

半構造データを扱うためのDBMSのデータベーススキーマの設計と実装

4 X - 4

原 圭志

山口大学大学院教育学研究科

中田 充

山口大学教育学部

1 はじめに

近年、データベースの応用分野が広まり、より多くの分野で利用されている。これらの分野で扱われるデータは、統一した構造を持たない等の特徴を持つ半構造データであることが多い。当研究室でも、半構造データを管理するためのデータモデルとしてDREAMモデル[1]を提案している。DREAMモデルはデータの実体を格納するデータエレメント、属性を表す名前付きエレメント、対象物の一側面を表現する視点、一つの対象物を表すオブジェクト、オブジェクトの集合であるバンドルの5つ要素を持ち、これらをデータベースエレメントと呼ぶ。

DREAMモデルはスキーマを定義することなくデータを格納することが可能であるが、大量のデータに対して問い合わせを行う段階では、データベースの構造を知る必要がある。そのためスキーマ情報に相当するシェイプ及びシェイプグラフ[2]と呼ばれる構造を提供している。シェイプの基本単位であるシェイプエントリは、名前付きエレメントの構造を表す情報であり、(id, name, DT)の3つ組で表される。ここで、idは識別子、nameは名前付きエレメントの名前、DTは名前付きエレメントに属するデータエレメントの持つデータ値のデータ型の集合である。シェイプはバンドル、オブジェクト、視点がどのような属性を持つかを表す情報であり、(id, name, S)の3つ組で表される。ここで、idは識別子、nameは対応するデータベースエレメントの名前、Sはそれに属している名前付きエレメントのシェイプエントリの集合である。例えば、名前付きエレメント (ne1, "A", {1}), (ne2, "B", {"xxx"}), (ne3, "B", {2,4})のシェイプエントリは、(se1, "A", {int}), (se2, "B", {string}), (se3, "B", {float})になる。また上の3つの名前付きエレメントを要素として持つ視点 (pe1, "D", {ne1, ne2, ne3})のシェイプは (s1, "D", {se1, se4})である。ここでse4は、se2とse3が同じ名前をもっているため一つ

にまとめた新たなシェイプエントリであり、(se4, "B", {float, string})である。シェイプグラフはバンドル、オブジェクトについてそれらがどのような構造のオブジェクト及び視点を持っているかを表した情報である。例えば、pe1, pe2の2つの視点を持つオブジェクト (obj1, "E", {pe1, pe2})で、pe1, pe2, obj1のシェイプがそれぞれspe1, spe2, sobj1とすると、このオブジェクトのシェイプグラフは、(osg1, sobj1, {spe1, spe2})である。ここでosg1は識別子である。

シェイプ、シェイプグラフは、データエレメントの挿入、削除、更新により変化する。シェイプ、シェイプグラフを生成する最も単純な方法は、データベースエレメントの操作のたびに一からすべてを生成する方法である。本研究では、シェイプ、シェイプグラフを導出するためシステムの設計と実装について述べる。以下、2節でシェイプを導出するためのシステムの設計と実装について述べ、3節でその評価について述べる。最後に4節でまとめる。

2 設計と実装

本システムの実装は、オブジェクトリレーショナルDBMS UniSQL release5.0を用いる。最初にUniSQLのデータモデルでDREAMのデータベースエレメントとシェイプを表現し、メソッドを用いて演算ならびにシェイプの作成機能を実装する。開発環境はCompaq ProLiant ML350 server(CPU Pentium III 600MHz, 256MBメモリ)、OSはRedHat Linux release6.1jでありプログラム言語はCを用いた。

名前付きエレメントなどの各データベースエレメント、シェイプ、シェイプエントリ、シェイプグラフは、各々一つのクラスで表現した。シェイプを表すクラスshapeは、属性nameと属性Sを持つ。属性nameはそのシェイプに対応するデータベースエレメントの名前で、属性Sはそのデータベースエレメントに属する名前付きエレメントに対応するシェイプエントリの識別子の

