

## B 2 CODASYL DBTG案に基づく汎用データベース管理システムの開発と評価について

片山 平, 真名垣昌夫, 水摩正行, 三上 徹, 後藤竜男 (日本電気㈱)

### 1. はじめに

EDPSが、その適応業務拡大と統合化、利用高度化に対処して、個別ファイル・システムから、統合化ファイル(データベース)・システムへ発展して行くことは、必然の姿であると思われる。

一方、多数のデータベース管理システムが開発されて来たが、EDPSの主要部をデータベース化するツールとしては、機能/性能上充分であるとは言えず、本格的な汎用データベース管理システムが求められている。

しかしながら、本格的な汎用データベース管理システムは、種々の複雑な問題を含み、種々の製品、技術領域と関連しているため、一朝一夕に実現することは難しく、その前に総合的な技術開発を必要としている。

我々は、このような総合的技術開発の一環として、CODASYL DBTG案に基づく汎用データベース管理システム“DBMS/2200”を開発しており、引続き、総合評価、各種の進んだ技術的試みを行う予定である。

この報告で、DBMS/2200の概要を紹介すると共に、汎用データベース管理システムの興味ある技術的問題とそれらに対する本システムのアプローチを(紙幅の制限から)2, 3点に絞って記述し、各位の御参考に供したい。

### 2. CODASYL DBTG案

1960年代から、多数のデータベース管理システムが開発されて来たが、それらは、特定の場合には、有効であったかも知れないが、広範囲のEDPSツールとして受容されるには、機能、性能的に不十分であり、又、それらの狙い、特徴は様々であった。このため、本格的汎用データベース管理システムとしてもつべき機能を明確にすることと、その言語等の標準化が要請されるようになった。

このような要請に対して、1960年代後半から、種々の活動が行われて来ているが、それらの中で、CODASYL DBTGの活動とその成果が、特別に注目されている。DBTGの提案<sup>[1]</sup>は、本格的データベース・システムを構築するに充分な基本的機能を調和した形で提供していることと、IDS,<sup>[5]</sup> APLを基にしており、具体的に現実的であること、等から、高く評価されている。

DBTG案の主な技術的特徴は、次のようなものである：

(1) データベース定義記述に、データベース全体の共通的な定義を与えるスキーマと、特定の応用プログラム群(適応業務)固有のデータベース定義を与えるサブスキーマを導入することによって、現実的で、優れたデータとプログラムの独立性と任意のホスト言語とのインタフェースを与えていること。

(2) IDSで使用されているチェーンを論理レベルに昇華して、セットとし、レコード・セット関係によって、インプリメント上の複雑さを避けながら、巧みに複雑な網構造までの任意のデータ構造を定義し、処理する手段を与えていること。

(3) IDSで使用されているカレンシ・インディケータを拡張して、複雑な網データ構造処理を、比較的容易にプログラム出来る手段を提供すると共に、網データ構造処理で懸念されている処理効率の問題を巧みに解決していること。

(4) データベースの柔軟な分割方法と、分割単位(エリア、DBLTG案では、レルム)毎に、きめ細かく処理モードを宣言し、管理する手段を提供して、プログラム間の干渉を防止しながら、データベースの効率良い同時処理を可能に

していること。

(5) データベースの要素クラスへの処理タイプ毎の機密ログ / キー機能を提供しており、きめ細かい機密防止システムを作成出来ること。

今後の汎用データベース管理システムの主流は、DBTG案に基いたものになると考えられる。しかしながら、DBTG案に基いた汎用データベース管理システムを開発するために、次のような問題を解決する必要がある：

(1) 処理効率、記憶効率を上げるための記憶装置、媒体の制御手段、データベース・システムを運用管理するための支援手段、非プログラマ向きの非定形データベース問合せ手段などのようなDBTGが未解決のまま残している技術領域。

