



解説

SQLの20年と 現状および今後の展開 (後編)

(株)日立製作所 **土田 正士**
mastsuch@itg.hitachi.co.jp

(株)日立製作所 **小寺 孝**
kotera_t@itg.hitachi.co.jp

東京外国語大学 **芝野 耕司**
shibano@aa.tufs.ac.jp

SQLの開発経緯

データベース言語SQLは、1987年の初版制定から、1992年版 (RDB完成形の追求および新たなニーズへの対応)、1999年版 (オブジェクトパラダイムへの対応) と改訂されてきており^{1), 2)}、現在は2003年版 (Java™/OLAP/XML連携、外部データ連携) が最新であり、ISO/IEC SC32/WG3で次期2007年版を目指し開発に着手したところである (表-1を参照)。2003年版 (SQL2003と略記) 仕様開発の特徴は、1999年版 (SQL99と略記) までと異なり、応用指向の開発にシフトしたところにある。また、SQL99の開発に7年も費やした反省を踏まえ、産業界に公約している2003年中の出版を目指す時間枠を守り、しかも仕様の安定性も確保するという半ば相反する要件を、いかに限られた時間で実現するかも最大の懸念であった。仕様の検討、提案書の開発、相互のレビューを通し、各国代表との共同作業によって、これらの要件を満たすべくSQL2003の開発が行われた。これらの要件が十分に満足されているのかについて日本から懸念を表明し、安定性と時間枠に関する議論を行う場面もあったが、最終的には約3,500ページからなるSQL2003の規格化が合意された。

応用指向の観点でのSQL概要

◆オブジェクト・リレーショナルデータベースとしてのSQL

SQL2003では、時代の要請に従って、オブジェクト指向の拡張、マルチメディアデータの格納基盤の提供、さらなるリレーショナル機能の強化を目指している^{4), 5), 6), 9)}。このような狙いでSQL2003の機能の位置付けと概要を示す。

オブジェクト拡張機能

SQL99では、オブジェクト指向の考え方を取り入れ、しかも既存のSQLデータモデルと親和性の高い方法でオブジェクトリレーショナル技術を導入した。

SQLデータベースの上にオブジェクト指向機能を実現するために利用者定義型が導入された。利用者定義型には、既定義型と同じ値の表現を用いる個別型と、値を属性によって構成する構造型がある。既定義型に対する操作があらかじめSQL言語の中で用意されるのとは異なり、利用者定義型では値の振る舞いを規定する操作を利用者が定義できる。

これらの操作はメソッドによって行われる。メソッドのインターフェースは利用者定義型の中で宣言される。メ

Java™およびすべてのJava™関連の商標およびロゴは米国およびその他の国における米国Sun Microsystems, Inc.の商標または登録商標です。

	SQL開発経緯	関連動向
1970		E.F.Codd氏がリレーショナルモデル提唱(CACM誌)
1973		IBM社が“System R”開発プロジェクトを開始
1979		最初のRDBMS製品“ORACLE V2”(現ORACLE社)を発表
1981		IBM社が“SQL/DS”を発表
1982	ANSIがリレーショナル言語制定作業を開始	
1983		
1984		リレーショナル基本機能の確立
1985		
1986		
1987	ISO 9075:1987 制定 (SQL-87)	JIS X3005:1987 制定 (SQL-87)
1988		
1989	ISO 9075:1989 および整合性機能拡張制定 (SQL-89)	
1990		JIS X3005:1990 および整合性機能拡張制定 (SQL-89)
1991		
1992	ISO 9075:1992 制定 (SQL-92)	RDB 完成形の追求および 新たなニーズへの対応
1993		
1994		
1995	ISO 9075-3:1995 制定 (SQL/CLI)	JIS X3005:1995 制定 (SQL-92)
1996	ISO 9075-4:1996 制定 (SQL/PSM)	JIS X3005-3:1996 制定 (SQL/CLI)
1997		
1998		JIS X3005-4:1998 制定 (SQL/PSM)
1999	ISO 9075-1,2,3,4,5:1999 制定 (SQL-99)	オブジェクトパラダイムへの対応
2000	ISO 9075-10:2000 制定 (SQL/OLB)	
2001	ISO 9075-9:2001 および追補 1 制定 (SQL/MED&OLAP)	
2002	ISO 9075-13:2002 制定 (SQL/JRT)	JIS X3005-1,2,3,4:2002 制定 (SQL-99)
2003	ISO 9075-1,2,3,4,9,10,11,13,14:2003 制定 (SQL-2003)	JIS X3005-9,10:2003 および追補 1 制定 (SQL/MED&OLB&OLAP)
2004	情報統合環境の構築基盤	JIS X3005-14:2004 制定予定 (SQL/XML)

