データベース(DB)を使ったWebページ制作

データベースとは？

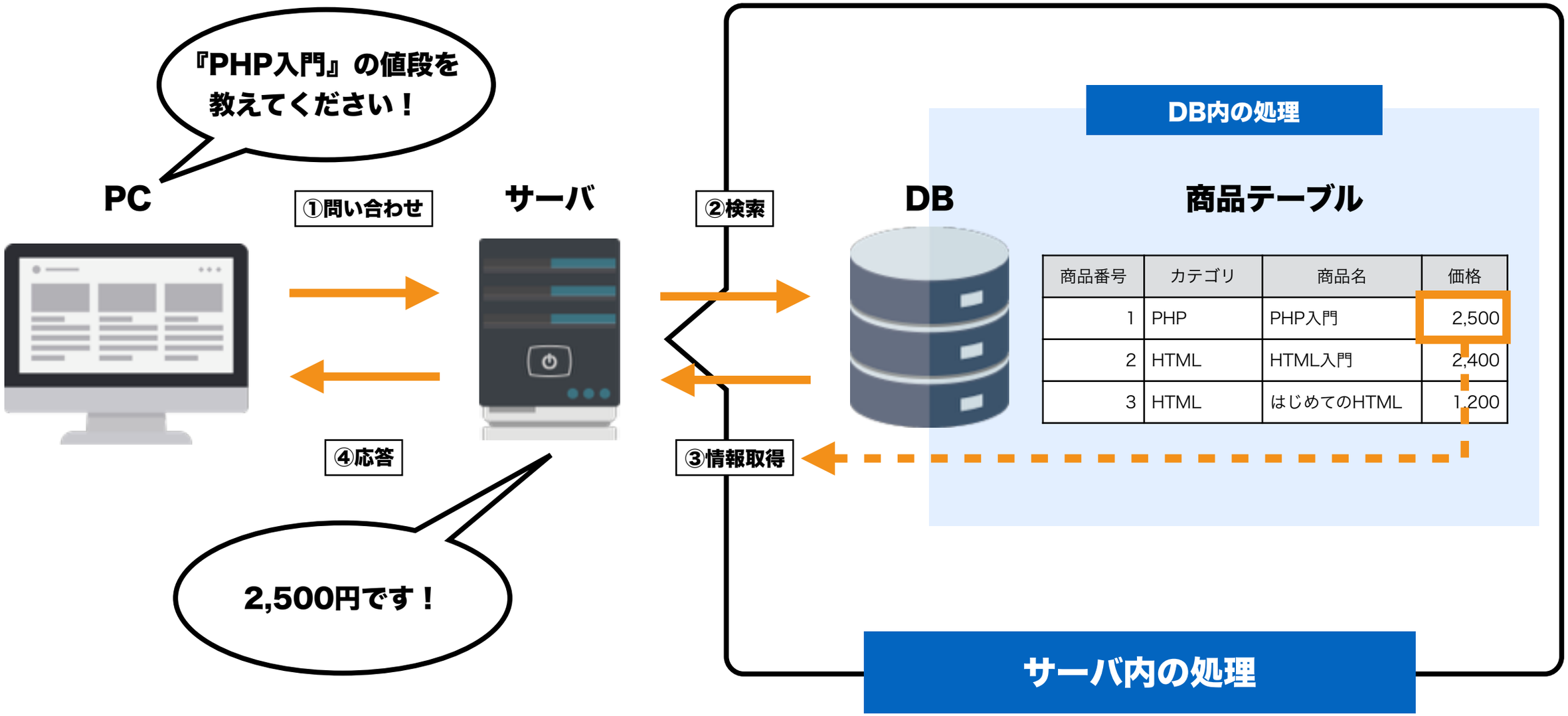
- 大量のデータを集めて、一元的に管理できるようにしたもの
- テーブル（エクセルの表のようなもの）で管理される場合が多い

←T→	m_id データID（主キー／連番）	m_name 名前	m_mail メールアドレス	m_message メッセージ	m_dt 書き込み日時
編集 コピー 削除	1	鈴木一郎	aaa@sangi.jp	おはようございます。	2016-01-01 09:00:00
編集 コピー 削除	2	鈴木二郎	bbb@sangi.jp	いただきます。	2016-02-02 12:00:00
編集 コピー 削除	3	鈴木三郎	ccc@sangi.jp	ごちそうさまでした。	2016-03-03 13:00:00
編集 コピー 削除	4	吉田幸央	yoshida@sangi.ac.jp	お世話になります。	2015-12-16 15:46:19
編集 コピー 削除	5	静岡太郎	shizuokatarou@sangi.jp	いってきます。 いってらっしゃい。	2015-12-16 15:48:11
編集 コピー 削除	6	静岡花子	shizuokahanako@sangi.jp	ただいま。 今日のご飯は？	2015-12-16 17:24:49
編集 コピー 削除	7	産技ユキオ	sangiyukio@sangi.jp	ハンバーグです。 コンソメスープもあるよ。 いや、コーンスープでした。	2015-12-17 16:31:29

