DATABASE

第6章

Empress Embeddedの使用例

C言語によるプログラミング(1)
――初期設定と検索

David Zhang

ここでは，Empress Software社のデータベースを例に，C言語インターフェースを使ったデータベース・プログラミングの実例について解説する．データベースのオープン，操作，クローズという一連の流れをC言語で操作できる．データベース専用の言語を知らなくても比較的容易に扱える．（編集部）

1 データベースの概要

● 概要

ここではEmpress Software社のデータベースである「Empress Embedded」を例に，組み込みデータベースのプログラミングについて解説していきます．このデータベースはテキスト・データとマルチメディア・データの両方を扱えるリレーショナル・データベース（RDBMS： Relational Database Management System）です．画像・音声などを扱うマルチメディア・データベース，あるいは気象，地図情報システム，インターネット/イントラネットのコンテンツ管理などで利用されています．

● 稼働環境と必要資源

このデータベースは，一般的なUNIX（32ビット/64ビット），Linux，各種リアルタイムOS，Windows 2000/XP/2003で動作します．

今回は，本稿で紹介するデータベース（試用版）とサンプル・プログラムを本誌のWebページに用意しました．

```
http://www.cqpub.co.jp/interface/download/
```

からダウンロードできます．インストールの詳細についてはアーカイブに含まれるドキュメントをご覧ください．なお，使用形態によっては，Cコンパイラ，TCP/IP，JDKなどが必要となります．

ここでは，C言語インターフェースを使ったデータベース・プログラミングの実例を，サンプル・コードを示しながら紹介します．

2 データベースの作成と操作

● データベースの作成

インストール後，環境変数の設定が終了した時点（再ログイン）でデータベース操作が可能となります．以下ではUNIX版を例に，操作方法について説明します．

次のコマンドでデータベースを作成できます．

```
# empmkdb testdb
```

testdbという名前のディレクトリが作成され，その下に**図1**のようなファイルが作成されます．これで，testdbという名前のデータベースが作成されたことになります．このtestdbに対するSQLによる簡単な操作の例を，**図2**に示します．操作方法

```
0001.rel      0003.rel      00060001.ix   00080001.ixl  0010.rel
00010001.ix   00030001.ix   00060008.dtf  00080003.dtf
00100001.dtf
00010004.dtf  00030001.ixl  00060009.dtf  0009.rel
00100001.ix
0002.rel      00030010.dtf  00060011.dtf  00090001.ix
00100001.ixl
00020001.ix   00030011.dtf  00060012.dtf  00090001.ixl
00100011.dtf
00020007.dtf  0004.rel      00060016.dtf  00090011.dtf  _lock
00020008.dtf  00040001.ix   0007.rel      00090012.dtf  _module
00020009.dtf  00040001.ixl  00070001.ix   00090013.dtf  _trn
00020010.dtf  0005.rel      00070004.dtf  00090014.dtf  cdinator
00020011.dtf  00050001.ix   00070007.dtf  00090015.dtf  dd_cache
00020014.dtf  00050001.ixl  0008.rel      00090016.dtf  tabzero
00020015.dtf  0006.rel      00080001.ix   00090017.dtf
```

図1　作成されるファイル一覧

```
# empsql testdb                          ← 対話型SQLの起動
EMPRESS V8.60
(c) Copyright Empress Software Inc. 1983, 1999
1* create testtbl (a1, a2 char, a3 date);   ← テーブルの作成
2* display testtbl;                        ← テーブルの表示
***  Table: testtbl  ***
  Attributes:
    a1  integer
    a2  character(25,1)
    a3  date(0)
3* stop;                                   ← 対話型SQLの終了
#
```

図2　testdbに対する，SQLによる簡単な操作

KEYWORD――データベースの作成，mrルーチン，データベースの検索

リスト1　mxルーチンを使った例

```
#include <mscc.h>
msmain()
{
        /* テーブルの Open  */
        mxopen ("database", "table", "u");
        mxgetbegin ("table", CHARNIL);
        while (mxget ())        /* 1 レコード入力    */
        mxdel("table");         /* 1 レコード削除    */
        mxgetend ();
        mxclose ("table");      /* テーブルの Close */
}
```

