

パーソナル・マルチメディアデータベースの試作

中埜 善夫 檜垣 伸俊
松下電器産業株式会社 中央研究所

オフィス業務用途を目的とした、文章・画像・図形の複合したマルチメディアデータを蓄積できるデータベースを試作したので、その概要について述べる。

本データベースでは、不定長データは、1レコードでしかも1属性しかない特殊な表として管理される。データモデルは階層表構造であり、表と不定長データが統一的に階層管理されている。このモデルはネットワーク型、リレーショナル型の融合形態になっており、ISOで標準化の方向にあるSQLおよびNDLとのスキーマの比較をし、その位置付けを示す。また、リレーショナル機能の特徴や、文章主導型のマルチメディアデータ表示について紹介する。

An experimental Multi-media Database for personal computer systems

Yoshio NAKANO Nobutoshi HIGAKI
Central Research Laboratory, Matsushita Electric Industrial Co., LTD.
3-15 Yagumo-Nakamachi, Moriguchi, Osaka, 570, Japan

We introduce an experimental personal database software which can accumulate multi-media data composed of text, image and graphic.

It manages an un-fixed length data, that is, text, image or graphic as a special table which has only one record and one attribute. A Hierarchic-Relational data model is used in it, and un-fixed data and tables are managed with the same manner. This model is union of network one and relational one, so we make its characteristic clear by comparing with SQL and NDL schemas. We also introduce a feature of relational operation and a text initiated presentation as a multi-media data processing.

1. はじめに

最近、ビットマップ・ディスプレイ、マウス、イメージスキャナなどの機器が充実し、ハードディスクも小型大容量化している。ワープロも普及し、文章・画像・図形を統合して扱えるシステムが身近になってきた。これら、データの表現形態・処理形態の異なるものを複合して扱うことを、マルチメディア処理と呼んでおり、これからのOAとマルチメディアは切り離せない状況にある。文章、グラフ、画像の混在した形で、1つの意味的にまとまったものとして、オフィス文書が電子化される。これらがディスク内に大量に蓄積できるようになると、文書を整理した形で蓄積でき、かつ必要ときに素早く検索できるシステムが要求されてくるであろう。

今回、文字や数値のほかに文章、グラフ、画像などの不定長大容量のデータが蓄積できるパソコン・データベース（以下、MMDB：Multi-Media DataBaseと記す）を試作したので、これについて述べる。MMDBは、ユーザの操作用語であるコマンド・インタープリタ、フルスクリーンでの表データ修正機能、レポート・ライタ、既存データベースとのデータ交換機能などを備えている。

2. 文書ファイル用データベースに要求される機能

オフィス内の共用文書と個人文書を比較すると、

(1) 共用文書

- ・管理性重視
- ・分類体系は決定的（再分類は稀）
- ・文書への分類の適用は機械的

(2) 個人文書

- ・実用性重視
- ・分類体系は暫定的（再分類は頻繁）
- ・文書への分類の適用は多様

の違いがあろう。したがって、個人用途を考えるならば文書の移動（再分類）が柔軟にできねばならない。また、個人用途であっても、パソコンは共用できることが望ましい。つまり、

文書利用の面からは、

- ・個人が個人の媒体にデータベースを格納できる
- ・個人に属さない共用のデータベースがもてる
- ・データベース相互間の情報交換ができる

ことである。

これらのことより、文書ファイル用データベースには、次の機能が要求されるであろう。

- (a) マルチメディアデータからなる文書データが蓄積できる

- (b) 文書の移動が柔軟に行なえる
- (c) 個人文書と共用文書とを分離して管理できる
- (d) 個人文書と共用文書とで情報交換できる

3. MMDBのデータ管理構造

3. 1 文書情報構造のモデル化

オフィス内での文書の管理形態を考えると、およそ、

- (1) キャビネット
- (2) バインダ
- (3) 文書
- (4) ページ

という階層的なものになっている。ページ内にはさらにその構成要素として、

- (5) データ

がある。(1)～(4)は管理情報であり、(5)は文章、画像、図形などのマルチメディアデータである。

このように、日常行なっているオフィスでの文書データ管理（文書ファイル）をそのまま対応づけて階層的管理体系をもたせると、利用者の理解が容易である。これに関しては階層表モデル^[1]を提案した。階層表モデルも論理的なモデルであるから、以下のような実現方法が考えられる。

