



TRUNG TÂM TIN HỌC  
Trường Đại học Khoa Học Tự Nhiên - Đại học quốc gia TP.HCM

# LẬP TRÌNH VIÊN MÃ NGUỒN MỞ PHP

## Module 3

---

*Click vào phụ lục để chuyển tới bài cần đọc*

### Phụ lục

Bài 1 Tổng quan CSDL MYSQL .....	2
Bài 2 Truy vấn trong CSDL MySQL - SELECT .....	24
Bài 3 Truy vấn trong CSDL MySQL – Truy vấn con & Cập nhập dữ liệu .....	42
Bài 4 Kết hợp PHP & MySQL – Thư viện PDO 1 .....	57
Bài 5 Kết hợp PHP & MySQL – Thư viện PDO 2 .....	69
Bài 6 Hiển thị dữ liệu trên trang Web .....	80
Bài 7 PHP Email & Secure Email .....	88
Bài 8 Smarty – Template Engine 1 .....	96
Bài 9 Smarty – Template Engine 2 .....	110
Bài 10 Xây dựng Web Service .....	123



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV MÃ NGUỒN MỞ PHP

### Module 3 – Bài 1: *Tổng quan CSDL MySQL*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Tổng quan



### □ CSDL

- Khái niệm
- Chức năng
- Các đối tượng chính của CSDL quan hệ
- Hệ quản trị CSDL
- SQL (Structure Query Language)





## Tổng quan

### □ CSDL

- Khái niệm: CSDL là một tập hợp dữ liệu được lưu trữ một cách có tổ chức nhằm giúp việc xem, tìm kiếm và lấy thông tin được nhanh chóng và chính xác, giúp giảm công sức và thời gian quản lý thông tin cần thiết



## Tổng quan

### □ CSDL

- Chức năng
  - Lưu trữ
  - Truy cập
  - Tổ chức
  - Xử lý





## Tổng quan



### □ CSDL

- Các đối tượng chính của CSDL quan hệ
  - Bảng (table)
  - Quan hệ (relationship)



## Tổng quan



- Bảng
  - Là thành phần trung tâm của CSDL, được dùng để lưu trữ thông tin của CSDL
  - Gồm hai thành phần: dòng và cột





## Tổng quan

### ● Bảng – Dòng (Row)

- Dòng: là tập hợp các thông tin của tất cả cột dữ liệu trong bảng
- Ví dụ: Bảng khách hàng KHACH\_HANG có các dòng dữ liệu sau:
  - Mỗi dòng trong bảng khách hàng lưu trữ thông tin về một khách hàng trong thực tế

MKH	TEN_KH	PHAI	DIA_CHI	DT	EMAIL
KH001	Trần Văn An	0	123 Nguyễn Du	8123456	tvan@yahoo.com
KH002	Nguyễn Thanh An	0	30 Lê Thánh Tôn	9852147	ntan@yahoo.com
KH003	Lê Thanh Thảo	1	22bis Pasteur	8976431	ltthao@gmail.com



LTV Mã nguồn mở PHP - Module 3

8

## Tổng quan



### ● Bảng – Cột (Column)

- Cột: là một khối dữ liệu trong bảng, có cùng loại dữ liệu, có các thông tin chính:
  - Tên cột: dùng để phân biệt với các cột khác trong bảng. Tên cột trong bảng phải duy nhất và không dùng các ký tự đặc biệt.
  - Kiểu dữ liệu của cột: xác định loại giá trị nào được phép lưu trữ trong cột
- Ví dụ: Bảng khách hàng KHACH\_HANG có các cột sau: MKH (mã khách hàng), TEN\_KH (tên khách hàng), PHAI (phái), ...



LTV Mã nguồn mở PHP - Module 3

9



## Tổng quan

### □ CSDL

#### ● Quan hệ

- Là thành phần được dùng để tạo mối liên kết giữa các bảng dữ liệu với nhau nhằm đảm bảo tính nhất quán, đúng đắn của dữ liệu trong CSDL.
- Có ba loại quan hệ chính:
  - Quan hệ 1 – 1
  - Quan hệ 1 – nhiều
  - Quan hệ nhiều – nhiều



## Tổng quan

#### ● Quan hệ - Quan hệ 1-1

- Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu bên bảng này có liên hệ với duy nhất với một dòng dữ liệu bên bảng kia và ngược lại

- Ví dụ:
 

NHÂN VIÊN

1 — 1

SƠ YẾU LÝ LỊCH

  - **Một** nhân viên chỉ có **một** sơ yếu lý lịch
  - **Một** sơ yếu lý lịch chỉ **thuộc về một** nhân viên





## Tổng quan

### • Quan hệ - Quan hệ 1- nhiều

- Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu bên bảng này có liên hệ với nhiều dòng dữ liệu bên bảng kia và một dòng dữ liệu bên bảng kia sẽ có liên hệ với duy nhất với một dòng dữ liệu bên bảng này. Quan hệ này thường gặp nhất trong CSDL

- Ví dụ:
 

PHÒNG

1 — ∞

NHÂN VIÊN

  - **Một** phòng có **nhiều** nhân viên
  - **Một** nhân viên **chỉ thuộc về** một phòng



LTV Mã nguồn mở PHP - Module 3

12

## Tổng quan



### • Quan hệ - Quan hệ nhiều - nhiều

- Mô tả mối quan hệ giữa hai bảng mà trong đó một dòng dữ liệu bên bảng này có liên hệ với nhiều dòng dữ liệu bên bảng kia và ngược lại. Trong CSDL không lưu trữ quan hệ nhiều nhiều vì vậy khi gặp quan hệ này, chúng ta sẽ chuyển thành các quan hệ một nhiều

- Ví dụ:
 

GIÁO VIÊN

∞ — ∞

LỚP HỌC

  - **Một** giáo viên dạy **nhiều** lớp học.
  - **Một** lớp học có **nhiều** giáo viên.

*Tương đương*

GIÁO VIÊN

1 — ∞

DẠY-HỌC

∞ — 1

LỚP HỌC



LTV Mã nguồn mở PHP - Module 3

13



## Tổng quan

### ❑ CSDL

- **Hệ quản trị CSDL:** Quản lý các dữ liệu được lưu trữ bên trong các CSDL, giúp cho CSDL dễ dàng đến được với người dùng khi cần truy cập thông tin. Có khả năng lưu trữ dữ liệu và cho phép dữ liệu có thể trao đổi với các CSDL khác, và có khả năng
  - Bảo vệ dữ liệu
  - Duy trì dữ liệu
  - Quản lý các giao dịch
  - ...



## Tổng quan

### ❑ CSDL

- **SQL:** Là loại ngôn ngữ cho phép thực hiện các thao tác rút trích, tính toán, cập nhật trên các dữ liệu được lưu trữ trong CSDL





## Tổng quan

### □ CSDL MySQL

- Giới thiệu
- Đặc điểm
- Các tập tin vật lý lưu trữ CSDL
- Quy tắc đặt tên
- Tạo CSDL
- Xóa CSDL



## Tổng quan

### □ CSDL MySQL

- Giới thiệu: CSDL MySQL là tập hợp các đối tượng: bảng, bảng ảo,... cho phép người dùng lưu trữ và truy xuất các thông tin đã được tổ chức và lưu trữ bên trong đó.





## Tổng quan

### □ CSDL MySQL

#### ● Đặc điểm

- Sử dụng cho các ứng dụng Web có quy mô vừa và nhỏ.
- Để thực hiện các thao tác trên CSDL, có thể sử dụng giao diện đồ họa hay dùng dòng lệnh (command line)



## Tổng quan

### □ CSDL MySQL

#### ● Các tập tin vật lý lưu trữ CSDL

- Mỗi bảng sẽ được lưu trữ dưới ba tập tin vật lý:
  - **.frm** : lưu định dạng (cấu trúc) của bảng
  - **.MYD** : lưu nội dung của bảng
  - **.MYI** : lưu chỉ mục của bảng
- Các tập tin này sẽ được tự động lưu trữ trong thư mục: `wamp\mysql\data\tên_CSDL`





## Tổng quan

### □ CSDL MySQL

- Chiều dài của tên CSDL, bảng, chỉ mục, cột, định danh

Loại	Chiều dài tối đa (bytes)	Chiều dài tối đa (ký tự không dấu)
CSDL (database)	64	64
Bảng (Table)	64	64
Chỉ mục (Index)	64	64
Cột (Column)	64	64
Định danh (Alias)	255	255



## Tổng quan

### □ CSDL MySQL

- Quy tắc đặt tên
  - Tên không kết thúc bằng khoảng trắng
  - Tên CSDL không có các ký tự **/, \, ., :, \*, ", <, >**
  - Tên bảng không có các ký tự **/, \, ., :, \*, ", <, >, |**
  - Chiều dài của tên tối đa là 64 ký tự không dấu. Khi sử dụng các ký tự đa byte thì chiều dài sẽ dựa trên tổng số byte của tất cả các ký tự được dùng.







## Tổng quan

### □ CSDL MySQL

#### ● Tạo CSDL

##### ■ Các thuộc tính của CSDL

- **Tên CSDL:** phải duy nhất trong một hệ quản trị CSDL MySQL
- **Vị trí lưu trữ:** Khi tạo mới một CSDL hệ thống sẽ tự động tạo ra một thư mục có tên của CSDL và được lưu tại thư mục `wamp\mysql\data\`

- Có hai cách để tạo một CSDL là dùng giao diện đồ họa hoặc dùng dòng lệnh

*(GV demo cách tạo CSDL bằng GD đồ họa cho HV)*



## Tổng quan



#### ▪ Cú pháp:

**CREATE DATABASE** Tên\_CSDL

**[[ DEFAULT] CHARACTER SET** <character set name>]

**[[ DEFAULT] COLLATE** <collation name>]

- Ví dụ: Tạo CSDL `dich_vu_an_uong`

**CREATE DATABASE** `dich_vu_an_uong`

**CHARACTER SET** `utf-8`

**COLLATE** `utf8_unicode_ci`

**Hay**

**CREATE DATADASE** `dich_vu_an_uong`



## Tổng quan



### ❑ CSDL MySQL

- Xóa CSDL: Dùng câu lệnh DROP DATABASE

- **Cú pháp:**

**DROP DATABASE TÊN\_CSDL**

- **Ví dụ: Xóa CSDL dịch\_vu\_an\_uong**

```
DROP DATABASE dịch_vu_an_uong
```

*(GV demo cách xóa bằng GD đồ học cho HV)*



## Nội dung





## Bảng

### ❑ Quy tắc đặt tên

- Tên bảng (table name)
  - Do người dùng tạo ra
  - Duy nhất trong CSDL
- Tên cột
  - Do người dùng tạo ra
  - Duy nhất trong bảng
  - Ví dụ: Đặt tên cho cột mã sữa trong bảng sữa là: Ma\_sua



LTV Mã nguồn mở PHP - Module 3

26

## Bảng



### ❑ Các kiểu dữ liệu trong bảng

#### ● Số nguyên

Kiểu dữ liệu	Kích thước	Miền giá trị
Tinyint	1 byte	-127 => 128 hay 0..255
Smallint	2 bytes	-32768 => 32767 hay 0..65535
Mediumint	3 bytes	-8388608 => 8388607 hay 0..16777215
Int	4 bytes	$-2^{31}$ => $2^{31}-1$ hay 0.. $2^{32}-1$
Bigint	8 bytes	$-2^{63}$ => $2^{63}-1$ hay 0.. $2^{64}-1$

#### ● Boolean (true/false)

Kiểu dữ liệu	Kích thước	Miền giá trị
Bool / boolean	1 byte	Có hai giá trị là True và False



LTV Mã nguồn mở PHP - Module 3

27



## Bảng

### □ Các kiểu dữ liệu trong bảng

#### • Số thập phân: decimal và numeric

- Là những kiểu dữ liệu được dùng để lưu trữ những giá trị số cụ thể.
- Giá trị được lưu với định dạng nhị phân.
- Cú pháp:

**Decimal(M[, N])**

- Trong đó: M là tổng ký số và N là số ký số thập phân Số nguyên

Kiểu dữ liệu	Kích thước	Miền giá trị
Decimal/ Numeric	4 bytes	



LTV Mã nguồn mở PHP - Module 3

28

## Bảng

### □ Các kiểu dữ liệu trong bảng

#### • Số thực

Kiểu dữ liệu	Kích thước	Miền giá trị
Float	4 bytes	-3.402823466E+38 => -1.175494351E-38; 0; 1.175494351E-38 => 3.402823466E+38
Double	8 bytes	-1.7976931348623157E+308 => -2.2250738585072014E-308; 0; 2.2250738585072014E-308 => 1.7976931348623157E+308

#### • Date/time

Kiểu dữ liệu	Miền giá trị	Diễn giải
Date	'1000-01-01' => '9999-12-31'	Ngày với định dạng yyyy-mm-dd
Datetime	'1000-01-01 00:00:00' => '9999-12-31 23:59:59'	Ngày giờ với định dạng yyyy-mm-dd hh:mm:ss
Time	'00:00:00' => '23:59:59'	Giờ với định dạng hh:mm:ss
Year[(2 4)]	4 ký số: '1901' => '2155' 2 ký số: '1970' => '2069'	Năm với định dạng 2 ký số hoặc 4 ký số
Timestamp [(kích cỡ định dạng)]	'1970-01-01 00:00:01'	Timestamp trình bày dưới dạng yyyy-mm-dd hh:mm:ss



LTV Mã nguồn mở PHP - Module 3

29

## Bảng



## ❑ Các kiểu dữ liệu trong bảng

- Chuỗi

Kiểu dữ liệu	Miền giá trị	Diễn giải
Char	1 => 255 ký tự	Chuỗi cố định
Varchar	1 => 255 ký tự	Chuỗi động
Tinyblob	1 => $2^8$ -1 bytes (255 bytes)	Kiểu đối tượng nhị phân cỡ 255 ký tự
Tinytext	1 => $2^8$ -1 ký tự (255 ký tự)	Kiểu đối tượng chuỗi kích cỡ 255 ký tự
Blob	1 => $2^{16}$ -1 bytes (65535 bytes)	Kiểu blob cỡ 65535 ký tự
Text	1 => $2^{16}$ -1 ký tự (65535 ký tự)	Kiểu chuỗi dạng văn bản cỡ 65535 ký tự
Mediumblob	1 => $2^{24}$ -1 bytes (16777215 bytes)	Kiểu blob vừa cỡ 16777215 ký tự
Mediumtext	1 => $2^{24}$ -1 ký tự (16777215 ký tự)	Kiểu chuỗi dạng văn bản vừa 16777215 ký tự
Longblob	1 => $2^{32}$ -1 bytes (4GB)	Kiểu blob lớn khoảng 4GB ký tự
Longtext	1 => $2^{32}$ -1 ký tự (4GB)	Kiểu chuỗi dạng văn bản lớn khoảng 4GB ký tự

