

〔特別講演〕

なぜデータベースが必要か

筑波大学社会工学系 教授 穂 鷹 良 介

1. データベースとは

データベースの必要性を説く前にデータベースとはどんなものかについて述べたい。

データベースの定義については最近 JIS の情報処理用語 JIS C 6230 - 1977 で

「一つ以上のファイルの集まりであって、その内容を高度に構造化することによって、検索や更新の効率化を図ったもの」

と定義されている。

電子技術総合研究所の西野博士は

「種々の業務（またはユーザ）に利用できるように統合化された共用ファイル」

と定義している。データベースという言葉は一種の流はりことば行語のような側面があって従来のファイル処理であってもそれをカッコよくデータベースと称するときには JIS 流の定義を用いているものと思われる。どちらかという JIS 流の定義の方が広い意味をもつ。

ここでは西野博士流のデータベースに話をしばらくしたい。

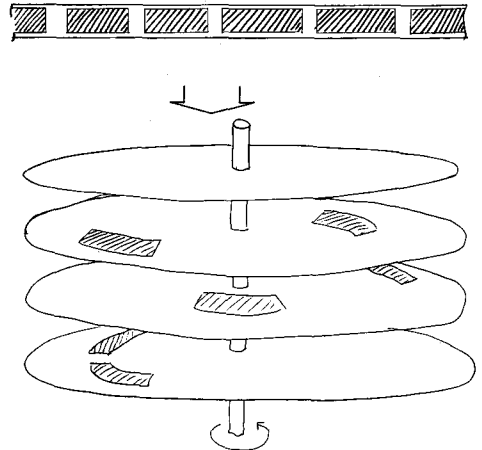


図 1

2. データベースの目的、機能

データの共有化を可能にした第一の原因は、従来の順処理（磁気テープ主体の）しかできなかった記憶媒体に乱処理ができる磁気ディスク等のハードウェアが出現したからである。これによって、今まではやりたくても一通りしかデータの取扱いができなかった状態から、任意の順番でデータにさわるできるようになった（図 1）。

共通に使えるデータをデータベースとして独立した存在価値を認めるようになった一つの理由は、データを作成するのに大変手数が、したがって費用がかかるということが挙げられる（図 2）。

一つの企業などでは、一つの部門

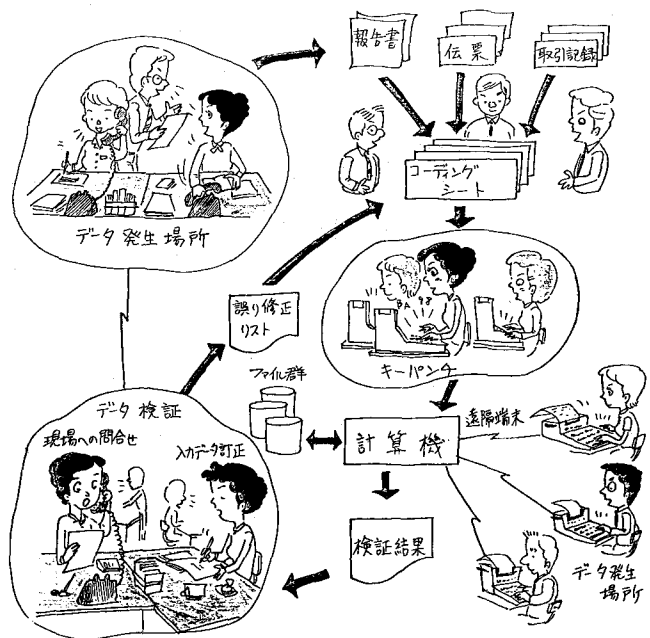


図 2

で使える情報というものが、他の部門と全く隔離して存在する訳ではなく、他の部門の情報と一部重複していることがよくある。同じ情報を整理の仕方が悪いために、異部門で利用するときに別々のものを利用してしまおうというトラブルも共用データの導入により防げる(図3)。

複雑な表現対象を表現するために、それに応じてデータ構造も複雑になる。蓄積データに秩序を持たせ、データベース管理システムでそれを統一的に扱うことにより、個々のプログラムが、データ処理の細かいロジックに意を用いなくてもよくなる(図4)。データベースに対する質問も特別の言語が用意されていて、データ処理の細かな指定をいちいちプログラミングしなくて

