzinach przetwarzania rozproszonego i równoległego [AW98, BEP+00]. Systemy te opracowano z wielu rożnych powodów, począwszy od potrzeby decentralizacji organizacji, zmniejszenia kosztów przetwarzania, a skończywszy na potrzebie zwiększenia autonomii poszczególnych lokacji baz danych. Zrównoleglenie i/lub rozproszenie przetwarzania w systemach baz danych zwiększa dostępność danych, niezawodność, autonomię, wydajność i elastyczność w porównaniu z systemami scentralizowanych baz danych.

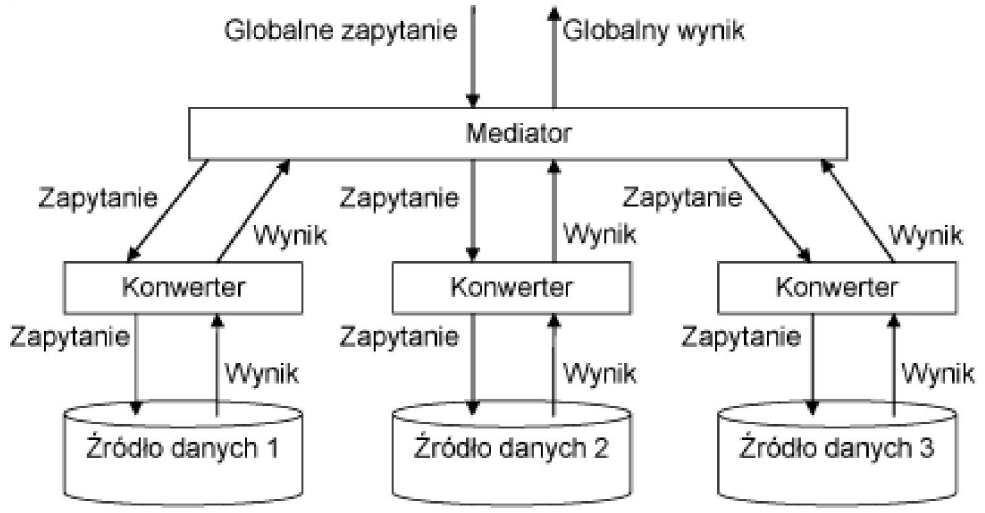
Istnieje kilka podejść pozwalających na zapewnienie jednolitego, zintegrowanego sposobu dostępu do rozproszonych baz danych. Najważniejszymi z nich są systemy mediacyjne i hurtownie danych.

3.11.1 Systemy mediacyjne

Systemy mediacyjne integrują wiele niezależnych źródeł (repozytoriów) danych, którymi mogą być systemy baz danych, systemy plików, arkusze kalkulacyjne, dokumenty XML, itp. Wspólny, „wirtualny”, obraz wszystkich tych

źródeł danych, jako jednej „scentralizowanej bazy danych", jest budowany przez komponent (program) nazywany *mediatorem*. Na rysunku **7** przedstawiono architekturę systemów mediacyjnych.

Systemy mediacyjne składają się ze źródeł danych oraz dwóch warstw oprogramowania: *konwerterów* i wcześniej wspomnianego *mediatora*. Źródła danych mogą być heterogeniczne. Nie czyni się żadnych założeń co do modelu danych w tych źródłach. Konwersję pomiędzy lokalnymi modelami i schematami danych, a globalnym modelem i schematem, na którym działa mediator, wykonują konwertery. Mediator pośredniczy pomiędzy użytkownikiem, a konwerterami przekazując do nich zapytania, a następnie odbierając i agregując wyniki. Procedura realizacji zapytania w systemie mediacyjnym jest zatem następująca. Użytkownik przesyła do mediatora zapytanie globalne, które może dotyczyć danych we wszystkich źródłach danych. Mediator przesyła zapytanie do wszystkich zarejestrowanych konwerterów. Konwertery transformują otrzymane zapytanie do postaci, która jest zrozumiała dla skojarzonych z nimi źródeł danych, a następnie przesyłają je do źródeł celem wykonania. Po wykonaniu zapytania jego wyniki są przesyłane ze źródeł danych do ich konwerterów, które transformują wyniki do wspólnego modelu i schematu danych. Przetransformowane dane są przesyłane do mediatora, który integruje wyniki otrzymane od konwerterów i przesyła je do użytkownika.



