Assignment/exercise 2: Database frontend workshop: PHP & SQL

Michał Pawlicki

From all 8 tasks, I was able to fully completed 6. First 4 tasks were done in the class, though I would like to focus on task no. 3. I used a different method then one that was presented in the class.

I made a single page that enables a user to manage all actions on categories: adding, deleting and changing the name.

For deleting and changing I created a drop-down menu where user can choose on which row action is going to be done. To achieve that I made 3 forms with submit buttons that contain different names in order to provide demanded action.

```php
if(!empty(filter_input(INPUT_POST, 'deleteSubmit'))){
    $deleteID = filter_input(INPUT_POST, 'selectID', FILTER_VALIDATE_INT);
    echo $deleteID;
    $sqlDelete = 'DELETE FROM category
            WHERE category_id =?';
    $stmtDelete = $con->prepare($sqlDelete);
    $stmtDelete->bind_param('i',$deleteID);
    $stmtDelete->execute();
    echo '<i>Succesfully deleted '.$deleteID.'</i>';
}


//CHANGE NAME

if(!empty(filter_input(INPUT_POST, 'changeSubmit'))){
    $updatedCatName = htmlspecialchars(filter_input(INPUT_POST, 'changedName'));
    $updateID = filter_input(INPUT_POST, 'updateID', FILTER_VALIDATE_INT);
    echo $updatedCatName.$updateID;
    $sqlUpdate='UPDATE category
                SET name=?
                WHERE category_id=?';
    $stmtUpdate = $con->prepare($sqlUpdate);
    $stmtUpdate->bind_param('si', $updatedCatName, $updateID);
    $stmtUpdate->execute();
    echo 'Correctet changed to '.$updatedCatName;


}

?>
```

Here is an example of deleting and changing the name. As it can be seen there is a simple condition that watches if the button has been clicked and then it executes the SQL action.

Task no. 4 was seemingly easy, but that was the first attempt during this exercise to use relations with other tables as it can be seen in the $sql statement down below.

```php
<ul>
<?php
    $categoryid = filter_input(INPUT_GET, 'categoryid', FILTER_VALIDATE_INT)
        or die('Missin correct category id');
    require_once('db_con.php');
    $sql ='
        SELECT f.film_id, f.title
        FROM film f, film_category fc
        WHERE f.film_id=fc.film_id
        AND fc.category_id=?';
    $stmt = $con->prepare($sql); //let's prepare to execute our sql, it's gonna be stored in $stmt
    $stmt->bind_param('i', $categoryid);
    $stmt->execute();
    $stmt->bind_result($filmID, $filmTitle);
    echo '<ul>';
    while($stmt->fetch()){?>
        <li><a href="filmdetails.php?filmid=<?php echo $filmID;?>"><?php echo $filmTitle;?></a></li>
    <?php } ?>
</ul>
</body>
```

Task no. 5 also included a cross reference to other tables and demanded more information to display.
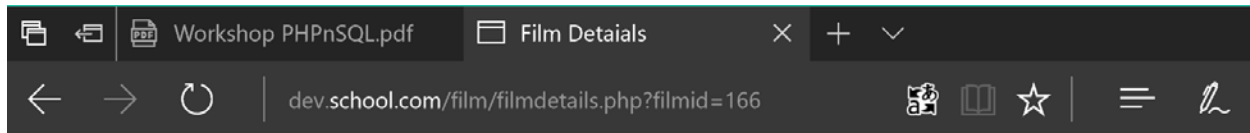
The ID is166were here

COLOR PHILADELPHIA (English)

Release year: 2006
Length: 149 minutes

A Thoughtful Panorama of a Car And a Crocodile who must Sink a Monkey in The Sahara Desert

This movie is in those categories:

- Classics

Starring:

- PENELOPE
- GROUCHO
- SALMA
- DARYL
- CHRISTOPHER
- OLYMPIA
- JOHN

Figure 1. screenshot of the results in task no.4

The code is actually split into three SELECTs: 1 for the basic information about the movie, 2 for categories and 3 for starring actors.

```php
require_once('db_con.php');

$filmId = filter_input(INPUT_GET, 'filmid', FILTER_VALIDATE_INT)
    or die('Missing or incorrect film ID');
echo 'The ID is'. $filmId;
$sqlSelectMovie ='SELECT f.title, f.description, f.release_year, f.length, l.name as lang
                  FROM film f, language l
                  WHERE f.film_id=?
                  AND f.language_id = l.language_id';
$stmtMovie=$con->prepare($sqlSelectMovie);
$stmtMovie->bind_param('i', $filmId);
$stmtMovie->execute();
$stmtMovie->bind_result($filmTitle, $filmDesc, $filmDate, $filmTime, $filmLang);
echo 'were here';
while($stmtMovie->fetch()){
    echo '<h1>'.$filmTitle.' ('.$filmLang.')</h1>';
    echo '<p>Release year: '.$filmDate.'<br />Length: '.$filmTime.' minutes </p>';
    echo '<p>'.$filmDesc.'</p>';
}?>

<h3>This movie is in those categories:</h3>
<ul>
<?php
    $sqlCat = 'SELECT c.category_id, c.name
    FROM film_category fc, category c
    WHERE film_id=?
    AND fc.category_id = c.category_id';
    $stmtCat = $con->prepare($sqlCat);
    $stmtCat->bind_param('i', $filmId);
    $stmtCat->execute();
    $stmtCat->bind_result($categoryId, $categoryName);
    //deletecategoryfromfilm.php
    while($stmtCat->fetch()) {
        echo '<li><a href="filmlist.php?categoryid='.$categoryId.'">'.$categoryName.'</a>';
        echo '</li>';
    }
?>
```

Figure 2. bakend - basic information on movies and categories

```php
<h3>Starring:</h3>
<ul>
<?php
    $sqlActors ='SELECT a.first_name, a.last_name, a.actor_id
                 FROM film_actor fa, actor a
                 WHERE film_id=?
                 AND fa.actor_id = a.actor_id';
    $stmtAc=$con->prepare($sqlActors);
    $stmtAc->bind_param('i',$filmId);
    $stmtAc->execute();
    $stmtAc->bind_result($actorName, $actorSurname, $actorId);
    while($stmtAc->fetch()){
        echo '<li><a href="actordetails.php?actorid='.$actorId.'">'.$actorName.'</a>';
        echo '</li>';
    }

?>
</ul>
```

Figure 3. backend - selecting actors

Here, also, every category and actor act as a link to a custom page on the topic.

Task no.6 was very similar to the previous one, so here is the screenshot on the backend.

```php
 8 ▼ <?php
 9         require_once('db_con.php');
10
11         $actorId = filter_input(INPUT_GET, 'actorid', FILTER_VALIDATE_INT)
12             or die('Missing or incorrect actor ID');
13         $sqlActor ='SELECT first_name, last_name
14                     FROM actor
15                     WHERE actor_id = ?';
16         $stmtActor=$con->prepare($sqlActor);
17         $stmtActor->bind_param('i', $actorId);
18         $stmtActor->execute();
19         $stmtActor->bind_result($actorName, $actorSurname);
20 ▼     while($stmtActor->fetch()){
21             echo '<h1>'.$actorName.' '.$actorSurname;
22         }
23 ?>
24 <h2>Starred movies:</h2>
25 ▼ <ul>
26 ▼    <?php
27         $sqlMovies = 'SELECT film.title, film.film_id
28                     from  film, actor, film_actor
29                     WHERE film_actor.actor_id = actor.actor_id
30                     AND film_actor.film_id = film.film_id
31                     AND actor.actor_id = ?';
32         $stmtMovie=$con->prepare($sqlMovies);
33         $stmtMovie->bind_param('i', $actorId);
34         $stmtMovie->execute();
35         $stmtMovie->bind_result($movieTitle, $movieId);
36 ▼     while($stmtMovie->fetch()){
37             echo "<li><a href='filmdetails.php?filmid=".$movieId."'>".$movieTitle.'</a></li>';
38         }
39     ?>
40
```

And the screenshot below represents sample output