データベース・システム 64－7
（１９８８．３．１５）

形式化された自然言語によるリレーショナルデータベースの問合せについて

河村一樹
日本電子専門学校

リレーショナルデータベースの普及にともない、エンドユーザーが直接データを操作したいという要求が生じつつある。そのためには、利用者インターフェイスの向上が必要となる。本稿では、リレーショナルデータベースの問合せ処理におけるユーザフレンドリなシステムの構築について論じる。形式化された自然言語をSQL (Structured Query Language)に変換し、動的SQL機能によってデータベースをアクセスするというシステムモデルを提案する。

Query of the Relational Database used by formalized natural language

Kazuki Kawamura
Nippon Electronics College
1－25－4, Hyakunin-cho Shinjuku-ku, Tokyo, Japan

Data and information cause by direct manipulation from the end user involve the diffusion of the Relational Database.

Accordingly, it is very important for us to improve the facilities of user interface.

This paper discusses the user friendly interface regarding query of the Relational Database. As a resort of implementation, this system produces formalized natural language which is translated into SQL.

Therefore, this system involves the SQL generator which is used by the natural language.

1. はじめに

情報システム構築のアプローチに、新しい考え方が導入されつつある。それは、従来型のプロセス指向からデータ指向への転換である。このことは情報システム構築のための主旨が、ソフトウェア生産から情報生産へかわっていくことを示している。表1に、プロセス指向とデータ指向の比較を示す。

表1　プロセス指向とデータ指向の比較

		プロセス指向	データ指向
適用工学分野		ソフトウェア工学	情報工学
生産物		ソフトウェア	情報
方法論		ライフサイクル論	プロトタイピング論
設計技法		トップダウン設計 構造化設計 複合設計 構造化プログラミング など	正規化技法 構造化分析 データフロー設計など
開発体制		SE、PG (エンドユーザ不在)	DBA、エンドユーザ (エンドユーザ共存)
実現環境	言語	手続言語(1～3GL)	非手続言語(4GL)
	DBMS	階層ネットワーク構造	リレーショナル構造 (DD/Dを含む)
	システム形態	中央集中型(TSS) スタンドアロン型	分散型(MML) ネットワーク型(LAN)

データ指向ということは、比較表の各項目をみてもわかるように、エンドユーザ主導型になりつつあるということになる。エンドユーザを中心に、データ指向アプローチを実現しようという姿勢である。その核になる技術分野に、リレーショナルデータベース(RDB)があげられる。

RDBは、表形式でスキーマを定義する。表によるデータ表現は、適用業務分野では台帳として最もよく用いられている。この結果、RDBはエンドユーザにとって大変なじみやすい情報資源管理用ツールとなる。
一方、エンドユーザ指向型のシステム環境を実現するためには、利用者インターフェイスの向上が必要になってくる。

利用者インターフェイス向上のためのアプローチには、ハードウェア機能(装置として)と、ソフトウェア機能(取扱い方法として)の拡充が要求される。ハードウェア機能としては、ライトペン、タッチスクリーン、マウス、ジョイステック、タブレット、高解像度ディスプレイ、そして、音声入力などがあげられる。ソフトウェア機能としては、ウィンドウスクロール、マルチウィンドウ、オンラインヘルプ、グラフィックエディタ、そして、自然言語処理などがあげられる。

このような背景から、本稿では、RDBにおける利用者インターフェイスを向上させるシステムの開発について取り上げる。RDBが情報資源管理システムの中核に位置づけられるためには、エンドユーザにとって使いやすいものになっていなければならない。そのための利用者インターフェイスとして、本システムでは、以下の機能をインプリメントの対象としている。

① 対話形式
② マルチウィンドウ
③ カーソル選択(ウィンドウガイド)
④ 形式化された自然言語処理
⑤ SQLに準拠した表検索処理
⑥ 検索しぼり込み処理

これらのことを前提条件に本稿では、まずシステム設計上の前提条件について述べる。本システムのシステム化における対象領域を明確に位置づけるとともに、設計にあたっての目標を設定する。次に、システム実現化のための環境について述べる。現在のところ、パーソナルコンピュータ上での開発環境となっているが、他のRDBへの移植も可能である。さらに、システムの中核部である構文解析の手順について述べる。形式化された自然言語によるRDB問合せ構文を、SQLに変換するための解析手順について解説することにする。