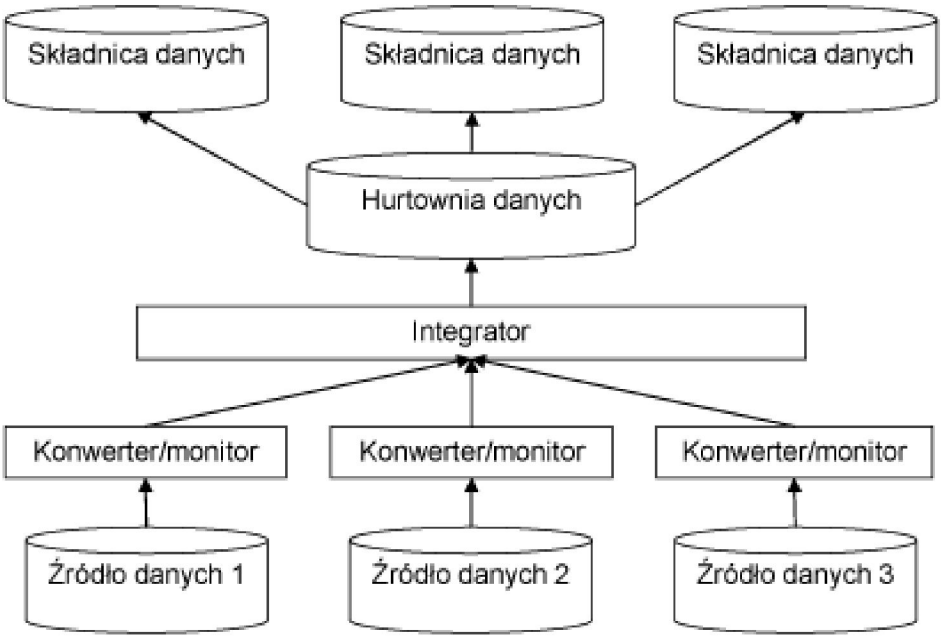
Rys. **7.** Architektura systemu mediacyjnego

Do wad systemów mediacyjnych należy fakt, iż wymagają one bardzo skomplikowanych konwerterów, które muszą transformować dużą liczbę potencjalnych zapytań globalnych do modelu i schematu podległego źródła danych. Dlatego też typową metodą realizacji konwerterów jest wstępne przewidzenie typów zapytań, jakie mogą być przesyłane do konwertera i przygotowanie specjalnej tablicy odwzorowań zapytań globalnych w zapytania lokalne. Takie rozwiązanie jednak nie zapewnia całkowitej uniwersalności systemu mediacyjnego, gdyż zbiór zapytań, jakie można wykonać, jest ograniczony zdolnością do konwersji tych zapytań przez konwerter. Dodatkowym problemami są: zwiększony czas oczekiwania na odpowiedź konieczny do obsługi całego procesu realizacji zapytania, dodatkowe obciążenie źródeł danych, które może obniżać efektywność wykonywania ich normalnych zadań, oraz fakt, iż system nie może funkcjonować poprawnie, jeżeli jedno albo więcej źródeł danych jest w danym momencie niedostępne.

Zaletą systemu mediacyjnego jest fakt, iż zapytanie użytkownika jest zawsze realizowane na danych aktualnych, w źródłach danych, a sam system mediacyjny przechowuje tylko bardzo niewiele własnych danych koniecznych do jego działania.

3.11.2 Hurtownie danych

Hurtownie danych, w przeciwieństwie do systemów mediacyjnych, składują w centralnym repozytorium dane pobrane i zintegrowane z wielu źródeł danych. Repozytorium to nazywane jest hurtownią danych i, z punktu widzenia użytkowników, nie różni się niczym od klasycznej bazy danych. Na rysunku 8 przedstawiono podstawową architekturę systemu hurtowni danych.



