

SQLによるデータの意味のサポート

遠山 元道 慶応義塾大学

村田 達彦 NTT(株)情報研

藤井 宏公 (株)リコー

SQLによるデータの意味のサポートについて、最近提案された部分を中心に述べる。解説の対象は、定義域(DOMAIN)、副定義域(SUBDOMAIN)、副表(SUBTABLE)である。このうち副定義域を除いてはISOによって、SQL3への導入がすでに確定している。

SQL ENHANCEMENTS ON DATA SEMANTICS

Motomichi Toyama (Keio University)

Tatsuhiko Murata (NTT Corp.)

Hirokimi Fujii (Ricoh Corp.)

Recent developments on SQL support for data semantics are reported. Described materials are domain, subdomain and subtable. The inclusion of domain and subtable in SQL draft standard is approved by ISO.

1. はじめに

従来のSQLは、データの意味（セマンティクス）や意味制約を表現する手段をほとんど持たず、主キー制約の表現さえも十分に行なえなかった。これに対し、情報システムが高度で大規模になるほど、データの保全性を高めるため、また応用プログラム開発の規範とするため、データの意味の表現の高度化が求められることになる。従来のSQLがこうした要求に応える機能を持たないことを、日本の国内委員会は特に重要視し、SQLによるデータの意味のサポートを強化する提案を推進した。本稿では、これらのうち定義域（DOMAIN）、副定義域（SUBDOMAIN）、および副表（SUBTABLE）について報告する。これらはいずれも、対象集合の間の汎化と専化を表現するための機能である。

定義域については、一部がSQL2、一部がSQL3に採用されることが決まっている。副定義域は、SQL3への採用提案を準備中である。副表はSQL3に含まれる。以下では、SQL3をSQL2と区別せず、単にSQLとして扱っている。

2. DOMAINおよびSUBDOMAIN

2.1 概要

定義域（DOMAIN）は、SQLにおいて、値の集合に関する制約を表から独立に定義するための機能である。副定義域（SUBDOMAIN）は、定義域の部分集合を階層化するためのものである。主な機能を以下に記す。

- (1) 定義域（domain）には、唯一のデータ型が対応する。
- (2) 定義域には、DEFAULT句によって、省略時既定値を指定することができる。
- (3) 定義域には、検査制約定義（check constraint definition）によって、とり得る値の制約を加えることができる。
- (4) 定義域には、空値指定によって、空値として扱われる値を指定することができる。
- (5) 副定義域は、既定義の定義域または副定義域を基に定義する。基となる定義域または副定義域をgendomain、定義される副定義域をsubdomainと呼ぶ。これらの用語は相対的な呼称である。
- (6) subdomainはgendomainの制約を継承し、さらに独自にDEFAULT句、空値指定、検査制約定義を持つことができる。
- (7) subdomainにDEFAULT句が指定された場合、これが継承された省略時既定値に優先する。
- (8) subdomainに空値指定があると、gendomainの空値に加えて、ここで指定された値も空値として扱われる。
- (9) subdomainに検査制約定義が与えられた場合、gendomainの有効な検査制約と、この検査制約の連言が、subdomainの有効な検査制約となる。

2.2 概念

- (1) 表に現れる値に制約を加える方法として、表定義中の列定義に、空値指定、省略時既定値指定、検査制約定義などを与えることができる。しかし、これらはその列が属する表の内部でのみ有効である。定義域は、値に関する

これらの制約を表から独立に行なうための、データベースの構成要素である。表の列定義に、制約そのものを書く代わりに定義域名を指定することによって、制約を直接指定するのと同じ効果が得られる。定義域の定義により、複数の表の列が同様の制約に従うとき、これらの記述を一括することができる。

- (2) `subdomain`は`gendomain`の部分集合を規定することを目的としている。これは、値に関する専化(*specialization*)に相当する。より狭い範囲の値には、より強い制約を課することができる。たとえば、ある会社の社員番号はアルファベット1字ではじまるが、特定の事業部の社員の社員番号は、特定のアルファベットではじまる、などである。この場合、全社の社員番号の定義域を`gendomain`、各事業部の社員番号の定義域を`subdomain`と定義することができる。

