MECHANIKA
CZASOPISMO TECHNICZNE
TECHNICAL TRANSACTIONS
MECHANICS
WYDAWNICTWO POLITECHNIKI KRAKOWSKIEJ
4-M/2011
ZESZYT 7
ROK 108
ISSUE 7
YEAR 108

DARIUSZ KARPISZ*

NIE TYLKO RELACYJNY MODEL DANYCH

RELATIONAL DATA MODEL IS NOT ONLY ONE

Streszczenie

W artykule przedstawiono przegląd *modeli danych* stosowanych obecnie do implementacji baz danych. Zagadnienie to jest fundamentalne dla zrozumienia określonego systemu bazodanowego przygotowanego według wybranego modelu. Mimo iż najbardziej rozpowszechnionym modelem jest *model relacyjny*, to jednak systemy opierające się na modelu *hierarchicznym* i *sieciowym* wciąż są używane. Odpowiedzią na ograniczenia poszczególnych modeli są modele mieszane *relacyjno-obiektowy* oraz *sieciowo-relacyjny*.

Słowa kluczowe: *model danych, model relacyjny, model hierarchiczny, model sieciowy*

Abstract

The paper presents an overview of data models currently used to implement the databases. This issue is fundamental to understanding a particular database system prepared according to the chosen model. Although the most common model is the relational, the systems based on hierarchical and network models are still being used yet. The answer to the limitations of individual models are models of mixed object-relational and network-relational.

Keywords: *data model, relational data model, hierarchical data model, network data model*

* Dr inż. Dariusz Karpisz, Instytut Informatyki Stosowanej, Wydział Mechaniczny, Politechnika Krakowska.

BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ

1. Wstęp

1.1. Model danych

Podstawowym i najważniejszym pojęciem charakteryzującym wszystkie bazy danych jest *model danych* [1, 2].

Model danych to zbiór ogólnych zasad posługiwania się danymi obejmujący następujące zagadnienia:

1. *Definicje danych*, czyli zbiór reguł określających strukturę danych w bazie danych (tabele, indeksy, typy danych, powiązanie danych między sobą).
2. *Operowanie danymi*, czyli zbiór reguł określających operacje na danych (sposób wstawiania, modyfikowania i kasowania danych).
3. *Integralność danych*, czyli zbiór reguł określających, które stany bazy danych są poprawne, a operacje na danych dozwolone (np. co należy wykonać, jeśli udało się zapisać jedynie część danych z większego pakietu).

W klasycznym podejściu do baz danych wyróżnia się cztery modele podstawowe [2]:

1. Hierarchiczny,
2. Sieciowy,
3. Relacyjny,
4. Obiektowy,

oraz modele mieszane:

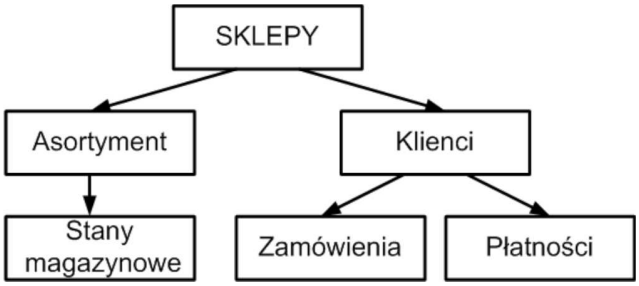
5. Relacyjno-obiektowy,
6. Sieciowo-relacyjny.

Za mniej popularne można uznać modele, sieciowy i hierarchiczny, które są jeszcze stosowane w systemach typu *mainframe*, często w aplikacjach finansowych. Obecnie najczęściej używa się modelu relacyjnego i jego relacyjno-obiektowych kombinacji.

1.2. Model hierarchiczny

Model hierarchiczny wywodzi się z systemu *MIS* (Management Information System) oraz IDS (Integrated Data Store), nad którymi prace rozpoczęto 1960 roku.

Jednym z bardziej znanych i używanych do dziś DBMS (DataBase Management System) opartym na modelu hierarchicznym, wraz z jego późniejszymi modyfikacjami, jest IBM IMS (Information Management System) [3]. W 1966 roku IBM rozpoczął prace nad IMS w ramach programu kosmicznego Apollo.



