

CPET 499/ITC 250 Web Systems

Chapter 11 Working with Databases Part 2 of 3

Text Book:

* Fundamentals of Web Development, 2015, by Randy Connolly and Ricardo Hoar, published by Pearson

Paul I-Hai Lin, Professor of ECET
<http://www.etcs.ipfw.edu/~lin>

Topics

- PHP API Fetch Functions
- PHP and MySQL Tasks
 - Making MySQL connection and closing connection
 - Display a List of Links
 - Search and Results Page
 - Editing a Record
 - Saving and Displaying Raw Files in the Database
 - Displaying BLOBs from the Database
 - Using Transactions

More on PHP mysqli Fetch Functions

CPET 499/ITC 250 Web Systems, Paul I. Lin

3

PHP MySQL Fetching Functions

- **mysqli_fetch_all()**: Fetches all result rows as an associate array, a numeric array, or both
 - <http://php.net/manual/en/mysqli-result.fetch-all.php>
- **mysqli_fetch_array()**: Fetches a result row as an associate array, a numeric array, or both
 - <http://php.net/manual/en/mysqli-result.fetch-array.php>
- **mysqli_fetch_assoc()**: Fetches a result row as an associate array
 - <http://php.net/manual/en/mysqli-result.fetch-assoc.php>
- **mysqli_fetch_field()**: Returns the definition of one column of a result set as an object. Call this function repeatedly to retrieve information about all columns in the result set.
 - <http://php.net/manual/en/mysqli-result.fetch-field.php>

CPET 499/ITC 250 Web Systems, Paul I. Lin

4

PHP MySQL: Procedural Style Fetching Functions

- **mysqli_fetch_fields():** Returns an array of objects which contains field definition information or FALSE if no field information is available
 - <http://php.net/manual/en/mysqli-result.fetch-fields.php>
- **mysqli_fetch_object():** Returns the current row of a result as an object
 - <http://php.net/manual/en/mysqli-result.fetch-object.php>
- **mysqli_fetch_row():** Fetch one row of data from the result set as a numeric array
 - <http://php.net/manual/en/mysqli-result.fetch-row.php>

Sample Database Techniques

- Database Display Tasks in PHP
 - Display a List of Links
- Search and Result Page
- Editing a Record
- Saving and Displaying Raw Files in the Database

Display a List of Links LAB 11 Exercise

```
$sql = "SELECT * FROM Categories ORDER BY  
CategoryName";  
$results = $pdo -> query($sql);  
  
while ($row = $results -> fetch() {  
    echo '<li>';  
    echo '<a href=\'list.php?category=\' .  
        row[\'ID\'] . \'>';  
    echo $row[\'CategoryName\'];  
    echo '</a.\>';  
    echo '</li>';  
}
```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

7

Listing 11.28 Alternating list of links example

```
<?php  
//Listing 11.28 Alternate list of links example  
?>  
<ul>  
<?php  
$result = getResult(); // some function that returns the result set  
while ($row = $result->fetch()) {  
?>  
    <li>  
        <a href="list.php?category=<?php echo $row['ID']; ?>">  
            <?php echo $row['CategoryName']; ?>  
        </a>  
    </li>  
<?php } ?>  
</ul>
```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

8

Markup List Generation

```
<ul>
<li><a href="list.php?category=7">Business</a></li>
<li><a href="list.php?category=2">Computer Science</a></li>
<li><a href="list.php?category=3">Economics</a></li>
<li><a href="list.php?category=9">Engineering</a></li>
<li><a href="list.php?category=4">English</a></li>
<li><a href="list.php?category=6">Mathematics</a></li>
<li><a href="list.php?category=8">Statistics</a></li>
<li><a href="list.php?category=5">Student Success</a></li>
</ul>
```

Database Tasks in PHP: Search and Result Page

- Database Task in PHP:
- A textBox in a HTML form
 - textSearch = 'user entered string'
 - A submit button
- Search/Filter the Book table: "Title" field
- Display matched records/results
- SQL
 - operator "LIKE"
 - wildcard character "%"