様々な情報（Data）を蓄積する基地（Base）

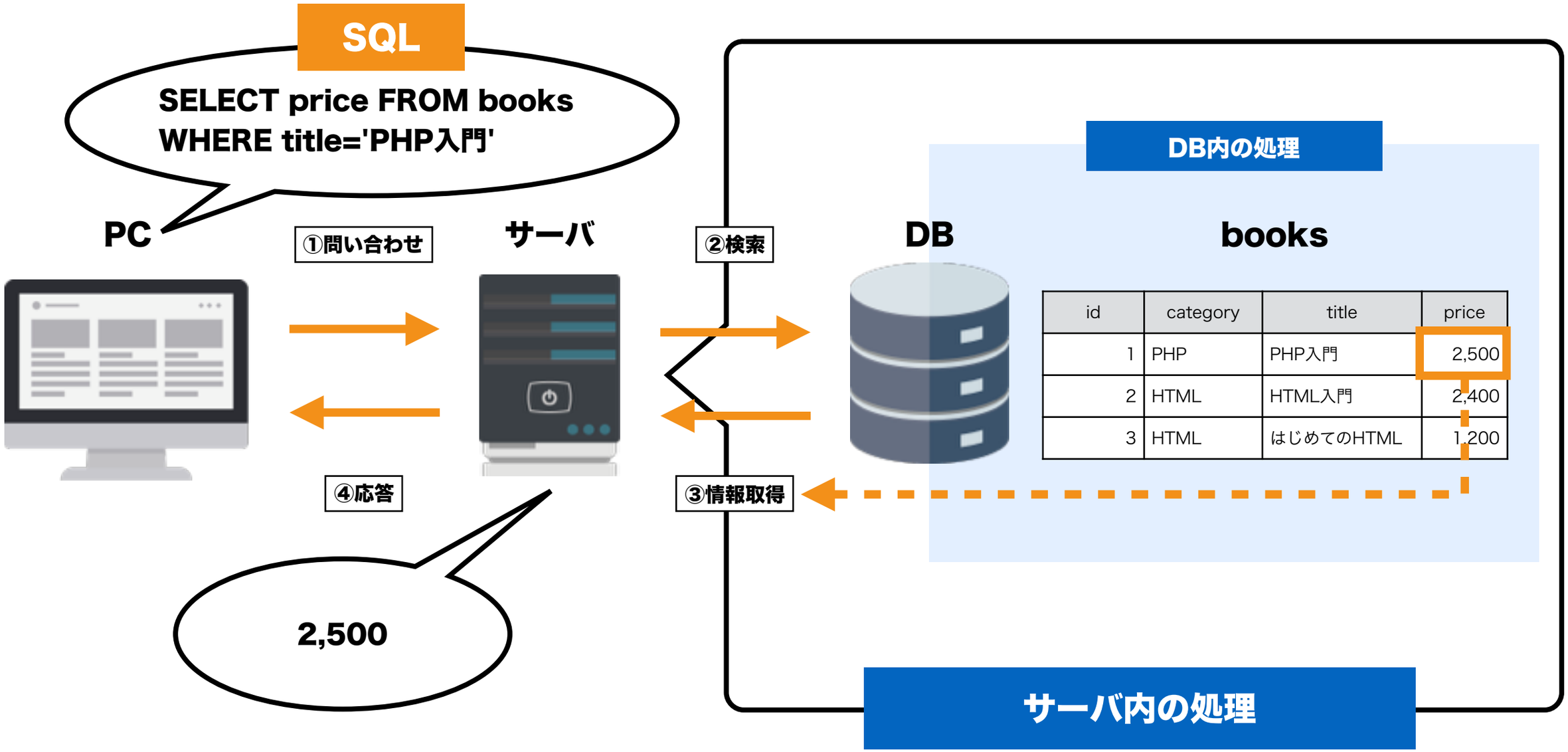
データベースを使う方法

- データベースサーバにお願いをする



データベースを使う方法

- 実際に問い合わせを行う際には、SQLという専用の言語を使う



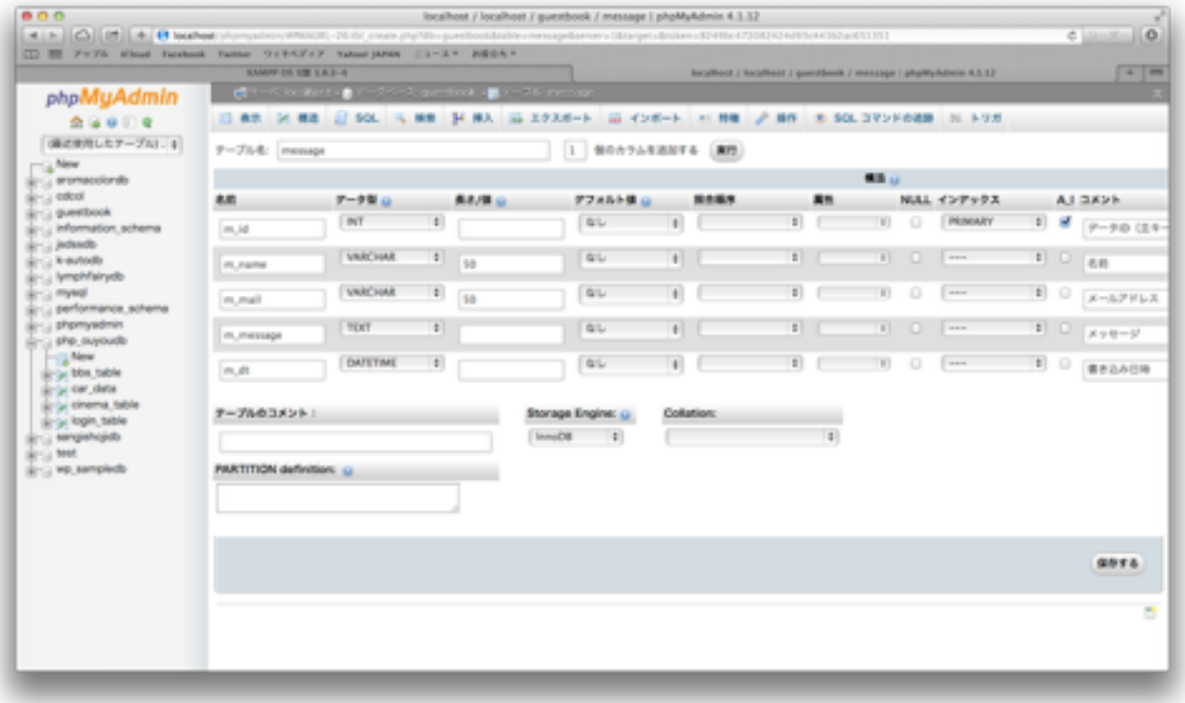
データベースを作る

- **phpMyAdmin**を起動して、データベースタブから作成する
  （XAMPP使用時）

まずはデータベースという「箱」を作る

データベースの中にテーブルを作る

- テーブルは**行列形式**で出来た表である（エクセルの表のようなもの）
- １つのデータベースに複数のテーブルが含まれることもある



「箱」の中の「テーブル」にデータを格納する

主キー（PRIMARY KEY）とは？

- テーブルの行を識別する列（学籍番号・社員番号・商品番号など）
- 空欄は不可、必ず重複しない値を設定する（一意のデータとする）

（例）学生管理

●主キーのないテーブル

名前	郵便番号	住所
鈴木一郎	123-4567	静岡市駿河区1-2-3
鈴木二郎	456-7890	静岡市葵区4-5-6
鈴木三郎	789-0123	静岡市清水区7-8-9
…	…	…
鈴木一郎	012-3456	焼津市1-2-3

どちらの鈴木一郎さんか識別できない

●「学籍番号」という主キーを持つテーブル

学籍番号	名前	郵便番号	住所
E00001	鈴木一郎	123-4567	静岡市駿河区1-2-3
E00002	鈴木二郎	456-7890	静岡市葵区4-5-6
E00003	鈴木三郎	789-0123	静岡市清水区7-8-9
…	…	…	…
E00100	鈴木一郎	012-3456	焼津市1-2-3

学籍番号で識別できる（学籍番号は重複しない）

データ型とは？

