

非定型的データ共有のためのデータベース論理構造とユーザビュー

高橋 淳一

日本アイ・ビー・エム株式会社
東京基礎研究所

人文科学など研究を目的としたデータベース応用では、一般利用者が生み出すデータ資源を利用者自らの手で管理できると同時に、共用のデータとして再利用できることが望まれている。このようなデータ資源は非定型的で多種多様な情報をもつため、共有化がなかなか達成されないという問題がある。本稿ではまず、国立民族学博物館における民族学標本資料の例をとりあげ、データ資源の退蔵の問題点を考察する。そして、このようなデータ資源はデータベース構築後も継続して収集され、その意味構造が利用されていく過程で明らかになることに注目して、その共有化をはかるためのデータ格納構造とそのユーザビューを提案し、その一実現法を示した。

A Logical Database Structure and User Views for Sharing Deposited Information

Junichi Takahashi

IBM Research, Tokyo Research Laboratory
5-19, Sanbancho, Chiyoda-ku, Tokyo 102, JAPAN

In many database applications where end users store and retrieve data for their individual professional research purposes, they want not only to manage data by themselves in their personal environments, but also to share the data among other users. End users are unclear what they can seek for from new information resources that have various meanings in an application area, and do not always fit into underlying database schema, which makes retrieval tremendously difficult. In this paper, the ethnographic sample database in National Museum of Ethnology is examined to show how many deposited information resources are found in the research field of ethnology. Based on the fact that information resources continue to grow and their meanings become clear in daily use, we propose a logical database structure and relational views for it to share deposited information. An implementation technique for user views is also described.

はじめに

データベース応用が、一般利用者の知的生産の場に広がりつつある。関係データベースが、定型データを対象とした事務処理分野におけるソフトウェア開発ツールとしての一解決であったように、一般利用者の創造活動のなかで非定型的に発生するデータを自らの手で管理できるツールとしての役割もよりいっそう重要になってきている。知的生産の場におけるデータベース利用者は、ひとりひとりが専門的知識を前提にして共有データの分類・整理・保存作業をおこない、新たなデータ資源を作り出している。その結果として、個人の解釈で注釈などの情報を付加したり、場合によっては共用の源データに手を加えたりすることが頻繁に起こる。このように発生したデータ資源もまた共用のデータとして再利用が望まれている[杉田87]。

しかし、非定型的に発生するデータ資源は各個人の主観に左右されやすく、また情報の精度が一律でないことが多い。そのため、せっかくの有用なデータ資源も退蔵したまま、事務処理分野の定型的なデータと比較すると、多目的な共用利用がなかなか達成されない。たとえば医療分野の症例検索では、カルテの所見情報は重要な項目であり、その記述がないものはほとんど参照されない。それにもかかわらず標準的な記法がない。画像データベースや文書データベースなどにおいても、共用のデータ資源が豊富になるにしたがって、それらの解釈も利用目的もまた多様になってくるので、データベース化するのには難しくなってくる。創造の成果はあくまで個人の資産というならば個人用のデータベースとして管理できれば十分であるが、データベースを情報伝達の手段とするならば、その成果が作成した本人以外も利用できる環境が提供されなければならない。

本稿ではまず、国立民族学博物館における民族学標本資料の例をとりあげ、データ資源の退蔵の問題点を考察する。そして、それを単にデータモデルやデータベース設計手法だけの問題ととらえず、非定型的に発生するデータの格納と利用のための利用者インタフェースの問題として検討してみる。その上で、退蔵データ資源の共有化・再利用をはかるための柔軟な格納構造とそのユーザビューを提案し、その一実現法を示す。

なお、本稿は特定の応用におけるデータモデルの提案や知識の構築を目的とするものではないので、データベースとして関係データベースを想定し、一般利用者は標準的なSQLが使えることを前提としている。

民族学研究のためのデータベース事例

一般利用者による知的生産の例として、国立民族学博物館(民博)所有の民族学標本資料の一部について、その民族学研究における利用形態の事例をみてみる[民博85]。標本資料とは、籠や仮面など、民族学者が各地で収集した標本実体そのものに関して記述された付加情報である。ここではそのうち、文字・数値情報を対象とする。

