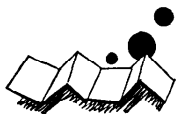


解説

データベース技術の現状と展望[†]植村 俊亮^{††}

1. はじめに

情報という現代人間の営為に不可欠な資源をなるべく共有し、有効利用をはかろうというのがデータベースの概念である。データベース技術は、今日の情報システム構築に欠くことのできない基礎技術となっている。1960年代初頭から20年にわたるデータベース技術の発展を、3段階にわけて考えることができる。

(1) 揺らん期。1963年から1970年ごろまでの時期。データベースの概念が生まれ、いくつかの先駆的なシステムが開発された。

(2) 成育期。1970年から1981年までの時期。データベース利用が盛んになるとともに、関係モデルを基礎とする理論的研究が充実し、従来の経験的なシステム作りに新しい光を投げかけた。関係パラダイムの時代であった。

(3) 成熟期。1981年以降。10年にわたる基礎研究、開発実験ののちに、商用の関係データベースシステムが登場し始めた。理論研究も新しい時期を迎えようとしている。成熟期というより、新転回期というべきかもしれない。

本稿では、まず揺らん期と成育期を俯瞰し、ついで主要な技術動向を個別に検討する。最後に今後の展望を述べる。なお本稿でいうデータベース技術は、正確にはデータベースシステム技術とすべきものである。簡単のために、しばしばシステムを省略した。

2. データベース揺らん期

データベース (database) という用語の語源は明らかではない。base は軍事用語では基地を意味するから、database とは「データの補給基地」であるという語源説が広く流布している。確証はない。1963年ごろすでに、「computer-centered data base」ということ

ばを冠した研究集會が開かれていたという。いわゆるデータベースシステムの嚆矢と考えられる C. W. Bachman の I-D-S (Integrated-Data-Store) が稼動し始めたのもこの年である¹⁾。Bachman 自身は、1957年に W. C. McGee が稼動させた IBM 702 Report Program Generator を「近代データベースの草分け」としている。計算機によるファイル処理では、計算機を使わない場合にくらべて、ファイルの性質そのものが変化し、より統一され統合された包括的なものになるべきであると、McGee は主張した²⁾。McGee の源泉ファイル (source file) は、包括性、共同利用指向のみならず、呼出し制御などデータ制御の概念まで含んでおり、現代のデータベースシステムに通じる構想を数多く先取りしていた。

1960年代中ごろから、磁気ディスクが本格的に普及し始め、データ処理に新しい地平が開かれた。COBOL 言語仕様の開発母体である CODASYL は、COBOL を拡張して I-D-S 流のデータベース機能を実現しようと仕様の確立に努力した。このように、既存のプログラム言語を母体として構築するシステムを親言語 (host language) 型とよぶ。その一方で、TDMS³⁾に代表される情報検索指向のシステムが流行した。この分野の先駆者の一人である T. W. Oile の論文⁴⁾は、揺らん期のデータベースシステムの特色を端的に表現している。

「データベース管理とデータ管理とは意味が異なる。後者は IBM 式の操作システムで、データ内容に関係なくファイルやレコードの転送を管理する機能をいう。前者は、プログラムの専門家でない人が、データの格納方式や転送方式を関知しなくても、必要な情報をデータベースから取り出せることを目的としている。TDMS が一つの例である。(中略) 手続き的なものと、非手続き的なものとの間は断絶しているのではなくて、両者は連続線上にある。計算機に『いかに』なすべきかを指示する度合と、『なにを』なすべきかを指示する度合とによって、どの程度非手続き的であ

[†] Database Systems, Where We Stand Now and Where to Go by Syunsuke UEMURA (Programming Research Section, Computer Science Division, Electrotechnical Laboratory).

^{††} 電子技術総合研究所ソフトウェア部プログラム研究室

るかがきまる。データベースから情報を検索する分野では、非手続的な方向、『いかに』よりも『なにを』の方向が絶対に重要である。』⁹⁾ Olle の UL/I は、記憶構造をいっさい意識しない末端利用者言語の設計を目指していた。このように、従来のプログラム言語にとらわれずに、新しく独自の非手続的システムをめざす方向を、**独立言語 (self-contained) 型**とよぶ。

揺らん期の終りに、**MIS (Management Information System)** ということばが流行したが、理論的、技術的裏付けにとほしく、定着するに至らなかった。しいていえば、当時の MIS とは、TDMS のような情報検索指向、独立言語型のデータベースシステムをオンライン利用することであった。

データベース揺らん期の主要事項を以下にまとめる。

(1) COBOL は、I-D-S 式に磁気ディスク上でつなぎ構造を実現し、これを複数の COBOL プログラムが共有する方式のデータベースシステムを目指した。その最初の成果が、DBTG 69 と通称される言語仕様案であった。

(2) いわゆる階層構造による McGee らの IMS が IBM 社から 1969 年に登場した。MARK IV, TOTAL など、今日も使われている商用パッケージがこの時期に市場に出現している。

