

## Part 2: 実践編

Linux上のPerl/PHPからデータベースアクセスを実現

# オープンソースソフトによる Symfowareの利用法

富士通株式会社 ソフトウェア事業本部 小幡孝司 OBATA, Takashi

Linux版Symfowareでは、Solaris版やWindows版で培った各種の機能に加え、PerlやPHPなどのオープンソースソフトウェアとの連携も可能である。ここでは、Windows版 / Linux版Symfowareの導入を行ない、その後、Perlを使ったデータベースの生成、およびPHPからのデータの表示を実践的に解説する。

## LinuxからのSymfowareの利用

Linuxは、開発者のコミュニティによって生み出され、GPL(GNU General Public License)によって管理されているオープンソースのOSである。ライセンスには使用制限もロイヤリティもないが、配布に際してはソースコードを提供することを義務づけている。

ここで解説するLinux版Symfoware Serverは、SymfowareがSolaris版、Windows版において培ってきた高性能 / 高信頼性に加え、Linux文化に則ったオープン性を活用し、豊富な機能を搭載したRDBエンジンである。

これまで、各データベースベンダは、SQL文を処理するインタープリタ型言語や、CやCOBOLにSQL文を埋め込むためのインターフェイスを提供してきた。これらのデータベースアクセスの手段はベンダによって異なり、完全な互換性の保証はなかった。

そこで、オープンソースであるLinuxには、PHP、Perlといったインタープリタ型プログラミング言語からデータベースにアクセスする際、複数のデータベースベンダのアクセスを1つのクラスインターフェイスにまとめる方法が用意された。

例えば、PHPスクリプトからデータベースにアクセスするためには「PEAR」と呼ばれるクラスインターフェイスが用意されている。Symfowareにおいても、データベース操作のための「Symfoware関数」をPHPLib、PEARから呼び出してハンドリングする「PHP Symfoware」を公開している。PHP Symfowareは、Symfowareホームページ(<http://software.fujitsu.com/jp/symfoware/>)から入手可能である。

また、Perlスクリプトからデータベースにアクセスするために「DBI」と呼ばれるクラスインターフェイスのPerlモジュールが用意されている。Symfowareにおいても、「Database Driver for Symfoware」(以降、DBD::Symfoware)を公開している。

DBD::Symfowareは、Symfowareホームページ（<http://software.fujitsu.com/jp/symfoware/>）から入手可能である。

これらを利用することで、簡単にSymfowareデータベースを利用することができる。ここでは、実際にSymfoware Serverをインストールし、PHP、Perlの簡単なサンプルスクリプトを実行してみたい。

### Linux版のインストール方法

Linux版 Symfoware Serverは、GUIでデータベースのセットアップを支援する「Web-Based Admin View」やデータベースの構築を支援する「WebDBtools」などの機能を揃えている。ここで

**注1:** 今回のインストーラでインストールした場合は、一度Symfoware Serverをアンインストールした後、再度インストールを行なうこと。

はPHP、Perlを使ったサンプルスクリプトの実行を前提に、簡易インストールおよびセットアップを行なう。

今回添付のサンプルスクリプトを実行するために、必要最低限の機能のインストールおよび設定を行なうインストーラ（install.sh）を用意した。以下に示す手順に沿って操作すれば、コマンドラインによるオペレーションで簡単にインストールから設定までを行なえる。

Linux版Symfoware Serverのすべての機能やGUIによる設定ツール「WebDBtools」などを利用したい場合は、インストールガイドやオンラインマニュアルを参照し、symfo\_instal( /mnt/cdrom/linux/SERVER配下)を使ってインストールしていただきたい<sup>注1</sup>。

