

ACOS4のデータベース管理システム

内田幸久 (日本電気)

1. はじめに

ACOS4とは日本電気が開発したACOSシリーズ77の内、中大型機用のOSである。ACOS4の特徴の一つは、データベース管理システム(DBMS)とその構成要素とし、利用者が容易にデータベースシステムを導入できるように設計している。

ACOS4のDBMSは下記の3種より成っている。

- ADBS (Advanced Data Base System)
DML, DDLが共にCODASYLの提案に従った言語仕様により開発されたもの。
- DDBS (Descriptor Data Base System)
通常のファイル構造と木構造のデータベースを対象にしたDBMSである。
DMLとスーパーマDDLは、ADBSのサブセットであるがスーパーマDDLではRELATION エントリが付け加えられて見かけ上動的にデータベースと変更してデータベースと処理できるもの。
- NL/II (Nonprogrammer Language/II)
エンドユーザ言語(EUL)でADBS及びDDBSのデータベースを対象にデータベースの検索、更新やレポートの作成を行うもの。

本稿ではこれらDBMSの概要について説明する。

2. 開発の背景と目的

日本電気では既に1968年に汎用DBMとしてCISS (Consolidated Information Storage System) をユーザリリースし鉄鋼、建設、電線の製造業や官庁関係のユーザに利用されている。また1971年にはEOLとしてNL/I (Nonprogrammer Language/I)の開発を完了している。その他HIS社よりIDS, CINCOM社よりTOTALを導入してユーザに提供している。ACOS4では、これらユーザの利用状況と要求を整理すると同時にCISS, NL/Iの開発経験を基礎にしてDBMSの開発を行った。この場合の基が方針は下記の如くである。

- 言語仕様は国際的に標準的なものとする
データベースが普及し利用者の数が増加するに従いシステムの互換性がより入る問題になることが予想される。このためADBSはCODASYLの仕様(1970年のDDLC, 1975年のDBLTGの提案)に従って開発した。またDDBSはCODASYL案で示されている機能に対応するものは全て同一の言語仕様とした。またNL/IIについてはCODASYL EULFGより標準案が示されている。またその性格上互換性が大きい問題となる

のではないと考へてNL/IIの拡張性として開発した。

- OSと一体化する

従来のOSはDBMSのことを考慮せずに設計され、DBMSの開発はOSと別個に行はれていた。このため処理効率は操作性の面で充分なものはなっていた。ACOS4ではDBMSと関連するOSのモジュールはDBMS側の要求を入れて設計した。

- DBMS内部でもそれぞれ有機的に結合したものとす。

ADBS, DDBS, NL/IIと一個のOSに複数個のDBMSが存在する場合それぞれがOSと一体化していると同時に相互に有機的に結合したものでなければならぬ。このためスキーマ/サブスキーマの形式及びこれと作成するプロセッサは同一のものとした。またDDBSは最少の変更で、ADBSに移行できる言語はなし、NL/IIが対象とするデータはADBS, DDBSのデータベースである。

- 通常のシステムからデータベースへの移行が容易に行くと同時に、従来のシステムとの併存が効率よく運営できる。

一般に通常のシステムとDBMSとは機能的にも概念的にも大きなギャップがある。従ってデータベースと採用する作業は容易なものではない。またシステム全体をデータベース化しないときこの部分とのデータの受渡しが無効率になるという場合が多い。

ACOS4ではDDBS, NL/IIが通常のファイルで利用できるため大きな変換作業とはほとんど必要とせずにデータベースが利用できる。続いて段階的に機能を増強しADBSへ進むことにより本格的なデータベースと移行できる。

- コンピュータの利用目的、利用者のレベルに合わせてプログラミング言語を提供する。

給与計算のプログラムの如く繰返し利用される定常業務と、突発的に要求により作られる非定常業務のプログラムがある。また通常コンピュータも使用するものはアプリケーションプログラムであり、一般の人々はプログラムの知識はなくアプリケーションプログラムより与えられたデータを利用するのみで直接目的自身でコンピュータを使用することはできない。

ACOS4では定常業務とアプリケーションプログラム用としてADBSとDDBSと、非定常及びエンドユーザ用としてNL/IIを開発した。

3. システムの概要.

ACOS4のDBMSはOS内の位置はデータ管理の一部であり、次図の如く、データベースアクセス管理とサービスプログラムに分けられる。

3.1 データベースアクセス管理.

① ADBSアクセス

ADBSとはCODASYLで提案された言語はなしに従ってDML, DDL

作られたDBMSでDMLのホスト言語はCOBOLである。次にADBSアクセスとは、DMLを用いてCOBOLで作られたアプリケーションプログラムの要求に従ってデータベースのデータの格納、読出し更新を行うモジュールである。なお対象となるデータベースのデータ構造は木及び網構造である。

② DDBSアクセス

DDBSアクセスとは標準ファイルと木構造のデータに対して下記の処理を行うものである。

- ・ データ属性と値の変換
- ・ バッファと作業エリアの移送処理及びフィールドの再配置
- ・ 条件指定によるレコードの選択
- ・ 複数レコードの集合せ処理
- ・ テーブルファイルのサーチ
- ・ 親子レコードの結合

