

LẬP TRÌNH WEB ĐỘNG VỚI PHP / MySQL

- ❖ GUESTBOOK
- ❖ CATALOG
- ❖ FORUM
- ❖ SHOPPING CART

PHẦN 2

Tống Phước Khải (tổng hợp & biên dịch)

Phương pháp truy xuất CSDL

MySQL

- 1- Từ Command Prompt
- 2- Bằng lệnh PHP
- 3- Dùng phpMyadmin

Để hiểu được chương này một cách rõ ràng, trước tiên bạn cần phải có một số kiến thức cơ bản về **Cơ Sở Dữ Liệu quan hệ**. Nếu bạn đã học qua một khoá căn bản về MS Access trong chương trình đào tạo chứng chỉ B chẳng hạn thì hẳn nhiên bạn có thể tiếp tục. Còn nếu như bạn chưa biết gì về nó thì tôi sẽ bàn đến nó trong phần Phụ Lục của giáo trình này hoặc bạn có thể tìm ngay một tài liệu tham khảo về CSDL, dễ nhất là tài liệu về MS Access ...

... Tôi chắc rằng bây giờ bạn đã có kiến thức về CSDL và hiểu biết Table là gì rồi! Có hàng khối công việc bạn sẽ phải làm việc đối với các Table và bạn sẽ được hướng dẫn cặn kẽ trong quyển sách này. Bạn sẽ phải vượt qua một số kiến thức về nó để mới có thể thành thạo trong thao tác với Table. Như bạn biết đấy: *Con đường đi đến thành công không có trải thảm sẵn đâu!*

Nếu bạn đã từng làm việc với *MS SQL Server* hay *Access* chúng đều có hỗ trợ việc tạo CSDL rất là dễ dàng với giao diện trực quan. Đối với MySQL bạn cũng có thể sử dụng công cụ trực quan đó là *phpMyadmin*.

Tuy nhiên, bạn phải học cách thao tác với CSDL bằng dòng lệnh, tôi chắc rằng điều này sẽ rất có ích cho bạn. Muốn chương trình của bạn trong lúc chạy thao tác tự động với CSDL thì bạn cần hàng tá lệnh PHP/SQL để thực hiện các yêu cầu của chương trình.

Trước khi chúng ta tạo các table trong CSDL của MySQL, có một vài thứ bạn cần phải hiểu rõ. Những khái niệm cơ bản mà tôi sắp giới thiệu sau đây rất quan trọng. Bạn hãy chắc rằng mình đã nắm kỹ về chúng trước khi thực hiện việc thiết kế dữ liệu.

Null

Việc đầu tiên bạn phải làm trong việc thiết kế một table là quyết định xem một field có cho phép giá trị NULL hay không.

Trong CSDL quan hệ, giá trị NULL của một field đồng nghĩa với nó có thể chấp nhận không có dữ liệu trong đó. Nên nhớ rằng giá trị NULL khác với giá trị của một chuỗi không có ký tự trong đó hoặc số có giá trị 0.

Đôi khi trong chương trình, bạn sẽ thực hiện một số động tác so sánh xem một chuỗi nào đó có chứa giá trị hay không, nó có thể là một câu lệnh IF. Xét một ví dụ PHP như sau:

```
$var //this is a variable used in the test
if ($var == "")
{
echo "Var is an empty string";
} else {
echo $var;
}
```

Nếu bạn thực hiện việc so sánh xem giá trị số có phải 0 hay không thì cũng thực hiện tương tự.

Động tác so sánh trên sẽ không làm việc được đối với giá trị NULL. Bạn nên hiểu rằng NULL là không chứa bất kỳ giá trị gì trong đó, cho nên việc so sánh trị không mang ý nghĩa gì cả. Trong chương 3 bạn sẽ thấy rằng đối với giá trị NULL đòi hỏi lập trình viên phải rất cẩn nhắc khi viết lệnh liên kết table.

Trong lệnh SELECT của SQL, có một số cách để bạn có thể kiểm tra nếu như một field chứa giá trị NULL. Trước hết bạn hãy sử dụng hàm Isnull(). Giả sử tìm một record trong table mà giá trị middle_name là NULL, bạn có thể sử dụng query sau:

select * from names where isnull(middle_name);

Hoặc lấy các record mà middle_name khác NULL:

select * from names where !isnull(middle_name);

Bạn cũng có thể sử dụng is null và is not null:

select * from users where addr2 is null;

select * from users where addr2 is not null;

Để rõ hơn, bạn hãy xem chuyện gì xảy ra khi tôi cố gắng liên kết hai table sau:

Khach_hang

ten	ho_lot	ba_xa
Trung	Le Anh	1321
Khai	Tong Phuoc	Null

Hon_nhan

ba_xa	ten	ho_lot
1321	Diem	Nguyen Thuy

Nếu bạn muốn tìm tên các **khách hàng** và tên những **bà xã** của họ, bạn sẽ phải liên kết 2 table này thông qua field ba_xa. (Xin bạn chớ lo lắng khi chưa hiểu về cú pháp, bạn sẽ học ngay ở phần tiếp theo thôi).

```
SELECT * FROM khach_hang, hon_nhan  
WHERE khach_hang.ba_xa = hon_nhan.ba_xa
```

Việc thực hiện này chỉ đúng đối với Trung, nhưng sẽ có vấn đề đối với Khai bởi vì anh ta hãy còn độc thân và ba_xa của anh ta là NULL.

