

構造化表出力を通じた直感的データベース更新の提案

岡林遼太郎[†] 遠山 元道^{††}

[†] 慶應義塾大学大学院 理工学研究科 開放環境科学専攻

^{††} 慶應義塾大学 理工学部 情報工学科

E-mail: [†]toka@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

あらまし データベースの更新を行う場合、DBMS を直接操作して更新する方法や HTML を利用し、Web 上から更新する方法など様々な方法が挙げられるが、構造化された表を利用してデータベースの内容を直接的に更新するような一般的なツールは存在せず、更新が頻繁に起こるような場合、その更新に要するユーザーの負担は非常に大きくなってしまふ。そこで本研究では構造化された表を容易に出力することができる SuperSQL の特徴を活かし、データベースのデータを構造化された表として HTML ファイルで出力すると同時に、それと同じ構造を持った更新するためのフォームも出力する。そして、ユーザーは Web 環境を通してそのフォームの値を書き換えるだけでデータベースの更新が可能となり、より直感的で、かつ容易な更新を実現する。また、構造化された表を通じた更新における問題点を取り上げ、それに対し、ユーザーの意図を把握するための具体的な解決方法について提案する。

キーワード DB 言語, ビュー更新, SuperSQL

A Proposal of Intuitive Update by Using Structured Table

Ryotaro OKABAYASHI[†] and Motomichi TOYAMA^{††}

[†] School of Science for OPEN and Environmental Systems,
Faculty of Science and Technology, Keio University

^{††} Department of Information and Computer Science, Faculty of Science and Technology, Keio University

E-mail: [†]toka@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

Abstract When updating a database, processes such as directly handling DBMS or using HTML to update from the Web can be taken. Though, a general tool using structured tables to update the database's content does not exist, and when if update must be done frequently, the cost would be enormous for the user. Our research uses SuperSQL, which can easily output structured tables, to output a database's data as a structured table in HTML format, and also outputs a form consisting of the same structure. The research implements a database-updating method in which a user can intuitively and easily update the database by just rewriting the form's values on the Web. Also, we go over the problems concerning the update of the database using structured tables, and propose a specific solution to understand the user's intentions.

Key words DB Language, View Update, SuperSQL

1. はじめに

現在、データベース内のデータをセルの結合やグルーピングなどを用い、構造化された表として Web 上などで表示することが大変多くなってきている。しかしながら、このような構造化された表に対して、データベースの更新が頻繁に起こるような場合、それを更新するユーザーの負担は非常に大きくなってしまふ。なぜなら、データベースを更新する際、その更新方法としては、SQL 等の検索言語によって直接更新を行う方法や、HTML などを使いフォームを作成し、それを使って更新を行う

方法など、複数の方法が存在するが、そのいずれも構造化された表を直接操作して更新を行うことは非常に困難であると言えらるからである。

そこで、本研究では SQL 等の検索言語を用いてのデータベースの更新やプログラミングに関する知識の習得やフォーム作成にかかる負担の大幅な削減を目的として、データベースに格納されているデータを容易に構造化された表として HTML ファイルで出力することができる SuperSQL の拡張を行う。そして、クエリの実行時に閲覧するための HTML と全く同じ構造の更新用フォームの出力を行い、そのフォームから送られてくる

データを受け取り、適切にデータベースの更新を行えるようなプログラムの開発を行う。これにより、ユーザーは、SuperSQLのクエリを1度実行しておけば、その後は出力されたフォームを通じて値を変更し、直感的にデータベースを更新することが可能になる。また、本研究では構造化された表を通じたデータベースの更新を行う際に生じる様々な問題点を解決し、ユーザーの意図を反映したデータベース更新の実現を試みる。

