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

2006－DBS－140（Ⅰ）（12）
2006／7／12

SuperSQLによる ビジネスオートメーション支援

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: † oka@db.ics.keio.ac.jp, ‡ toyama@ics.keio.ac.jp

データベースに格納されているデータは、それ自体に価値があるだけでなく、さらにそれらを分析することによって、様々な情報を引出すことができる。しかしながら、現状ではデータベースの利用とデータの分析は関連性が薄い、もしくはデータベースと連携していたとしても、ユーザーにとっては使いにくい場合が多く見受けられる。したがって、本論文ではSuperSQLを用いてデータベースと表計算ソフトを統合することにより、フォーマットの統一管理などを可能とし、ビジネスオートメーションの効率化を試みた。具体的には、質問言語の拡張により、XMLを介して自動的にExcel上にデータベースのデータを出力し、さらにデータのグルーピングを行うことで、グループごとのデータ出力が可能となった。

キーワード：DB言語，　OLAP，　SuperSQL

Support for Business Automation using SuperSQL

Ryotaro OKABAYASHI †　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 : †oka@db.ics.keio.ac.jp ‡toyama@ics.keio.ac.jp

The data stored in the database is worthy not only themselves but also the analysis of them. We can get out various infomation from those data. However, use of the database have little relevance to analysis of them or it is awkward for user even if work with database at present. In this paper, therefore, we propose integration of database and data processing and analysis system by using of SuperSQL, in doing so, enabled to unify management of format and attempted efficiency of Business Automation. In particular, we could automatically output data stored in database on Excel through XML and output each data to group them.

keyword : DB Language , OLAP , SuperSQL

1 はじめに

今日、顧客データや商品データなど様々なデータが注目されているが、それらのデータはそのままWEBページなどに表示し閲覧するといった使い方以外に、統計分析やグラフ化などを行うことで、隠れた情報を引き出すというデータマイニングにも利用可能である。しかしながら、現状ではデータベースの利用とデータの分析が必ずしも一致していない場合が見受けられる。具体的には、個々が様々な場所からデータを集め、自分のデータベースとして管理するといったことが挙げられる。この場合、非常に手間がかかる、独自の観点でデータの作成を行うことから、その形式がバラバラになってしまうといったデメリットが存在する。

このため、表計算ソフトのExcelにはDBQuery [6] やOLE2 [7] などのデータベース連携機能が用意されており、Accessで作成したデータベースをExcelに出力することは可能である。ただし、その出力結果はただのフラットな表であり、データの量が膨大すぎる場合などには、シート別に各グループのデータを分割したほうが扱いやすいといった問題点が存在する。

そこで、本論文ではSuperSQL [4] を拡張し、データベースとデータ処理系を統合を行った。具体的には、ExcelがXML対応となったことに着目し、クエリを実行したときに自動的にデータベースのデータが入ったExcel専用のXMLファイルを出力できるようSuperSQLを拡張した。さらにデータのグルーピングを行うことによって、1度クエリを実行するだけで、それぞれの条件に対応する複数のファイルの出力を可能にした。また、SuperSQLの特徴であるセル同士の結合やリンク生成などを含んだクエリにも対応できるようにした。このようにデータベースと表計算ソフトを融合することにより、それぞれのフォーマットの統一や一度に複数のファイル出力を可能し、ユーザーが行うデータ整理などの作業の負担を軽減することで、ビジネスオートメーションの効率化支援を試みた。

本稿の構成は以下の通りである。まず、2章ではSuperSQLについて述べる。次に3章ではExcelに対応するXMLについて、4章では実装の概要やアルゴリズムについて説明する。そして5章で実行結果を紹介し、6章で評価および今後の課題の検討を行った後、最後に7章でまとめを述べる。

2 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]。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。

2.1 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2.3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]。

2.2 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かえして表示する。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。

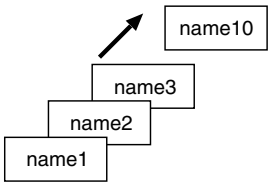
- 水平反復子 ([ ], : G1)
  データインスタンスがある限り、その属性のデータを横に繰り返す。
  例 : [Name],

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]%



2.3 レイアウト式

記述された SuperSQL 質問文は内部処理系における構文解析部によって *Plain query* と *Layout Expression*(レイアウト式) に変換される。例えば SuperSQL クエリーが

GENERATE ECXML[s.city ! [d.name],]!

