Episode 19 (Sales System III)

Tanuki, next we will deal with an INSERT statement within an SQL statement.

This allows us to store in a database the data of a customer who accesses an online store and tries to rent a CD. In other words, the table cduser will be used to store the customer data. So we need to increase security. Don’t release the directory ”/var/www/html” by ”chmod 777 ...”, right?

Roger, it would be a disaster if a customer’s information was compromised. Oh, I see, so you have to make sure that important directories are inaccessible to prevent hacking.

Security can be configured on a per－directory basis using the chmod command, on a per－logged－in user basis using SELinux, or by not allowing access to the actual directory using chroot. A combination of these methods is used to protect the server. However, there is a disadvantage: the higher the security, the harder it is to manage (operate) the server. But I think it can’t be helped. I will talk about security when I have a chance. For now, let’s talk about databases.

First, we have to check the availability of CDs to rent with a Select statement, so the program to run will be rtest021.php, which is a modified version of rtest02.php.

This will be incorporated into line 5 <FORM NAME=”myForm” ACTION=”rtest021.php” METHOD=”get”> of rtest01.html in episode 17.

And run rtest01.html first.

Prepare to add records to the table (Select statement)

rtest021.php (Script example ①)

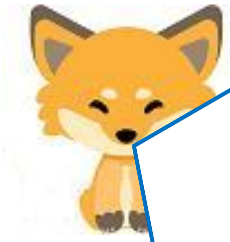
```
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE> search results</TITLE>
</HEAD>
<BODY>
<?php
$myCode = $_GET['myCode'];
$abc = mysqli_connect("localhost","ftpuser","ftpuser","cdfox");
$query = "select * from cdshohin where scode = '$myCode';";
    // Execute the query
if ($result = mysqli_query($abc, $query)) {
    //echo " select succeeded.   ¥n";
   foreach($result as $row){
       $scode = $row['scode'];
       $sname = $row['sname'];
       $jaketto = $row['jaketto'];
       $music = $row['music'];
    }
}
?>
<table border="1">
<TR>
<TD><?php echo $sname; ?></TD>
Jacket (image processing):<BR><img src="./jaketto/<?php echo $jaketto; ?>"><BR>
Music listening:<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> Customer Number (4-digit)
        <DD><INPUT TYPE="text" NAME="userCode" SIZE="4" MAXLENGTH="4" >
</DL>
<DL>
<DT> Customer name (25 characters)
```

rtest021.php (Script example ②)

```
        <DD><INPUT TYPE="text" NAME="userName" SIZE="12" MAXLENGTH="50">
</DL>
<DT> Purchase item number (4 characters)
        <DD><INPUT TYPE="text" NAME="shoCode" SIZE="4" MAXLENGTH="4">
</DL>
<DT> Purchase quantity (single-byte number)
        <DD><INPUT TYPE="text" NAME="shoSu" SIZE="4" MAXLENGTH="4">
</DL><BR>
<INPUT TYPE="submit" VALUE=" Registration">
</FORM>
</BODY>
</HTML>
```

The content is almost the same as rtest02.php!
By the way, what spells the following tag (< >) in the third line of HTML.

```
<meta http-equiv="Content-Type"
                    content="text/html; charset=UTF-8">
```



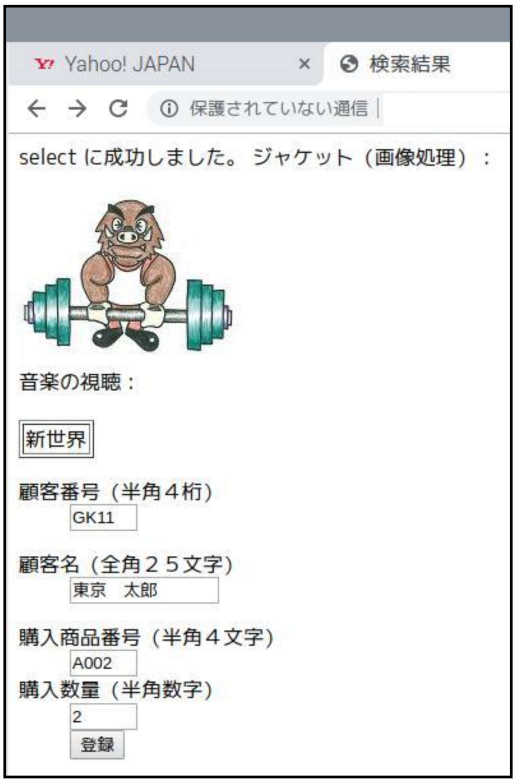
Tanuki, note that rtest03.php is called from rtest021.php.
The following tag (< >) in the third line of HTML is one of the solutions for Japanese (full−width Kanji and Hiragana) garbled characters in the browser.
When the browser must always encode (convert characters) to UTF−8, Insert the following character encoding tags between <HEAD> and <TITLE> tags.

```
<meta http-equiv="Content-Type"
                    content="text/html; charset=UTF-8">
```

