



## 解説

# SQLの20年と 現状および今後の展開 (前編)

東京外国語大学 芝野 耕司  
shibano@aa.tufs.ac.jp

## SQLの現在

ISO (国際標準化機構) でSQL言語の標準化を始めてから、20年が経った。当初、プログラム言語規格の1つとして開始されたデータベース言語の標準化は、その後、OSI (Open Systems Interconnection, 開放型システム間接続) の応用層に位置づけられ、開発を行い、ISO規格の中でも最も成功した規格の1つとなった。

最初にこの言語を開発したIBMでは、当初、Sequential English Query Language (順次英語問合せ言語) を略してSEQUEL (シーケル) と名付けていた。しかし、SEQUELがイギリスでネットワーク端末の商標として登録されていたことから、SEQUELをStructured Query Language (構造化問合せ言語) の頭文字をとって、SQLに変更し、SQLという名称が生まれた。SQLという名称は、IBMでの名称を継承したものではあるが、ISOでは、SQLを頭字語ではなく、固有名詞としている。しかし、一般には、エス・キュー・エルと呼ぶ以外に、シーケルと発音されることもある。

SQLの市場規模を正確に推定することは難しいことではある。しかし、少なく見積もっても、世界で数兆円以上の市場を形成しているであろう。SQLは、基幹ソフトウェアであるデータベース管理システム (DBMS) の実質上ただ1つの規格として、SQL専門メーカーであるOracleのみならず、IBMやMicrosoftにおいても、基幹製品として出荷され、大規模なシステムからMS Officeに含まれるデスクトップまで広く普及している。

また、基盤ソフトウェアであることから、多くのアプリケーション開発がSQLを前提として行われていること

も事実である。

プログラム言語の多くは、ANSI (米国規格協会) で開発され、ISOで承認を受けるのが普通であった。しかし、SQLを含むデータベース言語規格の開発では、ANSIがI (International) プロジェクトと位置づけたこともあって、ANSI内部での開発ではなく、ISOを開発主体とし、規格仕様を開発した。このことから日本もSQLの開発にあたっては、多くの本質的な貢献を行うことができたと自負している。

1980年代の日本は、アメリカに対抗し、最新鋭の大型計算機を開発できる唯一の国として、国際貢献を行うことができた。しかし、残念ながら、1990年頃には、20社以上の参加社を数え、多くの規格仕様を開発してきた国内委員会も、アメリカがメインフレームメーカーを主体とする委員会からソフトウェア会社を主体とする委員会に構造改革を遂げた一方、日本はこうした構造改革を実現できず、現在では、参加社は数社にとどまり、国際貢献も大きく減少したと言わざるを得ない。

本稿では、SQL言語の国際規格開発の主要な一角を担ってきた立場から、SQL言語の設計思想を整理し、同時に規格開発のあり方を考えてみよう。

## 前史

DBMSの標準化は、CODASYL (The Conference on Data Systems Languages) での検討に始まる。ISOでもCODASYLをもとに、ネットワークデータベース言語の標準化を行おうとしたが、この企ては、否定される。否定の原因は、Minority ReportとしてIBMの委員より指摘さ

れたその技術的欠陥にあった。

この否定を受けて、ISOではCODASYLとは別の新しいネットワークデータベース言語NDL (Network Database Language) の開発が25年ほど前に開始されることとなる。

NDLの開発に加えて、ANSIでは、リレーショナルデータベース言語の標準化の検討が始まる。検討グループは、リレーショナルデータベース言語の規格開発を勧告するが、具体的な構文ではなく、抽象的なインタフェースを規定すべきであると勧告する。当時は、OSIの開発でも同様であるが、IBMの圧倒的な市場支配力に対抗することが求められたことによる。

そして、ネットワークデータベース言語NDLに加えて、IBMの開発したSQLではなく、リレーショナルデータベース言語RDL (Relational Database Language) の開発が始まった。

ネットワークデータベース言語とは、親子集合型を基本とするネットワークデータモデルに基づくデータベース言語であり、リレーショナルデータモデルに基づくデータベース言語がリレーショナルデータベース言語である。また、オブジェクト指向データモデルに基づくのが、オブジェクト指向データベース言語になる。すなわち、データベース言語は、何らかのデータモデルに基づく言語である。

