

データ更新中の検索結果一貫性維持実現事例

工藤 司[†] 片岡 信弘^{††} 水野 忠則^{†††}

トランザクションによりデータベースを更新し、バッチ処理でデータベースを検索して統計資料などを作成するという運用は基幹系システムで広く行われており、効率的な運用のためには両者の処理を同時に実行できることが必要である。バイテンポラルデータベースを使用することにより、データ更新中にも検索結果の一貫性を維持することが可能である。しかし、基幹系システムに適用するためにはデータの入力誤り、あるいは入力遅れなど、実際の運用に際して発生する様々な課題にも対応できることが必要である。本論文では、データ更新中の検索結果一貫性維持方式の事例として、バイテンポラルデータベースを基幹系システムに適用した事例を示す。本事例により、データ更新中に、実際の運用で発生する様々な訂正を反映した、一貫性のある検索結果が得られることを確認した。さらに、短期間のデータ訂正だけでなく長期にわたるデータ訂正をとまなう業務にも効果があること、データの訂正を、内部処理での訂正と業務処理での訂正の両面から管理できることを確認した。

Implementation of Integrity Maintenance System of Database Query under Updating

TSUKASA KUDOU,[†] NOBUHIRO KATAOKA^{††} and TADANORI MIZUNO^{†††}

In many mission-critical systems, databases are updated by transaction processing, and are queried by batch processing to make statistics and so on. It is necessary that the processing of both can be executed at the same time for the efficient operation. The integrity of the querying result can be maintained by using a bitemporal database even while data are being updated. However, in actual operations of mission-critical systems, various kinds of problems occur, such as the incorrect input of the data or being behind in the data entry. Then, it is necessary that the bitemporal database can support those problems. In this paper, we show the case to have applied the bitemporal database to a mission-critical system, as an implementation of the method to maintain the integrity of the querying result under data being updated. By this case, we confirmed that the integrity of the querying result was maintained under data being updated, and that various corrections done by the actual operations of the system were reflected to the result. Furthermore, we confirmed that it is effective for the data corrections not only of a short term but also for a long term, and that the correction of the data could be managed from the both sides of the correction by the internal process and the correction of the business.

1. ま え が き

業務データをオンライン端末から入力（以下、オンライン入力と記載）してデータベースに蓄積し、定期あるいは随時のバッチ処理¹⁶⁾により統計資料や分析資料を作成するという運用は、小売、金融、製造などの基幹系システムで広く行われている。たとえば、小売システムでは業務時間帯に売上情報が店舗などでオ

ンライン入力され、データベースに蓄積される。そして、当日、月末などの業務終了時点で決算処理が行われるが、これらは大量のデータを対象とするためバッチ処理で処理される。したがって、システムの効率的な運用には、オンライン入力とバッチ検索処理を並行して実行できることが必要になる。

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²⁾では、大量のデータ検索をオンライン入力と並行して実施できないという課題がある。データベースシステムには複数ユーザが更新、参照する際にもデータのー貫性を維持するためのトランザクション処理¹⁶⁾が備えられ、大量データの処理でもミニバッチ¹⁶⁾により少量のトランザクションに分割する方式が示されている。しかし、大量データを長時間にわた

[†] 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

^{††} 東海大学情報理工学部
School of Information Science and Technology, Tokai University

^{†††} 静岡大学情報学部
Faculty of Information, Shizuoka University

り検索する場合には、個々のレコードの検索で更新処理との競合を避けたととしても、検索結果の一部は更新前、一部は更新後となってしまふ。

一方、時系列に変化するデータを扱うデータベースとして時制データベースがある^{3)-5),8)-12)}。時制データベースは、ある事実がデータベース内に存在していたトランザクション時間をサポートするトランザクション時間データベース、ある事実が実世界において有効だった有効時間をサポートする有効時間データベース、および有効時間とトランザクション時間の双方をサポートするバイテンポラルデータベースに区分される¹¹⁾。

トランザクション時間データベースでは、過去のトランザクション時間の状態をスナップショットとして表現できること¹¹⁾、検索処理中にデータベースが更新されてもこの表現を利用して検索結果の一貫性が維持できることが示されている¹²⁾。しかし、実際のシステム運用では、バッチ処理でデータの入力誤りが検出された場合は、データを訂正して再処理を行う必要がある。このとき、訂正データのトランザクション時間は指定時刻以降になるため、検索結果に反映されないという課題がある。

また、版管理データモデルは時間の経過とともに作成されるデータについて、作成や削除の時刻、版の導出関係などを管理するものであり^{6),7),13),14)}、時系列のデータ変化だけでなく枝分かれのある版の導出関係が管理できる¹³⁾。したがって、ある時刻のデータ集合を1つの版とすると、この版に対するオンライン入力によって時系列に作成される版の集合と、この版に対するデータ訂正を行った版の双方が構成できる。このモデルはソフトウェア開発やCADの版管理では不可欠なモデルになっている¹³⁾が、基幹系システムのようにデータ入力頻度の高いシステムに適用する場合、頻繁に訂正データを検出し新たな版を構成しなければならないという課題がある。