標本資料の検索は研究活動に不可欠な作業であり、検索結果をもとに収蔵庫から捜しだされた標本実体そのものを、眺めたり比較したりして、各個人が分類・整理し、その結果を保管する。その結果は、各個人の研究成果にとどまらず、他の研究者の研究のためにも有益な情報となる。すなわち、共用の標本資料から新たなデータ資源を生み出すという創造活動が行なわれている。

データ資源としての情報カード

民博では、標本実体に関する情報を標準的な情報カードに記入して管理している。情報カードには、標本番号・標本名・収集地など、標本実体に関して民族学研究に必要な属性項目が約30項目にわたって設けられ、標本収集とともに研究者の手で記入される。

情報カードには、たとえば『籠』に関する標本の標本名として、次のようなものが記入されている。

籠・小物入れかご・小籠・米入れかご・丸籠・飯籠・鳥籠・弁当籠・鬮鶏用鶏入れ籠・儀礼用かご・魚入籠・魚伏せかご・魚釣り用魚籠・魚籠・籠(魚とり用)・籠(漁業用)・籠(漁具)・背負籠・籠(背負用)・肩掛けかご・手さげ籠・籠(芋運搬用、把手付)・籠(ふた付)・籠(背負運搬用)・籠(手さげ用)・籠(手付)・籠(手付・竹製)・籠(手提用)・籠(竹製)・籠(竹製、弁当入れ)・籠(円形、竹製、弁当入れ)・籠(円形、浅型)・籠(円形蓋つき、浅型)・籠(丸形、弁当用)・籠(調理用)・籠(食物入れ)・籠(小型)・籠(弁当用)・籠(樹脂塗)・籠(畜産用)

このように、単に『標本名』という項目をとっていても様々な情報が記入されていることがわかる。たとえば、用途や製作に関する属性項目として、

使用目的 弁当用・漁業用・小物入れなど

使用法	手・肩・背など
使用者	女性・ラマ僧など
製作材料	竹など
付属部品	把手付き・ふた付きなど
形状	円形・浅型・小型
標本の状態	本体のみ・部分品・不完全品など

が抽出されてくる。ところが一方で、こうして抽出される属性項目のいくつかは、すでに情報カードに設定されている項目と類似しているものもある。情報カードには『使用法』に関する項目があって、たとえば、

『家で飼育している鶏をいれるために用いる。鶏を他所に移すばあい、このかごの中にその鶏をいれ、ヒモの部分を手でもって運ぶ。』

といった記述がされている。情報カードにおける『使用法』は文章記述であり、しかも、使用の目的などについても記述されている。情報カードは民博において一応は標準化されているが、その記入法はひとりひとり異なった分類の観点をもつ研究者各個人の興味と整理法に依存している。また、同じ記入者でも時間の経過によって観点が変化していくこともある。

データ資源の退蔵

情報カードをもとに構築された標本資料のデータベースを検索する際の問題を考えてみよう。関係データベースを想定し、標本実体に対応する関係表には、標本番号・標本名・使用法などが属性として含まれているとする。

いま、『手で提げて使うかご』という検索要求を考えてみる。この検索は、SQLによる問合せで次のように書ける。

```
SELECT 標本番号, 標本名
FROM 標本
WHERE 標本名 = '籠'
AND 使用法 = '手提'
```

この問合せでは、標本データベースにおいて標本名が『籠』で、使用法が『手提』の標本は確かに検索できる。しかし、問合せの結果が検索要求を満足するとは限らない。標本の収蔵庫に足を運んで標本そのものの中から手提籠をいくつか捜してみると、

データベースから検索されなかったものがいくつも出てくる。実際に手で提げて使う籠を検索するには、たとえば次のような複雑な条件を付けて問合せる必要があることも考えられる。

```
SELECT 標本番号, 標本名
FROM 標本
WHERE 標本名 IN ('籠', '手提籠')
AND ( 使用法 IN ('手', '手提') OR
      部品 IN ('把手', 'ひも') OR
      製作材料 = '手付竹製' )
```

ここでは、『手提』という標本の特徴が人によって標本名・使用法・部品・材料などの異なる認識のもとで観測され、そのままの観点で記述されてしまっている。いくら標本カードが標準的（おそらくここでの項目がデータベーススキーマに採用される）だからと言っても、記入時にどの項目に何をどう記入するかまで規定できない。これをデータとして格納しようとしても、共用のデータベースとしては次のような点で問題となってくる。