Rys. 8. Architektura systemu hurtowni danych

Na najniższym poziomie architektury znajdują się źródła danych. Źródła te mogą zawierać dane heterogeniczne, a nawet dane niestrukturalizowane. Źródła danych są, z reguły, rozproszone i autonomiczne. Podobnie jak w systemach mediacyjnych, w komunikacji źródeł danych z kolejnymi warstwami systemu pośredniczą konwertery, które są odpowiedzialne za transformację danych z formatu wykorzystywanego w źródle danych do wspólnego schematu i modelu wykorzystywanego w hurtowni danych. Konwertery w hurtowniach danych, w przeciwieństwie do analogicznych rozwiązań w systemach mediacyjnych, nie tłumaczą dowolnych zapytań na zapytania kierowane do źródeł danych, a jedynie odczytują pewien predefiniowany podzbiór danych zawarty w źródle. Odczytane i przekonwertowane dane są, następnie, przekazywane do integratora, który jest odpowiedzialny za załadowanie danych do hurtowni. Proces ten może wymagać, między innymi, filtrowania danych, łączenia danych z danymi pochodzącymi z innych źródeł, czy agregowania danych. Kolejnym elementem systemu hurtowni danych są monitory, których zadaniem jest wykrywanie zmian w źródłach danych. W przypadku wystąpienia zmian w źródle danych, od momentu ostatniego

pobrania danych, monitor źródła uaktywnia konwerter, który pobiera kolejna porcję ze źródła danych. Na najwyższym poziomie architektury hurtowni danych znajdują się składnice danych (ang. *data marts*). Składnica danych jest małą hurtownią, często nazywaną departamentową hurtownią danych, zawierającą podzbiór danych wywiedziony z głównej hurtowni danych. Dane w składnicy danych są wysoce zagregowane. Składnice danych realizują następujące cele. Po pierwsze, udostępniają dane, którymi może być zainteresowana określona grupa użytkowników, zmniejszając obciążenie podstawowej hurtowni danych. Po drugie, mogą one udostępniać dane na wyższym poziomie zagregowania danych, co może być wymagane dla potrzeb metod analizy danych.

Hurtownie danych stanowią jądro większości systemów wspomagania podejmowania decyzji przy zarządzaniu przedsiębiorstwami. Obecnie obserwowany jest gwałtowny wzrost liczby produktów pozwalających na tworzenie hurtowni danych i świadczonych przez te produkty usług, jak i akceptacji tej technologii w przemyśle. Technologia hurtowni danych znalazła zastosowanie, między innymi, w: handlu (do składowania transakcji dokonywanych w kasach sklepowych w celu późniejszej konstrukcji profili klientów oraz zarządzania magazynami), produkcji (w celu obsługi realizacji zamówień i obsługi technicznej klientów), usługach finansowych (analiza ryzyka, wykrywanie oszustw), opiece zdrowotnej (analiza wydatków) oraz integracji danych. Wszechstronna analiza organizacji, w tym jej wymagań i trendów na rynku, wymaga dostępu do wszystkich danych organizacji, gdziekolwiek by one nie były zlokalizowane. Co więcej, konieczny jest dostęp nie tylko do danych aktualnych, ale również danych historycznych. Hurtownia danych, przechowująca zarówno dane historyczne, jak i dane aktualne, idealnie nadaje się do tych celów. Hurtownia danych pozwala na przechowywanie i zarządzanie danymi oraz na wykonywanie skomplikowanych zapytań, których realizacja może wymagać dostępu do milionów rekordów i wykonywania dużej liczby kosztownych operacji. Aby ułatwić złożoną analizę i wizualizację wyników zapytań, hurtownia danych często wykorzystuje wielowymiarowy model danych (tak zwane kostki danych), które wspierają model przetwarzania analitycznego on-line (ang. *On-Line Analytical Processing* – OLAP), którego wymagania funkcjonalne i

wydajnościowe różnią się w znacznym stopniu od wymagań tradycyjnego przetwarzania transakcyjnego. Wśród nowych trendów pojawiających się w technologii hurtowni danych znajdują się próby adaptacji i integracji aktywnych i temporalnych technologii baz danych oraz próby rozszerzenia technologii hurtowni danych o przetwarzanie transakcyjne.

