

お節介データベースにおける問合せ処理

4F-10

友重理恵 阿部直子 猪首俊幸 伊藤倫奈
田島哲哉 立石教枝 水岡一茂 芝野耕司
東京国際大学

1. はじめに

関係データベースにおける検索は、SQL[1]の問合せにあたる。問合せ文を作成するSQL言語自体は、他のプログラミング言語に比べて簡単であり使いやすいが、データベースに格納されているデータ（表名、列名、値、データ型など）に関する知識は、事前にすべて覚えていることが前提となる。すなわち、検索対象のデータベースに関する知識をもたないことには、問合せを組み立てることができない。

特に、日常的に対象となるデータベースを取り扱っていないユーザのアドホックな問合せの場合、このデータベースに関する知識を必要とすることが障害となる。

そこで、データベース管理システム中にあるシステムカタログ[2]に格納されているデータをシステム自体が、積極的にユーザに順次開示することにより、ルーチンユーザでなくても、すぐに問合せ文を生成でき、検索を可能にすることができる「お節介」データベースを提案する。

また、今までの検索は一過性であり、検索し終わると問合せ文は消される、もしくは何も活用されないまま放置されていた。このシステムは、ユーザの検索履歴情報（問合せ文）を蓄積し、問合せパターンを解析することで、検索する過程をパーソナライズして、ユーザの一人一人に合わせた検索方法を提供する。

2. お節介の概要

ここで用いた「お節介」とは、既存のデータベースが常に、ユーザの入力を待ってから動きはじめるのに対し、能動的に情報を提供することを意味する。つまり、ユーザが何を選択していけばよいかわからない場合、システム側でただ入力を待ち続けるのではなく、何かを選択し、情報を開示していくことで、それが何を意味するのかをユーザに示す。すなわち、このシステムでは、ユーザの指示を待つのではなく、「お節介」をするのである。

平均的なデータベース環境では、数百の表が存在

し、数千の列が存在する。列の値に至っては、数百万以上の異なった値が存在する可能性がある。

こうしたデータベースに一時的なユーザ又は初めてのユーザが、問合せを行うことは難しい。この難しさを「お節介」によって、解消することがこのシステムの狙いである。同時に、この「お節介」では、一人一人のユーザの好みを検索履歴をもとに推測し、より適切な情報を提示することを行う。

3. SQL

SQL問合せ文の基本形は以下のとおりである。

```
SELECT  <列名1>, ... <列名n>  
FROM    <表名1>, ... <表名m>  
WHERE   <検索条件>
```

ここでこのSQL問合せ文を生成するために必要な情報は、SQL文法並びに表、列及び列の値に関する情報である。

また、構文からこれらの表、列及び列の値には、依存関係がある。すなわち、WHERE句（探索条件）が決定されれば、自動的にFROMの表名が決定される。同様に、SELECTの列名が決定されれば、FROMの表名も自動的に決まる。

4. ユーザインターフェイスとSQLの生成

実際の画面上でユーザに提供していく情報は、表名、列名及び列の値である。それらを能動的に流し、与え続けながらユーザを誘導していき、ユーザの流れに対する反応から、SQL問合せ文を組み立てていく。

ユーザの反応がなにもないときの情報の提供の仕方は、「カテゴリー」内の「表」を提供し、「表」の内容を「表→列→列の値」、そして、次の「表→列→列の値」というように流れていく縦の流れと、「表」から「表」へと情報を流してゆく横の流れの二通りが基本である。表、列及び列の値は、上から「表→列→列の値」という階層で「木構造」をなしている。マウスの操作によりユーザの望む方向へ、すなわち、深さ優先か又は、広がり優先提示のいずれかにその場その場で反応できるようになっている。ここで、「カテゴリー」とは、データベースのアプリケーションシステムのサブシステムに対する概念である。

Meddlesome DataBase

Rie Tomoshige, Naoko Abe, Toshiyuki Ikubi,
Michina Ito, Tetsuya Tajima, Norie Tateishi,
Kazushige Mizuoka, Kohji Shibano
Tokyo International University

「カテゴリー」、「表名」、「列名」及び「列の値」の各々の画面での表わし方の例を、下図に示す。

カテゴリーは画面に並べて表示し、デモが流れている。ユーザは、マウスで操作しながら検索を進めていくが、マウスのクリックは、一回が階層を下にいく選択で、二回のクリックは決定である。決定させると決定した内容が、下に設けた「問合せ文生成状況」のウィンドウの空欄を自動的に埋めて行く。マウスのドラッグは、上は階層をさかのぼり、下は下の階層に行くことを表わす。左右はその階層のそれぞれの同じ階層での横方向を操作する。

ウィンドウが開いてそれぞれの内容が流れてくるのであるが、その際の流す速度の調整は画面上のスライダーにより行われる。ウィンドウの上の一部をクリックするとデータの流し方を変化させるポップアップ表示が出る。内容はエントリー順、大小の順である。

ウィンドウは、カテゴリーごとに検索の途中でポップアップ状態で残しておくことが可能である。あわせて、現在の表、列及び列の値の表示ウィンドウはユーザにとっての周辺視野となる。また、他のカテゴリーで開いているウィンドウの流れもお節介の一つとなる周辺視野である。

すなわち、ユーザはマウスを駆使し、流れてくる情報を処理しながらSQL問合せ文を組み立てるのである。

5. 履歴情報の援用

このシステムでは、ユーザの問合せ文を蓄積し、検索の方法を解析する。つまり、表名、列名、列の値、および利用しているSQL言語要素のユーザごとの利用情報を解析し、システム側からユーザに提示する画面表示の提示順序をユーザに合わせ、変更する。

蓄積した情報の解析の基本は、問合せ文に使用し

た表名、列名、列の値、SQL言語要素の使用頻度及び使用時期である。この情報で、ユーザの検索傾向を解析する。

検索傾向とは、次に示す通りである。

(1) ユーザが普段使用するカテゴリ。

(2) SQL問合せ文の組み立て方。(表、列、列の値の何をユーザが一番重視するか。すなわちSELECT、WHEREのどちらから問合せ文を作成するのか、など。)

この検索経歴をもとに並べ替えを行うことによって、検索傾向に応じてよりユーザにあった情報を提示することができる。

次に利点を挙げる。

(1) 使用頻度の多い事柄に関連した事項も優先的に出す。

基本的には表名を最初に提示している場合でも、SELECTから問合せ文を組み立てるユーザには、列名のウィンドウが開かれている状態を提供する。その列名の提示の制御の方法もユーザの検索傾向に合わせる。

(2) 使用時期を考慮し、直前の検索履歴を重要視して解析する。

(3) 4.で述べたように情報提示の方法も「表」から「表」へとといったように、履歴情報によって変化していく。

6. 性能上の問題

このシステムでは、現在のコンピュータの性能向上を前提に、システム側から積極的に情報を提示することによって、データベース検索をより容易なものにすることを目指した。こうしたシステム側からの情報提供はシステムに負荷を与える。しかし、ここで提示する情報の多くはシステムカタログ中に存在し、SQL文のコンパイル時に用いる情報と同じである。この意味で、このシステム自体は殆どの場合、あまり多くの負荷をかけない。

7. おわりに

このシステムによって、検索に関わるユーザの負担を軽減するとともに、データベースの内容に熟知していないユーザにとっても、適切な検索を可能にする。

参考文献

- [1] 日本規格協会 JIS X3005 データベース言語 SQL 1990
- [2] 日本オラクル株式会社 Oracle7 Server 管理者ガイド 1992