

プログラミング言語教育における日本語利用について

梶 田 建 夫

はじめに

日本語によるプログラミング言語が開発用、教育用に開発されている。特にプログラミング教育用、入門用を目的とした日本語プログラム言語が多く発表されている。これは、初期のFORTRAN, BASICなどからプログラミング言語には基本的に英語が利用されており、プログラミングということ以前に英語での表現になじみにくいという問題があったことによる。

プログラミング言語教育において、どのような言語でも文法を教えて、アルゴリズムに沿ったプログラムコードを書かせるというようなことが行われる。しかし、いきなり英語を基本としたコードを書いていくことについては、学生にとってはかなり難しい場合がある。すなわち、コンピュータにやらせたいことと、それをどのように英語のコードで表すかが、なかなか結びつかないという点であり、それをどのように理解させるようにするかが教える側の問題でもある。

例えばある事項を繰り返し行いたい場合、どのようなプログラム言語でも繰り返しをさせるコード体系があるので、これに従えばよいのであるが、それが英語のコードであるとなかなか理解しにくいというようなことなどがある。

このようなことから、コンピュータでやりたいことを日本語で書き下したものと、それをある程度プログラムコードに対比できるようにす

れば、そのコードを編集することによりやりたいことに近づけることは、比較的容易ではないかと思われる。まずは日本語で考え、それをどのようにコードで表示させるかを行い、実際のコードを作成するという手順において、日本語での表現とコードとの対応をどのようにするかということが問題となる。

ただ、日本語でのコード表現に多くの規則があると、逆に日本語での表現が複雑になるという余分な問題が起きる可能性がある。これをさけるためには、できるだけ文でなく簡単な語での表現が利用できるようにするというようなことも考えられる。

ここでは、プログラム言語ではないが、データベースにおける処理言語であるSQLを対象とした教育用システムに日本語を利用することを試みた。したがって、日本語のプログラム言語とは異なり、内容を日本語のコードで書き、それをコンパイルして実行するというのではなく、実行内容を日本語のデータとして作成するというシステムとしている。

データ表現形式としてはXML形式を利用し、これを変換してSQL文を生成させ、実際にデータベースでの実行結果が表示できるようにしている。これを利用することにより、最初からSQL文を考えるのではなく、必要な結果を得るためにはどのようなデータを作成すればよいかを考え、それがどのようなSQL文となるかをチェックし、実際に実行してみて必要な結

果を得られるかどうかを判断することができるようにしている。

1. 日本語によるプログラミングについて

日本語によるプログラミング言語には、すでに「MIND」,「ドリトル」,「なでしこ」のようなものがあり、利用が行われている。これらでは、プログラムをする人になるべくイメージしやすいような日本語を利用し、プログラミングができるように考えられている。「ドリトル」などは小中学生でもプログラムできるような言語体系となっている。これらは、ある程度言語体系が理解できれば、容易にプログラムすることができるが、アルゴリズムをどのように表現するかという問題になると、一般のプログラム言語と同様の問題は生じることとなる。

プログラムコードが書きやすくなるという点では日本語でのプログラム言語は有効であるが、どのようなコード体系が利用しやすいのかは通常のプログラミング言語と同様にそれぞれの特徴を生かせるようなものであるかどうかによる。次に、主な日本語プログラム言語について示す。すべて現時点(2008年9月24日)で利用できるバージョンを対象としている。

(1) ドリトル

ホームページでの説明には以下のように書かれている。

「学校教育(小中高)を中心に、大学生や社会人にも適した入門用のプログラミング言語です。BASICやLOGOといった、30年以上前に設計された言語に代り、最新のオブジェクト指向の考え方でプログラムを学ぶことができるのが特徴です。」

ただ単に日本語でのプログラム言語というだ

けでなく、オブジェクト指向の考えを取り入れているという特徴がある。2001年頃に開発され、標準バージョンはV1.28b(2007年4月25日)となっている¹⁾。

(2) なでしこ

ホームページでの説明には以下のように書かれている。

「なでしこは日本語でプログラミングできる言語です。誰でも使える無料です。ファイルのコピーやバックアップ、Excel/Wordと連携、日々の定型処理に使える手軽な命令が1000以上用意されています。また、これからプログラミングを覚えたいという人にぴったりの手軽な道具です。」

