Kiểm tra/nhắc lại bài học trước

CÔNG NGHỆ WEB AN TOÀN

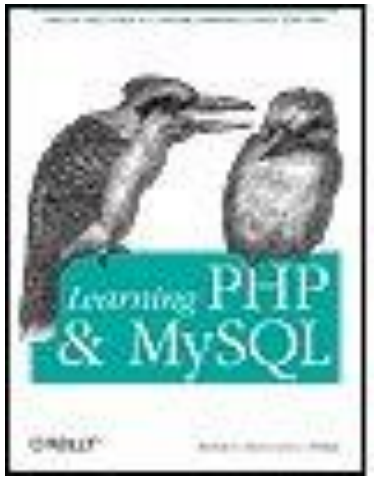
Bài 3-2. CƠ SỞ DỮ LIỆU

Mục tiêu buổi học

1. Trình bày được mô hình ba tầng
2. Trình bày được về quá trình kết nối và sử dụng CSDL trong ứng dụng web
3. Tạo được một trang web động sử dụng PHP và MySQL

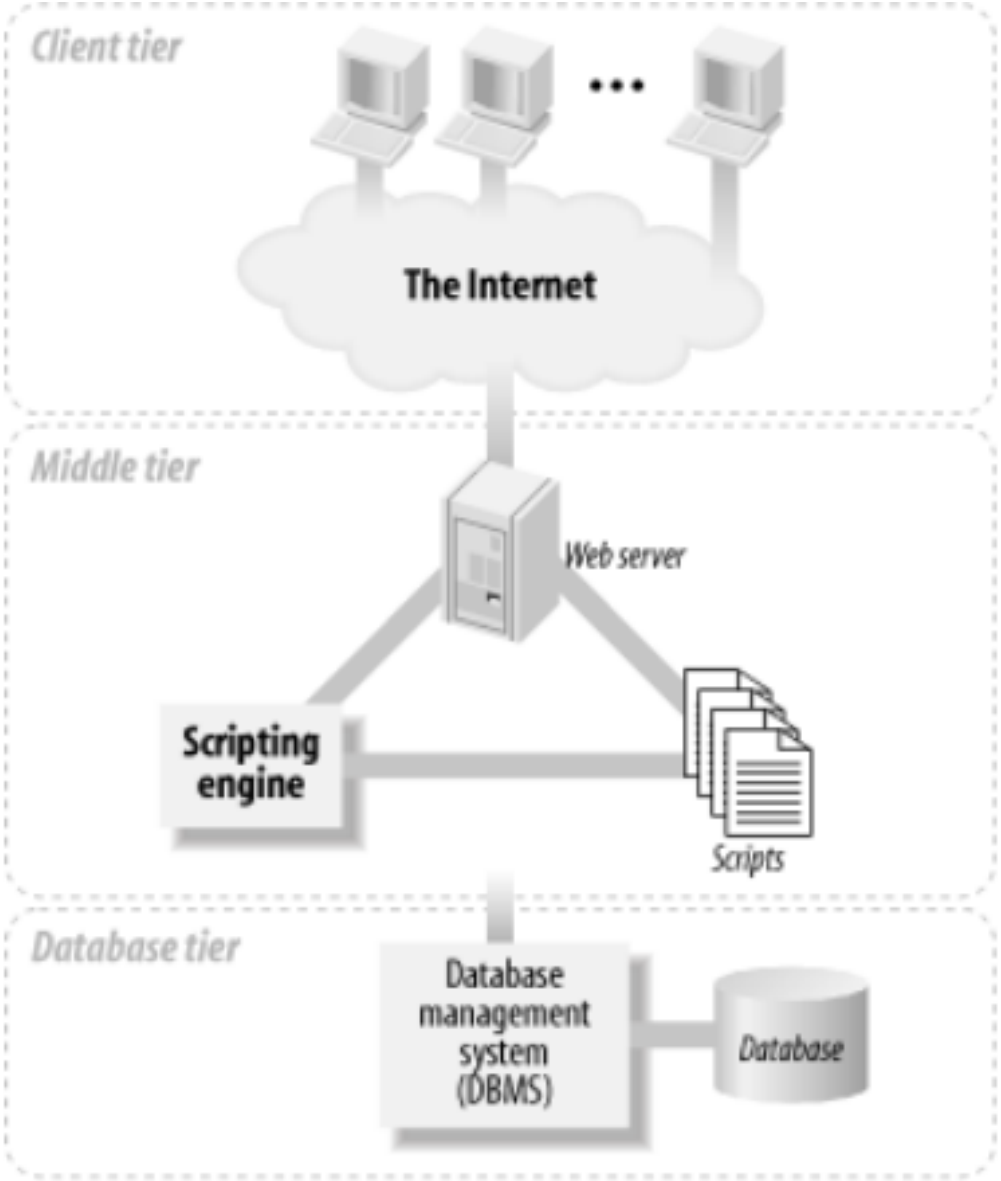
Tài liệu tham khảo

1. https://www.w3schools.com/sql/default.asp
2. https://www.w3schools.com/mysql/default.asp
3. Michele Davis & Jon Phillips, “Learning PHP and MySQL”
4. Hugh E. Williams & David Lane, ” Web Database Applications with PHP & MySQL”
5. Lê Đình Thanh, Bài giảng “Phát triển ứng dụng Web”, Trường Đại học Công nghệ, ĐHQGHN.
6. Bộ bài tập học phần.