2. 関連研究

本研究で取り扱う、構造化された表を通じたデータベースの更新における問題点は、以前より研究されている関係データベースにおけるビューの更新時に生じる問題点と大変似ている。ビューの更新はその基底関係となるテーブルの更新が存在して、初めて可能になるが、この際、射影や結合など様々な演算を用いて出力されるビューの更新をどのように基底関係への更新へと反映させるかが非常に大きな問題となる。Bancilhonら[9]やPaoliniら[4]やDayalら[5]はこの更新問題の数学的形式化を試みており、Paoliniら[4]は外部ビューと概念ビューのマッピングについて考察し、その形式的な定義を論じ、Dayalら[5]はビュー更新が完全に可能となる状況はごくわずかしかな存在しないということを示している。また、Osaman[6]はビュー更新を実現する基底関係更新を求める具体的方法を研究している。さらに増永ら[7][8]はビューの更新問題を意味論的立場から考え、ビューの生成の過程を木構造に落とし込み、ビューの更新命令が各演算の過程で変形可能かどうかを調べていくことでこのような更新問題に対する統一的解決を提示している。その際、ビューの更新における、4つの意味論的多義性問題(SAP1~4)を挙げ、これらを含めたビューの更新問題を解決するためには操作レベルでは得られない意味論的情報や時にはシステムと利用者の相互作用が必要であるということを示している。

3. SuperSQLとは

SuperSQLはSQLを拡張したワンソースマルチユースを実現する言語である[2][3]。その質問文はSQLのSELECT句をGENERATE<media><TFE>の構文を持つGENERATE句で置き換えたものである。ここで<media>は出力媒体を示し、HTML、XML、Excel、 \LaTeX などの指定ができる。また<TFE>はターゲットリストの拡張であるTarget Form Expressionを表し、結合子、反復子などのレイアウト指定演算子を持つ一種の式である。

3.1 結合子

結合子はデータベースから得られたデータをどの方向(次元)に結合するかを指定する演算子であり、以下の3種類がある。括弧内は左がクエリー中の演算子を示し、右がレイアウト式を示す。レイアウト式については??節で説明する。

- 水平結合子(; : C1)

データを横に結合して出力。

例: Name, Tel

| | |
|------|-----|
| name | tel |
|------|-----|

- 垂直結合子(! : C2)

データを縦に結合して出力。

例: Name! Tel

| |
|------|
| name |
| tel |

- 深度結合子(% : C3)

データを3次元方法へ結合。出力がHTMLならばリンクとなる。

例: Name % Tel

| |
|------|
| name |
|------|

→

| |
|-----|
| tel |
|-----|

また時間軸方向結合として、結合子「#」も存在する[1]。

3.2 反復子

反復子は指定する方向に、データベースの値があるだけ繰り返して表示する。また反復子はただ構造を指定するだけでなく、そのネストの関係によって属性間の関連を指定できる。例えば

[科目名! [学籍番号, 評点]!]!

とした場合には、その科目において学生の評点一覧が表示されるが、

[科目名]!, [学籍番号]!, [評点]!

とした場合には前者のような関連はなく、単に各々の一覧が表示されるだけである。以下その種類について説明する。

- 水平反復子([] : G1)

データインスタンスがある限り、その属性のデータを横に繰り返す。

例: [Name],

| | | | |
|-------|-------|-----|--------|
| name1 | name2 | ... | name10 |
|-------|-------|-----|--------|

- 垂直反復子([!] : G2)

データインスタンスがある限り、その属性のデータを縦に繰り返す。

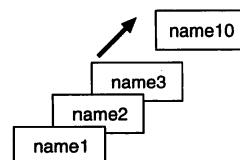
例: [Name]!

| |
|--------|
| name1 |
| name2 |
| ... |
| name10 |

- 深度反復子([]% : G3)

データインスタンスがある限り、その属性のデータを奥行き方向に繰り返す。

例: [Name]%



3.3 装飾子

SuperSQL では関係データベースより抽出された情報に、文字サイズ、文字スタイル、横幅、文字色、背景、高さ、位置などの情報を付加できる。これらは装飾演算子 (@) によって指定する。

<属性名>@[<装飾指定>]

装飾指定は”装飾子の名称 = その内容”として指定する。複数指定するときは各々を”,” で区切る。

3.4 関数

SuperSQL ではいくつかの関数が用意されている。ここでは代表的な関数を3つ紹介する。

3.4.1 imagefile 関数

imagefile 関数を用いると画像を表示することが可能となる。引数には属性名、画像ファイルの存在するディレクトリにパスを指定する。

```
imagefile(id, path="./pic")
```

3.4.2 sinvoke 関数 (出力メディアが HTML の場合のみ)

sinvoke 関数は FOREACH 句と同時に用いる。これらを用いることで深度結合子と同様にリンクを生成することができる。

```
sinvoke(cou.name, file="./menu.sql", att=co.country)
```