インストールの流れは、大きく以下のようになっている。

## Symfoware Serverをインストールする

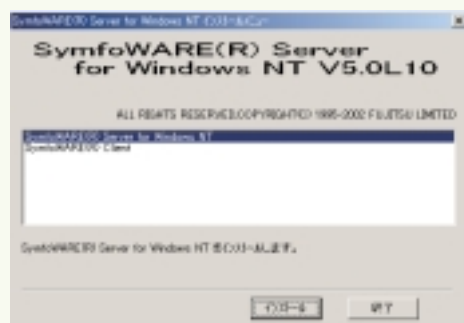
### Windows版のインストール方法

収録しているWindows版はPHP、Perlからの利用はできない。ここでは、インストール方法のみを簡単に解説する（CD-ROMについては表紙裏を参照）。インストール時のシステム要件は収録しているオンラインマニュアル（WINDOWS¥MANUAL ¥JAPANESE¥COVER.HTM）を参照していただきたい。

#### 1. インストーラを起動

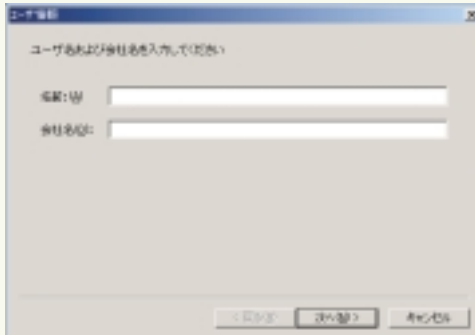
インストーラを起動するには、CD-ROMドライブ名 "¥windows¥Menu.exe" を実行する。インストールメニューが表示されるので、[ SymfoWARE(R)

Server for Windows NT ]を選択し、[ インストール ]ボタンをクリックする。



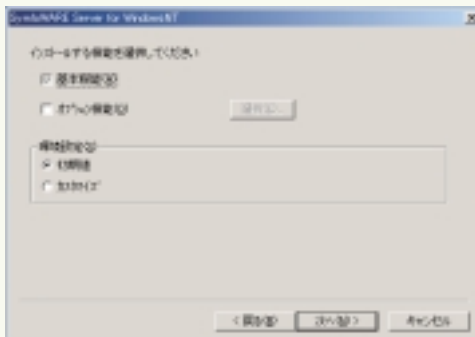
画面1: Windows版インストーラの起動

## 2. ユーザー名と会社名を入力



画面2: ユーザー情報の入力

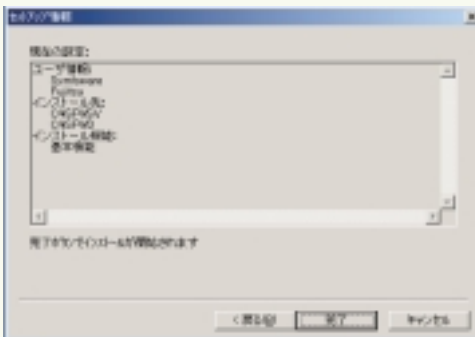
## 3. インストールする機能を選択



画面3: 機能の選択

## 4. インストール先を指定

[インストール先の指定]ダイアログボックスで、インストール先を指定する。ここではデフォルト値が入力されているが、利用者の環境に合わせて任意にカスタマイズできる。



画面4: インストール実行前の確認

## 5. インストール情報を確認

[セットアップ情報]ダイアログボックスで、インストール時に指定した情報を確認する。

以降は、画面の指示に従って、インストールを行なう。



画面5: インストール先の指定

## 6. システムの再起動を行なう

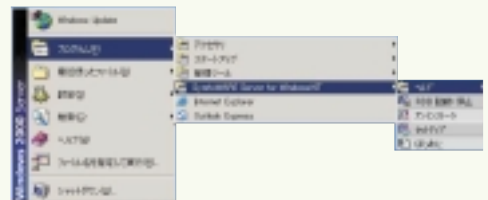
設定内容を有効にするため、システムを再起動する。

## 7. 再起動後のログオン

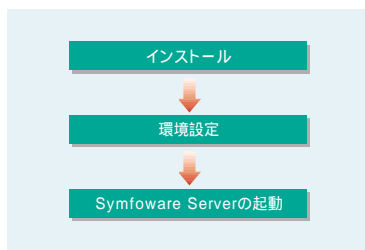
再起動後に自動的に行なわれる環境設定があるため、再起動後のログオンはインストール時と同じユーザーIDで実行する。

