

Introduction to Databases, Fall 2018

Project Final Report (Nov 8, 2018)

Student ID 2013311659

Name 곽 창 근

1. Objective of project

A. Objective

책 구매 사이트를 만들고자 하였다. “한빛출판네트워크”라는 온라인 책 구매 사이트를 참조하였다. 회원가입, 로그인, 책 구매 등의 책 구매 사이트의 일반적인 기능들을 구현하고자 하였다.

B. Functionality

1) 회원가입

아이디, 이름, 생년월일, 성별, 이메일 주소 등을 입력해주면 회원으로 등록해주는 간단한 회원가입 기능이다.

→ 구현 성공. 회원 정보를 더 추가해서 ID, 비밀번호, 이름, 성별, 생년월일, 전화번호, 이메일, 직업을 입력해주면 회원가입을 할 수 있게 하였다.

2) 로그인

회원가입이 되어있을 경우, ID, PW가 일치한다면 해당 회원으로 로그인 해주는 기능이다.

→ 구현 성공. 로그인을 Manager 로그인, 일반 회원 로그인 2가지로 구성하였다. 로그인 시 둘 간의 보이는 화면이 다르다.

3) 마이페이지

회원 정보를 간단히 보여주는 페이지이다. 아이디, 이름, 마일리지, 구매한 책 리스트를 보여줄 계획이다.

→ 구현 성공. 마일리지를 충전할 수 있는 공간과 내 개인정보가 나타나지는 공간, 구매한 책 정보를 나타내는 공간으로 이루어져 있다.

4) 마일리지 충전

마이페이지에서 충전할 마일리지 금액을 입력하고 “충전” 버튼을 누르면 마일리지 충전을 할 수 있는 기능이다. 마일리지를 충전할 때 특별한 조건은 넣지 않을 계획이다.

→ 구현 성공. 마이페이지에 마일리지를 충전할 수 있는 공간을 만들었다. 충전을 위한 특별한 조건은 넣지 않았다.

5) 책 검색

검색어를 입력해주면 해당 검색어가 들어있는 책 목록을 띄워주는 검색 기능이다.

→ 구현 성공. 검색어를 입력하면 해당 검색어가 들어있는 책을 찾아준다.

6) 책 구매

책을 구매할 수 있는 충분한 마일리지가 있을 경우 책 구매 버튼을 눌러 책을 구매할 수 있다.

→ 구현 성공. 책 정보 화면에서 만약 책 구매를 위한 충분한 마일리지를 갖고 있다면 책을 구매할 수 있다. 구매한 책은 마이페이지에서 확인할 수 있다.

7) 리뷰 쓰기

만약 해당 책을 구매하였을 경우, 리뷰를 작성할 수 있다. 리뷰는 별점(0개~5개)과 텍스트(500자 이하)로 만들 계획이다.

→ 구현 성공. 책을 구매한 사람만 리뷰를 쓸 수 있으며 한 사람당 리뷰는 최대 1개까지만 작성 가능하다.

8) 관리자의 책 추가

→ 처음 Proposal 보고서 작성 시에는 없었지만 온라인 책 구매 사이트를 만드는데 있어서 필요하다 느꼈기에 새로 추가한 기능이다. 새로운 책 종류(book_type)와 해당 책 종류에 대한 책(book)을 추가할 수 있다.

2. DB Design

ER diagram과 Relational data model이다.

중간보고서 이후에 한가지가 추가되었는데, Review 테이블에 title이 추가되었다.

A. ER diagram

Entity Type은 Customer, Manager, Book, Book_type 이렇게 4개로 구성되어 있다.

전체적인 구성을 간략하게 말하자면, 소비자(Customer)는 특정 제목, 저자, 가격 등을 가진 (Book_Type) 책(Book)을 구매할 수 있고, 서버 관리자(Manager)는 소비자가 구매할 책을 서버에 등록해주는 역할을 한다. 소비자는 책 정보 페이지에 리뷰를 등록할 수 있다.

Manager : 책(Book)을 추가할 수 있다. 소비자(Customer)가 원하는 책 종류(Book_type)의 책 개체(Book)를 등록해줘야 소비자는 책을 구매할 수 있다. Mid가 키 특성이다.

Customer : 책(Book)을 구매할 수 있는 사용자이다. 책을 구매하면 해당 책의 정보(Book_type) 페이지에 리뷰를 추가할 수 있다. 소비자 이름(name), 성별(sex), 이메일(email) 등등을 attribute로 갖고, Cid가 키 특성이다.

Book_Type : 책 정보 틀 역할을 한다. Book_type을 만들면 이후 Manager가 해당 Book_type의 Book을 만들 수 있다. 책 정보 틀 역할을 하기에 책에 대한 정보인 책 제목(name), isbn 코드(isbn), 가격(price), 저자(author) 등을 attribute로 갖고, isbn이 키 특성이다.

Book : Book_type에 속하는 실제 책 개체이다. 책 구매 사이트이기 때문에 한 종류의 책이 여러권 있을 수 있고, 이는 고유의 bid를 갖는 Book entity 개수를 늘려서 여러권의 책을 만들 수 있다. Book은 반드시 Book_type이 있어야 존재할 수 있기에 Book-Book_type 간의 관계에서 Book은 이중선으로 이어져 있다.

