

SQL とその特殊機能の考察

——副照会と視野機能を中心として——

齋 藤 孝・三 和 義 秀

はじめに

ビジネスパソコンの本格的な普及にともなって、データベースソフトウェアは誰もが手軽に使える人気の高いツールとなっている。一頃、パソコンの言語といえば BASIC が一般的であったが最近ではリレーショナルデータベース言語が BASIC に代わる新世代の高級言語として地位をかためつつある。その理由は、言語の文法が簡単なことが挙げられるが、何よりもファイルを中心にデータが処理できるからであろう。ビジネス、研究の情報管理はデータのファイリングが不可欠な機能なのである。

現在、リレーショナルデータベース言語は多くの製品が世に出ている。これらを分類してみると大型コンピュータ用にもともとインプリメントされたものとパソコン専用に設計開発されていたものがある。先に、後者を見るとパソコンの大衆化に合わせて使い勝手がいいのが特色である。しかし、リレーショナルデータベース本来の理論と仕様を忠実に実現されていないことが欠点である。その点前者は、設計の基礎がしっかりしていてリレーショナルデータベースの教育には最適である。

さて本稿では、リレーショナルデータベースの理論と仕様条件を満足しているとされる SQL を取り挙げ、実際にこれを使用することによって一部機能を評価する。なお、評価の素材は文献¹⁾に基づいている。

1. リレーショナルデータベースとは

リレーショナルデータベース (Relational Data Base) のリレーショナルとは「Relation」、つまり「関係」と言う意味であるが、ここで言う「関係」とは散在しているさまざまな情報の項目が何らかの関係に基づいて集合体を形成するという意味である。各集合体に存在する項目を関係づけながらテーブル (関係表) に整理することがリレーショナルデータベースの機能である。

もともと「関係」という概念は、Relational Algebra (関係代数) に理論的根拠をおく。²⁾

ただし、この理論を理解しなければリレーショナルデータベースが使えないわけではない。むしろ、理論にこだわらなければこれほど単純なデータベースのモデルは他に存在しない。ちなみにデータベースのモデルには、関係型モデル以外に階層（木型）モデル³⁾やネットワーク（網型）モデル⁴⁾がよく知られている。

関係型モデルのデータ構造は、図1に示すような行列から構成する2次元の表（テーブル）である。列はデータ項目（フィールド）に相当し、列の集まりが行（レコード）になる。

利用者は、これらの項目の整理されたテーブルに対してさまざまな問い合わせを行ない、要求する情報を抽出する。これは「リレーション（関係）の操作」と言われ、このリレーションの操作としては次の3つが挙げられる。

- ①レコードの検索……テーブルから行（レコード）を抽出する。
- ②項目の検索……テーブルの行（項目）を抽出する。
- ③テーブルの結合……2つ以上のテーブルを結合する。