## 8. インストール終了を確認

インストールが終了し、[スタート]メニューの[プログラム]に[SymfoWARE Server for Windows NT]が追加されていれば、インストールは完了である。



画面6: メニューにアイコンが追加される



図：Linux版インストールの概要

以下の手順に従ってLinux版のインストールからSymfoware Serverの起動までを行なう。

### 1. システム上でスーパーユーザーになる

```
#su
Password xxxxxx パスワード
```

### 2. CD-ROM を以下のようにマウントする

```
#mount -t iso9660 -o ro /dev/cdrom/ /mnt/cdrom
```

### 3. インストールを開始する

カレントディレクトリをCD-ROMがマウントされているディレクトリまで移動し、install.shを実行する。

```
#cd /mnt/cdrom/linux/SAMPLE
#./install.sh
```

コマンドを実行すると、インストール済パッケージをチェック後、以下のメッセージが出力されるので、“y”を選択してインストールを開始する。

**注2:** システム名は、rdbsys1となる。

**注3:** /SFWというディレクトリが作成され、その配下に各資源（ログ管理ファイル、テンポラリログ、RDBディクショナリ）が作成される。

```
Symfoware Server Standard Edition V5.0L10 trial
インストール済パッケージをチェックします
パッケージのインストールを開始しますか? [y,q]:y
```

以下のメッセージ出力後、インストールが完了する。

```
パッケージ<FJSVsymse>をインストールします
パッケージ<FSUNiconv>をインストールします
パッケージ<FJSVrdb2b>をインストールします
パッケージ<FJSVrdbap>をインストールします
パッケージ<FJSVrdbdb>をインストールします
インストールが完了しました
```

### 4. インストール後の環境設定

インストール後、/opt/symfoware/demo配下に以下のシェルスクリプトが作成される。

```
remove.sh : アンインストール用
setup.sh : 環境設定用
start.sh : Symfoware Server 起動用
stop.sh : Symfoware Server 停止用
```

ここでは、/opt/symfoware/demoに作成された環境設定用のスクリプト（setup.sh）を実行して、Symfoware Serverを動作させるために必要な環境を設定する<sup>注2</sup>。

```
# cd /opt/symfoware/demo
# ./setup.sh
```

以下のメッセージ出力後、設定が完了する<sup>注3</sup>。

環境設定を開始します

```
資源格納用ディレクトリの作成 : OK
ログ管理ファイル作成 : OK
```

```
テンポラリログ作成      : OK
RDB ディクショナリ作成  : OK
```

環境設定が完了しました

## 5. Symfoware Serverの起動/停止およびアンインストール

Symfoware Serverを起動するには、Symfoware/RDB 起動用スクリプト( start.sh )を実行する。

```
#!/start.sh
```

コマンドを実行すると、以下のメッセージを出力後、起動が完了する。

```
rdb: INFO: qdg02600i:RDBIIの起動を開始します
(システム名=rdbsys1)
rdb: INFO: qdg02602i:起動ステージ 0 完了
(システム名=rdbsys1)
rdb: INFO: qdg02602i:起動ステージ 1 完了
(システム名=rdbsys1)
rdb: INFO: qdg02602i:起動ステージ 2 完了
(システム名=rdbsys1)
rdb: INFO: qdg02602i:起動ステージ 3 完了
(システム名=rdbsys1)
rdb: INFO: qdg02601i:RDBIIの起動が完了しました
(システム名=rdbsys1)
```

