

訂正情報を持つ履歴データベースの提案

工藤 司[†] 片岡 信弘[‡]

あらまし 基幹系システムではオンライントランザクション処理で随時入力されたデータをデータベースに蓄積し、バッチ処理する機会が多い。この場合、統計処理等の大量のデータを対象とするバッチ検索処理は、処理中にデータが更新されるのを避けるために、オンライントランザクション処理と処理時間帯を分ける必要があり、夜間バッチ処理等の運用負荷が発生する。本論文では、訂正情報を付加することにより、オンライントランザクション処理とバッチ検索処理を並行して実施できる履歴データベース構成を提案する。提案のデータベースは、実際の基幹系システムに適用した結果、バッチ処理における運用面での改善が確認できた。

A Proposal of Temporal Database with Correcting Data

Tsukasa Kudou[†] and Nobuhiro Kataoka[‡]

Abstract Generally speaking, on mission-critical system, transaction data that is inputted by on-line is stored in database, and used by batch processing. Batch processing that retrieves mass-data, for example collecting the statistics, needs executing while on-line transaction processing doesn't update database. Accordingly, batch processing is executed as night duty. In this paper, we propose temporal database with correcting data, and try to show that batch processing can be executed during on-line transaction processing updating database by using the proposal database. Also, we describe a mission-critical system, which is made by the proposal database, and night duty is reduced in it.

1. まえがき

データベースは業務システムにおいて広く使用されており、入力されたデータはデータベースに蓄積され活用される。中でも基幹系システムでは、発生した業務データはオンライントランザクション処理のデータ入力（以下、オンライン入力と記載）により即時にデータベースに反映され、定期あるいは随時のバッチ処理により、統計などの蓄積データを活用した処理が行われるという形態が多い。例えば、業務で発生した異動データあるいは差し引きデータをオンライン入力により即時に入力し、バッチ処理により当日、月間などの異動統計、集計資料を作成するという処理形態は

在庫管理システム、小売システム、金融システムなど、基幹系システムに広く見られる処理形態となっている。

一方、業務システムの運用の面からは、クライアントサーバによるシステムの小型化、パッケージ化の進展に伴い業務システムの導入が容易になり、システム化領域が広がった反面、運用容易性の確保が重要になっている。特にユーザ部門主体で運用する場合には、入力データの一貫性維持、あるいはオンライン入力時間帯を避けて夜間に行われるバッチ処理は運用面で大きな負荷となっている。

入力データの一貫性に関しては、最新のデータだけでなく入力データそのものの履歴を蓄積し、データの経緯が管理できる時制データベースの研究が進められ、基幹系システムにも適用されている^{1),2)}。時制データベースではデータの履歴は、実世界において有効であった時間である有効時間、ある事実がデータベース内に存在していた時間であるトランザクション時間の点から管理される。時制データベースでレコードの削除

[†]三菱電機インフォメーションシステムズ株式会社
Mitsubishi Electric Information Systems Corporation

[‡]東海大学電子情報学部
School of Information Technology and Electronics,
Tokai University

を行う場合には、物理的な削除は行わず論理的な削除により該当レコードがデータベースに残される。レコードの更新についても同様の構成となっている。この構成によって時系列としてのデータの履歴管理を可能としている。さらに、トランザクション時間データベースにおいては任意のトランザクション時間のスナップショットが表現できること³⁾、このスナップショットがトランザクション時間に対して不変であることから検索処理中にデータベースが更新されても検索結果の一貫性が維持できることが示されている⁴⁾。

しかし、実際のシステム運用では様々なデータの誤入力、データの訂正が発生し、過去のトランザクション時刻を指定した検索ではデータの一貫性が維持されない場合が発生する。このため、過去のトランザクション時刻におけるデータの修正に伴いスナップショットが変化するため、データベース更新中の検索結果の一貫性が保証できなくなる。本論文は、この様な課題を解決し、データベース更新中の検索結果と、任意のトランザクション時刻での検索結果の双方の一貫性を維持するためのデータベースを提案するものである。2. で、本論文の解決しようとする時制データベースの課題を述べ、3. で、課題を解決するためのデータベースを提案する。4. で、基幹系システムである住民情報システムに提案のデータベースを適用した結果について述べ、5. で、適用結果の効果について評価する。

