社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DBS－138（27）
2006－GN－58（27）
2006／1／27

連邦型分散データベースシステムの構成法に関する研究
－属性・属性値・個体統合による DB 統合，及び更新について－

吉田 祥子　　岡林 誠治　　吉田 香　　打浪 清一

九州工業大学情報工学部 〒820-8502 福岡県飯塚市川津 680-4

E-mail:　{yoshida, okabayashi}@taurus10.cse.kyutech.ac.jp, kaori@ai.kyutech.ac.jp,
uchinami@cse.kyutech.ac.jp

あらまし　近年，インターネット網を利用して既存のデータベースシステムのサービスを継続させながら複数のデータベースシステムの情報資源を共有化・連携利用するシステムへの要求が高まっている．しかし，個々に設計されたデータベースではそれぞれの主観や目的によりデータ表現や内部構造が違うため，統合利用する上で異種性が問題となる．そこで我々は，個々のデータベース間の異種性を解決して，それらが持つ情報を統合利用することのできる連邦型分散データベースシステムの開発を行っている．本稿ではデータベース間の異種性を解決して統合を行う属性統合・属性値統合・個体統合システム，及び複数データベースを一括更新できる更新システムの試作・検証について報告する．

キーワード　連邦型データベース，データ統合，属性，属性値，個体，シソーラス，更新

Study on Federation of Distributed Database Systems
－Database Integration by Attribute, Attribute Value, and Entity Integration, and Update System－

Shoko YOSHIDA　Seiji OKABAYASHI　Kaori YOSHIDA　and　Seiichi UCHINAMI

Kyusyu Institute of Technology 680-4 Kawazu, Iizuka, Fukuoka, 820-8502 Japan

E-mail:　{yoshida, okabayashi}@taurus10.cse.kyutech.ac.jp,　kaori@ai.kyutech.ac.jp,
uchinami@cse.kyutech.ac.jp

Abstract　Recently, the demand for the database system which enable can re-use and integrate two or more databases has increased. However, it's difficult to integrate variety of data architecture. Then, we have developed the federation of distributed database system which can use integration by solving the heterogeneity of each database. In this report, we report about the attribute integration system, the attribute value integration system, and the entity integration system which can use integration by solving the heterogeneity of each database. Moreover, we report about the update system which can update two or more databases with batch processing.

Keyword　Federal Database，Data Integration，Attribute，Attribute Value，Entity，Thesaurus，Update

1.　はじめに

既存のデータベースシステムのサービスを継続させながら，複数のデータベースシステムの情報資源を共有化・連携利用したいという要求がある．しかし，各データベース(以下 DB と表記)では，それぞれの主観や目的によりデータ表現や内部構造が違うため，統合利用する上で異種性が問題となる．そこで我々は，各 DB 間の異種性を解決して，それらが持つ情報を統合利用できる連邦型分散データベースシステムの開発を行っている．

本稿では，各 DB の異種性を解決し，DB 統合を行う属性統合・属性値統合・個体統合システムに関する試作及び検証結果について報告する．また，各 DB を統合した連邦型分散 DB システムを利用した，複数の DB の一括更新を行う更新システムに関しても試作及び検証結果について報告する．

2.　連邦型分散 DB システムの構成

連邦型分散 DB システムは，エージェント群と構成源 DB，システム DB により構成される(図 1)．連邦型 DB 自身は実データを持たず，必要な実データ

は構成源 DB から取得する．構成源 DB の統合は，属性統合，属性値統合，個体統合の三段階統合で行う．連邦型 DB 自身が独自に保持するシステム DB には，構成源 DB の統合に必要な情報が格納される．エージェント群は，構成源 DB の統合に必要な各操作を独立して専属的に行うエージェントの集まりである．

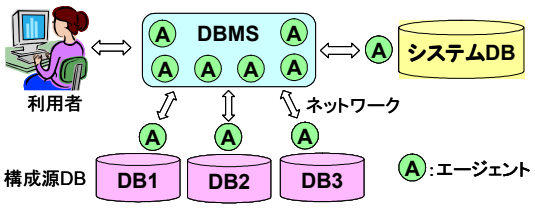

図 1：連邦型分散 DB システムの構成

3. データベース統合システム

構成源 DB の統合は，各構成源 DB のメタデータを用いて行う．連邦型分散 DB に参加する構成源 DB は，その概念スキーマ情報等のメタ情報を XML で記述し，提供することを前提としている．

統合される構成源 DB のデータは，それぞれにおいて属性名や属性の表現方法が異なるため，属性・属性値の統一を行う必要がある．また，各構成源 DB に存在する同一個体データを統合する必要がある (図 2)．

