# JAM：日本語質問文によるデータベース検索システム

日吉 茂樹　村木 一至
（日本電気(株) C&Cシステム研究所）

## 1. はじめに

データベースの普及に伴なってノンプログラマによるデータベース検索の機会が増大してきており、また検索要求の多様化，非定型化も進んでいる。そして、このような背景のもとに，データベース検索のための簡易でかつ柔軟性のある親しみやすいユーザインタフェースが要望されている。我々は、以上の観点から，データベース検索におけるユーザインタフェースの高度化を目指す1つのアプローチとして日本語質問文によるデーベース検索システムの方式を検討した。

自然言語によりデータベース検索を行なうシステムの研究開発は近年活発であり、日本語によるものとして藤崎等によるヤチマタ[1]や島津等によるMSSS'78 [2]などが既に発表されている。また，英語によるものとしてCoddによるRENDEZVOUS [3]，Waltzによる PLANES [4]，SacerdotiによるLADDER [5]などがある。このような研究開発の技術的背景として，データベース管理システムの確立、自然言語解析モデルの提案、LISPなどの言語によるシステム記述の容易化といった要因が大きく貢献している。しかしながら、質問文の意味解析や文脈管理に関してはまだ確定的な方式がなく，効率的かつ有効な方式が望まれている。また，データベース管理システム側にも言語表現の枠組に合わせたビューの提供，あいまい情報検索，文脈情報の保持といった利用者との会話を支援する機能の実現が必要である。我々は、以上の技術的要求に対する1つの解を求めることを目的として日本語質問文の意味解析モデル，文脈管理方式，およびデータベース問合せ機能の設計を行ない，モデル検証の目的でJAM（Japanese Access Method for a database）と名付けた実験システムを開発した。

JAMは、ACOS 700のTSS環境下でLISPインタプリタとデータベース管理システムINQ [6]を利用して実現されている。データベースには京都付近の名所（主に寺院，神社）に関する情報とその名所に行くための交通機関の情報とが格納されており、検索条件の対象となるデータ項目には索引を設けて効率的検索をはかっている。

本稿では、JAMの設計上の特徴，プログラムモジュール構成について紹介するとともに，質問文解析アプローチ、データベース検索言語、検索機能などについて述べている。

## 2. システムの概要

### 2.1 設計上の特徴

(1) 自然言語（日本語）とFQL（Formal Query Language）の2つの検索言語インタフェースを利用者に提供する。自然言語の質問文は、一旦FQLに翻訳して実行する。

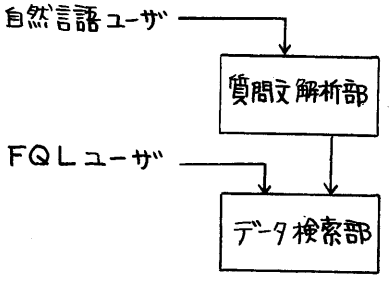


図1. JAMの言語インタフェース

(2) データ属性に基づく意味解析モデルとしてVISÉ(View Semantics)モデルを設けた。VISÉモデルを用いることにより、質問文からFQLを生成するための情報を効率的に抽出しうる。
(3) 文脈管理機能により以前の質問に関連した質問文における省略を解決する。また、以前の質問での検索結果に基づく文脈表現をFQLレベルで可能とする。

## 2.2 システムの構成

JAMシステムのプログラムモジュール構成を図2に示す。以下に各プログラムの機能を説明する。
(1) Syntax Analyzer: 質問文(ローマ字入力される)を単語に分かち書きすると同時に構文のチェックを行なって構文木を生成する。構文規則は、日本語の係り受け文法に基づく書換え規則である。語彙辞書には、各語に対応するデータ属性名の候補が記述されている。この情報は、そのまま構文木に取込まれる。
(2) Semantic Analyzer: 質問文の構文木を入力とし、問合せ(FQL)生成に必要な情報を抽出する。特にデータ属性名を確定的なものとする。
(3) Context Management: 質問応答における文脈情報を維持し、質問文での検索対象指定や検索条件指定の省略を文脈情報を用いて埋合わせる。また、生成されたFQLを利用者に提示して利用者の意図との整合性を確認する。
(4) Formal Query Generator: 質問に対応したFQL文を生成する。
(1)～(4)のプログラムが図1における質問文解析部を構成する。
(5) FQL Interpreter: FQL文を解釈し、DBMSに対する検索コマンド列に変換する。
(6) Reply Generator: データベース検索結果を編集し、表示出力する。