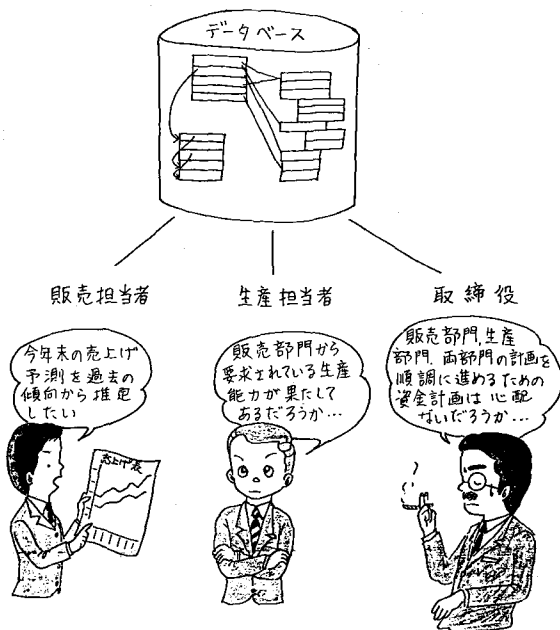


図3

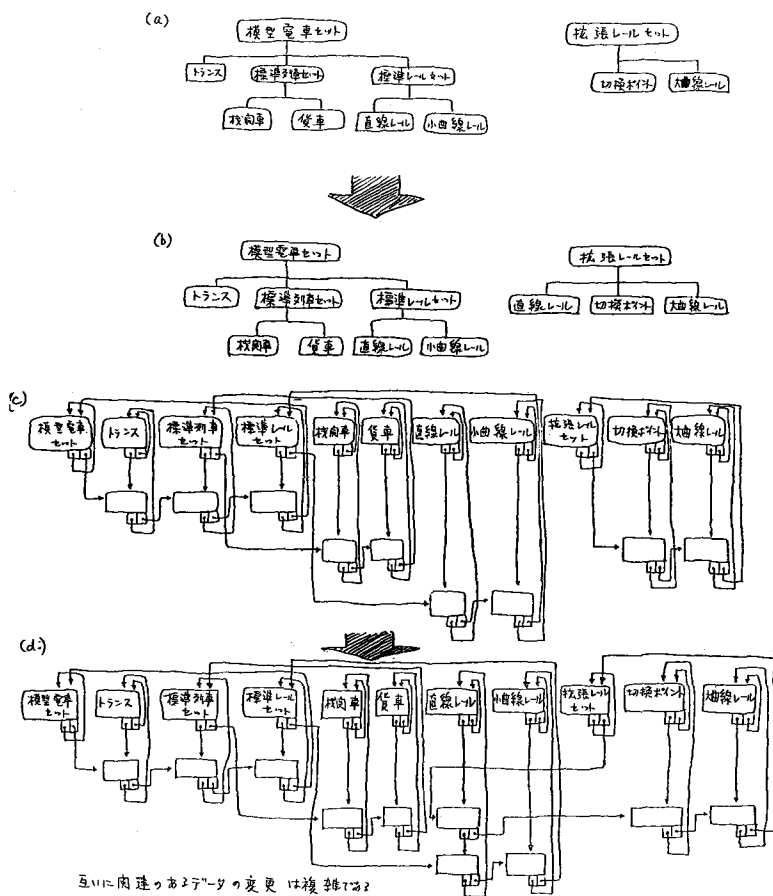


図4

も済む(図5)。

データベースはまた、蓄積データへの標準的なインタフェースを提供するので、自然にデータの処理方法の標準化に役立つ(図6)。データ処理要員の教育も安定した標準的な方法を樹立することができる。