## 基本リレーショナルデータベース： SQL-87とSQL-89

NDLとRDLでは、当時、IBM ResearchでJim Grayらが研究していた1つのDBMSカーネル上に複数のインタフェースを実装しようとするアーキテクチャと同様のアーキテクチャを採用した。IBM Researchでは、この複数のインタフェースにIMSの階層型インタフェースとSQLのリレーショナルインタフェースとを実装することを行っていたが、ISOでは、NDLとRDLとを同じように位置づけ、仕様開発を開始した。

RDLは、ANSIの勧告に従えば、具体的な言語構文を規定することはできず、抽象インタフェースだけの規定になるはずであった。これに対して、当時のエディタであったIBMのPhil Shawは、モジュール言語を提案する。このモジュール言語というインタフェースを介することによって、“抽象インタフェースを規定する”という勧告に反しないとされた。

データベース言語仕様では、データモデルを実現することが最も重要である。データモデルは、(1) データ構造を規定し、(2) そのデータ構造に対する操作および (3) データの正しさを保証する整合性を規定する。

最初のSQL言語規格も、リレーショナルデータモデルをサポートする言語仕様の開発を目指した。すなわち、表を定義し (CREATE TABLE)、表の列を取り出し (SELECT)、表の横の結合 (FROM)、表の縦の結合 (UNION) を実現し、リレーショナルデータモデルの機能を取り込んだ。同時に、データの挿入 (INSERT)、削除 (DELETE)、更新 (UPDATE) および取り出し (SELECT) を可能とすることによって、データベース言語機能を実現した。しかし、ここでは、上記 (3) の整合性が欠けている。

RDLは、ISOでの委員会素案 (CD) 投票の時点で、イギリスからのコメントによって、実際の製品であるSQLに名称を変更するとともに、規格仕様自体についても、実装され、安定した機能に限定し、規格自体を早く出版することとなった。

そして、最初の規格は、ANSIでは1986年10月に承認され、ISOでも同時期に承認されたが、最終校正は1986年末休みに著者が行い、1987年正月休み明けにエディタのPhil Shawに変更箇所をメールし、ISO、ANSIとも同一のテキストを1987年2月に出版した。このため、承認時点をとってSQL-86、出版時点をとってSQL-87<sup>1)</sup>と2つの呼び名が生まれた。

RDLからSQLへの変更で最も大きなものは、整合性制約の削除である。整合性制約は、データモデルの本質的な要素ではあっても、当時は実装がなく、必要とする機能の詳細が未分明であった。このため、整合性機能を補遺1として出すこととし、規格の出版を急いだ。また、この整合性機能以外にDBMSで必要とする機能については、補遺2として並行して開発を進めることとした。

この整合性機能拡張の補遺は、最小限必要となる参照整合性の機能仕様を開発し、最終的には、補遺の形式ではなく、SQL-89<sup>2)</sup>として出版した。そして、補遺2は、その後、SQL-92<sup>3)</sup>となる。

SQL-87およびSQL-89を一言で要約すると、リレーショナルモデルをサポートする最小システムと位置づけることができよう。

## リレーショナルデータベース管理システム： SQL-92

SQL-87開発時点でも、OracleやIBMのSQL/DSなどのいくつかの商用SQL DBMS (データベース管理システム) が存在した。これらのシステムでは、リレーショナルデータモデルを実現するだけでなく、最初のSQL DBMSであるIBMの研究部門で開発したSystem-R以来の多くの機能を持っていた。

これらの機能には、データベースでスキーマと呼ぶデータ構造定義の変更機能であるスキーマ操作、トラン

ザクシオン・セッション・コネクション管理, 診断機能, 動的SQL, 情報スキーマなどの機能がある。また, 日本語を含む世界の言語への対応機能やリレーショナルモデルサポートの拡張があった。

スキーマ操作や動的SQL, 情報スキーマなどの機能は, データベース研究や実装が先行していた。しかし, 日本語機能を含む各国語サポートや整合性機能の多くは, 十分な先行研究が存在せず, 国際規格の開発が最先端の研究を担うこととなった。

こうした仕様開発の結果として, SQL-92は, 最先端でかつリレーショナル商用DBMS全体のための仕様としての1つの完成形を提示することとなった。しかし, SQL-92は, 1つの大きな問題を孕んでいた。