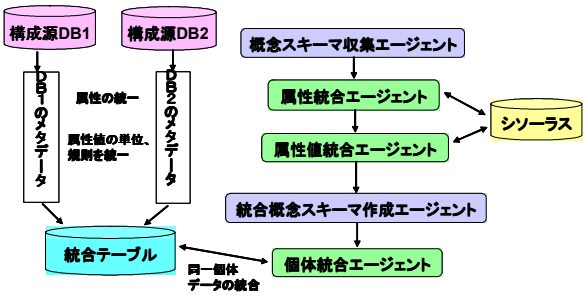

図 2：統合処理の流れ

3.1. 属性統合・属性値統合

属性統合及び属性値統合には，シソーラスという類義語集を利用する．市販のシソーラスでは，本研究に必要な情報が不足しているため，図 3 に示すような条件を満たすシソーラスを作成した．シソーラスは RDF で記述した．

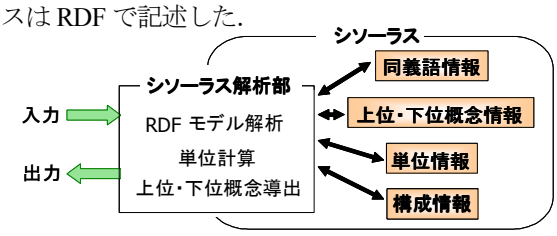

図 3：作成したシソーラス

属性統合では，シソーラスを用いて各構成源 DB において異なる属性名の統一を行う．シソーラスに格納されている同義語，上位語，下位語の情報を利用して変換・統一を行う．

属性統合を行った後は属性値統合を行う．属性値統合では，属性値の単位，表記法をシソーラスに格納された情報を基に統一する．身長や年齢等の単位や，日時・住所・氏名等の表記法を統一することで，構成源 DB 間の比較・統合を可能にする．

これら属性統合・属性値統合を行うことで各構成源 DB 間の異種性が解決され，構成源 DB を統合した統合テーブルが作成される．

3.2. 個体統合

属性統合・属性値統合を行って各構成源 DB を統合した統合テーブルでは，複数の DB の情報を統合しているため，同一の個体を表すデータが複数存在する場合がある．このため，同一個体データの統合を行う個体統合を行う必要がある．本研究では，知識ベースのプロダクションシステムの考えを用いて同一の個体を表すデータであるかどうかを判定し，同一個体データを統合する個体統合システムを試作した．

3.2.1. 個体統合処理

個体統合の処理の流れは図 4 のようになる．

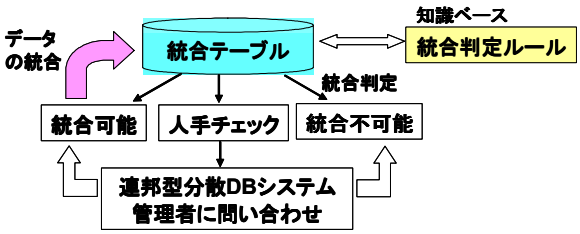

図 4：個体統合処理の流れ

まず，知識である統合判定ルールを参照し，統合テーブルのデータの統合判定を行い，統合可能・統合不可能・人手によるチェックの三種類の判定結果に分類する．判定結果が人手によるチェックとなったデータに関しては，連邦型分散 DB システムのシステム管理者に統合可・不可を尋ね，統合の判断を行う．システム管理者による統合の判断が済むと，統合テーブルの全てのデータは統合可能・統合不可能の二種類に分類されるので，統合可能と判定されたデータの統合を実際に統合テーブルに対して行う．この判定と統合までの一連の処理により新しく追加された事実を基に，さらに判定及び統合の処理を行い，同一の個体を表すデータが存在しなくなるまで一連の処理を繰り返す．

統合判定の結果は以下の三種類とする．

- 統合可能
  同一の個体を表すデータと判定された場合
- 統合不可能
  異なる個体を表すデータと判定された場合
- 人手によるチェック
  構成源DBの持つ情報だけでは統合判定が確実ではない場合に，人手によるチェックを行う．人手によるチェックという判定結果を取り入れることで，精度を考慮した個体統合処理を実現することができる．

3.2.2. 統合判定ルール

統合判定には，個体を判別するための一意性の高い属性項目を組み合わせた判定ルールを用いる．判定ルールには，以下のような属性項目を用いる．

(1) 個体ごとにその属性値の一意性が高く，個体を判別でき得るものであり，統合を行っている構成源DBに共通して存在している属性項目（例：住所，電話番号，生年月日など）

