

# SCLAMS: データ処理システム

相沢輝昭 畑田のぶ子

(NHK総合技術研究所 情報処理班)

## 1. はじめに

SCLAMS (SCaled Lattice Manipulation System) は小規模のデータベース・システムと言えるものであるが、とくに、非定型の少量のデータをオンライン会話形式で手軽に迅速に検索できることを最大の眼目としている。このことから、このシステムは、システムに蓄積されているデータに関する「小さな質問」を発しつつ、その結果を見てさらに次の適切な質問を行おうという、一連のサイクルを反復しながら、いわば試行錯誤的に最終結果を引き出すという形態の利用に特に適していると言えるであろう。

そのためにデータ・モデルにまでさかのぼって検討し、ファイルよりも小さい Scaled Lattice (以下 S.L. と略記) と称するものを処理の基本単位とした。

試作システムの特徴は次の通りである。

長所としては、

- 1) データの蓄積・検索・各種操作などのすべての処理は文字ディスプレイ装置を介してのオンライン会話型。特に少量のデータの検索や操作が手軽かつ高速。
- 2) 検索に当たっては、まず必要な S.L. を大まかに引き出し、ついでその中の細部のデータを取り出すという二段構えとし、前者についてはキーワード流の検索手法を援用して、システムに蓄積されている S.L. を知らないユーザーでも利用できるようにした。
- 3) エンド・ユーザー言語は数学流の記法に近い人工言語を基本とするが、一部に擬似日本語の併用も許し、数学流記法に伴う抵抗感をへらして人工言語と自然言語の長所をともに生かすようにした。

短所としては、

- 4) システムに蓄積されたデータを更新するには、S.L. 単位で蓄積し直すという形をとる。
- 5) 大量の検索結果を書式を整えて出力することは今のところ不得意。

以下、2. でこのシステムが立脚するデータ・モデルと、システムを構成する主要3モードの概要を説明し、3. ~ 5. に分けて、それら3モードをやや詳しく述べる。通常のデータベース・システムとのごく大ざっぱな関連を求めれば、3. がデータ記述、4. がデータ辞書、5. がデータ操作、となるであろう。特に5. でデータ操作言語の一部に擬似日本語の併用を許す試みについて述べる。最後に6. で、実働化および若干の使用経験に言及する。

## 2. データ・モデルとシステム構成

### 2.1 データ・モデル

論理的にまとめられていて、しかもできるだけ単純なデータの集りの基本単位として、われわれは図1に示すように、一定の対象(例えば人口)を、いくつかの

観点(例えば年次別, 県別, 性別)に着目して分類整理したデータ群, を取上げた。個々のデータは, 観点に対応するものさし(scale)の目盛り(例えば北海道, 青森, ...)の交叉する各格子点(lattice point)に配置されている形で, あるまとまりを持っているわけで, そのような感じをこめてこれらデータ群を以下 S.L. (Scaled lattice) と呼ぶ。

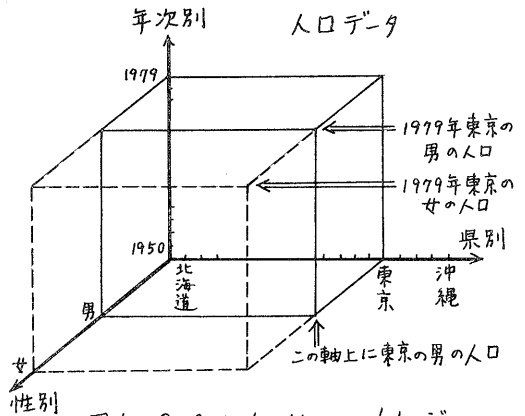


図1 Scaled Lattice のイメージ

通常の表形式のデータは, 紙面に表現される関係上やむなく2次元に折り畳まれることもあるが, 基本的には S.L. である。また, 多様な形式を持っているように見える計算機のファイルも, いくつかの S.L. の自然な結合とみなせる場合が多い。