注) SQL/CLI(Call-Level Interface)
 SQL/PSM (Persistent Stored Modules)
 SQL/OLAP (On-Line Analytical Processing)
 SQL/MED(Management of External Data)
 SQL/OLB(Object Language Bindings)
 SQL/JRT(Routines and Types Using the Java Programming Language)
 SQL/XML(XML-Related Specifications)

表-1 SQL開発経緯と関連動向

ソッドの実装は利用者定義型の定義とは別にメソッド定義によって行われる。構造型は他の構造型を継承することができる。上位型のメソッドを継承せずに下位型のメソッドとして再定義することもできる。

さらに、オブジェクト指向データベースと同様に、識別性を持つオブジェクトの集合として扱うために、行をオブジェクトと同一視することによって、SQLデータモデルにオブジェクト指向の考え方を融合させている。これは行を、オブジェクトとしてとらえた利用者定義型の値として扱うということであり、表を型付けすることを意味する。その表を特に型付き表と呼ぶ。型付き表にも継承の考え方が適用されている。

マルチメディア拡張機能

マルチメディアデータは膨大であり、その内容はテキストを中心として静止画像、動画像、音声等により構成される。これらの長大なデータを格納できる長大オブジェクト (LOB) 型が追加された。プログラムの中で実際に操作するデータ量を削減するために、実データではなくSQLサーバ内のLOBデータの識別情報によってデータ操作を行うための位置付け子 (ロケータ) の機能が提供されている。LOB型とオブジェクト指向の機能と組み合わせ、XMLデータへの検索機能を始めとしてマルチメディアデータの統一かつ拡張可能なデータ管理システムが構築可能となる。

リレーショナル機能およびその実行基盤機能の拡張

SQL99とSQL2003において、SQL92で拡充したリレーショナル機能にさらなる拡張が行われた。これらは概して、より多様なデータを扱うためのデータ型の拡張、アプリケーションの生産効率と保守性をさらに向上させるためのSQLデータ操作とデータベース動作の拡張およびSQLの実行基盤となるトランザクション機能と安全保護機能の拡充に大別できる。

• 新たなデータ型 (集まり型, 行型, ブール型) の導入

集まり型は同じデータ型の値の集団を1つの値として扱うためのデータ型である。集まり型には要素番号を持つ要素で構成される配列型と、要素の識別番号と順序性を持たないマルチ集合型がある。行型は行構造を持つ値を扱うためのデータ型である。行型は行の列に相当するフィールドで構成される。

集まり型も行型も、リレーショナルデータモデルを拡張したとも考えられるが、非正規形データ構造とも考えられる。データ構造とそれに対して可能なデータ操作の性格によっては、繰り返し項目を行に分解したり、複合項目を列に分解せずにひとかたまりのデータとして扱うことが望ましい場合もあり、これらのデータ型を使用することによって素分解しないデータの扱いも可能となる。これは正規化設計法と対立する機能というより、正規化設計での値のとらえ方を拡張した機能と考えた方がよいであろう。

ブール型は真偽値を扱うためのデータ型である。探索条件を値式として扱うことが可能となる。ブール値を戻り値とする利用者定義関数を定義することによって、条件評価の方法を利用者が定義することができる。また、条件評価の結果が評価対象行のデータだけによって決まるような条件評価の結果を、行の挿入あるいは更新時にブール型の列に格納しておくことで、問合せの際の条件評価を省略することもできる。

