

Webシステムプログラミングb 第13講目

Webシステムプログラミング講義資料



■ 到達目標

Webシステムを設計し、開発できること 最終成果物:学生生活で利用できるWebシステム

▪ 授業計画

■ 第12講(01/07)

■ 第13講(01/21)

- 第09講(12/03):Webシステムの基本設計(1)
- 第10講(12/10):Webシステムの基本設計(2)
- 第11講(12/17):WebシステムのDB設計
 - :WebシステムのUI設計
 - :Webシステムの処理
- 第14講(01/23):補講日、サンプル課題設計書
- 第15講(01/28):平常試験(成果物確認)、まとめ



■ DB設計はシステム設計の一部



Tokyo University of Information Sciences







Tokyo University of Information Sciences



■ 成果物一覧

- 第10回(12/03):要件定義書
- 第11回(12/10):基本設計書(機能、データ構造、画面)
- 第12回(12/17): ER図、テーブル定義表
- 第13回(01/07):画面遷移図

■ 第15回(01/28):成果物確認

- ■開発したWebシステムの動作確認
 - Webの入出力インタフェースとDBの対応を確認
 - 各自のPC上で教員・TA・SAが動作確認を行い評価

評価基準の目安

■目安

- C: サンプル課題の流用 or テーブル内容の表示のみ
 - サンプル課題をそのままの流用
 - レコードの追加、削除はSQL手動での実行のみ
- B:オリジナル課題、テーブルの追加・表示ができること
 - レコードの削除、変更はSQL手動で実行でもOK
- A~S:テーブルの追加・表示・変更・削除ができること
 - レコードの入出力は、基本Web上で実行可能
 - 発展部分の実装

※平常試験やレポート課題を含め総合的に評価する

■ サンプル課題





今日やること

1. これまでの設計書のレビュー

- ■教員・TA・SAと相談し、1週間で実現可能な設計に修正
- オリジナル課題で実現不可能な場合は、サンプル課題に変更
- 2. システム開発
 - 教員・TA・SAと相談し、システム開発を進める
 - テーブルのレコード一覧表示は最優先で進める

- テーブル結合を行った上で一覧を表示する

レコードの追加・変更・削除もできるようにする

3. 来週の動作確認に備える

サンプル課題の設計書は次ページ以降を参照
 サンプル課題のプログラムも合わせて提示予定

サンプル課題

- タイトル: ゼミ室の蔵書管理システム
- 業務分析
 - ゼミ室には教員の専門分野に関する書籍が豊富にある
 - 研究を進める際に、ゼミ生はゼミ室で書籍を読んだり、
 必要に応じて教員から許可を得て書籍を借りたりする
 - 教員は誰に何の書籍を貸しているかは管理できていない
- 要件定義
 - ゼミ室の蔵書と貸し出し記録を管理したい
 - 教員はゼミ生の名簿と蔵書を管理したい
 - 教員、またはゼミ生は貸し出し記録を付けた
 - ■書籍の貸し出し冊数は5冊まで、期限は2週間とする



機能

- ログイン機能
 - ユーザのID、パスワードでログイン
- ユーザ管理機能
 - ユーザの追加・更新・削除が可能
 - ユーザ毎に権限を付与し、利用できる機能を制限
- 蔵書登録機能
 - 書籍の追加・更新・削除が可能
 - 書籍のタイトル、カテゴリ、出版社、発行年などを管理
 - 条件を指定した蔵書検索が可能
- 貸し出し機能
 - ユーザへの書籍貸し出しを台帳に記録
 - ユーザへの貸し出し冊数の上限と期限を設ける



- データ構造
 - ユーザ情報
 - ユーザID、パスワード、名前、権限(管理者 or 利用者)
 - ユーザ情報を変更できるのは管理者のみ
 - 蔵書情報
 - 書籍のタイトル、カテゴリ、出版社、発行年
 - 貸出台帳
 - 書籍と貸し出したユーザの紐付け
 - 貸出日、返却予定日、貸出状態(貸出OK or 貸出中)

画面設計(1)

画面設計図:蔵書リスト画面



書籍の新規登録

※図作成には、リアルタイムコラボレーションツール「<u>Cacoo</u>」を利用

Tokyo University of Information Sciences

画面設計(2)

• 画面設計図: 蔵書 新規登録画面

- すべての項目を入力し、「登録」ボタンで登録可
 - タイトル、出版社:テキストボックス
 - カテゴリ、発行年: 選択ボックス

| 蔵書 新規 | 登録画面 | |
|-------|------|---|
| タイトル | | |
| カテゴリ | 理工系 | • |
| 出版社 | | |
| 発行年 | 2001 | |
| 登録 | | |

Tokyo University of Information Sciences

画面設計(3)