2. トランザクション時間での検索処理の課題

2.1. 想定する業務システム

本論文では、以下の要件を持つデータベースを使用したデータ蓄積型の基幹系システムを対象とする。

- (1) 業務データはオンライン入力され、データベースに蓄積される。業務時間帯はデータ入力頻度が高く入力停止はできない。また、オンライン入力であるため誤入力データの訂正が必要になる。
- (2) 蓄積されたデータは、バッチ処理で処理される。バッチ処理には統計処理などの検索処理を対象とする。データの検索は複数のテーブルに渡り、一定の時間を要するため、処理中にデータが更新される場合がある。

このようなシステムは、基幹系システムでは一般的に見られる形態であり、例えば、会計システムでは業務時間帯に伝票、売上の情報が店舗等でオンライン入力され、データベースに保存される。これらの入力は業務時間中停止することはできない。また、伝票などの入力に関する誤りについては、伝票の訂正が発生する。データベースに蓄積された各業務の情報は各種の事務処理で検索され活用されるが、例えば統計処理などの大量のデータを処理する場合にはバッチ処理によって

一括して処理される。バッチ処理では複数のテーブルを参照し、また長時間渡る場合があるため、処理中に検索対象データが更新されると正確な結果が保障できなくなる。バッチ処理として実行される検索処理には、例えば以下のものがある。

(1) データの検索処理

データベースに登録されたデータから定型的な各種資料の作成を行う。例えば、会計処理では締め処理として日、月、年度での各種会計データの集計処理が行なわれる。

また、経営計画にあたり非定型的な資料作成のため、必要な条件でデータを抽出し、アプリケーションソフトウェアを活用してデータを加工するエンドユーザコンピューティングも行われる。

(2) データの加工処理

蓄積されたレコードを検索、加工して関連するテーブルへの反映を行う。販売データから仕入れ、物流、資金管理データなど、検索されたデータ加工し、関係するデータベースへのレコードの挿入、更新、削除が行なわれる。

2.2. トランザクション時間を指定した検索処理

トランザクション時間データベースでは、データがデータベースに挿入された時刻を t とするとき、データのトランザクション時間は時区間 $[t, \text{now}]$ で表現される。ここで、“now” は現在時間を示す。その後、時刻 t' にデータがデータベースから削除されるとトランザクション時間は $[t, t']$ に変更され、データは物理的に削除されずデータベース内に残される。従って、トランザクション時刻 T を含む時区間を持つレコードを検索すれば、トランザクション時間 T のスナップショットとしてデータが表現される。図1にトランザクション時間データベースの例を示す。これは、入金データのレコードのトランザクション時間と入金日、入

(1) 入金データベースの例

ID	挿入時刻	削除時刻	入金日	金額
001	7/ 1	now	7/ 1	1,000
002	7/ 7	8/ 5	7/ 5	2,000
002	8/ 5	now	7/ 5	200

(2) トランザクション時間軸上の表現

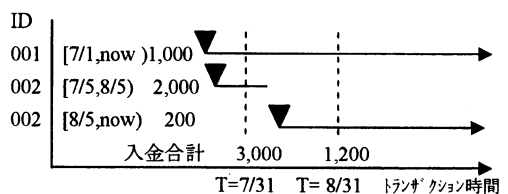


図1 入金データベースの例

金額を示している。例えば、IDが002の入金は7月7日に2,000円で挿入され、8月5日に200円に訂正されている。従って、トランザクション時刻に関して、7月31日現在の入金合計は3,000円であり、8月31日現在は1,200円となる。

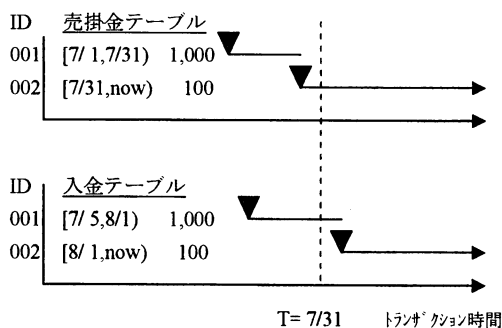
ここで、基幹系システムでは入金の合計は大量のデータ検索となるため、バッチ処理で実行される。従来の最新の状態のみを保存するスナップショットデータベースを使用した基幹系システムのバッチ処理では、長時間の検索処理を行うと検索処理中にデータが更新されるため検索データの一貫性が保障できない。このため、一般的に長時間のバッチ処理はデータベースが更新される業務時間帯を避けて、いわゆる夜間バッチとして処理される。