このように各リレーションを関係づけながらさまざまな情報を処理するシステムがリレーショナルデータベースである。

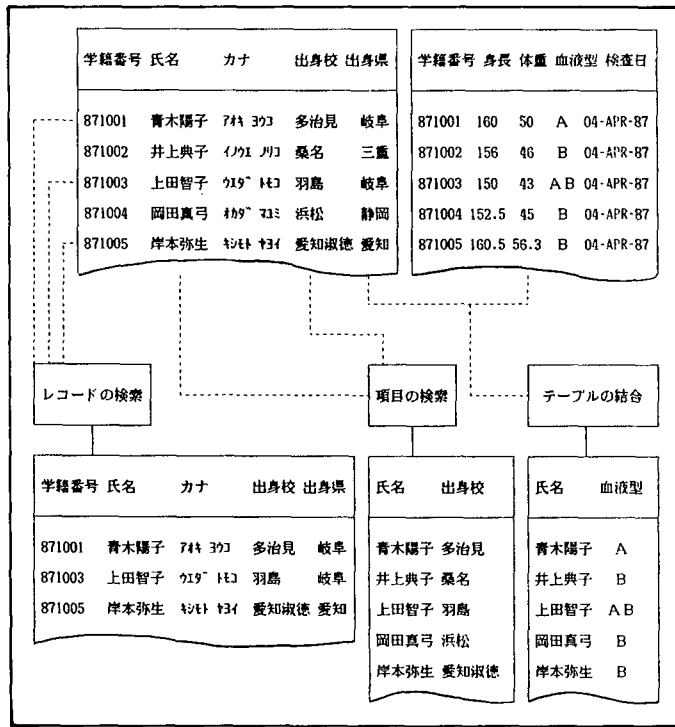


図1 関係型モデルのデータ構造

2. SQL とは

リレーショナルデータベースを実際のソフトウェアとしてインプリメントする試みは、1975

年代から始まったとされる。今日まで試作システムや商用化システムとして数多くのリレーショナルデータベース管理(RDBM)システムが誕生した。このあたりで RDBM システムの標準化の動きが当然のこととしてある。現在のところ、この標準化システムとしては、米国の ANSI 規格で制定された SQL (Structured Query Language)が知られている。⁵⁾

SQL は、リレーショナルデータベースを専門家でないエンドユーザに利用させるために、IBM 社のサンノゼ研究所で開発された言語である。今後、SQL は、RDBM システムの標準言語となり、超大型コンピュータでもパソコンでもインプリメントされて、RDBM システムの共通の言語として発展していくものと思われる。

SQL の基本的な構文は、次の3つのブロックから成り立っている。

- ① SELECT …… 選択する項目の指定。
- ② FROM …… 対象とするテーブルの指定。
- ③ WHERE …… 検索条件の指定。

例えば、基本というテーブルを対象として『出身校が“愛知淑徳”である学生の学籍番号、氏名、出身県を知りたい。』という問い合わせは、SQL では次のように書くことができる。

```
SELECT 学籍番号, 氏名, 出身県
FROM 基本
WHERE 出身校 = '愛知淑徳';
```

見ての通りきわめてシンプルな構文であるにもかかわらず、強力な機能を記述できるのである。ここで SQL の言語体系を示しておく。

SQL は、図2に示すようにデータベース定義言語(Data Description Language)とデータベース操作言語(Data Manipulation Language)に大別することができる。前者は、テーブル・索引・視野の作成や消去などで、後者は、データの挿入・検索・削除・更新などである。

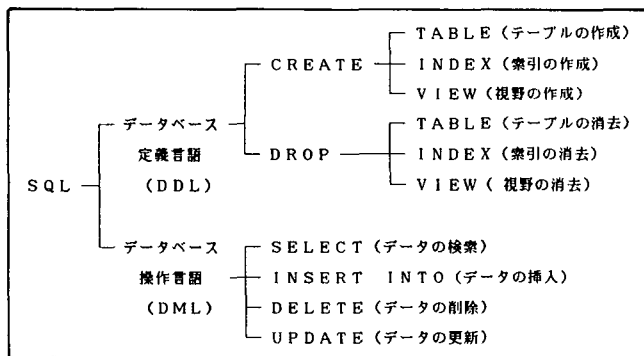


図2 SQL の言語体系

3. ORACLE

3-1. ORACLE の概要

本学の持っている SQL 仕様の RDBM システムは, ORACLE と呼ぶ。⁶⁾

ORACLE は, 米国の Oracle Crop. 社が開発した UNIX の下で稼働する RDBM システムである。ORACLE の言語は完全な SQL 仕様に準拠している。本論ではこの ORACLE を使用して, もっとも SQL らしく, またその名が示す構造化された問い合わせ (Structured Query) 機能について評価した。

さて, 構造をもつ問い合わせとは, どのようなものであるかを紹介しておこう。

いま, 例題として図 3 に示す学生情報管理のための基本テーブルと保健テーブルを考える。

基本テーブルには, 学籍番号, 氏名, カナ, 出身校, 出身県のデータが記録されている。また, 保健テーブルには, 学籍番号, 身長, 体重, 血液型が記録されている。この 2 つのテーブルを対象に問い合わせを行なうことにする。