さらにわれわれは, S.L. を写像としてとらえた。すなわち, 一般に  $F$  が S.L. であるとは, 集合  $S_1, \dots, S_n$  の直積を定義域とし, 今一つの集合  $A$  を値域とする,

$$F: S_1 \times \dots \times S_n \rightarrow A \quad (1)$$

の形の写像と考える。ここで  $S_1, \dots, S_n$  はそれぞれ, 観点に対応するものさしの目盛りを要素とする集合, また  $A$  は  $F$  の表わす対象を表現するのに十分なデータ値の集合である。個々の  $S_i$  を以下では scale と呼ぶ。

例えば図1の S.L. を写像の形で表わせば次のようになる。

$$F: S_1 \times S_2 \times S_3 \rightarrow A_1$$

ここで

$$\begin{aligned} S_1 &= \text{年次別の scale} = \{1950, 1951, \dots, 1979\} \\ S_2 &= \text{県別} \quad \quad \quad = \{\text{北海道, 青森, } \dots, \text{沖縄}\} \\ S_3 &= \text{性別} \quad \quad \quad = \{\text{男, 女}\} \\ A_1 &= \text{人口の集合} \end{aligned}$$

このようにとらえた時, S.L. は, 関係データベースで言ういわゆる3次元正規形にある関係を関数従属性ごとに分解したもので, とみなすこともできる。その意味で SCLAMS は functional database system とでも言うべきものである。

但し, 一つの差異があって, それは, 3次元正規形における関数従属性に対応する関数の定義域(すなわち主キーの集合)が一般にある直積集合の部分集合であるのに対して, 写像としての S.L. の定義域はつねにある直積集合そのものであることである。定義域を直積集合にとることのメリットは記憶容量の節約と検索時間の短縮である。実際, 式(1)のような S.L. を関係データベース流に蓄積するとすれば,

$$(x_1, \dots, x_n, F(x_1, \dots, x_n)), \quad x_i \in S_i, \dots, x_n \in S_n \quad (2)$$

の形の  $n+1$  組をすべて蓄積することになるが, 主キー  $(x_1, \dots, x_n)$  として  $S_1, \dots, S_n$  の要素のすべての組合せが出現することがわかっている場合, この方法は

明らかに無駄である。キーの値  $(x_1, \dots, x_n)$  と、それに対応する値  $F(x_1, \dots, x_n)$  の格納場所との間に簡単な関係が設定できるから、(2) のかわりに

$$S_1, \dots, S_n \text{ の各要素 } \text{および} \quad (3)$$

$$F(x_1, \dots, x_n), x_i \in S_i, \dots, x_n \in S_n$$

のみですませられる。これは記憶容量の節約になる。

また、特定のキー  $(x_1, \dots, x_n)$  に対する値  $F(x_1, \dots, x_n)$  を検索する際に、キー  $(x_1, \dots, x_n)$  を直積集合  $S_1 \times \dots \times S_n$  の中に探索する必要はなく、各  $S_i$  と  $x_i$  と  $S_i$  の中で探索すればよい。これは検索時間の短縮になる。

但し現実のデータの場合、表形式のデータの例に見られるように、しばしば欠番が現われる。これは S.L. の定義域が実質的に直積集合の部分集合であることを意味するが、そのような場合でも、欠番のある S.L. として蓄積するようにしている (3. 参照)。

写像としての S.L. をデータ・モデルに選ぶ今一つのメリットは、集合と写像を記述する数学記法がほぼそのチエンド・ユーザー言語に転用できることであるが、これについては 5. で述べる。

## 2.2 システムの主要構成

SCLAMS は大別して下記の 3 モードから成る。

### 蓄積モード

既成のファイルのレコード中の必要部分に対してのみ、そのフォーマットやデータ型等の記述を行うことにより、その部分のデータを S.L. の形式に分解してシステム・ファイル中に取り込む。以後の操作は専らシステム・ファイル中のデータに対してのみ行われる。システム・ファイル中のデータを更新する機能はないので、その必要が生じた時には、何らかの方法で原ファイルを更新した上で改めて蓄積し直すことになる。

