

視覚的インターフェースによる SuperSQL クエリ生成

吉沢 拓[†] 遠山 元道[†]

[†] ^{††} 慶應義塾大学理工学部情報工学科 〒 223-8522 神奈川県横浜市港北区日吉 3-14-1

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

あらまし SuperSQL では、表による出力結果とその構造を示すクエリとの間に感覚的な距離が存在する。そのためユーザが持っているイメージをクエリに書き表すには習熟が必要であり、不慣れなユーザは思い通りの出力が得られない事もあった。そこで本論文ではこの問題を解決するために、ユーザのイメージと出力結果を直結させるべく、ダイレクトマニピュレーションによる視覚的なインタフェースに基づいたクエリ作成支援システムを提案する。

キーワード SuperSQL, HTML, ダイレクトマニピュレーション, Wysiwyg

A Visual Interface for SuperSQL Query Formulation

Taku YOSIHZAWA[†] and Motomichi TOYAMA[†]

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

Hiyoshi3-14-1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223-8522 Japan

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

Abstract In SuperSQL, There is sensitive distance between the output by table and the query indicates structure. Therefore experience is required for expressing users' own image to query, inexperienced user sometimes can get unexpected result. To solve the problem, this thesis proposes the system for supporting query making with visual interface by direct manipulation, intended to connect own image and the output.

Key words SuperSQL, HTML, Direct Manipulation, Wysiwyg

1. はじめに

SuperSQL とは、関係データベースへの問い合わせと同時にその検索結果の構造化を行い、出力メディアや多様なレイアウト表現の指定を可能とした出力を行う処理系である。しかし出力結果が表構造という視覚的なものであるのに対し、その構造を表す SuperSQL クエリは論理的なものである。そのため両者には感覚的な距離が存在し、ユーザが持つイメージをクエリに反映するには習熟が必要であった。

本稿では上記の問題を解決するために、ダイレクトマニピュレーションによって出力と入力を近づけるためのインターフェースを提案する。グリッド画面上にユーザが自由に表を描画するが、その構造の解析はシステム側が行なうためユーザは論理的な思考を伴わずに入力する事が出来る。

以下、本稿の構成を示す。まず 2 章で SuperSQL の概要について述べる。次に 3 章でインターフェースの概要について述べ、さらに 4 章でユーザが入力した表構造が有効なものであるかを判別する条件やその判別法について述べ、そして 5 章では様々情報がどのように保存・処理されているかを述べる。続く 6 章では実験手法などについて述べ、7 章で結論およびまとめ

を述べる。

2. SuperSQL

この章では本論文で改善を試みる SuperSQL について簡単に述べる。SuperSQL は関係データベースの出力結果を構造化し、多様なレイアウト表現を可能とする SQL の拡張言語であり、慶應義塾大学遠山研究室で開発されている [1] [2]。そのクエリは 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, PDF 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。括弧内はクエリ中の演算子を示している。

- 水平結合子 (,)

データを横に結合して出力。

例：Name, Tel

name	tel
------	-----

- 垂直結合子 (!)

データを縦に結合して出力.

例：Name! Tel

name
tel

- 深度結合子 (%)

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

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]!

とした場合には各属性間に関連はなく, 単に各々の一覧が表示されるだけである. 一方, ネストを利用して

[科目名 ! [学籍番号 , 評点] !]!

とした場合には, その科目毎に学籍番号と評点の一覧が表示されるといったように, 属性間の関連が指定される. 以下, その種類について述べる.

- 水平反復子 ([,])

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して表示する.

例：[Name],

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

- 垂直反復子 ([!])

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して表示する.

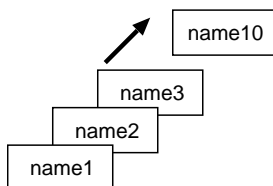
例：[Name]!

name1
name2
...
name10

- 深度反復子 ([] %)

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 (HTML ではリンク, PDF ではページ変換) に繰り返して表示する.

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 *Plainquery* と *LayoutExpression* (レイアウト式) に変換される. たとえば SuperSQL クエリーが

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

であるとする, そのレイアウト式は

(*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ブレースホルダーである.

2.4 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.

2.5 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 4 つ紹介する.