一方、時制データベースでは、過去のトランザクション時刻を指定したスナップショットは変化しないため、データ更新中にも長時間の検索処理が可能となる。また、過去の任意の時刻における検索処理の結果を得ることも可能となる。図1において8月5日のデータベース更新中に7月31日のトランザクション時刻を指定して検索を行った場合を考える。ID002のデータは更新前後であるにもかかわらず7月31日のスナップショットは変化しないため、更新処理中も検索結果の一貫性が維持される。すなわち、データベース更新中にバッチ処理を実施することが可能になる。

2.3. トランザクション時間での検索処理の課題

しかし、実際に基幹系システムで時制データベースのスナップショット機能を使用したバッチ処理を実現する場合には、過去のトランザクション時間のレコードを物理的に修正できないことが課題となる。なぜならば、データ更新中の検索処理においては、一旦、挿入されたレコードは削除時間の更新を除き変更しない点を利用してデータ検索の一貫性を確保しているが、このことは、トランザクション時間軸上でのデータの整合を常に保つ必要があることを意味している。ところが、実際の業務システムでは様々なパターンのデータの誤入力が発生し、それに伴うデータの訂正においては、レコードの削除、追加、キー情報の変更などが発生する。また、複数のテーブルが関連する場合には、訂正のタイミングがテーブル間で必ずしも一致しない場合がある。この結果、トランザクション時間を指定したスナップショットではデータの一貫性が維持されていない時間帯が発生する。この点を、トランザクション時間スナップショットのデータ一貫性と、トランザクション時間軸に沿ったデータの一貫性の点から示す。

(a) スナップショットのデータ一貫性に関する課題
データベースは複数のテーブルを含み、テーブルは



7/31 時点では、ID002の入金データは該当売掛なし。

図2 スナップショットにおけるデータ一貫性の課題

相互に関連するため、データ一貫性のための制約が存在する。しかし、データの修正にあたっては、一時的にこの制約が崩れる場合がある。例えば、基幹系システムは一般に複数の業務機能から構成され、データベースの各テーブルは該当業務の主管部門が管理する。従って、データの訂正のタイミングは必ずしも一致しておらず、矛盾が発生する時区間が存在することがある。例えば、図2に示す様に、売掛金のIDを訂正する場合を考える。売掛金の訂正に対応して、入金しているデータについてもIDの訂正が行なわれるが、両者の修正のタイミングが異なる場合には、図に示す様に7月31日時点では入金テーブルのID002に対応する売掛金テーブルのレコードがないという状況になる。

最新データを管理するスナップショットデータベースの場合には、バッチ処理の手順としてデータの整合チェックを実施し、エラーがあればデータを修正後、集計処理を行う。しかし、トランザクション時間データベースにおいて過去のトランザクション時間軸上のデータを修正した場合、データ更新中の検索処理におけるデータ一貫性が維持できなくなる。また、時制データベースの特性であるデータの経緯の履歴管理ができなくなる。

(b) トランザクション時間軸上の一貫性の課題

業務システムにおいては刻々と変化するステイツ型の情報が管理され、これらはイベント型の情報によって変化する⁵⁾。従って、トランザクション時間軸上でも両者の一貫性が保たれる必要がある。例えば、会計システムにおいて、ステイツ型の情報である売上高は刻々と変化し、その変化は入金というイベント型の情報と整合している。すなわち、例えば、7月末現在の売上は、6月末現在の売上に7月の入金を加算した金額となる。しかし、図3に示す様に、時制データベースにおいて7月に6月の入金データID002の金額を訂正した場合には、この一貫性が保障できなくなる。また、IDが003のデータについては9月3日に別のID

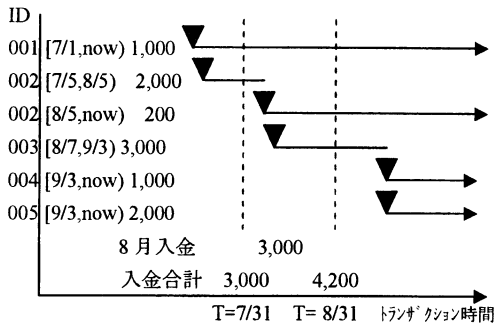


図3 トランザクション時間軸上の一貫性の課題

のデータ 004, 005 に変更されたのか、削除され新たな売上データが発生したのか、データベースの情報から判断がつかず、データの経緯の履歴管理が困難となる場合がある。