SQL-92には, 仕様全部を実装する以外に, 初級, 中級の2つの水準が設けられている。SQL-92の出版から10年以上を経過した現在でも, 実際の製品での実装は, 初級水準を基本とし, 中級水準の一部を含むのが一般的であり, SQL-92の仕様全体の実装は存在しない。この点で, SQL規格は, 現在に至るまで, 実装されない多くの仕様を含むこととなるとともに, 各製品固有の方言も残されることとなった。

どうしてこのようなことになったのであろうか? これらの実装されない多くの機能は, SQLへの批判に応えるかたちで開発された。こうした批判には, 簡単に説明すると, 言語要素がどこでも使えることを要求する言語の直交性やより高度な整合性制約機能などが含まれる。批判の多くは理論的には正しいと思われることが多いが実際には必要としないものも多くある。そして, 批判に応えるかたちで, 理論的な可能性を追求することによって, 10年たってもまったく実装されない機能の規格仕様を開発することとなった。

## 新しいニーズへの対応: SQL/CLIとSQL/PSM

SQL-87の制定によって, 雨後の竹の子のように多くの商用製品が市場に現れ, 最盛期には, 50以上の製品が出現することとなった。また, 当初は, 情報系のアプリケーションだけしか利用できない, と一部でいわれてきたSQL DBMSが基幹のトランザクション処理に利用可能であることが実証されるとともに, 応用分野も急激な広がりを見せた。

このことは, 同時にこれまでのデータベース研究では知られなかった多くの新しい応用分野を生んだ。それがSQL/CLI (Call Level Interface) 開発の発端である。

表計算ソフトとして代表的なソフトウェアに育ったExcelで, プリンタドライバの開発の延長として, SQL

DBMSを呼び出すためのインタフェースがMicrosoftで生まれた。業務用システムとしての表計算ソフトからDBMSを直接呼び出し, この問合せ結果を加工しての分析や集計, 印刷などに利用したいというのは, 1つの当然の帰結であろう。一種のColumbusの卵であろう。これがODBC (Open Database Connectivity) として, Microsoftで開発され, その後, 名称をインタフェースの特性に対応するCLI (Call Level Interface) と変更することとなった。すなわち, SQL/CLI<sup>4)</sup>の起源である。

一方, CLIは, 従来のSQLがプログラム言語規格以来引きずっていたアプリケーションプログラムのソースレベルでの可搬性の実現という標準化の位置付けを変えることになった。すなわち, プログラムソースをコンパイルすれば, どのシステムでも使えるというのではなく, シュリンクラップされたパッケージソフトを呼び出し, CALL LevelのAPIインタフェースを規定することによって, 新しい相互運用を可能とすることとなった。

また, データベース言語は, CODASYL以来データベースサブ言語として, COBOLやFORTRANなどのホスト言語のデータベースアクセス機能を拡張するものとしてとらえられてきた。SQL言語の計算完備な言語への拡張, すなわち, 変数やサブルーチン, IF文やLOOP文などの実行順序制御を可能とする言語機能をSQLの拡張に取り入れることを1992年のアルル会議で提案した。この提案の意図は, DBMSの並行実行性能を向上させるためには, 2相ロックを用いるだけではなく, トランザクションスケジューリングなどを組み込むことを可能にする必要がある, こうしたDBMSの性能向上には, データアクセスだけではなく, トランザクションやアプリケーション論理の全体をDBMSが知る必要があった。このためには, SQLだけでプログラムを書く必要があった。このために提案したのが, SQL/PSM (Persistent Stored Module)<sup>5)</sup>である。

SQL/PSMの必要性は, 商用システムでは明らかであり, OracleはPL/SQL, Sybaseは, Transact SQLとして実用化していた。そのため, PSMの開発では, この2つの商用システムとの位置付けが大きな問題となった。

## オブジェクトパラダイムへの対応: SQL-99

1990年代半ばになると, 50社以上を数えたSQLベンダも吸収合併や買収, 撤退によって, 急速に寡占化が進む。一方, メインフレームの基本ソフトとして開発されてきたDBMSの主力開発が新しいソフトウェア市場の拡大によって, メインフレームの付属物としてのDBMSから独立したソフトウェア製品として, パソコンからメイ