If it is not fixed to UTF−8 after inserting this, the default character encoding in the php.ini file may be something other than UTF−8 (EUC or Shift−JIS). If this happens, you may have to rebuild the database from scratch.

The right figure shows the result of executing rtest021.php.
If I enter the customer number (GK11), customer name (Tokyo Taro), [Shinsegae] purchase item number (A002), and purchase quantity (2) in the text boxes and click the [Register] button, four types of data will be sent to rtest03.php? Well done.
But, fox, my name is "Taro Tanuki". This would be an impersonation of me. I think we need to figure out a way to prevent impersonation.

Execution result of rtest021.php



select に成功しました。ジャケット（画像処理）：

音楽の視聴：

新世界

顧客番号（半角４桁）
GK11

顧客名（全角２５文字）
東京　太郎

購入商品番号（半角４文字）
A002

購入数量（半角数字）
2

登録

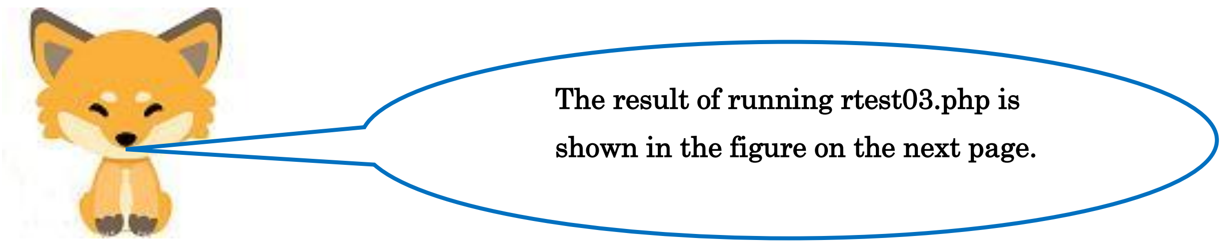
Tanuki, you've got a good point.
You are right, this allows for any number of impersonations. You have to come up with an item that uniquely identifies the person. On the Internet, both phone numbers and addresses can be faked, so an email address is the most reliable. If you fake an email, you won't get a confirmation email. Combine that with a cell phone number, though, and you have even more security.
Next, I will present the program for the INSERT statement, which is the subject of this article.

Adding records to a table (Insert into statement)