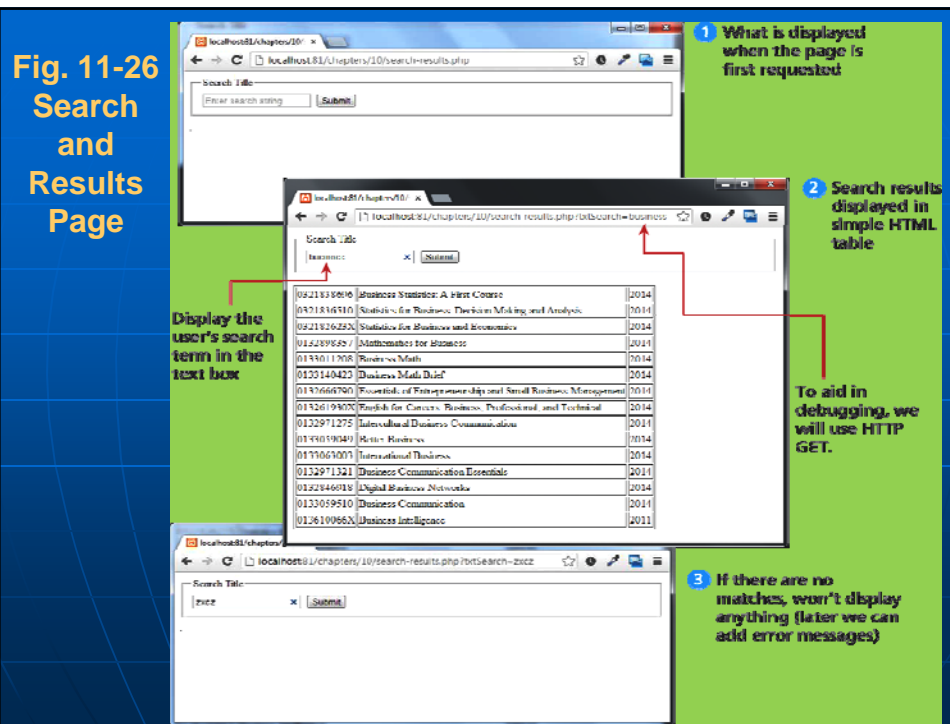
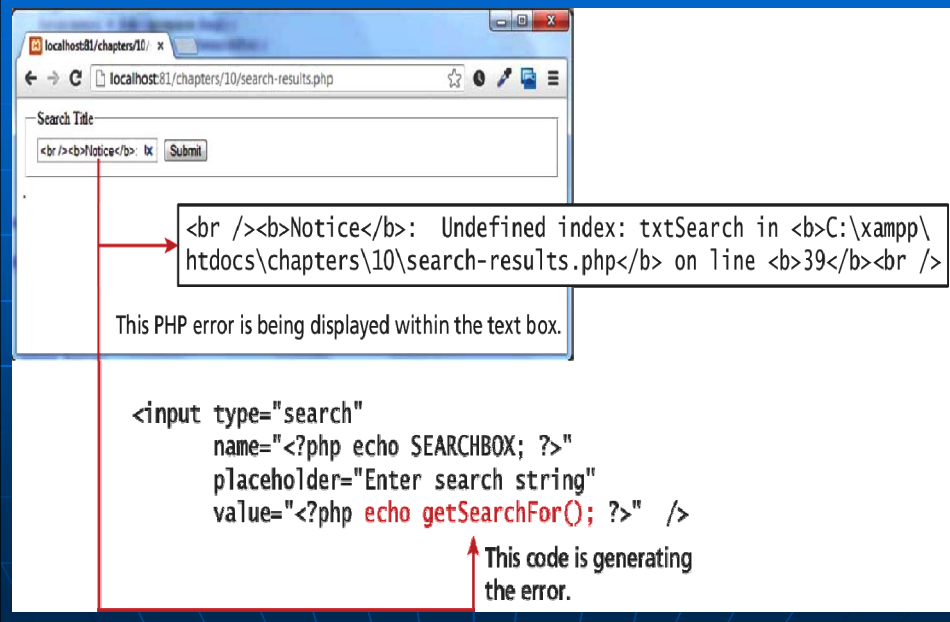


Figure 11.27 Problems with Listing 11.29



Listing 11.29 Partial Solution to search results page (search-results.php)

```
<?php//Listing 11.29 Partial solution to search results page
(search-results.php)
// defines a constant for query string parameter
namedef('SEARCHBOX', 'txtSearch');
// define a function to return the value of the search parameter
function getSearchFor(){
// this function is missing something ... do you know what it is?
return $_GET[SEARCHBOX];
}
function getDB()
{ //...
$pdo = new PDO($connString,$user,$pass);
$pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
return $pdo;}
```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