2. 設計上の目標と前提条件

本システムの設計にあたり、以下のような目標を設定した。

① 理解容易性
② 容易な操作性
③ 最小の記述量
④ 形式化仕様
⑤ 一貫性
⑥ 有益なフィードバック
⑦ 拡張性

システム全体に及ぶ設計理念として、理解容易性を前提条件とすることにする。このことから、システムに対する入力記述を最小にさせるために、一貫性のある様式のもとに形式化された仕様記述を採用する。また、対話処理における操作も簡便になるように、インターフェイスの設計を行なう。それとともに、システムの拡張性をはかるため、構文解析部のデータベース化やプログラムのモジュール化を行なう。これらの要素を考慮した設計を行なうことによって、利用者インターフェイスの向上がはかられることになる。

RDB問合せにおける利用者インターフェイスとしては、対話形式をとり上げる。対話形式においては、システムに対する指示命令の入力方式に以下のような種類がある。

対話形式
- メニュー選択方式
- カーソル選択方式
- コマンド入力方式
- 直接入力方式

4つの方式のうち、どれを採用するかは、システムを操作する利用者の能力レベルに依存する。本システムは、あくまでコンピュータに対しての初心者を対象としている。このため、コマンド入力や直接入力方式では、操作処理における負荷が大きく実用的ではない。そこで、メニュー選択かカーソル選択方式となる。

本システムでは、RDB問合せに対する処理記述において形式化仕様を導入する。具体的には、形式化された仕様記述言語を画面上に表示し、そこから入力することによって操作を進める。このことから、メニューよりもカーソルによる選択が有効となる。

また、データベース問合せ言語という面から言語体系を分類すると、以下のようになる。

データベース問合せ言語
- 表形式言語(QBEなど)
- 自然言語(ヤチマタなど)
- コマンド言語(SQLなど)
- 図形言語(IBSなど)

利用者インターフェイスという面からみると、コマンド言語は不利となる。一方、図形言語では現在のところ、動作効率と言語文法の特殊性という面から問題がある。これより表形式言語または、自然言語が対象となる。

RDB自体が表形式であることから、表形式言語が最適といえる。が、複雑な表検索では、表現上の限界がある。これより、自然言語形式を採用する。

自然言語処理においては、非形式化仕様と形式化仕様の2通りのアプローチがあげられる。前者の、自然言語そのものを用いると、質問事項を組み立ててから全文入力するという煩雑さがある。また、日本語文章のもつあいまい性、漠然性、多様性といったところにも問題がある。これに対して後者の形式化した自然言語を用いることによって、言語解析が容易になるといえる。また、全文入力は不要となり、必要事項だけをカーソル選択により入力することができる。RDB問合せという限られた適用分野であることから、仕様表現の範囲をしぼり込むことができ、形式化しやすい。形式化仕様を用いることによって、システムの一貫性も保たれることになる。

これらのことから、本システムでは、カーソル選択にもとずく形式化された自然言語による入力方式を、利用者インターフェイスとして設計することにする。

なお、RDB問合せを実現するコマンドとしては、SQLを基準とする。

SQLは、DDL(data definition language)とDML(data manipulation language)をもつが、本システムでは、DMLのうち検索コマンドのみを取り扱う。

DDLでRDBを定義(CREATE文)した後、DMLで表データを生成(INSERT文、UPDATE文、DELETE文)する。ここまでの作業が終了してから、生成した表に対しての検索(SELECT文)が行なえる。ここから、本システムを稼動することになる。ただし、現在のところ、単一表の検索だけをサポートする。

3. システム構成

本システムは現在パーソナルコンピュータ(PC9801)で開発を行なっている。使用しているDBMSの制約から、メモリは640KB、固定ディスク装置は20MBという構成である。ソフトウェアシステムとしては、表2にある構成をとる。

表2　本システムのソフトウェア構成

O　　S	MS-DOS 3.10
R　D　B　M	informix-SQL
SQLモジュール言語	informix-ESQL/C
C　言　語	MS － C
アセンブリ言語	MS-DOS Macro Assembler
日本語入力フロントプロセッサ	VJE-Σ 1.10