(a) 階層型ファイルシステムによる実現

- ・表ごとに1つのファイルが必要
 - ・同時にオープンできるファイル数に制約
 - ・検索機能がない
- これは全般的に機能が低く、効率はよくないと思われる。

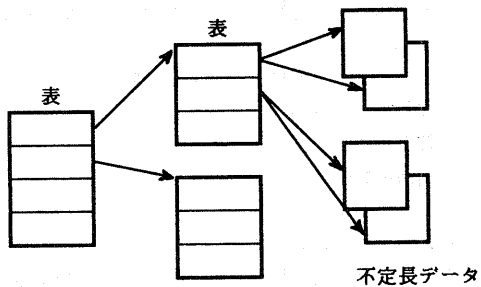
(b) リレーショナル型データベースによる実現

- ・レコード番号の概念がない
 - ・表間の関係が明示的でない
- この中で問題となるのが、レコード番号の概念がないことである。つまり、何らかのレコード情報を操作する前には必ず、検索なり、JOIN などの操作を行わなければならない。これは、処理の高速化からはオーバーヘッドとなる。

(c) 階層表モデルデータベースによる実現

- ・レコード番号の概念がある
 - ・表間の関係を階層で表わせる
- 内部モデルと論理モデルが一致しており効率がよい。

そこで、第1図に示す「階層表モデル」を内部構造に導入した。MMDBではレコードと表のリンク機能を付加し、階層表構造が容易に実現できるようになっている。



第1図 階層表構造

3.2 不定長データの管理

従来のパソコン・データベースはリレーショナル型が一般的で、数値・文字列を対象にしている。表はレコード管理の簡便さとランダムアクセスの高速性から、属性幅を固定長にする方が有利である。しかし、画像・文章など不定長かつ大容量のマルチメディアデータも固定長の1属性として扱くと、レコードアクセス速度やディスクの有効利用の面で得策ではない。属性幅の管理を可変長で行なえば解決できるが、全体の処理速度の低下を招くことになる。そこで、MMDBでは不定長データを分離して蓄積し、レコードとのリンクをとることにした。これにより、表も不定長データも統一的に階層管理することができるようになった。

4. MMDBのデータ蓄積構造

4.1 MMDBのデータ管理

MMDBで作られるデータベースは、第2図に示すように、データベース全体の管理情報のあるルートデータファイル、固定長のデータを格納する表データファイル、不定長データを格納する実体データファイル、B-treeのはいつている索引データファイル、処理時の作業用の作業データファイルという複数の論理データファイルにより構成される。この論理データファイルは第2図(a)のように物理データファイルと1対1に対応させることも、第2図(b)のように複数のものを1つにまとめて、物理データファイル数を減らすこともできる。

(1) データファイルの管理

すべてのデータファイルは、一定の長さのブロックから構成されている。その内の1つのブロックをルートブロックに割当て、使用ブロック数、使用最大ブロック番号などの管理情報を格納する。

(2) ルートデータファイル

ルートデータファイルは、データベースの名前やパスワードや他のデータファイルのルートブロックの番号などのデータベース固有の情報を持っている。

(3) 表データファイル

固定長の表データは、レコード長、レコード数などの表の管理情報を持つ「表」管理表、表を構成する形式情報を管理する「表形式」管理表などによって管理される。詳細については4.2で述べる。