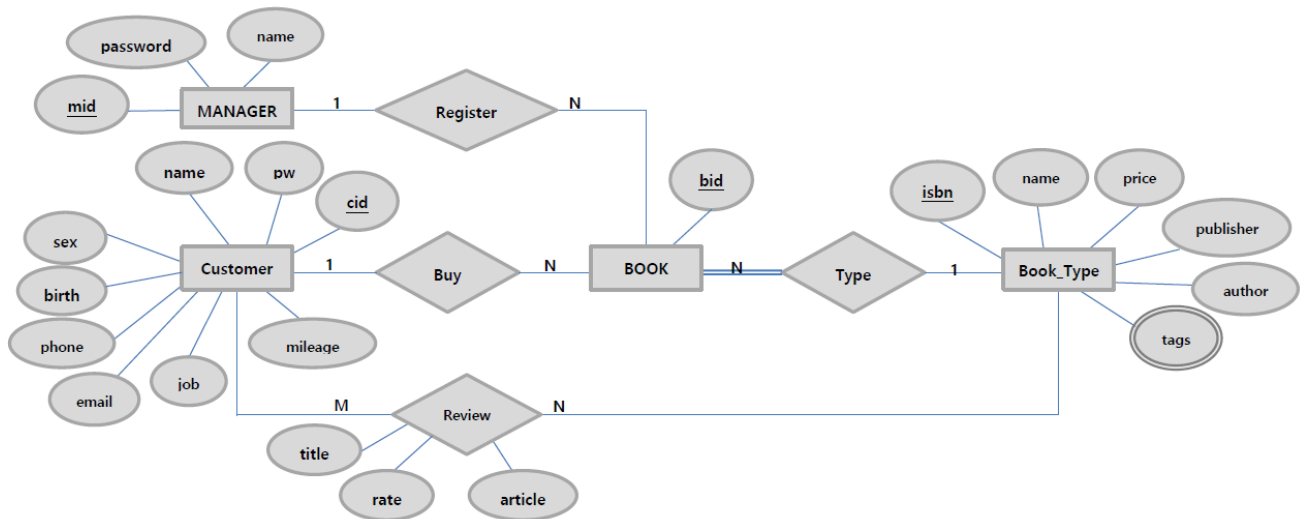
B. Relational data model

Relational data model로 만들면서 기존 Entity에 추가하여 Tag, Review 테이블이 추가되었다.

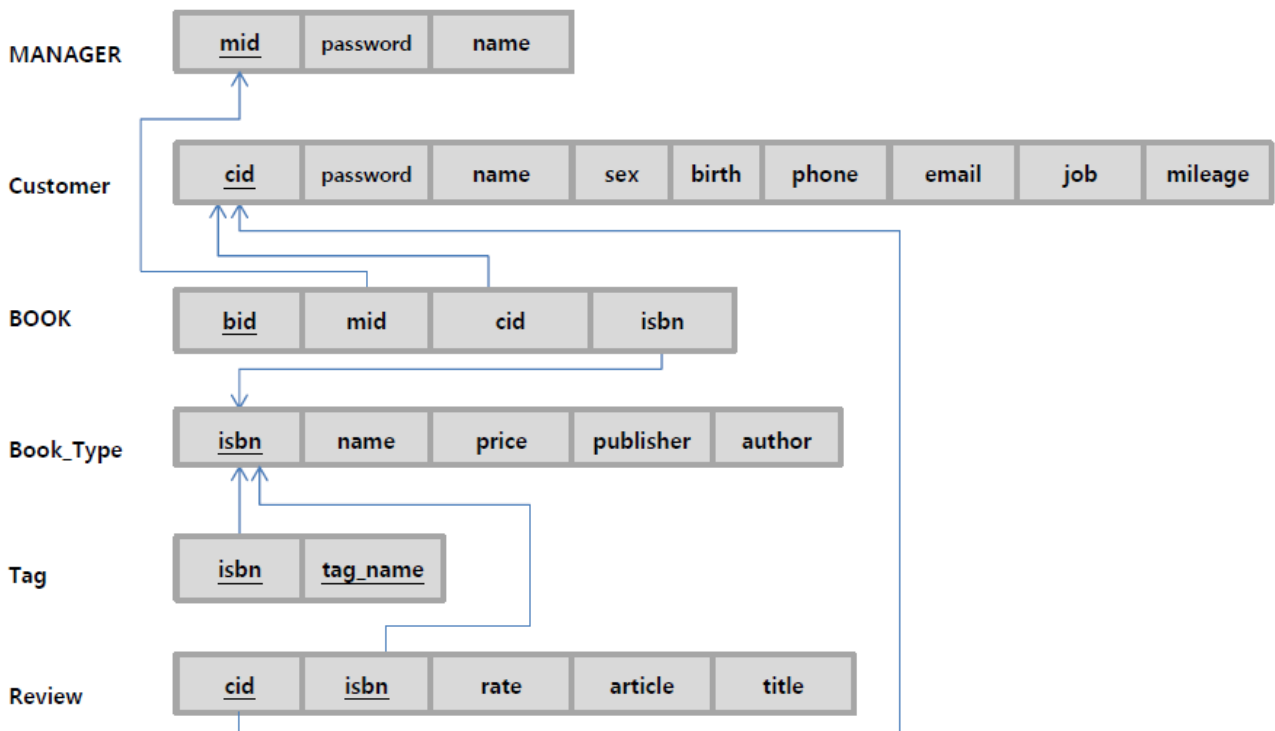
Tag : Book_type에 붙는 태그이다. 기존 Book_type의 multi-valued attribute였던 tags가 테이블로 바뀐 것이다. 그렇기에 Book_type의 key인 isbn을 foreign key로 갖는다. Isbn, tag_name이 합쳐져서 키 역할을 한다.

Review : Customer와 Book_Type 사이의 Relationship이었던 Review가 테이블로 추가되었다. Customer가 책을 구매하면, 해당 책의 Book_type 정보 페이지에 리뷰를 작성할 수 있다. 작성하는 정보는 title, rate, article이다. Isbn과 cid가 합쳐져서 키 역할을 한다.