(3) オンラインシステムの登場とともに、ファイルを主として情報検索目的に共有する TDMS や SYSTEM 2000 が開発された。ADABAS の開発もこの時期に開始された。

(4) 理論的研究は、古く 1962 年に情報代数 (Information Algebra)⁹⁾、1968 年に D. Childs の STDS⁹⁾ などをみるのみで、際立った展開がなかった。データベースシステムを議論する共通の基盤を欠いていた。

3. 成 育 期

3.1 概 観

1970 年代に入ると、計算機ハードウェアの性能価格比の向上を背景に、データベース技術はいちじるしい発展をとげた。CODASYL 方式のデータベースシステム用コンパイラが各計算機に出そろい、商用システムの利用が広がる一方で、関係モデルを中心とする基礎研究が進んだ。「データベース」は一種の流行となり、情報検索、データバンクといった関連分野でもいっせいにこのことばに切り替え始めた。情報システムに不可欠な基礎技術としてのデータベースは、この

時期に確立されたといえよう。

3.2 CODASYL 方式と商用パッケージ

CODASYL は、DBTG 69 をたたき台として検討を続けた。1971 年にいわゆる **DBTG 71 言語仕様案** をまとめて、これを COBOL を親言語とするデータベースシステム共通言語体系の基本とすることを決定した。4 章で詳述するように、この言語仕様体系をより良くする改訂作業は今日も続いている。またうよ曲折をたどりつつ標準化作業も行われている。この共通言語体系にもとづくデータベースシステムには、各社が独自の名称を与えており、一見して目立たないが、AIM, ADBS, IDS-II, DMS 1100, IDMS, IMAGE, MDBM, DEIMS などはいずれも基本的に CODASYL 方式である。データベースシステムにはじめて取り組む作成者にとっては、よく練られた言語仕様は頼りがいのある存在であり、まずこれを規範としたことは自然のなりゆきであった。COBOL や FORTRAN のようにコンパイラ言語の水準で使いこなせるデータベースシステムは、利用者にも親近感を与えた。

CODASYL 方式はプログラマ向きであって、末端利用者への対応が遅れていた。このことは、DBTG 71 を審議した当時から、重要な課題と認識されていたが、具体的な作業が遅れた。この方式のコンパイラ作成者は、それぞれに末端利用者向けの簡易な問合せ言語を用意したが、方言にとどまった。ようやく 1980 年に至って、CODASYL は末端利用者機能の言語仕様案¹⁰⁾を公表した。これは CODASYL 方式と、後述の関係データベースとの橋渡しと考えられる仕様である (4. 参照)。

また今日商用のデータベースパッケージとして普及している ADABAS, IDMS, MODEL 204 などがさっそうと市場に姿を現わしてきたのも、1970 年代前半である。1981 年以後の商用パッケージには、関係データベースを標榜するものがおおく、70 年代とは明らかに様変りをみせている。

3.3 データベースの普及

成育期にわが国で、現場へのデータベースシステム導入がどう進んだかを、表-1 に示す。これは三井情報開発が、わが国の 1,2 部上場企業をおもな対象として、1979 年および 1981 年の各 11 月に行った利用実態調査である⁷⁾。1981 年 11 月には、データベースシステムをすでに利用している組織が 54.2% に達し、利用を具体的に検討中および将来利用を検討予定

表-1 商用データベースパッケージの利用状況
(三井情報開発¹⁾による)

	1981年11月	1979年11月
現在利用中	54.2% (293)	37.6% (172)
利用を現在検討中	12.8% (69)	18.8% (86)
利用を将来検討の予定	11.1% (60)	13.5% (62)
利用を考えていない	21.9% (119)	30.1% (138)
合計 (回答数)	(541)	(458)

調査対象 1, 2部上場企業, 教育機関, 政府公共団体 1978組
 織 (1979年調査は上場企業, 生保のみ 1589社)
 回答率 27.4% (1979年調査は 28.8%)

表-2 よく利用されているパッケージ¹⁾

パッケージ名	利用数	提供者
AIM	62	富士通
PDM	53	日立
DL/I	37	IBM
IMS	37	IBM
ADBS	26	日電東芝
DMS1100	23	UNIVAC
IDS	17	日電東芝
ADM	16	日立
ADABAS	13	ソフトウェア・エージ
DMS-II	11	パロース
INFORM	11	UNIVAC

の組織を加えると、全体の約 80% に達する。導入率は 2年間に 37.6% から 54.2% へ 16.6% の増加を示している。

同調査で、利用しない組織にその理由を質問した例では、「使用中の計算機容量に制約がある」という理由が第 1 位であるが、その比率は 1979 年の 36.3% から、1981 年には 19.2% に減少して、計算機の性能向上を実感させる。