- Char vs Varchar

Giá trị	Char(4)	Số bytes	Varchar(4)	Số bytes
'a'	'a'	4 bytes	'a'	1 byte
'ab'	'ab'	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes



LTV Mã nguồn mở PHP - Module 3

30

## Bảng



### Thuộc tính của cột trong bảng

- Độ dài dữ liệu (length/value)
- Kiểu hiển thị (collation)
- Thuộc tính (attribute)
- Cho phép để trống dữ liệu (NULL)
- Giá trị mặc định (default)
- Thuộc tính mở rộng (extra)
- Ghi chú (comment)
- Tăng giá trị tự động (auto increment)



LTV Mã nguồn mở PHP - Module 3

31



## Bảng

### ❑ Thao tác với bảng

- Tạo bảng
- Thay đổi cấu trúc bảng
- Xóa bảng
- Tạo mối quan hệ giữa các bảng

*(GV demo các thao tác bằng GD đồ học cho HV)*



## Bảng

### ❑ Tạo bảng

- Cú pháp

```
CREATE TABLE Tên_bảng
(
    Tên_cột_1 kiểu_dữ_liệu[(kích_cỡ)] [NOT NULL] PRIMARY KEY,
    Tên_cột_2 kiểu_dữ_liệu[(kích_cỡ)] DEFAULT giá_trị,
    Tên_cột_3 kiểu_dữ_liệu[(kích_cỡ)] [NOT NULL] UNIQUE,
    Tên_cột_4 kiểu_dữ_liệu[(kích_cỡ)] [NOT NULL]
    ...
)
```







## Bảng

### □ Tạo bảng

#### ● Cú pháp

```
CREATE TABLE Tên_bảng
(
    Tên_cột_1 kiểu_dữ_liệu[(kích_cỡ)] [NOT NULL],
    Tên_cột_3 kiểu_dữ_liệu[(kích_cỡ)] DEFAULT giá_trị,
    Tên_cột_3 kiểu_dữ_liệu[(kích_cỡ)] [NOT NULL],
    ...
    UNIQUE (các_cột_độc_nhất),
    PRIMARY KEY (Các_cột_tham_gia_làm_khóa_chính)
)
```



LTV Mã nguồn mở PHP - Module 3

34

## Bảng

### □ Tạo bảng

```
CREATE TABLE hang_sua
(
    Ma_hang_sua varchar(20) NOT NULL AUTO_INCREMENT PRIMARY
KEY,
    Ten_hang_sua varchar(100) NOT NULL,
    Dia_chi varchar(200),
    Dien_thoai varchar(20),
    Email varchar(100)
)

CREATE TABLE hang_sua
(
    Ma_hang_sua varchar(20) NOT NULL
AUTO_INCREMENT,
    Ten_hang_sua varchar(100) NOT NULL,
    Dia_chi varchar(200),
    Dien_thoai varchar(20),
    Email varchar(100),
    UNIQUE(Ten_hang_sua, Email),
    PRIMARY KEY(Ma_hang_sua)
)
```



LTV Mã nguồn mở PHP - Module 3

35

## Bảng



### ❑ Thay đổi cấu trúc bảng

- Thêm cột

**ALTER TABLE** Tên\_bảng

**ADD**

Tên\_cột kiểu\_dữ\_liệu[(kích\_cỡ)] [...]

- Đổi kiểu dữ liệu của cột

**ALTER TABLE** Tên\_bảng

**CHANGE** tên\_cột\_cũ tên\_cột\_mới kiểu\_dữ\_liệu\_mới  
[kích\_cỡ]

- Hủy cột

**ALTER TABLE** Tên\_bảng

**DROP COLUMN** Tên\_cột, ...



LTV Mã nguồn mở PHP - Module 3

36

## Bảng



### ❑ Xóa bảng

**DROP TABLE** Tên\_bảng



LTV Mã nguồn mở PHP - Module 3

37





## Bảng

### ❑ Tạo mối quan hệ giữa các bảng

- Tạo mối quan hệ giữa các bảng với nhau (giữa bảng cha (là khoá chính – primary key) và bảng con (là khoá ngoại – foreign key))
- Giúp cho việc cập nhật, xoá dữ liệu được chính xác và nhanh chóng



LTV Mã nguồn mở PHP - Module 3

38

## Bảng

### ❑ Tạo mối quan hệ giữa các bảng

- Tạo khóa ngoại

FOREIGN KEY (ten\_cot\_khoa\_ngoai)  
REFERENCES ten\_bang\_cha  
(ten\_cot\_khoa\_chinh) ON UPDATE CASCADE,  
...

- Ví dụ

```
ALTER TABLE sua
ADD FOREIGN KEY (ma_loai_sua) REFERENCES
loai_sua (ma_loai_sua) ON UPDATE CASCADE,
ADD FOREIGN KEY (ma_hang_sua) REFERENCES
hang_sua (ma_hang_sua) ON UPDATE CASCADE;
```



LTV Mã nguồn mở PHP - Module 3

39



## Import và Export dữ liệu



### □ Import dữ liệu

- Đưa dữ liệu từ tập tin vào CSDL MySQL

**(GV demo các thao tác bằng GD đồ học cho HV)**

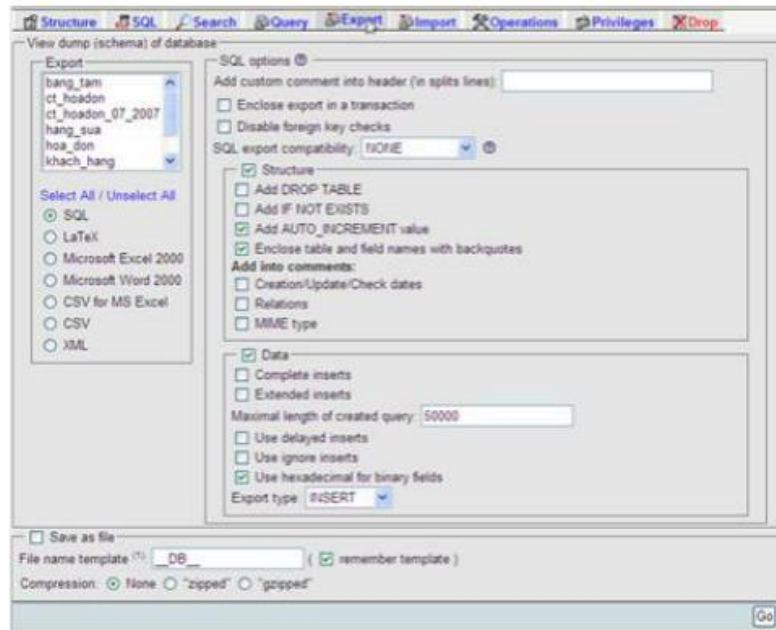


## Import và Export dữ liệu

### Export dữ liệu

**Xuất dữ  
liệu từ  
Database  
trong  
MySQL ra  
tập tin**

*(GV demo  
các thao tác  
bằng GD đồ  
học cho HV)*



LTV Mã nguồn mở PHP - Module 3

42



LTV Mã nguồn mở PHP - Module 3

43



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV MÃ NGUỒN MỞ PHP

Module 3 – Bài 2: *Truy vấn trong CSDL MySQL*

**- SELECT**

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Truy vấn đơn giản



### ❑ Chọn ra dữ liệu của các cột có trong một bảng

- Cú pháp:

```
SELECT Danh_sách_các_cột
```

```
FROM Tên_bảng
```

- Với tên các cột trong danh\_sách\_các\_cột phải chính xác
- Dùng \* khi muốn lấy dữ liệu từ tất cả các cột trong bảng





## Truy vấn đơn giản

### ❑ Chọn ra dữ liệu của các cột có trong một bảng

#### ● Ví dụ

```
SELECT Ma_hang_sua,
       Ten_hang_sua
FROM hang_sua
```

Hiển thị tất cả các thông tin có trong bảng hang\_sua

```
SELECT *
FROM hang_sua
```

Hiển thị thông tin của các hãng sữa có trong bảng hang\_sua gồm có các cột như sau: mã hãng sữa, tên hãng sữa, điện thoại

Ma_hang_sua	Ten_hang_sua	Dien_thoai
VNM	Vinamilk	8794561
NTF	Nutifood	7895632
AB	Abbott	8741258
DS	Daisy	5789321
DL	Dutch Lady	7826451
DM	Dumex	6258943
MJ	Mead Jonhson	8741258







## Sắp xếp

### ❑ ORDER BY

- Câu lệnh SELECT ... FROM kết hợp với mệnh đề ORDER BY giúp lấy dữ liệu của các cột bên trong bảng đồng thời sắp xếp lại dữ liệu theo thứ tự tăng dần hoặc giảm dần.

- Cú pháp:

**SELECT** Danh\_sách\_các\_cột

**FROM** Tên\_bảng

**ORDER BY** Tên\_cột\_sắp\_xếp [DESC, ...]

- Mặc định, cột sẽ được sắp xếp tăng dần, nếu muốn sắp xếp cột theo thứ tự giảm dần thì phía sau tên cột sắp xếp cần thêm từ khóa DESC



LTV Mã nguồn mở PHP - Module 3

6

## Sắp xếp

### ❑ ORDER BY

Liệt kê thông tin mã hãng sữa, tên hãng sữa và email, sắp tăng dần/giảm dần theo tên hãng sữa

```
SELECT Ma_hang_sua,
Ten_hang_sua, Email
FROM hang_sua
ORDER BY Ten_hang_sua
```

```
SELECT Ma_hang_sua,
Ten_hang_sua, Email
FROM hang_sua
ORDER BY Ten_hang_sua
DESC
```

Ma_hang_sua	Ten_hang_sua	Email
AB	Abbott	abbott@ab.com
DS	Daisy	daisy@ds.com
DM	Dumex	dumex@dm.com
DL	Dutch Lady	dutchlady@dl.com
MJ	Mead Jonhson	meadjohn@mj.com
NTF	Nutifood	nutifood@ntf.com
VNM	Vinamilk	vinamilk@vnm.com

Ma_hang_sua	Ten_hang_sua	Email
VNM	Vinamilk	vinamilk@vnm.com
NTF	Nutifood	nutifood@ntf.com
MJ	Mead Jonhson	meadjohn@mj.com
DL	Dutch Lady	dutchlady@dl.com
DM	Dumex	dumex@dm.com
DS	Daisy	daisy@ds.com
AB	Abbott	abbott@ab.com



LTV Mã nguồn mở PHP - Module 3

7



## Truy vấn có sử dụng As, Limit và Distinct



### □ AS

- Sử dụng AS để đặt lại tên (alias) cho các cột
- Cú pháp  
`SELECT Tên_cột AS Định_danh_cột, ...`  
`FROM Tên_bảng`  
`[ORDER BY Tên_cột_sắp_xếp [DESC, ...]]`
- Ví dụ: Đặt định danh cho cột Don\_gia là DG

```
SELECT Ma_sua, Ten_sua, Don_gia as DG,  
       Trong_luong  
FROM sua
```







## Truy vấn có sử dụng As, Limit và Distinct

### □ Giới hạn mẫu tin LIMIT

- Câu lệnh SELECT ... FROM kết hợp với mệnh đề LIMIT n,m giúp lấy ra m mẫu tin trong bảng tính từ vị trí n, theo một tiêu chuẩn sắp xếp nào đó

- Cú pháp:

```
SELECT Danh_sách_các_cột
FROM Tên_bảng
[ORDER BY Tên_cột_sắp_xếp [DESC, ...]]
LIMIT n,m
```



## Truy vấn có sử dụng As, Limit và Distinct

### □ Giới hạn mẫu tin limit

- Ví dụ: Cho biết 3 sữa trong bảng sữa có đơn giá lớn nhất gồm các thông tin: mã sữa, tên sữa, đơn giá, trọng lượng

```
SELECT Ma_sua, Ten_sua, Don_gia,
Trong_luong
FROM sua
ORDER BY Don_gia DESC
LIMIT 0,3
```

Ma_sua	Ten_sua	Don_gia	Trong_luong
MJ0005	EnfaPro A+	241000	900
MJ0004	EnfaPro	198000	900
MJ0001	Enfa Mama A+	196000	900





## Truy vấn có sử dụng As, Limit và Distinct

### ❑ DISTINCT

- Trong 1 bảng, 1 cột có thể chứa nhiều giá trị trùng lặp; và đôi khi ta chỉ muốn lấy ra danh sách những giá trị duy nhất không trùng lặp => Sử dụng từ khóa DISTINCT
- Cú pháp:  

```
SELECT DISTINCT Tên_cột_1, Tên_cột_2,...
FROM Tên_bảng
[ORDER BY Tên_cột_sắp_xếp [DESC, ...]]
```
- Ví dụ: Cho biết các mã hãng sữa có trong bảng Sữa

```
SELECT DISTINCT Ma_hang_sua
FROM sua
```



LTV Mã nguồn mở PHP - Module 3

12



LTV Mã nguồn mở PHP - Module 3

13



## Lọc theo mẫu tin

### □ WHERE

- Câu lệnh SELECT ... FROM kết hợp với mệnh đề WHERE giúp lọc các dòng dữ liệu bên trong bảng, dữ liệu này phải thỏa điều kiện đưa ra trong mệnh đề WHERE
- Cú pháp:  

```
SELECT Danh_sách_các_cột
FROM Tên_bảng
WHERE Điều_kiện_lọc
[ORDER BY Tên_cột_sắp_xếp [DESC, ...]]
```
- Các phép toán thường gặp trong điều kiện lọc
  - So sánh: >, >=, <, <=, =, !=, <>
  - Logic: and, or, not, in, not in, between, like, not like



LTV Mã nguồn mở PHP - Module 3

14

## Lọc theo mẫu tin



### □ WHERE

```
SELECT Ma_sua, Ten_sua,
Ma_hang_sua, Don_gia
FROM sua
WHERE Ma_hang_sua = "AB"
```

Ma_sua	Ten_sua	Ma_hang_sua	Don_gia
AB0001	Gain Advance	AB	107000
AB0002	Gain IQ	AB	107000
AB0003	Abbott Grow	AB	87000
AB0004	Abbott Grow School	AB	87000
AB0005	Abbott Pedia Sure	AB	146000
AB0006	Similac Neo Sure	AB	145000

```
SELECT Ma_sua, Ten_sua,
Trong_luong
FROM sua
WHERE (Ten_sua LIKE
"%Dielac%" OR Ten_sua
LIKE "%Enfa%") AND
Trong_luong IN(400, 500, 900)
```

Ma_sua	Ten_sua	Trong_luong
MJ0001	Enfa Mama A+	900
MJ0002	EnfaLac	400
MJ0003	EnfaGrow	400
MJ0004	EnfaPro	900
MJ0005	EnfaPro A+	900
VNM006	Dielac Alpha	900
VNM009	Dielac Canxi Premier 2000	400
VNM011	Dielac Sure	400
VNM012	Dielac Mamma	900