システム全体の機能構成を図1に示す。

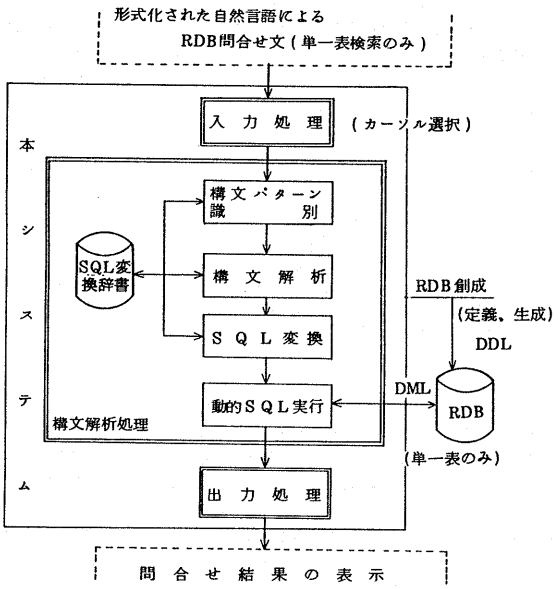


図1　システム全体の機能構成

次からは、本システムを構成する3つの処理について解説する。

3.1 入力処理

本システムを起動すると、RDBに登録されている全データベースの一覧をウィンドウで表示する。表示内容としては、個々のデータベース領域が有する表の名前、行の全長、列の総数、行の総数、作成日付が示される。なお、表示は、仮想ウインドウ形式(△▽マーク)で、上下左右自由にウィンドウスクロールができるようになっている。図2に例を示す。

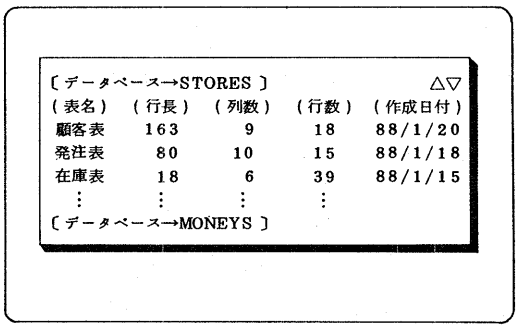


図2　データベース一覧

この中から、検索したい表が格納されているデータベース名をカーソルで選択する。そして、さらに単一表検索をカーソルで指定する。図3に例を示す。

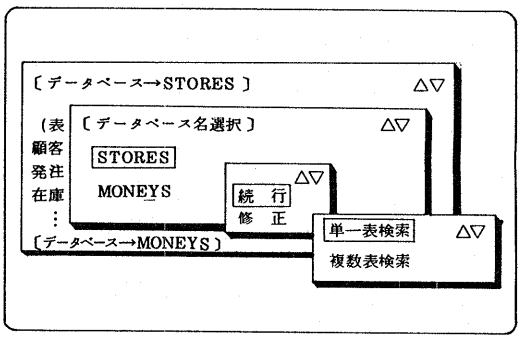


図3　データベース選択

次に、指定されたデータベース領域に登録されている全表の一覧を表示する。表示内容としては、個々の表が有する列の名前、列の型、列の長さが示される。利用者は、これによって、表に対する列名の情報すべてがわかることになる。図4に例を示す。

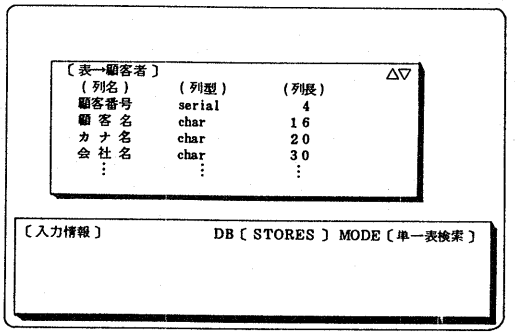


図4　表　一　覧

この中から、検索したい表名をカーソルで選択する。選択が終了すると表名の次に、「から，」空欄「ものを抽出し，」空欄「表示する。」という語句が自動的に挿入される。また、検索処理で、入力情報の表名を新たに指定し直さなければ、前回と同一の表が検索対象となる。この結果、ある1つの表に対するしぼり込み検索が可能となる。図5に例を示す。