なお、DBMSとして利用したINQは、関係データベースに対する検索機能を提供するものであり、複数の表を結合した仮想表の上での検索を行なうことができる。

また、Syntax Analyzerは構文解析用ツールである拡張LINGOL[7]の上に作成された。

## 3. 質問文解析モデル

ここでは、例を用いて質問文の意味解析と文脈解析の方式を簡単に説明する。詳しくは、[8]を参照されたい。

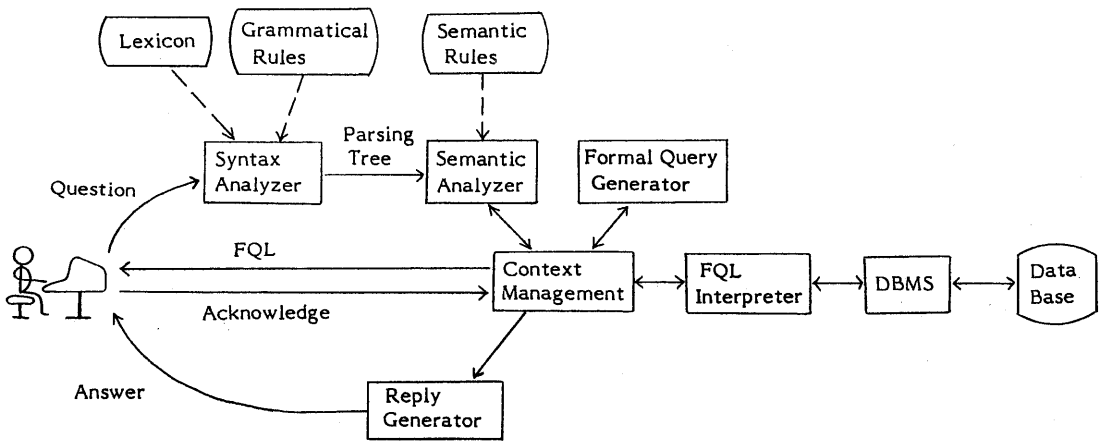


図2. JAMのプログラムモジュール構成と参照情報

いま、データベースは名所(MEISHO),交通(KOUTSU)の2つの表からなり、それぞれの表は次のような属性名を持つものとする。

MEISHO (MM, BA, MT, KR, KJ)

KOUTSU (KS, SE, ME, JR, JJ)

但し, MM: 名所名　KS: 交通手段名
BA: 場所　SE: 始点駅名
MT: 名所タイプ　ME: 目的駅名
KR: 見学料金　JR: 乗車料金
KJ: 開門時刻　SJ: 所要時間

なお, 下線部は候補キーを示す。

このデータベースに対する2つの質問文とそれらから生成されるFQL文の例を示す。

質問文1: 東京から京都まで何時間かかりますか?

FQL1: SELECT DATA (SJ HOUR, KS)
WHERE (SE EQ トウキョウ
AND ME EQ キョウト)

質問文2: 名古屋からの料金は?

FQL2: SELECT DATA (JR, KS)
WHERE (SE EQ ナゴヤ
AND ME EQ キョウト)

質問文1において, 単語としての「東京」や「京都」のデータ属性は(MM, SE, ME)のいずれかであり曖昧である。しかし, FQL文を生成するためには最終的に1つの属性名に対応づけなければならない。JAMでは、質問文中の各語に対し, <軸, 属性制約, 値>の3組の集合を一種のFrameとして割当てる。そして, Frame間のUnify演算により曖昧性を解決する。このしくみをVISÉモデルと呼ぶ。

VISÉモデルによる意味解析は3つのステップからなっている。

STEP1: 名詞類の属性の明確化
Frame間のUnify演算を行なう。

例えば、

東京 { <ATT (MM,SE,ME) ( )>
<VAL (MM,SE,ME) (トウキョウ)> }

から { <CASE (FROM-T,FROM-P) (カラ)
<ATT (KJ SE) ( )
<VAL (KJ SE) ( ) }

の2つのFrameのUnify演算で次の結果を得る。

東京から { <CASE (FROM-T, FROM-P) (カラ)
<ATT (SE) ( )>
<VAL (SE) (トウキョウ)> }

STEP2: 動詞の持つFrameと名詞類の属性との突合せ
例えば、動詞「かかる」は、KR, JR, SJの3つの属性に関するFrameを持ち、名詞類の属性との突合せにより整合性チェックが行なわれる。

STEP3: データベース・リレーション構造との突合せ
例えば、質問文1の構造は.

