データ・ベースシステム 71－4
（１９８９．５．１８）

日本語サポートと正規表現

平沼雄一郎　　芝野耕司
沖電気工業株式会社　　東京国際大学

データベース言語ＳＱＬを拡張し、ＳＱＬ２からＳＱＬ３へと発展していく中で、ＳＱＬで扱うデータ型およびデータ表現も拡張されている。その中でも、文字列型および文字列表現に関するＳＱＬ言語規格の拡張に関しては、日本の貢献が大きく、ＳＱＬ２では、日本語データ型および日本語表現のサポートが導入され、ＳＱＬ３では、正規表現による文字列の照合比較機能が追加された。ここでは、日本語サポートと正規表現の導入の必要性と導入の際の問題点について論じ、導入方法について説明し、今後の拡張方向について述べる。

"National Character Set Support and Regular Expression in SQL"

by Yuichiro HIRANUMA (OKI Electric Industry Co., Ltd. 1-16-8, Chuuou, Warabi-shi, Saitama pref. 335, Japan)
and Kohji SHIBANO (Tokyo International University)

A few enhancement features about the data type and the data representation are introduced into SQL2 and SQL3, the extended versions of the Dabase Language SQL. Owing to Japanese contributions, the national character set support is introduced into SQL2 and the pattern matching by the regular expression is introduced into SQL3. In this paper, we discuss requirements, problems, solutions, and future extensions of these features.

１．はじめに

現在のデータベース言語ＳＱＬでは、文字データは、１バイト文字しかサポートされておらず、２バイトの漢字コード等は、サポートされていない。文字列の照合比較でも、一般の正規表現を用いた照合比較をサポートしていない。

ＳＱＬの拡張版であるＳＱＬ２及びＳＱＬ３では、データ型およびデータ表現も拡張されている。その中でも、文字列型及び文字列表現に関する拡張に関しては、日本の貢献が大きい。ＳＱＬ２では、日本語データ型及び日本語表現のサポートが導入され、ＳＱＬ３では、正規表現による文字列の照合比較機能が追加された。ここでは、日本語サポートと正規表現の導入の必要性、導入の問題点、導入方法及び今後の拡張方向について述べる。

２．日本語サポート

まず、ＳＱＬ２に導入された日本語機能について述べる。

２．１　ＳＱＬにおける文字表現の現状

最初のＳＱＬでは、データベースに格納するデータの表現としては、１バイトの文字の列からなる文字列しかサポートしておらず、表名や列名などの識別子についても１バイトの文字からなる文字列しか許していない。

現在のミニ・マイクロでのＤＢＭＳの日本語化では、漢字が通常の英数字と完全に混在する方式での日本語化が見受けられる。しかし、こうした方式では、日本語を単にデータとして格納することは出来ても、一つの”文字”として処理することは困難であった。すなわち、比較において単にバイト列としての比較しか行わないために、文字としての比較の効率が落ちたり、インデックスやＬＩＫＥ述語での部分文字列検索のアルゴリズムを用いることができないため、処理が遅くなる。あるいは、漢字の２バイト目を１バイトコード又は漢字の１バイト目と間違うということが起こる。

しかし、日本におけるデータベースアプリケーションでは、データとして日本語を扱うことが多く、ＳＱＬにも日本語処理機能を追加することが必須であった。

２．２　ＳＱＬ２への日本語機能提案

データベース言語における日本語機能は、各国文字集合サポートとして、昭和６０年よりＩＳＯへ提案を行ってきた。日本語機能は、ＳＱＬ言語の将来の機能追加の一環として取り入れることが昭和６２年６月１５日に出版された現在のデータベース言語ＳＱＬ（ＩＳＯ９０７５）の規格に注記された。

データベース言語グループでは、その後各国文字集合サポート機能について、言語仕様の詳細について検討を行ってきた。また、昭和６２年６月以来日本語機能委員会及び SC22/FORTRAN WG と調整を行い、９月初めにＩＳＯ参加各国に寄書を送付した。

