

電子出版指向データベースシステム

4Q-2

— SQL 言語のマルチメディア拡張 —

田中 裕彦¹⁾ 橋野 勝久¹⁾ 杉田 卓也¹⁾ 中埜 善夫¹⁾ 陳 惠榮²⁾

松下電器産業(株) 情報システム研究所¹⁾ 松下電器技術開発(株) 台北技術研究所²⁾

1. はじめに

光ディスク等の大容量デバイスの普及に伴い、複数のデバイスに文字・図形・画像等のマルチオブジェクトを分散して格納し、それらを相互利用できるデータベースシステムの需要が高まってきている。またその際、標準的なインタフェースで汎用的なデータ表現にアクセスすることが必要である⁽¹⁾。

本報告では、JIS規格⁽²⁾のSQL言語をもとにしてマルチメディア対応に拡張したmSQLについて、拡張部分に絞って述べる。

2. 基本概念

mSQLでは、マルチデバイスに対応するためにスキーマ概念を拡張し、またマルチオブジェクトに対応するためにデータ型を拡張した。(図1)

①スキーマ概念の拡張

まず、環境を以下のように定義する。

「データベース使用時に指定されたすべてのデバイス内のすべてのスキーマからなる集合」

スキーマはスキーマ定義言語によって指定される永

続対象であり、テーブルとオブジェクトの集合からなる。スキーマはスキーマ名で識別される。1つのスキーマは必ず1つのデバイスに所属しなければならない。従って、1つのスキーマ内のテーブルとオブジェクトを複数のデバイスに分散して格納することはできない。

しかしながら、異なるスキーマ内のテーブルやオブジェクトを、データ操作言語を用いて同時に操作することができる。また、異なるスキーマ内のテーブル間で関係演算を行うことを可能にした。

②データ型の拡張

列定義でポインター型を追加した。

ポインター型のデータは、オブジェクト名とオブジェクトを持つ。テーブルの列内には論理的にはオブジェクト名のみが存在し、文字列型のデータと同様の演算・検索が可能である。オブジェクトは、文字・図形・画像等の不定長データに関する情報を保有する。

オブジェクト名はオブジェクトを識別する名前であり、各列ごとに一意性が保たれる。すなわち、同一列内で同じオブジェクト名が存在するときにはそれらは同じオブジェクトを指すが、異なる列内に同じオブジ

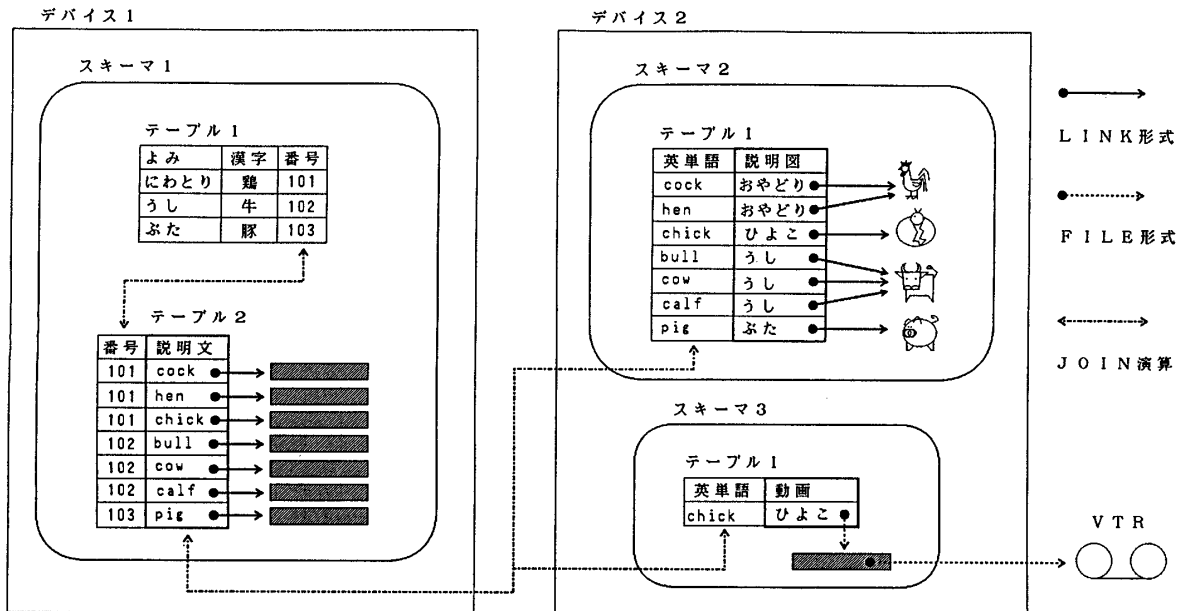


図1 スキーマ概念とデータ型の拡張

A Database Management System for Electronic Publishings - Extended SQL for Multi-Media Data Management - Hirohiko TANAKA¹, Katsuhisa HASHINO¹, Takuya SUGITA¹, Yoshio NAKANO¹, Hui-Yeen CHEN²

