

# 5.パスワードのハッシュ化

パスワードをハッシュ化します（ランチオーダーシステム）

## 5.1 empテーブルの再構築

パスワードをsha-512でハッシュ化する場合empテーブルのpassword項目には128文字分の領域が必要になります。ところがDBの初期構築時にはvarcharで10文字分しか確保していません。よって、テーブルを削除して新たに作成しなおします。

```
[root@aso lunch]# mysql -u testuser -p
```

```
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 84
```

```
Server version: 8.0.32 MySQL Community Server - GPL
```

```
Copyright (c) 2000, 2023, Oracle and/or its affiliates.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input  
statement.
```

```
mysql> use testdb
```

```
Reading table information for completion of table and column  
names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
mysql> select * from emptbl;
```

```
+-----+-----+-----+
```

```
| empid | empname | password |
```

```
+-----+-----+-----+
```

```
| 1     | watanabe | watanabe |
```

```
| 2     | kodama   | kodama   |
```

```
| 3     | tanimoto | tanimoto |
```

```
| 4      | ohuchi  | ohuchi  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

mysql>

mysql> **drop table emptbl;**

Query OK, 0 rows affected (0.05 sec)

mysql> **create table emptbl(**

**-> empid varchar(10) not null primary key,**

**-> empname varchar(30) not null,**

**-> password varchar(130)**

**-> );**

Query OK, 0 rows affected (0.08 sec)

mysql>

mysql> **insert into emptbl**  
**values('1','watanabe','watanabe');**

Query OK, 1 row affected (0.02 sec)

mysql> **insert into emptbl values('2','kodama','kodama');**

Query OK, 1 row affected (0.01 sec)

mysql> **insert into emptbl values('3','tanimoto','tanimoto');**

Query OK, 1 row affected (0.02 sec)

mysql> **insert into emptbl values('4','ohuchi','ohuchi');**

Query OK, 1 row affected (0.01 sec)

mysql> **select \* from emptbl;**

```
+-----+-----+-----+
| empid | empname | password |
+-----+-----+-----+
| 1     | watanabe | watanabe |
| 2     | kodama   | kodama   |
| 3     | tanimoto | tanimoto |
| 4     | ohuchi   | ohuchi   |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

mysql>

**一応テーブルスキーマを参照して列幅を確認します**

```
mysql> select  TABLE_NAME,COLUMN_NAME,COLUMN_TYPE
from        INFORMATION_SCHEMA.COLUMNS WHERE
```

```
TABLE_SCHEMA = 'testdb';
```

| TABLE_NAME    | COLUMN_NAME     | COLUMN_TYPE         |
|---------------|-----------------|---------------------|
| emptbl        | empid           | varchar(10)         |
| emptbl        | empname         | varchar(30)         |
| <b>emptbl</b> | <b>password</b> | <b>varchar(130)</b> |
| lunchorder    | orderid         | bigint unsigned     |
| lunchorder    | orderdate       | date                |
| lunchorder    | empid           | int                 |
| lunchorder    | lunchid         | int                 |
| lunchtbl      | lunchid         | bigint unsigned     |
| lunchtbl      | lunchname       | varchar(40)         |
| lunchtbl      | unitprice       | int                 |
| lunchtbl      | validflag       | tinyint(1)          |
| testtbl       | id              | int                 |
| testtbl       | name            | varchar(20)         |

13 rows in set (0.01 sec)

mysql> **exit**

Bye

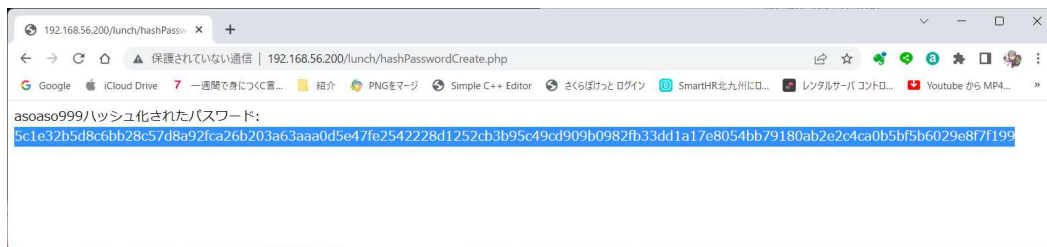
## 5.2 ハッシュ化アプリ作成

```
[root@aso lunch]# vi hashPasswordCreate.php
```

```
1 <?php
2 $password = "asoaso999";
3
4 $hashedPassword = hash('sha512', $password); // パスワードをハッシュ化
5 echo $password;
6 echo "ハッシュ化されたパスワード: " . $hashedPassword;
7 ?>
8
```

