

SQL のビューサポート —その標準化の経緯と現状—

増永 良文¹ 小寺 孝²

ビューとその更新可能性は国際標準リレーショナルデータベース言語 SQL の最初の規格である SQL-87 で標準化されて以来、最新の SQL:2023 に至るまで改正され続けている。多様なビューの更新可能性を少しでも向上させるがためである。しかしながら、改正を重ねるにつれ規格は複雑化して難解となり、ビューを利活用しようとするクライアントにとってはハードルが高く、一方、ベンダにとってもその機能のすべてを実装することが難しくなり、更新可能なビューの最新規格をサポートするプロプライエタリあるいは OSS のリレーショナル DBMS は現存していないことも事実である。本稿の目的は、SQL のビューサポート規格の改正の経緯を原点から最新まで忠実に辿ることで SQL が規格化している更新可能なビューの全貌を詳らかにし、更新可能なビューに関心を持つクライアントならびにリレーショナル DBMS ベンダに資することにある。

1. はじめに

ビュー (view) はリレーショナルデータベースの始祖コード (E. F. Codd) により 1974 年に導入された[1]。リレーショナルデータベースの現場では、繰り返し作成する問合せをビューとして定義しておくことで利便性や生産性を高め、実表 (stored table) そのものではなくその一部をビューとして公開することでセキュリティを高め、ANSI/X3/SPARC の DBMS の標準アーキテクチャにおいては論理的データ独立性を達成する手段として有用である。ビューへの問合せは質問変形 (query modification) という手法[2]により常に可能であるが、ビューは定義だけが存在して実体がない表 (table) なので、それを更新しようとする更新要求を基表 (underlying table) への更新に一意に変換しなければならず、更新できる場合もあればそれができない場合も生じる。では、ビューは一体どのような条件を満たすときに更新可能となるのであろうか？これをビュー更新問題という。

さて、国際標準リレーショナルデータベース言語 SQL (以下、単に SQL) は、その初めての標準である SQL-87[3]でビューとその更新可能性を規格化している。しかしながら、そこで規格化された更新可能なビューの概念は極めて初歩的であった。したがって、それ以降、SQL-92[4]、SQL:1999[5]、SQL:2003[7]、SQL:2008[10]、SQL:2016[11]などを経て SQL:2023[12]に至るまで、SQL はビューの更新可能性を向上させるべく改正に改正を重ねている。しかしながら、ビュー定義の基となる<問合せ指定>や<問合せ式>が機能豊かであることや、そもそもビューの更新可能性をどのような形で規格化するべきかなどさまざまな問題に直面し、改正作業は紆余曲折を経て現在に至っているように見受けられる。

しかしながら、更新可能な SQL ビューのサポートはクライアントにとってもリレーショナル DBMS ベンダにとっても大きな関心事であることは言を待たず、本稿の目的は、SQL-87 より始めて現在に至るまで、SQL のビューサポート規格の改正の経緯を忠実に辿ることにより更新可能なビューの現状を詳らかにし、SQL のビューサポートの経緯と現状を理解しておきたいと考えている者に資することにある。

2. SQL におけるビュー定義とその更新可能性

SQL でのビュー定義とその更新については、上述のとおり SQL の最初の国際標準規格である SQL-87 に記述がある。ビューとその更新については、当初からその重要性が認識されていたということであろう。SQL はそれ以降、主な改正だけでも SQL-92、SQL:1999、SQL:2003、SQL:2008、SQL:2011、SQL:2016、SQL:2023 と現在に至っているが、ビューの更新可能性についてもそれぞれで相応の改正がなされている。

さて、その改正の内容を時系列で見ると、これまでの改正を 3 段階に分けて考えてみると見通しが良いように考えられる。

第 1 段階：SQL-87 から SQL-92 に至る改正

第 2 段階：SQL:1999 での改正

第 3 段階：SQL:2003 から SQL:2023 に至る改正

まず、第 1 段階であるが、SQL 規格の最初である SQL-87 で与えられたビューとその更新可能性の概念が SQL-92 での改正で一応のまとまりを見せたということである。SQL-92 でも結合ビューや UNION ビューは定義可能ではあったが、それらの更新可能性は規格外であった。第 2 段階の SQL:1999 でビューの更新可能性が結合ビューや UNION ビューに拡張された。また、SQL:1999 以降、適合性 (conformance) のために SQL 規格の機能全体が細かい機能区分に分類され、最低限の適合が必要な部分 (中核 SQL) 以外は、適合性を主張する範囲を機能区分によって選択できるように改正された。第 3 段階は、SQL:2003 に始まり SQL:2008、SQL:2011、SQL:2016 を経て SQL:2023 に至る改正である。この段階で更新可能性に関する記述が整備され、概念的な整理が行われた。

以下、上述のような認識のもと、これら 3 段階で肝要と考えられる事項に言及し、更新可能なビューの全貌を明らかにしようとする。

3. SQL-87 と SQL-92 にみるビューサポート

SQL ではビューサポートに関する初めての規格が SQL-87 で与えられたが、まずは、そこでビューとその更新可能性がどのように定義されていたのか、それを見てもことから始める。なお、SQL では、<問合せ指定>や<問合せ式>の結果として導出される表をビュー表 (viewed table) として定義することで、その定義が永続化され、実表と同様に、あたかもその表が存在しているかのように操作できる。ここで、<問合せ指定>のような<と>で囲んだ表記は、SQL 規格において構文の形式を規定する BNF の非終端記号 (構文

¹ 正会員 お茶の水女子大学 名誉教授・SRA OSS LLC 技術顧問 masunaga.yoshifumi@ocha.ac.jp

² 正会員 日立製作所・元 ISO/IEC JTC1/SC32 WG3(SQL 言語)日本代表 takashi.kotera.xa@hitachi.com

要素) を表す.

3.1 SQL-87

SQL-87[3]でのビュー定義 (view definition) とその更新可能性についての記述は次のとおりである. なお, 以下, 全編にわたり, SQL 規格の翻訳は JIS X 3005-2 [13]の表現に準じ, ISO の原用語(英語)を JIS 用語(日本語)に置き換えている.

形式 (Format) :

```
<ビュー定義> ::=
CREATE VIEW <表名> [<ビューリスト>]
AS <問合せ指定>
[WITH CHECK OPTION]
```

構文規則 (Syntax Rules) : (文献[3]より抜粋し翻訳. 番号は原文のママ)

3) <問合せ指定>が更新可能 (updatable) ならば, ビュー表は更新可能な表 (table) とする. そうでなければ, 読取り専用の表とする.

9) WITH CHECK OPTION を指定するならば, ビュー表は更新可能でなければならない.

