

# データベース管理システムを用いた 万葉集データベースの一構成法

中田 充    吉村 誠    葛 崎偉  
山口大学   教育学部

本稿は、リレーショナルデータベース管理システムを用いた万葉集データベースの構築を目的として、データベーススキーマの設計とその評価について述べる。万葉集には多くの写本、注釈書が存在し、和歌本文以外にも和歌の題詞、左注、異訓、異同等の様々な情報がある。これらも含めて和歌を検索可能とするシステムが求められている。本システムのスキーマ設計の方針は、(1) 全ての本のデータを同一形式で格納する、(2) 和歌を和歌本文、題詞、左注に分けて格納し、さらに和歌本文は句ごとに格納する、(3) これら以外の情報は部分注と全体注にまとめて格納する、という3点である。評価は作成したデータベースにデータを挿入し、検索時間を計測することで行う。

## A Design of MANYO-SHU Database System Based on Database Management System

Mitsuru NAKATA,    Makoto YOSHIMURA,    Qi-Wei Ge  
Faculty of Education   Yamaguchi University

This paper is to propose a database schema and further to evaluate its performance in order to develop a practically useful relational database for Manyo-Shu, Japanese poems. In Manyo-Shu, there are lots of books of transcripts and annotated editions, and in addition to the texts of the Japanese poems there is also various information about titles, notes written in the left of the poems, different Japanese readings for the same phrases, different Chinese characters and so on. A database including such information is strongly expected in the research field of Manyo-Shu. In this paper, we design a relational database that satisfies the followings: (1) all the data of the books are stored in a same form; (2) all the poems are stored by dividing them into texts, titles and the notes in the left, and further the texts are stored by dividing into phrases; (3) all the other information about partial and total notes is stored together. Further we evaluate the performance of this database by carrying out some practical searches and measuring the search times.

## 1 はじめに

万葉集は、8世紀後半から9世紀初頭に編纂されたと考えられる全20巻からなる古代の和歌集である。4516首の歌が載せられており、皇族や貴族を中心としながら庶民の歌までも含むという、後の勅撰和歌集には見られない特徴を持っている。しかし、他の多くの古典作品と同様、原本は失われており、書写された写本という形でのみ伝えられている。そのため文字の誤写等による異本が生じており、約20種類の写本が現存している。さらに、複数の写本から原本を推定する校訂作業を経て作成された校定本、漢字表記の原文に対して後世に付された訓読をもとにした注釈書などが併せて30種類ばかり作られている。これらには、約3400箇所にあたる文字の相違(異同)と3500首あまりにあたる訓読の違い(異訓)が存在する(以降、本稿では写本、校訂本、注釈書などの様々な種類の本をまとめて諸本と呼ぶ)。

現代の国文学における万葉集研究は、文献学に基づいた校訂作業と、国語学、文学的手法を駆使した訓の特定を行うところから始まる。しかし、これも一つの原本復元の提案に過ぎず、結局は同じ和歌でも文字や読みなどが複数存在することになる。さらに、校訂本においては、文字の異同の箇所を一つの文字に特定したとしても、校訂の根拠を示すために他の異なる文字を示す校異が掲載されている。そして、注釈書においては、本文の原文、平仮名の他に、漢字にはルビを付した現代表記による読み下しが示される。国文学研究においては、万葉集の和歌に対して、こうした文字や訓の異同を含んだ的確な情報収集が要求される。

万葉集は歌集であるので、5音または7音からなる句を基本単位として、5句からなる短歌、複数の長句からなる長歌によって構成される。これに加え、和歌の右側に詠作の場や主題、作者などを記した題詞が、和歌の左側に題詞と同様のことを補足的に説明した左注が漢文体で任意に付けられている。また、個々の和歌は、国歌大観番号と称する明治期に付された和歌番号によって整理されており、研究者はこれを用いてそれぞれの和歌を識別している。しかし、題詞、