13

Listing 11.29 Partial Solution to search results page (search-results.php)

```
function getResults(){
    try {
        $db = getDB();
        // add SQL wildcard characters to search term
        $searchFor = '%' . getSearchFor() . '%';
        $sql = "SELECT * FROM Books WHERE Title Like ?";
        $statement = $db->prepare($sql);
        $statement->bindValue(1, $searchFor);
        $statement->execute(); return $statement;
    }
    catch (PDOException $e) {
        die($e->getMessage());
    }
}
}?>
```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

14

Listing 11.29 Partial Solution to search results page (search-results.php)

```
<html>
<body>
<form method="get" action="search-results.php" >
  <fieldset>
    <legend>Search Title</legend>
    <input type="search" name="<?php echo SEARCHBOX; ?>"
      placeholder="Enter search string" value="<?php echo
        getSearchFor(); ?>" />
    <input type="submit" />
  </fieldset>
</form> <table border="1">
```

Listing 11.29 Partial Solution to search results page (search-results.php)

```
<?php
if (! empty($_GET[SEARCHBOX]) && $result = getResults() ) {
  while ($row = $result->fetch()) {    ?>
    <tr>
      <td><?php echo $row['ISBN10']; ?></td>
      <td><?php echo $row['Title']; ?></td>
      <td><?php echo $row['CopyrightYear']; ?></td>
    </tr>    <?php
  } } ?>
</table>
</body>
</html>
```


Listing 11.30 Solution to search results page problem

```
<?php
//Listing 11.30 Solution to search results page problem
function getSearchFor()
{
    $value = "";
    if (isset($_GET[SEARCHBOX]))
    {
        $value = $_GET[SEARCHBOX];
    }
    return $value;
}
?>
```

Database Tasks in PHP: Editing a Record

■ A Record Editor

- The form must be populated with existing record data when the page is first displayed
- The page need logic to both save and retrieve data

Figure 11.28 Program Flow in record editor

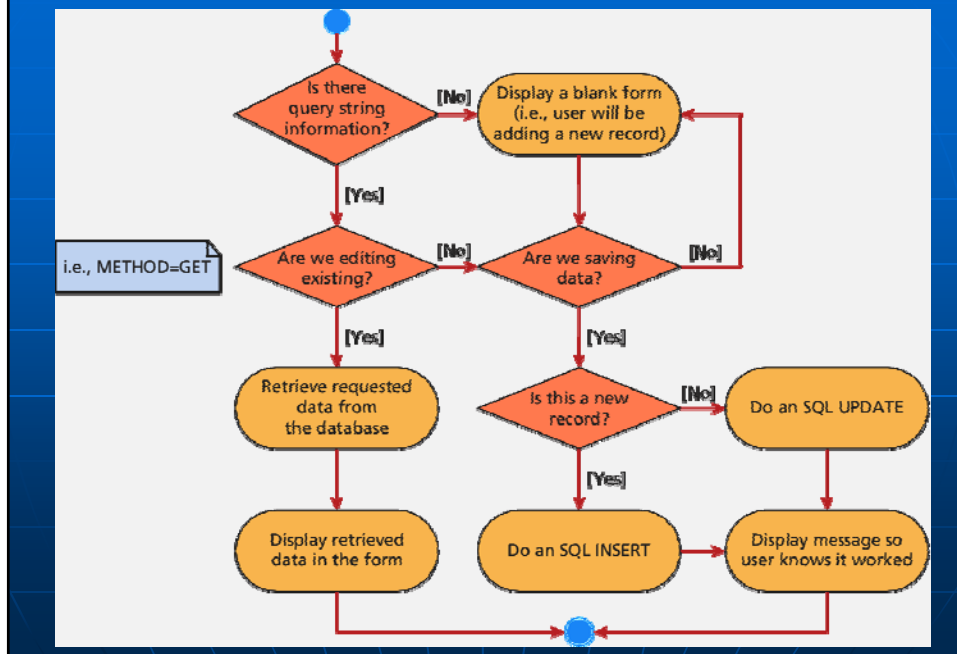
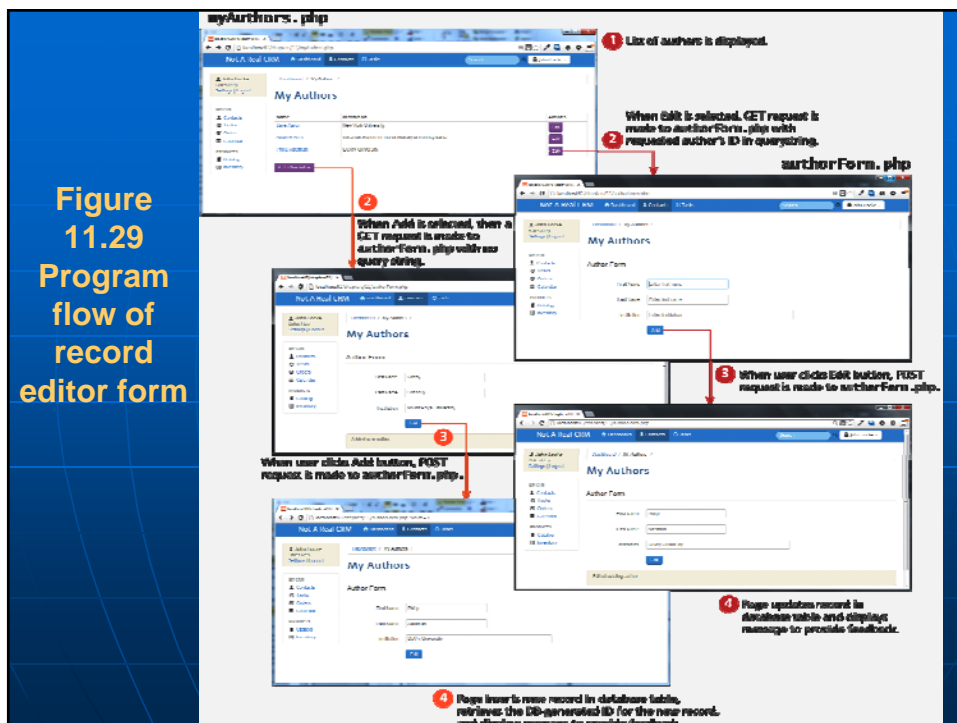