• 再帰的解法の記述：再帰問合せ機能

SQL操作の拡張として再帰問合せの機能が導入された。再帰問合せとは、問合せの結果表を同じ問合せの入力表として評価することを、再帰的に繰り返して問合せの結果を求める機能である。自己参照の他に、循環参照する複数の問合せによって1つの問合せを形成することもできる。従来はアプリケーション手続きの論理の中で実現するしかなかった部品展開や経路探索の問題を、再帰問合せによって1つの問合せ文で実現することが可能となる。

• 事象動作の記述：トリガ機能

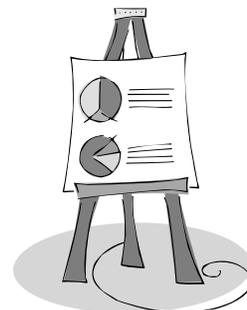
データベースに対する事象の発生を契機として、指定された動作を行う機能としてトリガが導入された。トリガによってデータに課されるルールをアプリケーション側で記述するのではなく、データベース手続きとして実現できる。業務にかかわる論理とデータそのものにかかわる論理を、アプリケーション側と、データベース側に分離することによって、より高い設計上の独立性が実現できる。トリガの動作には整合性制約動作とは異なり、一般的な手続きの記述が可能である。トリガの用途として、データのルールの実現だけではなく、履歴情報の蓄積や集計値の算出などの用途も想定される。

• 順序番号の生成機能

順序番号を割り当てる機能として、シーケンス生成子がある。データを識別するための番号を割り当てることは広く行われているが、アプリケーションが順序番号を管理することには難点がある。具体的には、トランザクション一貫性制御の対象となる表の列によって順序番号を割り当てると、1つのトランザクションがその列へのアクセスを占有するという問題が一般的に発生する。そのような問題を回避するため、非SQL資源によって順序番号を管理することも行われてきた。このような順序番号の割り当てをSQL操作によって可能にするために、トランザクション一貫性の対象とならないSQL資源としてシーケンス生成子が導入された。

• トランザクション機能の拡張

トランザクション機能としては、データ操作の取り消しの範囲を任意に設定する保存点 (セーブポイント) の機能が導入された。試行錯誤的な処理において試行に失敗するたびにトランザクション全体を取り消して再実行を行うことは非効率である。真に再実行が必要な実行点に保存点を設定し、データ操作の取り消しの範囲を局所化することによって、無駄な再実行処理を抑制することができる。



• 安全保護機能

安全保護機能に関して、データベース上での役割(ロール)に基づく権限管理の機能が導入された。従来は、SQL資源に対する権限の管理は、利用者識別子ごとに行われていた。利用者単位に権限を管理すると、利用者の数が多いと権限管理が煩雑になる。本来、利用者識別子は利用者として正当性の認証のために使用される。これを権限検査のためにも流用することに煩雑さの原因があったと考えられる。

役割に名前をつけて役割名に基づいて権限管理を行うことにより、利用者を役割によってまとめることができ、権限管理は単純化される。また、役割のみに基づく権限管理を行うと、個別の業務の遂行においてその業務のために担う役割の権限だけが行使でき、行使できる権限を個別の業務内に局所化できる。

◆データウェアハウス用途：SQL/OLAP (On-Line Analytical Processing)

一般的にOLAPのデータ構造は、集計の軸をさまざまに組み合わせた多次元構造をなし、多次元空間の格子に集計値を格納する。SQLの集計機能を拡張することで可能な分析業務も多く、SQLデータベースとして一元化できる点は好ましいことである⁴⁾。

基本OLAP機能

集計の軸としてのグループ化列の集合を組み合わせ、ドリルダウン集計(ROLLUP操作)やクロスタブ集計(CUBE操作)が可能である。一般的に、次元を追加することによって集計範囲をより詳細化する操作をドリルダウンと言い、たとえば店舗ごとの集計に対して、さらに商品ごとの内訳を求める操作に相当する。ROLLUP操作は、ドリルダウンとは逆に要約化することに相当する。このROLLUP操作によるグループ化の結果集合を利用して、ドリルダウン集計の表示に必要なデータがROLLUP操作で提供される。また、グループ化列のあらゆる組合せに基づき、クロスタブ集計の表示に必要なデータがCUBE操作で提供される。

