

関係データベースの XML による構造化ビュー

1Z-02

金澤 克 三浦 孝夫
法政大学工学部電気電子工学科

塩谷 勇
産能大学経営情報学部

概要

本論文では、関係データベースに格納されている関係データから SQL を介して構造化ビューを作りあげ、XML 文書として取り出すシステムについて述べる。

1 動機と解法

インターネット上で関係データベースへアクセスしてデータを取り出す際、保存データは単純値であるために構造を持たない。そのため取り出したデータはテキスト形式で取り出されるだけで、データが多くなればなるほど見にくいものとなってしまふ。そこで本研究では、データベース内での単純値を複合値へ変える事によって構造化をするのである。この単純値から複合値へと変え構造表現を実現するための方式として、SQL の GROUP-BY を使用することを提案する。これにより構造化されたデータは、構造表現そのものである XML 文書として、SQL システム系の GUI から独立してブラウザに表示可能である。

2 利用方法

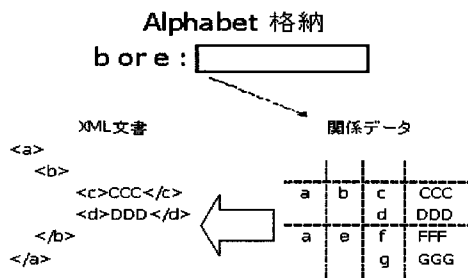


図 1: 関係データの入出力例

関係データベースに保存されているデータは、単純値からなる単純表として保存されている。この単純値を構造化することで、親子関係や前後関係など細かな部分を表現する事が出来る。また、必要とするデータのみの構造表現も可能である。

Structural View of Relations by XML
Masaru Kanazawa, Takao Miura, Isamu Shioya
Hosei University, Dept. of Elec. and Elec. Eng.
Kajino-cho 3-7-2, Koganei, Tokyo, JAPAN
Sanno University, Dept. of Management. and Info.
Kamikasuya 1563, Isehara, Kanagawa, JAPAN

図 1 にそのモデルを示す。ユーザは直接データベースに格納されている全データを見る事は出来ないが、用意されたフォームによってどのようなデータが格納されていて、どのデータに関しての検索が出来るのかわかる。あとは指定フォームに構造化して得たいデータを入力する。これによってユーザはブラウザを通して入力データの構造化された詳しい情報を得る事が可能となる。

3 システム構成

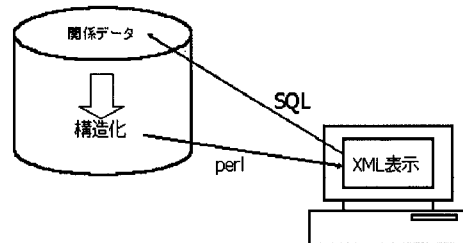


図 2: システムの流れ

システムの流れは図 2 のようになる。図を見てもわかるように、まず最初にフォームに検索したい情報を入力する。ここで入力された情報によって SQL が生成され、この SQL によって関係データベースへアクセスする。次にデータベースに格納されている関係データの中で、SQL によって要求されたデータのみを構造化を実現し、タグ付けを行った後ブラウザに XML 文書として表示される。

3.1 フォーム

HTML によって作成されたフォームに、ユーザが知りたい情報を入力する。ここではテキストボックスを入力フォームとして使用しているため、ユーザは格納されているデータをあらかじめ知っておく必要がある。

3.2 SQL 操作による構造化

フォームに入力されたユーザ指定のデータに関する親子関係を SQL を介して構造化して、XML 文書として取り出す。N 個あるフィールドの中で、フィールド A に属する値に対して構造化したい場合に介する SQL として select フィールド [A+I] from テーブル名 where フィールド A = '検索要素'

group by フィールド[A+1] なる SQL を用意して、これに先のフォームで入力した検索要素を入れて、第1SQL を生成してデータベースへアクセスする。関係データベースとのインタフェースとして Perl を用いる。そして生成された SQL の結果を pg_exec() で返して、その結果を\$result[i](i=0..)に格納する。\$result[0] から select フィールド[A+2] from テーブル名 where フィールド A = '検索要素' and フィールド[A+1] = '\$result[0]' group by フィールド[A+2] へ入れて第2のSQL を生成する。これを繰り返し、第3SQL、第4SQL と生成していき、select フィールド[N-1] まで繰り返す。ここで、フィールド[N-1]とフィールドN は <N-1の値>Nの値 </N-1の値> のような関係が成り立つように構造化するので、最終生成 SQL は select フィールド[N-1], フィールドN と2つ同時に取り出す。これで、検索要素の子の子の...と構造化する事が出来る。そして最初の分岐地点とも言うべき\$result[1], \$result[2]... も上記と同じように実行して繰り返していき、検索要素全体の構造化を実現する。生成するSQLの数は構造化の深さに比例することがわかる。