4. Eksploracja danych

Intensywny rozwój technologii generowania, gromadzenia i przetwarzania danych, z jednej strony, z drugiej, upowszechnienie systemów informatycznych, związane ze wzrostem świadomości użytkowników i znaczącym spadkiem cen sprzętu komputerowego, zaowocowały nagromadzeniem olbrzymich wolumenów danych przechowywanych w bazach danych, hurtowniach danych i różnego rodzaju repozytoriach danych. Na ten bezprecedensowy wzrost rozmiaru wolumenów gromadzonych danych, w ostatnich latach, złożyło się szerokie upowszechnienie narzędzi cyfrowego generowania danych (kodów paskowych, kart płatniczych, aparatów cyfrowych, poczty elektronicznej, sieci RFID, edytorów tekstu, itp.) oraz pojawienie się pojemniejszych i tańszych pamięci masowych. Według raportu UC Berkeley [Berk03], w samym 2002 roku wygenerowano 5 EB nowych danych. Według przedstawionego raportu, od 2002 roku obserwujemy około 30\% przyrost nowych danych rocznie. Sama poczta elektroniczna generuje, jak się szacuje, około 400 000 TB nowych danych rocznie. Dla porównania, zbiory biblioteki Kongresu USA zawierają około 10 TB danych. Co ciekawe, raport szacuje, że około 90% nowych danych jest gromadzonych na nośnikach magnetycznych, tylko niewielka część nowych danych jest gromadzona na innych nośnikach (film - 7%, papier - 0,01%, nośniki optyczne - 0,002%). Największym "`producentem danych'" są Stany Zjednoczone -- szacuje się, że produkują one około 40% wszystkich danych światowych.

Głównym źródłem danych, co oczywiste, jest bieżąca działalność przedsiębiorstw i instytucji: banków, ubezpieczalni, sieci handlowych, urzędów administracji publicznej i samorządowej, itp. Innym dostawcą danych są ośrodki naukowe, które generują olbrzymie ilości danych w każdej niemalże dziedzinie

naukowej (fizyka, astronomia, biologia, nauki techniczne, itp.). Wiele firm, przedsiębiorstw, instytucji administracji publicznej, ośrodków naukowych, dysponuje bazami i hurtowniami danych o rozmiarach sięgających 20 -- 30 TB. Największym repozytorium danych jest w chwili obecnej, oczywiście, sieć Web, zawierająca miliardy stron internetowych. Archiwum internetowe (Internet Archive), utworzone w 1996 r., zgromadziło do chwili obecnej ponad 300 TB danych multimedialnych.

Nasuwa się naturalne pytanie o celowość przechowywania tak olbrzymich wolumenów danych. Okazuje się, jak wynika z przeprowadzonych badań, że tylko niewielka część zgromadzonych danych jest analizowana w praktyce. Wiele firm i przedsiębiorstw dysponujących zasobami danych, przechowywanych w zakładowych bazach i hurtowniach danych, stanęło przed problemem w jaki sposób efektywnie i racjonalnie wykorzystać nagromadzoną w tych danych wiedzę dla celów wspomagania swojej działalności biznesowej. Przykładowo, nawet niewielkie sieci supermarketów rejestrują codziennie sprzedaż tysięcy artykułów w kasach fiskalnych. Czy można wykorzystać zgromadzone dane o transakcjach, aby zwiększyć sprzedaż i poprawić rentowność? Jak już wspomnieliśmy wcześniej, zdecydowana większość danych jest gromadzona na nośnikach magnetycznych w systemach baz i hurtowni danych. Tradycyjny dostęp do tych danych sprowadza się, najczęściej, do realizacji prostych zapytań poprzez predefiniowane aplikacje lub raporty. Sposób w jaki użytkownik korzysta i realizuje dostęp do bazy danych nazywamy modelem przetwarzania. Tradycyjny model przetwarzania danych w bazach danych -- "przetwarzanie transakcji w trybie on-line" (ang. on-line transaction processing) (OLTP), jest w pełni satysfakcjonujący w przypadku bieżącej obsługi działalności danej firmy, dla dobrze zdefiniowanych procesów (obsługa klienta w banku, rejestracja zamówień, obsługa sprzedaży, itp.). Model ten dostarcza efektywnych rozwiązań dla takich problemów jak: efektywne i bezpieczne przechowywanie danych, transakcyjne odtwarzanie danych po awarii, optymalizacja dostępu do danych, zarządzanie współbieżnością dostępu do danych, itd. W znacznie mniejszym stopniu model OLTP wspomaga procesy analizy danych, agregacji danych, wykonywania podsumowań, optymalizacji złożonych zapytań formułowanych ad hoc, czy wreszcie aplikacji wspomagających

