



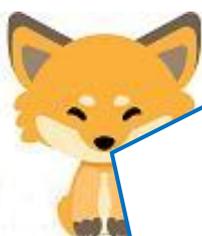
第 19 話 (販売システムⅢ)



タヌキ、次は SQL 文の内の **INSERT** 文を扱うぞ。
 これは、ネットショップにアクセスし、レンタル CD を借りようとする **顧客のデータをデータベースに記憶** することができるのだ。つまり、テーブル `cduser` に顧客データが記録されるんだ。だから **セキュリティも高めなくてはならない** のだ。「`/var/www/html`」のディレクトリは「`chmod 777 . . .`」で解放するなよな。



了解、顧客の情報漏洩がおきたら大変だからな。そうか、重要なディレクトリはハッキングされないようにアクセスできないようにしなければならないのか。



セキュリティには、**chmod** というコマンドを使って、ディレクトリ毎に設定する方法、**SELinux** を使ってログインユーザ毎にセキュリティを設定する方法、**chroot** を使って実際のディレクトリにはアクセスさせない方法などがあるよ。これらの複数の方法を組み合わせてサーバを守るのだ。ただ、セキュリティを高めれば高めるほど、管理 (操作) しづらくなる、というデメリットはあるけどね。でも仕方ないと思うよ。機会があれば、セキュリティの話もするけどね。今はデータベースの解説だ。

最初に `Select` 文でレンタルする CD の有無を確認しなければならないから、実行するプログラムは `rtest02.php` を改良した **`rtest021.php`** になるよ。これを、第 17 話の `rtest01.html` の 5 行目の
`<FORM NAME="myForm" ACTION="rtest021.php" METHOD="get">`
 に組み込むことになる。そして、最初に `rtest01.html` を実行してね。

テーブルへのレコードの追加準備 (Select 文)

rtest021.php(スクリプト例①)

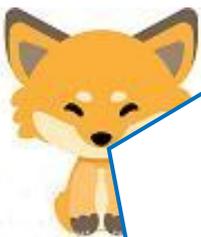
```
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>検索結果</TITLE>
</HEAD>
<BODY>
<?php
$myCode = $_GET['myCode'];
$abc = mysqli_connect("localhost","ftpuser","ftpuser","cdfox");
$query = "select * from cdshohin where scode = '$myCode'";
// クエリを実行します。
if ($result = mysqli_query($abc, $query)) {
    //echo "select に成功しました。 ¥n";
    foreach($result as $row){
        $scode = $row['scode'];
        $sname = $row['sname'];
        $jaketto = $row['jaketto'];
        $music = $row['music'];
    }
}
?>
<table border="1">
<TR>
<TD><?php echo $sname; ?></TD>
ジャケット (画像処理) : <BR><BR>
音楽の視聴 : <BR><audio src="./music/<?php echo $music; ?>" type="audio/mp3" autoplay
controls loop ></audio>
<BR>
</table>
<FORM NAME="isForm" ACTION="rtest03.php" METHOD="get">
<DL>
<DT>顧客番号 (半角 4 桁)
        <DD><INPUT TYPE="text" NAME="userCode" SIZE="4" MAXLENGTH="4" >
</DL>
<DL>
<DT>顧客名 (全角 2 5 文字)
```

rtest021.php(スクリプト例②)

```
<DD><INPUT TYPE="text" NAME="userName" SIZE="12" MAXLENGTH="50">
</DL>
<DT>購入商品番号 (半角 4 文字)
      <DD><INPUT TYPE="text" NAME="shoCode" SIZE="4" MAXLENGTH="4">
</DL>
<DT>購入数量 (半角数字)
      <DD><INPUT TYPE="text" NAME="shoSU" SIZE="4" MAXLENGTH="4">
</DL><BR>
<INPUT TYPE="submit" VALUE="登録">
</FORM>
</BODY>
</HTML>
```



内容は殆ど rtest02.php と同じだな！
ところで、HTML の 3 行目の以下のタグ (<>) は何のおまじないだ。
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">