Rys. 1. Przykład drzewa danych w modelu hierarchicznym
Fig. 1. Example of data tree in the hierarchical model

BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ

W modelu hierarchicznym (Hierarchical Data Model) [3, 4] baza danych (w zakresie modelu danych) jest zorganizowana w postaci *odwróconego drzewa* tak, że istnieje jeden *korzeń* i *gałęzie*, przy czym tabela nadrzędna to „ojciec”, a podrzędna to „syn”. Nietrudno więc zrozumieć, że ojciec może mieć wiele dzieci, ale każde dziecko tylko jednego ojca, co pokazuje rysunek 1. W odniesieniu do struktur bazy danych, każdy rekord prócz głównego (wierzchołek drzewa) powiązany jest dokładnie z jednym rekordem nadrzędnym.

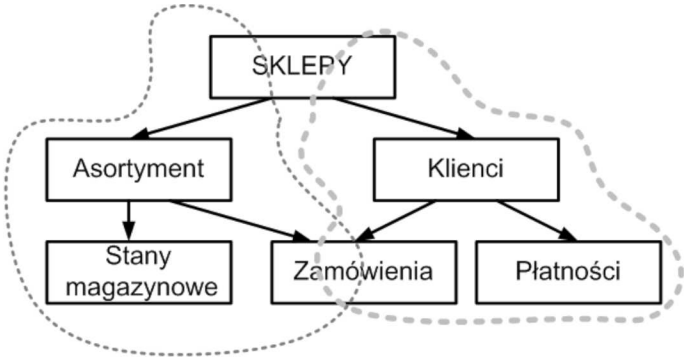
1.3. Model sieciowy

Model sieciowy został pierwotnie opracowany przez Charlesa Bachmana i ustandaryzowany w 1969 roku przez komitet standaryzacyjny CODASYL (Conference on Data Systems Languages). Nie przyjął się jednak w szerokich zastosowaniach, ale jego powstanie przyczyniło się do prac nad kolejnym modelem danych: relacyjnym.

W 1971 roku komitet DBTG (Data Base Task Group), będący częścią CODASYL, przedstawił raport na temat Data Manipulation Language (DML) i Data Definition Language (DDL), które stały się standardem w obsłudze relacyjnych baz danych.

Jednym z przedstawicieli modelu sieciowego był VAX DBMS nieistniejącej już firmy DEC – Digital Equipment Corporation.

W *modelu sieciowym* (Network Data Model) [3, 4] baza danych jest zorganizowana w formie odwróconego drzewa, ale tak, że gałęzie mogą być współdzielone przez kilka drzew – taka struktura tworzy graf, czyli sieć powiązań zrealizowanych za pomocą wskaźników. Różnicę w stosunku do modelu hierarchicznego obrazuje rysunek 2. Model sieciowy miał w ten sposób zniwelować ograniczenia modelu hierarchicznego.



Rys. 2. Przykład drzewa danych w modelu sieciowym
Fig. 2. Example of data tree in the network model

1.4. Model relacyjny

Model relacyjny został zaproponowany w 1970 roku przez Edgara Franka Codd’a [5] i z czasem okazał się najbardziej trafnym podejściem do problemu organizacji danych. Jest to tzw. pierwszy relacyjny model danych.

W relacyjnym *modelu danych* (Relational Data Model – RDM) baza danych składa się z relacji (tabel) powiązanych ze sobą za pomocą więzów integralności: *kluczy głównych* (*PK*) i *kluczy obcych* (*FK*).

BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ

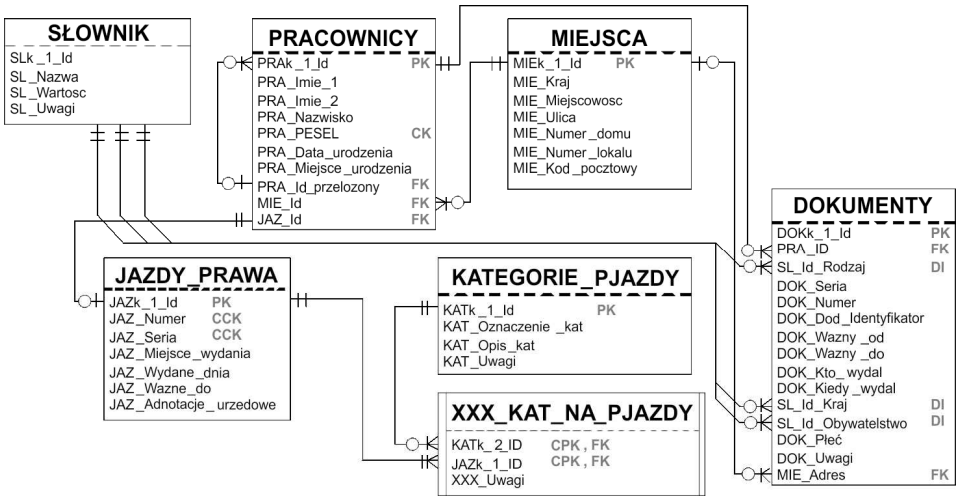
Relacja (tabela) – jest podstawową struktura relacyjnej bazy danych. Wszystkie tabele składają się z atrybutów (kolumn) oraz wierszy (krotek). Kolejność wierszy jest obojętna!