一般規則 (General Rules) : (文献[3]より抜粋し翻訳. 番号は原文のママ)

2) V が更新可能ならば, <問合せ指定>の最初の<FROM 句>で指定された<表名>によって識別される表を T とする. V の各行に対して, V のその行が導出される T の対応行が T に存在し, V の各列に対して, V のその列が導出される T の対応列が T に存在する. V への行の挿入は, T への対応行の挿入とする. V からの行の削除は, T への対応行の削除とする. V のある行のある列の更新は, T 中の対応行の対応列の更新とする.

3.2 SQL-92

SQL-92[4]はリレーショナル DBMS としての完全性を追求した規格として知られている. この規格で 3 つの適合性水準: 初級 (entry), 中級 (intermediate), 完全 (full) が導入された. ここで適合性 (conformance) とは, リレーショナル DBMS を実装するにあたっては, まずは初級を, できれば中級を, そして完全をといった具合に, いずれかの水準に準拠するようにと意図され導入された. しかしながら, 多くのリレーショナル DBMS が初級に準拠し, 中級以上の主要な機能を選択的に提供するに止まった. その理由として, 完全性を追求したが故に, 結果的にユーザでの利用方法, SQL 製品での実装技術を考慮した適合性水準が必ずしもうまく設定できなかったことが指摘されている[9]. なお, 適合性試験 (=コンFORMANCE試験) については7章で詳述する.

さて, SQL-92 でのビュー定義は次の通りである.

形式 :

```
<ビュー定義> ::=
CREATE VIEW <表名>
[ <左括弧> <ビューリスト> <右括弧> ]
```

AS <問合せ式>

[WITH [<レベル句>] CHECK OPTION]

構文規則 : (文献[4]より抜粋し翻訳. 番号は原文のママ)

6) <問合せ式>が更新可能 (updatable) ならば, ビュー表は更新可能な表とする. そうでなければ, それは, 読取り専用の表とする.

そこで, SQL-87 と SQL-92 のビュー定義の形式を比較してみると大きな相違点の一つあることに気が付く. それは, SQL-92 では <問合せ指定>ではなく<問合せ式>となっている点である. ちなみに, <問合せ式>の定義は次のようである.

```
<問合せ式> ::=
<非結合問合せ式> | <結合表>
<非結合問合せ式> ::=
<非結合問合せ項>
| <問合せ式> UNION [ ALL ] [ <CORRESPONDING 指定> ]
<問合せ項>
| <問合せ式> EXCEPT [ ALL ] [ <CORRESPONDING 指定> ] <問合せ項>
<問合せ項> ::= <非結合問合せ項> | <結合表>
<非結合問合せ項> ::= <非結合問合せ一次子>
| <問合せ項> INTERSECT [ ALL ] [ <CORRESPONDING 指定> ] <問合せ一次子>
```

つまり, SQL-92 ではビューとして, <問合せ指定>で定義できる結合ビューに加えて<問合せ式>で定義できる UNION ビュー, EXCEPT ビュー, INTERSECT ビューが更新可能なビューの対象となるように拡張がなされたということである. また, これらのビューに対する更新可能性については, 上述の構文規則 6)で与えられているように, <問合せ式>が更新可能の場合, ビュー表も更新可能と定義しているのだから, 規格としては, それらの更新可能性にまで踏み込んだビューサポートがあっても然るべきとの見識が SQL-92 規格の背景には存在したのだからと考えられる.

しかしながら, SQL-92 では適合性水準と関連して, ビュー定義に水準規則が次のように定められている.

水準規則 (Leveling Rules) : (文献[4]より抜粋し翻訳. 番号は原文のママ)

- 1) 中級 SQL には次の制限が適用される.
 - a) 中級 SQL 言語には <レベル句> が含まれてはならないが, その効果は CASCADED の <レベル句>に対して定義されたものとする.
- 2) 中級 SQL の制限に加えて, 次の制限が初級 SQL に適用される.
 - a) <ビュー定義>内の<問合せ式>は<問合せ指定>とする.

■ SQL-92 における更新可能なビュー

さて, リレーショナル DBMS の多くが初級 SQL を実装するにとどまっていることで, SQL-92 における更新可能なビューは実質的に問合せ式ではなく問合せ指定 (query specification) の更新可能

性で規定されることとなる。ここに、問合せ指定が更新可能であるための必要かつ十分条件は次の条件をすべて満たすことである。

- (a) 限定子として DISTINCT を使用していないこと。
- (b) 選択リスト中の値式 (value expression) がすべて列参照 (column reference) で、それらのどれもが 2 回以上現れないこと。
- (c) FROM 句はただ 1 つの表参照 (table reference) を有すること、そしてそれは実表か更新可能な導出表を特定していること。
- (d) 表参照によって最終的に識別される実表は、WHERE 句の副問合せの FROM 句で (直接あるいは間接的に) 参照されていないこと。
- (e) GROUP BY 句や HAVING 句が使われていないこと。

□

端的には、SQL-92 における更新可能なビューは「更新が可能なビューは、実表から選択または射影だけを行っており、ビュー表の行および列が基となる実表の行および列と 1 対 1 に対応している場合である」(文献[8]の p.128) と紹介されることが多い。

なお、結合ビュー (join view) は問合せ指定で定義できるので、その更新可能性が規格化の対象となってもおかしくないが、規格では上述の条件(c)により排除され、また、結合ビューは一般に更新変換の曖昧性によりスキーマレベルでは更新不可と結論されること[14]、そして「完全 SQL」を実装したリレーショナル DBMS は未だに存在しないことを勘案すれば、SQL-92 時点で上記のようにビューの更新可能性を規定したことは頷けることである。形式上、SQL-92 のビューは問合せ式 (query expression) を使って定義できるとされているが、SQL-92 では、問合せ式については、UNION も EXCEPT も INTERSECT も指定しないことが更新可能の条件とされている[4]。従って、UNION ビューなどの更新可能性も規格化の対象から外されている。

4. SQL:1999 にみるビューサポート

4.1 SQL:1999 におけるビューの更新可能性

SQL:1999[5]でビューの更新可能性について大幅な見直しが行われた。ビューを定義するにあたり、問合せ指定 (QS) ではなく問合せ式 (QE) を用いて定義できる点は SQL-92 と同じであるが、上述のように SQL-92 では実質的に QS のみが規格化の対象となったこととは異なり、SQL:1999 では QE もその対象とするべくその更新可能性を規格化しようと改正が行われたということである。一体どうということなのか？まず、ビュー定義から見ていく。

