

## 集合指向言語 SOL のデータベースへの応用

3N-3

- SQL のホスト言語への SOL の拡張 -

與那覇 誠      重松 保弘      吉田 将  
九州工業大学

## 1. まえがき

リレーショナルデータベース (RDB) 言語 SQL をあるホスト言語に埋め込んで使用する場合、両言語の結合が弱いためプログラムに余分な手続きが必要となり、また様々な制限が付加される。特に、両言語間でのデータの受渡しに対しての制限が強い。重要な点は、SQL 言語は集合単位の操作であるのに対し、従来のホスト言語 (Cobol, Fortran, Pascal, PL/I) <sup>(1)</sup> は集合をサポートしていない (Pascal は集合をサポートしてはいるが、SQL のホスト言語として使用するには不十分である) ため、集合単位の情報を一括して受け渡すことができないことである。データの受渡しが atom 単位に限られるため、ホスト言語のレベルで特別な処理が必要となり、その結果プログラムが複雑になる。

これらの問題点は、ホスト言語に著者らの開発した集合指向言語 SOL (Set Oriented Language) <sup>(2)</sup> を用いることにより解決できると考えられる。SOL は集合、写像および述語論理の記法を導入した言語であり、これを SQL のホスト言語として使用することによって、データの受渡しがスムーズに行える。

以上の理由により筆者らは、ユーザインタフェースの向上を目的として、集合指向言語 SOL を RDB 言語 SQL のホスト言語として使用できるように、SOL の言語処理系の拡張を行った。

## 2. 説明に使用するデータベース

S 表は供給者表であり、供給者番号 (SNO)、供給者名 (SNAME)、状態値 (STATUS)、所在場所 (CITY) を含む。

P 表は部品表であり、部品番号 (PNO)、部品名 (PNAME)、色 (COLOR)、重量 (WEIGHT) を含む。

SP 表は納入表であり、供給者 (SNO) が部品 (PNO) をどれだけ (QTY) 納入しているかを示している。

表 1 : 説明に使用するデータベース

S	SNO	SNAME	STATUS	CITY	SP	SNO	PNO	QTY
	S1	Smith	20	London		S1	P1	300
	S2	Jones	10	Paris		S1	P2	200
	S3	Blake	30	Paris		S1	P3	400
	S4	Clark	20	London		S1	P4	200
	S5	Adams	30	Athens		S1	P5	100
P	PNO	PNAME	COLOR	WEIGHT		SNO	PNO	QTY
	P1	Nut	Red	12		S2	P1	300
	P2	Bolt	Green	17		S2	P2	400
	P3	Screw	Blue	17		S3	P2	200
	P4	Screw	Red	14		S4	P2	200
	P5	Cam	Blue	12		S4	P4	300
						S4	P5	400

## 3. ホスト変数への検索結果の割り当て

## (1) ホスト言語が PL/I の場合

検索結果を atom 単位でしかホスト変数に渡すことができない。そのため検索結果が複数個になる可能性を持つ場合には、“カーソル”を用いた特別な処理が必要となる。それは、必ず“カーソル宣言”、“OPEN”、“FETCH”、“CLOSE”およびホスト言語におけるループ構造を必要とする。例えば、「ロンドンにある供給者の名前を求める」プログラムの一部を PL/I を用いて書くと図 1 のようになる。

```
EXEC SQL DECLARE C1 CURSOR FOR
      SELECT SNAME
      FROM   S
      WHERE  CITY='London';
EXEC SQL OPEN C1;
EXEC SQL FETCH C1 INTO :ANS;
DO WHILE (SQLCODE=0);
      PUT LIST(ANS);
      EXEC SQL FETCH C1 INTO :ANS;
END;
```

EXEC SQL CLOSE C1;  
図 1 : カーソルを用いた検索手順

## (2) ホスト言語が SOL の場合

SOL では集合をサポートしているため、検索結果を集合変数に割り当てることができる。図 1 と同様のプログラムを図 2 に示す。この例では、INTO 節において検索結果を集合変数 ANSSET に与えている。

```
EXEC SQL SELECT SNAME
      INTO   :ANSSET
      FROM   S
      WHERE  CITY='London';
write(ANSSET);
```

図 2 : SOL で書いた図 1 と同様のプログラム

また、検索結果が複数列の場合、それは関係を表している。SOL では、複数列の検索結果を一括して“集合と写像”に割り当てることが可能である。こうすることにより検索後の作業が行いやすくなる。また何度も同じような検索をする場合には、一度 SOL の方へデータを移し、それから検索するほうが実行効率がよい。

図 3 および図 5 に例を示す。INTO 節において割り当て形式が表現される。図 3 では写像 f の SNO に対する像が SNAME、写像 g の SNO に対する像が CITY となるように割り当てられる。

```
EXEC SQL SELECT SNO, SNAME, CITY
      INTO < f(SNO)=SNAME, g(SNO)=CITY >
      FROM S
      WHERE STATUS<30 ;
```