## 5.3 Webから確認

<http://192.168.56.200/lunch/hashPasswordCreate.php>



ハッシュ値をマウスで選択してメモ帳などにコピーします

→ その後以下で作成する hashPasswordUpdate.php 内にコピーします

## 5.4 emptbl修正

emptbl内のpassword項目をasoaso999のハッシュ値に置き換えます。  
(簡素化するために全員同じパスワードにします)

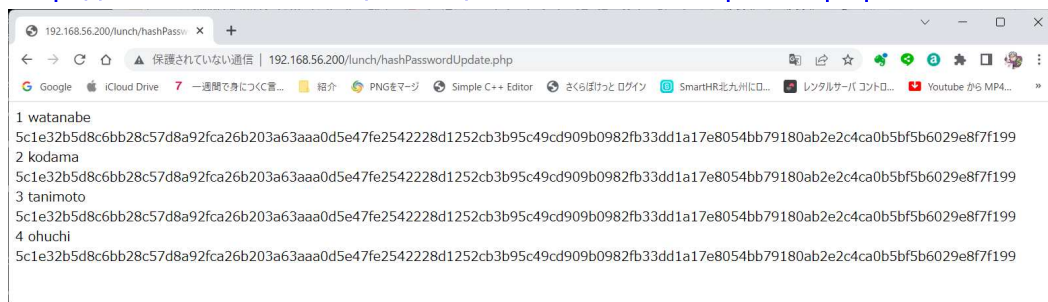
```
[root@aso lunch]# vi hashPasswordUpdate.php
```

A screenshot of a terminal window with a dark background. The title bar shows '192.168.56.200 - root@aso:/var/www/html/lunch VT'. The terminal displays the PHP code for 'hashPasswordUpdate.php'. The code connects to a MySQL database, updates the 'password' field in the 'emptbl' table with a specific hash, and then queries the table to display all records. The code is as follows:

```
1 <?php
2 // お弁当システム パスワードハッシュ化
3 // データベースに接続し、クエリを実行して結果を配列に保存する
4 error_reporting(0);
5
6 $conn = mysqli_connect('localhost', 'testuser', 'P@ssP@ss456', 'testdb' );
7 $sql = "update emptbl set password = '5c1e32b5d8c6bb28c57d8a92fca26b203a63aaa0d5e47fe2542228d1252cb3b95c49cd909b0982fb33dd1a17e8054bb79180ab2e2c4ca0b5bf5b6029e8f7f199'";
8
9 $result = mysqli_query($conn, $sql);
10
11 $sql = "select * from emptbl";
12 $result = mysqli_query($conn, $sql);
13 $rec_num = mysqli_num_rows($result);
14
15 $indexedArray = array(); // 結果保存用配列
16
17 for ($i=0;$i<$rec_num;$i++) {
18     $arr = mysqli_fetch_array($result,MYSQLI_ASSOC);
19     $indexedArray[] = array_values($arr); // 連想配列を2次元配列として保存
20 }
21
22 foreach ($indexedArray as $item) {
23     foreach ($item as $value) {
24         echo $value . " ";
25     }
26     echo "<br>";
27 }
28
29 ?>
30
```