供給者

供給者コード	氏名	住所
HD 047	田辺	小樽
HD 060	山元	札幌
TK 105	阿部	東京
TK 424	三原	東京

プロジェクト

プロジェクトコード	プロジェクト名
A 104	A 組立
B 105	B 組立
C 020	C 製品
D 204	D 製品

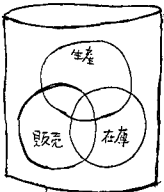
部品提供者

供給者コード	部 品	プロジェクトコード
HD 047	ネジ	A 104
HD 047	歯車	A 104
HD 047	モータ	C 020
HD 047	ブラシ	D 204
HD 060	ネジ	A 104
HD 060	コイル	B 105
HD 060	リレー	C 020
HD 060	タイヤ	D 204
TK 105	スイッチ	B 105
TK 105	ランプ	B 105
TK 105	トランス	D 204
TK 105	導線	D 204
TK 105	車	D 204
TK 105	カバー	D 204
TK 424	抵抗器	A 104
TK 424	コイル	B 105
TK 424	潤滑油	B 105
TK 424	ワッシャ	C 020
TK 424	リモートスイッチ	D 204

検索例「すべてのプロジェクトに部品を提供している供給者名と供給者住所を知りたい」

$$(r_1\{2\}, r_1\{3\}) : P_1 r_1 \wedge \forall P_2 r_2 \exists P_3 r_3 ((r_1\{1\} = r_3\{1\}) \wedge (r_3\{3\} = r_2\{1\}))$$

図 5



標準インタフェース

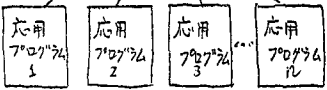


図 6

3. データベースの色々な見方

データベースというものは見る立場によって色々な見方がされる。

一般利用者から見ればそれは、実在の世界を記述した情報がコンピュータによって読み書きされる記憶媒体（ディスクなど）に格納されたもの（図7）で、それから情報を引き出すとするとときに、何もコンピュータについての詳細な知識はなくても、適当な簡便な方法が提供されているものと見える（図8）であろう。

データベースの利用に際しては、一般利用者にはその詳細は知らされていないが、一つのデータ処理、情報加工に際して必ず応用プログラムが介在している（図9）。簡単な検索、特殊な端末を利用してのデータのやりとりの際にもその背後に、指示を受けとってその通りにデータベースとやりとりをする応用プログラムをあらかじめ開発しておかなくてはならない。利用者言語といって、自然語に近い文章でデータベースに対する検索、更新を行なう場合でもその背後には、その利用者言語を解読し、その指示を実行する一種の応用プログラム（利用者言語プロセサ）が存在する。応用プログラマの立場からデータベースを眺めると、それは時には複雑な構造をもつファイルの一種として見える。彼がデータベースと情報のやりとりをするときには、通常の汎用プログラム言語でファイルの入出力を行なう時のようにやはりレコード単位で行なわれる（図10）。

