

kpt. mgr inż. Adam KRASUSKI
prof. dr hab. inż. Tadeusz MACIAK
Szkoła Główna Służby Pożarniczej

HISTORIA ROZWOJU SYSTEMÓW ZARZĄDZANIA BAZAMI DANYCH

Streszczenie

W artykule przedstawiony został rozwój systemów baz danych. Omówiono w nim technologie Hierarchicznych, Sieciowych, Relacyjnych oraz Obiektowych baz danych. Artykuł ma charakter przeglądowy a celem jego jest przybliżenie wiedzy na temat powstawania i kształtowania się współczesnych systemów baz danych. Opis technologii baz danych zaprezentowano w ujęciu historycznym, co zapewni lepsze poznanie materiału czytelnikom nieobeznany z tą technologią.

Summary

This paper outlines the historical development of data management systems. Hierarchical, Network, Relational, Object-oriented and Object-relational data management systems are reviewed. A short summary of related research is given.

1. Wstęp

Historia rozwoju systemów baz danych liczy ponad 40 lat. W tym czasie powstały systemy stanowiące obecnie najprawdopodobniej największe dotychczasowe osiągnięcie inżynierii oprogramowania. Systemy zarządzania bazą danych, popularnie z angielskiego zwane **DBMS** (*ang. Database Management Systems*) stały się podstawą systemów informacyjnych i istotnie zmieniły sposób działania wielu instytucji. Ich rozwój pobudził powstawanie oprogramowania do zarządzania informacjami, w środowisku, którego pracuje się w sposób prosty i naturalny. Niestety pozorna prostota obsługi baz danych niesie za sobą kilka zagrożeń. Jednym z nich jest to,

iż do pracy z aplikacjami operującymi na danych, kierowani są ludzie bez specjalnego przeszkolenia w tym zakresie. Skutkuje to korzystaniem z baz danych w sposób niewłaściwy lub nie wykorzystujący w pełni ich możliwości.

Rozwój **DBMS** wprowadził też istotne zmiany w funkcjonowaniu Państwowej Straży Pożarnej (**PSP**). Do codziennych obowiązków funkcjonariuszy **PSP** należy wprowadzanie meldunków do bazy danych, sporządzanie zestawień sprzętu i ludzi będących w podziale bojowym oraz wiele innych podobnych operacji. W sytuacjach wyjątkowych, wykorzystywane są chemiczne bazy danych, systemy wspomaganie decyzji **DSS** (*ang. Decision Support Systems*) czy systemy informacji przestrzennej **GIS** (*ang. Geographic Information Systems*). Podstawę *Historia rozwoju Systemów zarządzania bazami danych* prawidłowego działania tych aplikacji stanowi przetwarzanie danych. W związku z tym, iż w **PSP** wykorzystywane są zaawansowane technologie z dziedziny baz danych, świadomość pracowników odnośnie ich obsługi musi być, więc odpowiednio wysoka celem wykorzystania możliwości tych systemów.

Poniższy artykuł ma charakter przeglądowy a celem jego jest przybliżenie wiedzy na temat powstawania i kształtowania się współczesnych systemów baz danych. Opis technologii baz danych zaprezentowano w ujęciu historycznym co zapewni lepsze poznanie materiału czytelnikom nieobeznanym z tą technologią. Artykuł stanowić może wprowadzenie do technologii systemów zarządzania bazami danych dla funkcjonariuszy **PSP**.

2. Wprowadzenie

Istnieje wiele definicji określających, czym jest baza danych. Dodatkowo ciągle zmiany w tej dziedzinie rozszerzają znaczenia tego słowa o nowe obszary wiedzy. Baza danych nie koniecznie musi stanowić zasoby przechowywane w formie elektronicznej. Jednak w odwołaniu do informatyki można ją zdefiniować jako zbiór danych przechowywanych na komputerze w formie ustrukturyzowanej, pozwalającej systemowi komputerowemu ją obsługującego na udzielanie odpowiedzi dotyczących przechowywanych danych. System komputerowy obsługujący zasoby bazy nazwany jest "systemem zarządzania bazą danych" [13]. Jednakże na skutek skrótów myślowych oraz wygody, oprogramowanie to bardzo często nazywane jest również systemem baz danych lub po prostu "bazą danych".

Ze względu na rodzaj danych oraz sposób ich przechowywania i reprezentowania powstały różne opisy strukturalne zasobów bazy. Opisy te nazywane są schematami bazy danych [2].

Schemat bazy opisuje obiekty przechowywane w bazie a także relacje między nimi. Istnieje kilka sposobów organizacji schematów bazy danych. Budowa takiego schematu nazywana jest modelowaniem bazy danych. Sposoby modelowania baz danych zmieniały się wraz z rozwojem technologicznym. Powstawały kolejno modele baz danych: płaski (*ang. flat file*), hierarchiczny, sieciowy, relacyjny i obiektowy [1].

Bazy danych operują głównie na danych tekstowych i liczbowych. Większość współczesnych baz umożliwia również przechowywanie danych binarnych typu: grafika, dźwięk lub inne.

Do podstawowych funkcji, jakie powinien realizować system baz danych zaliczyć można [1]:

- Definiowanie bazy danych, czyli określenie struktury danych, typu przechowywanych danych oraz relacji między nimi;
- Umożliwienie dopisywania, modyfikacji oraz pozyskiwania danych z bazy;
- Kontrola dostępu do zasobów bazy, zapewnienie bezpieczeństwa danych;
- Dbanie o integralność i spójność danych;
- Umożliwienie i kontrolowanie wielodostępu do danych;
- Odtwarzanie danych po awarii sprzętu czy oprogramowania.

Wykorzystywane obecnie systemy zarządzania bazami danych są w stanie realizować wyżej wymienione funkcje. Osiągnięcie tych funkcjonalności poprzedzał jednak burzliwy rozwój technologiczny.

3. Historia rozwoju DBMS

Początki technologii baz danych sięgają urządzeń mechanicznych sterowanych perforowanymi metalowymi płytkami. Około roku 1725 sterowanie takie wykorzystywane było do kontroli pracy maszyn tkackich w zakładach Jacquarta [4]. W systemach tych otwory w płytkach pozycjonowały odpowiednie mechaniczne przełączniki sterujące pracą krosien (rys. 1).