### 検索モード

システムのデータベースを S.L. の集合と考えて、まず、必要になる S.L. を取出すためのモード。その方法に 2 通りあり、方式 1 は、蓄積モードで使った既成の原ファイルごとに、そのファイル・レコードの記述文を手がかりに行うもの。方式 2 は、蓄積されている全 S.L. の対象を扱える用語の K W I C 索引を手がかりとするもの。特に後者は、システムにどのような S.L. が蓄積されているのか前もって知らないユーザーにとりて便利であろう。

### 操作モード

検索モードで取出された S.L. に対して、それらを組合せたり、抽出条件を指定したりして、さらに細部のデータを引出すこと、またそれらデータに対する簡単な計算処理を施したりすることを行う。そのために用いるエンド・ユーザー言語 SML は、数学記法を転用した記号言語であるが、その柔軟性を高めるために一部に擬似日本語の併用も許している。

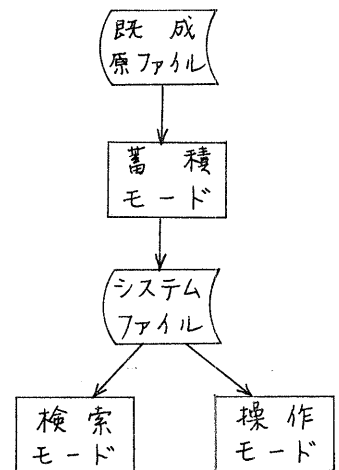


図2 SCLAMSの主要モード

### 3. 蓄積モード

データの供給源である既成の原ファイルは、図3(a)に示すように、キー、フィールド形式、テーブル形式の3部分から成ると考える。

区域	番号	店名	支配人	売上高(月別)													
				1	2	3	4	5	6	7	8	9	10	11	12		
1	1																
1	2																
1	3																
2	1																

キーの部分      フィールド形式の部分      テーブル形式の部分

図3(a) 既成ファイルの基本形式

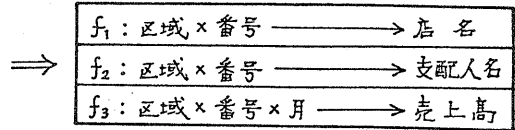


図3(b) 取込まれたS.L.の形

現実のファイルは必ずしもこれらの3部分をすべて含むとは限らない。そのような場合も含めて、なるべく種々の形式のファイルのデータがS.L.として効率よくスムーズにシステムに取込めるように各操作が配慮されている。

#### 操作1: 必要なレコードのみの抽出

いわば前処理で、蓄積容量の節約とデータ処理の高速化のために、大容量ファイルに対して、特定のフィールド項目が特定の条件を満すレコードのみを抽出した、新しい小容量ファイルを作成する。この操作を行った場合、以下の操作は小容量ファイルに対して行う。

#### 操作2: キー部分の処理 (scaleの作成)

キー部分の各項目(図3の例では区域と番号)について、そこに現われる値を要素とするscaleを作成する。また、それらscaleの直種を作り、その全要素が原ファイルのキー部分に現われているか否かをチェックする。現われていない要素があれば、S.L.のそれに対応する値は‘欠番’として取扱えるよう処理を施す。キー部分がないファイルに対しては、レコード番号に対応する一連番号を要素とするscaleを作成する。

#### 操作3: フィールド形式部分の処理

フィールド形式の部分の各項目(図3の例では店名と支配人名)のデータがそれぞれS.L.の値となる(図3(b)を参照)。注意すべきことは、操作2で作成したscaleの直種の全要素を一定のルールで並べ、適宜欠番データを挿入しながらその順にデータに対応させてやることである。