高度なOLAP機能

多次元データ処理では、たとえば商品の購入金額に基づき上位10%の顧客が、売上高全体の何%を占めるかを確認して、客層に応じた新商品のプロモーションコストの配分を決め、その戦略に基づいて売上推移や各種統計量で効果を計ることが必要となる。

これらの用途のために、ウィンドウ操作が導入された。ウィンドウ操作によって、集計対象の全体を1つ以上のウィンドウ区画に分割し、各区画内で順序付けを指定し

た結果であるウィンドウ枠に対して、どの部分を集計の対象にするかを指定する。集計処理として、順序リストの相対位置を返却する順位関数、累計相対度数や累計相対順位を返却する分布関数などが指定できる。また、分散、標準偏差、共分散および相関係数からなる統計関数や線形回帰関数が導入されている。さらに、逆分布関数や仮説集合関数も用意され、高度なOLAP機能が完備されている。

SQL2003による情報統合環境の構築基盤

最も古くは、1970年代にIBMがシステム370の次世代システムとしてFS(Future System)を検討した際、現在のAS/400につながる基本ソフトの根幹にDBMS機能を含めることが検討された。

一方、現在ではWebシステムのバックエンドとして、DBMSの利用が広がり、DBMSを通常のOSファイルと一緒に利用したいという要望が顕著になってきた。こうした要望に応えたのがSQL/MED(Management of External Data)である⁷⁾。

また、1995年にJavaが現れたとき、この重要性から日本はSQL/Javaを開発することを提案した。当時、この提案は、受け入れられなかった。しかし、時代が変わり、Javaが普及するとともに、主要SQLベンダでSQLJグループが結成され、JavaとSQLの結合が大きな課題となり、SQL/OLB(Object Language Bindings)およびSQL/JRT(Routines and Types Using the Java Programming Language)の開発を行うこととなった^{8), 10)}。

さらに、Webシステムのバックエンドとして広く利用されることになったSQLは、Web技術の変化にも対応が必要であった。このためISOでXML関連の開発を開始するとともに、SQLXグループが結成され、SQL/XMLの標準化が進行している¹¹⁾。

◆外部データ連携による情報統合基盤の提供：SQL/MED

SQLデータベースの外部にあるデータ(SQLデータベース以外の順次ファイルや階層型データベースなども含む)に、SQLでアクセスするための規格として位置付けられる。SQL/MEDでは外部データ(非SQLデータ)をSQLインタフェースでアクセスするためにDATA LINK型(外部データの格納位置などのファイル参照情報を格納する型)と外部表(外部データを外部データ覆い(ラッパー)を介してアクセスし、表データとして扱うための擬似的な表)の2つの機能を規定する。

DATALINK 型

すでにファイルシステムなどアプリケーション固有のインタフェースを介してデータ管理を行っている画像情報、文書情報などのデジタルコンテンツをSQLデータベースに格納せずに連携させることを狙いとする。SQLデータベースには、デジタルコンテンツの書誌情報が格納され、それら書誌情報とデジタルコンテンツはDATALINK型の列でリンクされている。このリンクには、HTTPあるいはFTPなどのオープンなプロトコルが想定されており、Web環境上でDATALINK型が適用できるようになっている。これによって、参照制約、回復、バックアップなどのデータベース運用操作と連携されることとなる。また、アプリケーションで管理されるデジタルコンテンツもSQLデータベースのアクセス権限管理の対象となり、データ操作が可能となる。