Atrybut (kolumna) – jest najmniejszą strukturą w modelu relacyjnym, przeznaczoną do przechowywania jednostkowych danych, reprezentujących pewną cechę tematu, jaki opisuje tabela, np. atrybut *IMIĘ* w relacji *OSOBA*.

Wiersz (krotka) – składa się z pełnego zestawu atrybutów, niezależnie od tego, czy mają one przypisaną wartość. Wiersz jest najmniejszą strukturą opisującą cały obiekt, którego dotyczy tabela, np. wszystkie informacje o danej osobie w tabeli *OSOBA*.

Klucz główny (*PK*) – ang. *Primary Key*, tworzy jeden lub więcej atrybutów tabeli (klucz złożony), na podstawie wartości którego można jednoznacznie zidentyfikować każdy wiersz w tej tabeli.

Klucz obcy (*FK*) – ang. *Foreign Key*, tworzy jeden lub więcej atrybutów tabeli (klucz złożony), które w innej tabeli są kluczem głównym. W ten sposób tworzone jest bezpośrednie powiązanie wierszy z danymi z dwóch różnych tabel.



Rys. 3. Przykład projektu bazy danych w modelu relacyjnym
Fig. 3. Example of project of database in a relational data model

1.5. Model obiektowy

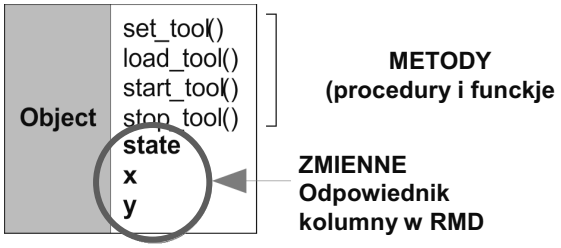
Jednym z nowych podejść do rozwiązań w dziedzinie organizacji baz danych jest obiektowy model danych. W modelu tym dane nie są gromadzone w plikach czy tabelach, ale w obiektach.

Obiekty to kapsuły zawierające komplet danych (analogia krotki w modelu relacyjnym) oraz dodatkowo metody wykonywania operacji na tych danych – co pokazuje rys. 4.

Obiektowe bazy danych bazują na obiektowym sposobie projektowania i programowania aplikacji **OO** (Object Oriented). Operowanie na dużej ilości obiektów jest czaso- i pamięciochłonne, dlatego w chwili obecnej obiektowe bazy danych są zazwyczaj wykorzystywane w czystej postaci jedynie w małej skali.

Obiektowe bazy danych oznaczane są skrótem **ODB** od ang. Object DataBase lub **OODB** od ang. Object Oriented DataBase.

BPK BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ



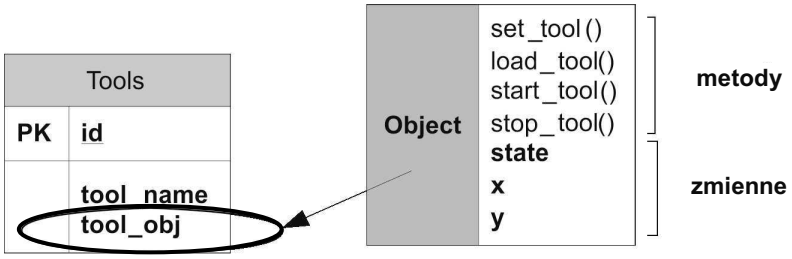
Rys. 4. Przykład budowy obiektowych struktur danych w OODB
Fig. 4. Example of object data structure in a OO databases

2. Mieszane modele danych

2.1. Model relacyjno-obiektowy

Model relacyjny [5] przyjął się ze względu na jasne kryteria i wydajność przetwarzania danych. Obiektowe systemy zarządzania bazami danych są trudniejsze do użytkowania i implementacji, a w wielu przypadkach wciąż zbyt mało wydajne. Dlatego też, aby wykorzystać zdobycze nowej technologii, stosuje się *model relacyjno-obiektowy*.