Symfoware Serverを停止する場合には、停止用スクリプト( stop.sh )を実行する。

```
#!/stop.sh
```

以下のメッセージ出力後、Symfoware が停止する。

```
rdb: INFO: qdg02608i:RDBIIの停止を開始します
(システム名=rdbsys1)
rdb: INFO: qdg02610i:停止ステージ 3 完了
(システム名=rdbsys1)
```

```
rdb: INFO: qdg02610i:停止ステージ 2 完了
(システム名=rdbsys1)
rdb: INFO: qdg02610i:停止ステージ 1 完了
(システム名=rdbsys1)
rdb: INFO: qdg02610i:停止ステージ 0 完了
(システム名=rdbsys1)
rdb: INFO: qdg02609i:RDBIIの停止が完了しました
(システム名=rdbsys1)
```

Symfowareのアンインストールが必要な場合は、アンインストール用スクリプト( remove.sh )を実行する。

```
#!/remove.sh
```

コマンドを実行すると、以下のメッセージを出力するので、“ y ”を選択してアンインストールを開始する。

```
Symfoware Server Standard Edition
V5.0L10 trial
アンインストールを開始しますか? [y,q]:y
```

以下のメッセージ出力後、アンインストールが完了する。

```
パッケージ<FJSVrddb>をアンインストールします
パッケージ<FJSVrdbap>をアンインストールします
パッケージ<FJSVrdb2b>をアンインストールします
パッケージ<FSUNiconv>をアンインストールします
パッケージ<FJSVsymse>をアンインストールします
アンインストール処理が完了しました
```

## Perl/PHPからSymfowareを利用する

以下では、Symfoware用モジュールの組み込みからサンプルテーブルの作成、コード作成、実行までの一連の作業を行なってみる。

いまやスクリプト言語の代表格であるPerlと、

基本的なプログラミング知識があれば誰でも簡単に活用できるPHPを使ってSymfowareデータベースを利用してみたい。

解説では、Perlでデータベースを生成し、データのロードまでを行なう。その後、PHPを使ってデータの取り出し、ブラウザへの出力までを行なう。

### Perl DBD::Symfowareによるデータベースの生成

Perl( Practical Extraction and Report Language: 実用データ取得レポート作成言語 )とは、インタプリタ型のプログラミング言語である。Perlが処理するプログラムのことをPerlスクリプトと呼ぶ。

Perlについての詳細は「CPAN( Comprehensive Perl Archive Network )」のホームページ <http://www.cpan.org/> を参照されたい。

#### 1. Perl DBIのインストール

CPAN( <http://www.cpan.org/modules/by-module/DBI/> )から、DBIソースのアーカイブを入手し、インストールする。

#### 2. Perl DBD::Symfowareのインストール

DBD::SymfowareソースのアーカイブをSymfowareホームページのダウンロードページ( <http://software.fujitsu.com/jp/symfoware/download/> )より入手する。

スーパーユーザーでログインする。

DBD::Symfowareソースのアーカイブを任意の作業ディレクトリ(ここでは/tmpと仮定)に置きDBD::Symfowareソースを展開する。

```
cd /tmp
gunzip -c DBD-Symfoware-1.00.tar.gz | tar xvf -
```

DBD::Symfowareソースディレクトリに移動する。

```
cd /tmp/DBD-Symfoware-1.00
```

DBD::Symfowareをビルドし、インストールする。

```
perl Makefile.PL
make
make install
```

#### 3. Perl サンプルスクリプト

ここでは、データベースを定義し、データを投入するまでの一連の作業を行なうサンプルスクリプトを実行してみる。

以下に、サンプルスクリプトの要点を簡単に説明する。サンプルスクリプトはCD-ROM内に格納されている。

スクリプトでは、まずDBIモジュールをロードする。続けて、DDL 定義文を設定するファイル「ddl.dat」を新規モードでオープンし、ファイルハンドルを取得する。

```
use DBI ;
$ret = open FileHandle, ">/tmp/ddl.dat" ;
$fh = FileHandle ;
```