タヌキ、rtest021.php からさらに rtest03.php を呼び出している点に注意してね。
HTML の 3 行目の以下のタグ (<>) はブラウザで日本語 (全角の漢字やひらがな) が文字化けする時の解決方法の一つだ。ブラウザ側で常に UTF-8 にエンコード (文字変換) しなければならない時には、<HEAD> と <TITLE> タグの間に以下の文字コード指定のタグを挿入するのだ。
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
これを挿入しても UTF-8 に固定されない場合は、php.ini ファイル中のデフォルト文字コードが UTF-8 以外 (EUC や Shift-JIS) になっていることが考えられる。こうなったら、最初からデータベースの作り直しになるかもしれない。



右図が `rtest021.php` の実行結果か。
顧客番号 (GK11)、顧客名 (東京太郎)、
[新世界] の購入商品番号 (A002)、
購入数量 (2) をテキストボックスに
入力し、[登録] ボタンをクリックす
ると4種類のデータが `rtest03.php` に
送信されるのか。よくできているね。
でも、キツネ、オイラ「タヌキ太郎」
だぞ。これだとオイラのなりすましに
なるぞ。なりすましを防ぐ方法を考え
なければならないのでは？

`rtest021.php` の実行結果

select に成功しました。ジャケット (画像処理) :



音楽の視聴 :

顧客番号 (半角4桁)

顧客名 (全角25文字)

購入商品番号 (半角4文字)

購入数量 (半角数字)



タヌキ、良いところに気が付いたな。
その通り、これではいくらでもなりすましをすることが
できるのだ。
その人物を一意に特定できる項目を考えなければならない。
ネット上では、電話番号も住所も偽ることができるので、
メールアドレスが一番確実なのだ。メールを偽ると
確認メールが届かないからね。それに、携帯の番号を
組み合わせると、よりセキュリティが高まるけどね。
では、次は今回のテーマの **INSERT 文のプログラムの
提示だ。**

テーブルへのレコードの追加 (Insert into 文)

rtest03.php (スクリプト例)

```
<HTML>
<HEAD>
<TITLE>データベースへの登録と顧客への通知</TITLE>
</HEAD>
<BODY>
<?php
$userCode = $_GET['userCode'];
$username = $_GET['userName'];
$shoCode = $_GET['shoCode'];
$shoSu = $_GET['shoSu'];
$conid1 = mysqli_connect("localhost","ftpuser","ftpuser","cdfox");
// 接続状況をチェックします
if (mysqli_connect_errno()) {
    die("データベースに接続できません:" . mysqli_connect_error() . "\n");
}

echo "データベースの接続に成功しました。 \n";
$query = "INSERT INTO cduser (ucode, uname, rireki, kazu) VALUES
('$userCode','$userName','$shoCode','$shoSu)";
// クエリを実行します。
if (mysqli_query($conid1, $query)) {
    echo "INSERT に成功しました。 \n";
}
mysqli_close($conid1);
?>
<BR>
cduser テーブルへ 1 件の顧客レコードを登録しました！<br>
</BODY>
</HTML>
```



rtest03.php の実行結果は、次ページの図のようになるぞ。

[実行結果]



キツネ、上図は、ブラウザに表示される内容だよな。
データベースのテーブル cduser に記憶される顧客（東京太郎）の確認はどうするの？



そうだなよな、データベースの確認はしなければならないな。
第18話で説明した操作の繰り返しだ。一応手順を再度記入しておくな。

- ① # mysql -u root -p (データベースサーバへのログイン)
 - ② MariaDB [(none)]>use cdfox; (cdfox データベースの使用)
 - ③ >show tables; (cduser テーブルの確認)
 - ④ >select * from cduser; (テーブル内容の表示)
- 下図のようになっていれば OK だ。

```
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
Reading table information for completion of t
You can turn off this feature to get a quicke

Database changed
MariaDB [ cdfox ]> show tables;
+-----+
| Tables_in_cdfox |
+-----+
| cdshohin         |
| cduser           |
+-----+
2 rows in set (0.01 sec)

MariaDB [ cdfox ]> select * from cduser;
+-----+-----+-----+-----+
| ucode | uname      | rireki | kazu |
+-----+-----+-----+-----+
| GK11  | 東京 太郎  | A002   | 2    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [ cdfox ]> exit
Bye
```



おもしろいな！勉強になったぞ。



次は、SQL 文の UPDATE (更新) と DELETE
(削除) のプログラムだ。

それを **第 20 話** で行うよ。