(2) DBTG案を汎用データベース管理システムとして具体化するシステム構成技術。

(3) DBTG案は高度過ぎる機能を含んでおり、現時点の実現技術、顧客ニーズに適合するようなDBTG案仕様の部分化。

### 3. DBMS / 2200 の概要

DBMS / 2200 は、NEAC / 2200 中大型機種上のホストOS (MOD-IV等) にうめ込まれたものとして、見方を換えれば、ホストOSにデータベース・ファシリティを強化した拡張OSとして開発されているものである。

DBMS / 2200 は、次のような特徴をもっている：

(1) DBTG案に基づくことにより、格納データの一元化、ネットワーク・データ構造の効率的処理、種々のデータ属性による格納データ選択、応用プログラムの格納データ定義からの独立性の強化、データベースの同時処理管理、機密保護、等のデータベースの基本機能を提供すること。

(2) 格納データ構造を論理データ構造と物理データ構造に分離すること、データベース空間と記憶スペースを完全に分離することによって、柔軟な性能管理手段を提供すること。

(3) データベース処理現況に基くページ入出力制御アルゴリズム、現実的な格納位置管理、記憶階層管理手段の提供等によって、一次記憶、二次記憶間の物理入出力頻度を減少出来ること。

(4) レコード間データ構造 (セット) を物理的に実現するポインタ / インデックス・モードとしては、更新効率を損わない範囲で、レコード選択効率を制御出来る手段を提供すること。

(5) 性能管理、障害管理は、従来の個別ファイル・システムに比べて、より重要な役割を果たすことになると考えられ、性能監視 / 制御、障害予防 / 回復等のデータベース管理者支援機能を重視したシステム設計を行っていること。

(6) DBTG案仕様のほぼ最少サブセットを切出し、初期開発対象としていることから、オーバ・ヘッドが小さく、現時点の利用者スキルに適すること。

以上のような特徴をもったDBMS / 2200 の機能、構成を表す概念図が、図1である。以下図に沿って、システムの機能、構成の概要を記述する。

データベースは、相互に関連がある一元化された格納データの集合である。DBMS / 2200 は、複数個のデータベースを取扱うことが出来る。データベースは、DBTG案に基づく論理データ構造以外に、物理データ構造と論理記憶構造をもつ。これらの詳細は、後に記述されている。

スキーマは、データベース全体の共通的な定義を与えるもので、応用プログラムが作成、実行されるときに、システムによって、参照利用される。スキーマは、論理データ構造を定義する論理スキーマ、物理データ構造を定義する物理スキーマ、論理記憶構造を定義する論理記憶スキーマから構成される。ひとつの論理スキーマに対応して、種々の物理