SQL:1999 のビュー定義は以下に示す通りであるが、それが SQL-92 のビュー定義と異なる点は、RECURSIVE ビューが定義可能となったことと、[<左括弧> <ビュー列リスト> <右括弧>] が <ビュー指定> となったことである。

形式：

```
<ビュー定義> ::=
CREATE [ RECURSIVE ] VIEW <表名>
```

<ビュー指定>

AS <問合せ式>

[WITH [<レベル句>] CHECK OPTION]

構文規則：(文献[5]より抜粋し翻訳。番号は原文のまま)

- 9) ビュー表が更新可能 (updatable) であるのは<問合せ式>が更新可能であるとき及びそのときのみ。
- 10) ビュー表が挿入可能 (insertable-into) であるのは<問合せ式>が挿入可能であるとき及びそのときのみ。

実は、SQL:1999では、規格の箇条7.11<問合せ指定>、箇条7.12<問合せ式>でそれぞれの更新可能性が、そして箇条11.21<ビュー定義>で、機能、形式に続き、24個の構文規則 (Syntax Rules)、3個のアクセス規則 (Access Rules)、11個の一般規則 (General Rules) に加えて3個の適合性規則 (Conformance Rules) が規定されている。適合性についてのSQL:1999のスタンスは、リレーショナルDBMSがSQL:1999のビュー定義に準拠したビューサポートを提供していると謳うには、そこに記されているすべての規則を満たしていることを保証しないとイケないということである。しかしながら、それを実践することは至難で、これまでそれを謳っているプロプライエタリあるいはOSSのリレーショナルDBMS製品は見当たらない。

さて、SQL:1999が更新可能とするビューはどのようなビューなのか。それをどのように規定していくのかを箇条7.11ならびに箇条7.12に見ていくが(文献[5]より抜粋し翻訳。番号は原文のまま)、SQL:1999における更新可能なビューは、「問合せ指定の更新可能性」と「問合せ式の更新可能性」の2つに分けて規定されている。

■ SQL:1999 における更新可能なビューの条件

一問合せ指定の更新可能性一

- 19) <問合せ指定> QS が潜在的に更新可能 (potentially updatable) であるための必要かつ十分条件は次がすべて成り立つこと：
 - a) DISTINCT が指定されていない。
 - b) <選択リスト (select list)>中に、同じ列参照 (column reference) である<導出列 (derived column)>が複数回指定されることはない。
 - c) QS に直接含まれる<表式 (table expression)>は<GROUP BY 句>または<HAVING 句>を含まない。なお、<表式> ::= <FROM 句> [<WHERE 句>] [<GROUP BY 句>] [<HAVING 句>] である。
- 20) <問合せ指定> QS は、QS のすべての基表 (underlying table) が挿入可能 (insertable-into) である場合にのみ挿入可能である。
- 21) <問合せ指定> QS が潜在的に更新可能である場合：
 - a) <表式>の<FROM 句>が正確に 1 つの<表参照 (table reference)> TR を指定している場合、QS の列に対応する更新可能な列が TR にある場合、QS のその列は更新可能な列であるといわれる。
 - b) それ以外の場合、QS の列は、QS が基表 UT に対して 1 対 1 であり、QS の UT の一部の列に対応するものがある場合、

更新可能な列であるといわれる。

- 22) <問合せ指定>は、それが潜在的に更新可能であり、かつ少なくとも 1 つの更新可能な列がある場合、更新可能 (updatable) である。
- 23) <問合せ指定> QS は、それが更新可能であり、 QS に直接含まれる<表式>に直接含まれる<FROM 句>が 1 つの<表参照>を含み、かつ QS のすべての結果列が更新可能であるとき、単純に更新可能 (simply updatable) といわれる。

一問合せ式の更新可能性一

- 6) <問合せ式> $QE1$ が更新可能 (updatable) であるための必要かつ十分条件は、すべての<問合せ式>あるいは $QE1$ にそのまま含まれている<問合せ式> $QE2$ に対して次がすべて成り立つこと：
- a) $QE1$ は、UNION DISTINCT, EXCEPT ALL, または EXCEPT DISTINCT を指定している<非結合問合せ式 (non-join query expression) > $NJQE$ の存在なしに $QE2$ を含んでいること。
- b) もし $QE1$ が UNION ALL を指定する<非結合問合せ式> $NJQE$ をそのまま含んでいるならば、次がすべて成り立つこと：
- i) $NJQE$ は、<問合せ式> LO と<問合せ項 (query term) > RO を直接含み、その結果として LO の通常は基表としてのリーフが RO の一般に基となる表であるリーフになることはないこと。
- ii) $NJQE$ の各列について、 LO および RO で識別されるテーブルの基底列 (underlying columns) は、それぞれ、共に更新可能または更新不可であること。
- c) $QE1$ は、INTERSECT を指定する<非結合問合せ式>の存在なしに $QE2$ を含むこと。
- d) $QE2$ は更新可能であること。
- 7) <問合せ式> $QE1$ が挿入可能であるための必要かつ十分条件は、 $QE1$ が<問合せ式>または<問合せ指定> $QE2$ をそのまま 1 つだけ含み、 $QE2$ が挿入可能であること。

□

さて、ここで注意するべきは、ビューの更新可能性について、SQL-92 では「更新可能」という 1 つの概念でそれが規格化されていたが、SQL:1999 では基本的に「更新可能」と「挿入可能」という 2 つの概念でそれが規格化されている点である (詳細には「潜在的に更新可能」、「挿入可能」、「更新可能」、「更新可能な列」、「単純に更新可能」と細分化されている)。これは SQL-92 では先述のごとく更新可能なビューは基表と行と列が 1 対 1 に対応していたので、ビューへの行の削除、挿入、(列への) 更新を区別する必要がなかったが、SQL:1999 では結合ビューや UNION ビューの更新可能性を考えると、更新 (削除を含む) と挿入を分けて考えないといけないという理由による。どういうことかということ、更新はできても挿入はできないというビューが存在するということである。たとえ

ば、表 R と S の UNION ビュー $V (=R \text{ UNION } S)$ を考えたとき、 V の更新は可能だが、 V にタプル t を挿入しようとしたときに、 t を R に挿入するべきか、 S に挿入するべきか、あるいは両方に挿入するべきか、意味的曖昧性が生じるので、挿入は不可としなければならない。

以下、まずは、これら多様な概念を整理することから始める。続けて、SQL:1999 では SQL 規格への適合性の主張のために機能全体が細かく分類され、分類された区分それぞれに機能識別子が割り当てられたが、それを概観する。さらに、本章後段で、簡単ではあるが的確な例を用いて SQL:1999 規格のもとでの結合ビューや UNION ALL ビューの更新可能性を議論する。