であるとすると、そのレイアウト式は

```
<?xml version="1.0" encoding="EUC-JP"?>
<?mso-application progid="Excel.Sheet"?>
<Workbook xmlns="urn:schemas-microsoft-com:office:spreadsheet"
 xmlns:x="urn:schemas-microsoft-com:office:excel"
 xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet">
 <Styles>
  <Style ss:ID="s21">
   <Font ss:FontName="MS Pゴシック" ss:Size="11"/>
  </Style>
</Styles>
<Worksheet ss:Name="Sheet1">
 <Table>
  <Row>
   <Cell ss:StyleID="s21"><Data ss:Type="String">…</Data></Cell>
  </Row>
 </Table>
</Worksheet>
</Workbook>
```

図 1: XML の全体的な構成

(G2 (C2 1 (G1 2)))

と表される。ここで数字の 1、2 はクエリー中の属性に対応するグループホルダーである。

3 Excel 対応の XML

この章では Excel 対応の XML について、その特徴やレイアウトなど、どのような指定ができるかについて述べる。

3.1 全体構成

Excel における XML サポート [5] によると、Excel へ出力するための XML はまず、ヘッダー部とシート内部の大きく 2 つに分かれる。

Excel に対応した XML ファイルの全体的な構成は図 1 のようになる。ヘッダー部では XML のヘッダーと Excel に対応しているというヘッダーの両方を付加する必要がある。

次にフォントや文字の大きさなどのスタイルの設定を行う。これは 6 行目にある<Styles></Styles>タグの中で行われる。このスタイルは<Style></Stysel>タグの ID を変えていくことで複数設定することが可能であり、さらに図 1 に掲載されていること以外にも、文字位置を中央に持ってくる、罫線を付けるといったことなどもできる。ここで設定したスタイルは後で出てくる<Cell>タグに埋め込むことでその効力を発揮する。

<Worksheet ss:Name=”Sheet1”> <Table>タグではワークシートやテーブルの定義を行う。ワークシートの名前を自由に変更することやテーブルのオ

```
<Row>
 <Cell ss:StyleID="s21"><Data ss:Type="String">名前</Data></Cell>
 <Cell ss:StyleID="s21"><Data ss:Type="String">学籍番号</Data></Cell>
</Row>
<Row>
 <Cell ss:StyleID="s21"><Data ss:Type="String">岡林遼太郎</Data></Cell>
 <Cell ss:StyleID="s21"><Data ss:Type="Number">602001</Data></Cell>
</Row>
<Row>
 <Cell ss:StyleID="s21"><Data ss:Type="String">中道亮</Data></Cell>
 <Cell ss:StyleID="s21"><Data ss:Type="Number">602002</Data></Cell>
</Row>
```

図 2: XML の例

	A	B
1	名前	学籍番号
2	岡林遼太郎	602001
3	中道亮	602002

図 3: Excel で開いた結果

プションとして テーブルの大きさを決めることなどが可能である。

3.2 出力に関するタグ

ワークシート内で定義されるタグは<Row><Cell><Data>の３種類である。まず、<Row>はワークシート内の１行を表している。Excelのデータを XML で表記するときの特徴として、それぞれのデータは１行単位で書かれるということが挙げられる。<Row><Row>で囲われた部分がその行におけるデータの内容となる。