左注には和歌番号がない。さらに異訓、異同、ルビ、校異などの重要な情報が存在するが、これらを容易に検索可能とするために、コンピュータを利用した検索システムが強く望まれている。なお、電子化にあたっては、特に写本に多い本文の修正(ミセケチ)や、挿入、補注などの付随情報も扱える機構が望まれている。これは、文献学的に万葉集の研究を行う上で、写本間で従来にない詳細な関連情報をとらえることが可能となるためである。

しかしながら、国文学研究におけるコンピュータ利用の研究はなされているものの[1]、その文学的方法論が十分に確立していないために普及が遅れている。現在のところプロジェクトや個人で作成された古典作品のテキストデータがインターネットで供給されはじめた段階であり、検索システムは万葉集以外の古典文学作品全体を対象を広げても多く存在しない。近年になって国文学研究資料館を中心とする原本データベースシステム構築のプロジェクト[2]、[3]や、角川書店の国歌大観[4]、岩波書店の八代集[5]などの電子化テキストに検索システムが付録として供給され始めている。

万葉集の検索システムとしては、吉村が作成した和歌のテキストファイルとそれに対する検索プログラムからなるシステム[6]と、その和歌データをデータベース管理システム(DBMS)によりデータベース化したシステム[7]が主なものであろう。吉村のシステムは万葉集を原文、訓読、仮名ごとにテキストファイル化し、Windows上の検索プログラムを作成したものである。文字列マッチングを利用した検索システムであるため複雑な検索(詳細は後述)は出来ない。また、データは写本の一つである西本願寺本を底本(基準となる本)として作成されており、異訓、異同、注などは、その一部のみが底本に対する差分情報として格納されている。そのため、異訓、異同を含んだ形態に十分に対応したものはなっていない。[7]のシステムはLinux上のDBMSとWebサーバ+CGIを用いたシステムであるが、基本的に吉村が作成したデータを一つのリレーション(テーブル)に格納したものであり同様の問題点を持つ。

そこで、本稿では異訓、異同を含めた検索が行える万葉集検索システムの構築を目的として、リレーショナルデータベース管理システムを用いた万葉集データベースを構築する。以降、2節では万葉集検索システムに対する要求と設計方針について述べ、3節ではそれに従ったデータベーススキーマについて説明する。4節ではデータベースを実装し、サンプルデータをデータベースに格納し検索時間を計測することで評価する。5節で本研究のまとめ、および今後の課題について述べる。なお、万葉集研究では和歌は唱和した歌を示すと解釈されているが、本稿では一般的名称である和歌を用いる。

## 2 システムへの要求と設計方針

これまでに述べたものを含めて、万葉集検索システムへの要求は以下のようにまとめられる。

- a) 読み、漢字、漢字仮名交じり文による和歌の検索。
- b) 異訓、異同等を含めた和歌の検索。
- c) 題詞(あるいは左注)は複数の和歌に対応し、和歌は複数の題詞(左注)を持つ。これらを表現し、さらに、一つの和歌に対する複数の題詞(左注)の順序関係も表現する必要がある。
- d) 題詞、左注にも諸本によって異同が存在する。これを表現する必要がある。
- e) 非常に多様な注が存在するが、これらを全て管理しなければならない。注とは和歌、題詞、左注に関して、本文に表記されている以外に考えられる原文や読みなどの情報である。
- f) 注に書かれた情報を適用することで得られる原文や仮名等でも検索可能でなければならない。

本研究で目的としているシステムは、これらの要求を満たし、従来のシステムの問題点を解決するものである。そのために万葉集の和歌をテキストファイルではなく、リレーショナル型DBMSを用いてデータを管理する。DBMSを用いることで、複雑な検索機能も容易に実現可能となり、検索速度やシステムの拡張性の向上が望める。さらにその際に、単一のリレーシ

ョンに格納するのではなく、以下のような基本方針の下に複数のリレーションに分割して格納する。

- 1) 基準となる底本を定めずに、全ての本のデータを同一形式で格納する。
- 2) 和歌を和歌本文、題詞、左注に分けて格納する。
- 3) 和歌本文は句ごとに分解して格納し、検索は句を単位とする。
- 4) 和歌に付加されている様々な注は、部分注と全体注とにまとめて格納する。全体注は和歌本文や題詞、左注全体に関わる注であり、部分注は部分的な注である。