外部表

外部表は、SQLサーバの管理外の外部ファイル、非SQLデータベース、あるいはSQLデータベース中の表から構成される。特に、外部表がSQLデータベースだけから構成される場合、連邦型データベースと呼ばれることもある。基本的には、異種あるいは同種なデータ源をSQLのAPIで統合化させてアクセスするシンプルかつ効果的なアーキテクチャを提供することが狙いである。データそのものは分散していても、データの管理は集中へと向かう動きと合致している。これによって、種々のデータ源のデータ項目、データ表現の差異を吸収し、統一的なデータ操作、検索機能が適用できることが期待されている。

また、外部のデータ源とは、外部データ覆いを介してアクセスされる。この外部データ覆いは外部データ源を実体化する外部サーバと関連付けされている。SQLサーバと外部サーバとの間のAPIを外部データ覆いインタフェースルーチン群として規定することによって、アプリケーションから発行されるSQL文でアクセス対象となる表が、ローカルなSQLデータベースに存在するのかわ、外部のデータ源であるのかを意識させずに透過性を実現している。

外部表はさまざまな外部データ源へのアクセスを可能とする。主な用途としては、ETL (Extraction, Transformation, and Loading) 機能、連邦型データアクセス機能が考えられている。

◆アプリケーション開発環境の整備：SQL/OLBおよびSQL/JRT

埋込み構文仕様：SQL/OLB

Javaを通してSQLデータベースにアクセスすることは

当然の要求であり、そのためのCLI (Call-Level Interface) としてJDBCがある。CLIには動的にSQLを組み立てて実行できるという高い柔軟性がある反面、SQL文の準備、入出力情報の記述子の取得と設定、実行という手順を踏む必要がある。実行するSQL文が定型的で、動的に組み立てる必要がない場合には、より単純な記述でSQL文を実行することができる埋込みインタフェースが望まれる。SQL/OLBはJavaプログラムの中にSQLを直接記述するための埋込み構文と必要な機能を規定する。

SQL/OLBのSQL埋込みJavaプログラムの仕様は、当初SQLJグループで開発されANSIでの審議を経てISOに提案された。SQL/OLBはISOで他のホスト言語のためのSQL埋込み言語の仕様記述と整合させる作業が行われた後、国際規格となった。が、SQL埋込みJavaプログラムには他のホスト言語のSQL埋込みプログラムにはない特長もある。

SQL埋込みJavaプログラムの埋込み構文は、Javaプログラムのランタイムメソッドの呼び出しに置き換えられ、SQLの記述情報はプロファイルクラスのオブジェクトとして表現される。これらのインタフェースをパッケージとして規定することは、Javaの持つ実行コードレベルの可搬性を実現するために必要となる。これはソースレベルの可搬性を目指す他のホスト言語とは異なる。

また、他のホスト言語のSQL埋込みプログラムは動的SQLのための構文を持つが、SQL埋込みJavaプログラムは動的SQLのための構文は持たない。SQL埋込みJavaプログラムで動的SQLを使用する場合には、JDBCインタフェースを使用する。さらに、JDBCインタフェースと埋込みインタフェースの間で接続オブジェクトを相互に利用することも可能である。また、例外処理もJDBCで使用されるjava.sql.SQLExceptionクラスが使用される。

さらに、変数しか埋め込むことができない他のホスト言語のSQL埋込みプログラムと異なり、SQL埋込みJavaプログラムではSQLに入力値を与えるためにJava式を埋め込むことが可能であり、拡張されている。他のホスト言語のSQL埋込みプログラムでは、問合せの結果集合から値を取り出すためにカーソルが使用される。一方、SQL埋込みJavaプログラムではオブジェクト指向インタフェースでカーソルに相当する機能を実現するための反復子を使用する。

ルーチンおよびデータ型マッピング仕様：SQL/JRT

SQL/JRTでは、JavaルーチンおよびJavaクラスに構造型を直接対応付ける外部Javaデータ型の機能が導入された。SQL/JRTの仕様も、当初SQLJグループで開発されANSIでの審議を経てISOに提案された。

Javaルーチンは、Javaで記述される外部ルーチンであ

る。Javaルーチンとして参照されるJavaプログラムはJavaメソッドとして記述される。Javaルーチンを実行するためには、SQL環境の中にJavaクラスをロードしてそのメソッドを実行する必要がある。そのためにJavaクラスをSQL環境のカatalogに登録し、Javaクラスの所在の探索パスを定義するためのいくつかの組込みプロシジャが提供される。