学籍番号	氏名	カナ	出身校	出身県	学籍番号	身長	体重	血液型	検査日
871001	青木陽子	アオキ ヨウコ	多治見	岐阜	871001	160	50	A	04-APR-87
871002	井上典子	イノウエ ノリコ	桑名	三重	871002	156	46	B	04-APR-87
871003	上田智子	ウエダ トモコ	羽島	岐阜	871003	150	43	AB	04-APR-87
871004	岡田真弓	オカダ マユミ	浜松	静岡	871004	152.5	45	B	04-APR-87
871005	岸本弥生	キシモト ヤヨイ	愛知淑徳	愛知	871005	160.5	56.3	B	04-APR-87
871006	近藤敏子	コンドウ トシコ	桑名	三重	871006	160	49	A	04-APR-87
871007	酒田裕子	サカタ ユウコ	多治見	岐阜	871007	156	46	AB	04-APR-87
871008	清水陽子	シミズ ヨウコ	愛知淑徳	三重	871008	156	42	B	04-APR-87
871009	中島敏子	ナカシマ トシコ	愛知淑徳	愛知	871009	146	40	A	04-APR-87
871010	西村里美	ニシムラ サトミ	桑名	三重	871010	156.3	46.5	AB	04-APR-87
871011	浜田幹子	ハマダ ミキコ	一宮	愛知	871011	147	46.8	B	04-APR-87
871012	福田静代	フクタ シズヨ	富山西	富山	871012	156	46.2	AB	04-APR-87
871013	本田和代	ホンダ カズヨ	員弁	三重	871013	170	63	B	04-APR-87
871014	松田康子	マツダ ヤスコ	愛知淑徳	滋賀	871014	158	46	B	08-APR-87
871015	三輪節子	ミワ セツコ	金沢南	石川	871015	150	40	B	04-APR-87

基本テーブル

保健テーブル

図 3 基本テーブルと保健テーブル

3-2 副照会

ORACLE では、問い合わせを主照会と呼び、その中に構造化されたいくつかの問い合わせ文を組み込める。それを副照会(Subquery)機能と呼ぶ。

では、副照会の概念について例題を通して説明していこう。

3-2-1 1つの値を返す副照会

例えば次の質問があったとする。

『“青木陽子”と同じ出身校である学生を見つけよ。』

一見、何げない質問であるがこれを問い合わせ文に分解すると次のようになる。
なお、? 記号は不明の値である。

基本テーブル 氏名=“青木陽子” → 出身校=? → 氏名=?

まず、氏名の“青木陽子”は明らかとなっている。ところが彼女の出身校がどこであるかは検索結果の答えを見なければわからない。

したがって、問い合わせ文は次の①と②の2つに分けて行なう必要がある。

- ① “青木陽子”の出身校がどこであるかを問い合わせる。

```
UF1> SELECT 学籍番号, 氏名, 出身校
      2 FROM 基本
      3 WHERE 氏名='青木陽子';
```

学籍番号	氏名	出身校
871001	青木陽子	多治見

- ② ①の結果より出身校が“多治見”である学生を問い合わせる。

```
UF1> SELECT 学籍番号, 氏名, 出身校
      2 FROM 基本
      3 WHERE 出身校='多治見';
```

学籍番号	氏名	出身校
871001	青木陽子	多治見
871007	酒田裕子	多治見

ここに ORACLE の副照会の機能を使うと1つの問い合わせ文として次のように記述することができる。この問い合わせ文は、先の①と②の2つの文を1つにしたものとなり、図4のように主照会を行なうために不明の値を求める副照会を含む文といえる。なお、副照会は、その中にまた副照会を含んでもよい。

```

UF1> SELECT 学籍番号, 氏名, 出身校
2 FROM 基本
3 WHERE 出身校=
4     (SELECT 出身校
5      FROM 基本
6      WHERE 氏名='青木陽子');
    
```