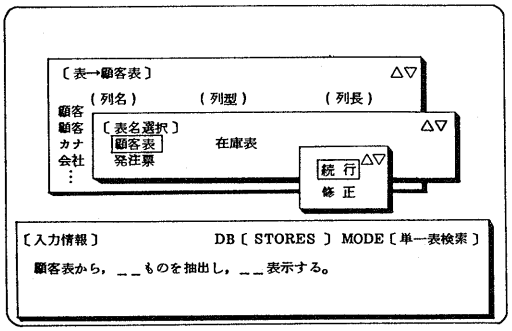


図5 表 選 択

ここからいよいよ単一表の検索にはいる。検索においては、列名、検索条件文を指定しなければならない。そのために、列名ウィンドウ、助詞ウィンドウ、接続詞ウィンドウ、条件用語ウィンドウを用意している。図6に各ウィンドウの表示一覧を示す。

助詞ウィンドウ

が	を

接続詞ウィンドウ

，かつ	「	もので，	が同一で，
，または	」	が同一の	

条件用語ウィンドウ

と等しい	と等しく	と等しくない	と等しくない	
より大きい	より大きく	より小さい	より小さく	(注1)
以上の	以上で	以下の	以下で	
から	までの	までで	までで ない	までで なく
か	のいずれかの	のいずれかで	のいずれかで ない	のいずれかで なく
で始まる	で始まり	で始まらない	で始まらずに	
で終わる	で終わり	で終わらない	で終わらずに	(注2)
をもつ	をもち	をもたない	をもたずに	
1桁目から	を含む	を含み	を含まない	を含まずに
桁目から	と一致する	と一致し	と一致しない	と一致せずに (注3)
桁から	桁までが	を持つ	を持ち	を持たない	を持たずに
の日付	の月	の年	の曜日	
の最大値	の最小値	その条件を満たす全件数		(注4)
の重複を除く件数	の平均値	の重複を除く平均値		
の合計値	の重複を除く合計値			
の昇順に	の降順に			
すべての	すべてを			

(注1)：関係演算子 (注2)：文字列関係子A
(注3)：文字列関係B (注4)：日付集計関数子

図6 ウィンドウ表示一覧

各ウィンドウから必要なキーワードをカーソルによって選択し、逐次入力情報の領域に入力していく。なお、検索条件を指定する条件値だけは、カーソルで選択するのではなく、条件値入力領域から、カナ漢字変換により入力する。図7に例を示す。

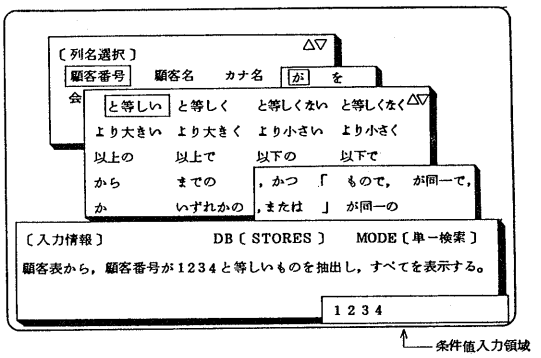


図7 検索条件指定

入力情報に、検索条件文が全文表示されてから、送信キーを押す。この段階で、入力処理が完了する。

以上のことから、本システムは、カーソル選択の操作が入力オペレーションの中心となり、条件値だけカナ漢字変換によって入力することになる。

3.2 構文解析処理

入力情報の領域から入力された、形式化自然言語を解析する処理である。自然言語そのものの場合は、形態素解析、構文解析、意味解析を行なうが、本システムでは、構文解析が中心となる。

形態素解析については、ＥＳＣ(escape sequence code)を付加することによって単語分解を行なう。本システムがもつ全ウィンドウ(列名、助詞、接続詞、条件用語)で表示されている語句の後に、ESCをつけてある。表示はされないが、内部コードとしてもつことになる。これを、プログラムで判別することによって、単語分解が可能となる。図8に、事例パターンを示す。