- その列に格納する値が**文字列**なのか、**数値**なのか、**日付**なのか等を識別する

（例）ゲストブック（掲示板）

数値 / 文字列 / 文字列 / 長い文字列 / 日付（時間含む）

データ番号	名前	メールアドレス	メッセージ	投稿日時
1	鈴木一郎	aaa@sangi.jp	おはようございます。	2016-01-01 09:00:00
2	鈴木二郎	bbb@sangi.jp	いただきます。	2016-02-02 12:00:00
3	鈴木三郎	ccc@sangi.jp	ごちそうさまでした。	2016-03-03 13:00:00

●データ型の種類

- 数値（整数） … INT
- 文字列 … VARCHAR
- 日付 … DATE
- 数値（小数） … DOUBLE
- 長い文字列 … TEXT
- 日付（時間含む） … DATETIME　他

データベースを操作する

- データベースを操作するには、SQLという専用の言語を使う
- SQLの命令文には、主に以下の4種類がある

SQLの命令文	
SELECT	データを取得する命令
INSERT	データを追加する命令
UPDATE	データを変更する命令
DELETE	データを削除する命令

データベースを操作する

・SELECT文　…　データを取得する命令文

（例）guestbookデータベース

※実際の作業においては、
データベース名、テーブル名、列名は半角英数字で付ける
単語を組み合わせるときは、アンダーバー（ _ ）で繋げる

●messageテーブル

m_id	m_name	m_mail	m_message	m_dt
1	鈴木一郎	aaa@sangi.jp	おはようございます。	2016-01-01 09:00:00
2	鈴木二郎	bbb@sangi.jp	いただきます。	2016-02-02 12:00:00
3	鈴木三郎	ccc@sangi.jp	ごちそうさまでした。	2016-03-03 13:00:00