● データ収集の不完全性

『手提』という概念がいつも認識されているわけではない。標本名が単に『籠』としかなく、使用法が『手提』ではない標本のなかにも手で提げて使う籠が存在する。

● データ収集の不均一性

『使用法』が『手提』という概念の唯一認識のしかたではない。『把手付き』の『籠』のなかにも手提籠が存在する（使用形態・構造・特徴・形など人によって認識のしかたが異なる）。

● 属性名の不統一性

『手提』という概念が『使用法』として認識されていても、これを標準的な属性名で参照できるとは限らない。

● 属性値の不統一性

同じ手提籠の標本でも、標本名が『籠』に統一されていないし、使用法が『手提』に統一されていない。

検索者はそれぞれ検索者の観点をもっている。使用法という観点で手提籠を検索しようとしている利用者が、部品や材料といった異なる観点があることをどのようにして知ることができるのか？ データベース化はしたがだれも参照しない、あるいはできないならば、データ資源は退蔵するだけである。

非定型データ共有の問題点

データ資源退蔵の問題は、基本的にはデータベース構築時に、十分な領域知識をもった設計者が、いかによりデータモデルの上でよい設計をするかにある。しかしそれと共に、新たな観点で得られたデータ資源を、どのようにデータベース構造に反映させてデータを投入するのか、それが既存の構造にどう影響してくるのか、といった問題を解決する必要がある。これらを一般利用者に開放しようとするならば、データの格納構造を含めて利用者インタフェースを検討しなければならない。

一般に情報内容を概念的に把握するためには、実体に対して属性という観点をいくつか設定し、データが何を意味するかを明確にする。関係データベース設計においては、その属性値に関する制約や属性間の従属性を明らかにしてデータベース構造を正規化し、更新操作の波及が局所的になるような望ましい形にすることを目的とする。この過程では、次のようなデータベース設計の難しさが指摘される。

- どこまで詳細に対象世界をモデル化するか
- 実体・関連・属性の識別は唯一のものか
- 属性相互間の影響が容易に検知できるか
- 属性名・属性型・値の単位・記法などに関して標準化できるか

このような問題は、『汎関係スキーマ』や『実体型・関連型』といった設計における仮定に起因する[南雲87]。ERモデルで設計すると、名付けようのない不自然な関連関係がいくつも必要になってくるのはその例である。

対象が非定型的に発生するデータ資源の場合、上に指摘される設計の難しさに加え、さらに次のような特徴を考慮する必要がある。

- データ収集が、データベース運用期に入っても継続的におこなわれ、データベース設計時には対象とならなかったデータが発生する。
- 初めからデータ項目が定まっているわけではなく、データの意味構造がデータベース構築後運用されていく過程で明らかになっていく。
- 収集された時点では、データの情報の精度に差異がある。

データベースは試行錯誤的に設計・改善され、スキーマは頻繁に成長に推移していくことが予想される。関係データベースでは対象世界の記述と個々の検索要求を切り離して考えているため、上記の問題に関して、データ収集者と設計者および検索者との間で何らかの形で同意が得られている必要がある。しかし、このような同意には長い時間をかけた要求分析を要し、検索利用者が満足するようなデータベースシステムはなかなか構築・稼働できないことになる。かりに運用されても検索時における検索者の不満は解消されない。もちろん基本的な設計作業は重要だが、新たなデータ資源発生のために設計作業を繰り返してデータベースを再編成していくのは、非現実的である。

注釈などのように、最初はテキストとして記述されるものは、そのまま長大フィールドとして格納し、検索は文字列マッチング機能を用いるというのが現状である。しかし、その内容にまで立ち入った検索要求に答えるには、さらにテキストとともにキーワードなどによるテキストの内容記述が必要である。情報検索ではキーワード検索は一般的だが、単なるキーワード付けでは検索利用法が定型的なものに限られてしまう上、適切な検索をするにはかなりの熟練を要する[神尾88]。また、キーワードの選定やシソーラスの整備が十分でないといえずしも一般利用者が使いやすいインタフェースと言えない [秋山88][民博85]。