LTV Mã nguồn mở PHP - Module 3

15



## Truy vấn trên nhiều bảng



### ❑ Loại 1

- Có thể sử dụng mệnh đề WHERE để liên kết dữ liệu của nhiều bảng trong truy vấn
- Cú pháp:

```
SELECT Danh_sách_các_cột  
FROM Tên_bảng_1, Tên_bảng_2, ...  
WHERE Tên_bảng_1.tên_cột =  
      Tên_bảng_2.tên_cột  
[ORDER BY Tên_cột_sắp_xếp [DESC, ...]]
```





## Truy vấn trên nhiều bảng

### ❑Loại 1

- Ví dụ: Hiển thị thông tin các sữa trong bảng sữa (sua) gồm mã sữa, tên sữa, tên hãng sữa sao cho chỉ chọn ra các sữa có mã hãng sữa khác "DL" và "DS" và "VNM"

```
SELECT Ma_sua, Ten_sua,
Ten_hang_sua
FROM hang_sua, sua
WHERE
hang_sua.Ma_hang_sua =
sua.Ma_hang_sua AND
sua.Ma_hang_sua NOT
IN("DL", "DS", "VNM")
```

Ma_sua	Ten_sua	Ten_hang_sua
AB0001	Gain Advance	Abbott
AB0002	Gain IQ	Abbott
AB0003	Abbott Grow	Abbott
AB0004	Abbott Grow School	Abbott
AB0005	Abbott Pedia Sure	Abbott
AB0006	Similac Neo Sure	Abbott
MJ0001	Enfa Mama A+	Mead Jonhson
MJ0002	EnfaLac	Mead Jonhson
MJ0003	EnfaGrow	Mead Jonhson
MJ0004	EnfaPro	Mead Jonhson
MJ0005	EnfaPro A+	Mead Jonhson
NTF001	Nuti Mum	Nutifood
NTF002	Obilac	Nutifood
NTF003	Nuti IQ	Nutifood
NTF004	Nuti Vita	Nutifood
NTF005	Sữa nguyên kem Nuti	Nutifood
NTF006	Nuti Vita	Nutifood
NTF007	DiabetCare	Nutifood



LTV Mã nguồn mở PHP - Module 3

18

## Truy vấn trên nhiều bảng



### ❑Loại 2

- Khi muốn liên kết các bảng có quan hệ với nhau để lấy ra dữ liệu chung → kết hợp **SELECT ... FROM** với mệnh đề **JOIN**.
- Khi sử dụng JOIN để nối các bảng cần phải lưu ý những bảng này phải có các cột liên hệ với nhau và thứ tự quan hệ được chỉ định giữa các bảng sẽ làm ảnh hưởng tới kết quả truy vấn.



LTV Mã nguồn mở PHP - Module 3

19





## Truy vấn trên nhiều bảng

- Ví dụ:
  - Trong ví dụ quản lý bán sữa, khi muốn in một hóa đơn bán sữa cho khách hàng, theo thiết kế sẽ có một số bảng liên quan đến nhau.
  - Nếu muốn biết các sản phẩm sữa được bán cho ai thì liên quan tới bảng khách hàng, khách hàng mua những sản phẩm gì thì liên quan tới bảng chi tiết hóa đơn, sữa được mua có tên là gì thì liên quan tới bảng sữa, ...
  - Để tạo được các mối quan hệ này và truy xuất dữ liệu → dùng JOIN để kết nối các bảng với nhau.



LTV Mã nguồn mở PHP - Module 3

20



## Truy vấn trên nhiều bảng

### □ Inner join

- Khi kết nối các bảng dùng **INNER JOIN** → chỉ định việc so sánh giá trị trong các cột của các bảng là tương đương – dữ liệu đều có ở cả hai bảng.
- Kết quả sau khi thực hiện câu lệnh truy vấn kết hợp **INNER JOIN** là các mẫu tin thỏa điều kiện quan hệ ở cả hai bảng
- Cú pháp:
 

```
SELECT Danh_sách_các_cột
FROM Tên_bảng
INNER JOIN Tên_bảng_liên_kết ON
Điều_kiện_liên_kết
[WHERE Điều_kiện_lọc]
[ORDER BY Danh_sách_các_cột_sắp_xếp [DESC]]
```



LTV Mã nguồn mở PHP - Module 3

21



## Truy vấn trên nhiều bảng

### ❑ Inner join

- Ví dụ: In ra thông tin chi tiết của các hóa đơn trong tháng 7 năm 2007, gồm: số hóa đơn, ngày hóa đơn, mã sửa, số lượng

```
SELECT ct_hoadon.So_hoa_don,
       Ngay_hd, Ma_sua, So_luong
FROM ct_hoadon INNER JOIN
     hoa_don ON ct_hoadon.So_hoa_don =
     hoa_don.So_hoa_don
WHERE MONTH(Ngay_hd) = 7 AND
       YEAR(Ngay_hd)=2007
```

So_hoa_don	Ngay_hd	Ma_sua	So_luong
D001	2007-07-31	AB0001	2
D001	2007-07-31	DL0001	12
D001	2007-07-31	NTF002	8
D001	2007-07-31	VNM012	4
D002	2007-07-30	DL0001	2
D002	2007-07-30	MJ0001	5
D002	2007-07-30	MJ0004	3



## Truy vấn trên nhiều bảng



### ❑ Left join, right join

- Khi kết nối các bảng dùng **LEFT|RIGHT JOIN** → chỉ định việc so sánh giá trị trong các cột của các bảng được ưu tiên cho mối quan hệ bên nhánh trái | nhánh phải. Việc thay đổi thứ tự ưu tiên này sẽ làm ảnh hưởng tới kết quả truy vấn.

- Cú pháp:

```
SELECT Danh_sách_các_cột
FROM Tên_bảng
LEFT|RIGHT JOIN Tên_bảng_liên_kết
ON Điều_kiện_liên_kết
[WHERE Điều_kiện_lọc]
[ORDER BY Danh_sách_các_cột_sắp_xếp [DESC]]
```





## Truy vấn trên nhiều bảng

### ❑ Left join, right join

- Ví dụ: Hiển thị danh sách tất cả các khách hàng hiện có trong bảng khách hàng và hóa đơn mua hàng trong bảng hóa đơn gồm: mã khách hàng, số hóa đơn, ngày hóa đơn.

```
SELECT khach_hang.Ma_khach_hang,
       So_hoa_don, Ngay_hd
FROM khach_hang LEFT JOIN hoa_don
ON khach_hang.Ma_khach_hang =
   hoa_don.Ma_khach_hang
```

Ma_khach_hang	So_hoa_don	Ngay_hd
kh001	D001	2007-07-31
kh002	D002	2007-07-30
kh002	D004	2007-08-11
kh003	D003	2007-08-10
kh004	NULL	NULL
kh005	NULL	NULL
kh006	NULL	NULL
kh008	NULL	NULL



## Truy vấn trên nhiều bảng



### ❑ Mệnh đề liên kết dữ liệu nhiều bảng

- Cú pháp:

```
SELECT Danh_sách_các_cột
FROM Tên_bảng_1
INNER|LEFT|RIGHT JOIN Tên_bảng_2
ON Điều_kiện_liên_kết_bang_1_2
INNER|LEFT|RIGHT JOIN Tên_bảng_3
ON Điều_kiện_liên_kết...
...
[WHERE Điều_kiện_lọc]
[ORDER BY Danh_sách_các_cột_sắp_xếp [DESC]]
```





## Truy vấn trên nhiều bảng



### ❑ Mệnh đề liên kết dữ liệu nhiều bảng

- Ví dụ: hiển thị thông tin chi tiết hóa đơn gồm có ngày hóa đơn, số hóa đơn, tên sữa và số lượng





## Truy vấn có nhóm & hàm tính toán trên nhóm

### □ Group by

- Câu lệnh SELECT ... FROM kết hợp với mệnh đề GROUP BY giúp nhóm dữ liệu của các dòng dữ liệu bên trong bảng và sử dụng thêm các hàm thống kê đi kèm để tính toán dữ liệu có tính chất thống kê
- Cú pháp:

```
SELECT Danh_sách_các_cột, Hàm_thống_kê [as tên]
FROM Tên_bảng
[WHERE Điều_kiện_lọc]
GROUP BY Danh_sách_các_cột_nhóm_dữ_liệu
[ORDER BY Tên_cột_sắp_xếp [DESC, ...]]
```



LTV Mã nguồn mở PHP - Module 3

28



## Truy vấn có nhóm & hàm tính toán trên nhóm

### □ Group by

- Các hàm thống kê:
  - AVG: hàm trả về giá trị trung bình theo nhóm trong câu lệnh truy vấn trên bảng
  - COUNT: hàm trả về số lượng mẫu tin theo nhóm trong câu truy vấn trên bảng
  - MIN: hàm trả về giá trị nhỏ nhất theo nhóm
  - MAX: hàm trả về giá trị lớn nhất theo nhóm
  - SUM: hàm trả về tổng các giá trị theo nhóm



LTV Mã nguồn mở PHP - Module 3

29



## Truy vấn có nhóm & hàm tính toán trên nhóm

### ❑ Group by

- Ví dụ: Thống kê số lượng sữa của mỗi hãng sữa, sắp xếp dữ liệu tăng dần theo số lượng sữa

```
SELECT Hang_sua.Ma_hang_sua,
       Ten_hang_sua, COUNT(sua.Ma_sua) AS
       So_luong_sua
FROM hang_sua INNER JOIN sua ON
hang_sua.Ma_hang_sua =
sua.Ma_hang_sua
GROUP BY Ma_hang_sua, Ten_hang_sua
ORDER BY So_luong_sua
```

Ma_hang_sua	Ten_hang_sua	So_luong_sua
DS	Daisy	2
MJ	Mead Jonhson	5
AB	Abbott	6
NTF	Nutifood	7
DL	Dutch Lady	11
VNM	Vinamilk	12



## Truy vấn có nhóm & hàm tính toán trên nhóm

### ❑ Group by

- Ví dụ: Căn cứ vào bảng chi tiết hóa đơn để thống kê tổng số tiền của mỗi hóa đơn, sắp xếp dữ liệu tăng dần theo tổng số tiền

```
SELECT So_hoa_don,
       Sum(So_luong*Don_gia) as
       Tong_so_tien
FROM ct_hoadon
GROUP BY (So_hoa_don)
ORDER BY Sum(So_luong*Don_gia)
```

So_hoa_don	Tong_so_tien
D001	1480000
D002	1656000
D003	2484500
D004	5573000





## Lọc theo nhóm



### □ HAVING

- Câu lệnh SELECT ... FROM kết hợp với mệnh đề HAVING giúp lọc lại dữ liệu sau khi đã gom nhóm dữ liệu bằng mệnh đề GROUP BY

- Cú pháp:

```
SELECT Danh_sách_các_cột, Hàm_thống_kê [as tên]
FROM Tên_bảng
[WHERE Điều_kiện_lọc]
GROUP BY Danh_sách_các_cột_nhóm_dữ_liệu
HAVING Điều_kiện_lọc_sau_khi_nhóm
[ORDER BY Tên_cột_sắp_xếp [DESC, ...]]
```





## Lọc theo nhóm

### □ HAVING

- Ví dụ: Thống kê số lượng sữa của mỗi hãng sữa, sao cho chỉ lọc ra những hãng sữa có số lượng sữa  $>5$





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

### Module 3 – Bài 3: *Truy vấn trong CSDL MySQL* – *Truy vấn con & Cập nhật dữ liệu*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Truy vấn con



### □ Khái niệm

- Truy vấn con là một câu lệnh truy vấn SELECT được lồng vào các câu lệnh truy vấn khác nhằm thực hiện các truy vấn tính toán phức tạp.
- **Lưu ý:** khi dùng truy vấn con cần tuân theo các quy tắc
  - Truy vấn con phải được đặt trong cặp ngoặc đơn ()
  - Truy vấn con chỉ có thể tham chiếu đến 1 cột hoặc 1 biểu thức.
- Kết quả trả về của truy vấn con có thể là một giá trị hoặc một danh sách các giá trị.



## Truy vấn con

## ❑ Truy vấn con trả về một giá trị

- Là truy vấn mà kết quả trả về của nó là một giá trị duy nhất
- Ví dụ: hãy cho biết sản phẩm sữa nào có trọng lượng lớn nhất.
  - Tìm trọng lượng lớn nhất

```
SELECT MAX(Trong_luong)
FROM sua
```

MAX(Trong\_luong)  
900



## Truy vấn con

- Lọc ra các sữa có trọng lượng bằng với trọng lượng lớn nhất này

```
SELECT Ma_sua, Ten_sua,
Trong_luong
FROM sua
WHERE Trong_luong = (SELECT
MAX(Trong_luong) FROM sua)
```

<b>Ma_sua</b>	<b>Ten_sua</b>	<b>Trong_luong</b>
DL0002	Canximex	900
DS0001	Daisy Không Đường	900
MJ0001	Enfa Mama A+	900
MJ0004	EnfaPro	900
MJ0005	EnfaPro A+	900
VNM006	Dielac Alpha	900
VNM012	Dielac Mamma	900







## Truy vấn con

### ❑ Truy vấn con trả về danh sách các giá trị

- Là truy vấn con mà kết quả trả về là tập hợp các giá trị.
- Toán tử IN hoặc NOT IN thường được dùng trong trường hợp này vì nó so sánh một phần tử có thuộc (hay không thuộc) tập hợp các giá trị hay không.