これらの処理の具体的な説明は後に行うがDDBSの目的はこれらの処理内容をサブ（仮想）スキーマDDLで指定することによりアプリケーションプログラムの作成と保存の工数を削減することである。

またDMLのホスト言語はCOBOLである。

③ NL/IIプロセッサ

NL/IIとはコンピュータについての知識が一切ない人々も簡単な学習で自分自身で直接これを利用できるセルフコンテイングのエンドユーザ言語である。テーブル形式のコーディング用紙に簡単な指定を行うことによりファイルまたはデータベース内のデータに対して検索、更新、レポート作成を行うものである。

NL/IIプロセッサとは要求に応じて処理と実行するものであり、ADBS、DDBSにコンパイル、リンクとなった処理が必要であるがNL/IIでは不要であり直ちに実行させることができる。

3. 2 サービスプログラム

① スキーマプロセッサ

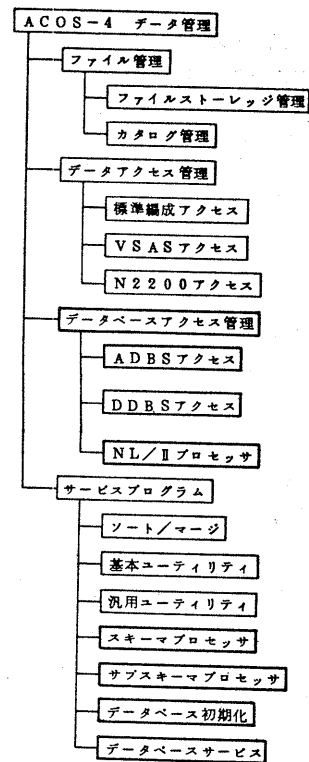
スキーマDDLを入力としてスキーマを作成するモジュールである。これはADBS、DDBS、NL/IIに対して共通に利用できるものである。しかしNL/IIに関してはNL/II言語と同様にテーブル形式の指定によりスキーマを作成することもできるが結果には変りはない。

② サブスキーマプロセッサ

これはサブスキーマDDLよりサブスキーマを作成するものである。個々のDBMSに共通のものである。この場合もNL/IIについてはテーブル形式での指定が可能である。

③ データベース初期化

ADBSのデータベースを作成するためにテストのアプリケーションを行い、同



データ管理におけるデータベース管理システムの位置づけ

時にスキーマ/サブスキーマDDLで指定できないデータベースに対する物理的な属性と定義するものである。

④ データベース サービス

ADBSのデータベースを運用する上で必要な下記の機能を提供するものである。

- データベースの作成

DMLを用いてデータベースの作成することもできる。しかし処理効率等を考えるとユーティリティを利用した方がよい。

- データベースの再編成

データベースに対して各種更新処理を行うとレコードの物理的な格納位置が乱れて次第に処理効率の悪いものへと変化して行く。再編成とは、レコードの並び変えやページサイズの変更等によりデータベースと処理効率の悪いものへと変換するユーティリティ。

- データベースの再構成

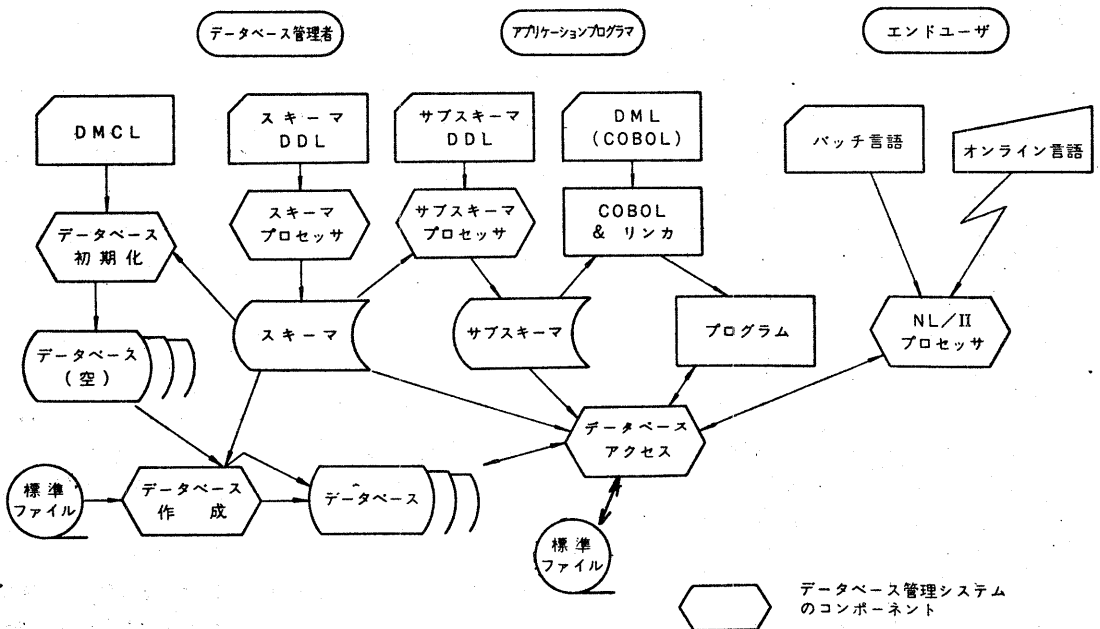
データ構造の変更に応じてデータベースを作り直すものである。これはデータベースに対する処理形式が変化したときに必要なものである。