リスト2　mrルーチンを使った例

```
#include <mscc.h>
msmain()
{
    addr    tabdesc, recdesc, retdesc;
    /* テーブルの Open */
    tabdesc = mropen ("database", "table", 'u');
    recdesc = mrmkrec (tabdesc);
    retdesc = mrgetbegin (ADDRNIL, recdesc, ADDRNIL);
    while (mrget (retdesc))     /* 1 レコード入力   */
        mrdel (recdesc);        /* 1 レコード削除   */
    mrgetend (retdesc);
    mrdelend (recdesc);
    mrfrrec (recdesc);
    mrclose (tabdesc);          /* テーブルの Close */
}
```

がわからないときは，コマンドラインからhelpと入力すると，ヘルプ画面が表示されます．

● データベースの削除

データベースを削除する場合には，かならずemprmdbコマンドを使用してください．システムのコマンド“rm -rf”で消去した場合には，予期せぬエラーが発生する場合があります．

```
# emprmdb testdb
```

そのほか，データベースの待避を行うempexptなどのユーティリティも用意されています．

3 C言語によるデータベース・アクセス・アプリケーションの作成

● 二つのAPI，mrルーチンとmxルーチン

本データベースは，データベース構造へ直接アクセスするために，「mxルーチン」と「mrルーチン」という2種類のデータベースAPIを用意しています．

mxルーチンは，mrルーチンよりかなり簡素であり，簡単にアプリケーションを作成できます．逆に複雑なアプリケーションや，速度が重視されるアプリケーションを作成したい場合はmrルーチンを使用することを推奨しています．

ここでは比較のために，テーブルの全レコードを削除するコーディング例を示します．**リスト1**がmxルーチン，**リスト2**がmrルーチンです．**リスト3**には，参考までにSQLを使った例を示します．

4 mrルーチンによるサンプル・プログラム

ここからは，高速にデータベースの操作が行えるmrルーチンの使用方法を詳細に解説します．C言語によるサンプル・プログラムを使うので，ソースを読み進めていけば，データベースを使ったプログラミングの概要がわかると思います．

以下のサンプル・プログラムでは，論理的なデータベース名としてrepairsを使用します．データベースの物理的な位置は，システムのいかなる場所であってもかまいません．

リスト3　SQLを使った例

```
#include        <mscc.h>
EXEC SQL INCLUDE SQLCA;
EXEC SQL BEGIN DECLARE SECTION ;
char    sqls[1024];
EXEC SQL END DECLARE SECTION ;
main()
{
        EXEC SQL INIT;
        EXEC SQL DATABASE IS "database";
        strcpy(sqls, "delete from table;");
        EXEC SQL EXECUTE IMMEDIATE :sqls;
        if( SQLCODE )
                printf ("++ IMMEDIATE Error ++¥n");
        EXEC SQL EXIT ;
    :
```

5 テーブル単位の操作

● テーブルのオープンとクローズ

テーブルへのアクセスの第1ステップは，テーブル記述子を取得することです．mropen関数はテーブルをオープンし，テーブル記述子を返します．テーブルをオープンできない場合は，呼び出しプログラムを終了します．

mropen関数では，データベースに関する各種情報を保存してある「データ・ディクショナリ」に設定されているレベルのロックをテーブルに施します．たとえばテーブル・レベル・ロックが設定されていれば，テーブル全体をロックしようとします．そのときすでにロック中であれば，変数MSLOCKRETRYとMSLOCKSLEEPに設定されている値によって，再度ロックを実行します．テーブルをオープンしたあとロックを設定できない場合は，エラー・メッセージを出力し，呼び出しプログラムを終了します．

データベースのオープン関数は次のようなものです．

```
table_desc = mropen(database_name,
                                table_name,mode);
table_desc = mrtopen(database_name,
                                table_name,mode);
```

mropen関数とmrtopen関数はよく似ていますが，mropen関数はエラー発生時にプロセスを終了するのに対し，‘t’が付

New Products――Oracle，データベース・セキュリティ・ソフトウェアを発表
Oracle社は，データやアプリケーション・ソフトウェアを保護するセキュリティ・ソフトウェア「Oracle Database Vault」を発表した．権限が1点に集中することで生じるリスクを解消したという．「Oracle Database 10g」に組み込んで利用する．