2.5.1 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”)

2.5.2 link 関数 (出力メディアが HTML の場合のみ)

link 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.

link(cou.name, file=”./menu.sql”, att=co.country)

2.5.3 invoke 関数

invoke 関数はリンクを生成するための関数である. link 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)

2.5.4 embed 関数

embed 関数を用いることでクエリを分割・合成することが可能になる. 利用方法は別ファイルに保存されたクエリ, もしくは HTML ファイルを埋め込みたい箇所に embed 関数を記述する.
embed(file=”./test.sql” where=”ca.id=” att=ca.id)

3. 提案システム

図 1 はインタフェースの概観である. ユーザは左部のグリッド画面上をドラッグする事により, 表構造を描画する. また描画した表に文字を入力, あるいは必要な情報をドロップする事で各セルに表示する情報を指定したり, 装飾子を指定する事で

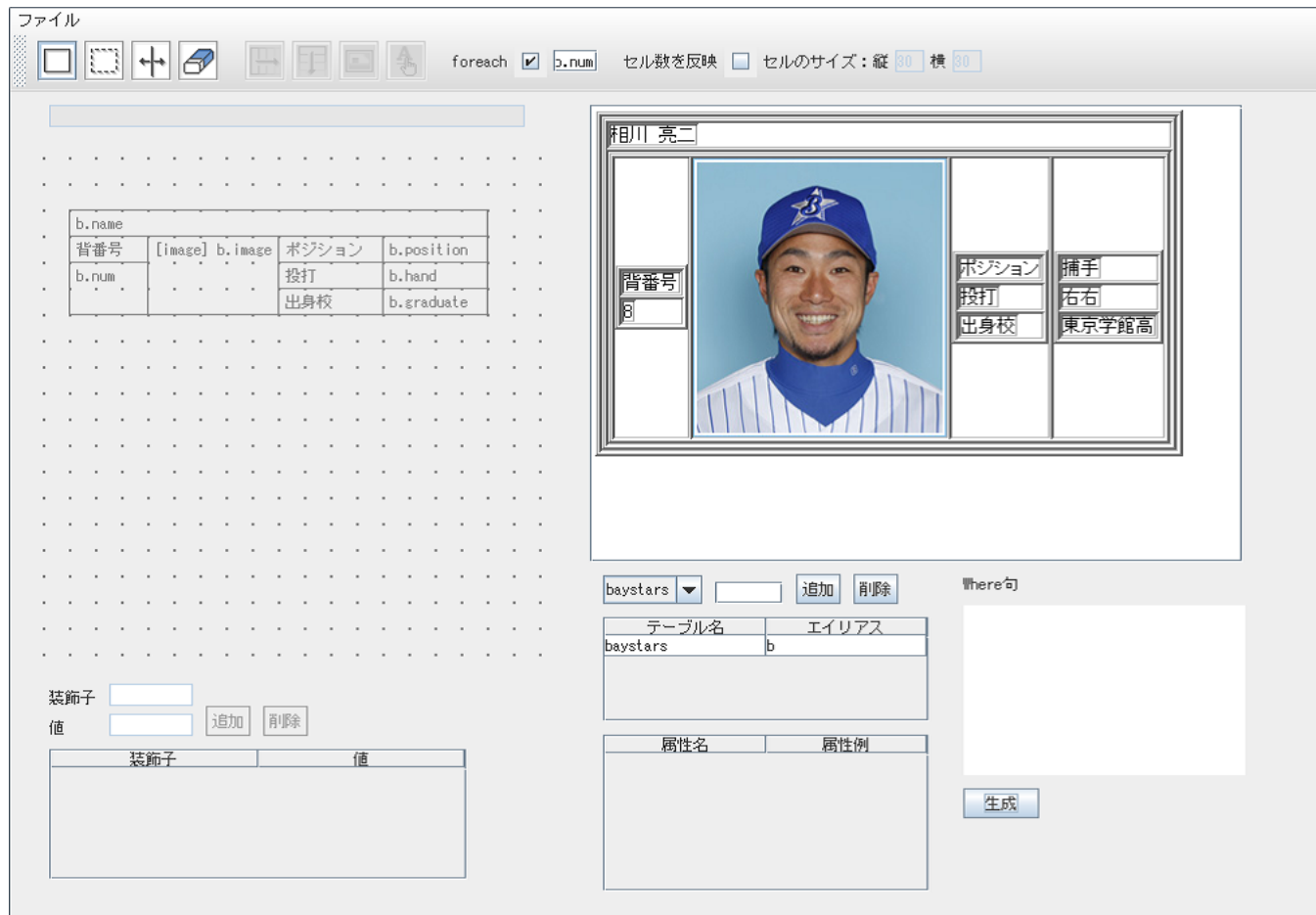


図 1 インタフェース概観

フォントやサイズなどを指定する事が出来る。

画面右部の生成ボタンを押すと、現在描かれた情報を元に SuperSQL クエリが生成される。また同時にその SuperSQL クエリが実行され、出力結果がプレビュー画面に表示される。以上がシステムの概要であり、以下では各部の詳細について述べる。

3.1 GENERATE 句

グリッド画面上の情報に対応して、GENERATE 句が生成される。グリッド上でのサイズは出力結果には表れないが、任意の設定により出力にも反映出来る。そのメリットとデメリットを以下に挙げる。

- メリット

- 入力と出力を更に近づける事が出来る
- 各データごと (例:複数の選手のデータが反復される時) のレイアウトを統一する事が出来る