ンフレームまでをプラットフォームとするように変わってきた。同時に、標準化の主体がハードウェアベンダからソフトウェアベンダに変わってきたことを意味する。

ANSIでは、IBM、UNIVAC、DECなどのハードウェアベンダは、合併や撤退などによって、数が減少し、一方、新たにソフトウェア企業を中心に躍り出てきたOracle、Informix、SybaseなどのSQL専門ソフトウェアベンダが研究開発の主体となった。残念ながら日本ではこのソフトウェアベンダの台頭現象はみられず、ハードウェアベンダの退場は、すなわち標準を含め、研究開発能力の衰退となって現れた。

一方、学会での研究開発の中心課題としてオブジェクト指向が注目されるとともに、データベースに関しても、オブジェクト指向データベースが研究開発の主要課題となっていた。SQLについても、別物ではあり得なかった。

同時に、SQL/PSMの開発によって、オブジェクト指向をSQLだけで実現できる基盤が存在した。この基盤をもとに、オブジェクト指向データベース機能をSQLに取り入れる開発が始まった。

一方、百数十ページの仕様で始まったSQL仕様は、SQL-92では、600ページを超え、SQL-99では、2000ページ近くなることが予想された。このことから、SQL仕様の見直しが必要となった。

日本からは、Javaなどに見られるように、言語の核仕様とライブラリ仕様とを分離し、抜本的にSQL言語仕様を見直すことを提案したが、この提案は受け入れられず、規格を複数パートに分割するにとどまった。

結果として、SQL-99は、オブジェクトリレーショナルデータベース管理システムの巨大な仕様となった。

SQLのパートとして開発してきたパートの中には、開発中止となったものもある。学会レベルの研究として注目され、ACMに特別に仕様開発委員会が設けられたSQL/Temporalは、この典型であろう。SQL/Temporalは、ACM仕様のTSQL (Temporal SQL) とUKが押し出したIXSQL (Interval eXtended SQL) が議論を呼び、かなりの時間を開発に費やしたが、結局は、SQL/Temporalの開発を中止した。SQL/Temporalが中止になったのは、時制論理という理論的興味だけで、実用化の視点と実装者の興味をひかず、実装が進まなかったことによる。

SQL/DTP (Distributed Transaction Processing) の場合は、TPモニタでの実現が一般的であったために開発を中止した。SQL/Replicaの場合は、ラップトップなどへのデータベースの内容の持ち出しよりも常時接続へとDBMSを取り巻く環境の変化が開発中止の理由である。

## SQL標準化過程での組織変革と日本の対応状況

1970年代にIBMによって発表され、1980年に最初の商用SQLがOracleによって出荷され、続いてIBMからもSQL/DS、DB2が出荷され、SQL市場が生まれた。

SQL-87が制定される頃には、市場への期待から多くの企業が参入し、SQL Accessグループなどのコンソーシアムが生まれる。一方、1990年代になると寡占化が進み、コンソーシアムや公的標準化も様相を変えることになる。1990年代後半には、コンソーシアムとはいっても、寡占企業同士の一種談合的な色彩が強まり、必ずしもオープンなコンソーシアムではなくなることになる。

開発主体もハードウェアベンダのCost and Fee部門にとどまった開発は徐々に撤退することになり、Profit部門である専門メーカーに主体を移した。こうした構造変化が日本でなぜ起こらなかったかを考える必要がある。

一方、利用分野および利用者は、広がり続け、一般の事務部門でも事務員が日常的にSQLを利用することになった。規格開発においても、古く妥当性を失った研究課題ではなく、新しいニーズへの対応を中心に開発を継続している。日本においても、他の言語に先駆けて、日本からの提案を行った日本語機能を含む各国語機能の開発を始め、今後とも、日本からのアプリケーションニーズをSQL仕様に反映させる必要があり、そのためには、現在のハードウェアベンダ主体の情報分野での国際標準化の体制を考え直すことが必要であろう。

### 参考文献

- 1) ISO/IEC 9075 ;1987 Database Language SQL
- 2) ISO/IEC 9075 ;1989 Database Language SQL
- 3) ISO/IEC 9075 ;1992 Database Language SQL
- 4) ISO/IEC 9075-3 ;1995 Database Language SQL/CLI
- 5) ISO/IEC 9075-4 ;1996 Database Language SQL/PSM

(平成16年3月30日受付)

