

CODASYL 型のデータベース・マネジメント・システム - DMS 1100 について

原 潔
日本ユニバック(株)

CODASYL の提案しているデータベース機能などのように開発されているのか、またそれほどのような使いかたをしているのかを DMS 1100 の場合について述べる。

1. はじめに

データ組織言語協議会 (CODASYL) におけるデータベースに関する諸活動などにより、最近ますますデータベースに対する関心が高まっている。ユニバックは CODASYL の設立以来、その作業委員会 (タスク・グループ) への参加や、その提案の開発などを通して積極的に CODASYL を支援してきた。実際、ユニバックの最初のコボル・コンパイラは 1960 年にリリースされている。

データベース作業委員会 (DBTG) における 1969 年の報告書、1971 年の報告書の準備にも積極的に参加し、それらを基礎としたデータベース・マネジメント・システム DMS 1100 の開発を行った。DMS 1100 の最初のリリースは 1971 年の 5 月になされている。

2. データベース・マネジメント・システム

CODASYL のデータベースに関する報告書は、データベースの構造を記述するための言語であるデータ定義言語 (DDL) と、データベースを処理するための言語であるデータ操作言語 (DML) の言語仕様を含んでいる。そのほかに、詳細な機能説明はないが、データベース・マネジメント・システムが持つべきいろいろな機能についての必要性が述べられている。そのような機能として次のようなものがあげられている。

1. システム・サポート機能

データベース・システムを運用していく際に必要となるユーティリティあるいはサービス・ルーチンのことで次のようなものがある。

- ・データベースのダンプおよびロード・ルーチン
- ・データベースの編集・プリント・ルーチン
- ・ガベージ・コレクション・ルーチン
- ・統計情報の収集、解析ルーチン

2. データベース・リカバリ機能

ログイン情報やチェック・ポイントを取り、それらの情報を使用してデータベースを正しい状態に修復する機能のことである。

3. スキーマ、サブスキーマの修正機能

スキーマやサブスキーマにおける定義記述を変更し、それに基づいて対応するデータベースを再編成する機能のことである。

4. デバイス・メディア・コントロール言語 (DMCL)

データを物理的な記憶装置に割り当てるための機能で、バッファリング、ページング、オーバフロー領域の指定とそのコントロールなどを指示する。

5. オンライン・ユーザ向けの機能

コンカレントにユーザがデータベースを処理することを可能にするための機能で、複数のラン・ユニットの制御管理や、ロック、キューイング、デッドロックの解除機能などが含まれる。

CODASYM 提案のデータベースの仕様を現実の開発するためにその有効なサブセットがいくつか提示されている。たとえば、英国電算機協会 (BCS) の 1971 年 10 月の会議における J. R. Lucking の報告などがある。

CODASYM の報告書を基礎として、ユニバック 1100 シリーズ用のデータベース・マネジメント・システムとして開発された DMS1100 も CODASYM の仕様の完全な開発ではなく、その有効なサブ・セットになっている。DMS1100 が CODASYM 仕様のどのようなサブ・セットになっているかは、次章以降で述べたい。

DMS1100 は、CODASYM 提案のデータ定義言語、データ操作言語に、この章のはじめで触れたデータベース・マネジメント・システムを実現するために必要となる機能を含め開発されている。それは次のような機能を含んでいる。

- ・データ定義言語機能
- ・デバイス・メディア・コントロール言語機能
- ・COBOL 親言語のデータ操作言語
- ・メインとポインタ・アレイを基礎としたデータ構造
- ・データベースのコンカレント処理機能
- ・データベース・リカバリ機能
- ・効果的なシステム・サポート機能
- ・オンライン・トランザクション処理ソフトウェア (TIP) とのインタフェース

DMS1100 のソフトウェア構成は、上で述べた機能を提供するために次のような構成からなっている。

1. データ定義言語プロセサ (DDL)
2. データ操作言語プロセサ (DML)
3. データ・マネジメント・ルーチン (DMR)
4. リカバリ・ルーチン
5. システム・サポート・ルーチン群

3. データ定義言語 (DDL)