4.2 ビューの更新可能性の多様な概念

上記のように SQL:1999 ではビューの更新可能性について多様な概念が導入されたが、文献[8]の 4.6 節にそれを「問合せ更新可能性」と「問合せ挿入可能性」という観点から整理した記述があるので、それを紹介する。SQL:1999 規格のビューサポートを実装しようとした場合の一助となるはずである。

[1] 問合せ更新可能性

① 表参照の更新可能性

- ・表参照が実表の表名である場合、表参照は更新可能であり、表参照のすべての結果列は更新可能である。
- ・表参照がビュー表の表名である場合、それを定義する問合せ式が更新可能ならば表参照は更新可能であり、問合せ式の更新可能な結果列は更新可能である。
- ・表参照が導出表、横参照導出表である場合、それを導出する問合せ式が更新可能ならば表参照は更新可能であり、問合せ式の更新可能な結果列は更新可能である。

② 問合せ指定と基表の 1 対 1 対応

③ 問合せ指定の潜在的更新可能性

「SQL:1999 における更新可能なビューの条件」の 19) 項で定義されている通り。

④ 問合せ指定の更新可能性

「SQL:1999 における更新可能なビューの条件」の 22) 項で定義されている通り。

更新可能列については「SQL:1999 における更新可能なビューの条件」の 21) 項で定義されている通りである。

- ・FROM 句がただ 1 つの更新可能な表参照で構成される場合、その表参照の更新可能な結果列は更新可能である。
- ・問合せ指定と 1 対 1 対応の基表の更新可能列は更新可能である。

⑤ 問合せ式の更新可能性

問合せ式が集合演算子である場合、次のすべてが満たされるならば、問合せ式は更新可能である (「SQL:1999 における更新可能なビューの条件」の 6) 項)。

- ・集合演算は UNION ALL である。
- ・2 つの演算項が同じ表を基としない。

- ・2つの演算項の結果列の対応する列どうしの更新可能性は同一である。
- ・集合演算項は更新可能である。

[2] 問合せ挿入可能性

① 表参照の挿入可能性

- ・表参照が実表の表名である場合、表参照は挿入可能である。
- ・表参照がビュー表の表名である場合、それを定義する問合せ式が挿入可能ならば表参照は挿入可能である。
- ・表参照が導出表、横参照導出表である場合、それを導出する問合せ式が挿入可能ならば表参照は挿入可能である。

② 問合せ指定の挿入可能性

「SQL:1999における更新可能なビューの条件」の20)項で定義されている通り。

③ 問合せ式の挿入可能性

「SQL:1999における更新可能なビューの条件」の7)項で定義されている通り。

4.3 ビューの更新可能性に関連する機能識別子の導入

SQL:1999では、SQL規格への適合性の主張のために機能全体が細かく分類され、分類された区分それぞれに機能識別子が割り当てられた。機能識別子の示す各機能は中核SQL (core SQL) と適合性を追加的に主張できる部分に大きく分類される。中核SQLは、SQL規格への適合性のため必須の最小言語仕様で、SQL-92の初級水準を完全に包含し、その中級と完全水準の一部機能を含む機能設定になっている。また、この中核SQLに準拠することに加えて、追加的に機能を選択して主張することができる。さらに、パッケージと呼ぶデータベースアプリケーションの特定の分野に向けて追加機能をまとめた設定に適合させることができる。中核SQLは、SQL規格に準拠するリレーショナルDBMS製品として最小限のユーザ要求を満たし、かつ容易に実装可能である範囲に限定されている。

注目すべきは、ビューの更新可能性に関連して、機能識別子T111、機能名称「更新可能な結合、UNION及び列」、機能記述「キー保存による一対一対応関係に基づく問合せ指定の更新可能性、問合せ式の更新可能性」が適合機能としてリストアップされていることである。SQLの適合性試験の歴史と経緯については7章で改めて論じる。

4.4 SQL:1999のもとの結合ビューの更新可能性

結合ビューは<問合せ式>を持ち出さずとも<問合せ指定>で定義できるのでSQL-92規格のもとでも結合ビューは定義できるが、更新不可能とされた理由は前述の通りである。では、SQL:1999規格のもとではどのような扱いとなるのか、それを見てみる。

まず、一般に結合ビューはそれに対して発行された更新操作を基表に一意に変換することができないことから更新不可能とされていることはよく知られたことであるが、一方で、一定の条件を満たすと更新可能となる場合があることも示せる。そのような簡単な

例を次に示す。

【例題1】(SQL:1999のもとで更新可能な結合ビューの例)

実表R(A, B)とS(B, C), ここに、BはSの主キーとする、の自然結合ビューを次のように定義する。

```
CREATE VIEW V
AS SELECT A, R.B, C
FROM R, S
WHERE R.B=S.B
```

このとき、BがSの主キーであるのでB=Cという関数従属性が成立している。したがって、VのタプルとRのタプルの間に1対1の関係性が出来上がり、Vは更新可能となる(文献[14]の7.6節)。

□

なお、SQLの規格化に深く関わってきたメルトン(J. Melton)らの著書[6]では結合ビューの更新可能性を紹介したあと、節を改めて関数従属性を(関係ありげに)紹介しているので、暗に上述のような関係性を意識しているのかなと想像されるが、結合ビューの更新可能性に関する規格と関数従属性を直接関連付ける記述は見当たらない。

さて、このような認識の下で、例題1の結合ビューVにつき、SQL:1999で結合ビューの更新可能性がどのように判定されていくのか、規格に照らし合わせながら見ていく。まず、箇条11.21<ビュー定義>の構文規則の9)項「ビュー表が更新可能(updatable)であるのは<問合せ式>が更新可能であるとき及びそのときのみ」なので、Vについて箇条7.11の「問合せ指定の更新可能」の19)項の検証を行うと、a), b), c)をすべて満たしているのでVは潜在的に更新可能であることが分かる。潜在的に更新可能であり表式が2つの表参照、この例ではRとS、をしているので、21)項のb)の検証に移る。「QSの列は、QSがUTに対して1対1であるように、QSの基表UTの一部の列に対応する更新可能なものがある場合、更新可能な列であるといわれる」については、QSの列は例題1ではA, R.B, Cで、AとR.BはそれぞれUTのRのAとBに1対1であり、CはUTのSのCと1対1であるので、更新可能な列である。そうすると、22)項「<問合せ指定>は、それが潜在的に更新可能であり、かつ少なくとも1つの更新可能な列がある場合、更新可能(updatable)である」により、Vは更新可能であると結論できる。