日本提案の審議は、ＩＳＯでは、昭和６２年１０月１２日から始まったアムステルダムでのデータベース言語ラポータグループでの会議で審議され、承認された。ＡＮＳＩにおいても、１０月初旬のＸ３Ｈ２の会議で承認された。この結果は、１１月に出されたデータベース言語ＳＱＬ２の作業文書以降、正式に作業文書に入れられている。（［１］）

ＪＩＳでは、ＪＩＳ　Ｘ３００５－１９８７データベース言語ＳＱＬがまだ普及の初期であり、国内では、日本語機能なしの処理系は考えられないこと。早期にこの日本語機能を取り入れないと日本語機能の標準化の意義及びＳＱＬ言語自体の標準化の意義が薄れると考えられたため、ＩＳＯでの提案に合わせてＪＩＳ　Ｘ３００５－１９８７の改正で導入されることになった。その結果、平成元年前半に発行されるＪＩＳ　Ｘ３００５－１９８９データベース言語ＳＱＬに、ＪＩＳ全体で初めての日本語機能が追加されている。

日本から提案した各国語サポート機能については、英国から複数のコード体系をサポートするこ

とによる問題点と、2バイトコードのＩＳＯでの扱いについての決着がついてないことを理由に難色が示されたが、データベース言語では、コードの物理的表現や照合順番等については、処理系作成者の定義にまかされており、現在の仕様では扱えない2バイトコードをサポートするための器を用意することには大きな意味があるとの理解で、日本提案が採用された。また、日本の強い要求が通りＳＱＬ２の核にも入ることのコンセンサスが得られている。

ＳＱＬ２における日本語機能の導入においては、以下の問題点があった。（［３］）

（１）シフトコード系と非シフトコード系

残念ながら、計算機内部での漢字コード表現には様々なものが用いられている。この漢字コードは、大きく分けて、ホストの漢字シフトを用いるコード系（ＪＩＳ、ＥＢＣＤＩＣ、ＪＥＦ、ＫＥＩＳ等)とミニ・マイクロでの漢字シフトを用いないコード系(シフトＪＩＳ、ＥＵＣ等)が混在している。ＳＱＬの日本語化では、これらのコード系については、コード系間の違いを吸収し、どのコード系でも実現可能なように日本語化を考える必要がある。

（２）日本語識別子の扱い

日本語識別子を従来の識別子と別のカテゴリとして扱うと、システム辞書において、表名や列名などのデータ型を文字列型とするのか、日本語文字列型とするのかなど、データベース管理システムの識別子の扱いが複雑になる。

（３）日本語文字列定数の表現方法

ＳＱＬにおける従来の文字列定数の表記方法にそのまま従うと、'日本語'のようになる。ところがこの表記方法では、日本語文字列定数の取り出しが困難である。

（４）英数字と日本語の混在

１バイトの英数字と２バイトの日本語を混在させた混在文字列は、定義が大変きごちない上に、異なったシステムに可搬するのが大変難しいことが知られている。このような、混在文字列をどう扱うのか。

（５）英数字と日本語の比較可能性

現在のＳＱＬでは、データ型の比較可能性は、大きく二つのクラスで定義されている。すなわち、文字と数の二つのクラスである。各々のクラスの中では、データ型は、比較可能であるが、文字と数との間では、比較可能ではない。

数については、例えば、DECIMALとINTEGERが比較可能であるように、比較可能性の範囲が拡張されている。このデータ型についての一般化は、数の場合、妥当であると考えられる。同じようなデータ型についての一般化が文字の場合に妥当であるか。つまり、1バイトのアルファベットAと２バイトのアルファベットＡを比較上同じ文字と見なすのかどうか。

