MYSQL

DBの基本 #3

PHPからMySQLを利用

```php
<?php
$username="xxxxx";
$password="yyyyy";
$dbname  ="zzzzz";

try{
    $pdo = new PDO("mysql:host=localhost;dbname=$dbname;charset=utf8",$username,$password);
    $sql = "SELECT * FROM tb1";
    $result = $pdo -> query($sql);

//  while($kekka=$result -> fetch(PDO::FETCH_ASSOC)){
    foreach ($result as $kekka){
        var_dump($kekka);
        echo "<br>\n";
    }
}catch(PDOException $e){
    print("Error:" . $e->getMessage());
    die();
}
$result = null;
$pdo = null;
?>
```

← 別ファイルから読み込む

PHPからMySQLを利用

```
<?php
require "../../pass.php";
```

←public_htmlの外に置く

又は

```
include "../../pass.php";
```

```
<?php
$username="xxxxx";
$password="yyyyy";
$dbname  ="zzzzz";
?>
```

●読み込めない時

require → Fatal Error

include → Warning

●関連機能

require_once , include_once

どちらも言語機能のため（）は不要。

書いてもほぼ問題は起きない

別ファイルにするメリット

- PHPが動作しない場合、すべてのコードが表示される。
  - public_htmlより上位に置くことで、見られることがない

- テスト環境・本番環境でのDBの相違を吸収
  - ローカルや開発環境と、本番のDBでユーザ名・パスワードが異なっていても、変更が不要

具体例

pass.php

```
<?php
    $username="xyz";
    $password="123";
    $dbname="test";
?>
```

ローカル実行環境

pass.php

```
<?php
    $username="abc";
    $password="999";
    $dbname="db1";
?>
```

サーバー実行環境

```
<?php
require "pass.php";

try{
    $pdo = new PDO("mysql:host=localhost;dbname=$dbname;charset=utf8",$username,$password);
    $sql = "SELECT * FROM tb1";
    $result = $pdo -> query($sql);

//  while($kekka=$result -> fetch(PDO::FETCH_ASSOC)){
    foreach ($result as $kekka){
```

前回の課題

- 自分のdbにあるテーブルtb1に含まれるデータを表形式にして、全て出力しなさい。
- プログラム：public_html/sql02/select.php

- テーブルの出力は関数にしてみよう。（db_output）

mysqliオブジェクト型

解答例 db_output()

```
function db_output($result){
    echo "<table>\n\t<tr>";
    $info = $result -> fetch_fields();  // field名の取得
    foreach ($info as $f){       // field名の表示
      echo "<th>".$f->name."</th>";
    }
    echo "</tr>\n";
    while($kekka=$result -> fetch_array(MYSQLI_ASSOC)){ //取出し
      echo "\t<tr>";
      foreach($kekka as $f){
          echo "<td>{$f}</td>" ;
      }
      echo "</tr>\n";
    }
    echo "</table>\n";
}
```

mysqliオブジェクト型

解答例 main部分

※html部分省略

```
<?php
function db_output($result){
    :
}
// main
$mysqli = new mysqli("localhost",$username,$password);
if($mysqli->connect_error){
  print('<p>データベースへの接続に失敗しました</p>'.$mysqli->connect_error);
  exit();
}
$mysqli->select_db($dbname);              //useと同等
$mysqli->set_charset("utf-8");
$result = $mysqli->query("select * from tb1");
db_output($result);
$result->free();
$mysqli->close();
?>
```

PDO

解答例 db_output()

```
function db_output($result){
    $column_name = array();
    for($i=0;$i<$result->columnCount();$i++){  // ヘッダ行の取り出し
        // $meta = $result->getColumnMeta($i);
        // $column_name[]=$meta['name'];
        $column_name[] = $result->getColumnMeta($i)['name'];
    }

    echo "<table>¥n";
    echo "¥t<tr><th>".implode("</th><th>",$column_name)."</th></tr>¥n";

//  while($td=$result -> fetch(PDO::FETCH_ASSOC)){
    foreach ($result as $kekka){
        $td=array();  // whileの場合は不要
        for($i=0;$i<$result->columnCount();$i++){
            $td[]=$kekka[$i];
        }
        echo "¥t<tr><td>".implode("</td><td>",$td)."</td></tr>¥n";
    }
    echo "</table>¥n";
}
```