リレーショナルデータベースでは関連するデータが複数のテーブルに分散して格納されている状況は多々あり、それらのテーブルを結合して生成される結合ビューをサポートすることにはとても意味がある。加えて、もし結合ビューが更新可能であればその有用性は格段に向上しよう。実働しているプロプライエタリあるいはOSSのリレーショナルDBMSのほとんどは先述の通りSQL-92に準拠しているので更新可能な結合ビューのサポートはできないが、中にはSQL:1999で新しく拡張された1対1対応に基づく更新可能性を踏まえて、更新可能な結合ビューのサポートを謳うリレーショナルDBMSが出現していることも事実である。

4.5 SQL:1999 のもとでの UNION ALL ビューの更新可能性

SQL:1999 のもとでは、ビューの定義に<問合せ指定>ではなく<問合せ式>を指定できるので、UNION [ALL | DISTINCT], EXCEPT [ALL | DISTINCT], INTERSECT [ALL | DISTINCT] を使って定義されるビューを生成できる (デフォルトは DISTINCT)。ここでは「SQL:1999 における更新可能なビューの条件」の一問合せ式の更新可能性一に従って、UNION ALL ビューの更新可能性を例題と共に見ていく。

そこで、一定の条件を満たす UNION ALL ビューが更新可能となることを SQL:1999 規格と照らし合わせて検証してみる。

【例題 2】(SQL:1999 のもとでの UNION ALL ビューの更新可能性)

和両立な 2 つの実表 `tokyo_emp(emp_id, emp_name, dept)` と `osaka_emp(emp_id, emp_name, dept)` の UNION ALL ビュー `emp(emp_id, emp_name, dept)` を次のように定義する。

```
CREATE VIEW emp(emp_id, emp_name, dept)
AS SELECT * FROM tokyo_emp
UNION ALL
SELECT * FROM osaka_emp
```

ここで、`tokyo_emp` と `osaka_emp` は更新可能とする。このとき、一問合せ式の更新可能性一で導入された `QE1` はキーワード `AS` に続いて指定されている<問合せ式>である。

```
SELECT * FROM tokyo_emp
QE1: UNION ALL
SELECT * FROM osaka_emp
```

したがって、`QE1` にそのまま含まれている<問合せ指定>`QE2` はこの場合 2 つあり、それらは次の通りとなる。

```
QE21: SELECT * FROM tokyo_emp
QE22: SELECT * FROM osaka_emp
```

そこで、<問合せ式> `QE1` が更新可能 (updatable) であるための必要かつ十分条件を吟味してみる。

6)項 a) の条件は、`QE1` が UNION ALL を用いて定義されているので成立している。

6)項 b)については、まさしく `QE1` は UNION ALL を用いており `NJQE = QE1` である。

そこで、b)-i) と b)-ii) を検証する：

b)-i) : 「`NJQE` は、<問合せ式> `LO` と<問合せ項 (query term) > `RO` を直接含み、その結果として `LO` の通常は基表としての末端 (leaf) が `RO` の一般に基となる表である末端になることはないこと」については、この例では、`LO = QE21`, `RO = QE22` であるので、成立している。

b)-ii) : 「`NJQE` のすべての列について、`LO` および `RO` で識別されるテーブルの基底列 (underlying columns) は、それぞれ、共に更新可能または更新不可能であること」については、`LO` および `RO` で識別されるテーブルはそれぞれ `tokyo_emp` と `osaka_emp` であり、それらは更新可能としているから、成立している。

c) : INTERSECT を使用していないので成立する。

d) : `QE2` が更新可能であることを要求しているが、`tokyo_emp` と `osaka_emp` は更新可能としているので成立している。

したがって、加法和ビュー `emp(emp_id, emp_name, dept)` は更新可能である。 □

なお、SQL:1999 におけるビュー表に対する更新については、次に示すように、問合せ式を構成する問合せ指定と 1 対 1 対応となる基表 (の行や行の列) の存在が前提となっていることを改めて記しておく (文献[8]の 4.6.5 項)。

- ① UPDATE 文の更新対象表にビュー表を指定する場合、ビュー定義で指定された問合せ式が更新可能でなければならない。問合せ式を構成する問合せ指定と 1 対 1 対応となる基表の行の列が更新対象となる。
- ② DELETE 文の削除対象表にビュー表が指定される場合には、ビュー定義で指定された問合せ式が削除可能でなければならない。問合せ式を構成する問合せ指定と 1 対 1 対応となる基表の行が削除対象となる。
- ③ INSERT 文の挿入対象表にビュー表が指定される場合、ビュー定義で指定された問合せ式が挿入可能でなければならない。問合せ式を構成する問合せ指定と 1 対 1 対応となる基表が挿入対象となる。

5. SQL:2003 以降 SQL:2023 に至るビューサポート

SQL:1999 で拡張されたビューの更新可能性について、その後の規格の改正で概念、用語の整理と記述の整備が行われたが、その主な改正は SQL:2008 と SQL:2016 でなされている。一方、SQL:2003 ではビューの更新可能性そのものに対する改正は見受けられないが、そこで導入された<生成列>がビューの更新可能性に影響を与えているので、それを指摘することから議論を始める。

5.1 SQL:2003

SQL:2003[7]ではビューとその更新可能性についての改正は見られず、その規格は SQL:1999 のそれと同一である。しかしながら、SQL:2003 では新しい列の種類として<生成列>(generated columns) が導入された[7, 9]。生成列は実表に定義する列で、その表の列に対する演算結果の値が保持される。この機能はビューのための機能というわけではないが、ビューの更新可能性を高めるための利用価値がある。ビュー表に演算を行う列を定義すると、その列は更新可能とはならないが、ビュー定義で演算を行う代わりに、同じ演算を行う生成列を基となる実表に定義して、その列をビューの列の基にするとその列は更新可能な列になる。同じ結果を得るビュー表ではあるが、生成列を基とすることで列が更新可能となる。具体例を示す。

【例題 3】(SQL:2003 の<生成列>とビューの更新可能性)

従業員の身長と体重を保持する `checkout` 表から、BMI 値を参照するビュー表 `bmi` を定義することを考える。BMI 値は `weight(kg)`

$\div((\text{height(m)})^2)$ のように計算される。

一つの方法は、これを単純に次のように実現することである（基表の列のデータ型は適当）。

```
CREATE TABLE checkup(emp_id CHAR(7), emp_name
    VARCHAR(128), height DEC(4,1),weight DEC(4,1))
CREATE VIEW bmi(emp_id, emp_name, bmi)
    AS SELECT emp_id, emp_name, weight/POWER(height,2)
    FROM checkup
    WHERE weight/POWER(height,2)<18.5
        OR weight/POWER(height,2)>=25
```