Mô hình ba tầng

- Tầng khách: trình diễn và tương tác với người dùng
- Tầng giữa: thực hiện các logic của ứng dụng
- Tầng CSDL: bao gồm hệ quản trị CSDL, CSDL của ứng dụng



1. Cơ sở dữ liệu
2. Lưu trạng thái

1. Cơ sở dữ liệu
2. Lưu trạng thái

Tổng quan

- Các bước chính
  1. Tạo kết nối đến máy chủ cơ sở dữ liệu
  2. Lựa chọn cơ sở dữ liệu để làm việc
  3. Xây dựng các truy vấn, cập nhật và thực hiện truy vấn, cập nhật
  4. Xử lý dữ liệu kết quả trả về khi thực hiện các truy vấn, cập nhật
  5. Đóng kết nối đến máy chủ cơ sở dữ liệu
- Hệ quản trị CSDL
  - MySQL, PostgreSQL, Oracle, MongoDB, MS SQL, …
- Thư viện thao tác CSDL cho PHP
  - PDO (khuyến cáo), mysqli, mysql (cũ)

1. Kết nối cơ sở dữ liệu
2. Thực thi SQL
3. Truy vấn với dữ liệu người dùng
4. Kết hợp mã hiển thị và thực hiện truy vấn trong một trang
5. Một số lưu ý

Kết nối cơ sở dữ liệu

- MySQLi (hướng đối tượng)
- MySQLi (thủ tục)

Cài đặt MySQLi: http://php.net/manual/en/mysqli.installation.php

- PDO

Cài đặt PDO: http://php.net/manual/en/pdo.installation.php

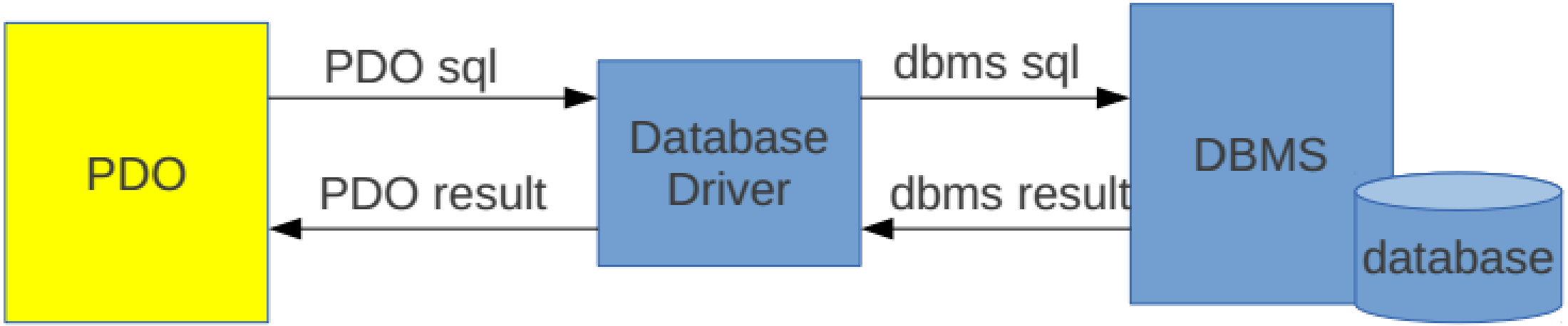
MySQLi (hướng đối tượng)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

MySQLi (thủ tục)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn=mysqli_connect($servername,$username,$password);
// Check connection
if (!$conn) {
  die("Connection failed: ". mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Nguyên lý hoạt động PDO...



Cần cài driver cho hệ QTCSDL được sử dụng

PDO…

- `$db=new PDO($dsn,$username,$password,[$options]);`
- Tên nguồn dữ liệu (dsn - database source name) khác nhau với các hệ QTCSDL
  - MySQL: "`mysql:host=...; dbname=...;`"
  - PostgressSQL: "`pgsql:host=...; dbname=...;`"
  - MS SQL: "`sqlsrv:Server=...;Database=...`"
  - …

PDO

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
try {
  $conn = new PDO("mysql:host=$servername;dbname=myDB",
$username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
  echo "Connected successfully";
} catch(PDOException $e) {
  echo "Connection failed: " . $e->getMessage();
}
?>
```

Đóng kết nối

- MySQLi Object-Oriented:

```
$conn->close();
```

- MySQLi Procedural:

```
mysqli_close($conn);
```

- PDO:

```
$conn = null;
```

1. Kết nối cơ sở dữ liệu
2. **Thực thi SQL**
3. Truy vấn với dữ liệu người dùng
4. Kết hợp mã hiển thị và thực hiện truy vấn trong một trang
5. Một số lưu ý

Thực thi SQL

- MySQLi hướng đối tượng
- MySQLi thủ tục

https://www.w3schools.com/php/func_mysqli_query.asp

- PDO
- Tham khảo:

  https://www.w3schools.com/php/php_ref_mysqli.asp

Thực thi SQL - MySQLi hướng đối tượng

- **Thực thi SQL:** `$ret = $conn->query($sql);`

  •Nếu thực thi thành công, hàm trả về true, ngược lại hàm trả về false và nguyên nhân lỗi được MySQL quản lý. Đọc nguyên nhân lỗi bằng hàm `$conn->error`

  •Với các lệnh update, delete, insert, MySQL quản lý số bản ghi chịu tác động. Lấy số bản ghi chịu tác động bằng hàm

  `$conn-> affected_rows`

  •Với lệnh select, hàm trả về recordset lưu kết quả truy vấn. Hàm `mysql_num_rows($recordset)` trả về số bản ghi nhận được.

- `$sql`: Lệnh insert, update, delete

Thực thi SQL - MySQLi thủ tục

- **Thực thi SQL:** `$ret = mysql_query($conn,$sql);`
  - Nếu thực thi thành công, hàm trả về true, ngược lại hàm trả về false và nguyên nhân lỗi được MySQL quản lý. Đọc nguyên nhân lỗi bằng hàm `mysql_error()`
  - Với các lệnh update, delete, insert, MySQL quản lý số bản ghi chịu tác động. Lấy số bản ghi chịu tác động bằng hàm `mysql_affected_rows()`
- **Cập nhật CSDL:** `$count = mysql_query($conn,$sql);`
  - `$sql`: Lệnh insert, update, delete
  - `$count`: Số bản ghi được cập nhật

Thực thi SQL – PDO…

- **Thực thi SQL: `$conn->exec($sql);`**

```
try {
  $conn = new PDO("mysql:host=$servername",
$username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
  $sql = "CREATE DATABASE myDBPDO";
  // use exec() because no results are returned
  $conn->exec($sql);
  echo "Database created successfully<br>";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
}
```

Thực thi SQL – PDO…

- Lấy id (tự tăng) của bản ghi vừa được thêm

```
$db->lastInsertId();
```

- Thực thi insert, update, delete với lệnh chuẩn bị trước

```
$stmt = $db->prepare("DELETE FROM table WHERE
id=:id and name LIKE ?");
$stmt->bindValue(':id', $id);
$stmt->bindValue(2, "%$search%");
$stmt->execute();
$affected_rows = $stmt->rowCount();
```

- Thực hiện truy vấn

```
$stmt = $db->query('SELECT * FROM ...');
```

Thực thi SQL – PDO…

- Duyệt các bản ghi

```
while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    echo $row['field1'].' '.$row['field2'];
}
```

- Đếm số bản ghi

```
$row_count = $stmt->rowCount();
```

Giao tác

```
try {
    $db->beginTransaction();
    $db->exec("SOME QUERY");
    $stmt = $db->prepare("SOME OTHER QUERY?");
    $stmt->execute(array($value));
    $stmt = $db->prepare("YET ANOTHER QUERY??");
    $stmt->execute(array($value2, $value3));
    $db->commit();
} catch(PDOException $ex) {
    $db->rollBack();
    echo $ex->getMessage();
}
```

1. Kết nối cơ sở dữ liệu
2. Thực thi SQL
3. **Truy vấn với dữ liệu người dùng**
4. Kết hợp mã hiển thị và thực hiện truy vấn trong một trang
5. Một số lưu ý

Truy vấn với dữ liệu người dùng…

- Các phương pháp nhập liệu
  - Nhập tham số và giá trị ngay sau URL
  - Bấm vào liên kết đến URL có các tham số kèm theo
  - Đệ trình form theo phương thức GET
  - Dữ liệu được đưa vào URL trong chuỗi truy vấn
  - Ví dụ http://localhost/example.php?regionName=Riverland&countryName=Jamaica
  - Đệ trình form theo phương thức POST
  - Dữ liệu được đưa vào thân của gói HTTP Request

Truy vấn với dữ liệu người dùng…

- Nhận dữ liệu được gửi theo phương thức GET

```
$bien = $_GET[' tenThamso '];
```

- Nhận dữ liệu được gửi theo phương thức POST

```
$bien = $_POST[' tenThamso '];
```

- Tạo truy vấn theo dữ liệu nhập

```
$sql="select…from…where thuoctinh=\'"+clean($bien)+ "\'";
```

hoặc

```
$stmt = $db->prepare("SELECT … FROM …WHERE thuoctinh=?");
$stmt->bindValue(1, $bien);
$stmt->execute();
```

Truy vấn với dữ liệu người dùng

❑ Xuất HTML theo dữ liệu truy vấn

```
while ($row = mysql_fetch_array($result))
{
    echo “Thẻ HTML mở”;
    echo $row["thuoctinh“];
    echo “Thẻ HTML đóng”;
}
```

Hoặc