SELECT　列名　FROM　テーブル名

（例）SELECT * FROM message　…　messageテーブルから全ての列を取得する
SELECT m_name FROM message　…　名前の列を取得する
SELECT m_id , m_name FROM message WHERE m_id=1
…　データ番号と名前の列を取得する、条件としてデータ番号が1のもの
SELECT * FROM message ORDER BY m_id DESC
…　全ての列を取得する、データ番号を降順（大きい順）に並べ替えて　　※ASC（昇順）

データベースを操作する

・INSERT文　…　データを追加する命令文

（例）guestbookデータベース

●messageテーブル

m_id	m_name	m_mail	m_message	m_dt
1	鈴木一郎	aaa@sangi.jp	おはようございます。	2016-01-01 09:00:00
2	鈴木二郎	bbb@sangi.jp	いただきます。	2016-02-02 12:00:00
3	鈴木三郎	ccc@sangi.jp	ごちそうさまでした。	2016-03-03 13:00:00
4	鈴木四郎	ddd@sangi.jp	こんにちは。	2016-04-04 15:00:00

INSERT　INTO　テーブル名（列名）　VALUES（値）

（例）INSERT INTO message(m_name , m_mail , m_message , m_dt)
VALUES('鈴木四郎' , 'ddd@sangi.jp' , 'こんにちは。' , '2016-04-04 15:00:00')
…　VALUESの値を、messageテーブルに追加する

※m_idは自動採番される（テーブル作成時に、AUTO_INCREMENTを設定する）

データベースを操作する

- **UPDATE文　…　データを変更する命令文**

（例）guestbookデータベース

●messageテーブル

m_id	m_name	m_mail	m_message	m_dt
1	静岡太郎	zzz@sangi.jp	おはようございます。	2016-01-01 09:00:00
2	鈴木二郎	bbb@sangi.jp	いただきます。	2016-02-02 12:00:00
3	鈴木三郎	ccc@sangi.jp	ごちそうさまでした。	2016-03-03 13:00:00

UPDATE　テーブル名　SET　列名 = 値　WHERE　条件

（例）UPDATE message
SET m_name='静岡太郎' , m_mail='zzz@sangi.jp' WHERE m_id=1
…　messageテーブルのデータ番号1の名前とメールアドレスを
セットされた値に変更する

※主キーの値は変更しないように注意すること

データベースを操作する

・DELETE文　…　データを削除する命令文

（例）guestbookデータベース

●messageテーブル

m_id	m_name	m_mail	m_message	m_dt
2	鈴木二郎	bbb@sangi.jp	いただきます。	2016-02-02 12:00:00
3	鈴木三郎	ccc@sangi.jp	ごちそうさまでした。	2016-03-03 13:00:00

DELETE　FROM　テーブル名　WHERE　条件

（例）DELETE FROM message WHERE m_id=1

…　messageテーブルのデータ番号1のデータ（行）を削除する

※間違って削除したデータは、基本的には元に戻せないので注意すること

ゲストブック（掲示板）を作る

・ゲストブックの構成

データ追加機能	メッセージの投稿
データ変更機能	メッセージの変更
データ削除機能	メッセージの削除
データ表示機能	メッセージの一覧表示・個別メッセージの表示

・guestbookデータベースのmessageテーブル

名前	データ型	長さ／値	NULL	インデックス	A_I	コメント
m_id	INT		□	PRIMARY	☑	データID（主キー／連番）
m_name	VARCHAR	50	□		□	名前
m_mail	VARCHAR	50	□		□	メールアドレス
m_message	TEXT		□		□	メッセージ
m_dt	DATETIME		□		□	書き込み日時