このビューは `bmi` 列以外の列が更新可能なので更新可能であるが、`bmi` 列は、基となる表の列そのものを基とする列ではなく演算結果なので更新可能ではない。

もう一つの方法として、`checkup` 表に生成列を定義する方法を考えてみる。

```
CREATE TABLE checkup(emp_id CHAR(7), emp_name
    VARCHAR(128), height DEC(5,2),weight DEC(4,2), bmi
    GENERATED ALWAYS AS weight/POWER(height,2))
CREATE VIEW bmi(emp_id, emp_name, bmi)
    AS SELECT emp_id, emp_name, bmi
    FROM checkup
    WHERE bmi<18.5 OR bmi>=25
```

このビューの `bmi` 列は基となる `checkup` 表の `bmi` 列をそのまま基とするので更新可能である。`bmi` 列の値は `height` 列と `weight` 列の値から決まる演算結果の値である必要があるので、更新や挿入の際に、更新値や挿入値を明示的に指定する場合には、値ではなく `DEFAULT` を指定する必要がある（例えば更新の指定では `bmi=DEFAULT` と指定する）。

□

5.2 SQL:2008

SQL:2008[10]での変更点は、更新可能性の定義が「機能識別子 T111」を用いて行われるようになった点と「実効的に更新可能」という用語が導入された点である。

■ 機能識別子 T111 による更新可能性の規定

ビューが単純に更新可能ではない場合は、問合せ指定と基となる表の 1 対 1 対応の関係がある場合に基となる表の更新可能な列は更新可能とされるが、これはリレーショナル DBMS が機能識別子 T111「更新可能な結合、和集合及び列」の機能を提供する場合に限られる。T111 を提供しないと実際には更新はできないのに、ビューに更新可能という特性が付与されることは矛盾したことからである。このような事情で、問合せの更新可能性の規則の書き方が変更された。

■ 実効的に更新可能という用語の導入

ビューを定義した認可識別子にはビューを操作するための権限が生成される。認可識別子とは、表のような SQL の資源へのアクセスやその利用の認可として権限を与える対象を示すための名前

である。利用者識別子（ユーザ名）や役割名（ロール名）が認可識別子として使用される。ビューの定義者に対して `SELECT` 権限は無条件に生成される。

`UPDATE` 権限、`INSERT` 権限、`DELETE` 権限についてはどうか。更新可能でないビューに `UPDATE` 権限を付加しても意味がないので、これらの権限が生成されるかどうかは、ビューの更新可能性や挿入可能性に関わってくる。SQL:1999 で見たとおり更新可能性には単純に更新可能な場合と、SQL:1999 で新しく拡張された 1 対 1 対応に基づく更新可能性があり、権限の生成においていちいち 2 種類の更新可能性を並べて書く煩雑を避けるために、これらをまとめて表す用語が必要となった。そのために導入された用語が「実効的に更新可能」である。つまり、実効的に更新可能とは「単純に更新可能」である場合か、または「1 対 1 対応に基づいて更新可能」である場合ということになる。

これらの 2 つの変更により、ビューの更新可能性の規則は次のように改められた。

構文規則：（文献[10]より抜粋し翻訳。番号は原文のまま）

- 7) <問合せ式>が更新可能ならば、ビュー表は、更新可能 (`updatable`) とする。
- 8) <問合せ式>が単純に更新可能ならば、ビュー表は、単純に更新可能 (`simply updatable`) とする。
- 9) ビュー表が単純に更新可能か、又は SQL 処理系が機能 T111「更新可能な結合、和集合及び列」を提供し、かつ、ビュー表が更新可能ならば、ビュー表は、実効的に更新可能 (`effectively updatable`) とする。

5.3 SQL:2016

さて、SQL:2008 は SQL:2011 を経て SQL:2016[11]へと改正された。ビュー定義の形式は SQL:1999 や SQL:2008 のそれと同一である。ビューの更新可能性についての規則も基本的に同一である。SQL:2016 では、実効的に更新可能という用語の使用範囲が、ビューだけではなく<問合せ指定>の更新可能性に拡大された。また、「一般に更新可能」という用語が導入された。これらの用語は、箇条 7.16 <問合せ指定>の中で次のように定義される。

構文規則：（文献[11]より抜粋し翻訳。番号は原文のまま）

- 26) <問合せ指定>が潜在的に更新可能で、かつ、少なくとも一つの潜在的に更新可能な列をもつならば、その<問合せ指定>は、一般に更新可能 (`generally updatable`) とする。
- 28) <問合せ指定>`QS` は、`QS` が単純に更新可能か、又は SQL 処理系が機能 T111「更新可能な結合、和集合及び列」を提供し、かつ、`QS` が一般に更新可能ならば、実効的に更新可能 (`effectively updatable`) とする。

一般に更新可能という用語は何の形容もない更新可能という用語が多用されると規則の理解の妨げになるという理由で導入されたが、あまり必然性はないように思える。ただし、少なくとも一つの潜在的に更新可能な列をもつという条件付けが行われることで、相応に意味が特定のなるにもかかわらず、何の形容もない更新可

能という用語を使うことは、単純に更新可能等の他の用語との釣り合いの上で好ましくないと考えられる。

ビューの更新可能性については次のように簡潔な形に変更されている。

構文規則：(文献[11]より抜粋し翻訳。番号は原文のまま)

- 10) QE が一般に更新可能ならば、ビュー表は、一般に更新可能 (generally updatable) とする。
- 11) QE が単純に更新可能ならば、ビュー表は、単純に更新可能 (simply updatable) とする。
- 12) QE が実効的に更新可能ならば、ビュー表は、実効的に更新可能 (effectively updatable) とする。

5.4 SQL:2023

SQL の最新規格として SQL:2023 規格が 2023 年 6 月に公開された[12]。この改正は ISO/IEC 9075-16:2023 Information technology — Database languages SQL — Part 16: Property Graph Queries の新設のためであり、既存機能についての修正は不備の修正と軽微な制限の緩和だけにとどまっている。したがって、ビューに関連する修正も見当たらず、SQL:2023 はビューサポートに関しては SQL:2016 に同じということである。

6. SQL におけるビューの更新可能性—総括—

SQL-87 に始まり SQL:2023 に到るまで、ビューの更新可能性を規定するために、さまざまな概念、用語、そして記述が出現した。本章では、ややもすると断片の集合体の感を与えてしまいかねない規格の諸要素の意味・意図するところを咀嚼し伝えることで、改めて<問合せ指定>と<問合せ式>の更新可能性の理解向上の一助をしたい。