```
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
  echo “Thẻ HTML mở”;
  echo htmlentities($row["thuoctinh“]);
  echo “Thẻ HTML đóng”;
}
```

❑ Ví dụ

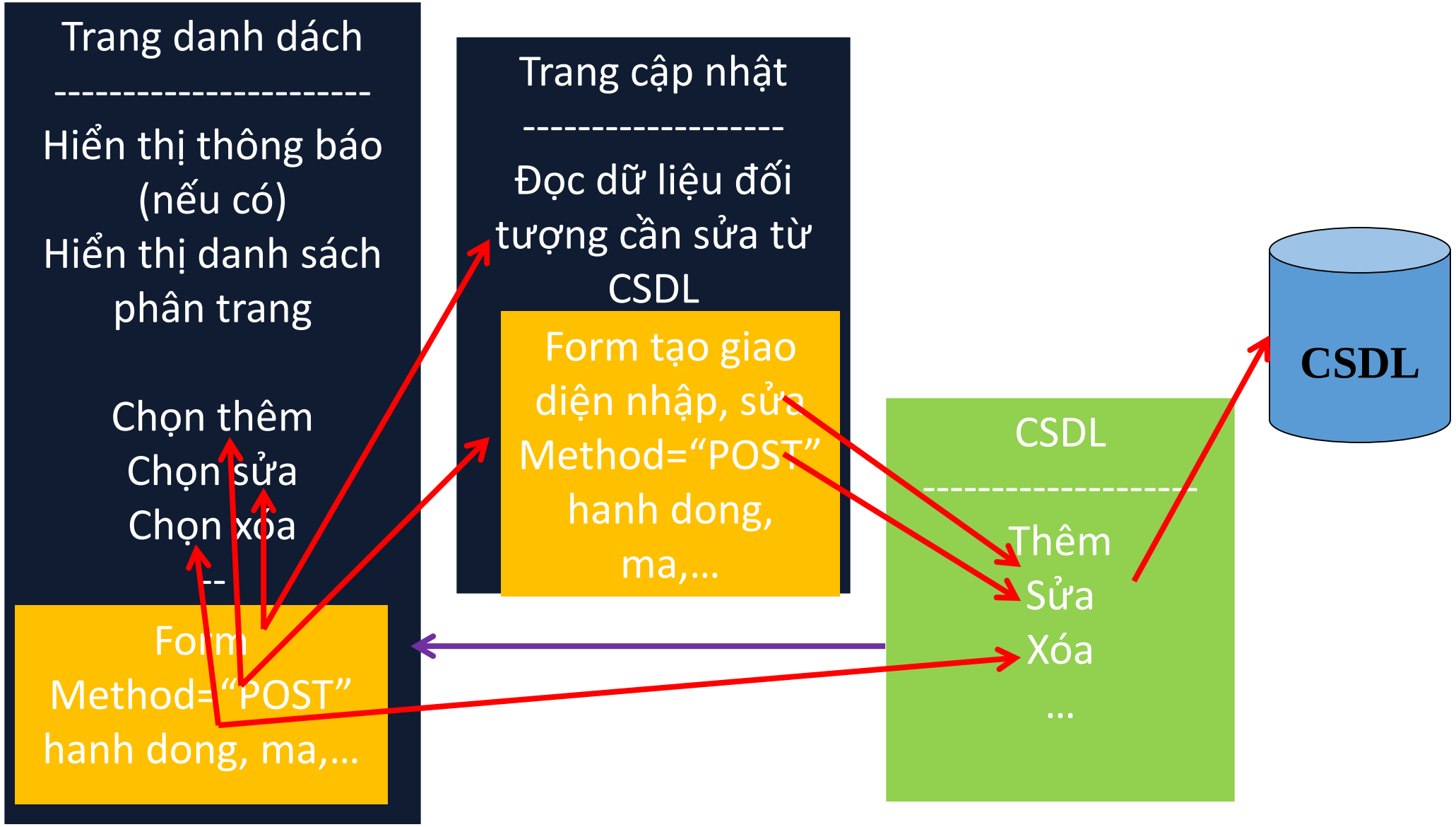
```
echo "<tr><td>"."<a href=\"example.php?qty=1&wineId="
.$row["wine_id"]."\">Add a bottle to the shopping cart</a>" .
"</td></tr>";
```

1. Kết nối cơ sở dữ liệu
2. Thực thi SQL
3. Truy vấn với dữ liệu người dùng
4. **Kết hợp mã hiển thị và thực hiện truy vấn trong một trang**
5. Một số lưu ý

Kết hợp mã hiển thị và thực hiện truy vấn trong một trang

```
if (!isset($thamso)) {
    //Tạo form nhập
} else {
    //Xử lý an ninh dữ liệu nhập
    //Thực hiện truy vấn và xuất HTML theo dữ liệu truy vấn
}
```

Ví dụ - Kiến trúc…



Ví dụ Phân trang hiển thị…

❑ Mô tả

Trang đầu Trang trước 1 2 3 4 5 Trang sau Trang cuối

Các tham số

- Số dòng trên một trang: rowsPerPage
- Trang hiện tại: currentPage

❑ Nhận tham số và kiểm tra

```
$currentPage = 0;
if (!empty($_GET[“currentPage ”]) {
    $s = clean($_GET[“currentPage ”], 4);
    if (is_numeric($s))
        $currentPage = intval($s);
}
```

Ví dụ Phân trang hiển thị …

- Hiển thị các bản ghi ứng với trang hiện tại

```
$firstRow = $currentPage*$rowsPerPage;
if (!mysql_data_seek($result, $firstRow))   showerror( );
for ($i=0;(($i< $rowsPerPage)&&($row=mysql_fetch_array($result)
));$i++) {
    echo "<tr><td>";
    echo $row["thuoctinh"];
    echo "</td></tr>";
}
```

- Thêm liên kết *Trang trước*

```
if ($currentPage == 0) echo "Trang trước";
else {
   echo "<a href = \"?currentPage=";
   echo (currentPage-1);
   echo "\">";
   echo "Trang trước";
   echo "</a>";
}
```

Ví dụ Phân trang hiển thị…

- Thêm liên kết *Trang sau*

```
$numPage = floor(mysql_num_rows($result)/$rowsPerPage);
if (mysql_num_rows($result) % $rowsPerPage != 0)     $numPage++;
if ($currentPage == $numPage-1) echo "Trang sau";
else {
    echo "<a href = \"?currentPage=";
    echo ($currentPage+1);
    echo "\">";
    echo "Trang sau";
    echo "</a>";
}
```

- Thêm liên kết *Trang đầu*