バイテンポラルデータベースでは、データ更新中にも訂正結果を反映した一貫性のある検索が可能である。ここで、これを基幹系システムに適用するためには、実際のシステム運用で発生する様々な訂正にも対応できることが必要である。しかし、実際に基幹系システムに適用し、その運用を評価した事例は見当たらない。我々は、バイテンポラルデータベースによりデータ更新中であっても、実際のシステム運用において発生する訂正が反映された、一貫性のある検索が可能であることを示す。さらに、これを基幹系システムに適用し、実際のシステム運用における効果を評価した。その結

果、データ更新中に一貫性のある検索が可能であるだけでなく、長期間にわたるデータ訂正を行う業務でもデータの履歴管理が容易になり、また、内部処理での訂正と業務処理での訂正を分けて管理できるという効果があることを確認した。

本論文では、2章で解決しようとするデータ更新中の検索処理の課題を述べ、3章でバイテンポラルデータベースにおいて課題が解決できることを示す。4章で基幹系システムに対するバイテンポラルデータベースの実装事例と、そのシステム運用について述べ、5章で実装および運用の結果について評価、考察する。

2. トランザクション時間データベースの課題

2.1 対象とする検索処理

本論文では、以下の検索処理を対象とする。

- (i) 更新処理との並行実行 高い頻度で更新処理を行うトランザクションと並行して検索が行われる。
- (ii) 大量データ一括処理 大量のデータを一括で検索するため、処理に時間を要する。
- (iii) 訂正後の再検索 検索データに誤りを検出した場合には、変更、削除、追加の訂正を行い、訂正後のデータを再検索する。

このような検索処理は、小売システムの決算処理など基幹系システムでは一般的に行われる形態であり、大量のデータ検索はバッチ処理で行われる。データベースシステムではトランザクションに一貫性制約が課され、複数部門での並行作業でも作業単位の協調実行でデータの一貫性を維持する方式が提案されている¹⁷⁾。これらにより一定の一貫性が確保されるが、大量のデータ検索をとまなう整合性の確認、たとえば複数のテーブルの突き合わせ、あるいは現金、棚卸し結果と突き合わせるための集計はバッチ処理によって行う必要がある。したがって、バッチ処理では誤ったデータを訂正したうえで再処理できることが必要である。

2.2 データ更新中の検索方式

トランザクション時間データベースにおいて、あるデータがデータベース内で有効だった時区間を $[t_a, t_d)$ で示す。ここで、 t_a は該当データがデータベースに追加された追加時刻、 t_d はデータが論理的に削除された削除時刻を示す。データが削除されていない場合には $t_d = \text{now}$ で表現され、“now” は検索時点の現在時刻を示す^{1),8)}。データの変更を行う場合には、変更前のデータに削除時刻を設定して論理的に削除し、変更後のデータを追加する。このように、一度追加されたデータは物理的に削除されずに残されることから、

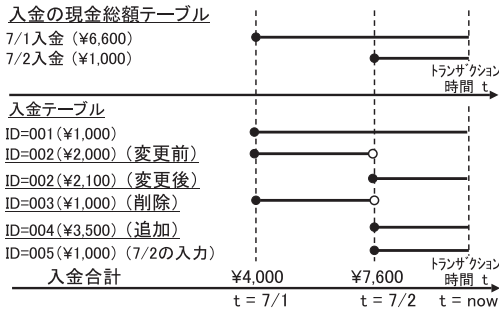


図 1 トランザクション時間データベースのスナップショット
Fig. 1 Snapshot of transaction time database.

トランザクション時間 t を指定して $t_a \leq t < t_d$ なるデータを検索すれば時刻 t 現在のスナップショットが表現され、データ更新中であってもデータの一意性が保たれる。

スナップショットによる検索を、小売システムに適用した例を図 1 に示す。図で、7 月 1 日の入金テーブルを入金の現金総額テーブルと突き合わせた結果、 $ID = 002$, $ID = 003$, $ID = 004$ で各々、入力誤り、重複入力、入力漏れが検出され、7 月 2 日に変更、削除、追加による訂正が行われた。また、7 月 2 日に新たに $ID = 005$ の入金データが追加されている。7 月 1 日のスナップショットでは、これらの 7 月 2 日の更新は反映されない。この特性により、オンライン入力によるデータ更新中であっても、長時間のデータベース検索において検索結果の一貫性を保つことができる。

2.3 データ訂正に関する課題

実際の業務においては 7 月 1 日の正しい集計結果、すなわち現金総額テーブルと整合した検索結果が得られることが必要になる。しかし、図 1 の例では、7 月 1 日のスナップショットでは訂正前のデータが検索され、7 月 2 日では 7 月 2 日の入金分である $ID = 005$ のデータが検索対象になるため、7 月 1 日の訂正された状態は検索できないという課題がある。