## Truy vấn con



- Ví dụ: Hãy cho biết các khách hàng nào chưa mua hàng (chưa có thông tin khách hàng trong bảng hóa đơn)
  - Tìm các khách hàng đã mua hàng
  - Lọc ra những khách hàng chưa mua hàng

```
SELECT Ma_khach_hang
FROM hoa_don
```

Ma_khach_hang
kh001
kh002
kh003
kh002

```
SELECT Ma_khach_hang
FROM khach_hang
WHERE Ma_khach_hang NOT IN
(SELECT Ma_khach_hang
FROM hoa_don)
```

Ma_khach_hang
kh004
kh005
kh006
kh008





## Truy vấn con

### ❑ Làm việc với toán tử so sánh

- Các toán tử so sánh thường được sử dụng trong truy vấn con có thể là:  $>$ ,  $>=$ ,  $<$ ,  $<=$ ,  $=$ ,  $<>$ .
- **Lưu ý:** Thông thường các toán tử so sánh được sử dụng khi dùng truy vấn con trả về một giá trị



LTV Mã nguồn mở PHP - Module 3

8



## Truy vấn con

### ❑ Làm việc với toán tử truy vấn con

- Các toán tử truy vấn con thường hay sử dụng là: ANY, SOME, ALL, IN, NOT IN, EXISTS, NOT EXISTS.
- Lưu ý: thông thường các toán tử truy vấn con được sử dụng khi dùng truy vấn con trả về tập hợp các giá trị.
- Quy tắc:  
 $IN \Leftrightarrow =ANY$   
 $NOT IN \Leftrightarrow <>ALL$



LTV Mã nguồn mở PHP - Module 3

9



## Truy vấn con

- Ví dụ: Viết lại câu lệnh truy vấn cho biết các khách hàng chưa mua hàng

```
SELECT Ma_khach_hang
FROM khach_hang
WHERE Ma_khach_hang <> ALL
(SELECT Ma_khach_hang
FROM hoa_don)
```

Ma\_khach\_hang

kh004

kh005

kh006

kh008



## Truy vấn con



### ❑ Làm việc với toán tử truy vấn con

- Sử dụng từ khóa EXISTS hoặc NOT EXISTS để kiểm tra tính tồn tại (hay không tồn tại) của dữ liệu.
- Sau EXISTS hoặc NOT EXISTS là câu lệnh truy vấn con mà kết quả trả về là một tập hợp trống hoặc có phần tử





## Truy vấn con

- Ví dụ: Tìm những khách hàng chưa mua hàng theo cách dùng NOT EXISTS

```
SELECT Ma_khach_hang
FROM khach_hang
WHERE NOT EXISTS (SELECT *
FROM hoa_don
WHERE khach_hang.Ma_khach_hang =
hoa_don.Ma_khach_hang)
```

Ma\_khach\_hang

kh004

kh005

kh006

kh008





## Insert

❑ Câu lệnh INSERT INTO được dùng để thêm mới một hay nhiều dòng dữ liệu vào bên trong một bảng

❑ Có thể thêm vào bảng:

- Giá trị trực tiếp
- Dữ liệu lấy từ các bảng khác

*(GV demo cách thêm dữ liệu bằng GD đồ học cho HV)*



## Insert

❑ Giá trị trực tiếp

- Cú pháp:

**INSERT INTO Tên\_bảng [(Ds các cột trong bảng)]  
VALUES (Danh\_sách\_các\_giá\_trị)**

- Ví dụ: Thêm mới một khách hàng có các thông tin sau vào bảng khách\_hang:

- Mã khách hàng: kh007, Họ tên: Phan Nam, Phái: 0 (nam), Địa chỉ: 12A – Pasteur – Q.1 – TP. HCM, Điện thoại: 8497625, Email: [phannam@yahoo.com](mailto:phannam@yahoo.com)





## Insert

### ● Ví dụ

```
INSERT INTO khách_hang(Ma_khach_hang,
Ten_khach_hang, Phai, Dia_chi, Dien_thoai, Email)
VALUES("kh001", "Phan Nam", 0, "12A – Pasteur – Q.1 – TP.
HCM", "8497625", "phannam@yahoo.com")
```

```
INSERT INTO khách_hang
VALUES("kh001", "Phan Nam", 0, "12A – Pasteur – Q.1 – TP.
HCM", "8497625", "phannam@yahoo.com")
```



## Insert

### ❑ Lấy dữ liệu từ các bảng khác

- Bằng cách kết hợp giữa INSERT và SELECT

- Cú pháp:

```
INSERT INTO Tên_bảng
[(Danh_sách_các_cột_cần_thêm_dữ_liệu)]
SELECT Danh_sách_các_cột_lấy_dữ_liệu
FROM Tên_bảng_nguồn
WHERE Điều_kiện_lọc_dữ_liệu
```

- Danh\_sách\_các\_cột\_cần\_thêm\_dữ\_liệu và Danh\_sách\_các\_cột\_lấy\_dữ\_liệu phải tương ứng nhau



## Insert

- Ví dụ: Thêm vào bảng ct\_hoadon\_07\_2013 (bảng chưa có dữ liệu) các chi tiết hóa đơn bán sữa của tháng 07/2013

```
INSERT INTO ct_hoadon_07_2013
SELECT ct_hoadon.So_hoa_don, Ma_sua, So_luong,
Don_gia
FROM ct_hoadon INNER JOIN hoa_don ON
ct_hoadon.So_hoa_don = hoa_don.So_hoa_don
WHERE month(Ngay_HD) = 7 AND year(Ngay_hd)=2013
```







## Update

### ❑ Giá trị trực tiếp

- Dùng câu lệnh UPDATE để cập nhật giá trị trực tiếp hay một biểu thức có giá trị trả về cho mẫu tin trong bảng

- Cú pháp:

UPDATE Tên\_bảng

SET Tên\_cột = Giá\_trị (hoặc biểu thức) [...]

WHERE Điều\_kiện\_cập\_nhật

*(GV demo cách cập nhật dữ liệu bằng GD đồ học cho HV)*



## Update



- Ví dụ: Cập nhật lại đơn giá cho sữa có tên là 'Abbott Grow' với đơn giá mới là 95000vnd

Tên_sua	Đơn_gia
Abbott Grow	87000

Đơn giá cũ

```
UPDATE sua
SET Đơn_gia = 95000
WHERE Tên_sua = "Abbott Grow"
```

Đơn giá mới

Tên_sua	Đơn_gia
Abbott Grow	95000





## Update

- Ví dụ: cập nhật lại đơn giá cho các sữa có mã hãng sữa là "AB" với đơn giá mới = đơn giá cũ + 5%

Ten_sua	Don_gia	Ma_hang_sua
Gain Advance	107000	AB
Gain IQ	107000	AB
Abbott Grow	87000	AB
Abbott Grow School	87000	AB
Abbott Pedia Sure	146000	AB
Similac Neo Sure	145000	AB

Đơn giá cũ

**UPDATE** sua

**SET** Don\_gia = Don\_gia + Don\_gia \* 0.05

**WHERE** Ma\_hang\_sua = "AB"

Đơn giá mới

Ten_sua	Don_gia	Ma_hang_sua
Gain Advance	112350	AB
Gain IQ	112350	AB
Abbott Grow	91350	AB
Abbott Grow School	91350	AB
Abbott Pedia Sure	153300	AB
Similac Neo Sure	152250	AB



LTV Mã nguồn mở PHP - Module 3

22

## Update



### ❑ Lấy dữ liệu từ các bảng khác

- Kết hợp giữa UPDATE và SELECT để lấy dữ liệu từ bảng khác cập nhật vào bảng
- Cú pháp:

**UPDATE** Tên\_bảng

**SET** Tên\_cột = (SELECT ... FROM ... WHERE ...)

**WHERE** Điều\_kiện\_cập\_nhật



LTV Mã nguồn mở PHP - Module 3

23

## Update

- Ví dụ: Cột đơn giá trong bảng ct\_hoadon chưa có giá trị, hãy cập nhật đơn giá cho cột này với giá trị được lấy từ cột đơn giá của bảng sửa.

```
UPDATE ct_hoadon
SET Don_gia = (SELECT Don_gia
FROM sua WHERE ct_hoadon.Ma_sua
= sua.Ma_sua)
```

Ma_hoa_don	Ma_sua	So_luong	Don_gia
D001	VNM012	4	103500
D001	AB0001	2	107000
D002	DL0001	2	41000
D002	MJ0004	3	198000
D001	DL0001	12	41000
D003	AB0001	8	107000
D004	NTF001	10	46500
D004	VNM012	8	103500
D003	DL0006	13	11500
D004	AB0001	15	107000
D002	MJ0001	5	196000
D001	NTF002	8	45000
D003	AB0003	17	87000
D004	AB0002	25	107000



## Delete

### ❑ Lệnh xóa dữ liệu đơn giản

- Cú pháp:

**DELETE FROM** Tên\_bảng

**WHERE** Điều\_kiện\_xóa

- Ví dụ: Hãy xóa khách hàng có mã khách hàng là “kh007” trong bảng khách hàng

```
DELETE FROM khách_hang
WHERE Ma_khach_hang = 'kh007'
```



## Delete

### ❑ Lệnh xóa dữ liệu có điều kiện được lấy từ các bảng khác

- Khi việc xóa phức tạp hơn vì có liên quan tới các quy tắc ràng buộc dữ liệu → kết hợp câu lệnh DELETE với SELECT
- Cú pháp:

**DELETE FROM** Tên\_bảng

**WHERE** Tên\_cột toán\_tử (SELECT ... FROM ... WHERE)



## Delete

- Ví dụ: xóa các hãng sữa không có sản phẩm sữa nào

```
DELETE FROM hang_sua  
WHERE Ma_hang_sua NOT IN (SELECT DISTINCT  
Ma_hang_sua FROM sua)
```





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

Module 3 – Bài 4: **Kết hợp PHP & MySQL – Thư viện PDO 1**

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014

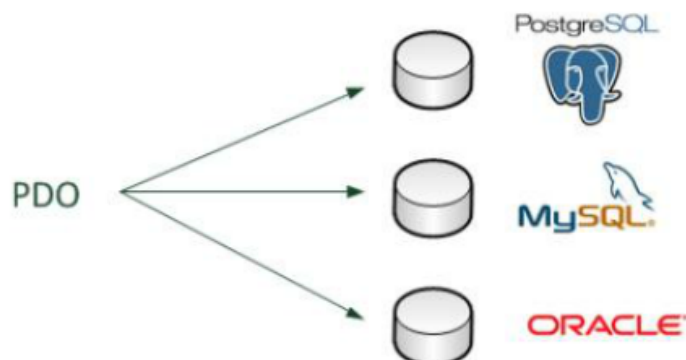




## Giới thiệu



- ❑ "PDO - PHP Data Objects – là một lớp truy cập dữ liệu (database access layer) cung cấp phương thức chuẩn để truy cập vào nhiều loại CSDL khác nhau."





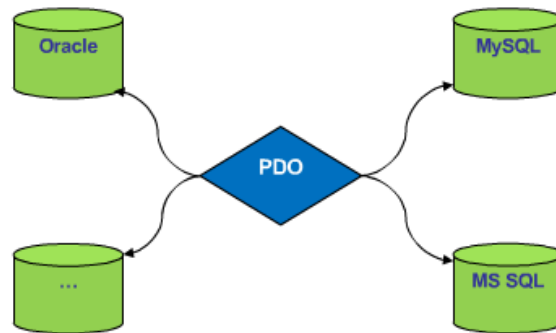
## Giới thiệu

- ❑ PDO cung cấp một **lớp truy cập dữ liệu trừu tượng** (data-access abstraction layer), nghĩa là, dù chúng ta đang sử dụng CSDL nào, chúng ta đều có thể sử dụng các phương thức như nhau để truy vấn và lấy dữ liệu.
- ❑ PDO không cung cấp một CSDL trừu tượng, vì vậy chúng ta cần phải sử dụng một lớp trừu tượng nếu chúng ta cần CSDL đó.



## Lý do chọn PDO

- ❑ **Hướng đối tượng: rất phù hợp để chúng ta tích hợp vào mô hình MVC (Rất nhiều framework lớn hiện nay đều sử dụng PDO)**



Mô hình làm việc giữa PDO và các loại CSDL

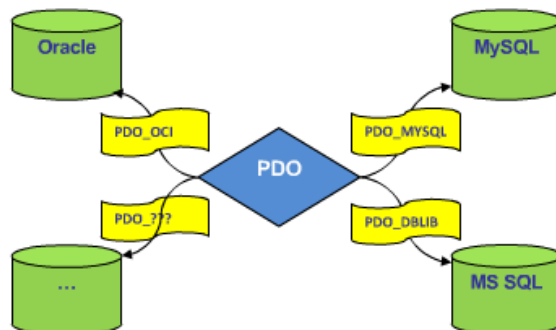


LTV Mã nguồn mở PHP - Module 3

6

## Lý do chọn PDO

- ❑ **Nhanh & đơn giản: việc thao tác với từng loại csdl sẽ do từng driver tương ứng đảm nhiệm**



Mô hình làm việc giữa PDO và các loại CSDL

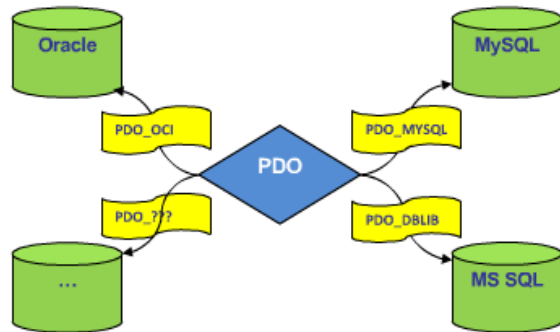


LTV Mã nguồn mở PHP - Module 3

7

## Lý do chọn PDO

- Linh động: dễ chuyển đổi sang CSDL khác ta chỉ việc thay đổi tên driver tương ứng.

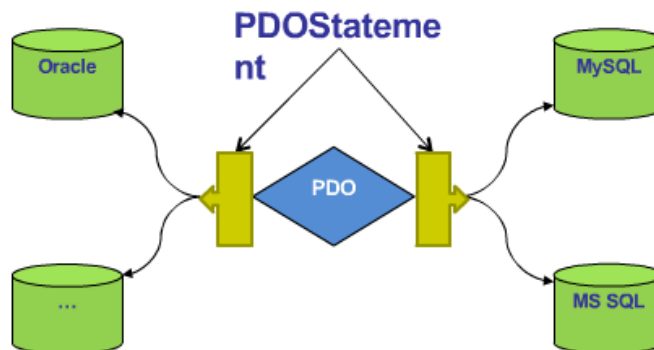


Mô hình làm việc giữa PDO và các loại CSDL



## Lý do chọn PDO

- An toàn: với giải pháp PDOStatement chúng ta không còn lo sợ SQL injection nữa.



Mô hình làm việc giữa PDO và các loại CSDL





## Tạo kết nối – hủy kết nối



### □ Tạo kết nối

- Cú pháp:

```
$dbh = new PDO(<driver>, <username>,  
<password>, <driver_options>);
```

- Tham số:

- <driver>: tên driver dùng để làm việc CSDL.
- <username>: tên đăng nhập vào CSDL.
- <password>: mật khẩu dùng để truy cập vào CSDL.
- <driver\_options>: tham số tùy chọn, dùng để bổ sung thêm các trạng thái kết nối của PDO.