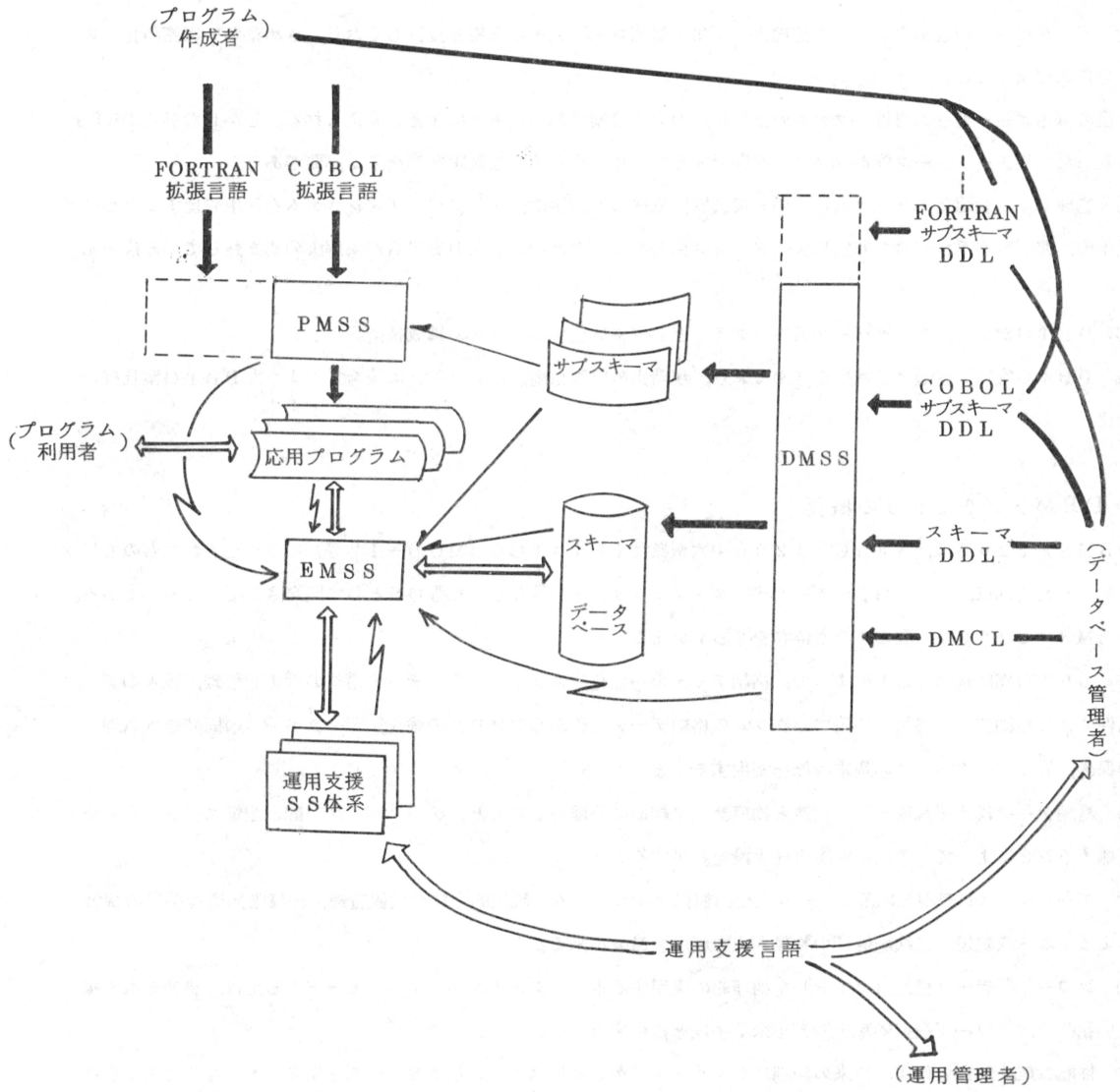


図 1

(註)

- : データベース・ディスクリプタ 応用プログラムの流れ
- : データの流れ
- : 参照
- : 管理, 支援の指示

スキーマ, 論理記憶スキーマが存在し得るが, その範囲での物理スキーマ, 論理記憶スキーマの変更が, 応用プログラムの処理論理に影響せず, 単に, 処理効率のみに影響を与えるということは, 性能管理上重要な特徴である。

サブスキーマは, 特定の応用プログラム群固有の定義を与えるもので, データベースに対する応用プログラマ/プログラムの観方を局所化し, ある範囲で論理スキーマと異らすことが出来, 応用プログラム作成, 変更負担を軽減する。

応用プログラムは、データベースを、参照サブスキーマで記述されたようなものとして処理すると共に、個別従来形ファイルも処理する定形プログラムである。

データベース管理者は、スキーマ、あるいは、応用プログラムの代りにサブスキーマを作成、変更したり、データベース応用システムの性能、障害状況を監視して、制御、回復する責任をもつ利用者クラスである。

スキーマDDL (Data Description Language) は、論理スキーマ作成、変更のために用いる言語である。

DMCL (Device Media Control Language) は、物理スキーマ、論理記憶スキーマ作成、変更のために用いる言語である。DMCLとスキーマDDLを混せて使用することも出来、論理スキーマ、物理スキーマ、論理記憶スキーマを別々のタイミングで、あるいは、同時に作成出来る。

COBOL サブスキーマDDLは、COBOL拡張言語で書かれた応用プログラムが参照するサブスキーマを記述する言語である。

COBOL拡張言語は、COBOLに、データベース処理言語 (Data Manipulation Language, DML) が附加拡張されたもので、応用プログラムの作成に用いる言語である。