なお、以下の例も含めてトランザクション時間は日単位としている。実装においてはデータ更新頻度などのシステム要件に応じて時間の単位が決定される。

3. バイテンポラルデータベースでの検索

バイテンポラルデータベースを使用することにより、2.3 節の課題が解決できることを示す。

3.1 バイテンポラルデータベースの構成

バイテンポラルデータベースのテーブルのリレーション¹⁸⁾ R は

$$R(K, T, V, A) \tag{1}$$

で表現される。各々の属性を以下に示す。

- $K = \{K_1, \dots, K_m\}$
トランザクション時間、有効時間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集合を示す。
- $T = \{T_a, T_d\}$
トランザクション時間の時区間属性であり、システムにより自動生成され、ユーザには公開されない。ここで、 T_a は該当データがデータベースに追加された追加時刻、 T_d はデータが論理的に削除された削除時刻を示す。データが削除されていない場合には T_d の属性値は “now” で表現される。データの変更を行う場合には、変更前のデータに削除時刻を設定して論理的に削除し、変更後のデータを追加する。
- $V = \{V_a, V_d\}$
有効時間の時区間属性であり、 V_a は該当データが実世界に存在した開始時刻、 V_d は終了時刻を示し、検索時点でデータが有効な場合にはトランザクション時間と同様に “now” で表現される。一度追加されたデータはデータベース内に残されることから、トランザクション時間による検索の場合と同様に、指定した有効時間現在のスナップショットが表現できる。
- $A = \{A_1, \dots, A_n\}$
その他の属性集合を示す。

3.2 データ更新中の検索方式

バイテンポラルデータベースは、トランザクション時間を管理している。したがって、2.2 節で示したスナップショットの検索により、トランザクション時間データベースと同様にデータ更新中でも一貫した検索結果を得ることができる。さらに、有効時間を管理しているため、実世界での状態を、有効時間を指定して検索することができる。

R のトランザクション時間 t_1 、有効時間 t_2 におけるスナップショットは、トランザクション時間 t_1 を含む時区間 $T = \{T_a, T_d\}$ と有効時間 t_2 を含む時区間 $V = \{V_a, V_d\}$ を持つデータ、すなわち以下の $R_1(t_1, t_2)$ で表現される。

$$R_1(t_1, t_2) = \{r | r \in R \wedge r[T_a] \leq t_1 \wedge t_1 < r[T_d] \wedge r[V_a] \leq t_2 \wedge t_2 < r[V_d]\} \tag{2}$$

ここで、 $r[T_a]$, $r[T_d]$, $r[V_a]$, $r[V_d]$ は R に含まれるタプル r の T_a , T_d , V_a , V_d の属性値を示す¹⁸⁾。したがって、たとえば、検索結果に誤りが検出され、ある時刻 t にデータを訂正した場合には、 $t_1 > t$ なるトランザクション時間 t_1 を指定して検索を行うことにより、訂正後のデータの検索が可能になる。

データベースの状態 (7/2現在)

現金総額テーブル

入金日	Ta	Td	Va	Vd	金額
7/1	7/1	now	7/1	now	6,600
7/2	7/2	now	7/2	now	1,000

入金テーブル

ID	Ta	Td	Va	Vd	金額	(1)	(2)	(3)	区分
001	7/1	now	7/1	now	1,000	●	●		
002	7/1	7/2	7/1	now	2,000	●			変更前
002	7/2	now	7/1	now	2,100		●		変更後
003	7/1	7/2	7/1	now	1,000	●			削除
004	7/2	now	7/1	now	3,500		●		追加
005	7/2	now	7/2	now	1,000			●	7/2の入力

入金テーブルのスナップショット

(1) $t_1=7/1, t_2=7/1$

ID	Ta	Td	Va	Vd	金額
001	7/1	now	7/1	now	1,000
002	7/1	7/2	7/1	now	2,000
003	7/1	7/2	7/1	now	1,000
					合計 4,000

(2) $t_1=7/2, t_2=7/1$

ID	Ta	Td	Va	Vd	金額
001	7/1	now	7/1	now	1,000
002	7/2	now	7/1	now	2,100
004	7/2	now	7/1	now	3,500
					合計 6,600

(3) $t_1=7/2, t_2=7/2$

ID	Ta	Td	Va	Vd	金額
005	7/2	now	7/2	now	1,000
					合計 1,000

図2 バイテンポラルデータベースのスナップショット
Fig.2 Snapshot of bitemporal database.

3.3 データ訂正に対する効果

図2に、図1においてトランザクション時間 t_1 と有効時間 t_2 を指定して検索した結果を示す。バイテンポラルデータベースでは、有効時間により実際の入金日を管理できる。したがって、図2の(2)に示すように、 $t_1 = 7月2日, t_2 = 7月1日$ を指定して式(2)の検索を行うことにより、7月2日に実施した変更、削除、追加の訂正を反映するとともに、7月2日の入金データを反映しない状態、すなわち訂正後の7月1日現在の状態を検索することが可能になる。

4. 基幹系システムへの適用

本章では、バイテンポラルデータベースを基幹系システムである自治体システムに適用した結果を示す。

4.1 自治体システムの概要

4.1.1 業務の構成

自治体システムは自治体の行政事務処理を支援するシステムであり、図3に示すように各種の業務システムから構成される。大きくは以下のように区分される。

- 住民情報系システム 住民票、印鑑登録証明などの住民の台帳管理、証明書発行業務
- 税関連システム 自治体管掌地方税の課税、徴収事務および税証明の発行業務
- 福祉系システム 保育所、児童手当などの資格管理、徴収、支給事務などの福祉行政業務
- 内部情報系システム 人事管理、給与支給、財務

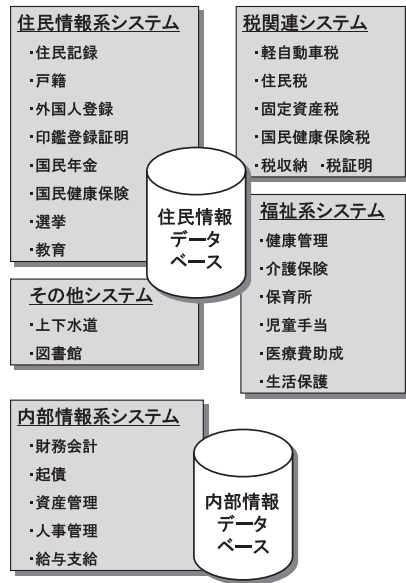


図3 自治体システムの構成
Fig.3 Composition of resident information system.