1. MATSUSHITA Electric Ind. Co., Ltd. 2. MATSUSHITA Electric Institute of Technology(Taipei) Co., Ltd.

エクト名があってもそれらは同一のオブジェクトを指さない。従って、同一列内においてのみ、オブジェクト名とオブジェクト間で多対1の関係が持てる。

オブジェクトは次の4種類の内容を持つ。

1. オブジェクトデータ・文字、図形、画像などの不定長データ
2. データ長・・・オブジェクトデータの長さ
3. データ種類・・・オブジェクトデータの種類
4. 格納形態・・・LINK形式またはFILE形式

オブジェクトデータの意味内容については、MSQLでは関知しない。すなわちオブジェクト名は文字列型のデータと同様に比較などの演算の対象となるが、オブジェクトは演算の対象にならない。

3. スキーマ定義言語

スキーマとテーブルの定義を行う。

① スキーマ定義

スキーマ名と、スキーマを作成するディレクトリパス名を指定する。スキーマ名は、同じ環境下にある他のどのスキーマ名とも異ならなければならない。

```
<スキーマ定義> ::=
CREATE SCHEMA <スキーマ名>
[INTO <ディレクトリパス名>]
```

② テーブル定義

テーブルをスキーマ内で一意に識別するための表識別子、テーブルを作成するスキーマのスキーマ名、及び複数の列定義を指定する。テーブルは必ずどれか1つのスキーマに属する。ポインター型の列定義で指定する長さは、ポインター型のデータを格納する列のオブジェクト名の長さを表す。

```
<テーブル定義> ::=
CREATE TABLE <表識別子>
IN <スキーマ名>
(<列定義>[ { , <列定義> } ... ])
<ポインター型の列定義> ::=
POINTER [ (<長さ> )]
```

4. データ操作言語

データの検索及び挿入・更新・削除を行う。以下、データの検索及び更新について述べる。

① データ検索

FROMで指定する表名にスキーマ名と表識別子を用いることにより、1つのSELECT文で同時に複数のスキーマ内のテーブルを検索することができる。

ポインター型のデータを探索条件で指定したときには、オブジェクト名を文字列として探索する。

ポインター型のデータの取出し方は次の3種類ある。

1. 指定無し---オブジェクト名
2. /FULLNAME---オブジェクト名にそのオブジェクトが

属するスキーマ名・表識別子・列名を付与した名前

3. /CONTENT---オブジェクトデータ

FULLNAMEを用いると、ユーザープログラムが環境中でのそのオブジェクトを一意に識別することができる。

```
<SELECT文> ::=
SELECT [ /CONTENT | /FULLNAME ]
[ DISTINCT ] <列指定> [ { , <列指定> } ... ] | *
INTO <相手指定> [ { , <相手指定> } ... ]
FROM <表名>
[ WHERE <探索条件> ]
<表名> ::= [ <スキーマ名> . ] <表識別子>
```

② データ更新

ポインター型のデータの更新時には、オブジェクト名とオブジェクトを指定する。

UPDATE文で指定したオブジェクト名と同じオブジェクト名がすでに同一列内に存在する時には、すべての同じオブジェクト名をもつポインター型のデータをUPDATE文で指定したオブジェクトに更新することにより、オブジェクト名による一意性を保つ。

更新の結果、更新前のオブジェクト名が同一列内に1つも存在しなくなった時には、そのオブジェクト名が指しているオブジェクトを削除する。

```
<UPDATE文:探索> ::=
UPDATE <表名>
SET <設定句> [ { , <設定句> } ... ]
WHERE <探索条件>
<設定句> ::=
<列名> = { <値式> | NULL | <ポインター型データ> }
<ポインター型データ> ::=
<オブジェクト名> [ <オブジェクト> ]
<オブジェクト> ::=
AS <格納形態>
KIND <データ種類>
LENGTH <データ長>
OBJECT <オブジェクトデータ>
<格納形態> ::= { 'LINK' | 'FILE' }
<データ種類> ::= { 'TXT' | 'PDL' | ... }
```

5. おわりに

現在のところ、MSQLはユーザープログラム中で埋込み文として指定する。今後はインタプリタや対話メニュー等を拡充して、マルチメディアデータに適したユーザーインタフェースを構築してゆきたい。あわせて、SQL2への対応も考えてゆきたい。

参考文献

- (1) 橋野他：電子出版指向データベースシステム－基本構想－，情処，第37回(63後)全国大会。
- (2) 日本規格協会：データベース言語SQL，JIS X3005-1987。