■ 画面設計図: 蔵書貸出画面

- 「貸出可」より蔵書の貸出が可能
- 「貸出書籍」より自分の書籍貸出状況を確認可

| | | | 1 | 貸出操作が可能 |
|------------------|------|------------|------|---------|
| ゼミ室 蔵書貸出シス | テム | | | |
| タイトル | カテゴリ | 返却予定日 | 貸出状況 | |
| Webエンジニアの教科書 | 理工系 | 2015/12/14 | 貸出中 | |
| たのしいRuby | 理工系 | - | 貸出可 | |
| 嫌われる勇気 | 心理学 | 2015/12/3 | 貸出中 | |
| PHP+MySQLマスターブック | 理工系 | - | 貸出可 | |

皆出書籍



- 以下の5テーブルで構成される
 - users:利用者テーブル, books:蔵書テーブル
 - categories:カテゴリテーブル, publishers:出版社テーブル
 - circulations:貸出テーブル



Tokyo University of Information Sciences

テーブル定義表(1)

表.利用者テーブル:users

| フィールド名 | 型 | Not null | Key | Auto increment | Default | 意味 |
|--------|-----------------|-------------|-----|-------------------|---------|-------------------|
| id | INT(11) | YES | 主キー | YES | NULL | 利用者ID |
| name | VARCHAR (40) | | | | NULL | 名前 |
| role | INT(1) | | | | NULL | 権限 (0:学生、1:教員) |



表. 蔵書テーブル: books

| フィールド名 | 型 | Not null | Key | Auto increment | Default | 意味 |
|--------------|-----------------|-------------|----------|-------------------|---------|--------|
| id | INT(11) | YES | 主キー | YES | NULL | 蔵書ID |
| title | VARCHAR (40) | | | | NULL | 書籍名 |
| category_id | INT(11) | YES | 外部 キー | | NULL | カテゴリID |
| publisher_id | INT(11) | YES | 外部 キー | | NULL | 出版社ID |
| year | INT(4) | | | | NULL | 発行年 |

テーブル定義表(3)

表. カテゴリテーブル: categories

| フィールド名 | 型 | Not null | Кеу | Auto increment | Default | 意味 |
|--------|-----------------|-------------|-----|-------------------|---------|--------|
| id | INT(11) | YES | 主キー | YES | NULL | カテゴリID |
| name | VARCHAR (20) | | | | NULL | カテゴリ名 |



表. 出版社テーブル: publishers

| フィールド名 | 型 | Not null | Кеу | Auto increment | Default | 意味 |
|--------|-----------------|-------------|-----|-------------------|---------|-------|
| id | INT(11) | YES | 主キー | YES | NULL | 出版社ID |
| name | VARCHAR (40) | | | | NULL | 出版社名 |

テーブル定義表(5)

表. 貸出テーブル: circulations

| フィールド名 | 型 | Not null | Кеу | Auto increment | Default | 意味 |
|-------------|---------|-------------|----------|-------------------|---------|----------------------------|
| id | INT(11) | YES | 主キー | YES | NULL | 貸出ID |
| user_id | INT(11) | YES | 外部 キー | | NULL | 利用者ID |
| book_id | INT(11) | YES | 外部 キー | | NULL | 蔵書ID |
| issue_date | DATE | | | | NULL | 貸出日 |
| return_date | DATE | | | | NULL | 返却予定日 |
| status | INT(1) | | | | NULL | 貸出状態 (0:貸出OK、 1:貸出中) |



ゼミ室蔵書管理システムの画面遷移図



Tokyo University of Information Sciences

サンプル課題の開発について



- フォーム、DB登録処理:PHP
- DB : MySQL
- 実行環境:XAMPP or 各研究室のLinuxサーバ

■進め方

- 最低限できて欲しい機能:蔵書一覧の表示、蔵書登録
- できるとよい機能:書籍の貸出機能
- できなくてもよい機能:
 - ログイン機能:勉強にはなるが、セッション管理が必要
 - ユーザ管理機能:ユーザはDBから直接追加でもよい

サンプルプログラムのソースコード

- サンプル課題の動作確認(学内のみアクセス可)
 - URL : <u>http://202.26.158.109/~ykawano/book_list.php</u>
 - 書籍リストの一覧表示、個別表示、登録、削除が利用可能
 - 貸出機能は未実装

• ソースコード

- book_list.php:書籍の一覧表示
- book_for.php:新規登録フォーム
- book_regist.php:DB登録処理
- book_show.php:書籍の個別表示
- book_del.php:書籍の削除
- zemi_books.sql:DBのSQLファイル





