

PHP 練習問題. 08 データベース処理と SQL

(参考資料)

PHP からデータベースに接続して SQL を実行する方法です。

- 共通部分

select、insert、update、delete で共通の部分です。

// データベースに接続するための文字列 (DSN 接続文字列)

```
$dsn = 'mysql:dbname=php_work;host=localhost;charset=utf8';
```

// PDO クラスのインスタンスを作ります。

// 引数は、上記の DSN、データベースのユーザー名、パスワード

// XAMPP の場合はデフォルトでパスワードなし、MAMP の場合は「root」

```
$dbh = new PDO($dsn, 'root', ''); // XAMPP の場合
```

```
$dbh = new PDO($dsn, 'root', 'root'); // MAMP の場合
```

// エラーが起きたときのモードを指定します。

// エラーのときは例外 (練習問題 09 で学習します) を発生させます。

```
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

- select のとき

// SQL 文を作成します。

```
$sql = 'select * from todo_items ';
```

```
$sql .= 'where id=:id ';
```

// 「:id」の部分は「パラメータ」といいます。

// SQL 文を実行する準備します。

```
$stmt = $dbh->prepare($sql);
```

// SQL 文のパラメータに値を割り当てます (「バインド」するといいます)。

```
$stmt->bindValue(':id', $id, PDO::PARAM_INT);
```

// バインドする値が整数型の場合は第3引数に「PDO::PARAM_INT」を指定します。

// 文字列、日付型の文字列、小数などの場合は「PDO::PARAM_STR」を指定します。

```

// SQL 文を実行します。
$stmt->execute();

// fetchAll()というメソッドを使って、SQL を実行した結果のレコードを連想配列で取得
します。
// 連想配列を HTML に表示する方法は、「練習問題 04 配列」を参照してください。
$list = $stmt->fetchAll(PDO::FETCH_ASSOC);

// 連想配列がどのような形になっているかについては、
// 下記のように書いて確認してみましょう。
echo '<pre>';
var_dump($list);
echo '</pre>';

● insert、update、delete のとき
// SQL 文を作成します。
$sql = 'insert into todo_items (';
$sql .= 'expiration_date,';
$sql .= 'todo_item'
$sql .= ') values (';
$sql .= ':expiration_date,';
$sql .= ':todo_item';
$sql .= ')';

// SQL 文を実行する準備します。
$stmt = $dbh->prepare($sql);

// SQL 文のパラメータに値を割り当てます（「バインド」するといいます）。
$stmt->bindValue(':expiration_date', $expiration_date, PDO::PARAM_STR);
$stmt->bindValue(':todo_item', $todo_item, PDO::PARAM_STR);
// バインドする値が整数型の場合は第3引数に「PDO::PARAM_INT」を指定します。
// 文字列、日付型の文字列、小数などの場合は「PDO::PARAM_STR」を指定します。

// SQL 文を実行します。
$stmt->execute();

```

設問1. 練習問題 08 では、簡単な TODO リストの Web アプリを作成します。
練習問題で使用する下記のデータベースとテーブルを MySQL に作成してください。(XAMPP の MySQL で OK です)

データベース名： php_work
テーブル名： todo_items
デフォルトの文字コード： utf8mb4_general_ci

名前	データ型	Null 不許可	デフォルト 値		備考
id	INT	yes		Primary Key AUTO_INCREMENT	ID
expiration_date	DATE	yes			期限日
todo_item	VARCHAR(100)	yes			TODO 項目
is_completed	TINYINT	yes	0		完了

✓ is_completed： 0 のとき未完了、1 のとき完了

設問2. 下記のフォームを index.php に作成してください。

- (1) 日付のテキストボックスの name 属性は「expiration_date」に、
TODO 項目の name 属性は「todo_item」にします。
✓ テキストボックスの name 属性は、データベースのフィールド名に
しておくと、後々楽です。
- (2) 日付のテキストボックスのデフォルトの日付は、日本標準時の「今
日」にします。
- (3) フォームの action 属性は「add.php」にします。

● index.php

TODOリスト

2020/02/25

TODO項目を入力してください

追加

(ヒント)

- (1) HTML5 の場合、日付を入力するテキストボックスは、type 属性を「date」にすると、カレンダーを表示させることができます。テキストボックスにマウスのカーソルを合わせると「▼」ボタンが現れます。それをクリックすると、カレンダーが表示されます。

2020/02/25

TODO項

2020年(令和2年) 2月

日	月	火	水	木	金	土
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

- (2) 「date」属性のテキストボックスにデフォルト値を設定するときは、日付の区切りを「/」ではなく、「-」（ハイフン）にします。










```
<input type="date" value="2020-02-25">
```

- (3) テキストボックスに「入力例」（プレースホルダー）を表示するには、placeholder 属性を使います。