Trong chương 3 bạn sẽ khảo sát kỹ hơn về vấn đề này.

Index

Người ta nói rằng ưu điểm vượt trội của Hệ quản trị CSDL quan hệ là nó thực hiện các việc tìm kiếm hay sắp xếp những khối lượng dữ liệu khổng lồ một cách rất nhanh chóng. Sở dĩ nó thực hiện được việc này là do nó có chứa một cơ cấu lưu trữ dữ liệu gọi là INDEX.

INDEX cho phép database server tạo được một field đặc trưng tìm kiếm với tốc độ khó ngờ. Các INDEX đặc biệt hỗ trợ một hoặc một nhóm các record trong một table chứa số lượng lớn các record. Chúng cũng hỗ trợ tốc độ cho các hàm liên kết hoặc tách nhóm dữ liệu như min(), max(), bạn sẽ tìm hiểu ở Chương 3.

Với các tính năng vượt trội này, tại sao người ta lại không tạo index trong tất cả các field của một table? Có một số điều trở ngại như sau: Thứ nhất, index có sẽ làm chậm một số tiến trình trong CSDL. Mỗi lần bảo trì các index Database Server phải mất khá nhiều thời gian. Có một vài trường hợp chính các index làm cho chúng chậm hẳn. Nếu như trên table của bạn tất cả các record đều giống y như nhau thì không có lý do gì để bạn tạo index. Các index dư thừa chỉ làm cho tốn thêm không gian đĩa của bạn mà thôi.

Đối với một table có gắn khoá chính (primary key) thì thông thường field có khoá này được dùng vào việc tìm kiếm cho nên index sẽ được gán tự động trên field này. Bạn sẽ gặp lại rất nhiều công việc tạo index ở các phần tiếp theo sau.

Lệnh CREATE Database

Trước khi tạo được một Table thì điều tất yếu là bạn phải tạo được một Database cái đã. Việc này dễ dàng và nhanh chóng thôi. Lệnh CREATE được sử dụng như sau:

```
mysql> create database database_name;
```

Nếu như bạn thắc mắc rằng sau khi tạo database nó sẽ nằm ở trong thư mục nào trên ổ đĩa của bạn thì bạn hãy tìm trong `..\mysql\data` xem có không.

Khi đặt tên cho database, hay đặt tên cho field và index gì đấy tránh trường hợp đặt những cái tên khó nhớ hoặc dễ bị lẫn lộn. Đối với một số hệ thống Unix chẳng hạn có sự phân biệt chữ HOA/thường thì CSDL chạy trên nó cũng ảnh hưởng theo.

Bạn hãy chọn một quy ước cho riêng mình trong việc đặt tên để khỏi nhầm lẫn về sau. Chẳng hạn tên của table và field đều đặt chữ thường chẳng hạn. Nên nhớ là không được sử dụng khoảng trắng.

Bây giờ bạn tìm hiểu cả hai cách tạo database: Cách thứ nhất tạo thủ công từ dấu nhắc dòng lệnh DOS, cách thứ hai sử dụng các lệnh trong PHP.

Cách thứ nhất tôi đã có trình bày ở chương giới thiệu và bạn đã tạo một database tên là guestbook. Cú pháp tạo như sau:

```
mysql> create database guestbook;
```

Cách thứ hai là sử dụng lệnh trong PHP, bạn có thể dùng hàm `mysql_create_db()` hoặc `mysql_query()`. Nhưng nên nhớ trước khi tạo bạn phải thực hiện được kết nối với database server.

```
$conn = mysql_connect("localhost", "username", "password")  
or die ("Could not connect to localhost");  
mysql_create_db("my_database") or  
die ("Could not create database");  
$string = "create database my_other_db";  
mysql_query($string) or  
die(mysql_error());
```

Lệnh USE Database

Sau khi đã tạo được một database mới trong database server bạn sẽ bắt đầu chọn nó để sử dụng cho công việc của mình. Cách thực hiện như sau:

1. Command Prompt:

```
mysql> use database_name;
```

2. Trong PHP:

```
$conn = mysql_connect("localhost", "username", "password")  
or die ("Could not connect to localhost");  
mysql_select_db("test", $conn) or  
die ("Could not select database");
```

Lệnh CREATE Table

*Lưu ý: Lệnh này thực hiện sau khi đã có lệnh **CREATE Database**.*

Một khi bạn đã tạo và chọn database, việc tiếp theo là tạo một table. Bạn sẽ sử dụng lệnh Create Table như sau:

```
create table table_name  
(  
column_1 column_type column_attributes,  
column_2 column_type column_attributes,  
primary key (column_name),  
index index_name(column_name)  
)
```

Đối với thuộc tính các field (cột) chúng ta cần bàn về:

- **null** hoặc **not null**
- **default**

Nếu bạn không định nghĩa NULL hay NOT NULL thì NULL sẽ được chọn làm giá trị mặc định. Hãy xét ví dụ sau:

```
create table topics2 (  
topic_id integer not null auto_increment,  
parent_id integer default 0 not null,  
root_id integer default 0,  
name varchar(255),  
description text null,  
create_dt timestamp,  
modify_dt timestamp,  
author varchar(255) null,  
author_host varchar(255) null,  
primary key(topic_id),  
index my_index(parent_id))
```

Trong ví dụ trên bạn tạo ra một table có tên topics2, có tất cả 8 field và có 2 index, một index cho khoá chính và một cho parent_id. Type của các field trên lần lượt là integer, varchar, text, timestamp. Giá trị đứng sau default là giá trị mặc định bạn gán cho một ô trong record khi không nhập liệu vào.