(2) 統合を行っている全ての構成源DBに共通して存在する属性ではないが、個体を一意に判別することができる属性項目（例：個体ごとに付けられた番号項目など）

これらの属性項目を組み合わせることにより，判定ルールを作成する．例として氏名，住所，電話番号という属性項目を用いて作成した判定ルールの一部を表1に示す．比較不可能とは，比較しているデータの内どちらか一方でもその項目の属性値がnullであった場合のことである．

表 1：判定ルールの例(一部)

氏名	住所	電話番号	判定結果
一致	一致	一致	統合可能
一致	不一致	不一致	統合不可能
一致	一致	不一致	人手チェック
一致	一致	比較不可能	人手チェック

作成した統合判定ルールは，XMLにより記述する．また，判定ルールは知識として処理部とは別に連邦型システム上に格納されているため，判定ルールの変更・追加は容易になっている．このため，判定ルールを変更することで，様々な場合の個体統合に対応することができるシステムとなっている．

3.2.3. 多値属性項目を用いた統合判定

前節までに述べた個体統合処理では，多値属性項目を用いずに統合判定を行っていた．しかし，住所や電話番号のように統合判定に用いる項目の内，同一人物であっても複数の値を持つ項目が存在する場合が考えられる．このため，多値属性項目を統合判定に用いるシステムを設計した．

統合判定に多値属性項目を用いる場合，判定結果を人手によるチェックとする判定ルールが増えることになる．例えば，多値属性を用いない統合判定では，表1のように氏名が一致していても，住所と電話番号の値が不一致であった場合は統合不可能と判定していた．しかし，電話番号が多値属性項目であった場合に，値が不一致であっても同一人物の多値属性値を比較している場合も考えられるため，統合不可能と判定できず人手によるチェックと判定する必要がある．このような理由から，多値属性項目を統合判定に利用する場合は，人手によるチェックとなる判定ルールが増えることになる．

このような人手チェックが増えることによる作業効率の低下を改善し，精度の高い統合判定を行うために，多値属性項目を用いた統合判定では事実ベースを作成・利用する．事実ベースとは，同一個体の持つ多値属性値を格納したものである．例えば電話番号が多値属性項目である場合に，ある人物の固定電話番号と携帯電話番号といった多値属性値の事実を格納するものである．

よって，多値属性項目を用いた個体統合処理は図5のようになる．前節までと同様に，統合判定ルールを参照して統合判定を行うが，その際に事実ベースを参照し利用する．また，統合判定により統合可能と判定されたデータにおいて新たな事実が判明した場合は，事実ベースに追加する．多値属性項目が存在しない場合，同一個体データの統合処理は統合テーブルに存在する複数タプルを一つのタプルに統合することで行う．しかし，多値属性項目が存在する場合の同一個体データの統合処理は，統合テーブルにおけるIDを等しくすることで行う．

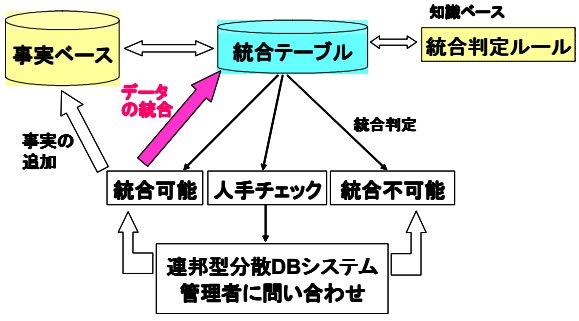

図 5：多値属性項目を用いた個体統合処理の流れ

4. 更新システム

連邦型分散 DB システムは複数の構成源 DB を統合しているため，連邦型システム上の統合テーブルを用いた更新はすべての構成源 DB に反映されなければならない．またこの時，データの整合性を保つために構成源 DB に対する更新処理は一斉に行われる必要がある．そこで，更新システムでは二相コミットを利用した更新処理を行う．

また，更新システムでは各構成源 DB における利用者の権限に違反しないように更新処理を行わなければならない．そこで，各構成源 DB における利用者の更新権限を適用し，それに沿った更新処理を行う．各構成源 DB における利用者の更新権限は，権限適用エージェントにより知ることができる．このエージェントは，各構成源 DB における利用者の権限情報を収集し，連邦型分散 DB システムにおける利用者の権限を決定するエージェントである．

以上のことから，更新システムの処理の流れは以下のようになる．

① 利用者から更新要求を受け取る
② 統合テーブルに対する更新 SQL 文を作成
③ 統合テーブルを構成する各構成源DBに対応した更新 SQL 文を作成
④ 権限適用エージェントへ利用者の更新権限を問い合わせる
⑤ 各構成源DBにおける利用者の更新権限に沿った更新処理