Rysunek 1. Perforowane metalowe płytki używane w warsztatach tkackich Jacquarta.

Źródło: [4]

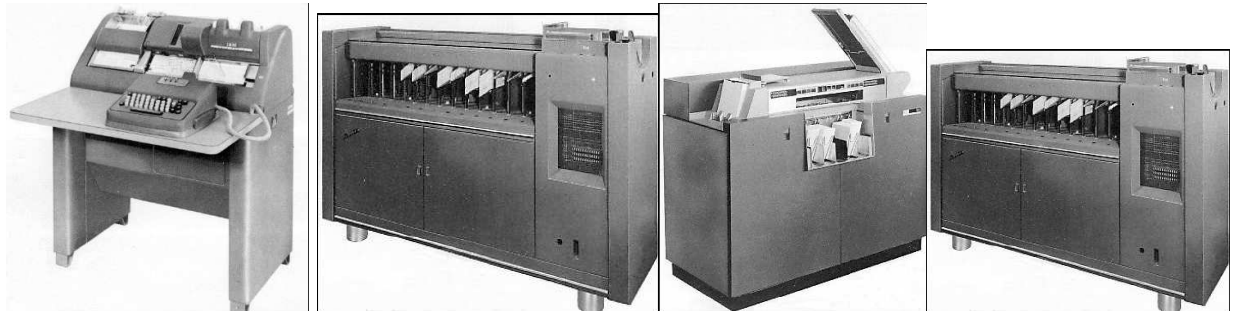
Technologia płytek perforowanych stała się podstawą do budowy maszyny licząco-analitycznej. Maszyna ta używała kart perforowanych jako nośników informacji. Dane na karcie zapisywane były z użyciem specjalnego kodu. Maszyna potrafiła sortować i wyszukiwać dane zgodnie z założonymi regułami oraz dokonywać porównań informacji zamieszczonych na kartach [5]. Konstrukctorem maszyny licząco-analitycznej – popularnie zwanej tabulatorem – był Herman Hollerith. Wykorzystano ją po raz pierwszy w Komisji ds. Zdrowia (*ang. Board of Health*) w Nowym Jorku [6]. Po wstępnych testach maszyna okazała się bardzo użyteczna w wykonywaniu żmudnych zadań sortowania oraz liczenia danych. Zostało to zauważone przez amerykańską administrację i w roku 1890 zmodernizowana maszyna Holleritha została użyta podczas narodowego spisu ludności w US . Wspomagała ona opracowanie wyników spisu [7].

Maszyna licząco-analityczna była urządzeniem elektromechanicznym i zazwyczaj w jej skład wchodziły następujące moduły [5]:

- perforatory kart, służące do wybijania otworów w kartach (ręczne lub automatyczne), jak również wyprowadzania wyników obliczeń;
- reproducery kart, służące do kopiowania kart już wydziurkowanych;
- kolatory, służące do porównywania oraz łączenia różnych kart w pliki według określonych reguł;
- sortery, służące do układania kart zgodnie z określonymi regułami;

- tabulatory, służące do drukowania zawartości kart perforowanych w formie tekstu;
- kalkulatory, służące do wykonywania obliczeń arytmetycznych;
- czytniki kart perforowanych, służące do wprowadzania danych do systemu.

Przykładowe moduły maszyny licząco-analityczne pokazano na rysunku nr 2.



Rysunek 2. Maszyny licząco-analityczne. Od lewej: perforator, reproducer, kalkulator, sorter. Źródło: [5]

3.1. Płaskie pliki

Wraz z pojawieniem się pierwszych systemów komputerowych rolę maszyn licząco-analitycznych przejęły komputery. Niemniej dane nadal zapisywane były na kartach perforowanych.

System pracy komputerów bazował na operacjach wsadowych [8]. Dane do programu oraz lista rozkazów (tzw. wsad) przygotowane były na kartach i wprowadzane do komputera. Po wykonaniu zaprogramowanych operacji pojawiały się informacje wyjściowe zawierające wynik działania programu, a niekiedy obraz jego pamięci – jeśli działanie programu zakończyło się błędem.

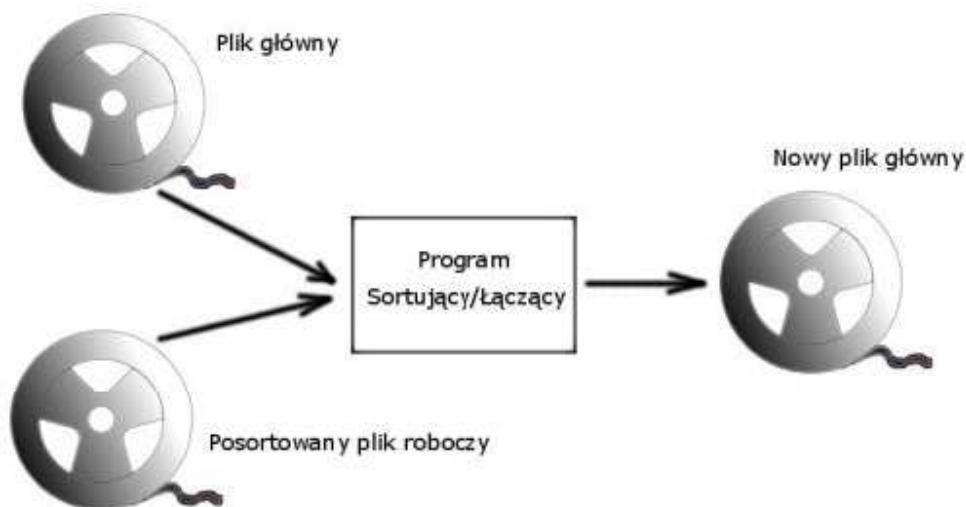
Informacje przechowywane były w tzw. „płaskich plikach”. Wszystkie dane zawarte były w jednym pliku a ich interpretacji dokonywała aplikacja zewnętrzna [9]. Strukturę danych przechowywanych w płaskich plikach przedstawia rysunek

3.