実行管理サブシステム (Execution Management Subsystem, EMSS) は、応用プログラムあるいは、他のサブシステムを構成するジョブ・プログラムが、データベース、スキーマ等のDBMS/2200管理対象を処理実行するときこの実行を管理、支援するもので、モニタ・エリアに常駐し、一部オーバレイ・モードで動作する。

EMSSは、次のような機能を果たす：

- (1) ジョブ実行の初期化、終了処理
- (2) データベース、記憶スペース使用の初期化、終了処理
- (3) ページ転送とバッファ管理
- (4) レコード選択
- (5) データの検索、更新
- (6) データベース定義記述要素の選択
- (7) ディレクトリ管理
- (8) 障害管理

ディスクリプタ作成管理サブシステム (Descriptor Management Subsystem, DMSS) は、外部形式のデータベース・ディスクリプタ (スキーマ、サブスキーマ) を、他のサブシステムが処理可能な内部形式、中間形式に変換するものである。

プログラム作成管理サブシステム (Program Management Subsystem, PMSS) は、外部形式の応用プログラムを実行可能な内部形式に変換するものである。PMSSは、ホストOSのCOBOLコンパイラを有効に利用するため、プリプロセッサ方式を取っている。

運用支援サブシステム体系は、データベース応用システムを効率的に、正常に運用するため、データベース管理者、運用管理者を支援するもので、次のようなものからなる：

- (1) 性能管理サブシステム
- (2) 障害管理サブシステム
- (3) 運用管理サブシステム

以上、特徴と機能、構成の概略を記述したが、以下に、処理効率に関係するテーマに絞って詳述したい。

#### 4. データ構造と記憶構造

DBTG案は、データベースの論理データ構造、すなわち、応用プログラマ/プログラムから見えるデータ構造を対象としており、格納データの記憶スペース上への割付け方、格納データを処理する際に必要な記憶スペース要素の管理については、システム製作者に任している。

DBMS/2200では、論理データ構造を自明な形で記憶スペースに割付けず、物理データ構造を導入し、物理データ構造を介して記憶スペースに割付けるようにして、性能管理上の柔軟性を高めている。又、ホストOSの論理記憶構造を強化し、論理記憶スペース単位とデータベース分割部分を、概念上、管理上分離することによって、記憶スペース割付けの柔軟性を高めている。図2は、データ構造、記憶構造、利用者、言語の関係を概念的に表わしたものである。

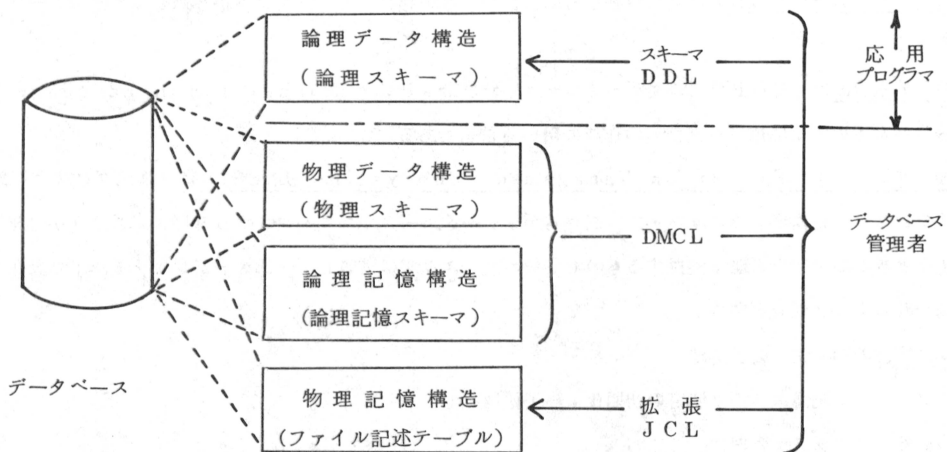


図 2

##### 4.1 物理データ構造

論理データ構造と物理データ構造の間の写像関係は、出来るだけ柔軟であることが、処理パターンの様々な変動に対処出来る物理スキーマを設計出来、性能制御の点から望ましい。しかしながら、無制限に柔軟な写像関係を実現すると、システムが複雑になり、オーバ・ヘッドが増大するので、現在技術に適したものを選ばねばならない。