4.1. 各構成源 DB に対応した SQL 文の作成

利用者は，連邦型 DB 上の統合テーブルに対して更新要求を行うため，各構成源 DB に対応した更新 SQL 文をシステム内で作成しなければならない．統合テーブル作成には属性統合・属性値統合・個体統合を行っているため，これらの統合を元に戻し更新 SQL 文を作成する必要がある．

各構成源 DB に対応した SQL 文を作成するには，まず更新要求のあった統合テーブルのタプルがどの構成源 DB の情報をもとに統合されたのか，統合元を調べなければならない．以下の 3 つのデータベースを参照して統合元を調べる．

- 個体統合バインドデータベース
  個体統合システムにおいて統合可能と判定され，統合が行われたデータの組を記録したデータベースである．個体統合前の統合テーブルの ID を利用して，どの ID のデータが同一個体データであったのかを記録している．
- 構成源参照データベース
  属性統合・属性値統合を行い作成された統合テーブルにおいて，統合テーブルの ID 毎にどの構成源DBから収集したデータであるかを記録したデータベースである．
- バインド表
  統合テーブル作成の際に，統合テーブルの各属性項目がどの構成源DBの属性項目をバインドしたのかを記録したデータベースである．

更新対象タプルの統合元情報が分かった後は，属性値統合エージェントにより，更新を行う各構成源 DB に対応した属性値へ更新データを逆変換する必要がある．これは，統合テーブル作成時に属性値を共通表現に統一しており，利用者はこの共通表現で更新要求を行うため，各構成源 DB に対応した表現方法へ変換する必要があるためである．これらの処理を行うことで，各構成源 DB に対応した更新 SQL 文を作成することができる．

4.2. 権限に沿った更新処理

各構成源 DB に対応した SQL 文を作成した後は，権限適用エージェントに問合せを行い，各構成源 DB における利用者の権限情報を取得する．利用者の更新権限には，以下の 3 種類がある．

- 更新権限がある場合
  構成源 DB において利用者がユーザ ID を持っている場合，利用者毎に更新権限の有無が決められている．このような場合に，構成源 DB の DBA から利用者に対して更新権限が与えられている場合のことである．
- 更新権限がない場合
  上記と同様に構成源DBにおいて利用者がユーザ ID を持っている場合において，その利用者が構成源 DB の DBA から更新権限を与えられていない場合のことである．
- 構成源 DB にユーザ ID がない場合
  構成源 DB において利用者のユーザ ID が存在しない場合のことである．例えば，市役所 DB のように一般の利用者がユーザ ID を持たない DB も存在するためである．この場合は，利用者が更新を行うことはできず，更新権限がないことになる．

更新システムでは，上記の 3 種類の更新権限に沿った更新処理を行う．各更新処理について次で述べる．

4.2.1. 更新権限がある場合

構成源 DB において更新権限がある場合は，二相コミットを用いて更新処理を行う(図 6).

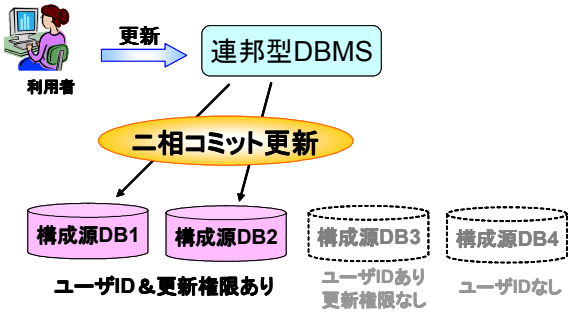

図 6：更新権限がある場合の更新処理

まず第一相では，連邦型分散 DB システム内で作成した各構成源 DB に対応した更新 SQL 文をそれぞれの構成源 DB に対して実行し，その返答を受け取る．次に第二相では，全ての構成源 DB からの返答が成功であった場合，コミット命令を出して更新処理の確定を行う．しかし，構成源 DB からの返答が一つでも失敗であった場合は，全ての構成源 DB に対してロールバック命令を出して更新処理の取り消しを行う．このような二相コミットを用いて更新処理を行うことで，統合テーブルを構成する全ての構成源 DB に対し一括更新を行うことが可能となる．

4.2.2. 更新権限がない場合

更新権限のない構成源 DB に対しては，利用者が更新を行えないため，構成源 DB の DBA に更新内容を知らせて更新の依頼を行う．また，連邦型分散 DB システム内のシステム DB に存在する更新データ抄録表に，依頼を行った更新の内容を記録・保持しておく(図 7).