日本語でのプログラミングということのみでなく、Word/Excelとの連携ができるという特徴を持ち、実用性を持たせることを目的として開発されている。

なでしこは、2001年夏に公開された「ひまわり」というプログラム言語の後継プログラミング言語である。利用できるバージョンは1.5071(2008/08/21)である²⁾。

(3) Mind

Scripts Lab. というところで1997年UNIX版として開発されたものである。Windows版の最新はバージョン7.5(2005年)である。これは、コンパイラ形式の日本語プログラミング言語であり、大規模なアプリケーション開発にも対応できるとしている。

これは、30日間の試用期間があるが、有料のソフトである³⁾。

(4) PEN

大阪学院大学情報学部の西田研究室と、大阪

市立大学大学院創造都市研究科の松浦研究室の共同プロジェクトとして開発されているものである。初学者向けのプログラミング学習環境として開発されており、大学入試センターの「情報関係基礎」の入試で用いられているDNCLを使用して、学生に容易にプログラムが理解できるようにされている。利用できるバージョンは1.17p1である⁴⁾。

これら以外にも種々のプログラミング言語が提案されているが、ここでは以上の4言語について、日本語での表現を比較してみる。例題としては以下のような問題をプログラムした結果を比較する。

『パスワードを入力させ、パスワード(PASS)が正しければ「受け付けました」と表示させ、間違っている場合には「違います」と表示し、再度入力させる。3回間違った場合には「受け付けられません」と表示させる。』

ドリトル

ラベル!"パスワード:" 作る 150 45 大きさ -200 200 位置。

入力=フィールド! 作る 180 50 大きさ -50 200 位置。

ラベル!"結果:" 作る 150 45 大きさ -130 100 位置。

結果=フィールド! 作る 220 50 大きさ -50 100 位置。

ラベル!"エラー回数:" 作る 150 45 大きさ -200 0 位置。

エラー=フィールド! 作る 180 50 大きさ -50 0 位置。

ボタン=ボタン! "入力" 作る 200 200 位置。

回数=0。

ボタン:動作=「

P=入力!読む。

「(P) == "PASS"」!なら

「結果!「受けけます」書く。」

そうでなければ

「結果!「違います」書く。

回数=回数+1。

「(回数) == 3」!なら

「結果!「受けられません」書く。

エラー! (回数) 書く。」

そうでなければ

「エラー! (回数)書く。入力!"”書く。」

実行。」

実行。

」。

なでしこ

3回

「パスワードは?」と尋ねる。

パスワードはそれ。

もし、パスワードが「PASS」ならば

「受けけます。」を表示。

抜ける。

違えば

「違います。」を表示。

結果は回数。

もし、結果が3ならば

「受けられません。」を表示。

MIND

メインとは

3を 回数指定し

「パスワードの入力:」を 表示し

文字列入力したものが

「PASS」と 等しい文字列

ならば 「受けます。」を 表示し

終わり

さもなければ 「違います。」を表示して改

行する

つぎに

繰り返す

「制限回数を超えました!受けられません。」を 表示する。

PEN

```
*****
```

```
文字列 str
```

```
整数 i
```

```
iを1から3まで1ずつ増やしながら,
```

```
| 「パスワードを入力してください」を表示する
```

```
| str ← input()
```

```
| もし str = "PASS" ならば
```

```
| | 「受け付けます」を表示する
```

```
| | 繰り返しを抜ける
```

```
| を実行し、そうでなければ
```

```
| | 「違います」を表示する
```

```
| を実行する
```

```
を繰り返す
```

```
もし i = 4 ならば
```

```
| 「受けられません」を表示する
```

```
を実行する
```

```
*****
```

それぞれ、最適なプログラムになっていないかもしれないが、繰り返し部分と、判断の部分がどのようにプログラムされるかは示されている。ただし、ドリトルは画面上での入力であるので、繰り返し部分は含まれていない。

いずれも繰り返し部分については「繰り返す」、判断の部分には「もし～ならば、そうでなければ」というような文が利用されており、初学者でも容易に内容は理解できるようになっている。ただ、書式についてはそれぞれの書き方が用いられており、どれが書きやすいかは一概には言えない。また、実際にプログラムする場合には当然それらの書式を十分理解する必要がある。