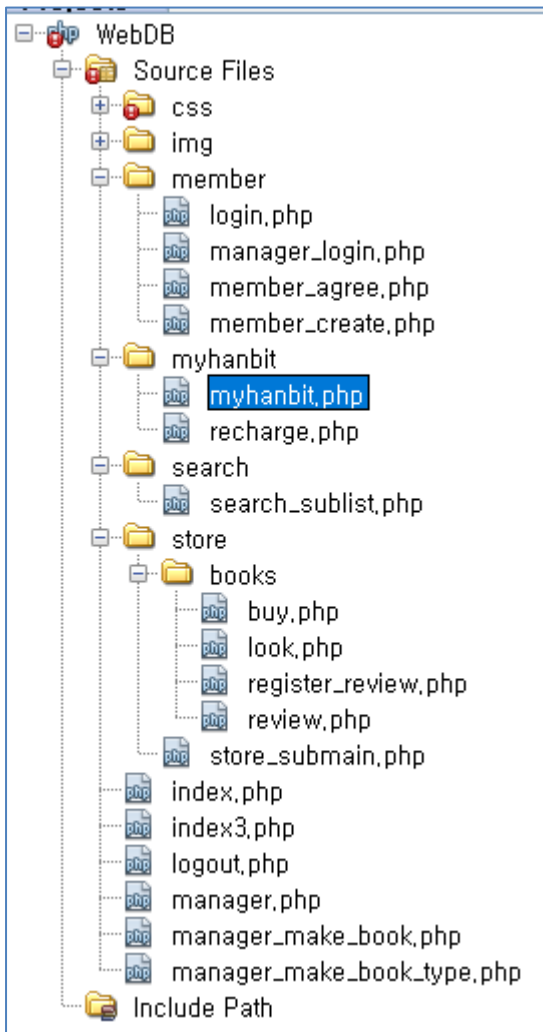
A. ER diagram



B. Relational data model



3. Implementation and results



파일 구성은 위와 같이 이루어져 있다. ‘css’ 폴더는 css파일들을 저장, ‘img’ 폴더는 이미지 파일들을 저장하였다. Source Files 내부 중 가장 상위 폴더에 있는 ‘index.php’ 파일은 메인 화면이며, 이 파일 외에도 매니저 페이지 파일, 로그아웃 페이지 파일 등이 위치해 있다. Member 폴더는 로그인, 회원가입을 관장하는 파일들이 들어있다. Myhanbit 폴더는 마이페이지, 그리고 마이페이지 내에서 진행하는 마이페이지 충전을 해주는 파일들이 들어있다. Search 폴더는 검색기능을 담당하는 파일이 들어있다. Store 폴더 안의 Books 폴더에는 책 정보, 리뷰 관련, 책 구매 관련 파일들이 들어있다.

1) 회원가입

회원정보 필수입력

회원정보 (필수)

* 는 필수 입력 사항입니다.

아이디 *

비밀번호 *

비밀번호 확인 *

이름 *

성별 *

생년월일 *

이메일 *

휴대전화 *

직업 *

회원가입하기

위 사진이 회원가입 페이지의 모습이다. 모든 정보를 입력해야 회원가입을 할 수 있다.

```

<form method="post" action="./member_agree.php">
  <fieldset>
    <legend>회원정보 필수입력</legend>
    <p class="register_tit"><strong>회원정보 <span>(필수)</span></strong></p>
    <p class="register_cap"><span>* </span></span>는 필수 입력 사항입니다.</p>

    <div class="register_essential">
      <div class="register_li">
        <div class="i_tit"><strong>아이디<span>* </span></strong></div>
        <div class="i_con">
          <input type="text" name="user_id" id="user_id" class="i_text" value="">
        </div>
      </div>

      <div class="register_li">
        <div class="i_tit"><strong>비밀번호<span>* </span></strong></div>
        <div class="i_con">
          <input type="password" name="user_pw" id="user_passwd" class="i_text" value="">
          <span id="pw_tip">8~20자리의 영문, 숫자 조합을 권장합니다</span>
        </div>
      </div>

      <div class="register_li">
        <div class="i_tit"><strong>이름<span>* </span></strong></div>
        <div class="i_con">
          <label>
            <input type="text" name="user_name" id="user_name" value="" class="i_text">
          </label>
        </div>
      </div>

      <div class="register_li">
        <div class="i_tit"><strong>성별<span>* </span></strong></div>
        <div class="i_con">
          <label class="ra_label">
            <input type="radio" name="user_sex" id="user_sex" value="m" class="i_radio" checked="">
            <span>남성</span>
          </label>
          <label class="ra_label">

```

구성은 위와 같이 하나의 <form></form> 안에 여러 값들을 input으로 입력 받아서 다시 현재 페이지(“./member_agree.php”)로 값을 넘긴다.

```
<?php
if( isset($_POST["user_id"]) && ($_POST["user_pw"]=="" || $_POST["user_name"]=="" || $_POST["user_sex"]=="" || $_POST["user_birth_year"]=="" ||
?>
<script>
    alert('모든 정보를 정확히 입력해주세요. ');
</script>
<?php
}
else if ( isset($_POST["user_id"]) ) {
    $id = $_POST["user_id"];
    $pw = $_POST["user_pw"];
    $name = $_POST["user_name"];
    if( $_POST["user_sex"] == "m" )
        $sex = 0;
    else if( $_POST["user_sex"] == "w" )
        $sex = 1;
    $d = $_POST["user_birth_year"]."-".$_POST["user_birth_month"]."-".$_POST["user_birth_day"];
    $birth = date_create($d)->format("Y-m-d");
    $email = $_POST["user_email_1"]."@".$_POST["user_email_2"];
    $phone = $_POST["user_phone_1"]."-".$_POST["user_phone_2"]."-".$_POST["user_phone_3"];
    $job = (int)$_POST["user_job"];
    $mileage = 0;

    $db_user="root";
    $db_pass="1234";
    $db_type="mysql";
    $db_host="localhost";
    $db_name="project";
    $tb_name="customer";
    $dsn = "$db_type:host=$db_host; dbname=$db_name; charset=utf8";

    try{
        $pdo=new PDO($dsn, $db_user, $db_pass);
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $sql = "insert into customer (cid, password, name, sex, birth, phone, email, job, mileage) values ( :id, :pw, :name, :sex, :birth, :pho
```

위의 사진이 member_agree.php의 맨 처음 부분인데, 보듯이 처음 if문을 통해서 값들이 모두 정확히 입력되었는지를 확인해서 입력이 안되었다면 alert창을 띄워주고, 제대로 입력되었다면 mysql을 불러서 사진의 \$sql에 저장한 sql문을 통해서 회원 정보를 저장해준다.

2) 로그인

로그인



매니저 로그인



일반 로그인 창과 매니저 로그인 창이다. 서로 구분을 지어주기 위해서 매니저 로그인 방식은 둘 다 똑같고, 차이점은 mysql을 불러서 Customer 테이블에서 찾을 건지, Manager 테이블에서 찾을지의 차이 뿐이다.

```

<form method="post" action="./login.php">
  <?php if(isset($warn)) print $warn; ?>
  <h3> 로그인 </h3>
  <fieldset>
    <legend>로그인</legend>

    <label class="i_label" for="login_id"><strong style="position: absolute;"></strong>
      <input name="user_id" id="user_id" type="text" value="" class="i_text" placeholder="아이디">
    </label>

    <label class="i_label" for="login_pw"><strong style="position: absolute;"></strong>
      <input name="user_pw" id="user_pw" type="password" value="" class="i_text" placeholder="비밀번호">
    </label>

    <label>
      <input type="submit" name="login_btn" id="login_btn" value="로그인" class="btn_login">
    </label>

  </fieldset>
</form>