コンピュータ システムの内部ロジックにも明るいシステムプログラマからデータベースを眺めれば、デ

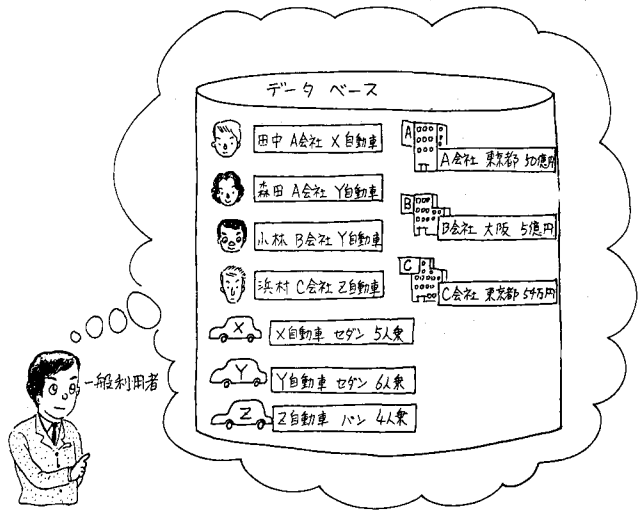


図 7

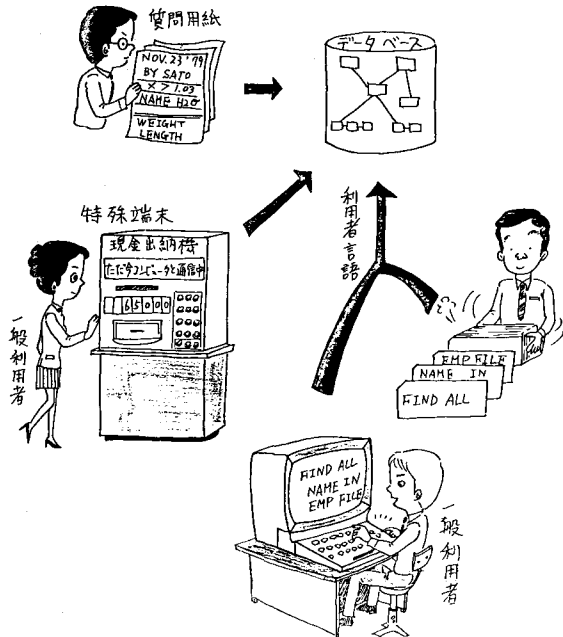


図 8

データベースといってもそれは単なるファイルとして見られデータベース管理システムはそのデータを処理する、オペレーティング システムの制御下のプログラム群と見える(図 11)。

データベースの運用、管理はデータベース管理者と呼ばれるデータベースについては何でもよく知っているデータベースのお守り役によってなされる。彼(または彼等)から見たデータベースには、これらのデータベースの利用の全側面が見えるであろう(図 12)。ことデータベースに関することならばなんでも彼は承知していなくてはならない。