#### 操作4: テーブル形式部分の処理

レコード中の繰返し部分を新たなscaleと見て、そのscaleを作成するとともに、データの取込みも行う。図3(b)のf3がそれに該当し、この取込みの際に、同時に「月別」に相当するscaleが作成され取込まれている。テーブル部分の項目数は可変(レコードごと)でもよく、その可変数はレコード中の他のフィールドにデータとして与えられていても、あるいは繰返し打切りマークの形でテーブル中に与えられていてもよい。さらにテーブル中の繰返し項目同士の間的位置に跳びがあってもよい。

#### 操作5: データ・コードの変換

キー、フィールド形式、テーブル形式のいずれにも含まれる項目のデータについて

ても、取込みの際に、データの表現形式を变换できる。例えばO E オトコに、I をオンナに、等々。

図3(a)のファイルをシステムに取込むのに必要なファイル定義文の例を図4に示す。この例は、ファイル名、キーの部分、フィールド形式の部分の指定が終って、それらが上の方にまとめて提示され、下の方に、テーブル形式の部分も指定しつゝある画面である。

```

SCLAMS MODE=S PH=7 DEFINITION OF TABLE SENTENCE                                79.06.25/18:16
? ( 1)FILE NAME=テスト,DDNAME=TESTFL,PASS WORD=TESTWORD,STORAGE=1,
? ( 2)KEY GROUP=2,SUBKEY NAME=ツイ#,SET ORDER=0,TYPE=B,SIGN=1,START=1,BYTE=2,
? ( 3)CODE NAME= ,ORDER NAME= ,MAKE SEQ= ,
? ( 4)SUBKEY NAME="コ"-,SET ORDER=0,TYPE=B,SIGN=1,START=3,BYTE=2,CODE NAME= .
? ( 5)ORDER NAME= ,MAKE SEQ= ,
? ( 6)SEG NAME=三セ,FIELD NAME=ナマエ,TYPE=C,SIGN=1,START=5,BYTE=12,POINT=0,UNIT= ,
? ( 7)CODE NAME= ,
? ( 8)SEG NAME=三ハイゴ,FIELD NAME=ナマエ,TYPE=C,SIGN=1,START=17,BYTE=20,POINT=0,
? ( 9)UNIT= ,CODE NAME= ,
TABLE NAME= ウリアケ"タ"カ                                OCCURS NAME=
TYPE= B                                                    STOP WORD=
SIGN=
START= 27
BYTE= 4
POINT=
SKIP=
UNIT= ID
CODE NAME=
SCALE NAME= ツキハ"ツ
  
```

図4 ファイル定義文の一例

#### 4. 検索モード

システムに取込まれ蓄積されているS.L.から希望のもののみを検索して、次の操作モードに引渡す機能を持つ。

検索の手がかりとしては、i)取込みに用いた原ファイル名とのS.L.の一覧表、ii)S.L.のデータの素性を説明する対象語から作ったKWIC型索引、のいづれかを用いる。

i)は、その原ファイルの専用ユ-ザ-向きの方法と言えよう。その例を図5に示す。

```

SCLAMS MODE=R PH=8 SELECTION OF FIELD                                          79.06.25/18:30
FILE NAME=キカイ
KEY= ( 1)キカイコート"
*** ツキ" ノ ナカカチ ヒツヨウナ モノ ヨ シテイ シナサイ ***
*( 2)KEY=ネン                                ? *カシタ"シニッスウ キカイ
? *キカイ                                    ? *キキメイ キカイ キカイ
? *カカク                                    ? *シュツク カカク キカイ
? *カシタ"シシヤ                             ? *シメイ カシタ"シシヤ キカイ
? *シヨリ"ク カシタ"シシヤ キカイ
? *ヨテイヒ"                                  ? *ハツキヤク ヨテイヒ" キカイ
? *シヨウ                                    ? *ハツハ"ソコ"ウ シヨウ キカイ
  