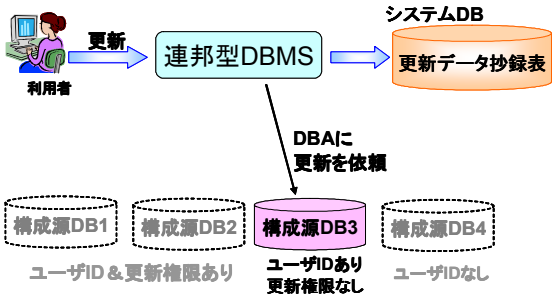

図 7：更新権限がない場合の更新処理

4.2.3. ユーザ ID がない場合

利用者がユーザ ID を持たない構成源 DB に対しては，利用者は更新を行えず，構成源 DB の DBA のみが更新を行えるようになっている．このため，構成源 DB の DBA に対して更新依頼を行う．更新の依頼には，通常この構成源 DB に対して更新要求を行う際に提出するようなフォーマットを作成し，更新依頼を行うことにする．例えば，市役所 DB の場合は婚姻届等のフォーマットである．また，連邦型分散 DB システム内のシステム DB に存在する更新データ抄録表に，依頼を行った更新の内容を記録・保持しておく(図 8).

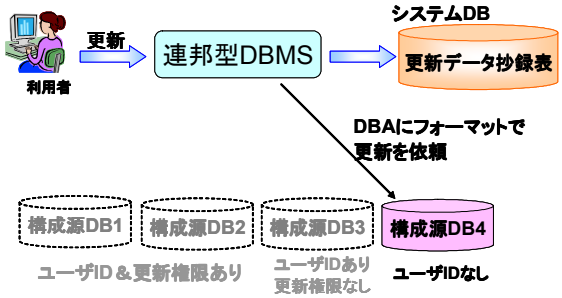

図 8：ユーザ ID がない場合の更新処理

4.3. 更新データ抄録表

利用者が直接構成源 DB に対して更新を行えず，構成源 DB の DBA に対して更新依頼を行った場合に，その更新内容を連邦型分散 DB システム上で保持するためのテーブルである. 更新データ抄録表は，更新対象の構成源 DB 名や更新 SQL 文，また更新依頼を行った日付，構成源 DB において更新が完了した日付等を格納するテーブルとなっている.

利用者が更新要求を出した後，更新対象となっていたタプルを再度利用する場合に更新データ抄録表を利用する．更新データ抄録表の更新完了日を格納する項目が null の場合は，構成源 DB において更新が実行されておらず，古いデータのままであることが分かる．この場合は，最新データである更新データ抄録表に格納した更新内容を用いて処理を行い，利用者には構成源 DB において前回の更新処理が完了していないことを提示する.

5. 試作システム

連邦型分散 DB システムは, 既存の DB を統合し，その統合テーブルを用いて一括処理を行うシステムを目的としている．そのため今回は，結婚手続きに関する会社や市役所, 保険組合の DB を統合処理し，そして更新処理を行うシステムを試作した．このシステムは，結婚する際の届出や住所，氏名変更等の手続きの処理を一括して行えるようにしたもので，会社，市役所，保険組合のデータを統合利用することで迅速な処理を行うことができる．統合利用する構成源 DB には，会社 1DB，会社 2DB，市役所 DB，

会社 1 の保険組合 DB，会社 2 の保険組合 DB の 5 つを用いた．

5.1. 属性統合

属性統合では，各構成源 DB において異なる属性名の統一を行った．各構成源 DB に対して属性統合を行った結果(抜粋)を図 9 に示す．図 9 において矢印で示された部分が属性名の変換を行った部分である．

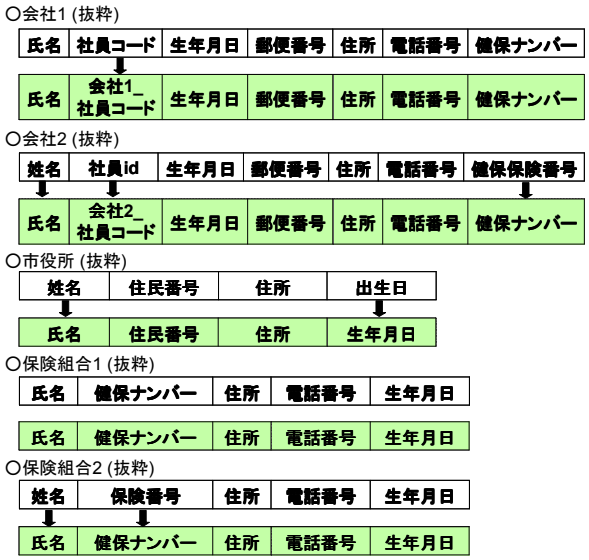

図 9：属性統合の結果(抜粋)