次に<Cell>タグは文字通りセルを表す。図１のように最初に設定したスタイルの ID を付加することで、その通りの文字の大きさやフォントになる。

最後にデータに関するタグ<Data>について説明する。この<Data></Data>の中にデータを書くことで、実際、Excel で開いた時にそのデータが出力されることになる。また、<Data ss:Type=’型の名前’>とすることで、そのデータの型を設定することも出来る。

これらのタグに関する XML のサンプルとそれを Excel で開いたときの結果をそれぞれ図 2、図 3 として以下に示す。

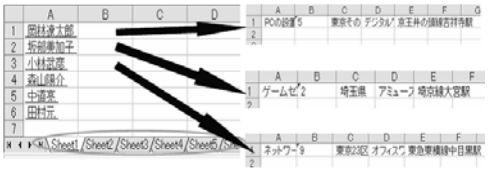


図 4: Excel におけるページ遷移

4 実装の概要およびアルゴリズム

３章で紹介したタグの構成に基づき、SuperSQL の HTML 出力部を拡張することで、Excel に対応した XML ファイルを出力できるよう実装した。実装の大まかな内容は以下の４つとなる。

1. 通常の表の出力
2. リンクを伴う表の出力
3. セルの結合を伴う表の出力
4. グルーピングによる複数ファイル出力

4.1 通常の表の出力

通常の表出力は前章で説明したタグの構成に従い、データベースから抽出した値をそれぞれ<Data>で囲うことで実現した。また、表を見易くするために最初のスタイルの設定において、セルの四方に罫線を加えた。

4.2 リンクを伴う表の出力

SuperSQL の特徴の一つとしてあるデータからそれに関連するデータへのリンクを張れるということが挙げられる。通常の HTML におけるリンクでは、クリックされると別のページに飛ぶことになるが、Excel では図 4 に示すとおり、そのページ遷移を Sheet で行うことで HTML と同様のリンク機能を実現した。

4.3 セルの結合を伴う表の出力

SuperSQL のもう１つの特徴として結合したセルの出力が挙げられる。HTML におけるセルの結合では table によるネストや colspan, rowspan を用いて表現することが可能であったが、Excel では前

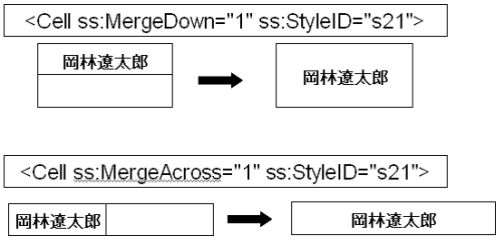


図 5: MargeDown, MargeAcross の例

述したように１行単位でデータが構成されるため、異ったアルゴリズムを用いる必要がある。そこで、まずはセルの結合を Excel で表現するための XML 表記から順に説明する。

4.3.1 MargeDown MargeAcross

Excel におけるセルの結合を行うには MargeDown や MargeAcross という句を加えなければならない。ここで MargeDown は縦方向の結合、MargeAcross は横方向の結合を行う。図 5 に示すように、<Cell> のオプションとして MargeDown や MargeAcross と結合する範囲を加えることで Excel で開いたときに結合したセルが出力される。結合する際に、結合されるセルは空白でなければならず、値が入っているときに結合しようとするとエラーになってしまう。

4.3.2 アルゴリズム

セルの結合を伴うような表を出力するためには、MargeDown や MargeAcross を用いる必要があることは上で述べた。しかし、それを利用するためにはどのセルにデータが入って、どこは空欄のままにしておくかといった情報が必要である。岡部 [1] もこのような問題を取り上げているが、Excel の場合には１行単位で表が構成されるので、データの順番等を途中で並べ替えなければならない。これまでの SuperSQL ではレイアウト式が与えられると、それを外から内、左から右の順番で処理していた。すなわち、

[C2 [C1 [C2 [0][1]] [2][3]] [4]]