スキーマが頻繁に成長し、しかもそれが常に望ましい状態に保たれているという保証がないならば、対象領域におけるシソーラスの利用など、情報精度の差異を吸収するような検索インタフェースだけでは十分ではない。一般利用者が自由度をもって情報記述ができるような、柔軟なデータ格納構造とその検索のための利用者インタフェースが是非とも必要になってくる。

格納構造とユーザビュー

非定型的データはデータベース構築後も継続して収集され、その意味構造が検索利用されていく過程で明らかにされることに注目すると、共有化のためには、次のようなデータの格納構造が要求される。

- 属性名もデータとして扱え、データ型を強要しないなど、自由度をもった情報の記述ができる。
- 意味構造が十分に検討されない間は、既存のスキーマに影響を与えないように格納でき、将来の再構成のための緩衝的役割を果たす。

本稿では上記の要件にもとづき、データを2項関係の基本要素に分解して管理する『保管表』という格納構造と関係表との併用を提案し、その利用法を示す。

保管表構造

基底表 (Base Table) は任意の関係表 (ビューを含む) であり、タプルを一意に識別するタプル識別子 (キーまたはサロゲート) をもつとする。各基底表に対して、〈タプル識別子・属性名・値〉の3組の集合によって定義される論理構造を、その基底表に対する保管表 (Depository) と呼ぶことにする。保管表は3つのフィールドからなる関係表の一種である。タプル識別子はその保管表に対応する基底表のキーである。保管表の各タプルは、同じキーをもつ基底表のあるタプルの属性と属性値を表している。

保管表は値フィールドに対する型をもつ。また、名前を付けることによって各基底表に対して複数の保管表が定義できる。ただし、型および名前はいずれも省略でき、その場合は省略値がとられる。保管表に格納された属性名は、関係表と同じく局所的に扱われる。保管表は関係表の定義のオプションとして定義する。標本番号と標本名を属性にもつ関係表標本と注釈に関する保管表の例を図-1に示す。

TABLE SAMPLE		DEPOSITORY COMMENT	
CNO	CNAME	CNO	FIELD VALUE
1	BASKET	1	USE SHOULDER
2	BASKET	1	PARTS STRING
3	BASKET	2	USE CARRIAGE
4	MASK	2	PARTS GRIP
		3	USE HAND

図-1. 基底表と保管表の例

図-1の保管表は、次のようにデータ定義する。

```
CREATE TABLE SAMPLE(
CNO SMALLINT PRIMARY KEY,
CNAME VARCHAR(20)) WITH
DEPOSITORY COMMENT(VARCHAR(20));
```

保管表のユーザビュー

基底表では実体の全属性を一タプルで表すが、保管表では実体の一つの属性を一タプルで表わしてい

る。いま、保管表からそれと等価な意味を持つ基底表への写像を転置といい、その像を転置表と呼ぶ。これはいわゆる値/属性変換[Kent82]である。なお、基底表のスキーマをこの転置表のスキーマに変更し、データを格納することを保管表の格上げと言う。転置表と対応する基底表をキーによって結合すれば、ユーザビューとして全属性を一タプルで表せる。ただし、ここでの結合はアウトジョイン [Date 83]である。このビューを混成ビュー (Hybrid View) と呼び、基底表と保管表の名前を記号 '+' で連結して書くことにする。たとえば、上記の例で関係表標本を参照するには *SAMPLE + COMMENT* または *SAMPLE +* とする。図-2に混成ビューの例を示す。

VIEW SAMPLE+COMMENT				
CNO	CNAME	USE	PARTS	USAGE
1	BASKET	SHOULDER	STRING	?
2	BASKET	CARRIAGE	GRIP	?
3	BASKET	?	?	HAND
4	MASK	?	?	?