〔入力情報〕

顧客者から，顧客番号が 1234 と等しいものを抽出し，すべてを表示する。
△ △ △△ △ △ △ △

(△：ESC)

(プログラムにより解析)

顧客者	から，	顧客番号	が	1234	と等しい	ものを抽出し，	すべてを	表示する。

図8 ＥＳＣによる単語分解

次の構文解析だが、本システムでは意味解析をあえて行なわない。これは、入力する語句を形式化しているため、単語の意味が入力段階で判別していることに依存する。構文パターンさえ識別できれば、規則的にSQLへ変換することができる。SQL変換については、次の章で詳細に述べる。特徴としては、SQL変換辞書を使うことがあげられる。

SQLに変換した後には、検索対象となるRDBの表にアクセスしなければならない。本システムでは、informix－ESQL/Cのもつdynamic SQLを用いる。SQLのDMLには、モジュール言語機能が組み込まれている。これには、static SQLとdynamic SQLの２つがある。が、プログラム実行時にSQLを生成するシステム形態より、後者の機能を用いる。

3.3 出力処理

変換したSQLによってRDBの表を検索した結果を、編集して表示する。構文解析処理において論理的エラーが発生し、解析できないときは、エラーメッセージを表示し再入力を要求する。また、SQLへ構文変換ができても、実際にアクセスしたときにinformixからエラーが戻されたときは、そのエラー情報を出力する。

4. 構文解析手順

ここから、3.2で示した構文解析処理の詳細について解説する。

SQLのコマンド構成を示す。

```
SELECT ……  〔 SELECT句：列名、集計関数指定 〕
FROM   ……  〔 FROM句：検索表名指定 〕
WHERE  ……  〔 WHERE句：検索条件指定 〕
GROUP BY …… 〔 GROUP句：集計処理指定 〕
 HAVING  ……
ORDER BY …… 〔 ORDER句 ：並ぶかえ指定〕
```

入力情報の領域から入力された文章は、次のような形式化された構文となる。

(a) から、 (b) ものを抽出し、 (c) 表示する。
<表名領域> <条件領域> <表示領域>

(a)を表名領域、(b)を条件領域、(c)を表示領域と称する。各領域とSQLの変換上の対応は、以下のようになる。

- (a)の表名領域 → FROM句
- (b)の条件領域 → WHERE句，GROUP句
- (c)の表示領域 → SELECT句、ORDER句

この対応によって、形式化された自然言語からSQLへの変換が可能となる。次からは、各領域での構文解析及びSQL変換手順について示す。

4.1 表名領域

(a)に入力された表名をそのままSQLのFROM句へ変換する。

(a) から， ⇒ FROM (a)
<自然言語> <SQL>

4.2 条件領域

4.2.1 WHERE句、GROUP句なしの場合

条件領域に「すべての」が入力されているときは、SQLの変換は不要となる

すべての ものを抽出し，⇒
<自然言語> <SQL>

4.2.2 WHERE句への変換

条件領域の文章に対して、以下の解析を行なう。

① 複合条件文の解析
，かつ，または「 」を抽出した場合
② 単一条件文の論理関係に展開
複合条件文を、論理関係子を介在させ単一条件文に分割。このとき、論理関係子はSQLに変換する。，かつ→AND，または→OR「→（ 」→）
（ 単一条件文1 AND 単一条件文2 ）OR 単一条件文3
③ 単一条件文の解析
④ SQL変換

単一条件文の構文パターンとしては、以下の5種類とする。

Ⅰ．<表現式>が<表現式><関係演算子>
Ⅱ．<表現式>が<表現式>から<表現式>{までの / までで / までではない / までではなく}
Ⅲ．<表現式>が<定数>か<定数>か……{のいずれかの / のいずれかで / のいずれかでない / のいずれかでなく}
Ⅳ．<列名>が<定数><文字列関係子A>
Ⅴ．<列名>が<定数><桁数><文字列関係子B>

ここで<表現式>：=<列名>|<列名の算術式>|<日付関数>|<定数>とする。また、<関係演算子>、<文字列関係子A>、<文字列関係子B>は、すべて条件用語ウインドウに表示されており、カーソルによって選択できる。分類は、図6にある。