[ 検索対象; SJ
検索条件; SE, ME ]

で与えられる。ここで, SJが(KS, SE, ME)に関数従属であることからKSの情情が必要となる。この場合, 検索対象として属性名KSを付加する。(但し, 文脈情報が空の場合に限る。)

次に質問文2をみると, その構造は

[ 検索対象; JR
検索条件; SE ]

で与えられる。しかし, JRは(KS, SE, ME)に関数従属であり、KSとMEの値を必要とする。このとき、文脈情報として質問文1でのMEの値'キョウト'を条件値として付加し, KSを検索対象として付加する。

以上のようにして質問文の意味と文脈に関する解析が、データ属性と属性間の意味的従属関係にもとづく手法により効率的に行なわれる。なお、会話例を5節に示している。

## 4. 形式的問合せ言語FQLとデータ検索機能

### 4.1 FQL表現

FQLは、日本語質問文に対する言語解析の出力表現であると同時に，またそれ自身がデータベース検索用問合せ言語である。従がって，JAMの利用者はFQLによっても直接データベースを検索できる。

FQLの表現形式は、SEQUEL[9]に近いものを採用している。特徴的な表現は次の点にある。

(1) ベースリレーションを結合した仮想的なリレーションに対する問合せ表現であり、従がって通常の検索ではリレーション名を指定する必要がない。例えば、3節に示したMEISHO, KOUTSUの2つのリレーションに対しては，名所名(MM)と目的駅名(ME)を結合キーとしてMEISHO-KOUTSUという仮想的なリレーションが定義されている。

(2) YESまたはNOを答として得る質問を表現できる。すなわち、属性値間にある関係が存在するか否かを問合せしうる。

(3) 前の検索結果を文脈として利用した問合せ表現が可能である。

FQLにより表現しうる検索機能を、例を用いて以下に示す。なお、FQLの構文形式は付録の頁を参照されたい。

[例1] YES-NO質問

金閣寺の見学料金は200円以上ですか？

```
SELECT YN
WHERE ( MM EQ キンカクジ
  AND KR GE 200 )
```

[例2] 件数表示

洛北にはいくつの名所がありますか？

```
SELECT CNT (MEISHO)
WHERE ( BA EQ ラクホク )
```

[例3] ユニーク件数表示

洛東には何種類の名所タイプがありますか？

```
SELECT UCNT (MT)
WHERE ( BA EQ ラクトウ )
```

[例4] データ検索

洛北または洛東にある名所の名前と開門時刻を求めよ。

```
SELECT DATA (MM, KJ)
WHERE ( BA EQ ラクホク
     OR BA EQ ラクトウ )
```

以上の4つがFQLの標準的検索機能であるが，この他に4.2～4.4に述べる特徴的機能を携えている。

### 4.2 FQL上での文脈表現

質問文における省略および指示代名詞の解決は、3節に述べた方式によりContext Management (CM)が行なうが、省略や指示代名詞の対象が以前の質問によるデータベース検索結果であるような場合には、文脈情報としての検索結果をデータベース中に保持し，次の検索と文脈情報との関連をFQL上に表現することによりCMの負荷を軽減しうる。また，このようにすることで、CMが直接データファイルをアクセスすることがなくなるので，質問文解析部とデータ検索部との独立性が高められる。

FQL上での文脈表現機能は，次の2つのタイプに分類できる。

(1) 以前の質問でのデータベース検索結果を対象とし，その中から次の検索結果を得るもの。

(2) 以前の質問でのデータベース検索結果を次の検索のための基準値として用いるもの。この場合、2つのリレーションの結合演算を含む検索処理を2つのFQL文に分割して簡単に表現することも可能となる。