3. 学習者の反応

筆者は現在C言語のプログラミングを教育するために、通常のC言語のコードを利用してプ

ログラムを作成する講義を行っている。プログラミング教育において日本語利用がどのように思われるかについて、この講義の受講者に対して講義終了後、以下のようなアンケートを行った。

アンケート

以下のようなプログラムの内容（キーボードから「PASS」が入力されるまで3回繰り返しを行う）がC言語と日本語データで書かれた場合の表示を見て設問に答えてください。

C言語で書いた場合

```
*****
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
main()
```

```
{
```

```
int i;
```

```
char str[20];
```

```
for (i = 1 ; i <= 3; i++) { printf( "パスワードを入力して下さい：" );
```

```
scanf( "%s", str );
```

```
if (strcmp(str,"PASS")==0){ printf( "受け付けます\n");
```

```
break; }
```

```
else{ printf( "違います\n"); }
```

```
}
```

```
return 0;
```

```
}
```

```
*****
```

このプログラム手順を日本語データで表現した場合

```
*****
```

```
<処理>
```

```
<繰り返し 変数名="i" 初期値=1 終値=3 増分=1>
```

```
<表示 内容="パスワード入力してください"/>
```

```
<入力 変数名="str"/>
```

```

<判断>
  <条件式>str="PASS"</条件式>
  <満たす>
    <表示内容="受付けます"/>
    <実行>繰り返しを抜ける</実行>
  </満たす>
  <満たさない>
    <表示内容="違います"/>
  </満たさない>
</判断>
</繰り返し>
</処理>
*****

```

設問1 この内容を理解することについて

(1) C言語で書かれているほうがよい (2) 日本語で書かれているほうがよい (3) どちらでもよい

設問2 プログラム言語に日本語を使うことについて

(1) 英語でよい (2) 日本語のほうがよい (3) どちらでもよい

日本語データでの表現に関する説明を全くしていないが、インデントの表現で意味合いがある程度理解できるのではと判断して行った。しかし、結果としては日本語での表現の説明が十分でなかったためか、C言語のコードが見やすかったという以下のような結果となった。

設問1 (1) 55% (2) 28% (3) 16%

設問2 (1) 50% (2) 25% (3) 25%

ただ、講義での成績が優秀（試験結果が満点）であった3人の学生、すなわち、ある程度プログラムとはどのようなものか理解できたと思われる学生のアンケート結果では、全ての学生が設問1で (2) を選択していたという結果も得られた。

4. SQL表示システムについて

SQL (Structured English Query Language) は関係データベースにおいてデータの定義、操作を行う言語であり、英語を基本とした言語である。データベース教育においてもこれを取り扱う必要があるが、一種のプログラミング言語であるので、学習者にとっては他のプログラム言語と同様な難しさを感じさせる部分がある。

ただ、単純なものであれば3行程度の表現で書き下すことができるので、プログラムをやるということよりも、必要な情報を得るためにはいかにSQL文でそれを表現するかということが問題となる。ここでは、SQL文の表現を理解させるために、SQL文を作り出すための日本語データ形式を作成することを試みている。

SQL文にも他のプログラミング言語同様に、条件分岐、判断のような部分が含まれており、これをどのように表現すればよいかについては初学者にとって理解しにくい部分でもある。したがって、このような部分で日本語を利用したデータ形式で表現し、これをSQL文に変換するシステムとしている。

システムの開発環境としては、Web上でのアプリケーションの開発に必要なフリーソフトをパッケージ化したXAMPP (Version2.5) というものを利用した。これは、サーバーソフトとしてのApache、データベースソフトのMySQL、プログラミング言語であるPHP、Perlなどのソフトよりなるものである。ここでは、データベースとしてはMySQLを対象とし、データの処理とSQL文の実行にはPHPを用いている。

5. データ構造の作成

SQL文を作る場合、プログラミングと同様に、何がしたいか、どのような結果が得たいかということをまず決める必要がある。これが決まれば、それをSQL文に書き下すわけであるが、直接SQL文を書くのではなく、その手順を書き下して、それに沿ってSQL文を作っていくというのが普通である。その手順を書き下した段階で、その手順をどのようにSQL文で表現するかが問題となる。ここでは、この部分で日本語を利用したデータ形式に分解して表現することができるようにしている。