単一条件文の解析には、SQL交換辞書を用いる。辞書自体も、RDBMのリレーショナルな表とする。

SQL変換辞書には、条件演算子とSQLの対応をもつSQL変換辞書Aと，組み込み関数とSQLの対応をもつSQL変換辞書Bの2つがある。図9、図10に、SQL変換辞書A、Bの構成を示す。

日本語文	SQL1文	SQL2文	フラグ	
と等しい	=	(NULL)	0	構文パターンⅠ
と等しく	=		0	
と等しくない	!=		0	
と等しくなく	!=		0	
より大きい	>		0	
より大きく	>		0	
より小さい	<		0	
より小さく	<		0	
以上の	>=		0	
以上で	>=		0	
以下の	<=		0	
以下で	<=		0	
までの	BETWEEN	AND	1	構文パターンⅡ、Ⅲ
までで	BETWEEN	AND	1	
までではない	NOT BETWEEN	AND	1	
までではなく	NOT BETWEEN	AND	1	
のいずれかの	IN(	,	1	
のいずれかで	IN(	,	1	
のいずれかでない	NOT IN(	,	1	
のいずれかでなく	NOT IN(	,	1	
で始まる	MATCES〝	*〃	2	構文パターンⅣ
で始まり	MATCES〝	*〃	2	
で始まらない	NOT MATCHES〝	*〃	2	
で始まらずに	NOT MATCHES〝	*〃	2	
で終わる	MATCHES〝*	〃	2	
で終わり	MATCHES〝*	〃	2	
で終わらない	NOT MATCHES〝*	〃	2	
で終わらずに	NOT MATCHES〝*	〃	2	
をもつ	MATCHES〝*	*〃	2	
をもち	MATCHES〝*	*〃	2	
をもたない	NOT MATCHES〝*	*〃	2	
をもたずに	NOT MATCHES〝*	*〃	2	
を含む	MATCHES〝	?〃	3	構文パターンⅤ
を含み	MATCHES〝	?〃	3	
を含まない	NOT MATCHES〝	?〃	3	
を含まずに	NOT MATCHES〝	?〃	3	
と一致する	MATCHES〝?	〃	3	
と一致し	MATCHES〝?	〃	3	
と一致しない	NOT MATCHES〝?	〃	3	
と一致せずに	NOT MATCHES〝?	〃	3	
を持つ	MATCHES〝?	?〃	3	
を持ち	MATCHES〝?	?〃	3	

日本語文	SQL1文	SQL2文	フラグ
を持たない	NOT MATCHES "?"	?"	3
を持たずに	NOT MATCHES "?"	?"	3

図9　SQL変換辞書A

日本語文	SQL1文	SQL2文	フラグ	
の日付	DAY(	)	1	日付関数
の月	MONTH(	)	1	
の年	YEAR(	)	1	
の曜日	WEEKDAY(	)	1	
の最大値	MAX(	)	1	集計関数
の最小値	MIN(	)	1	
その条件を満たす全件数	COUNT(*	)	1	
の重複を除く件数	COUNT(DISTINCT	)	1	

日本語文	SQL1文	SQL2文	フラグ	
の平均値	AVG(	)	1	集計関数
の重複を除く平均値	AVG(DISTINCT	)	1	
の合計値	SUM(	)	1	
の重複を除く合計値	SUM(DISTINCT	)	1	
の昇順に	ORDER BY	(NULL)	0	
の降順に	ORDER BY	DESC	1	

図10　SQL変換辞書B

構文パターンの識別は、SQL変換辞書Aのフラグによって行なう。そのために、条件領域の日本語入力文の最後の語句をキーとして、SQL変換辞書Aをアクセスする。図11に動作状況を示す。