} 主照会
} 副照会

学籍番号	氏名	出身校
871001	青木陽子	多治見
871007	酒田裕子	多治見

この文は SELECT 文の中に SELECT 文を入れ子として含む構造になる。

このように ORACLE は複雑な質問をきわめてシンプルな SQL 文法によって表現できる。

もし、この副照会の機能がないとしたら、このような質問は数回に分けて、中間結果を保存しておきそれぞれバッチ処理的に試みなければならない。

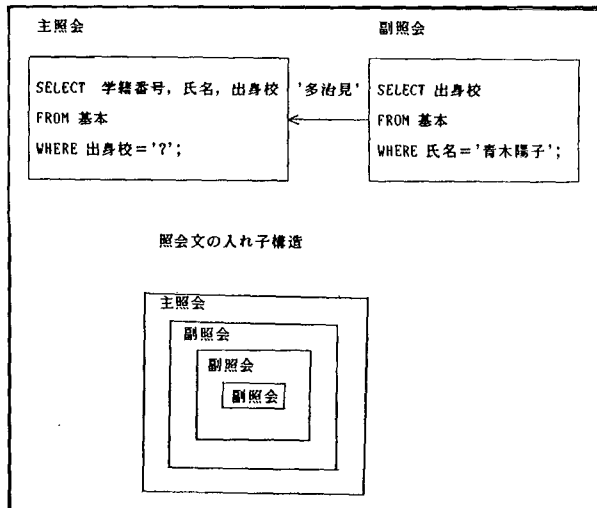
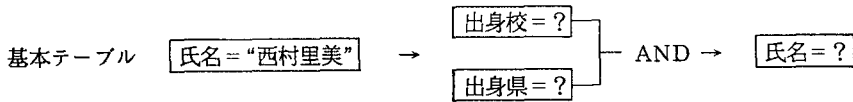


図4 主照会と副照会の構造

3-2-2 2つの項目の値をを返す副照会

次の例は不明なデータ項目として出身校と出身県の2つをもっている。そして検索の結果、不明データが明らかにされると論理積 (AND) によって条件づけをして氏名を求めるものである。

『“西村里美”と同じ出身校であり、かつ出身県も同じである学生を見つけよ。』



この問い合わせをまず単純な照会で考えると次のようになる。

- ① “西村里美” の出身校と出身県がどこであるか問い合わせる。

```
UF1> SELECT 学籍番号, 氏名, 出身校, 出身県
2 FROM 基本
3 WHERE 氏名='西村里美';
```

学籍番号	氏名	出身校	出身県
871010	西村里美	桑名	三重

- ② ①の結果より、出身校が“桑名”であり、かつ出身県が“三重”である学生を問い合わせる。

```
UF1> SELECT 学籍番号, 氏名, 出身校, 出身県
2 FROM 基本
3 WHERE 出身校='桑名'
4 AND 出身県='三重';
```

学籍番号	氏名	出身校	出身県
871002	井上典子	桑名	三重
871006	近藤敏子	桑名	三重
871010	西村里美	桑名	三重

ORACLE では、副照会の中で返される項目の値は2つ以上あってもかまわない。この問い合わせは副照会を使うと次のようになる。

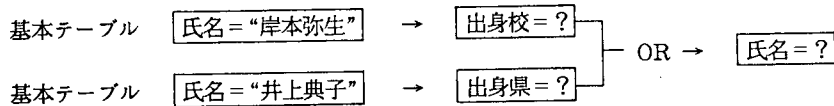
```
UF1> SELECT 学籍番号, 氏名, 出身校, 出身県 } 主照会
2 FROM 基本
3 WHERE (出身校, 出身県)=
4       (SELECT 出身校, 出身県 } 副照会
5         FROM 基本
6         WHERE 氏名='西村里美');
```