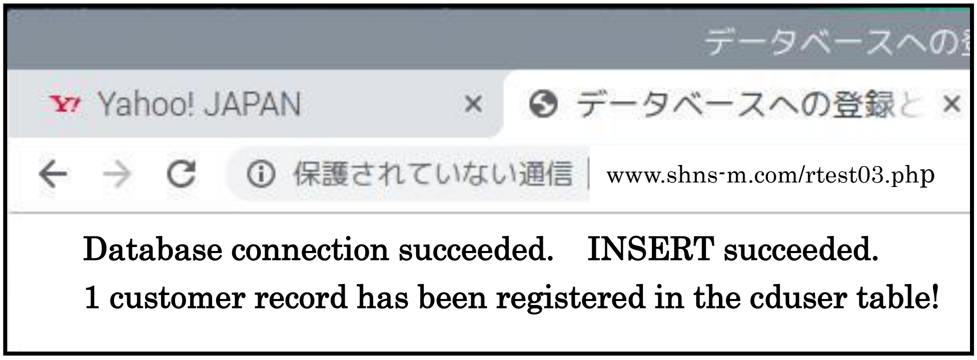
rtest03.php (example script)

```
<HTML>
<HEAD>
    <TITLE> Register in database and notify customers</TITLE>
</HEAD>
<BODY>
<?php
$userCode = $_GET['userCode'];
$userName = $_GET['userName'];
$shoCode = $_GET['shoCode'];
$shoSu = $_GET['shoSu'];
$conid1 = mysqli_connect("localhost","ftpuser","ftpuser","cdfox");
        // Check the connection status
  if (mysqli_connect_errno()) {
              die("Unable to connect to database:" . mysqli_connect_error() . "¥n");
  }


        echo " Database connection succeeded.¥n";
$query = "INSERT INTO cduser (ucode, uname, rireki, kazu) VALUES
('$userCode','$userName','$shoCode','$shoSu');";
      // Execute the query.
  if (mysqli_query($conid1, $query)) {
          echo " INSERT succeeded.¥n";
  }
mysqli_close($conid1);
?>
<BR>
1 customer record has been registered in the cduser table!<br>
</BODY>
</HTML>
```


The result of running rtest03.php is shown in the figure on the next page.

［The result of running］



Database connection succeeded.　INSERT succeeded.

1 customer record has been registered in the cduser table!

Kitsune, the above diagram is what is displayed in the browser, right? How do you confirm the customer (Tokyo Taro) stored in the table cduser in the database?

Right, right, we have to check the database. Repeat the procedure described in Episode 18. Let's fill in the steps again.

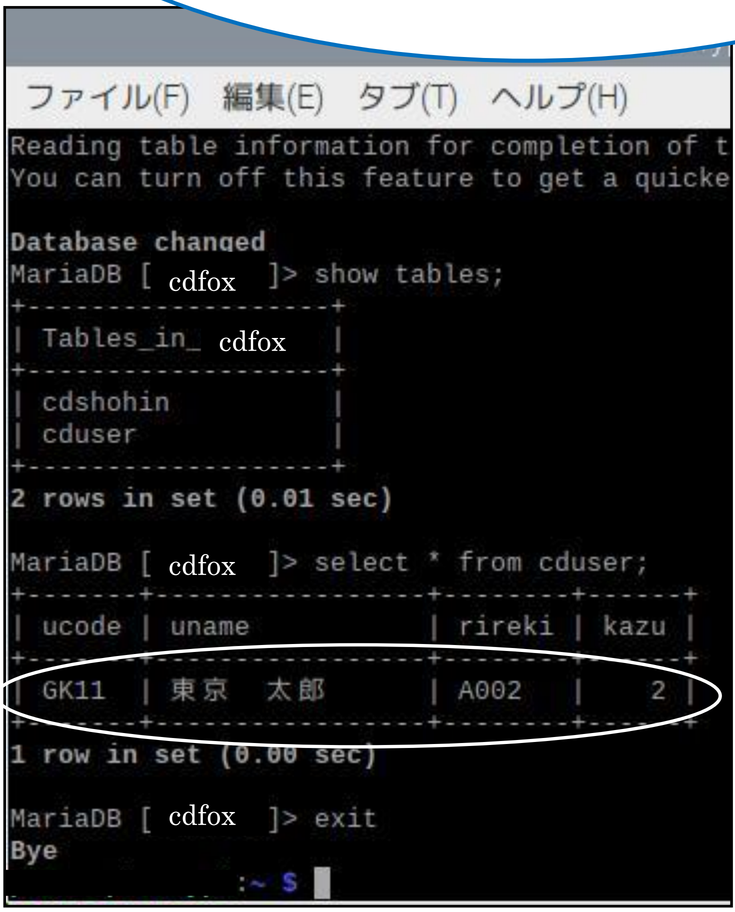
①　# mysql -u root -p (login to the database server)

②MariaDB [(none)]>use cdfox ; (use cdfox database)

③>show tables; (check cduser table)

④>select * from cduser; (display table contents)

If it looks like the figure below, it is OK.



```
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
:~ $
```

Interesting! I've learned a lot.

Next up is the UPDATE (update) and DELETE (delete) program of SQL statements.

We'll do that in Episode 20

Translated at DeepL