```

図5 ファイル単位のS.L. リストの一例

これに対して ii) は、システムに蓄積されている S.L. に関する知識を持たない一般ユーザー向きの方法であろう。「ジンコウ」の近辺の KWIC 索引の例を図 6 に示す。このような KWIC を作成する際に見出しの対象としない、いわゆるストップワードとしては、「スウ、ベツ、ミツド、ニ、ノ、ヲ、…」などの合計 79 語を用いた。

SCLAMS MODE=R PH=3 SELECTION OF KEYWORD

79.06.25/18:29

\*\*\* ツキ"ノ キーワード" ノ ナカカラ テキトウサ モノ ヲ イラヒ"ナサイ (20コ イナイ)\*\*\*  
(アヲサ ノ ニュウリョク シタ キーワード"=シ"ンコウ

```

*****
? 1          *シュヨウ ケイサ"イ シヒョウ キカイ シ"ュチュウ カ"ク。
? 2          *シ"ョウダ" スイト"ウ ネンカン キュウスイ リョウ。
? 3          *キンロウシヤ セタイ ノ セタイ シ"ンコウ。
? 4          *タ"イトシ" ノ シ"ンコウ。
? 5          *コクセイシヨウサ シ"ンコウ。
? 6          *シ"ユウメントウロク シ"ンコウ。
? 7          *シュヨウ ケイサ"イ シヒョウ シュウヒ スイシ"ユン。
? 8          *シ"ョウダ" スイト"ウ ネンカン キュウスイ リョウ。
? 9          *1970 ネン イ ニチ ノ セイカツ シ"カン ト"ヨウヒ"。
? 10         *イ ニチ ノ セイカツ シ"カン ニチヨウヒ"。
? 11         *1970"ネン イ ニチ ノ セイカツ シ"カン ニチヨウヒ"。
? 12         *イ ニチ ノ セイカツ シ"カン ハイシ"ツ。
? 13         *1970 ネン イ ニチ ノ セイカツ シ"カン ハイシ"ツ。
? 14         *ケンミン ショトク カンガイ シ"ユン セイザン。
? 15         *シュヨウ ケイサ"イ シヒョウ コクミン ソウ セイザン。

```

図6 S.L. 名に関する KWIC 索引の一例

## 5. 操作モード

このモードは、検索モードで検索された S.L. のさらに細部のデータをきめ細かに取出すためのものである。そのための指定は、エンド・ユーザー言語として用意した SML (Scaled lattice Manipulation Language) を通して行う。SML は、2.1 で述べたように、S.L. が写像であることを着目して、「集合と写像」を記述する数学流の記法をほぼそのまま転用したものである。

### 5.1 SML の諸相

いま検索モードにより、下記の二つの S.L. が検索されたとして、操作モードにおける SML の使用例を示そう。

F1:  $S1 \times S2 \times S3 \rightarrow$  人口値の集合

F2:  $S1 \times S2 \rightarrow$  テレビ契約数値の集合

ここで

$S1 =$  年次別の scale  $= \{1950, 1951, \dots, 1979\}$

$S2 =$  県 別 "  $= \{\text{ホ"カイドウ}, \text{アオモリ}, \dots, \text{オキナワ}\}$

$S3 =$  性 別 "  $= \{\text{オトコ}, \text{オンナ}\}$

例1 1975 年の東京の男の人口  
(S.L. の個々のデータの抽出)

LIST A;

A= F1(1975, トウキョウ, オトコ);

例2 1975年の男の人口が百万人以上の県  
(S.L.のデータに応じてscaleの要素を分類する)

```
LIST KEN;
KEN= < X: F1(1975,X,オトコ) >= 100マコ >;
```

例3 1960年代の大阪の女性人口の平均値  
(簡単な計算処理)

```
LIST A;
A= MEAN(B);
B= F1(NEN,オオサカ,オメナ);
NEN= < 1960,1961,1962,1963,1964,1965,1966,1967,1968,1969 >;
```