Figure 11.29 Program flow of record editor form



Listing 11.31 Author Class

```
<?php // Listing 11.31 Author class
class Author {
    public $id = "";
    public $firstName = "";
    public $lastName = "";
    public $institution = "";
    function __construct($id,$first,$last,$institute) {
        $this->id = $id;
        $this->firstName = $first;
        $this->lastName = $last;
        $this->institution = $institute; }
    // Returns true if this is a new author, false otherwise
    function isNew() {
        if (empty($this->id) )
            return true;
        else return false;
    }
}
```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

21

Listing 11.32 processAuthorFormInfo() function

```
<?php//Listing 11.32 processAuthorFormInfo() function
function processAuthorFormInfo($pdo) {
    // first let us see if there is any query string information
    // ... if not return empty author object
    if (!isThereQueryStringInfo() ) {
        return new Author("", "", "", ""); }
    // are we editing an existing author ...
    if ( areEditingExisting() ) {
        // since request method is GET, then this is either request for
        // inserting new or a request to edit if id attribute
        // NOTE: we are assuming ID in query string is ok
        // (should actually test it in real site)
        $which = $_GET['which'];
        // retrieve data from database
        return retrieveAuthor($pdo, $which); }
}
```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

22

Listing 11.32 processAuthorFormInfo() function

```
// ... or are we saving an author
if ( areSaving() ) {
    // if here then saving a record
    // we are going to use the existence of an ID querystring to
    // determine whether we should be inserting or updating
    $id = "";
    if ( isset($_POST['id']) ) {
        $id = $_POST['id'];
    }
    $author = saveAuthor( $pdo, $id, $_POST['firstname'],
                        $_POST['lastname'], $_POST['institution'] );
    return $author;
}
?>
```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

23

Listing 11.33 Helper function for listing 11.32

```
<?php
//Listing 11.33 Helper functions for Listing 11.32
/*Checks if there is any query string information passed in GET or
POST*/
function isThereQueryStringInfo() {
    if ( areEditingExisting() ) {
        return true;
    }
    if ( areSaving() ) {
        return true;
    }
    return false;}

```

CPET 499/ITC 250 Web Systems, Paul I.
Lin

24

Listing 11.33 Helper function for listing 11.32

```
/*Checks if query string info tells us whether we are editing existing
author*/
function areEditingExisting() {
    if ($_SERVER['REQUEST_METHOD'] == 'GET' &&
        isset($_GET['which'])) { return true; }}

/*Checks if query string info tells us whether we are saving author
info*/
function areSaving() {
    if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_
        POST['firstname']) &&
        isset($_POST['lastname']) ) {
        return true; }}
```

