

# COBOL 利用技術のご紹介

COBOL コンソーシアム利用技術分科会

安藤 理恵(NEC システムテクノロジー Web ビジネス事業部)

## 第7回 データベースの利用

データベースは大量のデータを効率よく管理するシステムで、階層型、ネットワーク型、リレーショナル型があります。その中でもリレーショナルデータベースは広く使用されており、文字データや数値だけでなく、マルチメディアなどのデータも取り扱える様に機能が拡張されています。近年において、データベースの利用は、データマイニングやCRMシステムといった経営戦略にも活用されています。

### 1. データベースへのアクセスについて

データベースをアクセスする言語としてSQLがあります。SQLは、元はIBM社が開発した構造化問い合わせ言語ですが、現在では、ANSI、ISO、JISで仕様が定義されており、リレーショナルデータベースの標準処理言語となっています。

また、データベースをアクセスするための標準的な仕様としては、ODBCが一般によく知られています。ODBCは、Microsoft社によって提唱された、データベースにアクセスするためのソフトウェアの標準仕様です。各データベースの違いはODBCドライバによって吸収されるため、ユーザはODBCに定められた手順に従ってプログラムを書けば、接続先のデータベースがどのようなデータベース管理システムに管理されているか意識することなくアクセスできます。同様なデータベースアクセス仕様として、JavaプログラムからリレーショナルデータベースにアクセスするためのAPIであるJDBCが存在します。

現在、オープン系システムにおける標準的なデータベースとしては、Oracle、INFORMIX、Sybase、SQL Server等がありますが、COBOLからこれらのデータベースにアクセスする方法について、次に説明します。

## 2. 埋め込み SQL によるアクセス

プリコンパイル方式やコンパイラの実装によって、DBベンダや各COBOLベンダはCOBOLプログラム内に直接SQL文を記述する手段を提供しています。COBOLプログラムからANSI標準の埋め込みSQL文によって、データベースのデータ参照、更新等、データベースオブジェクトの操作を行なうことができます。

仕様が標準化されているため、ソースの互換性が確保できる利点があります。

### 記述例

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SAMPLE01.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
- -
```

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.  
01 WKKOSU PIC S9(5) SIGN IS LEADING SEPARATE.  
01 WKBANGO COMP-2.  
EXEC SQL END DECLARE SECTION END-EXEC.  
EXEC SQL INCLUDE SQLCA END-EXEC.  
- -
```

```
PROCEDURE DIVISION.  
- -
```

```
MOVE 1 TO WKBANGO.  
EXEC SQL  
    SELECT COUNT(ZAIKO) INTO :WKKOSU  
    FROM SEIHINHYO  
    WHERE SEIHINBANGO = :WKBANGO  
END-EXEC.  
DISPLAY WKKOSU.  
- -
```

```
STOP RUN.
```

ホスト変数の定義

共通領域の定義

データベース  
へのアクセス

#### (1) ホスト変数の定義

ホスト変数は、COBOLプログラムとデータベース間のデータの引き渡しに使用します。データベースにデータを入力する場合、COBOLプログラムでホスト変数にデータを代入しSQL文を用いてデータベースにデータを格納します。またCOBOLプログラムからデータベースのデータを参照する場合には、SQL文を用いてホスト変数にデータを格納することによりデータを参照することができます。

ホスト変数をSQL文中で使用する場合には、ホスト変数の前にコロン(:)を付けます。

#### (2) 共通領域の定義

データベースとの共通領域として使用するSQLCAには、エラーコードやメッセージ等のCOBOLプログラムに引き渡す、SQL文を実行した情報が格納されます。

### (3) データベースへのアクセス

SQL文の実行はCOBOLプログラムのPROCEDURE DIVISIONで記述します。SQL文はデータベース操作要求を記述するものです。この例では、製品番号をキーにデータを検索し、COBOLプログラムへ引き渡します。

#### [ マルチメディアデータへのアクセス ]

ベンダによっては、マルチメディアデータ(大容量データ)の扱いをサポートしています。

マルチメディア・データ(文字データ型およびバイナリ・データ型)はラージ・オブジェクト(LOB:Large Object)として扱います。

#### 記述例

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.SAMPLE01.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
- -  
  
EXEC SQL BEGIN DECLARE SECTION END-EXEC.  
01 BLOB1 SQL-BLOB.  
01 BUFF1 PIC X(32767) VARYING.  
01 AMOUNT PIC S9(9) COMP.  
01 OFFSET PIC S9(9) COMP.  
EXEC SQL END DECLARE SECTION END-EXEC.  
EXEC SQL INCLUDE SQLCA END-EXEC.  
EXEC SQL VAR BUFF1 IS LONG RAW(32767) END-EXEC.  
- -  
  
PROCEDURE DIVISION.  
- -  
  
MOVE 1 TO OFFSET.  
EXEC SQL ALLOCATE :BLOB1 END-EXEC.  
EXEC SQL  
    SELECT FRAME INTO :BLOB1  
    FROM MEDIATABLE WHERE BANGO = 1  
END-EXEC.  
EXEC SQL LOB OPEN :BLOB1 END-EXEC.  
MOVE 32767 TO AMOUNT .  
EXEC SQL  
    LOB READ :AMOUNT FROM :BLOB1 INTO :BUFF1  
END-EXEC.  
.  
.
```

### 3. READ/WRITE 文によるアクセス

前述の埋め込み SQL 文によるアクセス以外に、標準のファイルシステムを操作するのと同様な形式で、READ/WRITE 文によるデータベースのアクセスをサポートしている製品もあります。このような製品では COMMIT/ROLLBACK 文もサポートしていますので、トランザクション処理も可能となります。

従来は標準ファイルにより管理していたデータを、データベースに移行するといった場合に、COBOL プログラムを変更する部分が少なくして便利です。

ただし、プログラムによっては性能的な面において劣化する可能性もありますので、実際にテスト確認をしたほうがよいでしょう。

#### 4. DB アクセスライブラリによるアクセス

また、CALL 文による呼び出しや、オブジェクト指向型のインタフェースを使ってデータベースのアクセスをサポートしている製品もあります。

オブジェクト指向型のライブラリは、COBOL で作成したアプリケーションからデータベースをアクセスするためのクラスライブラリとなっています。データベースアクセスクラスライブラリは、プラットフォームおよびデータベースの種類に依存せず、複数種のデータベースに対して統一されたインタフェースでアクセスできます。

コーディングの簡素化や、オブジェクト指向プログラミングとの親和性においては優位でしょう。

#### 5. 留意点

アプリケーションが複数のデータベースにアクセスする場合、または異なる環境からデータベースを移行した場合に、それぞれのデータベースのコード体系に注意しなければなりません。コード体系が異なるとコードの並び順が異なるため、ソート処理/範囲処理結果が異なることがあります。

また、埋め込みSQL文の使用において、SQL99の登場により最新技術にマッチした仕様が標準化されていますが、ベンダによっては標準の範囲を超えたSQL記述をサポートしており、これらの拡張されたSQL記述を使った場合には、後々ソースの互換性が問題となることがあるので、注意が必要です。

#### 6. まとめ

上記の方法により、COBOLアプリケーションからデータベースに容易にアクセスすることができます。なお、実装されている機能は各ベンダーにより異なります。実際に運用される場合には、ご利用になるベンダーの製品情報を確認してください。