- データベースの解析

データベース内のレコード分散状態、スペースの利用状況と調べてレポートを作成するもの

次に これら各モジュールの相互関係は下図の如くである。

データベース管理システムの相互関係



4. ADBSについて

既に述べた如くDDL, DMLは、共にCODASYLの提案に従ったDBMSである。従って言語名称や機能についての説明は省略するがデータ構造とこれに対

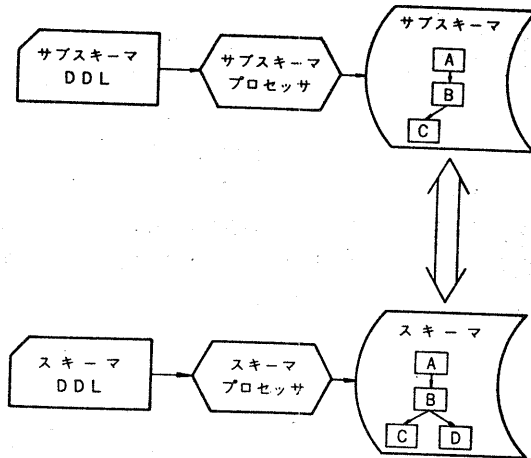
する処理の内容について簡単に記述する。

4.1 データ構造

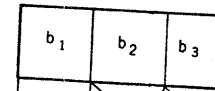
データ構造は次図の如く木構造、網構造が可能である。この場合階段の深さには制限はなく各レコードタイプが任意個のセツトのメンバレコードおよびオーブレコードとなることもできる。

データベース上のデータ構造はスキーマで定義するが実際に利用する場合に一部を変更することが出来る。例として図においてレコードタイプDが不要ならばサブスキーマ上では除いて定義することが出来る。また各レコードの内部ではフィールドの順序の変更、不要フィールドの削除、フィールド属性の変更と指定することが出来る。

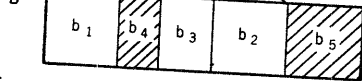
スキーマとサブスキーマの対応



サブスキーマ上のレコード B



スキーマ上のレコード B



4.2 データベースに対する処理

データベースに対する処理はDMLを用いたCOBOLのアプリケーションプログラムと作成して行う。DMLとしては右表の如く12種の動詞と同じように処理できるように作られている。

ADBSで使用可能なDMLとその機能

D M L	機 能
ACCEPT (データベーススキーマの操作)	指定したカレントインディケータの内容(データベーススキーマ)を対応エリアに確保する。
CONNECT (データベース処理)	データベースに格納されているレコードオカレンスを指定したセツトの適するメンバレコードとする。
DISCONNECT (データベース処理)	指定したセツトのメンバとしての関係を切る。
ERASE (データベース処理)	データベースから複数のレコードオカレンスを除去する。
FIND (データベース処理)	データベースからレコードオカレンスを選んで、これに続くDMLの処理対象とする。このときレコード検索表示を指定する。
FINISH (データベース処理)	このプログラムが扱っているレルムを、プログラムから取り消す。
GET (データベース処理)	指定した(FINDで決定した)レコードオカレンスの一部または全体をUWAに移送する。
MODIFY (データベース処理)	レコードオカレンス内の指定したデータアイテムの値を変更し、かつ処理後のセツト関係を保証する。
READY (データベース処理)	レルムの使用モードを宣言し、処理を可能にする。
SET (データベーススキーマの操作)	データキーアイテムの内容を複数データベーススキーマアイテムへ移送する。
STORE (データベース処理)	データベース内に新しくレコードを格納する。
USE (プライバスキーマの操作)	ADBSによって使用するためのプライバスキーマを指定する。

5. DDBS

DDBSはADBSが有する枠組データベースの機能と標準のファイルに、
延長すると同時にサブスキーマを定義し、データベースのサブリポートを指定するの
対して、DDBSでは更にデータベースの変更が可能である。またレコード間の関
係もサブスキーマのレポートで示されたリンクを用いて結合されたものに限定されて
いるのに対してDDBSではキーの値の等しいレコード検索結合することにより、
実行時にDBMS内部で動的にレコード関係が形成されることとなる。

以下DDBSの言語と、続いて最大の特徴であるデータベースの変更とは関係な
いものであるが、述べる。

5.1 言語

標準編成アクセスDMLとDDBS DMLの対応

標準編成アクセス		DDBSアクセス	
SELECT	内部ファイル名の定義とファイル 編成やアクセスモードを指定する。	SELECT	内部仮想ファイル名の定義とファイ ル編成(=仮想編成)やアクセス モードを指定する。
FD	実ファイルとレコードに関する情 報を記述する。	VF	仮想ファイルと仮想レコードに関 する情報を記述する(ただし、 DDLとDMLの非分離記述のと き)。
OPEN	実ファイルをオープンする。	READY	仮想ファイルをオープンする。
CLOSE	実ファイルをクローズする。	FINISH	仮想ファイルをクローズする。
READ	実ファイルから実レコードを読む。	BRING	仮想ファイルから仮想レコードを 読む。
WRITE	実ファイルへ実レコードを出力す る。	STORE	仮想ファイルへ仮想レコードを出 力する。
START	実ファイル内の特定の实レコード へ位置づける。	FIND	仮想ファイル内の特定の仮想レコ ードへ位置づける。
DELETE	実ファイル内の実レコードを削除 する。	ERASE	仮想ファイル内の仮想レコードを 削除する。
REWRITE	実レコードを置き換える。	MODIFY	仮想レコードを置き換える。