1960年代を表わす年次の集合NENを簡単に

```
NEN=S1.11-20;
```

と指定することもできる。これは年次別scale S1の11~20番目の要素の意。簡単な計算処理として可能なものは今のところ次の6種で、それぞれ、個数、最大値、最小値、合計、平均値、分散を表わす。

```
COUNT, MAX, MIN, SUM, MEAN, VAR
```

例4 1975年の女の人口が百万人以上の県の1970年のテレビ契約数  
(二つのS.L.の結合処理)

```
LIST A;
A= F2(1970,KEN);
KEN= < X: F1(1975,X,オメナ) >= 100マコ >;
```

例5 1975年の女の人口が百万人以上、かつ同年のテレビ契約数が20万以上の県  
(集合演算 &, |, および -)

```
LIST A;
A= B & C;
B= < X: F1(1975,X,オメナ) >= 100マコ >;
C= < Y: F2(1975,Y) >= 20マコ >;
```

集合演算としては、&(交わり)、|(結び)、-(差)が可能であるが、演算対象を、同一のS.L.の値域の部分集合同士、もしくは同一のscaleの部分集合同士に制限している。

例6 男女人口が1960年代にとともに百万人を越えたことのある県  
(論理記号 $\forall$ ,  $\exists$ の使用)

```
LIST A;
A= < X: E'Y#B, A'Z#S3 (F1(Y,X,Z)>100マコ) >;
B= S1.11-20;
```

この例で  $E'Y \# B$ ,  $A'Z \# S3$  の部分を通常の記法で書けば  $\exists Y \in B, \forall Z \in S3$  となる。また、Bは例3におけるNENと同じ要素を持つ。

以上の例からもわかるように、SMLによるリクエスツ文の記述の一般形は図7のような形をしている。

ここで  $a_1, \dots, a_m$  は  $b_1, \dots, b_n$  のうちのどれかであり、また  $b_1, \dots, b_n$  の並べ方は任意である。

```
LIST a1, ..., am;
b1 = expression 1;
b2 = expression 2;
⋮
bn = expression n;
```

図7 SMLによる記述の一般形

expression の型は基本的に次の 6 種

- i) 数字または文字リテラル 例: 19, トウキョウ, 100マン
- ii) 集合関数値 例: COUNT(x), MAX(y)
- iii) S.L. の値 例: F(x<sub>1</sub>, ..., x<sub>n</sub>)
- iv) 集合演算式 例: x & y, x | y, x - y
- v) 集合の定義式 例: <3, 5, 7, 11>, <トウキョウ, ナゴヤ, オオサカ>, <x<sub>i</sub>: F(x<sub>1</sub>, ..., x<sub>i</sub>, ..., x<sub>n</sub>) < y >
- vi) scale の要素の略式表現 例: S.1, S.11-20

左辺の変数記号  $s_i$  は当然右辺の表現式中に現われるが、その埋込みの程度は任意である。例えば、上の例 3 は次のように書いてもよい。

```
LIST MEAN( F1(S1.11-20, オオサカ, オナ) );
```

検索モードから渡されて来た S.L. およびそれらに関係する scales に対してシステムは、F1, F2, ..., および S1, S2, ... の形の略号を与えた上で図 8 のような情報を提示して来る。ユーザーはそれを見ながら SML 文を作成することになる。

変数記号  $s_i$  としては、システムの与えて来る略号 F1, ..., S1, ... と、集合関数名である COUNT, SUM, ... 等の登録語以外の 8 文字までの語が任意に使用できる。

```

SCLAMS MODE=M PH=2 SPECIFY REQUESTS BY SML                                79.06.26/14:39
F1 : コロセイヨウサ セイ スウ                                             ショイ:テスト ショイ
S1 : ネット" ^"ツ                                                       5/      1:1950                1955                1960
S2 : セイ クイセイ ^"ツ                                                 2/      1:フツク セイ        シュン セイ        フツク セイ
S3 : ト"ウフカク ^"ツ                                                   47/     1:ホウカイト"        アオモリ            イワテ
F2 : シ"ユウエン トウロク セイ スウ                                       ショイ:テスト ショイ
S4 : ネット" ^"ツ                                                       22/     1:1952                1953                1954
S3 : ト"ウフカク ^"ツ                                                   47/     1:ホウカイト"        アオモリ            イワテ