DBMS/2200では、論理データ構造レベルのレコード(論理レコード)は、

###### (1) 非構造物理レコード

これは、対応論理レコードのレコード内データ構造のみを維持、表現する、すなわち、論理レコードのすべてのデータ・アイテムを表現するフィールドの集合である。

###### (2) 構造物理レコード

これは、対応論理レコードのレコード間データ構造のみを維持、表現する、すなわち、セット・レコード関係を表現するポインタ・フィールドの集合である。

の2つの物理データ構造要素に分解される。

この写像は、論理レコードを自明な形で記憶スペースへ割付ける、すなわち、レコード内データ構造とレコード間データ構造の両方を同時に維持、表現する物理レコードへ写像する従来形的方式(例えば、IDSの場合)に比較して、

データベース要素の記憶スペースへの割当ての柔軟性を高め、きめ細かい性能制御を可能にする。すなわち、ディスク等の回転型磁気記憶装置を中心とする現在のコンピュータ・システムでは、物理入出力操作頻度が、処理効率に大きな影響を与えているが、DBMS/2200の写像方式は、従来形のものに比べて、物理入出力頻度を格段に少なくする機会を与えている。

例えば、一連の処理には、セット内相関選択（言ゆる構造選択）だけされる中継点的役割を果す論理レコードが存在する。論理レコードが、中継点的役割を果す場合、その処理には、そのレコード内データ構造は不要で、DBMS/2200の場合で言えば、対応する構造物理レコードだけで十分な場合が多く、非構造物理レコードを伴って処理する必要がない部分だけ、バッファ領域を、従って、物理入出力回数を節約出来る。処理されるレコードのうち、中継点的な役割をもつレコードの割合は、可成りあると予想され、しかも、普通の応用領域では、非構造物理レコードのサイズは、構造物理レコードのサイズより1桁大きいから、論理レコードを2つの物理レコードに分解する効果は、それだけ高いことが期待される。

DBMS/2200では、ソート・キ、サーチ・キによる選択効率を制御出来るように、物理データ構造要素として、それぞれ、セットインデックス、レコード・インデックスを導入している。

#### 4.2 論理記憶構造

DBMS/2200の論理記憶構造は、ホストOSのファイルに加えて、リージョン、ページ、ラインからなり、記憶媒体装置の属性から独立したもので、利用者は、データベース、ディスクリプタの格納管理、転送制御を論理記憶構造レベルで取扱うことが出来る。

ファイルは、論理記憶構造要素のうち、最大の単位であり、物理記憶スペースへの割付け単位である。

リージョンは、ファイルの分割部分であり、物理記憶スペース上の割付け制御単位である。リージョンは、記憶スペース使用の初期化処理の単位であり、物理レコード、インデックスの論理記憶スペースへの写像の定義記述単位である。

従来のファイル機能を新しく、ファイルとリージョンの2つの機能に分離することによって、記憶装置、媒体の操作を簡単にし、システム内部の制御情報を効率的に利用出来るようにしている。

ページは、リージョンの構成要素で、データベースを支える二次記憶とシステム・バッファ（一次記憶）の間の転送単位である。

ラインは、ページの構成要素で、物理レコードが割付けられる単位である。

データベース分割部分と論理記憶構造要素との関係は、従来データ管理システムでは、固定的である。例えば、データベース分割部分であるエリアが、同時に、物理記憶スペースへの割付け単位であるようなシステムが多い。

一方、DBMS/2200では、その関係は、動的で、処理パターンに従って、制御可能である。すなわち、

(1) レルムは、論理レコード・クラス単位で対応づけられ、1個以上の論理レコード・クラスが、レルムを構成する。

(2) リージョンは、物理レコード・クラスで対応づけられ、1個以上の物理レコード・クラスが、リージョン上に格納される。

例えば、異なるレルムに属している論理レコード・クラスに対応する物理レコード・クラスを同じリージョンに格納してもよく、又、1つの論理レコード・クラスに対応している構造物理レコード・クラスと非構造物理レコード・クラスを異なるリージョンに分けて格納してもよい。