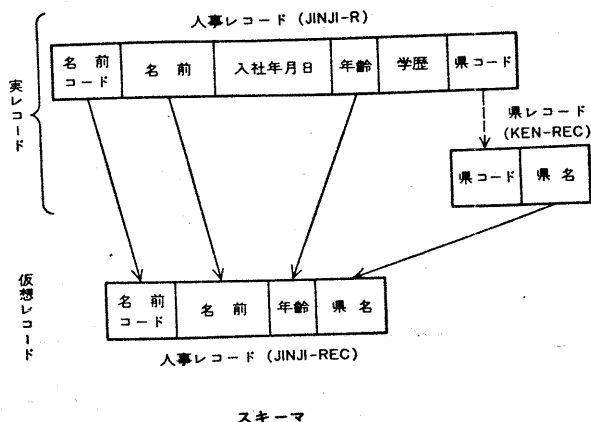
DDBSのDMLはCOBOLと
ホスト言語とするもので表の如く
READY, FINISH, BRING, STO
RE, FIND, ERASE, MODIFY
がある。これはBRINGを除いて
互換性のためにADBSと同一動詞
となっている。しかしこれも通常
のCOBOLに対応するものが存在す
るためDDBSは特にデータベース
を意識しなくとも利用できる。また
BRINGについてはCODASYL案
には含まれないが使い易さを考慮し
て加えた。

次にDDLであるが、これはAD
BSと異なり網構造の定義はできな
い。本構造と標準のファイルについ
て定義するところである。スキーマ
DDLの形式はADBSとは同じ

DDBSとADBSの用語
対応

DDBS		ADBS	
スキーマ		スキーマ	
- データベース - ファイル - レコード - セット		- データベース - エリア - レコード - セット	
仮想スキーマ		サブスキーマ	
- 仮想ファイル - 仮想レコード - 仮想セット - フィールド		- レルム - レコード - セット - フィールド	
DML		DML	
READY	仮想ファイル	READY	レルム
FINISH	仮想ファイル	FINISH	レルム
BRING	仮想レコード		X
FIND	仮想レコード	FIND	レコード
MODIFY	仮想レコード	MODIFY	レコード
ERASE	仮想レコード	ERASE	レコード
STORE	仮想レコード	STORE	レコード
X		CONNECT	レコード
X		DISCONNECT	レコード

仮想レコード記述例



実レコード JINJI-R と KEN-REC を記述します。

RECORD NAME IS JINJI-R WITHIN JINJI-FILE.

01 JINJI-R.

02 NAME-CODE TYPE IS PACKED DECIMAL 6.
02 NAME TYPE IS CHARACTER 80.
02 NYUSHYA TYPE IS CHARACTER 6.
02 NENREI TYPE IS PACKED DECIMAL 2.
02 GAKUREKI TYPE IS CHARACTER 1.
02 KEN-CODE TYPE IS PACKED DECIMAL 2.

RECORD NAME IS KEN-REC WITHIN KEN-TABLE.

01 KEN-REC.

02 KEN-CODE TYPE IS PACKED DECIMAL 2.
02 KEN-NAME CHARACTER 20.

仮想スキーマ

01 JINJI-REC.

SOURCE RECORDS ARE JINJI-R KEN-REC
ASSOCIATED RELATION IS TABLE-LOOK.

02 NAME-CODE PIC 9(6).
02 FILLER PIC X(2).
02 NAME PIC X(80).
02 FILLER PIC X(5).
02 NENREI PIC 99.
02 FILLER PIC X(5).
02 KEN-NAME PIC X(20).

RL TABLE-LOOK;

RELATION TYPE IS HR.

01 JINJI-R.

01 KEN-REC REFERENCED BY JINJI-R
USING KEN-CODE AND KEN-CODE
SORTED IN ASCENDING LOAD 50 DATA.

であり、サアスキーマ DDL もほ
ぼ同じであるが新にリレーションエン
トリ (RL) が付加されている。

この RL により複数個の実レコード
を結合して 1 個の仮想レコードと形
成することによりデータ構造を変
更するものである。図に示した例
では人事ファイルとテーブルファイルの
結合である。人事ファイルではコー
ド長を短くするため県名とコード比
して格納する。そのために別に県コー
ドと県名の対応表のファイルを作成し
実行時にはこれを用いて処理する
必要がある。しかし DDBS では
仮想 (サア) スキーマでこの関係を意
義し、レコード上に県名が格納され
ているような仮想レコードを作成され
ばアプリケーションプログラムでは、テ
ブルファイルのことは一切考える必
要はない。

仮想レコードを利用するプログラムの例

FD LIST-FILE

LABEL RECORD IS STANDARD

DATA RECORD IS PRINT-REC.