2.3 定義

(`sub`) `domain`の定義の構文を以下に示す。

```
<domain definition> ::=
CREATE DOMAIN <domain name>
{ [ [IS] <data type>
  [ <default clause> ]
  [ <domain constraint> ]
  [ <null definition> ] ]
| [ ISA <subdomain clause>
  [ <default clause> ]
  [ <domain constraint> ]
  [ <null definition> ] ] ] }
```

```
<domain constraint> ::=
<check constraint definition>
<constraint name definition>
```

```
<subdomain clause> ::=
<domain name>
```

2.4 操作機能

(1) SELECT

WHERE句において、異なる定義域の列どうしの比較は許されない。ただし、一方が他方の直接もしくは間接の`gendomain`である場合には比較が許される。

(2) INSERT

値の指定されなかった値については、対応する`domain`の省略時既定値が用いられる。

3. SUBTABLE

3. 1 SUBTABLEの概要

SQLに導入したSUBTABLE機能とは、いくつかの対象をまとめて1つの対象として扱う汎化機能 (generalization)、およびある対象をさらに細分化して別の対象に分解して扱う専化機能 (specialization) のことである。汎化/専化の例を図3-1に示す。

SQLへの導入は日本が主導的に検討を行い、現在SQL3のワーキング・ドキュメントに反映されている。そこでの主な機能を以下に示す。

- (1) 専化したテーブルを `subtable`、汎化したテーブルを `gentable` という名前で導入した。
- (2) `subtable` のテーブル定義時に `gentable` となる `base table` を指定する、トップダウン的な定義機能を備えている。
- (3) 複数の `gentable` から `subtable` を定義できる。即ち、`gentable-subtable` 間に、ネットワーク的な関係を定義できる。
- (4) `subtable` には `gentable` の定義内容 (全てのカラム定義、テーブル要素) が遺伝する。
- (5) 同一の対象を表す `gentable` と `subtable` の行は、`primary key` 値の一致により識別する。
- (6) 操作ステートメント (SELECT, INSERT, MODIFY, DELETE) ごとに、`gentable/subtable` それぞれ特有の機能を備えている。

3. 2 開発経過

- ・IRDSをSQLで記述する際に、テーブルの汎化機能、専化機能の必要性が生じ、国内委員会にてSQLへの導入の検討を開始した。
- ・1987年7月の東京会議に初めて提案。しかし操作機能が不十分であったため不採用となった。
- ・国内委員会での検討を重ね、定義機能とともに操作機能も完備して1988年2月のワシントン会議に提案し、SQL3機能ではあるものの全面的に導入が決定された。
- ・ワシントン会議で各国より指摘された点に応えるため、1988年11月のシドニー会議で拡張提案し採用された。

3. 3 機能概要

以下に、SQLに導入したSUBTABLE機能の概略を示す。

3. 3. 1 概念

(1) `subtable` と `gentable`

- ① `base table` を専化したテーブルを「`subtable`」、`subtable` によって専化された `base table` を「`gentable`」と呼ぶ。`gentable` も `subtable` も `base table` である。

- ② `gentable`と`subtable`は、`n:m`の対応が可能。ただし最上位の`gentable`は1つのみ。
- ③ `subtable`には`gentable`の定義内容（全てのカラム定義、テーブル要素）が全て遺伝する。
- ④ `subtable`のカラムのうち、`gentable`から遺伝したカラムを「遺伝カラム」、`subtable`独自に定義したカラムを「独自カラム」と呼ぶ。

(2) 行の対応

- ① `gentable`と`subtable`は同じ行を持つ。同一の対象を表す`gentable`と`subtable`の行は、`primary key`値の一致により識別する。
- ② ある`subtable`の行は必ず各`gentable`の1つの行と対応する。一方、`gentable`のある行は、対応する`subtable`の行を持たないこともある。
- ③ `gentable`の行と対応する`subtable`の行の遺伝カラムの値は、同じ値を持つ。