以上のように、格納データの処理モード宣言の対象であるレルムにある程度まで制限されないうで、格納データの記憶スペースへの割付けを行うことが出来、性能制御をインテグリティ管理とある程度まで独立に、より柔軟に実行出来る。

## 5. 記憶装置媒体制御とページ入出力制御

回転型磁気記憶装置を中心とする現在のコンピュータ・システムでは、多くの適応領域で、物理入出力が、性能上の隘路になる可能性が高く、物理入出力時間を減少させることが性能向上に最つとも効果がある。このために、次のような原理が利用出来る：

- (1) アクセス相関の高い物理レコード同士の記憶スペース内距離を出来るだけ短くして、それらが連続して処理される際に、余分の入出力動作を無くしたり、あるいは、シーク時間等を短くすること。
- (2) アクセス頻度の高い物理レコードは、相対的に記憶階層の高位に格納すること。
- (3) バッファからページを追出す場合、近い将来使用されないページ、入出力を伴わないページ、シーク時間等が短くて済むページ等から追出すこと。

DBMS/2200では、これらの原理を次のような手段で実現する：

- (1) 記憶装置媒体制御言語 (Device Media Control Language, DMCL) と物理スキーマ、論理記憶スキーマ
- (2) 初期セット挿入時の物理レコードの位置制御とデータベース再構成
- (3) ドラム/ディスク階層を利用出来るような記憶スペース割当手段
- (4) カレンシ・インディケータを利用したページ入出力制御

### 5.1 記憶装置媒体制御言語 (DMCL)

DMCLは、物理スキーマ、論理記憶スキーマ作成に利用される。

DMCLは、物理スキーマ作成のために、

- (1) セットを物理的に実現するポインタモード
- (2) ソート・キー、サーチ・キーに対するインデックス

と論理記憶スキーマ作成のために、

- (3) ファイル・タイプの指定
- (4) ファイル・リージョン関係
- (5) リージョン・物理レコード・クラス関係
- (6) ページ・サイズ
- (7) 物理レコードの格納制御

を記述する手段を与えている。

物理レコードの格納制御記述には、

- (1) Region Serially: 指定されたクラスに属す物理レコードを、格納順にリージョンの後の方から配置して行く。
- (2) Physical Record Serially: 指定されたクラスに属す物理レコード同士を隣接して格納順に配置する。
- (3) Interval: 指定されたクラスに属す物理レコードを、格納順に指定された間隔でページを飛ばしながら配置して行く。
- (4) Packing: 指定されたクラスに属す物理レコードを、同一ページに格納する際、指定された個数以内に押える。
- (5) First Insert: 指定されたクラスの物理レコードを、初期セット挿入時に位置制御する。次節参照。

(6) Near SPR : 指定されたクラスの非構造物理レコードを対応する構造物理レコードと同じページに配置する。

(7) Near OWNER : 指定されたクラスのインデックスを対応するセットのオーナー構造物理レコードと同じページに配置する。

のような種類があり、データベース管理者は各物理レコード、インデックスの処理態様に合ったものを選ばねばならない。格納制御には種々の方法が考えられるが、上の組みで可成りの範囲の処理態様にまで対処出来る。

### 5.2 初期セット挿入時の物理レコードの位置制御

物理レコードの位置は、関係するすべてのセットを考慮して、決定されることが理想的であるが、実現の難しさ、セット更新毎の位置制御によるオーバ・ヘッドなどから、現実的ではない。

DBMS/2200では、論理レコードが生成されるとき、対応する物理レコード、等は、論理記憶スキーマの定義に従って配置されるが、特に First Insert の指定のある構造物理レコードは、論理レコードが最初のセットに挿入されるとき、そのセットのオーナー構造物理レコードに出来るだけ近い位置に再配置される。再配置は、応用プログラム実行時にはこれ以外のタイミングでは行われず、データベース再構成によってのみ行われる。