PDO

解答例 main部分

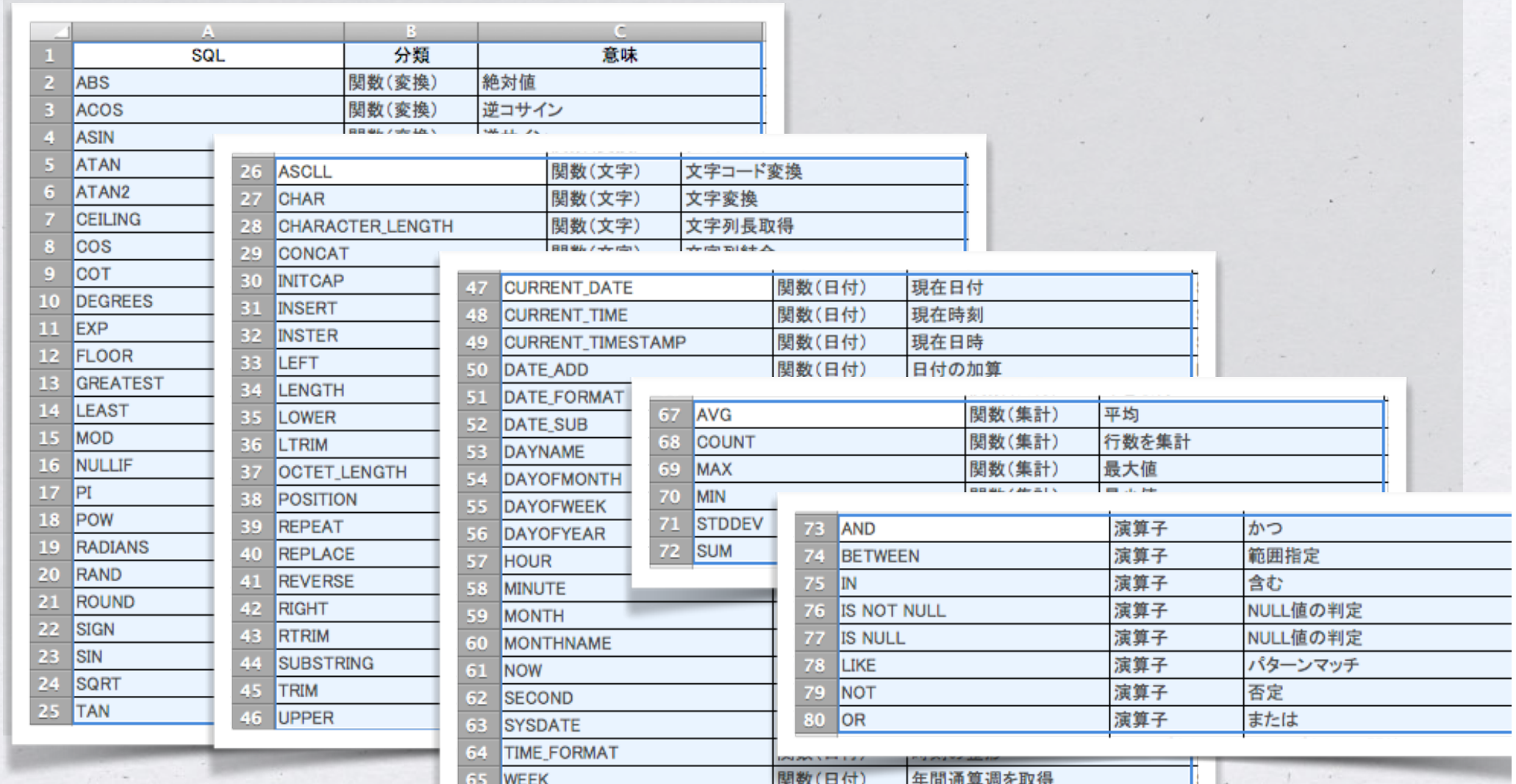
※html部分省略

```
<?php
function db_output($result){
   :
}
// main
require "../../pass.php";

try{
      $pdo = new PDO("mysql:host=localhost;dbname=$dbname;charset=utf8",
                                                $username,$password);
      $sql="select bang as '番号',nama as '名前',tosi as '年齢' from tb1";
      $result = $pdo->query($sql);
      db_output($result);
}catch(PDOException $e){
      print("Error:" . $e->getMessage());
      die();
}
$result = null;
$pdo = null;
?>
```