以下にFQLでの文脈表現の例を示す。

[例5] 文脈情報を検索対象とする表現 ── (1)の例

その中で，見学料金が300円以下の名所は？

```
SELECT * DATA (MM)
WHERE (KR LE 300)
```

SELECTの後に付された'*'が1つ前の質問の検索結果を検索対象とすることを示している。

[例6] 文脈情報を基準値として用いた検索表現 ── (2)の例

見学料金がそれより安い名所は？

```
SELECT DATA (MM)
WHERE (KR LE *)
```

条件値の位置に指定された'*'が，文脈情報を検索条件値として用いることを示している。

[例7] 結合条件の文脈表現 ── (2)の例

京都駅まで歩いて行ける名所の名前と見学料金は？

⇒ Q1：京都駅まで歩いて行ける所は？

```
SELECT DATA (SE)
WHERE (ME EQ キョウト
  AND KS EQ トホ)
```

Q2：そこの名所名と見学料金は？

```
SELECT DATA (MM, KR)
WHERE (MM EQ *SE)
```

このように，2つのリレーションに関連した検索を，それぞれのリレーションに対する検索文Q1，Q2に分解して表現することができる。但し，この際に2つのリレーションの結合キーを'*'を含む結合条件で明示している。Q2の実行時には，Q1での検索結果としてのSE値が*SEに代入されるので，この結合条件は単純条件式として扱われる。

## 4.3 経路探索機能

交通機関情報（KOUTSUリレーション）を用いてある地点から他の地点までの経路を求めるのに，内部的な結合を要する場合がある。いま，図3のような情報がデータベースに格納されているとしよう。

| KS | SE | ME |
|---|---|---|
| バス | 東京 | 京都 |
| 新幹線 | 東京 | 京都 |
| バス | 京都 | 金閣寺 |

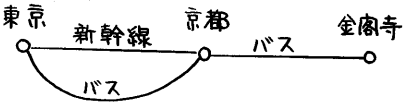


図3. 交通手段データと経路グラフ

このとき，次の質問文を考える。

[例8]

東京から金閣寺までの交通手段は？

```
SELECT DATA (KS)
WHERE (SE EQ トウキョウ
  AND ME EQ キンカクジ)
```

質問文を例8のようなFQL文に変換して実行すると解を得られない。なぜなら，図3の経路グラフによると東京 新幹線→ 京都，京都 バス→ 金閣寺のように2つの交通手段を利用すれば行くことができるが，一つの交通手段では行けないためである。しかし，質問者が2つ以上の交通手段を用いてでも目的地まで到達できる経路の探索を要求していることは質問の意図から明らかである。そこで，例8のFQL文は，タプル変数$t_1$, $t_2$を用いて次のように展開される。

```
SELECT DATA (t1·KS, t1·ME, t2·KS)
WHERE (t1·SE EQ トウキョウ
  AND t1·ME EQ t2·SE
  AND t2·ME EQ キンカクジ)
```

これは，1回の内部的結合操作を表現したものである。一般には，経路探索のためにn回の内部的結合を必要とする。

通常のデータベース検索言語では，結合演算を必要とする場合に陽に記述するが，自然言語の質問文ではこの例

のように陽には表現されないことがある。このためデータ検索部にデータの性質に応じて内部的結合演算を起動する機能を携えている。

経路探索処理は、交通路のnodeである駅とedgeである交通手段の数が多いと探索空間が時として膨大となる。従がって，問合せ言語FQLを直接利用するレベルの利用者には経路に関する知識をFQLに埋め込む形での検索効率化を許すことにした。すなわち、

① 乗換え駅名（XE）

② 乗換え回数（XE-CNT）

をFQL中に指定することにより、無駄な探索を制限するものである。XEとXE-CNTは、内部的結合演算で動的に生成される仮想的データ属性名である。

［例9］乗換え駅名指定を例8に付加

```
SELECT DATA(KS)
WHERE ( SE EQ トウキョウ
  AND XE EQ キョウト
  AND ME EQ キンカクジ )
```

［例10］乗換え回数制限を例8に付加

```
SELECT DATA(KS)
WHERE ( SE EQ トウキョウ
  AND ME EQ キンカクジ
  AND XE-CNT LE 1 )
```

日本語文での質問の場合にも利用者から会話的にXE値，XE-CNT値を得ることにより経路探索の効率化をはかりうる可能性がある。

## 4.4 あいまい値検索

日本語文による質問での表現のあいまいさは、データ属性名についてのあいまいさの他に検索条件値についてのあいまいさも存在する。しかし，データベース検索処理を行なう前にこのあいまいさは解決されなければならない。このための1つの解決方法は、実世界におけるあいまい表現をある意味に限定し，あらかじめ定義しておくことである。

FQLで扱かうあいまい表現は次の範囲のものである。

(1) 属性値の大小関係に関するあいまい表現

H：高い，長い，大きい，etc.

M：HでもLでもないもの

L：低い，安い，短かい，小さい，etc.