```
if ($currentPage == 0) echo "Trang đầu";
else {
    echo "<a href = \"?currentPage=0\">";
    echo "Trang đầu";
    echo "</a>";
}
```

Ví dụ Phân trang hiển thị…

- Thêm liên kết *Trang cuối*

```
$numPage = floor(mysql_num_rows($result)/$rowsPerPage);
if (mysql_num_rows($result) % $rowsPerPage != 0)
$numPage++;
if ($currentPage == $numPage-1) echo "Trang cuối";
else {
    echo "<a href = \"?currentPage=";
    echo ($numPage-1);
    echo "\">";
    echo "Trang cuối";
    echo "</a>";
}
```

Ví dụ Phân trang hiển thị

❑ Thêm liên kết số trang

```
$numPage = floor(mysql_num_rows($result)/$rowsPerPage);
if (mysql_num_rows($result) % $rowsPerPage != 0)
$numPage++;
for ($i = 0; $i < $numPage; $i++)
    if ($i == $currentPage) echo ($i+1);
    else {
        echo "<a href = \"?currentPage=";
        echo $i;
        echo "\">";
        echo (i+1);
        echo "</a> ";
    }
```

Ví dụ Mẫu trang tạo form cập nhật

```
//Nếu là cập nhật thì load giá trị bản ghi từ CSDL để đưa vào form
//Tạo form nhập
//Kiểm tra hợp thức phía client
//Yêu cầu server kiểm tra hợp thức phía server và kiểm tra trùng mã (nếu cần)
```

Ví dụ Mẫu trang cập nhật CSDL

```
$note = “”;
$input = clean($input);
$insertQuery= “lệnh sql được xây dựng theo $input”;
if ((mysql_query ($insertQuery, $connection)) &&
(($c = mysql_affected_rows( )) > 0)) {
    $note = “Thông báo đã thêm/cập nhật/xóa được bao
nhiêu bản ghi”;
} else {
    $note= “Thông báo không thêm/cập nhật/xóa được”;
}
header("Location: list.php? note=“.$note);
```

Trang hiển thị DS được chỉnh sửa để thông báo kết quả cập nhật

```
if (!empty($_GET[“note”]) {
     echo $_GET[“note”];
}
//mã xử lý còn lại để hiển thị danh sách theo trang
```

1. Kết nối cơ sở dữ liệu
2. Thực thi SQL
3. Truy vấn với dữ liệu người dùng
4. Kết hợp mã hiển thị và thực hiện truy vấn trong một trang
5. **Một số lưu ý**

Upload tệp và lưu vào CSDL

- Tạo form upload tệp

```
<form enctype="multipart/form-data“ action=“page.php“ method="post">
   <input name="userfile" type="file">
   <br><input type="submit">
</form>
```

- Nhận tệp:

```
// Tệp đã được upload?
if (is_uploaded_file($userfile)) {
   // Mở tệp
   $file = fopen($userfile, "r");
   // Đọc nội dung
   $fileContents = fread($file, filesize($userfile));
   //Xử lý các ký tự đặc biệt bằng cách thêm \ trước chúng
   $fileContents = AddSlashes($fileContents);
} else $fileContents = NULL;
```

- Lưu nội dung tệp vào CSDL:

```
$insertQuery = "INSERT INTO TableName
VALUES (..., \"".$fileContents."\")";
```

Đọc và hiển thị tệp ảnh đã lưu vào CSDL

- Tệp imgdisp.php hiển thị ảnh

```
$data = @ mysql_fetch_array($result);
if (!empty($data["map"])) {
    // Xuất dữ liệu ra GIF MIME
    header("Content-Type: image/gif");
    // Xuất dữ liệu ảnh
    echo $data["map"];
}
```

- Tệp sử dụng

```
echo "<img src=\"imgdisp.php?p=" . $p. "\">";
```

Vấn đề truy cập đồng thời…

- Nhiều người truy cập CSDL đồng thời có thể dẫn đến các tình huống sau:
  - *Mất cập nhật:* Người dùng A đọc giá trị từ CSDL. Người dùng B cũng đọc giá trị từ CSDL và cập nhật ngay lập tức. Người dùng A cập nhật, ghi đè giá trị của người B
  - *Đọc sai:* Người dùng A cập nhật giá trị. Người dùng B đọc giá trị đã cập nhật. Người dùng A undo lại thao tác

    => Giá trị B đọc được không còn đúng
  - *Tính tổng sai:* Người dùng A đang tính tổng thì người dùng B thay đổi giá trị một số mục
  - *Đọc giá trị không thể lặp:* A đọc giá trị, B thay đổi giá trị, A đọc lại thấy giá trị khác

Xử lý truy cập đồng thời...

- Khóa bảng cần thao tác
  - Kiểu khóa READ: cho người dùng khác được đọc nhưng không được ghi
  - Kiểu khóa WRITE: không cho người dùng khác đọc hay ghi
- Thực hiện truy vấn
- Thực hiện cập nhật
- Mở khóa bảng

Ví dụ Xử lý cập nhật đồng thời...