データベースを定義する定義文を先ほどオープンしたファイル「ddl.dat」に書き込む。

```
print $fh <<"EOF" ;
CREATE DATABASE RDBDB ;
CREATE DBSPACE DBSP1 ALLOCATE FILE /SFW/
DBSP
ATTRIBUTE SPACE(10M) ;
EOF
close $fh ;
```

データベース名: RDBDB  
 DBスペース名: DBSP1  
 ファイル名: DBSP(格納場所は/SFW)  
 DBスペース容量: 10MB

もちろん、この定義文はあらかじめ作成しておいても問題はなく、ここでは処理の中で定義文の作成を一緒に行なっているだけである。

```
system "rdbddlex -d RDBDB /tmp/
ddl.dat" ;
unlink "/tmp/ddl.dat" ;
```

先ほど定義したDDL定義文「ddl.dat」を実行し、データベースを設定する。その後、ddl.datは消去される。

データベースに接続し、データベースハンドルオブジェクトを生成する。

```
$dbh = DBI->connect( "dbi:Symfoware:RDBDB",
"username", "password" ) ;
```

以降、このデータベースハンドルオブジェクト(ここでは\$dbh)を使ってデータベースにアクセスする。

なお、ここではデータベースにローカルアクセスするので、データソース名が「dbi:Symfoware:データベース名」(データベース名は先ほど設定したとおり「RDBDB」となっている点に注意したい。usernameおよびpasswordの部分は、データベースに接続する認可識別子(ユーザー名)とパスワードである。

```
$dbh->do( "CREATE SCHEMA STOCKS" ) ;
```

```
$dbh->do( "CREATE TABLE STOCKS.STOCK (
PRODUCT_NUMBER
INTEGER NOT NULL, PRODUCT_NAME CHAR(20)
NOT NULL,
STOCK_QUANTITY INTEGER,
WAREHOUSE_NUMBER SMALLINT NOT NULL ) ON
DBSP1" ) ;
```

作成したデータベースにスキーマ名およびスキーマの構成要素である表の定義を行なう(表)。定義は以下のとおりである。

スキーマ名: STOCKS  
 表名: STOCK

INSERT文のSQLを準備し、ステートメントハンドルオブジェクトを生成する。

```
$sth = $dbh->prepare( "INSERT INTO
STOCKS.STOCK VALUES( ?, ?, ?, ? )" ) ;
```

ステートメントハンドル(ここでは\$sth)とは、準備されたSQL文を表わすオブジェクトである。

```
$sth->execute( 140, 'cassette deck', 120, 2 ) ;
$sth->execute( 110, 'television', 85, 2 ) ;
$dbh->disconnect ;
```

準備されたステートメントハンドルに対して、executeメソッドを発行してやることで、そのSQLが実行される。ここでは、データベースにデータを投入している。さらに、データベースへのコネクションを切断する。

表: STOCKテーブルの定義

列名	データ型	制約
PRODUCT_NUMBER	INTEGER	NOT NULL
PRODUCT_NAME	CHAR(20)	NOT NULL
STOCK_QUANTITY	INTEGER	
WAREHOUSE_NUMBER	SMALLINT	NOT NULL

#### 4. サンプルスクリプトの実行

サンプルアプリケーション「create.pl」を実行するためには、まず、環境設定が必要である。

```
export LANG=ja_JP.eucJP
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/
opt/FJSVrdb2b/lib:/etc/opt/FSuniconv/lib
export PATH=$PATH:/opt/FJSVrdb2b/bin:./
export RDBNAME=rdbsys1
```

誌面の都合により で折り返しています

実際に、サンプルアプリケーションを実行すると、データベースが定義されデータの投入が行なわれる。

この際、Symfoware Serverが起動されていない場合は、前述の「start.sh」にて必ず起動しておくこと。

また、「create.pl」内の「username」「password」の部分の正しく書き換えることを忘れないで欲しい。

```
perl create.pl
```