データ定義言語 (DDL) はデータ管理者がデータベースの論理的構造や物理的構造を定義するのに使われる。DMS/1100 の DDL は CODASYM の提案した DDL の機能に DMCL の機能をあわせた機能を持っている。DMCL の機能は、この DDL と ジョブ・コントロール言語 (JCL) により実現されている。

データ定義言語 (DDL) は、スキーマを一意にするスキーマ名の宣言を行なう部分とデータベースの構造を定義する部分と大きく二つの部分からなっている。データベースの構造を定義する部分では、データベースに関する論理的な構造と物理的な構造が定義される。この部分は次の3つのセクションからなっている。

3.1 エリア・セクション

ここでは、データベースを構成するエリアに関するいろいろな特性が定義される。エリアとは論理的にはデータベースのある部分のことであるが、DMS/1100 では、これは外部記憶媒体上の名前づけられた部分のことで、オペレーティング・システムが管理するファイルの概念に対応している。

データベースを構成するエリアが、外部記憶媒体上のどのような部分に割り当てられるかの対応は、ジョブ・コントロール言語により実現されるが、そのほかのエリアの物理的、論理的構造の定義はこのエリア・セクションでなされる。

エリア・セクションで定義される情報には次のものがある。

- ・各エリアに含まれるページの数
- ・ページの大きさ
- ・データ・ページ、オーバ・フロー・ページの指定
- ・エリアの拡張指定
- ・イニシャル・ロードのファクタ

ページとは、外部記憶装置上のデータベースからシステム・バッファに転送される物理的データの単位のことである。ページの数やその大きさはエリアごとに定義できるようになっている。

ページにはデータ・ページとオーバフロー・ページの区別があり、空きページを準備できるようになっている。このほかにロード・ファクタが指定でき、イニシャル・ロード時にページにあらかじめ空きスペースを準備することもできる。

これらの指定により、データ管理者は、記憶媒体の特性や入出力の効率を考慮して、データベースの構造を定義することができる。

これらの概念は、ユニバックが DDL に新しく追加したものである。CODASYM の仕様のうち、TEMPORARY AREA 句、ON 句、PRIVACY LOCK 句 は、DMS/1100 では現在開発されていない。

3.2 レコード・セクション

レコードとは1つ以上のデータ・アイテムの集りに名前づけたものである。レコードは、DMLコマンドの処理の論理的なデータの最小単位である。レコード・セクションでは、データベース中の各レコードの特性が記述される。それらには各レコードが蓄積される方式と、レコードを構成するデータ・アイテムに関する定義がある。

レコードがデータベース中に蓄積される方式は、LOCATION MODE 句により指示される。これには CODASYL の仕様と同じに DIRECT, CALC, VIA SET があるが、DMS/1100 ではそのほかに INDEX SEQUENTIAL が追加されている。

DIRECT はデータベース・キーによる蓄積方式である。DMS/1100 におけるデータベース・キーは、エリア名とページ番号、レコード番号からなる1語の情報である。これはデータベースに蓄積されたすべてのレコード・オカレンスに対し一意に割り当てられる。いったん割り当てられたデータベース・キーは、そのレコードがデータベースから削除されるまでは変更されることはない。これはレコードが MODIFY コマンドなどにより蓄積場所が物理的に変更になっても、データベース・キーは変わらないことを意味している。

CALC はランダムアクセス・ルーチンにより蓄積場所を決める方式で、システムでそのルーチンは提供されているが、ユーザが作成した任意のルーチンを CALC ルーチンとして使用することもできる。

VIA SET は、オーナー・レコードの蓄積位置に從属して、メンバ・レコードであるレコードの蓄積場所が決められる方式である。メンバ・レコードはできるだけ、オーナー・レコードに物理的に近い場所に蓄積される。このアルゴリズムを近接配置方式と呼んでいる。DMS/1100 では、VIA SET のローテーション・モードに対し INTERVAL 句が追加されている。これは、何ページおきにそのレコードを蓄積するかの指示で、階層の下レベルのレコードに対しクラスター・ポイントとしての役を果たしている。