学籍番号	氏名	出身校	出身県
871002	井上典子	桑名	三重
871006	近藤敏子	桑名	三重
871010	西村里美	桑名	三重

3-2-3 複数の照会を使った複合照会

次の例は最初から明らかな氏名が2名あり、それぞれ出身校と出身県の不明データを別々に求める。そして論理和 (OR) によって条件づけるものである。

『“岸本弥生”の出身校と同じであるか、あるいは“井上典子”の出身県と同じである学生を見つけよ。』



この問い合わせをまず単純な照会で考えると次のようになる。

- ① “岸本弥生”の出身校を問い合わせる。

```
UF1> SELECT 学籍番号, 氏名, 出身校
2 FROM 基本
3 WHERE 氏名='岸本弥生';
```

学籍番号	氏名	出身校
871005	岸本弥生	愛知淑徳

- ② “井上典子”の出身県を問い合わせる。

```
UF1> SELECT 学籍番号, 氏名, 出身県
2 FROM 基本
3 WHERE 氏名='井上典子';
```

学籍番号	氏名	出身県
871002	井上典子	三重

- ③ ①と②の結果より、出身校が“愛知淑徳”であるか、あるいは出身県が“三重”である学生を問い合わせる。

```
UF1> SELECT 学籍番号, 氏名, 出身校, 出身県
2 FROM 基本
3 WHERE 出身校='愛知淑徳'
4 OR 出身県='三重';
```

学籍番号	氏名	出身校	出身県
871002	井上典子	桑名	三重
871005	岸本弥生	愛知淑徳	愛知
871006	近藤敏子	桑名	三重
871008	清水陽子	愛知淑徳	三重
871009	中島敏子	愛知淑徳	愛知
871010	西村里美	桑名	三重
871013	本田和代	員弁	三重
871014	松田康子	愛知淑徳	滋賀

8 records selected.

このように3つの問い合わせによって結果を得ることができるが、副照会を使用すると次の1つの問い合わせ文で書くことができる。

```

UFI> SELECT 学籍番号,氏名,出身校,出身県
2 FROM 基本
3 WHERE 出身校=
4     (SELECT 出身校
5      FROM 基本
6      WHERE 氏名='岸本弥生')
7 OR 出身県=
8     (SELECT 出身県
9      FROM 基本
10     WHERE 氏名='井上典子');
    
```

} 主照会
 } 副照会
 } 副照会

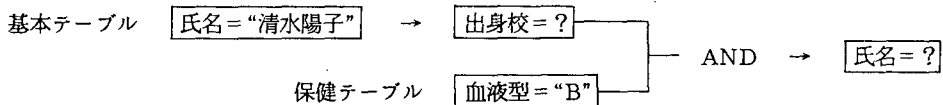
学籍番号	氏名	出身校	出身県
871002	井上典子	桑名	三重
871005	岸本弥生	愛知淑徳	愛知
871006	近藤敏子	桑名	三重
871008	清水陽子	愛知淑徳	三重
871009	中島敏子	愛知淑徳	愛知
871010	西村里美	桑名	三重
871013	本田和代	員弁	三重
871014	松田康子	愛知淑徳	滋賀

8 records selected.

3-2-4 結合条件と副照会の併用

次の例は、違った2つのファイル（テーブル）を検索の対象にするものである。テーブルは“基本”と“保健”で2つのテーブルを結合（JOIN）する操作を含めている。

『“清水陽子”と同じ出身校であり、かつ血液型が“B”である学生を見つけよ。』



この問い合わせをまず単純な照会で考えると次のようになる。

① “清水陽子”の出身校がどこであるか問い合わせる。

このためには、テーブル“基本”を対象にして次のような問い合わせが記述できる。

```

UFI> SELECT 学籍番号,氏名,出身校
2 FROM 基本
3 WHERE 氏名='清水陽子'
    
```