```

로그인 페이지의 핵심 코드는 위 부분으로, id, pw를 입력하고 로그인 버튼을 누르면 다시 이 페이지("./login.php")에 해당 값들을 post 해준다.

```

$sql = "select * from $tb_name where cid = :id and password = :pw";

$stmth = $pdo->prepare($sql);
$stmth->bindValue(':id', $id, PDO::PARAM_STR);
$stmth->bindValue(':pw', $pw, PDO::PARAM_STR);
$stmth->execute();
$count = $stmth->rowCount();

if($count<1) {
  //로그인 실패.
  $warn = "ID, 패스워드를 정확히 입력해주세요.";
  //Header("Location:./member/login.php?login=0");
}
else {
  //로그인 성공. 로그인 정보를 세션에 저장. index.php 페이지로 돌아감.
  $row = $stmth->fetch(PDO::FETCH_ASSOC);
  $name = $row['name'];

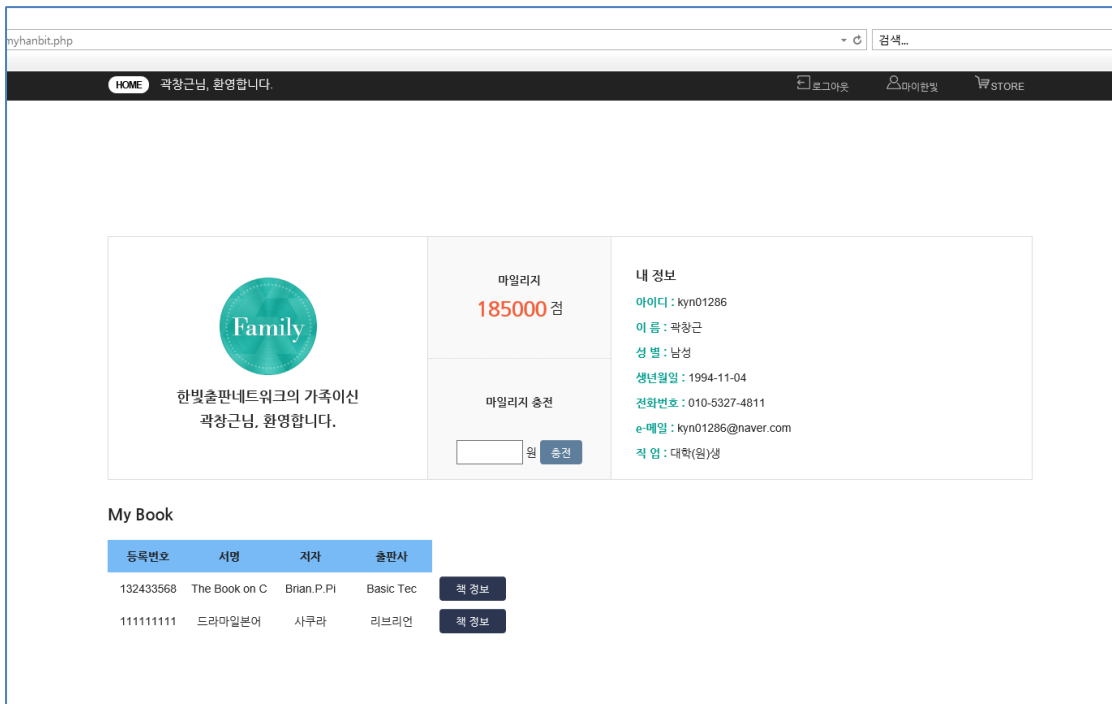
  session_start();
  $_SESSION['ses_id'] = $id;
  $_SESSION['ses_pw'] = $pw;
  $_SESSION['ses_name'] = $name;

  Header("Location:./index.php");
}

```

해당 post 값들을 받고 위 사진의 \$sql에 저장한 sql문을 통해서 해당 id와 pw가 맞는지 확인한다. 맞았으면 세션을 실행하여서 세션에 ses_id, ses_pw, ses_name 정보를 저장한다. 매니저 로그인 같은 경우는 일반 사용자 로그인과 차이를 주기 위해서 세션에 등록하는 변수 이름을 ses_mgr_id, ses_mgr_pw, ses_mgr_name 으로 저장하였다.

3) 마이페이지



위 사진은 마이페이지 실행 사진이다. 가운데에는 마일리지 얼마 남았는지 보여주고, 마일리지 충전을 원하면 충전할 수도 있다. 오른쪽 화면에는 내 정보를 나열해주며, 아래 화면에는 내가 구매한 책리스트를 띄워준다.

내 정보 및 마일리지는 Customer 테이블에서 세션에 저장되어있는 id에 맞는 정보들을 불러와서 띄워준다. 해당 코드는 아래와 같다.

```
<!-- 최근 구매이력 -->
<div class="sm_myorder">
  <p class="tit">내 정보</p>
  <p><span>아이디 : </span><?=htmlspecialchars($id)?></p>
  <p><span>이름 : </span><?=htmlspecialchars($name)?></p>
  <p><span>성별 : </span><?php if($sex==0) {print "남성";} else {print "여성";}?></p>
  <p><span>생년월일 : </span><?=htmlspecialchars($birth)?></p>
  <p><span>전화번호 : </span><?=htmlspecialchars($phone)?></p>
  <p><span>e-메일 : </span><?=htmlspecialchars($email)?></p>
  <p><span>직업 : </span><?php
    switch( (int) $job ) { ...58 lines }
  ?></p>
```

내가 구매한 책 정보는 Books, Book_type 테이블을 참조해야 하기에 다시 mysql을 부른다. Join 해서 하는 방법도 있겠지만, 필자는 아래와 같이 sql을 Books, Book_type에 한 번씩 총 두 번 보

내는 방법을 통해 구현하였다.

```
<?php
try{
    //sql2는 책 정보
    $sql= "select isbn from book where cid = :id";

    $stmh = $pdo->prepare($sql);
    $stmh->bindValue(':id', $id, PDO::PARAM_STR);
    // $stmh2->bindValue(':pw', $pw, PDO::PARAM_STR);
    $stmh->execute();

    while($temp_row = $stmh->fetch(PDO::FETCH_ASSOC)) {
        $sql = "select * from book_type where isbn = :isbn";

        $stmh2 = $pdo->prepare($sql);
        $stmh2->bindValue(':isbn', $temp_row['isbn'], PDO::PARAM_STR);
        // $stmh2->bindValue(':pw', $pw, PDO::PARAM_STR);
        $stmh2->execute();
        $row = $stmh2->fetch(PDO::FETCH_ASSOC);

        <tr>
            <td align="center" width=100 style="word-break:break-all"><?htmlspecialchars($row['isbn'])?></td>
            <td align="center" width=100 style="word-break:break-all"><?htmlspecialchars($row['name'])?></td>
            <td align="center" width=100 style="word-break:break-all"><?htmlspecialchars($row['author'])?></td>
            <td align="center" width=100 style="word-break:break-all"><?htmlspecialchars($row['publisher'])?></td>
        </tr>
        <tr>
            <td colspan="4">
                <form name="look_book" method="get" action="/store/books/look.php" style="margin: 0 auto;">
                    <div style="text-align: center">
                        <input type="hidden" name="isbn" value=<?htmlspecialchars($row['isbn'])?>>
                        <input type="submit" value="책 정보" style="background: #2d3651; width: 82px; height: 30px; margin: 0 auto;">
                    </div>
                </form>
            </td>
        </tr>
    </tr>
</tr>
<?php
}
} catch (PDOException $Exception) {
    die('Error: '.$Exception->getMessage());
}
```