- デメリット

- 全てのセルに強制的にサイズが指定されてしまうため、画像の原寸大表示など柔軟な表示が出来ない
- 指定サイズに入りきれないデータがある場合に表示が乱れる

3.2 FROM 句・WHERE 句

ユーザはプルダウンメニューからテーブルを選択し、エイリアスを付けて一覧に追加出来る。使用可能なテーブルのリスト

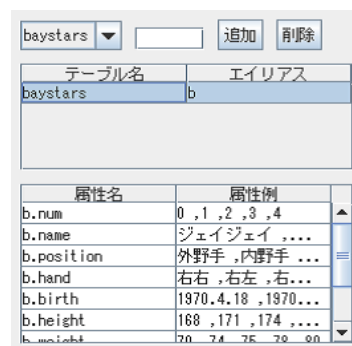


図 2 使用テーブル選択

はシステムカタログより取得され、これらは FROM 句に変換される。そしてユーザは表示された属性例をグリッド画面へドロップする事で入力が可能である。また、ユーザは画面右下に配置されたテキストボックスに WHERE 句を直接入力する。

4. 表構造の有効性

このインターフェースでは、ユーザがグリッド画面上に自由に表構造を描く事が出来る。そのため描かれた表構造が常に SuperSQL で再現出来るものであるとは限らず、その有効性を判断し、入力などの様々な操作を制限する必要がある。以下ではどのような表構造を有効とするかという定義 (4.1 節)、その

有効性を判断する手法 (4.2 節), またその例外と対処法 (4.3 節) について述べる.

4.1 有効な表構造

SuperSQL では, 長方形の中に入れ子として長方形を持たせる事で表構造を定義している. そのため, 以下の三つを表構造が有効である条件として定義する.

- (1) 線によって分割された全ての図形が長方形である
- (2) 各長方形の中に余分な線分が存在しない
- (3) 表構造全体が長方形である

ここで例と共に説明を添える. 以下図 3 において, (a) では内部に長方形でない図形が存在している. (b) では長方形の中に線が含まれている. (c) では内部の図形は全て長方形であるが, 全体が長方形ではない. 故にこの三つの例は全て不正である.

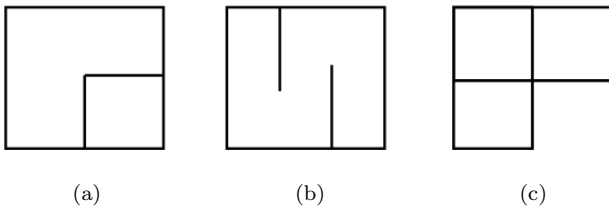


図 3 不正な表構造

4.2 有効性の判断

ここでは 4.1 節で定義した条件を, システム上でどのように判断するかについて述べる. その判断基準には各点に対し, どのように線が隣接しているかをを用いる. 各点からは最大 4 本の