これらのキーワードは、あらかじめシステムに登録された基準値を用いた表現に自動的に変換される。

［例11］

```
SELECT DATA (MM)
WHERE (KR EQ H)
        ⇓
SELECT DATA (MM)
WHERE ( KR GE 300 )
```

Hに対する基準値

(H,M,L)による表現は、属性毎に絶対的基準値に変換されるため、条件付検索では該当データなしとなる可能性がある。そこで，検索条件をみたす集合の中での相対的大小関係による表現としてキーワードIMIN,IMAXを導入した。これは、それぞれ検索条件をみたす集合の中から指定属性値の最大のものと最小のものを求めるためのものである。さらに，利用者の要求した件数だけ上位又は下位の値から順に出力する機能も検討中である。

(2) 属性値の集合名での表現

例えば、(ラクトウ ORラクセイ ORラクナン ORラクホク ORラクチュウ)なる場所に関する条件値を'キョウト'で代表させる。

(3) 属性名の同義語又は簡略表記

FQLで検索する場合に，同義語又は簡略名で属性名を記述することを許す。

(2)，(3)の機能は、DBMS INQがシソーラスを管理することにより提供されている。

(4) 条件値の誤つづり訂正

利用者があやふやな条件値に'#'印を付すことにより、一文字違いのデータまでをデータベースから自動的に検索する。

## 5. 会話例

日本語質問文によりデータベース検索を行なうときの利用者とJAMとの会話例を示す。

利用者が質問文を入力すると，JAMがFQL文に翻訳し，そのFQL文を日本語文で表現したものを表示して利用者に実行の了解を得る。次に検索結果の件数が表示され，利用者が結果の表示方法を指示するとそれに従がって結果が表示される。

なお，3の質問文での指示代名詞'KOKO'を'キョウト'と認識するしくみは，TRIP-NET[8]を利用した移動地点の履歴管理機能による。

```
ニュウリョク モードハ ?
     A   ケイシキ ゲンゴ
     B   ニホンゴ
= B
シツモン ハ ナーニ ?
$ TOKYO KARA KYOTO MADE IKURA
  KAKARIMASUKA?

アナタノ シツモンハ
     シテン エキメイ カ トウキヨウ デ
     カツ モクテキ エキメイ カ キヨウト デ アル
     ヨウナ
     ジョウシャ リョウキン ト
     シテン エキメイ ト
     モクテキ エキメイ ト
     コウツウ シュダン
     ヲ オシエテクダサイ
デスカ? カクニンシテ クダサイ
     A   ハイ ソウデス
     B   イイエ チカイマス
     C   モウ ヤメマス
? A
コタエ
     2. ケン カイトウ シマシタ
ヒョウジ シテホシイ ケンスウ ヲ キーイン スルカ ,     マタハ
シタノ A-D ヨリ エランデ キーイン シテクダサイ
     A   ゼンブ ヒョウジ スル
     B   ジョウケンチ ノ アタイ ヲ カエテ シツモン ヲシタイ
     C   ベツノ シツモン ヲ シタイ
     D   シュウリョウ シタイ
? A
     シテンーエキメイ トウキヨウ
     モクテキーエキメイ キヨウト
     コウツウーシユダン シンカンセン
     ジヨウシヤーリヨウキン 9500.

     シテンーエキメイ トウキヨウ
     モクテキーエキメイ キヨウト
     コウツウーシユダン バス
     ジヨウシヤーリヨウキン 7100.
ツギニ トウシマスカ ?
     A   ニホンゴ デ シツモン ヲ シタイ
     B   ケイシキゲンゴ デ シツモン ヲ シタイ
     C   シュウリョウ シタイ
? A
シツモン ハ ナーニ ?
$ NAGOYA KARAWA?
アナタノ シツモンハ
     シテン エキメイ カ ナゴヤ デ
     カツ モクテキ エキメイ カ キヨウト デ アル
     ヨウナ
     ジョウシャ リョウキン ト
     シテン エキメイ ト
     モクテキ エキメイ ト
     コウツウ シュダン
     ヲ オシエテクダサイ
デスカ? カクニンシテ クダサイ
     A   ハイ ソウデス
     B   イイエ チカイマス
     C   モウ ヤメマス
? A

コタエ
     2. ケン カイトウ シマシタ
ヒョウジ シテホシイ ケンスウ ヲ キーイン スルカ ,     マタハ
シタノ A-D ヨリ エランデ キーイン シテクダサイ
     A   ゼンブ ヒョウジ スル
     B   ジョウケンチ ノ アタイ ヲ カエテ シツモン ヲシタイ
     C   ベツノ シツモン ヲ シタイ
     D   シュウリョウ シタイ
? A
     シテンーエキメイ ナゴヤ
     モクテキーエキメイ キヨウト
     コウツウーシユダン シンカンセン
     ジヨウシヤーリヨウキン 3600.

     シテンーエキメイ ナゴヤ
     モクテキーエキメイ キヨウト
     コウツウーシユダン バス
     ジヨウシヤーリヨウキン 1600.
ツギニ トウシマスカ ?
     A   ニホンゴ デ シツモン ヲ シタイ
     B   ケイシキゲンゴ デ シツモン ヲ シタイ
     C   シュウリョウ シタイ
? A

シツモン ハ ナーニ ?
$ KOKOKARA SANZENIN EWA NANIDE
  IKIMASUKA?

アナタノ シツモンハ
     シテン エキメイ カ キヨウト デ
     カツ モクテキ エキメイ カ サンゼンイン デ アル
     ヨウナ
     コウツウ シュダン ト
     シテン エキメイ ト
     モクテキ エキメイ
     ヲ オシエテクダサイ
デスカ? カクニンシテ クダサイ
     A   ハイ ソウデス
     B   イイエ チカイマス
     C   モウ ヤメマス
? A
コタエ
     1 ケン シカ アリマセン
     シテンーエキメイ キヨウト
     ノリカエーエキメイ オオハラ
     モクテキーエキメイ サンゼンイン
     コウツウーシユダンー1 バス
     コウツウーシユダンー2 トホ
ツギニ トウシマスカ ?
     A   ニホンゴ デ シツモン ヲ シタイ
     B   ケイシキゲンゴ デ シツモン ヲ シタイ
     C   シュウリョウ シタイ
? C
```