3.3 XML 文書生成

生成されたSQL結果は perl によってタグ付けされ、XML 文書としての構造化が可能となる。そして CGI として全てが構造化されたXML文書の出力を、検索フォームのあるブラウザで表示する。

4 実験

prefecture	city	section	genre	element
chiba	narashino	food	beef_bowl	MATSUYA
chiba	narashino	food	family_restaurant	SAISERIA
chiba	narashino	food	family_restaurant	GUST
chiba	narashino	food	noodle	KATSURYOKU
chiba	narashino	food	noodle	NARITAKE
chiba	narashino	shopping	depart	EXSOTTIC
chiba	narashino	shopping	depart	PARCO
chiba	narashino	amusement	karaoke	SHIDAX
chiba	narashino	amusement	rental	TSUTAYA
chiba	narashino	amusement	pachinko	GIGA
chiba	funabashi	food	beef_bowl	MATSUYA
chiba	funabashi	food	beef_bowl	YOSHINOYA
chiba	funabashi	food	meat	ANRAKUTEI
chiba	funabashi	food	sushi	KAPPASUSHI
chiba	funabashi	shopping	book	KUMAZAWA

図 3: 関係データ

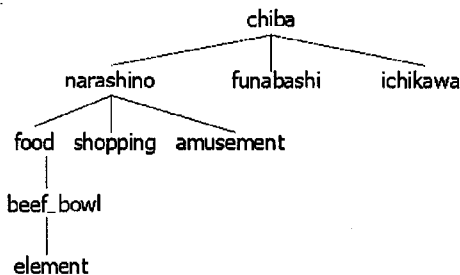


図 4: DTD グラフ

本研究では図3に示す関係データベースに格納されている千葉県の店情報に対して、ある市の店情報のみの構造化を実現

し、XML 文書としてブラウザ表示を行う。そのためにはまず、用意したフォームに検索したい市名を入力する。入力された市名は実現システムに渡され、それによって生成されたSQL を介して構造化を図る。これにタグ付けを行って出来たXML文書をブラウザ表示する。またこのXML文書は、図3を元に生成されるDTDグラフによって検証する事が出来る(図4)。実験には関係データベースとして PostgreSQL を、結果表示をするためのブラウザにIE6を使用する。また、CGI プログラムは文字列解析が比較的簡単に行える Perl を利用する。ここでは検索フォームに入力する市を'narashino'として、得られた結果を図5に示す。市名'narashino'の店情報のみ構造化されており、他の市の店情報は取り出されていない。この結果によりSQLを介しての構造化が実現している事がわかる。

```

<?xml version="1.0" ?>
- <chiba>
  - <narashino>
    - <food>
      <beef_bowl> MATSUYA </beef_bowl>
      <family_restaurant> GUST </family_restaurant>
      <family_restaurant> SAISERIA </family_restaurant>
      <noodle> KATSURYOKU </noodle>
      <noodle> NARITAKE </noodle>
    </food>
    - <shopping>
      <depart> EXSOTTIC </depart>
      <depart> PARCO </depart>
    </shopping>
    - <amusement>
      <karaoke> SHIDAX </karaoke>
      <pachinko> GIGA </pachinko>
      <rental> TSUTAYA </rental>
    </amusement>
  </narashino>
</chiba>
  
```

図 5: 検索結果

5 評価

実験では市名が'narashino'の検索を行って構造化し、XML 文書として取り出す事が出来た。格納されている関係データの中の他の市にも同様な検索をかけてみたところ、'narashino'と同様に検索した市の店情報のみをXML文書として取り出す事が出来た。これによってSQLを介して関係データベースへアクセスをし、データベース内で構造化処理されている事がわかる。しかし、今回の実験でSQLを介するためには、ユーザがどんなデータが関係データベースに格納されているかを知っておく必要がある。そのためにテキストボックスの代わりに、ドロップダウンメニューやラジオボタンを利用して、一般化を図ればよい。

6 結び

本研究ではXML文書として表示するまでだったが、表示されたXML文書を元に、マルチメディアデータを扱うためにXSLなどのスタイルシートを使用することで、様々なデータ表示が可能となると推測できる。

参考文献

- [1] 石井達夫: PostgreSQL 完全攻略ガイド, 技術評論社 (2001)
- [2] J.Shanmugasundaram, R.Krishnamurthy:A General Technique for Querying XML Documents using a Relational Database System, SIGMOD Record,pp.20-26 (2001)