外部Javaデータ型はJavaクラスに直接対応付けられた構造型である。外部Javaデータ型の属性はJavaクラスのフィールドに、外部Javaデータ型のメソッドはJavaクラスのメソッドに対応付けられる。外部Javaデータ型に対応付けるJavaクラスは、構造型とJavaクラス間のインスタンスの変換を行うためにjava.sql.SQLDataかまたはjava.io.Serializable インタフェースを実装する必要がある。

◆SQL/XMLによるXML連携

XMLがビジネス世界の注目を集め、SQLとXMLが相互運用しなければならない局面が多くなってきている。日々刻々増加し続けている世界中のビジネスデータがSQLデータベースシステムに格納され、一方XMLがWeb環境およびお互いに意識なくともアプリケーション間でデータを交換するためきわめて重要なツールの共通語になりつつある。

このような動きに対応して、アプリケーションの中でSQLおよびXMLを共存させる機能を積極的に追い求めているSQL/XMLと呼ぶ新たな開発作業が開始された。SQL/XMLは2つの言語間の相互作用に関連する仕様を規定している。SQL/XMLは当初、SQLの表データをXML形式に変換する機能(データ交換や帳票作成が主要な目的)がその主な内容であったが、SQLデータベースの中にXMLデータを格納する機能も取り入れられた。

また、現時点ではW3Cで議論されているXQueryが有力な候補であろうが、SQL/XMLの問合せ機能については規定されていない¹²⁾。読み出し機能しか持たず、データ操作機能が完備していないXQueryは、データ管理業務の構築に必須な機能面、運用面が提供されない課題がある。XMLがデータ管理基盤の中核になることは疑いがなく、仕様の安定性を重視して、XQueryデータモデルではなくInfosetデータモデルを採用することとした¹³⁾。将来的に、W3CのXQueryデータモデルに準拠することを明記するなど配慮を見せており、問合せ言語、更新機能など重要な機能開発を先送りにはしているものの、少なくとも利用者に受け入れられる要件が揃いつつある。

今後の展開

◆適合性について

SQL92は段階的に準拠できるように3つの適合レベル(初級、中級、完全)から成り立っていた。しかし、多くのSQL製品は初級レベルに準拠するのが大半であり、それより上位の適合レベルに準拠するSQL製品は現れなかった。その理由として、初級レベルより上位の適合レベルで規定される機能が、必ずしも利用者の要求を反映したものと言えなかったことがある。SQL92では、RDB完成形を追求したがゆえに、結果的に利用者の利用方法、SQL製品での実装技術を考慮した適合レベルが必ずしもうまく設定できなかった。

そこで、SQL2003では異なるアプローチを採用している。まず、SQL2003の全機能に番号を付与し、各機能を「中核SQL」と「パッケージ」に分類する³⁾。「中核SQL」はSQL92の初級レベルを完全に包含し、上位の適合レベルの一部機能を含む機能設定になっており、SQL2003規格の策定に関与していた米国、日本の主要なSQLベンダの製品ロードマップと照らし合わせて合意されたものである。また、「パッケージ」と呼ぶSQLの応用分野に向けた機能設定に適合することができる。SQLベンダのマニュアルなどで記述され始めており、応用分野ごとの適合性が木目細かく規定できるので、互換性の高いアプリケーション開発が期待されている。

◆SQL開発推進の競争・協調関係

SQL2003に準拠する製品開発時期が間近いこともあって、規格開発の間では主要SQLベンダのアーキテクトの意見が衝突する場面も多々あり、そのたびごとに意見調整が図られ、参加者が協調した仕様となるよう運営されている。市場のニーズ確認、開発方向性の明確化およびコンセンサス作りの場として、この規格開発はうまく機能している。