3. 3. 2 定義機能

(A) 定義

定義構文を図3-2に示す。

- (1) `<subtable句>`を指定することにより、そのテーブルが`<gentable name>`の`subtable`となる。
- (2) `<gentable name>`を複数指定することにより、複数の`gentable`を持つ`subtable`を定義できる。ただし、最上位の`gentable`は1個に制限する。
- (3) `<subtable句>`に続く`<table element>`で、`subtable`の独自カラムを指定する。
- (4) `AS`句を指定することにより、遺伝カラムのカラム名を変更できる。即ち、独自カラムおよび遺伝カラムのカラム名は`subtable`内で一意であればよい。（理由：現実の問題として名前を変えたいというニーズはある。（例えば、`gentable`では学生番号、`subtable`では男子学生番号）また、そもそもカラム名はテーブル内で一意であればよい。）
- (5) 最上位`gentable`には`primary key`が必須。一方、`subtable`には`primary key`の指定は不可。
- (6) `subtable`内のカラムの定義順は、先頭に`gentable`から遺伝したカラム、その後に`subtable`の独自カラムの順に定義される。
 - ① `gentable`から遺伝するカラムは、`gentable`での定義順。
 - ② `gentable`が複数あるときは、`subtable`定義時に指定した`gentable`の指定順で、上位の`gentable`から遺伝してきたカラムは同じとみなす。
- (7) `subtable`定義時に指定できる制約定義を表3-1に示す。

以上の定義機能の概要を図 3-3 に示す。

(B) 定義削除

- (1) gentable を削除すると、CASCADE が指定されていれば全ての配下の subtable の定義も削除する。
- (2) subtable の削除は、gentable の定義に影響を与えない。

(C) 定義変更

(1) 全体

- ① 遺伝カラムに関しては、指定できない。

(2) カラム追加

- ① gentable にカラム追加すると、追加カラムは配下の全ての subtable へ遺伝する。
- ② subtable にカラム追加すると、subtable の独自カラムとして追加される。

(3) カラム削除

- ① gentable のカラムを削除すると、CASCADE が指定されていれば subtable のカラムも削除する。

(4) 制約追加

- ① 表 3-1 参照。

3. 3. 3 操作機能

操作機能の概要を表 3-2 に示す。

(A) SELECT (→従来どおりの仕様)

- (1) FROM で指定したテーブルに定義されているカラム (遺伝カラムを含む) のみを参照できる。
- (2) * 指定のとき、遺伝カラムも含まれる。

(B) INSERT

- (1) 指定した primary key の値の行が、そのテーブルに既に格納されていたら重複エラー。(→従来どおりの仕様)
- (2) 格納対象のテーブルが最上位の gentable のとき、その gentable に行を格納する。配下の subtable については何もしない。
- (4) 格納対象のテーブルが subtable のとき、その subtable に行を格納するとともに、INSERT ステートメントで指定した primary key の値と同じ値を持つ gentable の行があるかを全ての gentable について調べ、
 - ① 同じ primary key 値を持つ行がない gentable については、その gentable にも行を格納する。(図 3-4 *①)
 - ② 同じ primary key 値を持つ行がある gentable については、その gentable の行の各カラムの値と、INSERT ステートメントで遺伝カラムに対して指定した値との一致性を調べ、

(a) 一致しなければ、INSERTステートメントをエラーとする。

(→「無視」にすると、後で指定した値が有効になったように誤解されるため)

(b) 一致すれば、何もしない (INSERT対象の subtable に行を格納するだけ)。 (図3-4 * ②)

以上の①、②を全ての gentable に対して行う。

(C) DELETE

(1) 削除対象のテーブルが gentable のとき、

① 該当の gentable の行を削除する。

② 同時に、同じ primary key 値を持つ全ての subtable の行も削除する。

(2) 削除対象のテーブルが subtable のとき、

① 該当の subtable の行を削除する。

② gentable および他の subtable には影響を与えない。

(→他の subtable に行が存在している場合があるので、gentable の行を削除できない)

同じ primary key のものに関する INSERT/DELETE の例を図3-4に示す。

(D) UPDATE

(1) 更新対象のテーブルが gentable のとき、 (図3-5 * 1)