## Tạo kết nối – hủy kết nối

### ❑ Tạo kết nối

- Server hỗ trợ khá nhiều driver:
  - PDO\_DBLIB (Microsoft SQL Server)
  - PDO\_IBM (IBM DB2)
  - PDO\_MYSQL (MySQL 3.x/4.x/5.x)
  - PDO\_OCI (Oracle Call Interface)
  - PDO\_PGSQL ( PostgreSQL )
  - ...
- Để xem server hỗ trợ những loại driver nào:

```
<?php
    print_r(PDO::getAvailableDrivers());
?>
```



## Tạo kết nối – hủy kết nối

### ❑ Tạo kết nối

- Ví dụ: Kết nối đến MySQL

```
$dbh = new
PDO('mysql:host=localhost;dbname=test',
'root', '');
```

- Ví dụ: Kết nối đến MSSQL

```
$dbh = new
PDO('dblib:host=localhost;dbname=test', 'root',
'');
```





## Tạo kết nối – hủy kết nối

### □ Tạo kết nối

- Ví dụ: Kết nối đến Oracle

```
$tns = "(DESCRIPTION =
        (ADDRESS_LIST =
            (ADDRESS = (PROTOCOL = TCP)(HOST =
yourip)(PORT = 1521))
        )
        (CONNECT_DATA = (SERVICE_NAME = orcl)
        )
)";
```

```
$dbh= new PDO("oci:dbname=".$tns, 'root', "');
```



## Tạo kết nối – hủy kết nối

### □ Hủy kết nối

- Cú pháp

```
$dbh = NULL;
```





## Tạo kết nối – hủy kết nối

### ❑ Xử lý lỗi kết nối

```
try {  
    $dbh = new PDO('mysql:host=localhost;dbname=test', 'root', '');  
    ...  
    $dbh = null;  
} catch (PDOException $e) {  
    print "Error!: " . $e->getMessage() . "<br/>";  
    die();  
}
```







## Thực thi lệnh SQL

### □ Câu lệnh Insert, update, delete

- Cú pháp

**`$dbh->exec(<query>);`**

- Tham số

- query: tham số bắt buộc, là câu lệnh truy vấn được gửi đi
- \$dbh->exec(): Kết quả là tổng số dòng bị ảnh hưởng. Nếu không có dòng nào bị tác động thì kết quả là 0. Nếu không thực thi được câu lệnh thì kết quả là False



## Thực thi lệnh SQL

- Ví dụ: Thực hiện lệnh xóa tất cả các dòng dữ liệu trong bảng hang\_sua





## Thực thi lệnh SQL

### □ Câu lệnh Select

- Cú pháp:

**`$dbh->query(<query>);`**

- Tham số:

- **query:** tham số bắt buộc, là câu lệnh truy vấn được gửi đi



## Thực thi lệnh SQL

- Ví dụ: Thực hiện lệnh lấy thông tin loại sữa







Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

Module 3 – Bài 5: *Kết hợp PHP & MySQL – Thư viện PDO 2*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





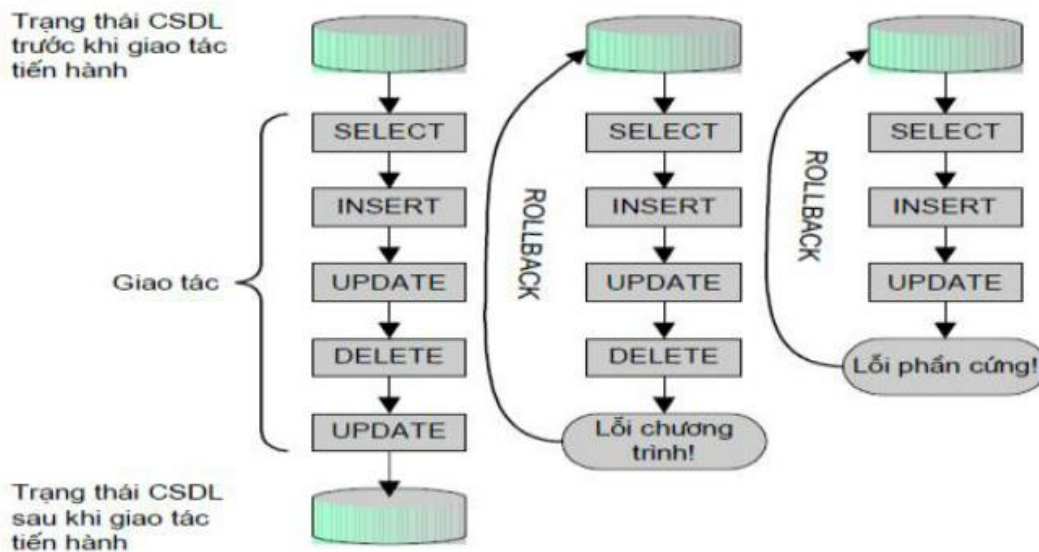
## Transaction



- ☐ Một giao tác (transaction) là một chuỗi một hoặc nhiều câu lệnh SQL được kết hợp lại với nhau thành một khối công việc. Việc kết hợp các câu lệnh lại với nhau trong một giao tác nhằm đảm bảo tính toàn vẹn dữ liệu và khả năng phục hồi dữ liệu.
  
- ☐ Trong một giao tác, các câu lệnh có thể độc lập với nhau nhưng tất cả các câu lệnh trong một giao tác đòi hỏi phải thực thi trọn vẹn hoặc không một câu lệnh nào được thực thi.



## Transaction



Hình 6.1: Giao tác SQL



LTV Mã nguồn mở PHP - Module 3

4

## Transaction

### ❑ Giao tác SQL được định nghĩa dựa trên các câu lệnh xử lý giao tác sau đây:

- **BEGIN TRANSACTION:** Bắt đầu một giao tác
- **COMMIT:** Đánh dấu kết thúc giao tác. Câu lệnh này báo hiệu sự kết thúc thành công của một giao tác. Sau câu lệnh này, một giao tác mới sẽ được bắt đầu.
- **ROLLBACK:** Hủy bỏ một giao tác và đưa CSDL về trạng thái như trước khi giao tác bắt đầu. Một giao tác mới sẽ bắt đầu sau khi câu lệnh ROLLBACK được thực thi



LTV Mã nguồn mở PHP - Module 3

5

## Transaction

---



□ Ví dụ



## PDO Statement



### ❑ Prepared statement

- Sử dụng prepared statement sẽ giúp chúng ta tránh SQL injection.

- Cú pháp

```
$sth = $dbh->prepare(<query>);
```

- Tham số:

- query: tham số bắt buộc, là câu lệnh truy vấn được gửi đi

- Ví dụ:

```
$sth = $dbh->prepare("SELECT * FROM sua");
```



## PDO Statement



### ❑ Thực thi



- Cú pháp

```
$sth->execute();
```





## PDO Statement

### ❑ Duyệt dữ liệu

- Duyệt lần lượt từng mẫu tin

- Cú pháp:

**PDOStatement->fetch([int \$fetch\_style])**

- Dữ liệu được lấy thông qua phương thức fetch(). Trước khi gọi fetch(), cần phải cho PDO biết các dữ liệu muốn lấy thông qua tham số tùy chọn \$fetch\_style:



LTV Mã nguồn mở PHP - Module 3

10

## PDO Statement

- Có các \$fetch\_style sau:

- **PDO::FETCH\_NUM:** Kết quả trả về một mảng (có chỉ số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng \$row[0], \$row[1], \$row[2],...
  - **PDO::FETCH\_ASSOC:** Kết quả trả về là một mảng (có chỉ số chuỗi) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng: \$row["tên cột 1"], \$row["tên cột 2"], \$row["tên cột 3"],...



LTV Mã nguồn mở PHP - Module 3

11



## PDO Statement

- Có các `$fetch_style` sau:
  - **PDO::FETCH\_BOTH**: kết quả trả về là một mảng (có chỉ số chuỗi và số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng: `$row["tên cột 1"]`, `$row["tên cột 2"]`, `$row["tên cột 3"]`,... hoặc `$row[0]`, `$row[1]`, `$row[2]`,...
  - **PDO::FETCH\_OBJ**: Kết quả trả về là một mẫu tin trong bộ các mẫu tin như là một đối tượng → truy cập bằng cách gọi từng thuộc tính của đối tượng: `$tên_đối_tượng` → `tên_cột_1`, `$tên_đối_tượng` → `tên_cột_2`,...



## PDO Statement

### ❑ Duyệt dữ liệu

- Duyệt lần lượt từng mẫu tin
  - Ví dụ:





## PDO Statement

### ❑ Duyệt dữ liệu

#### ● Duyệt danh sách các mẫu tin

**`PDOStatement->fetchAll([int $fetch_style, $column_index])`**

##### ▪ Tham số:

##### ▪ **`$fetch_style`:**

- *PDO::FETCH\_BOTH*: giá trị mặc định. Kết quả trả về là mảng hai chiều, mỗi phần tử là một mảng 1 chiều (có chỉ số chuỗi và số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột.
- *PDO::FETCH\_COLUMN*: Kết quả trả về là mảng 1 chiều chứa tất cả các giá trị của một cột được chỉ định

##### ▪ **`$column_index`:** kết quả trả về là cột được chỉ định khi `$fetch_style` có giá trị là *PDO::FETCH\_COLUMN*



LTV Mã nguồn mở PHP - Module 3

14

## PDO Statement



#### ● Ví dụ: In giá trị của cột đầu tiên trong bảng khách hàng



LTV Mã nguồn mở PHP - Module 3

15



## PDO Statement

### ❑ Truyền tham số trong câu truy vấn

#### ● Cú pháp:

`PDOStatement->bindParam(mixed $parameter, mixed &$variable [, int $data_type [, int $length [, mixed $driver_options]]])`

#### ● Tham số:

- \$parameter: tên tham số
- \$variable: giá trị tham số
- \$data\_type: tùy chọn, kiểu dữ liệu của tham số, có cú pháp PDO::PARAM\_\*
- \$length: tùy chọn, chiều dài của kiểu dữ liệu
- \$driver\_options



LTV Mã nguồn mở PHP - Module 3

16

## PDO Statement



- Ví dụ: Liệt kê danh sách sữa có ma\_hang\_sua là AB và đơn giá nhỏ hơn 100000



LTV Mã nguồn mở PHP - Module 3

17



## PDO Statement

- Ví dụ: Liệt kê danh sách sữa có ma\_hang\_sua là AB và đơn giá nhỏ hơn 100000



## PDO Statement

### ❑ Đếm số lượng mẫu tin

- Cú pháp:

**PDOStatement->rowCount()**

- Hàm này có kết quả trả về là số lượng mẫu tin bị tác động bởi câu lệnh insert, update, delete
- Ví dụ: Đếm số lượng mẫu tin bị xóa trong bảng sua có ma\_hang\_sua="AB"





LTV Mã nguồn mở PHP - Module 3

20



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

Module 3 – Bài 6: *Hiển thị dữ liệu trên trang Web*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Hiển thị dữ liệu theo yêu cầu



### □ Định dạng hiển thị dữ liệu

- Cách đơn giản nhất là hiển thị dữ liệu dạng bảng hoặc dạng text
- Ví dụ: Hiển thị thông tin khách hàng

THÔNG TIN KHÁCH HÀNG

Mã KH	Tên khách hàng	Giới tính	Địa chỉ	Số điện thoại
kh001	Khuất Thủy Phương	1	A21 Nguyễn Oanh quận Gò Vấp	9874125
kh002	Đỗ Lâm Thiên	0	357 Lê Hồng Phong Q.10	8351056
kh003	Phạm Thị Nhung	1	56 Đinh Tiên Hoàng quận 1	9745698
kh004	Nguyễn Khắc Thiện	0	12bis Đường 3-2 quận 10	8769128
kh005	Tô Trần Hồ Giáng	0	75 Nguyễn Kiệm quận Gò Vấp	5792564
kh006	Nguyễn Kiến Thi	1	357 Lê Hồng Phong Q.10	9874125
kh008	Nguyễn Anh Tuấn	0	1/2bis Nơ Trang Long Q.BT TP.HCM	8753159





## Hiển thị dữ liệu theo yêu cầu

### □ Định dạng hiển thị dữ liệu

- Yêu cầu đơn giản nhất của một trang Web là dữ liệu hiển thị phải được định dạng giúp cho người dùng dễ dàng đọc thông tin.
- Các định dạng thông thường là định dạng cột, dòng, kích thước, màu sắc, kiểu chữ,... cho nội dung hiển thị.
- Để thực hiện được các yêu cầu về định dạng → cần phải xem xét và tính toán cho các nội dung cần hiển thị.



LTV Mã nguồn mở PHP - Module 3

4

## Hiển thị dữ liệu theo yêu cầu

- Ví dụ: hiển thị thông tin khách hàng với định dạng như sau:
  - Dòng tiêu đề của bảng sẽ được canh giữa, chữ in đậm, màu sắc phân biệt so với các dòng khác.
  - Các dòng thông tin của khách hàng: dòng chẵn và dòng lẻ có màu nền khác nhau, cột giới tính được canh giữa

Dòng tiêu đề

### THÔNG TIN KHÁCH HÀNG

Mã KH	Tên khách hàng	Giới tính	Địa chỉ	Số điện thoại
kh001	Khuất Thủy Phương	1	A21 Nguyễn Oanh quận Gò Vấp	9874125
kh002	Đỗ Lâm Thiên	0	357 Lê Hồng Phong Q.10	8351056
kh003	Phạm Thị Nhung	1	56 Đinh Tiên Hoàng quận 1	9745698
kh004	Nguyễn Khắc Thiện	0	12bis Đường 3-2 quận 10	8769128
kh005	Tô Trần Hồ Giảng	0	75 Nguyễn Kiệm quận Gò Vấp	5792564
kh006	Nguyễn Kiên Thi	1	357 Lê Hồng Phong Q.10	9874125
kh008	Nguyễn Anh Tuấn	0	1/2bis Nơ Trang Long Q.BT TP.HCM	8753159

Dòng lẻ

Dòng chẵn



LTV Mã nguồn mở PHP - Module 3

5



## Hiển thị dữ liệu theo yêu cầu

### □ Tạo các cột tùy biến

- Nhu cầu hiển thị dữ liệu một cách rõ ràng, chi tiết, dễ hiểu trở thành một yêu cầu tất yếu khi thiết kế trang Web.
- Một trong những cách hiển thị dữ liệu đáp ứng nhu cầu đó là tạo các cột tùy biến



LTV Mã nguồn mở PHP - Module 3

6



## Hiển thị dữ liệu theo yêu cầu

- Ví dụ: trong CSDL thường dùng kiểu dữ liệu boolean cho cột giới tính → cột này sẽ có một trong hai giá trị là 0 hoặc 1 và tùy theo quy ước để xác định 0 hoặc 1 thuộc giới tính gì. Nếu cột giới tính hiển thị 0, 1 và sau đó lại có dòng chú giải thì sẽ không thuận tiện → dùng hình ảnh hoặc checkbox,... để hiển thị giới tính. Cách hiển thị này vừa đẹp mắt, vừa dễ hiểu.