우선 해당 id로 저장되어 있는 Book의 튜플들을 모두 불러온 후, 튜플의 isbn에 맞는 Book_type 정보를 하나씩 찾아서 게시하는 방식으로 하였다.

4) 마일리지 충전

```
<!-- 마일리지 -->
<div class="sm_my_mileage">
    <dl class="mileage_section1">
        <dt>마일리지</dt>
        <dd><span><?htmlspecialchars($mileage)?></span> 점</dd>
    </dl>
    <dl class="mileage_section2">
        <dt>마일리지 충전</dt>
        <br>
        <form name="recharge" id="recharge" method="post" action="./recharge.php">
            <div style="display:inline;">
                <span><input name="recharge" id="recharge" type="text" value="" style="width: 82px; height: 30px;"> 원
                <input type="submit" value="충전" style="background: #5e7e9b; width: 52px; height: 30px; margin:0; border: 0; padding: 2px 10px;">
            </div>
        </form>
    </dl>
</div>
<!-- //마일리지 -->
```

마일리지는 마이페이지에서 위의 form을 통해 얼마를 충전할 것인지 “recharge.php” 파일로 보내 주면 “recharge.php” 에서 아래 코드를 통해 mysql을 통해서 해당 id의 마일리지 값을 업데이트

해준 후 다시 마이페이지로 돌아온다.

```
<?php
session_start();

if(isset($_POST["recharge"]) && isset($_SESSION["ses_id"])) {
    $id = $_SESSION["ses_id"];
    $pw = $_SESSION["ses_pw"];
    $recharge = $_POST["recharge"];

    $db_user="root";
    $db_pass="1234";
    $db_type="mysql";
    $db_host="localhost";
    $db_name="project";
    $dsn      = "$db_type:host=$db_host; dbname=$db_name; charset=utf8";
    $tb_name="customer";

    try{
        $pdo=new PDO($dsn, $db_user, $db_pass);
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $sql = "update $tb_name set mileage = mileage + :recharge where cid = :id and password = :pw";

        $stmh = $pdo->prepare($sql);
        $stmh->bindValue(':recharge', $recharge, PDO::PARAM_INT);
        $stmh->bindValue(':id', $id, PDO::PARAM_STR);
        $stmh->bindValue(':pw', $pw, PDO::PARAM_STR);
        $stmh->execute();


        Header("Location:./myhanbit.php");
    } catch (PDOException $Exception) {
        die('Error: '.$Exception->getMessage());
    }
}

?>
```

아래 사진은 실제 마일리지 충전 전과 후의 사진이다.

 한빛출판네트워크의 가족이신 공새은님. 환영합니다.	마일리지 0 점
	마일리지 충전 <input type="text" value="100000"/> 원 <input type="button" value="충전"/>

→

 한빛출판네트워크의 가족이신 공새은님. 환영합니다.	마일리지 100000 점
	마일리지 충전 <input type="text"/> 원 <input type="button" value="충전"/>

5) 책 검색

우산



검색하고자 하는 책 제목을 입력해서 검색버튼을 누르면,

검색 결과는 3건입니다.

검색

등록번호	서명	가격	저자	출판사	
1231231231	우산	3100	송이	사이코드라마	책 정보
1231231232	우산2	6200	송이	사이코드라마	책 정보
1231231233	무지개우산	9900	송이	사이코드라마	책 정보

아래와 같이 해당 검색어가 들어있는 책들 목록과 검색된 결과 수를 띄워준다.

```
<?php
$db_user="root";
$db_pass="1234";
$db_type="mysql";
$db_host="localhost";
$db_name="project";
$dsn = "$db_type:host=$db_host; dbname=$db_name; charset=utf8";

$bookname = $_POST["bookname"];
$search_key = '%'.$bookname.'%';

try{
    $pdo=new PDO($dsn, $db_user, $db_pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "select * from book_type where name like :book_name";

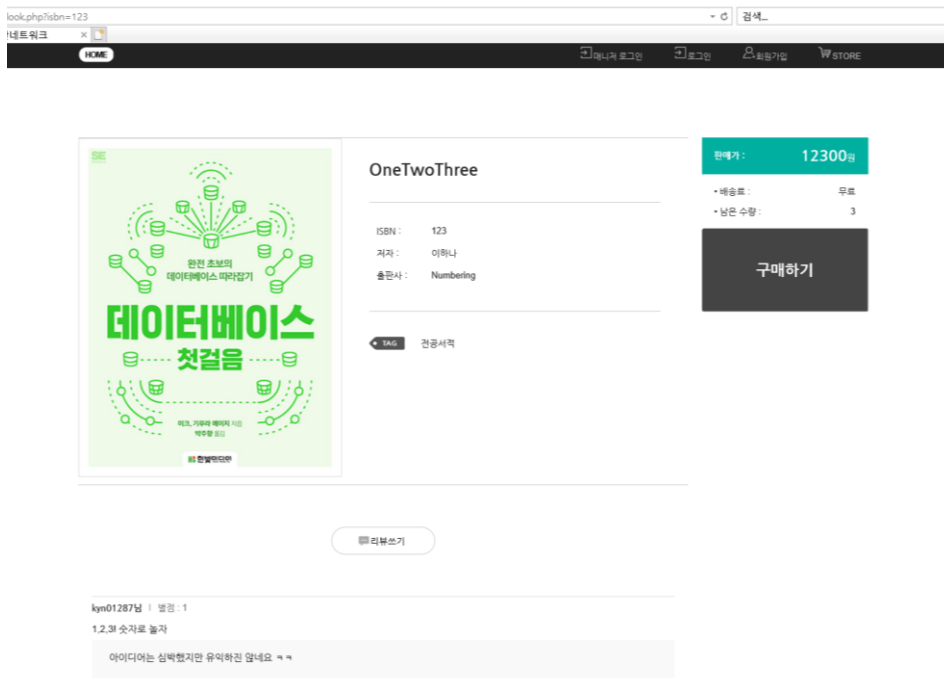
    $stmh = $pdo->prepare($sql);
    $stmh->bindValue(':book_name', $search_key, PDO::PARAM_STR);
    $stmh->execute();
    $count = $stmh->rowCount();

    if($count<1) {
        print "조회된 검색 결과가 없습니다.<br><br>";
    }
    else {
        print "검색 결과는 ".$count."<br><br>";
    }
} catch (PDOException $Exception) {
    die("Error: ".$Exception->getMessage());
}
?>
```