1981 年の調査に回答した 541 組織は、全部で 795 台の計算機を保有していた。そのうち 293 の組織が 364 のデータベースパッケージを使用していたが、計算機製造業者によるものが全体の 95% をしめていた。利用数の上位 11 種を表-2 に示す。このうちでソフトウェア会社提供のものは ADABAS のみである。AIM, ADBS, DMS 1100, IDS など は前述のように CODASYL 方式、PDM は TOTAL 型、DL/I は IMS のデータベース部分、ADM は IMS 型である。

しかし将来利用を検討の予定の 60 組織が候補として挙げている総計 130 のパッケージの内訳をみると、第 1 位はいぜん AIM (23) であるが、第 2 位に IBM 社の関係データベースシステム SQL/DS (14) が顔をみせており、近い将来の地図の塗り変えを予測させる。IBM 社以外の計算機製造業者が、いっせい

表-3 データベース研究の伸長

	1972	1974	1976	1978	1980
ACM 会員数 [†]	29,000 ^{†††}	28,315	32,736	38,214	48,157
ACM/SIGMOD 会員数 ^{††}	650	1,140	2,500	(不詳)	4,180
TODS 定期購読者数	—	—	2,331	5,120	6,907
情報処理学会会員数 [†]	5,735	7,196	9,155	10,669	14,263
データベースシステム研究会登録者数 ^{††}	—	163	197	311	374
全国大会データベース発表件数/総件数	2/166	36/358	43/398	67/510	74/672

[†] 個人会員の総数。

^{††} 名称変更などの詳細は略す。

^{†††} ACM 25 周年記念誌の公称値、実数発表は 1973 年 (28,675) から。なお全体に集計時期のずれなどがあり、厳密な比較には注意を要する。

に關係データベースシステムの提供を予告し始めたのも、この間の変化を物語っている。

3.4 關係パラダイム

なによりも、1970 年代は關係パラダイムあるいはコードパラダイムの時代であった。1970 年に IBM サンノゼ基礎研究所の E. F. Codd が提案したデータの關係モデルを基礎として、数おおくの研究が盛んに行われた (このように、ある期間、研究方法の基礎として広く受け入れられる業績を、パラダイムという)。データモデルのみならず、呼出し制御や同時実行制御などを含むデータベースシステム技術全般にわたって、關係モデルをふまえた研究がおおかった。すなわち關係モデルは、データベースを議論するにあたっての共通の土台として、きわめて有効であった。

アセンブリ言語がコンパイラ言語に進化したとき、プログラマは計算機の物理的構造、たとえばレジスタの種類や目的、ファイル記憶媒体の内部の詳細から解放された。關係モデルは、さらに、ファイル編成法 (呼出し法) から利用者を解放する。したがって、効率のよいシステム作成が課題とされてきたが、1981 年前後から、SQL/DS, INGRES, ORACLE など、關係モデルにもとづくデータベースシステム (これを単に關係データベースシステムとよぶ) が続々と発表され始めた。また 1981 年には、このモデルの提唱者 E. F. Codd が ACM チューリング賞を受賞した。

コードパラダイムを象徴する 1 例として、データベース研究の盛んな伸長を示す資料を表-3 に掲げる。1972 年から 1980 年にかけて、アメリカ計算機学会 (ACM) 会員数は約 1.7 倍に増加、情報処理学会は 2.5 倍の増加を示している。この間に ACM SIGMOD (Special Interest Group on Management of

Data) 会員は、約 6.5 倍に増加した (6.5 割ではない。ただし 1980 年度の増加は前年比 2 名だったという報告がある)。1976 年から刊行の始まったこの分野専門の論文誌 Transaction on Database Systems (TODS) 誌の 1980 年度の定期購読部数は約 6,900 部である。これは創刊以来 4 年で約 3 倍の伸びを示している。単純に計算すると、ACM 会員の約 8% が SIGMOD 会員であり、約 14% が TODS 誌の読者ということになる (もちろん会員でなくとも、TODS 誌を購入できる)。この急伸長は、データベースが夢物語から、足が地に着いた研究開発の段階へ移行してきたという認識の現われであろう。成育期の初期には、Codd 自身が SIGMOD で活躍し、大学研究所におけるデータベース研究の起爆剤となった。

ただし学会の会議に参加し、論文を発表する人の数はそんなにおおくはない。前記の TODS 誌の年間論文掲載数は 20~30 編、SIGMOD による Management of Data 国際会議の発表論文数も 20~30 編 (1982 年度は投稿 123, 採択 33), Very Large Data Bases 国際会議が 30~50 編 (1981 年度は投稿 230, 採択 45) である。IFIP TC 2 の研究会などを加えて、年間約 100 編となる。これは単純に計算すると、SIGMOD 会員の 2.5%, ACM 会員の 0.2% に相当する。