Bây giờ chúng ta áp dụng các lệnh này vào một chương trình PHP để tạo table, hàm `mysql_query ()` được sử dụng:

```
$conn = mysql_connect("localhost", "username", "password") or  
die ("Could not connect to localhost");  
mysql_select_db("test", $conn) or  
die("could not select database");  
$query = "create table my_table (col_1 int not null primary key,  
col_2 text)";  
mysql_query($query) or  
die(mysql_error());
```

** Lưu ý: username và password tùy thuộc vào MySQL của bạn. Ví dụ:*

```
$conn = mysql_connect("localhost", "minhtrung", "zadfdfaked") or
```

Thông thường các nhà cung cấp host PHP sử dụng localhost, tuy nhiên một số hosting không sử dụng localhost như Yahoo chẳng hạn. Do đó bạn cần xem hướng dẫn của nhà cung cấp host.

Kiểu dữ liệu

Bây giờ chúng ta hãy bàn về các kiểu dữ liệu (type) gán cho field trong table. Có rất nhiều kiểu khác nhau chúng ta sẽ lần lượt khảo sát từng kiểu một.

Kiểu chuỗi văn bản

MySQL có 7 kiểu dành cho dữ liệu kiểu chuỗi văn bản:

- char
- varchar
- tinytext
- text
- mediumtext
- longtext
- enum

CHAR

Cách sử dụng: `char(length)`

Length có giá trị tối đa là 255. Giả sử bạn sử dụng khai báo `char(10)` thì bạn chỉ được phép nhập vào tối đa 10 ký tự mà thôi.

VARCHAR

Cách sử dụng: `varchar(length)`

Kiểu này cũng gần giống như kiểu CHAR có độ dài tối đa cũng là 255. Điểm khác biệt của varchar là nó chỉ là biến lưu trữ độ dài, cho nên nó sẽ không thay đổi khi giá trị của của ô dữ liệu dài hay ngắn. MySQL sẽ sinh ra một ký dùng làm biến chứa độ dài của field kiểu varchar. Đồng thời MySQL sẽ thực hiện chức năng loại bỏ các khoảng trống trong mỗi ô dữ liệu nếu như không được sử dụng hết.

USING CHAR OR VARCHAR

Có sự khác nhau trong việc sử dụng CHAR và VARCHAR. Sau đây là phương hướng lựa chọn của bạn.

Giả dụ bạn tạo một field là ĐỊA CHỈ và bạn dự tính độ dài tối đa là 150. Có những trường hợp địa chỉ rất ngắn ví dụ: *1 Lê Lợi, Q.1, TPHCM*. Bạn chỉ sử dụng có 20 ký tự, như vậy còn

trống rất nhiều ký tự không dùng đến. Trong trường hợp này bạn nên sử dụng kiểu VARCHAR (150).

Trường hợp field của bạn là MÃ SỐ chẳng hạn, và bạn cho độ dài tối đa là 6 theo quy ước tạo mã của bạn ví dụ: KH0001 . Trong trường hợp này các ô khác đều được nhập theo chuẩn định sẵn luôn luôn là 6 ký tự cho nên không việc gì bạn phải sử dụng VARCHAR để MySQL phải nhọc công theo dõi độ dài của các ô mỗi khi nhập vào. Bạn chỉ sử dụng VAR(6) là được.

Trong trường hợp bạn chọn kiểu dữ liệu là varchar(4) thì MySQL sẽ tự động đổi lại là kiểu char.

TINYTEXT

Cách sử dụng: tinytext

Đây là một trong bốn kiểu text nhị phân. Tất cả 4 kiểu này (tinytext, text, mediumtext, largertext) đều là kiểu biến tương tự như varchar. Chúng khác nhau về độ dài của cho phép của ký tự mà thôi. Tuy nhiên, đối với TINYTEXT thì cho phép độ dài tối đa là 255, giống như varchar(255). Cho phép tạo index trên toàn bộ các ký tự của field này.

TEXT

Cách sử dụng: `text`

Cho phép độ dài tối đa là 65,535 ký tự. Có thể tạo index trên 255 ký tự đầu.

MEDIUMTEXT

Cách sử dụng: `mediumtext`

Cho phép độ dài tối đa là 16,777,215 ký tự. Có thể tạo index trên 255 ký tự đầu.

LONGTEXT

Cách sử dụng: `longtext`

Cho phép độ dài tối đa là 4,294,967,295 ký tự. Có thể tạo index trên 255 ký tự đầu. Tuy nhiên loại này không thông dụng bởi vì MySQL chỉ hỗ trợ chuỗi 16 triệu bytes.

ENUM

Cách sử dụng: `enum ('value1', 'value2', 'value3' ...) [default 'value']`

Với enum bạn có thể giới hạn các giá trị được định sẵn cho một field. Cho phép bạn định trước tối đa 65.535 giá trị.