図-2. 混成ビューの例

いま関係 R のキーを K_R とし、その保管表を R_X 、転置表を R_{XT} とする。また、 R と S の左アウトジョインを $R\{S$ で表すと、 $R +$ は、

$$R\{R.K_R = R_{XT}.K_{R_{XT}}\}R_{XT}$$

である。簡単のため、以後これを $R\{S\}R_{XT}$ と略記する (右アウトジョインも同様)。なお、SQL2 標準[ISO]ではアウトジョインがサポートされているので、 $R +$ は次のように記述できる。

```
SELECT * FROM R LEFT JOIN R_{XT}
ON R . K_R = R_{XT} . K_{R_{XT}}
```

保管表のような二項関係による情報表現は、PILE や LEAP 構造まで遡ることができる[有澤80]。[Motro 84]では、同様の構造にもとづいたデータベース構造とその検索機能を述べている。また、[牧之内85]では、オブジェクト指向データベースにおけるオブジェクトの格納法として、また[池田88]では、マルチメディアの格納法にそれぞれ適しているとして採用している。これらの研究と異なり、本稿における保管表は、基底表とともに利用者になじみやすい関係表としてビュー提示するところに特徴がある。

更新操作の意味

結合を含むビューは、SPJ 標準型(選択・射影・結合をこの順番で行い、射影によって結合属性が取り除かれない)ならばビューを通しての更新が可能である[Keller85]。混成ビューは、基底表と対応する保管表の転置表のキーによる結合で得られるから、この結果を用いれば、混成ビューに対する更新操作はそれぞれの表に対する更新操作に置換えることができる。しかし、転置表は問合せ時に動的に生成されるもので、さらに転置表に対するデータ操作を保管表に対するデータ操作に置換える必要がある。ここでは、保管表に対するデータ操作の意味について述べる。

1. 挿入操作

混成ビューに挿入するタプルで指定された属性名が基底表にないものについては、そのキーおよび属性名と属性値の3組として保管表へ格納する。一般には、混成ビューに対する挿入操作は、基底表への挿入操作と保管表への挿入操作になる。

2. 削除操作

保管表に対しては、基底表で削除されるタプルのキーをもつタプルはすべて削除対象となる。一般には、混成ビューに対する削除操作は、基底表の削除操作と保管表の削除操作になる。

3. 変更操作

混成ビューに対する変更操作は、基底表に対しては変更操作となるが、保管表に対しては、各属性についてその変更値が空値の場合は削除、空値以外でその属性名がすでに保管表に格納されている場合は変更、それ以外は挿入の各操作になる。

実現法

問合せの変換

混成ビューに対する問合せは、アウタージョインをサポートする SQL2 の構文の問合せに変換することで実現できる。ここでは、後述のように転置表生成時にいくつかの最適化が考えられるため、SQL への問合せ変換を示す。

混成ビュー $R+$ は、基底表 R と保管表 RX の転置表 RXT とのアウタージョインで得られる。いま、保管表における空タプル

$$RX_0 = \{ \langle NULL, NULL, NULL \rangle \}$$

を考える。TRANS を転置プログラム(後述)による転置オペレータとして、

$$RXT_0 = TRANS(RX \cup RX_0) - RXT$$

すると、 RXT_0 は RX_0 の転置表で、その全ての属性が空値である1タプルだけからなる関係となる。 $D(R)$ を R の全属性集合とし、関係 R の属性 A 上の射影を $R[A]$ で表すと、

$$R_1 = R[.]RXT[D(R)], \quad R_2 = R - R_1$$

とすれば、

$$R+ = R_1[.]RXT \cup R_2[NOT.]RXT_0 \quad (*)$$

ただし、 $[NOT.]$ は $[.]$ の否定である。これは、SQL で次のように記述できる[Date83]。

```
SELECT * FROM R,RXT WHERE KR=KRXT
UNION
SELECT R.* RXT0 .* FROM R,RXT, RXT0
WHERE NOT EXISTS
  (SELECT * FROM R WHERE KR=KRXT )
```

保管表の性質

混成ビューに対する SPJ クラスの問合せを、(*)式にもとづいて保管表への問合せに変換する際のいくつかの性質を示す。紙面の都合上、それぞれについての証明は略証にとどめる。