Imię	Nazwisko	Biuro	Miasto	Ulica
Jan	Kowalski	Delpol	Warszawa	Radarowa
Jan	Nowak	Delpol	Warszawa	Radarowa
Jan	Wiśniewski	Delpol	Warszawa	Radarowa
Janina	Kowalska	Elpol	Poznań	Radiowa
Janina	Kowalska	Delpol	Warszawa	Radarowa

Rysunek 3: Logiczna struktura danych przechowywanych w plikach płaskich

Zapis oraz uaktualnianie zawartości plików opierał się na fizyczno-sekwencyjnej metodzie zapisu [10]. Metoda ta polegała na przechowywaniu głównego pliku z danymi (*ang. Master file*), posortowanego według pewnego klucza oraz pliku, na którym przeprowadzano modyfikacje zawartości (*ang. Transaction file*). Co jakiś czas dane z pliku roboczego przenoszono do pliku głównego, sortowano ponownie i zapisywano. Rysunek nr 4 przedstawia koncepcję fizyczno-sekwencyjnej metody zapisu.



Rysunek 4. Koncepcja fizyczno-sekwencyjnego zapisu, wg [11]

Wynalezienie magnetycznych nośników pamięci wywołało prawdziwą rewolucję w systemach gromadzenia danych [9]. W porównaniu z kartami perforowanymi można było zapisać na magnetycznych nośnikach bardzo duże ilości danych. Dodatkowo podzielenie taśmy

na strony i przydzielenie im unikalnych adresów umożliwiło odejście od sekwencyjnego zapisu danych.

Korzystając z odpowiedniej funkcji haszującej [1] rozmieszczano rekordy na nośnikach w sposób odpowiadający rozkładowi jednostajnemu. Adres strony, do której miał być wpisany rekord wyliczany był za pomocą wyżej wymienionej funkcji. Analogicznie było z wyszukiwaniem danych, zamieniano wyszukiwane słowo kluczowe na fizyczny adres obszaru dysku lub taśmy. W obszarze tym przechowywane były informacje dotyczące danego słowa. Jedną z pierwszych szeroko wykorzystywanych metod pozyskiwania danych z nośników magnetycznych była metoda o nazwie **BDAM** (*ang. Basic Direct Access Method*) wykorzystująca wyżej opisaną funkcję haszującą [10].

Jednak metody te, mimo, iż relatywnie szybkie (komputery typu mainframe potrzebowały zaledwie 50 ms, aby zamienić słowo kluczowe na adres fizyczny) posiadały kilka wad, między innymi [1, 9, 10]:

- Brak możliwości implementacji relacji pomiędzy danymi. W płaskich plikach nie było możliwości tworzenia powiązań pomiędzy poszczególnymi plikami. Utrudniało to tworzenie struktury bazy danych odwzorowującej strukturę świata rzeczywistego.
- Powstawanie tzw. „wysp informacji”. Dane pomiędzy oddziałami firmy nie były wymieniane. Każdy dział posiadał swój własny plik i przeważnie inną aplikację napisaną do jego obsługi.
- Powtarzanie tych samych danych. Niewymienialność danych pomiędzy wydziałami powodowała, iż były one dublowane.
- Zależność aplikacji od struktury danych. Istniała ścisła zależność pomiędzy plikiem a aplikacją ją obsługującą. W przypadku zmiany struktury pliku należało również przeprogramować aplikację.
- Brak mechanizmów jednoczesnego dostępu do danych przez kilku użytkowników oraz mechanizmów przywracania zawartości bazy po awarii.
- Brak możliwości rozwiązania problemu relacji pomiędzy danymi.

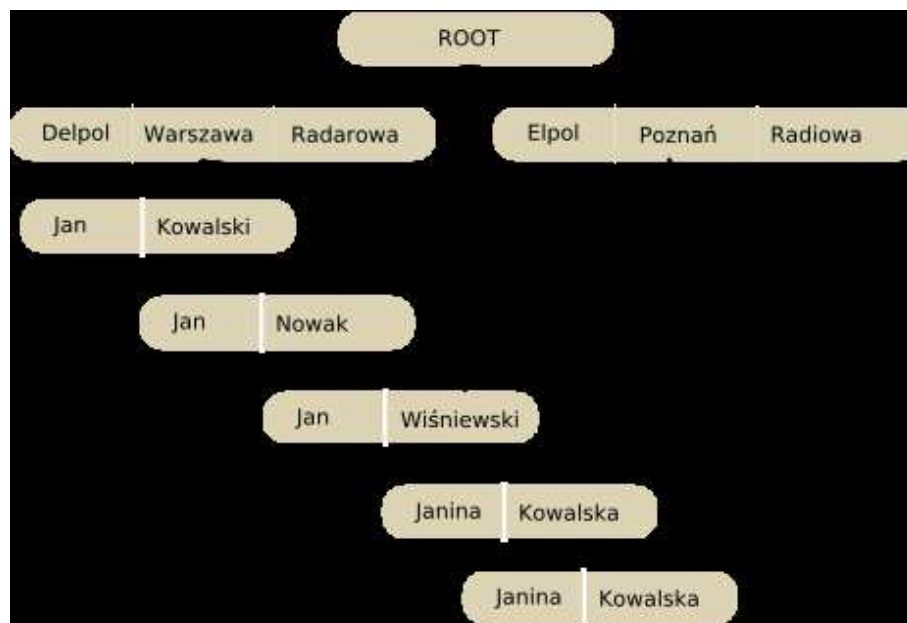
Problemy związane z systemem płaskich plików zapoczątkowały rozwój nowego oprogramowania – systemów zarządzania bazą danych.

3.2. Hierarchiczne bazy danych

Za pierwszy DBMS uważane jest oprogramowanie do obsługi projektu Apollo, który powstał na początku lat 60tych [1]. Związany on był z zadaniem postawionym przez Prezydenta Kennedy'ego lotu człowieka na księżyc. Systemy bazujące na płaskich plikach nie były w stanie zgromadzić i nadzorować takiej ilości danych powstających w związku z tym projektem. W rezultacie North American Aviation (NAA), główny wykonawca projektu, stworzyła system dostępu i modyfikacji danych znany jako **GUAM** (ang. *Generalized Update Access Method*).