6.1 <問合せ指定>の更新可能性

<問合せ指定>という SQL 規格の構文要素は、問合せの原子的な部分であり一つの SELECT で指定される範囲を示している。

● 潜在的に更新可能な<問合せ指定>

<問合せ指定>が次の条件を満たすとき、その<問合せ指定>は潜在的に更新可能という。

- DISTINCT なし。
- 結果列に同じ列が一つ以上指定されない。
- グループ表でない。

この条件によって FROM 句に指定される問合せの基となる表の行と、問合せ結果の表の行が 1 対 1 に対応しなくなるような問合せは排除される。DISTINCT によって行の重複排除が行われると基となる表と問合せの行は一般的に多対 1 対応の関係になる。グループ化されて基となる表の行の集まりが 1 個のグループに対応付けられる場合も同様である。結果列に同じ列が 1 つ以上指定されないことは、問合せ結果の表の複数の列が基となる表の同じ列に対応付けられることを排除する。潜在的に更新可能とは、更新可能であるために最低限必要な性質があって、更新可能となる可能性があることと考えることができる。

● 潜在的に更新可能な列

潜在的に更新可能な<問合せ指定>の結果の列が次のいずれかの条件を満たすとき、その列は潜在的に更新可能という。

- <問合せ指定>の基となる表 (FROM 句に指定される表) がただ一つであり、その基となる表の更新可能な列に対応する列である。
- 基となる表はただ一つではないが、問合せ結果の表と行が 1 対 1 対応の関係になる基となる表の更新可能な列に対応する列である。

潜在的に更新可能な<問合せ指定>の基となる表が一つであれば行は 1 対 1 対応の関係になるため、これらの条件は総じて 1 対 1 対応の関係になる基となる表の列に対応する列であることが必要となる。そのような列であっても基となる表の列が演算結果の列であるような場合は更新可能ではないので、さらに基となる表の更新可能な列である条件が付加されている。

● 一般に更新可能な<問合せ指定>

<問合せ指定>が潜在的に更新可能であり、かつ少なくとも一つの問合せ結果列が潜在的に更新可能な場合、その<問合せ指定>は一般に更新可能という。<問合せ指定>が潜在的に更新可能であっても、問合せの結果列がすべて演算結果の列であるような場合は、基となる表の列に対応する列がなく潜在的に更新可能な列が一つもない。このような<問合せ指定>は一般に更新可能とはいえない。

● 単純に更新可能な<問合せ指定>

一般に更新可能な<問合せ指定>が次の条件を満たすとき、その<問合せ指定>は単純に更新可能という。

- <問合せ指定>の基となる表 (FROM 句に指定される表) がただ一つである。
- <問合せ指定>のすべての結果列が潜在的に更新可能。
- <問合せ指定>の WHERE 句に指定される条件の中に副問合せが指定される場合、<問合せ指定>の基となる表 (FROM 句に指定される表) は、副問合せで参照されない。

単純に更新可能な<問合せ指定>は、SQL-92 での更新可能な<問合せ指定>に相当する。WHERE 句の条件によって選択される行の集まりは、更新の実行前に決定されるべきものであるが、現実の実装では、副問合せで<問合せ指定>の基となる表を参照すると基となる表の更新の結果として、選択される行の集まりが変化してしまう場合があり、そのような状況が起きないようにするために、このようなことが要請されると考えられる。

● 実効的に更新可能な<問合せ指定>

実効的に更新可能という用語が導入された経緯については 5.2 で述べた。そこで述べたとおり、機能 T111 の提供なしでは実際の更新ができないので、機能 T111 の提供のない場合に、<問合せ指定>を更新可能とすることはできない。そのため、適合性上の機能を取り込んだ概念となっている。

<問合せ指定>は、次のいずれかの条件を満たせば実効的に更新可能という。

- <問合せ指定>が単純に更新可能である。
- <問合せ指定>が一般に更新可能であり、DBMS が機能 T111「更新可能な結合、和集合及び列」を提供する。

● <問合せ指定>の更新可能な列

<問合せ指定>の列は、次のいずれかの条件を満たせば更新可能な列という。

- 単純に更新可能な<問合せ指定>の列である。
- 一般に更新可能な<問合せ指定>の潜在的に更新可能な列であり、DBMS が機能 T111「更新可能な結合、和集合及び列」を提供する。

● 挿入可能な<問合せ指定>

<問合せ指定>は、実効的に更新可能であり、基となる表が挿入可能な場合に、その<問合せ指定>は挿入可能という。

実は、挿入可能ということについて、更新可能についてのように多くの議論がなされたわけではないように見受けられる。1対1対応の関係にならない場合でも、挿入ができる場合もあるかもしれない。ただ、そういう場合が原理的にあり得たとしても、その実装は非常に難しいと考えられるので、挿入可能を実効的に更新可能な場合としてしまうことは一つの現実解と考えられる。

6.2 <問合せ式>の更新可能性

<問合せ式>は複数の<問合せ指定>に対して集合演算を行う問合せも可能な、<問合せ指定>を包含する構文要素である。ただ一つの<問合せ指定>で構成される問合せも<問合せ式>として定義可能である。<問合せ式>の更新可能性は、4.1 で説明されている通りだが、これについても纏めておく。

● 単純に更新可能な<問合せ式>

<問合せ式>は、集合演算を含まず、再帰や結果行の順序付けの指定 (ORDER BY 句) 等を指定せず、<問合せ式>が含む<問合せ指定>が単純に更新可能な場合に単純に更新可能という。WITH 句 (インラインビューのようなもの) を指定する場合を除くと、実質的には、<問合せ式>が単純に更新可能な<問合せ指定>である場合ということになる。

● 一般に更新可能な<問合せ式>

<問合せ式>は次の条件を満たすとき、その<問合せ式>は一般に更新可能という。

- <問合せ式>に含まれる<問合せ指定>が一般に更新可能である。
- <問合せ式>が集合演算を含む場合、集合演算子は UNION ALL である。
- <問合せ式>が集合演算を含む場合、演算項の各<問合せ指定>の対応する結果列はすべて更新可能であるかまたはすべて更新可能でない。
- <問合せ式>が集合演算を含む場合、演算項の<問合せ指定>の末端で基となる表を演算項の間で共有しない。

● 実効的に更新可能な<問合せ式>

<問合せ指定>の場合に対応した概念になっている。<問合せ式>は、次のいずれかの条件を満たすとき、その<問合せ式>は実効的に更新可能という。

- <問合せ式>が単純に更新可能である。
- <問合せ式>が一般に更新可能であり、DBMS が機能 T111「更新可能な結合、和集合及び列」を提供する。

● <問合せ式>の更新可能な列

<問合せ式>の結果の列は、次のいずれかの条件を満たすとき、その<問合せ式>は更新可能な列という。

- <問合せ式>が集合演算を含まない場合は、<問合せ式>に含まれる<問合せ指定>の更新可能な列である。
- <問合せ式>が集合演算を含む場合、集合演算子は UNION ALL であり、演算項の各<問合せ指定>の対応する結果列はすべて更新可能であるかまたはすべて更新可能でなく、DBMS が機能 T111「更新可能な結合、和集合及び列」を提供する。

