

直接起動/埋め込みSQL/モジュール言語

実行環境

| | |
|-------|----------------------|
| OS | Debian 7.2 Linux |
| RDBMS | RDBMS PostgreSQL 9.1 |
| ホスト言語 | C言語(gcc 4.7.2) |

事前に作成した表

表 sample

| | |
|------|---------|
| tel | name |
| 6457 | 情報通信実験室 |
| 6447 | 高橋研究室 |

比較

| | 直接起動 | ホスト言語方式 埋め込みSQL | ホスト言語方式 モジュール言語 |
|-------|--|--|--|
| 必要な知識 | SQL | C言語 埋め込みSQL | C言語 関連ライブラリ |
| 開発環境 | RDBMS | コンパイラ ・埋込言語→C言語 ・C言語→マシン語 関連ライブラリ | コンパイラ ・C言語→マシン語 関連ライブラリ・ヘッダ |
| 汎用性 | 日本工業規格で制定 | 日本工業規格で制定 | 汎用性なし |
| 長所 | 開発時間は最短 RDBMSを使うための最低 限な環境でよい | アイコンをクリック? プログラムやSQLの内容を知ら なくてもプログラムを実行する ことができる。 | アイコンをクリック? プログラムやSQLの内容を知ら なくてもプログラムを実行 することができる。 |
| 短所 | 黒画面は不人気? SQLは難しい? カジュアルユーザーに SQLを教えるのか? | プリコンパイラが必要 (提供しないベンダもある) 開発者の負担大(C言語+SQL) | ライブラリに互換性はない 開発者の負担大(C言語+SQL) RDBMSを他社に変更する =全面的に作り直す |

直接起動/埋め込みSQL/モジュール言語

直接起動

端末からSQLを入力する。

```
webmaster@Na:~/mydir/lecture/DB
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[webmaster@Na DB]$ psql
psql (8.4.9)
"help" でヘルプを表示します。

webmaster=> SELECT * FROM sample ;
tel |      name
-----+-----
6547 | 情報通信実験室
6447 | 高橋研究室
(2 行)

webmaster=> █
```

ホスト言語方式(モジュール言語)

C言語だけで記述する。データベースとの通信は関数として実装する。

```
sample2.c

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include "postgres.h"
#include "libpq-fe.h"

int main(int argc, char **argv)
{
    char dbName[255] = "webmaster"; /* データベース名はハードコーディング */
    char sql[255];
    int i;
    PGconn *con;
    PGresult *res;
    char *tel, *name;

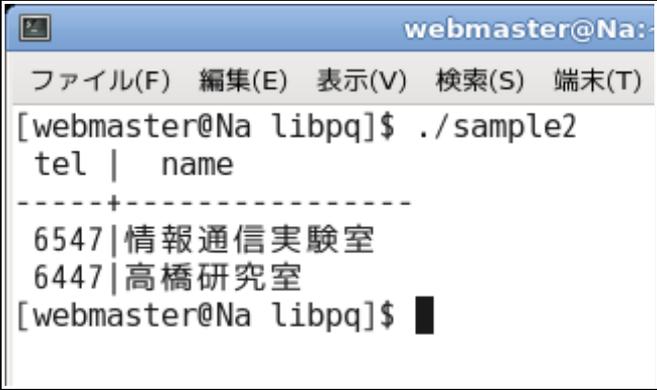
    con = PQsetdb("", "", NULL, NULL, dbName);
    sprintf(sql, "select * from sample");
    res = PQexec(con, sql);
    printf(" tel | name\n");
    printf("-----+-----\n");
    for(i = 0; i < 2 ;i++) {
        tel = PQgetvalue(res, i, 0);
        name = PQgetvalue(res, i, 1);
        printf("%5s | %s\n", tel, name);
    }
    PQclear(res);
    return 0;
}
```

直接起動/埋め込みSQL/モジュール言語

Makefile

```
## SQL library sample
All : sample2
    ./sample2
sample2 : sample2.c
    gcc -o sample2 sample2.c -I /usr/include/pgsql/server -lpq
clean :
    rm sample2
```

実行結果



```
webmaster@Na:~$ ./sample2
tel | name
-----+-----
6547 | 情報通信実験室
6447 | 高橋研究室
[webmaster@Na libpq]$
```

ホスト言語方式(埋め込み SQL)