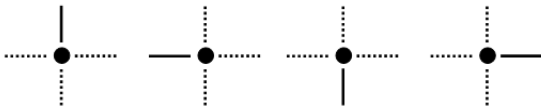


図 4 存在不可能な点



図 5 各 1 個のみ存在する点

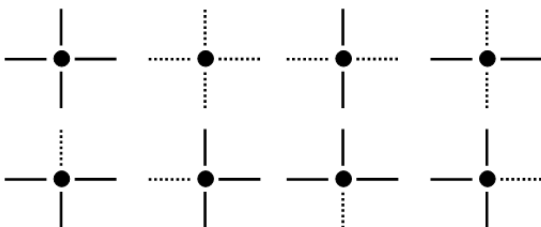


図 6 存在可能な点

線が隣接するが, もし隣接する線が 1 本である点 (図 4) が存在すれば, 必ず不正な構造となる. また隣接する線が 2 本で, か

つ折れ曲がっている点 (図 5) は 4 種類各 1 つずつのみ存在ししなければならない. この点は全体となる長方形の 4 隅を示すから, 必ず 1 つずつ必要である. 逆に 2 つ以上存在すると図 3(a) のように長方形でない図形が出来てしまう. 故に以上をまとめると

- (1) 図 4 の点が存在しない
- (2) 図 5 の各点が 1 つずつ存在する

以上の二つが条件となる.

4.3 例外

先ほど二つの条件を挙げたが, これには例外が存在する. 図 7 は 4.1 節の定義や前述の条件を満たしているが, この形状全体を縦断又は横断する直線が無いためにこれ以上分割が行なわれない. また, 縦あるいは横に並べた長方形を入れ子とする SuperSQL では表現出来ない形状でもある. この例外を排除するため, 分割が終わった後に全ての最小セルをチェックし, 内部に線が発見された時に不正な構造としている.

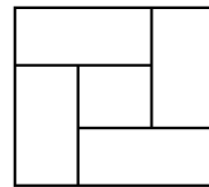


図 7 例外

5. 表構造及び情報の管理

このインターフェースでは, ユーザがグリッド画面上に自由に表構造を描く事が出来る. 以下では表構造を解析し保存する手法 (5.1), 各セルに入力された情報を保存する手法 (5.2), そしてそれらの情報からクエリを生成する手法 (5.3) を述べる. また先立って, 必要な単語の説明を添える.

- 構造情報...ユーザが描いた表の中で入れ子になっている長方形の情報 (始点座標, 終点座標, 子セルなど) を指す
- 最小セル...構造情報の中で, 子セルを持たない (それ以上分解出来ない) 長方形
- 要素情報...各最小セルが持つ, 文字列定数やデータベース属性などの情報

5.1 表構造の解析

ここでは表構造から最小セルに分割し, 構造情報を得る手法を示す. まずユーザによって入力された表全体を縦断 (横断) する線を全て探索する. するとそれらの線により表を分割する事で新たな長方形が発見される. そしてそれらの長方形について同様の処理を再帰的に行う事で全ての最小セルが発見され, 構造情報を得る事が出来る.

図 8 に例を示す. 表全体を領域に持った 0 番セルが生成され, 横線により分割された 1,2 番セルが生成される. その後 2 つのセルに探索が為され, 同様に 2 番セルから 3~5 番セルが生成される. 構造情報は以下の表 1 のように配列で保存される. また, 始点と終点はグリッド画面上の座標である.

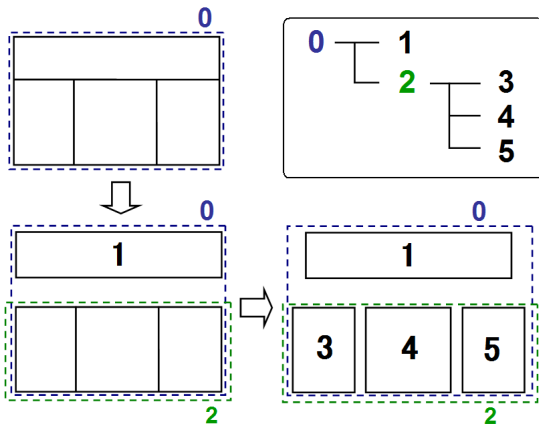


図 8 表構造解析

セル番号	始点	終点	連結	子セル
0	(0,0)	(3,3)	縦	1,2
1	(0,0)	(3,1)	null	null
2	(0,1)	(3,3)	横	3,4,5
3	(0,1)	(1,3)	null	null
4	(1,1)	(2,3)	null	null
5	(2,1)	(3,3)	null	null