(4) 実体データファイル

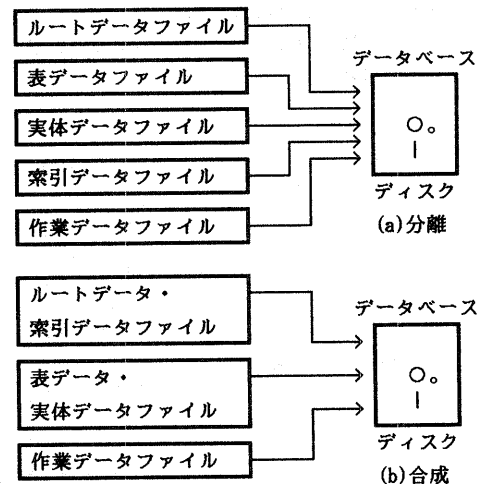
不定長のデータは、レコード長が不定で1レコードしかない特殊な表として表現される。表に対して表形式が定義されるのと同様に、不定長データも文章や画像などのデータ種別が定義され、大きさなどの情報とともに実体データ管理表によって管理される。

(5) 索引データファイル

索引データファイルは、キー指定した属性のB-treeによって構成されている。MMDBでは、属性の値と位置情報(表番号、レコード番号など)の両方を比較しており、各属性値の大小関係は一意的に決まる。

(6) 階層構造の構成

階層構造は、表データファイルに上方接続管理表と下方接続管理表を作って管理する。表やレコードと接続関係が独立しているので多対多の接続が可能になる。また、表形式間の接続関係の禁止・許可をすることができる。



第2図 データベースのファイル構成

4.2 システム表

データベースには、5種類のシステム表が予約されており、これらの表によって各種の表情報が管理される。

(1) 「表」管理表

データベースに作成されているすべての表の、

- ・表形式
- ・レコード数
- ・レコード長

- ・データ格納領域のエントリーブロック番号などが登録されている。
- (2) 「表形式」管理表
- データベースに定義されているすべての表形式の
- ・属性数
 - ・属性名
 - ・属性型
 - ・属性値のオカレンス
 - ・属性幅
 - ・レコード長
 - ・索引格納領域のエントリーブロック番号
 - ・接続規則管理表のエントリーレコード番号
- などが登録されている。

- (3) 「接続規則」管理表
- 表形式管理表で定義された接続規則の、
- ・接続許可される表形式
- が必要数だけ登録されている。

- (4) 「リレーション」管理表
- データベースに作成されているすべてのリレーションの、
- ・リレーション形式
 - ・レコード数
 - ・レコード長
 - ・リレーションを構成する表
 - ・データ格納領域のエントリーブロック番号
- などが登録されている。

- (5) 「リレーション形式」管理表
- データベースに定義されているすべてのリレーション形式の、
- ・リレーション形式を構成する表形式
 - ・属性数
 - ・結合条件
 - ・結果の属性名
 - ・レコード長
- などが登録されている。