すなわち、万葉集のもつ情報を本、和歌、句、題詞、左注、全体注、部分注に分けてデータベースに格納する。1)により異訓、異同を含めた検索が行え、柔軟な応用が可能となる。2)は和歌本文が基本的に複数の句から構成されているのに対して、多くの題詞、左注は区切られていないという構造的な相違によるものである。3)は一つの本の一つの和歌でありながら様々な注などによって句ごとに原文、仮名、訓読を複数持つことが少なくないためである。また、「ある和歌のある句が他の本においてはどのようになっているか」といった検索を行うためにも必要となる。和歌の注には様々な種類があり名称も多数存在する。どれだけの種類が存在するかは、万葉集の全ての諸本の和歌を詳細に調査しなければ判明せず、注の種類ごとにリレーションを作成するといったことは現実的ではない。そこで、注は4)のように、2種類に分類して格納する。全体注と部分注を分けるのは、これらの性質の違いにより、属性数が異なるためである。

## 3 スキーマ設計

本節ではデータベースのスキーマについて述べる。まず、和歌の本文、題詞等の万葉集の構成要素とそれらの相互関連を ER モデル[8]を用いて明らかにし、その後、ER モデルをリレーショナルデータモデルに変換することで、データベーススキーマを設計する。

図1は万葉集データベースの ER モデル図である。図中の矩形は実体集合であり、「本」、「和歌」、「句」、

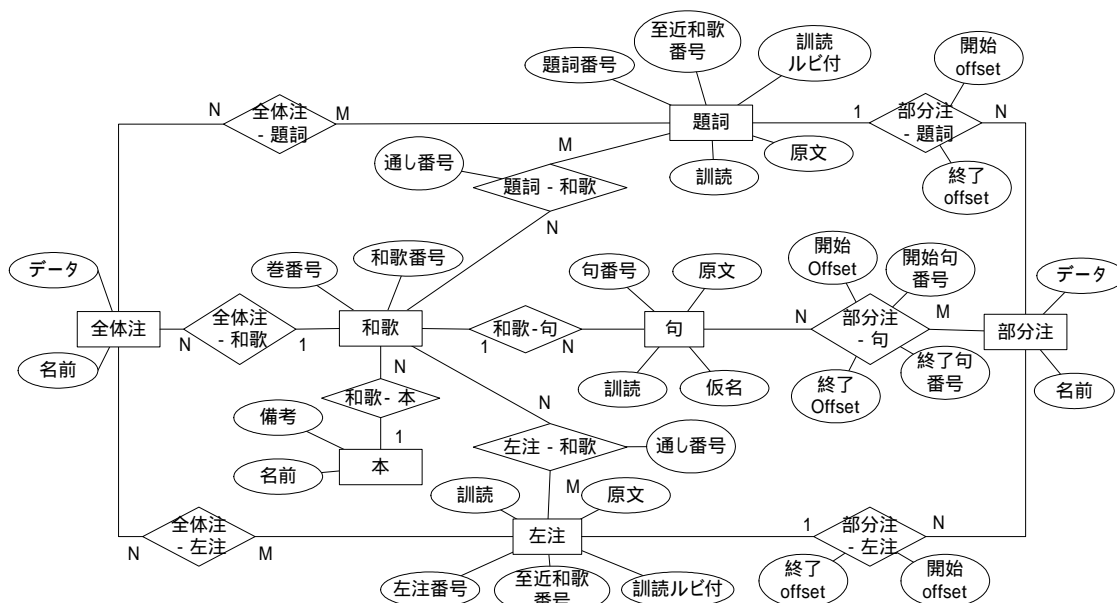


図1：ER モデル図

- (1) 本(本 id, 名前, 備考)
- (2) 和歌(和歌 id, 巻番号, 和歌番号, 本 id)
- (3) 句(句 id, 番号, 原文, かな, 訓読, 訓読ルビ付, 和歌 id)
- (4) 題詞(題詞 id, 至近和歌 id, 番号, 原文, 訓読, 訓読ルビ付)
- (5) 左注(左注 id, 至近和歌 id, 番号, 原文, 訓読, 訓読ルビ付)
- (6) 全体注(全体注 id, 至近和歌 id, 名前, データ)
- (7) 部分注(部分注 id, 至近和歌 id, 名前, データ, 開始句番号, 終了句番号, 開始 offset, 終了 offset)
- (8) 和歌-題詞関連(題詞 id, 和歌 id)
- (9) 和歌-左注関連(左注 id, 和歌 id)
- (10) 全体注-和歌関連(全体注 id, 和歌 id)
- (11) 全体注-題詞関連(全体注 id, 題詞 id)
- (12) 全体注-左注関連(全体注 id, 左注 id)
- (13) 部分注-句関連(部分注 id, 和歌 id)
- (14) 部分注-題詞関連(部分注 id, 題詞 id)