```
//Khóa các bảng cần thao tác
$query = "LOCK TABLES items READ, orders WRITE, customer READ";
if (!mysql_query($query, $connection)) showerror();
// Thực hiện truy vấn
$query = "SELECT SUM(price*qty) from FROM items, orders, customer
WHERE customer.cust_id = orders.cust_id AND orders.order_id =
items.order_id AND items.cust_id = orders.cust_id AND orders.order_id
= $orderId AND customer.cust_id = $custId";
if (!($result = mysql_query($query, $connection))) showerror( );
$row = mysql_fetch_array($result);
//rồi cập nhật
if ($row["SUM(price*qty)"] > $minimum) {
   $query = "UPDATE orders SET discount = $discount WHERE cust_id =
$custId AND order_id = $orderId";
   if (!mysql_query($query, $connection)) showerror();
}
// Mở khóa các bảng
$query = "UNLOCK TABLES";
```

1. Cơ sở dữ liệu
2. Lưu trạng thái

Trạng thái của ứng dụng…

- HTTP là giao thức phi trạng thái
  - Mỗi yêu cầu (request) được xử lý độc lập. Không yêu cầu server nhớ trạng thái của các xử lý trước
- Ứng dụng có thể cần nhớ trạng thái
  - Khi xử lí trên nhiều trang, cần sự tương tác phức tạp
    - Ví dụ: chuyển qua nhiều trang khác nhau để chọn nhiều mặt hàng đưa vào giỏ hàng
  - Cần tính cá nhân hóa
    - Ví dụ: phải biết người dùng nào đang sử dụng để cung cấp nội dung phù hợp

Các phương pháp lưu trạng thái...

- Lưu trạng thái ở trình khách
  - Sử dụng cookie

    https://www.w3schools.com/php/php_cookies.asp
- Lưu trạng thái ở trình phục vụ
  - Sử dụng phiên (session)

    https://www.w3schools.com/php/php_sessions.asp

Cookie…

- Một cookie là mẫu tin *(tên, giá trị, …)*
  - Server gửi cookie cho client (trong đáp ứng cho yêu cầu trước)
  - Client nhớ cookie và gửi cookie cho server trong các yêu cầu sau
  - Server xử lý theo cookie nhận được
- Server gửi cookie cho client: Trình duyệt nhớ cookie và gửi cookie trong tiêu đề của các requests tiếp sau, ví dụ

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: food=choco; tasty=strawberry
```

- Client gửi cookie cho server: Client nhớ cookie và gửi cookie cho server bằng đặt thuộc tính Cookie trong tiêu đề HTTP Request, ví dụ

```
GET /nextPage.htm HTTP/1.1
Host: www.example.com
Cookie: food=choco; tasty=strawberry
```

Các thành phần của cookie …

- Các thành phần
  - *key=<value>*
  - *Expires=<date>*
  - *Max-Age=<non-zero-digit>*
  - *Domain=<domain-value>*
  - *Path=<path-value>*
  - *Secure*
  - *HttpOnly*
- Ví dụ
  - *tasty=strawberry; Expires=Wed, 21 Oct 2017 07:28:00 GMT; Secure;*

Server tạo và gửi cookie…

- `int setcookie(string name, [string value], [int expire], [string path], [string domain], [bool secure], [bool httponly])`
  - *name*: tên cookie
  - *value*: giá trị của cookie
  - *expire*: thời điểm cookie hết hạn
  - *path*: đường dẫn trình duyệt sẽ gửi cookie
  - *domain*: tên miền trình duyệt sẽ gửi cookie
  - *secure*: chỉ truyền cookie qua kết nối an toàn
  - *httponly*: trình duyệt không được truy cập cookie bằng javascript

Server nhận và xử lý cookie…

- Server nhận và xử lý cookie:
  - Cookie được lưu trong mảng $_COOKIE
  - Nếu bật *register_globals* (trong php.ini), biến có tên cookie được khởi tạo
  - Ví dụ: `$tasty= $_COOKIE*"tasty"+;`
- Client truy cập cookie bằng javascript: Nếu httponly = false
  - Ví dụ

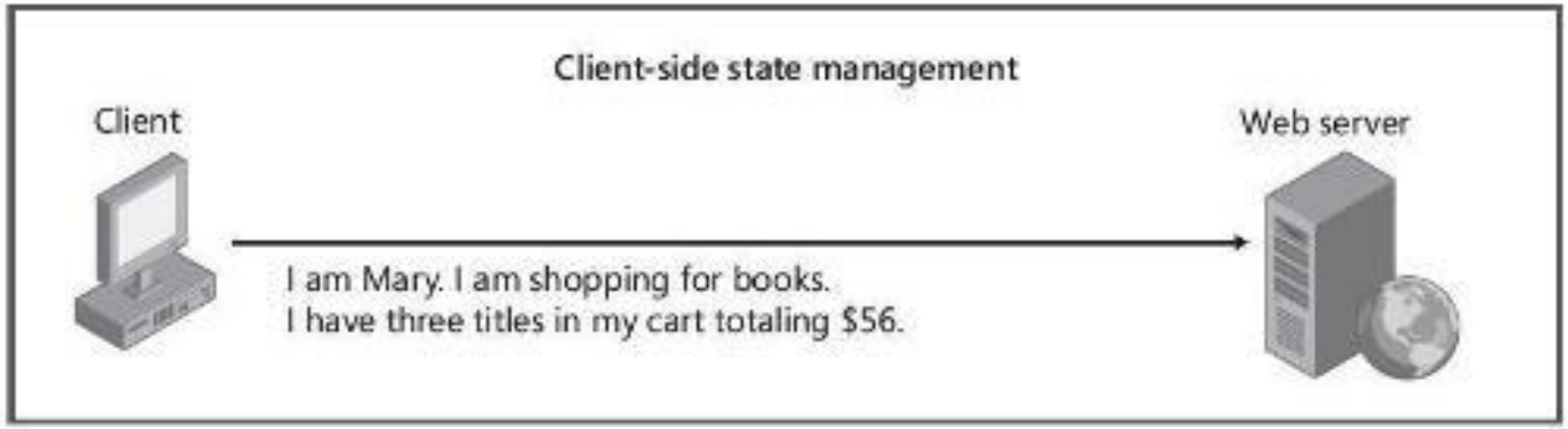
```
<script type="text/javascript">
    console.log(document.cookie);
    document.cookie = "amount=3";
    console.log(document.cookie);
</script>
```