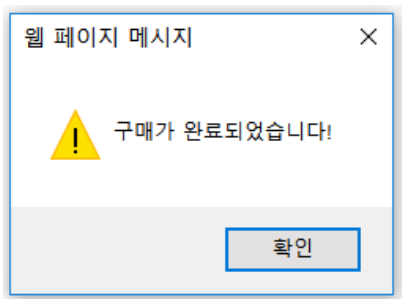
검색은 “Search_sublist.php” 파일에서 이루어진다. “Search_sublist.php”로 검색문을 보내주면, 해당 파일에서 위 코드를 통해서 해당 검색문이 들어있는 책 제목을 찾아준다. 해당 검색어 앞뒤에 ‘%’을 붙여주고 Like을 통해 찾는 방식으로 해당 검색어가 들어있는 책 제목을 찾을 수 있었다. 그래서 검색된 결과가 있다면 아래의 코드를 통해 while문을 돌면서 해당 책 정보들을 모두 출력해준다.

```
<table width="900" boarder="1" cellspacing="0" cellpadding="8">
  <tbody>
    <tr>
      <th>등록번호</th>
      <th>서명</th>
      <th>가격</th>
      <th>저자</th>
      <th>출판사</th>
    </tr>
    <?php
    while($row = $stmh->fetch(PDO::FETCH_ASSOC)) {
      ?>
      <tr>
        <td align="center" width=100 style="word-break:break-all"><?=htmlspecialchars($row['isbn'])?></td>
        <td width=300 style="word-break:break-all"><?=htmlspecialchars($row['name'])?></td>
        <td align="center" width=80 style="word-break:break-all"><?=htmlspecialchars($row['price'])?></td>
        <td align="center" width=100 style="word-break:break-all"><?=htmlspecialchars($row['author'])?></td>
        <td align="center" width=200 style="word-break:break-all"><?=htmlspecialchars($row['publisher'])?></td>
        <td>
          <form name="look_book" method="get" action="../store/books/look.php" style="margin: 0 auto; ">
            <div style="text-align: center">
              <input type="hidden" name="isbn" value=<?=htmlspecialchars($row['isbn'])?>>
              <!--
              <input type="hidden" name="name" value=<?=htmlspecialchars($row['name'])?>>
              <input type="hidden" name="price" value=<?=htmlspecialchars($row['price'])?>>
              <input type="hidden" name="author" value=<?=htmlspecialchars($row['author'])?>>
              <input type="hidden" name="publisher" value=<?=htmlspecialchars($row['publisher'])?>>
              -->
              <input type="submit" value="책 정보" style="background: #2d3651; width: 82px; height: 30px; margin: 0 auto; ">
            </div>
          </form>
        </td>
      </tr>
    <?php
    }
  </tbody>
</table>
```

6) 책 구매



위 사진은 책 정보 페이지이다. 위 페이지의 오른쪽의 “구매하기” 버튼을 누르면 마일리지가 판매가 이상으로 충분히 있고 남은 재고가 있다면 구매가 성공적으로 이루어진다.



구매에 성공하면 위의 메시지 창이 뜨고,



실제 마이페이지로 가보면 책 가격만큼 마일리지가 차감되었으며 “My Book” 리스트에 해당 책이 추가되어 있다.

책 정보는 “look.php” 파일에서 보여주며, “구매하기 버튼을 누르면 아래 코드를 통해서 책의 isbn, price 정보를 “buy.php” 파일로 post 해준다.

```
<form method="post" action="./buy.php">
<!-- 결제 정보 -->
<fieldset>
<legend>결제하기</legend>
<label class="payment_box_curr">
<p><span class="pbl"><strong>판매가 : </strong></span><span class="pbr"><strong><?htmlspecialc
</label>

<div class="shopping_charge">
<div class="shopping_charge_tit"><span>&#149; 배송료 : </span>무료</div>
</div>
<div class="shopping_charge">
<div class="shopping_charge_tit"><span>&#149; 남은 수량 : </span><?htmlspecialchars($count)
</div>

<label>
<button name="btnBuy" id="btnBuy" type="submit" value="구매하기" class="btn_buy"> 구매하기</button>
</label>
<input type="hidden" name="book_isbn" id="book_isbn" value=<?htmlspecialchars($isbn)?>>
<input type="hidden" name="book_price" id="book_price" value=<?htmlspecialchars($price)?>>

</fieldset>
</form>
```

아래 코드는 “buy.php”에서 실행되는 코드로, 해당 id의 마일리지가 얼마인지, 책 재고 수량이 남아있는지를 확인하는 부분이다.

```
try{
    $pdo=new PDO($dsn, $db_user, $db_pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "select * from customer where cid = :user_id";

    $stmh = $pdo->prepare($sql);
    $stmh->bindValue(':user_id', $_SESSION['ses_id'], PDO::PARAM_STR);
    $stmh->execute();
    //$pdo->commit();

    $row = $stmh->fetch(PDO::FETCH_ASSOC);
    $user_mileage = $row['mileage'];
} catch (PDOException $Exception) {
    //$pdo->rollBack();
    die('Error: ', $Exception->getMessage());
}

if( $book_price <= $user_mileage) {

    //만약 bid 남은거 없으면 rollBack, sql문을 통해 bid 중 아직 안 나눠준거 있는지 확인
    //유저에게 책 등록
    try{
        $pdo=new PDO($dsn, $db_user, $db_pass);
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $sql = "select bid from book where isbn = :isbn and cid is null";

        $stmh = $pdo->prepare($sql);

        $stmh->bindValue(':isbn', (int) $book_isbn, PDO::PARAM_INT);
        $stmh->execute();
        //$pdo->commit();

        $row = $stmh->fetch(PDO::FETCH_ASSOC);
        $bid = $row['bid'];
        print "bid is $bid<br>";
    } catch (PDOException $Exception) {
        //$pdo->rollBack();
        die('Error: ', $Exception->getMessage());
    }
}
```