C 言語で記述したソースプログラムの中に SQL に関する部分を記述する。

sample1.pgc

```
#include <stdio.h>
int main(){
    int i;
    EXEC SQL CONNECT TO webmaster;
    EXEC SQL BEGIN DECLARE SECTION;
        int tel=0;
        char name[20]="";
    EXEC SQL END DECLARE SECTION;

    EXEC SQL DECLARE sample_Cursor CURSOR FOR
        SELECT * FROM sample;

    EXEC SQL OPEN sample_Cursor;

    printf("SQLSTATE| tel | name\n");
    printf("-----+-----+-----\n");

    for(i = 0; i < 3 ;i++) {
        EXEC SQL FETCH sample_Cursor INTO :tel,:name;
        printf("%8s|%5d|%s\n",sqlca.sqlstate,tel,name);
    }

    EXEC SQL CLOSE sample_Cursor;

    EXEC SQL DISCONNECT webmaster;
    return 0;
}
```

```
Makefile

## Embedded SQL sample
All : sample1
    ./sample1
sample1 : sample1.c
    gcc -o sample1 sample1.c -lecp
sample1.c : sample1.pgc
    ecpg sample1.pgc
clean :
    rm sample1 sample1.c
```

実行結果

SQLSTATE が 00000 であるとき、問い合わせ結果が得られる。

何らかのエラーがあるときは SQLSTATE が 00000 以外の値になる。この例では「これ以上問い合わせ結果がない」ため SQLSTATE が 02000 になっている。



```
webmaster@Na:~/mydir/lecture
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[webmaster@Na embedded]$ ./sample1
SQLSTATE| tel | name
-----+-----
    00000| 6547|情報通信実験
    00000| 6447|高橋研究室
    02000| 6447|高橋研究室
[webmaster@Na embedded]$
```