Struktura **GUAM** powstawała z małych elementów łączonych stopniowo w większe komponenty, aż do powstania końcowego systemu. W wyniku tego zbudowano drzewiastą strukturę bazy danych. W połowie lat 60tych do projektu NAA przyłączyła się firma **IBM** w efekcie, czego na bazie **GUAM** powstał system **IMS** (ang. *Information Management System*).

Sposób zapisu danych w IMS opierał się na modelu hierarchicznym [1,912]. Przypominało to obecny schemat zapisu systemu plików. Struktura hierarchiczna zakładała, iż związki pomiędzy danymi mogły zachodzić tylko typu 1:N (rysunek nr 5).



Rysunek 5. Hierarchiczny model danych, wg [11]

Choć dopuszczano bliźniacze powiązania między rekordami podstawę struktury bazy stanowiły relacje w kategoriach ojciec-syn,[12]. Oznaczało to, iż rekord typu ojciec mógł być powiązany z wieloma rekordami typu syn. Sposób ich powiązania mógł być jawny, przez wskaźniki, lub poprzez fizyczną organizację rekordów wewnątrz pliku. Zmienna „poziom”

w modelu hierarchicznym definiowała jak głęboko dany rekord występował w strukturze drzewa [11]. Korzeń (*ang. root*) struktury drzewa miał numer 1. Następujące po nim rekordy miały kolejne numery. W modelu hierarchicznym dostęp do danych odbywał się po zdefiniowanych ścieżkach. Ścieżka dostępu jest to sekwencja rekordów począwszy od korzenia poprzez kolejne rekordy typu syn.

Do zalet modelu hierarchicznego można zaliczyć prostą i przejrzystą strukturę. Stanowiło to duże uproszczenie w obsłudze baz danych, których podstawą była nawigacja. Niemniej taki sposób opisu danych posiadał liczne ograniczenia. Jednym z nich był problem zaprojektowania bazy w przypadku, gdy rzeczywista struktura danych nie odzwierciedlała drzewa. W takich przypadkach dane były dublowane celem zachowania struktury związków jeden do wielu [9, 10]. Powodowało to problemy z zachowaniem spójności danych. Bardzo łatwo, bowiem mogło dojść do sytuacji, w której jeden ze zdublowanych rekordów został zaktualizowany natomiast odpowiadający mu rekord nadmiarowy pominięty [10]. Zwiększało to ponadto koszty systemu poprzez zakup dodatkowych nośników pamięci.

Wbrew swoim osobliwościom, systemy oparte o model hierarchiczny stanowiły w pewnym momencie znaczącą pozycję w dziedzinie baz danych. Ich sukces wynikał z faktu iż w latach sześćdziesiątych podstawowym nośnikiem informacji były taśmy magnetyczne z sekwencyjnym sposobem dostępu do danych [11]. Umieszczenie na taśmie rekordu rodzica a po nim sekwencji rekordów potomków (z odpowiednio ustawioną zmienną poziom) skutkowało tym, iż informacje z danej gałęzi struktury drzewa przechowywane były blisko siebie w pliku. Było to jednym z czynników decydującym o szybkości działania tych systemów.

IMS umożliwiał również jednoczesny dostęp do danych wielu użytkownikom. Dodatkowo wyposażony został w mechanizmy odzyskiwania danych [10].

IMS nie potrafił dokonywać identyfikacji danych przechowywanych w bazie poza reprezentacją relacji pomiędzy rekordami [9]. Poszczególne pola rekordów nie były identyfikowane przez System zarządzania bazą danych. Rekord widoczny był jako liczba bajtów zawartych w danym polu. W konsekwencji, w systemach tych nie można było realizować zapytań „ad hoc”.

3.3. Sieciowe bazy danych

Ograniczenia wynikające ze struktury hierarchicznej próbowano ominąć tworząc systemy sieciowe. Dawały one możliwość tworzenia związków między danymi typu wiele do wielu (N:N) [9].



Rysunek 6: Przykład stuktury sieciowej z widocznymi związkami N:N

Schemat oraz sposób budowy bazy danych opartej o model sieciowy dawał projektantom dosyć dużą swobodę. W związku z tym pojawiło się szereg rozwiązań promowanych przez poszczególnych producentów. Zazwyczaj jednak wykonywane projekty baz danych były mocno specjalizowane, czyli dostosowywane do specyfiki przechowywanych danych. Projektant tworzący strukturę bazy musiał również przewidzieć sposób, w jaki dane mogą być w przyszłości pozyskiwane z bazy. Aplikacja napisana do współpracy z bazą mogła tylko wykonywać zdefiniowane wcześniej operacje. Utrudniało to w znacznym stopniu przenoszenie danych a także komplikowało wszelkie zmiany w działaniu aplikacji [9].

Bazy danych, hierarchiczne oraz sieciowe, miały charakter nawigacyjny. Implementacja zaprojektowanej bazy danych polegała na fizycznym rozmieszczeniu rekordów zgodnie ze schematem lub utworzeniu odnośników między rekordami. Nawigacja w bazie polegała na poruszaniu się po tych wskaźnikach, przeglądanie poszczególnych rekordów i wybieraniu właściwych [11]. Kiedy baza danych została po raz pierwszy otwierana, program/użytkownik, który do niej się łączył, otrzymywał na wstępie dostęp do jej pierwszego rekordu. Rekord ten zazwyczaj zawierał zarazem wskaźnik do dalszych zasobów bazy. Aby znaleźć określone informacje programista musiał przyglądać poszczególne rekordy dopóki nie trafił na właściwy rekord [10]. Proste zapytanie typu „Znajdź wszystkich ludzi z Polski”, wymagało przejścia określonej ilości zdefiniowanych ścieżek i wybranie rekordów odpowiadających zapytaniu.