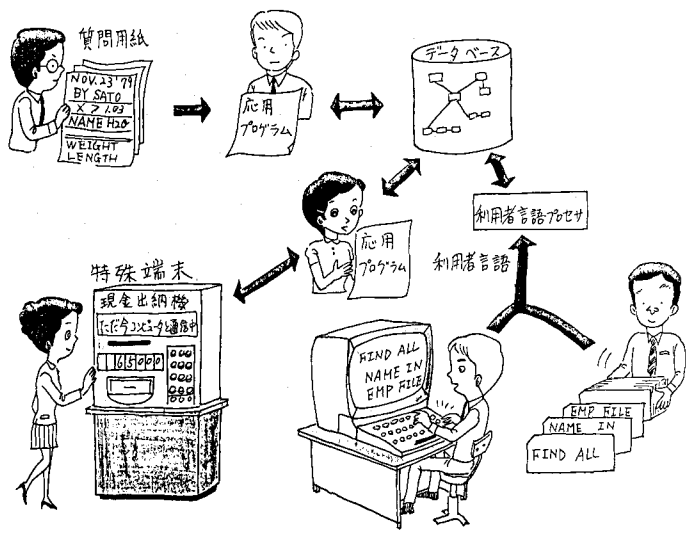


図 9

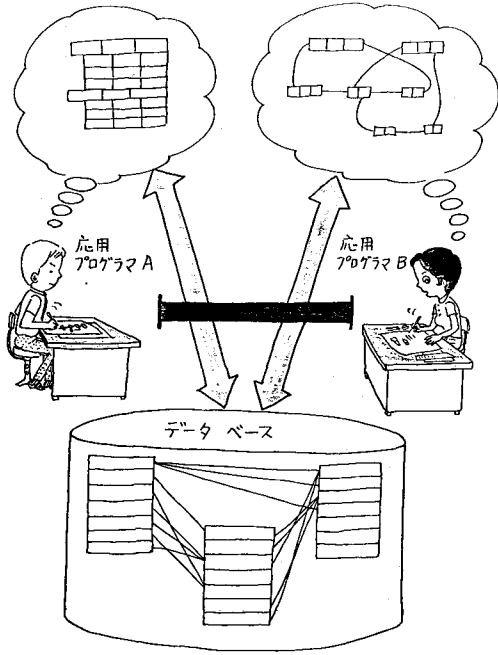


図 10

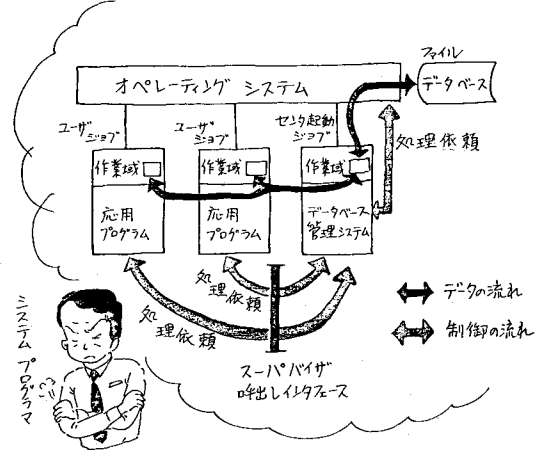


図 11

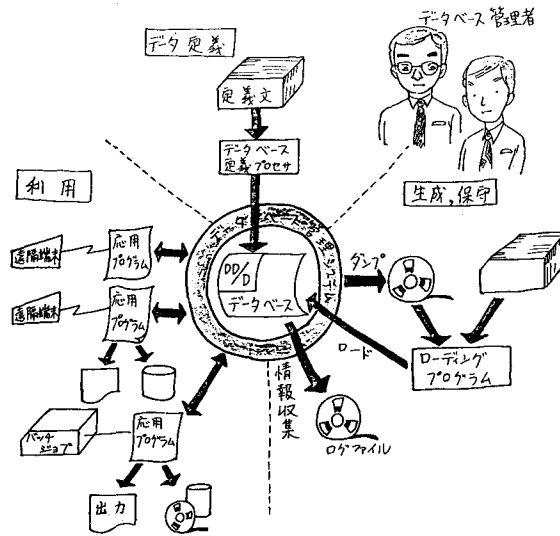


図 12

4. 順処理ファイルの限界

順処理ファイルのデータ処理方法をすこし反省して見る。磁気テープに代表される順処理ファイルは、蓄積データをある一定の順番にしか処理することができない。レコード件数が高々 1 万件位のファイルならば問題はないが、それが 10 万、100 万というオーダーになると、ファイルでのレコードの並び方が処理時間に大きな影響をもつ。たとえば従業員に関するファイルを考えたとき、レコードを従業員番号順に並べるか、あるいは部別、課別に並べるかというようなことが問題になる。従業員の個人個人に関するデータ処理（たとえば残業、欠勤などを含めた給料の計算など）だけが問題になるとときには従業員番号順にレコードを並べておくことで問題はなかりょう。しかし、部別、課別あるいはプロジェクト別に各個人の成績を出したいというような要求のためには、部、課別にレコードを並べたり、あるいはプロジェクト別にレコードを並べ直す必要がある。

レコードの並べ直し（ソート）には多くの時間がかかるから、システム全体としてソートの量がすくなくするようにレコードの並べ方を決めたりする。

マスタファイルの並べ方が以上のような理由で定まると、それに対するデータ処理はレコードの並び方からおのずから制限されてくる。マスタファイルに対する新しい要求が生じたとしても、レコードの並び方がそれに合わないとは簡単には対処できない。上にも述べたように、ファイルの並び方は大抵、処理時間の最も小さくなるように決定されていて、新規の要求への対処のし易さなどというのは考慮されていない。つまり、

順処理ファイルは、新しい問題に対して柔軟性を欠く。

マスタファイルの量が増すにつれて、それをはじめから終りまで読み、かつ、更新する、いわゆるマスタファイル処理は、非常に時間を要する大きなジョブとなる。したがって、マスタファイルに対する問合せとか、データの抽出、更新その他諸々の処理はすべてこの 1 回の処理（マスタファイル処理）に集中する傾向がある（図 13）。

マスタファイルに対する情報処理は将来的にも増加する傾向はあるが減る傾向は通常ない。したがって、

ハードウェアの資源もとかくこのマスタファイル処理をこなすために不足し勝ちとなる。

順処理ファイル処理を採用する以上、ファイルのごく小さな部分の情報を参照するためにも全体のファイルの探索が必要で、これが処理量を増加させる原因にもなる。

順処理ファイルはマスタファイル処理に全処理が集中しやすく、そのためコンピュータの負荷が一時に集中する。

順処理は、乱処理に比べてソフトウェア的にはずっと簡単な処理であるが、それだからといって応用プログラムのロジックがそれに比例して簡単になるかというと、必ずしもそうはいいい切れない。一時期に更新が1レコードに対して複数回なされたり、あるいは事務上の訂正が複数回なされたりするようなケースを考えると、かえって一件一件のトランザクションをその出現のたびに処理するロジックの方が簡単である。たとえばあまりよくないが毎日日記をつけるのは簡単だが、夏休みの終わりにまとめて日記をつけようとするのがかえって大変なのと似ている。