学籍番号	氏名	出身校
871008	清水陽子	愛知淑徳

- ② 基本、保健の両テーブルを対象として出身校が“愛知淑徳”であり、かつ血液型が“B”である学生を問い合わせる。この場合、両テーブルに共通して存在する“学籍番号”が結合条件の「キー」となる。

```

UF1> SELECT 基本,学籍番号,氏名,出身校,保健,血液型
2 FROM 基本,保健
3 WHERE 出身校='愛知淑徳'
4 AND 血液型='B'
5 AND 基本,学籍番号=保健,学籍番号;

```

学籍番号	氏名	出身校	血液型
871005	岸本弥生	愛知淑徳	B
871008	清水陽子	愛知淑徳	B
871014	松田康子	愛知淑徳	B

ORACLE では照会の対象となるデータベースファイルが2つ以上となる場合は、結合条件と副照会を同時に併用することができる。この問い合わせは結合条件と副照会を併用すると次のようになる。

```

UF1> SELECT 基本,学籍番号,氏名,出身校,保健,血液型
2 FROM 基本,保健
3 WHERE 血液型='B'
4 AND 基本,学籍番号=保健,学籍番号
5 AND 出身校 IN
6   (SELECT 出身校
7    FROM 基本
8    WHERE 氏名='清水陽子');

```

} 主照会
} 副照会

学籍番号	氏名	出身校	血液型
871005	岸本弥生	愛知淑徳	B
871008	清水陽子	愛知淑徳	B
871014	松田康子	愛知淑徳	B

3-3 視 野 (VIEW)

ここではSQLの強力な機能である視野 (VIEW) について紹介しよう。

共有データが蓄積されているデータベースファイル (テーブル) に対する問い合わせは、質問の利用者によってさまざまな観点から試みられる。利用者の使い勝手から考えると、自分専用の情報だけが蓄積されていて、そのデータ構造だけが見えると便利である。つまり、利用者

の個人用の専用テーブルが提供されていれば都合がよいわけである。このように考えると利用者の好みのフィールドを共有のテーブルから抽出してパーソナルユースのテーブルをいくつも定義することができる機能が必要になる。ただし、ディスク容量には限界があるし、システムから見ても多数のテーブルが生成されては管理がむずかしいし、またシステムのパフォーマンスを著しく低下させ不経済でもある。そこで考案された概念は、仮想 (Virtual) テーブルを生成することである。仮想テーブルとは、ディスク上には実在しないテーブルのことである。したがって実際にディスクメモリをテーブルのために割当ててくることもない。すなわちデータベースファイル全体のディスク容量を不変にしておき、あたかもいくつもの利用者の問い合わせニーズに合わせたテーブルが実在するかのように見せる機能といえる (図5)。SQL ではこの機能を視野 (VIEW) と呼ぶ。

いま、基本テーブルと保健テーブルにおいて『全学生の学籍番号、氏名、血液型を知りたい』という問い合わせが頻繁に発生するものとする。この場合、その都度、結合条件を使い複雑なプログラムを作らなければならない。しかし、この条件にあった視野を一度作成しておけば、問い合わせが非常に容易になる。この条件の視野 (基本2) を作成するプログラムは次のようになる。コマンドは「CREATE VIEW」である。

```

UF1> CREATE VIEW 基本2 (学籍番号, 氏名, 血液型) AS } 仮想テーブルの定義
2 SELECT 基本, 学籍番号, 氏名, 保健, 血液型
3 FROM 基本, 保健 } 検索条件の指定
4 WHERE 基本.学籍番号=保健.学籍番号;

View created.