表 1 構造情報の保存

5.2 要素情報の保存

各最小セルは、それぞれが対応する要素情報を持つ。しかしユーザが描画を行なう度に構造情報は更新され、古いものは失われてしまう。そのため要素情報は構造情報と別に保存する必要がある。要素情報にはポインタとなる座標と一緒に保存されており、各最小セルにおける左上の座標と照合されてクエリ生成の際に結合される。その詳細について以下 5.3 では例を挙げ、具体的に述べる。

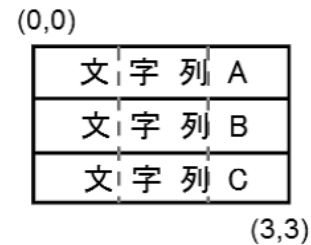
5.3 クエリの生成

入力と生成クエリの例を図 9 に、またその際に保存されている構造情報と要素情報を図 11 に示す。各セルはもし最小セルであれば対応するポインタを持った要素情報を検索してクエリとして持つ。例では 1 番セルは 1 番要素から文字列 A、2 番セルは 2 番要素から文字列 B、3 番セルは 3 番要素から文字列 C を得る。

また最小セルでなければ子セルのクエリを再帰的に得て、連結子で繋げたものをクエリとする。この処理を全体を指す 0 番セルから行なう事で GENERATE 句を得る。更にこの GENERATE 句に FROM 句, WHERE 句 (4.2 参照) を連結する事で SuperSQL クエリの生成が終了する。

6. 評価・検討

本システムによって SuperSQL クエリの作成に要する時間をどの程度短縮できたのかを検討するために、SuperSQL の構文を理解している 6 名の被験者を対象とした実験を行なった。まず実験用のデータベースと、そのデータベースを利用した徐々に複雑さを増す SuperSQL クエリを四題用意した。被験者には



```
GENERATE HTML [
  { "文字列 A"! "文字列 B"! "文字列 C" }
]!
```

図 9 入力情報と生成クエリ

セル番号	始点	終点	連結	子セル
0	(0,0)	(3,3)	縦	1,2,3
1	(0,0)	(3,1)	null	null
2	(0,1)	(3,2)	null	null
3	(0,2)	(3,3)	null	null

図 10 構造情報

要素番号	ポインタ	要素
0	(0,0)	文字列 A
1	(0,1)	文字列 B
2	(0,2)	文字列 C

図 11 要素情報

システムの使用方法と実験に用いるデータベースについて説明した後で、出力してもらいたいメディアデータのイメージ (6.1) を渡し、それを実現する SuperSQL クエリを本システムを用いて作成してもらい、生成に要する時間を測定した。手作業の場合と本システムを利用した場合のそれぞれで要したクエリ生成時間を比較することで、本システムがユーザのクエリ生成をどの程度支援できたのかを評価した。なお実験は公平を期するために、半数の被験者は本システムを先に、残りの半数の被験者には手作業による実験を先に行ってもらいそれぞれ時間を測定した。また、今回の実験では野球選手の個人情報を持ったリレーション baystars、各選手の成績を持つリレーション record をデータベースとして用いた。

6.1 メディアデータイメージ

被験者に渡した出力してもらいたいメディアデータのイメージとは次の表の様なものである。Q4 では各選手の情報、および各選手へのリンクが付いた選手一覧を合わせて一つの題とした。被験者にはこれを見ながら SuperSQL クエリを作成してもらった。

表 2 Q1. 選手名を全て表示する

選手名一覧
選手 1
選手 2
.
.

表 3 Q2. 選手の情報をポジションごとに分けて表示する

ポジション 1		
選手名 11	生年月日 11	出身校 11
選手名 12	生年月日 12	出身校 12
.	.	.
.	.	.
ポジション 2		
選手名 21	生年月日 21	出身校 21
.	.	.
.	.	.