5.2. 属性値統合

属性統合を行った後は属性値統合を行う．属性値統合では，各構成源 DB において異なる属性値の表記法を統一した．属性値統合の結果(抜粋)を図 10 に示す. 図 10 において矢印で示された部分が属性値の変換を行った部分である．

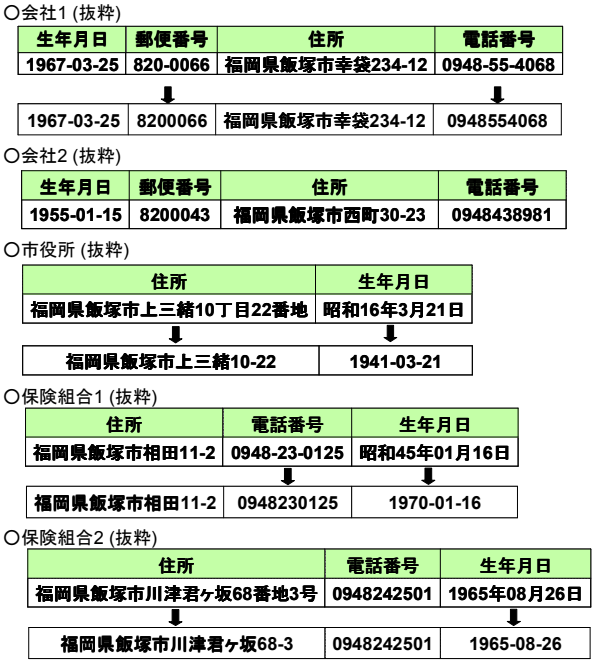

図 10：属性値統合の結果(抜粋)

5.3. 個体統合

属性統合・属性値統合を行ったことで各構成源 DB の異種性が解決され，統合される．作成された統合テーブルを図 11 に示す．図 11 の統合テーブルでは同じ氏名のデータが複数存在しており，このことから同一の個体を表すデータが複数存在していることが考えられる．このような同一個体データを統合するため, 図 11 の統合テーブルに対して個体統合を行った．

id	会社1_社員コード	会社2_社員コード	住民番号	健保ナンバー	氏名	郵便番号	住所	電話番号	生年月日
32	null	null	60092730	null	原口智弘	null	福岡県飯塚市本町2-5	null	1959-10-05
34	null	null	16204631	null	市原伸洋	null	福岡県飯塚市柏の森1-9	null	null
52	null	null	null	35-225694	阿藤純一	null	福岡県飯塚市川島5-12	0948243014	1972-01-16
65	null	null	null	92-182938	原口智弘	null	福岡県飯塚市菰田4-1	0948231605	1963-12-10
102	13125692	null	null	35-125692	市原伸洋	8200011	福岡県飯塚市柏の森1-9	0948223015	1972-08-29
103	13225694	null	null	35-225694	阿藤純一	8200002	福岡県飯塚市川島5-12	0948243014	1972-01-16

図 11：属性・属性値統合後に作成された統合テーブル(抜粋)

id	会社1_社員コード	会社2_社員コード	住民番号	健保ナンバー	氏名	郵便番号	住所	電話番号	生年月日
26	null	9562958785	67890227	92-985678	太田啓介	8200067	福岡県飯塚市大字川津字君ヶ坂68-3	0948242501	1965-08-26
27	null	9479687373	98680058	92-875806	村田陽子	8200043	福岡県飯塚市西町9-45	0948220420	1969-11-16
32	null	null	60092730	null	原口智弘	null	福岡県飯塚市本町2-5	null	1959-10-05
34	13125692	null	16204631	35-125692	市原伸洋	8200011	福岡県飯塚市柏の森1-9	0948223015	1972-08-29
52	13225694	null	16204639	35-225694	阿藤純一	8200002	福岡県飯塚市川島5-12	0948243014	1972-01-16
65	null	null	null	92-182938	原口智弘	null	福岡県飯塚市菰田4-1	0948231605	1963-12-10

図 12：個体統合後の統合テーブル(抜粋)

まず，統合判定に用いる属性項目を決め，判定ルールを作成した. 判定に用いる属性項目は 3.2.2 節で述べた(1)，(2)に基づいて決める．3.2.2 節の(1)に基づき，図 12 の統合テーブルのうち「氏名，住所，生年月日」を判定に用いる．また，3.2.2 節の(2)に基づき，「健保ナンバー」も判定に用いる．健保ナンバー項目は，市役所 DB には存在していないため，全ての構成源 DB に存在する属性項目ではないが，会社及び保険組合 DB において個体を一意に判別できる項目となっているため, 3.2.2 節の(2)に基づいて判定に用いることにした．これら 4 つの属性項目を用いて作成した判定ルールが表 2 である．表 2 では簡略化のため，一致を○，不一致を×，比較不可能を？として表現している.