会計事務などの、自治体内部業務

4.1.2 データベースの特徴

各業務システムでは時系列のデータ履歴管理が必要になる。以下に管理する履歴データの例を示す。

- 住民の異動 出生、転入から、転居、婚姻などを経て、死亡、転出に至るまでの住民の異動履歴
- 課税対象の物件 軽自動車などの課税対象物件の登録、譲渡、廃車などの異動履歴
- 福利の資格 保険料徴収、証明書発行、給付のための資格の取得、喪失履歴
- 職員の異動・勤怠 自治体職員の異動履歴、勤怠履歴、給与支給履歴

図4に住民記録と軽自動車税のデータフローの例を示す。窓口で届出が受け付けられ、システムにオンライン入力されてデータベースに蓄積される。蓄積されたデータは、各業務で相互に活用され、統計処理、課税処理などの大量のデータ検索をとまなう処理はバッチ処理で処理される。

4.2 データベースの実装

4.2.1 実装の方針

データベースは、商用のリレーショナルデータベースを使用し、各テーブルにトランザクション時間、有効時間の属性を付加してバイテンポラルデータベースを構成した。

4.2.2 トランザクション時間の実装

トランザクション時間の単位は、トランザクション時間がデータベース上のキー属性であるために、デー

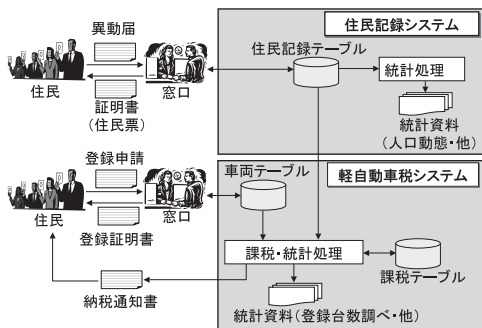


図 4 自治体システムのデータフロー

Fig. 4 Dataflow of resident information system.

タ更新の頻度から設定する必要がある。本システムでは、画面からのデータ入力に少なくとも数秒を要するため、時間の単位はこれ以下にする必要があった。このため、トランザクション時間の単位は1秒にした。

4.2.3 有効時間の実装

(1) 有効時間の単位

有効時間は業務に関連するため、業務で必要な単位で管理した。以下に有効時間の単位の例を示す。

- 時・分 職員の勤怠
- 日 住民の異動，課税対象物件の異動，福利資格の取得・喪失，口座振替期間，職員の異動
- 月 職員の給与支給情報
- 年度 軽自動車税，その他の税額

(2) 属性の表現

有効時間属性の実装にあたっては、以下の2つの表現を併用した。

- 開始時刻・終了時刻による表現
口座振替期間のようにあらかじめ有効，無効になる時刻を登録，管理する必要があるか，あるいは資格の取得・喪失のように開始，終了をセットで管理する必要のあるテーブルは，開始時刻・終了時刻で有効時間属性を表現した。
- イベント発生時刻による表現
住民票のように，出生，転居などのイベント発生により対象の状態が変化し，次のイベント発生まで前回の状態が保持されるものは，イベント発生時刻で有効時間属性を表現した。

開始時刻・終了時刻による表現では，図5の(1)に示すようにデータの変更を行う場合には，以下の操作が行われ，2レコードが追加される。図で， T_a ， T_d ， V_a ， V_d は3.1節の表記と同じである。

- $ID = 001 - 1(T_a = 6/2)$: 変更元データに削除時刻 T_d を設定
- $ID = 001 - 1(T_a = 7/3)$: 変更後データとして

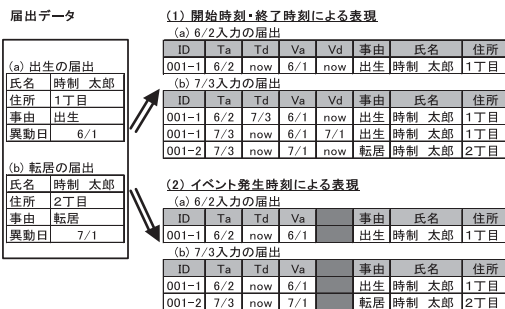


図 5 有効時間属性の表現

Fig. 5 Expression of valid time attribute.