選択

$P\{A\}$ を属性 $A \in D(R)$ を含む述語とし、関係 R の P による選択を $R[P]$ で表す。 $A_i \in D(R)$, $A_j \in D(RXT)$, $P\{A_i, A_j\} = P_1\{A_i\} \text{ AND } P_2\{A_j\}$ とすると、

$$\begin{aligned} R+ [P\{A_i, A_j\}] &= (R_1[.]RXT)[P_1\{A_i\} \text{ AND } P_2\{A_j\}] \cup \\ &\quad (R_2[NOT.]RXT_0)(P_1\{A_i\} \text{ AND } P_2\{A_j\}) \\ &= (R_1[P_1\{A_i\}][.]RXT[P_2\{A_j\}]) \cup \\ &\quad (R_2[P_1\{A_i\}][NOT.](RXT_0[P_2\{A_j\}])) \\ &= (R[P_1\{A_i\}][.]RXT[P_2\{A_j\}]) \end{aligned}$$

例: c が定数で、 $A \in D(RXT)$ のとき、

$$R+ [A = c] = R[.]RXT[A = c]$$

射影

$A_i, A_k \in D(R), A_j, A_l \in D(RXT)$ とすると,

$$\begin{aligned} R + [A_i, A_j] \\ &= (R_1[.]RXT)[A_i, A_j] \cup (R_2[NOT.]RXT_0)[A_i, A_j] \\ &= (R_1[A_i][.]RXT)[A_j] \cup (R_2[A_i][NOT.]RXT_0)[A_j] \\ &= (R[A_i][.])(RXT[A_j]) \end{aligned}$$

さらに、選択と射影を組合わせて、

$$\begin{aligned} R + [P\{A_j, A_k\}][A_i, A_j] \\ &= R[P_1\{A_j\}][A_i][.](RXT[P_2\{A_k\}][A_j]) \end{aligned}$$

射影される属性が選択の条件に含まれないときは、射影と選択が交換可能で、

$$(RXT[P_2\{A_k\}][A_j]) = (RXT[A_j])[P_2\{A_k\}]$$

ただし一般的には、射影される属性以外でも選択の条件に含まれるので、

$$(RXT[P_2\{A_k\}][A_j]) = (RXT[A_j, A_k])[P_2\{A_k\}]$$

例: $A \in D(R)$ のとき、

$$\begin{aligned} R + [A_1 = c][A, A_2, A_3] \\ &= (R[.])RXT[A, A_1, A_2, A_3][A_1 = c][A, A_2, A_3] \end{aligned}$$

結合

A, B をそれぞれ関係 R と S の属性とし、 R と S の A, B に関する θ 結合を $R[A\theta B]S$ で表す。 $A \in D(RXT), B \in D(SXT)$ のとき、

$$\begin{aligned} R + [A\theta B]S + \\ &= (R[.]RXT)[A\theta B](SXT[.]S) \\ &= R[.](RXT[A\theta B]SXT)[.]S \end{aligned}$$

例: 関係 R, S はそれぞれ A, B および C, D を属性にもち、

$$\begin{aligned} R &= \{ \langle a_1, b_1 \rangle, \langle a_2, b_2 \rangle, \langle a_3, b_3 \rangle \} \\ S &= \{ \langle c_1, d_1 \rangle, \langle c_2, d_2 \rangle, \langle c_3, d_3 \rangle \} \end{aligned}$$

とする。 $RX(A, FIELD, VALUE)$ が、

$$\begin{aligned} \langle a_1, X, x_1 \rangle, \langle a_1, Y, y_1 \rangle, \langle a_2, X, x_1 \rangle, \\ \langle a_3, X, x_2 \rangle, \langle a_3, Y, y_2 \rangle, \langle a_3, W, w_3 \rangle \end{aligned}$$

から成り、また $SX(C, FIELD, VALUE)$ が、

$$\begin{aligned} \langle c_1, U, u_1 \rangle, \langle c_2, U, u_2 \rangle, \langle c_2, Y, y_1 \rangle, \\ \langle c_3, U, u_2 \rangle, \langle c_3, Y, y_2 \rangle \end{aligned}$$

から成るとき、

$$\begin{aligned} R + [R + .Y = S + .Y] S + \\ [X = x_1 \text{ AND } U = u_2][A, B, C, D, Y, W] \\ &= R[.] \\ &\quad RXT[X = x_1][A, Y] [RXT.Y = SXT.Y] \\ &\quad SXT[U = u_2][C, Y, W] [.]S \\ &= \{ \langle a_1, b_1, c_2, d_2, y_1, \text{NULL} \rangle \} \end{aligned}$$