表 2：試作システムにおける判定ルール(抜粋)

氏名	健保ナンバー	住所	生年月日	統合判定
○	○			統合可能
○	×			統合不可能
○	？	○	○	統合可能
○	？	○	×	人手によるチェック
○	？	○	？	人手によるチェック
○	？	×	○	人手によるチェック
○	？	×	×	統合不可能

図 11 を用いた個体統合処理の代表例を，以下に 3 種類示す.

- 統合可能と判定される場合
  図 11 中の id=52,103 のデータは，氏名が「阿藤純一」で等しく，健保ナンバーの値も等しいので，表 2 の 1 番目に示された判定ルールより統合可能と判定され，統合処理が行われる.
- 統合不可能と判定される場合
  図 11 中の id=32,65 のデータは，氏名は等しいが，住所と生年月日の値が異なっているため，表 2 の 7 番目に示された判定ルールより統合不可能と判定される．よって，同姓同名の別の人物のデータであることが分かる.
- 人手によるチェックと判定される場合
  図 11 中の id=34,102 のデータは，氏名と住所が等しく，生年月日が id=34 において null のため比較ができないため，表 2 の 5 番目の判定ルールより人手によるチェックと判定される．このデータは人手チェックにより同一個体データと判定された.

以上のような統合判定を行い，同一個体データの統合処理を行った結果が図 12 である. 個体統合を行ったことで, 図 12 では同一の個体を表すデータが統合され，各構成源 DB のデータが統合利用可能となった.

5.4. 更新システム

今回は，図 12 中の「太田啓介」と「村田陽子」という人物が結婚をした場合の更新処理を行った．更新する情報は,「姓 : 太田(女性の情報のみ姓を更新)」,「住所 : 福岡県太宰府市青葉台 5-2-3」,「電話番号 : 0929230454」,「郵便番号 : 8180137」である.

5.4.1. 構成源 DB に対応した SQL 文作成

利用者から図 12 の統合テーブルへの更新要求を用いて, 各構成源 DB に対応した SQL 文を作成する必要がある． 図 13 の個体統合バインド DB 及び図 14 の構成源参照 DB から,更新対象のタプルの統合元を調べることができる。

id	integrated_id
26	73,114

図 13：個体統合バインドデータベース(抜粋)

id	dbname	tablename	id_koumoku	memberid
26	jdbc:postgresql://ocha.cse.kyutech.ac.jp/Shiyakusho	市民統合表backup	住民番号	67890227
114	jdbc:postgresql://192.168.1.50/Kaisya2	会社2社員	社員id	9562958785
73	jdbc:postgresql://ocha.cse.kyutech.ac.jp/Hoken2	保険加入者表	保険番号	92-985678

図 14：構成源参照データベース(抜粋)

図 13 の id 項目は統合テーブル上の id を表しており，integrated_id 項目に記されている id データを統合したことを意味している．つまり図 13 から，統合テーブル id=26 のデータは，id=73,114 を個体統合したことが分かる．また，図 14 には id 毎の構成源 DB の情報を記録している．このことから，図 13 及び図 14 より統合テーブル上の男性データ id=26 は，市役所 DB，会社 2 の保険組合 DB，会社 2DB のデータを統合したことが分かる．さらに，更新するデータを属性値統合エージェントにより各構成源 DB に対応した表現方法へ逆変換し，各構成源 DB に対応した更新 SQL 文を作成した結果が図 15 である．女性データに関しても同様に統合元の構成源 DB を調べ, 更新 SQL 文を作成した. 女性データも男性と同様に, 市役所 DB，会社 2 の保険組合 DB，会社 2DB のデータを統合したものである.