INDEX SEQUENTIAL は、CODASYL の仕様にはなくユニバックが追加した機能である。大容量のレコードに対し、逆方向アクセス能力を提供するために開発された。

レコードをどのエリアに蓄積するかの指示は WITHIN 句で行なわれるが、DMS/1100 では、これにさらに THRU 句が追加されている。これはレコードをエリアのどの範囲に蓄積するかを指示するための機能である。

LOCATION MODE 句、WITHIN 句、INTERVAL 句、THRU 句などを適切に使うことにより、アクセス効率を考慮したレコードのデータベース中での物理的な配置を設計することができる。次で述べる SET の定義機能ともあわせ、これは、論理的、物理的データ構造を設計するのに有効な手段をデータベース管理者に提供していることになる。

データ・アイテムの定義は COBOL にみけるそれと同じである。

CODASYL の仕様のうち、ON 句、PRIVACY LOCK 句、ACTUAL/VIRTUAL、SOURCE/RESULT、CHECK 句、ENCODE/DECODE 句は DMS/1100 では現在、開発されていない。

3.3 セット・セクション

セットとは、1つのオーナー・レコード・タイプと1つ以上のメンバ・レコード・タイプとの間の関係に名前づけられたものである。同一レコード・タイプが、異なるセットのオーナー・レコードやメンバ・レコードに同時になることができるので、セットを組み合わせるこにより複雑なデータ構造が表現できる。

セットにより定義されたレコード間の関係をデータベース上に物理的に表現する方法として DMS/1100 では現在2つの方法がある。1つは CHAIN 方式で、もう1つは POINTER ARRAY 方式である。

CHAIN 方式は、文字通り1つのセット・オカレンス内のレコード・オカレンスをチェーン状にリンクする方法である。それに対し、POINTER ARRAY 方式はメンバ・レコード・オカレンスのアドレスをアレイの形でまとめ、オーナー・レコード・オカレンスがその情報を持つという方法である。セットをどちらの方式で編成するかの指示は SET MODE の宣言でなされる。

CHAIN方式のセットの場合、リンクは無条件に前方向につけられるが (NEXT リンク)、指定によりさらに、逆方向へのリンク (PRIOR リンク) や、オーナーへのリンク (OWNER リンク) をつけることができる。

セットの論理的な構造は、ORDER 句で指示され、FIRST, LAST, NEXT, PRIOR, SORTED の宣言ができる。レコードがセット中に追加されると、その挿入位置はこの ORDER 句により決められる。

レコードがセットのオートマチック・メンバであると、そのレコードが蓄積されると同時に関係するセット・オカレンスの中にリンク付けられる。もしそのレコードがマニュアル・メンバであれば、セット・オカレンスの中にリンク付けるためには INSERT コマンドの実行が必要である。DMS/1100 においては、メンバ・レコードがオートマチックであることは、MANDATORY であることを意味しており、マニュアルであることは OPTIONAL であることを意味している。

CODASYM の仕様にある、オーナーが SYSTEM として指定されるレコードが1つだけからなるセットは DMS/1100 では開発されていない。また ON 句、PRIVACY LOCK 句、SEARCH KEY 句 は現在開発されていない。

3.4 DDL プロセサ

DDL プロセサは独立したデータ定義用のプロセサであり、データ管理者が DDL 言語を使って作成したソース・スキーマを入力し、DDL プロセサや、データ・マネジメント・ルーチンが参照するオブジェクト・スキーマを作成する。DDL プロセサはソース・スキーマのリストと同時に必要ならばスキーマ・レポートを作成する。スキーマ・レポートには、エリブ、レコード、セットのクロス・リファレンス情報やレコードを蓄積するために必要なキー情報、データ構造のパス情報、その他データベース管理者がデータベースを管理する上で有益な情報が含まれている。

4. データ操作言語 (DML)

CODASYMはその報告書において、データベース中のレコードを操作するためのいくつかのコマンドを提案している。これはコボルやフォートランなどのような独立した言語ではない。データベース操作のための機能のセットを示しているもので、その使用にはコボルやフォートランのような親言語を必要としている。