集合である。シェイプエントリを表すクラス `shape_entry` は、属性 `name` と属性 `S` を持つ。属性 `name` は対応する名前付きエレメントの名前で、属性 `S` はその名前付きエレメントが持っているデータエレメントのデータ型、またはオブジェクトのシェイプの集合である。シェイプは、データベースエレメントの操作に対応して変化するので、データベースエレメントとシェイプとの関係を管理するクラスを作成した。クラス `shape_administration` は、シェイプとバンドル、オブジェクト、視点の対応を表すクラスで、属性 `sid` と属性 `ID` を持つ。属性 `sid` はシェイプの識別子、属性 `ID` はそのシェイプに対応するデータベースエレメントの識別子である。同様にシェイプエントリと名前付きエレメントの対応を管理するクラス `se_administration` を定義した。

次に、すでに挿入されているデータベースエレメントからシェイプ、シェイプエントリ、及びシェイプグラフを導出し、シェイプ管理クラスに対応する情報を格納する関数を作成した。最後にそれらの関数を実行しデータベースをコミットするメソッドを作成した。

3 評価

評価は、山口大学のwebページデータベースを作成し、そのシェイプを導出しコミットが終了するまでの時間を測定することで行った。ここでは、1webページを1つのオブジェクトとして格納した。オブジェクトは1つの視点を持ち、そのページのリンクの数だけ名前付きエレメントを持つ構造になっている。

まず、データベースエレメントを挿入する時間を計測した。オブジェクトの数が200で、それに付随する視点、名前付きエレメントなども含めて合計の数が864のデータベースエレメントの挿入時間は5.05(sec)であった。これに対して、シェイプを導出するのに必要な時間を測定した。導出されるシェイプの数は400、シェイプエントリの数は764でシェイプ導出時間は10.4(sec)であった。データベースエレメントの挿入時間と比較してシェイプの導出時間は問題ない時間であると思われる。

次に、オブジェクトの数によるシェイプ導出時間の变化を測定した。表1は、オブジェクトの数によってシェイプの導出時間の变化の様子を示している。表よりオブジェクトの数に比例してシェイプの導出時間も

増えていることがわかる。シェイプはデータベースエレメントに応じて変化するため、データベースエレメントの挿入、削除、更新のたびに再構成する必要がある、オブジェクトの増加に応じて処理時間が増大するのは問題である。そこで、データベースエレメントの操作時に操作情報を記録し、必要のあるシェイプのみを作成、再構成する方式で実装を行ったが、現在は処理速度を大幅に短くすることは出来ていない。

オブジェクト	シェイプ導出時間(sec)
200	10.4
400	13.71
600	31.78
800	46.86
1000	62.27

表1 オブジェクトの数によるシェイプ導出時間

4 まとめ

本論文では、シェイプの導出するシステムの設計と実装を行ない、その評価を行った。その結果、オブジェクトの数が少ない時点では、ほぼ問題なく処理が行えるが、オブジェクトの数が多くなるとシェイプの更新に時間がかかりすぎるという問題が明らかになった。そこで、データベースエレメントの操作情報を記録しておき、必要なシェイプのみの作成、再構成を行うような方式を採用したが、問題点は解決できなかった。今後の課題としては、システムの実装方法を見直してシェイプの導出時間を短縮すること、シェイプを利用するためのGUIの設計・実装などがあげられる。

謝辞

本研究は、一部、文部省科学研究費奨励(A)(課題番号12780313)による。

参考文献

- [1] Nakata, M., Hochin, T. : "Bottom-up Scientific Database Based on Sets and Their Top-down Usage", Proc. of Int'l Database Engineering & Applications Symposium 97, pp.171-179 (1997)
- [2] 中田充: "半構造データベーススキーマの一表現法式とその管理法" 電子情報通信学会技術研報告, Vol.100, No.416, pp.31-38(2000)