LTV Mã nguồn mở PHP - Module 3

7



## Hiển thị dữ liệu theo yêu cầu

### ❑Liên kết trang có chuỗi tham số

- Chuỗi URL có cấu trúc định dạng như sau:

`http://<host>[:<port>][path][?chuỗi_tham_số]`

- Phần cuối của chuỗi URL là chuỗi tham số có cú pháp như sau:

`[?Tham_số_1=Giá_trị_1[&Tham_số_2=Giá_trị_2[&...]]]`

- Chú ý: trong trường hợp có nhiều tham số, các cặp [Tham\_số=Giá\_trị] phân cách nhau bằng dấu &



LTV Mã nguồn mở PHP - Module 3

8

## Hiển thị dữ liệu theo yêu cầu



### ❑Liên kết trang có chuỗi tham số

- Ví dụ: Khi người dùng muốn xem thông tin chi tiết của món ăn, người dùng click chọn món ăn đó, URL sẽ có đường dẫn là địa chỉ của trang chi tiết món ăn và chuỗi tham số là mã món

Trang danh sách món ăn



`http://localhost:81/nha_hang/mon_an/chi_tiet_mon.php?ma_mon=MM48`

Trang chi tiết món ăn



LTV Mã nguồn mở PHP - Module 3

9



## Hiển thị dữ liệu theo yêu cầu

### ❑ Liên kết trang có chuỗi tham số

- Nhận giá trị truyền qua tham số:

`$_REQUEST["tên_tham_số"]`

- Ví dụ: nhận giá trị của tham số mã món

```
$ma_mon = $_REQUEST["ma_mon"];
```





## Phân trang

### □ Giới thiệu

- Đối với những bảng có số lượng mẫu tin lớn khi hiển thị dữ liệu sẽ làm cho người dùng cảm thấy khó xem → thực hiện việc phân trang.
- Phân trang dữ liệu giúp cho việc xem và tìm kiếm thông tin dễ dàng, nhanh chóng và thuận tiện.



LTV Mã nguồn mở PHP - Module 3

12

## Phân trang



### □ Lớp Pager

- Sử dụng class pager được cung cấp để thực hiện việc phân trang
- Với lớp pager, chúng ta có thể:
  - Thiết lập loại phân trang và cách hiển thị
  - Chọn số lượng mẫu tin hiển thị trên 1 trang

***GV mở file chứa lớp Pager để giải thích cho HV các phương thức và ý nghĩa của các phương thức này.***



LTV Mã nguồn mở PHP - Module 3

13



## Phân trang

### □ Các dạng phân trang



**GV demo việc thực hiện phân trang cho HV bằng 1 ví dụ cụ thể**



LTV Mã nguồn mở PHP - Module 3

14



LTV Mã nguồn mở PHP - Module 3

15



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

### Module 3 – Bài 7: *PHP Email & Secure Email*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## PHP Email



### □ Giới thiệu

- Để gửi Email bằng PHP, cần phải có một Web Server cho hệ thống quản lý Email gọi là Mail Server.
- Khi cài đặt một Mail Server, cần phải có một server và một trong các phần mềm Mail Server như MS Exchange Server, Mdeamon, Apache...
- Khi cài đặt thành công Mail Server, ta có thể cấu hình nhiều địa chỉ Email cho nhiều người sử dụng là nhân viên trong công ty hay khách hàng.
- Với mỗi Email được gửi qua Internet, cần phải có địa chỉ Email người gửi và người nhận trong đó địa chỉ Email người gửi phải là địa chỉ Email tồn tại trong Mail Server.





## PHP Email



### □ Phương thức

- Phương thức mail() được sử dụng để gửi mail
- Cú pháp:  
`mail(to,subject,message,headers,parameters)`



LTV Mã nguồn mở PHP - Module 3

4

## PHP Email



### □ Phương thức

- Tham số:
  - to: tham số bắt buộc, là địa chỉ của người nhận mail
  - subject: tham số bắt buộc, là tiêu đề của email. Chú ý: trong tiêu đề của email không chứa ký tự xuống dòng
  - message: tham số bắt buộc, là nội dung được gửi đi. Chú ý: Mỗi dòng nên được tách ra với một LF – Line Feed (\n). Mỗi dòng không được vượt quá 70 ký tự.
  - headers: tham số tùy chọn, là các phần bổ sung vào header như: From, Cc, và Bcc. Chú ý: Các phần bổ sung nên được tách ra với một CRLF - Carriage Return Line Feed (\r\n)
  - parameters: tham số tùy chọn, chỉ định một tham số bổ sung cho chương trình sendmail



LTV Mã nguồn mở PHP - Module 3

5

## PHP Email

### □ Gửi email đơn giản

- Cách đơn giản nhất để gửi một email với PHP là gửi một email dạng văn bản.
- Ví dụ: lần lượt khai báo và gán giá trị cho các biến: \$to, \$subject, \$message, \$from, \$headers, sau đó sử dụng phương thức mail () để gửi e-mail:

```
$to = "tuvan@alphatek.edu.vn";
$subject = "Hỏi lịch khai giảng khóa mới";
$message = "Xin chào Tư vấn! Vui lòng gửi cho mình lịch khai giảng khóa mới. Cảm ơn";
$from = "nguyen.anh@gmail.com";
$headers = "From:" . $from;
mail($to,$subject,$message,$headers);
echo "Thư đã được gửi đi.";
```



## PHP Email

### □ Mail form

- Tạo một Mail form để thực hiện công việc gửi mail như sau:

**Gửi email bằng PHP**

Địa chỉ email người gửi:

Đến địa chỉ email người nhận:

Tiêu đề:

Nội dung:



## Mail form

```
if (isset($_REQUEST['to']) && isset($_REQUEST['subject']) &&
    isset($_REQUEST['message']) && isset($_REQUEST['from'])) {
    //khi người dùng điền đầy đủ thông tin -> gửi mail
    $to = $_REQUEST['to'];
    $subject = $_REQUEST['subject'];
    $message = $_REQUEST['message'];

    $from = $_REQUEST['from'];
    $headers = "From:" . $from;
    mail($to,$subject,$message,$headers);
    echo "Cảm ơn bạn đã sử dụng mail form";
} else {
    echo "Vui lòng nhập đầy đủ thông tin";
}
```





## Secure Email

### ❑ Đặt vấn đề

- Khi gửi mail trong PHP có một yếu điểm là người sử dụng trái phép có thể chèn dữ liệu vào mail headers thông qua việc nhập vào form.
- Và việc gì sẽ xảy ra nếu người dùng thêm các văn bản sau đây vào “from” input field trên form?

```
someone@gmail.com
%0ACc:person2@gmail.com
%0ABcc:person3@gmail.com,person3@gmail.com,
anotherperson4@gmail.com,person5@gmail.com
%0ABTo:person6@gmail.com
```



LTV Mã nguồn mở PHP - Module 3

10



## Secure Email

### ❑ Đặt vấn đề

- Như thường lệ, phương thức mail () đưa nội dung trên vào email headers, và bây giờ thì header có thêm Cc:, Bcc:, và: To. Khi người dùng nhấp vào nút submit, e-mail sẽ được gửi đến tất cả các địa chỉ trên!



LTV Mã nguồn mở PHP - Module 3

11



## Secure Email

### ❑ Cách giải quyết

- Cách tốt nhất để ngăn chặn việc gửi mail như trên là cần phải xác nhận dữ liệu đầu vào.
- Đoạn code ở trang tiếp theo giống như ở phần PHP email ta đã thực hiện, nhưng bây giờ được thêm vào một validator để kiểm tra “from” input field trên form:



## Secure Email



### ❑ Cách giải quyết

- Xây dựng phương thức spamcheck() có sử dụng PHP Filter để xác nhận dữ liệu đầu vào:

```
function spamcheck($field) {
    // FILTER_SANITIZE_EMAIL filter: loại bỏ tất cả các ký tự email bất hợp pháp e-mail từ chuỗi
    $field=filter_var($field, FILTER_SANITIZE_EMAIL);
    // FILTER_VALIDATE_EMAIL filter: xác nhận giá trị là một địa chỉ e-mail
    if(filter_var($field, FILTER_VALIDATE_EMAIL)){
        return TRUE;
    }
    } else {
        return FALSE;
    }
}
```





## Secure Email

### □ Cách giải quyết

- Gọi sử dụng phương thức này vào PHP Email

```
if (isset($_REQUEST['to']) && isset($_REQUEST['subject']) &&
    isset($_REQUEST['message']) && isset($_REQUEST['from'])) {
    $mailcheck = spamcheck($_REQUEST['from']);
    if ($mailcheck==FALSE){
        echo "Dữ liệu nhập không hợp lệ";
    } else {
        //khi người dùng điền đầy đủ thông tin và dữ liệu nhập hợp lệ -> gửi mail
        $to = $_REQUEST['to'];
        $subject = $_REQUEST['subject'];
        $message = $_REQUEST['message'];

        $from = $_REQUEST['from'];
        $headers = "From:" . $from;
        mail($to,$subject,$message,$headers);
        echo "Cảm ơn bạn đã sử dụng mail form";
    } else {
        echo "Vui lòng nhập đầy đủ thông tin";
    }
}
```



LTV Mã nguồn mở PHP - Module 3

14



LTV Mã nguồn mở PHP - Module 3

15



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

### Module 3 – Bài 8: *Smarty – Template Engine 1*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Giới thiệu



- ❑ Smarty là một hệ thống tạo mẫu web (web template system) được viết trên nền PHP, nó tạo điều kiện cho việc tách phần trình bày ra khỏi phần code.
- ❑ Smarty là một công cụ dùng cho việc chia nhỏ các quá trình thiết kế trang web. Đây là một chiến lược thiết kế cho các ứng dụng web hiện đại.
- ❑ Smarty là kết quả của việc tối ưu hóa và phân luồng công việc, cho phép trình diễn các trang web bằng cách chia nhỏ chúng ở đầu cuối (back-end). Ý tưởng đó đã giúp giảm thiểu rất nhiều giá thành và sự đồ sộ của các phần mềm. Dưới mỗi một ứng dụng thành công thì luôn phân chia thành 2 mảng là thiết kế, chính là lá chắn bảo vệ mã nguồn (back-end coding) và lập trình php được bảo vệ bởi code.







## Giới thiệu

- ❑ Smarty tạo ra các nội dung web từ các vị trí đặc biệt được gọi là smarty tag .
- ❑ Các tag này được nằm trong các mẫu (template), sẽ được xử lý và chèn vào các code khác nhau.
- ❑ Các tag này chỉ thị cho Smarty, được bao đóng bởi tag mở và tag khóa. Những nội dung được chèn vào code, và được định nghĩa bởi dấu dollar (\$), các chức năng, hàm logic hoặc các luồng điều khiển.
- ❑ Smarty cho phép các lập trình viên có thể tự định nghĩa các chức năng và truy cập sử dụng các Smarty tag.



## Giới thiệu

### ❑ Hai cách tiếp cận template

- Cách thứ nhất: cho rằng “PHP là template engine”. Cách tiếp cận này đơn giản chỉ là trộn lẫn mã PHP với thẻ HTML. Mặc dù cách này khi thực hiện rất nhanh chóng nhưng nhiều người cho rằng cách này sẽ gây khó khăn cho việc duy trì ứng dụng. PHP làm công việc lập trình rất tốt nhưng không vì thế mà nó tốt cho việc trình bày.
- Cách thứ hai cho rằng cần phải tách biệt giữa code và template, đây là cách tiếp cận mà Smarty thực hiện.





## Giới thiệu

### ❑ Mục tiêu:

- Tách phần trình bày khỏi phần code của ứng dụng
- PHP là back-end, còn Smarty template là front-end
- Hỗ trợ cho PHP chứ không thay thế PHP
- Giúp cho lập trình viên và nhà thiết kế có thể nhanh chóng triển khai/phát triển ứng dụng
- Dễ thực hiện và duy trì
- Dễ tổ chức và quản lý
- Cú pháp dễ hiểu, không đòi hỏi phải có kiến thức PHP
- Tính năng đầy đủ, dễ dàng mở rộng tùy nhu cầu của người dùng
- Linh hoạt trong việc tùy chỉnh ứng dụng
- Bảo mật tốt hơn vì tách biệt với PHP
- Mã nguồn mở, miễn phí



## Giới thiệu

### ❑ Lý do chọn Smarty

- Smarty thực hiện công việc đưa PHP vào phần hiển thị với một cú pháp rất đơn giản: tag-based syntax – cú pháp dựa trên thẻ. Các thẻ chứa nội dung của ứng dụng, thực hiện một cách tách bạch khỏi code PHP.
- Không cần phải có kiến thức về PHP cũng có thể quản lý được các template Smarty.
- Bên cạnh cú pháp dựa trên thẻ rõ ràng, Smarty còn cung cấp một loạt các công cụ để quản lý phần hiển thị: bộ nhớ đệm, kế thừa template... Các yêu cầu chức năng và PHP code Smarty đang được sử dụng sẽ đóng một vai trò to lớn trong việc xác định sự phù hợp của Smarty.





## Giới thiệu

### ❑ Lý do chọn Smarty

- Trong nhiều trường hợp, khi việc quản lý template hiệu quả là việc rất quan trọng, hoặc trong trường hợp các nhà thiết kế web (chứ không phải lập trình web) quản lý các template, sức mạnh của việc sử dụng template dựa trên thẻ nhanh chóng được nhận ra.
- Nếu các dự án có quy mô template lớn (hàng trăm template) thì việc kế thừa template sẽ giữ cho các mẫu được bảo trì và sắp xếp hợp lý.
- Smarty có thêm kỹ thuật (a tag-based syntax + 1-time compile step – một cú pháp thẻ + một bước biên dịch) giúp dễ dàng khắc phục bằng cách tiết kiệm thời gian bảo dưỡng template.



## Giới thiệu

### ❑ Cách thức làm việc

- Smarty biên dịch các bản sao của các template như các script PHP. Bằng cách này ta có các thuận lợi của cả cú pháp thẻ mẫu và tốc độ của PHP.
- Biên dịch xảy ra một lần khi mỗi template lần đầu tiên được gọi, và sau đó các phiên bản đã được biên dịch sẽ được sử dụng từ thời điểm đó. Smarty sẽ thực hiện công việc này, do đó các nhà thiết kế chỉ cần chỉnh sửa các template Smarty mà không cần phải quản lý các phiên bản đã được biên dịch.
- Cách tiếp cận này giúp cho các template dễ dàng duy trì, và vẫn giữ thời gian thực hiện nhanh chóng.