podejmowanie decyzji. Wspomniany wcześniej model przetwarzania danych w hurtowniach danych ma na celu wspomaganie procesów podejmowania decyzji,

Analiza danych w hurtowni danych, zgodnie z modelem OLAP, jest sterowana całkowicie przez użytkownika. Użytkownik formułuje zapytania i dokonuje analizy danych zawartych w hurtowni. Z tego punktu widzenia, OLAP można interpretować jako rozszerzenie standardu języka dostępu do baz danych SQL o możliwość efektywnego przetwarzania złożonych zapytań zawierających agregaty. Niestety, analiza porównawcza zagregowanych danych, która jest podstawą modelu OLAP, operuje na zbyt szczegółowym poziomie abstrakcji i nie pozwala na formułowanie bardziej ogólnych zapytań: jakie czynniki kształtują popyt na produkty? Czym różnią się klienci supermarketu w Poznaniu i Warszawie? Jakie produkty kupują klienci supermarketu najczęściej wraz z winem? Jakie oddziały supermarketu miały "`anormalną'" sprzedaż w pierwszym kwartale 2007 r.? Czy można przewidzieć popyt klientów na określone produkty? Czy istnieje korelacja pomiędzy lokalizacją oddziału supermarketu a asortymentem produktów, których sprzedaż jest wyższa od średniej sprzedaży produktów? Co więcej, "`ręczny'" charakter analizy OLAP uniemożliwia automatyzację procesu analizy i ogranicza tym samym zakres tej analizy.

Odpowiedzią na potrzebę bardziej zaawansowanej i automatycznej analizy danych, przechowywanych w bazach i hurtowniach danych, jest technologia eksploracji danych (ang. data mining). Zadaniem metod eksploracji danych, nazywanej również odkrywaniem wiedzy w bazach danych (ang. knowledge discovery in databases, database mining), jest automatyczne odkrywanie nietrywialnych, dotychczas nieznanych, zależności, związków, podobieństw lub trendów -- ogólnie nazywanych *wzorcami* (ang. patterns) -- w dużych repozytoriach danych. Odkrywane w procesie eksploracji danych wzorce mają, najczęściej, postać reguł logicznych, klasyfikatorów (np. drzew decyzyjnych), zbiorów skupień, wykresów, itp. Automatyczna eksploracja danych otwiera nowe możliwości w zakresie interakcji użytkownika z systemem bazy i\lub hurtownią danych. Przede wszystkim, umożliwia formułowanie zapytań na znacznie wyższym poziomie abstrakcji aniżeli pozwala na to standard SQL. Analiza danych sterowana zapytaniami (OLAP) zakłada, że użytkownik, po pierwsze, posiada

pełną wiedzę o przedmiocie analizy, i, po drugie, potrafi sterować tym procesem. Eksploracja danych umożliwia analizę danych dla problemów, które, ze względu na swój rozmiar, są trudne do przeprowadzenia przez użytkownika, oraz tych problemów, dla których nie dysponujemy pełną wiedzą o przedmiocie analizy, co uniemożliwia sterowanie procesem analizy danych. Celem eksploracji danych jest, przede wszystkim, poznanie i zrozumienie analizowanych procesów i generowanych przez nie danych.