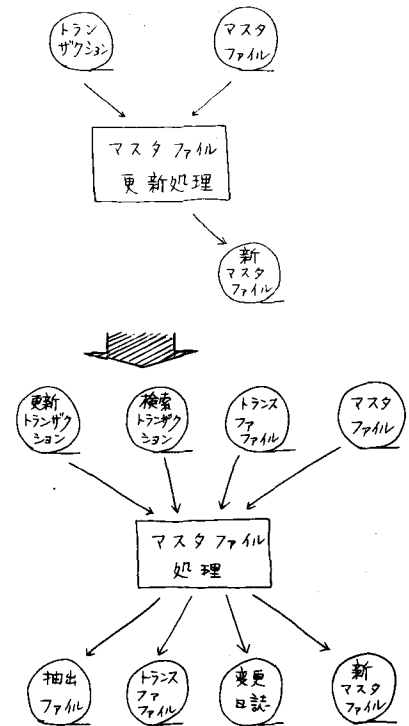


図 13

順処理ファイルは、複雑な更新、それに伴う検索に向かない。

上で従業員に関する情報を、あるいは従業員別に見る場合、あるいは部課別に分類してみる場合とでレコードの並べ方が違うといったが、これと同様に、一つの情報を異なった見方で見たいというとき順処理ファイル1個では対処ができなく、必ず別のソートキーを用いてもとのファイルをソートしてやる必要が出る。事務処理でソートの回数が非常に多いのは主にこの理由による。ソートに要する時間が多過ぎる場合、ソートせずにデータを取り出すデータベースの方法が効率的になる。

ファイルを色々のキーを指定してレコード探索するとき、キーの種類が多くなるにつれて順処理ファイルの維持は非効率的になる。

5. データの共用化

ある企業の設計部門と施工部門とを考える。設計部門で設計したデータは、施工部門に送られて実際の建築または施工が行われるであろう。このとき、同じ会社内であるから、設計部門が考えた個々のデータの意味、その名称が全く施工部門でも同様に用いられているかということ、しばしばそれが同じでない場合が多い。2つの部門には歴史的に物事を独立に考えるなどという習慣が残っていることが多いようである。

しかし、積算など設計で考えたことが実際の設工でどのように実現されたか、設計の良否、積算の良否はどうかなどということの評価しようとしたならば、両部門で使う言葉、情報の定義などは細部に至るまでピ

ピッタリと一致していなくてはならない(図14)。

設計, 施工はピッタリ合ったとしても資材調達部門, 財務部門等と企業の中での部門を広げて行けば行くほど, 共通して使われるデータ, 情報というものがすくなくなる。

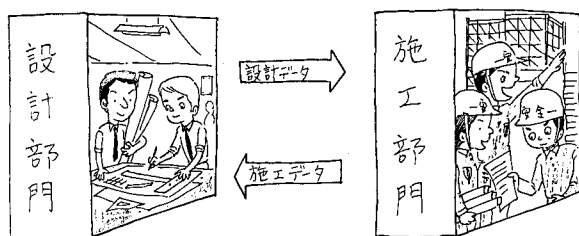


図14

データベースで生じるデータの共用化の

最も大きな問題は, この種のものである。データベースの導入によって, 本来異なる言葉, 異なる意味あい で使われている情報のやりとりがスムーズになるということはない。むしろ, データベースは情報を機械的に処理するが故に, 普段アイマイに使われている異部門間の情報の意味の違いを際立たせる。

2部門間で共用するデータベースが必要なら2部門間でまず通常の事務処理レベルで互いにコミュニケーションのできる言葉の使い方, 処理の標準化が必要である。3部門間の共用データなら3部門間での合意が必要である。データベースが企業全体の共用資産であるという考えならば企業内で, 情報処理法の標準化, 名前のつけ方の標準化等が図られなくてはならない(図15, 16)。