■ 蔵書のタイトル、カテゴリ、出版社、発行年を表示

カテゴリ、出版社は別テーブルを結合して表示



ゼミ室 蔵書管理システム

| タイトル | カテゴリ | 出版社 | 発行年 | 操作 |
|------------------|------|---------------|------|------|
| Webエンジニアの教科書 | 理工系 | シーアンドアール研究所 | 2015 | 表示削除 |
| たのしいRuby | 理工系 | ソフトバンククリエイティブ | 2010 | 表示削除 |
| 嫌われる勇気 | 心理学 | ダイヤモンド社 | 2013 | 表示削除 |
| PHP+MySQLマスターブック | 理工系 | マイナビ | 2014 | 表示削除 |
| 量子力学で生命の謎を解く | 物理学 | ソフトバンククリエイティブ | 2015 | 表示削除 |

書籍登録

URL: http://202.26.158.109/~ykawano/book_list.php











■ 書籍のタイトル、カテゴリ、出版社、発行年を入力

| ☐ 蔵書 新規登録画面 × |
|----------------------------|
| ← → C ☆ 202.26.158.1☆ » = |
| 蔵書 新規登録画面 |
| タイトル: Webシステムの開発技術と活用方法 |
| カテゴリー: 理工系 ▼ |
| 出版社: 共立出版 🔹 |
| 発行年: 2013 |
| 登録 |

URL: <u>http://202.26.158.109/~ykawano/book_form.php</u>

Tokyo University of Information Sciences

書籍の新規登録プログラム

登録画面をフォームで作成

- カテゴリテーブルの内容を選択肢に表示
- 出版社も同様に他テーブルから取得して表示する

```
<select name="category_id">
<?php
 try {
    $db = new PDO('mysql:host=localhost;????;charset=utf8',????;??);
    $db->setAttribute(PDO::ATTR ERRMODE, PDO::ERRMODE EXCEPTION);
    $sql = $db->prepare("select * from categories");
    $sql->execute();
                                                 カテゴリテーブルから取得
    while (srow = sql->fetch()) {
        echo("<option selected value=".$row['id'].">".$row['name']."</option>");
 } catch( PDOException $e ){ // データベース処理の例外キャッチ
        exit('データベース処理失敗:'.$e->getMessage());
?>
</select>
```

Tokyo University of Information Sciences



フォームからのデータ受け取り POSTで受け取ったデータに基づく登録処理

// フォームデータの取得 \$title = \$_POST["title"]; フォームからPOSTでデータ取得 \$category_id = \$_POST["category_id"]; \$publisher_id = \$_POST["publisher_id"]; \$year = \$_POST["year"]; // データベースへの接続 \$db = new PDO('mysql:host=localhost;dbname=????;charset=utf8',????,???); \$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); // SQL文の準備と実行 \$sql = \$db->prepare("insert INTO books(title,category_id,publisher_id,year) Value(?,?,?,?)"); \$sql->bindValue(1, \$title); \$sql->bindValue(2, \$category_id); \$sql->bindValue(3, \$publisher_id); \$sql->bindValue(4, \$year); \$sql->execute();

URL: <u>http://202.26.158.109/~ykawano/book_regist.php</u>



■ 書籍リストから書籍情報を個別に表示

| □ 書籍登録 | × |
|---|---------------------|
| ← → C fi | 🗅 ɔw.php?id=2 숬 » 🔳 |
| 書籍登録 | |
| 登録されている | 書籍情報を表示します。 |
| タイトル: たのしし カテゴリ: 理工系 出版社: ソフトバ 年: 2010 | NRuby ンククリエイティブ |
| <u>書籍一覧ページ</u> | <u> </u> |
| | |

URL: <u>http://202.26.158.109/~ykawano/book_show.php?id=book_id</u> ※ book_idに書籍IDをGETで渡す

Tokyo University of Information Sciences

書籍情報の表示プログラム

書籍リストからGETで受け取ったデータを表示



Tokyo University of Information Sciences

書籍情報の削除プログラム







<u>参考:JSPの場合のDBアクセス(1)</u>

■ 書籍の一覧表示プログラム



参考:JSPの場合のDBアクセス(2)

書籍の一覧表示プログラム(続き)

```
<%= rs.getString("title")%>
     <%= rs.getString("category_name")%>
     <%= rs.getString("publisher_name")%> 
                                                       フィールドの値
     <%= rs.getInt("year")%>
                                                       を表形式で出力
     <a href='/~ykawano/book_show?id="
        <%= rs.getString("id")%>>表示</a>¥t
        <a href='/~ykawano/book_del?id="
        <%= rs.getString("id")%>>削除</a>
<%
 } catch (Exception e) {
    e.printStackTrace();
 } finally {
    try { rs.close(); } catch (Exception e) {}
    try { stmt.close(); } catch (Exception e) {}
    try { con.close(); } catch (Exception e) {}
%>
```