3. 訂正情報をもつ履歴データベースの提案

提案のデータベースは、2.3に記載したトランザクション時間を指定したバッチ処理での問題点を解決し、オンライン入力によるデータ更新中にもバッチ処理において一貫性のあるデータベース検索が可能なデータベースを提供することを目的としている。

3.1. 提案するデータベースの構成

提案のデータベースは、トランザクション時間データベース、あるいはトランザクション時間と有効時間を持つバイテンポラルデータベースを以下の通り拡張した構成とする。概要を図4に示す。

(1) 訂正情報の追加

データ訂正の結果、レコードの主キー情報が変更される可能性のあるテーブルは、トランザクション時間に沿って訂正前後のレコードを追跡するための訂正情報を追加する。訂正情報は、テーブルの属性として追加しても、また別のテーブルのレコードとして管理してもよい。テーブルの属性として追加する場合には、データの訂正において訂正後のレコードに引き継がれる。

(2) ユーザ更新処理の制約

過去のトランザクション時間のデータが変更されるのを防ぐため、ユーザの更新処理に対し以下の制約を設け、一旦挿入されたレコードは論理的な訂正、削除を除きユーザから操作不可である。

① データの挿入

トランザクション時間はシステムが設定する。

② データの訂正

更新前レコードにはシステムが削除時刻を設定し、物理的には削除しない。更新後のレコ

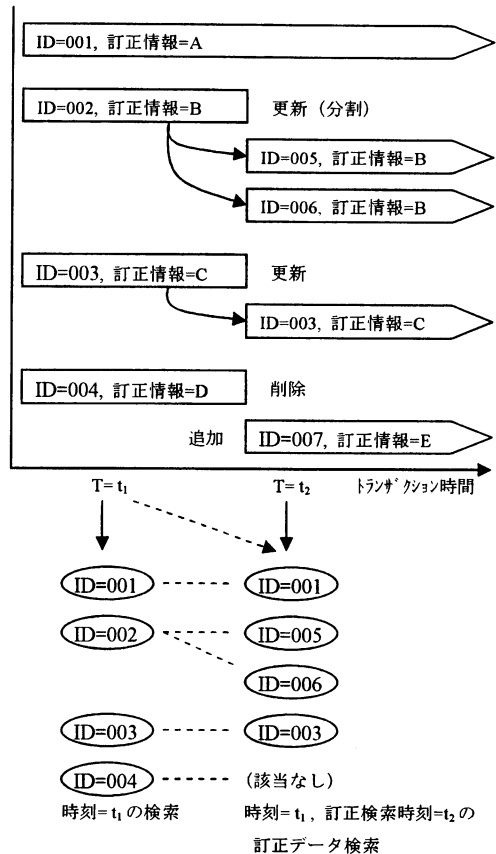


図4 訂正情報を持つ履歴データベースの概要

ードは新規レコードの挿入と同様に処理する。

③ データの削除

削除時刻をシステムが設定し、物理的には削除しない。

(3) トランザクション時刻を指定した訂正検索処理

通常の実験処理の他に、訂正検索処理の機能を付加する。訂正検索処理では検索用のトランザクション時刻として、検索用時刻と、訂正データ検索用時刻2つを指定し、検索用時刻のスナップショットのレコードに削除時刻の設定されているものは、以下に示す訂正データ検索用時刻のスナップショットのレコードに置き換えたレコードを返す。

(ア) 削除時刻が訂正データ検索時刻以降
検索用時刻のレコードを返す。

(イ) 削除時刻が訂正データ検索時刻以前

以下の訂正検索時刻のレコードを返す。この時、訂正情報が設定されているテーブルについては、主キーではなく訂正情報で検索を行う。また、該当レコードがない場合には、レ

コードを返さない。

このデータベースのテーブルのリレーションスキーマ R は、例えば $R(K_1, \dots, K_m, T_s, T_e, C, A_1, \dots, A_n)$ で表現される。ここで、 K_1, \dots, K_m は、トランザクション時刻を指定したスナップショットで生成されるテーブルの主キー属性、 A_1, \dots, A_n が同じく主キー以外の属性、 $[T_s, T_e)$ がトランザクション時間属性、 C が訂正情報の属性となる。トランザクション時刻データベースにおいては、 K_1, \dots, K_m, T_s が主キー属性となる。また、訂正情報属性 C は訂正前後のレコードの対応付けが可能な属性を設定する。例えば、図1の入金テーブルに訂正情報を付加したリレーションスキーマは、入金 (ID, 挿入時刻, 削除時刻, 訂正情報, 入金日, 金額) となる。訂正情報として例えば、挿入時のIDを訂正情報として使用できる。これは、IDがスナップショットデータベースに射影されたテーブルの主キーであるため、一意であることによる。