※サーバ接続・データベースの照合順序は、utf8_unicode_ciとする
※カラム数は、項目数を示す（上記はカラム数「5」）

ゲストブック（掲示板）を作る

• ゲストブックの処理イメージ

※装飾は別途CSSを用いる

トップページ（index.php）

名前	
メールアドレス	
メッセージ	
確認する	

7：産技ユキオ（2015/12/17 16:31）

ハンバーグです。
コンソメスープもあるよ。
いや、コーンスープでした。

変更 削除 詳細

6：静岡花子（2015/12/16 17:24）

ただいま。
今日のご飯は？

変更 削除 詳細

追加確認画面（confirm.php）

名前	産技六郎
メールアドレス	sangirokurou@sangi.jp
メッセージ	パンとライス、どちらか選べますか？
書き込む	

追加完了画面（submit.php）

データを追加しました。データ番号：8

名前	産技六郎
メールアドレス	sangirokurou@sangi.jp
メッセージ	パンとライス、どちらか選べますか？

トップページへ

変更画面（update.php）

名前	産技五郎
メールアドレス	sangigorou@sangi.jp
メッセージ	ハンバーグとコーンスープに、 サラダも付くよ。
確認する	

変更確認画面（update-confirm.php）

名前	産技五郎
メールアドレス	sangigorou@sangi.jp
メッセージ	ハンバーグとコーンスープに、 サラダも付くよ。
変更する	

変更完了画面（update-submit.php）

データを変更しました。

名前	産技五郎
メールアドレス	sangigorou@sangi.jp
メッセージ	ハンバーグとコーンスープに、 サラダも付くよ。

トップページへ

詳細表示画面（detail.php）

名前	産技ユキオ
メールアドレス	sangiyukio@sangi.jp
メッセージ	ハンバーグです。 コンソメスープもあるよ。 いや、コーンスープでした。

トップページへ

削除確認画面（delete-confirm.php）

名前	産技六郎
メールアドレス	sangirokurou@sangi.jp
メッセージ	パンとライス、どちらか選べますか？
削除する	

削除完了画面（delete-submit.php）

データを削除しました。

トップページへ

ゲストブック（掲示板）を作る

- データ追加機能

STEP1
データベースへの接続

↓

STEP2
SQLの実行（データの追加）

↓

STEP3
実行結果の処理

ユーザ認証

操作を許可される

INSERT文の実行

実行結果の取得
（正常に終了したか）

エラーチェックなど

サーバ

messageテーブル

m_id	m_name
1	鈴木一郎
2	鈴木二郎
3	鈴木三郎

エラーチェックなどを実行
（必須ではない）

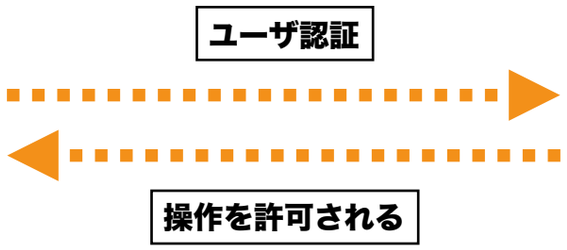
ゲストブック（掲示板）を作る

- データベースへの接続方法　…　PDO（PHP Data Object）

データベース　：MySQL　　ユーザー名：root
データベース名：guestbook　　パスワード：root
ホスト名　　：localhost

※左記のユーザ名、パスワードは
MAMPの初期設定のもの

STEP1
データベースへの接続



【記述例】

```
$conn=new PDO(
    "mysql:dbname=guestbook;host=localhost;charset=utf8" , "root" , "root"
);
```

データベース　使用するデータベース名　使用するホスト名　文字コード設定　MySQLのユーザ名・パスワード

ゲストブック（掲示板）を作る

・index.php

データの取得

```
$sql="SELECT * FROM message ORDER BY m_id DESC";
$stmt=$conn->prepare($sql);
$stmt->execute();
```

決まり文句と考えて良い

※prepare = 準備
※execute = 実行

$sql = messageテーブルから全ての列を取得する、データ番号を降順に並べ替えて
$stmt（ステートメント：宣言）= $sqlを実行するための準備
準備した$stmtを実行

※$connを通してPDOの命令を呼び出すときは、
「$conn->prepare($sql)」のように「->」という矢印でつながれる

ゲストブック（掲示板）を作る

- index.php

取得したデータを一覧表示

while($row=$stmt->fetch()){

※fetch = 取得

まだ次のデータはありますか？

「いいえ：false」の場合
（次のデータはない）

表示を終了

「はい：true」の場合
（次のデータがある）

表示を続ける

messageテーブル

m_id	m_name
1	鈴木一郎
2	鈴木二郎
3	鈴木三郎

nl2br($row["m_message"])

改行文字の前にHTMLの改行タグを挿入する（改行を有効にするための関数）

ゲストブック（掲示板）を作る

・index.php

変更・削除・詳細表示画面へのリンク

```
echo "<a href='update.php?m_id=".$row["m_id"]."'>変更</a> ";
```

「?」に続いて「変数（パラメータ）=値」を記述

update.phpへ移動する、ただし条件としてデータ番号が「$row["m_id"]」の記事のもの

【実際のURL】

●データ番号1の記事の変更画面　　http://localhost:8888/guestbook/update.php?m_id=1
●データ番号2の記事の変更画面　　http://localhost:8888/guestbook/update.php?m_id=2
●データ番号3の記事の変更画面　　http://localhost:8888/guestbook/update.php?m_id=3