SIGMOD 会員数は 1980 年に前年度比 2 人の伸びにとどまっていらい、停滞していると伝えられる。1982 年には、Very Large Data Bases 国際会議をめぐって不幸な混乱が起き⁸⁾、また一方では ACM Symposium on Principles of Database Systems が誕生するなど、変動をみせた。研究のすそ野のひろがりや新しい応用分野への発展を示しているのであろう。たとえば 1979 年末には SIGOA (Office Automation) が組織されており、SIGMOD もこの影響を受けたはずである。筆者はここにパラダイムの区切りの徴候をみる。

情報処理学会をみるかぎり、わが国のデータベース研究は盛んになりつつあるが、外国ほど極端ではない。全国大会に「データベース」を冠した発表部門ができたのは 1976 年のことで、それ以後この分野および関連分野の研究発表件数は全体件数の 10% 強、総会員数の 0.5% 程度である。学会誌の第 1 回のデータベース特集号も 1976 年 10 月号であった。筆者は学会誌論文数についての統計をも表-3 に含めようとしたが、意味のあるデータを得ることができなかつ

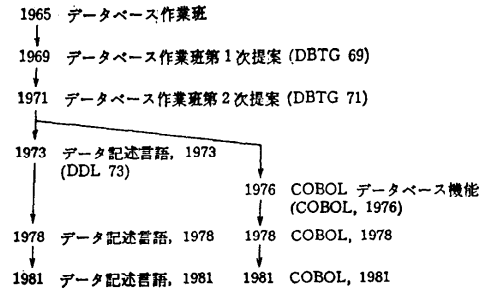


図-1 CODASYL 言語仕様の歴史

た。1970 年代を通して、情報処理学会誌にはデータベースを主題とする論文は年間 2 編ないし 3 編掲載されているのみで、わずかに 1980 年に全論文 57 編のうち 6 編を数えたにすぎない。さらに、データベースシステム研究会の登録会員数も着実に増加しているが、常に総会員数の 3% 未満である。データベース分野の研究発表は ACM、情報処理学会のみならず、他学会でも行われているから、以上をもって早急にわが国の研究水準を論じることはできない。しかし外国と比較すると、低調の感はいない。

4. CODASYL 方式の歩み

CODASYL 方式のデータベースシステムは、揺らぐ期から成育期にわたって着実な進展をみせた。関係モデル支持者や商用データベースパッケージの側からの盛んな攻撃にもかかわらず、今日なおこの方式がデータベースシステムの中心の一つであり続けているのは、次の理由によると考えられる。

(1) 現行のコンパイラ言語水準のデータベース言語仕様として、十分に検討された唯一の規範的存在であった。

(2) 技術的にも、実現可能な水準であった。

(3) COBOL の場合と同様に、言語仕様の改良を積極的に行い、作成者、利用者、学界の経験的、研究的知恵の吸収を怠っていない。

COBOL 言語仕様の開発組織である CODASYL によるこの言語仕様開発の歴史を図-1 に示す。

DBTG 提案あるいは DBTG 仕様によるという文章を現在でも散見するが、DBTG が素案作りの基礎作業を終えて解散してから、すでに十年になる。この間 CODASYL 方式のデータベース言語仕様は改良を重ねて、最新の 1981 年仕様に及んでいる。最新の言語仕様書は次の通りである。

(1) CODASYL DDL 委員会開発報告 1981⁹⁾—

データベースの概念スキーマ記述言語の仕様書。

(2) CODASYL COBOL 委員会開発報告 1981¹⁰⁾—DDL で記述したデータベースを COBOL で操作する場合の利用者インタフェースを定める。

(3) CODASYL FORTRAN データベース機能開発報告 1980¹¹⁾—DDL で記述したデータベースを FORTRAN で操作する場合の利用者インタフェースを定める。

(4) CODASYL 末端利用者機能委員会開発報告 1980¹²⁾—DDL で記述したデータベースを末端利用者が操作する場合の、利用者インタフェースを定める。

DBTG 71 以降の言語仕様改訂の要点を以下に列挙する。

(1) 資源管理や性能向上のための記述項を DDL から削除し、DSDL (データ記憶記述言語) という新しい言語にまとめた。3層スキーマ構成に対応させると概念スキーマは DDL が、内部スキーマは DSDL が、外部スキーマは親言語のデータベース機能が、それぞれ分担することが明確になった。DSDL 言語仕様案は前記の (1) の付録として公開されている。

(2) 同じレコード型が、一つの親子集合型の親でもあり子でもあることを許す。

(3) 領域 (area) や PICTURE 句を DDL から削除した。

(4) 呼出し制御 (access control) 機能を COBOL データベース機能から削除した。同時実行制御、障害回復機能についても、全面的に改訂した。

(5) レコード実現値を、そのレコード中の任意のデータ項目の値によって呼び出すことができる。この部分の言語仕様の変化は、データベースシステムの潮流を暗示していて興味深い。DBTG 71 当時は、レコードを直接呼び出す手段は、システムが与えた識別子によるか、ちらし (hashing) のキー値を指定するかのいずれかしかなかった。1978 年になって、ちらしのキー項目でなくても、陽に KEY と指定したデータ項目でありさえすれば、どの項目の値によってもレコードを直接呼び出せることになった。さらに 1981 年の仕様では、レコード中の任意のデータ項目の値によってレコードを直接に呼び出すことができるようになった。利用者は呼出し経路 (ちらしのキーや索引) を意識しなくてもすむようになり、反面システムの負担が増加した。