転置操作の実現法

保管表の転置は SQL では求まらないので、ホスト言語による特別なプログラムが必要になる。ここでは、動的 SQL [IBM]による実現を示す。

関係 R のキーを K_R 、問合せに含まれる属性で、 R に含まれず、かつキー以外の属性を A_1, \dots, A_n とする。値/属性変換プログラム TRANS は、保管表 $RX(K_R, FIELD, VALUE)$ を入力とし、転置表 RXT を出力する。まず以下のような SQL 文を文字列として生成して、これを動的に実行する。このなかで、FETCH 文および UPDATE 文は保管表の各タプルについて繰り返し実行する。

```
CREATE TABLE TRX( KR, A1, ..., An )
INSERT INTO TRX(KR)
  SELECT DISTINCT KR FROM RX
DECLARE C CURSOR FOR
  SELECT * FROM RX
  WHERE FIELD IN ( A1, ..., An )
FETCH C INTO $KR, $FIELD, $VALUE
UPDATE RXT SET $FIELD=$VALUE
  WHERE KR=$KR
```

この転置操作は、問合せのたびに実行する必要があるが、SPJ クラスの問合せに応じて、上で示した保管表の性質を使えば最適化が可能である。それについては別の機会に報告したい。

おわりに

本稿では、民博おける民族学標本データを例にとり、非定型的に発生するデータ資源の退蔵の問題点を検討した。民博では、IBM との共同研究で構築された標本画像検索システム[佐藤88]がすでに稼働中であるが、キーワードや注釈など、従来のデータベースでは退蔵されがちなデータ資源をより積極的に活

用できれば、設計作業に利用者の声を反映させ、利用者の検索要求の明確化と満足度の向上が期待できると思われる。

そのための第一歩として、本稿では退蔵データ資源の共有化をはかるための格納構造として保管表とそのユーザビューを提案した。保管表は、基底表とともに関係表としてビュー提示されるため、利用者インタフェースとしては SQL の構文がそのまま利用できるという利点がある。

現在、問合せ変換と最適化の一部についてはインプリメントを終え、保管表の有効を評価するための試験システムを作成中である。保管表の性質や更新操作に対する問合せ変換とその実現法などについてはさらに検討が必要で、今後の課題である。

謝辞

民族学標本のデータを参照させていただいた国立民族学博物館に感謝します。

文献

- [Date 83] Date, C.J.: The Outer Join. Proc. 2nd Int. Conf. on Databases (1983).
- [IBM] SQL/Data System Application Programming for VM/System Product. IBM Corporation (1984).
- [ISO] Database Language SQL2: Working Draft. ISO TC97 SC21 N382 (1987).
- [Keller 85] Keller, A.M.: Updating Relational Databases Through Views. Ph.D. Dissertation, Stanford University (1985).
- [Kent 82] Kent, W.: Choices in Practical Data Design. Proc. of the 8th Int. Conf. on VLDB (1982) pp. 165-180.
- [Motro 84] Motro, A.: Browsing in a Loosely Structured Databases. Proc. of ACM SIGMOD (1984) pp. 197-207.
- [秋山 88] 秋山 幸司: テキスト情報の知的検索における諸問題. 情報処理学会第 64 回データベース・システム研究会 (1988).
- [有澤 80] 有澤 博: データベース理論. 情報処理学会 (1980).
- [池田 88] 池田 秀人: 基本 5 列表によるマルチメディアデータ管理. 情報処理学会第 66 回データベースシステム研究会 (1988).
- [神尾 88] 神尾 達夫: 新聞記事データベースの最新動向. 情報処理学会第 18 回情報システム研究会 (1988).
- [佐藤 88] 佐藤 真知子 他: 民族学研究のための標本画像検索システム. 情報処理学会論文誌 (投稿中).
- [杉田 87] 杉田 繁治: 人文科学におけるマルチメディアデータベース. 情報処理 28(6) (1987).
- [南雲 87] 南雲 道明: 抽象概念を解消したデータベース設計方法論. 情報処理学会第 62 回データベースシステム研究会 (1987).
- [牧之内 85] 牧之内 顕文 他: マルチメディアデータベースシステムのアーキテクチャ. 情報処理学会第 50 回データベースシステム研究会 (1985).
- [民博 85] 国立民族学博物館十年史. 国立民族学博物館 (1985).