（６）日本語以外の各国語の導入

日本語以外の中国語、韓国語やアラビア語などの各国語のサポートをどうするのか。

（７）ホスト言語結合

ＳＱＬのデータ型は、すべてホスト言語のデータ型のいずれかに対応しなければならないが、ＳＱＬ２のホスト言語であるＡｄａ、Ｃ、ＣＯＢＯＬ、ＦＯＲＴＲＡＮ、Ｐａｓｃａｌ、ＰＬ／Ｉのいずれにも日本語データ型は存在しない。ＳＱＬ２の日本語データ型をホスト言語のどのデータ型に対応させるのか。

上に挙げた問題点に対し、日本のデータベース言語グループでは、次のような方法で日本語機能を導入した。（［４］）

（１）シフトコード系と非シフトコード系

シフトコード系と非シフトコード系の違いを吸収するために、日本語（各国語）文字集合に加えて、日本語（各国語）文字表現を導入した。すなわち、日本語（各国語）データ型の内部表現は、シフトコード系の場合でもシフトコードを外した日本語（各国語）コードのみを含む。とくに、ＩＳＯ／ＪＩＳ等のシフトコード系を考えた場合、シフトコード系でのデータ及び定数の表現は、一意ではない（すなわち、シフトイン／シフトアウトの対がデータ中の色々な所に出現することが考えられる）。そのためにも、定数として表記されたデータをデータベース言語処理系が前処理を行うことを規定する必要がある。

（２）日本語識別子の扱い

識別子は、単に他の識別子と区別可能であれば良く、日本語識別子であってもその日本語としての意味的処理を要求されることはないと考える。従って、日本語識別子は、従来の識別子と同じ型の文字列として扱うこととする。すなわち、日本語識別子の長さは、日本語識別子で用いられる日本語文字列表現を文字として解釈したときの文字の数とする。すなわち、（シフトコードを用いる漢字コードの場合には、シフトコードを含む）２バイトコードのバイト数にあたる。

（３）日本語文字列定数の表現方法

日本語文字列定数の表現方法として

N'日本語'

なる記法を導入することとする。これにより、日本語文字列定数と国際文字列定数が識別可能になった。

（４）英数字と日本語の混在

英数字との混在は、文字列データ型内で許すが、この場合の取り扱いは、通常の英数字と同様のビット長として取り扱い日本語／漢字についての特別な配慮は行わないこととする。

（５）英数字と日本語の比較可能性

通常の比較は、文字列として扱う。文字列は、各国の国語を表わす。すなわち、ＡＳＣＩＩ文字列は、英語を表わし、ＪＩＳ漢字列は、日本語を表わす。異なった国語間での比較には、基本的には、コード列の比較ではなく、もっと上位で意味的な比較を行う必要がある。例えば、

'desk' = N'机'

という等号の解釈は、ＳＱＬ言語の枠を越える。従って、ＳＱＬ２においては、英数字と日本語文字は比較不可能とする。

各国文字同士の比較を行う場合、国際文字の場合と同様に、短い方の文字列が各国文字集合中の空白で実効的に右に拡張された文字列に対し比較が行われる。比較順序は、国際文字の場合と同じく処理系作成者の定義による。

ＬＩＫＥ述語でのパターン照合比較は、各国文字についても拡張され、適用することができる。ただし、パターンには、各国文字定数を指定し、任意文字及び文字列指定子 の規定値は、それぞれ各国文字集合中の_と%とする。

（６）日本語以外の各国語の導入

日本語以外の各国語を扱えるようにするため、新しく導入したデータ型は、各国語データ型とし、データ型の名前は、以下に示す通りとした。また、このデータ型は、上述のようにマルチバイトの器を用意するに留めた。

NATIONAL CHARACTER, NATIONAL CHAR, NCHAR

（注：NATIONAL CHAR及びNCHARは、NATIONAL CHARACTERの同義語とする。）

そして、その各国語がどの言語に当たるかは、各国の標準化に委ねられた。また、各国語文字のバイト長は、規格上”Ｋ”と表現され、Ｋがいく

つであるかは、作成者定義となった。（注：ＪＩＳでは、Ｋ＝２と規定し、各国語文字を漢字と呼び、２バイトの漢字コード系を対象とした。）

また、ＳＱＬ２での各国文字集合サポート機能についての日本提案では、多バイト系の各国文字集合サポートとともに、拡張文字として１バイト系の各国文字のサポートも同時に提案している。