DMS1100におけるDMLプロセサはコボル・コンパイラに対するプリ・コンパイラになっており、DMLコマンドとしてコボル・プログラムに埋めこめる17個のコマンドを提供している。

このCOBOL DMLには、コボル・プログラムにそのプログラムで使われるスキーマを呼びこむ (INVOKE) ことを指示するためのスキーマ・セクションという新しいセクションが追加されている。DMS1100における17個のDMLコマンドとは次のものである。

OPEN	STORE	INSERT	IF	FREE	IMPART
CLOSE	FIND	REMOVE	MOVE	KEEP	DEPART
	GET			LOG	
	MODIFY				
	DELETE				
	FETCH				

データベースは同時に複数のラン・ユニットにより処理されることを前提としているので、データベースの整合性を保つための機能が必要となる。そのための機能の一つとしてDMS1100ではKEEP/FREEのコマンドがCODASYMの仕様とは少し異なって再定義されている。

KEEPはラン・ユニットのカレント・レコードにロックをかける機能を持っており、あとでそのレコードを再び処理するまで、他のどのラン・ユニットにも参照させない目的で使われる。FREEはそのようなロックを解除するための機能を提供している。

そのほかデータ・マネジメント・ルーチンがコンカレントなラン・ユニットを正しく制御管理することができるよう、IMPART/DEPARTのコマンドが追加されている。

ラン・ユニットはIMPART実行後のためDMLコマンドを実行することが許されており、DEPARTでデータベースの処理を終了することを宣言することになる。データ・マネジメント・ルーチンはIMPARTを実行し、またDEPARTを行っていったラン・ユニットのみを制御管理の対象としている。

このほか、ラン・ユニットが任意のデータをロギングとして取るためのLOGコマンドが追加されている。

CODASYMの報告書では、レコードの検索に対し、レコードのデータベース中における位置決めを行うFINDコマンドと、レコードをユーザのワーキング・エリアまで持ってくるGETコマンドを提案しているが、DMS1100ではさらにこれらのことを同時に行うFETCHコマンドを追加している。

以上説明した DML コマンド以外のコマンドは、DMS/1100 では CODASYN の仕様に沿っている。CODASYN の仕様のうち ORDER コマンド、USE コマンドはまだ開発されていない。

ユニバックでは、DML のプログラミングの機能を拡張する目的で、CODASYN の提案している各コマンドの形式に対し、ジェネライズド・コマンドと呼ばれる新しい形式を追加している。これは、実行中にコマンドの処理対象となるレコードやセット、エリアを決定するための機能である。

たとえば、CODASYN の仕様ではレコードの蓄積に対し

STORE record-name .

というコマンド形式が与えられているが、これを

STORE identifier .

と与えるのが DMS/1100 におけるジェネライズド・コマンドの形式である。

前者は record-name で指示された名前を持つレコードがデータベース中に蓄積されることを意味している。これに対し後者は identifier で示されるワーキング・エリアにセットされているレコード名を持つレコードを蓄積することを意味している。この機能は DML で汎用プログラムを作成するような場合に便利である。

このほか、DDN の章で説明したように DMS/1100 ではロケーション・モードに INDEX SEQUENTIAL が追加されており、そのようなレコードに対する効果的な読み取り命令として FIND/FETCH のコマンドが拡張されている。それはカレントなインデクス・キーをもとにした NEXT, PRIOR レコードの検索命令である。

DMS/1100 においては、COBOL DML プロセサはコボル・コンパイラに対するプリ・プロセサになっており、DML コマンドはコボルの CALL 文に変換される。コボルソース・プログラムにおいては DML コマンド自体はコメント文になる。

CODASYN の報告書では、ラン・ユニットにユーザ・プログラムとデータ・マネジメント・ルーチンとの間のインタフェースをつかさどる部分としてシステム・コミュニケーション・エリアが述べられているが、これに相当する部分は、DML がアプリ・コンパイル中にユーザ・プログラム内に自動的に作成する。