図3：検索結果を集合と写像に与える例1

これにより、左下の検索結果が右下の“集合と写像”に割り当てられる。

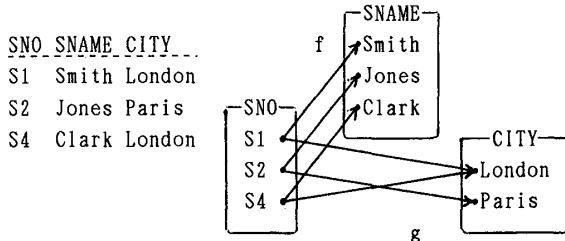


図4：図3の検索結果と割り当て状態

```
EXEC SQL SELECT SNO, PNO, SNAME
      INTO < f(SNO)={PNO}, g(SNO)=SNAME >
      FROM SP, P
      WHERE S.SNO=SP.SNO ;
```

図5：検索結果を集合と写像に与える例2

これにより、左下の検索結果が右下の“集合と写像”に割り当てられる。このような場合、正規形の表から非正規形の“集合と写像”へ変換される。

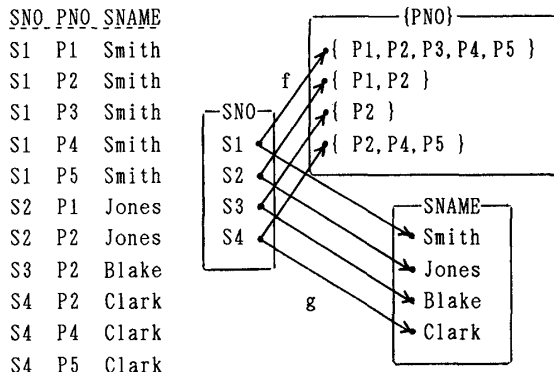


図6：図5の検索結果と割り当て状態

4. RDB言語SQL内でのホスト言語対象の参照

(1) ホスト言語がPL/Iの場合

ホストの対象で参照できるものは、変数のみである。変数の前にはコロンが必要である。しかも、変数は修飾されたり、添え字を付けられたりしてはいけない。集合をサポートしていないため、参照データは atom 単位に限られる。

(2) ホスト言語がSOLの場合

式のレベルでの参照が可能である。ただし、式を区別するために、式をコロンで囲む必要がある。集合情報の参照も可能である。例として、「入力された都市(複数個)にある供給者名を求める」ためのプログラムの一部を図7に示す。この例では、INSET述語内においてSOLの集合変数であるINPSETを参照している。また、PL/Iを用いて書いた同様のプログラムを図8に示す。

```
read(INPSET);
EXEC SQL SELECT SNAME
      INTO :ANSSET
      FROM S
      WHERE CITY IN :INPSET ;
write(ANSSET);
```

図7：SQL言語内からのSOLの対象の参照

```
EXEC SQL DECLARE C1 CURSOR FOR
      SELECT SNAME
      FROM S
      WHERE CITY = :X ;
GET LIST(X);
DO WHILE (X^='');
EXEC SQL OPEN C1;
EXEC SQL FETCH C1 INTO :ANS;
DO WHILE (SQLCODE=0);
PUT LIST(ANS);
EXEC SQL FETCH C1 INTO :ANS;
END;
EXEC SQL CLOSE C1;
GET LIST(X);
END;
```

図8：図7のプログラムをPL/Iで書いた場合

5. 処理系

IBMが提供しているデータベース管理システム(SQL/DS)を使用するため、処理系はIBM4381のCMS上で開発した。SQL言語を使うに当たってはPL/Iが最も制限が少ないという理由により、処理系はPL/Iで記述している(約4500行)。

SOLに埋め込まれたSQL命令は動的SQL文を用いて、動的に実行する。

6. あとがき

SQLのホスト言語にSOLを用いることによって、SQLからSOLの集合が参照でき、SQLの検索結果をSOLの集合と写像に割り当てることができるようになった。その結果、SQLとユーザ間のマン・マシンインタフェースを改善することができた。ただし、現在のところ、次のような問題点が残っている。

- ① SOLに埋め込まれたSQL命令は、動的SQL文を用いて実行しているため、実行可能なSQL命令に多少制限がある。
- ② SQLの対象を基にしてSOLの対象は参照できない。例えば、WHERE節において、表1のSTATUSを引数としてSOLの関数に与えることはできない。今後は引続き、ユーザインタフェースの向上を目的として、SOLの処理系の拡張を行う予定である。

参考文献

- 1) 「データベース言語SQL」 JIS X3005-1987
- 2) 重松、吉見、吉田、「集合指向言語SOLとその言語処理系の開発」情報処理学会論文誌 Vol.30 No.3 (1989)
- 3) IBM SQL/Data System Application Programming for VM/System Product Release 3 Program Number 5748-xxj