W połowie lat 60tych powstała już duża liczba oprogramowania do zarządzania danymi. Każde z nich miało zazwyczaj swój własny sposób implementacji oraz korzystania z zasobów bazy. W związku z tą różnorodnością powstał problem przenośności danych a także współpracy

pomiędzy tymi aplikacjami. Próbę utworzenia standardu w systemach baz danych rozpoczął Charles W. Bachmann (twórca systemu **IDS** ang. *Integrated Data Storage*). Zainicjował on utworzenie wewnątrz organizacji o nazwie **CODASYL** (ang. *Conference on Data Systems Languages*) grupy **DBTG** (ang. *Database Task Group*) [9, 10], która miała się zająć tworzeniem standardów w dziedzinie baz danych.

Wynikiem prac **DBTG** był opublikowany raport, który zdefiniował następujące pojęcia [10]:

- Termin „baza danych” (ustalono, iż w języku angielskim obowiązywać będzie termin „*database*” zamiast „*data base*”).
- Opracowano szkielet „słownika danych”. Słownik danych został zaprojektowany celem przechowywania metadanych, włączając informacje o rodzaju danych, relacjach pomiędzy nimi oraz informacje, w jaki sposób programy powinny korzystać z bazy danych.
- Opracowano założenia standardowej architektury bazy danych, opartej o metodę zapisu **BDAM** oraz listę relacji.
- Zaproponowano separację logicznej struktury bazy od jej fizycznego zapisu oraz sposobu pozyskiwania danych.
- Opracowano procedury mające na celu zapewnienie bezpieczeństwa oraz wielodostępu do bazy. Wprowadzono metodę blokowania modyfikowanych rekordów.
- Zaproponowano koncepcyjny sposób opisu bazy.

Koncepcyjny model danych wymagał aby każda baza danych zawierała logiczny opis swojej zawartości. Opis ten powinien być konstruowany na dwóch płaszczyznach, administratora systemu oraz użytkownika [1]. Wyodrębniono te dwie płaszczyzny ponieważ złożoność modelu sieciowego przysparzała wiele trudności użytkownikom bazy.

Schemat bazy przeznaczony dla administratora został nazwany „schematem sieci” i zawierał definicję bazy, jej nazwę, typy wszystkich rekordów oraz składniki każdego z typów rekordów, Natomiast schemat przeznaczony dla użytkownika otrzymał nazwę „podschematu” i zawierał opis fragmentu bazy, z którego użytkownik lub aplikacja miała korzystać. W zależności o ilości użytkowników lub aplikacji podschematów mogło być wiele.

Dodatkowo grupa **DBTG** zaproponowała zunifikowane języki zarządzania danymi. Pozwalało to między innymi na rozdzielenie funkcji projektanta bazy od programistów a także

zapewniło większą czytelność projektów. Zaproponowany język zarządzania danymi pozwalał opisywać dane, strukturę ich przechowywania a także je przetwarzać. Dzielił się na języki [1]:

- opisu danych schematu sieci **DDL** (*ang. Data Definition Language*),
- opisu danych podschematu,
- manipulacji danymi **DML** (*ang. Data Manipulation Language*), który umożliwiał programistom nawigowanie w strukturze bazy.

Jak wspomniano grupa **DBTG** rozpoczęła również pracę nad dwoma ważnymi aspektami dotyczącymi technologii baz danych: współbieżności oraz bezpieczeństwa [7]. Współbieżność zrealizowano poprzez blokowanie odpowiednich obszarów bazy danych w przypadku równoczesnego dostępu do tego obszaru przez kilku użytkowników. Natomiast bezpieczeństwo umożliwiała ochronę obszarów danych odpowiednimi hasłami.

Pod koniec lat 60tych istniały w przemyśle bazodanowym dwa rodzaje systemów zarządzania bazami danych. Model sieciowy rozwijany i regulowany przez **CODASYL** oraz hierarchiczny promowany przez firmę **IBM** i jej system **IMS**. Jednak te schematy opisu danych nie spełniały wszystkich wymagań stawianych przed **DBMS**. Projektowanie baz danych z wykorzystaniem modelu hierarchicznego czy **CODASYL** nie miało podstaw matematycznych. Jakość zaprojektowanej aplikacji a później sposób jej przeszukiwania zależał od doświadczenia projektantów [8].

3.4.Relacyjne bazy danych

Edgar F. Codd matematyk z Oxfordu postawił sobie problem opracowania założeń **DBMS**, który spełniałby następujące zasady:

- Całkowitą niezależność fizycznego zapisu danych od ich struktury logicznej;
- Matematyczne podstawy w sposobie przechowywania oraz pozyskiwania danych;
- Możliwość zadawania dowolnych zapytań przez użytkowników a nie tylko zdefiniowanych przez projektantów;

Badania przeprowadzone przez Coddę dały początek rozwojowi nowych systemów zarządzania bazami danych. W 1970 Codd opublikował w raportach technicznych **IBM** artykuł pod tytułem „*A Relational Model of Data for Large Shared Data Banks*” dając tym samym podstawy nowej organizacji przechowywania oraz dostępu do danych [13]. Schemat organizacji danych nazwany został przez Coddę modelem relacyjnym. Bazował on w opracowaniu swojego

modelu na dwóch gałęziach matematyki – teorii mnogości oraz rachunku predykatów pierwszego rzędu [1]. W swojej publikacji zaproponował nowy system do przechowywania i pozyskiwania danych dla dużych centrów informacyjnych. Zamiast rekordów o zmiennej długości i dużej swobodzie, uporządkowanych i połączonych według pewnego wzoru (**CODASYL**) zaproponował on przechowywanie danych w tabeli o rekordach stałej długości. Wydajność takiego systemu mógłby być niska w przypadku sortowania dużych tabel z pustymi komórkami. Zaproponowany model relacyjny pokonywał ten problemem poprzez rozbitcie jednej tabeli na wiele oraz przeniesienie opcjonalnych elementów z głównej tabeli do tabel z nią powiązanych. W tabelach tych tworzone były nowe rekordy wtedy tylko, gdy dane opcjonalne istniały [10].