- (15) 部分注-左注関連(部分注 id, 左注 id)

図2：データベーススキーマ

「題詞」、「左注」、「全体注」、「部分注」がある。図中の菱形は関連集合であり、実体集合間にどのような関連があるかを示す。例えば、「和歌」は巻番号と和歌番号の二つの属性を持ち、「本」、「句」、「全体注」と1対Nの関連がある。ここで1対Nの関連とは、一つの和歌が複数の句や全体注を持つ、あるいは、一つの本が複数の和歌を持つといった関連である。また、「題詞」は題詞番号、原文、訓読などの属性を持ち、「和歌」とN対Mの関連がある。すなわち、一つの和歌が複数の題詞を持ち、一つの題詞は複数の和歌に掛かるといったことがあり得る。なお、各実体、関連集合の属性の詳細については省略する。

図1のERモデル図より得られたリレーションを図2

に挙げる。図中の下線は、各リレーションにて識別子(id)以外でインスタンスを一意に識別するキーとなる属性の集合である。例えば、和歌を表すリレーション(2)では、「和歌番号」と「本 id」が決まればインスタンスが一意に決定される。

(1)は本を表すリレーションであり、本の識別子「本 id」、本の「名前」、「備考」を属性として持つ。備考には本の種別、著者、書かれた年代、現代の書籍であれば出版社などの情報が含まれる。

(2)は和歌を表すリレーションである。識別子「和歌 id」、その和歌が含まれる巻の「巻番号」、国歌大観番号に基づいた「和歌番号」、和歌が属する本の「本 id」を属性として持つ。本 id を属性に含むため、同じ和歌

であっても本が異なればインスタンスが異なる。すなわち、本が40冊あれば一つの和歌に対して40個のインスタンスが作成される。これらの40個のインスタンスにおいて、「和歌番号」は同じであるが、「和歌id」は異なることに留意されたい。

(3)のリレーション句は、「句id」、和歌における句の順番を示す「番号」、漢字のみの「原文」、原文の読みを示す「かな」、漢字仮名交じりの「訓読」、可読性を向上するために訓読にルビをつけた「訓読ルビ付」、句が含まれる和歌を示す「和歌id」を持つ。「和歌id」を持つために、「和歌」と同様に同じ和歌の同じ句でも本ごとにインスタンスが作成される。さらに、「番号」と「和歌id」の属性値が両方とも同じインスタンスが存在し得る。

(4)は題詞を表すリレーションである。「至近和歌id」は、本文中における題詞の出現位置をあらわすための属性であり、「題詞の左側にある至近の和歌」のidである。同じ至近和歌idを持つ題詞は複数存在し得るが、これらは属性「番号」で区別する。「番号」は、同じ至近和歌idを持つ題詞における通し番号である。(5)は左注を表すリレーションであり、(4)と同じ属性を持つ。