3.4.3 invoke 関数

invoke 関数はリンクを生成するための関数である。sinvoke 関数の場合、SuperSQL を手動で実行することでリンク先を生成しておくが、invoke 関数の場合、ユーザのリクエストに応じて動的にリンク先を生成する。

```
invoke(cou.name, file="./menu.sql",  
condition="ca.country="+co.country)
```

4. 既存のデータベース更新方法の問題点

データベースを更新する際、主に用いられるのは次の2つの方法である。

- (1) DBMS を直接操作することによって更新する方法
- (2) HTML のフォームを用いて、Web 上で更新する方法

ここではこれらの方法のメリット、デメリットについて述べる。

4.1 DBMS を操作して更新を行う方法

Access や Postgres といった DBMS を操作して更新を行う場合、ユーザーは更新を行うための SQL を作成するか、DBMS 側でサポートされている機能を利用して、表計算ソフトのように表形式のデータを直接操作して直感的にデータの更新を行うことが可能である。しかしながら、DBMS が提供するこの機能では、単一のテーブルごとの、しかもフラットな表の更新しか行うことができず、様々な演算を用い、さらに構造化された表のデータを直感的に操作してのデータベース更新は行えない。また、このような DBMS を直接操作する更新方法は、更新するユーザーがその DBMS を使える環境を用意しなければならない。そのため、多人数で管理、更新を行うような状況には不向きであると言え、更新が頻繁に起こる場合、管理者の負担は非常に大きいものになってしまうという欠点も存在する。

4.2 フォームを通して更新を行う方法

HTML などを使って更新用のフォームを作成し、データベースを更新する場合には、ユーザーは Web 上でフォームに必要な値を入力するだけで更新することが可能になる。このため、ユーザーはインターネットが使用できる環境さえあればデータベースの更新を行うことができ、DBMS を使って直接更新する方法に比べ、複数人でのデータベース更新に向いていると言える。また、DBMS の場合とは異なり、構造化された表のフォームを作ることもできるため、単一のテーブルごとの更新しか行うことができない DBMS の機能と比べて、柔軟性も高いと言える。しかしながら、更新を行うには、そのようなフォームをあらかじめ用意しておくなければならない。そのため、フォームの作成者には、負担がかかる (特に構造化された更新フォームを作成する場合) ことに加え、HTML や PHP などのプログラミング言語に関する知識も必要となり、誰でもすぐにフォームを作成することができるというわけではない。この点がフォームを通して、データベースを更新することのデメリットであると言える。

5. 提案手法

本研究では、このようなそれぞれの更新方法の問題点を克服し、あらかじめ更新を行いたいユーザーが HTML のフォームやデータベースを更新するためのプログラムを作成する必要なしに、Web 上で構造化された表の直感的な操作を通じたデータベースの更新を SuperSQL を利用することによって可能にする。

5.1 SuperSQL における更新フォームの出力

SuperSQL では次のようなクエリを実行することで表1のような構造化された表を HTML ファイルとして出力することができる。

```
GENERATE HTML [Lab, [Grade, [name]]]  
FROM Member
```

本研究ではこれを拡張し、update 句を付け加えることによって、図1のような値の部分フォームにした表を HTML ファイルとして出力する。

```
GENERATE HTML [Lab, [Grade, [name]]]!@ {update}  
FROM Member
```

まずユーザーは構造化された表を出力するような SuperSQL のクエリを作成し、実行する。その後、閲覧用の HTML ファイルと同時にデータの部分がフォームに変わった更新用の HTML ファイルも出力されるため、ユーザーはこのフォームの中の値を適切な値に変更する。そして、最後に更新プログラムがその値をデータベースに反映させることによって、構造化された表を通じた直感的なデータベースの更新が可能になる。

本研究では、データベース更新用の HTML ファイルの出力を可能にするよう SuperSQL の拡張を行う。さらに SuperSQL のクエリとフォームのデータを受け取り、SuperSQL のクエリをもとに適切なデータベース更新を行うような更新用プログラ

表 1 SuperSQL の実行結果

| | | 名前 |
|-----|----|--------|
| 遠山研 | M2 | 中道 亮 |
| | | 原田 哲志 |
| | M1 | 高橋 健太郎 |
| | | 佐藤 裕紀 |

| 研究室 | 学年 | 氏名 |
|-------|----|--------|
| 遠山研究室 | M2 | 中道 亮 |
| | | 原田 哲志 |
| | M1 | 高橋 健太郎 |
| | | 佐藤 裕紀 |