Thông thường người ta dùng kiểu này cho field chứa giá trị Yes hoặc No. Ví dụ:

```
create table my_table (  
id int auto_increment primary key,  
answer enum ('yes', 'no') default 'no'  
);
```

SET

Cách sử dụng: set ('value1', 'value2', 'value3' ...) [default 'value']

Kiểu này định nghĩa một tập hợp hàng loạt các giá trị định trước. Tuy nhiên, cách này ít được dùng bởi vì nó phá vỡ cấu trúc thiết kế CSDL (một field có quá nhiều kiểu) và các bạn sẽ không thấy tôi sử dụng trong quyển sách này.

Kiểu dữ liệu số

MySQL có tất cả 7 kiểu số. Lưu ý rằng các kiểu sau đây là giống nhau: int/ integer, double/double precision/real, decimal/numeric

- ❑ int/integer
- ❑ tinyint
- ❑ mediumint
- ❑ bigint

- ❑ float
- ❑ double/double precision/real
- ❑ decimal/numeric

Đối với tất cả kiểu số, giá trị lớn nhất cho phép là 255. Đối với một số kiểu cho phép bạn thể hiện các số 0 đứng đầu. Giả sử bạn có một field có độ dài là 10 thì số 25 trong ô dữ liệu sẽ được thể hiện 0000000025. Field số còn được định nghĩa là có dấu (signed) hoặc không dấu (unsigned). Mặc định là có.

INT/INTEGER

Cách sử dụng: int(display size) [unsigned] [zerofill]

Nếu bạn dùng không dấu thì giá trị của field cho phép là từ 0 đến 4.294.967.295. Nếu dùng có dấu thì giá trị từ -2.147.483.648 đến 2.147.483.647. Kiểu Int sẽ sử dụng auto_increment (tự động theo chiều tăng) để định nghĩa khoá chính của table.

```
create table my_table (  
table_id int unsigned auto_increment primary key,  
next_column text  
);
```

Đề ý rằng bạn sử dụng không dấu (unsigned) bởi vì auto_increment không dùng cho các giá trị âm.

TINYINT

Cách sử dụng: `tinyint(display size) [unsigned] [zerofill]`

Nếu không dấu, tinyint sẽ chứa các giá trị nguyên từ 0 đến 255. Nếu có dấu thì từ -128 đến 127.

MEDIUMINT

Cách sử dụng: `mediumint(display size) [unsigned] [zerofill]`

Có dấu: có giá trị từ -8.388.608 đến 8.388.607

Không dấu: có giá trị từ 0 đến 1677215

BIGINT

Cách sử dụng: `bigint(display size) [unsigned] [zerofill]`

Có dấu: -9.223.372.036.854.775.808 đến 9.223.372.036.854.775.807

Không dấu: from 0 to 18.446.744.073.709.551.615

FLOAT

Cách sử dụng 1: `FLOAT(precision) [zerofill]`

Với cách sử dụng này, cho phép chứa các số thập phân không dấu. Số lượng phần thập phân có thể là ≤ 24 đối với loại single và 25 đến 53 đối với loại double. Các version trước đây của MySQL, luôn chia làm 2 loại:

Cách sử dụng 2: `FLOAT[(M,D)] [ZEROFILL]`

Đây là loại single và giá trị có thể là từ $-3,402823466E+38$ đến $-1,175494351E-38$, số 0, và từ $1,175494351E-38$ đến $3,402823466E+38$. M là phần nguyên, D là phần thập phân. ????

DOUBLE/DOUBLE PRECISION/REAL

Cách sử dụng 1: `DOUBLE[(M,D)] [zerofill]`

Cho phép giá trị từ $-1,7976931348623157E+308$ đến $-2,2250738585072014E-308$, số 0 và $2,2250738585072014E-308$ đến $1,7976931348623157E+308$. M là phần nguyên, D là phần thập phân.

Cách sử dụng 2: `DECIMAL[(M[,D])] [ZEROFILL]`

Các số trong phần thập phân được lưu trữ như ký tự. Mỗi số được xem như một ký tự chuỗi. Nếu $D = 0$ thì sẽ không có phần thập phân. Giá trị thập phân giống như dạng Double. ????

Kiểu dữ liệu ngày, giờ

MySQL có 5 dạng ngày giờ:

- ❑ date_time
- ❑ datetime_year
- ❑ timestamp
- ❑ time
- ❑ year

Ngày và giờ trong MySQL rất uyển chuyển, nó có thể chấp nhận kiểu chuỗi hoặc số hãy xét ví dụ sau:

```
create table date_test(  
id int unsigned auto_increment,  
a_date date  
);
```

Sau đó dùng insert để đưa giá trị ngày vào a_date:

```
insert into date_test (a_date) values ('00-06-01');  
insert into date_test (a_date) values ('2000-06-01');
```

```
insert into date_test (a_date) values ('20000601');  
insert into test6 (a_date) values (000601);
```

MySQL tương thích với việc nhận giá trị ngày là kiểu chuỗi hơn. Cho nên "000501 là chọn lựa thích hợp hơn là việc nhập một số nguyên. Sử dụng giá trị chuỗi cho ngày giúp bạn sẽ tránh được một số sự cố về sau.

MySQL có hỗ trợ một số hàm giúp bạn trong việc rút trích dữ liệu dạng ngày.

DATE

Cách sử dụng: date

Định dạng của ngày như sau: YYYY-MM-DD. Cho phép bạn các giá trị từ 1000-01-01 đến 9999-12-31.

