

# インメモリデータ処理を用いた第三正規形を主体とする 業務システム実装法の提案

中西義尚<sup>†</sup> 川上拓也<sup>‡</sup> 金田重郎<sup>‡</sup>

同志社大学工学部<sup>†</sup> 同志社大学大学院工学研究科<sup>‡</sup>

## 1. はじめに

業務システムの設計では、概念クラス図、第三正規形の論理スキーマ、非第三正規形を含む物理スキーマ、と3種のデータ構造に関するドキュメントを作成せねばならず、一種の冗長性を持っている。

物理設計段階で第三正規形が崩れる大きな原因は、データを管理しているDBMSにおいて、FK(外部キー)を用いた結合処理のレスポンスが十分ではない点にある。しかし、状況は変わりつつある。64bitOSの登場とDRAM低価格化により、大容量メモリが容易に手に入るようになっている。

そこで、本稿では、1)データベースの全レコードをインメモリ・オブジェクトとして実現し、2)結合処理をポインタアクセスによって実現する業務システム構造を提案する。これにより、概念クラスからDBMSリレーションまでを単一構造により完結し、設計・保守を容易化する。性能評価実験の結果、固定資産税計算モデルで、固定資産数100万件として、HDD上のDBMSだけの場合に比較して、32倍の性能向上を確認できた。

## 2. 提案手法

### 2.1 基本構成

業務システムのDBMSアクセスの9割程度はRead動作であり、Create, Updateは少ない。Read動作を高速化すれば、性能は大幅に向上すると考え、本稿では以下の構成を検討する。

#### 【オブジェクトモデルの単一化】

図1に示すように、第三正規形を意識して概念ER図を作成する。そして、そのままメインオブジェクト、DBスキーマとする。概念設計段階から第三正規形とする必要があり、概念データモデリングでは、静的モデル表現を(本質は変わらないが)多少、変化させる必要がある[2]。

#### 【全オブジェクトのインメモリ化】

全レコードをインメモリ・オブジェクト化する。これにより、Readを高速化する。一方データ保存・検索・整合性管理は、DBMSに任せる。

#### 【ポインタアクセスによる高速化】

Modelオブジェクト群は第三正規形であり、Viewオブジェクト属性値を取得のために、外部キー(FK)を利用した結合処理が頻発する。次節で述べるように、FKアクセスにはポインタを併用して、Read高速化を図る。

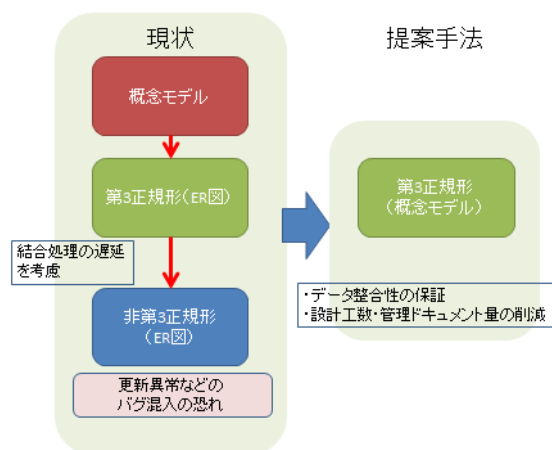


図1 設計モデル

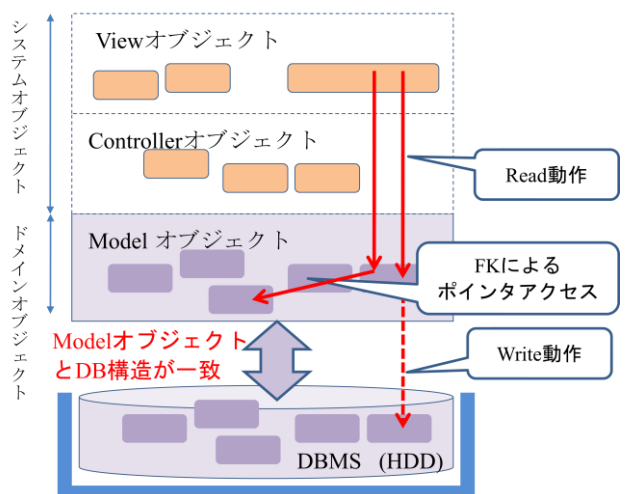


図2 MVCモデルと提案手法

An Approach of Implementation Method Based on Third Normal Form with In-Memory Data Processing for Business System

<sup>†</sup>Yoshinao Nakanishi, Faculty of Engineering, Doshisha University

<sup>‡</sup>Takuya Kawakami, Shigeo Kaneda, Graduate School of Engineering, Doshisha University