Ví dụ sử dụng cookie…

```
<?php
if (!isset($_COOKIE["guideshown"])) { //Truy cập trang lần đầu
?>
    <div id="guide">
    Chào mừng quý vị ghé thăm trang của chúng tôi.
    Quý vị vui lòng giành ít thời gian xem bản giới thiệu.<br><br>
    <button id="closeguide">Đóng</button>
    </div>
    <script>
        document.getElementById("closeguide").onclick = function()
{
        document.getElementById("guide").style.display = "none";
        };
    </script>
<?php
    setcookie("guideshown", "shown");
}
?>
```

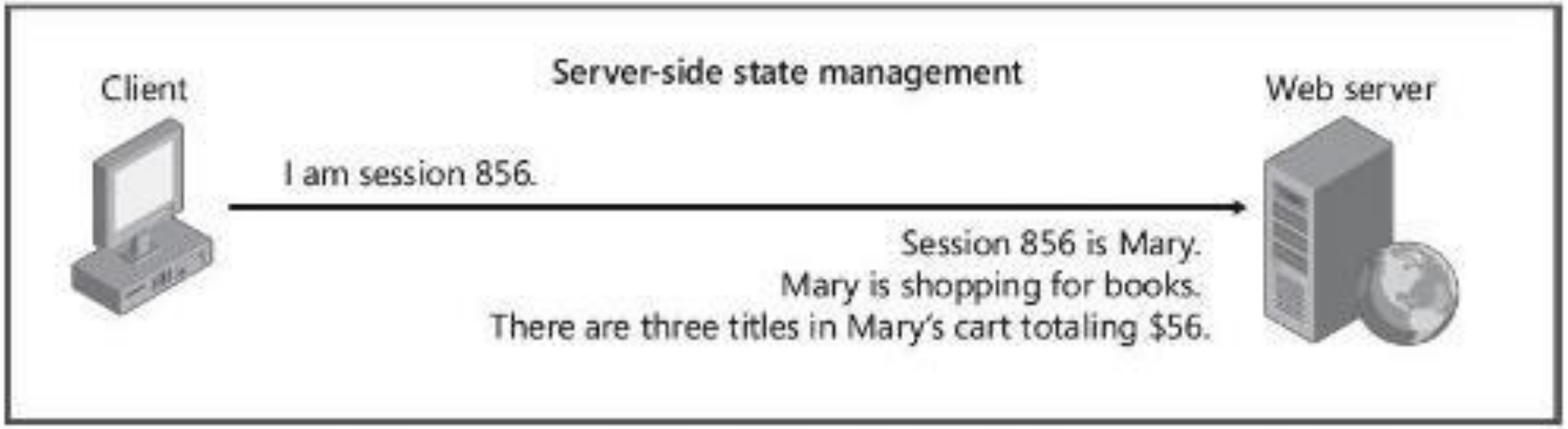
Vấn đề của cookie

- Kích thước request tăng khi bao hàm cookie
- Tính an toàn không cao
- Tính riêng tư bị vi phạm do trình duyệt nhớ cookie



Phiên…

- Phiên (session) là cuộc thoại (dialogue/conversation) giữa trình khách và trình phục vụ.
- Server lưu các biến phiên và gửi cho client định danh phiên. Định danh phiên được server tạo ngẫu nhiên.
- Trình duyệt bao gồm định danh phiên trong các requests sau, server ánh xạ định danh phiên sang các biến phiên
- Phiên phải có thời gian hết hạn (timeout).
- Các biến phiên bị hủy khi ngắt kết nối hoặc hết hạn phiên



Phiên…

- Biến phiên được sử dụng để lưu định danh người dùng đã đăng nhập thành công

```
$_SESSION["tdn"] = $_POST["tdn"];
```

- Kiểm tra định danh người dùng trước khi làm mọi công việc khác

```
if (!isset($_SESSION["tdn"]))
        header("Location:login.php");
```

Phiên…

- Đảm bảo an ninh cho phiên
  - Kiểm soát thời gian phiên
    - Thời gian hết hạn không nên quá dài, tuyệt đối không nên vô thời hạn
    - Thời gian nhàn rỗi
  - Cấp định danh phiên
    - Dài và phức tạp để tránh bị làm giả
  - Hủy phiên
    - Do hết thời gian phiên hoặc vượt quá thời gian nhàn rỗi
    - Chủ động đăng xuất

Phiên…

- Đăng xuất

```
<?php
  //Logout.php
  session_start();
  unset($_SESSION["tdn"]);
  header("Location:login.php");
?>
```

Đăng ký và sử dụng biến phiên...