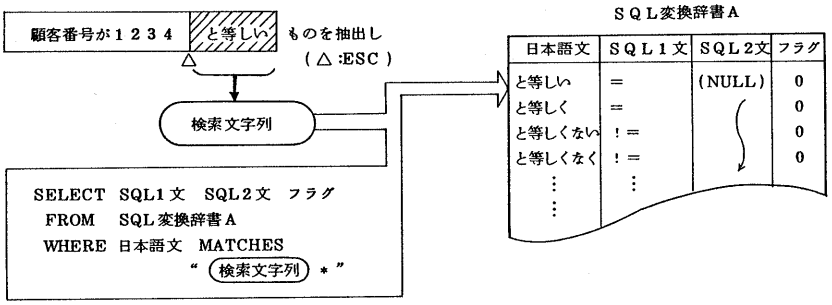


図11　SQL変換辞書Aへのアクセス

構文パターン(Ⅰ～Ⅴ)が決定した後、パターン毎にSQL変換を行なう。SQL変換では、検索によって該当したSQL変換辞書Aの列であるSQL1文とSQL2文を用いる。構文パターンⅠは、SQL1文だけ、以外はSQL1文とSQL2文を<表現式>や<定数>と組み合わせる。

(1) 構文パターンⅠ
<表現式>が<表現式><関係演算子>ものを抽出し ⇨ WHERE<表現式><SQL1文><表現式>
<自然言語>　<SQL>

(2) 構文パターンⅡ
<表現式>が<表現式>から<表現式>までのものを抽出し ⇨ WHERE<表現式><SQL1文><表現式><SQL2文><表現式>
<自然言語>　<SQL>

(3) 構文パターンⅢ
<表現式>が<定数>か<定数>のいずれかのものを抽出し ⇨ WHERE<表現式><SQL1文><定数><SQL2文><定数>)
<自然言語>　<SQL>

(4) 構文パターンⅣ
<列名>が<定数><文字列関係子A>ものを抽出し ⇨ WHERE<列名><SQL1文><定数><SQL2文>
<自然言語>　<SQL>

(5) 構文パターンⅤ
<列名>が<定数><桁数><文字列関係子B>ものを抽出し ⇨ WHERE<列名><SQL1文><定数><SQL2文>
<自然言語>　<SQL>

4.2.3 GROUP句への変換

WHERE句に相当する文章がない場合は、直接GROUP句変換をする。WHERE句がある場合は、日本語入力文から「もので，」という語句を抽出し、それ以降の文章がGROUP句の変換対象となる。図１２に列で示す。

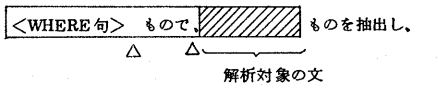
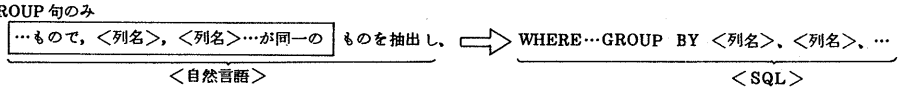


図１２　GROUP句変換

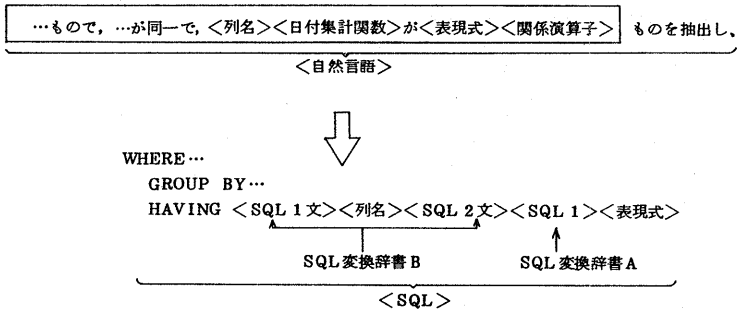
GROUP句の解析にあたっては、GROUP句のみとGROUP句＋HAVING句がある。この区分けは、解析対象文の中に「が同一の」という語句があればGROUP句のみ、「が同一で，」という語句があればGROUP句＋HAVING句とする。

(1) GROUP句のみ



(2) GROUP句＋HAVING句

解析対象の文の中に、<日付集計関数子>（図６）があるときは、SQL変換辞書Ｂをアクセスして対応するSQLコマンドをとり出し、SQLに変換する。アクセスする手順は、図１１と同一とする。