ゲストブック（掲示板）を作る

・confirm.php

セッションの開始　／　入力値をセッション変数に格納

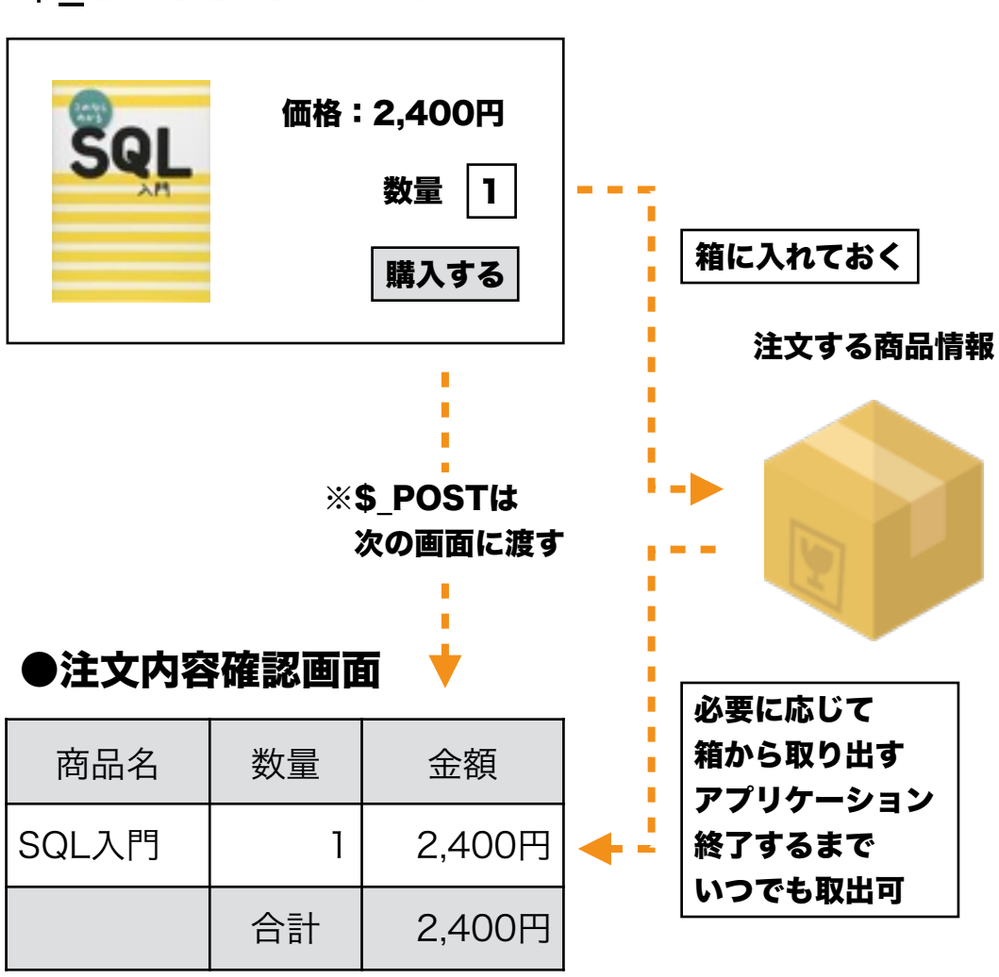
session_start();

セッションデータを格納する$_SESSIONを使用するために記述

```
$_SESSION["m_name"]=$m_name;
$_SESSION["m_mail"]=$m_mail;
$_SESSION["m_message"]=$m_message;
```

必要に応じていつでも取り出せるようにする

$_SESSIONのイメージ



ゲストブック（掲示板）を作る

・confirm.php

入力値の検証・加工（入力値の取得・検証・加工とリンクしている）

```
function chkString($temp="",$field,$accept_empty=false){
        //未入力チェック
        if(empty($temp) AND !$accept_empty){
            echo "{$field}には何か入力してください。";
            exit;
        }
```

【ユーザ定義関数】
自分で独自に定義した関数

```
function 自分で付けた関数名（引数リスト）{
    処理内容
}
```

$tempが空欄かどうか
field（項目）の名前
3番目のパラメータがfalseかどうか
（trueならチェックを省略する）

このユーザ定義関数をもとに
入力値の取得・検証・加工の中で
入力チェックを行っている

※temp = テンポラリ（一時ファイル）

ゲストブック（掲示板）を作る

・confirm.php

入力値を安全な値に

```
$temp=htmlspecialchars($temp,ENT_QUOTES,"utf-8");
```

- htmlspecialchars：悪意のあるスクリプト等を無効にして安全な文字列に置き換える
- $temp：文字列
- ENT_QUOTES：ENT_QUOTESのときは「"」も「'」も加工する
- "utf-8"：文字コード

【加工前の文字列】
<script>alert('いたずらだよ！');</script>　　スクリプトは実行されてしまう（危険）
【加工後の文字列】
<script>alert('いたずらだよ！');</script>　　実行されない（安全）