① 該当の gentable の行の更新カラムを更新する。

② 同じ primary key 値を持つ全ての subtable の行の該当カラム (遺伝カラム) も更新する。

(2) 更新対象のテーブルが subtable のとき、

① 該当の subtable の行の更新カラムを更新する。

② 更新カラムが遺伝カラムのときは、 (図3-5 * 2)

(a) そのカラムを定義 (時に指定) した gentable をさがし、その gentable の対応する行の該当カラムを更新する。

(b) (a) で更新した gentable の全ての subtable についても該当カラムを更新する。

(c) (a) と (b) を全ての遺伝カラムである更新カラムについて行う。

(図3-5 * 3)

4. おわりに

本稿では、SQLの意味サポートのうち、`domain`、`subdomain`、`subtable`について述べた。このうち、`subdomain`は内部検討段階で、まだ正式に提案されてはいない。`domain`の一部はSQL2の`draft standard`に含まれる。また`domain`の全部と`subtable`はSQL3規格案に含まれることが確定している。

内部検討段階ではあるが、定義域に関して、複数の列を一つの列にみせる機構についてSQLへの導入の提案を行なう予定である。

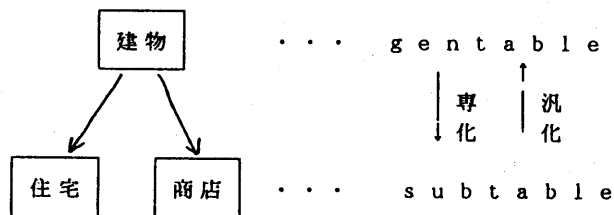


図 3 - 1 汎化／専化階層の例

```

<table definition> ::=
  CREATE TABLE <table name>
  (
    (<table element> [{(<table element>)}...])
    | <subtable clause> [{(<table element> [{(<table element>)}...])}]
  )

<subtable clause> ::= ISA <gentable name> [{(<gentable name>)}...]
  [AS (<column name> [{(<column name>)}...])]

<gentable name> ::= <table name>

<table element> ::= <column definition>
  | <table constraint definition>
  
```

図 3 - 2 `subtable` の定義構文

表 3 - 1 指定可能な制約定義

				遺伝カラム	独自カラム
〈table constraint definition〉					
table constrain t	unique constraint definition	unique specificat ion	UNIQUE	○	○
			PRIMARY KEY	×	×
		unique column list		○	○
	referentia l constraint definition	referentia l constraint	FOREIGN KEY ...	×	○
			references specificat ion	×	○
		triggered action		×	○
	check constraint definition		CHECK ...	○	○
〈column constraint definition〉					
column c onstrain t	NOT NULL			○	○
	unique specification	UNIQUE		○	○
		PRIMARY KEY		×	×
	references specification			×	○
	CHECK...			○	○

○：指定可能 ×：指定不可

CREATE TABLE TA (A1 INTEGER PRIMARY KEY, A2 INTEGER)

CREATE TABLE TB ISA TA (B1 INTEGER, B2 INTEGER)

CREATE TABLE TC ISA TA (C1 INTEGER, C2 INTEGER)

CREATE TABLE TD ISA TB,TC AS(D1,D2,D3,D4,D5,D6)(D7 INTEGER, D8 INTEGER)