## 6. おわりに

日本語文によってデータベースへの問合せを行なうシステムの方式について述べた。そしてデータ属性と属性間の関係に注目した質問文意味解析および文脈解析のモデルと問合せ言語FQLにより与えられるデータベース検索機能とを提案した。我々は、実験システムJAMを開発し，モデルの検証を行なっているが、通常の質問文の実行時間はコメント出力の時間を除くと3～10秒である。

今後，JAMの利用，改良を通してより使い易い会話インタフェースを目指したいと考えている。

最後に，本研究の機会を与えられた当研究所コンピュータシステム研究部の三上部長，箱崎課長，助言を励ましを頂いた牧野主任，水摩主任，検討と開発に協力頂いた市山氏，JSA社島村氏にそれぞれ感謝します。

## 参考文献


1. 藤崎,ほか「データベース照会システム「ヤチマタ」と名詞句データ模型」信学誌D，Vol.20，No.1
2. 島津「日本語の構文・意味解析ー質問応答システムMSSS/Bでの試み」信学会 AL78-94
3. Codd, E. F., "DATABASES: Improving Usability and Responsiveness," PROC. of the August '78 International Conf. on Database.
4. Waltz, D. L., "An English Language Question Answering System for a Large Relational Data Base," CACM, Vol-21, pp.526-539, 1978.
5. Sacerdoti, E. D., et. al.," A LADDER user's guide", SRI Technical Note 163, 1978.
6. ACOS-6 データ管理「INQ概説書」
7. 田中,ほか「拡張LINGOLマニュアル」電総研，1978.
8. 村木,市山「日本語質問文解析におけるデータベースセマンティックスの利用」情処 自然言語処理 27-4，1981.
9. Chamberlin, D.D. and Boyce, R.F. "SEQUEL: A Structured English Query Language," Proc. ACM SIGFIDET workshop, May 1974, pp.249-264.


## 【付録】 FQLの構文形式

FQL ::= SELECT target
WHERE (predicate list)
[ORD (ordering key attr. list)]
[GRP (grouping key attr.)
HVG (group predicate)]

target ::= { YN | CNT (rel. name) | UCNT (attr. name) | DATA (target attr. list) }

target attr. list ::= attr. name [unit] [target attr. list]

predicate list ::= predicate [{AND | OR} predicate list]

predicate ::= { attr. name op {value [unit] | MIN | MAX} | attr. name 1 op attr. name 2 }

ordering key attr. list ::= attr. name [order key attr. list]

grouping key attr. ::= attr. name

group predicate ::= {AVG | SUM | CNT | UCNT} attr. name op value [unit]

op ::= {EQ | GE | LE | GT | LT | NE}