```

また、作成した視野は、次のような単純なプログラムで何度でも見ることができる。なお、*は、テーブルに存在するすべてのフィールドを出力させる。

```
UF1> SELECT * FROM 基本2;
```

学籍番号	氏名	血液型
871001	青木陽子	A
871002	井上典子	B
871003	上田智子	AB
871004	岡田真弓	B
871005	岸本弥生	B
871006	近藤敏子	A
871007	酒田裕子	AB
871008	清水陽子	B
871009	中島敏子	A
871010	西村里美	AB
871011	浜田幹子	B
871012	福田静代	AB
871013	本田和代	B
871014	松田康子	B
871015	三輪節子	B

```
15 records selected.
```

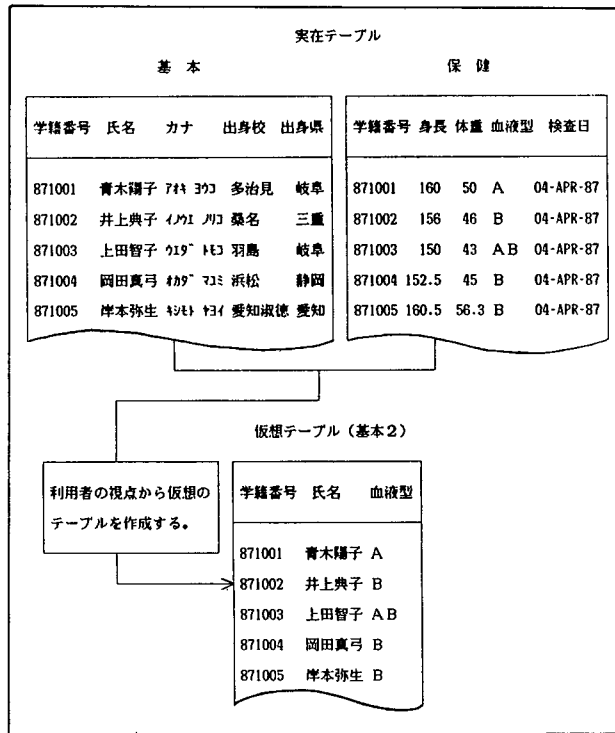


図5 仮想テーブル

おわりに

ORACLE を実際に使用して SQL の文字どおり構造化された問い合わせ機能（主照会と副照会）を評価した。

SQL は「SELECT, FROM, WHERE」というわずか3語のコマンドとシンプルな文法だけしか規定されていないにもかかわらず、きわめて多種多様な問い合わせ文に対処できる強力な言語であることを確認した。また、視野（VIEW）機能は、エンドユーザの使い勝手を十分に考慮しており、用途も非常に実用的である。

SQL は1987年秋に、JIS 版の標準仕様書の提案があると聞く。今後、日本を含め全世界で共通に使える初めてのデータベース言語となるわけである。このことは SQL をデータベース言語教育に採用している本大学にとって誠に喜ばしい。

参 考 文 献

- (1) 斎藤 孝 パソコン RDBMS 比較言語論 CQ 出版社 (1987)
- (2) Codd, E. F. "A Relational Model of Data for Large Shared Data Banks"
Comm. ACM, 13(6) P377-383. (1970)
- (3) McGee, W. C. "The IMB / VS System" IBM Systems Journal 16(2) P84-168.
(1977)
- (4) Taylor, R. W. "CODASYL Data-Base Management Systems" ACM Computing
Surveys 8(1) P67-104 (1976)
- (5) Chamberlin, D. D. "Relational Data-Base Management Systems" ACM Computing
surveys 8(1) P43-66 (1976)
- (6) DCL ORACLE オペレーションマニュアル (1984)

A study of the typical functions of SQL

Takashi Saito and Yoshihide Miwa

Information Science Education Center
Aichi Shukutoku College

Abstract

This paper describes the typical functions of SQL (Structured Query Language) which is ANSI standard language for relational database management system implemented on the mini-computer in Information Science Education Center of Aichi Shukutoku College.

The following two functions of SQL are evaluated.

1. The subquery function which provides simple query description and defines the combination of several queries as one structured query.
2. The view function which creates personal user oriented files based on the technique of virtual storage memory.

SQL is a language which provides a sophisticated syntax and a powerful query formula using only three commands such as SELECT, FROM, WHERE.