第１０回　データベース接続（INSERT）

10．新規登録

１０－１．トランザクション処理

トランザクション管理については、データベースの授業で既に学んでいますので、ここではPHPでのコードの記述方法を記載します。

■トランザクション管理メソッド

|  |  |
| --- | --- |
| PDOのメソッド | 意味 |
| PDO::beginTransaction | トランザクションを開始する |
| PDO::commit | トランザクションを完了する |
| PDO::rollback | トランザクション開始前の状態に戻す |

トランザクションの開始は、PDO::beginTransactionメソッドで行います。トランザクションは、PDO::commitメソッドもしくはPDO::rollbackメソッドが呼ばれるまで継続します。

　一連のデータベース処理を完了したいときは、PDO::commitメソッドでトランザクションの確定を行い、処理を取り消したいときは、PDO::rollbackでトランザクションのロールバックを行ってください。

|  |  |
| --- | --- |
| 警告 単色塗りつぶし | コミットもロールバックも行わなかったときは  トランザクションを開始したら、コミットするまではデータベース処理が確定されません。コミットもロールバックも行わなかった状態で、プログラムを終了すると、「ロールバックと同じ意味になります。　 また、トランザクションの状態を知るには、PDO::inTransactionメソッドで確認することができます。（トランザクションがアクティブであればTrue、アクティブでなければFalseが返ってきます。 |

＊自動コミットモードの指定方法は、「8回目の資料」で確認してください。

★Sampleコード（INSERT）

Sampleのデータベース情報**（皆さんが使用するDBとは違うものを使用しています）**

・ホスト名　　　　　：　localhost

・DB名　　　　　　：　malldb

・DBユーザー　　：　malluser

・DBパスワード　：　mall

・テーブル 　：products

　【概要】

ひとつ前のページからPOST形式で「product\_id」と「product\_name」が送信された状態で、

POSTデータを受け取り、処理をする一例です。

コードが長くなるため、次のページに記載します。

$dsn = "mysql:host=".DB\_HOST.";dbname=malldb;charset=".DB\_CHARSET;

  try {

    $db = new PDO($dsn, "malluser", "mall");

    $db->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

    $db->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

    $db->setAttribute(PDO::ATTR\_AUTOCOMMIT, false);

    // トランザクション開始

    $db->beginTransaction();

    $sql = "INSERT INTO PRODUCTS(PRODUCT\_ID, PRODUCT\_NAME) VALUES(:product\_id, :product\_name)";

    $stmt = $db->prepare($sql);

    $stmt->bindParam(':product\_id', $prduct\_id, PDO::PARAM\_INT);

    $stmt->bindParam(':product\_name', $product\_name, PDO::PARAM\_STR);

    $stmt->execute();

    // トランザクション確定

    $db->commit();

  }catch(PDOException $poe) {

    $db->rollBack();

　　　// TODO:debug用メッセージ（本番ではセキュリティ上表示しないこと！！）

    echo "DB接続エラー".$poe->getMessage();

  }finally{

    $stmt = null;

    $db = null;

  }

【注意事項】

|  |  |
| --- | --- |
| 警告 単色塗りつぶし | 例外処理 catch文に記載している処理について  「echo "DB接続エラー".$poe->getMessage();」  開発中にエラー内容が分かり易いように、**デバッグ用として出力**しています。  （本授業では以降の課題もデバッグ用コードを使用しますが、）  PDOのエラーメッセージを出力してしまうと、エラーメッセージから読み取れる情報（テーブルやカラム名など）が多いため、実稼働においてはセキュリティ上よろしくありません。**本番環境では画面出力はせず、ユーザーがアクセスできないサーバーのログファイルに書き込むなどの対策をとりましょう。** |