図 1 更新用フォームの出力例

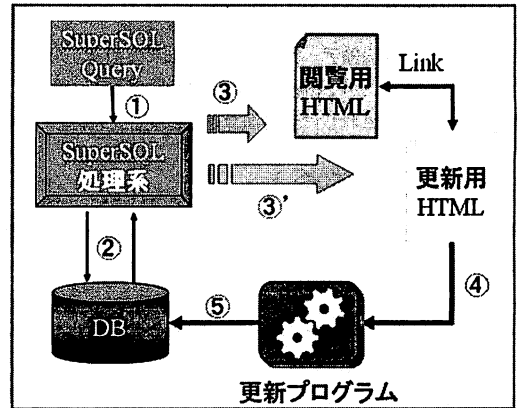


図 2 実際の処理の流れ

ムを開発する。

5.2 本研究における処理の流れ

具体的な処理の流れを図 2 として、またその説明を以下に示す。

- (1) SuperSQL のクエリが実行され、それが SuperSQL の処理系へと送られる
- (2) SuperSQL のクエリをもとにデータベースへ問い合わせを行い、データを取得する
- (3) SuperSQL のクエリのシンタックスに従って、データを構造化し HTML ファイルとして出力する
- (3) 通常の HTML ファイルと全く同じ構造の更新用フォームを同時に出力する。また、この際、実行された SuperSQL のクエリも加える。
- (4) フォームに記入されたデータなどの情報を更新用プログラムに送信する
- (5) SuperSQL のクエリからデータベースのスキーマ情報を取得し、それに従ってデータベースを更新する

まず、SuperSQL のクエリが実行されると、SuperSQL の処理系がそのクエリの要求を満たすような HTML ファイルを出力する。この際、本研究における拡張により、閲覧用の HTML ファイルと更新用の HTML ファイルを同時に出力し、更新用の HTML ファイルには<form> などの更新フォームを形成するためのタグを付加すると共に、後にデータベースのスキーマ情報を得るために実行された SuperSQL のクエリも加える。更新用プログラムでは、そのフォームの値とあらかじめ付加されていた、SuperSQL のクエリを受取り、フォーム内の値は配列に格納する。その後、SuperSQL のクエリからデータベースのスキーマ情報を取得し、それに従って、適切にデータベースの更新を行う。

6. 構造化表出力を通じたデータベース更新における問題点

2. 章で触れたように構造化された表を通じたデータベースの

更新には関係データベースにおけるビューの更新と同様の問題が発生する。また、それとは別に構造化された表に対する更新問題も新たに生じる。ここでは、中でも大きな問題となりうる、グルーピングと Join を使って出力された表の更新問題について論じる。

6.1 グルーピングされた表における更新問題

SuperSQL では 5. 章で掲載したようなクエリを書くことによって容易にグルーピングされた表を HTML ファイルとして出力することが可能である。しかしながら、そのような表を通して、データベースの更新を試みた場合、ユーザーの意図によっては更新がうまくいかないようなケースが存在する。例えば、表 2 の左側のようなグルーピングされた表に対する更新のパターンは大きく 2 つ考えられる。

- (1) 研究室名の変更を行う
- (2) メンバーの所属研究室を変更する

この時、最初のパターンである、研究室名を遠山研究室から遠山元道研究室に変更したい場合、そのまま値を書き換えることによって、つまり遠山元道研究室と書き換えることにより、変更を反映させることが可能である。しかしながら、2 つ目のパターンである、メンバーの所属を変更させたい、例えば原田という人が研究室を遠山研究室から大野研究室へ移し、その修正を行いたいような場合、この表のままではユーザーの意図する更新を行うことは難しい。なぜなら、遠山研究室の欄を書き換えてしまうと、他のメンバーの研究室の所属まで大野研究室に変えられてしまうという問題が生じるからである。すなわち、グルーピングで構造化された表を通してデータベースの更新を行うような場合、その更新における意味を正確に把握しなければ、ユーザーの本来の意図とは異なるタプルの更新が生じてしまう場合が存在する。

このようなグルーピングされた表の更新問題に対し、本研究では更新用のフォームを 2 通り出力することで、この問題の解消を試みる。まず、1 つ目の更新フォームは表 2 と同様のグルーピングされた表のフォームである。そして 2 つ目としてグルーピングされている部分をフラットな状態に戻した表 2 の右側の