これらのシステム表もすべて表管理表および表形式管理表によって管理されている。

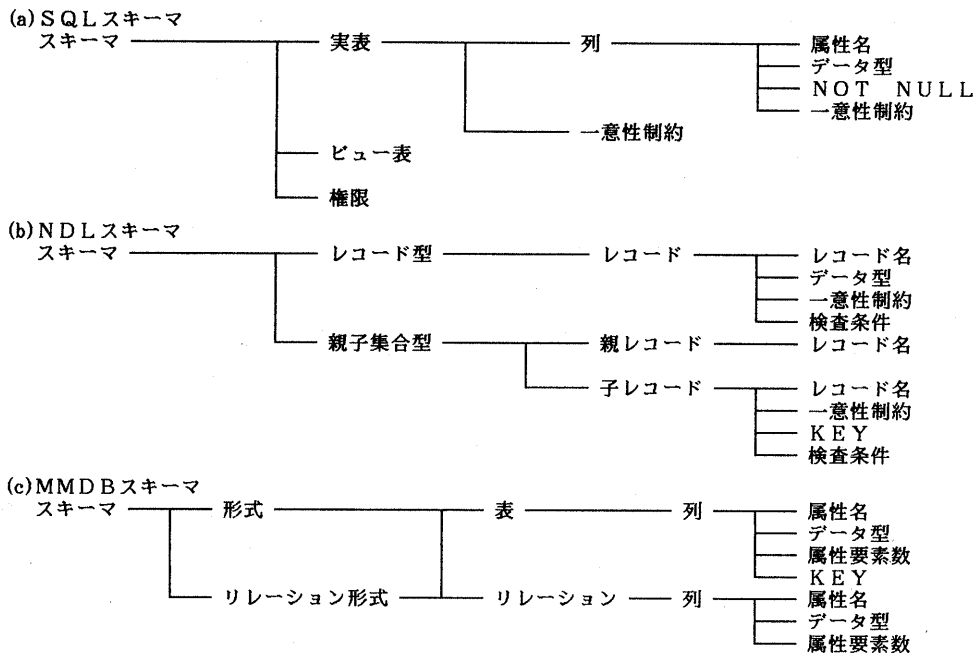
6. SQL、NDLとのスキーマ比較

6.1 スキーマの概要

(1) SQL

SQL^{[3][4]}のスキーマは<スキーマ> (CREATE SCHEMA) により定義される。スキーマは実表、ビュー表および権限からなる。スキーマの中の実表は<表定義> (CREATE TABLE) により定義され表名により識別される。表は列と一意性制約からなる。標準SQLにおいて、一意性制約を<表定義>又は<列定義>中で行なうことができる。

表の列は<列定義>より定義される。<列定義>では<列名>と<データ型>を定義する。<列定義>では、NOT NULLと一意性制約も指定できる。



第3図 スキーマの概略

ビュー表は、<ビュー定義> (CREATE VIEW) により定義される。ビュー表は、<問い合わせ指定> が実行されたとき生じる仮想的な表である。ビュー表が実際にデータベース内の表として実体化されるかどうかは各実装による。データベース内の表に対するアクセス権は<権限定義> (GRANT) により設定される。アクセス権の制御は認可識別子によりおこなわれる。

第3図(a)にSQLスキーマの概略を示す。

(2)NDL

NDL^{[5][6]}のスキーマは、<スキーマ> (SCHEMA) により定義される。スキーマはスキーマ名により識別される。スキーマは、レコード型および親子集合型からなる。レコード型は、<レコード型>で定義する。レコード型は、一意性制約、データ型および検査条件からなり、レコード型はレコード名により識別される。検査条件とは、レコードがデータベース内に存在するために必要な条件である。親子集合型は、<親子集合型>で定義する。親子集合型は、レコード間の関係を表現するもので、<親レコード句>と<子レコード句>及び親子集合中の子レコードの順序を指定する<順序句>からなり、親子集合名で識別する。

レコードはレコード型により管理され、データ型、検査条件、一意性制約を含む。

親子集合は親子集合型により管理され、1つの親レコードと1つ以上の子レコードからなる。

なお、NDLの中では、機密保護に関する機能は規定されていない。

第3図(b)にNDLスキーマの概略を示す。

(3)MMDB

MMDBのスキーマは、表形式、リレーション形式からなる。表形式は<表形式定義> (DEFINE STRUCT) で定義でき、表形式名で識別できる。リレーション形式は<リレーション形式定義> (DEFINE RELFORM) で定義でき、リレーション形式名で識別できる。表形式およびリレーション形式により、表およびリレーションはグループ化される。リレーションは2つ以上の表を結合演算したときに生じる仮想的な表である。リレーションが実際にデータベース内の表として実体化されるかどうかは、ユーザの指定による。