B_ID	Biuro	Miasto	Ulica	
1	Delpol	Warszawa	Radarowa	
2	Elpol	Poznań	Radiowa	

U_ID	Imię	Nazwisko	B_ID	U_ID
1	Jan	Kowalski	1	1
2	Jan	Nowak	1	2
3	Jan	Wiśniewski	2	4
4	Janina	Kowalska	1	4

Rysunek 7: Model relacyjny

Model zapisu danych zaprezentowanych przez Codd początkowo został przyjęty przez naukowców jako intelektualne kuriozum. Aby jego propozycja mogła doczekać się realizacji Codd musiał stoczyć dwie bitwy. Pierwszą wewnątrz **IBM** o realizację projektu, drugą ze społecznością naukową promującą tradycyjne systemy zarządzania bazami danych [7].

W **IBM** problemem był istniejący system zarządzania bazą danych – **IMS**. Firma zainwestowała olbrzymie nakłady finansowe i organizacyjne w infrastrukturę oraz ekspertów koniecznych do sprzedaży produktu. Wprowadzenie nowej technologii spowodowałoby, więc iż poczynione nakłady nie zostałyby zrekompensowane. Przesłanki te powodowały, że zainteresowanie **IBM** w rozbudowę projektu Codd'a było minimalne.

Dodatkowo jak wspomniano wcześniej propozycja Codd'a odnośnie systemu relacyjnego odbierana była przez naukowców z dziedziny baz danych jako nierealna [7]. Chcąc popularyzować swój projekt Codd zorganizował publiczną debatę pomiędzy nim a Charlesem Bachmanem, w owym czasie kluczową postacią w **CODASYL**. Debatę tę spotkała się z krytyką ze strony **IBM**

ponieważ podminowywała jej strategiczny produkt – IMS. Jednak odniosła wymierny skutek. Na początku lat siedemdziesiątych ukazały się dwa projekty mające na celu implementację założeń Codda. Pierwszy z nich o nazwie System R [15], rozwijany był przez IBM drugi o nazwie Ingres na Uniwersytecie Berkeley w Kalifornii [16].

Prace nad budową Systemu R trwały około 9 lat. Na przełomie 1974/75 zbudowano prototyp systemu celem dokonania prezentacji jego możliwości. Na pełnowartościowy produkt trzeba było jednak czekać do roku 1979. Najważniejszym osiągnięciem jednak zespołu pracującego nad Systemem R było opracowanie języka zapytań bazy danych o nazwie **SQL** (*ang. Structured Query Language*) [17]. Język ten stał się amerykańskim a później międzynarodowym standardem pozyskiwania oraz manipulacji danymi w relacyjnych bazach danych.

Pod koniec roku 1973 dwóch naukowców z Uniwersytetu Berkeley, Michael Stonebraker oraz Eugene Wong zainteresowało się możliwościami systemów relacyjnych. Pozyskawszy pieniądze na rozwój, rozpoczęli pracę nad budową systemu opartego o model relacyjny. System ten uzyskał nazwę Ingres (*ang. Interactive Graphics and Retrieval System*). Ingres rozwijany był podobnie do Systemu R jednakże budowany z przeznaczeniem na inne platformy sprzętowe i systemowe [7]. Projekt zapoczątkowany na Uniwersytecie Berkeley zaczął być rozwijany przez małe grupy pasjonatów zarówno na uczelniach jak i poza nimi. Ingres również posługiwał się ustrukturyzowanym językiem zapytań bazy danych o nazwie **QUEL** nieznacznie różniący się od **IBM SQL** [16].

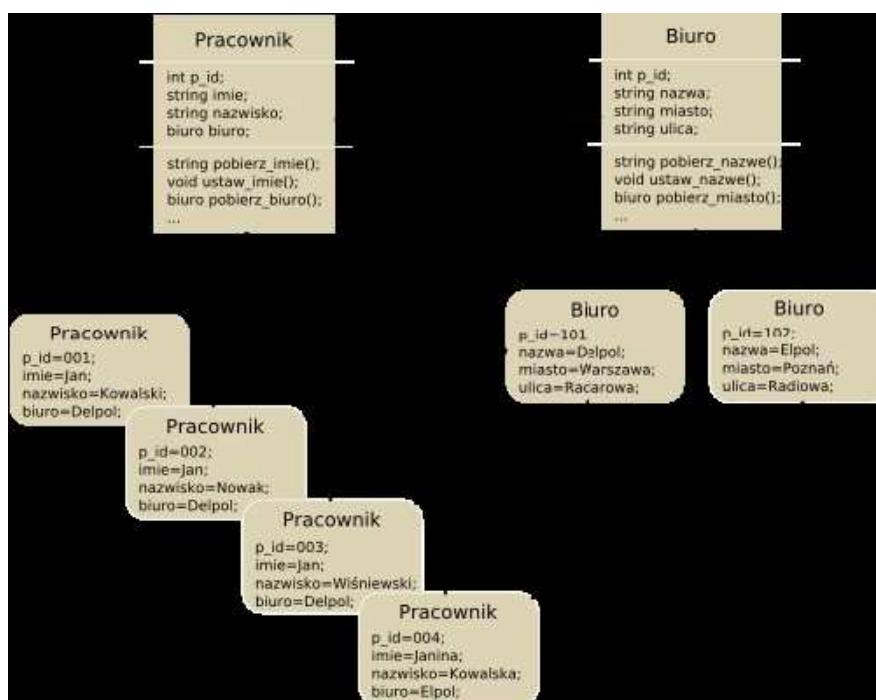
Fakt rozwijania Ingres przez szeroką społeczność informatyczną przyczynił się do sukcesu modelu relacyjnego. Liczne publikacje oraz otwartość kodu spowodowały iż zastosowane rozwiązania szybko zaczęły przyjmować się w biznesie [7]. Dzisiaj systemy relacyjne oparte na przechowywaniu danych w tabelach są najpowszechniej wykorzystywanym sposobem przechowywania danych.