(6)のリレーション全体注は、和歌、題詞、左注の全体に掛かる注を表す。「至近和歌id」は(4),(5)と同様である。「名前」には“頭注”、“脚注朱”などの全体注の名前が格納され、「データ」には全体注のデータが格納される。(7)のリレーション部分注は、和歌、題詞、左注の一部分に掛かる注を表す。全体注とほぼ同じ構造であるが、当該の部分注が、和歌のどの句にかかるかを示す「開始句番号」と「終了句番号」、句や題詞、左注の何文字目から何文字目までに掛かるかを表す「開始offset」と「終了offset」を含む点が異なる。つまり、<1, “部分注”, “3”, “仮名文字元青”, “とらしの”, “7”, “7”, “2”, “5”>は、和歌idが1の和歌の第7句目の2文字目から5文字目に関する部分注のインスタンスである。なお、題詞、左注に対する部分注は、「開始句番号」と「終了句番号」が空になる。これは、題詞、左注は句に分割されないためである。

(8)~(15)は(1)~(7)で表された和歌、句、題詞などの関連を表現するためのリレーションであり、それぞれ対応するインスタンスの識別子を要素に持つ。

これらのスキーマにより、2節で挙げた要求を満たすことが可能となる。和歌を構成する句は、属性「原文」、「仮名」、「訓読」を持つので、要求a)は満たされる。また、本毎に「和歌」のインスタンスを作成し、それに対応する「句」のインスタンスを作成するため、異訓、異同等を表現可能となっており、要求b)を満たしている。要求c)は、リレーション(4)が属性「番号」を持ち、リレーション(8),(9)で多対多の関連を表現されていることで満たされている。また、題詞と左注は「至近和歌id」を属性として持ち、諸本ごとにインスタンスが作成されるため要求d)も満たされる。要求e)は注を全体注と部分注に二分することで解決できる。

「句」を表すリレーション(3)では、「番号」と「和歌id」の属性値が両方とも同じインスタンスが存在し得るので、注を適用することによって得られる情報は新たな句として格納する。表1はリレーション(3)すなわち「句」のインスタンスの一部を示している。4行目の斜体で書かれた行は、その上の行と同じく第3句目に対応する行であるが、4行目は第3句目につけられた注(この注は本文とは異なる読みがあることを示している)を適用した結果、作成されたものである。これにより、要求f)が満たされる。

表1：注の適用によって得られる情報の格納

句番号	原文	仮名	...
1	八隅知之	やすみしし	...
2	我大王乃	わかおほきみの	...
3	朝庭	あさにはに	...
3		<i>あしたには</i>	...
4	取撫賜	とりなでたまふ	...
...	...	...	...

#### 4 システムの実装と考察・評価

本節では、3節で述べたスキーマを持つデータベースの実装とスキーマの評価について述べる。

##### 4.1 実装

システムの実装は、Compaq ProLiant ML350 server(Pentium III 600MHz, 256MB メモリ, RedHat Linux release6.1j)上で行い、DBMS には PostgreSQL Ver7.1.2 [9]を用いた。PostgreSQLのSQL文を用いて3

節で設計したリレーションに対応するテーブルを作成する。リスト1はリレーション「和歌」に対応するテーブル waka を定義する SQL 文である。ここでは、「和歌」の属性「和歌 id」は定義していないが、PostgreSQL では、テーブルのインスタンスごとにオブジェクト識別子 (oid) が自動的に付加されるので、これを和歌 id として利用する。他のリレーションに対しても同様に対応するテーブルを定義するがここでは省略する。

データベースへのデータの格納は、特定の単純な書式に沿ったテキストファイルにデータを入力し、それらをデータベースに挿入するためのプログラムを用いて行う。このプログラムの作成には、オブジェクト指向スクリプト言語 Ruby[10]を用いた。なお、現状では、データベースにデータを格納しただけであり、データの検索は PostgreSQL の psql インタプリタに対して SQL 文を直接入力する必要がある。なお、現代においては約 50 種の諸本が存在するが、本システムはそのうち 40 種類程度をデータベース化の対象とする。なお、現状は 40 種のうちの 3 つの本(西本願寺本[11]、おうふう社の万葉集[12](以降、おうふう本)、万葉集全注[13])の第 1 巻のみがデータベースに格納されている。和歌数にして 252 首(1 冊あたり 84 首)、句の数は一首が 5 句からなるとして約 1260 である。1 冊の万葉集につき約 4500 首であるので、最終的に約 18 万首、句の数は 90 万程度になる。

```
create table waka(  
  maki_no smallint not null,  
  waka_no smallint,  
  hon_id oid,  
  primary key(waka_no, hon_id)  
);
```