01 PRINT-REC PIC X(182).

WORKING-STORAGE SECTION.

01 HEADER-REC.

02 FILLER PIC X(8) VALUE "CODE".

X(4) "NAME".

X(81) SPACE.

X(7) "WAGE".

X(20) "KEN NAME".

PROCEDURE DIVISION.

READY INPUT JINJI-LIST.

OPEN OUTPUT LIST-FILE.

PRINT-HEAD.

WRITE PRINT-REC FROM HEADER-REC AFTER PAGE.
GET-REC.

BRING NEXT JINJI-REC FROM JINJI-LIST.

IF STATUS OF JINJI-LIST = "021"

GO TO END-PROC.

WRITE PRINT-REC FROM JINJI-REC

AFTER ADVANCING 2 LINES

AT END-OF-PAGE GO TO PRINT-HEAD.

GO TO GET-REC.

END-PROC.

FINISH JINJI-LIST.

CLOSE LIST-FILE.

STOP RUN.

5.2 テーブル構造の変更

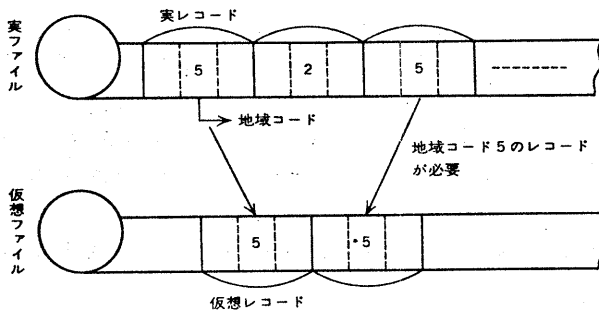
前節では、テーブルサテについての例を示したが、これ以外に(1)～(4)の様な指
を及ぼし、これの組合せが可能である。また本構造のテーブルには(5)の例の如く注意の仮
構構造と指をすることもある。

(1) レコードの選択

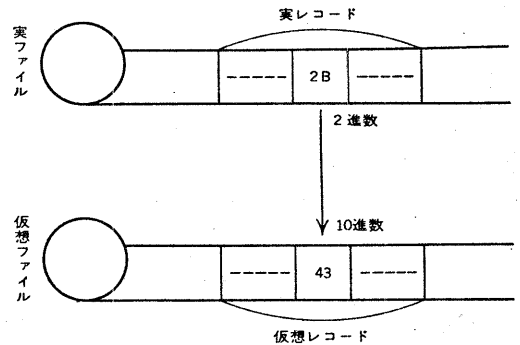
普通預金レコード

店番号	口座番号	地域コード	契約日	最終取引日	通帳残高	利息
-----	------	-------	-----	-------	------	----

普通預金ファイル



(2) フィールドの属性変換



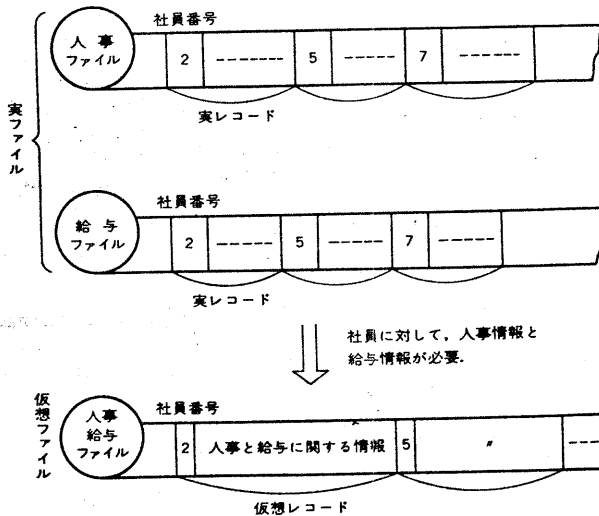
(3) ファイルの突き合わせ処理

人事レコード

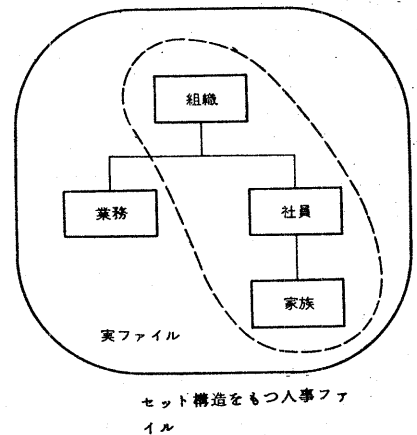
社員番号	人事に関するデータ
------	-----------

給与レコード

社員番号	給与に関するデータ
------	-----------



(4) 階層型リレーション



人事仮想レコード

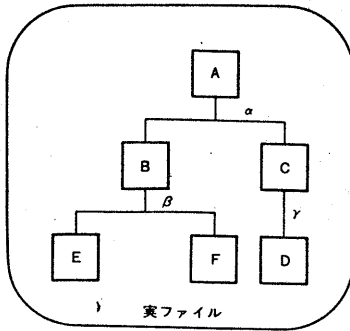
方式1

組織	社員	家族
----	----	----

方式2

組織	社員	C	家族1	家族2	家族3
----	----	---	-----	-----	-----

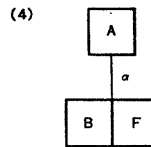
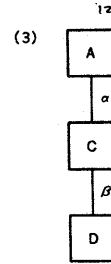
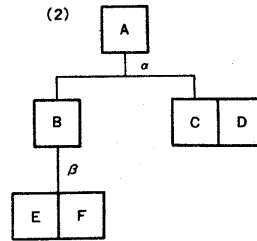
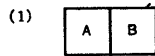
(5) 仮想セット構造の例