### 5.3 ページ入出力制御アルゴリズム

DBMS/2200では、次のような考え方で、ページ入出力制御を行っている：

- (1) 近い将来使用される可能性の低いページから追出すこと。
- (2) 入出力動作を必要としないページから追出すこと。
- (3) アクセス時間が短くて済むページから追出すこと。

このような考え方を実現するため、ページは次のようなクラスに階級分けされる：

- (1) PROCESSING ; 処理中の物理レコードを含むページ。
- (2) CURRENT ; タスク・カレントな構造物理レコードを含むページ。カレント・レコードを中心とした処理が多いため、最つとも再使用される可能性が高い。
- (3) QUASI-CURRENT ; レコード・クラス・カレント、セット・クラス・カレントな構造物理レコードを含むページ。このようなカレント・レコードを中心としたレコード選択処理も多いと予想され、再使用される可能性が高い。
- (4) WRITE-NEEDED ; バッファへ入力後、更新されたレコードを含むページ。このページを追出すとき、元の二次記憶上ページへ書き込むことが必要である。
- (5) ERASABLE ; バッファへ入力後、更新されたレコードを含まないページ。このページを追出すとき、二次記憶上の元のページへ書き込む必要がない。

以上のクラス分けに、ページの書き込み先のドラム、ディスクの違い、LRU情報などによって、追出しページが決定される。

## 6. おわりに

DBMS/2200は、

- (1) 適応業務の統合化が必要な
- (2) 複雑な情報構造を取扱う
- (3) 適応業務の変更、拡張が多い



EDP適応領域に適している。このような特性は、ほとんどすべての業種において、EDPS利用の高度化と共に強くなって行くものである。しかしながら、DBMS/2200で実現するものは汎用データベース管理システム機能のプリミティブな部分のみであり、DBTG案フルセットの実現化方法、オンライン・データベース更新、非プログラマ志向の非定形データベース問合せ、高度な性能管理、障害管理、等が大きな問題として残っている。

紙幅の制限から、DBTG案仕様のサブセット化、性能管理、障害管理等の支援機能等の重要なテーマに触れることが出来なかったが、別の機会に御報告させて頂きたい。

末筆ながら、本研究に便宜と御指導を頂いた当社、中研・村上研究部長、藤野研究部長代理、情報処理グループ、渡部システム営業本部長代理、田中システム営業本部長代理、橋本システム部長、並に当システム開発に当ったDBMS/2200開発グループ員に感謝します。

## 7. 参考文献

- [1] "DATA BASE TASK GROUP REPORT TO THE CODASYL PROGRAMMING LANGUAGE COMMITTEE, April 1971."
- [2] "DATA BASE LANGUAGE TASK GROUP PROPOSAL TO PLC, January 1973."
- [3] "FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYSTEMS" CODASYL SYSTEMS COMMITTEE, May, 1971.
- [4] "DATA BASE MANAGEMENT SYSTEM REQUIREMENTS;" JOINT GUIDE-SHARE DATA BASE REQUIREMENTS GROUP, November, 1970.
- [5] "INTEGRATED DATA STORE, REFERENCE MANUAL" GE, December, 1968.
- [6] "CISS説明書" NEC, 49. 6.
- [7] "INFORMATION MANAGEMENT SYSTEM/360, VERSION 2, GENERAL INFORMATION MANUAL" IBM, 1971.
- [8] "SYSTEM/360 DATA BASE ORGANIZATION AND MAINTENANCE PROCESSOR" IBM.
- [9] "GIS ユーザガイド" IBM, 1969. 7.
- [10] "MARK-IV REFERENCE MANUAL" INFORMATICS, January, 1969.

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング—シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの [https://www.ipsj.or.jp/topics/Past\\_reports.html](https://www.ipsj.or.jp/topics/Past_reports.html) に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

#### 過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 ([tsuji@math.s.chiba-u.ac.jp](mailto:tsuji@math.s.chiba-u.ac.jp)) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>