終了時刻 V_d を設定したデータを追加

- $ID = 001 - 2$: 住所が2丁目に変更されたデータを追加

ここで， $ID = 001 - 1$ の $T_a = 6/2$ のデータはトランザクション時間が6月2日から7月2日までの状態を示す履歴データであり， $ID = 001 - 1$ の $T_a = 7/3$ は7月3日以降に業務で使用される履歴データになる。

一方，イベント発生時刻による表現では，変更の際には図5の(2)に示すように変更後のデータのみを追加することでデータの増加を抑制した。この場合，指定した時刻の実世界での状態は，指定時刻の直前の開始時刻を持つデータにより表現される。

(3) システム入力が遅れるデータへの対応

当日の業務終了後の統計を翌日報告する業務のうち，実世界の状態がシステムに即時に反映されないデータを扱うものは，自治体への届出日を基準にして統計を作成した。たとえば，住民の出生，転居，転入の異動の届出は14日以内であり，実世界の状態がシステムに反映されるまで時間を要する。しかし，人口動態などは当日の業務終了後の状態を，翌日に報告する必要がある。このため，住民記録テーブルでは有効時間である異動日とあわせて届出日を管理した。この届出日は，時刻データベースの時間軸のうち，利用者によって与えられる時間軸であるユーザ定義時間に該当する¹¹⁾。

図6にユーザ定義時間である，届出日によるスナップショットの例を示す。図で， N_a ， N_d が届出日の開始時刻と終了時刻を示す。ここで，終了時刻は次の届出の届出日になる。すなわち，届出日の時区間属性は有効時間と同様であり，有効時間と同様の検索ができる。図では届出日が6月2日であるデータに対し，6月3日に訂正を行った例を示す。図の(1)では，届出日に6月2日，トランザクション時間に6月2日を指定することで訂正前の検索結果が，(2)では届出日に同じく6月2日を，トランザクション時間に6月3日を指定することで， $ID = 002 - 1$ の変更， $ID = 003 - 1$

データベースの状況

6/2の状況 [Na, Nd]: ユーザ定義時間(届出日)

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目
002-1	6/2	now	5/31	now	6/2	now	転入	変更 次郎	2丁目
003-1	6/2	now	5/21	now	6/2	now	出生	削除 三郎	3丁目

6/3の状況

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所	(1)	(2)
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目	●	●
002-1	6/2	6/3	5/31	now	6/2	now	転入	変更 次郎	2丁目	●	●
002-1	6/3	now	5/31	6/3	6/2	6/3	転入	変更 次郎	2丁目	●	●
002-2	6/2	6/3	5/31	6/2	6/3	6/3	職権修正	変更 次郎	5丁目	●	●
003-1	6/2	6/3	5/21	now	6/2	now	出生	削除 三郎	3丁目	●	●
003-1	6/3	now	5/21	6/3	6/2	6/3	出生	削除 三郎	3丁目	●	●
003-2	6/3	now	6/3	6/3	6/3	now	職権削除	削除 三郎	3丁目	●	●
004-1	6/3	now	5/21	now	6/2	now	転入	追加 四郎	4丁目	●	●

スナップショット

(1) トランザクション時間=6/2, 届出日=6/2

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目
002-1	6/2	6/3	5/31	now	6/2	now	転入	変更 次郎	2丁目
003-1	6/2	6/3	5/21	now	6/2	now	出生	削除 三郎	3丁目

(2) トランザクション時間=6/3, 届出日=6/2

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目
002-1	6/3	now	5/31	now	6/2	now	転入	変更 次郎	5丁目
004-1	6/3	now	5/21	now	6/2	now	転入	追加 四郎	4丁目

図 6 ユーザ定義時間(届出日)によるスナップショット

Fig. 6 Snapshot by user-defined time: notification date.

の削除, $ID = 004 - 1$ の追加というすべての訂正が反映された検索結果が得られた。

4.2.4 オンライン入力の実装

窓口で受け付けた届出の内容は業務画面からオンライン入力した。この入力は業務的に以下の特性があった。

- 住民の届出により入力を行うため、同一住民に対する複数端末からの同時入力は通常業務では起こりえない。
- 窓口では操作の際に届出内容の確認を行うため入力時間が長くなる。

このため、以下の手順でトランザクション時間の開始時刻を利用した楽観的ロック¹⁶⁾を行い、ロック時間を短縮した。

- ロックを取得せずにテーブルから該当レコードを読み出し、
- 読み出したレコードを画面で修正。
- 再度、同一条件でレコードロックによりレコードを読み出し、
- レコードがロックされている場合、あるいは検索レコードの追加時刻が更新されている場合は、他で更新済のためエラーとして更新は行わない。その他の場合には、更新を実行。

4.2.5 訂正機能の実装

業務画面には届出内容の入力機能のほかに、以下の訂正機能を付加した。

- 内部処理での訂正 誤入力データに対し内部処理の段階で行う訂正
- 業務処理での訂正 通常の業務手続としてのデータの訂正

内部処理での訂正の例は、図 6 に示したトランザクション時間のみによる更新になる。図で届出日に 6 月 2 日を、トランザクション時間に 6 月 3 日を指定して

データベースの状況

6/2の状況 [Na, Nd]: ユーザ定義時間(届出日)

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目
002-1	6/2	now	5/31	now	6/2	now	転入	変更 次郎	2丁目
003-1	6/2	now	5/21	now	6/2	now	出生	削除 三郎	3丁目