バベルの塔建設が言葉の混乱で失敗したという旧約聖書の話を持ち出すまでもなく, 業務の統合化にはそれに先立って使用概念の統一化が必要である。この準備があってはじめてデータベースは企業内の情報を縦横に使いこなす戦略的な基地(ベース)となりうる。

企業全体で使うデータベースが共用資産であるとするならば, 一企業に二つも三つも用途別にデータベースがあるというのはおかしい訳で, 現状はともかく将来的には統合されたデータベース以外に進む道はない(図17)。

将来にわたって所有する共用データが一種類ということとは, 新しく発生するデータ, 新しいデータへの要求というものはすべて一つのデータベースで処理されなくてはならないということを意味する。順処理ファイルの経験では, マスタファイルのレコードの並べ方が変更されたときには, 関与する応用プ



図15

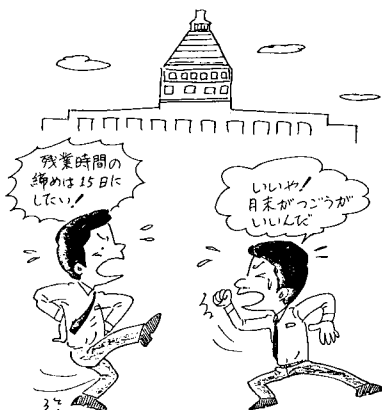


図16

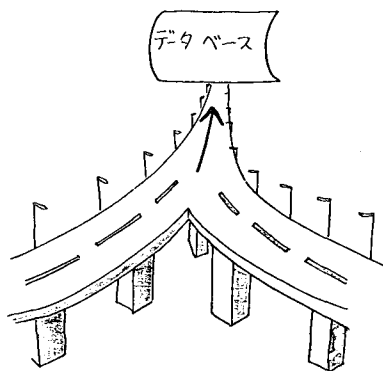


図17

プログラムが一斉に変更されなくてはならなかった。
データベースに対する業務が多種であることを考えると、将来のデータの変更がすこしでも応用プログラムに影響を与えるときには、システム全体が危機に瀕することになる。データベースの環境においては効率よりも何よりも、データが変更されてもそれに関係のない応用プログラムは無関係である（この性質を**データ独立性**という）ことが必要とされる。

近代的なデータベースはこのデータ独立性に対する考慮が十分に払われている（図18）。

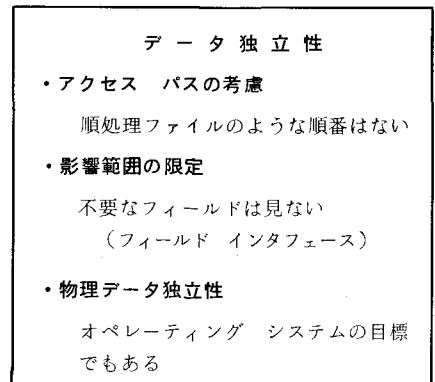


図18

おわりに

データベースの必要性をめぐっているいろいろな角度から眺めた。しかし、基本的に大切なことは、データベースというものはあくまでも日常の情報処理システムの一部であるということである。基礎となる情報処理システムが矛盾だらけで、その上に立つデータベースだけがあらゆることをコンピュータの力を借りて万事解決するというものではない。データベース導入があらゆる業務で可能ということもない。情報処理の種類によってはデータベースを用いるよりも他の方法を用いることの方が現状では得策である場合もある。また、データベースの導入コストが下がれば今までデータベースによる処理が不適當であった分野にデータベースを利用することが有利となることもある。

しかし、データベースの導入、非導入にかかわらず情報システムとして企業の中の概念の統一、処理方法の規準化ということは必要なことのように思われる。それなくしては経営者は明日のための決断に用いる情報を入手することはできないのである。

参 考 文 献

- 1) 穂鷹良介：データベース入門，オーム社，昭和53年12月刊行予定
- 2) 穂鷹良介：ファイル処理 ― 順ファイルの扱い方 ― bit に掲載予定

なお、データベースを情報システム全体の中に位置づけることに関してはPRIDEの開発者 M. Bryce氏から多くの示唆を得た。