3.2. 提案するデータベースの効果

3.2.1. データ更新中の検索処理の一貫性

このデータベースは、挿入時刻, 削除時刻に関してはトランザクション時間データベースの構成であり、かつ、一旦挿入されたレコードは削除時刻の設定以外には変更しない。このため、トランザクション時刻 $t < \text{now}$ を指定した検索処理ではデータ更新中であってもデータの一貫性が保たれる。すなわち、トランザクション時刻 t のデータを検索した場合、 $T_s \leq t < T_e$ を満たすレコードが抽出される。従って、以下の検索中に挿入, 削除されたレコードは検索対象から除外される。

(1) 検索処理中に新たに追加されたデータ

$T_s = \text{now} > t$ となるため該当しない。

(2) 削除されたデータ

$T_e = \text{now} > t$ となるため削除後も対象となる。

従って、オンライン入力によるデータ更新中であっても、長時間のバッチ処理におけるデータベース検索において、データの一貫性を保つことができる。

3.2.2. トランザクション時間指定の検索実現

訂正検索を利用することによってトランザクション時間を指定した検索処理の問題点を回避することができることを示す。

(a) スナップショットの一貫性

図2において、検索用時刻を7月31日、訂正データ検索用時刻を8月2日とした訂正検索の例を図5に示す。時刻7月31日で検索された入金テーブルのIDが001のレコードは、削除時刻が8月2日前的ため、IDが002のレコードに変更される。この様に過去のトランザクション時間においてエラー・データが検出された場合においても、訂正データ検索用時刻に訂正

(1) 売掛金テーブル

ID	挿入時刻	削除時刻	訂正情報	契約日	金額
001	7/ 1	7/31	001	7/ 1	1.000
002	7/31	now	001	7/ 1	100

(2) 入金テーブル

ID	挿入時刻	削除時刻	訂正情報	入金日	金額
001	7/ 5	8/ 1	001	7/ 5	1.000
002	8/ 1	now	001	7/ 5	100

(3) 売掛金訂正検索結果

ID	契約日	金額
001	-	-
002	7/1	100

(4) 入金訂正検索結果

ID	入金日	金額
001	-	-
002/	7/5	100

検索時刻 : 7/31 訂正検索時刻 : 8/2

図5 スナップショットの訂正検索の例

後のトランザクション時間を指定して訂正検索を行ない、過去のトランザクション時刻のレコードを物理的に修正することなく一貫性のある結果を得ることができる。

(b) トランザクション時間軸上の一貫性

トランザクション時間軸上の一貫性の課題についても、同様に最新の訂正後データに置き換えることにより一貫性を保つことができる。図6に、図3において訂正データ検索用時刻を9月4日として訂正検索で検索した結果を示す。ここで、IDが003のレコードはIDが004と005のレコードに分割されたとする。検索

用時刻が7月31日のIDが002のレコードは挿入時刻が8月5日のレコードに、IDが003のレコードは挿入時刻が9月3日のIDが004と005のレコードに置き換えられる。この結果、7月31日と8月31日の訂正検

(1) 入金テーブル

ID	挿入時刻	削除時刻	訂正情報	入金日	金額
001	7/ 1	now	001	7/ 1	1.000
002	7/ 7	8/5	002	7/ 5	2.000
002	8/ 5	now	002	7/ 5	200
003	8/ 7	9/3	003	8/ 7	3.000
004	9/ 3	now	003	8/ 7	1.000
005	9/ 3	now	003	8/ 7	2.000

(2) 7/31の訂正検索結果

ID	入金日	金額
001	7/ 1	1.000
002	7/ 5	200
	合計	1.200

(3) 8/31の訂正検索結果

ID	入金日	金額
001	7/ 1	1.000
002	7/ 5	200
004	8/ 7	1.000
005	8/ 7	2.000
	合計	4.200

図6 トランザクション時間軸での訂正検索例
 索用結果ではデータの一貫性が維持され、入金と入金
 合計の矛盾をなくすることができる。

4. 基幹系システムへの適用

提案のデータベースを基幹系システムである住民
 情報システムに適用した。

4.1. 住民情報システムの構成

住民情報システムは市町村における行政事務処理の
 ためのシステムであり、市町村管掌の行政事務を支援
 することを目的としている。住民情報システムは業務
 的に以下に区分される。