実レコード名: A, B, C, D, E, F

実セット名: α, β, γ

この場合以下の例が仮想データ構造として許されます。



備考 A B E F C D B F は、

階層型リレーションによる仮想レコードです。

カードタイプ一覧表

用紙	カード タイプ	意 味	役 割	説 明
1	RC	Run Control	実行の制御	NL/IIプロセッサの実行を制御する必須カード
2	FI	File of Input	入力ファイルの指定	入力ファイルを明らかにする必須カード
	FW	Fields for Working	作業項目の定義	入力データ項目以外に作業用の項目が必要であればそれを定義する。
3	SH	Selection Header	選択の範囲	主に処理効率を上げることが目的とし、この選択範囲で指定されたレコードのみを入力ファイルから読み出す。
	SC	Selection Condition	選択の条件	そのレポートに必要な情報のみをピックアップするための条件を指定する。必要であればピックアップ時に演算/移送の指示ができる。
	SF	Stored Formats Use	登録済レポート形式の使用	NL/IIライブラリに既に登録してあるレポート形式を利用するときに使用する。
4	OM	Output Media	出力装置	レポートをどの装置に出力するかを指定する。
	OD	Output Dimentions	出力レポートの概要	レポートの大きさなど、レポートの形式についてその概要を明らかにする。
	OL	Output Limit	出力制限	どこまでレポートを出力するかを指定する。
5	LD	Lines of Detail	レポート内容の詳細	レポートにどのような情報を盛り込むのかを指定する。また、選択したレコードのソートあるいは集計の指示もここでする。
6	LS	Lines of Summary	非標準集計処理	集計結果をユーザが自分で印字位置を定め、印字したいときに、このカードで指定する。
7	TR	Title of Report	レポート標題	レポートに表紙をつけたいときに使用する。
	TP	Title of Page	ページ標題	ページ標題をユーザが自分で印字したいとき使用する。

6. NL/II

NL/IIとは、ADBSとDDBSのデータベースとエンドユーザが直接利用可能にするものである。NL/IIがエンドユーザに容易に使用できる理由と、下記のようによくなることができる。

- ・ デフォルト値を多く活用している。
- ・ 指定は、非手段で表形式の用紙にパラメータで記述する。通常のプログラミング、COBOL, FORTRANは手順言語であり、処理手順を考えばければならず、プログラムの完成にはかなりの日時を要する。NL/IIでは、大部分の指定は必要とする機能に対応する欄に任意の1文字と記入(チェック)するか、数値あるいは単純な条件式や数式で記述すればよい。

- ・ テータの作成が通常の処理と分離してコンピュータ部門の人が行うためエンタープライズファイルについて考慮しなくてよい。
- ・ コンピュータについての知識はほとんど必要としないで利用できる。

6. 1 エンタープライズのコーディング用紙

エンタープライズが処理を要求する場合に予め準備されたコーディング用紙より自分の処理に必要なものを選べ。その処理に対応する欄に記述すればよい。以後はこの用紙に従ってカードをパンチしてNL/IIプロセッサにデータベースとスキーマ、サブスキーマを併せて入力すればその処理は実行される。なおここでパンチされるカードはその形式に従って3種類のタイプがある。コーディング用紙は13種のカードタイプとよんでNo1よりNo7まで7通り作られている。

またこの13タイプのうち特に良く利用される4種を抽出してNo0のコーディング用紙がある。

これは基本的な機能を利用する初歩的エンタープライズに有効である。次に示す例はこの用紙を用いてレポートを作成した場合である。

6. 2 スケーマ/サブスキーマDDL

NL/IIはA DBS用、DDBS用として作られたデータベースと利用するものであり特に専用のDDLは使用しない。しかしNL/II専用データベースと利用することを考えてNL/II用のテーブル形式のDDLが開発されているが機能には変りはない。

(1) 設定 指定例 (上得意名簿)