といったレイアウト式が与えられていた場合、まず [0][1] を縦につないで、その表に対して、次に [2][3] を横につなぎ、最後に [4] を縦つなぐことで図 6 のような表を出力することができた。ここで C1 は横結合、C2 は縦結合を表している。しかしながら、Excel においてはすでに [0][1] を縦につなぐという時点で不可能である。そのため、まず、レイアウト式を [C2 [0][1]]、[C1 [2][3]]、[C2 [4]] のように結合ごとに細分化し、それらを順に２次元配列に格納していくことで、図 7 のような２次元の仮想的な表が得られる。また、縦結合、横結合が何回行われたかを記録する、お互いに初期値が 1 であるような配列をそれぞれ用意することで、その情報をもとに、ある時点でのデータをどれくらい MargeDown および MargeAcross させるのかを判断することができる。

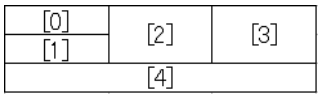


図 6: 構造の例

[0]	[2]	[3]
[1]		
[4]		

図 7: ２次元配列による仮想的な表

今の例だと、まず、縦結合のレイアウト式 [C2 [0][1]] が存在し、最初の [0] に対して、[1] が縦結合されていることから、縦結合の数を表す配列には 2 という値が入っている。したがって、次の横結合のレイアウト式 [C1 [2][3]] において、[2][3] それぞれのデータは１段階、MargeDown した形で出力されなければならない。また、[C2 [4]] についても同様のことが言え、１つ前に [0] のデータに対して、[2][3] が横結合していることから、[4] のデータは２段階、MargeAcross する必要がある。

このように、縦結合、横結合の情報を元にどのデータがどれくらい MargeDown および MargeAcross させるかを新たな配列に格納することによって、実際に XML のコードを生成し出力した際、うまくセルを結合した状態で出すことができる。

4.4 グルーピングによる複数ファイル出力

実社会で Excel を利用する場合、データの分析に使うのは勿論のことだが、その前段階として、名簿

や担当業務表などのような一覧表として使われることが多い。例えば、それぞれの授業における履修者名簿などは授業によってその中身が異ってくるが、元となるものは生徒情報のデータベースであり、すべてのExcelファイルはそのデータベースから作られることとなる。このような場合、それぞれの担当者ごとに１つ１つファイルを作成していくことは、非常に手間がかかり、写し間違いなどのミスも生じてくる。また、これに似たケースとして登録制アルバイトや営業の訪問先なども挙げられる。

したがって、本研究ではグルーピングを行うことで、このようなそれぞれに対応するデータのファイルを１度にまとめて出力することを可能にした。これは一番最初の例で言うと、１つクエリを実行するだけで、それぞれの担当者が受け持つ生徒のデータを、担当者ごとのファイルに分けた状態で出力することを意味している。こうすることによって、名簿の配布者の作業量を大幅に削減することができ、ビジネスオートメーションを支援することができると考えられる。

4.4.1 実現方法

それぞれの条件に合うファイルを複数出力するにはグルーピングを行い、そのグループごとに出力するファイルを分けることが必要となる。ここでは、わかりやすく説明するため登録制アルバイトのデータベースを例に挙げて説明する。

例えば、働く人ごとの情報を持つファイルに分けたい場合、次のようなクエリを実行すればよい。

```
GENERATE ECXML [j.mem_name %
[j.job_name, j.genre, j.station]!]!@{grouping=file}
FROM job j;
```

このようなクエリを実行するとまず、SuperSQLのデータコンストラクタ部により

```
Select j.mem_name, j.job_name, j.genre, j.station
From job j;
```

という通常のSQLに変換され、その結果として通常の表形式のデータ群が返ってくる。ここで、このデータに対し、SSQLのクエリからmem_nameという属性でグルーピングを行い、その結果ごとに表１のようなイメージでデータを切り離す。