表およびリレーションは、列からなる。<列定義>で一意性制約、NOT NULLの指定はない。

第3図(c)にMMDBスキーマの概略を示す。

6.2 SQL、NDLと比較したMMDB

(1)表およびリレーションを、その表形式およびリレーション形式でグループ化して管理できる。これは、NDLのレコード型、親子集合型に類似しており、SQLに

表1 インタープリタ・コマンド

データベース管理	create db delete db open close commit rollback
スキーマ定義	define struct define relform connect rule create table
表操作	append table copy table delete table disp table
不定長データ操作	append data copy data delete data disp data
検索操作	select table
階層操作	connect table discon up down
結合演算	join table
コマンドファイル制御	alias assign set if while switch goto

は該当する概念がない。

(2)1つの属性中に複数の要素を持つことができる。これは、NDLは可能であるが、SQLではできない。

(3)リレーションにみられるように、実体化されている表から新しい仮想的な表を作り出すことができる。これは、SQLのビュー表と類似しておりNDLにはないものである。

7. MMDBの特徴

7.1 インタープリタ・コマンド

インタープリタ・コマンドは約50あり、次のようにグループ化できる。

- (1)データベース管理
- (2)スキーマ定義
- (3)表操作
- (4)不定長データ操作
- (5)検索操作
- (6)階層操作
- (7)結合演算
- (8)コマンドファイル制御

そのうちの主要なものを表1に示す。

インタープリタ・コマンドの特徴を以下に挙げる。

(a)複数のデータベースがオープンできる

MMDBでは、データが階層的に連鎖しているの、外部ファイルを介したデータベース間の情報交換方式は向いていない。MMDBでは複数のデータベースを同時にオープンして相互アクセスできるようになっている。

(b)データベースへの更新操作の確定/取消ができる

MMDBは大容量のデータを蓄積するので、バックアップをとっておくのが大変である。不慮の誤操作からデータを保護するために、次の指示が選択できる。

- ・ COMMIT データベース名 …… 更新の確定
- ・ ROLLBACK データベース名 …… 更新の取消

(c)非更新モードがある

- ・ OPEN データベース名 …… 更新
- ・ OPEN データベース名 RETRIEVE …… 非更新

上述の ROLLBACK 機能の応用で、非更新モードでデータベースをアクセスできる。

(d)カレントレコード・ポインタ

MMDBでは「カレントレコード・ポインタ」が1つの重要な概念である。このポインタ (CRP) は、仮想的に利用者の注目しているレコードを指している。CRPの指しているレコードがカレントレコード (CREC) であり、CRECの存在する表がカレント表 (CTBL) である。CRPが表内や表の階層を移動していく。CRECは、ポインタ接続により

- ・ 文章データファイル
- ・ 画像データファイル
- ・ 図形データファイル
- ・ 下位表

をもつことができる。

CRPを対象としたコマンドの例をいくつか挙げると、次のものがある。

- ・ DOWN …… CRECの下位表にCRPを移動する
- ・ UP …… CTBLの上位レコードにCRPを移動する
- ・ APPEND TO 表名 …… CRECを指定表に複写する
- ・ TRANS TO 表名 …… CRECを指定表に移動する

・ DISP DATA …… CRECのマルチメディアデータを表示する

7.2 リレーショナル機能

リレーション形式は、リレーションを構成する属性情報 (属性の所属する表形式および属性名) と、表の結合条件情報 (表を結合する結合演算子、演算する2つの属性の所属する表形式および属性名) から定義される。これらの情報を持つリレーション形式は、結合演算の手続きとしてデータベースに登録できる。このリレーション形式に表を適用することによって、結合演算が行なわれ、リレーションが生成される。

結合演算は2つの属性値の条件比較によって行なわれるので、演算前に属性値をソートしておくのが通常の手法である。MMDBでは一方の属性に関して索引を生成し、他方は逐次処理をする。このとき、処理時間の短縮のために、レコード数の少ない方の属性に関して索引を生成し、レコード数の多い方の属性を逐次処理する。また、キー指定されて既に索引が存在するときは、その索引を利用する。