(1) 住民情報系システム

住民に関する住民基本台帳、印鑑登録証明、国民
 年金、国民健康保険などの資格管理、台帳管理、
 証明書発行などの業務を行う。

(2) 税関連系システム

市町村管掌の地方税の賦課、徴収事務および税証
 明の発行を行う。

(3) 福祉系システム

保育所、児童手当、老人医療などの資格管理、徴
 収、支給事務、健康診断の受診、データ管理など
 福祉行政に関連する事務を行う。

(4) 内部情報系システム

市町村の人事管理、給与支給、財務会計事務など
 の、行政機関内部の事務を行う。

業務データベースのデータは時系列に長期間に渡
 り蓄積、保存され活用されるものが多い。例えば、住
 民基本台帳のデータは住民に関しては出生、転入等
 で該当自治体の住民となった以降、転居、あるいは婚姻
 など、住民基本台帳に関連する全ての記録が保存され、
 死亡、転出等で除票となった後も5年間は保存される。
 データの入力はオンライン入力での即時入力を基本と
 する。各種の届出が窓口で受けられ、システムに入
 力されてデータベースに蓄積される。届出のうち住民
 票、あるいは印鑑登録証明書などは、請求に応じて届
 出の内容を反映した証明書を直ちに発行する必要がある
 が、業務時間中はオンライン入力を停止することはで
 きない。データベースに蓄積された各業務のデータは、
 各種の事務処理で検索され活用されるが、統計処理な
 ど大量のデータを処理する場合には、バッチ処理によ
 って処理される。

4.2. データベースの構成

データベースは、商用のリレーショナルデータベー
 ス上に3.1に記載した機能を付加して構築した。本シ
 ステムでの実装に関する事項を以下に記載する。

(1) データベース属性の適用範囲

提案データベースの以下の属性は、各テーブル毎

に業務仕様の点から以下の方針で必要な属性を選
 択して適用した。

① 挿入時刻

トランザクション時刻による履歴管理の必
 要なテーブルに適用した。例えば、年度毎に
 設定される税率、会計科目などは、業務上ト
 ランザクション時刻による履歴管理が不要
 のため適用を除外した。

② 削除時刻

挿入時刻を持つテーブルのうち、トランザク
 ション時刻によるデータの訂正が発生する
 ものに適用した。例えば、住民の情報に関す
 る入力誤りは、システムのデータを実世界に
 合わせて訂正する必要があるため、削除時刻
 による訂正が必要になる。逆に庁内の会計業
 務ではシステムの伝票データにより会計処
 理がなされ、実世界と一致するため削除時刻
 の使用を除外した。

③ 訂正情報

削除時刻を持つテーブルのうち、キーが変更
 となるテーブルに適用した。例えば、住民情
 報システムでは該当市町村の住民以外にも
 市町村内に資産を持つ者を住登外者として
 管理しているが、住登外者が転入した場合
 には同一人物として管理するため同一のキー
 を適用する必要がある。しかし、窓口の転入
 処理においては、同一人物の確認が困難な場
 合があり、キーの訂正が発生する場合がある。

(2) 挿入時間と削除時間の実装

訂正検索の際には、トランザクション時刻に沿
 って該当レコードの履歴を追跡する必要がある。こ
 のため、訂正検索主体のテーブルは、訂正後のレ
 コードにも挿入時刻として最初の挿入時刻を設定
 した。例として、図7に7月5日に挿入したレ
 コードを8月5日に訂正する場合を示す。図7の①
 は従来の方式であり、トランザクション時間を指
 定した検索が効率的に実装できるが、訂正検索で
 は同一IDのレコードを追跡する必要がある。一
 方、②は訂正検索でのレコードの追跡は不要であ

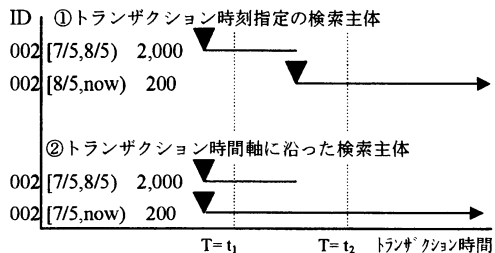


図7 挿入時刻・削除時刻の実装

るが、トランザクション時間を指定した検索はやや複雑になる。

(3) トランザクション時間の単位

トランザクション時間はデータベース上のキー属性となるため、データ更新の頻度から設定する必要がある。本システムでは、画面からデータ入力を行うため、1秒単位とした。

4.3. バッチ処理の運用

検索のみのバッチ処理は、現在時刻以前のトランザクション時刻を指定した検索を行うことにより、オンライン処理時間内に実行する運用を基本にした。このため、バッチ処理の実行にあたっては前回使用したトランザクション時間を自動的に管理する機能を付加し、バッチ処理で検索されたデータのトランザクション時刻を管理できる構成とした。