データ構造の定義はSQL文に対応して表1から表5のように作成した。SQL文はデータ操作、データ定義、データ制御の部分に分けられるが、ここではデータ制御以外の部分をまとめて取り扱っている。厳密なスキーマの定義は行っていないが、処理に必要な項目についてはすべてXMLの整形形式で表示するようにしている。

SQL文によるデータベース作成、データ入力処理では表1、2、3の要素と属性を用いて作

表1 データベースに関するデータ項目

要素	属性
データベース	名前：データベース名 操作：作成、削除、選択、一覧

成される。

例えば「daigaku」という名前のデータベースを作成し、作成されたデータベースを表示させる場合のデータは表1より次のようにする。

```
<データベース 名前="daigaku" 操作="作成"/>
<データベース 操作="一覧"/>
```

また、このデータベースに「gakusei」というテーブルを作成し、そこにデータを入力するには、表2、表3の要素、属性を用いて作成される。

データ構造の定義部分の例は次のようである。

```
<テーブル 名前="gakusei" 操作="作成">
  <カラム 名前="gakuseki" 型="固定長文字型"
    長さ="10"/>
  <カラム 名前="simei" 型="可変長文字型" 長
    さ="40"/>
</テーブル>
```

データの入力部分の例は次のようである。

```
<データ入力>
  <テーブル 名前="gakusei"/>
  <カラム 名前="gakuseki,simei">
    <データ>"n001","伊藤太郎"</データ>
    <データ>"n002","上田次郎"</データ>
    <データ>"n010","島田俊夫"</データ>
  </カラム>
```

表2 テーブルに関するデータ項目

要素	属性	要素	属性
テーブル	名前：テーブル名 操作：作成、削除、選択、一覧	カラム	名前：カラム名 型：整数型、固定長文字型、可 変長文字列型等 長さ、キー、デフォルト
ビューテーブル	名前：テーブル名 操作：作成、削除		

表 3 データ操作に関するデータ項目

要素	要素	属性	要素
データ入力	テーブル	名前：テーブル名	
	カラム	名前：カラム名	データ
データ削除	テーブル	テーブル名	
	条件	カラム名 式：等しい, 等しくない, 未満, より大きい等 値	
データ更新	テーブル	名前：テーブル名	
	カラム	更新	
	条件	名前：カラム名 式：等しい, 等しくない, 未満, より大きい等 値	副参照

表 4 データ参照に関するデータ項目

要素	要素	属性	要素	属性
データ参照	テーブル	名前：テーブル名	結合	種類：内部, 左外部等
	カラム	別名	場合	式：等しい, 等しくない, 未満 値, 真, 偽
			副参照	
	条件	カラム名 式：等しい, 等しくない, 未満 値	副参照	
	条件式			
	グループ	カラム名	条件	
	並び替え	カラム名 順序：昇順, 降順		

表 5 「副参照」要素に関するデータ項目

要素	要素	属性
副参照	テーブル	名前：テーブル名
	カラム	別名
	条件	名前：カラム名 式:等しい, 等しくない, 未満, より大きい等 値

</データ入力>

このようにXMLデータ形式にすることにより、データの内容が日本語で表現できるため、具体的な処理の内容が容易に理解できることとなる。

6. システムの概要

実際のシステムの利用は、表1から表5までの要素、属性を利用してデータを作成するが、作成する場合の基本的手順として以下のように考える。

- (1) 表示させる列を決める
- (2) そのデータがどの表に含まれているかを決める
- (3) どのような条件を用いるかを決める

例えば「gakuseiというテーブルからgakusekiがn006以降の学生のgakusekiとsimeiよりなる表を作成したい」というような場合、上の手順に従って以下のように書くことができる。

- (1) 求めるテーブルのカラム名はgakusekiとsimeiである。

- (2) 求めるデータはgakuseiというテーブルにある

- (3) 条件はgakusekiがn006以降である
これよりデータは次のように作成する。

<データ参照>

<カラム>gakuseki,simei</カラム>

<テーブル 名前="gakusei"/>

<条件 カラム名="gakuseki" 式="以上" 値="n006"/>

</データ参照>

システムの初期画面は図1に示すデータ入力画面であり、画面の下にある「データチェック」のボタンを押すとXMLEditorによる構文のチェックが行われる。また、「SQL文」のボタンを押すと、SQL文が表示され、「実行」ボタンを押すとそのSQL文による実行結果が表示されるようになっている。