DML のアプリ・コンパイル中にスキーマ・セクション中の指示に従って、DDN で作成したスキーマから主にレコードに関する情報がラン・ユニットに結合させられる。スキーマ中の大部分の残りの情報はラン・ユニットの実行中にデータ・マネジメント・ルーチンを介してラン・ユニットに結合させられる。実行中に結合するのはデータ構造に関する情報である。

5. データ・マネジメント・ルーチン(DMR)

データ・マネジメント・ルーチン(DMR)はDMS1100の中で最も中心となるものである。その主な目的は、データベースを維持し、その正当性を保つことである。すべてのDMMコマンドはDMRを通していく。どのラン・ユニットもデータベースを直接アクセスすることは許されていない。

データベースへの蓄積と検索の機能に加えて、DMRはラン・ユニットのロールバックとデータベース・リカバリの機能を持っている。DMRはリエントラントになっているので、ラン・ユニットのマルチ・プログラミング、マルチ・プロセスングの操作が可能になっている。

ラン・ユニットはたいていはそうであるが、必ずしもプログラムと考える必要はなく、トランザクションと考えるもよい。このために、DMRはトランザクション・オリエンティッドなラン・ユニットの要求をコンカレントに処理できるようにになっている。

DMRはこのほかに各種の統計情報を集める機能を持っており、システムの拡張、データベースの再編成などのための有効な情報を提供している。

DMRはユニバック1100シリーズのハードウェアとオペレーティング・システムの機能を十分に利用しており、DMRとラン・ユニット、オペレーティング・システムとの間の関係は、CODASYNの報告書で描かれているデータベース・マネジメント・システムの概念と基本的に同じである。

6. DMS1100の使用経験

前の章まで、CODASYNが提案しているデータベース機能が実際にどのような形で開発されているかをDMS1100の場合に沿って述べてきた。この章では、これらの機能がユーザにどのように使われているのか、またその際の問題点は何なのかを簡単に触れる。この章では、DDMを使ってのデータ定義、DMMを使ってのプログラミング上の話題だけにしぼっている。リカバリ機能やシステム・サポート機能を含めたデータベース・システム運営上の話は別の機会としたい。

DMS1100は、ユニバック1100シリーズのユーザの多くで利用されている。日本でも1971年のファースト・リリース以来30社近くリリースされており、今後さらに増える傾向にある。業種別分野としては製造工業関係の分野が一番多いが、建築業、損保、銀行などで使われている。アプリケーションとしては、部品管理、買材管理、工程管理、図面管理、在庫管理、受注管理、顧客管理、人事情報管理など多種にわたっている。これまでは、ネットワーク構造がセットを定義することにより簡単に処理できるので、製造工業の分野において部品管理などに使われるのが多かったが、最近ではMISの現実的な子プロダクトとしてのデータベース・システムがよく考えられており、トータルな社内情報の管理などに適用する動きがでてきている。

コボルに経験のあるユーザがはじめて DMS1100 を使用する場合、大体一ヶ月程度の学習期間が必要なのである。DDL, DML の機能の理解よりは概念の理解に時間がかかるようである。

タイプとオカレンスの概念、ロケーション・モードとセット・オーダなどは、ともすると DMS1100 の初心者にはデータベースの論理的構造と物理的構造の区別をひきおこしている。オートマチック・セット、マニユアル・セットの概念ともあわせ、レコードが物理的に蓄積されるということと、セットのリンクにつながられることが同一のことがらと思われようである。セットの SELECTION 句については、ほとんど誰もが最初は理解できないようである。

現在では、CODASYL の報告書も広く一般に知られており、その解説も数多くあり、また実際の適用例なども存在するので、以前よりは入内に関して状況は良くなっていると思う。これらの概念の理解の困難さが DDL や DML をむづかしいものと思わせているが、逆にひとたびこれらの概念を理解した者にとっては、セットによるデータ構造の表現や、DML によるデータ操作は、便利であるという感想を抱かせている。このような人達に対しては、むしろ効率などの実際の運用の面を考慮したデータ構造のデザインや DML プログラムの開発をどうしたら良いかなどの新たな問題を提起している。DDL では簡単にセットを定義することができるので、いくらでも複雑な構造が表現でき、現実には運用しきれないものになってしまう可能性がある。