（７）ホスト言語結合

各国語文字列型は、ホスト言語の文字列型とのみ対応することとした。つまり、ＳＱＬ２の各国語文字列型の長さをＬとした場合、対応するホスト言語の長さＫ＊Ｌの文字列型に対応することとした。ただし、将来ホスト言語で各国語文字列型が導入された場合には、ＳＱＬ２の改正でその各国語文字列型への対応に変更される予定であることをＳＱＬ２規格案中に注釈し、他のプログラム言語規格が早期に対応するように要請した。

ＳＱＬ２に導入された日本語機能を簡単にまとめると次のようになる。（［６］）

①データ型

NATIONAL CHARACTER | NATIONAL CHAR | NCHAR

漢字の２バイトコードのみからなるデータ型。シフトコードがある表現を用いる場合でもシフトコードは、内部表現としては含まない。ここで、NCHAR及び NATIONALCHAR は、NATIONAL CHARACTER の同意語である。

②定数

N'日本語'

漢字の２バイトコードのみからなるデータ型。シフトコードがある表現を用いる場合でもシフトコードは、内部表現としては含まない。

③識別子

＜漢字識別子＞

漢字識別子の長さは、漢字識別子で用いられる漢字列表現を文字として解釈したときの文字の数とする。すなわち、（シフトコードを用いる漢字コードの場合には、シフトコードを含む）２バイトコードのバイト数にあたる。

④文字列型、文字列定数、注釈

文字列型の列、文字列定数及び注釈には、漢字列を漢字列表現として含めることができる。このとき、漢字列は、文字として解釈される。

⑤ホスト言語結合

ホスト言語との結合は、漢字列型は、２倍の長さをもつ文字列型に対応する。ホスト言語のどのデータ型と漢字列型（ＩＳＯ では、各国語文字列型 NCHAR）についてのみアムステルダム会議で変更した。変更内容は、漢字列型は、ホスト言語の文字列型とのみ対応し、漢字列型の長さをＬとすると、対応する文字列型の長さは、Ｋ＊Ｌとする（もとは、２＊Ｌであった。）ただし、Ｋは、作成者の定義による。これは、３バイト及び４バイトコードなどを考慮したためである。

２．３　文字サポートに関する今後の拡張

ＳＱＬにおける文字サポートに関する今後の拡張として、照合順番表と変換表が検討されている。以下では、この照合順番表と変換表の導入について述べる。

文字コードには、二つの役割がある。第一の役割は、一つ一つの文字を識別することであり、第２の役割は、一つ一つの文字の相対的な順番を規定することにある。

通常は、一つ一つの文字の字体を識別する必要があると考えられるが、場合によっては、違った字体の文字を同一視することが要請される。例えば、アルファベットの大文字と小文字を同一視し

たり、ＡＳＣＩＩ文字のアルファベットとＪＩＳ２バイトコードのアルファベットを同一視したりしたいという要請が有り得る。文字変換の表を用意し、この表をデータベース管理システム内部での処理に用いる事ができれば、このようなことが可能になる。

コードによって定まる文字の照合順番は、単にソートのために用いられるだけではなく、すべての文字列処理を行う場合の前提となるものである。すなわち、プログラムでの文字列を対象としたすべての大小比較で、この照合順番は効力をもつ。欧米では、この照合順番は、各国語を計算機内部で扱う際に、常に問題になる点である。すなわち、アクセント付のａは、アクセント無しのａに近い照合順番を与えたいが、８ビットの各国文字コードを規定するＩＳＯ　８８５９では、離れた位置にくることにより、本来の照合順番との差異が生じる。アメリカでも照合順番に関連する問題が存在する。すなわち、ＡＳＣＩＩとＥＢＣＤＩＣのコード系の違いにより、処理結果に違いが生じる可能性がある。とくに、この問題は、最近のＭＭＬの流行により深刻となってきている。日本でも、欧米と同様にＪＩＳの第１水準と第２水準を同時に用いる際には、第１水準は、代表的な読みの順であり、第２水準は、部首順である事から、異なった基準に基づいた照合順番が混在する事になり、計算機以外では、決して用いられたことのない照合順番が用いられることになる。