ここまでで、データベースを作成し、データを投入してみた。次に、PHPを使ってデータベースにアクセスし、Webブラウザ上に結果を表示させてみる。

#### PHP Symfowareによるデータベースのアクセス

PHR (Hypertext Preprocessor) は、インタプリタ型のプログラミング言語である。WWWサーバー上で動作し、動的なHTML文書を作成できる。PHPが処理するプログラムのことを「PHPスクリプト」と呼ぶ。

PHPについての詳細は「The PHP Group」の

ホームページ <http://www.php.net/> を参照されたい。

#### 1. Apacheのインストール

The Apache Software Foundationのホームページ <http://www.apache.org/> から、Apacheのアーカイブを入手し、インストールする。

#### 2. PHP のインストール

PHPグループのホームページ <http://www.php.net/downloads.php> から、PHPソースのアーカイブを入手し、インストールする。

ここでは、Apacheモジュール版PHPインタプリタをインストールする(以下もApacheモジュール版PHPインタプリタをインストールしたものとして説明する)。

#### 3. PHP Symfowareのインストール、セットアップ

PHP SymfowareのアーカイブをSymfowareホームページのダウンロードページ <http://software.fujitsu.com/jp/symfoware/download/> より入手する。

スーパーユーザーでログインする。

Symfoware関数のアーカイブを任意の作業ディレクトリ(ここでは/tmpと仮定)に置き、PHP Symfowareソースを展開する。

```
cd /tmp gunzip -c php-symfoware-1.0.0.tar.gz | tar xvf -
```

Symfoware関数ソースディレクトリに移動する。

```
cd /tmp/php-symfoware-1.0.0
```

ビルドの準備を行なう。

```
phpize
```

以下のコマンドでビルドの準備を行なう。

```
./configure --with-symfoware
```

Symfoware関数をビルドする。

```
make
```

PHP Symfoware関数モジュールのディレクトリに移動し、PHP Symfoware関数モジュールを複写する。

```
cd /tmp/php-symfoware-1.0.0/modules
cp php_symfoware.so /usr/local/lib/
php/extensions
```

PHP動作環境ファイルのディレクトリに移動する。

```
cd /usr/local/lib
```

viエディタなどでPHP動作環境ファイル(`php.ini`)を編集し、下記のパラメータを設定する。

```
extension_dir = /usr/local/lib/php/extensions
extension=php_symfoware.so
```

#### 4. Apacheのセットアップ、起動

サンプルスクリプトの実行にはApacheの環境設定が必要である。手順は以下のようになってい

る。

スーパーユーザーでログインする。

Apache動作環境ファイルのディレクトリに移動する。

```
cd /usr/local/apache/conf
```

viエディタなどでApacheの動作環境ファイルを開き、下記のディレクティブの内容を確認する。

```
Port 80
ServerName ホスト名
ScriptAlias /cgi-bin/ "/usr/local/
apache/cgi-bin/"
```

Apache動作環境ファイルの下記のディレクティブの内容を確認する。

```
LoadModule php4_module libexec/libphp4.so
```

Apache動作環境ファイルに下記のディレクティブを設定する。

```
Alias /example/ "/任意の絶対パス/example/"
AddType application/x-httpd-php .php
```

Apache動作環境ファイルにClearModuleListディレクティブが指定されている場合、下記のディレクティブを指定する。

```
AddModule mod_php4.c
```

Apache設定ファイルに下記のディレクティブを設定する。

```
PassEnv LD_LIBRARY_PATH
PassEnv LD_PRELOAD
PassEnv PATH
PassEnv LANG
```

スーパーユーザーの環境変数LD\_LIBRARY\_PATH、LD\_PRELOADを設定する。環境変数LD\_LIBRARY\_PATH、LD\_PRELOADの詳細は、オンラインマニュアル「RDBユーザーズガイド 応用プログラム開発編」の“応用プログラムの起動”を参照していただきたい。