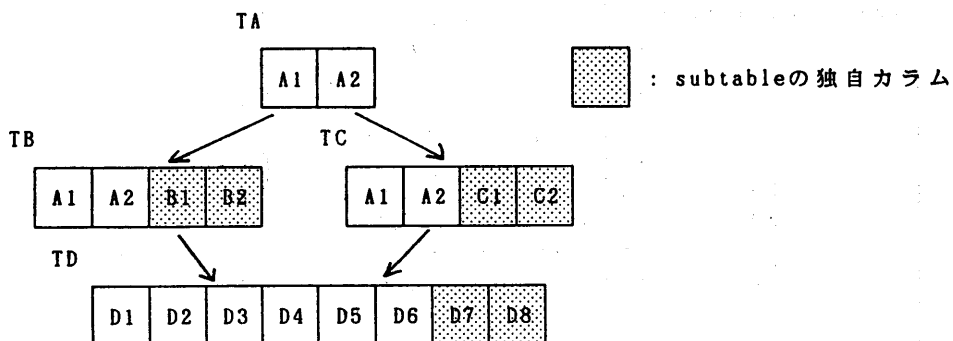


図 3 - 3 テーブル定義の例

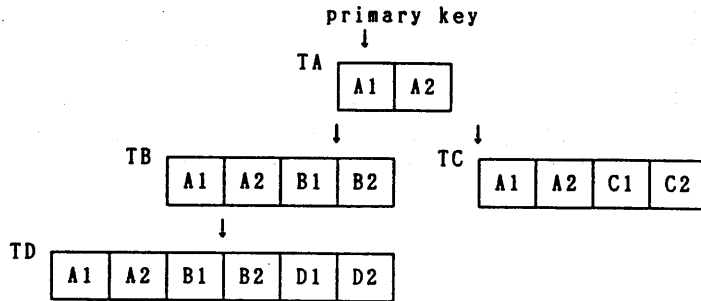
表 3 - 2 操作機能の概略

ステートメント	g e n t a b l e を対象	s u b t a b l e を対象
全体	扱えるカラムはgentableのカラムのみ	扱えるカラムは該当subtableのカラム（遺伝カラムを含む）のみ
SELECT	gentableのカラムのみ参照可能	subtableのカラム（遺伝カラムを含む）を参照可能
INSERT	・ gentableの行を作成	・ subtableの行を作成 ・ 対応するgentableの行がなければgentableの行も作成
DELETE	・ gentableおよび対応する全てのsubtableの行を削除する	・ subtableの行を削除する ・ 対応するgentableの行は影響なし
UPDATE	・ gentableの行を更新し、対応する全てのsubtableの行の該当カラムも更新する	・ subtableの行を更新する ・ 遺伝カラムを更新したときは、対応するgentableのうち該当カラムを持つgentableについて、そのカラムを更新する ・ さらにそのsubtableについても更新する

「対応する」とは、同じprimary keyを持つ行を意味する

Definition

CREATE TABLE TA (A1 PRIMARY KEY,A2)
 CREATE TABLE TB SUBTABLE OF TA (B1,B2)
 CREATE TABLE TC SUBTABLE OF TA (C1,C2)
 CREATE TABLE TD SUBTABLE OF TB (D1,D2).



Statement

INS TA: INSERT INTO TA (A1,A2) VALUES(10,11) a 1
 INS TB: INSERT INTO TB (A1,A2,B1,B2) VALUES(10,11,20,21) b 1
 INS TC: INSERT INTO TC (A1,A2,C1,C2) VALUES(10,11,30,31) c 1
 INS TD: INSERT INTO TD (A1,A2,B1,B2,D1,D2) VALUES(10,11,20,21,40,41) d 1

DEL TA: DELETE FROM TA WHERE A1=10
 DEL TB: DELETE FROM TB WHERE A1=10
 DEL TC: DELETE FROM TC WHERE A1=10
 DEL TD: DELETE FROM TD WHERE A1=10