DATETIME

Cách sử dụng: datetime [null | not null] [default]

Định dạng của ngày giờ như sau: YYYY-MM-DD HH:MM:SS. Cho phép bạn các giá trị từ 1000-01-01 00:00:00 đến 9999-12-31 23:59:59.

TIMESTAMP

Cách sử dụng: timestamp(size)

Đây là kiểu dữ liệu ghi nhận tự động giờ giấc sửa đổi gần nhất đối với một record, bất khi khi nào nó được tạo ra, hoặc cập nhật đều xảy ra việc ghi nhận này. Size của nó có thể định nghĩa trong khoảng từ 2 đến 14. Bảng sau trình bày các size. Giá trị mặc định là 14.

Size	Định dạng
2	YY
4	YYMM
6	YYMMDD
8	YYYYMMDD
10	YYMMDDHHMM
12	YYMMDDHHMMSS
14	YYYYMMDDHHMMSS

TIME

Cách sử dụng: `time`

Lưu trữ dạng giờ theo định dạng HH:MM:SS và có giá trị từ -838:59:59 đến 838:59:59. Lý do mà giá trị này lớn như vậy là để nó có thể chứa được các kết quả tính toán giờ giấc.

YEAR

Cách sử dụng: `year[(2|4)]`

Chứa dữ liệu dạng năm. Nếu sử dụng hai ký tự để biểu thị năm thì biểu diễn được từ Có giá trị từ 1970 cho đến 2069, nên nhớ: 70 đến 99 biểu thị từ 1970 đến 1999, còn 01 đến 69 biểu thị từ 2001 đến 2069.

Dùng 4 ký tự thì biểu diễn được từ 1901 đến 2155.

Tạo chỉ mục INDEX

Bắt đầu từ phiên bản 3.23.6 của MySQL bạn có thể tạo index trên bất kỳ field nào. Cho phép 1 table có 16 field chứa index. Cú pháp như sau:

```
index index_name (indexed_column)
```

Mặc dù index_name là tùy chọn, nhưng bạn nên luôn luôn cho nó một cái tên. Sau này nó rất cần thiết khi bạn muốn xoá bỏ index của một field nào đó trong lệnh SQL của bạn. Nếu bạn không cho tên thì MySQL sẽ chọn tên index của field đầu tiên.

Còn một cách nữa để tạo index là khai báo khoá chính trên field đó. Chú ý rằng bất kỳ field auto_increment (sort tự động) cũng phải được index, và bạn nên khai báo nó là khoá chính. Trong ví dụ sau id_col được index:

```
create table my_table (  
id_col int unsigned auto_increment primary key,  
another_col text  
);
```

Khoá chính có thể cũng được khai báo giống như các index khác ngay sau khi định nghĩa một field:

```
create table my_table (  
id_col int unsigned not null auto_increment,  
another_col text,  
primary key(id_col)  
);
```

Index có thể trải rộng ra hơn một cột. Nếu như query sử dụng 2 hàng phối hợp với nhau trong khi thực hiện việc tìm kiếm, bạn có thể tạo một index bao gồm luôn cả 2 với các lệnh sau:

```
create table mytable(  
id_col int unsigned not null,  
another_col char(200) not null,  
index dual_col_index(id_col, another_col)  
);
```

Index này sẽ được sử dụng cho việc tìm kiếm vừa trên id_col vừa trên another_col. Các index này làm việc từ trái sang phải. Do đó index này sẽ được sử dụng cho việc tìm kiếm exclusive trên id_col. Tuy nhiên, nó sẽ không exclusive cho việc tìm kiếm trên another_col.

Còn một điều về index nữa là bạn có thể tạo nó chỉ trên một phần của field. Bắt đầu từ phiên bản 3.23 của MySQL bạn có thể tạo index các field kiểu index tinytext, text, mediumtext và longtext trên 255 ký tự đầu. Đối với char và varchar, bạn có thể tạo index trên một số ký tự đầu của field. Cú pháp của nó như sau:

index index_name (column_name(column_length))

Ví dụ:

```
create table my_table(  
char_column char (255) not null,  
text_column text not null,  
index index_on_char (char_column(20)),  
index index_on_text (text_column(200))  
);
```

Một index có thể đảm bảo giá trị duy nhất tồn tại trong mọi record của table bằng cách sử dụng lệnh *unique*.

```
create table my_table(  
char_column char (255) not null,  
text_column text not null,  
unique index index_on_char (char_column));
```

Các loại Table

MySQL hỗ trợ các dạng table sau: ISAM, MyISAM, BDB và Heap. ISAM là dạng table đã xưa và trong các ứng dụng mới không được sử dụng. Dạng table mặc định là MyISAM. Cú pháp để khai báo loại table này là:

```
create table table_name type=table_type(  
col_name column attribute  
);
```

Table dạng MyISAM có tốc độ tốt và tính ổn định cao. Không cần thiết phải định nghĩa một dạng table mới nào khác trừ phi bạn cần dùng loại khác cho vì một lý do đặc biệt nào đó.

Heap là dạng table thường trú trong bộ nhớ. Chúng không được lưu trữ trong đĩa cứng hay các thiết bị dùng để trữ tin. Cho nên nếu bị mất điện heap sẽ mất theo. Vì được trữ trong bộ nhớ nên heap có tốc độ truy cập cao. Tuy nhiên bạn chỉ dùng cho các table tạm thời trong lúc chạy chương trình.