ただし、たとえば制度や運用、環境の問題も含めた総合的な対処が必要なセキュリティ機能や認証および監査機能などは、標準化の場で扱う必要性が高いと考えられるが、この規格開発の場に参画するベンダが少なくなりつつあることもあり、開発を中止した項目もある。応用分野の広がりとともに、積極的に応用指向の機能提案を行っていく必要がある。

◆データ管理基盤の将来像

情報統合手段としてのXMLを利用しXQueryおよびWebサービスをデータ管理基盤に適用することで、情報を共有化させ顧客に付加価値を提供させるさまざま

なアプローチが出現しつつある。具体的に、個別に最適化、固定化された業務システム、イントラネット上で統制されていないWebコンテンツ、文書、アプリケーションを低コストで統合したい用途などの要望に対応してデータ統合化の試みが行われており、新たな情報統合市場への対応と捉えることができる。

現時点で業界標準と呼ばれる製品あるいはソリューションがないため、新たな市場として覇権争いが行われていると考えられる。これらの動きを見ると、XMLを活用した統合化で新たな市場が形成されつつあり、SQLおよびXMLを共存させるSQL2003で市場の開拓が可能であると考えるべきであろう。

まとめ

DBMS開発者は、概して「保守的な文化」で育てられ、「更新済みのDB内容の保証は何者にも優先」とされている。この観点は、応用分野でのアプリケーション開発者が「再実行も可能」の世界に置かれている状況とはかなり異なる。それくらいにDBMS開発者からは、アプリケーションが遠い存在にみえる。

しかし、SQL2003は応用指向でDBMSを開発する必要があるため、各応用分野に対応するアプリケーション開発者の確保と先進的なアーキテクチャからDBMS基本構造の徹底した再設計技術力を持つDBMS開発者を確保しつつ、かつこれらを融合させるDBMS開発をSQLベンダは維持し続ける必要がある。DBMS開発者もアプリケーションが見えることを要望しており、新たな価値を生む応用に向けた先行開発も望まれる。

また、日本のSQLベンダもSQL2003に準拠する製品

を開発しており、積極的に仕様の提案を行い、主要SQLベンダとともに最新の動向、潮流を自らこの場に参画し創出することで、先行開発を目指している。規格開発の場で議論されるのは規格内容のみではなく、将来の方向性、実装技術などミドルウェア製品と密接に関連するので、将来の開発方向性も探ることができる場として、規格開発の場は重要である。SQLおよびXMLを併用してさらにデータ管理対象を広げる動きは今後とも継続し、そのような動きがミドルウェア製品に与えるインパクトはますます強まると考える

参考文献

- 1) 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解, ピアソン・エデュケーション (2003).
- 2) 山平耕作, 小寺 孝, 土田正士著: SQLスーパーテキスト, 技術評論社 (2004).
- 3) ISO/IEC 9075-1:2003 Database Language – SQL Part1: Framework (SQL/Framework).
- 4) ISO/IEC 9075-2:2003 Database Language – SQL Part2: Foundation (SQL/Foundation).
- 5) ISO/IEC 9075-3:2003 Database Language – SQL Part3: Call-Level Interface (SQL/CLI).
- 6) ISO/IEC 9075-4:2003 Database Language – SQL Part4: Persistent Stored Modules (SQL/PSM).
- 7) ISO/IEC 9075-9:2003 Database Language – SQL Part9: Management of External Data (SQL/MED).
- 8) ISO/IEC 9075-10:2003 Database Language – SQL Part10: Object Language Bindings (SQL/OLB).
- 9) ISO/IEC 9075-11:2003 Database Language – SQL Part11: Information and Definition Schemas (SQL/Schemata).
- 10) ISO/IEC 9075-13:2003 Database Language – SQL Part13: Routines and Types Using the Java Programming Language (SQL/JRT).
- 11) ISO/IEC 9075-14:2003 Database Language – SQL Part14: XML-Related Specifications (SQL/XML).
- 12) W3C: XQuery.1.0: An XML Query Language, W3C Working Draft 12 November 2003 (2003).
- 13) W3C: XML Information Set, W3C Recommendation 24 October 2001 (2001).

(平成16年4月28日受付)