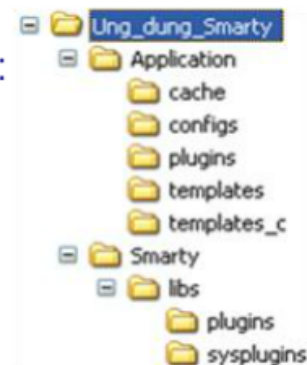


## Xây dựng ứng dụng đầu tiên với Smarty



### □ Các bước thực hiện

- Bước 1: Download thư viện Smarty mới nhất tại địa chỉ: <http://www.smarty.net/download>
- Bước 2: Tạo một thư mục để chứa ứng dụng và giải nén Smarty vào
- Bước 3: Xây dựng cây thư mục sau:





## Xây dựng ứng dụng đầu tiên với Smarty

- Bước 4: Tạo file template - application\templates\index.tpl
  - Ví dụ: file template có nội dung như sau: Nội dung là xuất ra câu chào với một tên được lấy từ code PHP vào

```
{* Smarty *}
Hello {$name}, wellcome to Smarty!
```



## Xây dựng ứng dụng đầu tiên với Smarty

- Bước 5: Tạo file code PHP – application\index.php

```
// chèn thư viện Smarty
require_once('../Smarty/libs/Smarty.class.php');

// khai báo lớp Smarty
$smarty = new Smarty();

// thiết lập đường dẫn đến các thư mục
$smarty->template_dir = '../application/templates/';
$smarty->compile_dir = '../application/templates_c/';
$smarty->config_dir = '../application/configs/';
$smarty->cache_dir = '../application/cache/';

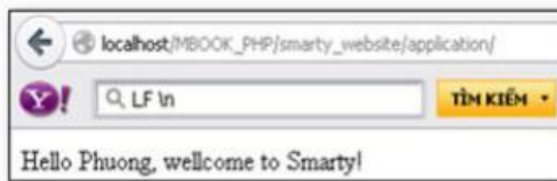
// viết code
$name = 'Phuong';
$smarty->assign('name', $name);

// hiển thị nội dung của file template đã chuẩn bị
$smarty->display('index.tpl');
```



## Xây dựng ứng dụng đầu tiên với Smarty

- Bước 6: Thực thi ứng dụng
  - Chạy ứng dụng trên local host và xem kết quả







## Smarty Template

### □Cú pháp cơ bản của Smarty

- Smarty hỗ trợ một vài cú pháp lập trình cấp cao, bao gồm:
  - Biến, hằng
  - if, elseif, else
  - Điều khiển luồng dữ liệu, foreach
  - Hàm do người dùng tự định nghĩa
  - ...



## Smarty Template

### □Cú pháp cơ bản của Smarty

Nội dung	PHP	Smarty
Mở & đóng thẻ PHP	<code>&lt;?php ?&gt; hoặc &lt;? ?&gt;</code>	<code>{ }</code>
Ghi chú	<code>//ghi chú một dòng hoặc /*ghi chú nhiều dòng */</code>	<code>{* ghi chú một dòng*} hoặc {* ghi chú nhiều dòng *}</code>
Hiển thị biến	<code>&lt;?php echo \$tên_biến; ?&gt;</code> Ví dụ: <code>&lt;?php echo \$name; ?&gt;</code>	<code>{ \$tên_biến }</code> Ví dụ: <code>{ \$name }</code>
Hiển thị các biến \$_GET \$_POST \$_COOKIE \$_SERVER \$_ENV \$_SESSION	<code>&lt;?php echo \$_GET['page']; ?&gt;</code> <code>&lt;?php echo \$_POST['page']; ?&gt;</code> <code>&lt;?php echo \$_COOKIE['username']; ?&gt;</code> <code>&lt;?php echo \$_SERVER['SERVER_NAME']; ?&gt;</code> <code>&lt;?php echo \$_SESSION['id']; ?&gt;</code>	<code>{ \$smarty.get.page }</code> <code>{ \$smarty.post.page }</code> <code>{ \$smarty.cookies.username }</code> <code>{ \$smarty.server.SERVER_NAME }</code> <code>{ \$smarty.session.id }</code>





## Smarty Template

### □Cú pháp cơ bản của Smarty

Nội dung	PHP	Smarty
Hiển thị giá trị một phần tử của mảng	<code>&lt;?php echo \$tên_mảng[vị_trí]; ?&gt;</code> Ví dụ: <code>&lt;?php echo \$foo[5]; ?&gt;</code>	<code>{ \$tên_mảng[vị_trí] }</code> Ví dụ: <code>{ \$foo[5] }</code>
Hiển thị giá trị của khóa trong mảng	<code>&lt;?php echo \$tên_mảng['tên_khóa']; ?&gt;</code> Ví dụ: <code>&lt;?php echo \$foo['bar']; ?&gt;</code>	<code>{ \$tên_mảng.tên_khóa }</code> Ví dụ: <code>{ \$foo.bar }</code>
Hiển thị giá trị của phần tử trong mảng	<code>&lt;?php echo \$tên_mảng[\$tên_khóa]; ?&gt;</code> Ví dụ: <code>&lt;?php echo \$foo[\$bar]; ?&gt;</code>	<code>{ \$tên_mảng.\$tên_khóa }</code> Ví dụ: <code>{ \$foo.\$bar }</code>



## Smarty Template

### □Cú pháp cơ bản của Smarty

Nội dung	PHP	Smarty
Duyệt mảng dùng vòng lặp foreach	<code>&lt;?php foreach(\$tên_mảng as \$tên_phần_tử): ?&gt;</code> ... <code>&lt;?php endforeach; ?&gt;</code>  Ví dụ: <code>&lt;?php foreach(\$foo as \$bar): ?&gt;</code> ... <code>&lt;?php endforeach; ?&gt;</code>	<code>{foreach \$tên_mảng as \$tên_phần_tử}</code> ... <code>{/foreach}</code>  Ví dụ <code>{foreach \$foo as \$bar}</code> ... <code>{/foreach}</code> Hoặc <code>{foreach from=\$foo item=bar}</code> ... <code>{/foreach}</code>







## Smarty Template

### □Cú pháp cơ bản của Smarty

Nội dung	PHP	Smarty
In cả khóa và giá trị của phần tử trong mảng	<pre>&lt;?php foreach(\$tên_mảng as \$khóa=&gt;\$giá_trị): ?&gt; echo \$khóa . ":" . \$giá_trị; &lt;?php endforeach; ?&gt;</pre> <p>Ví dụ:</p> <pre>&lt;?php foreach(\$myArray as \$k =&gt; \$v): ?&gt; echo \$k. ":" . \$v; &lt;?php endforeach; ?&gt;</pre>	<pre>{foreach from=\$tên_mảng key=khóa item=gia_trị} {\$khóa}: {\$gia_trị} {/foreach}</pre> <p>Ví dụ:</p> <pre>{foreach from=\$myArray key=k item=v} &lt;li&gt;{\$k}: {\$v}&lt;/li&gt; {/foreach}</pre>



## Smarty Template

### □Cú pháp cơ bản của Smarty

Nội dung	PHP	Smarty
Duyệt mảng dùng vòng lặp for	<pre>&lt;?php for(\$i=start; \$i&lt;end; \$i+=step_number): echo \$i; endfor; ?&gt;</pre> <p>Ví dụ</p> <pre>&lt;?php for(\$x=0; \$x&lt;20; \$x+=2): echo \$x+1; endfor; ?&gt;</pre>	<pre>{for \$i = start to end step step_number} {\$i} {/for}</pre> <p>Ví dụ</p> <pre>{for \$x = 1 to 20 step 2} {\$x} {/for}</pre>
Đếm phần tử của mảng	<pre>&lt;?php echo count(\$tên_mảng);?&gt;</pre> <p>Ví dụ:</p> <pre>&lt;?php echo count(\$myArray);?&gt;</pre>	<pre>{\$tên_mảng @count} Ví dụ: {\$myArray @count}</pre>





## Smarty Template

### □ Cú pháp cơ bản của Smarty

Nội dung	PHP	Smarty
if	<pre>&lt;?php if(isset(\$foo)) { ... }?&gt;</pre>	<pre>{if isset(\$foo) } ... {/if}</pre>
if, elseif	<pre>&lt;?php if(isset(\$name) &amp;&amp; \$name == 'Blog'){ ... } elseif(\$name == \$foo) { ... }</pre>	<pre>{if isset(\$name) &amp;&amp; \$name == 'Blog'} ... {elseif \$name == \$foo} ... {/if}</pre>
Chèn file	<pre>&lt;?php include('đường dẫn và tên file');?&gt; Ví dụ &lt;?php include('header.tpl');?&gt;</pre>	<pre>{include file='đường dẫn và tên file'} Ví dụ: {include file='header.tpl'}</pre>



## Smarty Template

### □ Định dạng dữ liệu

- Gọi sử dụng các phương thức định dạng dữ liệu hiển thị:
  - **Biến:**
    - Cú pháp: `{${tên_biến|tên_định_dạng}}`
    - Ví dụ: `{ $title|upper}`: Viết hoa nội dung chứa trong biến \$title
  - **Phương thức của Smarty:**
    - Cú pháp: `{ $smarty.ten_phuong_thuc|tên_định_dạng"`
    - Ví dụ: `{ $smarty.now|date_format:"%Y/%m/%d"`: In ngày hiện tại với định dạng năm – tháng – ngày
  - **Chú ý:** Khi sử dụng cùng lúc nhiều định dạng thì ngăn cách nhau bằng dấu |
    - Ví dụ: `{ $myArray|upper|truncate:20}`





## Smarty Template

### □ Định dạng dữ liệu

- Một số phương thức định dạng chuỗi:
  - Viết hoa đầu từ: {`$articleTitle|capitalize`}
  - Ráp chuỗi: {`$articleTitle|cat:' yesterday.'`}
  - Viết thường: {`$articleTitle|lower`}
  - Viết hoa: {`$articleTitle|upper`}
  - Xuống dòng khi gặp “\n”: {`$articleTitle|nl2br`}
  - Tìm kiếm và thay thế:  
{`$articleTitle|replace:'Garden':'Vineyard'`}



## Smarty Template

### □ HTML Tag

- HTML Image: hiển thị image

```
{html_image file='pumpkin.jpg'}
{html_image file='../path/from/docroot/pumpkin.jpg'}
{html_image file='../path/relative/to/currdir/pumpkin.jpg'}
```

- HTML options: tạo list/menu

PHP code:

```
<?php
$smarty->assign('myOptions', array(1800 => 'Joe Schmoe', 9904 => 'Jack
Smith', 2003 => 'Charlie Brown'));
$smarty->assign('mySelect', 9904);
?>
```

Smarty Template:

```
{html_options name=foo options=$myOptions selected=$mySelect}
```





LTV Mã nguồn mở PHP - Module 3

26



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

### Module 3 – Bài 9: *Smarty – Template Engine 2*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## PHP code và Smarty class



### □ Hằng SMARTY\_DIR

- Khai báo hằng số SMARTY\_DIR: đường dẫn chứa thư viện Smarty

```
define('SMARTY_DIR', 'đường dẫn đến thư mục  
Smarty/libs/');
```

- Gọi sử dụng lớp Smarty.class của thư viện Smarty:

```
require_once(SMARTY_DIR . 'Smarty.class.php');
```

- Ví dụ:

```
define('SMARTY_DIR', '../Smarty/libs/');  
require_once(SMARTY_DIR . 'Smarty.class.php');
```





## PHP code và Smarty class

### ❑ Biến của Smarty class

- \$template\_dir: khai báo đường dẫn của thư mục templates chứa các file tpl
- \$compile\_dir: khai báo đường dẫn của thư mục templates\_c chứa các file biên dịch template của ứng dụng
- \$config\_dir: khai báo đường dẫn của thư mục configs chứa các file cấu hình của ứng dụng
- \$cache\_dir: khai báo đường dẫn của thư mục cache chứa các file được lưu trữ tạm
- **Chú ý:** cần phải thiết lập giá trị cho các biến khi sử dụng



## PHP code và Smarty class

### ● Ví dụ

```
$smarty = new Smarty();

$smarty->template_dir = '../application/templates/';
$smarty->compile_dir = '../application/templates_c/';
$smarty->config_dir = '../application/configs/';
$smarty->cache_dir = '../application/cache/';
```







## PHP code và Smarty class

### ☐ Phương thức của Smarty class

- append() - Đưa thêm một phần tử vào mảng

- Cú pháp:

append(mixed var) hoặc

append(string varname, mixed var, bool merge)

- Ví dụ:

```
$smarty->append('foo', 'Fred'); // đưa phần tử 'Fred' vào mảng foo
```

```
$smarty->append('foo', 'Albert'); // lúc này mảng có 2 phần tử là 'Fred' và 'Albert'
```



## PHP code và Smarty class

### ☐ Phương thức của Smarty class

- assign(): đưa nội dung/ giá trị biến sang trang template (.tpl)

- Cú pháp:

assign(mixed var) hoặc

assign(string varname, mixed var)

- Ví dụ:

```
$myArray = array('no' => 10, 'label' => 'Peanuts');
```

```
$smarty->assign('foo', $myArray);
```







## PHP code và Smarty class

### ☐ Phương thức của Smarty class

- `clear_assign()` - xóa giá trị của biến đã được assign

- Cú pháp:

`clear_assign(mixed var)`

- Ví dụ:

```
$smarty->clear_assign('foo');
```



## PHP code và Smarty class

### ☐ Phương thức của Smarty class

- `config_load()` - load dữ liệu từ file config và đưa vào cho template

- Cú pháp:

`config_load(string file, string section)`

- Ví dụ:

```
// load config variables và assign chúng
$smarty->config_load('my.conf');

// load 1 section
$smarty->config_load('my.conf', 'foobar');
```





## PHP code và Smarty class

### ☐ Phương thức của Smarty class

- display() – hiển thị template

- Cú pháp:

display(string template, string cache\_id, string compile\_id)

- Ví dụ:

```
$smarty->display('index.tpl');
```



## PHP code và Smarty class

### ☐ Phương thức của Smarty class

- fetch() – trả về hiển thị của template

- Cú pháp:

fetch(string template, string cache\_id, string \$compile\_id)

- Ví dụ:

```
// lấy nội dung của template
$output = $smarty->fetch('index.tpl');
// hiển thị output
echo $output;
```





## ❑ Phương thức của Smarty class

- `template_exists()` - kiểm tra sự tồn tại của một template cụ thể

- Cú pháp:

`template_exists(string template)` // kết quả trả về là true hoặc false

- Ví dụ:

```
// giả sử tên file template index.inc.tpl  
$mid_template = $_GET['page'].'.inc.tpl';  
$smarty->template_exists($mid_template);
```





## Caching – bộ nhớ đệm

### □ Giới thiệu

- Bộ nhớ đệm được sử dụng để tăng tốc độ gọi sử dụng phương thức `display()` hoặc `fetch()` bằng cách lưu trữ dữ liệu xuất ra file.
- Nếu đã có một phiên bản được cache thì nó sẽ tự hiển thị thay vì phải tạo ra một dữ liệu xuất khác.
- Caching có thể đưa tốc độ tăng lên rất nhiều, nhất là đối với những template có thời gian tính toán dài. Khi dữ liệu xuất của `display()` hoặc `fetch()` được lưu vào bộ nhớ đệm, một file cache có thể chứa nội dung thực hiện của một số file template và file config...