結合演算子には表2の9種のもので使用でき、結果はレコード番号の組で生成される。

表2 結合演算子

演算子	判定条件
=	一致
<>	不一致
>	大きい
>=	以上
<	小さい
<=	以下
prefix	文字列前方一致
contain	文字列中間一致
suffix	文字列後方一致

7.3 情報検索

表はすべて形式管理され、すべての表はいずれかの表形式 (グループ) に属する。同じ「表形式」に属する表は索引に関して論理的に1つの表と見なされ、表形式ごとに索引が作成される。したがって「表形式」で検索することにより、「いずれのキャビネット、ポインタに属するかは不明、不問であるが、あるキーワードを持つ文書を捜したい」という検索が容易に実現できる。

また、レコードには、ユーザから隠されたリンクのためのポインタが対応付けられている。接続されている表または不定長データはこのポインタにより知ることがで

き、このポインタは双方向管理されている。したがって、キーワード検索の結果得られたレコードがいずれのキャビネット、バインダに属するか情報は、特に文書表の中に設定しておかなくても階層を逆にたどることで容易に知ることができる。

8. マルチメディア・データベースのプレゼンテーションへの応用

画像や図形を扱ったものには、地図やレントゲン写真のデータベースがあり、ワープロにおいても一つの要件になってきている。これらに共通しているのは、表現が静的 (static) なことである。つまり、表示情報は2次元空間内に固定されており、必要に応じてハードコピーがとれる。これに対し、表現が動的 (dynamic) なもの、つまり、電子情報化されていることの利点を生かし、2次元空間内に固定されないソフトコピーを対話機器上に実現するようなものが要求されていくと予想される。

MMDBではそのアプローチとして、第4図に示すように表示画面を3分割し、それぞれの領域に文章、画像、図形データが表示されるもの考えた。文章領域は、順次スクロールしていくが、画像領域、図形領域はスクロールしない。

データの主体は文章 (説明文) であり、画像、図形は文章では表現できないものや理解の補助となる従体と位置付ける。文章データに画面展開の制御権をもたせるために、文章内に制御情報を埋め込む。制御情報の種類としては、

- .P n n番目の画像を画像領域に表示する
- .G n n番目の図形を図形領域に表示する
- .W n n×0.1秒動作停止する
- .K キー入力待

などがある。これら制御情報を埋め込む箇所は、文章による説明の進行上必要となった所である。例えば、

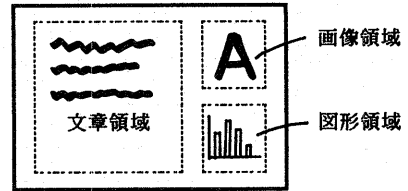
- 1 次に、図4に示したマルチメディアデータが、どのように構成され、画面表示
- 2 されていくかについて説明する。図4では、画像、図形をそれぞれ1つの
- 3 .P 5
- 4 場合を示しているが、実際には図5のように複数存在する。

のように構成する。6行目の文章を表示するときには図5の表示が必要となる。したがって、5行目に画像出力指示である

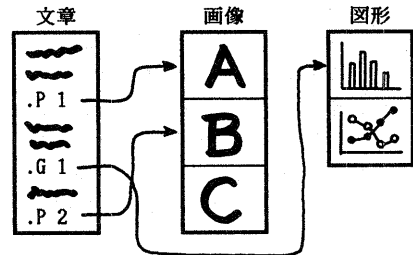
.P 5

という制御情報を埋め込む。表示プログラムは1~4行

目の文章を表示したのち説明用の画像を表示し、そのうち6行目からの文章の続きを表示する。制御情報を読み込む度に、順次画像データおよび図形データが表示されていく。(第5図)