表 4 Q3. 選手成績の一覧を表示する

名前	打率	得点	ヒット数	ホームラン
選手名 1	打率 1	得点 1	ヒット数 1	ホームラン 1
選手名 2	打率 2	得点 2	ヒット数 2	ホームラン 2
選手名 3	打率 3	得点 3	ヒット数 3	ホームラン 3
.
.

表 5 Q4-1. 選手の情報

背番号	背番号 1	選手画像 1	
名前	名前 1		
生年月日	生年月日 1		
ポジション	ポジション 1		
出身校	出身校 1		
2008 年成績			
打率	得点	ヒット数	ホームラン
打率 1	得点 1	ヒット数 1	ホームラン 1

表 6 Q4-2. リンク付き選手一覧

ポジション 1	
背番号 11	選手名 11
背番号 12	選手名 12
.	.
.	.
ポジション 2	
背番号 21	選手名 21
.	.
.	.

6.2 結 果

結果は図 12 のようになった。手作業時に比べて SuperSQL クエリを生成するのに要する時間を大幅に短縮できたことがわかる。今回の実験の被験者の多くは皆 SuperSQL に関して非常に高い知識を持った者であったが、それでもかなりの時間を短縮できた。対象をあまり SuperSQL に関する知識がない者とする

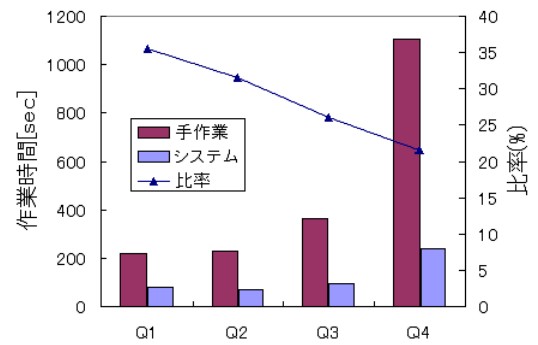


図 12 SuperSQL クエリ生成に要する時間の比較

れば、手作業での作業時間は増大し、システムの有用性はより顕著に示されることが予測できる。

そして、実験の後半になると手作業・システム共に所要時間は増加している事が分かる。一方で手作業によるクエリ生成からシステムによるクエリ生成で、生成時間がどのように変化したかを示す比率に注目すると、Q1 から Q4 に進むにつれ下がっている事が分かる。この比率に注目すると、システムの利用ではクエリの複雑化に柔軟に対応出来ている事が分かる。これは単純な表構造であればクエリも単純になるが、表構造が複雑化するとその度合いに比べてクエリはより複雑化してしまうためであると考えられる。

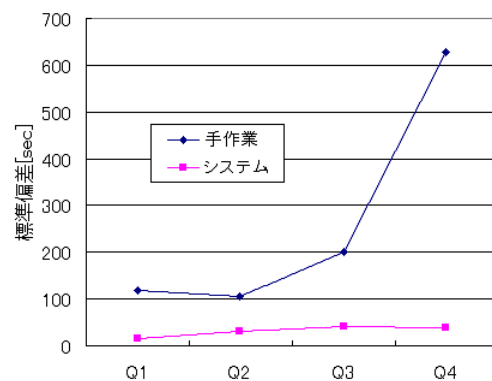


図 13 作業時間の標準偏差

さらに、両者の標準偏差を比較したものが図 13 である。今回対象とした被験者 6 名は、日常的に SuperSQL に触れている研究生や課題として触れた事があるのみの学生など習熟度に差があった。複雑なクエリになるほど習熟度の差が顕著に表れるため、手作業における標準偏差も大きく増加している事が見て取れる。一方システムを使用した場合には標準偏差に大きな変化は見られず、本システムの操作は SuperSQL の習熟度による影響を受けずに行なえるものである事が分かった。

7. 結 論

本研究では従来の SuperSQL で存在していた入力と出力の感覚的な距離を解消するため、視覚的なインターフェースを提案した。またそのために必要となる表構造を解析する手法や、情報管理構造及びそこからクエリを生成する手法を提案した。更

にクエリの生成にかかる時間をシステムの有無によって比べる
事で、本システムの有用性を確かめた。

文 献

- [1] SuperSQL: <http://SuperSQL.db.ics.keio.ac.jp/>
- [2] M. 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] 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] 横浜ベイスターズ - データベースサンプル：
<http://www.baystars.co.jp/>