Eksploracja danych jest dziedziną informatyki, która integruje szereg dyscyplin badawczych, takich jak: systemy baz i hurtowni danych, statystyka, sztuczna inteligencja, uczenie maszynowe i odkrywanie wiedzy, obliczenia równoległe, optymalizacja i wizualizacja obliczeń, teoria informacji, systemy reputacyjne. Powyższa lista dyscyplin nie jest pełna. Eksploracja danych wykorzystuje również szeroko techniki i metody opracowane na gruncie systemów wyszukiwania informacji, analizy danych przestrzennych, rozpoznawania obrazów, przetwarzania sygnałów, technologii Web, grafiki komputerowej, bioinformatyki. Jakie dane podlegają eksploracji danych? Początkowo, eksploracji poddawano proste typy danych (liczby, łańcuchy znaków, daty), przechowywane w plikach płaskich oraz relacyjnych bazach danych. Wraz z rozwojem narzędzi do generowania i przechowywania danych, z jednej strony, z drugiej, z rozwojem technologii eksploracji danych, eksploracji poddawane są coraz bardziej złożone typy danych: multimedialne (zdjęcia, filmy, muzyka), przestrzenne (mapy), tekstowe i semistrukturalne, przebiegi czasowe, sekwencje danych kategorycznych, grafy, struktury chemiczne (sekwencje DNA), sieci socjalne.

Z pewnego punktu widzenia, można rozpatrywać eksplorację danych jako zaawansowane wykonywanie zapytań w bazach i hurtowniach danych, gdyż otrzymana w wyniku zapytań eksploracyjnych wiedza w rzeczywistości jest zawarta w danych składowanych w bazie lub hurtowni danych, choć trudnym jest pozyskanie tej wiedzy „ręcznie”. Dlatego też, jedną z bardziej interesujących koncepcji w obecnych czasach jest zintegrowanie metod eksploracji danych i narzędzi z baz i hurtowni danych, aby móc korzystać z zalet obydwu tych podejść w procesie odkrywania wiedzy. Związane jest to z koniecznością opracowania nowych typów indeksów wspierających algorytmy eksploracji danych,

opracowania metod przetwarzania współbieżnego danych, wizualizacji wyników eksploracji danych, języków zapytań pozwalających na wykonywanie zapytań eksploracyjnych ad-hoc itp. Takie zintegrowane podejście nazwano *On-Line Data Mining* – OLAM. Oczekuje się, iż w niedalekiej przyszłości integracja baz danych i hurtowni danych z eksploracją danych utworzy nowy typ systemu baz danych, który potrafi składować i zarządzać danymi oraz wiedzą z nich wyekstrahowaną.

Technologia eksploracji danych może być użyta w wielu różnych dziedzinach zastosowań [HK00]: przemyśle, marketingu, produkcji, usługach finansowych, telekomunikacji, opiece zdrowotnej, badaniach naukowych, a nawet w sporcie. Eksploracja danych jest teraz jedną z najpopularniejszych i najszybciej rozwijających się technologii wyrosłych na gruncie technologii przetwarzania danych i baz danych.

5. Przepływy pracy

Procesy biznesowe często wymagają skoordynowanego wykonania wielu zadań wykonywanych przez różne jednostki przetwarzające, którymi mogą być ludzie, bądź programy, na przykład: systemy zarządzania bazą danych, aplikacje lub system poczty elektronicznej. Przykładowo, prosty proces zawarcia umowy o pożyczkę składa się z kilku kroków. Klient kontaktuje się z odpowiednim pracownikiem banku i wypełnia odpowiedni formularz. Pracownik komunikuje się z kontrolerem należności, aby sprawdzić wiarygodność klienta w bazie danych. Wówczas kontroler decyduje, czy zaakceptować podanie i informuje pracownika o podjętej decyzji, który przekazuje ją klientowi. Również prosty zakup realizowany jest w kilku krokach: zgłoszenie przez klienta chęci wykonania zakupu, oferta, zgoda na ofertę, wysłanie towaru, przygotowanie faktury i zapłata. Obydwa powyższe przykłady demonstrują typowy model przetwarzania: żądanie – odpowiedź nazywany przepływem pracy (ang. *workflow*).

