

TFE における関数の導入 - 集約関数およびデータ視覚化関数 -

2D-9

軽部和幸[†] 遠山元道[‡]

[†] 慶應義塾大学 理工学研究科 管理工学専攻 [‡] 慶應義塾大学 理工学部 管理工学科

1 はじめに

TFE(Target Form Expression) はターゲットリストの拡張で、関係データベースからの検索結果の出力表に階層構造と厚み方向を含むレイアウトの自由度を与えるものである。本研究では TFE にデータ加工のための関数の導入を提案する。TFE において有用となる関数としては、従来のデータベース問い合わせ言語で導入されている集約関数、および問い合わせ結果からグラフ生成などを行なうデータ視覚化関数などが考えられ、今回はこの2種類の関数に焦点をあてる。集約関数においては TFE のグルーピングを生かすことにより、自然な関数記述を実現し、グラフ生成関数においては、TFE 処理系から外部のグラフ生成ツールを利用するシステムの構築を図った。

2 集約関数

複数の店舗に複数の売場があるチェーン洋品店の毎月の売上を記すデータベースを考え、

GENERATE [店, [売場, [月, 売上]]!]!

の問い合わせ結果が次の表のようになるとする:

横浜	子供服	1	350
		2	400
		3	320
	婦人服	1	700
		2	820
		3	800
川崎	子供服	1	280
		2	300
		3	340
	婦人服	1	640
		2	600
		3	720

横浜	子供服	1070
	婦人服	2320
川崎	子供服	920
	婦人服	1960

従来の問い合わせ言語の集約関数においては、QUEL における by 節のような記述を行なうことにより、グルーピングを指定したうえでの集約計算を行なうことができる。一方で TFE においては出力表のグルーピングを考慮して、集約計算のグルーピングを決定することにより、より自然な集約関数の記述が可能である。

例えば上記の例に集約関数 sum を使用して、売場ごとに売上の累計を求めたいときに、従来の集約関数の記述に従えば、

GENERATE [店, [売場, sum(売上 by 店, 売場)]!]!

という形になる。しかし、この TFE 式においては、売上は既に店、売場ごとにグルーピングされている。したがってこの TFE 式のコンテキストにおいては、売上の合計は通常、店、売場ごとにグルーピングされたものとして捉えられる。したがって今回拡張した TFE においては、上記の式の sum 中の by 以下を省いて、

GENERATE [店, [売場, sum(売上)]!]!

と記述すれば、店、売場ごとにグルーピングされた合計が得られるようになっている。この問い合わせ結果は次のようになる:

ちなみに、あえて TFE 式のコンテキストに反するグルーピングで集約計算を行ないたいときは QUEL と同様の by 節あるいは by nothing(グルーピングなしの集約計算) を明示的に示せばよい。

3 グラフ生成関数

TFE はそのまま利用できる表現力豊かな検索結果の出力の生成を目標としている。そこで考えられる機能がデータからのグラフ生成である。TFE の特徴の一つは、

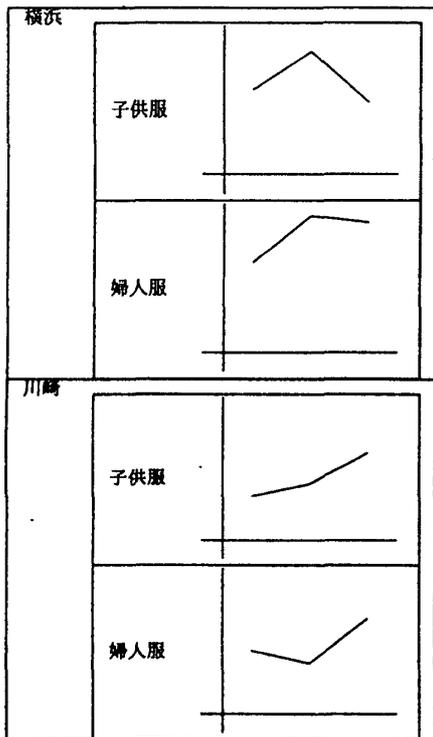
Introduction of Functions into TFE - Aggregate Functions and Data-Visualizing Functions -
 KARUBE Kazuyuki, TOYAMA Motomichi
 Department of Administration Engineering, Faculty of Science and Technology, Keio University.

従来の問い合わせ言語の出力にはなかった表の入れ子 (= グルーピング) の概念を導入したことであり、ある部分のデータからグラフを生成し、それを入れ子にするという考え方もこの延長線上にあるといえる。この意味では、グラフ生成関数もまた TFE においては、広い意味での集約関数の一つとして捉えられる。本研究では、TFE のグルーピングによって得られた部分データセットをグラフ生成ツールに渡し、そこで生成したグラフを TFE 処理系の出力に組み込むシステムを構築し、グラフ生成を試みた。

例えば前節の例において、店、売場ごとの毎月の売上の推移を示す折れ線グラフを描くとすると、次のような問い合わせとなる：

```
GENERATE [店, [売場,
graph('line', 月, 売上)]!];
```

この出力結果として、次のようなグラフを入れ子とする表が得られる：

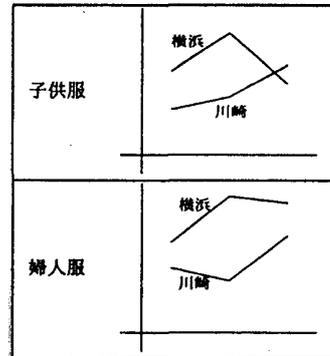


また、何枚かのグラフを重ねて1枚にまとめることもできる。例えば売場ごとに各店のグラフをまとめるには、次の問い合わせのように店を graph 関数の第4引数に持つて来る。：

```
GENERATE [売場,
graph('line', 月, 売上, 店)]!];
```

この例では、子供服、婦人服それぞれの売場について、横浜店と川崎店のグラフが一枚にまとめられ、次のよう

に個々の売場について店同士で売上の推移を比較できるグラフが得られる：



グラフ生成関数にはグラフ生成ツールの力次第で様々なオプション指定の可能性が考えられる。今回はそのうち基本的と思われるものを実現した。例えば、棒グラフを作成したいときは、上記の例の graph 関数の第1引数 "line" を "bar" にかえればよい。また、グラフが出力される際の大きさを自由に指定することも可能で、例えば横5cm、縦10cmの大きさにしたい場合は、
`graph(~)@{width=10cm,height=5cm}`
 と @ 以下に縦横の長さを指定すればよい。(縦または横のみの指定も可)

4 その他の関数

今回紹介した関数以外にも、TFE の出力の表現力を高めるために、次のような関数を導入している。

- イメージ組み込み関数。具体的には eps ファイルなどイメージデータを含むファイルの名称を、出力媒体上でイメージに変換する。
- その引数となった項目を表示しない omit 関数。クロス集計表の作成に有効である。

5 まとめ

最後に本研究の成果をまとめる。

- TFE のグルーピングの概念に基づいて、集約関数の記述法の改善やグラフ生成関数の実現を行なったことにより、TFE のデータ編集機能を向上させた。
- データのプレゼンテーションのための新しい関数 (グラフ生成関数を含む) を導入したことにより、より視覚的表現力に富んだ出力が可能となった。

参考文献

[1] 遠山元道他: レイアウト式 TFE の拡張, 情報処理学会データベースシステム研究会資料, 95-DBS-104(1995) pp. 217-224