表2 (左) グルーピングされた更新フォーム

(右) フラットな状態に戻した更新フォーム

| 更新フォーム | | 更新フォーム | |
|--------|----|--------|--------------|
| 遠山研究室 | M2 | 中道 | 遠山研究室 M2 中道 |
| | | 原田 | 遠山研究室 M2 原田 |
| | M1 | 高橋 | 遠山研究室 M1 高橋 |
| | | 佐々木 | 遠山研究室 M1 佐々木 |
| | B4 | 鈴木 | 遠山研究室 B4 鈴木 |
| | | 仲 | 遠山研究室 B4 仲 |

表4 (左) member テーブル (右) club テーブル

| member | name | club |
|--------|------|------|
| M2 | 中道 | 1 |
| M2 | 原田 | 2 |
| M1 | 高橋 | 4 |
| M1 | 佐々木 | 2 |
| B4 | 鈴木 | 4 |
| B4 | 仲 | 1 |

| club | name |
|------|------|
| 1 | 野球 |
| 2 | テニス |
| 3 | 合気道 |
| 4 | サッカー |

表3 多段階のグルーピングを用いた表

| 学部 | 学科 | 研究室 | メンバー | | |
|----------|-------|-----------------|-----------------|------|---------------|
| 理工学部 | 情報工学科 | 遠山研究室 | 中道 亮 原田 哲志 | | |
| | | 大野研究室 | 高橋 健太郎 佐藤 裕紀 | | |
| | | 山崎研究室 | 岡林 遼太郎 小林 武彦 | | |
| | 機械工学科 | 田中研究室 | 浅見 裕光 佐々木 健人 | | |
| | | 環境情報学部 | 環境学科 | 小林ゼミ | 田村 元 西岡 孝明 |
| | | | | 脇谷ゼミ | 内山 英昭 荒倉 敏 |
| メディア情報学科 | 大林ゼミ | | 神谷 基文 下永田 真人 | | |
| | 片山ゼミ | 谷口 大樹 渋谷 慧一郎 | | | |

表5 結合後の表

| member | name | club |
|--------|------|------|
| M2 | 中道 | 野球 |
| M2 | 原田 | テニス |
| M1 | 高橋 | サッカー |
| M1 | 佐々木 | テニス |
| B4 | 鈴木 | サッカー |
| B4 | 仲 | 野球 |

ようなフォームを出力する。これにより、研究室名の変更を行う1つ目のパターンに対してはグルーピングされた表を使っている更新を行うことが可能になり、個々メンバーの所属研究室もしくは学年を修正するような場合には2つ目のフラットな表を使い、それぞれのタブごとに更新を行うことによって問題なく更新することができるように考えると考えられる。

しかしながら、この解決方法の問題点として属性数が増え、グルーピングが何段階にもなったような場合に、グルーピングしている属性に対する値の更新に手間がかかってしまうということが挙げられる。例えば、表3のようなデータが存在し、メディア情報学科が理工学部の学科に移ったとする。このような変更に対し、本研究の手法を使ってデータベースを更新しようとした場合、全てグルーピングを解除したフラットな表で更新を行わなければならない。そのため、ユーザーはメディア情報学科に属する全ての生徒の学部を1タブずつ書き換えていかなければならず、非常に手間のかかる作業と言える。このように、グルーピングを行う属性数が多くなった場合の更新方法については今後、課題として検討していかなければならない。

6.2 Join を用いた表における更新問題

グルーピングを用いた表における更新問題のようなユーザーの意図によって複数の更新方法が考えられるケースは表の結合などを行った場合にも生じる。表4に示すような member テーブルと club テーブルの2つのテーブルをクラブ属性をキーとして Join を行い、その結果、表5が出力されたとする。この

時にも、グルーピングの場合と同様、2つの更新パターンが考えられる。

- (1) メンバーの所属クラブの変更を行う
- (2) クラブ名の変更を行う

このような場合、1つ目のパターンの変更を行うには、実際には基底関係である member テーブルの club 属性の外部キーを書き換えなければならない。例えば、原田という人物のクラブがテニスから合気道へ変わった場合、実際には member テーブルの原田の club 属性のキーの値を2から3へ書き換える必要がある。また、2つ目のパターンである、クラブ名自体の変更を行う場合、例えば野球部を軟式野球部に変えたいような場合には、club テーブルの name 属性の値を書き換えなければならない。このようにフォームの値を書き換えるという操作の面ではどちらも同じであるが、ユーザーの意図によって更新するテーブル、属性が変わってきてしまうという問題が生じ、ユーザーがどちらの更新パターン望んでいるかはシステム側で把握することは極めて難しいと言える。