そして、それぞれの名前に対応するデータ(岡林の場合、1.HP設計、デジタル、日吉 2.ソフト開発、デジタル、渋谷　3.パン屋、販売、横浜)を１つのExcelファイルごとにばらばらに書き出すことで、グルーピングによる複数ファイル出力が可能となる。さらに、グルーピングした属性の値をファイル名にすることで、図8のようにすぐにファイルの内容が認識できるようになる。また、データベースが大きくなると、一度に出力されるファイル数も比例して多くなることが考えれる。そのため、出力したファイルは最初からプログラム実行時に指定した名前を持つフォルダ(指定しない場合はoutというフォルダ)に格納するようにした。

説明では、働く人ごとにグルーピングされたファイルの出力を例に挙げたが、他にもジャンルごとのファイルであったり、最寄駅ごとのファイルを出力することも可能である。

表 1: データの分割イメージ

岡林	HP設計 ソフト開発 パン屋	デジタル デジタル 販売	日吉 渋谷 横浜
中道	焼肉屋 牛丼屋	飲食 飲食	日吉 川崎
小林	データ入力 家庭教師 歯科助手	オフィス 教育 医療	大倉山 綱島 日吉

図 8: 出力されたファイル

5 実行結果

本研究における通常の表とセルの結合を含む表の出力結果を図9, 図10として次ページに示す。ここで用いたデータベースは4.4.1章で用いたような登録制アルバイトデータベースであり、各リレーションとその属性は次の通りである。

表 2: member テーブル

属性名	型	内容
name	character	会員名
log_id	character	ログイン名
password	character	ログインパスワード
phone	character	電話番号
birth	date	誕生日

表 3: job テーブル

属性名	型	内容
id	integer	アルバイト ID
name	character	アルバイト名
number	interger	残り登録可能数
area	character	勤務エリア
genre	character	仕事ジャンル
jikann	character	勤務時間帯
station	character	最寄駅

1. 通常の表出力のクエリ

```
GENERATE ECXML [n.name, j.name, j.number,
 j.genre, j.station ]!
FROM job j, member n, match m
WHERE m.job_id = j.id AND m.mem_id=n.log_id;
```

2. セルの結合を伴う表出力のクエリ

```
GENERATE ECXML [ {{n.name! j.name,
 j.number}! j.genre}.j.station ]!
FROM job j, member n, match m
WHERE m.job_id = j.id AND m.mem_id=n.log_id;
```

6 今後の課題

6.1 装飾の強化

SuperSQL によるＨＴＭＬの出力ではテーブルの大きさや背景の設定、セルや文字の大きさや色など様々な装飾を指定することができる。これらの指定の内、テーブルのサイズに関する指定や文字およびセルに関する指定などは XML による Excel ファイルでも出力可能であるが、現在の実装状況ではクエリにこれらの指定を組み込むことはできない。したがって、テーブル全体の大きさの指定、１つのセルごとの大きさの指定、セルごとの文字サイズの指定、セルごとの背景色、文字色の指定など、装飾に関する細かい設定ができるように、本研究によるシステムを拡張する必要があると思われる。

表 4: match テーブル

属性名	型	内容
job_id	integer	アルバイトの ID
mem_id	character	会員の ID

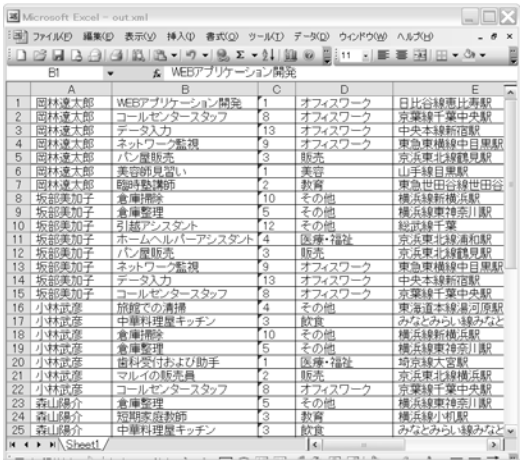
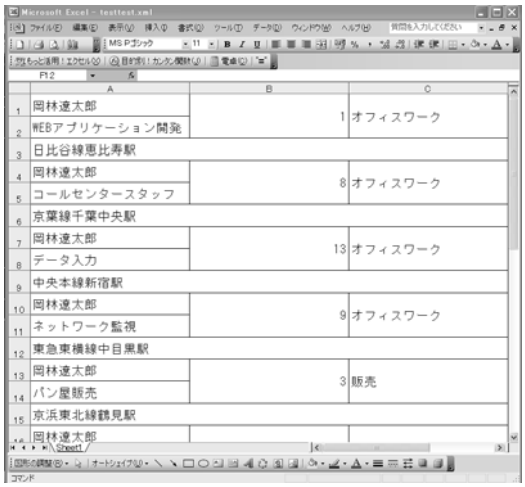