Status

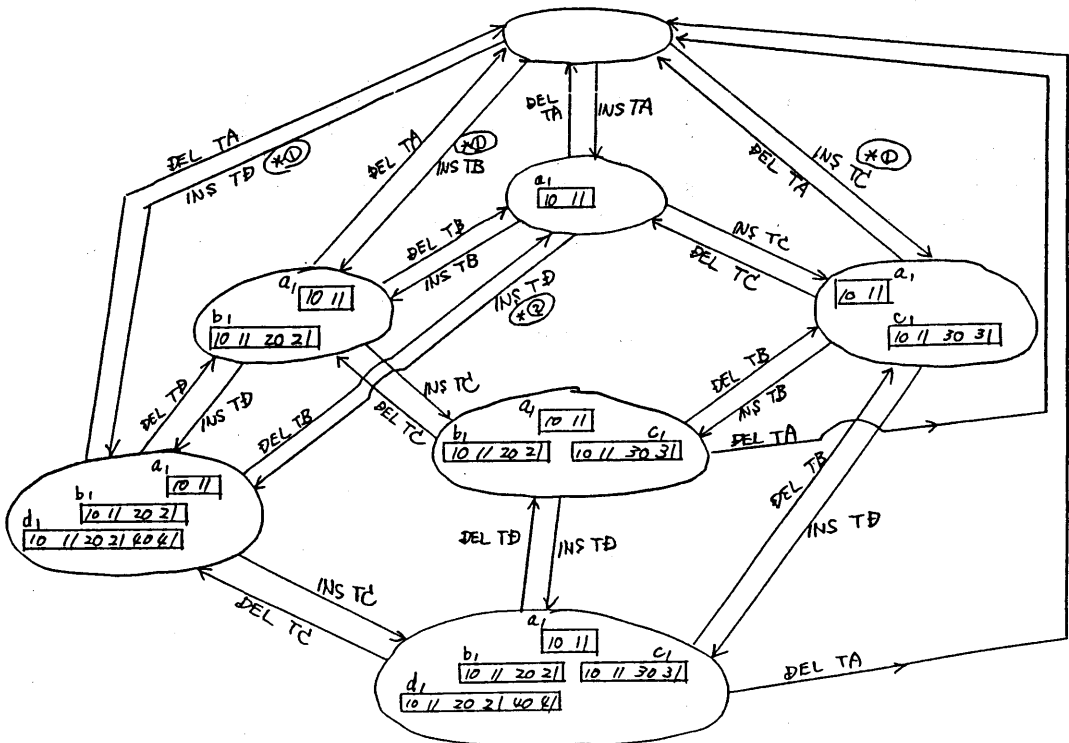
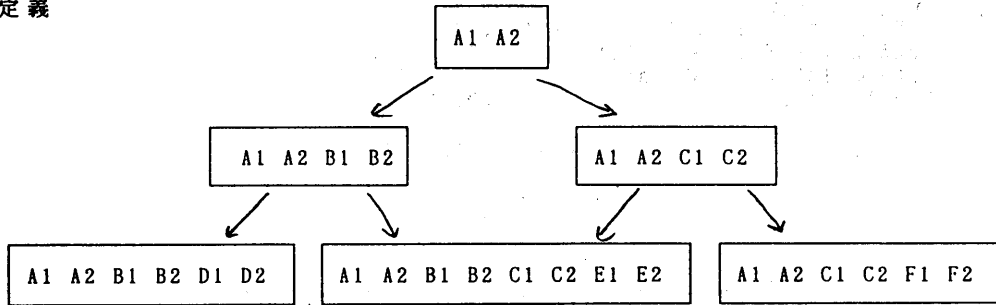
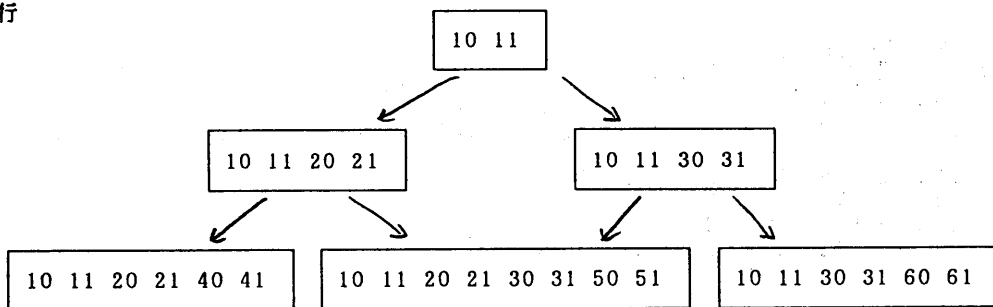


図 3-4 INSERT/DELETE の例

定義



行



UPDATEステートメント

(*1):UPDATE TA SET A2=13 WHERE A1=10 → a1～f1のA2を更新
 (*1):UPDATE TB SET B2=23 WHERE A1=10 → b1,d1,e1のB2を更新
 (*2):UPDATE TB SET A2=13 WHERE A1=10 → a1～f1のA2を更新
 (*2):UPDATE TD SET A2=13 WHERE A1=10 → a1～f1のA2を更新
 (*2):UPDATE TE SET A2=13,B2=23,C2=33 WHERE A1=10
 → a1～f1のA2, b1,d1,e1のB2, c1,e1,f1のC2を更新

図 3 - 5 UPDATE の例