Zarządzanie przepływami pracy w przedsiębiorstwie jest kluczowe dla jego poprawnego i sprawnego działania. Zadanie to jednak jest komplikowane przez fakt, iż wiele organizacji wykorzystuje wiele niezależnie zarządzanych systemów wspierających różne fragmenty przepływu pracy. Oczywistym jest, iż przepływy

pracy wymagają specjalnych sposobów przetwarzania danych, które wspierają sekwencje powiązanych ze sobą zadań. Innymi słowy, przepływy pracy wymagają specjalnego, dedykowanego dla nich „systemu zarządzania przepływami pracy", które wspierają ich specyficzne wymagania. Po pierwsze, procesy przepływu pracy wymagają nowych modeli transakcji, które pozwalają, aby częściowe wyniki przetwarzania były widoczne poza przepływem pracy. Pozwala to na niezależne zatwierdzanie fragmentów przepływu. Procesy przepływów pracy wymagają również specjalnych narzędzi ich specyfikowania, tworzenia i zarządzania. Podsumowując, kompletny, transakcyjny system przepływów pracy powinien wspierać wielozadaniowe, wielo-systemowe czynności, w których: (1) różne zadania mogą mieć różne własności i zachowania podczas realizacji, (2) zadania mogą być wykonywane przez różne jednostki przetwarzające, (3) wspierana jest koordynacja wykonywania różnych zadań zarówno przez człowieka, jak i przez aplikacje programowe, i (4) obsługa awarii zgłaszanych zarówno przez aplikację, jak i przez użytkownika, pozwala na wycofywanie niekompletnie wykonanych zadań. należy się spodziewać, iż w niedalekiej przyszłości modele tworzenia aplikacji będą ewoluowały w kierunku możliwości przetwarzania transakcji i przepływów pracy tak, aby odpowiadały one potrzebom złożonych aplikacji wykorzystujących różne systemy.

6. Wpływ systemów baz danych na rozwój informatyki w Polsce

Systemy baz danych miały, i mają nadal, zasadniczy wpływ na rozwój informatyki w świecie i w Polsce. Wpływ ten wiąże się, w podstawowym stopniu, z wpływem rozwoju technologii baz i hurtowni danych na rozwój systemów informatycznych. Zdecydowana większość stosowanych aktualnie systemów informatycznych: systemów bankowych, systemów rezerwacji lotniczej, hotelowej i kolejowej, administracyjnych, systemów gospodarki materiałowej i magazynowej, systemów informatycznych w służbie zdrowia, systemów bibliotecznych, systemów uczelnianych, itd., jest budowana z wykorzystaniem systemów baz danych. Systemy baz danych pełnią, w systemach informatycznych, rolę jądra odpowiedzialnego za: bezpieczne przechowywanie danych, efektywne

wyszukiwanie danych, zapewnienie mechanizmów autoryzacji dostępu do danych, odtwarzanie danych po awarii, czy zapewnienie współbieżnego dostępu do danych. Jednym z pierwszych systemów baz danych był hierarchiczny system bazy danych IMS (Information Management System**)** opracowywany wspólnie przez IBM i firmy Rockwell i Caterpillar, od roku 1966, na potrzeby tworzonego wówczas programu kosmicznego Apollo. System jest rozwijany do dzisiaj i obecnie obsługuje m.in. takie standardy, jak Java, JDBC, XML Web services. Pierwsze, polskie systemy informatyczne powstałe w ośrodkach ZETO, takie jak EMIR czy RENTIER, których zadaniem była obsługa naliczania i wypłacania rent i emerytur, powstawały z wykorzystaniem jednego z pierwszych, komercyjnych systemów „relacyjnych" baz danych ADABAS firmy Software A.G. W późniejszym okresie, w Polsce, powstały własne, oryginalne systemy zarządzania bazami danych JANTAR oraz RODAN (system sieciowej bazy danych), eksploatowane z powodzeniem w systemach PESEl oraz polskich fabrykach (np. w odlewni w Śremie czy Wiepofamie w Poznaniu). I ile w początkowym okresie rozwoju technologii systemów baz danych systemy te były wykorzystywane do budowy systemów informatycznych przetwarzających wyłącznie proste dane liczbowe i tekstowe, o tyle w późniejszym czasie, nowe typy systemów baz danych zostały wykorzystane do implementacji geograficznych systemów informatycznych (GIS), przestrzenno-czasowych, systemów przetwarzania mobilnego (SAP), czy budowy systemów obiektowych i multimedialnych. Aktualnie, nowoczesne systemy baz danych są szeroko wykorzystywane w świecie Internetu, głównie do budowy systemów wykorzystywanych w handlu elektronicznym (Amazon, Merlin), aukcjach internetowych (Allegro, e-Bay), czy do budowy portali internetowych. Systemy hurtowni danych, jak wspominaliśmy wcześniej, stanowią jądro większości systemów wspomagania podejmowania decyzji (systemy DSS). Obecny rozwój systemów baz danych i hurtowni danych, jak również innych dziedzin związanych z szeroko rozumianą technologią przetwarzania danych, jest „napędzany", głównie, potrzebami nowych aplikacji i nowych zastosowań, które pojawiają się na rynku m. in. wraz z rozwojem semantycznej sieci Web, sieci społecznościowych, serwisów Facebook, Flickr, itp.