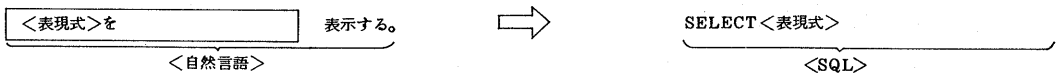


4.3 表示領域

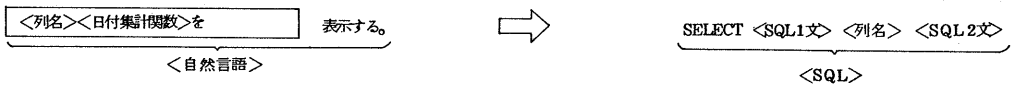
4.3.1 SELECT句への変換

表示領域に「すべてを」が入力されているとき、SQLの変換は、次のようにする。

以外のときは、列名及び日付集計関数が並ぶ。列名や列名の計算式の場合は、そのまま変換する。

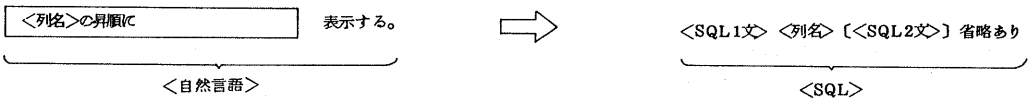


日付集計関数の場合は、4.2.3と同様SQL変換辞書Ｂをアクセスし、SQLへ変換する。



4.3.2 ORDER句への変換

表示領域に「の昇順に」「の降順に」が入力されているとき、同じくSQL変換辞書Ｂをアクセスする。



5. おわりに

RDBMが普及するためには、エンドユーザ指向的な利用者インターフェイスの提供が重要なポイントとなってくる。台帳にファイリングするような気持でRDBを構築し利用するためには、利用者の立場にたった操作性のよいインターフェイスが必要となる。そのための設計目標を設定し、システムとしての実現化をはかっている。カーソル選択にもとずく形式化された自然言語による入力方式を採用し、マルチウィンドウや検索しぼり込み機能を付加し、使いやすさやわかりやすさを追求している。

現在のところ、プロトタイプシステムとして開発中のため、システム全体の処理効率や利用者の評価を直接測定することはできない。が、RDB問合せにおける利用者インターフェイス実現のためのひとつのアプローチであることを提示したい。

今後の課題としては、単一表における副照合処理やデータ変更処理を組み込むことを予定している。前者は、SELECT文のネスト構造による検索パターンに相当する。ある条件によって検索した結果を条件として、さらに検索をしぼり込みながら続けていく処理である。後者は、DMLの検索コマンドから操作コマンドへの拡張を意味する。実在する表の行を削除(DELETE文)または、追加(INSERT文)するか、表の要素を更新(UP−DATE文)するかのいずれかの処理を行なうパターンである。

また、単一表だけでなく複数表の検索についても拡張をはかりたい。RDBの特徴として位置づけられる給合機能(JOIN)や、複数表間での集合演算処理(UNION, EXITS, NOT EXITS)などについても取り上げたい。

謝　辞

本報告は、日本大学大学院理工学研究科で著者が行なっている研究の一部をまとめたものである。御指導いただいている理工学研究科の高橋寛教授に感謝します。

参考文献

[1] Shneiderman, B. : Designing the User Interface, Carol Wald, 1987

[2] Zloof, M.M. : Office − by − Example A business language that unifies data and word prceussing and electronics mail, IBM Systems J., Vol 21, No 3, pp.272 − 304, 1982

[3] 藤崎哲之助他：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、情報処理学会論文誌、Vol 20, No 1, pp.77 − 84, 1979

[4] ISO 9075 − 1987 Database language SQL, 1987

[5] JIS X3005−1987 データーベース言語 SQL, 日本工業標準調査会審議, 1987

[6] 平尾隆行：関係データベースシステム、近代科学社、1986

[7] 片貝孝夫：Informix入門、アスキー出版局、1985

[8] 例ASCⅡマニュアル
informix − SQL USER'S MANUAL
informix − SQL REFERENCE MANUAL Ⅰ
informix − SQL REFERENCE MANUAL Ⅱ
informix − ESQL/C PROGRAMMER'S MANUAL