Lệnh Alter table

Nếu như bạn muốn thay đổi các thành phần của table bạn sẽ sử dụng lệnh *alter table*. Bạn có thể thực hiện các thay đổi như: đổi tên table, field, index; thêm hoặc xoá field và index; định nghĩa lại các field và index. Cú pháp cơ bản của lệnh này là:

```
alter table table_name
```

Các lệnh còn lại tùy thuộc vào thao tác mà chúng ta sẽ bàn tiếp theo đây:

Đổi tên Table

Cú pháp như sau:

```
alter table table_name rename new_table_name
```

If you have MySQL version 3.23.27 or higher you can make use of the rename statement. The basic syntax is
rename table_name to new_table_name

Thêm và xoá cột trong Table

Khi thêm field vào trong table, bạn sẽ cần phải có những định nghĩa cần thiết cho field đó. Từ phiên bản 3.22 của MySQL cho phép bạn chọn vị trí để đặt field mới vào trong table. Chức năng này không bắt buộc.

```
alter table table_name add column column_name column attributes
```

Ví dụ:

```
alter table my_table add column my_column text not null
```

Cách sử dụng định vị một field trong table: Sử dụng lệnh first để chèn field mới vào vị trí đầu của table. After để chèn vào vị trí sau cùng trên table:

```
alter table my_table add column my_next_col text not null first  
alter table my_table add column my_next_col text not null after  
my_other_column
```

Để xoá một field, bạn thực hiện lệnh sau:

```
alter table table_name drop column column name
```

Khi thực hiện lệnh alter đối với một table, bạn chỉ nên thực hiện một thao tác đối với table mà thôi. Ví dụ: Bạn thực hiện việc xoá một index, sau đó tạo một index mới, thì không nên gộp chung vào một lệnh alter mà nên thực hiện 2 lần.

Thêm và xoá Index

Bạn có thể thêm index bằng cách sử dụng lệnh index, unique và primary key, tương tự như việc sử dụng lệnh create vậy.

```
alter table my_table add index index_name (column_name1,  
column_name2, ...)
```

```
alter table my_table add unique index_name(column_name)
```

```
alter table my_table add primary key(my_column)
```

Bạn cũng có thể bỏ index bằng cách sử dụng lệnh drop:

```
alter table table_name drop index index_name
```

```
alter table_name test10 drop primary key
```


Đổi thuộc tính của cột (field)

Thay đổi cách thành tố của field bằng lệnh change hoặc modify:

```
alter table table_name change original_column_name new_column_name  
int not null
```

Lệnh sau là sai:

```
alter table table_name change my_col2 my_col3;
```

Nếu bạn muốn chỉ đổi thuộc tính của field thì dùng lệnh change và tạo field mới cùng tên nhưng thay đổi thuộc tính. Giả dụ bạn đổi field col_1 từ kiểu char(200) sang varchar(200):

```
alter table table_name change col_1 col_1 varchar(200)
```

Với MySQL phiên bản 2.22.16 trở đi bạn có thể dùng lệnh modify:

```
alter table table_name modify 1 col_1 varchar(200)
```

Lệnh Insert

Bây giờ bạn đã biết tất cả những kiến thức cần thiết để tạo và thực hiện các thay đổi trên định dạng table, bây giờ bạn sẽ tìm hiểu cách thức để nhập thông tin vào table. Bạn sẽ thực hiện việc này bằng lệnh INSERT:

```
insert into table_name (column_1, column2, column3,...) values  
(value1, value2, value3 ...)
```

Nếu trong một field cho phép giá trị NULL bạn có thể không cần đưa vào trong lệnh INSERT.

Như bạn thấy các chuỗi tên field và giá trị đều được đặt trong dấu ngoặc đơn (). Ngoài ra giá trị kiểu chuỗi phải được bao bọc bởi dấu nháy đơn. Như vậy nếu như trong chuỗi có chứa dấu nháy đơn và dấu ngoặc đơn thì sao? Để tránh trường hợp lẫn lẫn giữa dấu chuỗi và dấu phân định nghĩa lệnh SQL có các qui định rằng muốn đưa ký hiệu đặc biệt vào chuỗi thì phải đặt chúng sau dấu \

' (single quote)

“ (double quote)

\ (backslash)

`%` (percent sign)

`_` (underscore)

Bạn có thể thoát ra khỏi dấu nháy đơn bằng cách sử dụng 2 dấu ngoặc đơn đóng mở ().
Bạn sử dụng các ký tự đặc biệt sau để thực hiện một thao tác đặc thù trong câu lệnh:

`\n` (newline)

`\t` (tab)

`\r` (carriage return)

`\b` (back space)

Nên lưu ý một điều là, bạn không cần phải lo lắng về các ký tự thoát ở đây trong khi lập trình PHP. Bạn sẽ gặp được những hàm và xác lập trong PHP dùng để thực hiện việc này một cách tự động. Hàm `addslashes()` và các xác lập trong `php.ini` sẽ hỗ trợ việc này.