何等かの統一された基準に基づく照合順番表を用意し、これらの照合順番表をデータベース管理システム内での処理で用いられるようにすることが望ましい。

以上の理由で、変換表と照合順番表をサポートすることが日本の提案により検討されている。（［２］）

３．正規表現

ここでは、ＳＱＬ３に導入された正規表現による文字列の照合比較機能と正規表現部分文字列関数について述べる。

３．１　ＳＱＬにおける文字列照合比較

ＳＱＬには、ＬＩＫＥ述語と呼ばれる述語があり、文字列照合比較の機能を持っている。例えば

```
SELECT EMP#, EMPNAME, DEPTNAME
FROM EMP, DEPT
WHERE   EMPNAME LIKE '%JOHN__'
AND     EMP.DEPT# = DEPT.DEPT#
```

なる問合せ指定の中で" EMPNAME LIKE '%JOHN__' " の部分がＬＩＫＥ述語である。これは、EMPNAMEという属性がJOHNという綴りを含みJOHNのあとが３文字の場合に真となる述語である。例えば、EMPNAMEがBEN JOHNSONの場合に真になる。

３．２　正規表現による文字列照合比較の提案

ＵＮＩＸやエディタ中での検索では、一般に正規表現を用いることができる。この正規表現は、ＳＱＬ　ＤＢＭＳをＩＲ系のシステムとして用いることや全文データベースには、必要な機能である。ところが、現在のＳＱＬでは、正規表現を用いた文字列の照合比較は不可能である。

そこで、正規表現による文字列照合比較機能の導入が、１９８８年初めから日本国内で検討され、１９８８年７月に開催されたＩＳＯのデータベース言語ラポータグループ会議で導入が承認された。その後、正規表現の拡張と正規表現部分文字列関数の検討が日本国内で行われ、この検討に基づく拡張案が現在ＩＳＯのデータベース言語ラポータグループに提案されている。（［７］）

正規表現による照合比較機能を導入する際の問題点として以下のものがあった。

（１）ＬＩＫＥ述語との整合性

正規表現による照合比較を行う機能を実現するためには、現在ＳＱＬが持っている２つの特殊文字％と＿以外の特殊文字を導入する必要がある。しかし、現在のＳＱＬのＬＩＫＥ述語に特殊文字を導入すると、現在のＳＱＬからＳＱＬ２への上位互換性が保てなくなる。上位互換を保持し、正規表現による照合比較機能を実現するにはどうすればよいか。

（２）正規表現の選択

現在、正規表現の表記方法は世の中に多数存在する。照合比較をする場合、パターンを指定する必要があるが、パターンで使用する正規表現の表記法をどれにするのか。

また、正規表現による照合比較を行った場合、パターンマッチしたときに、文字列中からマッチした文字列を取り出したいという要求が考えられる。例えば、'from TOKYO to KYOTO'なる文字列は、'from * to *'なる正規表現にマッチするが、TOKYOやKYOTOなる部分文字列を取り出せる関数があると便利である。そこで、そのような機能を持つ正規表現部分文字列関数の導入も検討された。

検討が行われた結果、次のような方法で正規表現による照合比較機能がＳＱＬ３に提案された。

（１）ＬＩＫＥ述語との整合性

ＬＩＫＥ述語以外の新しい述語　ＳＩＭＩＬＡＲ述語を導入することとした。これにより、ＬＩＫＥ述語は、現在のＳＱＬのまま残るためＳＱＬ３のＳＱＬとの上位互換性の問題は、解決された。