得意先ファイルの中から、取り引き金額(トリヒキ)が10万円以上の得意先を抜き出してリストを作成せよ。なお、取り引き金額は次のように編集して印字せよ。

取り引き金額 Z.ZZZ.ZZZ

(2) 指定例

0 リクエスト名 レコード選択と簡易レポートの作成 ACOS-77NL/II

作成者 作成 年 月 日

入力 ファイル名 524 スキーマ名 TOKU プライマリキー 1 区別番号 1

作業項目の定義

項目名	項目1	項目2	項目3	項目4	項目5	項目6	項目7	項目8	項目9	項目10	項目11	項目12	項目13	項目14	項目15	項目16	項目17	項目18	項目19	項目20	項目21	項目22	項目23	項目24	項目25	項目26	項目27	項目28	項目29	項目30	項目31	項目32	項目33	項目34	項目35	項目36	項目37	項目38	項目39	項目40	項目41	項目42	項目43	項目44	項目45	項目46	項目47	項目48	項目49	項目50	項目51	項目52	項目53	項目54	項目55	項目56	項目57	項目58	項目59	項目60	項目61	項目62	項目63	項目64	項目65	項目66	項目67	項目68	項目69	項目70	項目71	項目72	項目73	項目74	項目75	項目76	項目77	項目78	項目79	項目80	項目81	項目82	項目83	項目84	項目85	項目86	項目87	項目88	項目89	項目90	項目91	項目92	項目93	項目94	項目95	項目96	項目97	項目98	項目99	項目100
項目名	項目1	項目2	項目3	項目4	項目5	項目6	項目7	項目8	項目9	項目10	項目11	項目12	項目13	項目14	項目15	項目16	項目17	項目18	項目19	項目20	項目21	項目22	項目23	項目24	項目25	項目26	項目27	項目28	項目29	項目30	項目31	項目32	項目33	項目34	項目35	項目36	項目37	項目38	項目39	項目40	項目41	項目42	項目43	項目44	項目45	項目46	項目47	項目48	項目49	項目50	項目51	項目52	項目53	項目54	項目55	項目56	項目57	項目58	項目59	項目60	項目61	項目62	項目63	項目64	項目65	項目66	項目67	項目68	項目69	項目70	項目71	項目72	項目73	項目74	項目75	項目76	項目77	項目78	項目79	項目80	項目81	項目82	項目83	項目84	項目85	項目86	項目87	項目88	項目89	項目90	項目91	項目92	項目93	項目94	項目95	項目96	項目97	項目98	項目99	項目100

選択の条件

レポートの内容

(3) 出力結果

DATE 50/04/01				PAGE 0001			
シャメイ	シ・ウシ	ユウビ	TEL	トリヒキ			
ウチタ	シ・ウシ	トカ	4-5-6	182	(64)	1118	382400
ヒラサワ	シ・ムキ	トカ	4-5-6	182	(68)	6580	200100
オオタ	バンハイ	トカ	イナキ・シ・ヤノグマ	192-02	(77)	1284	183000
タニタ	タニタ	ス・ナミタ	イズ・ミ	168	(822)	5746	183000
オオヤマ	シ・ウシ	トカ	ハチオシ・シ・ママ	192	(21)	0182	152000
KK	モリゾノ	トカ	タマシ・ナカ・ヤマ	192-02	(45)	5467	240800
ナカムラ	カ・オキ	サイタマ	ケン・トコロサ・ワシ	859	(24)	9460	350000
ナカシ	マ・シ・ウシ	セタガ	ヤタ・マツハラ	156	(822)	0889	240560
イワノ	フ・ンホ・ウタ	トカ	フ・ウシ・ニシ	188	(64)	1118	200000
イワモト	KK	トカ	タマシ・ナカ・ヤマ	192-02	(78)	7728	216600
オハ・ヤシ	シ・ウカイ	シンジ	ムタ・キタシン	160	(871)	7888	181800
トヨカワ	サンキ・ウ	サイタマ	ケン・イルマ・ン・ヒタ・カ	850-12	(8)	0182	152000