図 9: 通常の表の出力結果

6.2 グルーピングによる出力の応用

本研究では、グルーピングを行った際、グルーピングされた値に対応するデータをそれぞれファイルに書き出すことしかできなかった。しかし、ユーザーによってはファイルごとではなくシートごとに出力したいといった要望を持つことも十分に考えられる。また、グルーピングの段数が３段以上になると、フォルダ⇒ファイルやファイル⇒シートといった様々なパターンが考えられる。そのため、これらのいくつものパターンに柔軟に対応できるよう、さらに拡張する必要がある。



	A	B	C
1	岡林遼太郎	1	オフィスワーク
2	WEBアプリケーション開発		
3	日比谷線恵比寿駅		
4	岡林遼太郎	8	オフィスワーク
5	コールセンタースタッフ		
6	京葉線千葉中央駅		
7	岡林遼太郎	13	オフィスワーク
8	データ入力		
9	中央本線新宿駅		
10	岡林遼太郎	9	オフィスワーク
11	ネットワーク監視		
12	東急東横線中目黒駅		
13	岡林遼太郎	3	販売
14	パン屋販売		
15	京浜東北線鶴見駅		
16	岡林遼太郎		

図 10: セルの結合を含む表の出力結果

7 まとめ

今日、大量のデータを効率よく保存し、かつ、有効利用するために様々なところでデータベースが利用されている。しかしながら、いざ出力したデータを分析しようとしたときに、分析しやすいようデータを加工しなければならないなどの問題点があった。

本研究ではこの点に着目し、この２つの処理を統合することによって、ユーザーの作業量を減らし、ビジネスオートメーションを効率化することを試みた。その媒体として SuperSQL を用いることで、XML を介して Excel 上にデータベースのデータを出力することが可能となった。さらに通常の表だけでなく、リンクを伴ったものやセルの結合を含んだ表、加えてグルーピングを行うことによって、一度で複数のファイルを出力させることにも成功した。これにより、今まで手作業で行うことの多かった一連のデータ加工などの時間が短縮され、すぐにデータの分析に取り掛かることができるようになった。このことはユーザーにとって非常に有益であると言える。

今後はレイアウト力のさらなる強化やグルーピングによる出力の汎用性向上といった拡張を加えることで、さらにクオリティの高い環境を作り上げ、より多くのユーザーに使ってもらえるようなシステムを目指す。

参考文献

[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,” *Preceedings of ACM SIGMOD ’98 International Conference on Management of Data*, pp. 584-586, 199

[3] 遠山 元道 : 『ターゲットリストの拡張によるデータベース出版と概視の実現』、信学技報、Vol.93,No.152、P79-88、電子情報通信学会、1993

[4] SuperSQL: http://ssql.db.ics.keio.ac.jp/

[5] Excel による XML サポート:
http://www.microsoft.com/japan/office/previous/xp/techinfo/whitepaper/ XML_Excel2002.asp

[6] @ IT：Windows&TIPS:
http://www.atmarkit.co.jp/fwin2k/win2ktips/296exceldb/exceldb.html

[7] OLE2 on. NET Framework:
http://www5.plala.or.jp/atata/net/