そこで本研究では、このような問題に対し、join されている属性にリンクを張り、そのリンクがクリックされると図3のようなサブウィンドウを表示させることで解決を試みる。このサブウィンドウによりどちらのパターンの変更を行いたいかをユーザーにラジオボタンで選ばせ、システム側でユーザーの意図を把握できるようにすることによって、適切な更新を行う。ユーザーがキーの書き換え、つまり何らかの所属の変更を行いたい場合には、上のセレクトボックスから候補となる値を選択し、システム側ではその外部キーを値として持つことで、キーの値を変更することができる。例の場合、ユーザーに原田のクラブの変更候補の中から合気道を選択させ、セレクトボックスの値に合気道のキーである3を持たせておくことで、member テーブルのクラブ属性の値の書き換えが可能になると考えられる。また、値そのものを書き換える時には、下のテキストエリアに値を入力することで変更を行える。例では、野球を軟式野球と書き換え、システムが club テーブルを更新する

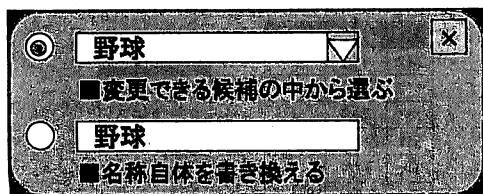


図3 サブウィンドウの出力例

ことで、ユーザーの意図を満たすようなデータベースの更新を行うことができる。このように、両方の選択肢をユーザーに表示することにより join を含んでいる表を通した更新も行えるようになると思われる。

7. 結論および今後の課題

現在、構造化された表を利用してデータベースの内容を直接更新するような一般的なツールは存在せず、更新が頻繁に起こるような場合のユーザーの負担が非常に大きくなってしまふ。そこで本稿では、そのような点に着目し、SuperSQL を利用することで、ユーザーがあらかじめ HTML のフォームを用意しなくても、Web 上で直感的にデータベース更新を行える方法を提案した。また、構造化された表を通した更新における問題点を取り上げ、それに対し、ユーザーの意図を把握するための具体的な解決方法についても述べた。

今後の課題としては、Join やグルーピングなどを含む複雑に構造化された表における更新アルゴリズムの詳細の検討や問題点および解決方法の形式化などを行い、それに伴って、順次実装を進めていく。また、構造化された表に対して、データの更新だけでなく、挿入や削除をどのように行っていくかといった問題やセキュリティ面など、実用性を考慮した際、不十分と見られる点について考察し、その解決を試みる。さらに実行時間や関連研究との比較といった評価方法についても考えていく。

文 献

- [1] 岡部 康矢：『SuperSQL における HTML 出力の高品質化』、学士論文、慶應義塾大学理工学部情報工学科、2002
- [2] Motomichi Toyama, "SuperSQL: An Extended SQL for Database Publishing and Presentation," *Proceedings of ACM SIGMOD '98 International Conference on Management of Data*, pp. 584-586, 1998
- [3] SuperSQL: <http://ssql.db.ics.keio.ac.jp/>
- [4] P. Paolini and G. Pelagatti. Formal Definition of Mappings in a Database. In *Proceedings of ACM SIGMOD*, pp.40-46, 1977.
- [5] U. Dayal and P. A. Bernstein. On the Updatability of Relational Views. In *Proceedings of VLDB*, pp.368-377, 1978.
- [6] I. M. Osman. Updating Defined Relations. In *Proceedings of AFIPS NCC*, pp.733-740, 1979.
- [7] Y. Masunaga A Relational Database View Update Translation Mechanism. In *Proceedings of VLDB*, pp.309-320, 1984.
- [8] 増永 良文、野口 正一：『関係データベースビュー更新問題の意味論的解決法』、情報処理学会論文誌、Vol.25, No.1、P66-73 1984.
- [9] F. Bancilhon and N. Spyratos. Update Semantics of Relational Views. In *ACM Trans. Database Syst.*, pp.557-575, 1981.