(1) 期日指定の検索処理

市町村事務においては、人口統計あるいは入金などの統計は毎日の業務終了時点のデータを使用する。このような統計処理は、翌日の業務時間内に実行する運用とした。

(2) 期日の指定されていない検索処理

現在時刻以前の適当なトランザクション時間を指定して、オンライン入力と並行して実行する運用とした。

5. 評価と考察

提案したデータベースを基幹系システムに適用した結果について、システムの運用面とシステムの構築面から評価する。

5.1. システム運用面の評価

(a) バッチ処理の運用

オンライン入力中に任意のトランザクション時間を指定して一貫性のあるデータ検索処理が可能になった。これにより、従来のシステム運用では窓口業務の時間帯終了後にシステム運用を延長して実施していたバッチ処理を、翌日以降の業務時間帯に実行する運用にできた。また、データベースの更新処理では、時制データベースの採用により入金データ反映などのデータ挿入のバッチ処理は、オンライン入力中に実行する運用にできた。従来のシステム運用においてはオンライン入力後、バッチ処理開始前に、窓口部門へオンライン入力完了確認、あるいは残業時間におよぶオンライン入力の完了待ちという作業が発生していた。これらの作業も含め、人口4万人程度の自治体の事例では従来システムにおいて1日平均1.5時間程度あった残業時間帯のバッチ処理作業を削減できた。さらに、夜間の運用がなくなったことからバッチ検索処理をシ

ステム部門から業務主管部門に移管でき、バッチ処理の依頼、あるいは帳票の受取りなどの連絡・調整に関する作業を効率化できた。

ただし、オンライン入力と同一のレコードを更新する一部のバッチ処理は、オンライン入力とバッチ処理の時間帯を分ける必要があった。

(b) トランザクション時刻検索でのデータ一貫性

訂正検索によって、過去のトランザクション時刻のデータを修正することなく、過去のトランザクション時刻を指定した検索データの一貫性が維持でき、この結果、(a)の運用が実現できた。

また、住民情報システムでは、定期的な集計と集計期間の異動集計が多い。従来のスナップショットデータベースでは訂正データがシステムで管理できないため、齟齬が発生した場合の調査に時間を要していた。現在時点における前回・今回集計およびその間の異動を正確に把握できる訂正検索結果と、前回処理期日時点における前回集計の双方を管理できる機能が提供された結果、前回と今回の齟齬の原因追跡が容易になった。この効果は、集計を月平均20表、齟齬の発生率を10%、調査時間を0.5人日とすると、1人日/月(20×10%×0.5人日)の効果となる。

5.2. システム構築面の評価

提案したデータベース属性の適用割合を表1に示す。なお、トランザクション時間を指定してバッチ処理を実施し、結果の再現性をもたせるためには、本来、時制データベースとしての管理不要なテーブルについてもトランザクション時間データベースとして管理を行う必要があり、挿入時間、削除時間の適用範囲は広くなる。例えば、郵送物を送付させるための宛名は、従来のシステムでは履歴の管理は不要であった。しかし、トランザクション時間を指定してバッチ処理を行う場合には、例えば地区毎に分離して集計する場合など、処理の途中で異動が発生すると一貫性が保てなくなるため、バッチ処理において参照されるデータについてはトランザクション時間データベースの構成が必要となる。これに伴うデータ容量は、該当自治体の住民異動の状況により異なるが、年に20%の住民が異動する自治体であれば、5年間で必要なデータの容量は2倍程度となる。

表1 提案したデータベース属性の適用割合

業務区分	全体	履歴なし	挿入時刻	削除時刻	訂正情報
住民情報系	36	4	32	18	1
税関連携	70	17	53	31	0
福祉系	38	3	35	24	0
内部情報系	61	16	45	8	0
合計	205	40	165	81	1