6/3の状況

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所	(1)	(2)
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目	●	●
002-1	6/2	6/3	5/31	now	6/2	now	転入	変更 次郎	2丁目	●	●
002-1	6/3	now	5/31	6/3	6/2	6/3	転入	変更 次郎	2丁目	●	●
002-2	6/2	6/3	5/31	6/2	6/3	6/3	職権修正	変更 次郎	5丁目	●	●
003-1	6/2	6/3	5/21	now	6/2	now	出生	削除 三郎	3丁目	●	●
003-1	6/3	now	5/21	6/3	6/2	6/3	出生	削除 三郎	3丁目	●	●
003-2	6/3	now	6/3	6/3	6/3	now	職権削除	削除 三郎	3丁目	●	●
004-1	6/3	now	5/21	6/3	6/2	6/3	職権記載	追加 四郎	4丁目	●	●

スナップショット

(1) トランザクション時間=6/3, 届出日=6/2

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目
002-1	6/3	now	5/31	6/3	6/2	6/3	転入	変更 次郎	2丁目
003-1	6/3	now	5/21	6/3	6/2	6/3	出生	削除 三郎	3丁目

(2) トランザクション時間=6/3, 届出日=6/3以前

ID	Ta	Td	Va	Vd	Na	Nd	事由	氏名	住所
001-1	6/2	now	6/1	now	6/2	now	出生	時制 太郎	1丁目
002-1	6/3	now	5/31	6/3	6/2	6/3	転入	変更 次郎	5丁目
004-1	6/3	now	5/21	6/3	6/2	6/3	職権記載	追加 四郎	4丁目

図 7 業務処理でのデータ訂正

Fig. 7 Correction of data by feature of business.

検索することにより、6 月 2 日届出のデータとして訂正を反映した結果が得られた。この訂正は、住民票発行などで公式に入力データを使用する前に、入力データの誤りを訂正する場合に使用した。したがって、住民票に訂正履歴は出力しなかった。

一方、業務処理での訂正を行う場合は訂正履歴が必要になる。たとえば住民票では、変更を行う職権修正、削除を行う職権削除、変更を行う職権記載があり、訂正の履歴が記載された。業務処理での訂正では、データベースは図 5 に示す Na, Nd として行われる追加および変更の場合と同様に更新された。

図 7 に図 6 に示した内部処理での訂正を、業務処理での訂正として実施した例を示す。図 7 で (1) はトランザクション時間が 6 月 3 日、届出日が 6 月 2 日のスナップショットであり、図 6 の (2) に対応する。業務処理での訂正は 6 月 3 日の届出日となるため、訂正結果は検索対象にならなかった。一方、届出日による訂正の履歴が残されるため、(2) に示すようにトランザクション時間が 6 月 3 日、届出日が 6 月 3 日以前、すなわち、 $Na \leq 6/3$ なる条件を指定した場合には、訂正結果とともに訂正前の履歴も検索対象になった。

4.3 バッチ処理の運用

図 4 に示すように、入力されたデータは各種のバッチ処理によって検索される。検索処理は、以下の 2 種類に区分された。

(1) 業務終了時点の検索処理

日次、あるいは月次で業務終了時点の人口動態、登録台数調べなどの統計が行われ、翌日に結果が報告された。これらは、従来のシステム運用では窓口業務終了後に夜間バッチ処理としていた。適用システムでは前日末のトランザクション時間のデータを検索するこ

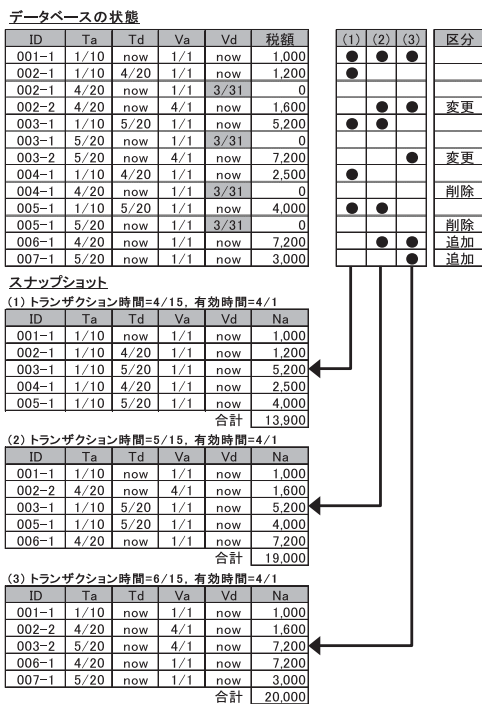


図 8 指定した有効時間での検索
Fig. 8 Query with specified valid time.

とで翌日以降の業務時間内に実行する運用とし、夜間バッチ処理を削減した。

バッチ処理では最初の段階で検索結果のチェックを行い、データの誤りが検出された場合には、データの訂正が完了した後で再処理を行った。図 6 に示すように、変更、削除、追加のすべての訂正が反映された結果が検索できた。また、自治体への届出までに時間を要するデータに関しては、4.2.3 項 (3) の届出日を使用して検索した。

(2) 有効時間を指定した検索処理

課税処理では課税の基準日が設定され、その一定期間後に課税処理が行われた。しかし、課税対象物件の異動届出は時間を要する場合が多く、入力の違い、誤りに対する課税の訂正が必要であった。

たとえば、軽自動車税の課税処理は、4月1日を基準日として行われ、以降は定期的に訂正処理が行われた。図 8 に 4月15日を課税処理日とし、以降、1カ月ごとに訂正処理を行う例を示す。有効時間を基準日の4月1日、トランザクション時間を課税処理日、あるいは訂正処理日として検索することで、該当処理日までの訂正を反映した、基準日現在の状態を検索できた。