Listing 11.33 Helper function for listing 11.32

```
/*Actually perform the database insert or update*/
function saveAuthor($pdo, $id, $first, $last, $institute)
{
    $GLOBALS['updateStatus'] = "";
    $author = new Author($id, $first, $last, $institute);
    // set up sql statement and page's message
    if ( $author->isNew() ) {
        $sql = "INSERT INTO authors (FirstName,LastName,Institution)
VALUES (:first,:last,:institute)";
        $GLOBALS['saveMessage'] = 'Added new '; }
    else {
        $sql = "UPDATE authors SET FirstName=:first, LastName=:last,
Institution=:institute WHERE ID=:id";
```

Listing 11.33 Helper function for listing 11.32

```
$GLOBALS['saveMessage'] = 'Edited existing ';  
}  
// setup the parameters for the query  
$statement = $pdo->prepare($sql);  
$statement->bindValue(':first', $first);  
$statement->bindValue(':last', $last);  
$statement->bindValue(':institute', $institute);  
if ( ! $author->isNew() ) $statement->bindValue(':id', $id);  
// execute the query  
$statement->execute();  
// retrieve auto generated id if this was an insert and update  
// author object  
if ( $author->isNew() ) {  
    $author->id = $pdo->lastInsertId();  
}  
return $author;}
```

CPET 499/ITC 250 Web Systems, Paul I. Lin

27

Listing 11.33 Helper function for listing 11.32

```
/*Retrieve a populated author from the database*/  
function retrieveAuthor($pdo, $id)  
{  
    $sql = "SELECT * FROM Authors WHERE ID=:id";  
    $statement = $pdo->prepare($sql);  
    $statement->bindValue(':id', $id);  
    $statement->execute();  
    $row = $statement -> fetch(PDO::FETCH_ASSOC);  
    return new Author($row['ID'], $row['FirstName'], row['LastName'],  
        $row['Institution']);  
}  
?>
```

CPET 499/ITC 250 Web Systems, Paul I. Lin

28

Listing 11.34 authorForm.php page

```
<?php
//Listing 11.34 authorForm.php page
// initialize page globals
require_once('includes/config-books.inc.php');
require_once('includes/Author.class.php');
// class name for hiding a <div>
$GLOBALS['updateStatus'] = 'hide';
// the message to be displayed after saving
$GLOBALS['saveMessage'] = '';
```

CPET 499/ITC 250 Web Systems, Paul I. Lin

29

Listing 11.34 authorForm.php page

```
try {
    // set up the PDO connection to database
    $pdo = new PDO(DBCONNECTION,DBUSER,DBPASS);
    $pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    // perform the algorithm and return populated Author object
    $author = processAuthorFormInfo($pdo);
    // change form Submit button text based on author object
    if ( $author->isNew() ) { $buttonText = 'Add'; }
    else { $buttonText = 'Edit'; }
}
catch (PDOException $e) {
    die( $e->getMessage() );
}
//...
?>
```

CPET 499/ITC 250 Web Systems, Paul I. Lin

30

Listing 11.34 authorForm.php page

```
<!DOCTYPE html>
<html>
<head lang="en">
...
<form class="form-horizontal" method="post"
action="authorForm.php">
<fieldset>
<legend>Author Form</legend>
<input type="hidden" name="id" value="<?php echo $author->id
?>" />
<label>First Name</label>
<input type="text" name="firstname"
placeholder="Enter first name"
value="<?php echo $author->firstName; ?>">
<label>Last Name</label>
```

CPET 499/ITC 250 Web Systems, Paul I. Lin

31

Listing 11.31 Author Class

```
<input type="text" name="lastname"
placeholder="Enter last name"
value="<?php echo $author->lastName; ?>">
<label>Institution</label>
<input type="text" name="institution"
placeholder="Enter Institution"
value="<?php echo $author->institution; ?>">
<button type="submit" >
<?php echo $buttonText; ?>
</button>
</fieldset>
</form>
<div class="alert alert-info <?php echo GLOBALS['updateStatus']; ?>">
<p> <?php echo $GLOBALS['saveMessage']; ?> author</p>
</div>
```

-*

3-as

er46453lkjh0 sdfghjkl QWER456P

32

Saving and Displaying Raw Files in the Database

- Allows a user to upload an image file
- Saves it within a BLOB field
- Examples
 - Travel Photo database