(5) 末端利用者機能は長く待たれていたが、1980 年によく言語仕様として登場した。用紙 (forms,

書式、帳票、書類) という概念を中心にすえて、事務処理環境を計算機上に具現しようとする。用紙、メモ、とじ込み (バインダ)、ファイルといった事務用品名がそのまま利用者インタフェースに現われる。用紙中心の方式は、RPG, MARK IV, Query-by-Example, VISICALC に通じる輝かしい伝統をもっている。しかしなんといっても用紙は正規化されていない関係表であるから、この言語仕様は当面 CODASYL 方式の関係モデルインタフェースとして活用されようである。DDL で記述したデータベースを用紙上に写像する言語 EUF DDL がこの開発報告に含まれている。

CODASYL 方式の言語仕様群の標準化については、筆者は当初 1978 年ごろと予想していたが、実際には大幅に遅れ、混乱している。アメリカ規格協会には、DDL 標準化のために X3H2 が設置されて、1980 年には原案をまとめたが、アメリカ国内でも、ISO でも十分な支持をえられなかった。このあおりを受けて、COBOL の第 3 次アメリカ規格作成作業も、データベース機能を含めないままに進行している。混乱の一つの原因は、データ操作から独立して DDL を設定したり、DDL から切り離して COBOL インタフェースだけを考えたりするところにある。無理があることは最初からはっきりしていたのであって、やはりという感がある。混乱を打開するために、X3H2 が一般的なデータ操作機能まで含めて審議し始めており、COBOL インタフェースとの兼合いが新しい混乱を生みつつある。

5. 関係データベースシステム

関係モデルの提唱者 E. F. Codd は、チューリング賞受賞記念講演¹⁵⁾の中で、このモデルの動機を次のように挙げている。

(1) データ独立。データベースの論理的側面と物理的側面とを明確に区別したい (最初このことばは、データとプログラムとを互いに独立にするという意味で使われていた。現在では、Codd のいう意味で使われることがおおい)。

(2) 相互理解。単純で、あらゆる種類の利用者やプログラマが互いに意志を通じ合うことのできるモデルがあるべきである。

(3) 集合演算。複数のレコードを一時に処理する高水準データ操作を設定すべきである。

関係モデルはパラダイムを形成して、おのずから

(2)を達成した。計算機自身の性能向上にも助けられて、(1)(3)を実用水準で実現しようとしている。

関係モデルに刺激されて、数おおくのデータモデルが研究されてきたが、このモデルほどに完結したものはまだ現われていない。同講演中で Codd は、データモデルはすくなくとも次の三つの要素からなるべきであるとしている。

- (1) 構造部分。データ構造型の集まり。
- (2) 操作部分。演算子、推論規則の集まり。
- (3) 一貫性部分。一般性をもつ意味制約則の集まり。

図-2 に、関係モデルに関連するおもな研究分野を示す。Codd が主張するように、こうした三つの要素が互いにかみ合って発展しているのは、関係モデルだけだと思われる。関係データベースの基礎理論については、本学会誌に徹底的な講座²⁰⁾が連載されており、この分野に興味をもつ研究者の必読資料である。

実務からみた場合、たとえば関係代数は情報検索用の演算子だけしか含んでおらず、これだけではファイル更新、整列、報告書作成などを行うことができない。しかしこうした点は外部からうまく補いながら、関係データベースシステムの商品化が進んだ。

IBM 社の関係データベースシステム SQL/DS 開発の歴史をたぐると、10 年におよぶと推定される(図-3。発表論文からの類推による)¹⁶⁾。SQL/DS では、構造部分、操作部分が関係モデルによるのみならず、呼出し制御、同時実行制御などにも関係パラダイムの最新の研究成果が活用されており、壮観である¹⁷⁾。ただし現在のところ一貫性部分を欠いている。

利用者インタフェースの立場からは、SQL/DS はむ

しろ古典的であり、QBE のような新鮮さはない。親言語型、独立言語型という揺らん期のシステム分類は、成育期に入って両者併存という方向がみられたが、SQL/DS はこの点ではきわ立った特徴を示している。末端利用者は SQL という問合せ言語を使う。応用プログラマが親言語中でデータベースを操作する場合にも、同じ SQL 言語を使うのである。Codd はこれを関係統一システムと称している。今後の重要な方向づけと考えられる。

1981 年になって、商用関係データベースシステムが続々と登場する一方で、理論研究にも一つの転機が訪れつつある。「属性の有限個の集合 U を論議の対象とする。 U の部分集合 R を関係スキーマという。」といった極端な定義が行われており、いわば関係という概念の影が薄くなりつつある。