5. 評価と考察

5.1 評価

5.1.1 訂正データの検索に関する評価

(1) 業務終了時点の検索処理

バッチ処理で誤ったデータが検出された場合にも、通常業務のオンライン入力と並行して、データの訂正および訂正を反映した再処理が可能になった。オンライン入力では業務処理での異動、訂正のほかに、トランザクション時間のみを使用して行う内部処理での訂正が行われた。この両方の更新について一貫性のある検索結果を得ることができた。

オンライン入力中にバッチ検索処理が可能になったため、夜間バッチ処理と、残業時間に及ぶオンライン入力の完了待ちなどの夜間バッチ処理に付帯する作業が削減でき、夜間のシステム運用負荷が削減された。さらに、処理の重複する日には緊急性の低い処理を後日実施するなど、柔軟にバッチ処理のスケジュールを組めるようになった。

なお、図 6 に示す住民の異動のように、実世界のデータは必ずしも即時にシステムへは反映されなかった。このようなデータの統計処理のうち、当日の業務終了時点の結果を翌日に報告する必要があるものは、有効時間ではなくユーザ定義時間である届出日を使用することによって当日のデータの状態を検索できた。

(2) 長期にわたる訂正の検索

図 8 に示す課税処理のように、長期にわたり訂正データを管理し、異動統計によって訂正の状況を把握する必要のある業務があった。このような業務に対しても変更、削除、追加のすべての訂正を反映した検索が可能であり、異動統計として容易に前回と今回の集計、および異動の合計を検索することができた。また、この検索処理もスナップショットを利用しており、オンライン入力中に実行することができた。

(3) 訂正データの管理

オンライン入力されたデータは内部で確認を行ってから、業務処理で利用される。住民記録システムなど履歴を管理する業務システムでは、この段階での訂正履歴は業務として不要であるため、内部処理での訂正が行われた。一方、業務処理で利用された後は、業務での訂正が行われた。パイテンポラルデータベースにより、内部処理での訂正と、業務処理での訂正の両方を管理することができた。

5.1.2 データベースの実装に関する評価

パイテンポラルデータベースとしての履歴管理により、データ量の増大と、検索機能の複雑化という課題

が発生した。

データ量については、図5の(1)に示すようにデータ変更の際には、変更前、変更後の2件のデータを追加する必要があった。これは、図5の(2)に示すイベント発生時刻による表現を採用することで1件の追加に削減できた。このため、特に有効時間の終了時刻を管理する必要のあるテーブルを除いては、イベント発生時刻による表現を採用した。これにより、業務で必要とする履歴情報に対するデータ量の増加は、内部処理での訂正データ分の増加のみになった。適用システムのテーブルでは、この増加は最大でも年に20%程度であり、5年間のライフサイクルで2倍以下だった。なお、近年は記憶媒体の単位容量の価格が下がっており、データ量の増大は構築コストの面での問題にはならなかった。

また、検索機能は複数のテーブルの結合処理などで複雑化したが、モジュール言語¹⁵⁾を使用することにより商用リレーショナルデータベースの機能範囲で構築できた。

5.2 考 察

(1) 業務終了時点の検索処理

バイテンポラルデータベースでは、オンライン入力中であっても、訂正データを反映した一貫性のある検索結果が得られることが分かった。この機能により、業務終了後に実施していたバッチ処理を翌日以降の業務時間内に実施するなど、システム運用面の効果があることが確認できた。

たとえば、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関連作業が削減できた。さらに、夜間の運用がなくなったため、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、バッチ検索処理の依頼あるいは帳票の受取りなどの連絡・調整に要する作業を効率化できた。近年、電子申請、電子商取引などの進展でノンストップサービスが拡大しており、このような分野ではオンライン入力を停止することなくバッチ検索処理を行う運用は有効と考えられる。

また、実際の業務運用では実世界の状態が即時にシステムに反映されるとは限らないため、業務によっては実世界におけるデータの有効時間とは別に、ユーザ定義時間である届出日を管理する必要がある。たとえば、自治体システムでは住民の年齢計算は有効時間である誕生日、人口動態の統計は届出日を使用しており、業務内容により両方の時間が使い分けられた。

(2) 長期にわたる訂正の検索

長期にわたる訂正が発生するデータの検索処理では、

指定した有効時間を基準時刻として、この時点のデータがシステム内でどのように訂正されてきたかという管理が行われる。バイテンポラルデータベースは、基準時刻に対応する有効時間と、システム内の状態を示すトランザクション時間の双方を管理しており、長期にわたるデータの経緯把握や、異動統計の作成に効果があることが確認できた。

(3) 訂正データの管理

バイテンポラルデータベースでは、業務での訂正のほか、内部処理での訂正についても訂正履歴を管理できた。内部処理での訂正は、外部的には履歴を残さないが、データベース内部ではトランザクション時間により訂正履歴が蓄積される。したがって、データの変更経緯の管理や、高い監査性を必要とするシステムで有効と考えられる。

6. む す び

本論文では、バイテンポラルデータベースを基幹系システムに適用し、実際のシステム運用において発生する誤入力や入力遅れなどの訂正についても、オンライン入力中に訂正を反映した一貫性のある検索結果が得られること確認した。さらに、短期間のデータ訂正だけでなく長期にわたるデータ訂正をとまなう業務に対しても、異動統計などの訂正データの把握に効果があること、データの訂正履歴を、内部処理での訂正と業務処理での訂正の両面から管理できるという効果があることが確認できた。