リスト1：和歌テーブルを定義する SQL 文

#### 4.2 考察と評価

万葉集は現存する最古の和歌集であり、その他の古典文学作品において多く引用されているため、国文学研究においては、万葉集研究者に限らず和歌を参照することが少なくない。しかしながら、従来の検索システムは、和歌の持つ様々な異訓、異同を、底本(基準となる本)を特定してそれに対する差分情報という形で保持している。このため利用者は、まず、自らが検索

したい和歌やその一部分がどの差分情報と一致しているかを知る必要があり、これが、和歌の検索を煩雑なものにしている。本システムは、底本を特定せず諸本の和歌全てのデータを句ごとに区切って格納している。また、和歌に付された注などを適用した生じる異同や異訓も新たな句として格納する。これにより、どのような訓からでも検索可能でかつ、それがどの写本や注釈書の訓であるのかといったことが容易に分かる。また国語学的な研究においても、検索対象が異訓を含んだ本文に広がることにより精度の高い研究が期待できる。しかしながら、現在の方式では全ての和歌を格納した場合に、全諸本において変化がない句や和歌、題詞などについても本の数だけデータが作成される。このことは、和歌数の増加とともに検索時間の増加という形で現れるため、データ格納効率だけではなく、検索速度の向上の点からも検討が必要である。

次に、実際にデータを挿入したデータベースに対して、ある条件についての検索を行い検索時間を測定することでデータベースを評価する。測定にあたり入力済みの 252 首では和歌数が少なすぎるため、これらをコピーしてダミーのデータを生成する。具体的には、3 冊の本の第 1 巻のデータをコピーし、西本願寺本 1、西本願寺本 2、..., 万葉集全注 1、万葉集全注 2、..., おうふう本 1、おうふう本 2、...なる仮想の本を作成する。それぞれ 750 個のコピーを作成すれば和歌の数は 189000 となり、万葉集の諸本を全てデータベース化した際の数とほぼ等しくなるが、今回は測定時間の都合上 64260 首(全体の約 1/3)の和歌でのみ検索時間を測定する。測定する検索条件は以下のとおりである。

条件(1) 原文に漢字“尔”を含む句を持つ和歌の和歌番号と本の名前を検索。

条件(2) 「仮名“やまとのくに”を含む句の原文」と「仮名“とりよるふ”を含む句の原文」の両方を含む和歌を検索。

条件(1)、(2)の検索を行う SQL の問い合わせ文をリスト 2 に示す。条件(1)の SQL 文は、正規表現を用いた文字列マッチングを含む以外は、単純な問い合わせであり、テーブル hon、waka、ku を「本 id」と「和歌 id」より

結合し、それから「原文」が“尔”という漢字を含む行の必要な属性のみを返す。句の数を  $N_k$ 、和歌の数を  $N_w$  とすると、この問い合わせの計算量は、理論上  $O(N_k N_w)$  となる。これに対して、条件(2)の SQL 文は2段の副問い合わせを行う文となっている。この問い合わせの計算量は  $O(N_k^2 N_w)$  となる。

検索時間の測定は、これらの SQL 文を DBMS に対して発行するプログラムを作成することで行う。具体的には、SQL 文の実行前と実行後の時刻を記録し、それらの差を計測する。これは、実際のデータベースの検索は作成したプログラムからの依頼を受け取った DBMS のサーバプロセスが行うため、検索プロセスのユーザ時間やシステム時間を計測することが困難であることによる。