- Khởi động phiên
  - `session_start( );`
- Sử dụng biến phiên
  - `$_SESSION*"svName"+;`
- Hủy phiên
  - `session_destroy( );`

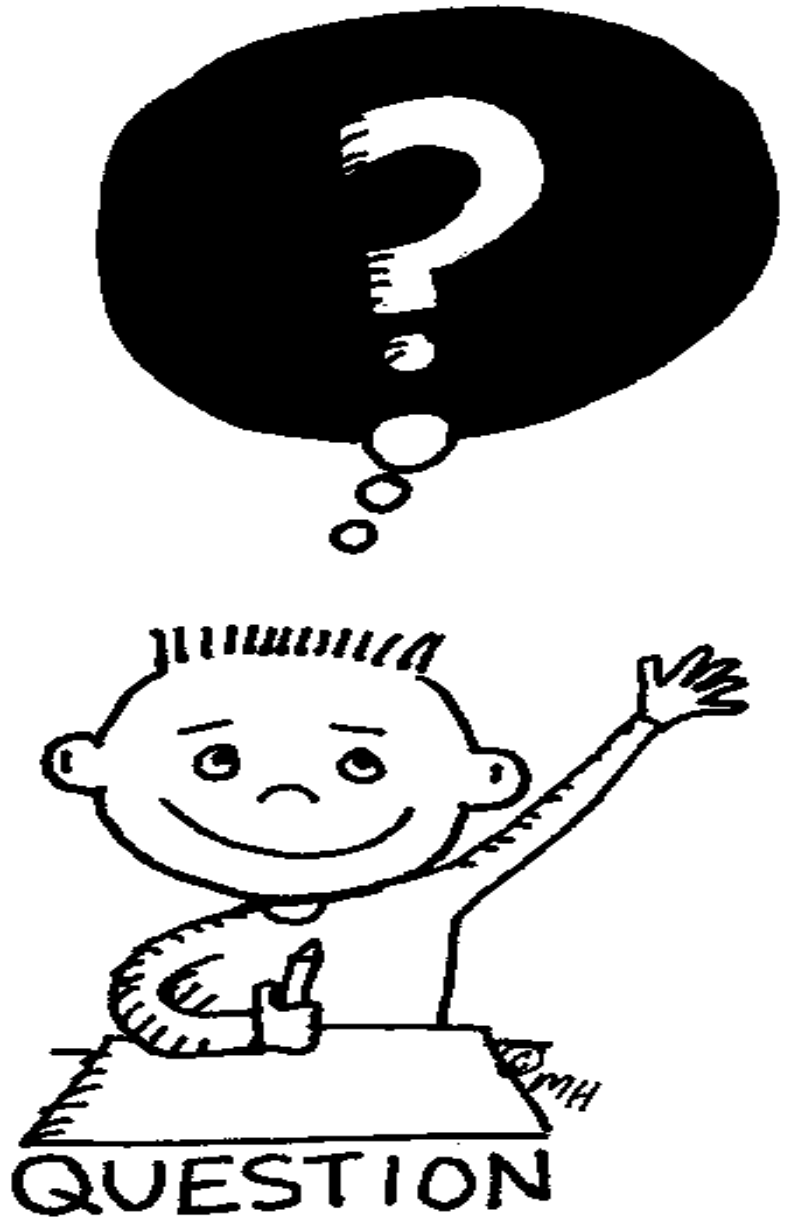
https://www.w3schools.com/php/php_sessions.asp

Ví dụ sử dụng biến phiên…

```
<?php
if (!isset($_SESSION["guideshown"])){//Truy cập trang lần đầu trong phiên
?>
    <div id="guide">
        Chào mừng quý vị ghé thăm trang của chúng tôi.
        Quý vị vui lòng giành ít thời gian xem bản giới
thiệu.<br><br>
        <button id="closeguide">Đóng</button>
    </div>
    <script>
        document.getElementById("closeguide").onclick = function() {
            document.getElementById("guide").style.display = "none";
         };
    </script>
<?php
$_SESSION["guideshown"] = "shown";
}
?>
```

Khi nào nên/không nên sử dụng biến phiên

- Nên
  - *Tăng hiệu năng*: Thực hiện tính toán phức tạp một lần, lưu kết quả trong biến phiên, sử dụng kết quả nhiều lần
  - *Cần chuỗi các tương tác*: Người dùng cần nhập liệu trên nhiều giao diện khác nhau, nếu cần có thể quay về giao diện trước để sửa dữ liệu đã được nhập ở giao diện trước
  - *Kết quả trung gian*: Nhiều kết quả trung gian nên được ghi nhớ cho các tính toán tiếp sau
  - *Cá nhân hóa*: Lưu định danh người dùng ở dạng biến phiên, căn cứ vào định danh người dùng để cung cấp nội dung phù hợp
- Không nên
  - *Lưu trữ trên server*: Nếu lạm dụng sử dụng biến phiên, server sẽ phải dành nhiều bộ nhớ để lưu
  - *An toàn*: Hacker có thể lợi dụng phiên để thực hiện các tấn công


QUESTION

Tổng kết

1. Đã tìm hiểu về CSDL
2. Đã tìm hiểu về vấn đề lưu trạng thái trong ứng dụng web

Bài tập về nhà

1. Đọc thêm tài liệu về MySQL
2. Làm bài tập số 4