実際に前のデータを入力し、実行する場合は、データベース使用の宣言部分を含む必要があるので図1のようなデータ表現となる。「実行」タグで囲まれた部分がSQL文の1行分に対応

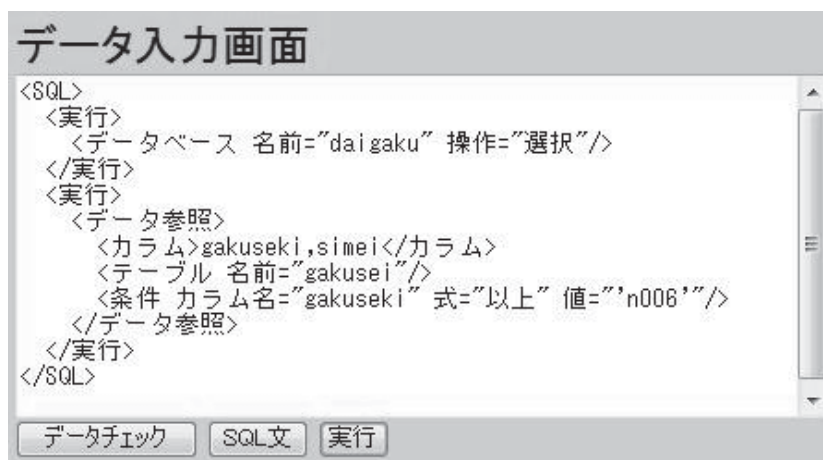


図1 データ入力画面

作成されたSQL文

```
use daigaku
select gakuseki,simei
from gakusei
where gakuseki>='n006'
```

図2 結果SQL文

していることになる。

ボタン「SQL文」を押した結果は図2のようになる。

ボタン「実行」を押した結果は図3のようになる。

実行結果

gakuseki	simei
n006	木村洋二
n007	久保久司
n008	近藤四郎
n009	佐藤郁夫
n010	島田俊夫

図3 実行結果

このようにシステムでは、XML形式で書かれたデータを「SQL文」ボタンが押されると、データがPHPで処理されてSQL文が作成され、「実行」ボタンが押されると、作成されたSQL文がMySQLで処理されるようになっている。

7. SQLコマンドとの対応

SQLコマンドはすべて英語であり、ここではこれを表1から表5に示した要素、属性で置

き換えることによりSQL文の作成、実行ができるようにしている。上のような単純な問い合わせであれば、データから「SELECT カラム名 FROM データベース名 WHERE 条件」のように置き換えるのみでよいが、より複雑なSQL文に対応させるために表6のような処理内容の表示ができるようにしている。