アメリカ規格協会では、将来の関係データベースシステム標準化に備えて、1979 年に ANSI/X3/SPARC Database Systems Study Group (DBS-SG) 中に Relational Database Task Group を設置した。その報告書はすでにまとめられていて¹⁸⁾、標準化の作業を開始すべきである(ただし言語の構文などには立ち入らずに、関係モデルの概念および機能を標準化すべきである)と X3 に提言している。

6. 飛躍のための課題

6.1 基礎研究

関係パラダイムほどではなかったが、成育期のもう一つのパラダイムは概念スキーマであった¹⁹⁾。データベース記述を概念スキーマと内部スキーマ(および外部スキーマ)にわける考え方は、Codd のデータ独立の目的にも合致し、広く受け入れられてきた。しかしデータベースの論理設計の立場からは、概念スキーマをさらに2層にわける考え方が根強く支持されている(図-4)。用語の定説はないので、図-4 にしたがうと、狭義の概念スキーマは実世界の描写、論理スキーマは

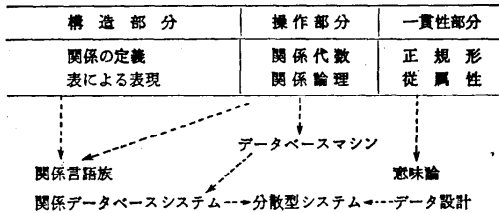


図-2 関係モデルの構造

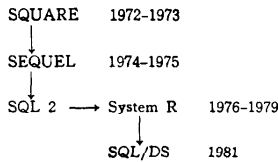


図-3 SQL/DS 開発史

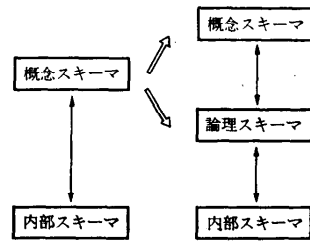


図-4 データベース設計のスキーマ

論理的なファイル構造の決定あるいは、特定のシステムのわく内での論理設計という。そして前者については、実体関連 (E-R) モデル、TH モデルなど、関係モデル以外のモデルを使う²⁵⁾。これはデータモデルの本質にかかわる使い分けであろうか。実体関連モデルは実世界の直観的把握にすぐれ、下向設計に向いている。関係モデルは数学的厳密さに一長があるが、属性から出発するので、上向設計にならざるをえない。しかし両者の特徴を兼ね備えることは不可能ではないはずである。データベースの下向設計に耐えうる直観的な意味把握能力をもち、かつ数学的にも厳密で、Codd のいう意味での三位一体のデータモデルの研究が大きい課題である。

6.2 データ辞書

データベース中のデータに関するデータ、すなわちメタデータを人間にわかりやすい形に整理したものを、データ辞書 (data dictionary) という。操作システム内でのデータディレクトリに対応する。データベースが大規模になり、属性も数がふえてくると、こうした機能は不可欠になる。現行のデータベースシステムでも、データ辞書機能の重要性を認識して、追加提供しているものが多い。また商用パッケージの最近版は、きそうようにデータ辞書中心を主張している。関係データベースシステムでは、データ辞書もまた関係表として、システムに組み込んで提供されている。

データ辞書には、データの名前、属性、データ相互間の関連、データの利用者、データを使用するプログラムなどを記述する。まだデータベース化するにいたらない紙のデータなどについても、データ辞書に記載しておけば、組織体全体を見渡す手段になる。さらに機能を拡大して、データの一貫性 (意味) 制約、分散データの所在位置などを含めていく方向が考えられる。データ辞書は情報システム全体を記述するキーともなりうる重要な要素である。

すでにアメリカ規格協会では、1980年に X3H4 (Information Resource Dictionary System) を設置して、データ辞書標準化に着手している。規格の候補となる抜き出したデータ辞書システムがないので、時間を要するであろう。

6.3 データベースマシン

Codd 自身が述べているように、関係モデルでは、表の要素の番号、行や列の並ぶ順番といった位置的な概念は存在しない¹⁵⁾。利用者はすべての情報を表の値として記録し、関係名、属性名、それに実際の値で

データを呼び出す。これはハードウェアの分野で連想呼出しとか内容呼出し (content addressing) とよぶ技法に対応する。従来からも内容呼出し記憶や情報検索専用装置の研究があったが、集積回路技術の進歩と上述の関係モデルの特徴とを取り入れて、データベースマシンとよばれる新しい研究分野が誕生した²³⁾。データベース応用専門のアーキテクチャをもつ計算機構をデータベースマシンという。トロント大学の RAP、電子技術総合研究所の EDC など、1980 年ごろまでに登場した多彩なデータベースマシンについては、文献²³⁾にゆずる。新しい方式、新しい素材応用などの模索が続く一方で、ここでも続々と商用のデータベースマシンが登場し始めている。これをおよそ二つの型に分類することができる。