以上の構成によって、安価なフリーDBMS と DRAM で一定の性能を確保することを狙いとしている。ターゲットは比較的小規模のシステムである。概念 ER 図とデータベースの一致が可能なのは、小規模システムに限定されるからである。尚、大規模でミッションクリティカルなシステムは、IMDB の利用が妥当であると考えられる。

## 2.2 オブジェクトのポインタ管理

全レコード（オブジェクト）が第三正規形であることと、インメモリ化しているため、オブジェクト間のデータ整合性（更新異常等）の担保は容易であり、コーディング量も減少すると思われる。

反面、View オブジェクトから、Model オブジェクトへの FK を用いたアクセスが頻発する。この性能が、システム全体の性能のボトルネックとなる。そこで、ターゲットオブジェクトのポインタを属性値としてオブジェクト内に記憶しておく、値を得るトリガ関数は、このアドレスを利用して、属性値の取得を行う。

図 3 は具体的なイメージ例である。申請者 ID (FK) は、申請（帳票）に唯一、必然的に記入されていなければならないから、申請 ID に対して、申請者 ID は関数従属である。ただし、申請者氏名や申請者の年齢は、第三正規形ではない。しかし、アプリケーションとしては必要であるから、帳票オブジェクトに属性として記載されている。このような場合は、トリガによって自動的に、申請者オブジェクトから取得する。

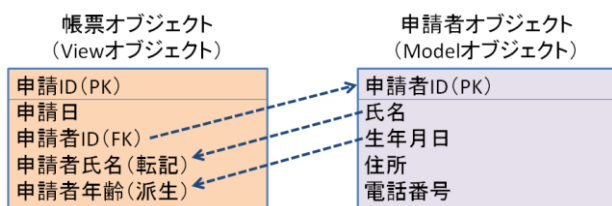


図 3 View オブジェクトの導出

## 3 評価実験

業務システムの一例である固定資産税の税額計算には毎年大きな負荷がかかる。そこで本章では負荷の思い業務システムの例として、固定資産税の税額計算をモデル化して、これにより、従来手法と提案手法での処理にかかる実行時間を比較した。処理の手順は次のとおりである。

- ・固定資産をテーブルの上から順に参照し、属性値から税額を計算する。
- ・固定資産が持つ所有者キーから当該所有者が支払うべき税額を総計する。

上記の処理を資産の数 1 万から 100 万で行い、その結果を表 1 に示す。使用した DBMS はフリーソフトの MySQL、プログラミング言語は C# である。100 万件の処理では提案手法の方が従来手法より 32 倍のパフォーマンスを確認できた。

表 1 従来手法と提案手法との比較

固定資産数(個)	従来手法(ms)	提案手法(ms)
10,000	234	20
20,000	468	32
100,000	2,356	62
200,000	4,727	156
1,000,000	24,211	764

## 4 まとめ

本稿では、DB 上のリレーションを厳密に第三正規形に準拠することによって、設計コストを下げる手法を検討した。Write 動作があまり多くない業務アプリケーションを念頭に置いている。厳密に第三正規化すると、HDD 上の DBMS では、性能が問題となる。そこで、すべてのオブジェクトをインメモリに展開した。実務を想定した性能評価では、100 万レコードで、30 倍の性能向上を確認できた。DRAM のコストが下がれば、DRAM とフリーの DBMS によって、提案手法の性能を確保できると考える。

ただし、現段階では、初歩的な性能確認にとどまっている。実際の業務システムを提案手法で実現するためには、1)複数ユーザからのアクセス処理、2)ロック処理など書き込み時の対応、3)DBMS のトランザクション管理（特に、中断処理のリカバリ等）の実現方法等、今後検討すべき課題も多い。たとえば、複数ユーザからの利用については、本手法の主旨に従えば、Model レイヤーに対して、複数のスレッドがアクセスする構成の検討が必要になると考えている。一方で、DRAM の集積度は、すでに 4Gbits/chip に達している (DIMM モジュールで 4GB)。メモリのコスト低減と、多数の設計ドキュメント管理するコストを鑑みれば、どちらを削減すべきは自明であり、提案手法の実現性について、引き続き、検討してゆきたい。

### 参考文献

- [1]真野正,「実践的データモデリング入門」,2003年3月,翔泳社
- [2]金田重郎,「三上文法に基づく ER 図関連名に対する一考察 — 関連の説明における視点移動と概念データモデリング(CDM) —」,電子情報通信学会, SIG-KBSE, 2011年3月