また、データの参照以外にデータ入力、更新、生成、削除というようなデータ定義を含むデータ操作部分に関しては表7のようである。

これらの表より具体的な条件でのSQL文作成と実行は次のように行われる。

テーブルsyukketuの作成とデータ入力は図4に示すように行う。また、テーブルkougiの作成とデータ入力は図5に示すように行う。

テーブルgakusei, syukketu, kougiを利用したデータの抽出例として次の条件でデータを作成し実行する。

* 表示させるカラム名はgakuseki:別名(学籍番号), simei:別名(氏名), hiduke:別名(日付), tantou:別名(担当者)とする。

* テーブルgakuseiとテーブルsyukketuをカラムgakusekiで結合し、さらにテーブルkougiをカラムkougiidで結合する。

* 条件はkouginameが「経営情報処理」である。

* 結果をカラムtantouで昇順、カラムhidukeで昇順、カラムgakusekiで昇順に並び替える。

この場合のデータは次のようになる。

<データ参照>

<カラム 別名="学籍番号">g.gakuseki</カラム>

<カラム 別名="氏名">simei</カラム>

<カラム 別名="日付">hiduke</カラム>

<カラム 別名="担当者">tantou</カラム>

<テーブル 名前="gakusei g">

<結合 種類="内部" テーブル名="syukketu s">g.gakuseki = s.gakuseki</結合>

表 6 データ参照に関するデータ表示

処理内容	SQL コマンド例	対応するデータ表示
表示させるカラムを指定する	SELECT カラム名	表 4 カラム
表示させるカラムに別名を付ける	SELECT カラム名 AS 別名	表 4 カラム
カラム値の重複を除く	SELECT DISTINCT カラム名	表 4 カラム
カラムまたはカラム間の関数等を用いた演算を行う	SELECT カラム名 *1.05, SUM (カラム名), AVG (カラム名) 等	表 4 カラム
カラム値により条件分岐させる	SELECT CASE WHEN カラム値の条件 THEN 条件を満たす ELSE 条件を満たさない	表 4 場合
選択するテーブルを指定する	FROM テーブル名	表 4 テーブル
複数テーブルを結合する	FROM テーブル名 INNER JOIN テーブル名 ON 結合条件	表 4 結合
カラム値に関する参照条件の表示	WHERE カラム名 関係演算 値	表 4 条件
副参照の条件のもとでのカラム値に関する参照条件の表示	WHERE カラム名 (副参照)	表 4 副参照
複数条件からの条件式作成	WHERE 条件 OR 条件, 条件 AND 条件	表 4 条件式
副参照	SELECT カラム名 FROM データベース名 WHERE 条件	表 5 副参照
カラムを指定してグループ化をする	GROUP BY カラム名 HAVING 条件, 条件式	表 4 グループ
カラムまたはカラム間の演算結果で並び替えを行う	ORDER BY カラム名, カラム間の演算, 昇順, 降順	表 4 並び替え

表 7 データ構造に関するデータ表示

処理内容	SQL コマンド例	対応するデータ表示
データベース作成	CREATE データベース名	表 1 データベース
テーブル作成	CREATE テーブル名	表 2 テーブル
参照条件により生じたテーブルを仮想テーブルとして作成する	CREATE VIEW テーブル名 AS	表 2 テーブル
データを入力する	INSERT INTO カラム名 値, 条件, 副参照	表 3 データ入力
データを更新する	UPDATE カラム名 SET 値, 条件, 副参照	表 3 データ更新
データを削除する	DELETE テーブル名, カラム名, 条件, 副参照	表 3 データ削除
データベースを削除する	DROP データベース名	
テーブルを削除する	DROP テーブル名	

データ入力画面

```

<実行>
<テーブル 名前="syukketu" 操作="作成">
  <カラム 名前="gakuseki" 型="固定長文字型" 長さ="10"/>
  <カラム 名前="hiduke" 型="日付型"/>
  <カラム 名前="jigen" 型="整数型"/>
  <カラム 名前="kougiid" 型="固定長文字型" 長さ="10"/>
</テーブル>
</実行>
<実行>
<データ入力>
  <テーブル 名前="syukketu"/>
  <カラム 名前="gakuseki,hiduke,jigen,kougiid">
    <データ>"n001","2008-07-01",1,"k002"</データ>
    <データ>"n001","2008-07-01",2,"k005"</データ>
    <データ>"n001","2008-07-01",3,"k008"</データ>
    <データ>"n002","2008-07-01",2,"k005"</データ>
    <データ>"n002","2008-07-01",3,"k008"</データ>
    <データ>"n004","2008-07-01",1,"k002"</データ>
    <データ>"n004","2008-07-01",3,"k004"</データ>
    <データ>"n004","2008-07-01",4,"k006"</データ>
  </カラム>
</データ入力>

```

データチェック SQL文 実行

図4 テーブル syukketu 作成画面

データ入力画面

```

<実行>
<テーブル 名前="kougi" 操作="作成">
  <カラム 名前="kougiid" 型="固定長文字型" 長さ="10"/>
  <カラム 名前="kouginame" 型="可変長文字型" 長さ="40"/>
  <カラム 名前="tantou" 型="可変長文字型" 長さ="40"/>
</テーブル>
</実行>
<実行>
<データ入力>
  <テーブル 名前="kougi"/>
  <カラム 名前="kougiid,kouginame,tantou">
    <データ>"k001","経営学総論","愛知太郎"</データ>
    <データ>"k002","商学総論","愛知次郎"</データ>
    <データ>"k003","経済学","愛知三郎"</データ>
    <データ>"k004","情報処理演習","長野太郎"</データ>
    <データ>"k005","経営情報処理","長野次郎"</データ>
    <データ>"k006","統計学","長野三郎"</データ>
    <データ>"k007","経営学総論","福井太郎"</データ>
    <データ>"k008","商学総論","福井次郎"</データ>
    <データ>"k009","情報処理演習","福井三郎"</データ>
  </カラム>
</データ入力>