CPET 499/ITC 250 Web Systems, Paul I. Lin

33

Figure 11.30 Storing file location in the database

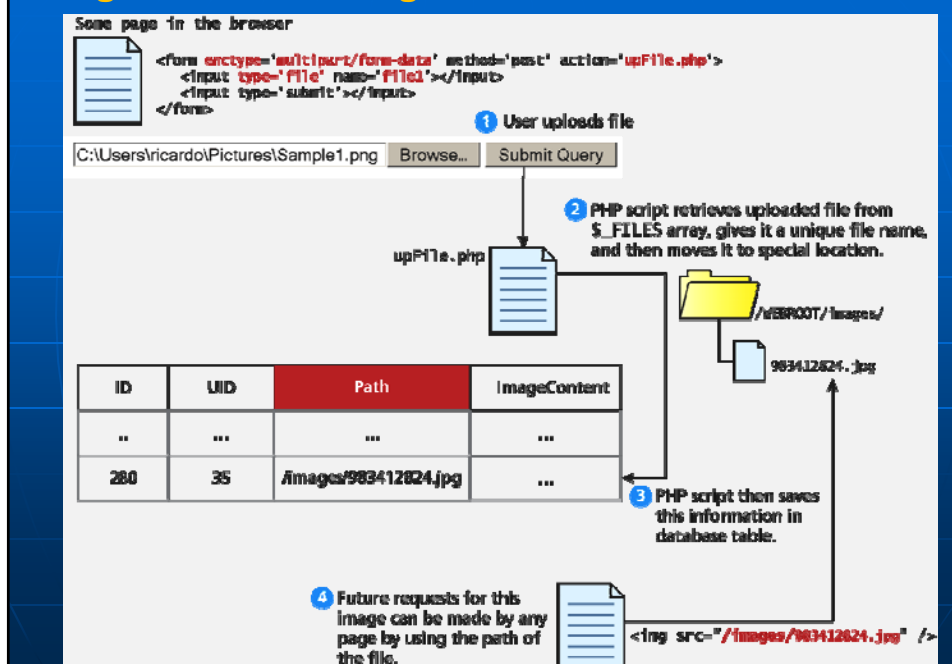
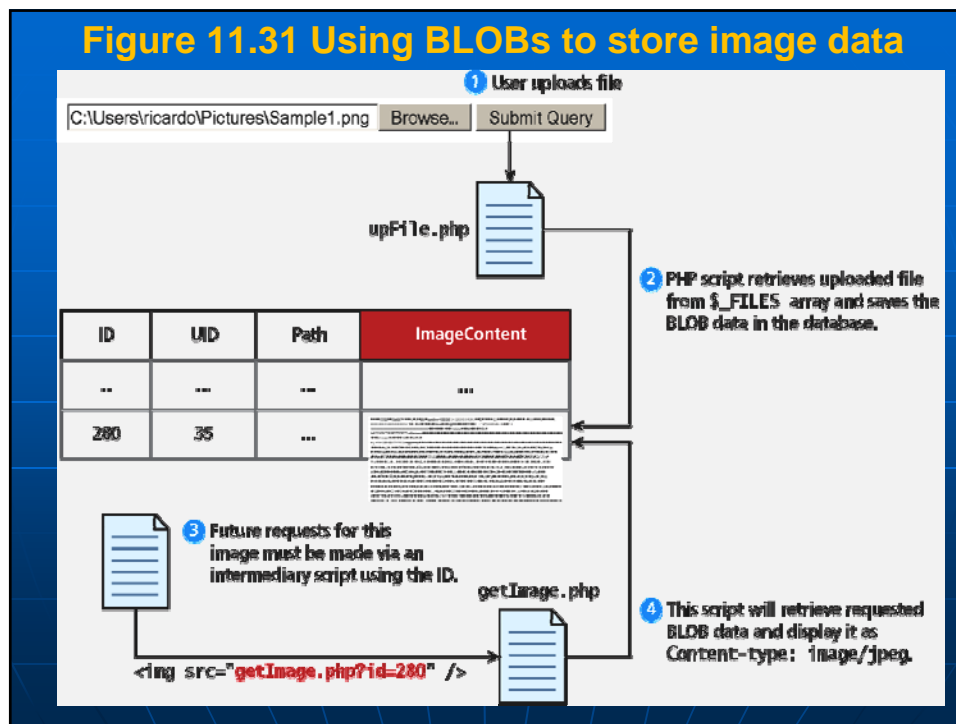


Figure 11.31 Using BLOBs to store image data



Summary & Conclusion

Q & A