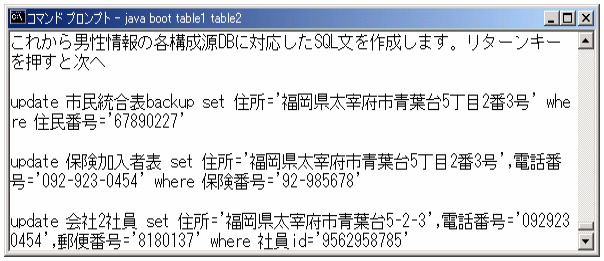

図 15：男性に関する各構成源 DB 更新 SQL 文

5.4.2. 更新結果

図 15 のように作成された SQL 文を用いて更新処理を行った．男性は会社 2 の保険組合 DB 及び会社 2DB に対して更新権限を持っており，女性は会社 2DB のみ更新権限があり，保険組合 DB には更新権限を持っていなかった．また，男女とも市役所 DB に対してはユーザ ID を持っていなかった．これらの更新権限に沿って更新処理を行った．

図 16 には会社 2DB の更新結果を，図 17 には会社 2 の保険組合 DB の更新結果を示す．会社 2DB には男女とも更新権限を持っていたため更新処理が実行されているが，会社 2 の保険組合 DB には男性しか更新権限を持っていなかったため，男性データのみ更新が実行されている．女性データは更新されていないことが確認できる．女性データに関しては，会社 2 保険組合 DB の DBA に対して更新依頼を行った．

また，更新権限がなかった場合及びユーザ ID がなかった場合には，構成源 DB の DBA に更新を依頼し，更新データ抄録表に更新内容を記録する．図 18 に更新データ抄録表を示す．更新権限のなかった女性の保険組合 DB に対する更新，及び男女ともユーザ ID を持っていなかった市役所 DB に対する更新の内容が記録されていることが確認できる．

以上のことから，複数の DB に対する更新処理を一括で行えるシステムを試作できたことが確認できる．また，更新処理を行う際には利用者の更新権限に沿った処理を行うことができた．

6. おわりに

既存のデータベースシステムを統合・利用できる連邦型分散データベースシステムの試作・検証を行った．試作した属性統合，属性値統合，個体統合システムを用いることで複数 DB 間の異種性を解決し，既存の DB を統合することが可能となった．また，試作した更新システムを用いることで，統合されたテーブルを用いた複数 DB の一括更新が可能となった．

今後の課題としては，統合処理を行う際に個人に関するシソーラスを利用し個体統合を行うことが挙げられる。また，複数の処理段階を踏んで更新を行う場合等のより複雑な更新処理への対応が挙げられる。

文献

[1] 大江諭，永井秀樹，岡林誠治，吉田香，打浪清一：「連邦型分散データベースシステムの構成法に関する研究」，電気関係学会第 55 回連合大会，911，465 ページ，2002

[2] 岡林誠治，吉田祥子，石戸大介，吉田香，打浪清一：「連邦型分散データベースシステムの構築法に関する研究」，電気関係学会九州支部第 57 回連合大会，10-1A-05，2004

[3] 吉田祥子，岡林誠治，吉田香，打浪清一：「連邦型分散データベースシステムの構成法に関する研究－属性統合、属性値統合、個体統合－」，電気関係学会九州支部第 58 回連合大会，11-2P-08，2005

社員id	姓	名	性別	生年月日	郵便番号	住所	電話番号	健康保険番号
9562958785	太田	啓介	男	1965-08-26	8180137	福岡県太宰府市青葉台5-2-3	0929230454	92-985678
9479687373	太田	陽子	女	1969-11-16	8180137	福岡県太宰府市青葉台5-2-3	0929230454	92-875806

図 16：会社 2DB(抜粋)

個人番号	保険番号	姓	名	住所	電話番号	生年月日	性別	加入届出日	期限
074658	92-985678	太田	啓介	福岡県太宰府市青葉台5丁目2番3号	092-923-0454	1965年08月26日	男	2003年10月02日	2005年10月01日
685784	92-875806	村田	陽子	福岡県飯塚市西町9番地45号	0948-22-0420	1969年11月16日	女	2003年08年30日	2005年08月29日

図 17：会社 2 の保険組合 DB(抜粋)

id	dbname	tablename	id_koumoku	memberid	sql	date	update_date
1	jdbc:postgresql://ocha.cse.kyutech.ac.jp/Hoken2	保険加入者表	保険番号	92-875806	update 保険加入者表 set 姓='太田',住所='福岡県太宰府市青葉台5丁目2番3号',電話番号='092-923-0454' where 保険番号='92-875806'	2005/12/03	
2	jdbc:postgresql://ocha.cse.kyutech.ac.jp/Shiyakusho	市民統合表backup	住民番号	67890227	update 市民統合表backup set 住所='福岡県太宰府市青葉台5丁目2番3号' where 住民番号='67890227'	2005/12/03	
3	jdbc:postgresql://ocha.cse.kyutech.ac.jp/Shiyakusho	市民統合表backup	住民番号	98680058	update 市民統合表backup set 姓名='太田 陽子',住所='福岡県太宰府市青葉台5丁目2番3号' where 住民番号='98680058'	2005/12/03	

図 18：更新データ抄録表