```

データチェック SQL文 実行

図5 テーブル kougi 作成画面

```

<結合 種類="内部" テーブル名="kougik">s.kougiid=k.kougiid</結合>
</テーブル>
<条件 カラム名="kouginame" 式="等しい" 値="経営情報処理"/>
<並べ替え カラム名="tantou" 順序="昇順"/>
<並べ替え カラム名="hiduke" 順序="昇順"/>
<並べ替え カラム名="g.gakuseki" 順序="昇順"/>
</データ参照>

```

実行結果は図6のようになる。

8. エラーのチェックについて

データの入力ミス、すなわちデータがXML形式に対応しているかどうかについては、XMLEditorを用いて行う。これはデータをIE (Internet Explorer) でXML形式のファイルとして表示させることにより行うことができる。

SQLでのエラーについては、エラーがある場合にはMySQLからエラーメッセージが返されてくるので、これをPHPで処理して画面に表示させるようにしている。

作成されたSQL文

```

use daigaku
select g.gakuseki as 学籍番号,simei as 氏名,hiduke as 日付,tantou as 担当者
from (gakusei g inner join syukketu s on g.gakuseki=s.gakuseki) inner join kougik k on s.kougiid=k.kougiid
where kouginame='経営情報処理'
order by tantou asc,hiduke asc,g.gakuseki asc

```

実行結果

学籍番号	氏名	日付	担当者
n007	久保久司	2008-07-02	石川太郎
n001	伊藤太郎	2008-07-03	石川太郎
n003	遠藤憲治	2008-07-03	石川太郎
n010	島田俊夫	2008-07-03	石川太郎
n001	伊藤太郎	2008-07-01	長野次郎
n002	上田次郎	2008-07-01	長野次郎
n006	木村洋二	2008-07-01	長野次郎
n008	近藤四郎	2008-07-01	長野次郎
n006	木村洋二	2008-07-02	長野次郎
n002	上田次郎	2008-07-03	長野次郎
n004	大田啓祐	2008-07-03	長野次郎
n009	佐藤郁夫	2008-07-03	長野次郎

図6 具体例のSQL文と実行結果

おわりに

このシステムでは、一般的にデータの作成量が多くなるが、目的のデータを得るための手順を書き下し、それに沿って日本語を用いたデータをXML形式で書いていくことにより自動的にSQL文が作成できるようになっている。したがって、学習者はまず、目的のデータを抽出するための手順を考え、その手順を分解し、並び替え、それをここでのデータ形式で記述することにより目的の結果を得るためのSQL文がどのように表示されるかが確認できる。また、実際に作成されたSQLによる実行結果を表示させ、確認することができる。これにより、基本的なデータ表示形式が理解できれば、SQL文の学習に利用できるとともに、SQL文の構造を理解しやすくなると思われる。

SQL作成のための支援システムとして、XML形式のデータを作成し、SQLの表示、実行を行うシステムについて示した。これまでも教材としてのXMLデータ形式の利用というこ

とを行ってきたが、その一環としてこのようなシステムの作成を行った。

種々の教材がデータベース化され、利用されることが必要であり、今後ともこのようなデータ形式の適用を考えていく必要があると思われる。

参考文献

- 1) ドリトル : <http://dolittle.eplang.jp/> 2008.9.24
- 2) なでしこ : <http://nadesi.com/index.php> 2008.9.24
- 3) Mind : <http://www.scripts-lab.co.jp/mind/whatsmind.html> 2008.9.24
- 4) PEN : <http://www.media.osaka-cu.ac.jp/PEN/> 2008.9.24
- 5) 兼宗進, 教育用プログラミング言語と授業利用: 教育用プログラミング言語の動向, 情報処理, Vol. 48 No. 6 2007.6, pp. 589-593
- 6) 奥村晴彦, 教育用プログラミング言語と授業利用: 情報科学教育への利用, 情報処理, Vol. 48 No. 6 2007.6, pp. 598-601
- 7) クジラ飛行機, 日本語プログラム言語「なでしこ」公式バイブル, ソシム, 2008年6月