（２）正規表現の選択

現在、ＵＮＩＸの標準として標準化が進んでいるＰＯＳＩＸにおける正規表現を正規表現の標準であると考え、このＰＯＳＩＸの正規表現をベースにした正規表現をＳＱＬ３における正規表現に採用することとした。

ＳＱＬ３に導入された正規表現による文字列照合比較機能と正規表現部分文字列関数を簡単にまとめると次のようになる。（［５］）

（１）ＳＩＭＩＬＡＲ述語

ＳＱＬ３に導入された正規表現を以下に示す。

〈正規表現〉 ::=
　〈正規項〉 |
　〈正規表現〉〈縦棒文字〉〈正規項〉
〈正規項〉 ::=
　〈正規因子〉 |
　〈正規項〉〈正規因子〉
〈正規因子〉 ::=
　〈正規一次子〉 |
　〈正規一次子〉* |
　〈正規一次子〉+
〈正規一次子〉 ::=
　〈文字〉 |
　〈漢字〉 |
　〈文字集合〉 |
　(〈正規表現〉)
〈文字集合〉 ::=
　[〈文字列挙〉...] |
　[^〈文字文字列挙〉...] |
　[:〈文字集合指示子〉:]
〈文字列挙〉 ::=
　〈文字〉... |
　〈文字〉 - 〈文字〉
〈漢字列挙〉 ::=
　〈漢字〉... |
　〈漢字〉 - 〈漢字〉
〈文字集合指示子〉 ::=
　ALPHA | UPPER | LOWER | DIGIT |
　ALNUM

ＳＩＭＩＬＡＲ述語は、

　文字列　SIMILAR TO パターン

の形をしており、パターンに上で示した正規表現を記述することができる。ＳＩＭＩＬＡＲ述語は、文字列がパターンに示す正規表現に照合合致するときに真となり、照合合致しないときに偽ととなる。

（２）正規表現部分文字列関数

ＳＱＬ３に導入を提案した正規表現部分文字列関数は、次の形をしている。

```
SUBSTRING(文字列,位置指定付正規表現
                エスケープ文字)
```

位置指定付正規表現とは、上で述べた正規表現の中に、取り出したい部分文字列に当たる部分をエスケープ文字と" の列で囲んだものである。例えば、'from TOKYO to KYOTO'なる文字列からTOKYOを取り出す場合、

```
SUBSTRING ('from TOKYO to KYOTO',
                'from ¥"*¥" to *', '¥')
```

と指定すると、この関数の結果は、'TOKYO'となる。

３．３　文字列照合比較に関する今後の拡張

文字列照合比較に関する今後の拡張としては、正規表現中で文字のユーザ定義クラスを指定可能にすることを検討している。例えば、 [:常用漢字:], [:かな及びカナ:], などの文字集合をユーザが定義できるようにすることである。

４．おわりに

ＳＱＬ２及びＳＱＬ３で文字列型と文字列表現に関して、どのような拡張がなされているかを述べた。今後は、上で述べたような拡張が現在検討中であり、上に挙げたもの以外の拡張も行われることが有り得る。

参考文献

［１］芝野耕司：「４．５　データベース言語」、情報処理学会情報規格調査会日本語機能委員会昭和６２年度報告書

［２］芝野耕司：「ユーザ定義による照合順番表のサポート」、日本語機能ＮＷＩ小委員会（昭和６３年９月）

［３］芝野耕司：「ＳＱＬ２における日本語機能」、情報処理学会第３６回全国大会（昭和６３年前期）

［４］芝野耕司：データ管理調査研究委員会昭和６２年度報告書、日本規格協会

［５］Y.Hiranuma: "SIMILAR Predicate and Regular Expression Substring Function in SQL3", ISO/IEC JTC1/SC21/WG3 DBL-SYD-94

［６］J.Melton: "Database Language SQL2", ISO/IEC JTC1/SC21 N3155, Jan 1989

［７］J.Melton: "Database Language SQL2 and SQL3", ISO/IEC JTC1/SC21/WG3 DBL-CAN-3, Feb 1989