● 挿入可能な<問合せ式>

<問合せ式>は、挿入可能な<問合せ指定>であるときに挿入可能という。

集合演算があると挿入先がどの演算項の表かが一般には特定できないので、<問合せ式>が<問合せ指定>である場合というのは自然なことと考えられる。

6.3 ビューの更新可能性

ビューは、<ビュー定義>でキーワード AS に続いて指定される<問合せ式>が単純に更新可能、一般に更新可能、実効的に更新可能かに応じて、単純に更新可能、一般に更新可能、実効的に更新可能となる。<問合せ式>の更新可能な列に対応するビューの列が更新可能な列になる。

7. 適合性試験—その歴史と経緯—

SQL:1999 以降、適合性 (conformance) のために SQL 規格の機能全体が細かい機能区分に分類され、最低限適合が必要な部分 (中核 SQL) 以外は、適合性を主張する範囲を機能区分によって選択できるようになった。機能区分それぞれに機能識別子が割り当てられ、結合ビューや UNION ビューの更新可能性のために T111「更新可能な結合、和集合及び列」という機能識別子が割り当てられた。T111 が設けられた背景を次のように推察できる。つまり、ビューの更新可能性は改正を重ねるにつれて SQL の構文規則で定義するにはあまりにも複雑になりすぎて、これをプロプライエタリあるいは OSS のリレーショナル DBMS ベンダが忠実に実装することはほぼ無理ではないかとの認識となり、この解決策として、SQL としてビューサポートのための標準機能を標榜はするが、それをどのように実装しその結果がどれほど SQL 標準に適合しているかはベンダに任せる、という判断に至ったのだと考えられる。このスタンスはそれ以降現在に至るまで変わっていない。

さて、SQL:1999 で、適合性の主張のために SQL の全機能が細

分化されて、細分化された機能ごとに番号が付与され、各機能は中核 SQL (core SQL) とパッケージに分類されたものと先述した。そうすると、適合しているとの主張の正しさを一体誰がどのような立場で認めるのが問題となるが、これについては次のような経緯がある。

まず、SQL の適合性については SQL-92 の頃は認定機関による適合性試験 (conformance test) があった (SQL-92 では機能識別子ではなくて水準が 3 つあっただけである)。米国では NIST (National Institute of Standards and Technology) が適合性試験を行っていた。日本では JIS の適合性試験というものはないが、一部のマルチベンダシステムでは JIS 規格を基にした調達仕様が規定され、それに対する試験が行われることがあった。SQL:1999 の頃になるとメインフレームの時代が終わってマルチベンダ環境での厳格なソースプログラムの可搬性の要求が薄くなって適合性試験というものはなくなった。その後は、適合性は自己申告のような形になっている。この場合、「SQL:1999 に準拠しています」のような大雑把な言い方がされることもあるが、「T111 の機能を提供します」とまで細かいレベルでマニュアルのような公式文書に記載する場合は相応に厳格だと考えられる。第三者が認定するわけではなくあくまで自己申告ではあるが、実態を伴わないと組織の信頼性に関わることや、訴訟とまではいなくてもクレームの対象にはなりうるだろう。DBMS の評判にも関わるので影響は大きいと考えられる。機能を完全にサポートしていない場合は、大部分はサポートしていても「T111 の機能のサブセットを提供している」というようにより正確な表現がなされることもある。

8. おわりに

SQL におけるビューの更新可能性を規格の制定から始まりその改正の歴史と共に辿った。ビューの定義が<問合せ指定>から<問合せ式>に拡張されるに従いビューの更新可能性の定義は SQL の提供する多様な機能と関連してその複雑さは増加の一途を辿った。言うまでもなく、規格の難解さは多様なビューの更新可能性を享受しようとするクライアントを悩ませる一方で、そのような規格をフルスペックで実装しようとするリレーショナル DBMS ベンダの高い障壁ともなった。SQL 規格が実装に配慮して「適合性」という概念を取り入れたことはごく自然な流れであったといえよう。

今後、SQL のビューサポートがどのような展開を見せるのか予断は許さないが、本稿が SQL のビューサポートの経緯と現状を理解しておきたいと考えている者の一助となれば幸いである。

謝辞

末筆ながら、幾つかの有益なコメントを下された査読者に深謝する。

参考文献

[1] Edgar F. Codd. Recent Investigations in Relational Database Systems. Information Processing 74, pp. 1017-1021,

North-Holland, 1974.
[2] Michael Stonebraker, Eugene Wong, Peter Kreps, and Gerald Held. The Design and Implementation of INGRES. ACM Transactions on Database Systems, Vol. 1, No. 3, pp. 189-222, September 1976.
[3] ISO 9075 First edition, Information processing systems — Database language SQL, 1987-06-15.
[4] ISO/IEC 9075:1992, Information technology — Database languages — SQL, July 30, 1992.
[5] ISO/IEC 9075-2:1999, Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation), 1999-12.
[6] J. Melton and A.R. Simon. SQL:1999 Understanding Relational Language Components. Morgan Kaufman Publishers, 2002. 邦訳: ジム・メルトン, アラン・サイモン (著). 芝野耕司 (監訳), 小寺孝, 白鳥孝明, 田中章司郎, 土田正士, 山平耕作 (訳). SQL: 1999 リレーショナル言語詳細. ピアソン・エデュケーション, 2003.
[7] ISO/IEC 9075-2:2003, Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation), December 2003.
[8] 山平耕作, 小寺孝, 土田正士. SQL スーパーテキスト. 技術評論社, 2004.
[9] 土田正士, 小寺孝. SQL2003 ハンドブック—SQL 最新標準規格—. ソフト・リサーチ・センター, 2004.
[10] ISO/IEC 9075-2:2008, Information Technology — Database Languages — SQL — Part 2: Foundation (SQL/Foundation), 2008.
[11] ISO/IEC 9075-2:2016(E) Fifth edition, Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation), 2016.
[12] ISO/IEC 9075-2:2023, Information technology — Database languages SQL — Part 2: Foundation (SQL/Foundation), 2023-06.
[13] JIS X 3005-2:2015 データベース言語 SQL 第 2 部:基本機能 (SQL/Foundation). 日本規格協会, 2015.
[14] 増永良文. リレーショナルデータベース入門[第 3 版]. サイエンス社, 2017.