위 코드 실행결과 문제가 없다면 아래 코드를 통해서 cid가 부여되지 않은 Book 중 하나에 구매한 유저의 cid를 넣어주고, 구매자의 마일리지를 깎아준다. 이를 통해 구매가 완료된다.

```
try{
    $pdo=new PDO($dsn, $db_user, $db_pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "update book set cid = :user_id where bid = :bid";

    $stmh = $pdo->prepare($sql);

    $stmh->bindValue(':bid', $bid, PDO::PARAM_STR);
    $stmh->bindValue(':user_id', $_SESSION['ses_id'], PDO::PARAM_STR);
    $stmh->execute();
    //$pdo->commit();

    //$row = $stmh->fetch(PDO::FETCH_ASSOC);
    //$user_mileage = $row['mileage'];
} catch (PDOException $Exception) {
    //$pdo->rollback();
    die('Error: '.$Exception->getMessage());
}

//유저 마일리지 깎아줌
try{
    $pdo=new PDO($dsn, $db_user, $db_pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

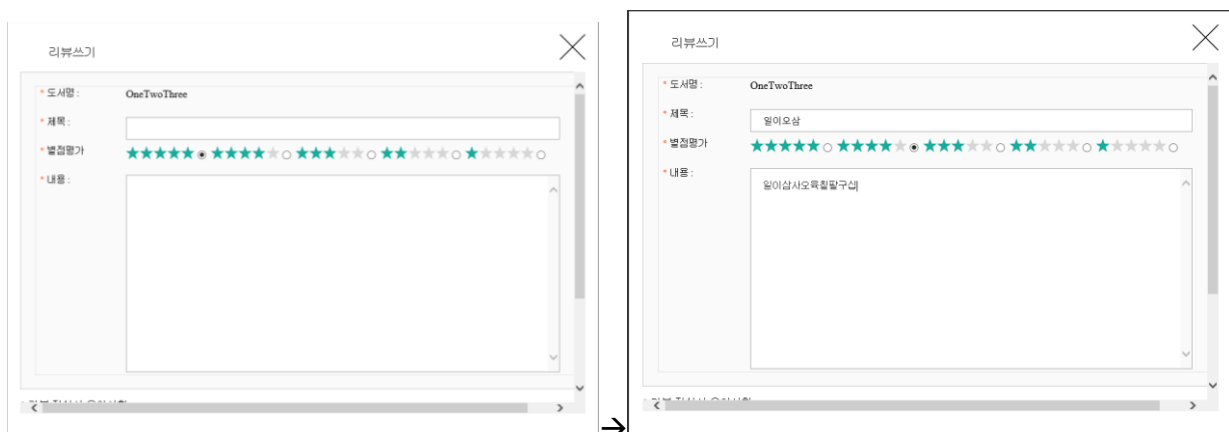
    $sql = "update customer set mileage = mileage - :mileage where cid = :user_id";

    $stmh = $pdo->prepare($sql);
    $stmh->bindValue(':user_id', $_SESSION['ses_id'], PDO::PARAM_STR);
    $stmh->bindValue(':mileage', (int) $book_price, PDO::PARAM_INT);
    $stmh->execute();
    //$pdo->commit();

    //$row = $stmh->fetch(PDO::FETCH_ASSOC);
    //$user_mileage = $row['mileage'];
} catch (PDOException $Exception) {
    //$pdo->rollback();
    die('Error: '.$Exception->getMessage());
}
```

7) 리뷰 쓰기

책 정보 페이지에서 책을 구매하였다면 리뷰쓰기 버튼을 눌러서 책을 구매할 수 있다.



리뷰쓰기 버튼을 누르면 위와 같은 창이 뜨고, 리뷰 정보들을 입력해서 리뷰를 작성할 수 있다.

작성 후에 등록 버튼을 누르면 아래 사진과 같이 책정보 페이지 아래에 리뷰가 정상적으로 등록된 것을 볼 수 있다.

ISBN : 123

저자 : 이하나

출판사 : Numbering

TAG : 전공서적

배송료 : 무료

남은 수량 : 2

구매하기

리뷰쓰기

kyn01287님 | 별점 : 1

1,2,3! 숫자로 놀자

아이디어는 심박했지만 유익하진 않네요 ㅋㅋ

kyn01288님 | 별점 : 4

일리오삼

일이삼사오육칠팔구십

리뷰 쓰는 것은 “review.php” 파일에서 이루어지고, 역시 <form> 태그를 통해서 이 안에 값들(리뷰 제목, 별점평가, 내용)을 입력해주면 “register_review.php”라는 파일에서 리뷰 등록을 이루어준다.

```

$db_user="root";
$db_pass="1234";
$db_type="mysql";
$db_host="localhost";
$db_name="project";
$tb_name = "review";
$dsn = "$db_type:host=$db_host; dbname=$db_name; charset=utf8";

$book_isbn = $_POST["book_isbn"];
$book_price = $_POST["book_price"];

try{
    $pdo=new PDO($dsn, $db_user, $db_pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "insert into $tb_name (cid, isbn, rate, article, title) values (:cid, :isbn, :rate, :article, :title)";

    $stmt = $pdo->prepare($sql);
    $stmt->bindValue(':cid', $_SESSION['ses_id'], PDO::PARAM_STR);
    $stmt->bindValue(':isbn', $book_isbn, PDO::PARAM_STR);
    $stmt->bindValue(':rate', (int) $r_rate, PDO::PARAM_STR);
    $stmt->bindValue(':article', $r_content, PDO::PARAM_STR);
    $stmt->bindValue(':title', $r_title, PDO::PARAM_STR);
    $stmt->execute();
    //$pdo->commit();

    Header("Location:./look.php?isbn=$book_isbn");
} catch (PDOException $Exception) {
    //$pdo->rollBack();
    //die("Error : ".$Exception->getMessage());
}
?>
<script>
alert("리뷰는 한 번만 작성 가능합니다. (여러 권 샀어도 한 번만 가능)");
//history.go(-1);//첫번째 방법
//history.back();//두번째 방법
location.href = "../store_submain.php";//해당 주소로 이동하는방법
</script>
<?php
}
?>

```

위 코드는 리뷰를 등록하는 과정으로, 유저 id, isbn, 리뷰로 적은 정보들을 “insert into”를 통해 입력해준다. 리뷰를 여러 번 등록하려고 한다면 경고창이 뜨며 등록이 실패된다.