Literatura

[AW98] M. Abdelguerfi, K-F. Wong (ed.), Parallel database techniques IEEE Computer Press, Los Alamitos, 1998

[BBC+98] P. Bernstein, M. Brodie, S. Ceri et al., The Asilomar report on database research, SIGMOD Record 27(4), 1998, 74-80.

[BEP+00] J. Błażewicz, K. Ecker, B. Plateau, D. Trystram (ed.) Handbook on parallel and distributed processing, Springer-Verlag, 2000.

[Berk03] Berkeley Report, How Much Information? 2003, http://www.sims.berkeley.edu/research/projects/how-much-info-2003/, 2003.

[Codd70] E. F. Codd, A Relational Model of Data for Large Shared Databanks, CACM 13.6, Czerwiec 1970.

[DBS90] Database systems: Achievements and Opportunities, Report of the NSF Invitational Workshop on Future Directions in DBMS Research, Luty 1990.

[EN94] R. Elmasri, S. B. Navathe Fundamentals of Database Systems, Second Edition, The Benjamin/Cummings Publishing Company, 1994

[Gray96] J. Gray, Evolution of data management, IEEE Computer 29(10), 1996, 38-46.

[HK00] J. Han, M. Kamber, Data mining: concepts and techniques, Morgan Kaufman Pub., 2000.

[Kim95] W. Kim, Modern database systems: the object model, interoperability and beyond, ACM Press, Nyew York, 1995

[Lew06] M. Lew, et al., Content-based Multimedia Information Retrieval: State of the Art and Challenges, ACM Transactions on Multimedia Computing, Communications, and Applications 2(1), 2006.

[Mart04] J. M. Martínez, MPEG-7 Overview (version 10), ISO, 2004.

[ME01] J. Melton, A. Eisenberg, SQL Multimedia and Application Packages (SQL/MM), SIGMOD Record 30(4), 2001.

[ME99] J. Melton, A. Eisenberg, SQL:1999, formerly known as SQL3, SIGMOD Record 28(1), 1999.

[Olle06] T. W. Olle. Nineteen sixties history of database management, IFIP International Federation for Information Processing, vol. 215/2006, 67-75.

[S84] J. Shurkin, Engines of the Mind: A History of the Computer, W.W. Norton & Co. 1984.

[SSU91] A. Silberschatz, M. J. Stonebraker, J. Ullman, Database systems: achievements and opportunities, SIGMOD Record 19(4), 1991, 6-22 (also in Communication of the ACM 34(100), 1991, 119-120).

[SSU96] A. Silberschatz, M. J. Stonebraker, J. Ullman (eds.), Database research: achievements and opportunities into the 21st Century, SIGMOD Record 25(1), 1996, 52-63.

[Subr98] V.S. Subrahmanian, Principles of Multimedia Database Systems, Morgan Kaufmann, 1998.

[SZ96] A. Silberschatz, S. B. Zdonik, Strategic directions in database systems - breaking out of the box, ACM Computing Surveys 28(4), 1996, 764-778.

[Voss91] G. Vossen, Data Models, Database Languages and Database Management Systems, Addison-Wesley Pub. Company, 1991.