MySQLの演算子・関数

http://dev.mysql.com/doc/refman/5.6/ja/functions.html



	A	B	C
1	SQL	分類	意味
2	ABS	関数(変換)	絶対値
3	ACOS	関数(変換)	逆コサイン
4	ASIN		
5	ATAN		
6	ATAN2		
7	CEILING		
8	COS		
9	COT		
10	DEGREES		
11	EXP		
12	FLOOR		
13	GREATEST		
14	LEAST		
15	MOD		
16	NULLIF		
17	PI		
18	POW		
19	RADIANS		
20	RAND		
21	ROUND		
22	SIGN		
23	SIN		
24	SQRT		
25	TAN		
26	ASCLL	関数(文字)	文字コード変換
27	CHAR	関数(文字)	文字変換
28	CHARACTER_LENGTH	関数(文字)	文字列長取得
29	CONCAT		
30	INITCAP		
31	INSERT		
32	INSTER		
33	LEFT		
34	LENGTH		
35	LOWER		
36	LTRIM		
37	OCTET_LENGTH		
38	POSITION		
39	REPEAT		
40	REPLACE		
41	REVERSE		
42	RIGHT		
43	RTRIM		
44	SUBSTRING		
45	TRIM		
46	UPPER		
47	CURRENT_DATE	関数(日付)	現在日付
48	CURRENT_TIME	関数(日付)	現在時刻
49	CURRENT_TIMESTAMP	関数(日付)	現在日時
50	DATE_ADD	関数(日付)	日付の加算
51	DATE_FORMAT		
52	DATE_SUB		
53	DAYNAME		
54	DAYOFMONTH		
55	DAYOFWEEK		
56	DAYOFYEAR		
57	HOUR		
58	MINUTE		
59	MONTH		
60	MONTHNAME		
61	NOW		
62	SECOND		
63	SYSDATE		
64	TIME_FORMAT		
65	WEEK	関数(日付)	年間通算週を取得
67	AVG	関数(集計)	平均
68	COUNT	関数(集計)	行数を集計
69	MAX	関数(集計)	最大値
70	MIN		
71	STDDEV		
72	SUM		
73	AND	演算子	かつ
74	BETWEEN	演算子	範囲指定
75	IN	演算子	含む
76	IS NOT NULL	演算子	NULL値の判定
77	IS NULL	演算子	NULL値の判定
78	LIKE	演算子	パターンマッチ
79	NOT	演算子	否定
80	OR	演算子	または

DBにデータを格納

- 空のtb1と同じ構造のtable（tb1X）を作成する。
  はじめはデータ無しでOK。（手作業で mysqlコマンド）

- 以下、PHPで作成（sql03/insert.php）
- formから番号・名前・年齢を入力し、tbl1Xへ格納する。
  エラー処理は省略しても良い。（可能なら実装）
- 格納したデータを表形式（db_output）で出力する。
- サンプル&動作テストとして、10件以上入力すること。

起こりうる問題

- これまでの流れ&知識で、作成すると以下の問題が発生しやすい。
  - データ登録後F５（リロード）すると、フォームデータの再送信が行われ（確認画面あり）、同一データが登録されてしまう。
  - これを防ぐには、どうしたら良いだろうか？ 検討して欲しい。
    - 案のみ
    - 実装例