直接起動/埋め込みSQL/モジュール言語

sample1.pgc を ecpg(プリプロセッサ)でコンパイルし、sample.c を生成する。

この後、gcc(C言語コンパイラ)でマシン語にする。

sample.c (sample1.pgc から生成されたソースプログラム)

```
/* Processed by ecpg (4.5.0) */
/* These include files are added by the preprocessor */
#include <ecpglib.h>
#include <ecpgerrno.h>
#include <sqlca.h>
/* End of automatic include section */

#line 1 "sample1.pgc"
#include <stdio.h>
int main(){
    int i;
    { ECPGconnect(__LINE__, 0, "webmaster", NULL, NULL, NULL, 0); }
#line 4 "sample1.pgc"

    /* exec sql begin declare section */

#line 6 "sample1.pgc"
    int tel = 0 ;

#line 7 "sample1.pgc"
    char name [ 20 ] = "" ;
/* exec sql end declare section */
#line 8 "sample1.pgc"

    /* declare sample_Cursor cursor for select * from sample */
#line 11 "sample1.pgc"

    { ECPGdo(__LINE__, 0, 1, NULL, 0, ECPGst_normal, "declare sample_Cursor cursor for select * from
sample", ECPGt_EOIT, ECPGt_EORT);}
#line 13 "sample1.pgc"
```

```
printf("SQLSTATE| tel | name\n");
printf("-----+-----+-----\n");

for(i = 0; i < 3 ;i++) {
    { ECPGdo(__LINE__, 0, 1, NULL, 0, ECPGst_normal, "fetch sample_Cursor", ECPGt_EOIT,
            ECPGt_int,&(tel),(long)1,(long)1,sizeof(int),
            ECPGt_NO_INDICATOR, NULL , 0L, 0L, 0L,
            ECPGt_char,(name),(long)20,(long)1,(20)*sizeof(char),
            ECPGt_NO_INDICATOR, NULL , 0L, 0L, 0L, ECPGt_EORT);}
#line 19 "sample1.pgc"

    printf("%8s|%5d| %s\n",sqlca.sqlstate,tel,name);
}

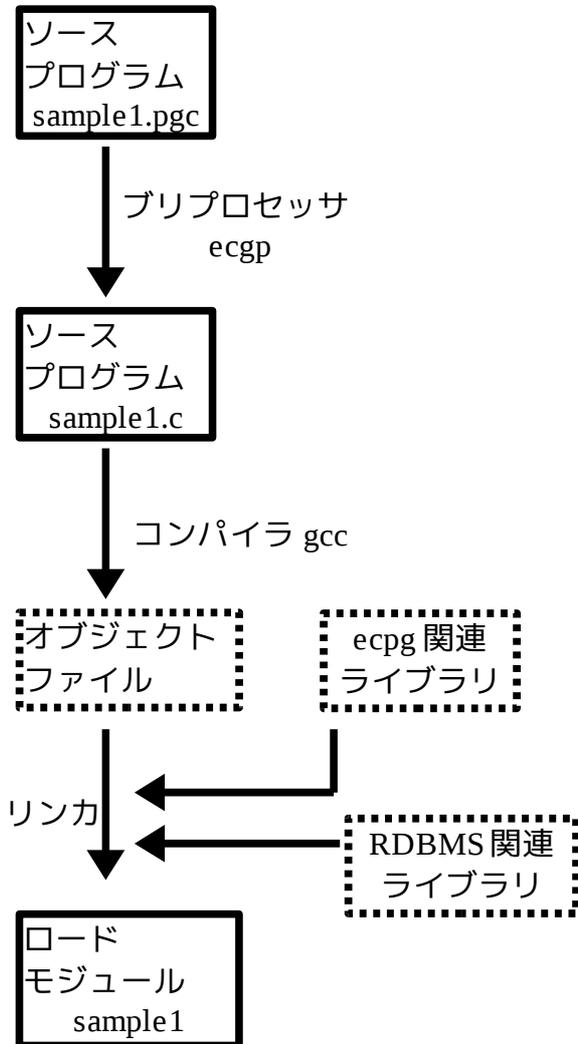
{ ECPGdo(__LINE__, 0, 1, NULL, 0, ECPGst_normal, "close sample_Cursor", ECPGt_EOIT,
ECPGt_EORT);}
#line 29 "sample1.pgc"

{ ECPGdisconnect(__LINE__, "webmaster");}
#line 31 "sample1.pgc"

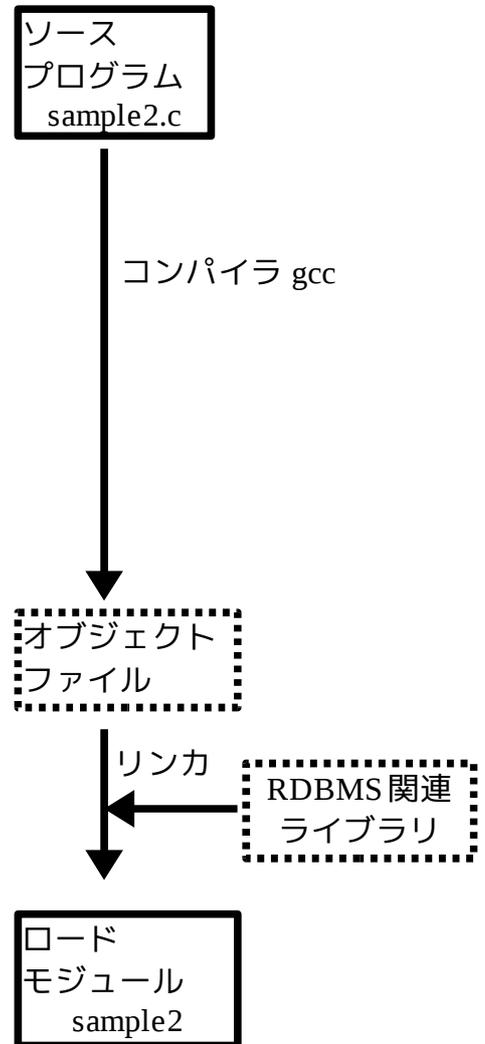
return 0;
}
```

ロードモジュールを生成する手順

埋め込み SQL



ホスト言語方式



| SQL Server 2000の埋め込みSQL | | | |
|--|-------------------|-------|------------|
| 対象製品 | Advantage Gen 6.5 | 文書番号 | 071150007 |
| プラットフォーム | Windows | 最終更新日 | 2004/06/23 |
| ◆ 詳細内容 | | | |
| <p>Q : マイクロソフト社は、SQL Server 2000 SP3において埋め込みSQLのサポートを終了しました。</p> <p>A : SQL Server 2000 Service Pack 3以降、マイクロソフト社による埋め込みSQLのサポートは行われなくなり ます。 この結果、MS/SQLテクニカル設計のDBMSアクセス方法をODBCへ変更し、すべてのコードを再生成する 必要性が出てきます。 さらに、ご使用アプリケーションに合ったODBCデータ ソースを定義しなければなりません。</p> <p>ODBCの代わりにMS/SQL DBMSアクセス方法を引き続き使用すると、MS/SQLプリコンパイラ は、.sqc (ソース) ファイルから生成するCファイル内にある埋め込みSQLをコメントアウトします。 その後このファイルをコンパイルすると、エラーを発生させることなくコンパイルは完了しますが、埋め込み SQLステートメントはコメントアウトされているため、予想どおりに動作しません。</p> | | | |