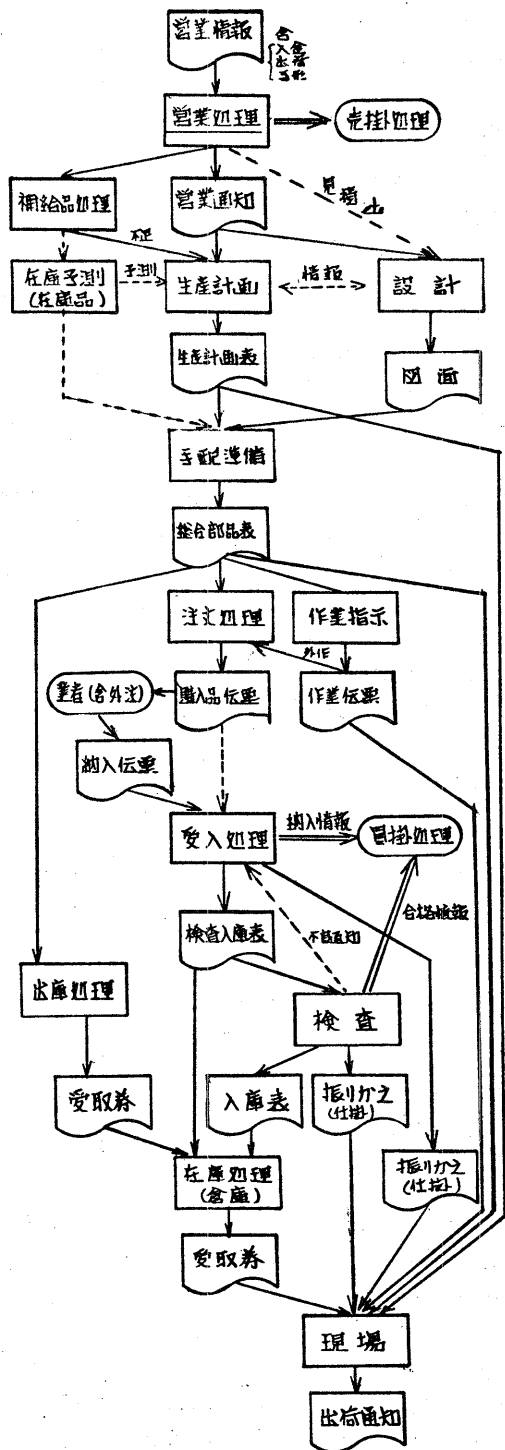
7. データベースの使用例

以上で概説したDBMSが実際に如何に利用されているかを述べる。ここで示す例は車輛の製造会社の営業システムにADBSを適用したものである。営業システムは、日常の営業活動から得る引合情報より、受注情報、出荷情報、入金情報と続く販売情報を一貫処理するものである。このシステムにより収差、加工された情報は販売計画、資金計画等のための管理資料として売掛金補助元帳、前受金補助元帳等の取務資料として活用している。システム内での情報の流れ、出力されるレポートの種類は右図の如くである。この業務で必要とする情報を一個のデータベースにして一元的に管理を行い各アプリケーションプログラムで共同で利用できるようにした。データベース化の目的の主なものは下記の項目である。

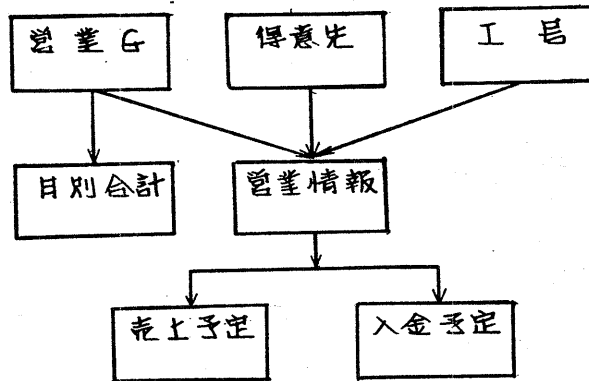
- ・情報を常に最新のものと保つ。
- ・最新の情報をいつでも参考可能な状態にする。
- ・営業システム内の情報の流れと他のシステムとの情報の結合を有機的にもてるシステムとを構築させる。

以上の目的のため、まず第一段階は営業システムに対してバッチ処理でデータベースの実現を計ったものである。第二段階としてはオンライン化と残された業務を総てデータベース化して完全な統合データベース・システムを確立することが今後の課題として考えられている。

営業システムの流れ及び機能説明



データ構造図



スキーマ DDL

ACOS-4 VER. 3.0 SCHEMA PROCESSOR REV.NO: 03.001 DATE: 75-10-06
 SOURCE SCHEMA : STEPS OBJECT SCHEMA : EIGYO

***** TRANSLATION REPORT FOR SCHEMA: EIGYO *****

```

10 SCHEMA NAME IS EIGYO.
20 AREA NAME IS EG-AREA.
30 RECORD NAME IS KUGO-R
40 LOCATION MODE IS CALC USING KG-CD
50 DUPLICATES ARE NOT ALLOWED WITHIN EG-AREA.
60 02 KG-CD TYPE IS CHARACTER 7.
70 02 KG-PU TYPE IS PACKED DECIMAL 5.
80 02 KG-ST TYPE IS CHARACTER 4.
90 RECORD NAME IS CUST-R
100 LOCATION MODE IS CALC USING CS-CD
110 DUPLICATES ARE NOT ALLOWED WITHIN EG-AREA.
120 02 CS-CD TYPE IS CHARACTER 4.
130 02 CS-KM TYPE IS CHARACTER 21.
140 RECORD NAME IS GROUP-R
150 LOCATION MODE IS CALC USING GP-CD
160 DUPLICATES ARE NOT ALLOWED WITHIN EG-AREA.
170 02 GP-CD TYPE IS CHARACTER 1.
180 02 GP-JI TYPE IS CHARACTER 1.
190 02 GP-KM TYPE IS CHARACTER 10.
200 RECORD NAME IS EIGYO-R
210 LOCATION MODE IS VIA CUST-SET SET WITHIN EG-AREA.
220 02 EG-KG TYPE IS CHARACTER 7.
230 02 EG-CS TYPE IS CHARACTER 4.
240 02 EG-GP TYPE IS CHARACTER 1.
  
```

```

1210 SET SELECTION IS THRU GROUP1-SET OWNER IDENTIFIED BY
1220 CURRENT OF SET
1230 THEN THRU YOTEI-SET WHERE OWNER IDENTIFIED BY EG-KG.
1240 SET NAME IS GROUP2-SET
1250 OWNER IS GROUP-R
1260 SET IS PRIOR PROCESSABLE
1270 ORDER IS PERMANENT INSERTION IS NEXT.
1280 MEMBER IS GOKEI-R OPTIONAL AUTOMATIC
1290 LINKED TO OWNER
1300 SET SELECTION IS THRU GROUP2-SET OWNER IDENTIFIED BY
1310 CURRENT OF SET.
1320 END SCHEMA.
  
```