参 考 文 献

- 1) Bækgaard, L. and Mark, L.: Incremental Computation of Time-Varying Query Expressions, *IEEE Trans. Knowledge and Data Eng.*, Vol.7, No.4, pp.583-590 (1995).
- 2) Bhargava, G. and Gadia, S.K.: Relational Database Systems with Zero Information Loss, *IEEE Trans. Knowledge and Data Eng.*, Vol.5, No.1, pp.76-87 (1993).
- 3) Edelweiss, N., Hübler, P.N., Moro, M.M. and Demartini, G.: A Temporal Database Management System Implemented on top of a Conventional Database, *Proc. XX International Conference of the Chilean Computer Science Society*, pp.58-67 (2000).
- 4) Jensen, C.S., Mark, L. and Roussopoulos, N.: Incremental Implementation Model for Relational Database with Transaction Time, *IEEE Trans. Knowledge and Data Eng.*, Vol.3, No.4, pp.461-473 (1991).
- 5) Jensen, C.S. and Snodgrass, R.T.: Temporal

- Data Management, *IEEE Trans. Knowledge and Data Eng.*, Vol.11, No.1, pp.36-44 (1999).
- 6) Park, H.-J. and Yoo, S.I.: Implementation of Checkout/Checkin Mechanism on object-Oriented Database System, *Proc. 7th International workshop on Database and Expert System Applications*, pp.298-303 (1996).
 - 7) Shrira, L. and Xu, H.: SNAP: Efficient Snapshots for Back-in-Time Execution, *Proc. 21st International Conference on Data Engineering*, pp.434-445 (2005).
 - 8) Stantic, B., Thornton, J. and Sattar, A.: A Novel Approach to Model NOW in Temporal Databases, *Proc. 10th International Symposium on Temporal Representation and Reasoning and Fourth International Conference on Temporal Logic*, pp.174-180 (2003).
 - 9) Tansel, A.U.: Integrity Constraints in Temporal Relational Databases Extended Abstract, *Proc. Int. Conf. on Information Technology: Coding and Computing*, pp.460-464 (2004).
 - 10) Terenziani, P.: Symbolic User-Defined Periodicity in Temporal Relational Databases, *IEEE Trans. Knowledge and Data Eng.*, Vol.15, No.2, pp.489-509 (2003).
 - 11) 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*, Vol.42, No.10, pp.972-979 (2001).
 - 12) 大山敬三: 情報検索システムのオンライン更新における一貫性維持方式, *学術情報センター紀要*, No.10, pp.29-35 (1998).
 - 13) 北川博之, 田中 肇, 大保信夫, 鈴木 功: 履歴データ型を用いた版管理データモデルの提案, *情報処理学会論文誌*, Vol.34, No.5, pp.1031-1044 (1993).
 - 14) 徐 海燕, 古川哲也, 史 一華: マルチバージョンデータベースにおけるワークフローに基づく並行処理制御, *情報処理学会論文誌: データベース*, Vol.42, No.SIG4, pp.27-35 (2001).
 - 15) 西川洋一: 最新データベース技術, *秀和システム* (2002).
 - 16) Gray, J. and Reuter, A. (著), 喜連川優 (監訳): *トランザクション処理—概念と技法*, 日経 BP (2001).
 - 17) 春本 要, 八幡 孝, 西尾章治郎: 協調作業支援のためのデータ管理モデル, *信学論 (D-I)*, Vol.J79-D-I, No.5, pp.271-279 (1996).
 - 18) 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).

(平成 18 年 12 月 20 日受付)

(平成 19 年 4 月 6 日採録)

(担当編集委員 春本 要)



工藤 司 (正会員)

1980年北海道大学大学院工学研究科修士課程修了。同年三菱電機(株)入社。並列処理アーキテクチャの研究, 業務パッケージソフトウェアの開発に従事。2005年より三菱電機インフォメーションシステムズ(株)で業務システムの開発に従事。技術士(情報工学)。電子情報通信学会会員。



片岡 信弘 (正会員)

1968年大阪大学大学院工学研究科修士課程修了。同年三菱電機(株)入社。汎用機ミドルウェア, 業務パッケージソフトウェアの開発, 情報システムのERPパッケージによる革新の推進に従事, 2000年東海大学教授, 現在情報理工学部情報メディア学科教授。博士(情報科学), 技術士(情報工学)。ビジネスプロセスモデリングによるシステム開発の研究に従事。本会理事(1996~1998年), IEEE, 電子情報通信学会, 経営情報学会各会員。



水野 忠則 (フェロー)

1945年生。1969年名古屋工業大学経営工学科卒業。同年三菱電機(株)入社。1993年静岡大学工学部情報知識工学科教授。1996年情報学部情報科学学科教授。2006年より創造科学技術大学院長。工学博士。情報ネットワーク, モバイルコンピューティング, 放送コンピューティングに関する研究に従事。著訳書としては『コンピュータネットワーク』(日経BP), 『モダンオペレーティングシステム』(ピアソン・エデュケーション)等がある。電子情報通信学会, IEEE, ACM 各会員。情報処理学会監事。