8) 관리자의 책 추가

새롭게 Book_type을 추가합니다.
정보를 모두 입력해주세요.

isbn

Name

Price

Publisher

Author

tags

☐ 인공지능
 ☐ 프로그래밍 언어

☐ 웹
 ☐ 게임/취미

☐ 자기계발
 ☐ 사진/예술

☐ 외국어
 ☐ 디자인

☐ 시집
 ☐ 소설

☐ 경제/경영
 ☐ 컴퓨터책

등록하기

등록된 Book_type의 Book을 추가합니다.
정보를 모두 입력해주세요.

bid

isbn

등록하기

등록번호	서명	가격	저자	출판사
0		0		
123	OneTwoThree	12300	이하나	Numbering
13131	태그 부자 김태그	9500	김태그	태그 컴퍼니
123112	인공지능개론	58000	이지철	성균관대학교
13111311	토지	13000	박경림	고보문고
111111111	드라마일본어	15000	사쿠라	리브리언
131313131	데이터베이스개론	99999	이종욱	성균관대학교
132433568	The Book on C	20000	Brian.P.Pi	Basic Tec
894500610	Computer Networking : A Top-Down Approach	36000	James F Kurose	PEARSON
978893448	하늘과 바람과 별과 시 : 윤동주 시집	8000	윤동주	해원
978898241	(개념을 콕콕 잡아주는) 데이터베이스	28000	김경향	고양
1231231231	우산	3100	송이	사이코드라마
1231231232	우산2	6200	송이	사이코드라마
1231231233	무지개우산	9900	송이	사이코드라마
2121141231	덴서플로우 기초	18000	이지철	성균관대학교
2147483647	Startup!	15000	Kim Belly	startNew

bid	mid	cid	isbn
0			0
1	bluesquanium	kyn01286	132433568
2	bluesquanium	kyn01287	123
3	bluesquanium	kyn01286	111111111
4	bluesquanium	kyn01288	123
5	bluesquanium		123
6	bluesquanium		123
7	bluesquanium	kyn01287	111111111
8	bluesquanium	kyn01287	111111111

관리자로 로그인 하면 위와 같은 창이 뜨고 Book_type, Book을 등록하거나 현재 등록된 Book_type, Book 정보를 볼 수 있다. 위의 표가 Book_type이고, 아래 표가 Book 정보이다. 이 페이지는 “manager.php”에서 띄워준다. Book_type, Book을 띄워주는 방식은 책 검색 기능에서 책 정보 테이블을 띄워주는 방식과 같다. mysql을 통해서 모든 Book_type, Book 정보를 얻어와서 while 문을 돌면서 하나씩 띄워주는 것이다.

화면 왼쪽에는 책 등록하는 form이 2개가 있는데 첫번째가 Book_type을 추가, 두 번째가 Book을 추가하는 것이다. 아래 사진에 나와있는 Book을 등록해주는 form 태그를 보면 필요한 값들을 입력 받은 후 “등록하기” 버튼을 눌러주면 “manager_make_book.php” 또는 “manager_make_book_type.php” 파일을 통해서 등록해주는 것이다.

```

<form class="mgr" method="post" action="/manager_make_book.php">
  <fieldset>
    <legend>책 추가하기</legend>
    <p><strong>등록된 Book_type의 Book을 추가합니다.</strong></p>
    <p>정보를 모두 입력해주세요.</p>

    <div class="register_essential">
      <div class="register_li">
        <div class="i_tit"><strong>bid</strong></div>
        <div class="i_con">
          <input type="text" name="bid" id="bid" class="i_text" value="">
        </div>
      </div>

      <div class="register_li">
        <div class="i_tit"><strong>isbn</strong></div>
        <div class="i_con">
          <input type="text" name="isbn" id="isbn" class="i_text" value="">
        </div>
      </div>
    </div>

    <br>

    <div>
      <label>
        <input type="submit" name="make_btn" id="make_btn" value="등록하기" class="btn_blue2">
      </label>
    </div>
  </fieldset>
</form>

```

역시 mysql을 불러와서 등록을 한다. Book을 만드는 “manager_make_book.php”의 경우에는

```
$sql = "insert into $tb_name (bid, mid, isbn) values (:bid, :mid, :isbn)";
```

아래의 sql문을 이용해서 Book을 등록한다.

Book_type을 만드는 “manager_make_book_type.php”의 경위는 msyql을 2번 부르는데, 이는 Book_type을 등록하고, 해당 Book_type의 태그를 저장하는 Tags도 만들어주기 위함이다.

아래 사진 중 위의 sql문이 Book_type을 등록, 아래의 sql문이 Tags를 등록하는 것이다.

```

try{
  $pdo=new PDO($dsn, $db_user, $db_pass);
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

  // book_type에 등록
  $sql = "insert into $tb_name (isbn, name, price, publisher, author) values (:isbn, :name, :price, :publisher, :author)";

  $stmh = $pdo->prepare($sql);
  $stmh->bindValue(':isbn', (int) $isbn, PDO::PARAM_STR);
  $stmh->bindValue(':name', $name, PDO::PARAM_STR);
  $stmh->bindValue(':price', (int) $price, PDO::PARAM_INT);
  $stmh->bindValue(':publisher', $publisher, PDO::PARAM_STR);
  $stmh->bindValue(':author', $author, PDO::PARAM_STR);

  $stmh->execute();

  // tags가 있을 경우 tags에 등록
  if(isset($_POST["tags"])) {
    $tags = $_POST["tags"];
    $tb_name = "tags";
    foreach($tags as $tag) {
      $sql = "insert into $tb_name (isbn, tag_name) values (:isbn, :tag_name)";

      $stmh = $pdo->prepare($sql);
      $stmh->bindValue(':isbn', (int) $isbn, PDO::PARAM_STR);
      $stmh->bindValue(':tag_name', $tag, PDO::PARAM_STR);

      $stmh->execute();
    }
  }
}

```

태그를 어떻게 저장하느냐에 따라서 책 정보 페이지에서 뜨는 태그의 종류와 수가 달라진다.



태그 부자 김태그

ISBN : 13131
저자 : 김태그
출판사 : 태깅 컴패니

• TAG 디자인 사진/예술 소설 시집 여행/취미 외국어 웹
인공지능 자기계발 전공서적 프로그래밍 언어