3.5. Obiektowe bazy danych

Model relacyjny mimo odniesionego sukcesu posiadał kilka ograniczeń. Jednym z nich była obsługa bardzo małej ilości typów danych. Najczęściej były to tekst, data, liczby i kilka innych. W połowie lat osiemdziesiątych powstała cała gama aplikacji typu **GIS**, **CAD** (*ang. Computer Aided Design*), **CAM** (*ang. Computer Aided Manufacture*), które wymagały przechowywania danych, o typach niezbyt nadających się dla relacyjnych baz danych [18,19]. Dodatkowo coraz

większą popularność zdobywały obiektowe języki programowania, które sposobem organizacji aplikacji i reprezentacji danych różniły się istotnie od programowania strukturalnego stosowanego przy relacyjnych bazach danych. Zawsze zachodziła możliwość przekształcenia na obiekty danych przechowywanych w relacyjnych bazach, jednak wymagało to wielu pracochłonnych przeróbek aplikacji i odchodziło od istoty aplikacji obiektowych. Braki te dały początek rozwojowi obiektowych systemów zarządzania bazami danych. Obiektowe bazy danych opierały się na paradygmacie obiektowym zapoczątkowanym przez Norwega Kristena Nygarra. W tradycyjnych systemach dane oraz procedury przechowywane były oddzielnie (dane i relacje w bazie natomiast procedury w programie użytkowym) [19]. W modelu obiektowym następuje połączenie procedur wykonywanych na relacjach – lub bardziej ogólnie na encjach [18] – z jej danymi.

W wyniku tego połączenia encje stają się niezależnymi jednostkami, które stosunkowo łatwo mogą być wielokrotnie wykorzystywane i przenoszone. Zachowanie encji nie jest już powiązane z określonym programem użytkowym, lecz stanowi część samej encji, tak więc bez względu na to, gdzie jest użyta, zachowuje się w przewidywalny, znany sposób.



Rysunek 8: Struktura obiektowej bazy danych

Obiektowe bazy danych nie były ewolucją baz relacyjnych lecz raczej alternatywą dla nich. Obiektowe bazy danych uważane są jako powrót do systemów hierarchicznych i sieciowych. Rozwój obiektowych baz danych wynikał z popularyzacji języków programowania takich jak Smalltalk, C++ czy Java, które wymagały obiektowego podejścia do przechowywania oraz pozyskiwania informacji z bazy.

3.6. Obiektowo-relacyjne bazy danych

Powstanie obiektowych baz danych miał bezpośredni wpływ na relacyjne bazy danych. Zauważono, iż obiektowe bazy danych wprawdzie doskonale nadają się do kompleksowej obsługi wszystkich typów danych, jednak posiadają duże ograniczenia w obsłudze języków zapytań [9]. Implementacja języków zapytań w obiektowych bazach danych wydała się na tyle trudna, iż prostsza byłaby rozbudowa relacyjnych baz danych o obsługę dodatkowych typów danych.

Badania przeprowadzone między innymi przez Michaela Stonebrakera przyczyniły się do powstania nowej grupy systemów zarządzania bazami danych. Nowe oprogramowanie zostało nazwane obiektoworelacyjne i porównaniu do tradycyjnych systemów relacyjnych zostało rozbudowane o dodatkowe możliwości, w postaci [20]:

- obsługi rozszerzonych typów danych definiowanych przez użytkowników,
- obsługi obiektów przez ustrukturyzowany język zapytań **SQL**,
- możliwość dziedziczenia wbudowana w język **SQL**,
- wprowadzenie tzw. wyzwalaczy czyli procedur uruchamianych automatycznie w przypadku zaistnienia zdefiniowanego przypadku.

Mimo wielu niedoskonałości obiektoworelacyjne DBMS stały się obecnie najczęściej wykorzystywanym oprogramowaniem do obsługi baz danych. Szacuje się, iż około 95% systemów gromadzenia i przetwarzania danych bazuje na tym modelu danych.

4. Kierunki rozwoju

Technologia baz danych znajduje się w ciągłym rozwoju. Obecnie dla naukowców największym problemem jest przechowywanie bardzo dużych ilości danych. Próby rozwiązania tych problemów ukierunkowane są na:

- opracowanie nowego modelu danych – strumieniowe bazy danych [21,22],
- modernizację sposobu zapisu danych na nośnikach pamięci [23],
- rozproszenie danych w sieci komputerowej [4, 9, 10, 24]

Prace nad rozproszonymi systemami baz danych zapoczątkowane zostały już około roku 1980 [24]. Jednak dopiero rozwój technik telekomunikacyjnych i związany z nim wzrost przepustowości łączy umożliwił wykorzystanie tych systemów do celów komercyjnych. Z rozproszonymi **DBMS** wiąże się szereg bardzo skomplikowanych problemów i obecnie są one jednym z głównych projektów rozwijanych przez naukowców z dziedziny baz danych.

Oprócz **DBMS** istnieje szereg innych technologii związanych z przetwarzaniem danych. Korzystają one pośrednio z Systemów zarządzania bazami danych, w skład ich wchodzi:

- Hurtownie danych [1]. Zgodnie z definicją hurtownia danych to zorientowana podmiotowo, zintegrowana, zróżnicowana czasowo i trwała kolekcja danych przeznaczona do wspomagania procesu podejmowania decyzji. Hurtownie informacji zostały zaproponowane organizacjom przemysłowym jako technologia pozwalająca użyć swoich archiwów danych jako środka do uzyskania przewagi nad konkurencją.

- Narzędzia **OLAP** (*ang. online analytical processing*) [25]. Jest to technologia, która używa wielowymiarowych perspektyw zagregowanych danych w celu zapewnienia szybkiego dostępu do strategicznych informacji przeznaczonych do zaawansowanych analiz. Technologia **OLAP** pozwala użytkownikom na uzyskanie głębszego zrozumienia oraz dodatkowej wiedzy o różnorodnych aspektach swoich danych korporacyjnych poprzez szybki, konsekwentny i interakcyjny dostęp do wielu odmian możliwych sposobów widzenia danych.

- Eksploracja danych (*ang. Data mining*) [26]. Przed eksploracją danych postawione jest zadanie odkrywania wartościowej wiedzy z bardzo dużych baz danych, hurtowni danych i innych informacyjnych repozytoriów.

Eksploracja danych jest dziedziną ukierunkowaną na zastosowania. Problemy badawcze, które się w niej podejmuje, są umotywowane specyfiką dostępnych zbiorów danych dotyczących świata rzeczywistego.