## Caching – bộ nhớ đệm

### □ Giới thiệu

- Khi các mẫu template động thì cần phải chú ý cân nhắc xem nội dung nào được cache và trong thời gian bao lâu.
- Ví dụ, ta đang hiển thị front page của website mà không thay đổi nội dung của nó thường xuyên, sẽ rất tốt nếu lưu cache trang này trong một giờ và có thể lâu hơn. Ngược lại, nếu một trang mà nội dung của nó bao gồm các thông tin mới thay đổi thường xuyên thì việc sử dụng cache sẽ không có ý nghĩa.





## Caching – bộ nhớ đệm

### ❑ Thiết lập cache

- Bằng cách đặt caching = 1 hoặc bằng 2
- Khi caching = 1

```
require('Smarty.class.php');  
$smarty = new Smarty;  
$smarty->caching = 1;  
$smarty->display('index.tpl');
```



## Caching – bộ nhớ đệm

- Khi caching = 2: có thiết lập thời gian cho mỗi cache

```
require('Smarty.class.php');  
$smarty = new Smarty;  
$smarty->caching = 2;  
// thiết lập cache_lifetime cho trang index.tpl là 5 phút  
$smarty->cache_lifetime = 300;  
$smarty->display('index.tpl');  
// thiết lập cache_lifetime cho trang home.tpl là 1 giờ  
$smarty->cache_lifetime = 3600;  
$smarty->display('home.tpl');
```







## Caching – bộ nhớ đệm

### ❑ Kiểm tra cache

- Sử dụng phương thức `is_cached()` để kiểm tra một template có được cache hay không

```
require('Smarty.class.php');
$smarty = new Smarty;
$smarty->caching = 1;
if(!$smarty->is_cached('index.tpl')) {
    // nếu chưa được cache
    $contents = get_database_contents();
    $smarty->assign($contents);
}
$smarty->display('index.tpl');
```



## Caching – bộ nhớ đệm

### ❑ Xóa cache

- Sử dụng phương thức `clear_cache()` để xóa cache

```
require('Smarty.class.php');
$smarty = new Smarty;
$smarty->caching = 1;
// chỉ xóa cache cho trang index.tpl
$smarty->clear_cache('index.tpl');
// xóa tất cả các file cache
$smarty->clear_all_cache();
$smarty->display('index.tpl');
```





## Caching – bộ nhớ đệm

### ❑ Sử dụng nhiều cache trên một trang

- Có thể sử dụng nhiều cache trên một trang

```
require('Smarty.class.php');  
$smarty = new Smarty;  
$smarty->caching = 1;  
$my_cache_id = $_GET['article_id'];  
// với mỗi giá trị $my_cache_id duy nhất, thì sẽ có một cách được  
tạo ra cho trang for index.tpl.  
$smarty->display('index.tpl', $my_cache_id);
```







## Kế thừa – Template Inheritance

### □ Giới thiệu

- Kế thừa template là một cách tiếp cận để quản lý các template giống như kỹ thuật lập trình hướng đối tượng. Thay vì sử dụng cách truyền thống là dùng thẻ {include...} để quản lý các phần của template, thì ta có thể kế thừa các nội dung của một template này cho template khác (như việc kế thừa một lớp) và thay đổi các khối nội dung (như phương thức ghi đè của class). Điều này giúp cho việc quản lý template nhẹ nhàng hơn và hiệu quả hơn, vì mỗi mẫu chỉ chứa những nội dung mới so với lớp mà nó kế thừa.



## Kế thừa – Template Inheritance

### □ So sánh template có kế thừa và không kế thừa

Template không kế thừa	Template có kế thừa
<b>header.tpl:</b> <code>&lt;html&gt;</code> <code>&lt;head&gt;</code> <code>&lt;title&gt;{\$title default:"Default Page Title"} &lt;/title&gt;</code> <code>&lt;/head&gt;</code> <code>&lt;body&gt;</code> <b>footer.tpl</b> <code>&lt;/body&gt;</code> <code>&lt;/html&gt;</code>	<b>layout.tpl</b> <code>&lt;html&gt;</code> <code>&lt;head&gt;</code> <code>&lt;title&gt;{block name=title}Default Page Title{/block}</code> <code>&lt;/title&gt;</code> <code>&lt;/head&gt;</code> <code>&lt;body&gt;{block name=body}/{/block}</code> <code>&lt;/body&gt;</code> <code>&lt;/html&gt;</code>
<b>mypage.tpl</b> <code>{include file="header.tpl" title="My Page Title"}</code> My HTML Page Body goes here <code>{include file="footer.tpl"}</code>	<code>{extends file="layout.tpl"}</code> <code>{block name=title}My Page Title{/block}</code> <code>{block name=body} My HTML Page Body goes here {/block}</code>





## Kế thừa – Template Inheritance

- ❑ So sánh template có kế thừa và không kế thừa

Template không kế thừa	Template có kế thừa
<b>Nội dung xuất ra của file mypage.tpl</b> <pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;My Page Title&lt;/title&gt; &lt;/head&gt; &lt;body&gt; My HTML Page Body goes here &lt;/body&gt; &lt;/html&gt;</pre>	<b>Nội dung xuất ra của file mypage.tpl</b> <pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;My Page Title&lt;/title&gt; &lt;/head&gt; &lt;body&gt; My HTML Page Body goes here &lt;/body&gt; &lt;/html&gt;</pre>





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

## LTV MÃ NGUỒN MỞ PHP

### Module 3 – Bài 10: *Xây dựng Web Service*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Mô hình hoạt động của Web Service



### □ Gồm 3 thành phần:

- Web Service Provider – nhà cung cấp dịch vụ web
- Web Service Consumer – nhà khai thác dịch vụ web
- End-user – người dùng cuối





## Mô hình hoạt động của Web Service

### ❑ Web Service Provider

- Xây dựng và cài đặt đối tượng dịch vụ lên web server và cung cấp bản mô tả dịch vụ để hướng dẫn người sử dụng cách khai thác.
- Bản mô tả dịch vụ được viết bằng **SDL - Service Description Language**, với cấu trúc cú pháp của **XML**



## Mô hình hoạt động của Web Service

### ❑ Web Service Consumer

- Xem xét bản mô tả dịch vụ của nhà cung cấp, dựa vào đó để xây dựng lớp trung gian truy xuất đối tượng dịch vụ, và cuối cùng, thiết kế giao diện cho ứng dụng



## Mô hình hoạt động của Web Service



### □ End-user

- Truy cập trang web của các nhà khai thác dịch vụ, sử dụng (một cách gián tiếp) **Web Service** của **nhà cung cấp dịch vụ**





## Chuẩn SOAP

### ❑ SOAP - Simple Object Access Protocol

- Là một trong những giao thức "chuẩn" dựa trên nền tảng XML cho việc xây dựng Web Services. Đây là chuẩn chính thức do **W3C** đưa ra.
- **SOAP** là sự kết hợp giữa **HTTP** và **XML**



## Chuẩn SOAP



### ❑ XML - eXtensible Markup Language

- Khi sử dụng tài liệu XML thì phải định dạng đúng và hợp lệ
  - Định dạng đúng: là tuân thủ chính xác cú pháp của ngôn ngữ **XML**.
  - Hợp lệ: là tên của các phần tử trong tài liệu phải phù hợp với các quy định nêu ra trong **DTD** (Document Type Definon) hoặc **XSD** (XML Schema Definition) của nó.







## Chuẩn SOAP

- **DDT**: Là 1 file dạng text, chỉ định các phần tử nào được phép trong tài liệu XML
- **XSD**: là một file dạng XML có chức năng tương tự DTD, nhưng cho phép quy định rõ hơn về kiểu dữ liệu của các phần tử tài liệu

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schemas.xmlsoap.org/soap/envelope/"
  targetNamespace="http://localhost/shema.xml">
  <xs:element name="MathPowResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="num" type="xs:integer"/>
        <xs:element name="result" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



LTV Mã nguồn mở PHP - Module 3

10

## Chuẩn SOAP



- Các **DTD** và **XSD** được đưa vào hồ sơ XML thông qua thuộc tính **xmlns** => tạo ra các **namespace** - không gian tên.
- Nhờ các namespace, vấn đề trùng tên của các phần tử trong một tài liệu XML kết hợp từ nhiều nguồn, được giải quyết một cách dễ dàng



LTV Mã nguồn mở PHP - Module 3

11



## Chuẩn SOAP

- Ví dụ: Dịch vụ **Google News** cung cấp các bản tin dạng **RSS**, với phần tử description mô tả tóm tắt về một tin tức, trong khi **Amazon** lại cung cấp thông tin về 10 cuốn sách xuất bản gần nhất bằng một file dạng XML mà trong đó chứa các phần tử description giới thiệu về nội dung cuốn sách.
- => Nếu tổng hợp 2 tài liệu trên thành 1 tài liệu XML => đưa các namespace vào để phân biệt 2 dạng sử dụng của description trong tài liệu



LTV Mã nguồn mở PHP - Module 3

12

## Chuẩn SOAP



```
<gn:description xmlns:gn="http://news.google.com/Schema">
New York utility workers searched through the night for...
</gn:description>

<am:description xmlns:gn="http://amazon.com/Schema">
Here in our Harry Potter store we have something for everyone...
</am:description>
```

- gn và am ở đây là tên rút gọn (short name) của các **namespace**. Chúng nằm trước tên của phần tử và ngăn cách với tên phần tử bằng dấu 2 chấm ":"



LTV Mã nguồn mở PHP - Module 3

13

## Chuẩn SOAP



```
<SOAP-ENV:Envelope
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<sn:MathPowResponse xmlns:sn="uri:http://localhost/shema.xml">
<sn:result>25</sn:result>
</sn:MathPowResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

*(Giáo viên giải thích chi tiết các nội dung trong ví dụ trên cho HV)*





## Xây dựng Web Service với NuSOAP

### □ NuSOAP:

- Là một thư viện PHP cho phép các developer tạo và sử dụng các SOAP web service. Nó không yêu cầu bất kỳ một phần mở rộng đặc biệt nào của PHP.
- Hỗ trợ SOAP và có thể tạo ra cũng như sử dụng WSDL
- Cả rpc/encoded và document/literal service được hỗ trợ



LTV Mã nguồn mở PHP - Module 3

16



## Xây dựng Web Service với NuSOAP

### □ Tạo Web Service

- Bước 1:
  - Download gói NuSOAP.zip tại địa chỉ <http://sourceforge.net/projects/nusoap/>
  - Tạo ứng dụng làm việc (Ví dụ: FirstWebService)
  - Giải nén gói NuSOAP (thành thư mục libs) và để trong ứng dụng (FirstWebService/libs)
- Bước 2: Tạo tập tin service.php và lưu cùng nơi với thư mục libs chứa những nội dung của Web Service dự định làm.



LTV Mã nguồn mở PHP - Module 3

17

## Xây dựng Web Service với NuSOAP

- Ví dụ: tạo một dịch vụ Services, với phương thức TongHaiSo để tính tổng của hai số bất kỳ > Nội dung của file service.php như sau:

```
// Pull in the NuSOAP
require_once('lib/nusoap.php');
// Create the server instance
$server = new soap_server();
// Initialize WSDL support
$server->configureWSDL('Service', 'urn:Service');
// Character encoding
$server->soap_defencoding = 'utf-8';
//-----
//Registrations of our functions
$server->register('TongHaiSo',
    array('a'=>'xsd:int', 'b'=>'xsd:int'),
    array('tong'=>'xsd:int'),
    'urn:ServiceWSDL',
    'urn:ServiceWSDL#TongHaiSo',
    'rpc',
    'literal',
    'Tong Hai So');
```

```
//-----
//Our web service functions will be here.

function TongHaiSo($a,$b)
{
    if(is_numeric($a) && is_numeric($b))
    {
        $response=$a+$b ;
        return $response;
    }
    else
        return "";
}
//-----

$HTTP_RAW_POST_DATA =
isset($HTTP_RAW_POST_DATA) ?
$HTTP_RAW_POST_DATA : "";
$server->service($HTTP_RAW_POST_DATA);
```



(Giáo viên giải thích chi tiết các nội dung trong ví dụ trên cho HV)

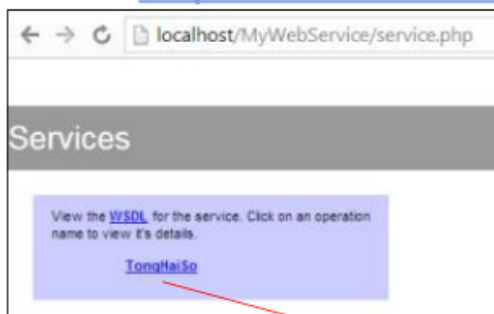
LTV Mã nguồn mở PHP - Module 3

18

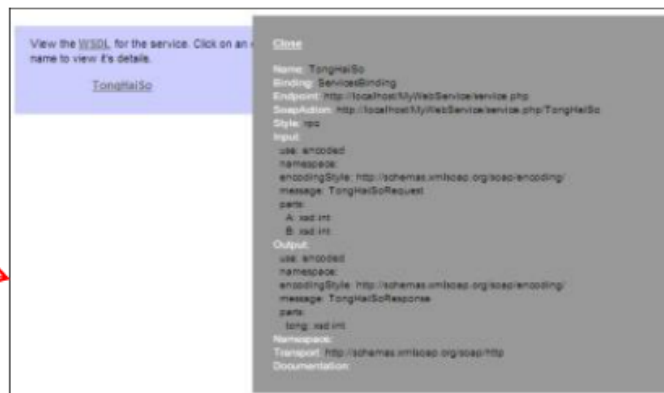
## Xây dựng Web Service với NuSOAP

- Bước 3: Kiểm tra phương thức đã thực hiện bằng cách gọi service đã làm (Ví dụ:

<http://localhost/FirstWebService/service.php>



(Giáo viên giải thích chi tiết cách thức làm việc trong kết quả trên cho HV)



LTV Mã nguồn mở PHP - Module 3

19



## Xây dựng Web Service với NuSOAP

### □ Khai thác Web Service

- Bước 1: Tạo một tập tin chứa những nội dung để gọi sử dụng Web Service vừa xây dựng.



## Xây dựng Web Service với NuSOAP

- Ví dụ: Tập tin TongHaiSo.php để gọi sử dụng web service tính tổng hai số

```
<?php
require_once ('lib/nusoap.php');
//Give it value at parameter
$params = array( 'a' => 5, 'b' => 10);
$namespace = "http://localhost/phpServices/services.php";
//Create object that referer a web services
$client = new soapclient($namespace);
//Call a function at server and send parameters too
$response = $client->call('TongHaiSo',$params);
echo $response;
?>
```

**(Giáo viên giải thích chi tiết các nội dung trong ví dụ trên cho HV)**





## Xây dựng Web Service với NuSOAP

- Bước 2: Kiểm tra công việc đã làm (Ví dụ: gõ địa chỉ <http://localhost/FirstWebService/TongHaiSo.php>)