(1) 双対型。汎用の親計算機に、これを同一アーキテクチャの計算機をチャンネル結合し、親計算機のデータベース負荷をすべて専用機に移す方式。ADABAS マシンがこの型である (図-5)。同じアーキテクチャの計算機を 2 台使うので、一方に障害が発生しても、もう一方で代行するということができる。ハードウェアとしての新味はすくなく、ファームウェア化による性能向上が期待される。

(2) データベースサーバ型。構内計算機網の中で、旧来ファイルサーバと称されてきた機能を分担するもの。IDM 500 (200) による Shared Database System, IDBP はこの方向である。図-6 に示すとおり、こうしたデータベースマシンは、複数の高性能個人用計算機の共有データベースとして動作する。図-6 (a) は通常のオンラインシステムに似ている。しかしここでは、中心にある計算機 (データベースマシン) はデータベース専用機であり、それ以外の日常業務は端末に代わる個人用計算機が担当している。図-6 (b) では、構内計算機網のイメージがより明確に打ち出されている。なお 6.4 でも再説する。

以上二つの方式のうち、とくに (2) では、ハードウェアの詳細にはっきりしない部分が多いが、公称

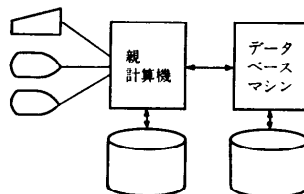


図-5 双対型データベースマシン

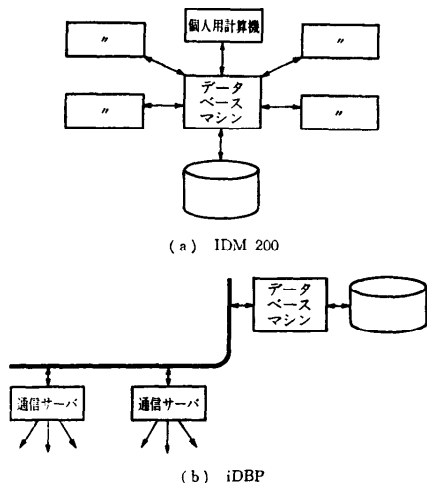


図-6 データベースサーバ型データベースマシン

性能として、RDM 1100 や IDM 200 は関係データベースシステムを内蔵して毎秒5ないし8トランザクション以下という値を表示している。したがって超高性能をめざした製品ではなくて、計算機のハードウェア価格の低下をうまくとらえた専用装置であることがわかる。

集積回路技術の発展は、これからいくつかの方向からこの分野に影響を与えていくと考えられる。

(1) 中間記憶。主記憶装置と2次記憶装置との間の呼出し時間の断絶をうめるために、低速大容量の半導体記憶をディスクキャッシュ的に使う方式がある。記憶装置自身に論理機能があるわけではない。データベース応用にかぎらず、汎用性があり、有効であろう。Intel社の半導体ディスクFAST-3800シリーズ、科学技術用高速計算機CRAY X-MPシリーズのSSD (Solid-state Storage Device) などはこの方向である。SSDは64~256メガバイトの容量であるが、10ギガビット/秒の転送速度を有する。データベースの立場からみた半導体記憶の難点は、揮発性である。不揮発性を獲得する努力も行われているが、補助電源を必要としたり、書換えに時間がかかったりして、なお十分でない。磁気バブル記憶が個人用計算機の補助記憶として人気を博しているのは、不揮発性でかつ物理運動をとまなわなないためである。EDCは磁気バブル記憶を使ったデータベースマシン試作機であった。こうした新しい媒体の探求もおこたってはなるまい。

(2) データベースチップ。単なる半導体記憶にとどまらずに、論理機能も組み込んだチップを考えるこ

とができる。半導体記憶の歴史が始まった時点で、論理と記憶とを均一にとらえることが可能になったはずで、もうそろそろ記憶と論理とを統合一体化した知能チップが出現してもおかしくない。そこに達する前に、比較的単純なデータベース操作を組み込んだデータベースチップが考えられ、提案も出つつある^{21), 22)}。

(3) 多様なデータへの対応。データベースシステム全般にいえることが、ここでも成立する(6.5参照)。これまでに研究、商品化されてきたデータベースマシンは、もっぱら文字データを対象としていた。これを図形、音声などにも拡張するべきである。レーザディスクの応用も興味ある課題であろう。

6.4 分散型データベースシステム

計算機網(ネットワーク)は、分散した計算資源を物理的に統合する。分散型データベースは、計算機網で物理的に結合されたデータを、論理的に統合してデータベース化する。

大規模な網方式におけるデータベースも重要であるが、分散型データベースシステムの当面の課題は、構内計算機網上のシステム構成方式であろう。データベースマシンの項で述べたように、商用のデータベースマシンはこの方向を先取りしようとしている。しかし個人用計算機を含めたシステムの全体像はまだ明確ではない。比較的近い将来、主記憶1MB、補助記憶100MBていどの高性能個人用計算機が各個人の机上をしめると考えられるが、こうした個人用計算機とデータベースサーバとの機能分担など、未解決の課題は多い。事務自動化とデータベースとの接点として、緊急の研究開発が必要とされている。