- Mobilne bazy danych [1]. Są to przenośne bazy danych, fizycznie odseparowane od serwera scentralizowanej bazy danych, lecz posiadające możliwość komunikacji z odległymi lokalizacjami z serwerem, dzięki czemu pozwalają na współdzielenie danych firmowych. Rozwój tych systemów jest następstwem rozwoju komunikacji bezprzewodowej.

Oprogramowanie to w większości przypadków znajduje się w fazie rozwoju jednak powstała już znaczna liczba produktów komercyjnych implementujących te założenia.

5. Podsumowanie

Informacja stanowi kluczową rolę w działaniu każdej organizacji. Dlatego też firmy starają się gromadzić jak najwięcej danych i traktując je jako swój kapitał, odpowiednio chronić. Posiadanie odpowiednich informacji oraz właściwy sposób ich wykorzystania pozwala firmom uzyskać przewagę biznesową nad konkurencją.

Chociaż Państwowa Straż Pożarna jest organizacją „non profit” i nie posiada konkurencji, to presja społeczna wymusza na niej skuteczność oraz ekonomiczność w działaniu.

Zakres oraz różnorodność danych przetwarzanych przez **PSP** jest bardzo duży, począwszy od informacji o materiałach i związkach niebezpiecznych a skończywszy na konstrukcji elementów danego budynku i liczbie jego mieszkańców. Aby podnieść swoją skuteczność oraz ekonomiczność, **PSP** stanęła przed trudnymi problemami technicznymi. Obecne systemy gromadzenia i przetwarzania danych [27] nie są w stanie sprostać lawinowo przyrastającym ilościom informacji oraz wymaganiom co do sposobu ich przetwarzania i udostępniania. Dlatego też, konieczna jest reorganizacja systemów informatycznych w **PSP**. Modernizacja powinna być przeprowadzona w obszarach:

- budowy odpowiednich struktur przechowywania danych, dopasowanych do rodzaju informacji – systemy relacyjne, obiektowe,
- metod pozyskiwania i aktualizacji informacji – budowa kanałów przepływu informacji z innych instytucji np. urzędów geodezyjnych, służb miejskich a także pozyskiwanie danych z przeprowadzonych akcji i ćwiczeń,
- wymiany informacji pomiędzy komórkami **PSP** oddalonymi od siebie geograficznie – budowa rozproszonych baz danych,
- zastosowania odpowiednich narzędzi do analizy i eksploracji danych,
- dystrybucji danych oraz budowy mobilnych baz danych – dostęp do danych z samochodów pożarniczych lub innych mobilnych jednostek.

Dla większości z przedstawionych problemów rozwiązanie znaleźć można w najnowszych osiągnięciach techniki z dziedziny baz danych. Dlatego też kluczem do podniesienia skuteczności

działania **PSP** a także obniżenia kosztów jej działalności jest wiedza z zakresu systemów baz danych.

Literatura

1. Connolly T. Begg C.: Systemy Baz Danych – projektowanie, wdrażanie i zarządzanie w praktyce. Tom 1 i 2. Warszawa 2003
2. Ullman J. D., Windom J.: Podstawowy wykład z systemów baz danych. WNT Warszawa (2000)
3. Date C. J.: Wprowadzenie do systemów baz danych. WNT Warszawa 2000
4. Jones D. W.: *Punched Cards A brief illustrated technical history.*
<http://www.cs.uiowa.edu/~jones/cards/history.html>
5. Philips N.V.: *Everything about punch cards.* (1967)
<http://www.tno.nl/instit/fel/museum/computer/en/punchcards.html>
6. Gulliver D.: *Sillicon Valley and Beyond.* Berkeley Area Government Press Berkeley CA (1981)
7. National Research Council: *Funding a Revolution: Government Support for Computing Research.* NATIONAL ACADEMY PRESS Washington D. C. (1999)
8. Silberschatz A., Galvin P. B.: *Podstawy systemów operacyjnych.* WNT Warszawa (2000)
9. Jackson M.: *Thirty years (and more) of databases.* Information and Software Technology Nr 41 (1999)
10. Burleson D. K.: *Managing Distributed Databases: Building Bridges between Database Islands.* John Wiley & Sons New York (1994)
11. Date C. J.: *Wprowadzenie do systemów baz danych.* WNT Warszawa (1981)
12. Courtney J. F., Paradise D. B.: *Database Systems for Management.* Times Mirror/Mosby College Publishing St. Louis (1988)
13. Codd E. F.: *A relational model of data for large shared data banks.* Communications of the ACM Nr 13 (1970)
14. GarciaMolina H., Ullman J. D.: *Implementacja systemów baz danych.* WNT Warszawa (2000)
15. Chamberlin D. D. i inni: *A History and Evaluation of System R.* CACM 24, 10 (October 1981) pages 632646.
16. Stonebraker M.: *Getting started in INGRES – a tutorial.* Memorandum Nr ERLM518 23 kwiecień 1975. db.cs.berkeley.edu/papers/

17. McJones P.: The 1995 SQL Reunion: People, Projects, and Politics. 1995
http://www.mcjones.org/System_R/SQL_Reunion_95/sqlr95.html
18. Lausen G., Vossen G.: Obiektowe bazy danych. WNT Warszawa (2000)
19. Harrington J. L.: Obiektowe bazy danych. Micom Warszawa 2001.
20. Cattell R. G. G.: Object Data Management – ObjectOriented and Exetended Relational Database Systems. AddisonWesley Reading MA (1994)
21. <http://wwwdb.stanford.edu/sdt/>
22. <http://www.cs.cornell.edu/database/cougar/index.htm>
23. <http://www.sybase.com/tpcwhitepaper>
24. Bell D. Grimson J.: Distributed Database Systems. AddisonWesley Reading MA 1994
25. OLAP Council White Paper <http://www.olapcouncil.org/research/whtpapco.html>
26. Kryszkiewicz M.: Eksploracja danych w telekomunikacji. Warszawa 2004 Konferencja Hurtownie Danych i Bussiness Intelligence
27. Praca zbiorowa: Efektywne metody zarządzania ryzykiem związane z ochroną ludności wg właściwości terenowych organów administracji państwowej i samorządowej. SGSP Warszawa 2001