戻り値

```
return $temp;
```

← 処理した後（加工後）の値を返す

ゲストブック（掲示板）を作る

- submit.php

セッションの開始

```
session_start();
if(empty($_SESSION)){
        exit;
}
```

セッションの開始

直接アクセス（URL直打ち等）
されるのを防ぐため
$_SESSIONが空なら強制終了

入力内容の取得（$_SESSIONから）

```
$m_name=htmlspecialchars($_SESSION["m_name"],ENT_QUOTES,"utf-8");
$m_mail=htmlspecialchars($_SESSION["m_mail"],ENT_QUOTES,"utf-8");
$m_message=htmlspecialchars($_SESSION["m_message"],ENT_QUOTES,"utf-8");
```

安全な値に変換した上で箱（$_SESSION）から取出す

ゲストブック（掲示板）を作る

・submit.php

データの追加

```
$sql="INSERT INTO message(m_name,m_mail,m_message,m_dt)
        VALUES(:m_name,:m_mail,:m_message,NOW())";
```

「:」を付けると目印（パラメータ）になる

●入力された実際の値

'鈴木一郎' , 'aaa@sangi.jp' , 'おはようございます。'

●目印を実際の値に置き換えた後のINSERT文

```
$sql="INSERT INTO message(m_name,m_mail,m_message,m_dt)
        VALUES('鈴木一郎' , 'aaa@sangi.jp' , 'おはようございます。' ,NOW())";
```

【bindParamメソッド】
「目印（パラメータ）」を「変数」に置き換える命令

直に$_POSTや$_SESSIONを指定して受け取るのではなく、
「目印（パラメータ）」と「bindParam」メソッドを使用することで、セキュリティの向上につながる

ゲストブック（掲示板）を作る

- **submit.php**

エラーチェック

```
$error=$stmt->errorInfo();
if($error[0]!="00000"){
    $message="データの追加に失敗しました。{$error[2]}";
}else{
    $message="データを追加しました。データ番号：".$conn->lastInsertId();
}
```

エラーコードが「00000」ではなかったら追加失敗、エラーメッセージを出力する
エラーメッセージは、データベースハンドラによって実行された直近の操作に関するエラー情報を配列として返す
エラーメッセージは、errorInfo配列の[2]の要素に格納されている

●errorInfo配列

要素	情報
0	SQLSTATE エラーコード
1	ドライバ固有のエラーコード
2	ドライバ固有のエラーメッセージ

セッションデータの破棄

```
$_SESSION=array();
session_destroy();
```

$_SESSIONを全て解除する
$_SESSIONを全て破壊する

ゲストブック（掲示板）を作る

- データ表示機能

STEP1
データベースへの接続

↓

STEP2
SQLの実行（データの表示）

↓

STEP3
実行結果の処理

ユーザ認証 ▶

◀ 操作を許可される

SELECT文の実行 ▶

◀ 実行結果の取得

データを一覧表示 ▶

サーバ

messageテーブル

m_id	m_name
1	鈴木一郎
2	鈴木二郎
3	鈴木三郎

実行結果（ブラウザ表示）

1：鈴木一郎
2：鈴木二郎
3：鈴木三郎

ゲストブック（掲示板）を作る

- detail.php

表示するデータの主キーを取得

```
if(!isset($_GET["m_id"])){
        exit;
}else{
        $m_id=$_GET["m_id"];
}
```

$_GET

データがURLで受け渡される

1：鈴木一郎（2016/01/01 09:00）

おはようございます。

変更 削除 詳細　**データ番号1の詳細リンクをクリック**

detail.php?m_id=1

データがURLで受け渡される

※$_POSTはURLで受け渡されない
※$_POSTはフォームに入力された値で受け渡される

ゲストブック（掲示板）を作る

・detail.php

データの取得（1件のみ）

```
$sql="SELECT * FROM message WHERE(m_id=:m_id);";
$stmt=$conn->prepare($sql);
$stmt->bindParam(":m_id",$m_id);
$stmt->execute();
$row=$stmt->fetch();
```

messageテーブルから該当するデータ番号の列を全て取得する命令文
SQLの命令を準備して
目印（パラメータ）を変数に置き換えて
SQLの命令を実行して
実行したデータを取得する（while文ではないので1件のみ）

ゲストブック（掲示板）を作る

・データ変更機能

データを変更するには

まずは変更対象となるデータを1件取得するために
SELECT文が必要、その後にUPDATE文を実行

1：鈴木一郎（2016/01/01 09:00）

おはようございます。

変更 削除 詳細

●変更リンクのタグ

update.php?m_id=1

●実行されるSQL

SELECT * FROM message WHERE(m_id=1)