6.5 多彩なデータへの高度の対応

成育期のデータベース研究開発は、主として事務データ処理を指向しており、文字列データをおもな操作対象としていた。図形、音声を操作する場合でも、2次情報とでもいうべき文字列表現を付加して、これを手がかりとしていた。これからのデータベースシステムは、文字のみならず、図形、音声、数値などの多彩なデータを直接操作する方向へ進むべきである。人工知能研究との結び付きも、この辺から始まる。

またデータベースに推論機能をもたせた知識ベースシステム研究も盛んになろうとしている。データベースシステム側では、データベース設計の比較的単純な問題の解決に力をそそいでおり、設計支援システムの開発が行われている。こうした面に人工知能研究を結合させた「データベース設計相談(コンサルテーショ

ン) システム」あたりに、両者の有効な接点が期待できるのではないだろうか。

参考文献

- 1) Bachman, C.W. and Williams, S.B.: A General Purpose Programming System for Random Access Memories, Proc. AFIPS (FJCC), Vol. 26, pp. 411-422 (1964).
- 2) McGee, W.C.: Generalization: Key to Successful Electronic Data Processing, Jour. ACM, Vol. 6, No. 1, pp. 1-23 (1959).
- 3) Bleier, R.E.: Treating hierarchical data structures in the SDC time-shared data management system (TDMS), Proc. ACM National Conf. 1967, pp. 41-49 (1967).
- 4) Olle, T.W.: UL/1: A Non-procedural Language for Retrieving Information from Data Bases, Information Processing 68, pp. 572-578 (1968).
- 5) An Information Algebra Phase I Report—Language Structure Group of the CODASYL Development Committee, Commun. ACM, Vol. 5, No. 4, pp. 190-204 (1962).
- 6) Childs, D.L.: Description of a Set-Theoretic Data Structure, Proc. AFIPS (FJCC) Part One, Vol. 33, pp. 557-564 (1968).
- 7) 三井情報開発: ソフトウェア・パッケージの利用状況—商用 DBMS の利用実態とユーザ評価, 日経コンピュータ No. 20 掲載予定.
- 8) ACM SIGMOD RECORD, Vol. 12, No. 3 (Apr. 1981).
- 9) CODASYL Data Description Language Committee Journal of Development 1981, Publishing Centre, Canadian Government (1981).
- 10) CODASYL COBOL Committee Journal of Development 1981, Publishing Centre, Canadian Government (1981).
- 11) CODASYL FORTRAN Data Base Facility Journal of Development 1980, Publishing Centre, Canadian Government (1980).
- 12) CODASYL End User Facilities Committee Journal of Development 1980, Publishing Centre, Canadian Government (1980).
- 13) Codd, E.F.: A Relational Model of Data for Large Shared Data Banks, Commun. ACM, Vol. 13, No. 6, pp. 377-387 (1970).
- 14) Codd, E.F.: Extending the database relational model to capture more meanings, ACM TODS, Vol. 4, No. 4, pp. 397-434 (1979).
- 15) Codd, E.F.: Relational Database: A Practical Foundation for Productivity, Commun. ACM, Vol. 25, No. 2, pp. 109-117 (1982).—邦訳 bit 1982年9月号.
- 16) Chamberlin, D.D. et al.: A history and evaluation of System R, Commun. ACM, Vol. 24, No. 10, pp. 632-646 (1981).
- 17) SQL/Data System Planning and Administration, IBM SH 24-5014-0 (1981).
- 18) Brodie, M.L. and Schmidt, J.W.: Final Report of the ANSI/X3/ SPARC DBS-SG Relational Database Task Group, Doc. No. SPARC-81-690 (1981).
- 19) ANSI/X3/SPARC Study Group on Data Base Management Systems, Interim Report, ACM FDT, Vol. 7, No. 2 (1975).
- 20) 上林弥彦: 講座データベースの基礎理論(1)~, 情報処理, 23巻, 3号, 7号, 以下続載 (1982).
- 21) 田中 譲: サーチ, ソート・ハードウェアと記号処理への応用, 情報処理, 23巻, 8号, pp. 742-747 (1982).
- 22) 市川, 平川: 連想処理とデータベース, 同上, pp. 748-756 (1982).
- 23) 植村, 前川: データベースマシン, 情報処理叢書 1, 情報処理学会 (1980).
- 24) 有沢 博: データベース理論, 情報処理叢書 2, 情報処理学会 (1980).
- 25) 穂鷹良介: データベースの論理設計, 情報処理叢書 6, 情報処理学会.
- 26) 植村俊亮: データベースシステムの基礎, オーム社 (1979).

(昭和57年8月6日受付)