測定結果を表 2 に示す。なお、これらはおなじ SQL 文を 5 回実行した時間の平均値である。検索時間を和歌の数を軸としてグラフ化したものが図 3、図 4 である。条件(1)の検索では、和歌数が 64260 首の時点で 10.1 秒で検索できるのに対して、条件(2)では、10080 首の時点で 264.2 秒もかかっている。これは、2 段の副問い合わせを用いることで、条件(2)の検索が句の数の 2 乗のオーダーとなっているためである。そこで、副問い合わせを用いずに、「仮名“やまとのくに”を含む句の原文を持つ和歌の id」を持つ一時テーブル t1 と「仮名“とりよるふ”を含む句の原文を持つ和歌の id」をもつ一時テーブル t2 を作成し、これらのテーブルに対して検索を行い、検索時間を測定する。リスト 3 はこの検索を行う SQL 文であり、表 2 の“条件(2)の 2”、図 5 は測定結果とそのグラフである。結果より、条件(2)の SQL 文と比較して検索時間の改善が得られている。さらに、今回の測定では、少数の和歌データをコピーしたダミーデータを利用しているため、一時テーブル中の行数が現実にはありえないほど多くの行を含んでいると考えられる。従って、実際のデータを格納した際には、現在よりも短い時間で検索できるものと推測される。

## 5 まとめ

本稿では異訓、異同を含めた検索が行える万葉集検索システムの構築を目的として、リレーショナルデータベース管理システムを用いた万葉集データベースを

構築した。まず、万葉集検索システムに対する要求を整理し、それを満たすため 4 つの設計方針に従ってデータベーススキーマを設計した。また、入力した少数のデータをもとにダミーのデータを作成し、データベースに全体の約 1/3 の和歌を格納し、それに対する 2 種類の検索の検索時間を計測することで評価を行った。単純な検索では実用的な時間で検索が終了したが、もう一方の検索は和歌数が少ない時点で実用的な時間で検索が出来なくなった。これに対しては、検索において一時テーブルを用いることで検索時間をかなり短縮することが出来たが、全ての諸本をデータベース化した場合には、検索時間の増加が問題となることが予想される。今後の課題は、検索時間の短縮、データ格納の効率化、GUI の実現などが挙げられる。

謝辞：本研究は、一部、文部科学省科学研究費 基盤 (C) (課題番号 11680417) による。

### 条件(1)の SQL 文

```
select waka.waka_no, hon.name from hon, ku,
waka where genbun ~ '尔' and ku.waka_id =
waka.oid and hon.oid = waka.hon_id;
```

### 条件(2)の SQL 文

```
select hon.name, waka.waka_no, ku.genbun,
ku.kana, ku.kundoku from hon, ku, waka where
hon.oid = waka.hon_id and ku.waka_id = waka.oid
and waka_id in (select waka_id from waka, ku
where ku.kana ~ 'やまとのくに' and ku.waka_id
= waka.oid and waka.oid in (select ku.waka_id
from ku where ku.kana ~ 'とりよるふ'));
```

## リスト 2：検索用 SQL 文

表 2：測定結果

条件(1)		条件(2)		条件(2) の 2	
和歌数	時間 (s)	和歌数	時間 (s)	和歌数	時間 (s)
3780	0.3	1260	0.9	3780	2.3
11340	0.9	2520	5.3	11340	16.9
18900	1.7	3780	15.8	18900	44.2
26460	2.7	5040	35.6	26460	84.9
34020	3.8	6300	66.9	34020	140.6
41580	5.7	7560	113.2	41580	209.1
49140	6.6	8820	177.8	49140	286.0
56700	8.3	10080	264.2	56700	388.3
64260	10.1			64260	486.3