変更画面

名前	鈴木一郎
メールアドレス	aaa@sangi.jp
メッセージ	おはようございます。
確認する	

フォーム上に表示

●messageテーブル

m_id	m_name	m_mail	m_message
1	鈴木一郎	aaa@sangi.jp	おはようございます。

messageテーブルから取得

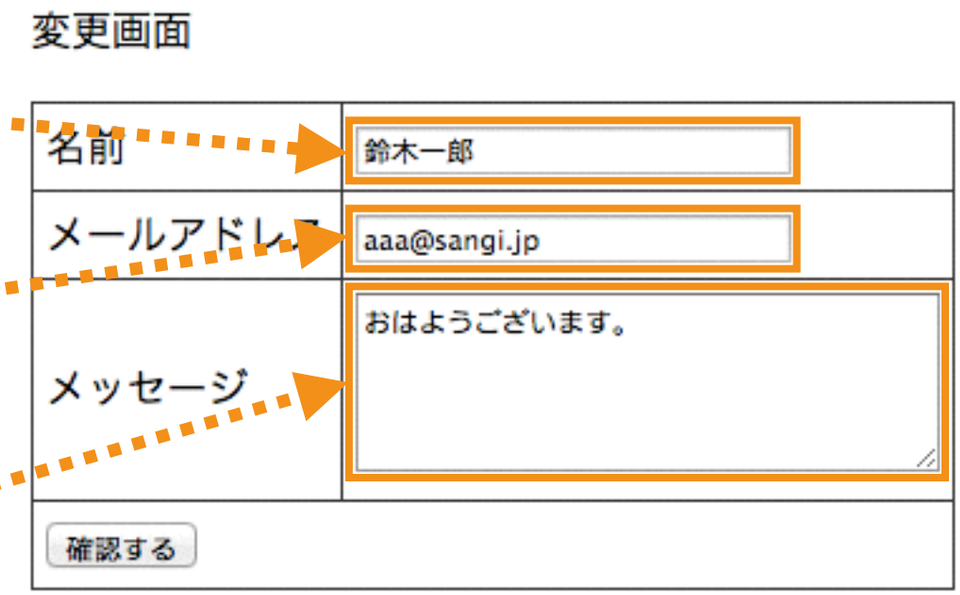
ゲストブック（掲示板）を作る

- update.php

データ変更フォーム

各項目の値を変更フォーム内に表示するには、textはvalue属性、textareaは要素内に
$row["○○○"]のように列名を指定する、text以外のinput typeもvalue属性で表示可

```
<tr>
  <td>名前</td>
  <td><input type="text" name="m_name" size="30"
        value="<?php echo $row["m_name"]; ?>"></td>
</tr>
<tr>
  <td>メールアドレス</td>
  <td><input type="text" name="m_mail" size="30"
        value="<?php echo $row["m_mail"]; ?>"></td>
</tr>
<tr>
  <td>メッセージ</td>
  <td><textarea rows="5" cols="30" name="m_message">
        <?php echo $row["m_message"]; ?></textarea></td>
</tr>
```



ゲストブック（掲示板）を作る

- update-submit.php

変更内容を取得（変更データの主キーも含む）

```
$m_id=$_SESSION["m_id"];
$m_name=htmlspecialchars($_SESSION["m_name"],ENT_QUOTES,"utf-8");
$m_mail=htmlspecialchars($_SESSION["m_mail"],ENT_QUOTES,"utf-8");
$m_message=htmlspecialchars($_SESSION["m_message"],ENT_QUOTES,"utf-8");
```

データ追加処理とは異なり、変更対象となるデータを識別するため、そのデータの主キーも取得する必要あり
また主キー以外の値は安全な値に変換しておく（htmlspecialchars関数）

ゲストブック（掲示板）を作る

- update-submit.php

データを変更

```
$sql="UPDATE message SET
        m_name=:m_name,m_mail=:m_mail,m_message=:m_message,m_dt=NOW()
        WHERE m_id=:m_id";
```

messageテーブルの該当するデータ番号の列（WHERE句で条件指定）を変更する
変更する列はSET句にセットされた列（m_name,m_mail,m_message,m_dt）
変更する内容はその値（:m_name,:m_mail,:m_message,:m_dt）

「:」が付いた目印（パラメータ）は、この後bindParamメソッドで変数に置き換えられる

ゲストブック（掲示板）を作る

・データ削除機能

データを削除するには

まずは削除対象となるデータを１件取得するために
SELECT文が必要、その後にDELETE文を実行

1：鈴木一郎（2016/01/01 09:00）

おはようございます。

変更 削除 詳細

●削除リンクのタグ

delete-confirm.php?m_id=1

●実行されるSQL

SELECT * FROM message WHERE(m_id=1)

削除確認画面

名前	鈴木一郎
メールアドレス	aaa@sangi.jp
メッセージ	おはようございます。
削除する	

フォーム上に表示

●messageテーブル

m_id	m_name	m_mail	m_message
1	鈴木一郎	aaa@sangi.jp	おはようございます。

messageテーブルから取得

ゲストブック（掲示板）を作る

- delete-submit.php

削除データの主キーを取得

```
$m_id=$_SESSION["m_id"];
```

データ削除に必要な、削除対象データの主キー（m_id）を$_SESSIONから取得する

データを削除

```
$sql="DELETE FROM message WHERE(m_id=:m_id);";
```

messageテーブルの該当するデータ番号の列（WHERE句で条件指定）を削除する

memo