Lệnh Update

Lệnh UPDATE có một chút khác biệt so với các lệnh mà chúng ta đã khảo sát, nó thực hiện thông qua lệnh WHERE. Cú pháp thông thường là:

```
update table_name set col_1=value1, col_2=value_2 where col=value
```

Xin nhắc lại lần nữa, nếu bạn muốn thêm vào một chuỗi, bạn cần phải bao nó trong các dấu nháy đơn và dấu thoát. Nên nhớ rằng lệnh WHERE trong câu lệnh UPDATE có thể thực hiện bất kỳ phép so sánh nào ở phía sau nó. Thông thường nó hay được dùng để định xác định một record đơn với khoá chính. Trong table **folks** sau ID là khoá chính.

id	fname	lname	salary
1	Don	Liu	25,000
2	Don	Corleone	800,000
3	Don	Juan	32,000
4	Don	Johnson	44,500

Câu lệnh sau sẽ tác động đến Don Corleone:

```
update folks set fname='Vito' where id=2
```

Như bạn thấy, nếu như bạn dùng lệnh UPDATE dựa trên field fname thì thật không nên chút nào, vì bạn có thể cập nhật tất cả các field trong table này.

```
update folks set fname='Vito' where fname='don'
```

Bạn cũng có thể dùng UPDATE để thực hiện việc điều chỉnh tăng lương đối với nhân viên chẳng hạn:

```
update folks set salary=50000 where salary<50,000
```

Lệnh drop table/drop database

Lệnh DROP dùng để xoá table hoặc cả database. Nên nhớ một điều là một khi bạn thực hiện lệnh này rồi thì bạn không thể khôi phục lại dữ liệu của bạn. Hãy cẩn trọng!

```
drop table table_name
```

```
drop database database_name
```

Lệnh DROP TABLE được chuyển đổi sang PHP thông qua hàm mysqlquery(). Nếu bạn muốn xoá database trong PHP, bạn cần sử dụng hàm mysql_drop_db(). Sẽ được trình bày căn kẽ ở các phần sau!

Lệnh show tables

Để trình bày một danh sách các table có trong database, bạn sử dụng lệnh SHOW TABLES. Để thực hiện được lệnh này bạn nên lưu ý là phải chọn database trước đã bằng lệnh USE DATABASE

Hình sau là kết quả của việc thực hiện lệnh SHOW TABLES từ dấu nhắc lệnh.

```
Linux Box - CRT
File Edit View Options Transfer Script Window Help
mysql> show tables;
+-----+
| Tables in test |
+-----+
| companies      |
| contacts       |
| contributions   |
| my_new_table   |
| mytime_table   |
| mytable        |
| mytable2       |
| sessions       |
| show_stuff     |
| table1         |
| test10         |
| test3          |
| test4          |
| test5          |
| test6          |
| test8          |
| topics         |
| topics_wow     |
| user_aggs      |
| users         |
+-----+
20 rows in set (0.00 sec)
```

Ready Telnet 27, 8 27 Rows, 86 Cols VT100 NUM

Start 7:35 AM

Lệnh Show table

Trong PHP, bạn có thể cho hiển thị một danh sách table bằng cách sử dụng hàm `MYSQL_LIST_TABLES()`:

```
<?
mysql_connect("localhost", "root", "");
$result = mysql_list_tables("test");
while($row = mysql_fetch_array($result))
{
echo $row[0] . "<br>\n";
}
?>
```

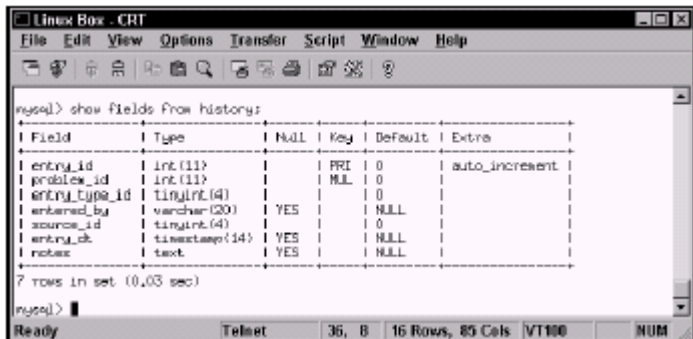
Lệnh show columns /show fields

Các lệnh cho kết quả giống nhau. Bây giờ bạn hãy dùng lệnh CREATE đã học để tạo một table sau đó dùng một trong hai lệnh trên để cho hiển thị các field của nó:

```
create table topics (
topic_id integer not null auto_increment primary key,
parent_id integer default 0 not null,
root_id integer default 0,
name varchar(255),
```


description text null,
create_dt timestamp,
modify_dt timestamp,
author varchar(255) null,
author_host varchar(255) null,
index my_index(parent_id)
)

Hình sau trình bày kết quả của lệnh SHOW FIELDS thực hiện từ dấu nhắc lệnh:



```
mysql> show fields from history;
```

Field	Type	Null	Key	Default	Extra
entry_id	int(11)		PRI	0	auto_increment
problem_id	int(11)		NULL	0	
entry_type_id	tinyint(4)			0	
entered_by	varchar(20)	YES		NULL	
source_id	tinyint(4)			0	
entry_dt	timestamp(14)	YES		NULL	
notes	text	YES		NULL	

```
7 rows in set (0.03 sec)

mysql>
```