次にセットがどのような構造に対してどのように使われているかを簡単にみてみよう。

部品展開などでは同一レコード・タイプ同士の関係を実現する必要があるが、これは部品レコードに対し構成レコードを作りそれらのセットに ALIAS 句を使用して実現されている。ALIAS は部品管理のようなアプリケーションでは、有効な不可欠機能である。一つのセットが二つ以上のメンバ・レコードを持つ必要はないという議論が CODASYL 内でもあるそうであるが、実際にも、ほとんどの場合セットは一つのメンバしか持たない。しかしそのような例が全くないわけではなく、たとえば保険業務において、契約レコードに対し、事故の支払レコード、未支払レコードの二つのレコード・タイプがメンバになっている例がある。これは、支払、未支払レコードを一緒にするわけにはいかないので、処理からみると同一セットのリンクにつながっていた方がよいからである。マニユアル・セットは人事ファイルにおいて従業員レコードに役職というオーナがあり、役職者だけがリンクづけられている例とか、契約レコード中仮契約のものだけが特別なセットにマニユアルとしてリンクづけられており、本契約と同時にリンクをはずすなどの例がある。

DMS1100 の使用例からみると、セットの階層の深さは 5~6 レベル、広がり（3~4 レコード）というのが平均的なデータ構造であるが、レコードやセットの数が 30~60 におよぶ例もある。

DML で問題になるのは、カレンシの概念である。ユーザ・プログラムのデバック中に一番出会うトラブルは、カレンシに関する処理を誤ったコーディング、キー・イニシヤライズを誤ったコーディングである。CODASYL タイプの DML は一般に、Bachman が述べているようにデータ構造をユーザが航海 (Navigate)

する必要があり、その航海術に慣れるまでが大変なようである。しかし COBOL プログラミングに慣れた者に対しては、データ構造を自由に航海することができるのは便利であろうである。

DML のプログラムが、正しくデータ構造を航海しているかどうか、またその結果データベースは意図した通り正しく作成されているかどうかをチェックすることは必ずしも、プログラム開発に先立ってデバッグをどう行うかを含めたデザインが他のプログラミングの場合以上に必要だと思われる。特にデータベースが正しく作成されているかどうかは、構造が複雑であればあるほどチェックが必ずしも、場合によってはそのチェックのためにユーティリティが必要かもしれない。DMS1100 では、プログラムの中から特定のエラー・コードに対しデバッグングの資料を出したり、データベースのページの内容を出すデバッグング・エイトが提供されている。

DDL/DM は最初から全機能を使いこなそうとするのは大変で、初心者に対しては機能をうまく選択して使わせ、慣れてきたら残りの機能の使用を許していくというやり方がよいようである。

7. その他の機能

第2章でも触れたように、データベース・マネジメント・システムを実現するにはその運用をサポートするためのいくつかの機能が必要である。リカバリ機能やシステム・サポート機能である。これらはそれだけで一つのテーマであり、詳細は別の機会にということにして、ここでは DMS1100 におけるこれらの機能の列挙にとどめておきたい。

リカバリ機能 --- ビフォア・ルック、アフタ・ルック、クイック・ビフォア・ルックをもとにしたロールバック、クイック・リカバリ、ロング・リカバリ、セレクトティブ・リカバリの機能が提供されている。

システム・サポート機能 --- データベースの内容をプリント・アウトする機能、データベースの構造をチェックする機能、リオーガナイズのための機能などが提供されている。

8. おわりに

DMS1100 は CODASYL の提案をベースに、ユニバック1100 シリーズのハードウェア、オペレーティング・システムの機能を有効に使用するようにデザインされており、ユニバック独自のリカバリ機能、システム・サポート機能、オンライン向け機能などを含めた完全な DBMS を目指したデータベース・マネジメント・システムである。DMS1100 は CODASYL の提案のサブ・セットの開発から出発しているが、サブ・スキーマ機能や、ENCODE/DECODE、PRIVACY LOCK 機能なども開発しようとしており、近く FORTRAN DML もリリースされる予定である。

DMS1100 はこれから成長していくソフトウェアといえよう。