```
<input type="text" placeholder="TODO リストを入力してください">
```

設問3. add.php を新規作成してください。add.php では、下記の処理を行います。

- (1) データベースに接続します。
- (2) inde.php から POST された値で todo_items テーブルに新規レコードをインサートします。
- (3) 処理終了後、index.php にリダイレクトします。
- (4) phpMyAdmin などを使って、レコードが正しくインサートされていることを確認してください。

← T →						id	expiration_date	todo_item	is_complete	
<input type="checkbox"/>		編集		コピー		削除	1	2020-02-25	テストの項目 1	0
<input type="checkbox"/>		編集		コピー		削除	2	2020-08-25	テストの項目 2	0
<input type="checkbox"/>		編集		コピー		削除	3	2020-05-25	テストの項目 3	0

設問4. index.php で、todo_lists テーブルに追加されているレコードを全件取得して、一覧表示する部分を追加します。

- (1) table タグを使って一覧表示します。
- (2) 期限日の古いものから順番に表示します。
- (3) レコードが 1 件もないときは、何も表示しません。

- index.php

TODOリスト

2020/02/25

TODO項目を入力してください

追加

期限日	TODO項目
2020-02-25	テストの項目 1
2020-05-25	テストの項目 3
2020-08-25	テストの項目 2

設問5. index.php に form を追加し、TODO の項目の「完了・未完了」を切り替えるラジオボタン、削除するためのチェックボックスを作ります。

- (1) レコード（行）ごとに form が作成されるようにします。
- (2) form の POST 先は action.php にします。
- (3) `<input type="hidden">`を使って、レコード（行）の id の値を POST できるようにします。
- (4) ラジオボタンの値は、未完了のときは「0」、完了のときは「1」にします。
- (5) チェックボックスの値は、1 にします。

● index.php

TODOリスト

2020/02/25

TODO項目を入力してください

追加

期限日	TODO項目	
2020-02-25	テストの項目 1	<input type="radio"/> 未完了 <input type="radio"/> 完了 <input type="checkbox"/> 削除 <div>実行</div>
2020-05-25	テストの項目 3	<input type="radio"/> 未完了 <input type="radio"/> 完了 <input type="checkbox"/> 削除 <div>実行</div>
2020-08-25	テストの項目 2	<input type="radio"/> 未完了 <input type="radio"/> 完了 <input type="checkbox"/> 削除 <div>実行</div>

設問6. action.php を新規追加します。action.php では、下記の処理を行ってください。

- (1) データベースに接続します。
- (2) index.php から POST された値で
 - ✓ 削除のチェックボックスにチェックが入っているときは、該当の id のレコードを削除します。
 - ✓ 削除のチェックボックスにチェックが入っていないときは、該当 id のレコードをアップデートします。
- (3) 処理終了後、index.php にリダイレクトします。
- (4) phpMyAdmin などを使って、レコードが正しく削除されていること、アップデートされていることを確認してください。

設問7. index.php で、

- (1) 完了しているレコード（行）の期限日と TODO 項目に打消し線を入れます。打ち消し線は、
 - ✓ 親要素に対して
 - ✓ タグなどを使ってスタイルに「text-decoration: line-through」を指定します。
- (2) 完了しているレコードは、ラジオボタンの「完了」にチェックが入り、未完了のレコードは「未完了」にチェックが入ります。
- (3) 削除されたレコードは、テーブル上には存在しませんので、表示されません。

● index.php

「完了」にしたとき

TODOリスト			
期限日	TODO項目		
2020/02/25	TODO項目を入力してください	<input type="button" value="追加"/>	
2020-02-25	テストの項目 1	<input type="radio"/> 未完了 <input checked="" type="radio"/> 完了 <input type="checkbox"/> 削除	<input type="button" value="実行"/>
2020-05-25	テストの項目 3	<input checked="" type="radio"/> 未完了 <input type="radio"/> 完了 <input type="checkbox"/> 削除	<input type="button" value="実行"/>
2020-08-25	テストの項目 2	<input checked="" type="radio"/> 未完了 <input type="radio"/> 完了 <input type="checkbox"/> 削除	<input type="button" value="実行"/>

「削除」したとき

TODOリスト			
期限日	TODO項目		
2020/02/25	TODO項目を入力してください	<input type="button" value="追加"/>	
2020-05-25	テストの項目 3	<input checked="" type="radio"/> 未完了 <input type="radio"/> 完了 <input type="checkbox"/> 削除	<input type="button" value="実行"/>
2020-08-25	テストの項目 2	<input checked="" type="radio"/> 未完了 <input type="radio"/> 完了 <input type="checkbox"/> 削除	<input type="button" value="実行"/>