スーパーユーザーの環境変数PATH、LANGを設定する。環境変数PATH、LANGの詳細は、オンラインマニュアル「スタートガイド サーバー編」の“環境変数の設定”を参照していただきたい。

Apache起動コマンドのディレクトリに移動し、Apache動作環境ファイルの妥当性をチェックする。問題がなければApacheを起動する。

```
cd /usr/local/apache/bin
./apachectl configtest
./apachectl start
```

## 5. PHP サンプルスクリプトの実行

ここでは、先ほど定義したデータベースからデータを取り出し、HTMLで表示するというサンプルスクリプトを実行してみる。

以下に、サンプルスクリプトの要点を簡単に説明する。サンプルスクリプトはmnt/cdrom/

linuxsample/Select.ph に格納されている。まず、環境変数「RDBNAME」にRDBシステム名「rdbsys1」を定義する。

```
<HTML>
<BODY>
<PRE><?phpputenv( "RDBNAME=rdbsys1" );
```

次に、データベース「RDBDB」に接続(データベース名はPerlで設定したとおり「RDBDB」)し、接続IDを生成する。

```
$connection_id = symfoware_connect(
"RDBDB", "username", "password" );
```

スクリプト中の、usernameおよびpasswordはデータベースに接続する認可識別子(ユーザー名)とパスワードである。

```
$result_id = symfoware_prepare(
$connection_id, "SELECT *
FROMSTOCKS.STOCK" );
symfoware_execute( $result_id );
?>
```

symfoware\_prepareは、SQL文を準備し、結果IDを返す。ここでは、テーブル「STOCK」の一覧を取得するSQL文を準備し、\$result\_idに結果IDを格納している。実際のSQL文実行は、symfoware\_executeで行なう。

```
<TABLE BORDER="1">
<?
while( symfoware_fetch_row( $result_id ) ) {
?>
<TR>
<?
    for( $x = 1 ;
```

```

    $x <= symfoware_num_fields( $result_id ) ;
    $x ++ ) {
?>
<TD>
<?
    printf( "%s",
    symfoware_result( $result_id, $x ) ) ;
?>
</TD>
<?
    }
?>
</TR>
<?
}
?>
</TABLE>

```

ここで実際にデータベースに入っている情報を1件ずつ取得し、テーブルとして表示する。

symfoware\_fetch\_rowでフェッチ後、symfoware\_num\_fieldsにて列数を取得し、その列数と同じだけforループを回している。

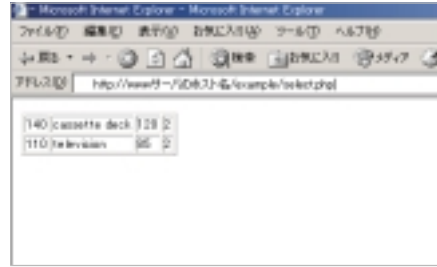
ループ内では、symfoware\_resultで行データを取り出し、表示を行なっている。

最後に、データベースへのコネクションを切断する。

```

<?
symfoware_close( $connection_id ) ;
?>
</BODY>
<PRE>
</HTML>

```



画面7: select.phpの実行

サンプルスクリプトの実行にあたっては「select.php」内の「username」「password」の部分を正しく書き換えることを忘れないで欲しい。

実行した結果は画面7のとおり。

PHP Symfowareによってデータベースから取り出されたデータがHTML文書となり、ブラウザに表示できていることが確認できる。

以上駆け足で説明したが、ご覧のように、PHP/PerlからもSymfowareは簡単に利用できる。PerlはWeb開発者に人気が高く、解説書も豊富であり、PHPはデータベースとの連携を考慮して作成されているため、データベースアクセスの記述が簡単かつ柔軟である。また、PHPは処理が高速なため、Perl/CGIと比較してもサーバー負荷の低減が図れる。

いずれにせよSymfowareとPHP/Perlを組み合わせることで、広範なWebアプリケーションを容易に作成可能になることがお分かりいただけたらう。