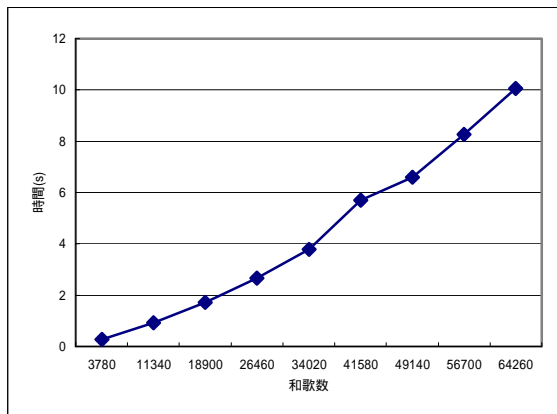


図3：条件(1)の検索時間

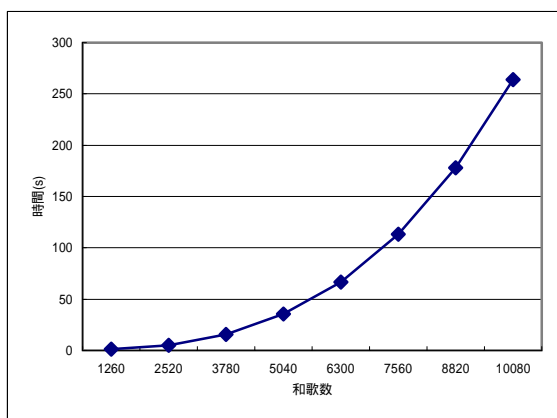


図4：条件(2)の検索時間

```

create temp table t1 as select ku.waka_id from
ku where ku.kana ~ 'とりよるふ';
create temp table t2 as select ku.waka_id from
ku where ku.kana ~ 'やまとのくに';
select hon.name, waka.waka_no, ku.genbun,
ku.kana, ku.kundoku from hon, ku, waka
where hon.oid = waka.hon_id and waka.oid =
ku.waka_id and ku.waka_id in (select
t1.waka_id from t2, t1 where t1.waka_id =
t2.waka_id);
drop table t1;drop table t2;

```

リスト3：検索用SQL文2

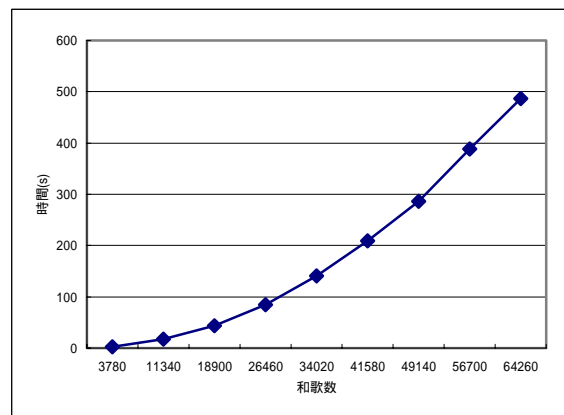


図5：条件(2)の2 検索時間

#### 参考文献

- [1] 安永尚志：“国文学研究とコンピュータ”、勉誠社、1998。
- [2] 佐竹昭廣、立川美彦：“重層型情報時代に対応する国文学高機能情報形成手法の開発とその実用化に関する研究”、平成7年度～平成9年度科学研究費基盤研究(A)(2)研究成果報告書(課題番号 07401014)、国文学研究資料館、1998。
- [3] 中村康夫、安道百合子 監修：“国文学研究資料館データベース 古典コレクション”、岩波書店、1999。
- [4] 「新編国歌大観」編集委員会：“CD-ROM 版 新編国歌大観”、角川書店、1996。
- [5] 久保田淳 監修：“新日本古典文学大系 CD-ROM 版 八代集”、岩波書店、1995。
- [6] 吉村誠：“万葉集テキスト Ver 4.0”、  
<http://yoshi01.kokugo.edu.yamaguchi-u.ac.jp/manyou/m>

anyou.html, 2001.

- [7] 松尾圭子、新原久仁子、中田充、吉村誠：“万葉集検索システム Ver.1.00”、  
[http://yoshi01.kokugo.edu.yamaguchi-u.ac.jp/~manyou/manyou\\_kensaku.html](http://yoshi01.kokugo.edu.yamaguchi-u.ac.jp/~manyou/manyou_kensaku.html), 1999。
- [8] 北川博之：“データベースシステム”、昭晃堂、1996。
- [9] 石井達夫：“PostgreSQL 完全攻略ガイド”、技術評論社、2001。
- [10] まつもとゆきひろ、石塚圭樹：“オブジェクト指向スクリプト言語 Ruby”、アスキー出版局、1999。
- [11] 書写者未詳、“西本願寺本”、14世紀ごろ書写。
- [12] 鶴久、森山隆：“萬葉集”、おうふう、1972。
- [13] 伊藤博：“万葉集全注”、有斐閣、1983。