Model relacyjno-obiektowy charakteryzuję się tym, że w relacyjnych tabelach można utworzyć kolumny do przechowywania danych typu obiektowego [6]. Na rysunku 5 przedstawiono tabelę *Tools* zawierającą kolumnę obiektową, służącą do zdefiniowania zmiennych oraz metod operowania na instancji (egzemplarzu) obiektu. W modelu relacyjnym można by zdefiniować jedynie nazwy metod do sterowania narzędziem.



Rys. 5. Obiektowy atrybut relacyjnej tabeli
Fig. 5. Example of object column in a relational table

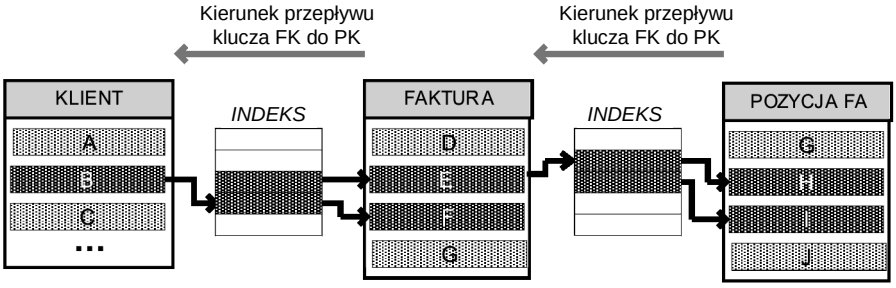
2.2. Model sieciowo-relacyjny

Model *relacyjny-obiektowy* nie jest jedynym modelem mieszanym. Warto wiedzieć, że istnieją systemy wykorzystujące model sieciowy lub hierarchiczny z modelem relacyjnym.

Do wyjaśnienia *modelu sieciowo-relacyjnego* [7] najlepiej posłużyć się przykładem. Rysunek 4 prezentuje prosty schemat bazy do obsługi faktur. Dla każdego klienta możliwe jest zarejestrowane dowolnej ilości faktur, przy czym każda faktura może posiadać dowolną liczbę pozycji faktury. W celu zapewnienia więzów między tabelami tabele połączone są parami kluczy *FK-PK*. W tabeli *FAKTURA* znajduje się *FK* będący kluczem głównym w tabeli *KLIENT*, natomiast w tabeli *POZYCJA FA* znajduje się *FK* będący

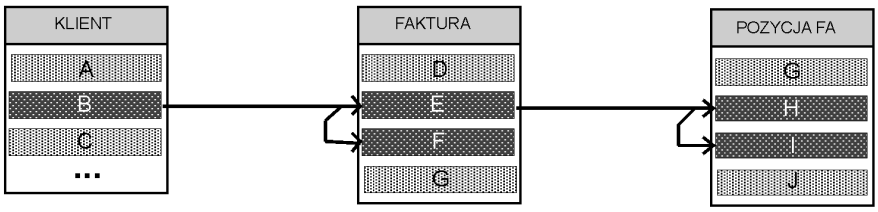
BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ

kluczem głównym w tabeli *FAKTURA*. W celu odnalezienia faktury danego klienta należy najpierw przeszukać tabelę *FAKTURY* po wartości klucza głównego klienta, a następnie dla odnalezionej faktury przeszukać tabelę *POZYCJA FA* po kluczu głównym danej faktury. W takim wypadku najlepiej posłużyć się indeksami stworzonymi na potrzeby takiego właśnie wyszukiwania.

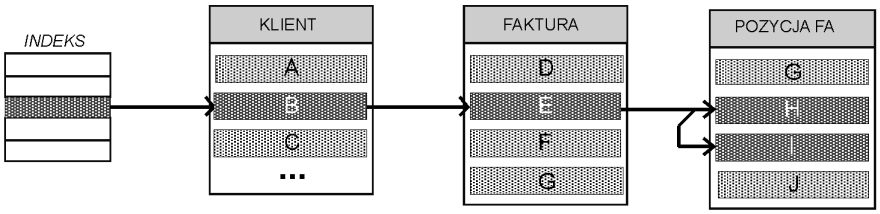


Rys. 6. Przykład realizacji powiązań danych dla modelu relacyjnego
Fig. 6. Example of data connections in the relational data model

Model sieciowy przedstawiony na rysunku 7 dla wcześniej opisanej bazy danych (rys. 6) wydaje się znacznie prostszy. Różnica między RDM, a *modelem sieciowym* polega na tym, że *KLIENT* udostępnia wskaźnik do faktury, natomiast w tabeli *FAKTURA*, kolejne faktury należące do danego klienta wskazują na ten wiersz tabeli, do którego dany klient się odnosi (za pomocą struktur wskaźnikowych). Zatem między tabelami *KLIENT* i *FAKTURA* występuje *relacja 1:1*, a *relacja 1:n* realizowana jest przez wskaźniki tabeli *FAKTURA*. Na podobnej zasadzie zrealizowane jest powiązanie między tabelami *FAKTURA* i *POZYCJA FA*. Model sieciowy charakteryzuje się znacznie szybszym czasem dostępu do tych samych danych niż w RDM.