第4図 表示画面構成



第5図 制御構造

9. 評価

以下に示す表データ1と表データ2の2つの表を用いて、MMDBとR:BASE、dBASEの処理速度比較を行なった。結果を表3に示す。

《表データ1》500件 1496byte/件

(例)

記事番号 : E86140001
 資料番号 : Elektrotech.....
 巻 : 103
 号 : 3
 発行国 : AUS
 発行年 : 86
 ページ : 103-106
 著者名 : (1) VOGLER H
 機関名 : (1*) Gewerkschaft.....
 原文標題 : Missbrauch macht.....
 和文標題 : 乱用は不安の源, 協議は信用を創り出す
 キーワード : 技術 技術導入 技術者
 抄録文 : 新技術の普及の際, 受け入れ問題が.....
 《表データ2》30件 18byte/件
 記事番号のみ

表3 市販データベースとの処理速度比較

		MMDB	R:BASE5	dBASEIII	
検索	キー指定なし	1分55秒	1分38秒	1分57秒	
	キー指定あり	1.5秒	3.5秒	1.6秒	
ソート		1分54秒	1分41秒	*	
結合演算	キー指定なし	処理1	2分04秒	2分08秒	2分14秒
		処理2	2分04秒	49分	59分
	キー指定あり	処理1	2.6秒	2分08秒	*
		処理2	2.6秒	24秒	*

(注)結合演算子は '=' で結果は30件

処理1と処理2は表の指定順序が逆なこと以外は同一の内容

MMDB 処理1:join 表^テ-^タ1 表^テ-^タ2...
 処理2:join 表^テ-^タ2 表^テ-^タ1...
 R:BASE5000 処理1:intersect 表^テ-^タ1 with 表^テ-^タ2..
 処理2:intersect 表^テ-^タ2 with 表^テ-^タ1..
 dBASEIII 処理1:use 表^テ-^タ1 ; join 表^テ-^タ2...
 処理2:use 表^テ-^タ2 ; join 表^テ-^タ1...

MMDBの結合演算の速度は、4つの場合とも他のデータベースに比べて速く、キー指定ありのときは特に顕著である。これは、7.2で述べた索引を有効利用する効果が現われたものである。

また、R:BASEとdBASEは、結合演算での表の指定の順序によって結合演算の速度が大きく異なるのに対してMMDBは表の指定の順序に左右されない。これも、MMDBでは2つの表のレコード数を考慮して内部処理しているからである。

検索、ソート速度は他のデータベースとほぼ同等の性能を有している。

10. おわりに

ここでは、オフィス文書用のデータベースモデルとして「階層表モデル」を使用することにより、2次元の表データに加えて文章・画像などの不定長大容量データも蓄積できる、マルチメディア・データベースについて述べた。これによりスキャナから入力した画像データやワープロ文書を大量に蓄積できる文書ファイルシステムが、容易に実現できる。

また、マルチメディアの一例として、文章主導型の表示について述べた。今後さらに、動画や音声をも統合し、さらに柔軟なマルチメディア構造を研究していく予定である。

表4 MMDB規格諸元

項目	最大
同時に扱えるDB	16
表数	65,536/DB
レコード数	20億/表
レコード長	4096バイト
属性数	128/レコード
マルチメディア	16/レコード
マルチメディア長	20億バイト

参考文献

- [1] 中埜 他；「マルチメディアデータベース用データモデルの提案」,第31回情報処理学会全国大会,4B-10
- [2] 中埜 他；「パーソナル・マルチメディアデータベースの試作」,第35回情報処理学会全国大会,6Bb-4
- [3] ISO/DIS 9075 "Information processing systems - Database language SQL"
- [4] 芝野；「データベース言語SQL」,情報処理学会研究会,データベース・システム59-6
- [5] ISO/DIS 8907 "Information processing systems - Database language NDL"
- [6] 横山；「データベース言語NDL」,情報処理学会研究会,データベース・システム59-5