Lệnh Show Fields

Bạn có thể cho ra kết quả tương tự trong PHP bằng cách sử dụng hàm `MYSQL_FIELD_NAME()`, `MYSQL_FIELD_TYPE()`, và `MYSQL_FIELD_LEN()`. Tất cả các hàm này được trình bày cặn kẽ ở các phần sau.

```
$db = mysql_connect("localhost","root", "")
or die ("Could not connect to localhost");
mysql_select_db("test", $db)
or die ("Could not find test");
$db_name = "topics";
$query = "select * from $db_name";
$result = mysql_query($query);
$num_fields = mysql_num_fields($result);
//create table header
echo "<table border = 1>";
echo "<tr>";
for ($i=0; $i<$num_fields; $i++)
{
echo "<th>";
echo mysql_field_name ($result, $i);
echo "</th>";
```

```
}  
echo "</tr>";  
//end table header  
//create table body  
echo "<tr>";  
for ($i=0; $i<$num_fields; $i++)  
{  
echo "<td valign = top>";  
echo mysql_field_type ($result, $i) . "<br> \n";  
echo "(" . mysql_field_len ($result, $i) . "<br> \n";  
echo mysql_field_flags ($result, $i) . "<br> \n";  
echo "</td>";  
}  
echo "</tr>";  
//end table body  
echo "</table>";
```

Sử dụng phpMyAdmin

Tôi không loại trừ trường hợp bạn là tín đầu của DẤU NHẮC LỆNH. Có khả năng là bạn không thích GIAO DIỆN ĐỒ HOẠ vì có thể bạn cho rằng giao diện này dành cho những tay mơ mà thôi. Nhưng tôi khuyên bạn rằng bạn hãy tập làm việc với GIAO DIỆN ĐỒ HOẠ. Bản thân tôi xuất thân từ thời dấu nhắc cổ lỗ, và có thể nói nếu như nhắm mắt tôi vẫn gõ được tất cả các ký tự trên phím nhanh hơn cả việc nhìn để gõ chúng. Tuy nhiên, trời phú cho bạn đôi mắt bạn hãy biết hưởng thụ khả năng của nó đừng nên bỏ qua! Và tôi cũng vậy!

Được vậy thì chúng ta hãy bắt tay vào việc sử dụng chương trình phpMyAdmin. Đây là chương trình có giao diện đồ hoạ hỗ trợ mọi truy cập trên CSDL MySQL. Nếu như bạn chưa có thì hãy chạy ra các cửa hàng ngoài Bùi Thị Xuân hay Tôn Thất Tùng kiếm ngay một đĩa đi. Không thôi thì download trên các website phpMyAdmin. Chương trình này rất dễ sử dụng. Nếu như bạn cảm thấy khó thì hãy xem phần Phụ Lục của quyển sách, tôi có hướng dẫn chi tiết trong đó.

The screenshot shows the phpMyAdmin 2.0.5 web interface running in Microsoft Internet Explorer. The browser's address bar shows the URL `http://192.168.1.1/mysqladmin/phpMyAdmin/index.php3`. The interface features a left-hand sidebar with a tree view of database structures, including `database_name`, `mysql`, and `best`. The main content area displays a "Welcome to phpMyAdmin 2.0.5" message and indicates that MySQL 3.22.27-log is running on localhost. A list of actions is provided, including "Create new database" (with a text input field and a "Create" button), "Show MySQL runtime information", "Show MySQL system variables", "Show processes", "Reload MySQL", "phpMyAdmin-Homepage", and "phpMyAdmin Documentation". The Windows taskbar at the bottom shows the Start button, system tray icons, and the time 10:41 AM.

phpMyAdmin

phpMyAdmin - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit

Address http://192.168.1.1/mysqladmin/phpMyAdmin/index.php3

Home
database_name
mysql
columns_priv
func
host
tables_priv
user
test
companies
contacts
mytable
mytable2
table1
test10
test3
test4
test5
test6
test8
topics

Database test - table companies

Select fields (at least one):

- company_id
- company_name
- company_address

• Add search conditions (body of the "where" clause):
 [Documentation]

• Do a "query by example" (wildcard: "%")

Field	Type	Value
company_id	int	<input type="text"/>
company_name	string	<input type="text"/>
company_address	string	<input type="text"/>

Start

10:40 AM

Tóm tắt

Trong chương này bạn đã tìm hiểu tất cả các lệnh cần thiết để tạo và thao tác với CSDL trong môi trường MySQL. Bạn đã biết mục đích của index. Biết các kiểu dữ liệu và mục đích sử dụng theo tùy trường hợp. Ngoài ra tôi cũng giới thiệu một chương trình quản lý CSDL có giao diện đồ họa phpMyAdmin giúp bạn thao tác thủ công trên CSDL một cách dễ dàng. Trong phần III và IV bạn sẽ tìm hiểu kỹ hơn và thực hiện những ứng dụng quan trọng của CSDL trong chương trình. **(Còn tiếp)**

Lưu ý:

Bạn có thể tìm thấy các CD thiết kế web PHP tại các cửa hàng CD (Tôn Thất Tùng, tp.HCM chẳng hạn). Trên CD có trình cài đặt Apache, PHP, MySQL thông dụng là FOX. Ngoài ra còn có các Website PHP mẫu.

CÁC LỆNH Ở BÀI HỌC TRÊN TỐT NHẤT BẠN NÊN GÕ LẠI, KHÔNG NÊN COPY!

tongphuockhai@mail15.com

- Kiến thức là kho báu không phải của riêng ai. Vì vậy bạn đừng ngần ngại khi chia sẻ cho mọi người.