```

図 8 S.L. リストの表示例

### 5.2 擬似日本語との併用

上で示したように SML はそれなりに強かつ柔軟な表現能力を持っているが、その独特の記号的性格がなじみにくいことも確かであろう。そこで SML の一部に日本語を併用することとを考えてみた。自然言語によるデータベースへのアクセスの試みは英語を用いたものがいくつかあり<sup>3)</sup>、日本語ではヤチマタ<sup>2)</sup> などがある。これらの試みに対して、われわれの方法の特徴は、SML の骨格中に日本語を組込んだこと、さらに逆に日本語表現中に SML で表現した変数を組込むようにして、SML と日本語の長所をともに生かそうとしたことにある。

その詳細は別報に譲るが、一例を示しておこう。これは 5.1 の F1 を使い、しかも F1 = ジンコウ, S2 = ケン という同義語が与えてあるものとする。

**例 7** 1975 年の男の人口が、1965 年の東京の女の人口以上の県とその個数

```

LIST A,B;
A= 1975 ノ オトコ ノ シ"ンコウ カ" C イ"ョウ ノ ケン;
B= COUNT(A);
C= 1965 ノ トウキョウ ノ オナ ノ シ"ンコウ;

```



## 6. 実働化と使用実験

SCLAMS の試作システムは、PL/I (Fコンパイラ) とアセンブラを用いて、IBM の S/370 モデル 138 に対して実働化した。

現在は 512 K バイトのメモリ領域で使用しているが、そのうち約半分は作業域である。処理の高速化のために、操作モードで用いる S.L. の全データは一旦この作業域に移して使用している。それを外部メモリに回せば、必要は内部メモリ容量はかき減らすはずである。

各種の使用実験はいしは使用経験はこれから種々で行く予定であるが、当初目標にした操作モードでの処理の高速化はほぼ満足されており、データ点数が数千程度の S.L. に対する各種応答はほぼ瞬時である。

検索処理時間の短縮は、データ蓄積の手間の犠牲の結果得られたものと言えるが、蓄積モードでの処理時間については次のデータを持っている。

130 レコードのファイルから、1 scale の S.L. を 4 個、2 scales の S.L. を 3 個分解して蓄積するのに約 2 分 (経過時間) ; 同じく

5,342 レコードのファイルから、2 scales の S.L. を 3 個取り出すのに約 4 分。

## 7. むすび

検索時間優先の小規模データベース・システム SCLAMS についてその概要を説明した。システムに蓄積されているデータに関する「小さな塊」と柔軟に構成でき、またその答を迅速に与えることができるように、ファイルよりも小さい S.L. を処理の基本単位としたことが一つの特徴である。ただし、ファイルを S.L. に分解することによって、データ間のある種の関係を失う面もあり、それらの利害得失を今後の使用経験の蓄積によって評価する必要がある。

最後に、実働化に当って各種の助言を頂いた当所の普及主任研究員に感謝する。

### 参考文献

- [1] C. J. Date: An Introduction to Database Systems, Addison-Wesley (1977).
- [2] 藤崎, 岡下, 諸橋, 渋谷, 鷹尾: データベース照会システム「ヤチマタ」と名詞句データ模型, 情報処理, Vol. 20, No. 1 (1979), 77-84.
- [3] たとえば G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz & J. Slocum: Natural Language Interface to Complex Data, ACM Trans. Database Systems, Vol. 3, No. 2 (1978), 105-147 とその引用文献など。