5.3. 考察

(a) バッチ処理の運用に関する考察

訂正情報と訂正検索の導入により過去のトランザクション時刻のレコードを修正することなく、検索データの一貫性を維持できる様にした結果、基幹システムにおいても、オンライン入力中のバッチ検索処理が可能であることが分かった。これは住民情報システムにおいては有効であった。なお、業務システムの運用は業務分野毎に異なるため、有効性に関しては個別に検討する必要がある。例えば、当日のバッチ処理の結果を翌朝直ちに使用するシステムの場合には、夜間バッチとしての処理も必要と考えられる。反面、24時間運転の場合には、定期的に時間を区切って入力処理と並行したバッチ処理が可能となり、本方式は有効と考えられる。

(b) 適用範囲に関する考察

実際の業務システムでは、トランザクション時刻、訂正情報の管理の要否は、業務のデータモデルによって決まり、適用範囲としては図8に示す包含関係にあると考えられる。例えば、表1に示す様に、トランザクション時刻の管理のないテーブルは年度単位の情報を管理する税業務(24%)、内部情報系(26%)では随時入力する住民情報系(11%)、福祉系(8%)の倍以上となっている。反面、内部情報系はシステムの伝票を正として運用が多いため削除時刻は他に比較して少なくなっている。なお、訂正情報は住民情報系の1件であるが、これは以下の理由による。

- (1) 住民情報システムでは住民の情報が中心であるため、テーブル間の一貫性は住民の情報に関するテーブルの訂正検索で対応できること。
- (2) 実世界と関連する属性については、訂正検索が必ずしも必要とならないこと。

(2)については、例えば、住民を識別するIDは実世界にはないため、システム内で変更あるいは訂正検索による訂正前後を統合した処理が有効になる。一方、軽自動車税においては車両番号を識別IDとしているため、訂正には業務的な手続きが必要になる。この様に、適用の範囲、方法は業務のデータモデルの検討が必要となる。

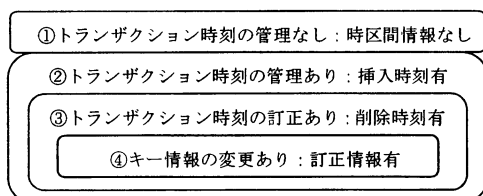


図8 提案したデータベースの属性の適用範囲

(c) システム構築に関する考察

5.2.でも述べた通り、トランザクション時刻を指定してバッチ処理を行う場合には、関連するテーブルについてもトランザクション時間の管理を行う必要がある。この結果、データベースの容量はデータの追加・更新頻度に比例して増加する。ただし、近年、記憶媒体の単位価格が下がっているため、データベース容量自体が増加することは構築コストの面でそれ程の問題にはならなかった。むしろ、トランザクション時間を管理することによるデータの訂正頻度は、業務内容、実現するシステム機能、運用などによって大幅に異なるため、開発段階で容量の見積りに注意する必要があると考えられる。

むすび

トランザクション時間を活用することにより、データ更新中にも一貫性のあるデータ検索が可能となる。反面、一旦、挿入したレコードを変更できないという制約があり、トランザクション時間によってはデータの一貫性が保障されない場合がある。

本論文では訂正検索によりトランザクション時間を指定した検索データの一貫性を維持しながら、データ更新中に一貫性のあるデータ検索を可能とする方法について提案した。また、実際にトランザクション時間データベースを住民情報システムに適用した結果、オンライン入力中のバッチ検索処理実行という運用面での効果と、訂正検索による検索データ一貫性維持の効果を確認した。また、適用の要否はシステム化対象業務の内容によって異なるため、業務のデータモデルを十分に検討する必要があることが分かった。

今後の課題としては、オンライン入力中にバッチ処理でデータベースを更新する場合の運用面での改善方法の実現がある。

参考文献

- 1) 西川洋一：最新データベース技術、秀和システム(2002)。
- 2) 川上潤司：不正を見抜く情報システム、日経情報ストラテジー、1996.9, pp.116-125(1996)。
- 3) 天笠俊之、有次正義、金森吉成：時間的に変化するデータに対する索引技術、情報処理、Vol42, pp.972-979(2001)。
- 4) 大山敬三：情報検索システムのオンライン更新における検索集合の一貫性について、学術情報センター紀要、Vol10, pp.29-35(1998)。
- 5) 田中聡：業績責任会計における時制マスタ管理システムの重要性、情報処理学会データベースシステム研究会研究報告、Vol119, No56, pp.333-338(1999)。
- 6) 増永良文、リレーショナルデータベースの基礎—データモデル編—、オーム社(1990)。