## 5.5 Webから確認

<http://192.168.56.200/lunch/hashPasswordUpdate.php>

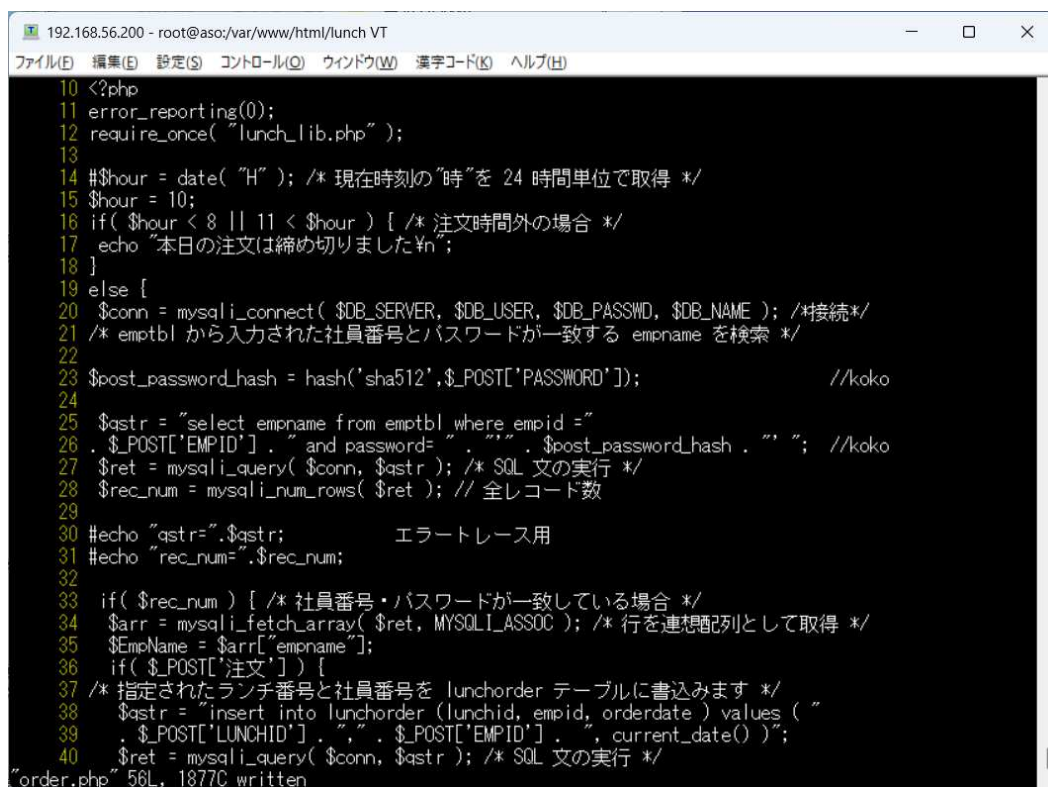


## 5.6 ハッシュ対応にプログラムを修正

order.phpのパスワードチェック部分をhash対応に変更します。

```
[root@aso lunch]# vi order.php
```

パスワードチェック部分を2か所（//kokoの部分）変更します。以下のスクショでは23行目と26行目。



## 5.7 Webから確認

パスワードがasoaso999で全員注文できますか？

<http://192.168.56.200/lunch/lunch.php>

The screenshot shows a web browser window displaying the 'お弁当注文システム (オーダーフォーム)' (Lunch Ordering System (Order Form)). The page has a title bar with 'お弁当注文システム' and a single tab. The address bar shows '192.168.56.200/lunch/lunch.php'. The form includes a '社員番号' (Employee Number) dropdown menu with '1 watanabe' selected, and a 'パスワード' (Password) input field with '\*\*\*\*\*' displayed. Below these is a table of lunch options with radio buttons for selection:

|                                         |       |
|-----------------------------------------|-------|
| <input type="radio"/> 日の丸弁当             | 300円  |
| <input checked="" type="radio"/> おにぎり弁当 | 350円  |
| <input type="radio"/> たこやき弁当            | 400円  |
| <input type="radio"/> ラーメン弁当            | 500円  |
| <input type="radio"/> 三色弁当              | 800円  |
| <input type="radio"/> 豚ノ内弁当             | 1500円 |
| <input type="radio"/> しゃぶしゃぶ弁当          | 1500円 |
| <input type="radio"/> 特製活き造り弁当          | 2000円 |

At the bottom of the form are two buttons: '注文' (Order) and 'クリア' (Clear). A small '戻る' (Back) link is visible in the bottom left corner.

## 5.8 バグフィックス

お弁当を選択せずに注文ボタンを押すと以下のようにエラーとなります。明らかにバグですので修正してください。

This screenshot shows the same web form as in the previous image, but with an error message displayed. The '社員番号' (Employee Number) dropdown menu now shows '1 watanabe' and the 'パスワード' (Password) input field is empty. The table of lunch options is the same. However, the '注文' (Order) button is disabled, and a red error message is visible at the bottom left, indicating a problem with the order process. The '戻る' (Back) link is also present.

LUNCHIDの入力チェックが必要です！