Rys. 7. Przykład realizacji powiązań danych dla modelu sieciowego
Fig. 7. Example of data connections in the network data model



Rys. 8. Przykład realizacji powiązań danych dla modelu sieciowo-relacyjnego
Fig. 8. Example of data connections in the network-relational data model

BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ

Model mieszany, ***sieciowo-relacyjny*** wykorzystuje zarówno znane z *RDM* struktury indeksowe, jak i *wskaźniki* charakterystyczne dla *modelu sieciowego*. W omawianym przykładzie (rys. 8) relacyjny indeks wskazuje na bezpośrednią ścieżkę *„konkretny KJLIENT” – „szukana FAKTURA”*, przez co wyszukiwanie danych w tabelach *KLIENT* i *FAKTURA* odbywa się przez przeszukanie indeksu. Dozwolone są również odwołania wskaźnikowe co wykorzystano w tabeli *POZYCJA FA*.

Model *sieciowo-relacyjny* został zaimplementowany w RDM Server rozwijanym od 1982 roku przez Raima Corporation oraz RDM Embedded firmy Centura Software.

3. Wnioski

W artykule przedstawiono podstawowe zagadnienia związane z modelami danych stosowanymi w systemach bazodanowych. Mimo iż obecnie najbardziej rozpowszechnionym wydaje się *model relacyjny* i jego obiektowe rozszerzenia [6], to jak pokazano – nie jest to jedyny wybór. Zdaniem autora, droga do modelu relacyjnego była znacznie bardziej interesująca [8] niż obecne niewielkie zmiany w specyfikacjach dla modelu relacyjnego. W zaawansowanych systemach biznesowych klasy *Business-to-Business* (B2B) wciąż działają systemy klasy IBM IMS [3] i nic nie wskazuje na to, aby miały być zastąpione relacyjnymi bazami danych, które jako uzupełnienie funkcjonalności IMS potrzebowałyby dodatkowych monitorów przetwarzania transakcji biznesowych (np. Oracle Tuxedo). Ciekawą alternatywą dla tradycyjnych, relacyjnych systemów DBMS jest *model sieciowo-relacyjny*. Obecnie wykorzystują go systemy wymagające z jednej strony dużej szybkości przetwarzania żądań i niezawodności, a z drugiej, oszczędzania zasobów. Należą do nich systemy *wbudowane* (*embeded*), a zwłaszcza urządzenia i systemy na potrzeby militarne. Jak pokazano w pierwszej części artykułu, mimo że podstawowe i znane modele są proste do zrozumienia oraz istnieją ich przemysłowe implementacje, to jednak nie jesteśmy skazani jedynie na model relacyjno-obiektowy. Niska znajomość kombinacji sieciowo-relacyjnych lub hierarchiczno-relacyjnych skłoniła autora do przedstawienia niniejszego artykułu.

Literatura

[1] Fry J.P., Sibley E., *Evolution of Data-Base Management Systems*, ACM Computing Surveys, Vol. 8, No. 1, 1976.

[2] Navathe S.B., *Evolution of Data Modeling for Databases*, Communications of the ACM, Vol. 35, No. 9, 1992.

[3] Long R., Harrington M., Hain R., Nicholls G., *IMS Primer*, IBM ITSO, 2000.

[4] Elmasri R., Navathe S.B., *Fundamentals of Database Systems (3th Edition)*, Addison Wesley 2000.

[5] Codd E.F., *A Relational Model of Data for Large Shared Data Banks*, Communications of the ACM, Vol. 13, No. 6, 1970.

[6] Bergholt L., Steen Due J.S., Hohn T., Knudsen J.L., Nielsen K.H., Olesen T.S., Pedersen E.H., *Database Management Systems: Relational,*

BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ

Object-Relational, and Object-Oriented Data Models, Report COT/4-02-V1.1, Centre of Object Technology 1998.
[7] J o h n s o n P., *Network and Relational Data Modeling*, Raima Inc. 2008.
[8] N o r t h K., *Database Systems: The First Generation*, KN Computing, LLC, 2007.

BIBLIOTEKA CYFROWA POLITECHNIKI KRAKOWSKIEJ